OPEN SOURCE CLUB

Camal Şahverdiyev
Açıq qaynaqlı müasir həllər

Müəllif: Camal Şahverdiyev

Oxucuya müraciət:

Bu sahə üzrə Azərbaycan dilində kitab ilk dəfə nəşr olunduğundan istifadə edilən termin və sözlər məlumatın daha anlaşıla bilən olması üçün tətbiq edilmişdir. Kitabın daxilində səhv aşkar etsəniz, xahiş edirik, sərt şəkildə tənqid etməyəsiniz. Yalnış söz və ya sintaksis səhvini gördüyünüz halda, bookcorrector@gmail.com mail ünvanına yazmağınız xahiş olunur. Bununla növbəti kitabların daha mükəmməl edilməsinə yardımçı olarsınız.

ISBN: 978-9952-8290-2-0

Kitabdan istifadə qaydaları

Aşağıdakı açıqlamalar kitabın mütaliəsində oxucuya yardımçı olacaq:
Əsas başlıq - **Bold və böyük hərflər**
Əsas başlığa 1-ci dərəcəli alt başlıq - **Arxa fon qara, şrift ağ**
Əsas başlığa 2-ci dərəcəli alt başlıq - Altdan xətt

Əmrlər bold qeyd olunub. Əgər hansısa faylın içərisində olan sintaksisdən danışılırsa, öncədən faylın adı və tərkibinə əlavə ediləcək sətirlər bildirilir.

Qeydlər altdan xətt və bold edilmişdir - **Qeyd**:

- İstənilən UNIX/Linux əməliyyat sistemində faylların içində şərh üçün istifadə edilir.
Simvoldan sonrakı sözlər oxunmur.
/* şərh */ - DNS BİND-da və PHP proqramlaşdırma dilində yazılmış kodlarda göstərilən simvolların daxilində olan istənilən yazı şərhdir.
// - DNS BİND-da və PHP proqramlaşdırma dilində yazılmış kodlarda göstərilən simvollardan
sonra olan ixtiyari yazı şərhdir.
; - DNS BIND-da sətirin sonu deməkdir.

Oxucu tərəfindən kitabın başa düşülməsi üçün tələb edilən biliklər:
1. UNİX/Linux əməliyyat sistemlərində biliklərə sahib olmalı
2. CCNA şəbəkə səviyyəsinə sahib olmalıdır
3. Windows MCITP səviyyəsinə sahib olmalıdır

BÖLÜM 1

Proyektlərin idarə edilməsi sistemləri

- **Ubuntu 14.04 Redmine 3.0.1 yüklənməsi və quraşdırılması**
- **Ubuntu 14.04 x64 xWiki yuklənməsi və quraşdırılması**
- **xWiki Domain Controller ilə inteqrasiya edilməsi**

Hər hansısa bir proyektin bir neçə şöbə və ya bir neçə şirkətlə birgə kollektiv şəklində aparılmasında müəyyən problemlər ortaya çıxa bilər. Bunlardan bir neçəsini misal olaraq deyə bilərik. Məsələn sifarişçi yerinə yetirilən işin düzgün olmamasını, sifarişi qəbul edən tərəf isə əksinə görülən işin doğru olmasını bildirir və mübahisə yaranır. Bu problemlərin həlli üçün avtomatlaşdırılmış iş axını olmalıdır ki, hər iki tərəf özünə aid olan işin yazılı sübutuna sahib olsun. Başlığımız belə sistemlərin qurulmasını açıqlayır.

Ubuntu 14.04 Redmine 3.0.1 yüklənməsi və quraşdırılması

Redmine - proyektlərin və tapşırıqların idarə edilməsi(eynilə də səhvlərin izlənilməsi) üçün açıq qaynaqlı WEB proqram təminatıdır. WEB mühiti Ruby on Rails-ə əsaslır və Ruby-də yazılmışdır. Rəsmi saytı http://www.redmine.org/

Aşağıdakı bacarıqlara sahibdir:

- Proyekt və alt proyektlərin yaradılması
- Rollara əsaslanan dinamik hüquqlar sistemi
- Səhvlərin izlənilməsi sistemi
- Gantt diaqramları və təqvim
- Proyektin xəbərləri, sənədləri və fayllarının idarə edilməsinə imkan
- RSS axınlar və elektron məktubun köməkliyi ilə dəyişikliklər haqqında xəbərdarlıq
- Hər proyekt üçün wiki
- Hər proyekt üçün forum
- Müvəqqəti xərclərin hesabatı
- İnsidentlər, müvəqqəti xərclər, proyektlər və istifadəçilər üçün idarə edilən təsadüfi sütunlar
- Versiyanin idarə edilməsi(SVN, CVS, Git, Mercurial, Bazaar və Darcs) sistemləri ilə asan inteqrasiya
- Əldə edilmiş məktubların əsasında səhvlər haqqında yazıların yaradılması
- Çoxsaylı LDAP qeydiyyat metodu
- Yeni istifadəçilərin sərbəst qeydiyyatı imkanı
- Çoxdilli interfeys(həmçinin rus)
- Verilənlər bazası MySQL, Microsoft SQL Server [1], PostgreSQL, SQLite, Oracle-ın dəstəyi.

Qurulmasına başlayaq

Sistem yüklədikdə **sudo** istifadəçisi yaradılır və nəzərimizdə tuturuq ki, həmin istifadəçi adi **sysuser** və təyin etdiyimiz şifrəsini bilirik. Mütləq şəkildə bütün yüklənmə və quraşdırmaları sudo istifadəçisi adından etməliyik. Nəzərdə tutulur ki, siz Redmine-i daxili şəbəkənizdə qurursunuz və bu səbəbdən də PhpMyAdmin rahatçılıq üçün yüklənir(Əgər Public-də istifadə edəcəksinizsə, qətiyyən PhpMyAdmin yükləməyin).

Sistemi yeniləyirik:

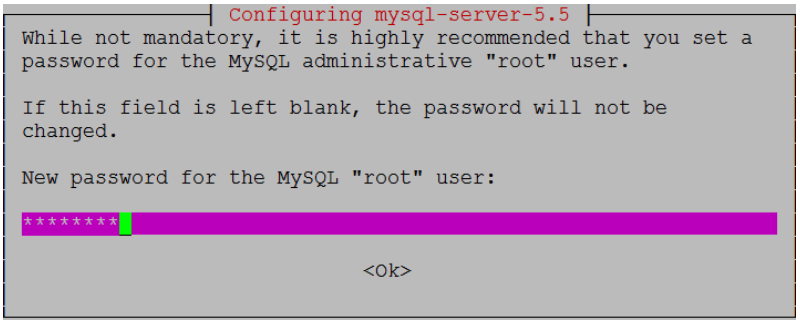
```
sysuser@redmine:~$ sudo apt-get update && sudo apt-get upgrade -y
```

LAMP üçün tələb olunan paketləri və asılılığında olan bütün paketləri yükləyirik(Yalnız sizin halda PhpMyAdmin tələb edilməyə də bilər):

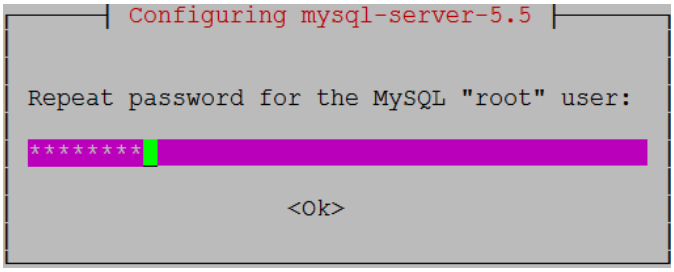
```
sysuser@redmine:~$ sudo apt-get install apache2 php5 libapache2-mod-php5
mysql-server php5-mysql phpmyadmin libapache2-mod-perl2 libcurl4-openssl-dev
libssl-dev apache2-prefork-dev libapr1-dev libaprutil1-dev libmysqlclient-dev
```

libmagickcore-dev libmagickwand-dev curl git-core patch build-essential bison zlib1g-dev libssl-dev libxml2-dev libxml2-dev sqlite3 libsqlite3-dev autotools-dev libxslt1-dev libyaml-0-2 autoconf automake libreadline6-dev libyaml-dev libtool imagemagick apache2-utils

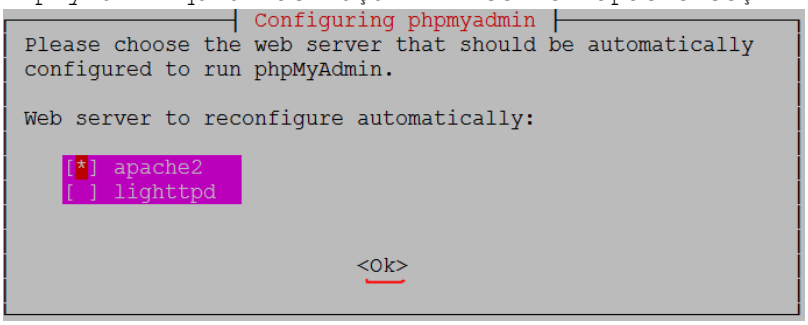
Yüklənmə zamanı bizdən MySQL üçün root şifrəsinin təyin edilməsi istəniləcək(Şəkildə göstərildiyi kimi):



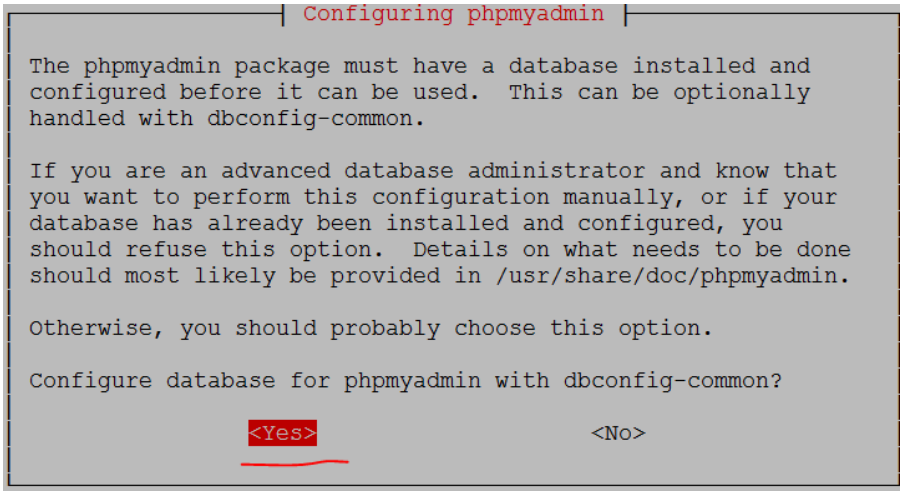
şifrəni təkrar daxil edirik:



PhpMyAdmin qurulması üçün WEB server apache seçirik:



PhpMyAdmin-in bazasını dbconfig-common ilə quraşdırırıq:



root istifadəçi üçün şifrəni daxil edirik ki, phpmyadmin adlı baza yaradıb lazımı cədvəl və sxemləri qurulsun.

Subversion yüklənməsi və quraşdırılması

```
sysuser@redmine:~$ sudo apt-get install subversion libapache2-svn
```

SVN üçün qovluq yaradırıq, həmin qovluq üçün web serverimizə yetki veririk və dav_svn modulunu aktivləşdiririk:

```
sysuser@redmine:~$ sudo mkdir -p /var/lib/svn
sysuser@redmine:~$ sudo chown -R www-data:www-data /var/lib/svn
sysuser@redmine:~$ sudo a2enmod dav_svn
```

Faylı açırıq:

```
sysuser@redmine:~$ sudo nano /etc/apache2/mods-enabled/dav_svn.conf
```

Və aşağıdakı sətirlərin qarşısından şərhi silirik:

```
<Location /svn>
    DAV svn
    SVNParentPath /var/lib/svn
    AuthType Basic
    AuthName "My repository"
    AuthUserFile /etc/apache2/dav_svn.passwd
    AuthzSVNAccessFile /etc/apache2/dav_svn.authz
    <LimitExcept GET PROFIND OPTIONS REPORT>
    Require valid-user
    </LimitExcept>
</Location>
```

SVN qeydiyyat modulunu aktivləşdiririk:

```
sysuser@redmine:~$ sudo a2enmod authz_svn
```

redmine istifadəçisini əlavə edirik ki, bu repository-dən oxuya bilsin:
sysuser@redmine:~$ sudo htpasswd -c /etc/apache2/dav_svn.passwd redmine
New password: **şifrə**
Re-type new password: **şifrə_tekrar**
Adding password for user **redmine**

Apache servisini yenidən işə salırıq:
sysuser@redmine:~$ sudo service apache2 restart
 * Restarting web server apache2   [OK]

Repository yaradırıq:
sysuser@redmine:~$ **sudo svnadmin create --fs-type fsfs /var/lib/svn/my_repository**
sysuser@redmine:~$ **sudo chown -R www-data:www-data /var/lib/svn**

Repository yetkisinin quraşdırılması üçün faylı açın:
sysuser@redmine:~$ **sudo nano /etc/apache2/dav_svn.authz**

redmine-in repository-ə yetki alması üçün quraşdırma faylında əlavə edirik(faylı yadda saxlayaraq çıxırıq):
[my_repository:/]
redmine = r

Ruby və Ruby on Rails-i yükləyirik
sysuser@redmine:~$ **sudo apt-get install ruby1.9.3 ruby1.9.1-dev ri1.9.1 libruby1.9.1 libssl-dev zlib1g-dev**

sysuser@redmine:~$ **sudo update-alternatives --install /usr/bin/ruby ruby /usr/bin/ruby1.9.1 400 **
**> --slave   /usr/share/man/man1/ruby.1.gz ruby.1.gz **
**> /usr/share/man/man1/ruby1.9.1.1.gz **
**> --slave   /usr/bin/ri ri /usr/bin/ri1.9.1 **
**> --slave   /usr/bin/irb irb /usr/bin/irb1.9.1 **
> --slave   /usr/bin/rdoc rdoc /usr/bin/rdoc1.9.1

Redmine-in yüklənməsi

Hal-hazırda yüklədiyimiz versiya 3.0.1-dir amma siz öz istədiyiniz versiyaya dəyişə bilərsiniz.

sysuser@redmine:~$ **cd /usr/share**
sysuser@redmine:/usr/share$ **sudo wget http://www.redmine.org/releases/redmine-3.0.1.tar.gz**

sysuser@redmine:/usr/share$ **sudo tar xvfz redmine-3.0.1.tar.gz**
sysuser@redmine:/usr/share$ **sudo rm redmine-3.0.1.tar.gz**
sysuser@redmine:/usr/share$ **sudo mv redmine-3.0.1/ redmine**
sysuser@redmine:/usr/share$ **sudo chown -R root:root /usr/share/redmine**
sysuser@redmine:/usr/share$ **sudo chown www-data /usr/share/redmine/config/environment.rb**

```
sysuser@redmine:/usr/share$ sudo ln -s /usr/share/redmine/public
/var/www/html/redmine
```

MySQL

RedMine qoşulub məlumatlarını yaza bilməsi üçün MySQL verilənlər bazası, istifadəçi adı və şifrə yaradırıq.

MySQL console-a daxil oluruq:

```
sysuser@redmine:/usr/share$ mysql -uroot -p'mysqlpass'
```

MySQL console-unda aşağıdakı əmrləri yerine yetiririk:

```
mysql> CREATE DATABASE redmine character SET utf8;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE user 'redmine'@'localhost' IDENTIFIED BY 'redminedbpass';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL privileges ON redmine.* TO 'redmine'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
```

Redmine-in bazaya qoşulmasını konfiqurasiya edək:

```
sysuser@redmine:/usr/share$ sudo cp redmine/config/database.yml.example
redmine/config/database.yml
```

Verilənlər bazası quraşdırma faylını açırıq:

```
sysuser@redmine:/usr/share$ sudo nano redmine/config/database.yml
```

İstifadəçi adı, şifrə və verilənlər bazasının şifrəsini yaratdığımıza uyğun olaraq aşağıdakı şəkildəki kimi dəyişirik:

```
production:
  adapter: mysql2
  database: redmine
  host: localhost
  username: redmine
  password: "redminedbpass"
  encoding: utf8
```

Qururuq:

```
sysuser@redmine:/usr/share$ cd /usr/share/redmine/
sysuser@redmine:/usr/share/redmine$ sudo gem install bundler
sysuser@redmine:/usr/share/redmine$ sudo bundle install --without development
test postgresql sqlite
```

```
sysuser@redmine:/usr/share/redmine$ sudo rake generate_secret_token
sysuser@redmine:/usr/share/redmine$ sudo RAILS_ENV=production rake db:migrate
sysuser@redmine:/usr/share/redmine$ sudo RAILS_ENV=production rake
redmine:load_default_data
Select language: ar, az, bg, bs, ca, cs, da, de, el, en, en-GB, es, et, eu,
fa, fi, fr, gl, he, hr, hu, id, it, ja, ko, lt, lv, mk, mn, nl, no, pl, pt,
pt-BR, ro, ru, sk, sl, sq, sr, sr-YU, sv, th, tr, uk, vi, zh, zh-TW [en]ENTER

sysuser@redmine:/usr/share/redmine$ sudo mkdir public/plugin_assets
sysuser@redmine:/usr/share/redmine$ sudo chown -R www-data:www-data files log
tmp public/plugin_assets
sysuser@redmine:/usr/share/redmine$ sudo chmod -R 755 files log tmp
public/plugin_assets
```

Phusion Passenger yüklənilməsi

Phusion Passenger Ruby-nin dəstəklədiyi WEB serverdir. Dizayn edilmişdir ki, apache və nginx web serverlə birlikdə işləyə bilsin.

Phusion Passenger üçün Repository əlavə edirik:

```
sysuser@redmine:/usr/share/redmine$ sudo apt-key adv --keyserver
keyserver.ubuntu.com --recv-keys 561F9B9CAC40B2F7
sysuser@redmine:/usr/share/redmine$ sudo apt-get install apt-transport-https
ca-certificates
```

Yeni repository qquraşdırma faylını açırıq:

```
sysuser@redmine:/usr/share/redmine$ sudo nano
/etc/apt/sources.list.d/passenger.list
```

Aşağıdakı sətiri fayla əlavə edib yadda saxlayaraq çıxırıq:

```
deb https://oss-binaries.phusionpassenger.com/apt/passenger trusty main
```

Fayla uyğun olan yetkiləri təyin edirik:

```
sysuser@redmine:/usr/share/redmine$ sudo chown root:
/etc/apt/sources.list.d/passenger.list
sysuser@redmine:/usr/share/redmine$ sudo chmod 600
/etc/apt/sources.list.d/passenger.list
```

Yükləyirik

```
sysuser@redmine:/usr/share/redmine$ sudo apt-get update
sysuser@redmine:/usr/share/redmine$ sudo apt-get install libapache2-mod-
passenger
```

Qurulması:

passenger konfiqruasiya faylını açırıq:

```
sysuser@redmine:/usr/share/redmine$ sudo nano /etc/apache2/mods-
available/passenger.conf
```

PassengerDefaultUser www-data sətirini passenger quraşdırma faylına aşağıdakı şəkildə əlavə edirik:

```
<IfModule mod_passenger.c>
  PassengerRoot /usr/lib/ruby/vendor_ruby/phusion_passenger/locations.ini
  PassengerDefaultRuby /usr/bin/passenger_free_ruby
  PassengerDefaultUser www-data
</IfModule>
```

apache2 quraşdırma faylını açırıq:

```
sysuser@redmine:~$ sudo nano /etc/apache2/sites-available/000-default.conf
```

faylı aşağıdakı şəklə gətiririk (Faylda edilən dəyişikliklər yaşıl, əlavələr isə qırmızı rəngdə qeyd edilmişdir):

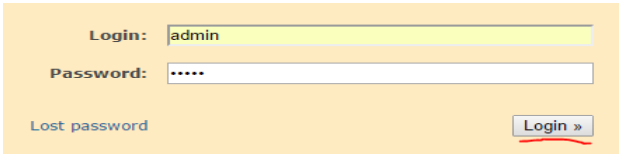
```
<VirtualHost *:80>
        ServerAdmin server.admin@email.com
        DocumentRoot /var/www/html/redmine
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
<Directory /var/www/html/redmine>
    RailsBaseURI /redmine
    PassengerResolveSymlinksInDocumentRoot on
</Directory>
```

Modulu aktivləşdiririk və apache servisi yenidən işə salırıq ki, dəyişikliklər işə düşə bilsin:

```
sysuser@redmine:~$ sudo a2enmod passenger
Module passenger already enabled
sysuser@redmine:~$ sudo service apache2 restart
 * Restarting web server apache2        [ OK ]
```

Redmine-i işə salırıq

Artıq redmine-in web səhifəsinə http://server_IP/ yazmaqla daxil ola bilərsiniz.



Login: **admin**
Pass: **admin**

eMail quraşdıraq

SMTP və şifrələnmə üçün fayl yaradırıq

Quraşdırma faylını açırıq:

```
sysuser@redmine:~$ sudo nano /usr/share/redmine/config/configuration.yml
```

Aşağıdakı sətirləri yaratdığımız redmine email quraşdırma faylına əlavə edirik:

```
production:
  email_delivery:
    delivery_method: :smtp
    smtp_settings:
      enable_starttls_auto: true
      address: "smtp.gmail.com"
      port: '587'
      domain: "smtp.gmail.com"
      authentication: :plain
      user_name: "redmine@gmail.com"
      password: "remineemailpass"
```

Siz email-in işlənməsini WEB interfeysdə yoxlaya bilərsiniz. Haqqında ətraflı aşağıda danışacayıq.

Subversion repository-sinə baxışın avtomatik yenilənməsi

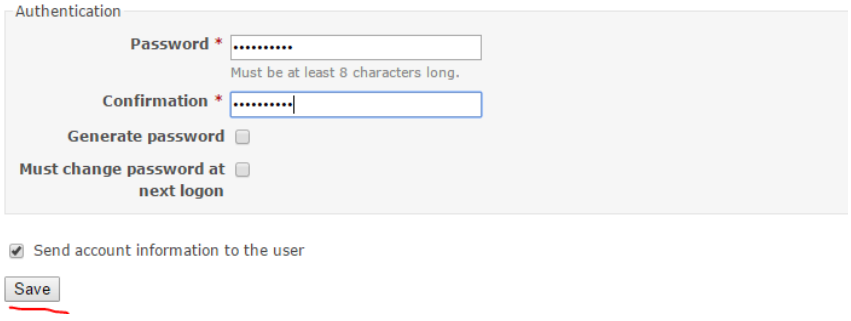
Web interfeys üzərindən proyektin arxiv quraşdırmalarında aktivləşdirilməsinə və api açarın generasiya edilməsinə gərək var.

Göstərilən crontab redmine-i hal-hazırkı subversion-a hər 15 dəqiqədən bir yenileyir. Aşağıdakı əmrlə istifadəçinin cron faylına daxil oluruq:
sysuser@redmine:~$ **sudo crontab -e**

cron sətirini fayla əlavə edirik:
***/15 * * * * curl "http://server_IP/sys/fetch_changesets?key=APIKEY" > /dev/null**

Redmine WEB interfeysin ilkin quraşdırmaları

Web səhifəmizə admin istifadəçi adı və admin şifrəsi ilə daxil olduqdan sonra, ilk işimiz şifrənin dəyişdirilməsidir. Bunun üçün **Administration** -> **Users** -> **admin** seçirik və aşağıdakı şəkildəki kimi şifrəni iki dəfə təkrar daxil etdikdən sonra, **Save** düyməsinə sıxırıq (Həmçinin admin istifadəçisi üçün vaxt enliyi, email və dil kimi imkanları da seçə bilərsiniz):



Email-in göndərilməsini sınaqdan keçirmək üçün bəzi səliqə işləri görmək lazımdır. Bunun üçün WEB səhifədə **Administration** -> **Settings** -> **Geteral** Tab altında öz WEB ünvanınızı daxil edib **Save** düyməsinə sıxmalısınız.

Sonra WEB interfeysdə **Administration** -> **Settings** -> **Email notifications** unvanına daxil oluruq və **Send a test email** düyməsinə sıxmaqla hansı istifadəçi adı ilə sistemə daxil olmuşduqsa o istifadəçinin quraşdırmalarında olan email ünvanına aşağıdakı mətn-lə məktub yollanacaq:

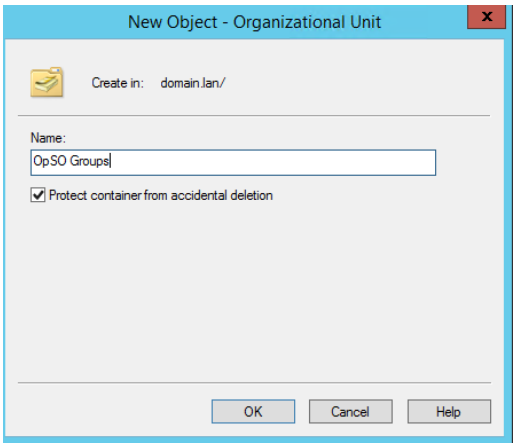
```
This is a test email sent by Redmine.
Redmine URL: http://redmine.opensource.az/
```

Redmine Active Directory Integration

Deyək ki, sizin şirkətinizin daxilində artiq mövcud DC quraşdırılmışdır və şirkətin tələbi ondan ibarətdir ki, istənilən portala giriş eyni istifadəçi hesabları mənbəsindən götürülməlidir(Single Sign On). Bu halda siz RedMine-i Active Directory ilə inteqrasiya etməlisiniz. Həmçinin tələb ondan ibarətdir ki, Redmine-a yalnız seçilmiş DC qrupda olan istifadəçilər daxil ola bilərlər. Gəlin işimizə başlayaq. Sınaqlarımızda Windows 2012 server R2 Standart x64 istifadə edilmişdir.

DC FQDN: **domain.lan**

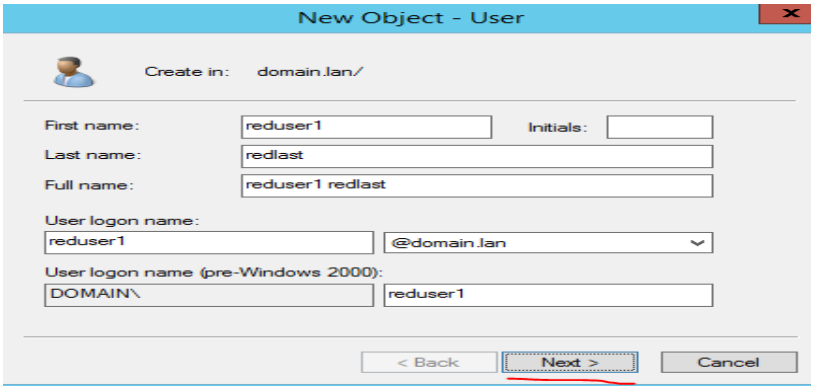
Öncə bir OU yaradırıq ki, müəssisəmizə aid olan qruplar həmin qrupda cəmlənsin. Sonra həmin OU-nin içində bir qrup yaradaq ki, yalnız bu qrup üzvlüyündə olan istifadəçilər redmine-a daxil ola bilsinlər. Windows serverdə **Server Manager** -> **Active Directory Users and Computers** -> DC FQDN üstündə sağ düyməni sıxırıq (yəni **domain.lan**) -> **New** -> **Organizational Unit** və aşağıdakı şəkildəki kimi, **OpSO Groups** adlı **OU** yaradırıq.



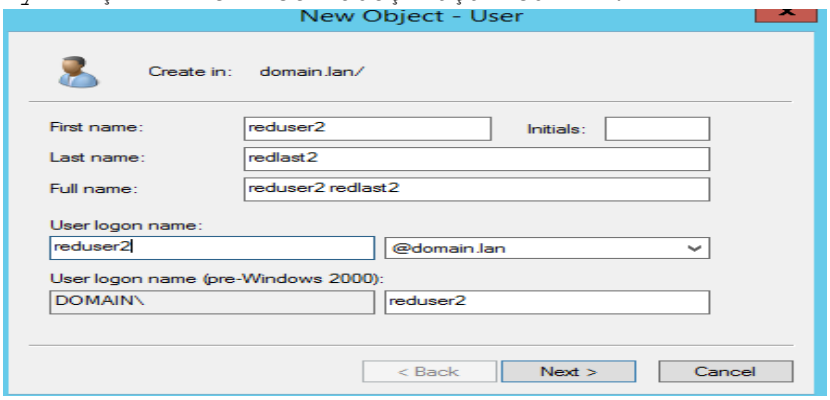
Sonra bu **OpSO Groups** OU üstündə sağ düyməni sıxırıq **New** -> **Group** və şəkildəki kimi, qrupun adını **RemineUsers** yazıb, **OK** düyməsini sıxırıq.

Sonra sınaqlarımızı keçirə bilməmiz üçün iki ədəd istifadəçi yaradırıq və bir istifadəçini həmin qrupun üzvü edirik, digərini isə yox.

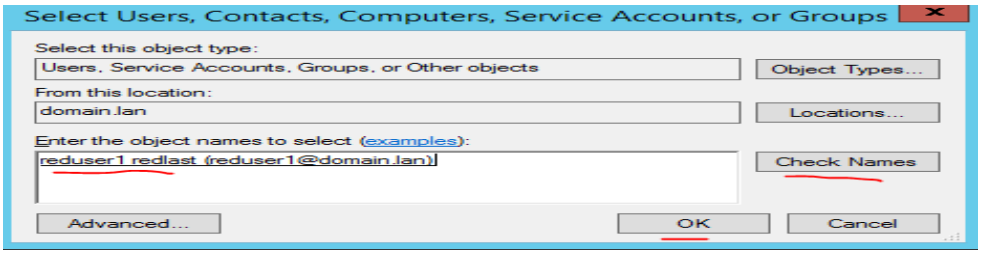
Server Manager -> **Active Directory Users and Computers** -> DC FQDN üstündə sağ düyməni sıxırıq(yəni **domain.lan**) -> **New** -> **User** və şəkildəki kimi istifadəçiyə müəyyən ad və şifrə təyin edib **Next** düyməsini sıxırıq. Şifrəni daxil edirik və şifrənin vaxtının heç bir zaman bitməməsini seçiv **ok** düyməsini sıxırıq.



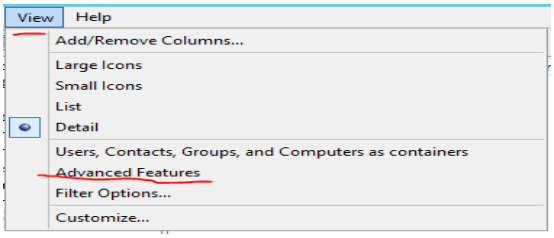
Eyni işi ikinci istifadəçi üçün edirik:



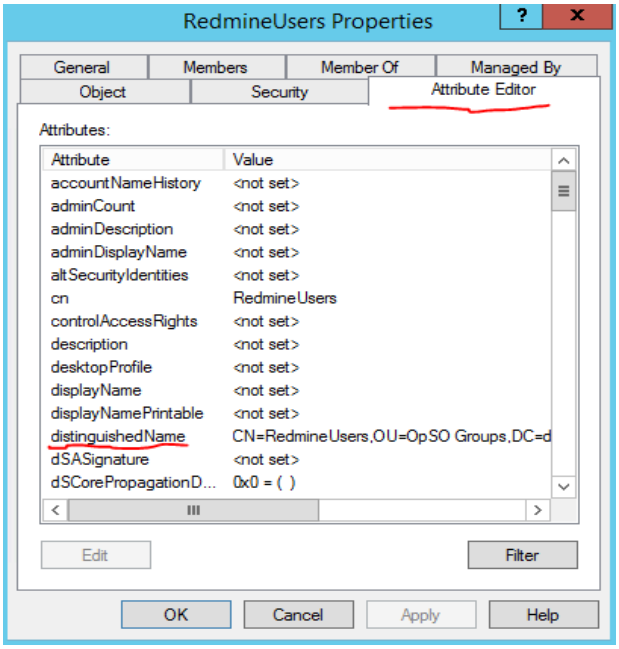
Sonra yaratdığımız **RedmineUser** qrupun üstündə sağ düyməni sıxırıq və **Properties** -> **Members** bölümünə daxil oluruq -> **Add** düyməsini sıxırıq və şəkildə göründüyü kimi, **reduser1** daxil edib, **Check Names** düyməsi ilə axtardıqdan sonra **Ok -> OK** düyməsini sıxırıq.



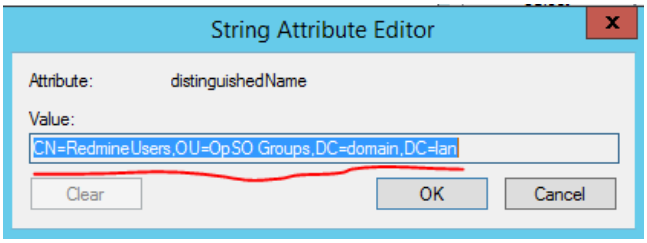
Redmine-in LDAP-la inteqrasiyasında qrup süzgəci üçün bizə qrupun Distinguised Name-i tələb olunacaq. Bunun üçün **Server Manager** -> **Active Directory Users and Computers** -> **View** -> **Advanced Features** bölümünə daxil olmaq lazımdır (şəkildəki kimi).



Sonra yaratdığımız **RedmineUser** qrupun üstündə sağ düyməni sıxıb **Properties** -> **Attribute Editor** bölümünə daxil olub, **distinguished Name** sətirini tapmaq lazımdır (Şəkildəki kimi).



distinguished Name sətirin üstündə iki defe sıxırıq və şəkildəki məzmuna uyğun məlumatı nüsxələyirik:

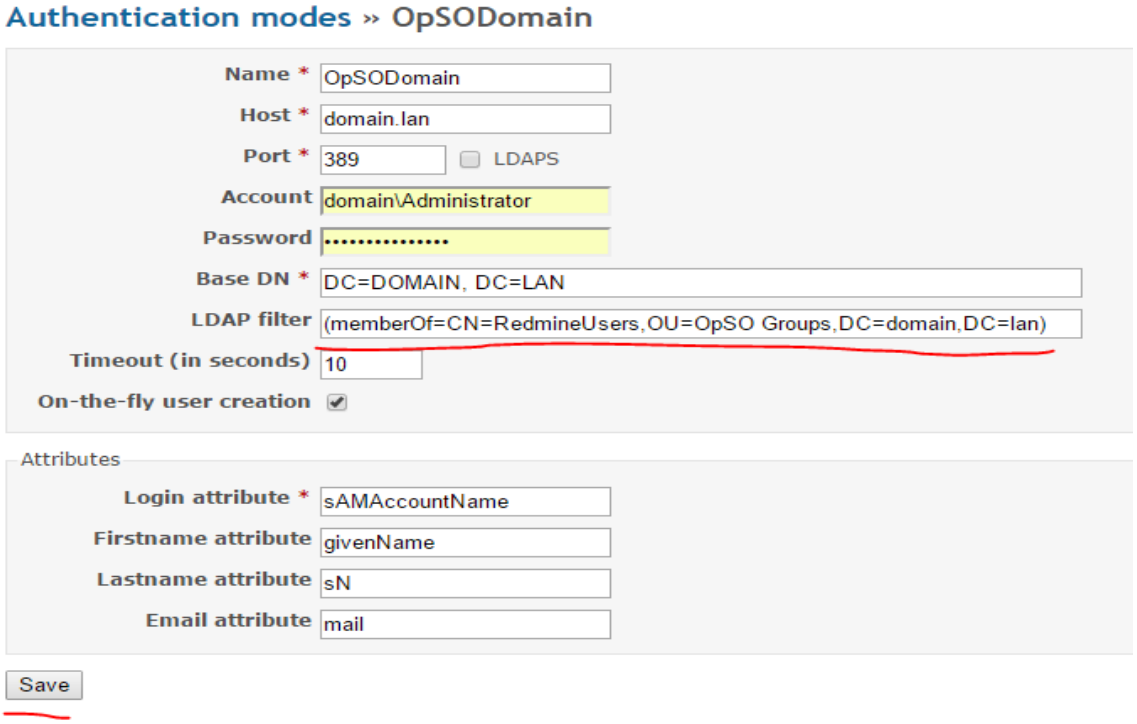


Nüsxələdiyimiz məlumat aşağıdakından ibarətdir(Bu məlumat bizə qrupun süzgəcində tələb olunacaq):

CN=RedmineUsers,OU=OpSO Groups,DC=domain,DC=lan

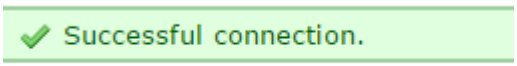
Qeyd: Unutmayın Redmine DC-ni resolve etməsi üçün DC DNS-ni öz **/etc/resolv.conf** faylında yazmalıdır. Öz sınaqlarımda DC IP **10.50.3.158** idi və **resolv.conf** faylında **nameserver 10.50.3.158** idi.

Artıq gedirik redmine web səhifəsinin qurulmasına. http://server_IP/ ünvanına daxil oluruq. **Administration -> LDAP authentication -> New authentication mode** düyməsini sıxırıq və açılan pəncərədə xanaları şəkildəkinə uyğun olaraq doldurub, **Create** düyməsini sıxırıq (**LDAP Filter** xanasına fikir versəniz görəcəksiniz ki, bayaq nüsxələdiyimiz DN-i yazmışıq).



Uğurla yaradıldığı halda aşağıdakı şəkil çap edilir. Sınaq üçün **Test** düyməsini sıxıb yoxlaya bilərsiniz.

Uğurlu sınaq aşağıdakı cavabı verməlidir:



RedMine serverdə LDAP alətlərindən istifadə müəyyən sınaqları edə bilərsiniz. Bu paket vasitəsilə serverimizin LDAP-a uğurlu qoşulmasını və qrupun axtarışını sınaqdan keçirə bilərik.

```
root@redmine:~# apt-get install ldap-utils
```

Əgər DC-də **redmineusers** kriteriyasına əsaslanaraq axtarış etmək istəsək aşağıdakı əmrdən istifadə edirik:

```
root@redmine:~# ldapsearch -x -b  "dc=domain,dc=lan" -H ldap://domain.lan/ -D "DOMAIN\Administrator" -w A123456789a redmineusers
```

Nəticədə aşağıdakı sətirlər sizin ekrana çap edilməlidir:

```
# RedmineUsers, OpSO Groups, domain.lan
dn: CN=RedmineUsers,OU=OpSO Groups,DC=domain,DC=lan
```

Artıq yalnız DC-də təyin etdiyimiz qrupda olan istifadəçilər redmine-a daxil olub istifadə edə biləcəklər.

Ubuntu 14.04 x64 xWiki yuklənməsi və quraşdırılması

xWiki - Javada yazılmış açıq qaynaqlı genişlənə bilən dizayna sahib bir wiki proqram platformasıdır. Wiki proqramı olaraq, strukturlaşmış datanın saxlanılması və server tərəfdə olan scriptlərin wiki interfeysində işə salınması imkana sahibdir. Script dilləri wiki macros-ları istifadə edilərək, Velocity, Groovy, Python, Ruby və PHP daxil olmaqla birbaşa wiki səhifələrinin içində yazıla bilər.

Aşağıdakı imkanlara sahıbdir:

- Wiki proqramlarının ququrlmasın imkan yaradan strukturlaşmış mətn və

daxili script yazma.

- İstifadəçi hüquqlarının idarə edilməsi
- PDF export
- Tam-mətn axtarışı
- Versiya kontrolu
- Ofis sənədlərinin OpenOffice üzərindən wiki sintaksisinə İmport edilməsi
- Wiki-yə yetki almaq üçün çeşidli protokollar(WebDAV, REST, XmlRpc, GWT)
- Tərkib, sayt dizaynı, Export və Import
- Pluginlər, APİ
- Bütün imkanları rəsmi saytından http://www.xwiki.org/ əldə edilə bilər

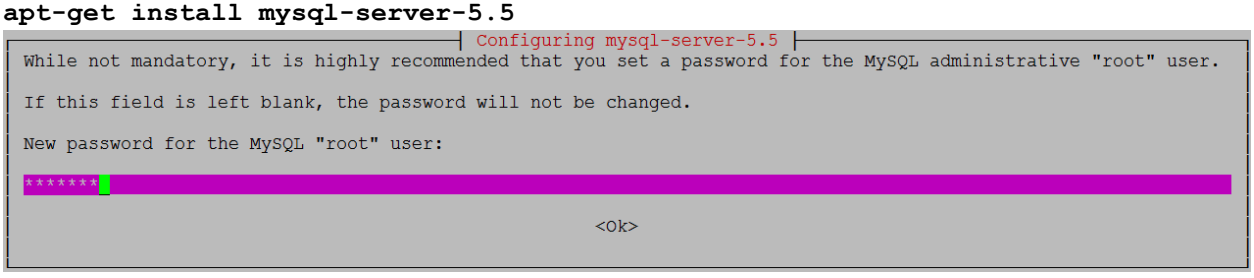
Qurulmasına başlayaq

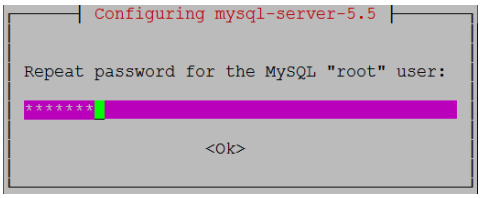
Öncədən qeyd edim ki, siz **Ubuntu-Tomcat7-http-https.docx** sənədi ilə tomcat7-ni yükləyib quraşdırmalısınız və yalnız bundan sonra xWiki yüklənməsinə baxmalısınız. Çünki xWiki **Tomcat7**, **MySQL** və **JDBC-MySQL-Connector** ilə işləyir. Həmçinin xWiki tomcat-in susmaya görə quraşdırmasında olduğu RAM həcmindən çox həcm istifadə etdiyine görə aşağıdakı quraşdırmanı mütləq etməlisiniz:

```
vi /etc/default/tomcat7 # Faylda JAVA_OPTS dəyişəninə qeyd qoyub aşağıdakı
                        dəyişəni əlavə edirik
JAVA_OPTS="-Xmx1024m -Xms1024m"

/etc/init.d/tomcat7 restart          # Servisi yenidən başladırıq ki,
                                     dəyişiklik işə düşsün
```

Yuxarıda göstərilən sənədi tam olaraq oxuyub lazım olanları quraşdırdıqdan sonra isə, MySQL-i serverimizə yükləyirik (root şifrəmizi iki dəfə daxil edirik):

```
apt-get install mysql-server-5.5
```





Sonra **mysql-connector-java-5.1.31-bin.jar** və **xwiki-enterprise-web-6.1.war** fayllarını serverimizə yükləyirik. **xwiki-enterprise-web-6.1.war** faylını **/var/lib/tomcat7/webapps** qovluğuna **xwiki.war** adı ilə köçürürük.

```
cp /home/jamal/xwiki-enterprise-web-6.1.war
/var/lib/tomcat7/webapps/xwiki.war
```

Ardınca isə **mysql-connector-java-5.1.31-bin.jar** faylını **/var/lib/tomcat7/webapps/xwiki/WEB-INF/lib** ünvanına köçürürük.

```
cp /home/jamal/mysql-connector-java-5.1.31-bin.jar
/var/lib/tomcat7/webapps/xwiki/WEB-INF/lib
```

CLI-dan xWiki üçün MySQL baza, login və şifrə yaradırıq:

```
mysql -u root -pfreebsd -e "create database xwiki default character set utf8
collate utf8_bin"
mysql -u root -pfreebsd -e "grant all privileges on xwiki.* to
xwiki@localhost identified by 'freebsd'"
```

Əmin olun ki, **/etc/hosts** faylında **127.0.0.1 localhost** sətiri mövcuddur.

Sonra **/var/lib/tomcat7/webapps/xwiki/WEB-INF/hibernate.cfg.xml** faylında yaratdığımız MySQL istifadəçi, şifrəsini və bazasını quraşdırırıq. HSQLDB-ni şərh edirik. MySQL üçün isə şərhi silib lazımı baza, istifadəçi adı və şifrəni daxil edirik. Tomcat üçün şərh **<!--** ilə başlayır **-->** ilə bitir.

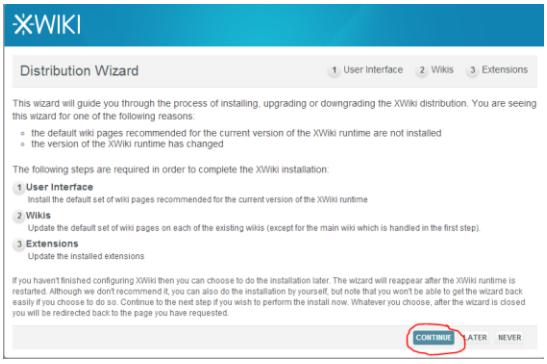
```
    <property name="connection.url">jdbc:mysql://localhost/xwiki</property>
    <property name="connection.username">xwiki</property>
    <property name="connection.password">freebsd</property>
    <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
    <property
name="dialect">org.hibernate.dialect.MySQL5InnoDBDialect</property>
    <property name="dbcp.ps.maxActive">20</property>
    <mapping resource="xwiki.hbm.xml"/>
    <mapping resource="feeds.hbm.xml"/>
    <mapping resource="activitystream.hbm.xml"/>
    <mapping resource="instance.hbm.xml"/>
```

Admin istifadəçi və şifrəni təyin etmək üçün isə **/var/lib/tomcat7/webapps/xwiki/WEB-INF/xwiki.cfg** faylında aşağıdakı sətirdə olduğu kimi şərhi silib, **superadmin** istifadəçisinə şifrə yazırıq (Şifrəmiz **freebsd** olacaq):

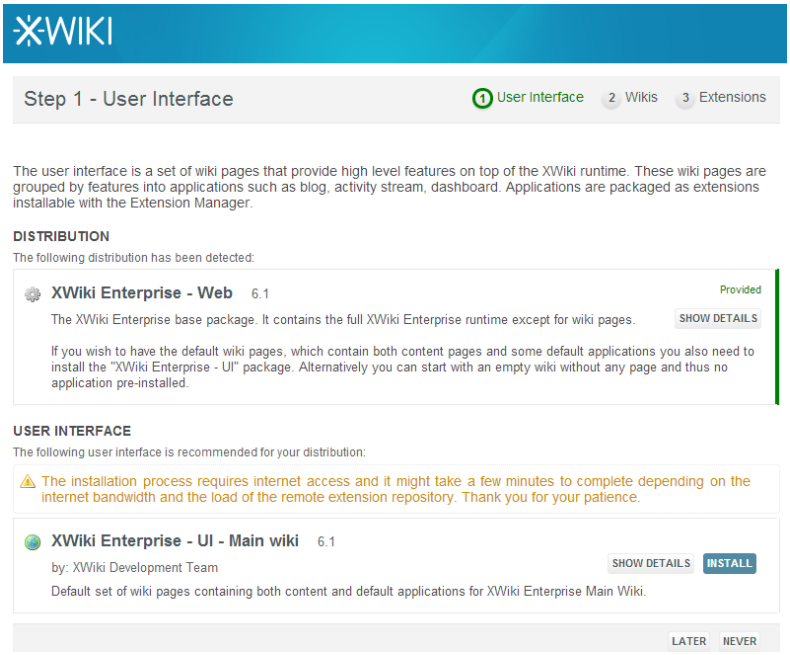
```
xwiki.superadminpassword=freebsd
```

```
/etc/init.d/tomcat7 restart          # Sonda tomcat7-ni restart edirik
```

https://server-ip-address/xwiki/ # xWiki interfeysimizə daxil oluruq. Aşağıdakı şəkil çap ediləcək

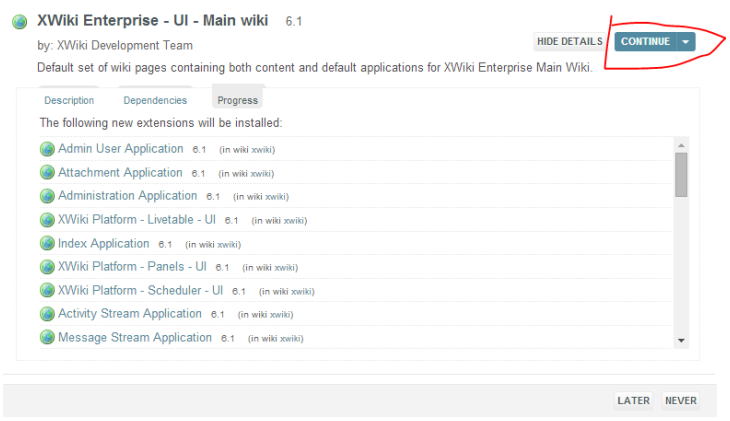


CONTINUE seçirik.

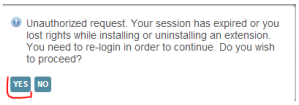


INSTALL seçirik.

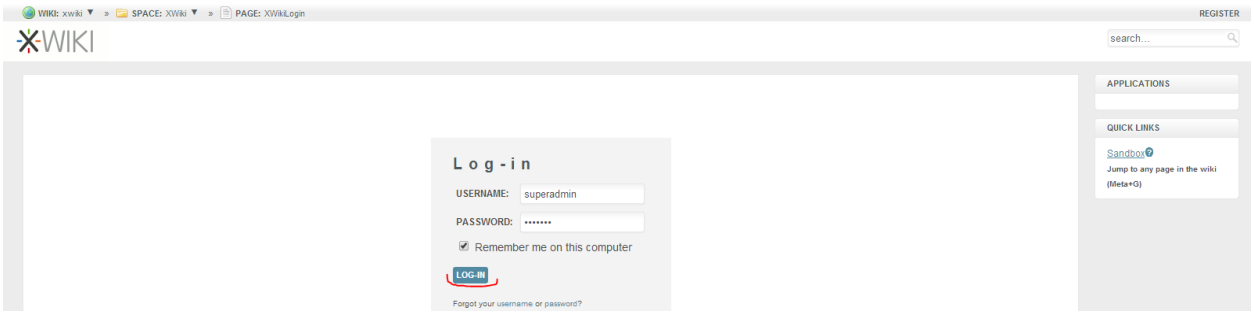
Şəkildə göründüyü kimi yüklənmə bitir və **CONTINUE** seçirik:



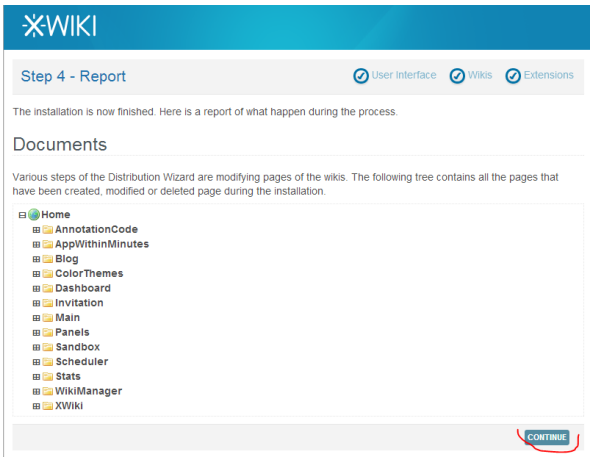
Sonda yüklənmə bitdikden sonra sessiya bizi atacaq və yeniden login olmağı təklif edəcək. Şəkildəki kimi **Yes** sıxırıq:



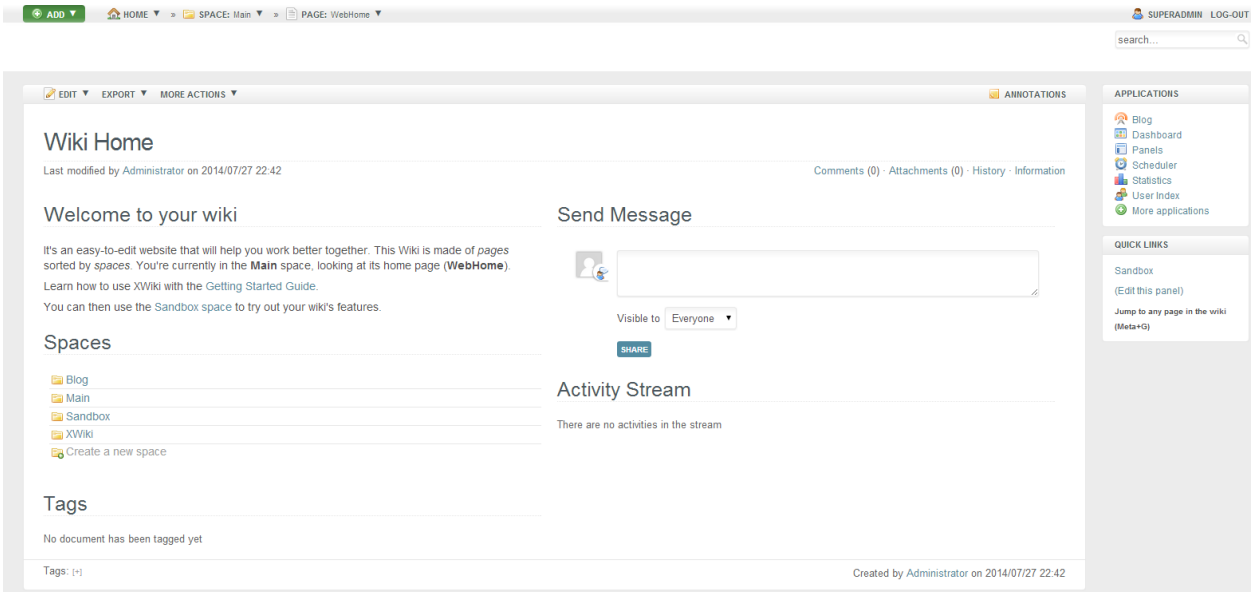
Aşağıdakı kimi səhifə çap olunacaq. **superadmin** istifadəçi adı və şifrəni daxil edirik.



Sonda bir dənə yenidən şəkil çap ediləcək, orda da **CONTINUE** sıxırıq və aşağıdakı şəkil çap edilir:



Yenidən **CONTINUE** sıxırıq və yüklənmə bitir.
Uğurlu nəticədə aşağıdakı şəkil çap edilməlidir:



xWiki Domain Controller ilə inteqrasiya edilməsi

xWiki serverimizi DC ilə inteqrasiya eləmək üçün biz aşağıdakı quraşdırmaları etməliyik.

DC haqqında öncədən lazımı məlumatları verək:
DC Name: **DOMAİN.LAN**
xWiki GROUP Name: **xWikiMembers**
DC Auth User: **Administrator**
DC Auth Pass: **A123456789a**

DC-mizdə lazımı istifadəçiləri xWikiMembers qrupuna əlavə edirik ki, daxil ola bilsinlər.

/var/lib/tomcat7/webapps/xwiki/WEB-INF/xwiki.cfg # Faylda aşağıdakı sətirləri uyğun olaraq quraşdırırıq

```
xwiki.authentication.authclass=com.xpn.xwiki.user.impl.LDAP.XWikiLDAPAuthServiceImpl
xwiki.authentication.ldap=1
xwiki.authentication.ldap.server=domain.lan
xwiki.authentication.ldap.port=389
xwiki.authentication.ldap.bind_DN=domain\\{0}
xwiki.authentication.ldap.bind_pass={1}
xwiki.authentication.ldap.base_DN=DC=domain,DC=lan
xwiki.authentication.ldap.user_group=CN=xWikiMembers,
OU=OpSO Groups,DC=domain,DC=lan
xwiki.authentication.ldap.UID_attr=sAMAccountName
xwiki.authentication.ldap.fields_mapping=name=sAMAccountName,last_name=sn,first_name=givenName,fullname=displayName,email=mail,ldap_dn=dn
xwiki.authentication.ldap.update_user=1
xwiki.authentication.ldap.trylocal=0
```

/var/lib/tomcat7/webapps/xwiki/WEB-INF/xwiki.properties # Faylda aşağıdakı sətirləri uyğun olaraq quraşdırırıq(Qovluqlar yoxdursa yaradırıq və tomcat7 user,qrup üzvü edirik)

```
environment.permanentDirectory=/var/cache/tomcat7/Catalina/localhost/xwiki/
solr.embedded.home=/var/cache/tomcat7/Catalina/localhost/xwiki/solr
```

/etc/init.d/tomcat7 restart # Sonda tomcat7-ni restart edirik

BÖLÜM 2

Bulud sistemləri

- **FreeBSD 10.2 x64 server üzərində OwnCloud yüklənməsi və qurulması**
- **OwnCloud-un Domain Controller ilə integrasiya edilməsi**
- **FreeBSD 10.1 x64 Pydio Cloud qurulması**

Şirkətin daxili tələbləri genişləndikcə, informasiya önəmliliyi və təhlükəsizliyi tələbləri böyüməyə başlayır. Eynilə istifadəçilərin arasıda informasiya paylaşımı komfortu tələbi də yaranır. Misal üçün paylaşım Domain Controllerdə olan istifadəçi və qruplar arasında seçimə görə, xüsusi keşlə generasiya edilmiş URL-ə(Bu URL-lə şifrə təyin edilməsi imkanı var) görə, paylaşılmış ünvana paylaşım vaxtının bitməsi tarixinin təyin edilməsinə görə və vaxtın bitməsi zamanı məktubla xəbərdarlığın edilməsinə görə bacarıqlara sahibdir. Bu tip tələbləri qarşılayan tanıdığımız DropBox və GoogleDrive mövzuddur. Başlığımızım mövzuları eyni tələbləri qarşılayan açıq qaynaqlı proqram təminatları haqqındadır.

FreeBSD 10.2 x64 server üzərində OwnCloud yüklənməsi və qurulması

ownCloud — məlumatların sinxronlaşdırması, faylların paylaşılması və sənədlərin uzaq serverdə saxlanılması üçün açıq qaynaqlı web proqram təminatıdır.

ownCloud PHP və JavaScript proqramlaşdırma dillərində yazılmışdır. OwnCloud serveri SQLite, MariaDB, MySQL, Oracle və PostgreSQL məlumat bazalarıyla inteqrasiya edilib işlədilə bilər.

KDE yaradıcılarından biri, Karliçek Frank məlumatların saxlanılması üçün ticari xidmətlərinə pulsuz alternativ kimi 2010-cu ilin yanvarında ownCloud-un hazırlanmasına başladı. Ticari fayl mübadiləsi xidmətlərindən fərqli olaraq, ownCloud-u əlavə xərclər tələb etmədən, şəxsi serverə yükləmək olar.

Məlumatların sinxronlaşdırmasında Windows, Mac OS, Linux və həmçinin iOS, Android mobil əməliyyat sistemləri üçün müştəri proqramlarına sahibdir. Eynilə saxlanılmış məlumatlar OwnCloud web-interfeysinin köməyi ilə istifadə edilə bilər.

ownCloud artıq Debian GNU Linux anbarına əlavə edilmiş və Gnome iş stoluna inteqrasiya edilmişdir.

İmkanları:

- Faylların adi qovluqlar strukturunda ya da WebDAV istifadə edilərək saxlanılması.
- Şifrələnmə
- İstənilən Windows(Windows XP, Vista, 7 və 8), Mac OS X (10.6 və ya daha yeni) ya da Linux desktoplar arasında sinxronizasiya
- Təqvim(Həmçinin CalDAV)
- Məsələlərin planlaşdırıcısı
- Ünvan kitabçası(Həmçinin CardDAV)
- Axınlı multimedia(Ampache istifadə edilir)
- İstifadəçi və qrupların idarə edilməsi(OpenID ya da LDAP istifadə edərək)
- Kontentin qruplar, istifadəçilər ya da dünya URL vasitəsilə paylaşdırılması
- Sintaksis göstəricisi və qatlanmayla onlayn mətn redaktoru
- Əlfəcinlər
- URL-in qısaldılması mexanizmi
- Şəkil qalereyası
- PDF sənədlərə baxış(PDF.js istifadə edilir)
- ODF faylllara baxış(.odt, .odp, .ods)
- Jurnallanma modulu

İndi isə biz FreeBSD OS-da bu proqram təminatını yükləyib quraşdıracayıq. Clientlər isə şifrələnmiş kanal üzərindən öz məlumatlarını serverə yükləyəcəklər.

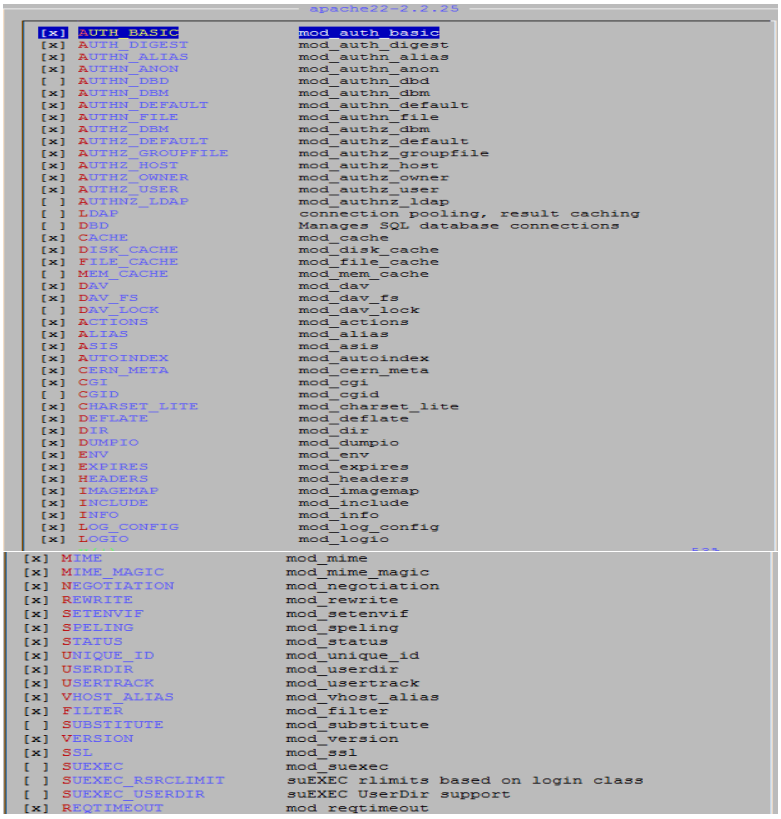
192.168.11.200 - Serverimizin IP ünvanı
owncloud.az - Serverimizin HostName-i

Qeyd: Mütləq **/etc/hosts** faylına nəzərinizdə tutduğunuz adı uyğun IP ilə əlavə edin. Əks halda errorlar çap
ediləcək.

```
cat /etc/hosts          # Hosts faylimiz
127.0.0.1               localhost localhost.my.domain
192.168.11.200          owncloud.az owncloud
```

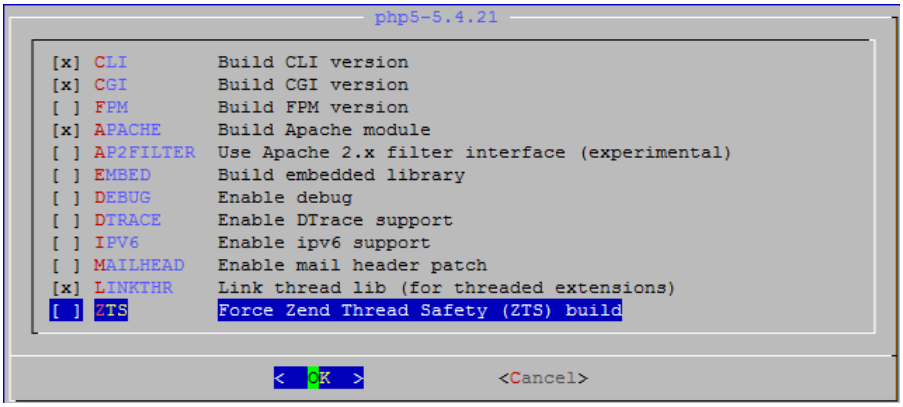
Öncə Web Serveri və PHP-ni yükləyək.

```
cd `whereis apache22 | awk '{ print $2 }'`# Apache-ın portuna daxil oluruq.
make config                               # Lazımı modulları seçirik.
```




```
make install                              # Yükləyirik.
```

```
cd `whereis php5 | awk '{ print $2 }'`    # PHP5 portuna daxil oluruq.
make config                               # Lazımı modulları seçirik.
```



make install # Yükləyirik

Ümumiyyətlə OwnCloud üçün tələb edilən bütün php genişlənmələri tampaq üçün öncə ona aid olan Makefile-ın içini mütləq oxumaq lazımdır. Tünd qara simvollar tələb edilən modullardır.

```
root@owncloud:/usr/local/etc # cat /usr/ports/www/owncloud/Makefile
# $FreeBSD: www/owncloud/Makefile 336609 2013-12-16 05:57:04Z kevlo $

PORTNAME=       owncloud
PORTVERSION=    6.0.0a
CATEGORIES=     www
MASTER_SITES=   http://download.owncloud.org/community/

MAINTAINER=     kevlo@FreeBSD.org
COMMENT=        Personal cloud which runs on your own server

LICENSE=        AGPLv3

BUILD_DEPENDS=  mp3info:${PORTSDIR}/audio/mp3info
RUN_DEPENDS:=   ${BUILD_DEPENDS}

USE_BZIP2=      yes
USE_PHP=        ctype curl dom fileinfo filter gd hash iconv json ldap \
                mbstring openssl pdo session simplexml xml xmlreader \
                xsl wddx zip zlib
WANT_PHP_WEB=   yes

WRKSRC=         ${WRKDIR}/${PORTNAME}
NO_BUILD=       yes
SUB_FILES=      pkg-message

OPTIONS_MULTI=  DB
OPTIONS_MULTI_DB=       MYSQL PGSQL SQLITE
OPTIONS_DEFAULT=        SQLITE
MYSQL_USE=      MYSQL=client PHP=mysql,pdo_mysql
PGSQL_USE=      PGSQL=yes PHP=pdo_pgsql,pgsql
SQLITE_USE=     PHP=pdo_sqlite,sqlite3
```

```
do-install:
        @${MKDIR} -m 0755 ${STAGEDIR}${WWWDIR}
        @cd ${WRKSRC} && ${COPYTREE_SHARE} . ${STAGEDIR}${WWWDIR}

.include <bsd.port.mk>
```

PHP üçün Lazımı genişlənmələri yükləyək.

```
cd `whereis php5-extensions | awk '{ print $2 }'`      # Portuna daxil oluruq.
make config                                            # Lazımı modulları seçirik.
```

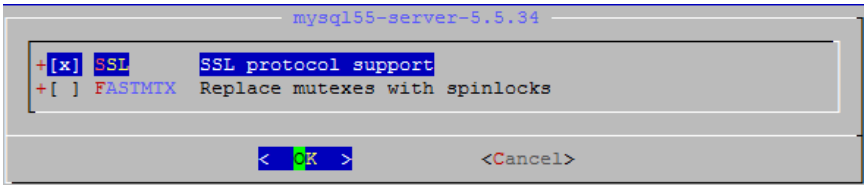

```
make -DBATCH install                                   # Yükləyirik
```

OwnCloud-u yükləyək.

```
cd `whereis owncloud | awk '{ print $2 }'` # OwnCloud-un ünvanına daxil oluruq
make install                               # Yükləyirik.
```

Servislərimizi Startup-a əlavə edək və işə salaq.

```
echo 'apache22_enable="YES"' >> /etc/rc.conf
echo 'apache22ssl_enable="YES"' >> /etc/rc.conf

cd `whereis mysql55-server | awk '{ print $2 }'`       # MySQL-i yükləyək
make config                                            # Lazımı modulları seçək
```



```
make install                                    # Yükləyirik

echo 'mysql_enable="YES"' >> /etc/rc.conf       # Startup-a əlavə edək


/usr/local/etc/rc.d/mysql-server start    # MySQL-i işə salırıq.
/usr/local/etc/rc.d/apache22 start        # Apache22-ni işə salırıq.
```

Aşağıdakı sətirləri WEB Serverimizin configinə əlavə edək ki, həm PHP işləsin həmdə VirtualHost-ları aktiv edək.

```
echo 'AddType application/x-httpd-php .php' >> /usr/local/etc/apache22/httpd.conf
echo 'AddType application/x-httpd-php-source .phps' >> /usr/local/etc/apache22/httpd.conf

echo 'Include /usr/local/domen/*' >> /usr/local/etc/apache22/httpd.conf
```

Həmçinin **'/usr/local/etc/apache22/httpd.conf'** faylından **DirectoryIndex** sətirinin qarşısına **index.php** əlavə edirik ki, PHP scriptlər ilk index edənlərdən olsun.

```
mkdir -p /usr/local/domen/          # VirtualHost-lar üçün qovluq yaradaq.
```

OwnCloud üçün VirtualHost faylı yaradaq və aşağıdakıları içinə əlavə edək

```
cat /usr/local/domen/owncloud.az
<VirtualHost *:80>
        RewriteEngine on
        ReWriteCond %{SERVER_PORT} !^443$
        RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R,L]
</VirtualHost>
<VirtualHost *:443>
        SSLEngine on
        SSLCertificateFile /usr/local/etc/apache22/ssl/owncloud.pem
        SSLCertificateKeyFile /usr/local/etc/apache22/ssl/owncloud.key
        DocumentRoot /usr/local/www/owncloud/
<Directory "/usr/local/www/owncloud">
 AllowOverride All
 order allow,deny
 Allow from all
</Directory>
</VirtualHost>

mkdir /usr/local/etc/apache22/ssl/  # Sertifikatımız üçün qovluq yaradaq.
cd /usr/local/etc/apache22/ssl/     # Ünvana daxil oluruq ki, sertifikatı
                                    orda yaradaq.
```

Sertifikatı aşağıdakı verilənlərlə generasiya edirik.

```
openssl req -new -x509 -days 365 -nodes -out
/usr/local/etc/apache22/ssl/owncloud.pem -keyout
/usr/local/etc/apache22/ssl/owncloud.key
Generating a 1024 bit RSA private key
.........++++++
```

```
.............++++++
writing new private key to '/usr/local/etc/apache22/ssl/owncloud.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:Baku
Locality Name (eg, city) []:Xatai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:OpSO
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:owncloud.az
Email Address []:admin.admin@owncloud.az
```

Lazımı unvanlara lazımı yetkiləri verək.

```
chown -R www:www /usr/local/etc/apache22/ssl/
chown -R www:www /usr/local/www/owncloud/
chmod -R 600 /usr/local/etc/apache22/ssl/
chown -R www:www /usr/local/domen/
```

/usr/local/etc/apache22/httpd.conf faylında da 443-cü port üçün Listen əlavə edib restart edin ki, https işləsin.

```
Listen 80
Listen 443
```

Qeyd: Əgər siz apache24-den istifadə edirsınızsə, **/usr/local/etc/apache24/httpd.conf** faylında aşağıdakı sətirlərin qarşısından şərhi silməyi unutmayın:

```
LoadModule rewrite_module libexec/apache24/mod_rewrite.so
LoadModule ssl_module libexec/apache24/mod_ssl.so
LoadModule dav_module libexec/apache24/mod_dav.so
LoadModule vhost_alias_module libexec/apache24/mod_vhost_alias.so
```

Qeyd: Həmçinin apache24-də **/usr/local/dome/bvimcloud.domain.az** vhost config faylı aşağıdakı kimi olacaq:

```
<VirtualHost *:80>
        RewriteEngine on
        ReWriteCond %{SERVER_PORT} !^443$
        RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R,L]
</VirtualHost>
<VirtualHost *:443>
        SSLEngine on
        SSLCertificateFile /usr/local/etc/apache24/ssl/bvimcloud.pem
        SSLCertificateKeyFile /usr/local/etc/apache24/ssl/bvimcloud.key
        DocumentRoot /usr/local/www/owncloud/
<Directory "/usr/local/www/owncloud">
```

```
 AllowOverride All
 Require all granted
</Directory>
</VirtualHost>
```

```
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini # php quraşdırma
                                                            faylını düzəldək
```

/usr/local/etc/php.ini faylın içində aşağıdakı sətiri uyğun olaraq edək:

```
date.timezone = 'Asia/Baku'
```

OwnCloud üçün MySQL-de baza istifadəçi adı və şifrə yaradaq.

```
mysqladmin -u root -h localhost password 'freebsd' # MySQL-in root
                                                     istifadəçisi üçün şifrə
                                                     təyin edək.
```

```
mysql -uroot -p'freebsd'              # MySQL-e daxil olaq ve baza yaradaq.
mysql> CREATE DATABASE owncloud;      # Bazanı yaradırıq.
```

owncloud istifadəçisini yaradırıq və owncloud bazasına localhost-dan yetki veririk.

```
mysql> GRANT ALL PRIVILEGES ON owncloud.* TO 'owncloud'@'localhost'
IDENTIFIED BY "freebsd";
```

Yetkiləri FLUSH edək ki, aktivləşsin

```
mysql> FLUSH PRIVILEGES;
```

Sonra OWNCLOUD-un istifadəçiləri üçün Global qovluq yaradırıq və web server üçün yetki veririk.

```
mkdir /home/owncloud_data
chown -R www:www /home/owncloud_data
```

Sonra Windows maşınımızda test üçün **c:\windows\system32\drivers\etc\hosts** faylına Aşağıdakı sətiri əlavə edirik.

```
192.168.11.200 owncloud.az
```

Windows maşınımızın Browserinde http://owncloud.az daxil edirik və görürük ki, https linkinə forward edilirik. Şəkildə gorunduyu kimi, admin user və parol, həmçinin MySQL bazası üçün verilənlərini daxil edib **Finish setup** düyməsinə sıxırıq.

Mənim halımda 7.2.1 release idi və aşağıdakı fayl tələb edilirdi:

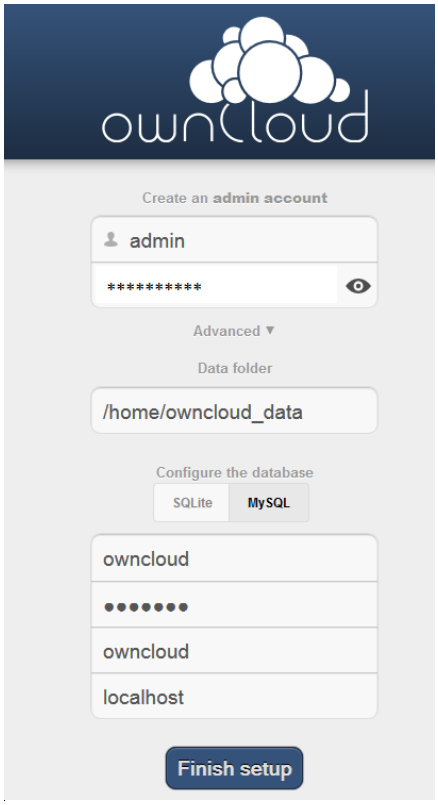
```
touch /home/owncloud_data/.ocdata
```

Həmçinin OwnCloud üçün nəzərdə tutduğumuz data qovluğu üçün **www** istifadəçisi və qrupuna yetki veririk:

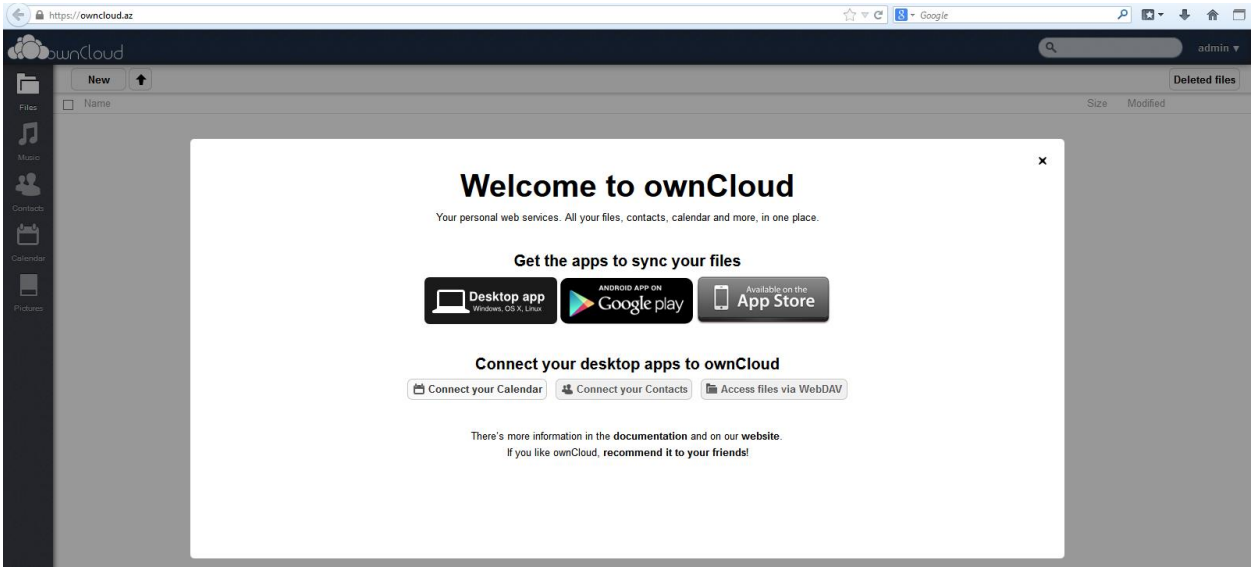
```
chown -R www:www /home/owncloud_data/
```

Eynilə **php5-bcmath** modulu tələb edilirdi:

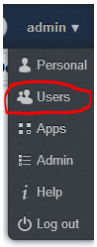
```
cd /usr/ports/math/php5-bcmath            # Port ünvanına daxil oluruq
make install                              # Yükləyirik
```



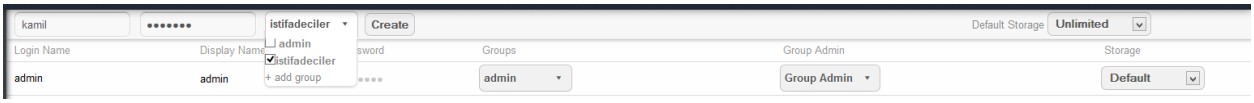
Sonra isə aşağıdakı şəkil çıxdısa demək hərşey əladır.



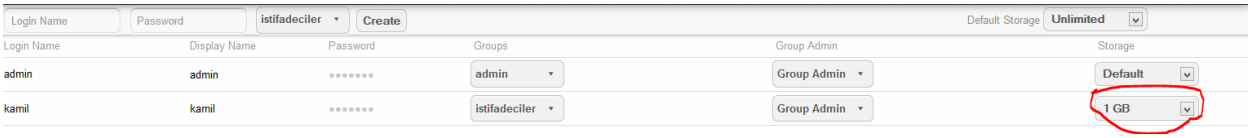
Artıq istifadəçiləri yaradaq və sinxronizasiya üçün yer verək. Sağ tərəfdə **admin -> Users** düyməsinə sıxırıq.



Sonra **Groups -> add group** düyməsini sıxırıq və istifadəçilər adlı qrup yaradırıq. Həmin səhifədə **kamil** adli istifadəçisi və **parol** daxil edib **istifadəçilər** qrupunu seçirik və **Create** düyməsinə sıxırıq.

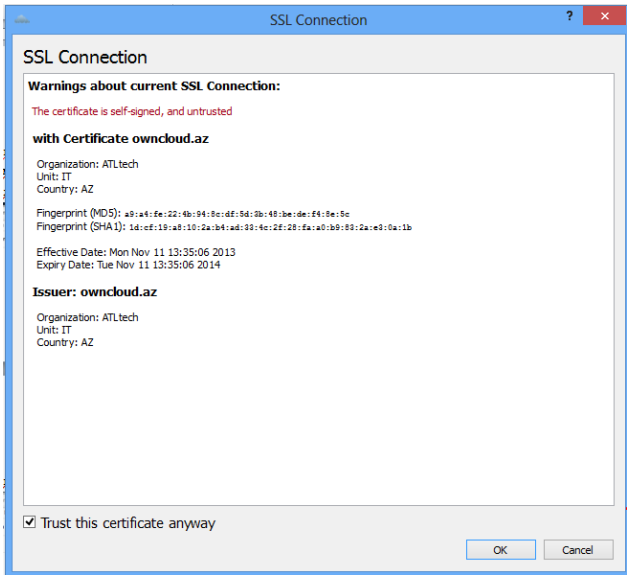


Sonra **kamil** adli istifadəçiyə 1GB yer istifadə etmək imkani veririk.

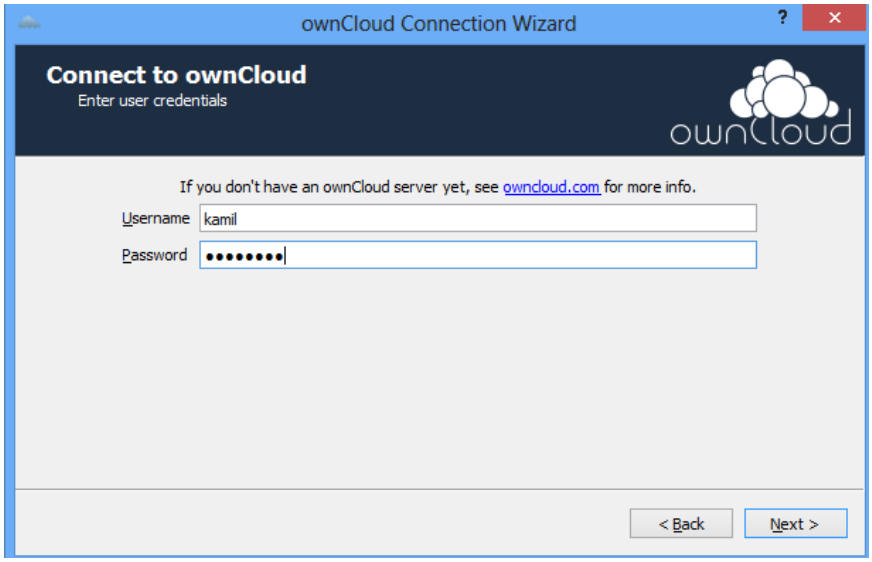


Sonra isə Windows maşınımızda Client proqramını yükləyək və sınaqdan keçirək. Yüklənmə proseduru çox asandır. Sadəcə **Full install** seçirik və **Next** düyməsinə sıxırıq. Sonda **'Run owncloud'** seçib **Finish** edirik. Açılan səhifədə serverimizin adını vəya IP ünvanını daxil edirik(Bizim halda **https://owncloud.az**).

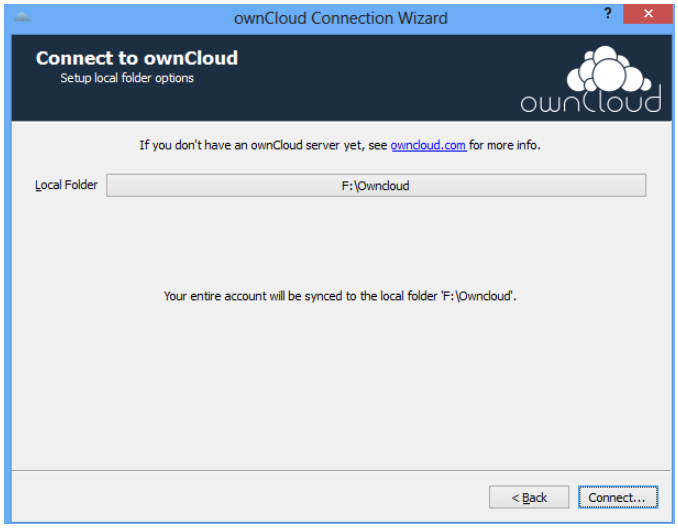
Çixan səhifədə **'Trust Certificate anyway'** seçib **OK** düyməsinə sıxırıq. Şəkildəki kimi.



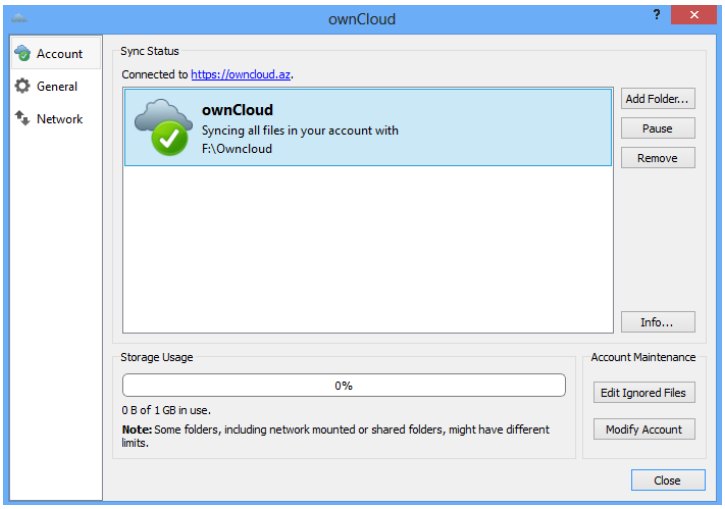
İstifadəçi adı və **şifrəni** daxil edib **Next** düyməsinə sıxırıq.



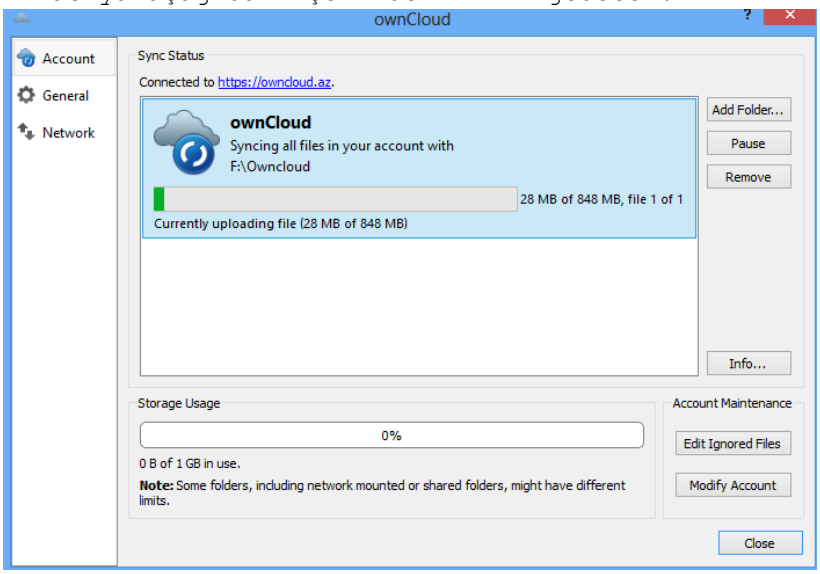
Uyğun qovluğumuzu seçirik və **Connect** düyməsinə sıxırıq. Şəkildəki kimi.



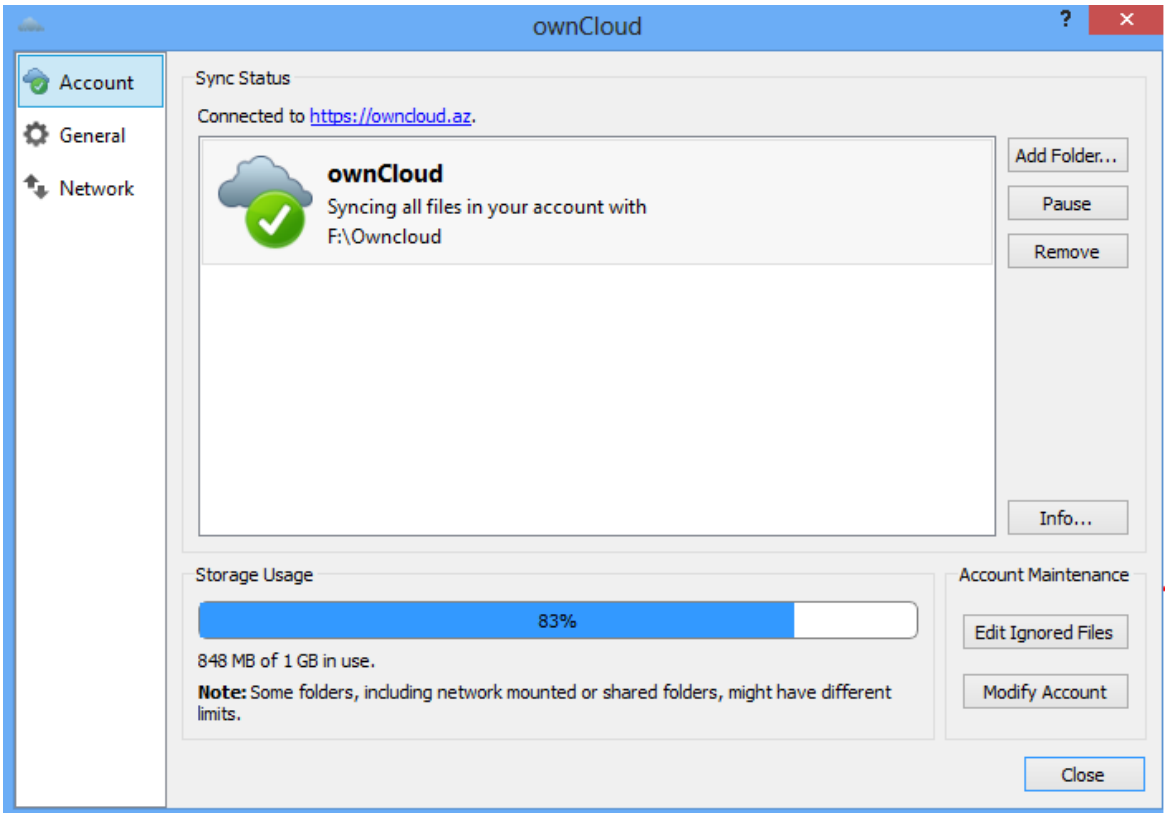
Sonda aşağıdakı şəkildəki kimi nəticə əldə etmiş olacağıq. Artıq kamil adlı istifadəçi öz kompunda, **F:\Owncloud** adlı folderə nə informasiya atsa o avtomatik olaraq https://192.168.11.200 serverine sinxronizasiya ediləcək.



Sinxronizasiya aşağıdakı şəkildeki kimi gedəcək.



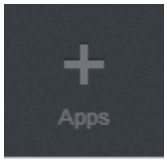
Nəhayət sonda müəyyən sinxronizasiya bitdikdən sonra aşağıdakı şəkil çap ediləcək. Bununla da OwnCloud serverimiz test edilmiş olacaq.



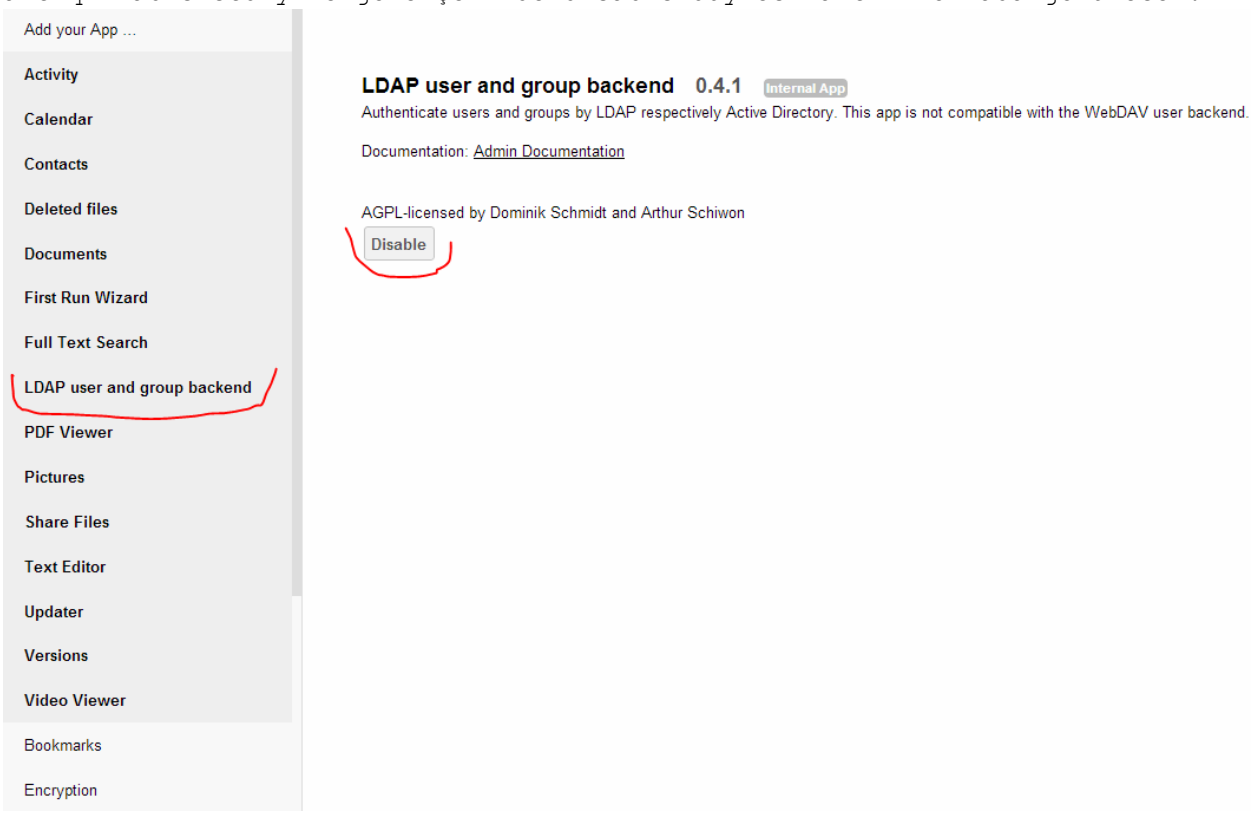
WebDav qoşub Browser Clientlərdən istifadə eləmək istəsəniz bu linkdən http://doc.owncloud.org/server/5.0/user_manual/files/files.html yararlana bilərsiniz.

OwnCloud-un Domain Controller ilə integrasiya edilməsi

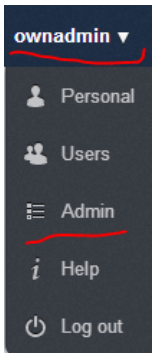
OwnCloud-u FreeBSD maşına yüklədikdən sonra, WEB ilə lazımı linkə daxil olursunuz və sol tərəf aşağıda **Apps** düyməsinə sıxırsınız(Şəkildəki kimi):



Sonra isə LDAP üçün lazım olan App-ı enable eləmək lazımdır. Mənim halımda artıq Enable etdiyimə görə şəkildə disable düyməsi aktiv formada görünəcək.



Sonra isə sistemi yüklədiyimizdə yaratdığımız **ownadmin** userin **Admin** interfeysinə daxil oluruq(Şəkildəki kimi).

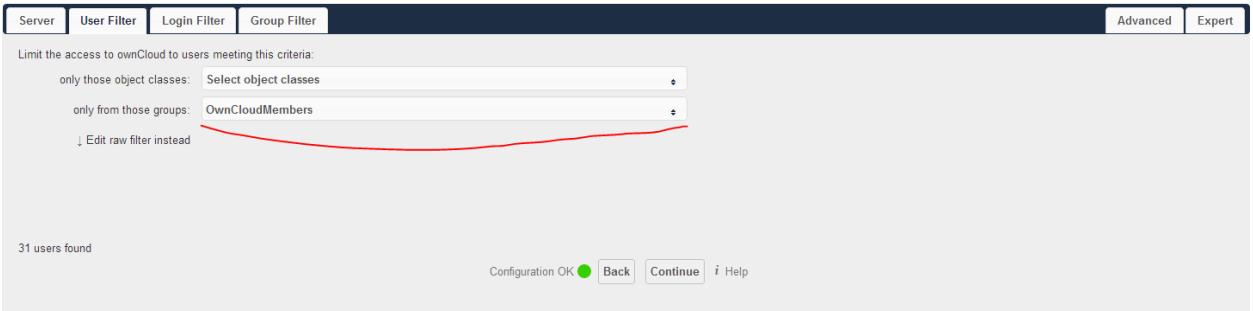


Bizim Domain Controllerimiz üçün verilənlər aşağıdakılardır:

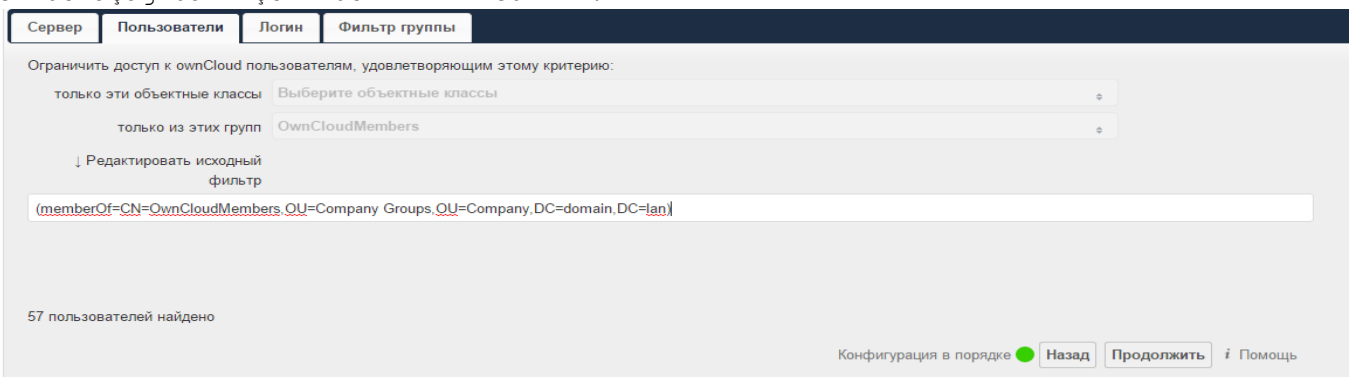
DC: **DOMAIN.LAN (Port Ldap: 389)**
Domain Admin: **DCADM**
Domain Admin Pass: **freebsd**
Cloud istifadəçilər üçün qrup: **OwnCloudMembers**

Öncə **Server** başlığında Şəkildə göstərildiyi kimi məlumatları yerləşdiririk və **Continue** düyməsinə sıxırıq(Şəkildəki kimi).
domain.lan 389
CN=DCADM,CN=Users,DC=DOMAIN,DC=lan
Shifre
DC=DOMAIN,DC=lan

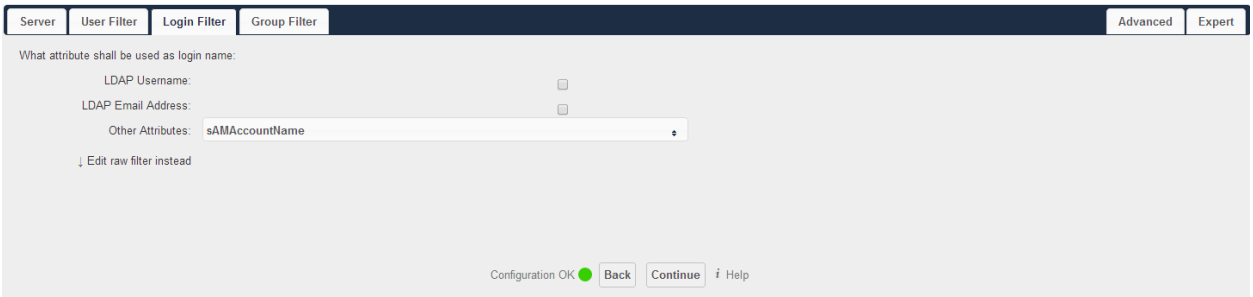
Sonra **User Filter** TAB-ına daxil oluruq və heç bir class seçmədən sadəcə bizə lazım olan istifadəçilərin yerləşdiyi qrupu seçirik. Yeni **OwnCloudMembers** qrupunu(Şəkildəki kimi):



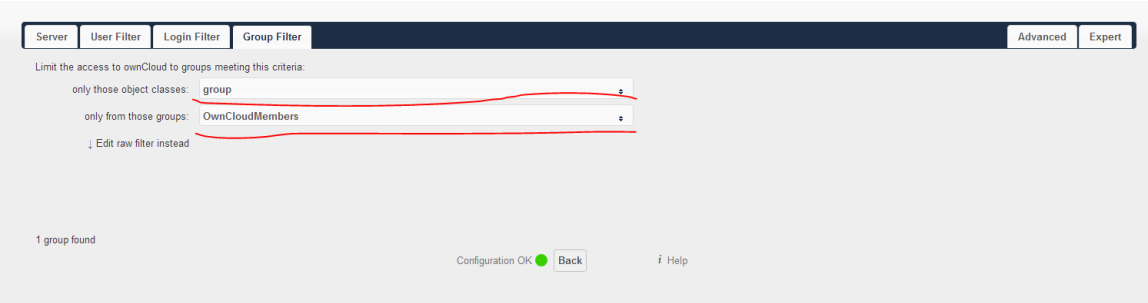
yada ki, **Edit raw filter instead** düyməsini sıxıb, lazımı LDAP filteri özünüz yaza bilərsiniz.
Misal üçün **(memberOf=CN=OwnCloudMembers,OU=Company Groups,OU=Company,DC=domain,DC=lan)** sintaksisi ilə **domain.lan** DC-də **Company Groups, Company** OU-da olan və yalnız **OwnCloudMembers** qrupunun bütün üzvlərinin Cloud-dan istifadəsinə izin veririk. Misal üçün Domain.LAN DC-sində aşağıdaki şəkildəki kimi edirik:



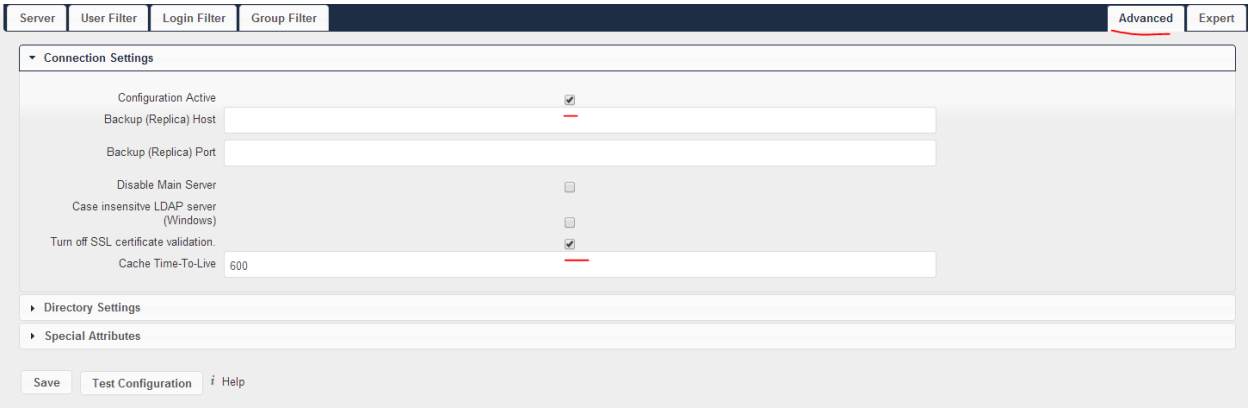
Login Filter bölümündə **LDAP Username**-i seçmirik və **Other Attributes**-de **sAMAccountName** seçirik(Şəkildəki kimi):



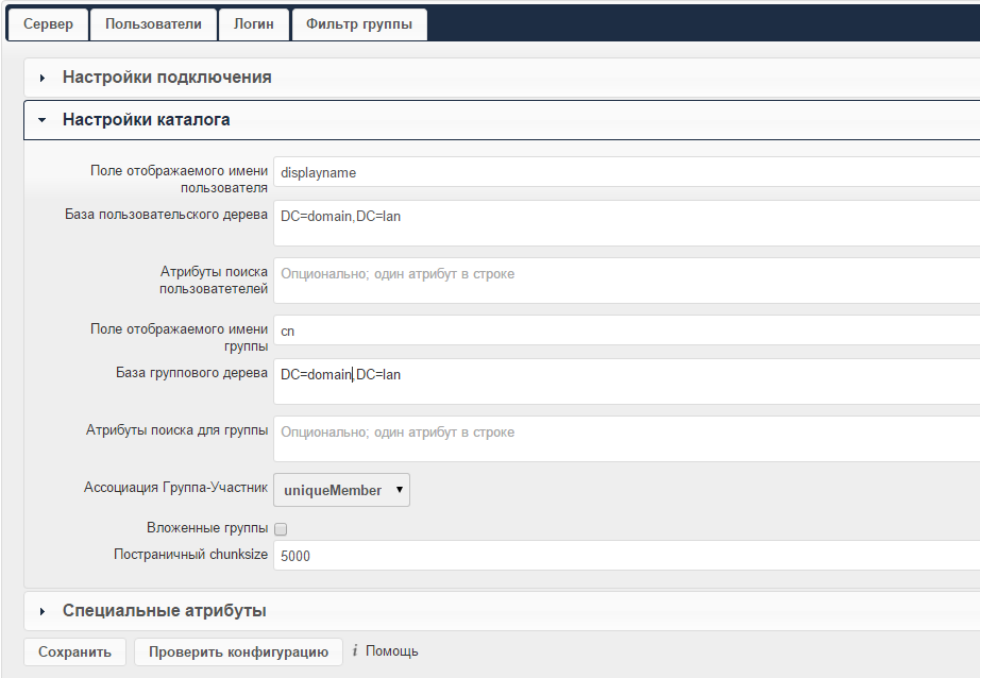
Group Filter-de **Object Flass** olaraq **group** seçirik və **only from those groups**-da **OwnCloudMembers** qrupunu seçirik(Şəkildəki kimi):



Advanced-e daxil oluruq və **Connections Settings**-de **Configuration Active** və **Turn off SSL certificate validation** seçirik



Advanced bölümündə **Directory Settings**-de şəkilde göstərilən verilənlər seçilir və mütləq **Group-Member association: uniqueMember** seçilir:



Advanced-de Special Attributes-də heç bir şey əlavə edilmir və boş qalır. **Exterpt**-e də daxil olmadan **Save** və **Test Configuration** düyməsi sıxılır.

FreeBSD 10.1 x64 Pydio Cloud qurulması

Pydio(əvvəl Ajaxplorer) - ayrılmış serverdə məlumatların sinxronizasiya edilməsi üçün, açıq qaynaqlı proqram təminatıdır. Proyekt 2009-cu ildə Charles du Jeu tərəfindən yaradılmışdır. Php dilində yazılmışdır. İdarəetmə üçün MySQL verilənlər bazası istifadə edilir. İstənilən tip desktop(Windows, MAC və Linux) və mobil əməliyyat sistemləri(Android, iOS) üçün müştəri proqram təminatına malikdir. Məlumatlar eynilə WEB interfeys vasitəsilə də idarə edilə bilər.

İmkanları:

- Faylların qovluq ağac strukturunda saxlanılması(WebDAV vasitəsilə)
- Məlumatların SSL/TLS vasitəsilə şifrələnməsi
- Desktop proqramından sinxronizasiya
- İstifadəçi rollarının idarəedilməsi(LDAP istifadəçi bazası ilə inteqrasiya)
- Digər istifadəçilərlə qovluq və faylların paylaşılması
- Sintaksis göstəricisi ilə mətn redaktoru
- Şəkil fayllarının redaktoru
- Audio və video faylların işə salınması
- Kənar anbarlarla inteqrasiya. Amazon S3, FTP ya da MySQL verilənlər bazası.
- Mobil platformalar üçün proqram təminatı

Məqsədimiz FreeBSD10.1 üzərində Cloud serverin qurulmasıdır. Nəzərdə tutulur ki, artıq FreeBSD serverimizdə portlarla paketlər yenilənib, Apache PHP MySQL yüklənmiş və qurulmuşdur. Ancaq **/usr/port/lang/php56-extensions** ünvanından php genişlənmələri yüklədikdə, mütləq **bcmath, bz2, calendar, Core, ctype, curl, date, dom, ereg, exif, fileinfo, filter, gd, gettext, hash, iconv, imap, json, ldap, libxml, mbstring, mcrypt, mhash, mysql, mysqli, mysqlnd, openssl, pcre, PDO, Phar, posix, pspell, Reflection, session, SimpleXML, snmp, SPL, standard, tokenizer, xml, xmlreader, xmlrpc, xmlwriter, xsl, Zend OPcache, zip, zlib** modullarını seçmək lazımdır.

Pydio-nun PHP üçün tələb elədiyi dəyişiklikləri edirik. Bunlar upload dəyişənləri və **session.save_path** ilə sessiyaların saxlanılması ünvanıdır:

```
root@pydio:~ # cd /usr/local/etc/
root@pydio:/usr/local/etc # cp php.ini-production php.ini
```

/usr/local/etc/php.ini faylında aşağıdakı dəyişənləri uyğun olaraq edirik:

```
upload_max_filesize = 1024M
post_max_size = 1024M
output_buffering = Off
session.save_path = "/tmp"
```

Serverimizin adı **pydio.opensource.az** olacaq. Buna görə də serverimizi VirtualHost-la ad prinsipinə uyğun olaraq qurmaq lazımdır.

Apache serverimizə yeni quraşdırma fayllarının işləyəcəyi qovluğu təyin edirik.

```
root@pydio:~ # echo "Include /usr/local/domen/*" >>
/usr/local/etc/apache24/httpd.conf

root@pydio:~ # mkdir /usr/local/domen
```

/usr/local/domen/pydio.opensource.az faylı yaradırıq və tərkibinə aşağıdakı sətirləri əlavə edirik ki, virtual hostumuz işləsin. Faylda olan SSL sertifikatı ardıcıl yaradacağıq çünki, pydio ilk yüklənməsində bu sertifikatı tələb edir:

```
<VirtualHost *:80>
        RewriteEngine on
        ReWriteCond %{SERVER_PORT} !^443$
        RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R,L]
        ServerAdmin webmaster@email.com
        ServerName pydio.opensource.az
</VirtualHost>
<VirtualHost *:443>
        SSLEngine on
        SSLCertificateFile /usr/local/etc/apache24/ssl/pydio.pem
        SSLCertificateKeyFile /usr/local/etc/apache24/ssl/pydio.key
        DocumentRoot /usr/local/www/pydio/
        CustomLog "/var/log/pydio_access.log" common
        ErrorLog /var/log/pydio_error.log
        <Directory "/usr/local/www/pydio">
                AllowOverride all
                Require all granted
        </Directory>
</VirtualHost>
```

Qeyd: Unutmayın **/usr/local/etc/apache24/httpd.conf** faylında mütləq **rewrite_module** və **ssl_module** sətirlərinin qarşısından şərhləri silmək və **Listen 443** sətirini yazimaq lazımdır ki, port qulaq assın.

Jurnal fayllarını yaradaq:

```
root@pydio:~ # touch /var/log/pydio_access.log /var/log/pydio_error.log
```

SSL qovluğu yaradıb içinə daxil oluruq və ardınca **pydio** https-lə işləyə bilməsi üçün sertifikat yaradırıq:

```
root@pydio:~ # mkdir /usr/local/etc/apache24/ssl/
root@pydio:~ # cd /usr/local/etc/apache24/ssl/
root@pydio~ # openssl req -new -x509 -days 365 -nodes -out pydio.pem -keyout
pydio.key
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:Baku
Locality Name (eg, city) []:Narimanov
Organization Name (eg, company) [Internet Widgits Pty Ltd]:OpSO
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:pydio.opensource.az
Email Address []:admin@opensource.az
```

MySQL verilənlər bazası yaradırıq və istifadəçiyə hüquq təyin edirik:

```
root@pydio:~ # mysql -uroot -p
```

```
mysql> create database pydiodb;
Query OK, 1 row affected (0.02 sec)

mysql> GRANT ALL ON pydiodb.* TO pydiouser@localhost IDENTIFIED BY 'freebsd';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

Pydio-nun son versiya mənbə kodlarını Internetdən endirib açırıq:

```
root@pydio:~ # pkg install wget
root@pydio:~ # wget
http://sourceforge.net/projects/ajaxplorer/files/pydio/stable-
channel/6.0.8/pydio-core-6.0.8.zip
root@pydio:~ # unzip pydio-core-6.0.8.zip
```

Pydio üçün **PUBLIC_HTML** qovluğu yaradaq və data qovluğuna lazımı yetkiləri təyin edək:

```
root@pydio:~ # mv pydio-core-6.0.8 /usr/local/www/pydio
root@pydio:~ # chown -R www:www /usr/local/www/pydio/
root@pydio:~ # chmod -R 777 /usr/local/www/pydio/data/
```

Öncədən **/usr/local/www/pydio/conf/bootstrap_conf.php** faylında aşağıdakına uyğun olaraq **define** sətirlərinin qarşısına **setlocal** ilə **UTF-8** kodirovkasını dəyişirik:

```
setlocale(LC_ALL, "en_US.UTF-8");
//define("AJXP_LOCALE", "en_EN.UTF-8");
//define("AJXP_LOCALE", "");
```

Web serverimizi yenidən işə salırıq ki, dəyişikliklər işə düşsün:

```
root@pydio:~ # /usr/local/etc/rc.d/apache24 restart
```

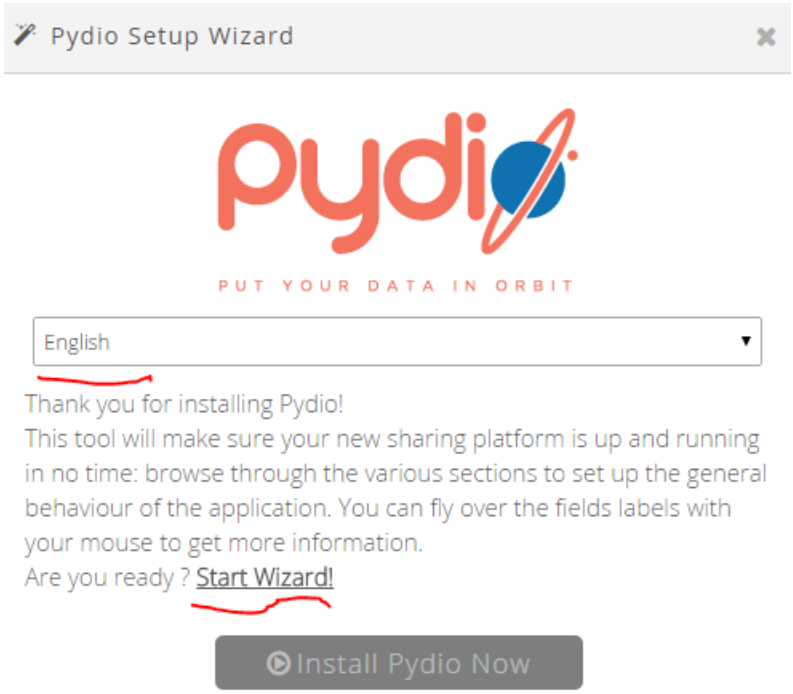
Artıq istənilən desktop maşında hansısa browser açırıq və http://pydio.opensource.az/ səhifəsinə daxil oluruq.

Qeyd: Əgər sizdə DNS yoxdursa və web səhifəyə adla daxil olmaq istəyirsinizsə istənilən Windows maşında **C:\Windows\System32\drivers\etc\hosts** faylına və istənilən UNIX/Linux maşında **/etc/hosts** faylına aşağıdakı sintaksislə sətiri əlavə etməniz yetər:

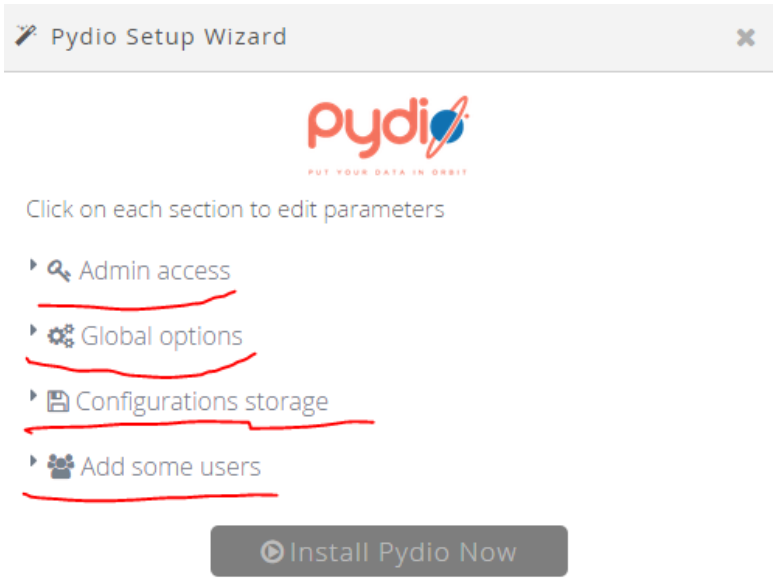
XX.XX.XX.XX pydio.opensource.az pydio

WEB səhifə qırmızı açılacaq çünki sertifikatınız Self Signed(Özünüz özünüzü imzalamışsınız. Normaldir)-dir. Web səhifənizdə **Pydio** yoxlanışlarının nəticəsini yaşıl rəngli **OK** sətirləri ilə görəcəksiniz. Mövcud versiya üçün heç bir səhv çap edilməyəcək ancaq, gələcək versiyalarda səhvlər səhifədə görünəcək və onlari həll etməlisiniz. Əks halda yükləməni davam etmək mümkün olmayacaq.

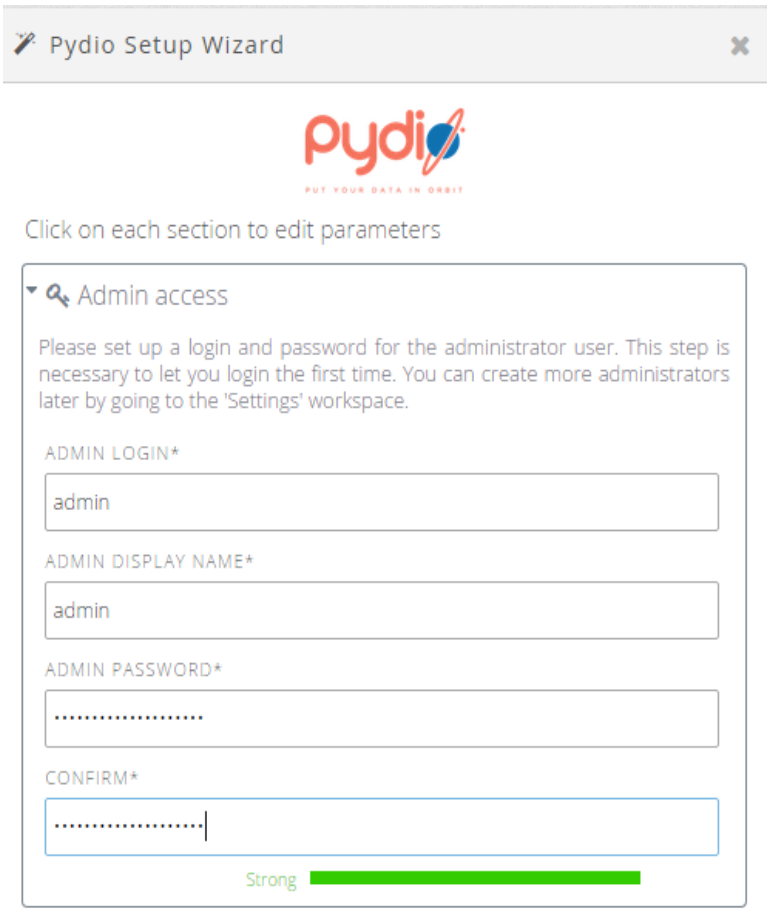
Aşağıdakı kimi, **English** seçirik və **Start Wizard** düyməsinə sıxırıq:



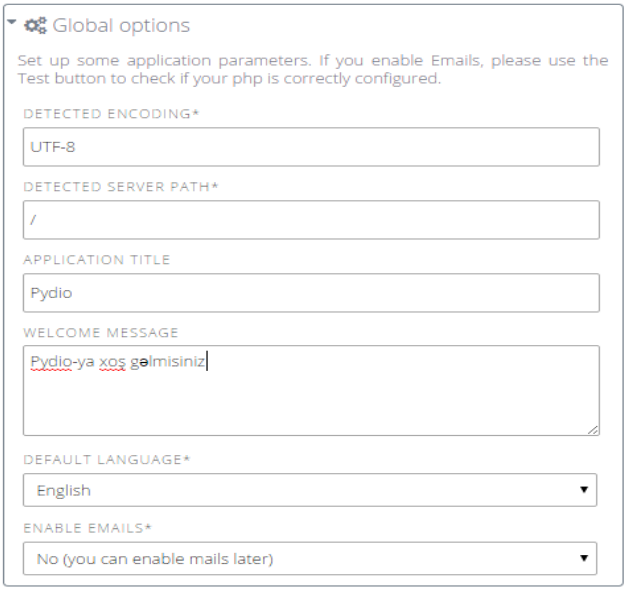
Növbəti səhifədə hər bir seksiyanı ardıcıllıqla quraşdırırıq:



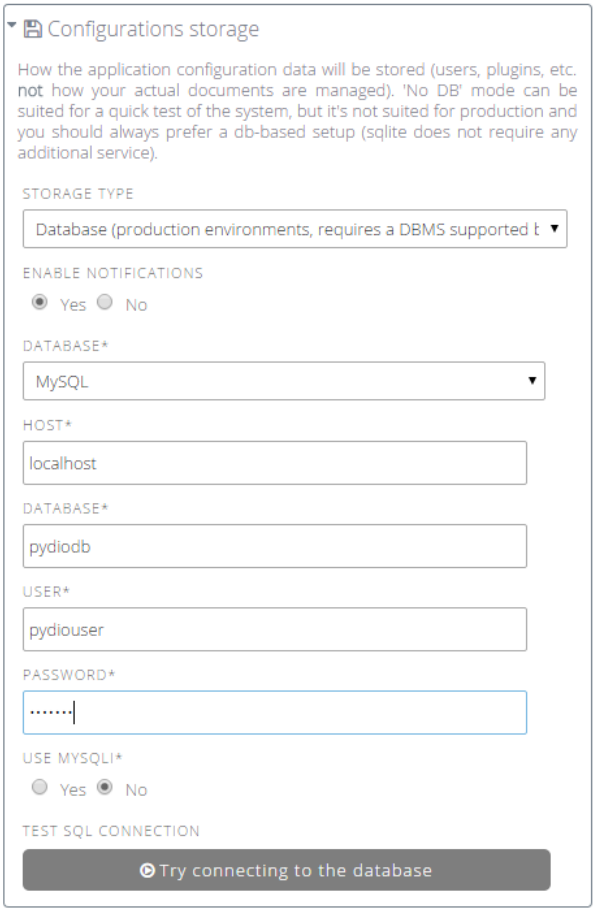
Pydio WEB inzibatçı üçün istifadəçi adı və şifrə təyin edirik(Şifrə çətin olmalıdır. Əks halda növbəti səhifəyə izin verməyəcək):



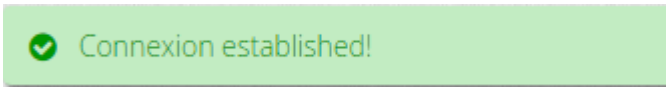
Qlobal quraşdırmaları edirik:



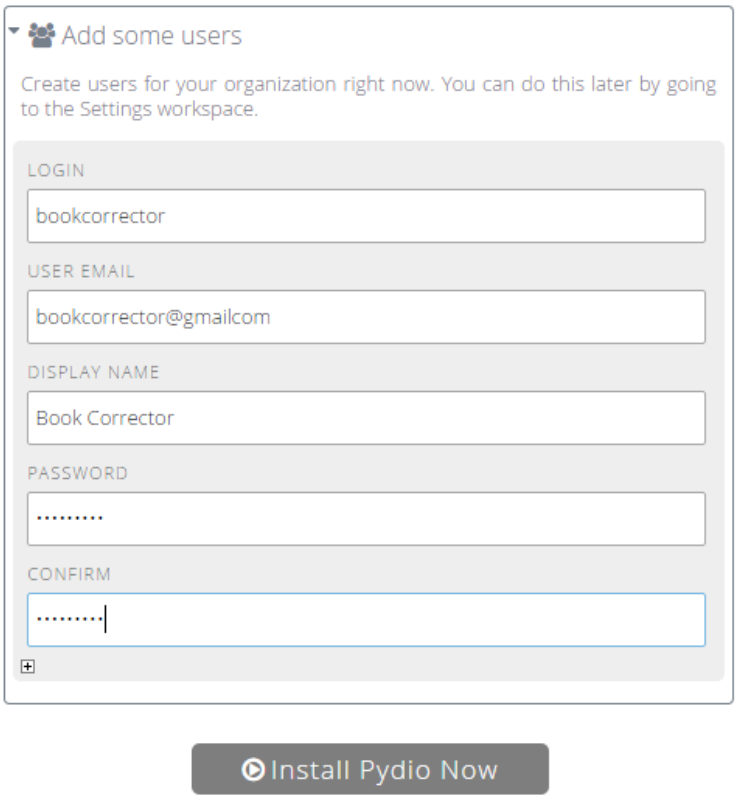
Verilənlər bazası üçün quraşdırmaları edirik. Öncə yaratdığımız MySQL istifadəçi, baza və şifrəsini daxil edib, **Try connecting to the database** düyməsinə sıxırıq ki, qoşulmanı yoxlayaq:



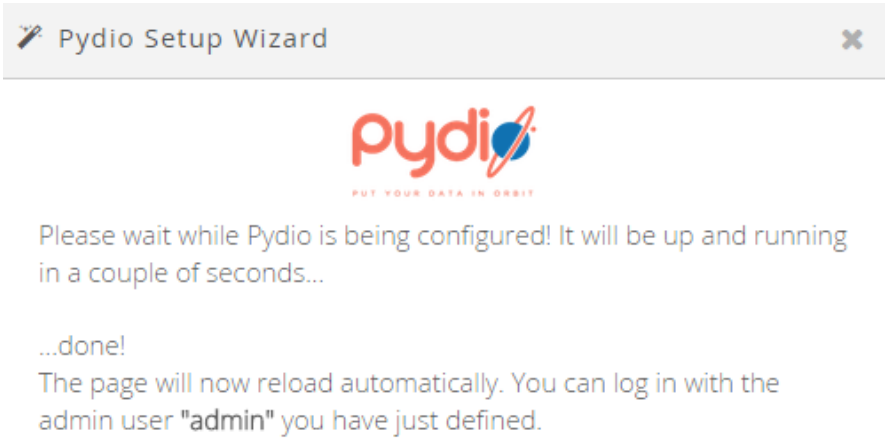
Uğurlu nəticə aşağıdakı kimi yaşıl rəngdə olmalıdır:



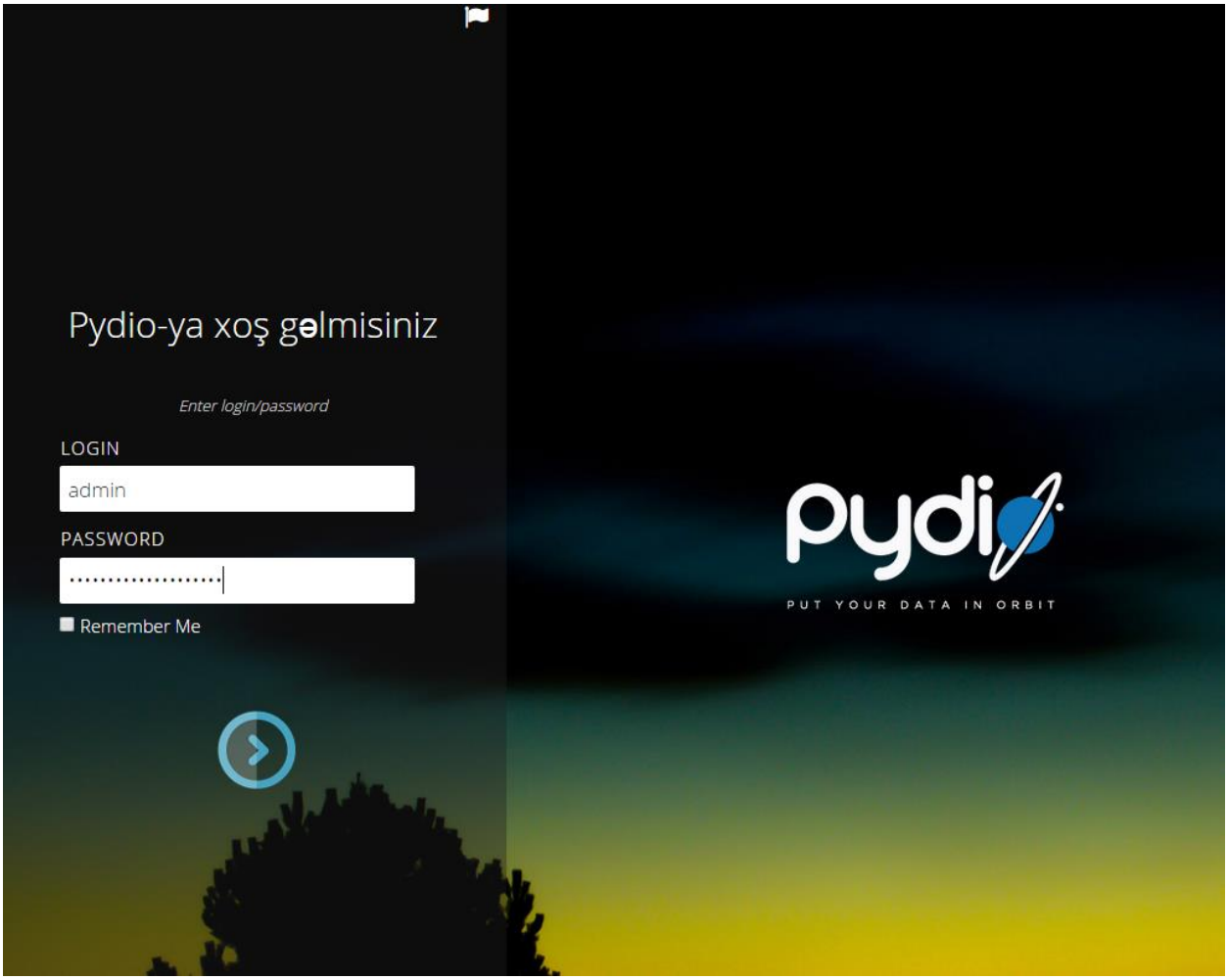
Sınaq üçün **bookcorrector** adlı yeni bir istifadəçi verilənləri daxil edirik və **Install Pydio Now** düyməsini sıxırıq ki, yükləməmiz davam etsin:



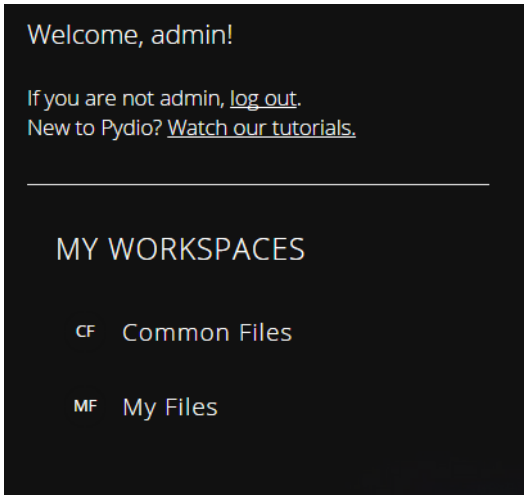
Aşağıdakı şəkildəki kimi bir neçə saniyə vaxt keçəcək:



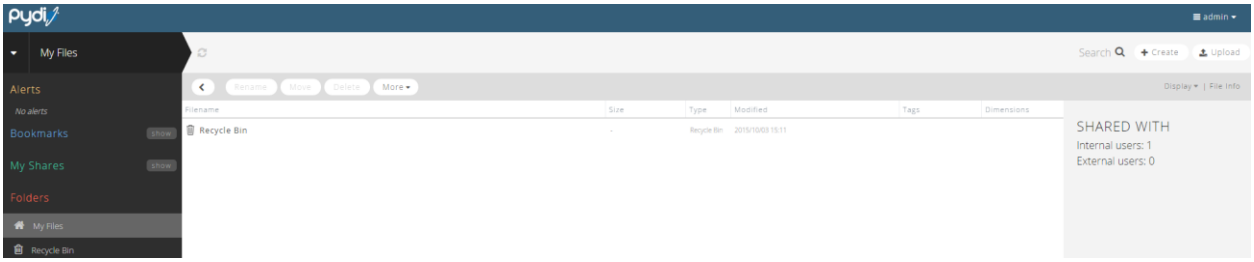
Son nəticə aşağıdakı şəkildəki kimi olacaq. Istifadəçi adı **admin** və şifrəni yazıb, səhifəyə daxil oluruq:



Daxil olduqdan sonra **My Files** düyməsini sıxıb özümüzə aid olan admin fayllarına baxa bilərik:

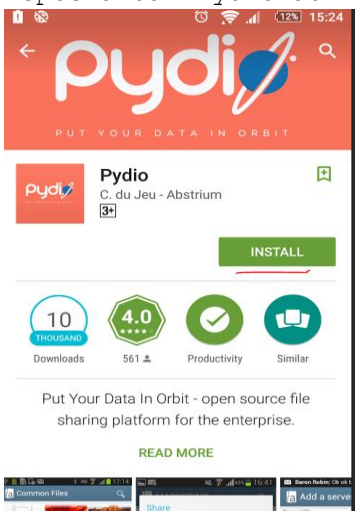


Açılan səhifə aşağıdakı kimi olacaq:

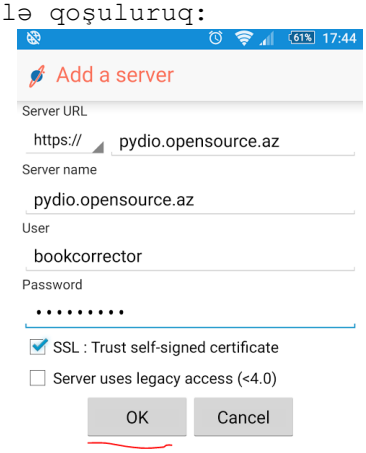


Pydio bütün platformalar üçün pulsuz client proqramları təklif edir. Bu proqramları Android və iOS üçün öz reposlarından və Windows, Linux/UNIX, MAC üçün isə https://pyd.io/apps/pydio-sync/ linkindən əldə edə bilərsiniz.

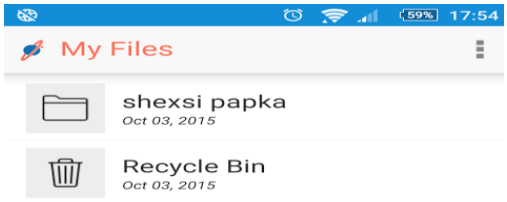
Adroid üçün reposlardan Pydio adlı proqramı yükləməlisiniz(Şəkildəki kimi):



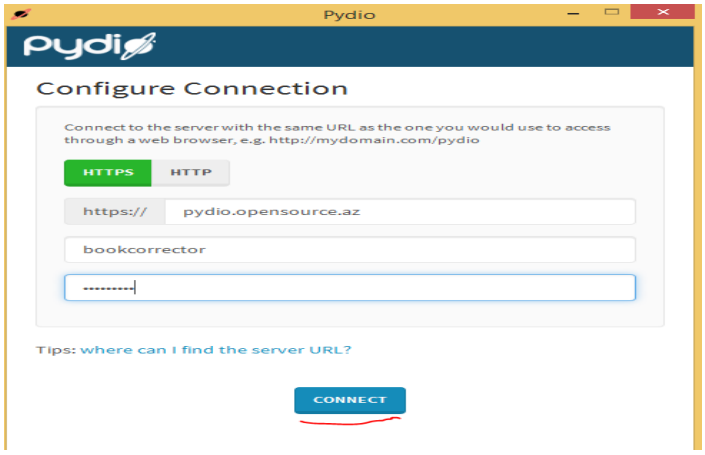
Yüklədikdən sonra yaratdığımız istifadəçi adı ilə pydio cloud-umuza https protokolu ilə qoşuluruq:



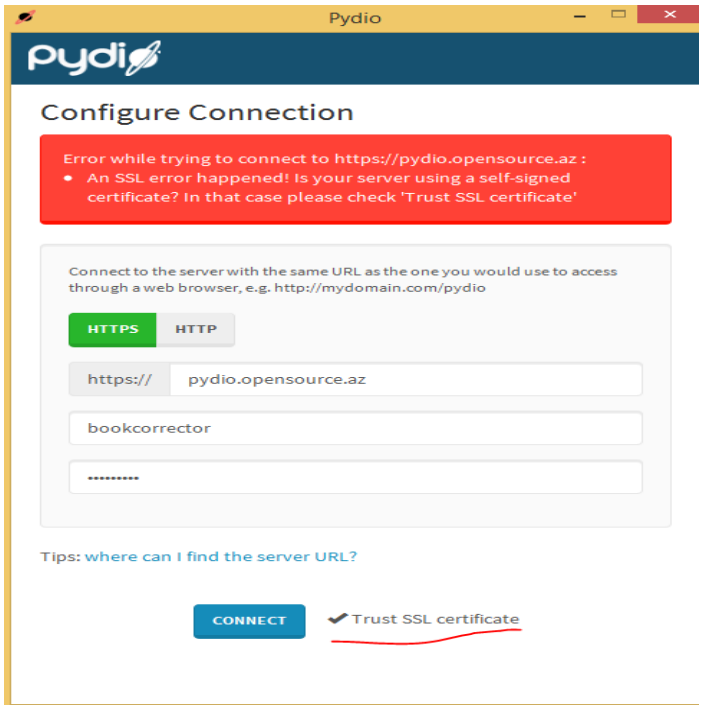
Uğurlu nəticə aşağıdakı kimi olacaq:



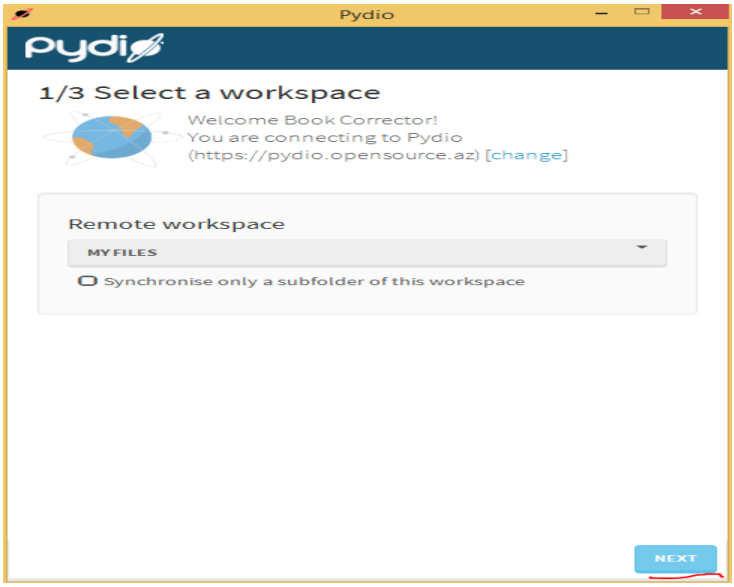
Windows üçün isə platformaya uyğun olan PydioSync versiyasını yükləmək lazımdır. Yüklədikdən sonra, ilk quraşdırmalar aşağıdakı kimi olacaq:



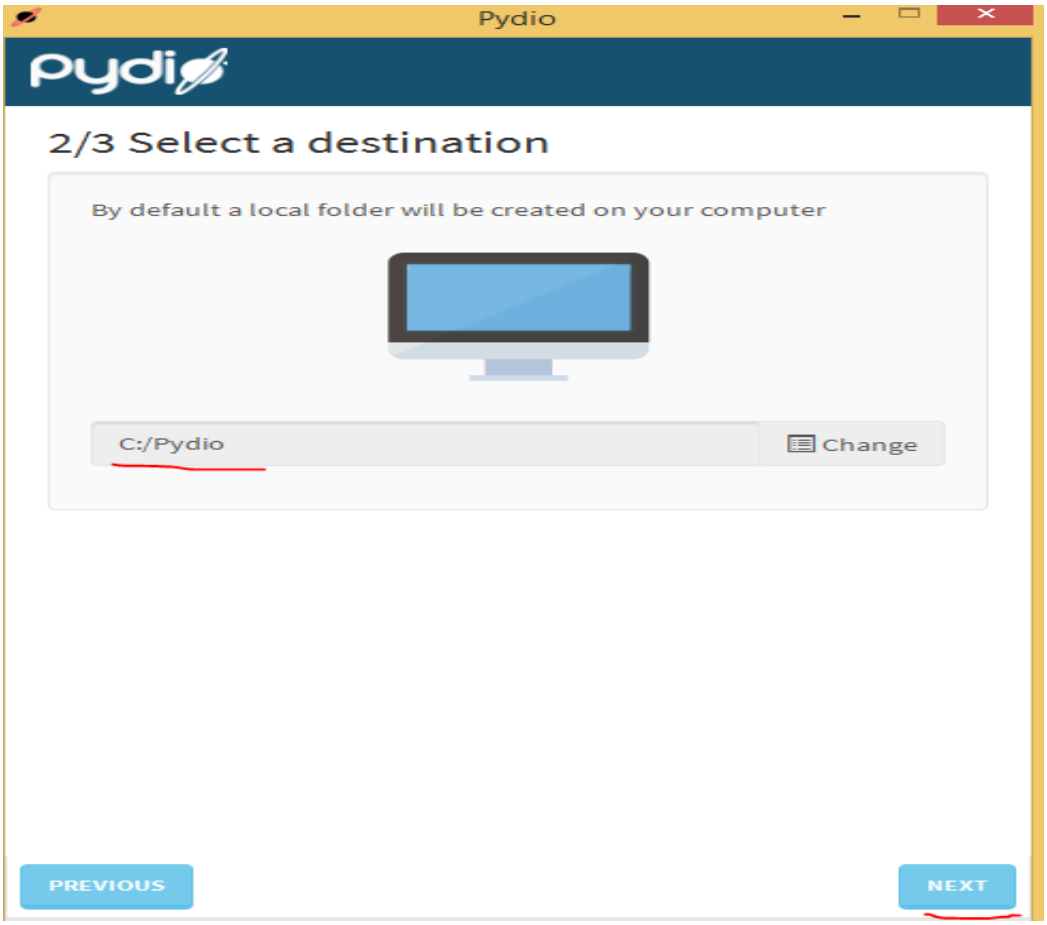
SSL səhvi çap ediləcək və şəkildəki kimi seçirik:



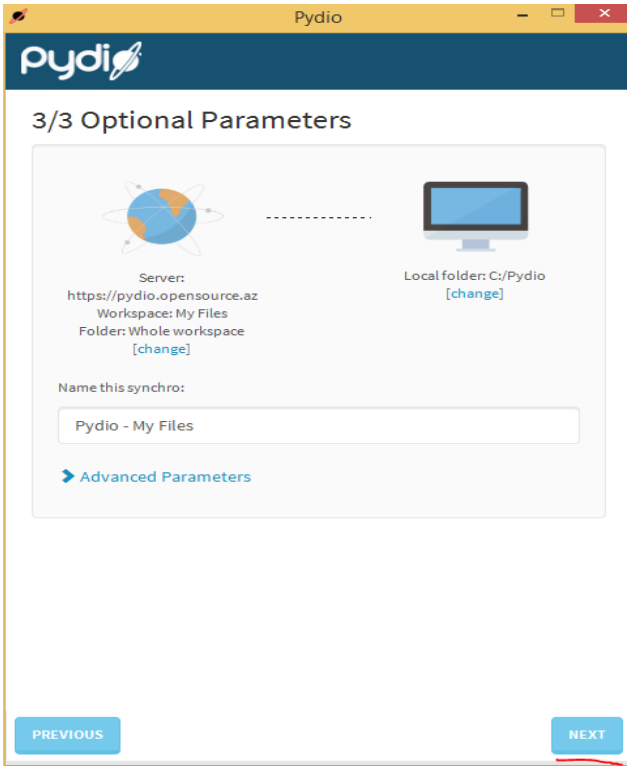
Sonra uzaq serverdə iş yerini təyin edib, **Next** düyməsini sıxırıq:



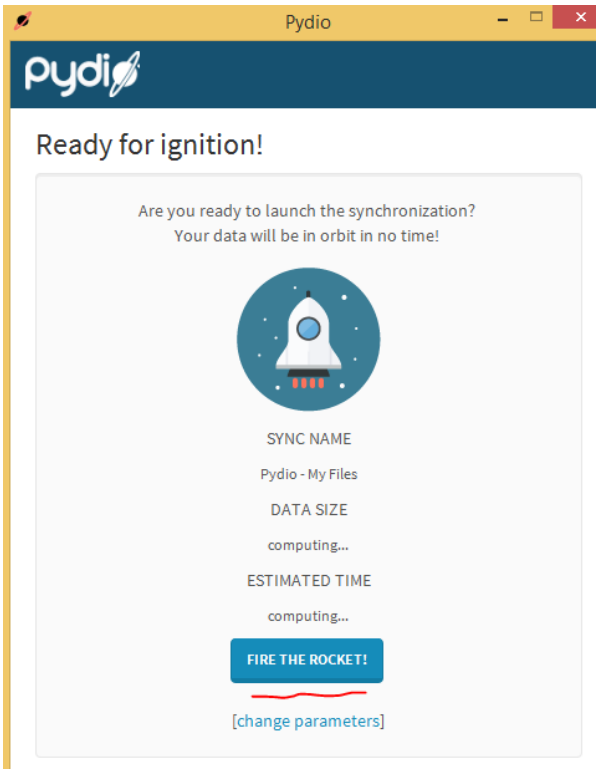
Öz desktopumuzda sinxronizasiya ediləcək qovluğu təyin edib, **Next** düyməsinə sıxırıq:



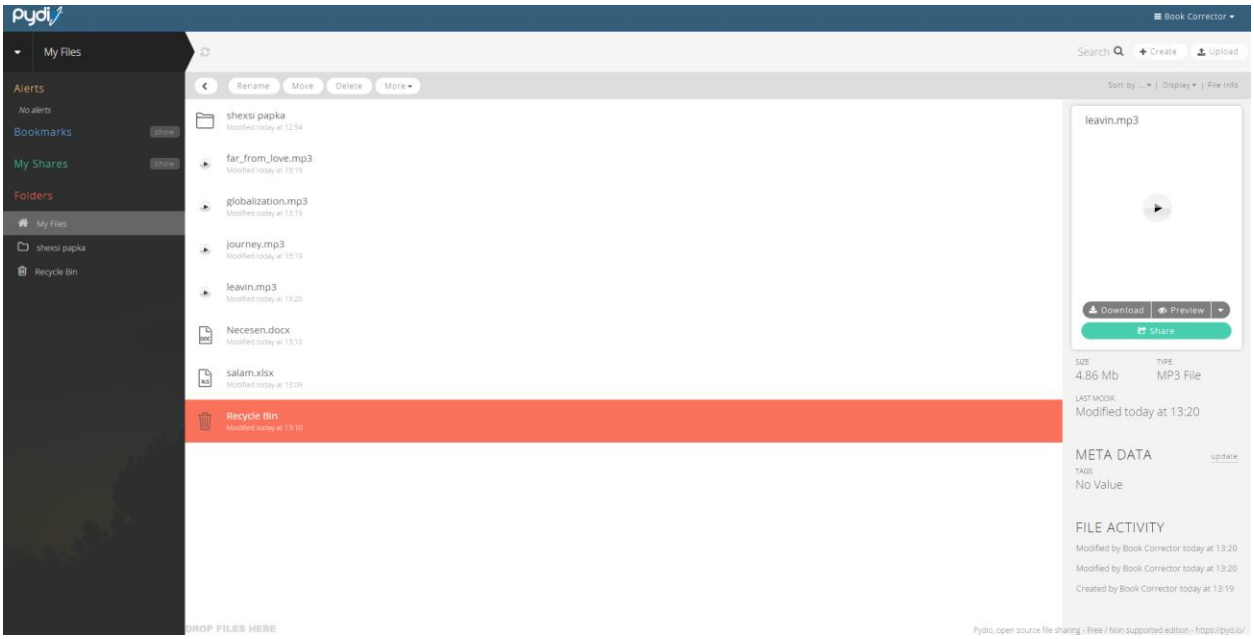
Növbəti səhifəni olduğu kimi qəbul edib **Next** düyməsinə sıxırıq:



Son nəticə aşağıdakı kimi olacaq(**FIRE THE ROCKET!** düyməsinə sıxırıq):



Həm Android və həm də Windows-da qovluqlara hansısa faylları yerləşdirdikdən sonra WEB serverimizdə gördüyümüz nəticə aşağıdakı kimi olacaq:



BÖLÜM 3

Daxili resursların planlaşdırılması sistemləri(ERP)

- **Dolibarr ERP CRM qurulması yüklənməsi və qurulması**
- **Ubuntu 14.04 üzərində OpenERP oDoo-nun qurulması**

Müəyyən sayda daxili işçi tərkibinə sahib olan şirkətin bir neçə şöbəsi olur. Şirkət ən azı təchizat, insan resursları, anbar, mühasibatlıq və İT şöbələrindən ibarət olarsa bu şöbələr arasında rəsmi sənədlərin axını istər-istəməz yaranacaq. Bu halda kağızla işləmək axının qeyri düzgün işləməsi və narahatçılığa gətirib çıxaracaq. Başlığımız bu axının avtomatlaşdırılmasını açıqlayır.

Dolibarr ERP CRM qurulması yüklənməsi və qurulması

Dolibarr ERP CRM – tərkibində resursların planlamasını(ERP) və müştərilərlə qarşılıqlı əlaqənin idarəetməsinə sahib olan, kiçik və orta biznes üçün pulsuz modullu proqram təminatıdır. Funksiyalar tələbdən asılı olaraq işə salına və ya dayandırıla bilər.

Dolibarr verilənlərinin saxlanılması üçün MySQL, PostgreSQL ve SQLite3 istifadə edilə bilər. Bu bolümdə biz Dolibarr proqram təminatının FreeBSD OS üzərində PostgreSQL verilənlər bazasının istifadəsi ilə qurulmasını açıqlayırıq.

Nəzərdə tutulur ki, artıq şəbəkə qurulmuş və portlar yenilənmişdir. Hər hal üçün paketləri yeniləyirik:

```
root@dolibarr:~ # pkg update -f
```

Server **dolibarr.opensource.az** adı ilə işleyəcək.

Istifadəçimizin ev qovluğuna dolibarr-i endiririk:

```
root@dolibarr:~ # pkg install wget
root@dolibarr:~ # cd ~
root@dolibarr:~ # wget --no-check-certificate
https://github.com/Dolibarr/dolibarr/archive/develop.zip
```

Arxivi açırıq:

```
root@dolibarr:~ # unzip develop.zip
```

Arxiv **dolibarr-develop** adlı qovluğa açılır.

Ardınca Apache2.4-u portlardan yükləyirik:

```
root@dolibarr:~ # cd /usr/ports/www/apache24
root@dolibarr:/usr/ports/www/apache24 # make -DBATCH all install clean
```

Sonra PostgreSQL9.4 verilənlər bazasını portlardan yükleyirik:

```
root@dolibarr:~ # cd /usr/ports/databases/postgresql94-server
root@dolibarr:/usr/ports/databases/postgresql94-server # make all install
clean
```

Yüklədikdən sonra bazanın inisializasiyasını edirik. Öncə PostgreSQL-i startup-a əlavə edirik ki, inisializasiya edə bilək. Inisializasiyadan sonra işə salırıq:

```
root@dolibarr:~ # echo 'postgresql_enable="YES"' >> /etc/rc.conf
root@dolibarr:~ # /usr/local/etc/rc.d/postgresql initdb
root@dolibarr:~ # /usr/local/etc/rc.d/postgresql start
```

Bundan sonra PHP5.6-nı portlardan yükləyirik(IPv6-dan başqa qalan hər şey susmaya görə qalır):

```
root@dolibarr:~ # cd /usr/ports/lang/php56
root@dolibarr:/usr/ports/lang/php56 # make -DBATCH all install clean
```

PHP5.6-nın genişlənmələrini portlardan yükləyirik:
root@dolibarr:~ # **cd /usr/ports/lang/php56-extensions**
root@dolibarr:/usr/ports/lang/php56-extensions # **make config**

Açılan dialoq pəncərəsində bu modulları seçirik: **BCMATH BZ2 CALENDAR CTYPE CURL DOM FILTER GD HASH ICONV JSON MBSTRING MCRYPT PGSQL**
root@dolibarr:/usr/ports/lang/php56-extensions # **make -DBATCH all install**

Həmçinin portlardan apache-in modulunu yükləyirik(IPv6-dan başqa hər şey susmaya gore qalır):
root@dolibarr:~ # **cd /usr/ports/www/mod_php56**
root@dolibarr:/usr/ports/www/mod_php56 # **make all install clean**

Yükləmələrimizdən sonra quraşdırmalara başlayaq. Php üçün ini faylını nusxeleyek və teleb edilən huquqları verək.
root@dolibarr:~ # **cd /usr/local/etc/**
root@dolibarr:/usr/local/etc # **cp php.ini-production php.ini**
root@dolibarr:/usr/local/etc # **chmod u+w php.ini**

Ardınca php genişlənmələrin apache-imizdə tanına bilməsi üçün **/usr/local/etc/apache24/Includes** qovluğunda fayl yaradaq.
root@dolibarr:~ # **cd /usr/local/etc/apache24/Includes**
root@dolibarr:/usr/local/etc/apache24/Includes # **touch php-application.conf**

Yaratdıqımız **/usr/local/etc/apache24/Includes/php-application.conf** faylının tərkibinə aşaqıdakı sətirləri əlavə edək:
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phpsXsource

/etc/hosts faylına aşaqıdakı sətirləri əlavə edirik ki, apache işə düşdükdə heç bir səhv çap eləməsin:
127.0.0.1 localhost localhost.my.domain
XX.XX.XX.XX dolibarr.opensource.az dolibarr

VirtualHost-ların işləməsi üçün apache-in **httpd.conf** faylina **Include** əlavə edirik:
root@dolibarr:~ # **echo "Include /usr/local/domen/*" >> /usr/local/etc/apache24/httpd.conf**

VirtualHost qovluqu yaradırıq:
root@dolibarr:~ # **mkdir /usr/local/domen/**

/usr/local/domen/dolibarr.opensource.az faylının tərkibinə aşaqidakı sətirləri əlavə edirik:

```
<VirtualHost *>
        ServerAdmin webmaster@email.com
        ServerName dolibarr.opensource.az
        CustomLog "/var/log/dolibarr_access.log" common
        ErrorLog /var/log/dolibarr_error.log
        DocumentRoot /usr/local/www/dolibarr/htdocs
<Directory "/usr/local/www/dolibarr/htdocs">
        AllowOverride All
        Require all granted
```

```
</Directory>
</VirtualHost>
```

/usr/local/etc/apache24/httpd.conf faylında **DirectoryIndex** sətirinin qarşısına **index.php** əlavə edirik:

```
DirectoryIndex index.php index.html
```

Dolibarr-i endirdiyimiz qovluğu **/usr/local/www** ünvanına **dolibarr** adi ilə köçürürük:

```
root@dolibarr:~ # mv /root/dolibarr-develop /usr/local/www/dolibarr
```

Indi **/usr/local/www/dolibarr/documents** adı ilə qovluq yaradaq. Apache üçün yazilma və huquqları təyin edək:

```
root@dolibarr:~ # mkdir /usr/local/www/dolibarr/documents
root@dolibarr:~ # chown -R www:www /usr/local/www/dolibarr/documents
root@dolibarr:~ # chmod 777 /usr/local/www/dolibarr/documents
```

Eyniylə növbəti qovluqlari yaradib hər kəs tərəfindən yazila bilən edirik:

```
root@dolibarr:~ # mkdir /usr/local/www/dolibarr/documents/doctemplates
root@dolibarr:~ # mkdir /usr/local/www/dolibarr/documents/propale
root@dolibarr:~ # mkdir /usr/local/www/dolibarr/documents/ficheinter
root@dolibarr:~ # mkdir /usr/local/www/dolibarr/documents/facture
root@dolibarr:~ # chmod 777 /usr/local/www/dolibarr/documents/doctemplates
root@dolibarr:~ # chmod 777 /usr/local/www/dolibarr/documents/propale
root@dolibarr:~ # chmod 777 /usr/local/www/dolibarr/documents/ficheinter
root@dolibarr:~ # chmod 777 /usr/local/www/dolibarr/documents/facture
```

Həmçinin apache-ın qovluğuna yetki veririk:

```
root@dolibarr:~ # chown -R www:www /usr/local/www/dolibarr/htdocs
```

Tələb edilən faylları nüsxələyirik:

```
root@dolibarr:~ # cp -R /usr/local/www/dolibarr/htdocs/install/doctemplates/*
/usr/local/www/dolibarr/documents/doctemplates/
```

Sonra apache servisi StartUP-a əlavə edib, işe saliriq:

```
root@dolibarr:~ # echo 'apache24_enable="YES"' >> /etc/rc.conf
root@dolibarr:~ # /usr/local/etc/rc.d/apache24 start
```

Sonra **/usr/local/pgsql/data/pg_hba.conf** faylinda olan **host all all 127.0.0.1/32 trust** sətirini dəyişib aşağıdakı şəklə gətiririk:

```
host    all             all             127.0.0.1/32            md5
```

Həmçinin **/usr/local/pgsql/data/postgresql.conf** faylında aşağıdakı sətrin qarşısından şərhi silmək lazımdır:

```
isten_addresses = 'localhost'
```

PostgreSQL-i yenidən işə salırıq ki, dəyişiklikləri götürsün:

```
root@dolibarr:~ # /usr/local/etc/rc.d/postgresql restart
```

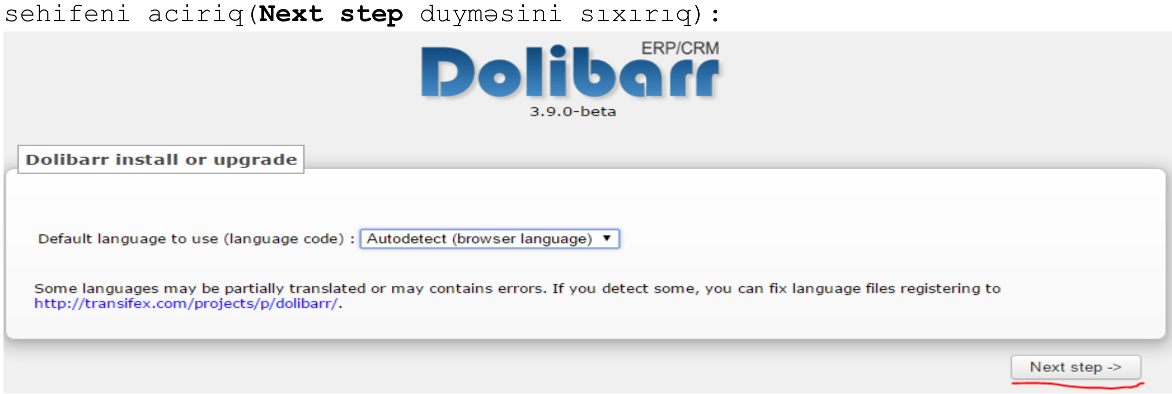
PostgreSQL istifadəçisi üçün şifrə təyin edirik:

```
root@dolibarr:~ # passwd pgsql
Changing local password for pgsql
```

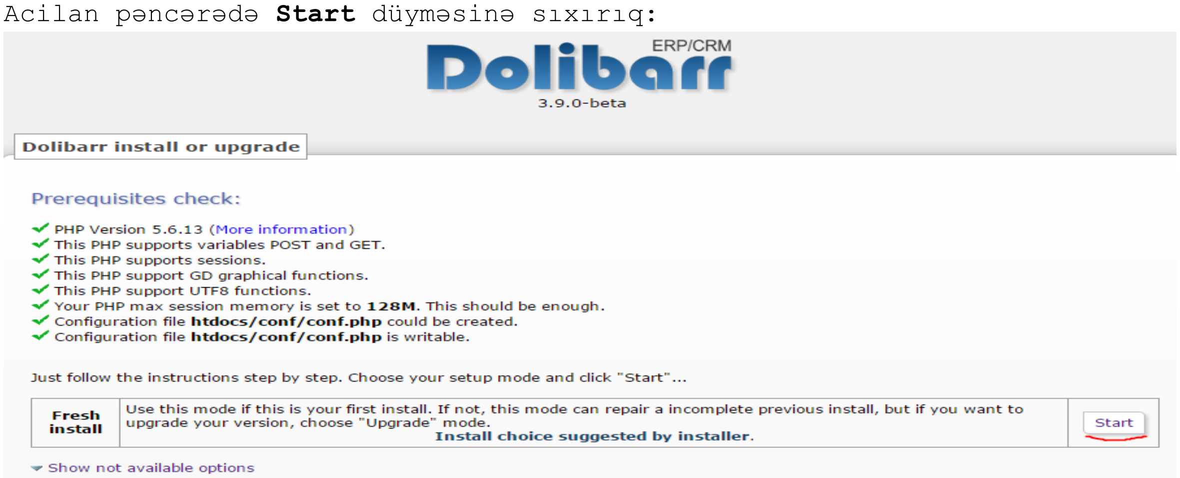
New Password: **şifrə**
Retype New Password: **şifrə_təkrar**

PostgreSQL istifadəçi adından daxil olub, **dolibarr** adlı istifadəçi və verilənlər bazası yaradırıq:
root@dolibarr:~ # **passwd pgsql**
Changing local password for pgsql
New Password:
Retype New Password:
root@dolibarr:~ # **su pgsql**
$ **createuser -sdrP dolibarr**
Enter password for new role: **db_şifrə**
Enter it again: **db_şifrə_tekrar**
$ **createdb dolibarr --owner=dolibarr**
$ **exit**

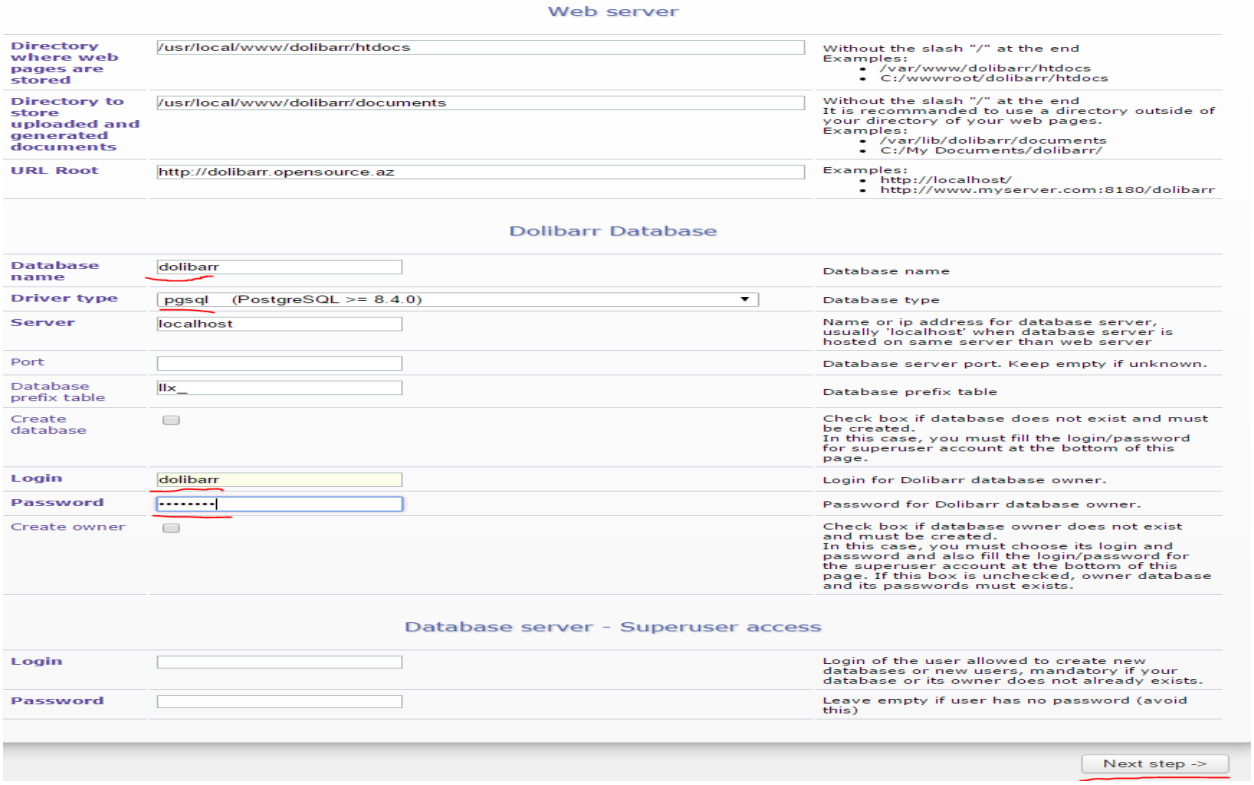
Artiq server hazırdır. Istənilən desktop maşının browserində http://dolibarr.opensource.az/install/ linkini daxil edirik və şekildəki sehifeni aciriq(**Next step** duyməsini sıxırıq):



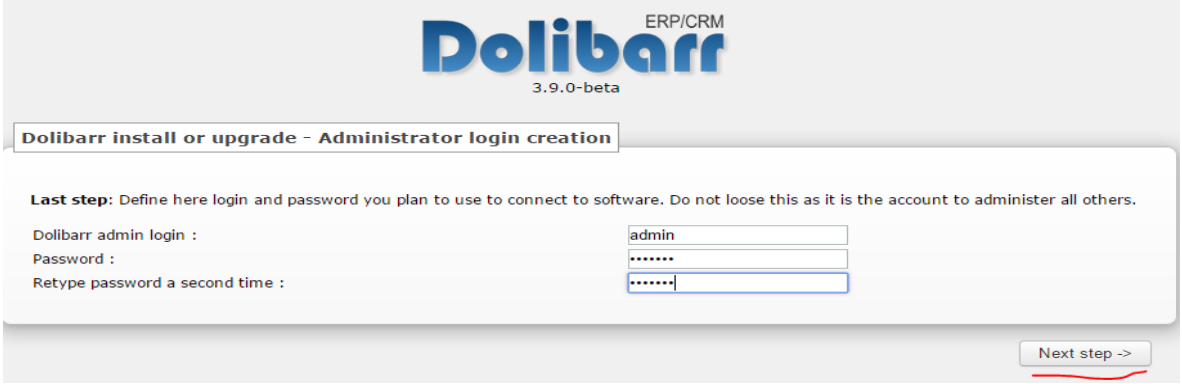
Acilan pəncərədə **Start** düyməsinə sıxırıq:



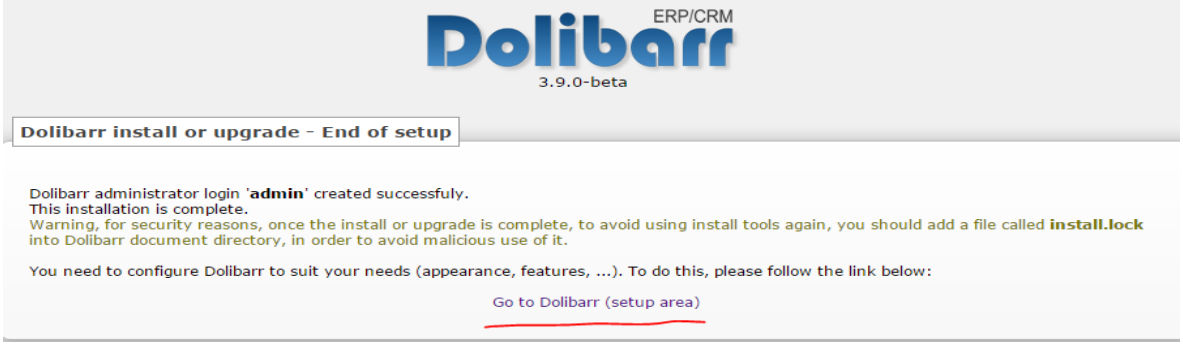
Açılan pəncərədə verilənlər bazası üçün istifadəçi adi şifrə və verilənlər bazasının adını daxil edirik və **Next step** düymesini sıxırıq:



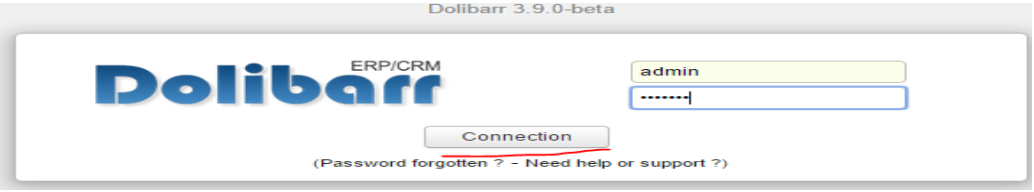
Dolibarr admin paneli üçün istifadəçi adı və şifrə təyin edib, **Next** step düyməsini sıxırıq:



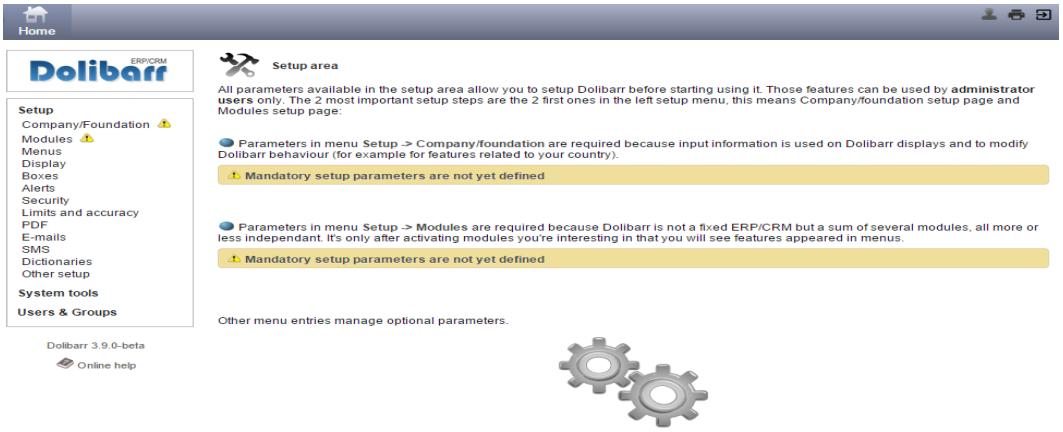
Nəticə aşağıda şəkildəki kimi olacaq. **Go to Dolibarr(setup area)** düyməsinə sıxırıq:



Istifadəçi adı və şifrəsini təyin edib, şəkildəki kimi **Connection** düyməsinə sixiriq:



Əgər aşağıdakı səhifəni görürüksə artıq hazırdır:



Ubuntu 14.04 üzərində OpenERP oDoo-nun qurulması

Odoo(Həmişəki TinyERP, OpenERP) – Belçika şirkəti OpenERP tərəfindən yaradılmış açıq kodlu ERP ve CRM sistemdir. xml-rpc protocol üsulu ilə işləyən Python proqram dilində yazılmış client-server tipli proqram təminatıdır. Server tərəf üçün PostgreSQL verilənlər bazası istifadə edilir.

Sistemdə Realizasiya edilmiş modullardan – mühasibatlıq, CRM, şəxsiyyətin idarəedilməsi, istehsal, satış, alış, anbarın idarəedilməsi, proyektlərin idarəedilməsi, nəqliyyatın idarəedilməsi, prezentasıyaların idarəedilməsi, POS və social şəbəkələrlə inteqrasiya edilə bilən modulu var.

Bu məqalədə biz Odoo sevrerin Ubuntu 14.04 server əməliyyat sistemi üzərində yüklənməsinə baxacağıq.

Bütün işləri görməzdən öncə nəzərdə tutulur ki, serverdə artıq şəbəkə qurulmuşdur və internet mövcuddur.

Ilk işimiz odoo istifadəçisini sistemə elavə edirik(Bütün işlər sudo istifadəçisi vasitəsilə görülür):

```
sysuser@redmine:~$ sudo adduser --system --home=/opt/odoo --group odoo
sysuser@redmine:~$ sudo su - odoo -s /bin/bash
odoo@redmine:~$ exit
```

PostgreSQL verilənlər bazasını yükləyirik:

```
sysuser@redmine:~$ sudo apt-get install postgresql
```

PostgreSQL quraşdırmasında dəyişiklik edirik:

```
sysuser@redmine:~$ sudo nano /etc/postgresql/9.3/main/postgresql.conf
```

Bu **#listen_addresses = 'localhost'** sətiri tapırıq və qarşısından şərhi silirik:

```
      listen_addresses = 'localhost'
```

Indi konsol-a postgres istifadəçi adı ilə daxil oluruq və orda **openerp** adlı istifadəçi yaradırıq(Eynilə **odoo** adlı DB yaradıb üzvünü **openerp** təyin edirik):

```
sysuser@redmine:~$ sudo su - postgres
postgres@redmine:~$ createuser -sdrP openerp
Enter password for new role: şifre
Enter it again: şifre_tekrar
postgres@redmine:~$ createdb odoo —owner=openerp
postgres@redmine:~$ exit
```

Artıq Python-a tələb edilən komponentlər və GIT-i yükləyirik:

```
sysuser@redmine:~$ sudo apt-get install python-cups python-dateutil python-
decorator python-docutils python-feedparser python-gdata python-geoip python-
gevent python-imaging python-jinja2 python-ldap python-libxslt1 python-lxml
python-mako python-mock python-openid python-passlib python-psutil python-
psycopg2 python-pybabel python-pychart python-pydot python-pyparsing python-
pypdf python-reportlab python-requests python-simplejson python-tz python-
```

```
unicodecsv python-unittest2 python-vatnumber python-vobject python-werkzeug
python-xlwt python-yaml wkhtmltopdf

sysuser@redmine:~$ sudo apt-get install git
```

Artıq **odoo** istifadəçi adı ilə daxil olub, Odoo-nu yükləyəcəyik:

```
sysuser@redmine:~$ sudo su - odoo -s /bin/bash
odoo@redmine:~$ git clone https://www.github.com/odoo/odoo --depth 1 --branch
8.0 --single-branch
odoo@redmine:~$ exit
```

Quraşdırma faylı yaradaq və ona odoo istifadəçi hüququnu vərək:

```
sysuser@redmine:~$ sudo touch /etc/odoo-server.conf
sysuser@redmine:~$ sudo chown odoo: /etc/odoo-server.conf
sysuser@redmine:~$ sudo chmod 640 /etc/odoo-server.conf
```

Faylı açırıq:

```
sysuser@redmine:~$ sudo nano /etc/odoo-server.conf
```

Verilənlərin tərkibinə aşağıdakı sətirləri əlavə edirik:

```
[options]
; Bu şifrə verilənlər bazası üzərində əməliyyatlar aparmaqa icazə verir:
; admin_passwd = admin
db_host = localhost
db_port = 5432
db_user = openerp
db_password = rumburak
addons_path = /opt/odoo/odoo/addons
logfile = /var/log/odoo/odoo-server.log
```

Bundan sonra **odoo** istifadəçi adı ilə daxil oluruq:

```
sysuser@redmine:~$ sudo su - odoo -s /bin/bash
```

WEB serverin işləməsini yoxlayırıq:

```
odoo@redmine:~$ /opt/odoo/odoo/openerp-server
2015-09-30 02:54:38,347 9784 INFO ? openerp: OpenERP version 8.0
2015-09-30 02:54:38,347 9784 INFO ? openerp: addons paths:
['/opt/odoo/.local/share/Odoo/addons/8.0', u'/opt/odoo/odoo/openerp/addons',
u'/opt/odoo/odoo/addons']
2015-09-30 02:54:38,347 9784 INFO ? openerp: database hostname: localhost
2015-09-30 02:54:38,347 9784 INFO ? openerp: database port: 5432
2015-09-30 02:54:38,347 9784 INFO ? openerp: database user: odoo
2015-09-30 02:54:38,699 9784 INFO ? openerp.service.server: HTTP service
(werkzeug) running on 0.0.0.0:8069
```

Əgər console-da uzun müddət dayanarsa **Ctrl+C** əmri ilə durdura bilərsiniz.

Konsol-dan çıxırıq:

```
odoo@redmine:~$ exit
```

Artıq **/etc/init.d/odoo-server** işə salma skriptini yaradaq:
sysuser@redmine:~$ **sudo touch /etc/init.d/odoo-server**

Faylı açırıq:
sysuser@redmine:~$ **sudo nano /etc/init.d/odoo-server**

Tərkibinə aşağıdakı sətrləri əlavə edirik:

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:             odoo-server
# Required-Start:       $remote_fs $syslog
# Required-Stop:        $remote_fs $syslog
# Should-Start:         $network
# Should-Stop:          $network
# Default-Start:        2 3 4 5
# Default-Stop:         0 1 6
# Short-Description:    Complete Business Application software
# Description:          Odoo is a complete suite of business tools.
### END INIT INFO

PATH=/bin:/sbin:/usr/bin
DAEMON=/opt/odoo/odoo/openerp-server
NAME=odoo-server
DESC=odoo-server

# Specify the user name (Default: odoo).
USER=odoo

# Specify an alternate config file (Default: /etc/odoo-server.conf).
CONFIGFILE="/etc/odoo-server.conf"

# pidfile
PIDFILE=/var/run/$NAME.pid

# Additional options that are passed to the Daemon.
DAEMON_OPTS="-c $CONFIGFILE"

[ -x $DAEMON ] || exit 0
[ -f $CONFIGFILE ] || exit 0

checkpid() {
    [ -f $PIDFILE ] || return 1
    pid=`cat $PIDFILE`
    [ -d /proc/$pid ] && return 0
    return 1
}

case "${1}" in
        start)
                echo -n "Starting ${DESC}: "

                start-stop-daemon --start --quiet --pidfile ${PIDFILE} \
                        --chuid ${USER} --background --make-pidfile \
                        --exec ${DAEMON} -- ${DAEMON_OPTS}
```

```
                echo "${NAME}."
                ;;

        stop)
                echo -n "Stopping ${DESC}: "

                start-stop-daemon --stop --quiet --pidfile ${PIDFILE} \
                        --oknodo

                echo "${NAME}."
                ;;

        restart|force-reload)
                echo -n "Restarting ${DESC}: "

                start-stop-daemon --stop --quiet --pidfile ${PIDFILE} \
                        --oknodo

                sleep 1

                start-stop-daemon --start --quiet --pidfile ${PIDFILE} \
                        --chuid ${USER} --background --make-pidfile \
                        --exec ${DAEMON} -- ${DAEMON_OPTS}

                echo "${NAME}."
                ;;

        *)
                N=/etc/init.d/${NAME}
                echo "Usage: ${NAME} {start|stop|restart|force-reload}" >&2
                exit 1
                ;;
esac

exit 0
```

root istifadəçisi üçün fayla hüquq veririk və faylı yerinə yetirilən edirik:

```
sysuser@redmine:~$ sudo chown root: /etc/init.d/odoo-server
sysuser@redmine:~$ sudo chmod 755 /etc/init.d/odoo-server
```

Sonra jurnallar üçün qovluq yaradırıq:

```
sysuser@redmine:~$ sudo mkdir /var/log/odoo
sysuser@redmine:~$ sudo chown odoo:root /var/log/odoo
```

Serverimizi yenidən yükləyirik:

```
sysuser@redmine:~$ sudo shutdown -r now
```

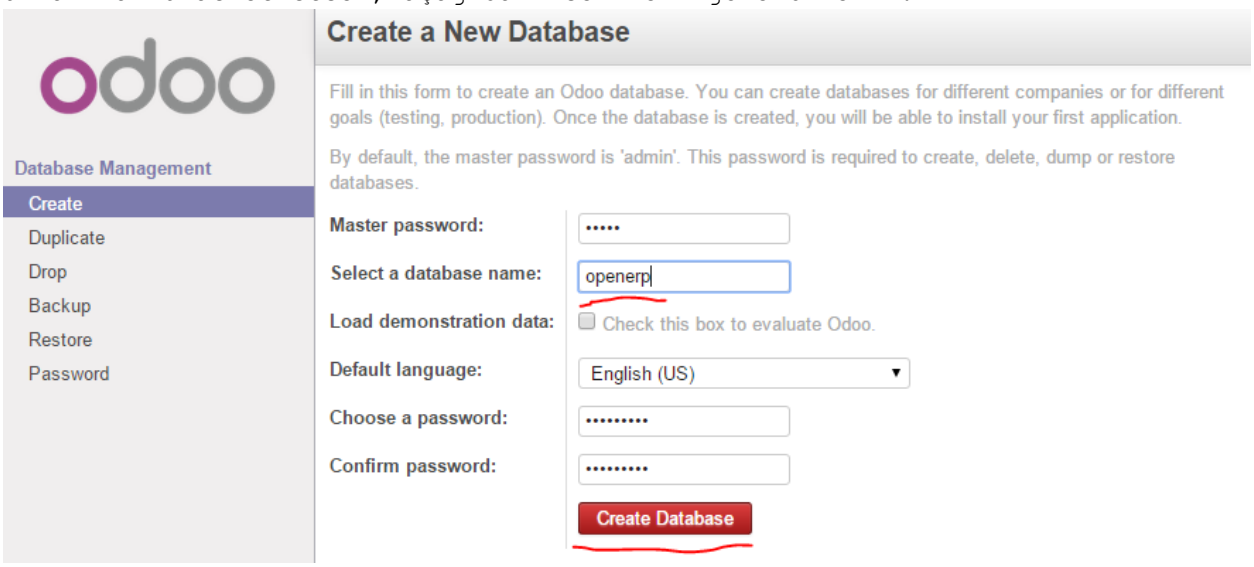
Sistem qalxdiqdan sonra odoo-server servisini əlimizlə işə salırıq:

```
sysuser@redmine:~$ sudo service odoo-server start
Starting odoo-server: odoo-server.
```

Servisi StartUP-a əlavə etmək üçün aşağıdakı əmri işə salmaq lazımdır(artıq sistemi reboot etsəniz odoo servisi avtomatik işə düşəcək):

```
sysuser@redmine:~$ sudo update-rc.d odoo-server defaults
Adding system startup for /etc/init.d/odoo-server ...
   /etc/rc0.d/K20odoo-server -> ../init.d/odoo-server
   /etc/rc1.d/K20odoo-server -> ../init.d/odoo-server
   /etc/rc6.d/K20odoo-server -> ../init.d/odoo-server
   /etc/rc2.d/S20odoo-server -> ../init.d/odoo-server
   /etc/rc3.d/S20odoo-server -> ../init.d/odoo-server
   /etc/rc4.d/S20odoo-server -> ../init.d/odoo-server
   /etc/rc5.d/S20odoo-server -> ../init.d/odoo-server
```

Yükləməni bitirdik və artıq istənilən Desktop maşından http://server_IP:8069 ünvanına müraciət etsək, aşağıdakı səhifəni görə bilərik:



Bu səhifədə cıxan formanı aşağıda açıqlayırıq.

Master Password – Susmaya görə daxil olma səhifəsində olan E-Mail istifadəçi hesabının adı **admin** və şifrəsi **admin** olur. Burda həmin istifadəçinin **admin** şifrəsi yazılır.

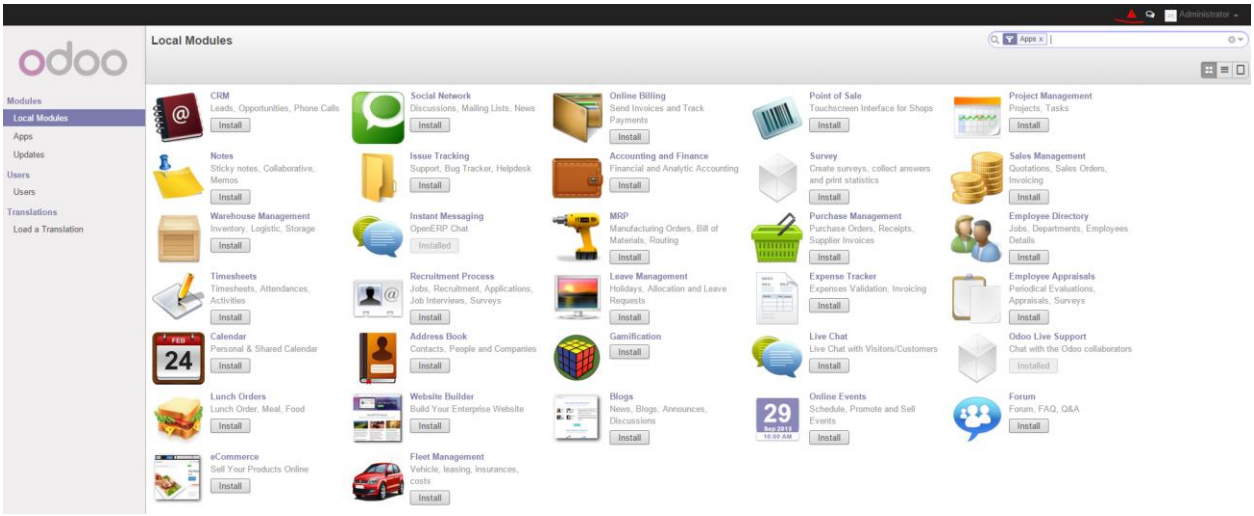
Select a database name – Odoo özü üçün biraz öncə yaratdığımız PostgreSQL istifadəçi adını **/etc/odoo-server.conf** faylından oxuyub, tamam fərqli adlı bir baza yaradacaq bu sütünda həmin bazanın adı yazılır(**openerp** adlı baza yaradılmasını deyirik). Yaradılan verilənlər bazası üçün **create**, **delete**, **dump** ya da **restore** etmək hüququ olmalıdır.

Default language – WEB səhifənın susmaya görə olan dilini seçirik(**English(US)**).

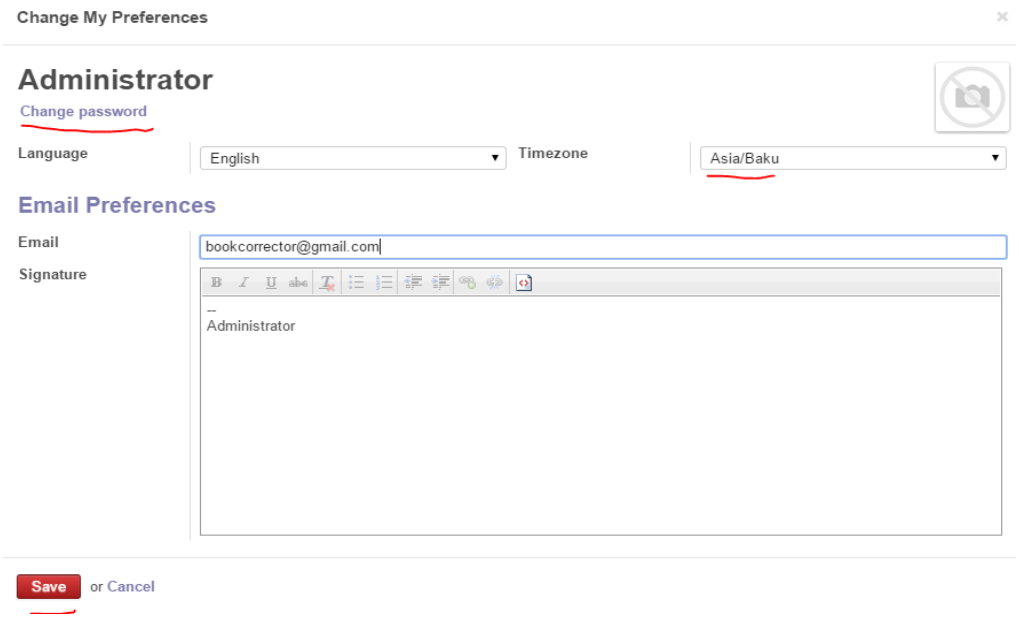
Choose a password – **admin** istifadəçi hesabı üçün yeni şifrə

Confirm password – **admin** istifadəçi hesabı üçün yeni şifrə təkrar

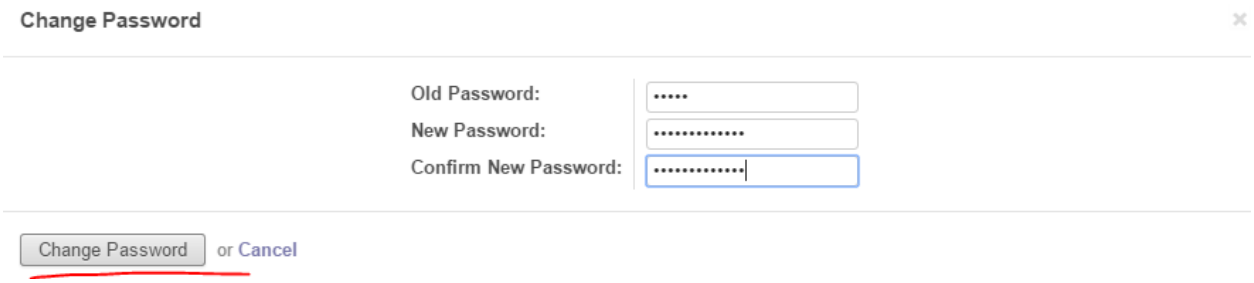
Nəticədə aşağıdakı səhifəni əldə etmiş olacağıq:



Nida simvolu time zone-un səhv olmasını deyir və onu düzəltmək üçün həmin düyməyə sıxmaq lazımdır. Daxil olub dəyişiklikləri edirik və **Save** düyməsinə sıxırıq.

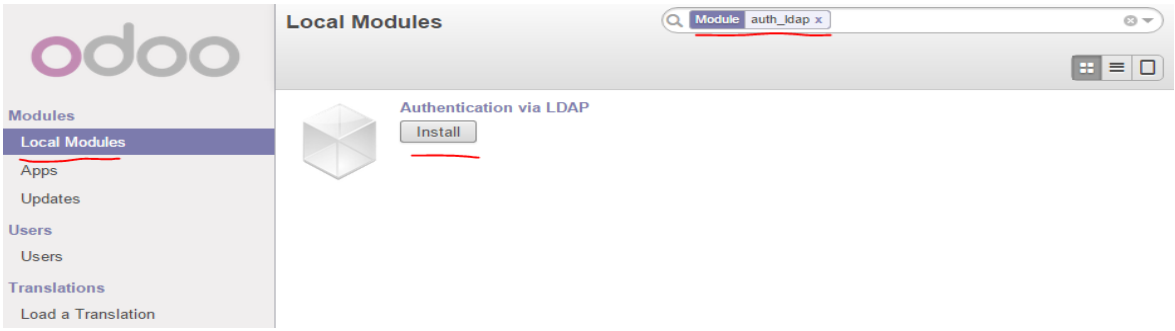


Sonra sağ tərəfdə olan **Administrator** -> **Preferences** -> **Change Password** və şəkildəki kimi köhnə şifrə və iki dəfə yeni şifrəni daxil edirik:

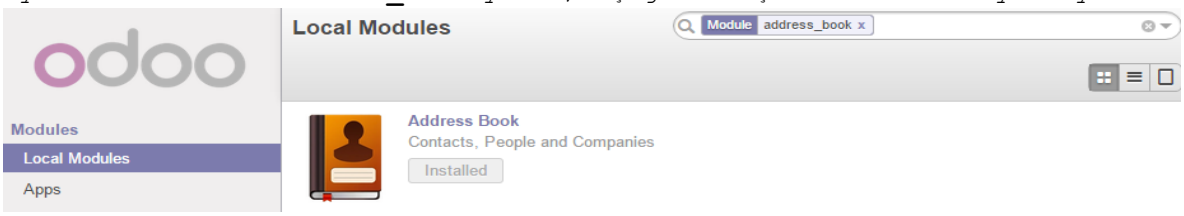


oDoo ERP sisteminizin müəsisenizin Active Directory-si ilə inteqrasiya etmək istəsəniz, öncə **python-ldap** paketini resposlardan yükləmək və sonra oDoo web interfeysdən **user_ldap** modulunu yükləmək lazımdır:
sysuser@redmine:~$ **sudo apt-get install python-ldap**

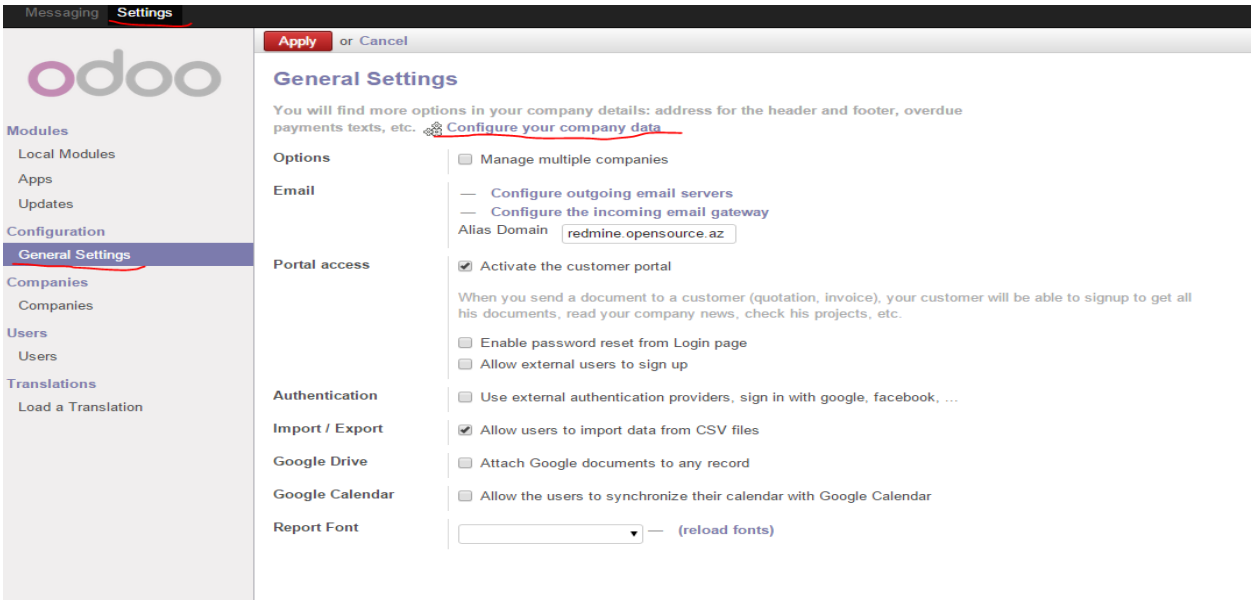
Sonra gedirik **Modules** -> **Local Modules** və əgər **Search** olan xanada **Apps** varsa onu silib, **auth_ldap** axtarırıq. Açılan pəncərədə şəkildəki kimi, Install düyməsinə sıxırıq:



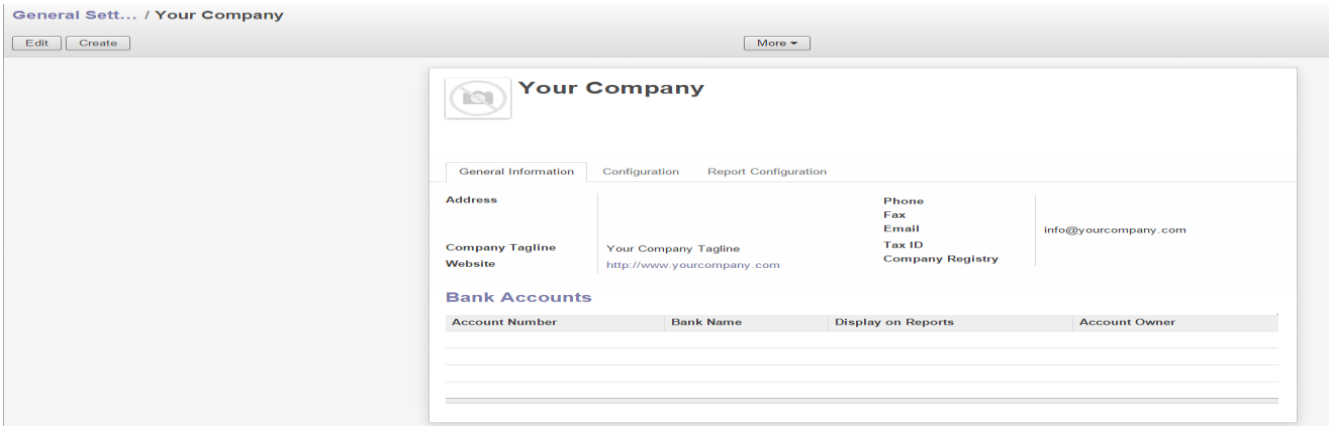
Eynilə Search-də **address_book** yazıb, aşağıdakı şəkildəki kimi yükləyirik:



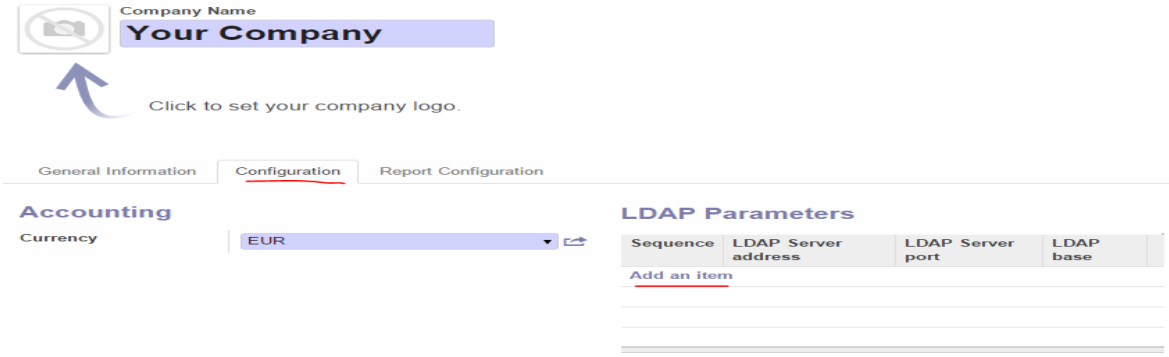
Sonra panelin yuxarısında **Settings** -> **General Settings** -> **Configure your company data**



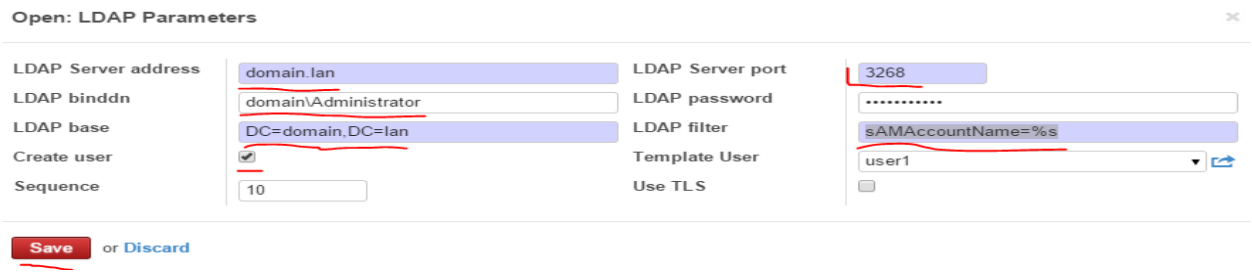
Açılan pəncərədə **Edit** düyməsinə sıxırıq:



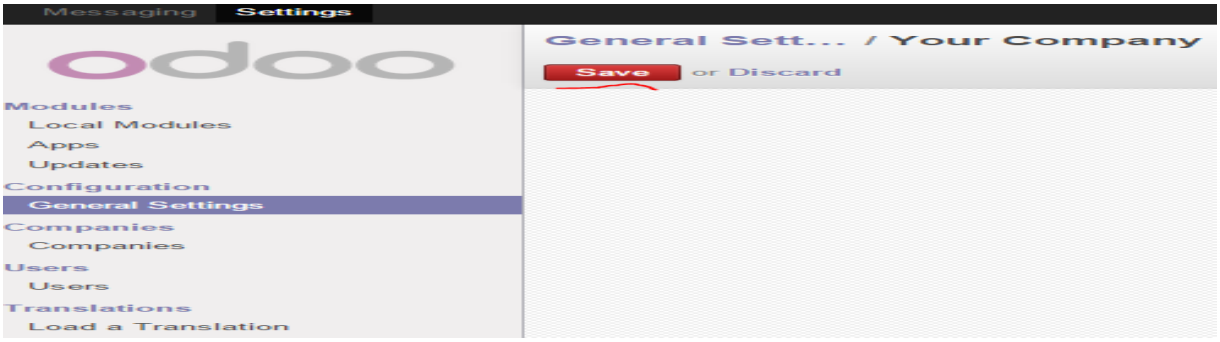
Sonra **Configuration**-a daxil oluruq və **LDAP Parameters** altında **Add a item** düyməsinə sıxırq:



Sonra aşağıdakı parametrləri şəkildəki kimi öz DC-mizə uyğun olaraq yazırıq:
LDAP Server address: **domain.lan**
LDAP Server port: **3268**
LDAP binddn: **domain\Administrator**
LDAP password: **DC_nin_admin_Şifrəsi**
LDAP base: **DC=domain,DC=lan**
LDAP filter: **sAMAccountName=%s**
Create user: **Yes**
Template User: **Oz yaratdiqiniz hansisa şablonu secin**



Sonra ümumi səhifədə də **Save** düyməsinə sıxırıq ki, dəyişiklik yadda qalsın:



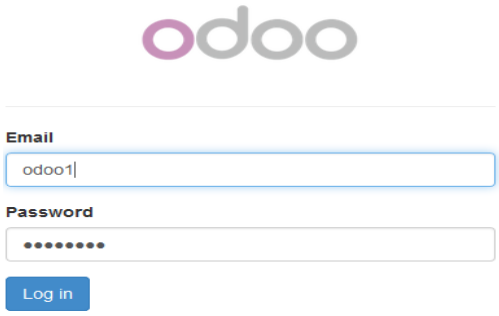
oDoo serverdə LDAP alətlərindən istifadə müəyyən sınaqları edə bilərsiniz. Bu paket vasitəsilə serverimizin LDAP-a uğurlu qoşulmasını və qrupun axtarışını sınaqdan keçirə bilərik.

```
root@redmine:~# apt-get install ldap-utils
```

Aşağıdakı əmrlə DC-nizdə olan bütün domain strukturunu darta bilərsiniz:

```
root@redmine:~# ldapsearch -x -b  "dc=domain,dc=lan" -H ldap://domain.lan/ -D "DOMAIN\Administrator" -w A123456789a
```

Artıq http://server_İP_Address:8069/ unvanına daxil olduqdan sonra aşağıdakı səhifəyə DC istifadəçi adı və şifrə ilə daxil ola bilərsiniz. Misal üçün bizim halda **odoo1** adlı istifadəçi artıq DC-mizdə yaradılmışdır:



BÖLÜM 4

Wireless şəbəkəsində olan tələblərin qarşılanması

- **FreeBSD 10.1 üzərində Freeradiusun portlardan yüklənməsi və LDAP-la inteqrasiyası**
- **FreeBSD 10.1-də FreeRadiusun NTLM-MSCHAP vasitəsi ilə AD ilə inteqrasiyası**
- **CentOS üzərində DaloRadius qurulması**
- **FreeBSD FreeRADİUS EAP-TLS**
- **FreeBSD 10.1 x64 WiFi Hotspot**

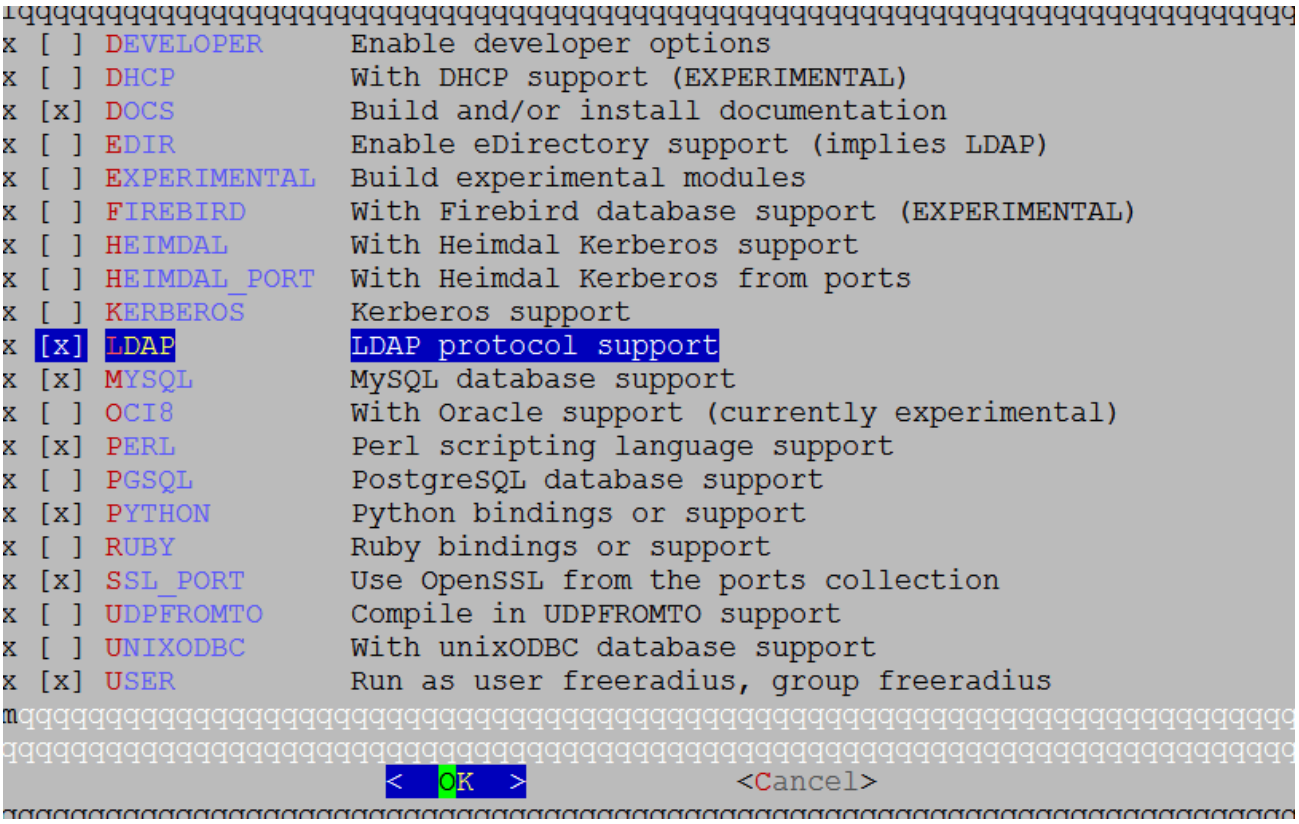
Müasir zamanda hər şirkətin Wireless şəbəkəsi mövcud olur. Lakin wireless şəbəkəsinin təhlükəsizliyi adi ethernet-dən daha təhlükəli olur. Həmçinin istifadəçi qeydiyyatıda şirkətin eyni bazasına baxılması tələbi yaranır ki, hər kəs öz Domain Controller istifadəçi adı və şifrəsindən istifadə eləsin. Ya da digər tələb ola bilər ki, qonaqlar üçün müvəqqəti bir istfiadəçi adı və şifrə olmalıdır. Bu başlıqda istifadəçilərin wireless şəbəkəsinə sertifikatla, active reictory istifadəçi adı və şifrə ilə, qonaqlar üçün ayrılmış istifadəçi adları ilə qoşulma üsullarının qurulması açıqlanır.

FreeBSD 10.1 üzərində Freeradiusun portlardan yüklənməsi və LDAP-la inteqrasiyası

Məqsədimiz FreeRADİUS-a LDAP-dan istifadəçi bazasını ötürməkdir.

 # cd /usr/ports/net/freeradius2 => Port ünvanına daxil oluruq

 # make config => LDAP protokolunu seçib OK düyməsini sıxırıq



 # make install clean -DBATCH **=>** Freeradius2-ni portlardan yükləyirik

 # rehash => bu əmri daxil edib binar fayllarının bazasını yeniləyirik

/etc/rc.conf startup faylına aşağıdakı sətiri əlavə edirik ki, daemonumuz yenidən yüklənmədən sonra avtomatik işə düşsün:
radiusd_enable="YES"

/usr/local/etc/rc.d/radiusd start - Daemonu işə salırıq

Freeradius-u əməliyyat sisteminə yüklədikdən sonra LDAP-la inteqrasiya edək

 # cd /usr/local/etc/raddb/ => Freeradiusun quraşdırma fayllarının yerləşdiyi qovluğa daxil oluruq

 # ee /usr/local/etc/raddb/modules/ldap => LDAP modulunu quraşdırırıq

"**ldap {**" bölməsinin altında aşağıdakı kimi LDAP serverimizə uyğun dəyişiklikləri edirik.

```
ldap {
        # LDAP serverinin İP ünvanı ya da adı
        server = "kofe.az"

        # "identity" qarşısına Ldap serverindən istifadəçilərini oxumaq üçün
izin
        # verilmiş hər hansı bir istifadəçinin LDAP serverdəki ünvanını
yazırıq
        identity = "CN=Administrator,CN=Users,DC=kofe,DC=az"

        # Həmin istifadəçinin şifrəsini qeyd edirik
        password = Zxcasdqwe123

        # basedn bölməsində isə Domainimizin LDAP serverindəki ünvanını
yazırıq
        basedn = "DC=kofe,DC=az"

        # Aşağıdakıları da burada olduğu kimi eynilə qeyd edin
        filter = "(sAMAccountName=%{Stripped-User-Name:-%{User-Name}})"
        base_filter = "(objectclass=radiusprofile)"
        chase_referrals = yes
        rebind = yes
        ldap_connections_number = 5
        max_uses = 0
        port = 389
        timeout = 4
        timelimit = 3
        net_timeout = 1
        tls {
                start_tls = no
        }

        keepalive {
                idle = 60
                probes = 3
                interval = 3
        }
}
```

 # ee /usr/local/etc/raddb/sites-available/default => quraşdırma faylına daxil oluruq

Faylda aşağıda göstərilən bölmələrdə göstərilən sətrlərin qarşısındakı şərhləri silirik.

```
authorize {
      ...
        #
        ldap
        #
        #
      ...
```

```
}

authenticate {
      ...
      Auth-Type LDAP {
                  ldap
          }
      ...
}
```

 # ee /usr/local/etc/raddb/clients.conf => Radiusa qoşulmaq üçün klientlərə izni buradan veririk

Freeradius susmaya görə localhost-u klient kimi özünə qoşulmağa izin verir. Bunu **clients.conf** faylının içində görə bilərsiniz.

```
client localhost {
      ipaddr = 127.0.0.1
      secret          = testing123
      require_message_authenticator = no
      nastype     = other     # localhost isn't usually a NAS...
}
```

Buna görə də elə freeradius quraşdırdığımız maşından Radiusun Ldap-la inteqrasiyasını test edə bilərik.

Bir konsolda
 # service radiusd stop => FreeRadius serveri əgər işlək vəziyyətdədirsə dayandırırıq.
 # radiusd -fX => əmrini yığıb gözləyirik. Bu əmr FreeRadius-u debug etmək üçün bizə kömək edir. Əmri daxil etdikdən sonra əgər heç bir səhv çıxartmadan aşağıdakı kimi nəticə göstərirsə demək ki, quraşdırma faylların sintaksisində heç bir problem yoxdur.

```
.....
Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on command file /var/run/radiusd/radiusd.sock
Listening on authentication address 127.0.0.1 port 18120 as server inner-
tunnel
Listening on proxy address * port 1814
Ready to process requests.
```

FreeRadiusun Ldap-la düzgün inteqrasiyasını isə ikinci konsolda **"radtest** *istifadeciadi "şifrə" freeradius-server-ip* **10** *pre-shared-secret-key"* Yəni bizim vəziyyətimizdə aşağıdakı kimi əmri daxil edirik.
 # radtest camal "Zxcasdqwe123" 127.0.0.1 10 testing123
Nəticədə əgər aşağıdakı kimi **Access-Accept** gördüksə bu o deməkdir ki, hər şey işləyir.

```
Sending Access-Request of id 137 to 127.0.0.1 port 1812
        User-Name = "camal"
        User-Password = "Zxcasdqwe123"
        NAS-IP-Address = 127.0.53.53
```

```
        NAS-Port = 10
        Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=137,
length=20
```

 # radiusd -fX əmrini yazdığımız konsolda isə nəticə aşağıdakı kimi olmalıdır.

```
.......
Found Auth-Type = LDAP
# Executing group from file /usr/local/etc/raddb/sites-enabled/default
+group LDAP {
[ldap] login attempt by "camal" with password "Zxcasdqwe123"
[ldap] user DN: CN=camal shahverdiyev,OU=test,DC=kofe,DC=az
  [ldap] (re)connect to kofe.az:389, authentication 1
  [ldap] bind as CN=camal shahverdiyev,OU=test,DC=kofe,DC=az/Zxcasdqwe123 to
kofe.az:389
  [ldap] waiting for bind result ...
  [ldap] Bind was successful
[ldap] user camal authenticated succesfully
++[ldap] = ok
+} # group LDAP = ok
# Executing section post-auth from file /usr/local/etc/raddb/sites-
enabled/default
+group post-auth {
++[exec] = noop
+} # group post-auth = noop
Sending Access-Accept of id 243 to 127.0.0.1 port 41919
Finişəd request 3.
Going to the next request
Waking up in 4.9 seconds.
Cleaning up request 3 ID 243 with timestamp +753
Ready to process requests.
```

Hər şeyin işlədiyinə əmin olduqdan sonra FreeRADİUS-u əməliyyat sisteminin yenidən yüklənməsindən sonra işə düşməsi üçün **/etc/rc.conf** faylına aşağıdakı sətirləri əlavə edirik.

 # echo 'radiusd_enable="YES"' >> /etc/rc.conf

 # /usr/local/etc/rc.d/radiusd start **=>** FreeRadiusu işə salırıq

Freebsd 10.1-də FreeRadiusun NTLM-MSCHAP vasitəsi ilə AD ilə inteqrasiyası

Windows 2008 R2-də AD və DNS servis qaldırmalıyıq. Bu sənəddə biz test olaraq qaldırdığımız serverin və Freeradius FreeBSD maşının məlumatları aşağıdakı kimidir.

AD (Active Directory) və DNS: **VELO.LAN**
AD hostname: **DC.VELO.LAN**
AD İp address: **10.0.0.10**

Freeradius İP address **10.0.0.1**
Freeradius hostname: **FREERADIUS.VELO.LAN**

 # hostname FREERADIUS.VELO.LAN => Hostname-i domain adına uyğun olaraq təyin
edirik

 # ee /etc/rc.conf => Startup faylına hostname adını əlavə edirik

```
hostname="FREERADIUS.VELO.LAN"
```

 # ntpdate 10.0.0.10 => Vaxtı AD serverinə uyğun yeniləyirik
 # ee /etc/resolv.conf => DNS serverimizin məlumatlarını **resolv.conf**
faylına əlavə edirik

```
search VELO.LAN
nameserver 10.0.0.10
```

 # ee /etc/hosts => **hosts** faylına öz maşınımızın hostname və
interfeys İP-i aşağıdakı kimi əlavə edirik

```
10.0.0.1                FREERADIUS.VELO.LAN FREERADIUS
```

 # ee /etc/sysctl.conf => bu fayla aşağıdakı sətrləri əlavə edirik

```
kern.maxfiles=25600
kern.maxfilesperproc=16384
net.inet.tcp.sendspace=65536
net.inet.tcp.recvspace=65536
```

 # pkg update => Repozitorları yeniləyirik
 # pkg install samba41 => Samba 4.1 paketini quraşdırırıq
 # rehash => Binar fayllar bazasını yeniləyirik

 # ee /etc/krb5.conf => **krb5.conf** faylı yaradıb aşağıdakı sətrləri əlavə
edirik

```
[libdefaults]
        default_realm = VELO.LAN
        dns_lookup_realm = true
```

```
        dns_lookup_kdc = true
        ticket_lifetime = 24h
        renew_lifetime = 7d
        forwardable = yes
```

 # ee /etc/nsswitch.conf => **nsswitch.conf** faylında sarı fondakı sətrləri
yeniləyirik

```
group: files winbind
group_compat: nis
hosts: files dns
networks: files
passwd: files winbind
passwd_compat: nis
shells: files
services: compat
services_compat: nis
protocols: files
rpc: files
```

 # ee /usr/local/etc/smb4.conf => **smb4.conf** faylı yaradıb aşağıdakı sətrləri əlavə
edirik (sarı fondakı adlara diqqət yetiririk)

```
[global]
        workgroup = VELO
        server string = Samba Server Version %v
        security = ads
        realm = VELO.LAN
        domain master = no
        local master = no
        preferred master = no
        socket options = TCP_NODELAY IPTOS_LOWDELAY SO_RCVBUF=131072
        use sendfile = true

        idmap config * : backend = tdb
        idmap config * : range = 100000-299999
        idmap config VELO : backend = rid
        idmap config VELO : range = 10000-99999
        winbind separator = +
        winbind enum users = yes
        winbind enum groups = yes
        winbind use default domain = yes
        winbind nested groups = yes
        winbind refresh tickets = yes
        template homedir = /home/%D/%U
        template shell = /bin/false
```

 # net ads join -U administrator => Samba-nı VELO.LAN domainində qoşuruq
Enter administrator's password: ****** => Çıxan sətrdə domainimizin Administrator

şifrəsini daxil edirik. Aşağıdakı sətr kimi
bir şey çıxarsa hər şey qaydasındadır.

```
Using short domain name -- VELO
Joined 'FREERADIUS' to dns domain 'VELO.LAN'
```

 # net ads testjoin => Qoşulmanı bir də test edirik. Bizə "Join is OK"
yazısını qaytarmalıdır

 # ee /etc/rc.conf => Aşağıdakı sətrləri startup faylına əlavə edirik

```
samba_server_enable="YES"
winbindd_enable="YES"
smbd_enable="YES"
nmbd_enable="YES"
```

 # service samba_server start => Samba servisini işə salırıq

 # kinit administrator => Kerberos vasitəsi ilə Domainə giriş
üçün bilet əldə edirik.

administrator@VELO.LAN's Password:****** => Çıxan sətrə Domain Admin şifrəsini
daxil edirik

 # klist => Bileti aldığımızı yoxlayırıq.
Aşağıdakı kimi hesabat çıxmalıdır.

```
Credentials cache: FILE:/tmp/krb5cc_0
        Principal: administrator@VELO.LAN

  Issued                Expires               Principal
May  5 10:33:43 2015  May  5 20:33:43 2015  krbtgt/VELO.LAN@VELO.LAN
```

Sonra Winbind-i test edirik

 # wbinfo -u => Bu əmr sizə Domain-də olan istifadəçi adlarını çıxartmalıdır

 # wbinfo -g => Bu əmr sizə Domain-də olan qrup adlarını çıxartmalıdır

 # wbinfo -a istifadeciadi%shifre => Domaində olan hər hansı bir istifadəçi adını və
şifrəsini bu şəkildə daxil edib test etsək aşağıdakı kimi nəticə verməlidir.

```
plaintext password authentication succeeded
challenge/response password authentication succeeded
```

 # service samba_server restart => Samba servisini yenidən işə salırıq

Domainimizə qoşulma uğurla həyata keçdikdən sonra Freeradius-un NTLM (NT Lan Manager) modulu vasitəsi ilə Domainlə inteqrasiyasına keçək.

Qeyd: NTLM modulu LDAP modulundan fərqli olaraq PAP qeydiyyat üsulundan başqa MSCHAP,EAP kimi digər şifrələmə metodikasını dəstəkləyir. Yəni qeydiyyat daha təhlükəsiz şəkildə həyata keçirilir.

 # pkg install freeradius => freeradius: 2.2.7 paketini quraşdırırıq
 # rehash => Binar fayllar bazasını yeniləyirik
 # ntlm_auth --request-nt-key --username=administrator => NTLM qeydiyyatını test edirik
Password:****** => Çıxan sətrə domain admin şifrəsini daxil edirik. Bizə "NT_STATUS_OK: Success (0x0)" hesabatını qaytarırsa NTLM qeydiyyatı işləyir
 # cd /usr/local/etc/raddb/ => Freeradiusun quraşdırma fayllarının yerləşdiyi qovluğa daxil oluruq

MSCHAP modulunun quraşdırma faylını açırıq və sarı fonda qeyd olunmuş sətri tapıb qarşısında olan şərhi ("#"-nı) silirik və **/path/to/ntlm_auth** yerinə ntlm_auth binar faylının tam ünvanını yazırıq. **ntlm_auth** binar faylının tam ünvanını tapmaq üçün isə "**# whereis ntlm_auth**" yazaraq çıxan nəticədə görə bilərik.

 # whereis ntlm_auth => ntlm_auth binar faylının tam ünvanını tapmaq üçün bu əmrdən istifadə edirik və nəticədə tam ünvanı görürük.

```
ntlm_auth: /usr/local/bin/ntlm_auth
```

 # ee modules/mschap => **mschap** modulunun quraşdırma faylına daxil oluruq, şərhi silirik və yalnız **/path/to/ntlm_auth** olan yeri sarı fonda gördüyünüz kimi düzgün ünvana dəyişirik və digər yerləri olduğu kimi saxlayırıq.

```
ntlm_auth = "/usr/local/bin/ntlm_auth --request-nt-key --
username=%{%{Stripped-User-Name}:-%{%{User-Name}:-None}} --
challenge=%{%{mschap:Challenge}:-00} --nt-response=%{%{mschap:NT-Response}:-
00}"
```

Sonra test etmək üçün bizə 2 konsol pəncərəsi lazım olacaq.

1-ci konsol pəncərəsində

 # radiusd -fX => FreeRadiusun debug əmrini daxil edirik. Aşağıdakı sətrlər
kimi sətrlər çıxır və gözləyirik

```
.........
Sending Access-Accept of id 198 to 127.0.0.1 port 56224
        MS-CHAP-MPPE-Keys =
0x00000000000000000f1eef4a31ec3792beebab6d25e82b72a0000000000000000
        MS-MPPE-Encryption-Policy = 0x00000001
        MS-MPPE-Encryption-Types = 0x00000006
Finished request 0.
Going to the next request
Waking up in 4.9 seconds.
Cleaning up request 0 ID 198 with timestamp +4
Ready to process requests.
```

2-ci konsol pəncərəsində isə
"radtest *istifadeciadi "şifrə" freeradius-server-ip* **10** *pre-shared-secret-key"*
Yəni bizim vəziyyətimizdə aşağıdakı kimi əmri daxil edirik.
 # radtest -t mschap camal "C123456789c" localhost 0 testing123
Nəticədə əgər aşağıdakı kimi **Access-Accept** gördüksə bu o deməkdir ki, hər şey işləyir.

```
Sending Access-Request of id 33 to 127.0.0.1 port 1812
        User-Name = "camal"
        NAS-IP-Address = 10.0.0.1
        NAS-Port = 0
        Message-Authenticator = 0x00000000000000000000000000000000
        MS-CHAP-Challenge = 0x106f51f972e17124
        MS-CHAP-Response =
0x000100000000000000000000000000000000000000000000000024c9483ba63f967effa5e49
9715e679c5d1f0f942a36420b
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=33,
length=84
        MS-CHAP-MPPE-Keys =
0x00000000000000000f1eef4a31ec3792beebab6d25e82b72a0000000000000000
        MS-MPPE-Encryption-Policy = 0x00000001
        MS-MPPE-Encryption-Types = 0x00000006
```

1-ci konsol pəncərəsində isə nəticə aşağıdakı kimi olmalıdır.

```
.........
....
....
Found Auth-Type = MSCHAP
# Executing group from file /usr/local/etc/raddb/sites-enabled/default
+group MS-CHAP {
[mschap] Client is using MS-CHAPv1 with NT-Password
[mschap]        expand: %{Stripped-User-Name} ->
[mschap]        ... expanding second conditional
[mschap]        expand: %{User-Name} -> camal
[mschap]        expand: %{%{User-Name}:-None} -> camal
[mschap]        expand: --username=%{%{Stripped-User-Name}:-%{%{User-Name}:-
None}} -> --username=camal
```

```
[mschap]  mschap1: 58
[mschap]        expand: %{mschap:Challenge} -> 5872c80af597f400
[mschap]        expand: --challenge=%{%{mschap:Challenge}:-00} -> --
challenge=5872c80af597f400
[mschap]        expand: %{mschap:NT-Response} ->
bd785869e3086f6f8af55af3ac177b59e925e2a8bafe1f9f
[mschap]        expand: --nt-response=%{%{mschap:NT-Response}:-00} -> --nt-
response=bd785869e3086f6f8af55af3ac177b59e925e2a8bafe1f9f
Exec output: NT_KEY: F1EEF4A31EC3792BEEBAB6D25E82B72A
Exec plaintext: NT_KEY: F1EEF4A31EC3792BEEBAB6D25E82B72A
[mschap] Exec: program returned: 0
[mschap] adding MS-CHAPv1 MPPE keys
++[mschap] = ok
+} # group MS-CHAP = ok
# Executing section post-auth from file /usr/local/etc/raddb/sites-
enabled/default
+group post-auth {
++[exec] = noop
+} # group post-auth = noop
Sending Access-Accept of id 133 to 127.0.0.1 port 38369
        MS-CHAP-MPPE-Keys =
0x0000000000000000f1eef4a31ec3792beebab6d25e82b72a0000000000000000
        MS-MPPE-Encryption-Policy = 0x00000001
        MS-MPPE-Encryption-Types = 0x00000006
Finished request 3.
Going to the next request
Waking up in 4.9 seconds.
Cleaning up request 3 ID 133 with timestamp +295
Ready to process requests.
```

Hər şeyin işlədiyini gördükdən sonra Freeradiusu startup faylına əlavə edirik və sonra da işə salırıq

```
# echo 'radiusd_enable="YES"' >> /etc/rc.conf
# service radiusd start
```

CentOS üzərində DaloRadius qurulması

daloRADIUS - Əsasən İSP yükləmələrini qarşılayan və HotSpot idarəetməsi üçün nəzərdə tutulan qabaqcıl RADİUS web idarəetmə proqramıdır. Istifadəçilərin idarəedilməsi, qrafik hesabatların hazırlanması, hesablar, billing motoru və coğrafi təyinat üçün GoogleMaps-lə inteqrasiya imkanına sahibdir.

Öncə sistemin reposlarını və yüklənmiş paketlərlə kernel yeniləyirik:

```
yum update
yum upgrade
```

Sonra FreeRADIUS, MySQL və PHP serveri və mysql-ə qoşulma üçün digər paketləri yükləyək:

```
yum install freeradius freeradius-mysql freeradius-utils mysql-server mysql
php-mysql php

chkconfig mysqld on            # MySQL serveri startup servislərə əlavə edirik
/etc/init.d/mysqld start       # MySQL serveri işə salırıq

/usr/bin/mysql_secure_installation  # root şifrəsi təyin edirik, anonim
                                    qoşulmanı söndürürük, test bazanı silirik
                                    və uzaqdan root istifadəçi ilə qoşulmağa
                                    qadağa təyin edirik.

mysql -uroot -pfreebsd              # MySQL-ə root istifadəçi ilə daxil
                                    oluruq
```

RADIUS bazası, istifadəçisi, şifrəsi yaradıb uzaqdan qoşulmağa izin veririk.

```
CREATE DATABASE radius;
GRANT ALL PRIVILEGES ON radius.* TO radius@localhost IDENTIFIED BY "freebsd";
FLUSH PRIVILEGES;
exit
```

RADIUS bazası üçün FreeRADIUS sxemini qururuq

```
mysql -uradius -pfreebsd radius < /etc/raddb/sql/mysql/schema.sql

service iptables stop   # IPTABLES-i söndürürük(şəxsi istəyinizə baxır)
chkconfig --level 0123456 iptables off
chkconfig --level 0123456 ip6tables off
```

Və şəxsi praktikamda daloradius-un php ilə bağlı çoxlu yükləmələrini gördükdən sonra php-yə aid bütün paketləri yüklədim. Ancaq siz bunu süzgəcdən keçirib yalnız öz tələbinizə uyğun olanı seçə bilərsiniz:

```
yum install `yum search php- |grep php- | grep -v === | awk '{ print $1 }'`
pear install DB          # Mütləq bu paketi yükləyirik ki, Daloradius DB-yə
                         qoşula bilsin
```

/etc/ raddb/sql.conf faylını aşağıdakı kimi quraşdırırıq:

```
sql {
        database = "mysql"
        driver = "rlm_sql_${database}"
        server = "localhost"
        port = 3306
        login = "radius"
        password = "freebsd"
        radius_db = "radius"
        acct_table1 = "radacct"
        acct_table2 = "radacct"
        postauth_table = "radpostauth"
        authcheck_table = "radcheck"
        authreply_table = "radreply"
        groupcheck_table = "radgroupcheck"
        groupreply_table = "radgroupreply"
        usergroup_table = "radusergroup"
        deletestalesessions = yes
        sqltrace = no
        sqltracefile = ${logdir}/sqltrace.sql
        num_sql_socks = 5
        connect_failure_retry_delay = 60
        lifetime = 0
        max_queries = 0
        nas_table = "nas"
        $INCLUDE sql/${database}/dialup.conf
}
```

/etc/raddb/radiusd.conf faylının içində aşağıdakı sətiri tapıb qarşısından şərhi silirik:

```
$INCLUDE sql.conf
```

/etc/raddb/sites-available/default faylının içində isə **'authorize {}'**, **'accounting {}'** və **'session {}'** bölmələrinin içində **sql** sətirini tapıb qarşısından şərhi silin.

Həmçinin uyğun olaraq **/etc/raddb/sites-available/inner-tunnel** faylında da **'authorize {}'** və **'session {}'** bölmələrinin içində **sql** sətirini tapıb qarşısından şərhi silin.

Sonra **/etc/raddb/clients.conf** faylının içinə istədiyiniz client-i əlavə edin. Mən test üçün **localhost**-u əlavə etdim. Aşağıdakı kimi:

```
client localhost {
        ipaddr = 127.0.0.1
        secret = freebsd
        require_message_authenticator = no
```

```
        shortname = localhost
        nastype = other
}

service radiusd start          # RADIUS serveri işə salırıq
chkconfig radiusd on           # RADIUS servisini starupa əlavə edirik

radius -fX  # RADIUS serveri debug etmək üçün bu rejimdə işə sala bilərsiniz
```

Sonra FreeRADIUS-un WEB management alətini quraşdırırıq, yəni Daloradius-u:

```
cd /tmp/           # TEMP qovluğuna daxil oluruq və daloradius paketini
                   endiririk
wget
http://sourceforge.net/projects/daloradius/files/latest/download?source=files
mv download\?source\=files daloradius-0.9-9.tar.gz
tar zxvf daloradius-0.9-9.tar.gz
```

Daloradius bazasının strukturunu MySQL-ə import edirik:

```
mysql -uradius -pfreebsd radius < /tmp/daloradius-0.9-9/contrib/db/fr2-mysql-
daloradius-and-freeradius.sql
```

/tmp/daloradius-0.9-9/library/daloradius.conf.php faylında aşağıdakı sətirləri uyğun olaraq config edirik:

```
$configValues['DALORADIUS_VERSION'] = '0.9-9';
$configValues['FREERADIUS_VERSION'] = '2';
$configValues['CONFIG_DB_ENGINE'] = 'mysql';
$configValues['CONFIG_DB_HOST'] = 'localhost';
$configValues['CONFIG_DB_PORT'] = '3306';
$configValues['CONFIG_DB_USER'] = 'radius';
$configValues['CONFIG_DB_PASS'] = 'freebsd';
$configValues['CONFIG_DB_NAME'] = 'radius';
```

Sonda quraşdırdığımız qovluğu artıq WEB serverimizin işlək public_html qovluğuna köçürürük:

```
mv /tmp/daloradius-0.9-9 /var/www/html/daloradius
```

Öncədən aşağıdakı qovluq və faylları yaradırıq ki, daloradius şikayət etməsin. Mənim halımda gileylənirdi və ona görədə istədiyi hər şeyi etdim ki, işləsin:

```
mkdir /var/www/html/themes
mkdir /var/www/html/themes/blue
mkdir /var/www/html/themes/blue/css
touch /var/www/html/themes/blue/css/auto-complete.css
chown -R apache:apache /var/www/html/
touch /tmp/daloradius.log
```

/etc/php.ini faylında ölkə ərazimizi aşağıdakı sətirdəki kimi təyin edirik:

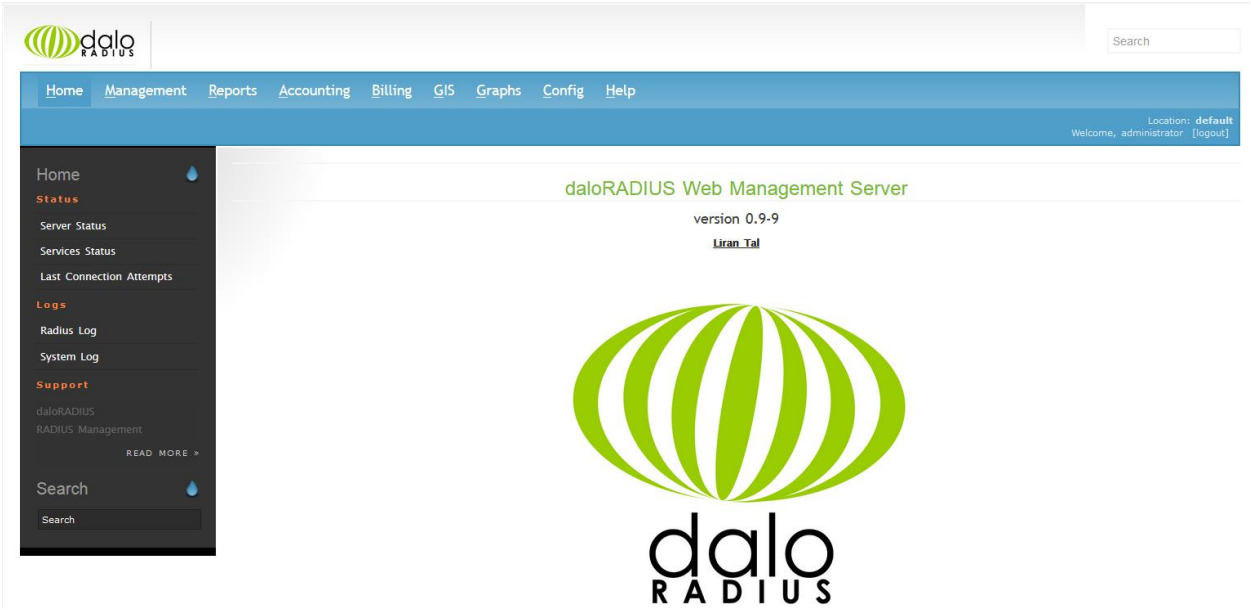
```
date.timezone = 'Asia/Baku'
```

```
service apache2 restart        # Apache-ni restart edirik ki,
                               quraşdırmalarımız işə düşsün
```

Çıxan səhvləri təyin etmək üçün **/var/log/httpd/error.log** faylında WEB serverin verdiyi səhvlərə baxırıq.

Sonda **http://10.50.3.202/daloradius/** linkinə aşağıdakı istifadəçi adı və şifrə ilə daxil oluruq(şəkildəki kimi):
login: **Administrator**
password: **radius**

Əgər uğurla daxil olsanız aşağıdakı şəkil çap edilməlidir:

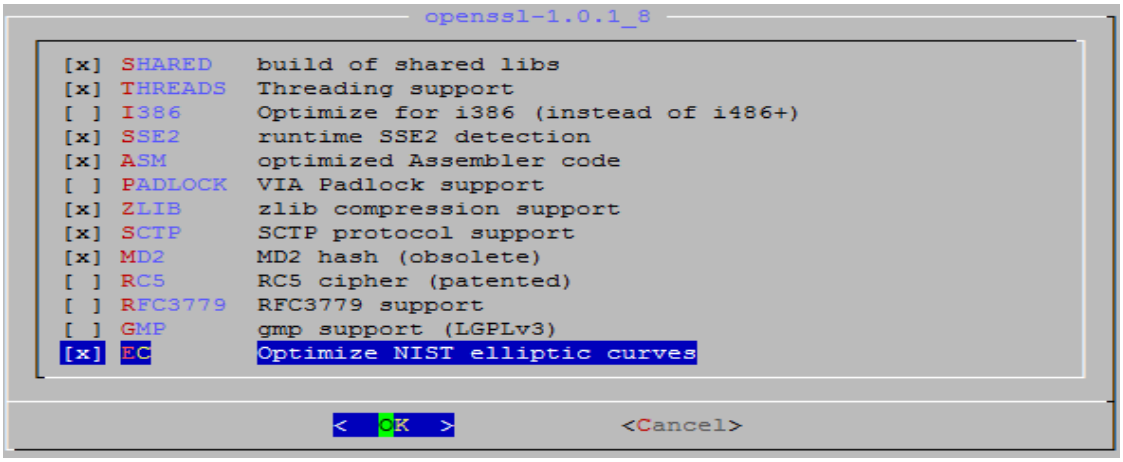


FreeBSD FreeRADİUS EAP-TLS

Bu başlıqda biz FreeRADIUS vasitəsilə WiFi qoşulmasını Sertifikatla edəcəyik. Yeni istifadəçi FreeRADIUS Server tərəfindən generasiya edilmiş CA sertifikatını və həmin CA ilə imzalanmiş açarı öz Desktop-unda yüklədikdən sonra WiFi-ya qoşula biləcək.
Bütün işlər FreeBSD 9.2 x64 və FreeRADIUS 2.2.2 üzərində görülmüşdür.

Yada mobil telefonlar sadəcə istifadəçi adı və şifrə ilə qoşulacaq.

cd /usr/ports/security/openssl
make config



make install

```
root@backupbsd:~ # tar -zxf CA_scritps.tgz      # TGZ paketi açırıq.
root@backupbsd:~ # chmod -R +x scritps/         # Sertifikatları yaratmaq
                                                üçün yetki veririk
root@radius:~ # cd scritps/   # Scriptin ünvanına daxil oluruq. İşə salırıq.
root@owncloud:~/scritps # ./CA_root.sh ROOTPASSWORD
Generating a 1024 bit RSA private key
.....................................++++++
...++++++
writing new private key to 'pem/newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
```

```
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:BAKU
Locality Name (eg, city) []:Narimanov
Organization Name (eg, company) [Internet Widgits Pty Ltd]:DOMAIN
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:RADIUS Root Certificate
Email Address []:jamal.shahverdiyev@domain.az
MAC verified OK
```

Server sertifikatlarını yaradırıq.

```
root@backupbsd:~/scritps # echo "01" > ./demoCA/serial
root@backupbsd:~/scritps # touch ./demoCA/index.txt
root@owncloud:~/scritps # ./CA_server.sh server.name.local SERVERPASSWORD
ROOTPASSWORD
Generating a 1024 bit RSA private key
.............++++++
.......++++++
writing new private key to 'pem/newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:BAKU
Locality Name (eg, city) []:Narimanov
Organization Name (eg, company) [Internet Widgits Pty Ltd]:DOMAIN
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:server.name.local
Email Address []:user@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:SERVERPASSWORD
An optional company name []:
Using configuration from /etc/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Dec  9 05:25:17 2013 GMT
            Not After : Dec  7 05:25:17 2023 GMT
        Subject:
            countryName               = AZ
            stateOrProvinceName       = BAKU
```

```
            localityName              = Narimanov
            organizationName          = DOMAIN
            organizationalUnitName    = IT
            commonName                = server.name.local
            emailAddress              = user@gmail.com
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
Certificate is to be certified until Dec  7 05:25:17 2023 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
MAC verified OK
```

Klientin sertifikatını yaradırıq.

```
root@owncloud:~/scritps # ./CA_client.sh client.name.local CLIENTPASSWORD
ROOTPASSWORD
Generating a 1024 bit RSA private key
.....................................................................++++++
.................++++++
writing new private key to 'pem/newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:BAKU
Locality Name (eg, city) []:Narimanov
Organization Name (eg, company) [Internet Widgits Pty Ltd]:DOMAINinfo
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:client.name.local
Email Address []:admin@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:CLIENTPASSWORD
An optional company name []:
Using configuration from /etc/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 2 (0x2)
        Validity
```

```
            Not Before: Dec  9 05:36:15 2013 GMT
            Not After : Dec  7 05:36:15 2020 GMT
        Subject:
            countryName               = AZ
            stateOrProvinceName       = BAKU
            localityName              = Narimanov
            organizationName          = DOMAINinfo
            organizationalUnitName    = IT
            commonName                = client.name.local
            emailAddress              = admin@gmail.com
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
Certificate is to be certified until Dec  7 05:36:15 2020 GMT (2555 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
MAC verified OK
```

Diffie-Hellman açarı yaradırıq.

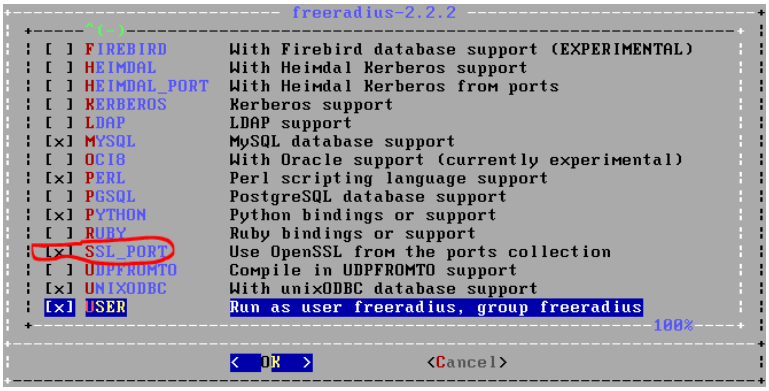
```
root@backupbsd:~/scritps # openssl dhparam -out dh1024.pem 1024
```

Təsüdufi uzunluqda 1024 bayt yazırıq.

```
root@backupbsd:~/scritps # dd if=/dev/urandom of=random count=2
2+0 records in
2+0 records out
1024 bytes transferred in 0.000061 secs (16843009 bytes/sec)
```

FreeRADIUS 2.2 versiyasını yükləyək.

```
cd `whereis freeradius2 | awk '{ print $2 }'`    # FreeRADIUS2 portuna daxil
                                                 oluruq.
make config                                      # Lazımi modulları seçirik.
```




```
make install                                     # Yükləyirik.
```

Növbəti açarları **'/usr/local/etc/raddb/certs'** ünvanına nüsxələyək.

```
root@radius:/usr/ports/net/freeradius2 # cd /root/scritps/  # sertifikatların
                                                            qovluğuna daxil
                                                            olaq
root@owncloud:~/scritps # cp ./pem/root.pem /usr/local/etc/raddb/certs/
root@owncloud:~/scritps # cp ./pem/server.name.local.pem
/usr/local/etc/raddb/certs/
root@owncloud:~/scritps # cp ./dh1024.pem /usr/local/etc/raddb/certs/
root@owncloud:~/scritps # cp ./random /usr/local/etc/raddb/certs/
```

'/usr/local/etc/raddb/clients.conf' faylına aşağıdakı sətirləri əlavə edirik.

```
client accessp {
        secret          = qwerty                # WiFi ile RADIUS arasinda
olan Pre-Shared key
        ipaddr          = 10.50.12.200          # WiFI AP-nin IP adresi
        shortname       = Test Access point
}
```

Həmçinin **'/usr/local/etc/raddb/radiusd.conf'** faylında aşağıdakı sətirlərin şərhsiz olmasını yoxlayın.

```
modules {
........
        $INCLUDE ${confdir}/modules/
........
        $INCLUDE eap.conf
}
```

"**/usr/local/etc/raddb/eap.conf**" faylında **eap {** bölümündə aşağıdakı sətirləri quraşdırırıq.

```
default_eap_type = tls # EAP-TLS protokolu istifadə edirik.
........
tls {
        certdir = ${confdir}/certs
        cadir = ${confdir}/certs

        private_key_password = SERVERPASSWORD
        private_key_file = ${certdir}/server.name.local.pem
        certificate_file = ${certdir}/server.name.local.pem
        CA_file = ${cadir}/root.pem

        dh_file = ${certdir}/dh1024.pem
        random_file = ${certdir}/random
}
```

Ancaq şəxslər ola bilər ki, onların iPAD və Android olan telefonları ola bilər və onlara sertifikatları yükləyə bilmərik. Bunun üçün isə EAP-PEAP quraşdırmalıyıq. **/usr/local/etc/raddb/modules/mschap** faylında aşağıdakı dəyişikliyi edəcəyik.

```
mschap {
        use_mppe = yes                  # mppe alqoritmini istifadə et
        require_encryption = yes        # şifrələnmə istifadə et
        require_strong = yes            # Həmişə 128 bitlik açar tələb edir
```

```
        with_ntdomain_hack = yes        # Windows bizə istifadəçinin adını
        # DOMAIN\username, formasında yollayır, ancaq cavab olaraq yalnız
        # istifadəçi ilə qayıdır. Bu HACK həmin problemi həll edir.
```

Aşağıdakı sətirləri isə **/usr/local/etc/raddb/modules/realm** faylına əlavə edirik.

```
realm ntdomain {
        format = prefix
        delimiter = "\\"
        ignore_default = no
        ignore_null = no
}
```

/usr/local/etc/raddb/sites-available/default faylında aşağıdakı sətirləri dəyişdiririk.

```
authorize {
.........
#       suffix
        ntdomain
.........
}
```

/usr/local/etc/raddb/proxy.conf faylına aşağıdakı sətirləri əlavə edirik.

```
realm DEFAULT {
        type            = radius
        authhost        = LOCAL
        accthost        = LOCAL
}
```

Sonra **eap.conf** faylını açırıq və **eap {** başlığına peap artırdıqdan sonra bizim **TLS**-imizə həmçinin **PEAP** əlavə edirik. /usr/local/etc/raddb/eap.conf faylında aşağıdakı dəyişiklikləri edirik.

```
default_eap_type = tls peap
........
peap {
        default_eap_type = mschapv2
}
```

İndi isə FreeRADIUS üzərində istifadəçi bazasını yaratmaq lazımdır. FreeRADIUS istifadəçi bazası olaraq MySQL, LDAP, PgSQL və hətta sistemin passwd faylından istifadə edə bilər. Biz sadəcə adi fayldan götürəcəyik. /usr/local/etc/raddb/modules/files faylında aşağıdakı dəyişiklikləri edəcəyik.

```
files {
        usersfile = ${confdir}/users
        compat = no
}
```

/usr/local/etc/raddb/users faylına isə aşağıdakı istifadəçiləri əlavə edirik.

```
user1 Cleartext-Password := "user1pass"
```

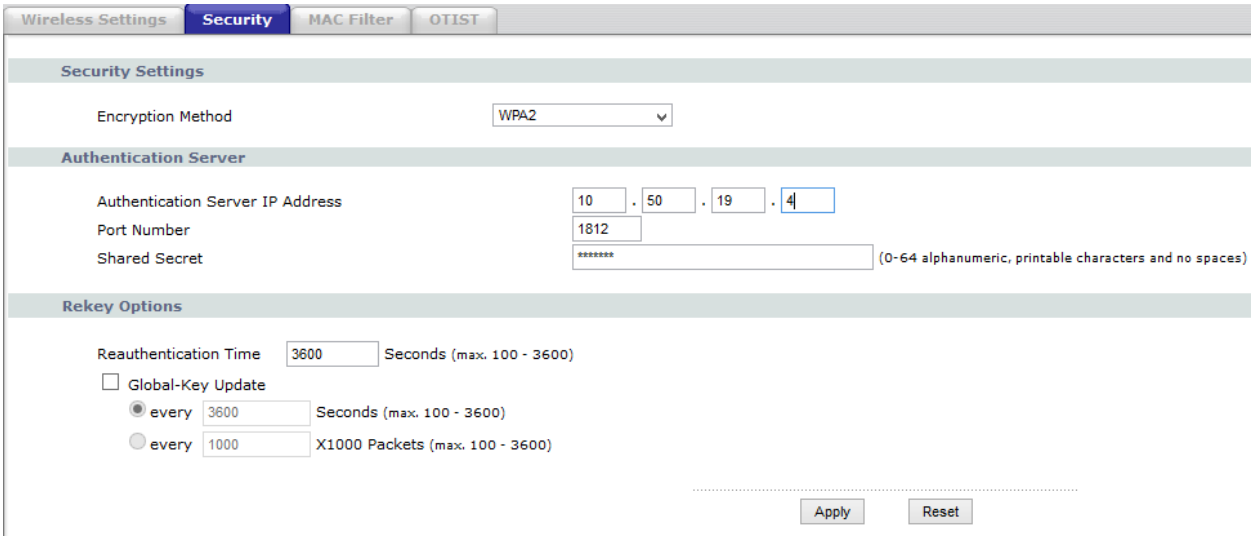
```
user2 Cleartext-Password := "user2pass"
user3 Cleartext-Password := "user3pass"
user4 Cleartext-Password := "user4pass"
```

FreeRADIUS-u işə salaq və test edək.

```
echo 'radiusd_enable="YES"' >> /etc/rc.conf      # FreeRADIUS-u Startup-a
                                                  əlavə edirik.

chown freeradius:freeradius /usr/local/etc/raddb/certs/*      #
                                        Sertifikatlarının Ownerini təyin edirik
                                        Artıq ADSL modemə gələn müraciətləri
                                        FreeBSD RADIUS serverimizə
                                        yönləndiririk.

radiusd -fX              # RADIUS-u debug rejimdə işə salırıq.
```

AP-nin quraşdırılması və qoşulması

Access Point - quraşdırdıqda, sadəcə NAT rejimdə özü DHCP vasitəsilə IP paylayacaq. Qalanı isə istifadəçi adı, şifrə üçün ünvanı RADIUS-a yönləndirməkdir. Məsələn ZyXel üçün qurqaşdırma aşağıdakı şəkildəki kimi olacaq.



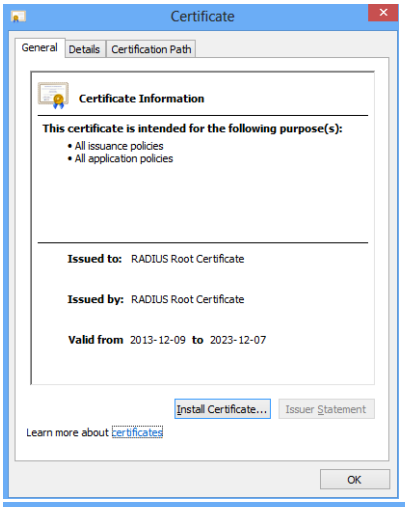
Clientin quraşdırılması və qoşulması.

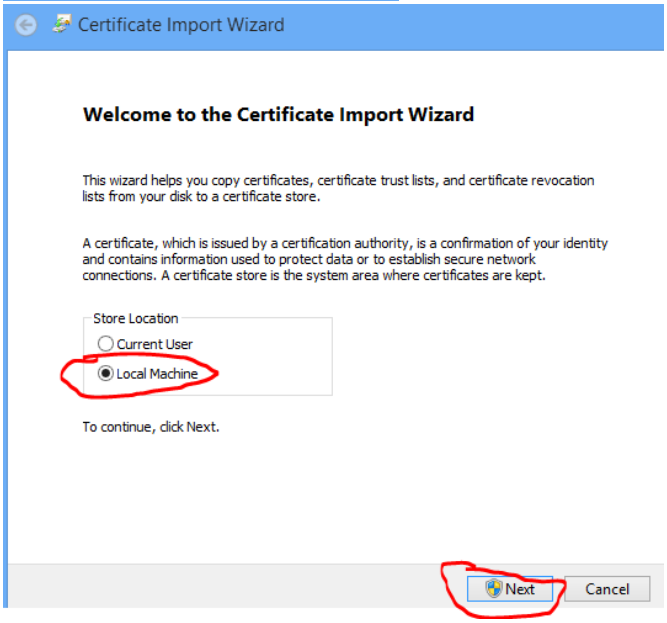
Hər bir clientin quraşdırılması və qoşulması üçün biz ayrıca sertifikat generasiya etməliyik. Bunlardan birini öncə generasiya etmişdik. Client sertifikatlarını **'/root/scripts/p12'** qovluğundan və root sertifikatı isə **'/root/scripts/der'** qovluğundan **'/mnt'** qovluğuna **nüsxələyirik**. Və **WinSCP** vasitəsi ilə ordan götürürük. Sonra isə həmin sertifikatı WiFi vermək istədiyimiz istifadəçinin Desktop-una yükləyirik. Sözsüz ki, öncə root sertifikatı və sonra isə **P12** genişlənməsində olan istifadəçi sertifikatını yükləmək lazımdır.

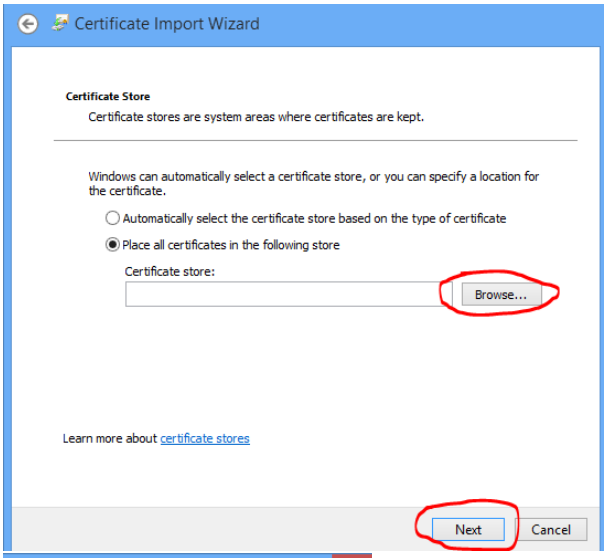
```
root@owncloud:~ # cp /root/scritps/der/root.der /mnt/
root@owncloud:~ # cp /root/scritps/p12/client.name.local.p12 /mnt/
```

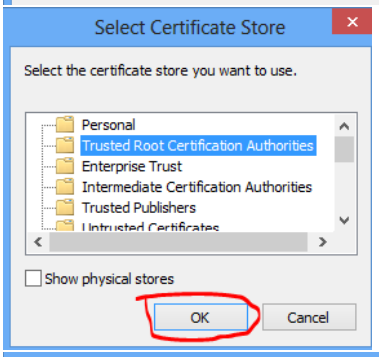
Windows7-də və Windows8-də sertifikatın yüklənməsi və quraşdırılması.

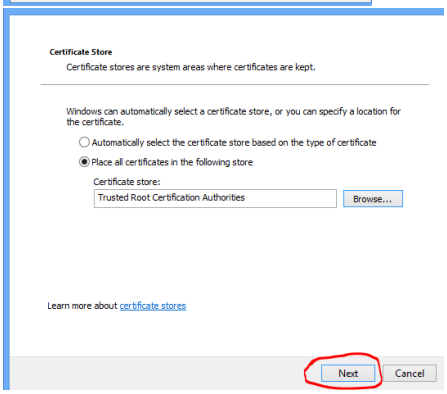
Deyək ki, Windows8 maşını üçün COMMON NAME-də olan adla camal.client.local adlı client sertifikatını RADIUS Root Ceritifacate vasitəsilə **CLIENTPASSWORD** şifrəsi ilə imzalamışıq. Ona görə öncə dediyimiz kimi, həmin **root** açarı və **camal.client.local.p12** açarını həmin **windows8** maşına upload edirik. Windows8 maşında mütləq öncə root sertifikatı **'Trusted root certificates'** bölməsinə yükləyirik. Aşağıdakı ardıcıllıqda göstərilir. **root** sertifikatın üstündə iki dəfə sıxılır və **Install certificate** düyməsinə sıxılır.

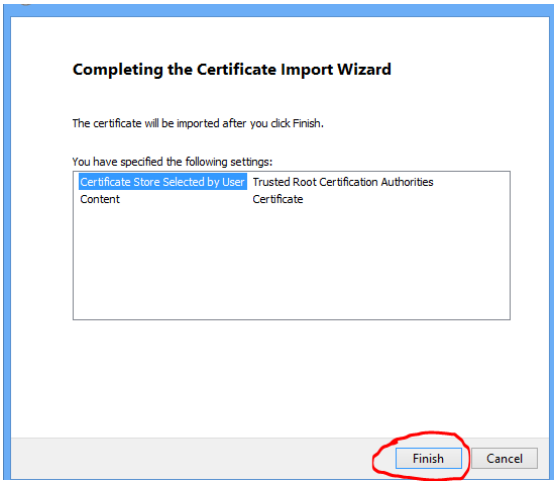






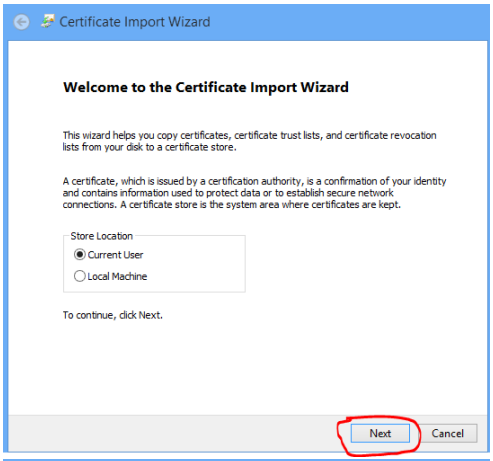


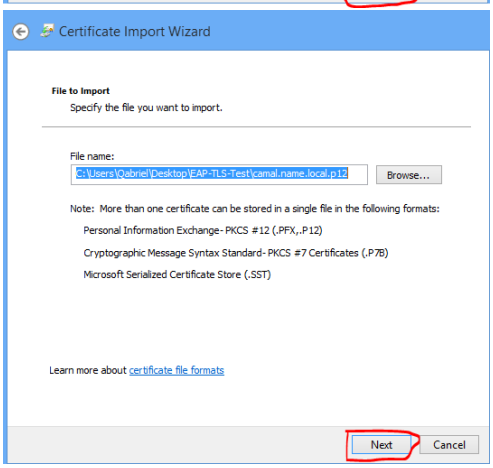


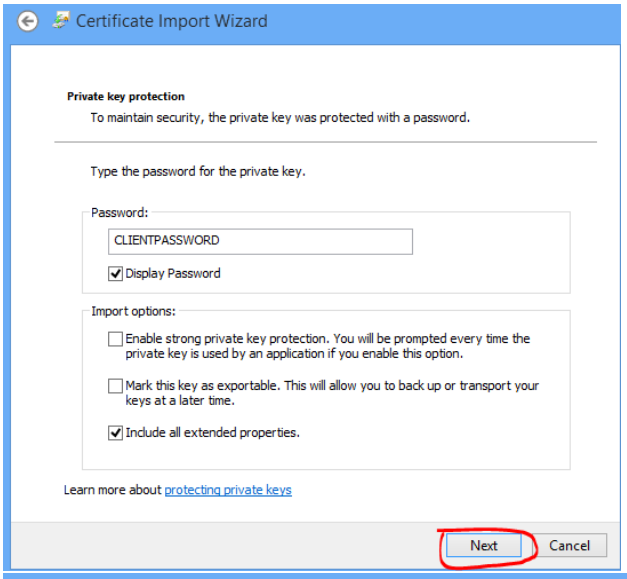


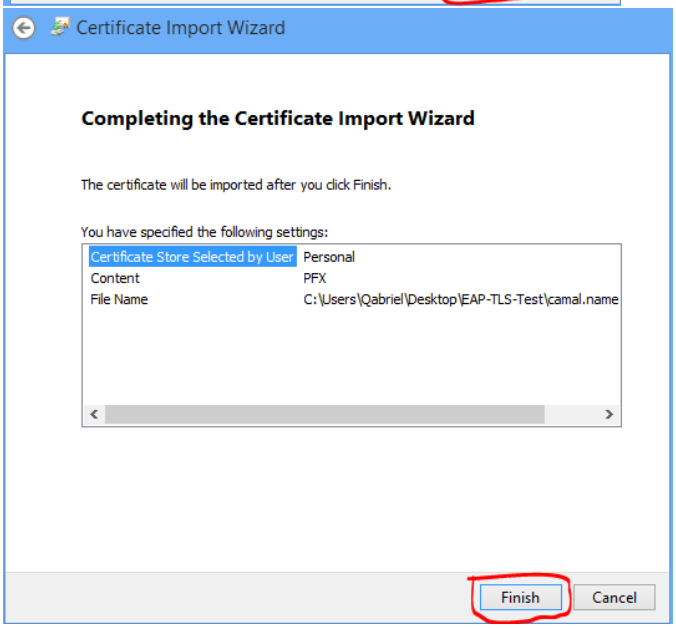
Finish -> OK -> OK

İndi isə Client **client.name.local.p12** sertifikatını yükləyək.



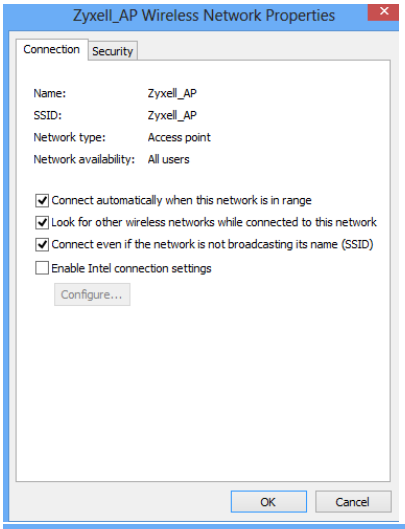


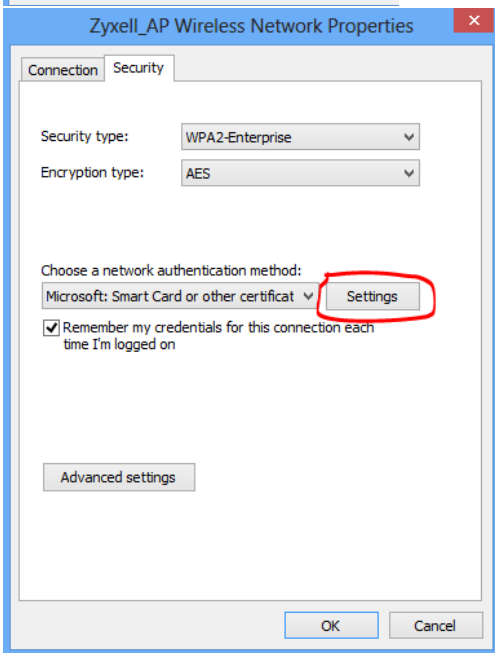


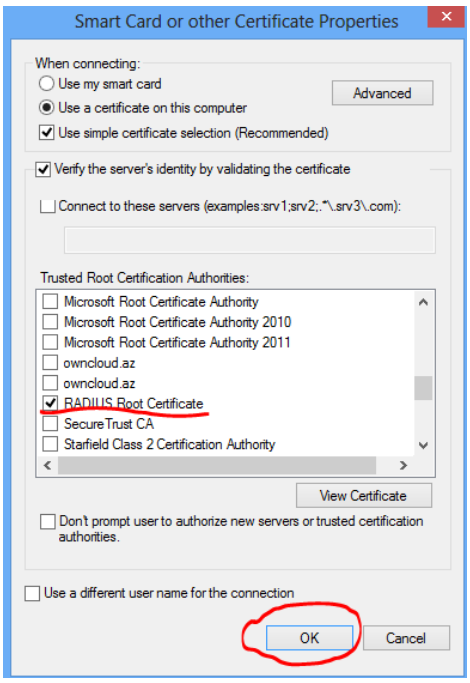


FINISH -> OK

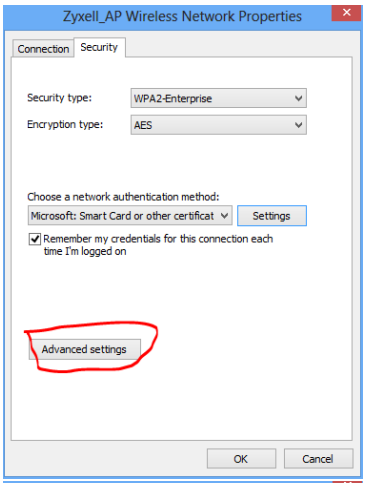
Öncə seçdiyimiz Access Point-in Properties-ni aşağıda formada quraşdırırıq.

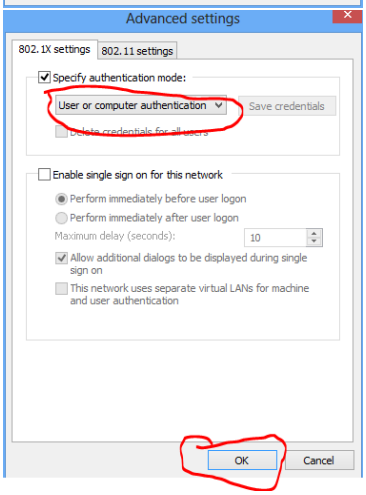


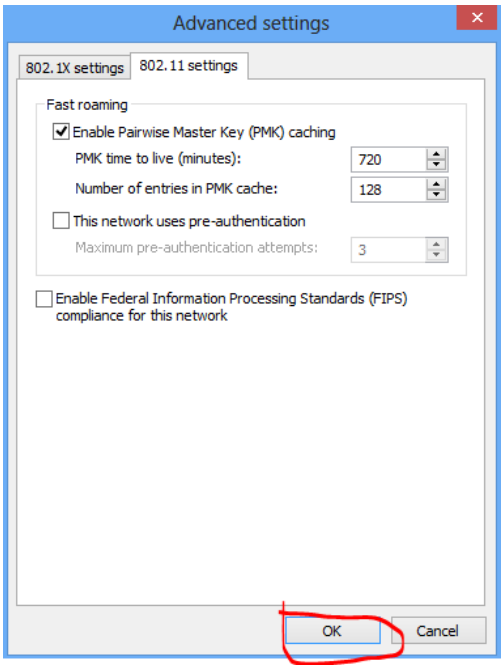




Sonra aşağıdakı şəkildən **Advanced Settings** bölməsinə keçirik.

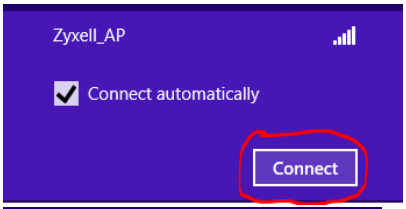


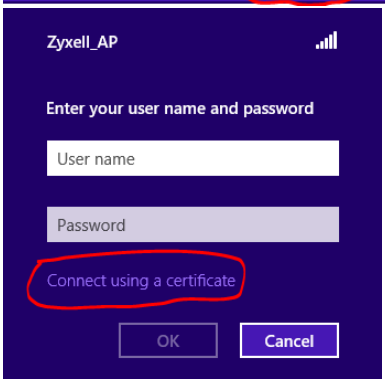


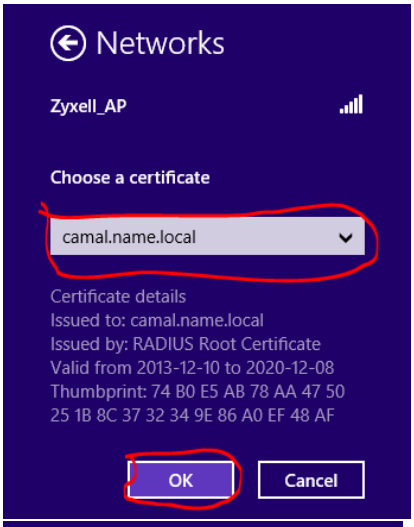


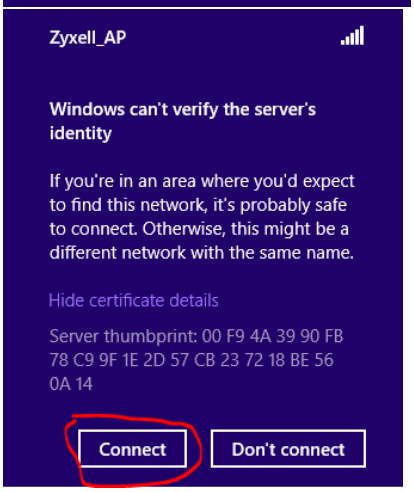
Sonda **OK** -> **OK**

Sonra Windows8 maşından qoşulaq Access Point-imizə.









Client sertifikatının revoke edilməsi

Sizdə işdə elə bir meqamlar ola bilər ki, kimsə işdən çıxar və ehtiyac olacaq ki, həmin işçinin sertifikatlarını sıfırlayasınız. Bu ona görədir ki, həmin istifadəçi artıq sizin WiFi şəbəkəsinə ümumiyyətlə daxil ola bilməsin(İşdən azad edilmənin adi prosedurudur)

Öncə açdığınız arxivin içində **CA_revoke.sh** adlı script mövcuddur hansı ki, bu işdə bizə kömək edəcək.
Misal üçün **client.namel.local** adlı istifadəçinin sertifikatını revoke edirik.

```
root@owncloud:~ # echo "01" >> /root/scritps/demoCA/crlnumber
root@owncloud:~/scritps # ./CA_revoke.sh client.name.local ROOTPASSWORD

rm: revoke/root-revoked.pem: No such file or directory      # Fikir verməyin
                                                            indi yaradılacaq.
rm: revoke/revoke.crl: No such file or directory            # Fikir verməyin
                                                            indi yaradılacaq.
Using configuration from /etc/ssl/openssl.cnf
Revoking Certificate 02.
Data Base Updated
```

```
Using configuration from /etc/ssl/openssl.cnf
pem/client.name.local.pem:
/C=AZ/ST=BAKU/L=Narimanov/O=DOMAINinfo/OU=IT/CN=client.name.local/emailAddres
s=admin@gmail.com
error 23 at 0 depth lookup:certificate revoked
```

/root/scritps/revoke/ qovluğunda **root-revoked.pem** adlı fayl yaranacaq. Bu açarı **/usr/local/etc/raddb/certs** qovluğuna nüsxələyirik.

```
root@owncloud:~/scritps # cp /root/scritps/revoke/root-revoked.pem
/usr/local/etc/raddb/certs/
```

/usr/local/etc/raddb/eap.conf faylında isə aşağıdakı formada dəyişiklik edirik.

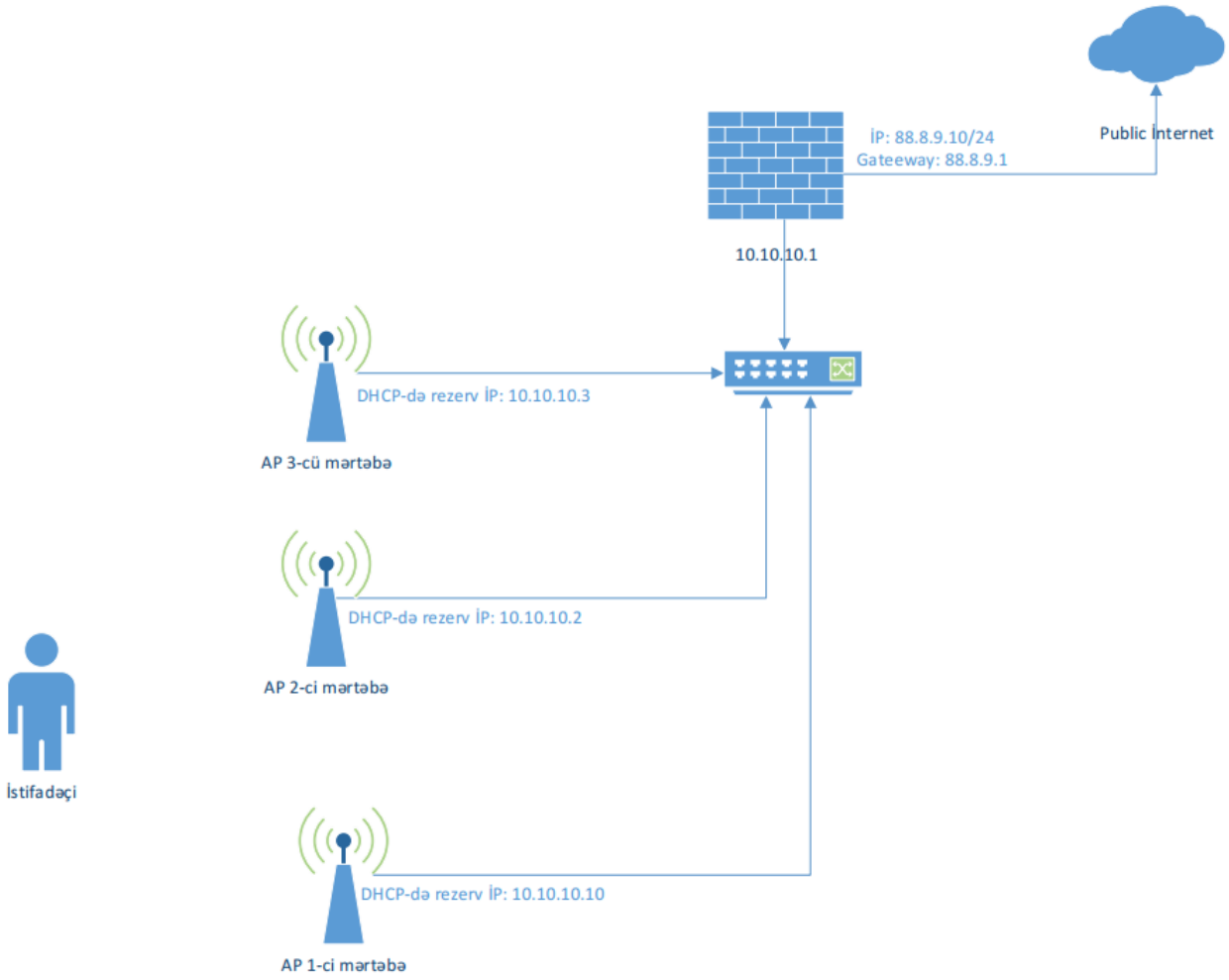
```
tls {
.........
#       CA_file = ${cadir}/root.pem # dəyişirik aşağıdakına

        CA_file = ${cadir}/root-revoked.pem
.........
#### əlavə edirik.
        check_crl = yes
}
```

```
root@owncloud:~/scritps # /usr/local/etc/rc.d/radiusd restart # Sonda
                                                    FreeRADIUS-u restart edirik
```

FreeBSD 10.1 x64 WiFi Hotspot

Məqsədimiz FreeBSD server üzərində aeroport-larda və otellərdə olduğu kimi, Captive Portalın qurulmasıdır. Loru dildə desək qonaq wifi-a istifadəçi adı və şifrə daxil etmədən qoşulur amma, internet resurslarından istifadə etməyə çalışdıqda onun veb browserinə istifadəçi adı və şifrənin daxil edilməsi çıxacaq. Əgər daxil edilən istifadəçi adı və şifrəsi döğru olarsa, qonaq internetdən istifadə edə biləcək.

Şəbəkə quruluşu aşağıdakı kimi olacaq:



Nəzərdə tutulur ki, siz artıq FAMP qurmusunuz və artıq Apache PHP stabil işləyir. Apache web serverimiz öncədən sistemdə yaratdığımız **jamal** istifadəçi adı və qrupu adından işləyir(Yəni **httpd.conf** faylında bu direktivlər mövcuddur: **User jamal** və **Group jamal**). Bütün AP-lərdə İP ünvanlar şəkildəki kimi, qurulmuş və DHCP server ilə ROUTER, FreeBSD Serverimizin daxili şəbəkə kartının İP ünvanı göstərilmişdir. Apache-da VirtualHost yaradılmışdır və **wifi.domain.az** domain adında işləyir. **wifi.domain.az** VirtualHost-un PUBLIC_HTML qovluğu **/usr/local/www/wifi/** ünvanıdır və qovluğun uzvluk, qrup yetkisi **jamal** təyin edilib. Eynilə **/usr/local/etc/apache24/httpd.conf** quraşdırma fayllında **Listen 80** və **Listen 443** təyin edilmişdir. **/usr/local/domen/wifi.domain.az** virtualhost faylının tərkibi isə aşağıdakı kimidir:

```
<VirtualHost *:80>
        RewriteEngine on
        ReWriteCond %{SERVER_PORT} !^443$
        RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R,L]
</VirtualHost>
<VirtualHost *:443>
        SSLEngine on
        SSLCertificateFile /usr/local/etc/apache24/ssl/wifi.pem
        SSLCertificateKeyFile /usr/local/etc/apache24/ssl/wifi.key
        DocumentRoot /usr/local/www/wifi/
<Directory "/usr/local/www/wifi">
        AllowOverride All
        Require all granted
</Directory>
</VirtualHost>
```

php üçün **pear** yükləmək və MySQL-də lazım olan verilənlər bazası ilə istifadəçini yaratmaq gərəkdir.

Pear yükləyirik və PHP üçün quraşdırırıq:

```
# cd `whereis pear | awk '{ print $2 }'`  - Port ünvanına daxil oluruq
# make -DBATCH install                    - Yükləyirik
```

php.ini faylını nüsxələyirik və göstərilən dəyişənləri məzmununda uyğun olaraq dəyişirik:

```
# cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini
```

/usr/local/etc/php.ini faylinda aşağıdakı dəyişiklikləri edirik:

```
date.timezone = 'Asia/Baku'
include_path = '.:/usr/local/share/pear'
```

MySQL verilənlər bazasını yaradaq, həmin bazaya istifadəçi təyin edək və wifi istifadəçilər üçün cədvəli yaradaq:

```
mysql> CREATE database wifi;
mysql> grant all privileges on wifi.* to wifidbuser@localhost identified by
'wifidbpassword';
mysql> use wifi;
mysql> CREATE TABLE `users` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `username` varchar(50) default NULL,
  `password` varchar(50) default NULL,
  `created` timestamp NOT NULL default CURRENT_TIMESTAMP on update
CURRENT_TIMESTAMP,
  `time_begin` timestamp NOT NULL default '0000-00-00 00:00:00',
  `time_end` timestamp NOT NULL default '0000-00-00 00:00:00',
  `status` tinyint(4) NOT NULL default '0',
  `rule_num` smallint(5) unsigned NOT NULL,
  PRIMARY KEY  (`id`),
  KEY `rule_num` (`rule_num`)
```

```
) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=cp1251 AUTO_INCREMENT=6 ;
```

/etc/rc.conf quraşdırma faylımız aşağıdakı kimi olacaq:

```
hostname="wifinat.domain.az"
ifconfig_em0="inet 88.8.9.10 netmask 255.255.255.0"
ifconfig_em1="inet 10.10.10.1 netmask 255.255.255.0"
defaultrouter="88.8.9.1"
sshd_enable="YES"
dumpdev="NO"

#### Local Disabled Services ####
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
sendmail_rebuild_aliases="NO"
syslogd_enable="YES"
syslogd_program="/usr/sbin/syslogd"
syslogd_flags="-ss"

#### Local worked services ####
tcp_drop_synfin="YES"
icmp_drop_redirects="YES"
dhcpd_enable="YES"
dhcpd_ifaces="em1"
dhcpd_conf="/usr/local/etc/dhcpd.conf"
gateway_enable="YES"
natd_enable="YES"
natd_interface="em0"
firewall_enable="YES"
firewall_type="UNKNOWN"
firewall_script="/etc/ipfw.conf"

#### Third Party Services ####
apache24_enable="YES"
mysql_enable="YES"
```

/etc/ipfw.conf faylına firewall quraşdırmalarımızın reboot-dan sonra işləməsi üçün, aşağıdakı sətirləri fayla əlavə edirik(Görünən qaydalarda NAT edilir, istənilən istifadəçinin **80** və **443**-cü portlara etdiyi müraciətləri daxili WEB serverimizə yönləndirilir ki, qeydiyyatdan keçsinlər və uğurlu halda **400**-dən başlayaraq qayda əlavə ediləcək):

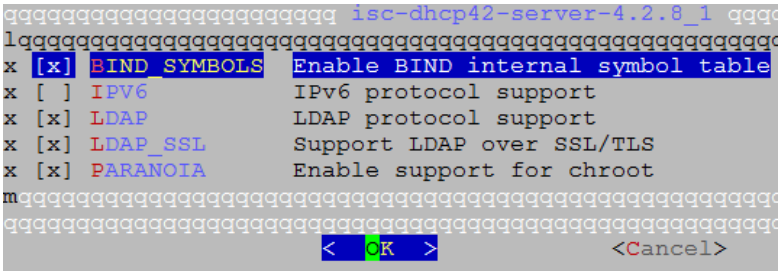
```
ipfw add 00200 divert 8668 ip from any to any via em0
ipfw add 10800 allow ip from any to 85.132.57.58
ipfw add 10900 allow ip from 85.132.57.58 to any
ipfw add 11000 allow ip from any to 85.132.57.59
ipfw add 12000 allow ip from 85.132.57.59 to any
ipfw add 60000 fwd 10.10.10.1,80 tcp from any to any dst-port 80 via em1
ipfw add 60001 fwd 10.10.10.1,443 tcp from any to any dst-port 443 via em1
ipfw add 65000 allow ip from any to any
ipfw add 65535 deny ip from any to any
```

Sistemimizin kernel minimallaşdırılmış və aşağıdakı opsiyalar əlavə edilib kompilyasiya edilmişdir:

```
options         IPDIVERT
options         IPFIREWALL
options         IPFIREWALL_VERBOSE
options         IPFIREWALL_VERBOSE_LIMIT=3
options         DUMMYNET
options         IPFIREWALL_FORWARD
options         IPFIREWALL_NAT
options         LIBALIAS
```

Portlardan DHCP-ni yükləyirik:

```
# cd /usr/ports/net/isc-dhcp42-server/     - Port ünvanına daxil oluruq.
# make config                              - Tələb edilən modulları seçirik
```




```
# make -DBATCH install          - Yükləyirik
```

/usr/local/etc/dhcpd.conf quraşdırma faylının tərkibini aşağıdakı şəklə gətiririk:

```
option domain-name "wifiofis.domain.az";
option domain-name-servers ns1.domain.az, ns2.domain.az;
default-lease-time 3600;
max-lease-time 86400;
ddns-update-style none;
authoritative;
subnet 10.10.10.0 netmask 255.255.255.0 {
        range 10.10.10.26 10.10.10.254;
        option routers 10.10.10.1;
}
# Access Pointləri şəkildəki İP ünvanlara görə rezerv edirik
host Wifi-1f.1 {
        hardware ethernet 04:18:76:8c:9a:a3;
        fixed-address 10.10.10.10;
}
host Wifi-1f.2 {
        hardware ethernet 04:18:66:43:11:11;
        fixed-address 10.10.10.2;
}
host Wifi-1f.3 {
        hardware ethernet 04:18:36:68:a9:9b;
        fixed-address 10.10.10.3;
}
```

DHCP üçün jurnal faylı yaradırıq və Syslog-dan süzgəcdən keçiririk(DHCP arenda jurnallarına **/var/db/dhcpd/dhcpd.leases** faylında baxa bilərsiniz):

```
# touch /var/log/dhcp.log
```

/etc/syslog.conf faylının sonuna aşağıdakı sətirləri əlavə edirik:

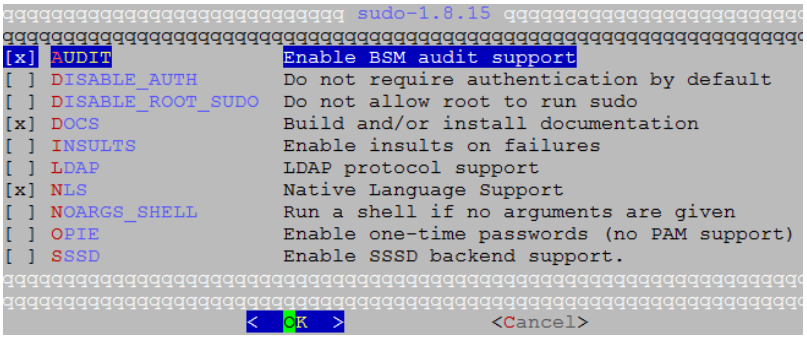
```
!dhcpd
*.*                                             /var/log/dhcp.log
```

DHCP-ni işə salırıq və qulaq asmasını yoxlayırıq:

```
# /usr/local/etc/rc.d/isc-dhcpd start
# sockstat -l|grep dhcp
dhcpd    dhcpd      4695  7  udp4   *:67                  *:*
dhcpd    dhcpd      4695  20 udp4   *:8997                *:*
```

Portlardan sudo-nu yükləyirik:

```
# cd `whereis sudo | awk '{ print $2 }'`   - Port ünvanına daxil oluruq
# make config                              - Tələb edilən modulları seçirik
```




```
# make -DBATCH install                     - Yükləyirik
```

/usr/local/etc/sudoers faylına aşağıdakı sətiri əlavə edirik(Bu web serverimizin firewall-a yetki alması üçün tələb edilir):

```
jamal ALL=(ALL) NOPASSWD: SETENV: ALL
```

/usr/local/www/wifi/config.php quraşdırma faylının tərkibi aşağıdakı kimi olacaq(istifadəçilərin qeydiyyatı və dayandırılması, İPFW qaydaların əlavə edilməsi/silinməsi vasitəsi ilə yerinə yetirilir):

```
# cat /usr/local/www/wifi/config.php
<?php

define('DEBUG', true);

define('conf_DB_HOST', 'localhost');            //Bazanin IP-si
define('conf_DB_USER', 'wifidbuser');           //Bazanin istifadeci adi
define('conf_DB_PASS', 'wifidbpassword');       //Bazanin shifresi
define('conf_DB_NAME', 'wifi');                 //Bazanin adi
define('RULE_NUM_MIN', 400);
define('RULE_NUM_MAX', 600);

define('CLIENTS_IP_BEGIN', '10.10.10.26'); // Mushterilen hansi IP unvandan bashlayacaqlar
define('CLIENTS_IP_COUNT', '200');
```

```
define('CLIENTS_TIME', '3600'); // Mushterinin Internet istifade ede bileceyi
zaman (1 saat)

define('RULE_ADD_IP', 'sudo ipfw add %s allow ip from any to %s');
define('RULE_ADD_IP2', 'sudo ipfw add %s allow ip from %s to any');
define('RULE_DEL_IP', 'sudo ipfw del %s');
define('RULE_DEL_IP2', 'sudo ipfw del %s');

/*
        STATUS:
        0 - Qoshulmanin melumati, duzdurse qoshulmusuz, eks halda qayda elave
edilmedi!
        1 - Artiq qoshulmusunuz
        2 - Istifadeci adi artiq istifade edilmishdir
        3 - Istifadeci dondurulmushdur
*/

$db_link = mysql_connect(conf_DB_HOST, conf_DB_USER, conf_DB_PASS);

if (!$db_link) return cms_errors('Verilenler bazasina qoshulmaq mumkun
olmadi!');

if (!mysql_select_db(conf_DB_NAME, $db_link)) return cms_errors('Verilenler
bazasina qoshulmaq mumkun olmadi!!!');

function cms_errors($text)
{
        if (DEBUG) echo $text;
        return false;
}

function dumpVarX(&$Var, $Var_s = null)
{
        echo "<div align='left' class='debug'>";
        dumpVar($Var, 0, $Var_s);
        echo "<div>";
        return true;
}

function dumpVar(&$Var, $Level = 0, $Var_s = null)
{
        if ($Level > 4)
        {
                echo "<b>...</b> LEVEL > 4<br>\n";
                return;
        }
        $is_ob_ar = false;
        $Type = gettype($Var);
        if (is_array($Var)) {$is_ob_ar = true; $Type =
"Array[".count($Var)."]";}
        if (is_object($Var)) $is_ob_ar = true;
        if ($Level == 0)
```

```
        {
                if ($Var_s) echo "\n<br>\n<b><span
style=\"color:#ff0000\">".$Var_s." = {</span></b>";
                if ($is_ob_ar && count($Var)) echo "<pre>\n";
                else echo "\n<tt>";
                $Level_zero = 0;
        }
        if ($is_ob_ar)
        {
                echo "<span style=\"color:#05a209\">$Type</span>\n";
                for (Reset($Var), $Level++; list($k, $v)=each($Var);)
                {
                        if (is_array($v) && $k==="GLOBALS") continue;
                        for ($i=0; $i<$Level*3; $i++) echo " ";
                        echo "<b>".HtmlSpecialChars($k)."</b> => ";
                        dumpVar($v, $Level);
                }
        }
        else
        {
                if (is_string($Var) && strlen($Var)>400)
                        echo '('.$Type.') <span style="color:#35BBFA">strlen
= '.strlen($Var).'</span>'."\n";
                else echo '('.$Type.') "<span
style="color:#0000FF">',HtmlSpecialChars($Var),'</span>"'."\n";
        }
        if (isset($Level_zero))
        {
                if ($is_ob_ar && count($Var)) echo "</pre>";
                else echo "</tt>";
                if ($Var_s) echo "<b><span
style=\"color:#ff0000\">}</span></b><br>\n";
        }
        return true;
}

?>
```

İstifadəçilərin qeydiyyatı skripti yəni **/usr/local/www/wifi/add.php** faylı aşağıdakı kimi olacaq:

```
# cat /usr/local/www/wifi/add.php
<?php

require_once('config.php');

$user = get_user($_GET['login'], $_GET['pass']);

if ($user)
{
        switch ($user['status'])
        {
                case 0:
```

```
                        if (add_rule($user)) echo '<h2>Siz
qoshulmusunuz!</h2>';
                                else echo 'Yalnish qayda elave edilmedi!';
                        break;
                case 1: echo '<h2>Siz artiq qoshulmusunuz</h2>'; break;
                case 2: echo '<h2>Username artiq istifade edilmishdir</h2>';
break;
                case 2: echo '<h2>Istifadeci adi dondurulmushdur</h2>';
break;
                default: echo 'Error'; break;
        }
} else echo '<h2>istifadeci/shifre yalnishdir!</h2>';

// Qeydiyyat

function get_user($login, $pass)
{
        $user = null;
        if (!$login || !$pass) return null;
        $login = addslashes($login);
        $sql = 'SELECT * FROM users WHERE username="'.$login.'" AND
password="'.$pass.'" LIMIT 1';
        $res = mysql_query($sql);
        if ($res) $user = mysql_fetch_assoc($res);
        return $user;
}

// Qaydanin elave edilmesi

function add_rule($user)
{
        $user_ip = $_SERVER['REMOTE_ADDR'];
        $current_date = time();

        if (!checkip($user_ip)) return false;
        $temp = 0;
        $sql = 'SELECT rule_num FROM users WHERE status=1 ORDER BY rule_num';

        $res = mysql_query($sql);
        if ($res)
        {
                $t = mysql_fetch_array($res);
                if (!$t) $rule_num = RULE_NUM_MIN;
                else {
                        while ($temp = mysql_fetch_array($res))
                        {
                                if (($t[0]+1) < $temp[0]) break;
                                $t = $temp;
                        }
                        if ($t[0] < RULE_NUM_MAX) $rule_num = $t[0]+1; else
return false;
                }
        } else return false;
```

```
        $command = sprintf(RULE_ADD_IP, $rule_num, $user_ip);
        exec($command);

        $command2 = sprintf(RULE_ADD_IP2, $rule_num+100, $user_ip);
        exec($command2);

        $sql = 'UPDATE users SET status=1, time_begin=NOW(),
rule_num='.$rule_num.' WHERE id='.$user['id'];
        mysql_query($sql);

        return true;
}

function checkip($ip)
{
        if (!$ip) return false;
        $user_ip = explode('.', $ip);
        $check_ip = explode('.', CLIENTS_IP_BEGIN);
        if (($check_ip[0] != $user_ip[0]) && $check_ip[0] != "*") return
false;
        if (($check_ip[1] != $user_ip[1]) && $check_ip[1] != "*") return
false;
        if (($check_ip[2] != $user_ip[2]) && $check_ip[2] != "*") return
false;
        if (!(($check_ip[3] <= $user_ip[3] && ($check_ip[3] +
CLIENTS_IP_COUNT) >= $user_ip[3])) && $check_ip[3] != "*") return false;
        return true;
}
?>
```

Vaxtın bitməsinə görə istifadəçinin bağlanışı skripti(Yəni /usr/local/www/wifi/cron.php faylı):

```
# cat /usr/local/www/wifi/cron.php
<?php

require_once('config.php');

check_users();

function check_users()
{
        $sql = 'SELECT * FROM users WHERE status=1 AND time_begin > 0 AND
(TIME_TO_SEC(TIMEDIFF(NOW(), time_begin)) > '.CLIENTS_TIME.')';
        $res = mysql_query($sql);
        if ($res)
        {
                while ($user = mysql_fetch_assoc($res))
                {
                        $command = sprintf(RULE_DEL_IP, $user['rule_num']);
                        exec($command);
                        $command2 = sprintf(RULE_DEL_IP2,
$user['rule_num']+100);
```

```
                        exec($command2);
                        $sql = 'UPDATE users SET status=2, time_end=NOW()
WHERE id='.$user['id'];
                        mysql_query($sql);
                }
        }
        return true;
}

?>
```

Qeyd: Fayllarda olan hərflərin tam Azərbaycan dilində görsənməsini istəsəniz, İnternetdə "**azerbaijan html unicode characters**" başlığı ilə axtarış edib, simvolların kodlraını tapa bilərsiniz. Məsələn: http://usefulwebtool.com/en/characters_azerbaijani.php

İstifadəçi adı və şifrənin daxil edilməsi üçün forma(Yəni **index.html** faylı):

```
# cat /usr/local/www/wifi/index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Inzibachtiliq</title>
        <link rel="stylesheet" type="text/css" href="admin.css" />

        <!--[if lt IE 7]><link rel="stylesheet" type="text/css" href="style-
ie.css" /><![endif]-->
</head>

<body>

<div class="login">

<div class="form">
<form method="get" action="add.php">
        <p><label>Login:</label><input class="text" name="login" type="text"
size="17"/></p>
        <p><label>Parol:</label><input class="text" name="pass"
type="password" size="16"/></p>
        <p><input class="submit" type="submit" value="Her shey doqrudur!"/>
</form>
</div>
<div class="rules">
        <h1>Wi-Fi istifadesi qaydasi</h1>
        <ol>
                <li>Qonaqlar ucun WiFi odenishsizdir!</li>
                <li>Reseption-a yaxinlashin</li>
                <li>Istifadeci adi ve shifre alib</li>
                <li>WiFi-dan yararlanin</li>
        </ol>

</div>
```

```
</div>
</body>
</html>
```

Admin paneli **/usr/local/www/wifi/admin** qovluğunda olacaq. Təhlükəsizlik üçün həmin qovluğu **htpasswd** ilə qorumanız lazımdır.
/usr/local/www/wifi/admin/admin.php faylının tərkibi aşağıdakı kimi olacaq:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
        <head>
        <title>Inzibatci.paneli</title>
        <link type="text/css" rel="stylesheet" href="style.css">
        </head>
<body>
<form method="post" action="admin.php">
        Istifadecilerin sayi: <input type="text" value="" name="num" size=2>
eded.<br><br>
        <input type="submit" value="Generasiya et">
</form><hr>

<?php
        require_once('/usr/local/www/wifi/config.php');
        $n = (int) $_POST['num'];
        if ($n > 10) { echo 'Yaradila bilecek istifadeci sayi heddi
ashilmishdir!<br><br>'; $n=0; }
  function generate_password($number=10)
  {
    $arr = array('1','2','3','4','5','6',
                 '7','8','9','0');
    // Shifre generasiya edirik
    $pass = "";
    for($i = 0; $i < $number; $i++)
    {
      // Massivin tesadufi indeksini hesablayiriq
      $index = rand(0, count($arr) - 1);
      $pass .= $arr[$index];
    }
    return $pass;
  }

        for ($i=0; $i<$n; $i++)
        {
                $login = generate_password(4);
                $pass = generate_password(6);
                $sql = 'INSERT INTO users (username, password, status,
rule_num) VALUES("apt'.$login.'", "'.$pass.'", 0, 0)';
                $res = mysql_query($sql);
        }
        if ($res) echo 'Sayda istifadeci <b>'.$n.'</b> ed. elave
edilmishdir.<br><br>';
```

```
        $sql = 'SELECT * FROM users';
        $res = mysql_query($sql);
        echo '<table
width=\'30%\'><td><b>Ad</b></td><td><b>Shifre</b></td><td><b>Status</b></td><
td><b>Qayda</b></td>';
        while ($data = mysql_fetch_assoc($res))
        {
                echo '<tr>';
                echo '<td>'.$data['username'].'</td>';
                echo '<td>'.$data['password'].'</td>';
                if ($data['status'] == 0) { echo '<td class=\'blue\'>Aktiv
deyil</td>'; }
                if ($data['status'] == 1) { echo '<td
class=\'green\'>Istifade edilir</td>';
                echo '<td>'.$data['rule_num'].'</td>';}
                if ($data['status'] == 2) { echo '<td class=\'reds\'>Istifade
edilmishdir</td>'; }
                if ($data['status'] == 3) { echo
'<td><b>Durdurulmushdur</b></td>'; }
                echo '</tr>';
        }
        echo '</table>';
?>

</body>
</html>
```

/usr/local/www/wifi/admin/style.css faylını tərkibi aşağıdakı kimi olacaq:

```
.reds {color:#f30;}
.blue {color:#0000cc;}
.green {color:#0f0;}
```

/usr/local/www/wifi qovluğunda **.htaccess** adlı bir fayl yaradırıq və məzmununa aşağıdakı sətirləri əlavə edirik.

```
AuthUserFile /usr/local/www/wifi/.htpasswd
AuthName "Soft Admin"
AuthType Basic
Require valid-user
```

/usr/local/www/wifi ünvanında istifadəçi adı ilə şifrəni yaradırıq.

```
htpasswd -bc .htpasswd admin freebsd
```

- **.htpasswd** faylina **admin** istifadəçi adını **freebsd** şifrəsi ile yaz
-b - command line-dan istifadəçi adı ve şifrəni gotur.
-c – göstərilən faylı yarat və ona daxil et(eger varsa silib yeniden yazacaq)

CRON skriptimizin istifadəçi limitlərinin yoxlanılması üçün 1 dəqiqədən bir işə salmaq məqsədilə **/etc/crontab** faylına əlavə edib, daemon-u yenindən işə salırıq:

```
# echo "*/1    *    *    *    *    root    /usr/local/bin/php
/usr/local/www/wifi/cron.php" >> /etc/crontab

# /etc/rc.d/cron restart
```

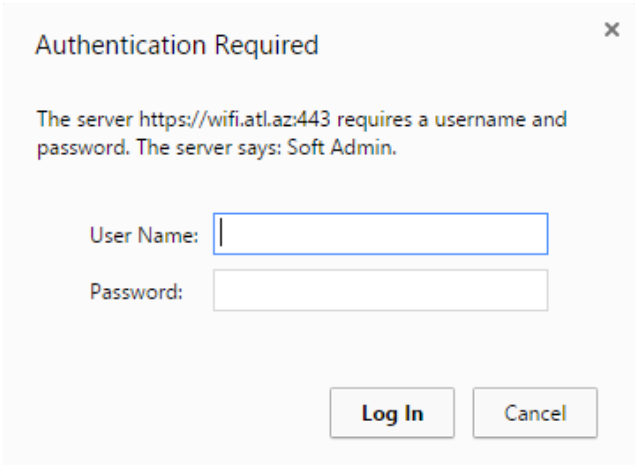
/usr/local/www/wifi/admin.css faylının tərkibi aşağıdakı kimi olacaq(Bu fayl arxa fonda olan şəkilləri təyin edir. Şəkillər isə **/usr/local/www/wifi/img/** qovluğundan oxunur. Şəkillər **/usr/local/www/wifi/img/gp.gif** və **/usr/local/www/wifi/img/logo.png** fayllarıdır. Siz bu şəkilləri istədiyinizə dəyişə bilərsiniz):

```
.login {width:800px; height:540px; position:absolute; left:50%; top:50%;
margin:-250px 0 0 -400px; border:dashed 1px #ddd;
background:url(img/logo.png) 30px 30px no-repeat #fff;}
.login .form {margin:120px 0 0 450px;}
.login .form p {position:relative; margin:0 0 30px 0;}
.login .form label {font:normal 18px arial; position:absolute; margin:3px 0 0
0; color:#aaa;}
.login .form input {margin:0 0 0 100px; padding:2px; font:normal 18px arial;}
.login .form input.text {border-right:solid 1px #ccc; border-bottom:solid 1px
#ccc; border-left:solid 1px #888; border-top:solid 1px #888;}
.login .rules {padding:10px 20px; margin:50px 30px;
background:url(img/gp.gif) 420px 20px no-repeat #ececec;}
h1 {margin:10px 0; font:normal 20px tahoma; color:#c00;}
ol {margin:20px 0 0 30px; padding:0;}
ol li {margin:0 0 10px 15px; font: normal 16px arial; }
```

Bütün qovluqda olan yetkilər yeniləyirik ki, yeni fayllara da mənimsədilsin:

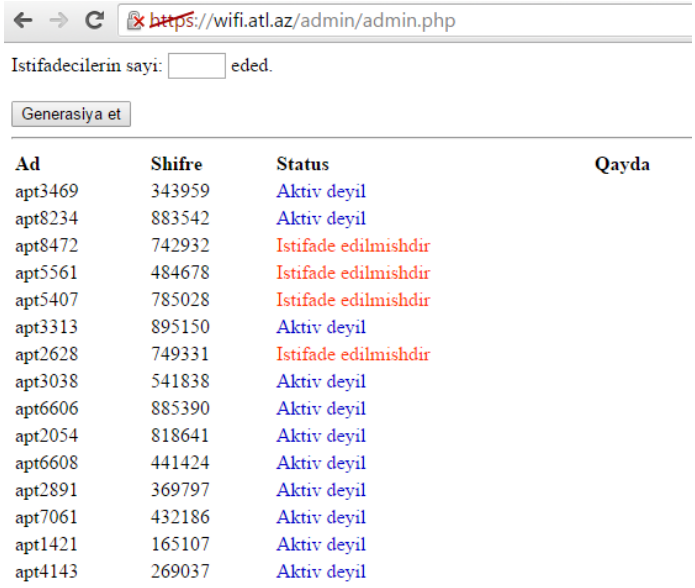
```
# chown -R jamal:jamal /usr/local/www/wifi/
```

Səhifəyə ilk Daxil olduğumuzda istifadəçi adı və şifrə istəniləcək(yaratdığımız **admin** istifadəçi adı və şifrəsini daxil edib **Enter** sıxırıq):



Sonda istifadəçi üçün ilk görünən səhifə belə olacaq:

İnzibatçı interfeysi isə aşağıdakı şəkildəki kimi olacaq:



BÖLÜM 5

Daxili və dünya DNS serveri

- **DNS məntiqi**
- **FreeBSD DNS-in Windows Active Directory ilə inteqrasiya edilməsi**

Başlığımız DNS-in dünyada işləmə prinsipini, xırda nəzəriyyələrini və ümumiyyətlə DNS serverlərin bir-birləri ilə neçə əlaqəyə girmələrini açıqlayır. Eynilə bu başlıqda FreeBSD serverin DNS BİND-ı ilə Windows Active Directory arasında əlaqə yaradılacaq.

DNS məntiqi

DNS bazasının individual yazılar olur hansı ki, **RR**(**Resource Records**) adındadır. DNS bazasının individual hissələrinə isə **zone**-lar deyilir. Misal üçün əgər biz 64.223.167.147 IP ünvanlı www.google.com saytını açmaq istəsək, aşağıdaki şəkildə olan ardıcıllıq gedəcək.

Şəkildə göstərildiyi kimi, www.google.com saytına girmək istədiyimiz andaca, DNS server sizə onun IP ünvanına yönləndirəcək. Bu adi halda, əgər DNS server işləmədiyi halda, adın əvəzinə IP ünvanın istifadə edilməsinə oxşayır. Bunu aşağıdaki kimi istifadə edə bilərik:

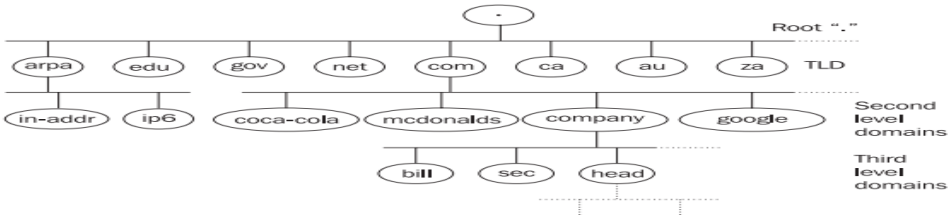
http://64.233.167.147

Yada email yollamaq lazımdır.

izmail@[64.233.167.147]

Domain quruluşu **root**, ikinci dərəcəli və sonrakı alt domainlər quruluşunda gedir. Misal üçün bizim **company.com** adlı ikinci dərəcəli domain-miz var. Həmin domainin billing ilə məşğul olan **bill.company.com** adlı alt domain-i və **sec.company.com** adlı təhlükəsizlik departamenti var.

Ad quruluşu ardıcıllığı aşağıdaki şəkildəki kimi gedəcək:



DNS sistemdə olan ad quruluşu 3 strukturda gedir

Aşağıdaki listdə bəzi **gTLDS**(**Generic Top-Level Domain**) səviyyənin domainlərini açıqlayırıq:

- **.org** domain-i kommersiya xarakteri olmayan ictimaiyyət-ə aiddir.
- **.aero** domain-i yalnız dünya aeroportları üçün rezerv edilmişdir.
- **.biz** domain-i biznes xarakterli işlər üçün rezerv edilmişdir.
- **.coop** domain-i kooperativ birləşmələr üçün nəzərdə tutulmuşdur.
- **.int** domain-i isə ölkələr arasında olan razilaşmalar üçün istifadə edilir.
- **.museum** domain-i isə dünya muzeyləri üçün rezerv edilmişdir.
- **.name** domain-i individual xarakterlə rezerv edilmişdir.
- **.pro** domain-i isə məhdudlaşdırılmışdır və yalnız professional xarakterli məqsədlərdə istifadə edilə bilər.

Name Syntax

Domain adı nöqtələrlə ayrılaraq bir neçə hissyə bölünə bilər. Sadə DNS standartlarına riayət edərək bu ardıcıllığı istədiyiniz qədər davam etdirə bilərsiniz(**abc.head.company.com** bir misaldır). Aşağıdaki misal kimi:

string.string.string ……**string.**

Bütün ad **255** simvoldan çox ola bilməz. Bir subdomain **63** simvoldan çox ola bilməz. Ad hərflərdən, rəqəmlərdən və defis-dən ibarət ola bilər. Defis domain-in əvvəlində və ya axırında ola bilməz. Həmçinin adlarda istfadə edilə biləcək digər spesifik simvollarda movcuddur ancaq, siz bu simvolları istifadə eləməsəniz daha yaxşı olar çünki, bu sumvollar bir çox programlar tərəfindən istifadə edilməyə bilər. Böyük və kiçik simvollar istifadə edilə bilər ancaq, bunu istifadə eləmək çox narahat olacaq. DNS bazasının əsasları ilə düşünsək, misal üçün **newyork.com** adı bazada **NewYork.com** və ya **NEWYORK.COM** kimi saxlana bilər.

Beləliklə ad IP ünvana translyasiya edildikdə, istifadəçi üçün adın böyük və ya kiçik simvollarla daxil edilməsinin fərqi olmur. Ancaq ad bazada böyük və kiçik simvollarla saxlana bilər. Beləliklə əgər biz **DNS** bazasında **NewYork.com** kimi saxlamışıqsa, onda müraciət edilən zaman verilənlər bazası bu adı "**NewYork.com.**" kimi qaytaracaq. Sondaki '.' nöqtə simvolu adın hissəsini göstərir.

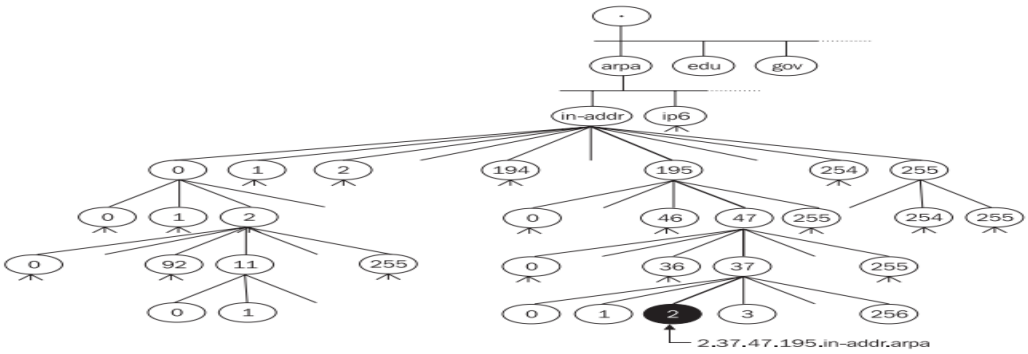
Bəzi hallar ola bilər ki, biz domain-in sağ hissəsini istifadə etmədən istifadə edə bilərik. Bu adətən proqramçıların programlarında istifadə edilir. Domain adlarının verilənlərində bu vəziyyət xeyli çətindir:

- Demək olar ki, əksər hallarda son nöqtəni yazmamaq olar.
- Adətən domain-in sağ tərəfini o halda yazmamaq olur ki, domain-in ortada olan hissəsinin sonu IETF standartında olan ad ilə bitir. Yəni misal üçün, sizdə **DNS** adı **computer.ru.company.com**-dursa siz bu adın əvəzinə **computer.ru** yaza bilərsiniz çünki, hər iki adı son nəticə etibarilə eyni IP ünvana yönləndirmiş olacaqsınız.

Reverse domain-lər

Bəzi program təminatları olur ki, DNS adını IP ünvana əsaslanaraq tapmaq istəyirlər. Bu halda isə biz IP ünvanı ada çevirməliyik. Buna reverse dns yazısı deyilir. IP ünvanın ada şevrilməsinə isə **reverse translation** deyilir.

Domain adlarında olduğu kimi, IP ünvanlarında ağac tipli strukturu olur. IP ünvanlara əsaslanaraq yaradılmış domainlər reverse domainlər adlanır. Pseudo domainlər **in-addr.arpa IPv4** üçün və **IP6.arpa** isə **IPv6** üçündür. Bu domainlərin tarixi açıqlanması var hansı ki, **inverse addresses in the Arpanet** mənasını kəsb edir. in-addr.arpa domain-nin altında təyin etdiyinin IP ünvanın rəqəmi olur. Misal üçün **in-addr.arpa** domain-i üçün **0**-dan **255** aralığınadək subdomainlər. Məsələn əgər bizim **195.47.37.0/24** şəbəkəsi var və bu şəbəkənin subdomain-i **195.in-addr.arpa** olacaq. Və 47.195.in-addr.arpa onun subdomain-dir(Beləliklə sonadək belə gedəcək). Diqqət yetirin ki, burda yazılan SUBDOMAIN-lər, IP ünvan kimi geriyə doğru yazılır. 195.47.37.2 IP ünvanı üçün quruluş aşağıdaki kimi olacaq.



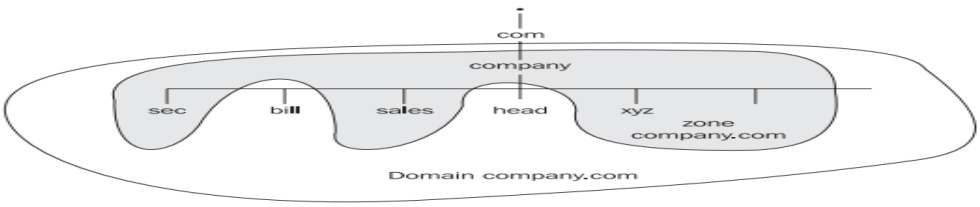
Bütün bu quruluş yalnız IP class **A,B** və ya **C**-də üçün işləyəcək. Bəs sizdə yalnız **C** class-ın özü olsa nə edəcəksiniz? Siz özünüz üçün reverse DNS qaldıra biləcəksinizmi? Bəli qaldıracaqsınız. Baxmayaraq ki, IP ünvan **4** bayt-dan ibarətdir və classic **PTR** subdomain adı **3** ardıcıl rəqəmdən ibarətdir(**4**-cü rəqəmin özü IP ünvanın əlaməti olacaq). Buna görə də **C class**-ı üçün subdomainlər **4** elementlə yazılır. Misal üçün **195.149.150.16/28** şəbəkəsi üçün biz **16.150.149.194.in-addr.arpa** adını istifadə edəcəyik. Bəs əgər IP ünvan 5 bayt-dan ibarət olsa necə olacaq? Düzdür bu DNS qurulduğu andan etibarən səhv fikirləşdirilmişdir. Ancaq sonra bu səhv praktik olaraq qəbul edildi və RFC standartına əlavə edildi. Biz bunu 7-ci başlıqda daha detallı açıqlayacayıq. Siz IPv6-nın reverse yazılışı haqqında 3-cü bölmədə baxacaqsınız.

Domain 0.0.127.in-addr.arpa

127.0.0.1 IP ünvanının maraqlı komplektasiyası vardır. **127** şəbəkəsi hər bir kompüter üçün **LoopBack** adapter kimi **rezerv** edilmişdir. Ancaq bütün digər IP ünvanlar internetdə birmənalı olurlar. Hər bir Name Server tanınmış domainlər üçün avtoritar olmurlar ancaq, **0.0.127.in-addr.arpa** domain-i üçün avtoritardır(primary name server). Unutmayın hətta adi cache-lənmə serveri bu domain üçün avtoritar olur. Windows 2000 özünü elə aparırdı ki, onda elə deyil ancaq, bu hətta bu onun üçün belə çətin deyil.

Zone

Gəlin **company.com** domain-nin istifadəçisini açıqlayaq. Misal üçün deyək ki, domain müəyyən qrup kompüterlər üçün ərazidir. Məhz bu qrupda olan kompüter adlarının sonu **company.com** ilə bitir. Ancaq company.com domain-i çoxlu əraziyə malikdir və özündə 10 ədəd subdomain təşkil edir(**bill.company.com**, **sec.company.com** və **sales.company.com** və.s). Biz bu domain-i özümüzə Name server qaldıraraq, heç kəsdən asılı olmadan administrasiya edə bilərik. Bu domain-in altında istənilən sayda alt domain yarada bilərik. Aşağıdakı şəkildə biz company.com domain altında yaratdığımız alt domainlərin siyahısını açıqlayırıq:



Spesifik Zone-lar

Adi klassik zonalar adi domain və ya subdomainlərdən ibarət olur. Həmçinin DNS realizasiyasında Spesifik Zone-lardan da istifadə edilir. Bunlar aşağıdakilardir:

- **Zone stub:** Bu sadece asılı zonadır hansı ki, özündə hansı domain vəya subdomain-in administrasiya edilə bilməsi üçün name server haqqında informasiyanı təşkil edir(Onda zona üçün **NS** yazılar olur).

Ona gore də **Zone Stub**-da bütün zone məlumatları olmur.

- **Zone cache/hint:** Bu zona-da root name serverlər haqqında məlumat olur(Name server start edilən kimi avtorizasiya edilməyən verilənlər yaddaşın içinə oxunur). Ancaq BIND8 və yeni versiyalarda bu zone üçün ad göstəricisi mövcuddur. Köhnə versiyalarda isə name cache zone istifadə edilirdi. Unutmayın **authoritar root name serverler**-i **noqte** '.' simvolu ilə qeyd edilmişdir.

Reserve edilmiş Domain və Pseudo Domain-lər

Sonra qərara alındı ki, domainlərin digər əraziləri də həmçinin TLD kimi istifadə edilə bilər və bəzi TLD-lər RFC2606-da rezerv edildi.

- Test üçün nəzərdə tutulmuş domainlər
- Sənəd və misalların yaradılması üçün example domain.
- Error statuslarını çağırmaq üçün invalid domain.
- software qayıdışları üçün **localhost** domain-i.

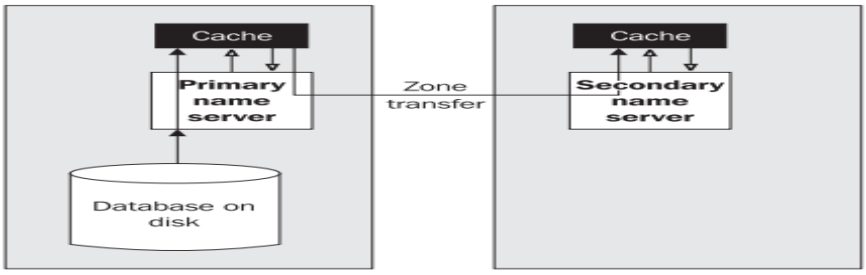
Internetə qoşulmayan hostlarda həmçinin Domain adlarına sahib ola bilərlər. Hətta onlar TCP/IP protokolundan istifadə etməyə bilərlər. Bu halda onlara pseudo domain-lər deyilir. Bunlar böyük əhəmiyyət kəsb edirlər, əsasəndə maillər üçün. Bunun sayəsində Mail vasitəsilə digər şəbəkələrə məlumat ötürmək olur və Internetlə pseudo domain sayəsində edilir(Məsələn **DECnet** yada **MS Exchange**). Kompaniya öz daxili şəbəkəsində öncə TCP/IP, sonra isə DECNet protokolu istifadə edə bilər. Misal üçün (Daniel@computer.company.com) istifadəçi Internet vasitəsilə TCP/IP vasitəsilə ünvanlanır. Bəs DECnet protokolu işləyən kompüter olan istifadəçilərdə necə edəcəksiniz? Bunun üçün biz yalançı dnet adlı ünvan əlavə edirik. Istifadəçi Daniel isə daniel@computer.dnet.company.com adını tapmaq üçün DNS-də təyin edilən mail serverin dnet.company.com domain-ə müraciət edəcək. O isə öz növbəsində DECnet protokolu olan Gateway-e yönləndiriləcək(company.com domain özü). Məhz burdada TCP/IP (SMTP) DECNet-e convert edilir.

Müraciətlər(Translyasiyalar)

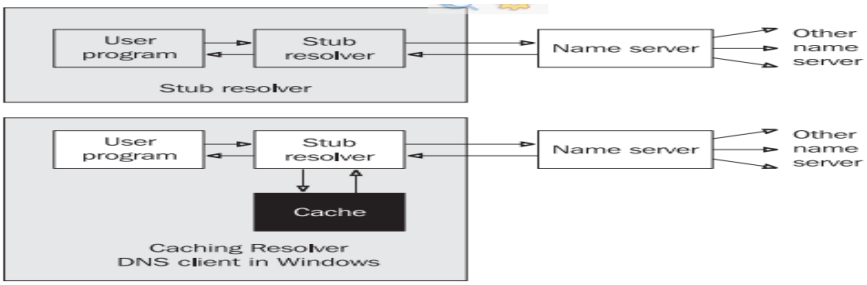
Əksər vacib müraciətlər hostname-i IP ünvana translyasiya edir. Bu məlumatı həmçinin DNS vasitəsilə əldə eləmək olur. Müraciətlər resolver tərəfindən vasitəçilik edir. Resolver isə DNS clientdir və name serveri-i soruşur. DNS bazası bütün dünyada yayımlandığı üçün, yaxın name server-in son cavabı gözləməyə ehtiyacı olmur və o kömək üçün digər serverlərə də həmçinin müraciət yollaya bilər. Name server isə resolver-ə cavab verir və sonra

aldığı cavabı və ya cavabın olmaması haqqında məlumatı ona qaytarır. Bütün mesajlar müraciət və cavablardan ibarət olur.

Name server işə başlayan kimi, zone haqqında məlumatı öz cache yaddaşında axtarır. Primary name server isə datanı daxili diskdən oxuyur, secondary isə edilən müraciət cavabını primary-dən alır və onu öz cache yaddaşına saxlamaqla qənaət edir. Primary və Secondary name serverlərdə saxlanılan informasiyaya avtoriativ data deyilir. Həmçinin name server müəyyən məlumatları öz cache-indən oxuyur hansı ki, bu datalar onun local zonaları haqqında olan məlumatlar deyil və öz daxili diskində saxlanılmır ancaq, izin verir ki, bu verilənlər root name serverlərlə əlaqə saxlaya bilsinlər. Bu dataya qeyri rəsmi verilənlər deyilir. BIND 8,9 versiyasının terminalogiyasında biz onlar haqqında **primary** və **secondary** kimi yox, **master** və **slave** kimi danışırıq. Aşağıdakı şəkildə göstərildiyi qaydada, **Secondary** server **zone transfer data** müraciəti gələn kimi, **Primary** server bu datanı öz daxili diskindən cache-nə yükləyir ki, **Secondary** serverə ötürə bilsin.

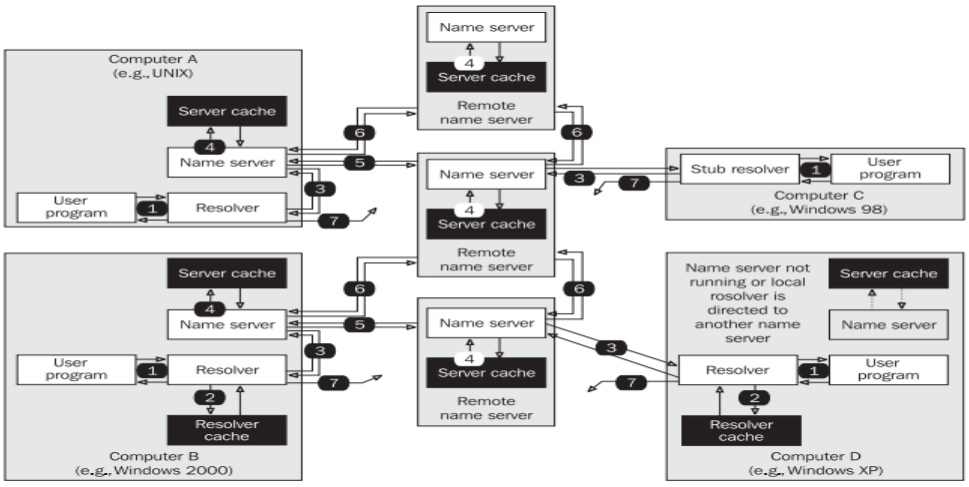


Name serverlər öz cache-lərində pozitiv olan datanı saxlayırlar(bəzi hallarda neqativ olur) ki, onlara gələn real müraciətlərə tez cavab versinlər. Bizim name serverin misalında göstərildiyi kimi bu data digər name serverlərdən alınmışdı və avtoritar deyil. Həmçinin DNS clientlər özləridə öz cache-lərində müəyyən məlumatları saxlayırlar.



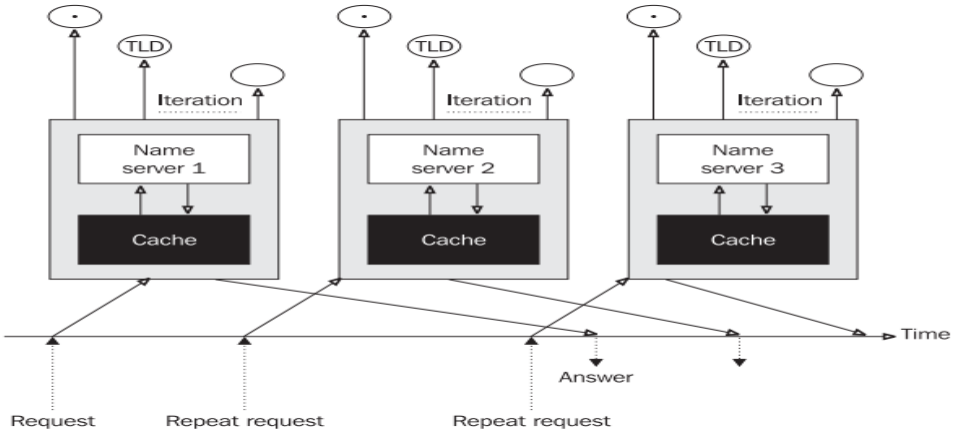
Translyasiya tələbatları istifadəçinin programı tərəfindən tələb edilir. İstifadəçinin programı translyasiya üçün əməliyyat sisteminin komponentindən, yəni resolverindən məlumat alır. Resolver isə transyasiya üçün müraciəti name serverə transfer edir. Kiçik sistemlərdə əksər hallarda ancaq, son resolver olur. Bu hallarda DNS protokolu tərəfindən gələn bütün tələbatları, resolver başqa bir name server işləyən kompyüterin üzərinə yönləndirir. **Cache yaddaşı olmayan resolverə stub resolver** deyilir. Windows maşında buna DNS client deyilir.(Son stub resolverin necə DNS client olmammasında özünüzü caşdırmayın). Bəzi kompüterlər ancaq resolverlerlə işləyirlər(stub yada cachelenmə), digərləri isə həm resolver həmdə name server kimi işləyirlər. Indiki dövrə çoxlu birləşmə metodları mövcuddur ancaq, prinsip eyni olaraq qalır:

1. Istifadəçi əmri yerinə yetirir sonra isə hostname-i IP ünvana translyasiya eləmək lazım olur.
2. Əgər resolver cache-in üzvüdürsə o nəticəni birbaşa almağa çalışacaq.
3. Əgər cavab resolverin cache-inde(yada stub-da) tapılmadısa, resolverlər müraciəti name servere yönləndirəcək.
4. Name server cavab üçün öz cache yaddaşına baxacaq.
5. Əgər name server müraciəti öz cache yaddaşında tapa bilmədisə, o kömək üçün digər name serverlərə müraciət edəcək.
6. Name server lazımi nəticə əldə edənədək kifayət qədər çox name serverlərlə əlaqə quracaq. İş baş verəcək müddətdə name server özüdə həmçinin avtoritar name server ilə əlaqə qurmağa çalışacaq. Avtoritar name server son filter edilmiş cavabını verəcək(əgər edilən müraciətdə qeyri düzgün ad olarsa mənfi cavab qayıdacaq)
7. Əgər öncə yazdığımız əməliyyatda, müraciət cavabı tez müddət ərzində qaytarmazsa, müraciət təkrarlanacaq. Əgər resolver quraşdırmasında 1-dən çox name server göstərilibsə o növbəti müraciəti növbəti name serverə yönləndirəcək. Name serverlərin direktoriyası dövr şəklində işə düşür. **Cycle** name serverin konkret müraciəti ilə başlayır hansı ki, öncə göstərilmişdir.



DNS müraciət/cavabların transport üçün həm **TCP** həmdə **UDP** protokollarından istifadə edir. O hər iki protocol üçün **53**-cü port-u istifadə edir(port **53/UDP** və **53/TCP**). Müraciətlərin çoxu translyasiya vaxtı UDP porotokolunu istifadə edir(Bütün adların IP ünvana və geriyə çevrilməsində). **UDP protokolu** ilə **ötürülən** verilənlərin uzunluğu **512Bayt** ilə **məhdudlaşdırılmışdır**(truncation flagi istifadə edilə bilər hansı ki, qayıdan cavabın **512Bayt**-dan artıq olmayacağını təyin edir və qayıdan cavabın **TCP** ilə olacağı təyin edilir). UDP paketlər 512Bayt ilə limitlənirlər ona görə ki, fragmentasiya böyük həcmli IP diagramlar üçün nəzərdə tutulur. DNS öz növbəsində UDP fragmentasiyanı məntiqli saymır. **Primary** və **Secondary serverlər** arasında baş verən **transpartirovka** isə **TCP** protokolu vasitəsilə həyata keçirilir. Ümumi müraciətlər(hansı ki, adın IP ünvana və geriyə) **UDP protocol datagramları** vasitəsilə həyata keçirilir. Translyasiyalar client(resolver) tərəfindən name serverlərə translyasiya edilir. Əgər name server nə cavab verəcəyini bilməsə o kömək üçün digər name serverlərə müraciət edəcək. Name serverlər bu müraciətin cavabını öz aralarında qərara alırlar hansı ki, adi halda root

name serverlərdən başlayırlar. Aşağıdaki şəkildə translyasiya üçün cavaba tələbat var:



Internetdə bir qayda var hansı ki, verilənlərin olan bazası ən azı iki(asılı olmayan name serverlər) serverdən ibarət olmalıdır ki, biri çökdükdə digəri işləyə bilsin. Ümumiyyətlə biz ümidlənə bilmərik ki, bütün name serverlərə qoşulmaq həmişə mümkün olacaq. Əgər məlumat ötürülməsində TCP protokolu istifadə edilirsə, o halda name serverin qoşulmaq istədiyi serverin özü cavab verməyərsə TCP-nin öz cavablarına əlavə gecikmələr səbəb olacaq. Bu problemin mədəni hell forması UDP protocolundadır. Müraciət datagramı translasiya üçün ilk serverə göndərilir. Əgər edilən müraciət qısa vaxt intervalı ilə qayıtmazsa, onda datagram digər serverə göndəriləcək həmçinin, yenədə cavab qayıtmazsa digər bir serverə yönləndiriləcək(sonadək belə davam edəcək). Əgər bütün mövcud olan serverlərin heç birindən cavab gəlmərsə, dövr ən əvvələ qayıdacaq və cavab yenidən qayıtmazsa, onda timeout baş verəcək.

Round Robin

Bu texnika serverlər arasında yükün bölüşdürülməsi üçün istifadə edilir. Bu halda bizim DNS serverlərimiz üçün bir neçə PUBLIC IP ünvan tələb ediləcək. Misal üçün vacib olan WEB server ola bilər hansı ki, onun dayanıqlıq üçün bir neçə server tələb edilir. Deyək ki, biz WEB server-i 3 maşında işə salmışıq(Məsələn **www.company.com**). Birincisinin IP ünvanı 195.1.1.1, ikincisinin IP ünvanı 195.1.1.2 və üçüncüsünün IP ünvanı 195.1.1.3-dur. DNS Serverimizdə www.company.com üçün 3 yazı olacaq və onların hər birində ayrı IP ünvan olacaq. Round Robin texnikasında cavab aşağıdaki kimi olacaq:

1. Ilk istifadəçi üçün, ilk müraciətdə WEB server üçün qayıdan cavab 195.1.1.1, 195.1.1.2 və 195.1.1.3 cavabını qaytaracaq
2. Ikinci istifadəçi üçün olan növbəti müraciətdə WEB servere aid olan cavab 195.1.1.2, 195.1.1.3 və 195.1.1.1 qayıdacaq
3. Üçüncü istifadəçi üçün olan növbəti müraciətdə WEB Server-ə aid olan cavab 195.1.1.3, 195.1.1.1 və 195.1.1.2 qayıdacaq
4. Bu procedur ilk müraciətdən başlayaraq sonadək eyni formada davam edəcək.

Resolverlər

Resolver sistemin bir hissəsidir hansı ki, IP ünvan transilyasiyası ilə əlaqəlidir. Resolver clientdir ancaq, o konkret program kimi təyin edilmir. O sadəcə olaraq müəyyən bir kitabxana yığmasından ibarətdir hansı ki, **telnet**, **FTP**, **browser**lər və bəzi programların tətbiqində istifadə edilir. Misal üçün əgər telnet programına kompüterin adını IP ünvana çevirmək lazım olsa, o lazımi kitabxanaya müraciət edəcək. Client isə(bizim halda telnet programıdır) kitabxana funksyalarını(**gethostbyname**) çağırır hansı ki, müraciəti formulyasiya edir və onları name serverə oturur. Vaxt məhdudiyyətlərinə də həmçinin baxmaq lazımdır. Həmçinin ola bilər ki, resolver öz ilk müraciətinin cavabını ala bilmədi ancaq, o ikinci müraciətin cavabını ala bildi(server ilk müraciətin cavabını gözlədiyi halda ola bilər ki, o ikinci müraciətin cavabını başqa bir name serverdən aldı. Bu ona görə olur ki, ilk name server müraciətə daha gec cavab verir). Istifadəçi nöqtəyi nəzərdən buna baxdıqda elə gəlir ki, ilk müraciətə cavab qayıtmadı və ikinci müraciətdə buna cavab qayıtdı. Həmçinin UDP protokolun istifadəsi eyni nəticə verə bilər. Gəlin diqqətli olaq ona görə ki, elə hallar ola bilər ki, UDP protokolu istifadə edilir və şəbəkə yüklü olduğuna görə cavab yolda itmişdir.

UNIX OS tipli serverlərdə resolver-in quraşdırılması

Adətən **UNIX OS** tipli maşınlarda resolver faylı **'/etc/resolv.conf'** faylında olur və iki sətiri təşkil edir. Bu sətirlər aşağıdakilardır(ikinci sətir bir neçə dəfə təkrarlana bilər):

```
domain LOCAL_Domain-in_adi
nameserver Name_serverinizin_IP_adresi
```

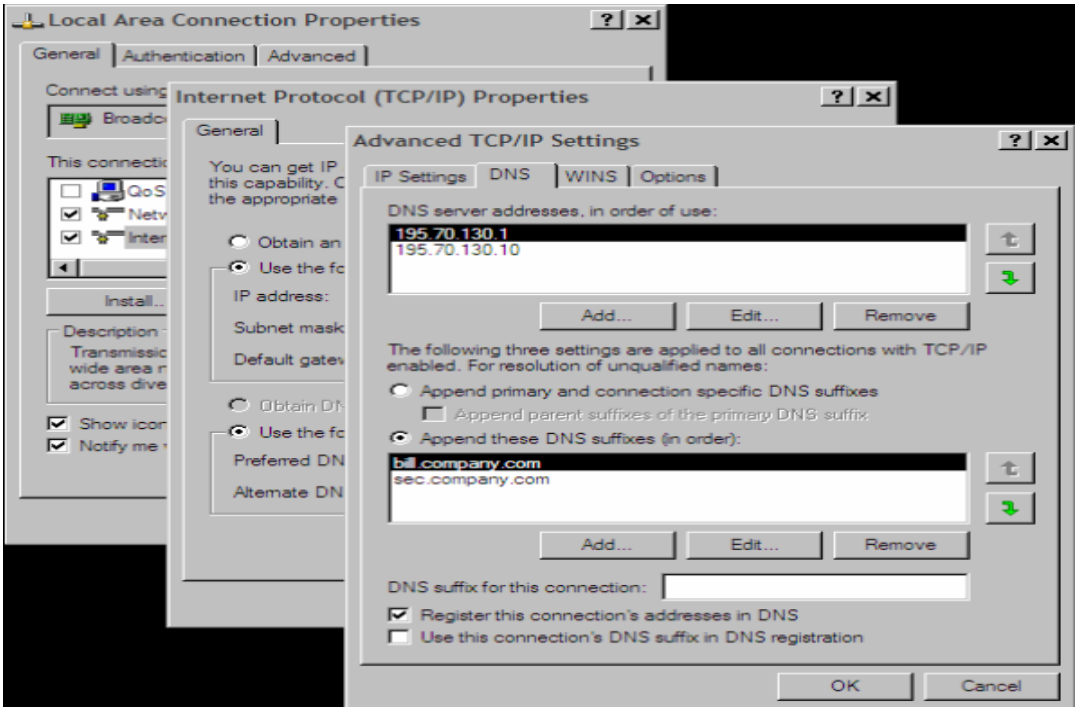
Əgər istifadəçi yazdığı domain-in sonunda nöqtə yazmasada belə, resolver özü həmin domain-in sonuna nöqtə simvolunu əlavə edir və sonra cavabın qayıtması üçün müraciəti name serverə yollayır. Əgər translyasiya yerinə yetirilmədisə(cavab negative olarsa), resolver cavabı suffix olmadan qaytarmağa çalışacaq. Bəzi resolverlər özündə axtarış əmrini aktiv edirlər. Bu əmr sayəsində bir neçə local domain təyin eləmək olar. Name serverlərin IP ünvanları, resolver tərəfindən nameserver əmri ilə təyin edilirlər. Məsləhətdir ki, bir neçə nameserver əmri istifadə edəsiniz çünki, name serverlərdən hansısa biri düşəndə digərinə keçid edə biləsiniz.

Qeyd: Unutmayın ki, resolver faylında nameserver əmrinin qarşısında həmişə IP ünvan təyin edilməlidir. Domain adı yazmaq qəti şəkildə olmaz.

Əgər siz NameServer və **resolver** maşını elə serverin özünü təyin eləmək istəsəniz, onda **resolv.conf** faylında sadəcə **127.0.0.1** nameserver-ni təyin etməniz yetər. Resolverin içində **nameserver**-in sayını limitləmək istəsək isə kernelin parametrini dəyişmək lazımdır. Bu fayl adətən **'/usr/include/resolv.h'** ünvanında olur. Ancaq mümkündür ki, istənilən yeni compu DNS-siz istifadə edəsiniz. Ancaq bu halda lazımi resolv siyahısını **Linux** maşınlarında **'/etc/hosts'** faylında, **Windows** maşınlarında isə **'%System_Root%/System32/Drivers/etc/hosts'** faylında yazmalısınız. Ancaq bu faylda olan təyinatlarla ehtiyatlı olun çünki, siz səhv olaraq real domain adlarını burda qeyd edə bilərsiniz. Həmçinin bütün maşınlar **DNS**-ə müraciət etməzdən öncə ilk olaraq **/etc/hosts** faylına müraciət edirlər.

Windows maşında resolver-in quraşdırılması

Windows maşında siz resolver tərkibini çap etmək üçün **ipconfig /displayDNS** əmrini daxil etməniz yetər. Silmək üçün isə **ipconfig /flushDNS** əmrini daxil etməniz yetər. Ancaq bu əmrdən sonra **%System Root%/System32/Drivers/etc/hosts** faylında olan tərkibin çıxışında heç bir dəyişiklik olmayacaq. **Windows** maşında cache parametrlərini **HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/Dnscache/Parameters** registrində dəyişə bilərsniz. Misal üçün **NegativeCacheTime key** parametri ilə biz negative cavabların cache-də nə qədər müddət qalacağını təyin edə bilərik.



Windows-un köhnə versiyalarında resolver-in quraşdırılması **UNIX** maşınlardakı kimi idi. Yalnız fərq onda idi ki, config text quraşdırma faylında deyildi. Ancaq yeni versiyalarında dahada yeni imkanlar artırıldı. Misal üçün LAN Manager System(NETBIOS-a əsaslanır). Windows **TCP/IP** protocolunu istifadə elədikdə, resolver adı IP ünvana translyasiya eləməyə çalışacaq. **LAN Manager** isə Windows-un özünün ad sistemi kimi qurulub. Və bu **%SystemRoot%/System 32/Drivers/etc/lmhosts** faylından təyin edilir. Sonra isə **Windows DNS** prinsipinə əsaslanan **WINS**(**Windows Internet Name Service**) adlı bir database yaratdı.

1. **LAN Manager cache**-i local kompüter-də saxlayır(**nbtstat -c** əmri cache-i list edir). Bu **NETBIOS** protokolun cache-dir. **LMHOSTS** faylında olan **#PRE** sətirləri parameter olaraq kompüter açılanda **cache**-ə yüklənir. Əgər **LMHOSTS** faylında hansısa dəyişiklik edilərsə biz **nbtstat -R** əmri ilə **cache**-i reload edə bilərik.
2. WINS serverler broadcast vəya LAN ilə multicast-la işləyirlər.
3. lmhosts faylı ilə.
4. Resolver cache-lə.
5. DNS serverlərdə

Həmçinin bəzi programlar ola bilər(Misal üçün **ping** programı) hansı ki, Internet-də axtarışa kömək edə bilər.

1. Resolver cache-də(əgər hosts faylının tərkibi içində oxunarsa)
2. DNS serverlərdə
3. WINS Serverlərdə
4. NETBIOS protocol ilə broadcast yada multicast paketi.
5. lmhosts faylı ilə.

Əgər siz **ping** programı vasitəsilə ada müraciət etdikdə və adın təsadüfən səhv yazdığınız halda Ethereal(program haqqında daha da ətraflı http://www.ethereal.com saytından əldə edə bilərsiniz) programı vasitəsilə NetBIOS-un broadcast edilməsini görə bilərsiniz.
Gəlin indi XP maşının DNS resolver-ni quraşdıraq.
Orda iki imkan mövcuddur:

1. DNS quraşdırmasını təyin elədikdən sonra translyasiya aşağıdaki hallarda baş verir:

- Əgər tələb edilən ad nöqtə ilə bitərsə onda, resolver adı suffix təyin etmədən translyasiya eləməyə çalışacaq.
- Əgər adda nöqtə simvolu olmazsa, o daxil edilən adın sonuna özü nöqtə əlavə edərək resolve etməyə və ya öz Windows domain(hansı ki, Properties-də Computer name-ə görə təyin edilir)-nde axtarmaqa çalışacaq.
- O çalışacaq ki, daxil edilən adı translyasiya etsin hansı ki, özü nöqtə əlavə edib və adda qoşulma üçün DNS suffix zənciri mövcuddur.

2. DNS suffixlərin əlavə edilməsində translyasiya aşağıdakı qaydada yerinə yetirilir:

- Əgər tələb edilən adda nöqtə varsa, resolver suffix əlavə etmədən translyasiya eləməyə çalışacaq.
- O əksər hallarda siyahısına uyğun olan suffixləri əlavə etməyə çalışacaq.

Əgər siz ad daxil etdikdə səhv edərsəniz və mövcud olmayan ad daxil etsəniz, misal üçün xxx, o halda siz ikinci opsiyanı seçmiş olacaqsınız. Onda resolver ilk olaraq **xxx.bill.company.com** adını çevirməyə etməyə və sonra isə **xxx.sec.company.com** adını çevirməyə eləməyə çalışacaq. Hər iki halda o müraciəti **195.70.130.1 IP** ünvanına yönləndirməyə çalışacaq və əgər siz təyin edilmiş vaxt ərzində cavabı almamışsınızsa, o müraciəti **195.70.130.10 IP** ünvanına təkrar edəcək və timeout baş verməyənədək dövr gedəcək.

Name Server

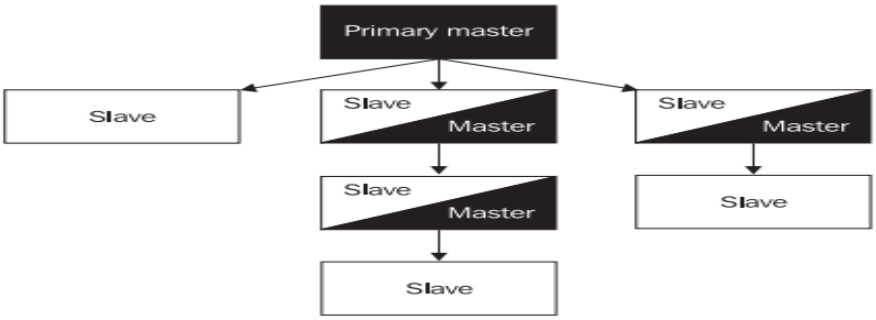
Name server özündə kompüter adlarının IP ünvanlara çevrilməsinin informasiyasını saxlayır(həmçinin IP-nin ada çevrilməsində). Name severlər müəyyən aralıq kompüterlərin hissəsinin adlarını özündə saxlayır. Bu hissəyə zona-lar deyilir(minimum vəziyyətdə o 0.0.127.in-addr.arpa). Domain və ya onun hissələri zone yaradır. Name Server NS tipli yazı ilə təyin edilir. Name server program teminatıdır hansı ki, resolverdən gələn müraciəti başqa bir Name server-ə translyasiya edir. **UNIX** maşınlarda **name server**-in adı **named**

adlanır. Həmçinin **BIND**(**Berkeley Internet Name Domain**) name server kimi istifadə edilir. Name serverlərin bir neçə tipi var və aşağıdaki kimi olur:

- **Primary name server/primary master** zone-a üçün əsas data mərkəzidir. Bu zone-a üçün avtoritativ serverdir. Bu server zone-a haqqında verilənləri öz daxili diskindən əldə edir. Bu tip serverlərin adları BIND-in versiyasından asılı olur. Ona görə ki, primary server adı BIND4.x-da idi, ancaq BIND8-dən sonrakı versiyalarında Primary Master adını almışdır. Administrator bu server üçün verilənləri əllə yaradır. Primary server SOA yazısında təyin edilən domain üçün avtoritar name server kimi təyin edilməlidir. Hər bir zone üçün ancaq bir belə server mövcud olur.
- **Master name server** zone-a üçün avtoritar serverdir. Master server NS yazılarında olan domain üçün həmişə avtoritar server olur. Master server zone-da təyin edilən asılı(**slave/secondary server**) serverlər üçün datanın mənbəsidir. Bu tip serverlər BIND8 və ya yuxarı versiyalarda işləyir.
- **Secondary name server/slave name server** isə müəyyən vaxt intervalı ilə verilənləri əsas **primary name** serverdən alır. Onların üzərində hansısa dəyişiklik etmək ağılsızlıq olacaq ona görə ki, primary serverdə olan növbəti dəyişiklikdən sonra onlar bura nüsxələnəcək və burda etdiyiniz dəyişiklik silinib yenidən yazılacaq. Belə name server həmçinin təyin edilən zone-lar üçün avtoritar sayılır. Bu tip name server BIND4-də başqa cür adlanırdır ancaq, BIND8-dən yuxarı həm Secondary həmdə Slave name server deyilir.
- **Caching-only name server** name server istənilən zone üçün nə Primary nədə Secondary sayılır(avtoritar deyil). Buna baxmayaraq o adi Name Serverin bütün xarakteristikalarını özündə cəmləşdirir. Bütün verilənləri öz cache-ində saxlayır. Bu verilənlərə qeyri rəsmi deyilir. Hər bir server cache-lənmə serveridir ancaq, biz anlayırıq ki, o hansısa bir zone üçün nə Master nədə ki, Slave-dir. (Sözsüz ki, ancaq 0.0.127.inaddr.arpa üçün primary name serverdir ancaq bu sayılmır)

- **Root name server** – root domain üçün avtoritardir(nöqtə üçün). Hər bir root name server Primary-dir hansı ki, özünü digər bütün serverlərdən fərqləndirir.
- **Slave name server** – (**BIND4** versiyasının terminidir)Özünə gələn müraciətləri digər serverlərə ötürür ancaq, özü heç bir müraciətə cavab vermir.
- **Stealth name server** – **secret** serverdir. Bu tip Name server heç bir yerdə elan edilmir. Ancaq özlərində quraşdırmalarında statik IP təyin etmiş tərəflər bilir. Avtoritardır. O zone haqqında məlumatı həmin zonanın ötürülməsinə kömək edərək əldə edir. Bu tip serverlər Name serverin local nüsxəsinin saxlanılması kimi istifadə edilə bilər.

Master/Slave server sxemi aşağıdaki şəkildə göstərilən kimi olacaq:



Eyni Name server həm Master həmdə Slave ola bilər. Məsələn bir zone üçün master və digər zone üçün isə Slave.

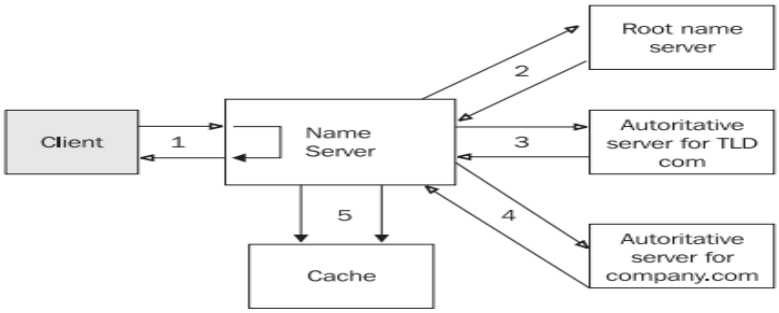
Client tərəfindən baxıldıqda nə **master(primary)** nədəki **slave(secondary)** server arasında heç bir fərq yoxdur. Hər bir önəmli məlumatları özlərində saxlayaraq avtoritar olurlar. Client üçün heç maraqlı olmalı deyil ki, hansı server Master-dir vəya hansı Slave. Digər tərəfdən fikirləşsək isə, cahce-lənmə serverləri avtoritar deyil və əgər o translyasiya eləmək gücünə malik olmazsa, o tələb edilən zone üçün avotoritar serverə müraciət edəcək.

Bu o deməkdir ki, əgər hostmaster öz Master serverində hansısa verilənlərdə dəyişiklik etdisə(Öz bazasına hansısa bir ad əlavə elədi), onda bütün digər slave serverlərdə olan bazalar avtomatik şəkildə dəyişdiriləcək. Bu onların **SOA(resource record)** yazılarında olan vaxt intevalında təyin edilmiş müddətə əsasən sinxronlaşdırılır(Yəni dəyişiklik hostmaster tərəfindən olan kimi, Secondary serverə getmir). Xəta yalnız o halda ola bilər ki, istifadəçi master serverdə edilən dəyişiklik slave gedib çatmazdan öncə, slave serverə müraciət edə bilər. Cavab düzgün olmayacaq çünki, o zaman hələ slave serverin bazasında olan məlumat köhnə olacaq.

Daha pis o halda olacaq ki, əsas server normal işləyir ancaq, təyin edilmiş zone haqqında heç bir məlumat Secondary serverdə yoxdur ona görə ki, zone ötürülməsi uğursuz olmuşdur. Clientlər öz müraciətlərinə cavabları Master və ya Slave serverdən təsadufi alırlar. Əgər client cavabı Master serverdən alacaqsa, bu düzgün olacaq. Əgər client cavabı Slave serverdə alacaqsa bu səhv cavabdır. Ancaq istifadəçi bilmir ki, bunlardan hansı doğru və hansı səhv cavabdır. Onda istifadəçi deyir: **"Birinci dəfə mən müraciətimə cavab aldım amma, ikinci dəfə yox"**

Avtoritar datalar primary master serverin disklərindən qəbul edilir. Qeyri rəsmi informasiya isə şəbəkədə olan digər Name serverlərdə qəbul edilir. Ancaq bir istisna mövcuddur. Name server root name serverləri tanımalıdır ki, dəqiq cavab verə bilsin. Ancaq adi halda bu onlar üçün avtoritar olmur ona görə ki, öncəki kimi hər bir name server, root name serverlər haqqında məlumatlı deyillər. Bu cache serverlər BIND4 və BIND8-də Cace/Hint serverlərdə olur.

abc.company.com domain adına **IP** ünvanın alınması procesinə siz aşağıdaki şəkildə ətraflı formada baxa bilərsiniz:



Ardıcıl olaraq addımları açıqlayaq:

1. Resolver, name serverə gedən tələbləri formulalaşdırır və birmənalı cavab gözləyir. Əgər Name server cavab vermə imkanına malikdirsə, o gözləmədən cavabı yollayacaq. O cavabı öz cache memory-sində axtarır. Avtoritar verilənlər diskin özündən götürülür və həmçinin öncəki ötürmələrdə olan qeyri rəsmi verilənlər. Əgər server cavabı öz cache-ində tapa bilmirsə, o digər serverlərlə əlaqəyə girəcək. Bu həmişə root Name Server ilə başlayır. Əgər Name Server cavabı özündə tapa bilmirsə, o birbaşa root name server ilə əlaqəyə girəcək. Məhz buna görə də hər bir name server, root name serverin IP ünvanlarını bilməlidir. Əgər root name serverə çatmaq mümkün deyilsə(misal üçün əlaqə yalnız localdadırsa), onda bir neçə uğursuz cəhddən sonra bütün proses məhv olacaq.
2. **root name server** isə öz növbəsində, ona gələn müraciətin cavabını yetki verilmiş **NS**(avtoritar nameserver üçün təyin edilən IP ünvan, **.com zone**-sı üçün) yazılarının üzərində **.com TLD**-sində tapır.
3. Bizim name server isə avtoritar server **.com**-a müraciət edir və ondan **company.com** haqqında məlumat əldə edir və görür ki, onun haqqında NS resource record-a burda yetki verilib. Məhz bu server bütün alt domainləri təyin edə bilər.
4. Bizim Name server təyin edir ki, company.com domain-i avtoritardır və bizim müraciətə cavab verir.
5. Serverin vaxtaşırı aldığı informasiya, həmçinin cache-də saxlanılır. Bu tip növbəti müraciət gələrsə, cavab cache-dən qaytarılacaq. Ancaq bu növbəti cavabdır və avtoritar kimi qeydə alınmır.

Name server hətta keçid(abc.company.com-la translyasiya edilən) etdiyi son 5 nöqtənin yolunu belə öz chace-ində saxlayır. Bu yəqin ki, növbəti müraciətlərin gəlişində vaxta qənaət edib onu öz chace-indən oxumaq üçün edilir(həmçinin root name serverlərə də kömək edir). Ancaq sizə cache-də olmayan və TLD-də olan domain adının translyasiyası tələb edilsə, root name serverlərlə əlaqə qurulacaq. Bundan da bizə bəlli olur ki, root name serverlər hər bir halda mütləq şəkildə həmişə PUBLIC şəbəkədə görünməlidir və görünmədiyi halda çox ciddi problemlərə gətirib çıxaracaq.

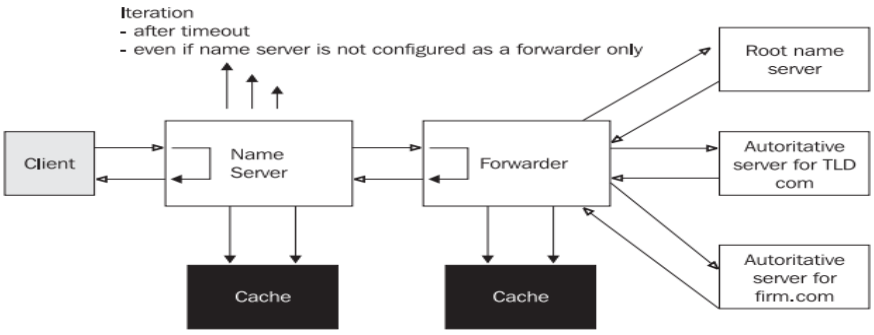
Name serverin tam olan rekursiv cavaba ehtiyacı yoxdur(root Name serverlər və TLD name serverlər). Vacib name serverlərin hətta özünə gələn müraciətlərin rekursiv cavablandırılmasına belə ehtiyac yoxdur. Mütləq vacibdir ki, ona gələn bu tip müraciətlər məhdudlaşdırılsın və yetki kəsilsin. Resolverləri birbaşa bu tip serverlərə yönləndirmək mümkün deyil.

nslookup programı administrator üçün çox vacib utilitlərdən biridir. Həmçinin utilitin istifadəsində də belə öncədən siz recursiya və iterasiyanı söndürməlisiniz ki, heç kəsə artıq müraciət etməyəsiniz. Aşağıdaki qaydada:

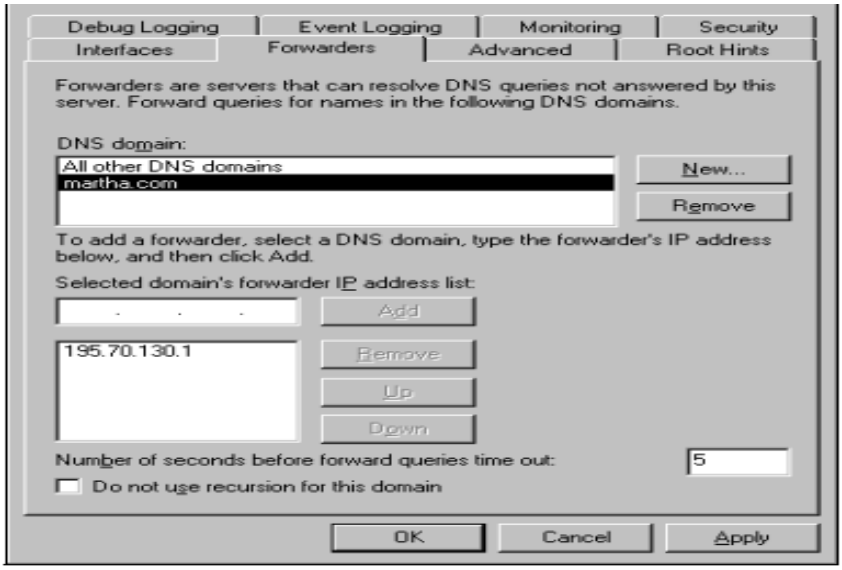
```
# nslookup
 set norecurse
 set nosearch
```

Forwarder Serverləri

Başqa tip serverdə vardır hansı ki, forwarder server adlanır. Bu serverin xarakteristikası istənilən zone-a üçün Primary və ya Secondary server ilə əlaqə qurmaq deyil, ancaq təyin edir ki, hansı DNS müraciətində translyasiya gəlib. Indiki vaxtadək biz danışırdıq ki, resolver ona gələn müraciəti name serverə yollayır(client rekursiv müraciət yollayır və cavabı gözləyir) və son cavab gələnədək gözləyir. Əgər name server cavab verə bilmirsə, o recursive olmayan müraciətlərə rekursiv cavab yollamağa başlayır. Ilk olaraq o root name server ilə əlaqə qurur. root name server resolverə deyir ki, hansı name server bu müraciətə cavab verməlidir. Sonra o məsləhət görülən name serverlə əlaqə yaradır. Bu name server isə internetə çoxlu paketlər yollayır. Əgər sizin şirkətinizdə şəbəkə sürəti azdırsa onda, forwarder name server məntiqini istifadə etməniz kifayətdir. Çünki forwarder sadəcə paketləri başqa serverə yollayır və cavab gözləyir. Aşağıda local name server ilə forwarder name server arasında olan əlaqəni göstəririk:



Local Name server müraciətləri forwarder name servere yollayır. Bu o halda olur ki, local name server gələn müraciətləri rekursiv kimi qeyd edir. Forwarder name server isə öz növbəsində müraciəti local name serverdən alır və bunları qeyri rekursiv müraciətləri kimi Internet üzərindən çıxarır. Bu yalnız bizim name serverə son nəticəni qaytarır. Local name server isə, forwarder name server-dən gələn cavabı son nəticə olaraq gözləyir. Əgər local Name serverdə həmçinin təyin edilən vaxt aralığında cavab verə bilmədisə o root name server ilə əlaqə yaradacaq. Əgər local name serverə root name serverlər ilə əlaqə qurmağa izin verilmirsə və yalnız gözləməyə izin verilirsə, onda quraşdırmada onun yalnız forwarder server olduğunu göstərməliyik. BIND4.x serverlərində buna Slave server deyilirdi. Forwarder-only(slave) daxili şəbəkədə istfadə edilir(FireWall arxasında) hardakı, root name serverlərlə əlaqə saxlamaq mümkün deyil. Forwarder server isə hər iki variantda cache vəziyyətində işləyir və həmçinin zone-lar üçün həm primary həmdə secondary ola bilər. Həmçinin mümkündür ki, Windows 2003 serverin üzərində forwarder server kimi quraşdırmaq olar. Aşağıdaki şəkildə göstərilən kimi:



Sadəcə **Administrative tool**-dan **DNS**-ə daxil olun. DNS serverin üzərində sağ düyməni sıxıb **Properties**-ə daxil olun. Sonra da **Forwarders** düyməsinə sıxın. **New** düyməsinə sıxın və sizə forwarder tərəfindən resolve ediləcək domain adını daxil edin. Siz həmçinin serverlərin forwarder serverdən gələn cavabının gözlənilmə vaxtını belə saniyələrlə təyin edə bilərsiniz. Biz həmçinin slave serverə keçidi Do not use recursion for this domain düyməsini istifadə edərək edə bilərsiniz.

FreeBSD DNS-in Windows Active Directory ilə inteqrasiya edilməsi

Məqsədimiz Windows Active Directory serverdə olan DNS serverin əvəzinə UNİX DNS serverin istifadə edilməsidir. Hal-hazırda UNİX DNS BİND-i Windows Domain Controller ilə inteqrasiya edəcəyik.

Windows 2008 Server
DC Name - example.com
IP address - **192.168.10.10**

Unix DNS Bind9
IP - **192.168.10.100**

```
ee /etc/namedb/named.conf            # Aşağıdakı kontenti Faylın daxilinə
                                     əlavə edirik. Dynamic DNS quraşdırırıq.

zone "example.com" {
        type master;
        check-names ignore;
        allow-query {any;};
        allow-update {192.168.10.10;};
        file "/etc/namedb/dynamic/example.com.zone";
};
zone "10.168.192.in-addr.arpa" {
        type master;
        check-names ignore;
        allow-query {any;};
        allow-update {192.168.10.10;};
        file "/etc/namedb/dynamic/0-168-192.zone";
};


// Mütləq Aşağıdakı sətiri şərh edirik, əks halda example.com işləməyəcək.
//zone "example.com" { type master; file "/etc/namedb/master/empty.db"; };

ee /etc/namedb/dynamic/example.com.zone              # Faylın daxilinə
                                                     Aşağıdakı mətni əlavə
                                                     edirik
$TTL 86400      ; 1 day
@     IN    SOA  ns1.example.com. dnsadmin.example.com. (
                                 22          ; serial
                                 604800      ; refresh (1 week)
                                 86400       ; retry (1 day)
                                 2419200     ; expire (4 weeks)
                                 86400       ; minimum (1 day)
                                 )
@     IN    NS       ns1.example.com.
ns1     IN  A        192.168.10.100


ee /etc/namedb/dynamic/0-168-192.zone          # Faylın daxilinə  Aşağıdakı
                                               mətni əlavə edirik.
```

```
$TTL 86400      ; 1 day
@      IN     SOA  ns1.example.com. dnsadmin.example.com. (
                                4          ; serial
                                604800     ; refresh (1 week)
                                86400      ; retry (1 day)
                                2419200    ; expire (4 weeks)
                                86400      ; minimum (1 day)
                                )
@      IN     NS       ns1.example.com.


touch /var/log/named.log        # DNS üçün jurnal fayl yaradırıq

ee /etc/syslog.conf             # Faylın sonuna Aşağıdakı mətni əlavə edirik.
!named
*.*                                             /var/log/named.log

/etc/rc.d/named restart         # Servisi restart edirik
```

Sonra gedirik Windows 2008 serverə. Unutmayın Windows maşında DC qaldırmazdan öncə, mütləq şəbəkə kartında Primary DNS UNIX IP(**192.168.10.100**) ünvanını yazın. **Start -> run -> dcpromo** daxil edirik.(Yüklənmə müddətində Mütləq **DNS**-dən **quşu götürün**)

Use advanced mode installation(seçirik) -> **Next** -> **Next** -> **Create a new domain in a new forest**
-> **example.com** (**FQDN** of the forest root domain-ə yazırıq) -> **Next** ->
EXAMPLE(**Domain NetBIOS** name yazırıq) -> **Next**
-> **Windows Server 2008 R2**(**Forest functional level**) -> **Next** -> **DNS Server**(**DNS server**-dən seçimi silirik) -> **Next**
-> **Next** -> **DC** üçün **backup pass** yazırıq -> **Next** -> **Next**
DC ad **FQDN** olaraq **example.com** istifadə edirik.

```
tail -f /var/log/named.log      # DNS işə  düşən müddətdə Online olaraq Loglara
                                baxırıq.
'example.com/IN': adding an RR at '_kerberos._tcp.Default-First-Site-
Name._sites.example.com' SRV
'example.com/IN': adding an RR at '_gc._tcp.example.com' SRV
```

BÖLÜM 6

İnternet Resurslarının paylaşdırılması

- **Squid MSLDAP inteqrasiyası**
- **Squid Cluster-in Domain Controller-də external group-larla inteqrasiya edilməsi.**
- **Squid-in debug və troubleshoot edilməsi**
- **Squid başlıqlara görə süzgəc**
- **Windows yenilənməsi**

İstənilən şirkətin daxilində internet resurslarının rəhbərlik tərəfindən təyin edilmiş müəyyən bir siyasəti olur. Bu siyasət fərqli şöbələrə, fərqli quruluşda tətbiq edilir. Həmçinin nəzərə almaq lazımdır ki, resursların hər bir şəxs üçün qeydiyyatı aparılmalıdır. Lazım olarsa, rəhbərlik üçün qrafik hesabatın hazırlanması bacarığı da olmalıdır. Bu başlığımız bütün bu funksionallığı açıqlayır.

Squid MSLDAP inteqrasiyası

Squid3.4 versiya üzərində MSLDAP inteqrasiyası konfiqi aşağıdakı kimi olacaq:

DC: **domain.lan**
Daxil ola biləcək qruplar **DCADM** OU-sunun içində yerləşir. Məhz buna görə də search filterini OU ucun yazmışam.
DC Admin login: **dcadm**
DC pass: **Dcp123@$$**

/usr/local/etc/squid/squid.conf faylımızda authentifikasiya bölümü aşağıdakı kimi olacaq:

```
#  TAG: auth_param
auth_param basic program /usr/local/libexec/squid/basic_ldap_auth -R -b
"dc=bvim,dc=gov,dc=lan" -D "CN=DCADM,CN=Users,DC=domain,DC=lan" -w
"Dcp123@$$" -f sAMAccountName=%s -h bvim.gov.lan
auth_param basic children 5
auth_param basic realm Please insert your Windows credentials to navigate
auth_param basic credentialsttl 1 hour
auth_param basic casesensitive off

external_acl_type ldap_group %LOGIN
/usr/local/libexec/squid/ext_ldap_group_acl -R -b "dc=domain,dc=lan" -D
"CN=DCADM,CN=Users,DC=domain,DC=lan" -w "Dcp123@$$" -f
"(&(objectclass=person)(sAMAccountName=%v)(memberof=cn=%a,OU=Domain
Groups,OU=Domain,DC=domain,DC=lan))" -h domain.lan
```

External qruplar üçün ACL-lərimiz aşağıdakı kimi olacaq:

```
#### Added by Jamal
acl inet_unlimited external ldap_group Proxy_Unlimited
acl inet_limited external ldap_group Proxy_Limited
acl inet_limwyout external ldap_group Proxy_Limited_w_Youtube
acl inet_limwsoc external ldap_group Proxy_Limited_w_Social
acl inet_limwyousoc external ldap_group Proxy_Limited_w_Youtube_Social
acl inet_limwmail external ldap_group Proxy_Limited_w_Mail
acl inet_limwyoumail external ldap_group Proxy_Limited_w_Youtube_Mail
acl inet_lim112 external ldap_group Proxy_Limited_112
```

Qeyd: Unutmayın MSLDAP tərəfdə hər hansısa bir istifadəçinin qrupunu dəyişərsinizsə, ondan sonra mütləq FreeBSD-də **squid -k reconfigure** əmrini daxil etmək lazımdır ki, LDAP-da yenidən axtarış getsin.

Həmçinin unutmayın ki, hətta DC-də olan belə maşınlar internetə giriş üçün öz istifadəçi adlarını və şifrələrini daxil etməlidirlər.

Ümumumiyyətlə squid.conf faylında istifadə etdiyim bütün siyasətə squid qovluğunda baxa bilərsiniz.

Squid Cluster-in Domain Controller-də external group-larla inteqrasiya edilməsi.

Meqsedimiz: Domain-də olan istifadəçilərin internetə girişinin kontrolunu Squid proxy server üzərindən Domain qruplarına görə edilməsidir. Ancaq internetə giriş Domain-də olan konkret seçilmiş qrup istifadəçilərinin müxtəlif yetkiləri ilə olacaq. Yeni bir qrup istifadəçilər müəyyən saytalara baxa və müəyyən şeyləri download edə bilər. Digərləri isə ancaq müəyyən internet səhifələri aça və download edə bilər.

Hər iki maşına aid olan resurslar:
OS: **FreeBSD 9.2 x64**
DC: **domain.lan**
Squid version: **2.7** (Stable)
DC Groups: **inet_full, inet_minimal, inet_mudriyyet**
Users: **full, minimal, mudriyyet, kenarda**

Görünən istifadəçilər uyğun olan qrupların üzvləridir, yəni **full** adlı istifadəçi **inet_full** qrupun, **minimal** adlı istifadəçi **inet_minimal** qrupun, **mudriyyet** adlı istifadəçi **inet_minimal** qrupun üzvüdür və hər biri fərqli yetkiyə malikdir. Ancaq **kenarda** adlı istifadəçi heç bir qrupun üzvü deyil və **Domain Users** qrupunun üzvüdür.

Qeyd: Əgər bu maşınları VmWare-də virtual olaraq istifadə edirsinizsə, sizin CARP ilə bağlı probleminiz çıxacaq. Bunun üçün isə "**FreeBSD_ESXi_CARP**" adlı sənədə müraciət edin və ordakı qaydada quraşdırın ki, hər şey işləsin.

Hər iki maşında **/etc/sysctl.conf** faylına aşağıdakı sətirləri əlavə edirik:

```
security.bsd.see_other_uids=0
kern.corefilə="/root/%N.core"
net.inet.tcp.blackhole=2
net.inet.udp.blackhole=1
net.inet.carp.preempt=1
net.inet.carp.allow=1
net.inet.carp.log=1
net.inet.carp.drop_echoed=1
net.inet.tcp.sendspace=65536
net.inet.tcp.recvspace=65536
```

Hər iki maşının kernel-ni aşağıdakı opsiyalarla kompilyasiya edirik:

```
cd /sys/amd64/conf      # Kernel üçün lazımi ünvana daxil oluruq
                        GENERIC adlı faylın sonuna aşağıdakı sətirləri əlavə
                        edirik:
device          carp    # Əgər iki ədəd Squid server qursaz ki, Cluster
                        edəsiniz onda bu modul lazım olacaq.

# IPFW Firewall
options         IPFIREWALL
options         IPFIREWALL_VERBOSE
```

```
options         IPFIREWALL_VERBOSE_LIMIT=10
options         IPFIREWALL_FORWARD
options         IPDIVERT
options         DUMMYNET
options         IPSTEALTH
options         HZ=1000

## Squid Diskd modulunu CACHE kimi istifadə edəndə aşağıdakı opsiyalar
kerneldə olmalıdır ki, o işləsin.
options         SYSVMSG
options         MSGMNB=8192     # max # of bytes in a queue
options         MSGMNI=40       # number of message queue identifiers
options         MSGSEG=512      # number of message segments per queue
options         MSGSSZ=64       # size of a message segment
options         MSGTQL=2048     # max messages in system



cd /usr/src                     # Kompilyasiya üçün ünvana daxil oluruq
make buildkernel                # Kernel-i kompilyasiya edirik
make installkernel              # Kernel-i yükləyirik
```

Hər iki maşında **/etc/rc.conf** faylına aşağıdakı sətirləri əlavə edirik(Hər iki maşında **IP** və **default gateway** artıq quraşdırılmışdır)

```
hostname="squidthird.domain.lan"
ifkonfiq_em0=" inet 10.70.3.150 netmask 255.255.255.0"
defaultrouter="10.70.3.1"
sshd_enable="YES"

#### Disabled Services ####
# SendMail
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
sendmail_rebuild_aliases="NO"
# SysLog
syslogd_enable="YES"
syslogd_program="/usr/sbin/syslogd"
syslogd_flags="-ss"
ipv6_enable="NO"

#### Local Services ####
tcp_drop_synfin="YES"
icmp_drop_redirects="YES"
gateway_enable="YES"
sshd_enable="YES"
firewall_enable="YES"
firewall_type="UNKNOWN"
firewall_script="/etc/ipfw.conf"
```

```
# CARP Cluster IP üçün
cloned_interfaces="carp0"
ifkonfiq_carp0="up 10.70.3.222/24 vhid 1 pass VeryStr0ngp@$$w0rd"

#### Third party Services ####
atop_enable="YES"                                   # Monitoring üçün
atop_keepdays="30"
atop_interval="5"
mysql_enable="YES"                                  # Hər hal üçün
apache22_enable="YES"                               # Jurnallar üçün
apache22ssl_enable="YES"                            # Jurnallar üçün
samba_enable="YES"                                  # DC-ə qoşulmaq üçün
winbindd_enable="YES"                               # DC-ə qoşulmaq üçün
kerberos5_server_enable="YES"                       # DC istifadəçi və qrupların UID və
                                                        GID vermək üçün
kadmind5_server_enable="YES"
squid_enable="YES"
nrpe2_enable="YES"                                  # NAGIOS monitoring stansiyası üçün
cdpd_enable="YES"                                   # CDP ilə Cisco-nun görməsi üçün
```

Hər iki maşında **/etc/ipfw.conf** faylı aşağıdakı kimi olacaq:

```
ipfw add 11000 deny ip from any to any ipoptions rr
ipfw add 11100 deny ip from any to any ipoptions ts
ipfw add 11200 deny ip from any to any ipoptions lsrr
ipfw add 11300 deny ip from any to any ipoptions ssrr
ipfw add 11400 deny tcp from any to any tcpflags syn,fin
ipfw add 11500 deny tcp from any to any tcpflags syn,rst
ipfw add 11600 reject tcp from any to any tcpflags syn,fin,ack,psh,rst,urg
ipfw add 65000 allow ip from any to any
```

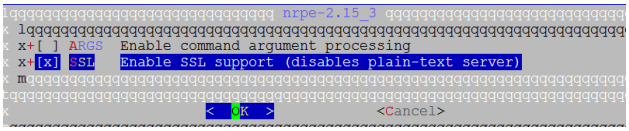
İndi isə Hər iki maşına lazımi paketləri yükləyək:

```
cd /usr/ports/sysutils/atop                 # port ünvanına daxil oluruq
make install                                # Yükləyirik
```

/etc/crontab faylına aşağıdakı sətiri əlavə edirik:

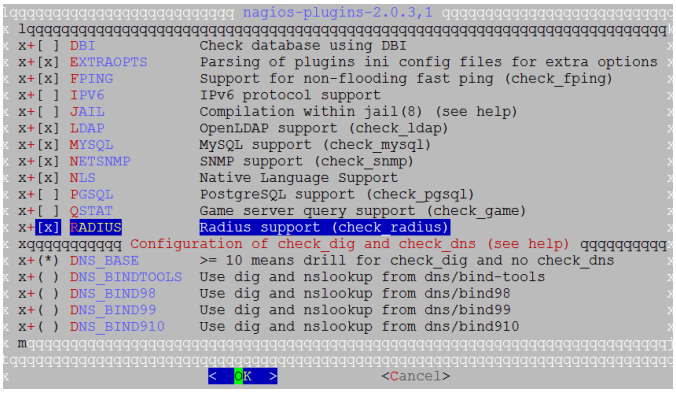
```
# ATOP
0       0       *       *       *       root    /usr/local/etc/rc.d/atop
rotate >/dev/null
```

```
cd /usr/ports/net-mgmt/nrpe                 # NRPE-nin portuna daxil oluruq
make config                                 # lazımi modulları aşağıdakı kimi
                                            seçirik.
```

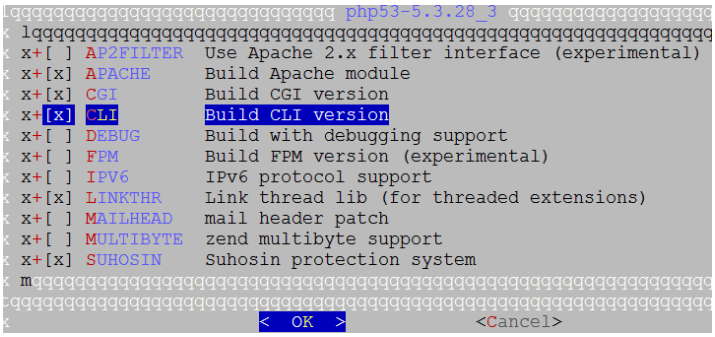



```
make install                                # Yükləyirik
```

nagios pluginlərdən isə aşağıdakı modulları seçirik




```
cd `whereis cdpd | awk '{ print $2 }'`          # CDP portuna daxil oluruq
make install                                    # Yükləyirik


cd `whereis apache22 | awk '{ print $2 }'`      # Apache22-nin portuna daxil
                                                oluruq
make config                                     # Susmaya görə olan modulları
                                                seçirik(SSL olsun)

echo "DEFAULT_VERSIONS+=apache=2.2" >> /etc/make.conf # sistemə elan edirik
                                                ki, apache22 istifadə
                                                edəcəyik
make install                                    # yükləyirik


cd /usr/ports/lang/php53                        # PHP-ni yükləyirik
make config                                     # Lazımi modulları seçirik
```

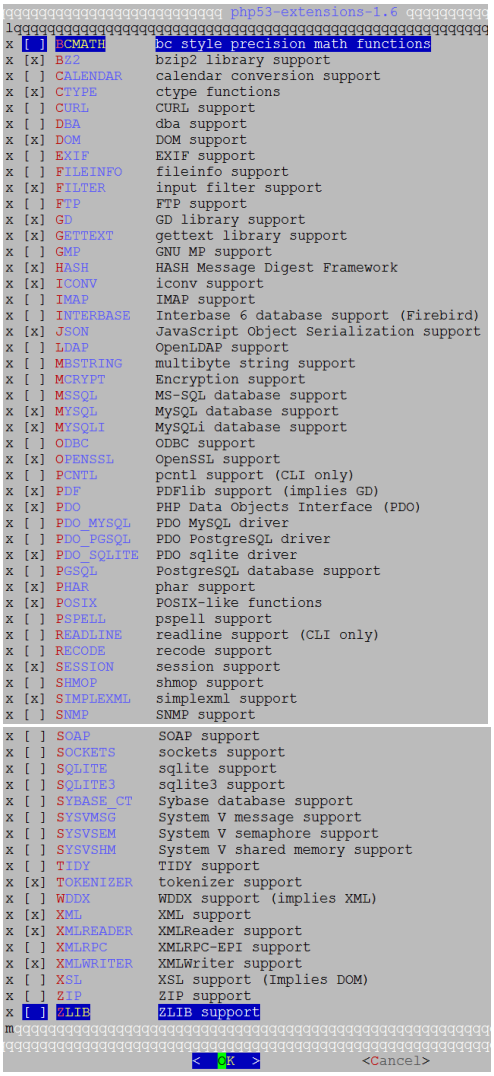



```
make install                                    # Yükləyirik
```

Aşağıdakı sətirləri **/usr/local/etc/apache22/httpd.conf** faylın sonuna əlavə edirik və faylda **DirectoryIndex** bölümünün qarşısına **index.php** əlavə edirik

```
DirectoryIndex index.html index.php             # Bu formada
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps

cat /etc/hosts                  # faylı aşağıdakı formaya gətiririk
127.0.0.1               localhost localhost.my.domain
10.70.3.150             squidthird.domain.lan squidthird
```

```
cd /usr/ports/lang/php53-extensions/ # PHP-nin genişlənmələrini Yükləyirik
make config                          # Lazımi modulları seçirik
```




```
make install                                    # Yükləyirik
```

Konfiqlərimiz üçün ünvanı təyin edirik eynilə **/usr/local/etc/apache22/httpd.conf** faylında **Listen 443** sətiri artırmağı unutmayın.

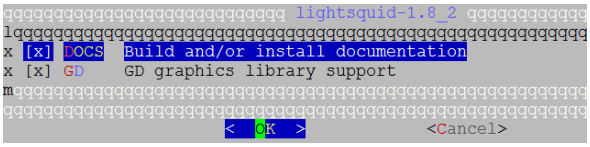
```
echo "Include /usr/local/domen/*" >> /usr/local/etc/apache22/httpd.conf


mkdir /usr/local/domen/     # Vhost-lar üçün qovluq yaradırıq.
```

Jurnallarımız üçün Vhost yaradırıq(sertifikatlarla):

```
cat /usr/local/domen/squidcluster.domain.lan
<VirtualHost *:80>
        RewriteEngine on
        ReWriteCond %{SERVER_PORT} !^443$
        RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R,L]
</VirtualHost>
```

```
<VirtualHost *:443>
        SSLEngine on
        SSLCertificateFilə /usr/local/etc/apache22/ssl/squid.pem
        SSLCertificateKeyFilə /usr/local/etc/apache22/ssl/squid.key
        DocumentRoot /usr/local/www/lightsquid/
<Directory "/usr/local/www/lightsquid">
        AddHandler cgi-script .cgi
        AllowOverride None
        order allow,deny
        Allow from all
        Options FollowSymLinks ExecCGI
        DirectoryIndex index.cgi
        AuthName "Lightsquid Admin Panel"
        AuthType Basic
        AuthUserFilə /etc/htpasswd
        require valid-user
</Directory>
</VirtualHost>

mkdir /usr/local/etc/apache22/ssl/  # Sertifikatlar üçün qovluq yaradırıq

cd /usr/local/etc/apache22/ssl/     # Ünvana daxil oluruq ki, sertifikatı
                                    orda yaradaq.

# Sertifikatı generasiya edirik
openssl req -new -x509 -days 365 -nodes -out squid.pem -keyout squid.key
Generating a 1024 bit RSA private key
......++++++
....++++++
writing new private key to 'squid.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguişəd Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:BAKU
Locality Name (eg, city) []:Yasamal
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FHN
Organizational Unit Name (eg, section) []:Statistika
Common Name (e.g. server FQDN or YOUR name) []:squidcluster.domain.lan
Email Address []:anar.aghayev@fhn.gov.az


Lazımi ünvanlara lazımi yetkiləri verək.
chown -R www:www /usr/local/etc/apache22/ssl/
chmod -R 600 /usr/local/etc/apache22/ssl/
chown -R www:www /usr/local/domen/
```

```
mkdir /usr/local/www/lightsquid/          # Squid jurnalların generasiya
                                          edilməsi üçün istifadə ediləcek
                                          ünvan

cd /usr/ports/www/lightsquid/             # Port ünvanina daxil oluruq
make config                               # Lazımi modulları seçirik
```




```
make install                              # Yükləyirik

chown -R www:www /usr/local/www/          # Lightsquid qovluğunu da www
                                          istifadəçi və qrupun üzvü edirik
```

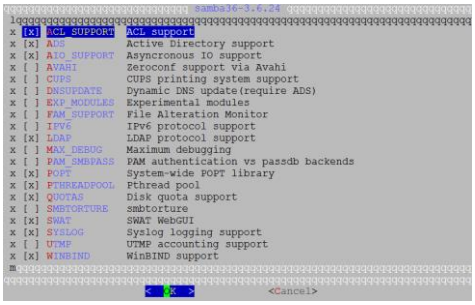
/usr/local/etc/lightsquid/lightsquid.cfg faylında global konfiqləri aşağıdakı hala gətiririk(log ünvanını squidin konfiq faylına uyğun olaraq dəyişin)

```
$cfgpath ="/usr/local/etc/lightsquid";
$tplpath ="/usr/local/www/lightsquid/tpl";
$langpath ="/usr/local/share/lightsquid/lang";
$reportpath ="/usr/local/www/lightsquid/report";
$logpath ="/var/squid/logs";
$ip2namepath ="/usr/local/libexec/lightsquid";
$debug = 0;
$debug2 = 0;
$squidlogtype = 0;
$ip2name="squidauth";
$timereport = 1;
$lang ="ru-koi8";
$templatename ="base";
$showgrouplink = 0;
$userealname = 0;
```

```
/usr/local/www/lightsquid/report          # Lightsquid üçün report qovluq
                                          yaradırıq
/usr/local/www/lightsquid/check-setup.pl  # Scripti işə salaraq LightSQUID-in
                                          konfiq faylının işləməsini test
                                          edirik.

ee /etc/crontab    # İndi isə istifadəçilərin hesabatını açıqlayaq.
                   Məsləhətdir ki,hesabatı Hər yarım saatdan bir edəsiniz və
                   biz onu CRON-a əlavə eləmişik.
*/30  *     *     *     *     root  /usr/local/www/lightsquid/lightparser.pl

root@squidthird:/ # htpasswd -c /etc/htpasswd admin    # Admin şifrəsini
                                                       yaradırıq
New password:
Re-type new password:
Adding password for user admin

/usr/local/etc/rc.d/apache22 restart      # Sonda WEB serveri restart edirik
```

```
cd /usr/ports/databases/mysql55-server/     # MySQL bazanı Yükləyirik
make config                                 # Lazımi modulları seçirik
```

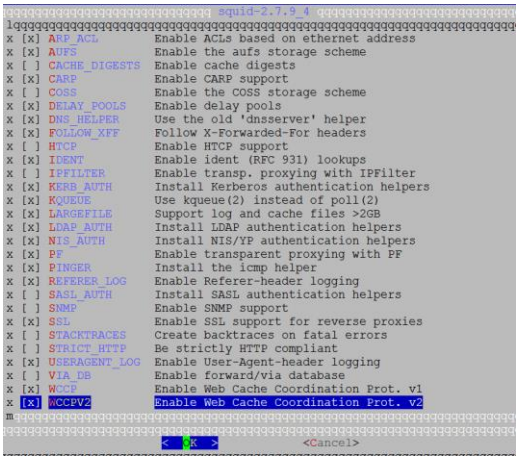

```
make install                                # Yükləyirik

cd /usr/ports/net/samba36                   # SAMBA36 port ünvanına daxil
                                            oluruq
make config                                 # Lazımi modulları seçirik
```




```
make install                                # Yükləyirik


cat /usr/local/etc/smb.conf                 # Serverin quraşdırma faylı
                                            aşağıdakı kimi olacaq
[global]
        workgroup       = DOMAIN
        server string   = Squidprimary Samba
        security        = ADS
        realm           = DOMAIN.LAN
        password server = domain.lan
        netbios name    = squidprimary
        load printers = no
        domain master = no
        local master = no
        preferred master = no
        interfaces = em0
        bind interfaces only = yes
        idmap backend = tdb
        idmap uid       = 10000-20000
        idmap gid       = 10000-20000
        idmap konfiq DOMAIN:backend = rid
        idmap konfiq DOMAIN:range = 10000-99999
        winbind separator = +
        winbind enum users = yes
        winbind enum groups = yes
        winbind use default domain = yes
        winbind nested groups = yes
        winbind refresh tickets = yes
        template homedir = /home/%D/%U
        template shell = /bin/sh
        client use spnego = yes
```

```
        client ntlmv2 auth = yes
        encrypt passwords = yes
        restrict anonymous = 2
        log level       = 10
        log filə        = /var/log/samba/%m.%U.log
        max log size    = 50000


mkdir /var/log/samba/                   # Jurnallar üçün qovluq yaradırıq
mkdir /usr/local/etc/samba              # SAMBA konfiqlər üçün qovluq yaradırıq
mkdir /var/db/samba                     # Samba bazası üçün qovluq yaradırıq


cat /usr/src/crypto/heimdal/krb5.conf     # Kerberos quraşdırma faylını
                                          aşağıdakı kimi edirik
[libdefaults]
        default_realm = DOMAIN.LAN
        clockskew = 300
        v4_instance_resolve = false
        v4_name_convert = {
                host = {
                        rcmd = host
                        ftp = ftp
                }
                plain = {
                        something = something-else
                }
        }

[realms]
        DOMAIN.LAN = {
                kdc = DOMAIN.LAN
                admin_server = DOMAIN.LAN
                kpasswd_server = DOMAIN.LAN

        }
[domain_realm]
        .domain.lan = DOMAIN.LAN


reboot                                  # reboot edirik

ntpdate domain.lan                      # DC-mizdən vaxtı alırıq
kinit -p jamaladm                       # Admin account ilə DC-yə login oluruq
jamaladm@DOMAIN.LAN's Password:

klist                                   # DC-dən aldığımız ticket-ə baxırıq
Credentials cache: FILə:/tmp/krb5cc_0
        Principal: jamaladm@DOMAIN.LAN

  Issued           Expires          Principal
Jul 19 18:31:50  Jul 20 04:31:50  krbtgt/DOMAIN.LAN@DOMAIN.LAN
```

```
cat /etc/nsswitch.conf                # Faylı aşağıdakı şəklə getiririk
group: files winbind
group_compat: nis
hosts: files dns
networks: files
passwd: files winbind
passwd_compat: nis
shells: files
services: compat
services_compat: nis
protocols: files
rpc: files


net join -U jamaladm                  # Artıq admin account ilə DC-ə üzv olduq
Enter jamaladm's password:
Using short domain name -- DOMAIN
Joined 'SQUIDTHIRD' to dns domain 'domain.lan'

net ads testjoin                      # Qoşulmanı test edirik
Join is OK

/usr/local/etc/rc.d/samba restart     # Samba-nı restart edirik ki, WinBind işə
                                      düşsün

wbinfo -u                             # Domain istifadəçilərini list edirik
wbinfo -g                             # Domain qruplarını list edirik

getent passwd                         # DC userlərin UID-nə baxırıq
getent group                          # DC userlərin GID-nə   baxırıq


cd /usr/ports/www/squid               # Squid27 port ünvanına daxil oluruq
make config                           # Lazımi modulları seçirik
```




```
make install                          # Yükləyirik
chmod -R 750 /var/db/samba/winbindd_priviləged/       # Squid üçün SAMBA
                                                       qovluğuna yetki veririk
```

```
chown -R root:squid /var/db/samba/winbindd_privileged/      # Samba qovluğuna
                                                            squid qrupunu
                                                            mənimsədirik
```

/usr/local/etc/squid/squid.conf faylında əsas konfiqlərimizi açıqlayaq(log, cache konfiqlərini istədiyiniz qovluğa təyin ede bilərsiniz, Hər hal üçün **squid.conf** faylı ayrıca hazır olacaq).

```
#  TAG: auth_param
auth_param ntlm program /usr/local/bin/ntlm_auth --helper-protocol=squid-2.5-
ntlmssp --domain=DOMAIN.LAN
auth_param ntlm children 250
auth_param ntlm keep_alive on

auth_param basic program /usr/local/bin/ntlm_auth --helper-protocol=squid-
2.5-basic --domain=DOMAIN.LAN
auth_param basic children 250
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off

external_acl_type nt_group ttl=60 negative_ttl=60 grace=90 children=10 %LOGIN
/usr/local/libexec/squid/wbinfo_group.pl
# TAG: acl bölümündə MIME type-lar üçün ACL təyin edirik
acl deny_mime rep_mime_type -i ^application/octet-stream
acl deny_mime rep_mime_type -i ^application/x-shockwave-flash$
acl deny_mime rep_mime_type -i ^application/octet-stream$
acl deny_mime rep_mime_type -i ^application/x-tar$
acl deny_mime rep_mime_type -i ^application/zip$
acl deny_mime rep_mime_type -i ^application/x-gtar$
acl deny_mime rep_mime_type -i ^application/x-tar$
acl deny_mime rep_mime_type -i ^audio/mpeg$
acl deny_mime rep_mime_type -i ^audio/x-aiff$
acl deny_mime rep_mime_type -i ^audio/x-wav$
acl deny_mime rep_mime_type -i ^audio/mp3$
acl deny_mime rep_mime_type -i ^video/mpeg$
acl deny_mime rep_mime_type -i ^video/quicktime$
acl deny_mime rep_mime_type -i ^video/x-msvideo$
acl deny_mime rep_mime_type -i ^video/x-sgi-movie$
acl deny_mime rep_mime_type -i ^video/vnd.mpegurl$
acl deny_mime rep_mime_type -i ^audio/x-realaudio$
acl deny_mime rep_mime_type -i ^audio/x-pn-realaudio$
acl deny_mime rep_mime_type -i ^application/x-rar-compressed

#### Added by Jamal
acl inet_full external nt_group inet_full
acl inet_minimal external nt_group inet_minimal
#### Birinci ACL DC istifadəçilərinin seçilmiş qrupunu təyin edir ####
#### İkinci isə bu istifadəçiləri həftənin bütün günləri bütün vaxtlarda
təyin edir ####
acl inet_mudriyyet external nt_group inet_mudriyyet
acl inet_mudriyyet_time time MTWHFAS 00:00-23:59
```

```
#### Birinci ACL DC istifadəçilərini təyin edir
#acl inet_mudriyyet proxy_auth
"/usr/local/etc/squid/db/inet_mudriyyet.dcusers"


#### faylda olan root domain-nə giriş qadağandır ####
acl deny_rootdomain dstdom_regex "/usr/local/etc/squid/db/deny_rootdomain"
#### faylda olan terminlər qadağandır ####
acl terminler url_regex -i "/usr/local/etc/squid/db/terminler"
#### faylda olan genişlənmələrdə download etmək qadağandır ####
acl down_deny url_regex "/usr/local/etc/squid/db/down_deny"


#  TAG: http_access - Bu bölümdə isə http_access deny all-dan öncə
aşağıdakıları əlavə edirik
http_access allow localnet inet_mudriyyet !terminler !down_deny
http_access allow all inet_mudriyyet !terminler !down_deny
http_access allow localnet inet_minimal !deny_rootdomain !terminler
!down_deny
http_access allow all inet_minimal !deny_rootdomain !terminler !down_deny
http_access allow localnet inet_full
http_access allow all inet_full
http_access deny all


#  TAG: http_reply_access           - Eynilə reply üçün
http_reply_access allow localnet inet_mudriyyet !terminler !down_deny
http_reply_access allow all inet_mudriyyet !terminler !down_deny
http_reply_access allow localnet inet_minimal !deny_rootdomain !terminler
!down_deny
http_reply_access allow all inet_minimal !deny_rootdomain !terminler
!down_deny
http_reply_access allow localnet inet_full
http_reply_access allow all inet_full
http_reply_access deny all


/usr/local/etc/squid/squid.conf faylı aşağıdakı kimi olacaq:
cat /usr/local/etc/squid/squid.conf | grep -v '^$' | grep -v "#"
auth_param ntlm program /usr/local/bin/ntlm_auth --helper-protocol=squid-2.5-
ntlmssp --domain=DOMAIN.LAN
auth_param ntlm children 250
auth_param ntlm keep_alive on
auth_param basic program /usr/local/bin/ntlm_auth --helper-protocol=squid-
2.5-basic --domain=DOMAIN.LAN
auth_param basic children 250
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
external_acl_type nt_group ttl=60 negative_ttl=60 grace=90 children=10 %LOGIN
/usr/local/libexec/squid/wbinfo_group.pl
acl all src all
acl manager proto cache_object
```

```
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl SSL_ports port 443
acl CONNECT method CONNECT
acl deny_mime rep_mime_type -i ^application/octet-stream
acl deny_mime rep_mime_type -i ^application/x-shockwave-flash$
acl deny_mime rep_mime_type -i ^application/octet-stream$
acl deny_mime rep_mime_type -i ^application/x-tar$
acl deny_mime rep_mime_type -i ^application/zip$
acl deny_mime rep_mime_type -i ^application/x-gtar$
acl deny_mime rep_mime_type -i ^application/x-tar$
acl deny_mime rep_mime_type -i ^audio/mpeg$
acl deny_mime rep_mime_type -i ^audio/x-aiff$
acl deny_mime rep_mime_type -i ^audio/x-wav$
acl deny_mime rep_mime_type -i ^audio/mp3$
acl deny_mime rep_mime_type -i ^video/mpeg$
acl deny_mime rep_mime_type -i ^video/quicktime$
acl deny_mime rep_mime_type -i ^video/x-msvideo$
acl deny_mime rep_mime_type -i ^video/x-sgi-movie$
acl deny_mime rep_mime_type -i ^video/vnd.mpegurl$
acl deny_mime rep_mime_type -i ^audio/x-realaudio$
acl deny_mime rep_mime_type -i ^audio/x-pn-realaudio$
acl deny_mime rep_mime_type -i ^application/x-rar-compressed
acl inet_full external nt_group inet_full
acl inet_minimal external nt_group inet_minimal
acl inet_mudriyyet external nt_group inet_mudriyyet
acl inet_mudriyyet_time time MTWHFAS 00:00-23:59
acl deny_rootdomain dstdom_regex "/usr/local/etc/squid/db/deny_rootdomain"
acl terminler url_regex -i "/usr/local/etc/squid/db/terminler"
acl down_deny url_regex "/usr/local/etc/squid/db/down_deny"
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localnet inet_mudriyyet !terminler !down_deny
http_access allow all inet_mudriyyet !terminler !down_deny
http_access allow localnet inet_minimal !deny_rootdomain !terminler
!down_deny
http_access allow all inet_minimal !deny_rootdomain !terminler !down_deny
http_access allow localnet inet_full
http_access allow all inet_full
http_access deny all
http_reply_access allow localnet inet_mudriyyet !terminler !down_deny
http_reply_access allow all inet_mudriyyet !terminler !down_deny
http_reply_access allow localnet inet_minimal !deny_rootdomain !terminler
!down_deny
http_reply_access allow all inet_minimal !deny_rootdomain !terminler
!down_deny
http_reply_access allow localnet inet_full
http_reply_access allow all inet_full
http_reply_access deny all
icp_access allow localnet
icp_access deny all
```

```
http_port 3128
hierarchy_stoplist cgi-bin ?
cache_mem 256 MB
cache_dir diskd /var/squid/cache 5000 16 512 Q1=72 Q2=64
access_log /var/squid/logs/access.log squid
cache_log /var/squid/logs/cache.log
cache_store_log /var/squid/logs/store.log
mime_table /usr/local/etc/squid/mime.conf
netdb_filəname /var/squid/logs/netdb.state
diskd_program /usr/local/libexec/squid/diskd-daemon
unlinkd_program /usr/local/libexec/squid/unlinkd
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .               0       20%     4320
acl shoutcast rep_header X-HTTP09-First-Line ^ICY.[0-9]
upgrade_http0.9 deny shoutcast
acl apache rep_header Server ^Apache
broken_vary_encoding allow apache
cache_effective_user squid
cache_effective_group squid
delay_pools 3
delay_class 1 2
delay_access 1 allow inet_mudriyyet
delay_access 1 deny all
delay_parameters 1 1048576/1048576 1048576/1048576
error_directory /usr/local/etc/squid/errors/Azerbaijani
cache_dns_program /usr/local/libexec/squid/dnsserver
dns_children 100
hosts_filə /etc/hosts
forwarded_for off
coredump_dir /var/squid/cache
```

Yetki təyin etmek üçün lazım olan qovluq və lazımi faylları yaradib içini dolduraq.

```
mkdir /usr/local/etc/squid/db
root@squidthird:/var/log/samba # cat /usr/local/etc/squid/db/deny_rootdomain
\.am$
\.ru$
\.org$
root@squidthird:/var/log/samba # cat /usr/local/etc/squid/db/terminler
sex
porn
pron
durty
gay
root@squidthird:/var/log/samba # cat /usr/local/etc/squid/db/down_deny
.[Tt][Oo][Rr][Rr][Ee][Nn][Tt]$
.[Aa][Vv][Ii]$
.[Jj][Pp][Ee][Gg]$
.[Zz][Ii][Pp]$
.[Mm][Pp]3$
```

```
.[Ee][Xx][Ee]$

chown -R squid:squid /usr/local/etc/squid/      # Squid qovluğunu squid user
                                                və qrup üzvü edirik

chown -R squid:squid /var/squid/                # Cache və logları squid user
                                                və qrup üzvü edirik

squid -z                                        # Cache generasiya edirik

/usr/local/etc/rc.d/squid start                 # Squid-i işə salırıq
```

Bütün istifadəçilərlə test edirik və uğurlu nəticə əldə edənədək logları analiz edirik.

Squid-in debug və troubleshoot edilməsi

Squid NTLM Group ACL-lər yazılanda əksər hallarda aşağıdakı səhvlər baş verir:

1. Squid DC-yə qoşula bilmir.
2. Squid istifadəçini qrupdan ala bilmir
3. Squid DC ayırıcısını əlavə edə bilmir.

Misal üçün aşağıdakı jurnalı göstərə bilərik:

```
failed to call wbcSidToGid: WBC_ERR_WINBIND_NOT_AVAILABLE
Could not convert sid S-1-5-21-3786744645-3232078785-4224732712-4109 to gid
failed to call wbcGetGroups: WBC_ERR_WINBIND_NOT_AVAILABLE
Could not get groups for user fizuli.ahmedov
could not obtain winbind interface details: WBC_ERR_WINBIND_NOT_AVAILABLE
could not obtain winbind separator!
failed to call wbcLookupName: WBC_ERR_WINBIND_NOT_AVAILABLE
Could not lookup name Internet_Medium_Access
```

```
tail -f /var/log/samba/log.wb-DomainName  # Həmçinin Samba-da olan jurnalları
                                          araşdıraq
```

Hal-hazırda işləyən **/usr/local/etc/smb.conf** faylının məzmunu aşağıdakı kimidir:

```
[global]
        workgroup = DOMAIN
        realm = DOMAIN.LAN
        security = ADS
        encrypt passwords = true
        dns proxy = no
        socket options = TCP_NODELAY
        domain master = no
        local master = no
        preferred master = no
        os level = 0
        domain logons = no
# Mütləq bu sətiri təyin edin əks halda heç nə işləməyəcək çünki digər
trust_domainlər arasında
# Timeout baş verir və siz problemin harda olduğunu anlaya bilmirsiz.
        allow trusted domains = no
        load printers = no
        show add printer wizard = no
        printcap name = /dev/null
        disable spoolss = yes
        idmap config * : range = 10000 - 40000
        idmap config * : backend = tdb
        winbind enum groups = yes
        winbind enum users = yes
        winbind use default domain = yes
        template shell = /bin/bash
        winbind refresh tickets = yes
```

```
        log level = 3
        log file = /var/log/samba/%m.%U.log
        max log size = 50000
```

```
wbinfo -n internet_full_access      # Bu qrupun sid-ni axtarırıq və nəticə
                                    aşağıdakı kimi olacaq
S-1-5-21-3786744645-3232078785-4224732712-4108 SID_DOM_GROUP (2)

wbinfo -Y S-1-5-21-3786744645-3232078785-4224732712-4108 # Həmçinin SID-i
                                                          GID-e convert edəndə
                                                          10002 aşağıdakı nəticə
                                                          olmalıdır
```

Əgər bu cavab **Could not convert sid to gid** çıxarsa, demək winbind DC-dən cavab ala bilmir.

```
wbinfo -G 10002                     # GID-dən SID-ə qayıdaq
S-1-5-21-3786744645-3232078785-4224732712-4108


wbinfo -s S-1-5-21-3786744645-3232078785-4224732712-4110    #SID-i qrupname-ə
                                                            qaytaraq
DOMAIN+internet_low_access 2

wbinfo -S S-1-5-21-3786744645-3232078785-4224732712-2200    # User SID-i UNIX
                                                            ID-ə convert
                                                            edirik
11949

getent passwd | grep 11949    # UID ilə bazamızda axtarış edirik
parviz.mammadov:*:11949:10006:Parviz
Mammadov:/home/DOMAIN/parviz.mammadov:/sbin/nologin

wbinfo -U 11949         # UNIX ID-ni Windows SID-ə yenidən convert edirik
S-1-5-21-3786744645-3232078785-4224732712-2200

testparm                # Once Samba-nı test edək.
Load smb config files from /usr/local/etc/smb.conf
Loaded services file OK.
Server role: ROLE_DOMAIN_MEMBER
Press enter to see a dump of your service definitions
```

/etc/rc.conf-umuzda bu movzu üçün aşağıdakı sətirlər movcuddur:

```
samba_enable="YES"
winbindd_enable="YES"
kerberos5_server_enable="YES"
squid_enable="YES"
```

Hal-hazırda işləyən **/etc/krb5.conf** quraşdırma faylımız aşağıdakı kimidir (Qeyd: Nəzərə alın ki, siz default_realm-da təyin etdiyiniz DC adı böyük

hərflərlə yazıldığına görə də, siz kinit-lə login olanda DC adını böyük hərflə yazmalısınız):

```
[libdefaults]
        default_realm = DOMAIN.LAN
        kdc_timesync = 1
        ccache_type = 4
        forwardable = true
        proxiable = true
        v4_instance_resolve = false
        v4_name_convert = {
                host = {
                        rcmd = host
                        ftp = ftp
                }
                plain = {
                        something = something-else
                }
        }
        fcc-mit-ticketflags = true

[realms]
        DOMAIN.LAN = {
                kdc = dc01
                kdc = dc02
                admin_server = dc01
                default_domain = DOMAIN.LAN
        }

[domain_realm]
        .domain.lan = DOMAIN.LAN
        domain.lan = DOMAIN.LAN
[login]
        krb4_convert = false
        krb4_get_tickets = false
```

Lazımı yetkiləri verək ki, **squid wbinfo_group.pl** scripti öz istifadəçi adı və şifrəsi ilə çağıra bilsin:

```
chown -R squid:squid /var/squid/
chown -R squid:squid /usr/local/libexec/squid/
chown -R squid:squid /usr/local/etc/squid/
```

Bu ona görədir ki, squid öz konfiqində squid istifadəçi və qrup adından işləməsini aşağıdakı kimi deyib:

```
cache_effective_user squid
cache_effective_group squid
```

```
/usr/local/etc/rc.d/squid stop          # Squid-i dayandırırıq
/usr/local/etc/rc.d/samba stop          # Sambani dayandırırıq (Həmçinin
                                        winbind dayanır)
```

Squid-in WinBind-ə qoşula bilməsi üçün lazımı yetkiləri veririk:

```
chown -R root:squid /var/db/samba/winbindd_privileged/
chmod -R 750 /var/db/samba/winbindd_privileged/

ntpdate domain.lan              # DC-mizdən ən son və düzgün tarixi alırıq

net cache flush                 # Samba şəbəkə cache-ni təmizləyirik

kdestroy                        # Aldığımız açarı silirik

kinit -p jamaladm       # DC-dən yeni açar alırıq
Credentials cache: FILE:/tmp/krb5cc_0
        Principal: SQUIDPRIMARY$@DOMAIN.LAN

  Issued           Expires          Principal
Oct 31 16:00:00  Nov  1 02:00:00  krbtgt/DOMAIN.LAN@DOMAIN.LAN


kinit --renewable jamaladm@DOMAIN.LAN           # Yuxarıda qeyd etdiyim kimi,
                                                DC böyük hərflərlə yazılır
                                                çünki, /etc/krb5.conf
                                                quraşdırma faylında
                                                default_realm-da DC böyük
                                                hərflərlə qeyd edilmişdir.

kinit -renewable        # açarın şifrəsiz yenilənə bilməsinə yetki veririk

kinit -renew            # Bu əmrlə yeniləyirik
kinit -R                # Yada bu əmrlə yeniləyirik

rm -rf /var/db/samba/*.tdb      # ID xəritələnməsi faylını silirik


net ads join -U jamaladm@domain.lan       # DC-mizə yenidən login oluruq

# Aşağıdakı əmri birbaşa şifrə daxil edilmədən script-də istifadə etmek olar
net ads join -W domain.lan -I 10.70.3.2 -U Jamaladm%DC@c0untp#$

net groupmap list               # Qrup-ların xəritələnməsinə baxırıq
Administrators (S-1-5-32-544) -> internet_low_access
Users (S-1-5-32-545) -> BUILTIN\users

net ads lookup                  # Domain controller quruluşuna baxaq
Information for Domain Controller: 10.70.3.3

Response Type: LOGON_SAM_LOGON_RESPONSE_EX
GUID: 271fef32-c64e-4d10-a8ae-cd8aedf8993b
Flags:
        Is a PDC:                                       no
        Is a GC of the forest:                          yes
        Is an LDAP server:                              yes
        Supports DS:                                    yes
        Is running a KDC:                               yes
        Is running time services:                       yes
```

```
        Is the closest DC:                              yes
        Is writable:                                    yes
        Has a hardware clock:                           no
        Is a non-domain NC serviced by LDAP server: no
        Is NT6 DC that has some secrets:                no
        Is NT6 DC that has all secrets:                 yes
Forest:                 domain.lan
Domain:                 domain.lan
Domain Controller:      dc02.domain.lan
Pre-Win2k Domain:       DOMAIN
Pre-Win2k Hostname:     DC02
Server Site Name :              Main
Client Site Name :              Main
NT Version: 5
LMNT Token: ffff
LM20 Token: ffff
```

```
net ads info                    # DC haqqında məlumat alırıq
LDAP server: 10.70.3.3
LDAP server name: dc02.domain.lan
Realm: DOMAIN.LAN
Bind Path: dc=DOMAIN,dc=LAN
LDAP port: 389
Server time: Sat, 08 Nov 2014 19:18:52 AZT
KDC server: 10.70.3.3
Server time offset: 3
```

```
net sam createbuiltingroup admininstrators      # Bu əmrlə BuiltIn qrupları
                                                yarada bilərsiniz. Ancaq sizə
                                                lazım deyil çünki groupmap-də
                                                bütün qruplar görsənir.

/usr/local/etc/rc.d/samba start                 # Sambanı işə salırıq
/usr/local/etc/rc.d/samba restart               # WinBind serisi yoxlamaq
                                                üçün sambanı yenidən işə
                                                salırıq
```

Squid-in işə salmazdan və **/var/squid/logs/cache.log**-u analiz etməzdən öncə aşağıdakı yoxlanışları bir daha edirik:

```
klist             # açarımıza yenidən baxırıq
Credentials cache: FILE:/tmp/krb5cc_0
        Principal: jamaladm@DOMAIN.LAN

  Issued           Expires          Principal
Oct 31 18:17:55  Nov  1 04:17:55  krbtgt/DOMAIN.LAN@DOMAIN.LAN
Oct 31 18:19:40  Nov  1 04:17:55  ldap/dc02.domain.lan@DOMAIN.LAN
Oct 31 18:19:49  Nov  1 04:17:55  ldap/dc01.domain.lan@DOMAIN.LAN
```

```
wbinfo -u                              # DC-i istifadəçilərə baxırıq
javad.javadov
khumar.kazimova
aydin.jafarov
rofat.guliyev
dilara.ahmadova
anvar.bagiyev
zenfira.huseynova
jamil.zeynalov
nijat.shukurov
simuzar.huseynova

wbinfo -t                              # RPC çağırışı yoxlayırıq
checking the trust secret for domain DOMAIN via RPC calls succeeded

wbinfo -p                              # WinBind-i ping edirik
Ping to winbindd succeeded


wbinfo -P                              # NetLogon DC qoşulmasını yoxlayırıq
checking the NETLOGON dc connection succeeded

wbinfo -g                              # DC-i qruplarına baxırıq
cspersistentchatadministrator
cshelpdesk
csviewonlyadministrator
csserveradministrator
csarchivingadministrator
cslocationadministrator


getent passwd                  # DC istifadəçilərini UNIX formatında alırıq
rubaba.baghirova:*:10021:10000:Rubaba
Baghirova:/home/DOMAIN/rubaba.baghirova:/bin/sh
ramiz.asilbayli:*:10022:10000:Ramiz
Asilbayli:/home/DOMAIN/ramiz.asilbayli:/bin/sh
mansura.zeynalova:*:10023:10000:Mansura
Zeynalova:/home/DOMAIN/mansura.zeynalova:/bin/sh
gazanfar.bagirov:*:10024:10000:Gazanfar
Bagirov:/home/DOMAIN/gazanfar.bagirov:/bin/sh
ayda.ibrahimkhalilov:*:10025:10000:Ayda
Ibrahimkhalilova:/home/DOMAIN/ayda.ibrahimkhalilov:/bin/sh
ariz.verdiyev:*:10026:10000:Ariz Verdiyev:/home/DOMAIN/ariz.verdiyev:/bin/sh
lachin.babayev:*:10027:10000:Lachin
Babayev:/home/DOMAIN/lachin.babayev:/bin/sh

getent group                           # DC qruplarını UNIX formatda alırıq
enterprise admins:x:10006:dcadm
enterprise read-only domain controllers:x:10014
rtccomponentuniversalservices:x:10044:lync01$
```

```
id full                                 # full adlı istifadəçi üçün UNIX ID-ni
                                        belə alırıq
uid=11476(full) gid=10006(domain users) groups=10006(domain
users),10030(inet_full),10007(internet_full_access),10029(tacacsadmin),10028(
openvpnma)
```

DC-de olan qruplarin SID-ə convert edilməsinə baxaq:

```
wbinfo -n Internet_Full_Access
S-1-5-21-3786744645-3232078785-4224732712-4108 SID_DOM_GROUP (2)

wbinfo -n Internet_Low_Access
S-1-5-21-3786744645-3232078785-4224732712-4110 SID_DOM_GROUP (2)

wbinfo -n Internet_Medium_Access
S-1-5-21-3786744645-3232078785-4224732712-4109 SID_DOM_GROUP (2)
```

Hemcinin SID-dən GID-ə convert edilməsinə baxaq:

```
wbinfo -Y S-1-5-21-3786744645-3232078785-4224732712-4108
10075

wbinfo -Y S-1-5-21-3786744645-3232078785-4224732712-4110
10077

wbinfo -Y S-1-5-21-3786744645-3232078785-4224732712-4109
10076
```

Hemcinin **/usr/local/etc/squid/squid.conf** quraşdırma faylında müraciət header-in həcmini aşağıdakı kimi biraz artırırıq:

```
request_header_max_size 35 KB
```

```
/usr/local/etc/rc.d/squid start     # Sonda Squid Daemon-u işə salırıq

tail -f /var/squid/logs/cache.log   # Online-da jurnalları araşdırırıq ki,
                                    bir daha belə səhv olmasın
```

Debug Rejimde full istifadəçisi ilə **Internet_Full_Access** qrupunda qeydiyyatdan keçməyə çalışaq.

```
echo "full Internet_Full_Access" | /usr/local/libexec/squid/wbinfo_group.pl -
d
Debugging mode ON.
Got full Internet_Full_Access from squid
User:  -full-
Group: -Internet_Full_Access-
SID:   -S-1-5-21-3786744645-3232078785-4224732712-4108-
GID:   -10003-
Sending OK to squid
OK
```

Əgər istifadəçidən sonra DC adını yazsaq səhv çap edilir:

```
echo "full@domain.lan Internet_Full_Access" |
/usr/local/libexec/squid/wbinfo_group.pl -d
Debugging mode ON.
Got full@domain.lan Internet_Full_Access from squid
User:  -full@domain.lan-
Group: -Internet_Full_Access-
SID:   -S-1-5-21-3786744645-3232078785-4224732712-4108-
GID:   -10003-
failed to call wbcGetGroups: WBC_ERR_DOMAIN_NOT_FOUND
Could not get groups for user full@domain.lan
Sending ERR to squid
ERR
```

```
perl /usr/local/libexec/squid/wbinfo_group.pl    # full user ilə
                                                 internet_full_access qrupunu
                                                 test edək
full internet_full_access
```

NTLM ilə yoxlayırıq:

```
/usr/local/bin/ntlm_auth --username=full
password:
NT_STATUS_OK: Success (0x0)
```

```
wbinfo -a full%A123456789a          # Yenə də full istifadəçisi və
                                    A123456789a şifrəsi ilə qoşulduq (uğurlu
                                    nəticə)
plaintext password authentication succeeded
challenge/response password authentication succeeded
```

```
wbinfo -a full@domain.lan%A123456789a     # Eyni ilə ancaq DC ilə
plaintext password authentication failed
Could not authenticate user full@domain.lan%A123456789a with plaintext
password
challenge/response password authentication failed
error code was NT_STATUS_NO_SUCH_USER (0xc0000064)
error message was: No such user
Could not authenticate user full@domain.lan with challenge/response
```

```
wbinfo --allocate-uid         # ID yerləşməsini test edə bilərik
```

Həmçinin Samba-nın jurnal faylında olan domain-imizə aid olan WinBind jurnalını analiz edirik. Ancaq jurnalları həmişə result-a görə araşdırmaq lazımdır. Aşağıdakı kimi:

```
tail -f /var/log/samba/log.wb-DOMAIN
              type                    : *
```

```
    type                    : SID_NAME_USER (1)
domain                  : *
    domain                  : *
        domain                  : 'DOMAIN'
name                    : *
    name                    : *
        name                    : 'elnur.alizade'
result                  : NT_STATUS_OK
```

Həmçinin əgər siz DC-də olan istifadəçilərdən hansısa birinin yerini dəyişsəniz, yəni yetkisini artırmaq vəya azaltmaq istəsəniz bu vaxt alacaq. Bu ona görədir ki, istifadəçi Sambada olan **winbind cache time** müddətinə baxacaq:

```
cat /usr/local/etc/smb.conf | grep cache        # Məndə olan vaxt 15 dəq ya
                                                  da 900 saniyədir
        winbind cache time = 900
```

Bu problemi həll etmek üçün isə istifadəçini CLI-dan əlimizlə qeydiyyatdan keçiririk:

```
wbinfo --authenticate=full%A123456789a          # full istifadəçisini
                                                  A123456789a şifrə ilə
                                                  tez login edirik ki,
                                                  tez qrupu dəyişsin.
```

Biraz external ACL **nt_group** strukturunu açıqlayaq:

```
external_acl_type nt_group ttl=120 negative_ttl=120 grace=90 children=500
%LOGIN  /usr/local/libexec/squid/wbinfo_group.pl
```

ttl=n (**Time-To-Live** yaşama vaxtı) Kənar ACL-in emalı nəticələrinin saniyələrlə olan saxlanma müddətidir (Susmaya görə **3600** saniyədir yəni **1 saat**).

negative_ttl=n TTL Kənar ACL-in neqativ nəticələrinin saxlanılması üçün saniyələrlə olan müddətdir(Susmaya görə TTL-in mənası ilə eyni olur yəni **3600** saniye)

grace=n TTL-in faizlərlə gözləmə müddətidir hansı ki, cache verilənlərinin yenilənməsi, yəni cavabın gözlənməsinə ehtiyacı olmadan inisializasiya edilməlidir (Susmaya görə **0-dir** gözləmə period yoxdur).

Bütün bu yazdıqlarımdan sonra siz serveri reboot və ya **samba daemon**-u restart etsəniz belə, hər halda **getent passwd** və **getent group** əmrinin nəticəsini uğurla almalısınız.

Squid başlıqlara görə süzgəc

Əgər siz Squid-də genişlənmələrə görə nəyisə download etmək üçün bağlasanız bu heç də o demek deyil ki, onları yenə də download etmək olmaz. Çünki, adi halda istifadəçilər download-da http_request edir. Hal-hazırda əksər saytlar download üçün ünvanı mürəciətdən sonra verir yeni http_reply-da bu halda sizə genişlənmələr kömək edə bilməyəcəklər. Sizə yalnız fayl tiplərinin başlıqları kömək edə bilər. Yəni MIME-Types. Ümumiyyətlə Squid-in ev qovluğunda yəni, **'/usr/local/etc/squid/'**-də artıq **mime.conf** adlı fayl mövcud olur və onun içində bütün başlıqlar aydın şəkildə yazılmışdır.

Biz sadəcə **'/usr/local/etc/squid/squid.conf'** faylında bizim mime cədvəlimizin hansı fayldan oxuduğunu elan edəcəyik və özümüzə uyğun olan MIME ACL-ni yaradacıyıq.

```
# Bu '_sams_52732c3181187' Müəyyən bir ACL-dir və
/usr/local/etc/squid/52732c3181187.sams faylında authentifikasiyadan keçən
istifadəçilərin listini saxlayır.

acl _sams_52732c3181187 proxy_auth "/usr/local/etc/squid/52732c3181187.sams"
acl _sams_52732c3181187_time time MTWHFAS 00:00-23:59
# deny_time ACL isə artıq lazım olan MIME-type-ları təyin edir.
acl deny_mime rep_mime_type -i ^application/octet-stream
acl deny_mime rep_mime_type -i ^application/x-shockwave-flash$
acl deny_mime rep_mime_type -i ^application/octet-stream$
acl deny_mime rep_mime_type -i ^application/x-tar$
acl deny_mime rep_mime_type -i ^application/zip$
acl deny_mime rep_mime_type -i ^application/x-gtar$
acl deny_mime rep_mime_type -i ^application/x-tar$
acl deny_mime rep_mime_type -i ^audio/mpeg$
acl deny_mime rep_mime_type -i ^audio/x-aiff$
acl deny_mime rep_mime_type -i ^audio/x-wav$
acl deny_mime rep_mime_type -i ^audio/mp3$
acl deny_mime rep_mime_type -i ^video/mpeg$
acl deny_mime rep_mime_type -i ^video/quicktime$
acl deny_mime rep_mime_type -i ^video/x-msvideo$
acl deny_mime rep_mime_type -i ^video/x-sgi-movie$
acl deny_mime rep_mime_type -i ^video/vnd.mpegurl$
acl deny_mime rep_mime_type -i ^audio/x-realaudio$
acl deny_mime rep_mime_type -i ^audio/x-pn-realaudio$
acl deny_mime rep_mime_type -i ^application/x-rar-compressed

# Artıq aşağıdakı qaydada yazırıq ki, bu ACL-də _sams_52732c3181187 olan
istifadəçilərə
# deny_mime MIME type-ları qadağandır.
#  TAG: http_reply_access
http_reply_access deny deny_mime _sams_52732c3181187

#  TAG: mime_table
mime_table /usr/local/etc/squid/mime.conf       # Mime cədvəlini elan edirik
```

Windows yenilənməsi

Domain-də olan istifadəçilər Squid üzərindən Windows və Windows Antivirus Essentials Update etdikdə çoxlu problemlər çıxır. Bunları aradan qaldırmaq üçün aşağıdakıları etməyiniz yetərlidir.

```
acl windowsupdate dstdomain windowsupdate.microsoft.com
acl windowsupdate dstdomain .update.microsoft.com
acl windowsupdate dstdomain download.windowsupdate.com
acl windowsupdate dstdomain redir.metaservices.microsoft.com
acl windowsupdate dstdomain images.metaservices.microsoft.com
acl windowsupdate dstdomain c.microsoft.com
acl windowsupdate dstdomain www.download.windowsupdate.com
acl windowsupdate dstdomain wustat.windows.com
acl windowsupdate dstdomain crl.microsoft.com
acl windowsupdate dstdomain sls.microsoft.com
acl windowsupdate dstdomain productactivation.one.microsoft.com
acl windowsupdate dstdomain ntservicepack.microsoft.com
acl windowsupdate dstdomain office15client.microsoft.com
acl windowsupdate dstdomain login.microsoftonline.com
acl CONNECT method CONNECT
acl wuCONNECT dstdomain www.update.microsoft.com
acl wuCONNECT dstdomain sls.microsoft.com

# və əsas istifadəçi ACL-lərindən sonra aşağıdakı ACL-lər yazmağınız yetərlidir.
# hansı ki, deyirik CONNECT metodu ilə wuCONNECT ACL-ində olan LINK-lərə, bütün localnet ACL-də olan IP
# adreslərlə qoşulmaya izin veririk
http_access allow CONNECT wuCONNECT localnet
http_access allow windowsupdate localnet


##### RFC-nin təsdiqlədiyi LOCAL IP ünvanların aralığı hansı ki, localnet ACL-indədir
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
```

BÖLÜM 7

Daxili resursların şifrələnmiş kanalla idarə edilməsi

- **FreeBSD OpenVPN**
- **FreeBSD serverdə OpenVPN Active Directory ilə inteqrasiyası**
- **Ubuntu serverdə OpenVPN Active Directory ilə inteqrasiyası**
- **Ubuntu serverdə OpenVPN FreeRADIUS AD inteqrasiyası**

Hər bir şirkətin müəyyən zamandan sonra daxili informasiya resurslarına girişi üçün tələb yarana bilər. Çünki, hər hansı bir istifadəçinin çox təcili işi çıxa bilər və eyni anda da işə gəlmək imkanı olmaya bilər. Həmçinin qeyri iş vaxtlarında və ya şənbə,bazar günlərində hansısa işin təcili görülməsi tələbi yarana bilər. Bu hallarda şirkətə qoşulmaq tələbi yaranacaq. Başlığımızda OpenVPN vasitəsilə istifadəçilərin uzaqdan qoşulması üçün VPN Server quraşdırırıq. VPN serverimizi həm Active Directory, həm də FreeRADİUS-la inteqrasiya edəcəyik. FreeRADİUS server isə öz növbəsində Active Directory ilə inteqrasiya edilmişdir.

FreeBSD OpenVPN

İstənilən şirkətin daxilində olan informasiya resurslarına xidmət tələb edilir. Bu xidməti isə şirkətin İT və Proqramlaşdırma şöbələri edir. İş prosesində həm proqramlaşdırma və həm də İT işçilərindən tələb edilə bilər ki, qeyri iş vaxtları və şənbə, bazar günləri hansısa iş yerinə yetirilməlidir. Artıq hər kəs öz evindən işə hansısa vasitə ilə qoşulmaq məcburiyyətində qalır çünki, əks halda işə gəlməli olacaqlar. Bu halda bizim köməyimizə açıq qaynaqlı OpenVPN çacatcaq. Bu proqram təminatı təhlükəsizlik baxımından da mükəmməldir və istənilən client əməliyyat sistemi üçün proqrama sahibdir. Başlığımızda FreeBSD server üzərində OpenVPN qurulması açıqlanacaq.

İlk işimiz serverimizi Router rejimində işə salmaqdır cünki, OpenVPN-in virtual şəbəkəsi yönləndirilmə tələb edir.

Sistemin yenidən yüklənməsindən sonra işləməsi üçün aşağıdakı əmri yerinə yetiririk:

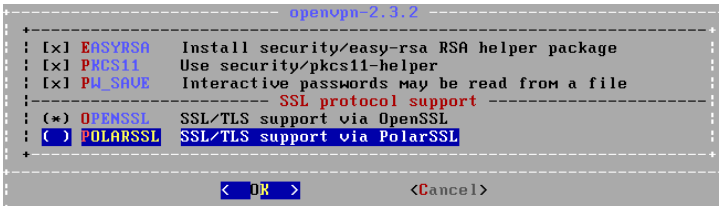
```
# echo 'gateway_enable="YES"' >> /etc/rc.conf
```

Hal-hazırki senasda yerinə yetirmək üçün aşağıdakı əmri yerinə yetiririk:

```
# sysctl -w net.inet.ip.forwarding=1
```

OpenVPN-i portlardan yükləyək.

```
cd /usr/ports/security/openvpn          # Port ünvanına daxil oluruq.
make config                             # Lazımi modulları seçirik.
```




```
make install                            # Yükləyirik
```

Qeyd: FreeBSD serverimizdə kernel-in **tap** və **tun** tipli alətlərin dəstəklənməsi üçün **/sys/amd64/conf/GENERIC** faylında "**Pseudo devices**" bölümündə "**device tap**" və "**device tun**" fərqli sətirlərdə əlavə edib kerneli yenidən kompilyasiya eləmək lazımdır. OpenVPN-in versiyası 2.3.32-dir.

Sözsüz ki, gələcək üçün OpenVPN-in OpenSSL sertifikatlarını daha rahat idarə etmək üçün ssl-admin portunu da yükləmək lazımdır.

Port ünvanına daxil oluruq:

```
root@siteA:~ # cd /usr/ports/security/ssl-admin/
root@siteA:/usr/ports/security/ssl-admin # make install     # Yukleyirik
```

PKİ infrastruktur üçün easy-rsa-ni portlardan yükləyirik:

```
# cd /usr/ports/security/easy-rsa
# make -DBATCH install
```

Easy-RSA tələb elədiyi üçün BASH yükləyirik:

```
# pkg install bash
```

Public və Private açarların qurulması

Client/Server VPN yaratmazdan öncə biz PUBLIC açar (**PKI**) infrastrukturunu yaratmalıyıq. PKI özünə Certificate Authority, Private açarları və certificates (Public açarları) həm client və həm də server üçün daxil edir. Həmçinin biz Diffie-Hellman parametrli açar generasiya etməliyik ki, gizliliyi ideal forward edə bilək.

PKI yaratmaq üçün biz OpenVPN tərəfindən yaradılmış **easy-rsa** scriptlərindən istifadə etməliyik.

İşə başlayaq

PKI tam inandığımız bir serverdə olmalıdır. O həmçinin elə OpenVPN serverin özündə də ola bilər ancaq, təhlükəsizlik tələblərinə görə o tamam ayrı bir server üzərinde olmalıdır. Əsas tələblərindən biri odur ki, **CA(Certificate Authority)** açarı tamam başqa yerdə saxlayaq, misal üçün external storage hansı ki, yalnız tələb ediləndə istifadə edilsin. Digər əsas tələb odur ki, CA private açarı tamam şəbəkədən ayrılmış bir serverdə saxlamaq lazımdır.

Bu resepti FreeBSD9.2 x64 maşında istifadə etmişəm. Linux və Windows maşında da eyni əmrlərlə istifadə edə bilərsiniz. Ancaq easy-rsa scriptlərin işlənməsi üçün BASH tələb edilir ona görə də maşınınıza öncədən baş-ı yükləməyi unutmayın. (easy-rsa portlarda **/usr/ports/security/easy-rsa** ünvanında yerləşir)

Necə edək

1. PKI üçün qovluqları yaradın və easy-rsa scriptlərini həmin qovluğa nüsxələyin:

```
root@siteA:~ # mkdir -m 700 -p /usr/local/etc/openvpn/scripts/keys

root@siteA:~ # cd /usr/local/etc/openvpn/scripts
root@siteA:~ # cp -R /usr/local/share/easy-rsa/*.
```

2. Bu əmrlərin root istifadəçi adından işə salınmasına gərək yoxdur.

3. Sonra biz **vars** faylını yaradaq. Faylı yaradın və aşağıdakıları içinə əlavə edin.

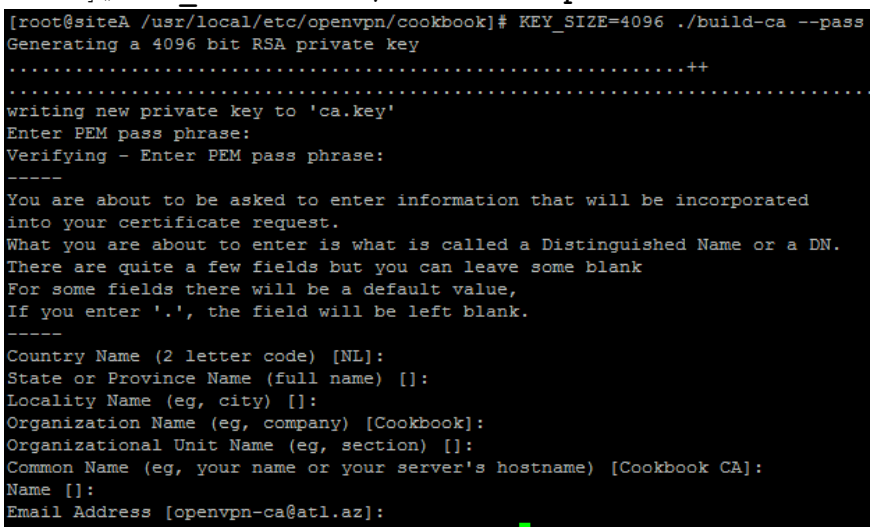
```
export EASY_RSA=/usr/local/etc/openvpn/scripts
export OPENSSL="openssl"
export KEY_CONFIG=`$EASY_RSA/whichopensslcnf $EASY_RSA`
export KEY_DIR="$EASY_RSA/keys"
export PKCS11_MODULE_PATH="dummy"
export PKCS11_PIN="dummy"
export KEY_SIZE=2048
export CA_EXPIRE=3285
export KEY_EXPIRE=1000
export KEY_COUNTRY="NL"
export KEY_PROVINCE=
export KEY_CITY=
```

```
export KEY_ORG="Scripts"
export KEY_EMAIL="openvpn-ca@scripts.example.com"
```

Qeyd: **PKCS11_MODULE_PATH** ve **PKCS11_PIN** verilənləri o halda tələb edilir ki, siz SmardCard istifadə etmirsiniz. Susmaya görə olan **KEY_SIZE** 2048 bitdir və bu uzunluq növbəti 2-3 il üçün təhlükəsizdir. Həmçinin geniş uzunluqlu **4096**-bitlik açar mümkündür ancaq şifrələnmə böyük olduğuna görə performance aşağı düşəcək. Biz 4096 bitlik CA private açar yaradacağıq ona görə ki, burda performace heç nəyə gərək deyil. Həmçinin dəyişənlər var ki, sizin təşkilata(**KEY_ORG**, **KEY_EMAIL**) xasdır. Bu quraşdırmaların açıqlanmasını birazdan daha detallı şəkildə danışacağıq.

4. 4096 bitlik modul istifadə edərək **vars** faylı yerinə yetirək, CA private açar **ca** sertifikat genereasiya edək. CA sertifikat üçün çətin şifrə seçin. Bundan sonra hər dəfə script işə düşdükdən sonra həmin şifrə daxil edin:

```
root@siteA:~ # cd /usr/local/etc/openvpn/scripts/

root@siteA:~ # bash                    # BASH-a keçirik.
[root@siteA ]# source ./vars
[root@siteA ]# ./clean-all
[root@siteA]# KEY_SIZE=4096 ./build-ca --pass
```



5. Sonra biz server sertifikatını generasiya edəcəyik. Script daxil edilməsini istəyəndə şifrəni daxil edib enter-i sıxın. Script **ca.key** şifrəsini istəyəndə isə CA sertifikatı üçün şifrəni daxil edin. Sonda isə script soruşacaq **[y,n]** siz **y** edin.

```
[root@siteA /usr/local/etc/openvpn/scripts]# ./build-key-server openvpnserver
Generating a 2048 bit RSA private key
.....................................+++
```

```
.....................+++
writing new private key to 'openvpnserver.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [NL]:
State or Province Name (full name) []:
Locality Name (eg, city) []:
Organization Name (eg, company) [Scripts]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [openvpnserver]:
Name []:
Email Address [openvpn-ca@domain.az]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /usr/local/etc/openvpn/scripts/openssl-0.9.8.cnf
Enter pass phrase for /usr/local/etc/openvpn/scripts/keys/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'NL'
organizationName      :PRINTABLE:'Scripts'
commonName            :PRINTABLE:'openvpnserver'
emailAddress          :IA5STRING:'openvpn-ca@domain.az'
Certificate is to be certified until Oct  9 19:15:14 2016 GMT (1000 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

6. Client üçün ilk sertifikat **build-key** ilə yaradılır. Bu client sertifikatının yaradılması üçün çox sürətli metodikadır ancaq, bu halda clientin private key faylına şifrə təyin etmək olmur.

```
[root@siteA /usr/local/etc/openvpn/scripts]# ./build-key openvpnclient1
```

7. Ikinci client sertifikatı şifrə ilə generasiya edilmişdir. Çətin şifrə seçin(Yalnız CA sertifikat-da seçdiyiniz şifrədən fərqli olmalıdır!). Aydınlıq üçün çıxış qısa göstərilmişdir:

```
[root@siteA ]# ./build-key-pass openvpnclient2
Using configuration from /usr/local/etc/openvpn/scripts/openssl-0.9.8.cnf
Enter pass phrase for /usr/local/etc/openvpn/scripts/keys/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'NL'
organizationName      :PRINTABLE:'Scripts'
commonName            :PRINTABLE:'openvpnclient2'
emailAddress          :IA5STRING:'openvpn-ca@domain.az'
Certificate is to be certified until Oct 10 05:08:03 2016 GMT (1000
days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

8. Sonra isə server üçün Diffie-Hellman parametrli fayl qurun:
   `[root@siteA /usr/local/etc/openvpn/scripts]# ./build-dh`

9. Ardınca **tls-auth** key faylı:

```
[root@siteA /usr/local/etc/openvpn/scripts]# openvpn --genkey --secret
keys/ta.key
```

Bütün bu gördüyümüz işlərdən sonra **/usr/local/etc/openvpn/scripts/keys** qovluğunda aşağıdakı fayllar yaranacaq:
ca.crt - Əsas CA sertifikat, bu fayl həm client və həm də serverə lazımdır
dh2048.pem - Diffie Hellman açarı, bu fayl yalnız serverə lazımdır
Qeyd: Əgər bu açar yaranmazsa, sadəcə **/usr/local/etc/openvpn/keys** ünvanında **./build-dh** əmrini daxil etməniz yetər ki, **dh2048.pem** açarı yaransın.

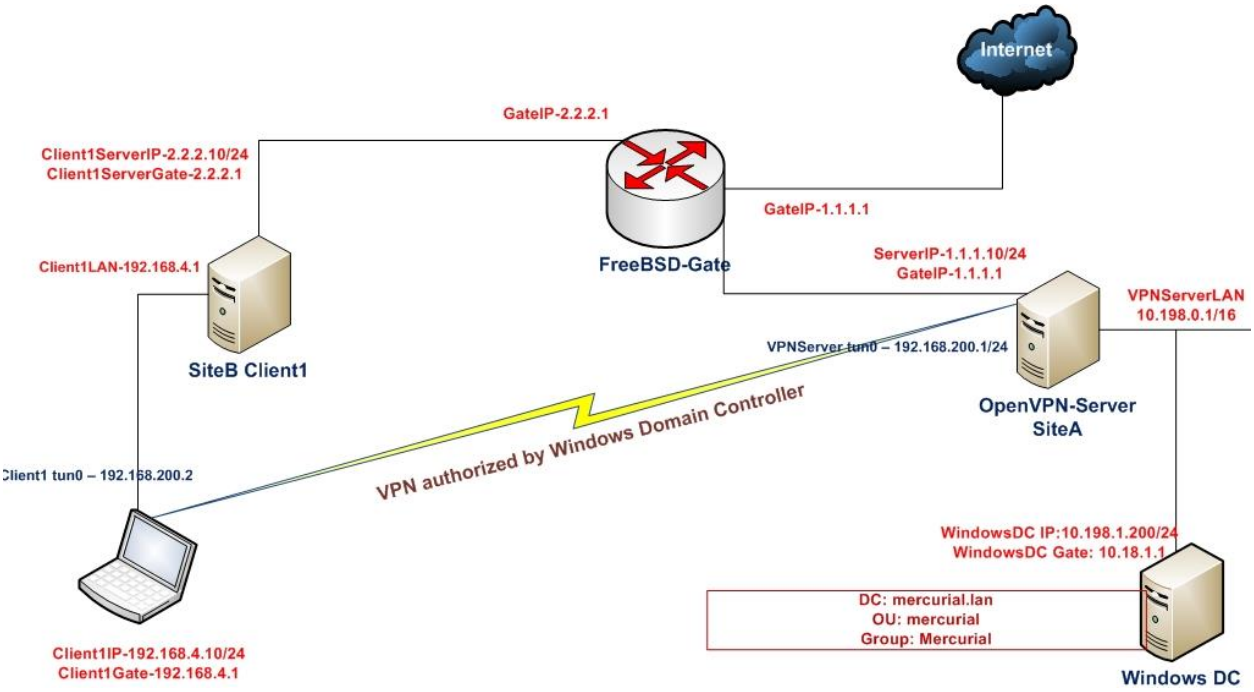
openvpnserver.crt - Serverin sertifikatı, yalnız server üçündür
openvpnserver.key - Serverin açarı, yalnız server üçündür (gizli fayl)
openvpnclient1.crt - Clientin sertifikatı, yalnız client üçündür
openvpnclient1.key - Clientin açarı, yalnız client üçündür (gizli fayl)
ta.key - TLS-açar, həm client və həm də serverə lazımdır

Uyğun olaraq serverdə **ca.crt**, **dh2048.pem**, **openvpnserver.crt**, **openvpnserver.key**, **ta.key** faylları və ilk client-də isə **ca.crt**, **dh2048.pem**, **openvpnclient1.crt**, **openvpnclient1.key**, **ta.key** faylları olmalıdır.

FreeBSD serverdə OpenVPN Active Directory ilə inteqrasiyası

Bu başlıqda biz OpenVPN-i Windows Domain Controller ilə inteqrasiya edəcəyik. Ancaq hər bir halda client və server arasında olan yol generasiya elədiyimiz CA açarıyla yoxlanacaq və openvpnserver açarı ilə şifrələnəcək.

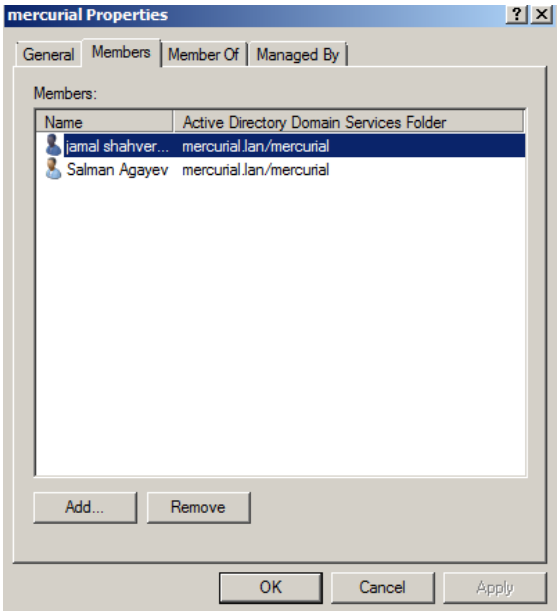
Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:

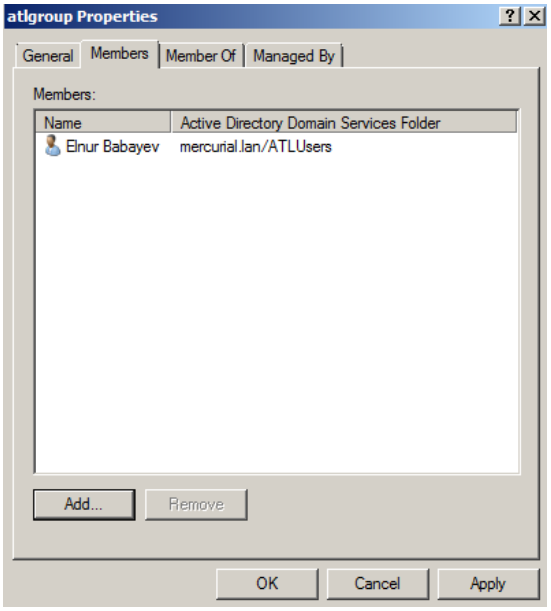


Bu misalımızda 2-ci başlıqda generasiya elədiyimiz CA və server sertifikatlarını həm server və həmdə client üçün istifadə edəcəyik. Server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Həmçinin OpenVPN serverimizin local şəbəkəsinə qoşulmuş Windows 2008 server Domain Controller olacaq.
Domain Controller aşağıdaki verilənlərdən ibarətdir:

DC: **mercurial.lan**
OU: **mercurial**
Group: **mercurial**
Test user: **jamal**

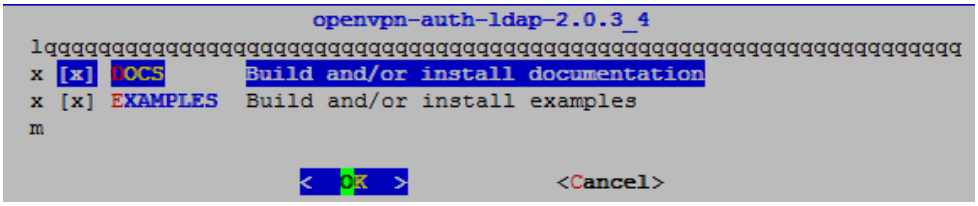
Domain controller maşında **mercurial** adlı qrupun içində **jamal** adlı istifadəçi mövcuddur ki, testlərimizi edə bilək. Həmçinin adi **Users** qrupunun içində **elnur** adlı istifadəçi mövcuddur ki, giriş edə bilməyən istifadəçi kimi onunla test edək.





1. Öncə server maşınımıza lazımi paketləri yükləyək:

```
root@siteA:/usr/local/etc/openvpn # cd /usr/ports/security/openvpn-auth-ldap/

root@siteA:/usr/ports/security/openvpn-auth-ldap # make config     #
                                                    Lazımi modulları seçirik
```




```
root@siteA:/usr/ports/security/openvpn-auth-ldap # make -DBATCH install
                                                    # Yükləyirik
```

2. Auth-LDAP paketi serverə yükləndikdən sonra o **/usr/local/lib/openvpn-auth-ldap.so** ünvanına öz pluginini əlavə edir. Biz məhz bu plugini AD-yə qoşulmaq üçün istifadə edəcəyik. **/usr/local/etc/openvpn/ad-auth.conf** adlı server konfig faylını yaradaq və içinə aşağıdaki sətirləri əlavə edək:

```
plugin /usr/local/lib/openvpn-auth-ldap.so
"/usr/local/etc/openvpn/openvpn-auth-ldap.conf"
proto udp
port 1194
dev tun
server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
```

```
key /usr/local/etc/openvpn/openvpnserver.key
client-cert-not-required
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

push "route 10.198.0.0 255.255.0.0"
topology subnet

user nobody
group nobody

daemon
log-append /var/log/openvpn.log
verb 5
```

Domain Controller-ə qoşulmaq üçün **/usr/local/etc/openvpn/openvpn-auth-ldap.conf** konfig faylının məzmunu aşağıdaki kimi olacaq:

```
<LDAP>
      URL             ldap://10.198.1.200
      BindDN          Administrator@mercurial.lan
      Password        B123456789b
      Timeout         15
</LDAP>
<Authorization>
     BaseDN          "DC=mercurial,DC=lan"
     SearchFilter
"(&(sAMAccountName=%u)(memberOf=CN=mercurial,OU=mercurial,DC=mercurial,DC=lan
))"
</Authorization>
```

Həmçinin unutmayın ki, OpenVPN server-də **/usr/local/etc/openvpn/openvpn-auth-ldap.conf** faylın içində olan Domain adının resolve edilməsi üçün **/etc/resolve.conf** faylına aşağıdakı sətir əlavə edilmişdir.

```
nameserver 10.198.1.200
```

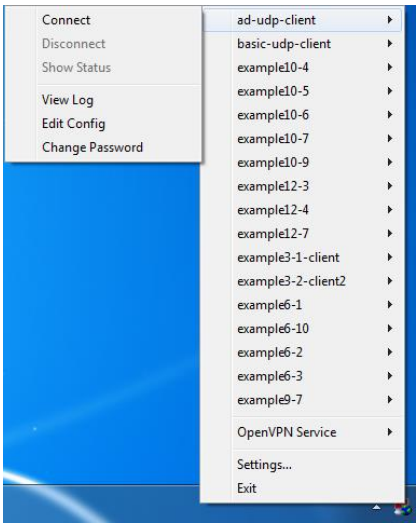
3. OpenVPN serveri işə salaq:
   root@siteA:/usr/local/etc/openvpn # **openvpn --config ad-auth.conf**

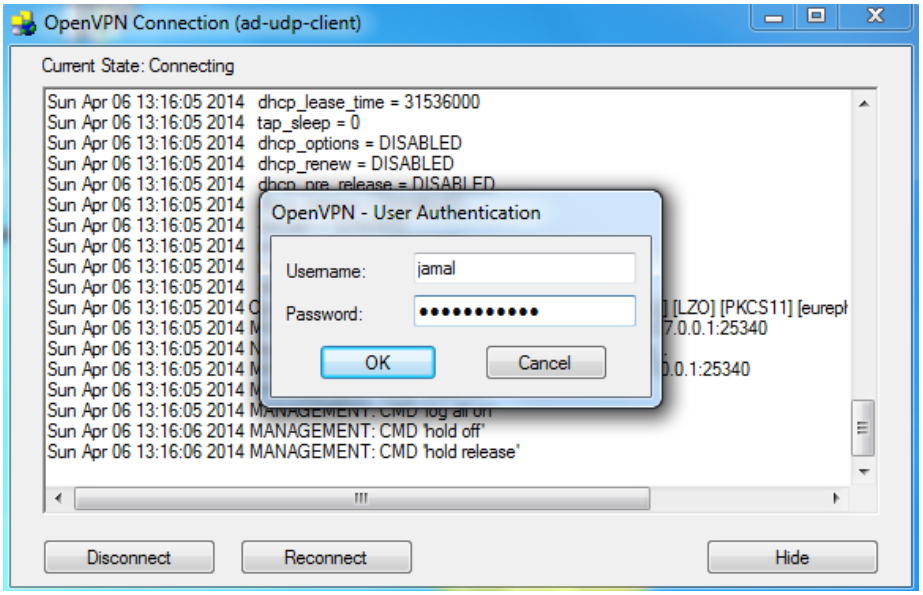
4. Indi isə Windows7 maşında client konfigurasiyasını yaradaq. **C:\Program Files\OpenVPN\config** ünvanında **ad-udp-client.ovpn** adlı fayl yaradaq və içinə aşağıdakı məzmunu əlavə edək:

```
client

auth-user-pass
```

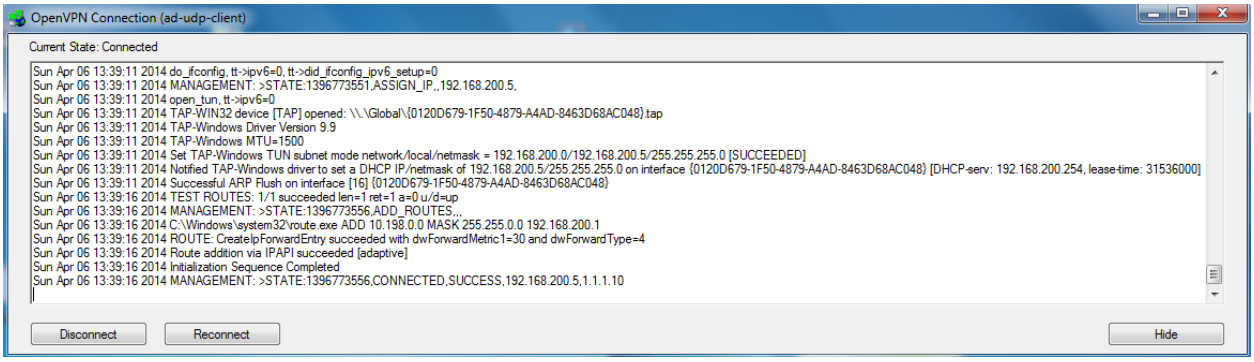
```
proto udp

      remote openvpnserver.example.com

port 1194

dev tun

nobind



ca "c:/program files/openvpn/config/ca.crt"

tls-auth "c:/program files/openvpn/config/ta.key" 1


ns-cert-type server

verb 5
```

5. Windows7 Client maşını işə salaq:





Client maşının statusunda aşağıdaki şəkil çap edilməlidir:



6. Server maşında **/var/log/openvpn.log** log faylına baxıb aşağıdaki sətirləri görməliyik:

```
Sun Apr  6 13:17:43 2014 us=626543 2.2.2.10:53829 PLUGIN_CALL: POST
/usr/local/lib/openvpn-auth-ldap.so/PLUGIN_AUTH_USER_PASS_VERIFY status=0

Sun Apr  6 13:17:43 2014 us=626715 2.2.2.10:53829 TLS: Username/Password
authentication succeeded for username 'jamal'

Sun Apr  6 13:17:43 2014 us=627135 2.2.2.10:53829 Data Channel Encrypt:
Cipher 'BF-CBC' initialized with 128 bit key

Sun Apr  6 13:17:43 2014 us=627163 2.2.2.10:53829 Data Channel Encrypt: Using
160 bit message hash 'SHA1' for HMAC authentication

Sun Apr  6 13:17:43 2014 us=627235 2.2.2.10:53829 Data Channel Decrypt:
Cipher 'BF-CBC' initialized with 128 bit key
```

```
Sun Apr  6 13:17:43 2014 us=627282 2.2.2.10:53829 Data Channel Decrypt: Using
160 bit message hash 'SHA1' for HMAC authentication
```

Qeyd: Nəzərə alın ki, OpenVPN server ilk dəfə işə düşəndə, yolun şifrələnməsindən sonra yalnız ilk client ilk dəfə qoşulanda, qoşulmaya bilər. Ancaq bundan sonra bütün qoşulmalarda həm birinci client və həm də digər clientlər problemsiz qoşulma edəcək.

Əgər OpenVPN serveri startup-a əlavə eləmək istəsəniz, sadəcə aşağıdakı sətirləri **/etc/rc.conf** faylına əlavə etməniz yetər:

```
openvpn_enable="YES"
openvpn_if="tun"
openvpn_configfile="/usr/local/etc/openvpn/ad-auth.conf "
openvpn_dir="/usr/local/etc/openvpn"
```

Ubuntu serverdə OpenVPN Active Directory ilə inteqrasiyası

Ubuntu 14.04 server üzərində OpenVPN yükləyib quraşdıraq ki, VPN-ə qoşulduqda istifadə edilən istifadəçi bazasını Domain controller-dən götürək.
Istifadə edilən OS-lar:

Windows 2012 Server R2 - DC
Windows 8.1 x64 - Client maşını
Ubuntu 14.04 x64 - OpenVPN

Öncə istifadə etdiyimiz Domain Controller-in verilənlərini açıqlayaq
Domain Controller: **domain.lan**
DC RO User: **DCADM**
DC RO PASS: **DcP@$$f0rd**
DC VPN Group: **OpenVPNFAUsers** - Tam yetkisi olan VPN istifadəçiləri(Bütün şəbəkəyə routing olacaq)

Windows 8.1 client quraşdırma faylı aşağıdakı kimi olacaq(faylımızın adı **domain-ad-auth.ovpn)**. Faylın genişlənməsi mütləq **.ovpn** olmalıdır:

```
client
```

```
auth-user-pass
auth-nocache
reneg-sec 86400
proto tcp
remote ovpndc.domain.az
port 1194
dev tun
nobind

key-direction 1

ns-cert-type server

# OpenVPN serverdə yaradılan ca.crt
<ca>
-----BEGIN CERTIFICATE-----
MIIGxDCCBKygAwIBAgIJALsV/eQc/V5+MA0GCSqGSIb3DQEBBQUAMIGcMQswCQYD
VQQGEwJBWjENMAsGA1UECBMEQkFLVTEUMBIGA1UEBxMLWWVuaVlhc2FtYWwxFDAS
BgNVBAoTC0FUTEluZm9UZWNoMQswCQYDVQQLEwJJVDEXMBUGA1UEAxMOQVRMSW5m
b1RlY2ggQ0ExLDAqBgkqhkiG9w0BCQEWHWphbWFsLnNoYWh2ZXJkaXlldkBhdGx0
ZWNoLmF6MB4XDTE0MDYwODE3NDUzOFoXDTIzMDYwNjE3NDUzOFowgZwxCzAJBgNV
BAYTAkFaMQ0wCwYDVQQIEwRCQUtVMRQwEgYDVQQHEwtZZW5pWWFzYW1hbDEUMBIG
A1UEChMLQVRMSW5mb1RlY2gxCzAJBgNVBAsTAklUMRcwFQYDVQQDEw5BVExJbmZv
VGVjaCBDQTEsMCoGCSqGSIb3DQEJARYdamFtYWwuc2hhaHZlcmRpeWV2QGF0bHRl
Y2guYXowggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC0kYn6jZf/R1eA
Xs1YH/g36sIQJcxJBmcbXh/atZTy7W8rlXsCw05+RU7OaXrFQUEbed0lnjYiKfri
CutMpT5c7iY6fgfMMoPaIqk8q17qydk8HvqQoac3kjo9wMD7XWlDYiLFk1FxQjEW
BIqI2z6vh9/9ka54s6WNRgzT+7+EZqSuwCfC6Dm/0qxp4AvEjapwjaURJ6yEuQYe
Odh5ydTsIcueNnBzkuFZRx5O5iNcaBQZ2fUVpQvueTCCsHkPt1BGU3TqWIYTUVZl
O4wPQoOyXC9YUvWaYWSLTDMNDvCvGFYfc5C3++nijtfWpO8LLDZgiwC7ScYj+Boo
SZ9dkEpIYdb03KBnn+LCO3STVukpwTr+vyKjPITceuElHXdWvXi7wgtopwQhQ+3j
sDCvB+Wg2Bt5zBPC43WTeLANOGZFQN1f1kyBNXlBm1tM0kl3k75skkj9TXHjrM44
+aVdxlPjkQ86e6/A04wCUOBNf4aO0Q8r6PWCfpkqatDn6hCh6ChAYYuqAR5W3eRs
p2D3lAWGEHlBlf/+397E66f3ByHvPGQ5nlAQ3wI7q+tLH+qPsoFUKcyfEbctuYvG
D0+9jPhvxAAQwc4hBhn+TXRXPkaaaI89iiaJoiF1//R8kqs8t3yxpxjEy0hs2nrx
tboZl91cO2fj8e2HvhbMs9v+j6oVTQIDAQABo4IBBTCCAQEwHQYDVR0OBBYEFIci
KzboRxhacra8qkU+xvRM4df7MIHRBgNVHSMEgckwgcaAFIciKzboRxhacra8qkU+
xvRM4df7oYGipIGfMIGcMQswCQYDVQQGEwJBWjENMAsGA1UECBMEQkFLVTEUMBIG
A1UEBxMLWWVuaVlhc2FtYWwxFDASBgNVBAoTC0FUTEluZm9UZWNoMQswCQYDVQQL
EwJJVDEXMBUGA1UEAxMOQVRMSW5mb1RlY2ggQ0ExLDAqBgkqhkiG9w0BCQEWHWph
bWFsLnNoYWh2ZXJkaXlldkBhdGx0ZWNoLmF6ggkAuxX95Bz9Xn4wDAYDVR0TBAUw
AwEB/zANBgkqhkiG9w0BAQUFAAOCAgEAT+K70oaUfXDEfSFmBTrppvbcGqoVsaEl
5NjMh206D5KWtERhKbP7id20sdt6YgqlPQWW3I3thVQ0L686rhbZ/cR6Vzj41cFI
EqCt4uqZrkoMcvPq82POnvrzKCauxv5kmZJhWQTB3WXMo0A4KnQqW6/HVzSmbQgC
QR6CqNTt1Z21a1RIQrR1CmqRankKC4yQBKbzDwBlXLHvjITdyhJlHXZxBcdXurMX
Uh7AsHOTxbHy4nbyB+Zz1nO37wza6FBeunIqJ/I5eKDcN1lyGELjDsEvrSUcbRRg
IenV9/D9LP4y2KghMkiuDn7vhY3IifCjxQg3JWIa5BdQ/lU1Accsx1i0/nyQtZF7
5NadlwoSOjEe2H6bwxhlngcItQyiC34HghNKUF16eYLlMEzGkP7UNLwQN32b3IiA
q9+HTP6TQoci43AoaA3NFaUjuKC3zHykesNS8QqOH7MVB4L38/piaGD/K8CsiZH+
QhkICaJJ7hx/Cfp3VUIKr9yxtAnC5QNbXr9QVCC+mwi/sH9laThPlm1Xd2tKdoZa
My/K6o5fZnZSpzOeFa9j6bRgF2tpbG3jxiWT00F9xUv5EtXZdfies5BRHa1FYGK4
yvVIA/ZJBSB/6CT8mnMjGJcn85CcRggOrOc7lQNmgFKw/YopPYyAKzjgi1EKtNm3
```

```
pmPKIhXPdvc=
-----END CERTIFICATE-----
</ca>

# OpenVPN serverdə yaradılan TA açarı
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
7148f7b12478b04aee1445e18bb96509
b7f8d3c62d20ffb59241a13b714e951d
6e14ef9254097803e76b75e051866287
2cb6db296bbb2a7322b4d641d235b6e3
6426f086ecb6d0650ed61285a5e2a78b
f0f7b2352193c12cbff21ccc82054d00
a00a13d304d7d1365e955eeb30aece8f
15ca06b1c2f504de1ce03f9e955d17f6
a70db5635fd3d3fce914dc090a3f3d59
71db3e9955adf3797c50c50bbe0cbc4b
1aa8d3f363de18474eaeb0b7116edaba
00325fa6fd15da57ad10f9e81cf8d7f2
f1c16d95af55071365cefd8513c906af
e830c0c83f01eea30add98f734fd6011
f5c89c1822d516e0a0c3452c869a5940
929a37e3e064f307b17b8fbe8acb73c3
-----END OpenVPN Static key V1-----
</tls-auth>

# Jurnalları detallı görmək istəsək aşağıdaki sətirdən şərhi silirik
#verb 3
```

Qeyd: Işimizin topologiyasını, Domain Controller-də qrup yaradılması və ora istifadəçinin əlavə edilməsini, OpenVPN client proqramının istifadəsi qaydasını siz **OpenVPN-nin Active Directory ilə inteqrasiya edilməsi** başlığından oxuya bilərsiniz.

Ubuntu maşınımız yükləndikdən sonra onu yeniləyirik və lazımi paketləri yükləyirik:

```
apt-get update                  # Reposları yeniləyirik
apt-get dist-upgrade            # Kernel və mövcud paketləri yeniləyirik

# OpenVPN, OpenSSL, LDAP, və hər hal üçün RADIUS üçün inteqrasiya paketləri
yükləyirik
apt-get install openvpn easy-rsa openvpn-auth-ldap openvpn-auth-radius
openvpn-auth-radius-dbg
apt-get install freeradius freeradius-common freeradius-dbg freeradius-utils
freeradius-ldap

# LDAP utilitlərini yükləyirik
apt-get install ldap-utils
```

```
cd /etc/openvpn          # OpenVPN quraşdırma qovluğuna daxil olub, aşağıdakı
                         kimi konfig faylını yaradırıq
cat openvpn.conf         # Konfig faylımız aşağıdaki kimi olacaq
plugin /usr/lib/openvpn/openvpn-auth-ldap.so "/etc/openvpn/openvpn-auth-
ldap.conf"
proto tcp
port 1194
dev tun
server 192.168.200.0 255.255.255.0

# Açarlarının generasiya edilməsi qaydasını FreeBSD OpenVPN başlığından oxuya
bilərsiniz.
ca /etc/openvpn/keys/keys/ca.crt
cert /etc/openvpn/keys/keys/openvpnserver.crt
client-cert-not-required
key /etc/openvpn/keys/keys/openvpnserver.key
dh /etc/openvpn/keys/keys/dh2048.pem
tls-auth /etc/openvpn/keys/keys/ta.key 0

reneg-sec 86400
persist-key
persist-tun
keepalive 10 60

# Client-lərimizin hər birinə ayrı quraşdırma yazmaq istəsək aşağıdakı
sətirlərdən şərhi silirik.
#client-to-client
#client-config-dir /usr/local/etc/openvpn/ccd
push "route 10.50.2.0 255.255.255.0"
push "route 10.50.3.0 255.255.255.0"
push "route 10.50.12.0 255.255.255.0"
push "route 10.50.14.0 255.255.255.0"
push "route 10.50.17.0 255.255.255.0"
push "route 10.50.19.0 255.255.255.0"
push "route 192.168.10.0 255.255.255.0"
push "dhcp-option DNS 10.50.3.2"
push "dhcp-option DNS 10.50.3.3"
topology subnet

user root
group root

log-append /var/log/openvpn.log


Active Directory-ə qoşulmaq üçün LDAP quraşdırma faylımız isə aşağıdaki kimi
olacaq:
cat /etc/openvpn/openvpn-auth-ldap.conf          # LDAP qoşulmamız üçün
                                                 quraşdırma faylımız aşağıdakı
                                                 kimidir
<LDAP>
        URL             ldap://domain.lan
        BindDN          "CN=DCADM,CN=Users,DC=domain,DC=lan"
```

```
        Password        "DcP@$$f0rd"
        Timeout         15
        TLSEnable       no
        FollowReferrals no
</LDAP>

<Authorization>
        BaseDN          "DC=domain,DC=lan"
        SearchFilter    "(&(sAMAccountName=%u))"
        RequireGroup    true
        <Group>
                BaseDN          "DC=domain,DC=lan"
                SearchFilter    "(cn=OpenVpnFAUsers)"
                MemberAttribute "member"
        </Group>
</Authorization>
```

Ubuntu maşınımızda şəbəkə və routing quraşdırması aşağıdaki kimi olacaq:

```
cat /etc/network/interfaces                # Şəbəkə konfig faylı
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
        address 99.97.96.131
        netmask 255.255.255.240
        network 99.97.96.128
        broadcast 99.97.96.143
        gateway 99.97.96.129
        # dns-* options are implemented by the resolvconf package, if
installed
        dns-nameservers 10.90.3.2 10.90.3.3
        dns-search domain.az

auto eth1
iface eth1 inet static
        address 10.90.3.40
        netmask 255.255.255.0
        network 10.90.3.0
        broadcast 10.90.3.255
# Lazımi route-larımız
        up route add -net 10.90.2.0/24 gw 10.90.3.1
        up route add -net 10.90.12.0/24 gw 10.90.3.1
        up route add -net 10.90.14.0/24 gw 10.90.3.1
        up route add -net 10.90.17.0/24 gw 10.90.3.1
        up route add -net 10.90.19.0/24 gw 10.90.3.1
        up route add -net 192.168.10.0/24 gw 10.90.3.1
```

Qeyd: Unutmayın ki, yazdığınız routing eynilə şəbəkənizdə olan Router-in

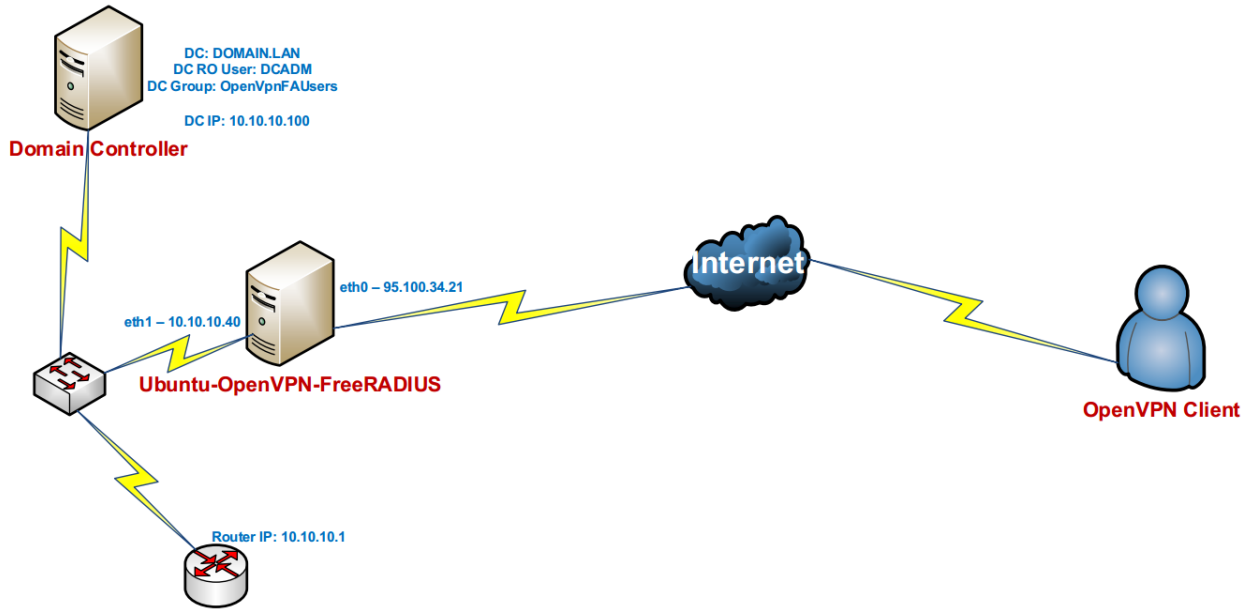
üzərindən geriyə qayıtmalıdır. Yəni sizin virtual VPN şəbəkənizin hamı tərəfindən görünməsi üçün router-nizdə aşağıdakina uyğun olan bir routing mütləq əlavə etməlisiniz(Yəni virtual 192.168.200.0/24 şəbəkəsini görmək üçün 10.90.3.40 IP-sindən keçmək lazımdır):
ip route 192.168.200.0 255.255.255.0 10.90.3.40

Həmçinin Ubuntu maşınınızı Routing rejimə salmalısınız. Bunu aşağıdaki kimi edəcəyik:
sysctl -w net.ipv4.ip_forward=1 # CLI-dan işə salırıq

reboot-dan sonra işləməsi üçün **/etc/sysctl.conf** faylında aşağıda sətirin qarşısından şərhi silirik:
net.ipv4.ip_forward=1

Ubuntu serverdə OpenVPN FreeRADIUS AD inteqrasiyası

Məqsədimiz Ubuntu 14.04 OS üzərində OpenVPN server qaldırmaq və onu FreeRADIUS ilə inteqrasiya etməkdir. Sonra isə FreeRADIUS serveri Active Directory ilə inteqrasiya edib, seçilmiş MS LDAP qrupdan istifadəçilərə yetki verməkdir. Şəbəkə strukturu aşağıdakı şəkildəki kimi olacaq:



Istifadə edilən OS-lar:

Windows 2012 Server R2 - DC
Windows 8.1 x64 - Client maşını
Ubuntu 14.04 x64 - OpenVPN

Öncə istifadə etdiyimiz Domain Controller-in verilənlərini açıqlayaq
Domain Controller: **domain.lan**
DC RO User: **DCADM**
DC RO PASS: **DcP@$$f0rD0m**
DC VPN Group: **OpenVPNFAUsers** - Tam yetkisi olan VPN istifadəçiləri(Bütün şəbəkəyə routing olacaq)

Windows 8.1 client quraşdırma faylı aşağıdaki kimi olacaq(faylımızın adı **domain-ad-auth.ovpn**). Faylın genişlənməsi mütləq **.ovpn** olmalıdır:

```
client
auth-user-pass
auth-nocache
proto tcp
remote ovpndc.domain.az
port 1194
dev tun
nobind

key-direction 1

ns-cert-type server
```

```
# OpenVPN serverdə yaradılan ca.crt
<ca>
-----BEGIN CERTIFICATE-----
MIIGxDCCBKygAwIBAgIJALsV/eQc/V5+MA0GCSqGSIb3DQEBBQUAMIGcMQswCQYD
VQQGEwJBWjENMAsGA1UECBMEQkFLVTEUMBIGA1UEBxMLWWVuaVlhc2FtYWwxFDAS
BgNVBAoTC0FUTEluZm9UZWNoMQswCQYDVQQLEwJJVDEXMBUGA1UEAxMOQVRMSW5m
b1RlY2ggQ0ExLDAqBgkqhkiG9w0BCQEWHWphbWFsLnNoYWh2ZXJkaXlldkBhdGx0
ZWNoLmF6MB4XDTE0MDYwODE3NDUzOFoXDTIzMDYwNjE3NDUzOFowgZwxCzAJBgNV
BAYTAkFaMQ0wCwYDVQQIEwRCQUtVMRQwEgYDVQQHEwtZZW5pWWFzYW1hbDEUMBIG
A1UEChMLQVRMSW5mb1RlY2gxCzAJBgNVBAsTAklUMRcwFQYDVQQDEw5BVExJbmZv
VGVjaCBDQTEsMCoGCSqGSIb3DQEJARYdamFtYWwuc2hhaHZlcmRpeWV2QGF0bHRl
Y2guYXowggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC0kYn6jZf/R1eA
Xs1YH/g36sIQJcxJBmcbXh/atZTy7W8rlXsCw05+RU7OaXrFQUEbed0lnjYiKfri
CutMpT5c7iY6fgfMMoPaIqk8q17qydk8HvqQoac3kjo9wMD7XWlDYiLFk1FxQjEW
BIqI2z6vh9/9ka54s6WNRgzT+7+EZqSuwCfC6Dm/0qxp4AvEjapwjaURJ6yEuQYe
Odh5ydTsIcueNnBzkuFZRx5O5iNcaBQZ2fUVpQvueTCCsHkPt1BGU3TqWIYTUVZl
O4wPQoOyXC9YUvWaYWSLTDMNDvCvGFYfc5C3++nijtfWpO8LLDZgiwC7ScYj+Boo
SZ9dkEpIYdb03KBnn+LCO3STVukpwTr+vyKjPITceuElHXdWvXi7wgtopwQhQ+3j
sDCvB+Wg2Bt5zBPC43WTeLANOGZFQN1f1kyBNXlBm1tM0kl3k75skkj9TXHjrM44
+aVdxlPjkQ86e6/A04wCUOBNf4aO0Q8r6PWCfpkqatDn6hCh6ChAYYuqAR5W3eRs
p2D3lAWGEHlBlf/+397E66f3ByHvPGQ5nlAQ3wI7q+tLH+qPsoFUKcyfEbctuYvG
D0+9jPhvxAAQwc4hBhn+TXRXPkaaaI89iiaJoiF1//R8kqs8t3yxpxjEy0hs2nrx
tboZl91cO2fj8e2HvhbMs9v+j6oVTQIDAQABo4IBBTCCAQEwHQYDVR0OBBYEFIci
KzboRxhacra8qkU+xvRM4df7MIHRBgNVHSMEgckwgcaAFIciKzboRxhacra8qkU+
xvRM4df7oYGipIGfMIGcMQswCQYDVQQGEwJBWjENMAsGA1UECBMEQkFLVTEUMBIG
A1UEBxMLWWVuaVlhc2FtYWwxFDASBgNVBAoTC0FUTEluZm9UZWNoMQswCQYDVQQL
EwJJVDEXMBUGA1UEAxMOQVRMSW5mb1RlY2ggQ0ExLDAqBgkqhkiG9w0BCQEWHWph
bWFsLnNoYWh2ZXJkaXlldkBhdGx0ZWNoLmF6ggkAuxX95Bz9Xn4wDAYDVR0TBAUw
AwEB/zANBgkqhkiG9w0BAQUFAAOCAgEAT+K70oaUfXDEfSFmBTrppvbcGqoVsaEl
5NjMh206D5KWtERhKbP7id20sdt6YgqlPQWW3I3thVQ0L686rhbZ/cR6Vzj41cFI
EqCt4uqZrkoMcvPq82POnvrzKCauxv5kmZJhWQTB3WXMo0A4KnQqW6/HVzSmbQgC
QR6CqNTt1Z21a1RIQrR1CmqRankKC4yQBKbzDwBlXLHvjITdyhJlHXZxBcdXurMX
Uh7AsHOTxbHy4nbyB+Zz1nO37wza6FBeunIqJ/I5eKDcN1lyGELjDsEvrSUcbRRg
IenV9/D9LP4y2KghMkiuDn7vhY3IifCjxQg3JWIa5BdQ/lU1Accsx1i0/nyQtZF7
5NadlwoSOjEe2H6bwxhlngcItQyiC34HghNKUF16eYLlMEzGkP7UNLwQN32b3IiA
q9+HTP6TQoci43AoaA3NFaUjuKC3zHykesNS8QqOH7MVB4L38/piaGD/K8CsiZH+
QhkICaJJ7hx/Cfp3VUIKr9yxtAnC5QNbXr9QVCC+mwi/sH9laThPlm1Xd2tKdoZa
My/K6o5fZnZSpzOeFa9j6bRgF2tpbG3jxiWT00F9xUv5EtXZdfies5BRHa1FYGK4
yvVIA/ZJBSB/6CT8mnMjGJcn85CcRggOrOc7lQNmgFKw/YopPYyAKzjgi1EKtNm3
pmPKIhXPdvc=
-----END CERTIFICATE-----
</ca>

# OpenVPN serverdə yaradılan TA açarı
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
7148f7b12478b04aee1445e18bb96509
b7f8d3c62d20ffb59241a13b714e951d
6e14ef9254097803e76b75e051866287
2cb6db296bbb2a7322b4d641d235b6e3
6426f086ecb6d0650ed61285a5e2a78b
```

```
f0f7b2352193c12cbff21ccc82054d00
a00a13d304d7d1365e955eeb30aece8f
15ca06b1c2f504de1ce03f9e955d17f6
a70db5635fd3d3fce914dc090a3f3d59
71db3e9955adf3797c50c50bbe0cbc4b
1aa8d3f363de18474eaeb0b7116edaba
00325fa6fd15da57ad10f9e81cf8d7f2
f1c16d95af55071365cefd8513c906af
e830c0c83f01eea30add98f734fd6011
f5c89c1822d516e0a0c3452c869a5940
929a37e3e064f307b17b8fbe8acb73c3
-----END OpenVPN Static key V1-----
</tls-auth>

# Jurnalları detallı görmək istəsək aşağıdaki sətirdən şərhi silirik
#verb 3
```

Qeyd: Domain Controller-də qrup yaradılması və ora istifadəçinin əlavə edilməsini, OpenVPN client proqramının istifadəsi qaydasını siz **OpenVPN-nin Active Directory ilə inteqrasiya edilməsi** başlığından oxuya bilərsiniz. Topologiya eynidir sadəcə, orda OpenVPN maşında FreeBSD-dir burda isə Ubuntu.

Ubuntu maşınımız yükləndikdən sonra onu yeniləyirik və lazımi paketləri yükləyirik:

```
apt-get update                  # Reposları yeniləyirik
apt-get dist-upgrade            # Kernel və mövcud paketləri yeniləyirik
```

OpenVPN, OpenSSL, LDAP, və hər hal üçün RADIUS üçün inteqrasiya paketləri yükləyirik:

```
apt-get install openvpn easy-rsa openvpn-auth-radius openvpn-auth-radius-dbg
```

FreeRADIUS-u da yükləyirik ki, eyni maşında RADIUS quraşdıraq:

```
apt-get install freeradius freeradius-common freeradius-dbg freeradius-utils
freeradius-ldap
```

LDAP utilitlərini yükləyirik

```
apt-get install ldap-utils
```

```
cd /etc/openvpn          # OpenVPN quraşdırma qovluğuna daxil olub, aşağıdakı
                         kimi konfig faylını yaradırıq

cat openvpn.conf         # Konfig faylımız aşağıdaki kimi olacaq
plugin /usr/lib/openvpn/radiusplugin.so /etc/openvpn/radiusplugin.cnf
proto tcp
port 1194
dev tun
server 192.168.200.0 255.255.255.0
```

```
# Açarlarının generasiya edilməsi qaydasını FreeBSD OpenVPN başlığından oxuya
bilərsiniz.
ca /etc/openvpn/keys/keys/ca.crt
cert /etc/openvpn/keys/keys/openvpnserver.crt
client-cert-not-required
key /etc/openvpn/keys/keys/openvpnserver.key
dh /etc/openvpn/keys/keys/dh2048.pem
tls-auth /etc/openvpn/keys/keys/ta.key 0

persist-key
persist-tun
keepalive 10 60

# Client-lərimizin hər birinə ayrı quraşdırma yazmaq istəsək aşağıdakı
sətirlərdən şərhi silirik.
#client-to-client
#client-config-dir /usr/local/etc/openvpn/ccd
push "route 10.99.2.0 255.255.255.0"
push "route 10.99.3.0 255.255.255.0"
push "route 10.99.12.0 255.255.255.0"
push "route 10.99.14.0 255.255.255.0"
push "route 10.99.17.0 255.255.255.0"
push "route 10.99.19.0 255.255.255.0"
push "route 192.168.10.0 255.255.255.0"
push "dhcp-option DNS 10.99.3.2"
push "dhcp-option DNS 10.99.3.3"
topology subnet

user root
group root

log-append /var/log/openvpn.log


OpenVPN ilə FreeRADIUS-u birləşdirən quraşdırma faylı isə aşağıdaki kimi
olacaq:
cat /etc/openvpn/radiusplugin.cnf          # RADIUS-a qoşulan quraşdırma fayl
NAS-Identifier=OpenVpn
Service-Type=5
Framed-Protocol=1
NAS-Port-Type=5
NAS-IP-Address=127.0.0.1
OpenVPNConfig=/etc/openvpn/openvpn.conf    # OpenVPN quraşdırma faylı
overwriteccfiles=true
server
{
        acctport=1813                      # RADIUS accounting portu
        authport=1812                      # RADIUS authentifikasiya portu
        name=127.0.0.1                     # RADIUS IP
        retry=1
        wait=1
        sharedsecret=freebsd               # FreeRADIUS ilə OpenVPN arasında
                                           istifadə edilən açar
```

```
}
```

Indi isə keçək FreeRADIUS-un quraşdırmasına:

OpenVPN client-i qoşulmaq üçün FreeRADIUS-un clientlər siyahısına əlavə edirik(Quraşdırma faylı aşağıdaki kimi olacaq):

```
cat /etc/freeradius/clients.conf          # Clientlər quraşdırma faylı
client localhost {
        ipaddr = 127.0.0.1                # OpenVPN server
        secret = freebsd                  # OpenVPN pass
        require_message_authenticator = no
        shortname = localhost
        nastype = other
}
```

FreeRADIUS-un öz quraşdırma faylı aşağıdaki kimi olacaq:

```
cat /etc/freeradius/radiusd.conf                # FreeRADIUS quraşdırma faylı
prefix = /usr
exec_prefix = /usr
sysconfdir = /etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log/freeradius
raddbdir = /etc/freeradius
radacctdir = ${logdir}/radacct
name = freeradius
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/${name}
db_dir = ${raddbdir}
libdir = /usr/lib/freeradius
pidfile = ${run_dir}/${name}.pid
user = freerad
group = freerad
max_request_time = 30
cleanup_delay = 5
max_requests = 1024
listen {
        type = auth
        ipaddr = 127.0.0.1
        port = 1812
}
listen {
        ipaddr = 127.0.0.1
        port = 1813
        type = acct
}
hostname_lookups = no
```

```
allow_core_dumps = no
regular_expressions     = yes
extended_expressions    = yes
log {
        destination = files
        file = ${logdir}/radius.log
        syslog_facility = daemon
        stripped_names = no
        auth = no
        auth_badpass = no
        auth_goodpass = no
}
checkrad = ${sbindir}/checkrad
security {
        max_attributes = 200
        reject_delay = 1
        status_server = yes
}
proxy_requests  = no
$INCLUDE proxy.conf
$INCLUDE clients.conf
thread pool {
        start_servers = 5
        max_servers = 32
        min_spare_servers = 3
        max_spare_servers = 10
        max_requests_per_server = 0
}
modules {
        $INCLUDE ${confdir}/modules/
        $INCLUDE eap.conf
}
instantiate {
        exec
        expr
        expiration
        logintime
}
$INCLUDE policy.conf
$INCLUDE sites-enabled/
```

FreeRADIUS ilə OpenVPN arasında olan qoşulmanın düzgünlüyünü test etmək üçün isə **/etc/freeradius/users** faylına aşağıdaki sətiri əlavə etmək lazımdır(Test üçün Vasif adlı istifadəçi freebsd şifrəsi ilə):

```
"vasif" Cleartext-Password := "freebsd"

/etc/init.d/openvpn restart          # OpenVPN serveri restart edirik
/etc/init.d/freeradius restart       # OpenVPN serveri restart edirik

freeradius -fX    # FreeRADIUS-u debug etmək üçün bu əmrdən istifadə edirik
```

Windows 8.1 client-dən OpenVPN serverə qoşulub uğurlu nəticə əldə etməlisiniz. Əgər uğurlu nəticə olmasa debug edilir. Əgər hər şey uğurlu olsa keçirik RADIUS-un MS LDAP-la inteqrasiya edilməsinə.

OpenVPN-in istifadecilerinin AD-den yoxlanilmasi ucun FreeRADIUS serveri MS LDAP ile inteqrasiya elemek lazimdir. Bunun ucun ashaqidakilari edirik:

```
cat /etc/freeradius/sites-enabled/default              # Susmaya gore olan
                                                       Virtual RADIUS-u
                                                       quraşdırırıq
authorize {
        files
        ldap                                           # Avtorizasiya LDAP-dan
        if (LDAP-Group == "OpenVpnFAUsers") {          # OpenVpnFAUsers DC
                                                       qrupundan alsaq her
                                                       shey OK-dir.
                ok
        }
        else {
                reject                                 # Eks halda baqlayiriq
        }
}
authenticate {
        Auth-Type LDAP {
                ldap    # Həmçinin authentifikasiya ldap qrupundan alacayıq
        }
}
preacct {
        preprocess
        acct_unique
        suffix
        files
}
accounting {
        detail
        unix
        radutmp
        exec
        attr_filter.accounting_response
}
session {
}
post-auth {
        exec
}
pre-proxy {
}
post-proxy {
}
```

LDAP modulunu quraşdırırıq ki, müraciət edən istifadəçilərin təyin edilməsi üçün, Domain Controller-ə qoşulub filter edə bilsin.

```
cat /etc/freeradius/modules/ldap                # LDAP modulunun quraşdırması
                                                aşağıdaki kimi olacaq
ldap {
        server = "domain.lan"
        identity = "CN=DCADM,CN=Users,DC=domain,DC=lan"
        password = "DcP@$$f0rD0m"
        basedn = "DC=domain,DC=lan"
        filter = "(sAMAccountName=%{%{Stripped-User-Name}:-%{User-Name}})"
        ldap_connections_number = 5
        timeout = 4
        timelimit = 3
        net_timeout = 1
        tls {
                start_tls = no
        }
        dictionary_mapping = ${confdir}/ldap.attrmap
        edir_account_policy_check = no
        groupname_attribute = "cn"
        groupmembership_filter =
"(|(&(objectClass=GroupOfNames)(member=%{control:Ldap-
UserDn}))(&(objectClass=GroupOfUniqueNames)(uniquemember=%{control:Ldap-
UserDn})))"
        groupmembership_attribute = "memberOf"
        compare_check_items = no
        do_xlat = yes
        access_attr_used_for_allow = yes
        chase_referrals = yes
        rebind = yes
        set_auth_type = yes
        ldap_debug = 0
        keepalive {
                idle = 60
                probes = 3
                interval = 3
        }
}


freeradius -fX                          # Debug rejimdə loglarında uğurlu nəticə
                                        olaraq seçdiyim aşağıdakı logların
                                        oxşarlarını sizdə mütləq görməlisiniz.

[ldap] performing user authorization for jamal
[ldap]  expand: %{Stripped-User-Name} ->
[ldap]  ... expanding second conditional
[ldap]  expand: %{User-Name} -> jamal
[ldap]  expand: (sAMAccountName=%{%{Stripped-User-Name}:-%{User-Name}}) ->
(sAMAccountName=jamal)
[ldap]  expand: DC=domain,DC=lan -> DC=domain,DC=lan
  [ldap] ldap_get_conn: Checking Id: 0
  [ldap] ldap_get_conn: Got Id: 0
```

```
  [ldap] attempting LDAP reconnection
  [ldap] (re)connect to domain.lan:389, authentication 0
  [ldap] bind as CN=DCADM,CN=Users,DC=domain,DC=lan/DcP@$$f0rD0m to
domain.lan:389
  [ldap] waiting for bind result ...
  [ldap] Bind was successful

[ldap] Setting Auth-Type = LDAP
[ldap] user jamal authorized to use remote access
  [ldap] ldap_release_conn: Release Id: 0
++[ldap] returns ok
++? if (LDAP-Group == "OpenVpnFAUsers")
  [ldap] Entering ldap_groupcmp()
        expand: DC=domain,DC=lan -> DC=domain,DC=lan
        expand: (|(&(objectClass=GroupOfNames)(member=%{control:Ldap-
UserDn}))(&(objectClass=GroupOfUniqueNames)(uniquemember=%{control:Ldap-
UserDn}))) -> (|(&(objectClass=GroupOfNames)(member=CN\3dJamal
Shahverdiyev\2cOU\3dDOMAINTech
Users\2cOU\3dDOMAINTech\2cDC\3ddomain\2cDC\3dlan))(&(objectClass=GroupOfUniqu
eNames)(uniquemember=CN\3dJamal Shahverdiyev\2cOU\3dDOMAINTech
Users\2cOU\3dDOMAINTech\2cDC\3ddomain\2cDC\3dlan)))
  [ldap] ldap_get_conn: Checking Id: 0
  [ldap] ldap_get_conn: Got Id: 0
  [ldap] performing search in DC=domain,DC=lan, with filter
(&(cn=OpenVpnFAUsers)(|(&(objectClass=GroupOfNames)(member=CN\3dJamal
Shahverdiyev\2cOU\3dDOMAINTech
Users\2cOU\3dDOMAINTech\2cDC\3ddomain\2cDC\3dlan))(&(objectClass=GroupOfUniqu
eNames)(uniquemember=CN\3dJamal Shahverdiyev\2cOU\3dDOMAINTech
Users\2cOU\3dDOMAINTech\2cDC\3ddomain\2cDC\3dlan))))

  [ldap] performing search in CN=Jamal Shahverdiyev,OU=DOMAINTech
Users,OU=DOMAINTech,DC=domain,DC=lan, with filter (objectclass=*)
  [ldap] performing search in CN=OpenVpnFAUsers,OU=DOMAINTech
Groups,OU=DOMAINTech,DC=domain,DC=lan, with filter (cn=OpenVpnFAUsers)
rlm_ldap::ldap_groupcmp: User found in group OpenVpnFAUsers
  [ldap] ldap_release_conn: Release Id: 0
? Evaluating (LDAP-Group == "OpenVpnFAUsers") -> TRUE
++? if (LDAP-Group == "OpenVpnFAUsers") -> TRUE
++- entering if (LDAP-Group == "OpenVpnFAUsers") {...}
+++[ok] returns ok

[ldap] user jamal authenticated succesfully
++[ldap] returns ok
```

Qeyd: Unutmayın ki, **/etc/freeradius/users** faylında heç bir istifadəçi qeyd edilməyib və fayl tamamilə boşdur.

```
/etc/init.d/freeradius start              # Sonda FreeRADIUS-u start edirik
```

BÖLÜM 8

Elektron poçt infrastrukturunun qurulması

- **FreeBSD Postfix Postfixadmin inteqrasiya edilməsi**
- **FreeBSD Postfix Dovecot ilə AD integrasiyası**

Hər bir şirkətin daxili poçt infrastrukturu olmalıdır, çünki müasir dövrdə bu qaçılmaz bir seçim və məcbur tələbdir. Daxili poçt infrastrukturu vasitəsilə şirkətin özünü bəyan edən domain suffiksi ilə digər bütün dövlət qurumları və kompaniyalara rəsmi məktublar yollanır və həmin suffikslə də məktublar qəbul edilir. Siz bu mail sisteminin qurulması üçün hər hansı bir pullu proqram təminatı ala bilər, ya da açıq qaynağı olan postfix-i qura bilərsiniz. Bu başlıqda Postfix-i Postfixadmin web interfeyslə quracayıq. Həmçinin postfix dovecot birləşməsi Active Directory ilə inteqrasiya ediləcək.

FreeBSD Postfix Postfixadmin inteqrasiya edilməsi

Məqsədimiz FreeBSD əməliyyat sistemi üzərində OpenSource mail serverin qurulmasıdır. Mail server tam funksionallıqla həm spamdan qorunmalı və həmdə istifadə komfortuna sahib olmalıdır. Mail serverimiz **saas.az** domain adı üzərində qurulmuşdur. Postfix proqram təminatı SMTP və SMTPS xidmətlərinə cavabdehdir. Dovecot isə POP3, POP3S, İMAP və İMAPS xidmətlərinə cavabdehdir. Mail serverin WEB browser vasitəsilə idarə edilməsi üçün, Postfixadmin və istifadəçi bazasının saxlanılması üçün MySQL verilənlər bazası yaradılacaq. MaiaMailGuard isə hər bir istifadəçi tərəfindən qurula biləcək və təhsil alma qabiliyyətinə sahib olan spam filterdir.

DNS-də olan olan zone faylımızın quraşdırma sətirləri aşağıdaki kimi olacaq:

```
$TTL 172800     ; 2 days
saas.az.        IN      SOA     ns1.saas.az. root.saas.az. (
                                2015092315      ; Serial
                                86400           ; Refresh
                                7200            ; Retry
                                604800          ; Expire
                                172800          ; Minimum TTL
                        )

; DNS Servers
                IN      NS      ns1.saas.az.
                IN      NS      ns2.saas.az.

; MX Records
                IN      MX 10   mail.saas.az.
                IN      A       155.123.145.97

; SRV
_jabber._tcp.jabber.saas.az. IN SRV   0       0       5269    jabber.saas.az.
_sip._tls.saas.az.           IN SRV   0       0       442     access.saas.az.
_sipfederationtls._tcp.saas.az.   IN  SRV     0 0     5061    access.saas.az.
_xmpp-client._tcp.jabber.saas.az. IN  SRV     0 0     5222    jabber.saas.az.
_xmpp-server._tcp.jabber.saas.az. IN  SRV     0 0     5269    jabber.saas.az.

; Machine Names
;localhost       IN      A       127.0.0.1
cloud           IN      A       155.123.145.97
conference.jabber IN    A       155.123.145.97
elearn          IN      A       155.123.145.97
fs              IN      A       155.123.145.97
fssip           IN      A       155.123.145.97
fscurl          IN      A       155.123.145.97
fussip          IN      A       155.123.145.97
jabber          IN      A       155.123.145.97
gts             IN      A       155.123.145.97
imap            IN      A       155.123.145.97
imaps           IN      A       155.123.145.97
lists           IN      A       155.123.145.97
maia            IN      A       155.123.145.97
```

```
mail            IN      A       155.123.145.97
mailman         IN      A       155.123.145.97
madmin          IN      A       155.123.145.97
mpanel          IN      A       155.123.145.97
moodle          IN      A       155.123.145.97
ns1             IN      A       85.132.57.58
ns2             IN      A       85.132.57.59
om              IN      A       155.123.145.97
pop3            IN      A       155.123.145.97
pop3s           IN      A       155.123.145.97
rainloop        IN      A       155.123.145.97
smpp            IN      A       155.123.145.97
smtp            IN      A       155.123.145.97
snort           IN      A       155.123.145.97
sip             IN      A       155.123.145.97
sqmail          IN      A       155.123.145.97
bbb             IN      A       155.123.145.97
openvpn         IN      A       155.123.145.97
; Aliases
www             IN      CNAME   @
```

portsnap fetch extract update - Sistemimizin portlarını yeniləyirik

Nəzərdə tutulur ki, **FAMP** artıq qurulmuş və hazır vəziyyətdədir.

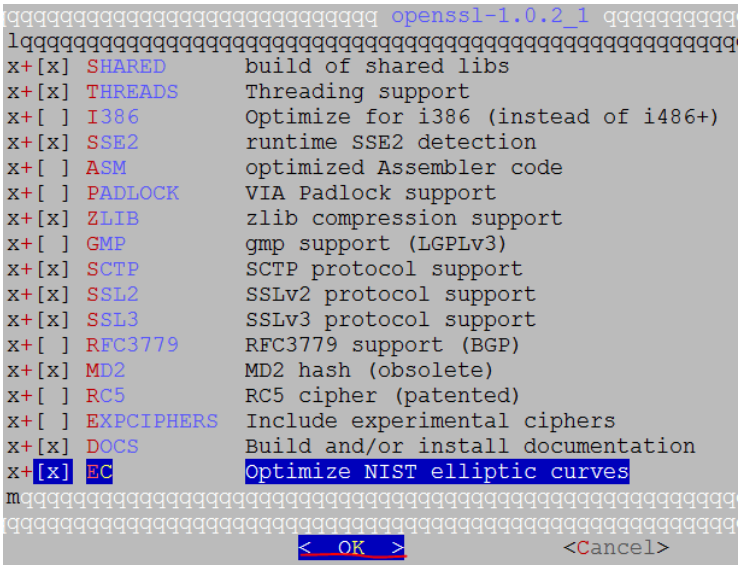
Yalnız MySQL-i qurduqda, **/etc/my.cnf** faylında aşağıdaki dəyişiklikləri uyğun olaraq etmək lazımdır:

```
[mysqld]
...
max_allowed_packet = 10M        - RoundCubde-da mail attachment tələb edəcək
...
innodb_data_home_dir = /var/db/mysql/
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /var/db/mysql/
...
innodb_buffer_pool_size = 16M
innodb_additional_mem_pool_size = 2M
...
innodb_log_file_size = 5M
innodb_log_buffer_size = 8M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50
```

```
mysql -uroot -p    - MySQL konsola daxil oluruq
mysql> CREATE DATABASE postfix;      - Postfix adlı baza yaradırıq
mysql> GRANT ALL PRIVILEGES ON postfix.* TO postfix@localhost IDENTIFIED BY
'postfixdbpass';              - Postfix adlı bazaya istifadəçi adı və
                                    şifrə yaradırıq
```

```
mysql> FLUSH PRIVILEGES;              - Yetkiləri sıfırlayırıq
```

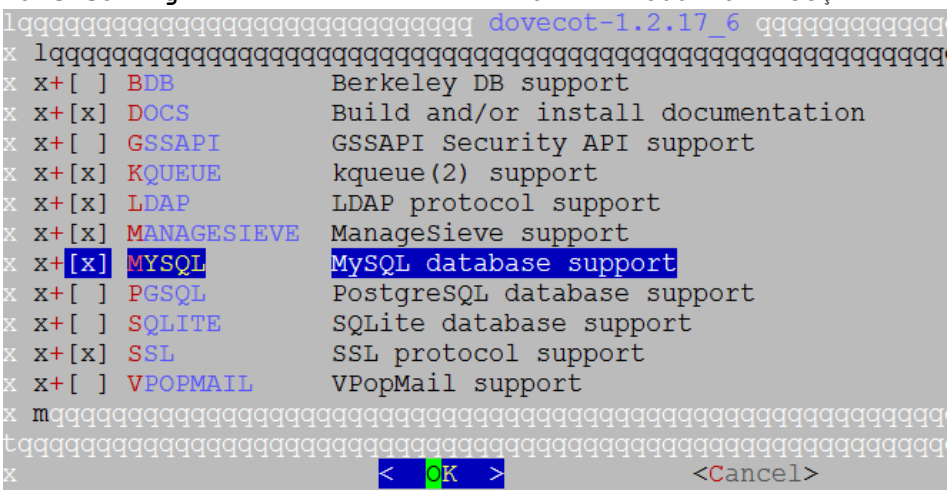
```
cd /usr/ports/security/openssl        - Portlardan OpenSSL-i yükləyirik
make config                           - Lazımi modulları seçirik
```




```
make install                          - Yükləyirik
```

Dovecot-u yukleyek ve qurashdiraq

```
cd /usr/ports/mail/dovecot            - IMAP/POP/IMAPS/POP3S üçün paketi yükləyirik
make config                           - Lazımi modulları seçirik
```




```
make install                          - Yükləyirik
```

```
echo 'dovecot_enable="YES"' >> /etc/rc.conf     - Dovecot servisi StartUP-a
                                                  əlavə edirik
```

cp /usr/local/share/examples/dovecot/dovecot.conf /usr/local/etc/dovecot.conf
- Quraşdırma faylını nüsxələyirik

cp /usr/local/share/examples/dovecot/dovecot-sql.conf /usr/local/etc/dovecot-sql.conf
- Bazaya qoşulmaq üçün quraşdırma faylını nüsxələyirik

mkdir /etc/ssl/dovecot - Dovecot sertifikat faylları üçün qovluq yaradırıq

cd /etc/ssl/dovecot - Dovecot-un SSL qovluğu faylına daxil oluruq

openssl req -new -x509 -nodes -out cert.pem -keyout key.pem -days 365 - Sertifikat generasiya edirik

```
Generating a 1024 bit RSA private key
..........................++++++
.++++++
writing new private key to 'key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:Baku
Locality Name (eg, city) []:Khatai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:OpSO
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:mail.saas.az
Email Address []:jamal.shahverdiyev@saas.az
```

/usr/local/etc/dovecot.conf quraşdırma faylını aşağıdaki şəklə gətiririk:

```
protocols = imap imaps pop3 pop3s
disable_plaintext_auth = no
ssl_cert_file = /etc/ssl/dovecot/cert.pem
ssl_key_file = /etc/ssl/dovecot/key.pem
login_greeting = ISP Mail Server Ready.
log_path = /var/log/dovecot.log
mail_debug = yes
verbose_ssl = yes
mail_location = maildir:/usr/local/virtual/%d/%n
first_valid_uid = 125
last_valid_uid = 125
first_valid_gid = 125
last_valid_gid = 125
```

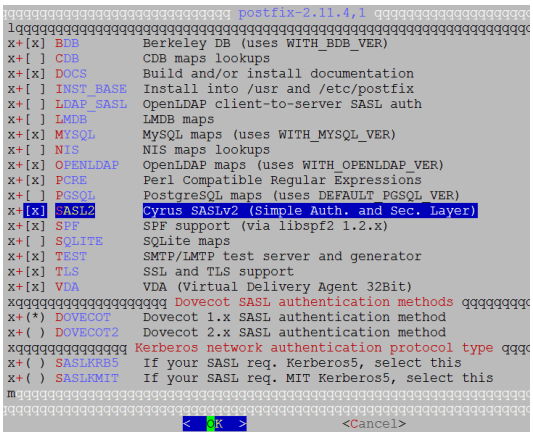
```
protocol imap {
  mail_plugins = quota imap_quota
}
protocol pop3 {
  mail_plugins = quota
}
protocol lda {
  postmaster_address = postmaster@saas.az
}
auth default {
  mechanisms = plain login
  passdb sql {
    args = /usr/local/etc/dovecot-sql.conf
  }
  userdb sql {
    args = /usr/local/etc/dovecot-sql.conf
  }
  socket listen {
    client {
      path = /var/spool/postfix/private/auth
      mode = 0660
      user = postfix
      group = postfix
    }
  }
}
```

/usr/local/etc/dovecot-sql.conf – Dovecot üçün SQL faylı yaradıb içinə aşağıdaki məzmunu əlavə edirik:

```
driver = mysql
connect = host=localhost dbname=postfix user=postfix password=postfixdbpass
default_pass_scheme = MD5
password_query = SELECT password FROM mailbox WHERE username = '%u'
user_query = SELECT maildir, 125 AS uid, 125 AS gid,
CONCAT('maildir:storage=', FLOOR( quota / 1024 ) ) AS quota FROM mailbox
WHERE username = '%u' AND active = '1'
```

Postfix-i yükləyək və quraşdıraq.

```
cd /usr/ports/mail/postfix          - Port ünvanına daxil oluruq
make config                         - Lazımi modulları seçirik
```



make install - Yükləyirik

Yüklənmədə sonda çıxan suala aşağıdaki kimi **y** cavabı veririk:

Would you like to activate Postfix in /etc/mail/mailer.conf [n]? **y**

/etc/rc.conf faylına aşağıdaki sətirləri əlavə edib sistemi yenidən yükləyirik ki, SendMAIL-i dayandıraq və SysLOG yalnız daxildə qulaq assın.

Disable SendMail
sendmail_enable="NONE"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
sendmail_cert_create="NO"
sendmail_rebuild_aliases="NO"
syslogd_enable="YES"
syslogd_program="/usr/sbin/syslogd"
syslogd_flags="-ss"

Ya da ki, sistemi yenidən yüklənmə etmədən aşağıdaki əmrlərlə sendmail-i dayandıra bilərsiniz:

sh
for i in `ps auxwww|grep sendmail|awk '{print $2}'`;do kill $i;done && exit

/etc/periodic.conf faylı yaradıb içinə aşağıdaki sətirləri əlavə edirik:

daily_clean_hoststat_enable="NO"
daily_status_mail_rejects_enable="NO"
daily_status_include_submit_mailq="NO"
daily_submit_queuerun="NO"

mkdir -p /etc/ssl/postfix - Postfix sertifikat faylları üçün qovluq yaradırıq

cd /etc/ssl/postfix - Qovluğa daxil oluruq

SMPT üçün SSL sertifikati yaradırıq:

openssl req -new -x509 -nodes -out smtpd.pem -keyout smtpd.pem -days 3650
Generating a 1024 bit RSA private key

```
..................................... ++++++++++++
writing new private key to 'smtpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:Baku
Locality Name (eg, city) []:Khatai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:OpSO
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:mail.saas.az
Email Address []:jamal.shahverdiyev@saas.az
```

chmod 640 /etc/ssl/postfix/smtpd.pem - Sertifikat faylına minimal yetkiləri veririk

chgrp -R postfix /etc/ssl/postfix - Sertifikat qrupunu postfix təyin edirik

/usr/local/etc/postfix/main.cf quraşdırma faylının məzmunu aşağıdaki kimi olacaq:

```
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
message_size_limit = 10000000
soft_bounce = no
broken_sasl_auth_clients = yes
inet_protocols = ipv4
smtpd_sender_restrictions = permit_sasl_authenticated, permit_mynetworks
smtpd_recipient_restrictions =
  permit_mynetworks,
  permit_sasl_authenticated,
  reject_non_fqdn_sender,
  reject_non_fqdn_recipient,
  reject_unauth_destination,
  reject_unauth_pipelining,
  reject_invalid_hostname,

smtpd_sasl_auth_enable = yes
smtpd_sasl_authenticated_header = yes
smtpd_sasl_local_domain = $myhostname
smtpd_sasl_security_options = noanonymous
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtp_use_tls = yes
smtpd_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /etc/ssl/postfix/smtpd.pem
smtpd_tls_cert_file = /etc/ssl/postfix/smtpd.pem
```

```
smtpd_tls_CAfile = /etc/ssl/postfix/smtpd.pem
smtpd_tls_loglevel = 0
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
content_filter=smtp-amavis:[127.0.0.1]:10024
queue_directory = /var/spool/postfix
tls_random_source = dev:/dev/urandom
transport_maps = hash:/usr/local/etc/postfix/transport
vacation_destination_recipient_limit = 1
mailman_destination_recipient_limit = 1
virtual_alias_maps = mysql:/usr/local/etc/postfix/mysql_virtual_alias_maps.cf
virtual_gid_maps = static:125
virtual_mailbox_base = /usr/local/virtual
virtual_mailbox_domains =
mysql:/usr/local/etc/postfix/mysql_virtual_domains_maps.cf
virtual_mailbox_limit = 51200000
virtual_mailbox_maps =
mysql:/usr/local/etc/postfix/mysql_virtual_mailbox_maps.cf
virtual_minimum_uid = 125
virtual_transport = virtual
virtual_uid_maps = static:125
virtual_mailbox_limit_maps =
mysql:/usr/local/etc/postfix/mysql_virtual_mailbox_limit_maps.cf
proxy_read_maps = $local_recipient_maps $mydestination $virtual_alias_maps
  $virtual_alias_domains $virtual_mailbox_maps $virtual_mailbox_domains
  $relay_recipient_maps $relay_domains $canonical_maps $sender_canonical_maps
  $recipient_canonical_maps $relocated_maps $transport_maps $mynetworks
  $virtual_mailbox_limit_maps
virtual_mailbox_limit_override = yes
virtual_maildir_limit_message = Uzr isteyrik, bu istifadeci ucun teyin
edilmish disk mehdidiyyeti oz heddine catmishdir. Xahish olunur birazdan
yoxlayasiniz.
virtual_overquota_bounce = yes
relay_domains = mysql:/usr/local/etc/postfix/mysql_relay_domains_maps.cf
lists.saas.az
```

/usr/local/etc/postfix/master.cf faylında SMTPS bölümünü aşağıdaki şəklə gətirib yadda saxlayaraq çıxırıq:

```
smtps     inet  n       -       n       -       -       smtpd
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
```

/usr/local/etc/postfix/mysql_virtual_alias_maps.cf faylı yaradaq və məzmununa aşağıdaki sintaksisə uyqun olaraq, yaratdığımız postfix bazası ilə istifadəçi verilənlərini daxil edək:

```
user = postfix
password = postfixdbpass
hosts = localhost
dbname = postfix
query = SELECT goto FROM alias WHERE address='%s' AND active = '1'
```

/usr/local/etc/postfix/mysql_virtual_domains_maps.cf faylı yaradaq və məzmununa aşağıdaki sintaksisə uyğun olaraq, yaratdığımız postfix bazası ilə istifadəçi verilənlərini daxil edək:

```
user = postfix
password = postfixdbpass
hosts = localhost
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' and backupmx = '0' and
active = '1'
```

/usr/local/etc/postfix/mysql_virtual_mailbox_maps.cf faylı yaradaq və məzmununa aşağıdaki sintaksisə uyğun olaraq, yaratdığımız postfix bazası ilə istifadəçi verilənlərini daxil edək:

```
user = postfix
password = postfixdbpass
hosts = localhost
dbname = postfix
query = SELECT maildir FROM mailbox WHERE username='%s' AND active = '1'
```

/usr/local/etc/postfix/mysql_virtual_mailbox_limit_maps.cf faylı yaradaq və məzmununa aşağıdaki sintaksisə uyğun olaraq, yaratdığımız postfix bazası ilə istifadəçi verilənlərini daxil edək:

```
user = postfix
password = postfixdbpass
hosts = localhost
dbname = postfix
query = SELECT quota FROM mailbox WHERE username='%s'
```

/usr/local/etc/postfix/mysql_relay_domains_maps.cf faylı yaradaq və məzmununa aşağıdaki sintaksisə uyğun olaraq, yaratdığımız postfix bazası ilə istifadəçi verilənlərini daxil edək:

```
user = postfix
password = postfixdbpass
hosts = localhost
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' and backupmx = '1'
```

Şifrələr olan fayllarda təhlükəsizliyi təmin edirik:

```
chmod 640 /usr/local/etc/postfix/mysql_*
chgrp postfix /usr/local/etc/postfix/mysql_*
```

Transport uyğunluğu üçün bazanı yeniləyirik:

```
touch /usr/local/etc/postfix/transport
postmap /usr/local/etc/postfix/transport
```

/etc/aliases faylında dəyişiklik edərək sonuna aşağıdaki sətiri əlavə edək ki, root istifadəçisinə gələn sistem mesajlarının göndərilə bilməsi üçün düzgün email ünvanı yazaq.

root: admin@saas.az

/usr/bin/newaliases - **aliases.db** faylı bu əmrlə yaradırıq

/usr/local/etc/postfix/master.cf faylının ümümi məzmunu aşağıdaki kimi olacaq(Bu məzmun yükləmə proseduramuzla yavaş-yavaş doldurulacaq):

```
vacation  unix  -       n       n       -       -       pipe
  flags=DRhu user=vacation argv=/var/spool/vacation/vacation.pl
smtp      inet  n       -       n       -       -       smtpd
smtps     inet  n       -       n       -       -       smtpd
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
pickup    unix  n       -       n       60      1       pickup
cleanup   unix  n       -       n       -       0       cleanup
qmgr      unix  n       -       n       300     1       qmgr
tlsmgr    unix  -       -       n       1000?   1       tlsmgr
rewrite   unix  -       -       n       -       -       trivial-rewrite
bounce    unix  -       -       n       -       0       bounce
defer     unix  -       -       n       -       0       bounce
trace     unix  -       -       n       -       0       bounce
verify    unix  -       -       n       -       1       verify
flush     unix  n       -       n       1000?   0       flush
proxymap  unix  -       -       n       -       -       proxymap
proxywrite unix -       -       n       -       1       proxymap
smtp      unix  -       -       n       -       -       smtp
relay     unix  -       -       n       -       -       smtp
showq     unix  n       -       n       -       -       showq
error     unix  -       -       n       -       -       error
retry     unix  -       -       n       -       -       error
discard   unix  -       -       n       -       -       discard
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       n       -       -       lmtp
anvil     unix  -       -       n       -       1       anvil
scache    unix  -       -       n       -       1       scache
mailman   unix  -       n       n       -       -       pipe
  flags=FR user=mailman:mailman argv=/usr/local/mailman/postfix-to-mailman.py
  ${nexthop} ${user}
smtp-amavis unix -      -       n       -       2       smtp
      -o smtp_data_done_timeout=2400
      -o smtp_send_xforward_command=yes
      -o disable_dns_lookups=yes
      -o max_use=20
127.0.0.1:10025 inet    n       -       n       -       -       smtpd
      -o content_filter=
      -o local_recipient_maps=
      -o relay_recipient_maps=
```

```
        -o smtpd_restriction_classes=
        -o smtpd_delay_reject=no
        -o smtpd_client_restrictions=permit_mynetworks,reject
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks_style=host
        -o mynetworks=127.0.0.0/8
        -o strict_rfc821_envelopes=yes
        -o smtpd_error_sleep_time=0
        -o smtpd_soft_error_limit=1001
        -o smtpd_hard_error_limit=1000
        -o smtpd_client_connection_count_limit=0
        -o smtpd_client_connection_rate_limit=0
        -o
receive_override_options=no_header_body_checks,no_unknown_recipient_checks,no
_address_mappings
```

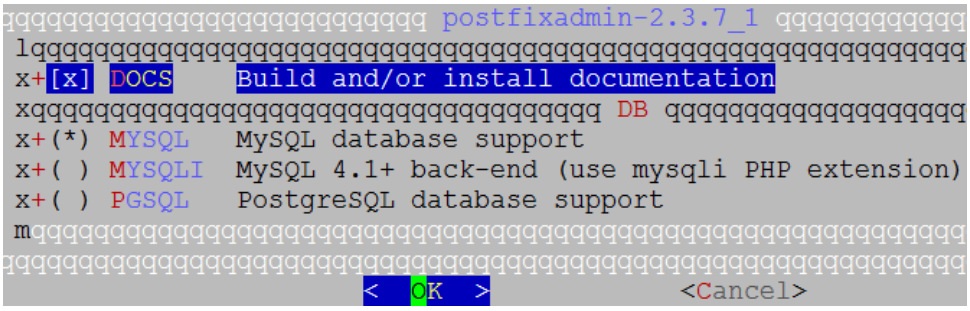
Virtual domainlər və onların məktub yeşikləri üçün virtual qovluqları yaradaq. Və qovluğa lazımi yetkiləri təyin edək.

```
mkdir /usr/local/virtual
chown -R postfix:postfix /usr/local/virtual
chmod -R 700 /usr/local/virtual
```

PostfixAdmin yüklənməsi və qurulması

PostfixAdmin – Bizim domainlərimizin və istifadəçilərimizin idarə edilməsi üçün əla alətdir. Çoxlu opsiyaya sahibdir və proqramın quraşdırılması işini asanlaşdırır. Proqram haqqında daha da ətraflı oxumaq istəsəniz, http://sourceforge.net/projects/postfixadmin/ linkinə baxa bilərsiniz. Hal-hazırda **2.3.7_1** versiyasını yükləyirik.

```
cd /usr/ports/mail/postfixadmin/     - Port ünvanına daxil oluruq
make config                          - Lazımi modulları seçirik
```




```
make install                         - Yükləyirik
```

PostfixAdmin öncədən postfix üçün yaratdığımız bazaya qoşulur. Əgər xatırlamırsınızsa, aşağıdaki əmrlərlə bazanı yarada bilərsiniz:

```
mysql -uroot -p    - MySQL konsola daxil oluruq
mysql> CREATE DATABASE postfix;      - Postfix adlı baza yaradırıq
```

```
mysql> GRANT ALL PRIVILEGES ON postfix.* TO postfix@localhost IDENTIFIED BY
'postfixdbpass';              - Postfix adlı bazaya istifadəçi adı və
                                    şifrə yaradırıq
mysql> FLUSH PRIVILEGES;            - Yetkiləri sıfırlayırıq
```

PostfixAdmin fayllarina lazımi yetkiləri təyin edərək müdafiə edirik:

```
cd /usr/local/www/postfixadmin
find . -type f -exec chmod 640 {} \;
find . -type d -exec chmod 750 {} \;
```

/usr/local/www/postfixadmin/config.inc.php faylında dəyişiklik edərək aşağıdaki şəklə gətiririk:

```
<?php
$CONF['configured'] = true;
// Aşağıdaki göstərilən MD5-də olan şifrə postfixadminin web vasitəsilə ilk
// inzibatçı hesabın əlavə edilməsi zamanı əldə ediləcək və sonra
// burda qeyd ediləcək.
$CONF['setup_password'] =
'bb6fe8e8ff6a155c0edb1d7b9f437315:f87d8d3325386a8ee91fb75fe26a30de3dcb7106';
$CONF['postfix_admin_path'] = dirname(__FILE__);
$CONF['default_language'] = 'en';
$CONF['database_type'] = 'mysql';
$CONF['database_host'] = 'localhost';
$CONF['database_user'] = 'postfix';
$CONF['database_password'] = 'postfixdbpass';
$CONF['database_name'] = 'postfix';
$CONF['database_prefix'] = '';
$CONF['database_tables'] = array (
    'admin' => 'admin',
    'alias' => 'alias',
    'alias_domain' => 'alias_domain',
    'config' => 'config',
    'domain' => 'domain',
    'domain_admins' => 'domain_admins',
    'fetchmail' => 'fetchmail',
    'log' => 'log',
    'mailbox' => 'mailbox',
    'vacation' => 'vacation',
    'vacation_notification' => 'vacation_notification',
    'quota' => 'quota',
    'quota2' => 'quota2',
);
$CONF['admin_email'] = 'postmaster@saas.az';
$CONF['smtp_server'] = 'mail.saas.az';
$CONF['smtp_port'] = '25';
$CONF['encrypt'] = 'md5crypt';
$CONF['authlib_default_flavor'] = 'md5raw';
$CONF['dovecotpw'] = "/usr/sbin/dovecotpw";
$CONF['min_password_length'] = 8;
$CONF['generate_password'] = 'YES';
$CONF['show_password'] = 'YES';
```

```
$CONF['page_size'] = '100';
$CONF['default_aliases'] = array (
    'abuse' => 'abuse@saas.az',
    'hostmaster' => 'hostmaster@saas.az',
    'postmaster' => 'postmaster@saas.az',
    'webmaster' => 'webmaster@saas.az'
);
$CONF['domain_path'] = 'YES';
$CONF['domain_in_mailbox'] = 'NO';
$CONF['maildir_name_hook'] = 'NO';
$CONF['aliases'] = '100';
$CONF['mailboxes'] = '500';
$CONF['maxquota'] = '3000';
$CONF['quota'] = 'YES';
$CONF['quota_multiplier'] = '1024000';
$CONF['transport'] = 'NO';
$CONF['transport_options'] = array (
    'virtual',
    'local',
    'relay'
);
$CONF['transport_default'] = 'virtual';
$CONF['vacation'] = 'YES';
$CONF['vacation_domain'] = 'autoreply.saas.az';
$CONF['vacation_control'] = 'YES';
$CONF['vacation_control_admin'] = 'YES';
$CONF['alias_control'] = 'YES';
$CONF['alias_control_admin'] = 'NO';
$CONF['special_alias_control'] = 'NO';
$CONF['alias_goto_limit'] = '0';
$CONF['alias_domain'] = 'YES';
$CONF['backup'] = 'YES';
$CONF['sendmail'] = 'YES';
$CONF['logging'] = 'YES';
$CONF['fetchmail'] = 'YES';
$CONF['fetchmail_extra_options'] = 'NO';
$CONF['show_header_text'] = 'NO';
$CONF['header_text'] = ':: Postfix Admin ::';
$CONF['show_footer_text'] = 'YES';
$CONF['welcome_text'] = <<<EOM
Salam,
Yeni hesabiniza xosh gelmishsiniz.
EOM;
$CONF['emailcheck_resolve_domain']= 'YES';
$CONF['show_status']= 'YES';
$CONF['show_status_key']= 'NO';
$CONF['show_status_text']= '  ';
$CONF['show_undeliverable']= 'NO';
$CONF['show_undeliverable_color']= 'tomato';
$CONF['show_undeliverable_exceptions']=
array("unixmail.domain.ext","exchangeserver.domain.ext","gmail.com");
$CONF['show_popimap']= 'NO';
$CONF['show_popimap_color']= 'darkgrey';
```

```
$CONF['show_custom_domains']= array("subdomain.domain.ext","domain2.ext");
$CONF['show_custom_colors']= array("lightgreen","lightblue");
$CONF['recipient_delimiter'] = "";
$CONF['create_mailbox_subdirs_prefix']= '';
$CONF['used_quotas'] = 'NO';
$CONF['new_quota_table'] = 'NO';
$CONF['theme_logo'] = 'images/logo-default.png';
$CONF['theme_css'] = 'css/default.css';
$CONF['xmlrpc_enabled'] = true;
if (file_exists(dirname(__FILE__) . '/config.local.php')) {
    include(dirname(__FILE__) . '/config.local.php');
}
```

Vacation adlı istifadəçi və qrup yaradaq:

```
pw groupadd vacation
pw useradd vacation -c Virtual\ Vacation -d /nonexistent -g vacation -s
/sbin/nologin
```

Vacation qovluğu yaradaraq lazımi scripti ora nüsxələyək və ardınca yetkiləri qovluğa təyin edək. Həmçinin vacation üçün **log** və **debug** faylları yaradıb lazımi yetkiləri təyin edək:

```
mkdir /var/spool/vacation
cp /usr/local/www/postfixadmin/VIRTUAL_VACATION/vacation.pl
/var/spool/vacation/
chown -R vacation:vacation /var/spool/vacation/
chmod 700 /var/spool/vacation/
chmod 750 /var/spool/vacation/vacation.pl
touch /var/log/vacation.log /var/log/vacation-debug.log
chown vacation:vacation /var/log/vacation*
```

/var/spool/vacation/vacation.pl scriptində aşağıdaki sətirlərdə uyğun olaraq dəyişiklik edirik:

```
our $db_type = 'mysql';
our $db_host = 'localhost';
our $db_username = 'postfix';
our $db_password = 'postfixdbpass';
our $db_name = 'postfix';
our $vacation_domain = 'autoreply.saas.az';
our $smtp_server = 'localhost';
our $smtp_server_port = 25;
our $logfile = "/var/log/vacation.log";
our $debugfile = "/var/log/vacation-debug.log";
our $syslog = 1;
```

/usr/local/etc/postfix/master.cf faylının əvvəlinə vacation filter əlavə edirik:

```
vacation  unix  -       n       n       -       -       pipe
  flags=DRhu user=vacation argv=/var/spool/vacation/vacation.pl
```

/usr/local/etc/postfix/main.cf faylına aşağıdaki sətirləri əlavə etməyi unutmayın(Ancaq biz öncədən postfix qurulmasında bu sətirləri nəzərə almışdıq)

```
transport_maps = hash:/usr/local/etc/postfix/transport
vacation_destination_recipient_limit = 1
```

Transport faylına lazımi sətirimizi əlavə edirik:

```
echo 'autoreply.saas.az vacation:' >> /usr/local/etc/postfix/transport
```

```
postmap /usr/local/etc/postfix/transport  - Postfix üçün transport bazası
                                            yaradırıq
```

/usr/local/domen/mail.saas.az faylına aşağıdaki sətirləri əlavə edirik:

```
<VirtualHost *:80>
        ServerAdmin jamal.shahverdiyev@saas.az
        ServerName mail.saas.az
        AcceptPathInfo On
        DocumentRoot /usr/local/www/postfixadmin/
<Directory "/usr/local/www/postfixadmin">
        AllowOverride All
        Require all granted
</Directory>
        ErrorLog /var/log/httpd/mail-error.log
        CustomLog /var/log/httpd/mail-access.log combined
</VirtualHost>
```

```
mkdir /var/log/httpd/          - Jurnallar üçün qovluq və fayllar yaradırıq
touch /var/log/httpd/mail-error.log /var/log/httpd/mail-access.log
```

```
chown -R www:www /usr/local/www/postfixadmin/    - PostfixAdmin qovluğunun
                                                   yetkilərini apache üçün təyin
                                                   edirik
```

```
apachectl configtest     - Apache quraşdırmalarını yoxlayırıq
apachectl graceful       - Apache httpd daemonu yenidən işə salırıq
```

Lazım olan proqramları işə salırıq:

```
/usr/local/etc/rc.d/mysql-server start
/usr/local/etc/rc.d/dovecot start
/usr/local/etc/rc.d/postfix start
```

Səhvlər üçün **/var/log/messages** və **/var/log/maillog** fayllarını araşdırın.

Postfixadmin inzibatçısı təyin edək və test edək:
http://mail.saas.az/setup.php linkinə daxil oluruq və şəkildəki kimi **hash** şifrəni generasiya edirik:

postfix.admin

Postfix Admin Setup Checker

Running software:

- PHP version 5.4.40
- Apache/2.4.12 (FreeBSD) OpenSSL/1.0.1j-freebsd PHP/5.4.40

Checking for dependencies:

- Magic Quotes: Disabled - OK
- Depends on: presence config.inc.php - OK
- Checking $CONF['configured'] - OK
- Depends on: MySQL 3.23, 4.0 - OK
- Depends on: MySQL 4.1 - OK (change the database_type to 'mysqli' in config.inc.php!!)
- Testing database connection - OK - mysql://postfix:xxxxx@localhost/postfix
- Depends on: session - OK
- Depends on: pcre - OK
- Depends on: multibyte string - OK
- Depends on: IMAP functions - OK

Everything seems fine... attempting to create/update database structure

Updating database:

- old version: 0; target version: 740

updating to version 1 (MySQL)... done

updating to version 2 (MySQL)... done

updating to version 3 (MySQL)... done

updating to version 4 (MySQL)... done

updating to version 5 (MySQL)... done

updating to version 79 (MySQL)... done

updating to version 81 (MySQL)... done

updating to version 90 (all databases)... done

updating to version 169 (MySQL)... done

updating to version 318 (MySQL)... done

updating to version 344 (MySQL)... done

updating to version 373 (MySQL)... done

updating to version 438 (MySQL)... done

updating to version 439 (MySQL)... done

updating to version 473 (MySQL)... done

updating to version 479 (MySQL)... done

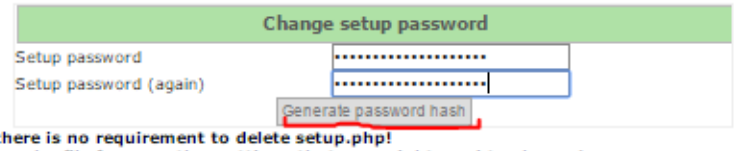
updating to version 483 (MySQL)... done

updating to version 495 (MySQL)... done

updating to version 504 (MySQL)... done

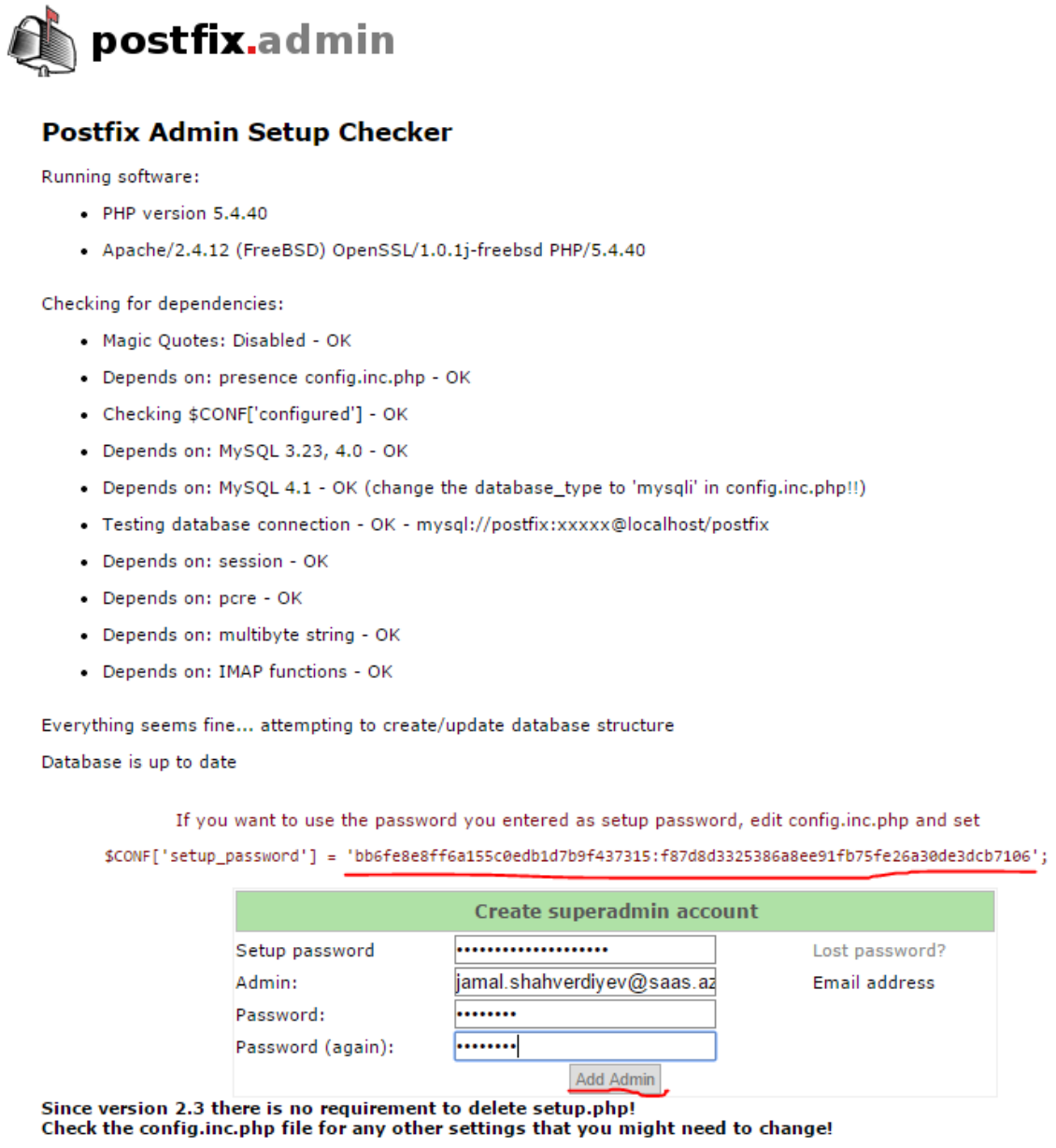
updating to version 655 (all databases)... done

updating to version 729 (all databases)... done

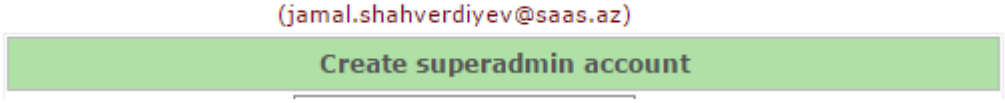


Since version 2.3 there is no requirement to delete setup.php!
Check the config.inc.php file for any other settings that you might need to change!

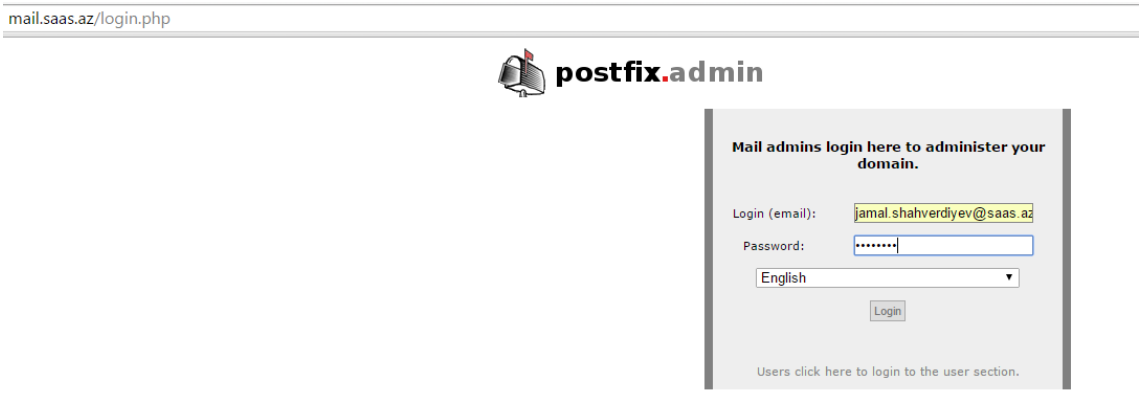
Sonra isə aşağıdaki şəkildəki kimi, generasiya edilmiş hash şifrəsini **/usr/local/www/postfixadmin/config.inc.php** faylının **$CONF['setup_password']** sətirində təyin edirik və inzibatçı əlavə edərək ona şifrə təyin edirik.



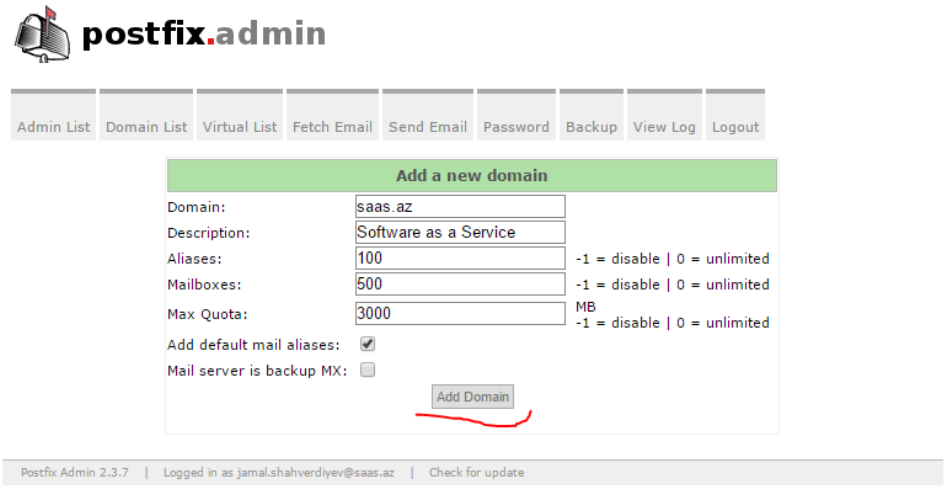
Nəticədə aşağıdakı şəkili əldə edirik və **/usr/local/www/postfixadmin/setup.php** faylını serverimizdən başqa bir ünvana köçürürük.



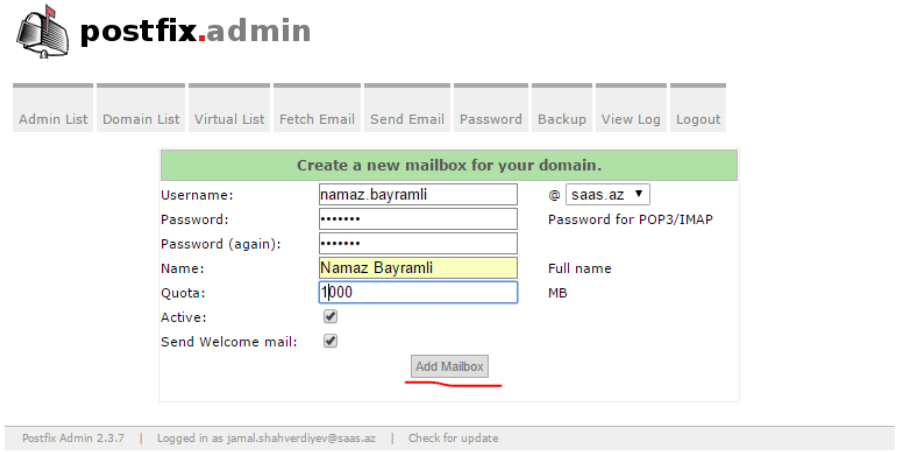
Artıq aşağıdaki şəkildəki kimi, mail serverimizin inzibatçı interfeysinə http://mail.saas.az linki ilə daxil oluruq:



Ardınca isə **Domain List -> New Domain** düyməsini sıxıb, şəkildəki kimi yeni domain əlavə edirik və **Add Domain** düyməsinə sıxırıq:



Sonra **Virtual List -> Add Mailbox** düyməsinə sıxıb, istifadəçi verilənlərini daxil edirik və **Add Mailbox** düyməsinə sıxırıq(Hələki mail istifadəçiyə getməyəcək çünki smtp-amavis hazır deyil):



25-ci portumuza **telnet** ataraq test edək:

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 raos.localdomain ESMTP Postfix
EHLO saas.az
250-raos.localdomain
250-PIPELINING
250-SIZE 10000000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
STARTTLS
220 2.0.0 Ready to start TLS
quit
quit
```

465-ci portumuza telnet ataraq test edirik(Siz bu qoşulmada fərqli heç bir nəticə əldə etməyəcəksiniz çünki, iş üçün SSL şifrələnmə tələb edilir və siz bunu indi istifadə etmirsiniz. Əgər qoşulma varsa, bu artıq uğurlu nəticədir və novbəti yoxlanışları SMTP SSL vasitəsilə edəcəyik):

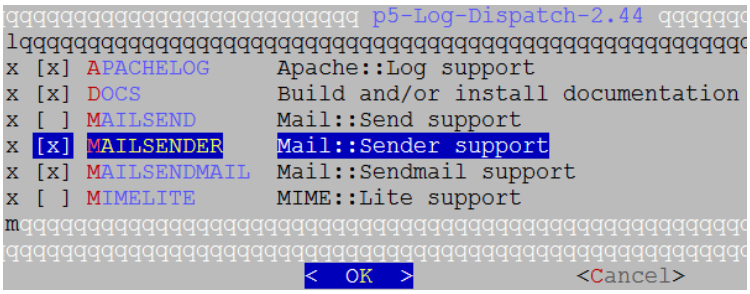
```
telnet localhost 465
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
quit
quit
```

telnet vasitəsilə **110**-cu porta qoşulaq:

```
telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK SAAS mail serveri hazirdir.
user namaz.bayramli@saas.az
+OK
pass freebsd
+OK Logged in.
quit
+OK Logging out.
Connection closed by foreign host.
```

vacation.pl scripti üçün bəzi portları yükləyirik(Yüklədikdə, opsiyalarda **MAIL-SENDER**-i seçməyi unutmayın):

```
cd /usr/ports/mail/p5-MIME-EncWords
make install clean

cd /usr/ports/mail/p5-Email-Valid
make install clean

cd /usr/ports/mail/p5-Mail-Sender
make install clean

cd /usr/ports/devel/p5-Log-Log4perl
make install clean

cd /usr/ports/devel/p5-Log-Dispatch
make config              - Lazımi modullar seçirik
```




```
make install clean
```

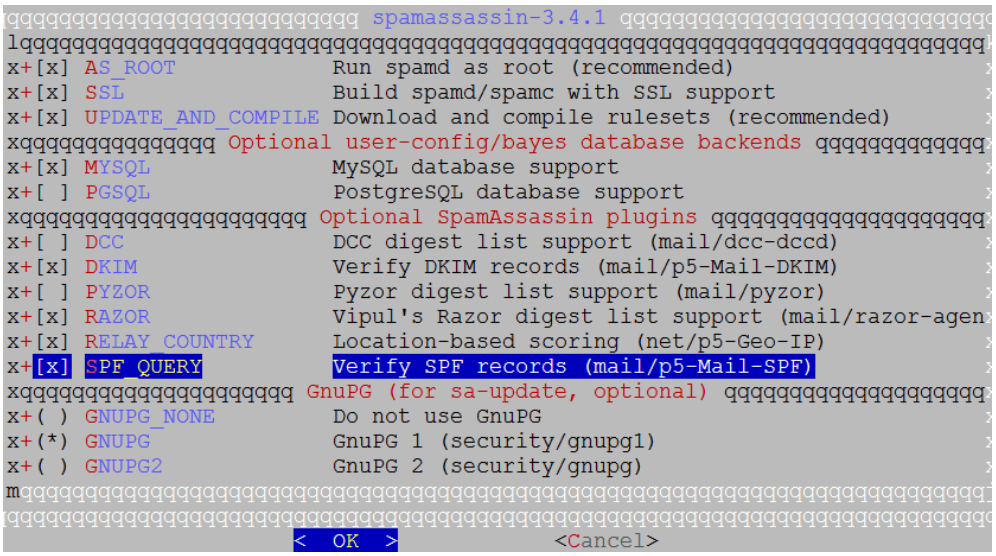
/etc/hosts faylına aşağıdakı sətiri əlavə edirik:
127.0.0.1 autoreply.saas.az

Vacation haqqında dahada ətraflı **/usr/local/www/postfixadmin/VIRTUAL_VACATION/INSTALL.TXT** faylından oxuya bilərsiniz.

SpamAssassin-i yuklemesi ve qurulmasi

SpamAssassin – Spamla mübarizə aparmaq üçün elan proqram təminatıdır. Ancaq Spamd-nidə tərifləyirlər. Spamassassin haqqında ətraflı oxumaq istəsəniz, məlumatı http://spamassassin.apache.org/ linkindən əldə edə bilərsiniz.

```
cd /usr/ports/mail/spamassassin/     - Port ünvanına daxil oluruq
make config                          - Lazımi modulları seçirik
```



make install - Yükləyirik

Sonda çıxan suala Yes cavabı veririk
config: no rules were found! Do you need to run 'sa-update'? **Y**

vScan adlı istifadəçi yaradırıq
pw groupadd vscan
pw useradd -n vscan -c Amavisd\ User -d /var/amavisd -g vscan -m
passwd vscan

/usr/local/etc/mail/spamassassin/local.cf adlı fayl yaradırıq və məzmununa aşağıdakı sətirləri əlavə edirik:

```
use_bayes 1
    bayes_store_module Mail::SpamAssassin::BayesStore::MySQL
    bayes_sql_dsn DBI:mysql:maia
    bayes_sql_username vscan
    bayes_sql_password freebsd
    auto_whitelist_factory Mail::SpamAssassin::SQLBasedAddrList
    user_awl_dsn DBI:mysql:maia
    user_awl_sql_username vscan
    user_awl_sql_password freebsd
    user_razor2 1
    razor_timeout 10
    bayes_auto_expire 0
    internal_networks 192.168.1.0/24
    trusted_networks 192.168.1/24
```

Həmçinin **/usr/local/etc/mail/spamassassin/v310.pre** faylında **razor2** aktiv edildiyinə görə, aşağıdaki sətirləri də **/usr/local/etc/mail/spamassassin/local.cf** faylına əlavə edirik:

```
user_razor2 1
razor_timeout 10
```

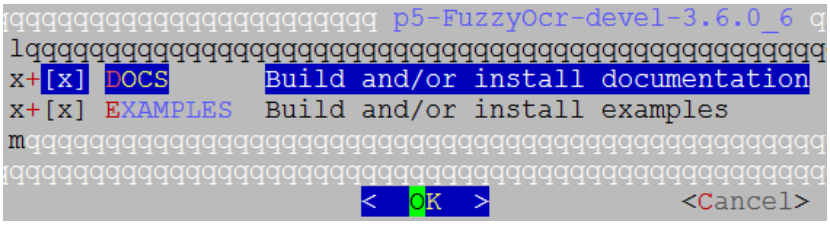
RAZOR hesablarını quraşdırırıq
su - vscan
razor-admin -discover
razor-admin -create
razor-admin -register -l -user=postmaster@saas.az -pass=freebsd
Register successful. Identity stored in /var/amavisd/.razor/identity-postmaster@saas.az
exit

Qeyd: Yuxarıda təyin etdiyiniz istifadəçinin email yeşiyi tez-tez yoxlanılmalıdır çünki, razor2 sapmalar haqqında təyinat və hesabatları bu ünvana yollayacaq.

/var/amavisd/.razor/razor-agent.log jurnal faylında gördüyümüz işlərin nəticəsini yoxlayırıq:
May 03 09:48:33.572996 admin[62561]: [2] [bootup] Logging initiated LogDebugLevel=3 to file:/var/amavisd/.razor/razor-agent.log
May 03 09:48:33.573571 admin[62561]: [2] Razor-Agents v2.84 starting razor-admin -register -l -user=postmaster@saas.az -pass=freebsd
May 03 09:48:34.002369 admin[62561]: [3] Attempting to register.
May 03 09:48:34.437572 admin[62561]: [3] Register successful. Identity stored in /var/amavisd/.razor/identity-postmaster@saas.az

FuzzyOCR-in yüklənməsi
FuzzyOCR – alətdir hansı ki, şəkillərdə Spam-ı təyin edə bilir. Çox əla işləyən alətdir. Haqqında ətraflı oxumaq istəsəniz http://fuzzyocr.own-hero.net/ linkinə müraciət edə bilərsiniz.

cd /usr/ports/mail/p5-FuzzyOcr-devel – Port ünvanına daxil oluruq
make config – Lazımi modulları seçirik



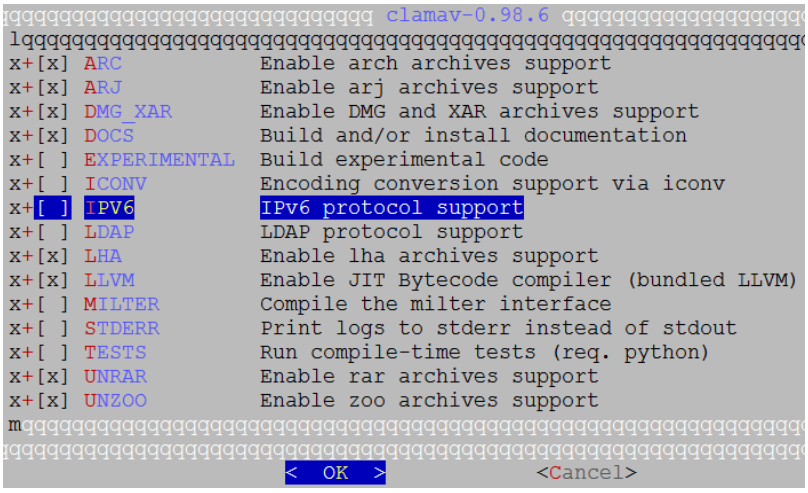
make install – Yükləyirik

Sonra FuzzyOCR nüsxə fayllarını spasmassassin qovluğuna nüsxələyirik:
cp /usr/local/share/examples/FuzzyOcr/FuzzyOcr.* /usr/local/etc/mail/spamassassin

Clam AntiVirus-un yüklənməsi
Clam AntiVirus – havayı antivirus proqram təminatıdır hansı ki, əla işləyir. Ancaq siz MaiaMailguard işləməsi üçün digər antiviruslardan da istifadə edə

bilərsiniz. Clamd haqqında ətraflı oxumaq istəsəniz, http://www.clamav.net/index.html linkinə müraciət edə bilərsiniz.

cd /usr/ports/security/clamav - Port ünvanına daxil oluruq
make config - Lazımi modulları seçirik



make all install clean CLAMAVUSER=vscan CLAMAVGROUP=vscan - **vscan** istifadəçi adı və qrupundan kompilyasiya edirik

/etc/make.conf faylına yüklənmə parametrlərini əlavə edirik. Bu gələcəkdə proqramın portlardan yenilənməsi zamanı çıxacaq problemin qarşısını alacaq:
echo 'CLAMAVUSER=vscan' >> /etc/make.conf
echo 'CLAMAVGROUP=vscan' >> /etc/make.conf

ClamAV-i sistemin StartUP-na əlavə edirik.
echo 'clamav_freshclam_enable="YES"' >> /etc/rc.conf
echo 'clamav_clamd_enable="YES"' >> /etc/rc.conf

Lazımi jurnal faylları yaradırıq:
touch /var/log/clamav/freshclam.log
touch /var/log/clamav/clamd.log
touch /var/log/clamav/razor-agent.log
chown -R vscan:vscan /var/run/clamav/ - Prosesin işə salınması üçün qovluğun yetkisini vscan edirik

chown -R vscan:vscan /var/log/clamav/
chown -R vscan:vscan /var/db/clamav/ - ClamAV yenilənmə bazalarını da vscan istifadəçi və qrupun üzvü edirik

freshclam - Əmri işə salırıq ki, **/var/db/clamav/** ünvanina ən yeni ***.cvd** yada ***.cld** bazalarını endirib gündəmdə saxlasın(Nəticə aşağıdaki kimi olmalıdır).

```
ClamAV update process started at Sun May  3 10:52:03 2015
WARNING: Your ClamAV installation is OUTDATED!
WARNING: Local version: 0.98.6 Recommended version: 0.98.7
```

```
DON'T PANIC! Read http://www.clamav.net/support/faq
Downloading main.cvd [100%]
main.cvd updated (version: 55, sigs: 2424225, f-level: 60, builder: neo)
Downloading daily.cvd [100%]
daily.cvd updated (version: 20409, sigs: 1381309, f-level: 63, builder: neo)
Downloading bytecode.cvd [100%]
bytecode.cvd updated (version: 254, sigs: 45, f-level: 63, builder: anvilleg)
Database updated (3805579 signatures) from database.clamav.net (IP:
195.228.75.149)
```

FreshClam və ClamAV proqramlarını işə salaq.

/usr/local/etc/rc.d/clamav-clamd start - ClamD-ni işə salırıq

sockstat -l|grep vscan - İşə düşməsini yoxlayırıq

```
vscan    clamd      24282 4  stream /var/run/clamav/clamd.sock
```

/usr/local/etc/rc.d/clamav-freshclam start - FreshClam-i işə salırıq

ps waux | grep freshclam | grep -v grep - İşə düşməsini yoxlayırıq

```
vscan   24312   0.0  0.4  60020  15264  -  Is   11:02AM     0:04.54
/usr/local/bin/freshclam --daemon -p /var/run/clamav/freshclam.pid
```

PEAR-in yüklənməsi

PEAR – PHP-də genişlənmələrin saxlanılması üçün əlavə kimi tərcümə edilir. Əgər siz WEB proqram təminatları ilə çox işləyirsinizsə, PEAR istifadəsi işinizi çox asanlaşdıracaq. Haqqında ətraflı oxumaq üçün http://pear.php.net/ linkinə müraciət edə bilərsiniz.

cd /usr/ports/devel/pear - Port ünvanına daxil oluruq

make install - Yükləyirik

/usr/local/etc/php.ini faylında aşağıdaki sətiri uyğun olaraq dəyişirik:

```
; UNIX: "/path1:/path2"
include_path = ".:/usr/local/share/pear"

; Windows: "\path1;\path2"
;include_path = ".;c:\php\includes"
```

chown -R www:www /usr/local/share/pear/ - kodları www istifadəçi adı və qrupuna mənimsədirik

HTMLPurfier-i yükləyirik:

pear channel-discover htmlpurifier.org - Yeni yüklənmə kanalı əlavə edirik

```
Adding Channel "htmlpurifier.org" succeeded
Discovery of channel "htmlpurifier.org" succeeded
```

pear install hp/HTMLPurifier - HTMLPrufier-i yükləyirik

```
downloading HTMLPurifier-4.6.0.tgz ...
Starting to download HTMLPurifier-4.6.0.tgz (239,621 bytes)
.....................................................done: 239,621 bytes
install ok: channel://htmlpurifier.org/HTMLPurifier-4.6.0
```

Maia-Mailguard yüklənməsi

MaiaMailguard – Spam və antivirus filterləri üçün əla havayı alətdir. O imkan verir ki, spam və antivirus filter üçün şəxsi quraşdırmaları təyin edəsiniz. Inzibatçının işini çox rahatlaşdırır. Haqqında http://www.maiamailguard.com/ linkindən oxuya bilərsiniz.

cd /usr/ports/security/maia/ - Port unvanına daxil oluruq

make config - Lazımi modulları seçirik

make install - Yükləyirik

```
mysql -uroot -p
mysql> CREATE DATABASE maia;
mysql> GRANT ALL PRIVILEGES ON maia.* TO vscan@localhost IDENTIFIED BY
'maiashifresi';
mysql> FLUSH PRIVILEGES;
```

cd /usr/local/share/doc/maia - Maia qovluğuna daxil oluruq

mysql -u root -p maia < maia-mysql.sql - Maia bazasının MySQL sxemini yaradırıq

/usr/local/etc/maia/maia.conf quraşdırma faylını aşağıdaki qaydada dəyişdiririk(verilənlər bazasını, maia scriptlərini, spamassassin **local.cf** scriptini və qaydalarını,PHP maia scriptlər üçün URL-i təyin edirik)

```
$dsn = 'DBI:mysql:maia:localhost:3306';
$username = 'vscan';
$password = 'maiashifresi';
$script_dir = '/usr/local/share/maia/scripts';
$sa_learn = '/usr/local/bin/sa-learn';
$address_rewriting_type = 0;
$routing_domain = '';
$auth_method = 'sql';
$preserve_case = 0;
$local_cf_dir = '/usr/local/etc/mail/spamassassin';
$system_rules_dir = '/var/db/spamassassin';
$user_rules_dir = '/var/maiad/.spamassassin';
$pid_dir = '/var/run/maia/';
$log_level= 8;
$pq_log_level = 'info';
$log_dir = '/var/log/maia';
$workers = 10;
$key_file = undef;
$default_max_size = 256*1024;
$learning_options = 1;
$autolearn_ham_threshold = undef;
$autolearn_spam_threshold = undef;
$autoreport_spam_threshold = undef;
$report_options = 1 + 2 + 4 + 8;
$mail_types = 1 + 2 + 4 + 8 + 16;
$base_url = 'http://mail.saas.az';
$template_dir = '/usr/local/etc/maia/templates/';
%sort = (
            'ham'    => "score DESC",
            'spam'   => "score ASC",
            'virus' => "received_date DESC",
            'attachment' => "received_date DESC",
            'header'  => "received_date DESC",
);
$titles = { 'spam'         =>  "Spam Quarantine",
            'virus'        =>  "Virus Quarantine",
            'attachment' =>  "Banned File Attachments",
            'header'   =>  "Invalid Email Headers",
            'ham'          =>  "Delivered Email"
};
@report_order = ('spam','ham','virus','attachment','header');
```

/usr/local/etc/maia/maiad.conf quraşdırma faylını da eynilə lazımi qaydada düzəldirik(Faylda olan domain adı, MySQL quraşdırmaları, ozünüzə uyğun olaraq dəyişməyi unutmayın):

```
use strict;
$max_servers = 2;
$daemon_user  = 'vscan';
$daemon_group = 'vscan';
$MYHOME = '/var/maiad';
$daemon_chroot_dir = undef;
$X_HEADER_TAG = 'X-Virus-Scanned';
$X_HEADER_LINE = "Maia Mailguard 1.0.4";
$mydomain = 'saas.az';
$myhostname = 'mail.saas.az';
$inet_socket_bind = '127.0.0.1';
$inet_socket_port = 10024;
@inet_acl = qw( 127.0.0.1 );
$forward_method = 'smtp:[127.0.0.1]:10025';
$log_level = 5;
$DO_SYSLOG = 1;
$SYSLOG_LEVEL = 'mail.debug';
$LOGFILE = "/var/log/maia/maiad.log";
@lookup_sql_dsn = ( ['DBI:mysql:maia:localhost:3306', 'vscan',
'maiashifresi'] );
$enable_db = 1;
$enable_global_cache = 1;
$path = '/usr/local/sbin:/usr/local/bin:/usr/sbin:/sbin:/usr/bin:/bin';
$file        = '/usr/bin/file';
$gzip        = 'gzip';
$bzip2       = 'bzip2';
$lzop        = 'lzop';
$rpm2cpio    = ['rpm2cpio.pl','rpm2cpio'];
$cabextract  = 'cabextract';
$uncompress  = ['uncompress', 'gzip -d', 'zcat'];
$unfreeze    = ['unfreeze', 'freeze -d', 'melt', 'fcat'];
$arc         = ['nomarch', 'arc'];
$unarj       = ['arj', 'unarj'];
#$unrar      = ['rar', 'unrar'];
$zoo         = 'zoo';
$lha         = 'lha';
$cpio        = ['gcpio','cpio'];
$ar          = 'ar';
$dspam       = 'dspam';
$pax         = 'pax';
$ripole      = 'ripole';

$MAXLEVELS = 14;
$MAXFILES = 1500;
$MIN_EXPANSION_QUOTA = 100*1024;
$MAX_EXPANSION_QUOTA = 300*1024*1024;
$defang_virus  = 1;
$defang_banned = 1;
$sa_spam_subject_tag = '***SPAM*** ';
$sa_mail_body_size_limit = 512*1024;
```

```
$sa_local_tests_only = 0;
$sa_timeout = 60;
$banned_filename_re = new_RE(
  qr'\.[^./]*\.(exe|vbs|pif|scr|bat|cmd|com|cpl|dll)\.?$'i,
  qr'^application/x-msdownload$'i,
  qr'^application/x-msdos-program$'i,
  qr'^application/hta$'i,
  qr'^message/partial$'i, qr'^message/external-body$'i,
  qr'.\.(ade|adp|app|bas|bat|chm|cmd|com|cpl|crt|exe|fxp|hlp|hta|inf|ins|isp|
        js|jse|lnk|mda|mdb|mde|mdw|mdt|mdz|msc|msi|msp|mst|ops|pcd|pif|prg|
        reg|scr|sct|shb|shs|vb|vbe|vbs|wsc|wsf|wsh)$'ix,
  qr'^\.(exe-ms)$',
  qr'^\.(exe|lha|cab|dll)$',
);
@av_scanners = (
['ClamAV-clamd',
   \&ask_daemon, ["CONTSCAN {}\n", "/var/run/clamav/clamd.ctl"],
   qr/\bOK$/m, qr/\bFOUND$/m,
   qr/^.*?: (?!Infected Archive)(.*) FOUND$/m ],
);
@av_scanners_backup = (
['ClamAV-clamscan', 'clamscan',
  "--stdout --no-summary -r --tempdir=$TEMPBASE {}",
  [0], qr/:.*\sFOUND$/m, qr/^.*?: (?!Infected Archive)(.*) FOUND$/m ],
);

@viruses_that_fake_sender_maps = (new_RE(
  [qr'\bEICAR\b'i => 0],
  [qr/.*/ => 1],
));

@keep_decoded_original_maps = (new_RE(
  qr'^MAIL-UNDECIPHERABLE$',
  qr'^(ASCII(?! cpio)|text|uuencoded|xxencoded|binhex)'i,
));

@non_malware_viruses_maps = (new_RE(
   qr'^(Email|E-Mail)\.(Ecard|Faketube|FreeGame|PornTeaser)',
   qr'^(Email|E-Mail)\.(Hoax|Phishing)\.',
   qr'^(HTML|Heuristics)\.Phishing\.',
   qr'^Sanesecurity\.Junk\.',
   qr'^Sanesecurity\.Jurlbl\.',
   qr'^Sanesecurity\.Jurlbl\.Auto\.',
   qr'^Sanesecurity\.Lott\.',
   qr'^Sanesecurity\.(Auction|Casino|Doc|Phishing)\.',

qr'^Sanesecurity\.(PhishingTestSig|TestSig_Type3_Bdy|TestSig_Type4_Bdy|TestSi
g_Type4_Hdr)',

qr'^Sanesecurity\.(Casino|Cred|Dipl|Hdr|Img|Img0|Job|Loan|Porn|Scam|Scam4|Sca
mL|Spam|Spam4|SpamL|Stk)\.',
   qr'^Sanesecurity\.TestSig',
   qr'^Sanesecurity\.Spam\.',
```

```
    qr'^Sanesecurity\.SpamAttach\.',
    qr'^Sanesecurity\.SpamImg\.',
    qr'^Sanesecurity\.Spear\.',
    qr'^Sanesecurity\.SpearL\.',
    qr'^MSRBL-Images\.[0-5,S]-',
    qr'^MSRBL-Images.Test-',
    qr'^MSRBL-SPAM\.',
    qr'^Email\.Spam\d+-SecuriteInfo\.com',
    qr'^Doppelstern\.Attachment\.',
    qr'^winnow\.(phish|scam)\.',
));
1;
```

Maia üçün scriptləri lazımi ünvana nüsxələyirik:

```
mkdir -p /var/amavisd/maia/
cp -R /usr/local/share/maia/* /var/amavisd/maia/
chown -R vscan:vscan /var/amavisd/maia/
```

/var/amavisd/maia/scripts/configtest.pl scriptini işə salırıq və aşağıdaki nəticəni əldə edirik(Çatışmayan paketləri əlimizlə yükləyirik):

```
Application/Module       Version    Status
=========================================================================
Perl                   :   5.18.4 : OK
file(1)                :     5.19 : OK
Archive::Tar           :     1.90 : OK
Archive::Zip           :     1.46 : OK
BerkeleyDB             :     0.55 : OK
Compress::Zlib         :     2.06 : OK
Convert::TNEF          :     0.18 : OK
Convert::UUlib         :      1.4 : OK
Crypt::OpenSSL::RSA    :     0.28 : OK
Data::UUID             :    1.220 : OK
DB_File                :    1.827 : OK
DBD::mysql             :    4.031 : OK
DBD::Pg                :      N/A : NOT INSTALLED (required if you use
PostgreSQL as your Maia Mailguard database)
DBI                    :    1.633 : OK
Digest::MD5            :     2.52 : OK
Digest::SHA            :   5.8402 : OK
Digest::SHA1           :     2.13 : OK
Encode::Detect         :     1.01 : OK
File::Spec             :     3.40 : OK
forks                  :     0.36 : OK
HTML::Parser           :     3.71 : OK
HTTP::Date             :     6.02 : OK
IO::Stringy            :    2.111 : OK
IO::Socket::INET6      :     2.72 : OK
IO::Zlib               :     1.10 : OK
IP::Country::Fast      :  604.001 : OK
libdb                  :      5.3 : OK
```

```
LWP                     :      N/A : NOT INSTALLED (SpamAssassin's sa-update
script requires this)
Mail::Address           :     2.13 : OK
Mail::DKIM              :      0.4 : OK
Mail::Internet          :     2.13 : OK
Mail::SpamAssassin      :    3.4.1 : OK
Mail::SPF               :    2.009 : OK
MIME::Base64            :     3.13 : OK
MIME::Parser            :    5.506 : OK
MIME::QuotedPrint       :     3.13 : OK
Net::CIDR::Lite         :     0.21 : OK
Net::Cmd                :     2.29 : OK
Net::DNS                :     0.83 : OK
Net::Server             :    2.008 : OK
Net::SMTP               :     2.31 : OK
NetAddr::IP             :    4.075 : OK
Pod::Usage              :     1.61 : OK
Razor2::Client::Agent   :     2.84 : OK
Template                :     2.26 : OK
Time::HiRes             :   1.9726 : OK
Unix::Syslog            :      1.1 : OK
URI                     :     1.67 : OK
Text::CSV               :     1.33 : OK
MySQL Server            :   5.5.43 : OK
Database DSN test       : PASSED
```

cd /usr/ports/www/p5-LWP-UserAgent-WithCache - Çatışmayan paketi yükləyirik

make install

Maia quraşdırma faylları üçün lazımi yetkiləri təyin edirik:

```
chown -R vscan:vscan /usr/local/etc/maia/
chmod -R 470 /usr/local/etc/maia/
```

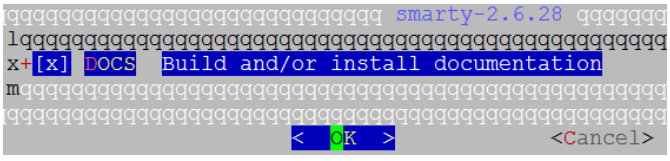
Spamassassin üçün bütün qaydaları yükləyirik:

```
# su - vscan
$ /var/amavisd/maia/scripts/load-sa-rules.pl --debug
```

Smarty motoru şablonlarını yükləyirik:

cd /usr/ports/www/smarty - Port ünvanına daxil oluruq

make config - Lazımi modulları seçirik



make install clean - Yükləyirik

/usr/local/etc/php.ini faylının içində uyğun olaraq aşağıdaki dəyişikliyi edirik:

```
;;;;;;;;;;;;;;;;;;;;;;;;
; Paths and Directories ;
;;;;;;;;;;;;;;;;;;;;;;;;

; UNIX: "/path1:/path2"
include_path = ".:/usr/local/share/pear:/usr/local/share/smarty"
```

chown -R www:www /usr/local/www/maia/ - Maia qovluğuna apache üçün yetki veririk

/usr/local/www/maia/config.php PHP quraşdırma faylında aşağıdaki dəyişiklikləri edirik(Faylda Maia və Postfix üçün yaratdığımız bazaların qoşulma quraşdırmalarını düzəldirik):

```
<?php
    $loglevel = PEAR_LOG_DEBUG;
    $debug_popup = false;
    $debuglevel = PEAR_LOG_DEBUG;
    $default_display_language = "en";
    date_default_timezone_set("Asia/Baku");
    $html_charset = "UTF-8";
    $default_session_timeout = 15;
    $maia_sql_dsn = "mysql://vscan:maiashifresi@tcp(localhost:3306)/maia";
    $purifier_cache = null;
        $protection = array( 'off'     => array
('Y','Y','Y','Y','Y','Y','Y','Y','N','N','N','N','N','999','999','999'),
                             'low'     => array
('N','Y','Y','Y','N','Y','Y','Y','N','N','N','N','N','999','999','999'),
                             'medium'  => array
('N','N','Y','Y','N','N','Y','Y','N','N','N','N','Y','5','999','999'),
                             'high'    => array
('N','N','N','N','N','N','N','N','N','N','N','N','N','1','5','5')
        );
    $chart_font = '';
    $address_rewriting_type = 4;
    $routing_domain = "";
    $auth_method = "sql";
    $auth_pop3_host = "localhost";
    $auth_pop3_port = 110;
    $auth_imap_host = "localhost";
    $auth_imap_port = 143;
    $auth_ldap_server = "hostname";
    $auth_ldap_password = "password";
    $auth_ldap_use_tls = "false";
    $auth_ldap_version = 2;
    $auth_ldap_query =
"(|(mailLocalAddress=%%USER%%)(mailLocalAddress=%%USER%%@domain.tld))";
    $auth_ldap_bind_dn = "cn=company, dc=domain, dc=tld";
    $auth_ldap_base_dn = "dc=domain, dc=tld";
    $auth_ldap_attribute = "mailroutingaddress";
    $auth_ldap_opt_referrals = 1;
    $auth_exchange_nt_domain = "NTDomain";
    $auth_exchange_only_one_domain = False;
```

```
    $auth_exchange_params =
"{hostname:port/imap/norsh/notls/%%NTDOMAIN%%/%%USER%%}INBOX";
    $auth_sql_dsn =
"mysql://postfix:postfixdbpass@tcp(localhost:3306)/postfix";
    $auth_sql_table = "mailbox";
    $auth_sql_username_column = "username";
    $auth_sql_password_column = "password";
    $auth_sql_email_column = "username";
    $auth_sql_password_type = "crypt";
    $auth_external = "/bin/true";
?>
```

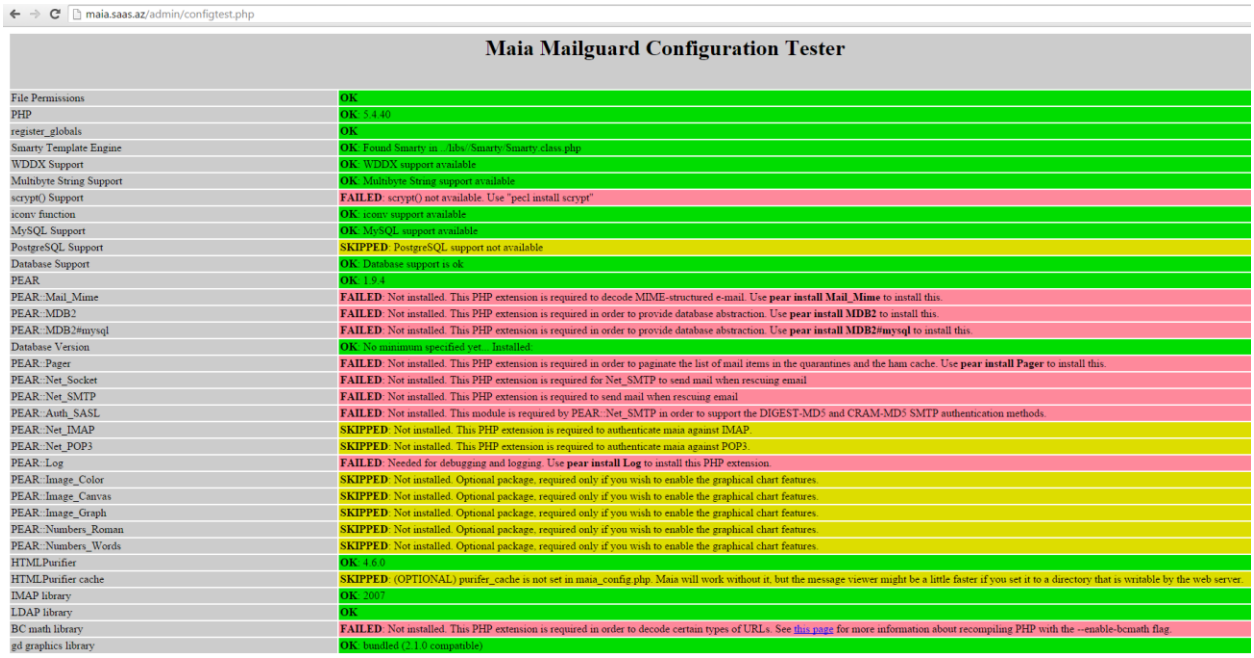
/usr/local/domen/maia.saas.az virtual maia hostu yaradırıq və məzmununa aşağıdaki sətirləri əlavə edirik:

```
<VirtualHost *:80>
        ServerAdmin jamal.shahverdiyev@saas.az
        ServerName maia.saas.az
        AcceptPathInfo On
        DocumentRoot /usr/local/www/maia/
<Directory "/usr/local/www/maia">
        AllowOverride All
        Require all granted
</Directory>
        ErrorLog /var/log/httpd/maia-error.log
        CustomLog /var/log/httpd/maia-access.log combined
</VirtualHost>
```

touch /var/log/httpd/maia-error.log /var/log/httpd/maia-access.log - Maia üçün web jurnal faylları yaradırıq

/usr/local/etc/rc.d/apache24 restart - Apache24-ü yenidən işə salırıq

http://maia.saas.az/admin/configtest.php linkinə müraciət edərək maia üçün tələb edilən bütün paketlərin siyahısını çap edirik(aşağıdaki şəkildəki kimi):



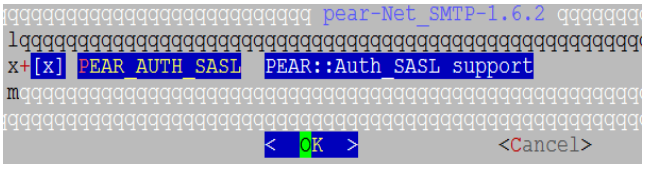
maia.saas.az/admin/configtest.php

Maia Mailguard Configuration Tester

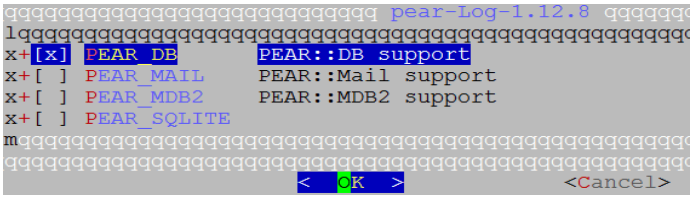
File Permissions	**OK**
PHP	**OK**: 5.4.40
register_globals	**OK**
Smarty Template Engine	**OK**: Found Smarty in ../libs/Smarty/Smarty.class.php
WDDX Support	**OK**: WDDX support available
Multibyte String Support	**OK**: Multibyte String support available
scrypt() Support	**FAILED**: scrypt() not available. Use "pecl install scrypt"
iconv function	**OK**: iconv support available
MySQL Support	**OK**: MySQL support available
PostgreSQL Support	**SKIPPED**: PostgreSQL support not available
Database Support	**OK**: Database support is ok
PEAR	**OK**: 1.9.4
PEAR::Mail_Mime	**FAILED**: Not installed. This PHP extension is required to decode MIME-structured e-mail. Use **pear install Mail_Mime** to install this.
PEAR::MDB2	**FAILED**: Not installed. This PHP extension is required in order to provide database abstraction. Use **pear install MDB2** to install this.
PEAR::MDB2#mysql	**FAILED**: Not installed. This PHP extension is required in order to provide database abstraction. Use **pear install MDB2#mysql** to install this.
Database Version	**OK**: No minimum specified yet... Installed.
PEAR::Pager	**FAILED**: Not installed. This PHP extension is required in order to paginate the list of mail items in the quarantines and the ham cache. Use **pear install Pager** to install this.
PEAR::Net_Socket	**FAILED**: Not installed. This PHP extension is required for Net_SMTP to send mail when rescuing email
PEAR::Net_SMTP	**FAILED**: Not installed. This PHP extension is required to send mail when rescuing email
PEAR::Auth_SASL	**FAILED**: Not installed. This module is required by PEAR::Net_SMTP in order to support the DIGEST-MD5 and CRAM-MD5 SMTP authentication methods.
PEAR::Net_IMAP	**SKIPPED**: Not installed. This PHP extension is required to authenticate maia against IMAP.
PEAR::Net_POP3	**SKIPPED**: Not installed. This PHP extension is required to authenticate maia against POP3.
PEAR::Log	**FAILED**: Needed for debugging and logging. Use **pear install Log** to install this PHP extension.
PEAR::Image_Color	**SKIPPED**: Not installed. Optional package, required only if you wish to enable the graphical chart features.
PEAR::Image_Canvas	**SKIPPED**: Not installed. Optional package, required only if you wish to enable the graphical chart features.
PEAR::Image_Graph	**SKIPPED**: Not installed. Optional package, required only if you wish to enable the graphical chart features.
PEAR::Numbers_Roman	**SKIPPED**: Not installed. Optional package, required only if you wish to enable the graphical chart features.
PEAR::Numbers_Words	**SKIPPED**: Not installed. Optional package, required only if you wish to enable the graphical chart features.
HTMLPurifier	**OK**: 4.6.0
HTMLPurifier cache	**SKIPPED**: (OPTIONAL) purifier_cache is not set in maia_config.php. Maia will work without it, but the message viewer might be a little faster if you set it to a directory that is writable by the web server.
IMAP library	**OK**: 2007
LDAP library	**OK**
BC math library	**FAILED**: Not installed. This PHP extension is required in order to decode certain types of URLs. See this page for more information about recompiling PHP with the --enable-bcmath flag.
gd graphics library	**OK**: bundled (2.1.0 compatible)

Başlayırıq bütün çatışmayan paketləri yükləməyə:

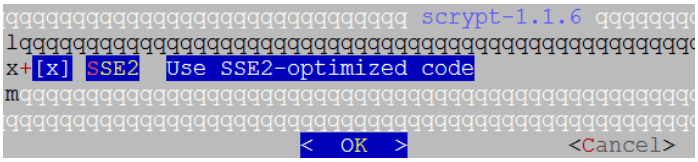
```
cd /usr/ports/mail/pear-Mail_Mime
make install

cd /usr/ports/mail/pear-Mail_mimeDecode
make install

cd /usr/ports/databases/pear-DB
make install

cd /usr/ports/devel/pear-Pager
make install

cd /usr/ports/net/pear-Net_Socket
make install


cd /usr/ports/net/pear-Net_SMTP
make config
```




```
make install


cd /usr/ports/sysutils/pear-Log
make config
```

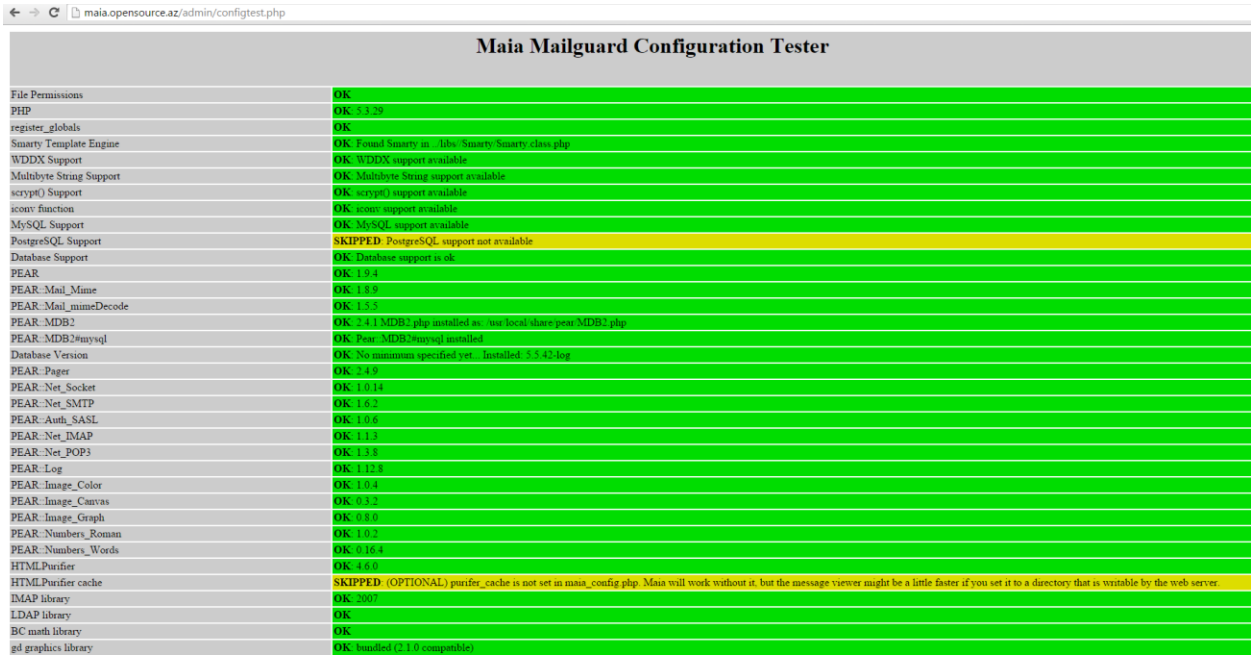



```
make install


cd /usr/ports/databases/pear-MDB2
make install

cd /usr/ports/databases/pear-MDB2_Driver_mysql
make install

cd /usr/ports/security/scrypt
make config
```




```
make install

pecl install scrypt      - Script-i PHP vasitəsilə yükləyirik.
echo "extension=scrypt.so" >> /usr/local/etc/php/extensions.ini    - PHP genişlənməl
                                                                    ərinə əlavə
                                                                    edirik


cd /usr/ports/math/php5-bcmath
make install

cd /usr/ports/mail/pear-Net_IMAP
make install

cd /usr/ports/net/pear-Net_POP3
make install

cd /usr/ports/graphics/pear-Image_Color
make install

cd /usr/ports/graphics/pear-Image_Canvas
make instll

cd /usr/ports/graphics/pear-Image_Graph
make install

cd /usr/ports/textproc/pear-Numbers_Roman
make install
cd /usr/ports/textproc/pear-Numbers_Words
make install
```

/usr/local/etc/rc.d/apache24 restart - Sonda Web serverimizi yenidən işə salırıq və aşağıdaki şəkildəki nəticəni test edib əldə edirik.



pear-Net_IMAP modulunu patch edirik:

```
cd /usr/local/share/pear/
fetch http://www.purplehat.org/downloads/postfix_guide/Pie.php.diff
Pie.php.diff                                  100% of  482  B 4245 kBps
00m00s
patch -p0 < Pie.php.diff
```

/usr/local/etc/amavisd.conf üçün quraşdırma faylının məzmunu aşağıdaki kimi olacaq:

```
use strict;
$max_servers = 2;
$daemon_user  = 'vscan';
$daemon_group = 'vscan';
$mydomain = 'saas.az';
$TEMPBASE = "$MYHOME/tmp";
$ENV{TMPDIR} = $TEMPBASE;
$QUARANTINEDIR = '/var/virusmails';
$log_level = 5;
$log_recip_templ = undef;
$do_syslog = 1;
$syslog_facility = 'mail';
$enable_db = 1;
$nanny_details_level = 2;
$enable_dkim_verification = 1;
$enable_dkim_signing = 1;
```

```
@local_domains_maps = ( [".$mydomain"] );
@mynetworks = qw( 127.0.0.0/8 [::1] [FE80::]/10 [FEC0::]/10
                  10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 );
$unix_socketname = "$MYHOME/amavisd.sock";
$inet_socket_port = 10025;
$policy_bank{'MYNETS'} = {
  originating => 1,
  os_fingerprint_method => undef,
};
$interface_policy{'10026'} = 'ORIGINATING';
$policy_bank{'ORIGINATING'} = {
  originating => 1,
  allow_disclaimers => 1,
  virus_admin_maps => ["virusalert\@$mydomain"],
  spam_admin_maps  => ["virusalert\@$mydomain"],
  warnbadhsender   => 1,
  forward_method => 'smtp:[127.0.0.1]:10027',
  smtpd_discard_ehlo_keywords => ['8BITMIME'],
  bypass_banned_checks_maps => [1],
  terminate_dsn_on_notify_success => 0,
};
$interface_policy{'SOCK'} = 'AM.PDP-SOCK';
$policy_bank{'AM.PDP-SOCK'} = {
  protocol => 'AM.PDP',
  auth_required_release => 0,
};
$sa_tag_level_deflt  = 2.0;
$sa_tag2_level_deflt = 6.2;
$sa_kill_level_deflt = 6.9;
$sa_dsn_cutoff_level = 10;
$sa_crediblefrom_dsn_cutoff_level = 18;
$penpals_bonus_score = 8;
$penpals_threshold_high = $sa_kill_level_deflt;
$bounce_killer_score = 100;
$sa_mail_body_size_limit = 256*1024;
$sa_local_tests_only = 0;
@lookup_sql_dsn = ( ['DBI:mysql:maia:localhost', 'vscan', 'maiashifresi'] );
$virus_admin = "virusalert\@$mydomain";  # notifications recip.
$mailfrom_notify_admin = "virusalert\@$mydomain";
$mailfrom_notify_recip = "virusalert\@$mydomain";
$mailfrom_notify_spamadmin = "spam.police\@$mydomain";
$mailfrom_to_quarantine = '';
@addr_extension_virus_maps = ('virus');
@addr_extension_banned_maps = ('banned');
@addr_extension_spam_maps = ('spam');
@addr_extension_bad_header_maps = ('badh');
$path = '/usr/local/sbin:/usr/local/bin:/usr/sbin:/sbin:/usr/bin:/bin';
$MAXLEVELS = 14;
$MAXFILES = 3000;
$MIN_EXPANSION_QUOTA = 100*1024;
$MAX_EXPANSION_QUOTA = 500*1024*1024;
$sa_spam_subject_tag = '***Spam*** ';
$defang_virus  = 1;
```

```
$defang_banned = 1;
$defang_by_ccat{CC_BADH.",3"} = 1;
$defang_by_ccat{CC_BADH.",5"} = 1;
$defang_by_ccat{CC_BADH.",6"} = 1;
$myhostname = 'mail.saas.az';
@keep_decoded_original_maps = (new_RE(
  qr'^MAIL$',
  qr'^MAIL-UNDECIPHERABLE$',
  qr'^(ASCII(?! cpio)|text|uuencoded|xxencoded|binhex)'i,
));
$banned_filename_re = new_RE(
  qr'^\.(exe-ms|dll)$',
  [ qr'^\.(rpm|cpio|tar)$'       => 0 ],
  qr'.\.(pif|scr)$'i,
  qr'^application/x-msdownload$'i,
  qr'^application/x-msdos-program$'i,
  qr'^application/hta$'i,
  qr'^(?!cid:).*\.[^./]*[A-Za-
z][^./]*\.\s*(exe|vbs|pif|scr|bat|cmd|com|cpl|dll)[.\s]*$'i,
  qr'.\.(exe|vbs|pif|scr|cpl)$'i,
);
@score_sender_maps = ({
  '.' => [
   new_RE(  # regexp-type lookup table, just happens to be all soft-blacklist
    [qr'^(bulkmail|offers|cheapbenefits|earnmoney|foryou)@'i         => 5.0],
    [qr'^(greatcasino|investments|lose_weight_today|market\.alert)@'i=> 5.0],
    [qr'^(money2you|MyGreenCard|new\.tld\.registry|opt-out|opt-in)@'i=> 5.0],
    [qr'^(optin|saveonlsmoking2002k|specialoffer|specialoffers)@'i   => 5.0],
    [qr'^(stockalert|stopsnoring|wantsome|workathome|yesitsfree)@'i  => 5.0],
    [qr'^(your_friend|greatoffers)@'i                                => 5.0],
    [qr'^(inkjetplanet|marketopt|MakeMoney)\d*@'i                    => 5.0],
   ),
   {
     'nobody@cert.org'                     => -3.0,
     'cert-advisory@us-cert.gov'           => -3.0,
     'owner-alert@iss.net'                 => -3.0,
     'slashdot@slashdot.org'               => -3.0,
     'securityfocus.com'                   => -3.0,
     'ntbugtraq@listserv.ntbugtraq.com'    => -3.0,
     'security-alerts@linuxsecurity.com'   => -3.0,
     'mailman-announce-admin@python.org'   => -3.0,
     'amavis-user-admin@lists.sourceforge.net'=> -3.0,
     'amavis-user-bounces@lists.sourceforge.net' => -3.0,
     'spamassassin.apache.org'             => -3.0,
     'notification-return@lists.sophos.com'   => -3.0,
     'owner-postfix-users@postfix.org'     => -3.0,
     'owner-postfix-announce@postfix.org'  => -3.0,
     'owner-sendmail-announce@lists.sendmail.org'   => -3.0,
     'sendmail-announce-request@lists.sendmail.org' => -3.0,
     'donotreply@sendmail.org'             => -3.0,
     'ca+envelope@sendmail.org'            => -3.0,
     'noreply@freshmeat.net'               => -3.0,
     'owner-technews@postel.acm.org'       => -3.0,
```

```
    'ietf-123-owner@loki.ietf.org'          => -3.0,
    'cvs-commits-list-admin@gnome.org'      => -3.0,
    'rt-users-admin@lists.fsck.com'         => -3.0,
    'clp-request@comp.nus.edu.sg'           => -3.0,
    'surveys-errors@lists.nua.ie'           => -3.0,
    'emailnews@genomeweb.com'               => -5.0,
    'yahoo-dev-null@yahoo-inc.com'          => -3.0,
    'returns.groups.yahoo.com'              => -3.0,
    'clusternews@linuxnetworx.com'          => -3.0,
    lc('lvs-users-admin@LinuxVirtualServer.org')    => -3.0,
    lc('owner-textbreakingnews@CNNIMAIL12.CNN.COM') => -5.0,
    'sender@example.net'                    =>  3.0,
    '.example.net'                          =>  1.0,
  },
 ],
});
@decoders = (
  ['mail', \&do_mime_decode],
  ['F',    \&do_uncompress, ['unfreeze', 'freeze -d', 'melt', 'fcat'] ],
  ['Z',    \&do_uncompress, ['uncompress', 'gzip -d', 'zcat'] ],
  ['gz',   \&do_uncompress, 'gzip -d'],
  ['gz',   \&do_gunzip],
  ['bz2',  \&do_uncompress, 'bzip2 -d'],
  ['xz',   \&do_uncompress,
           ['xzdec', 'xz -dc', 'unxz -c', 'xzcat'] ],
  ['lzma', \&do_uncompress,
           ['lzmadec', 'xz -dc --format=lzma',
            'lzma -dc', 'unlzma -c', 'lzcat', 'lzmadec'] ],
  ['lrz',  \&do_uncompress,
           ['lrzip -q -k -d -o -', 'lrzcat -q -k'] ],
  ['lzo',  \&do_uncompress, 'lzop -d'],
  ['lz4',  \&do_uncompress, ['lz4c -d'] ],
  ['rpm',  \&do_uncompress, ['rpm2cpio.pl', 'rpm2cpio'] ],
  [['cpio','tar'], \&do_pax_cpio, ['pax', 'gcpio', 'cpio'] ],
  ['deb',  \&do_ar, 'ar'],
  ['rar',  \&do_unrar, ['unrar', 'rar'] ],
  ['arj',  \&do_unarj, ['unarj', 'arj'] ],
  ['arc',  \&do_arc,   ['nomarch', 'arc'] ],
  ['zoo',  \&do_zoo,   ['zoo', 'unzoo'] ],
  ['doc',  \&do_ole,   'ripole'],
  ['cab',  \&do_cabextract, 'cabextract'],
  ['tnef', \&do_tnef_ext, 'tnef'],
  ['tnef', \&do_tnef],
  [['zip','kmz'], \&do_7zip,  ['7za', '7z'] ],
  [['zip','kmz'], \&do_unzip],
  ['7z',   \&do_7zip,  ['7zr', '7za', '7z'] ],
  [[qw(gz bz2 Z tar)],
           \&do_7zip,  ['7za', '7z'] ],
  [[qw(xz lzma jar cpio arj rar swf lha iso cab deb rpm)],
           \&do_7zip,  '7z' ],
  ['exe',  \&do_executable, ['unrar','rar'], 'lha', ['unarj','arj'] ],
);
@av_scanners = (
```

```
 ['ClamAV-clamd',
   \&ask_daemon, ["CONTSCAN {}\n", "/var/run/clamav/clamd.sock.sock"],
   qr/\bOK$/m, qr/\bFOUND$/m,
   qr/^.*?: (?!Infected Archive)(.*) FOUND$/m ],
  ['KasperskyLab AVP - aveclient',
    ['/usr/local/kav/bin/aveclient','/usr/local/share/kav/bin/aveclient',
     '/opt/kav/5.5/kav4mailservers/bin/aveclient','aveclient'],
    '-p /var/run/aveserver -s {}/*',
    [0,3,6,8], qr/\b(INFECTED|SUSPICION|SUSPICIOUS)\b/m,
    qr/(?:INFECTED|WARNING|SUSPICION|SUSPICIOUS) (.+)/m,
  ],
  ['KasperskyLab AntiViral Toolkit Pro (AVP)', ['avp'],
    '-* -P -B -Y -O- {}', [0,3,6,8], [2,4],
    qr/infected: (.+)/m,
    sub {chdir('/opt/AVP') or die "Can't chdir to AVP: $!"},
    sub {chdir($TEMPBASE) or die "Can't chdir back to $TEMPBASE $!"},
  ],
  ['KasperskyLab AVPDaemonClient',
    [ '/opt/AVP/kavdaemon',       'kavdaemon',
      '/opt/AVP/AvpDaemonClient', 'AvpDaemonClient',
      '/opt/AVP/AvpTeamDream',    'AvpTeamDream',
      '/opt/AVP/avpdc', 'avpdc' ],
    "-f=$TEMPBASE {}", [0,8], [3,4,5,6], qr/infected: ([^\r\n]+)/m ],
  ['CentralCommand Vexira (new) vascan',
    ['vascan','/usr/lib/Vexira/vascan'],
    "-a s --timeout=60 --temp=$TEMPBASE -y $QUARANTINEDIR ".
    "--log=/var/log/vascan.log {}",
    [0,3], [1,2,5],
    qr/(?x)^\s* (?:virus|iworm|macro|mutant|sequence|trojan)\ found:\ (
[^\]\s']+ )\ \.\.\.\.\ /m ],
  ['Avira AntiVir', ['antivir','vexira'],
    '--allfiles -noboot -nombr -rs -s -z {}', [0], qr/ALERT:|VIRUS:/m,
    qr/(?x)^\s* (?: ALERT: \s* (?: \[ | [^']* ' ) |
         (?i) VIRUS:\ .*?\ virus\ '?) ( [^\]\s']+ )/m ],
  ['Avira AntiVir', ['avscan'],
   '-s --batch --alert-action=none {}', [0,4], qr/(?:ALERT|FUND):/m,
   qr/(?:ALERT|FUND): (?:.* <<< )?(.+?)(?: ; |$)/m ],
  ['Command AntiVirus for Linux', 'csav',
    '-all -archive -packed {}', [50], [51,52,53],
    qr/Infection: (.+)/m ],
  ['Symantec CarrierScan via Symantec CommandLineScanner',
    'cscmdline', '-a scan -i 1 -v -s 127.0.0.1:7777 {}',
    qr/^Files Infected:\s+0$/m, qr/^Infected\b/m,
    qr/^(?:Info|Virus Name):\s+(.+)/m ],
  ['Symantec AntiVirus Scan Engine',
    'savsecls', '-server 127.0.0.1:7777 -mode scanrepair -details -verbose
{}',
    [0], qr/^Infected\b/m,
    qr/^(?:Info|Virus Name):\s+(.+)/m ],
   ['F-Secure Linux Security',
    ['/opt/f-secure/fsav/bin/fsav', 'fsav'],
    '--virus-action1=report --archive=yes --auto=yes '.
    '--list=no --nomimeerr {}', [0], [3,4,6,8],
```

```
  qr/(?:infection|Infected|Suspected|Riskware): (.+)/m ],
['CAI InoculateIT', 'inocucmd',  # retired product
  '-sec -nex {}', [0], [100],
  qr/was infected by virus (.+)/m ],
['CAI eTrust Antivirus', 'etrust-wrapper',
  '-arc -nex -spm h {}', [0], [101],
  qr/is infected by virus: (.+)/m ],
['MkS_Vir for Linux (beta)', ['mks32','mks'],
  '-s {}/*', [0], [1,2],
  qr/--[ \t]*(.+)/m ],
['MkS_Vir daemon', 'mksscan',
  '-s -q {}', [0], [1..7],
  qr/^... (\S+)/m ],
['ESET Software ESETS Command Line Interface',
  ['/usr/bin/esets_cli', 'esets_cli'],
  '--subdir {}', [0], [1,2,3],
  qr/:\s*action="(?!accepted)[^"]*"\n.*:\s*virus="([^"]*)"/m ],
['ESET NOD32 for Linux File servers',
  ['/opt/eset/nod32/sbin/nod32','nod32'],
  '--files -z --mail --sfx --rtp --adware --unsafe --pattern --heur '.
  '-w -a --action=1 -b {}',
  [0], [1,10], qr/^object=.*, virus="(.*?)",/m ],
['Norman Virus Control v5 / Linux', 'nvcc',
  '-c -l:0 -s -u -temp:$TEMPBASE {}', [0,10,11], [1,2,14],
  qr/(?i).* virus in .* -> \'(.+)\'/m ],
['Panda CommandLineSecure 9 for Linux',
  ['/opt/pavcl/usr/bin/pavcl','pavcl'],
  '-auto -aex -heu -cmp -nbr -nor -nos -eng -nob {}',
  qr/Number of files infected[ .]*: 0+(?!\d)/m,
  qr/Number of files infected[ .]*: 0*[1-9]/m,
  qr/Found virus :\s*(\S+)/m ],
['NAI McAfee AntiVirus (uvscan)', 'uvscan',
  '--secure -rv --mime --summary --noboot - {}', [0], [13],
  qr/(?x) Found (?:
      \ the\ (.+)\ (?:virus|trojan)  |
      \ (?:virus|trojan)\ or\ variant\ ([^ ]+)  |
      :\ (.+)\ NOT\ a\ virus)/m,
],
['VirusBuster', ['vbuster', 'vbengcl'],
  "{} -ss -i '*' -log=$MYHOME/vbuster.log", [0], [1],
  qr/: '(.*)' - Virus/m ],
['CyberSoft VFind', 'vfind',
  '--vexit {}/*', [0], [23], qr/##==>>>> VIRUS ID: CVDL (.+)/m,
],
['avast! Antivirus', ['/usr/bin/avastcmd','avastcmd'],
  '-a -i -n -t=A {}', [0], [1], qr/\binfected by:\s+([^ \t\n\[\]]+)/m ],
['Ikarus AntiVirus for Linux', 'ikarus',
  '{}', [0], [40], qr/Signature (.+) found/m ],
['BitDefender', 'bdscan',
  '--action=ignore --no-list {}', qr/^Infected files\s*:\s*0+(?!\d)/m,
  qr/^(?:Infected files|Identified viruses|Suspect files)\s*:\s*0*[1-9]/m,
  qr/(?:suspected|infected)\s*:\s*(.*)(?:\033|$)/m ],
['BitDefender', 'bdc',
```

```
    '--arc --mail {}', qr/^Infected files *:0+(?!\d)/m,
    qr/^(?:Infected files|Identified viruses|Suspect files) *:0*[1-9]/m,
    qr/(?:suspected|infected): (.*)(?:\033|$)/m ],
  ['ArcaVir for Linux', ['arcacmd','arcacmd.static'],
    '-v 1 -summary 0 -s {}', [0], [1,2],
    qr/(?:VIR|WIR):[ \t]*(.+)/m ],
);
@av_scanners_backup = (
  ['ClamAV-clamscan', 'clamscan',
    "--stdout --no-summary -r --tempdir=$TEMPBASE {}",
    [0], qr/:.*\sFOUND$/m, qr/^.*?: (?!Infected Archive)(.*) FOUND$/m ],
  ['F-PROT Antivirus for UNIX', ['fpscan'],
    '--report --mount --adware {}',
    [0,8,64],  [1,2,3, 4+1,4+2,4+3, 8+1,8+2,8+3, 12+1,12+2,12+3],
    qr/^\[Found\s+[^\]]*\]\s+<([^ \t(>]*)/m ],
  ['FRISK F-Prot Antivirus', ['f-prot','f-prot.sh'],
    '-dumb -archive -packed {}', [0,8], [3,6],
    qr/(?:Infection:|security risk named) (.+)|\s+contains\s+(.+)$/m ],
  ['Trend Micro FileScanner', ['/etc/iscan/vscan','vscan'],
    '-za -a {}', [0], qr/Found virus/m, qr/Found virus (.+) in/m ],
  ['drweb - DrWeb Antivirus',
    ['/usr/local/drweb/drweb', '/opt/drweb/drweb', 'drweb'],
    '-path={} -al -go -ot -cn -upn -ok-',
    [0,32], [1,9,33], qr' infected (?:with|by)(?: virus)? (.*)$'m ],
   ['Kaspersky Antivirus v5.5',
     ['/opt/kaspersky/kav4fs/bin/kav4fs-kavscanner',
      '/opt/kav/5.5/kav4unix/bin/kavscanner',
      '/opt/kav/5.5/kav4mailservers/bin/kavscanner', 'kavscanner'],
     '-i0 -xn -xp -mn -R -ePASBME {}/*', [0,10,15], [5,20,21,25],
     qr/(?:INFECTED|WARNING|SUSPICION|SUSPICIOUS) (.*)/m,
   ],
  ['Sophos Anti Virus (savscan)',
    ['/opt/sophos-av/bin/savscan', 'savscan'],
    '-nb -f -all -rec -ss -sc -archive -cab -mime -oe -tnef '.
    '--no-reset-atime {}',
    [0,2], qr/Virus .*? found/m,
    qr/^>>> Virus(?: fragment)? '?(.*?)'? found/m,
  ],
);
1;
```

echo 'maiad_enable="YES"' >> /etc/rc.conf - Sistem StartUP-a əlavə edirik.

maiad debug-sa - Debug rejimdə daemon- işə salırıq. Uğurlu nəticə aşağıdaki sətirləri çap etməlidir. Sonra dayandırmaq üçün **Ctrl+C** istifadə etmək lazımdır.

```
May  3 22:35:53 mail.saas.az /usr/local/sbin/maiad[80885]: SpamControl: done
May  3 22:35:53 mail.saas.az /usr/local/sbin/maiad[80892]: TIMING [total 4
ms] - bdb-open: 4 (100%), rundown: 0 (0%)
```

```
May  3 22:35:53 mail.saas.az /usr/local/sbin/maiad[80893]: TIMING [total 5
ms] - bdb-open: 5 (100%), rundown: 0 (0%)
```

/usr/local/etc/rc.d/maiad start - Daemon-u işə Salırıq

ps waux | grep maia - MaiaD daemonun proseslərdə olmasını axtarırıq

```
vscan 81069 22.0 3.1 262572 131108 - Ss 10:39PM 0:02.45 maiad (master) (perl)
vscan 81074 22.0 3.1 263956 131412 - S 10:39PM 0:00.01 maiad (virgin child) (perl)
vscan 81075 22.0 3.1 263956 131432 - S 10:39PM 0:00.01 maiad (virgin child) (perl)
```

/usr/local/etc/postfix/main.cf faylına aşağıdaki sətiri əlavə edirik(Ancaq biz postfix yüklənməsində artıq əlavə etmişdik):

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

/usr/local/etc/postfix/master.cf faylının sonuna aşağıdaki sətirləri əlavə edirik:

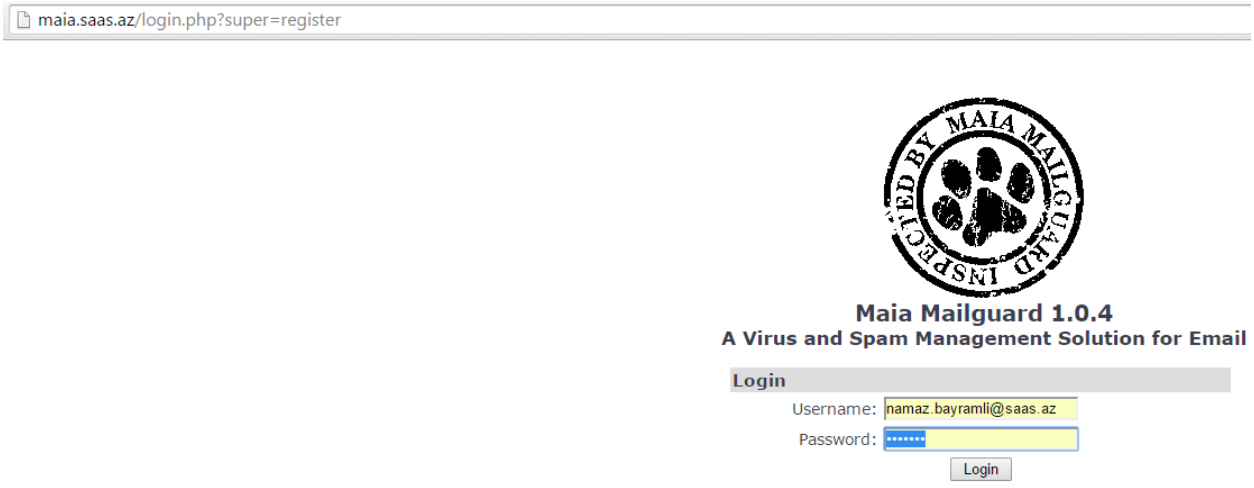
```
smtp-amavis unix -      -       n       -       2       smtp
      -o smtp_data_done_timeout=2400
      -o smtp_send_xforward_command=yes
      -o disable_dns_lookups=yes
      -o max_use=20
127.0.0.1:10025 inet    n       -       n       -       -       smtpd
      -o content_filter=
      -o local_recipient_maps=
      -o relay_recipient_maps=
      -o smtpd_restriction_classes=
      -o smtpd_delay_reject=no
      -o smtpd_client_restrictions=permit_mynetworks,reject
      -o smtpd_helo_restrictions=
      -o smtpd_sender_restrictions=
      -o smtpd_recipient_restrictions=permit_mynetworks,reject
      -o mynetworks_style=host
      -o mynetworks=127.0.0.0/8
      -o strict_rfc821_envelopes=yes
      -o smtpd_error_sleep_time=0
      -o smtpd_soft_error_limit=1001
      -o smtpd_hard_error_limit=1000
      -o smtpd_client_connection_count_limit=0
      -o smtpd_client_connection_rate_limit=0
      -o
receive_override_options=no_header_body_checks,no_unknown_recipient_checks,no
_address_mappings
```

crontab -u vscan -e – vscan istifadəçisi üçün aşağıdakı cron-ları əlavə edirik

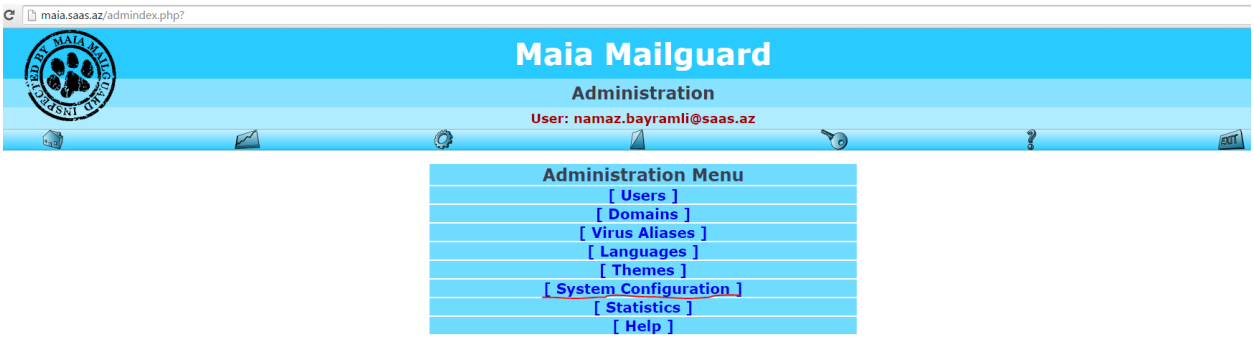
```
#Maia bazasında saxlanılması üçün yeni qaydaların yüklənilməsi.
30 4 * * * /var/amavisd/maia/scripts/load-sa-rules.pl > /dev/null

#SpamAssassin qatarı.
0 * * * * /var/amavisd/maia/scripts/process-quarantine.pl --learn --report >
/dev/null
```

```
#Hər saatın işə düşməsində olan statusların snapshotunun götürülməsi.
0 * * * * /var/amavisd/maia/scripts/stats-snapshot.pl > /dev/null

#Təsdiqlənməyən məktubların silinməsi.
0 23 * * * /var/amavisd/maia/scripts/expire-quarantine-cache.pl > /dev/null

#Karantin xəbərdarlığının yollanılması.
0 15 * * * /var/amavisd/maia/scripts/send-quarantine-reminders.pl > /dev/null

#İcmalların karantinini göstərmək.
0 15 * * * /var/amavisd/maia/scripts/send-quarantine-digests.pl > /dev/null

#Pik olmayan saatlardaq bayesian auto-expiry çağırılması.
25 2 * * * /usr/local/bin/sa-learn --sync --force-expire > /dev/null
```

http://maia.saas.az/login.php?super=register linkinə daxil oluruq ki, MAIA üçün super inzibatçını əlavə edək. Nəzərinizdə saxlayın ki, əlavə etmək istədiyiniz istifadəçi mütləq oncədə postfixadmin tərəfindən əlavə edilmiş mövcud istifadəçi olmalıdır. Bizim misalda namaz.bayramli@saas.az öncədən əlavə edildiyinə görə, onu **maia.saas.az**-in inzibatçısı əlavə edirik:



Giriş etdikdən sonra, yuxarıda olan **Admin**(açar simvoluna sıxırıq) -> **System Configuration**



Açılan səhifədə aşağıdaki şərtlər uyğun olmalıdır:

1. Əmin olun ki, təyin edilmiş bütün fayllar tam ünvanla göstərilmişdir.
2. Mütləq nəzərə alın ki, **"Mail Size Limit"** parametri **/etc/my.cnf** faylında olan **max_allowed_packet** həcmindən böyük olmalı deyil. Bu həcm həmçinin **/usr/local/etc/php.ini** faylında **upload_max_filesize = 10M** və **post_max_size = 10M** parametrlərində uyğun olaraq təyin edilməlidir. Ona görə ki, 10*1024*1024 nəticəsində 1048576 (10M) alınır.

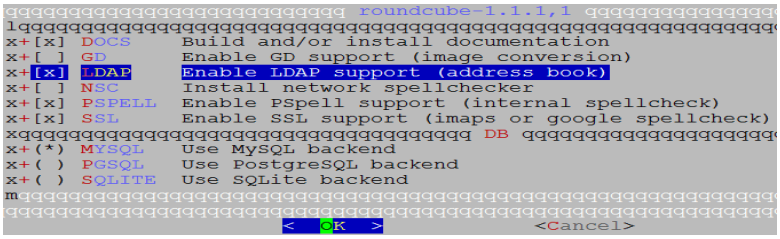
/usr/local/etc/rc.d/postfix restart – sonda postfix-ə yenidən yüklənmə əmri daxil edirik

RoundeCube Yüklənməsi və quraşdırılması

Roundcube – məktubun ötürülməsi və qəbul edilməsi üçün çox rahat web client-dir. Həmçinin Azerbaycan dili də mövcuddur. Çox gözəl görünüşlü interfeysə malikdir. Haqqında daha da ətraflı oxumaq istəsəniz, http://roundcube.net/ linkinə müraciət edə bilərsiniz.

cd /usr/ports/mail/roundcube – Port ünvanına daxil oluruq

make config – Lazımi modulları seçirik



make install – Yükləyirik

Roundcube üçün baza, istifadəçi və şifrəsini yaradırıq:

```
mysql -uroot -p
mysql> CREATE DATABASE roundcube;
mysql> GRANT ALL PRIVILEGES ON roundcube.* TO roundcube@localhost IDENTIFIED
BY 'roundcbedbpass';
mysql> FLUSH PRIVILEGES;
```

Rouncube bazasını dolduraq:

cd /usr/local/www/roundcube/SQL – SQL sxem faylı yerləşən ünvana daxil oluruq

mysql -u roundcube -p roundcube < mysql.initial.sql – SQL sxemini roundcube bazasına doldururuq

Quraşdırma faylını nüsxələyirik:

```
cp /usr/local/www/roundcube/config/config.inc.php.sample
/usr/local/www/roundcube/config/config.inc.php
```

/usr/local/www/roundcube/config/config.inc.php faylında olan sətirləri roundcube bazasına və istifadəçisi ilə şifrəsinə uyğun olaraq quraşdırırıq:

```
<?php
$config = array();
$config['db_dsnw'] = 'mysql://roundcube:roundcbedbpass@localhost/roundcube';
$config['default_host'] = 'localhost';
$config['smtp_server'] = 'localhost';
$config['smtp_port'] = 25;
$config['smtp_user'] = '';
$config['smtp_pass'] = '';
$config['support_url'] = '';
$config['product_name'] = 'OpenSource Webmail';
$config['des_key'] = 'rcmail-!24ByteDESkey*Str';
$config['plugins'] = array(
    'archive',
    'zipdownload',
);
$config['skin'] = 'larry';
```

/usr/local/www/roundcube/config/defaults.inc.php faylı aşağıdaki kimi olacaq(Vacib quraşdırmalar qırmızı rənglə seçilmişdir):

```
<?php
$config = array();
$config['db_dsnw'] = 'mysql://roundcube:roundcbedbpass@localhost/roundcube';
$config['db_dsnr'] = '';
$config['db_dsnw_noread'] = false;
$config['db_persistent'] = false;
$config['db_prefix'] = '';
$config['db_table_dsn'] = array(
);
$config['db_max_allowed_packet'] = null;
$config['debug_level'] = 4;
$config['log_driver'] = 'file';
$config['log_date_format'] = 'd-M-Y H:i:s O';
$config['log_session_id'] = 8;
$config['syslog_id'] = 'roundcube';
$config['syslog_facility'] = LOG_USER;
$config['per_user_logging'] = false;
$config['smtp_log'] = true;
$config['log_logins'] = false;
$config['log_session'] = false;
$config['sql_debug'] = false;
$config['imap_debug'] = false;
$config['ldap_debug'] = false;
$config['smtp_debug'] = false;
$config['default_host'] = 'localhost';
$config['default_port'] = 143;
$config['imap_auth_type'] = null;
```

```
$config['imap_conn_options'] = null;
$config['imap_timeout'] = 0;
$config['imap_auth_cid'] = null;
$config['imap_auth_pw'] = null;
$config['imap_delimiter'] = null;
$config['imap_ns_personal'] = null;
$config['imap_ns_other']    = null;
$config['imap_ns_shared']   = null;
$config['imap_force_caps'] = false;
$config['imap_force_lsub'] = false;
$config['imap_force_ns'] = false;
$config['imap_disabled_caps'] = array();
$config['imap_log_session'] = false;
$config['imap_cache'] = null;
$config['messages_cache'] = false;
$config['imap_cache_ttl'] = '10d';
$config['messages_cache_ttl'] = '10d';
$config['messages_cache_threshold'] = 50;
$config['smtp_server'] = '';
$config['smtp_port'] = 25;
$config['smtp_user'] = '';
$config['smtp_pass'] = '';
$config['smtp_auth_type'] = '';
$config['smtp_auth_cid'] = null;
$config['smtp_auth_pw'] = null;
$config['smtp_helo_host'] = '';
$config['smtp_timeout'] = 0;
$config['smtp_conn_options'] = null;
$config['ldap_cache'] = 'db';
$config['ldap_cache_ttl'] = '10m';
$config['enable_installer'] = false;
$config['dont_override'] = array();
$config['disabled_actions'] = array();
$config['advanced_prefs'] = array();
$config['support_url'] = '';
$config['skin_logo'] = null;
$config['auto_create_user'] = true;
$config['user_aliases'] = false;
$config['log_dir'] = RCUBE_INSTALL_PATH . 'logs/';
$config['temp_dir'] = RCUBE_INSTALL_PATH . 'temp/';
$config['temp_dir_ttl'] = '48h';
$config['force_https'] = false;
$config['use_https'] = false;
$config['login_autocomplete'] = 0;
$config['login_lc'] = 2;
$config['skin_include_php'] = false;
$config['display_version'] = false;
$config['session_lifetime'] = 10;
$config['session_domain'] = '';
$config['session_name'] = null;
$config['session_auth_name'] = null;
$config['session_path'] = null;
$config['session_storage'] = 'db';
```

```
$config['memcache_hosts'] = null;
$config['memcache_pconnect'] = true;
$config['memcache_timeout'] = 1;
$config['memcache_retry_interval'] = 15;
$config['ip_check'] = false;
$config['proxy_whitelist'] = array();
$config['referer_check'] = false;
$config['x_frame_options'] = 'sameorigin';
$config['des_key'] = 'rcmail-!24ByteDESkey*Str';
$config['username_domain'] = '';
$config['username_domain_forced'] = false;
$config['mail_domain'] = '';
$config['password_charset'] = 'ISO-8859-1';
$config['sendmail_delay'] = 0;
$config['max_recipients'] = 0;
$config['max_group_members'] = 0;
$config['product_name'] = 'Roundcube Webmail';
$config['useragent'] = 'OpenSource Webmail';
$config['include_host_config'] = false;
$config['generic_message_footer'] = '';
$config['generic_message_footer_html'] = '';
$config['http_received_header'] = false;
$config['http_received_header_encrypt'] = false;
$config['mail_header_delimiter'] = NULL;
$config['line_length'] = 72;
$config['send_format_flowed'] = true;
$config['mdn_use_from'] = false;
$config['identities_level'] = 0;
$config['identity_image_size'] = 64;
$config['client_mimetypes'] = null;  # null == default
$config['mime_magic'] = null;
$config['mime_types'] = null;
$config['im_identify_path'] = null;
$config['im_convert_path'] = null;
$config['image_thumbnail_size'] = 240;
$config['contact_photo_size'] = 160;
$config['email_dns_check'] = false;
$config['no_save_sent_messages'] = false;
$config['use_secure_urls'] = false;
$config['assets_path'] = '';
$config['assets_dir'] = '';
$config['plugins'] = array();
$config['message_sort_col'] = '';
$config['message_sort_order'] = 'DESC';
$config['list_cols'] = array('subject', 'status', 'fromto', 'date', 'size',
'flag', 'attachment');
$config['language'] = null;
$config['date_format'] = 'Y-m-d';
$config['date_formats'] = array('Y-m-d', 'Y/m/d', 'Y.m.d', 'd-m-Y', 'd/m/Y',
'd.m.Y', 'j.n.Y');
$config['time_format'] = 'H:i';
$config['time_formats'] = array('G:i', 'H:i', 'g:i a', 'h:i A');
$config['date_short'] = 'D H:i';
```

```
$config['date_long'] = 'Y-m-d H:i';
$config['drafts_mbox'] = 'Drafts';
$config['junk_mbox'] = 'Junk';
$config['sent_mbox'] = 'Sent';
$config['trash_mbox'] = 'Trash';
$config['create_default_folders'] = false;
$config['protect_default_folders'] = true;
$config['show_real_foldernames'] = false;
$config['quota_zero_as_unlimited'] = false;
$config['enable_spellcheck'] = true;
$config['spellcheck_dictionary'] = false;
$config['spellcheck_engine'] = 'googie';
$config['spellcheck_uri'] = '';
$config['spellcheck_languages'] = NULL;
$config['spellcheck_ignore_caps'] = false;
$config['spellcheck_ignore_nums'] = false;
$config['spellcheck_ignore_syms'] = false;
$config['recipients_separator'] = ',';
$config['sig_max_lines'] = 15;
$config['max_pagesize'] = 200;
$config['min_refresh_interval'] = 60;
$config['upload_progress'] = false;
$config['undo_timeout'] = 0;
$config['compose_responses_static'] = array(
);
$config['address_book_type'] = 'sql';
$config['ldap_public'] = array();
$config['autocomplete_addressbooks'] = array('sql');
$config['autocomplete_min_length'] = 1;
$config['autocomplete_threads'] = 0;
$config['autocomplete_max'] = 15;
$config['address_template'] = '{street}<br/>{locality}
{zipcode}<br/>{country} {region}';
$config['addressbook_search_mode'] = 0;
$config['contact_search_name'] = '{name} <{email}>';
$config['default_charset'] = 'ISO-8859-1';
$config['skin'] = 'larry';
$config['standard_windows'] = false;
$config['mail_pagesize'] = 50;
$config['addressbook_pagesize'] = 50;
$config['addressbook_sort_col'] = 'surname';
$config['addressbook_name_listing'] = 0;
$config['timezone'] = 'auto';
$config['prefer_html'] = true;
$config['show_images'] = 0;
$config['message_extwin'] = false;
$config['compose_extwin'] = false;
$config['htmleditor'] = 0;
$config['compose_save_localstorage'] = true;
$config['prettydate'] = true;
$config['draft_autosave'] = 300;
$config['preview_pane'] = false;
$config['preview_pane_mark_read'] = 0;
```

```
$config['logout_purge'] = false;
$config['logout_expunge'] = false;
$config['inline_images'] = true;
$config['mime_param_folding'] = 1;
$config['skip_deleted'] = false;
$config['read_when_deleted'] = true;
$config['flag_for_deletion'] = false;
$config['refresh_interval'] = 60;
$config['check_all_folders'] = false;
$config['display_next'] = true;
$config['default_list_mode'] = 'list';
$config['autoexpand_threads'] = 0;
$config['reply_mode'] = 0;
$config['strip_existing_sig'] = true;
$config['show_sig'] = 1;
$config['force_7bit'] = false;
$config['search_mods'] = null;
$config['addressbook_search_mods'] = null;
$config['delete_always'] = false;
$config['delete_junk'] = false;
$config['mdn_requests'] = 0;
$config['mdn_default'] = 0;
$config['dsn_default'] = 0;
$config['reply_same_folder'] = false;
$config['forward_attachment'] = false;
$config['default_addressbook'] = null;
$config['spellcheck_before_send'] = false;
$config['autocomplete_single'] = false;
$config['default_font'] = 'Verdana';
$config['default_font_size'] = '10pt';
$config['message_show_email'] = false;
$config['reply_all_mode'] = 0;
```

```
chown -R www:www /usr/local/www/roundcube/      - Roundcube-un lazımi
                                                  istifadəçi və qrup adından
                                                  işə düşməsi üçün yetkilər
                                                  təyin edirik

chmod 600 /usr/local/www/roundcube/config/*     - Bütün roundcube quraşdırma
                                                  fayllarını təhlükəsiz edirik
```

/usr/local/domen/mpanel.saas.az virtual host faylına aşağıdaki sətirləri əlavə edirik ki, roundcube panel işləsin:

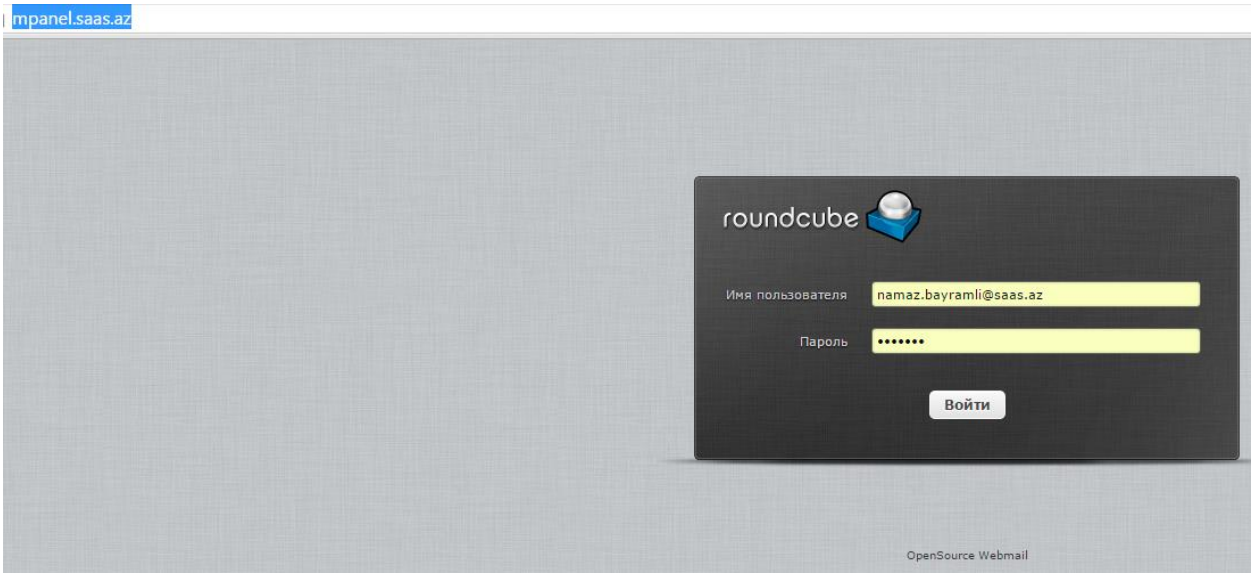
```
<VirtualHost *:80>
        ServerAdmin jamal.shahverdiyev@saas.az
        ServerName mpanel.saas.az
        AcceptPathInfo On
        DocumentRoot /usr/local/www/roundcube/
<Directory "/usr/local/www/roundcube">
        AllowOverride All
        Require all granted
</Directory>
```

```
        ErrorLog /var/log/httpd/mpanel-error.log
        CustomLog /var/log/httpd/mpanel-access.log combined
</VirtualHost>
```

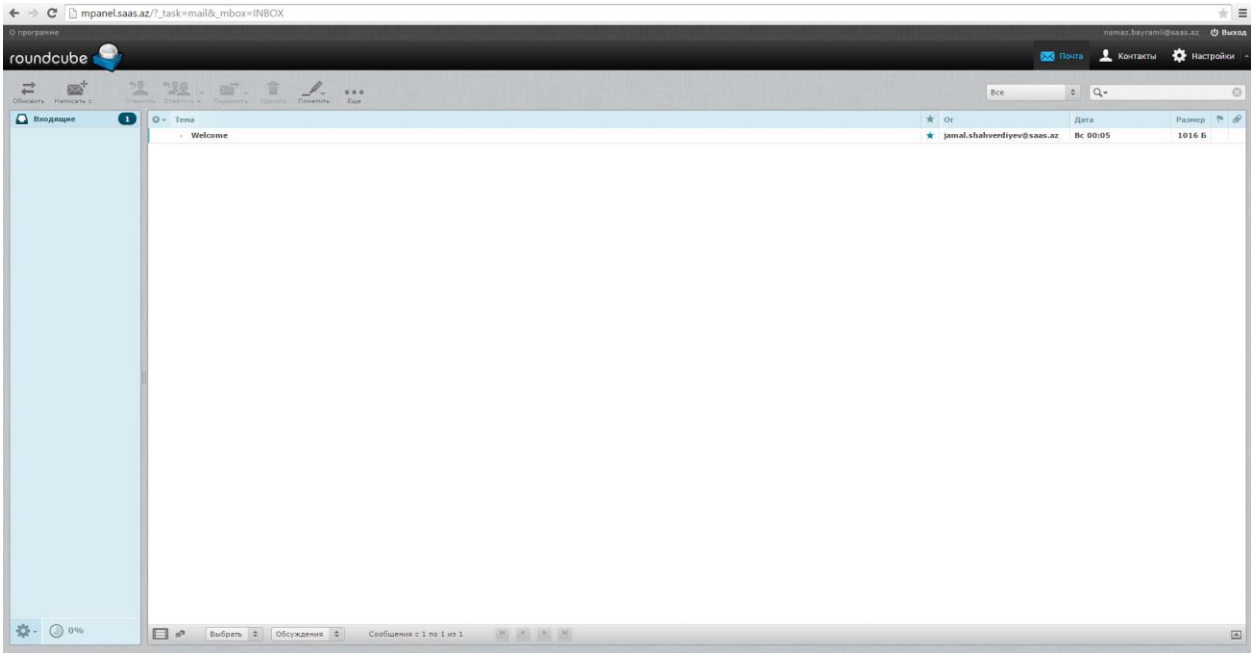
touch /var/log/httpd/mpanel-error.log /var/log/httpd/mpanel-access.log – Jurnal fayllarını yaradırıq

apachectl graceful – Apache-a restart əmri daxil edirik

http://mpanel.saas.az/ linkinə müraciət edirik və aşağıdaki şəkildəki nəticəni əldə edirik:



Daxil olduqdan sonra əldə etdiyimiz son nəticə aşağıdaki kimi olacaq:

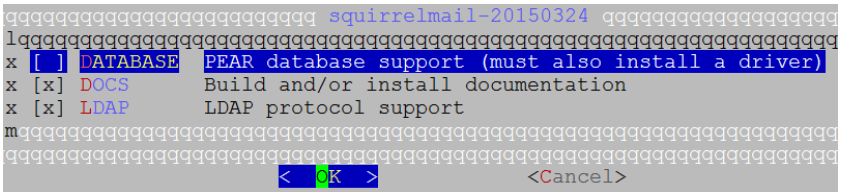


Qeyd: Əgər problem yaranarsa jurnallar **/usr/local/www/roundcube/logs/** qovluğunda yığılacaq.

SquirrelMail-in yuklənməsi və quraşdırılması

SquirrelMail – active inkişaf etdirilir. Böyük pluginlərə sahibdir. Rəsmi saytından http://www.squirrelmail.org/ dahada ətraflı məlumat əldə edə bilərsiniz.

cd /usr/ports/mail/squirrelmail - Port ünvanına daxil oluruq
make config - Lazımi modulları seçirik



make install - Yükləyirik

/usr/local/domen/sqmail.saas.az faylına aşağıdakı sətirləri əlavə edirik ki, **sqmail.saas.az** adlı virtual host işləyə bilsin:

```
<VirtualHost *:80>
        ServerAdmin jamal.shahverdiyev@saas.az
        ServerName sqmail.saas.az
        AcceptPathInfo On
        DocumentRoot /usr/local/www/squirrelmail/
<Directory "/usr/local/www/squirrelmail/">
        AllowOverride All
        Require all granted
</Directory>
        ErrorLog /var/log/httpd/squirrelmail-error.log
        CustomLog /var/log/httpd/squirrelmail-access.log combined
</VirtualHost>
```

Jurnal fayllarını yaradırıq:

```
touch /var/log/httpd/squirrelmail-error.log /var/log/httpd/squirrelmail-access.log
```

chown -R www:www /usr/local/www/squirrelmail/ - SquirrelMail-in ev qovluğunu apache üçün təyin edirik ki, webdən işlədə bilək.

apachectl graceful - WEB serveri yenidən işə salırıq

/usr/local/etc/php.ini faylında aşağıdaki sətirləri uyğun olaraq, düzəldirik:

```
file_uploads = On
short_open_tag = On
```

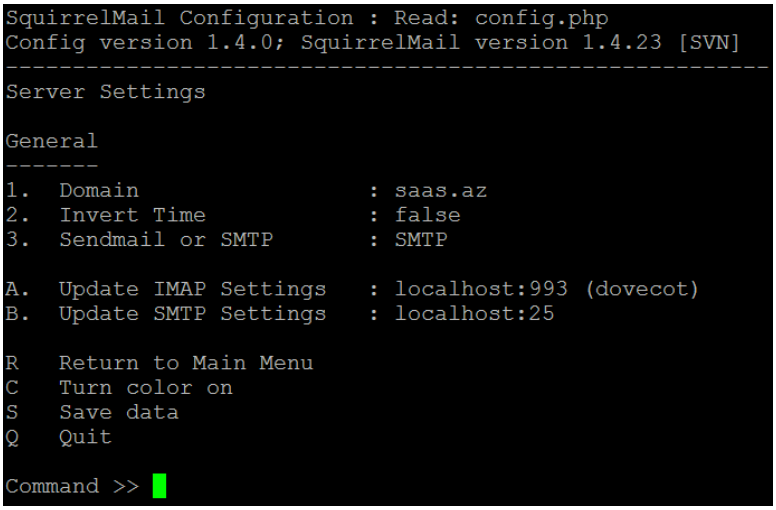
cd /usr/local/www/squirrelmail && ./configure - SquirrelMail-i quraşdırırıq. Aşağıdakı səhifə açılacaq. Uyğun olaraq rəqəmlər və simvollarla keçid edərək. Quraşdırmaq lazımdır. Ancaq hər quraşdırmadan sonra **S(Save data)** düyməsinə sıxmağı unutmayın.

```
SquirrelMail Configuration : Read: config.php
Config version 1.4.0; SquirrelMail version 1.4.23 [SVN]
---------------------------------------------------------
Main Menu --
1.  Organization Preferences
2.  Server Settings
3.  Folder Defaults
4.  General Options
5.  Themes
6.  Address Books
7.  Message of the Day (MOTD)
8.  Plugins
9.  Database
10. Languages

D.  Set pre-defined settings for specific IMAP servers

C   Turn color on
S   Save data
Q   Quit
```

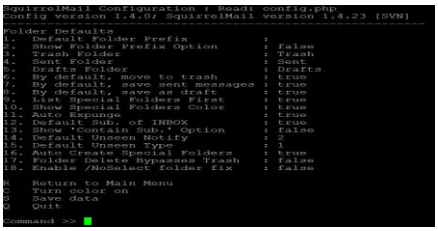
1-i sıxırıq və aşağıdaki şəkildəki kimi quraşdırırıq:

```
SquirrelMail Configuration : Read: config.php
Config version 1.4.0; SquirrelMail version 1.4.23 [SVN]
---------------------------------------------------------
Organization Preferences
1.  Organization Name      : SaaS
2.  Organization Logo      : ../images/sm_logo.png
3.  Org. Logo Width/Height : (308/111)
4.  Organization Title     : OpenSource Mail Server
5.  Signout Page           :
6.  Top Frame              : _top
7.  Provider link          : http://sqmail.saas.az
8.  Provider name          : OpenSource

R   Return to Main Menu
C   Turn color on
S   Save data
Q   Quit

Command >>
```

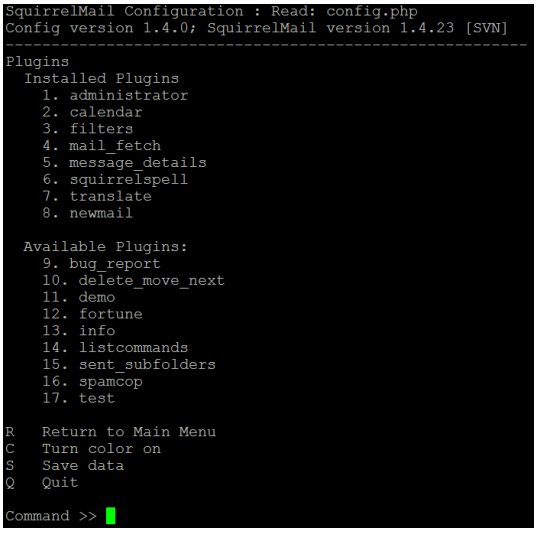
Sonra **R** düyməsini sıxaraq əsas menyuya daxil oluruq və **2** düyməsini sıxıb aşağıdakı şəkildəki kimi IMAPS-i quraşdırırıq(**Server software** bölümündə **dovecot** seçməyi unutmayın):



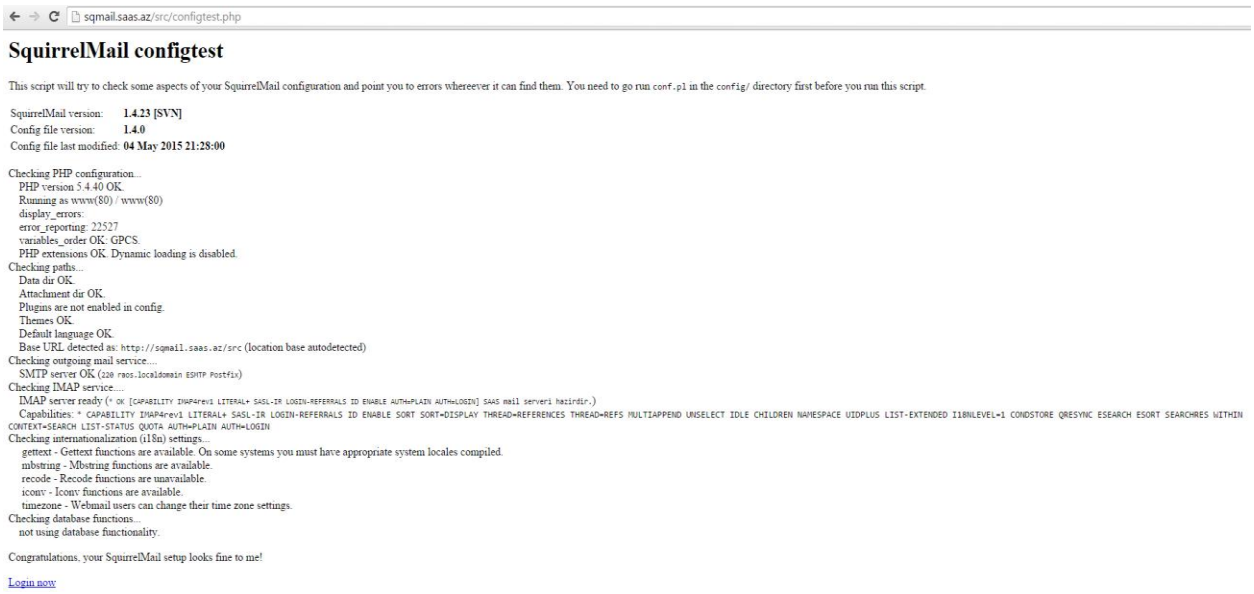
Sonra **R** düyməsini sıxıb əsas menyuya qayıdırıq və **3** düyməsini sıxıb quraşdırırıq(Quraşdırma aşağıdaki şəkildəki kimi olmalıdır. **S** ilə yadda saxlamağı unutmayin):



Sonra **8** düyməsi ilə pluginləri seçib aşağıdaki kimi, quraşdırırıq(**S** ilə yadda saxlayıb, **Q** düyməsini sıxaraq çıxırıq):



Test üçün http://sqmail.saas.az/src/configtest.php linkinə müraciət edirik və aşağıdaki nəticəni əldə etməliyik(**Login now** düyməsinə sıxırıq):



Açılan pəncərədə istifadəçi adı və şifrəni daxil edirik:

Sonda əldə etdiyimiz səhifə dəyişdirilmiş tema **Forest** ilədir və aşağıdakı kimidir:

Qeyd: Ancaq siz eynilə **Horde** və **Rainloop**-dan da istifadə edə bilərsiniz.

Mailman yüklenilməsi və quraşdırılması

Mailman – mektubların göndərilməsi üçün istifadə edilən və dəstəklənən çox gözəl alətdir. Əgər siz göndərilmə serveri yaratmaq istəyirsinizsə, bu aləti seçməniz düzgün qərardır. Ətraflı melumatı rəsmi saytından http://www.gnu.org/software/mailman/index.html əldə edə bilərsiniz. Mövcud misalımızda 2.1.20-ci versiyadan istifadə edilmişdir.

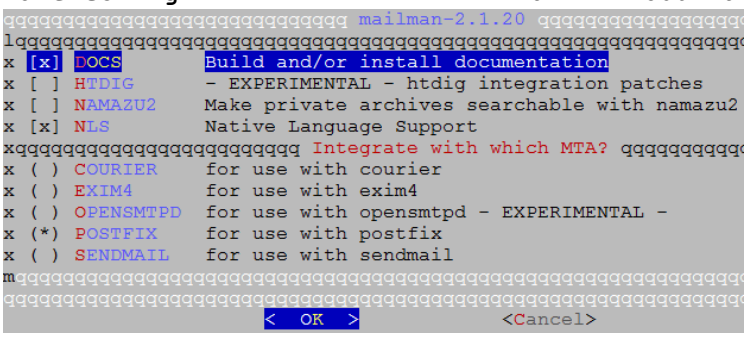
Ilk işimiz apache WEB serverimizin **py** genişlənməli kodların işə sala bilməsi üçün CGI-i aktive etməkdir. Bunun üçün **/usr/local/etc/apache24/httpd.conf** quraşdırma faylında aşağıdaki sətirlərin qarşısından şərhi silirik:

```
<IfModule mpm_prefork_module>
        LoadModule cgi_module libexec/apache24/mod_cgi.so
</IfModule>

AddHandler cgi-script .cgi
```

/usr/local/etc/rc.d/apache24 restart - WEB Serverimizi yenidən işə salırıq ki, dəyişikliklər aktivləşsin

cd /usr/ports/mail/mailman – Port ünvanına daxil oluruq

make config - Lazımi modulları seçirik



make install - Yükləyirik

echo 'mailman_enable="YES"' >> /etc/rc.conf - StartUP-a əlavə edirik

Postfix-To-Mailman scriptini əldə edək:

cd /usr/local/mailman - Mailman qovluğuna daxil oluruq

fetch http://www.gurulabs.com/downloads/postfix-to-mailman-2.1.py - Scripti endiririk

mv postfix-to-mailman-2.1.py postfix-to-mailman.py - Scriptin adını dəyişirik

chmod 750 postfix-to-mailman.py - Lazımi yetki təyin edirik

chown mailman:mailman postfix-to-mailman.py - Fayl hüquqlarını dəyişirik

which python2.7 - Python binary faylının ünvanını tapırıq

/usr/local/mailman/postfix-to-mailman.py faylda aşağıdaki sətirlərdə dəyişiklik edirik:

```
#!/usr/local/bin/python2.7
...
MailmanHome = "/usr/local/mailman";
MailmanOwner = "postmaster@saas.az";
```

/usr/local/etc/postfix/main.cf faylında aşağıdaki qırmızı rənglə qeyd edilmiş ünvanları dəyişiklik edirik:

```
...
relay_domains = mysql:/usr/local/etc/postfix/mysql_relay_domains_maps.cf
lists.saas.az
...
transport_maps = hash:/usr/local/etc/postfix/transport
vacation_destination_recipient_limit = 1
mailman_destination_recipient_limit = 1
```

/usr/local/etc/postfix/transport faylına ötürücünü əlavə edirik:

```
echo 'lists.saas.az mailman:' >> /usr/local/etc/postfix/transport
```

/usr/local/etc/postfix/master.cf faylının sonuna aşağıdaki sətirləri əlavə edirik:

```
mailman   unix  -       n       n       -       -       pipe
  flags=FR user=mailman:mailman argv=/usr/local/mailman/postfix-to-mailman.py
  ${nexthop} ${user}
```

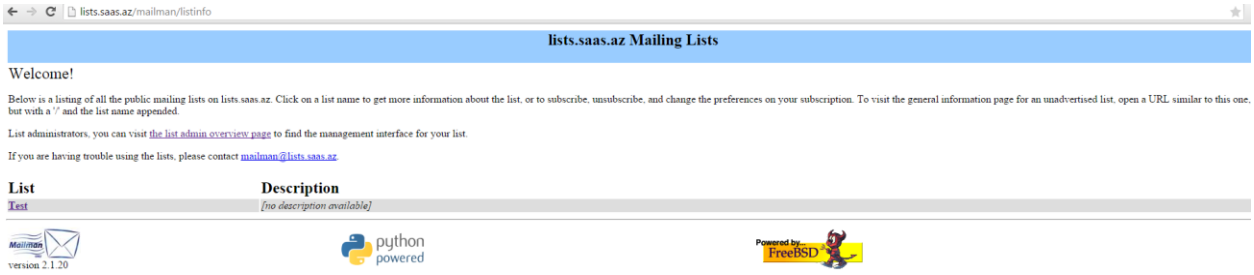
```
postmap /usr/local/etc/postfix/transport  - Transport faylının bazasını
                                            yeniləyirik

/usr/local/etc/rc.d/postfix restart       - Postfix daemonu yenidən işə
                                            salırıq
```

/usr/local/domen/lists.saas.az adli **lists.saas.az** saytı üçün virtual host yaradırıq və məzmununa aşağıdaki sətirləri əlavə edirik:

```
<Virtualhost *:80>
        ServerAdmin webmaster@saas.az
        DocumentRoot "/usr/local/mailman"
        ServerName lists.saas.az
        ServerAlias lists.saas.az
        ScriptAlias /cgi-bin/ "/usr/local/mailman/cgi-bin/"
        ScriptAlias /mailman/ "/usr/local/mailman/cgi-bin/"
        Alias /pipermail "/usr/local/mailman/archives/public"
        Alias /icons "/usr/local/mailman/icons"
<Directory "/usr/local/mailman">
        AllowOverride All
        Options FollowSymlinks ExecCGI
        Require all granted
</Directory>
        ErrorLog /var/log/httpd/mailman-error.log
        CustomLog /var/log/httpd/mailman-access.log combined
</Virtualhost>
```

```
apachectl graceful      - WEB serverimizi yenidən işə salırıq ki,
                        dəyişikliklərimiz aktivləşsin.
```

Sonra http://lists.saas.az/mailman/listinfo linkinə müraciət edirik və aşağıdaki şəkildə çap edilən səhifəni əldə edirik:



Sistemimiz üçün şifrə təyin edirik:

cd /usr/local/mailman - MailMan ünvanına daxil oluruq
bin/mmsitepass - Şifrə qeyd əmrini daxil edirik
New site password: **shifre**
Again to confirm password: **shifre_tekrar**
Password changed.

Yeni siyahı yaradırıq:

bin/newlist - Yeni siyahı generasiya edirik
Enter the name of the list: **mailman**
Enter the email of the person running the list: **namaz.bayramli@saas.az**
Initial mailman password: **list_shifresi**
Hit enter to notify mailman owner...

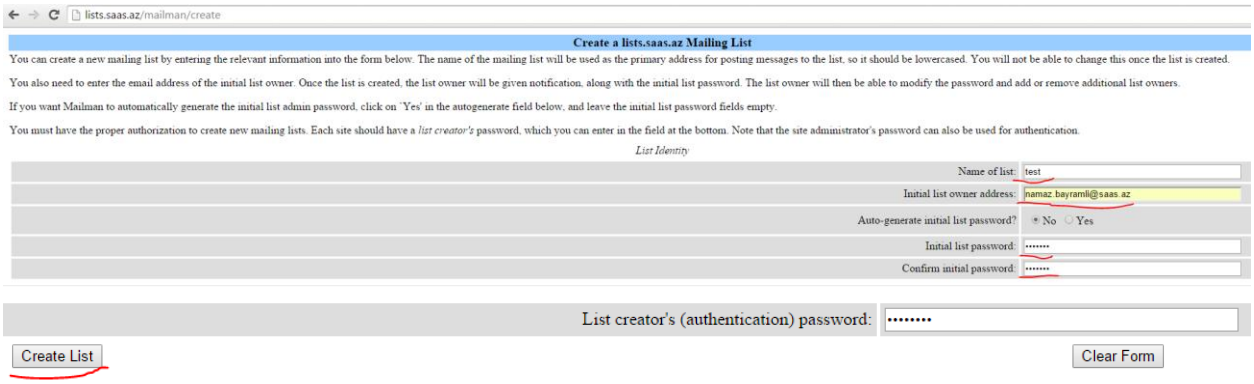
Enter sıxırıq ki, MailMan sahibinə xəbərdarlıq yollansın.

Mailman quraşdırma faylına siyahını əlavə edirik:

echo "add_virtualhost('lists.saas.az','lists.saas.az')" >> /usr/local/mailman/Mailman/mm_cfg.py

/usr/local/etc/rc.d/mailman start - Mailman-i işə salırıq. İşə salma müddətində səhvləri özü düzəldib, daemon-un yenidən işə salınması haqqında sizə məlumat verəcək.

Test üçün http://lists.saas.az/mailman/listinfo linkinə daxil oluruq. Açılan səhifədə **the list admin overview page** linkinə sıxırıq və sonra **create a new mailing list** sıxırıq ki, yeni istifadəçilər siyahısı yaradaq. Aşağıdakı qaydada siyahı əlavə edirik:



Vacibdir

ANY ünvanlaması siyahısını **lists.saas.az** üçün kənar serverlərdən gələn istənilən müraciət qəbul edəcək. Əgər bu spamer hücumu olsa, onun qarşısını almaq mümkün olmayacaq. Ona görə də biz hər bir ünvanlandırıcı siyahısı üçün ayrı xəritələnmə siyahısı hazırlamalıyıq.

Bütün ünvanlandırıcı siyahısını tapırıq:

```
cd /usr/local/mailman
bin/genaliases
```

/usr/local/etc/postfix/relay_recipients faylı yaradaq və öncəki əmrdən əldə etdiyimiz nəticəni tamlıqla bu fayla aşağıdaki sintaksislə əlavə edək. Hər bir ünvanın sonunda "**OK**" olmalıdır. Digər sözlə desək bizim users@lists.saas.az adlı yayımlanma siyahımız mövcuddur.

```
users@lists.saas.az OK
users-admin@lists.saas.az OK
users-bounces@lists.saas.az OK
users-confirm@lists.saas.az OK
users-join@lists.saas.az OK
users-leave@lists.saas.az OK
users-owner@lists.saas.az OK
users-request@lists.saas.az OK
users-subscribe@lists.saas.az OK
users-unsubscribe@lists.saas.az OK
```

postmap /usr/local/etc/postfix/relay_recipients - Postfix üçün xəritələnmə faylı yaradırıq

Qeyd: Siz hər yeni domain üçün yuxarıda edilən ardıcıllığı təkrarlamalısınız, əks halda postfix məktub ünvanlarını qəbul etməyəcək. Sözsüz ki, bütün üvnanları bir faylda qeyd etmək olar ancaq, hər dəfə **postmap** əmrindən istifadə etməyi unutmayın. Həmçinin Postfix-də olan **'relay_recipients'** direktivində hər edilən dəyişiklikdən sonra, postfix daemon-a restart etməyi unutmayın.

/usr/local/etc/postfix/main.cf faylında aşağıdaki dəyişikliyi edin:

```
...
relay_recipient_maps = hash:/usr/local/etc/postfix/relay_recipients
...
```

postfix reload - Postfix quraşdırmalarını yenidən oxuyuruq

http://lists.saas.az/mailman/listinfo linkinə daxil olun. Yeni yaradılmış siyahısının adının yəni **Test**-in üstünə sıxın. **"Subscribing to listname"** bölümündə olan çatışmamazlığı doldurun və göndərin düyməsinə sıxın. Ekektron məktubunuzu yoxlayən və məktubu təsdiqləyin. listname@lists.domain.tld ünvanına məktub yollayın. Əgər hər şey düzgün qurulubsa, məktub gedəcək və bütün mümkün ola biləcək səhvlər **/var/log/maillog** ünvanına yığılacaq. Əgər səhvlər yoxdursa onda, http://lists.saas.az/pipermail/listname linkini yoxlayın ki, gondərilmiş məktuba baxaq. Həmçinin serverinizdə olan normal istifadəçiyə mailman@domain.tld adlı alias yaratmağı unutmayın əks, halda **/var/log/maillog** faylında səhvləri görəcəksiniz.

Yeni göndərilmənin yaradılması üçün ardıcıllıq aşağıdaki kimi olacaq:

Mailman siyahımıza yenisi olan lists2.domain2.tld əlavə edək:

```
cd /usr/local/mailman
bin/newlist -u lists.domain2.tld -e lists.domain2.tld listname
```

Mailman quraşdırma faylına yeni siyahı əlavə edirik:

```
echo "add_virtualhost('lists.domain2.tld','lists.domain2.tld')" >>
/usr/local/mailman/Mailman/mm_cfg.py
```

/usr/local/etc/postfix/main.cf faylımızda **relay_domains** bölümünü aşağıdakı şəklə gətiririk:

```
...
relay_domains = mysql:/usr/local/etc/postfix/mysql_relay_domains_maps.cf
lists.saas.az lists.domain2.tld
...
```

Postfixin transport faylına yenisini əlavə edirik:

```
echo 'lists.domain2.tld mailman:' >> /usr/local/etc/postfix/transport
```

postmap /usr/local/etc/postfix/transport – Transport xəritələnməsini yeniləyirik

postfix reload - Postfix-i yenidən işə salırıq

/usr/local/domen/lists.domain.tld faylına aşağıdaki sətirləri əlavə edirik:

```
<Virtualhost *:80>
        ServerAdmin webmaster@domain2.tld
        DocumentRoot "/usr/local/mailman"
        ServerName lists.domain2.tld
        ServerAlias lists.domain2.tld
```

```
        ScriptAlias /cgi-bin/ "/usr/local/mailman/cgi-bin/"
        ScriptAlias /mailman/ "/usr/local/mailman/cgi-bin/"
        Alias /pipermail "/usr/local/mailman/archives/public"
        Alias /icons "/usr/local/mailman/icons"
<Directory "/usr/local/mailman">
        AllowOverride All
        Options FollowSymlinks ExecCGI
        Require all granted
</Directory>
        ErrorLog /var/log/httpd/domain2-error.log
        CustomLog /var/log/httpd/domain2-access.log combined
</Virtualhost>
```

/usr/local/etc/rc.d/apache24 restart - sonda **apache24** web server yenidən işə salırıq

Nəticədə http://lists.domain2.tld/mailman/listinfo səhifəsini yoxlayırıq.

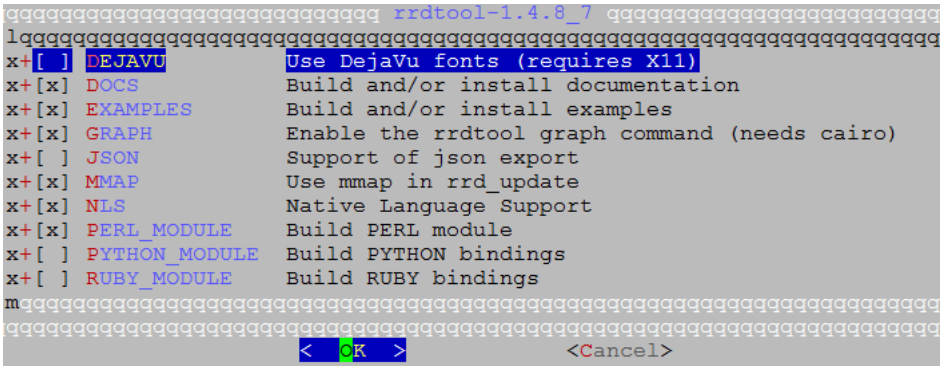
Mailgraph yüklənilməsi və quraşdırılması

Mailgraph – Sizin poçt serverdən statistikanın əldə edilməsi üçün əla CGI scriptdir. Haqqında daha ətraflı http://mailgraph.schweikert.ch/ rəsmi linkindən oxuya bilərsiniz.

RRDTool-u yükləyək:

cd /usr/ports/databases/rrdtool - Port ünvanına daxil oluruq

make config - Susmaya görə olan modulları seçirik



make install - Yükləyirik

Mailgraph-i patch edirik:

```
cd /usr/ports/mail/mailgraph
make extract
fetch http://www.purplehat.org/downloads/postfix_guide/mailgraph-1.14-postfix.diff
patch -p0 < mailgraph-1.14-postfix.diff
make all install clean
```

mailgraph-1.14-postfix.diff faylının məzmunu aşağıdaki kimi olacaq:

```
--- files/mailgraph.in.orig     Tue Sep 18 16:25:41 2007
+++ files/mailgraph.in  Tue Sep 18 16:25:19 2007
@@ -27,7 +27,7 @@
 : ${mailgraph_enable="NO"}
 : ${mailgraph_pidfile="%%DATADIR%%/mailgraph.pid"}
 : ${mailgraph_flags="--logfile /var/log/maillog --daemon-rrd=%%DATADIR%% --
ignore-localhost --daemon --daemon-pid=${mailgraph_pidfile}"}
-: ${mailgraph_user="%%MAILGRAPH_USER%%"}
+# : ${mailgraph_user="%%MAILGRAPH_USER%%"}
 : ${mailgraph_chdir="%%DATADIR%%"}

 load_rc_config $name
--- work/mailgraph-1.14/mailgraph.pl.orig       Tue Sep 18 16:26:18 2007
+++ work/mailgraph-1.14/mailgraph.pl    Tue Sep 18 16:27:30 2007
@@ -575,7 +575,10 @@
        if($prog =~ /^postfix\/(.*)/) {
                my $prog = $1;
                if($prog eq 'smtp') {
-                       if($text =~ /\bstatus=sent\b/) {
+                       if($text =~ /VIRUS/) {
+                               event($time, 'virus');
+                       }
+                       elsif($text =~ /\bstatus=sent\b/) {
                                return if $opt{'ignore-localhost'} and
                                          $text =~
/\brelay=[^\s\[]*\[127\.0\.0\.1\]/;
                                if(defined $opt{'ignore-host-re'}) {
```

chown -R root:www /var/log/maillog - Faylı web server tərəfindən oxunulabilən edirik ki, mailgraph daemon işə düşsün.

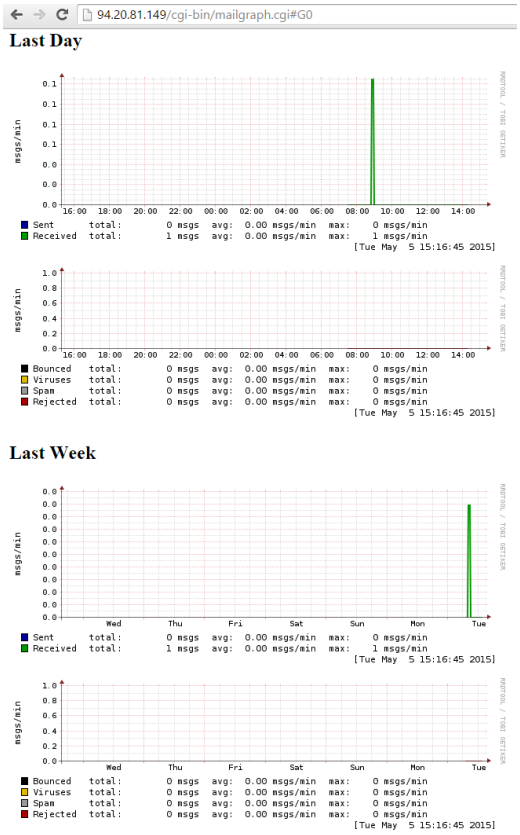
echo 'mailgraph_enable="YES"' >> /etc/rc.conf - StartUP-a əlavə edirik

/usr/local/etc/rc.d/mailgraph start – İşə salırıq

CGI script və CSS scriptləri lazımi ünvanlara nüsxələyirik:

```
cp /usr/local/www/cgi-bin/mailgraph.cgi /usr/local/www/apache24/cgi-bin/
cp -Rp /usr/local/www/data/mailgraph/ /usr/local/www/apache24/data/
```

Nəticədə WEB səhifəmizdən açdıqda, aşağıdaki nəticəni əldə etmiş olacayıq:



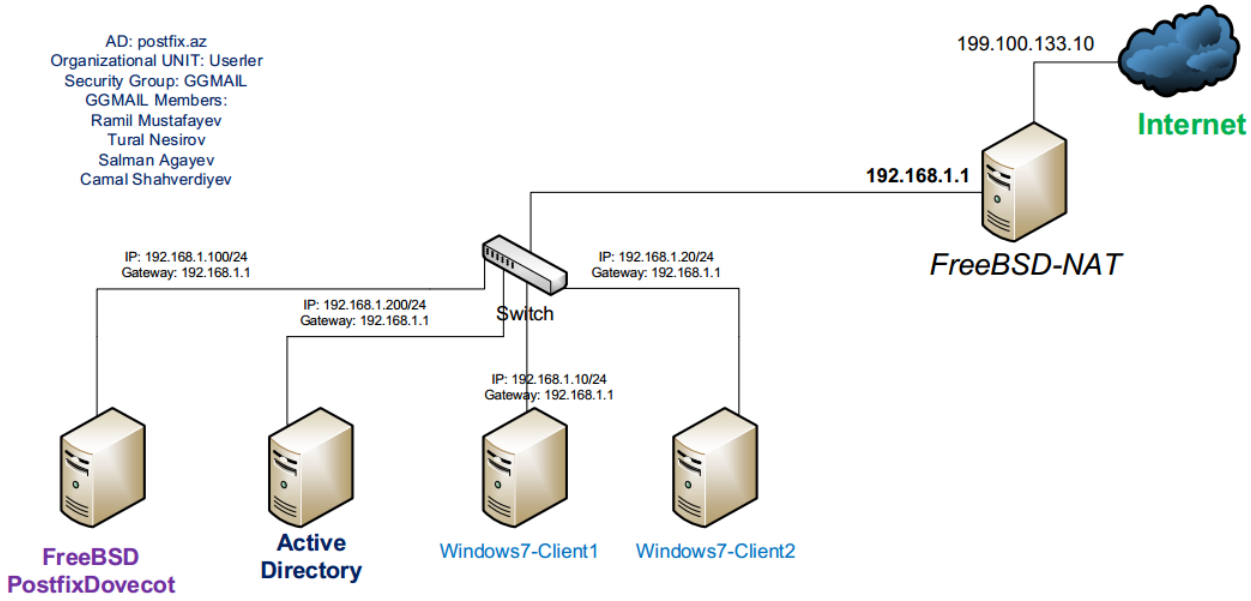
FreeBSD Postfix Dovecot ilə AD integrasiyası

Məqsədimiz FreeBSD əməliyyat sisteminin üzərində olan Postfix(SMTP/S) və Dovecot(POP/S, IMAP/S)-i Active Diretory-nin MSLDAP istifadəçi bazası ilə inteqrasiya eləməkdir. Bunu ona görə edirik ki, şirkətin daxilində olan istifadəçi adı və şifrə fərqli bazalarda olmasın. İstifadəçi fərqli servislərdən yararlanmaq üçün bir neçə şifrə yadında saxladıqda onu bezdirəcək və bu da narahatçılığa gətirib çıxaracaq. Bu texnologiyaya Single Sign On deyilir.

Tələb edilən Server Təminatları.

FreeBSD: 9.1 AMD64(Postfix Server – 192.168.1.100) - 1 ədəd
Windows Server 2008 R2 x64(Active Directory – 192.168.1.200) - 1 ədəd
Windows7 32(MS Outlook 2007 Client – 192.168.1.10 ve 20) - 2 ədəd

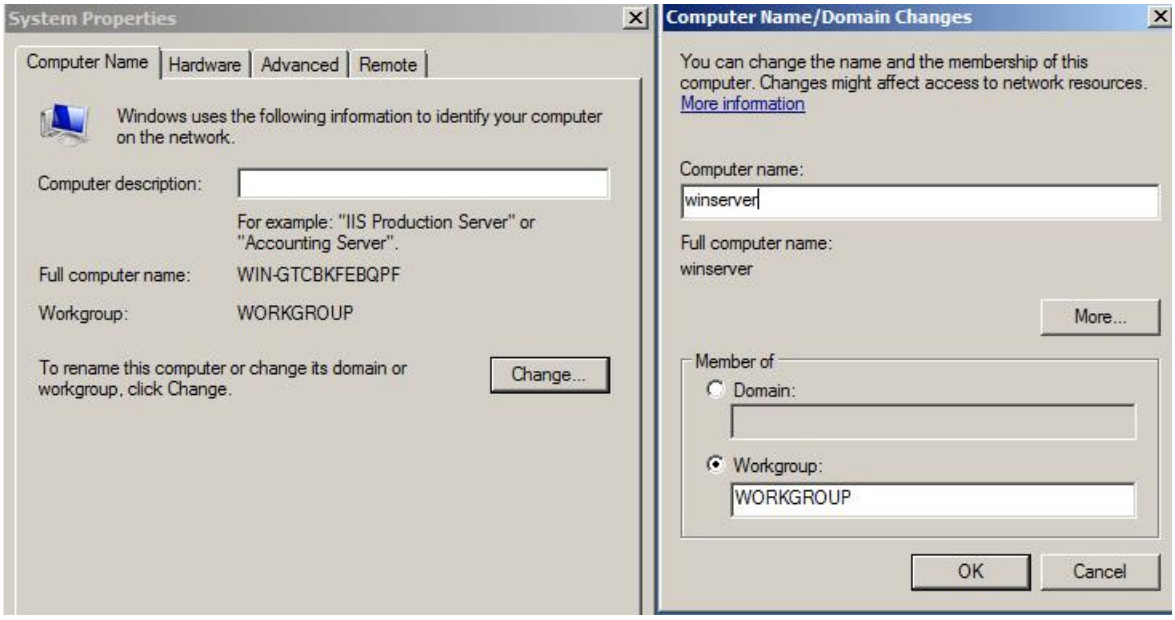
Bu serverlərdən əlavə 1 ədəd-də **FreeBSD-NAT** server vardır hansı ki, yuxarıda göstərilən avadanlıqları NAT ilə İnternetə çıxarır ki, testlərimizi edə bilək. Şəbəkə quruluşu şəkildəki kimidir:



Virtual Maşınlarımız Vmware Workstation-un **VmNet3** şəbəkəsində işləyir həmçinin **VmNet3** eynilə bizim **LoopBACK** şəbəkə kartımız ilə **Bridge** edilmişdir.

Windows 2008 serverdə bütün görəcəyimiz işlərdən yüklənmə bitən kimi **Computer Name**-i dəyişmək lazımdır. Biz **'winserver'** istifadə edəcəyik.
Start -> Computer -> Right Click -> Properties -> Change settings -> Change (Computer Name Tab-ında) -> Adı yazırıq -> Ok -> OK -> Close -> Restart now

GGEMA

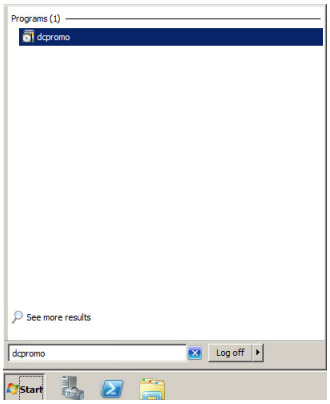


Həmçinin FireWall-ı söndürməyi unutmayın

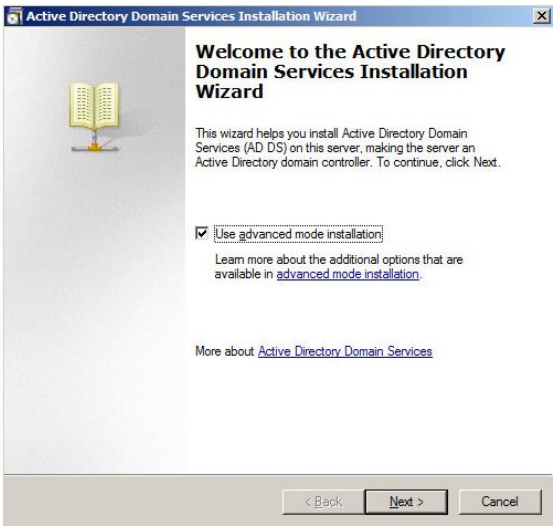
Sonrakı işimiz **Windows 2008 Serverdə Domain Controller** qaldırıb tələb edilən **Unit,Group** və istifadəçiləri əlavə etməkdir. Həmçinin yeni Group-u yeni Unit-ə əlavə edib, sonra ardınca da yeni istifadəçiləri həmin Group-a əlavə edirik.

Qeyd: Unutmayın ki, **sAMAccountName** istifadəçinin Domain-ə girişi zamanı istifadə edəcəyi Atributdur. **mail** isə FreeBSD mail serverin LDAP-dan istifadəçilər haqqında məlumat aldıqda istifadəçi email unvanını göstərən atributdur və bunun üçündə mail atributunu həmişə doldurmaq lazımdır. Həmçinin unutmayın ki, istifadəçi yaratdıqda o müəyyən bir standarta uyğun olaraq yaradılmalıdır. Bizim halda **sAMAccountName: username.surname** kimi göstərilməlidir.

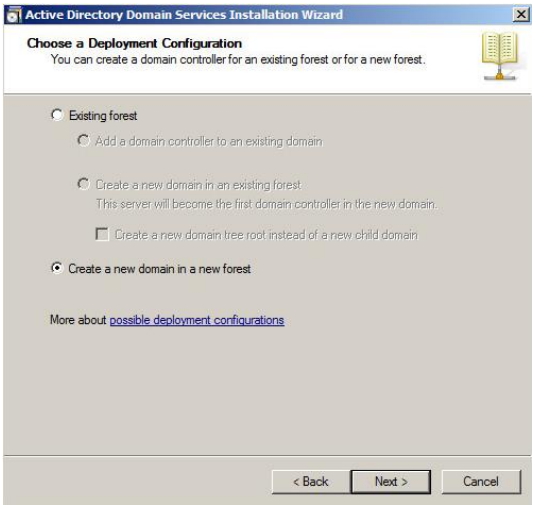
Start -> Run -> dcpromo # Əmri daxil edib Domain Controlleri quraşdıraq.



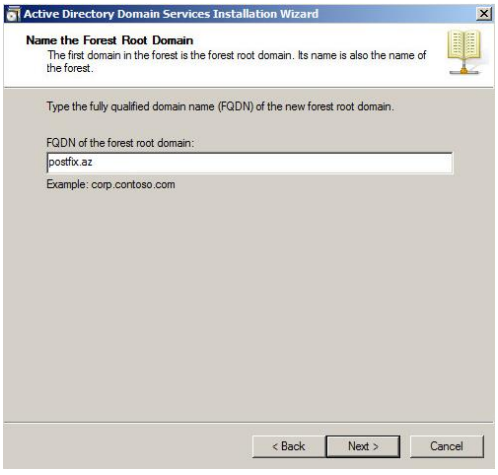
Şəkildə göstərildiyi kimi "**Use advanced Mode installation**" seçirik və iki dəfə ''**Next**" düyməsinə sıxırıq.



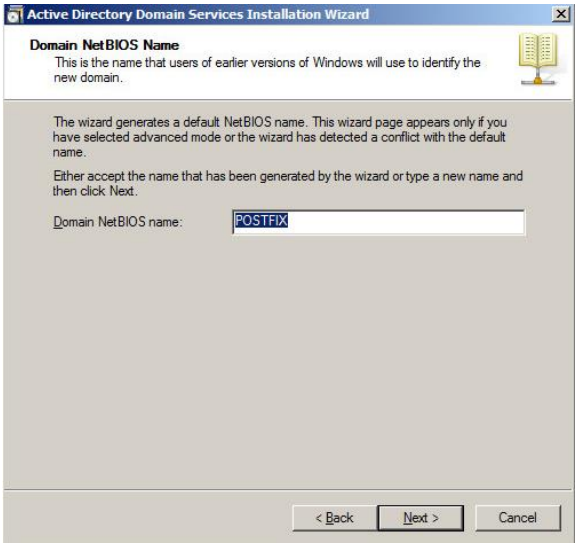
Və **'Create new Domain in the forest'** seçib **'Next'** edirik.



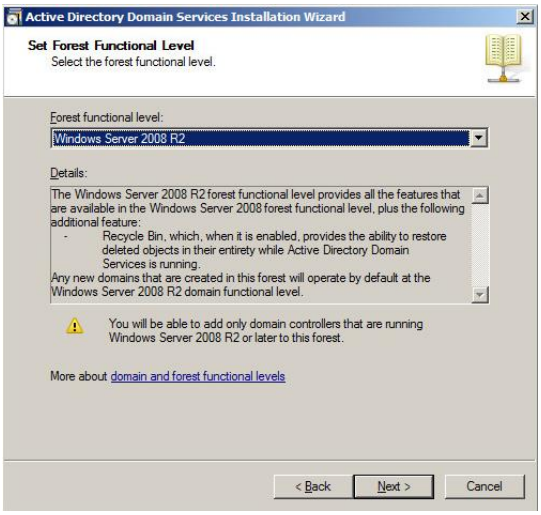
FQDN-də gördüyümüz kimi **'postfix.az'** yazıb "**NEXT**" düyməsinə sıxırıq. Ancaq siz istənilən ad verə bilərsiniz. Bu sizin Domain Controller adıdır.



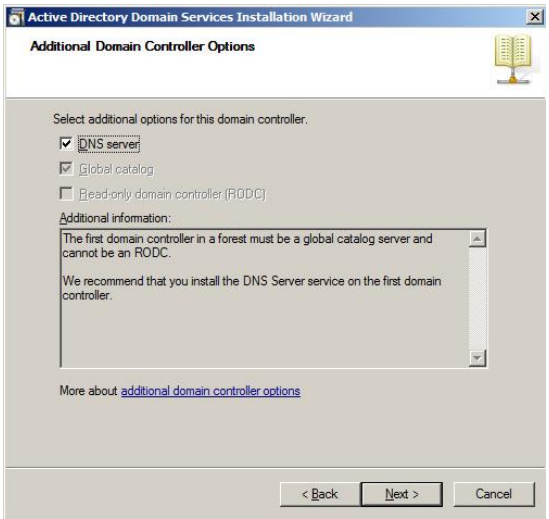
NetBIOS adı olduğu kimi saxlayıb "**NEXT**" düyməsinə sıxırıq.



Forest functionality Level-i yalnız serverimizin öz Release olan "**Windows 2008 server R2**" seçirik və "**NEXT**"

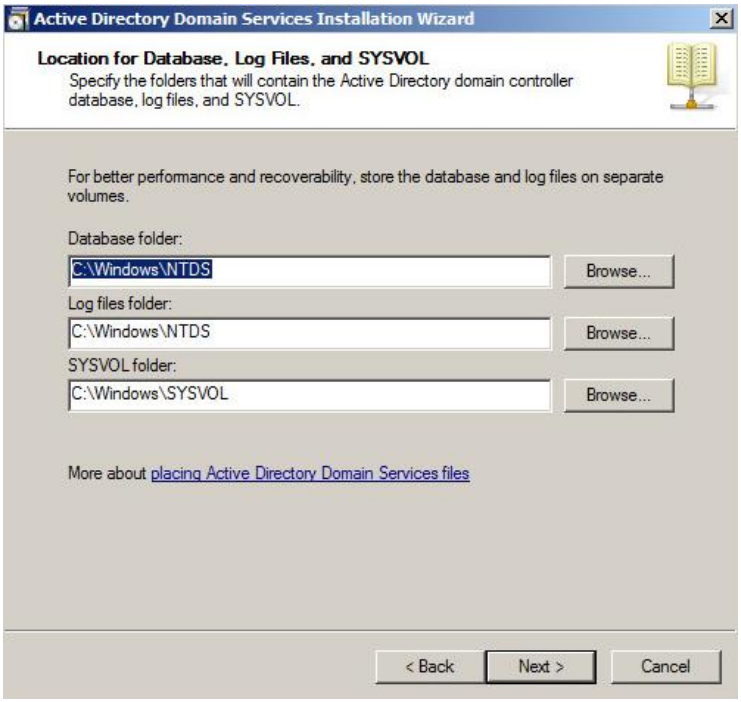


DNS serverin yüklənməsi üçün seçib, "**Next**" düyməsinə sıxırıq.

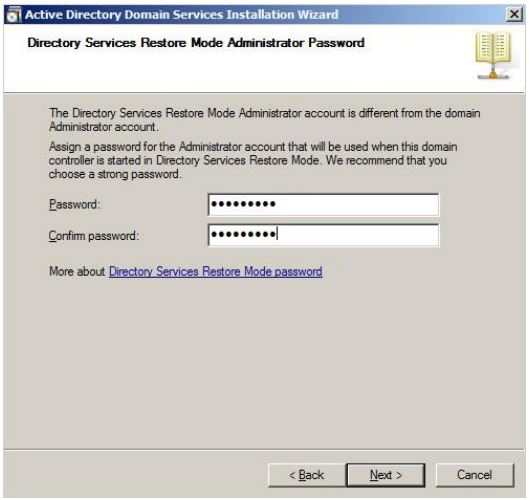


Çıxan mesaja fikir vermədən "**Yes**" düyməsinə sıxırıq.

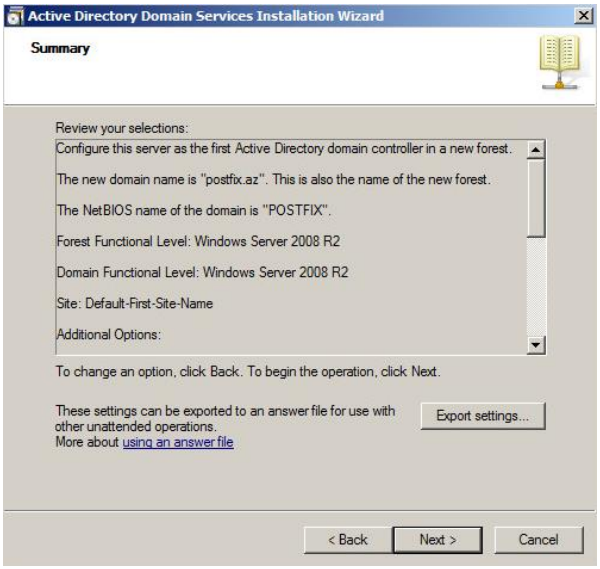
Baza, Log və **SysVol** ünvanının heç birinin ünvanını dəyişmədən "**Next**" istifadə edirik.



Admin şifrəsini itirildikdə **LDAP**-ı bərpa eləmək üçün şifrəni iki dəfə daxil edib "**Next**" düyməsinə sıxırıq.

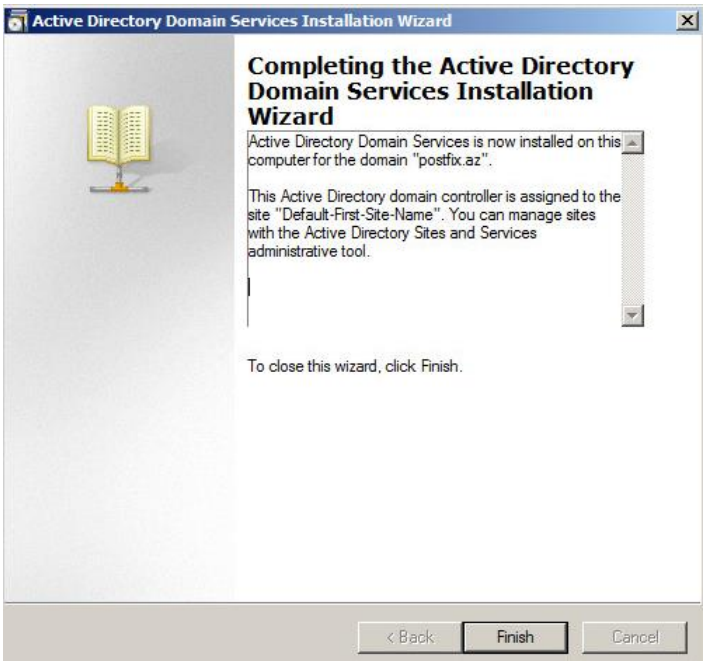


Sonda ümumi quraşdırmalarımızı nəzərdən keçirib "**Next**" düyməsinə sıxırıq.

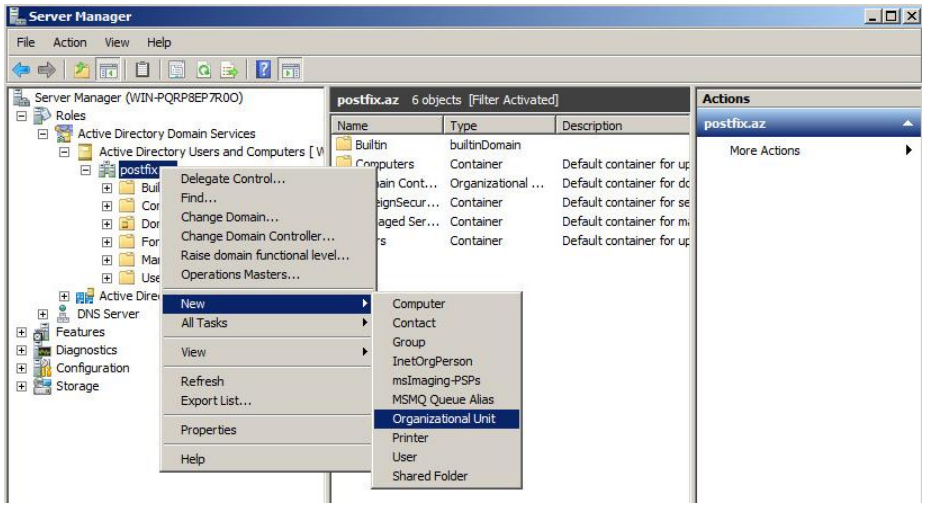


LDAP və **DNS** qurulmağa başlayır.

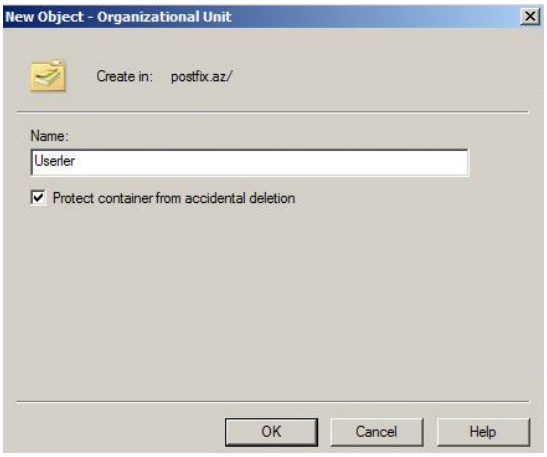
Sonda **Finish** və **"Restart now"** düyməsini sıxırıq.



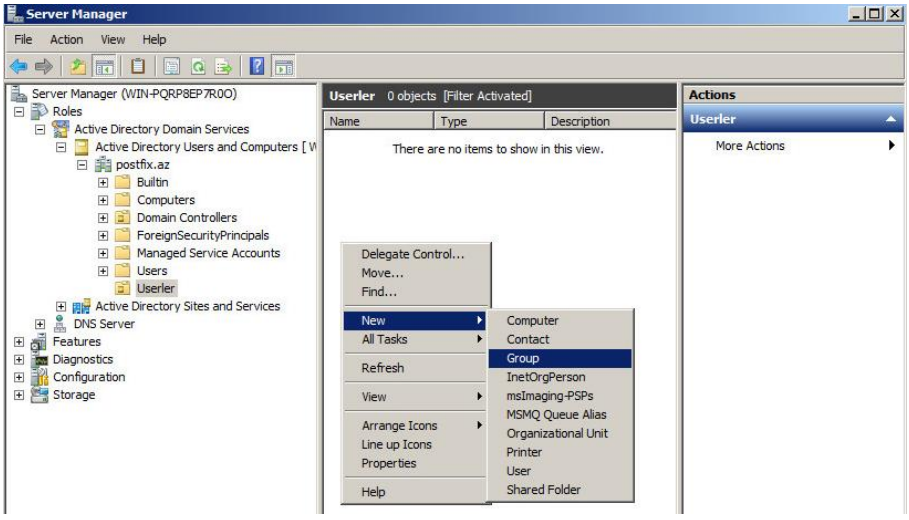
Yaratdığımız **AD**-nin içində yeni **Organizational UNIT** yaradaq.



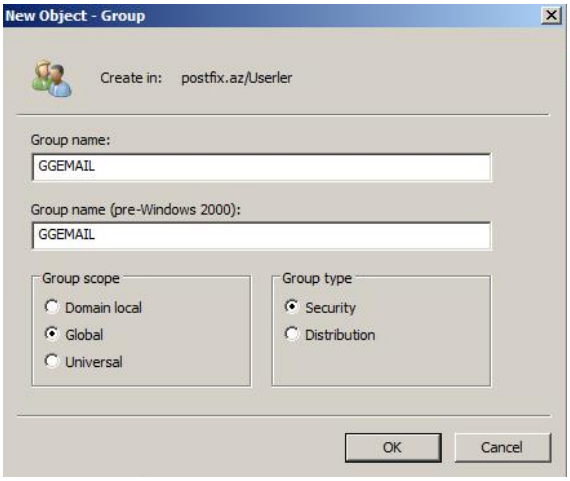
Adını "**Userler**" təyin edib "**OK**" düyməsinə sıxırıq.



Yeni "**Group**" əlavə edək.



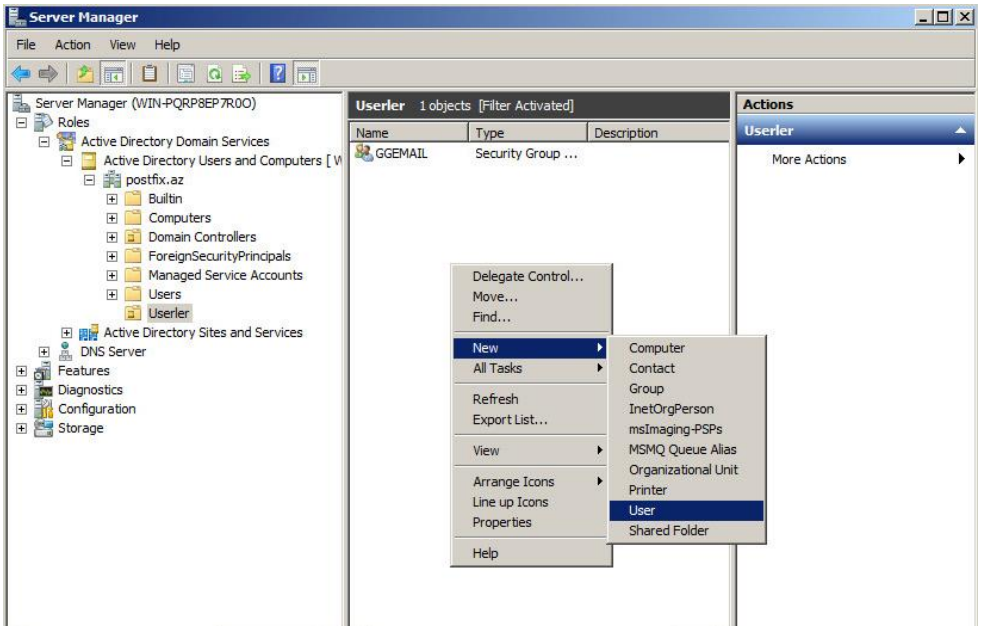
"**GGEMAIL**" adlı "**Global Security Gorup**" yaradaq və "**OK**" düyməsini sıxaq.



Təyin elədiyimiz müəyyən standarta əsasən istifadəçiləri əlavə edək və həmin istifadəçiləri "**GGEMAIL**" qrupuna əlavə edek. Əlavə edəcəyimiz istifadəçilər aşağıdakilar olacaq.

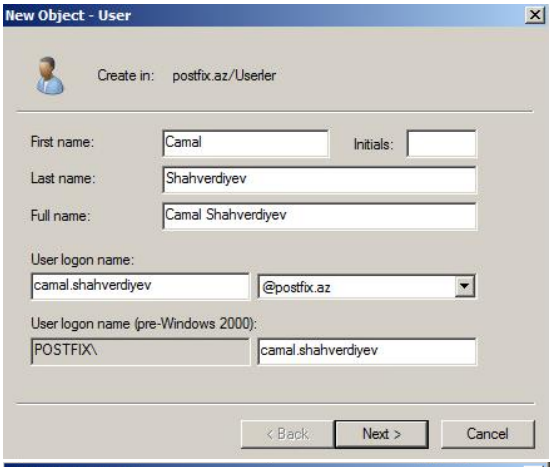
1. **Kamil Babayev**
2. **Salman Agayev**
3. **Ramil Mustafayev**
4. **Tural Nesirov**
5. **Namaz Bayramli**
6. **Mail Postmaster(postmaster@postfix.az – mail bildiriş üçün Mail Admin istifadəçi)**
7. **Camal Shahverdiyev(Domain, Enterprise Admin, Administrators qruplarının üzvü)**

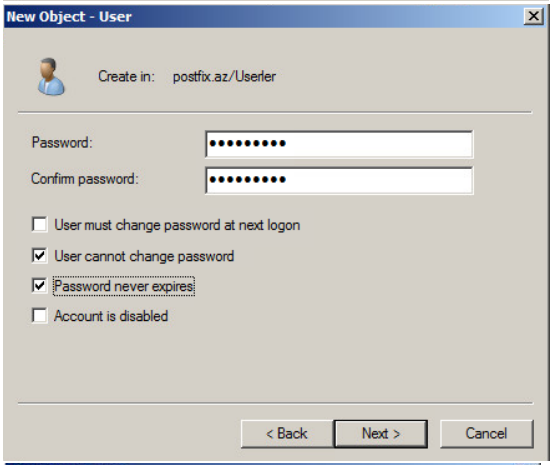
İlk olaraq '**Camal Shahverdiyev**' istifadəçisini yaradaq.

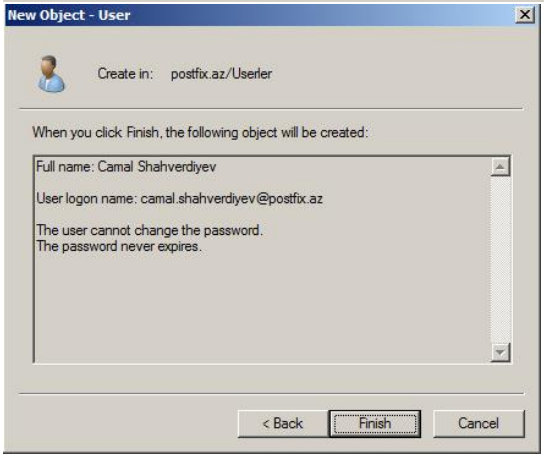


Aşağıdakı şəkildəki standarta uyğun olaraq bütün xanaları doldurub istifadəçiləri yaradırıq. Sadəcə hal-hazırda bizim misalda "**Camal**

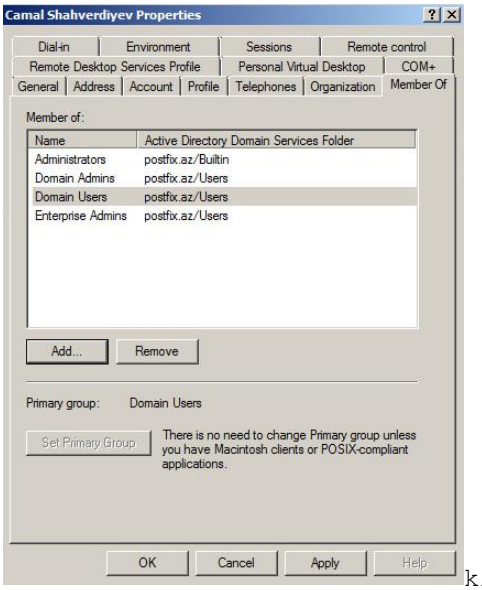
Shahverdiyev" istifadəçisini **Admin** kimi yaradırıq və **Admin** qruplarına əlavə edirik.





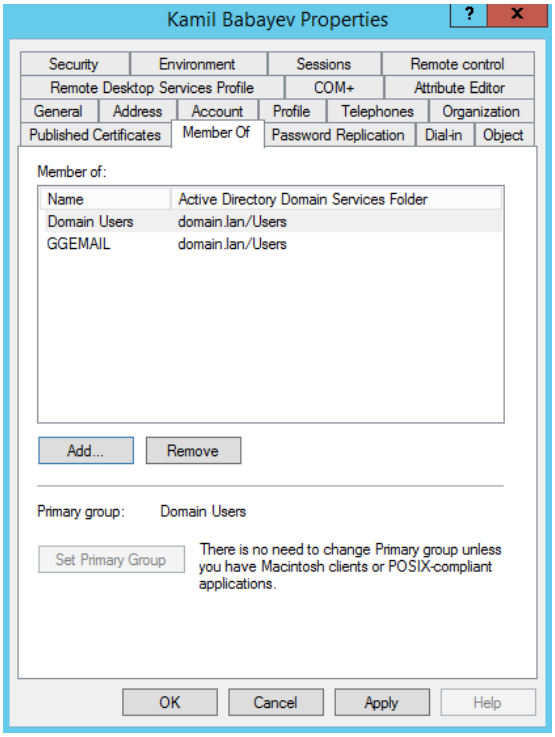


'camal.shahverdiyev' istifadəçisini "**Domain Admins**", "**Enterprise Admins**" ,"**Administrators**", "**Domain Users**", "**Group Policy Creator Owners**", "**GGEMAIL**" və "**Scheme Admins**" qruplarına əlavə edək. Əlavə eləmək üçün isə gördüyünüz **Add** düyməsindən istifadə edirsiniz.

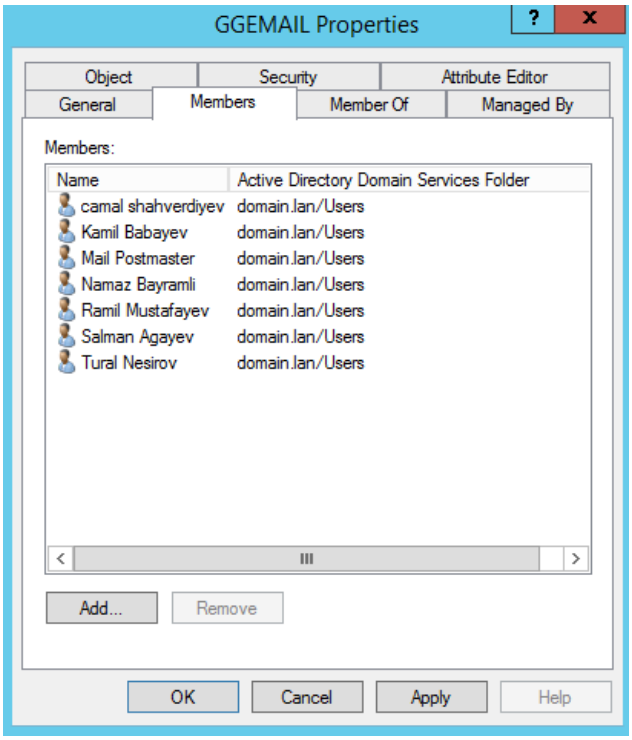


kl

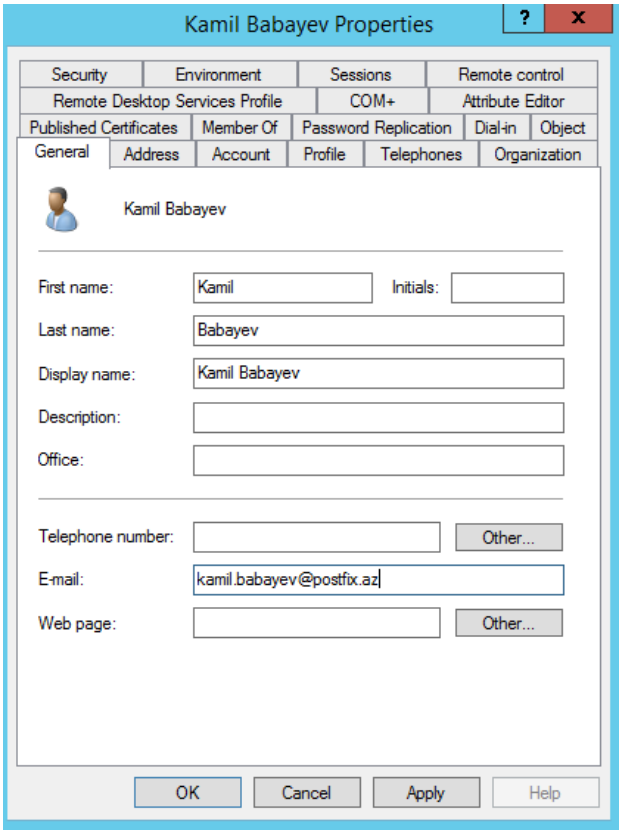
Digər istifadəçilər isə **"Domain Users"** və **"GGEMAIL"** qrupunun üzvü olmalıdır. Məhz **"GGEMAIL"** qrupu üzvlərinin email yeşikləri postfix-də yaradılacaq. Aşağıda sadəcə **'Kamil Babayev'** adlı istifadəçinin hansı qrupların üzvlüyündə olduğunu göstəririk.



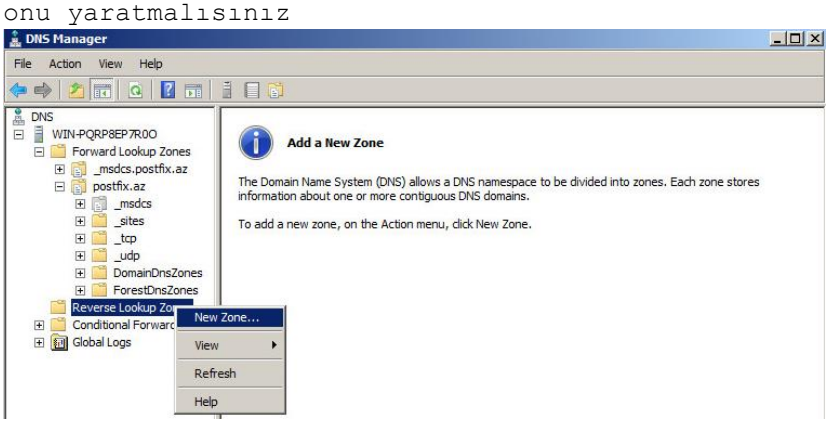
"GGEMAIL" qrupunun üzvlərini aşağıdaki şəkildə çap edirik.



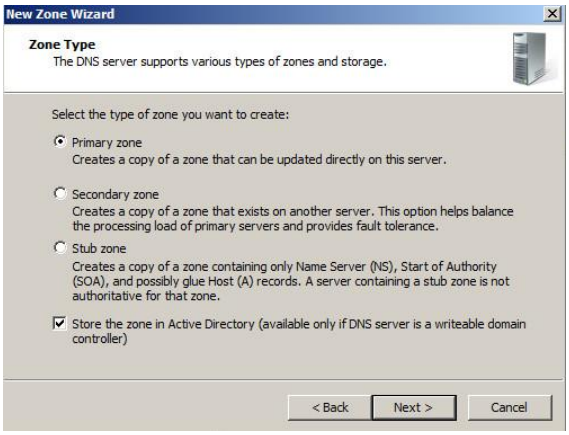
Unutmayın ki, istifadəçinin **Properties**-ində onun **email**-ı haqqında məlumat yazılmasa istifadəçilər **LDAP**-dan onun email-ı haqqında məlumat əldə etməyəcəklər. Şəkildə göstərildiyi kimi.



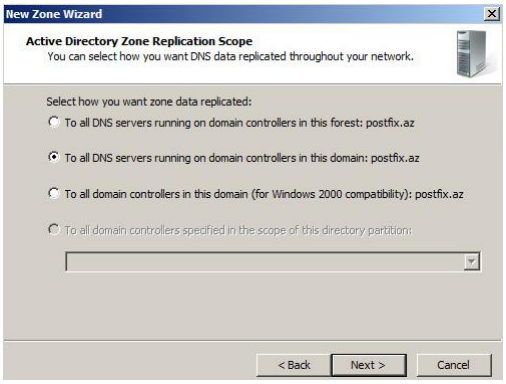
AD olan maşınımızda **FreeBSD** –**Postfix** maşın üçün **DNS**-də ad əlavə edək. (**A** və **MX** yazıları olacaq və **192.168.1.100 IP** ünvanına yönəldiləcək)
Start -> Run -> DNS -> Enter
Siz **A** yazısı üçün **PTR**-i yaratmaq istədikdə **xəta** çıxacaq ona görə ki, əsas **AD** adının **postfix.az**-ın özünün revers zonası hazırlanmayıb. Ona görə siz öncə onu yaratmalısınız



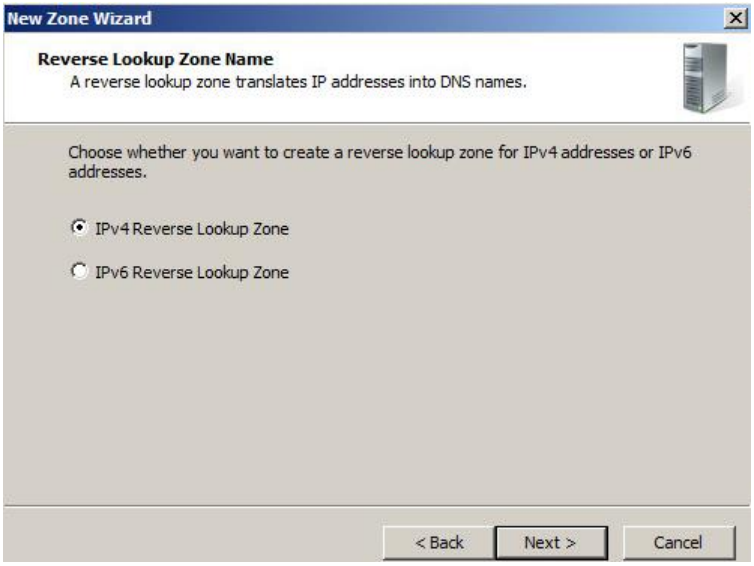
Next düyməsini sıxaraq "**Primary Zone**" seçirik və yenə də "**Next**" düyməsini sıxırıq.



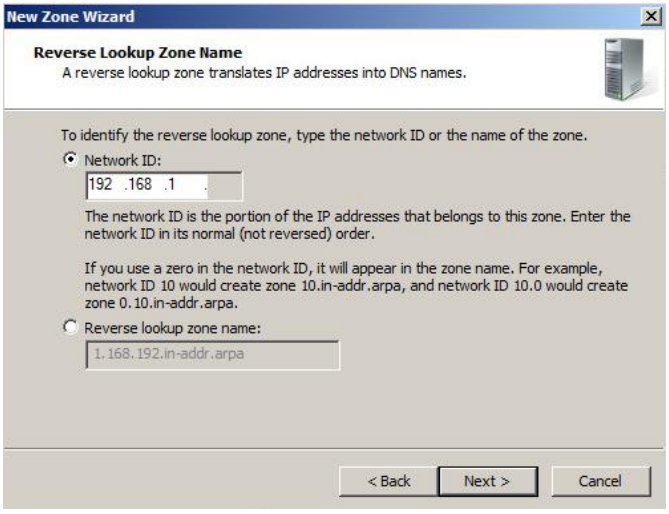
Data replicasiyasını susmaya görə saxlayıb "**Next**" düyməsini sıxırıq.



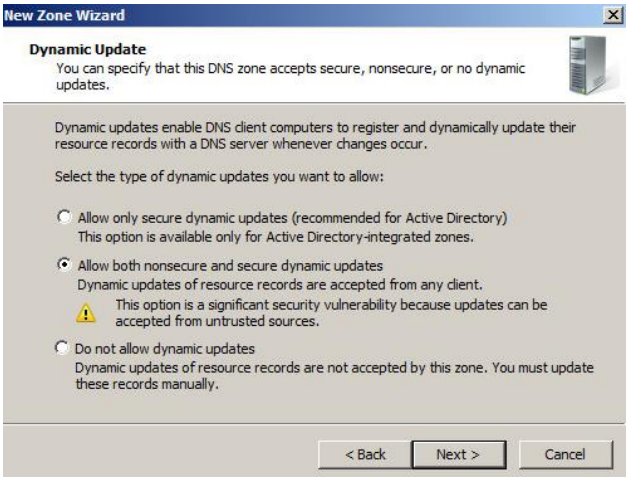
IPv4 LookUP zona seçirik və **Next** düyməsini sıxırıq.



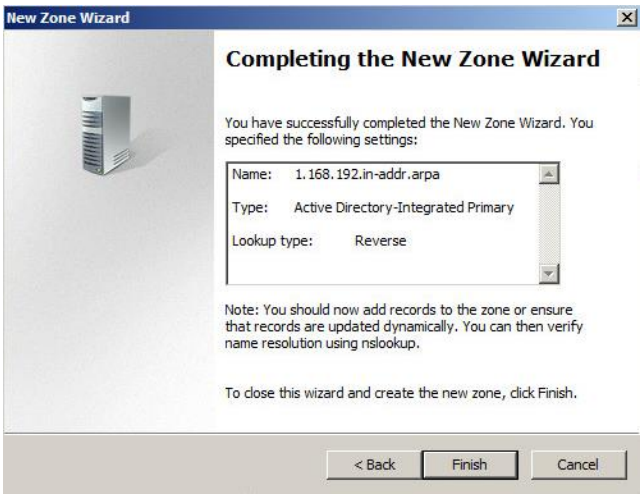
192.168.1 şəbəkəsi üçün **Revers Zona** təyin edirik və **Next**.



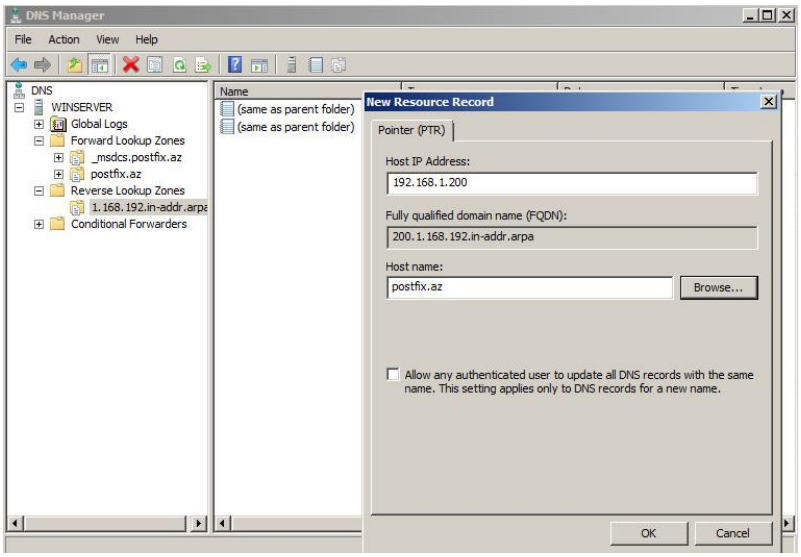
Hər kəsin **DNS**-dən Update dartmasına izin veririk və **Next** düyməsinə sıxırıq.



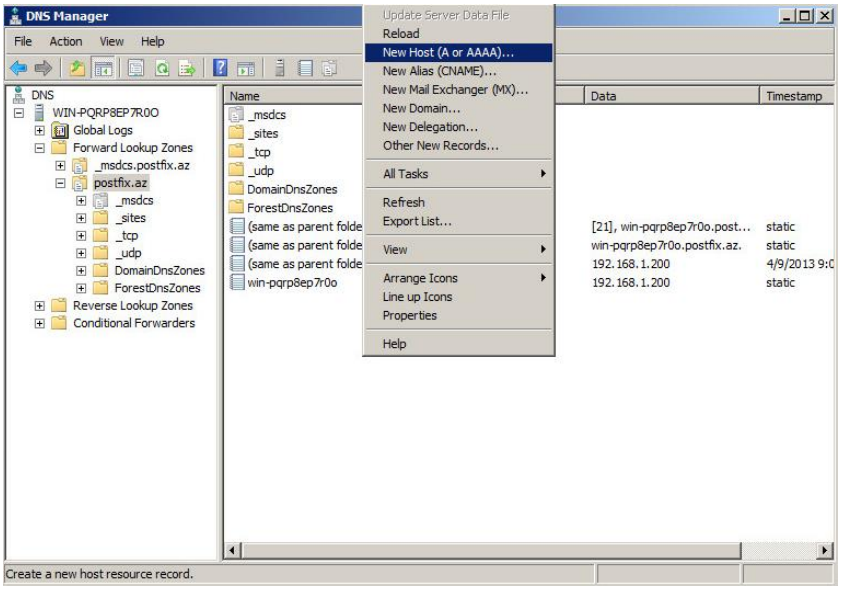
Sonda **Finish** düyməsinə sıxırıq.

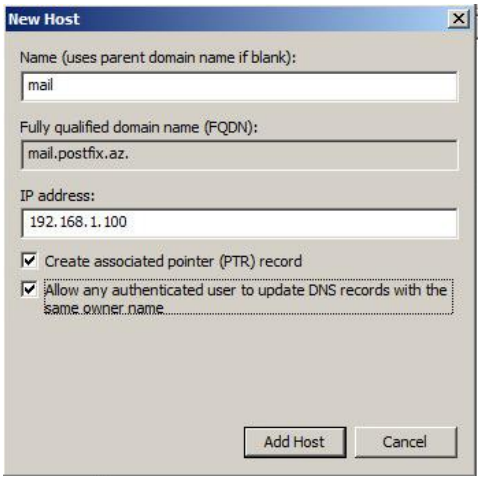


Öncə **Windows AD** maşının özü üçün **PTR** yaradırıq və eyni **Domain** adı(Yəni: **postfix.az**) veririk.Ardınca **OK** düyməsini sıxırıq.

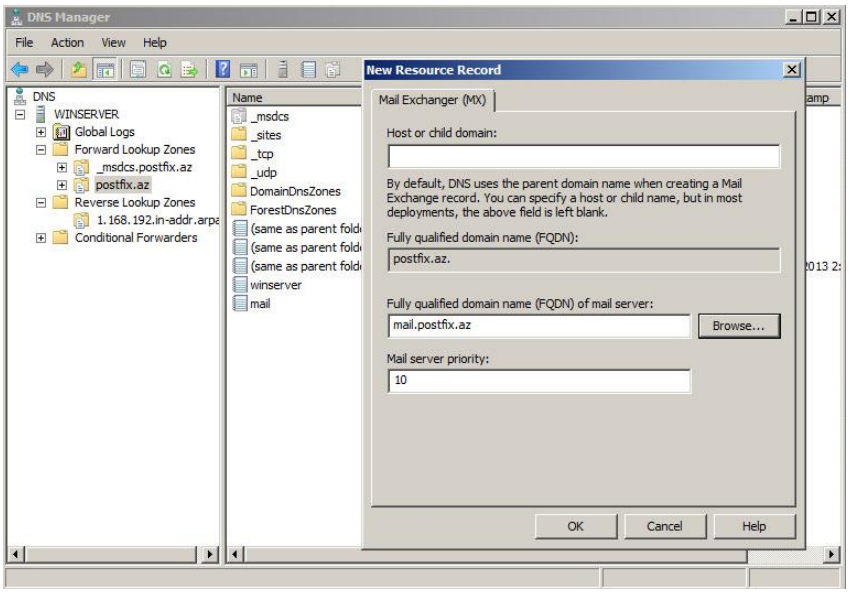


Sonra **FreeBSD** maşınımız üçün **'mail.postfix.az'** adını **192.168.1.100 IP** ünvanına **A** yazısı kimi əlavə edək.





Nəhayət sonda isə **mail.postfix.az A** yazısını eynilə **MX** kimi qeyd edirik və **OK.**



Windows **Domain Controllerimizdə DNS** quraşdırmaları bitdikdən sonra serverin özü üçün **Şəbəkə** kartında **DNS** kimi ilk **IP** ünvanı özünü yazırıq yəni **192.168.1.200**. Eynilə də bütün digər maşınlardada **DNS** kimi Windows **AD**-nin **DNS**-ni istifadə etməliyik.

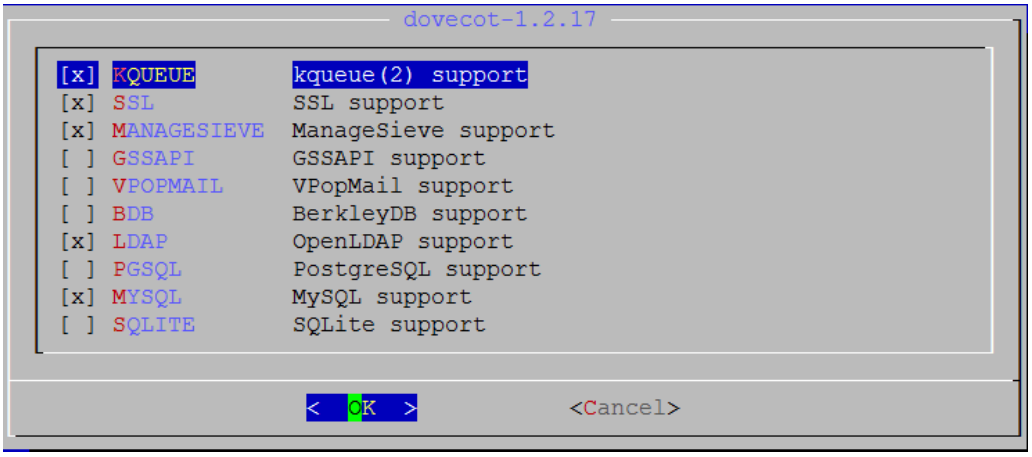
Windows7 clientləri isə **Computer name** dəyişib "**win7-1**" və "**win7-2**" etdikdən sonra "**POSTFIX**" **netbios** adı ilə **Domain**-ə qoşuruq.Unutmayın ki, **Windows7** clientləri **Domain**-ə qoşduqda **Domain admin** istifadəçisi kimi **'camal.shahverdiyev'** istifadə edirik(Sonda Sistemə Local istifadəçi yox, Domain İstifadəçisi kimi daxil olmağı unutmayın). Həmdə **Microsoft Outlook 2007**-ni yükləyirik ki, testlərimizi edə bilək.

Artıq FreeBSD maşınımızda Postfix və Dovecot birləşməsinin qurmasının vaxtı gəlib çatdı. (FreeBSD x64 9.1 – IP: 192.168.1.100)

```
portsnap fetch extract update        # Öncə portlarımızı yeniləyirik.(Sonda
                                     reboot edirik.)

cat /etc/rc.conf                     # Faylın Sonuna aşağıdaki sətirləri əlavə
                                     edərək lazımsız servisləri söndürürük.
#### Disabled Services          (Bunlardan sonra mütləq reboot elə)
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
sendmail_rebuild_aliases="NO"
syslogd_enable="YES"
syslogd_program="/usr/sbin/syslogd"
syslogd_flags="-ss"

# cat /etc/periodic.conf                    # Periodik işlərimizdə
                                            aşağıdakiları söndürürük.
```

```
daily_clean_hoststat_enable="NO"
daily_status_mail_rejects_enable="NO"
daily_status_include_submit_mailq="NO"
daily_submit_queuerun="NO
```

```
# pw group add vmail -g 1000                    # Sistemə Mail üçün 'vmail' adlı
                                                qrup əlavə edirik.
# pw user add vmail -u 1000 -g 1000 -d /dev/null -s /sbin/nologin # vmail
                                                istifadəçini yaradıb vmail qrupuna
                                                əlavə edirik.

# mkdir /var/virtual                            # İstifadəçilərin Mail-ləri
                                                yarananda, bu qovluqda yaranacaq.

# chown -R vmail:vmail /var/virtual             # Bu qovluğu vmail user və qrupun
                                                üzvü edirik.

# chmod -R 700 /var/virtual                     # vmail istifadəçilərinə bu qovluq
                                                üçün tam yetki veririk.

# cd /usr/ports/mail/dovecot                    # POP/S və IMAP/S istifadə edə
                                                bilməyimiz üçün dovecot-1.2.17-ni
                                                yükləyirik.
# make config                                   # Aşağıdakı şəkildə olan modulları
                                                seçirik. Qalan depends-lərdə isə
                                                IPv6 və SQLITE seçmirik və  başqa
                                                hər şeyi susmaya görə seçirik
```




```
# make install clean                                    # Yükləyirik.

# cd /usr/ports/mail/dovecot-sieve                      # Bu paket gələn Mail Spam ilə
                                                        təyin edilərsə email yeşiyinə
                                                        düşməzdən əvvəl onun filteri ilə
                                                        məşğul olacaq.
# make install clean                                    # Yükləyirik.

# echo 'dovecot_enable="YES"' >> /etc/rc.conf           # Dovecot-u Startup-a
                                                        əlavə edirik ki,
                                                        reboot-dan sonra
                                                        işləsin.
```

Dovecot-un **IMAP/S** və **POP/S** üçün sertifikatlarını hazırlayaq.

```
#mkdir /etc/ssl/dovecot                # Sertifikatlar üçün qovluq yaradaq
#cd /etc/ssl/dovecot                   # Qovluğa daxil olaq
#openssl req -new -x509 -nodes -out cert.pem -keyout key.pem -days 365  #
                                                                  Sertifikatımızı
                                                                  yaradaq.
                                                                  Verilənləri
                                                                  aşağıdaki formada
                                                                  əlavə edirik.
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:Baki
Locality Name (eg, city) []:Yasamal
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Azersu
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:mail.postfix.az
Email Address []:postmaster@postfix.az

# cat /usr/local/etc/dovecot.conf                        # Dovecot Config
                                                         faylımızın tərkibi belə
                                                         olmalıdır.

protocols = imap imaps pop3 pop3s
disable_plaintext_auth = no
log_path = /var/log/dovecot.log                  # Jurnal faylımızın ünvanı
info_log_path = /var/log/dovecot.log             # Info jurnal faylımızın ünvanı
auth_debug = yes
auth_debug_passwords = yes
auth_verbose = yes
ssl = yes
ssl_cert_file = /etc/ssl/dovecot/cert.pem
ssl_key_file = /etc/ssl/dovecot/key.pem
login_greeting = Camal's Mail Server Ready.
mail_location = maildir:/var/virtual/%n/Maildir
mail_uid = vmail
mail_gid = vmail
mail_privileged_group = mail
first_valid_uid = 1000
last_valid_uid = 1000
first_valid_gid = 1000
last_valid_gid = 1000
valid_chroot_dirs = /var/virtual
maildir_copy_with_hardlinks = yes
protocol imap {
  mail_plugins = quota imap_quota
  mail_plugin_dir = /usr/local/lib/dovecot/imap
  imap_client_workarounds = delay-newmail netscape-eoh tb-extra-mailbox-sep
}
protocol pop3 {
  pop3_uidl_format = %08Xu%08Xv
  mail_plugins = quota
  mail_plugin_dir = /usr/local/lib/dovecot/pop3
  pop3_client_workarounds = outlook-no-nuls oe-ns-eoh
```

```
}
protocol lda {
  debug = yes
#  mail_plugins = cmusieve quota
  mail_plugins = sieve quota
  mail_plugin_dir = /usr/local/lib/dovecot/lda
  postmaster_address = postmaster@postfix.az
  sendmail_path = /usr/sbin/sendmail
  auth_socket_path = /var/run/dovecot/auth-master
  log_path = /var/log/dovecot-lda.log
  info_log_path = /var/log/dovecot-lda.log
  global_script_path = /usr/local/etc/dovecot/dovecot.sieve
  sieve_global_path = /usr/local/etc/dovecot/dovecot.sieve
}
auth_username_format = %Lu
auth default {
  mechanisms = plain login
  passdb ldap {
       args = /usr/local/etc/dovecot-ldap.conf
  }
  userdb ldap {
       args = /usr/local/etc/dovecot-ldap.conf
  }
  user = root
  socket listen {
    master {
      path = /var/run/dovecot/auth-master
      mode = 0600
      user = vmail
      group = vmail
    }
    client {
      path = /var/run/dovecot/auth-client
      mode = 0660
      user = postfix
      group = postfix
    }
  }
}
dict {
}
plugin {
  quota_rule = *:storage=102400
  quota = maildir
  quota_warning = storage=85%% /usr/local/bin/quota-warning.sh 85%
      # İstifadəçiyə çıxacaq Quota warning
  sieve  = /usr/local/etc/dovecot/dovecot.sieve
}

# touch /var/log/dovecot.log /var/log/dovecot-lda.log # Jurnal fayllarımızı
                                                      yaradırıq.
```

```
# chown vmail /var/log/dovecot*                    # Jurnal fayllarımızın
                                                   sahibini 'vmail' adlı
                                                   istifadəçini təyin
                                                   edirik.
```

Aşağıdaki əmrlə siz LDAP-ı test edə bilərsiniz.

```
# Əmri daxil etdikdən sonra camal.shahverdiyev istifadəçisi üçün parol
yığmanız yetər
ldapsearch -x -b "dc=postfix,dc=az" -D "camal.shahverdiyev@postfix.az" -h
postfix.az -W
```

Sieve Scripti SpamAssasindən Spam kimi alınan məktubları istifadəçilərin **INBOX.Spam** qovluğuna ötürəcək. Script qlobaldır və bütün istifadəçilər üçün istifadə ediləcək.

```
#mkdir /usr/local/etc/dovecot                    # Dovecot Sieve Scripti üçün qovluq
                                                   yaradaq.

#touch /usr/local/etc/dovecot/dovecot.sieve      # 'dovecot.sieve' script faylını
                                                   yaradaq.

#chown -R vmail /usr/local/etc/dovecot           # Yaratdığımız qovluq 'vmail'
                                                   istifadəçisinin üzvü edək.

# cat dovecot.sieve                              # Faylın məzmununa aşağıdaki
                                                   sətirləri əlavə edirik.
####################
#
require ["fileinto"];
if header :contains "X-Spam-Level" "**********" {
        discard;
        stop;
}
elsif
header :contains "X-Spam-Status" "Yes" {
        fileinto "INBOX.Spam";
        stop;
}
#
####################
```

Bütün istifadəçilərin email yeşikləri 20MB(20480) həcmində olacaq. Əgər bu həcm 85%-ə çatsa həmin istifadəçilərə email yollanacaq. Siz bunu özünüzə uyğun olaraq quraşdıra bilərsiniz. Indi isə gəlin onun scriptini hazırlayaq.

```
# touch /usr/local/bin/quota-warning.sh            # Istifadəçi üçün
                                                   Warning Scriptimizi
                                                   yaradaq.

# chown vmail /usr/local/bin/quota-warning.sh      # Scripti 'vmail'
                                                   istifadəçinin üzvü
                                                   edirik.
```

```
# cat /usr/local/bin/quota-warning.sh                # Scriptimizin tərkibi
                                                     aşağıdaki kimi olacaq.

####################
#!/bin/sh

PERCENT=$1
FROM=" postmaster@postfix.az"
qwf="/tmp/quota.warning.$$"

echo "From: $FROM
To: $USER
To: postmaster@postfix.az
Subject: Sizin e-mail yeşiyiniz $PERCENT% istifadə edilir.
Content-Type: text/plain; charset="UTF-8"

Xəbərdarlıq: Sizin e-mail yeşiyiniz $PERCENT% istifadə edilir." >> $qwf

cat $qwf | /usr/sbin/sendmail -f $FROM "$USER"
rm -f $qwf
exit 0
####################
```

Dovecot LDAP-ı quraşdıraq.

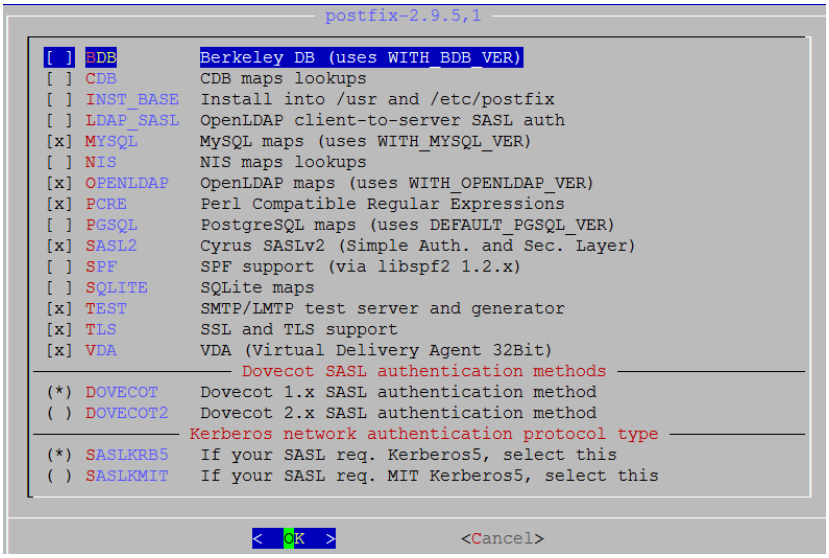
```
# cat /usr/local/etc/dovecot-ldap.conf          # Faylımızın tərkibi
                                                aşağıdaki kimi olacaq.
####################
debug_level = 0
hosts = 192.168.1.200:3268                  # Domain Controllerin IP və LDAP
portu(alternative portu)
base = dc=postfix,dc=az
ldap_version = 3
scope = subtree
deref = searching
dn = CN=Camal Shahverdiyev,OU=Userler,DC=postfix,DC=az
dnpass = Zuzubala
auth_bind = yes
user_filter =
(&(ObjectClass=person)(sAMAccountName=%u)(memberOf=CN=GGEMAIL,OU=Userler,DC=p
ostfix,DC=az))
pass_filter =
(&(ObjectClass=person)(sAMAccountName=%u)(memberOf=CN=GGEMAIL,OU=Userler,DC=p
ostfix,DC=az))
####################
```

Qeyd: Unutmayın **Dovecot-u start** etmək istəyəndə o hələki **qalxmayacaq** çünki,

sistemdə **'postfix'** adlı istifadəçi və qrup yoxdur. Ona görə də öncə onu yükləmək və sonra işə salmaq lazımdır.

Postfix-in yüklənməsi. O bizim MTA(Mail Transfer Agent) rolunda işləyəcək.

```
# cd /usr/ports/mail/postfix                    # Postfix default olaraq portlarda
                                                2.9.5 idi. Lazımi modulları seçək.
# make config                                   # lazımi modulları seçirik.
                                                Dependslərin hamısında SQLITE və
                                                IPv6-dan başqa hər şeyi seçirik.
```




```
# make install                                  # Yukleyirik.
Would you like to activate Postfix in /etc/mail/mailer.conf [n]? y
      # Suala 'yes' cavabı veririk.

# cat /etc/passwd | grep postfix                # Postfix adlı istifadəçi
                                                yaranmasını yoxlayırıq.
postfix:*:125:125:Postfix Mail System:/var/spool/postfix:/usr/sbin/nologin

# cat /etc/group | grep postfix                 # Postfix adlı qrup
                                                yaranmasını yoxlayırıq.
postfix:*:125:

# /usr/local/etc/rc.d/dovecot start             # Dovecot-u işə salırıq.
# cat /var/log/dovecot.log                      # Jurnal faylında işləməsini
                                                yoxlayırıq. Aşağıdakı
                                                sətirlər oxşar sətirlər
                                                olmalıdır.
Apr 10 12:11:10 dovecot: Info: Dovecot v1.2.17 starting up
Apr 10 12:11:11 auth(default): Info: new auth connection: pid=29668
Apr 10 12:11:11 auth(default): Info: new auth connection: pid=29669
Apr 10 12:11:11 auth(default): Info: new auth connection: pid=29671
Apr 10 12:11:11 auth(default): Info: new auth connection: pid=29672
Apr 10 12:11:11 auth(default): Info: new auth connection: pid=29670
Apr 10 12:11:11 auth(default): Info: new auth connection: pid=29673
```

Dovecot-u test edək.

```
# telnet localhost 143              # IMAP Serverimizin portuna qoşuluruq.
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE
STARTTLS AUTH=PLAIN AUTH=LOGIN] C
amal's Mail Server Ready.
a login camal.shahverdiyev Zuzubala # camal.shahverdiyev istifadəçisi və
Zuzubala şifrəsi ilə qoşuluruq
a OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE SORT
SORT=DISPLAY THREAD=REFERENC
ES THREAD=REFS MULTIAPPEND UNSELECT IDLE CHILDREN NAMESPACE UIDPLUS LIST-
EXTENDED I18NLEVEL=1 CONDSTORE
QRESYNC ESEARCH ESORT SEARCHRES WITHIN CONTEXT=SEARCH LIST-STATUS QUOTA]
Logged in
a EXAMINE INBOX                 # INBOX qovluğumuzu yoxlayırıq
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS ()] Read-only mailbox.
* 0 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1365578088] UIDs valid
* OK [UIDNEXT 1] Predicted next UID
* OK [HIGHESTMODSEQ 1] Highest
a OK [READ-ONLY] Select completed.
a LOGOUT                        # Və çıxırıq.
* BYE Logging out
a OK Logout completed.
Connection closed by foreign host.

# ls -la /var/virtual/          # İstifadəçinin qovluq yaranmasına baxırıq.
drwx------  3 vmail  vmail  512 Apr 10 12:14 camal.shahverdiyev/
```

Indi isə Postfix üçün SSL /TLS sertifikatlarını yaradaq

```
# mkdir  /etc/ssl/postfix       # Postfix üçün sertifikat qovluğunu yaradaq
# cd /etc/ssl/postfix           # İçinə daxil olaq
# openssl req -new -x509 -nodes -out smtpd.pem -keyout smtpd.pem -days 3650
      # 10 illik sertifikat yaradaq
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:Baku
Locality Name (eg, city) []:Yasamal
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Azersu
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:mail.postfix.az
Email Address []:postmaster@postfix.az

# chmod 640 /etc/ssl/postfix/smtpd.pem     # Yetkini azaldaq
# chgrp -R postfix /etc/ssl/postfix        # Qovluğun qrup üzvlüyünü
                                           'postfix'-ə verək.
```

```
# cd /usr/local/etc/postfix/                # Postfix-in qovluğuna daxil olaq.

# cat /usr/local/etc/postfix/main.cf        # Quraşdırma faylının tərkibini
                                            aşağıdaki kimi edirik.
####################
## Global config
queue_directory = /var/spool/postfix
command_directory = /usr/local/sbin
daemon_directory = /usr/local/libexec/postfix
mail_owner = postfix
myhostname = mail.postfix.az
mydomain = postfix.az
myorigin = $mydomain
mydestination = $myhostname, localhost.$mydomain, localhost
unknown_local_recipient_reject_code = 550
mynetworks_style = host
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
debug_peer_level = 3
debugger_command =
     PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
     xxgdb $daemon_directory/$process_name $process_id & sleep 5
sendmail_path = /usr/local/sbin/sendmail
newaliases_path = /usr/local/bin/newaliases
mailq_path = /usr/local/bin/mailq
setgid_group = maildrop
html_directory = no
manpage_directory = /usr/local/man
sample_directory = /usr/local/etc/postfix
readme_directory = no

## Antivirus Filter edilməsi(Aşağıdaki sətirin Kommentini Amavis-new yüklənib
quraşdırandan sonra silmək lazımdır)
#content_filter=smtp-amavis:[localhost]:10024

## SASL-ın quraşdırılması
broken_sasl_auth_clients = yes
smtpd_sender_restrictions = permit_sasl_authenticated, permit_mynetworks
smtpd_recipient_restrictions =
  permit_mynetworks,
  permit_sasl_authenticated,
  reject_non_fqdn_hostname,
  reject_non_fqdn_sender,
  reject_non_fqdn_recipient,
  reject_unauth_destination,
  reject_unauth_pipelining,
  reject_invalid_hostname,
  reject_rbl_client list.dsbl.org,
  reject_rbl_client bl.spamcop.net,
  reject_rbl_client sbl-xbl.spamhaus.org
smtpd_sasl_auth_enable = yes
smtpd_sasl_authenticated_header = yes
smtpd_sasl_local_domain = $myhostname
smtpd_sasl_security_options = noanonymous
```

```
smtpd_sasl_type = dovecot
smtpd_sasl_path = /var/run/dovecot/auth-client

## TLS/SSL-in quraşdırılması
smtp_use_tls = yes
smtpd_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /etc/ssl/postfix/smtpd.pem
smtpd_tls_cert_file = /etc/ssl/postfix/smtpd.pem
smtpd_tls_CAfile = /etc/ssl/postfix/smtpd.pem
smtpd_tls_loglevel = 0
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom

## LDAP/AD-nin quraşdırılması
home_mailbox = Maildir/
virtual_mailbox_base = /var/virtual
virtual_uid_maps = static:1000
virtual_gid_maps = static:1000
smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination
alias_maps = hash:/etc/aliases
command_directory = /usr/local/sbin
daemon_directory = /usr/local/libexec/postfix
virtual_mailbox_domains = POSTFIX.AZ
virtual_mailbox_maps = ldap:ldapvirtual
ldapvirtual_server_host = ldap://192.168.1.200:3268
ldapvirtual_search_base = dc=postfix,dc=az
ldapvirtual_bind = yes
ldapvirtual_bind_dn = POSTFIX\camal.shahverdiyev
ldapvirtual_bind_pw = Zuzubala
ldapvirtual_query_filter = (sAMAccountName=%u)
ldapvirtual_result_attribute = sAMAccountName
ldapvirtual_version = 3
ldapvirtual_chase_referrals = yes
ldapvirtual_result_format=%s/Maildir/

## Dovecot LDA Agent Delivery
virtual_transport= dovecot
dovecot_destination_recipient_limit=1
####################

Postfix-in master.cf faylında lazımi dəyişiklikləri edək.
# cat /usr/local/etc/postfix/master.cf                # master.cf faylında
                                                      SMTPS-i aşağıdaki
                                                      formada quraşdırırıq
smtps     inet  n       -       n       -       -       smtpd
#  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
#  -o milter_macro_daemon_name=ORIGINATING
```

Və **'/usr/local/etc/postfix/master.cf'** faylının sonuna aşağıdaki sətirləri əlavə edək. Unutmayın ki, bu sətirdən sonra Postfix-i yalnız SmapAssasin yüklənib quraşdırıldıqdan sonra start edib test edə bilərsiniz.

```
dovecot   unix  -       n       n       -       -       pipe
   flags=DRhu user=vmail:vmail argv=/usr/local/bin/spamc -u ${user} -e
/usr/local/libexec/dovecot/deliver -d ${user}
```

Postfix-in alias bazasını yaradaq.

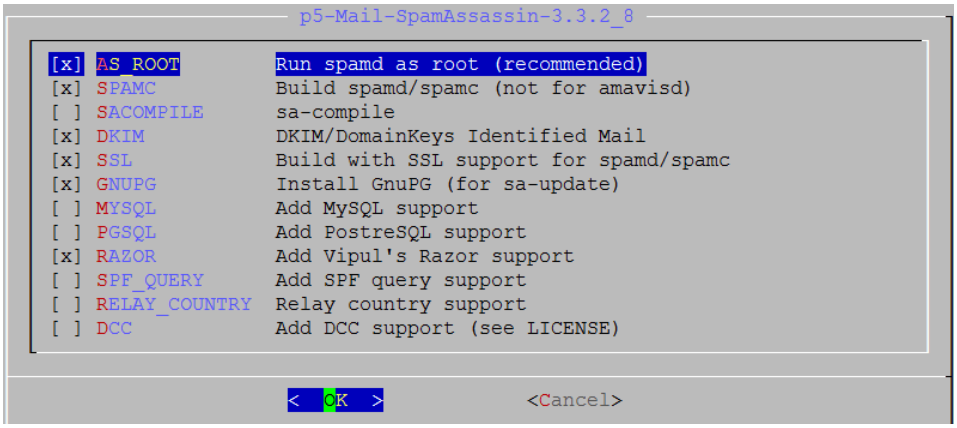
```
# mv /etc/aliases /etc/aliases.OFF
# ln -s /usr/local/etc/postfix/aliases /etc/aliases
# touch /usr/local/etc/postfix/aliases
# postalias /usr/local/etc/postfix/aliases
```

Domain Controller istifadəçiləriniz üçün şifrə generasiya eləmək üçün aşağıdaki sintaksisdən istifadə edə bilərsiniz.

```
# printf '\0Userler\0camal.shahverdiyev' | mmencode
```

AntiSpam quraşdıraq.

```
# cd /usr/ports/mail/p5-Mail-SpamAssassin          # Hal-hazırda 3.3.2-ci versiya istifadə edilir.
# make config                                      # şəkildəki Depends-ləri seçirik.IPv6 və SQLITE-dan başqa bütün modulları susmaya görə seçirik.
```




```
# make install clean                               # Yükləyirik.
Do you wish to run sa-update to fetch new rules [N]? Y      # Suala Yes cavabı veririk.
```

SpamAssassin-i startupa əlavə edirik.

```
# echo 'spamd_enable="YES"' >> /etc/rc.conf
# echo 'spamd_flags="-u spamd -H /var/spool/spamd"' >> /etc/rc.conf
# cd /usr/local/etc/mail/spamassassin              # local.cf faylını quraşdıraq.
```

```
# cat local.cf                                    # Faylın tərkibi aşağıdaki
                                                  kimidir.
rewrite_header Subject *****SPAM*****
use_bayes 1
ifplugin Mail::SpamAssassin::Plugin::Shortcircuit


# /usr/local/etc/rc.d/sa-spamd start      # SpamAssassin-i işə salırıq.

# echo 'postfix_enable="YES"' >> /etc/rc.conf    # Postfix servisi Startup-a
                                                  əlavə edirik.

# /usr/local/etc/rc.d/postfix start              # Postfix-i işə salırıq.
```

Postfix-i test edək.

```
# telnet localhost 25          # Postfix-in portuna qoşulaq. Tünd qara
                               simvollar əmrlərdir.
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.postfix.az ESMTP Postfix (2.9.5)
helo localhost
250 mail.postfix.az
mail from: camal.shahverdiyev@postfix.az  # İstifadəçidən
250 2.1.0 Ok
rcpt to: kamil.babayev@postfix.az         # İstifadəçiyə göndəririk
250 2.1.5 Ok
Data
354 End data with <CR><LF>.<CR><LF>
Salam Necesen kamil?                      # Mesaj
.
250 2.0.0 Ok: queued as D3230112A85
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

Və Mail yeşikləri olan qovluğa baxırıq ki, **kamil.babayev** adlı istifadəçi üçün yeşik yaranıb.

```
# ll /var/virtual/                    # Kamil üçün yeşik yaranıb.
drwx------  3 vmail  vmail  512 Apr 10 12:14 camal.shahverdiyev/
drwx------  3 vmail  vmail  512 Apr 10 13:35 kamil.babayev/
```

SpamAssassin-i test edək.

Aşağıdaki sətiri hansısa istifadəçi adından kiməsə yollayın.

```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

Əgər siz düzgün quraşdırmısınızsa onda **'/var/log/maillog'** və **'/var/log/dovecot-lda.log'** fayllarında bunun sübütlarını görə bilərsiniz.

```
# cat /var/log/maillog | grep "identified spam"
```

```
Apr 10 14:09:30 postfix-ldap spamd[733]: spamd: identified spam (1002.0/5.0)
for camal.shahverdiyev:58 in 32.8 seconds, 468 bytes.

# cat /var/log/dovecot-lda.log | grep "marked message to be discarded"
Apr 10 14:09:30 deliver(camal.shahverdiyev): Info: sieve:
msgid=<20130410090847.7F30F112AA2@mail.postfix.az>: marked message to be
discarded if not explicitly delivered (discard action)
```

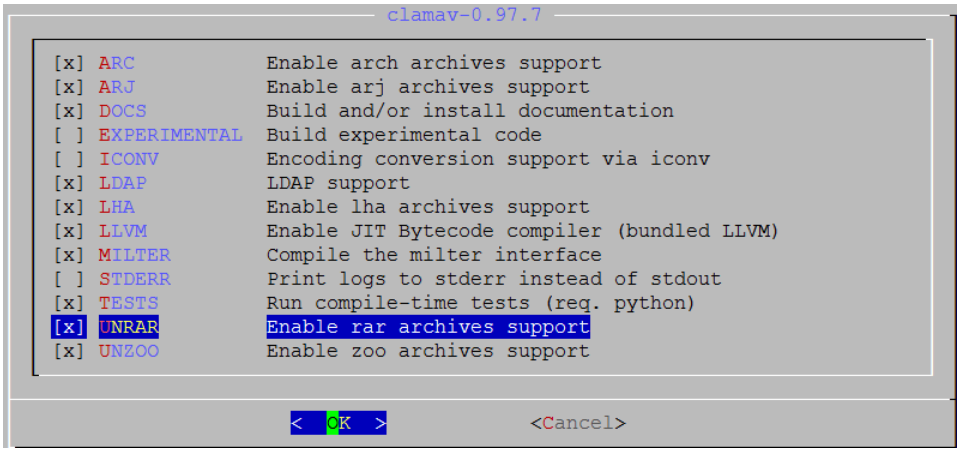
Əgər istəsəniz ki, SPAM istifadəçinin **INBOX.spam** qovluğuna yığılsın və aydın görünsün onda, **"/usr/local/etc/dovecot/dovecot.sieve"** faylında aşağıdakı dəyişikliyi eləmək lazımdır.

```
require ["fileinto"];
if header :contains "X-Spam-Status" "Yes" {
        fileinto "INBOX.Spam";
        stop;
}
```

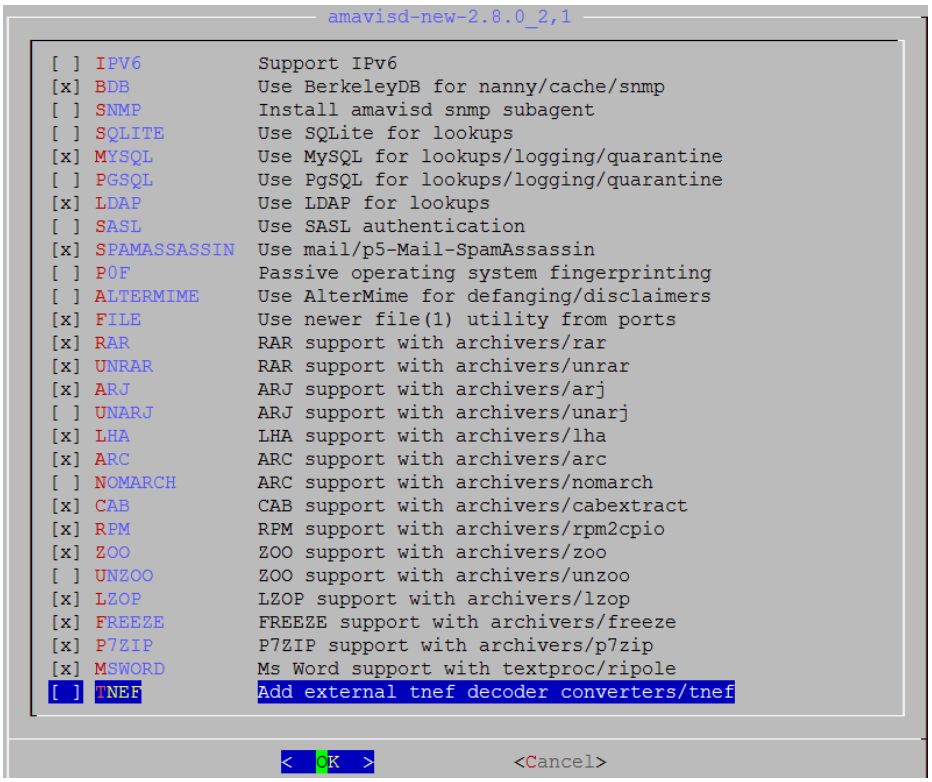
Antivirus-u quraşdıraq.

Clamav və Amavisd-New Mail-in virus filtrasiyasından cavabdehdir.

```
# cd /usr/ports/security/clamav      # Clamavi yükləyirik(0.97.7 versiyası)
# make config                        # Lazımi modulları seçirik.
```




```
# make install clean                 # Yükləyirik.

# cd /usr/ports/security/amavisd-new      # amavisd-new paketini
                                          yükləyək.(2.8.0 versiyadır.)
# make config                        # Aşağıdaki modulları seçirik.
```




```
# make install clean                    # Yükləyirik.
```

```
# cat /usr/local/etc/clamd.conf         # CLAMD quraşdırma faylına
                                        yalnız aşağıdaki sətirləri
                                        əlavə edirik.

LogFile /var/log/clamav/clamd.log
LogFileMaxSize 2M
LogTime yes
PidFile /var/run/clamav/clamd.pid
DatabaseDirectory /var/db/clamav
LocalSocket /var/run/clamav/clamd.sock.sock
FixStaleSocket yes
User clamav
AllowSupplementaryGroups yes
ScanMail yes
```

```
# cat /usr/local/etc/freshclam.conf     # FreshClam quraşdırma
                                        faylına yalnız aşağıdakı
                                        sətirləri əlavə edirik.

DatabaseDirectory /var/db/clamav
UpdateLogFile /var/log/clamav/freshclam.log
LogFileMaxSize 2M
LogTime yes
PidFile /var/run/clamav/freshclam.pid
DatabaseOwner clamav
AllowSupplementaryGroups yes
DatabaseMirror database.clamav.net
```

```
NotifyClamd /usr/local/etc/clamd.conf

# ee /usr/local/etc/amavisd.conf          # Faylda yalnız aşağıdakı sətirləri
                                          uyğun olaraq öz ünvanlarına
                                          dəyişirik və qalan sətirləri
                                          susmaya görə saxlayırıq.

$max_servers = 2;
$daemon_user  = 'vscan';
$daemon_group = 'vscan';
$mydomain = 'postfix.az';
$MYHOME = '/var/amavis';
$TEMPBASE = "$MYHOME/tmp";
$ENV{TMPDIR} = $TEMPBASE;
$QUARANTINEDIR = '/var/virusmails';
$log_level = 5;
$log_recip_templ = undef;
$do_syslog = 1;
$syslog_facility = 'mail';
$enable_db = 1;
$nanny_details_level = 2;
$enable_dkim_verification = 1;
$enable_dkim_signing = 1;
@local_domains_maps = ( [".$mydomain"] );

...
 ['ClamAV-clamd',
   \&ask_daemon, ["CONTSCAN {}\n", "/var/run/clamav/clamd.sock"],
   qr/\bOK$/m, qr/\bFOUND$/m,
   qr/^.*?: (?!Infected Archive)(.*) FOUND$/m ],
...

['ClamAV-clamscan', 'clamscan',
    "--stdout --no-summary -r --tempdir=$TEMPBASE {}",
    [0], qr/:.*\sFOUND$/m, qr/^.*?: (?!Infected Archive)(.*) FOUND$/m ],


# touch /var/log/clamav/clamd.log               # Lazımi jurnal faylları
                                                yaradırıq.

# touch /var/log/clamav/freshclam.log           # Lazımi jurnal faylları
                                                yaradırıq.

# chown -R vscan:clamav /var/log/clamav/        # Jurnal faylımız üçün vscan
                                                istifadəçi və clamav qrupu
                                                üzvlüyü veririk

# chmod -R 770 /var/log/clamav/                 # Jurnal faylımız üçün vscan
                                                istifadəçi və clamav qrupu üçün
                                                yetki veririk

# chown -R vscan:clamav /var/db/clamav/         # Clamav bazasını vscan
                                                istifadəçi və clamav qrupunun
                                                üzvü edirik.
```

```
# chmod -R 770 /var/db/clamav/                    # Clamav bazasına vscan
                                                  istifadəçi və clamav qrupu üçün
                                                  yetki veririk.

# chown -R vscan:clamav /var/amavis/              # Amavis qovluğunu vscan
                                                  istifadəçi və clamav qrupunun
                                                  üzvü edirik.

# chown -R vscan:clamav /var/run/clamav/          # Eyni işi PID faylları üçün
                                                  edirik.

# chmod -R 770 /var/run/clamav/                   # Eyni işi PID faylları üçün
                                                  edirik.
```

Antivirusumuzu Startup-a əlavə edirik.

```
# echo 'clamav_clamd_enable="YES"' >> /etc/rc.conf          # Clamd Startup
# echo 'clamav_freshclam_enable="YES"' >> /etc/rc.conf      # FreshClam
                                                            Startup

# echo 'amavisd_enable="YES"' >> /etc/rc.conf               # Amavisd-New Startup

# /usr/local/etc/rc.d/clamav-freshclam start                # Öncə FreshClam-ı
                                                            start edirik.
```

Clamavda bug olduğuna görə aşağıdakı addımları əlimizlə **'sock.sock'** faylı üçün edirik. ☺

```
# touch /var/run/clamav/clamd.sock.sock
# chown -R vscan:clamav /var/run/clamav/
# chmod -R 770 /var/run/clamav/

# freshclam                                 # Antivirus Bazamızı yeniləyirik.
# /usr/local/etc/rc.d/clamav-clamd start        # ClamD-ni işə salırıq.
# /usr/local/etc/rc.d/amavisd start             # AmavisD-ni işə salırıq.
```

Sonda isə inteqrasiyanı bitirmək üçün **'/usr/local/etc/postfix/main.cf'** və **'/usr/local/etc/postfix/master.cf'** faylının sonlarına lazımi sətirləri əlavə eləmək lazımdır.

```
# ee /usr/local/etc/postfix/main.cf       # Faylın içində content_filter
                                          sətirin qarşısından şərhi silirik.
## Antivirus Filter edilməsi
content_filter=smtp-amavis:[localhost]:10024

# cat /usr/local/etc/postfix/master.cf    # Faylın sonuna aşağıdakı sətirləri
                                          əlavə edirik.
# Amavis listen
smtp-amavis  unix    -     -     n       -       2  smtp
    -o smtp_data_done_timeout=1200
    -o smtp_send_xforward_command=yes
```

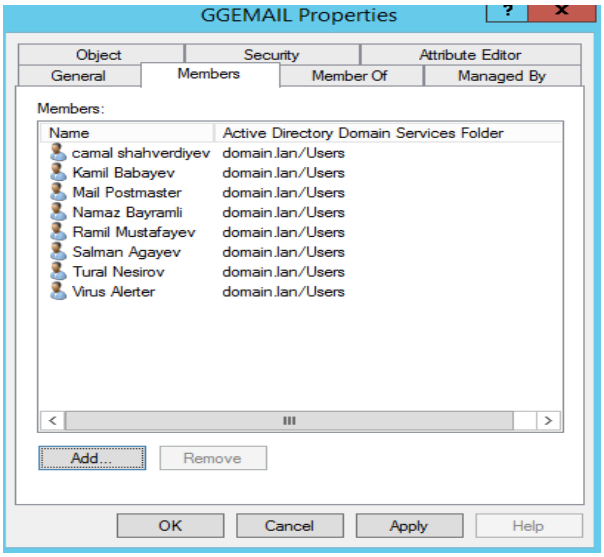
```
    -o disable_dns_lookups=yes

127.0.0.1:10025 inet  n   -     n       -       -  smtpd
    -o content_filter=
    -o local_recipient_maps=
    -o relay_recipient_maps=
    -o smtpd_restriction_classes=
    -o smtpd_delay_reject=no
    -o smtpd_client_restrictions=permit_mynetworks,reject
    -o smtpd_helo_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o mynetworks_style=host
    -o mynetworks=127.0.0.0/8
    -o strict_rfc821_envelopes=yes
    -o smtpd_error_sleep_time=0
    -o smtpd_soft_error_limit=1001
    -o smtpd_hard_error_limit=1000
    -o smtpd_client_connection_count_limit=0
    -o smtpd_client_connection_rate_limit=0
    -o
receive_override_options=no_header_body_checks,no_unknown_recipient_checks

# /usr/local/etc/rc.d/postfix restart      # Və sonda Postfix-i yenidən işə
                                           salırıq.
```

Test üçün aşağıdakı sətirdə olan tərkibi əlavə edərək email yollayın və nəticəyə baxın. Email virus kimi **'/var/log/maillog'** faylında qeydə alınacaqdır.

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

# telnet localhost 25                      # Tünd qara simvollar əmrlərdir.
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.postfix.az ESMTP Postfix (2.9.5)
helo localhost
250 mail.postfix.az
mail from: kamil.babayev@postfix.az
250 2.1.0 Ok
rcpt to: camal.shahverdiyev@postfix.az
250 2.1.5 Ok
Data
354 End data with <CR><LF>.<CR><LF>
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

.
250 2.0.0 Ok: queued as ADA33112C72
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

Ashaqidaki setirleri log faylinda gormelisiniz.
Apr 10 16:08:07 postfix-ldap amavis[28607]: (28607-01) **Blocked INFECTED** (Eicar-Test-Signature) {**DiscardedInternal,Quarantined**}, MYNETS LOCAL [127.0.0.1]:25791 [127.0.0.1] <**kamil.babayev@postfix.az**> -> <**camal.shahverdiyev@postfix.az**>, quarantine: **virus-9XJQSiKlfwnh**, Queue-ID: ADA33112C72, Message-ID: <20130410110753.ADA33112C72@mail.postfix.az>, mail_id: 9XJQSiKlfwnh, Hits: -, size: 394, 360 ms

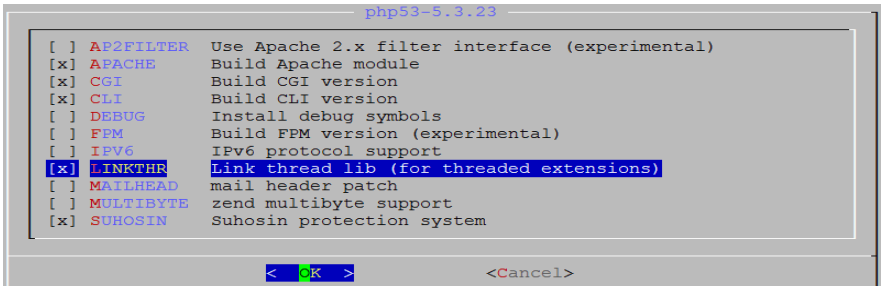
Əgər siz jurnallara tam diqqətlə baxsanız görəcəksiniz ki, virus mənşəli emaillər **'virusalert@postfix.az'** istifadəçisinə dovecot tərəfindən yönləndirilir. Bunun üçün siz AD-də həmin istifadəçini yaradıb **'GGEMAIL'** qrupuna əlavə eləməlisiniz. Beləliklə sonda AD-mizdə test üçün GGEMAIL qrupunda aşağıda şəkildə göstərilən istifadəçilər olacaq. Bunlardan mütləq olanlar. **Admin: camal.shahverdiyev, Mail Postmaster: postmaster** və **Virus Alerter: virusalert**



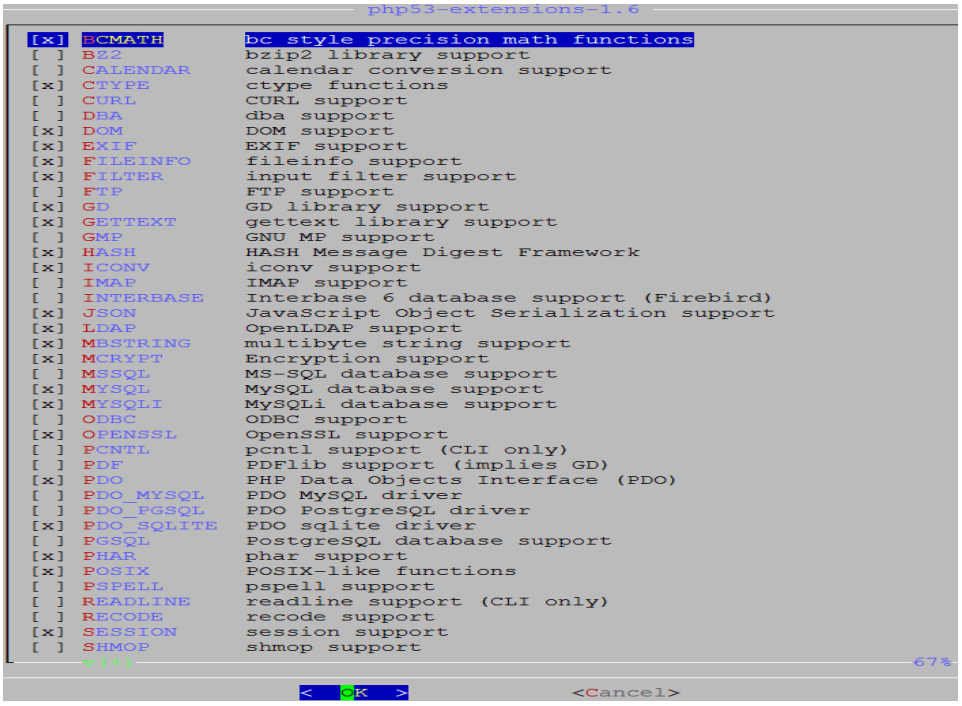

```
# ll /var/virusmails/    # Bu ünvanda isə həmin virusları görə bilərsiniz.
-rw-r-----  1 vscan  vscan  1028 Apr 10 16:08 virus-9XJQSiKlfwnh
-rw-r-----  1 vscan  vscan  1028 Apr 10 16:16 virus-Bamtri8mxBRp
```

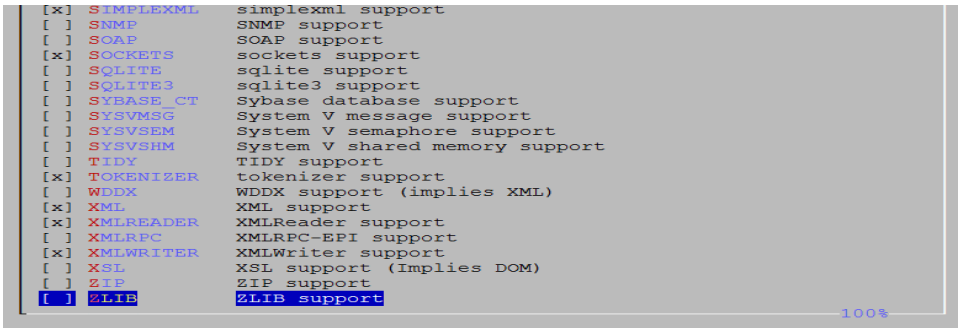
Indi isə Maillərimizə WEB-dən yetki alaq.
Bunun üçün öncə **Apache, PHP5** və **MySQL**-ı yükləmək lazımdır. Çünki biz həm **SquirrelMail** həmdə **RoundCube** istifadə edəcəyik.

```
# cd /usr/ports/www/apache22          # apache22-nin portuna daxil oluruq.
# make config                              # Susmaya görə olan modulları daxil
                                           edirik.(Bütün dependslərdə IPv6-dan
                                           başqa)
# make install clean                       # Yükləyirik.

# echo 'apache22_enable="YES"' >> /etc/rc.conf  # Startup-a əlavə edirik.

# cd /usr/ports/lang/php53            # PHP5.3-ü yükləyirik(5.3.23 versiyası)
# make config                         # Aşağıdaki modulları seçirik.
```




```
# make install clean                    # Yükləyirik. Bütün Dependslərdə IPv6-dan
                                        başqa hər şey susmaya görə

# ee /usr/local/etc/apache22/httpd.conf    # Aşağıdaki sətirləri faylın sonuna
                                           əlavə edirik.
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps

DirectoryIndex index.html index.php        # index.php-ni bu sətirin qarşısına
                                           yazırıq.

# cd /usr/ports/lang/php53-extensions/     # PHP53 üçün lazım olan
                                           genişlənmələr yükləməliyik.
# make config                              # Aşağıdakı modulları
                                           seçirik(SQLITE və IPV6-dan başqa)
```



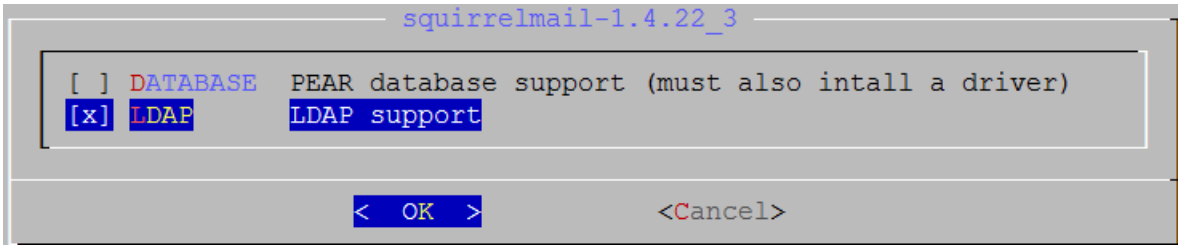



```
# make install clean                                    # Yükləyirik.
```

WEBMail-in qurulması.

Biz həm **SquirrelMail** həmdə **Roundcube**-un qurulmasını edəcəyik. Ancaq birinci **SquirrelMail**-dən başlayaq.

```
# cd /usr/ports/mail/squirrelmail    # SquirrelMail-i portlardan yükləyək.
# make config                        # Lazımi modulları seçək.
```




```
# make install clean                 # Yükləyirik.

# ee /usr/local/etc/apache22/Includes/squirrelmail.conf # SquirrelMail-in
                                                        WEB-dən açılması
                                                        üçün onun
                                                        quraşdırmasını
                                                        apache-a əlavə
                                                        edək.Faylın
                                                        tərkibi
                                                        aşağıdaki
                                                        sətirlərdən
                                                        ibarət
                                                        olacaq.
Alias /squirrelmail/ "/usr/local/www/squirrelmail/"
<Directory "/usr/local/www/squirrelmail">
   AllowOverride None
   Options None
   Order allow,deny
   Allow from all
</Directory>


# cd /usr/local/www/squirrelmail/    # SquirrelMail-in quraşdırılması üçün
                                     qovluğa daxil oluruq
# ./configure                        # Scripti işə salırıq və Menyu açılır
```

1. Seçirik Opsiya: "**2. Server Settings**"
2. Seçirik Opsiya: "**1. Domain**" # Domain adını veririk(Bizim halda: **postfix.az** və **Enter**)
3. Seçirik Opsiya: "**A. Update IMAP Settings**" və "**5. IMAP Port**" sonra isə **993**-ü daxil edirik və **Enter**.
4. Seçirik Opsiya: "**7. Secure IMAP (TLS)**" və "**Enable TLS (y/n) [n]: y**" edib **Enter** sıxırıq.
5. Seçirik Opsiya: "**8. Server software**" və "**dovecot**" sözünü daxil edib **Enter** sıxırıq.
6. Seçirik Opsiya: "**R Return to Main Menu**" sıxıb əsas menyuya qayıdırıq.
7. Seçirik Opsiya: "**10. Languages**" və "**2. Default Charset**"-i **utf-8** yazıb **ENTER** sıxırıq.
8. Seçirik Opsiya: "**S Save data**" və iki dəfə **ENTER** sıxırıq. "**Q Quit**" çıxırıq.

SquirrelMail-in **Quota Pluginini** yükləyək.

```
# cd /usr/ports/mail/squirrelmail-check_quota-plugin/ # Port-una daxil olaq.
# make install clean                                  # Yükləyək.
# cd /usr/local/www/squirrelmail/plugins/check_quota  # Config qovluğuna
                                                      daxil olaq ki,
                                                      quraşdıraq.
# cp config.sample.php config.php                     # Sample faylını
                                                      quraşdırma faylına
                                                      nüsxələyək.

# ee config.php                                       # Faylın içində
                                                      aşağıdaki sətirlərə
                                                      uyğun dəyişiklikləri
                                                      edin.

$settings['quota_type'] = 1;
$settings['graph_type'] = 1;
$settings['info_above_folders_list'] = 0;
$settings['show_intro_texts'] = 1;
$settings['details_above_graph'] = 0;
```

AutoSubscribe Pluginin yüklənməsi.

Bu Plugin bütün istifadəçilər üçün Spam qovluğunun yaradılmasına cavabdehdir. O həmçinin '**Maildir**' qovluğunuda yeniləyir.

```
# cd /usr/local/www/squirrelmail/plugins                # Ünvana daxil
                                                        oluruq ki, plugin
                                                        yükləyək.
                                                        autosubscribe-
                                                        1.1-1.4.2.tar.gz
                                                        adlı modulu
```

internetdən bu qovluğa endirin.

```
# tar -zvxf autosubscribe-1.1-1.4.2.tar.gz
```
tar.gz faylı plugins qovluğuna açırıq.

```
# cd autosubscribe
```
Açdığımız qovluğa daxil oluruq.

```
# cp config_sample.php config.php
```
Sample faylını config faylına nüsxələyək.

```
# ee config.php
```
config faylında aşağıdaki iki sətiri uyğun olaraq dəyişin.

```
$autosubscribe_folders='INBOX.Spam';
$autosubscribe_special_folders='INBOX.Spams';
```

TimeOut Plugin-in yüklənməsi

```
# cd /usr/ports/mail/squirrelmail-timeout_user-plugin
```
Port-una daxil oluruq.

```
# make install clean
```
Yükləyirik.

İşə salmaq üçün isə **'/usr/local/www/squirrelmail'** qovluğuna daxil olub **'configure'** scriptini işə salmaq lazımdır.

```
# cd /usr/local/www/squirrelmail
```
SquirrelMail qovluğuna daxil oluruq.

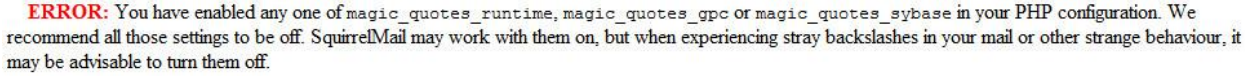
```
# ./configure
```
Scripti işə salırıq.

1. Seçirik: "**8. Plugins**"
2. Plugini yükləmək üçün sadəcə onun rəqəminə sıxmaq yetər. Və istədiyiniz Pluginləri seçə bilərsiniz.
3. Seçirik: **"compatibility", "check_quota", "timeout_user", "autosubscribe", "calendar", "administrator"**
4. Seçirik: "**S Save data**" sonra **Enter** və "**Q Quit**"

```
# echo "192.168.1.100 `hostname`" >> /etc/hosts
```
Apache-i aldadaq ki, tez işə düşsün.

```
# /usr/local/etc/rc.d/apache22 restart
```
Sonda apache-i işə salaq.

Sonda isə aşağıdaki linkə daxil olaraq SquirrelMail-mizi test edirik.
http://mail.postfix.az/squirrelmail/src/configtest.php
Əgər sizdə Date/TimeZone səhvi və aşağıdaki şəkildə olan səhv çıxsa


ERROR: You have enabled any one of magic_quotes_runtime, magic_quotes_gpc or magic_quotes_sybase in your PHP configuration. We recommend all those settings to be off. SquirrelMail may work with them on, but when experiencing stray backslashes in your mail or other strange behaviour, it may be advisable to turn them off.

Onu aşağıdaki qaydada düzəldə bilərsiniz.

```
# cd /usr/local/etc/
```
PHP inisializasiya üçün qovluğuna daxil oluruq

```
# cp php.ini-production php.ini
```
Inisializasiya faylını copy edək.

```
# ee php.ini
```
PHP-nin inisializasiya faylını aşağıdaki sətirlərə uyğun olaraq dəyişin

```
short_open_tag = On
```

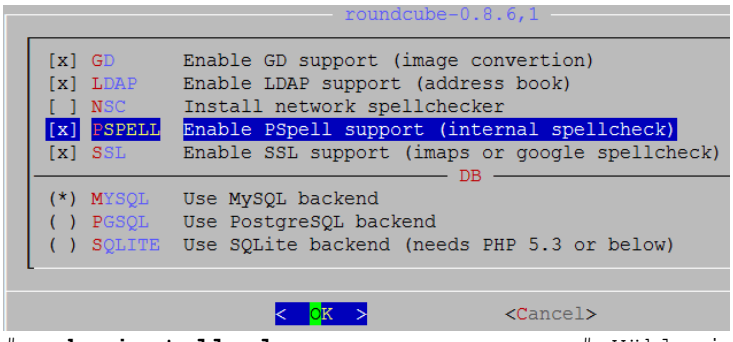
```
date.timezone = "Asia/Baku"
```

```
# apachectl graceful      # Apache-i reload edək və yenədə Browserdən test
                          edək. Aşağıdaki şəkilə uyğun bir şəkil çap
                          edilməlidir. Yəni nəticə uğurludur.
```

Sonda isə aşağıdaki linkə daxil olub AD-də yaradılan istifadəçi və şifrə ilə daxil oluruq.
http://mail.postfix.az/squirrelmail/

WEBMail RoundCube

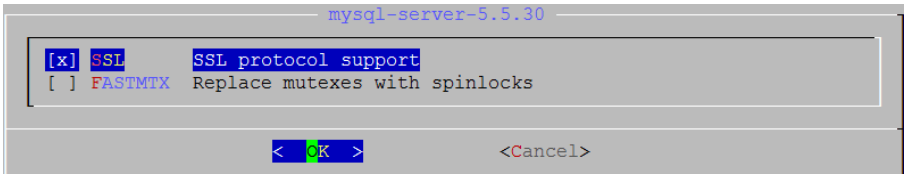
```
# cd `whereis roundcube | awk '{ print $2 }'`    # RoundCube-un Portuna daxil
                                                  oluruq.
# make config                          # Şəkildə göstərilən modulları seçirik.
```




```
# make install clean                   # Yükləyirik.
```

RoundCube-un işləməsi üçün biz ona MySQL baza, istifadəçi adı və şifrə yaratmalıyıq. Bunun üçün isə MySQL-i yükləmək lazımdır.

```
# cd /usr/ports/databases/mysql55-server/        # Portuna daxil oluruq.
# make config                                    # Yalnız SSL modulu seçirik.
```




```
# make install clean                             # Yükləyirik.

# echo 'mysql_enable="YES"' >> /etc/rc.conf      # MySQL-i Startup-a əlavə
                                                 edirik.
# /usr/local/etc/rc.d/mysql-server start         # İşə salırıq.

# /usr/local/bin/mysql_secure_installation       # MySQL-i quraşdıraq.
Enter current password for root (enter for none):
Set root password? [Y/n] Y                       # Yes deyirik.
New password:                                    # Yeni şifrəni yazırıq.
Re-enter new password:                           # Yeni şifrəni təkrar
                                                 yazırıq.
Remove anonymous users? [Y/n] Y                  # Yes deyirik
Disallow root login remotely? [Y/n] Y            # Yes deyirik
Remove test database and access to it? [Y/n] Y   # Yes deyirik
Reload privilege tables now? [Y/n] Y             # Yes deyirik

# mysql -u root -p                               # MySQL-ə daxil oluruq.
mysql> CREATE DATABASE roundcubemail;            # RoundeCube üçün baza
                                                 yaradırıq.
Query OK, 1 row affected (0.00 sec)

# Həmin baza üçün istifadəçi adı və şifrə yaradırıq.
mysql> GRANT ALL PRIVILEGES ON roundcubemail.* TO roundcube@localhost
IDENTIFIED BY 'freebsd';
Query OK, 0 rows affected (0.00 sec)

# chown -R www:www /usr/local/www/roundcube/     # RoundCube fayllarına Apache
                                                 üçün yetki veririk.

# Apache üçün yeni quraşdırma faylları ünvanı yaradırıq.
# echo "Include /usr/local/domen/*" >> /usr/local/etc/apache22/httpd.conf

# mkdir -p /usr/local/domen/          # Yetki verdiyimiz qovluğu yaradırıq.
# chown -R www:www /usr/local/domen   # Apache üçün həmin qovluğa yetki
                                      veririk.

# mv /usr/local/www/roundcube/.htaccess /root/homefold-htaccess    # Mütləq
                                                                   bunu
                                                                   edirik. Əks
                                                                   halda WEB
                                                                   ilə yetki
                                                                   ala
```

bilməyəcəksiniz

Həmçinin qeyd eləmək istəyirəm ki, roundcube-dan çıxan error mesajları siz **'/var/log/httpd-error.log'** faylından əldə edə bilərsiniz.

```
# ee /usr/local/domen/mail.postfix.az  # Yeni VirtualHost yaradıb içinə aşağıdaki tərkibi əlavə edirik
<VirtualHost *>
   ServerName mail.postfix.az
   ServerAlias www.mail.postfix.az
   DocumentRoot "/usr/local/www/roundcube"
<Directory "/usr/local/www/roundcube">
   Options All
   Options FollowSymLinks
   AllowOverride AuthConfig
   Order allow,deny
   Allow from all
</Directory>
</VirtualHost>


# apachectl graceful                    # Apache-ı reload edirik.
# apachectl -t                          # Apache-ı test edirik.
# httpd -S                              # VirtualHost-u Test edirik
```

Sonra WEB ilə aşağıdaki linkə daxil olaq ki, RoundCube-un tələbatlarını yoxlayaq. Ancaq quraşdırmalarımız bitdikdən sonra mütləq **'/usr/local/www/roundcube/installer/'** qovluğunu ya silin yada yerini dəyişin.
http://mail.postfix.az/installer

Aşağıdaki şəkilə uyğun formada bir şəkil çap ediləcək. Və **'Next'** düyməsini sıxırıq.

Checking PHP version

Version: **OK** (PHP 5.3.23 detected)

Checking PHP extensions

The following modules/extensions are *required* to run Roundcube:

PCRE: **OK**
DOM: **OK**
Session: **OK**
XML: **OK**
JSON: **OK**

The next couple of extensions are *optional* and recommended to get the best performance:

FileInfo: **OK**
Libiconv: **OK**
Multibyte: **OK**
OpenSSL: **OK**
Mcrypt: **OK**
Intl: **OK**
Exif: **OK**

Checking available databases

Check which of the supported extensions are installed. At least one of them is required.

MySQL: **OK**
MySQLi: **OK**
PostgreSQL: **NOT AVAILABLE** (Not installed)
SQLite (v2): **NOT AVAILABLE** (Not installed)

Check for required 3rd party libs

This also checks if the include path is set correctly.

PEAR: **OK**
MDB2: **OK**
Net_SMTP: **OK**
Net_IDNA2: **OK**
Mail_mime: **OK**

Checking php.ini/.htaccess settings

The following settings are *required* to run Roundcube:

file_uploads: **OK**
session.auto_start: **OK**
zend.ze1_compatibility_mode: **OK**
mbstring.func_overload: **OK**
magic_quotes_runtime: **OK**
magic_quotes_sybase: **OK**
date.timezone: **OK**

The following settings are *optional* and recommended:

allow_url_fopen: **OK**

NEXT

Quraşdırma faylları **'/usr/local/www/roundcube/config'** qovluğuna nüsxələdikdən sonra aşağıdaki sətiri uyğun olaraq **'main.inc.php'** faylında dəyişin.

$rcmail_config['support_url'] = 'http://mail.postfix.az';

MySQL-i Roundcube WEB ilə quraşdıraq.

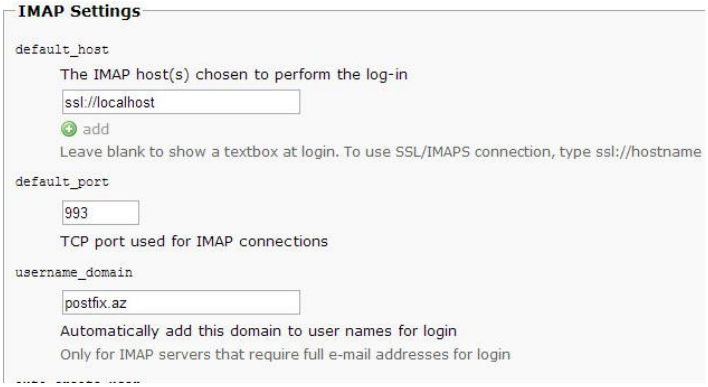
Database setup

db_dsnw

Database settings for read/write operations:

MySQL Database type
localhost Database server (omit for sqlite)
roundcubemail Database name (use absolute path and filename for sqlite)
roundcube Database user name (needs write permissions)(omit for sqlite)
••••••• Database password (omit for sqlite)

Imap-i quraşdıraq.

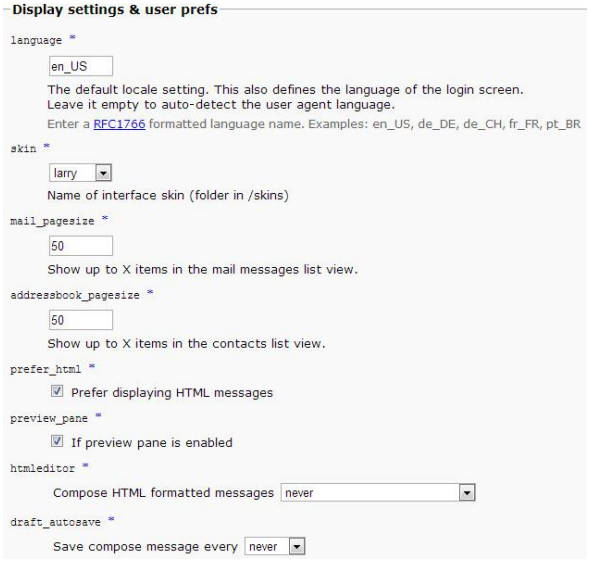


CLI-dan **IMAPS**-in test edilməsi üçün aşağıdaki əmrdən istifadə edə bilərsiniz.

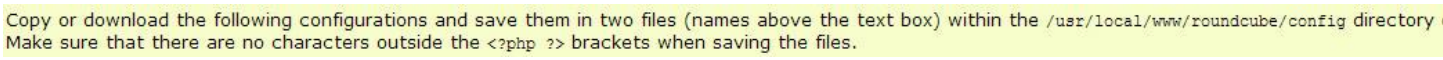
openssl s_client -connect localhost:993 # Bu əmrlə **SSL** ilə **Dovecot**-un Port-una qoşuluruq.

SMTP-ni quraşdıraq.

Ekran opsiyalarından aşağıdakilarda dəyişiklik edirik. Və **"Create Config"** düyməsinə sıxırıq.



Sonra şəkildəki göstərilən **'main.inc.php'** və **'db.inc.php'** kimi faylları yükləyib **'/usr/local/www/roundcube/config'** qovluğuna yerləşdirmək lazımdır.

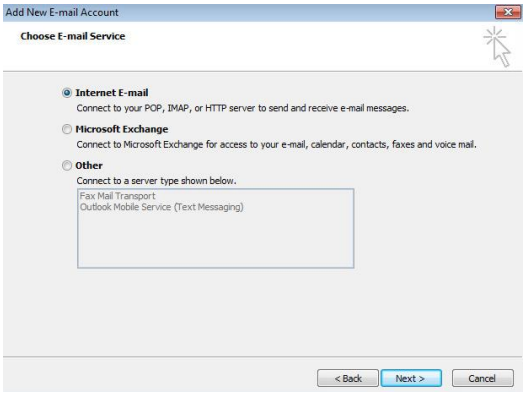



```
chown -R www:www roundcube/          # Apache istifadəçisinə Roundcube qovluğu üçün yetki veririk. Ardınca da "Continue" düyməsinə sıxmaq lazımdır.
```

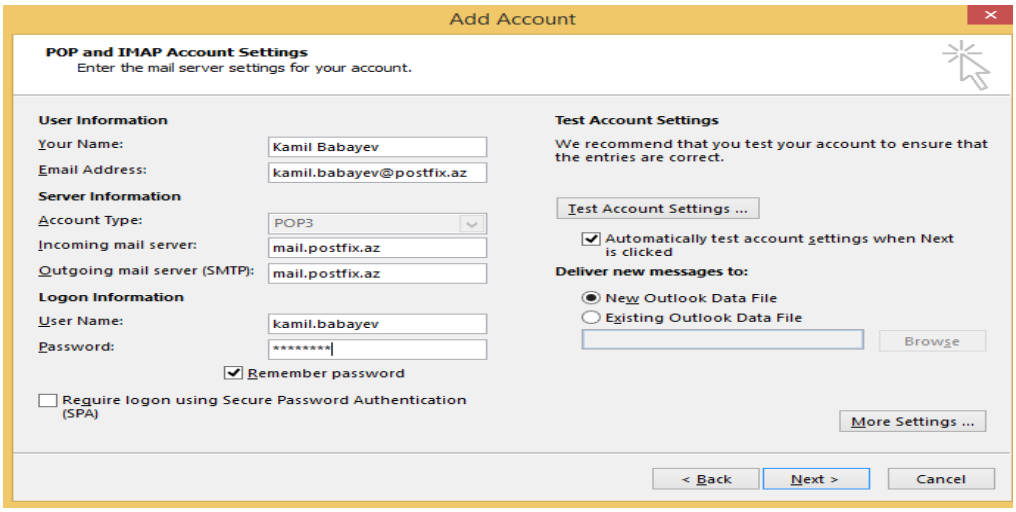
SMTP və IMAP test etdikdə aşağıdaki nəticəni verməlidir sizə.

İndi isə İki Client arasında OutLook quraşdıraq.

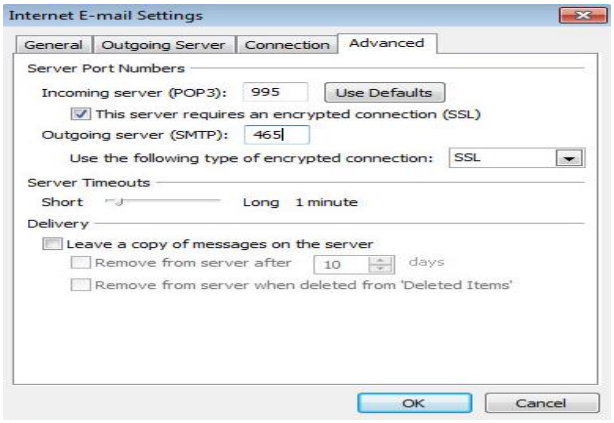
Client-in biri **kamil.babayev** digəri isə **ramil.mustafayev** olacaq. Hal-Hazırda **Kamil Babayev** üçün POP quraşdıracıyıq. Şəkildə göründüyü kimi POP/IMAP seçib **Next** edirik.



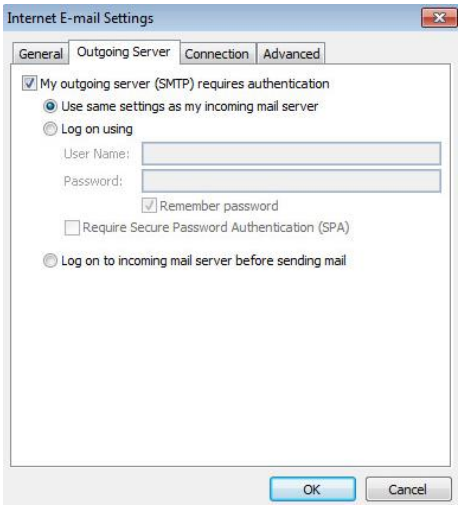
Indi isə **POP/Imap**, **Login** və **Şifrə** quraşdırmalarımızı edek və ardınca **'More Settings'** düyməsini sıxaq.



Sonra isə "**Advanced**" bölümündə **POP3**-də **'This server requires an encrypted connection (SSL)'** düyməsinə quş qoyuruq və **SMTP**-də isə **465**-ci port yazıb **SSL** seçirik.

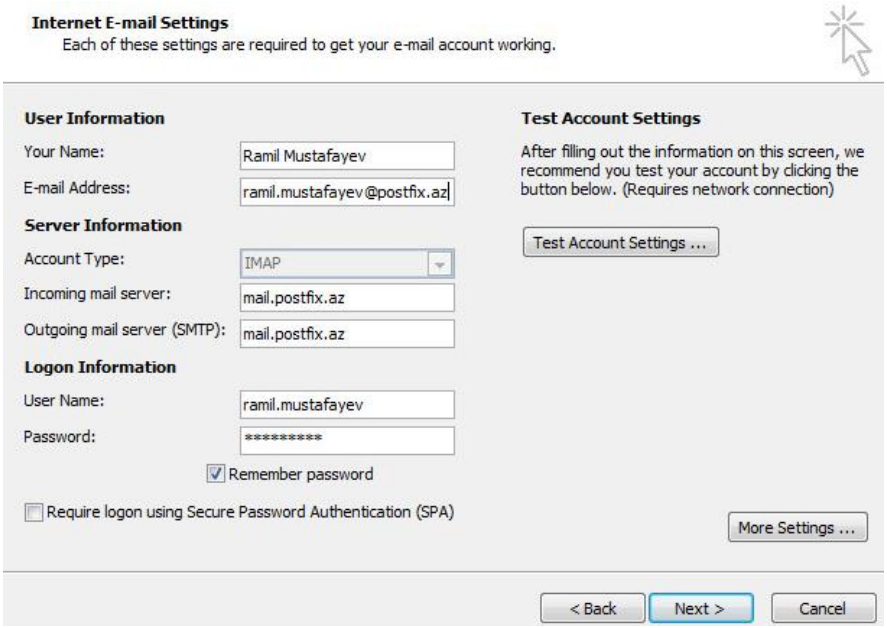


Və **'Outgoing Server'** bölümündə isə **'My outgoing server (SMTP) requires authentication'** seçirik.

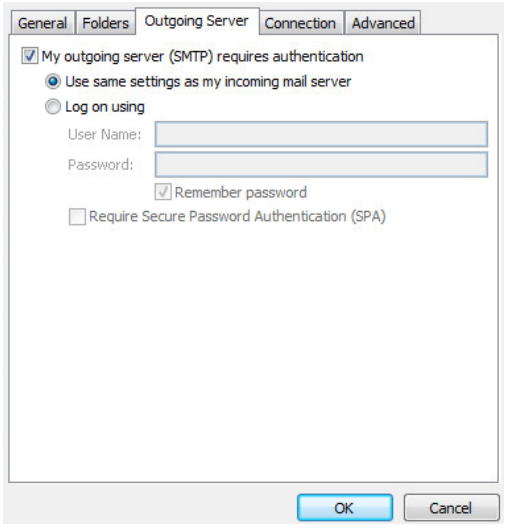


Sonra **"Ok"**, **"Next"** və **"Finish"**. Email yeşiyi istifadəçi adına yaradılmasını təklif edəndə şəkildəki kimi qəbul edib **'OK'** düyməsini sıxırıq.

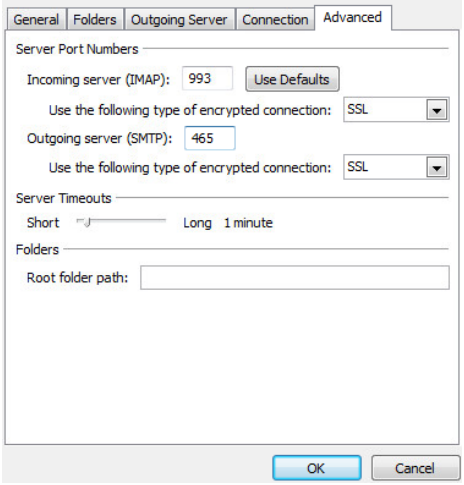
İndi isə Digər clientlə **ramil.mustafayev** istifadəçisinin email clientini **IMAPS/SMTPS** üçün quraşdıraq. **kamil** istifadəçisində etdiyimiz kimi eyni qaydada olacaq. Ancaq burda protocol **POP** yox **IMAP** seçiləcək.



Eyni qaydada olaraq **"More Settings"** ardınca **"Outgoing Server"** və **"My outgoing server (SMTP)requires authentication"** seçirik.

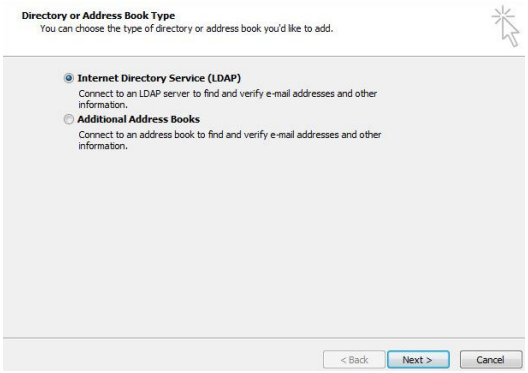


Sonra da **"Advanced"**-ə keçib **IMAP/SSL** seçirik və **SMTP/SSL** seçib portu **465** edirik və **"OK"** sıxırıq.

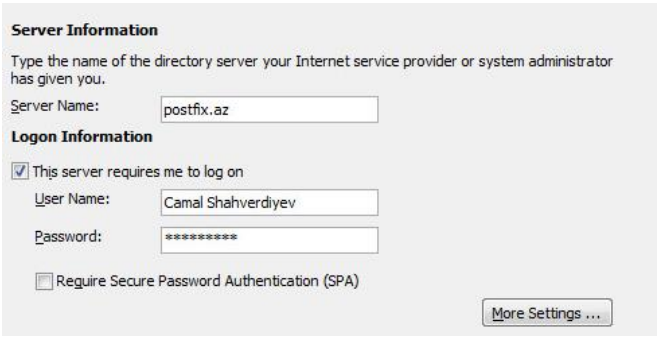


"**Next**" və "**Finish**". Çıxan istifadəçi adı inisializasiyasına **OK** cavabı veririk. Test üçün fayl attach edərək mail yollayin. Həmçinin WEB ilə.

Istifadəçinin LDAP bazasından istifadəçi listlərini əldə eləmək istəsəniz aşağıdakı qaydanı hər bir istifadəçidə eləsəniz yetər.
Microsoft Outlook 2007 Client-də **Tools -> Account Settings -> Address Books -> new** və ardıcıllıq aşağıdaki qaydada edəcəksiniz. Şəkildə göründüyü kimi. Next düyməsini sıxın.

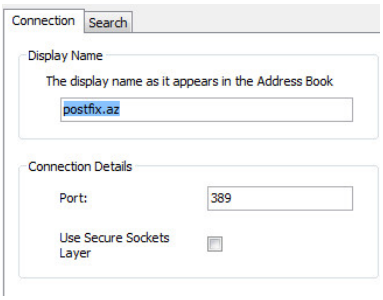


Atributları şəkildəki kimi doldururuq.



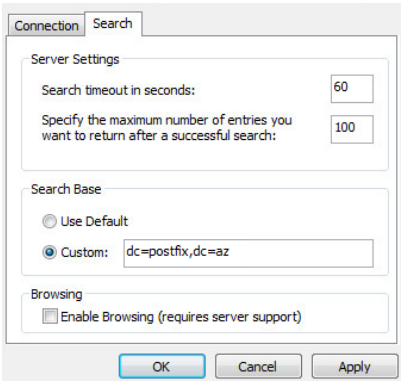
Server Name: **postfix.az** (This server requires me to log on – seçirik)
User Name: **Camal Shahverdiyev**
Password: *************

Sonra **'More Settings'** düyməsini sıxırıq. "**Connection**" bölümündə isə aşağıdaki quraşdırmaları edirik.



Display name: **postfix.az**
Port: **389**

Və sonda **'Search'** bölümünə keçib şəkildəki quraşdırmaları edirik. Və **OK** -> **Finish** düyməsini sıxırıq.



Custom: **dc=postfix,dc=az**
Ctrl+Shift+B -> Tools -> Options Köhnə **Contacts**-i seçib **remove** düyməsini sıxsaz və "**Show this address list first: postfix.az**" seçsəz axtarış üçün daha rahat olar. İstifadəçinin **Contacts Address Book**-da **postfix.az** seçməyi unutmayın. Sonra **kamil** istifadəçisini axtarış edin və nəticəni görəcəksiniz. Unutmayın **LDAP** quraşdırmalarını **Admin yox** hər istifadəçinin **öz adından** eləsəniz də **işləyəcək**.

BÖLÜM 9

Linux üçün disk və şəbəkə dayanıqlığı

- **Linux BOND**
- **Linux FCoE**
- **Multipath disklərin işlək vəziyyətdə genişləndirilməsi**

İstənilən Linux və Unix əməliyyat sistemlərinin üstündə şəbəkə kartının birləşdirilməsi imkanı mövcuddur. Tələb, mövcud serverin 1 Gigabitlik şəbəkə kartının keçirilmə qabiliyyəti tab gətirmədikdə yaranır. Bu tələbin qarşılığında bond deyilən proqram təminatı var hansı ki, Cisco-nun channel-group-na uyğun metodu ilə öz daxili şəbəkə kartlarını virtual strukturda düzür. Bu başlıqda bondu açıqlayırıq. Eyni zamanda da artıq fiber-channel qoşulmaları ethernet uzərindən daha təkmilləşdirdiyinə görə, ethernet üzərindən fiber-channel trafikinin ötürülməsi üçün şəbəkə kartında lazımi quraşdırmaların edilməsi açıqlanacaq. Şirkətinizin işlək bir sisteminin üzərində belə bir tələb yarana bilər ki, FC vasitəsilə paylaşılmış disklər yenidən formatlanmadan artırılmış hissəsi istifadəyə verilsin. Bu halda siz diski umount edə bilməyəcəksiz və məcburi extend edib formatlayacaqsınız. Başlığımız bunun haqqında da danışır.

Linux BOND

Bonding nədir və bu necə işleyir

Bonding portun trunk edilməsi ilə eyni şeydir. Bonding terminini məhz ona görə istifadə edirik ki, bir neçə şəbəkə kartını 1 nöqtəyə cəmləşdiririk.

Bonding sizə izin verir ki, çoxlu portları 1 qrup daxilində əlaqələndirəsiz hansı ki, şəbəkə genişliyini effektiv şəkildə birləşdirir. Bonding həmçinin sizə şərait yaradır ki, multi-gigabitlik kanallar yaradaraq trafikinizi geniş şəbəkə axını üzərindən ötürə biləsiniz. Misal üçün siz 3 ədəd megabitlik portlarınızı 1 ədəd 3 megabitlik trunk portun üzərindən birləşdirə bilərsiniz. Bu 3 megabit sürətin ekvivalenti olacaq.

Harda bonding-i istifadə etməliyəm?

Siz istənilən dayanıqlı linklər, səhvə davamiyyət yada yükün bölüşdürülməsi üçün bunu istifadə edə bilərsiniz. Bu yüksək davamiyyətli şəbəkə segmentinin əldə edilməsi üçün ən yaxşı yoldur. Bonding-i əksər hallarda 802.1q VLAN dəstəklənməsində istifadə edirlər(həmçinin sizin şəbəkə avadanlığı da 802.1q protokolun istifadə edilməsini dəstəkləməlidir)

Bonding-in hansı tipləri mövcuddur

mode=1 (active-bakup)
Active-backup politikası: Yalnız bond tabeçiliyində olan şəbəkə kartlarından biri aktiv vəziyyətdə olur. Digər interfeyslərdən biri yalnız və yalnız o halda aktiv vəziyyətdə gəlir ki, aktiv olan interfeys-də səhv baş verir yada hansısa səbəbdən deaktiv vəziyyətə keçir. Bond-un MAC ünvanı çöl tərəfdə yalnız bir şəbəkə kartı üzərində görünür ki, şəbəkə Swith-ini caşdırmasın. Bu rejim səhvə davamlı şəraiti yaradır.

mode=2(balance-xor)
XOR politikası: Qayıdışa əsaslanır[(soure MAC ünvan, destination MAC ünvan ilə XOR-laşdırırlır) ikinci dərərcəli şəbəkə kartlarını sayğaca salır. Bu mənsəbdə olan hər bir MAC ünvan üçün, asılılığında olan eyni slave-i seçir. Bu rejim yükə davamiyyət və səhvə davamlılıq üçün şərait yaradır.

mode=3(broadcast)
Broadcast politikası: Asılılığında olan bütün şəbəkə kartları üzərindən hər şeyi ötürür. Bu rejim səhvə davamlılıq üçündur.

mode=4(802.3ad)
IEEE 802.3ad Dynamic link aggregation. Ümumi qrup yaradır hansı ki, bunda öz növbəsində eyni sürət və duplex quraşdırmalarını yayımlayır. Bütün asılılığında olan slavə-ləri bir aktiv birləşdiricidə utilizasiya edir hansı ki 802.3ad spesifikasiyasında bu haqda ətraflı yazılır.

- Planlı tələblər:

- **Ethtool** aləti ilə siz sürət və duplex haqqında, hər bir şəbəkə kartı haqqında ətraflı melumat əldə edə bilərsiniz.
- Switch IEEE 802.3ad Dynamic Linc aggregation-u dəstəkləməlidir. Əksər Switchlər bəzi tip quraşdırmalarda hər bir hal üçün 802.3ad rejiminin aktivləşdirilməsini tələb edir.

mode=5(balance-tlb)
Adaptive transmit load balancing: Kanal bonding-i hansı ki, heç bir switch dəstəklənməsinə ehtiyacı yoxdur. Cıxış trafiki hal-hazırkı yükləmənin içində hər bir slave üçün yayımlanır(Sürətdən asılı olaraq hesablanır). Gələn trafik isə hal-hazırkı slave-dən daxil olur. Əgər daxil olan trafikin slave-ində problem olarsa, digər slave adapter düşən slave-in MAC ünvanını özünə götürür.

- Plan: Baza driverlərində slave-lərin sürətinin hesablanması üçün **ethtool** istifadə edilir

mode=6(balance-alb)
Adaptive load balancing: IPv4 trafiki üçün balance-tlb və load-balancing-i özündə cəmləşdirir və heç bir switch dəstəklənməsinə eythiyacı yoxdur. Qayıdış yük bölünməsi ARP razılaşması ilə həll edilir. Bond driver-i local sistemdən gələn ARP cavabları cıxışda tutur və mənbənin MAC ünvanını silib öz bond Slave-ində olan adapterlərin birinin MAC ünvanını yazır ki, fərqli ünvanlar server üçün fərqli avadanlıq ünvanı istifadə etsin.

CentOS6.5-də bunun quraşdırması aşağıdakı kimi olur.

Deyək ki, iki ədəd şəbəkə kartımız var **eth0** və **eth1**. Bu şəbəkə kartlarını **bond0** adında birləşdiririk.

```
[root@bimn1 network-scripts]# cat ifcfg-bond0    # Bond0-u yaradırıq
DEVICE=bond0
IPADDR=10.40.7.50
NETMASK=255.255.255.0
GATEWAY=10.40.7.1
ONBOOT=yes
BOOTPROTO=static
BONDING_OPTS="mode=1 miimon=100 primary=eth0"    # Bond üçün mode1-i seçirik,
                                                 yəni active-backup
USERCTL=no

[root@bimn1 network-scripts]# cat ifcfg-eth0     # eth0-i bond0-a əlavə edirik
DEVICE="eth0"
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
MASTER=bond0
SLAVE=yes
USERCTL=no

[root@bimn1 network-scripts]# cat ifcfg-eth1     # eth1-i bond0-a əlavə edirik
DEVICE="eth1"
```

```
BOOTPROTO="static"
ONBOOT="yes"
TYPE="Ethernet"
MASTER="bond0"
SLAVE=yes
USERCTL=no
```

```
[root@bimn1 ~]# cat /proc/net/bonding/bond0      # Bond0 statusuna baxırıq
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: load balancing (round-robin)
MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth0
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 84:8f:69:50:a8:ae
Slave queue ID: 0

Slave Interface: eth1
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 84:8f:69:50:a8:b0
Slave queue ID: 0
```

Linux FCoE

FCoE - Fibre Channel Over Ethernet yeni şəbəkə texnologiyasıdır hansı ki, Fibre Channel çərçivələrini Ethernet şəbəkəsi üzərindən enkapsulyasiya edir. Bu 10 Gigabit Ethernet(və daha da çox) şəbəkəsi üzərindən Fibre Channel protokolun istifadə edilməsinə şərait yaradır.

Öncə RHEL-in rəsmi 6.5-ci versiya diskini serverimizə mount edirik.

```
mkdir /media/CentOS/                # Bu ünvan Mount edəcəyimiz RHEL DVD
                                    diski üçündür.

mount /dev/sr0 /media/CentOS/       # DVD diski öncə yaratdığımız qovluğa
                                    mount edirik.
```

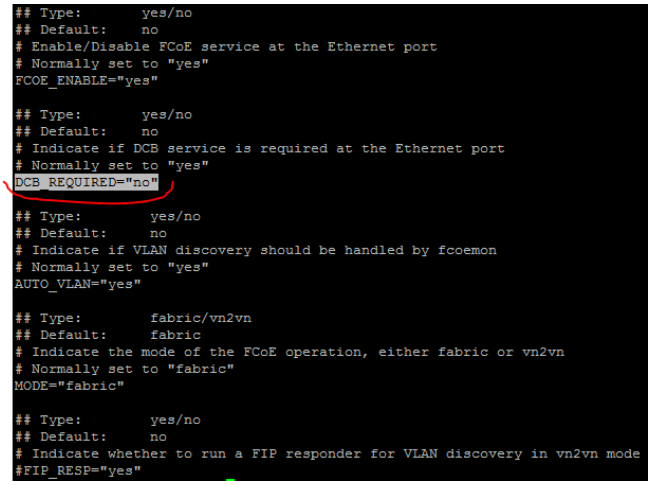
Sonra **/etc/yum.repos.d/CentOS-Media.repo** adlı fayl yaradıb içinə aşağıdakı kontenti əlavə edirik:

```
[c5-media]
name=CentOS-$releasever - Media
baseurl=file:///media/CentOS/
        file:///media/cdrom/
        file:///media/cdrecorder/
gpgcheck=0
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-beta
```

```
yum update                                  # Reposları yeniləyirik.
yum install fcoe-utils.x86_64 fcoe-target-utils.noarch      # Lazımı
                                                    paketləri yükləyirik.

cd /etc/fcoe/                               # FCoE ünvanına daxil oluruq

[root@hp_proliant fcoe]# ethtool eth2       # FC adapterimizi tapırıq
Settings for eth2:
        Supported ports: [ FIBRE ]
        Supported link modes:   1000baseT/Full
                                10000baseT/Full
        Supported pause frame use: Symmetric Receive-only
        Supports auto-negotiation: Yes
        Advertised link modes:  1000baseT/Full
                                10000baseT/Full
        Advertised pause frame use: Symmetric Receive-only
        Advertised auto-negotiation: Yes
        Link partner advertised link modes:  1000baseT/Full
                                             10000baseT/Full
        Link partner advertised pause frame use: Symmetric
        Link partner advertised auto-negotiation: Yes
        Speed: 8000Mb/s
        Duplex: Full
        Port: FIBRE
        PHYAD: 1
        Transceiver: internal
```

```
        Auto-negotiation: on
        Supports Wake-on: g
        Wake-on: g
        Current message level: 0x00000000 (0)

        Link detected: yes

[root@hp_proliant fcoe]# ethtool eth3           # Ikinci FCoE kartı tapırıq
Settings for eth3:
        Supported ports: [ FIBRE ]
        Supported link modes:   1000baseT/Full
                                10000baseT/Full
        Supported pause frame use: Symmetric Receive-only
        Supports auto-negotiation: Yes
        Advertised link modes:  1000baseT/Full
                                10000baseT/Full
        Advertised pause frame use: Symmetric Receive-only
        Advertised auto-negotiation: Yes
        Link partner advertised link modes:  1000baseT/Full
                                             10000baseT/Full
        Link partner advertised pause frame use: Symmetric
        Link partner advertised auto-negotiation: Yes
        Speed: 8000Mb/s
        Duplex: Full
        Port: FIBRE
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: g
        Wake-on: g
        Current message level: 0x00000000 (0)

        Link detected: yes

[root@hp_proliant fcoe]# cp cfg-ethx cfg-eth2   # eth2 şəbəkə kartı üçün FCoE
                                                quraşdırmasını nüsxələyirik

[root@hp_proliant fcoe]# cp cfg-ethx cfg-eth3   # eth3 şəbəkə kartı üçün FCoE
                                                quraşdırmasını nüsxələyirik
```

Sonra həm **/etc/fcoe/cfg-eth2** quraşdırma faylında və həm də **/etc/fcoe/cfg-eth3** quraşdırma faylında **DCB_REQUIRED="no"** edirik(Aşağıdakı şəkildəki kimi):



Qeyd: Əgər siz DCB(Data Center Bridging - QoS üçün istifadə edilir)-i FCoE NetCard-larda aktivləşdirmək istəyirsinizsə, onda hər bir FCoE card-ın quraşdırmasında faylında(Misal üçün: **/etc/fcoe/cfg-eth2**) **DCB_REQUIRED="yes"** etmək lazımdır və aşağıdakı əmrlə aktivləşdirmək lazımdır:

dcbtool sc eth2 dcb on
dcbtool sc eth2 app:fcoe e:1

Ardıncada **/etc/fcoe/config** config faylında **SUPPORTED_DRIVERS** dəyişənini **"fcoe bnx2fc"** edirik(Aşağıdakı şəkildəki kimi):

```
[root@hp_proliant fcoe]# service lldpad start          # lldpad-i start edirik
Starting lldpad:                                            [  OK  ]

[root@hp_proliant fcoe]# service fcoe start            # fcoe-ni start edirik
Starting FCoE initiator service:                            [  OK  ]


[root@hp_proliant fcoe]# fcoeadm -i                    # FCoE kartlara baxırıq
    Description:      NetXtreme II BCM57810 10 Gigabit Ethernet
    Revision:         10
    Manufacturer:     Broadcom Corporation
    Serial Number:    9CB6549A5270
    Driver:           bnx2x 1.710.10
    Number of Ports:  1

        Symbolic Name:     bnx2fc (Broadcom BCM57810) v2.4.2e over eth3
        OS Device Name:    host2
        Node Name:         0x50060B0000C26613
        Port Name:         0x50060B0000C26612
        FabricName:        0x10000027F8DBEC63
        Speed:             Unknown
        Supported Speed:   1 Gbit, 10 Gbit
        MaxFrameSize:      2048
        FC-ID (Port ID):   0x010203
        State:             Online

        Symbolic Name:     bnx2fc (Broadcom BCM57810) v2.4.2e over eth2
```

```
        OS Device Name:     host3
        Node Name:          0x50060B0000C26611
        Port Name:          0x50060B0000C26610
        FabricName:         0x10000027F8DBF943
        Speed:              Unknown
        Supported Speed:    1 Gbit, 10 Gbit
        MaxFrameSize:       2048
        FC-ID (Port ID):    0x010203
        State:              Online
```

```
[root@hp_proliant fcoe]# cat /proc/partitions   # Partitionları yoxlayırıq
major minor  #blocks  name
   8        0  292935982 sda
   8        1     512000 sda1
   8        2  292422656 sda2
 253        0   52428800 dm-0
 253        1   29290496 dm-1
 253        2  210702336 dm-2
   8       16   10485760 sdb
   8       32   10485760 sdc
 253        3   10485760 dm-3
   8       48   10485760 sdd
   8       64   10485760 sde
```

Hər iki daemon, yəni **lldpad** və **fcoe**-ni startup-a əlavə edirik:

```
[root@hp_proliant fcoe]# chkconfig lldpad on
[root@hp_proliant fcoe]# chkconfig fcoe on
```

Öncədən Multipath-i Linux maşınımıza yükləyirik.

```
[root@hp_proliant fcoe]# yum install device-mapper-multipath.x86_64 device-
mapper-multipath-libs.x86_64 -y
```

Multipath ünvanların tapılmasını işə salırıq

```
[root@hp_proliant fcoe]# mpathconf --enable
[root@hp_proliant fcoe]# mpathconf --find_multipaths y
```

Multipath daemon-u işə salırıq

```
[root@hp_proliant fcoe]# /etc/init.d/multipathd start
[root@hp_proliant fcoe]# chkconfig multipathd on     # Startup-a əlavə edirik
[root@hp_proliant fcoe]# multipath -ll      # Multipath-da diskimizə baxırıq.
mpathb (360002ac0000000000000000300009a26) dm-3 3PARdata,VV
size=10G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 2:0:0:0 sdb 8:16 active ready running
  |- 2:0:1:0 sdc 8:32 active ready running
  |- 3:0:0:0 sdd 8:48 active ready running
  `- 3:0:1:0 sde 8:64 active ready running
```

Firewall və Selinux-u sondürürük

```
[root@hp_proliant fcoe]# chkconfig --level 0123456 iptables off
[root@hp_proliant fcoe]# chkconfig --level 0123456 ip6tables off
```

/etc/selinux/config faylında **SELINUX=disabled** edirik.

```
[root@hp_proliant fcoe]# reboot            # Hər hal üçün sonda reboot edirik
```

Troubleshoot üçün bəzi əmrləri sınaqdan keçirək

```
[root@hp_proliant fcoe]# yum -y install lsscsi.x86_64        # Lazımı paketi
                                                             yükləyirik

[root@rac ~]# lsscsi | grep disk                          # Diskləri yoxlayırıq
[0:0:0:0]    disk    HP       LOGICAL VOLUME   5.22  /dev/sda
[1:0:0:0]    disk    3PARdata VV               3123  /dev/sdb
[1:0:1:0]    disk    3PARdata VV               3123  /dev/sdc
[2:0:0:0]    disk    3PARdata VV               3123  /dev/sdd
[2:0:1:0]    disk    3PARdata VV               3123  /dev/sde
```

FCoE ilə gələn disklərimizə baxırıq

```
[root@rac ~]# fcoeadm -t
    Interface:        eth3
    Roles:            FCP Target
    Node Name:        0x2FF70002AC009A26
    Port Name:        0x20120002AC009A26
    Target ID:        0
    MaxFrameSize:     2048
    OS Device Name:   rport-1:0-3
    FC-ID (Port ID):  0x010400
    State:            Online

    LUN ID  Device Name   Capacity   Block Size  Description
    ------  -----------  ----------  ----------  ----------------------------
         0  /dev/sdb      10.00 GiB      512     3PARdata VV (rev 3123)

    Interface:        eth3
    Roles:            FCP Target
    Node Name:        0x2FF70002AC009A26
    Port Name:        0x21120002AC009A26
    Target ID:        1
    MaxFrameSize:     2048
    OS Device Name:   rport-1:0-6
    FC-ID (Port ID):  0x010500
    State:            Online

    LUN ID  Device Name   Capacity   Block Size  Description
    ------  -----------  ----------  ----------  ----------------------------
         0  /dev/sdc      10.00 GiB      512     3PARdata VV (rev 3123)

    Interface:        eth3
    Roles:            FCP Target
    Node Name:        0x500143801603302F
```

```
    Port Name:          0x5001438016033030
    Target ID:          2
    MaxFrameSize:       2048
    OS Device Name:     rport-1:0-7
    FC-ID (Port ID):    0x010600
    State:              Online


    Interface:          eth2
    Roles:              FCP Target
    Node Name:          0x2FF70002AC009A26
    Port Name:          0x21110002AC009A26
    Target ID:          0
    MaxFrameSize:       2048
    OS Device Name:     rport-2:0-4
    FC-ID (Port ID):    0x010500
    State:              Online

    LUN ID  Device Name   Capacity   Block Size  Description
    ------  -----------  ----------  ----------  ----------------------------
         0  /dev/sdd      10.00 GiB      512     3PARdata VV (rev 3123)

    Interface:          eth2
    Roles:              FCP Target
    Node Name:          0x2FF70002AC009A26
    Port Name:          0x20110002AC009A26
    Target ID:          1
    MaxFrameSize:       2048
    OS Device Name:     rport-2:0-6
    FC-ID (Port ID):    0x010400
    State:              Online

    LUN ID  Device Name   Capacity   Block Size  Description
    ------  -----------  ----------  ----------  ----------------------------
         0  /dev/sde      10.00 GiB      512     3PARdata VV (rev 3123)

    Interface:          eth2
    Roles:              FCP Target
    Node Name:          0x500143801603302C
    Port Name:          0x500143801603302D
    Target ID:          2
    MaxFrameSize:       2048
    OS Device Name:     rport-2:0-7
    FC-ID (Port ID):    0x010600
    State:              Online
```

LUN-larımıza baxaq:

```
[root@rac ~]# fcoeadm -l
    Interface:          eth3
```

```
Roles:            FCP Target
Node Name:        0x2FF70002AC009A26
Port Name:        0x20120002AC009A26
Target ID:        0
MaxFrameSize:     2048
OS Device Name:   rport-1:0-3
FC-ID (Port ID):  0x010400
State:            Online

LUN #0 Information:
    OS Device Name:     /dev/sdb
    Description:        3PARdata VV (rev 3123)
    Ethernet Port FCID: 0x010218
    Target FCID:        0x010400
    Target ID:          0
    LUN ID:             0
    Capacity:           10.00 GiB
    Capacity in Blocks: 20971520
    Block Size:         512 bytes
    Status:             Attached

Interface:        eth3
Roles:            FCP Target
Node Name:        0x2FF70002AC009A26
Port Name:        0x21120002AC009A26
Target ID:        1
MaxFrameSize:     2048
OS Device Name:   rport-1:0-6
FC-ID (Port ID):  0x010500
State:            Online

LUN #0 Information:
    OS Device Name:     /dev/sdc
    Description:        3PARdata VV (rev 3123)
    Ethernet Port FCID: 0x010218
    Target FCID:        0x010500
    Target ID:          1
    LUN ID:             0
    Capacity:           10.00 GiB
    Capacity in Blocks: 20971520
    Block Size:         512 bytes
    Status:             Attached

Interface:        eth3
Roles:            FCP Target
Node Name:        0x500143801603302F
Port Name:        0x5001438016033030
Target ID:        2
MaxFrameSize:     2048
OS Device Name:   rport-1:0-7
FC-ID (Port ID):  0x010600
State:            Online
```

```
Interface:        eth2
Roles:            FCP Target
Node Name:        0x2FF70002AC009A26
Port Name:        0x21110002AC009A26
Target ID:        0
MaxFrameSize:     2048
OS Device Name:   rport-2:0-4
FC-ID (Port ID):  0x010500
State:            Online

LUN #0 Information:
    OS Device Name:     /dev/sdd
    Description:        3PARdata VV (rev 3123)
    Ethernet Port FCID: 0x010218
    Target FCID:        0x010500
    Target ID:          0
    LUN ID:             0
    Capacity:           10.00 GiB
    Capacity in Blocks: 20971520
    Block Size:         512 bytes
    Status:             Attached

Interface:        eth2
Roles:            FCP Target
Node Name:        0x2FF70002AC009A26
Port Name:        0x20110002AC009A26
Target ID:        1
MaxFrameSize:     2048
OS Device Name:   rport-2:0-6
FC-ID (Port ID):  0x010400
State:            Online

LUN #0 Information:
    OS Device Name:     /dev/sde
    Description:        3PARdata VV (rev 3123)
    Ethernet Port FCID: 0x010218
    Target FCID:        0x010400
    Target ID:          1
    LUN ID:             0
    Capacity:           10.00 GiB
    Capacity in Blocks: 20971520
    Block Size:         512 bytes
    Status:             Attached

Interface:        eth2
Roles:            FCP Target
Node Name:        0x500143801603302C
Port Name:        0x500143801603302D
Target ID:        2
MaxFrameSize:     2048
OS Device Name:   rport-2:0-7
FC-ID (Port ID):  0x010600
State:            Online
```

FCoE kartımızda statistikalara baxaq:

```
[root@rac ~]# fcoeadm -s eth2
eth2    interval: 1                                    Err  Inv  IvTx Link Cntl Input
Input     Output    Output
Seconds TxFrames  TxBytes      RxFrames  RxBytes        Frms CRC  Byte Fail Reqs
Requests  MBytes    Requests  MBytes
------- --------- ---------- --------- -------------- ---- ---- ---- ---- ---- -----
---- --------- --------- --------
0       1387      125068       2552      2417480        0    0    0    0    7    1125
2         0         0
1       1387      125068       2552      2417480        0    0    0    0    7    1125
2         0         0
```

FCoE ilə ping edək:

```
[root@rac ~]# fcping -c3 -h eth3 -F 0x010218
Maximum ECHO data allowed by the Fabric (0xfffffd) : 2108 bytes.
Maximum ECHO data allowed by the Source (0x010218) : 2044 bytes.
Maximum ECHO data allowed by the Target (0x010218) : 32 bytes.
Maximum ECHO data requested from user input (-s) : 32 (default 32) bytes.
Actual FC ELS ECHO data size used : 32 bytes.
Actual FC ELS ECHO payload size used : 36 bytes (including 4 bytes ECHO
command).
Sending FC ELS ECHO from 0x10218 (fc_host1) to 0x10218:
echo    1 accepted                          0.225 ms
echo    2 accepted                          0.222 ms
echo    3 accepted                          0.225 ms
3 frames sent, 3 received 0 errors, 0.000% loss, avg. rt time 0.224 ms
```

-c – Göndəriləcək ping sayı(Bizim halda 3 ədəd)
-h - hansı FCoE kartımızın üzərindən
-F - FC-ID (Bunu **fcoeadm -i** əmri ilə əldə edə bilərsiniz.)

FCoE-nin digər maşınlara paylaşılması.

Siz FCoE-ni özünüz itifadə etdiyiniz kimi, başqa maşınlara da paylaşa bilərsiniz.

```
yum install fcoe-target-utils       # Öncə lazımı paketi yükləyirik.
service fcoe-target start           # Servisi işə salırıq

chkconfig fcoe-target on            # Servisi startup-a əlavə edirik.

targetcli               # Əmri daxil edirik ki, quraşdırma faylımızı yaradaq
```

Avadanlığın təyinatı aşağıdakı kimi olur:

```
      backstores/block create example1 /dev/sda4
```

example1 adlı **/dev/sda4** diskini yaradiriq.

Digər avadanlığı təyin edirik:

```
backstores/fileio create example2 /srv/example2.img 100M    # 100M-baytlıq
                                                              img faylını
                                                              example2 adla
                                                              paylaşırıq

tcm_fc/ create 00:11:22:33:44:55:66:77    # FCoE interfeysde FCoE target
                                                    yaradırıq
cd tcm_fc/00:11:22:33:44:55:66:77         # target instansin xəritələnməsi
luns/ create /backstores/fileio/example2

acls/ create 00:99:88:77:66:55:44:33      # FCoE initiator-a yetkini veririk.
```

Multipath disklərin işlək vəziyyətdə genişləndirilməsi

Multipath I/O — bir neçə marşrutdan istifadə eləməklə, məlumatlar saxlanılan şəbəkə üzvlərinin qoşulması texnologiyasıdır. Məsələn, bir SCSI-disk iki SCSI-kontrollerə birləşdirilmiş ola bilər. Kontrollerlərdən biri, sıradan çıxdığı halda əməliyyat sistemi diskə giriş üçün digərindən istifadə edəcək. Bu arxitektura sistemin səhvə davamlılığını artırır və yüklənmənin bölüşdürülə bilməsinə şərait yaradır.

Unutmayın ki, işə başlamazdan əvvəl avadanlıq inzibatçısı həmin diski öz Management serverində öncədən artırmalıdır.

Serverimizdə olan Fibre Channel linklər üçün axtarış edirik:

```
echo "- - -" > /sys/class/scsi_host/host0/scan
echo "- - -" > /sys/class/scsi_host/host1/scan
```

Yüxarıdan əmrlə eyni işi görür sadəcə, burda dövr bütün işi avtomatlaşdırır:

```
for host in `ls /sys/class/fc_host`; do
      echo "- - -" > /sys/class/scsi_host/${HOST}/scan
done
```

Genişləndiriləcək diskimizi tapırıq. Genişləndiriləcək diskimiz **'/dev/mapper/mpathg'** adindadir. Ancaq onun böyümesindən öncə, bizə gələn kanallar üzərində olan disklərin həcmini **resize** etməliyik və sonra da **MPATH** diski **resize** etməliyik

```
fdisk -l | grep Disk | grep -v identifier
```

```
Disk /dev/sdf: 1610.6 GB, 1610612736000 bytes
Disk /dev/sdg: 1610.6 GB, 1610612736000 bytes
Disk /dev/mapper/mpathg: 1610.6 GB, 1610612736000 bytes
Disk /dev/sdh: 1610.6 GB, 1610612736000 bytes
Disk /dev/sdi: 1610.6 GB, 1610612736000 bytes
```

MPath diskimizin həcminə baxırıq və bir yerdə qeyd edirik ki, sonra dəyişmiş həcmi görə bilək.

```
blockdev --getsz /dev/mapper/mpathg
```

Disklərimizin Optika ilə gələn kanallarını yenidən scan edirik

```
echo 1 > /sys/block/sdf/device/rescan
echo 1 > /sys/block/sdg/device/rescan
echo 1 > /sys/block/sdh/device/rescan
echo 1 > /sys/block/sdi/device/rescan
```

Yenidən disklərimizə baxıb görürük ki, fiziki disklərdə həcm artıb, ancaq Mpath diskində həcm köhnə olaraq qalır.

```
fdisk -l | grep Disk | grep -v identifier
```

```
Disk /dev/sdf: 1825.4 GB, 1825361100800 bytes
Disk /dev/sdg: 1825.4 GB, 1825361100800 bytes
Disk /dev/mapper/mpathg: 1610.6 GB, 1610612736000 bytes
Disk /dev/sdh: 1825.4 GB, 1825361100800 bytes
Disk /dev/sdi: 1825.4 GB, 1825361100800 bytes
```

Multipath diskin özünü **resize** edib **reconfigure** edirik, sonra da menyudan **exit** əmri ilə çıxırıq.

multipathd -k 'resize map /dev/mapper/mpathg'
multipathd> **reconfigure**
ok

Yenidən Disklərimiz-də axtarış edib **mpathg** diskinin həcminə göz yetirib görürük ki, həcm artiq **1825.4Gb**-dır.
fdisk -l | grep Disk | grep -v identifier | grep mpathg
Disk /dev/mapper/mpathg: **1825**.4 GB, 1825361100800 bytes

Əməliyyat sisteminə **mpathg** diskində olan partition **table**-ın dəyişməsi haqqında məlumat ötürürük.
partprobe /dev/mapper/mpathg

Phisical Volume-mu resize edirik.
pvresize /dev/mapper/mpathg

Artırılan həcmdən **232GB** həcmi **vg-1TB** Volume group-unda olan **u02** LVOL-una artırırıq. Nəticədə aşağıdakı jurnallarda görünən formada olmalıdır
l**vextend -L +300G /dev/vg-1TB/u01 /dev/mapper/mpathg**

Extending logical volume **u01 to 1.07 TiB**
Logical volume **u01 successfully resized**

Sonda mövcud olan fayl sistemə toxunmadan yeni yaranan həcmə fayl sistemi artırırq.
resize2fs -p /dev/vg-1TB/u01

BÖLÜM 10

Korporativ şəbəkədə yazışma sistemi

- OpenFire XMMP serverin qurulması
- OpenFIRE ilə Active Directory inteqrasiyası

Hər bir korporativ şəbəkənin daxili yazışma sistemi olmalıdır. Bu yazışma sistemi istifadəçiləri həmin şəbəkə daxilində məlumat ötürülmələrində təhlükəsiz edir, şirkətin özünü təhlükəsiz edir və istifadəçilərin arasında danışıqları jurnallayır. Şirkət daxilində olan daxili yazışma sistemi domain controllerlə inteqrasiya qabiliyyətinə malik olmalıdır ki, istifadəçilər hər bir proqram təminatı üçün, fərqli istifadəçi adı və şifrə daxil etməsinlər. Başlığımızda bunların hamısı müzakirə ediləcək.

OpenFire XMMP serverin qurulması

OpenFire – JAVA da yazılmış Jabber/XMPP serverdir. İkili lisenziya altında işləyir. Həm pulsuz proqram təminatıdır və həmdə rəsmi dəstəyi mövcuddur. İdarəetmə üçün WEB panelə sahibdir, 9090(http) ve 9091(https) portlar üzərindən işləyir. Pluginləri(genişlənmələr), SSL/TLS dəstəkləyir, JDBC vasitəsilə verilənlər bazasına qoşula bilir(Oracle, MSSQL, PostgreSQL, DB2, Sybase ASE, MySQL və ya daxili verilənlər bazası HSQLDB), LDAP-a qoşula və qruplara görə süzgəcdən keçirə bilir, digər mənbələrə əsaslanaraq istifadəçi qeydiyyatını aparmaq və fərqli dillərin dəstəklənməsi imkanına sahibdir. İdarə edilməsinin əksər hissəsi WEB interfeys vasitəsilə edilir.
Rəsmi saytı http://www.igniterealtime.org/ .

Aşağıdakı funksionallıqları mövcuddur:

- WEB ilə idarəetmə
- Çoxlu pluginlərə sahibdir
- SSL/TLS dəstəkləyir
- Mesajların saxlanılması və istifadəçi detalları üçün, verilənlər bazaları ilə işləmə qabiliyyəti
- LDAP ilə əlaqə
- Istifadəçilərin kənar verilənlər vasitəsilə qeydiyyatdan keçirilmə imkanı
- Qeyri asılı paforma, təmiz JAVA
- Spark ilə tam inteqrasiya edilə bilir

Dəstəklənən klient proqramları:

- **Miranda IM**
- **QIP Infium**
- **Spark**
- **Trillian Pro**
- **Gaim**
- **Pandion**
- **Psi**
- **Exodus**
- **Pidgin**
- **Kopete**
- **Jitsi**

DNS serverinizdə aşağıdakılara uyğun olaraq SRV yazıları əlavə edin:

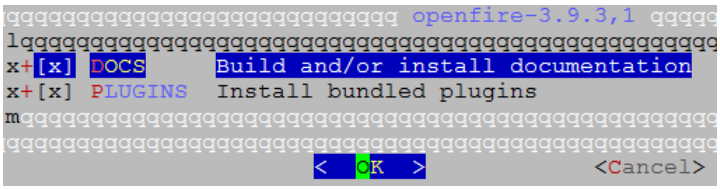
```
openfire                                IN A   94.20.81.149
_jabber._tcp.jabber.opensource.az.      IN SRV 0 0  5269
      jabber.opensource.az.
_xmpp-client._tcp.jabber.opensource.az. IN SRV 0 0  5222
      jabber.opensource.az.
_xmpp-server._tcp.jabber.opensource.az. IN SRV 0 0  5269
      jabber.opensource.az.
```

MySQL serverimizdə öncədən verilənlər bazası yaradaq ki, sonrakı quraşdırmalarımızda bizdən tələb ediləndə hazır olaq:

```
mysql -uroot -p
mysql> CREATE DATABASE openfire;
mysql> GRANT ALL PRIVILEGES ON openfire.* TO openfire@localhost IDENTIFIED BY
'openfiredbpass';
mysql> FLUSH PRIVILEGES;
```

Yükləməyə başlamazdan öncə mütləq portları yeniləmək lazımdır.

cd /usr/ports/net-im/openfire - Port ünvanına daxil oluruq
make config - Lazımi modulları seçirik

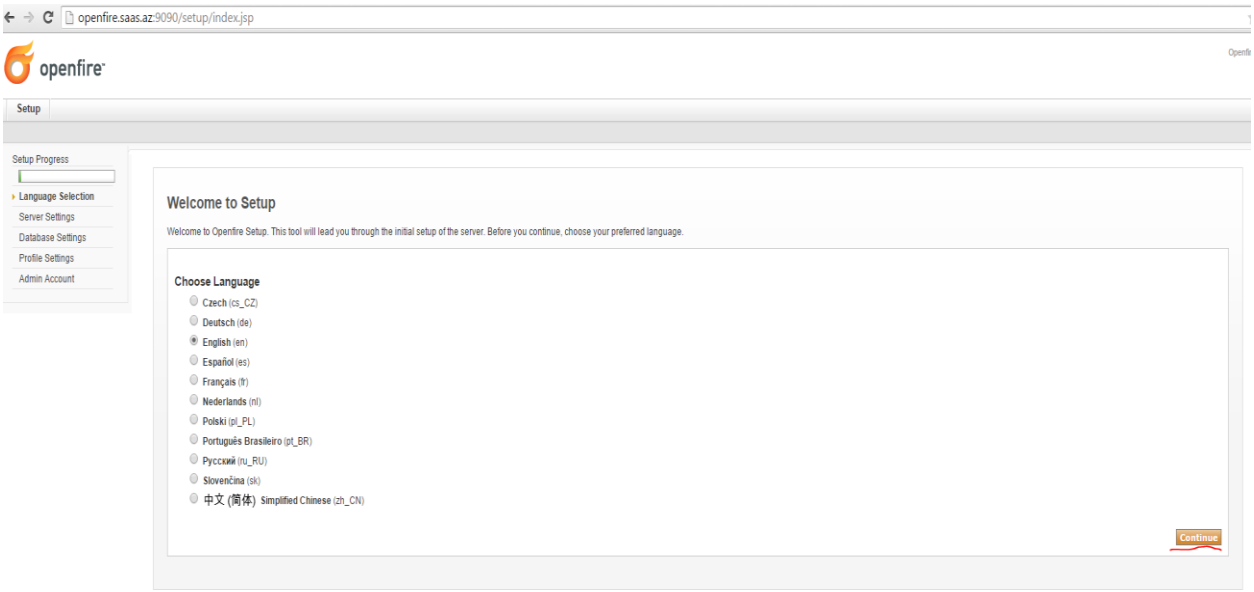


make install - Yükləyirik

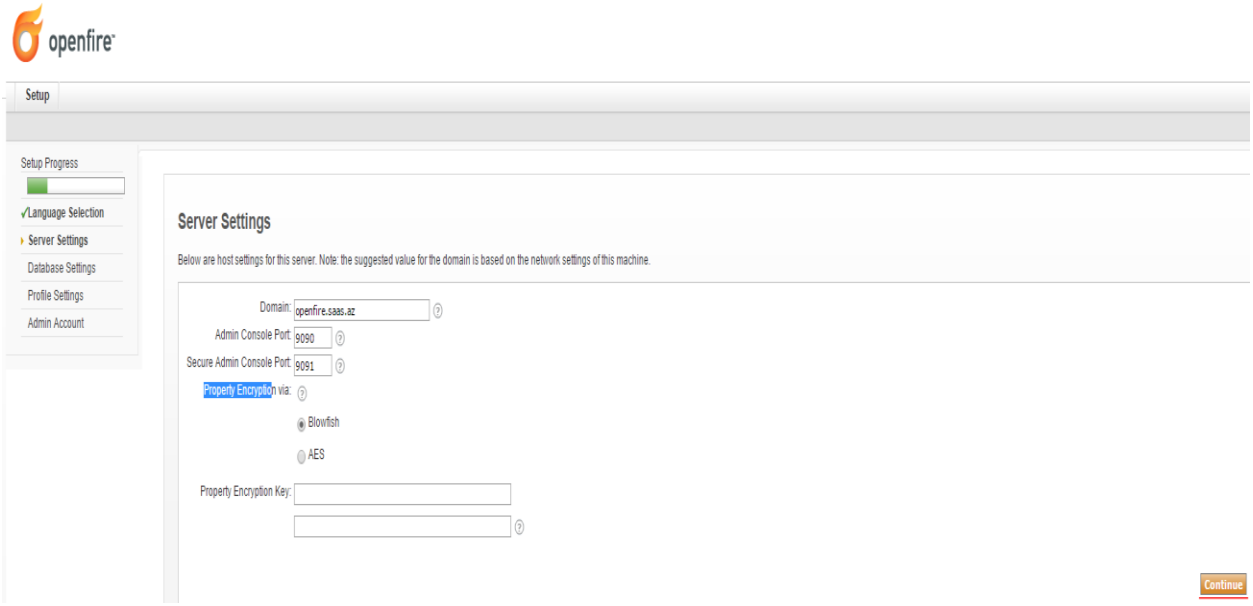
echo 'openfire_enable="YES"' >> /etc/rc.conf - OpenFire-ı StartUP-a əlavə edirik
/usr/local/etc/rc.d/openfire start - İşə salırıq

```
sockstat -l | grep openfire          - İşə düşməsini yoxlayırıq
openfire java       56187 26 tcp4   *:9090                 *:*
openfire java       56187 29 stream (not connected)
```

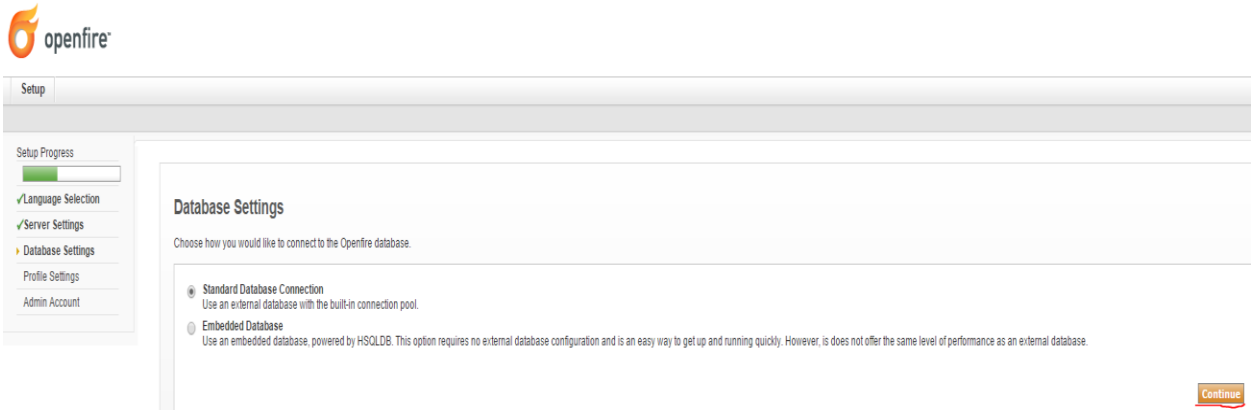
Ardınca http://openfire.opensource.az:9090 səhifəsinə daxil oluruq və aşağıdakı şəkili əldə etmiş olacayıq(English seçib **Continue** düyməsinə sıxın):



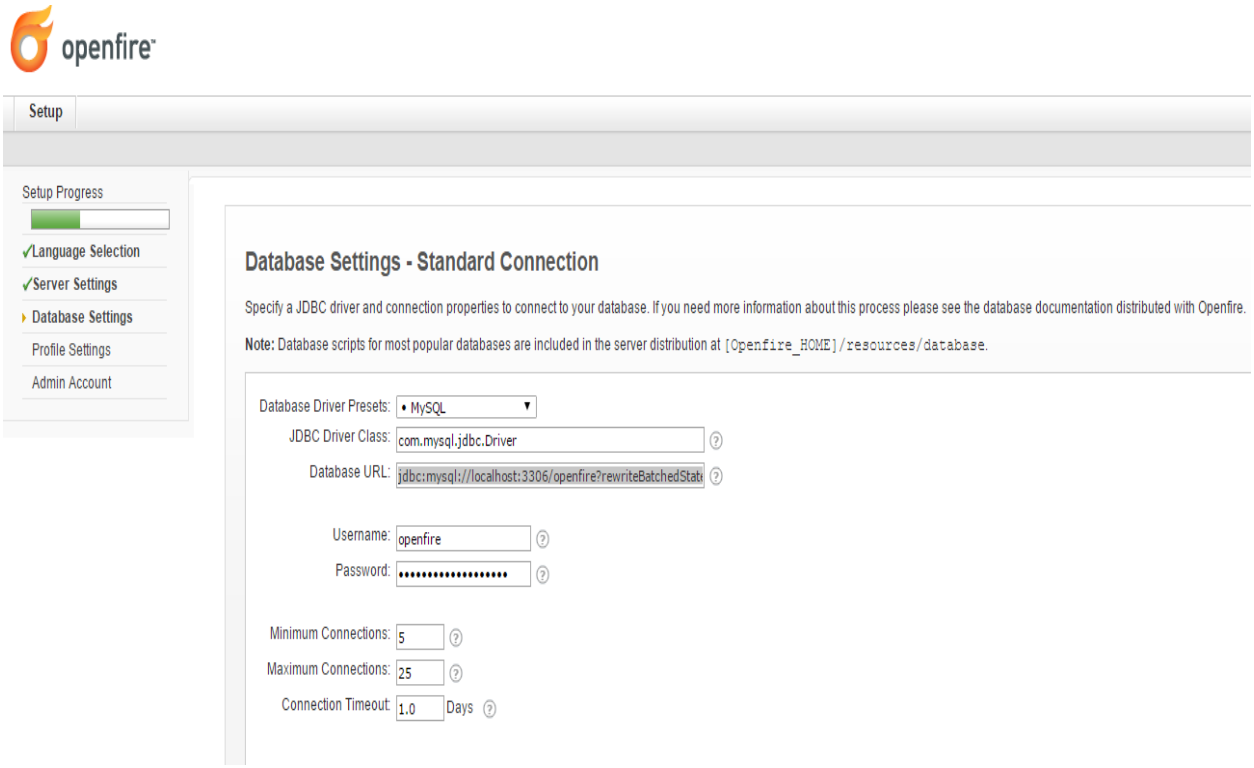
Göstərilən linkdə domain adı seçirik və **Continue** düyməsinə sıxırıq:



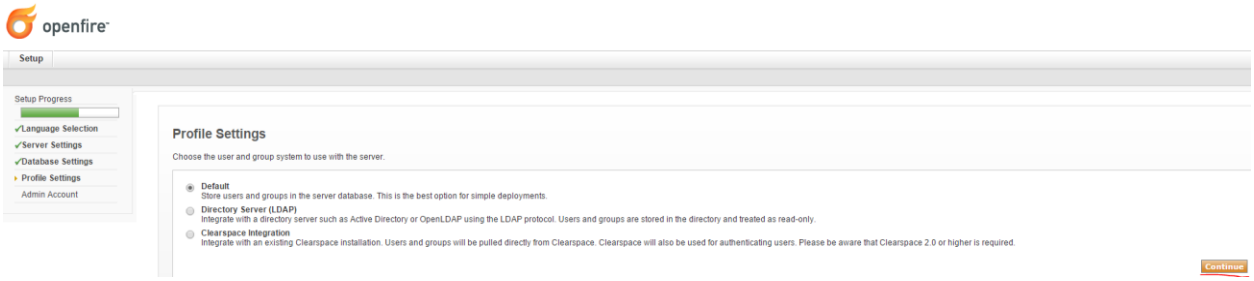
Ardınca kənar baza seçmək üçün **Standart Database Connection** seçirik və **Continue** düyməsinə sıxırıq:



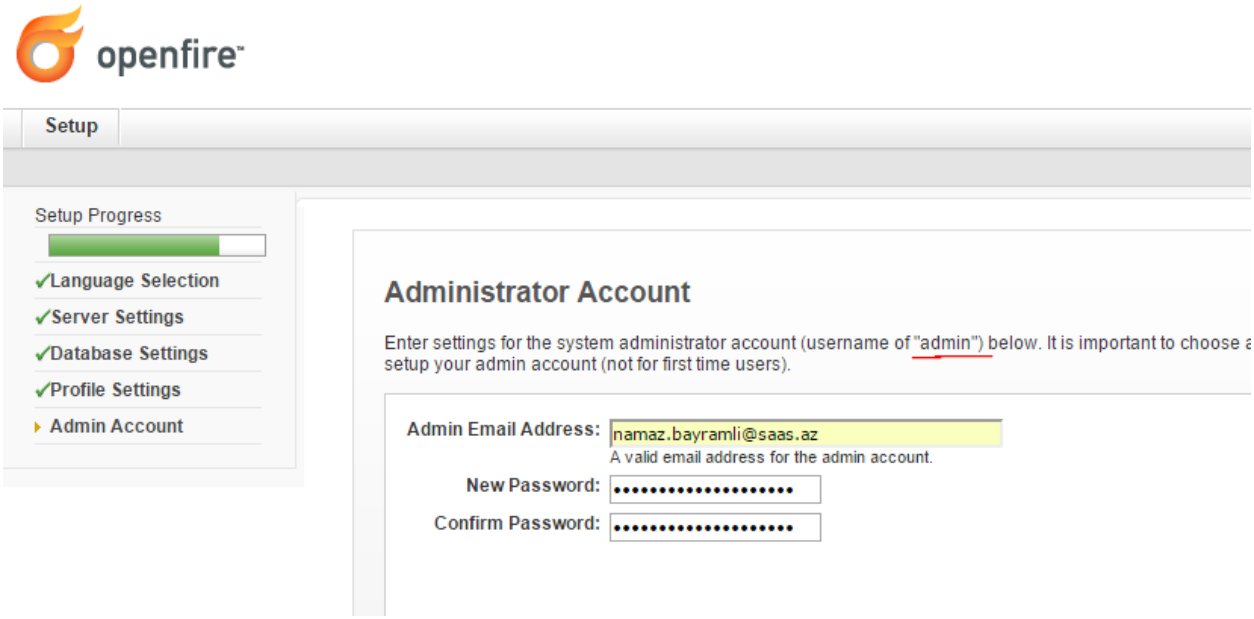
Sonra verilənlər bazası **MySQL** seçirik, Database URL: **jdbc:mysql://localhost:3306/openfire?rewriteBatchedStatements=true** sintaksisilə yazırıq və şəkildə göstərildiyi qaydada, openfire üçün MySQL istifadəçi adı və şifrəsini daxil edib **Continue** düyməsinə sıxırıq:



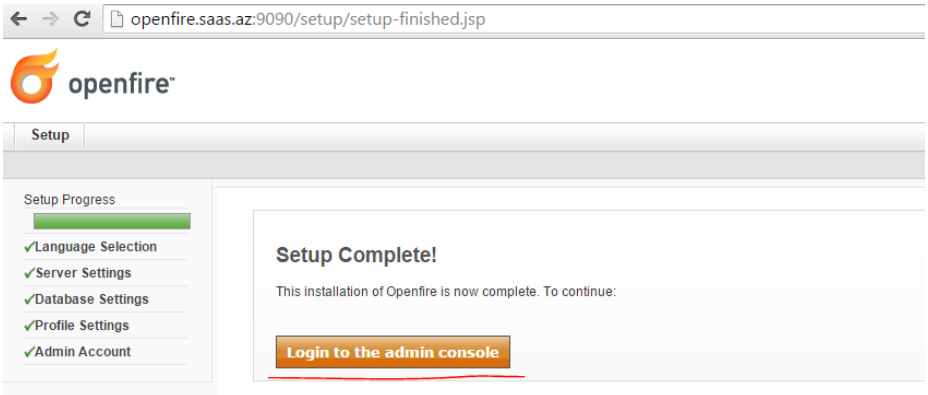
Növbəti şəkildə **Default** seçib **Continue** düyməsinə sıxırıq:



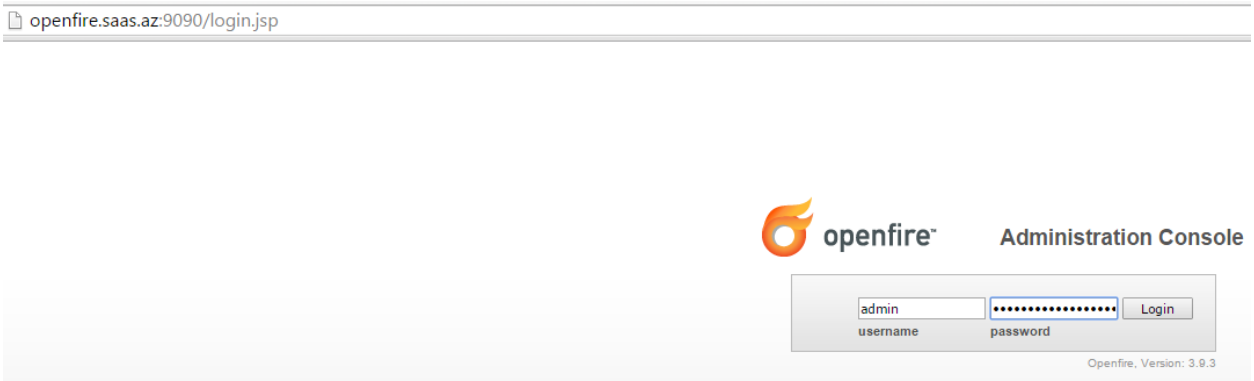
Açılan səhifədə, **admin** adlı həsab üçün email ünvanı və şifrəni iki dəfə daxil edib,(**admin** adlı istifadəçi adı və təyin etdiyimiz şifrə ilə gələcəkdə sistemimizdə daxil olacayıq. **admin** adı şərtdir) continue düyməsinə sıxırıq:



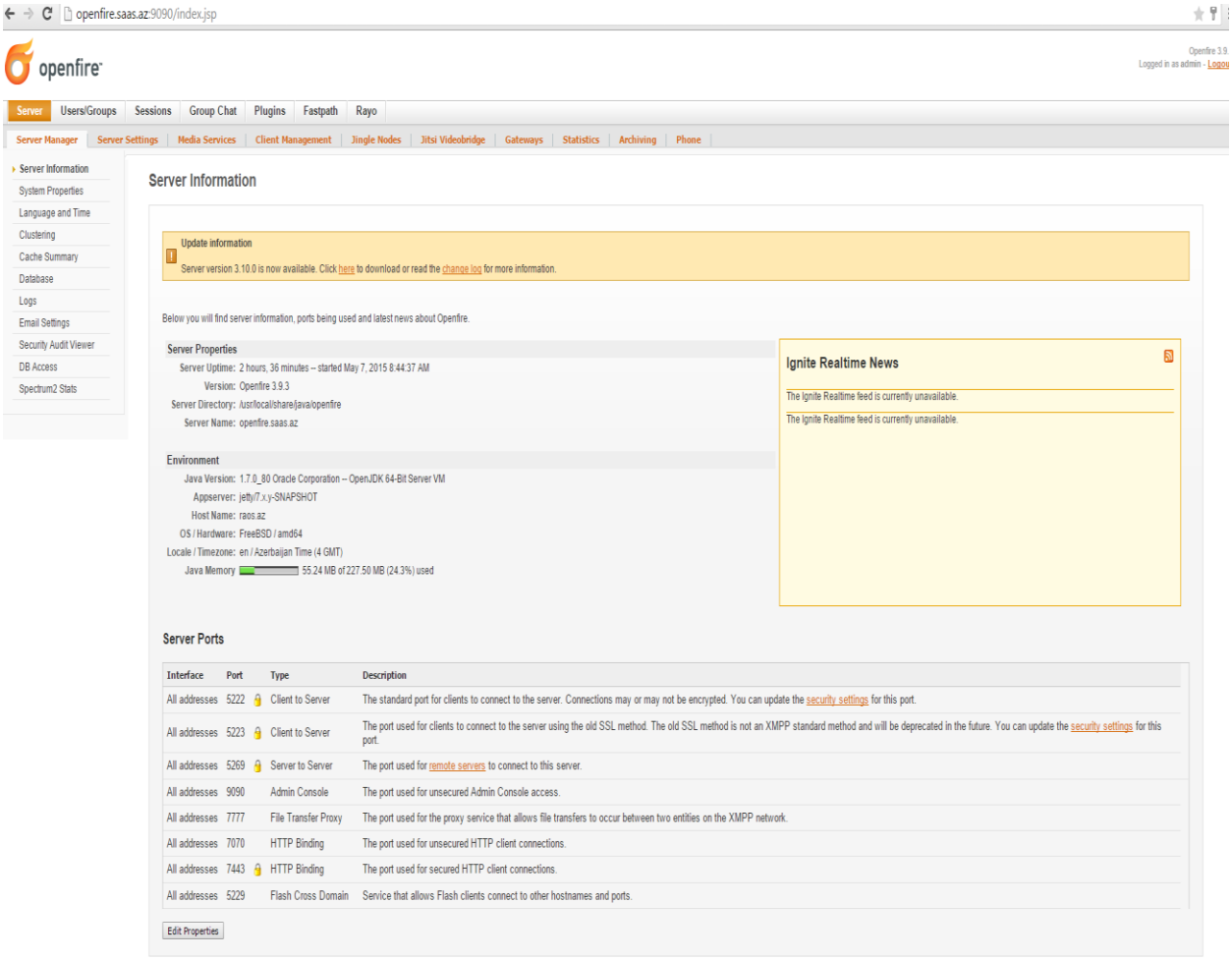
Sonda açılan səhifə aşağıdakı kimi olacaq və **Login to the admin console** düyməsinə sıxıb sistemimizə daxil oluruq:



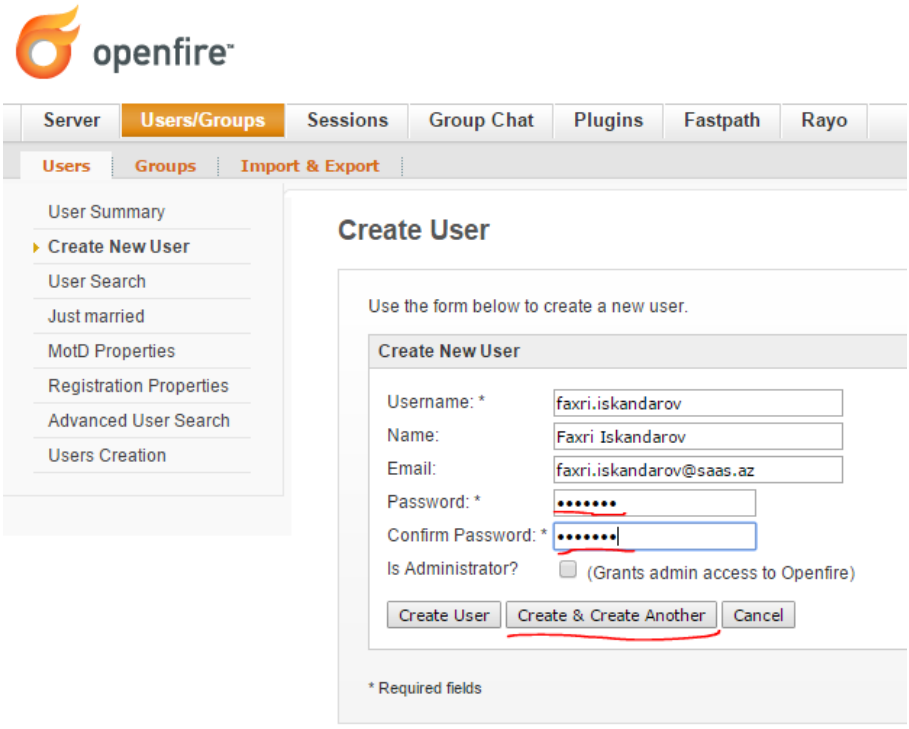
admin istifadəçi adı və biraz öncə yazdığımız şifrəni qeyd edərək sistemimizə daxil oluruq:



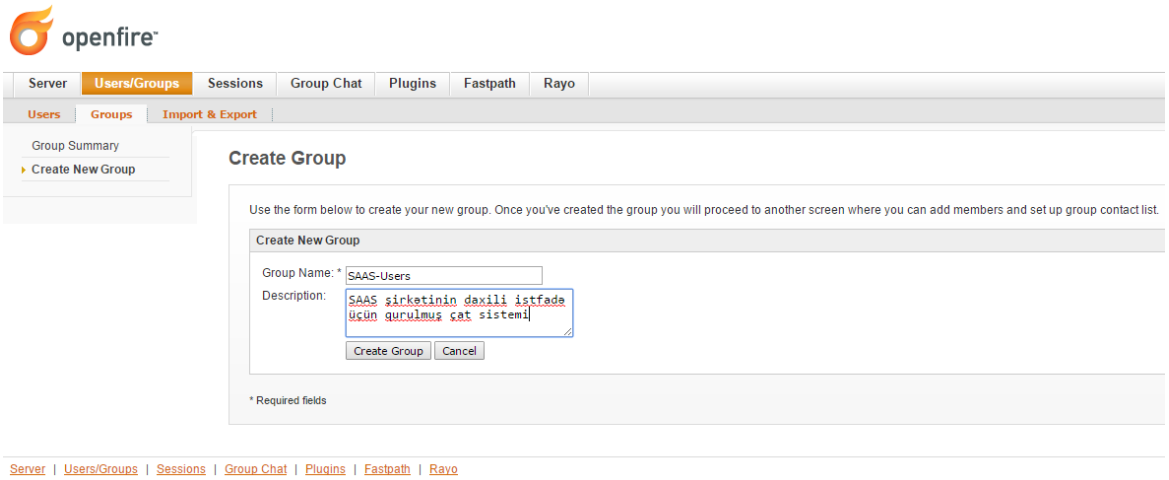
Sonda açılan pəncərə aşağıdakı kimi olacaq:



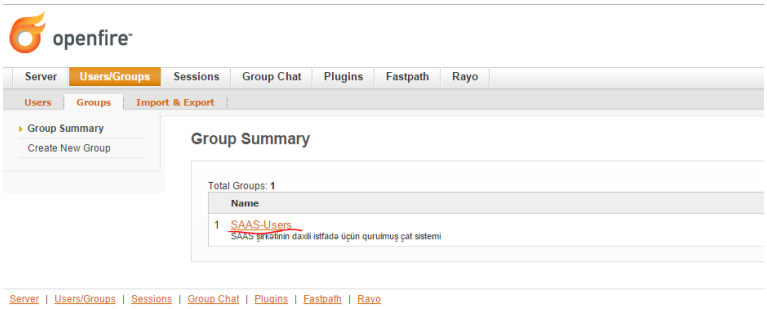
Aşağıdakı şablonda göstərildiyi kimi, bir neçə istifadəçi yaradaq:

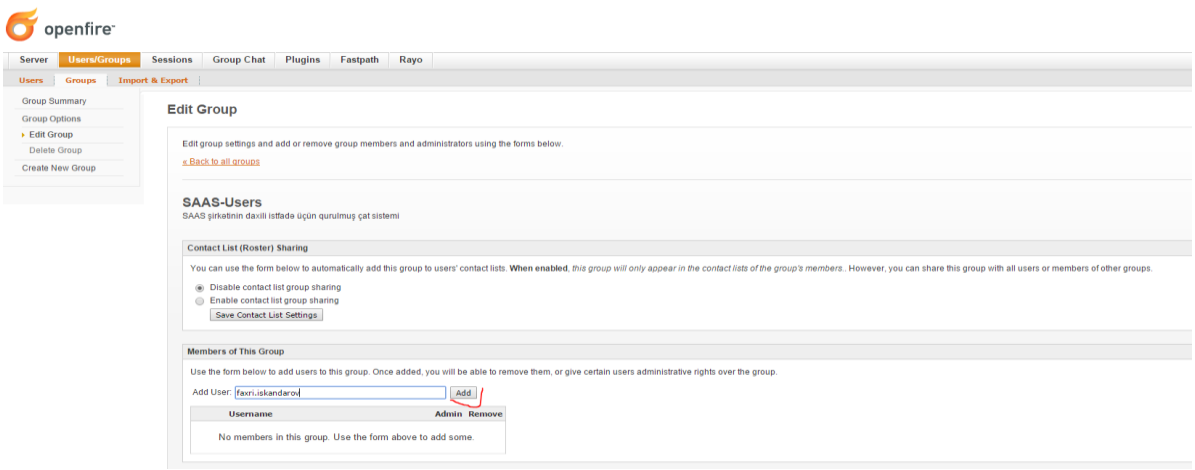


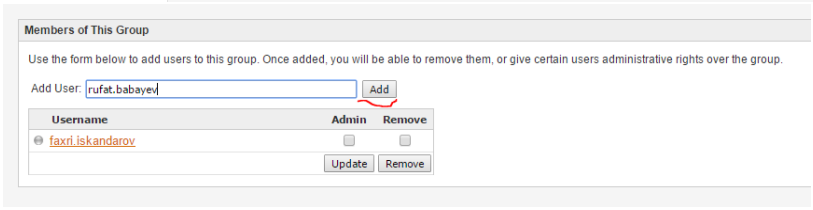
Sonra qrup əlavə edirik:



Sonra həmin qrupa daxil olaraq yaratdığımız istifadəçiləri həmin qrupa əlavə edirik:



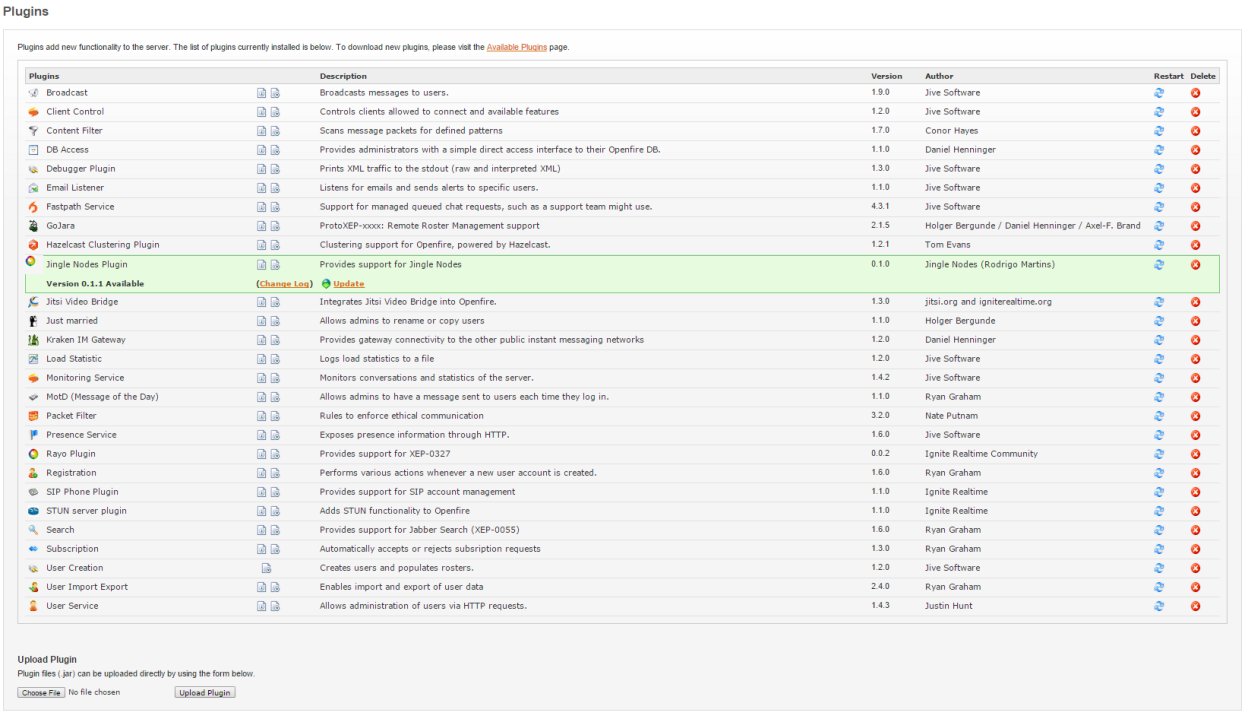




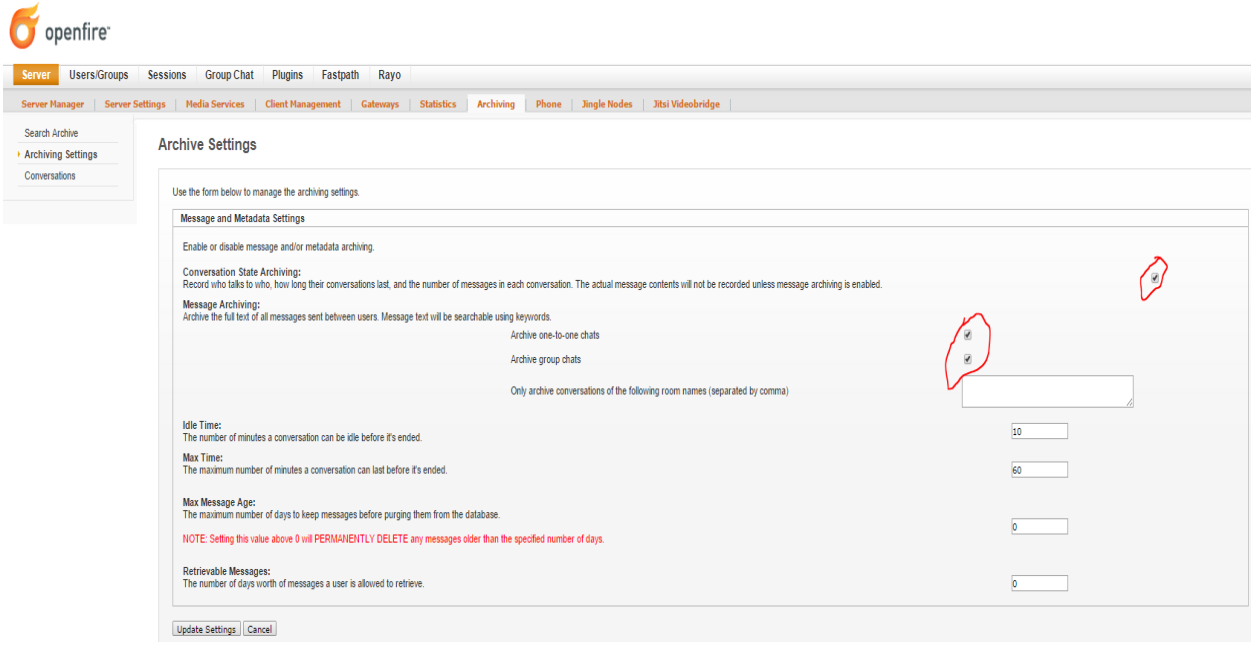
İndi isə clientin quraşdırılmasına baxaq. Bunun üçün öncə Spark client proqramını göstərilən linkdən http://www.igniterealtime.org/downloads/download-landing.jsp?file=spark/spark_2_7_0.exe dartırıq və yükləyirik.

Aşağıdakı şəkildə uyğun şəkildə **faxri.iskandarov** adlı istifadəçini quraşdırırıq:

Sonra Monitoring service pluginin yüklənməsini yoxlayırıq:



Ardınca **Server -> Archiving -> Archiving Settings** bölümünə daxil oluruq və daxili yazışmaların loqlanmasını aktivləşdiririk (Aşağıdakı şəkildəki kimi):



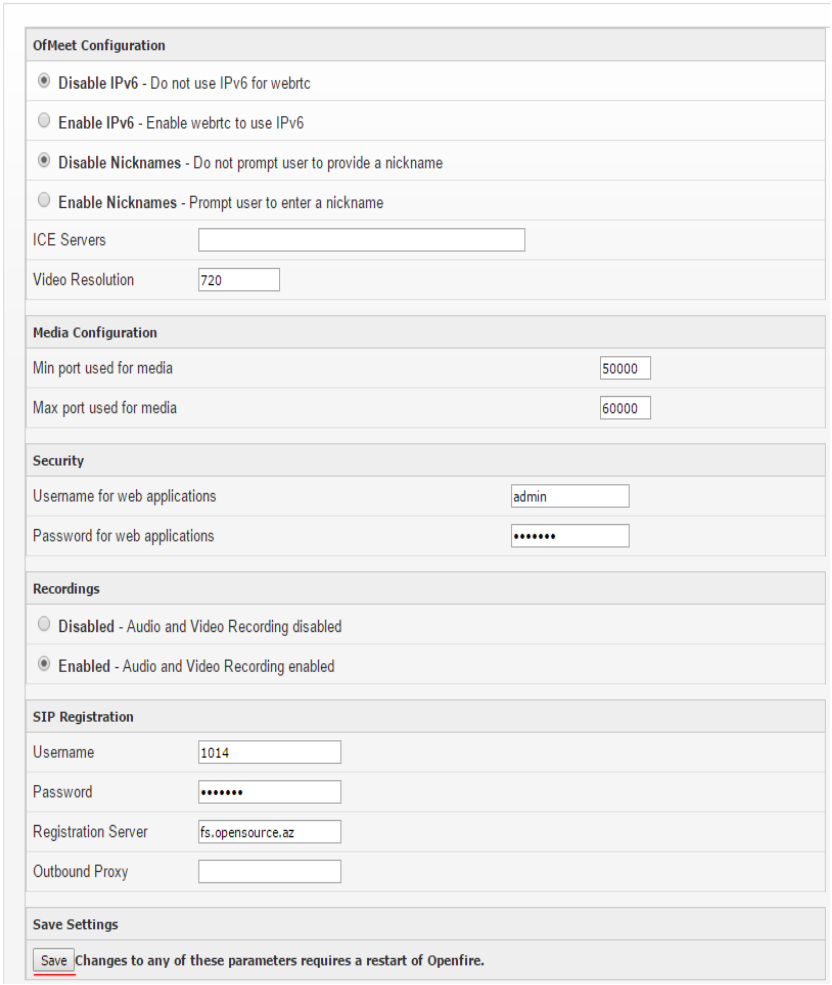
Hətta siz Online web vasitəsilə danışıqlar apara bilərsiniz. Bunun üçün http://openfire.opensource.az:7070/jitsi/apps/ofmeet linkinə daxil etməniz yetər. Bu kanal şifrələnmiş olmayacaq. Şifrələnmiş kanal üçün isə

http://openfire.opensource.az:7443/jitsi/apps/ofmeet linkinə daxil olmaq lazımdır.

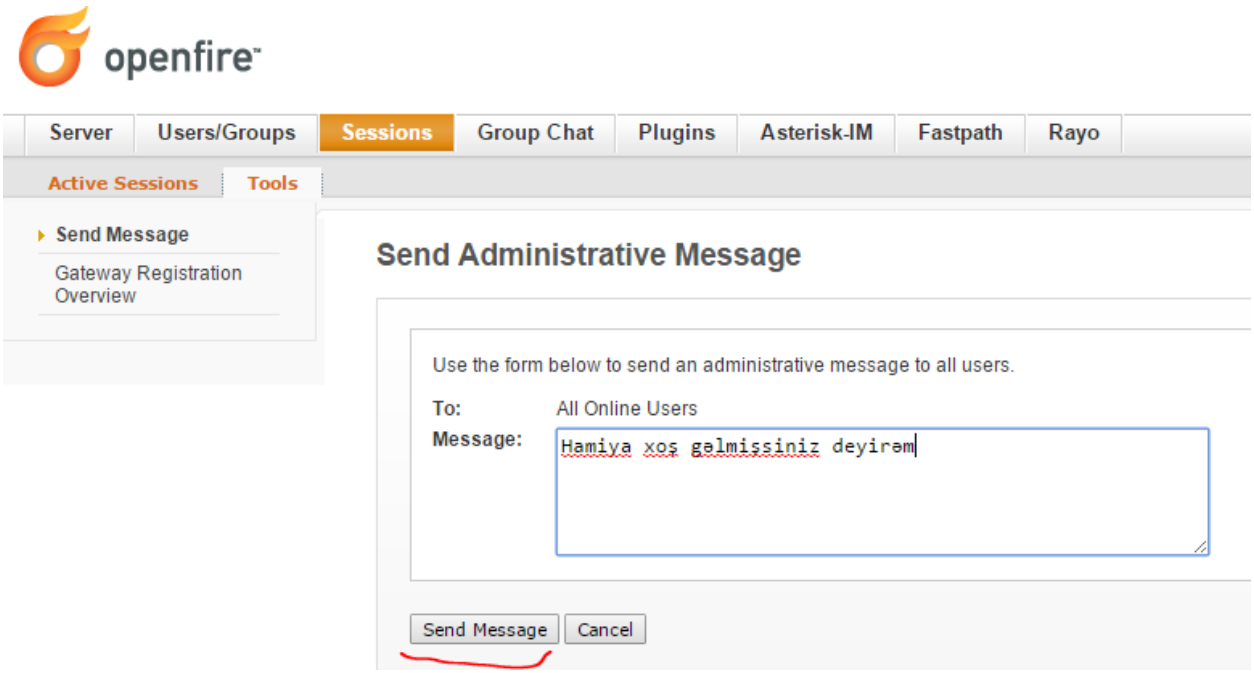
Qeyd: Yükləmədən susmaya görə əgər siz Jitsi client proqramdan istifadə edirsinizsə, o halda OpenFire tərəfdə hər bir müştəri üçün ayrıca SIP nömrə yaratmağa ehtiyac yoxdur. Çünki Jitsi client proqramı vasitəsilə XMPP üzərindən görüntü, səs, data ötürmək və həmdə əkranı paylaşmaq olur. Ancaq Jitsi client proqramı https://jitsi.org/Main/Download ünvanından endirilir, yüklənilir və XMPP protokolu istifadə edilərək quraşdırılır. Aşağıda jitsi proqramın qurulması göstəriləcək.

Ümumiyyətlə pluginlər **Server** tab-ın altında quraşdırılır. Həmçinin **Server -> Jitsi Videobridge** bölümünə daxil oluruq və aşağıdakı şəkildəki kimi, jitsi keçidə istifadəçi adı ilə şifrə təyin edirik və eynilə telefon quraşdırırıq:

Jitsi Videobridge Settings Page



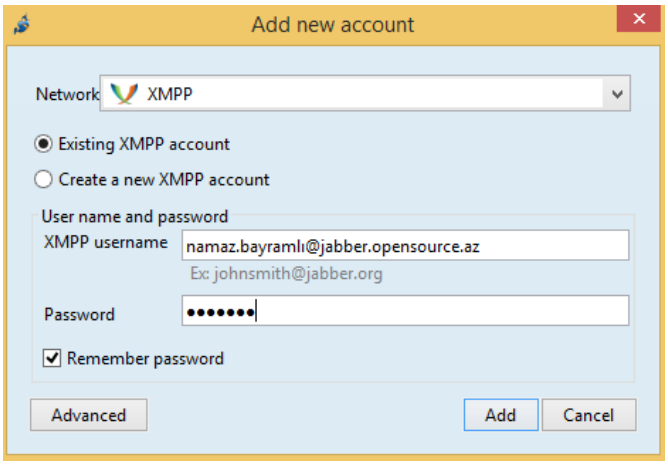
Siz həmçinin **Sessions -> Tools -> Send Message** bölümündən hər kəsə xəbərdarlıq yollaya bilərsiniz. Aşağıdakı şəkildə bu göstərilir:



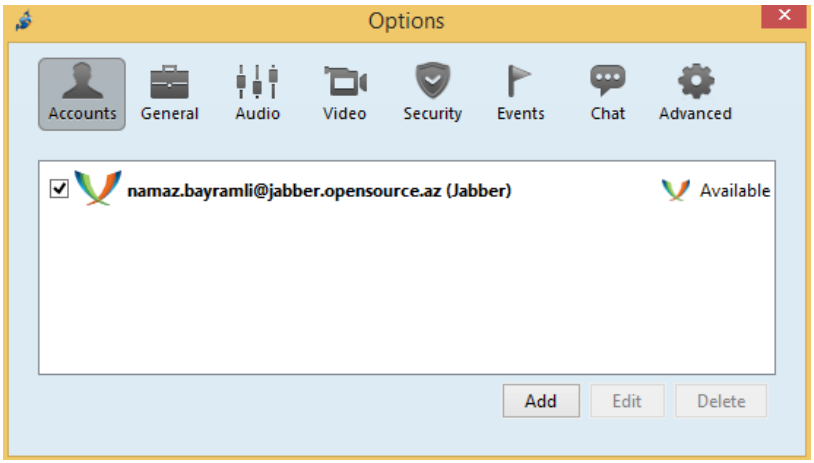
Əgər hər bir istifadəçi üçün SIP nömrə təyin etmək istəsək, öncədən serverə XMPP istifadəçilər əlavə edilir və ardınca **Server -> Phone -> Add new Phone Mapping** bölməsinə daxil olub SIP istifadəçiləri əlavə edirik(SIP server elə XMMP olan serverin özündədir). Misal üçün mövcud **namaz.bayramli** adlı XMPP istifadəçisi üçün SIP nömrə yaradırıq.

Sonra Windows maşınımıza Jitsi XMPP/SIP klient proqramını endiririk və aşağıdakı kimi quraşdırırıq(Rəsmi saytı: https://jitsi.org/Main/Download :

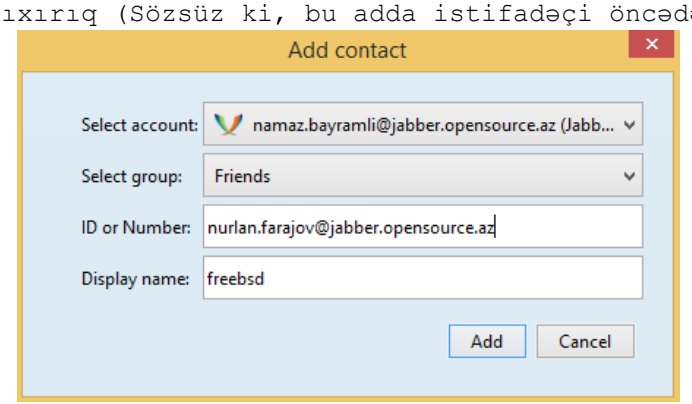
File -> Add new account -> XMPP -> XMPP Username – Password -> Add



Nəticədə aşağıdakı kimi istifadəçinin həm XMPP həsabı və həmdə SIP hesabı olacaq:

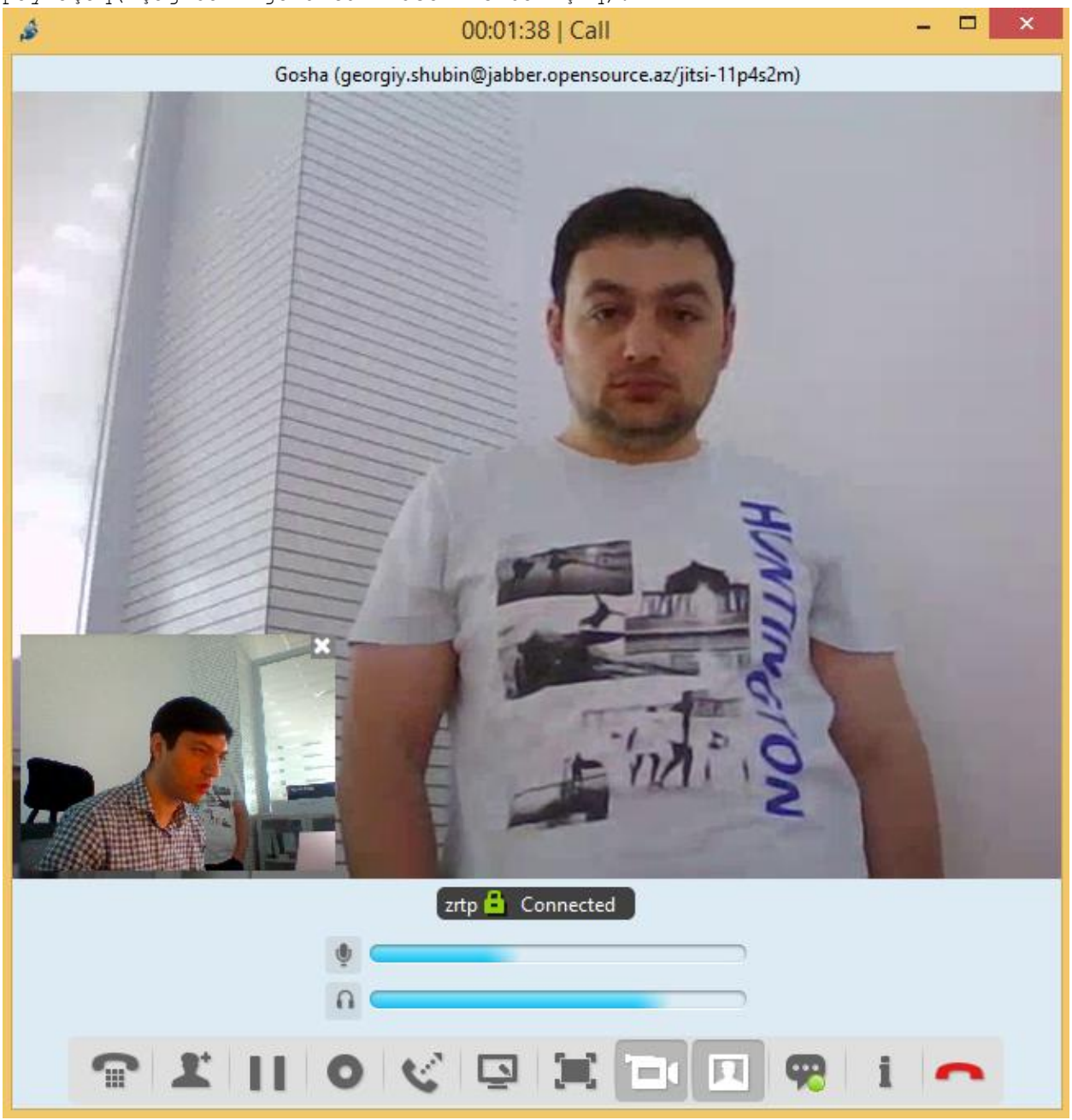


Sonra **File -> Add contact** və şəkildəki kimi verilənləri əlavə edib, **Add** düyməsinə sıxırıq (Sözsüz ki, bu adda istifadəçi öncədən mövcud idi):

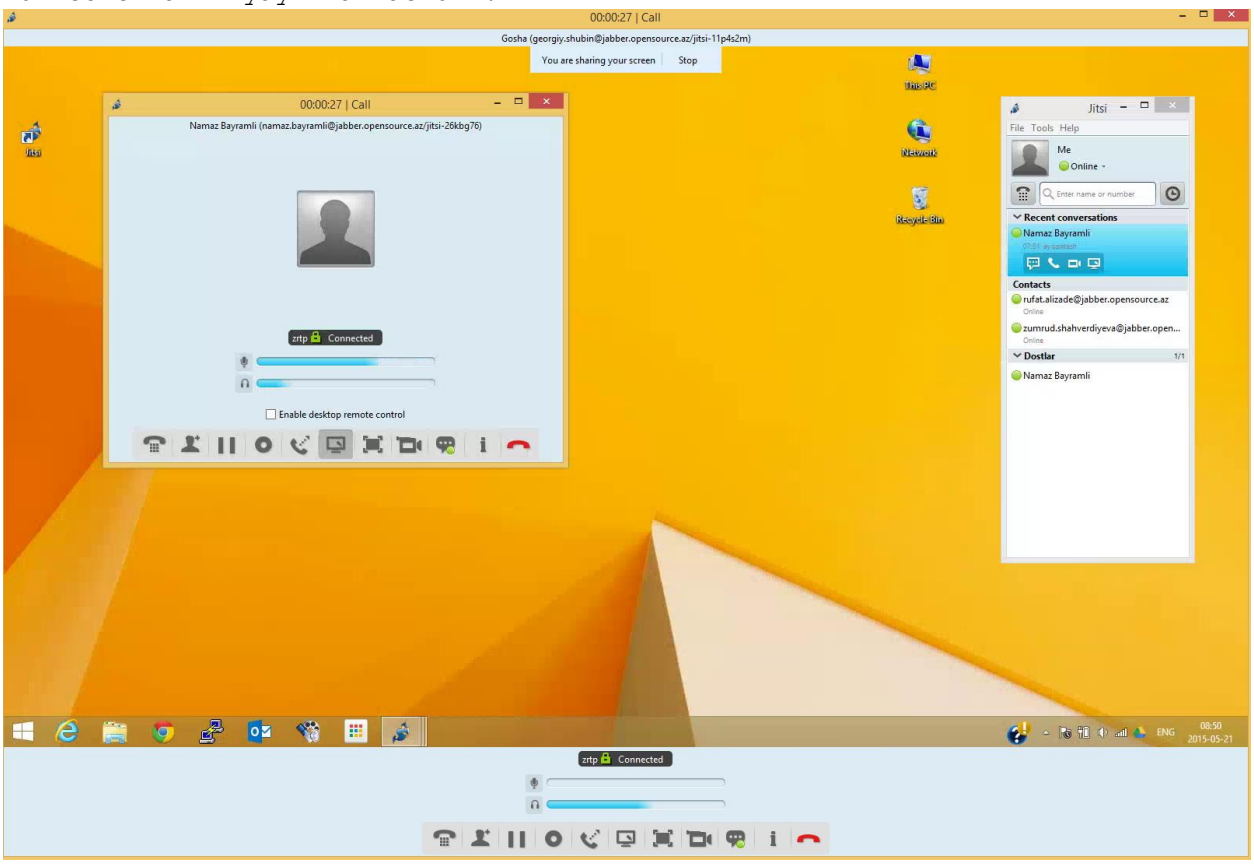


Yuxarıda göstərilən quraşdırmanı nurlan.farajov@jabber.opensource.az istifadəçisi üçündə edirik və həmin istifadəçi siyahısına eynilə namaz.bayramli@jabber.opensource.az istifadəçisini əlavə edirik.

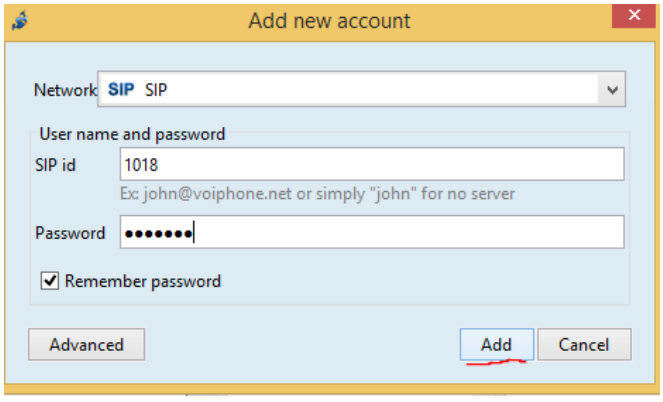
Nəticədə görüntü ilə bir maşından digərinə zəng edək və sonra ekranı paylaşaq(Aşağıdakı görüntü video ilə danışıq):



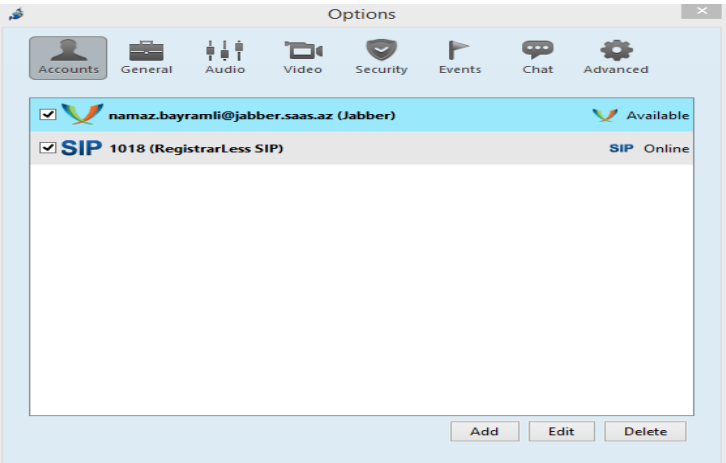
Bu isə ekranın yayımlanmasıdır:



Əgər SIP quraşdırma ilə birgə etsəniz aşağıdakı misal uyğun olacaq. Ancaq burada domain adı opensource.az istifadə edilir. Sonra yenidən yenədə **Tools -> Options -> Add -> SIP**(Network-da seçilir) və **SIP** istifadəçi adı ilə şifrə daxil edilir(Şəkildəki kimi):



Nəticədə aşağıdakı kimi istifadəçinin həm XMPP hesabı və həmdə SIP hesabı olacaq:



OpenFIRE ilə Active Directory inteqrasiyası

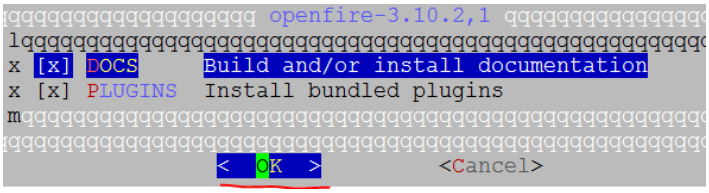
OpenFIRE – Əvvəllər Wildfire server və Jive Messenger kimi məşhur olan XMPP(Extendible Messaging and Presence Protocol – mövcud olma haqqında məlumat və genişlənə bilən məlumat mübadiləsi protokolu. Əvvəllər jabber protocol kimi tanınırdı. Java-da yazılmışdır, serverdir.

İdarəetmə üçün WEB interfeysə sahibdir. İnzibatçılar istənilən yerdən qoşula və rahat şəkildə istifadəçiləri silə, yarada və konfrans zallarına qoşa bilərlər.

Bu bölmədə biz FreeBSD 10.1 maşına OpenFIRE 3.10.2-nin PostgreSQL verilənlər bazası istifadə edərək yüklənməsinə baxacayıq. Həmçinin istifadəçi bazası müəssisəmizə aid olan Domain Controller-də olacaq. Yüklənmə və quraşdırmaya başlamazdan öncə nəzərdə tutulur ki, FreeBSD maşınənəzda artıq portlar və paketlər yüklənmiş və hazır vəziyyətdədir.

OpenFIRE-ı portlardan yükləyirik:

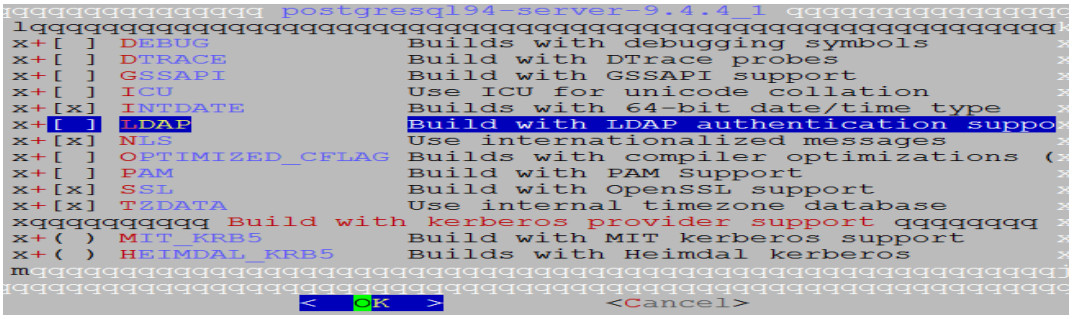
```
root@dolibarr:~ # cd /usr/ports/net-im/openfire
root@dolibarr:/usr/ports/net-im/openfire # make config
```




```
root@dolibarr:/usr/ports/net-im/openfire # make -DBATCH install
```

PostgreSQL verilənlər bazasını yükləyirik:

```
root@frfs:~ # cd /usr/ports/databases/postgresql94-server/
root@frfs:/usr/ports/databases/postgresql94-server # make config
```




```
root@frfs:/usr/ports/databases/postgresql94-server # make -DBATCH install
```

OpenFIRE və PostgreSQL-i StartUP-a əlavə edirik(Yəni **/etc/rc.conf** faylına):

```
root@frfs:~ # echo 'postgresql_enable="YES"' >> /etc/rc.conf
root@frfs:~ # echo 'openfire_enable="YES"' >> /etc/rc.conf
```

PostgreSQL inisializasiyasını işə salırıq:

```
root@frfs:~ # /usr/local/etc/rc.d/postgresql initdb
```

/usr/local/pgsql/data/postgresql.conf faylında aşağıdakı sətirin qarşısından şərhi silirik:

```
listen_addresses = 'localhost'
```

/usr/local/pgsql/data/pg_hba.conf faylında **host all all 127.0.0.1/32 trust** sətirini dəyişib aşağıdakı kimi edirik:
host all all 127.0.0.1/32 md5

PostgreSQL və OpenFIRE servislərini işə salırıq:
root@frfs:~ # **/usr/local/etc/rc.d/postgresql start**
root@frfs:~ # **/usr/local/etc/rc.d/openfire start**

Artıq **pgsql** istifadəçisi üçün şifrə təyin edirik:
root@frfs:~ # **passwd pgsql**
Changing local password for pgsql
New Password: **pgsql_şifrəsi**
Retype New Password: **pgsql_şifrəsi_təkrar**

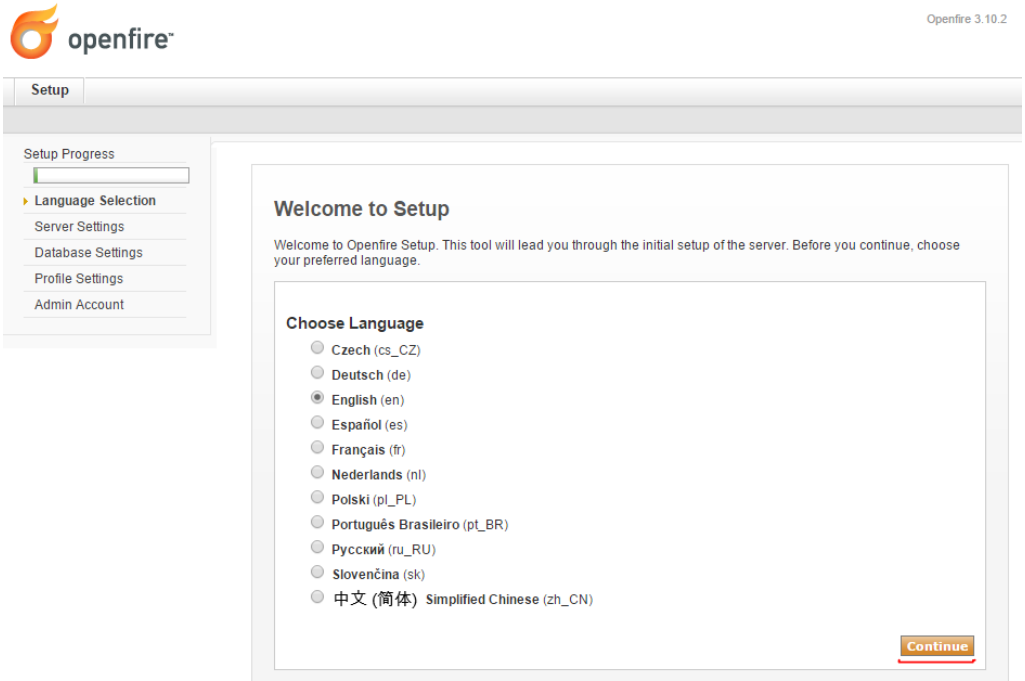
pgsql istifadəçi adı ilə daxil oluruq, **openfire** üçün istifadəçi və bu istifadəçinin qoşulması üçün verilənlər bazası yaradırıq:
root@frfs:~ # **su pgsql**
$ **createuser -sdrP openfire**
Enter password for new role: **şifrə**
Enter it again: **təkrar_şifrə**

$ **createdb openfire --owner=openfire**

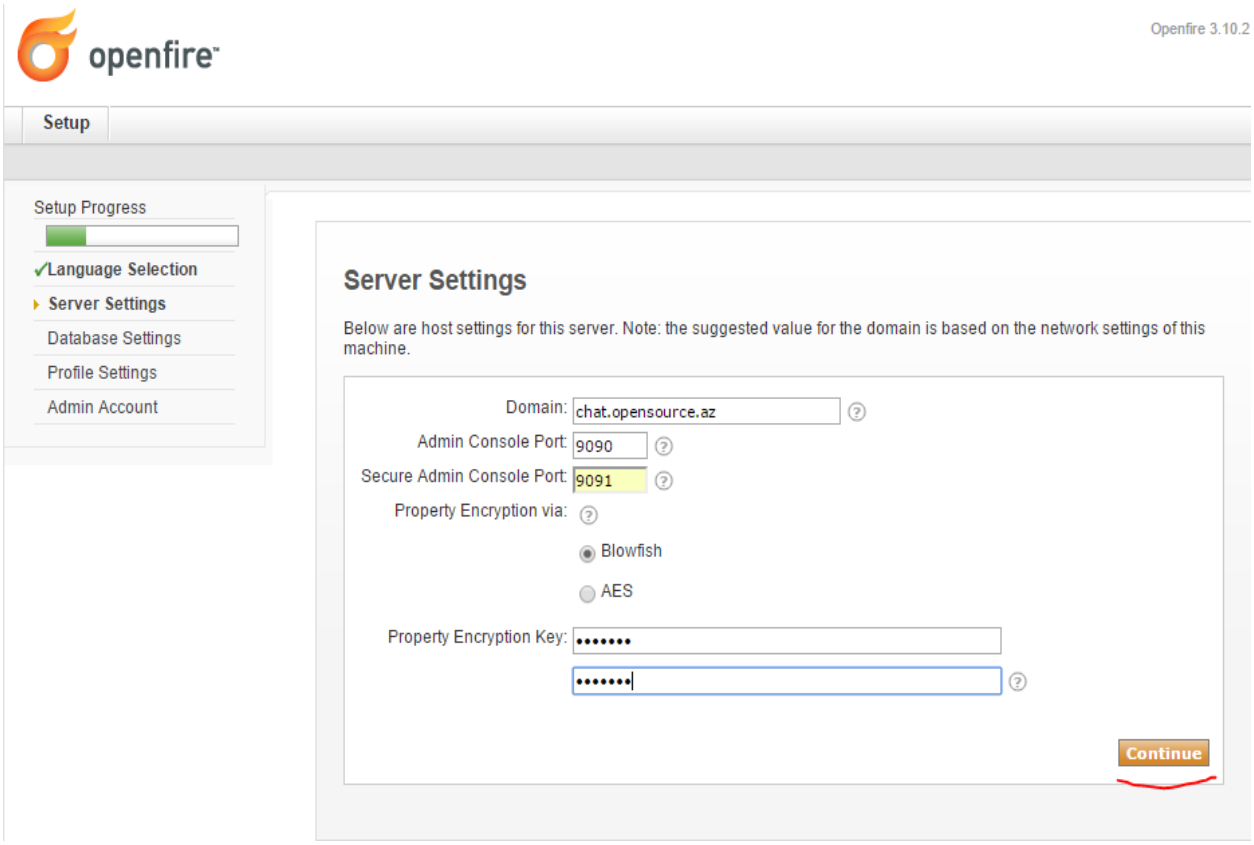
Konsoldan çıxırıq:
$ **exit**

PostgreSQL servisini yenidən işə salırıq:
root@frfs:~ # **service postgresql restart**

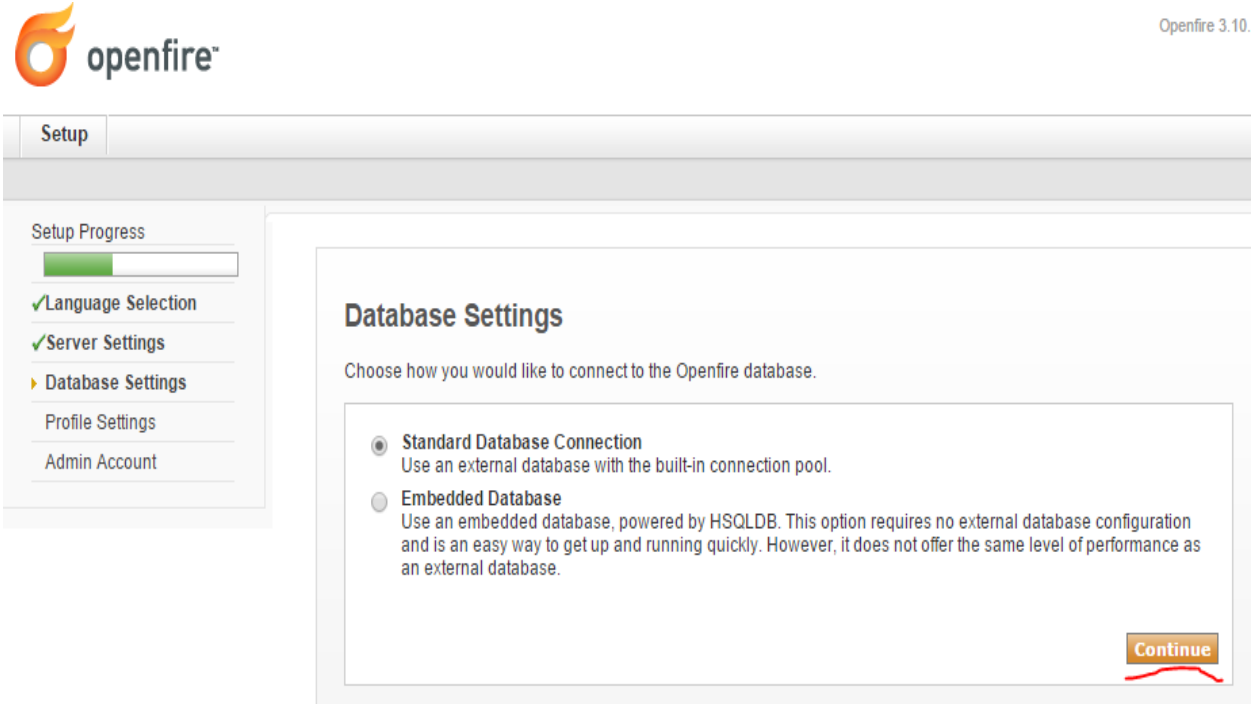
Hazırdır! Artıq istənilən Desktop maşındakı hansısa web browserdə http://server_IP:9090/ ünvanına daxil olsanız aşağıdakı səhifəni görəcəksiniz(**English** seçib **Continue** düyməsinə sıxırıq):



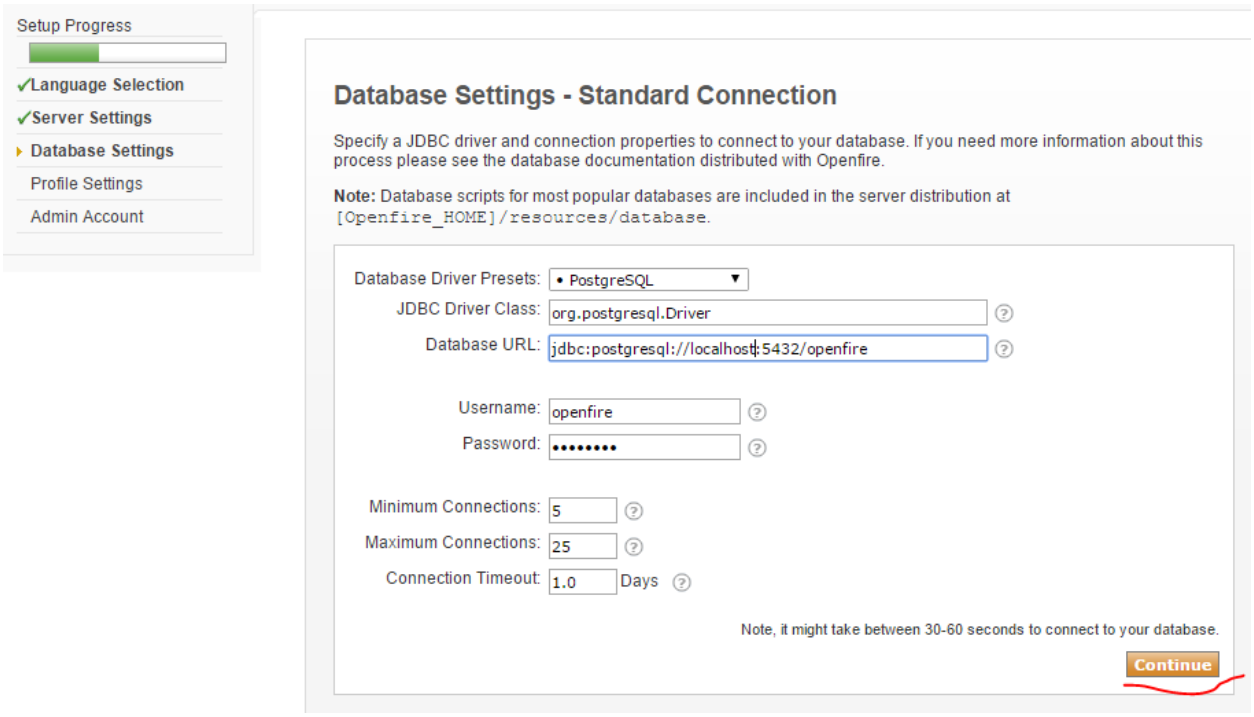
Açılan pəncərədə domain adı olaraq **chat.opensource.az** yazırıq və şifrələnəcək kanal üçün açara şifrə təyin edib, **Continue** düyməsinə sıxırıq:



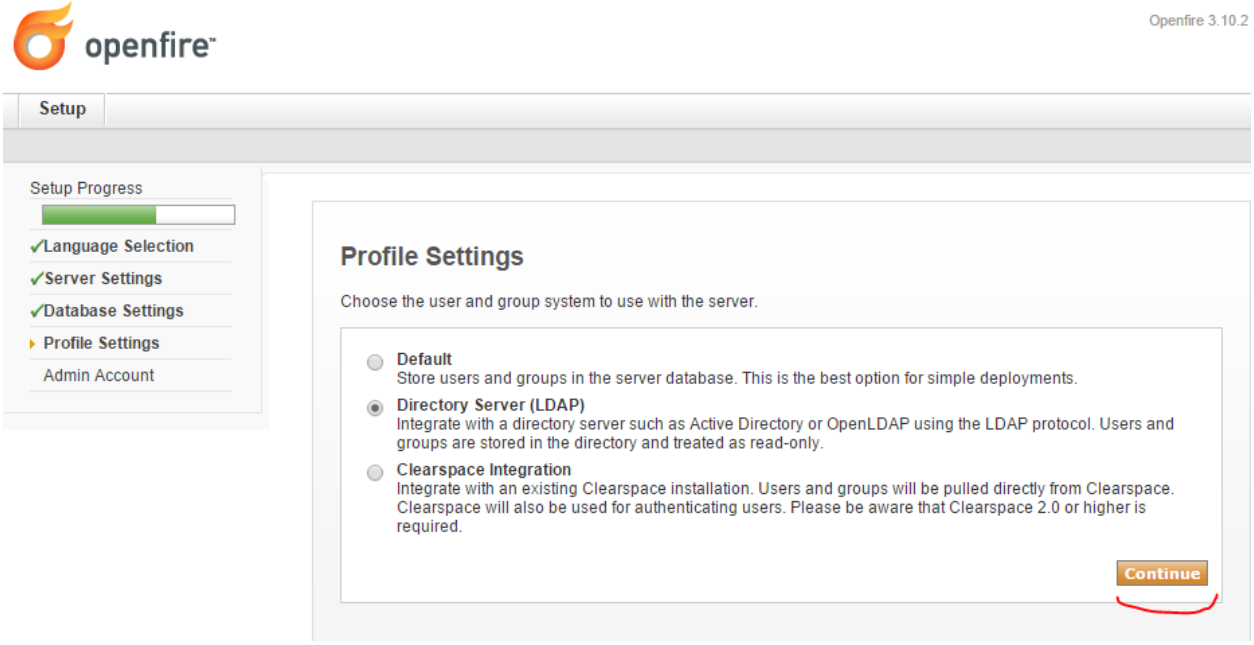
Standart Database Connection seçib **Continue** düyməsinə sıxırıq:



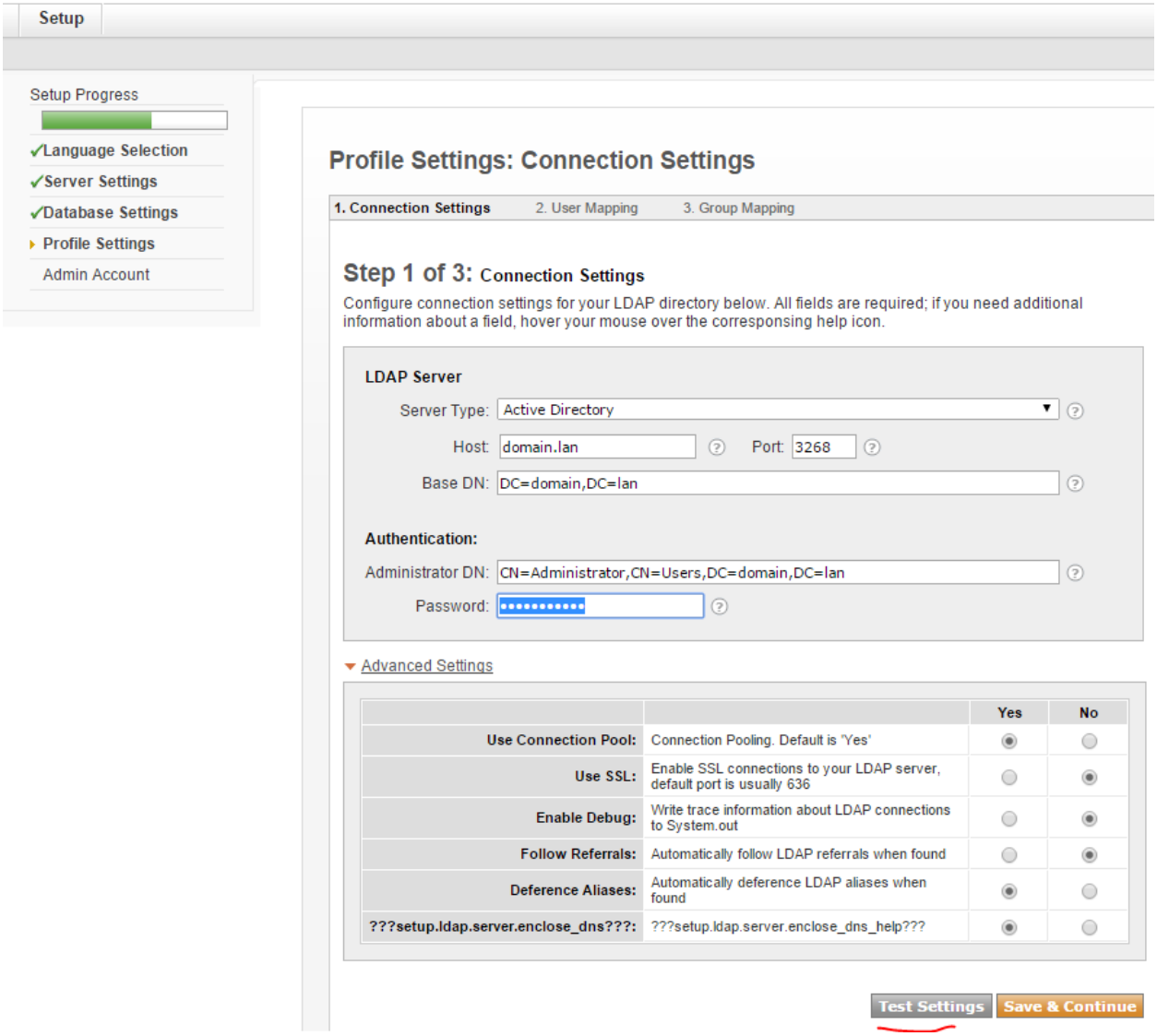
Verilənlər bazasına qoşulması üçün, database tipi PostgreSQL, qoşulacaq IP ünvan, verilənlər bazasının adı, istifadəçi adı və şifrəni yazıb, **Continue** düyməsinə sıxırıq:



Istifadəçi bazası olaraq LDAP(Yəni Active Dircetory) seçib, **Continue** düyməsinə sıxırıq:

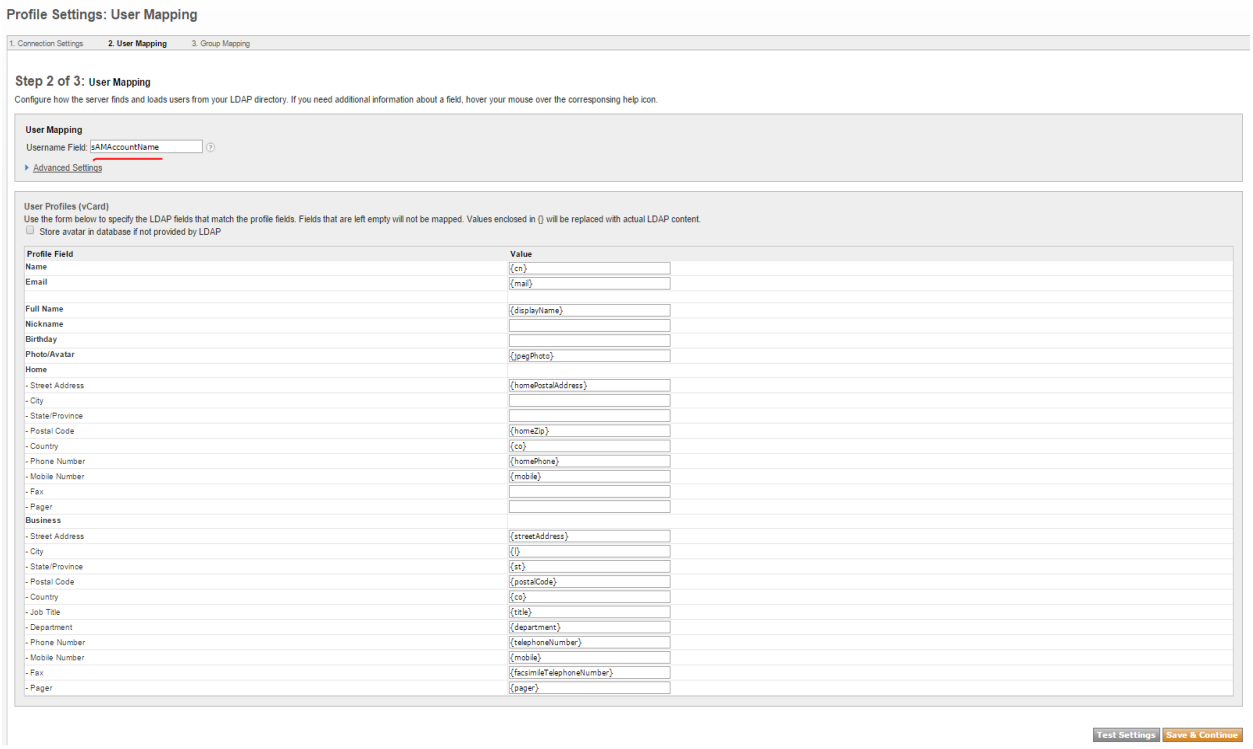


Active Directory-ə qoşulmaq üçün domain.lan-a aid olan **Distinguişəd Name** və Administrator istifadəçisi üçün Distinguished Name ilə şifrəsini yazırıq. Unutmayın ki, LDAP port **3268** yazırıq və **Test Settings** düyməsini sıxırıq:
DC adı: **domain.lan**
Filter edilən qrup adı: **CN=openfireUsers,OU=OpSO Groups,DC=domain,DC=lan**
Domain Administrator: **CN=Administrator,CN=Users,DC=domain,DC=lan**



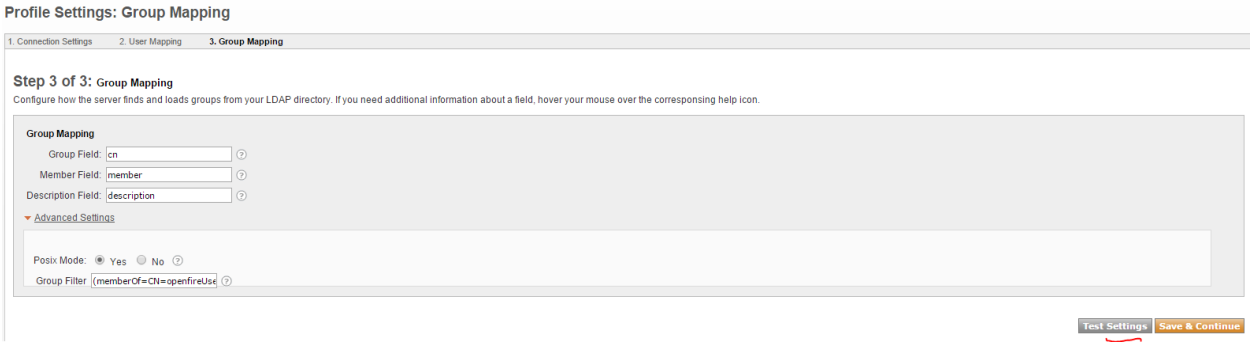
Uğurlu nəticə aşağıdakı şəkildəki kimi olacaq:

Status: Success! olduqdan sonra **Save & Continue** düyməsinə sıxırıq:

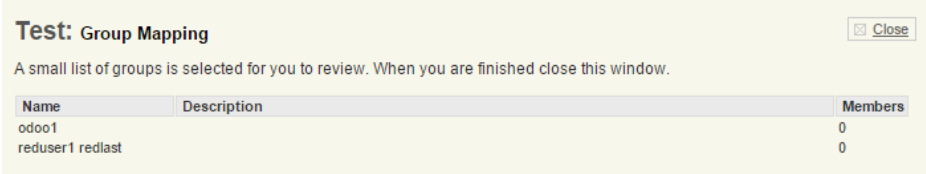


Qrupa görə filter edilməsi üçün Advanced Settings-in altında Group Filter bölümündə aşağıdakı sintaksisi yazırıq ki, yalnız DC-mizə aid olan **openfireUsers** qrupunun üzvləri serverimizə giriş edə bilsinlər(**Test Settings** düyməsini sıxıb, sınaqdan keçiririk):

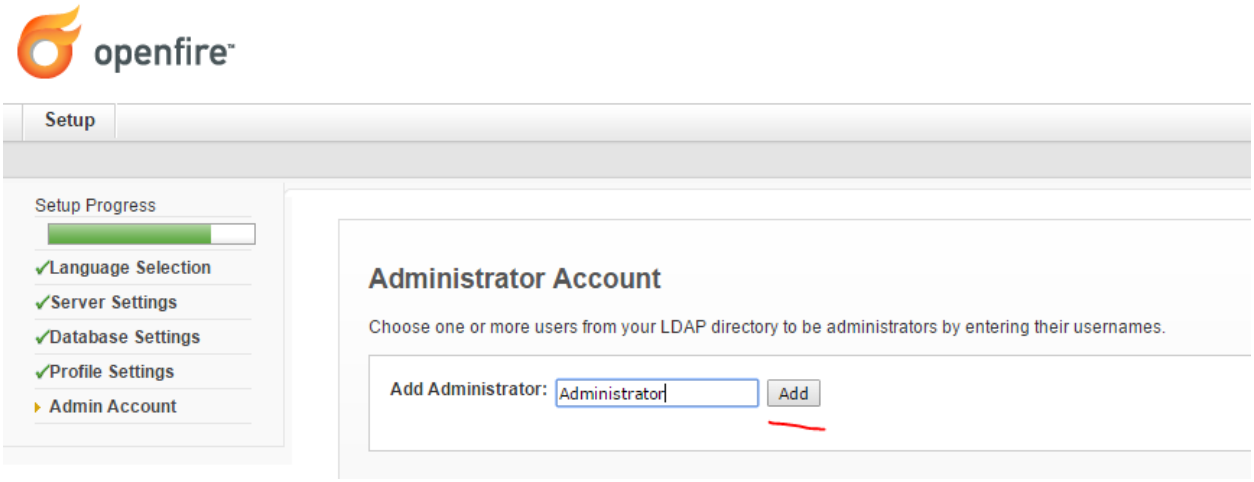
(memberOf=CN=openfireUsers,OU=OpSO Groups,DC=domain,DC=lan)



Əgər aşağıdakı kimi siyahı çap edilsə demək ki, qrupla birləşmə uğurla alınmışdır və içində olan istifadəçiləri aşağıdakı şəkildəki kimi görəcəksiniz:

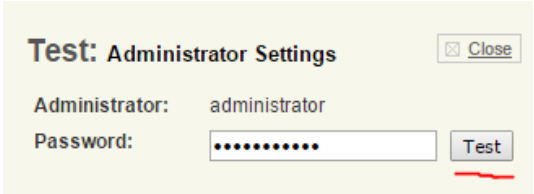


Save & Continue düyməsinə sıxaraq davam edirik. OpenFire üçün iznibatçı olacaq LDAP-da mövcud olan bir və ya bir neçə istifadəçi adını daxil edirik:

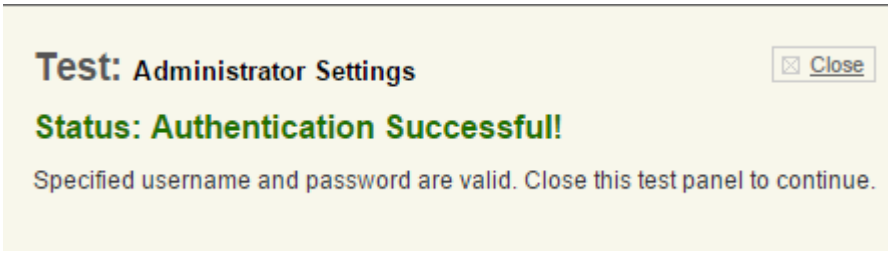


Açılacaq şəkildə **Administrator** LDAP istifadəçi həsabı üçün **test** düyməsini sıxıb sınaqdan keçiririk:

Istifadəçi şifrəsini daxil edib **test** düyməsinə sıxırıq:

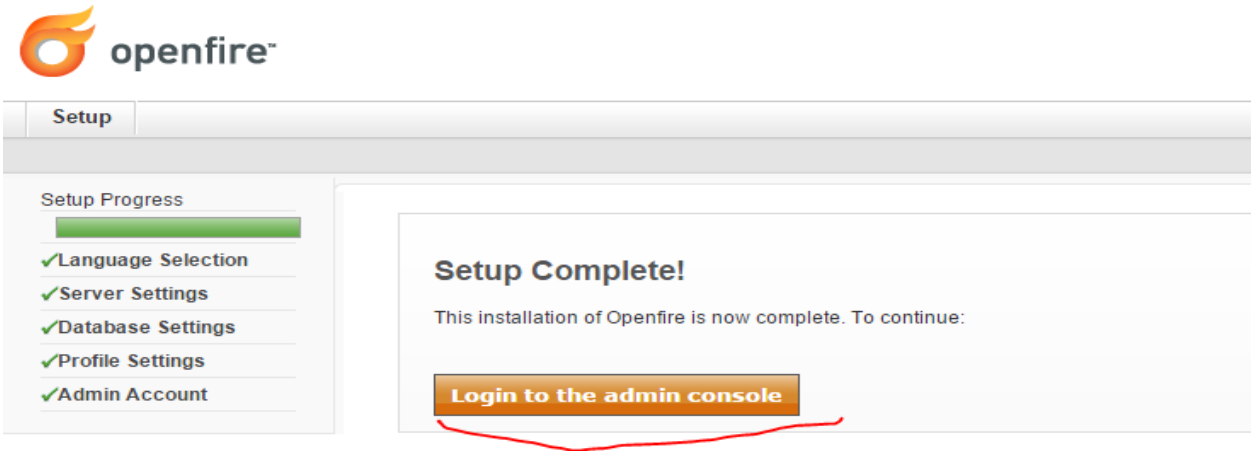


Uğurlu nəticə aşağıdakı kimidir:

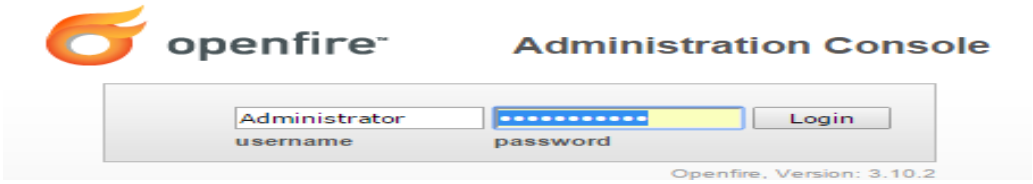


Ardınca **Continue** düyməsini sıxıb davam edirik.

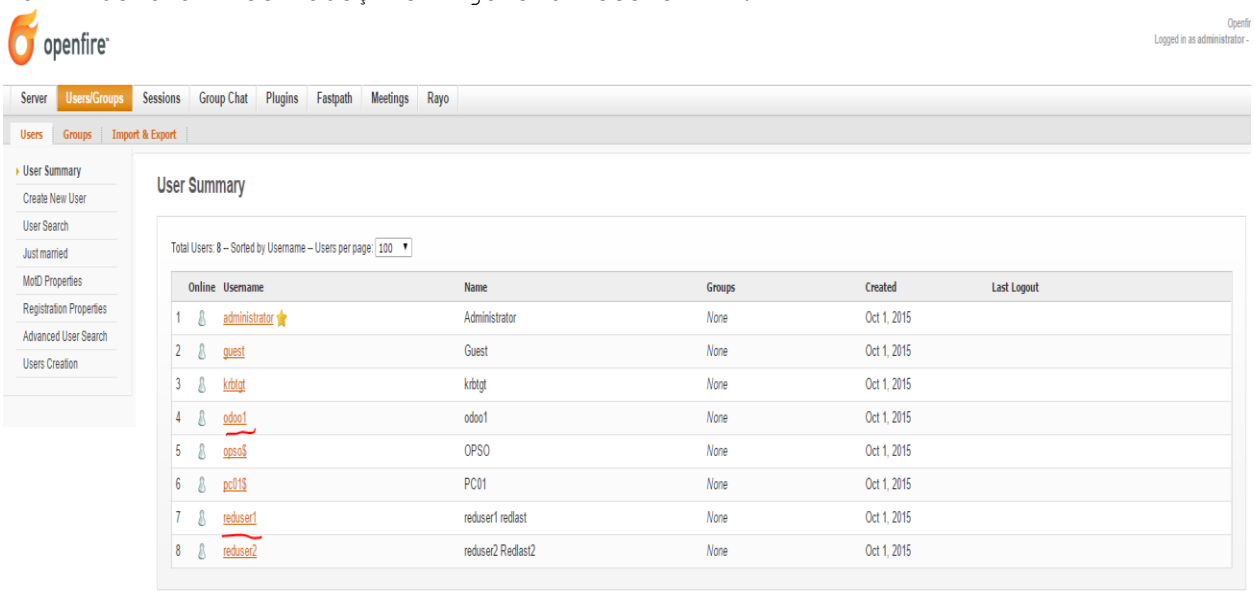
Artıq yüklənmə bitmişdir və bəhrəli nəticəni aşağıdakı kimi alıb, **Login to the admin console** düyməsini sıxmalısınız.



Qeyd etdiyimiz Domain admini **Administrator** istifadəçi həsabı adı və şifrəsini daxil edib **login** düyməsinə sıxırıq.



Sınaq üçün **Users/Groups** -> **Users** bölümünə daxil olsaz, **User Summary** altında DC-nizdə olan istifadəçiləri görə biləcəksiniz:



Sınaqların edilə bilməsi üçün **"OpenFire XMMP serverin qurulması"** bölümündə yazıldığı kimi, hər hansısa bir XMMP client vasitəsilə serverimizə qoşuluruq. Şəxsi təcrübəmə əsaslanaraq deyə bilərəm ki, ən funksionalı Jitsi-dir. Sadəcə DC-də olan iki istifadəçi ilə fərqli Desktop-lardan qoşulub sinaqlarınızı etməniz kifayətdir.

BÖLÜM 11

Bütün həllər üçün WEB serverlər

- **CentOS OCİ8 PHP5-FPM nGinx**
- **nGinx yüksək dayanıqlı reverse proxy**
- **Apache Tomcat8 yüklənməsi və quraşdırılması**
- **Apache ANT yüklənməsi və quraşdırılması**
- **Apache Maven yüklənməsi və quraşdırılması**
- **CentOS PDO_OCI inteqrasiyası**
- **Oracle JDK8-in yüklənməsi və quraşdırılması**
- **Ubuntu 14.04 x64 tomcat7 Java8 yüklənməsi və quraşdırılması**
- **Ubuntu Tomcat serverdə http və https portların dəyişdirilməsi**

Bu başlıqda demək olar ki gündəmdə istifadə olunan bütün web serverlərdən danışacıyıq. Adətən tələb, PHP işləyən serverin üstündə ORACLE verilənlər bazasına qoşulmasına yaranır çünki, əksər veb proqramlar php-də yazılır və şirkət bazası oracle-da olur. Həmçinin java proqramçıların öz yazdıqları kodları müəyyən bir veb application serverdə işlədə bilmələrinə ehtiyacları var. Java üçün tomcat server gündəmdə istifadə edilənlərdəndir. Eynilə proqramçıların kod anbarı üçün **ant** və **maven** haqqında danışılacaq. Tomcat serverdə standart portların istifadəsinin quraşdırılması açıqlanacaq.

CentOS OCİ8 PHP5-FPM nGinx

Məqsədimiz CentOS serverin üzərində nGinx WEB server, PHP-FPM və Oracle Client yükləyib quraşdırmaqdır. Lakin, PHP-nin oracle-a qoşulması üçün OCI(Oracle Call Interface) tələb edilir. Bu başlıqda PHP üzərində OCİ-ın quraşdırılması göstərilir.

Lazımı reposları endirək və quraşdıraq.

```
rpm --import https://fedoraproject.org/static/0608B895.txt
rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
rpm --import http://rpms.famillecollet.com/RPM-GPG-KEY-remi
rpm -ivh http://rpms.famillecollet.com/enterprise/remi-release-6.rpm

yum install yum-priorities

vi /etc/yum.repos.d/epel.repo    # "priority="-ni 10 edirik.
[epel]
name=Extra Packages for Enterprise Linux 6 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-6&arch=$basearch
failovermethod=priority
enabled=1
priority=10
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
[...]


vi /etc/yum.repos.d/remi.repo # Sonra "remi" sreposunda "enabled=1" edirik
[remi]
name=Les RPM de remi pour Enterprise Linux $releasever - $basearch
#baseurl=http://rpms.famillecollet.com/enterprise/$releasever/remi/$basearch/
mirrorlist=http://rpms.famillecollet.com/enterprise/$releasever/remi/mirror
enabled=1
priority=10
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi
failovermethod=priority

[remi-test]
name=Les RPM de remi en test pour Enterprise Linux $releasever - $basearch
#baseurl=http://rpms.famillecollet.com/enterprise/$releasever/test/$basearch/
mirrorlist=http://rpms.famillecollet.com/enterprise/$releasever/test/mirror
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi


yum install nginx        # nGinx Paketini yükləyirik
```

```
chkconfig --levels 235 nginx on      # nGinx-i startup-a əlave edirik.
/etc/init.d/nginx start              # Servisi Start edirik.

# PHP və modullarını yükləyirik.
yum -y install php-cli.x86_64 php.x86_64 php-common.x86_64 php-fpm.x86_64
php-devel.x86_64 php-odbc.x86_64 php-pear.noarch php-pecl-apc.x86_64 php-
pecl-apc-devel.x86_64

# '/etc/php.ini' faylın icində aşağıdakı sətirləri quraşdırırıq. Düzgün vaxtı
siz "cat /etc/sysconfig/clock" bu fayldan götürə bilərsiniz.
cgi.fix_pathinfo=0
date.timezone = "Europe/Berlin"

# PHP-FPM-i startup-a əlavə edib işə salırıq
chkconfig --levels 235 php-fpm on         # StartUP-a əlavə edirik.
/etc/init.d/php-fpm start                 # Start edirik.

vi /etc/nginx/nginx.conf    # Faylın icində aşağıdakı dəyişiklikləri edirik.
worker_processes  4;
keepalive_timeout  2;

vi /etc/nginx/conf.d/default.conf    # Faylı aşağıdakı kimi config edirik.
server {
    listen       80;
    server_name  _;
    autoindex on;
    #charset koi8-r;
    #access_log  logs/host.access.log  main;
    location / {
        root   /usr/share/nginx/html;
        index  index.php index.html index.htm;
    }
    error_page  404              /404.html;
    location = /404.html {
        root   /usr/share/nginx/html;
    }
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
    location ~ \.php$ {
        root           /usr/share/nginx/html;
        try_files $uri =404;
        fastcgi_pass   127.0.0.1:9000;     # Bu Port-da PHP-FPM qulaq asır
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME  $document_root$fastcgi_script_name;
        include        fastcgi_params;
    }
    location ~ /\.ht {
        deny  all;
    }
}
```

```
/etc/init.d/nginx reload            # Servisi reload edirik


vi /usr/share/nginx/html/info.php   # Test ücün php script yaradıb aşğıdakı
                                    məzmunu əlavə edirik.
<?php
  phpinfo();
?>

http://server_ip/test.php           # Test edirik.


## Gecikmələrin olmaması ücün biz PHP-FPM-i UNIX Socket faylında qulaq asdıra
bilərik.
vi /etc/php-fpm.d/www.conf          # Faylda aşağıdakı dəyişiklikləri edirik.
;listen = 127.0.0.1:9000
listen = /tmp/php5-fpm.sock
listen.owner = nginx
listen.group = nginx
user = nginx
group = nginx

/etc/init.d/php-fpm reload          # PHP-FPM-i reload edirik.


# Eyniylə nGinx-in icində-də dəyişikliyi etməliyik
vi /etc/nginx/conf.d/default.conf   # Faylda 9000-ci port əvəzinə Unix Socket
                                    yazırıq.
location ~ \.php$ {
        root           /usr/share/nginx/html;
        try_files $uri =404;
        fastcgi_pass   unix:/tmp/php5-fpm.sock;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME  $document_root$fastcgi_script_name;
        include        fastcgi_params;
    }

/etc/init.d/nginx reload            # Servisi reload edirik.

http://www.oracle.com/technetwork/topics/linuxx86-64soft-092277.html ->
Ünvandan OracleInstanceClient-i dartırıq.

rpm -ivh oracle-instantclient11.2-basic-11.2.0.3.0-1.x86_64.rpm   # Paketi
                                                                  yükləyirik
rpm -ivh oracle-instantclient11.2-devel-11.2.0.3.0-1.x86_64.rpm   # Paketi
                                                                  yükləyirik

'/usr/lib/oracle/11.2/client64/' - ORACLE_HOME bu ünvana yüklənir.

pecl install oci8                   # oci8 modulunu yükləyirik. 'autodetect'
                                    seçirik ki, özü oci8 ünvanını tapsın.
                                    Əgər, tapmasa ünvan
```

'/usr/lib/oracle/11.2/client64/bin'
ünvanı yazın.

```
vi /etc/php.ini                # Faylda oci8 genişlənməsini aktivləşdiririk.
extension=oci8.so

vi /root/.bash_profile         # Fayla aşağıdakı sətirləri əlavə edirik.
export LD_LIBRARY_PATH=/usr/lib/oracle/11.2/client64/lib

/etc/init.d/nginx reload       # Servisi reload edirik.
```

Test üçün **/usr/share/nginx/html** ünvanında **index.php** faylı yaradıb içinə aşağıdakı məzmunu əlavə edirik.

```
<?php
   // put real credentials
   $conn = oci_connect('test', 'test', 'localhost/SMPP');
   if (!$conn) {
       $e = oci_error();
       trigger_error(htmlentities($e['message'], ENT_QUOTES), E_USER_ERROR);
   }else{
       echo 'Success';
       oci_close($conn);
   }
?>
```

Qeyd: Ancaq **/etc/hosts** faylına maşınınızın adını IP unvan ilə əlavə etməyi və **127.0.0.1** üçün localhost adının əlavə edilməsini unutmayın. Əks halda işləməyəcək. Aşağıdakı qaydada:

```
cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
10.70.3.221 smapp.lan smapp
```

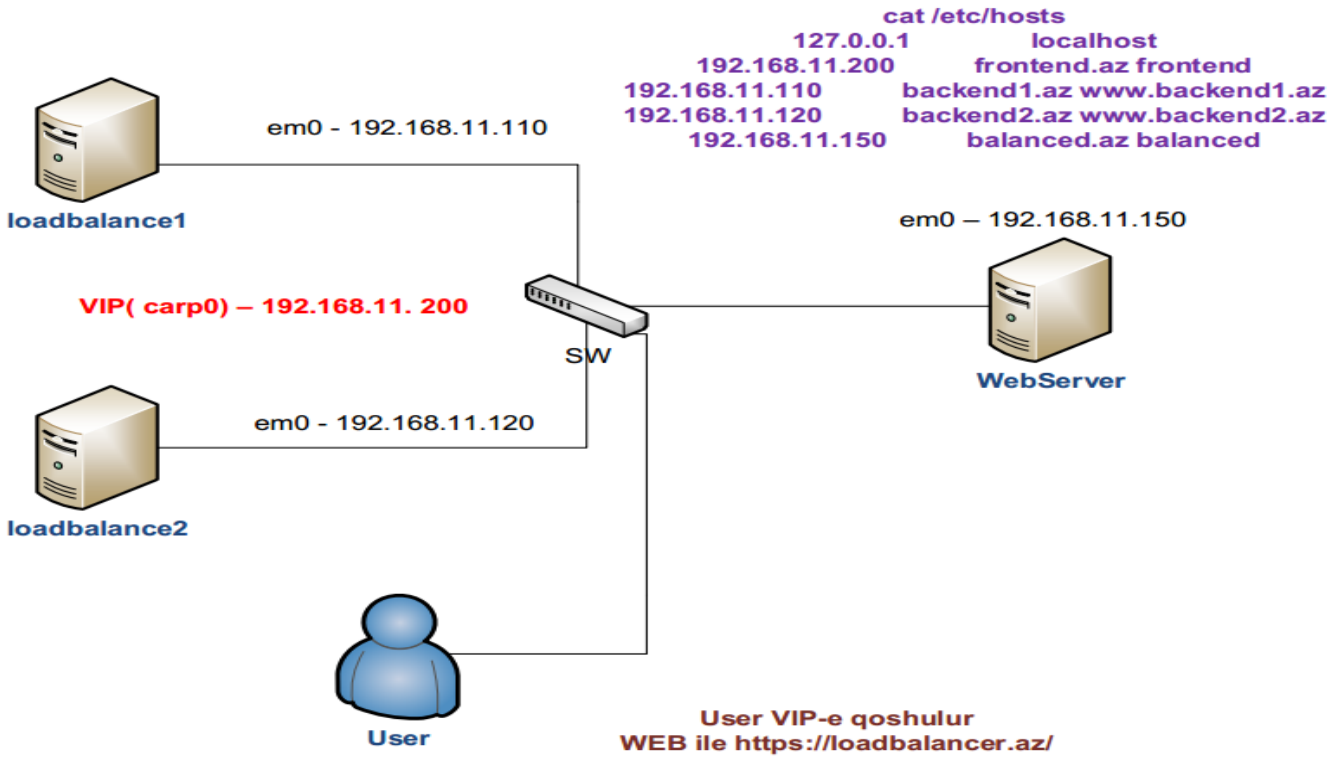
Sonda nginx və phpfpm servislərini yenidən işə salırıq.

```
/etc/init.d/nginx restart
/etc/init.d/php-fpm restart
```

nGinx yüksək dayanıqlı reverse proxy

Məqsədimiz müəyyən bir WEB xidmətinin dayanıqlı işləməsidir. Yəni həm yükün paylaşılması və həm də yüksək dayanıqlıq tələbi yaranarsa, siz bu sənədə müraciət etməlisiniz. Şəkildə göründüyü kimi, işlək vəziyyətdə olan bir Apache web serverimiz var. Tələb bu web serverin dayanıqlığını təmin etməkdən ibarətdir. Serverin öz sayı bizi maraqlandırmalı deyil cünki, bizə həmin serverin Virtual İP ünvanı da verilə bilər və siz də elə təsəvvür etsəniz yaxşı olar. Hal-hazırda bu dayanıqlığı nGinx vasitəsilə edəcəyik.

Şəbəkə quruluşu aşağıdakı şəkildəki kimidir:



Loadbalance1 maşının qurulması

em0 - 192.168.11.110

Redundancy üçün Virtual carp aləti yaradaq və ona IP mənimsədək.
Aşağıdakı sətirləri **'/etc/rc.conf'** faylına əlavə edirik.

```
cloned_interfaces="carp0"
ifconfig_carp0="vhid 1 advskew 0 pass VeRySeCrEtPaSsWoRd 192.168.11.200/24"
```

hosts faylı loadbalance1 maşınında aşağıdakı kimi olacaqdır.

cat /etc/hosts

```
127.0.0.1               localhost
192.168.11.200          frontend.az frontend
192.168.11.110          backend1.az www.backend1.az
192.168.11.120          backend2.az www.backend2.az
192.168.11.150          balanced.az balanced
```

balanced.az - Daxildə olan WEB serverin adı
loadbalancer.az – Istifadəçi öz WEB browserində bu adla Loadbalancer-ə müraciət edəcək.

```
cd /usr/ports/www/nginx              # nGinx-i yükləyək
make config                          # Lazımı modulları seçək
```

```
[x] HTTP                  Enable HTTP module
[x] HTTP_CACHE            Enable http_cache module
[x] HTTP_REALIP           Enable http_realip module
[x] HTTP_REWRITE          Enable http_rewrite module
[x] HTTP_SSL              Enable http_ssl module
[x] HTTP_STATUS           Enable http_stub_status module
[x] WWW                   Enable html sample files
[x] SYSLOG_SUPPORT        3rd party syslog support
[x] TCP_PROXY             3rd party tcp_proxy module
```

```
make -DBATCH install                 # Yükləyək
```

SSL Sertifikatları yaradaq.

```
cd /usr/local/etc/nginx              # nGinx sertifikatları yaradaq
mkdir ssl                            # SSL üçün qovluq yaradaq
cd ssl/                              # Qovluğa daxil olaq
```

```
openssl genrsa -des3 -out loadbalance.in.key 1024  # Gizli açarı yaradaq.
```

```
root@backend1:/usr/local/etc/nginx/ssl # openssl genrsa -des3 -out loadbalance.in.key 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
.............................++++++
e is 65537 (0x10001)
Enter pass phrase for loadbalance.in.key:
Verifying - Enter pass phrase for loadbalance.in.key:
```

Certificate Signing Request yaradırıq

```
openssl req -new -key loadbalance.in.key -out loadbalance.in.csr
```

```
root@backend1:/usr/local/etc/nginx/ssl # openssl req -new -key loadbalance.in.key -out loadbalance.in.csr
Enter pass phrase for loadbalance.in.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:Baku
Locality Name (eg, city) []:Garadag
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ATLtech
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:loadbalance.az
Email Address []:qabriel@mail.ru

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Açarı backup edək və şifrəni silək

```
cp loadbalance.in.key loadbalance.in.key.bak
openssl rsa -in loadbalance.in.key.bak -out loadbalance.in.key
```

Açarı imzalayaq.

openssl x509 -req -days 365 -in loadbalance.in.csr -signkey loadbalance.in.key -out loadbalance.in.crt

```
root@backend1:/usr/local/etc/nginx/ssl # openssl x509 -req -days 365 -in loadbal
ance.in.csr -signkey loadbalance.in.key -out loadbalance.in.crt
Signature ok
subject=/C=AZ/ST=Baku/L=Garadag/O=ATLtech/OU=IT/CN=loadbalance.az/emailAddress=q
abriel@mail.ru
Getting Private key
```

nGinx quraşdırma faylını aşağıdakı kimi edək.(Bu fayl hər iki maşında loadbalance1 və loadbalance2-də eyni olmalıdır)

192.168.11.200 - Virtual IP ünvandır hansı ki, istifadəçilər DNS ilə ad aldıqdan sonra bu IP ünvana yönləndiriləcəklər.

cat /usr/local/etc/nginx/nginx.conf # Quraşdırma faylı aşağıdakı kimi olacaq.

```
worker_processes  1;
events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;

    sendfile        on;
    keepalive_timeout  65;

server {
        listen          192.168.11.200:443;
        ssl             on;
        server_name     loadbalancer.az;

        access_log      /var/log/nginx/ssl-access.log;
        error_log       /var/log/nginx/ssl-error.log;

        ssl_certificate      /usr/local/etc/nginx/ssl/loadbalance.in.crt;
        ssl_certificate_key  /usr/local/etc/nginx/ssl/loadbalance.in.key;

        ssl_protocols        SSLv3 TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers RC4:HIGH:!aNULL:!MD5;
        ssl_prefer_server_ciphers on;
        keepalive_timeout    60;
        ssl_session_cache    shared:SSL:10m;
        ssl_session_timeout  10m;

        location / {
                proxy_pass  http://balanced.az;
```

```
                proxy_next_upstream error timeout invalid_header http_500
http_502 http_503 http_504;

                proxy_set_header        Accept-Encoding   "";
                proxy_set_header        Host            $host;
                proxy_set_header        X-Real-IP       $remote_addr;
                proxy_set_header        X-Forwarded-For
$proxy_add_x_forwarded_for;

                proxy_set_header        X-Forwarded-Proto $scheme;
                add_header              Front-End-Https   on;

                proxy_redirect     off;
        }
 }
}
```

```
mkdir /var/log/nginx/                # Jurnal üçün qovluq yaradaq
touch /var/log/nginx/ssl-access.log # access jurnal faylını yaradaq
touch /var/log/nginx/ssl-error.log  # error üçün jurnal faylını yaradaq

/usr/local/etc/rc.d/nginx start      # nGinx-i işə salırıq.

nginx -t                             # Quraşdırmalarımızın düzgünlüyünü test
                                     edək(nəticə aşağıdakı kimi olmalıdır)
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful

nginx -s reload          # nGinx-i reload edirik.
```

nginx.conf faylını ikinci serverə nüsxələyək.

```
scp /usr/local/etc/nginx/nginx.conf root@192.168.11.120:/usr/local/etc/nginx/
```

loadbalance1 mashından **loadbalance2** maşınında SSL sertifikatlar üçün qovluq yaradaq və onları ora nüsxələyək.

```
ssh root@192.168.11.120 'mkdir /usr/local/etc/nginx/ssl' # Qovluğu yaradırıq
scp /usr/local/etc/nginx/ssl/* root@192.168.11.120:/usr/local/etc/nginx/ssl/
```

```
root@backend1:/root # scp /usr/local/etc/nginx/ssl/* root@192.168.11.120:/usr/lo
cal/etc/nginx/ssl/
Password:
loadbalance.in.crt                            100%  936     0.9KB/s   00:00
loadbalance.in.csr                            100%  692     0.7KB/s   00:00
loadbalance.in.key                            100%  887     0.9KB/s   00:00
loadbalance.in.key.bak                        100%  963     0.9KB/s   00:00
```

loadbalancer2 maşınını quraşdırırıq.

em0 - 192.168.11.120/24

hosts faylı loadbalance1 maşınında aşağıdakı kimi olacaqdır.

```
cat /etc/hosts
127.0.0.1               localhost
192.168.11.200          frontend.az frontend
192.168.11.110          backend1.az www.backend1.az
192.168.11.120          backend2.az www.backend2.az
192.168.11.150          balanced.az balanced
```

Aşağıdakı sətirləri **'/etc/rc.conf'** faylına əlavə edirik.

```
cloned_interfaces="carp0"
ifconfig_carp0="vhid 1 advskew 1 pass VeRySeCrEtPaSsWoRd 192.168.11.200/24"
```

```
cd /usr/ports/www/nginx              # nGinx-i yükləyək
make config                          # Lazımı modulları seçək
```

```
[x] HTTP                 Enable HTTP module
[x] HTTP_CACHE           Enable http_cache module
[x] HTTP_REALIP          Enable http_realip module
[x] HTTP_REWRITE         Enable http_rewrite module
[x] HTTP_SSL             Enable http_ssl module
[x] HTTP_STATUS          Enable http_stub_status module
[x] WWW                  Enable html sample files
[x] SYSLOG_SUPPORT       3rd party syslog support
[x] TCP_PROXY            3rd party tcp_proxy module
```

```
make -DBATCH install                      # Yükləyək
```

nGinx üçün jurnal qovluğu və faylları yaradaq

```
mkdir /var/log/nginx/                # Jurnal üçün qovluq yaradaq
touch /var/log/nginx/ssl-access.log # access jurnal faylını yaradaq
touch /var/log/nginx/ssl-error.log  # error üçün jurnal faylını yaradaq

/usr/local/etc/rc.d/nginx start      # nGinx-i işə salırıq.

nginx -t                             # Configimizin düzgünlüyünü test
                                     edək(nəticə aşağıdakı kimi olmalıdır)
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful

nginx -s reload          # nGinx-i reload edirik.
```

Client maşından WEB Serveri sertifikat ilə test eləmək üçün aşağıdakı əmrləri yazmağınız yetər.

```
openssl s_client -connect loadbalancer.az:443
```

balanced.az maşında isə adi apache22 WEB server qaldırılmışdır və 192.168.11.150 IP ünvanında işləyir.

```
em0 - 192.168.11.150                # backend WEB Server IP

pkg_add -r apache22                 # apache22-ni yükləyirik

echo 'apache22_enable="YES"' >> /etc/rc.conf     # apache22-ni startup-a əlavə
                                                 edirik

/usr/local/etc/rc.d/apache22 start               # Daemon-u işə salırıq
```

index.html faylını aşağıdakı kimi düzəldirik.

```
echo "<html><center><h1>This is redundent site!</h1></center></html>" >
/usr/local/www/apache22/data/index.html
```

Sonda Client-in birindən https://loadbalancer.az/ ünvanında daxil olub F5-i sıxaraq sınaqdan keçirin. Eyni zamanda **192.168.11.110** IP ünvanlı serveri söndürün. Hər şey miqrasiya ediləcək **192.168.11.120** IP ünvanlı serverin üstünə.

Apache Tomcat8 yüklənməsi və quraşdırılması

Apache tomcat – açıq kodlu web serverdir hansı ki, Java Servlet və JavaServer səhifələri texnologiyaları üçündür. Java WEB-də yazılmış kodlar bu web server vasitəsilə işə dsalınır.Demək olarki, Tomcat web server dünyada ən vacib sayılan və Javada yazılmış web kodlarını öz üzərində daşıyır.

Rəsmi saytından ən son sıxılmış versiyasını endiririk:
wget http://mirrors.advancedhosters.com/apache/tomcat/tomcat-8/v8.0.23/bin/apache-tomcat-8.0.23.zip

Endirdiyimiz zip faylı açırıq:
unzip apache-tomcat-8.0.23.zip

Açdığımız qovluğu **/opt/tomcat** ünvanına köçürürük:
mv apache-tomcat-8.0.23 /opt/tomcat

Tomcat mühit dəyişənlərini elan etmək üçün **/etc/profile.d/tomcat.sh** faylı yaradırıq və məzmununa aşağıdakı dəyişənləri əlavə edirik:

```
#!/bin/bash
CATALINA_HOME=/opt/tomcat
PATH=$CATALINA_HOME/bin:$PATH
export PATH CATALINA_HOME
export CLASSPATH=.
```

Yaratdığımız faylı yerinə yetirən edirik:
chmod +x /etc/profile.d/tomcat.sh

Mövcud seansımızda dəyişənləri aşağıdakı əmrlə işə salırıq:
source /etc/profile.d/tomcat.sh

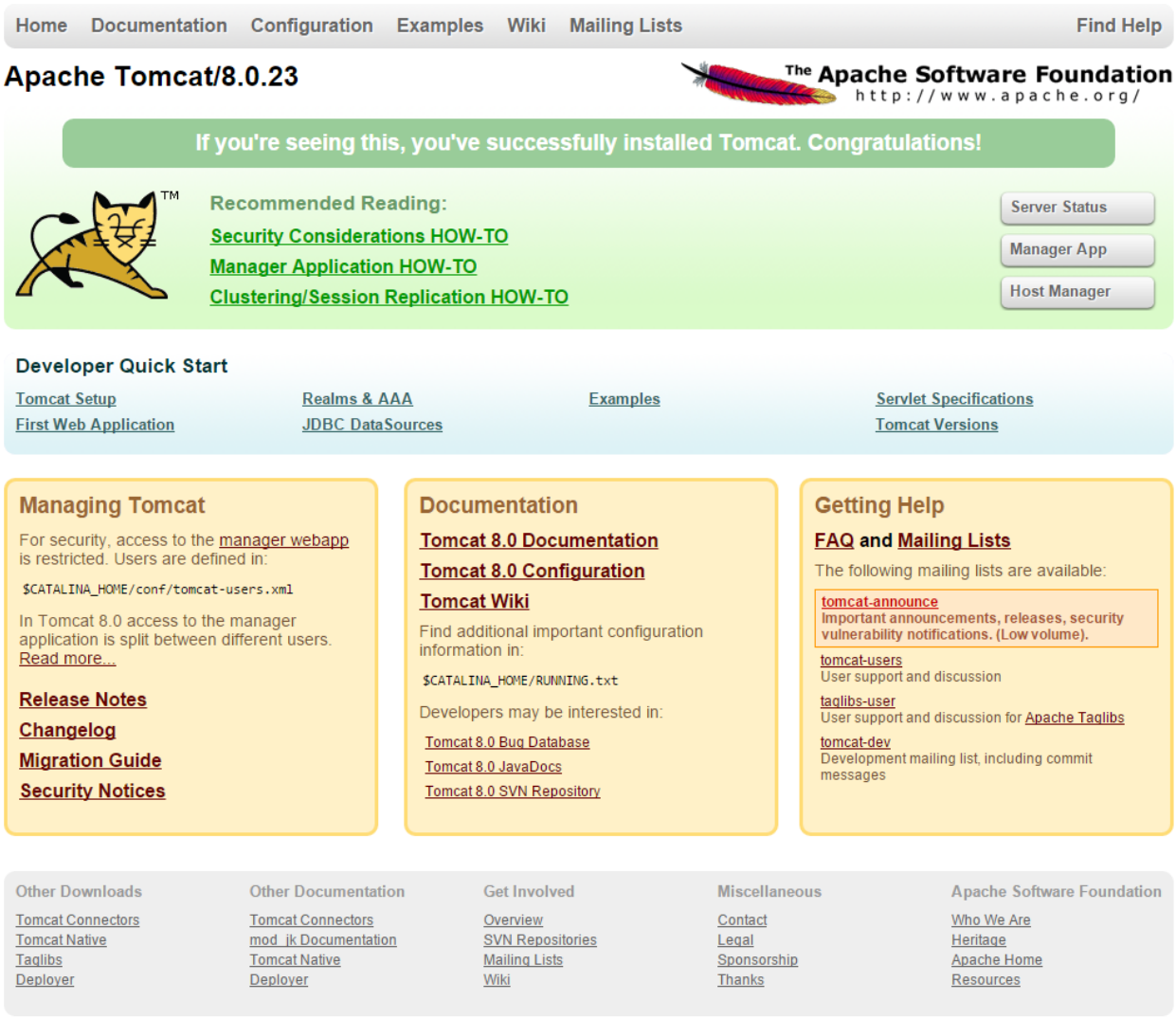
Artıq biz tomcat-ı işə sala bilərik. Ancaq işə salmazdan öncə aşağıdakı scriptləri yerinə yetirən edirik:

```
# chmod +x $CATALINA_HOME/bin/startup.sh
# chmod +x $CATALINA_HOME/bin/shutdown.sh
# chmod +x $CATALINA_HOME/bin/catalina.sh
```

Aşağıdakı əmrlə tomcat işə salırıq:

```
# cd $CATALINA_HOME/bin
# ./startup.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

Işə düşdükdən sonra, tomcat8 serverimiz 8080-ci porta qulaq asacaq.Serverinizəhttp://IP_Address:8080qoşulub yoxlayın və aşağıdakı şəkildə olan nəticəni əldəetməlisiniz:



Serveri dayandırmaq üçün isə aşağıdakı əmrdən istifadə etmek lazımdır:

```
# cd $CATALINA_HOME/bin
# ./shutdown.sh
```

Tomcat-ın system yenidən yüklənmsindən sonra avtomatik işə düşməsi üçün **/etc/init.d/tomcat** adlı script yaradırıq və məzmununa aşağıdakı sətirləri əlavə edirik(**JAVA_HOME** dəyişəninin ünvanını **OracleJDK8.docx** sənədi ilə Java 8-ci versiyanın yüklənməsindən əldə edəbilərsiniz):

```
#!/bin/sh
# chkconfig: 2345 80 20
# Description: Tomcat Start/Shutdown script

export JAVA_HOME=/usr/java/jdk1.8.0_45

case $1 in
```

```
start)
cd /opt/tomcat/bin/
./startup.sh
;;
stop)
cd /opt/tomcat/bin/
./shutdown.sh
;;
restart)
cd /opt/tomcat/bin/
./shutdown.sh
cd /opt/tomcat/bin/
./startup.sh
;;
esac
exit 0
```

Startup scriptimizi yerinə yetirilən edirik:

```
# chmoda+x /etc/init.d/tomcat
```

Yaratdığımız tomcat scriptini daemon siyahısına əlavə edirik:

```
# chkconfig --add tomcat
```

Tomcat-i işə salırıq:

```
# service tomcat start
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr/java/jdk1.8.0_45
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-
juli.jar
Tomcat started.
```

Tomcat daemon-u system yenidən yüklənməsinə əlavə edirik:

```
# chkconfig tomcat on
```

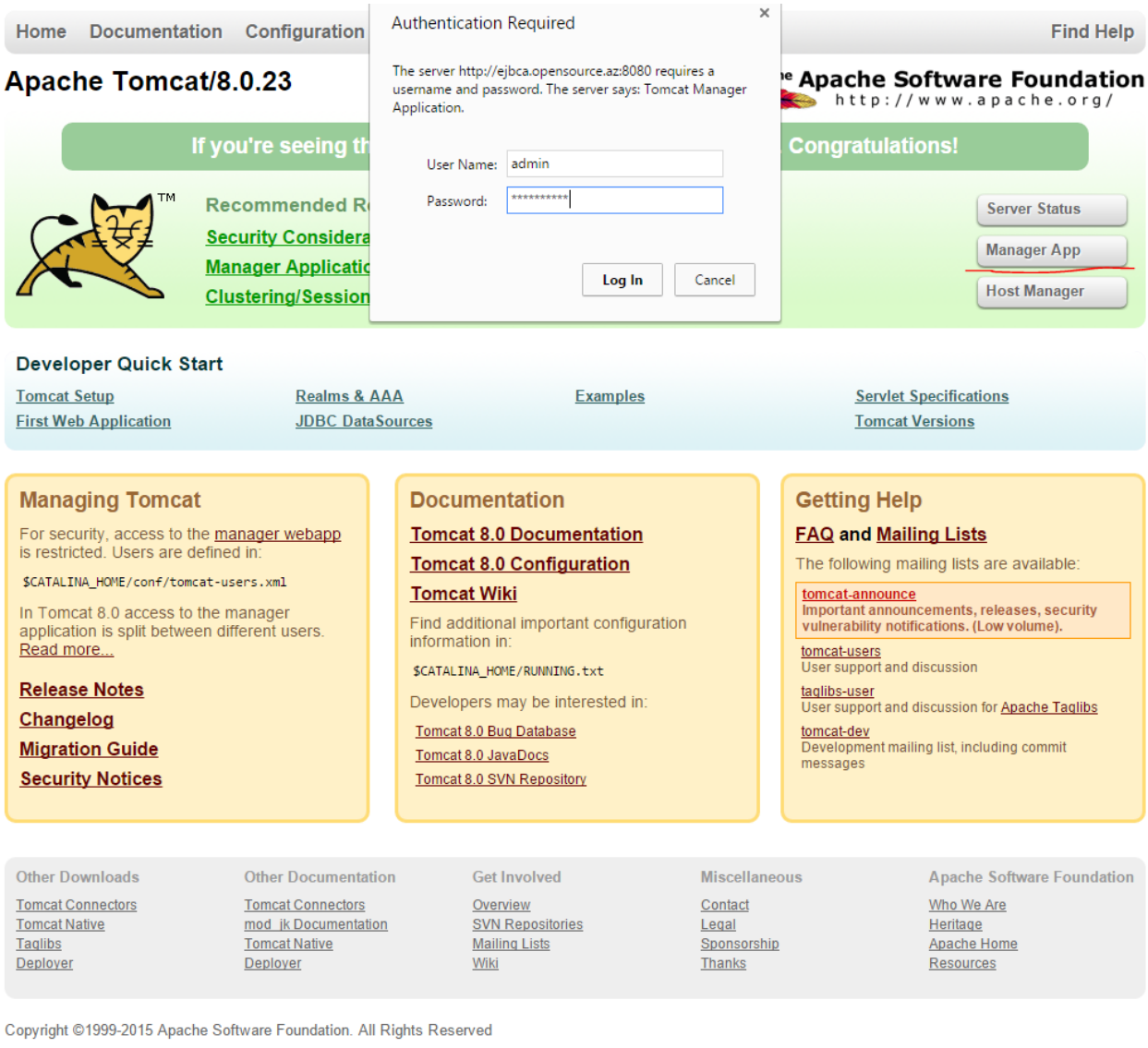
Tomcat manager role-unuyaratmaqüçün**$CATALINA_HOME/conf/tomcat-users.xml**faylınaaşağıdakısətirləri,**<tomcat-users>** … **</tomcat-users>**direktivlərіarasınaəlavəedibyaddasaxlayaraqcıxırıq:

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<user username="admin" password="t0mc@tp@$$" roles="manager-gui,manager-
script,manager-jmx,manager-status,admin-gui,admin-script"/>
```

Sonda tomcat daemonu yenidən işə salırıq:

```
# service tomcat restart
```

Sonda tomcat web serverimizə eb browser vasitəsilə daxil oluruq və aşağıdakı şəkildəki kimi, istifadəçi adı və şifrəni daxil edirik:



Uğurlu nəticə aşağıdakı şəkildəki kimi olmalıdır:

The Apache Software Foundation
http://www.apache.org/

Tomcat Web Application Manager

Message:	OK

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):
XML Configuration file URL:
WAR or Directory URL:
Deploy

WAR file to deploy

Select WAR file to upload Choose File No file chosen
Deploy

Apache ANT yüklənməsi və quraşdırılması

Java əmrlər sətiri üçün Apache ANT kitabxana və alətdir hansı ki, bir-birindən asılı olan genişlənmə nöqtələrinin yığım fayllarında yığım prosesini idarə edir. ANT-ın istifadə edilməsinin əsas səbəbi, Java proqramlarının yığılmasıdır. ANT çoxlu sayda daxili imkanlara malikdir ki, kompilyasiyaya şərait yaradır, test edir və java proqramlarını işə salır. Həmçinin ANT vasitəsilə qeyri java proqramlarını da kompilyasiya etmək mümkündür. Misal üçün C və C++ proqramlar üçün. http://www.us.apache.org/dist/ant/binaries/ səhifəsindən son versiyanı əldə edə bilərsiniz.

Apache ANT üçün ən yeni versiyanı internetdən endiririk:
wget http://mirror.sdunix.com/apache//ant/binaries/apache-ant-1.9.4-bin.zip

yada
wget http://mirror.sdunix.com/apache/ant/binaries/apache-ant-1.9.5-bin.zip

Endirdiyimiz zip faylı açırıq:
unzip apache-ant-1.9.5-bin.zip

Açılan qovluğu **/opt** qovluğun altına **ant** adı ilə köçürürük:
mv apache-ant-1.9.5/ /opt/ant

/opt/ant/bin/ant binar faylı sistem binar faylları üçün link edirik:
ln -s /opt/ant/bin/ant /usr/bin/ant

ant mühit dəyişənləri üçün **/etc/profile.d/ant.sh** scripti yaradırıq və məzmununa aşağıdakı sətirləri əlavə edirik:

```
#!/bin/bash
ANT_HOME=/opt/ant
PATH=$ANT_HOME/bin:$PATH
export PATH ANT_HOME
export CLASSPATH=.
```

Faylı yerinə yetirən edirik:
chmod +x /etc/profile.d/ant.sh

Apache Ant **tools.jar** faylını tələb edir və **"ant -version"** əmrini daxil etdikdə həmin səhvi çap edəcək. Bunu aşmaq üçün isə aşağıdakı əmrlə **java-devel** programını yükləmək lazımdır:

```
yum -y install `yum search java|grep java-1.7.0-openjdk-devel.$(uname -p) |
awk '{ print $1 }'`
```

Işə salırıq ki, sessiyamızda aktiv olsun:

```
source /etc/profile.d/ant.sh
```

Sonra sistemi yenidən yükləyib antın versiyasına aşağıdakı əmrlə baxırıq:

```
# ant -version
Apache Ant(TM) version 1.9.4 compiled on April 29 2014
```

Sistemdə olan əmrlər ünvanlarına baxırıq:

```
# echo $ANT_HOME
/opt/ant
```

```
# echo $PATH
/usr/lib64/qt-
3.3/bin:/opt/ant/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr
/bin:/root/bin
```

Apache Maven yüklənməsi və quraşdırılması

Apache MAVEN – proyektlərin idarə edilməsi və asan başa düşülməsi üçün istifadə edilən alətdir. Proyekt obyekti modelinə əsaslanır(PoM. Maven proyektləri yığa, hesabatları hazırlaya və mərkəzi inrofmasiya hissəsindən sənədləşmə işini görə bilir. Maven-i internetdən endiririk:

```
wget http://www.interior-dsgn.com/apache/maven/maven-3/3.3.3/binaries/apache-maven-3.3.3-bin.zip
```

Endirdiyimiz zip faylı açırıq:

```
unzip apache-maven-3.3.3-bin.zip
```

Açılan kontenti **/opt/maven** ünvanına köçürürük:

```
mv apache-maven-3.3.3 /opt/maven
```

Maven binar qovluğu üçün symlink yaradırıq:

```
ln -s /opt/maven/bin/mvn /usr/bin/mvn
```

Maven üçün mühit dəyişənləri yaradırıq. Bunun üçün **/etc/profile.d/maven.sh** scripti yaradırıq və məzmununa aşağıdakı sətirləri əlavə edirik:

```
#!/bin/bash
MAVEN_HOME=/opt/maven
PATH=$MAVEN_HOME/bin:$PATH
export PATH MAVEN_HOME
export CLASSPATH=.
```

Scripti yerinə yetirən edirik:

```
chmod +x /etc/profile.d/maven.sh
```

CLI-dan dəyişənləri işə salmaq üçün aşağıdakı əmri daxil edirik(Ancaq hər halda işləməsindən əmin olmaq üçün sistemi yenidən yükləyirik):

```
source /etc/profile.d/maven.sh
```

Maven versiyasına baxırıq:

```
# mvn -version
Apache Maven 3.3.3 (7994120775791599e205a5524ec3e0dfe41d4a06; 2015-04-22T16:57:37+05:00)
Maven home: /opt/maven
Java version: 1.7.0_79, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.79.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "2.6.32-504.16.2.el6.x86_64", arch: "amd64", family: "unix"
```

Maven mühit dəyişənlərini yoxlayırıq:

```
# echo $MAVEN_HOME
/opt/maven
# echo $PATH
/usr/lib64/qt-3.3/bin:/opt/maven/bin:/opt/ant/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

CentOS PDO_OCI inteqrasiyası

Nəzərdə tutulur ki, artıq **CentOS PHP5-FPM nGinx** başlığında olan bütün işlər görülüb artıq. PHP Data Objects (PDO) - PHP üçün genişlənmədir və bir çox proqramçılar tərəfindən PDO istifadə edilir. Buna görə də siz PDO-nun OCİ ilə inteqrasiya edilməsi tələbi ilə qarşılaşa bilərsiniz. Bu başlıq PDO OCİ inteqrasiyasını açıqlayır.

Proqramlaşdırma ücün tələb edilən bütün paketləri yükləyirik:

```
# yum install php-pear php-devel zlib zlib-devel bc libaio glibc
# yum groupinstall "Development Tools"
```

Oracle client ünvanını link edək ki, 32 bitlik kimi görünsün:

```
# ln -s /usr/include/oracle/11.2/client64 /usr/include/oracle/11.2/client
# ln -s /usr/lib/oracle/11.2/client64 /usr/lib/oracle/11.2/client
```

/etc/profile.d/oracle.sh adlı fayl yaradıb içinə aşağıdakı sətri əlavə edirik(Bu sətir Oracle kitabxanaları yerləşən ünvanın dəyişənini təyin edir. Siz bu ünvanı "**CentOS-nGinx-phphpm-oci8.docx**" sənədində görəcəksiniz):

```
#!/bin/bash
export ORACLE_HOME=/usr/lib/oracle/11.2/client64
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export C_INCLUDE_PATH=/usr/include/oracle/11.2/client64
export NLS_LANG=AMERICAN_AMERICA.AL32UTF8
```

Qeyd: **NLS_LANG** dəyişənin dəqiq mənasını **phpinfo()** funksiyası ilə axtarıb tapa bilərsiniz.

Faylı tez işə salaq ki, dəyişənimiz işə düşsün:

```
# source /etc/profile.d/oracle.sh
```

PDO_OCI

Pecl istifadə edərək PDO_OCI-ni endirək:

```
# mkdir /root/pdooci ; cd /root/pdooci
# pecl download PDO_OCI
# tar -xvf PDO_OCI-1.0.tgz
# cd PDO_OCI-1.0
```

PDO_OCI-1.0 qovluğunun içində **config.m4** adlı faylda dəyişiklik edirik və təqribən 10-cu sətirdən sonra uyğun ardıcıllıqda gedən digər sətirlərin əvvəlinə aşağıdakı sətirləri əlavə edirik(Diqqətlə fikir verin 11.2 bizim Oracle Clientin versiyası olduğuna görə burda da 11.2 istifadə edirik):

```
elif test -f $PDO_OCI_DIR/lib/libclntsh.$SHLIB_SUFFIX_NAME.11.2; then
  PDO_OCI_VERSION=11.2
```

Həmçinin aşağıdakı sətirlərə uyğun olan ərazini tapıb **10.2**-dən sonra əlavə edirik(təqribən 101-ci sətirə yaxın olan bir ərazidir) və uyğun ardıcıllıqda aşağıdakı iki sətiri əlavə edirik(Versiya **11.2**-dir):

```
11.2)
  PHP_ADD_LIBRARY(clntsh, 1, PDO_OCI_SHARED_LIBADD)
  ;;
```

Genişlənməni kompilyasiya edək və yükləyək:

```
# phpize
# ./configure --with-pdo-oci=instantclient,/usr,11.2
```

Ardınca pdo_oci.c faylında aşağıdakı sətirləri dəyişib:

```
/* {{{ pdo_oci_functions[] */
function_entry pdo_oci_functions[] = {
    {NULL, NULL, NULL}
};
/* }}} */
```

Bu formaya gətiririk:

```
/* {{{ pdo_oci_functions[] */
zend_function_entry pdo_oci_functions[] = {
    {NULL, NULL, NULL}
};
/* }}} */
```

Sonra **/root/pdooci/PDO_OCI-1.0/oci_statement.c** faylında **oci_blob_write** və **oci_blob_read** funksiyalarının içində olan aşağıdakı şərti dəyişərək!:

```
if (r != OCI_SUCCESS) {
  return (size_t)-1;
}
```

Əvəz edirik buna:

```
if ((r != OCI_SUCCESS) && (r != OCI_NEED_DATA)) {
  return (size_t)-1;
}
```

Qeyd: Bu sizin php kodlarınızda oracle verilənlər bazasından blob datanın əldə edilməsində düzgün simvol kodirovkasının seçilməsinə kömək olacaq. Həmin sətir **oci:dbname=HOST/TNS_NAME;charset=AL32UTF8** şəklində olmalıdır.

```
# make            - Kompilyasiya edirik
# make install    - Yükləyirik
# make test       - Əmri işə salaraq yoxlayırıq(mənim halımda /etc/php.ini
                    faylında disable_functions-da proc_open funksiyası bağlı
                    idi və ona görə aşağıdakı səhvi çap elədi)
+------------------------------------------------------------+
|                       ! ERROR !                            |
| The test-suite requires that proc_open() is available.     |
| Please check if you disabled it in php.ini.                |
+------------------------------------------------------------+
```

php.ini faylından **proc_open**-i **disable_functions**-dan sildikdən sonra yenidən əmri işə salırıq və aşağıdakı nəticəni əldə etmiş oluruq:

```
# make test
Build complete.
Don't forget to run 'make test'.
=====================================================================
PHP         : /usr/bin/php
CWD         : /root/pdooci/PDO_OCI-1.0
Extra dirs  :
```

```
VALGRIND    : Not used
=====================================================================
TIME START 2015-07-28 09:04:48
=====================================================================
No tests were run.
```

make install - Yükləyirik

```
Installing shared extensions:     /usr/lib64/php/modules/
```

Sonra **/etc/php.d/pdo_oci.ini** faylı yaradıb içinə aşağıdakı sətiri əlavə edirik:

extension=pdo_oci.so

Uğurlu yüklənməsini aşağıdakı əmrlə yoxlayırıq(Oxşar sətirləri görməliyik):

```
# php -i | grep oci
/etc/php.d/pdo_oci.ini,
PDO drivers => mysql, oci, odbc, sqlite
```

OCI8

pear istifadə edərək, OCI8-i endirək.

```
# pear download pecl/oci8
# tar -xvf oci8-1.4.9.tgz
# cd oci8-1.4.9
```

Genişlənməni kompilyasiya edək və yükləyək:

```
# phpize
# ./configure --with-
oci8=shared,instantclient,/usr/lib/oracle/11.2/client64/lib
# make
# make install
```

Genişlənməni işə salmaq üçün, **/etc/php.d/oci8.ini** faylına aşağıdakı sətiri əlavə edirik:

extension=oci8.so

Uğurla yüklənməsini yoxlayaq:

```
# php -i | grep oci8
```

Aşağıdakı sətirlərə oxşar bir sətir əldə etməlisiniz:

```
/etc/php.d/oci8.ini,
oci8
oci8.connection_class => no value => no value
oci8.default_prefetch => 100 => 100
oci8.events => Off => Off
oci8.max_persistent => -1 => -1
oci8.old_oci_close_semantics => Off => Off
oci8.persistent_timeout => -1 => -1
oci8.ping_interval => 60 => 60
oci8.privileged_connect => Off => Off
oci8.statement_cache_size => 20 => 20
```

Oracle JDK8-in yüklənməsi və quraşdırılması

Oracle-ın Java proqramçılar üçün xüsusi alətlər toplusu olan bir yığılması mövcuddur. Əksər proqramçılar bunu istifadə edir. Bu başlıq Oracle Java Development Kit-in quraşdırılmasını açıqlayır. Sistemin paketlərini reposlardan yeniləyirik:

```
yum update -y
```

Sistemimizdə yüklənmiş olan JDK versiyalarını çap edirik:

```
# rpm -qa | grep -E '^open[jre|jdk]|j[re|dk]'
perl-Object-Accessor-0.34-136.el6_6.1.x86_64
libbasicobjects-0.1.1-11.el6.x86_64
java-1.7.0-openjdk-devel-1.7.0.79-2.5.5.3.el6_6.x86_64
java-1.7.0-openjdk-1.7.0.79-2.5.5.3.el6_6.x86_64
openjpeg-libs-1.3-10.el6_5.x86_64
eject-2.1.5-17.el6.x86_64
java-1.6.0-openjdk-1.6.0.35-1.13.7.1.el6_6.x86_64
```

Java versiyasına baxırıq:

```
# java -version
java version "1.7.0_79"
OpenJDK Runtime Environment (rhel-2.5.5.3.el6_6-x86_64 u79-b14)
OpenJDK 64-Bit Server VM (build 24.79-b02, mixed mode)
```

Öncədən sistemə yüklənmiş olan 1.6 və 1.7-ci versiyanı silmək üçün aşağıdakı əmrdən istifadə etmək lazımdır:

```
# yum remove java-1.6.0-openjdk
# yum remove java-1.7.0-openjdk
```

Oracle rəsmi saytından http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html ən son 8-ci versiyanı endirib serverimizə WinSCP vasitəsilə yükləyirik.

Endirdiyimiz RPM paketi serverə yükləyirik:

```
# rpm -ivh jdk-8u45-linux-x64.rpm
Preparing...                ########################################### [100%]
   1:jdk1.8.0_45            ########################################### [100%]
Unpacking JAR files...
        rt.jar...
        jsse.jar...
        charsets.jar...
        tools.jar...
        localedata.jar...
        jfxrt.jar...
        plugin.jar...
        javaws.jar...
        deploy.jar...
```

Yüklənmiş java versiyasına baxırıq:
java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)

Java mühit dəyişənlərinin işləməsi üçün **/etc/profile.d/java.sh** faylı yaradırıq və məzmununa aşağıdakı sətirləri əlavə edirik:
#!/bin/bash
JAVA_HOME=/usr/java/jdk1.8.0_25/
PATH=$JAVA_HOME/bin:$PATH
export PATH JAVA_HOME
export CLASSPATH=.

Yaratdığımız faylı yerinə yetirən edirik:
chmod +x /etc/profile.d/java.sh

Mühit dəyişənlərini işə salmaq üçün scripti seansımızda işə salırıq:
source /etc/profile.d/java.sh

Əgər siz köhnə versiyaları silməsəydiniz nə baş verərdi?

Əgər siz sistemdə olan köhnə versiyaları öncədən silməsəniz, onda siz sisteminizdə java ilə işləyəcək proqramların hansı java versiyası üzərindən işləməsini bildirməlisiniz. Susmaya göre **JDK1.8.x** paketi **/usr/java/jdk1.8.0_25/** ünvanına yüklənəcək. Sisteminizə Javanın hansı ünvandan işə düşməsini bildirmək üçün səliqə ilə aşağıdakı ardıcıllıqla addımları yerinə yetirmək lazımdır:
alternatives --install /usr/bin/java java /usr/java/jdk1.8.0_25/jre/bin/java 20000
alternatives --install /usr/bin/jar jar /usr/java/jdk1.8.0_25/bin/jar 20000
alternatives --install /usr/bin/javac javac /usr/java/jdk1.8.0_25/bin/javac 20000
alternatives --install /usr/bin/javaws javaws
/usr/java/jdk1.8.0_25/jre/bin/javaws 20000
alternatives --set java /usr/java/jdk1.8.0_25/jre/bin/java
alternatives --set jar /usr/java/jdk1.8.0_25/bin/jar
alternatives --set javac /usr/java/jdk1.8.0_25/bin/javac
alternatives --set javaws /usr/java/jdk1.8.0_25/jre/bin/javaws

Bitdi və alternative-ləri yoxlayırıq:
ls -lA /etc/alternatives/
lrwxrwxrwx. 1 root root 29 May 31 19:29 jar -> /usr/java/jdk1.8.0_45/bin/jar
lrwxrwxrwx. 1 root root 34 May 31 19:29 java ->
/usr/java/jdk1.8.0_45/jre/bin/java
lrwxrwxrwx. 1 root root 31 May 31 19:29 javac ->
/usr/java/jdk1.8.0_45/bin/javac
lrwxrwxrwx. 1 root root 32 May 31 19:29 javaws ->
/usr/java/jdk1.8.0_45/bin/javaws

Nəticədə Java versiyasına baxırıq:
java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)

Ubuntu 14.04 x64 tomcat7 Java8 yüklənməsi və quraşdırılması

Məqsədimiz Ubuntu-un öz repositorylərində olan oracle java yükləyicisinin yüklənməsidir.

```
apt-get update                # reposları yeniləyirik
apt-get dist-upgrade          # paketləri ən son versiyaya yeniləyirik

apt-get install tomcat7                    # Tomcat7-ni yükləyirik
apt-get install tomcat7-docs tomcat7-admin tomcat7-examples # Tomcat
                                           sənədləri və misallarını yükləyirik

vi /etc/tomcat7/tomcat-users.xml           # Tomcat web management-ə
                                           istifadəçi əlavə edirik
<tomcat-users>
    <user username="admin" password="freebsd" roles="manager-gui,admin-gui"/>
</tomcat-users>


add-apt-repository ppa:webupd8team/java    # Oracle reposu əlavə edirik
apt-get update                             # Reposları yeniləyirik
apt-get install oracle-java8-installer     # Java8-i yükləyirik
apt-get install oracle-java8-set-default   # Java8-i susmaya görə elan edirik
```

Ubuntu Tomcat serverdə http və https portların dəyişdirilməsi

Məqsədimiz Tomcat serverin ən-ənəvi 80 və 443-cü portda qulaq asmasının quraşdırılmasıdır.

```
apt-get update                  # Reposları yeniləyirik
apt-get dist-upgrade            # Sistemdə olan paketləri və kerneli
                                yeniləyirik
```

```
apt-get install `apt-cache search tomcat7 | awk '{ print $1 }'`# Tomcat7 və
                                                               ona aid olan
                                                               digər paketlərin
                                                               hamısını
                                                               yükləyirik
```

/usr/share/tomcat7/bin/catalina.sh faylında **JAVA_OPTS** dəyişəninə **-Djava.net.preferIPv4Stack** əlavə edirik. Aşağıdakı kimi:
JAVA_OPTS="-Djava.net.preferIPv4Stack"

/etc/tomcat7/tomcat-users.xml faylında **<tomcat-users>** seksiyasının daxilinə aşağıdakı sətirə uyğun olaraq istifadəçi, şifrə və yazımı yetki veririk(aşağıdakı kimi):

```
<tomcat-users>
      <user username="admin" password="freebsd" roles="tomcat,manager-
script,manager-gui"/>
</tomcat-users>
```

Tomcat7 susmaya görə http üçün 8080-ci portda və https üçün 8443-cü portda qulaq asır. Ancaq bunu dəyişib **80** və **443** eləmək olar.
Bunun üçün aşağıdakıları edirik:
/etc/sysctl.conf faylına aşağıdakı sətiri əlavə edirik:
net.ipv6.conf.all.disable_ipv6=1

```
sysctl net.ipv6.conf.all.disable_ipv6=1          # CLI-dan işə salırıq
```

/etc/default/tomcat7 faylında AUTHBIND sətirini aşağıdakı kimi edirik:
AUTHBIND=yes

Sonra AuthBind üçün lazımı portların fayllarını və yetkilərini veririk ki, portumuz qulaq asa bilsin:

```
touch /etc/authbind/byport/80
touch /etc/authbind/byport/443
chmod 0755 /etc/authbind/byport/80
chmod 0755 /etc/authbind/byport/443
chown tomcat7:tomcat7 /etc/authbind/byport/80
chown tomcat7:tomcat7 /etc/authbind/byport/443
```

/etc/tomcat7/server.xml faylında da 8080,8443-cü portları aşağıdakı kimi dəyişib 80,443 edirik:

```
      <Connector port="80" protocol="HTTP/1.1"
                 connectionTimeout="20000"
                 URIEncoding="UTF-8"
                 redirectPort="443" />


# Bu sətir HTTPS üçün JKS istifadə biz onu aşağıdakı config edəcəyik.
      <Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
                maxThreads="150" scheme="https" secure="true"
                clientAuth="false" sslProtocol="TLS"
                keystoreFile="/etc/tomcat7/srccodes.jks"
                keystoreType="JKS"
                keystorePass="javapass"
                keyPass="javapass" />
```

Indi isə tomcat7 https üçün **keystore** və Self Signed Certificate yaradaq. Bunun üçün **/etc/tomcat7** ünvanına daxil oluruq(JKS and Cert pass: **javapass**):

```
cd /etc/tomcat7
keytool -genkey -alias srccodes -keyalg RSA -keystore srccodes.jks
Enter keystore password: javapass
Re-enter new password: javapass
What is your first and last name?
  [Unknown]:  Jamal Shahverdiyev
What is the name of your organizational unit?
  [Unknown]:  Statistika
What is the name of your organization?
  [Unknown]:  DOMAIN
What is the name of your City or Locality?
  [Unknown]:  Yasamal
What is the name of your State or Province?
  [Unknown]:  Baku
What is the two-letter country code for this unit?
  [Unknown]:  AZ
Is CN=Jamal Shahverdiyev, OU=Statistika, O=DOMAIN, L=Yasamal, ST=Baku, C=AZ
correct?
  [no]:  yes

Enter key password for <srccodes>
        (RETURN if same as keystore password):
```

Beləliklə **/etc/tomcat7/server.xml** faylında göstərilən **keystoreFile="/etc/tomcat7/srccodes.jks"** fayl ünvanı və şifrəni dəqiq təyin etməyi unutmayın.

BÖLÜM 12

Proqramçıların effektiv iş mühitləri

- **Mercurial Active Directory ilə inteqrasiyası**
- **GitLAB Active Directory inteqrasiyası**

Əgər şirkətinizin daxili proqramlaşdırma şöbəsi varsa və proqramçıların bir neçəsi eyni zamanda eyni layihə üzərində işləyirsə, müəyyən mübahisələr yarana bilər. Məsələ ondan ibarətdir ki, proqramçılardan biri hansısa kodu dəyişdikdə, bir neçə vaxtdan sonra onun kimin tərəfindən dəyişildiyi və əvvəlki vəziyyəti haqqında olan məlumatı tapmaq əsl problemə çevrilir. Bu tip problemlərin aradan qaldırılması üçün mərkəzi sistemlər olur və kodlar həmin mərkəzdə qalır. Hər bir şəxs öz hesabı ilə daxil olub fərqi qeyd edir və fərqin kimin tərəfindən dəyişildiyi görünür. Bu başlıqda uyğun sistemlərin qurulması haqqında danışılacaq.

Mercurial Active Directory ilə inteqrasiyası

Mercurial - Eynilə HG, çox böyük kod repositoriyalarla effektiv işləmək, versiyaların idarə edilməsi üçün yaradılan çox platformalı paylaşılmış sistemdir. Konsol proqramıdır. Proqramçılar üçün tələb edilir.

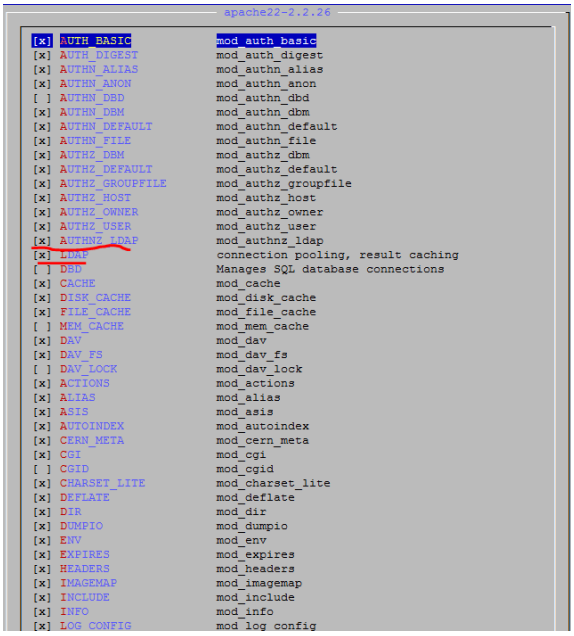
Nəzərdə tutulur ki, Domain controller artıq qurulub və aşağıdakı verilənlərlə yaradılmışdır.

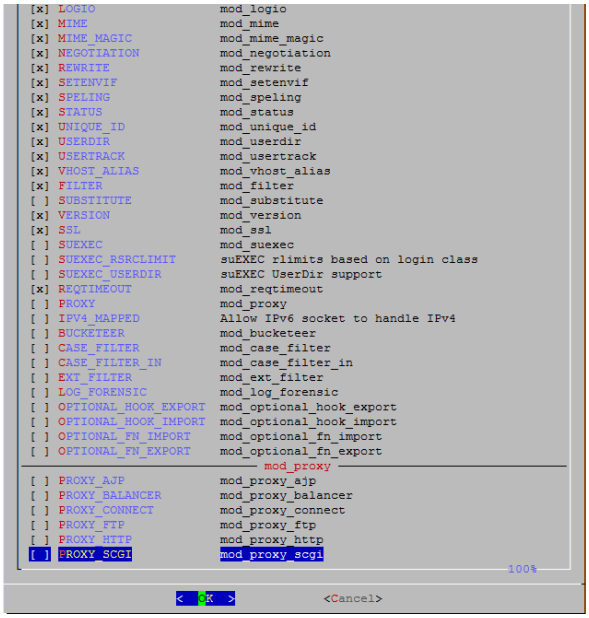
FreeBSD9.2 x64(10.10.10.210 - VmNet4)
FreeBSD maşında DNS resolver kimi Active Directory istifadə edilir.

```
cat /etc/resolv.conf
nameserver  10.10.10.200
```

DC: **mercurial.lan** **(10.10.10.200 - Vmnet4)**
OU: **mercurial**
Group: **mercurial**
2 ədəd istifadəçimiz var: **jamal** və **salman** (Istifadəçilər **mercurial** organization unit-indədirlər və **mercurial** qrupunun üzvüdürlər).
Istifadəciləri mercurial qrupunun üzvü ona görə edirik ki, apache22 yalnız bu qrupun üzvlərinə mercurial səhifəsinə girişə izin verəcək.

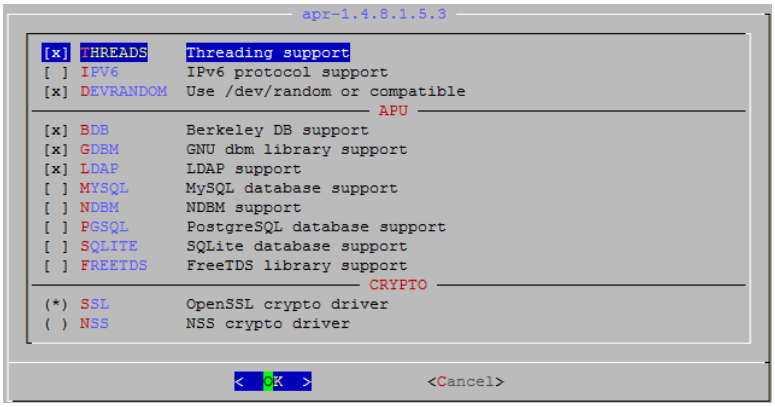
```
portsnap fetch extract update                # Ilk öncə portları yeniləyək.
reboot                                       # sistemi restart edirik ki, portlar bazası yenilənsin

root@mercuri:~ # cd /usr/ports/www/apache22  # Apache22-nin port ünvanına daxil oluruq.
root@mercuri:/usr/ports/www/apache22 # make config   # Lazımi modulları seçirik.
```






```
root@mercuri:/usr/ports/www/apache22 # make install          # yükləyirik
```

Yüklənmə prosedurunda həmçinin **apr1** modulunuda seçirik.



Mercurial-i hazırlayaq

```
root@mercuri:/ # cd /usr/ports/devel/mercurial && make install clean    #
                                                                        Mercurial-ı
                                                                        yükləyək

root@mercuri:/ # cd /usr/ports/devel/py-mercurialserver && make install clean
                                                  # Lazımi componentləri yükləyirik

root@mercuri:/ # cd /usr/ports/www/mod_wsgi3 && make install clean      #
                                                                    wsgi işləməsi
                                                                    üçün apache modul
```

Qovluq yaradaq hansı ki, merkurialın quraşdırma fayları saxlanılacaq.

```
root@mercuri:/ # mkdir /usr/local/www/hg
```

```
root@mercuri:/ # cp /usr/local/share/mercurial/www/hgweb.wsgi
/usr/local/www/hg/


hgweb.cgi-ın quraşdırmalarını redakte edirik.
ee /usr/local/www/hg/hgweb.wsgi
config = "/usr/local/www/hg/hgweb.config" # Config fayl üçün yolu dəyişirik.

ee /usr/local/www/hg/hgweb.config    # hgweb.config faylının tərkibini
                                     aşağıdakı kimi edirik.
[web]
allow_push = *
push_ssl = false

[trusted]
users = *

[collections]
/usr/local/www/hg/repos = /usr/local/www/hg/repos


mkdir /usr/local/www/hg/repos              # Repos üçün qovluq yaradırıq

chown -R www:www /usr/local/www/hg         # Lazımi yetkiləri veririk


Apache-i quraşdırırıq.
echo 'apache22_enable="YES"' >> /etc/rc.conf     # Startup-a əlavə edirik
echo "Include /usr/local/domen/*" >> /usr/local/etc/apache22/httpd.conf
                                                 # Yeni Include əlavə edirik.
root@mercuri:/ # mkdir /usr/local/domen/         # Include üçün qovluq
                                                 yaradırıq.

root@mercuri:/ # cat /usr/local/domen/mercuri.az       # Virtual mercuri.az
domain
                                                       contenti aşağıdakı kimi
                                                       edirik
<VirtualHost *>
        ServerName mercuri.az
        ServerAlias www.mercuri.az
        DocumentRoot /usr/local/www/hg
        ErrorLog /var/log/mercuri-error.log
        CustomLog /var/log/mercuri-access.log common
        WSGIScriptAlias / /usr/local/www/hg/hgweb.wsgi
<Directory "/usr/local/www/hg">
        AllowOverride None
        order allow,deny
        Allow from all
</Directory>
<Location />
        AuthType Basic
        AuthBasicProvider ldap
        AuthBasicAuthoritative off
```

```
        AuthName "ENTER YOUR AD LOGIN & PASSWD"

        AuthLDAPURL
"ldap://mercurial.lan:389/DC=mercurial,DC=lan?sAMAccountName?sub?(objectClass
=*)"
        AuthLDAPBindDN  "administrator@mercurial.lan"
        AuthLDAPBindPassword "Zumrud123"
        Require ldap-group cn=mercurial,ou=mercurial,dc=mercurial,dc=lan
</Location>
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
</VirtualHost>
```

```
touch /var/log/mercuri-error.log /var/log/mercuri-access.log # Lazımi jurnal
                                                              faylları yaradırıq.

root@mercuri:/ # chown -R www:www /usr/local/domen/    # Lazımi yetkiləri
                                                       veririk.
```

Sonda **/usr/local/etc/openldap/ldap.conf** faylında aşağıdakı sətiri əlavə edirik və **apache22**-ni işə salırıq.

```
echo "REFERRALS off" >> /usr/local/etc/openldap/ldap.conf
```

```
root@mercuri:/ # /usr/local/etc/rc.d/apache22 start    # Apache-ı işə salırıq.
```

Sonda eyni şəbəkədə olan client-də **c:\windows\system32\drivers\etc\hosts** faylına aşağıdakı sətiri əlavə edib browserdə **mercuri.az** domain-i **jamal** adlı istifadəçi ilə test etməyiniz yetər.

Debug etmək üçün isə **/usr/local/etc/apache22/httpd.conf** faylının içinə **LogLevel debug** əlavə edib daemonu restart etdikdən sonra **/var/log/mercuri-error.log** faylını araşdırmanız lazımdır.

GitLAB Active Directory inteqrasiyası

Məqsədimiz proqramçılar üçün Ubuntu 14.04 x64 OS üzərində source code-ların yerləşməsi və sinxronizasiyası üçün server qurmaqdır. Bu WEB serverdir və idarəetməsi çox asandır. Proqramçılar öz mənbə kodlarını bu serverə git client ilə sinxronizasiya edir. Code-lar diff və checksum-a görə yoxlanısh edilir. WEB portalda qrup yaradılır və bu qrupa proqramistlər təyin edilir. Eyni code-da edilən deyişikliklərin yalnız dəyişmiş hissəsi sinxronizasiya edilir və jurnallanır. Bir sözlə proqramçılar üçün can dərmanıdır ☺.

GitLab - Web bazalı wiki və hadisələrin izlənilməsi imkanı ilə olan Git repository idarəedicisidir. Proqram Ukrainalı Dmitriy Zaporozhets tərəfindən Ruby-də yazılmışdır.

Qurulmasına başlayaq. Öncədən aşağıdakı verilənləri nəzərə alaq:

```
DC: DOMAIN.LAN
port: 636
bind_dn: 'CN=DCADM,CN=Users,DC=domain,DC=lan'
password: 'DC_PASSWORD'
user_filter: '(memberOf=CN=GITUsers,OU=DOMAINTech
Groups,OU=DOMAINTech,DC=domain,DC=lan)'
```

Ubuntu 14.04 x64 üçün virtual mühitdə 2 CPU 2 Core, 4GB DDR ve 200GB HDD ayrılmışdır.

1. Paketlər və asılılıqlar

```
apt-get update          # Sistem yüklənikdən sonra apt reposları yeniləyirik
apt-get dist-upgrade    # Sistem yüklənikdən sonra sistem paketləri və
                        kerneli yeniləyirik
```

Sənədə diqqətlə baxın və mütləq **sudo** olan yerlərdə, yüklənməli **root** adından etməyin.

```
sudo apt-get install -y vim          # VIM-i yükləyirik
sudo update-alternatives --set editor /usr/bin/vim.basic   # VIM-i susmaya
                                                            görə olan fayl
                                                            editor təyin
                                                            edirik
```

```
# Ruby və Ruby GEMS genişlənmələri üçün tələb edilən paketləri yükləyək.
sudo apt-get install -y build-essential zlib1g-dev libyaml-dev libssl-dev
libgdbm-dev libreadline-dev libncurses5-dev libffi-dev curl openssh-server
redis-server checkinstall libxml2-dev libxslt-dev libcurl4-openssl-dev
libicu-dev logrotate python-docutils pkg-config cmake libkrb5-dev


sudo apt-get install -y git-core          # GIT-i yükləyirik
git --version     # GIT versiyasına baxırıq, mütləq 1.7.12 yuxarı və 2.0.0
                  aralığında olmalıdır
git version 1.9.1
```

Yox əgər siz yenede kohnelmish GIT-i silib yeniden source code-lardan yüklənməsini isteseniz, onda aşağıdakı addimlarla once giti silirik ve code-lardan yükləyirik.

```
sudo apt-get remove git-core  # Öncə yüklənmiş GIT-core-u silirik
```

Tələb edilən asılılıq paketlərini yükləyirik.

```
sudo apt-get install -y libcurl4-openssl-dev libexpat1-dev gettext libz-dev
libssl-dev build-essential

cd /tmp      # Qaynaq kodu yükləyib kompilyasiya etmək üçün /tmp qovluğuna
             daxil oluruq
```

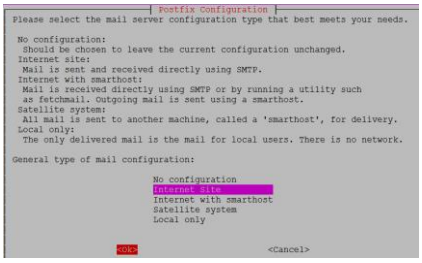
Qaynaq kodu dartırıq və açırıq

```
curl -L --progress https://www.kernel.org/pub/software/scm/git/git-
2.1.2.tar.gz | tar xz

cd git-2.1.2/            # GIT code-ların qovluğuna daxil oluruq
./configure              # Kompilyasiya üçün quraşdırırıq
make prefix=/usr/local all     # mənsəb ünvanı olaraq /usr/local təyin edilir

sudo make prefix=/usr/local install        # /usr/local/bin ünvanına
                                           yüklənilir GIT
```

Qeyd: **5-ci hissədə** olan quraşdırmalarımızda **config/gitlab.yml** config faylında git başlığını **bin_path** üçün aşağıdakı kimi etməyi unutmayın:

```
git:
    bin_path: /usr/local/bin/git
```

```
sudo apt-get install -y postfix            # mail server yükləyirik ki, mail
                                           yollaya bilək. Aşağıdakı kimi
                                           quraşdırırıq. Internet site seçirik
                                           və domain adını daxil edirik
```



2. Ruby-ni yükləyək

GitLab Shell OpenSSH ilə çağırılır və mövcud olan versiya manager-in SSH ilə ötürüb qəbul edilməsinin qarşısını almaq olur. Versiya managerləri dəstək edilmir və buna görə məsləhət görülür ki, mütləq ruby istifadə edəsiniz. Əgər köhnə ruby varsa onu silirik.

```
sudo apt-get remove ruby1.8          # köhnə ruby-ni silirik

mkdir /tmp/ruby && cd /tmp/ruby      # Ruby-ni dartıb kompilyasiya etmək üçün
                                     qovluq yaradırıq və içinə daxil oluruq
```

Dartırıq və yerləşdiyimiz qovluqda açırıq

```
curl -L --progress http://cache.ruby-lang.org/pub/ruby/2.1/ruby-2.1.5.tar.gz
| tar xz

cd ruby-2.1.5/           # Açdığımız qovluğa daxil oluruq
./configure --disable-install-rdoc  # Compilyasiya üçün quraşdırırıq
make                     # Kompilyasiya edirik
sudo make install        # yükləyirik

sudo gem install bundler --no-ri --no-rdoc      # Bundler GEM-i yükləyək
```

3. Sistem istifadəçiləri

```
sudo adduser --disabled-login --gecos 'GitLab' git    # GitLab üçün git adlı
                                                       istifadəçi yaradaq
```

4. Verilənlər bazasi

GitLAB özu baza olaraq PostgreSQL məsləhət görur. Genişlənmələrin istifadə edilməsi üçün isə PostgreSQL9.1 tələb edilir. PostgreSQL yükləyək, baza və istifadəçi yaradaq.

```
sudo apt-get install -y postgresql postgresql-client libpq-dev    # Baza üçün
                                                                  paketləri
                                                                  yükləyək

sudo -u postgres psql -d template1              # PostgreSQL-le daxil oluruq
template1=# CREATE USER git CREATEDB;           # git adlı baza istifadəçisi
                                                yaradırıq (template1=#
                                                console prompt-dur və o əmr
                                                kimi daxil edilə bilməz)

template1=# CREATE DATABASE gitlabhq_production OWNER git;        # Gitlab
                                                                  production
                                                                  bazası
                                                                  yaradılır
                                                                  və bu baza
                                                                  üçün tam
                                                                  yetki
                                                                  verilir

template1=# \q          # Bazadan çıxırıq

sudo -u git -H psql -d gitlabhq_production      # Yeni bazaya yeni istifadəçi
                                                ilə qoşulmağa calışırıq

gitlabhq_production=> \q      # Baza sessiyasından çıxırıq
```

5. Redis

```
sudo apt-get install redis-server   # Redis serverin paketini yükləyirik
```

```
sudo cp /etc/redis/redis.conf /etc/redis/redis.conf.orig          # Redis-i
                                                                  socket-
                                                                  lərin
                                                                  istifadə
                                                                  edilməsi
                                                                  üçün izin
                                                                  veririk


# Redis-in TCP-də qulaq asmasını dayandırmaq üçün portunu 0-ir təyin edirik.
sed 's/^port .*/port 0/' /etc/redis/redis.conf.orig | sudo tee
/etc/redis/redis.conf

# Susmaya görə olan Debian/Ubuntu üçün Redis socket-i işə salırıq
echo 'unixsocket /var/run/redis/redis.sock' | sudo tee -a
/etc/redis/redis.conf

# Redis qrup-da olan hər kəs üçün socket-ə yetki veririk
echo 'unixsocketperm 770' | sudo tee -a /etc/redis/redis.conf

# Socket-in yerləşməsi üçün qovluq yaradaq, lazımi istifadəçi və qrupa
mənimsədib, yetkini verək
sudo mkdir /var/run/redis
sudo chown redis:redis /var/run/redis
sudo chmod 755 /var/run/redis/

# Əgər özündə socket saxlayan qovluq varsa, saxla
if [ -d /etc/tmpfiles.d ]; then
   echo 'd  /var/run/redis  0755  redis  redis  10d  -' | sudo tee -a
/etc/tmpfiles.d/redis.conf
fi

sudo service redis-server restart          # redis.conf-da olan dəyişiklikləri
                                           servisi restart edərək işə salaq

sudo usermod -aG redis git          # git useri redis qrupa əlavə edək
```

6. GitLab(yükləyək ve config edek)

```
cd /home/git      # GitLab-ı git istifadəçisinin ev qovluğuna yükləyəcəyik.
                  Buna görə də bu qovluğa daxil oluruq

Source code-u clone edirik
# GitLsb reposu Clone edək
sudo -u git -H git clone https://gitlab.com/gitlab-org/gitlab-ce.git -b 7-6-
stable gitlab


Config edek
cd /home/git/gitlab           # GitLab yüklənməsi qovluğuna gedək
sudo -u git -H cp config/gitlab.yml.example config/gitlab.yml # nüsxə
                                                        faylından 1 nüsxə
```

çıxarırıq

```
sudo -u git -H editor config/gitlab.yml          # Quraşdırma faylının
                                                 əvvəlində quraşdırmaları
                                                 aşağıdakı kimi edirik. Nəzərə
                                                 alaq ki, HTTPS üçün nginx-i
                                                 birazdan quraşdıracayıq.
  gitlab:
    host: git.domain.lan
    port: 443
    https: true
    email_from: jamal.shahverdiev@gmail.com


# Əmin olaq ki, log/ və tmp/ qovluqlarına yazmaq yetkisi var
sudo chown -R git log/
sudo chown -R git tmp/
sudo chmod -R u+rwX,go-w log/
sudo chmod -R u+rwX tmp/

# Satellite üçün qovluq yaradaq və yetki verek.
sudo -u git -H mkdir /home/git/gitlab-satellites
sudo chmod u+rwx,g=rx,o-rwx /home/git/gitlab-satellites

# Əmin olaq ki, tmp/pids/ ve tmp/sockets/ qovluqlarina GitLab yazma yetkisinə
sahibdir.
sudo chmod -R u+rwX tmp/pids/
sudo chmod -R u+rwX tmp/sockets/

# Əmin olaq ki, public/uploads/ qovluğuna GitLab yazmaq yetkisinə sahibdir
sudo chmod -R u+rwX  public/uploads

sudo -u git -H cp config/unicorn.rb.example config/unicorn.rb  # Unicorn
                                                               nüsxə faylını
                                                               nüsxələyək

nproc       # CPU-da olan core-ların sayını tapırıq
4

# Əgər siz çox böyük yük bölgüsü edirsinizsə, cluster mode-u aktivləşdirin
# Əgər sizdə RAM 4GB-dirsə, onda worker_processes-in sayını sizdə olan CORE-
ların sayına bərabər edin
sudo -u git -H editor config/unicorn.rb

# Rack attack quraşdırma faylını nüsxələyək
sudo -u git -H cp config/initializers/rack_attack.rb.example
config/initializers/rack_attack.rb

# Git global konfigləri git istifadəçi üçün quraşdıraq, web üzərindən
dəyişiklik edəndə lazım olur,
# user.email-i gitlab.yml faylında təyin etdiyiniz kimi edin.
sudo -u git -H git config --global user.name "GitLab"
sudo -u git -H git config --global user.email "jamal.shahverdiyev@gmail.com"
```

```
sudo -u git -H git config --global core.autocrlf input

# Redis qoşulmasını quraşdıraq
sudo -u git -H cp config/resque.yml.example config/resque.yml

# Əgər siz Debian/Ubuntu-da susmaya görə olan socket istifadə etmirsinizsə,
ünvanı aşağıdakı faylda dəyişə bilərsiniz.
```
Vacib qeyd: Əmin olun ki, **gitlab.yml** və **unicorn.rb** configləri eyni edilib.

GitLab DB configlerini edek

```
# Yalnız PostgreSQL üçün quraşdırma faylı nüsxələyək
sudo -u git cp config/database.yml.postgresql config/database.yml

# PostgreSQL və MySQL üçün aşağıdakı quraşdırma faylında lazımi
dəyişiklikləri etmək lazımdır:
# config/database.yml faylında istifadəçi_adı/şifrə quraşdırmaq lazımdır.
# Biz yalnız 1-ci hissədə etdiyimiz baza, istifadəçi və şifrəni eynilə burada
da təyin etməliyik.
# Əgər siz şifrə dəyişmisinizsə onu password: sətirinin qarşısına
yazmalısınız və şifrə
# tək dirnaqların '' daxilində yazila bilər
sudo -u git -H editor config/database.yml

# PostgreSQL ve MySQL üçün:
# config/database.yml faylını git istifadəçi üçün oxunan edirik.
sudo -u git -H chmod o-rwx config/database.yml
```

GEMS-i yükləyirik

Qeyd: Bundler 1.5.2 üçün siz **bundle install -jN** əmrindən istifadə edə bilərsiniz(**N** - CPU-da olan core-ların sayıdır. Core-ların sayını isə **nproc** əmri ilə yoxlaya bilərsiniz). Bu işi **60%** daha sürətli edir. Ancaq əmin olun ki, sizin bundler **1.5.2**-dən yuxarı versiyadır. Siz bunu **bundle -v** əmri ilə yoxlaya bilərsiniz.

```
bundle -v                    # Mənim halımda aşağıdakı versiya idi
Bundler version 1.7.9

# PostgreSQL üçün (nəzərə alın ki, opsiya deyir ki, MySQL-siz yüklə)
sudo -u git -H bundle install --deployment --without development test mysql
aws
```

GitLab Shell-i yükləyək

```
GitLab Shell spesifik GitLab-ın özu üçün yazılmış program təminatıdır hansı
ki, SSH-a yetki və repository idarəetməsi üçün istifadə edilir.
# gitlab-shell yüklənməsi üçün aşağıdakı əmri daxil edirik (əgər `redis
ünvanı` dəyişmisinizsə
# burda da dəyişmək lazımdır). Əmri tam bir sətirdə yazmaq lazımdır
```

```
sudo -u git -H bundle exec rake gitlab:shell:install[v2.4.0]
REDIS_URL=unix:/var/run/redis/redis.sock RAILS_ENV=production

# Susmaya görə gitlab-shell konfigi sizin əsas Gitlab konfiginizdən
generasiya edilib.
# siz GitLab-shell konfiginizə aşağıdakı əmrlə baxa və ya dəyişə bilərsiniz.
sudo -u git -H editor /home/git/gitlab-shell/config.yml      # Əmrin nəticəsi
                                                             aşağıdakı kimidir
user: git
gitlab_url: https://git.domain.lan/
http_settings:
  self_signed_cert: true
repos_path: "/home/git/repositories/"
auth_file: "/home/git/.ssh/authorized_keys"
redis:
  bin: "/usr/bin/redis-cli"
  namespace: resque:gitlab
  socket: "/var/run/redis/redis.sock"
log_level: INFO
audit_usernames: false
```

Bazanı inisializasiya edək və geniş imkanları aktivləşdirək

```
sudo -u git -H bundle exec rake gitlab:setup RAILS_ENV=production

# Baza cədvəllərinin yaranması üçün yes daxil edin və ENTER düyməsini sıxın
# Sonda aşağıdakı sətirləri görəcəksiniz:
Administrator account created:

login.........root
password......5iveL!fe
```

Qeyd: Siz Administrator şifrəsini **GITLAB_ROOT_PASSWORD** mühit dəyişəni ilə dəyişə bilərsiniz. Həmçinin WEB üzərindən etmək mümkündür.

```
sudo -u git -H bundle exec rake gitlab:setup RAILS_ENV=production
GITLAB_ROOT_PASSWORD=newpassword
```

Init scripti yükləyək

```
sudo cp lib/support/init.d/gitlab /etc/init.d/gitlab         # Init skripti
                                                             startup skriptlər
                                                             yerləşən ünvana
                                                             nüsxələyək

# Əgər siz susmaya görə olan qovluqdan kənara yükləmisinizsə onda aşağıdakı
fərqli ünvandan
# lazımi ünvana nüsxələmək lazımdır. Bizim halda susmaya görədir
sudo cp lib/support/init.d/gitlab.default.example /etc/default/gitlab
```

Və əgər siz GitLab-ı susmaya görə olandan fərqli istifadəçi ilə və fərqli qovluğa yüklənmisinizsə onda, **/etc/default/gitlab** faylında bu dəyişiklikləri etmək lazımdır. Nəzərə alın ki, **/etc/init.d/gitlab** faylında dəyişiklik etmək olmaz çünki, yenilənmədə bu fayl özu yenilənir.

sudo update-rc.d gitlab defaults 21 # GitLab-ı startup-a əlavə edirik

LogRotasiyasını işə salırıq
sudo cp lib/support/logrotate/gitlab /etc/logrotate.d/gitlab # Lograte nüsxələyirik

Programın statusunu yoxlayırıq
Yoxlayaq görək GitLab və onun mühiti düzgün işləyirmi:
sudo -u git -H bundle exec rake gitlab:env:info RAILS_ENV=production

Aktivləri kompilyasiya edək
sudo -u git -H bundle exec rake assets:precompile RAILS_ENV=production

GitLab servisini işə salaq
sudo service gitlab start
Yada
sudo /etc/init.d/gitlab restart

7. nGinx yüklənməsi və quraşdırılması

Rəsmi olaraq nGinx web server GitLab tərəfindən dəstəklənir. Əgər siz nGinx web server yox başqasını istifadə etmək istəsəniz onda GitLab portalından məsləhətlər alın.

Yuklenme
sudo apt-get install -y nginx

Site quraşdırılması
#Nüsxə sayt konfigini düzgün ünvana nüsxə və link edək(**HTTP** üçün **gitlab HTTPS** üçün isə **gitlab-ssl**)
Http üçün
sudo cp lib/support/nginx/gitlab /etc/nginx/sites-available/gitlab
sudo ln -s /etc/nginx/sites-available/gitlab /etc/nginx/sites-enabled/gitlab

HTTPS üçün isə aşağıdakı kimi edirik:
sudo cp lib/support/nginx/gitlab-ssl /etc/nginx/sites-available/gitlab-ssl
sudo ln -s /etc/nginx/sites-available/gitlab-ssl /etc/nginx/sites-enabled/gitlab-ssl

```
# öz quruluşumuza əsasən öz konfig faylımızda dəyişiklik edək:
sudo editor /etc/nginx/sites-available/gitlab   # HTTP üçün bu fayl
sudo editor /etc/nginx/sites-available/gitlab-ssl      # Mənim halımda HTTPS
                                                      olduğu üçün bu fayl

Əsas quraşdırma sətirləri aşağıdakılardır hansı ki, düzgün quraşdırılmalıdır
ki, DNS-də bu host üçün əlavə etdiyiniz A yazısı düzgün resolve edə
biləsiniz.
upstream gitlab {
  server unix:/home/git/gitlab/tmp/sockets/gitlab.socket fail_timeout=0;
}
server {
  listen 10.50.3.206:80;
  server_name git.domain.lan;
  server_tokens off;
  return 301 https://$server_name$request_uri;
  access_log  /var/log/nginx/gitlab_access.log;
  error_log   /var/log/nginx/gitlab_error.log;
}
server {
  listen 10.50.3.206:443 ssl;
  server_name git.domain.lan;
  server_tokens off;  root /home/git/gitlab/public;
  client_max_body_size 20m;
  ssl on;
# Sertifikatlari aşağıdakı yaradacayiq
  ssl_certificate /etc/nginx/ssl/gitlab.crt;
  ssl_certificate_key /etc/nginx/ssl/gitlab.key;
  ssl_ciphers "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:DHE-
RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-
SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA:ECDHE-RSA-AES128-SHA:DHE-
RSA-AES256-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA:DHE-RSA-AES128-
SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES256-GCM-SHA384:AES128-GCM-
SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-SHA:DES-CBC3-
SHA:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4";
  ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
  ssl_prefer_server_ciphers on;
  ssl_session_cache shared:SSL:10m;
  ssl_session_timeout 5m;
  access_log  /var/log/nginx/gitlab_access.log;
  error_log   /var/log/nginx/gitlab_error.log;
  location / {
      try_files $uri $uri/index.html $uri.html @gitlab;
  }
  location @gitlab {
    gzip off;
    proxy_read_timeout      300;
    proxy_connect_timeout   300;
    proxy_redirect          off;
    proxy_set_header    Host                $http_host;
    proxy_set_header    X-Real-IP           $remote_addr;
    proxy_set_header    X-Forwarded-Ssl     on;
    proxy_set_header    X-Forwarded-For     $proxy_add_x_forwarded_for;
```

```
    proxy_set_header    X-Forwarded-Proto   $scheme;
    proxy_set_header    X-Frame-Options     SAMEORIGIN;
    proxy_pass http://gitlab;
  }
  location ~ ^/(assets)/ {
    root /home/git/gitlab/public;
    gzip_static on;
    expires max;
    add_header Cache-Control public;
  }
  error_page 502 /502.html;
}

# Sertifikatları düzgün ünvanda yaratdıqdan sonra, aşağıdakı əmr ilə nGinx-in
statusunu
# yoxlayırıq. Görünən cavab qayıtmalıdır
sudo nginx  -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful


# nGinx-in servisini restart edirik
sudo service nginx restart
```

HTTPS üçün sertifikatlarimizi yaradaq.

Bunun üçün aşağıdakı addımları dəqiq etmək lazımdır.

1. **gitlab.yml** faylında
   a. **port**-u **443** etmək lazımdır
   b. 1-ci seksiyada **https**-i **true** etmək lazımdır
2. **config.yml** faylında
   a. **gitlab_url** opsiyasını **https** üçün təyin etmək lazımdır(**https://git.domain.lan**)
   b. Sertifikatların istifadəsinə **ca_file** və **ca_path** təyin etmək olar
3. nGinx-in quraşdırma faylında **gitlab** faylı əvəzinə **gitlab-ssl** istifadə etmək lazımdır
   a. Serverin **FQDN**-ni düzgün yazın
   b. **ssl_certificate** və **ssl_certificate_key** ünvanlarını dəqiq yazın
   c. Config-ə dəqiq baxın və diger təhlukəsizlik quraşdırmalarını edin

Özümüz tərəfimizdən generasiya edilən və imzalanan sertifikat üçün isə aşağıdakı addımları edirik:

1. Self-Signed SSL sertifikatını generasiya edək:

```
sudo mkdir -p /etc/nginx/ssl/

cd /etc/nginx/ssl/

sudo openssl req -newkey rsa:2048 -x509 -nodes -days 3560 -out gitlab.crt -
keyout gitlab.key
```

```
Country Name (2 letter code) [AU]:AZ

State or Province Name (full name) [Some-State]:Baku

Locality Name (eg, city) []:YeniYasamal

Organization Name (eg, company) [Internet Widgits Pty Ltd]:DOMAIN

Organizational Unit Name (eg, section) []:IT

Common Name (e.g. server FQDN or YOUR name) []:git.domain.lan

Email Address []:jamal.shahverdiyev@domain.az
```

```
sudo chmod o-r gitlab.key
```

2. gitlab-shell-ində istifadə etdiyimiz **config.yml** faylında **self_signed_cert** opsiyasını **true** edin.

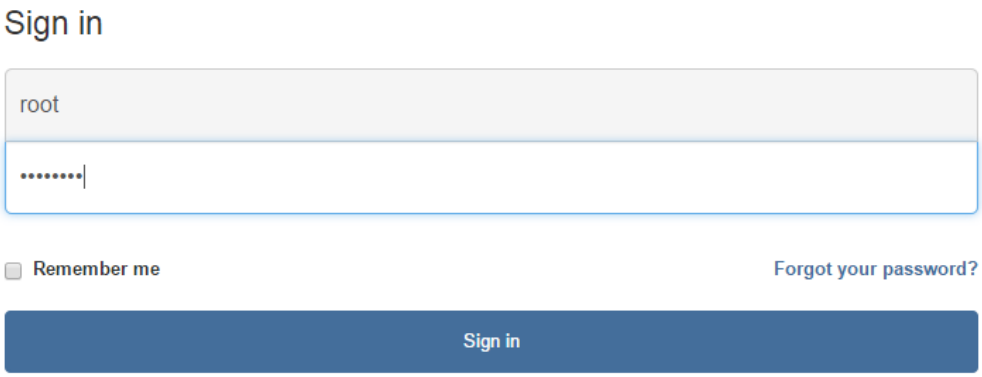
Program statusunu yenidən yoxlayaq
Bütün quraşdırmalarımızın qaydada olmasını yoxlamaq üçün aşağıdakı əmri yenidən daxil edirik:

```
cd /home/git/gitlab
sudo -u git -H bundle exec rake gitlab:check RAILS_ENV=production
```

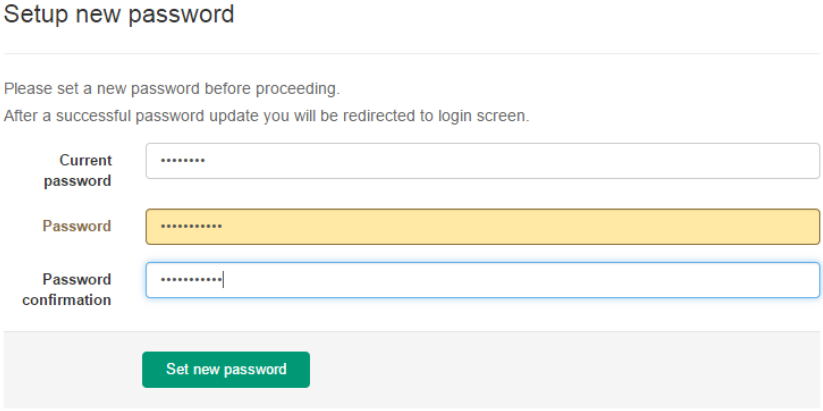
```
Nəticə səhvsiz, yaşıl və aşağıdakı kimi olmalıdır:
Redis version >= 2.0.0? ... yes
Ruby version >= 2.0.0 ? ... yes (2.1.5)
Your git bin path is "/usr/bin/git"
Git version >= 1.7.10 ? ... yes (1.9.1)

Checking GitLab ... Finished
```

Qeyd: **SANITIZE=true** mühit dəyişənin təyinatı ilə siz **gitlab:check** əmrinin çıxışında proyektlər haqqında çıxışın nəticəsinin çap edilməsinin qarşısını almış olacaqsınız.

https://git.domain.lan ünvanına aşağıdakı istifadəçi adı, şifrə ilə daxil olun və şifrəni dəyişin.
login: **root**
pass: **r00tpass**



Sonra **Sign in** düyməsini sıxırıq və aşağıdakı şəkildəki kimi şifrəni dəyişirik.



Siz servisləri aşağıdakı əmrlər ilə **restart** və ya **stop, start** edə bilərsiniz.

```
sudo service gitlab restart
[sudo] password for jamal:
Shutting down both Unicorn and Sidekiq.
GitLab is not running.
Starting both the GitLab Unicorn and Sidekiq.
The GitLab Unicorn web server with pid 28862 is running.
The GitLab Sidekiq job dispatcher with pid 28904 is running.
GitLab and all its components are up and running.
```

Redis qoşulmasını istəyimizə görə dəyişə bilərik:

Əgər siz Redis-ə fərqli host və port ilə qoşulmaq istəsəniz onda **config/resque.yml** quraşdırma faylında dəyişiklik etməlisiniz.

```
# nüsxə
production: redis://redis.example.tld:6379
```

Əgər siz redis-ə "**unix:**" socket ilə qoşulmaq istəsəniz onda **config/resque.yml** faylında aşağıdakı quraşdırmanı etməlisiniz.

```
# nüsxə
production: unix:/path/to/redis/socket
```

Fərqli SSH qoshulması

Əgər siz SSH-ın qulaq asdığı portu dəyişmisinizsə, onda siz GitLab istifadəçisinin SSH konfigini dəyişməlisiniz.

/home/git/.ssh/config faylına aşağıdakı sətirləri əlavə etməlisiniz

```
host localhost          # hostname
    user git            # remote git istifadəçi adı
    port 2222           # SSH port rəqəmi
    hostname 127.0.0.1; # Server adı yada IP
```

Həmçinin siz düzgün configləri **ssh_user**, **ssh_host**, **admin_uri** opsiyaları üçün **config/gitlab.yml** faylında dəyişməlisiniz.

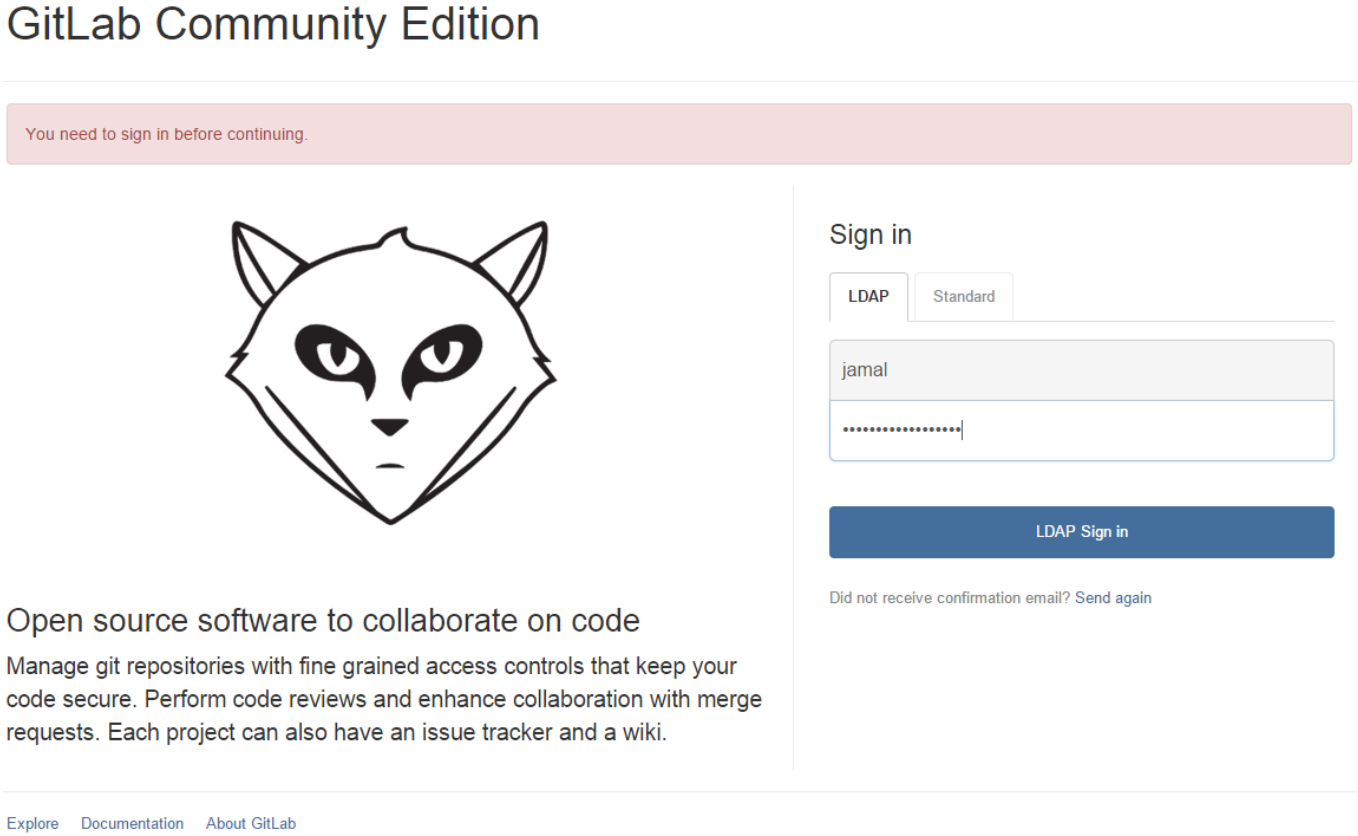
MSLDAP authentifikasiya

Əgər biz GitLAB-ı öz müəssisəmizə aid olan domain controller ilə inteqrasiya etmək istəsək, onda **config/gitlab.yml** faylında düzgün dəyişiklikləri etməliyik ki, DC-yə qoşulub istifadəçiləri yoxlanış edə bilək.

```
cd /home/git/gitlab                          # Konfig qovluğuna daxil oluruq
sudo -u git editor config/gitlab.yml         # Konfig faylımızın LDAP başlığında
                                             lazımi dəyişiklikləri aşağıdakı
                                             kimi edirik.

  ldap:
    enabled: true
    servers:
      main:
        label: 'LDAP'
        host: 'domain.lan'
        port: 636
        uid: 'sAMAccountName'
        method: 'ssl' # "tls" or "ssl" or "plain"
        bind_dn: 'CN=DCADM,CN=Users,DC=domain,DC=lan'
        password: 'DC_PASSWORD'
        active_directory: true
        allow_username_or_email_login: false
        base: 'DC=domain,DC=lan'
        user_filter: '(memberOf=CN=GITUsers,OU=DOMAINTech
Groups,OU=DOMAINTech,DC=domain,DC=lan)'


sudo /etc/init.d/gitlab restart      # Gitlab servisi yenidən işə salırıq


# LDAP konfigimizi yoxlayırıq və istifadəçiləri görməliyik artıq
sudo -u git -H bundle exec rake gitlab:ldap:check RAILS_ENV=production
Checking LDAP ...

LDAP users with access to your GitLab server (only showing the first 100
results)
```

```
Server: ldapmain
DN: CN=Eldaniz Ibrahimov,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
sAMAccountName: eldaniz
DN: CN=Jamal Shahverdiyev,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
sAMAccountName: jamal
DN: CN=Sukur Rzayev,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
        sAMAccountName: SukurR
DN: CN=Musaqil Musabeyli,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
sAMAccountName: MusaqilM
DN: CN=Hidayat Soltanzade,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
sAMAccountName: HidayatS
DN: CN=Alakbar Velizade,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
sAMAccountName: AlakbarV
DN: CN=Rufat Babakishiyev,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
sAMAccountName: RufatBa
DN: CN=Javid Ismayilzade,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
sAMAccountName: JavidI
DN: CN=Yunis Babayev,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
sAMAccountName: YunisB
DN: CN=Rovshan Baghirov,OU=DOMAINTech Users,OU=DOMAINTech,DC=domain,DC=lan
sAMAccountName: RovshanB

Checking LDAP ... Finished
```

Sonra yenidən https://git.domain.lan ünvanına daxil oluruq və DC istifadəçisi ilə şəkildə göstərilən kimi daxil oluruq.



Artıq proqramçılar desktoplarından istənilən git client vasitəsilə öz mənbə codelarını bizim qurduğumuz serverə sinxronizasiya edə bilərlər.

Sonda bir daha qeyd edim ki, 1-ci başlıqda qeyd edilmiş, **/home/git/gitlab/config/gitlab.yml** faylında **bin_path** opsiyası üçün **git**(**/usr/local/bin/git**) binar faylının düzgün ünvanını təyin etməyi unutmayın.

BÖLÜM 13

İnternet üzərindən canlı iclaslar

- **OpenMeetings qurulması və istifadəsi**
- **BigBlueButton qurulması və istifadə edilməsi**

Böyük müəssisələrin tələbləri yarana bilər ki, şirkətlərinin və ya filiallarının arasında danışıq onlayn şəkildə olsun. Bunun üçün onlayn iclaslar keçirmək imkanına sahib olan spesifik avadanlıqlar və bahalı proqram təminatları mövcuddur. Yalnız bu başlığımızda açıq qaynaqlı proqramların vasitəsilə bütün pullu distributivlərin bacardıqları eyni funksionallığı və hətta artığının qurulmasından danışacıyıq.

OpenMeetings qurulması və istifadəsi

Məqsədimiz WEB üzərindən onlayn şəkildə şəxslərin bir-biri ilə kamera və səs ilə iclas keçirməsi, yaza bilməsi, ekranın yayımlanması, ekranın video/audio yazılması və DOC/PDF sənədin birgə baxılması imkanlarına malik olan bir sistemin qurulmasıdır.

Öncədən qeyd edim ki, testlərinizdə surprizlərlə qarşılaşmayasınız. Windows7/8/8.1, Ubuntu Desktop 14.04 və MacOS-da problemsiz hər şey işlədi. Ancaq windows XP-de işləmir. Bundan bashqa flash-da işlədiyi üçün Windows-da IE browserdə tamamilə problem olmadı. Amma hər hal üçün bütün testlərinizi fərqli browserlərdə etsəniz düzgün nəticə əldə etmiş olacaqsınız.

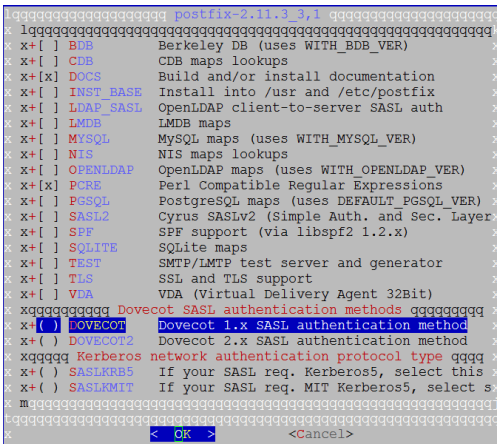
OpenMeetings - Bu proqram təminatı prezentasiyaların edilməsi, onlayn təhsil, web konfrans, ümumi şəkil lövhəsi və sənədlərin redaktə edilməsi funksionallığına sahibdir. Bu başlığımızda quracağıq.

İşə başlayaq.

```
portsnap fetch extract update        # Öncə portları yeniləyək
```

OpenMeetings istifadə edəcəyi üçün sendmail-i söndürürük və postfix-i yükləyib işə salırıq:

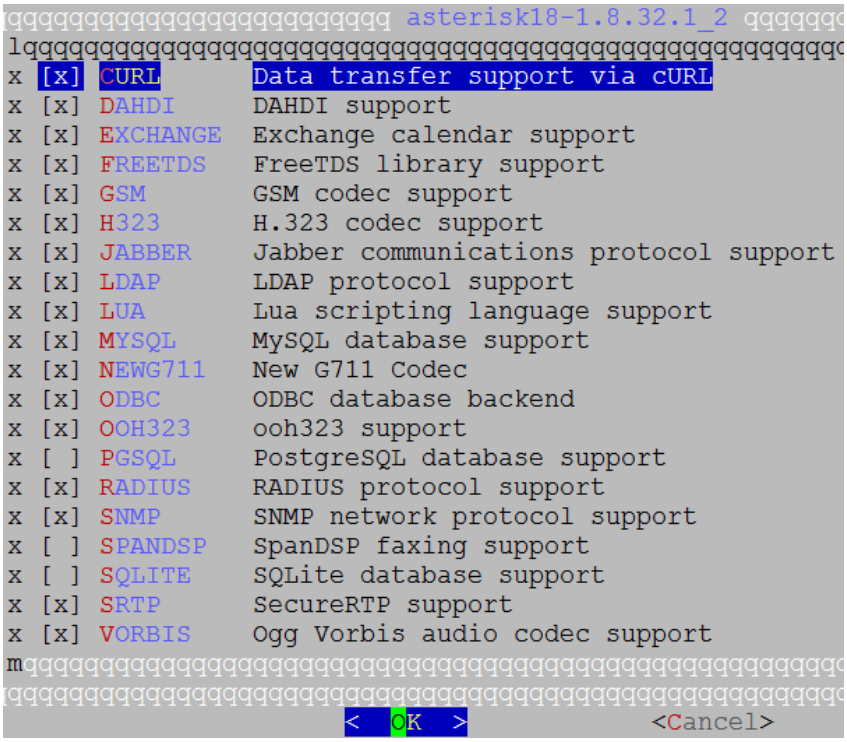
```
cd /usr/ports/mail/postfix           # Port ünvanına daxil oluruq
make config                          # Lazımı modulları seçirik
```




```
make install                                 # Yükləyirik
```

SIP dəstəklənməsi üçün Asterisk-i öncədən yükləyirik.

```
cd /usr/ports/net/asterisk                   # Port ünvanına daxil oluruq
make config                                  # Lazımı modulları seçirik
```



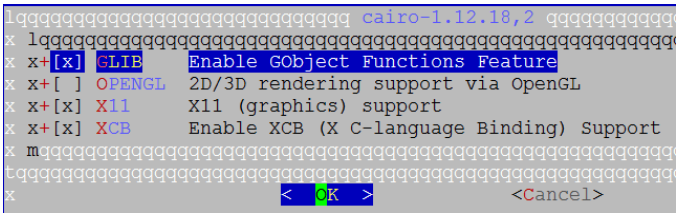

```
make install                                    # Yükləyirik
```

OpenMeetings BASH ilə işlədiyinə görə bash-ı serverimizə yükləyirik:

```
cd /usr/ports/shells/bash                       # Port unvanına daxil oluruq
make config                                     # Lazımı modulları susmaya görə seçirik
make install                                    # yükləyirik(/usr/local/bin/bash binar
                                                faylı yaranacaq)
```

Ofis programların və ImageMagick-ın işləməsi üçün cairo tələb edilir. Ona görə də onu X11 dəstəklənməsi ilə yükləyirik.

```
cd /usr/ports/graphics/cairo                    # Port ünvanına daxil oluruq
make config                                     # Şəkildə görünən moduları seçirik
```




```
make install                                    # Yükləyirik
```

Şəkillərlə işləyə bilmək üçün ImageMagick yükləmək lazımdır.

```
cd /usr/ports/graphics/ImageMagick              # Port ünvanına daxil oluruq
make config                                     # Lazımı modulları seçirik
```

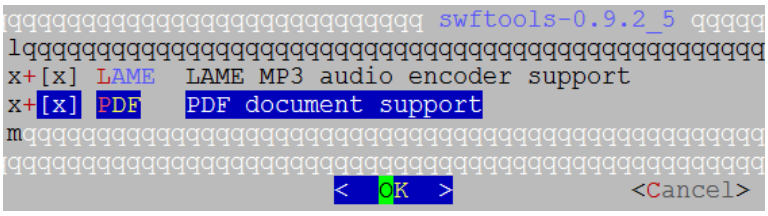

```
make -DBATCH install                                    # Yükləyirik
```

OpenMeetings-i startup-a əlavə etmək üçün expect lazım olacaq. Bunun üçün onu yükləyirik.

```
cd /usr/ports/lang/expect                               # Port ünvanına daxil oluruq
make install                                            # Yükləyirik
```

PFD sənədlərin import edilə bilməsi üçün swftools-u yükləyirik.

```
cd /usr/ports/graphics/swftools                         # Port unvanına daxil oluruq
make config                                             # lazımı modulları seçirik
```

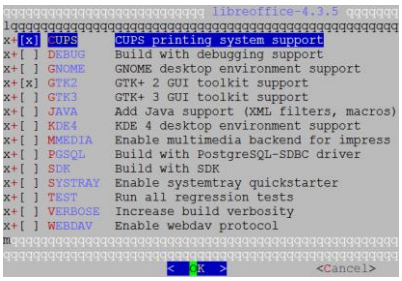



```
make install                                            # Yükləyirik
```

.doc, **.docx**, **.odp**, **.xls**, **.xlsx**, **.ppt**, **.pptx** tipli sənədlərin import edilə bilməsi üçün **libreoffice**-i yükləyirik.

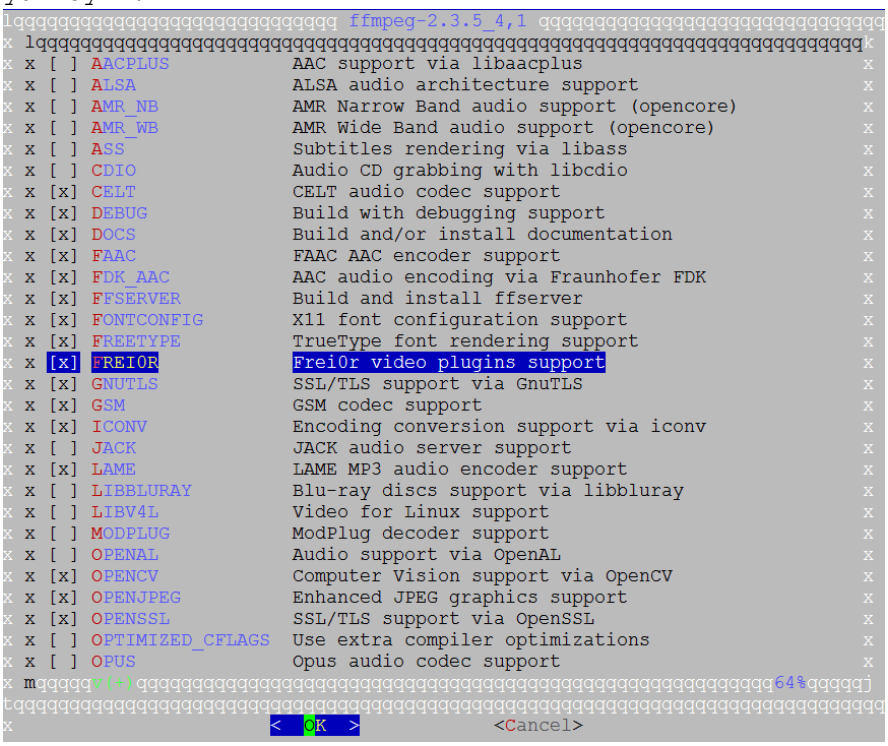
```
cd /usr/ports/editors/libreoffice                       # Port ünvanına daxil oluruq
make config                                             # Lazımı modulları
                                                        seçirik(Hər şəy susmaya görə
                                                        olmalıdır, əks halda
```

yüklənməyəcək)



make -DBATCH install # Yükləyirik(yüklənmə həddən artıq çox vaxt alacaq)

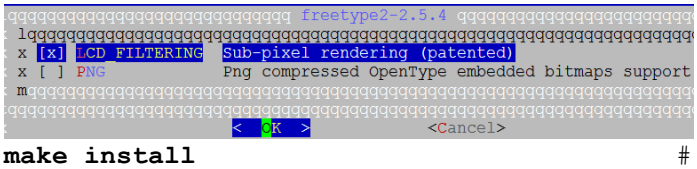
Yüklənmə müddətində **ffmpeg** menyusu açılacaq ki, seçim edək. Orda mütləq **LAME** və **FDK_AAC** seçirik. Əgər yüklənmə müddəti çıxmasa, mütləq özünüz **/usr/ports/multimedia/ffmpeg** port ünvanına daxil olub əlinizlə seçib yükləyin.



Əgər yüklənmə müddətində freetype2 menyusu açılmasa ki, seçim edib yükləyək onu aşağıdakı qaydada portuna daxil olub yükləmək lazımdır:

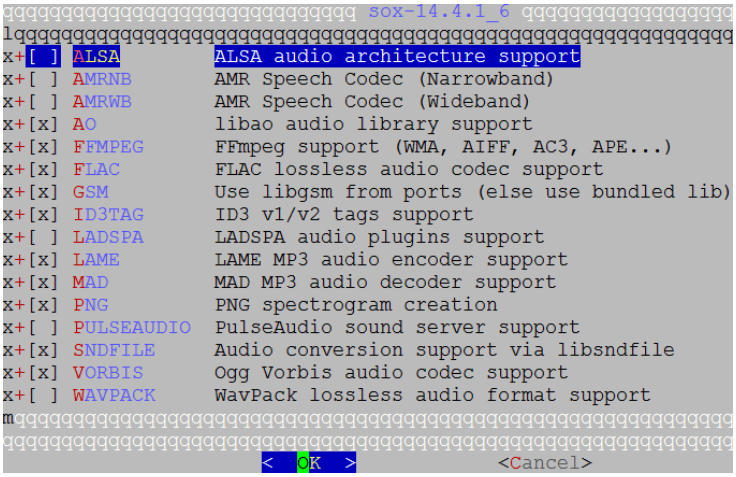
cd /usr/ports/print/freetype2 # Port ünvanına daxil oluruq

make config # Lazımı modulları seçirik



make install # Yükləyirik

```
cd /usr/ports/audio/sox     # Audio convert və qulaq asmaq üçün sox yükləyirik
make config                 # lazımı modulları seçirik(Mutləq Lame olmalıdır)
```



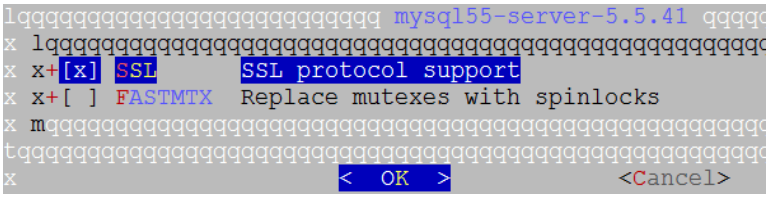

```
make install                # Yükləyirik
```

/etc/rc.conf startup quraşdırma faylımız aşağıdakı kimi olacaq:

```
hostname="om.domain.az"
ifconfig_em0="inet 98.97.96.140 netmask 255.255.255."
defaultrouter="98.97.96.1"
sshd_enable="YES"
dumpdev="NO"

#### Disabled Local Services ####
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
sendmail_rebuild_aliases="NO"
syslogd_enable="NO"
syslogd_program="/usr/sbin/syslogd"
syslogd_flags="-ss"
mysql_enable="YES"

#### Third party Services ####
postfix_enable="YES"

/usr/local/etc/rc.d/postfix start          # Mail serverimizi işə salırıq
```

OpenMeetings üçün MySQL bazası tələb edilir ona görə də onu yükləyək və konfiq edək:

```
cd `whereis mysql55-server | awk '{ print $2 }'` # MySQL port unvanına daxil
                                                 oluruq
make config                                      # lazımı modulları seçirik
```



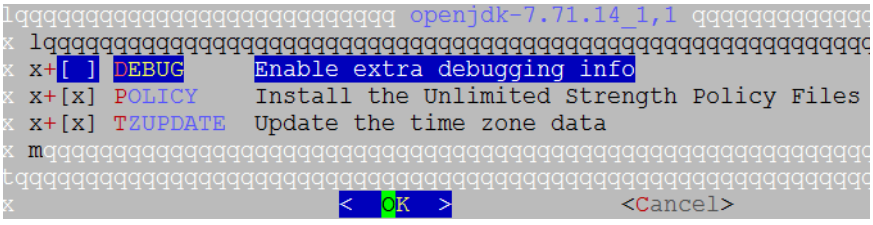

```
make install                                          # Yükləyirik

echo 'mysql_enable="YES"' >> /etc/rc.conf             # Starup-a əlavə edirik
/usr/local/etc/rc.d/mysql-server start                # Işə salırıq

/usr/local/bin/mysql_secure_installation              # Bu əmr ilə MySQL-i
                                                      tehlükəsiz
                                                      quraşdırırıq. Aşağıdakı
                                                      addimları eynilə
                                                      edirik.

Enter current password for root (enter for none):
OK, successfully used password, moving on...
Set root password? [Y/n] Y
New password: Yeni_şifrə
Re-enter new password: Təkrar_Yeni_şifrə
Password updated successfully!

Remove anonymous users? [Y/n] Y
 ... Success!

Disallow root login remotely? [Y/n] Y
 ... Success!

Remove test database and access to it? [Y/n] Y
 - Dropping test database...
... Success!
 - Removing privileges on test database...
 ... Success!

Reload privilege tables now? [Y/n] Y
 ... Success!

cp /usr/local/share/mysql/my-large.cnf /etc/my.cnf    # Quraşdırma faylını
                                                      nüsxələyirik

/etc/my.cnf faylının [mysqld] bashlığında aşağıdakı dəyişiklikləri
edirik(Kodlaşdırma UTF8 edirik):
collation-server = utf8_unicode_ci
init-connect='SET NAMES utf8'
character-set-server = utf8
bind-address = 127.0.0.1
log = /var/log/mysql.log

Həmçinin /etc/my.cnf faylında [client] bashlığında aşağıdakı sətri mütləq
əlavə etmək lazımdır:
default-character-set=utf8
```

```
touch /var/log/mysql.log                          # Jurnal faylını yaradırıq

chown mysql:mysql /var/log/mysql.log              # mysql istifadəçi və qrup
                                                  üçün həmin fayla yetki
                                                  veririk

mysql -uroot -p'freebsd'                          # MySQL-in console-una daxil
                                                  oluruq və aşağıdakı əmrlərlə
                                                  openmeetings baza, istifadəçi
                                                  adı, şifrə və hansı host ilə
                                                  qoshulmanı təyin edirik

CREATE DATABASE IF NOT EXISTS `openmeetings`;     # Əgər yoxdursa, bazanı
                                                  yaradırıq

GRANT ALL PRIVILEGES ON `openmeetings`.* TO 'openmeetings'@'localhost'

IDENTIFIED BY 'freebsd';

FLUSH PRIVILEGES;                                 # Yetkiləri yenidən oxuyuruq
exit;                                             # çıxırıq
```

Javani yükləyirik

Ümumiyyətlə java müxtəlif versiyalarda fərqli işləyir və elə ola bilər ki, siz openmeetings-in işləməsi üçün fərqli versiya Java yükləyə bilərsiniz. Mənim halımda openjdk6-jre, openjdk8-jre yüklədim işləmədi və sonda hamısını silib, openjdk7-jrə yüklədim işlədi. Ona görə də openjdk7-jre-ni yükləyirik.

```
cd /usr/ports/java/openjdk7/                      # Port ünvanında daxil oluruq
make config                                       # Lazımı modulları seçirik
```




```
make install                                      # Yükləyirik
```

Java yükləndikdən sonra öz işləməsi üçün **procfs** və **fdescfs** fayl sistəmlərinin mount edilməsini və startup-a əlavə edilməsini tələb edir. Bunun üçün aşağıdakı sətirləri **/etc/fstab** faylına əlavə edirik.

```
echo "fdesc   /dev/fd       fdescfs         rw      0       0" >> /etc/fstab
echo "proc    /proc         procfs          rw      0       0" >> /etc/fstab

mount -a          # Lazımı fayl sistemlərinin /etc/fstab faylından mount
                  edirik və startup-u yoxlayırıq
```

```
java -version                # Java versiyasını yoxlayırıq. Çıxış aşağıdakı
                                                    kimi olacaq
openjdk version "1.7.0_71"
OpenJDK Runtime Environment (build 1.7.0_71-b14)
OpenJDK 64-Bit Server VM (build 24.71-b01, mixed mode)

OpenMeetings-i dartaq və yükləyek
mkdir /usr/local/om          # Yükləmək üçün qovluq yaradırıq
cd /usr/local/om             # həmin qovluğa daxil oluruq

# Lazımı versiyanı yerləşdiyimiz qovluğa dartırıq(Ümumiyyətlə ən son
versiyanı
# https://builds.apache.org/view/M-R/view/OpenMeetings/ linkindən əldə edə
bilərsiniz)
fetch http://apache-mirror.rbc.ru/pub/apache/openmeetings/3.0.3/bin/apache-
openmeetings-3.0.3.tar.gz

tar zxf apache-openmeetings-3.0.3.tar.gz  # Paketi yerləşdiyimiz ünvana
                                          açırıq
```

Qeyd: Paketi **/usr/local/om** ünvanına açdıqdan sonra **.sh** genişlənməli bütün Scriptlərin içində, bash-ın ünvanını **/usr/local/bin/bash** təyin etmək lazımdır. Həmçinin **/usr/local/om/red5.sh** scriptinin içində **OS** dəyişəni üçün **FreeBSD** şərti yazmaq lazımdır. Ona görə ki, bu deyişənin sayəsində, OpenMeetings üçün **JAVA_HOME** mühiti tanınır. Eynilə **root** istifadəçisinin ev qovluğunda **.bashrc** faylının içinə də

```
JAVA_HOME=/usr/local/openjdk7/jre
export JAVA_HOME
```

sətirlərini əlavə etmək lazımdır. Aşağıdakı sətirləri uyğun olaraq, **/usr/local/bin/red5.sh** faylında dəyişmək lazımdır(**Darwin case**-i silinir və yerinə **FreeBSD** yazılır. Aşağıdakı kimi☺)

```
OS=`uname`
case "$OS" in
  CYGWIN*|MINGW*) # Windows Cygwin or Windows MinGW
  P=";" # Since these are actually Windows, let Java know
  ;;
  FreeBSD*)
        if [ -z "$JAVA_HOME" ]; then
                export JAVA_HOME=/usr/local/openjdk7/jre;
        fi
  ;;
```

Javanın MySQL-ə qoşulması üçün connectoru download edirik və serverdə **/usr/local/om/webapps/openmeetings/WEB-INF/lib/** ünvanına yerləşdiririk. http://dev.mysql.com/downloads/file.php?id=454396 linkində MySQL connectoru endirmək üçün qeydiyyatdan keçirik və MySQL connector-u http://dev.mysql.com/downloads/connector/j/ linkindən endiririk.

```
# Connectoru nüsxələyirik kitabxanalar olan ünvana
cp /home/jamal/mysql-connector-java-5.1.34.tar.gz
/usr/local/om/webapps/openmeetings/WEB-INF/lib/
```

```
cd /usr/local/om/webapps/openmeetings/WEB-INF/lib/     # Connector olan ünvana
                                                       daxil oluruq
tar zxf mysql-connector-java-5.1.34.tar.gz             # sıxılan faylı
                                                       yerləşdiyimiz ünvana
                                                       açırıq

Ancaq jar faylı lib-ə atırıq və qovluğu silirik
mv mysql-connector-java-5.1.34/mysql-connector-java-5.1.34-bin.jar .
rm mysql-connector-java-5.1.34.tar.gz                  # Sıxılmış faylın özünü
                                                       də silirik

cd /usr/local/om/webapps/openmeetings/WEB-INF/classes/META-INF/    # Sonra bu
                                                       ünvana daxil oluruq

cp persistence.xml old_persistence.xml     # Sonra persistence.xml faylını
                                           köhnə adla nüsxələyirik

rm persistence.xml      # Sonra original persistence.xml faylını silirik

cp mysql_persistence.xml persistence.xml               # Sonra MySQL ilə olan
                                                       konfiq faylını original
                                                       fayla nüsxələyirik

Sonra persistence.xml faylında Url=jdbc:mysql://localhost:3306/ sətrini
tapırıq və Username=, Password= sətirlərində bazada yaratdığımız istifadəçi
ilə şifrə təyin edirik. Aşağıdakı kimi:
    , Username=openmeetings
    , Password=freebsd" />

cd /usr/local/om        # Yükləməyə başlamaq üçün bu ünvana daxil oluruq

# Aşağıdakı əmr ilə yükləməyə başlayırıq(Ardınca əmri açıqlayırıq).
Ümumiyyətlə yüklənmə
# proseduruna http://openmeetings.apache.org/installation.html rəsmi
linkindən baxa bilərsiniz
# ancaq burdakı qədər detallı və açıq yazılmayıb.
sh ./admin.sh -i -v -tz Asia/Baku -email jamal.shahverdiyev@domain.az -group
Users -user admin --smtp-server localhost --db-type mysql --db-user
openmeetings --db-pass freebsd --db-name openmeetings --db-host localhost --
skip-default-rooms --password rumburak

-tz - Time Zone dəməkdir(Bizim halda Asia/Baku)
-email - inizibatçının email ünvanıdır(Mənim halımda öz emailim)
-group - Istifadeçilər yerləşdiyi susmaya görə olan qrup(Mənim halımda Users)
-user  - inzibatçı logini(Bizim halda elə admin)
--smtp-server - localhost(Ancaq əvvəldə yazdığım kimi, postfix-i yükləməyi
unutmayın)
--db-type - Bazanın tipini seçirik(Bizim halda MySQL)
--db-user - Baza istifadəçi adı(bizim halda openmmetings)
--db-pass - Baza istifadəçisinin şifrəsi(Bizim halda freebsd)
--db-name - Bazanın adı(openmeetings)
--db-host - Bazaya qoshulan host(Bizim halda localhost özüdür)
--password - inzibatçı şifrəsi(Bizim halda rumburak)
```

Sonda hər şey uğurlu olarsa aşağıdakı sətirlər çap edilməlidir:

```
[INFO] [main] org.apache.openmeetings.db.dao.user.UserDao - [get] Info: No
USER_ID given
[INFO] [main] org.apache.openmeetings.db.dao.user.UserDao - [get] Info: No
USER_ID given
[INFO] [main] org.apache.openmeetings.db.dao.user.UserDao - [get] Info: No
USER_ID given
[INFO] [main] org.apache.openmeetings.db.dao.user.UserDao - [get] Info: No
USER_ID given
... Done
```

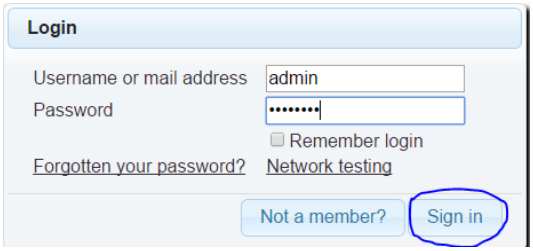
Qeyd: Yüklənmə müddətincə çıxan səhvlərdən narahat olmayın çünki, siz **openmeetings** bazasını asanlıqla silib yenidən yarada bilərsiniz və yüklənməni yənidən edə bilərsiniz. Aşağıdakı qaydada:

```
drop database openmeetings;
CREATE DATABASE openmeetings DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci;
```

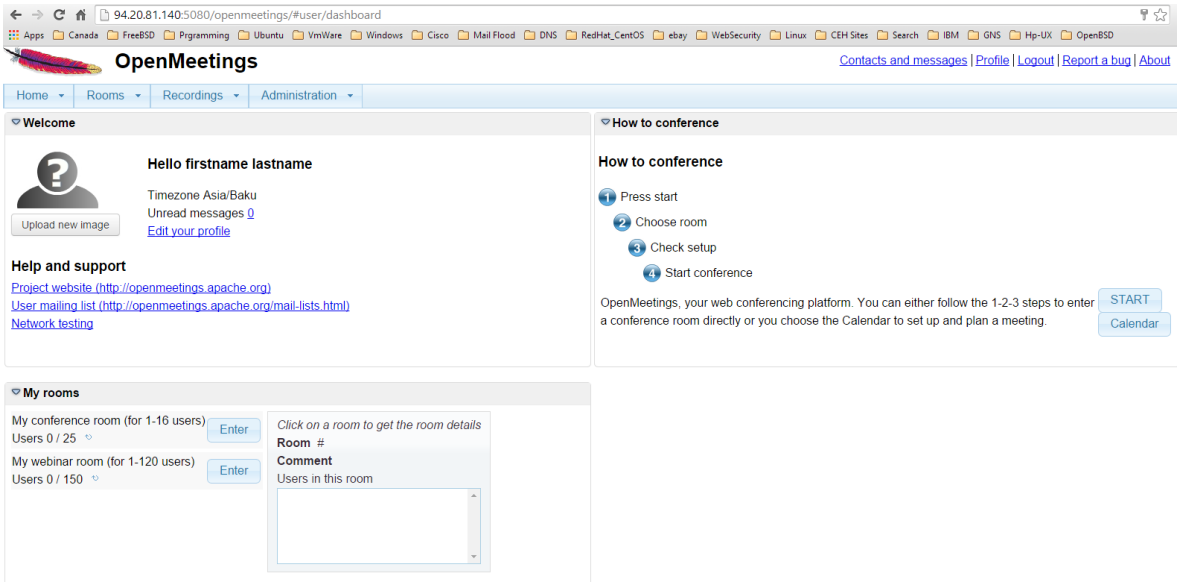
Işə salmaq üçün isə eynilə **/usr/local/om** ünvanına daxil olub **red5.sh** scriptini işə salmaq lazımdır:

```
cd /usr/local/om/       # OpenMeetings yerleşən ünvana daxil oluruq
sh ./red5.sh            # Sevisi işə salırıq(Nəticə aşağıdakı kimi olacaq)
        ###################################################
        #               Openmeetings is up                #
        #     3.0.3-RELEASE 1621852 2-September-2014      #
        #                and ready to use                 #
        ###################################################
```

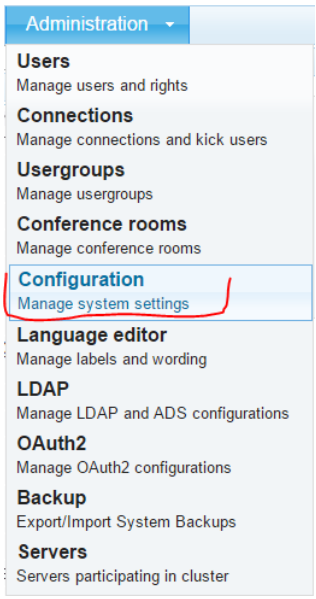
Yuxarıda görünən nəticəni aldqıdan sonra, SSH ilə əmri işə saldiğımız sessiyanı bağlamırıq çünki, bağladıqda servis-də sönəcək. Bunun üçün birazdan startup script yazacağıq və onu cron-da təyin edəcəyik ki, reboot-dan sonra avtomatik işə düşsün. Sessiyamız açıq vəziyyətdə qalaraq serverimizin http://98.97.96.140:5080/openmeetings/install linkinə müraciət edirik. Aşağıdakı şəkildəki kimi istifadəçi adı və şifrəni daxil edirik. Daxil edilən istifadəçi adı və şifrə **admin.sh** scriptində yazdiğımızdır.



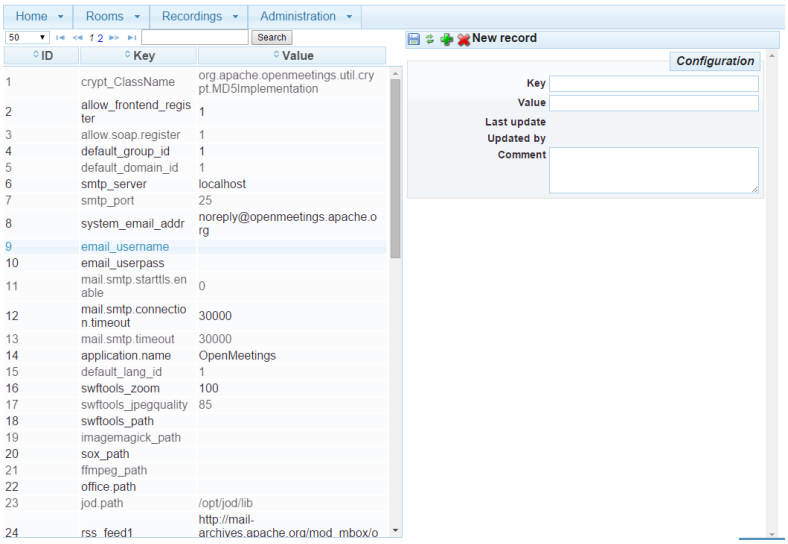
Aşağıdakı şəkilə uyğun olan bir nəticə əldə etməlisiniz:



Yüklənmədə istifadə etdiyimiz bütün quraşdırmaları **Administration -> Configuration** bolümündə görə bilərsiniz:



Aşağıdakı şəkildəki kimi configləri görə bilərsiniz:



Indi isə OpenMeetings-in avtomatik işə düşməsi üçün quraşdırmalarımızı edək. Bunun üçün **/usr/local/etc/rc.d** ünvanında **red5.sh** adlı script yaradaq. Bu scriptin sayəsində bizim OpenMeetings servisi restartdan sonra avtomatik olaraq işə düşəcək. Bunu aşağıdakı qaydada edirik.
/usr/local/etc/rc.d/red5.sh faylının içinə aşağıdakı sətirləri elavə edirik:

```
#!/bin/sh

RED5_DIR=/usr/local/om
test -x $RED5_DIR/red5.sh || exit 5

case "$1" in
	start)
		cd "$RED5_DIR"
		"$RED5_DIR"/red5.sh &
		sleep 2
		;;
	stop)
		echo Shutting down Red5
		killall java
		sleep 2
		;;
	restart)
		$0 stop
		$0 start
		;;
esac


chmod +x /usr/local/etc/rc.d/red5.sh	# Scripti yerinə yetirilən edirik

/etc/rc.conf faylının sonuna aşağıdakı sətri əlavə edirik:
red5_enable="YES"

reboot	# Serveri reboot edirik ki, görək servis özü avtomatik işə düşürmü
```

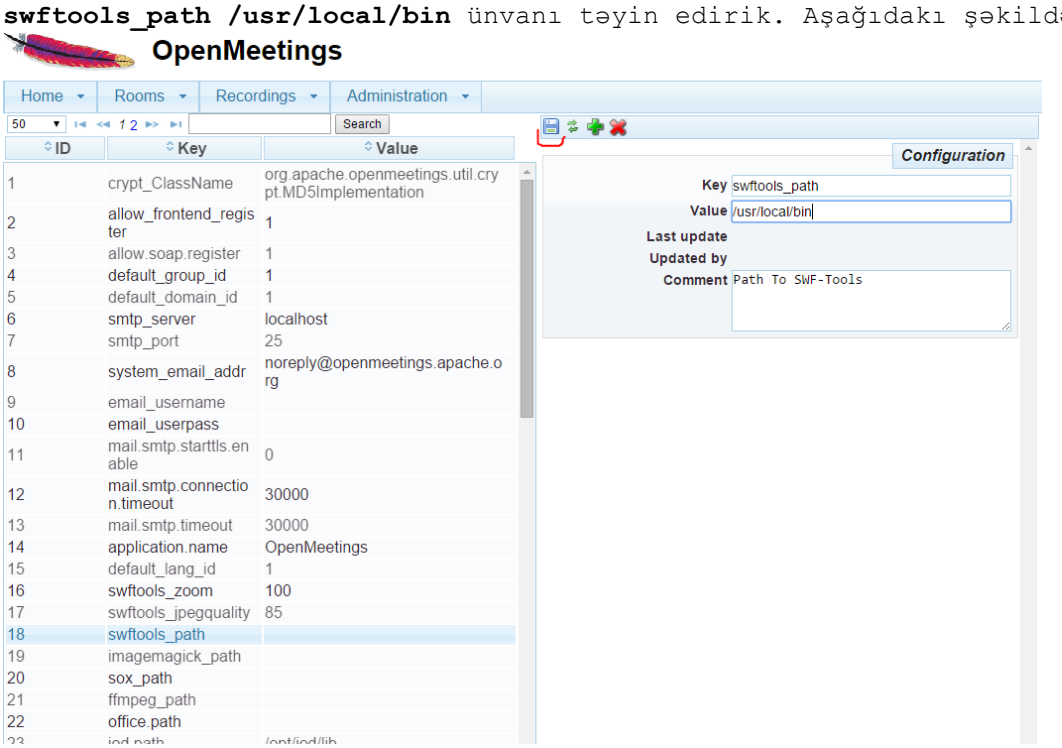
```
netstat -na | grep 5080        # reboot-dan sonra işə düşməsini
                               yoxlayırıq(nəticə aşağıdakı kimi olmalıdır)
tcp46      0      0 *.5080                  *.*                    LISTEN

ps waux | grep -v "grep" | grep red5      # Processlərdə olmasını yoxlayırıq
root    1103   1.0  9.5 1394160 397072  -  I     8:55PM 1:46.39
/usr/local/openjdk7/bin/java -Dred5.root=/usr/local/om -
Dlogback.ContextSelect
```

Indi isə əlavə quraşdırmaları edək

Paketleri yüklədikdə **swftools** var idi. O susmaya görə **/usr/local/bin** ünvanına yüklənir. Aşağıdakı əmr ilə yoxlaya bilərik. Həmçinin bu qovluqda **/usr/local/bin/pdf2swf** olmalıdır.

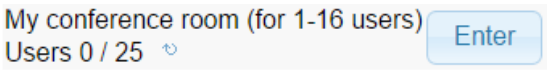
```
ll /usr/local/bin/swf*
-r-xr-xr-x  1 root  wheel  636848 Dec 23 15:41 /usr/local/bin/swfbbox*
-r-xr-xr-x  1 root  wheel  986832 Dec 23 15:41 /usr/local/bin/swfc*
-r-xr-xr-x  1 root  wheel  111312 Dec 23 15:41 /usr/local/bin/swfcombine*
-r-xr-xr-x  1 root  wheel  653344 Dec 23 15:41 /usr/local/bin/swfdump*
-r-xr-xr-x  1 root  wheel  676464 Dec 23 15:41 /usr/local/bin/swfextract*
-r-xr-xr-x  1 root  wheel  787120 Dec 23 15:41 /usr/local/bin/swfrender*
-r-xr-xr-x  1 root  wheel  628624 Dec 23 15:41 /usr/local/bin/swfstrings*
```

http://98.97.96.140:5080 linkimizdə **Administration -> Configuration** bölümündə **swftools_path /usr/local/bin** ünvanı təyin edirik. Aşağıdakı şəkildəki kimi:

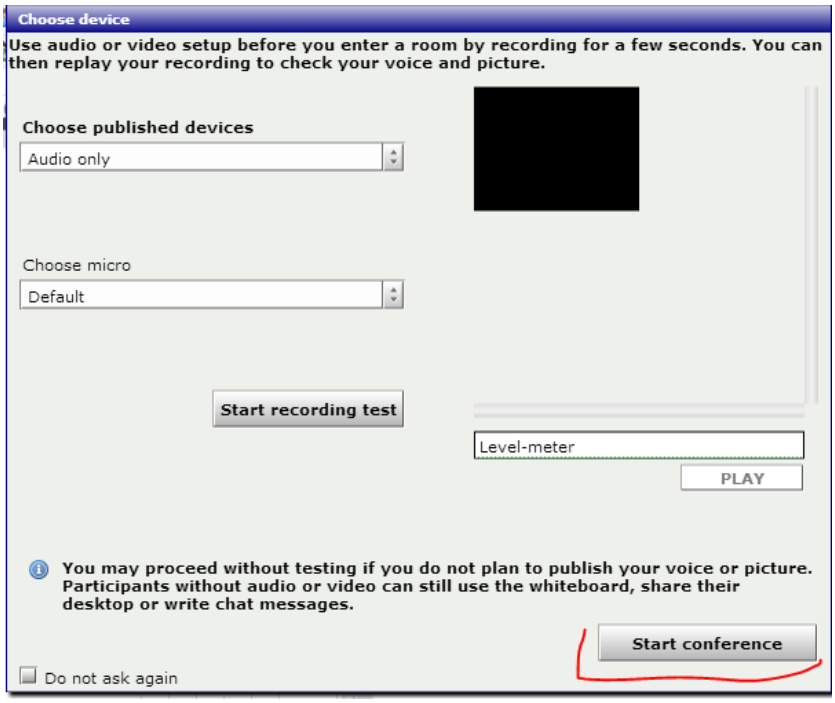


Sonra yazı otağında daxil oluruq. Bunun üçün **Rooms -> My rooms** ünvanına daxil oluruq.

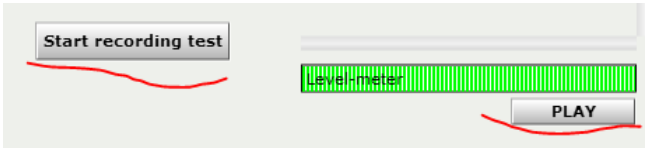
Sonra isə **My conference room -> Enter** düyməsini sıxırıq



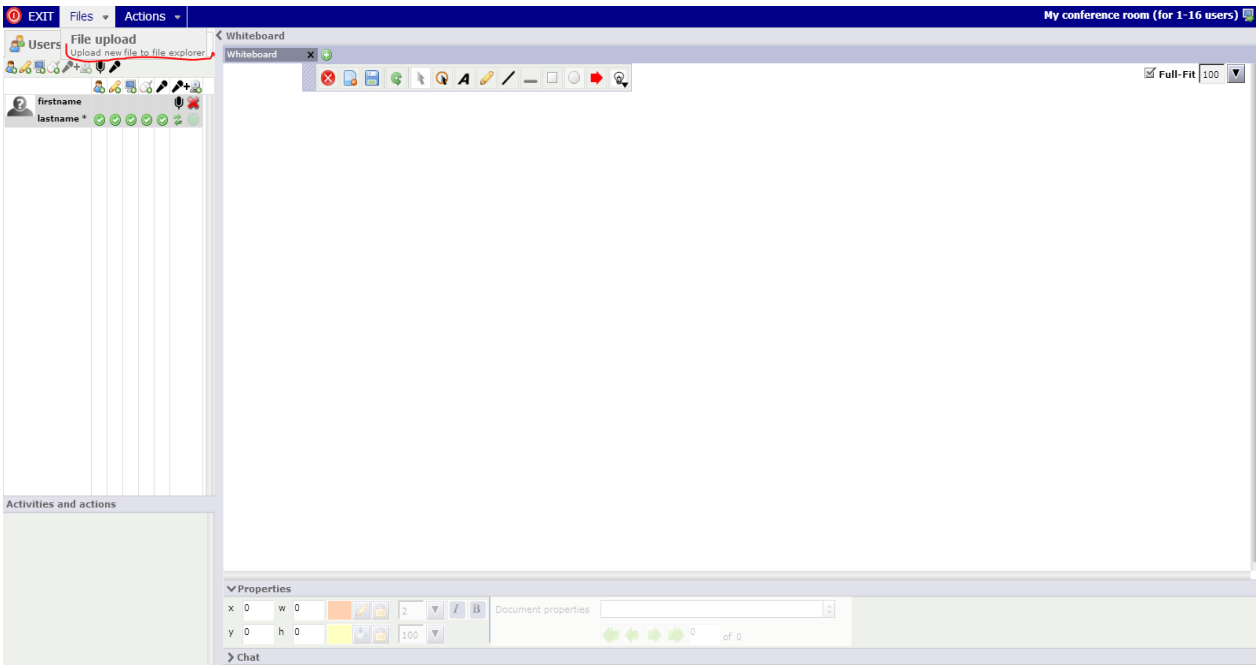
Əgər Java-nın avadanlıqlarımızın driverlərini istifadə edilməsi ilə bağlı Browserimiz və ekranımıza xəbər çıxsa mütləq **allow** edirik. Sonra isə **Start conference** düyməsinə sıxırıq.



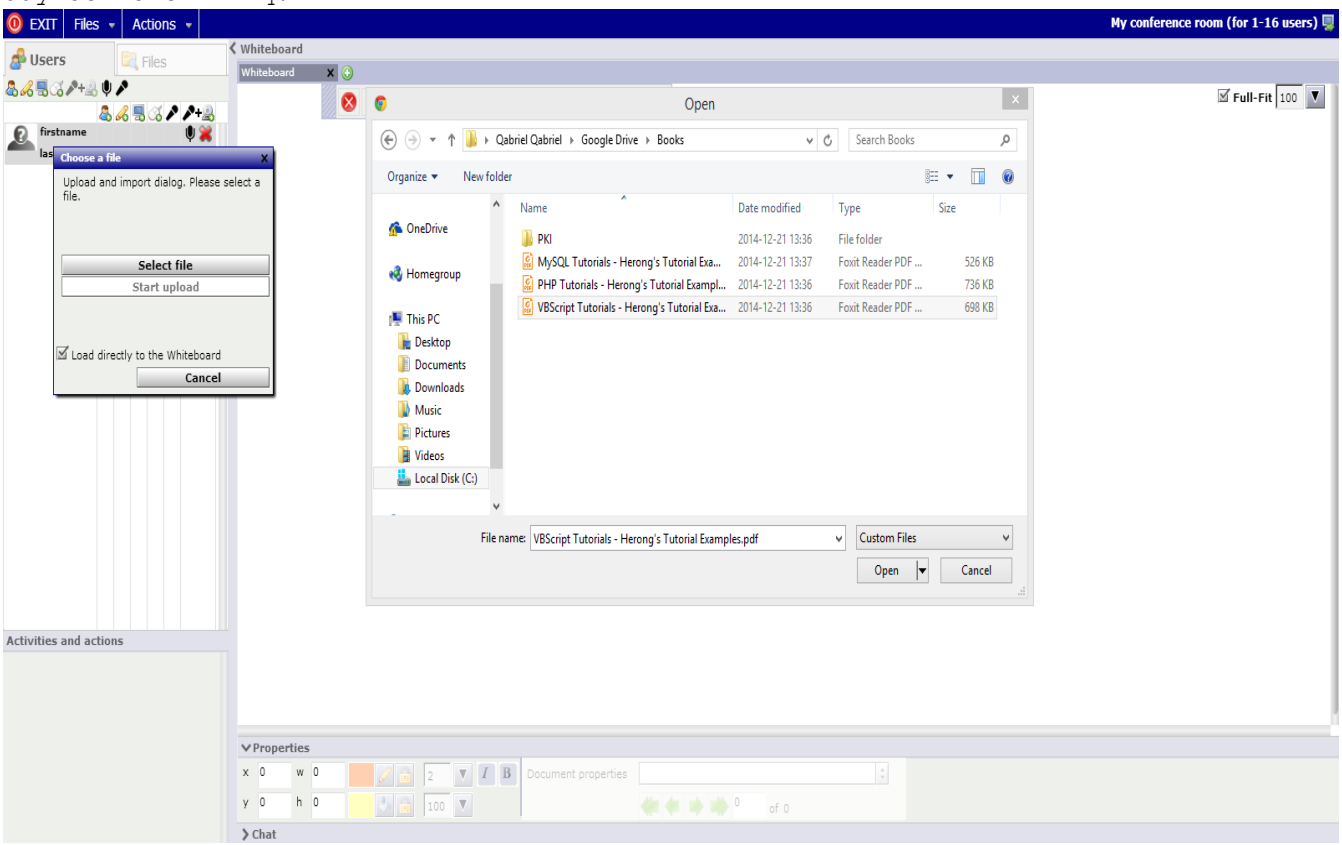
Həmçinin test üçün səsi yoxlaya bilərsiniz. Şəkildə göstərilir:



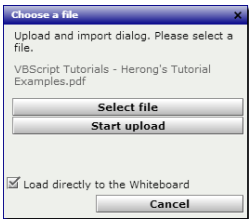
Sonra danışıq otağımızın içində **Files -> File upload** düyməsinə sıxırıq.



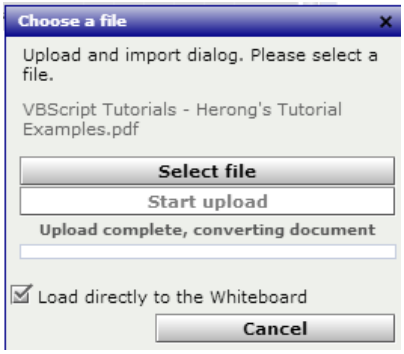
Sonra **Select file ->** Sistemdə PDF yerləşən ünvanda **PDF** faylı seşirik və **Open** düyməsinə sıxırıq.



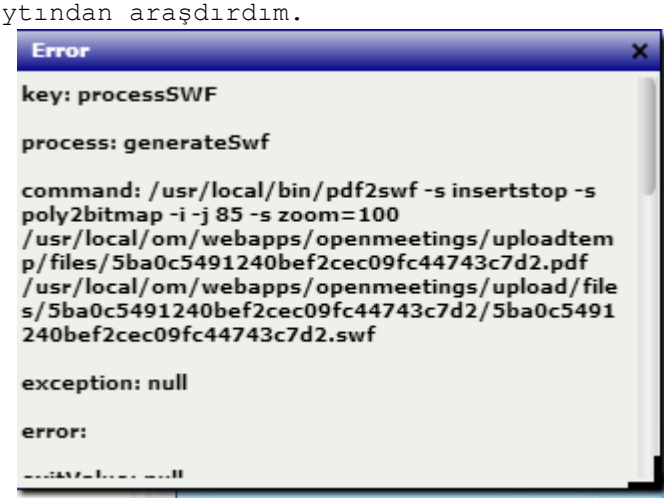
Sonra aşağıdakı şəkildəki kimi **Start upload** düyməsinə sıxırıq.



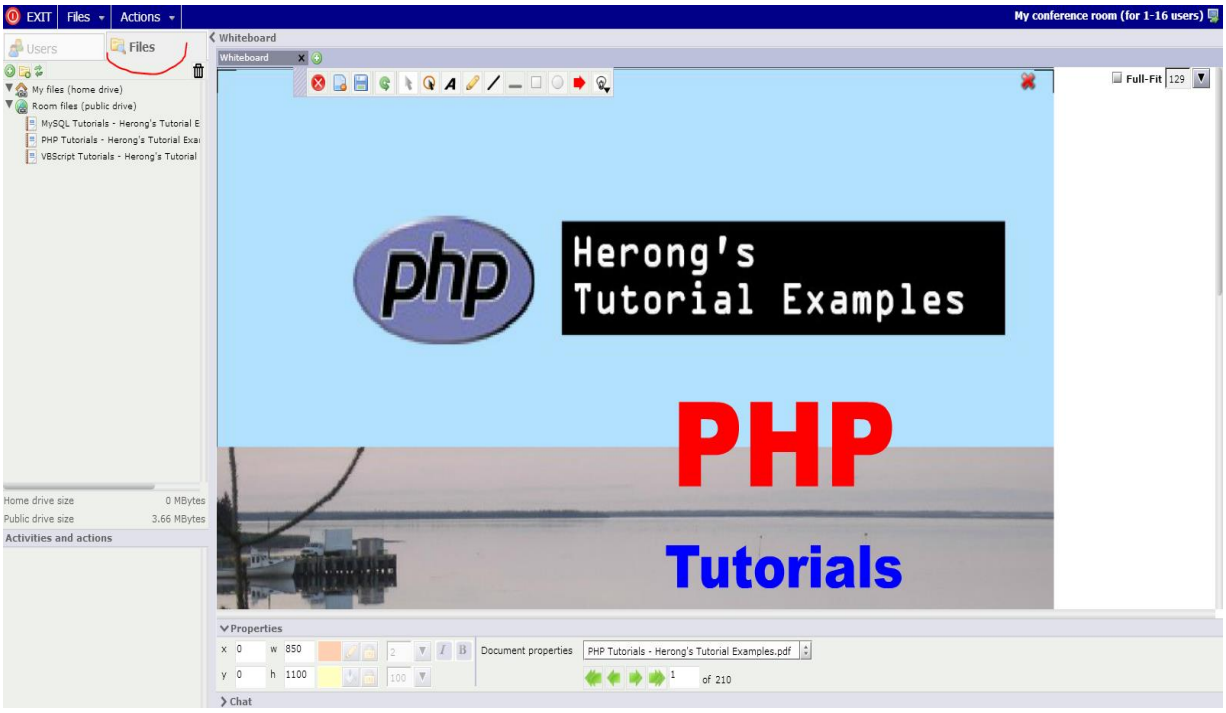
File Upload bitdikdən sonra isə faylın convert edilməsi başlayacaq(Şəkildəki kimi):



Convert bitdikdən sonra mənim halımda aşağıdakı səhv çap edildi. Bu Code səhvidir rəsmi saytından araşdırdım.



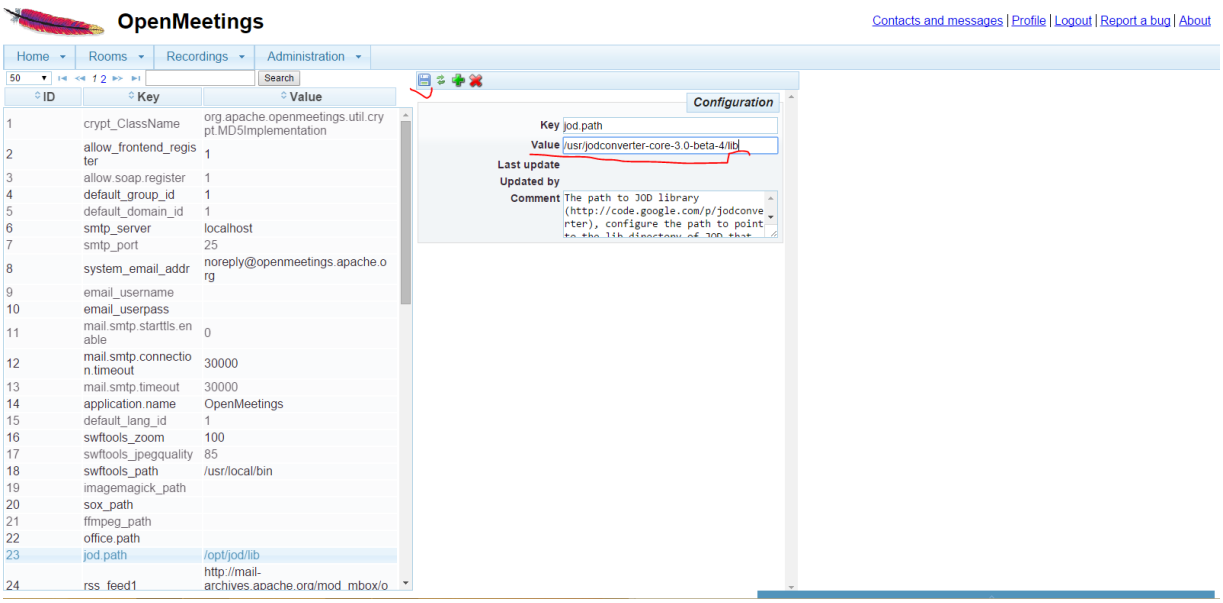
Ancaq PDF sənəd normal şəkildə convert edildi və **Files** bölümündə list edildi. Upload edilmiş PDF sənədləri **Files** bölümündə görə bilərsiniz.



Sonra ofis sənədlərinin import edilməsı üçün Libreoffice-i quraşdırırıq. Öncə paketi yükləmişdik. Mütləq **JodConverter** yükləmək lazımdır. Bunun üçün aşağıdakı likndən onu serverimizə dartırıq
https://code.google.com/p/jodconverter/downloads/detail?name=jodconverter-core-3.0-beta-4-dist.zip&can=2&q=

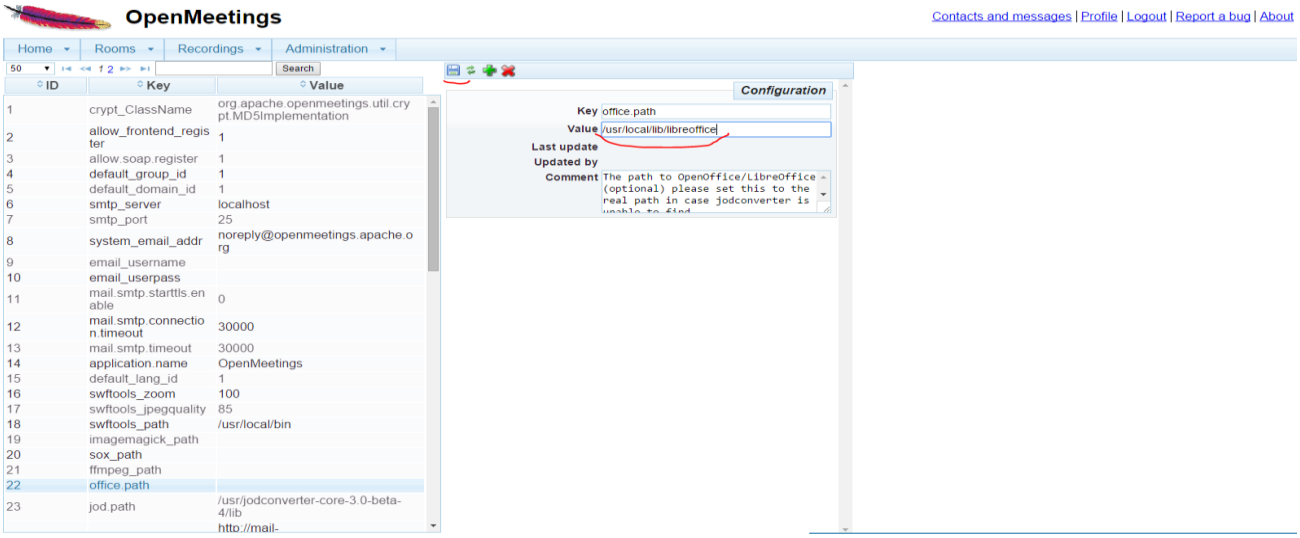
```
cd /usr                    # WinSCP ilə bu ünvana upload edirik.
cd /usr/ ; tar zxf jodconverter-core-3.0-beta-4-dist.zip     # Upload
                                                  etdiyimiz qovluqda açırıq
```

Sonra http://94.20.19.140:5080 serverimizdə **Administration -> Configuration** bölümündə **jod.path value**-sini **/usr/jodconverter-core-3.0-beta-4/lib** edirik və **save** düyməsini sıxırıq.



Sonra serverimizdə yüklənən ofisin binary ünvanını axtarıb tapırıq.
find / -name soffice.bin # Binar ünvanı axtarırıq və aşağıdakı ünvandır.
/usr/local/lib/libreoffice/program/soffice.bin

Sonra yenede **Administration -> Configuration** və **office.path** value-si olaraq **/usr/local/lib/libreoffice** təyin edirik. Şəkildə göstərildiyi kimi:



Qeyd: Diqqət əlavə edilən **PDF**, **.doc** və ya hansısa sənədlərin açılması və silinməsi üçün düymələr yox **Drag & Drop** işləyir. Nəzərə alın ki, onları istifadə edəsiniz.

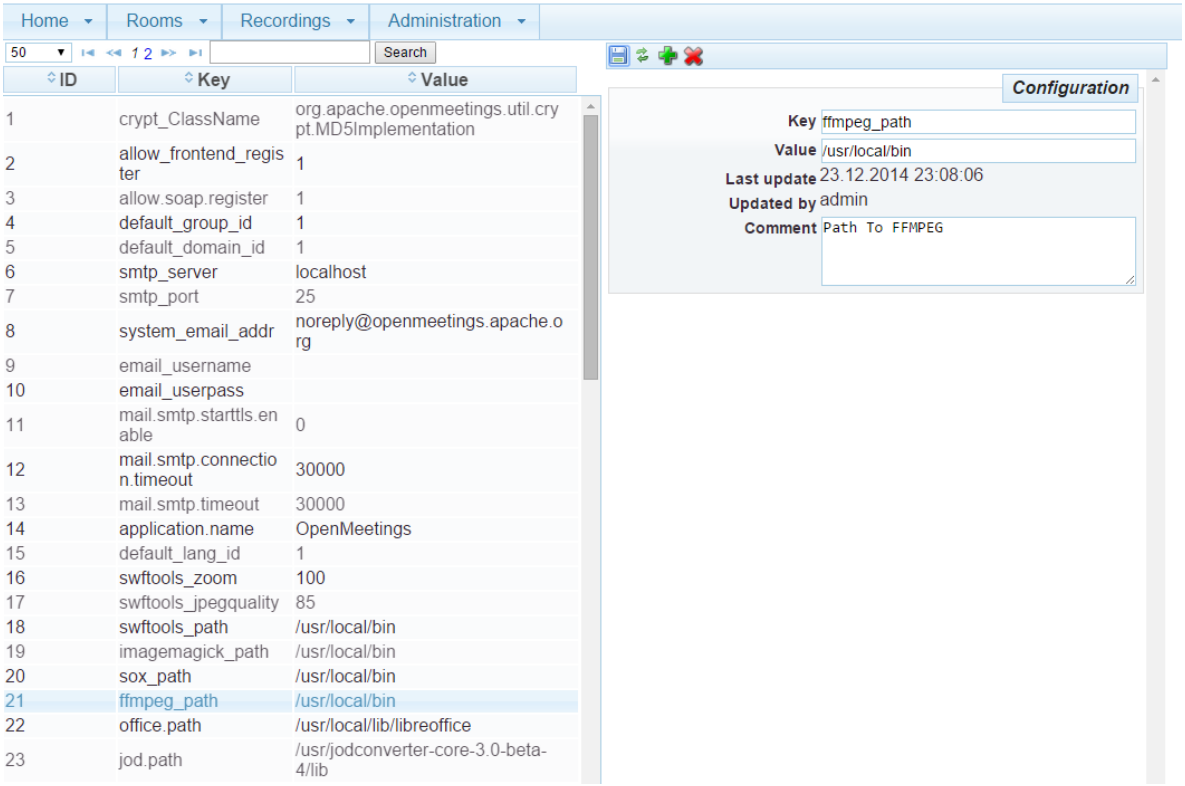
Sonra yenə də danışıq otağımız **Rooms -> My Rooms -> My Conference room -> Enter**, ardınca **Files -> MyFiles** və sol tərəfdə küncdə **Upload file** düyməsini

sıxırıq. Ancaq mənim halımda doc və docx sənəd convert edilə bilmədi ama PDF işlədi.

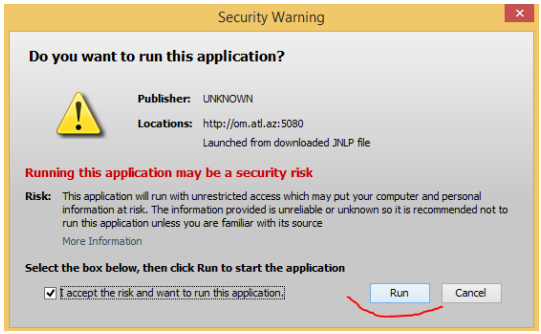
Həmçinin **Administration -> Configuration** bölümünün altında **sox, ffmpeg** və **imagemagick** üçün **keyvalue**-ları aşağıdakı kimi axtarıb sonra da şəkildəki kimi təyin etmək lazımdır.

```
which ffmpeg                          # ffmpeg ünvanını tapırıq
/usr/local/bin/ffmpeg

which sox                             # Sox ünvanını tapırıq
/usr/local/bin/sox

which /usr/local/bin/image_to_j2k     # ImageMacgik ünvanı
/usr/local/bin/image_to_j2k
```

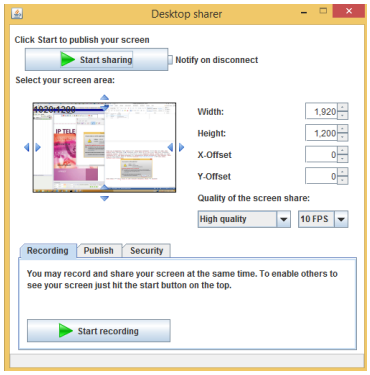
Şəkildəki kimi ünvanlar olur:



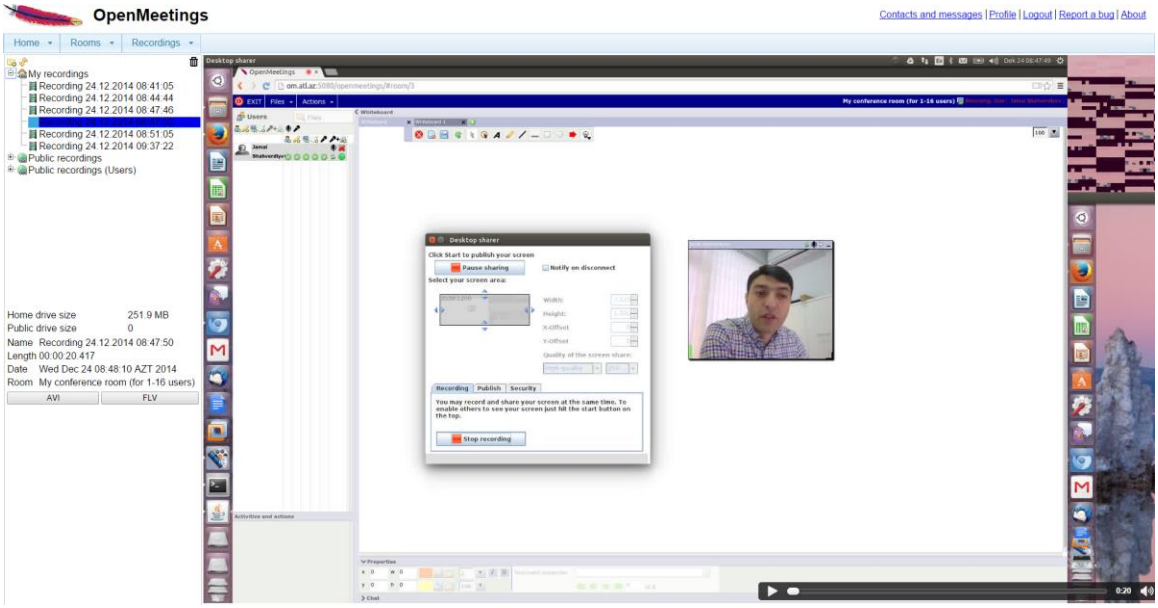
Sonda isə öz ekranımızı paylaşaq və baş vərən bütün hadisələri **record** edib test edək. Yenə gedirik **Rooms → My Rooms → My conference room → Enter** və Java-nı accept edib **Start Conference** düyməsinə sıxırıq. Sonra **Actions → Share/Record** screen düymesinə sıxırıq, bundan sonra java fayl yüklənəcək və biz onu açmaq istədikdə Java təhlükəsizlik menyusu açılacaq ki, izin verilmir. Bunun üçün siz **Java Control Panel**-de **Security** bölümündə http://om.domain.az:5080 -i **Add** edib inamlı siyahiya əlavə etməlisiniz.



Həmin java faylı açırıq və işə saldıqda aşağıdakı səhifə çap ediləcək:



Start sharing və **Start recording** düymələrinə sıxırıq və ekranımız yazılmağa başlayır. Sonra **Exit** düyməsini sıxırıq. Şəkildəki kimi, **Recordings -> Recordings(Watch recording and interviews) -> My recordings** düyməsinə sıxıb aşağıdakı şəkildə olan nəticəni əldə etmiş olacaqsınız:



BigBlueButton qurulması və istifadə edilməsi

BBB - web-konfransın keçirilməsi üçün açıq qaynaqlı proqram təminatıdır. Sistem ilk növbədə distans təhsil üçün hazırlanmışdır. OnlineMeetings-də olan bütün funksionallığa sahibdir lakin, BBB(BigBlueButton) öz API-larını md5 və salt alqoritmi ilə şifrələnmiş kanal üzərindən istənilən serverə qaytarır. Yəni birbaşa inzibatçı idarəetməsi üçün interfeysə sahib deyil.

Qeyd: Mütləq bu serverdə PUBLIC IP uzerində işləməlidir.

BBB serverin işləməsi üçün Ubuntu serverin tələbləri aşağıdakı kimi optimal sayılır:
DDR: 8GB
CPU: 2, Core2(2.6Ghz yada daha çox)
TCP portlar: 80, 1935 və 9123 açıq olmalıdır
UDP portlar: 16384-32768 aralığı açıq olmalıdır
HDD: 500GB
NIC: 1(Internet üzərində 100Megabit simmetric)

```
apt-get update                  # Bütün reposları yeniləyirik
apt-get dist-upgrade            # Kernel və system paketlərini yeniləyirik

reboot                    # Sistemə restart edirik ki, paketlər mənimsənsin

cat /etc/default/locale         # Sistemin daxili dili və kodlaşdırması belə
                                olmalıdır(susmaya görə belədir)
LANG="en_US.UTF-8"              # Qeyd: faylda yalnız bir sətir olmalıdır
```

Əgər daxili dil **en** və kodlaşdırma **UTF8** olmazsa, aşağıdakı əmr ilə bunu edə bilərsiniz:

```
apt-get install language-pack-en
update-locale LANG=en_US.UTF-8

uname -m                  # Ubuntu aşağıdakı tip platformada olmalıdır
x86_64
```

Həmçinin Ubuntu-nun 14.04 versiyası olmalısını mütləq yoxlayın.

```
cat /etc/lsb-release            # Əmri daxil etdikdə aşağıdakı nəticə çap
                                edilməlidir.
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04.1 LTS"
```

BigBlueButton üçün açarı yükləyək yəni repos əlavə edək və bütün reposları yeniləyək.

```
wget http://ubuntu.bigbluebutton.org/bigbluebutton.asc -O- | sudo apt-key add
-
echo "deb http://ubuntu.bigbluebutton.org/trusty-090/ bigbluebutton-trusty
main" | sudo tee /etc/apt/sources.list.d/bigbluebutton.list

apt-get update            # Reposları yeniləyirik
```

```
cat /root/install-ffmpeg.sh          # Fayla gördüyümüz kimi aşağıdakı
                                     sətirləri əlavə edirik
apt-get install build-essential git-core checkinstall yasm texi2html
libvorbis-dev libx11-dev libvpx-dev libxfixes-dev zlib1g-dev pkg-config
netcat libncurses5-dev

FFMPEG_VERSION=2.3.3

cd /usr/local/src
if [ ! -d "/usr/local/src/ffmpeg-${FFMPEG_VERSION}" ]; then
  wget "http://ffmpeg.org/releases/ffmpeg-${FFMPEG_VERSION}.tar.bz2"
  tar -xjf "ffmpeg-${FFMPEG_VERSION}.tar.bz2"
fi

cd "ffmpeg-${FFMPEG_VERSION}"
./configure --enable-version3 --enable-postproc --enable-libvorbis --enable-
libvpx
make
checkinstall --pkgname=ffmpeg --pkgversion="5:${FFMPEG_VERSION}" --backup=no
--deldoc=yes --default

chmod +x /root/install-ffmpeg.sh          # Scripti yerinə yetirilən edirik
                                          ki, işə sala bilək

/root/install-ffmpeg.sh                   # Scripti işə salırıq və ffmpeg
                                          paketi avtomatik olaraq yüklənəcək


ffmpeg -version          # Yüklənmə bitdikdən sonra ffmpeg versiyasını
                         yoxlayırıq. Aşağıdakı kimi olmalıdır.
ffmpeg version 2.3.3 Copyright (c) 2000-2014 the FFmpeg developers
built on Jan 24 2015 16:07:33 with gcc 4.8 (Ubuntu 4.8.2-19ubuntu1)
configuration: --enable-version3 --enable-postproc --enable-libvorbis --
enable-libvpx
libavutil      52. 92.100 / 52. 92.100
libavcodec     55. 69.100 / 55. 69.100
libavformat    55. 48.100 / 55. 48.100
libavdevice    55. 13.102 / 55. 13.102
libavfilter     4. 11.100 /  4. 11.100
libswscale      2.  6.100 /  2.  6.100
libswresample   0. 19.100 /  0. 19.100

apt-get update                # Reposları yenidən yeniləyirik
apt-get install bigbluebutton # Paketi yükləyirik

Bir dəfə səhv çap ediləcək. Fikir verməyin və yenidən əmri təkrarlayın.
apt-get install bigbluebutton        # Paketi yükləyirik(Bu dəfə səhv çap
                                     edilməyəycək)

apt-get install bbb-demo             # Test üçün bbb-demo-nu yükləyirik
```

Qeyd: Nəzərə alın ki, **bbb-demo** paketi yükləndikdən sonra public-də

hamı tərəfindən istifadə edilə biləcək. Onu silmək üçün isə **apt-get purge bbb-demo** əmrindən istifadə etmək lazımdır.

```
bbb-conf --enablewebrtc              # WebRTC audio işə salırıq və aşağıdakı
                                     nəticə əldə edilir
WebRTC audio enabled.  To apply settings to your server, do

      sudo bbb-conf --clean
```

Öncə IPv6-ni söndürürük. **/etc/sysctl.conf** faylına aşağıdakı sətirləri əlavə edirik:

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

/etc/default/grub faylında aşağıdakı sətri uyğun formaya gətiririk:

```
GRUB_CMDLINE_LINUX="ipv6.disable=1"
```

/etc/nginx/sites-enabled/default faylında mütləq aşağıdakı sətri silirik:

```
listen [::]:80 default_server ipv6only=on;
```

```
reboot            # Sonda server restart edirik
```

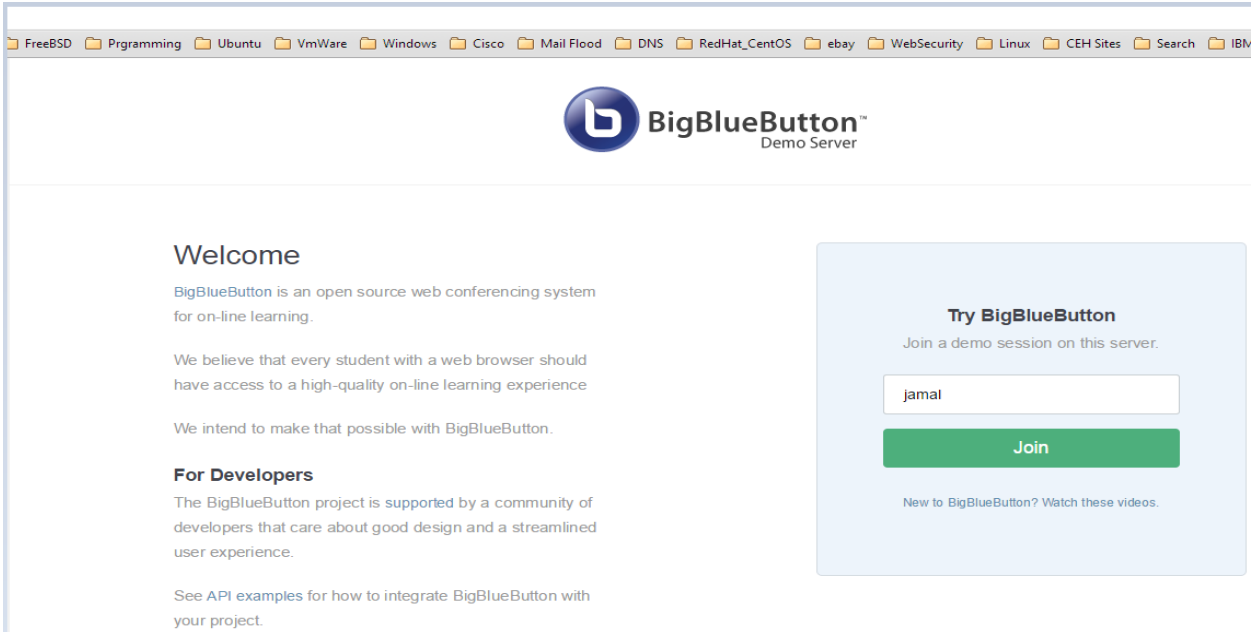
BigBlueButton-un normal işə düşməsini yoxlamaq üçün isə öncə təmizlik işləri edirik.

```
bbb-conf --clean              # Bu əmri daxil edirik
Doing a restart of BigBlueButton and cleaning out all log files...
 * Stopping daemon monitor monit                                    [ OK ]
 * Stopping Red5 Server red5                                 [ OK ]
 * Stopping Tomcat servlet engine tomcat7                    [ OK ]
Killing: 925
 * Stopping bbb-record-core

Cleaning Log Files ...
 * nginx is not running
 * Red5 Server is not running.
 * Tomcat servlet engine is not running.

1791 Backgrounding.
1791 (process ID) old priority 19, new priority -10
Waiting for FreeSWITCH to start: ......................
 * Starting Red5 Server red5                                 [ OK ]
 * Starting Tomcat servlet engine tomcat7                    [ OK ]
 * Starting daemon monitor monit                                   [ OK ]

Note: monit will automatically start bbb-record-core and LibreOffice within
60 seconds.

Waiting for BigBlueButton to finish starting up (this may take a minute):
.............. done


** Potential problems described below **
```

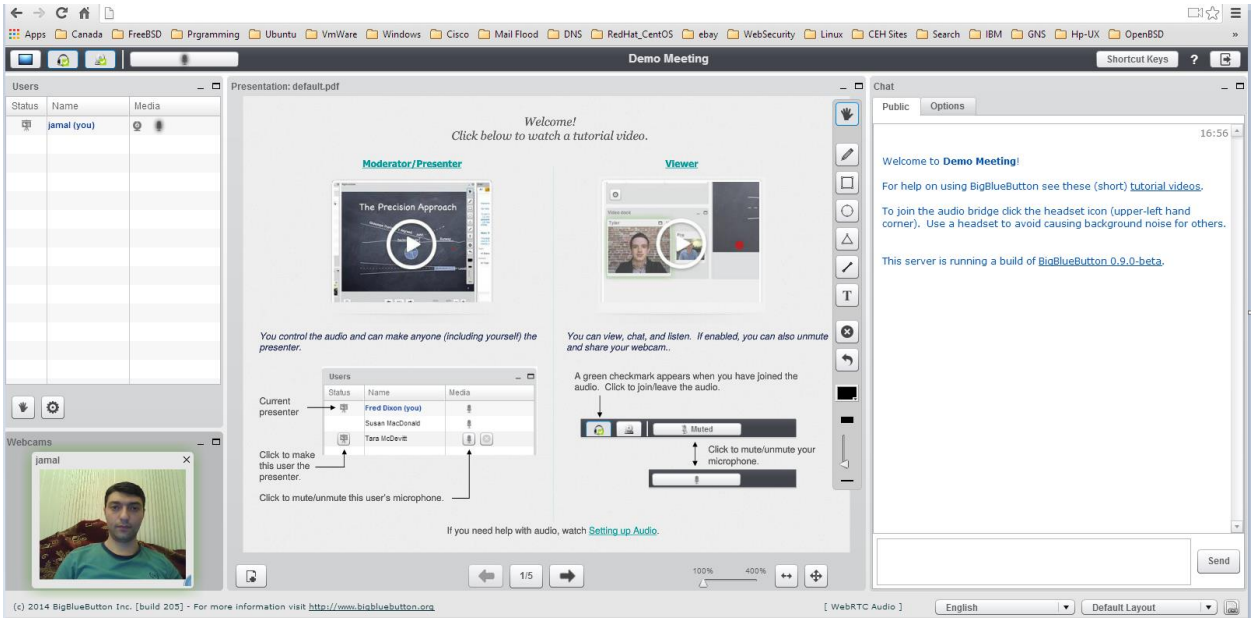
```
# Warning: The API demos are installed and accessible from:
#
#    http://188.99.88.76/
#
# These API demos allow anyone to access your server without authentication
# to create/manage meetings and recordings. They are for testing purposes
only.
# If you are running a production system, remove them by running:
#
#    sudo apt-get purge bbb-demo
```

Öncəki sətirlərdə gördüyümüz kimi yoxlanışın nəticəsi bizə bildirdi ki, bbb çölə açıqdır və hər kəs onu istifadə edə bilər. Və bunun qarşısını almaq üçün hansı əmrdən istifadə etmək lazım olduğu da çap edildi. Ancaq hələ ki, test edəcəyimiz üçün **bbb-demo** paketini silmirik.

http://bbb.domain.az/ linkinə daxil olduqda aşağıdakı səhifə açılacaq. Test üçün öz istifadəçi adını daxil edib giriş etdim(Aşağıdakı şəkildəki kimi). Sizdə sual yaranmasın ki, jamal adlı istifadəçi öncədən yaranmamışdı, bəli elədir axı öncə yazdığım kimi, demo hər kəs üçün açıqdır.



Bütün driverlərin düzgün işləməsi üçün browserimizdə çıxan hər bir suala **allow** cavabı veririk və nəticədə aşağıdakı oxşar səhifəni əldə edirik:



Serverimizin heç bir səhv olmadan normal yüklənib quraşdırılmasını yoxlanış edirik:

```
bbb-conf --check                    # Bu əmr ilə quraşdırmaları yoxlayırıq

BigBlueButton Server 0.9.0-beta (571)
                    Kernel version: 3.13.0-44-generic
                      Distribution: Ubuntu 14.04.1 LTS (64-bit)
                            Memory: 7984 MB
/var/www/bigbluebutton/client/conf/config.xml (bbb-client)
                 Port test (tunnel): 188.99.88.76
                                Red5: 188.99.88.76
              useWebrtcIfAvailable: true
/opt/freeswitch/conf/sip_profiles/external.xml (FreeSWITCH)
                    websocket port: 5066
                    WebRTC enabled: true

/etc/nginx/sites-available/bigbluebutton (nginx)
                       server name: 188.99.88.76
                              port: 80
                    bbb-client dir: /var/www/bigbluebutton

/var/lib/tomcat7/webapps/bigbluebutton/WEB-
INF/classes/bigbluebutton.properties (bbb-web)
                       bbb-web host: 188.99.88.76

/var/lib/tomcat7/webapps/demo/bbb_api_conf.jsp (API demos)
                             api url: 188.99.88.76

/usr/share/red5/webapps/bigbluebutton/WEB-INF/red5-web.xml (red5)
                   voice conference: FreeSWITCH
                      capture video: true
                    capture desktop: true
```

```
/usr/local/bigbluebutton/core/scripts/bigbluebutton.yml (record and playback)
                    playback host: 188.99.88.76

** Potential problems described below **
# Warning: The API demos are installed and accessible from:
#
#    http://188.99.88.76/
#
# These API demos allow anyone to access your server without authentication
# to create/manage meetings and recordings. They are for testing purposes
only.
# If you are running a production system, remove them by running:
#
#    sudo apt-get purge bbb-demo
```

Ümumiyyətlə mümkün olan biləcək əmrlərin sıyahısına aşağıdakı kimi baxa bilərsiniz:

```
bbb-conf -h              # Mövcud əmrlərin siyahısını əldə edirik.
BigBlueButton Configuration Utility - Version 0.9.0-beta
http://code.google.com/p/bigbluebutton/wiki/BBBConf

   bbb-conf [options]

Configuration:
   --version                        Display BigBlueButton version (packages)
   --setip <host>                   Set IP/hostname for BigBlueButton
   --setsecret <secret>             Change the shared secret in
bigbluebutton.properties

Monitoring:
   --check                          Check configuration files and processes
for problems
   --debug                          Scan the log files for error messages
   --watch                          Scan the log files for error messages
every 2 seconds
   --secret                         View the URL and shared secret for the
server
   --lti                            View the URL and secret for LTI (if
installed)

Administration:
   --restart                        Restart BigBueButton
   --stop                           Stop BigBueButton
   --start                          Start BigBueButton
   --clean                          Restart and clean all log files
   --zip                            Zip up log files for reporting an error

Testing:
   --enablewebrtc                   Enables WebRTC audio in the server
   --disablewebrtc                  Disables WebRTC audio in the server
```

Serverdə qulaq asılan portların sıyahısına baxaq:

```
netstat -na|grep -i LISTEN | grep -v unix
tcp        0      0 188.99.88.76:5090   0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:9123        0.0.0.0:*       LISTEN
tcp        0      0 127.0.0.1:8100      0.0.0.0:*       LISTEN
tcp        0      0 188.99.88.76:5060   0.0.0.0:*       LISTEN
tcp        0      0 127.0.0.1:8005      0.0.0.0:*       LISTEN
tcp        0      0 188.99.88.76:5066   0.0.0.0:*       LISTEN
tcp        0      0 127.0.0.1:6379      0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:5070        0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:9998        0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:1935        0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:9999        0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:80          0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:8080        0.0.0.0:*       LISTEN
tcp        0      0 127.0.0.1:8081      0.0.0.0:*       LISTEN
tcp        0      0 127.0.0.1:8021      0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:22          0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:5080        0.0.0.0:*       LISTEN
```

Client WebRTC error code-ları:

1001: !Websocket disconnected - WebSocket uğurla qoşuldu və indi qoşulmadan ayrıldı ona görə ki:
* Internet yoxdur
* Nginx-in restart edilməsinə səbəb ola bilər

1002: Websocket qoşulmasını etmək mümkün deyil - WebSocket qoşulması uğursuz oldu ona görə ki:
* Firewall tərəfindən ws protocol bağlıdır
* Server sönülüdür ya da düzgün quraşdırılmayıb

1003: Browser versiyası dəstəklənmir - Browser tələb edilən WebRTC API methodlarını istifadə etmir ona görə ki:
* Köhnəlmiş browserdir

1004: Zəngdə səhv baş verdi – Zəng edildi ancaq səhv baş verdi:
* Səhvlərin tam siyahısına bu linkdən -> **http://sipjs.com/api/0.6.0/causes/** baxa bilərsiniz

1005: Zəng səbəbsiz sona çatdı – Zəng uğurlu oldu ancaq, istifadəçi müraciət etmədən sona yetdi. Səbəbi:
* Bəlli deyil

1006: Zəng vaxtı bitdi - Kitabxanadan asılı olaraq baş verir:
* Firefox 33beta-da Mac-da səhv baş veriridi.

1007: ICE razılaşma uğursuz oldu - Browser və !FreeSWITCH portların razılaşmasına cəhd edir hansı ki, görüntü və axının istifadəsi üçün nəzərdə tutulur və uğursuz oldu ona görə ki:
* NAT qoşulmanı block edir
* UDP qoşulma/port-larını Firewall block edir

BÖLÜM 14

İP üzərindən səsin ötürülməsi

- **Asterisk VoIP serverin qurulması və sınaqdan keçirilməsi**
- **FreeSWITCH VoIP serverin qurulması və sınaqdan keçirilməsi**

Artıq gündəmin tələbi elə bir vaxta gəlib çatıb ki, telefon sisteminin qurulması üçün mini ATS və ya fiziki avadanlıq tələb edilmir. Əksər şirkətlər artıq zəngə kompyüterində olan proqram təminatından və ya mobil telefonundan cavab vermək istəyir. Bunun üçün xeyli proqram təminatları və hətta voice over İP-ni dəstəkləyən avadanlıqlar da mövcuddur. Yalnız bütün bu bahalı həlləri istifadə eləmək əvəzinə, başlığımızda açıq qaynaqlı Asterisk/FreeSWITCH haqqında danışacayıq və bir neçə telefon üçün quraşdırmalar edəcəyik.

Asterisk VoIP serverin qurulması və sınaqdan keçirilməsi

Asterisk — kompüterlər üçün açıq qaynaqlı telefon həllidir. Mark Spencer tərəfindən hazırlanmışdır. Proqram Linux, FreeBSD, OpenBSD və Solaris əməliyyat sistemlərində işləyir. Layihənin adı "*" simvolundan yaranmışdır. (asterisk — "ulduz"). Daha ətraflı http://asterisk.ru/ linkindən oxuya bilərsiniz.

Asterisk istənilən avadanlıqla(server və ya PC) klassik ATS-ın bütün imkanlarına malikdir. Bir çox VoIP-protokolları dəstəkləyir və onların arasında zəngləri idarə etmə funksiyaları da var. Aşağıdakı bacarıqları sadalaya bilərik:

- Səsli məktub
- Konfrans əlaqə
- İVR(İnteraktiv səs menyusu)
- Zənglərin emalı mərkəzi(fərqli alqoritmlər istifadə edərək, zənglərin növbəli təyinatı və abonentlərə görə paylaşdırılması)
- Call Detail Record(zəng haqqında ətraflı məlumat)

Qurulmasına başlayaq.
asterisk.opensource.az adlı serverimiz var.
FreeBSD 10.1 x64 üzərində Asterisk13-ü aşağıdakı əmrlərlə yükləyirik:

```
# cd /usr/ports/net/asterisk13/      - Port ünvanına daxil oluruq
# make config                        - Lazımi modulları seçirik
```

```
qqqqqqqqqqqqqqqqqqqqqqqqqq asterisk13-13.5.0_1 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x+[ ] BACKTRACE  Stack backtrace support via (lib)execinfo                    x
x [x] CURL        Data transfer support via cURL                             x
x [x] DAHDI       DAHDI support                                              x
x [ ] EXCHANGE    Exchange calendar support                                  x
x [x] FREETDS     FreeTDS library support                                    x
x [ ] LDAP        LDAP protocol support                                      x
x [x] LUA         Lua scripting language support                             x
x+[x] MP3PLAYER   Install MP3 Player for Music-On-Hold (mpg123)              x
x [x] OOH323      ooh323 support                                             x
x [x] PJSIP       Build the PJSIP based SIP channel                          x
x [x] PORTAUDIO   PortAudio library support                                  x
x [x] RADIUS      RADIUS protocol support                                    x
x [x] SNMP        SNMP network protocol support                              x
x [ ] SPANDSP     SpanDSP faxing support                                     x
x [x] SRTP        SecureRTP support                                          x
x [ ] SYSINFO     Use devel/libsysinfo to get system information             x
x [x] XMPP        XMPP/GTALK support                                         x
xqqqqqqqqqqqqqqqqqqqqq Encoder/Decoder (Codec) Support qqqqqqqqqqqqqqqqqqqqx
x [x] GSM         GSM codec support                                          x
x [x] NEWG711     New G711 Codec                                             x
x [x] SPEEX       Speex audio format support                                 x
x [x] VORBIS      Ogg Vorbis audio codec support                             x
xqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Database Support qqqqqqqqqqqqqqqqqqqqqqqqqqqqx
x [ ] MYSQL       MySQL database support                                     x
x [x] ODBC        ODBC database backend                                      x
x [ ] PGSQL       PostgreSQL database support                                x
x [ ] SQLITE2     SQLite 2 database support                                  x
xqqqqqqqqqqqqqqqqq Core and Music-on-Hold (MoH) Sound Files qqqqqqqqqqqqqqqqx
x+[ ] G729                                                                   x
xqqqqqqqqqqqqqqqqqqqqqqqq Menuselect Interface Backend qqqqqqqqqqqqqqqqqqqqx
x+[x] NCURSES     Console (text) interface support                           x
x+[x] NEWT        Newt User Interface                                        x
xqqqqqqqqqqqqqqqqqqqqqqqq Compiler to use qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqx
x ( ) BASE        Use base compiler (experimental)                           x
x (*) GCC         Use gcc from ports                                         x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                      < OK >                  <Cancel>
```

```
# make -DBATCH install               - Yükləyirik
```

```
# echo 'asterisk_enable="YES"' >> /etc/rc.conf  - StartUP-a əlavə edirik
# /usr/local/etc/rc.d/asterisk start            - Işə salırıq
```

/usr/local/etc/asterisk/sip.conf faylında yalnız aşağıdakı sətirlərdə dəyişiklik etmişik:

```
transport=udp,tcp
tcpenable=yes
```

/usr/local/etc/asterisk/sip.conf faylının sonuna aşağıdakı sətiri əlavə edib yadda saxlayaraq çıxırıq(Bununla **sip_additional.conf** faylının da yüklənmə zamanı oxunmasını deyirik):

```
#include sip_additional.conf
```

Eynilə **/usr/local/etc/asterisk/extensions.conf** faylının sonuna aşağıdakı sətiri əlavə edirik ki, yüklənmədə **extensions_fs.conf** faylı da oxunsun:

```
#include extensions_fs.conf
```

/usr/local/etc/asterisk/sip_additional.conf faylında iki ədəd genişlənmə(**7000** və **7001**) və **fsar** adlı SIP Trunk quraşdırması olacaq. faylının tərkibi aşağıdakı kimi olacaq:

```
[7000]
defaultuser=7000
secret=freebsd
host=dynamic
context=phones
qualify=yes
transport=udp,tcp
insecure=port,invite
canreinvite=no
disallow=all
allow=alaw
type=friend

[7001]
defaultuser=7001
secret=freebsd
host=dynamic
context=phones
qualify=yes
transport=udp,tcp
insecure=port,invite
canreinvite=no
disallow=all
allow=alaw
type=friend
```

/usr/local/etc/asterisk/extensions_fs.conf faylının tərkibi aşağıdakı kimidir:

```
[incoming]
exten => _7XXX,1,Dial(SIP/${EXTEN})
```

```
exten => _7XXX,n,Hangup()

[outgoing]
exten => _1XXX,1,Dial(SIP/${EXTEN})
exten => _1XXX,n,Hangup()

[phones]
include => incoming
include => outgoing
```

/usr/local/etc/rc.d/asterisk restart – Asteriski yenidən yükləyirik ki, dəyişikliklər işə düşsün.

Ya da etdiyimiz dəyişikliklərin dərhal işə düşməsi üçün asterisk console-da aşağıdakı əmri daxil etməniz yetər:

```
asterisk*CLI> sip reload
```

Sonda Asterisk console-a verbose rejimdə daxil oluruq və uğurlu SIP qoşulmalarına baxırıq:

```
# asterisk -rvvv
asterisk*CLI> sip show peers
Name/username Host            Dyn Forcerport Comedia ACL Port  Status  Description
7000/7000     85.132.57.60    D   Auto(No)   No      53945    OK (11 ms)
7001/7001     (Unspecified)   D   Auto(No)   No      0        UNKNOWN
2 sip peers [Monitored: 1 online, 1 offline Unmonitored: 0 online, 0 offline]
```

SİP Debug eləmək üçün aşağıdakı əmrdən istifadə edə bilərsiniz:

```
snort*CLI> sip set debug on
```

Phones adlı yaratdığınız yeni dialplan-a aşağıdakı əmrlə baxa bilərsiniz:

```
snort*CLI> dialplan show phones
[ Context 'phones' created by 'pbx_config' ]
  Include =>        'incoming'
[pbx_config]
  Include =>        'outgoing'
[pbx_config]

-= 0 extensions (0 priorities) in 1 context. =-
```

Sonda isə iki SİP client proqramı ilə **7000** və **7001** qeydiyyatdan keçirib bir-birlərinə çox rahatlıqla zəng edə bilərsiniz.

FreeSWITCH VoIP serverin qurulması və sınaqdan keçirilməsi

FreeSWITCH - açıq qaynaqlı VoİP proqram təminatıdır. VoİP-lə ağlınıza gələcək istənilən imkana sahibdir. APİ mövcuddur və Event prinsipi ilə işləyir. Haqqında daha ətraflı https://ru.wikipedia.org/wiki/FreeSWITCH linkindən oxuya bilərsiniz.

Məqsədimiz **FreeBSD 10.1** x64 maşının üzərində FreeSWITCH serverin yüklənməsi və WEB ilə quraşdırılmasıdır. Bunun üçün öncə **FAMP**(**FreeBSD Apache MySQL PHP**) quraşdırmaq lazımdır. Ancaq php5-extensions-da mütləq aşağıdakı modulları seçmək lazımdır:

```
qqqqqqqqqqqqqqqqqqqqqqqqq php5-extensions-1.7 qqqqqqqqqqq
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x+[ ] BCMATH         bc style precision math functions
x+[x] BZ2             bzip2 library support
x+[ ] CALENDAR        calendar conversion support
x+[x] CTYPE           ctype functions
x+[x] CURL            CURL support
x+[ ] DBA             dba support
x+[x] DOM             DOM support
x+[x] EXIF            EXIF support
x+[ ] FILEINFO        fileinfo support
x+[x] FILTER          input filter support
x+[x] FTP             FTP support
x+[x] GD              GD library support
x+[x] GETTEXT         gettext library support
x+[ ] GMP             GNU MP support
x+[x] HASH            HASH Message Digest Framework
x+[x] ICONV           iconv support
x+[ ] IMAP            IMAP support
x+[ ] INTERBASE       Interbase 6 database support (Firebird)
x+[x] JSON            JavaScript Object Serialization support
x+[ ] LDAP            OpenLDAP support
x+[ ] MBSTRING        multibyte string support
x+[ ] MCRYPT          Encryption support
x+[ ] MSSQL           MS-SQL database support
x+[x] MYSQL           MySQL database support
x+[x] MYSQLI          MySQLi database support
x+[ ] ODBC            ODBC support
x+[x] OPENSSL         OpenSSL support
x+[ ] PCNTL           pcntl support (CLI only)
x+[ ] PDF             PDFlib support (implies GD)
x+[x] PDO             PHP Data Objects Interface (PDO)
x+[ ] PDO_DBLIB       PDO DBLIB-DB driver
x+[ ] PDO_FIREBIRD    PDO Firebird driver
x+[x] PDO_MYSQL       PDO MySQL driver
x+[ ] PDO_ODBC        PDO ODBC driver
x+[ ] PDO_PGSQL       PDO PostgreSQL driver
x+[ ] PDO_SQLITE      PDO sqlite driver
x+[ ] PGSQL           PostgreSQL database support
x+[x] PHAR            phar support
x+[x] POSIX           POSIX-like functions
x+[ ] PSPELL          pspell support
x+[ ] READLINE        readline support (CLI only)
x+[ ] RECODE          recode support
x+[x] SESSION         session support
```

```
x+[ ] SHMOP          shmop support                    :
x+[x] SIMPLEXML      simplexml support
x+[ ] SNMP           SNMP support
x+[ ] SOAP           SOAP support
x+[ ] SOCKETS        sockets support
x+[ ] SQLITE3        sqlite3 support
x+[ ] SYBASE_CT      Sybase database support
x+[ ] SYSVMSG        System V message support
x+[ ] SYSVSEM        System V semaphore support
x+[ ] SYSVSHM        System V shared memory support
x+[ ] TIDY           TIDY support
x+[x] TOKENIZER      tokenizer support
x+[ ] WDDX           WDDX support (implies XML)
x+[x] XML            XML support
x+[x] XMLREADER      XMLReader support
x+[x] XMLRPC         XMLRPC-EPI support
x+[x] XMLWRITER      XMLWriter support
x+[x] XSL            XSL support (Implies DOM)
x+[x] ZIP            ZIP support
x+[x] ZLIB           ZLIB support
```

```
cat /usr/local/domen/fussip.domain.az     # VirtualHost-un quraşdırması
                                          aşağıdakı kimi olacaq
<VirtualHost *:80>
        ServerAdmin jamal.shahverdiyev@domain.az
        ServerName fussip.domain.az
        AcceptPathInfo On
        DocumentRoot /usr/local/www/fusionpbx/
<Directory "/usr/local/www/fusionpbx">
        AllowOverride All
        Require all granted
</Directory>
</VirtualHost>
```

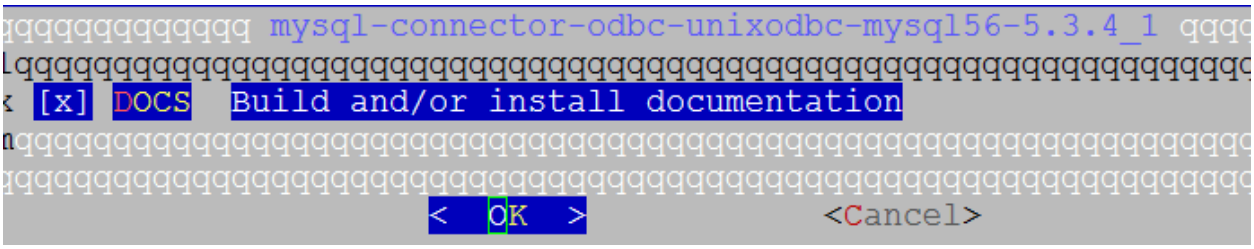
Qeyd: Əgər biz FreeSWITCH-in core səviyyədə ODBC dəstəklənməsini istəsək onda,
ODBC-ə aid olan paketləri yükləməliyik. Bunu aşağıdaki kimi edə bilərsiniz.

```
root@frfs:~ # cd /usr/ports/databases/unixODBC
root@frfs:/usr/ports/databases/unixODBC # make config
```

```
qqqqqqqqqqqqqqqqqqqqqqqqqqqq unixODBC-2.3.4 qqqqqqq
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
x+[x] DOCS  Build and/or install documentation
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                    <  OK  >            <Cancel>
```

```
root@frfs:~ # make -DBATCH install

MySQL üçün  MySQL connector-dan istifadə edilir
root@frfs:~ # cd /usr/ports/databases/mysql-connector-odbc
```

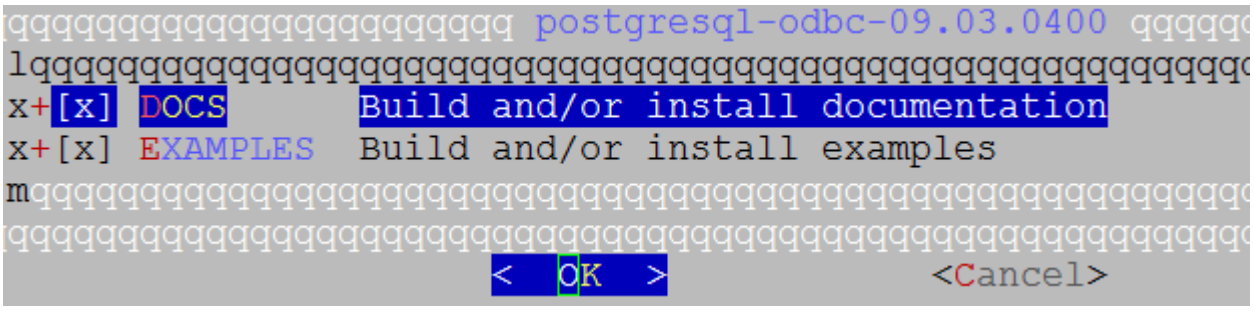
```
root@frfs:~ # make config
```




```
root@frfs:~ # make -DBATCH install
```

PostgreSQL üçün PostgreSQL connector-dan istifadə edilir:

```
root@frfs:~ # cd /usr/ports/databases/postgresql-odbc/
root@frfs:~ # make config
```




```
root@frfs:~ # make -DBATCH install
```

Sonra isə FreeSWITCH-in yüklənməsinə başlayırıq. Öncə kompilyasiya mühiti yaratmalıyıq. Məhz bunun üçündə lazımi paketləri yükləyirik.

```
pkg install autoconf automake curl git gmake jpeg ldns libedit libtool
openssl pcre pkgconf speex sqlite3 wget sudo
```

```
mkdir ~/src        # Mənbə kodları endirəcəyimiz ünvanı yaradırıq
cd ~/src           # Source kod-ların ünvanına daxil oluruq
```

Mənbə kodları local qovluğumuza sinxronizasiya edirik:

```
git clone -b v1.5.final https://stash.freeswitch.org/scm/fs/freeswitch.git
```

```
cd freeswitch           # clone edilən qovluğa daxil oluruq
./bootstrap.sh -j       # Kompilyasiya mühitini hazırlayırıq
```

Qeyd: Biz FreeSWITCH-in core səviyyədə ODBC dəstəklənməsini istəsək, mütləq onu aşağıdakı göstərilən şəkildə kompilyasiya etməliyik.

```
./configure --enable-core-odbc-support
```

```
./configure  # Quraşdırırıq(Bitdikdən sonra aşağıdakı sətirləri görməliyik)
------------------------- FreeSWITCH configuration ------------------------
  Locations:
      FHS enabled:      no

      prefix:           /usr/local/freeswitch
      exec_prefix:      ${prefix}
```

```
      bindir:          ${exec_prefix}/bin
      sysconfdir:      /usr/local/freeswitch/conf
      libdir:          ${exec_prefix}/lib

      certsdir:        /usr/local/freeswitch/certs
      dbdir:           /usr/local/freeswitch/db
      grammardir:      /usr/local/freeswitch/grammar
      htdocsdir:       /usr/local/freeswitch/htdocs
      logfiledir:      /usr/local/freeswitch/log
      modulesdir:      /usr/local/freeswitch/mod
      pkgconfigdir:    ${exec_prefix}/lib/pkgconfig
      recordingsdir:   /usr/local/freeswitch/recordings
      runtimedir:      /usr/local/freeswitch/run
      scriptdir:       /usr/local/freeswitch/scripts
      soundsdir:       /usr/local/freeswitch/sounds
      storagedir:      /usr/local/freeswitch/storage
      cachedir:        /usr/local/freeswitch/cache
-----------------------------------------------------------------------------
```

gmake # Kompilyasiyaya başlayırıq

sudo gmake install cd-sounds-install cd-moh-install # səsləri və imkanları yükləyirik

/usr/local/etc/rc.d/freeswitch faylı yaradırıq və içinə aşağıdakı sətirləri əlavə edirik. Bu fayl freeswitch üçün startup scriptdir hansı ki, reboot-dan sonra işə salınması üçündür.

```
#!/bin/sh
#
# PROVIDE: freeswitch
# REQUIRE: LOGIN cleanvar
# KEYWORD: shutdown
#
# Add the following lines to /etc/rc.conf to enable freeswitch:
# freeswitch_enable:       Set it to "YES" to enable freeswitch.
#                          Default is "NO".
# freeswitch_flags:        Flags passed to freeswitch-script on startup.
#                          Default is "".
#
. /etc/rc.subr
name="freeswitch"
rcvar=${name}_enable
load_rc_config $name
: ${freeswitch_enable="NO"}
: ${freeswitch_pidfile="/usr/local/freeswitch/run/freeswitch.pid"}
start_cmd=${name}_start
stop_cmd=${name}_stop
pidfile=${freeswitch_pidfile}
freeswitch_start() {
        /usr/local/freeswitch/bin/freeswitch ${freeswitch_flags}
        echo -n "Starting FreeSWITCH: "
}
```

```
freeswitch_stop() {
        /usr/local/freeswitch/bin/freeswitch -stop
}
run_rc_command "$1"
```

chmod u-w,ugo+x /usr/local/etc/rc.d/freeswitch # Scripti yerinə yetirən edirik

/etc/rc.conf faylına lazımi sətirləri əlavə edirik ki, reboot-dan sonra freeswitch-i işə salsın:

freeswitch_enable="YES"
freeswitch_flags="-nc"

Hal-hazırda root adlı istifadəçimizin SHELL mühiti **CSH** olduğuna görə, **/root/.cshrc** faylıda path dəyişənini aşağıdakı formaya gətiririk(freswitch-in binary faylları **/usr/local/freeswitch/bin** ünvanında yerləşir):

set path = (/sbin /bin /usr/sbin /usr/bin /usr/local/freeswitch/bin /usr/local/sbin /usr/local/bin $HOME/bin)

-nc - no console deməkdir ancaq, siz sonra **fs_cli** əmri ilə console-a daxil ola biləcəksiniz

-nonat – Əgər sizin freeswitch-in PUBLIC IP ünvanı varsa və o NAT arxasında işləmirsə, bu parametr istifadə edilir(Bu freeswitch üçün NAT traversal parametrini söndürür).

Sonra fusionpbx üçün baza, istifadəçi adı və şifrə yaradırıq:

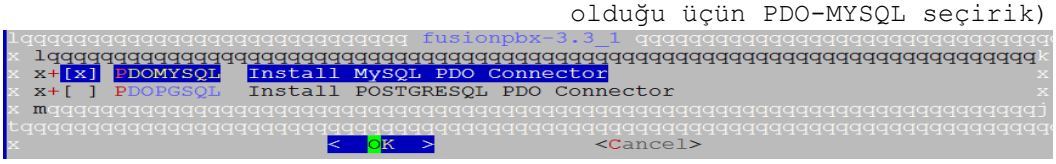
mysql -uroot -pfreebsd # Bazamıza daxil oluruq
CREATE database freeswitch;

GRANT ALL PRIVILEGES ON freeswitch.* TO 'freeswitch'@'localhost' IDENTIFIED BY 'freebsd';

FLUSH PRIVILEGES; # Bu əmri daxil edirik ki, son dəyişikliklər dərhal işə düşsün

Indi isə FusionPBX-i yükləyək və quraşdıraq. Əvvəl portlardan yükləyək və sonra WEB quraşdırmasını edək.

cd /usr/ports/www/fusionpbx/ # Port ünvanına daxil oluruq
make config # lazımi modulları seçirik(Baza Mysql olduğu üçün PDO-MYSQL seçirik)



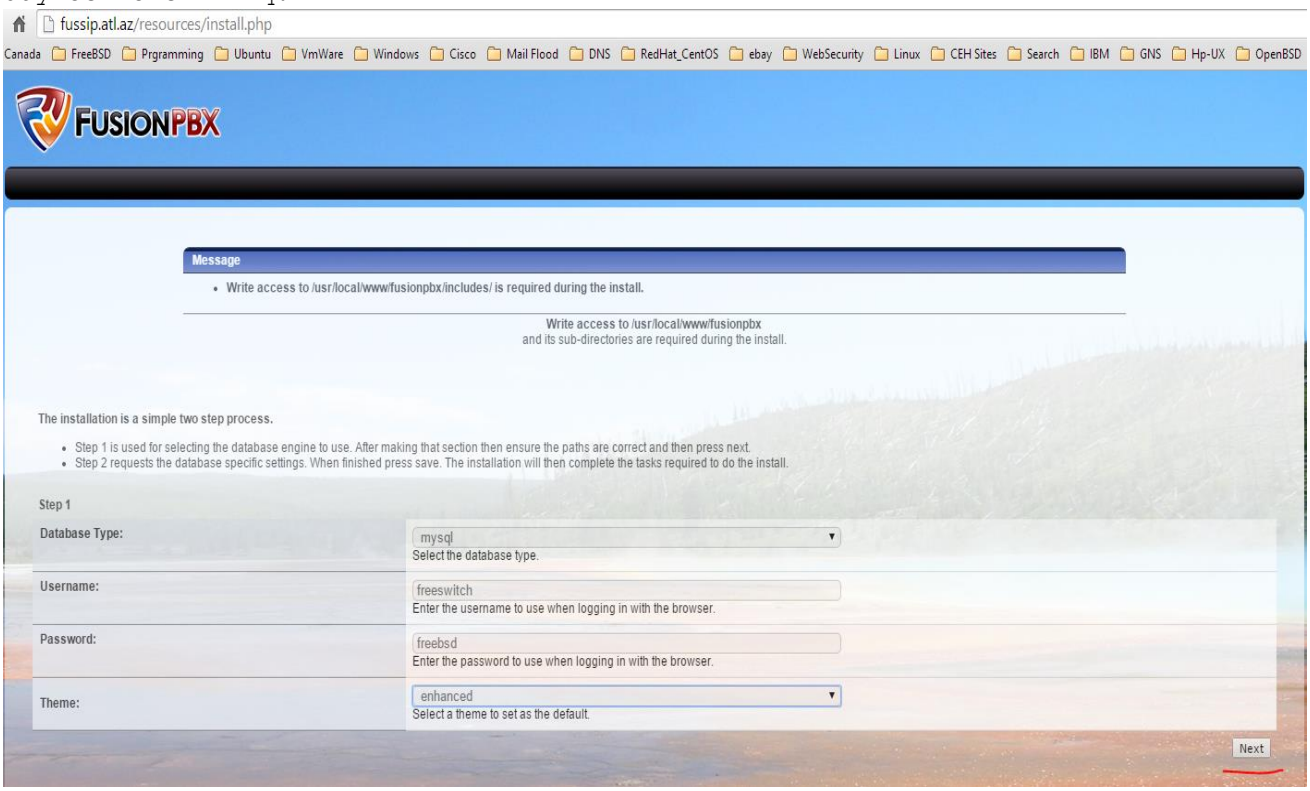
make install # Yükləyirik

Freeswitch-n quraşdırma fayllarına və WEB serverə yetki alması üçün lazımi hüquqları veririk:

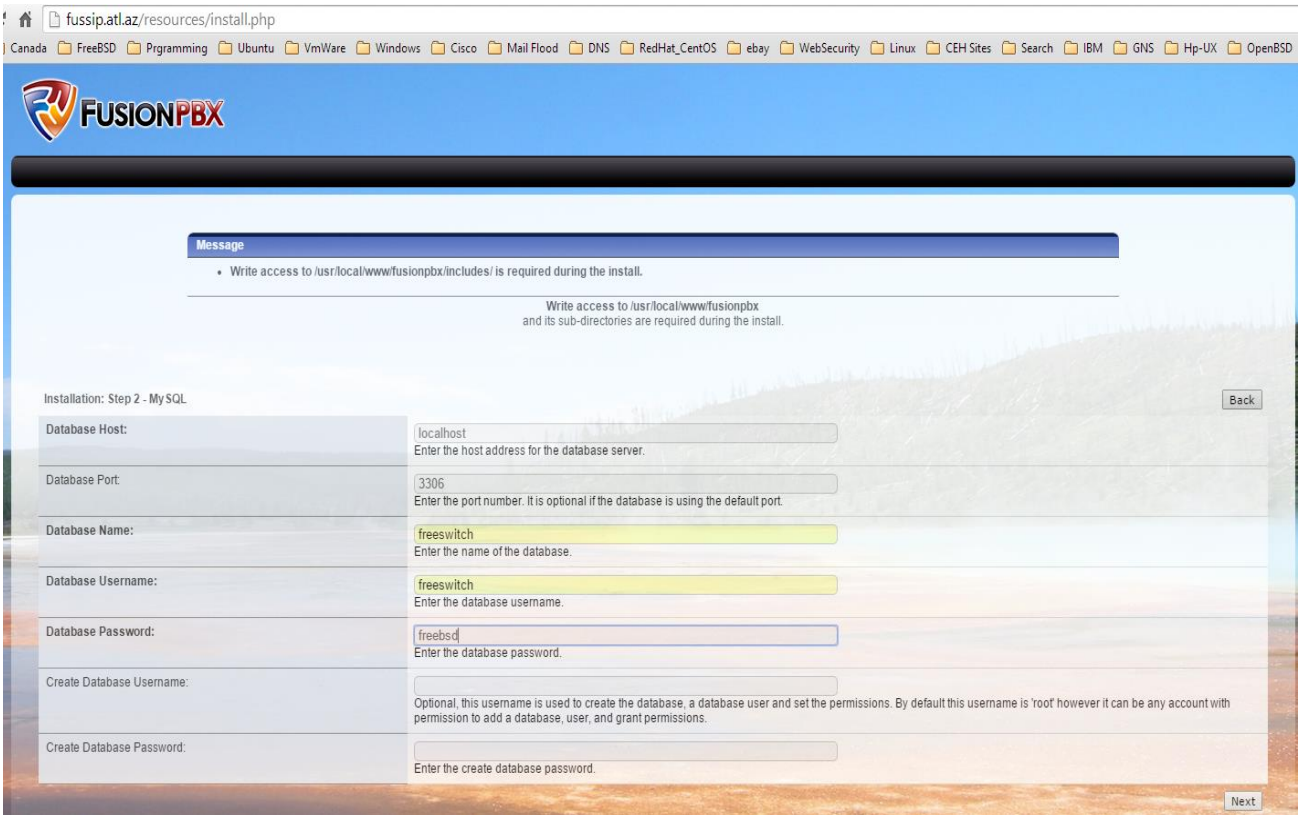
chown -R www:www /usr/local/www/fusionpbx/
chmod -R 770 /usr/local/www/fusionpbx/
chown -R root:www /usr/local/freeswitch/

chmod -R 770 /usr/local/freeswitch/

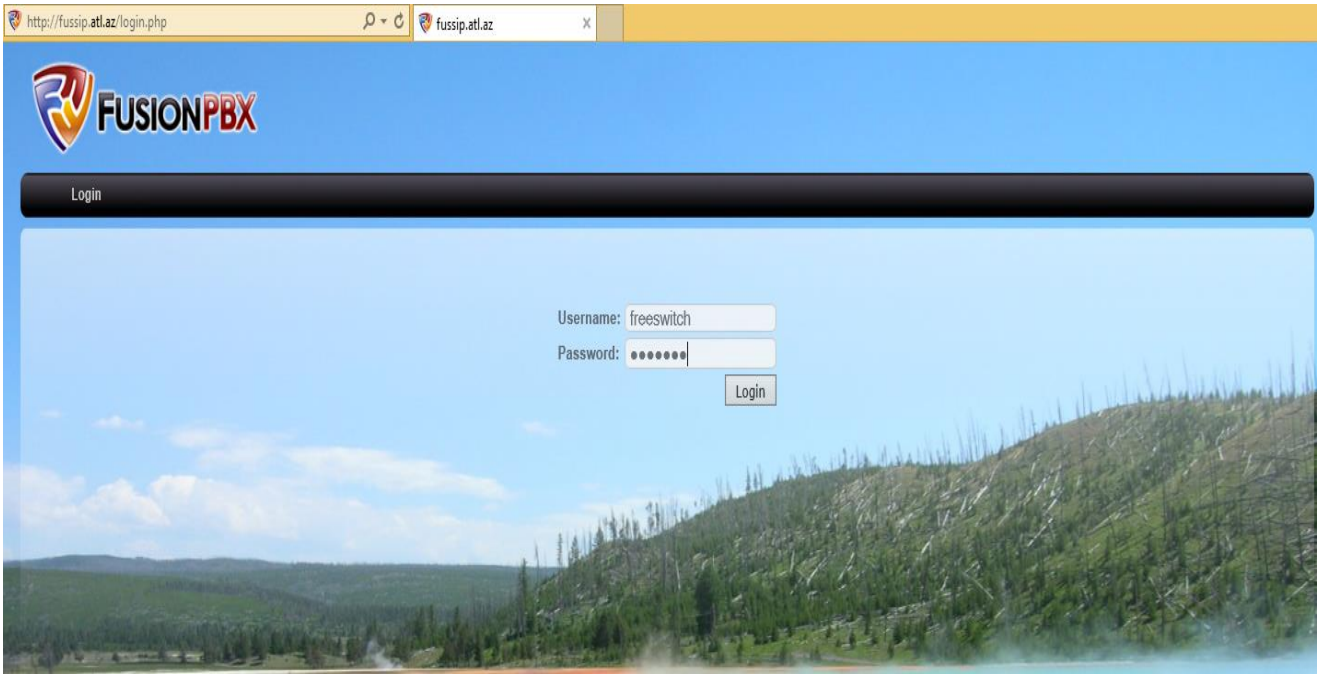
Yüklənmə bitdikdən sonra isə WEB-dən fusionPBX linkinə daxil oluruq. Sənədi yazdığım vaxtda public DNS-lərim mövcud idi və **fussip.domain.az** adında subdomain yaradıb A yazısında PUBLIC IP-mi qeyd etmişdim. Ona görə də **http://fussip.domain.az** ünvanına müraciət edirik və şəkildəki səhifə açılır. WEB Administrator girişi üçün istifadəçi və şifrəsini daxil edib, **Next** düyməsinə sıxırıq:



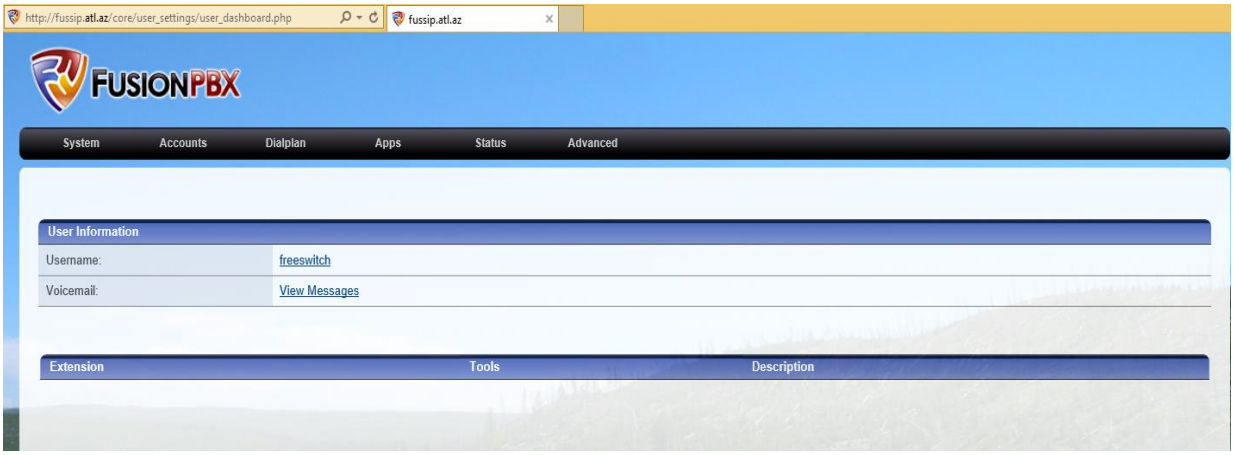
Baza verilənləri, istifadəçi adı və şifrəni daxil edib **Next** düyməsinə sıxırıq:



Sonda login düyməsinə sıxıb, yaratdığımız WEB administrator istifadəçi adı və şifrəni aşağıdaki şəkildəki kimi daxil edirik:



Nəticədə aşağıdaki səhifəni əldə etmiş olmalıyıq:



BÖLÜM 15

Şəbəkə və resurslarnın təhlükəsizliyi

- **FreeBSD Tacacs yüklənməsi və quraşdırılması.**
- **Linux-da Tacacs-ın Domain Controller ilə inteqrasiya edilməsi**
- **SSH Domain controller İnteqrasiyası**
- **Snort İDS**
- **OpenSSL RSA imzalanması və yoxlanılması qaydası**
- **OpenSSL şifrələnmə və deşifrələmə**
- **OpenSSL RSA açarlar və sertifikatlar**
- **OpenSSL imzalama və şifrələmə**
- **OpenSSL OCSP Responder**

Hər bir orta ölçülü və böyük ölçülü kompaniyanın daxilində şəbəkə avadanlıqları mövcud olur. Bunlar switch-lər,router-lər,ASA Firewall və digər şəbəkə səviyyəsində işləyən avadanlıqlar da ola bilər. Bu avadanlıqlar bir neçə şəbəkə inzibatçısı tərəfindən administrasiya edilirsə, onların arasında müəyyən bir konflikt və problem yarana bilər ki, hər kəs dəyişdiyi konfiqurasiya haqqında məlumatlı deyil və digəri bununla razılaşmır. Bu səbəbdən ortaq bir yerə gəlmək üçün tələbə uyğun olan TACACS adlı bir imkan var. Bu başlığımızda TACACS-ın özünün ayrıca qurulması və onun domain controllerlə inteqrasiyasından danışılacaq. Şəbəkə təhlükəsizliyi üçün Snort İDS-dən və şifrələnmə üçün OpenSSL haqqında ətraflı danişacayıq.

FreeBSD Tacacs yüklənməsi və quraşdırılması.

TACACS+ - (Terminal Access Controller Access Control System plus) — seans protokoludur, Cisco-u tərəfindən TACACS-ın təkmilləşdirməsinin nəticəsidir.

Protokolun (şifrələmə) təhlükəsizliyi yaxşılaşdırılmışdır, həmçinin ayrı-ayrılıqda istifadə edilə bilməsi üçün, müəyyənləşdirilmə, avtorizasiya və hesab funksiyaları əlavə edilmişdir.

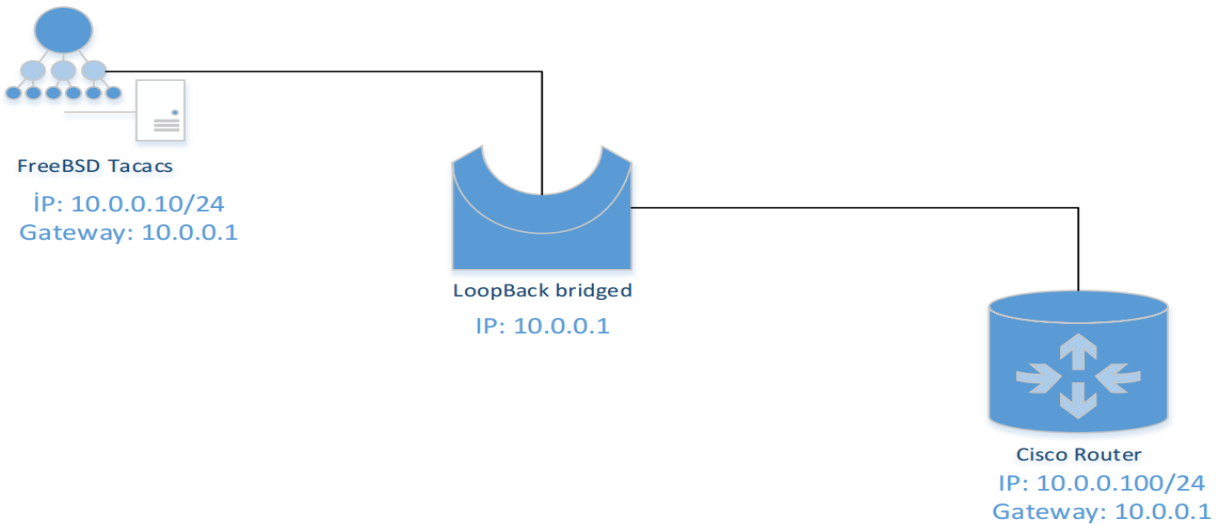
TACACS+ seanslar anlayışlarından istifadə edir. TACACS+ anlayışında **AAA** (authentication, authorization, accounting) seanslarına üç müxtəlif tipin təyin edilməsi mümkündür. Ümumi mənada seansın bir tipinin qurulması hər hansı başqasının ilkin müvəffəqiyyətli qurmasını tələb etmir. Protokolun spesifikasiyası avtorizasiya seansın açılışı üçün, öncədən müəyyənləşdirilmə seansı açmağı tələb etmir. TACACS+ serveri müəyyənləşdirilməni tələb edə bilər, amma protokol bu şərti qoymur.

Istifadə edilən resurslar.

1. FreeBSD_Tacacs x64 (VM, IP: 10.0.0.10)
2. GNS3 (Router 3700, IP: 10.0.0.100)
3. Windows LoopBACK_Adapter (IP: 10.0.0.1)

Struktur aşağıdakı formada olacaq.
FreeBSD x64 10.0.0.10 => LoopBACK Adapter (10.0.0.1)
=> GNS Cloud => Cisco Router 10.0.0.100

Şəbəkə quruluşu aşağıdakı şəkildəki kimi olacaq:



Serverimizin qurulmasına başlayaq

Qeyd: Virtual maşın kimi VmWare-dən istifadə edilmişdir(VirtualBox-la Işləmədi. Şəbəkə karltarında ciddi problemləri var).

Ilk olaraq FreeBSD Əməliyyat sisteminin işlərinə başlayaq.

```
portsnap fetch extract update # LoopBack adapteri qoşmazdan öncə,internetə
                              cıxışımız olmalıdır ki, program portlarını
                              yeniləyək(Sonra reboot mütləq edin.)

cd /usr/ports/net/tac_plus4   # tac_plus4 paketini yükləyirik ki, tacacs-i
                              işlədə bilək.
make install clean

rehash                        # Yüklədikdən sonra binar bazanı yeniləyirik
                              ki, əmrlər rebootsuz sistemdə tanınsın.

ee /etc/rc.conf               # Tacacs-ı Startup-a əlavə edirik ki, yenidən
                              yüklənmədən sonra işləsin.
ifconfig_em0="inet 10.0.0.10  netmask 255.255.255.0"
hostname="tacacs.az"
sshd_enable="YES"
tac_plus_enable="YES"   # Tacacsın startupda işləməsini aktivləşdiririk
tac_plus_flags="-d 8 -d 16 -d 32 -d 64 -C /usr/local/etc/tac_plus.conf"
                  # Lazımı flaglar təyin edirik ki, startup-da özü yoxlasın.
                  '-d' - debug elə deməkdir, qarşısındakı rəqəmlər isə
                  müxtəlif rejimlərdir.
                  8  - authorization debugging
                  16 - authentication debugging
                  32 - şifrə faylının işə düşməsini debug
                  64 - accounting debugging
                  '-C' - '/usr/local/etc/tac_plus.conf' quraşdırma faylı ilə
                              yoxlanış elə.

ee /usr/local/etc/tac_plus.conf     # Quraşdırma faylını aşağıdakı
                                    sintaksislə yazırıq.
# Accounting faylının ünvanını təyin edirik.
accounting file = /var/log/tac_plus.acct

# Cisco avadanlıqla TACACS server arasında istifadə edilən Pre-Shared açar
key = "freebsd"

# Groups
# Qruplar yaradırıq 'admin' və 'service' adında. Tələb olunan yetkiləri də
veririk.

group = admin {
    default service = permit              # Susmaya görə hər şey açıqdır.
    service = exec {                      # Idarəetme səviyyəsi
    priv-lvl = 15                         # 15-ci səviyyədir
    }
}

group = service {
    default service = deny                # Susmaya görə hər şey bağlıdır.
    service = exec {                      # Idarəetme səviyyesi 15-dir
    priv-lvl = 15
    }
}
```

```
# Users
# Istifadəçilər yaradıb lazımı qruplara əlavə edirik, həmdə istifadəçiləri
əmrlərə görə idarə edirik.

user = camal {                  # 'camal' adında istifadəçi yaradırıq və
    member = admin              # 'admin' qrupuna əlavə edirik.
    login = des NQU3rObo2Ntoc # şifrəmizi 'des' alqoritmlə şifrələyirik.
                                (Şifrənin kodlaşdırılması haqqında aşağıda
                                'tac_pwd' əmrinin açıqlanmasında danışacayıq)
}

user = auditor {                # 'auditor' adlı istifadəçi yaradıb,
    member = admin              # 'admin' qrupuna əlavə edirik. Aşağıda
                                sıralanan əmrləri istifadə eləmək yetkisindən
                                məhrum edirik.
    cmd = configure {
        deny .*
        }
    cmd = enable {
        deny .*
        }
    cmd = clear {
        deny .*
        }
    cmd = reload {
        deny .*
        }
    cmd = write {
        deny .*
        }
    cmd = copy {
        deny .*
        }
    cmd = erase {
        deny .*
        }
    cmd = delete {
        deny .*
        }
    cmd = archive {
        deny .*
        }
    login = cleartext secret        # Burda isə 'auditor' istifadəçisinin
                                    şifrəsini açıq'cleartext' şəkildə
                                    yazmışıq.
}
user = event_manager {              # 'event_manager' adlı istifadəçi,
    member = service                # 'service' qrupunun üzvüdür. (qrupda isə
                                    susmaya görə hər şey bağlıdır)
    cmd = clear {                   # Burda isə yalnız sadalanan əmrlərin
                                    istifadəsinə izin veririk.
        permit .*
```

```
        }
    cmd = tclsh {
        permit .*
        }
    cmd = squeeze {
        permit .*
        }
    cmd = event {
        permit .*
        }
    cmd = more {
        permit .*
        }
    cmd = show {
        permit version
        }
    cmd = delete {
        permit .*
        }
    cmd = "delete /force" {
        permit .*
        }
    cmd = "enable" {
        permit .*
        }
    login = des 07xU3lvh1hC3I # Həmçinin burda da şifrəni 'des' alqoritmlə
                              kodlaşdırırıq.
}
```

Qeyd: Şifrələrimizin cleartext yox **'des'** aqloritmi ilə şifrələnmiş formada Görünməsini istəyirsinizsə, onda çox rahat **'tac_pwd'** əmrindən istifadə edin.

```
tac_pwd     # Əmri daxil etdikdən sonra 'ENTER'-i sıxın və lazım olan şifrəni
            daxil edib yenə də 'ENTER'-i sıxın, yeni sətirdə çıxan nəticə isə
            daxil etdiyimiz şifrənin 'des' alqoritmi ilə şifrələnmiş forması
            olur. Həmin kodları nüsxələyib 'login = des'-in qarşısına
            yerləşdiririk ki, şifrəmiz crypt görunsun.

touch /var/log/tac_plus.acct        # tacacs-in accounting faylını yaradırıq
                                    accounting jurnallarını görə bilək.

chown tacacs /var/log/tac_plus.acct # öz istifadəçisi üzvlüyündə təyin edirik

chmod 755 /var/log/tac_plus.acct    # Və yetkiləri veririk.

/usr/local/etc/rc.d/tac_plus start  # servisi işə salırıq

netstat -a | grep tac               # Daemonun qalxmasını test edirik.
tcp4       0      0 *.tacacs                *.*                    LISTEN
```

Indi isə GNS3-də acılmış Routeri config edək.

```
conf t                                  # global rejimə keçirik.
interface fastEthernet 0/0              # GNS3-ün Cloud-na birləşmiş
                                        interfeysi quraşdırırıq.
ip address 10.0.0.100 255.255.255.0 # IP adress təyin edirik.

aaa new-model                                   # AAA modelinə daxil oluruq
tacacs-server host 10.0.0.10 key 0 freebsd      # Və deyirik ki, tacacs
                                                server '10.0.0.10'-dur və
                                                aramızda olan açar 'freebsd'
                                                sözüdür.

tacacs-server timeout 2                 # giriş vaxtının bitməsi 2 saniyədən çox
                                        olmasın

tacacs-server directed-request          # Müraciət birbaşa olsun

aaa group server tacacs+ tac-int        # 'tac-int' adında aaa tacacs+ qrup
                                        yaradırıq

server 10.0.0.10                        # Və '10.0.0.10' adlı tacacs serverin
                                        həmin siyahıya əlavə edirik.
```

Butun aaa-nı **tac-int** admin qrupuna əlavə edirik:

```
aaa authentication login admin group tac-int local
aaa authorization exec admin group tac-int local
aaa authorization commands 15 admin group tac-int local
aaa accounting update newinfo
aaa accounting commands 15 admin start-stop group tac-int
```

Və terminal girişimizdə deyirik ki, AAA-larda admin girişini özümə mənimsəyirəm:

```
line vty 0 4
 authorization commands 15 admin
 authorization exec admin
 accounting commands 15 admin
 login authentication admin
```

Router-i debug eləmək üçünsə aşağıdakı üsüllardan istifadə edə bilərik.

AAA-nı debug elemək üçün:

```
debug aaa per-user
debug aaa authentication
debug aaa authorization
debug aaa accounting
```

TACACS-i debug eləmək üçün aşağıdakı üsulları istifadə edə bilərik:

debug tacacs authentication
debug tacacs authorization
debug tacacs accounting
debug tacacs events
debug tacacs packet

Və sonda öz desktopumuzdan Router-ə **'telnet'** edib yoxlayırıq:

telnet 10.0.0.100

Əgər aşağıdakı formada görüntü çıxsa demək TACACS artıq işləyir:

User Access Verification

Username:

Əgər işləmirsə onda aşağıdakı forma gələcək və bu o deməkdir ki, tacacs serverə çata bilmirik.

Password:

Linux-da Tacacs-ın Domain Controller ilə inteqrasiya edilməsi

Artıq FreeBSD üzərində Tacacs+ haqqında nəzəri olaraq danışmışıq. Tələb yarana bilər ki, şirkətin daxilində Eyni Tacacs+ serveri mütləq şəkildə Domain Controller ilə inteqrasiya eləmək lazımdır. Yəni DC-də şəbəkə inzibatçıları üçün nəzərdə tutulan bir NetAdmins və ya hansısa digər bir qrupunuz var. Bu qrupların hər birində olan üzvlər şəbəkəyə daxil olmaq üçün fərqli yetkilərə sahib olmalıdırlar. Əgər şəbəkə inzibatçısıdırsa tam yetkiyə sahib olmalı və əgər operatordursa limitli yetkiyə sahib olmalıdır. Bu tələblər sizdən edilərsə, məqalə sizin köməyinizə çox yarayacaq.

Istifadə etdiyim resurslar:

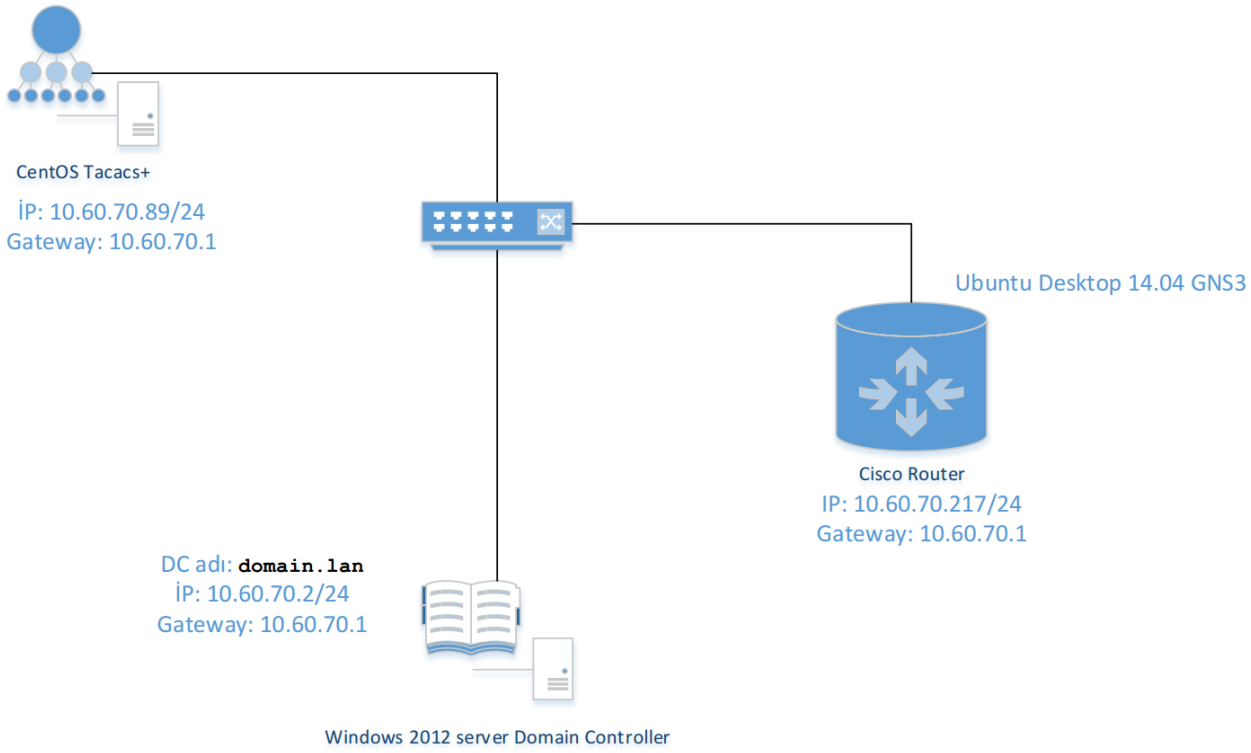
Linux Ubuntu Desktop 14.04 x64 - 10.60.70.217 GNS3 yüklənmiş və qurulmuşdur(Router 3600)

CentOS 6.5 x64(Tacacs+) - 10.60.70.89

Windows 2012 server - DC01-10.60.70.2, DC02-10.60.70.3

DC : **domain.lan**
DC user: **dcadm**
DC pass: **DCAdminPass**
MSLDAP Port: **3268**
DC qrupları: **tacacsadmin, tacacsguest, tacacsmedium**
Istifadəçilər: **full, low, medium** (uyğun olaraq full istifadəçisi **tacacsadmin**, low **tacacsguest** və medium **tacacsmedium** qruplarındadır)

Şəbəkə quruluşu aşağıdakı şəkildəki kimi olacaq:



CentOS maşınımızda reposları yeniləyək və lazımı paketləri yükləyək:

```
history | grep yum | awk '{ $1=""; print }' | grep -v history     # yum
                                                  əmrinin tarixçəsində axtarırıq
yum update
yum -y install gcc
yum -y install perl-LDAP
yum -y install bind-utils
yum -y install telnet.x86_64
yum -y install atop iotop nload iftop htop
yum -y install perl-IO-Socket-SSL
yum -y install pam-devel
yum -y install ld-linux.so.2

cat /etc/resolv.conf          # DC-mizin DNS IP-lərini yazırıq
search domain.lan
nameserver 10.60.70.2
nameserver 10.60.70.3

Lazımı qovluqları öncədən yaradaq:
history | grep mkdir | awk '{ $1=""; print }' | grep -v history   # yum
                                                  əmrinin tarixçəsində
                                                  axtarırıq
mkdir /root/tacacs
mkdir /var/log/tac_plus
mkdir /var/log/tac_plus/access
mkdir /var/log/tac_plus/acct
chmod 760 -R /var/log/tac_plus/

Artıq tacacs paketini dartaq və kompilyasiya edək:
cd /root/tacacs
wget http://www.pro-bono-publico.de/projects/src/DEVEL.201407301604.tar.bz2
tar jxf DEVEL.201407301604.tar.bz2        # Faylı açırıq
cd PROJECTS/                              # Açdığımız qovluğa daxil oluruq
./configure                               # quraşdırırıq
echo $?           # true sıfır olmalıdır
make              # Kompilyasiya edirik
echo $?           # true sıfır olmalıdır
make install      # Yükləyirik
echo $?           # true sıfır olmalıdır

cp /root/tacacs/PROJECTS/tac_plus/extra/tac_plus.cfg-ads
/usr/local/etc/tac_plus.cfg
chmod 755 /etc/init.d/tac_plus
chmod 660 /usr/local/etc/tac_plus.cfg
chkconfig --level 0123456 iptables off
vi /etc/selinux/config        # Faylda aşağıdakı sətiri uyğun olaraq edirik
SELINUX=disabled

chkconfig --add tac_plus                  # Tacacs-i servislərə əlavə edirik
chkconfig --level 2345 tac_plus on        # Tacacs servisini startup-a əlavə
                                          edirik
```

Qeyd: Unutmayın tacacs quraşdırma faylında olan qrupların adında olan **tacacs**

başlığı yazılmır çünki, tacacs bu adla avtomatik özündə axtarış edir və DC-də hər bir halda **tacacs** başlığı ilə özündə olan qrupları axtarış edir. Yəni əgər quraşdırma faylında **guest** və **admin** adlı qruplar olsa, DC-də **tacacsguest** və **tacacsadmin** adlı qruplar yaradılmalıdır.

```
cat /usr/local/etc/tac_plus.cfg           # Quraşdırma faylımızı yoxlayırıq
#!/usr/local/sbin/tac_plus
id = spawnd {
        listen = { port = 49 }
        spawn = {
                instances min = 1
                instances max = 10
        }
        background = yes
}

id = tac_plus {
        access log = /var/log/tac_plus/access/%Y%m%d.log
        accounting log = /var/log/tac_plus/acct/%Y%m%d.log

# MSLDAP-a aid olan quraşdırmalarımız aşağıdakı kimi olacaq:
        mavis module = external {
                setenv LDAP_SERVER_TYPE = "microsoft"
                setenv LDAP_HOSTS = "dc01:3268 dc02:3268"
                setenv LDAP_BASE = "dc=domain,dc=lan"
                setenv LDAP_USER = "dcadm@domain.lan"
                setenv LDAP_PASSWD = "DCAdminPass"
                setenv REQUIRE_TACACS_GROUP_PREFIX = 1
                setenv FLAG_USE_MEMBEROF = 1
                exec = /usr/local/lib/mavis/mavis_tacplus_ldap.pl
        }

        login backend = mavis
        user backend = mavis
#       pap backend = mavis

        host = world {
                address = ::/0
                prompt = "Welcome to FHN Statistika\n"
                  #şifrəmizi bu əmrlə "openssl passwd -1 clear_text_password"
                  şifrələyib generasiya edirik
                enable 15 = crypt $1$8hAByjzi$7tIDLo.9cHJBfW1EQN3N8.
                #enable 15 = clear secret
                key = "t@c@csp@$$w0rd"          # Cisco avadanlıqla Linux
                                                tacacs arasında olan tacacs
                                                açarı
        }
# tacacsadmin qrupun üzvlərinə tam yetki veririk
        group = admin {
                message = "[Admin privileges]"
                default service = permit
                service = shell {
                        default command = permit
```

```
                        default attribute = permit
                        set priv-lvl = 15
                }
        }

# tacasguest qrupunun üzvləri yalnız 1-ci səviyyədə işləyə bilər və enable
edə bilərlər
# ancaq configure və write əmrlərini daxil edə bilməzlər
        group = guest {
                message = "[Guest privileges]"
                default service = permit
                enable = permit
                service = shell {
                        default command = permit
                        default attribute = permit
                        set priv-lvl = 1
                        cmd = configure { deny .*}
                        cmd = write { deny .* }
                }
        }
# tacacsmedium qrupun üzvləri tam yetkiyə malikdir ancaq, configure və enable
əmrləri yığa bilməzlər:
        group = medium {
                message = "[Medium privileges]"
                default service = permit
                service = shell {
                        default command = permit
                        default attribute = permit
                        set priv-lvl = 15
                        cmd = configure { deny .*}
                        cmd = enable { deny .* }
                }
        }
}

ll /usr/local/lib/mavis/mavis_tacplus_ldap.pl    # Faylın uyğun ünvanda
                                                 olmasını yoxlayırıq

/usr/local/sbin/tac_plus -P /usr/local/etc/tac_plus.cfg      # Quraşdırmamızın
                                                             düzgünlüyünü
                                                             yoxlayırıq (hər
                                                             şey qaydasında
                                                             olduqda, heçnə
                                                             çap edilməyəcək)

service tac_plus start              # Servisi işə salırıq(uyğun olaraq stop
                                    və restart yaza bilərik)

netstat -nlp | grep tac_plus        # portun qulaq asmasını yoxlayırıq
tcp    0  0 :::49    :::*    LISTEN      1793/tac_plus

tcpdump -nn port 49                 # 1 console-da porta qulaq asırıq
```

```
tail -f /var/log/tac_plus/access/20140820.log    # Digər consol-da jurnalı
                                                 faylı analiz edirik

tcpdump -n -e -i eth0 port 3268      # 3-cü console-da isə DC-yə gedən
                                     müraciəti analiz edirik
```

Tam debug edib nəticə əldə eləmək üçün isə açağıdakı addımları edə bilərik:

1. Consolumuzun 1-ində aşağıdakı əmri daxil edirik(Mütləq **perl-ldap** modulu yüklənmiş olmalıdır):

```
env LDAP_HOSTS="10.60.70.2" LDAP_SERVER_TYPE="microsoft"
/usr/local/lib/mavis/mavis_tacplus_ldap.pl
```

2. Ikinci console-muzda isə aşağıdakı əmri daxil edirik. **Output attribute-value-pairs**-da **Result - OK** qayıtmalıdır:

```
/usr/local/bin/mavistest /usr/local/etc/tac_plus.cfg tac_plus TACPLUS full
A123456789a

Input attribute-value-pairs:
TYPE                TACPLUS
TIMESTAMP           mavistest-2101-1408505825-0
USER                full
PASSWORD            A123456789a
TACTYPE             AUTH


Output attribute-value-pairs:
TYPE                TACPLUS
TIMESTAMP           mavistest-2101-1408505825-0
USER                full

RESULT              ACK
PASSWORD            A123456789a
SERIAL              uxnEq26iaDtAp12X5kKImA=
DBPASSWORD          A123456789a
TACMEMBER           admin
TACTYPE             AUTH
```

Daemonun debug rejimdə işləməsini yoxlamaq üçün isə aşağıdakı əmrdən istifadə edə bilərsiniz. Ancaq düşünürəm ki, öncəki əmrlər troubleshoot eləmək üçün yetəcək:

```
/usr/local/sbin/tac_plus -d 4088 -fp /var/run/tac_plus.pid
/usr/local/etc/tac_plus.cfg
```

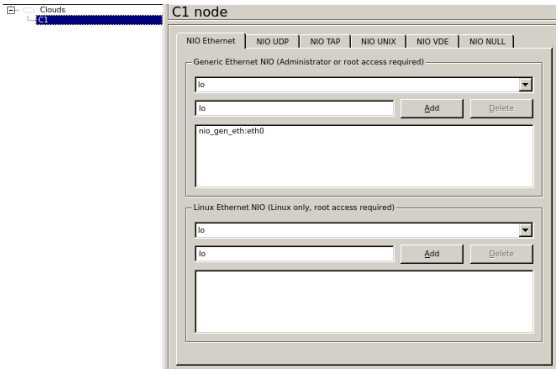
Indi isə gedirik Linux Ubuntu Desktop maşınımızda GNS3-ü yükləyib quraşdıraq ki, 3600 Routerimiz normal işləsin.
Öncə SSH-i açaq və GNS3-ü yükləyək:

```
apt-get update                # sistemi UpToDate edirik
apt-get dist-upgrade
```

```
apt-get install ssh            # SSH-i yükləyək və işə salaq
/etc/init.d/ssh start

apt-get install gns3           # GNS-i yükləyirik
```

Istədiyimiz ünvanda qovluq yaradırıq və 3600 Router-in IOS-nu ora WinSCP ilə Upload edirik. Eynilə GNS3-müzdə 3600 Router-in ünvanını təyin edirik ki, yaratdığımız qovluqdan götürsün. Sonra isə GNS3-ün quraşdırmasını edirik:

Cloud-umuzu aşağıdakı kimi quraşdırırıq:



Sonda Router-ımızı aşağıdakı kimi quraşdırırıq:

```
aaa new-model
aaa group server tacacs+ TACSERVICE
 server 10.60.70.89
aaa authentication login default group TACSERVICE local
aaa authentication login CONSOLE local
aaa authentication enable default group TACSERVICE enable
aaa authorization config-commands
aaa authorization exec default group TACSERVICE local
aaa authorization exec CONSOLE local
aaa authorization commands 15 default group TACSERVICE local
aaa accounting commands 15 default start-stop group TACSERVICE

ip name-server 10.60.70.2
ip name-server 10.60.70.3

interface FastEthernet0/0
```

```
 ip address 10.60.70.217 255.255.255.0
 no shutdown

ip default-gateway 10.60.70.1

tacacs-server host 10.60.70.89
tacacs-server timeout 2
tacacs-server key t@c@csp@$$w0rd          # Tacacs server ilə danışıqda
                                          istifadə edəcəyimiz açar
line con 0
    login authentication CONSOLE
line vty 0 15

do write memory                       # Quraşdırmalarımızı yadda saxlayırıq
```

Ən sonda da hansısa **10.60.70.0/24** şəbəkəsinde olan PC-dən telnet ilə **10.60.70.217** IP ünvanına qoşulmağa çalışırıq:

```
root@squidprimary:~ # telnet 10.60.70.217
Trying 10.60.70.217...
Connected to 10.60.70.217.
Escape character is '^]'.

Welcome to FHN Statistika

Username: low
Password: A123456789a
[Guest privileges]
R1>
```

/var/log/tac_plus/access/20140820.log faylında aşağıdakı sətiri görməliyik.

```
2014-08-20 09:33:02 +0500    10.60.70.217: shell login for 'low' from
10.60.70.50 on tty226 succeeded
```

Router-i debug eləmək üçünsə aşağıdakı əmrlərdən istifadə edə bilərik.

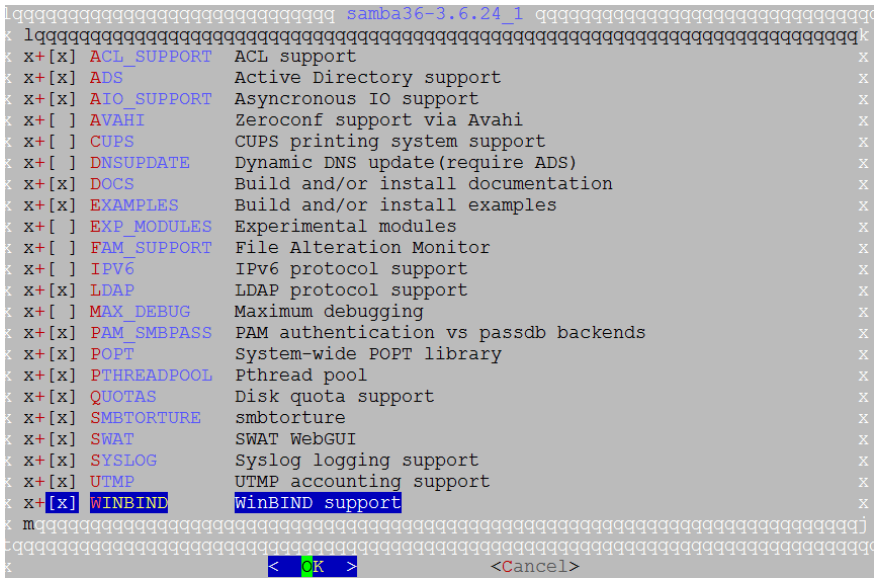
```
# AAA-nı debug eləmək üçün
debug aaa per-user
debug aaa authentication
debug aaa authorization
debug aaa accounting

# TACACS-ı debug eləmək üçün aşağıdakı əmrləri istifadə edə bilərik.
debug tacacs authentication
debug tacacs authorization
debug tacacs accounting
debug tacacs events
debug tacacs packet
```

SSH Domain controller İnteqrasiyası

Məqsədimiz FreeBSD OS üzərində olan SSH serverin istifadəçiləri login və şifrələrini Domain controller-dən almasıdır.

Doman Controller adı: **DOMAIN.LAN**

```
cd /usr/ports/net/samba36            # Öncə Sambani FreeBSD maşına yükləyirik
make config                          # Lazımı modulları seçirik
```




```
make install                         # Yükləyirik
```

SAMBA quraşdırma faylı aşağıdakı kimi olacaq:

```
cat /usr/local/etc/smb.conf
[global]
        workgroup       = DOMAIN
        server string   = FTP Samba
        security        = ADS
        realm           = DOMAIN.LAN
        password server = DOMAIN.lan
        netbios name    = ftp
        load printers = no
        domain master = no
        local master = no
        preferred master = no
        interfaces = em0
        bind interfaces only = yes
        idmap backend = tdb
        idmap uid       = 10000-20000
        idmap gid       = 10000-20000
        idmap config DOMAIN:backend = rid
        idmap config DOMAIN:range = 10000-99999
        winbind separator = +
        winbind enum users = yes
        winbind enum groups = yes
        winbind use default domain = yes
        winbind nested groups = yes
```

```
        winbind refresh tickets = yes
        template homedir = /home/%D/%U
        template shell = /bin/sh
        client use spnego = yes
        client ntlmv2 auth = yes
        encrypt passwords = yes
        restrict anonymous = 2
        log level       = 10
        log file        = /var/log/samba/%m.%U.log
        max log size    = 50000
```

```
mkdir /var/log/samba/               # jurnal qovluğunu yaradırıq
mkdir /var/db/samba                 # Samba baza qovluğunu yaradırıq
mkdir /usr/local/etc/samba/         # Samba qovluğunu yaradırıq
```

/etc/nsswitch.conf faylında **group** və **passwd** atributlarını aşağıdakı şəkilə gətiririk:

```
      group: files winbind
      passwd: files winbind
```

Kernel parametrləri olaraq **/etc/sysctl.conf** faylına aşağıdakı sətirləri əlavə edirik:

```
      security.bsd.see_other_uids=0
      kern.maxfiles=25600
      kern.maxfilesperproc=16384
      net.inet.tcp.sendspace=65536
      net.inet.tcp.recvspace=65536
```

/etc/resolv.conf faylında resolver kimi DC-lərimizin IP ünvanlarını təyin edirik:

```
domain  DOMAIN.lan
nameserver      10.99.9.2
nameserver      10.99.9.3

ntpdate DOMAIN.lan                  # DC-mizdən dəqiq vaxt alırıq
hostname freebsd.DOMAIN.lan         # FreeBSD OS-ə hostname təyin
                                    edirik(/etc/rc.conf-ada əlavə edirik).
```

Kerberos quraşdırma faylı aşağıdakı kimi olacaq:

```
cat /usr/src/crypto/heimdal/krb5.conf
[libdefaults]
        default_realm = DOMAIN.LAN
        clockskew = 300
        v4_instance_resolve = false
        v4_name_convert = {
                host = {
                        rcmd = host
                        ftp = ftp
                }
```

```
                plain = {
                        something = something-else
                }
        }

[realms]
        DOMAIN.LAN = {
                kdc = DOMAIN.LAN
                admin_server = DOMAIN.LAN
                kpasswd_server = DOMAIN.LAN

        }
[domain_realm]
        .DOMAIN.lan = DOMAIN.LAN


root@freebsd:/usr/ports/net/samba36 # kinit jamal      # DC-yə yetkisi olan
                                                       istifadəçi ilə daxil
                                                       oluruq
jamal@DOMAIN.LAN's Password:

root@tstftp:/ # net ads join -U jamal                  # Eyni istifadəçi ilə
                                                       DC-yə daxil oluruq
Enter jamal's password:
Using short domain name -- DOMAIN
Joined 'FTP' to dns domain 'DOMAIN.lan'
```

/etc/rc.conf faylına aşağıdakı sətirləri əlavə edirik ki, Samba və WinBind startup-da işə düşsün.

```
samba_enable="YES"
winbindd_enable="YES"
kerberos5_server_enable="YES"
kadmind5_server_enable="YES"

/usr/local/etc/rc.d/samba start                        # Sambanı işə salırıq

wbinfo -u                       # DC istifadəçiləri siyahılayırıq
wbinfo -g                       # DC qrupları siyahılayırıq
getent passwd                   # FreeBSD UID-ləri siyahılayırıq
getent group                    # FreeBSD GID-ləri siyahılayırıq
```

Indi isə SSH-in inteqrasiyası ilə məşğul olaq.

PAM-la authentifikasiya olduqda, SSH istifadəçi üçün avtomatik qovluğun yaradılmasını istəyiriksə, aşağıdakı portu yükləyirik.

```
cd /usr/ports/security/pam_mkhomedir/     # Port ünvanına daxil oluruq
make install                              # Yükləyirik

mkdir /home/DOMAIN/             # Domain istifadəçiləri üçün qovluq yaradırıq.
```

/etc/pam.d/sshd adlı fayl yaradıb içinə aşağıdakı məzmunu əlavə edirik:

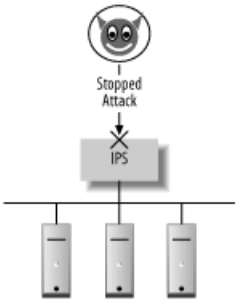
```
# auth
auth       sufficient      pam_opie.so             no_warn no_fake_prompts
auth       requisite       pam_opieaccess.so       no_warn allow_local
# DC-də olan hər kəsə izin veririk
auth            sufficient      /usr/local/lib/pam_winbind.so
#auth      sufficient      pam_krb5.so             no_warn try_first_pass
#auth      sufficient      pam_ssh.so              no_warn try_first_pass
auth       required        pam_unix.so             no_warn try_first_pass

# account
account    required        pam_nologin.so
#account   required        pam_krb5.so
account    required        pam_login_access.so
account    required        pam_unix.so

# session
#session   optional        pam_ssh.so
# AD-dən qeydiyyatdan keçmiş istifadəçilər üçün ev qovluğu yaradır
session    required        /usr/local/lib/pam_mkhomedir.so
session    required        pam_permit.so
# password
#password sufficient       pam_krb5.so             no_warn try_first_pass
password   required        pam_unix.so             no_warn try_first_pass
```

Sonda test üçün reboot edirik və DC istifadəçi adı ilə daxil olmağa çalışırıq:

```
reboot
```

Snort İDS

IDS passiv sistemdir. Ancaq onun analizi ilə insanlar məşğul olmalıdır. Bir müddət sonra isə hücumun qarşısını almaq üçün Intrusion Prevention System yaradılmışdır. IPS isə IDS-in aktiv versiyasıdır. Ona görə ki, IDS yalnız məlumat verirdi, IPS isə həmdə pis trafiki bloklamaq qabiliyyətinə malikdir. IDS-də olduğu kimi məntiqi quruluş IPS-də də eynidir. Ancaq IPS-in funksionallığı Firewall kimi daxili şəbəkəyə olan yetkini idarə edir. Aşağıdakı şəkildə IPS-in necə hücumun qarşısını aldığını göstərir.



Təhlükə ondan ibarətdir ki, IPS çoxlu düzgün trafiki belə bağlaya bilər. Xatırlayırsınızsa IDS-də belə olan hallarda o səhv sala bilərdi. Yalnız IDS səhv salsada bu barədə ancaq məlumat verirdi. IPS-də isə o trafiki bütövlüklə bağlayır.

Qeyd: Unutmayın ki, əgər hər bil halda IDS yalançı virus məlumatları ötürsə Də belə, heç vaxt buna boş şey kimi baxmayın.

Ancaq IPS-də siz buna boş şey kimi baxıb inamlı trafik kimi qeydə alsaz, Xaker bunu öz xeyrinə istifadə edə bilər. Önəmli baza sayt və ya məktublarla belə ötəri yanaşmalar uçuruma gətirib çıxara bilər.

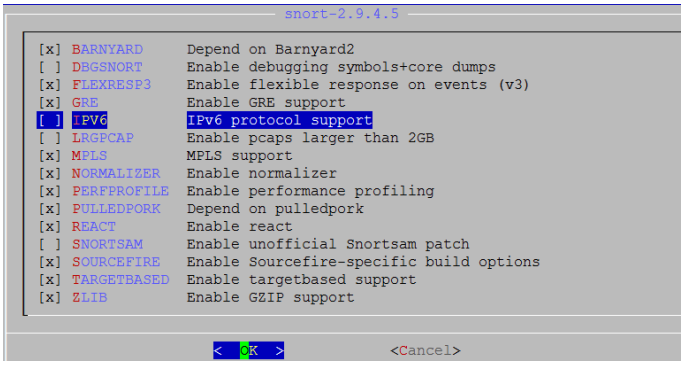
SNORT-un yüklənməsi və quraşdırılması

Ilk növbədə NIDS yüklənən serverin iki şəbəkə kartı olmalıdır. Bir tərəf şəbəkəyə qulaq asmaq üçün, digər tərəf isə management üçün. Serverin resurslarını isə istəklərə uyğun təyin etməlisiniz. Əgər trafik çox olarsa təbii ki, resurs çox olmalıdır.

Snort FreeBSD əməliyyat sistemində birbaşa portlardan yada rəsmi saytından qaynaq kodlarından kompilyasiya edilə bilər http://snort.org/snort-downloads . Snort-un yüklənməsi və quraşdırılması uzun müddət ala bilər. Ancaq əsas məqamlardan biri isə onun hücumlar haqqında fayllarda saxladığı informasiyanın formatıdır. Adi halda o flat fayllarda saxlayır. Həmçinin SNORT-un imkanı vardır ki, **MySQL**-də və ya **MsSQL**-də saxlasın. Əgər MySQL-də istəyirsinizsə onda source code-u **--with-mysql** ilə kompilyasiya eləməlisiniz. IDS-dən sistemə gələn jurnallar həddən artıq böyük olur. Hətta haker özü belə yalançı trafik yollaya bilər ki, IDS-də lazımsız jurnallar şişib onun işini dayandırsın. Ona görə də snort üçün mütləq əlavə qovluq yaradın və xüsusi həcm verin ki, jurnallar ora yığılsın(Məsələn: **/var/snort**).

FreeBSD OS üzərində SNORT-u portlardan yükləyə bilərsiniz. Ancaq öncədən bildirim ki, portları mütləq yeniləyin.

```
cd `whereis snort | awk '{ print $2 }'`   # Snort-u portlardan yükləyirik.
make config                               # lazımı modullarını seçirik.
```




```
make install clean                        # Yükləyək. Ancaq depends-lərdə
                                          barnyard2 gəldikdə MySQL-i mütləq
                                          seçin

# Sistemə Snort işləməsi üçün snort adlı istifadəçi əlavə edirik.(şifrəsiz və
nologin shell ilə)
Username    : snort
Password    : <disabled>
Full Name   : Snort User
Uid         : 1003
Class       :
Groups      : snort
Home        : /home/snort
Home Mode   :
Shell       : /usr/sbin/nologin
Locked      : no


cd /usr/local/etc/snort/rules/                  # Bu ünvana SNORT-un
                                                saytından endirdiyimiz rule-
                                                ları yükləyirik. Hal-hazırki
                                                snortrules-snapshot-
                                                2940.tar.gz

tar -zxf snortrules-snapshot-2940.tar.gz  # Həmin qovluqda rule-ları açırıq.
rm snortrules-snapshot-2940.tar.gz        # Sonra da rule-ları silirik.

echo 'snort_enable="YES"' >> /etc/rc.conf # SNORT servisini Startup-a əlavə
                                          edirik.
echo 'snort_interface="em0"' >> /etc/rc.conf
echo 'snort_conf=" /usr/local/etc/snort/snort.conf"' >> /etc/rc.conf
echo 'snort_group="snort"' >> /etc/rc.conf
echo 'snort_flags="-D -q"' >> /etc/rc.conf


# '/usr/local/etc/snort/snort.conf' faylında WHITE və BLACK list
konfiglərinin ünvanını təyin edirik.
var WHITE_LIST_PATH ./rules/rules
var BLACK_LIST_PATH ./rules/rules
```

```
# Eynilə '/usr/local/etc/snort/snort.conf' faylında adi rule-lar, so-rule-lar
və preproc-rule-llar üçün unvanı
# redaktə edib düzəldirik. Unutmayın ki, error jurnallar '/var/log/messages'
ünvanına yığılır.
var RULE_PATH ./rules/rules/
var SO_RULE_PATH ./rules/so_rules
var PREPROC_RULE_PATH ./rules/preproc_rules

whitelist $WHITE_LIST_PATH/whitelist.rules, \    # white_list.rules faylının
                                                 adını dəyişib whitelist.rules
                                                 edirik
blacklist $BLACK_LIST_PATH/blacklist.rules       # BLACKlist faylının adını
                                                 black_list.rules-dan dəyişib
                                                 blacklist.rules edirik.
                                                 snort.conf faylını yadda
                                                 saxlayıb, çıxırıq.

touch /usr/local/etc/snort/rules/rules/whitelist.rules      # whitelist rule
                                                            faylı yaradırıq
                                                            ki, snort
                                                            deyinməsin

touch /usr/local/etc/snort/rules/rules/blacklist.rules      # blacklist rule
                                                            faylı yaradırıq
                                                            ki, snort
                                                            deyinməsin

# BARNYARD2-ni quraşdırdıqda bizə 'sid-msg.map' faylına ehtiyac olacaq. Ona
görə də onu öncədən
# '/usr/local/etc/snort' qovluğuna nüsxələyirik.
cp /usr/local/etc/snort/rules/etc/sid-msg.map /usr/local/etc/snort

echo hw.usb.no_pf=1 >>/boot/loader.conf    # USBUS interfeysi söndürürük,
                                           çünki SNORT şəbəkəni sniff edəndə
                                           ilk olaraq usbus0 alətinə müraciət
                                           edəcək və səhv çap edəcək. Mütləq
                                           sonra reboot edin.

netstat -i              # Bu əmrlə reboot-dan sonra usbus0 alətinin sönülü
                        olduğunu görə bilərsiniz.

chown -R snort:snort /usr/local/etc/snort       # Snort qovluğunun istifadəçi
                                                və qrupunu snort-a
                                                mənimsədirik.
```

SNORT-un içində həddən artıq vacib quraşdırma faylları mövcuddur. Əsas **snort.conf** faylında digər quraşdırma fayllarına çağırışlar və şəbəkə çıxışlarının quraşdırmaları mövcuddur. Local tərəfin şəbəkələri **HOME_NET** dəyişən adı ilə Public tərəfin şəbəkələri isə **EXTERNAL_NET** dəyişən adı ilə elan edilir. Bunun sayəsində SNORT təyin edə bilir ki, trafik daxildən və ya PUBLIC-dən gəlir. Susmaya görə aşağıdakı sintaksisdə göstərildiyi kimi, **PUBLIC**-də də **LOCAL**-da da **any** yerləşdirilir.

```
var HOME_NET any
var EXTERNAL_NET any
```

Əgər sizin daxili şəbəkəniz **192.168.0.0/24**-dən və **172.16.0.0/24**-dən ibarətdirsə onda **HOME_NET** sintaksisi aşağıdakı kimi olmalıdır. **EXTERNAL_NET** isə **any** qalsada olar. Hər bir halda unutmayın ki, HOME_NET-i təyin etməsəz siz SNORT servisini start edə bilməyəcəksiniz.

```
var HOME_NET [192.168.0.0/24,172.16.0.0/24]
/usr/local/etc/rc.d/snort start     # Sonda da SNORT servisini işə salırıq.
```

threshold.conf – Bu quraşdırma faylı IDS-in məhdudiyyətlərini idarə eləmək üçün istifadə edilir. Yəni əgər siz istəsəz ki, müəyyən trafiklərin haqqında sizə məlumat gəlməsin və ya məlumatları sayca məhdudlaşdırmaq istəsəz, onda siz bu konfiq faylına müraciət etməlisiniz.

Snort **signature** bazalı IDS sistemdir. Bu o deməkdir ki, hər bir gələn paketi özündə olan rule-larla müqayisə edib yoxlayır ki, görək paket pis niyyətlidir ya yox. SNORT-un rule-ları hər gün yenilənir. Ona görə də siz onların statusunu həmişə yeniləməlisiniz. Ancaq təəssuf ki, bu pulludur(ayı 30$). Göstərilən linkdən ən yeni rule-ları əldə edə bilərsiniz.
http://www.snort.org/snort-rules/

Hər bir halda yenədə əgər siz saytda qeydiyyatdan keçmiş olsanız sizə müəyyən məhdudiyyəti olmuş rule-ları endirmək üçün izin verəcəklər. Ancaq endirim arasında 15 dəqiqə limit var.

Siz əldə elədiyiniz yeni rule-ları **'/usr/local/etc/snort/rules'** qovluğunda yerləşdirməlisiniz.

Event-lərin flat fayllarda saxlanılması

Susmaya görə SNORT bütün çıxan xəbərdarlıqları daxili fayl sistemdə **'/var/log/snort'** ünvanında saxlayır. SNORT yeganə **alert** adlı jurnal faylından ibarətdir hansı ki, SNORT rule-ları ilə üst-üstə düşən trafik haqqında məlumatı bu faylda jurnallanır. Siz bu fayla **tail -f** əmri ilə online baxa bilərsiniz. Misal üçün IIS serverin üstünə gələn çoxlu trafikin eventini göstərək.

```
# Sətir hücumun tipini təyin edir.
[**] [119:2:1] (http_inspect) DOUBLE DECODING ATTACK [**]

# WEB hücum olduğu təyin edilir və priority böyükdür
1 rəqəmi hücumun uğurlu olduğu haqda host-la kompramisə getməsi haqqında
məlumat verir.
[Classification: Web Application Attack] [Priority: 1]

# IP mənbəsi, mənsəbi və vaxtını göstərir
11/01-20:29:19.163907 192.168.0.99:52571 -> 192.168.0.10:80

# Bu sətirlər isə tam aşağı səviyyə paketin gedişatı haqqında danışır.
```

```
TCP TTL:64 TOS:0x0 ID:5115 IpLen:20 DgmLen:212 DF
***AP*** Seq: 0x71850B78  Ack: 0xCBB1AFB1  Win: 0xFFFF  TcpLen: 32
TCP Options (3) => NOP NOP TS: 549495890 43275571
```

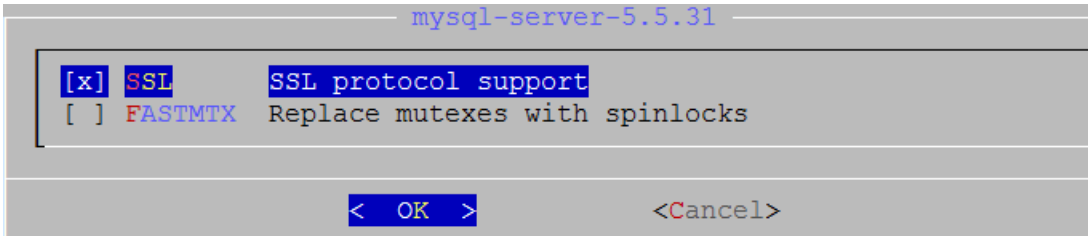
SNORT hər bir host-dan qəbul elədiyi alert üçün ayrıca bir qovluq yaradır və həmin qovluğun daxilində də gələn hər bir source port üçün ayrı-ayrı fayllar yaradır.

```
ls -al /var/log/snort        # Alert göndərən hostlar üçün yaradılan qovluqlar
drwx------  2 snort  snort    512 Nov  1 20:54 10.0.0.1
drwx------  2 snort  snort    512 Nov  1 20:54 192.168.0.56
drwx------  2 snort  snort    512 Nov  1 20:54 192.168.0.99
-rw-------  1 snort  snort  70646 Nov  1 20:55 alert

ls -al /var/log/snort/192.168.0.99/ # Seçilmiş host-un dinamik portlarına
                                    görə olan hər müraciətə bir fayl.
-rw-------  1 snort  snort  1044 Nov  1 02:16 TCP:49455-80
-rw-------  1 snort  snort  1044 Nov  1 02:16 TCP:49536-80
-rw-------  1 snort  snort  1041 Nov  1 20:54 TCP:52571-80
-rw-------  1 snort  snort  1041 Nov  1 20:54 TCP:52600-80
-rw-------  1 snort  snort  1038 Nov  1 20:54 TCP:52601-80
-rw-------  1 snort  snort  1041 Nov  1 20:54 TCP:52610-80
```

Eventlərin MySQL-də saxlanılması.

Eventlərin **MySQL**-də saxlanılması üçün biz **SNORT**-u **Barnyard2** ilə əlaqələndirməliyik. Bunun üçün isə öncə **MySQL**-i sonra da **Barnyard2** paketini sistemə yükləməliyik. Həmçinin unutmayın ki, **barnyard2** üçün errorlar **'/var/log/messages'** unvanında tapılır.

```
cd /usr/ports/databases/mysql55-server     # Port unvanına daxil oluruq.
make config                                # Şəkildəki asılılıqları seçirik.
```

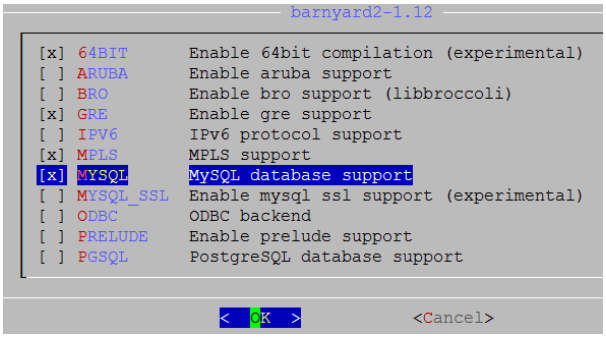



```
make install clean                         # Yükləyirik.

echo 'mysql_enable="YES"' >> /etc/rc.conf      # MySQL servisini Startup-a
                                               əlavə edilir.

/usr/local/etc/rc.d/mysql-server start         # Servisi işə salırıq.

/usr/local/bin/mysql_secure_installation       # Aşağıdakı suallara cavab
                                               verərək susmaya görə
                                               quraşdırırıq.

Set root password? [Y/n] Y
New password:
```

```
Re-enter new password:
Remove anonymous users? [Y/n] Y
Disallow root login remotely? [Y/n] Y
Remove test database and access to it? [Y/n] Y
Reload privilege tables now? [Y/n] Y

mysql -u root -pfreebsd        # root istifadəçi və yaratdığımız şifrə ilə
                               MySQL-ə daxil oluruq.

CREATE DATABASE snort;         # SNORT bazasını yaradırıq.

# snort adlı bazaya istənilən table-a localhost-dan snort istifadəçi adı
freebsd şifrəsi ilə qoşulmağa izin veririk
GRANT ALL PRIVILEGES ON snort.* TO 'snort'@'localhost' IDENTIFIED BY
'freebsd';
```

Indi isə Barnyard2-ni yükləyək

```
cd /usr/ports/security/barnyard2     # Port ünvanına daxil oluruq.
make config                          # Lazımı asılılıqları seçirik.
```




```
make install clean                   # Yükləyirik.

## Sistemə 'barny' adla UID və GID-i 999 olan istifadəçi əlavə edək.
Aşağıdakı göstəricilərlə
Username    : barny
Password    : <disabled>
Full Name   : Barnyard2 User
Uid         : 999
Groups      : barny
Home        : /home/barny
Shell       : /usr/sbin/nologin

# CLI-dan əmri daxil edib barnyard SQL strukturunu yaradırıq.
mysql -u snort -pfreebsd snort <
/usr/local/share/examples/barnyard2/create_mysql

ee /usr/local/etc/barnyard2.conf     # Barnyard-ı quraşdıraq.
config utc                           # Sistem vaxtımızı UTC elan edirik
config reference_file:      /usr/local/etc/snort/reference.config
```

```
config classification_file: /usr/local/etc/snort/classification.config
config gen_file:            /usr/local/etc/snort/gen-msg.map
config sid_file:            /usr/local/etc/snort/sid-msg.map # Bu faylı
                                                             öncə nüsxələməli
                                                             idiniz.

config event_cache_size: 4096       # Cache-mizin həcmini böyüdürük
config logdir: /var/log/barnyard2   # Jurnal ünvanı olaraq
                                    ‘/var/log/barnyard2’ təyin edirik.

#output alert_fast: stdout    # Sətirin əvəzinə
output alert_fast            # Sətiri yazırıq.

#Hostname və hansı şəbəkəyə qulaq asdığını təyin edirik.
config hostname:   ssh-agent2
config interface:  em0
config daemon     # Konfiq tipinin Daemon kimi işləyəcəyini elan edirik.

config set_gid: 999           # Hansı qrup adından işləyəcəyini deyirik
config set_uid: 999           # Hansı istifadəçi adından işləyəcəyini deyirik
config waldo_file: /var/log/snort/barnyard2.waldo     # WALDO faylının
                                                      ünvanını göstəririk

input unified2
output alert_fast
      output log_tcpdump: tcpdump.log     # tcpdump jurnalı aktiv edirik.

# Yaratdığımız SNORT bazası üçün quraşdırmamızı edək. Və faylı yadda saxlayıb çıxaq.
output database: log, mysql, user=snort password=freebsd dbname=snort
host=localhost

mkdir /var/log/barnyard2      # Barnyard2 üçün jurnal qovluğu yaradaq.
touch /var/log/snort/barnyard2.waldo      # faylı yaradırıq.

## Snort və barny yetkilərini hər iki jurnal üçün təyin edirik.
chown -R barny:snort /var/log/barnyard2/
chmod -R 770 /var/log/barnyard2/
chown -R barny:snort /var/log/snort
chmod -R 770 /var/log/snort

## Barnyard servisini Startup-a əlavə edirik.
echo 'barnyard2_enable="YES"' >> /etc/rc.conf
echo ‘barnyard2_flags="-d /var/log/snort -f snortunified2.log -w
/var/log/snort/barnyard2.waldo -D"’ >> /etc/rc.conf
echo ‘barnyard2_conf="/usr/local/etc/barnyard2.conf"’ >> /etc/rc.conf


/usr/local/etc/rc.d/barnyard2 start # Servisi işə salırıq.

ps -ax | grep barn                  # Prosesslərdə olduğunu yoxlayırıq.
1235 ??  Ss    0:25.44 /usr/local/bin/barnyard2 -d /var/log/snort -f
snortunified2.log -w /var/log/snort/barnyard2.waldo -D -c
/usr/local/etc/barnyard2.conf -D
```

SNORT işləyir PF ilə

PF FireWall-nın paketləri xüsusi **pflog0** log interfesyinə yönləndirmə imkanı mövcuddur. pflog0-a göndərilən paketlər pcap formatındadır və ona görə də pcap proqramı tərəfindən oxuna bilər. SNORT-da həmçinin öz növbəsində bu **pflog0** interfeysində qulaq asa bilir. Əgər siz bütün trafiki bağlayaraq jurnallasanız. Onda SNORT bu bağlı trafikdən belə hücumu təyin eləmə imkanına malikdir. Trafiki bağlayaraq jurnallamaq üçün **'/etc/pf.conf'** faylına aşağıdakı sətiri əlavə etməniz yetər.

```
block in log all
```

Ancaq onu bilinki SNORT hər scan görən kimi onu hücum kimi qələmə verəcək. Yada ki, misal üçün **Unicode** tipli hücum **Microsoft IIS** web serverə gedirsə. Bu halda o ilk qoşulma üçün **TCP** sessiyanı **HTTP** müraciətlə açmalıdır. Ancaq əgər bizim firewall **SYN** paketlərini blocklayırsa və onu **pflog0** alətində loglayırsa, onda hücum edən şəxsin HTTP müraciət yollamağa heç vaxt şansı olmayacaq.

Digər üsulla ilə isə siz sadəcə bütün trafiki hər yerə açıb loglaya bilərsiniz. Bu üsul daha ağıllı olar ona görə ki, IDS sistem bütün informasiyanı görüb analiz eləmək qabiliyyətinə malik olacaqdır. Sadəcə **'/etc/pf.conf'** faylına aşağıdakı sətiri əlavə eləməniz yetər.

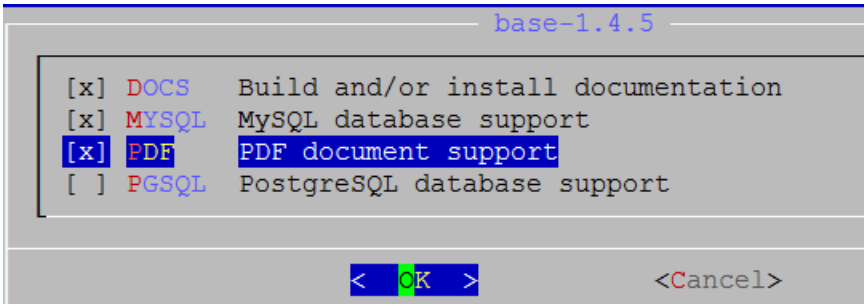
```
pass in log all from any to any keep state
```

BASE

SNORT xəbərdarlıqlarından baş çıxarmaq əməlli başdı çətin məsələdi. Ancaq bu logların analizi üçün kifayət qədər utilitlər vardır. Bunlardan ən məhşurlarından biri **Basic Analysis and Secure Engine(BASE)**-dir. PHP bazalıdır. ACID əsaslarında qurulmuşdur.

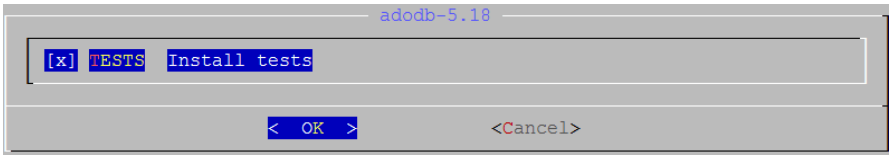
BASE-in yüklənməsi

```
cd /usr/ports/security/base          # Port unvanına daxil oluruq.
make config                          # Lazımı modulları seçirik.
```




```
make install clean                   # Yükləyirik.
```

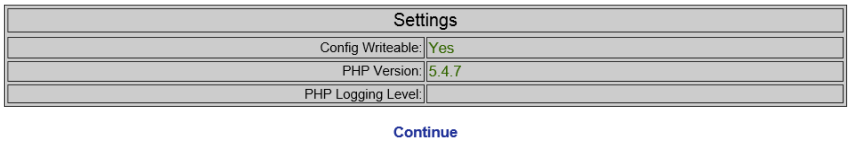
Eynilə modullarda **adodb** testlərinidə seçirik



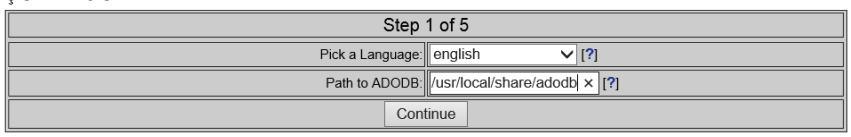
Unutmayın BASE işləməsi üçün sistemə portlardan **'/usr/ports/www/apache22'** və **'/usr/ports/lang/php5'** yüklənməlidir. **BASE** öz **PHP** scriptlərində sistemin **TimeZone**-na baxdığı üçün mülteq bu problemi öncədən həll etməliyik, əks halda siz **WEB**-də **PHP** time errorları görəcəksiniz.

```
cp /usr/share/zoneinfo/Asia/Baku /etc/localtime # Sistem vaxtı AZST edirik.

# PHP-nin quraşdırma faylını yaradaq.
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini

# '/usr/local/etc/php.ini' faylında aşağıdakı sətirləri tapıb güstərilən
formada redaktə edirik.
date.timezone = 'Asia/Baku'
error_reporting = E_ALL & ~E_NOTICE

/usr/local/etc/rc.d/apache22 restart  # Sonda isə apache22-yə restart edirik.

Base işləməsi üçün VirtualHost yaradaq.

mkdir /usr/local/domen          # Yeni VirtualHost üçün ünvan yaradaq.

# Apache-də həmin virtualHost-u aktiv edək.
echo 'Include /usr/local/domen/*' >> /usr/local/etc/apache22/httpd.conf

# Yeni VirtualHost faylı yaradırıq və içinə aşağıdakı məzmunu əlavə edirik.
ee /usr/local/domen/snort.az          # VirtualHost faylı
<VirtualHost *>
   ServerName snort.az
   ServerAlias www.snort.az
   DocumentRoot "/usr/local/www/base"
<Directory "/usr/local/www/base">   # Yüklədiyimiz BASE-in unvanı.
   Options All
   Options FollowSymLinks
   AllowOverride AuthConfig
   Order allow,deny
   Allow from all
</Directory>
</VirtualHost>

chown -R www:www /usr/local/www/base/      # BASE qovluğunu www üzvlüyü edirik
                                           ki, Apache işləsin.
```

Və WEB ilə linkimizə daxil oluruq. http://snort.az/ aşağıdakı şəkil çıxacaq. **Continue** düyməsini sıxırıq.



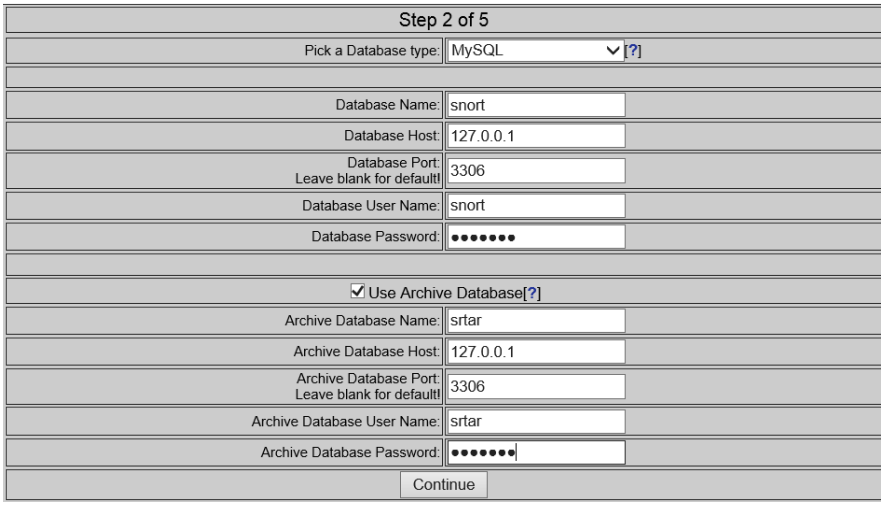
Və **English** seçərək **ADODB** ünvanı təyin edib **continue** düyməsini sıxırıq. Şəkildəki kimi



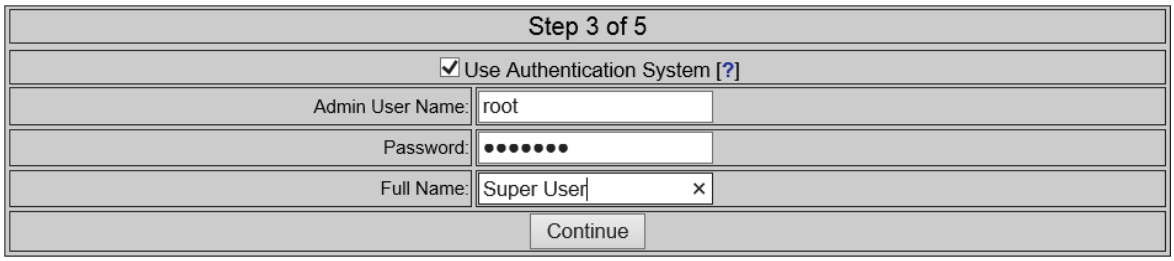
Snort üçün yaratdığımız bazanın ünvanını, host-dan girişini, istifadəçi adı və şifrəsini təyin edirik. Əgər siz arxiv bazası istifadə eləmək istəyirsinizsə onda siz öncədən onu yaradıb, snort ilə eyni olan **table** strukturunu əlavə eləməlisiniz.

```
# Arxiv üçün bazanı yaradaq.
mysql -u root -pfreebsd -e 'CREATE DATABASE srtar;'

# Yaratdığımız srtar bazasına eyni adlı istifadəçiyə localhost-dan freebsd şifrəsi ilə qoşulmaya izin veririk.
mysql -u root -pfreebsd -e "GRANT ALL PRIVILEGES ON srtar.* TO 'srtar'@'localhost' IDENTIFIED BY 'freebsd';"

# Və eyni baza strukturunu srtar bazası üçün yaradırıq ki, arxiv logları işləsin.
mysql -u srtar -pfreebsd srtar < /usr/local/share/examples/barnyard2/create_mysql
```

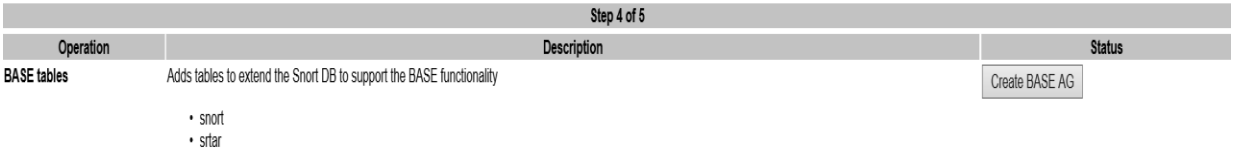
Baza ilə işimizi bitirdikdən sonra qayıdırıq **WEB** ilə baza quraşdırmalarımızı yeridib **Continue** düyməsini sıxaq. Şəkildə göstərilən qaydada.



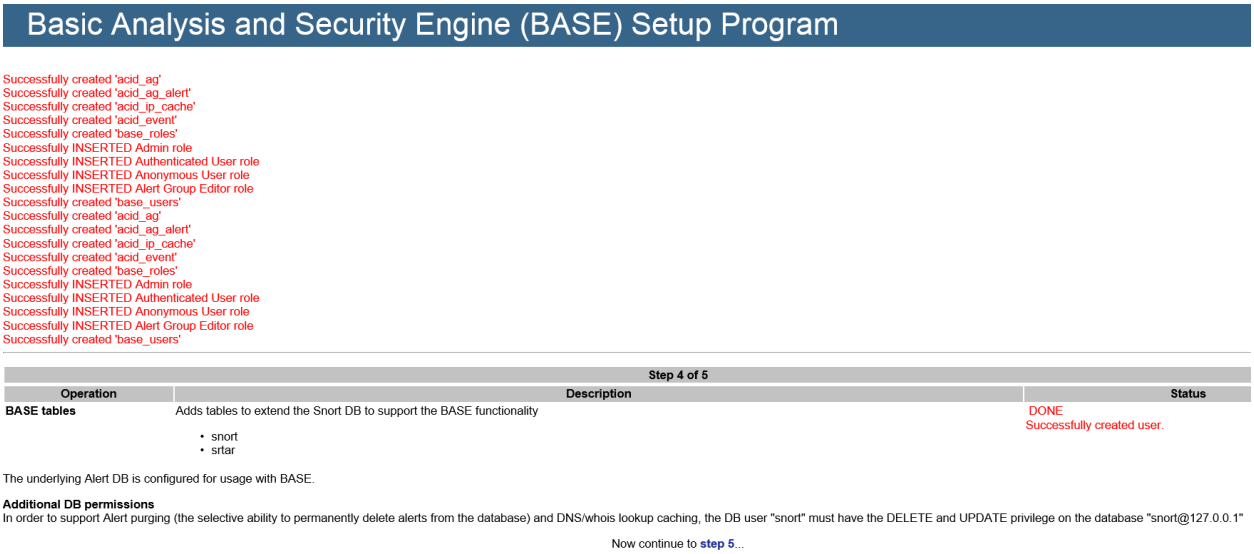
Sonra isə **BASE**-ə authentifikasiya ilə girmək istəyirsinizsə(**Mütləq lazımdır**), onda sistem istifadəçisini istifadə edərək bura daxil olmaq üçün selectorla seçirik. Mən **root** seçdim.



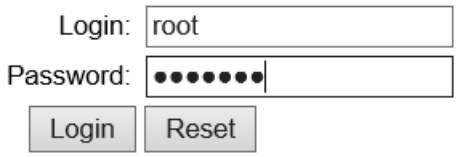
Sonda isə **Create BASE AG** düyməsini sıxırıq. **Step 5** düyməsinə sıxırıq.



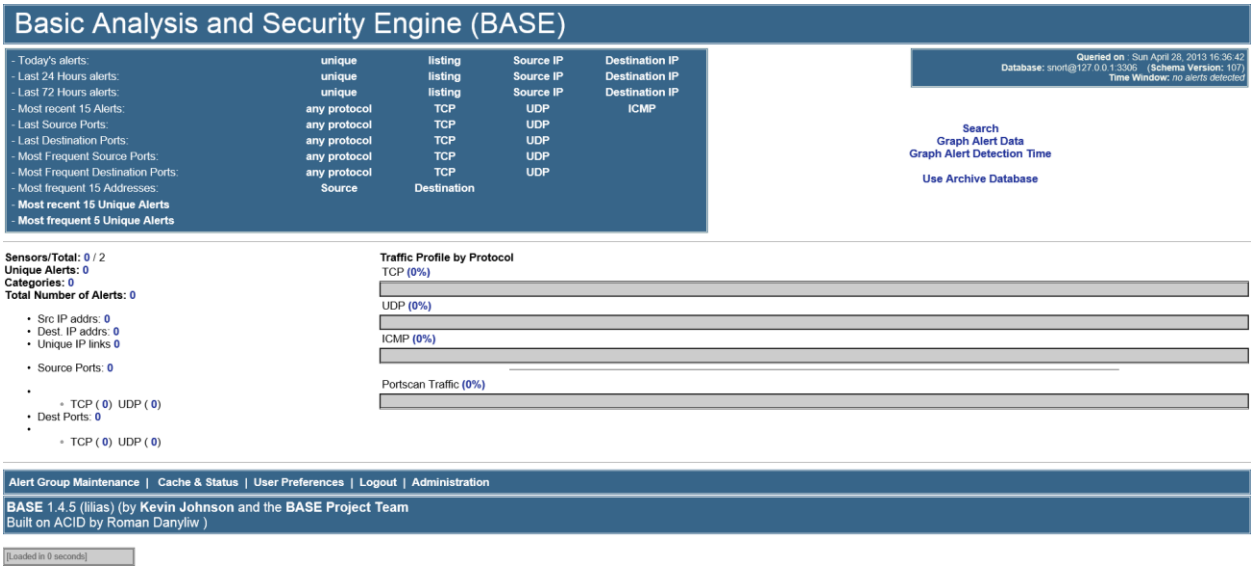
Və nəticə aşağıdakı şəkilə uyğun formada çap edilməlidir.



Sistem root istifadəçisi və şifrəsini daxil edib giriş edirik.



Aşağıdakı formada şəkil çap ediləcək.



OpenSSL RSA imzalanması və yoxlanılması qaydası

OpenSSL asan yollu imkan yaradır ki, RSA alqoritmi ilə data imzalansın. RSA ilə imzalama verilənlərin bütövlüyü və doğruluğuna təminat verir.

RSA imzalama alqoritmi

Bütöv verilənlərin imzalanması əvəzinə, hash alqoritmi(məsələn **SHA256**) istifadə edərək, birtərəfli hash verilənlərini yaradacayıq, hash-i imzalayacayıq(faktiki imzanı generasiya edir), sonra datanı ardıcıl olaraq imzaya ötürəcəyik.

Bitən son verilənlərin hash-ni hesablayacaq(eyni HASH alqoritmini istifadə edərək), sonra açıq açarı istifadə edərək imzanı yoxlayacaq.

Aşağıda RSA alqoritmini istifadə edərək detallı şəkildə datanın imzalanmasını və yoxlanılmasını açıqlayırıq.

RSA alqoritmi istifadə edərək datanın imzalanması

Addım1. Private/Public açar cütlüyünün yaradılması(əlavə)

```
openssl genrsa -out private.pem 1024
```

Bu **private.pem** adlı key faylı yaradır. Bu fayl həm Private həmdə Public açarı özündə təşkil edir. Həmçinin biz Public açarı bu fayldan ayırmalıyıq.

```
openssl rsa -in private.pem -out public.pem -outform PEM -pubout
```

Artıq **public.pem** adlı PUBLIC açar var. Siz bu açarı istənilən 3-cü tərəf program təminatı ilə istifadə edə bilərsiniz.

Addım2. Datanın HASH-ni yaradaq.

```
echo 'data to sign' > data.txt
openssl dgst -sha256 < data.txt > hash
```

Addım3. Private açarı istifadə edərək datanı imzalayaq.

```
openssl rsautl -sign -inkey private.pem -keyform PEM -in hash  > signature
```

Artıq **'signature'** və hal-hazırki faktiki **'data.txt'** faylı son bitənlə əlaqələndirilə bilər. Hash alqoritmi(bizim halda **SHA256**) public açar kimi, qəbul edilən son tərəf üçün tanınmalıdır.

Public açarı istifadə edərək datanı authentifikasiyadan keçirək

Addım4. signature-ni yoxlayaq

```
openssl rsautl -verify -inkey public.pem -keyform PEM -pubin -in signature >
verified
```

```
diff -s verified hash
```

Əgər öncəki əmrimizdə **verified** faylı tam olaraq Addım3-də generasiya elədiyimiz **hash** faylı ilə tam üst-üstə düşürse(əmrin nəticəsi **'Files verified and hash are identical'** sözlərini çap etməlidir), onda signature doğrudur və datanın **doğruluğu/həqiqiliyi** tam sübut edilmiş sayılır.

OpenSSL şifrələnmə və deşifrələmə

Ilk olaraq **file.txt** adlı faylı **des3** alqoritmi ilə şifrələyib **encrypted.txt** adlı fayla yazaq.

```
root@openssl:/root/folder # openssl des3 -in file.txt -out encrypted.txt
enter des-ede3-cbc encryption password: Şifrələmə parolu
Verifying - enter des-ede3-cbc encryption password: Şifrələmə parolu təkrar

root@openssl:/root/folder # openssl des3 -d -in encrypted.txt -out normal.txt
enter des-ede3-cbc decryption password: Şifrələmədə yazılan parol
```

OpenSSL RSA açarlar və sertifikatlar

Əksər istifadə edilən əmrlər

Test üçün RSA public/private açarları yaradırıq

Əlaqəli private/public açarların yaradılması

```
root@owncloud:/root/openssltest # openssl genrsa -des3 -out private-3des-
2048.pem 2048
Generating RSA private key, 2048 bit long modulus
.....................................................................
.....................................................................+++
.............+++
e is 65537 (0x10001)
Enter pass phrase for private-3des-2048.pem: Şifrələmə Parolu
Verifying - Enter pass phrase for private-3des-2048.pem: Şifrələmə parolu
təkrar
```

3DES ilə şifrələnmiş **PEM** açarı deşifrə edək və onu **DER**-ə convert edək.

```
openssl rsa -in private-3des-2048.pem -outform DER -out private-2048.der
```

PKI CA əməliyyatları

PKI CA yaradılması

- OpenSSL-i yükləyin.
- CA üçün qovluq yaradın.

```
root@owncloud:/root/openssltest # mkdir /root/CA
```

- **CA.pl** faylının ünvanını tapın və həmin faylı **/root/CA** qovluğuna nüsxələyin.
- **'/etc/ssl/openssl.cnf'** faylını özünüzə uyğun yeniləyin.
- Yeni CA yaradın.

```
root@owncloud:/root/CA # find / -name CA.pl
/usr/local/openssl/misc/CA.pl
/usr/src/crypto/openssl/apps/CA.pl
/usr/ports/security/openssl/work/openssl-1.0.1e/apps/CA.pl
```

Ən yenisini götürürük.

```
root@owncloud:/root/CA # cp /usr/ports/security/openssl/work/openssl-
1.0.1e/apps/CA.pl /root/CA/
root@owncloud:/root/CA # chmod +x CA.pl

root@owncloud:/root/CA # ./CA.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
```

```
.........................++++++
..........++++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase: Şifrələmə parolu daxil edirik
Verifying - Enter PEM pass phrase: Şifrələmə parolu daxil edirik təkrar
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:BAKU
Locality Name (eg, city) []:XATAI
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ATL
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:domain.az
Email Address []:jamal.shahverdiyev@domain.az

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            ce:2c:98:70:f5:62:e4:eb
        Validity
            Not Before: Dec 22 02:12:28 2013 GMT
            Not After : Dec 21 02:12:28 2016 GMT
        Subject:
            countryName               = AZ
            stateOrProvinceName       = BAKU
            organizationName          = DOMAIN
            organizationalUnitName    = IT
            commonName                = domain.az
            emailAddress              = jamal.shahverdiyev@domain.az
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                7E:D5:18:9B:6C:14:35:4C:E1:A0:38:A9:33:3C:40:7F:EB:5E:9B:C8
            X509v3 Authority Key Identifier:

keyid:7E:D5:18:9B:6C:14:35:4C:E1:A0:38:A9:33:3C:40:7F:EB:5E:9B:C8

DirName:/C=AZ/ST=BAKU/O=DOMAIN/OU=IT/CN=domain.az/emailAddress=jamal.shahverd
iyev@domain.az
                serial:CE:2C:98:70:F5:62:E4:EB
```

```
            X509v3 Basic Constraints:
                CA:TRUE
Certificate is to be certified until Dec 21 02:12:28 2016 GMT (1095 days)

Write out database with 1 new entries
```

SSL sertifikatlarını yaradaq

- Sertifikat müraciətlərini yaradaq

```
root@owncloud:/root/CA # ./CA.pl -newreq
Generating a 1024 bit RSA private key
.....................................+++++
.....................................+++++
writing new private key to 'newkey.pem'
Enter PEM pass phrase: Şifrələnmə parolunu daxil edirik.
Verifying - Enter PEM pass phrase: Şifrələnmə parolunu təkrar daxil edirik.
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AZ
State or Province Name (full name) [Some-State]:BAKU
Locality Name (eg, city) []:Xatai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:DOMAIN
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:client
Email Address []:client@domain.az

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request is in newreq.pem, private key is in newkey.pem
```

- Müraciətləri imzalayaq ki, SSL sertifikatları generasiya edə bilək.

```
root@owncloud:/root/CA # ./CA.pl -sign
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            ce:2c:98:70:f5:62:e4:ec
        Validity
            Not Before: Dec 22 02:20:17 2013 GMT
            Not After : Dec 22 02:20:17 2014 GMT
        Subject:
```

```
            countryName               = AZ
            stateOrProvinceName       = BAKU
            localityName              = Xatai
            organizationName          = DOMAIN
            organizationalUnitName    = IT
            commonName                = client
            emailAddress              = client@domain.az
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                EB:85:67:45:EC:31:DF:BA:63:6E:8A:54:DE:A5:0B:3F:D9:34:83:4D
            X509v3 Authority Key Identifier:

keyid:7E:D5:18:9B:6C:14:35:4C:E1:A0:38:A9:33:3C:40:7F:EB:5E:9B:C8

Certificate is to be certified until Dec 22 02:20:17 2014 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
```

- Yeni generasiya edilmiş sertifikatı, açar və müraciətin yerini dəyişək.

```
root@owncloud:/root/CA # mkdir someone ; mv new*.* ./someone/
```

pkcs12 SSL sertifikatlarını yaradaq.

```
root@owncloud:/root/CA/someone # openssl pkcs12 -export -in newcert.pem -
inkey newkey.pem -out certificate.p12
Enter pass phrase for newkey.pem: PEM açarın parolunu daxil edirik
Enter Export Password: Çıxış P12 parolunu daxil edirik
Verifying - Enter Export Password: Çıxış P12 parolunu təkrar daxil edirik
```

Digər PKI əməliyyatları

Inamlı root CA SSL sertifikatlarını import edirik.

Burda OpenSSL sertifikatının hash faylının necə yaradılması və hash faylın sertifikata necə symlink edilməsi açıqlanır.

- 1. Script-i **certlink.sh** adı ilə **/etc/ssl/certs** ünvanına nüsxələyin.

```
mkdir /etc/ssl/certs                # Qovluğu yaradaq
```

```
ee /etc/ssl/certs/certlink.sh        # Fayla aşağıdakı məzmunu əlavə edirik.
```

```
#!/bin/sh
#
# usage: certlink.sh filename [filename ...]

for CERTFILE in $*; do
  # make sure file exists and is a valid cert
  test -f "$CERTFILE" || continue
  HASH=$(openssl x509 -noout -hash -in "$CERTFILE")
  test -n "$HASH" || continue

 # use lowest available iterator for symlink
 for ITER in 0 1 2 3 4 5 6 7 8 9; do
   test -f "${HASH}.${ITER}" && continue
   ln -s "$CERTFILE" "${HASH}.${ITER}"

   test -L "${HASH}.${ITER}" && break
 done
done
```

- 2. Scripti işə salaq.
  certlink.sh filename

filename yazılan yerdə **root(.pem)** CA SSL sertifikatdır.

```
root@owncloud:/root/CA/someone # ./certlink.sh newcert.pem
```

Client sertifikatının içindən CA sertifikatı(PEM-ə) açaq.

```
root@owncloud:/root/CA/someone # openssl pkcs12  -in certificate.p12  -out cacert.pem -cacerts -nokeys
Enter Import Password: Giriş şifrəsini daxil edirik
MAC verified OK
```

(.pem) key faylını və sertifikatı, clientin .p12 sertifikatından export edək:

```
root@owncloud:/root/CA/someone # openssl pkcs12  -in certificate.p12 -out certificate-cert.pem -clcerts -nokeys
Enter Import Password:
MAC verified OK
```

```
root@owncloud:/root/CA/someone # openssl pkcs12  -in certificate.p12 -out example-key.pem -nocerts
Enter Import Password: Giriş şifrəsini daxil exirik
MAC verified OK
Enter PEM pass phrase: Yeni PEM şifrəsini daxil edirik
Verifying - Enter PEM pass phrase: Yeni PEM şifrəsini təkrar daxil edirik
```

p7b(Windows-da generasiya edilmiş CA Sertifikatlar)-dan CA sertifikatı-ın .pem-ə açılması:

```
sopenssl pkcs7   -in certnew.p7b -out cacert.pem -inform DER -text -print_certs
```

OpenSSL imzalama və şifrələmə

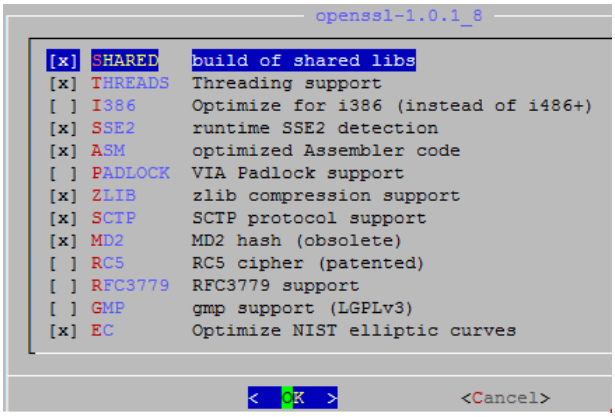
OpenSSL — SSL/TLS-lə işləmək üçün, açıq mənbə kodlu kriptoqrafiya paketidir. RSA, DH, DSA və X.509 sertifikatları açarlarını yaratmağa, onları imzalamağa, CSR-ı və CRT-nı formalaşdırmağa imkan yaradır. Həmçinin məlumatların şifrələnməsinin və SSL/TLS qoşulmalarının yoxlanmasının imkanı var.

UNIX/Linux(Solaris/OpenSolaris daxil olmaqla, Linux, Mac OS X, QNX4 [4], QNX6 və açıq mənbə koduyla BSD-ın dörd əməliyyat sistemi) tipli əksər əməliyyat sistemləri üçün mövcuddur, həmçinin OpenVMS və Microsoft Windows üçün mövcuddur.

OpenSSL SSLeay-a əsaslandırılaraq, Erik Yanq(Eric A. Young) və Tim Xadson(Tim Hudson) tərəfindən yazılmışdır hansı ki, 1998-ci ilin dekabrında **RSA Security** layihəsinin üzərində işləməyə başladıqda OpenSSL üzərində olan işin qeyri-rəsmi olaraq bitməsini elan etmişlər.

Once OpenSSL-i FreeBSD 9.2 x64 maşınımıza yükləyək.

```
cd /usr/ports/security/openssl          # Portuna daxil oluruq
make config                             # Lazımı modulları seçirik.
```




```
make install                            # Paketimizi yükləyirik
```

Açar cütlüyünü generasiya edək.

Açar cütlüyünün generasiya edilməsi çox asandır ancaq, öncə onun necə işlədiyini açıqlayaq. Öncə private açarınızı generasiya edəcəyik hansı ki, heç vaxt heç kəsə verməyəcəksiniz. Bu private açarı başlanğıc riyazi hesablmaları istifadə edərək generasiya edilir. Private key vasitəsilə public key generasiya edilir. Bu açarı siz hamı ilə böluşməlisiniz ancaq, sizin Public Key Infrastructure-nuz olmadığına görə siz, bu açarı ehtiyatla yayımlamalısınız.

Siz öz **Private** açarınız ilə nə isə şifrələyəndə, yalnız ona uyğun olan **PUBLIC** açarı onu deşifrə edə bilər. Bu o deməkdir ki, siz öz **PRIVATE** açarınız ilə nəsə şifrələyəndən sonra informasiya ötürdüyünüz şəxsdə sizin verdiyiniz **PUBLIC** açar olarsa, onu deşifrə edib açıb oxuya bilər. Uyğun olaraq sizin **PUBLIC** açar ilə şifrələnmiş məlumat da, yalnız sizin **PRIVATE** açar ilə deşifrə edilə bilər. Bu o deməkdir ki, əgər kimsə öz mail-ni sizin **PUBLIC** açar ilə şifrələyərsə, yalnız siz bunu oxuya bilərsiniz(Ona görə ki, **PRIVATE** açar yalnız sizin özünüzdə olur).

PRIVATE açarı generasiya edək.

```
root@owncloud:/root/folder # openssl genrsa -aes256 -out priv.pem
Generating RSA private key, 512 bit long modulus
..............++++++++++++
.++++++++++++
e is 65537 (0x10001)
Enter pass phrase for priv.pem: PAROL
Verifying - Enter pass phrase for priv.pem: PAROL
```

Sizdən şifrə soruşulacaq. Bu şifrə sizin PRIVATE açar faylınızı təhlükəsiz eləmək üçün istifadə edilir və buna görə də siz açarın istifadə edilməsi üçün şifrə daxil etməlisiniz. Indi isə biz uyğun olan PUBLIC açarı generasiya edək.

```
root@owncloud:/root/folder # openssl rsa -in priv.pem -out public.pem -
outform PEM -pubout
Enter pass phrase for priv.pem:
writing RSA key
```

Əgər siz yerləşdiyiniz qovluğun daxilində **ls** əmrini daxil eləsəniz görəcəksiniz ki, orda **priv.pem** və **public.pem** açar cütlüyü mövcuddur. Gəlin içində müəyyən məlumat olan fayl yaradaq.

```
root@owncloud:/root/folder # echo "this is secret" > file.txt
```

İndi isə **PUBLIC** açarınız ilə faylı şifrələyək və şifrələnmiş mətni **file.txt.enc** adlı fayla ötürək.

```
root@owncloud:/root/folder # openssl rsautl -inkey public.pem -pubin -encrypt
-in file.txt > file.txt.enc
```

Bu faylda artıq oxuna bilməyən şifrələnmiş məlumat olmalıdır. Əgər biz indi həmin məlumatı **PRIVATE** açarımız ilə açsaq, oxunulacaq şəkildə normal məlumatı görəcəyik.

```
root@owncloud:/root/folder # openssl rsautl -inkey priv.pem -decrypt -in
file.txt.enc
```

Hər şey işləyir, artıq bizim inamlı olan şəxslərimizə PUBLIC açarı verə bilərik. Onlar şifrələdiyi istənilən məlumatı yalnız biz özümüz deşifrə edə biləcəyik. Orda digər nə isə varmı ki, biz **PRIVATE** açarı istifadə edə bilək? Bəli imzalanma. Siz hansısa foruma mesaj yerləşdirirsinizsə, faktiki olaraq siz onu imzalaya bilərsiniz. Bu **'Bəli bu həqiqətəndə, mənəm'** deməkdir. Yəni mən bunu yazan şəxsəm. **PRIVATE** açar şifrələyə biləcəyi simvol uzunluğuna məhdudiyyət var. Ona görədə biz öncə **sha1** hash-ni fayla istifadə edəcəyik və sonra həmin hash-i faylın yerinə şifrələyəcəyik. Artıq forumda post yazmaq istəyən istifadəçi sha1 ilə bu imzanı yoxlayır, imzanı deşifrə edir və onların uyğun olmasını yoxlayır.

```
root@owncloud:/root/folder # openssl dgst -sha1 -sign priv.pem file.txt >
file.txt.sig
Enter pass phrase for priv.pem: ŞIFRƏLƏNMƏ parolunu daxil edirik
```

```
root@owncloud:/root/folder # openssl dgst -sha1 -verify public.pem -signature
file.txt.sig file.txt
Verified OK
```

Sonuncu əmrdən sonra siz "**Verified OK**" görməlisiniz. Artıq siz böyük faylı şifrələmək istəyirsinizsə, bunu simmetrik açar ilə etməlisiniz və sonra həmin faylı simmetrik açar ilə şifrələməlisiniz. Biz bunu aşağıdakı kimi edəcəyik(Aşağıdakı əmrlər **BASH** SHELL mühitindədir):

```
[root@owncloud ~/folder]# MYKEY="" ; for((a=1;a<=100;a++)) do
MYKEY=$MYKEY$RANDOM ; done ; echo $MYKEY > file.txt.symkey ; MYKEY=""

[root@owncloud ~/folder]# openssl des3 -e -kfile file.txt.symkey -in file.txt
-out file.txt.symenc

[root@owncloud ~/folder]# openssl des3 -d -kfile file.txt.symkey -in
file.txt.symenc
      this is secret
```

Yuxarıda biz təsadüfi açar generasiya elədik və cıxışını **file.txt.symkey** faylına yazdıq. Ardınca **file.txt** faylını **file.txt.symkey**(açar kimi istifadə elədik) faylı ilə şifrələdik və çıxışını **file.txt.symenc** faylına yazdıq. Sonra **file.txt.symenc** faylını, **file.txt.symkey** faylı ilə açar kimi istifadə edib deşifrə elədik və çıxışı ekrana çap elədik.

Kimsə bunu reallıqda istifadə edirmi?

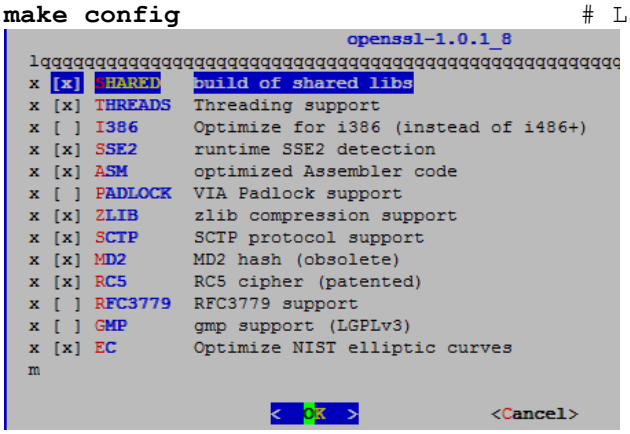
Bəli. Hər kəs bunun müəyyən bir versiyasını istifadə edir. HTTPS-lə olan sayta daxil olduqda nə baş verir? Adi halda HTTPS aktiv olan sayta daxil olduqda nə baş verir? Gəlin açıqlayaq:

1. Server sizə öz **PUBLIC** açarını yollayır. Bu sertifikat **ca1.random.com** Certificate Authority tərəfindən imzalanmışdır. **ca1.random.com** sertifikatı isə **VeriSign CA** tərəfindən imzalanmışdır. Sizin browser VeriSign CA-yə inanır və buna görə də, siz qəbul elədiyiniz sertifikatı etibarlı sayır.
2. Sizin browser təsadufi sessiya açarı generasiya edir(Bizim **MYKEY**-ə oxşar bir şey).
3. Sizin browser şifrələmə açarı kimi, saytdan gələn **PUBLIC** sertifikatı istifadə edir və şifrələnmiş mətni serverə yollayır.
4. Server isə öz **PRIVATE** açarını istifadə edir ki, session açarı və cavabları **decrypt** eləsin. Həmçinin verdiyi cavabları həmin sessiya açarı ilə şifrələyir.
5. Sizin browser və server artıq birlikdə təsadüfi session açarı istifadə edirlər və etibarlı əlaqə qururlar.

OpenSSL OCSP Responder

OCSP Responder - CA sertifikatın generasiya elədiyi sertifikatların köhnəlmiş üsul, CRL ilə yoxlanışın üstələdiyi yeni üsuldur. **Online Certificate Status Protocol** sayəsində biz müştərilərin sertifikatlarının onlayn yoxlanışını təmin edə bilərik. Hal-hazırda biz bu işi OpenSSL vasitəsilə FreeBSD9.2 x64 həm server və həm də client maşını olaraq istifadə edəcəyik.

```
cd /usr/ports/security/openssl/      # OpenSSL-i portlardan yükləyirik
make config                          # Lazımı modulları seçirik
```




```
make install                         # Yükləyirik
```

Ən başda olaraq qeyd edək ki, bizim server maşınızmda istifadə etdiyimiz **easy-rsa** scriptlərimiz OpenSSL üçün quraşdırma faylı olaraq özündə olan **openssl-0.9.8.cnf** versiyasını susmaya görə istifadə edir.

Client-imiz isə misal üçün FreeBSD maşında bunun üçün susmaya görə olan **/etc/ssh/openssl.cnf** faylından istifadə edir. Bunun üçün client-in **/etc/ssh/openssl.cnf** faylında olan **[usr_cert]** başlığına aşağıdakı sətiri əlavə edirik və faylı yadda saxlayıb çıxırıq.

authorityInfoAccess = OCSP;URI: http://192.168.214.131:8888

```
# Server maşına ilk olaraq easy-rsa scriptlərini CA üçün seçdiyimiz ünvana
nüsxələyirik. Misal üçün /root/certificates/ ünvanına.
cp -R /usr/local/share/easy-rsa/* /root/certificates

bash                 # Sonra isə BASH shell-inə keçid edirik

source ./vars        # Susmaya görə olan sertifikat dəyişənlərini elan edirik

./clean-all          # Susmaya görə olan açarları təmizləyirik.

KEY_SIZE=4096 ./build-ca --pass      # Sonra 4096 bitlik şifrəli CA server
                                     yaradırıq
Generating a 4096 bit RSA private key
..............................++
.....++
writing new private key to 'ca.key'
Enter PEM pass phrase: CA-Şifrəsi
Verifying - Enter PEM pass phrase: CA-Şifrəsi
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:AZ
State or Province Name (full name) [CA]:BAKU
Locality Name (eg, city) [SanFrancisco]:XATAI
Organization Name (eg, company) [Fort-Funston]:ITCom
Organizational Unit Name (eg, section) [changeme]:IT
Common Name (eg, your name or your server's hostname) [changeme]:responder
Name [changeme]:
Email Address [mail@host.domain]:ocspresponder@gmail.com


# Ardında isə valid adlı həqiqətəndə aktiv olan sertifikat yaradırıq
[root@ocsp-responder ~/certificates]# ./build-key valid
Generating a 1024 bit RSA private key
.........++++++
..............++++++
writing new private key to 'valid.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:AZ
State or Province Name (full name) [CA]:Baku
Locality Name (eg, city) [SanFrancisco]:Valley
Organization Name (eg, company) [Fort-Funston]:OPSO
Organizational Unit Name (eg, section) [changeme]:IT
Common Name (eg, your name or your server's hostname) [valid]:certchecker
Name [changeme]:
Email Address [mail@host.domain]:jamal.shahverdiyev@opensource.az

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /root/certificates/openssl-0.9.8.cnf
Enter pass phrase for /root/certificates/keys/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'AZ'
stateOrProvinceName   :PRINTABLE:'Baku'
localityName          :PRINTABLE:'Valley'
organizationName      :PRINTABLE:'OpSO'
organizationalUnitName:PRINTABLE:'IT'
commonName            :PRINTABLE:'certchecker'
name                  :PRINTABLE:'changeme'
emailAddress          :IA5STRING:'jamal.shahverdiyev@opensource.az'
Certificate is to be certified until Mar 15 19:35:26 2024 GMT (3650 days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated


# Həmçinin revoked adlı ancaq birazdan ləğv ediləcək sertifikat yaradırıq
[root@ocsp-responder ~/certificates]# ./build-key revoked
Generating a 1024 bit RSA private key
.........................++++++
...++++++
writing new private key to 'revoked.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:AZ
State or Province Name (full name) [CA]:Baku
Locality Name (eg, city) [SanFrancisco]:Valley
Organization Name (eg, company) [Fort-Funston]:OPSO
Organizational Unit Name (eg, section) [changeme]:IT
Common Name (eg, your name or your server's hostname) [revoked]:
Name [changeme]:
Email Address [mail@host.domain]:revoked@opensource.az

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /root/certificates/openssl-0.9.8.cnf
Enter pass phrase for /root/certificates/keys/ca.key: CA-nin şifrəsini daxil
edirik
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'AZ'
stateOrProvinceName   :PRINTABLE:'Baku'
localityName          :PRINTABLE:'Valley'
organizationName      :PRINTABLE:'OPSO'
organizationalUnitName:PRINTABLE:'IT'
commonName            :PRINTABLE:'revoked'
name                  :PRINTABLE:'changeme'
emailAddress          :IA5STRING:'revoked@opensource.az'
Certificate is to be certified until Mar 15 19:37:00 2024 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

revoked adlı sertifikatı ləğv edirik.
[root@ocsp-responder ~/certificates]# ./revoke-full revoked
```

```
Using configuration from /root/certificates/openssl-0.9.8.cnf
Enter pass phrase for /root/certificates/keys/ca.key:
Revoking Certificate 02.
Data Base Updated
Using configuration from /root/certificates/openssl-0.9.8.cnf
Enter pass phrase for /root/certificates/keys/ca.key:
revoked.crt:
/C=AZ/ST=Baku/L=Valley/O=OPSO/OU=IT/CN=revoked/name=changeme/emailAddress=rev
oked@opensource.az
error 23 at 0 depth lookup:certificate revoked

# Serveri işə salırıq ki, 8888-ci portda qulaq assın
[root@ocsp-responder ~/certificates]# openssl ocsp -index keys/index.txt -CA
keys/ca.crt -rsigner keys/ca.crt -rkey keys/ca.key -port 8888
Enter pass phrase for keys/ca.key: CA serverin şifrəsini daxil edirik
Waiting for OCSP client connections...

# Valid sertifikatı yoxlamaq üçün ca.crt, valid.crt fayllarını client maşının
root qovluğuna Upload edirik ki, yoxlanış edə bilsin.
root@ocsp-client:~ # openssl ocsp -CAfile ca.crt -issuer ca.crt -cert
valid.crt -url http://192.168.214.131:8888
Response verify OK
valid.crt: good
        This Update: Mar 18 21:18:19 2014 GMT

# Sonra isə Revoke edilmiş sertifikatı revoked.crt və CA sertifikatı ca.crt-
ni client maşının root qovluğuna Upload edib yoxlayırıq.
root@ocsp-client:~ # openssl ocsp -CAfile ca.crt -issuer ca.crt -cert
revoked.crt -url http://192.168.214.131:8888
Response verify OK
revoked.crt: revoked
        This Update: Mar 18 20:28:19 2014 GMT
        Revocation Time: Mar 18 19:37:57 2014 GMT
```

BÖLÜM 16

Təhlükəsizlik kamera görüntülərinin qeydiyyatı

- **NGINX və FFMPEG vasitəsilə kamera yayımının canlı izlənilməsi və köhnə yazılarına**

Əgər şirkətinizin daxili kamera görüntüləri sistemi varsa, kameralar İP ilə işləyirsə və standart RTSP protokolunu dəstəkləyirsə açıq qaynaqlı proqram təminatı vasitəsilə bu görüntü əldə oluna və ya canlı izlənilə bilər. Bu başlığımızda açıq qaynaqlı proqram təminatı FFMPEG və NGİNX vasitəsilə bu işi yerinə yetirəcəyik.

NGINX və FFMPEG vasitəsilə kamera yayımının canlı izlənilməsi və köhnə yazılarına baxılması

FFmpeg-in FREEBSD 10.1 üzərində quraşdırılması və video/audio fayllarının formatlarının dəyişilməsinə aid misallar

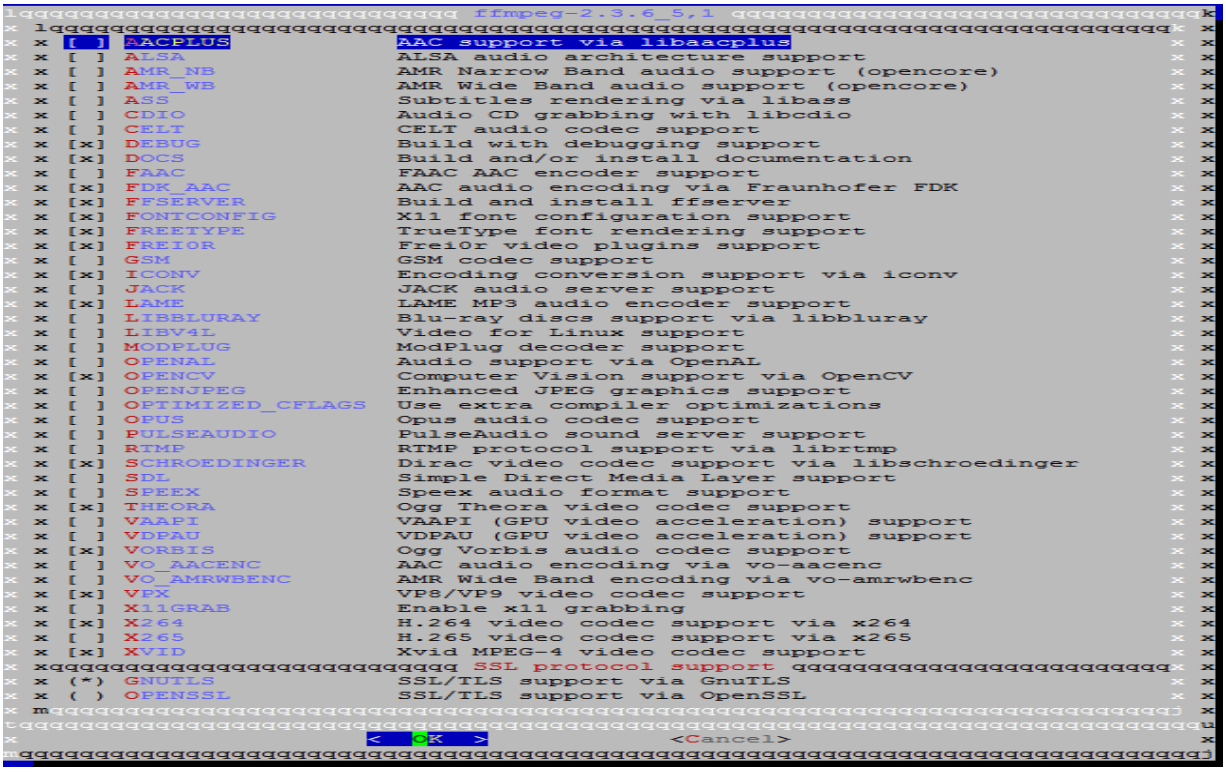
FFmpeg – açıq mənbə kodlu kitabxanaların yığımıdır hansı ki, rəqəmsal audio və video yazıları yazmağa, konversiya etməyə və fərqli formatlarda ötürməyə imkan yaradır. Tərkibinə audio/video kodlaşdırma/dekodlaşıdırma işini görən **libavcodec** kitabxanasını və mediakonteynerə multipleksləşmə/demultipleksləşmə libavformat daxil edir. Adı ekspert qrupu **MPEG** və FF-dən əsaslanır.

ffmpeg aşağıdakı komponentlərdən ibarətdir:
ffmpeg – Video faylın bir formatdan digər formata konvertasiya edilməsi üçün CLİ utilitidir. Onun köməyiylə həmçinin TV-kartdan real vaxtda videonu tutmaq olar.
ffserver – **HTTP**(RTSP hal-hazırda işlənir) video üçün axın və ya radioverilişlər serveri.
ffplay – SDL və FFmpeg kitabxanalarına əsaslanan sadə mediapleer.
libavcodec – Bütün audio/video kodekləri olan kitabxanadır. Kodeklərin əksəriyyəti ən yaxşı məhsuldarlıq təminatı üçün "sıfırdan" hazırlanmışdılar.
libavformat – müxtəlif audio,video formatlar üçün multipleksorlar və demultipleksorların kitabxanasıdır.
libavutil – ffmpeg-in müxtəlif komponentləri üçün standart ümumi alt proqramlarla köməkçi kitabxanadır. Tərkibinə Adler-32, CRC, MD5, SHA1, LZO-dekompressor, Base64 - şifrləyici/dekoder, DES - şifrləyici/şifraçan, RC4 - şifrləyici/şifraçan və AES - şifrləyici/şifraçan daxil edir.
libpostproc – videonun emalının standart alt proqramlarının kitabxanasıdır.
libswscale – videonun böyüdülməsi üçün kitabxanadır.
libavfilter – **vhook** əvəzinədir, hansı ki, dekoder və koder arasında video axının dəyişdirilməsinə şərait yaradır.

RTSP - Real Time Streaming Protocol axın protokolu, 1998-ci ildə IETF hazırlanmış və RFC 2326-da təsvir edilmişdir. Tətbiqi protokoldur, multimedia ilə işləyən sistemlərdə məlumat axının idarə edilməsinin istifadəsi üçün nəzərdə tutulmuşdur. Sayəsində **"Start" "Stop"** kimi əmrlərin isitfadəsi həmçinin serverdə yerləşdirilmiş fayllara vaxt üzrə girişə şərait yaradılır.

RTMP - Real Time Messaging Protocol axın məlumatların ötürülməsi üçün üstün sayılan protokoldur. Əsasən internet vasitəsilə veb-kameralardan video və audio axınların ötürülməsi üçün istifadə olunur.

```
# portsnap fetch extract update          => Portları yeniləyirik
# cd /usr/ports/multimedia/ffmpeg        => Qovluğa daxil oluruq
# make config                            => Aşağıdakı kimi quraşdırırıq
```




```
# make install clean -DBATCH          => FFMPEG-i portlardan yükləyirik
# rehash                              => Binar fayllarını yeniləyirik
```

FFMPEG-in dəstəklədiyi video və audio kodek-lərin siyahısını aşağıdakı əmrlə görə bilərik

```
# ffmpeg -codecs
```

FFMPEG-in dəstəklədiyi video və audio format-ların siyahısını aşağıdakı əmrlə görə bilərik

```
# ffmpeg -formats
```

Bir video formatını (misalçün .MP4) digər bir formata (.AVİ) aşağıdakı misaldakı kimi dəyişə bilərik

```
# ffmpeg -i test.mp4 test.avi
```

Hər hansı bir səsli video-dan səsi ayrıca .mp3 formatında çıxaraq:

```
# ffmpeg -i test.avi -vn -ar 44100 -ac 2 -f mp3 test_ses.mp3
```

-i – girişdə istifadə ediləcək faylın adı
-vn – video yazmaq işini dayandır
-ar – audio nüsxənin frekansını təyin elə
-ac – audio kanalın nömrəsini təyin elə
-f – çıxış faylının formatını təyin elə

.WAV faylının .mp3 formatına çevrilməsi

```
# ffmpeg -i test.wav -vn -ar 44100 -ac 2  -f mp3 test.mp3
```

.avi formatının .flv formatına çevrilməsi və müəyyən ölçünün təyin olunması

```
# ffmpeg -i test.avi -ab 56 -ar 44100 -b 200 -r 15 -s 320x240 -f flv test.flv
```

-r – giriş/çıxış faylarının bir saniyədəki kadrın sayını təyin edir
-s – çıxış faylının ekran ölçüsünü təyin edir

Və s.

openRTSP və FFMPEG vasitəsi ilə İP kameradan canlı görüntünün saxlanılması

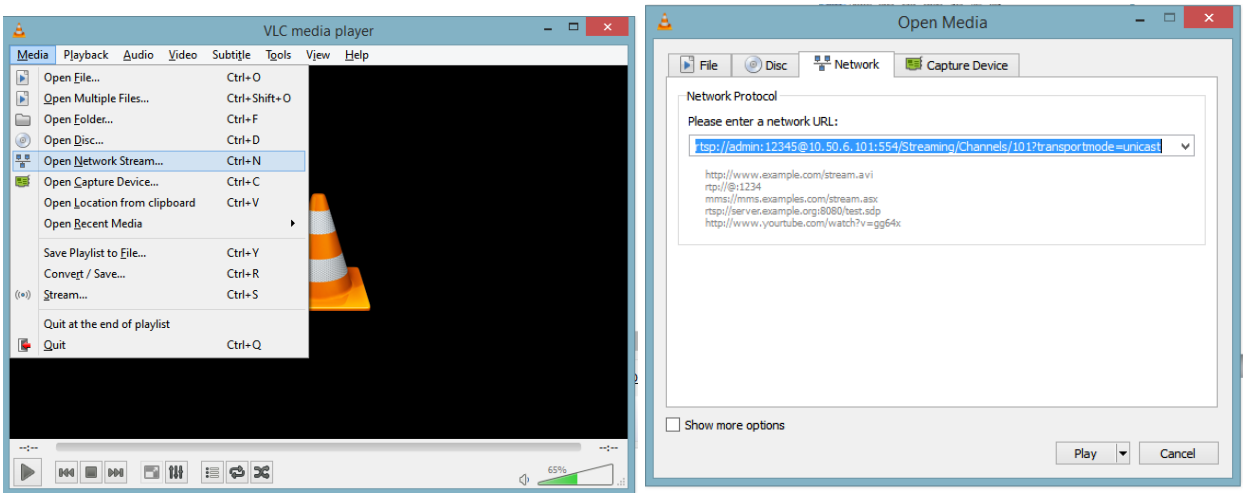
RTSP protokolu vasitəsi ilə VLC player-də hər hansı bir İP kameranın görüntüsünə baxmaq

Windows maşınımıza **VLC player** yükləyirik və şəbəkisinə girişimiz olduğumuz bir İP kameranın sənədlərindən RTSP URL-lərinə baxırıq.

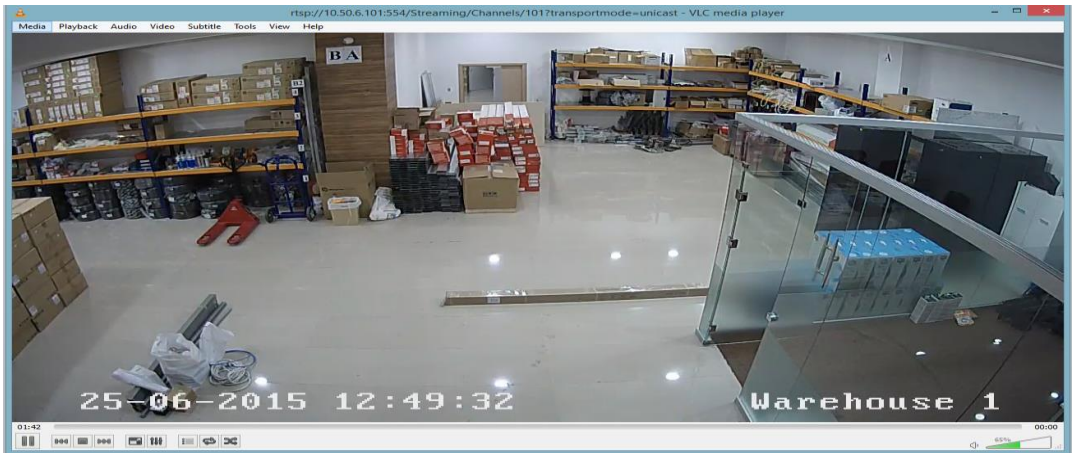
Misal üçün biz şəbəkəmizdə ip ünvanı "10.50.6.101", istifadəçi adı "admin" və şifrəsi "12345" olan bir Hikvision İP kamerasının RTSP ilə canlı görüntüsünə baxacayıq. Hikvision İP kameralarının rəsmi sənədindən təyin etdim ki, rtsp url aşağıdakı kimi olmalıdır.

rtsp://admin:12345@10.50.6.101:554/Streaming/Channels/101?transportmode=unicast

VLC media playeri açıb **Media->Open Network Stream** edib URL-ni daxil edib Play düyməsini sıxırıq.



Və nəticəni gördükdən sonra əmin oluruq ki, RTSP URL işləyir.



Sonra FREEBSD maşınımıza qayıdırıq.

 # cd /usr/ports/net/liveMedia => qovluğa daxil oluruq
 # make install clean => OpenRTSP-ni (LiveMedia) portlardan yükləyirik
rehash => Binar fayllarını yeniləyirik

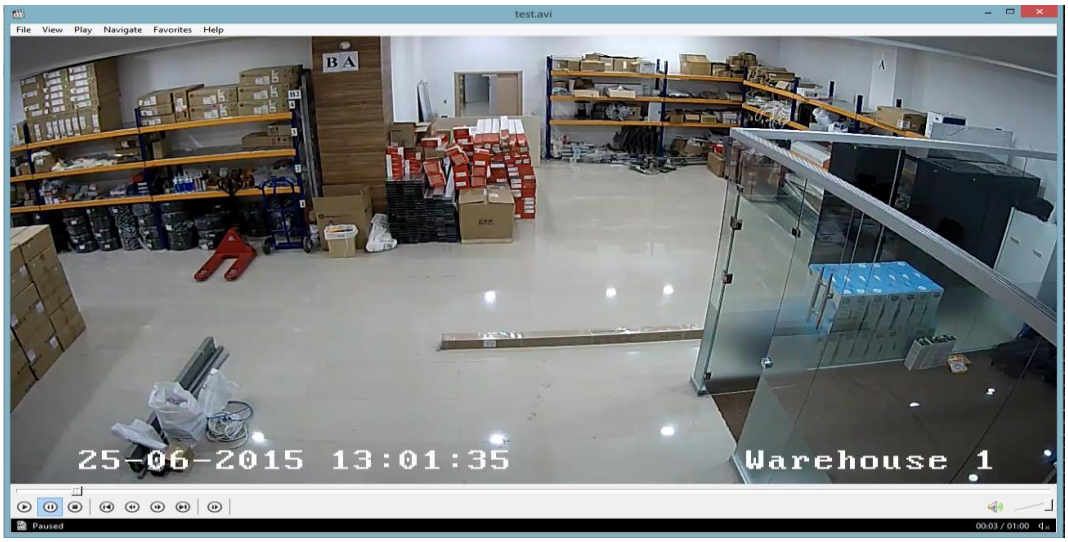
Sonra aşağıdakı əmrlə bu kameramızdan 1 dəqiqəlik görüntünü .avi formatında freebsd maşınımıza yazaq:
openRTSP -v -t -d 60s "rtsp://admin:12345@10.50.6.101:554/Streaming/Channels/101?transportmode=unicast" | ffmpeg -i - -y -r 20 -b 1000k -vcodec h264 -f avi test.avi

openRTSP
-v – Yalnız video yayımı oxut
-t – RTP/RSTP yayımını TCP üzərindən oxut (susmaya görə UDP olur)
-d – Yayımı oxutma müddəti təyin et

ffmpeg
-y – Çıxış faylının soruşmadan üzərinə yaz
-vcodec – çıxış faylının video kodekini təyin et

Kamera görüntüsünün yazısı bitdikdən sonra "**winscp.exe**" vasitəsi ilə "**test.avi**" faylını windows maşınımıza atıb windows player-də oxudub test edə bilərik.

Aşağıdakı şəkildən bu nümunənin nəticəsini görə bilərsiniz.



FFserver vasitəsilə video fayllarının və kameradan canlı yayımın web səhifəyə ötürülməsi

Səsli video faylın flv formatında web səhifəyə ötürülməsi

FFserver FFMPEG distrubitivinin bir hissəsi olduğu üçün FFmpeg paketi yükləndikdə FFserver servisi də hazır vəziyyətdə olur.

 # cd /usr/local/etc/ => FFserver quraşdırma faylının yerləşdiyi qovluğa daxil oluruq

 # ee ffserver.conf => Quraşdırma faylını açıb aşağıdakı kimi dəyişikliklər edirik

```
Port 8090                  # FFserver-in qulaq asdığı portu təyin edir
BindAddress 0.0.0.0        # Hansı interfeys ip-de qulaq asdığını təyin edir
MaxHTTPConnections 2000    # En cox ne qeder HTTP qoşulma ola bilər
MaxClients 1000            # En cox ne qədər istifadəçi qoşula bilər
MaxBandwidth 20480         # Istifadəçiyə video yayımi zamanı ən çox izin
                           verdiyin #sürət (kbit/s)
CustomLog /var/log/ffserver.log     # Jurnal faylının ünvanı

<Feed feed1.ffm>           # Hər bir mənbə yayım üçün təyin olunmus ana yayım
File /tmp/feed1.ffm        # Ana yayımın yerləşdiyi ünvan
FileMaxSize 500M           # Ana yayımın fayl ölçüsünə qoyulmuş limit
</Feed>                    # Ana yayımı sonlandırmaq üçün istifadə olunur

<Stream video.flv>         # İstifadəçilərə nümayiş olunan son yayım
    Format flv             # son yayımın formatı
    Feed feed1.ffm         # Hansı ana yayıma aid oldugu qeyd olunur (mənbənin
                           # təyini)
    VideoCodec libx264     # Son video yayımın kodekini təyin edir
    VideoFrameRate 30      # Son video yayımın bir saniyəsində olan kadrlarin sayı
    VideoBitRate 800       # Son video yayımın bit reytini (kb/s) təyin edir
    VideoSize 720x576      # Son video yayımın ekran ölçülərini təyin edir
```

```
    # aşağıdakı "AVoption" dəyişənləri birbaşa libavformat, libavdevice və
    #libavcodec kitabxanaları ilə əlaqəlidir və 2 cür mövcuddurlar,
    #Generic (hər bir kodek üçün istifadə oluna bilən) və Private (yalniz xas
    #olduqlari kodek üçün istifadə oluna bilən).
    AVOptionVideo crf 23
    AVOptionVideo preset medium
    AVOptionVideo me_range 16
    AVOptionVideo qdiff 4
    AVOptionVideo qmin 10
    AVOptionVideo qmax 51
    AVOptionVideo flags +global_header

    AudioCodec aac              # Ses kodekini təyin edir
    Strict -2             # Eksperimental kodekləri məcbur işə
                               # salmaq  üçün istifadə olunur
    AudioBitRate 128      # Səs kodekinin bit reytini (kb/s) təyin edir
    AudioChannels 2       # Yayım zamanı səs kanallarının sayını təyin edir
    AVOptionAudio flags +global_header

</Stream>                       # Yayımi sonlandırmaq üçün istifadə olunur

<Stream index.html>             # Index səhifəsini təyin edir
Format status             # Index səhifəsində bizə yayımlar barədə məlumat
verir
</Stream>                       # Yayımı sonlandırmaq üçün istifadə olunur
```

ee /etc/rc.conf =>Startup faylına ffserverin avtomatik işə düşməsi üçün aşağıdakı sətiri əlavə edirik

```
ffserver_enable="YES"
```

service ffserver start => Ffserveri işə salırıq

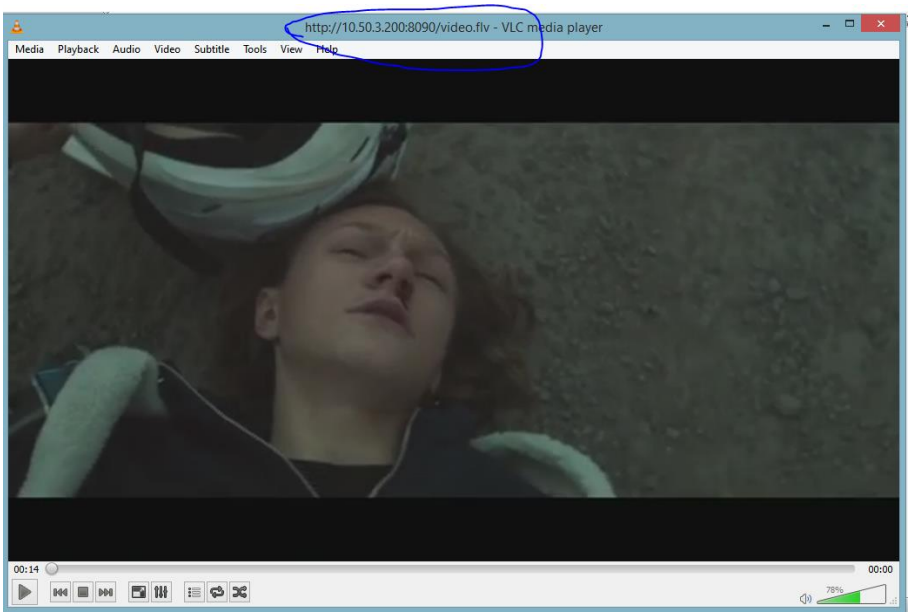
FFserverin jurnallarına quraşdırma faylında qeyd etdiyimiz "**/var/log/ffserver.log**" faylından baxa bilərsiniz.

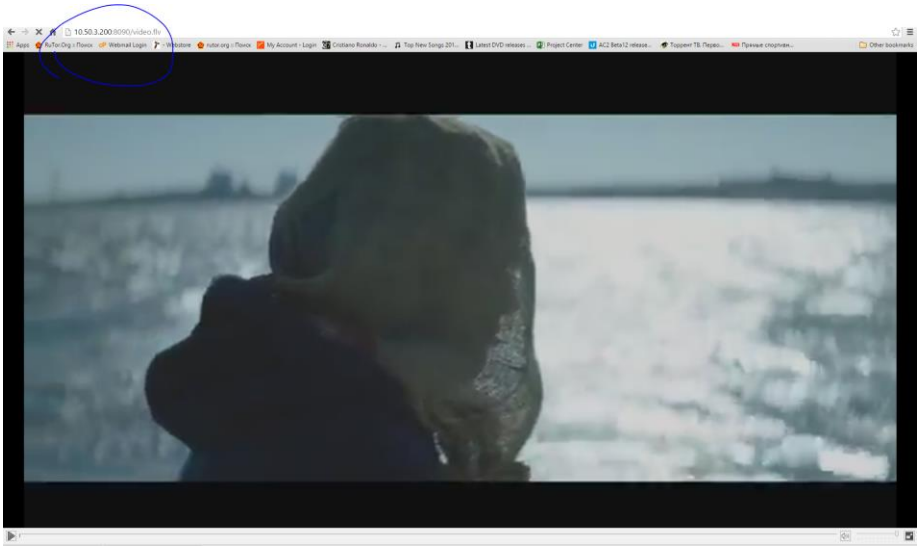
```
root@live:~ # tail -f /var/log/ffserver.log
Mon Jun 29 15:22:39 2015 FFserver started.
```

İndi isə FFserver-ə bir video faylını ötürək

ffmpeg -i /root/test.video/team.mp4 http://localhost:8090/feed1.ffm

Sonra isə istər uyğun kodekiniz varsa web browser-imizdə, istərsə də VLC player-imizdə http://ffserver.ip.add.ress:8090/video.flv linkini yazıb videomuzun yayınına baxa bilərik.

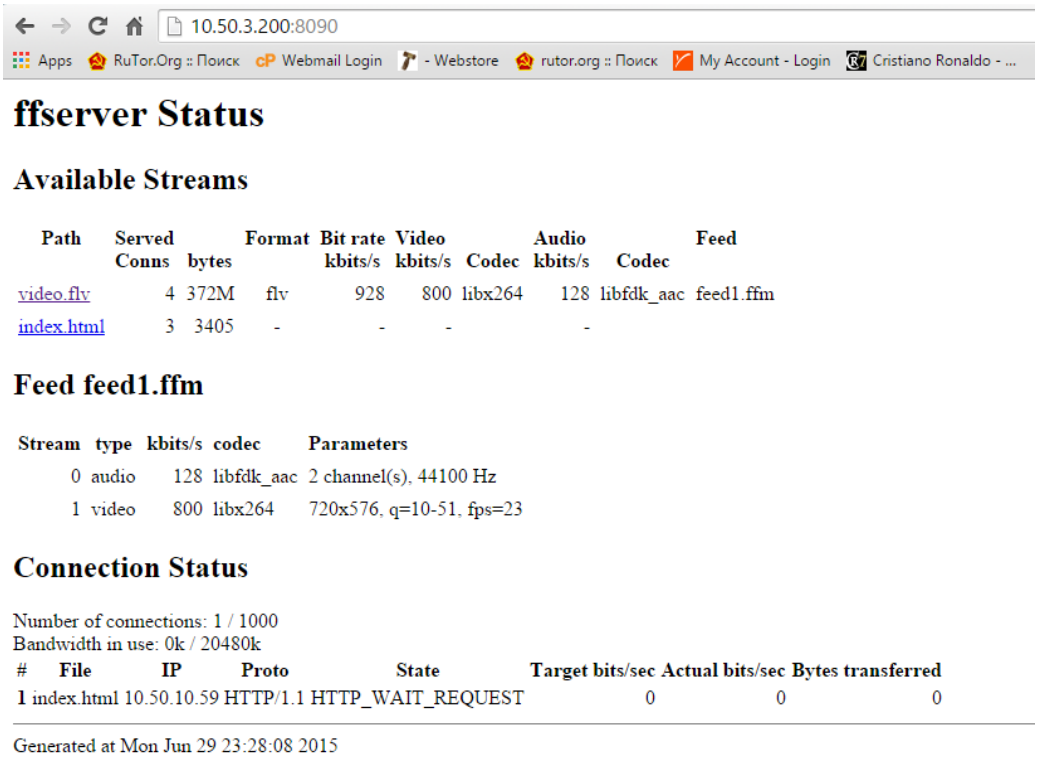




Budur, hər şey işlək vəziyyətdədir. **ffserver.conf** quraşdırma faylımızın sonuna əlavə etdiyimiz hissə bizim üçün yayımlarımız barədə status indeks səhifəsi yaradır.

```
</Stream>
<Stream index.html>
Format status
</Stream>
```

Siz web səhifənizdən http://ffserver.ip.add.ress:8090/ yığıb daxil olsanız, aşağıdakı kimi bir səhifə görəcəksiniz. Yayımlarınıza burdan da daxil ola bilərsiniz.



İndi isə kamera yayımını **ffserver** serverinə ötürək. 2-ci bir "Feed" yaradaq, həm videomuzu, həm də kamera yayımımızı serverimizə ötürək.

Bunun üçün eyni quraşdırma faylına aşağıdakı sətrləri **əlavə edirik.** Köhnə dəyişikliklərimiz olduğu kimi qalır.

ee /usr/local/etc/ffserver.conf => quraşdırma faylımıza daxil olub Aşağıdakı **qırmızı** rəngli sətrləri əlavə edirik.

```
Port 8090
BindAddress 0.0.0.0
MaxHTTPConnections 2000
MaxClients 1000
MaxBandwidth 20480
CustomLog /var/log/ffserver.log

<Feed feed1.ffm>
File /tmp/feed1.ffm
FileMaxSize 500M
</Feed>

<Stream video.flv>
    Format flv
    Feed feed1.ffm
    VideoCodec libx264
```

```
    VideoFrameRate 30
    VideoBitRate 800
    VideoSize 720x576
    AVOptionVideo crf 23
    AVOptionVideo preset medium
    AVOptionVideo me_range 16
    AVOptionVideo qdiff 4
    AVOptionVideo qmin 10
    AVOptionVideo qmax 51
    AVOptionVideo flags +global_header
    AudioCodec aac
    Strict -2
    AudioBitRate 128
    AudioChannels 2
    AudioSampleRate 44100
    AVOptionAudio flags +global_header
</Stream>

# Ikinci bir ana yayım yaradırıq
<Feed feed2.ffm>
Filə /tmp/feed2.ffm
FiləMaxSize 500M
</Feed>

# Yeni bir yayım yaradırıq və onu ikinci ana yayıma təyin edirik
<Stream camera.flv>
    Format flv
    Feed feed2.ffm
    VideoCodec libx264
    VideoFrameRate 25
    VideoBitRate 800
    VideoSize 1280x720
    AVOptionVideo crf 23
    AVOptionVideo preset medium
    AVOptionVideo me_range 16
    AVOptionVideo qdiff 4
    AVOptionVideo qmin 10
    AVOptionVideo qmax 51
    AVOptionVideo flags +global_header
    NoAudio
</Stream>

<Stream index.html>
Format status
</Stream>
```

```
# service ffserver restart          => Ffserver servisini yenidən işə salırıq
```

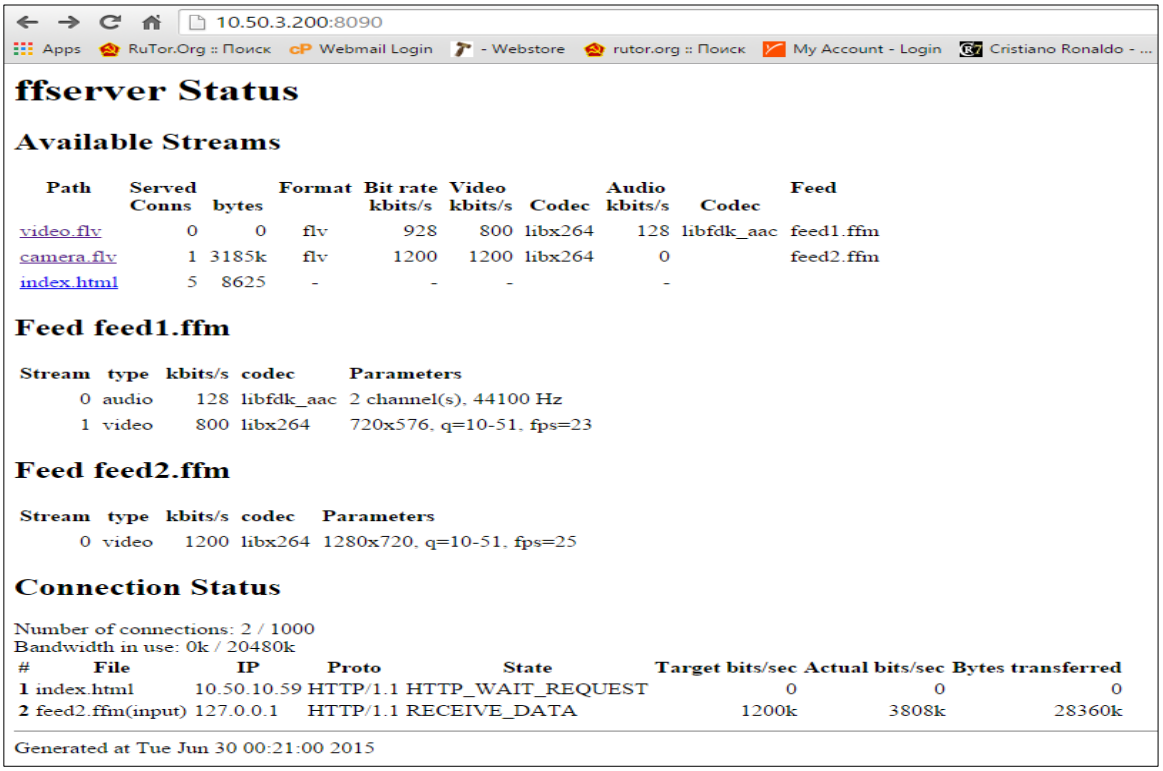
Sonra maşınımızda video faylını ötürmək üçün yenə də aşağıdakı əmri daxil edirik:

```
# ffmpeg -i /root/test.video/team.mp4 http://localhost:8090/feed1.ffm -loglevel debug
```

Eyni zamanda da kamera yayımını ötürmək üçün isə aşağıdakı əmri daxil edirik:

 # ffmpeg -i "rtsp://admin:12345@10.50.6.101:554/Streaming/Channels/101?transportmode=unicast" http://localhost:8090/feed2.ffm -loglevel debug

Status səhifəmizə web browser-dən daxil olub yayımlarımıza baxırıq



10.50.3.200:8090

Apps | RuTor.Org :: Поиск | Webmail Login | - Webstore | rutor.org :: Поиск | My Account - Login | Cristiano Ronaldo - ...

ffserver Status

Available Streams

Path	Served Conns	bytes	Format	Bit rate kbits/s	Video kbits/s	Codec	Audio kbits/s	Codec	Feed
video.flv	0	0	flv	928	800	libx264	128	libfdk_aac	feed1.ffm
camera.flv	1	3185k	flv	1200	1200	libx264	0		feed2.ffm
index.html	5	8625	-	-	-		-		

Feed feed1.ffm

Stream	type	kbits/s	codec	Parameters
0	audio	128	libfdk_aac	2 channel(s), 44100 Hz
1	video	800	libx264	720x576, q=10-51, fps=23

Feed feed2.ffm

Stream	type	kbits/s	codec	Parameters
0	video	1200	libx264	1280x720, q=10-51, fps=25

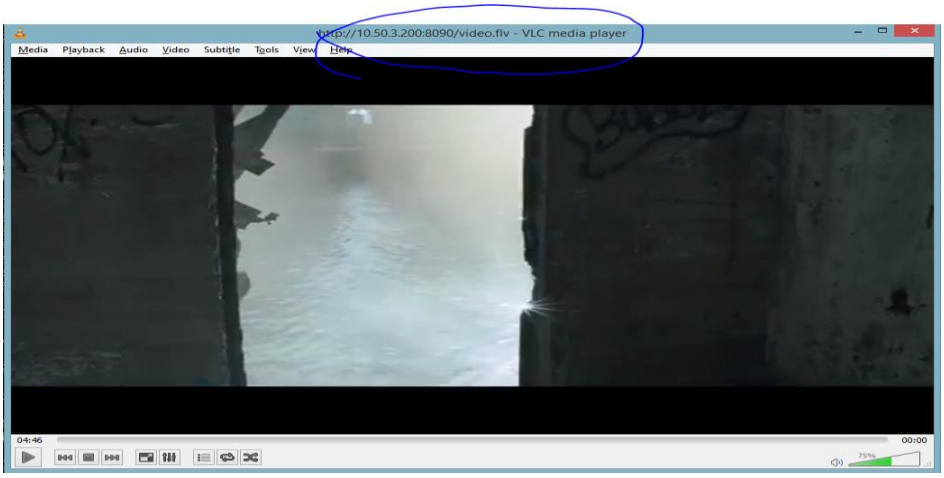
Connection Status

Number of connections: 2 / 1000
Bandwidth in use: 0k / 20480k

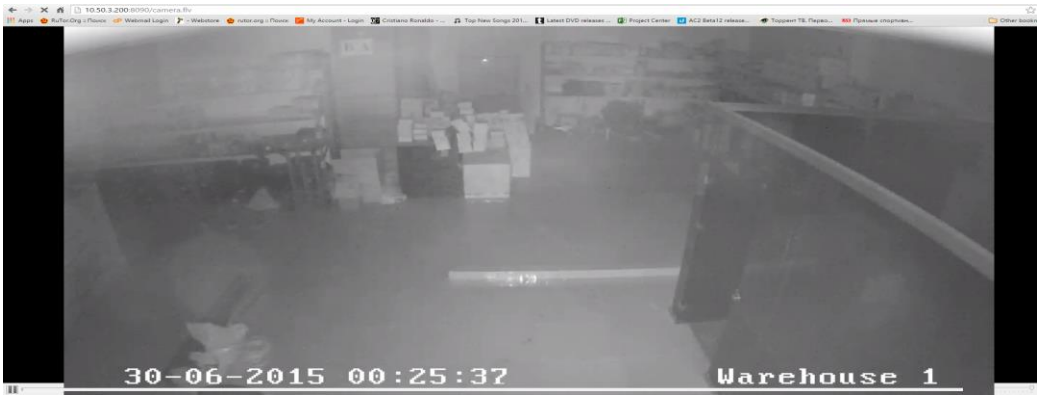
#	File	IP	Proto	State	Target bits/sec	Actual bits/sec	Bytes transferred
1	index.html	10.50.10.59	HTTP/1.1	HTTP_WAIT_REQUEST	0	0	0
2	feed2.ffm(input)	127.0.0.1	HTTP/1.1	RECEIVE_DATA	1200k	3808k	28360k

Generated at Tue Jun 30 00:21:00 2015

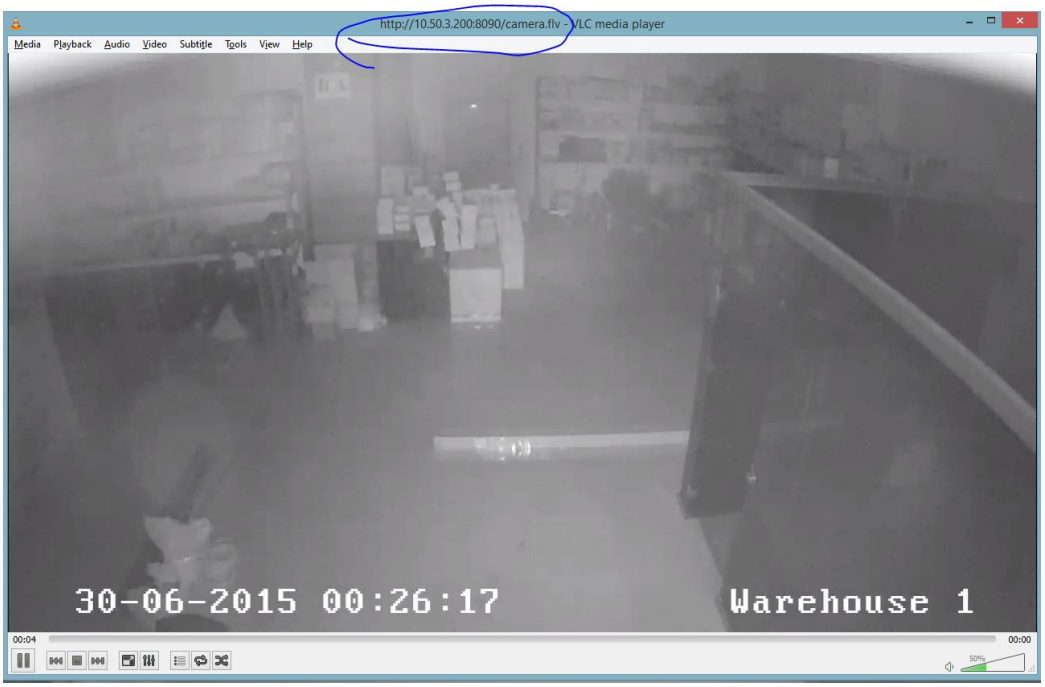
Yenə də **video.flv** yayımına daxil olsaq səsli videomuzu görəcəyik. VLC playerə də daxil edərək bunu əldə edə bilərik. Əvvəlki misalda bunu necə etdiyimizi qeyd etmişik.



Camera.flv yayımına daxil olsaq kameramızın canlı yayımını görəcəyik. FLV kodekini web browseriniz dəstəkləyirsə aşağıdakı kimi nəticə əldə edəcəksiniz.



VLC player-də isə aşağıdakı kimi nəticə əldə edəcəksiniz.



Kamera yayımınızı FFserver işə düşdükdən sonra avtomatik olaraq ffserver-ə dartmağını istəyirsinizsə, **/usr/local/etc/ffserver.conf** quraşdırma faylında aşağıda göstərilən **qırmızı** rənglə olan dəyişiklikləri edirik:

```
Port 8090
BindAddress 0.0.0.0
MaxHTTPConnections 2000
MaxClients 1000
MaxBandwidth 20480
CustomLog /var/log/ffserver.log
```

```
<Feed feed1.ffm>
Filə /tmp/feed1.ffm
FiləMaxSize 500M
</Feed>

<Stream video.flv>
    Format flv
    Feed feed1.ffm
    VideoCodec libx264
    VideoFrameRate 30
    VideoBitRate 800
    VideoSize 720x576
    AVOptionVideo crf 23
    AVOptionVideo preset medium
    AVOptionVideo me_range 16
    AVOptionVideo qdiff 4
    AVOptionVideo qmin 10
    AVOptionVideo qmax 51
    AVOptionVideo flags +global_header
    AudioCodec aac
    Strict -2
    AudioBitRate 128
    AudioChannels 2
    AudioSampleRate 44100
    AVOptionAudio flags +global_header
</Stream>

<Feed feed2.ffm>
Filə /tmp/feed2.ffm
FiləMaxSize 500M

# Aşağıdakı əmri FFserver işə düşdükdə avtomatik olaraq yerinə yetirərək
# "feed2" ana yayımı üçün mənbəni kameranın RTSP yayımından alır
Launch ffmpeg -i
"rtsp://admin:12345@10.50.6.101:554/Streaming/Channels/101?transportmode=unic
ast"
</Feed>

<Stream camera.flv>
    Format flv
    Feed feed2.ffm
    VideoCodec libx264
    VideoFrameRate 25
    VideoBitRate 800
    VideoSize 1280x720
    AVOptionVideo crf 23
    AVOptionVideo preset medium
    AVOptionVideo me_range 16
    AVOptionVideo qdiff 4
    AVOptionVideo qmin 10
    AVOptionVideo qmax 51
```

```
    AVOptionVideo flags +global_header
    NoAudio
</Stream>


<Stream index.html>
Format status
</Stream>
```

```
# service ffserver restart          => Ffserver servisini yenidən işə salırıq

# tail -f /var/log/ffserver.log     => Jurnal faylında görəcəksiniz ki,
                                     ffserver özü əmri işə salır
```

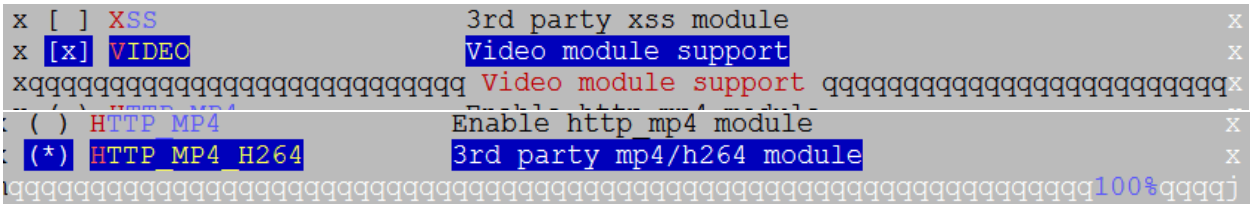
Tue Jun 30 00:31:02 2015 FFserver started.
Tue Jun 30 00:31:02 2015 Launch command line: /usr/local/bin/ffmpeg –i rtsp://admin:12345@10.50.6.101:554/Streaming/Channels/101?transportmode=unicast http://127.0.0.1:8090 feed2.ffm

Yenə də VLC player-də http://ffserver.ip.add.ress:8090/camera.flv yayımına daxil olsaq hər şeyin işlədiyinin şahidi olacayıq.

nGinx web server üzərində RTMP protokolu və FFmpeg vasitəsi ilə 2 ədəd kameranın RTSP yayımına JWPLAYER web player-lə baxmaq, yayımın yaddaşa qeyd olunması və köhnə yayımlara baxmaq

```
# cd /usr/ports/www/nginx           => qovluğa daxil oluruq
# make config                   => Standart olanlardan başqa aşağıdakı
modulları                             da seçirik
```

```
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq nginx-1.8.0_2,2 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x [x] DEBUG                Build with debugging support                     x
x+[ ] REDIS2                 3rd party redis2 module                          x
x+[x] RTMP                   3rd party rtmp module                            x
x x [ ] HTTP_DAV             Enable http_webdav module                        x
x x [x] HTTP_FLV             Enable http_flv module                           x
x x [ ] HTTP_GEOIP           Enable http_geoip module                         x
```



 # make install clean -DBATCH => Nginx web server (versiya 1.8.0_2,2) portlardan yükləyirik

 # rehash => Binar fayllarını yeniləyirik

 # cd /usr/local/etc/nginx/ => Nginx quraşdırma fayllarının olduğu qovluğa daxil oluruq

 # ee nginx.conf => NGİNX-in susmaya görə olan quraşdırma faylını açıb **qırmızı** ilə qeyd etdiyim sətrləri əlavə edirik

```
#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

# Jurnal faylını və tam ünvanını təyin edirik
error_log /var/log/nginx/nginx-error.log debug;

#pid        logs/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request"
'
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;
    # Virtual hostların quraşdırma fayllarının yerləşdiyi qovluqları daxil
```

```
    # edirik
    include sites-enabled/*;
    include sites-available/*;
    server {
        listen       80;
        server_name  localhost;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location / {
            root   /usr/local/www/nginx;
            index  index.html index.htm;
        }

        #error_page  404              /404.html;

        # redirect server error pages to the static page /50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   /usr/local/www/nginx-dist;
        }
    }

}

# RTMP protokolunun quraşdırmasını edirik
rtmp {
        # Giriş jurnal faylını və tam ünvanını təyin edirik
        access_log /var/log/nginx/rtmp_access.log;
        server {
                listen 1935;              # qulaq asdığı portu təyin edirik
                application live {
                live on;                  # 'Live' adında tətbiqetmə
yaradırıq

# Aşağıdaki exec_static əmrləri vasitəsi ilə NGINX işə düşdüyü zaman
#avtomatik olaraq "live" tətbiqetməsində "camera1" və "camera2" yayımlarına
#hərəsinə fərqli bir kamera yayımı ötürəcək

                exec_static /usr/local/bin/ffmpeg -i rtsp://10.41.10.25:554/
-c copy -f flv rtmp://localhost/live/camera1;
                exec_static /usr/local/bin/ffmpeg -i rtsp://10.41.10.4:554/ -
c copy -f flv rtmp://localhost/live/camera2;

                record all;   # Görüntünün yaddaşda saxlanmasını təmin edir
                record_path /var/videos;        # saxlanılan videoların
ünvanı
```

```
                record_suffix _%d-%b-%y-%T.flv;       # hər saxlanılan .flv
                                                      #videonun adına tarix və
                                                      #vaxt möhrü vurur
                record_interval 60m;      # 60 deqiqəlik video fayllar
yaradır

                }

        }
}
```

```
# mkdir /var/log/nginx/        => Jurnal faylı üçün qovluq yaradırıq
# mkdir /usr/local/etc/nginx/sites-enabled      => qovluğunu yaradırıq
# mkdir /usr/local/etc/nginx/sites-available    => qovluğunu yaradırıq
# mkdir /var/videos            => Videoların saxlanacağı qovluğu yaradırıq
# chown www:www /var/videos/   => NGINX demonuna bu qovluğa kamera yayımını
                               saxlamağa izin verilir
# nginx -t                     => əmri ilə əsas quraşdırma faylımızı
yoxlayırıq
```

Bizə bu çıxarışı qaytarırsa, quraşdırma faylında sintaksis səhvi yoxdur.

```
root@live:/usr/local/etc/nginx # nginx -t
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful
root@live:/usr/local/etc/nginx #
```

```
# cd /usr/local/etc/nginx/sites-enabled/        => Qovluğuna daxil oluruq
# ee camera1.conf       => "camera1.lan" virtual hostu üçün quraşdırma faylı
                         yaradırıq və aşağıdakı vəziyyətə gətiririk
```

```
server {
        listen 80;
        server_name camera1.lan;     # virtual hostun adını təyin edirik

# Virtual hostun bütün fayllarının yerləşdiyi qovluğun ünvanını göstəririk və
# index fayllarını təyin edirik
        location / {
                root /usr/local/www/camera1.lan;
                index index.php index.html index.htm;
        }
}
```

```
# mkdir /usr/local/www/camera1.lan        => camera1.lan virtual hostunun
                                           faylları üçün qovluğu yaradırıq
# cd /usr/local/www/camera1.lan           => Həmin qovluğa daxil oluruq


# ee index.html                    => Indeks səhifəsi yaradırıq və aşağıdakı
                                   kimi əlavələr edirik
```

JWPLAYER-i http://www.adrive.com/public/pN4j4w/jwplayer.zip linkindən Windows maşınıza endirib, içindəkiləri **WINSCP.EXE** vasitəsi ilə Freebsd serverinizdə **/usr/local/www/camera1.lan** qovluğuna atırsınız.

```
# ee index.html          => İndeks faylı yaradırıq və quraşdırmamızı edirik
```

```
# Indeks səhifəsine JWPLAYER-i daxil edirik
<script type="text/javascript" src="jwplayer.js"></script>

<div id="jwplayer.flash.swf">Loading the player ...</div>

    <script type="text/javascript">

        jwplayer('jwplayer.flash.swf').setup({

# Jwplayer üçün oxudacağı faylı təyin edirik. Burada Live tətbiqetməsi
#altında yaratdığımız camera1 yayımı olacaq, hansı ki, buna NGINX-in əsas
#quraşdırma faylında exec_static sintaksisin köməyi ilə yerinə yetirdiyimiz
#10.41.10.25 ünvanli kameramizin RTSP yayımı olacaq. Yuxarıda NGINX web
#serverimizin əsas quraşdırma faylında bunu görə bilərsiniz.

# aşağıdaki linkdə NGINX web serverimizin interfeys ip ünvanını yazırıq

        file: 'rtmp://10.50.3.200/live/camera1',

        # Jwplayer-in indeks səhifəsindeki ölçülər
        width: '1280',
        height: '720',
        aspectratio: '16:9'
});
</script>
```

```
# cd /usr/local/etc/nginx/sites-enabled/        => qovluğuna daxil oluruq
# cp camera1.conf camera2.conf                  => "camera2.lan" virtual hostunun
                                                 quraşdırma faylını camera1.conf-dan
                                                 nüsxələyirik
# ee camera2.conf              => Qırmızı ilə qeyd olunan dəyişiklikləri
edirik
```

```
server {
        listen 80;
        server_name camera2.lan;    # virtual hostun adını təyin edirik

# Virtual hostun bütün fayllarının yerləşdiyi qovluğun ünvanını göstəririk və
# index fayllarını təyin edirik
        location / {
                root /usr/local/www/camera2.lan;
                index index.php index.html index.htm;
        }
}
```

```
# cd /usr/local/www/          => Qovluğuna daxil oluruq
# cp -r camera1.lan/ camera2.lan/          => camera1.lan qovluğunu bütün fayl
```

və alt-qovluqları ilə birgə **camera2.lan** adı altında nüsxələyirik.

```
# cd  camera2.lan/              => Nüsxələnmiş qovluğa daxil oluruq
# ee index.html                 => İndeks səhifəsinin quraşdırma faylını açıb aşağıda qırmızı ilə göstərilmiş dəyişiklikləri edirik
```

```
# Indeks səhifəsinə JWPLAYER-i daxil edirik
<script type="text/javascript" src="jwplayer.js"></script>

<div id="jwplayer.flash.swf">Loading the player ...</div>

    <script type="text/javascript">

        jwplayer('jwplayer.flash.swf').setup({

# Jwplayer üçün oxudacağı faylı təyin edirik. Burada Live tətbiqetməsi
#altında yaratdığımız camera1 yayımı olacaq, hansı ki, buna NGINX-in əsas
#quraşdırma faylında exec_static sintaksisin köməyi ilə yerinə yetirdiyimiz
#10.41.10.4 ünvanli kameramizin RTSP yayımı olacaq. Yuxarıda NGINX web
#serverimizin əsas quraşdırma faylında bunu görə bilərsiniz.

 # aşağıdaki linkdə NGINX web serverimizin interfeys ip ünvanını yaziriq
        file: 'rtmp://10.50.3.200/live/camera2',

        # Jwplayer-in indeks səhifəsindəki ölçülər
        width: '1280',
        height: '720',
        aspectratio: '16:9'
});
</script>
```

```
# cd /usr/local/etc/nginx/sites-enabled/        => qovluğuna daxil oluruq
# cp camera1.conf play.conf          => Kameranın köhnə yazılarına baxmaq üçün yaratmaq istədiyimiz "play.lan" virtual hostunun quraşdırma faylını mövcud camera1.conf quraşdırma faylından nüsxələyirik
# ee play.conf                  => Qırmızı ilə qeyd olunan dəyişiklikləri edirik
```

```
server {
        listen 80;
        server_name play.lan; # virtual hostun adını təyin edirik

# Virtual hostun bütün fayllarının yerləşdiyi qovlugun ünvanını göstəririk və indeks fayllarını ləğv edib əvəzinə bir başqa qovluqda olan faylları indeks olaraq göstərməyini tələb edirik
        location / {
                root /var/videos;
                autoindex on;
```

```
                #index index.php index.html index.htm;
        }
}
```

ee /etc/rc.conf =>Startup faylına NGINX-in avtomatik işə düşməsi üçün aşağıdakı sətiri əlavə edirik

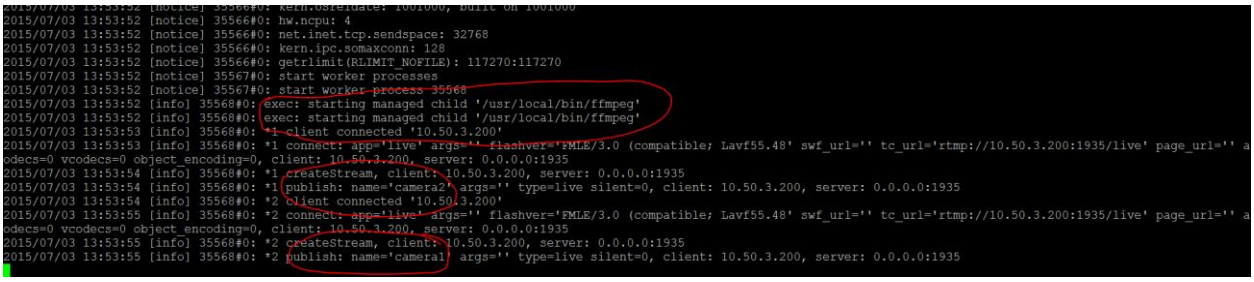
```
nginx_enable="YES"
```

NGİNX-i işə salırıq

service nginx start

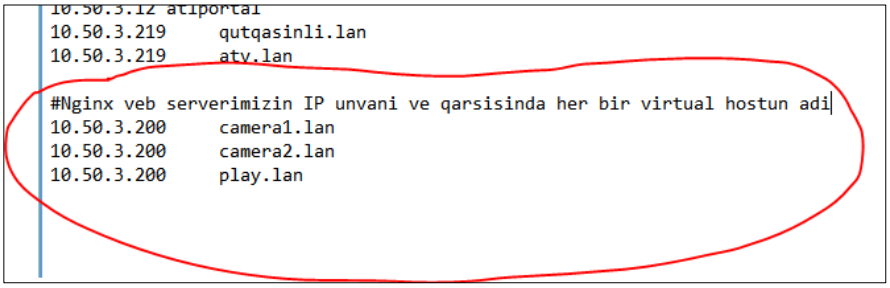
Jurnal faylını fərqli pəncərədə açırıq

tail -f /var/log/nginx/nginx-error.log



"Exec" əmrlərinin işə düşdüyünü və 2 ədəd (camera1 və camera2) yayımın avtomatik yarandığını görə bilərik.

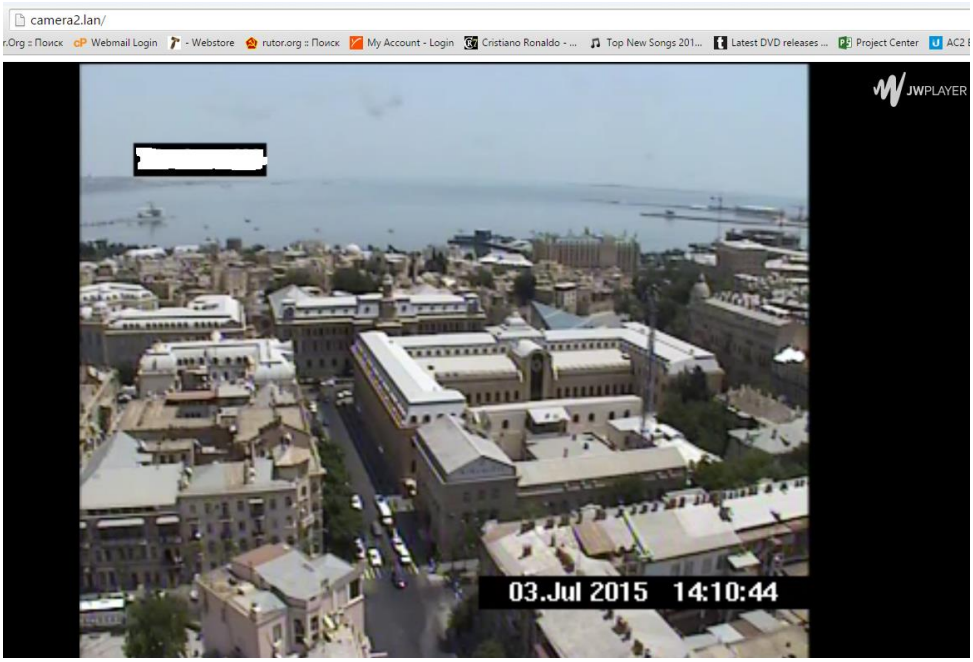
Yayımlara baxmaq üçün ilk öncə Virtual Host məntiqinin işə düşməsi üçün windows maşınımızda **C:\Windows\System32\drivers\etc\hosts** faylına aşağıdakı sətrləri əlavə etmək lazımdır.



Dəyişiklikləri edib yadda saxladıqdan sonra, web browser-imizdə virtual hostlara daxil oluruq və Jwplayer-də **PLAY** düyməsini sıxırıq.

http://camera1.lan/

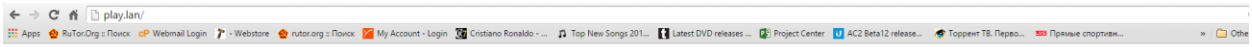
http://camera2.lan/



Gördüyümüz kimi hər bir fərqli virtual host-da fərqli kameraların canlı yayımını görürük

İndi isə Kamera yayımlarının köhnə video yazılarına baxaq

web browser-imizdən http://play.lan/ səhifəsini açırıq. Aşağıdakı kimi bir səhifə açılacaq və oradaq yayımımızın adları və tarix möhrü olan **1 saatlıq .flv** video fayllarını görəcəksiniz.



Index of /

../		
camera1_03-Jul-15-14:25:13.flv	03-Jul-2015 09:27	7862784
camera2_03-Jul-15-14:25:12.flv	03-Jul-2015 09:27	8235389

Bunlardan hansınınsa üzərinə sıxsaq, web browser-iniz kodeki dəstəkləyirsə əlavə səhifədə açacaq. Əgər, yoxdursa bu video faylını maşınınıza endirəcək.

Gördüyümüz kimi hər şey işləyir ☺

BÖLÜM 17

Sistem və şəbəkə resurslarının monitorinqi

- **FreeBSD Cacti yüklənməsi və qurulması**
- **Ubuntu üzərində Nagios server və client qurulması**
- **FreeBSD server üzərində NRPE agentin yüklənməsi**

Hər bir müəssisənin daxilində şəbəkə və sistem resursları kifayət qədər böyüdükdə və onların **24/7** işləməsi tələbi olduqda, həmin sistem və şəbəkə avadanlıqlarının monitorinqi tələbi mütləq şərt olacaq. Başlığımızda monitorinq üçün açıq qaynaqlı Nagios proqram təminatından istifadə edəcəyik. Proqram təminatı şəbəkəni SNMP protokolu, serverləri isə spesifik agent vasitəsili monitorinq edir və təyin edilən şərtlərə əsaslanaraq məktub və ya sms yollayır.

FreeBSD Cacti yüklənməsi və qurulması

CACTI – açıq qaynaqlı veb proqramdır hansı ki, RRDtool-un köməyi ilə qrafikləri qurmağa şərait yaradır. Cacti müəyyən edilmiş müvəqqəti intervallar üçün statistik məlumatları yığır və qrafik şəkildə onları əks etdirməyə icazə verir. Prosessorun yükləməsi, əməli yaddaşın ayrılması, işə salınmış proseslərin miqdarı, daxil olan/çıxan trafikdən istifadədə olan statistikanın təsviri üçün standart şablonlar istifadə olunur.

```
portsnap fetch extract update              # Öncə portları yenileyirik
```

Sistemi yeniləyək:

```
freebsd-update fetch
freebsd-update install
```

```
reboot                                     # Restart edirik
ntpdate 0.asia.pool.ntp.org                # Serverimizdə vaxtı dəqiq
                                           alırıq(Mütləq edilməlidir)
```

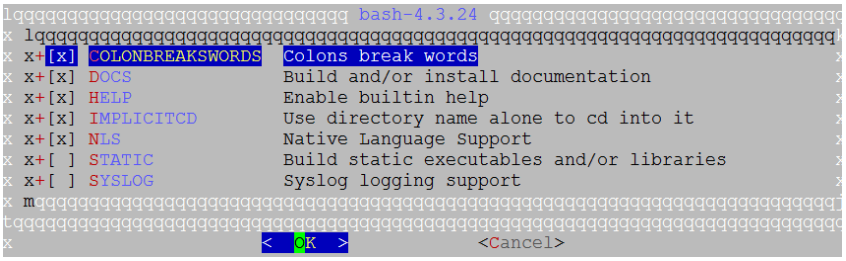
Sistemə aid ola yeniliklərin yüklənməsi üçün siz crona bir sətir əlavə edə bilərsiniz, ancaq bu sətir yenilikləri dartıb saxlayır. Vacibliyin təyin edilməsi üçün, yükləməni isə siz root istifadəçisinə gələn mailə baxdıqdan sonra etməlisiniz.

```
echo '@daily root freebsd-update cron' >> /etc/crontab
```

Lazımı paketləri yükləyək:

```
echo "NO_WARNING_PKG_INSTALL_EOL=yes" >> /etc/make.conf
```

```
cd /usr/ports/shells/bash                  # BASH shell portuna daxil oluruq
make WITHOUT="LIBSIGSEGV" install          # Lazımı modulları seçirik
```




```
echo "fdesc            /dev/fd          fdescfs rw      0       0" >>
/etc/fstab

chsh -s bash       # root istifadəçisi üçün shell-i bash edirik
```

Yeni sessiya ilə sistemə yenidən daxil oluruq ki, BASH shell işləsin.

NET-SNMP paketini yükləyirik:

```
cd /usr/ports/net-mgmt/net-snmp            # Port ünvanına daxil oluruq
BATCH=yes make WITHOUT="IPV6" install      # Yükləyirik
echo 'snmpd_enable="YES"' >> /etc/rc.conf        # Startup-a əlavə edirik
```

SNMPD-ni quraşdırırıq:

```
cd /usr/local/etc
mkdir snmp
cd snmp/
ee snmpd.conf      # snmpd.conf faylı yaradıb daxilinə aşağıdakı sətirləri
                   əlavə edirik.
syslocation  "Azerbaijan"
syscontact cacti
rwuser  freebsd noauth
rocommunity  freebsd    # Router-lə danışmaqda istifadə edilən pre-shared key
rwcommunity  freebsd
trapsink  localhost freebsd          # Localhost üçün pre-shared key
trap2sink  localhost freebsd
informsink  localhost freebsd
trapcommunity  freebsd
authtrapenable  2

/usr/local/etc/rc.d/snmpd start     # İşə salırıq
```

RRDTool-u yükləyirik (Asılılığında çoxlu paketlər olduğuna görə uzun vaxt alacaq):

```
cd /usr/ports/databases/rrdtool                  # Port ünvanına daxil oluruq
BATCH=yes make WITHOUT="PERL_MODULE" install     # Perl modulsuz yükləyirik
echo 'rrdcached_enable="YES"' >> /etc/rc.conf    # Startup-a əlavə edirik
```

MySQL-i yükləyək:

```
cd /usr/ports/databases/mysql55-server           # Port ünvanına daxil oluruq
BATCH=yes make -DWITH_OPENSSL install            # Yükləyirik
echo 'mysql_enable="YES"' >> /etc/rc.conf        # Startup-a əlavə edirik
/usr/local/etc/rc.d/mysql-server start           # İşə salırıq
```

Cacti üçün baza istifadəçi və şifrə yaradırıq:

```
mysql -uroot -p               # MySQL-ə qoşuluruq

mysql> CREATE DATABASE cacti;
Query OK, 1 row affected (0.01 sec)

mysql> GRANT ALL ON cacti.* TO cacti@localhost IDENTIFIED BY 'freebsd'; FLUSH
PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
```

Bazamızın root istifadəçisinə şifrə təyin edirik və şifrəsiz istifadəçiləri söndürürük:

```
mysql> use mysql
mysql> update user set password=password("freebsd") where user="root";
mysql> delete from user where user="";
mysql> FLUSH PRIVILEGES;
```

Apache-i yükləyirik:

```
echo "DEFAULT_VERSIONS+=apache=2.2" >> /etc/make.conf
cd /usr/ports/www/apache22                      # Port ünvanına daxil oluruq
BATCH=yes make -DWITHOUT_IPV6 install           # Yükləyirik
echo 'apache22_enable="YES"' >> /etc/rc.conf    # Startup-a əlavə edirik
```

/usr/local/etc/apache22/httpd.conf - Aşağıdakı sətirləri əlavə edirik və **DirectoryIndex** sətirinin qarşısını görünən kimi edirik:

```
DirectoryIndex index.php index.html
AddType application/x-httpd-php .php
AddHandler php5-script .php
```

/usr/local/etc/apache22/Includes/cacti.conf - Fayla aşağıdakı mətni əlavə edirik və yadda saxlayırıq

```
<Directory "/usr/local/share/cacti/">
AllowOverride None
Options None
Order allow,deny
Allow from all
</Directory>
Alias /cacti "/usr/local/share/cacti/"
```

```
/usr/local/etc/rc.d/apache22 start              # Apache-ı işə salırıq
```

PHP5-i yükləyirik:

```
cd /usr/ports/lang/php53            # Port ünvanına daxil oluruq
BATCH=yes lang_php53_UNSET=CGI lang_php53_UNSET=IPV6 lang_php53_SET=APACHE
make install                        # Yükləyirik

cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini # Konfiq faylını
                                                             nüsxələyirik
```

/usr/local/etc/php.ini - faylın içində **date.timezone** sətirini aşağıdakı kimi edirik:

```
[Date]
date.timezone = 'Asia/Baku'
```

```
cd /usr/ports/databases/php53-mysql    # MySQL connect üçün istifadə edilir
BATCH=yes make install                 # Yükləyirik

cd /usr/ports/net-mgmt/php53-snmp               # SNMP üçün tələb edilir
BATCH=yes make install                          # Yükləyirik

cd /usr/ports/lang/php53-extensions       # PHP5 genişlənmələrini yükləyirik
BATCH=yes make install

cd /usr/ports/www/php53-session           # Session-u yükləyirik
BATCH=yes make install
```

```
cd /usr/ports/net/php53-sockets            # Socket-lərə üçün tələb edilir
BATCH=yes make install

cd /usr/ports/textproc/php53-xml           # Reportlar üçün tələb edilə bilər
BATCH=yes make install

cd /usr/ports/graphics/php53-gd            # Həmçinin lazımdır və yükləyirik
BATCH=yes make WITHOUT="X11" install


CACTI-ni yükləyək və config edək:
cd /usr/ports/net-mgmt/cacti               # Portuna daxil oluruq
BATCH=yes make install                     # Yükləyirik

cd /usr/ports/net-mgmt/cacti-spine  # Sürəti artırmaq üçün istifadə edilir.
BATCH=yes make install

mysql -u cacti -pfreebsd cacti < /usr/local/share/cacti/cacti.sql
                                                  # Bazanı import edirik

/usr/local/share/cacti/include/config.php - Faylda aşağıdakı sətirləri uyğun
olaraq quraşdırırıq:
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "freebsd";
$database_port = "3306";
$database_ssl = false;


/etc/crontab faylına aşağıdakı sətiri əlavə edirik ki, 5 dəqiqədən bir poller
işə düşsün:
# Cacti Cron
*/5   *     *     *     *     root  /usr/local/bin/php
/usr/local/share/cacti/poller.php >> /usr/local/share/cacti/log/poller.log
2>&1

Öz rahatçılığımız üçün CACTI qovluğuna symlink yaradırıq və ünvana daxil
oluruq:
ln -s /usr/local/share/cacti/ /
cd /cacti
mkdir /usr/local/share/cacti/log/                # Jurnal qovluğu yaradırıq
touch /usr/local/share/cacti/log/poller.log      # Poller jurnal faylı yaradırıq
mkdir /var/log/cacti/                            # CACTI jurnal fayl üçün
                                                 qovluq yaradırıq
touch /var/log/cacti/log              # CACTI jurnal faylı yaradırıq
mkdir -p /var/db/cacti/rra/           # CACTI RRD bazası üçün qovluq yaradırıq
chown -R root:wheel /var/db/cacti/  # Bütün CACTi-e aid olan ünvanları root
                                          adından edirik(BUG)
chown -R /var/log/cacti/
```

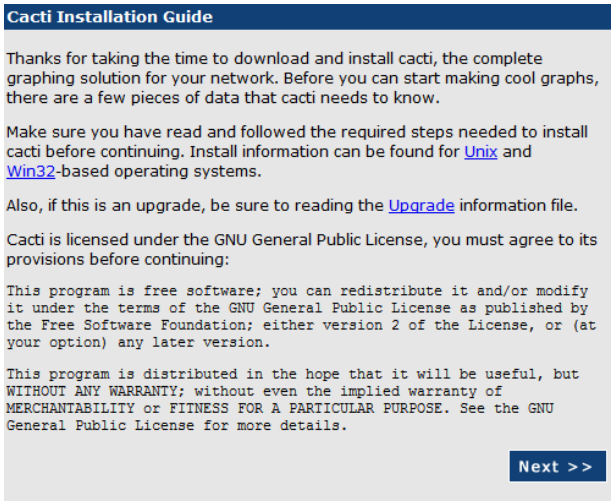
Mütləq tələb edilməyən portları yalnız rahatçılığımız üçün yükləyirik:

```
cd /usr/ports/ftp/wget
BATCH=yes make WITHOUT="IDN IPV6 NLS" install

cd /usr/ports/sysutils/screen
BATCH=yes make WITHOUT="INFO NETHACK" install

cd /usr/ports/editors/vim-lite
BATCH=yes make install
```

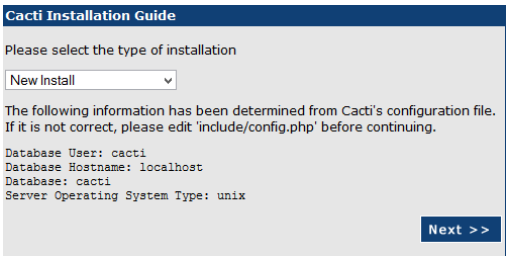
Reboot edirik və prosesləri yoxlayırıq:

```
reboot
ps aux | egrep 'httpd|snmpd|mysqld|rrdcached|sshd'
root    989   0.0  0.1  46876  3868 ??  Is   10:10AM   0:00.00 /usr/sbin/sshd
mysql 14126   0.0  0.0  14536  1896 ??  Is   11:37AM   0:00.02 /bin/sh
/usr/local/bin/mysqld_safe --defaults-extra-file=/var/db/mysql/my.cnf --
user=mysql --datadir=/var/db/
mysql 14224   0.0  1.2 267504 51632 ??  I    11:37AM   0:02.07
/usr/local/libexec/mysqld --defaults-extra-file=/var/db/mysql/my.cnf --
basedir=/usr/local --datadir=/var/db/m
root  18540   0.0  0.3 106096 12248 ??  Is   12:57PM   0:00.00
/usr/local/bin/rrdcached -s www -l /var/run/rrdcached.sock -p
/var/run/rrdcached.pid
root  34290   0.0  0.2  64684  6512 ??  S    10:47AM   0:30.41
/usr/local/sbin/snmpd -p /var/run/net_snmpd.pid
root  63849   0.0  0.2 150580  9244 ??  Ss   12:09PM   0:00.16
/usr/local/sbin/httpd -DNOHTTPACCEPT
www   63850   0.0  0.2 150580  9256 ??  S    12:09PM   0:00.01
/usr/local/sbin/httpd -DNOHTTPACCEPT
www   63851   0.0  0.2 150580  9256 ??  I    12:09PM   0:00.00
/usr/local/sbin/httpd -DNOHTTPACCEPT
www   63852   0.0  0.2 150580  9264 ??  I    12:09PM   0:00.00
/usr/local/sbin/httpd -DNOHTTPACCEPT
www   63853   0.0  0.2 150580  9256 ??  I    12:09PM   0:00.00
/usr/local/sbin/httpd -DNOHTTPACCEPT
www   63854   0.0  0.2 150580  9256 ??  I    12:09PM   0:00.00
/usr/local/sbin/httpd -DNOHTTPACCEPT
root  95844   0.0  0.1  72136  4400 ??  Ss   10:34AM   0:07.40 sshd:
root@pts/1 (sshd)
root  18547   0.0  0.0  16312  1792  1  S+   12:58PM   0:00.00 egrep
httpd|snmpd|mysqld|rrdcached|sshd
```

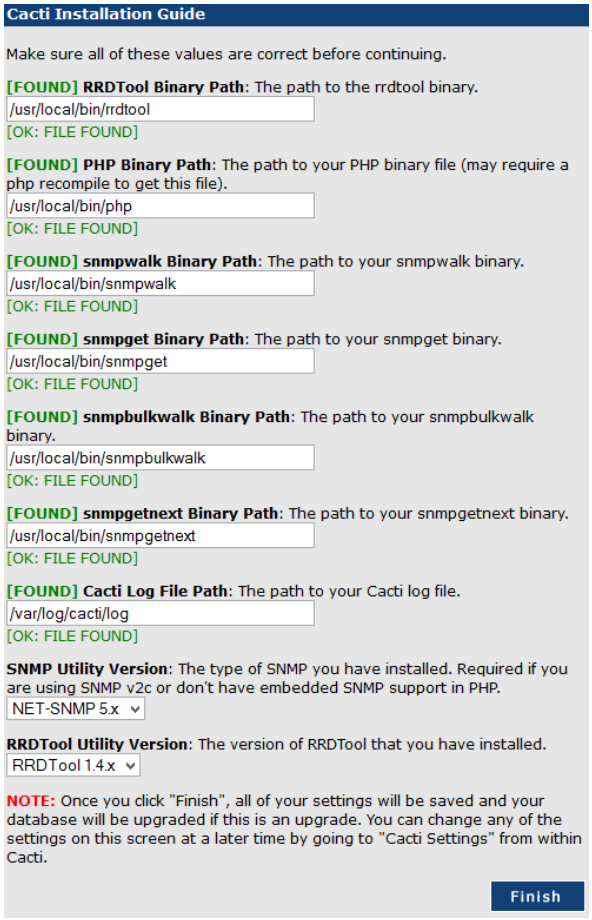
Artıq CACTi-yə webdən yetki ala bilərsiniz:
http://server-ip-address/cacti avtomatik olaraq
http://10.99.3.197/cacti/install/ səhifəsinə yönləndirəcək:



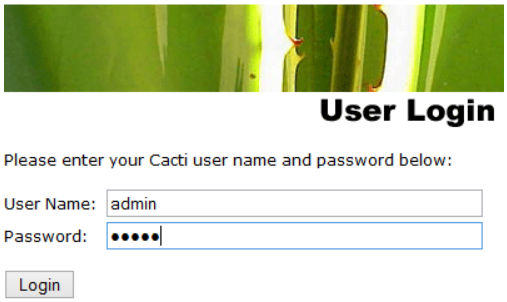
NEXT düyməsinə sıxırıq və aşağıdakı şəkil çap olunur:



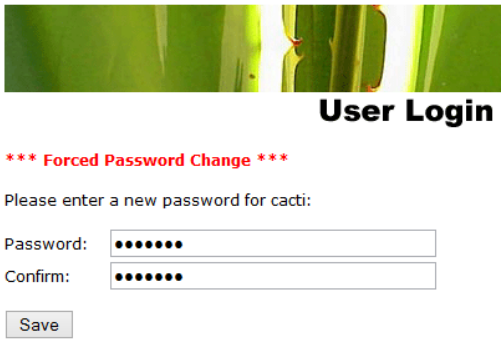
New Install seçirik və **Next** düyməsinə sıxırıq(Aşağıdakı şəkil çap edilir, RRDTool və NET-SNMP-nib versiyasını düzgün seçib **Finish** düyməsinə sıxırıq):



Aşağıdakı səhifədə susyama görə olan istifadəçi adı və şifrə **admin**-dir:



Növbəti şəkildə göstərildiyi kimi şifrəni dəyişirik:



Spine

Sürəti artırmaq üçün spine-i quraşdırırıq(**cmd.php** əvəzinə **spine** istifadə edirik):

cp /usr/local/etc/spine.conf.sample /usr/local/etc/spine.conf # Quraşdırma faylını nüsxələyirik

/usr/local/etc/spine.conf faylında aşağıdakı sətirləri uyğun olaraq quraşdırırıq.

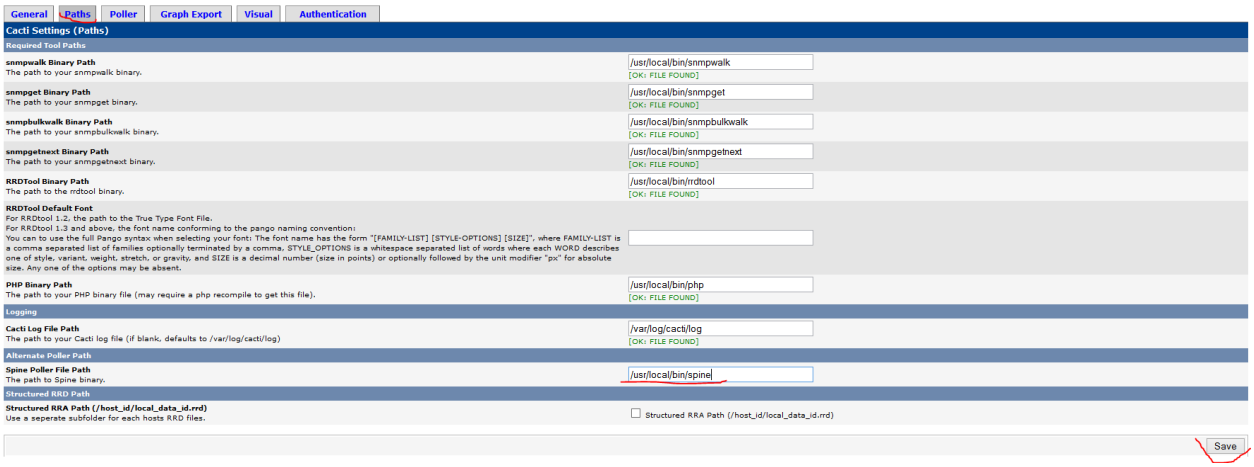
```
DB_Host          localhost
DB_Database      cacti
DB_User          cacti
DB_Pass          freebsd
DB_Port          3306
DB_PreG          0
```

ICMP ilə yoxlanış eləmək üçün SETUID yetkisini spine-a veririk:

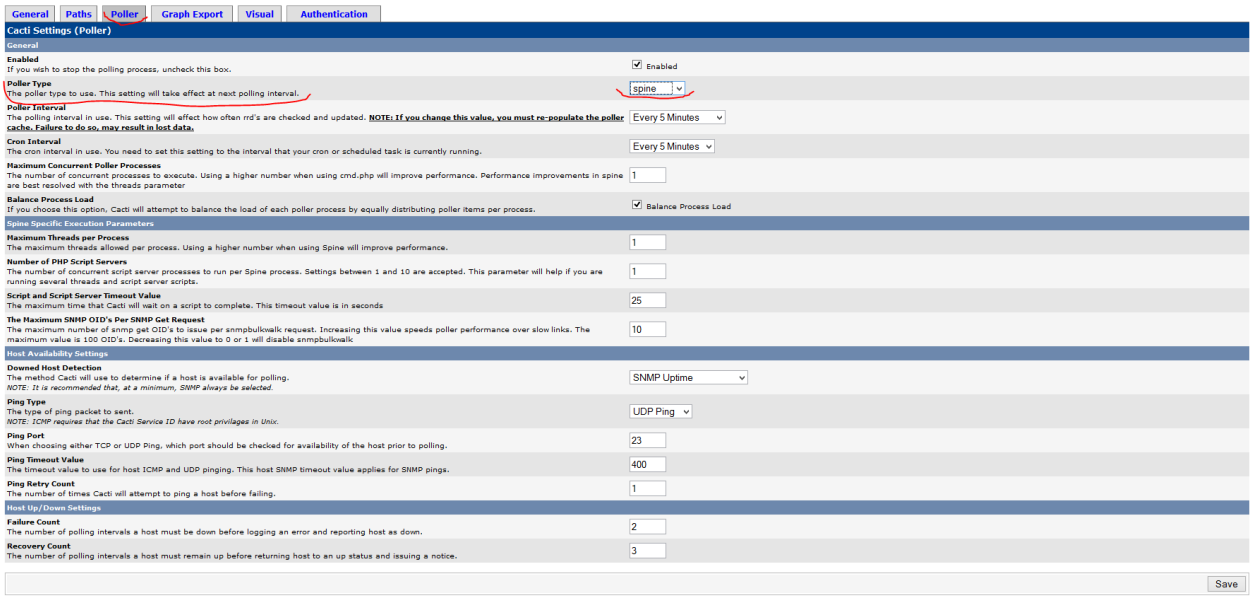
chmod +s /usr/local/bin/spine ; chown 0:0 /usr/local/bin/spine

Sonra Cacti interfeysində **Console -> Configuration -> Settings** bölümünə daxil oluruq:

Sonra **PATHS** TAB altında **Spine Poller File Path: /usr/local/bin/spine** edirik və **SAVE** düyməsinə sıxırıq:



Sonda **Poller** TAB-da **Poller Type**-i **spine** seçirik və **Save**(şəkildəki kimi):



Poller cache-in yenidən yığılması:

cmd.php-dən spine-a keçdikdən sonra qrafiklər yaranmaya bilər. Bu problem həll eləmək üçün isə CLI-dan **php /usr/local/share/cacti/cli/rebuild_poller_cache.php** əmrini yerinə yetiririk.

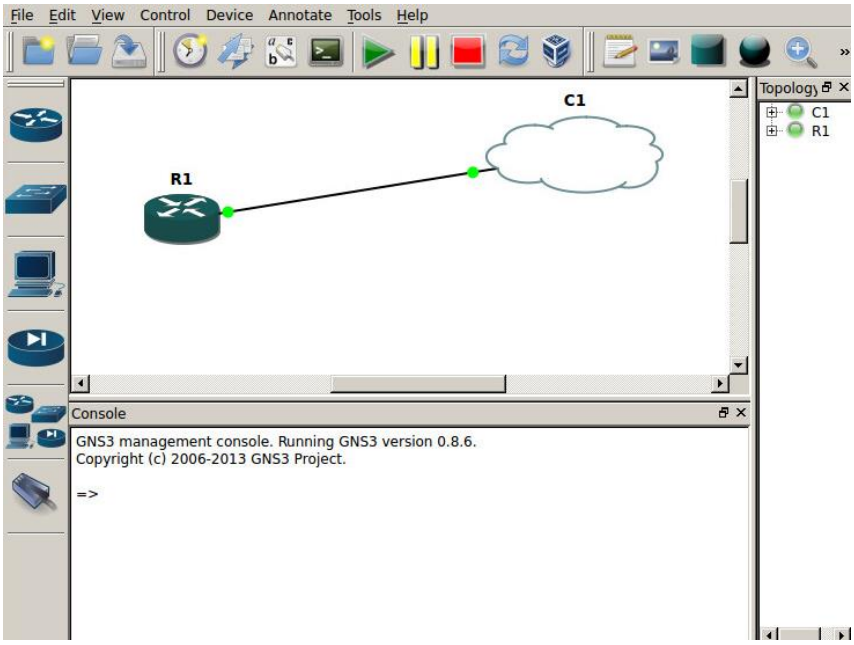
Sonra **Console -> Utilities -> System Utilities -> Rebuild Poller Cache** və **Localhost - Processes** seçirik.
Ardınca isə **Turn On Data Source Debug Mode** düyməsini sıxırıq və ekranda Data Source Debug-da görünən əmrləri CLI-dan işə salırıq(aşağıdakı kimi):

```
/usr/local/bin/rrdtool create \
/var/db/cacti/rra/localhost_proc_7.rrd \
--step 300  \
DS:proc:GAUGE:600:0:1000 \
RRA:AVERAGE:0.5:1:600 \
RRA:AVERAGE:0.5:6:700 \
RRA:AVERAGE:0.5:24:775 \
RRA:AVERAGE:0.5:288:797 \
RRA:MAX:0.5:1:600 \
RRA:MAX:0.5:6:700 \
RRA:MAX:0.5:24:775 \
RRA:MAX:0.5:288:797 \
```

Artıq 5 dəqiqədən sonra **/var/db/cacti/rra** qovluğunda aşağıdakı kimi **rrd** fayllar yaranacaq:

```
[root@cacti /var/db/cacti/rra]# ll
total 332
-rw-r--r--  1 root  wheel    46k Aug 27 00:25 localhost_users_6.rrd
-rw-r--r--  1 root  wheel    46k Aug 27 00:25 localhost_proc_7.rrd
-rw-r--r--  1 root  wheel    46k Aug 27 00:25 localhost_mem_swap_4.rrd
-rw-r--r--  1 root  wheel    46k Aug 27 00:25 localhost_mem_buffers_3.rrd
-rw-r--r--  1 root  wheel   138k Aug 27 00:25 localhost_load_1min_5.rrd
```

Indi isə GNS3-də olan Cisco Router ilə Cacti maşını qonşu olaraq quraşdıraq və nəticə alaq. GNS3 maşını Ubuntu Linux Desktop-da quraşdırılmışdır. Şəkildə görünən cloud avadanlığı Ubuntu Linux-un **eth0** şəbəkə kartı ilə bridge edilmişdir:



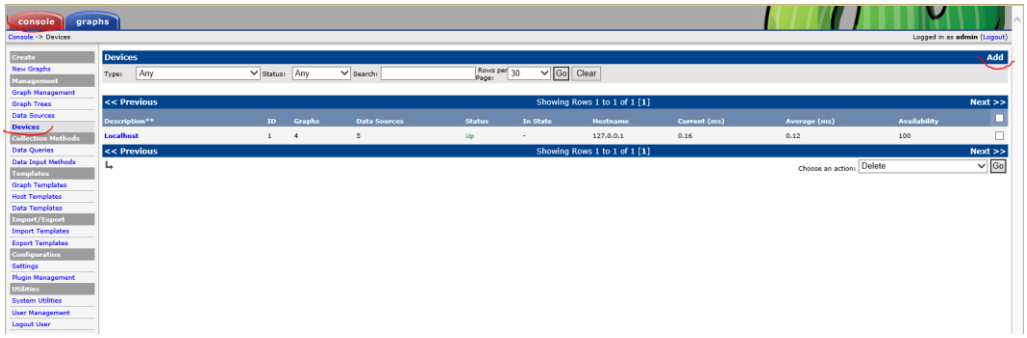
Router-imizin IP-si 10.99.3.212, Cacti maşın IP-si 10.99.3.197 və Ubuntu Desktop maşının IP-si 10.99.3.192-dir.

R1 routerimizin config-i aşağıdakı kimi olacaq:

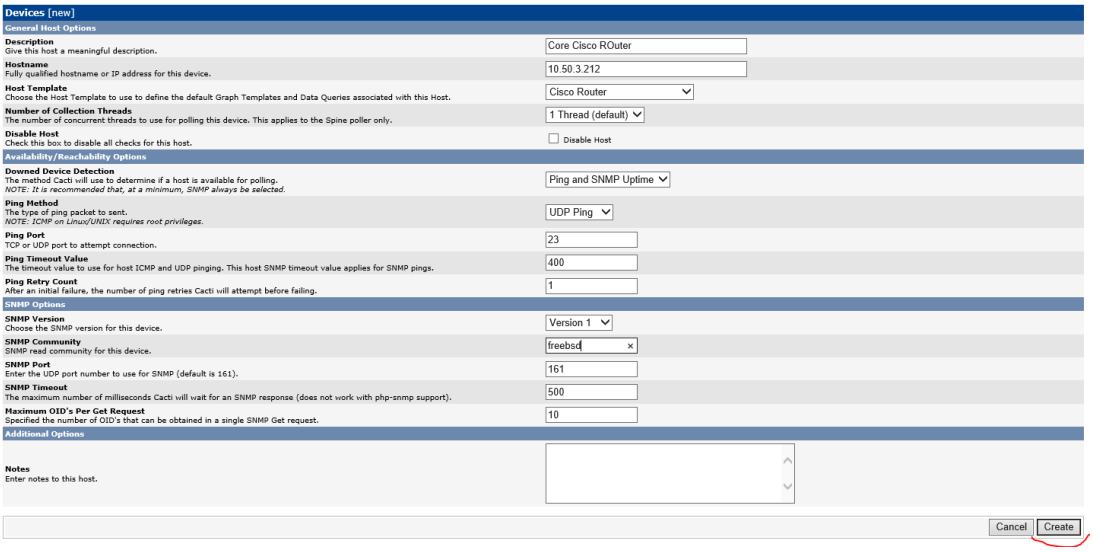
```
interface FastEthernet0/0
 ip address 10.99.3.212 255.255.255.0
 duplex auto
 speed auto
ip default-gateway 10.99.3.1
snmp-server community freebsd RO     # SNMP serverimizin community-si ilə eyni
                                     yazırıq yəni freebsd
snmp-server host 10.99.3.197 freebsd
ip name-server 10.99.3.2
ip name-server 10.99.3.3
```

Indi isə CACTI maşını Cisco router üçün quraşdıraq:

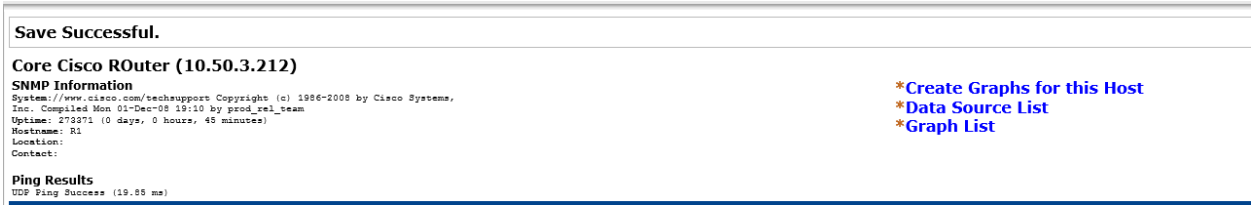
Console -> Devices -> Add



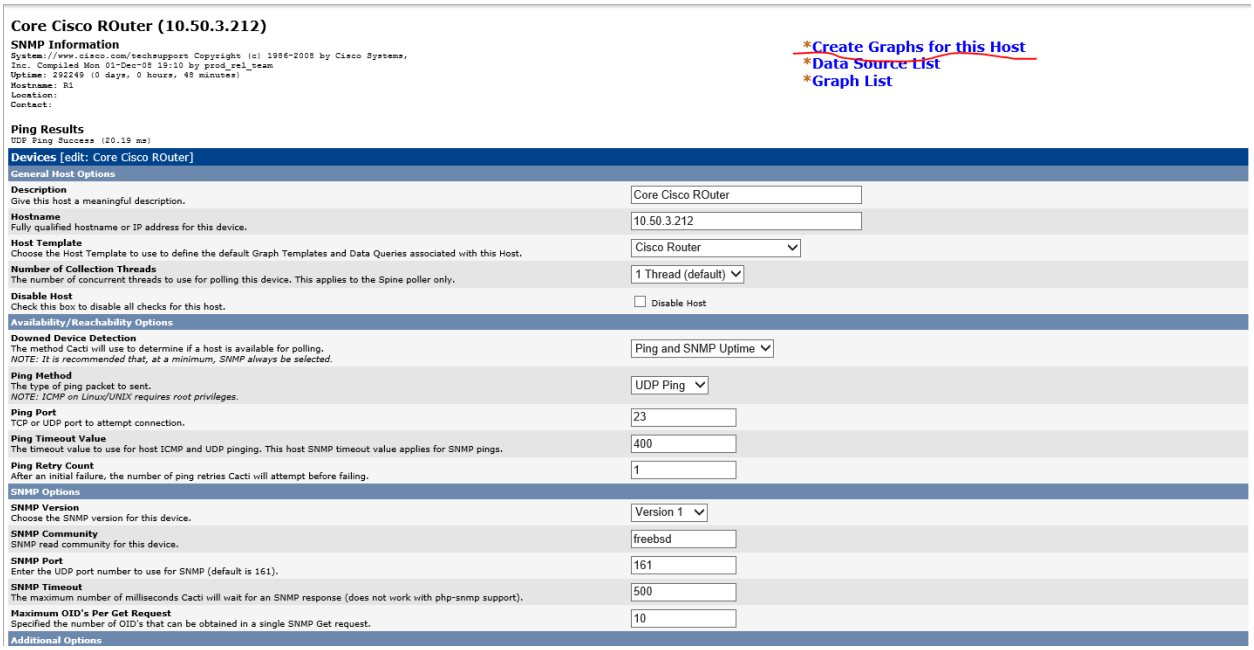
Sonra isə şəkildə göründüyü kimi quraşdırırıq və **Create** düyməsinə sıxırıq:



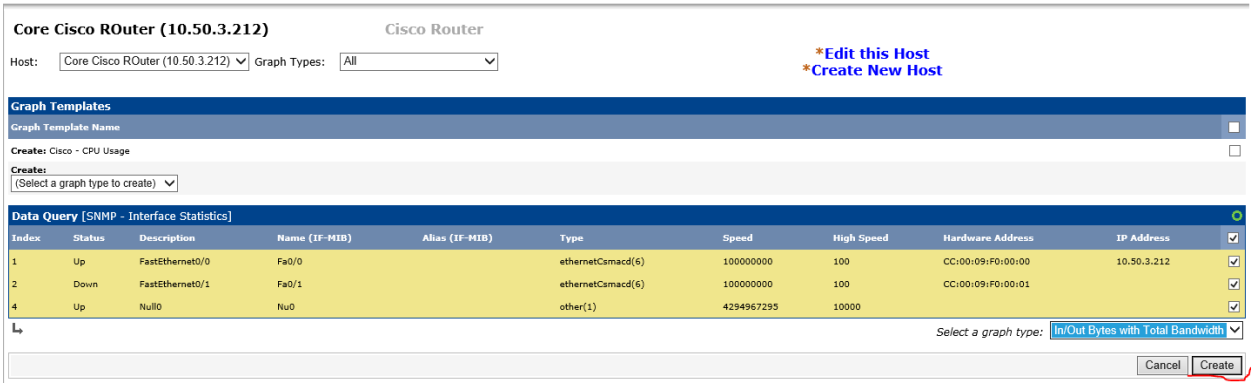
Uqurlu nəticədə aşağıdakı şəkil çap edilməlidir(**Save** düyməsinə sıxırıq):



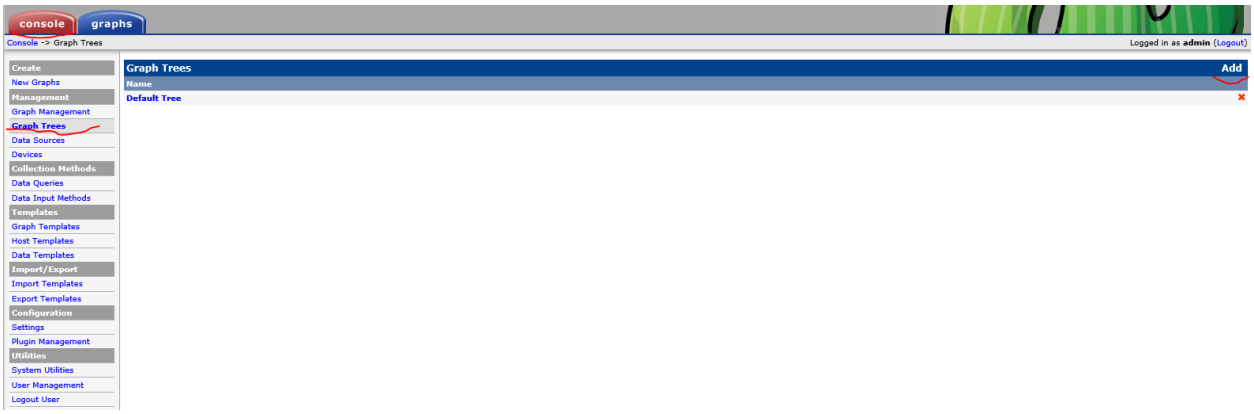
Sonra işə şəkildə göründüyü kimi **Create Graphs for this Host** düyməsinə sıxırıq:



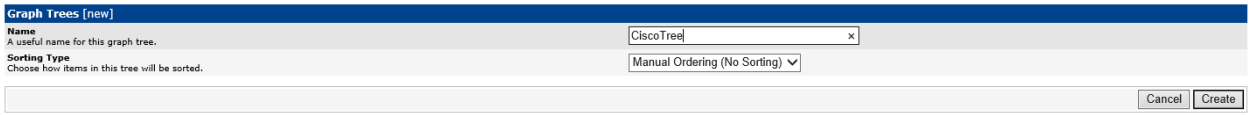
Şəkildə göründüyü kimi bütün interfeysləri seçirik və **Create** düyməsinə sıxırıq:



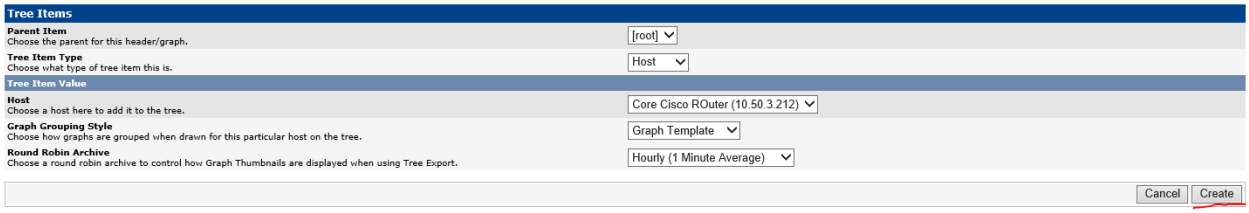
Sonra qrafikləri görmək üçün onları Cacti console-da aktivləşdiririk:
Console -> Graph Trees -> Add (Şəkildəki kimi)



Və **CiscoTree** adlı yenisini əlavə edib **Create** düyməsinə sıxırıq(Şəkildəki kimi):



Sonra **Tree Items**-də **Add** düyməsini sıxırıq və **Tree Items Type: Host** seçirik. Sonda şəkildəki kimi **create** düyməsinə sıxırıq:



Ubuntu üzərində Nagios server və client qurulması

Nagios — kompüter sistemlərinin və şəbəkələrin monitorinqi üçün nəzərdə tutulmuş açıq kodlu proqram təminatıdır. Eynilə servislərin və daxili resursların monitorinqini aparıb, təyin edilmiş şərtə əsaslanaraq email ya da sms-lə xəbərdarlıq etmək imkanına sahibdir.

Nagios əvvəlcə Netsaint adının altında Ethan Galstad tərəfindən yaradılmışdı. O bu gün sistemi komandası ilə birgə dəstəkləyir və inkişaf etdirir. Rəsmi həm də qeyri-rəsmi plaqinlərlə də məşğul olurlar.

Əvvəlcə Nagios Linux-un altında işləmək üçün hazırlanmışdı, amma o həmçinin Sun Solaris, FreeBSD, AIX və HP-UX kimi əməliyyat sistemlərində də stabil işləyir.

Öncə Ubuntu maşınımıza reposları və paketləri ən son statusa yeniləyirik.

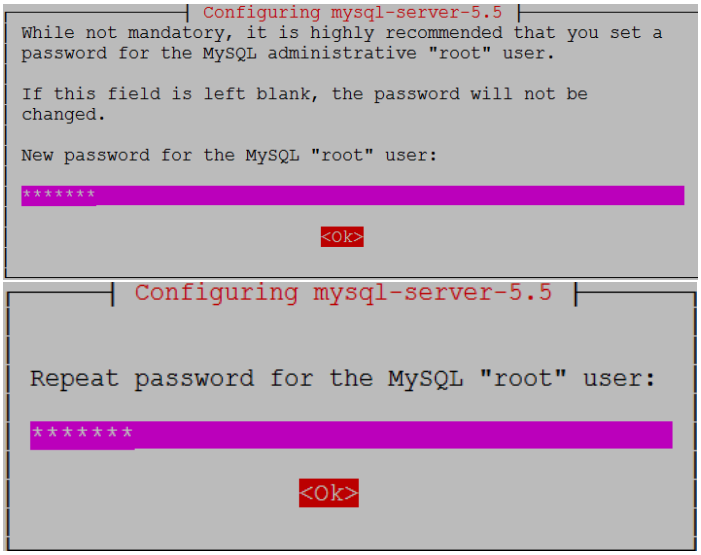
```
apt-get update                  # Reposları yeniləyirik
apt-get dist-upgrade            # Ən son paketlərə yeniləyirik
```

LAMP(Linux Apache MySQL PHP) serveri hazırlayaq.

```
apt-get install apache2         # Apache web serveri yükləyirik
ifconfig |grep "inet " | grep -v 127.0.0.1 | awk '{ print $2 }' | cut -f2 -
                                                    d":"  # Əmrlə IP-ni əldə
                                                    edirik və broswerimizdə web
                                                    serveri yoxlayırıq.
```

http://10.100.7.122/

```
apt-get install mysql-server mysql-client       # MySQL DB serveri
                                                yükləyirik(Yüklənmə müddəti
                                                aşağıdaki suallara cavab
                                                veririk)
```



apt-get install php5 php5-mysql libapache2-mod-php5 # PHP5-i yükləyirik

cat /var/www/html/index.php # PHP info səhifə yaradırıq ki, test edə bilək.

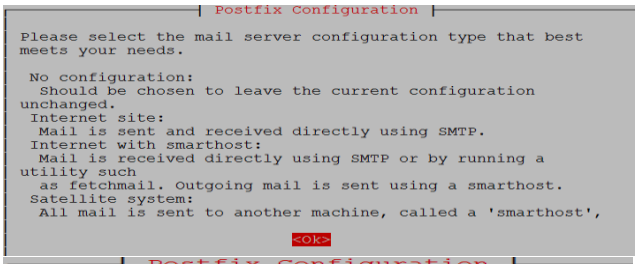
```
<?php
  phpinfo();
?>
```

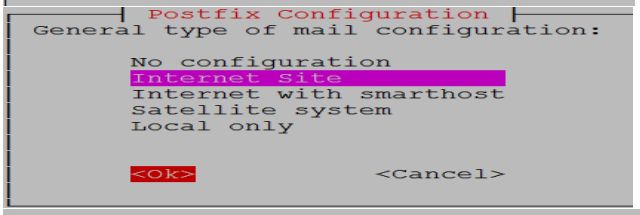
service apache2 restart # Apache2-ni restart edirik

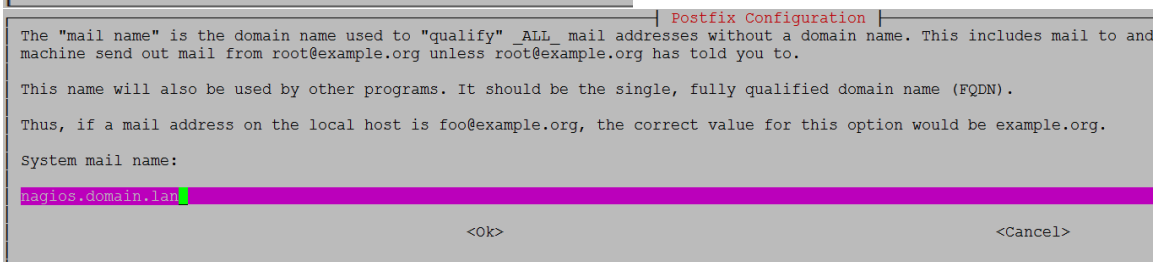
http://10.100.7.122/index.php # Səhifəyə müraciət etdikdə PHP dəyişənləri ekrana çap edilməlidir

Nagios-u yükləyək

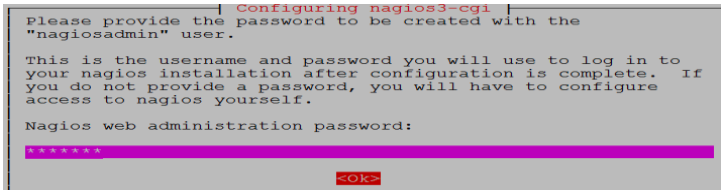
apt-get install nagios3 nagios-nrpe-plugin # Nagios və pluginlərini yükləyirik(Bu postfix-i yükləyəcək və onu aşağıdaki şəkildə quraşdıracayıq)



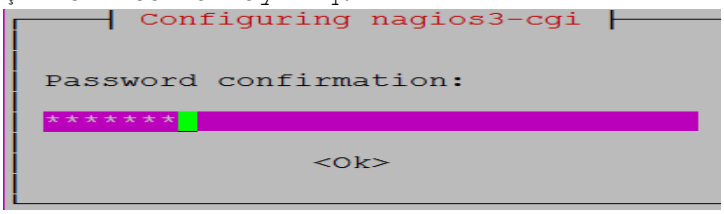




Aşağıdaki şəkildə isə Nagis WEB Interfeys üçün admin şifrəsi təyin edirik(login: **nagiosadmin**)



Şifrəni təkrarlayırıq:




```
usermod -a -G nagios www-data        # Nagios adlı istifadəçini www-data
                                     qrupuna əlavə edirik

chmod -R +x /var/lib/nagios3/        # Qovluğa yerinə yetirilmə yetkisi
                                     veririk
```

Susmaya görə Nagios kənar əmrləri qəbul eləmir. Ona görə **/etc/nagios3/nagios.cfg** faylında **check_external_commands=1** edirik.
Ümumiyyətlə **/etc/nagios3/nagios.cfg** faylı aşağıdaki kimi olacaq:

```
root@nagios:~# cat /etc/nagios3/nagios.cfg | grep -v "^$" | grep -v "#"
log_file=/var/log/nagios3/nagios.log
cfg_file=/etc/nagios3/commands.cfg
cfg_dir=/etc/nagios-plugins/config
cfg_dir=/etc/nagios3/conf.d
object_cache_file=/var/cache/nagios3/objects.cache
precached_object_file=/var/lib/nagios3/objects.precache
resource_file=/etc/nagios3/resource.cfg
status_file=/var/cache/nagios3/status.dat
status_update_interval=10
nagios_user=nagios
nagios_group=nagios
check_external_commands=1
command_check_interval=-1
command_file=/var/lib/nagios3/rw/nagios.cmd
external_command_buffer_slots=4096
lock_file=/var/run/nagios3/nagios3.pid
temp_file=/var/cache/nagios3/nagios.tmp
temp_path=/tmp
event_broker_options=-1
log_rotation_method=d
log_archive_path=/var/log/nagios3/archives
use_syslog=1
log_notifications=1
log_service_retries=1
log_host_retries=1
log_event_handlers=1
log_initial_states=0
log_external_commands=1
log_passive_checks=1
service_inter_check_delay_method=s
max_service_check_spread=30
service_interleave_factor=s
host_inter_check_delay_method=s
max_host_check_spread=30
max_concurrent_checks=0
```

```
check_result_reaper_frequency=10
max_check_result_reaper_time=30
check_result_path=/var/lib/nagios3/spool/checkresults
max_check_result_file_age=3600
cached_host_check_horizon=15
cached_service_check_horizon=15
enable_predictive_host_dependency_checks=1
enable_predictive_service_dependency_checks=1
soft_state_dependencies=0
auto_reschedule_checks=0
auto_rescheduling_interval=30
auto_rescheduling_window=180
sleep_time=0.25
service_check_timeout=60
host_check_timeout=30
event_handler_timeout=30
notification_timeout=30
ocsp_timeout=5
perfdata_timeout=5
retain_state_information=1
state_retention_file=/var/lib/nagios3/retention.dat
retention_update_interval=60
use_retained_program_state=1
use_retained_scheduling_info=1
retained_host_attribute_mask=0
retained_service_attribute_mask=0
retained_process_host_attribute_mask=0
retained_process_service_attribute_mask=0
retained_contact_host_attribute_mask=0
retained_contact_service_attribute_mask=0
interval_length=60
check_for_updates=1
bare_update_check=0
use_aggressive_host_checking=0
execute_service_checks=1
accept_passive_service_checks=1
execute_host_checks=1
accept_passive_host_checks=1
enable_notifications=1
enable_event_handlers=1
process_performance_data=0
obsess_over_services=0
obsess_over_hosts=0
translate_passive_host_checks=0
passive_host_checks_are_soft=0
check_for_orphaned_services=1
check_for_orphaned_hosts=1
check_service_freshness=1
service_freshness_check_interval=60
service_check_timeout_state=c
check_host_freshness=0
host_freshness_check_interval=60
additional_freshness_latency=15
```

```
enable_flap_detection=1
low_service_flap_threshold=5.0
high_service_flap_threshold=20.0
low_host_flap_threshold=5.0
high_host_flap_threshold=20.0
date_format=iso8601
p1_file=/usr/lib/nagios3/p1.pl
enable_embedded_perl=1
use_embedded_perl_implicitly=1
illegal_object_name_chars=`~!$%^&*|'"<>?,()=
illegal_macro_output_chars=`~$&|'"<>
use_regexp_matching=0
use_true_regexp_matching=0
admin_email=root@localhost
admin_pager=pageroot@localhost
daemon_dumps_core=0
use_large_installation_tweaks=0
enable_environment_macros=1
debug_level=0
debug_verbosity=1
debug_file=/var/log/nagios3/nagios.debug
max_debug_file_size=1000000
```

Bizim üçün yeni monitoring olunacaq host-un əlavə ediləcəyi quraşdırma ünvanı **/etc/nagios3/conf.d** qovluğudur.

Yeni client üçün quraşdırma edək(Monitoring ediləcək host - **10.100.7.57**).

```
root@nagios:/etc/nagios3/conf.d# cat /etc/nagios3/conf.d/appdevserv.cfg
define host{
        use                             generic-host
        host_name                       devapp
        alias                           devapp
        address                         10.100.7.57
        max_check_attempts              5
        check_period                    24x7
        notification_interval           30
        notification_period             24x7
}
define service {
        use                             generic-service
        host_name                       devapp
        service_description             SSH
        check_command                   check_ssh
        notifications_enabled           0
        }
define service{
        use                             generic-service
        host_name                       devapp
        service_description             CPU Load
        check_command                   check_nrpe_1arg!check_load
        }
```

```
define service{
        use                             generic-service
        host_name                       devapp
        service_description             Swap Usage
        check_command                   check_nrpe_1arg!check_swap
}
define service{
        use                             generic-service
        host_name                       devapp
        service_description             Memory Usage
        check_command                   check_nrpe_1arg!check_mem
}
define service{
        use                             generic-service
        host_name                       devapp
        service_description             Current Users
        check_command                   check_nrpe_1arg!check_users
        }
define service{
        use                             generic-service
        host_name                       devapp
        service_description             /dev/mapper/vg_developer-lv_root Free
Space
        check_command                   check_nrpe_1arg!check_hda1
        }
define service{
        use                             generic-service
        host_name                       devapp
        service_description             /dev/mapper/vg_developer-lv_home Free
Space
        check_command                   check_nrpe_1arg!check_hda2
        }
define service{
        use                             generic-service
        host_name                       devapp
        service_description             Total Processes
        check_command                   check_nrpe_1arg!check_total_procs
        }
define service{
        use                             generic-service
        host_name                       devapp
        service_description             Zombie Processes
        check_command                   check_nrpe_1arg!check_zombie_procs
        }
```

Ancaq bu yeni client işə salınmazdan öncə biz **check_nrpe** haqqında biraz danışaq. Gördüyümüz kimi **check_nrpe** quraşdırma faylında **/usr/lib/nagios/plugins/check_nrpe** əmrinin ünvanı çap edilir və bizim istənilən NRPE yüklənmiş clientler-ə göndərilən əmr kimi **check_nrpe_1arg** əmri istifadə edilir çünki, clientlərə **1** arqument ötürülür.

cat /etc/nagios-plugins/config/check_nrpe.cfg

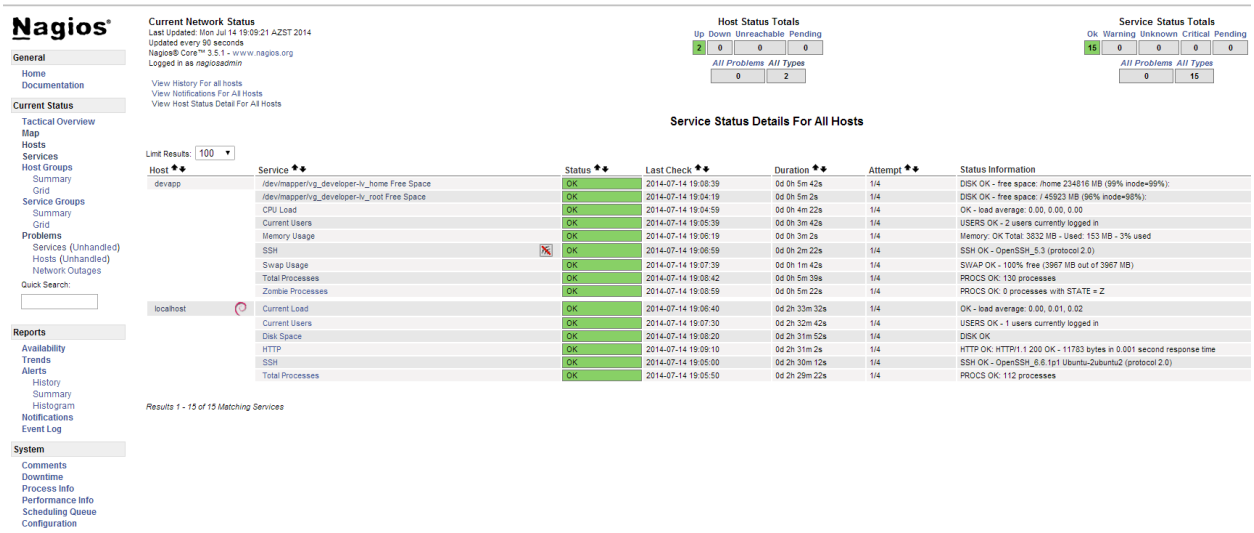
```
# this command runs a program $ARG1$ with arguments $ARG2$
define command {
        command_name    check_nrpe
        command_line    /usr/lib/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -
c $ARG1$ -a $ARG2$
}

# this command runs a program $ARG1$ with no arguments
define command {
        command_name    check_nrpe_1arg
        command_line    /usr/lib/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -
c $ARG1$
}
```

Qeyd: Client özünə lazımi paketləri yüklədikdən və servisi işə saldıqdan sonra biz serverdən client-ə müraciət yollayıb test edə bilərik. **-c**(command) və **/etc/nagios3/conf.d/appdevserv.cfg** faylında göstərilən **check_nrpe_1arg** ilə ötürülən əmrlərdən birini daxil edirik

```
/usr/lib/nagios/plugins/check_nrpe -H 10.100.7.57 -c check_hda1
/usr/lib/nagios/plugins/check_nrpe -H 10.100.7.57 -c check_swap
/usr/lib/nagios/plugins/check_nrpe -H 10.100.7.57 -c check_mem
/usr/lib/nagios/plugins/check_nrpe -H 10.100.7.57 -c check_load
/usr/lib/nagios/plugins/check_nrpe -H 10.100.7.57 -c check_users
```

```
nagios3 -v /etc/nagios3/nagios.cfg  # Serverin quraşdırmasını bu əmrlə
                                    yoxlanış edirsiniz

/etc/init.d/nagios3 start   # Client-i quraşdırdıqdan sonra isə restart edin
```

Sonda isə browserimizdə http://nagios_ip/nagios3/ daxil edirik
login: **nagiosadmin**
pass: **yuklemede_olan_shifre**



Ümumiyyətlə serverlə bağlı çıxan problemlərin hamısını **/var/log/nagios3/nagios.log** ünvanından axtarıb tapırıq.

Indi isə hansısa bir client üçün lazımi paketləri yükləyək və quraşdıraq

```
yum install -y gcc glibc glibc-common gd gd-devel make net-snmp openssl-devel
yum -y install nrpe.x86_64 nagios-plugins-nrpe.x86_64

cat /root/nagiosplugins
nagios-plugins.x86_64
nagios-plugins-check-updates.x86_64
nagios-plugins-check_sip.x86_64
nagios-plugins-all.x86_64
nagios-plugins-bonding.x86_64
nagios-plugins-by_ssh.x86_64
nagios-plugins-cluster.x86_64
nagios-plugins-dhcp.x86_64
nagios-plugins-dig.x86_64
nagios-plugins-disk.x86_64
nagios-plugins-disk_smb.x86_64
nagios-plugins-dns.x86_64
nagios-plugins-fping.x86_64
nagios-plugins-http.x86_64
nagios-plugins-icmp.x86_64
nagios-plugins-ldap.x86_64
nagios-plugins-linux_raid.x86_64
nagios-plugins-load.x86_64
nagios-plugins-log.x86_64
nagios-plugins-mailq.x86_64
nagios-plugins-mrtg.x86_64
nagios-plugins-mrtgtraf.x86_64
nagios-plugins-mysql.x86_64
nagios-plugins-nagios.x86_64
nagios-plugins-nrpe.x86_64
nagios-plugins-nt.x86_64
nagios-plugins-ntp.x86_64
nagios-plugins-ntp-perl.x86_64
nagios-plugins-nwstat.x86_64
nagios-plugins-oracle.x86_64
nagios-plugins-perl.x86_64
nagios-plugins-ping.x86_64
nagios-plugins-procs.x86_64
nagios-plugins-radius.x86_64
nagios-plugins-smtp.x86_64
nagios-plugins-snmp.x86_64
nagios-plugins-ssh.x86_64
nagios-plugins-swap.x86_64
nagios-plugins-tcp.x86_64
nagios-plugins-time.x86_64
nagios-plugins-users.x86_64
```

```
yum -y install `cat /root/nagiosplugins`


cat /etc/nagios/nrpe.cfg      # Client-in NRPE quraşdırması aşağıdaki kimi
                              olacaq
log_facility=daemon
pid_file=/var/run/nrpe/nrpe.pid
server_port=5666
nrpe_user=nrpe
nrpe_group=nrpe
allowed_hosts=127.0.0.1, 10.100.7.122     # Nagios server-ə və localhost-a
                                          izin veririk
dont_blame_nrpe=1                         # NRPE yoxlanışına izin veririk
allow_bash_command_substitution=0
debug=0
command_timeout=60
connection_timeout=300
command[check_users]=/usr/lib64/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib64/nagios/plugins/check_load -w 15,10,5 -c
30,25,20
# Client-mizin / diski
command[check_hda1]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p
/dev/mapper/vg_developer-lv_root
# Client-imizin /home
command[check_hda2]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p
/dev/mapper/vg_developer-lv_home
command[check_zombie_procs]=/usr/lib64/nagios/plugins/check_procs -w 5 -c 10
-s Z
command[check_total_procs]=/usr/lib64/nagios/plugins/check_procs -w 150 -c
200
command[check_swap]=/usr/lib64/nagios/plugins/check_swap -w 20 -c 10
command[check_mem]=/usr/lib64/nagios/plugins/check_mem -w 80 -c 90
include_dir=/etc/nrpe.d/
```

Sonda isə özümüz əlavə etdiyimiz **check_mem** scriptini öz ünvanında aşağıda göstərildiyi kimi yerləşdiririk:

```
cat /usr/lib64/nagios/plugins/check_mem
#!/bin/bash

if [ "$1" = "-w" ] && [ "$2" -gt "0" ] && [ "$3" = "-c" ] && [ "$4" -gt "0"
]; then

        memTotal_b=`free -b |grep Mem |awk '{print $2}'`
        memFree_b=`free -b |grep Mem |awk '{print $4}'`
        memBuffer_b=`free -b |grep Mem |awk '{print $6}'`
        memCache_b=`free -b |grep Mem |awk '{print $7}'`

        memTotal_m=`free -m |grep Mem |awk '{print $2}'`
        memFree_m=`free -m |grep Mem |awk '{print $4}'`
        memBuffer_m=`free -m |grep Mem |awk '{print $6}'`
        memCache_m=`free -m |grep Mem |awk '{print $7}'`
```

```
        memUsed_b=$(($memTotal_b-$memFree_b-$memBuffer_b-$memCache_b))
        memUsed_m=$(($memTotal_m-$memFree_m-$memBuffer_m-$memCache_m))

        memUsedPrc=$((($memUsed_b*100)/$memTotal_b))


        if [ "$memUsedPrc" -ge "$4" ]; then
                echo "Memory: CRITICAL Total: $memTotal_m MB - Used:
$memUsed_m MB - $memUsedPrc% used!|TOTAL=$memTotal_b;;;; USED=$memUsed_b;;;;
CACHE=$memCache_b;;;; BUFFER=$memBuffer_b;;;;"
                $(exit 2)
        elif [ "$memUsedPrc" -ge "$2" ]; then
                echo "Memory: WARNING Total: $memTotal_m MB - Used:
$memUsed_m MB - $memUsedPrc% used!|TOTAL=$memTotal_b;;;; USED=$memUsed_b;;;;
CACHE=$memCache_b;;;; BUFFER=$memBuffer_b;;;;"
                $(exit 1)
        else
                echo "Memory: OK Total: $memTotal_m MB - Used: $memUsed_m MB
- $memUsedPrc% used|TOTAL=$memTotal_b;;;; USED=$memUsed_b;;;;
CACHE=$memCache_b;;;; BUFFER=$memBuffer_b;;;;"
                $(exit 0)
        fi

else
        echo "check_mem v1.1"
        echo ""
        echo "Usage:"
        echo "check_mem.sh -w <warnlevel> -c <critlevel>"
        echo ""
        echo "warnlevel and critlevel is percentage value without %"
        echo ""
        echo "Copyright (C) 2012 Lukasz Gogolin (lukasz.gogolin@gmail.com)"
        exit
fi


chmod +x /usr/lib64/nagios/plugins/check_mem          # yerinə yetirən edirik

/etc/init.d/nrpe start              # Client-də NRPE daemon-u işə salırıq
chkconfig --level 0123456 nrpe on   # Servisi startup-a əlavə edirik
```

Əgər client Ubuntudursa **chkconfig** üçün aşağıdaki paketi yükləyirik

```
apt-get install sysv-rc-conf  # Ubuntu 14.04-də artıq chkconfig əvəzinə
                              istifadə ediləcək paket sysv-rc-conf-dir

sysv-rc-conf --list           # Bütün daemon səviyyələrinə startup üçün bu
                              əmrlə baxa bilərik

/usr/lib64/nagios/plugins/check_nrpe -H localhost     # Client-in özünü
                                                      özündə yoxlayırıq
```

Əgər client Ubuntu olarsa, onda aşağıdakı paketləri yükləyirik
apt-get install nagios-nrpe-server nagios-plugins

Eynilə **check_mem** scriptini Ubuntu üçün uyğun qovluğa nüsxələyirik və yerinə yetirilmə yetkisi veririk.
chmod +x /usr/lib/nagios/plugins/check_mem

Uyğun olaraq **nrpe.cfg** faylı aşağıdakı kimi olacaq:
```
cat /etc/nagios/nrpe.cfg | grep -v "^$" | grep -v "#"
log_facility=daemon
pid_file=/var/run/nagios/nrpe.pid
server_port=5666
nrpe_user=nagios
nrpe_group=nagios
allowed_hosts=127.0.0.1, 10.100.7.122
dont_blame_nrpe=1
allow_bash_command_substitution=0
debug=0
command_timeout=60
connection_timeout=300
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20
command[check_hda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p
/dev/sda1
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s
Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c 200
command[check_swap]=/usr/lib/nagios/plugins/check_swap -w 20 -c 10
command[check_mem]= /usr/lib/nagios/plugins/check_mem -w 80 -c 90
include=/etc/nagios/nrpe_local.cfg
include_dir=/etc/nagios/nrpe.d/
```

```
/etc/init.d/nagios-nrpe-server restart    # Sonda servisi restart edirik
```

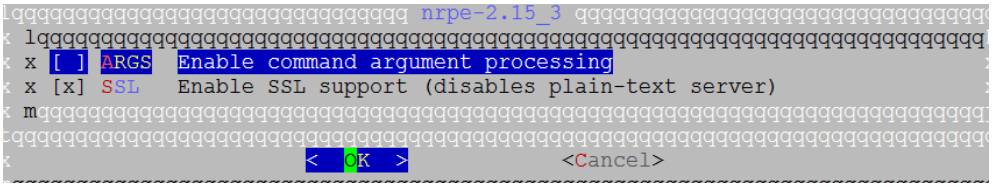
Serverimizdə Ubuntu üçün quraşdırma aşağıdakı kimi olacaq:
```
cat /etc/nagios3/conf.d/tomcat7.cfg
define host{
        use                             generic-host
        host_name                       tomcat7
        alias                           tomcat7
        address                         10.100.7.125
        max_check_attempts              5
        check_period                    24x7
        notification_interval           30
        notification_period             24x7
}
define service {
```

```
        use                             generic-service
        host_name                       tomcat7
        service_description             SSH
        check_command                   check_ssh
        notifications_enabled           0
        }
define service{
        use                             generic-service
        host_name                       tomcat7
        service_description             CPU Load
        check_command                   check_nrpe_1arg!check_load
        }
define service{
        use                             generic-service
        host_name                       tomcat7
        service_description             Swap Usage
        check_command                   check_nrpe_1arg!check_swap
}
define service{
        use                             generic-service
        host_name                       tomcat7
        service_description             Memory Usage
        check_command                   check_nrpe_1arg!check_mem
}
define service{
        use                             generic-service
        host_name                       tomcat7
        service_description             Current Users
        check_command                   check_nrpe_1arg!check_users
        }
define service{
        use                             generic-service
        host_name                       tomcat7
        service_description             /dev/sda Free Space
        check_command                   check_nrpe_1arg!check_hda1
        }
define service{
        use                             generic-service
        host_name                       tomcat7
        service_description             Total Processes
        check_command                   check_nrpe_1arg!check_total_procs
        }
define service{
        use                             generic-service
        host_name                       tomcat7
        service_description             Zombie Processes
        check_command                   check_nrpe_1arg!check_zombie_procs
        }
```

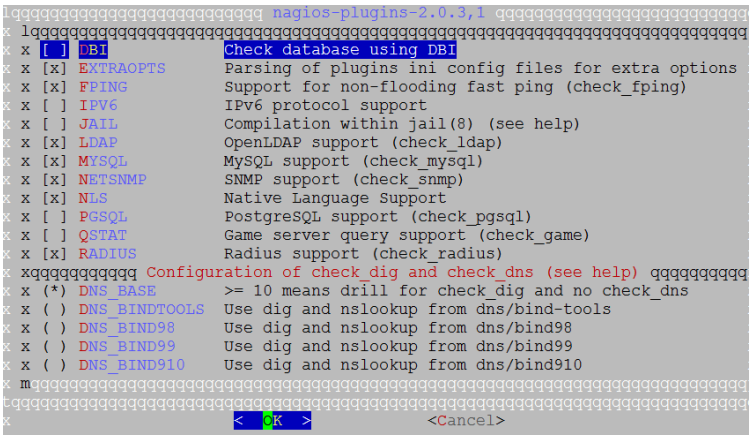
FreeBSD server üzərində NRPE agentin yüklənməsi

NRPE(Nagios Remote Plugin Executor) - Nagios agentdir uzaq maşınlarda olan scriptləri istifadə edərək onların monitorinqinə şərait yaradır. Səyəsində disklərin yüklənməsini, hal-hazırda sistemdə olan istifadəçilərin siyahısını, prosessorun və ya ram-ın yüklənməsini monitorinq eləmək olar. Nagios **check_nrpe** istifadə edərək, periodik olaraq uzaq maşında olan agent-dən məlumat alır. Nagios pluginlərinin uzaqdan digər Linux/UNIX maşınlarda yerinə yetirilməsinə NRPE şərait yaradır. **NRPE** kimi Windows agent isə **NSClient++** proqram təminatıdır.

```
cd /usr/ports/net-mgmt/nrpe
make config
```




```
make install                                    # Yükləyirik

cd /usr/ports/net-mgmt/nagios-plugins
make config
```




```
chown -R nagios:nagios /usr/local/libexec/nagios/

cp /usr/local/etc/nrpe.cfg-sample /usr/local/etc/nrpe.cfg    # Quraşdırma
                                                             faylını Sample-dan
                                                             nüsxələyirik.

echo 'nrpe2_enable="YES"' >> /etc/rc.conf        # Startup-a əlavə edirik.

ee /usr/local/etc/nrpe.cfg     # Faylda quraşdırmanı aşağıdakı qaydada edirik
log_facility=daemon
pid_file=/var/run/nrpe2/nrpe2.pid
server_port=5666
nrpe_user=nagios
nrpe_group=nagios
allowed_hosts=127.0.0.1, 10.99.7.57 # 10.99.7.57 Nagios Serverin IP-sidir.
dont_blame_nrpe=0
```

```
allow_bash_command_substitution=0
debug=0
command_timeout=60
connection_timeout=300
command[check_users]=/usr/local/libexec/nagios/check_users -w 5 -c 10
command[check_load]=/usr/local/libexec/nagios/check_load -w 15,10,5 -c
30,25,20

# Root Disk
command[check_root]=/usr/local/libexec/nagios/check_disk -w 20% -c 10% -p /

# MySQL üçün ayrılan disk
command[check_mysqldisk]=/usr/local/libexec/nagios/check_disk -w 20% -c 10% -
p /var/db/mysql

command[check_zombie_procs]=/usr/local/libexec/nagios/check_procs -w 5 -c 10
-s Z
command[check_total_procs]=/usr/local/libexec/nagios/check_procs -w 150 -c
200
command[check_swap]=/usr/local/libexec/nagios/check_swap -w 20 -c 10
command[check_mem]=/usr/local/libexec/nagios/check_mem -w 85 -c 90
```

Qeyd: Öncədən Linux **free** və **BASH** sistemdə yüklənmiş olmalıdır.

```
ee /usr/local/libexec/nagios/check_mem  # Fayla aşağıdaki məzmunu əlavə
                                          edirik.
#!/usr/local/bin/bash
#
# Script to check memory usage on Linux. Ignores memory used by disk cache.
#
# Requires the bc command
#
print_help() {
        echo "Usage:"
        echo "[-w] Warning level as a percentage"
        echo "[-c] Critical level as a percentage"
        exit 0
}

while test -n "$1"; do
        case "$1" in
                --help|-h)
                print_help
                exit 0
        ;;
                        -w)
                warn_level=$2
                shift
        ;;
                        -c)
        critical_level=$2
```

```
        shift
        ;;
                *)
                echo "Unknown Argument: $1"
                print_help
                exit 3
        ;;
        esac
                shift
        done

if [ "$warn_level" == "" ];
        then
                echo "No Warning Level Specified"
                print_help
                exit 3;
fi

if [ "$critical_level" == "" ];
        then
                echo "No Critical Level Specified"
                print_help
                exit 3;
fi

#free=`free -m | grep "buffers/cache" | awk '{print $4}'`
#used=` free -m | grep "buffers/cache" | awk '{print $3}'`

free=`/usr/local/bin/free | grep 'mem_avail:' | awk '{print $3}'`
used=`/usr/local/bin/free | grep 'mem_used:' | awk '{print $2}'`

total=$(($free+$used))

result=$(echo "$used / $total * 100" |bc -l|cut -c -2)

if [ "$result" -lt "$warn_level" ];
        then
                echo "Memory OK. $result% used."
                exit 0;
elif [ "$result" -ge "$warn_level" ] && [ "$result" -le "$critical_level" ];
        then
                echo "Memory WARNING. $result% used."
                exit 1;
elif [ "$result" -gt "$critical_level" ];
        then
                echo "Memory CRITICAL. $result% used."
                exit 2;
fi
```

```
chmod 755 /usr/local/libexec/nagios/check_mem          # 'nagios' istifadəçisi
                                                       üçün oxuma, yazma və
```

yerinə yetirilmə yetkisi veririk.

```
chown nagios:nagios /usr/local/libexec/nagios/check_mem  #'check_mem' scriptini 'nagios' istifadəçi və qrupunun üzvü edirik.
```

Linux **free**-ni FreeBSD maşınımıza yükləyək və quraşdıraq.

```
# Bizə lazım olan free paketini Internetdən dartırıq.
fetch http://www.cyberciti.biz/files/scripts/freebsd-memory.pl.txt

# adını dəyişib "free" edirik və sistem PATH-i olan "/usr/local/bin"-ə yerləşdiririk
mv freebsd-memory.pl.txt /usr/local/bin/free

chmod +x /usr/local/bin/free  # Yerinə yetirən edirik ki, əmr kimi işləsin

/usr/local/etc/rc.d/nrpe2 start     # NRPE Daemon-u işə salırıq.

/usr/local/libexec/nagios/check_nrpe2 -H localhost    # Yoxlanış aşağıdaki nəticəni verir
NRPE v2.14

/usr/local/libexec/nagios/check_nrpe2 -H localhost check_swap # Eynilə Swap yoxlanılır FreeBSD-də.
NRPE v2.14

/usr/local/libexec/nagios/check_mem -w 85 -c 90 # RAM-ı yoxlayırıq. Nəticə aşağıdaki kimidir.
Memory OK. 22% used.
```

Sonra isə Nagios serverdə yeni Clientin əlavə edilməsi procedurunu yerinə yetiririk.

İstifadə olunmuş ədəbiyyat siyahısı

1. https://en.wikipedia.org/
2. http://openssl.org/
3. http://freeradius.org/
4. http://www.xwiki.org/
5. http://www.redmine.org/
6. https://www.owncloud.org/
7. https://pyd.io/
8. http://www.dolibarr.org/
9. https://www.odoo.com/
10. https://www.google.com/
11. http://www.squid-cache.org/
12. https://openvpn.net/
13. http://www.postfix.org/
14. https://www.centos.org/
15. https://www.centos.org/
16. http://www.apache.org/
17. http://nginx.org/ru/
18. http://www.ubuntu.com/
19. http://www.oracle.com/index.html
20. https://github.com/
21. https://www.mercurial-scm.org/
22. http://bigbluebutton.org/
23. http://openmeetings.apache.org/
24. http://www.asterisk.org/
25. https://freeswitch.org/
26. http://www.tacacs.net/
27. https://www.snort.org/
28. https://www.ffmpeg.org/
29. http://www.cacti.net/
30. https://www.nagios.com/