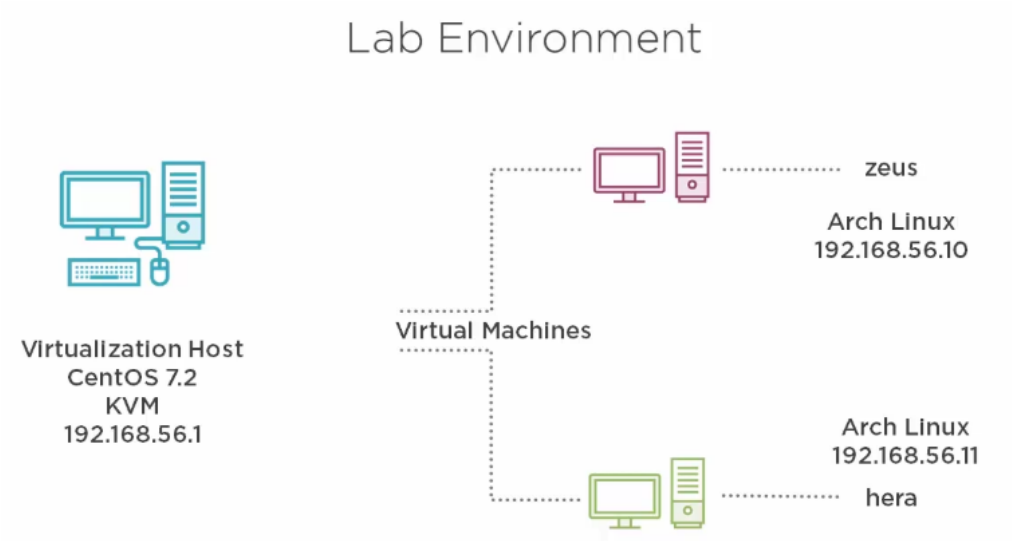
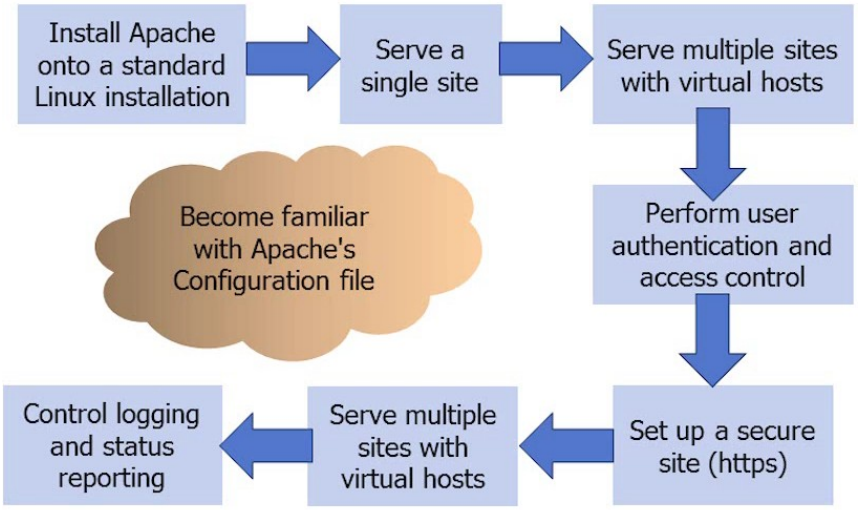


What You Will Learn in This Course



Our Scenario

- **A small lending library needs three web sites**
  - A public-facing site that anyone can use to browse the library's list of books
  - A "members only" section available only to subscribed members
  - An internal site that the librarians use to track which books are on loan and when they are due back

- **The site needs to be secure**
  - Members store personal information there

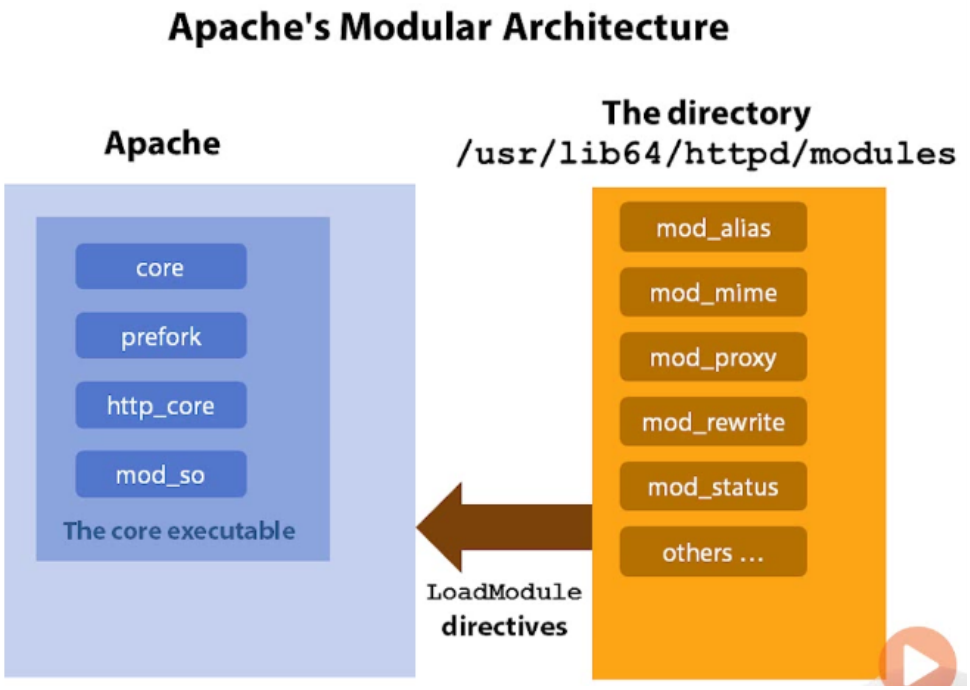
- **A second branch library needs their own site**
  - Same features
  - Different URL

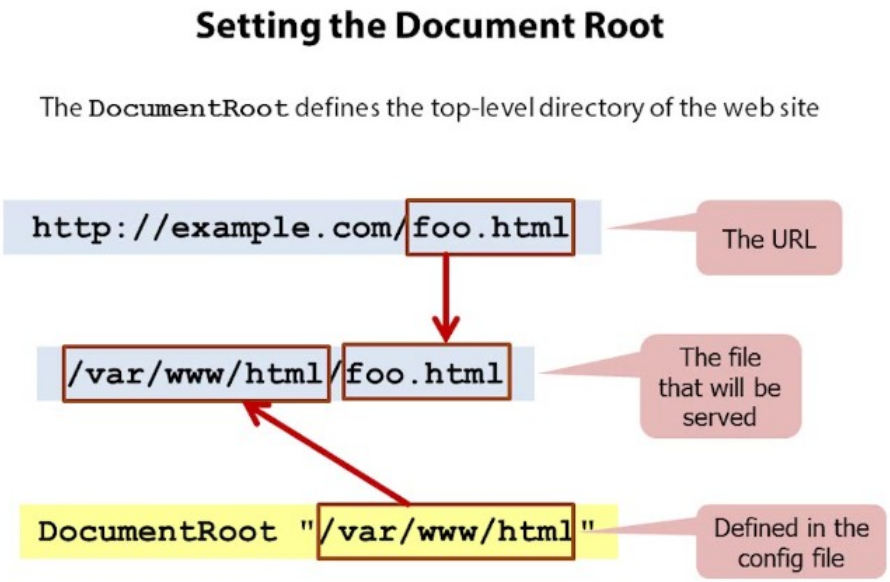
2.5.Installing Apache

[vagrant@zeus ~]$ sudo yum -y install httpd (optionally: install httpd-manual - Browse to http://localhost/manual)
[vagrant@zeus ~]$ sudo systemctl enable httpd
[vagrant@zeus ~]$ sudo systemctl start httpd
[vagrant@zeus ~]$ ss -ntl (port 80 should be open)

[vagrant@zeus ~]$ sudo su -c "echo hello > /var/www/html/index.html"
[vagrant@zeus ~]$ lynx localhost (you should see the “hello” message)

3.Apache HTTPD Minimal Configuration





3.1.Apache Configuration

httpd.conf is the main configuration file for the Apache Web Server. Located in /etc/httpd/conf/ in Centos, but Ubuntu uses /etc/apache2 directory.

```
ServerName zeus
Listen 80
User http
Group http
LoadModule dir_module modules/mod_dir.so
```

Configuration

The configuration file will contain directives. Some of which require modules to be loaded to support their operation. For example, the dir_module allows use of the DirectoryIndex option used to load the index.html where a page is not supplied in the URI.

```
$ sudo apachectl -M #Displays loaded modules
```

Modules

The Apache binary will load shared modules with the LoadModule directive. It is likely that more modules are loaded by default than are actually needed. We can reduce LoadModule and other directives to a bare minimum and then add those which we need.

3.2.Creating a Minimal Configuration

[vagrant@zeus ~]$ sudo apachectl -M

[vagrant@zeus ~]$ sudo yum install lynx tree

[vagrant@zeus ~]$ lynx localhost (you should see the “hello” message)

[vagrant@mail01 ~]$ tree /etc/httpd/
/etc/httpd/
├── conf
│   ├── httpd.conf
│   └── magic
├── conf.d
│   ├── autoindex.conf
│   ├── README
│   ├── userdir.conf
│   └── welcome.conf
├── conf.modules.d
│   ├── 00-base.conf
│   ├── 00-dav.conf
│   ├── 00-lua.conf

```
    ├── 00-mpm.conf
    ├── 00-proxy.conf
    ├── 00-systemd.conf
    └── 01-cgi.conf
├── logs -> ../../var/log/httpd
├── modules -> ../../usr/lib64/httpd/modules
└── run -> /run/httpd
```

[vagrant@zeus conf]$ grep -Ev '^\ ?\ ?\ ?\ ?#|^$' /etc/httpd/conf/httpd.conf

3.4.Testing the Apache Configuration

[vagrant@zeus conf]$ sudo apachectl configtest
Syntax OK
[vagrant@zeus conf]$ sudo systemctl restart httpd
[vagrant@zeus conf]$ sudo systemctl status httpd

[vagrant@zeus ~]$ lynx localhost (you should see the “hello” message)

[vagrant@zeus conf]$ sudo apachectl -M

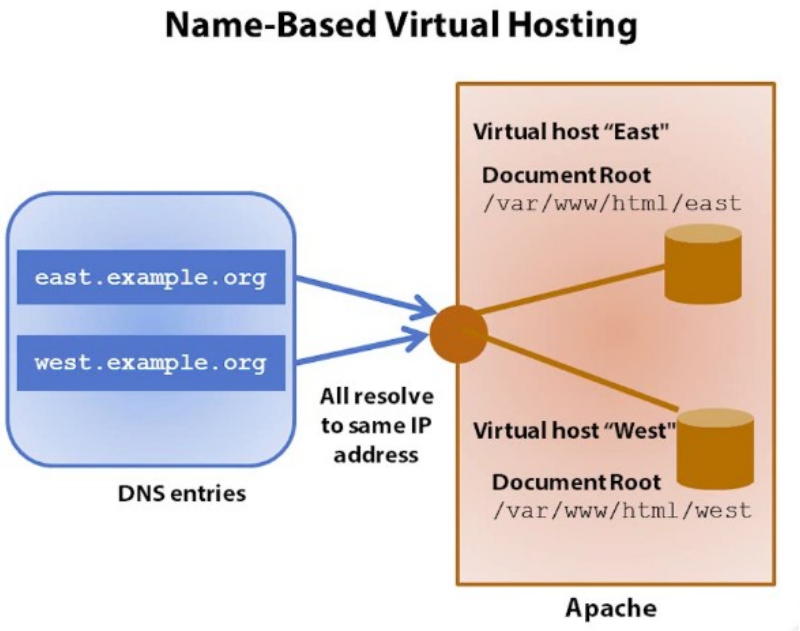
3.5.Understanding Virtual Hosts

Apache **Virtual Hosts** will allow for different DocumentRoot settings for sites accessed with different Host Names, IP Adderss or Ports.

```
<VirtualHost *:80>
  ServerName sales.example.com
  DocumentRoot /srv/vhosts/sales
</VirtualHost>
```

Virtual Host Blocks

This example shows a Named Based Virtual Host. When accessing the host sales.example.com we will be taken to the specified DocumentRoot. Whilst in Apache 2.4 it is not required Apache 2.2 and earlier need the Server directive NameVirtualHost .



How Does Apache Know Which Site to Serve?

The `host` field in the http request header identifies which site (virtual host) to serve:

```
GET http://east.example.org/ HTTP/1.1
Host: east.example.org
```

Matched against the `<VirtualHost>` containers in the configuration file

Defining a Virtual Host

Enables virtual hosting for requests arriving on port 80

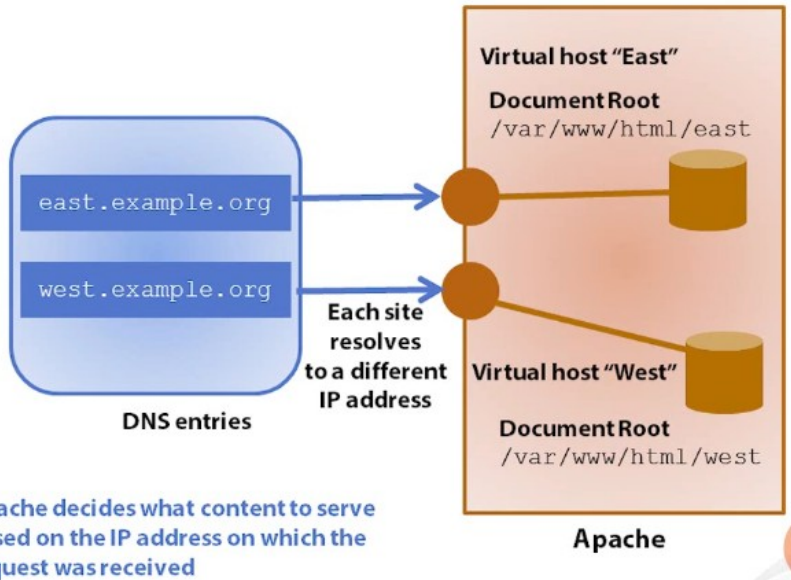
Requests that don't match any virtual hosts are served from the first one

```
NameVirtualHost    *:80

<VirtualHost *:80>
  ServerName  east.example.org
  DocumentRoot  /var/www/html/east
</VirtualHost>

<VirtualHost *:80>
  ServerName  west.example.org
  DocumentRoot  /var/www/html/west
</VirtualHost>
```

The site name

Pathname to the site's content

IP-Based Virtual Hosting



Apache decides what content to serve based on the IP address on which the request was received

3.6.Creating Named Based Virtual Hosts

[vagrant@zeus conf]$ sudo vi /etc/httpd/conf/httpd.conf

[…]
Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf

[vagrant@zeus conf]$ sudo vi /etc/httpd/conf.d/sales.frozza.com.conf
<VirtualHost *:80>
  ServerAdmin root@sales.com
  DocumentRoot "/var/www/html/sales"
  ServerName sales
  ServerAlias www.sales.com
  <Directory /var/www/html/sales>
    Require all granted
  </Directory>
</VirtualHost>

[vagrant@zeus conf]$ sudo mkdir -p /var/www/html/sales
[vagrant@zeus conf]$ sudo echo Sales > /var/www/html/sales/index.html
[vagrant@zeus conf]$ sudo apachectl configtest
Syntax OK
[vagrant@zeus conf]$ sudo systemctl restart httpd
[vagrant@zeus conf]$ sudo su -
[root@zeus ~]# echo "127.0.0.1 sales.com sales" >> /etc/hosts

[root@zeus extra]# lynx sales

[root@zeus extra]# lynx localhost (you shold see the “Sales” message, and its not good as sales.com became the default site. We need to adjust this – later).

3.7.Creating Port Based and IP Based Virtual Hosts

[root@mail01 conf.d]# vi default.frozza.com.conf
<VirtualHost *:80>
  ServerAdmin root@default.frozza.com
  DocumentRoot "/var/www/html"
  <Directory /var/www/html>
    Require all granted
  </Directory>
</VirtualHost>

[vagrant@zeus conf]$ sudo systemctl restart httpd
[vagrant@zeus conf]$ lynx mail01 → should show “hello”

[vagrant@zeus conf]$ lynx sales → should show “sales”

==========
[root@zeus extra]# vi vhost-zeus-port.conf
Listen 9000
<VirtualHost *:9000>

```
    ServerAdmin root@zeus.com
    DocumentRoot "/srv/vhosts/zeus_port"
    <Directory /srv/vhosts/zeus_port>
       Require all granted
    </Directory>
</VirtualHost>
```

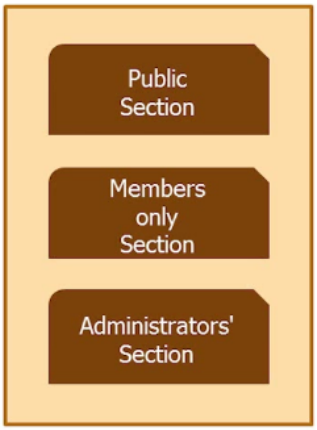
```
[root@zeus extra]# mkdir /var/www/html/9000port
[root@zeus extra]# echo port_9000 > /var/www/html/9000port/index.html

[root@zeus extra]# apachectl configtest
Syntax OK
[root@zeus extra]# systemctl restart httpd

[root@zeus extra]# lynx http://localhost:9000          → should show “zeus_9000”
```

In case of use of IP based, the server should have one IP for each vhost and it is defined in the block as “<VirtualHost 10.0.1.1:80>”.

4.Apache Access Control

Scenario

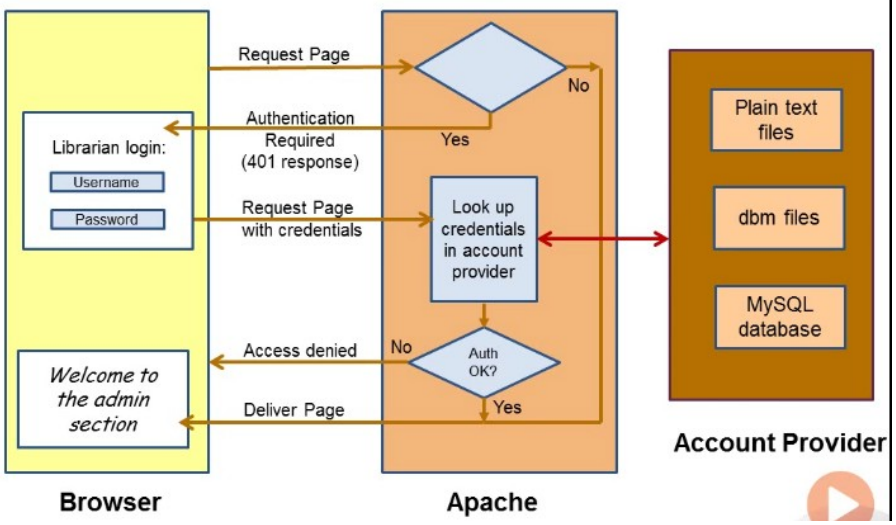
- Some parts of our library's web site are public
  - Browse the available books
- Some parts are accessible only to registered members
  - Browse their borrowing history
- Some parts are accessible only to librarians
  - Add book, add borrowers, check books out and in



Controlling Access Based on User Identity

- Access to parts of a site can be restricted
  - Require authentication and authorization of the user
- User account information can be kept in several places
  - Plain text files
  - DBM files
  - MySQL database
- Here, we'll restrict access to the "admin" section of our site to the librarians Jim and Carla
  - Since the number of users is small, we'll store them in a plain text file

How HTTP Authentication Works



Controlling Access Based on Client Machine Identity

The container directive limits the access control to a specific area of the site

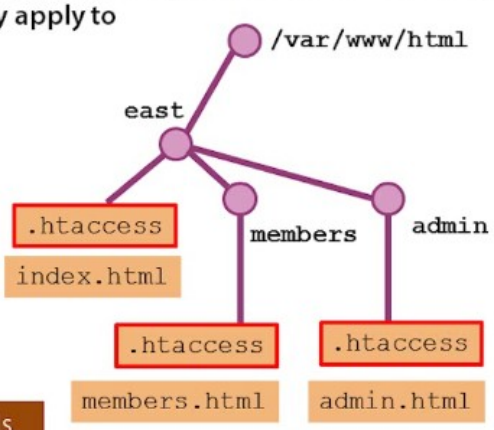
```
<Directory /var/www/html/east/admin>
  order deny,allow
  deny from all
  allow from 192.168.1.0/24
</Directory>
```

Here you can specify:
- A CIDR address block (shown here)
- A network/netmask
- A single IP address
- A full or partial domain name

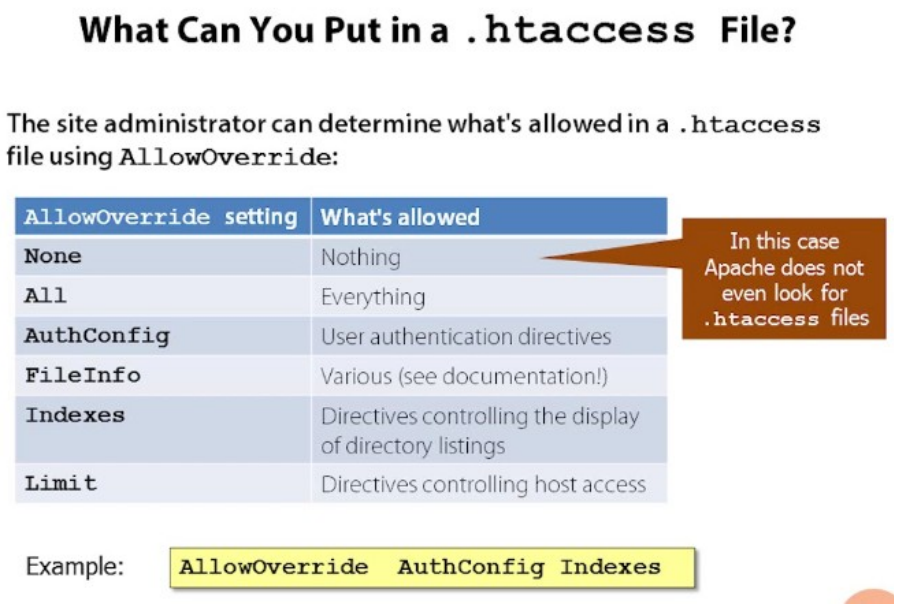
Sets the order in which the rules are applied. The *last match* wins

.htaccess Files

.htaccess files allow (some) configuration directives to be placed in the content folders they apply to



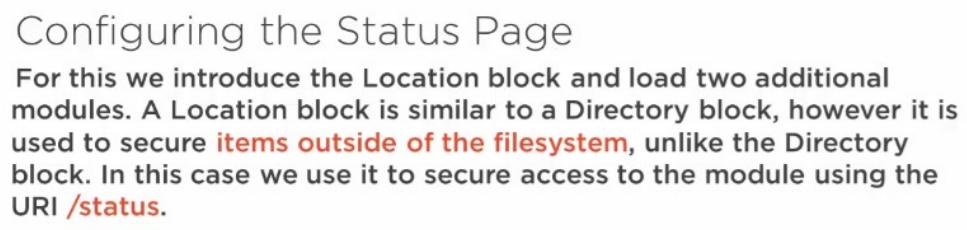
If they exist at several levels, their content "accumulates" as Apache follows the path down to the file it's serving



4.1.Introduction to Apache Authorization and Authentication

The Server Status page can be viewed via web browser when this is enabled. Access to this page may be restricted to the Server itself and perhaps your Management Workstation.

```
LoadModule status_module modules/mod_status.so

LoadModule authz_host_module modules/mod_authz_host.so

<Location "/status">
  SetHandler server-status
  Require ip 127.0.0.1
</Location>
```



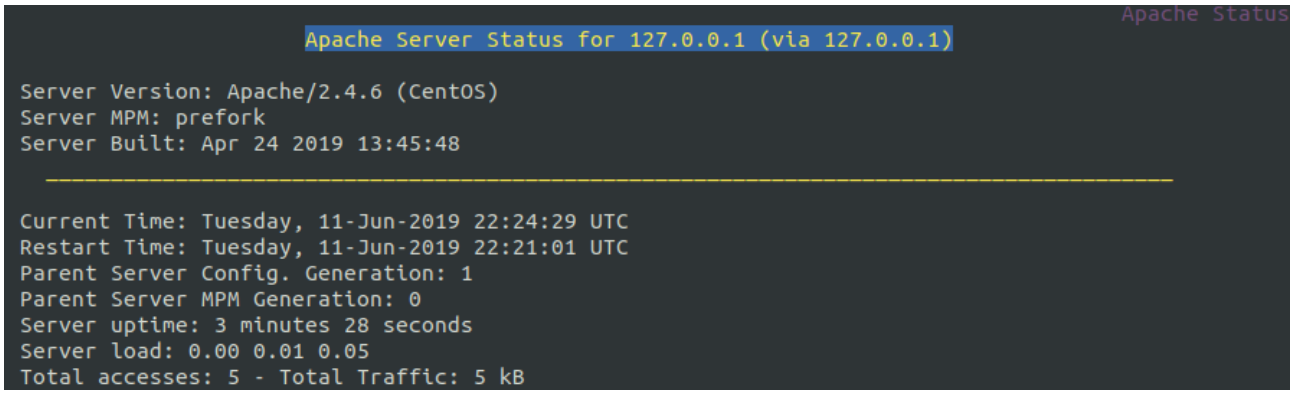
4.2.Displaying and Securing the Server Status Page

[root@mail01 conf.d]# vi 9000port.frozza.com.conf
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule status_module modules/mod_status.so
Listen 9000

```
<VirtualHost *:9000>
   ServerAdmin root@9000port.frozza.com
   DocumentRoot "/var/www/html/9000port"
   <Directory /var/www/html/9000port>
      Require all granted
   </Directory>
   <Location "/status">
      SetHandler server-status
      Require ip 127.0.0.1
   </Location>
</VirtualHost>
```

```
[root@zeus conf]# apachectl configtest
Syntax OK
```

```
[root@zeus conf]# systemctl restart httpd
[root@zeus conf]# lynx http://127.0.0.1:9000/status                (should work)
```




```
[root@zeus conf]# lynx http://10.0.0.12:9000/status          (should be denied)
```

4.3.Understanding User Authentication

```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authz_user_module modules/mod_authz_user.so
```

User Access

To be able to authenticate users we need to load an additional 3 modules as a minimum. An Authentication Type, an Authentication Provider and an Authorization module.

```
LoadModule authn_core_module modules/mod_authn_core.so
LoadModule authz_core_module modules/mod_authz_core.so
```

Core Modules

The core modules need to be loaded as well.

```
AuthType Basic
AuthName “Only authorized users allowed”
AuthUserFile /etc/httpd/sales.pwd
Require valid-user
#Require user fred joe
```

Authentication Settings

To allow user authentication to a file we can add code similar to this.

```
htpasswd -c /etc/httpd/sales.pwd fred
htpasswd /etc/httpd/sales.pwd joe
htpasswd -D /etc/httpd/sales.pwd joe
htpasswd -v /etc/httpd/sales.pwd fred
echo Password1 | htpasswd -i /etc/httpd/sales.pwd joe
```

Configure User Passwords

To manage file based users and passwords we have the htpasswd command.

4.4.Additional Modules for User Authentication

Authenticating Users

Authenticating users will require the modules:
authz_user_module
auth_basic_module
authn_file_module

The password file can be populated with the command:
htpasswd

Authorization can be either a list of users or all users:
Require user bob joe
Require valid-user

[root@zeus conf]# vi httpd.conf
[…]
LoadModule authz_core_module modules/mod_authz_core.so → does authorization
LoadModule authn_core_module modules/mod_authn_core.so → does autentication
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule mime_module modules/mod_mime.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mpm_event_module modules/mod_mpm_event.so

LoadModule unixd_module modules/mod_unixd.so
LoadModule dir_module modules/mod_dir.so
[…]

auth = authentication type (basic, file, etc)
authn = authentication provider
authz = authorization

4.5.Configuring the Virtual Host to Authenticate Users

[root@mail01 conf.modules.d]# grep autoindex /etc/httpd/conf.modules.d/*
00-base.conf:LoadModule autoindex_module modules/mod_autoindex.so

It’s usefull to list the content of a directory, for ex, a downloads area.

[root@zeus conf]# apachectl configtest
[root@zeus conf]# mkdir /var/www/html/downloads
[root@zeus conf]# cp /etc/hosts !$

[root@mail01 conf.modules.d]# tree /var/www/html
/var/www/html
├── 9000port
│   └── index.html
├── downloads
│   └── hosts
├── index.html
└── sales
    └── index.html

[root@zeus extra]# vi vhost-zeus.conf
<VirtualHost *:80>
   ServerAdmin root@zeus.com
   DocumentRoot "/srv/http"
   <Directory /srv/http>
     Require all granted
   </Directory>
   <Directory /srv/http/downloads>
     Options Indexes
     AuthType Basic
     AuthName “Only users allowed”
     AuthUserFile /etc/httpd/sales.pwd
     Require valid-user
   </Directory>
</VirtualHost>

[root@zeus **httpd**]# htpasswd -c /etc/httpd/sales.pwd fred
New Password: <fred>
Re-type new password: <fred>

No need of ‘-c’ if the file is already created.

[root@zeus **httpd**]# htpasswd /etc/httpd/sales.pwd bob

New Password: <bob>
Re-type new password: <bob>

[root@zeus **httpd**]# htpasswd -D /etc/httpd/sales.pwd bob (to delete a user)

[root@zeus **httpd**]# echo joe | htpasswd -i /etc/httpd/sales.pwd joe (useful to use it on scripts)

[vagrant@zeus conf]$ sudo systemctl restart httpd

[root@mail01 conf.d]# lynx http://localhost/downloads
(should be prompted for user and password … and allow navigation for fred and joe)

4.7.Authorizing User Groups

[root@zeus **httpd**]# vi groups
sales: fred joe

[root@zeus **httpd**]# echo jack | htpasswd -i sales.pwd jack

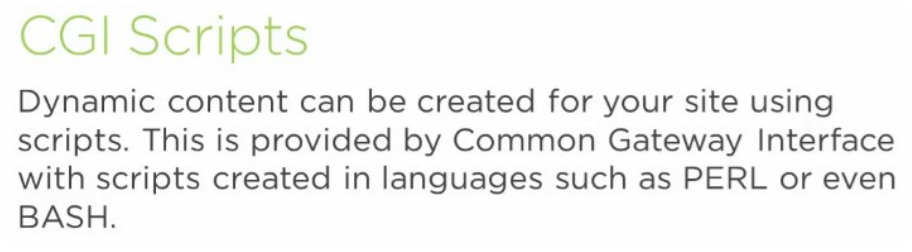
[root@zeus extra]# vi vhost-zeus.conf
<VirtualHost *:80>
ServerAdmin root@zeus.com
DocumentRoot "/srv/http"
<Directory /srv/http>
Require all granted
</Directory>
<Directory /srv/http/downloads>
Options Indexes
AuthType Basic
AuthName “Only users allowed”
AuthUserFile /etc/httpd/sales.pwd
AuthGroupFile /etc/httpd/groups
Require group sales
</Directory>
</VirtualHost>

[root@zeus conf]# vi httpd.conf
[…]
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule mime_module modules/mod_mime.so
[…]

[vagrant@zeus conf]$ sudo systemctl restart httpd

[vagrant@zeus conf]$ lynx http://zeus/downloads
(should allow navigation for fred and joe and deny for user jack)

5.Using Scripts to Deliver Dynamic Content



5.1.Dynamic Web Content

```
LoadModule alias_module modules/mod_alias.so
LoadModule cgid_module modules/mod_cgid.so
```

Adding Modules

Our Server configuration within the httpd.conf will require two additional modules. The CGI External Daemon Module allows script execution and the Alias Module to allow us to create directory aliases.

```
ScriptAlias "/cgi-bin/" "/srv/cgi-bin"
<Directory /srv/cgi-bin>
  AddHandler cgi-script .sh .pl
  Options +ExecCGI
  Require all granted
</Directory>
```

VHosts Settings

Adding a ScriptAlias allows us to have executable content in a central directory. Within the Directory block we allow the CGI module to be used with .pl and .sh files and ensure they can be executed.

```
#!/bin/bash
echo 'Content-type: text/html'
echo ''
echo "<h1>$(uname -n)</h1>"
echo "<h2>Root Filesystem</h2>"
echo "$(df -hT / | grep -v Filesystem ) "
echo "<h2>Swap Filesystem</h2>"
echo "$(swapon -s | grep -v File)"
```

A Simple Shell Script

The echo command is used to print text to the browser. The output from commands can be displayed using either back ticks or $( ). Within our example the file should be saved in the /srv/cgi-bin directory

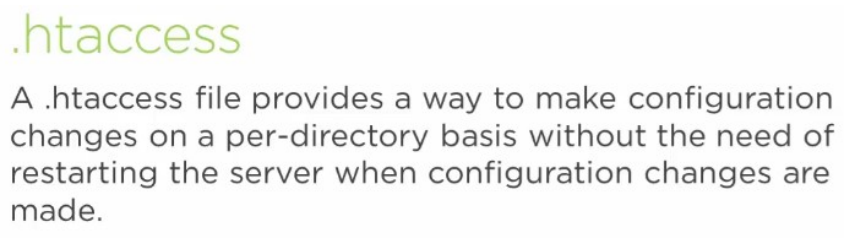
5.2.Configuring the Server to Run BASH Scripts

[root@mail01 conf.modules.d]# grep mod_alias *
00-base.conf:LoadModule alias_module modules/mod_alias.so
[root@mail01 conf.modules.d]# grep mod_cgid *
01-cgi.conf: LoadModule cgid_module modules/mod_cgid.so

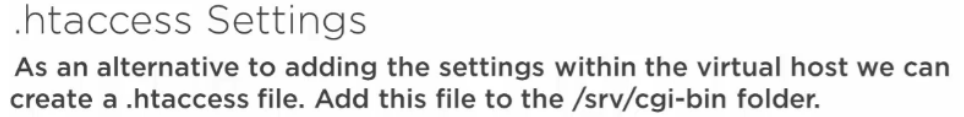
[root@mail01 conf.d]# vi default.frozza.com.conf
<VirtualHost *:80>
  ServerAdmin root@default.frozza.com
  DocumentRoot "/var/www/html"
  <Directory /var/www/html>
    Require all granted
  </Directory>
  <Directory /var/www/html/downloads>
    Options Indexes
    AuthType Basic

```
        AuthName "Only users allowed"
        AuthUserFile /etc/httpd/sales.pwd
        Require valid-user
    </Directory>
    ScriptAlias "/cgi-bin/" "/var/www/cgi-bin/"
    <Directory "/var/www/cgi-bin">
        AddHandler cgi-script .sh .pl
        Options +ExecCGI
        Require all granted
     </Directory>
</VirtualHost>

[root@mail01 conf.d]# vi /var/www/cgi-bin/report.sh
#!/bin/bash

echo 'Content-type: texxt/html'
echo ''
echo "<h1>$(uname -n)</h1>"
echo "<h2>Root Filesystem</h2>"
echo "$(df -hT / | grep -v Filesystem)"

[root@mail01 conf.d]# chmod +x !$

[root@zeus extra]# systemctl restart httpd

[root@mail01 conf.d]# lynx http://localhost/cgi-bin/report.sh
```

5.4.Understanding .htaccess Files




```
AddHandler cgi-script .sh .pl
Options +ExecCGI
Require all granted
```




```
AllowOverride none # Used where .htaccess files are not
required. Use of these files slows the server down

AllowOverride all # Where tenants are allowed complete use
of .htaccess files

AllowOverride AuthConfig Options FileInfo # Where tenants
have a restricted set of options that are allowed from
.htaccess files
```

5.5.Migrating to .htaccess Files

```
[root@mail01 conf.d]# vi default.frozza.com.conf
<VirtualHost *:80>
   ServerAdmin root@default.frozza.com
   DocumentRoot "/var/www/html"
   <Directory /var/www/html>
      Require all granted
   </Directory>
   <Directory /var/www/html/downloads>
      Options Indexes
      AuthType Basic
      AuthName "Only users allowed"
      AuthUserFile /etc/httpd/sales.pwd
      Require valid-user
   </Directory>
   ScriptAlias "/cgi-bin/" "/var/www/cgi-bin/"
   <Directory "/var/www/cgi-bin">
      AllowOverride AuthConfig Options FileInfo
    </Directory>
</VirtualHost>

[root@zeus extra]# systemctl restart httpd
[root@mail01 conf.d]# lynx http://localhost/cgi-bin/report.sh     (this should fail!)

[root@zeus extra]# vi  /var/www/cgi-bin/.htaccess
AddHandler cgi-script .sh .pl
Options +ExecCGI
Require all granted

(no need to reboot after .htaccess creation or modification)

[root@mail01 conf.d]# lynx http://localhost/cgi-bin/report.sh     (should work this time)
```

5.6.Running PERL Scripts in Apache

```
[root@zeus extra]# vi  /var/www/cgi-bin/date.pl
#!/usr/bin/perl

use strict;
use warnings;
print qq(Content-type: text/html\n\n);
print scalar localtime;

[root@zeus extra]# chmod +x /var/www/cgi-bin/date.pl
```

[root@zeus extra]# lynx http://zeus/cgi-bin/date.pl (should show the system date)

5.7.Understanding PHP and Apache

5.8.Configuring PHP

[vagrant@mail01 ~]$ sudo yum install php

10-php.conf file loads the PHP module and associate .php files with it. IMPORTANT: The php7 module does not work in threaded MPM mode so we need to run in pre-fork mode (it requires changes in httpd.conf). However, as we've installed php5, we can keed the default configuration:

[vagrant@mail01 ~]$ sudo rpm -qf /etc/httpd/conf.modules.d/10-php.conf
php-5.4.16-46.el7.x86_64

[vagrant@mail01 ~]$ cat /etc/httpd/conf.modules.d/10-php.conf
<IfModule prefork.c>
  LoadModule php5_module modules/libphp5.so
</IfModule>

Simple PHP Test Page: The phpinfo function is often used as a simple test. This canbe saved in the DocumentRoot as test.php:

[vagrant@mail01 ~]$ sudo vi /var/www/cgi-bin/test.php
<?php phpinfo(); ?>

[vagrant@mail01 ~]$ sudo systemctl restart httpd
[vagrant@mail01 ~]$ lynx http://127.0.0.1/cgi-bin/test.php (should show the php test page)

6.Securing Apache with HTTPS

6.1Using SSL and TLS on the Internet

SSL/TLS Timeline

SSLv2 1994	SSLv3 1995	TLSv1.0 1999
TLSv1.1 2006	TLSv1.2 2008	TLSv1.3 2017?

SSL or TLS

The SSL protocol is very much defunct nowadays. We can see this from the timeline. The name TLS as a replacement to SSL was more driven by politics rather than any key differences. Many people use the term SSL when in fact they are referring to the TLS protocol. Both work at the **Presentation Layer (6)** of the OSI Model.

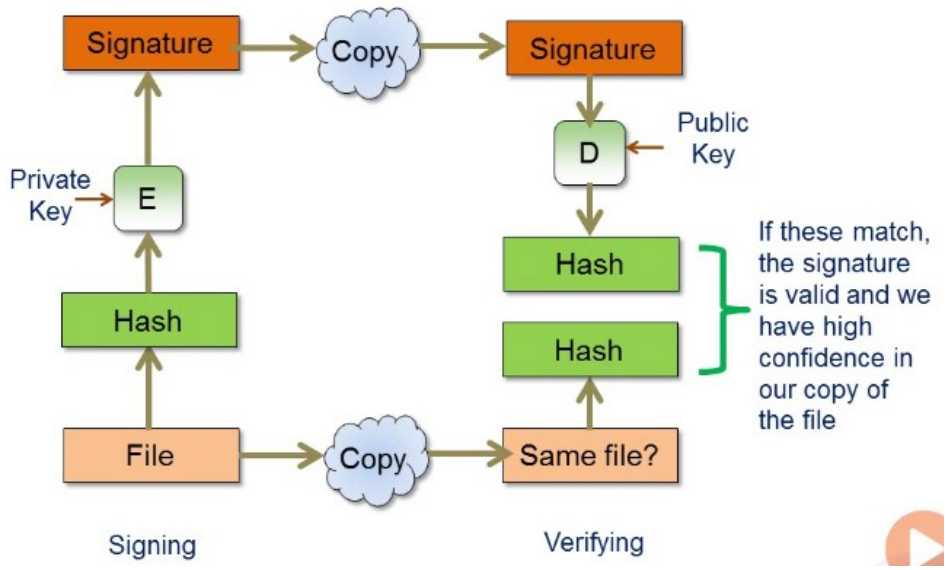
TLS Provides the CIA Triad

Confidentiality: Encryption - keeping data secure across public networks including the infamous coffee-shop WiFi

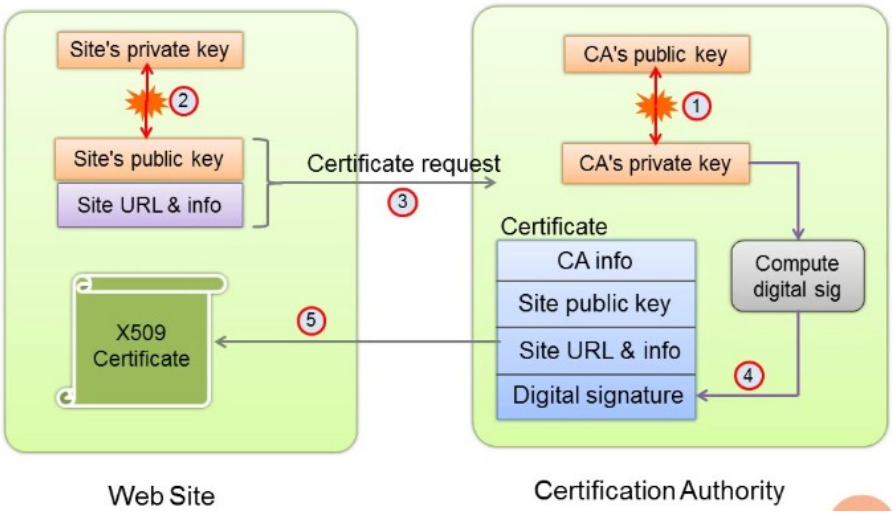
Integrity: Has the data been interfered with since leaving the server. Has my mobile network provider inserted advertisements in a third-party site that I am viewing across their network

Authenticity: Are we talking to the correct server and when I enter my password is it being sent to the server I think I am sending it to

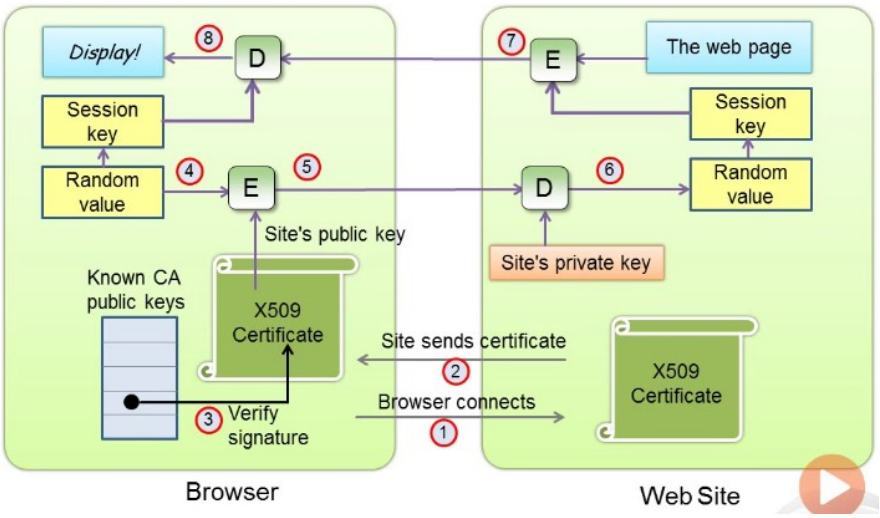
Digital Signatures



Obtaining a Digital Certificate



The SSL Connection




```
[root@trainsignal ~]# mkdir /etc/httpd/ssl
[root@trainsignal ~]# openssl req -x509 -nodes -days 365 \
> -newkey rsa:2048 -keyout /etc/httpd/ssl/apache.key \
> -out /etc/httpd/ssl/apache.crt
Generating a 2048 bit RSA private key
.....................+++
.........................+++
writing new private key to '/etc/httpd/ssl/apache.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) [XX]:uk
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:Literature
Organization Name (eg, company) [Default Company Ltd]:The Library
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:east.example.org
Email Address []:jim@east.example.org
```

6.2 Creating Private Keys

End-Entries such as HTTPS Servers will require their own puplic and private key pair. We start by generating the private key. This should be kept secure:

[root@zeus ~]# openssl version (Make sure version is later than 1.0.1f)

[root@zeus ~]# cd /etc/httpd/conf

[root@zeus conf]# openssl genrsa -out server.key 2048

[root@zeus conf]# openssl rsa -noout -text -in server.key

6.3Creating a Certificate Signing Request

Generate the CSR, or Certificate Signing Request, to creates the End-Entries Public Key. As the name suggest it will need to be signed before it is a fully functional x509 certificate. Note, that there are a relationshio between the csr and the server private key:

[vagrant@mail01 conf]$ sudo openssl req -new -key server.key -out server.csr
[...]
Country Name (2 letter code) [XX]:**BR**
State or Province Name (full name) []:**Parana**
Locality Name (eg, city) [Default City]:**Curitiba**
Organization Name (eg, company) [Default Company Ltd]:**Frozza**
Organizational Unit Name (eg, section) []:**IT**
Common Name (eg, your name or your server's hostname) []:**mail01.frozza.com**
Email Address []:**root@mail01.frozza.com**

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

[vagrant@mail01 conf]$ cat server.csr
[vagrant@mail01 conf]$ openssl req -noout -text -in server.csr

6.4 Self-signing the Public Key

Self-sign the CSR: The Private Key is also an Untrusted-CA that can be used to sign CSRs. A Self-Signed Certificate can be used just fine, but will produce Browser UI warnings:

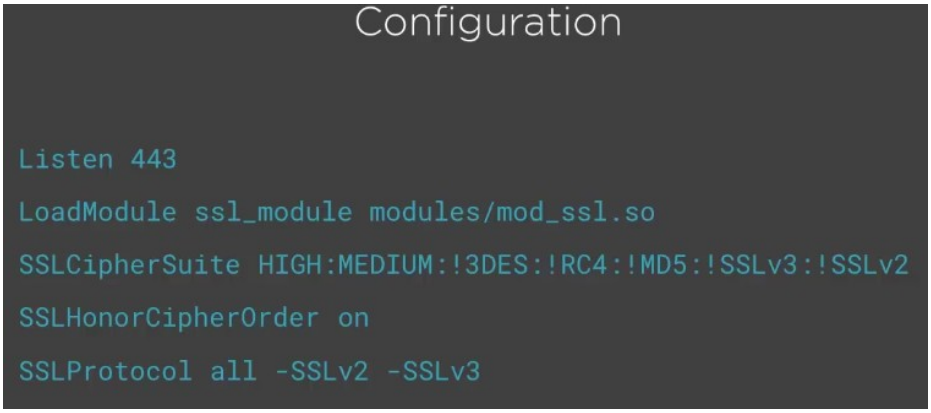
[vagrant@mail01 conf]$ sudo openssl x509 -req -sha256 -in server.csr -signkey server.key -out server.crt

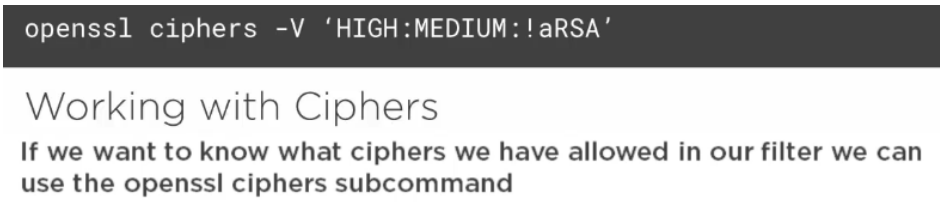
[vagrant@mail01 conf]$ sudo openssl x509 -noout -text -in server.crt

[vagrant@mail01 conf]$ ls -lrt server*
-rw-r--r--. 1 root root 1675 Jun 13 23:00 server.key
-rw-r--r--. 1 root root 1062 Jun 13 23:07 server.csr
-rw-r--r--. 1 root root 1322 Jun 13 23:15 server.crt

[vagrant@mail01 conf]$ man x509 ; man req ; man genrsa

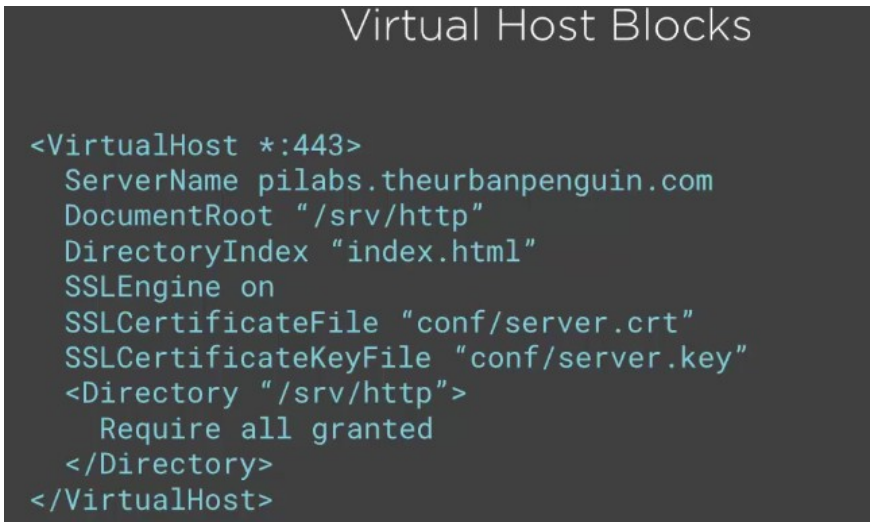
6.5Understanding CipherSuites





[root@zeus conf]# openssl ciphers
[root@zeus conf]# openssl ciphers -v HIGH

6.6Deploying Virtual Servers with TLS/SSL



[root@zeus conf]# vi httpd.conf

[…]
Listen 443
[…]
LoadModule ssl_module modules/mod_ssl.so
SSLCipherSuite HIGH:MEDIUM:!3DES:!RC4:MD5:!SSLv3
SSLHonorCipherOrder on
SSLProtocol all -SSLv2 -SSLv3

```
[…]
<VirtualHost *:80>
        ServerName pilabs.theurbanpenguin.com
        DirectoryIndex index.html
        DocumentRoot "/srv/http"
        <Directory "/srv/http">
                Require all granted
        </Directory>
</VirtualHost>

<VirtualHost *:443>
        ServerName pilabs.theurbanpenguin.com
        DocumentRoot "/srv/http"
        DirectoryIndex "index.html"
        SSLEngine on
        SSLCertificateFile "/etc/httpd/conf/server.crt"
        SSLCertificateKeyFile "/etc/httpd/conf/server.key"
        <Directory "/srv/http">
                Require all granted
        </Directory>
</VirtualHost>

[root@zeus conf]# systemctl restart httpd

[root@zeus conf]# ss -ntlm              (should see 80 and 443 open)
```

Now, you can try to open the page on browser: https://pilabs.theurbanpenguin.com

6.7Setting up the Acme Client Signing Client and Adding CA Signed Certificates to the Server

```
# mkdir -p /srv/http/.well-known/acme-challenge

# cd /etc/httpd/conf ; openssl genrsa -out le.key 2048

# acme-tiny.py --account-key le.key \
  --csr server.csr \
  --acme-dir /srv/http/.well-known/acme-challenge/ \
  > server.crt
```

To work with all this, you need to have a valid IP and registered FQDN. So, it can't be worked from the VMs in a PC.

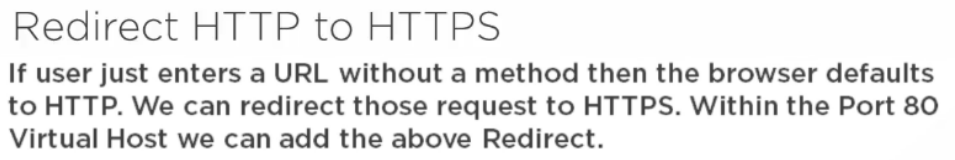
6.9Understanding the Certification Chain

```
# wget http://cert.init-x3.letsencrypt.org/ -O issue.der

# openssl x509 -in issue.der -inform DER \
  -out issue.crt -outform PEM

# cat issue.crt >> server.crt

# apachectl configtest && systemctl restart httpd
```

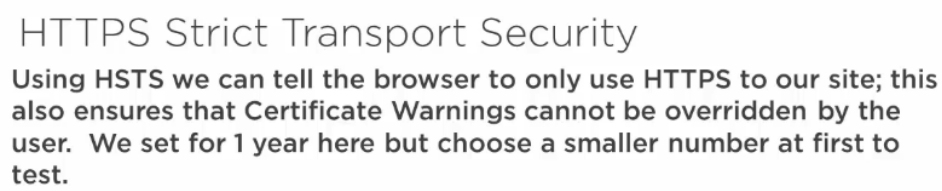
openssl x509 -noout -text -in server.crt | grep Issue (then you get the URI to do the wget)

6.10Redirecting HTTP Requests and HSTS

```
Redirect permanent / https://pilabs.theurbanpenguin.com
```




```
LoadModule headers_module modules/mod_headers.so

Header always set Strict-Transport-Security "max-
age=31536000; includeSubDomains"
```



```
[root@zeus conf]# vi httpd.conf
[…]
LoadModule headers_module modules/mod_headers.so
[…]
<VirtualHost *:80>
        ServerName pilabs.theurbanpenguin.com
        DirectoryIndex index.html
        DocumentRoot "/srv/http"
        Redirect permanent / https://pilabs.theurbanpenguin.com
        <Directory "/srv/http">
                Require all granted
        </Directory>
</VirtualHost>

<VirtualHost *:443>
        ServerName pilabs.theurbanpenguin.com
        Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
        DocumentRoot "/srv/http"
        DirectoryIndex "index.html"
        SSLEngine on
        SSLCertificateFile "/etc/httpd/conf/server.crt"
        SSLCertificateKeyFile "/etc/httpd/conf/server.key"
        <Directory "/srv/http">
                Require all granted
        </Directory>
</VirtualHost>
```

Note: Only **Redirect permanent** doesn't prevent browser warnings about certificates or man-in-the-midle attack.

```
[root@zeus conf]# systemctl restart httpd
```

Now, try to open the http page on browser: http://pilabs.theurbanpenguin.com … you should be redirected to https.

HSTS is something you set on the browser so it will not even try http connections to certain sites … those you specify to the browser.

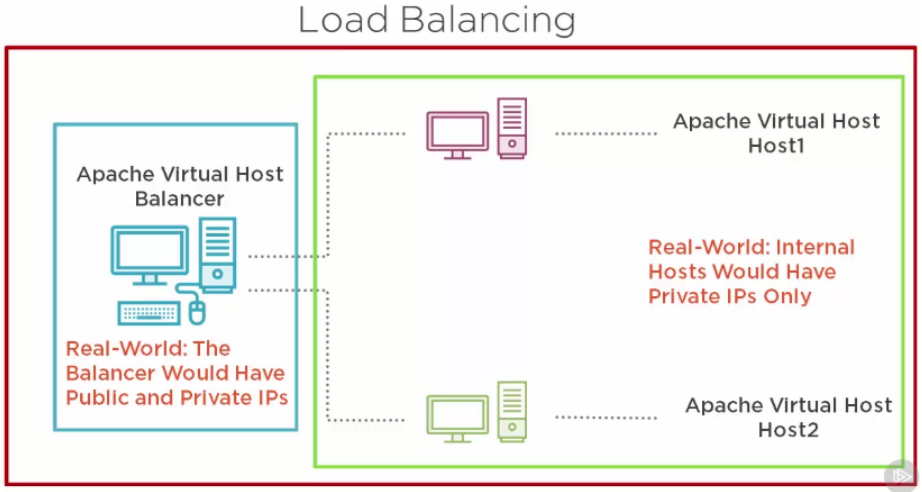
7.Load Balancing HTTP Requests

Create 3 Virtual Hosts

One Virtual Host will be used as the load balancer

The remaining two will receive the incoming requests

(Virtual Hosts are used in the demonstration to reduce the hardware required to test and setup)



Load Balancing in Apache

Apache makes use of the mod_proxy and the mod_proxy_balancer modules to load balance requests across a web server farm. All requests are received by the load balancer and sent to an available host.

```
LoadModule proxy_module modules/mod_proxy.so

LoadModule proxy_balancer_module
modules/mod_proxy_balancer.so

LoadModule proxy_http_module modules/mod_proxy_http.so

LoadModule lbmethod_byrequests_module
modules/mod_lbmethod_byrequests.so

LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
```

Adding Modules

If you are using Apache 2.2 then only the first two modules need loading. Apache 2.4 requires all of these modules to be loaded.

```
ProxyRequests off
```

Disable Standard Proxy

We do not want the server to act as a standard web proxy so we disable Proxy Requests

The LB wold be a dedicated server executing only this function, so, the configuration can be put into the httpd.conf directly (“webfarm” is just the name and can be anything):

Load Balancing Virtual Host

```
<VirtualHost *:80>
  ServerName balancer
  DocumentRoot /srv/vhosts/balancer
  <Directory /srv/vhosts/balancer>
    Require all granted
  </Directory>
  <Proxy balancer://webfarm >
    BalancerMember http://host1:80
    BalancerMember http://host2:80
    ProxySet lbmethod=byrequests
  </Proxy>
  ProxyPass "/" "balancer://webfarm/"
  ProxyPassReverse "/" "balancer://webfarm/"
</VirtualHost>
```

7.1Load Balancing in the Apache HTTPD Server

For demonstration, all 3 apache elements will be configured in one VM, the zeus server. In a more realistic situation, 3 different hosts would be used.

NOTE: The "balancer" doesn't have any content in it. Only host 1 and 2.

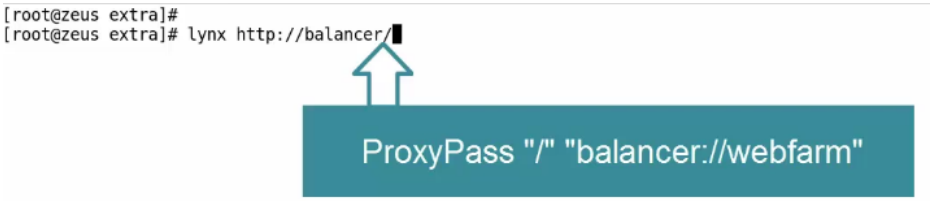
```
[root@zeus conf]# echo "127.0.0.1 balancer" >> /etc/hosts
[root@zeus conf]# echo "127.0.0.1 host1" >> /etc/hosts
[root@zeus conf]# echo "127.0.0.1 host2" >> /etc/hosts
[root@zeus conf]# mkdir /srv/vhosts/balancer
[root@zeus conf]# mkdir /srv/vhosts/host{1,2}
[root@zeus conf]# ls /srv/vhosts/
9000  balancer  host1  host2  sales
[root@zeus conf]# echo "host1" > /srv/vhosts/host1/index.html
[root@zeus conf]# echo "host2" > /srv/vhosts/host2/index.html
[root@zeus conf]# pwd
/etc/httpd/conf
[root@zeus conf]#
```

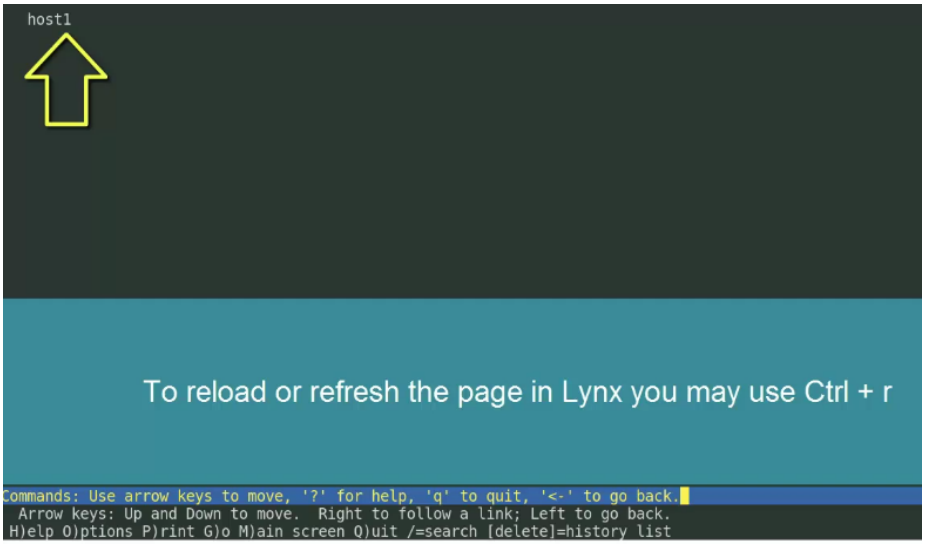
[root@zeus conf]# vi httpd.conf
[…]
Include conf/extra/proxy.conf

[root@zeus conf]# vi extra/proxy.conf
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
ProxyRequests off
<VirtualHost *:80>
    DocumentRoot /srv/vhosts/balancer
    ServerName balancer

```
        <Directory /srv/vhosts/balancer>
                Require all granted
        </Directory>
        <Proxy balancer://webfarm>
                BalancerMember http://host1:80
                BalancerMember http://host2:80
                ProxySet lbmethod=byrequests
        </Proxy>
        ProxyPass "/" "balancer://webfarm/"
        ProxyPassReverse "/" "balancer://webfarm/"
</VirtualHost>
<VirtualHost *:80>
        DocumentRoot /srv/vhosts/host1
        ServerName host1
        <Directory /srv/vhosts/host1>
                Require all granted
        </Directory>
</VirtualHost>
<VirtualHost *:80>
        DocumentRoot /srv/vhosts/host2
        ServerName host2
        <Directory /srv/vhosts/host2>
                Require all granted
        </Directory>
</VirtualHost>

[root@zeus conf]# apachectl configtest
[root@zeus conf]# systemctl restart httpd
```

7.4Testing Load Balancing





7.5Configuring the Load Balancing Manager

```
[root@zeus conf]# vi extra/proxy.conf
        <Proxy balancer://webfarm>
                BalancerMember http://host1:80
                BalancerMember http://host2:80
                ProxySet lbmethod=byrequests
        </Proxy>
        <Location /balancer-manager>
                SetHandler balancer-manager
                Require ip 127.0.0.1
        </Location>
        ProxyPass "/balancer-manager" !
        ProxyPass "/" "balancer://webfarm/"
        ProxyPassReverse "/" "balancer://webfarm/"

[root@zeus conf]# apachectl configtest
[root@zeus conf]# systemctl restart httpd
```

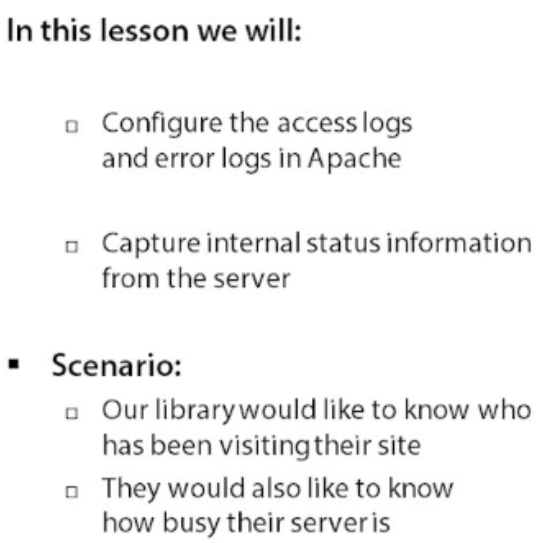
Test it a few times trough [root@zeus conf]# lynx http://manager/

Then check for statistics with the LB-manager:

[root@zeus conf]# lynx http://manager/balancer-manager

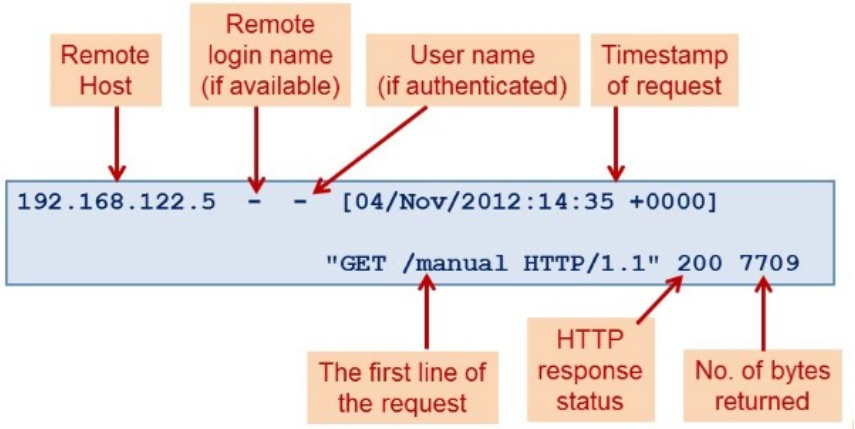
NOTE: In real life, for some web services in special, one client should remain connected to one web host only, in order to not loose any data, tracking of connection, etc. To resolve this, we use cookies stikiness.

8. Apache Logs



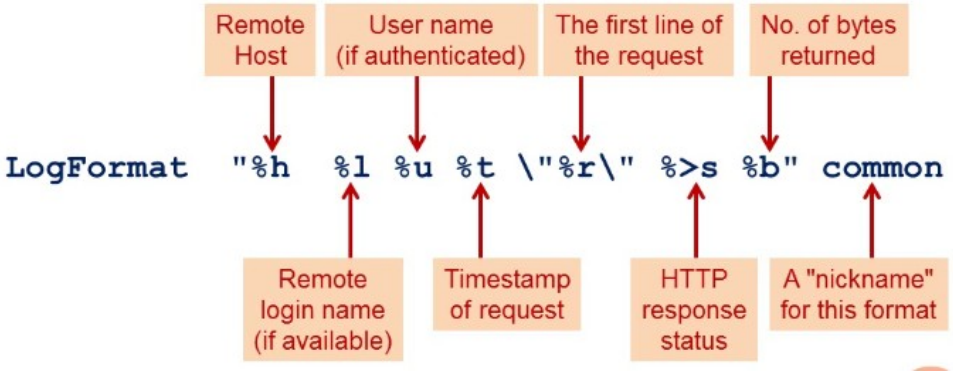
The Common Log Format

Many web servers write access logs according to an industry-standard common log format



Defining the Log Format

The `LogFormat` directive defines the format of the access logs that Apache will write



Using Custom Formats

- Apache can write multiple log formats
  - Provide a **`LogFormat`** (with a nickname) for each
  - Add a **`CustomLog`** directive referencing the nickname

```
LogFormat   "%h %t … ${Referer}i"   myformat
CustomLog   logs/referer_log   myformat
```

The Error Log

- The `ErrorLog` directives specifies where Apache will log any operational errors

Example	Description
`ErrorLog /var/log/httpd/error_log`	Writes errors to the specified file
`ErrorLog "|/bin/somecommand"`	Pipes errors to the specified program
`ErrorLog syslog:local3`	Sends errors to the **`syslog`** (or **`rsyslog`**) service using the "**`local3`**" facility name

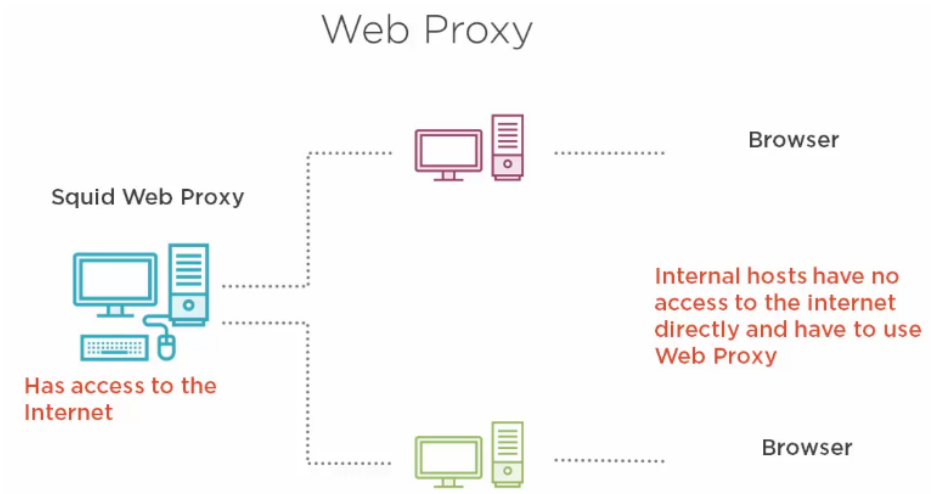
Log Levels

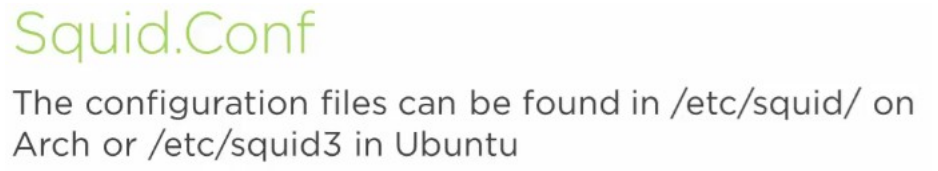
- Error messages are tagged with a *severity*
- The `LogLevel` directive determines the minimum severity level that will be logged
  - Determines verbosity of the log

debug	Debugging messages
info	Informational
notice	Normal but significant condition
warn	Warnings – no impact on operation
error	Error conditions
crit	Critical conditions – may impact operation
alert	Action should be taken immediately
emerg	System is unusable

Proxy with Squid

8.1Implementing a Web Proxy with Squid





8.2Installing Squid in Arch Linux

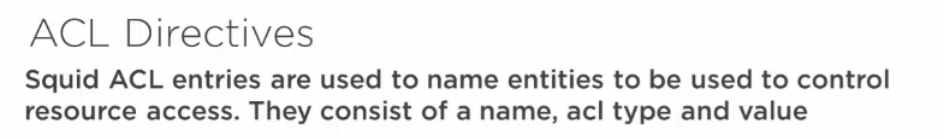
[root@zeus ~]# pacman -S squid

[root@zeus ~]# ls /etc/squid

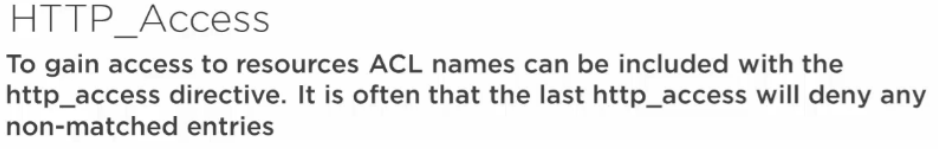
[root@zeus ~]# systemctl start squid
[root@zeus ~]# systemctl enable squid
[root@zeus ~]# systemctl satus squid

8.3Investigating the Squid Configuration

```
#Directive   Name         Type   Value
acl          localnet     src    192.168.0.0/16
acl          Safe_ports   port   80
```




```
http_access  allow localnet
http_access deny all
```



8.4Configuring Squid

[root@zeus ~]# vi /etc/squid/squid.conf
Recommended minimum configuration:
#
Example rule allowing access from your local networks.
Adapt to list your (internal) IP networks from where browsing
should be allowed
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
acl localnet src fc00::/7 # RFC 4193 local private network range
acl localnet src fe80::/10 # RFC 4291 link-local (directly plugged) machines

```
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443                 # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210                 # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280                 # http-mgmt
acl Safe_ports port 488                 # gss-http
acl Safe_ports port 591                 # filemaker
acl Safe_ports port 777                 # multiling http
acl CONNECT method CONNECT

#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost
# And finally deny all other access to this proxy
http_access deny all

# Squid normally listens to port 3128
http_port 3128

# Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/cache/squid 100 16 256

# Leave coredumps in the first cache dir
coredump_dir /var/cache/squid
```

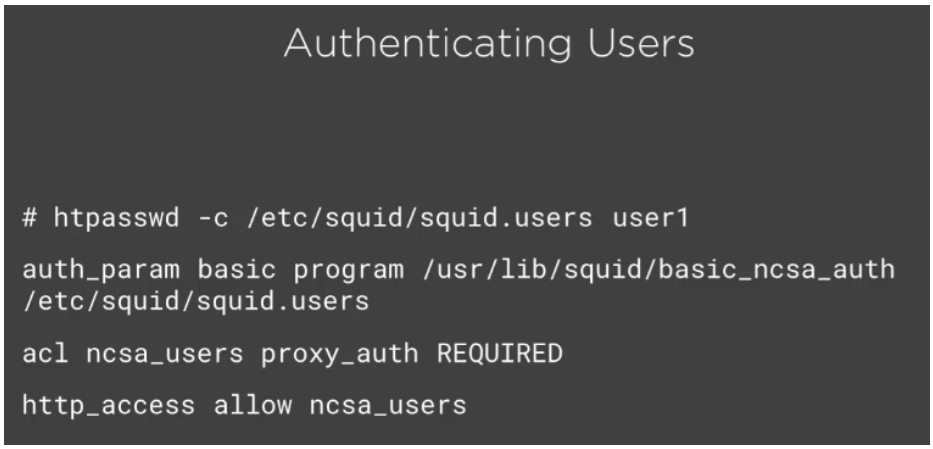
#
Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:		1440	20%	10080
refresh_pattern ^gopher:	1440	0%	1440
refresh_pattern -i (/cgi-bin/|\?) 0	0%	0
refresh_pattern .		0	20%	4320

By now, this default configuration is active but not really controling or blocking anything relevant. You can setup the proxy server to a browser and test. You can verify by checking the logs:

[root@zeus ~]# tail -f /var/log/squid/access.log

8.5Understanding User Authentication

Useful when dealing with external users (from other networks). Or diferentiate users an access privileges inside your ntwork.



8.6Authenticating Users in Squid

[root@zeus /etc/squid]# htpasswd -c squid.users user1

[root@zeus ~]# vi /etc/squid/squid.conf
Recommended minimum configuration:
#
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/squid.users
acl ncsa_users proxy_auth REQUIRED
Example rule allowing access from your local networks.
Adapt to list your (internal) IP networks from where browsing
should be allowed
acl localnet src 10.0.0.0/8	# RFC1918 possible internal network
acl localnet src 172.16.0.0/12# RFC1918 possible internal network
[...]
INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

Example rule allowing access from your local networks.
Adapt localnet in the ACL section to list your (internal) IP networks
from where browsing should be allowed

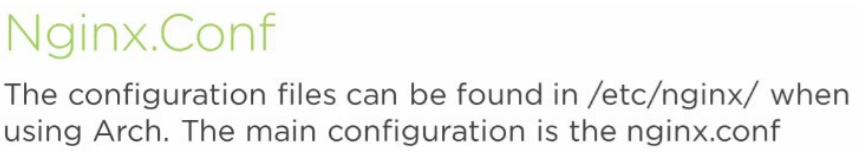
#http_access allow localnet
http_access allow localhost
http_access allow ncsa_users
And finally deny all other access to this proxy
http_access deny all
[…]

Now, if you try to access again, you should be required to authenticate with user and password!

NGINX

9.2Installing NGINX

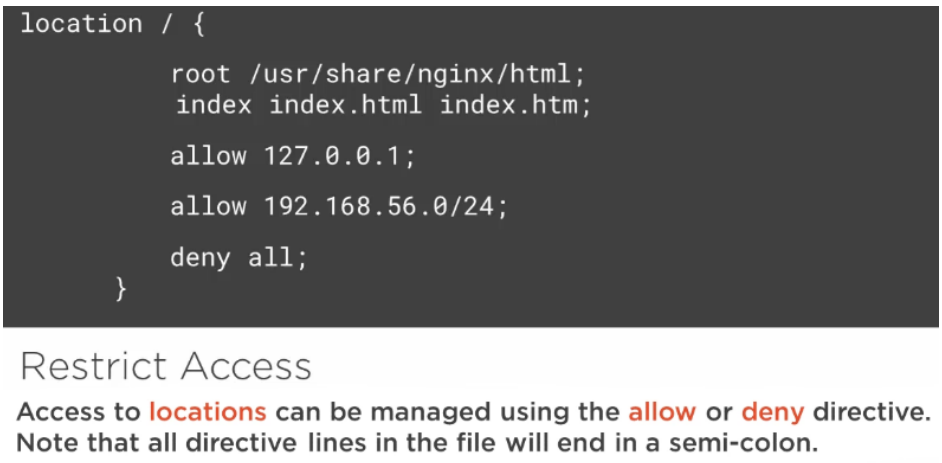
[root@hera ~]# yum install nginx



[root@hera ~]# systemctl start nginx
[root@hera ~]# systemctl enable nginx

[root@hera ~]# lynx localhost

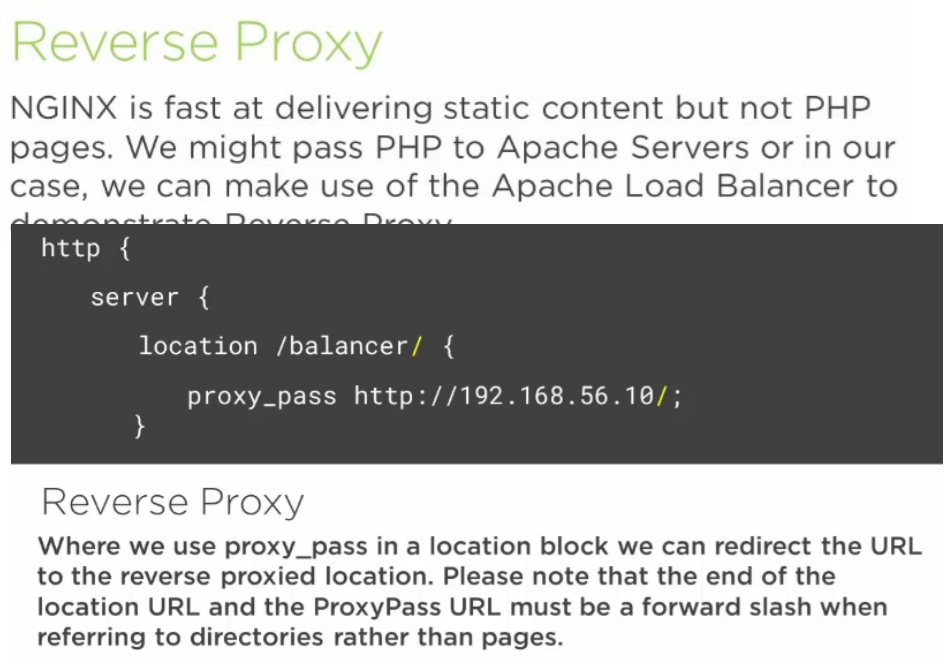
9.3Understanding the NGINX Configuration



9.4Configuring NGINX

```
[root@hera ~]# vi /etc/nginx/nginx.conf
worker_processes  1;
events {
   worker_connections  1024;
}
http {
   include       mime.types;
   default_type  application/octet-stream;[root@hera ~]#
   sendfile        on;
   keepalive_timeout  65;
   server {
      listen       80;
      server_name  localhost;
      location / {
         root   /usr/share/nginx/html;
         index  index.html index.htm;
            allow 127.0.0.1;
            allow 192.168.56.0/24;
            deny all;
      }
      error_page   500 502 503 504  /50x.html;
      location = /50x.html {
         root   /usr/share/nginx/html;
      }
   }
}
```

9.5Using NGINX as a Reverse Proxy



```
[root@hera ~]# vi /etc/nginx/nginx.conf
worker_processes  1;
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    keepalive_timeout  65;
    server {
        listen       80;
        server_name  localhost;
        location / {
            root   /usr/share/nginx/html;
            index  index.html index.htm;
                allow 127.0.0.1;
                allow 192.168.56.0/24;
                deny all;
        }
            location /balancer/ {
                    proxy_pass http://192.168.56.10/;
            }
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   /usr/share/nginx/html;
        }
    }
}

[root@hera ~]# systemctl restart nginx

[root@hera ~]# lynx 192.168.56.11

[root@hera ~]# lynx 192.168.56.11/balancer   (refresh a few times  with CTRL r)
```