Apache2

Installation

Install the package `apache2`

```
apt install apache2
```

Configuration files

The configuration folder is located in the following folder `/etc/apache2`

Explanation of the configuration files

- **apache2.conf:** This is the main configuration file for the server. Almost all configuration can be done from within this file, although it is recommended to use separate, designated files for simplicity. This file will configure defaults and be the central point of access for the server to read configuration details.
- **ports.conf:** This file is used to specify the ports that virtual hosts should listen on. Be sure to check that this file is correct if you are configuring SSL.
- **conf.d/:** This directory is used for controlling specific aspects of the Apache configuration. For example, it is often used to define SSL configuration and default security choices.
- **sites-available/:** This directory contains all of the virtual host files that define different web sites. These will establish which content gets served for which requests. These are available configurations, not active configurations.
- **sites-enabled/:** This directory establishes which virtual host definitions are actually being used. Usually, this directory consists of symbolic links to files defined in the "sites-available" directory.
- **mods-[enabled,available]/:** These directories are similar in function to the sites directories, but they define modules that can be optionally loaded instead.

Global Configuration Section

Timeout

By default, this parameter is set to "300", which means that the server has a maximum of 300 seconds to fulfill each request.

This is probably too high for most set ups and can safely be dropped to something between 30 and 60 seconds.

KeepAlive

This option, if set to "On", will allow each connection to remain open to handle multiple requests from the same client.

If this is set to "Off", each request will have to establish a new connection, which can result in significant overhead depending on your setup and traffic situation.

MaxKeepAliveRequests

This controls how many separate request each connection will handle before dying. Keeping this number high will allow Apache to serve content to each client more effectively.

Setting this value to 0 will allow Apache to serve an unlimited amount of request for each connection.

KeepAliveTimeout

This setting specifies how long to wait for the next request after finishing the last one. If the timeout threshold is reached, then the connection will die.

This just means that the next time content is requested, the server will establish a new connection to handle the request for the content that make up the page the client is visiting.

Enabling Sites and Modules in Apache

Once you have a Virtual Host file that meets your requirements, you can use the tools included with Apache to transition them into live sites.

To automatically create a symbolic link in the "sites-enabled" directory to an existing file in the "sites-available" directory, issue the following command:

```
a2ensite website_config_file
```

After enabling a site, issue the following command to tell Apache to re-read its configuration files, allowing the change to propagate:

```
service apache2 reload
```

There is also a companion command for disabling a Virtual Host. It operates by removing the symbolic link from the "sites-enabled" directory:

```
a2dissite website_config_file
```

Again, reload the configuration to make the change happen:ç

```
service apache2 reload
or
systemctl reload apache2
```

Modules can be enabled or disabled by using the "a2enmod" and "a2dismod" commands respectively. They work in the same way as the "site" versions of these commands.

Remember to reload your configuration changes after modules have been enabled or disabled as well.

SSL Module

```
a2enmod ssl
```

Redirect http request to https

```
<IfModule mod_ssl.c>
    <VirtualHost 127.0.0.1:80>
        ServerName midominio.com
        Redirect / https://midominio.com/
    </VirtualHost>
    ...
</IfModule>
```

LDAP Module

Enable the module

```
a2enmod authnz_ldap
```

Config directory `/var/web/intranet` to be access by an ldap valid user and password

```
<Directory /var/web/intranet>
            Options Indexes FollowSymLinks
            AllowOverride None

            AuthName "LDAP Authentication"
            AuthType Basic
            AuthBasicProvider ldap
            AuthLDAPURL ldap://ldap.aragon.local/dc=aragon,dc=local?uid?sub?
(objectClass=*)
            Require valid-user
</Directory>
```

The Require Directives

Require ldap-user

The Require ldap-user directive specifies what usernames can access the resource.

```
Require ldap-user "Barbara Jenson"
Require ldap-user "Fred User"
Require ldap-user "Joe Manager"
```

If the uid attribute was used instead of the cn attribute in the URL above, the above three lines could be condensed to

```
Require ldap-user bjenson fuser jmanager
```

Require ldap-group

This directive specifies an LDAP group whose members are allowed access.

```
dn: cn=Administrators, o=Example
objectClass: groupOfUniqueNames
uniqueMember: cn=Barbara Jenson, o=Example
uniqueMember: cn=Fred User, o=Example
```

Restrict folders to Hosts

Require ip

The ip provider allows access to the server to be controlled based on the IP address of the remote client. When Require ip ip-address is specified, then the request is allowed access if the IP address matches.

A full IP address:

```
Require ip 10.1.2.3
Require ip 192.168.1.104 192.168.1.205
```

An IP address of a host allowed access

A partial IP address:

```
Require ip 10.1
Require ip 10 172.20 192.168.2
```

The first 1 to 3 bytes of an IP address, for subnet restriction.

A network/netmask pair:

```
Require ip 10.1.0.0/255.255.0.0
```

A network a.b.c.d, and a netmask w.x.y.z. For more fine-grained subnet restriction.

A network/nnn CIDR specification:

```
Require ip 10.1.0.0/16
```

Require host

The host provider allows access to the server to be controlled based on the host name of the remote client. When Require host host-name is specified, then the request is allowed access if the host name matches.

A (partial) domain-name

```
Require host example.org
Require host .net example.edu
```

Require local

The local provider allows access to the server if any of the following conditions is true:

the client address matches 127.0.0.0/8 the client address is ::1 both the client and the server address of the connection are the same This allows a convenient way to match connections that originate from the local host:

```
Require local
```

Authorization Containers

The authorization container directives `<RequireAll>`, `<RequireAny>` and `<RequireNone>` may be combined with each other and with the Require directive to express complex authorization logic.

The example below expresses the following authorization logic. In order to access the resource, the user must either be the superadmin user, or belong to both the admins group and the Administrators LDAP group and either belong to the sales group or have the LDAP dept attribute sales. Furthermore, in order to access the resource, the user must not belong to either the temps group or the LDAP group Temporary Employees.

```
<Directory "/www/mydocs">
    <RequireAll>
        <RequireAny>
            Require user superadmin
            <RequireAll>
                Require group admins
                Require ldap-group "cn=Administrators,o=Airius"
                <RequireAny>
                    Require group sales
                    Require ldap-attribute dept="sales"
                </RequireAny>
            </RequireAll>
        </RequireAny>
        <RequireNone>
            Require group temps
            Require ldap-group "cn=Temporary Employees,o=Airius"
        </RequireNone>
    </RequireAll>
</Directory>
```

Configuration examples

www.aragon.local

Parte01

- Sitio web público, debe ser accesible desde cualquier sitio, pero sólo mediante el protocolo HTTPS, por lo que las peticiones HTTP serán redirigidas a HTTPS.
- Utilizará el primer certificado creado en la tarea anterior (su "Common Name" es www.CCAA.com).
- Cuando se accede a un directorio del sitio el recurso que se abrirá automáticamente será "main.html". Si no existe ese recurso, se abrirá "main2.html". Si no existiese ninguno de los dos, mostrará el contenido del directorio.
- Este sitio estará en la ruta /var/web/www. Crea los directorios y archivos necesarios para demostrar que funciona.

Parte02

- Cada usuario LDAP puede tener su propia página web, que estará ubicada en la ruta /home/ldap/usuario/www.
- Mediante navegador web, se accederá a ella utilizando la ruta www.CCAA.com/~usuario.
- Sólo será accesible desde la red 192.168.z.0/24.
- Cuando se accede a un directorio del sitio el recurso que se abrirá automáticamente será "public.html". Si no existe ese recurso, no se mostrará el contenido del directorio.

```
<IfModule mod_ssl.c>
        <VirtualHost www.aragon.local:443>
                ServerAdmin webmaster@localhost

                DocumentRoot /var/web/www

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                SSLEngine on

                SSLCertificateFile      /root/certs/www/www.aragon.local.crt
                SSLCertificateKeyFile /root/certs/www/www.aragon.local.key

                <Directory /var/web/www>
                        Options Indexes FollowSymLinks
                        AllowOverride None
                        Require all granted
                        DirectoryIndex main.html main2.html
                </Directory>

                UserDir /home/ldap/*/www

                <Directory /home/ldap>
                        Options -Indexes +FollowSymLinks
                        AllowOverride None
                        Require all granted
                        Require ip 192.168.30.0/24
                        DirectoryIndex public.html
                </Directory>
        </VirtualHost>

        <VirtualHost www.aragon.local:80>
                ServerName www.aragon.local
                Redirect / https://www.aragon.local/
        </VirtualHost>
</IfModule>
```

intranet.aragon.local

- Sitio web que requiere autenticación contra el servidor LDAP de la máquina ldap. Sólo estará accesible mediante el protocolo HTTPS, por lo que las peticiones HTTP serán redirigidas a HTTPS.
- Utilizará el segundo certificado creado en la tarea anterior (su "Common Name" es intranet.CCAA.com).

- Dentro de la estructura del sitio web, debe haber un directorio llamado "private" al que sólo se podrá acceder desde la máquina ldap y desde la propia máquina server.
- Este sitio estará en la ruta /var/web/intranet. Crea los directorios y archivos necesarios para demostrar que funciona.

```
<IfModule mod_ssl.c>
        <VirtualHost intranet.aragon.local:443>
                ServerAdmin webmaster@localhost

                DocumentRoot /var/web/intranet

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                SSLEngine on

                SSLCertificateFile      /root/certs/intranet/intranet.aragon.local.crt
                SSLCertificateKeyFile /root/certs/intranet/intranet.aragon.local.key

                <Directory /var/web/intranet>
                        Options Indexes FollowSymLinks
                        AllowOverride None

                        AuthName "LDAP Authentication"
                        AuthType Basic
                        AuthBasicProvider ldap
                        AuthLDAPURL ldap://ldap.aragon.local/dc=aragon,dc=local?uid?
sub?(objectClass=*)
                        Require valid-user
                </Directory>
                <Directory /var/web/intranet/private>

                        Require ip 192.168.30.98 192.168.30.99
                </Directory>
        </VirtualHost>

        <VirtualHost intranet.aragon.local:80>
                ServerName intranet.aragon.local
                Redirect / https://intranet.aragon.local:443/
        </VirtualHost>
</IfModule>
```