

CentOS Server - Configs

Additional Configs For Server Applications

- Config List
 - Configs List
- Configs
 - Apache Virtual Hosts
 - Let's Encrypt Certificates

Config List

Configs List

Configs List

	√	To Do
1		Apache Virtual Hosts
2		Let's Encrypt Certificates

Configs

Apache Virtual Hosts

Virtual Hosts Folder Structure

1. Entering on `/etc/httpd/` path and create `sites-available` and `sites-enabled` directories

```
Enter httpd folder
=====

cd /etc/httpd/

Create necessary directories
=====

mkdir sites-available sites-enabled
```

Httpd.conf Changes

1. Edit `http.conf` file and add the below directive line at the bottom of the file, which will make Apache read and parse all files located on `/etc/httpd/sites-enabled/` ended in `.conf` extension

```
Open httpd.conf
=====

vi /etc/conf/httpd.conf

Add the below line
=====

IncludeOptional sites-enabled/*.conf
```

Create Enable/Disable Site Scripts

Enable Site Script

1. Create the following file, a2ensite, in /usr/local/bin and add the script to it

Create file

=====

vi /usr/local/bin/a2ensite

Add the script

=====

```
#!/bin/bash
if test -d /etc/httpd/sites-available && test -d /etc/httpd/sites-enabled ; then
echo "-----"
else
mkdir /etc/httpd/sites-available
mkdir /etc/httpd/sites-enabled
fi

avail=/etc/httpd/sites-available/$1.conf
enabled=/etc/httpd/sites-enabled/
site=`ls /etc/httpd/sites-available/`

if [ "$#" != "1" ]; then
    echo "Use script: a2ensite virtual_site"
    echo -e "\nAvailable virtual hosts: \n$site"
    exit 0
else

if test -e $avail; then
sudo ln -s $avail $enabled
else

echo -e "$avail virtual host does not exist! Please create one! \n$site"
exit 0
fi

if test -e $enabled/$1.conf; then
```

```
echo "Success!! Now restart Apache server: sudo systemctl restart httpd"
else
echo -e "Virtual host $avail does not exist!\nPlease see available virtual hosts:\n$site"
exit 0
fi
fi
```

Disable Site Script

1. Create the following file, a2dissite, in /usr/local/bin and add the script to it

Create file

=====

vi /usr/local/bin/a2dissite

Add the script

=====

```
#!/bin/bash
avail=/etc/httpd/sites-enabled/$1.conf
enabled=/etc/httpd/sites-enabled
site=`ls /etc/httpd/sites-enabled/`

if [ "$#" != "1" ]; then
    echo "Use script: a2dissite virtual_site"
    echo -e "\nAvailable virtual hosts: \n$site"
    exit 0
else

if test -e $avail; then
sudo rm $avail
else
echo -e "$avail virtual host does not exist! Exiting!"
exit 0
fi

if test -e $enabled/$1.conf; then
echo "Error!! Could not remove $avail virtual host!"
else
```

```
echo -e "Success! $avail has been removed!\nPlease restart Apache: sudo systemctl restart\nhttpd"\nexit 0\nfi\nfi
```

Make The Scripts Executable

1. Ensure both scripts are executable

```
chmod +x /usr/local/bin/a2*
```

Create Virtual Host Configs

The Virtual Host Config File

1. Create a new Virtual Host on sites-available location using a descriptive name

```
/etc/httpd/sites-available/test.lan.conf
```

2. Use the below file as a template (replacing `test.lan` with appropriate details

The template enforce the use of SSL (https connections to the site)

```
<VirtualHost test.lan:80>\n    ServerName test.lan\n\n    Redirect permanent / https://test.lan\n\n</VirtualHost>\n\n<VirtualHost test.lan:443>\n    ServerName test.lan\n\n    DocumentRoot "/var/www/test.lan"\n    <Directory "/var/www/test.lan">\n\n        Options Indexes FollowSymLinks MultiViews\n        # AllowOverride controls what directives may be placed in .htaccess files.\n        AllowOverride All
```

```
# Controls who can get stuff from this server file
Order allow,deny
Allow from all
```

```
</Directory>
```

```
<IfModule mpm_peruser_module>
    ServerEnvironment apache apache
</IfModule>
```

```
</VirtualHost>
```

Folder Structure

1. Create the corresponding folder if the default path was changed

```
mkdir -p /var/www/test.lan
```

Remember to create a corresponding DNS record to access the site in the Xneelo config console

Change folder and file ownership | `chown -R apache:apache /folder location/folder` |

Enable a Site

1. Run the below commands one-after-the-other to enable a site

```
a2ensite rheltest.lan
```

```
systemctl restart httpd
```

Remember to issue a certificate for the site or else the site will be inaccessible

Let's Encrypt Certificates

Obtaining a Certificate

1. Run the command below (replacing example.com with the actual domain details)

```
|certbot --apache -d example.com|
```

2. Follow the prompts of the application
3. At the option, "Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access," choose 1. Https redirection is configured in the domains conf file

```
Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
```

```
- - - - -
```

```
1: No redirect - Make no further changes to the webserver configuration.
```

```
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for  
new sites, or if you're confident your site works on HTTPS. You can undo this  
change by editing your web server's configuration.
```

```
- - - - -
```

```
Select the appropriate number [1-2] then [enter] (press 'c' to cancel):1
```

Checking a Certificate Status

1. Open the below link in your preferred web browser, replacing example.com with your base domain

```
|https://www.ssllabs.com/ssltest/analyze.html?d=example.com|
```

2. You will land on a page that immediately begins testing the SSL connection to your server

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) >

SSL Report:

⌘ Please wait...
Testing renegotiation

Certificate #1: RSA 2048 bits (SHA256withRSA)



Server Key and Certificate #1



Subject	Fingerprint SHA256: Pin SHA256:
Common names	
Alternative names	
Serial Number	
Valid from	Tue, 21 May 2019 20:40:45 UTC
Valid until	Mon, 19 Aug 2019 20:40:45 UTC (expires in 2 months and 28 days)
Key	
Weak key (Debian)	No
Issuer	Let's Encrypt Authority X3 AIA: http://cert.int-x3.letsencrypt.org/
Signature algorithm	SHA256withRSA
Extended Validation	No
Revocation information	OCSP OCSP: http://ocsp.int-x3.letsencrypt.org
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows



Additional Certificates (if supplied)

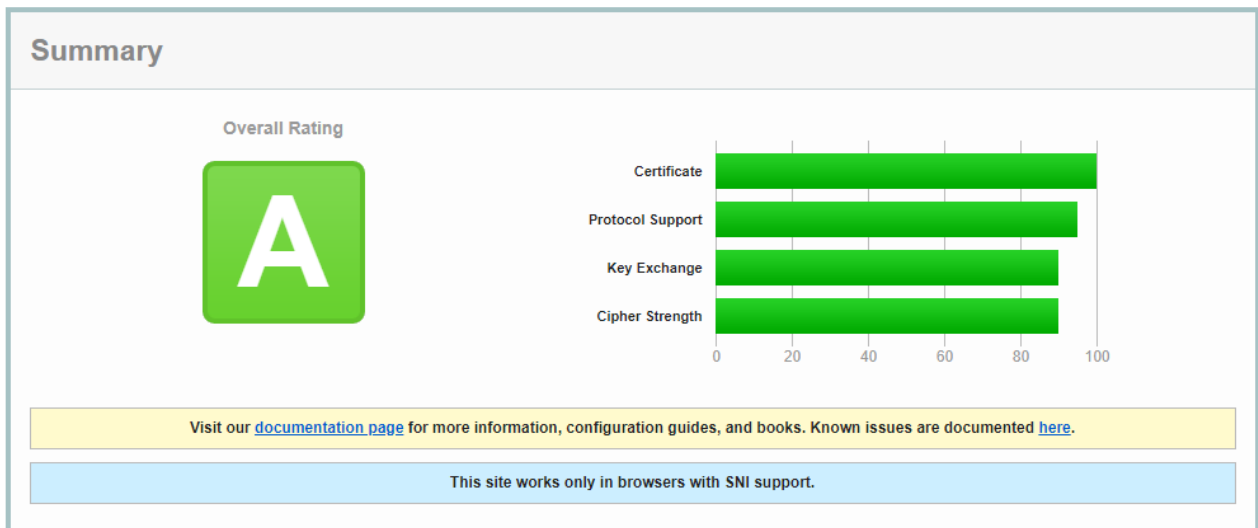


Certificates provided	2 (2555 bytes)
Chain issues	None

#2

Subject	Let's Encrypt Authority X3 Fingerprint SHA256: Pin SHA256:
Valid until	Wed, 17 Mar 2021 16:40:46 UTC (expires in 1 year and 9 months)
Key	RSA 2048 bits (e 65537)
Issuer	DST Root CA X3
Signature algorithm	SHA256withRSA

3. A result page, similar to the one below, will be displayed once the test has completed



Setting Up Auto Renewal

Dry Run Test

1. Do a dry run to test renewal with the command below

```
certbot renew --dry-run
```

2. The result should be similar to

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log

-----
Processing /etc/letsencrypt/renewal/example.com.conf
-----

Cert not due for renewal, but simulating renewal for dry run
Plugins selected: Authenticator apache, Installer apache
Starting new HTTPS connection (1): acme-staging-v02.api.letsencrypt.org
Renewing an existing certificate
Performing the following challenges:
http-01 challenge for example.com
http-01 challenge for www.example.com
Waiting for verification...
Cleaning up challenges
Resetting dropped connection: acme-staging-v02.api.letsencrypt.org

-----
```

```
new certificate deployed with reload of apache server; fullchain is
/etc/letsencrypt/live/example.com/fullchain.pem

- - - - -

- - - - -

** DRY RUN: simulating 'certbot renew' close to cert expiry
**          (The test certificates below have not been saved.)

Congratulations, all renewals succeeded. The following certs have been renewed:
/etc/letsencrypt/live/example.com/fullchain.pem (success)
```

Auto Renew

1. Open the crontab file

```
| crontab -e |
```

2. Add the command below to auto renew certificates

```
| 0 0,12 * * * python -c 'import random; import time; time.sleep(random.random() * 3600)'
&& certbot renew |
```

3. Save and exit