Linux Servers

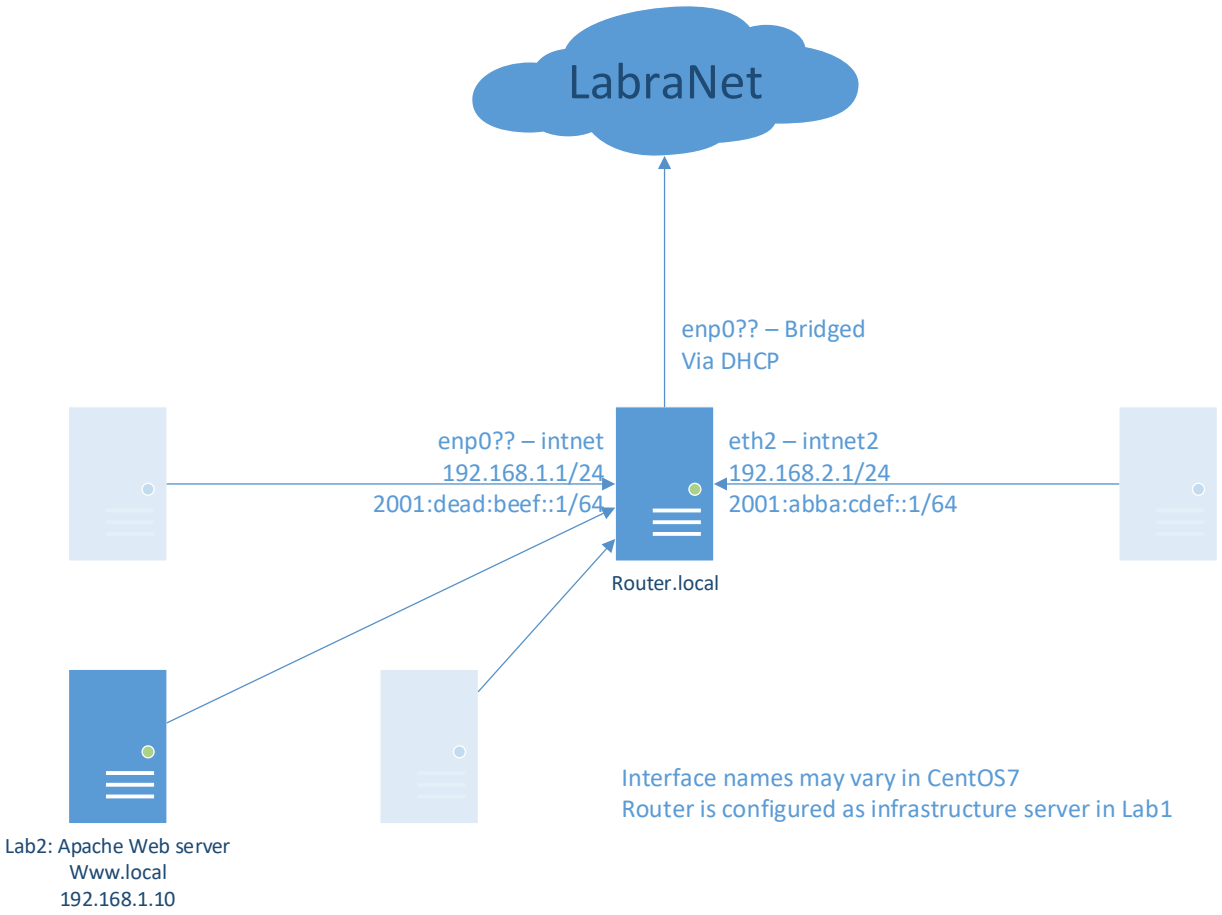
Lab2 – Apache HTTP Server

Mikael Romanov

Document your commands or take screenshots. Answer questions in english or finnish.

All configuration must be persistent (survive a reboot). You may use your own notes and Internet resources for your aid. You may consult the teacher or other students.

This lab will use the following topology:



- **Apache installation (1p)**

Install Apache (package and service name httpd), mod_ssl, Mariadb/MySQL server (mariadb-server, mariadb) and PHP (php, php-mysql). Turn on both httpd and mariadb-server -services. Add firewall rules for http and https traffic.

I first installed all the services, could've done with yum install -y httpd mariadb mariadb-server php php-mysql ,

```
[root@localhost yum.repos.d]# yum install -y httpd
```

```
[root@localhost yum.repos.d]# yum install -y mariadb_
```

```
[root@localhost yum.repos.d]# yum install -y mariadb-server_
```

```
[root@localhost yum.repos.d]# yum install php
```

```
[root@localhost yum.repos.d]# yum install php-mysql_
```

After the installs I started the mysql

```
[root@localhost yum.repos.d]# systemctl start mariadb.service
```

And also the apache

```
[root@localhost yum.repos.d]# systemctl start httpd.service
```

Then I changed the hostname of the webserver and made firewall rules for enabling http and httpd for the internal zone. And reloaded the firewall for the rules to take affect.

```
[root@localhost yum.repos.d]# hostname
www
[root@localhost yum.repos.d]# firewall-cmd --permanent --zone=internal --add-ser
vice=http
success
[root@localhost yum.repos.d]# firewall-cmd --permanent --zone=internal --add-ser
vice=https
success
[root@localhost yum.repos.d]# firewall-cmd --reload
success
```

- **Port forwarding rules (1p)**

Create a port forwarding rule for the infrastructure server you made in Lab 1. Forward http and https ports to www.local. Use either firewallD or iptables depending on what you used in Lab1.

To www computer:

```
[root@localhost html]# firewall-cmd --add-service=http --permanent
success
[root@localhost html]# firewall-cmd --add-service=https --permanent
success
[root@localhost html]# firewall-cmd --reload
success
```

To router:

I made a port forwarding rule, so when someone connects to the routers ip:80 it will be forwarded to the apache web server.

```
[root@router named]# firewall-cmd --zone=external --add-forward-port=port=80:pro
to=tcp:toport=80:toaddr=192.168.1.10 --permanent_
```

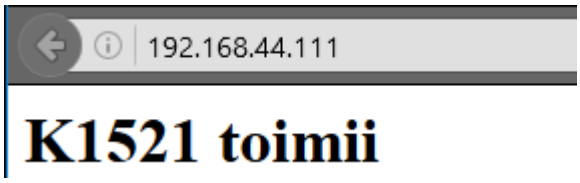
The same also for the https (port 443)

```
[root@router named]# firewall-cmd --zone=external --add-forward-port=port=443:pr
oto=tcp:toport=443:toaddr=192.168.1.10 --permanent
success
```

Added the firewall rules to have http and https

```
[root@router named]# firewall-cmd --zone=trusted --add-service=http --permanent
success
[root@router named]# firewall-cmd --zone=trusted --add-service=https --permanent

success
```

After the firewall rules I tested that the forwarding works. Ofcourse I modified prematurely the insides of /var/www/html/ and added ther and index.html file and wrote inside <h1>K1521 toimii</h1>



- **Basic hardening (1p)**

Run mysql_secure_installation and define a root password for the database. Another options may be left for defaults.

So I runned the mysql installation and changed the roots password. Remainder to self password is the same as root.

```
Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!
```

Turn off Apache default page (welcome page). Turn off sending of ServerTokens and ServerSignature. Disable indexing if it is on.

I commented everything in the welcome conf, allthought this is not necessary since I made the index.html file which disables this conf file:

```
  GNU nano 2.3.1       File: /etc/httpd/conf.d/welcome.conf

#
# This configuration file enables the default "Welcome" page
# is no default index page present for the root URL.  To dis
# Welcome page, comment out all the lines below.
#
# NOTE: if this file is removed, it will be restored on upgr
#
#<LocationMatch "^/+$">
#    Options -Indexes
#    ErrorDocument 403 /.noindex.html
#</LocationMatch>
#
#<Directory /usr/share/httpd/noindex>
#    AllowOverride None
#    Require all granted
#</Directory>
#_
#Alias /.noindex.html /usr/share/httpd/noindex/index.html
#Alias /noindex/css/bootstrap.min.css /usr/share/httpd/noind
#Alias /noindex/css/open-sans.css /usr/share/httpd/noindex/c
```

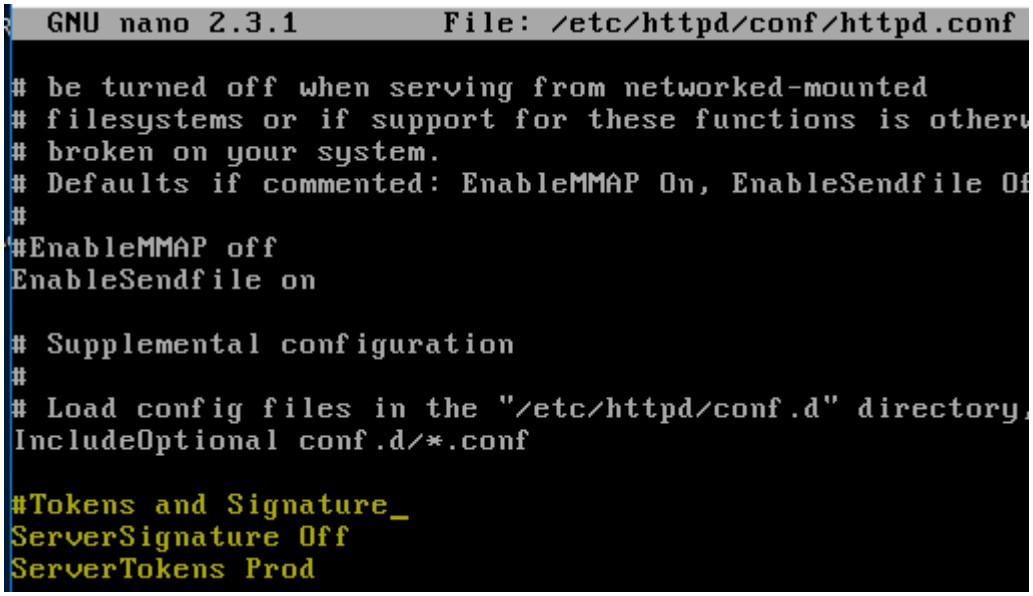
I made the index html file earlier in the start of this lab:

```
[root@localhost html]# pwd
/var/www/html
[root@localhost html]# ls -lah
total 4.0K
drwxr-xr-x. 2 root root 23 Nov 18 10:21 .
drwxr-xr-x. 4 root root 31 Nov 18 09:26 ..
-rw-r--r--. 1 root root 22 Nov 18 09:58 index.html
```

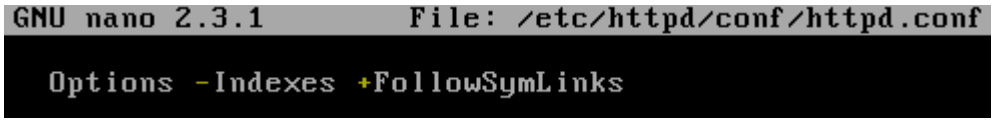
Inside the index.html file

```
  GNU nano 2.3.1              File: index.html

<h1>K1521 toimii</h1>
```

Inside the httpd.conf I commented the Tokens and Signature and added two lines ServersSignature Off and ServerTokens Prod

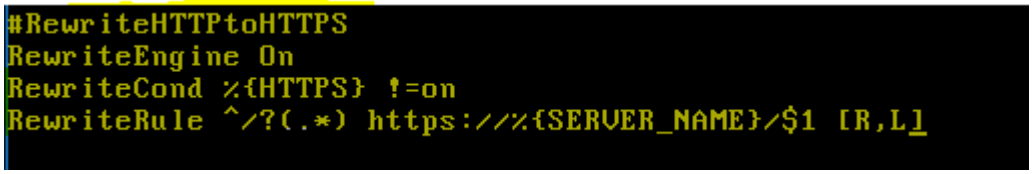


Then modified the indexes



Create a redirect/rewrite rule that forces SSL to be used if the server is accessed via HTTP. (See Apache RedirectSSL or RewriteHTTPToHTTPS)

I modified ??? and Added few lines to achieve the server to redirect to https when using port 80.



As seen below the server redirected the traffic to https.

- **phpinfo and basic authentication (1p)**

Create a directory /var/www/info and create a phpinfo page in there (examples in internet). Protect the directory with basic authentication (with login: test/test). Use .htaccess file (do not configure

authentication directly in /etc/httpd/coclenf/httpd.conf, although you might have to modify this file also for AllowOverride).

I made a folder named it info and moved it inside of the html folder.

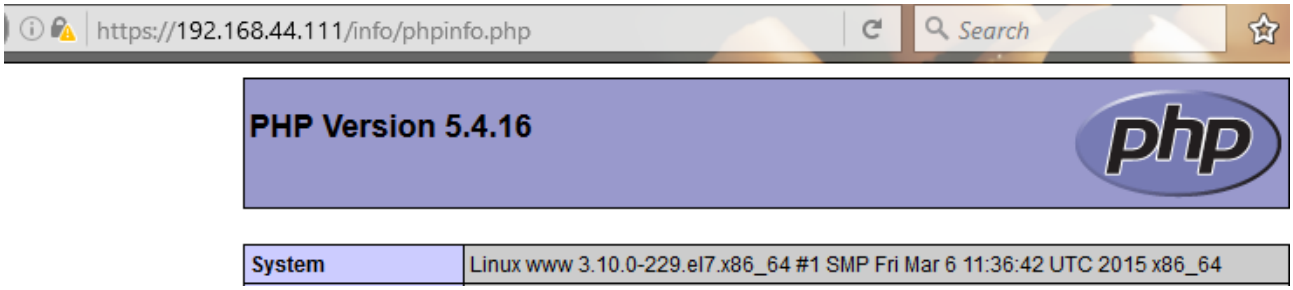
```
drwxr-xr-x. 2 root root   6 Jul 18 18:30 cgi-bin
drwxr-xr-x. 3 root root  75 Nov 25 09:02 html
drwxr-xr-x. 2 root root  24 Nov 25 08:18 info
-rw-r--r--. 1 root root 127 Nov 25 08:35 php-fastcgi
[root@www www]# pwd
/var/www
[root@www www]# mv info html/_
```

```
[root@www info]# ls -l
total 4
-rw-r--r--. 1 root root 66 Nov 18 11:07 phpinfo.php
[root@www info]# pwd
/var/www/html/info
```

Inside info folder I made a phpinfo.php file and added there few lines. The default indicators off php (<?php ?> and inside of the indicator I wro phpinfo();

```
  GNU nano 2.3.1                File: phpinfo.php

<?php

//Show all information, defaults to INFO_ALL
phpinfo();
?>
```

I then tested with the desktop browser that I can reach the phpinfo.



After I tested the php worked, I then made 2 files. The file were .htaccess and .htpasswd for apache login.

```
[root@localhost www]# touch .htaccess
[root@localhost www]# touch .htpasswd
```

```
[root@www www]# pwd
/var/www
[root@www www]# ls -lah
total 12K
drwxr-xr-x.  5 root root   74 Nov 25 08:18 .
drwxr-xr-x. 22 root root 4.0K Nov 25 08:03 ..
drwxr-xr-x.  2 root root    6 Jul 18 18:30 cgi-bin
-rw-r--r--.  1 root root   79 Nov 18 11:22 .htaccess
drwxr-xr-x.  3 root root   59 Nov 18 11:51 html
-rw-r--r--.  1 root root   16 Nov 18 11:23 .htpasswd
drwxr-xr-x.  2 root root   24 Nov 25 08:18 info
```

I moved the .htaccess file into the info folder, because that was the page that needed to be password protected. The password file remainded in the /var/www/ folder

```
[root@www html]# mv .htaccess /var/www/html/info
```

I then configured the .htaccess file and wrote inside the path to the password file, AuthName “Something that pops up” Authorisation type Basic and that the login requires a valid person.

```
  GNU nano 2.3.1        File: /var/www/html/info/.htaccess

AuthUserFile /var/www/.htpasswd
AuthName "Restricted area"
AuthType Basic
require valid-user
```

Then I configured the .htpasswd file where the user is defined on the left and separated with a colon and on the right is salted md5 hash. The password is test and the hashed password can be made with the unix tool htpasswd. The command is htpasswd -c /var/.htpasswd k1521. The password is prompted

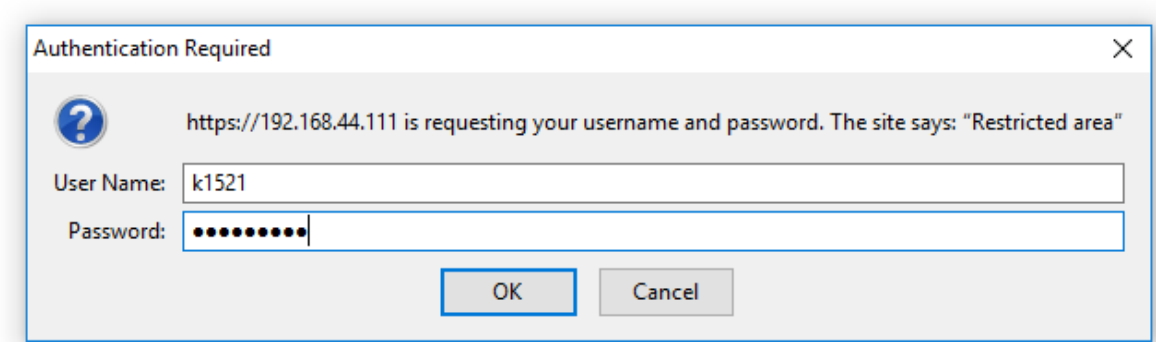
```
  GNU nano 2.3.1            File: /var/www/.htpasswd

k1521:dGRkPurkuWmW2
```

After the .htaccess and .htpasswd were done, I allowed override from the httpd.conf for the .htaccess to take affect.

```
  GNU nano 2.3.1        File: /etc/httpd/conf/httpd.conf

# Relax access to content within /var/www.
#
<Directory "/var/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>

<Directory "/var/www/html/info">
        AllowOverride All
</Directory>
```

Of course I had to reload apache before the it worked. But as seen below the htaccess works.

l toimii



NOTE: Do NOT put htpasswd -file anywhere in /var/www/html as this will make the file world-readable.

- **Wordpress installation (1p)**

Install Wordpress from sources (wordpress.org link for tar.gz) to directory /var/www/html/wordpress. Use any installation guide you want.

I downloaded wordpress from wordpress site with wget.

```
[root@localhost www]# wget http://www.wordpress.org/latest.tar.gz
```

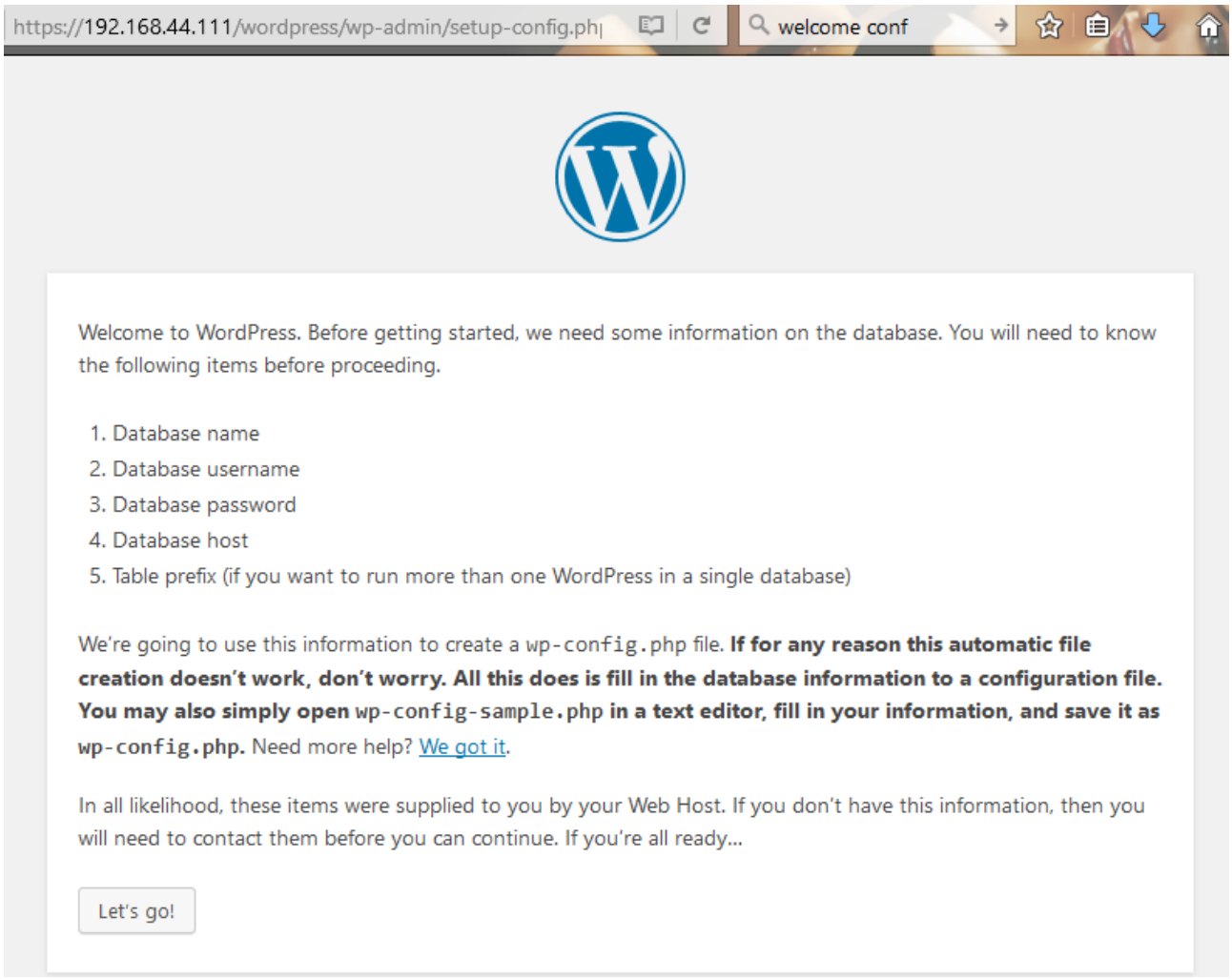
Then extracted the tarball into the html folder

```
[root@localhost wordpress]# tar fzxv latest.tar.gz
```

As seen below the wordpress folder

```
[root@localhost html]# ls -l
total 7784
-rw-r--r--. 1 root   root       22 Nov 18 09:58 index.html
-rw-r--r--. 1 root   root  7961036 Sep  7 17:59 latest.tar.gz
drwxr-xr-x. 5 nobody 65534    4096 Sep  7 17:59 wordpress
```

I then tested if the wordpress worked



I started mysql

```
[root@www info]# systemctl start mariadb
```

Enabled it so it starts on boot and logged in

```
[root@www info]# systemctl enable mariadb
```

I then created a new database

```
MariaDB [(none)]> CREATE DATABASE wordpress;
```

Created a new user and a password which is kumiankka

```
MariaDB [(none)]> CREATE USER rommi@localhost IDENTIFIED BY 'kumiankka';
```

Then I granted all privileges to the user I made

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO rommi@localhost IDENTIF
IED BY 'kumiankka';
```

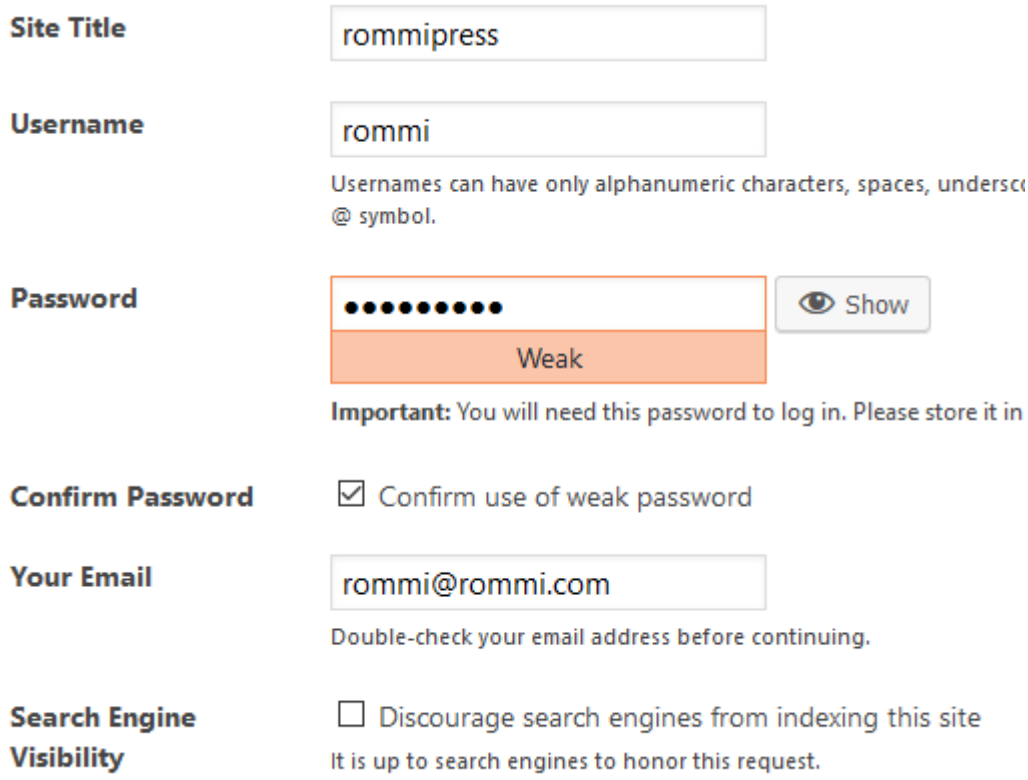
And flushed the privileges

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

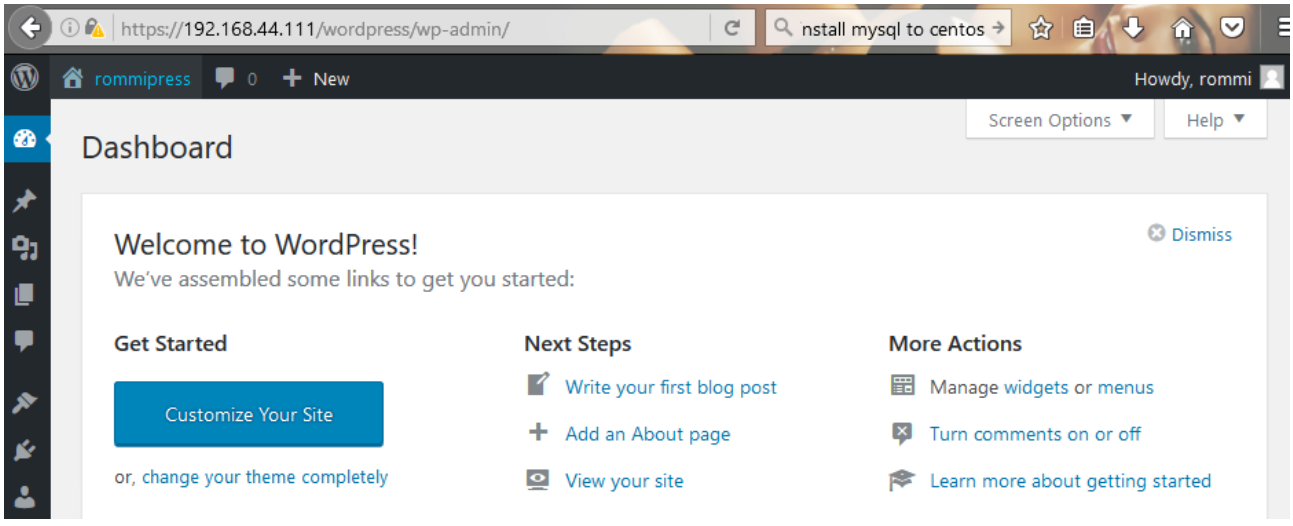
After the database config

```
  GNU nano 2.3.1                File: wp-config.php                          M

 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'rommi');

/** MySQL database password */
define('DB_PASSWORD', 'kumiankka');
```

After the configuration I went to my site/wordpress and I was prompted with a new screen. I then configured all the necessary parameters.



As seen I was able to configure wordpress and I'm inside to start making a site



Take care of SELinux contexts (search wordpress SELinux). You need at least the following SELinux boolean:

httpd_can_network_connect_db

Then I configured the SELinux for wordpress.

```
[root@www ~]# setsebool -P httpd_can_network_connect_db true
```

```
[root@www ~]# getsebool httpd_can_network_connect_db
httpd_can_network_connect_db --> on
```

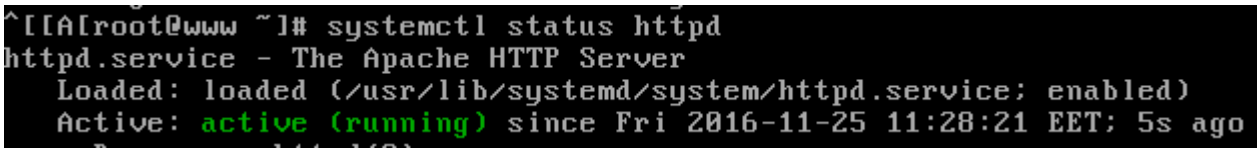
Testing

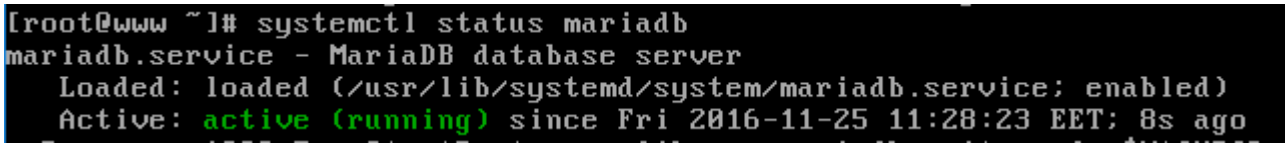
Reboot www.local verify that your configuration persists. Test that you can access the server via the router LabraNet IP address (port forwards http and https to www.local).

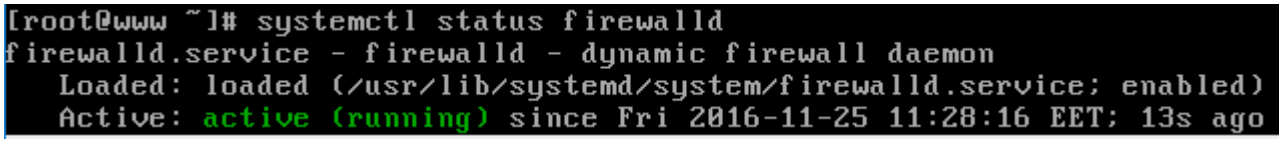
Checklist for screenshots/documentation:

- Services are running after a reboot (systemctl status service)

Everything works after boot





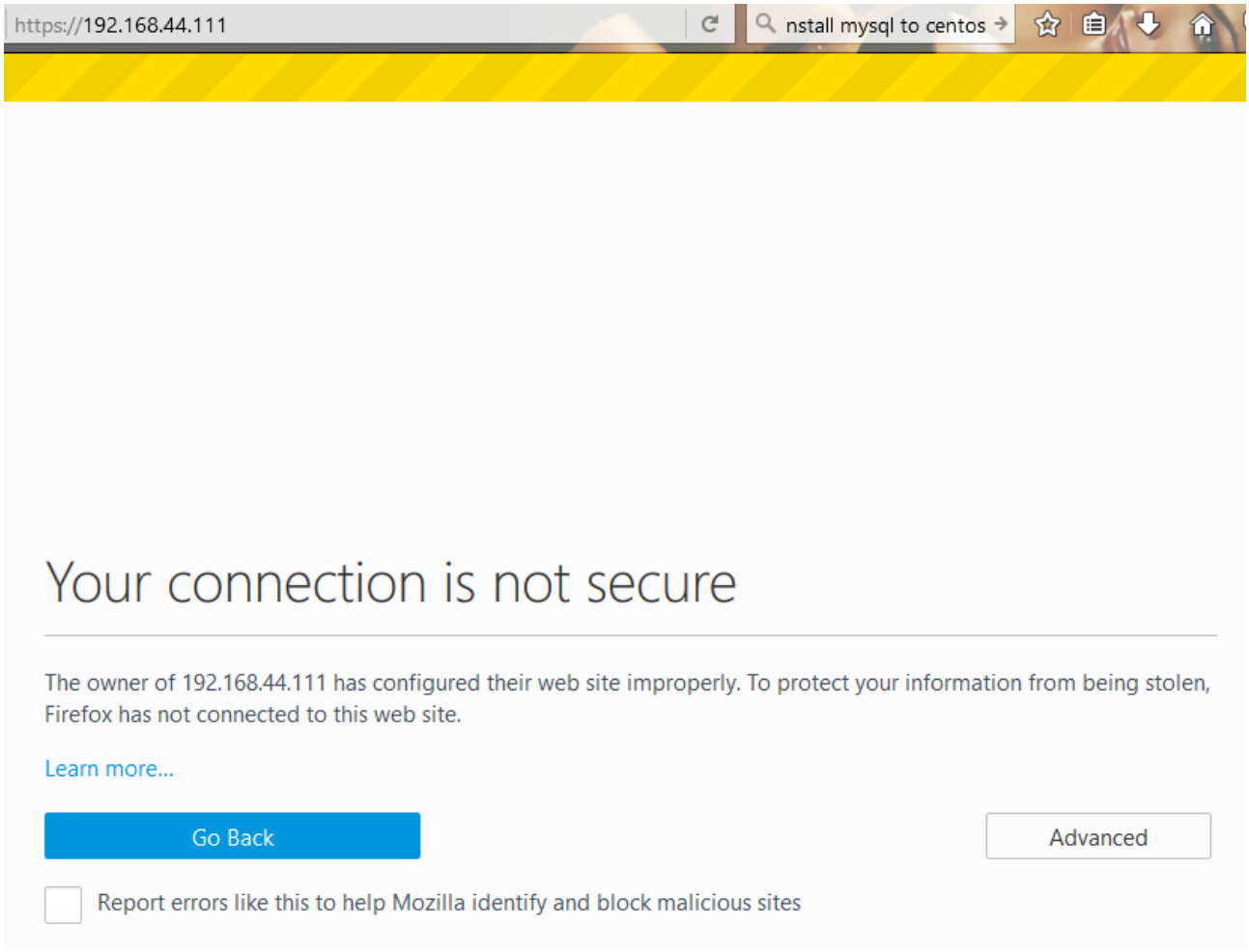


- Error pages show minimal information about the server

As seen below the server shows only the minimal info

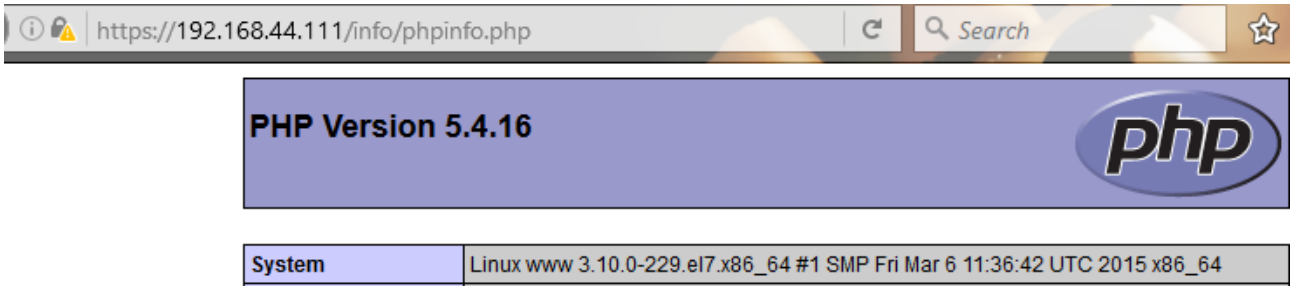
- HTTPS is forced even if you access with HTTP

As seen below the https is forced on the server.



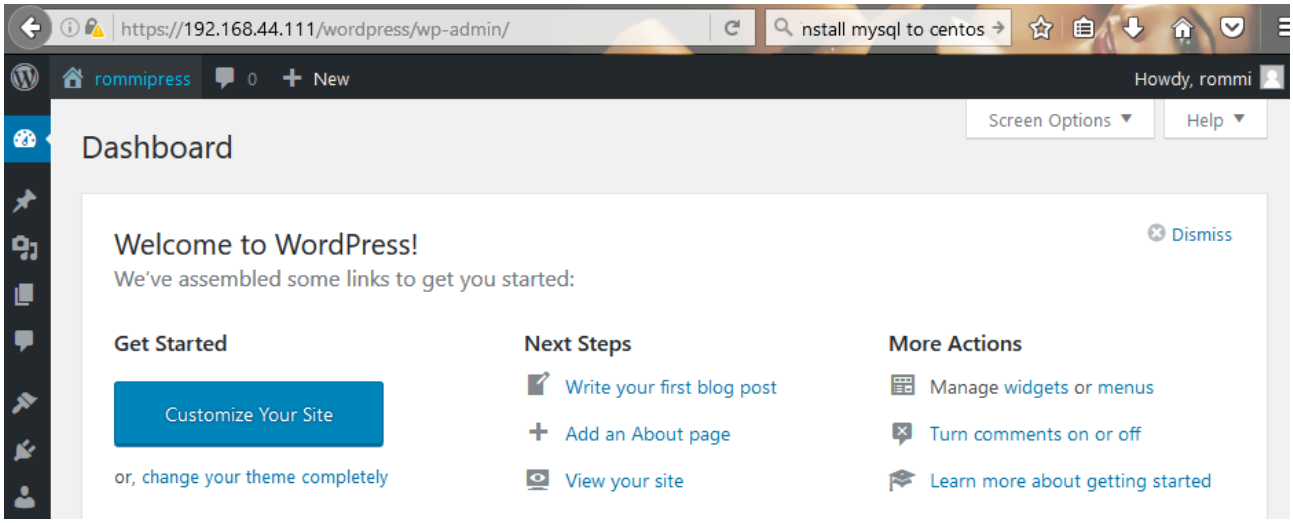
- /phpinfo shows PHP info page but after Basic Authentication login

The php info page prompts a login window and then shows the phpinfo.



- /wordpress has a working instance of Wordpress installed

Wordpress working as it should


https://192.168.44.111/wordpress/wp-admin/
nstall mysql to centos
rommipress
0
New
Howdy, rommi
Screen Options
Help
Dashboard
Welcome to WordPress!
We've assembled some links to get you started:
Dismiss
Get Started
Customize Your Site
or, change your theme completely
Next Steps
Write your first blog post
Add an About page
View your site
More Actions
Manage widgets or menus
Turn comments on or off
Learn more about getting started