This document provides comprehensive instructions for Part III of the Final Project I for the course 420-635-AB Network Installation and Administration I. The project involves configuring the Apache web server with specific parameters, enabling server monitoring, using CGI and PHP scripts, integrating MySQL with PHP, and implementing SSL for secure access. Each task is designed to enhance understanding and skills in network installation and administration

Project Part III

420-635-AB-Network Installation and Administration I

Teacher: Antoine Tohme
Student: Monica Perez Mata
Student id : 2498056

1 Introduction

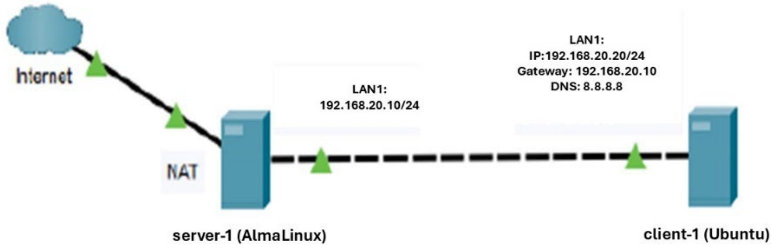
This document provides detailed instructions for Part III of the Final Project I for the course 420-635-AB Network Installation and Administration I.
The requirements for this project include configuring the Apache web server with specific parameters, enabling server monitoring, using Common Gateway Interface (CGI) and Hypertext Preprocessor (PHP) scripts, integrating My Structured Query Language (MySQL) with PHP, and implementing Secure Sockets Layer (SSL) for secure access. Each task is designed to enhance your understanding and skills in network installation and administration.

The project is divided in 6 different tasks.

- TASK 1 – Multi-Process Module (MPM)
- TASK 2 – Apache Server Monitoring
- TASK 3- CGI
- TASK 4 – PHP
- TASK 5 – MYSQL/PHP
- TASK 6 - SSL

2 Topology

The network topology to be used is the same as used in previous assignments



The base of this project is Part II
The node under test is the AlmaLinux server1.
No configuration nor test is to be done in client side Ubuntu.

3 TASK 1 – Multi-Process Module (MPM)

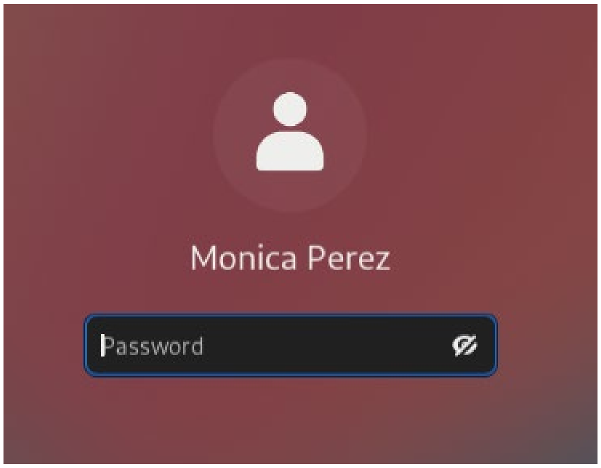
3.1 Requirements

The task includes the following requirements:

- The web pages for this project should be in the directory: /var/www/html_ project3
- Add links to all the web pages of this project in the following file: **var/www/html_project3/master_project3.html.**
- Configure your Apache web server with the following parameters:
  - ➢ Start with 12 httpd server processes when the service starts.
  - ➢ Maintain a minimum of 6 idle server processes at all times.
  - ➢ Allow a maximum of 12 idle server processes.
  - ➢ Handle up to 180 simultaneous client requests.
  - ➢ Queue a maximum of 100 pending requests when the maximum of 180 simultaneous requests is reached.
  - ➢ Enable persistent (KeepAlive) connections, with:
    - ○ A maximum of 50 consecutive requests per connection.
    - ○ A 20-second timeout between two consecutive requests from the same client.
  - ➢ Limit the maximum request duration to 55 seconds per client request.

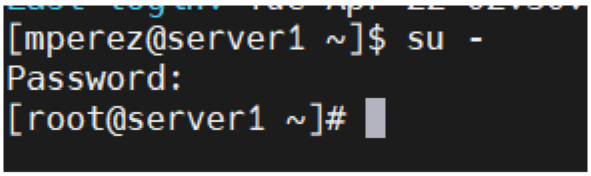
3.2 Pre-activity

1. Connect to AlmaLinux



2. Connect as root

su –



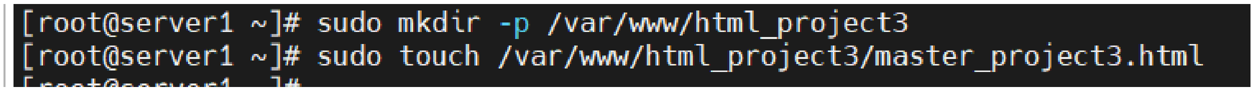
```
[mperez@server1 ~]$ su -
Password:
[root@server1 ~]#
```

3.3 Create the project directory and master page

1. The web pages for this project should be in the directory /var/www/html_ project3

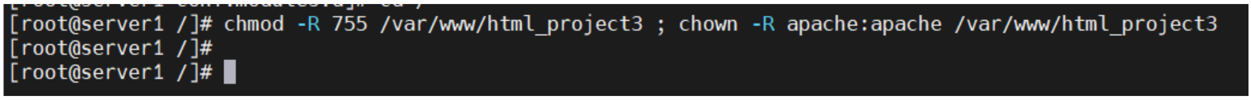
   sudo mkdir -p /var/www/html_project3

   sudo touch /var/www/html_project3/master_project3.html



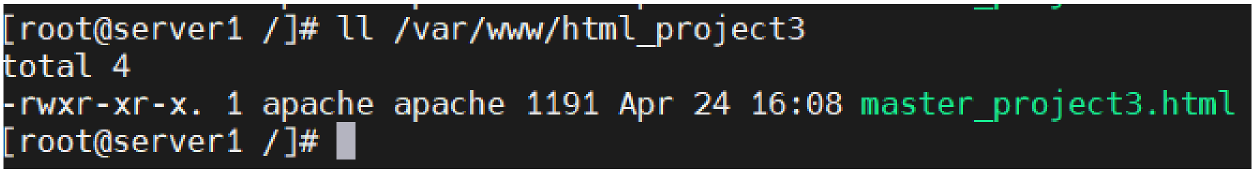
```
[root@server1 ~]# sudo mkdir -p /var/www/html_project3
[root@server1 ~]# sudo touch /var/www/html_project3/master_project3.html
[root@server1 ~]#
```

2. Set Permissions: Ensure Apache can access the directory and file
   chmod -R 755 /var/www/html_project3 ; chown -R apache:apache /var/www/html_project3



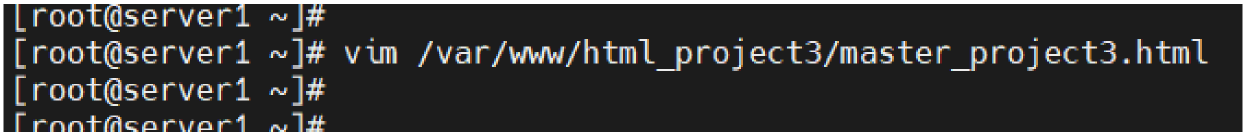
```
[root@server1 /]# chmod -R 755 /var/www/html_project3 ; chown -R apache:apache /var/www/html_project3
[root@server1 /]#
[root@server1 /]#
```

   ll /var/www/html_project3



```
[root@server1 /]# ll /var/www/html_project3
total 4
-rwxr-xr-x. 1 apache apache 1191 Apr 24 16:08 master_project3.html
[root@server1 /]#
```

3. Edit the master page with your content and links to other project pages.

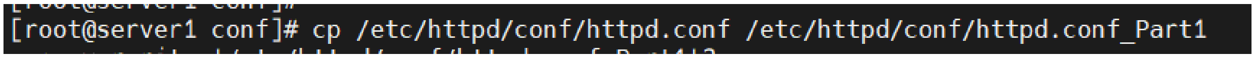


```
[root@server1 ~]#
[root@server1 ~]# vim /var/www/html_project3/master_project3.html
[root@server1 ~]#
[root@server1 ~]#
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Project Part III - Homepage</title>
</head>
<body>
    <h1>Welcome to Project Part 3</h1>
    <hr>

    <!-- Task 2 -->
    <h2>Task 2</h2>
    <ul>
        <li><a href="http://192.168.50.10/server-status"
target="_blank">To view server status</a></li>
```

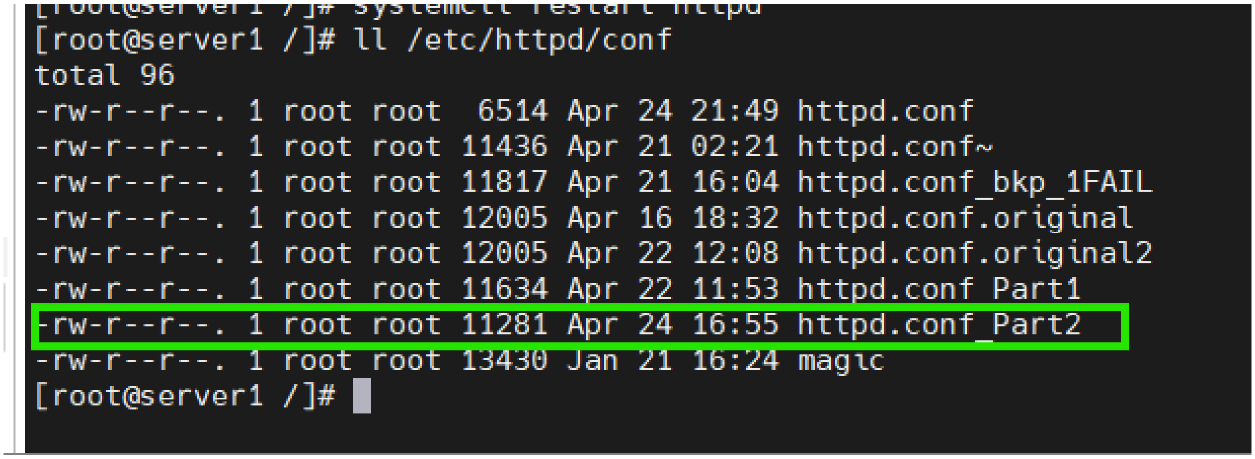
```
        <li><a href="http://192.168.50.10/server-info"
target="_blank">To view server information</a></li>
    </ul>

    <!-- Task 3 -->
    <h2>Task 3</h2>
    <ul>
        <li><a href="http://192.168.50.10/cgi-bin/task3.pl"
target="_blank">To execute CGI Script</a></li>
    </ul>

    <!-- Task 4 -->
    <h2>Task 4</h2>
    <ul>
        <li><a href="http://192.168.100.1/q4/task4.php"
target="_blank">To execute the PHP script using
192.168.100.1</a></li>
        <li><a href="http://192.168.50.10/q4/task4.php"
target="_blank">to execute the PHP scritp using another
subnet</a></li>
    </ul>

    <!-- Task 5 -->
    <h2>Task 5</h2>
    <ul>
        <li><a href="http://192.168.50.10/q5/task5.php"
target="_blank">To list employees mysql table</a></li>
    </ul>

    <!-- Task 6 -->
    <h2>Task 6</h2>
    <ul>
        <li><a href="https:192.168.50.10//" target="_blank">To
view the website on SSL</a></li>
    </ul>

</body>
</html>
```

1. Make a copy of Project1 httpd



```
[root@server1 conf]# cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf_Part1
```

ll /etc/httpd/conf



```
[root@server1 /]# systemctl restart httpd
[root@server1 /]# ll /etc/httpd/conf
total 96
-rw-r--r--. 1 root root  6514 Apr 24 21:49 httpd.conf
-rw-r--r--. 1 root root 11436 Apr 21 02:21 httpd.conf~
-rw-r--r--. 1 root root 11817 Apr 21 16:04 httpd.conf_bkp_1FAIL
-rw-r--r--. 1 root root 12005 Apr 16 18:32 httpd.conf.original
-rw-r--r--. 1 root root 12005 Apr 22 12:08 httpd.conf.original2
-rw-r--r--. 1 root root 11634 Apr 22 11:53 httpd.conf_Part1
-rw-r--r--. 1 root root 11281 Apr 24 16:55 httpd.conf_Part2
-rw-r--r--. 1 root root 13430 Jan 21 16:24 magic
[root@server1 /]#
```

2. Open httpd config to edit

   vim /etc/httpd/conf/httpd.conf

3. Modify the DocumentRoot

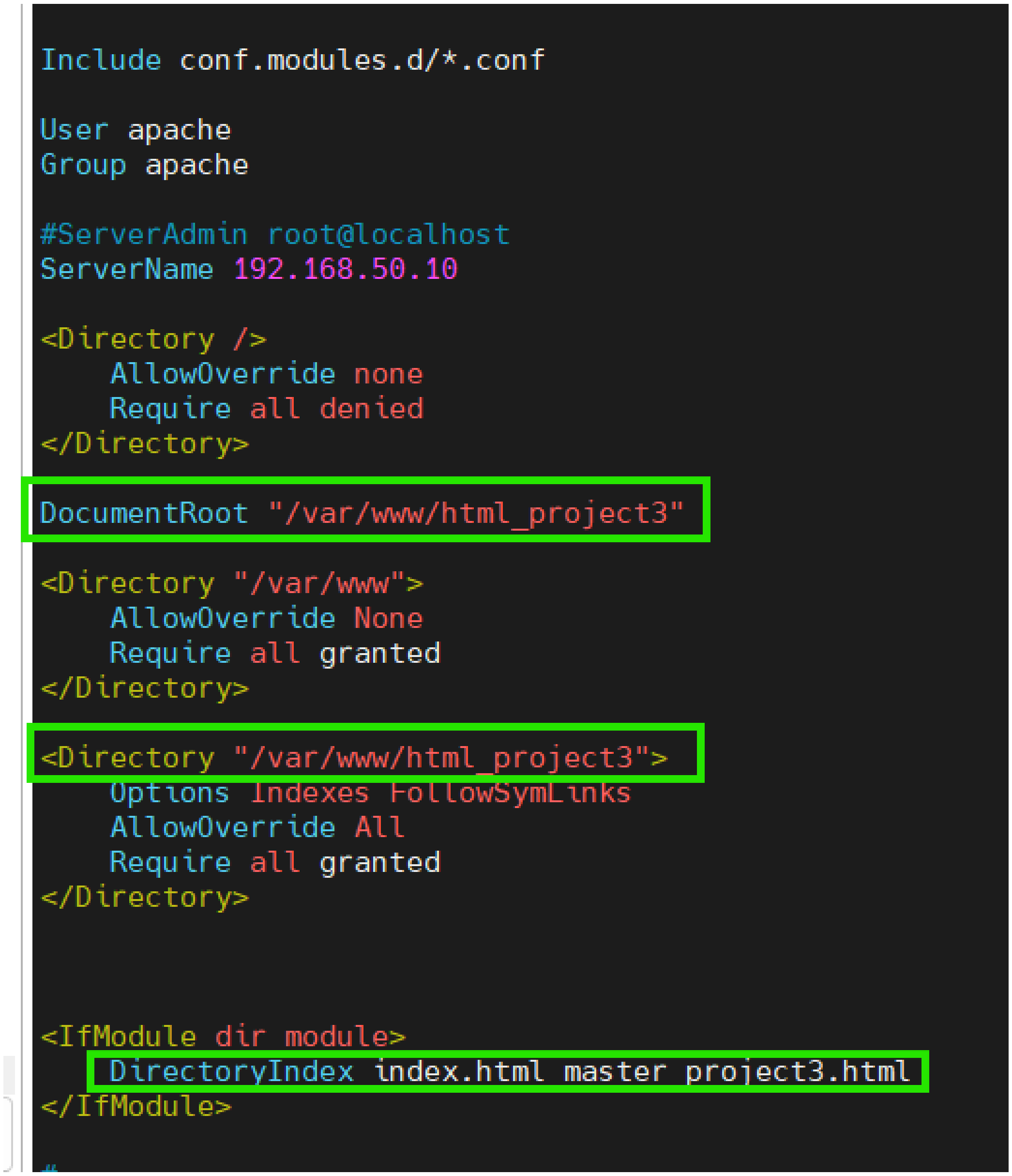
   a) Change DocumentRoot to point to /var/www/html_project2

   b) Set <Directory> permissions

   ```
   <Directory "/var/www/html_project3">
     AllowOverride None
     Require all granted
   </Directory>
   ```

   c) Delete all directory block related to Part1

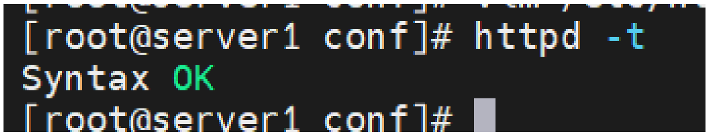
   d) Update the DirectoryIndex directive in the Apache configuration to explicitly set master_project3.html as the default



```
Include conf.modules.d/*.conf

User apache
Group apache

#ServerAdmin root@localhost
ServerName 192.168.50.10

<Directory />
    AllowOverride none
    Require all denied
</Directory>

DocumentRoot "/var/www/html_project3"

<Directory "/var/www">
    AllowOverride None
    Require all granted
</Directory>

<Directory "/var/www/html_project3">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>



<IfModule dir module>
    DirectoryIndex index.html master project3.html
</IfModule>
```

4. Verify the syntax of configuration file after changes

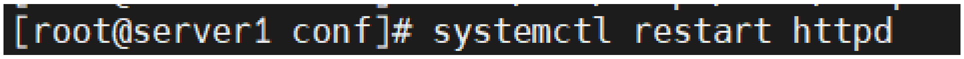
   httpd -t



```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

5. Restart Apache: Apply the configuration changes

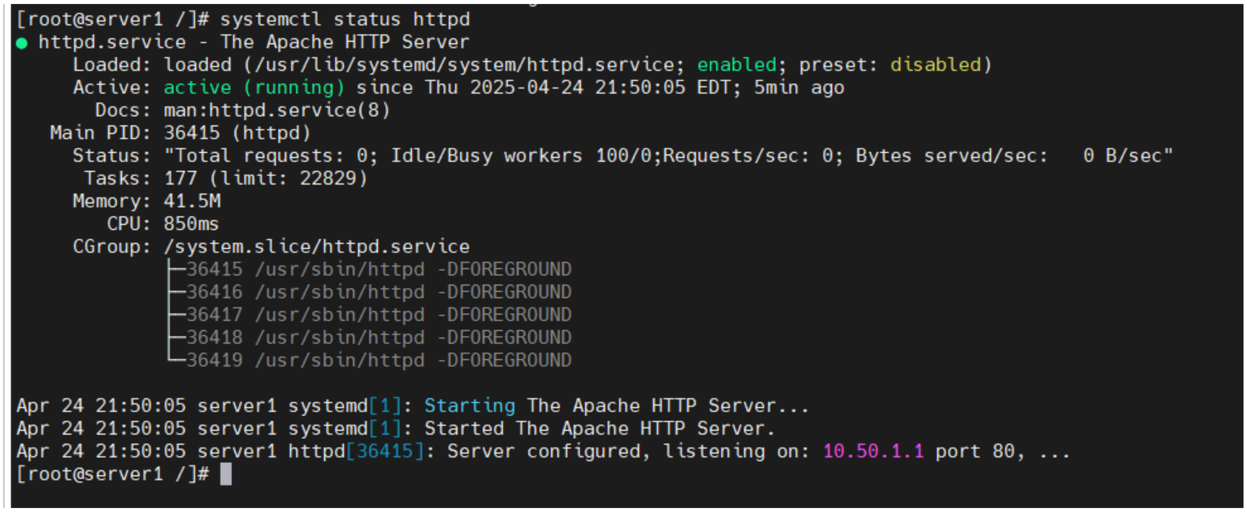
   systemctl restart httpd



```
[root@server1 conf]# systemctl restart httpd
```

6. Verify httpd status

systemctl status httpd

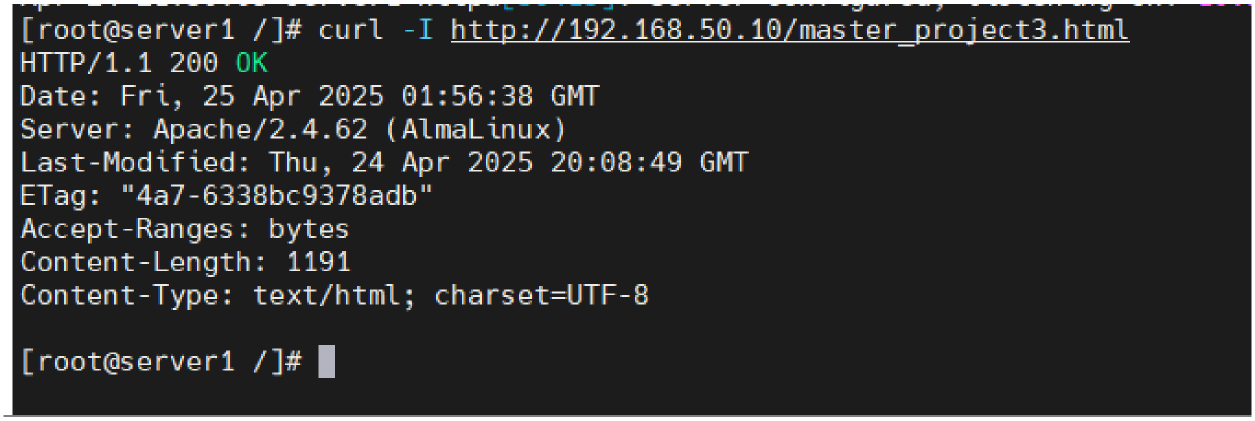


```
[root@server1 /]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-04-24 21:50:05 EDT; 5min ago
       Docs: man:httpd.service(8)
   Main PID: 36415 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 22829)
     Memory: 41.5M
        CPU: 850ms
     CGroup: /system.slice/httpd.service
             ├─36415 /usr/sbin/httpd -DFOREGROUND
             ├─36416 /usr/sbin/httpd -DFOREGROUND
             ├─36417 /usr/sbin/httpd -DFOREGROUND
             ├─36418 /usr/sbin/httpd -DFOREGROUND
             └─36419 /usr/sbin/httpd -DFOREGROUND

Apr 24 21:50:05 server1 systemd[1]: Starting The Apache HTTP Server...
Apr 24 21:50:05 server1 systemd[1]: Started The Apache HTTP Server.
Apr 24 21:50:05 server1 httpd[36415]: Server configured, listening on: 10.50.1.1 port 80, ...
[root@server1 /]#
```

3.4 Test index page

1. Test with curl

   curl -I http://192.168.50.10/master_project3.html



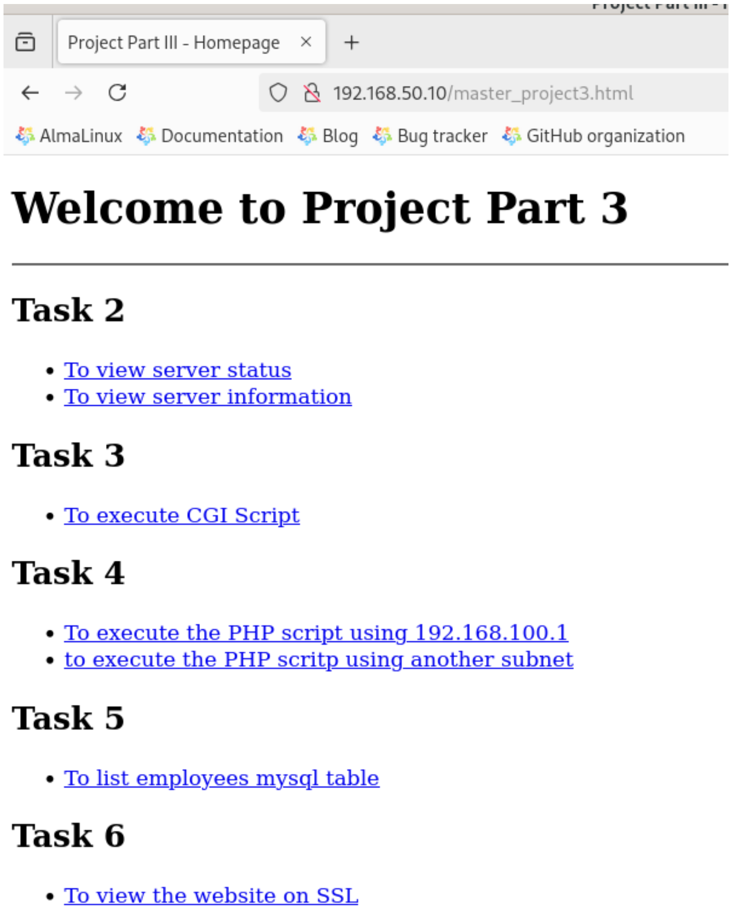
```
[root@server1 /]# curl -I http://192.168.50.10/master_project3.html
HTTP/1.1 200 OK
Date: Fri, 25 Apr 2025 01:56:38 GMT
Server: Apache/2.4.62 (AlmaLinux)
Last-Modified: Thu, 24 Apr 2025 20:08:49 GMT
ETag: "4a7-6338bc9378adb"
Accept-Ranges: bytes
Content-Length: 1191
Content-Type: text/html; charset=UTF-8

[root@server1 /]#
```

   See result indicate 200 **OK**

2. Test in browser

   http://192.168.50.10/master_project3.html

   Note – Links do not work



Project Part III - Homepage
192.168.50.10/master_project3.html
AlmaLinux
Documentation
Blog
Bug tracker
GitHub organization

Welcome to Project Part 3

Task 2

- To view server status
- To view server information

Task 3

- To execute CGI Script

Task 4

- To execute the PHP script using 192.168.100.1
- to execute the PHP scritp using another subnet

Task 5

- To list employees mysql table

Task 6

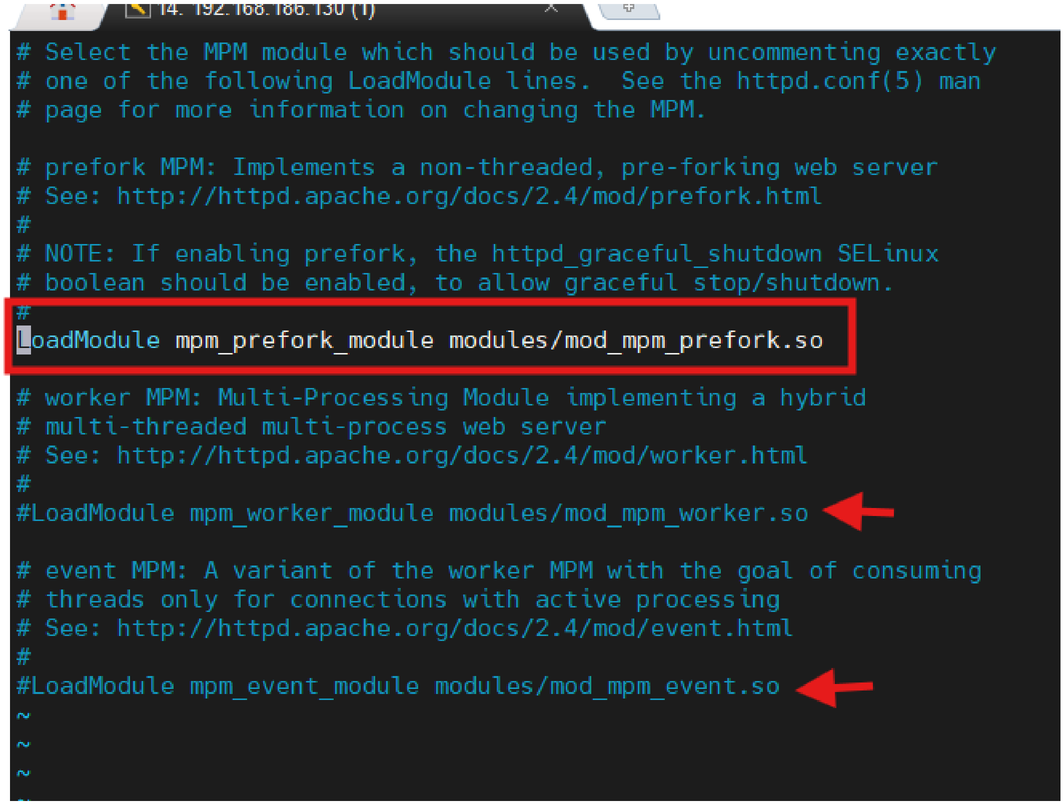
- To view the website on SSL

3.5 Apache configuration (httpd.conf) to configure Apache MPM (Prefork)

1. Modify the MPM settings for prefork

   Open the MPM configuration file and edit it , to have only line related to prefork not commented. Make sure worker and event lines are commented.

   vim /etc/httpd/conf.modules.d/00-mpm.conf



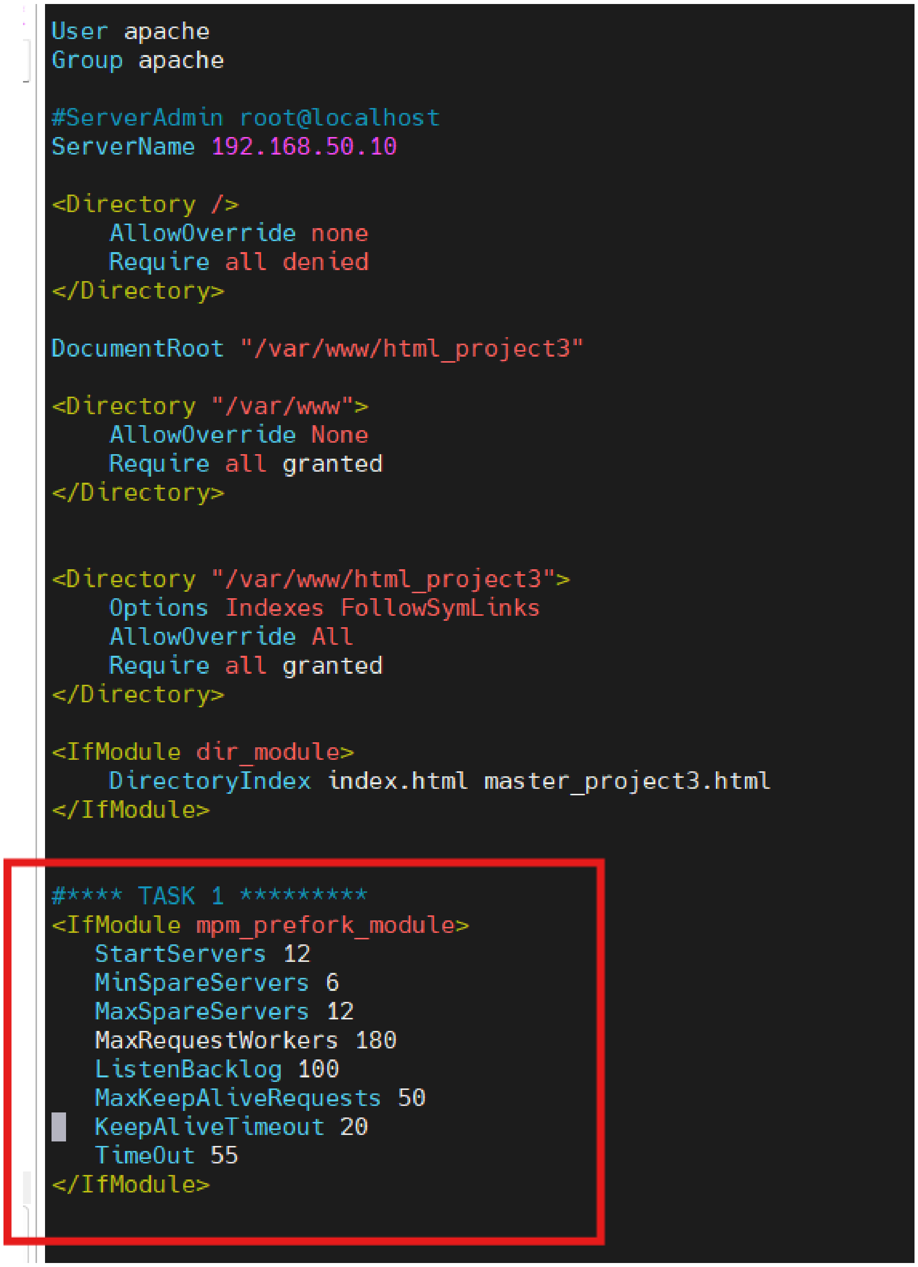
```
# Select the MPM module which should be used by uncommenting exactly
# one of the following LoadModule lines.  See the httpd.conf(5) man
# page for more information on changing the MPM.

# prefork MPM: Implements a non-threaded, pre-forking web server
# See: http://httpd.apache.org/docs/2.4/mod/prefork.html
#
# NOTE: If enabling prefork, the httpd_graceful_shutdown SELinux
# boolean should be enabled, to allow graceful stop/shutdown.
#
LoadModule mpm_prefork_module modules/mod_mpm_prefork.so

# worker MPM: Multi-Processing Module implementing a hybrid
# multi-threaded multi-process web server
# See: http://httpd.apache.org/docs/2.4/mod/worker.html
#
#LoadModule mpm_worker_module modules/mod_mpm_worker.so

# event MPM: A variant of the worker MPM with the goal of consuming
# threads only for connections with active processing
# See: http://httpd.apache.org/docs/2.4/mod/event.html
#
#LoadModule mpm_event_module modules/mod_mpm_event.so
~
~
~
```

2. Edit the main Apache configuration:

   vim /etc/httpd/conf/httpd.conf

3. Add or modify these directives in the main configuration:

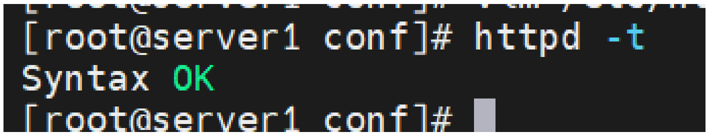
```
<IfModule mpm_prefork_module>
  StartServers 12
  MinSpareServers 6
  MaxSpareServers 12
  MaxRequestWorkers 180
  ListenBacklog 100
  MaxKeepAliveRequests 50
  KeepAliveTimeout 20
  TimeOut 55
</IfModule>
```



```
User apache
Group apache

#ServerAdmin root@localhost
ServerName 192.168.50.10

<Directory />
    AllowOverride none
    Require all denied
</Directory>

DocumentRoot "/var/www/html_project3"

<Directory "/var/www">
    AllowOverride None
    Require all granted
</Directory>


<Directory "/var/www/html_project3">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

<IfModule dir_module>
    DirectoryIndex index.html master_project3.html
</IfModule>


#***** TASK 1 *********
<IfModule mpm_prefork_module>
   StartServers 12
   MinSpareServers 6
   MaxSpareServers 12
   MaxRequestWorkers 180
   ListenBacklog 100
   MaxKeepAliveRequests 50
   KeepAliveTimeout 20
   TimeOut 55
</IfModule>
```

4. Verify the syntax of configuration file after changes

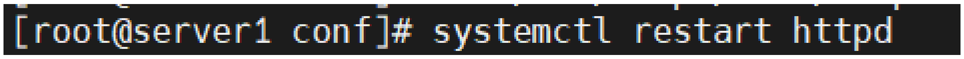
   httpd -t



```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

5. Restart Apache: Apply the configuration changes

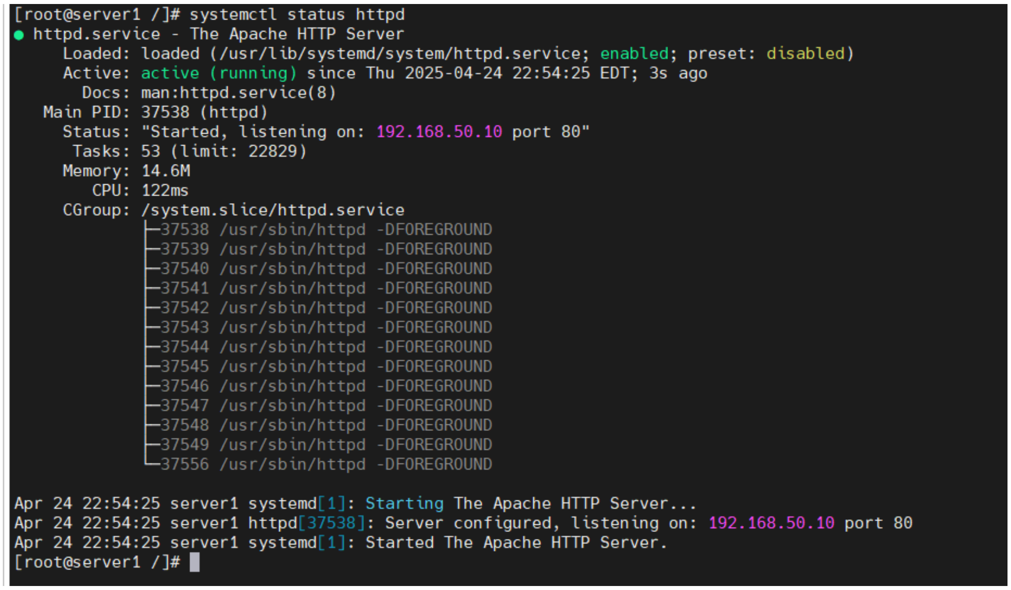
   systemctl restart httpd



```
[root@server1 conf]# systemctl restart httpd
```

6. Verify httpd status

systemctl status httpd



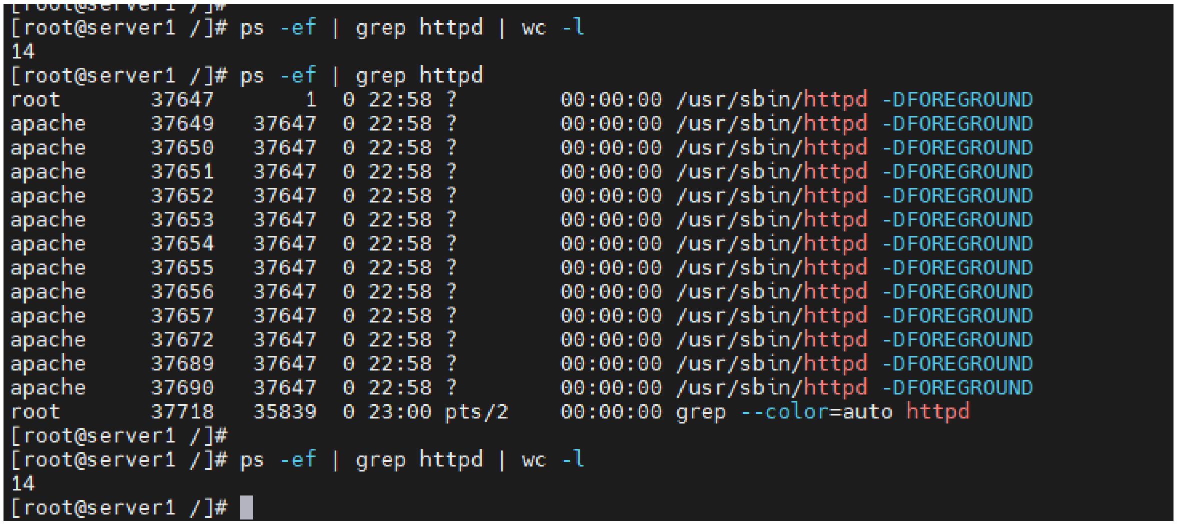
```
[root@server1 /]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-04-24 22:54:25 EDT; 3s ago
       Docs: man:httpd.service(8)
   Main PID: 37538 (httpd)
     Status: "Started, listening on: 192.168.50.10 port 80"
      Tasks: 53 (limit: 22829)
     Memory: 14.6M
        CPU: 122ms
     CGroup: /system.slice/httpd.service
             ├─37538 /usr/sbin/httpd -DFOREGROUND
             ├─37539 /usr/sbin/httpd -DFOREGROUND
             ├─37540 /usr/sbin/httpd -DFOREGROUND
             ├─37541 /usr/sbin/httpd -DFOREGROUND
             ├─37542 /usr/sbin/httpd -DFOREGROUND
             ├─37543 /usr/sbin/httpd -DFOREGROUND
             ├─37544 /usr/sbin/httpd -DFOREGROUND
             ├─37545 /usr/sbin/httpd -DFOREGROUND
             ├─37546 /usr/sbin/httpd -DFOREGROUND
             ├─37547 /usr/sbin/httpd -DFOREGROUND
             ├─37548 /usr/sbin/httpd -DFOREGROUND
             ├─37549 /usr/sbin/httpd -DFOREGROUND
             └─37556 /usr/sbin/httpd -DFOREGROUND

Apr 24 22:54:25 server1 systemd[1]: Starting The Apache HTTP Server...
Apr 24 22:54:25 server1 httpd[37538]: Server configured, listening on: 192.168.50.10 port 80
Apr 24 22:54:25 server1 systemd[1]: Started The Apache HTTP Server.
[root@server1 /]#
```

3.6 Verify the Configuration

1. Check running Apache processes:
   Should show around 12 processes initially

   ps -ef | grep httpd

   ps -ef | grep httpd | wc -l

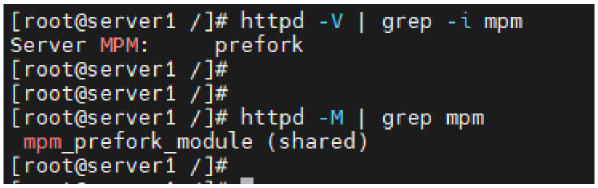


```
[root@server1 /]#
[root@server1 /]# ps -ef | grep httpd | wc -l
14
[root@server1 /]# ps -ef | grep httpd
root       37647       1  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37649   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37650   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37651   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37652   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37653   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37654   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37655   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37656   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37657   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37672   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37689   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     37690   37647  0 22:58 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
root       37718   35839  0 23:00 pts/2    00:00:00 grep --color=auto httpd
[root@server1 /]#
[root@server1 /]# ps -ef | grep httpd | wc -l
14
[root@server1 /]#
```

2. Check the configured values:

   httpd -V | grep -i mpm

   httpd -M | grep mpm



```
[root@server1 /]# httpd -V | grep -i mpm
Server MPM:     prefork
[root@server1 /]#
[root@server1 /]#
[root@server1 /]# httpd -M | grep mpm
 mpm_prefork_module (shared)
[root@server1 /]#
```

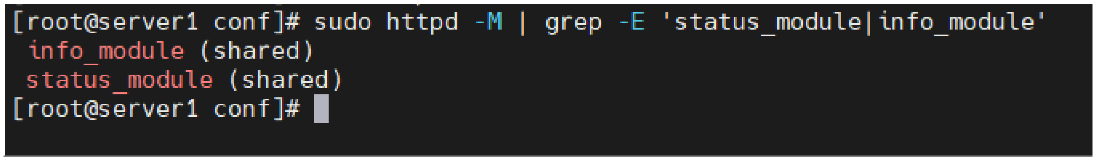
4 TASK 2 – Apache Server Monitoring

4.1 Requirements

- Enable **ExtendedStatus** to allow detailed monitoring of Apache server activity.
- Configure the **server-status** and **server-info handlers** so that they are accessible only to users on the **192.168.50.0/24** subnet.

4.2 Verify modules

sudo httpd -M | grep -E 'status_module|info_module'



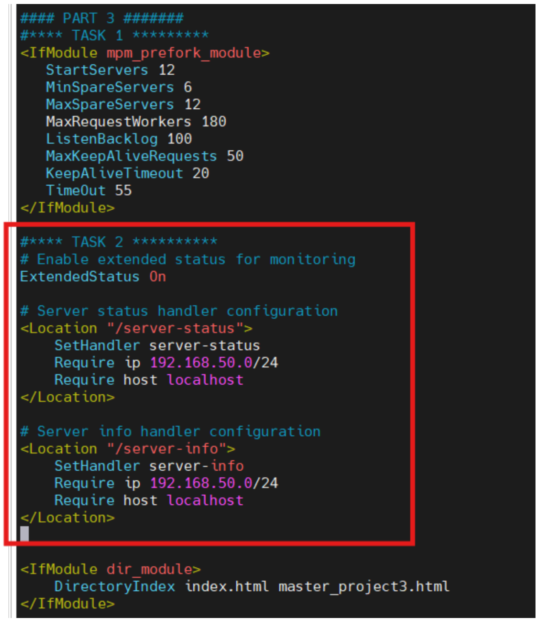
```
[root@server1 conf]# sudo httpd -M | grep -E 'status_module|info_module'
 info_module (shared)
 status_module (shared)
[root@server1 conf]#
```

4.3 Apache Configuration (httpd.conf)

1. Edit the main Apache configuration file and add the following lines

```
#**** TASK 2 **********
# Enable extended status for
monitoring
ExtendedStatus On

# Server status handler
configuration
<Location "/server-status">
    SetHandler server-status
    Require ip 192.168.50.0/24
    Require host localhost
</Location>

# Server info handler
configuration
<Location "/server-info">
    SetHandler server-info
    Require ip 192.168.50.0/24
    Require host localhost
</Location>
```

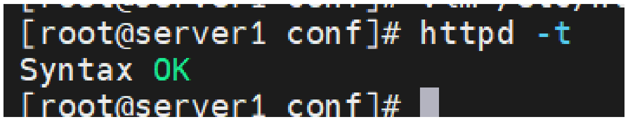
vim /etc/httpd/conf/httpd.conf



```
#### PART 3 #######
#**** TASK 1 *********
<IfModule mpm_prefork_module>
   StartServers 12
   MinSpareServers 6
   MaxSpareServers 12
   MaxRequestWorkers 180
   ListenBacklog 100
   MaxKeepAliveRequests 50
   KeepAliveTimeout 20
   TimeOut 55
</IfModule>

#**** TASK 2 **********
# Enable extended status for monitoring
ExtendedStatus On

# Server status handler configuration
<Location "/server-status">
    SetHandler server-status
    Require ip 192.168.50.0/24
    Require host localhost
</Location>

# Server info handler configuration
<Location "/server-info">
    SetHandler server-info
    Require ip 192.168.50.0/24
    Require host localhost
</Location>


<IfModule dir_module>
    DirectoryIndex index.html master_project3.html
</IfModule>
```

2. Verify the syntax of configuration file after changes

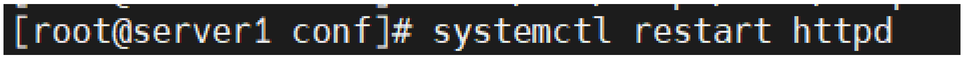
httpd -t



```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

3. Restart Apache: Apply the configuration changes

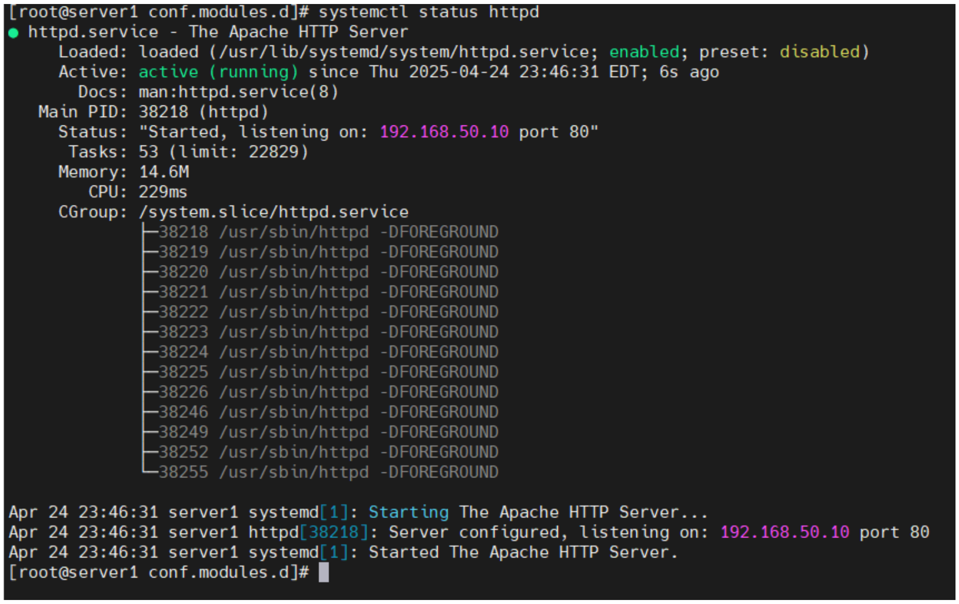
systemctl restart httpd



```
[root@server1 conf]# systemctl restart httpd
```

4. Verify httpd status

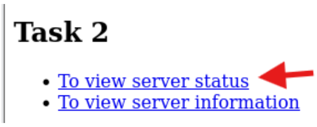
sytemctl status httpd



```
[root@server1 conf.modules.d]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-04-24 23:46:31 EDT; 6s ago
       Docs: man:httpd.service(8)
   Main PID: 38218 (httpd)
     Status: "Started, listening on: 192.168.50.10 port 80"
      Tasks: 53 (limit: 22829)
     Memory: 14.6M
        CPU: 229ms
     CGroup: /system.slice/httpd.service
             ├─38218 /usr/sbin/httpd -DFOREGROUND
             ├─38219 /usr/sbin/httpd -DFOREGROUND
             ├─38220 /usr/sbin/httpd -DFOREGROUND
             ├─38221 /usr/sbin/httpd -DFOREGROUND
             ├─38222 /usr/sbin/httpd -DFOREGROUND
             ├─38223 /usr/sbin/httpd -DFOREGROUND
             ├─38224 /usr/sbin/httpd -DFOREGROUND
             ├─38225 /usr/sbin/httpd -DFOREGROUND
             ├─38226 /usr/sbin/httpd -DFOREGROUND
             ├─38246 /usr/sbin/httpd -DFOREGROUND
             ├─38249 /usr/sbin/httpd -DFOREGROUND
             ├─38252 /usr/sbin/httpd -DFOREGROUND
             └─38255 /usr/sbin/httpd -DFOREGROUND

Apr 24 23:46:31 server1 systemd[1]: Starting The Apache HTTP Server...
Apr 24 23:46:31 server1 httpd[38218]: Server configured, listening on: 192.168.50.10 port 80
Apr 24 23:46:31 server1 systemd[1]: Started The Apache HTTP Server.
[root@server1 conf.modules.d]# 
```

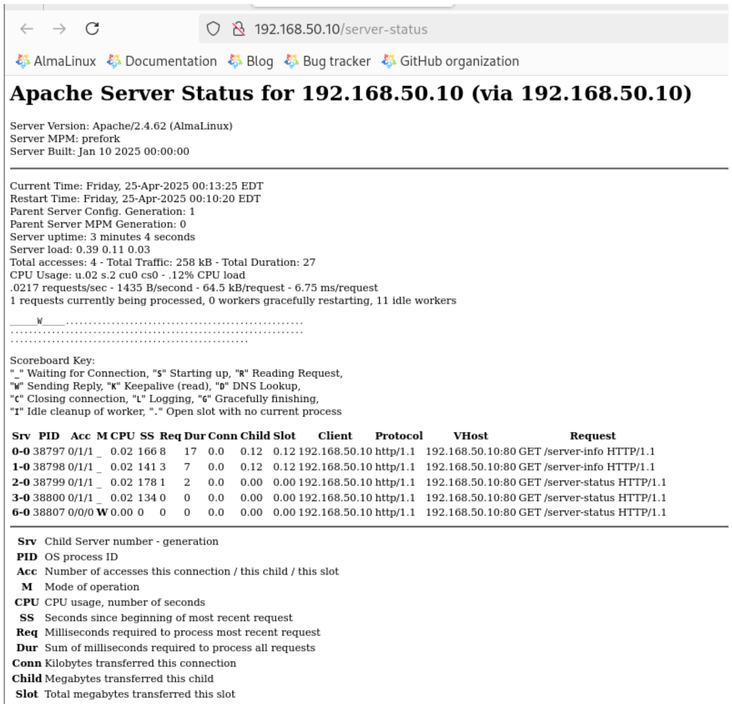
4.4 Test

4.4.1 Server status



Task 2

- To view server status
- To view server information



Apache Server Status for 192.168.50.10 (via 192.168.50.10)

Server Version: Apache/2.4.62 (AlmaLinux)
Server MPM: prefork
Server Built: Jan 10 2025 00:00:00

Current Time: Friday, 25-Apr-2025 00:13:25 EDT
Restart Time: Friday, 25-Apr-2025 00:10:20 EDT
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 3 minutes 4 seconds
Server load: 0.39 0.11 0.03
Total accesses: 4 - Total Traffic: 258 kB - Total Duration: 27
CPU Usage: u.02 s.2 cu0 cs0 - .12% CPU load
.0217 requests/sec - 1435 B/second - 64.5 kB/request - 6.75 ms/request
1 requests currently being processed, 0 workers gracefully restarting, 11 idle workers

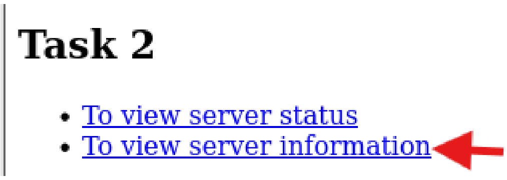
```
______W_____....................................................
................................................................
....................................................
```

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
0-0	38797	0/1/1	_	0.02	166	8	17	0.0	0.12	0.12	192.168.50.10	http/1.1	192.168.50.10:80	GET /server-info HTTP/1.1
1-0	38798	0/1/1	_	0.02	141	3	7	0.0	0.12	0.12	192.168.50.10	http/1.1	192.168.50.10:80	GET /server-info HTTP/1.1
2-0	38799	0/1/1	_	0.02	178	1	2	0.0	0.00	0.00	192.168.50.10	http/1.1	192.168.50.10:80	GET /server-status HTTP/1.1
3-0	38800	0/1/1	_	0.02	134	0	0	0.0	0.00	0.00	192.168.50.10	http/1.1	192.168.50.10:80	GET /server-status HTTP/1.1
6-0	38807	0/0/0	**W**	0.00	0	0	0	0.0	0.00	0.00	192.168.50.10	http/1.1	192.168.50.10:80	GET /server-status HTTP/1.1

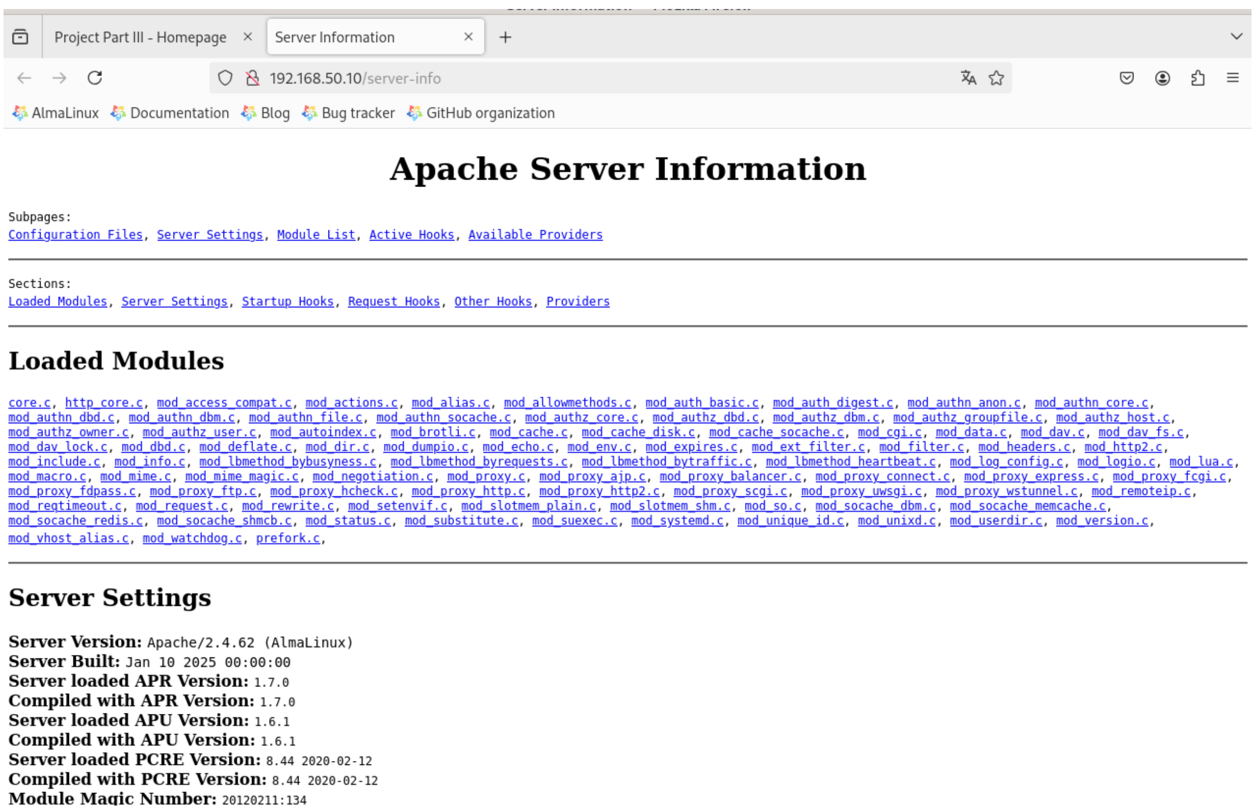
Srv	Child Server number - generation
PID	OS process ID
Acc	Number of accesses this connection / this child / this slot
M	Mode of operation
CPU	CPU usage, number of seconds
SS	Seconds since beginning of most recent request
Req	Milliseconds required to process most recent request
Dur	Sum of milliseconds required to process all requests
Conn	Kilobytes transferred this connection
Child	Megabytes transferred this child
Slot	Total megabytes transferred this slot

4.4.2 Server information



Task 2

- To view server status
- To view server information



Apache Server Information

Subpages:
Configuration Files, Server Settings, Module List, Active Hooks, Available Providers

Sections:
Loaded Modules, Server Settings, Startup Hooks, Request Hooks, Other Hooks, Providers

Loaded Modules

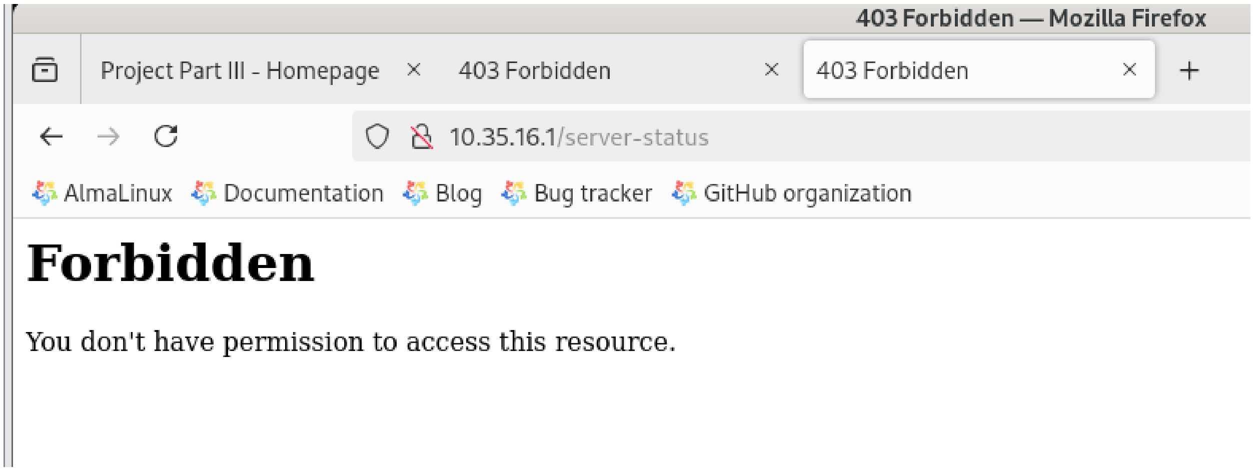
core.c, http_core.c, mod_access_compat.c, mod_actions.c, mod_alias.c, mod_allowmethods.c, mod_auth_basic.c, mod_auth_digest.c, mod_authn_anon.c, mod_authn_core.c, mod_authn_dbd.c, mod_authn_dbm.c, mod_authn_file.c, mod_authn_socache.c, mod_authz_core.c, mod_authz_dbd.c, mod_authz_dbm.c, mod_authz_groupfile.c, mod_authz_host.c, mod_authz_owner.c, mod_authz_user.c, mod_autoindex.c, mod_brotli.c, mod_cache.c, mod_cache_disk.c, mod_cache_socache.c, mod_cgi.c, mod_data.c, mod_dav.c, mod_dav_fs.c, mod_dav_lock.c, mod_dbd.c, mod_deflate.c, mod_dir.c, mod_dumpio.c, mod_echo.c, mod_env.c, mod_expires.c, mod_ext_filter.c, mod_filter.c, mod_headers.c, mod_http2.c, mod_include.c, mod_info.c, mod_lbmethod_bybusyness.c, mod_lbmethod_byrequests.c, mod_lbmethod_bytraffic.c, mod_lbmethod_heartbeat.c, mod_log_config.c, mod_logio.c, mod_lua.c, mod_macro.c, mod_mime.c, mod_mime_magic.c, mod_negotiation.c, mod_proxy.c, mod_proxy_ajp.c, mod_proxy_balancer.c, mod_proxy_connect.c, mod_proxy_express.c, mod_proxy_fcgi.c, mod_proxy_fdpass.c, mod_proxy_ftp.c, mod_proxy_hcheck.c, mod_proxy_http.c, mod_proxy_http2.c, mod_proxy_scgi.c, mod_proxy_uwsgi.c, mod_proxy_wstunnel.c, mod_remoteip.c, mod_reqtimeout.c, mod_request.c, mod_rewrite.c, mod_setenvif.c, mod_slotmem_plain.c, mod_slotmem_shm.c, mod_so.c, mod_socache_dbm.c, mod_socache_memcache.c, mod_socache_redis.c, mod_socache_shmcb.c, mod_status.c, mod_substitute.c, mod_suexec.c, mod_systemd.c, mod_unique_id.c, mod_unixd.c, mod_userdir.c, mod_version.c, mod_vhost_alias.c, mod_watchdog.c, prefork.c,

Server Settings

Server Version: Apache/2.4.62 (AlmaLinux)
Server Built: Jan 10 2025 00:00:00
Server loaded APR Version: 1.7.0
Compiled with APR Version: 1.7.0
Server loaded APU Version: 1.6.1
Compiled with APU Version: 1.6.1
Server loaded PCRE Version: 8.44 2020-02-12
Compiled with PCRE Version: 8.44 2020-02-12
Module Magic Number: 20120211:134

4.4.3 Accessible only to users on the 192.168.50.0/24 subnet

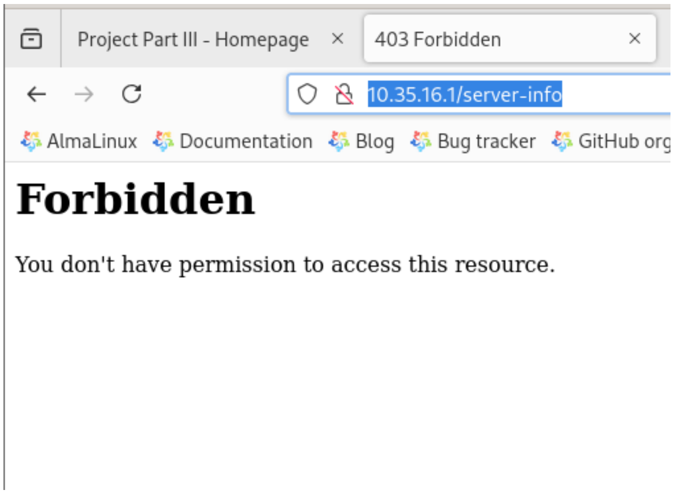
4.4.3.1 Server status



Forbidden

You don't have permission to access this resource.

4.4.3.2 Server-info



Forbidden

You don't have permission to access this resource.

5 TASK 3– CGI

5.1 Requirements

- Configure your web server to use **cgi-bins.**
- Create a **Perl script** that displays the text **“This is Task 3 (CGI)”** in the web browser.
- Place this script in the **/var/www/cgi-bin** directory.

5.2 Apache configuration (httpd.conf)

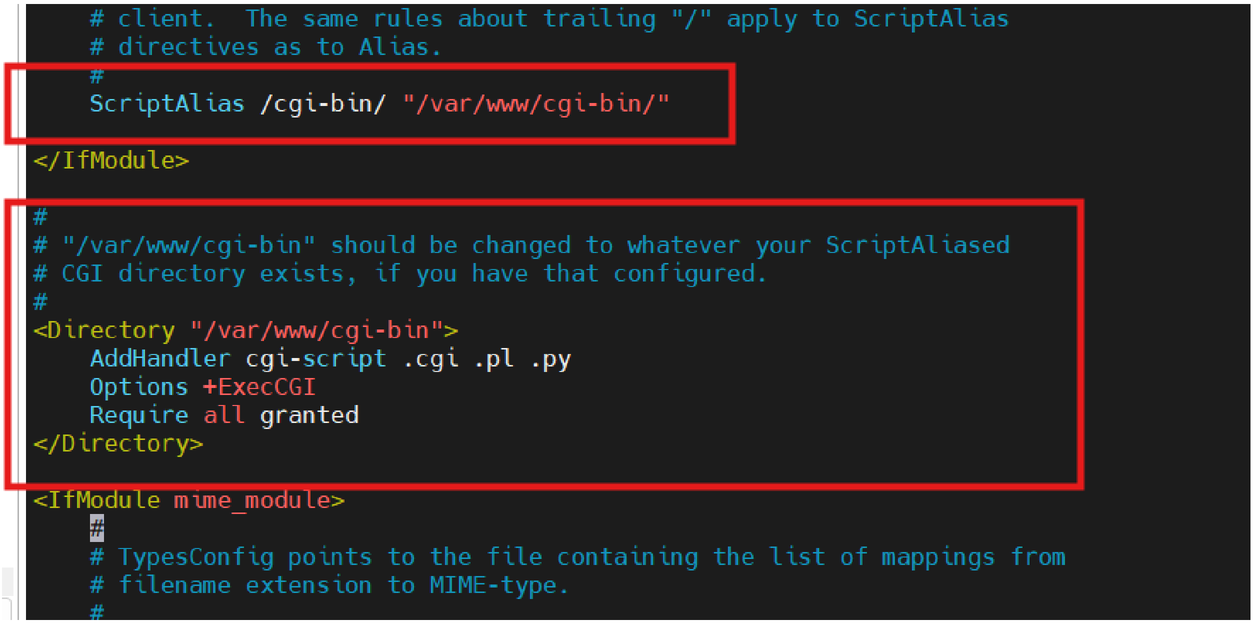
Make sure this line exist

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

Modify httpd.conf and add these lines

```
<Directory "/var/www/cgi-bin">
    AddHandler cgi-script .cgi .pl .py
    Options +ExecCGI
    Require all granted
</Directory>
```

vim /etc/httpd/conf/httpd.conf



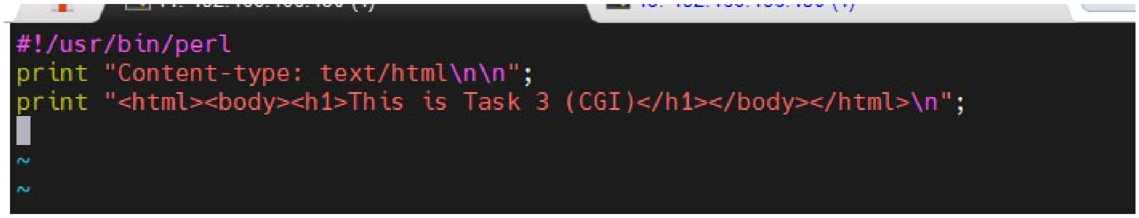
```
    # client.  The same rules about trailing "/" apply to ScriptAlias
    # directives as to Alias.
    #
    ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

</IfModule>

#
# "/var/www/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "/var/www/cgi-bin">
    AddHandler cgi-script .cgi .pl .py
    Options +ExecCGI
    Require all granted
</Directory>

<IfModule mime_module>
    #
    # TypesConfig points to the file containing the list of mappings from
    # filename extension to MIME-type.
    #
```

5.3 Create the CGI Script (perl script)

1. Create the Perl script:

vim /var/www/cgi-bin/task3.pl

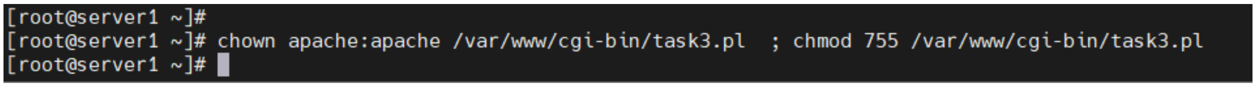
```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "<html><body><h1>This is Task 3 (CGI)</h1></body></html>\n";
```



```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "<html><body><h1>This is Task 3 (CGI)</h1></body></html>\n";

~
~
```

2. Make the script executable and change file permission to the perl script.

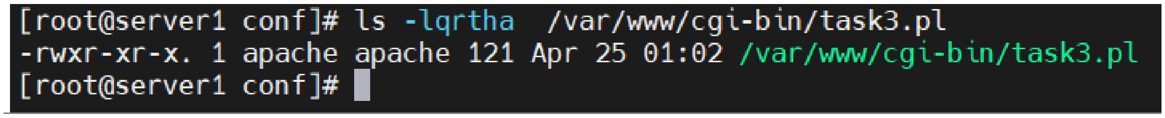
chown apache:apache /var/www/cgi-bin/task3.pl ; chmod 755 /var/www/cgi-bin/task3.pl



```
[root@server1 ~]#
[root@server1 ~]# chown apache:apache /var/www/cgi-bin/task3.pl  ; chmod 755 /var/www/cgi-bin/task3.pl
[root@server1 ~]#
```

3. Verify file task3.pl

ls -lqrtha /var/www/cgi-bin/task3.pl

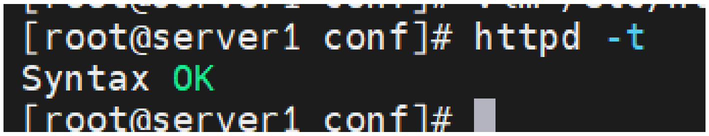


```
[root@server1 conf]# ls -lqrtha  /var/www/cgi-bin/task3.pl
-rwxr-xr-x. 1 apache apache 121 Apr 25 01:02 /var/www/cgi-bin/task3.pl
[root@server1 conf]#
```

5.4 Restart httpd service

1. Verify the syntax of configuration file after changes

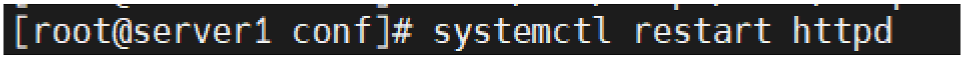
httpd -t



```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

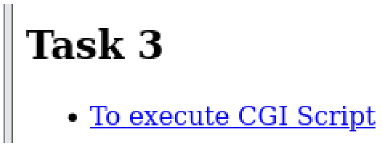
2. Restart Apache: Apply the configuration changes

   systemctl restart httpd



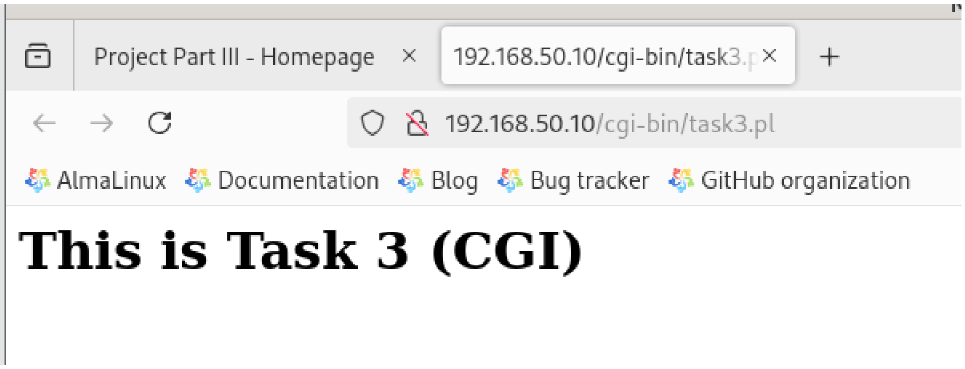
```
[root@server1 conf]# systemctl restart httpd
```

5.5 Test Execute CGI script



Task 3

- To execute CGI Script



Project Part III - Homepage
192.168.50.10/cgi-bin/task3.p
192.168.50.10/cgi-bin/task3.pl
AlmaLinux
Documentation
Blog
Bug tracker
GitHub organization

This is Task 3 (CGI)

6 Task 4 – PHP

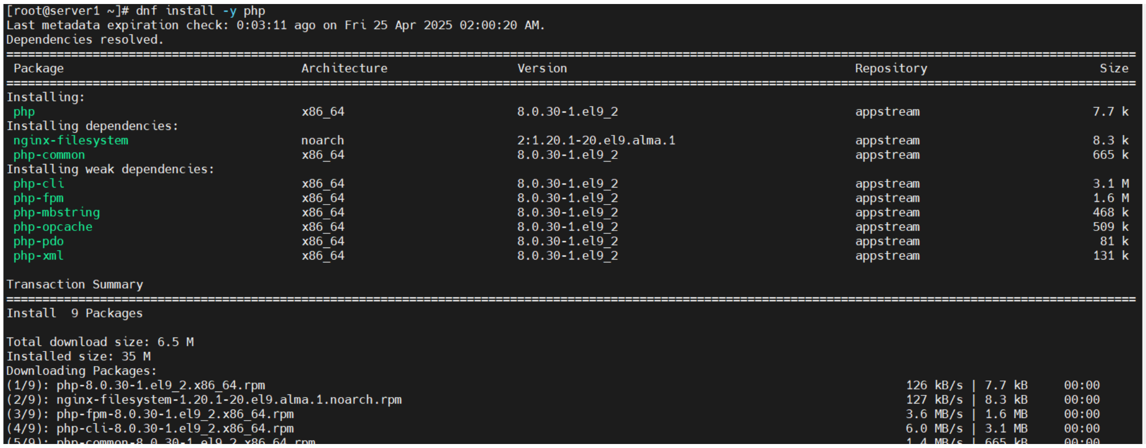
6.1 Requirements

- Configure your web server to use **PHP**.
- Create a **PHP script** that displays the message **"Hello my friend!"** in the **web browser** for users whose IP address belongs to the **192.168.100.0/24** subnet and displays **"Hello stranger!"** for all other users outside this subnet.
- Place this PHP script in the **/var/www/html_project3/q4** directory.

6.2 Install PHP

1. Install PHP and Start the Apache Service

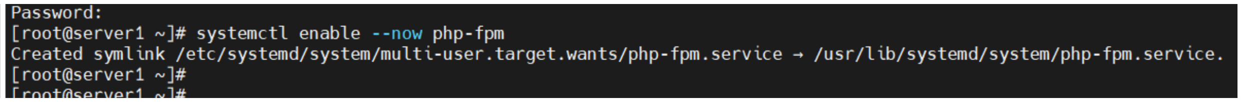
dnf install -y php



```
[root@server1 ~]# dnf install -y php
Last metadata expiration check: 0:03:11 ago on Fri 25 Apr 2025 02:00:20 AM.
Dependencies resolved.
================================================================================================================================
 Package                         Architecture              Version                                Repository               Size
================================================================================================================================
Installing:
 php                             x86_64                    8.0.30-1.el9_2                         appstream               7.7 k
Installing dependencies:
 nginx-filesystem                noarch                    2:1.20.1-20.el9.alma.1                 appstream               8.3 k
 php-common                      x86_64                    8.0.30-1.el9_2                         appstream               665 k
Installing weak dependencies:
 php-cli                         x86_64                    8.0.30-1.el9_2                         appstream               3.1 M
 php-fpm                         x86_64                    8.0.30-1.el9_2                         appstream               1.6 M
 php-mbstring                    x86_64                    8.0.30-1.el9_2                         appstream               468 k
 php-opcache                     x86_64                    8.0.30-1.el9_2                         appstream               509 k
 php-pdo                         x86_64                    8.0.30-1.el9_2                         appstream                81 k
 php-xml                         x86_64                    8.0.30-1.el9_2                         appstream               131 k

Transaction Summary
================================================================================================================================
Install  9 Packages

Total download size: 6.5 M
Installed size: 35 M
Downloading Packages:
(1/9): php-8.0.30-1.el9_2.x86_64.rpm                                                          126 kB/s | 7.7 kB     00:00
(2/9): nginx-filesystem-1.20.1-20.el9.alma.1.noarch.rpm                                       127 kB/s | 8.3 kB     00:00
(3/9): php-fpm-8.0.30-1.el9_2.x86_64.rpm                                                      3.6 MB/s | 1.6 MB     00:00
(4/9): php-cli-8.0.30-1.el9_2.x86_64.rpm                                                      6.0 MB/s | 3.1 MB     00:00
(5/9): php-common-8.0.30-1.el9_2.x86_64.rpm                                                   1.4 MB/s | 665 kB     00:00
```

2. Enable PHP

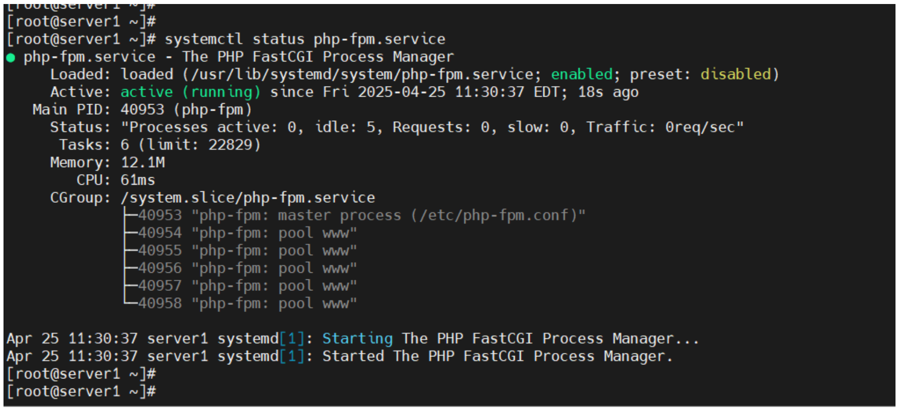
systemctl enable –now php-fpm



```
Password:
[root@server1 ~]# systemctl enable --now php-fpm
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/systemd/system/php-fpm.service.
[root@server1 ~]#
[root@server1 ~]#
```

3. Verify status

systemctl status php-fpm.service

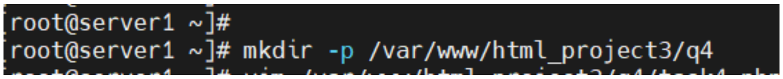


```
[root@server1 ~]#
[root@server1 ~]# systemctl status php-fpm.service
● php-fpm.service - The PHP FastCGI Process Manager
     Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-04-25 11:30:37 EDT; 18s ago
   Main PID: 40953 (php-fpm)
     Status: "Processes active: 0, idle: 5, Requests: 0, slow: 0, Traffic: 0req/sec"
      Tasks: 6 (limit: 22829)
     Memory: 12.1M
        CPU: 61ms
     CGroup: /system.slice/php-fpm.service
             ├─40953 "php-fpm: master process (/etc/php-fpm.conf)"
             ├─40954 "php-fpm: pool www"
             ├─40955 "php-fpm: pool www"
             ├─40956 "php-fpm: pool www"
             ├─40957 "php-fpm: pool www"
             └─40958 "php-fpm: pool www"

Apr 25 11:30:37 server1 systemd[1]: Starting The PHP FastCGI Process Manager...
Apr 25 11:30:37 server1 systemd[1]: Started The PHP FastCGI Process Manager.
[root@server1 ~]#
[root@server1 ~]#
```

6.3 PHP script

1. Create the PHP Script Directory

   mkdir -p /var/www/html_project3/q4

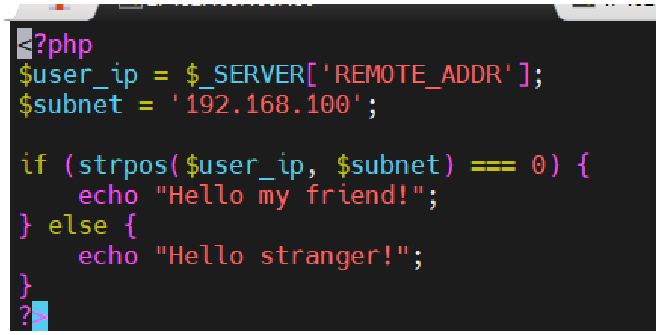


```
[root@server1 ~]#
[root@server1 ~]# mkdir -p /var/www/html_project3/q4
```

2. Create the PHP Script
   vim /var/www/html_project3/q4/task4.php

   Add this content:

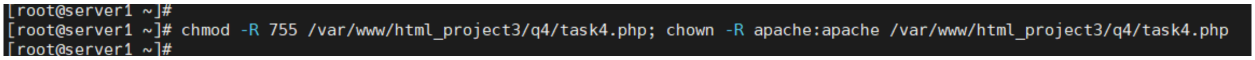
```
<?php
$user_ip = $_SERVER['REMOTE_ADDR'];
$subnet = '192.168.100';

if (strpos($user_ip, $subnet) === 0) {
    echo "Hello my friend!";
} else {
    echo "Hello stranger!";
}
?>
```



```
<?php
$user_ip = $_SERVER['REMOTE_ADDR'];
$subnet = '192.168.100';

if (strpos($user_ip, $subnet) === 0) {
    echo "Hello my friend!";
} else {
    echo "Hello stranger!";
}
?>
```

3. Make the script executable and change file permissions.

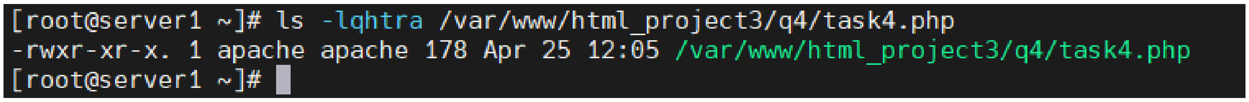
chmod -R 755 /var/www/html_project3/q4/task4.php; chown -R apache:apache /var/www/html_project3/q4/task4.php



```
[root@server1 ~]#
[root@server1 ~]# chmod -R 755 /var/www/html_project3/q4/task4.php; chown -R apache:apache /var/www/html_project3/q4/task4.php
[root@server1 ~]#
```

4. Verify file status

   ls -lqhtra /var/www/html_project3/q4/task4.php



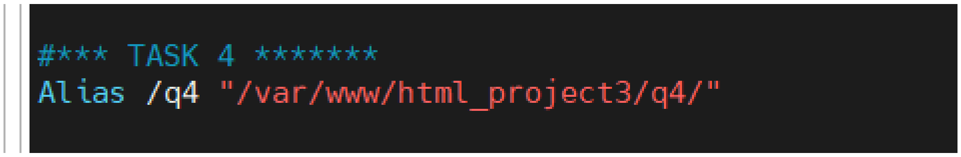
```
[root@server1 ~]# ls -lqhtra /var/www/html_project3/q4/task4.php
-rwxr-xr-x. 1 apache apache 178 Apr 25 12:05 /var/www/html_project3/q4/task4.php
[root@server1 ~]#
```

6.4 Apache configuration (httpd.conf)

1. Use an Alias for the PHP Directory, add this line to httpd.conf

```
Alias /q4 "/var/www/html_project3/q4/"
```

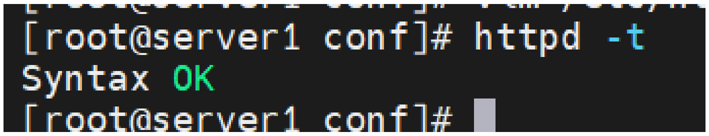
vim /etc/httpd/conf/httpd.conf



```
#*** TASK 4 *******
Alias /q4 "/var/www/html_project3/q4/"
```

2. Verify the syntax of configuration file after changes

   httpd -t



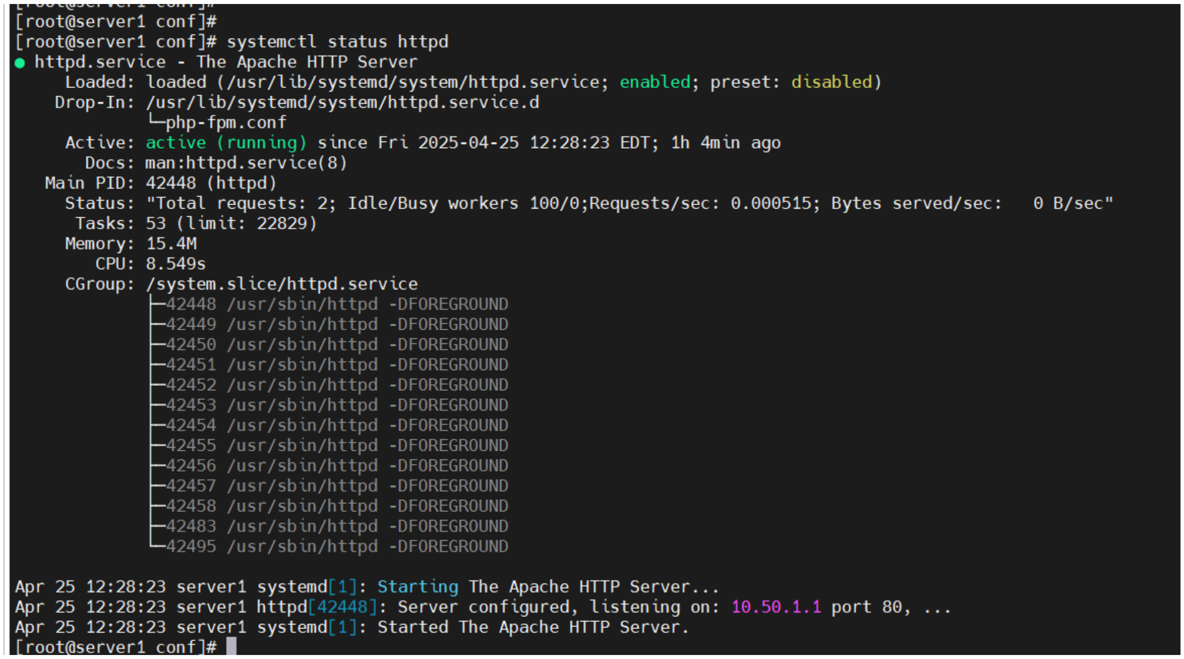
```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

3. Restart Apache: Apply the configuration changes

   systemctl restart httpd

```
[root@server1 conf]# systemctl restart httpd
```

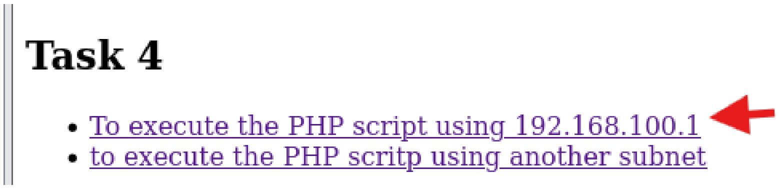
4. **Verify httpd service status**



```
[root@server1 conf]#
[root@server1 conf]#
[root@server1 conf]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Fri 2025-04-25 12:28:23 EDT; 1h 4min ago
       Docs: man:httpd.service(8)
   Main PID: 42448 (httpd)
     Status: "Total requests: 2; Idle/Busy workers 100/0;Requests/sec: 0.000515; Bytes served/sec:   0 B/sec"
      Tasks: 53 (limit: 22829)
     Memory: 15.4M
        CPU: 8.549s
     CGroup: /system.slice/httpd.service
             ├─42448 /usr/sbin/httpd -DFOREGROUND
             ├─42449 /usr/sbin/httpd -DFOREGROUND
             ├─42450 /usr/sbin/httpd -DFOREGROUND
             ├─42451 /usr/sbin/httpd -DFOREGROUND
             ├─42452 /usr/sbin/httpd -DFOREGROUND
             ├─42453 /usr/sbin/httpd -DFOREGROUND
             ├─42454 /usr/sbin/httpd -DFOREGROUND
             ├─42455 /usr/sbin/httpd -DFOREGROUND
             ├─42456 /usr/sbin/httpd -DFOREGROUND
             ├─42457 /usr/sbin/httpd -DFOREGROUND
             ├─42458 /usr/sbin/httpd -DFOREGROUND
             ├─42483 /usr/sbin/httpd -DFOREGROUND
             └─42495 /usr/sbin/httpd -DFOREGROUND

Apr 25 12:28:23 server1 systemd[1]: Starting The Apache HTTP Server...
Apr 25 12:28:23 server1 httpd[42448]: Server configured, listening on: 10.50.1.1 port 80, ...
Apr 25 12:28:23 server1 systemd[1]: Started The Apache HTTP Server.
[root@server1 conf]#
```

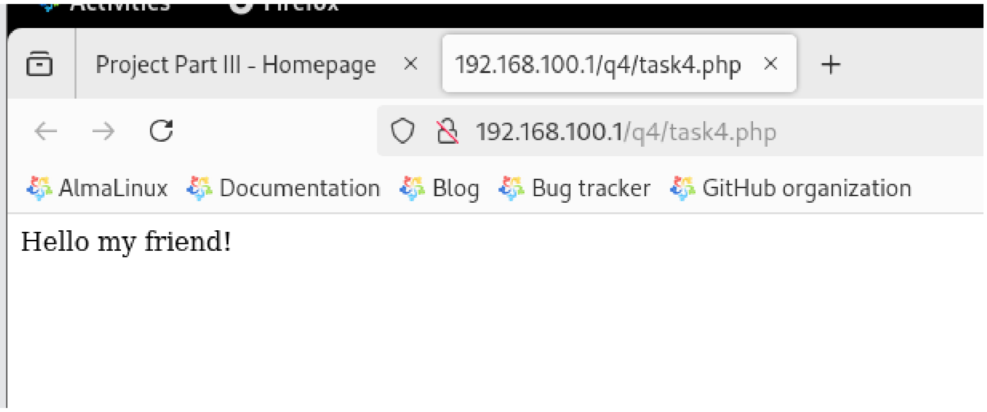
6.5 Test

6.5.1 Execute the PHP script using 192.168.100.1



Task 4

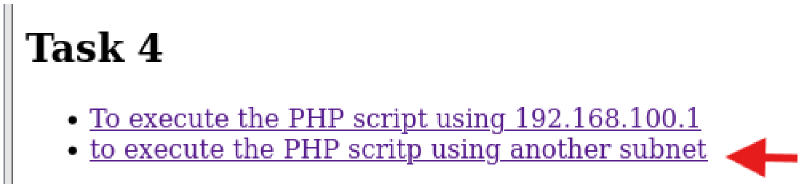
- To execute the PHP script using 192.168.100.1
- to execute the PHP scritp using another subnet



Activities
Firefox
Project Part III - Homepage
192.168.100.1/q4/task4.php
192.168.100.1/q4/task4.php
AlmaLinux
Documentation
Blog
Bug tracker
GitHub organization

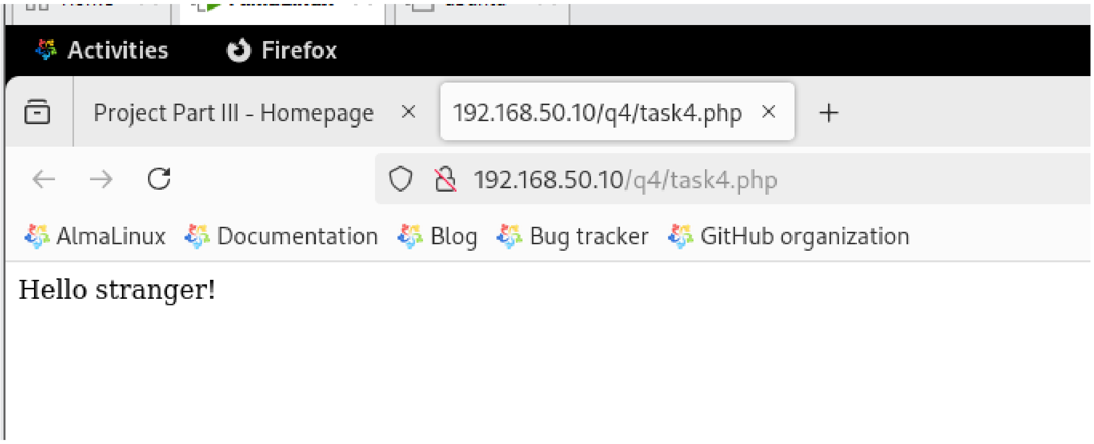
Hello my friend!

6.5.2 Execute the PHP script using another subnet\



Task 4

- To execute the PHP script using 192.168.100.1
- to execute the PHP scritp using another subnet



Activities
Firefox
Project Part III - Homepage
192.168.50.10/q4/task4.php
192.168.50.10/q4/task4.php
AlmaLinux
Documentation
Blog
Bug tracker
GitHub organization

Hello stranger!

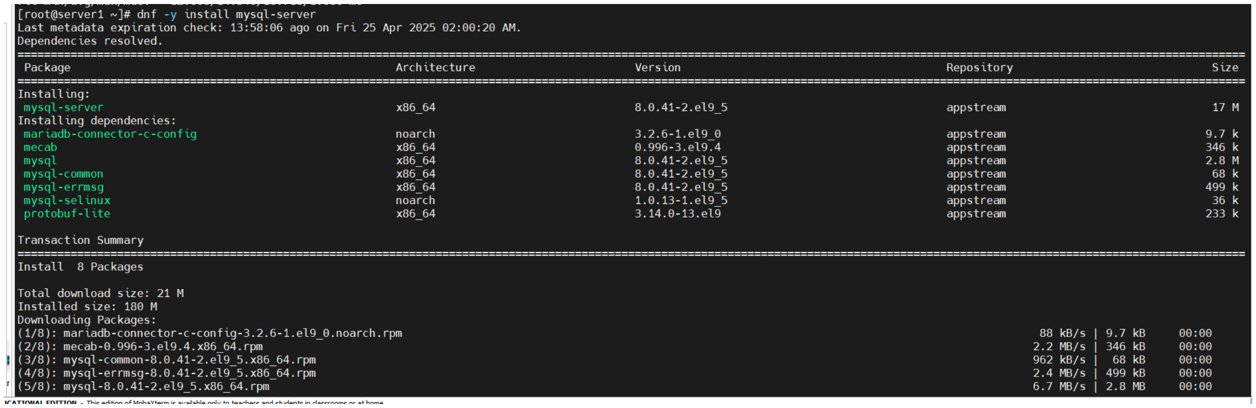
7 TASK 5 – MYSQL/PHP

7.1 Requirements

1. Install **MySQL** and create a database **company**.
2. Create in this database an "**employees**" table that contains two fields: **name** and **salary**.
3. Insert several records into the table, then verify the contents.
4. Create a **PHP script** that connects to this database and displays the contents of the **employees** table in the web browser as an **HTML table**.
5. Place this PHP script in the **/var/www/html_project3/q5** directory**.**

7.2 Installing MySQL Server

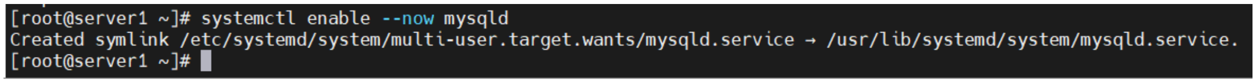
1. Begin by installing the MySQL server package using the following command:
   dnf -y install mysql-server



```
[root@server1 ~]# dnf -y install mysql-server
Last metadata expiration check: 13:58:06 ago on Fri 25 Apr 2025 02:00:20 AM.
Dependencies resolved.
==========================================================================================================================================================
 Package                                   Architecture                Version                             Repository                                Size
==========================================================================================================================================================
Installing:
 mysql-server                              x86_64                      8.0.41-2.el9_5                      appstream                                 17 M
Installing dependencies:
 mariadb-connector-c-config                noarch                      3.2.6-1.el9_0                       appstream                                9.7 k
 mecab                                     x86_64                      0.996-3.el9.4                       appstream                                346 k
 mysql                                     x86_64                      8.0.41-2.el9_5                      appstream                                2.8 M
 mysql-common                              x86_64                      8.0.41-2.el9_5                      appstream                                 68 k
 mysql-errmsg                              x86_64                      8.0.41-2.el9_5                      appstream                                499 k
 mysql-selinux                             noarch                      1.0.13-1.el9_5                      appstream                                 36 k
 protobuf-lite                             x86_64                      3.14.0-13.el9                       appstream                                233 k

Transaction Summary
==========================================================================================================================================================
Install  8 Packages

Total download size: 21 M
Installed size: 180 M
Downloading Packages:
(1/8): mariadb-connector-c-config-3.2.6-1.el9_0.noarch.rpm                                                            88 kB/s | 9.7 kB     00:00
(2/8): mecab-0.996-3.el9.4.x86_64.rpm                                                                                2.2 MB/s | 346 kB     00:00
(3/8): mysql-common-8.0.41-2.el9_5.x86_64.rpm                                                                        962 kB/s |  68 kB     00:00
(4/8): mysql-errmsg-8.0.41-2.el9_5.x86_64.rpm                                                                        2.4 MB/s | 499 kB     00:00
(5/8): mysql-8.0.41-2.el9_5.x86_64.rpm                                                                               6.7 MB/s | 2.8 MB     00:00
```

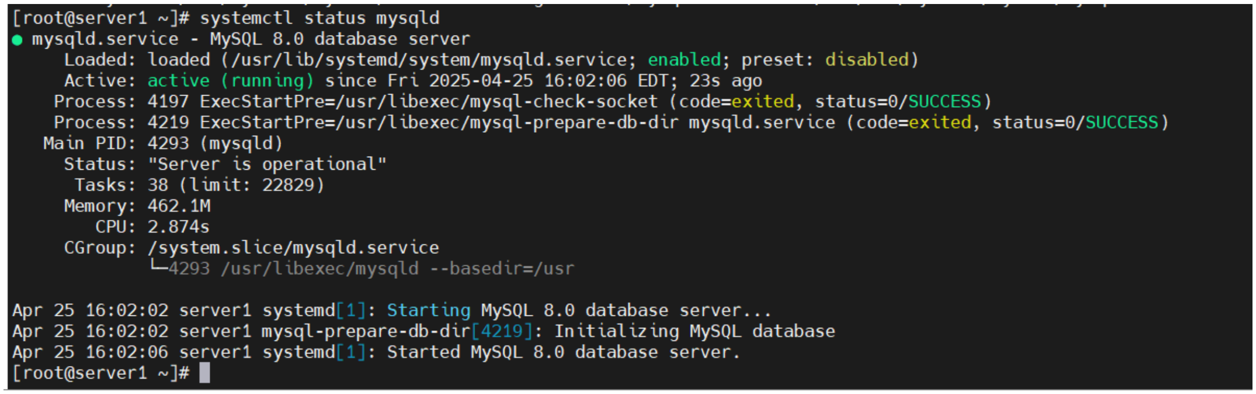
2. Start and enable the MySQL service to launch at boot time:

   systemctl enable --now mysqld



```
[root@server1 ~]# systemctl enable --now mysqld
Created symlink /etc/systemd/system/multi-user.target.wants/mysqld.service → /usr/lib/systemd/system/mysqld.service.
[root@server1 ~]#
```

3. Check the status of the MySQL service:
   systemctl status mysqld



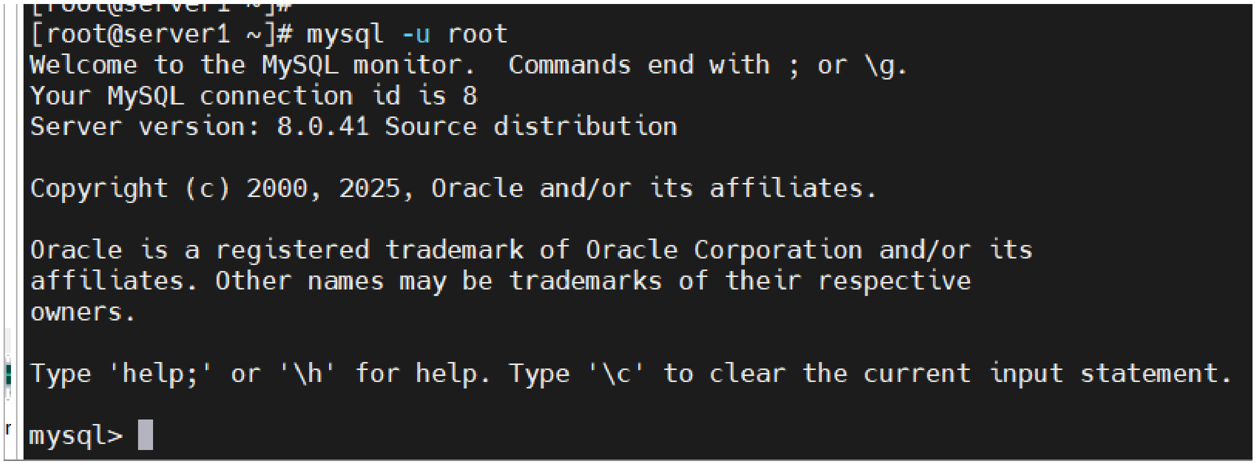
```
[root@server1 ~]# systemctl status mysqld
● mysqld.service - MySQL 8.0 database server
     Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-04-25 16:02:06 EDT; 23s ago
    Process: 4197 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, status=0/SUCCESS)
    Process: 4219 ExecStartPre=/usr/libexec/mysql-prepare-db-dir mysqld.service (code=exited, status=0/SUCCESS)
   Main PID: 4293 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 22829)
     Memory: 462.1M
        CPU: 2.874s
     CGroup: /system.slice/mysqld.service
             └─4293 /usr/libexec/mysqld --basedir=/usr

Apr 25 16:02:02 server1 systemd[1]: Starting MySQL 8.0 database server...
Apr 25 16:02:02 server1 mysql-prepare-db-dir[4219]: Initializing MySQL database
Apr 25 16:02:06 server1 systemd[1]: Started MySQL 8.0 database server.
[root@server1 ~]#
```

7.3 MySQL

7.3.1 Connecting to MySQL

Access the MySQL command-line interface as root:

mysql -u root

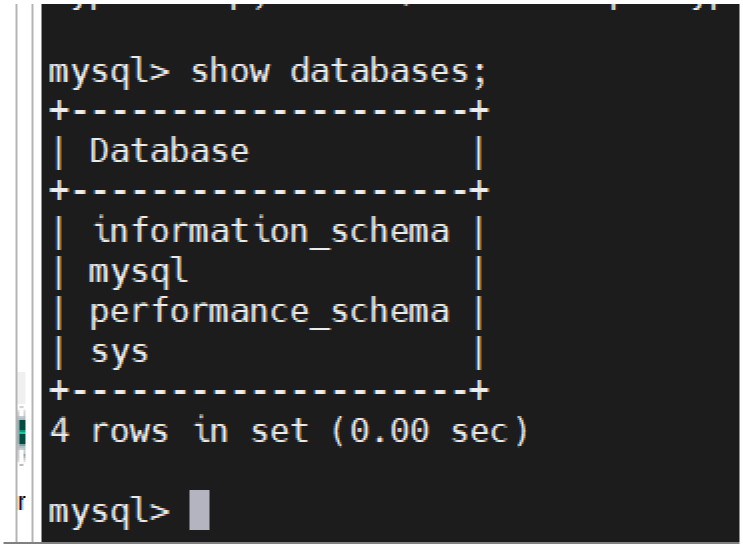


```
[root@server1 ~]#
[root@server1 ~]# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

7.3.2 Creating the Database

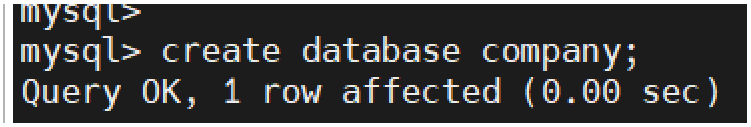
1. List existing databases:

   show databases;



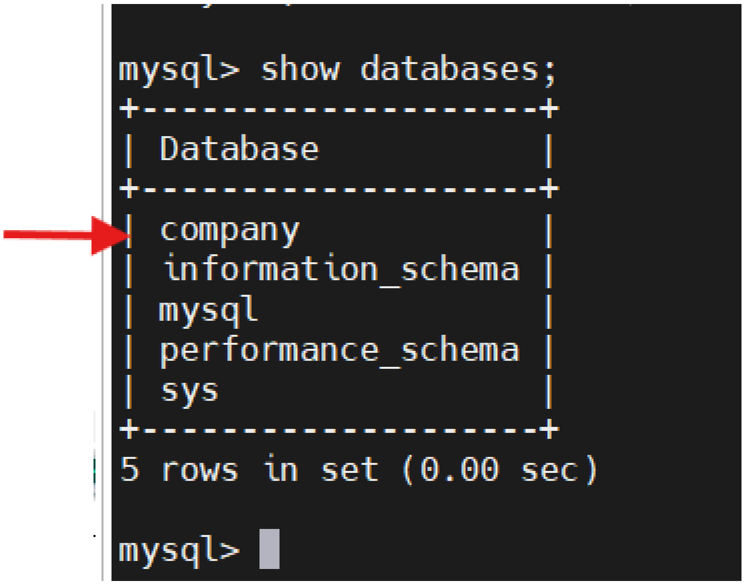
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

2. Create a new database named 'company':
   create database company;



```
mysql>
mysql> create database company;
Query OK, 1 row affected (0.00 sec)
```

3. Verify its creation:
   show databases;

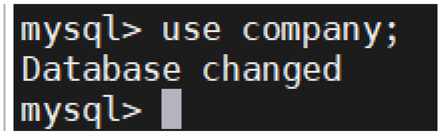


```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| company            |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql>
```

7.3.3 Creating the Employees Table

1. Select the 'company' database:

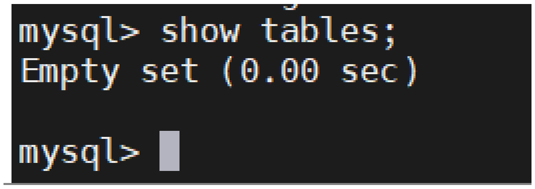
   use company;



```
mysql> use company;
Database changed
mysql>
```

2. List the existing tables:

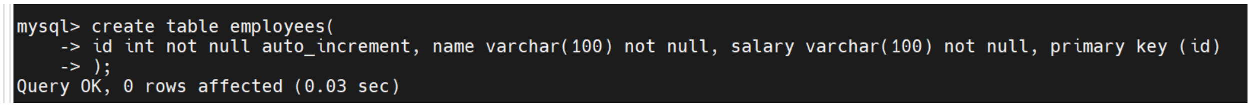
   show tables;



```
mysql> show tables;
Empty set (0.00 sec)

mysql>
```

3. Create the 'employees' table:

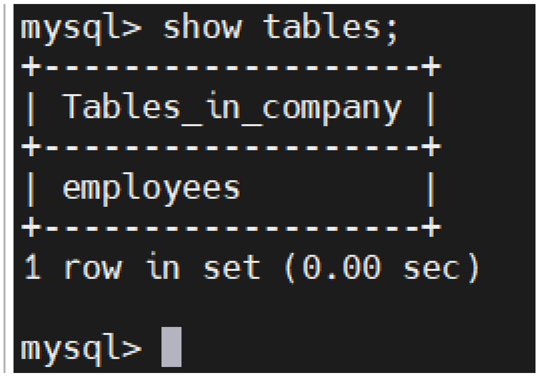
```
create table employees(
id int not null auto_increment, name varchar(100) not null,
salary varchar(100) not null, primary key (id)
);
```



```
mysql> create table employees(
    -> id int not null auto_increment, name varchar(100) not null, salary varchar(100) not null, primary key (id)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

4. Verify the table creation:

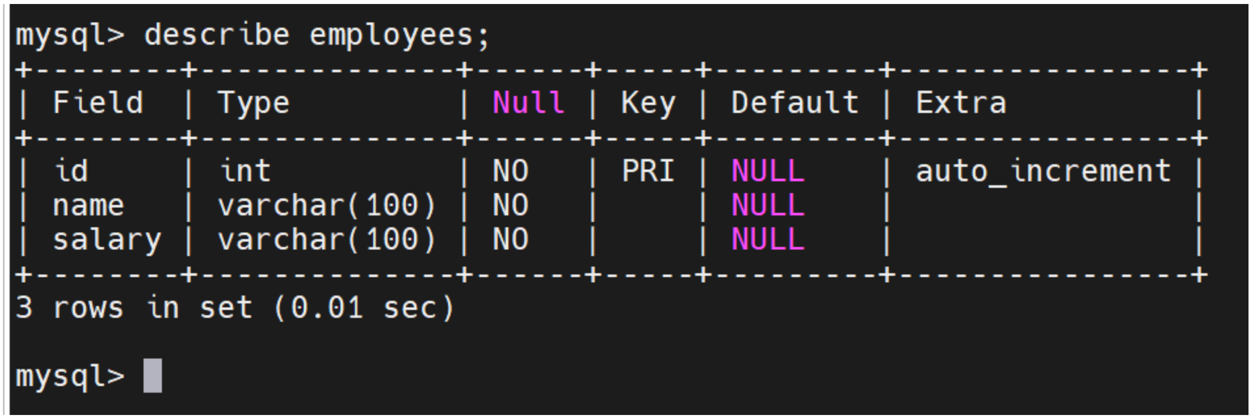
   show tables;



```
mysql> show tables;
+-------------------+
| Tables_in_company |
+-------------------+
| employees         |
+-------------------+
1 row in set (0.00 sec)

mysql>
```

5. Display the table structure:

   describe employees;

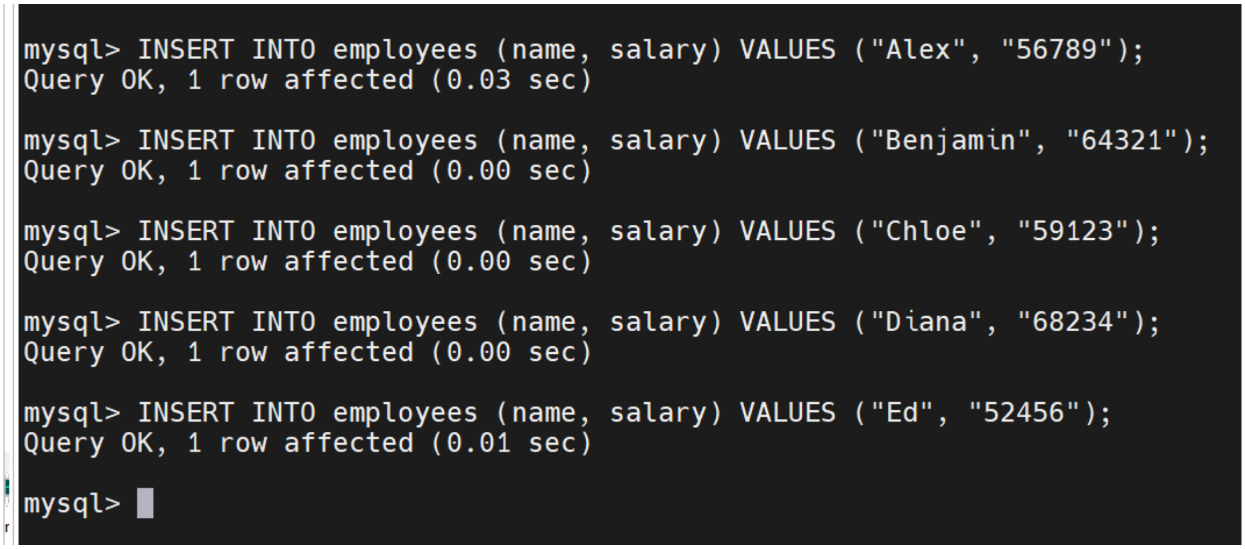


```
mysql> describe employees;
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| id     | int          | NO   | PRI | NULL    | auto_increment |
| name   | varchar(100) | NO   |     | NULL    |                |
| salary | varchar(100) | NO   |     | NULL    |                |
+--------+--------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)

mysql> 
```

7.3.4 Populate database

1. Insert records into the 'employees' table:

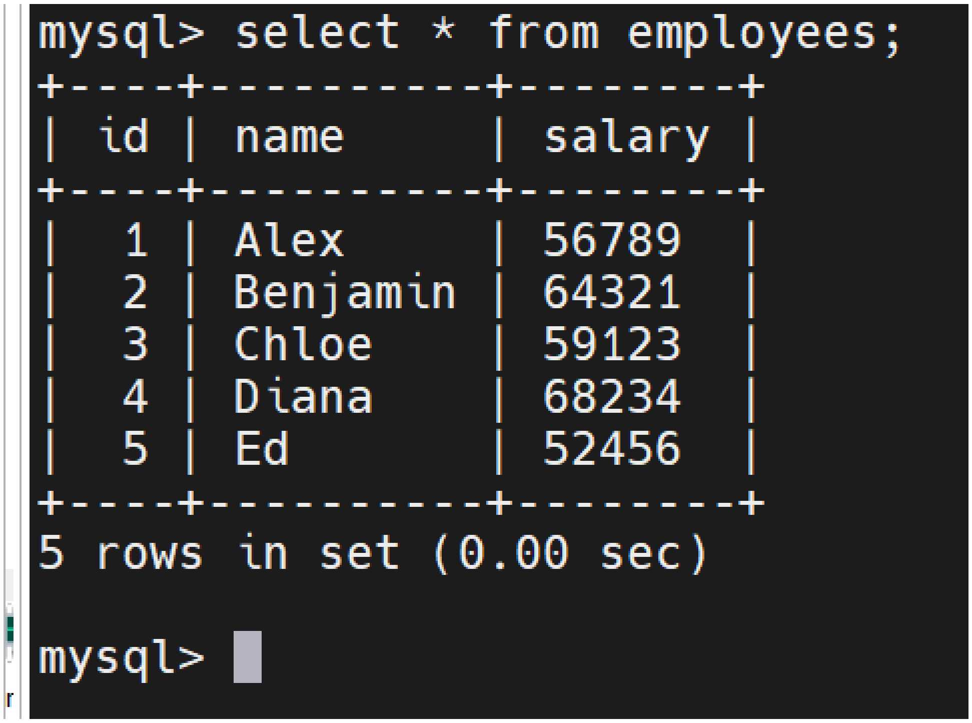
```
INSERT INTO employees (name, salary) VALUES ("Alex", "56789");
INSERT INTO employees (name, salary) VALUES ("Benjamin", "64321");
INSERT INTO employees (name, salary) VALUES ("Chloe", "59123");
INSERT INTO employees (name, salary) VALUES ("Diana", "68234");
INSERT INTO employees (name, salary) VALUES ("Ed", "52456");
```



```
mysql> INSERT INTO employees (name, salary) VALUES ("Alex", "56789");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO employees (name, salary) VALUES ("Benjamin", "64321");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employees (name, salary) VALUES ("Chloe", "59123");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employees (name, salary) VALUES ("Diana", "68234");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employees (name, salary) VALUES ("Ed", "52456");
Query OK, 1 row affected (0.01 sec)

mysql> 
```

2. Check the contents of the table:

```
select * from employees;
```

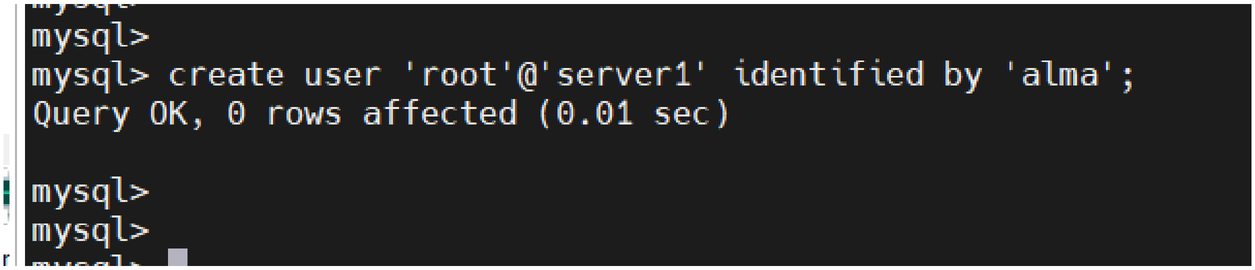


```
mysql> select * from employees;
+----+----------+--------+
| id | name     | salary |
+----+----------+--------+
|  1 | Alex     | 56789  |
|  2 | Benjamin | 64321  |
|  3 | Chloe    | 59123  |
|  4 | Diana    | 68234  |
|  5 | Ed       | 52456  |
+----+----------+--------+
5 rows in set (0.00 sec)

mysql>
```

7.3.5 Granting Privileges

1. Create a user with full access from a remote host:

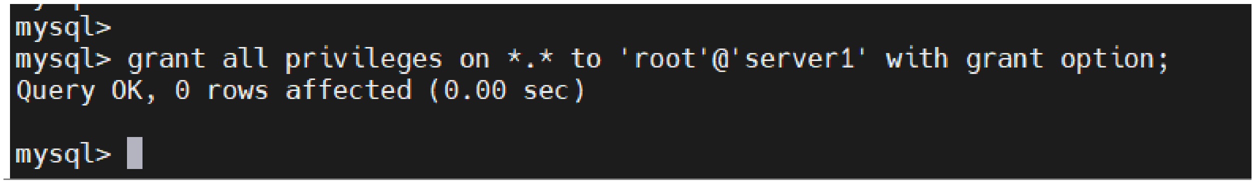
   create user 'root'@'server1' identified by 'alma';



```
mysql>
mysql> create user 'root'@'server1' identified by 'alma';
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql>
```

2. Grant all privileges to this user:

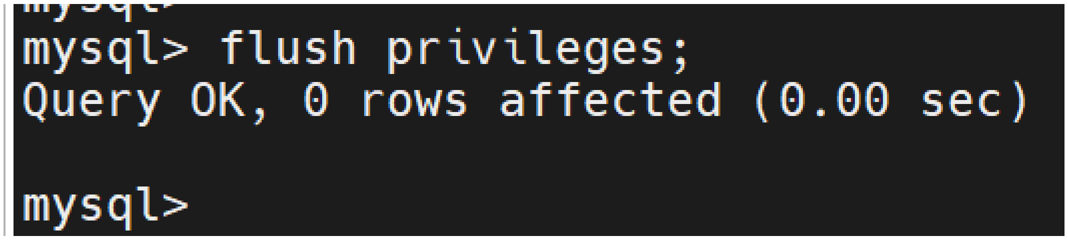
   grant all privileges on *.* to 'root'@'server1' with grant option;



```
mysql>
mysql> grant all privileges on *.* to 'root'@'server1' with grant option;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

3. Apply the changes:

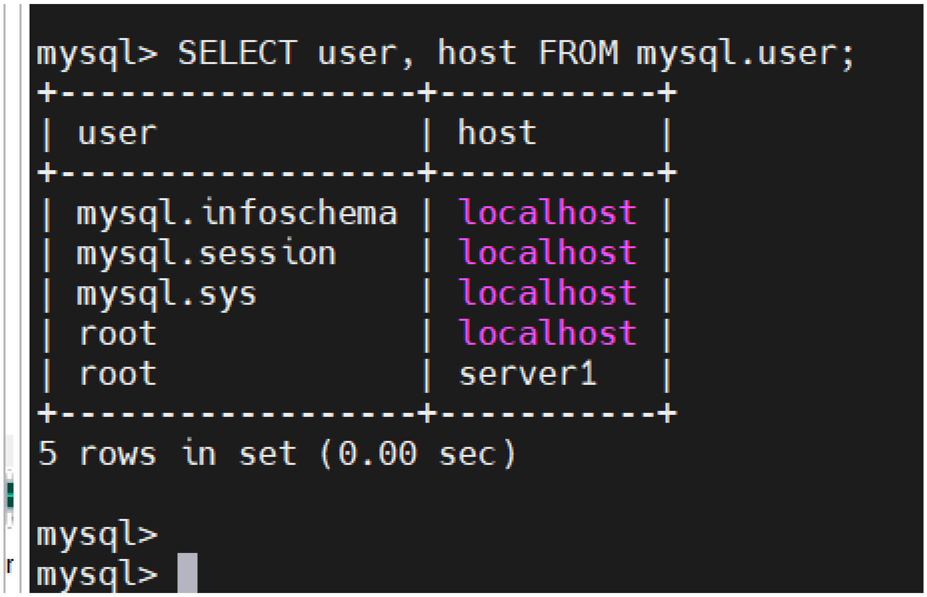
   flush privileges;



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

mysql>
```

4. Verify user is created

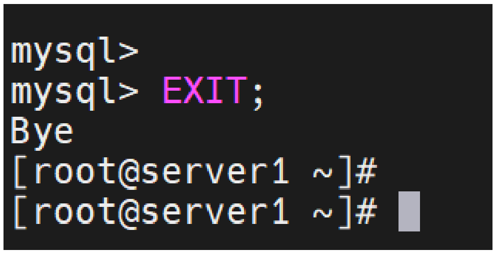
   SELECT user, host FROM mysql.user;



```
mysql> SELECT user, host FROM mysql.user;
+------------------+-----------+
| user             | host      |
+------------------+-----------+
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
| root             | server1   |
+------------------+-----------+
5 rows in set (0.00 sec)

mysql>
mysql>
```

5. Exit the MySQL shell:

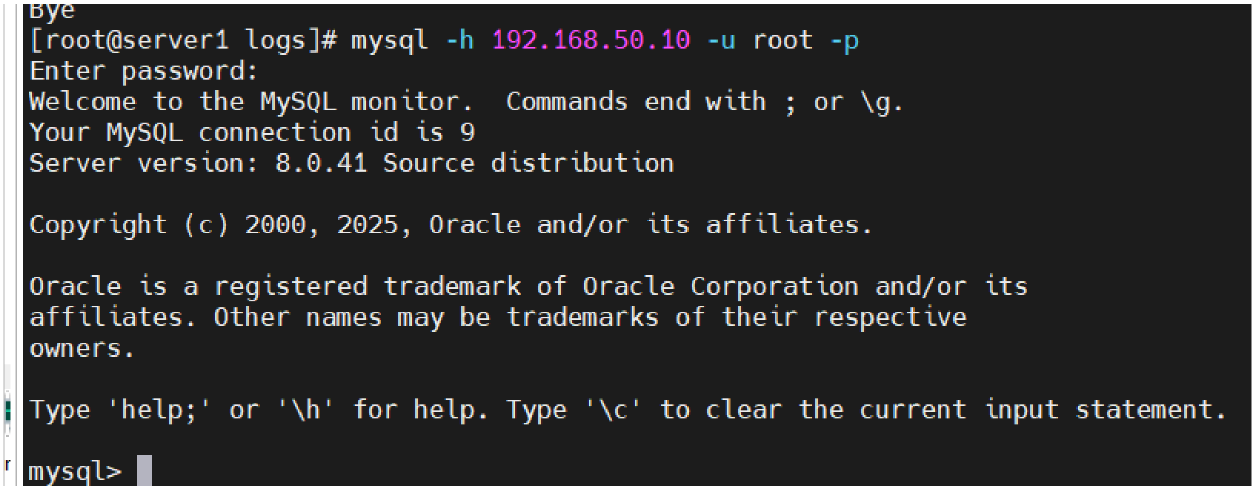
   EXIT;



```
mysql>
mysql> EXIT;
Bye
[root@server1 ~]#
[root@server1 ~]#
```

6. Verify MySQL connection works from command line:

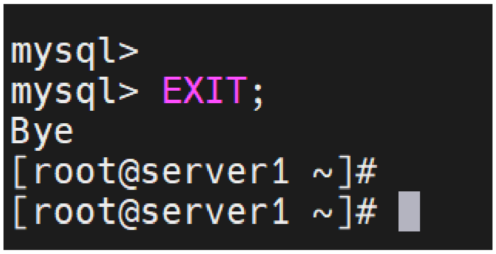
   mysql -h 192.168.50.10 -u root -p



```
Bye
[root@server1 logs]# mysql -h 192.168.50.10 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.41 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

7. Exit the MySQL shell:

   EXIT;

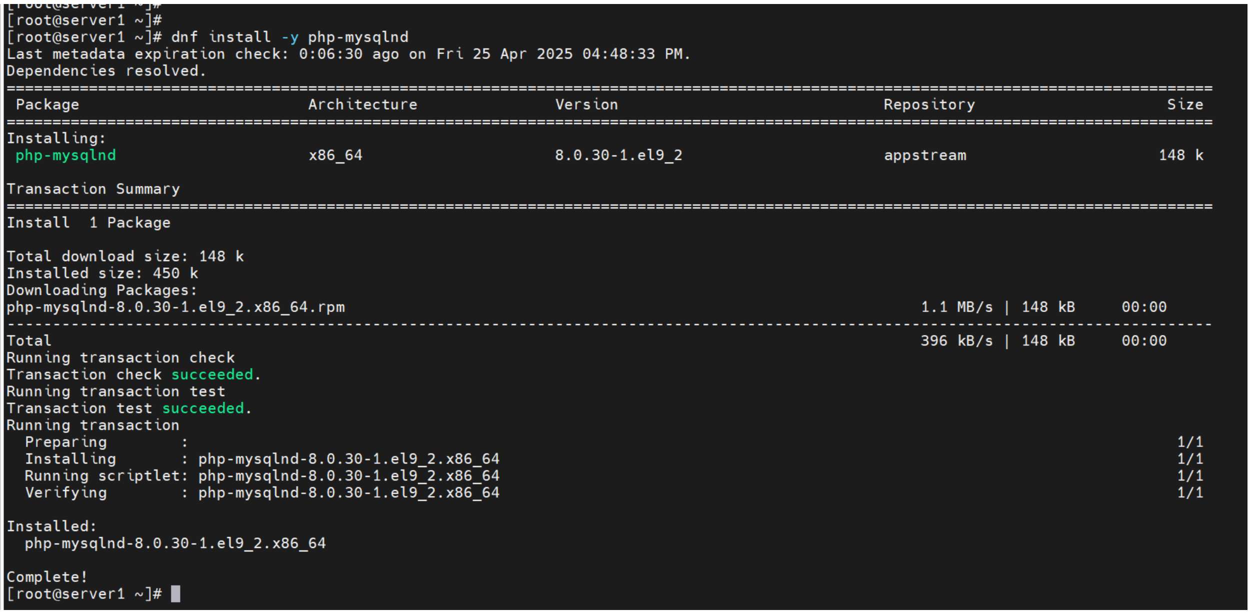


```
mysql>
mysql> EXIT;
Bye
[root@server1 ~]#
[root@server1 ~]# 
```

7.3.6 Installing PHP MySQL Connector

1. Install the PHP MySQL Native Driver:

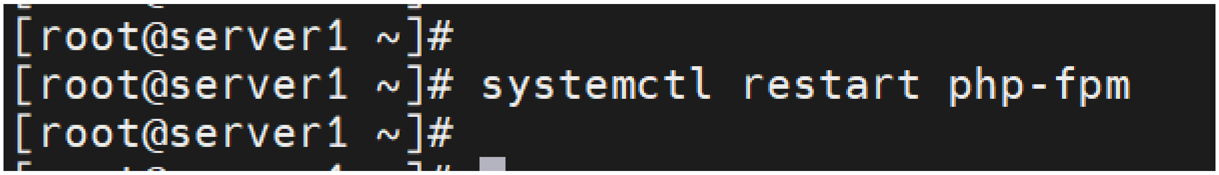
   dnf install -y php-mysqlnd



```
[root@server1 ~]#
[root@server1 ~]# dnf install -y php-mysqlnd
Last metadata expiration check: 0:06:30 ago on Fri 25 Apr 2025 04:48:33 PM.
Dependencies resolved.
=====================================================================================================================
 Package                         Architecture              Version                        Repository             Size
=====================================================================================================================
Installing:
 php-mysqlnd                     x86_64                    8.0.30-1.el9_2                 appstream             148 k

Transaction Summary
=====================================================================================================================
Install  1 Package

Total download size: 148 k
Installed size: 450 k
Downloading Packages:
php-mysqlnd-8.0.30-1.el9_2.x86_64.rpm                                            1.1 MB/s | 148 kB     00:00
---------------------------------------------------------------------------------------------------------------------
Total                                                                            396 kB/s | 148 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                        1/1
  Installing       : php-mysqlnd-8.0.30-1.el9_2.x86_64                                                      1/1
  Running scriptlet: php-mysqlnd-8.0.30-1.el9_2.x86_64                                                      1/1
  Verifying        : php-mysqlnd-8.0.30-1.el9_2.x86_64                                                      1/1

Installed:
  php-mysqlnd-8.0.30-1.el9_2.x86_64

Complete!
[root@server1 ~]#
```

2. Restart PHP-FPM to apply the configuration:

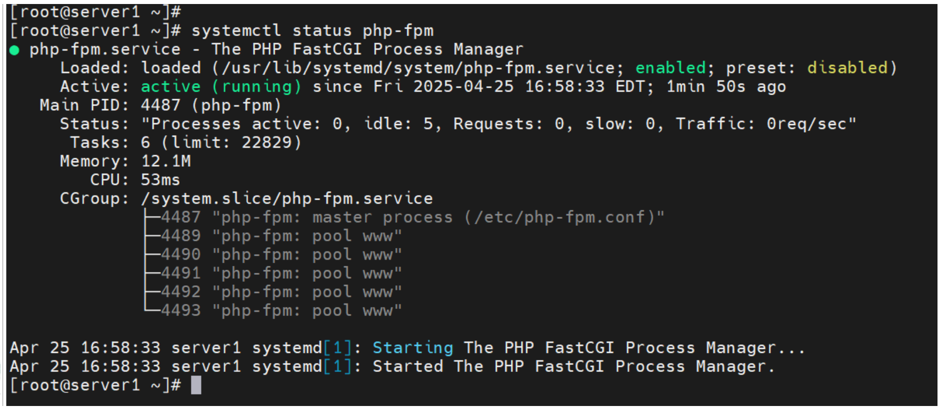
   systemctl restart php-fpm



```
[root@server1 ~]#
[root@server1 ~]# systemctl restart php-fpm
[root@server1 ~]#
```

3. Verify status

   systemctl status php-fpm



```
[root@server1 ~]#
[root@server1 ~]# systemctl status php-fpm
● php-fpm.service - The PHP FastCGI Process Manager
     Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-04-25 16:58:33 EDT; 1min 50s ago
   Main PID: 4487 (php-fpm)
     Status: "Processes active: 0, idle: 5, Requests: 0, slow: 0, Traffic: 0req/sec"
      Tasks: 6 (limit: 22829)
     Memory: 12.1M
        CPU: 53ms
     CGroup: /system.slice/php-fpm.service
             ├─4487 "php-fpm: master process (/etc/php-fpm.conf)"
             ├─4489 "php-fpm: pool www"
             ├─4490 "php-fpm: pool www"
             ├─4491 "php-fpm: pool www"
             ├─4492 "php-fpm: pool www"
             └─4493 "php-fpm: pool www"

Apr 25 16:58:33 server1 systemd[1]: Starting The PHP FastCGI Process Manager...
Apr 25 16:58:33 server1 systemd[1]: Started The PHP FastCGI Process Manager.
[root@server1 ~]#
```

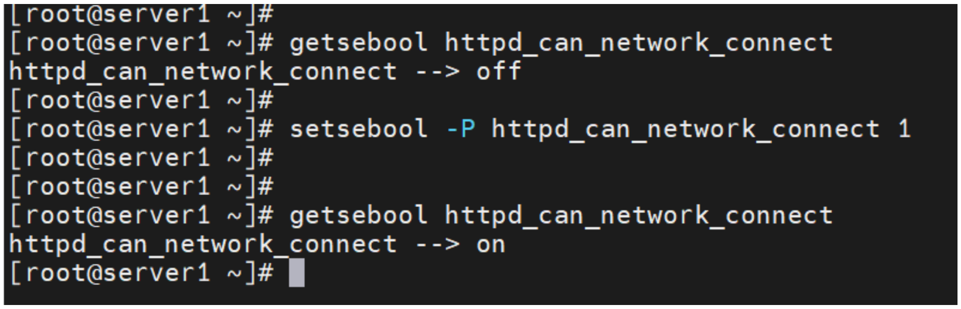
7.4 Selinux

1. Enable network connection with httpd in selinux

   getsebool httpd_can_network_connect

setsebool -P httpd_can_network_connect 1

getsebool httpd_can_network_connect

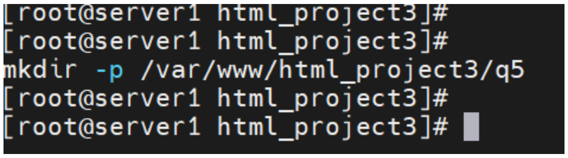


```
[root@server1 ~]#
[root@server1 ~]# getsebool httpd_can_network_connect
httpd_can_network_connect --> off
[root@server1 ~]#
[root@server1 ~]# setsebool -P httpd_can_network_connect 1
[root@server1 ~]#
[root@server1 ~]#
[root@server1 ~]# getsebool httpd_can_network_connect
httpd_can_network_connect --> on
[root@server1 ~]#
```

7.5 Create PHP script

1. Create the directory structure:

   mkdir -p /var/www/html_project3/q5



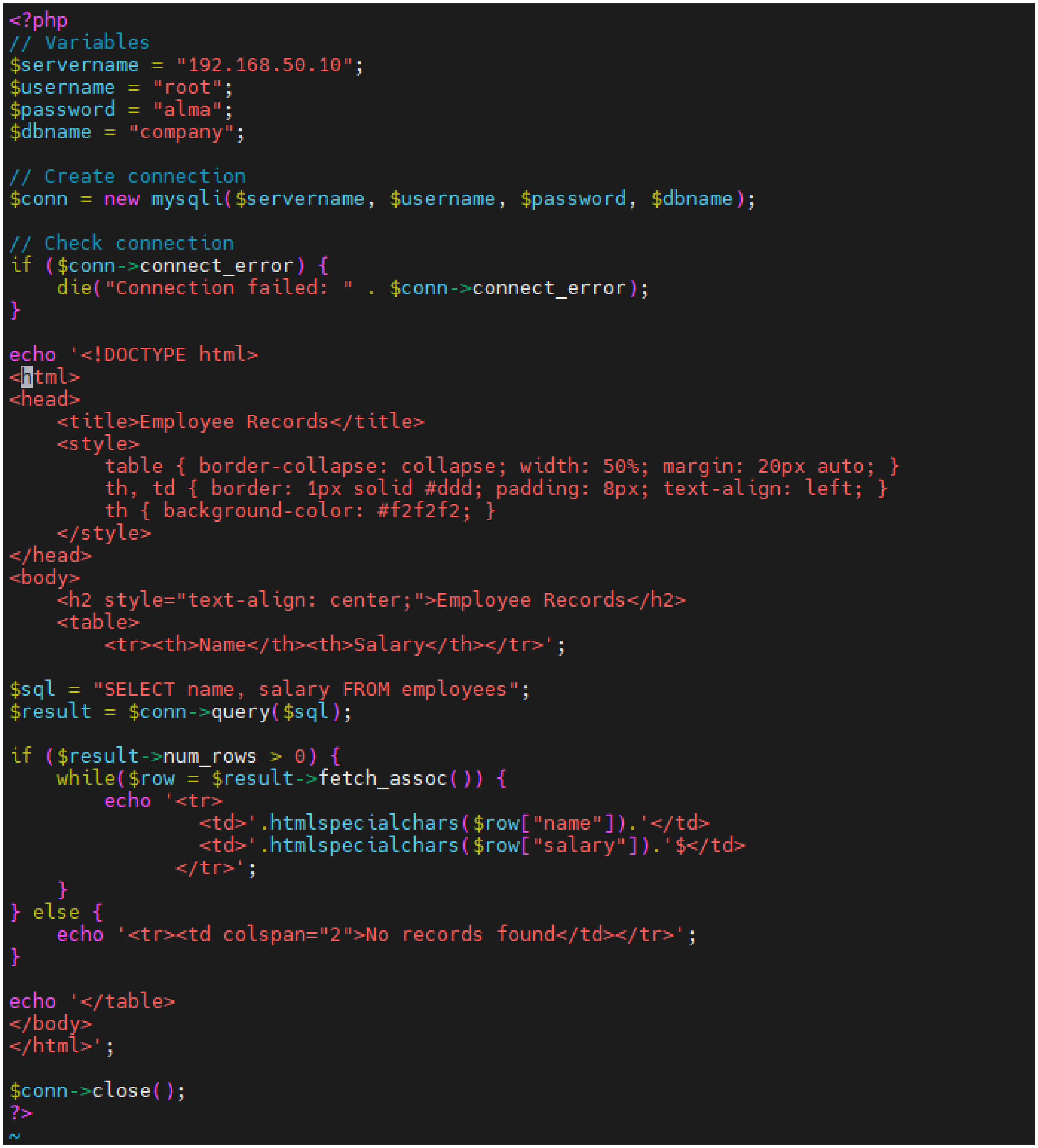
```
[root@server1 html_project3]#
[root@server1 html_project3]#
mkdir -p /var/www/html_project3/q5
[root@server1 html_project3]#
[root@server1 html_project3]#
```

2. Creating a PHP script to connect with MySQL database through httpd

   vim /var/www/html_project3/q5/task5.php

```
<?php
// Variables
$servername = "192.168.50.10";
$username = "root";
$password = "alma";
$dbname = "company";

// Create connection
$conn = new mysqli($servername, $username, $password,
$dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

echo '<!DOCTYPE html>
<html>
<head>
    <title>Employee Records</title>
    <style>
        table { border-collapse: collapse; width: 50%;
margin: 20px auto; }
        th, td { border: 1px solid #ddd; padding: 8px;
text-align: left; }
```

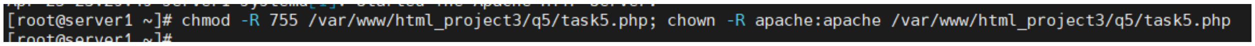
```
        th { background-color: #f2f2f2; }
    </style>
</head>
<body>
    <h2 style="text-align: center;">Employee Records</h2>
    <table>
        <tr><th>Name</th><th>Salary</th></tr>';

$sql = "SELECT name, salary FROM employees";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo '<tr>

<td>'.htmlspecialchars($row["name"]).'</td>

<td>'.htmlspecialchars($row["salary"]).'$</td>
              </tr>';
    }
} else {
    echo '<tr><td colspan="2">No records
found</td></tr>';
}

echo '</table>
</body>
</html>';

$conn->close();
?>
```



```
<?php
// Variables
$servername = "192.168.50.10";
$username = "root";
$password = "alma";
$dbname = "company";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

echo '<!DOCTYPE html>
<html>
<head>
    <title>Employee Records</title>
    <style>
        table { border-collapse: collapse; width: 50%; margin: 20px auto; }
        th, td { border: 1px solid #ddd; padding: 8px; text-align: left; }
        th { background-color: #f2f2f2; }
    </style>
</head>
<body>
    <h2 style="text-align: center;">Employee Records</h2>
    <table>
        <tr><th>Name</th><th>Salary</th></tr>';

$sql = "SELECT name, salary FROM employees";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo '<tr>
                <td>'.htmlspecialchars($row["name"]).'</td>
                <td>'.htmlspecialchars($row["salary"]).'$</td>
              </tr>';
    }
} else {
    echo '<tr><td colspan="2">No records found</td></tr>';
}

echo '</table>
</body>
</html>';

$conn->close();
?>
~
```

3. Make the script executable and change file permissions.

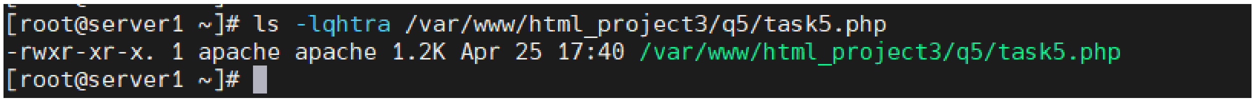
   chmod -R 755 /var/www/html_project3/q5/task5.php; chown -R apache:apache /var/www/html_project3/q5/task5.php



```
[root@server1 ~]# chmod -R 755 /var/www/html_project3/q5/task5.php; chown -R apache:apache /var/www/html_project3/q5/task5.php
[root@server1 ~]#
```

4. Verify the changes in file

   ls -lqhtra /var/www/html_project3/q5/task5.php



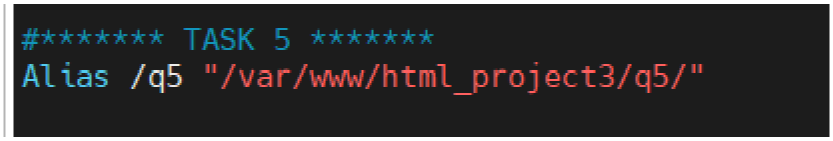
```
[root@server1 ~]# ls -lqhtra /var/www/html_project3/q5/task5.php
-rwxr-xr-x. 1 apache apache 1.2K Apr 25 17:40 /var/www/html_project3/q5/task5.php
[root@server1 ~]#
```

7.6 Apache configuration (httpd.conf)

1. Use an Alias for the PHP Directory, add this line to httpd.conf

```
Alias /q5 "/var/www/html_project3/q5/"
```

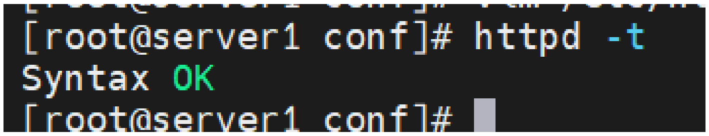
vim /etc/httpd/conf/httpd.conf



```
#******** TASK 5 ********
Alias /q5 "/var/www/html_project3/q5/"
```

2. Verify the syntax of configuration file after changes

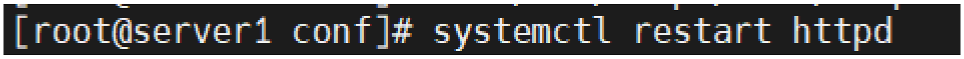
httpd -t



```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

3. Restart Apache: Apply the configuration changes

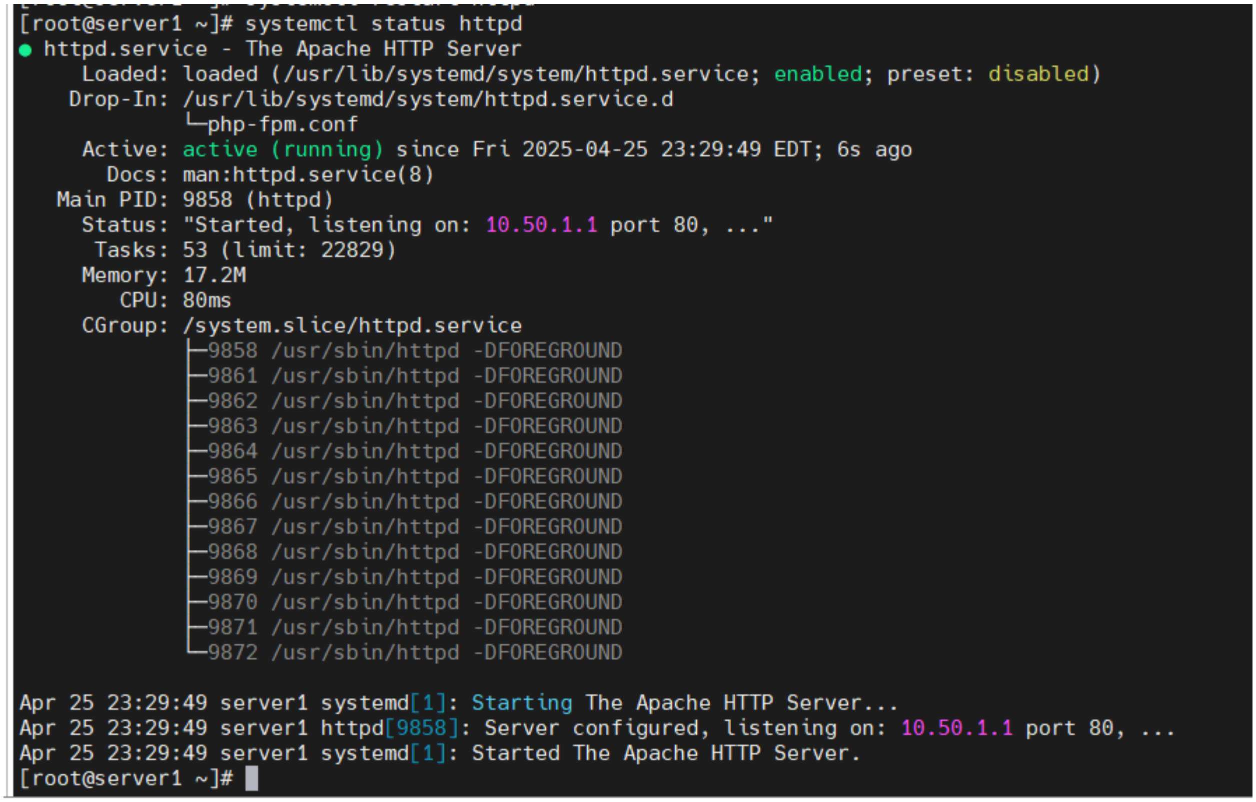
systemctl restart httpd



```
[root@server1 conf]# systemctl restart httpd
```

4. Verify httpd service status

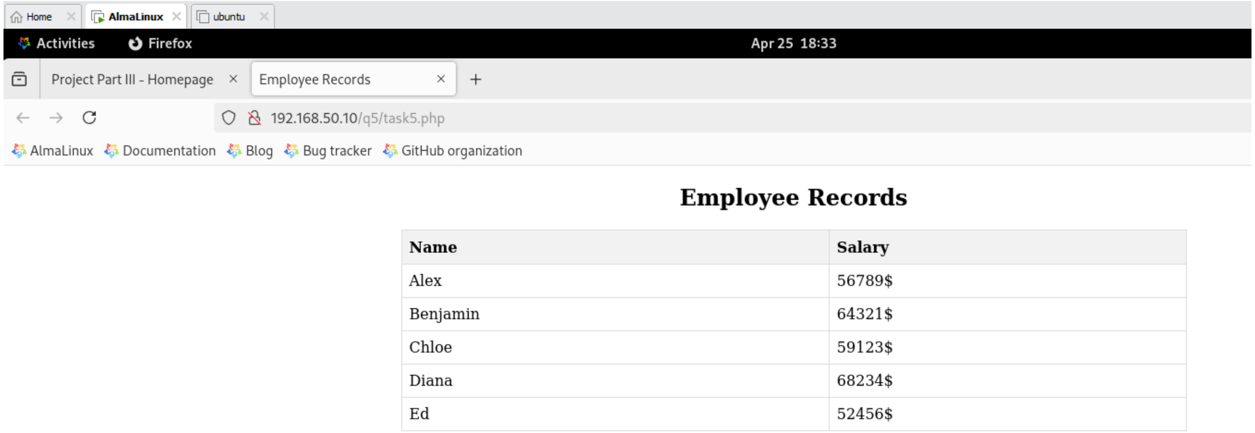
systemctl status httpd



```
[root@server1 ~]# systemctl restart httpd
[root@server1 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Fri 2025-04-25 23:29:49 EDT; 6s ago
       Docs: man:httpd.service(8)
   Main PID: 9858 (httpd)
     Status: "Started, listening on: 10.50.1.1 port 80, ..."
      Tasks: 53 (limit: 22829)
     Memory: 17.2M
        CPU: 80ms
     CGroup: /system.slice/httpd.service
             ├─9858 /usr/sbin/httpd -DFOREGROUND
             ├─9861 /usr/sbin/httpd -DFOREGROUND
             ├─9862 /usr/sbin/httpd -DFOREGROUND
             ├─9863 /usr/sbin/httpd -DFOREGROUND
             ├─9864 /usr/sbin/httpd -DFOREGROUND
             ├─9865 /usr/sbin/httpd -DFOREGROUND
             ├─9866 /usr/sbin/httpd -DFOREGROUND
             ├─9867 /usr/sbin/httpd -DFOREGROUND
             ├─9868 /usr/sbin/httpd -DFOREGROUND
             ├─9869 /usr/sbin/httpd -DFOREGROUND
             ├─9870 /usr/sbin/httpd -DFOREGROUND
             ├─9871 /usr/sbin/httpd -DFOREGROUND
             └─9872 /usr/sbin/httpd -DFOREGROUND

Apr 25 23:29:49 server1 systemd[1]: Starting The Apache HTTP Server...
Apr 25 23:29:49 server1 httpd[9858]: Server configured, listening on: 10.50.1.1 port 80, ...
Apr 25 23:29:49 server1 systemd[1]: Started The Apache HTTP Server.
[root@server1 ~]# 
```

7.7 Test

> # Task 5
>
> - To list employees mysql table



Employee Records

Name	Salary
Alex	56789$
Benjamin	64321$
Chloe	59123$
Diana	68234$
Ed	52456$

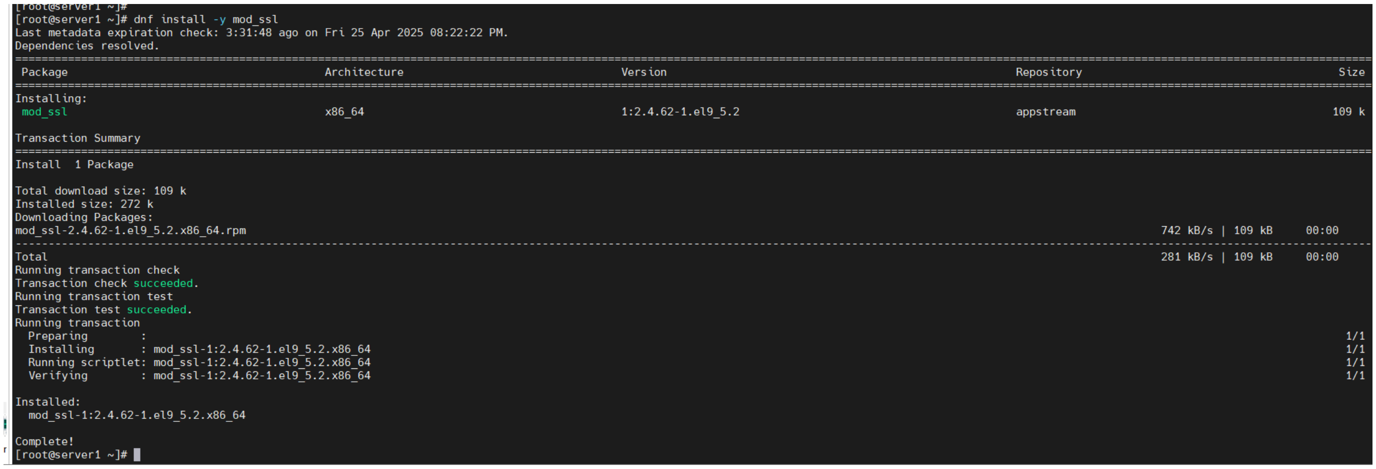
8 TASK 6 – SSL

8.1 Requirements

1. Configure your web server to use **SSL**.
2. Test access to your server using the **https** security protocol by navigating to: **<u>*https://serverX*</u>** *(where **X** is your assigned server number)*.

8.2 Install the SSL Module

dnf install -y mod_ssl



```
[root@server1 ~]#
[root@server1 ~]# dnf install -y mod_ssl
Last metadata expiration check: 3:31:48 ago on Fri 25 Apr 2025 08:22:22 PM.
Dependencies resolved.
==========================================================================================================================================
 Package                       Architecture                  Version                                   Repository                    Size
==========================================================================================================================================
Installing:
 mod_ssl                       x86_64                        1:2.4.62-1.el9_5.2                        appstream                    109 k

Transaction Summary
==========================================================================================================================================
Install  1 Package

Total download size: 109 k
Installed size: 272 k
Downloading Packages:
mod_ssl-2.4.62-1.el9_5.2.x86_64.rpm                                                                     742 kB/s | 109 kB     00:00
------------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                                   281 kB/s | 109 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                                1/1
  Installing       : mod_ssl-1:2.4.62-1.el9_5.2.x86_64                                                                              1/1
  Running scriptlet: mod_ssl-1:2.4.62-1.el9_5.2.x86_64                                                                              1/1
  Verifying        : mod_ssl-1:2.4.62-1.el9_5.2.x86_64                                                                              1/1

Installed:
  mod_ssl-1:2.4.62-1.el9_5.2.x86_64

Complete!
[root@server1 ~]#
```

8.3 Generate a Private Key and SSL Certificate Using OpenSSL

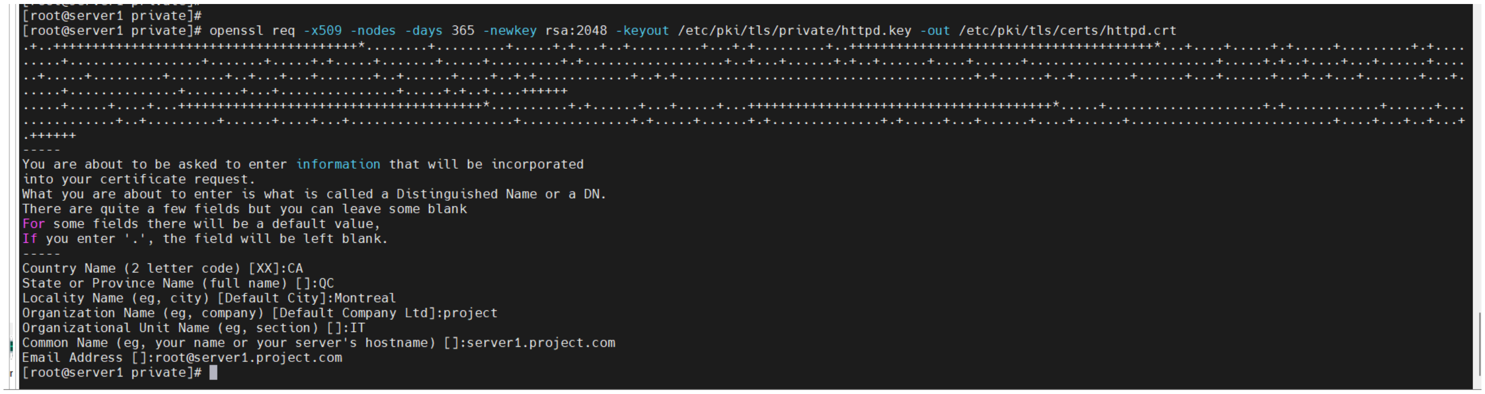
Self-Signed Certificate is for testing purposes.

1. Create a 2048-bit private key and a self-signed certificate valid for 1 year.

   openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/pki/tls/private/httpd.key -out /etc/pki/tls/certs/httpd.crt

   During this process, you'll be asked to provide details such as your country, state, organization, etc. answer accordingly:

```
Country Name (2 letter code) [XX]:CA
State or Province Name (full name) []:QC
Locality Name (eg, city) [Default City]:Montreal
Organization Name (eg, company) [Default Company Ltd]:project
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:server1.project.com
Email Address []:root@server1.project.com
```

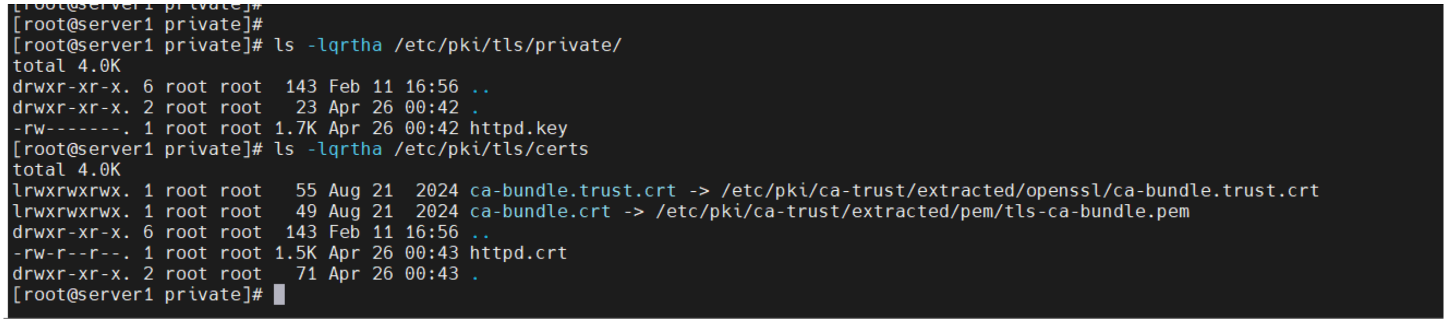


```
[root@server1 private]#
[root@server1 private]# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/pki/tls/private/httpd.key -out /etc/pki/tls/certs/httpd.crt
.+..++++++++++++++++++++++++++++++++++++++*.......+.........+.....+.+...+..+.........+...+.+.........+..++++++++++++++++++++++++++++++++++++++*...+....+.....+.+.....+.........+.+....
.....+................+.......+.....+.+.....+.......+.....+.........+.+.................+..+...+......+.+..+......+....+......+.......................+.....+.+..+....+...+......+....
..+.....+.........+.......+..+...+...+.......+..+......+....+..+.+............+..+.+...................................+.+......+..+.......+......+...+......+...+..+...+.......+...+.
.....+.............+.......+...+...............+.....+.+..+....++++++
.....+.....+....+...++++++++++++++++++++++++++++++++++++++*..........+.+......+...+.....+...++++++++++++++++++++++++++++++++++++++*.....+.....................+.+............+......+...
...........+..+.........+......+....+...+.....................+.............+.+.....+......+.+..............+.+.....+...+......+....+......+..........................+....+...+..+...+
.++++++
-----
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]:CA
State or Province Name (full name) []:QC
Locality Name (eg, city) [Default City]:Montreal
Organization Name (eg, company) [Default Company Ltd]:project
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:server1.project.com
Email Address []:root@server1.project.com
[root@server1 private]#
```

2. Verify files are created

   ls -lqrtha /etc/pki/tls/private/

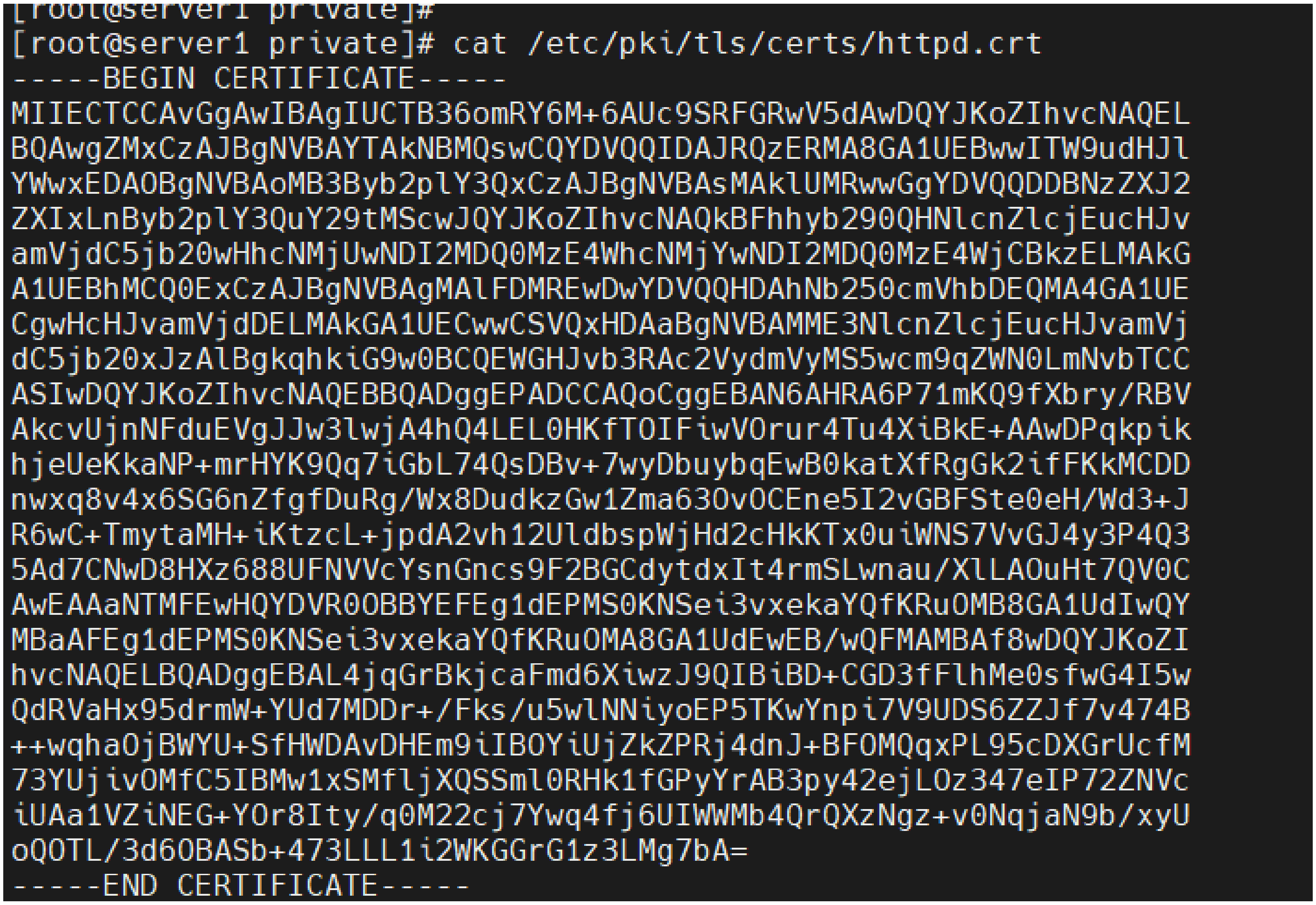
   ls -lqrtha /etc/pki/tls/certs



```
[root@server1 private]#
[root@server1 private]# ls -lqrtha /etc/pki/tls/private/
total 4.0K
drwxr-xr-x. 6 root root  143 Feb 11 16:56 ..
drwxr-xr-x. 2 root root   23 Apr 26 00:42 .
-rw-------. 1 root root 1.7K Apr 26 00:42 httpd.key
[root@server1 private]# ls -lqrtha /etc/pki/tls/certs
total 4.0K
lrwxrwxrwx. 1 root root   55 Aug 21  2024 ca-bundle.trust.crt -> /etc/pki/ca-trust/extracted/openssl/ca-bundle.trust.crt
lrwxrwxrwx. 1 root root   49 Aug 21  2024 ca-bundle.crt -> /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
drwxr-xr-x. 6 root root  143 Feb 11 16:56 ..
-rw-r--r--. 1 root root 1.5K Apr 26 00:43 httpd.crt
drwxr-xr-x. 2 root root   71 Apr 26 00:43 .
[root@server1 private]#
```

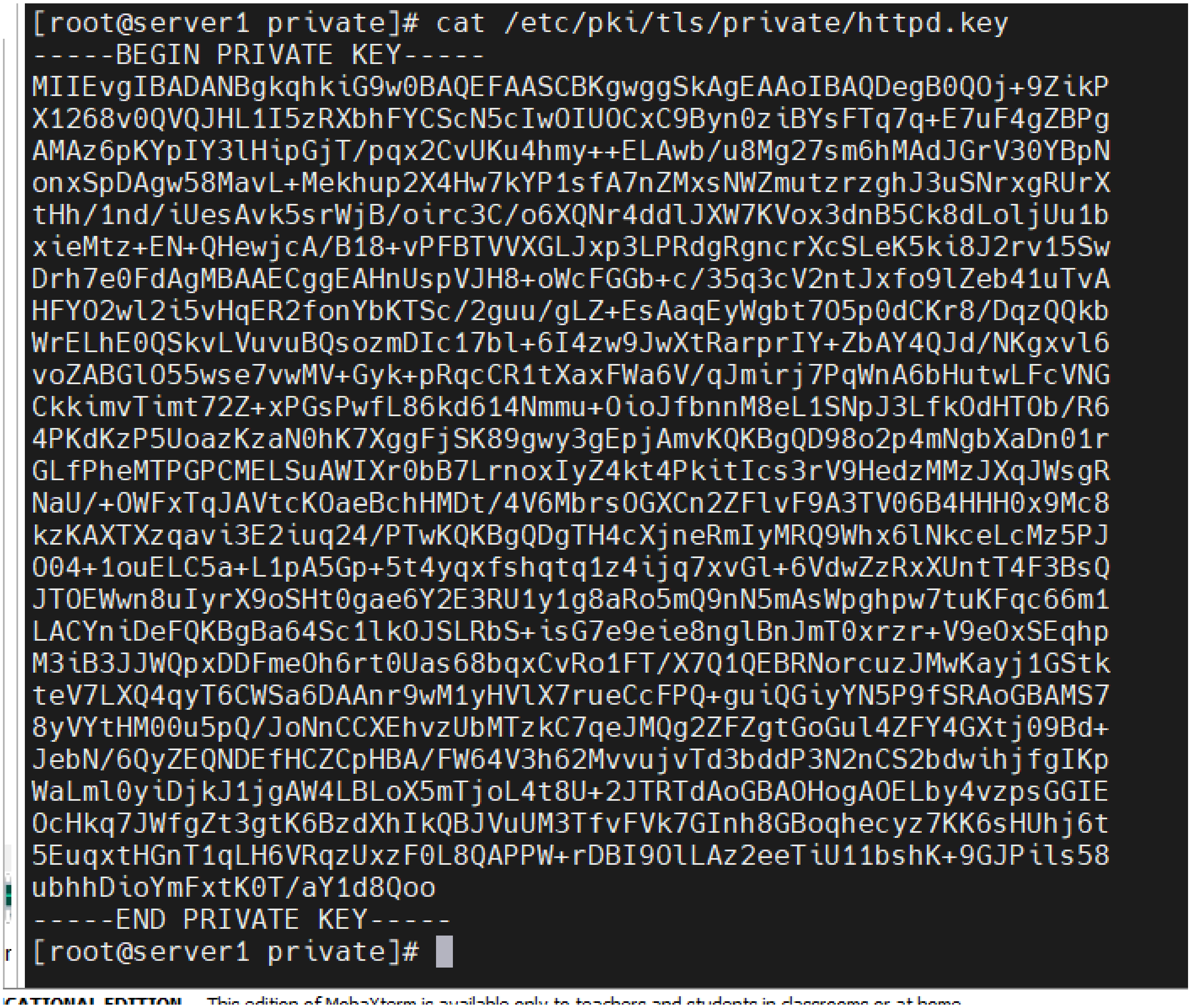
3. List contents

   cat /etc/pki/tls/certs/httpd.crt



```
[root@server1 private]#
[root@server1 private]# cat /etc/pki/tls/certs/httpd.crt
-----BEGIN CERTIFICATE-----
MIIECTCCAvGgAwIBAgIUCTB36omRY6M+6AUc9SRFGRwV5dAwDQYJKoZIhvcNAQEL
BQAwgZMxCzAJBgNVBAYTAkNBMQswCQYDVQQIDAJRQzERMA8GA1UEBwwITW9udHJl
YWwxEDAOBgNVBAoMB3Byb2plY3QxCzAJBgNVBAsMAklUMRwwGgYDVQQDDBNzZXJ2
ZXIxLnByb2plY3QuY29tMScwJQYJKoZIhvcNAQkBFhhyb290QHNlcnZlcjEucHJv
amVjdC5jb20wHhcNMjUwNDI2MDQ0MzE4WhcNMjYwNDI2MDQ0MzE4WjCBkzELMAkG
A1UEBhMCQ0ExCzAJBgNVBAgMAlFDMREwDwYDVQQHDAhNb250cmVhbDEQMA4GA1UE
CgwHcHJvamVjdDELMAkGA1UECwwCSVQxHDAaBgNVBAMME3NlcnZlcjEucHJvamVj
dC5jb20xJzAlBgkqhkiG9w0BCQEWGHJvb3RAc2VydmVyMS5wcm9qZWN0LmNvbTCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAN6AHRA6P71mKQ9fXbry/RBV
AkcvUjnNFduEVgJJw3lwjA4hQ4LEL0HKfTOIFiwVOrur4Tu4XiBkE+AAwDPqkpik
hjeUeKkaNP+mrHYK9Qq7iGbL74QsDBv+7wyDbuybqEwB0katXfRgGk2ifFKkMCDD
nwxq8v4x6SG6nZfgfDuRg/Wx8DudkzGw1Zma63OvOCEne5I2vGBFSte0eH/Wd3+J
R6wC+TmytaMH+iKtzcL+jpdA2vh12UldbspWjHd2cHkKTx0uiWNS7VvGJ4y3P4Q3
5Ad7CNwD8HXz688UFNVVcYsnGncs9F2BGCdytdxIt4rmSLwnau/XlLAOuHt7QV0C
AwEAAaNTMFEwHQYDVR0OBBYEFEg1dEPMS0KNSei3vxekaYQfKRuOMB8GA1UdIwQY
MBaAFEg1dEPMS0KNSei3vxekaYQfKRuOMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZI
hvcNAQELBQADggEBAL4jqGrBkjcaFmd6XiwzJ9QIBiBD+CGD3fFlhMe0sfwG4I5w
QdRVaHx95drmW+YUd7MDDr+/Fks/u5wlNNiyoEP5TKwYnpi7V9UDS6ZZJf7v474B
++wqhaOjBWYU+SfHWDAvDHEm9iIBOYiUjZkZPRj4dnJ+BFOMQqxPL95cDXGrUcfM
73YUjivOMfC5IBMw1xSMfljXQSSml0RHk1fGPyYrAB3py42ejLOz347eIP72ZNVc
iUAa1VZiNEG+YOr8Ity/q0M22cj7Ywq4fj6UIWWMb4QrQXzNgz+v0NqjaN9b/xyU
oQOTL/3d6OBASb+473LLL1i2WKGGrG1z3LMg7bA=
-----END CERTIFICATE-----
```

cat /etc/pki/tls/private/httpd.key



```
[root@server1 private]# cat /etc/pki/tls/private/httpd.key
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDegB0QOj+9ZikP
X1268v0QVQJHL1I5zRXbhFYCScN5cIwOIUOCxC9Byn0ziBYsFTq7q+E7uF4gZBPg
AMAz6pKYpIY3lHipGjT/pqx2CvUKu4hmy++ELAwb/u8Mg27sm6hMAdJGrV30YBpN
onxSpDAgw58MavL+Mekhup2X4Hw7kYP1sfA7nZMxsNWZmutzrzghJ3uSNrxgRUrX
tHh/1nd/iUesAvk5srWjB/oirc3C/o6XQNr4ddlJXW7KVox3dnB5Ck8dLoljUu1b
xieMtz+EN+QHewjcA/B18+vPFBTVVXGLJxp3LPRdgRgncrXcSLeK5ki8J2rv15Sw
Drh7e0FdAgMBAAECggEAHnUspVJH8+oWcFGGb+c/35q3cV2ntJxfo9lZeb41uTvA
HFYO2wl2i5vHqER2fonYbKTSc/2guu/gLZ+EsAaqEyWgbt7O5p0dCKr8/DqzQQkb
WrELhE0QSkvLVuvuBQsozmDIc17bl+6I4zw9JwXtRarprIY+ZbAY4QJd/NKgxvl6
voZABGlO55wse7vwMV+Gyk+pRqcCR1tXaxFWa6V/qJmirj7PqWnA6bHutwLFcVNG
CkkimvTimt72Z+xPGsPwfL86kd614Nmmu+OioJfbnnM8eL1SNpJ3LfkOdHTOb/R6
4PKdKzP5UoazKzaN0hK7XggFjSK89gwy3gEpjAmvKQKBgQD98o2p4mNgbXaDn01r
GLfPheMTPGPCMELSuAWIXr0bB7LrnoxIyZ4kt4PkitIcs3rV9HedzMMzJXqJWsgR
NaU/+OWFxTqJAVtcKOaeBchHMDt/4V6MbrsOGXCn2ZFlvF9A3TV06B4HHH0x9Mc8
kzKAXTXzqavi3E2iuq24/PTwKQKBgQDgTH4cXjneRmIyMRQ9Whx6lNkceLcMz5PJ
O04+1ouELC5a+L1pA5Gp+5t4yqxfshqtq1z4ijq7xvGl+6VdwZzRxXUntT4F3BsQ
JTOEWwn8uIyrX9oSHt0gae6Y2E3RU1y1g8aRo5mQ9nN5mAsWpghpw7tuKFqc66m1
LACYniDeFQKBgBa64Sc1lkOJSLRbS+isG7e9eie8nglBnJmT0xrzr+V9eOxSEqhp
M3iB3JJWQpxDDFmeOh6rt0Uas68bqxCvRo1FT/X7Q1QEBRNorcuzJMwKayj1GStk
teV7LXQ4qyT6CWSa6DAAnr9wM1yHVlX7rueCcFPQ+guiQGiyYN5P9fSRAoGBAMS7
8yVYtHM00u5pQ/JoNnCCXEhvzUbMTzkC7qeJMQg2ZFZgtGoGul4ZFY4GXtj09Bd+
JebN/6QyZEQNDEfHCZCpHBA/FW64V3h62MvvujvTd3bddP3N2nCS2bdwihjfgIKp
WaLml0yiDjkJ1jgAW4LBLoX5mTjoL4t8U+2JTRTdAoGBAOHogAOELby4vzpsGGIE
OcHkq7JWfgZt3gtK6BzdXhIkQBJVuUM3TfvFVk7GInh8GBoqhecyz7KK6sHUhj6t
5EuqxtHGnT1qLH6VRqzUxzF0L8QAPPW+rDBI9OlLAz2eeTiU11bshK+9GJPils58
ubhhDioYmFxtK0T/aY1d8Qoo
-----END PRIVATE KEY-----
[root@server1 private]# 
```

CATIONAL EDITION   This edition of MobaXterm is available only to teachers and students in classrooms or at home

8.4 Edit the SSL Configuration File

1. Verify the contents of SSL configuration file.

   cat -n /etc/httpd/conf.d/ssl.conf

```
[root@server1 private]# cat -n /etc/httpd/conf.d/ssl.conf
     1  #
     2  # When we also provide SSL we have to listen to the
     3  # standard HTTPS port in addition.
     4  #
     5  Listen 443 https
     6
     7  ##
     8  ##  SSL Global Context
     9  ##
    10  ##  All SSL configuration in this context applies both to
    11  ##  the main server and all SSL-enabled virtual hosts.
    12  ##
```

```

#   Pass Phrase Dialog:
#   Configure the pass phrase gathering process.
#   The filtering dialog program (`builtin' is a internal
#   terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog exec:/usr/libexec/httpd-ssl-pass-dialog

#   Inter-Process Session Cache:
#   Configure the SSL Session Cache: First the mechanism
#   to use and second the expiring timeout (in seconds).
SSLSessionCache         shmcb:/run/httpd/sslcache(512000)
SSLSessionCacheTimeout  300

#
# Use "SSLCryptoDevice" to enable any supported hardware
# accelerators. Use "openssl engine -v" to list supported
# engine names.  NOTE: If you enable an accelerator and the
# server does not start, consult the error logs and ensure
# your accelerator is functioning properly.
#
SSLCryptoDevice builtin
#SSLCryptoDevice ubsec

##
## SSL Virtual Host Context
##

<VirtualHost _default_:443>

# General setup for the virtual host, inherited from global configuration
#DocumentRoot "/var/www/html"
#ServerName www.example.com:443

# Use separate log files for the SSL virtual host; note that LogLevel
# is not inherited from httpd.conf.
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
LogLevel warn

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   List the protocol versions which clients are allowed to connect with.
#   The OpenSSL system profile is used by default.  See
#   update-crypto-policies(8) for more details.
#SSLProtocol all -SSLv3
#SSLProxyProtocol all -SSLv3

#   User agents such as web browsers are not configured for the user's
#   own preference of either security or performance, therefore this
#   must be the prerogative of the web server administrator who manages
#   cpu load versus confidentiality, so enforce the server's cipher order.
SSLHonorCipherOrder on

#   SSL Cipher Suite:
#   List the ciphers that the client is permitted to negotiate.
#   See the mod_ssl documentation for a complete list.
#   The OpenSSL system profile is configured by default.  See
#   update-crypto-policies(8) for more details.
SSLCipherSuite PROFILE=SYSTEM
SSLProxyCipherSuite PROFILE=SYSTEM

```

```
#   Point SSLCertificateFile at a PEM encoded certificate.  If
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that restarting httpd will prompt again.  Keep
#   in mind that if you have both an RSA and a DSA certificate you
#   can configure both in parallel (to also allow the use of DSA
#   ciphers, etc.)
#   Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
#   require an ECC certificate which can also be configured in
#   parallel.
SSLCertificateFile /etc/pki/tls/certs/localhost.crt

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
#   ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convenience.
#SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#SSLCACertificateFile /etc/pki/tls/certs/ca-bundle.crt

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   Access Control:
#   With SSLRequire you can do per-directory access control based
#   on arbitrary complex boolean expressions containing server
#   variable checks and other lookup directives.  The syntax is a
#   mixture between C and Perl.  See the mod_ssl documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#            and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#            and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#            and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#            and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20       ) \
#           or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used for access control.  The
#     user name is the `one line' version of the client's X.509 certificate.
#     Note that no password is obtained from the user. Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment variables: SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#     server (always existing) and the client (only existing when client
#     authentication is used). This can be used to import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_*' environment variables.
#     Per default this exportation is switched off for performance reasons,
#     because the extraction step is an expensive operation and is usually
#     useless for serving static content. So one usually enables the
#     exportation for CGI and SSI requests only.
#   o StrictRequire:
#     This denies access when "SSLRequireSSL" or "SSLRequire" applied even
#     under a "Satisfy any" situation, i.e. when it applies access is denied
#     and no other module can change it.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when SSL
#     directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory "/var/www/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is sent or allowed to be received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#     This forces an accurate shutdown when the connection is closed, i.e. a
#     SSL close notify alert is sent and mod_ssl waits for the close notify
#     alert of the client. This is 100% SSL/TLS standard compliant, but in
#     practice often causes hanging connections with brain-dead browsers. Use
#     this only for browsers where you know that their SSL implementation
#     works correctly.
#   Notice: Most problems of broken clients are also related to the HTTP
#   keep-alive facility, so you usually additionally want to disable
#   keep-alive for those clients, too. Use variable "nokeepalive" for this.
#   Similarly, one has to force some clients to use HTTP/1.0 to workaround
#   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
#   "force-response-1.0" for this.
BrowserMatch "MSIE [2-5]" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0

#   Per-Server Logging:
#   The home of a custom SSL log file. Use this when you want a
#   compact non-error SSL logfile on a virtual host basis.
CustomLog logs/ssl_request_log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

```

```
   202  </VirtualHost>
   203
[root@server1 private]#
```

2. Modify the Apache SSL configuration file /etc/httpd/conf.d/ssl.conf to specify the path to your SSL certificate and private key:

   vim /etc/httpd/conf.d/ssl.conf

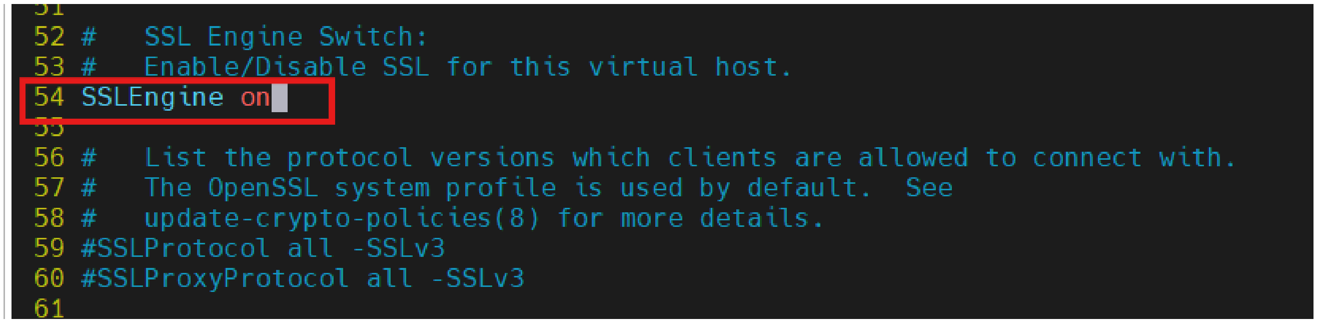
   As per the line numbers in the file ssl.conf, verify this line is available:

   54 SSLEngine on

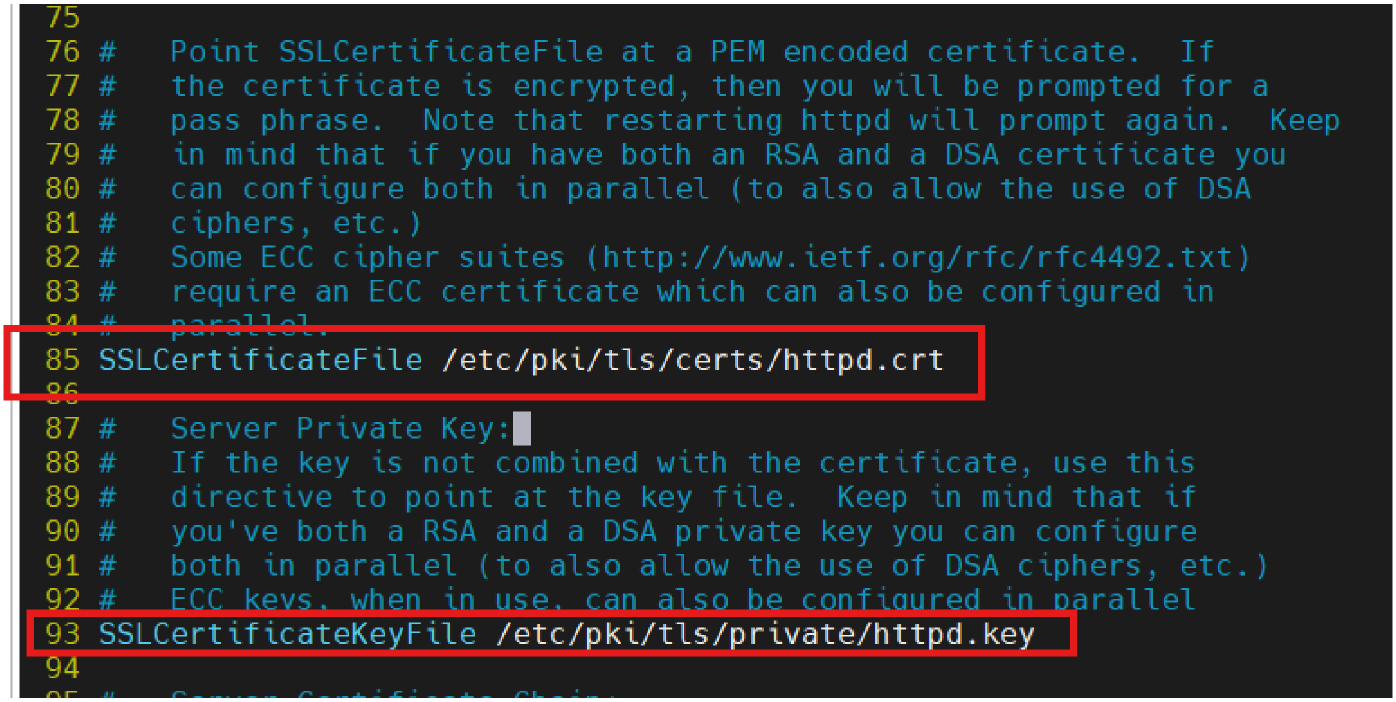
   As per the line numbers in the file ssl.conf , change the file name for the certificate created above:

   85 SSLCertificateFile /etc/pki/tls/certs/**httpd.crt**

   93 SSLCertificateKeyFile /etc/pki/tls/private/**httpd.key**



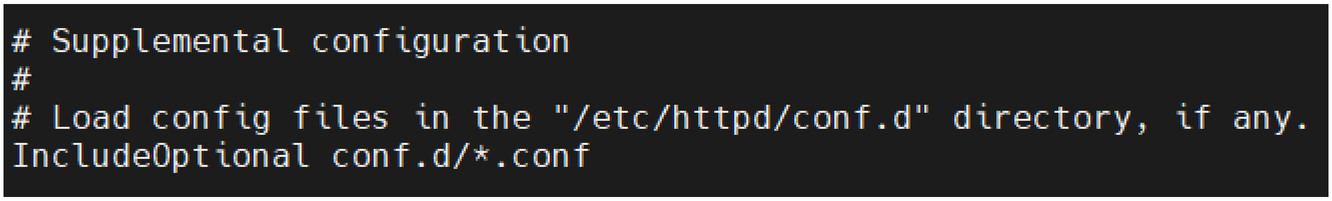
```
51
52 #   SSL Engine Switch:
53 #   Enable/Disable SSL for this virtual host.
54 SSLEngine on
55
56 #   List the protocol versions which clients are allowed to connect with.
57 #   The OpenSSL system profile is used by default.  See
58 #   update-crypto-policies(8) for more details.
59 #SSLProtocol all -SSLv3
60 #SSLProxyProtocol all -SSLv3
61
```



```
75
76 #   Point SSLCertificateFile at a PEM encoded certificate.  If
77 #   the certificate is encrypted, then you will be prompted for a
78 #   pass phrase.  Note that restarting httpd will prompt again.  Keep
79 #   in mind that if you have both an RSA and a DSA certificate you
80 #   can configure both in parallel (to also allow the use of DSA
81 #   ciphers, etc.)
82 #   Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
83 #   require an ECC certificate which can also be configured in
84 #   parallel.
85 SSLCertificateFile /etc/pki/tls/certs/httpd.crt
86
87 #   Server Private Key:
88 #   If the key is not combined with the certificate, use this
89 #   directive to point at the key file.  Keep in mind that if
90 #   you've both a RSA and a DSA private key you can configure
91 #   both in parallel (to also allow the use of DSA ciphers, etc.)
92 #   ECC keys, when in use, can also be configured in parallel
93 SSLCertificateKeyFile /etc/pki/tls/private/httpd.key
94
95 #   Server Certificate Chain:
```

3. Verify the fiel httpd and make sure this line is completed and/or not commented

IncludeOptional conf.d/*.conf

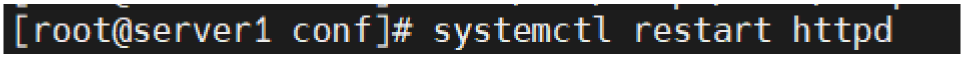


```
# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
```

8.5 Restart htpd

1. Restart Apache: Apply the configuration changes

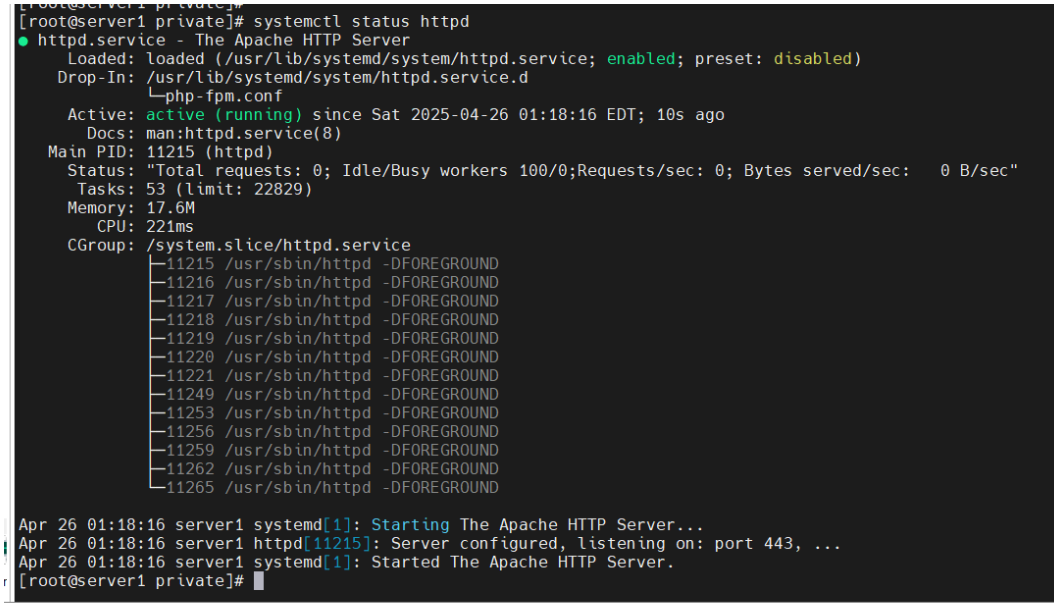
   systemctl restart httpd



```
[root@server1 conf]# systemctl restart httpd
```

2. Verify httpd service status

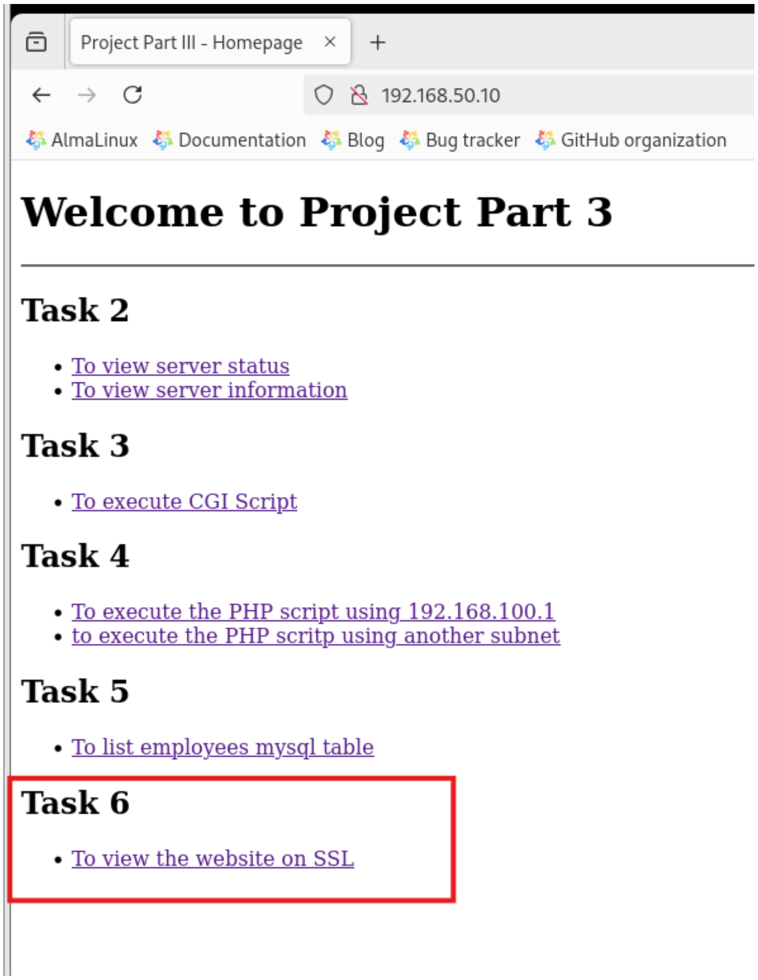
   systemctl status httpd



```
[root@server1 private]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Sat 2025-04-26 01:18:16 EDT; 10s ago
       Docs: man:httpd.service(8)
   Main PID: 11215 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 53 (limit: 22829)
     Memory: 17.6M
        CPU: 221ms
     CGroup: /system.slice/httpd.service
             ├─11215 /usr/sbin/httpd -DFOREGROUND
             ├─11216 /usr/sbin/httpd -DFOREGROUND
             ├─11217 /usr/sbin/httpd -DFOREGROUND
             ├─11218 /usr/sbin/httpd -DFOREGROUND
             ├─11219 /usr/sbin/httpd -DFOREGROUND
             ├─11220 /usr/sbin/httpd -DFOREGROUND
             ├─11221 /usr/sbin/httpd -DFOREGROUND
             ├─11249 /usr/sbin/httpd -DFOREGROUND
             ├─11253 /usr/sbin/httpd -DFOREGROUND
             ├─11256 /usr/sbin/httpd -DFOREGROUND
             ├─11259 /usr/sbin/httpd -DFOREGROUND
             ├─11262 /usr/sbin/httpd -DFOREGROUND
             └─11265 /usr/sbin/httpd -DFOREGROUND

Apr 26 01:18:16 server1 systemd[1]: Starting The Apache HTTP Server...
Apr 26 01:18:16 server1 httpd[11215]: Server configured, listening on: port 443, ...
Apr 26 01:18:16 server1 systemd[1]: Started The Apache HTTP Server.
[root@server1 private]# █
```

8.6 Test

1. From main menu select



Welcome to Project Part 3

Task 2

- To view server status
- To view server information

Task 3

- To execute CGI Script

Task 4

- To execute the PHP script using 192.168.100.1
- to execute the PHP scritp using another subnet

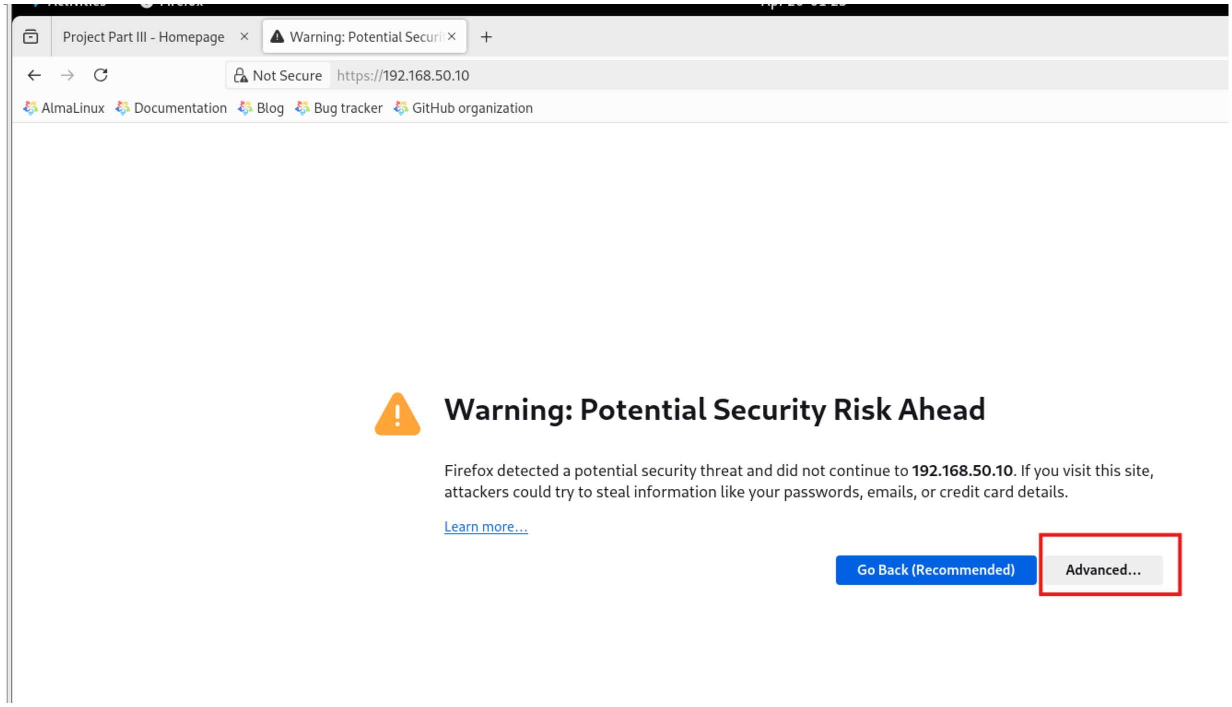
Task 5

- To list employees mysql table

Task 6

- To view the website on SSL

2. A page opens in a new tab , select Advanced...

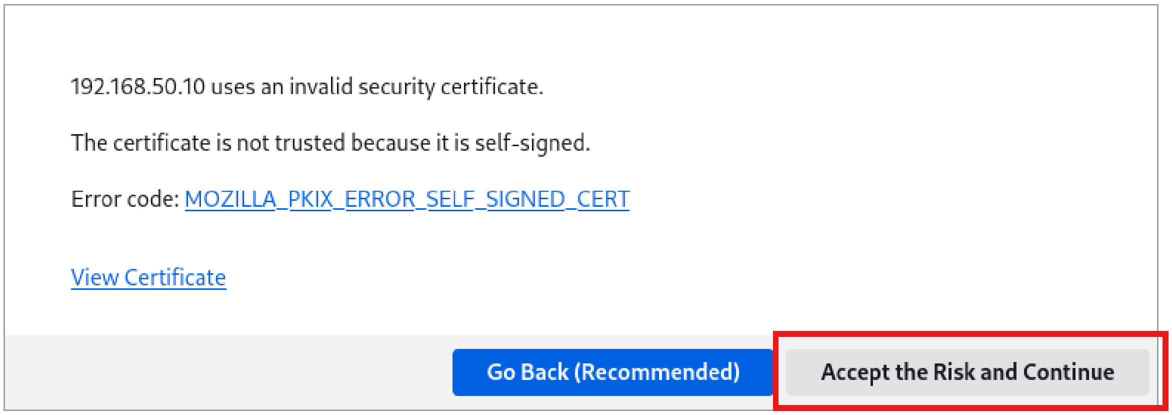


Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **192.168.50.10**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

Learn more…

3. Another window opens below, select “Accept the Risk and Continue”

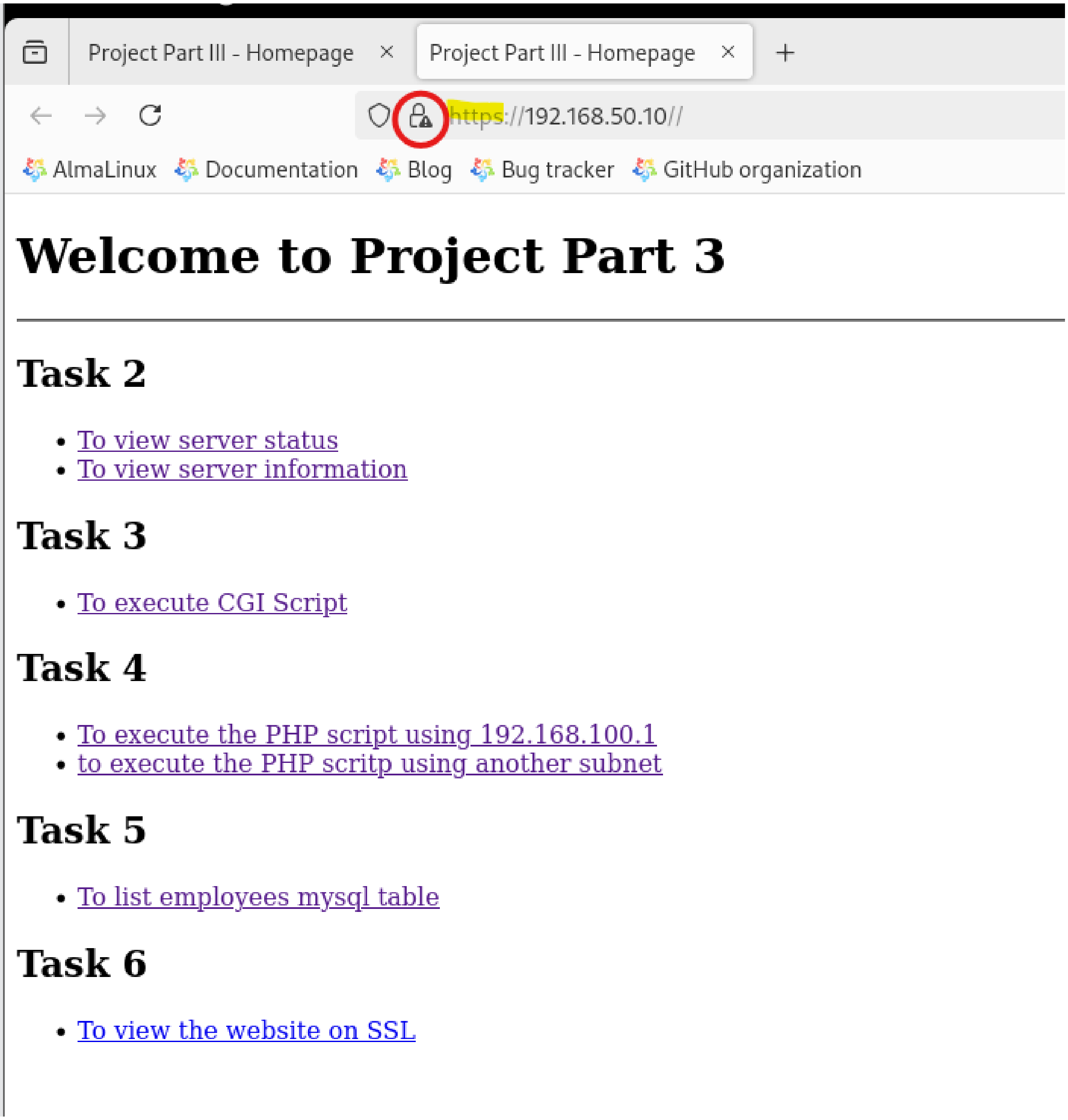


192.168.50.10 uses an invalid security certificate.

The certificate is not trusted because it is self-signed.

Error code: MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT

4. A new page with main Meni appears, note an open lock and web page https://192.168.50.10



Welcome to Project Part 3

Task 2

- To view server status
- To view server information

Task 3

- To execute CGI Script

Task 4

- To execute the PHP script using 192.168.100.1
- to execute the PHP scritp using another subnet

Task 5

- To list employees mysql table

Task 6

- To view the website on SSL

9 Configuration files for Project 3

9.1 master_project3.html

cat /var/www/html_project3/master_project3.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Project Part III - Homepage</title>
</head>
```

```
<body>
  <h1>Welcome to Project Part 3</h1>
  <hr>

  <!-- Task 2 -->
  <h2>Task 2</h2>
  <ul>
    <li><a href="http://192.168.50.10/server-status" target="_blank">To view server status</a></li>
    <li><a href="http://192.168.50.10/server-info" target="_blank">To view server information</a></li>
  </ul>

  <!-- Task 3 -->
  <h2>Task 3</h2>
  <ul>
    <li><a href="http://192.168.50.10/cgi-bin/task3.pl" target="_blank">To execute CGI Script</a></li>
  </ul>

  <!-- Task 4 -->
  <h2>Task 4</h2>
  <ul>
    <li><a href="http://192.168.100.1/q4/task4.php" target="_blank">To execute the PHP script using 192.168.100.1</a></li>
    <li><a href="http://192.168.50.10/q4/task4.php" target="_blank">to execute the PHP scritp using another subnet</a></li>
  </ul>

  <!-- Task 5 -->
  <h2>Task 5</h2>
  <ul>
    <li><a href="http://192.168.50.10/q5/task5.php" target="_blank">To list employees mysql table</a></li>
  </ul>

  <!-- Task 5 -->
  <h2>Task 6</h2>
  <ul>
    <li><a href="https:192.168.50.10//" target="_blank">To view the website on SSL</a></li>
  </ul>

</body>
</html>
```

9.2 httpd.conf

cat /etc/httpd/conf/httpd.conf

```
ServerRoot "/etc/httpd"
Listen 192.168.50.10:80
Listen 192.168.100.1:80
Listen 10.35.16.1:80
Listen 10.35.16.1:8000
Listen 10.35.17.1:80
Listen 10.35.17.1:8000
Listen 10.35.16.1:8080
Listen 10.35.16.1:8081
Listen 10.35.16.1:8082
Listen 10.35.16.1:8083
Listen 10.50.1.1:80

Include conf.modules.d/*.conf

User apache
Group apache

#ServerAdmin root@localhost
ServerName 192.168.50.10

<Directory />
   AllowOverride none
   Require all denied
</Directory>

DocumentRoot "/var/www/html_project3"

<Directory "/var/www">
   AllowOverride None
   Require all granted
</Directory>


<Directory "/var/www/html_project3">
   Options Indexes FollowSymLinks
   AllowOverride All
   Require all granted
</Directory>

#### PART 3 #######
#**** TASK 1 *********
<IfModule mpm_prefork_module>
  StartServers 12
  MinSpareServers 6
  MaxSpareServers 12
  MaxRequestWorkers 180
  ListenBacklog 100
  MaxKeepAliveRequests 50
  KeepAliveTimeout 20
  TimeOut 55
```

```
</IfModule>

#**** TASK 2 **********
# Enable extended status for monitoring
ExtendedStatus On

# Server status handler configuration
<Location "/server-status">
   SetHandler server-status
   Require ip 192.168.50.0/24
   Require host localhost
</Location>

# Server info handler configuration
<Location "/server-info">
   SetHandler server-info
   Require ip 192.168.50.0/24
   Require host localhost
</Location>


<IfModule dir_module>
   DirectoryIndex index.html master_project3.html
</IfModule>

#*** TASK 4 *******
Alias /q4 "/var/www/html_project3/q4/"

#******* TASK 5 *******
Alias /q5 "/var/www/html_project3/q5/"

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ".ht*">
   Require all denied
</Files>

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog "logs/error_log"

#
```

```
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel debug

<IfModule log_config_module>
    #
    # The following directives define some format nicknames for use with
    # a CustomLog directive (see below).
    #
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common Logfile Format).
    # If you do not define any access logfiles within a <VirtualHost>
    # container, they will be logged here.  Contrariwise, if you *do*
    # define per-<VirtualHost> access logfiles, transactions will be
    # logged therein and *not* in this file.
    #
    #CustomLog "logs/access_log" common

    #
    # If you prefer a logfile with access, agent, and referer information
    # (Combined Logfile Format) you can use the following directive.
    #
    CustomLog "logs/access_log" combined
</IfModule>

<IfModule alias_module>
    #
    # Redirect: Allows you to tell clients about documents that used to
    # exist in your server's namespace, but do not anymore. The client
    # will make a new request for the document at its new location.
    # Example:
    # Redirect permanent /foo http://www.example.com/bar

    #
    # Alias: Maps web paths into filesystem paths and is used to
    # access content that does not live under the DocumentRoot.
    # Example:
    # Alias /webpath /full/filesystem/path
    #
```

```
  # If you include a trailing / on /webpath then the server will
  # require it to be present in the URL.  You will also likely
  # need to provide a <Directory> section to allow access to
  # the filesystem path.

  #
  # ScriptAlias: This controls which directories contain server scripts.
  # ScriptAliases are essentially the same as Aliases, except that
  # documents in the target directory are treated as applications and
  # run by the server when requested rather than as documents sent to the
  # client.  The same rules about trailing "/" apply to ScriptAlias
  # directives as to Alias.
  #
  ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

</IfModule>

#
# "/var/www/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "/var/www/cgi-bin">
  AddHandler cgi-script .cgi .pl .py
  Options +ExecCGI
  Require all granted
</Directory>

<IfModule mime_module>
  #
  # TypesConfig points to the file containing the list of mappings from
  # filename extension to MIME-type.
  #
  TypesConfig /etc/mime.types

  #
  # AddType allows you to add to or override the MIME configuration
  # file specified in TypesConfig for specific file types.
  #
  #AddType application/x-gzip .tgz
  #
  # AddEncoding allows you to have certain browsers uncompress
  # information on the fly. Note: Not all browsers support this.
  #
  #AddEncoding x-compress .Z
  #AddEncoding x-gzip .gz .tgz
  #
  # If the AddEncoding directives above are commented-out, then you
  # probably should define those extensions to indicate media types:
  #
```

```
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz

    #
    # AddHandler allows you to map certain file extensions to "handlers":
    # actions unrelated to filetype. These can be either built into the server
    # or added with the Action directive (see below)
    #
    # To use CGI scripts outside of ScriptAliased directories:
    # (You will also need to add "ExecCGI" to the "Options" directive.)
    #
    #AddHandler cgi-script .cgi

    # For type maps (negotiated resources):
    #AddHandler type-map var

    #
    # Filters allow you to process content before it is sent to the client.
    #
    # To parse .shtml files for server-side includes (SSI):
    # (You will also need to add "Includes" to the "Options" directive.)
    #
    AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>

#
# Specify a default charset for all content served; this enables
# interpretation of all content as UTF-8 by default.  To use the
# default browser choice (ISO-8859-1), or to allow the META tags
# in HTML content to override this choice, comment out this
# directive:
#
AddDefaultCharset UTF-8

<IfModule mime_magic_module>
    #
    # The mod_mime_magic module allows the server to use various hints from the
    # contents of the file itself to determine its type.  The MIMEMagicFile
    # directive tells the module where the hint definitions are located.
    #
    MIMEMagicFile conf/magic
</IfModule>

#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
```

```
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
ServerRoot "/etc/httpd"


#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files.  This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on

# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
```

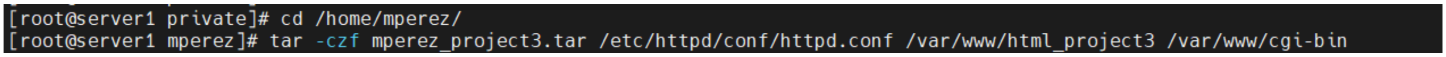
9.3 Compress file

Submit a compressed file containing of the /etc/httpd/conf/httpd.conf file and containing of /var/www/html_project3 and /var/www/cgi-bin directories.

1. Compress file

cd / home/mperez

tar -czf mperez_project3.tar /etc/httpd/conf/httpd.conf /var/www/html_project3 /var/www/cgi-bin

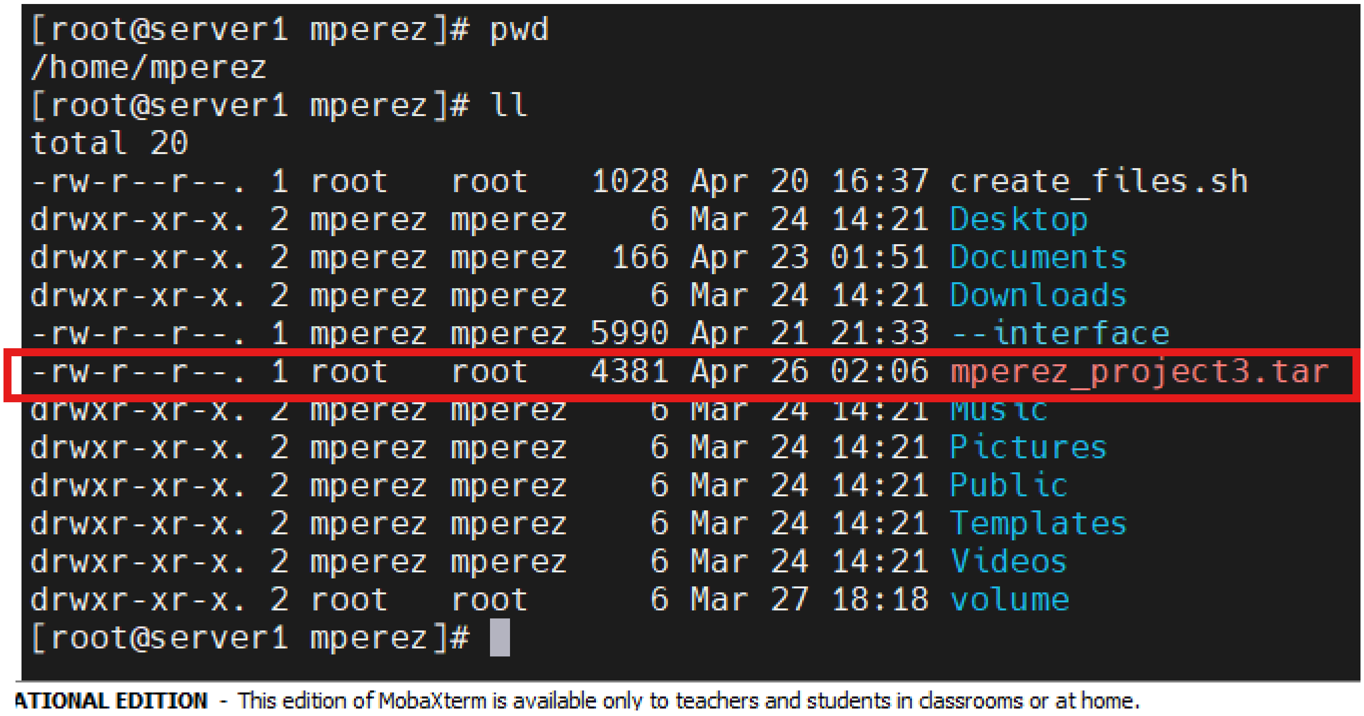


```
[root@server1 private]# cd /home/mperez/
[root@server1 mperez]# tar -czf mperez_project3.tar /etc/httpd/conf/httpd.conf /var/www/html_project3 /var/www/cgi-bin
```

2. Verify file exists

pwd

ll



```
[root@server1 mperez]# pwd
/home/mperez
[root@server1 mperez]# ll
total 20
-rw-r--r--. 1 root   root   1028 Apr 20 16:37 create_files.sh
drwxr-xr-x. 2 mperez mperez    6 Mar 24 14:21 Desktop
drwxr-xr-x. 2 mperez mperez  166 Apr 23 01:51 Documents
drwxr-xr-x. 2 mperez mperez    6 Mar 24 14:21 Downloads
-rw-r--r--. 1 mperez mperez 5990 Apr 21 21:33 --interface
-rw-r--r--. 1 root   root   4381 Apr 26 02:06 mperez_project3.tar
drwxr-xr-x. 2 mperez mperez    6 Mar 24 14:21 Music
drwxr-xr-x. 2 mperez mperez    6 Mar 24 14:21 Pictures
drwxr-xr-x. 2 mperez mperez    6 Mar 24 14:21 Public
drwxr-xr-x. 2 mperez mperez    6 Mar 24 14:21 Templates
drwxr-xr-x. 2 mperez mperez    6 Mar 24 14:21 Videos
drwxr-xr-x. 2 root   root      6 Mar 27 18:18 volume
[root@server1 mperez]# 
```

ATIONAL EDITION - This edition of MobaXterm is available only to teachers and students in classrooms or at home.