Apache Project Part-III

420-635-AB-Network Installation and Administration I

Submitted by: Sharma, Varun 4/25/25 Teacher: Antoine Tohme

Contents

Objective:

- ✓ *Optimize the performance of an Apache Server.*
- ✓ *View usage statistics and create scripts with cgi and php.*
- ✓ *Configure a secure server*

TASK 1 – MULTI-PROCESS MODULE (MPM)

- The web pages for this project should be in the directory: **/var/www/html_ project3**

```
[root@server18 www]# mkdir html_project3
```

```
DocumentRoot "/var/www/html_project3"

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

- Add links to all the web pages of this project in the following file: **var/www/html_project3/master_project3.html.**

```
[root@server18 www]# cd html_project3
[root@server18 html_project3]# vim master_project3.html
```

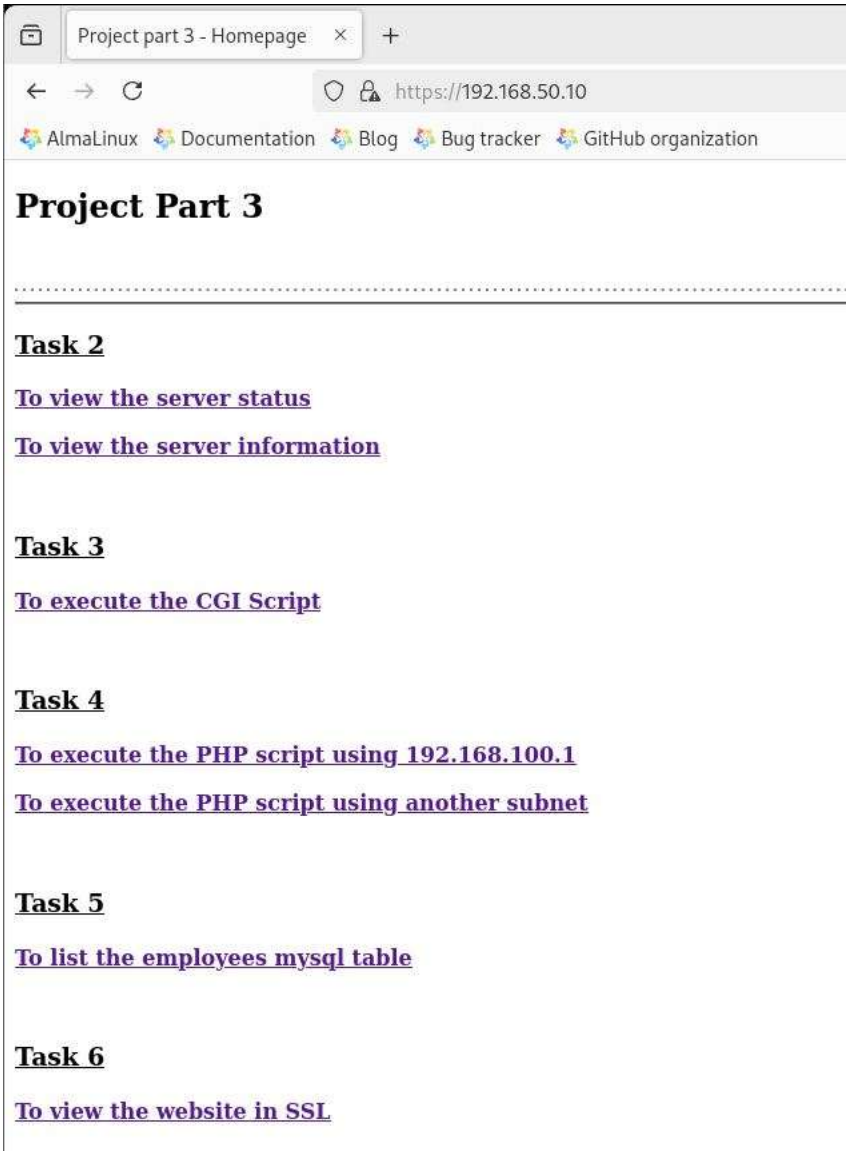
```
<!doctype html>
<html>
        <head>
                <title> Project part 3 - Homepage</title>
        </head>
        <body>
                <h2 class="title">Project Part 3</h2>
                <br>
                <hr style="border:1px dashed gray">
                <hr>

                <h3 class="subtitle"><u><b>Task 2</b></u></h3>
                <p><b><u><a href=http://192.168.50.10/server-status//>To view the server status</a></u></b></p>
                <p><b><u><a href=http://192.168.50.10/server-info//>To view the server information</a></u></b></p>
                <br>
                <h3 class="subtitle"><u><b>Task 3</b></u></h3>
                <p><b><u><a href=http://server18.local.itmt.qc.ca/cgi-bin/>To execute the CGI Script</a></u></b></p>
                <br>
                <h3 class="subtitle"><u><b>Task 4</b></u></h3>
                <p><b><u><a href=http://192.168.100.1/php/>To execute the PHP script using 192.168.100.1</a></u></b></p
>
                <p><b><u><a href=http://server18.local.itmt.qc.ca/php/>To execute the PHP script using another subnet</
a></u></b></p>

                <br>
                <h3 class="subtitle"><u><b>Task 5</b></u></h3>
                <p><b><u><a href=http://server18.local.itmt.qc.ca/employees/>To list the employees mysql table</a></u><
/b></p>
```

```
                <br>
                <h3 class="subtitle"><u><b>Task 6</b></u></h3>
                <p><b><u><a href=https://192.168.50.10/>To view the website in SSL</a></u></b></p>

        </body>
</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.

Conf file:

```
<IfModule mpm_prefork_module>
StartServers 12
MinSpareServers 6
MaxSpareServers 12
MaxRequestWorkers 180
ListenBackLog 100
KeepAlive On
MaxKeepAliveRequests 50
KeepAliveTimeout 20
Timeout 55
</IfModule>
```

```
[root@server18 conf.modules.d]# vim 00-mpm.conf
#
LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
```

TASK 2 – APACHE SERVER MONITORING

- Enable **ExtendedStatus** to allow detailed monitoring of Apache server activity.

```
<IfModule status_module>
ExtendedStatus On
</IfModule>
```

- Configure the **server-status** and **server-info handlers** so that they are accessible only to users on the **192.168.50.0/24** subnet.

```
<Location "/server-status">
    SetHandler server-status
    Require ip 192.168.50
</Location>

<Location "/server-info">
    SetHandler server-info
    Require ip 192.168.50
</Location>
```

Apache Server Status for 192.168.50.10 (via 192.168.50.10)

Server Version: Apache/2.4.62 (AlmaLinux) OpenSSL/3.2.2
Server MPM: prefork
Server Built: Jan 10 2025 00:00:00

Current Time: Friday, 25-Apr-2025 09:27:15 EDT
Restart Time: Friday, 25-Apr-2025 00:34:06 EDT
Parent Server Config. Generation: 6
Parent Server MPM Generation: 5
Server uptime: 8 hours 53 minutes 8 seconds
Server load: 0.00 0.05 0.10
Total accesses: 73 - Total Traffic: 2.0 MB - Total Duration: 494
CPU Usage: u7.55 s14.16 cu3.48 cs6.27 - .0983% CPU load
.00228 requests/sec - 64 B/second - 27.7 kB/request - 6.76712 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
0-5	21139	0/5/8	_	1.54	25398	0	29	0.0	0.01	0.44	192.168.50.10	http/1.1	server18.local.itmt.ca:80	GET / HTTP/1.1
1-5	21141	0/4/7	_	2.03	1781	6	48	0.0	0.01	0.43	192.168.50.10	http/1.1	server18.local.itmt.ca:80	GET /php/ HTTP/1.1
2-5	21142	0/4/7	_	2.03	1783	33	81	0.0	0.01	0.29	192.168.50.10	http/1.1	server18.local.itmt.ca:80	GET / HTTP/1.1
3-5	21143	0/6/9	_	2.60	109	1	112	0.0	0.01	0.30	192.168.50.10	http/1.1	server18.local.itmt.ca:80	GET /server-status/ HTTP/1.1
4-5	21137	0/4/4	_	1.33	25492	0	4	0.0	0.01	0.01	192.168.50.10	http/1.1	server18.local.itmt.ca:80	GET / HTTP/1.1
5-5	21159	0/9/11	_	2.22	570	0	176	0.0	0.16	0.30	192.168.50.10	http/1.1	server18.local.itmt.ca:80	GET /favicon.ico HTTP/1.1
6-5	21171	0/6/6	W	1.20	0	0	12	0.0	0.15	0.15	192.168.50.10	http/1.1	server18.local.itmt.ca:80	GET /server-status/ HTTP/1.1

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_a
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_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, m
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
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_prox
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_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_memcach
mod_socache_shmcb.c, mod_ssl.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

---

Server Settings

Server Version: Apache/2.4.62 (AlmaLinux) OpenSSL/3.2.2
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

TASK 3 – CGI

- Configure your web server to use **cgi-bins.**

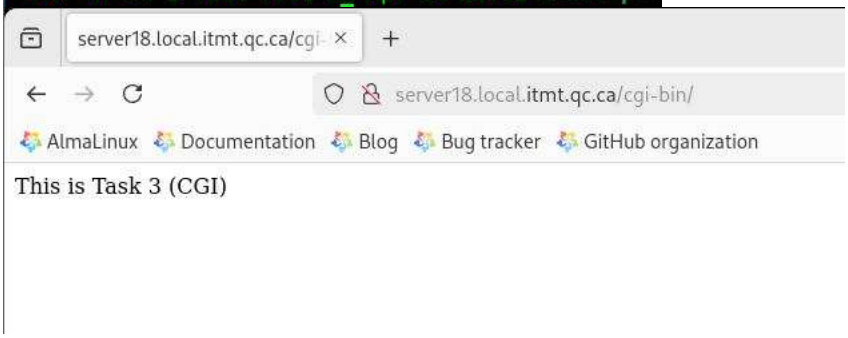
```
<IfModule alias_module>
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
</IfModule>

<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options +ExecCGI
    AddHandler cgi-script .cgi .pl .py
    Require all granted
</Directory>
```

- 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.

```
[root@server18 cgi-bin]# vim index.pl
#!/usr/bin/perl


print "content-type: text/html\n\n";
print "This is Task 3 (CGI)";
[root@server18 cgi-bin]# chmod a+x index.pl
[root@server18 cgi-bin]# ls -l index.pl
-rwxr-xr-x. 1 root root 86 Apr 24 11:32 index.pl
```



TASK 4 – PHP

- Configure your web server to use PHP.

```
[root@server18 ~]# dnf install -y php
Last metadata expiration check: 1:46:01 ago on Tue 22 Apr 2025 10:36:10 AM.
Dependencies resolved.
================================================================================
 Package              Arch       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
```

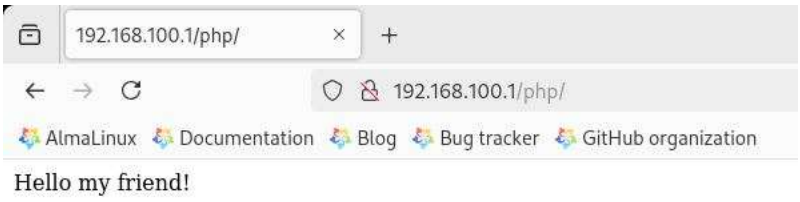
- Create a **PHP script** that displays the message **"Hello my friend!"** in the **web browser** for users whose IP address belongs to the **192.168.100.0/24** subnet and displays **"Hello stranger!"** for all other users outside this subnet.
- Place this PHP script in the **/var/www/html_project3/q4**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

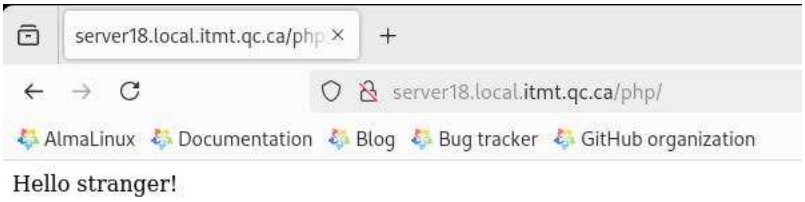
```
[root@server18 html_project3]# mkdir q4
[root@server18 html_project3]# vim q4/index.php
```

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


$subnet = '192.168.100';
$ip_parts = explode('.', $user_ip);


if ($ip_parts[0] == 192 && $ip_parts[1] == 168 && $ip_parts[2] == 100) {
    echo "Hello my friend!";
} else {
    echo "Hello stranger!";
}
?>
```





TASK 5 – MYSQL/PHP

1. Install **MySQL** and create a database **company**.

```
[root@server18 ~]# dnf -y install mysql-server
Last metadata expiration check: 3:23:51 ago on Thu 24 Apr 2025 08:33:23 AM.
Dependencies resolved.
================================================================================
 Package                      Arch       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
```

```
[root@server18 ~]# systemctl enable --now mysqld
Created symlink /etc/systemd/system/multi-user.target.wants/mysqld.service → /usr/
lib/systemd/system/mysqld.service.
[root@server18 ~]# systemctl status mysqld
● mysqld.service - MySQL 8.0 database server
     Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: dis>
     Active: active (running) since Thu 2025-04-24 11:58:34 EDT; 23s ago
    Process: 10841 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, sta>
    Process: 10863 ExecStartPre=/usr/libexec/mysql-prepare-db-dir mysqld.service >
   Main PID: 10937 (mysqld)
     Status: "Server is operational"
```

```
[root@server18 ~]# 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
```

```
mysql> create database company;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| company            |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)
```

2. Create in this database an "**employees**" table that contains two fields: **name** and **salary**.

```
mysql> use company
Database changed
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 (0.02 sec)
```

```
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)
```

3. Insert several records into the table, then verify the contents.

```
mysql> insert into employees (name, salary)
    -> values ("Adam", "10000"),
    -> ("Tony", "20000"),
    -> ("Emile", "50000");
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from employees;
+----+-------+--------+
| id | name  | salary |
+----+-------+--------+
|  1 | Adam  | 10000  |
|  2 | Tony  | 20000  |
|  3 | Emile | 50000  |
+----+-------+--------+
3 rows in set (0.00 sec)
```

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**.**

```
mysql> create user 'root'@'server18.local.itmt.qc.ca' identified by 'alma';
Query OK, 0 rows affected (0.01 sec)

mysql> grant all privileges on *.* to 'root'@'server18.local.itmt.qc.ca' with
grant option;
Query OK, 0 rows affected (0.00 sec)

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

```
[root@server18 ~]# dnf install -y php-mysqlnd
Last metadata expiration check: 0:52:35 ago on Thu 24 Apr 2025 12:11:50 PM.
Dependencies resolved.
================================================================================
 Package           Architecture Version                 Repository       Si
================================================================================
Installing:
 php-mysqlnd       x86_64       8.0.30-1.el9_2          appstream       148

Transaction Summary
================================================================================
```

```
[root@server18 ~]# systemctl restart php-fpm
[root@server18 ~]# setsebool -P httpd_can_network_connect 1
```

```
[root@server18 html_project3]# mkdir q5
```

```
[root@server18 q5]# pwd
/var/www/html_project3/q5
[root@server18 q5]# vim index.php
```

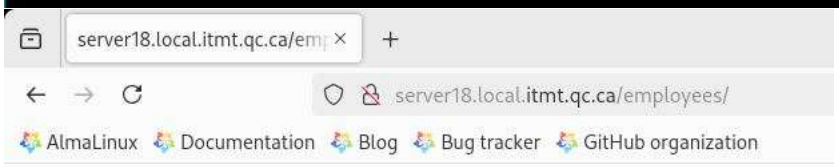
```php
<?php

// To debug
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

// Variables
$servername = "192.168.50.10";
$username = "root";
$password = "alma";
$dbname = "company";

// Create the connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Verify the connexion
if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
}

$sql = "select * from employees";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
        // Results
        while($row = $result->fetch_assoc()) {
                echo "Name: " . $row["name"]. " - Salary: " .
                $row["salary"]."$". "<br>";
                }
} else {
        echo "0 results";
}
$conn->close();
?>
```



Name: Adam - Salary: 10000$
Name: Tony - Salary: 20000$
Name: Emile - Salary: 50000$

TASK 6 – SSL

1. Configure your web server to use SSL.

```
[root@server18 ~]# dnf install -y mod_ssl
Last metadata expiration check: 1:26:23 ago on Thu 24 Apr 2025 12:11:50 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
```

```
[root@server18 ~]# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/pki/tls/private/server.key -out /etc/pki/tls/certs/server.crt
....+.....+.+.....+.........+.+.....+.+.....+.......++++++++++++++++++++++++++++
++++++++++++*.+....++++++++++++++++++++++++++++++++++++++++*.........+.+.......
....+..................+.+.........+..........+.+.....+....+..+.........+..+.
..+...+.+......+...+..+.........+......+....+.........+..+...+.+...+........+..
.+....+............+........+.+......+......+............+..+....+..+...+....+.
.+....+......+.........+..............+.......+...+.....+.+......+...........+.
...+.....+.+.....+...+.+...........+.......+..+...+.+......+.....+...+......+..
..................+.....................+...+....+..+.+............+..+.+.....
+...+....+.....+.......+...+.....+......+.+...+.........+...+.........+........
```

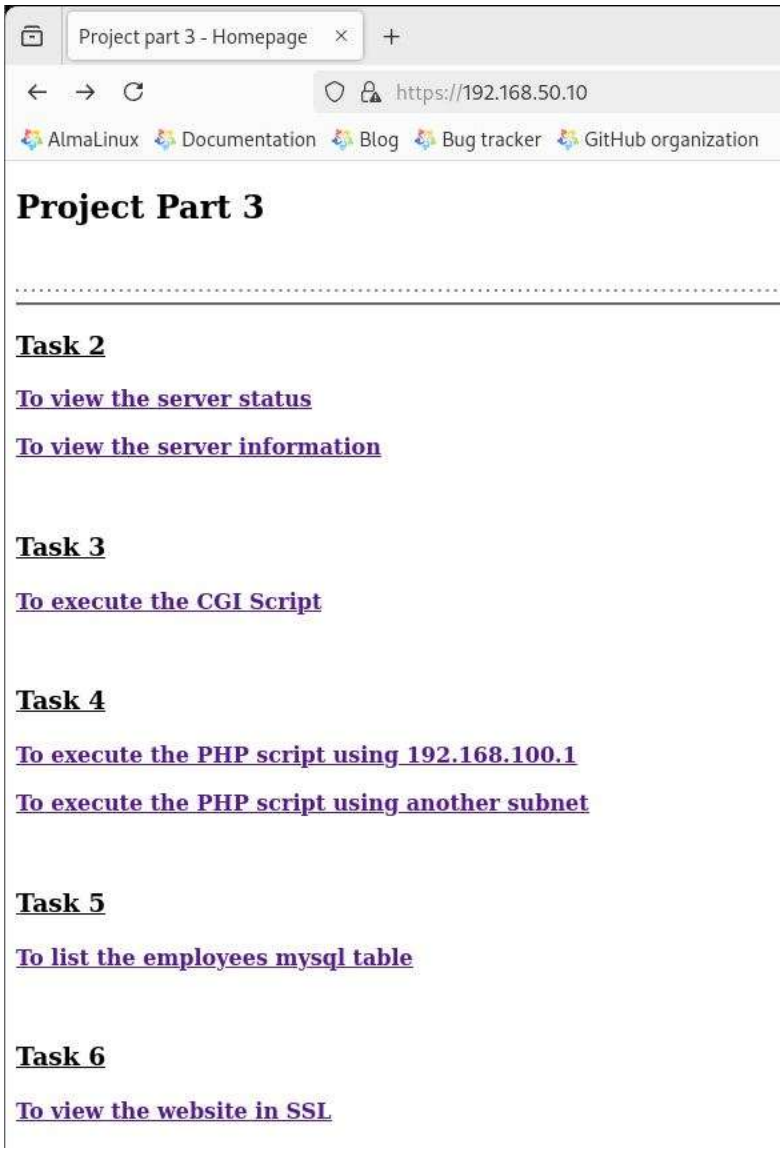
```
[root@server18 conf.d]# vim ssl.conf
```

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

```
SSLCertificateFile /etc/pki/tls/certs/server.crt

#   Server Private Key:
#   If the key is not combined with the certificate, u
#   directive to point at the key file.  Keep in mind
#   you've both a RSA and a DSA private key you can co
#   both in parallel (to also allow the use of DSA cip
#   ECC keys, when in use, can also be configured in p
SSLCertificateKeyFile /etc/pki/tls/private/server.key
```

2. Test access to your server using the **https** security protocol by navigating to: **https://*serverX*** *(where **X** is your assigned server number).*


Project part 3 - Homepage
https://192.168.50.10
AlmaLinux
Documentation
Blog
Bug tracker
GitHub organization
Project Part 3
Task 2
To view the server status
To view the server information
Task 3
To execute the CGI Script
Task 4
To execute the PHP script using 192.168.100.1
To execute the PHP script using another subnet
Task 5
To list the employees mysql table
Task 6
To view the website in SSL