Final exam

1 TASK 1

1.1 Requirements

1. Server preparation:
   - ☑ Modify the server name to **serverX.exam.com** *(where **X** is your number assigned by the teacher).*
   - ☑ Install and configure an **Apache server** to serve web pages from the **/var/www/exam** directory.
   - ☑ Make sure to:
     - Place an **index.html** web page in the directory above.
     - The index.html page must **contain links to test all the questions** in this exam.

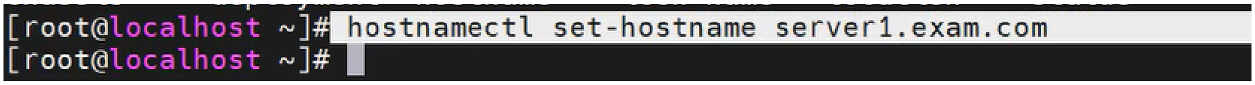
You must test both **allowed access and denied access** scenarios. *See the example provided on the last page of this document.*

1.1.1 Modify server name

1. Open a terminal

2. Rename hostname

hostnamectl set-hostname server1.exam.com



```
[root@localhost ~]# hostnamectl set-hostname server1.exam.com
[root@localhost ~]#
```

3. Verify hostname is updated

cat /etc/hostname

4. Change host name in file /etc/hosts

   a. See current name

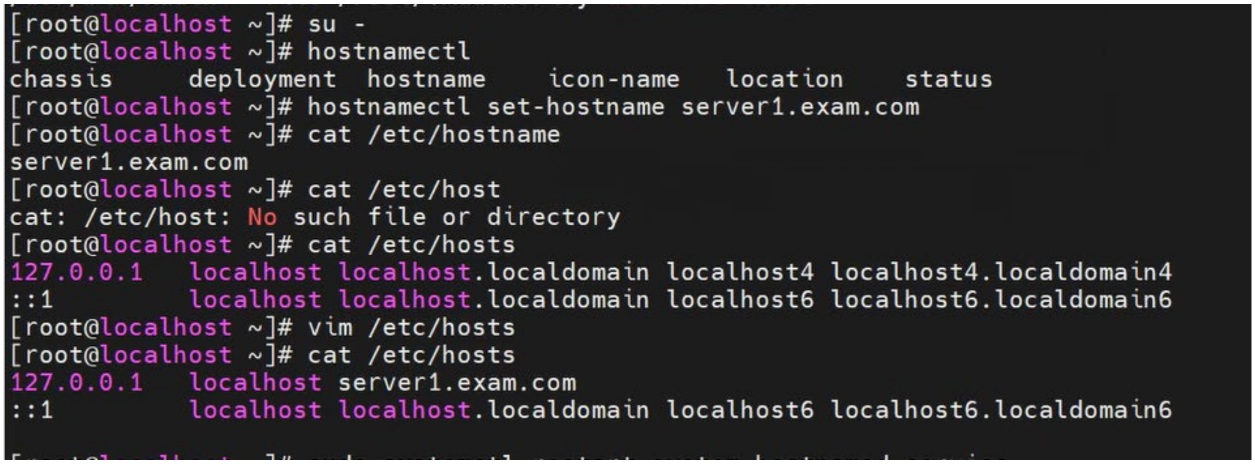
cat /etc/hosts

   b. Edit hosts file

vim /etc/hosts

   c. Verify change

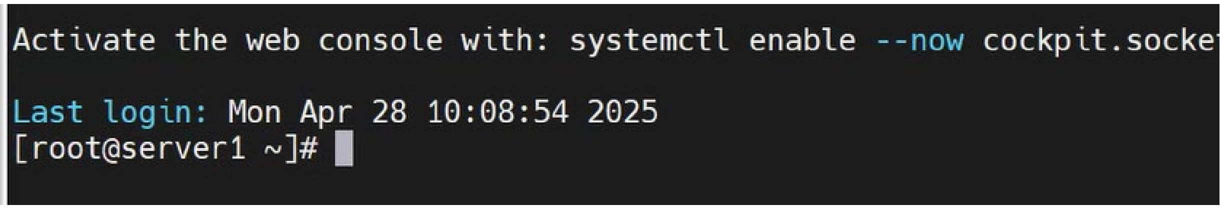
cat /etc/hosts



```
[root@localhost ~]# su -
[root@localhost ~]# hostnamectl
chassis     deployment  hostname    icon-name   location    status
[root@localhost ~]# hostnamectl set-hostname server1.exam.com
[root@localhost ~]# cat /etc/hostname
server1.exam.com
[root@localhost ~]# cat /etc/host
cat: /etc/host: No such file or directory
[root@localhost ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@localhost ~]# vim /etc/hosts
[root@localhost ~]# cat /etc/hosts
127.0.0.1   localhost server1.exam.com
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

d. Verify prompt change

Open another tab and you see the change in the prompt



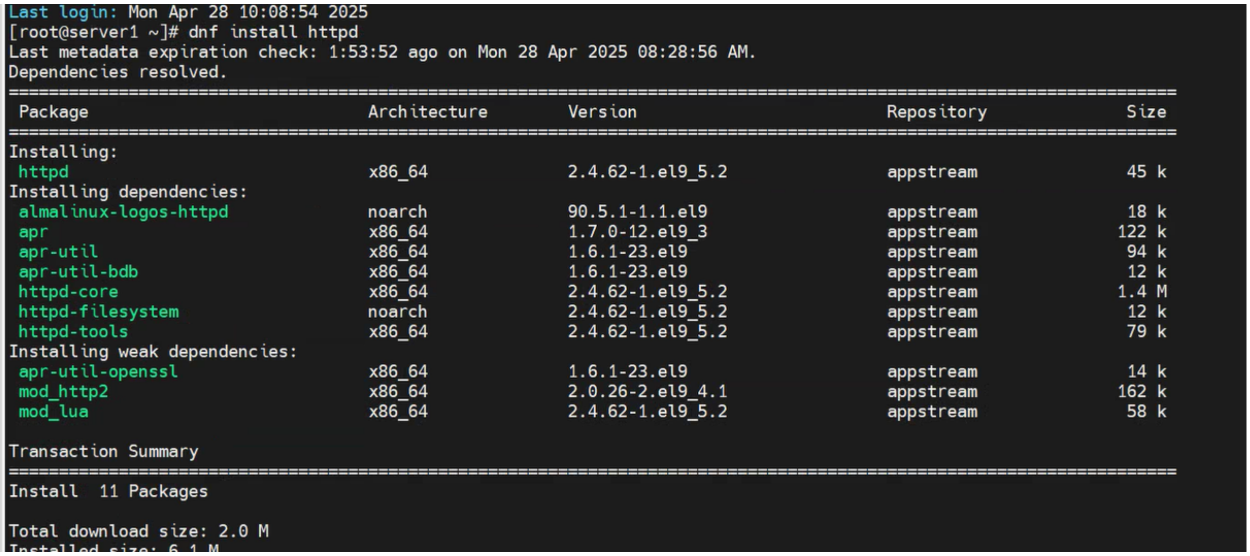
```
Activate the web console with: systemctl enable --now cockpit.socke

Last login: Mon Apr 28 10:08:54 2025
[root@server1 ~]#
```

1.1.2 Configure install apache

1.1.2.1 Installing the Apache Web Server

To begin the installation, execute the following command in the terminal:

dnf install httpd



```
Last login: Mon Apr 28 10:08:54 2025
[root@server1 ~]# dnf install httpd
Last metadata expiration check: 1:53:52 ago on Mon 28 Apr 2025 08:28:56 AM.
Dependencies resolved.
==========================================================================================================
 Package                         Architecture         Version                         Repository       Size
==========================================================================================================
Installing:
 httpd                           x86_64               2.4.62-1.el9_5.2                appstream        45 k
Installing dependencies:
 almalinux-logos-httpd           noarch               90.5.1-1.1.el9                  appstream        18 k
 apr                             x86_64               1.7.0-12.el9_3                  appstream       122 k
 apr-util                        x86_64               1.6.1-23.el9                    appstream        94 k
 apr-util-bdb                    x86_64               1.6.1-23.el9                    appstream        12 k
 httpd-core                      x86_64               2.4.62-1.el9_5.2                appstream       1.4 M
 httpd-filesystem                noarch               2.4.62-1.el9_5.2                appstream        12 k
 httpd-tools                     x86_64               2.4.62-1.el9_5.2                appstream        79 k
Installing weak dependencies:
 apr-util-openssl                x86_64               1.6.1-23.el9                    appstream        14 k
 mod_http2                       x86_64               2.0.26-2.el9_4.1                appstream       162 k
 mod_lua                         x86_64               2.4.62-1.el9_5.2                appstream        58 k

Transaction Summary
==========================================================================================================
Install  11 Packages

Total download size: 2.0 M
Installed size: 6.1 M
```

1.1.2.2 Manage the httpd service

a) Start httpd to initiate the Apache service

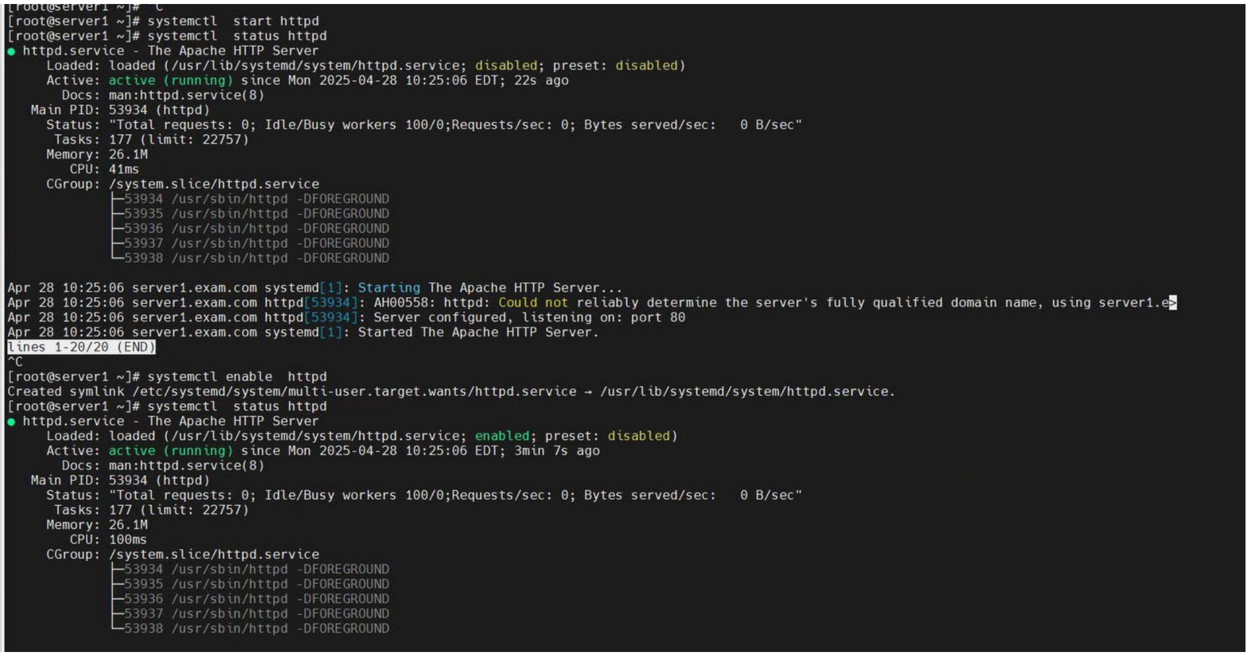
systemctl start httpd

systemctl enable httpd

b) Verify the status, by default status is disabled

systemctl status httpd

Note httpd is **active (running)** but service is disabled



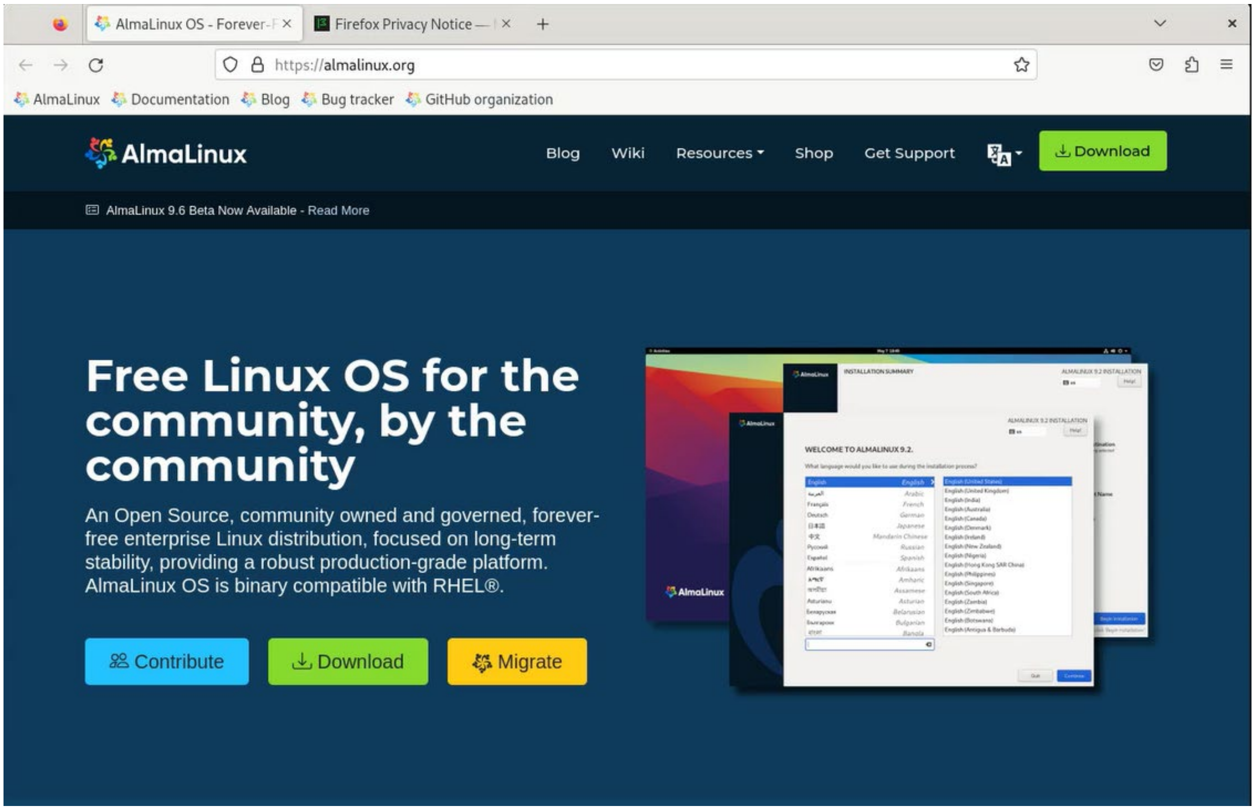
```
[root@server1 ~]# ^C
[root@server1 ~]# systemctl  start httpd
[root@server1 ~]# systemctl  status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Mon 2025-04-28 10:25:06 EDT; 22s ago
       Docs: man:httpd.service(8)
   Main PID: 53934 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 22757)
     Memory: 26.1M
        CPU: 41ms
     CGroup: /system.slice/httpd.service
             ├─53934 /usr/sbin/httpd -DFOREGROUND
             ├─53935 /usr/sbin/httpd -DFOREGROUND
             ├─53936 /usr/sbin/httpd -DFOREGROUND
             ├─53937 /usr/sbin/httpd -DFOREGROUND
             └─53938 /usr/sbin/httpd -DFOREGROUND

Apr 28 10:25:06 server1.exam.com systemd[1]: Starting The Apache HTTP Server...
Apr 28 10:25:06 server1.exam.com httpd[53934]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using server1.e>
Apr 28 10:25:06 server1.exam.com httpd[53934]: Server configured, listening on: port 80
Apr 28 10:25:06 server1.exam.com systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
^C
[root@server1 ~]# systemctl enable  httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[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 Mon 2025-04-28 10:25:06 EDT; 3min 7s ago
       Docs: man:httpd.service(8)
   Main PID: 53934 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 22757)
     Memory: 26.1M
        CPU: 100ms
     CGroup: /system.slice/httpd.service
             ├─53934 /usr/sbin/httpd -DFOREGROUND
             ├─53935 /usr/sbin/httpd -DFOREGROUND
             ├─53936 /usr/sbin/httpd -DFOREGROUND
             ├─53937 /usr/sbin/httpd -DFOREGROUND
             └─53938 /usr/sbin/httpd -DFOREGROUND
```

1.1.2.3 Verify Apache is working on localhost

Open firefox

firefox &

Per default Alma Linux page opens



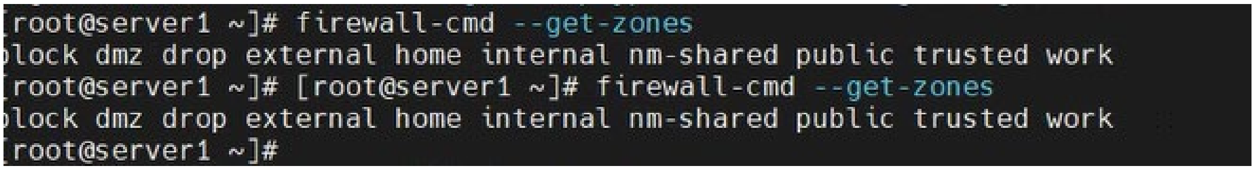
AlmaLinux 9.6 Beta Now Available - Read More

Free Linux OS for the community, by the community

An Open Source, community owned and governed, forever-free enterprise Linux distribution, focused on long-term stability, providing a robust production-grade platform. AlmaLinux OS is binary compatible with RHEL®.

1.1.2.4 Firewall

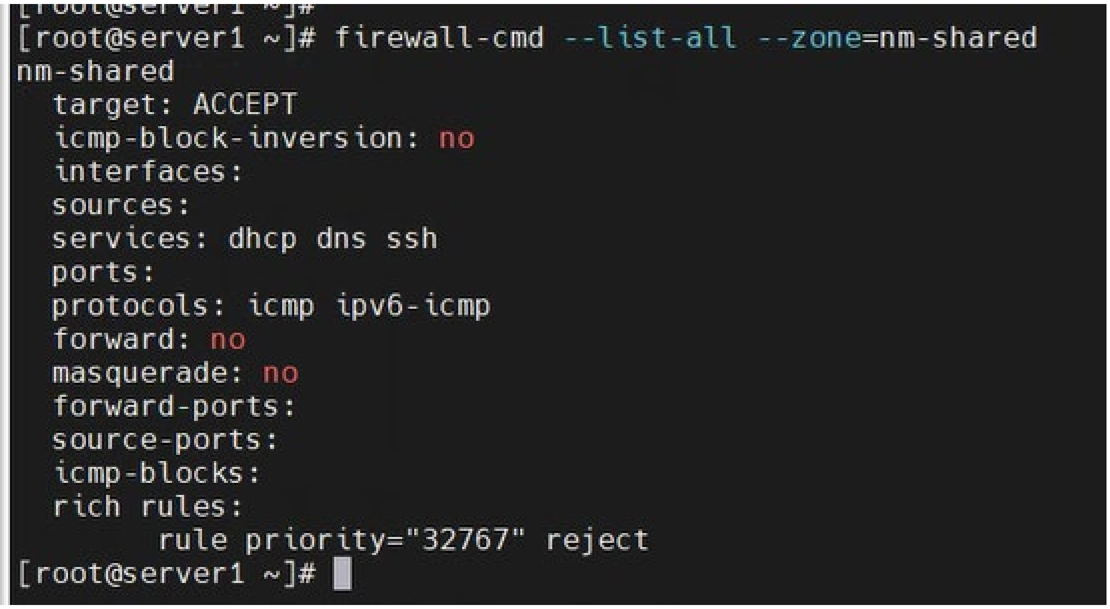
1. Verify existing active zones
   firewall-cmd --get-active-zones



```
[root@server1 ~]# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
[root@server1 ~]# [root@server1 ~]# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
[root@server1 ~]#
```

2. List the firewall sets for the zone

```
firewall-cmd --list-all --zone=nm-shared
```



```
[root@server1 ~]#
[root@server1 ~]# firewall-cmd --list-all --zone=nm-shared
nm-shared
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcp dns ssh
  ports:
  protocols: icmp ipv6-icmp
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule priority="32767" reject
[root@server1 ~]#
```

3. Add a rule that allows TCP traffic on port 80 for nm-shared zone, execute the following command:

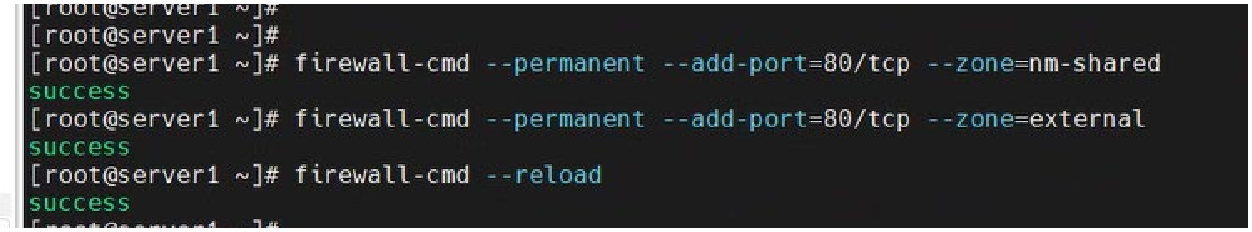
```
firewall-cmd –permanent –add-port=80/tcp –zone=nm-shared
```

4. Reload to apply permanent firewall rules to the running configuration.

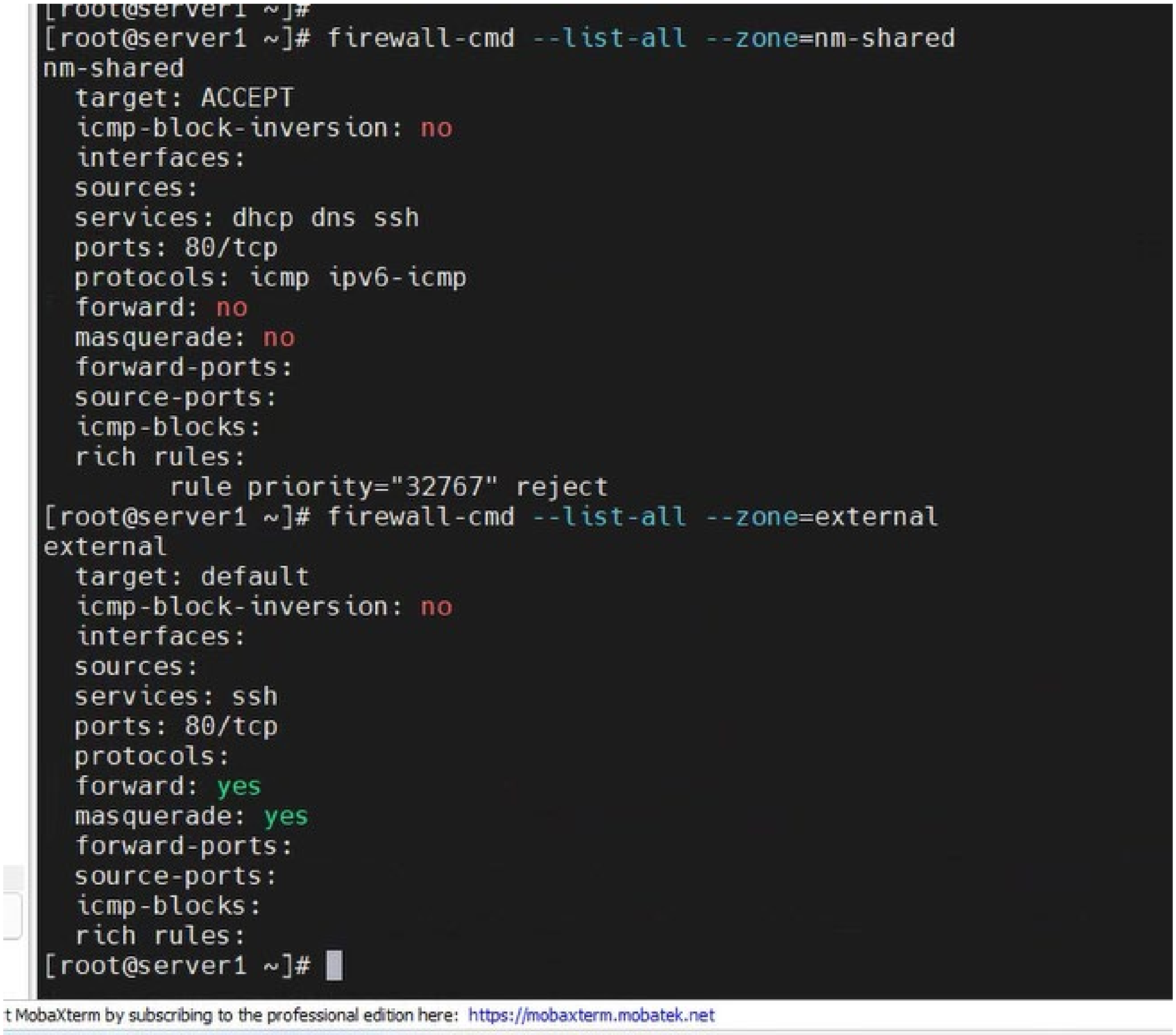
```
sudo firewall-cmd –reload
```

5. List the firewall sets for the zone

```
sudo firewall-cmd --list-all --zone=nm-shared
```



```
[root@server1 ~]#
[root@server1 ~]#
[root@server1 ~]# firewall-cmd --permanent --add-port=80/tcp --zone=nm-shared
success
[root@server1 ~]# firewall-cmd --permanent --add-port=80/tcp --zone=external
success
[root@server1 ~]# firewall-cmd --reload
success
```



```
[root@server1 ~]#
[root@server1 ~]# firewall-cmd --list-all --zone=nm-shared
nm-shared
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcp dns ssh
  ports: 80/tcp
  protocols: icmp ipv6-icmp
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule priority="32767" reject
[root@server1 ~]# firewall-cmd --list-all --zone=external
external
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports: 80/tcp
  protocols:
  forward: yes
  masquerade: yes
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@server1 ~]# █
```

t MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

1.1.2.5 Configure Apache

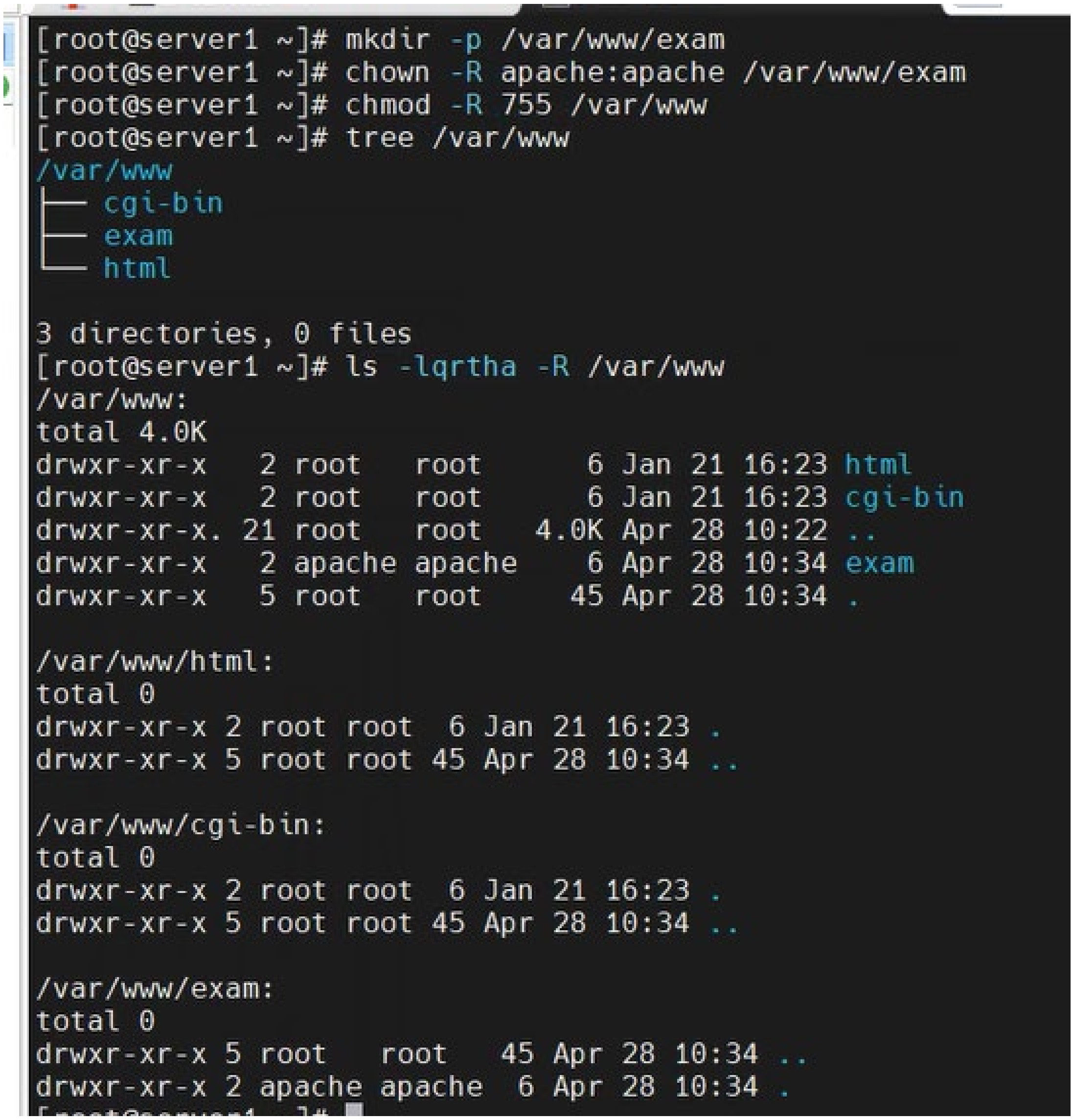
Install and configure an Apache server to serve web pages from the /var/www/exam directory.

Create the web directory:

```
sudo mkdir -p /var/www/exam
```

Set proper permissions:

```
sudo chown -R apache:apache /var/www/exam
sudo chmod -R 755 /var/www
```

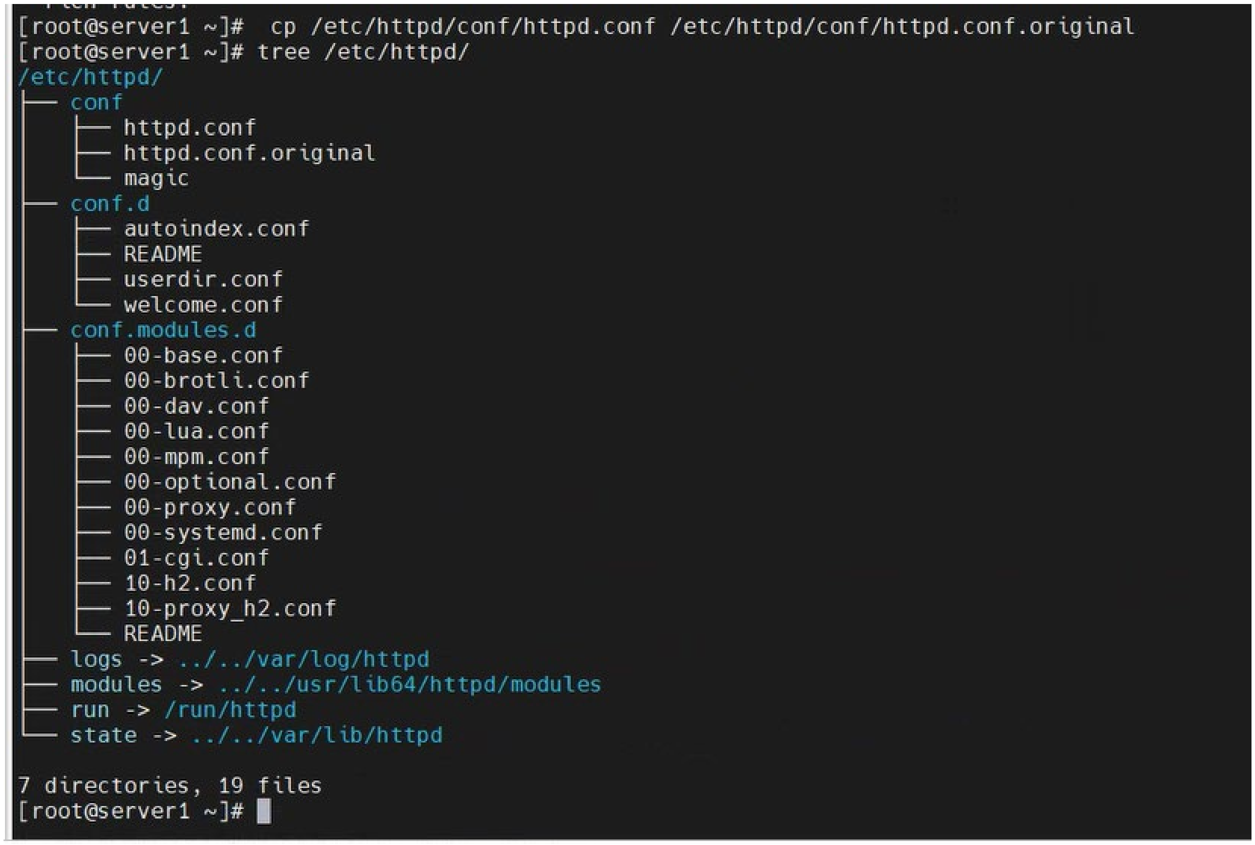


```
[root@server1 ~]# mkdir -p /var/www/exam
[root@server1 ~]# chown -R apache:apache /var/www/exam
[root@server1 ~]# chmod -R 755 /var/www
[root@server1 ~]# tree /var/www
/var/www
├── cgi-bin
├── exam
└── html

3 directories, 0 files
[root@server1 ~]# ls -lqrtha -R /var/www
/var/www:
total 4.0K
drwxr-xr-x   2 root   root      6 Jan 21 16:23 html
drwxr-xr-x   2 root   root      6 Jan 21 16:23 cgi-bin
drwxr-xr-x. 21 root   root   4.0K Apr 28 10:22 ..
drwxr-xr-x   2 apache apache    6 Apr 28 10:34 exam
drwxr-xr-x   5 root   root     45 Apr 28 10:34 .

/var/www/html:
total 0
drwxr-xr-x 2 root root  6 Jan 21 16:23 .
drwxr-xr-x 5 root root 45 Apr 28 10:34 ..

/var/www/cgi-bin:
total 0
drwxr-xr-x 2 root root  6 Jan 21 16:23 .
drwxr-xr-x 5 root root 45 Apr 28 10:34 ..

/var/www/exam:
total 0
drwxr-xr-x 5 root   root   45 Apr 28 10:34 ..
drwxr-xr-x 2 apache apache  6 Apr 28 10:34 .
[root@server1 ~]#
```

1.1.3 Httpd.conf

1.1.3.1 Backup

1. Copy the httpd.conf file to httpd.conf.original



```
  rich rules:
[root@server1 ~]#  cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.original
[root@server1 ~]# tree /etc/httpd/
/etc/httpd/
├── conf
│   ├── httpd.conf
│   ├── httpd.conf.original
│   └── magic
├── conf.d
│   ├── autoindex.conf
│   ├── README
│   ├── userdir.conf
│   └── welcome.conf
├── conf.modules.d
│   ├── 00-base.conf
│   ├── 00-brotli.conf
│   ├── 00-dav.conf
│   ├── 00-lua.conf
│   ├── 00-mpm.conf
│   ├── 00-optional.conf
│   ├── 00-proxy.conf
│   ├── 00-systemd.conf
│   ├── 01-cgi.conf
│   ├── 10-h2.conf
│   ├── 10-proxy_h2.conf
│   └── README
├── logs -> ../../var/log/httpd
├── modules -> ../../usr/lib64/httpd/modules
├── run -> /run/httpd
└── state -> ../../var/lib/httpd

7 directories, 19 files
[root@server1 ~]# █
```

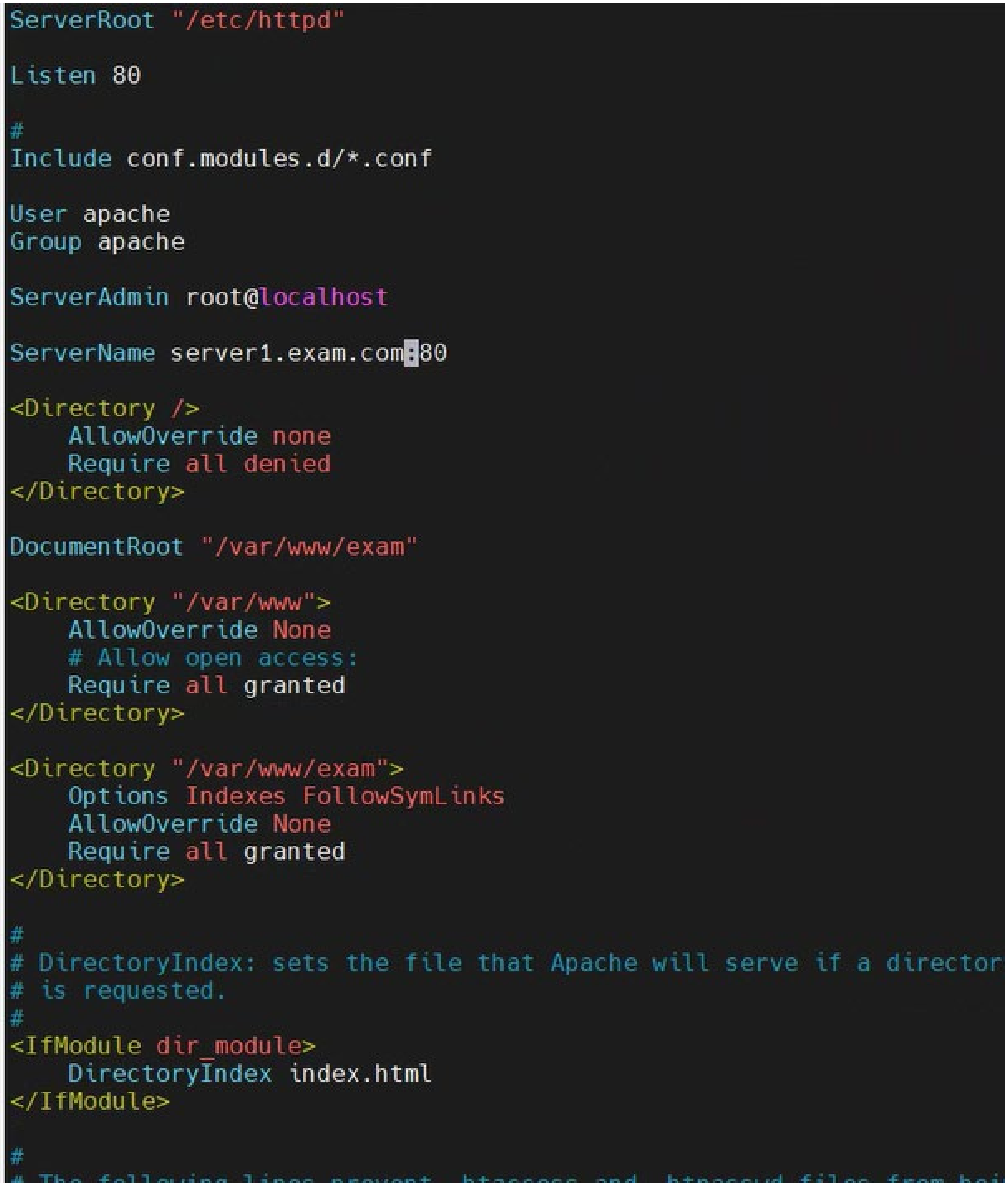
1.1.3.2 Modify httpd.conf

Current Configuration	**Updated Configuration**
ServerName	server1.exam.com
DocumentRoot "/var/www/html"	DocumentRoot "**/var/www/exam**"

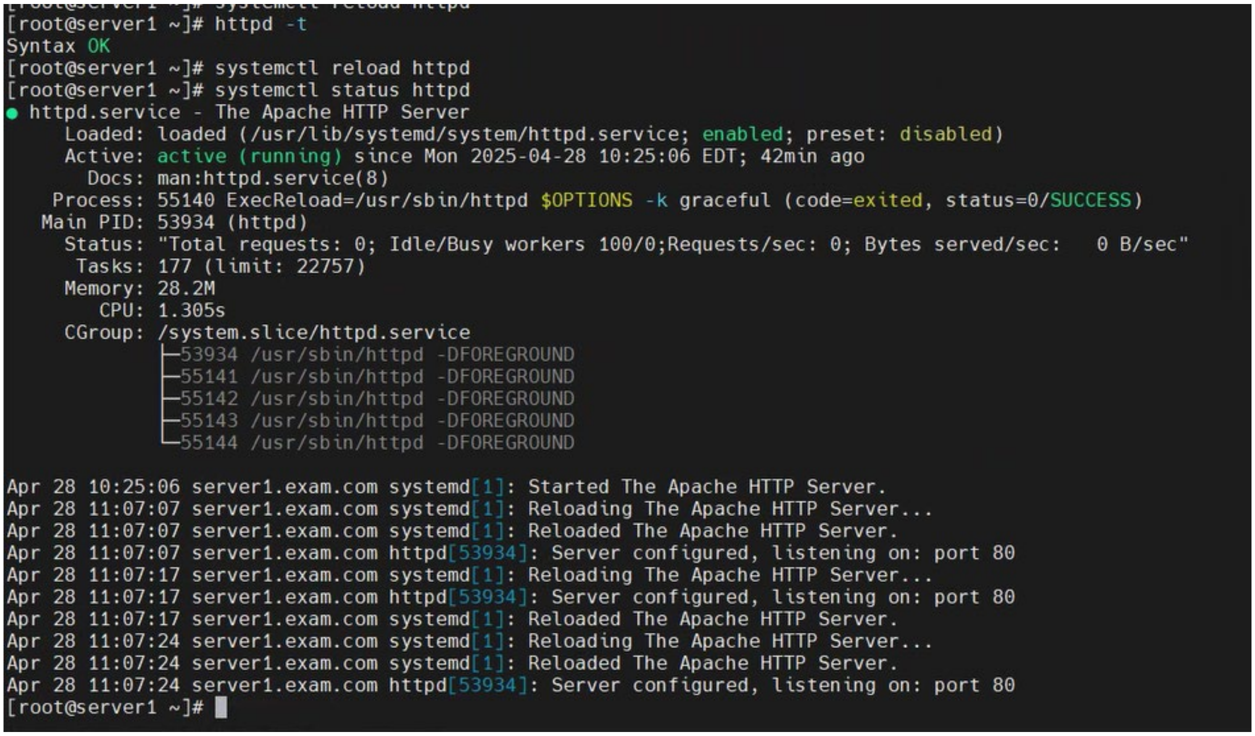
<Directory "/var/www/html">	<Directory "/var/www/exam">
	  Options Indexes FollowSymLinks
	  AllowOverride All
	  Require all granted
	</Directory>
	am

Open the file for edition and update the configuration according to table above

vim /etc/httpd/conf/httpd.conf



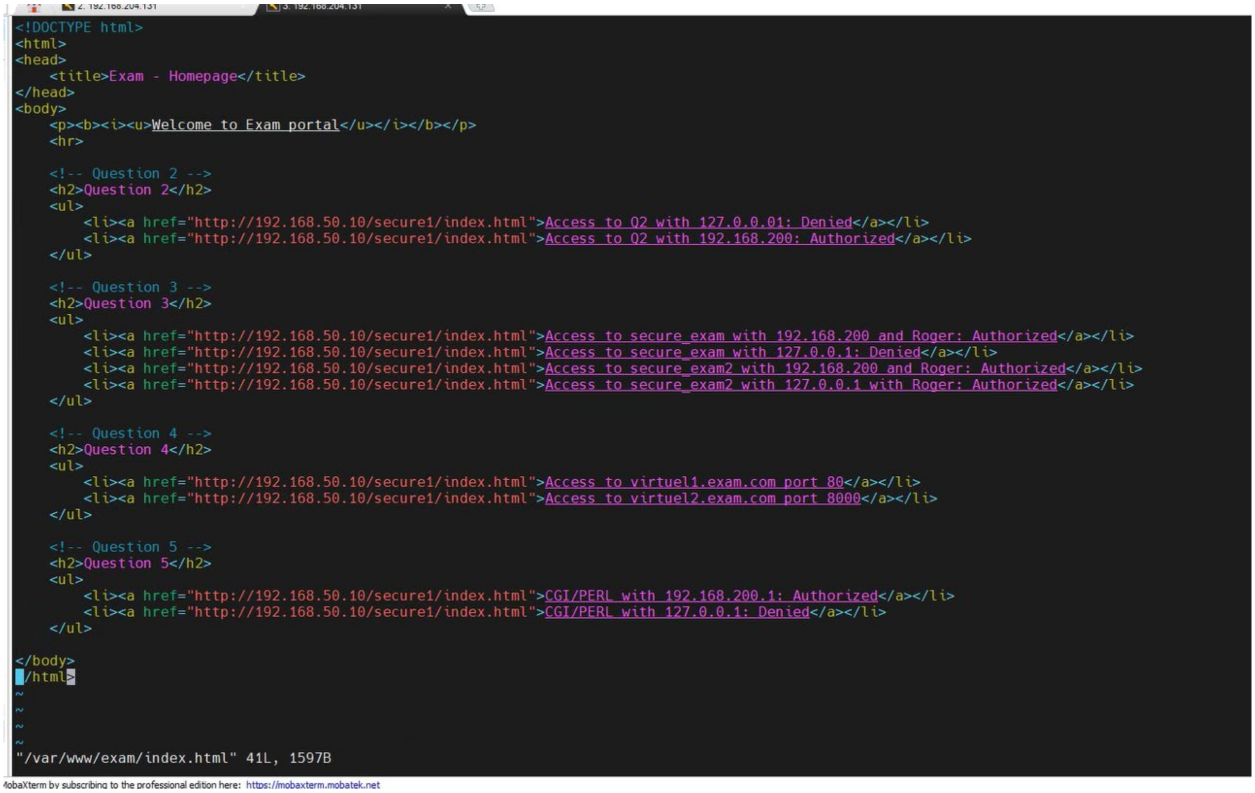
```
ServerRoot "/etc/httpd"

Listen 80

#
Include conf.modules.d/*.conf

User apache
Group apache

ServerAdmin root@localhost

ServerName server1.exam.com:80

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

DocumentRoot "/var/www/exam"

<Directory "/var/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>

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

#
# DirectoryIndex: sets the file that Apache will serve if a director
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>

#
# The following lines prevent .htaccess and .htpasswd files from bei
```



```
[root@server1 ~]# systemctl reload httpd
[root@server1 ~]# httpd -t
Syntax OK
[root@server1 ~]# systemctl reload 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 Mon 2025-04-28 10:25:06 EDT; 42min ago
       Docs: man:httpd.service(8)
    Process: 55140 ExecReload=/usr/sbin/httpd $OPTIONS -k graceful (code=exited, status=0/SUCCESS)
   Main PID: 53934 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 22757)
     Memory: 28.2M
        CPU: 1.305s
     CGroup: /system.slice/httpd.service
             ├─53934 /usr/sbin/httpd -DFOREGROUND
             ├─55141 /usr/sbin/httpd -DFOREGROUND
             ├─55142 /usr/sbin/httpd -DFOREGROUND
             ├─55143 /usr/sbin/httpd -DFOREGROUND
             └─55144 /usr/sbin/httpd -DFOREGROUND

Apr 28 10:25:06 server1.exam.com systemd[1]: Started The Apache HTTP Server.
Apr 28 11:07:07 server1.exam.com systemd[1]: Reloading The Apache HTTP Server...
Apr 28 11:07:07 server1.exam.com systemd[1]: Reloaded The Apache HTTP Server.
Apr 28 11:07:07 server1.exam.com httpd[53934]: Server configured, listening on: port 80
Apr 28 11:07:17 server1.exam.com systemd[1]: Reloading The Apache HTTP Server...
Apr 28 11:07:17 server1.exam.com httpd[53934]: Server configured, listening on: port 80
Apr 28 11:07:17 server1.exam.com systemd[1]: Reloaded The Apache HTTP Server.
Apr 28 11:07:24 server1.exam.com systemd[1]: Reloading The Apache HTTP Server...
Apr 28 11:07:24 server1.exam.com systemd[1]: Reloaded The Apache HTTP Server.
Apr 28 11:07:24 server1.exam.com httpd[53934]: Server configured, listening on: port 80
[root@server1 ~]#
```

1.1.4 Create the index file:

vim /var/www/exam/index.html

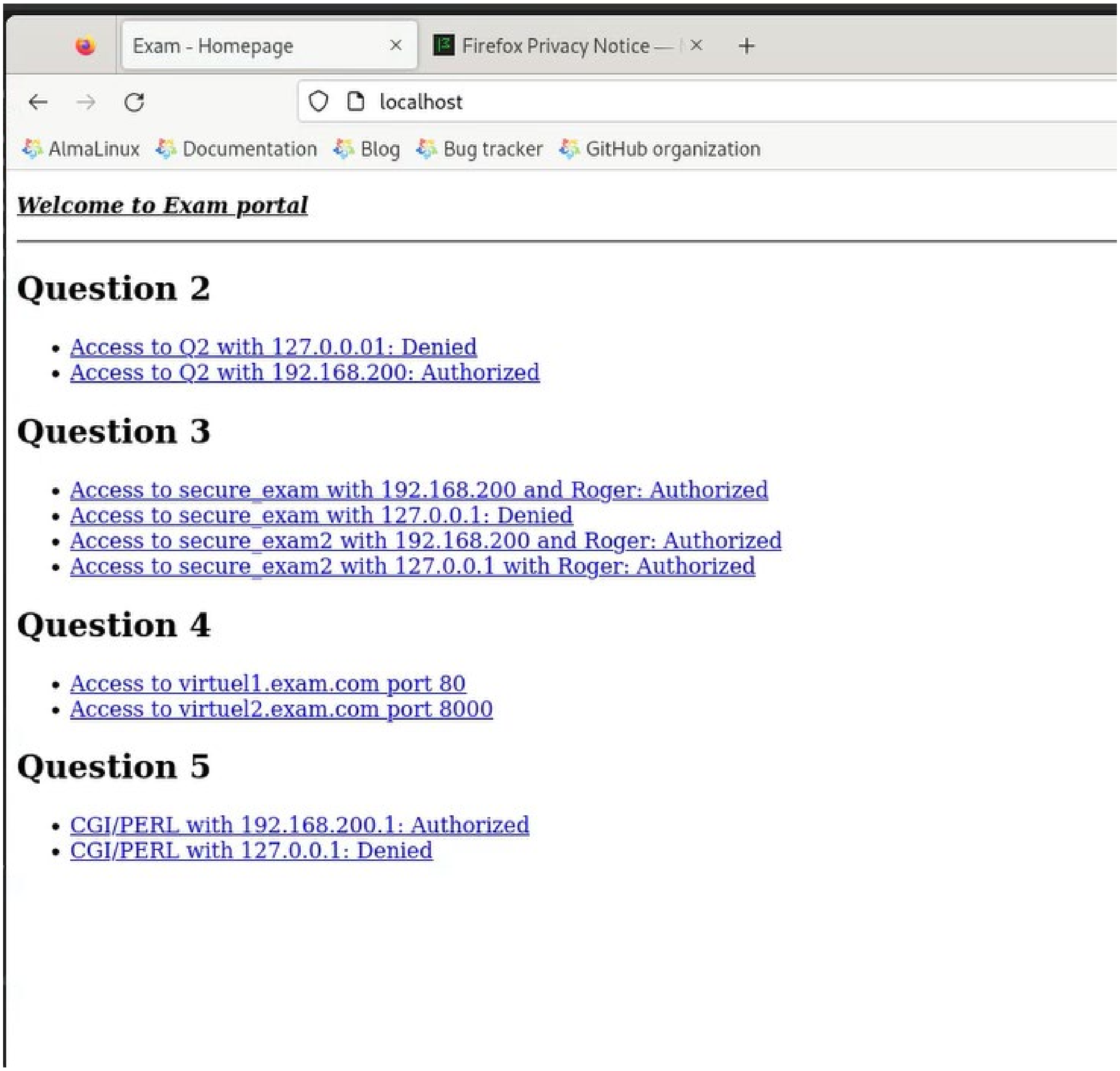


```
<!DOCTYPE html>
<html>
<head>
    <title>Exam - Homepage</title>
</head>
<body>
    <p><b><i><u>Welcome to Exam portal</u></i></b></p>
    <hr>

    <!-- Question 2 -->
    <h2>Question 2</h2>
    <ul>
        <li><a href="http://192.168.50.10/secure1/index.html">Access to Q2 with 127.0.0.01: Denied</a></li>
        <li><a href="http://192.168.50.10/secure1/index.html">Access to Q2 with 192.168.200: Authorized</a></li>
    </ul>

    <!-- Question 3 -->
    <h2>Question 3</h2>
    <ul>
        <li><a href="http://192.168.50.10/secure1/index.html">Access to secure_exam with 192.168.200 and Roger: Authorized</a></li>
        <li><a href="http://192.168.50.10/secure1/index.html">Access to secure_exam with 127.0.0.1: Denied</a></li>
        <li><a href="http://192.168.50.10/secure1/index.html">Access to secure_exam2 with 192.168.200 and Roger: Authorized</a></li>
        <li><a href="http://192.168.50.10/secure1/index.html">Access to secure_exam2 with 127.0.0.1 with Roger: Authorized</a></li>
    </ul>

    <!-- Question 4 -->
    <h2>Question 4</h2>
    <ul>
        <li><a href="http://192.168.50.10/secure1/index.html">Access to virtuel1.exam.com port 80</a></li>
        <li><a href="http://192.168.50.10/secure1/index.html">Access to virtuel2.exam.com port 8000</a></li>
    </ul>

    <!-- Question 5 -->
    <h2>Question 5</h2>
    <ul>
        <li><a href="http://192.168.50.10/secure1/index.html">CGI/PERL with 192.168.200.1: Authorized</a></li>
        <li><a href="http://192.168.50.10/secure1/index.html">CGI/PERL with 127.0.0.1: Denied</a></li>
    </ul>

</body>
</html>
~
~
~
~
"/var/www/exam/index.html" 41L, 1597B
```

/obaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

Test

Open Firefox (links do not wokr)



Welcome to Exam portal

Question 2

- Access to Q2 with 127.0.0.01: Denied
- Access to Q2 with 192.168.200: Authorized

Question 3

- Access to secure_exam with 192.168.200 and Roger: Authorized
- Access to secure_exam with 127.0.0.1: Denied
- Access to secure_exam2 with 192.168.200 and Roger: Authorized
- Access to secure_exam2 with 127.0.0.1 with Roger: Authorized

Question 4

- Access to virtuel1.exam.com port 80
- Access to virtuel2.exam.com port 8000

Question 5

- CGI/PERL with 192.168.200.1: Authorized
- CGI/PERL with 127.0.0.1: Denied

2 TASK2

2.1 Requirements

2. Website creation:

☑ Create the directory **/var/www/exam/q2** and configure the Apache server so that it is only accessible to computers on the **192.168.200** subnet.

Create a web page that automatically displays in the browser when the directory address is entered. The page must display the text "**Q2**".

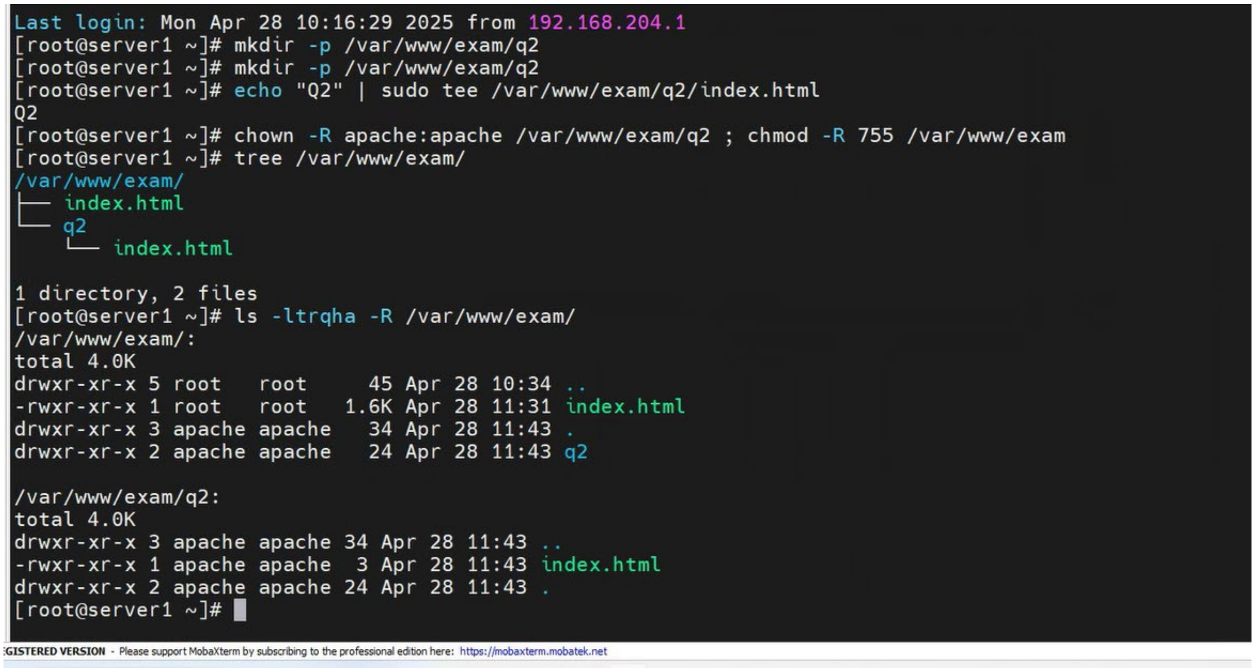
2.2 Configurations

1. **Create the Q2 directory and content:**

   mkdir -p /var/www/exam/q2

   echo "Q2" | sudo tee /var/www/exam/q2/index.html

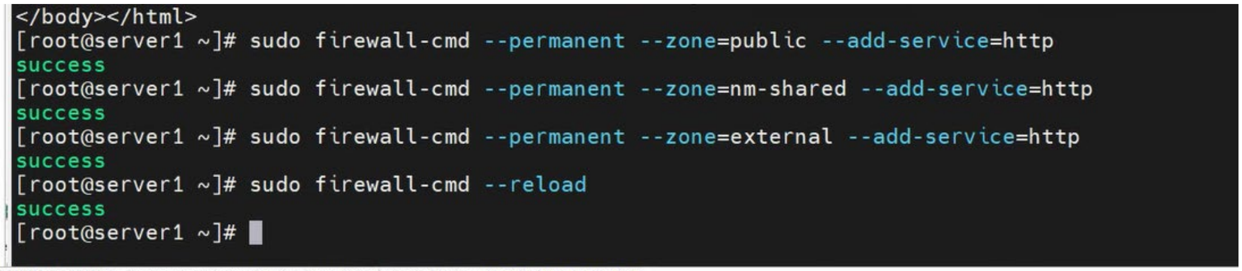
   chown -R apache:apache /var/www/exam/q2 ; chmod -R 755 /var/www/exam



```
Last login: Mon Apr 28 10:16:29 2025 from 192.168.204.1
[root@server1 ~]# mkdir -p /var/www/exam/q2
[root@server1 ~]# mkdir -p /var/www/exam/q2
[root@server1 ~]# echo "Q2" | sudo tee /var/www/exam/q2/index.html
Q2
[root@server1 ~]# chown -R apache:apache /var/www/exam/q2 ; chmod -R 755 /var/www/exam
[root@server1 ~]# tree /var/www/exam/
/var/www/exam/
├── index.html
└── q2
    └── index.html

1 directory, 2 files
[root@server1 ~]# ls -ltrqha -R /var/www/exam/
/var/www/exam/:
total 4.0K
drwxr-xr-x 5 root   root     45 Apr 28 10:34 ..
-rwxr-xr-x 1 root   root   1.6K Apr 28 11:31 index.html
drwxr-xr-x 3 apache apache   34 Apr 28 11:43 .
drwxr-xr-x 2 apache apache   24 Apr 28 11:43 q2

/var/www/exam/q2:
total 4.0K
drwxr-xr-x 3 apache apache 34 Apr 28 11:43 ..
-rwxr-xr-x 1 apache apache  3 Apr 28 11:43 index.html
drwxr-xr-x 2 apache apache 24 Apr 28 11:43 .
[root@server1 ~]#
```

:GISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

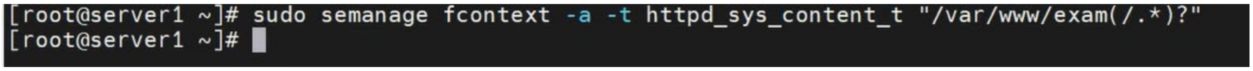
Firewall



```
</body></html>
[root@server1 ~]# sudo firewall-cmd --permanent --zone=public --add-service=http
success
[root@server1 ~]# sudo firewall-cmd --permanent --zone=nm-shared --add-service=http
success
[root@server1 ~]# sudo firewall-cmd --permanent --zone=external --add-service=http
success
[root@server1 ~]# sudo firewall-cmd --reload
success
[root@server1 ~]#
```

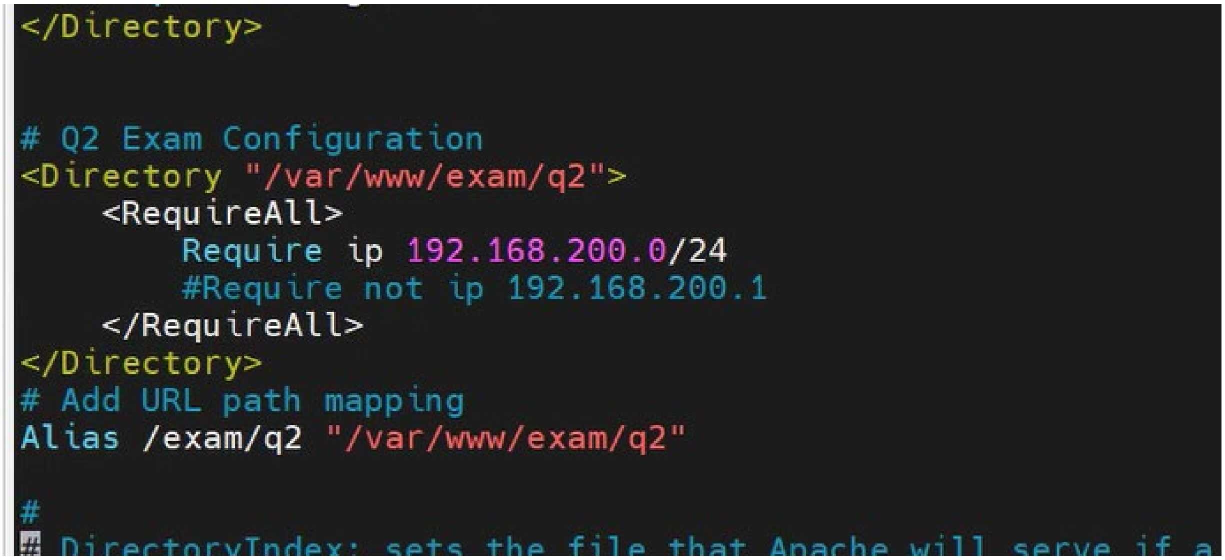
REGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

SELINUX



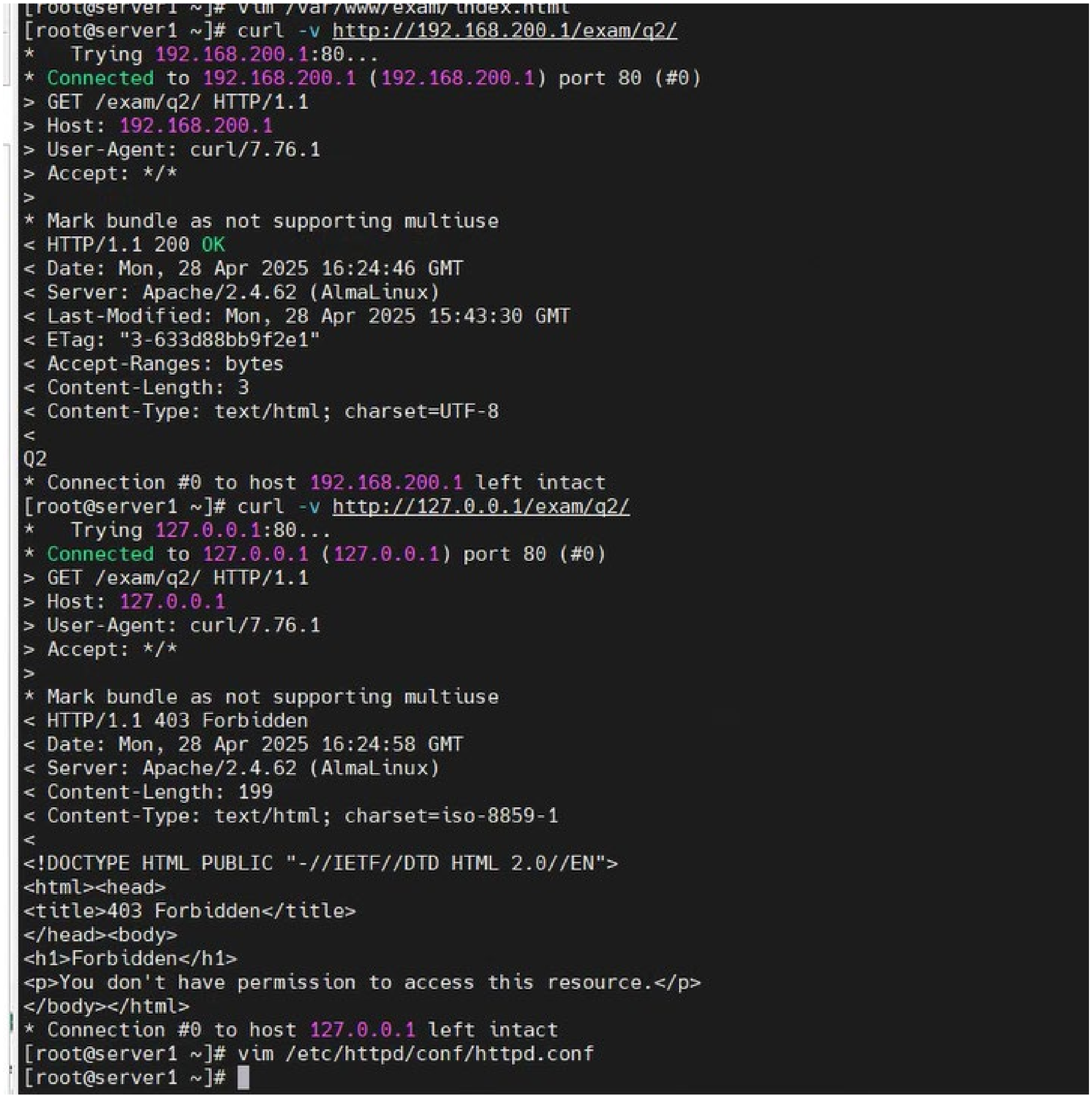
```
[root@server1 ~]# sudo semanage fcontext -a -t httpd_sys_content_t "/var/www/exam(/.*)?"
[root@server1 ~]#
```

Modify httpd.conf

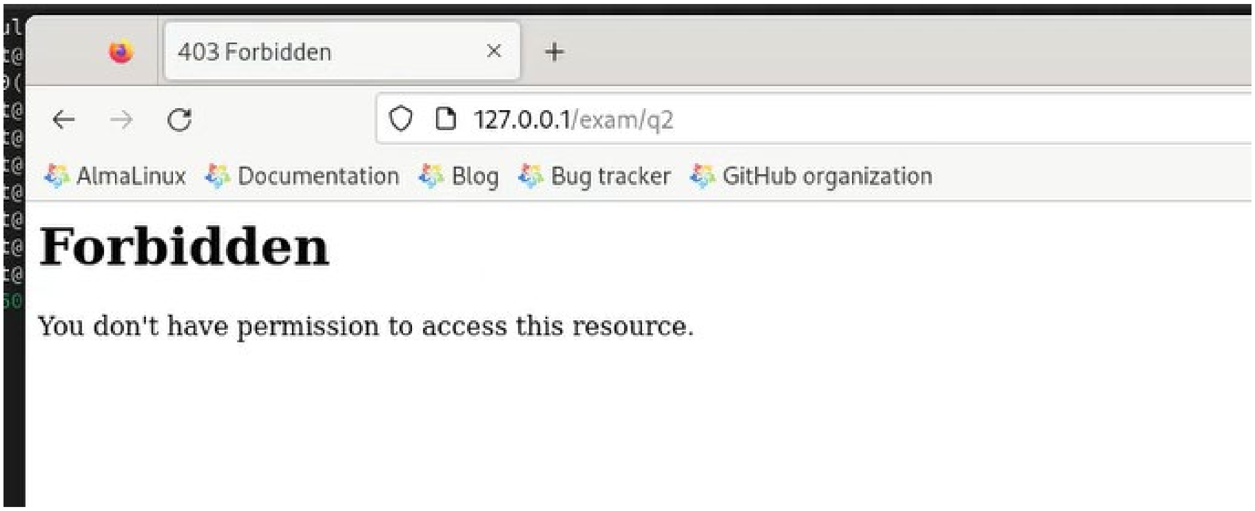


```
</Directory>


# Q2 Exam Configuration
<Directory "/var/www/exam/q2">
    <RequireAll>
        Require ip 192.168.200.0/24
        #Require not ip 192.168.200.1
    </RequireAll>
</Directory>
# Add URL path mapping
Alias /exam/q2 "/var/www/exam/q2"

#
# DirectoryIndex: sets the file that Apache will serve if a
```

Curl test

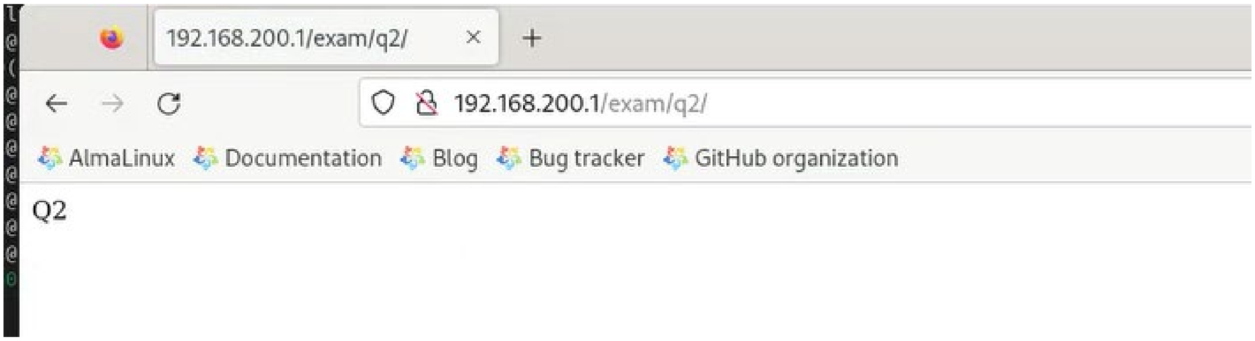


```
[root@server1 ~]# vim /var/www/exam/index.html
[root@server1 ~]# curl -v http://192.168.200.1/exam/q2/
*   Trying 192.168.200.1:80...
* Connected to 192.168.200.1 (192.168.200.1) port 80 (#0)
> GET /exam/q2/ HTTP/1.1
> Host: 192.168.200.1
> User-Agent: curl/7.76.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Mon, 28 Apr 2025 16:24:46 GMT
< Server: Apache/2.4.62 (AlmaLinux)
< Last-Modified: Mon, 28 Apr 2025 15:43:30 GMT
< ETag: "3-633d88bb9f2e1"
< Accept-Ranges: bytes
< Content-Length: 3
< Content-Type: text/html; charset=UTF-8
<
Q2
* Connection #0 to host 192.168.200.1 left intact
[root@server1 ~]# curl -v http://127.0.0.1/exam/q2/
*   Trying 127.0.0.1:80...
* Connected to 127.0.0.1 (127.0.0.1) port 80 (#0)
> GET /exam/q2/ HTTP/1.1
> Host: 127.0.0.1
> User-Agent: curl/7.76.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 403 Forbidden
< Date: Mon, 28 Apr 2025 16:24:58 GMT
< Server: Apache/2.4.62 (AlmaLinux)
< Content-Length: 199
< Content-Type: text/html; charset=iso-8859-1
<
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
</body></html>
* Connection #0 to host 127.0.0.1 left intact
[root@server1 ~]# vim /etc/httpd/conf/httpd.conf
[root@server1 ~]# 
```



Forbidden

You don't have permission to access this resource.



Q2

3 TASK 3

1. Authentication and Secure Server:

   ☑ Create the **secure_exam** directory inside **/var/www/exam** and configure Apache so that **only** the user **roger** using the IP address **192.168.200.1**, and the password **exam**, can access it. **All other users must be denied access**.

   ☑ Create a **secure_exam2** directory inside **/var/www/exam** and place an **.htaccess** file in it to define that only the user **roger** can access the web pages in this directory **or** just using the IP address **192.168.200.1.**

Make directory

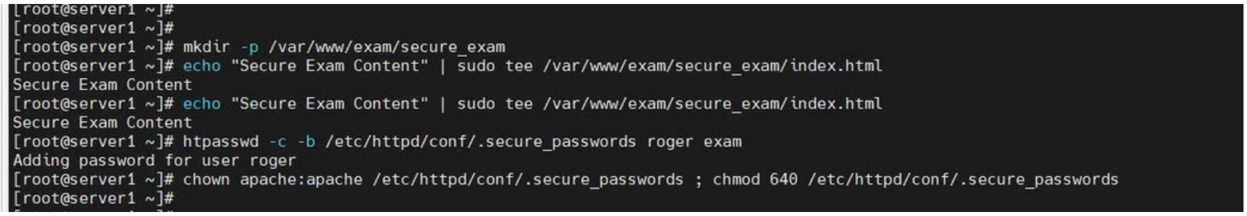
```
mkdir -p /var/www/exam/secure_exam
```

Create index file

echo "Secure Exam Content" | sudo tee /var/www/exam/secure_exam/index.html

Create password file and add user

```
htpasswd -c -b /etc/httpd/conf/.secure_passwords roger exam
chown apache:apache /etc/httpd/conf/.secure_passwords ; chmod 640 /etc/httpd/conf/.secure_passwords
```



```
[root@server1 ~]#
[root@server1 ~]#
[root@server1 ~]# mkdir -p /var/www/exam/secure_exam
[root@server1 ~]# echo "Secure Exam Content" | sudo tee /var/www/exam/secure_exam/index.html
Secure Exam Content
[root@server1 ~]# echo "Secure Exam Content" | sudo tee /var/www/exam/secure_exam/index.html
Secure Exam Content
[root@server1 ~]# htpasswd -c -b /etc/httpd/conf/.secure_passwords roger exam
Adding password for user roger
[root@server1 ~]# chown apache:apache /etc/httpd/conf/.secure_passwords ; chmod 640 /etc/httpd/conf/.secure_passwords
[root@server1 ~]#
```

Modify Apache configuration

Configure /var/www/exam/secure_exam2 (.htaccess Method)

Create directory and test file

mkdir -p /var/www/exam/secure_exam2

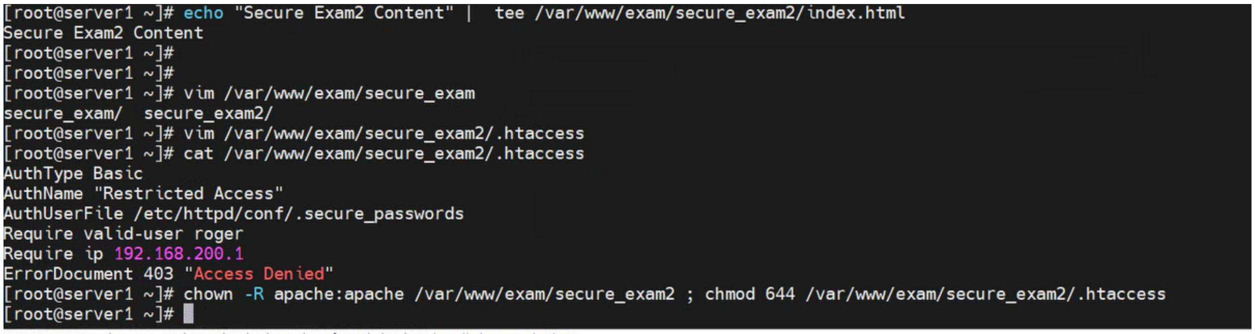
echo "Secure Exam2 Content" | tee /var/www/exam/secure_exam2/index.html

Create .htaccess file

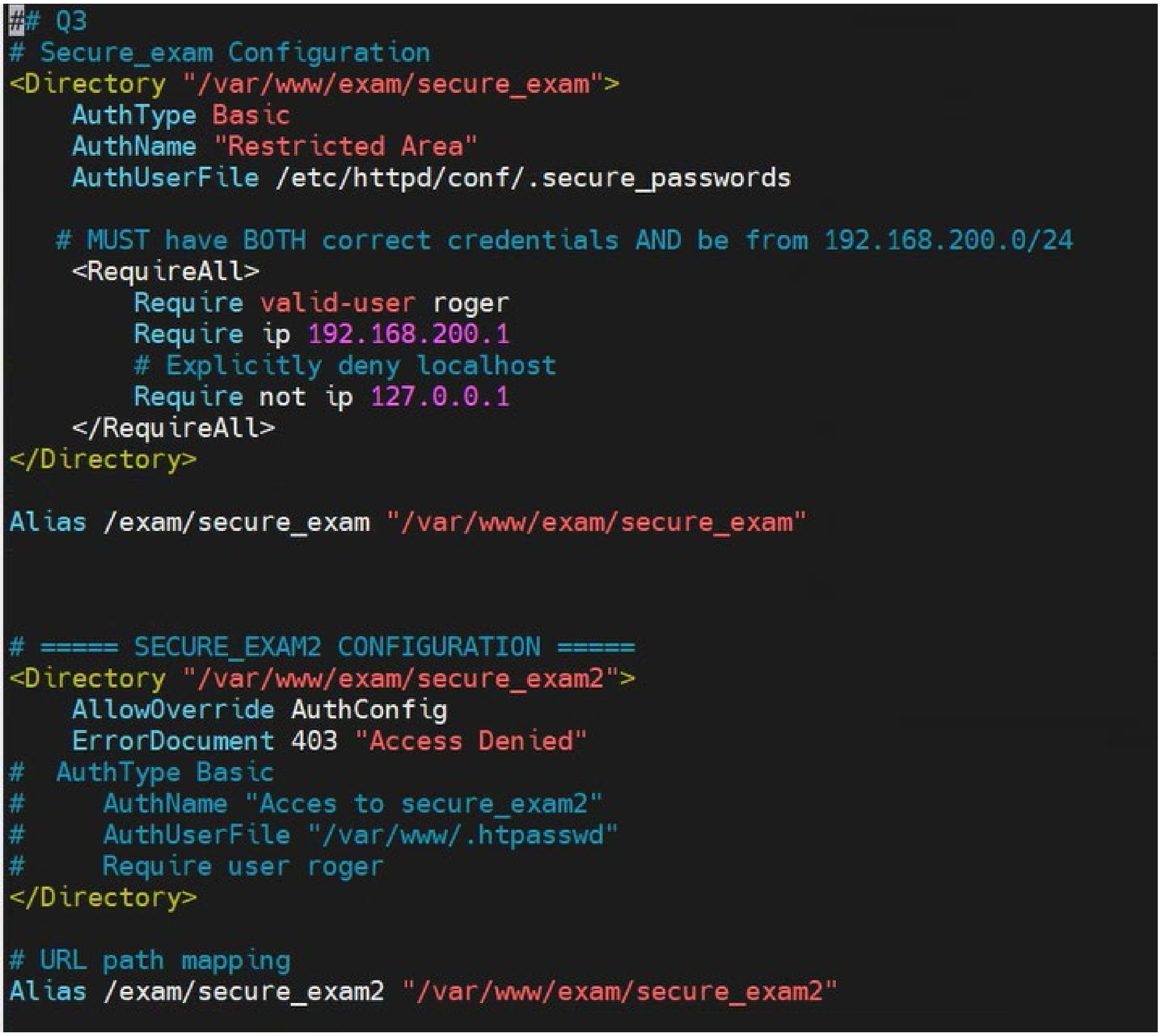
```
 tee /var/www/exam/secure_exam2/.htaccess <<'EOF'
AuthType Basic
AuthName "Restricted Access"
AuthUserFile /etc/httpd/conf/.secure_passwords
Require valid-user roger
Require ip 192.168.200.1
ErrorDocument 403 "Access Denied"
EOF
```

Set permissions

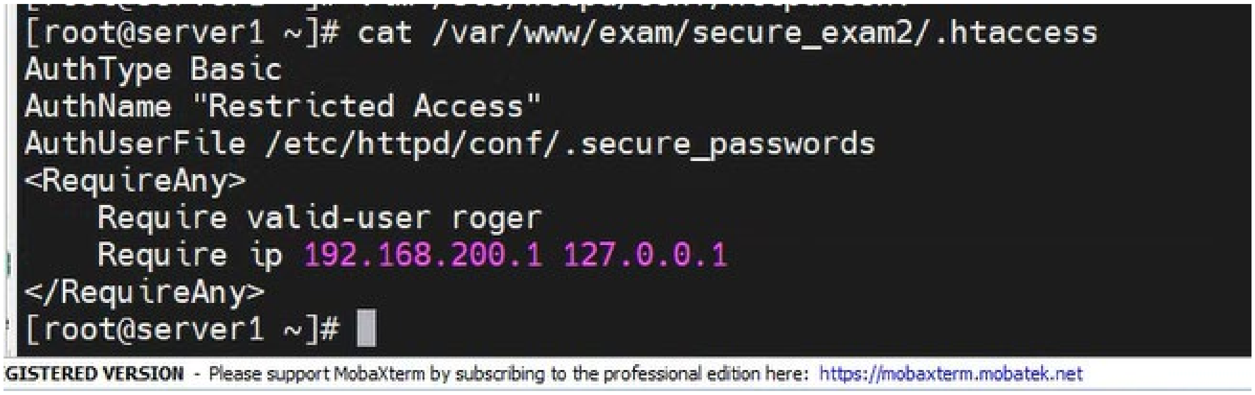
 chown -R apache:apache /var/www/exam/secure_exam2 ; chmod 644 /var/www/exam/secure_exam2/.htaccess



```
[root@server1 ~]# echo "Secure Exam2 Content" |  tee /var/www/exam/secure_exam2/index.html
Secure Exam2 Content
[root@server1 ~]#
[root@server1 ~]#
[root@server1 ~]# vim /var/www/exam/secure_exam
secure_exam/  secure_exam2/
[root@server1 ~]# vim /var/www/exam/secure_exam2/.htaccess
[root@server1 ~]# cat /var/www/exam/secure_exam2/.htaccess
AuthType Basic
AuthName "Restricted Access"
AuthUserFile /etc/httpd/conf/.secure_passwords
Require valid-user roger
Require ip 192.168.200.1
ErrorDocument 403 "Access Denied"
[root@server1 ~]# chown -R apache:apache /var/www/exam/secure_exam2 ; chmod 644 /var/www/exam/secure_exam2/.htaccess
[root@server1 ~]#
```



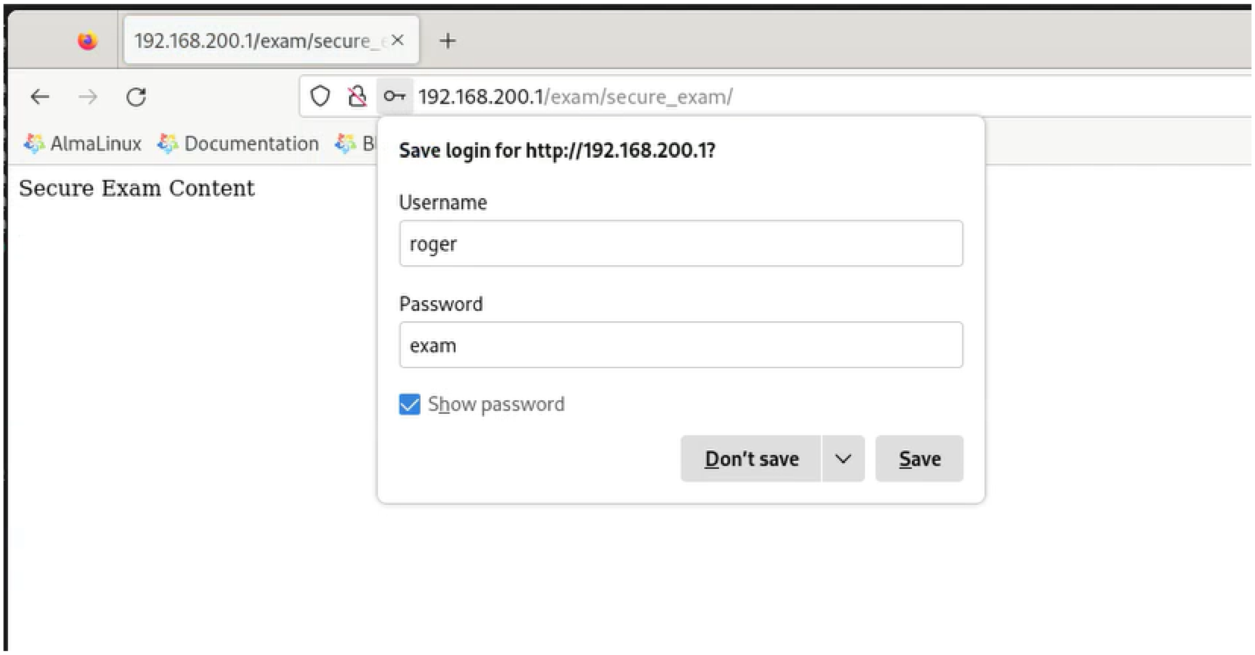
```
## Q3
# Secure_exam Configuration
<Directory "/var/www/exam/secure_exam">
    AuthType Basic
    AuthName "Restricted Area"
    AuthUserFile /etc/httpd/conf/.secure_passwords

   # MUST have BOTH correct credentials AND be from 192.168.200.0/24
    <RequireAll>
        Require valid-user roger
        Require ip 192.168.200.1
        # Explicitly deny localhost
        Require not ip 127.0.0.1
    </RequireAll>
</Directory>

Alias /exam/secure_exam "/var/www/exam/secure_exam"



# ===== SECURE_EXAM2 CONFIGURATION =====
<Directory "/var/www/exam/secure_exam2">
    AllowOverride AuthConfig
    ErrorDocument 403 "Access Denied"
#  AuthType Basic
#     AuthName "Acces to secure_exam2"
#     AuthUserFile "/var/www/.htpasswd"
#     Require user roger
</Directory>

# URL path mapping
Alias /exam/secure_exam2 "/var/www/exam/secure_exam2"
```



```
[root@server1 ~]# cat /var/www/exam/secure_exam2/.htaccess
AuthType Basic
AuthName "Restricted Access"
AuthUserFile /etc/httpd/conf/.secure_passwords
<RequireAny>
    Require valid-user roger
    Require ip 192.168.200.1 127.0.0.1
</RequireAny>
[root@server1 ~]#
```

GISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

Test



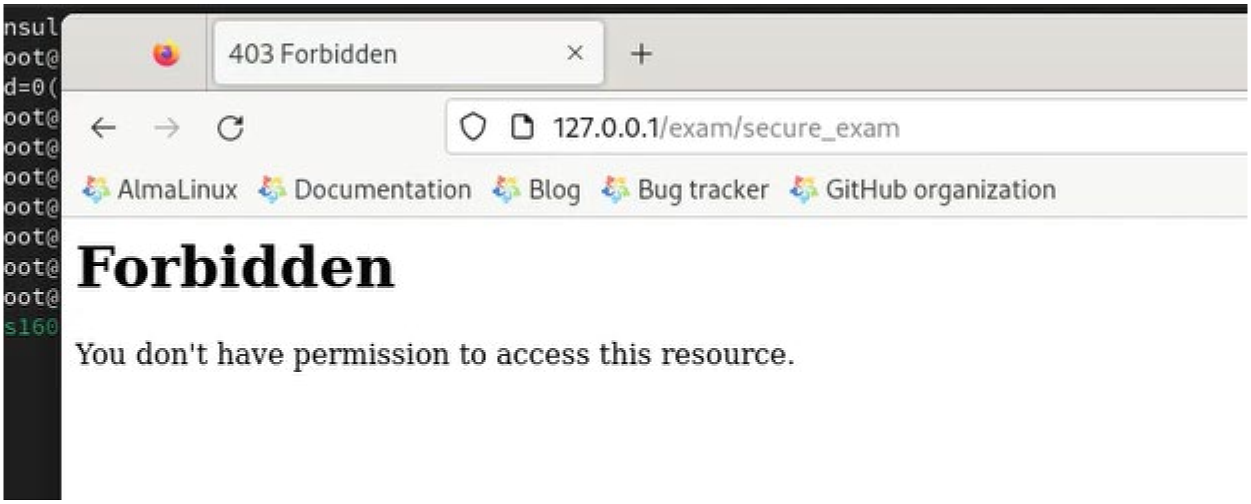
Secure Exam Content

Save login for http://192.168.200.1?

Username: roger

Password: exam

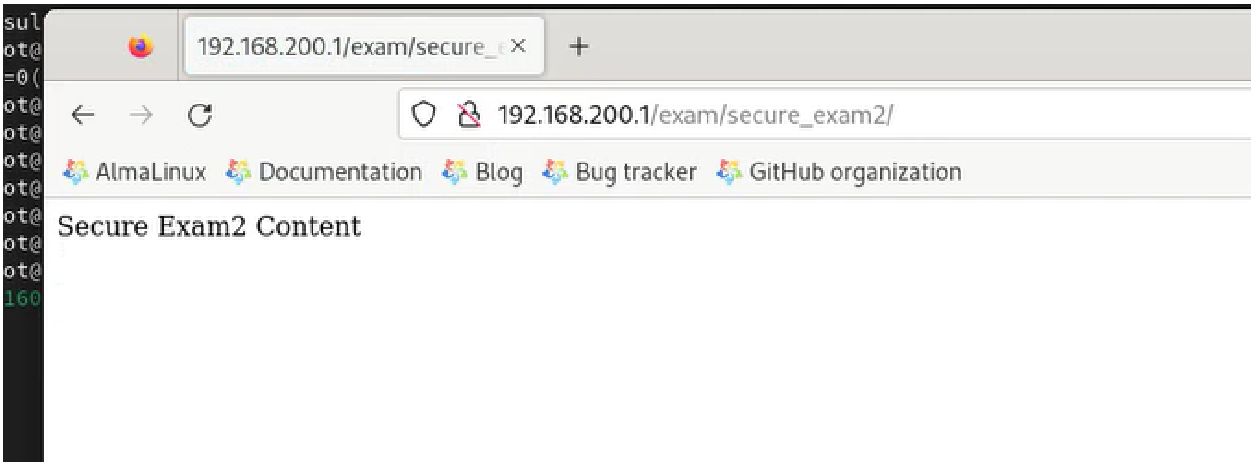
☑ Show password



127.0.0.1/exam/secure_exam

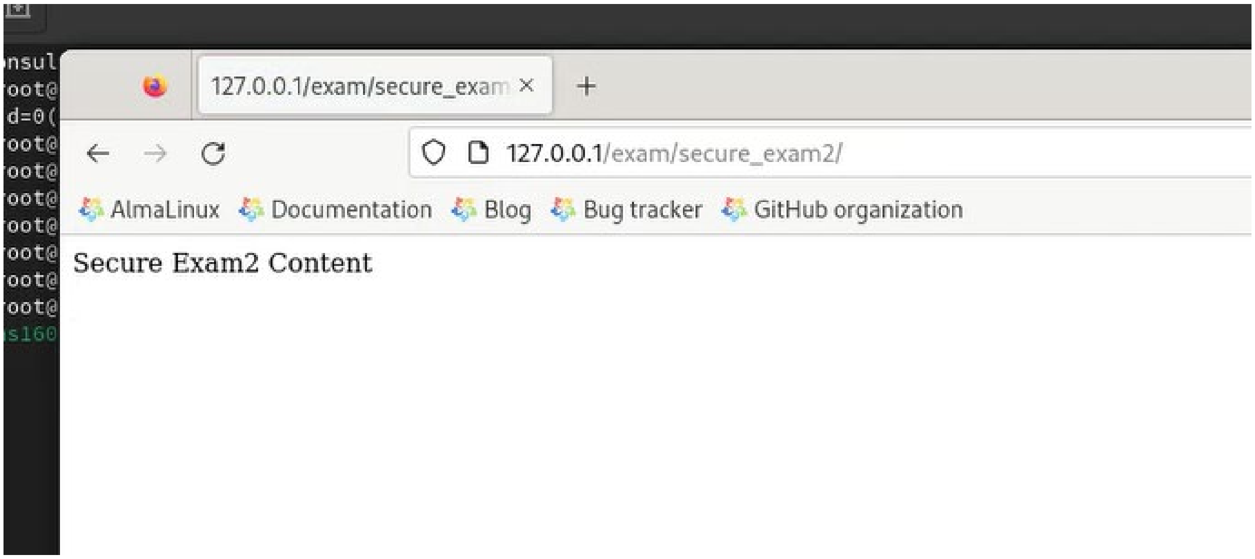
Forbidden

You don't have permission to access this resource.



192.168.200.1/exam/secure_exam2/

Secure Exam2 Content



Secure Exam2 Content

4 Task 4

4.1 Requirements

DNS and Virtual Hosts:

☑ Create **name-based** and **port-based** virtual hosts using the hostnames listed below.

☑ Use the same IP address **192.168.200.1** for all virtual servers.

Note: Since these are virtual hosts, the basic configuration is the same for all. Each server must use its own directory to store its web pages. **DNS-based name resolution must be used for hostname resolution.**

☑ Hostname: **virtuel1.exam.com**

Port: **80**
Web directory**: /var/www/exam/q5/virtuel1_80**

☑ Hostname**: virtuel2.exam.com**

Port: **8000**
Web Directory: **/var/www/exam/q5/virtuel2_8000**

Create directories

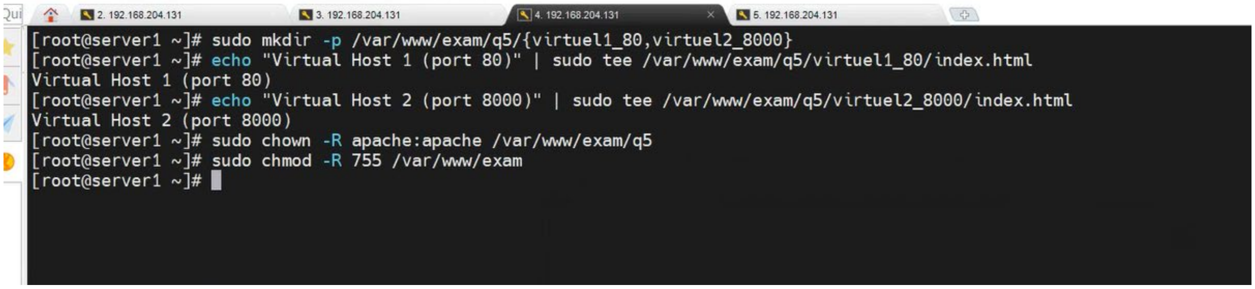
```
sudo mkdir -p /var/www/exam/q5/{virtuel1_80,virtuel2_8000}
```

Create index files

```
echo "Virtual Host 1 (port 80)" | sudo tee /var/www/exam/q5/virtuel1_80/index.html
echo "Virtual Host 2 (port 8000)" | sudo tee /var/www/exam/q5/virtuel2_8000/index.html
```

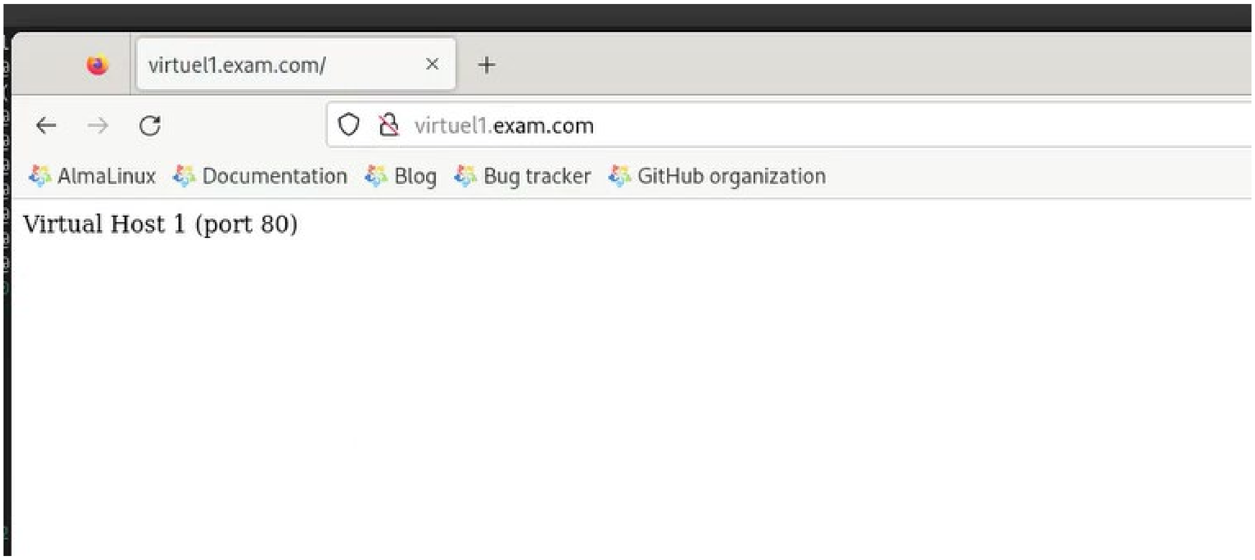
Set permissions

```
sudo chown -R apache:apache /var/www/exam/q5
sudo chmod -R 755 /var/www/exam
```

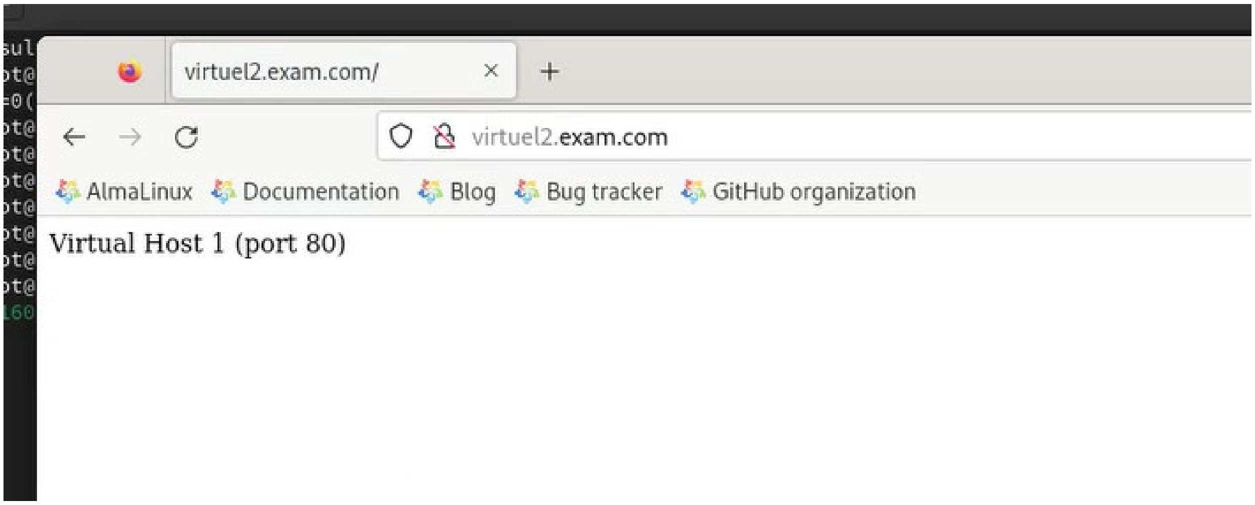


```
[root@server1 ~]# sudo mkdir -p /var/www/exam/q5/{virtuel1_80,virtuel2_8000}
[root@server1 ~]# echo "Virtual Host 1 (port 80)" | sudo tee /var/www/exam/q5/virtuel1_80/index.html
Virtual Host 1 (port 80)
[root@server1 ~]# echo "Virtual Host 2 (port 8000)" | sudo tee /var/www/exam/q5/virtuel2_8000/index.html
Virtual Host 2 (port 8000)
[root@server1 ~]# sudo chown -R apache:apache /var/www/exam/q5
[root@server1 ~]# sudo chmod -R 755 /var/www/exam
[root@server1 ~]# 
```

Test



Virtual Host 1 (port 80)



Virtual Host 1 (port 80)