Name: Tushar Panchal

En.No: 21162101014

Sub: ITIM (IT Infrastructure & Management)

Branch: CBA

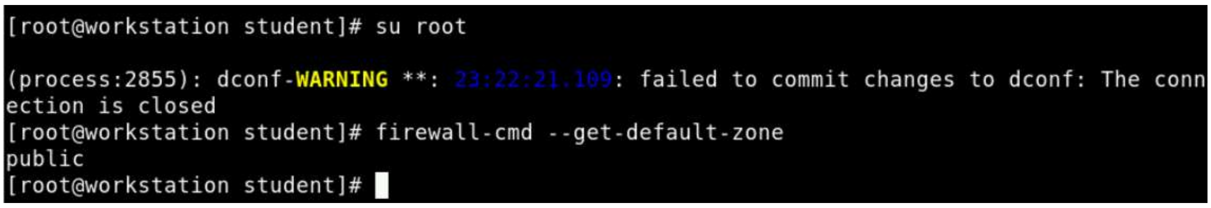
Batch:61

----------------------------PRACTICAL 09----------------------------

Managing Network Security

AIM : To enhance the security of the device and control the activities on it perform the below mentioned task.

1. Find out your device is configured in which zone. Provide the default zone of your machine :

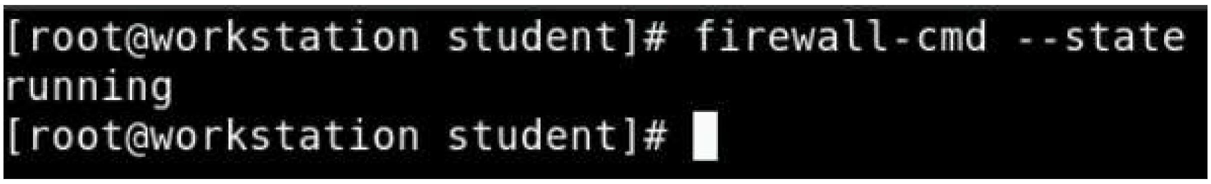


```
[root@workstation student]# su root

(process:2855): dconf-WARNING **: 23:22:21.109: failed to commit changes to dconf: The conn
ection is closed
[root@workstation student]# firewall-cmd --get-default-zone
public
[root@workstation student]# 
```

command : `firewall-cmd --get-default-zone`

The command **firewall-cmd --get-default-zone** is used in Linux systems to retrieve the default firewall zone that is currently set on the system. This command is specific to firewalld, which is a dynamic firewall manager that manages firewall rules in Linux.

2. Find out the status of the firewall service and in which mode it is?



```
[root@workstation student]# firewall-cmd --state
running
[root@workstation student]#
```

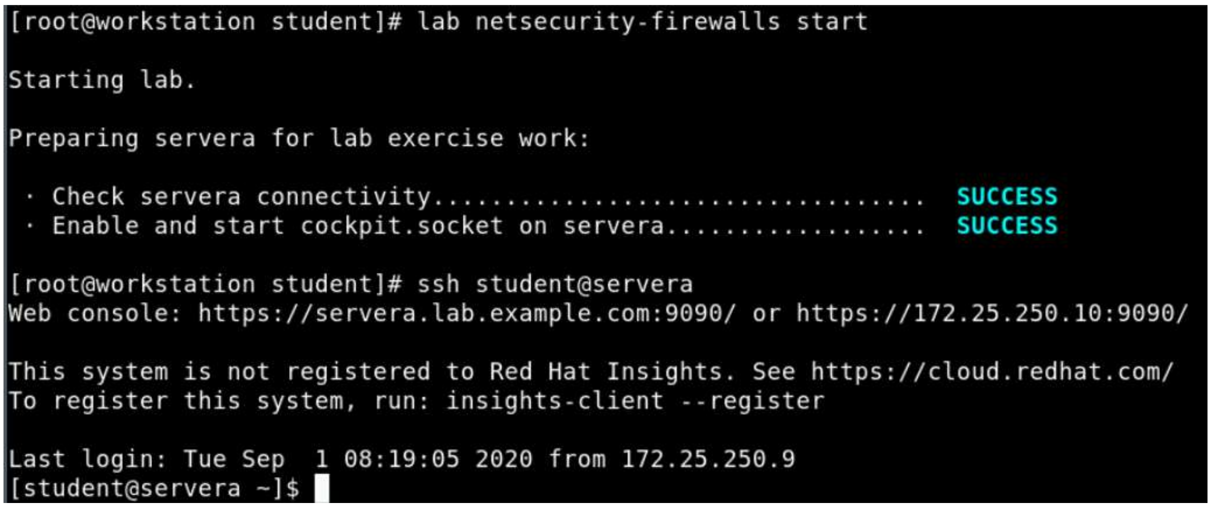
command : `firewall-cmd --state`

When you run **firewall-cmd --state**, it returns the current state of the firewalld service. The possible states are typically:

- **running**: Indicates that the firewalld service is actively running.
- **not running**: Indicates that the firewalld service is not running.

3. Install httpd and mod_ssl packages. These packages provide the Apache web server you will protect with a firewall, and the necessary extensions for the web server to serve content over SSL. (Guided Exercise) :

» **Log into servera :**

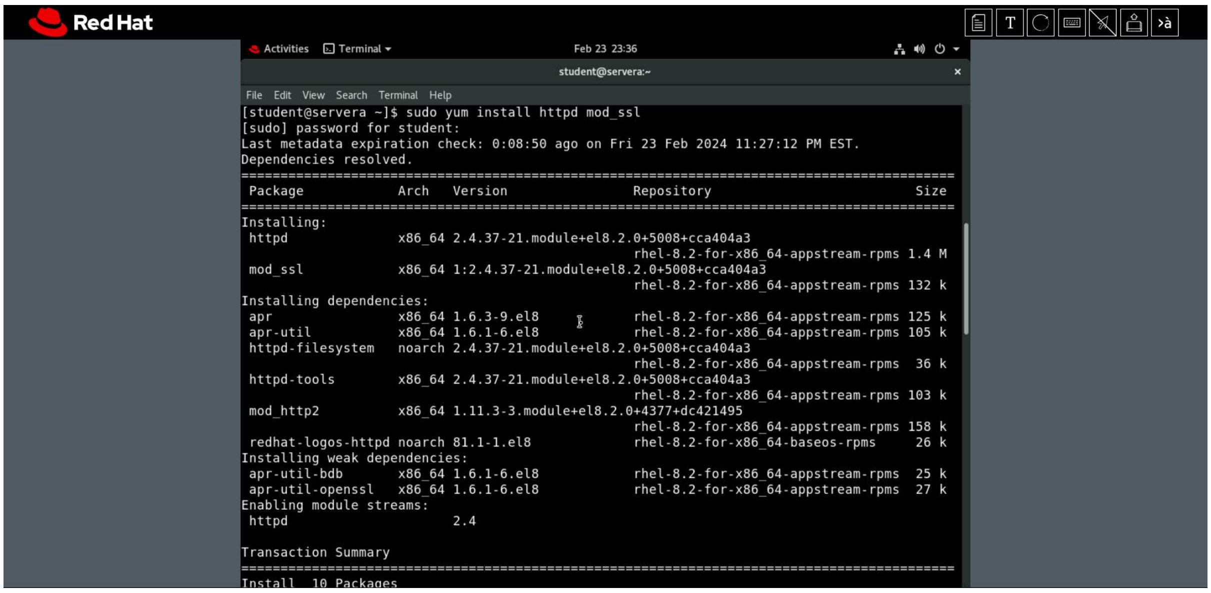


```
[root@workstation student]# lab netsecurity-firewalls start

Starting lab.

Preparing servera for lab exercise work:

 · Check servera connectivity.................................  SUCCESS
 · Enable and start cockpit.socket on servera.................  SUCCESS

[root@workstation student]# ssh student@servera
Web console: https://servera.lab.example.com:9090/ or https://172.25.250.10:9090/

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Tue Sep  1 08:19:05 2020 from 172.25.250.9
[student@servera ~]$
```

Commands :

```
lab netsecurity-firewalls start
ssh student@servera
```

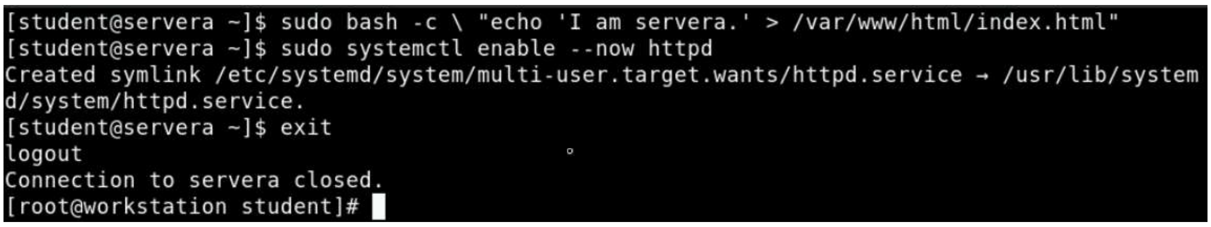
» Now install httpd and mod_ssl packages :



```
[student@servera ~]$ sudo yum install httpd mod_ssl
[sudo] password for student:
Last metadata expiration check: 0:08:50 ago on Fri 23 Feb 2024 11:27:12 PM EST.
Dependencies resolved.
================================================================================
 Package              Arch   Version                        Repository     Size
================================================================================
Installing:
 httpd                x86_64 2.4.37-21.module+el8.2.0+5008+cca404a3
                                                rhel-8.2-for-x86_64-appstream-rpms 1.4 M
 mod_ssl              x86_64 1:2.4.37-21.module+el8.2.0+5008+cca404a3
                                                rhel-8.2-for-x86_64-appstream-rpms 132 k
Installing dependencies:
 apr                  x86_64 1.6.3-9.el8        rhel-8.2-for-x86_64-appstream-rpms 125 k
 apr-util             x86_64 1.6.1-6.el8        rhel-8.2-for-x86_64-appstream-rpms 105 k
 httpd-filesystem     noarch 2.4.37-21.module+el8.2.0+5008+cca404a3
                                                rhel-8.2-for-x86_64-appstream-rpms  36 k
 httpd-tools          x86_64 2.4.37-21.module+el8.2.0+5008+cca404a3
                                                rhel-8.2-for-x86_64-appstream-rpms 103 k
 mod_http2            x86_64 1.11.3-3.module+el8.2.0+4377+dc421495
                                                rhel-8.2-for-x86_64-appstream-rpms 158 k
 redhat-logos-httpd   noarch 81.1-1.el8         rhel-8.2-for-x86_64-baseos-rpms     26 k
Installing weak dependencies:
 apr-util-bdb         x86_64 1.6.1-6.el8        rhel-8.2-for-x86_64-appstream-rpms  25 k
 apr-util-openssl     x86_64 1.6.1-6.el8        rhel-8.2-for-x86_64-appstream-rpms  27 k
Enabling module streams:
 httpd                       2.4

Transaction Summary
================================================================================
Install  10 Packages
```

command : `sudo yum install httpd mod_ssl`

» create the /var/www/html/index.html file. Add one line of text that reads: I am servera. Start and enable the httpd service on your servera system :



```
[student@servera ~]$ sudo bash -c \ "echo 'I am servera.' > /var/www/html/index.html"
[student@servera ~]$ sudo systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/system
d/system/httpd.service.
[student@servera ~]$ exit
logout
Connection to servera closed.
[root@workstation student]#
```

After Exit from servera

command :

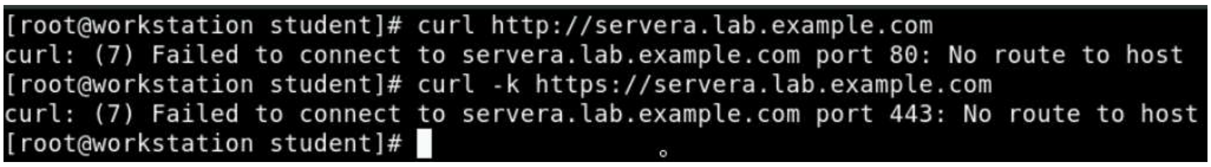
```
sudo bash -c \ "echo 'I am servera.' > /var/www/html/index.html"
sudo systemctl enable --now httpd
```

» attempt to access your web server on servera using both the cleartext port 80/TCP and the SSL encapsulated port 443/TCP. Both attempts should fail :

This command should fail : `curl http://servera.lab.example.com`

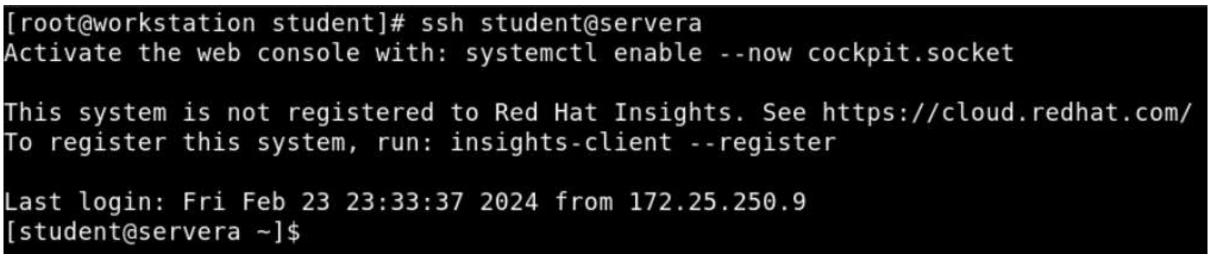
This command should also fail :

```
curl -k https://servera.lab.example.com
```



```
[root@workstation student]# curl http://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 80: No route to host
[root@workstation student]# curl -k https://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 443: No route to host
[root@workstation student]#
```

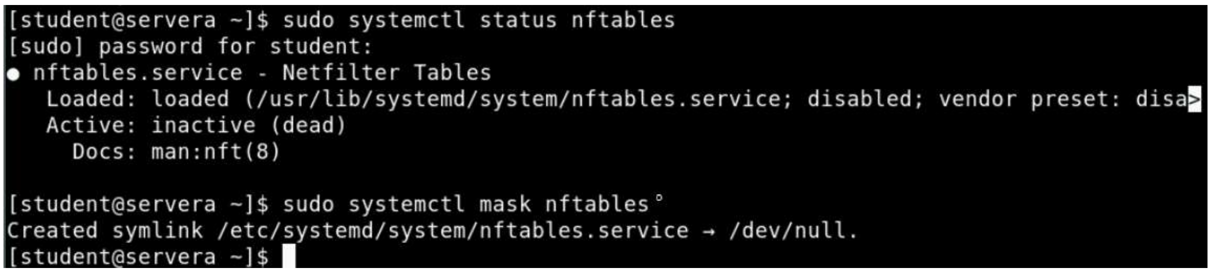
Log in to servera as the student user :



```
[root@workstation student]# ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Fri Feb 23 23:33:37 2024 from 172.25.250.9
[student@servera ~]$
```

command : `ssh student@servera`

On servera, make sure that the nftables service is masked and the firewalld service is enabled and running. If it is not running make it run :



```
[student@servera ~]$ sudo systemctl status nftables
[sudo] password for student:
● nftables.service - Netfilter Tables
   Loaded: loaded (/usr/lib/systemd/system/nftables.service; disabled; vendor preset: disa>
   Active: inactive (dead)
     Docs: man:nft(8)

[student@servera ~]$ sudo systemctl mask nftables
Created symlink /etc/systemd/system/nftables.service → /dev/null.
[student@servera ~]$
```

commands :

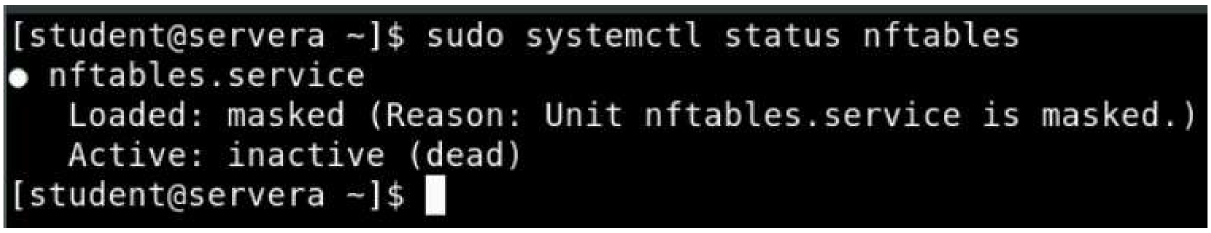
```
sudo systemctl status nftables
```

This command is used to check the current status of the nftables service.

```
sudo systemctl mask nftables
```

This command is used to mask (disable) the nftables service.

Verify that the status of the nftables service is masked :



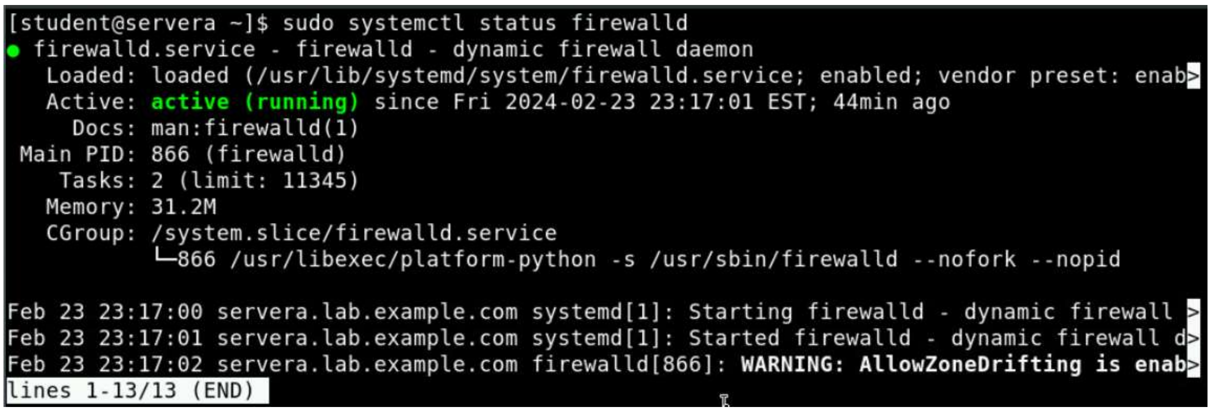
```
[student@servera ~]$ sudo systemctl status nftables
● nftables.service
   Loaded: masked (Reason: Unit nftables.service is masked.)
   Active: inactive (dead)
[student@servera ~]$
```

commands :

`sudo systemctl status nftables`

This command is used to check the current status of the nftables service.

Verify that the status of the firewalld service is enabled and running :



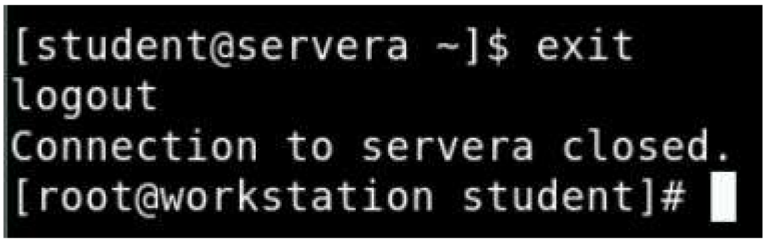
```
[student@servera ~]$ sudo systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enab>
   Active: active (running) since Fri 2024-02-23 23:17:01 EST; 44min ago
     Docs: man:firewalld(1)
 Main PID: 866 (firewalld)
    Tasks: 2 (limit: 11345)
   Memory: 31.2M
   CGroup: /system.slice/firewalld.service
           └─866 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid

Feb 23 23:17:00 servera.lab.example.com systemd[1]: Starting firewalld - dynamic firewall >
Feb 23 23:17:01 servera.lab.example.com systemd[1]: Started firewalld - dynamic firewall d>
Feb 23 23:17:02 servera.lab.example.com firewalld[866]: WARNING: AllowZoneDrifting is enab>
lines 1-13/13 (END)
```

commands :

`sudo systemctl status firewalld`

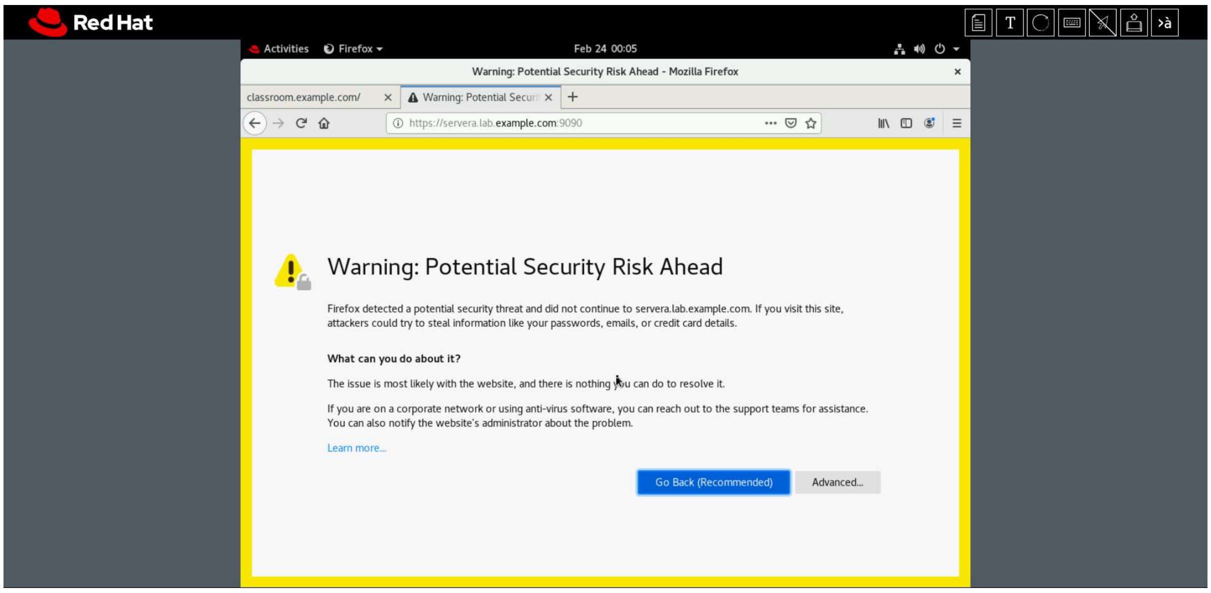
This command is used to check the status of the firewalld service on a Linux system

Exit from servera :



```
[student@servera ~]$ exit
logout
Connection to servera closed.
[root@workstation student]#
```

- **Open Firefox and browse to https://servera.lab.example.com:9090 to access the Web Console. Accept the self-signed certificate used by servera by adding an exception:**


Red Hat
Activities
Firefox
Feb 24 00:05
Warning: Potential Security Risk Ahead - Mozilla Firefox
classroom.example.com/
Warning: Potential Secur
https://servera.lab.example.com:9090

Warning: Potential Security Risk Ahead

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

What can you do about it?

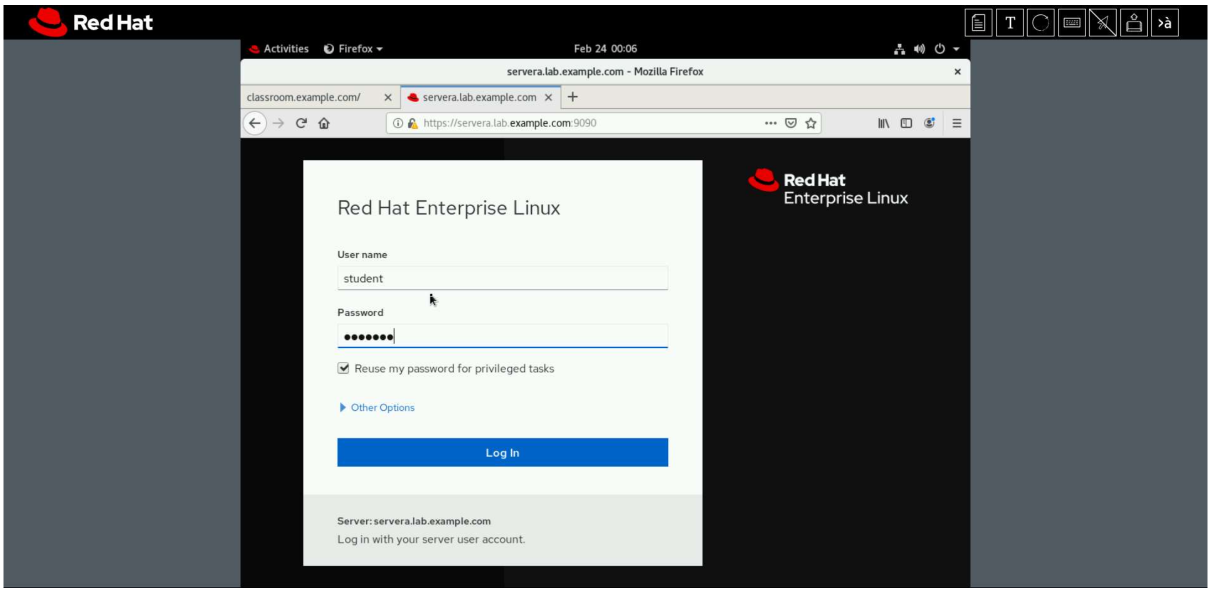
The issue is most likely with the website, and there is nothing you can do to resolve it.

If you are on a corporate network or using anti-virus software, you can reach out to the support teams for assistance. You can also notify the website's administrator about the problem.

Learn more...

Go Back (Recommended) | Advanced...

- **Log in as student user with student as the password :**



Red Hat Enterprise Linux

User name: student

Password: •••••••

☑ Reuse my password for privileged tasks

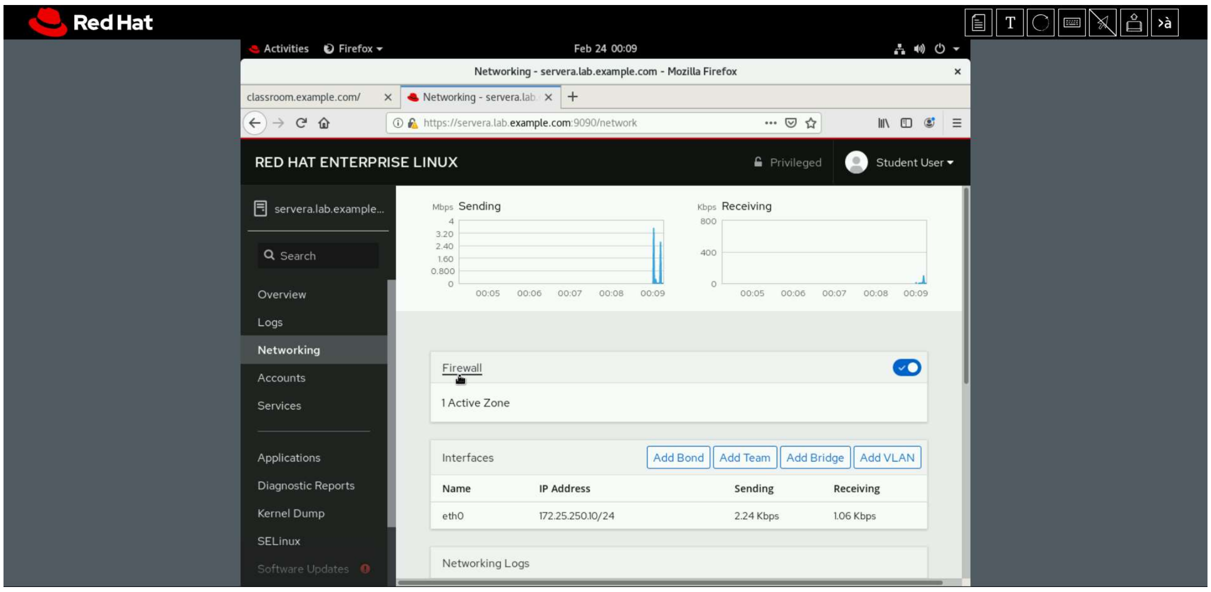
Other Options

Log In

Server: servera.lab.example.com

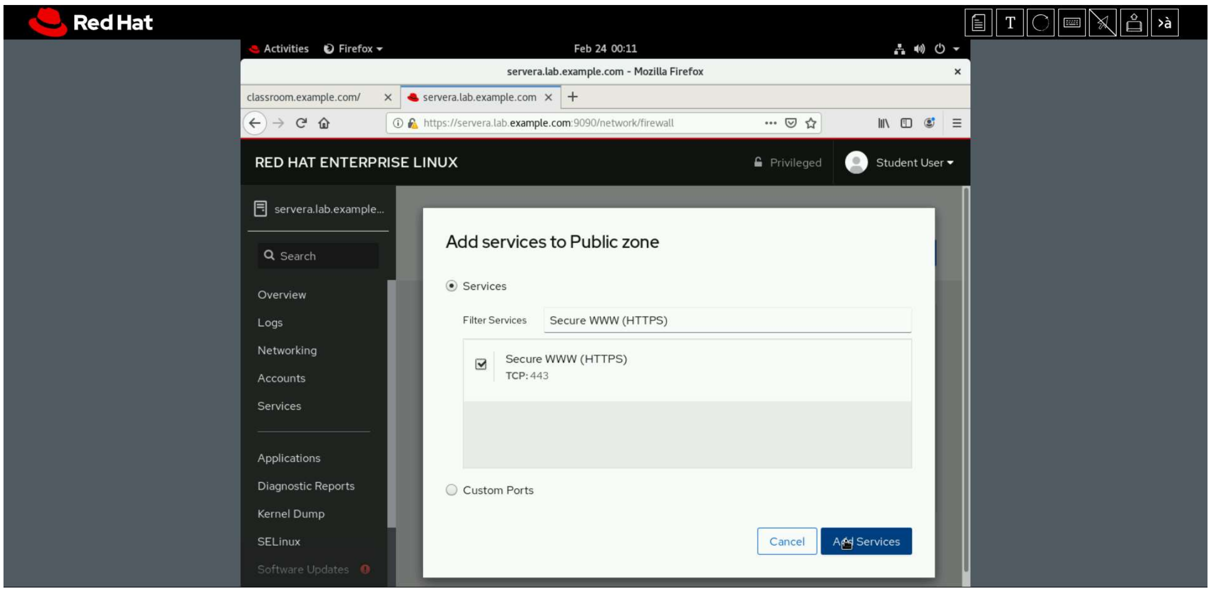
Log in with your server user account.

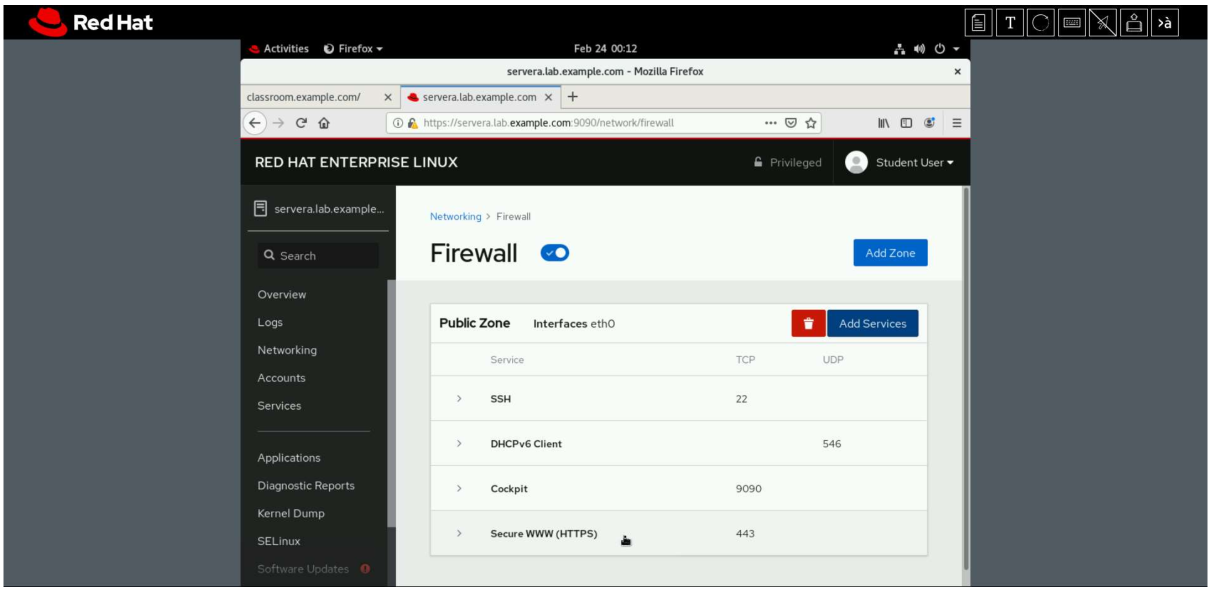
- **Click Networking in the left navigation bar.**
- **Click the Firewall link in main Networking page.**



Name	IP Address	Sending	Receiving
eth0	172.25.250.10/24	2.24 Kbps	1.06 Kbps

- **Click the Add Services... button located in the upper right side of the Firewall page.**
- **In the Add Services user interface, scroll down or use Filter Services to locate and select the check box next to the Secure WWW (HTTPS) service.**
- **Click the Add Services button located at the lower right side of the Add Services user interface.**





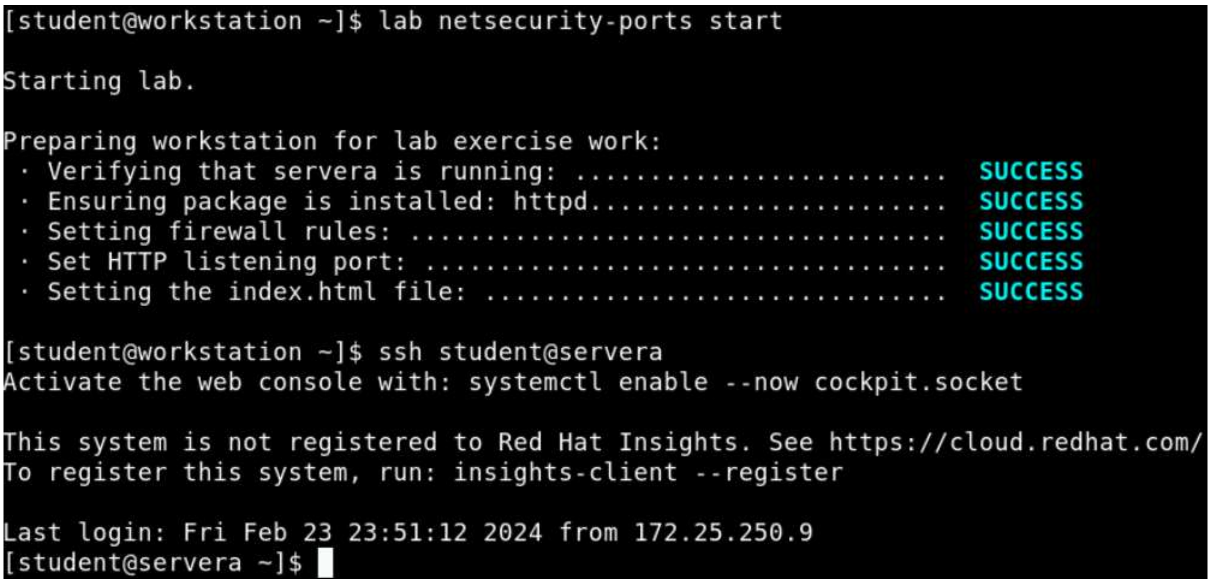
Firewall

Public Zone Interfaces eth0

Service	TCP	UDP
SSH	22	
DHCPv6 Client		546
Cockpit	9090	
Secure WWW (HTTPS)	443	

4. Your organization is deploying a new custom web application. The web application is running on a nonstandard port; in this case, 82/TCP. One of your junior administrators has already configured the application on your servera. However, the web server content is not accessible. (Guided Exercise) :

» Start netsecurity-ports & Log into servera :



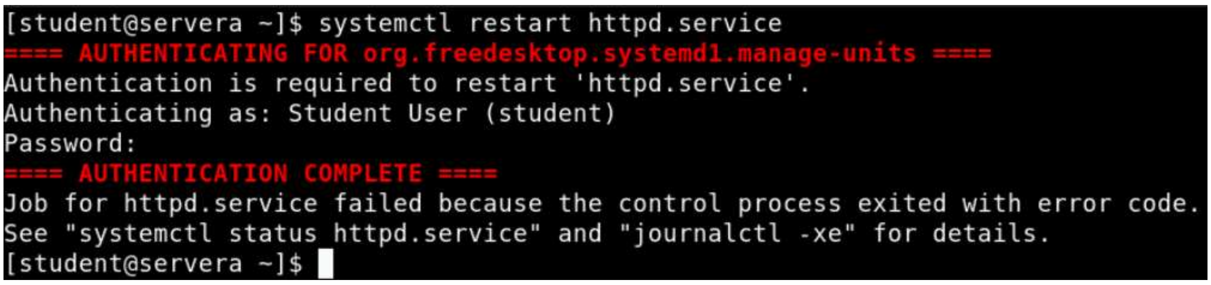
```
[student@workstation ~]$ lab netsecurity-ports start

Starting lab.

Preparing workstation for lab exercise work:
 · Verifying that servera is running: ........................  SUCCESS
 · Ensuring package is installed: httpd.......................  SUCCESS
 · Setting firewall rules: ...................................  SUCCESS
 · Set HTTP listening port: ..................................  SUCCESS
 · Setting the index.html file: ..............................  SUCCESS

[student@workstation ~]$ ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Fri Feb 23 23:51:12 2024 from 172.25.250.9
[student@servera ~]$ 
```

Commands :

```
lab netsecurity-ports start
ssh student@servera
```

» **According to question 4**

» **Attempt to fix the web content problem by restarting the httpd service :**



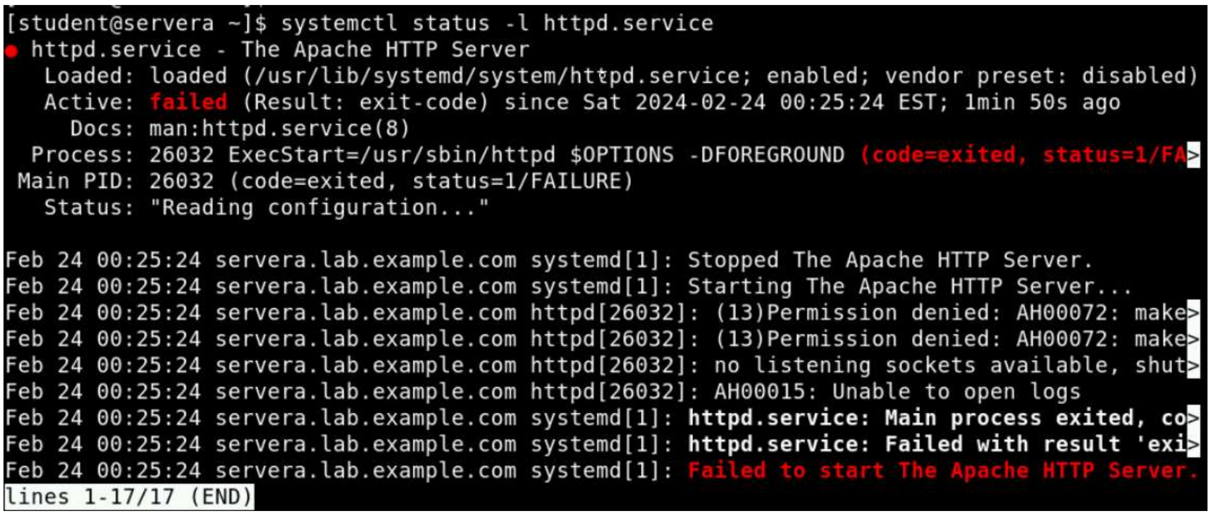
```
[student@servera ~]$ systemctl restart httpd.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'httpd.service'.
Authenticating as: Student User (student)
Password:
==== AUTHENTICATION COMPLETE ====
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xe" for details.
[student@servera ~]$
```

Command :

`systemctl restart httpd.service`

This command is used to restart the Apache HTTP Server service (**httpd**).

» **Check the status of the httpd service. Note the permission denied error :**



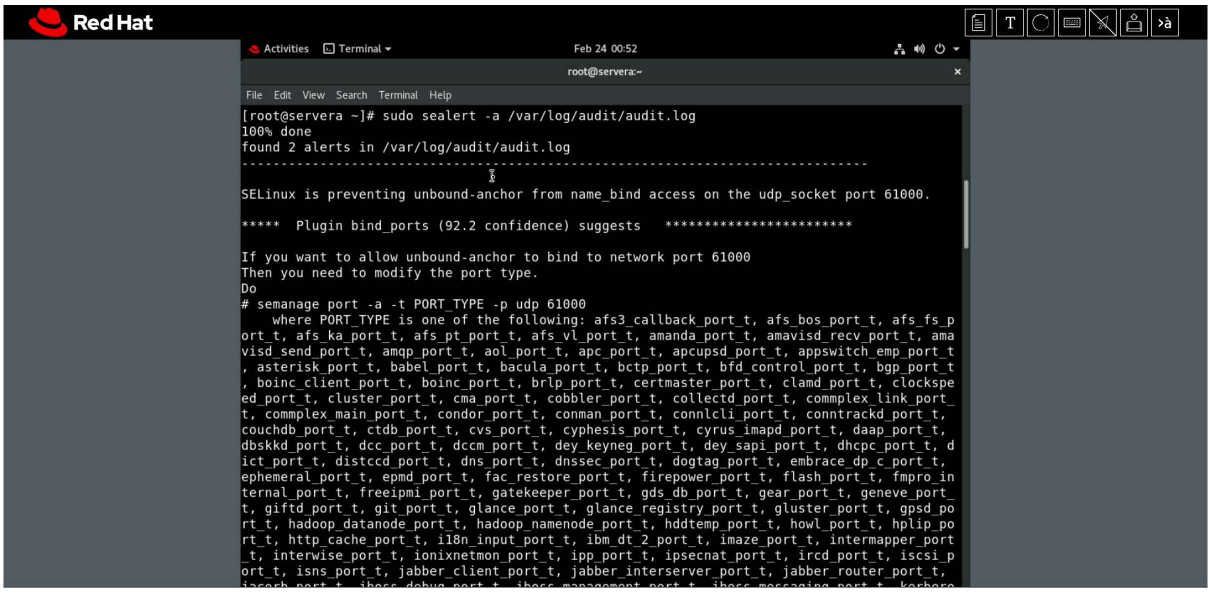
```
[student@servera ~]$ systemctl status -l httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Sat 2024-02-24 00:25:24 EST; 1min 50s ago
     Docs: man:httpd.service(8)
  Process: 26032 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FA>
 Main PID: 26032 (code=exited, status=1/FAILURE)
   Status: "Reading configuration..."

Feb 24 00:25:24 servera.lab.example.com systemd[1]: Stopped The Apache HTTP Server.
Feb 24 00:25:24 servera.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Feb 24 00:25:24 servera.lab.example.com httpd[26032]: (13)Permission denied: AH00072: make>
Feb 24 00:25:24 servera.lab.example.com httpd[26032]: (13)Permission denied: AH00072: make>
Feb 24 00:25:24 servera.lab.example.com httpd[26032]: no listening sockets available, shut>
Feb 24 00:25:24 servera.lab.example.com httpd[26032]: AH00015: Unable to open logs
Feb 24 00:25:24 servera.lab.example.com systemd[1]: httpd.service: Main process exited, co>
Feb 24 00:25:24 servera.lab.example.com systemd[1]: httpd.service: Failed with result 'exi>
Feb 24 00:25:24 servera.lab.example.com systemd[1]: Failed to start The Apache HTTP Server.
lines 1-17/17 (END)
```

Command :

`systemctl status -l httpd.service`

This command is used to check the detailed status of the Apache HTTP Server service (**httpd**).

» check if SELinux is blocking httpd from binding to port 82/TCP :



```
[root@servera ~]# sudo sealert -a /var/log/audit/audit.log
100% done
found 2 alerts in /var/log/audit/audit.log
--------------------------------------------------------------------------------

SELinux is preventing unbound-anchor from name_bind access on the udp_socket port 61000.

*****  Plugin bind_ports (92.2 confidence) suggests   ************************

If you want to allow unbound-anchor to bind to network port 61000
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p udp 61000
    where PORT_TYPE is one of the following: afs3_callback_port_t, afs_bos_port_t, afs_fs_p
ort_t, afs_ka_port_t, afs_pt_port_t, afs_vl_port_t, amanda_port_t, amavisd_recv_port_t, ama
visd_send_port_t, amqp_port_t, aol_port_t, apc_port_t, apcupsd_port_t, appswitch_emp_port_t
, asterisk_port_t, babel_port_t, bacula_port_t, bctp_port_t, bfd_control_port_t, bgp_port_t
, boinc_client_port_t, boinc_port_t, brlp_port_t, certmaster_port_t, clamd_port_t, clockspe
ed_port_t, cluster_port_t, cma_port_t, cobbler_port_t, collectd_port_t, commplex_link_port_
t, commplex_main_port_t, condor_port_t, conman_port_t, connlcli_port_t, conntrackd_port_t,
couchdb_port_t, ctdb_port_t, cvs_port_t, cyphesis_port_t, cyrus_imapd_port_t, daap_port_t,
dbskkd_port_t, dcc_port_t, dccm_port_t, dey_keyneg_port_t, dey_sapi_port_t, dhcpc_port_t, d
ict_port_t, distccd_port_t, dns_port_t, dnssec_port_t, dogtag_port_t, embrace_dp_c_port_t,
ephemeral_port_t, epmd_port_t, fac_restore_port_t, firepower_port_t, flash_port_t, fmpro_in
ternal_port_t, freeipmi_port_t, gatekeeper_port_t, gds_db_port_t, gear_port_t, geneve_port_
t, giftd_port_t, git_port_t, glance_port_t, glance_registry_port_t, gluster_port_t, gpsd_po
rt_t, hadoop_datanode_port_t, hadoop_namenode_port_t, hddtemp_port_t, howl_port_t, hplip_po
rt_t, http_cache_port_t, i18n_input_port_t, ibm_dt_2_port_t, imaze_port_t, intermapper_port
_t, interwise_port_t, ionixnetmon_port_t, ipp_port_t, ipsecnat_port_t, ircd_port_t, iscsi_p
ort_t, isns_port_t, jabber_client_port_t, jabber_interserver_port_t, jabber_router_port_t,
```

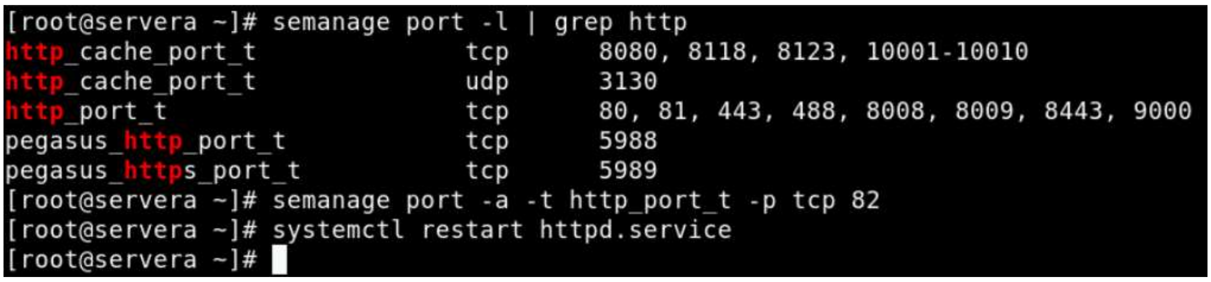
Command :

```
sudo sealert -a /var/log/audit/audit.log
```

» find an appropriate port type for port 82/TCP.

» assign port 82/TCP the http_port_t type.

» restart the httpd.service service.

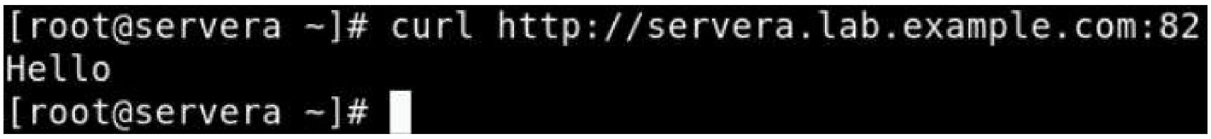


```
[root@servera ~]# semanage port -l | grep http
http_cache_port_t              tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989
[root@servera ~]# semanage port -a -t http_port_t -p tcp 82
[root@servera ~]# systemctl restart httpd.service
[root@servera ~]# 
```

Commands :

```
semanage port -l | grep http
semanage port -a -t http_port_t -p tcp 82
systemctl restart httpd.service
```

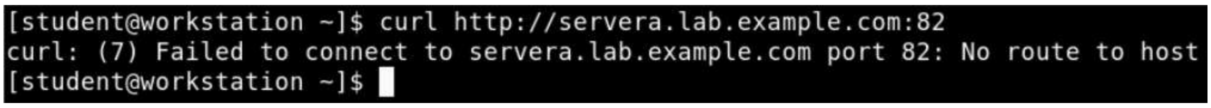
» **Check if you can now access the web server running on port 82/TCP :**


```
[root@servera ~]# curl http://servera.lab.example.com:82
Hello
[root@servera ~]#
```

Command :

```
curl http://servera.lab.example.com:82
```

» **In a different terminal window, check whether you can access the new web service from workstation :**

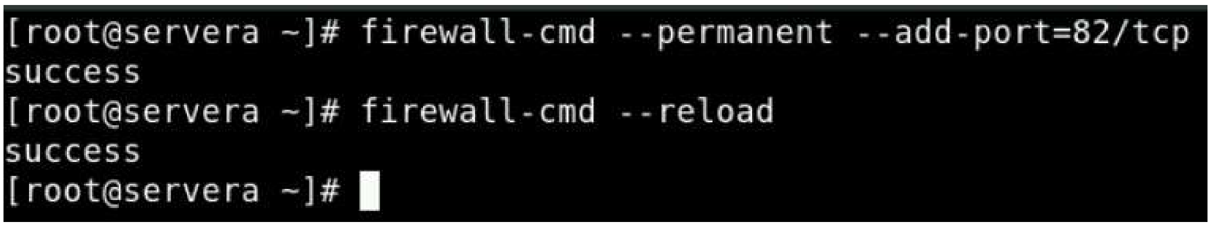

```
[student@workstation ~]$ curl http://servera.lab.example.com:82
curl: (7) Failed to connect to servera.lab.example.com port 82: No route to host
[student@workstation ~]$
```

That error means you still can not connect to the web service from workstation.

Command :

```
curl http://servera.lab.example.com:82
```

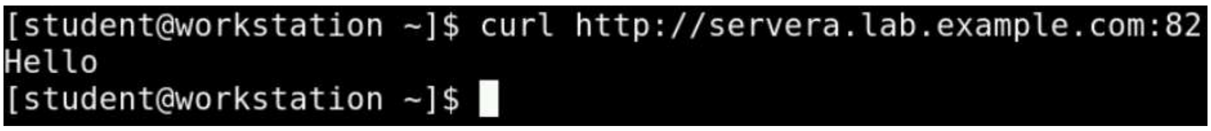
» **open port 82/TCP in the permanent configuration for the default zone on the firewall on servera, Activate your firewall changes on servera :**


```
[root@servera ~]# firewall-cmd --permanent --add-port=82/tcp
success
[root@servera ~]# firewall-cmd --reload
success
[root@servera ~]#
```

Commands :

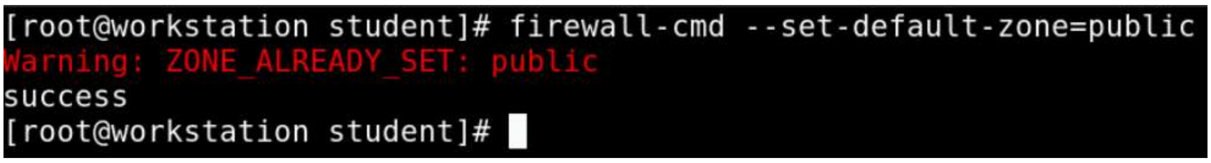
```
firewall-cmd --permanent --add-port=82/tcp
firewall-cmd --reload
```

» **Now access the web service from workstation :**


```
[student@workstation ~]$ curl http://servera.lab.example.com:82
Hello
[student@workstation ~]$
```

Command : `curl http://servera.lab.example.com:82`

5. Set your firewall into public zone.



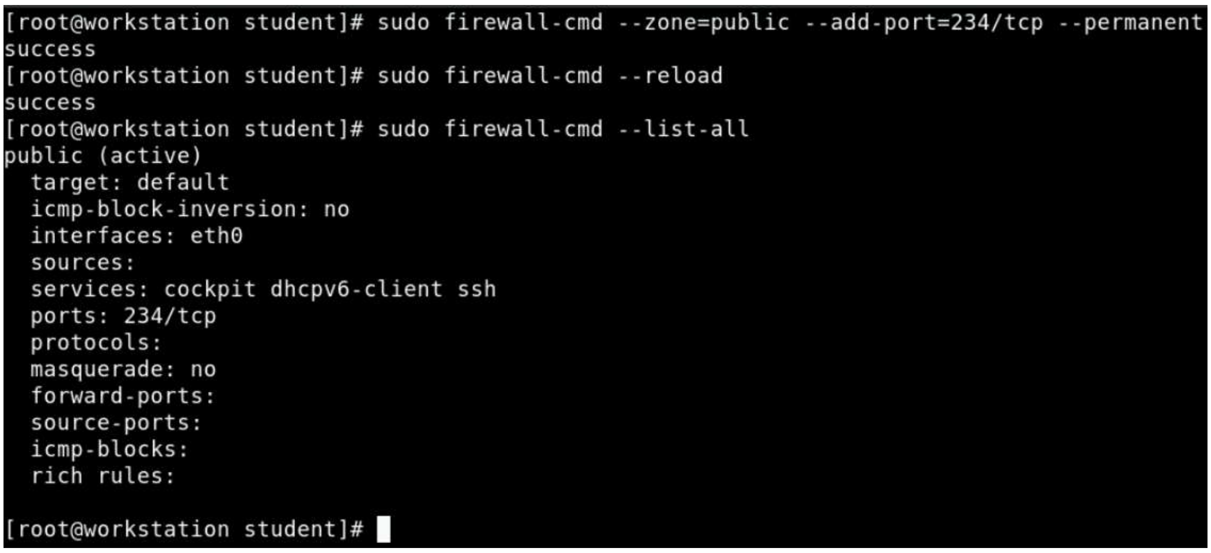
```
[root@workstation student]# firewall-cmd --set-default-zone=public
Warning: ZONE_ALREADY_SET: public
success
[root@workstation student]# 
```

Command :

```
firewall-cmd --set-default-zone=public
```

6. Allow the port 234, so the services running on this port is allowed in your network. After performing the configuration, demonstrate how to check the configuration.

- **First of all Allow the port 234 .**
- **Than reload the firewall to apply a new configuration persistently.**
- **Than check configuration.**



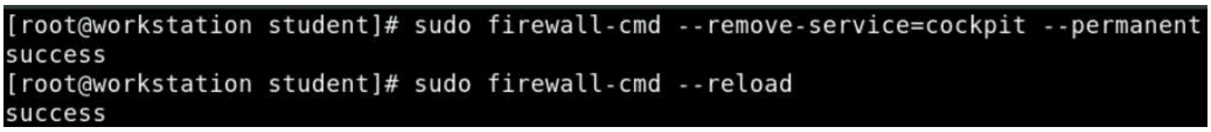
```
[root@workstation student]# sudo firewall-cmd --zone=public --add-port=234/tcp --permanent
success
[root@workstation student]# sudo firewall-cmd --reload
success
[root@workstation student]# sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 234/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@workstation student]# 
```

Commands :

```
sudo firewall-cmd --zone=public --add-port=234/tcp --permanent
sudo firewall-cmd --reload
sudo firewall-cmd --list-all
```

7. Demonstrate how to block a service, how a user can check that which services are blocked. Try to access the blocked services and let us know what type of error you will get.

- **First of all block a service .**
- **So i blocked cockpit service and make that configuration persistent.**

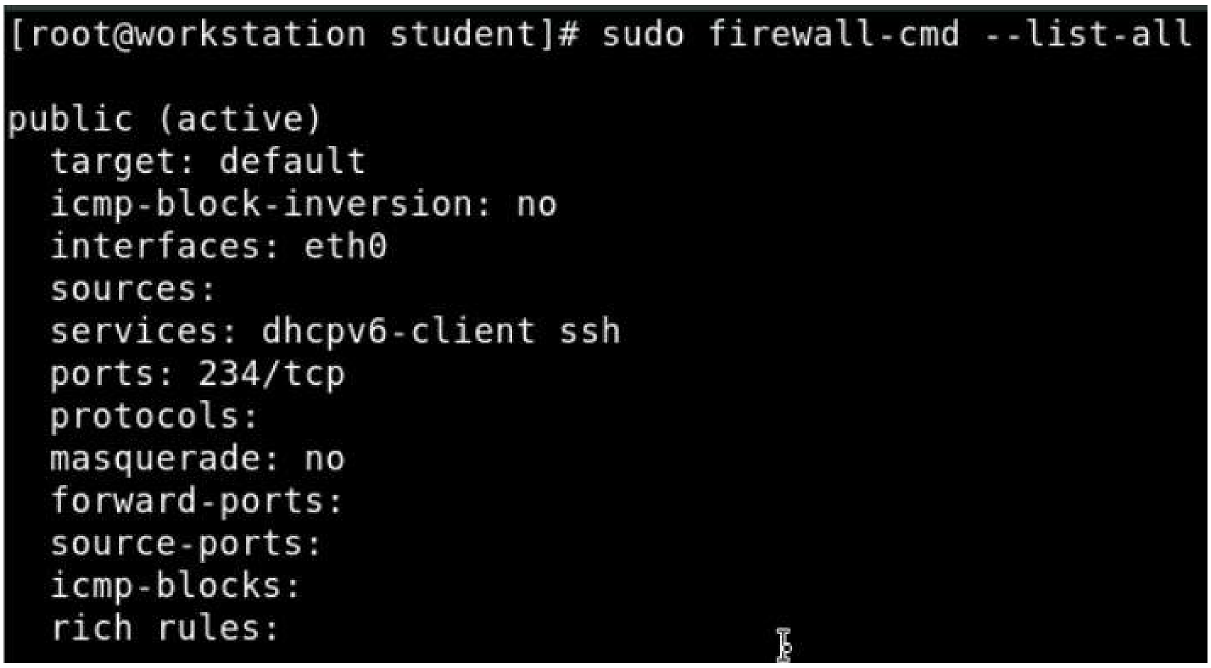


```
[root@workstation student]# sudo firewall-cmd --remove-service=cockpit --permanent
success
[root@workstation student]# sudo firewall-cmd --reload
success
```

Commands :

```
sudo firewall-cmd --remove-service=cockpit --permanent
sudo firewall-cmd --reload
```

- **Check Blocked Services.**

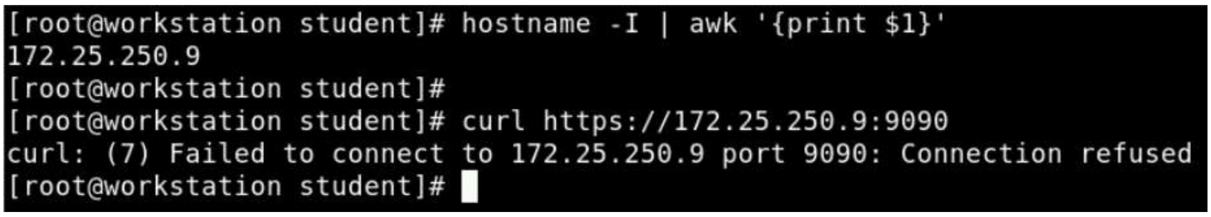


```
[root@workstation student]# sudo firewall-cmd --list-all

public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources:
  services: dhcpv6-client ssh
  ports: 234/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Command : `sudo firewall-cmd --list-all`

Now if i'll try to access cockpit service , so i'll send http request via curl.

First find ip address of your server. Then using curl send request



```
[root@workstation student]# hostname -I | awk '{print $1}'
172.25.250.9
[root@workstation student]#
[root@workstation student]# curl https://172.25.250.9:9090
curl: (7) Failed to connect to 172.25.250.9 port 9090: Connection refused
[root@workstation student]# 
```

Commands :

`hostname -I | awk '{print $1}'`

This command extracts the first IP address associated with the hostname.

`curl https://172.25.250.9:9090`

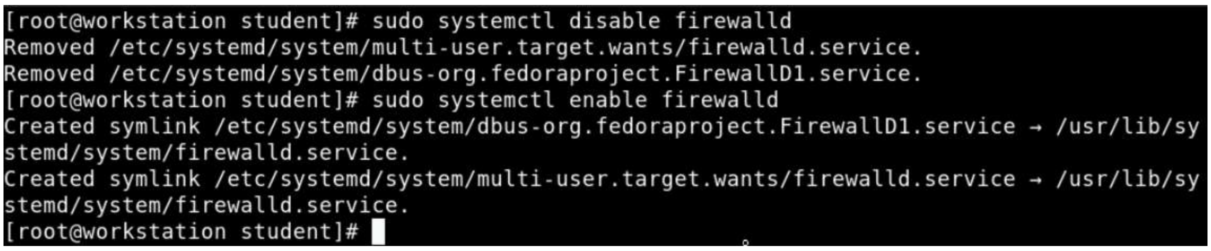
This command is attempting to make an HTTPS request to the server with the IP address 172.25.250.9 on port 9090.

If the service is blocked, you will likely encounter an error. The specific error message can vary, but common ones include:

- **Connection Refused:** This means that the connection to the service was actively rejected.
- **Connection Timed Out:** This indicates that the connection attempt took too long, possibly due to the firewall blocking the communication.

8. <u>Demonstrate how to disable the firewall, so that there will be no security check on the services as well as network traffic coming to your device.</u>

- **<u>First of all disable firewall .</u>**
- **<u>Then enable firewall again if you need to.</u>**



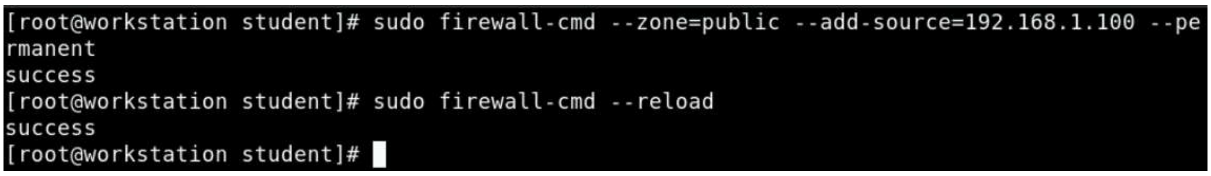
```
[root@workstation student]# sudo systemctl disable firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@workstation student]# sudo systemctl enable firewalld
Created symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service → /usr/lib/sy
stemd/system/firewalld.service.
Created symlink /etc/systemd/system/multi-user.target.wants/firewalld.service → /usr/lib/sy
stemd/system/firewalld.service.
[root@workstation student]#
```

<u>Commands :</u>

```
sudo systemctl disable firewalld
sudo systemctl enable firewalld
```

9. Demonstrate how to allow the traffic from a specific IP address.

» Suppose you want to allow traffic from the ip address 192.168.1.100 . after that reload it.



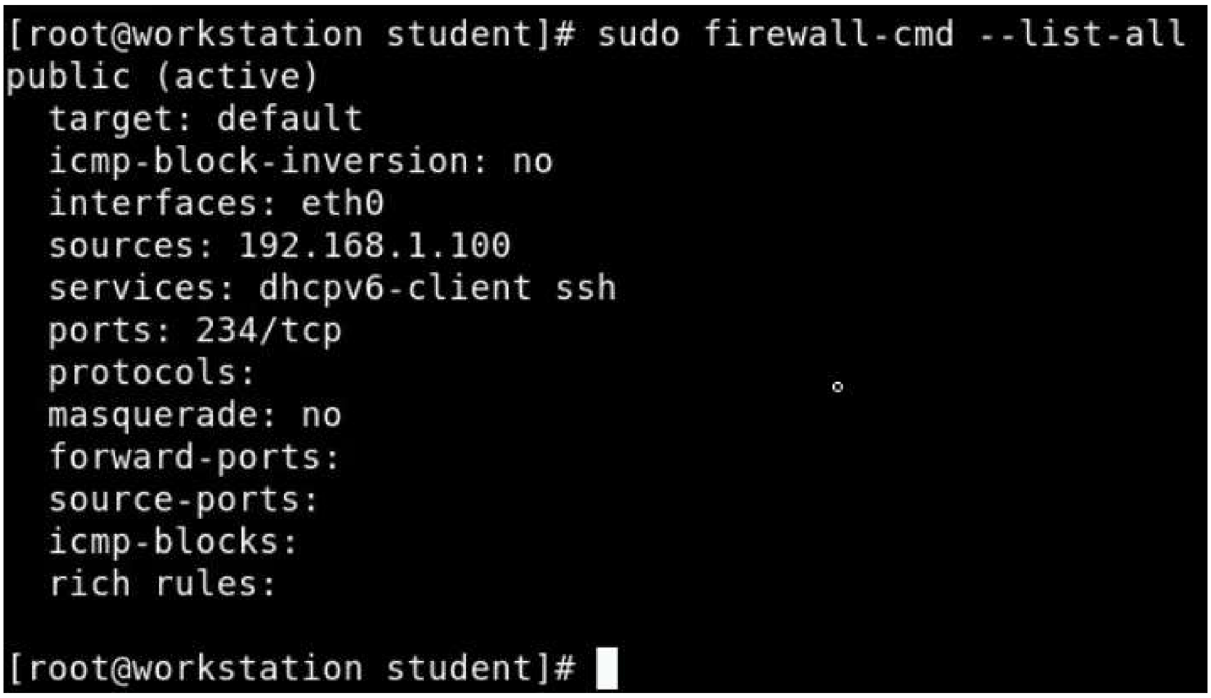
```
[root@workstation student]# sudo firewall-cmd --zone=public --add-source=192.168.1.100 --pe
rmanent
success
[root@workstation student]# sudo firewall-cmd --reload
success
[root@workstation student]# 
```

Commands :

```
sudo firewall-cmd --zone=public --add-source=192.168.1.100 --permanent
sudo firewall-cmd -reload
```

- The first command adds a rule allowing traffic from a specific IP address to the specified firewall zone.
- The second command reloads the firewall to apply the changes made by the first command.

» Now Check the firewall Services.



```
[root@workstation student]# sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources: 192.168.1.100
  services: dhcpv6-client ssh
  ports: 234/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@workstation student]# 
```

Command : `sudo firewall-cmd --list-all`

This command lists all the firewall rules to verify the current configuration.

10. <u>In your words provide the information about the semange command that you have used previously to solve question 4.</u>

» **<u>The semanage commands that I used in question 4 are shown below with break down of that commands.</u>**

<u>Commands :</u>

```
semanage port -l | grep http
semanage port -a -t http_port_t -p tcp 82
```

➥ The **semanage** command is used to manage SELinux policy, including port types and other policy components. SELinux (Security-Enhanced Linux) is a security feature in Linux that provides a mandatory access control mechanism, enforcing fine-grained policies over the system. It defines rules and policies to control the actions that processes and users can perform on the system.

Now, let's break down the two **semanage** commands that I have used in question 4:

1. <u>semanage port -l | grep http</u>

- **Explanation:**
  - **semanage port**: This is the main command for managing SELinux port definitions.
  - **-l**: The option stands for "list," and it instructs **semanage** to list all defined port types.
  - **|**: This is a pipe, taking the output of the command on its left and using it as input for the command on its right.
  - **grep http**: This part is a filtering command. It searches for lines containing the string "http" in the output of the previous command.
- **Purpose:**

- This command is used to list all SELinux port types and then filter the results to show only those related to "http." It helps you identify existing port types associated with the HTTP service.

2. semanage port -a -t http_port_t -p tcp 82

- **Explanation:**
  - **semanage port**: Again, this is the main command for managing SELinux port definitions.
  - **-a**: The option stands for "add," indicating that a new port type association should be added.
  - **-t http_port_t**: This specifies the SELinux type to be associated with the port. In this case, it's associating the **http_port_t** type, which is typically used for HTTP ports.
  - **-p tcp**: This indicates that the port type association is for the TCP protocol.
  - **82**: This is the port number for which the new association is being defined.
- **Purpose:**
  - This command adds a new SELinux port type association, specifying that port 82 (TCP) should be associated with the **http_port_t** type. This is necessary to allow the Apache HTTP Server (**httpd**) to bind to and serve content on this non-standard port.

In summary, **semanage** is a command-line tool used for managing SELinux policy, and these specific commands are used to query and modify SELinux port types, allowing the configuration of the system to support the Apache HTTP Server on a non-standard port.