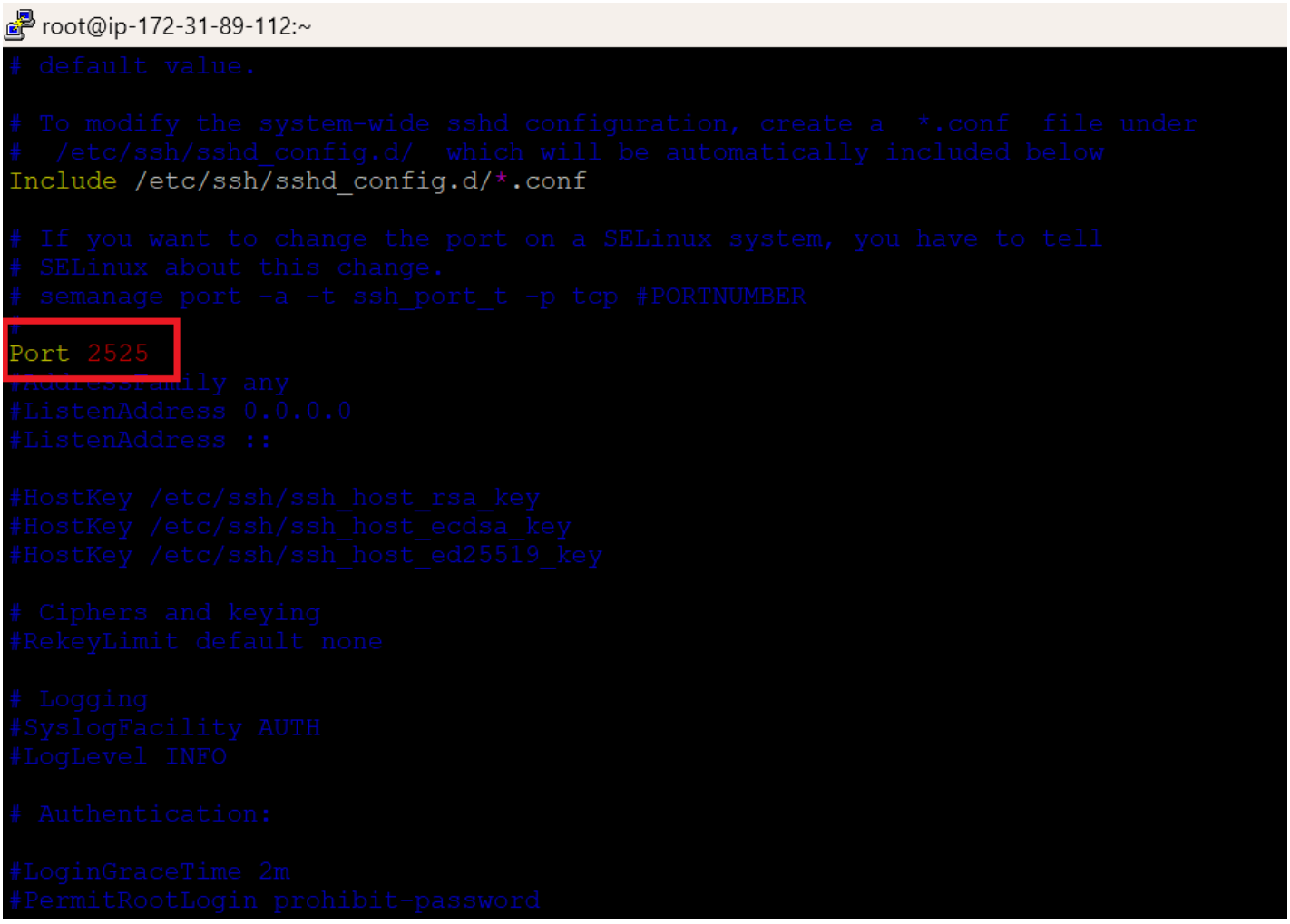
Linux Tasks:

Tasks 2: Linux

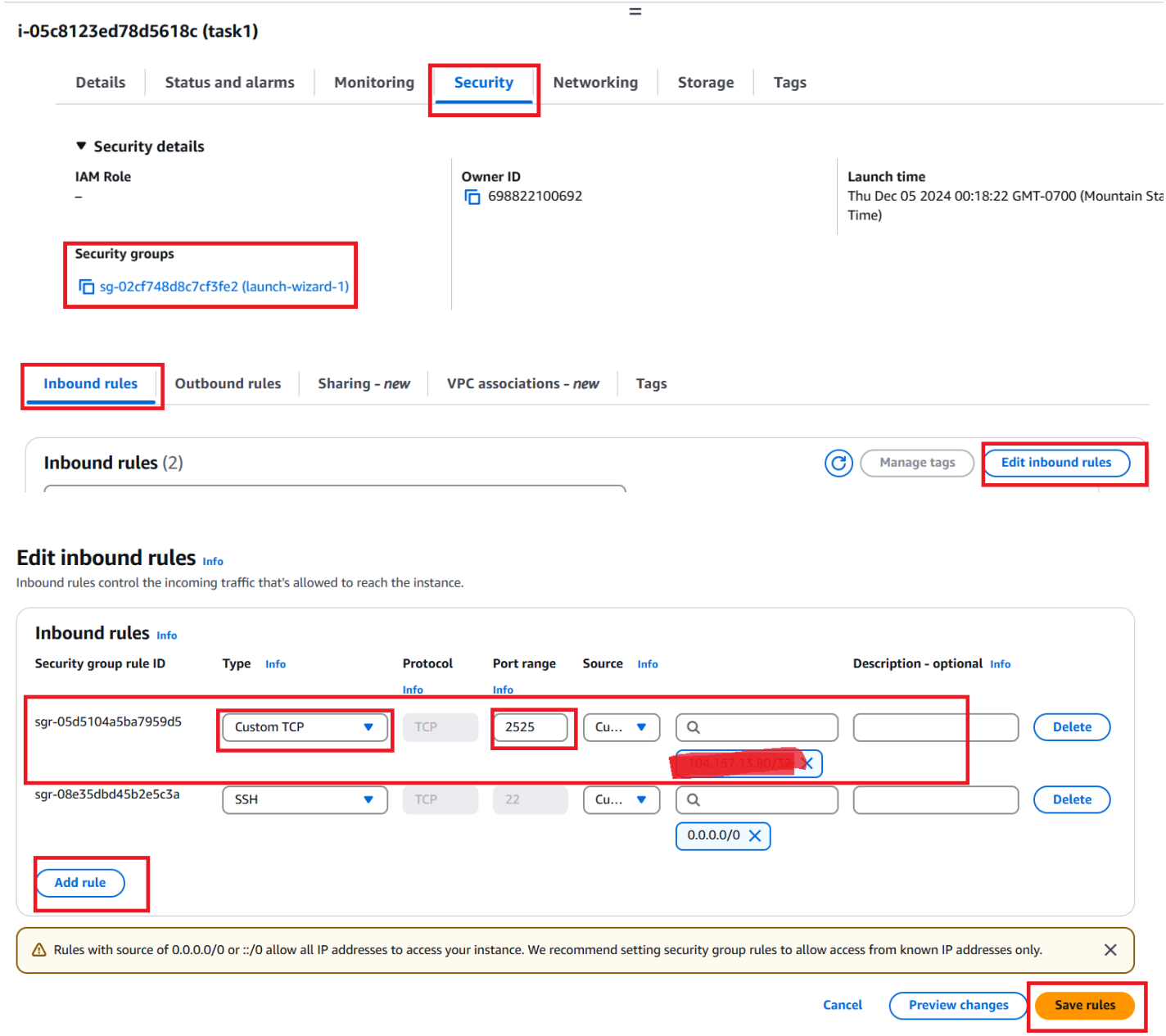
1. 1. What is the default port for ssh and how to change it?



Steps to manage the firewall depending on your setup in EC2 instance Storage group:

1. Navigate to EC2 Dashboard:
    - Go to Instances, find your instance, and note its associated Security Group.
2. Edit Security Group Rules:
    - Go to Security Groups under the Network & Security section.
    - Select your instance's security group and click Edit inbound rules.
3. Add a New Rule:
    - Add a rule for your desired port and protocol (e.g., SSH, TCP, or `Custom`).
    - Specify the port range (e.g., 2222 for a custom SSH port).
    - Set the source IP range (e.g., `0.0.0.0/0` for global access, or limit it to your IP).

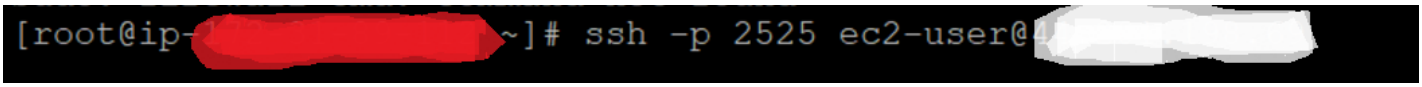
4. Save Rules:
    - Click Save to apply the new rule.



Test SSH Connection: Try connecting to your instance using the new port:

```
ssh -p 2222 ec2-user@<instance-public-ip>
```

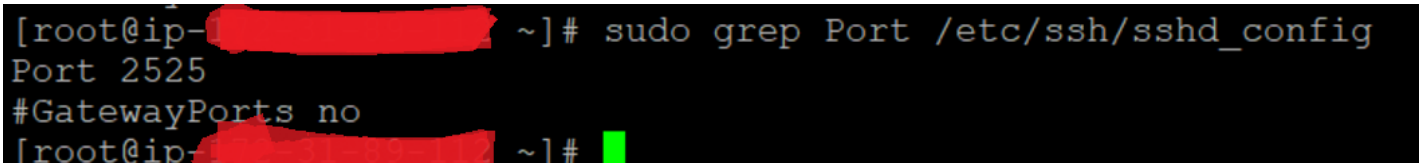
Replace 2222 with the port you configured and `<instance-public-ip>` with your instance's public IP.



Check the SSH Configuration: If the SSH connection works, confirm the port is updated in the instance's `sshd_config` file:

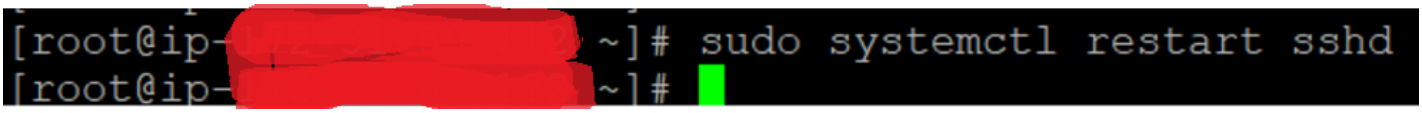
```
sudo grep Port /etc/ssh/sshd_config
```

This will display the port being used by the SSH service.



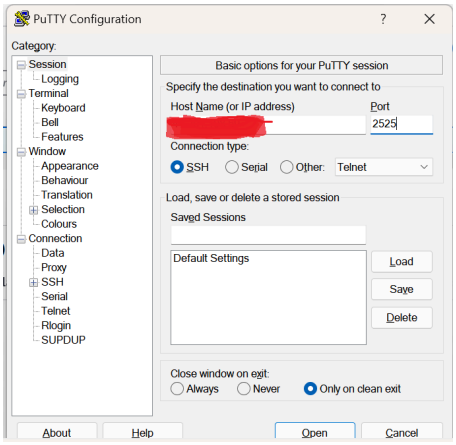
Make sure to restart the SSH service on the instance after changing the port in `sshd_config`:

```
sudo systemctl restart sshd
```



Steps to Check and Test the New SSH Port in PuTTY

1. Launch PuTTY
   Open the PuTTY application on your local machine.
2. Enter the Hostname or IP Address
   - In the "Host Name (or IP address)" field, enter the server's IP address or hostname.
3. Specify the New Port
   - In the "Port" field (default is 22), replace it with the new port number (e.g., 2525).
4. Connect to the Server
   - Click Open to initiate the SSH session.
5. Verify the Connection
   - If the connection is successful, you'll see the login prompt:
     ```
     login as:
     ```



2. Linux commands for OS Distributions Kernel version, RAM, CPU, Storage, Network?

Command to check the Linux OS distribution:

```
cat /etc/os-release
```

Command to check the Linux kernel version:

```
uname -r
```

Command to check RAM

```
free -m ---------------- (here, m is megabyte (mb))
free -k -------------(here, k is kilobyte kb)
free -h
```

In the command free -h, the -h flag stands for "human-readable" format. It makes the output easier to understand by displaying memory sizes in a more readable form, such as MB (megabytes) or GB (gigabytes), instead of bytes.

```
[root@ip-[redacted] ~]# free -k
               total        used        free      shared  buff/cache   available
Mem:          972264      132548      599604         452      240112      699000
Swap:              0           0           0
[root@ip-[redacted] ~]# free -m
               total        used        free      shared  buff/cache   available
Mem:              949         129         585           0         234         682
Swap:             0           0           0
[root@ip-[redacted] ~]# free -h
               total        used        free      shared  buff/cache   available
Mem:           949Mi       129Mi       585Mi       0.0Ki       234Mi       682Mi
Swap:             0B          0B          0B
[root@ip-[redacted] ~]# free
               total        used        free      shared  buff/cache   available
Mem:          972264      132632      599364         452      240268      698912
Swap:              0           0           0
[root@ip-[redacted] ~]#
```

CPU information

`commands top, lscpu` show CPU information on how many CPUs are available

```
root@ip-172-31-25-23:~
top - 03:12:47 up 33 min,  2 users,  load average: 0.00, 0.00, 0.00
Tasks: 102 total,   1 running, 101 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.3 sy,  0.0 ni, 97.7 id,  0.0 wa,  0.0 hi,  0.0 si,  2.0 st
MiB Mem :    949.5 total,    591.4 free,    130.5 used,    227.6 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    681.6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   3325 root      20   0  223924   3360   2700 R   0.3   0.3   0:00.12 top
      1 root      20   0  105700  16984  10580 S   0.0   1.7   0:00.75 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
```

In the above image
Top: 03:12:47 (sys time)
33min (since when system is up and running)
CPU 0 means 1 cpu available, CPU 1 means 2 CPUs etc
under CPU 97.7id means ideal 97.7@ space is left and only 1% is used
MiB Mem 949.5 total memory
MiB swap 681.6 is available memory

Storage Information

`Command df -h` for Storage Information

```
[root@ip-[redacted] ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M     0  4.0M   0% /dev
tmpfs           475M     0  475M   0% /dev/shm
tmpfs           190M  448K  190M   1% /run
/dev/xvda1      8.0G  1.6G  6.4G  20% /
tmpfs           475M     0  475M   0% /tmp
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi
tmpfs            95M     0   95M   0% /run/user/1000
```

Network information

`Commands ip a and ip addr` both gives the same information about Network Connections

3. What is the difference between $ vs # in Linux?

 # (Hash/Pound Sign)

Indicates the root user prompt: The # symbol represents the root user or superuser prompt. It indicates that the user has administrative privileges and can execute commands that require elevated permissions, such as system configuration, installation of software, etc.
Example:

`root@hostname:~#` `[root@ip-172-31-89-112 ~]#`

- Command Line: The root user has the ability to modify system files, install or remove software, and perform tasks that could affect the entire system.

$ (Dollar Sign)

Indicates a regular user prompt: The $ symbol is typically displayed in the shell prompt when a normal user is logged in (i.e., a non-root user). It signifies that the user has limited privileges and is not performing administrative tasks.
Example:

`user@hostname:~$`

`[ec2-user@ 12 etc]$`

- **Command Line:** Regular users use this prompt to execute commands that do not require root privileges.

4. (a) How do you Install two Java 11 and 8 versions?

Install Java 8:

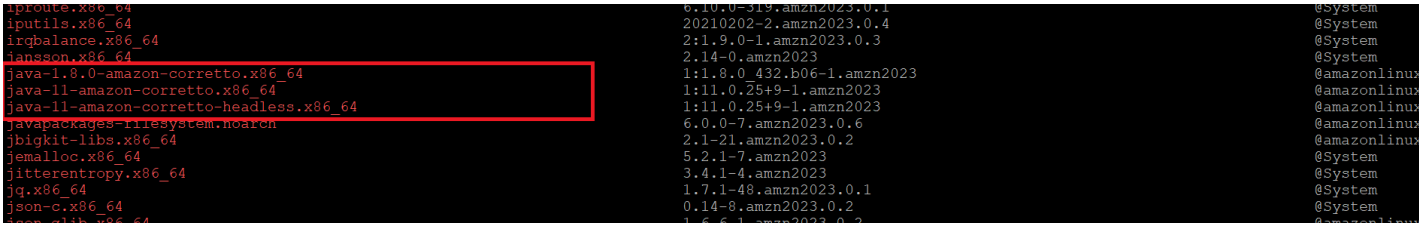
```
yum install java-1.8.0
```

Install Java 11:

```
yum install java-11
```

Verify Installations: Check that both versions are installed:

```
java -version
```



4. (b) 32-bit and 64-bit java 11 installation?

5. How do you List all packages that are installed?

To list all installed packages on an Amazon Linux instance, you can use the command

`Yum list installed`

```
[root@ip-   4 yum.repos.d]# yum list installed
Installed Packages
acl.x86_64                          2.3.1-2.amzn2023.0.2              @System
acpid.x86_64                        2.0.32-4.amzn2023.0.2             @System
alsa-lib.x86_64                     1.2.7.2-1.amzn2023.0.2            @amazonlinux
alternatives.x86_64                 1.15-2.amzn2023.0.2               @System
amazon-chrony-config.noarch         4.3-1.amzn2023.0.4                @System
amazon-ec2-net-utils.noarch         2.5.1-1.amzn2023.0.1              @System
amazon-linux-repo-s3.noarch         2023.6.20241121-0.amzn2023        @System
amazon-linux-sb-keys.noarch         2023.1-1.amzn2023.0.5             @System
amazon-rpm-config.noarch            228-4.amzn2023.0.1                @System
amazon-ssm-agent.x86_64             3.3.987.0-1.amzn2023              @System
amd-ucode-firmware.noarch           20210208-117.amzn2023.0.6         @System
at.x86_64                           3.1.23-6.amzn2023.0.2             @System
attr.x86_64                         2.5.1-3.amzn2023.0.2              @System
audit.x86_64                        3.0.6-1.amzn2023.0.2              @System
audit-libs.x86_64                   3.0.6-1.amzn2023.0.2              @System
aws-cfn-bootstrap.noarch            2.0-31.amzn2023                   @System
awscli-2.noarch                     2.15.30-1.amzn2023.0.1            @System
basesystem.noarch                   11-11.amzn2023.0.2                @System
bash.x86_64                         5.2.15-1.amzn2023.0.2             @System
bash-completion.noarch              1:2.11-2.amzn2023.0.2             @System
```

6. How do you check Java is installed?

`Java -version`

```
[root@ip-   4 yum.repos.d]# java -version
openjdk version "23.0.1" 2024-10-15
OpenJDK Runtime Environment Corretto-23.0.1.8.1 (build 23.0.1+8-FR)
OpenJDK 64-Bit Server VM Corretto-23.0.1.8.1 (build 23.0.1+8-FR, mixed mode, sharing)
```

7. How do you check which applications are running?

Check which applications are running

`command ps  -ef`

1. Using the ps Command

The ps command provides a snapshot of currently running processes.

`ps`

Displays processes running in the current shell session.

2. Show All Processes:

`ps -e`

Lists all processes on the system.

3. Detailed View:

`ps -ef`

Provides a detailed view, including the user, PID, and command line of each process.

8. How to change permissions/ownerships recursively? folder1/ file1 file2 file3

Steps to Change Permissions Recursively

Step 1: Create folder Command **`mkdir folder1`**

Step 2: Create files inside the folder **`touch file1 file2 file3`**

Step 3: Run command **`sudo chmod -R 751 folder1/`**

Step 4: Check if permissions changed or not using command l**`s -li`**

Step 5: Go to folder1 using command **`cd folder1`**

Step 6: check the list of files if the files permissions are changed using command **`ls -li`**

```
[ec2-user@ip-172-31-17-131 ~]$ sudo chmod -R 755 folder1/
[ec2-user@ip-172-31-17-131 ~]$ ls -li
total 0
8542065 drwxr-xr-x. 2 ec2-user ec2-user 45 Dec 12 06:52 folder1
[ec2-user@ip-172-31-17-131 ~]$ cd folder1
[ec2-user@ip-172-31-17-131 folder1]$ ls -li
total 0
8542066 -rwxr-xr-x. 1 ec2-user ec2-user 0 Dec 12 06:52 file1
8542067 -rwxr-xr-x. 1 ec2-user ec2-user 0 Dec 12 06:52 file2
8542068 -rwxr-xr-x. 1 ec2-user ec2-user 0 Dec 12 06:52 file3
```

```
[ec2-user@ip-172-31-17-131 folder1]$ cd
[ec2-user@ip-172-31-17-131 ~]$ sudo chmod -R 751 folder1/
[ec2-user@ip-172-31-17-131 ~]$ ls -li
total 0
8542065 drwxr-x--x. 2 ec2-user ec2-user 45 Dec 12 06:52 folder1
[ec2-user@ip-172-31-17-131 ~]$ cd folder1
[ec2-user@ip-172-31-17-131 folder1]$ ls -li
total 0
8542066 -rwxr-x--x. 1 ec2-user ec2-user 0 Dec 12 06:52 file1
8542067 -rwxr-x--x. 1 ec2-user ec2-user 0 Dec 12 06:52 file2
8542068 -rwxr-x--x. 1 ec2-user ec2-user 0 Dec 12 06:52 file3
[ec2-user@ip-172-31-17-131 folder1]$
```

Steps to Change Ownership Recursively

Step 1: Create folder Command **`mkdir folder1`**

Step 2: Create files inside the folder **`touch file1 file2 file3`**

Step 3: Add group using Command **`groupadd groupname`**

Step 4: Add group using Command **`useradd -g groupname username`**

Step 5: Run command **`sudo chown -R username:groupname folder1/`**

Step 6: Check if permissions changed or not using command l**`s -li`**

Step 7: Go to folder1 using command **`cd folder1`**

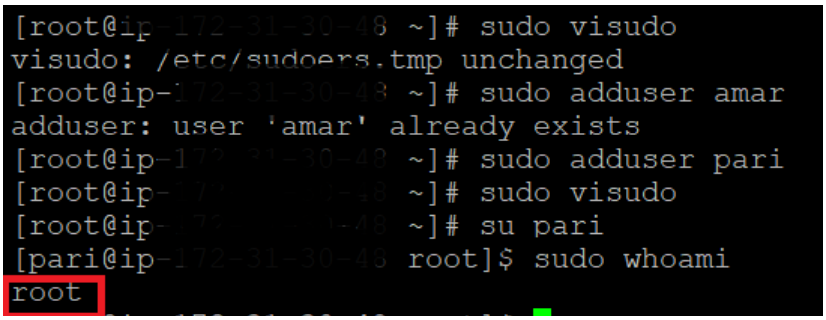
Step 8: check the list of files if the files permissions are changed using command **`ls -li`**

```
[root@ip-172-31-17-131 ~]# cd /folder1
-bash: cd: /folder1: No such file or directory
[root@ip-172-31-17-131 ~]# Groupadd development
-bash: Groupadd: command not found
[root@ip-172-31-17-131 ~]# groupadd dev
[root@ip-172-31-17-131 ~]# useradd -g dev parijatha
[root@ip-172-31-17-131 ~]# useradd -g dev pari
[root@ip-172-31-17-131 ~]# id parijatha
uid=1001(parijatha) gid=1001(dev) groups=1001(dev)
[root@ip-172-31-17-131 ~]# su ec2-user
```

```
[ec2-user@ip-172-31-17-131 ~]$ sudo chown -R parijatha:dev folder1/
[ec2-user@ip-172-31-17-131 ~]$ ls -li
total 0
8542065 drwxr-x--x. 2 parijatha dev 45 Dec 12 06:52 folder1
[ec2-user@ip-172-31-17-131 ~]$ cd folder1/
```

```
[ec2-user@ip-172-31-17-131 folder1]$ sudo ls -li
total 0
8542066 -rwxr-x--x. 1 parijatha dev 0 Dec 12 06:52 file1
8542067 -rwxr-x--x. 1 parijatha dev 0 Dec 12 06:52 file2
8542068 -rwxr-x--x. 1 parijatha dev 0 Dec 12 06:52 file3
```

9. How do you provide sudo access to Amar only for one particular command? ex: useradd sudo useradd prasad



Steps to Grant Sudo Access

1. Add User to the Sudo Group

This is the simplest way to provide sudo access.

Check if the User Already Exists: If the user does not exist, create it:
`sudo adduser <username>`

Example:
`sudo adduser john`

2. Edit the Sudoers File (Advanced Method): Open the Sudoers File Safely with `visudo`:
`sudo visudo`

3. This ensures you do not accidentally corrupt the sudoers file.: Grant Sudo Privileges to the User: Add the following line at the end of the file:

SQL:
`<username> ALL=(ALL) NOPASSWD:ALL`

Example:
css
`john ALL=(ALL) ALL`

- `ALL=(ALL) ALL`: The user can run any command as any user on any host.
- `NOPASSWD`: If added, the user will not be prompted for a password when using `sudo`.

4. Save and Exit:

- **In `visudo`, press `Ctrl+X`, then `Y`, and `Enter` to save and exit.**

5. Test the Sudo Access: Switch to the user and try running a command with sudo:

```
su - <username>
sudo whoami
```

6. If the setup is correct, the output will be:

```
root
```

10. What are the log files and location of a log file in Amazon Linux?

In Amazon Linux, log files are typically used to store system, application, and service messages. These files are essential for troubleshooting and monitoring the system's health.

Common Log Files in Amazon Linux

1. **System Logs:**
   - **System Messages:**
     Path: `/var/log/messages`
     Purpose: General system activity logs, including kernel, boot processes, and system services.
   - **Kernel Logs:**
     Path: `/var/log/kern.log` **(if available)**
     Purpose: Logs specific to kernel activity.
   - **Authentication Logs:**
     Path: `/var/log/secure`
     Purpose: Logs related to user authentication, SSH access, and sudo usage.
2. **Service Logs:**
   - **HTTP Server Logs (Apache/NGINX):**
     - **Apache:** `/var/log/httpd/`
     - **NGINX:** `/var/log/nginx/`
       Purpose: Logs for web server access and error messages.
   - **Cron Logs:**
     Path: `/var/log/cron`
     Purpose: Logs of scheduled tasks via `cron`.
   - **MySQL Logs:**
     Default path: `/var/log/mysqld.log`
     Purpose: Logs related to MySQL database operations.
3. **Systemd Journals:**
   Path: `/var/log/journal/`
   Purpose: Logs for systemd-managed services. If `journalctl` is enabled, these are binary logs accessed using `journalctl` commands.
4. **Boot Logs:**
   Path: `/var/log/boot.log`
   Purpose: Logs from the system boot process.

5. **Dmesg Logs:**
   Path: `/var/log/dmesg`
   Purpose: Kernel ring buffer messages during boot or runtime.

Viewing and Managing Logs

To view a specific log file:

```
cat /var/log/<filename>
```

1. **or use `less`, `more`, or `tail` for a better view.**

To watch logs in real time:

```
tail -f /var/log/<filename>
```

2. **Using `journalctl` (if applicable):**

View all logs:

```
Journalctl
```

View logs for a specific service:

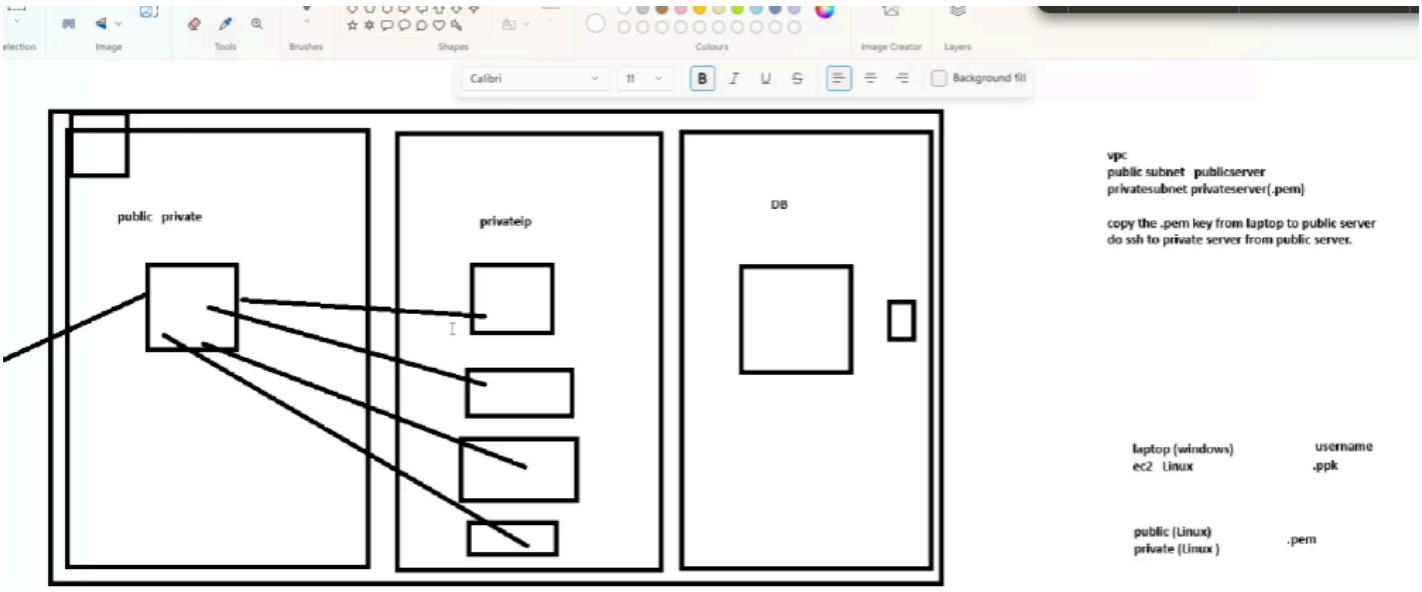
```
journalctl -u <service_name>
```

Task 8: IAM ROLE

1.1. Create a program in Lambda to shutdown EC2 daily at 6pm IST start at 9am IST?

a. How to schedule?
b. Create a ROLE with EC2 policy

2.2. Create 2 AWS accounts and SSO (Singlesignon it means by connecting to one account we can switch to another account) setup



Create 1 EC2
Install

Increase and decrease load using autoscaling
How to delete servers when load is decreased?

13- 02-2025

1) how to optimize docker image size ?

2) best practices while creating image?

3) try to build below images using docker file

a) tomcat

b) apache

c) jenkins