Session-2

Linux For DevOps

Linux is an open-source, Unix-like operating system kernel that serves as the foundation for various operating systems, collectively known as "Linux distributions" or "Linux distros."

Key components of a typical Linux-based system:

1. **Kernel**:
   - The kernel is the core component of any operating system. It manages the system's resources, including the CPU, memory, and hardware devices.
   - Linux, the kernel, is responsible for tasks like process management, memory management, device management, and system calls.
2. **Shell**:
   - The shell is the interface between the user and the kernel. It interprets user commands and communicates them to the operating system.
   - Common Linux shells include Bash (Bourne Again Shell), Zsh (Z shell), and others.
3. **File System**:
   - The file system is a method used by the operating system to organize and store files on storage devices (like hard drives or SSDs).
   - Linux supports a variety of file systems including ext4, Btrfs, XFS, and more.
4. **User Interface**:
   - Linux can have various types of user interfaces. The most common ones are:
     - **Command Line Interface (CLI)**: Users interact with the system by typing commands into a terminal or console.
     - **Graphical User Interface (GUI)**: Provides a graphical environment

•

•

○

with windows, icons, buttons, and menus. Common Linux GUI environments include GNOME, KDE, XFCE, etc.

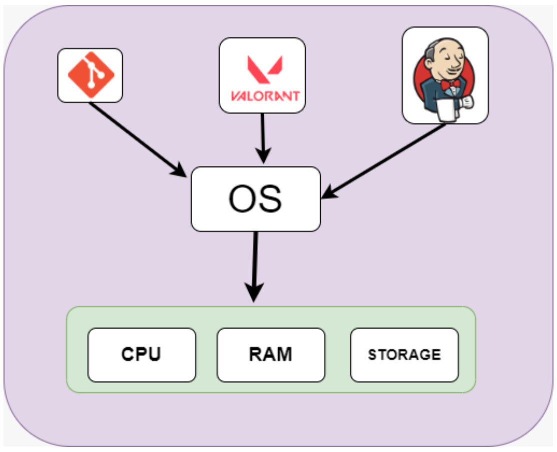
5. **System Libraries**:
   - These are collections of pre-compiled code that provide common functions to programs. They simplify programming and improve efficiency.
6. **System Utilities**:
   - These are programs used for tasks like managing files, system configuration, monitoring system performance, and more. Examples include ls, cp, mv, ps, and top.
7. **Device Drivers**:
   - Device drivers are software components that enable the operating system to communicate with hardware devices like printers, graphics cards, and network cards.
8. **Applications and Software Packages**:
   - Linux provides a vast repository of software packages that can be installed to extend the functionality of the system. Package managers like `apt` (used by Debian-based systems), `yum` (used by Red Hat-based systems), and others handle the installation and management of software.
9. **User Accounts and Permissions**:
   - Linux is a multi-user system, meaning it can have multiple user accounts. Each user has their own environment, settings, and files.
   - Permissions control what users and groups can do with files and directories.
10. **Networking**:
    - Linux supports a wide range of networking protocols and services. It can function as a client or server for services like web browsing, email, file sharing, and more.
11. **Security**:
    - Linux provides various security features, including user authentication, access control lists, firewalls, and encryption.
12. **Process Management**:
    - The kernel is responsible for managing processes, which are running instances of programs. It schedules them, allocates

-
-

◦

resources, and ensures they don't interfere with one another.

13. **Virtualization and Containers**:
    - Linux supports virtualization technologies (e.g., KVM, Xen) and containerization platforms (e.g., Docker) that allow multiple isolated environments to run on a single physical machine.



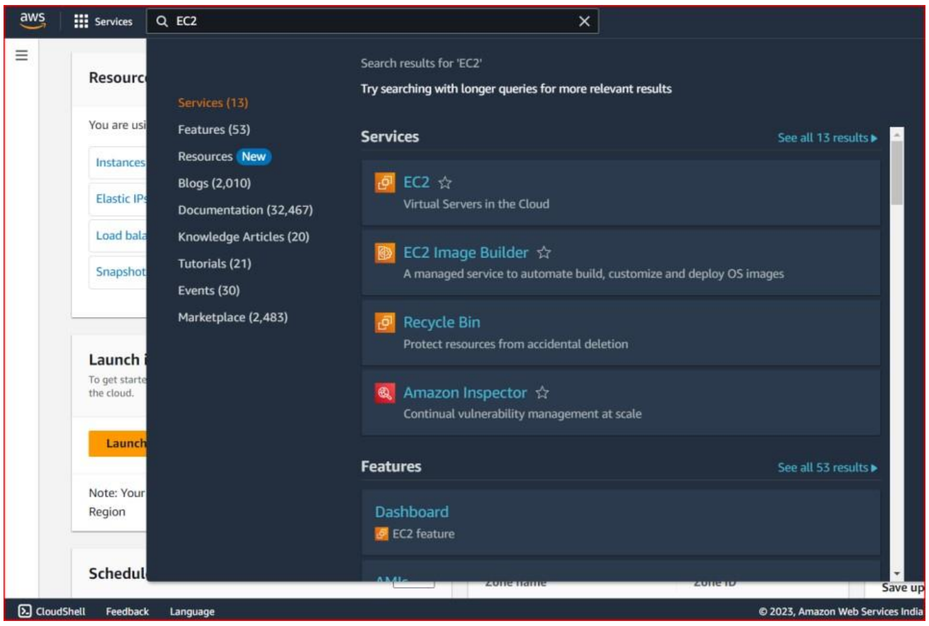
Linux is an operating system (OS) that serves as an intermediary between hardware and software applications, enabling them to communicate and work together seamlessly.

Creating a Linux Machine in AWS

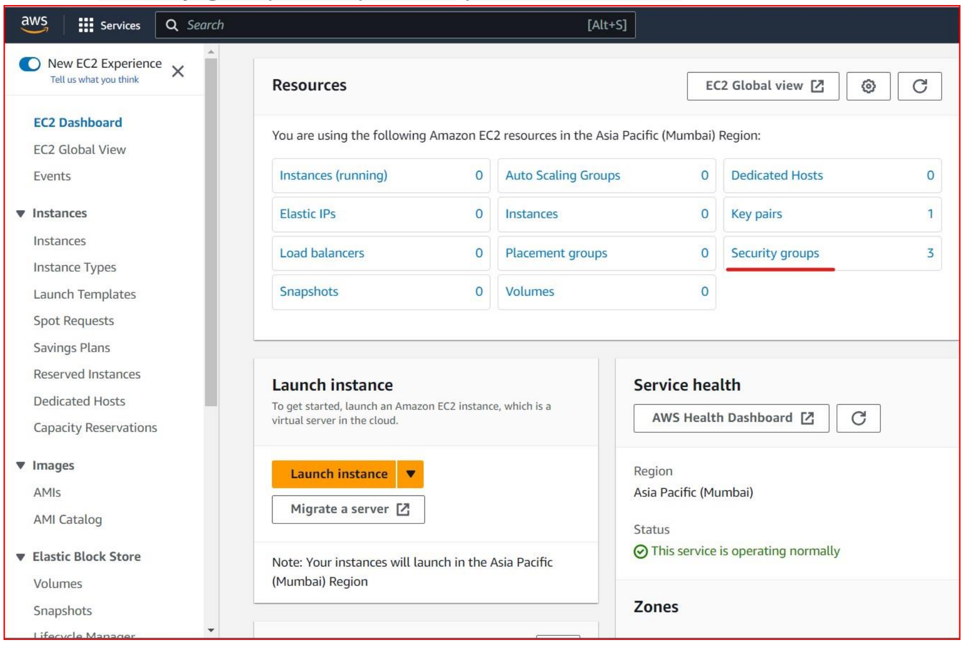
1. Create an AWS Account and log in to AWS Management console.
2. Go to search, and type EC2

-
-

◦



3. Click on EC2 and below page will open , in it click on security groups to open the ports on the VM.

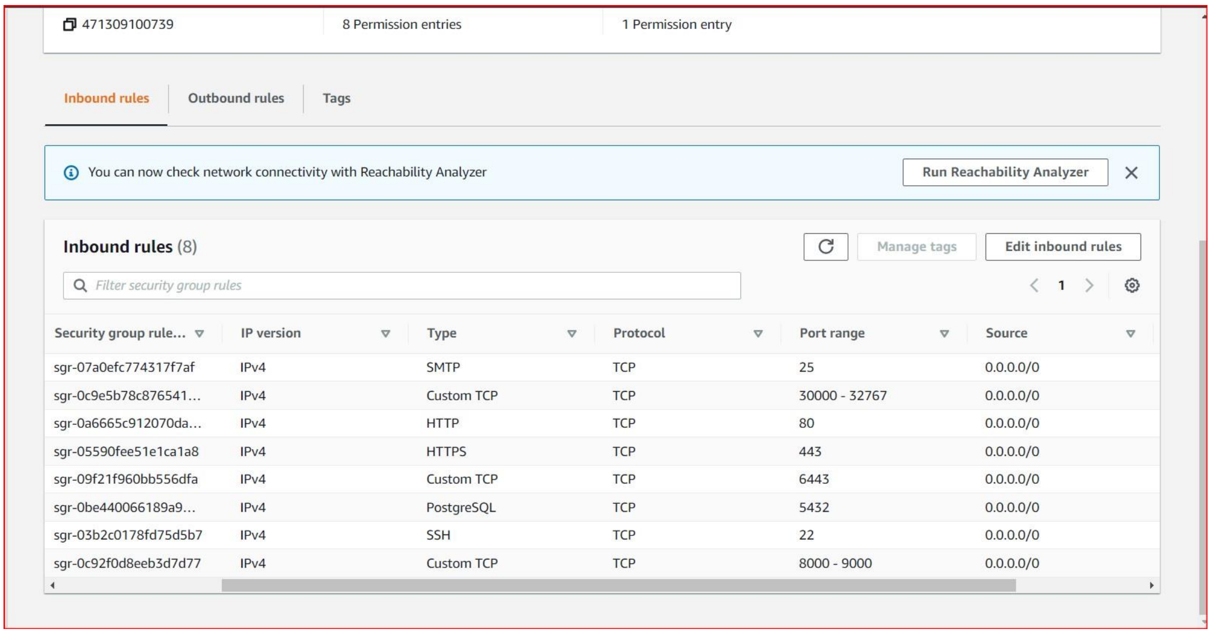


4. Inside the security group, I would request you to open below ports, as we will be needing those.

•

•

◦


471309100739 | 8 Permission entries | 1 Permission entry

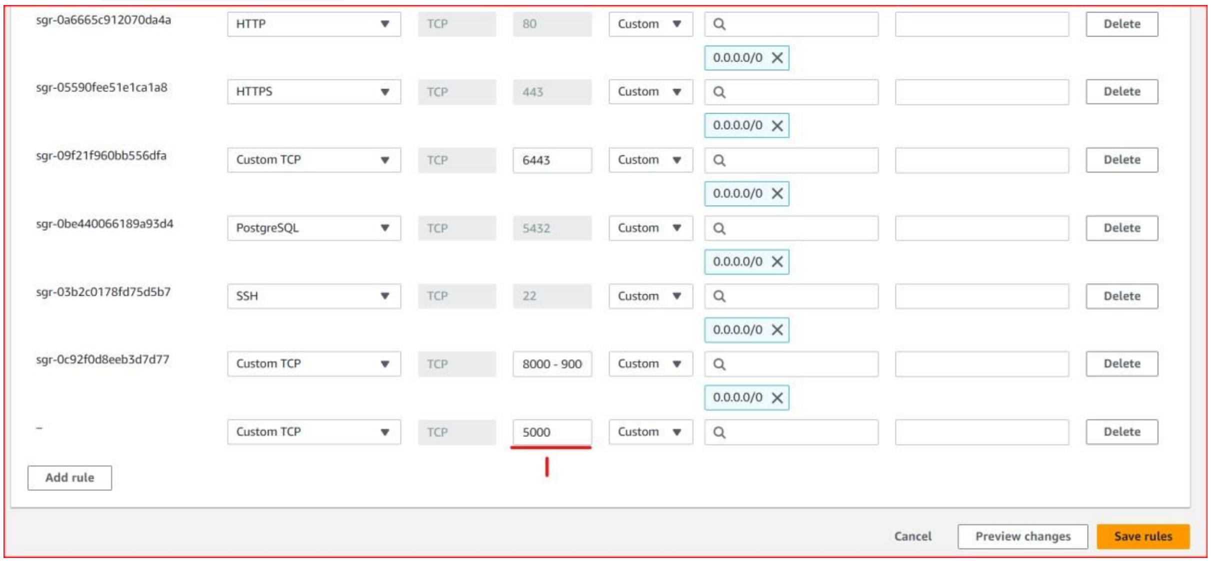
Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer

Inbound rules (8)

Security group rule...	IP version	Type	Protocol	Port range	Source
sgr-07a0efc774317f7af	IPv4	SMTP	TCP	25	0.0.0.0/0
sgr-0c9e5b78c876541...	IPv4	Custom TCP	TCP	30000 - 32767	0.0.0.0/0
sgr-0a6665c912070da...	IPv4	HTTP	TCP	80	0.0.0.0/0
sgr-05590fee51e1ca1a8	IPv4	HTTPS	TCP	443	0.0.0.0/0
sgr-09f21f960bb556dfa	IPv4	Custom TCP	TCP	6443	0.0.0.0/0
sgr-0be440066189a9...	IPv4	PostgreSQL	TCP	5432	0.0.0.0/0
sgr-03b2c0178fd75d5b7	IPv4	SSH	TCP	22	0.0.0.0/0
sgr-0c92f0d8eeb3d7d77	IPv4	Custom TCP	TCP	8000 - 9000	0.0.0.0/0

4. To Open a port you provide details as below.
   ❼Click create security group and click on add rule and add as below.
   ❼If editing an already existing group then click on edit inbound rules.



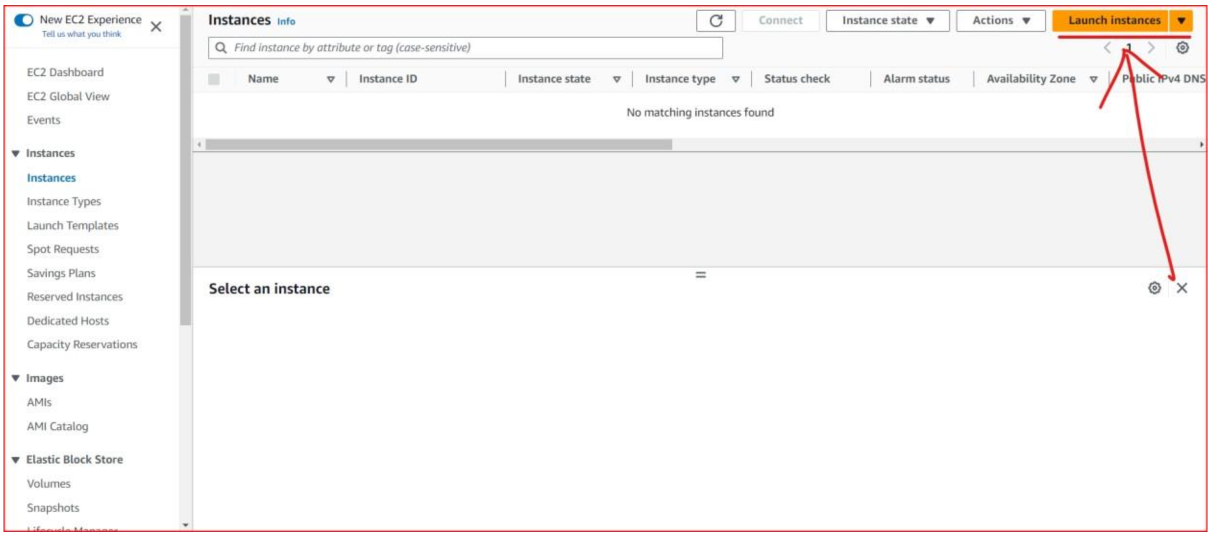
sgr-0a6665c912070da4a	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
sgr-05590fee51e1ca1a8	HTTPS	TCP	443	Custom	0.0.0.0/0		Delete
sgr-09f21f960bb556dfa	Custom TCP	TCP	6443	Custom	0.0.0.0/0		Delete
sgr-0be440066189a93d4	PostgreSQL	TCP	5432	Custom	0.0.0.0/0		Delete
sgr-03b2c0178fd75d5b7	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-0c92f0d8eeb3d7d77	Custom TCP	TCP	8000 - 900	Custom	0.0.0.0/0		Delete
–	Custom TCP	TCP	5000	Custom			Delete

5. Once done, then click on instances or launch instance .

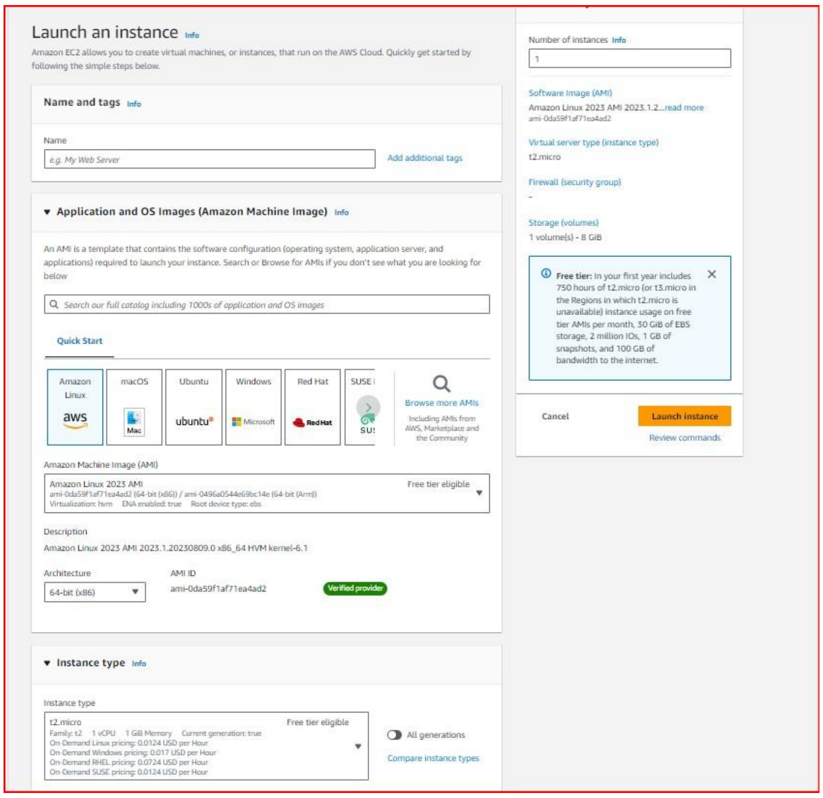
•

•

◦



6. Next up, we will select the configuration for our VM. If doing for the first time then make sure to create a new key pair and keep the key safe.


Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: e.g. My Web Server

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Amazon Machine Image (AMI)
Amazon Linux 2023 AMI — Free tier eligible
ami-0da59f1af71ea4ad2 (64-bit (x86)) / ami-0496a0544e69bc14e (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2023 AMI 2023.1.20230809.0 x86_64 HVM kernel-6.1

Architecture: 64-bit (x86)
AMI ID: ami-0da59f1af71ea4ad2 Verified provider

Instance type Info

Instance type
t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour
On-Demand RHEL pricing: 0.0724 USD per Hour
On-Demand SUSE pricing: 0.0124 USD per Hour

Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.1.2...read more
ami-0da59f1af71ea4ad2

Virtual server type (instance type)
t2.micro

Firewall (security group)
-

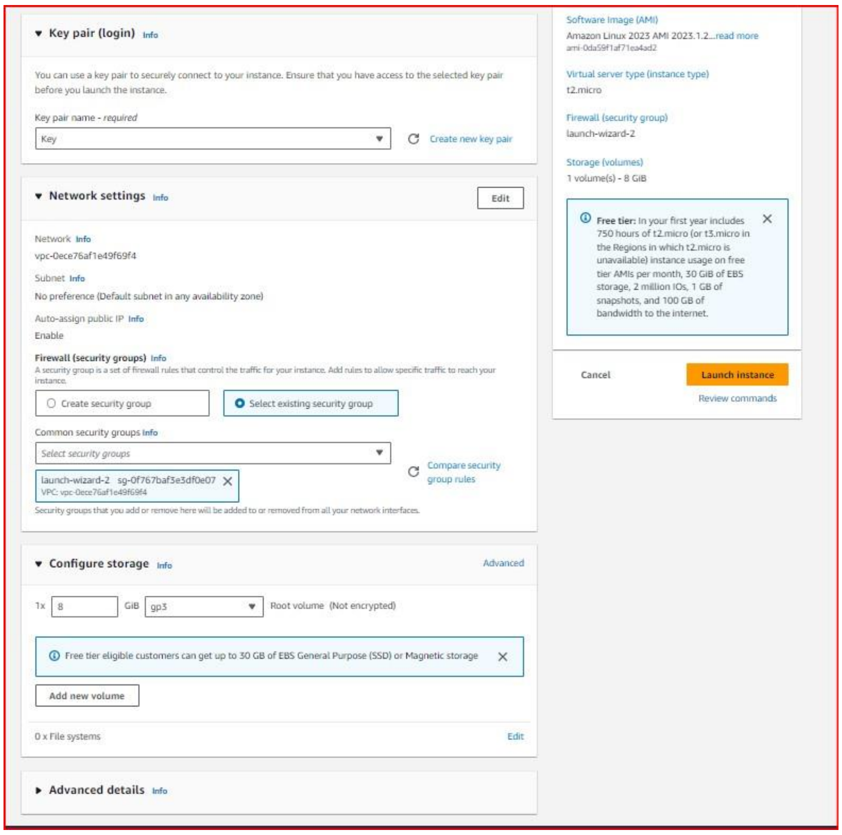
Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

•

•

○



Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: Key

Network settings Info

Network: vpc-0ece76af1e49f69f4

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group [✓ Select existing security group]

Common security groups: launch-wizard-2 sg-0f767baf3e3df0e07 VPC: vpc-0ece76af1e49f69f4

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Configure storage Info

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

0 x File systems

Advanced details Info

Software Image (AMI): Amazon Linux 2023 AMI 2023.1.2...read more ami-0da59f1af71ea4ad2

Virtual server type (instance type): t2.micro

Firewall (security group): launch-wizard-2

Storage (volumes): 1 volume(s) - 8 GiB

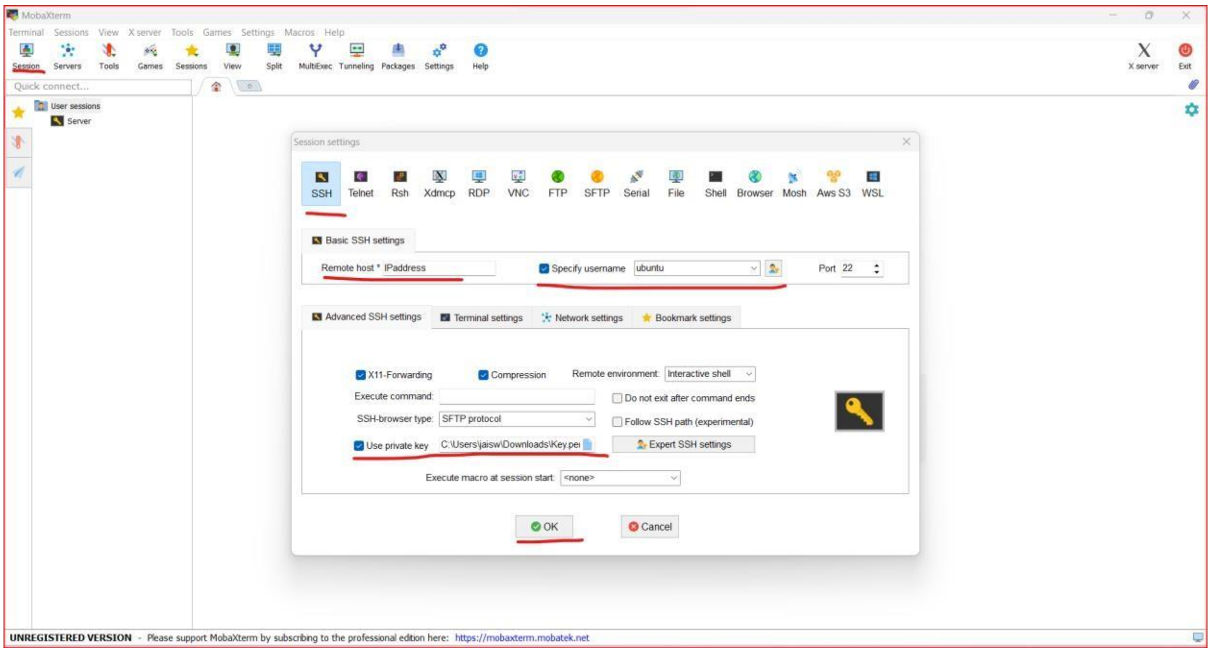
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

7. Once your VM is created, you can access it using a tool Mobaxterm(https://download.mobatek.net/2322023060714555/MobaXterm_Portable_v23.2.zip)
    - Remote host-> Public IP
    - Username-> ubuntu if created ubuntu machine
    - Use private key-> select the pem file downloaded while creating the new key

•

•

○


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
User sessions
Server
Session settings
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL
Basic SSH settings
Advanced SSH settings
Terminal settings
Network settings
Bookmark settings
Expert SSH settings
OK
Cancel

Remote host * IPaddress

✓ Specify username: ubuntu

Port: 22

✓ X11-Forwarding

✓ Compression

Remote environment: Interactive shell

Execute command:

☐ Do not exit after command ends

SSH-browser type: SFTP protocol

☐ Follow SSH path (experimental)

✓ Use private key: C:\Users\jaisw\Downloads\Key.pe

Execute macro at session start: <none>

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

Linux File system:

1. **/home:** This directory contains user home directories. Each user typically has a subdirectory here where they can store their personal files and configuration settings.

2. **/root:** This is the home directory for the root user, which is the superuser or administrator of the system.

3. **/bin:** Short for "binary," this directory contains essential system binaries (executable files) that are required for basic system operations, such as system maintenance and recovery.

4. **/sbin:** Similar to "/bin," this directory contains system binaries, but these are typically used by the system administrator for system maintenance and configuration tasks.

•

•

5. **/lib:** This directory holds essential libraries (shared files) needed for the system and applications to run properly.

6. **/usr:** This directory contains a variety of subdirectories including:

    - /usr/bin: User-level binaries and executables.
    - /usr/sbin: System binaries for administrative tasks.
    - /usr/lib: Libraries for user-level programs.
    - /usr/local: Locally installed software and libraries (not managed by the package manager).

7. **/opt:** This is often used for installing optional or third-party software.
Software packages are installed in their own subdirectories within "/opt."

8. **/etc:** This directory contains system-wide configuration files. Configuration settings for various applications and services are stored here.

9. **/tmp:** Temporary files are stored in this directory. The contents of this directory are often cleared upon system reboot.

10. **/boot:** Contains files related to the boot process, such as the Linux kernel and bootloader configuration.

11. **/dev:** This directory contains device files that represent hardware devices on the system. These files allow programs to interact with hardware without needing to know specific details about the hardware.

◦

12. **/var:** This directory contains variable data files that are expected to grow during the system's operation. This includes log files, spool directories, and other transient data.

13. **/media:** Mount point for removable media devices like USB drives and CD/DVD-ROM drives.

14. **/mnt:** A traditional location to temporarily mount additional filesystems or devices. This directory can also be used to manually mount filesystems.

Linux Must known concepts:

-
-

○

# Changing Permissions

In Linux, permissions for files and directories are represented using a three-character string that stands for Read (R), Write (W), and Execute (X) permissions. These permissions are assigned to three categories of users: Owner (u), Group (g), and Others (o).

- **Numeric Method**:
  In the numeric method, each permission is represented by a number:
  - Read (R) = 4
  - Write (W) = 2
  - Execute (X) = 1
  - No permission = 0

  To assign permissions, you sum up these values. For example:
  - Read and write (RW) = 4 (read) + 2 (write) = 6
  - Read, write, and execute (RWX) = 4 (read) + 2 (write) + 1 (execute) = 7

  Permissions are then represented as a three-digit number, where the first digit represents the owner, the second digit represents the group, and the third digit represents others.
  - **Chmod Command**:
    To change permissions using the numeric method, you would use the `chmod` command. For example, to give the owner read, write, and execute permissions, and the group read and execute permissions, but only give others read permission, you would use:

    ```
    chmod 751 filename
    ```

    - `7` (Owner) = 4 (read) + 2 (write) + 1 (execute)
    - `5` (Group) = 4 (read) + 1 (execute)
    - `1` (Others) = 1 (execute)
- **Symbolic Method**:
  In the symbolic method, permissions are represented using letters and symbols:
  - `u` stands for owner, `g` for group, and `o` for others.

-

-

◦

- **+** adds a permission, **-** removes a permission, and **=** sets the permission.
- **r** for read, **w** for write, and **x** for execute.

For example:

- To give the owner read and write permissions, you would use:

```
chmod u+rw filename
```

- To remove execute permission for others, you would use:

```
chmod o-x filename
```

- To set read-only permissions for everyone, you would use:

```
chmod a=r filename
```

Remember to replace "filename" with the actual name of the file or directory you want to modify.

These methods give you flexibility in managing permissions in Linux, allowing you to specify precisely who can do what with a file or directory.

Is this conversation helpful so far?

# Changing Ownership of Files/Folders

In Linux, ownership of files and directories is determined by two attributes: the user who owns the file (owner) and the group associated with the file. You can change ownership using the **chown** command.

### Changing Ownership of Files and Directories

1. *Changing Owner:*

Syntax:

```
sudo chown [new_owner] [file/directory]
```

Example:

Let's say you want to change the owner of a file named **example.txt** to a user named **newuser**:

•

•

○

```
sudo chown newuser example.txt
```

This will change the owner of **example.txt** to **newuser**.

2. *Changing Group:*
Syntax:

```
sudo chown :[new_group] [file/directory]
```

Example:

To change the group of **example.txt** to a group called **newgroup**:

```
sudo chown :newgroup example.txt
```

This will change the group of **example.txt** to **newgroup**.

3. *Changing Both Owner and Group:*
Syntax:

```
sudo chown [new_owner]:[new_group] [file/directory]
```

Example:

To change both the owner and group of **example.txt**:

```
sudo chown newuser:newgroup example.txt
```

This will change the owner to **newuser** and the group to **newgroup**.

4. *Changing Ownership Recursively:*
If you want to change ownership recursively for all files and subdirectories within a directory, you can use the **-R** flag.

Syntax:

```
sudo chown -R [new_owner]:[new_group] [directory]
```

Example:

To change the owner and group of all files and directories in a folder named **my_folder** to **newuser** and **newgroup**, respectively:

-
-

○

```
sudo chown -R newuser:newgroup my_folder
```

This will change the ownership recursively.

Additional Tips:

- **Using sudo**: In order to change ownership, you typically need superuser privileges. This is why **sudo** is used.
- **Checking Ownership**: You can check the ownership of a file or directory with the **ls -l** command. The owner and group will be listed in the third and fourth columns, respectively.
- **Tab Completion**: When typing file or directory names, you can use tab completion to save time. For example, if you're changing ownership of a file named **my_file.txt**, you can type **sudo chown newuser my_** and then press **Tab** to auto-complete the rest.

Remember, be cautious when changing ownership, especially with the **-R** flag, as it can affect a large number of files and directories. Always double-check your commands to avoid unintended consequences.

# User & group management

User and group management in Linux involves creating, modifying, and deleting user accounts and groups. This is typically done through commands and utilities provided by the Linux system.

User Management:

1. Creating a User:

To create a new user, you can use the adduser or useradd command.

- Using adduser (with interactive prompts):
  ```
  sudo adduser username
  ```
- Using useradd (with options, less interactive):
  ```
  sudo useradd username
  ```

Example:

-
-

○

```
sudo adduser john_doe
```

2. Setting Password for a User:
To set or change the password for a user, you can use the passwd command.

```
sudo passwd username
```

Example:

```
sudo passwd john_doe
```

3. Modifying User Properties:
The usermod command allows you to modify user properties like username, home directory, shell, and more.

- To change the username:
  ```
  sudo usermod -l new_username old_username
  ```
- To change the home directory:
  ```
  sudo usermod -d /new_home_directory username
  ```
- To change the default shell:
  ```
  sudo usermod -s /path/to/new_shell username
  ```

Example (changing username):

```
sudo usermod -l jdoe john_doe
```

4. Deleting a User:
To delete a user, you can use the userdel command.

```
sudo userdel username
```

Example:

```
sudo userdel john_doe
```

Group Management:

1. Creating a Group:
To create a new group, you can use the addgroup or groupadd command.

- Using addgroup:
  ```
  sudo addgroup groupname
  ```
- Using groupadd:
  ```
  sudo groupadd groupname
  ```

  Example:

•

•

◦

```
sudo addgroup developers
```

2. Adding a User to a Group:
To add a user to an existing group, you can use the usermod command with the -aG option.

```
sudo usermod -aG groupname username
```

Example:

```
sudo usermod -aG developers jdoe
```

3. Changing a User's Primary Group:
To change a user's primary group, you can use the usermod command with the -g option.

```
sudo usermod -g groupname username
```

Example:

```
sudo usermod -g developers jdoe
```

4. Deleting a Group:
To delete a group, you can use the groupdel command.

```
sudo groupdel groupname
```

Example:

```
sudo groupdel developers
```

Additional Tips:

- **Listing Users and Groups**:
  - To list all users:
    bashCopy code
    cat /etc/passwd
  - To list all groups:
    bashCopy code
    cat /etc/group
- **Managing Passwords**:
  - You can use the passwd command without specifying a username to change your own password.
- **Be Cautious**:
  - When modifying user or group settings, be careful to avoid making changes that could lock you out of the system or cause unexpected behavior.
- **Backup Data**:

-

-

- ○

- When deleting users, make sure to back up any important data associated with the user account before proceeding.

Remember to replace "username" and "groupname" with actual usernames and group names when using these commands.

Top Linux Commands :

Navigation:

- ls: List files and directories. ○ Example: ls -l /path/to/directory
- cd: Change directory. ○ Example: cd /path/to/directory
- pwd: Print working directory.
    - Example: pwd
- mkdir: Create a directory.
    - Example: mkdir new_folder

File Operations:

- cp: Copy files or directories.
    - Example: cp file.txt /path/to/destination
- mv: Move or rename files or directories. ○ Example: mv file.txt new_name.txt

rm: Remove files or directories. ○ Example: rm file.txt touch: Create an empty file.

Example: touch new_file.txt

Text Manipulation:

-
-

○

- cat: Concatenate and display file content. ○ Example: cat file.txt
- grep: Search for a pattern in files. ○ Example: grep "pattern" file.txt
- sed: Stream editor for text manipulation.
  - Example: sed 's/old/new/' file.txt
- awk: Text processing tool.
  - Example: awk '{print $1}' file.txt

Process Management:

- ps: Display information about running processes. ○ Example: ps aux
- top: Display live system resource usage. ○ Example: top
- kill: Terminate processes by ID or name.
  - Example: kill PID

Package Management:

- apt: Advanced Package Tool (Debian-based systems). ○ Example: apt install package_name
- yum: Package manager (RHEL-based systems). ○ Example: yum install package_name
- dnf: Next-generation package manager (Fedora).
  - Example: dnf install package_name

Networking:

ping: Send ICMP echo requests to a host. ○ Example: ping google.com ifconfig / ip: Network interface configuration.

Example: ifconfig or ip addr

- netstat: Network statistics. ○ Example: netstat -tuln

•

•

o

- ssh: Secure shell for remote access. o Example: ssh username@hostname
- curl: Transfer data using URLs.
    - Example: curl https://example.com

Compression and Archiving:

- tar: Create and extract archive files. o Example: tar -cvf archive.tar files
- gzip / gunzip: Compress or decompress files.
    - Example: gzip file.txt

Disk Usage:

- df: Display disk space usage.
    - Example: df -h
- du: Display file and directory space usage.
    - Example: du -sh /path/to/directory

System Information:

- uname: Display system information.
    - Example: uname -a
- lsb_release: Display distribution-specific information.
    - Example: lsb_release -a
- uptime: Display system uptime.
    - Example: uptime

User and Group Management:

useradd: Add a new user. o
Example: useradd newuser
passwd: Change user password.

•

•

    -

      Example: passwd username
- groupadd: Add a new group.
    - Example: groupadd newgroup

Disk and Filesystem:

- fdisk: Manipulate disk partition table. ○ Example: fdisk /dev/sdX
- mount: Mount a filesystem. ○ Example: mount /dev/sdb1 /mnt
- umount: Unmount a filesystem.
    - Example: umount /mnt

File Searching:

- find: Search for files and directories. ○ Example: find /path/to/search -name "*.txt"
- locate: Quickly find files using a prebuilt index.
    - Example: locate file.txt

System Control:

- reboot: Reboot the system. ○ Example: reboot
- shutdown: Shutdown the system.
    - Example: shutdown -h now
- init: Change system runlevel (SysV init systems).
    - Example: init 3 (switch to text mode)

Text Editors:

- nano: Simple text editor.
  Example: nano file.txt
- vim / vi: Advanced text editor. Example: vim file.txt

File Transfer:

- scp: Securely copy files between hosts. ○ Example: scp file.txt remoteuser@remotehost:/path
- rsync: Efficiently transfer and synchronize files.
  - Example: rsync -av source/ destination/

Printing:

- lp: Print files.
  - Example: lp file.txt
- lpstat: Print queue status.
  - Example: lpstat -p

Monitoring:

- htop: Interactive process viewer.
  - Example: htop

Shell Features:

- history: Display command history. ○ Example: history
- !!: Execute the last command. ○ Example: !!
- Ctrl + R: Search command history.
  - Example: Press Ctrl + R, type search term, press Enter.

Remote Access:

- ssh: Secure shell for remote access. ○ Example: ssh user@host
- scp: Securely copy files between hosts.
  - Example: scp file.txt user@host:/path

File Permissions:

- chmod: Change file permissions. ○ Example: chmod 755 file.txt
- chown: Change file ownership.
    - Example: chown user:group file.txt

Compression and Archiving:

- tar: Create and extract archive files. ○ Example: tar -cvf archive.tar files
- gzip / gunzip: Compress or decompress files.
    - Example: gzip file.txt

Process Management:

- ps: Display information about running processes. ○ Example: ps aux
- kill: Terminate processes by ID or name.
    - Example: kill PID

Package Management:

- apt: Advanced Package Tool (Debian-based systems). ○ Example: apt install package_name
- yum: Package manager (RHEL-based systems). ○ Example: yum install package_name
- dnf: Next-generation package manager (Fedora).
    - Example: dnf install package_name

Networking:

- ping: Send ICMP echo requests to a host. ○ Example: ping google.com
- ifconfig / ip: Network interface configuration.
    - Example: ifconfig or ip addr

- netstat: Network statistics.
    - Example: netstat -tuln

Text Manipulation:

- cat: Concatenate and display file content. ○ Example: cat file.txt
- grep: Search for a pattern in files. ○ Example: grep "pattern" file.txt
- sed: Stream editor for text manipulation.
    - Example: sed 's/old/new/' file.txt
- awk: Text processing tool.
    - Example: awk '{print $1}' file.txt

Disk Usage:

- df: Display disk space usage.
    - Example: df -h
- du: Display file and directory space usage.
    - Example: du -sh /path/to/directory

System Information:

- uname: Display system information.
    - Example: uname -a
- lsb_release: Display distribution-specific information.
    - Example: lsb_release -a
- uptime: Display system uptime.
    - Example: uptime

User and Group Management:

- useradd: Add a new user. ○ Example: useradd newuser
- passwd: Change user password. ○ Example: passwd username
- groupadd: Add a new group.
    - Example: groupadd newgroup

Disk and Filesystem:

- fdisk: Manipulate disk partition table. ○ Example: fdisk /dev/sdX
- mount: Mount a filesystem. ○ Example: mount /dev/sdb1 /mnt
- umount: Unmount a filesystem.
    - Example: umount /mnt

File Searching:

- find: Search for files and directories. ○ Example: find /path/to/search -name "*.txt"
- locate: Quickly find files using a prebuilt index.
    - Example: locate file.txt

System Control:

- reboot: Reboot the system. ○ Example: reboot
- shutdown: Shutdown the system.
    - Example: shutdown -h now
- init: Change system runlevel (SysV init systems).
    - Example: init 3 (switch to text mode)

Text Editors:

- nano: Simple text editor. ○ Example: nano file.txt
- vim / vi: Advanced text editor.
    - Example: vim file.txt

File Transfer:

- scp: Securely copy files between hosts.
    - Example: scp file.txt remoteuser@remotehost:/path
- rsync: Efficiently transfer and synchronize files.
    - Example: rsync -av source/ destination/

Printing:

- lp: Print files.
    - Example: lp file.txt
- lpstat: Print queue status.
    - Example: lpstat -p

Please note that this is a comprehensive list, and you might not need to use all these commands in your daily tasks. It's also recommended to consult the respective command's manual pages (man command) for detailed information and options.