

Module 2: Linux Fundamentals

Abhijit Lalasaheb Zende

1. L1 - In EC2 Ubuntu Instance Create a new user and SSH Key pair with an authorized key

Ans.

(*** Note: Screen shots attached to end of each question ***)

Step 1: Launch EC2 Ubuntu instances

1. Launch the EC2 Instance:

- a. Log in to your AWS Management Console.
- b. Navigate to EC2 and launch a new Ubuntu instance.
- c. Configure the instance type, security group, and key pair.
- d. Note the public IP address of this instance.

Step 2: Connect to the First EC2 Instance:

Step 3: Create a New User on the EC2 Instance:

1. Switch to Root user:

Gain root privileges using command:

```
`sudo su`
```

2. Add a new user:

Create a new user using command:

```
`useradd newusername -s /bin/bash -m -d /home/newuserhomedir`
```

3. Set a password for the newly created user

Create a new password using command:

```
`passwd newusername`
```

Step 4: Create a new SSH key pair for new user:

1. Switch to the new user

```
`su - newuser`
```

2. Generate an SSH key pair:
`ssh-keygen -t ecdsa -b 521`
3. Save the key pair in the default location (/home/newuser/.ssh/)

Step 5: Authorize the SSH key:

1. Copy the public key to the authorized keys file:
`cat id_ecdsa.pub > authorized_keys`
2. Set appropriate permissions:
`chmod 600 /home/devopsadmin/.ssh/*`

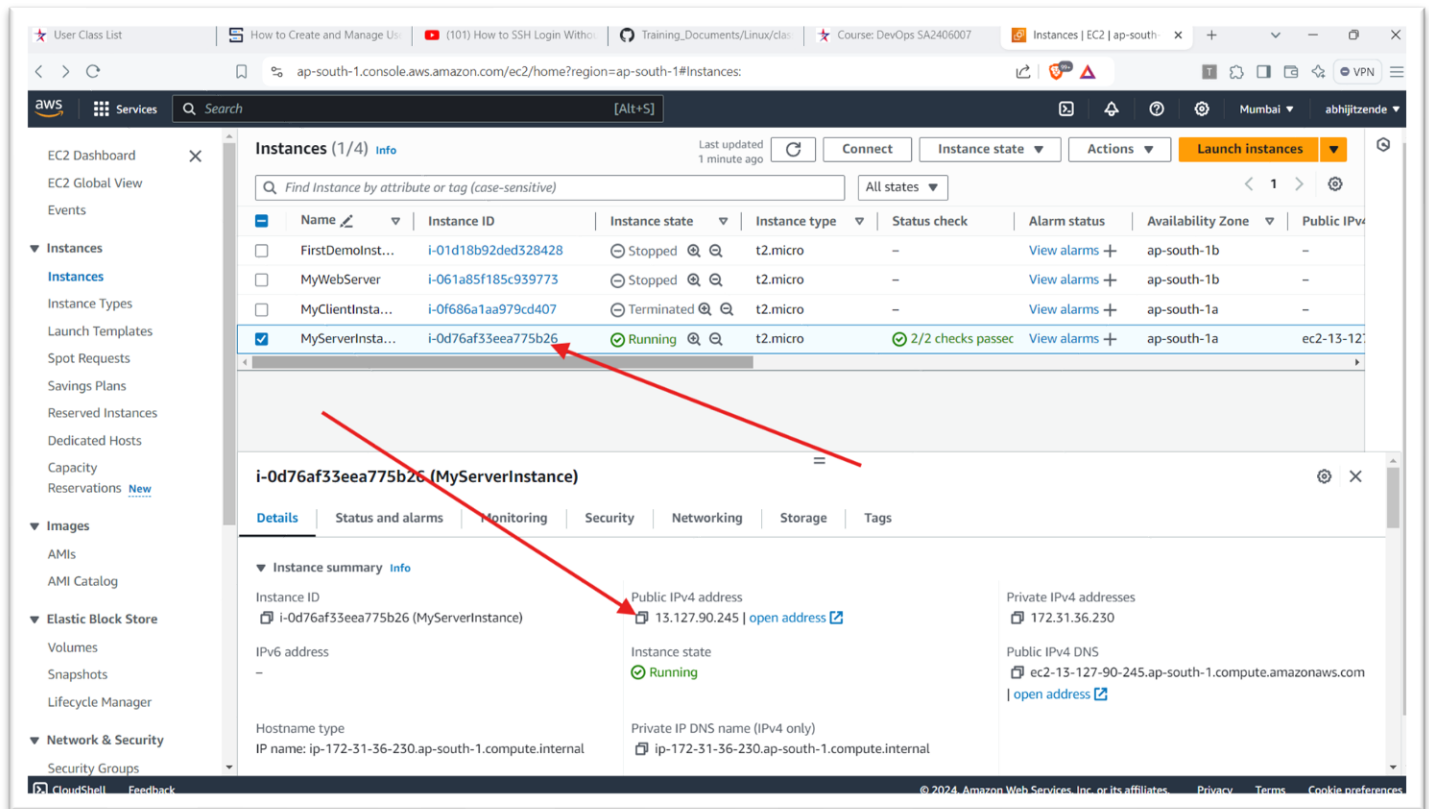


Fig. 1.01: New instance creation and copying its public IPv4 address for ssh

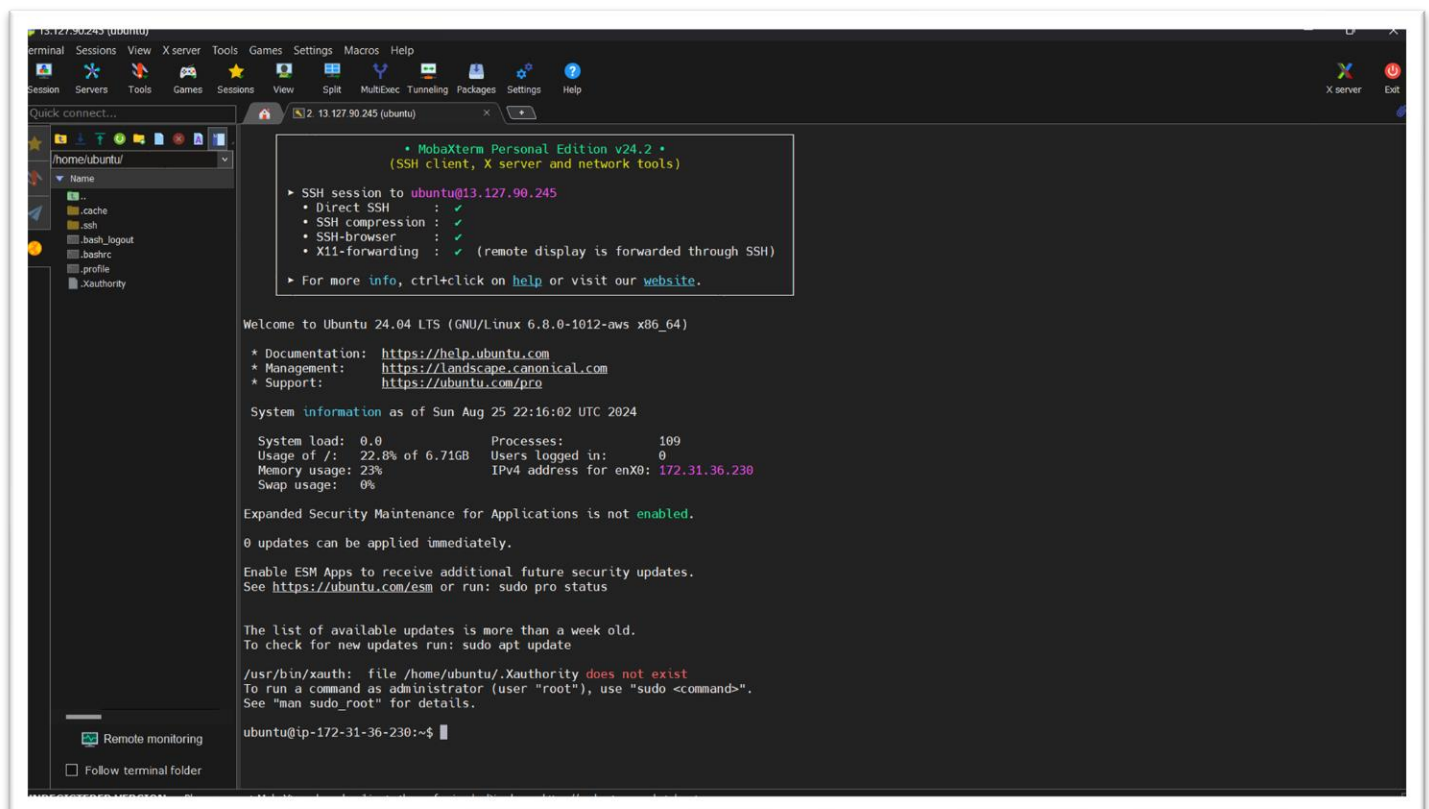


Fig. 1.02: Successfully connected to the instance using SSH through MobaXterm

2. L2 - As a Linux root user Create Files/Directory in the same Instance and change the ownership to a new user

Ans.

1. Connect to Your EC2 Instance as the Root User:

If not already connected, SSH into your EC2 instance as the root user or switch to root:

```
`sudo su`
```

2. Create Files and Directories:

a. Navigate to the desired directory (e.g., /home/newuser)

```
`cd /home/newuser`
```

b. Create a new file and directory:

```
`touch rootfile.txt`
```

```
`mkdir rootdirectory`
```

3. Change Ownership to the New User:

a. Change the ownership of the file and directory to the new user:

```
`chown newuser:newuser rootfile.txt`
```

```
`chown newuser:newuser rootdirectory`
```

b. Verify the change:

```
`ls -l`
```

The image shows a terminal window with a dark background. On the left, there is a sidebar with a file explorer view showing the contents of the /home/ubuntu directory, including files like .cache, .ssh, .bash_history, .bash_logout, .bashrc, .profile, .sudo_as_admin_successful, and .Xauthority. The main terminal area displays a series of commands and their outputs. Red arrows point to specific parts of the terminal output: the 'cd' command, the 'touch' command, the 'mkdir' command, the 'ls -l' command, the 'chown' command, and the final 'ls -l' command. The terminal output shows the user switching to root with 'sudo su', then navigating to /home/devopsadmin/. The user then creates 'rootfile.txt' and 'rootdirectory'. A subsequent 'ls -l' command shows the permissions and ownership of these files. Finally, the 'chown' command is used to change the ownership of both files to devopsadmin:devopsadmin, and a final 'ls -l' command confirms the change.

```
ubuntu@ip-172-31-36-230:~$ sudo su
root@ip-172-31-36-230:/home/ubuntu# cd /home/devopsadmin/
root@ip-172-31-36-230:/home/devopsadmin# touch rootfile.txt
root@ip-172-31-36-230:/home/devopsadmin# mkdir rootdirectory
root@ip-172-31-36-230:/home/devopsadmin# ls -l
total 4
drwxr-xr-x 2 root root 4096 Aug 25 23:28 rootdirectory
-rw-r--r-- 1 root root 0 Aug 25 23:27 rootfile.txt
root@ip-172-31-36-230:/home/devopsadmin# chown devopsadmin:devopsadmin rootfile.txt
root@ip-172-31-36-230:/home/devopsadmin# chown devopsadmin:devopsadmin rootdirectory
root@ip-172-31-36-230:/home/devopsadmin# ls -l
total 4
drwxr-xr-x 2 devopsadmin devopsadmin 4096 Aug 25 23:28 rootdirectory
-rw-r--r-- 1 devopsadmin devopsadmin 0 Aug 25 23:27 rootfile.txt
root@ip-172-31-36-230:/home/devopsadmin#
```

Fig. 2.01: Creating a file and directory as root user and then changing its ownership

3. L3 - In EC2 Ubuntu Instance Create Files and Directories and Grand R/W/X Access only to the Owner and User Group

Ans.

1. Connect to the EC2 instance as new user:

SSH into your instance using the new user or switch to the new user:

```
`su - newuser`
```

2. Create Files and Directories:

Create a file and directory in the user's home directory:

```
`touch userfile.txt`
```

```
`mkdir userdirectory`
```

3. Set Permissions:

Grant Read/Write/Execute access only to the owner and group:

```
`chmod 770 userfile.txt`
```

```
`chmod 770 userdirectory`
```

4. Verify the permissions:

```
`ls -l`
```

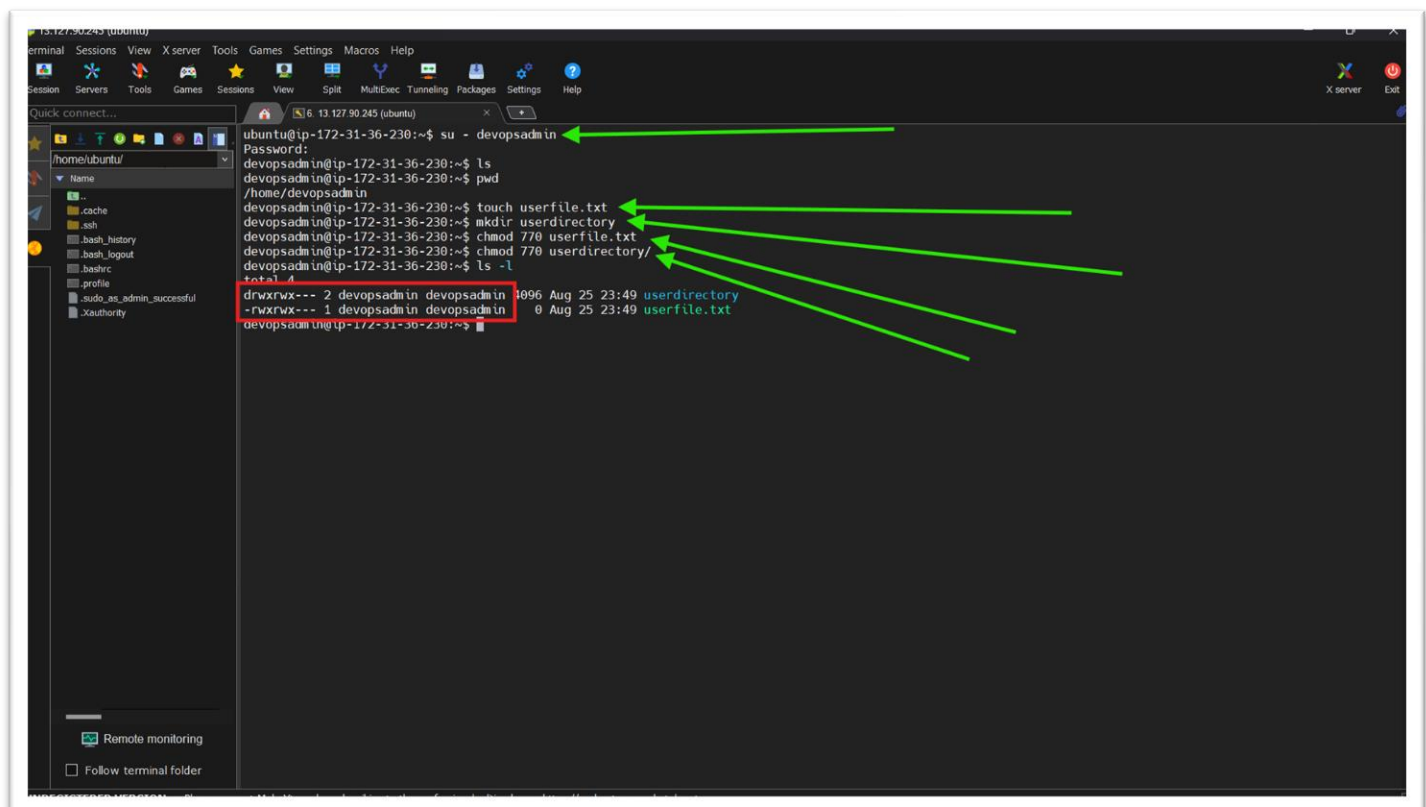


Fig. 3.01: Grant Read/Write/Execute Access Only to Owner and User Group

4. L4 - In EC2 Ubuntu Instance install JDK and setup JAVA_HOME path environment variable

Ans.

1. Update the Package List:

Connect to your EC2 instance and update the package list

```
`sudo apt update`
```

2. Install JDK:

Install the OpenJDK package:

```
`sudo apt install openjdk-11-jdk -y`
```

3. Verify the Installation:

Check the installed Java version:

```
`java -version`
```

4. Set Up JAVA_HOME Environment Variable:

a) Find the Java installation path:

```
`sudo update-alternatives --config java`
```

b) Set the JAVA_HOME variable:

```
`echo "export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64" >> ~/.bashrc`
```

```
`echo "export PATH=\$JAVA_HOME/bin:\$PATH" >> ~/.bashrc`
```

```
`source ~/.bashrc`
```

5. Verify the JAVA_HOME Variable:

Check the environment variable:

```
`echo $JAVA_HOME`
```



```
ubuntu@ip-172-31-36-230:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Fetched 126 kB in 1s (235 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
86 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-36-230:~$ sudo apt install openjdk-11-jdk -y
Reading package lists... Done
Building dependency tree... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig-config
  fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono gsettings-desktop-schemas java-common libasound2-data libasound2t64 libatk-bridge2.0-0t64
  libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64 libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdconf1
  libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libfontconfig1 libglib2.0-0t64 libgl1 libgl1-mesa-dri libglapi-mesa libglvnd0
  libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b libice-dev libice6 libjpeg-turbo8 libjpeg8 liblcms2-2 libllvm17t64 libpciaccess0 libpcsclite1
  libpthread-stubs0-dev libsm-dev libsm6 libvulkan1 libwayland-client0 libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0
  libxcb-present0 libxcb-randr0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xf86vm0 libxcb1-dev libxcomposite1 libxdmcp-dev libxfixes3 libxft2 libxi6
  libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt-dev libxt6t64 libxtst6 libxv1 libxxf86dga1 libxxf86vm1
  mesa-vulkan-drivers openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless session-migration x11-common x11-utils x11proto-dev xorg-sgml-doctools
  xtrans-dev
Suggested packages:
  default-jre alsa-utils libasound2-plugins cups-common libice-doc liblcms2-utils pcsd libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo
  openjdk-11-source visualvm libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic mesa-utils
Recommended packages:
  luit
The following NEW packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig-config
  fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono gsettings-desktop-schemas java-common libasound2-data libasound2t64 libatk-bridge2.0-0t64
  libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64 libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdconf1
  libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libfontconfig1 libglib2.0-0t64 libgl1 libgl1-mesa-dri libglapi-mesa libglvnd0
  libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b libice-dev libice6 libjpeg-turbo8 libjpeg8 liblcms2-2 libllvm17t64 libpciaccess0 libpcsclite1
  libpthread-stubs0-dev libsm-dev libsm6 libvulkan1 libwayland-client0 libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0
  libxcb-present0 libxcb-randr0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xf86vm0 libxcb1-dev libxcomposite1 libxdmcp-dev libxfixes3 libxft2 libxi6
  libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt-dev libxt6t64 libxtst6 libxv1 libxxf86dga1 libxxf86vm1
  mesa-vulkan-drivers openjdk-11-jdk openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless session-migration x11-common x11-utils x11proto-dev
  xorg-sgml-doctools xtrans-dev
0 upgraded, 96 newly installed, 0 to remove and 86 not upgraded.
Need to get 177 MB of archives.
After this operation, 521 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 alsa-topology-conf all 1.2.5.1-2 [15.5 kB]
```

Fig. 4.01: Updating package manager and installing jdk

```
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jmod to provide /usr/bin/jmod (jmod) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jps to provide /usr/bin/jps (jps) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jrunscript to provide /usr/bin/jrunscript (jrunscript) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jshell to provide /usr/bin/jshell (jshell) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstack to provide /usr/bin/jstack (jstack) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstat to provide /usr/bin/jstat (jstat) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstatd to provide /usr/bin/jstatd (jstatd) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/rmic to provide /usr/bin/rmic (rmic) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/serialver to provide /usr/bin/serialver (serialver) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jaotc to provide /usr/bin/jaotc (jaotc) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jhsdb to provide /usr/bin/jhsdb (jhsdb) in auto mode
Setting up openjdk-11-jdk:amd64 (11.0.24+8-1ubuntu3-24.04.1) ...
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-36-230:~$ java -version
openjdk version "11.0.24" 2024-07-16
OpenJDK Runtime Environment (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1)
OpenJDK 64-Bit Server VM (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1, mixed mode, sharing)
ubuntu@ip-172-31-36-230:~$ sudo update-alternatives --config java
There is 1 choice for the alternative java (providing /usr/bin/java).

  Selection    Path                                          Priority  Status
  -----
  * 0          /usr/lib/jvm/java-11-openjdk-amd64/bin/java 1111    auto mode
  1           /usr/lib/jvm/java-11-openjdk-amd64/bin/java 1111    manual mode

Press <enter> to keep the current choice[*], or type selection number:
ubuntu@ip-172-31-36-230:~$ echo ""
ubuntu@ip-172-31-36-230:~$ echo "export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64" >> ~/.bashrc
ubuntu@ip-172-31-36-230:~$ echo "export PATH=$JAVA_HOME/bin:$PATH" >> ~/.bashrc
ubuntu@ip-172-31-36-230:~$ source ~/.bashrc
ubuntu@ip-172-31-36-230:~$ echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64
ubuntu@ip-172-31-36-230:~$
```

Fig. 4.02: Setting Java environment variable path

```
ubuntu@ip-172-31-36-230:~$ java -version
openjdk version "11.0.24" 2024-07-16
OpenJDK Runtime Environment (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1)
OpenJDK 64-Bit Server VM (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1, mixed mode, sharing)
ubuntu@ip-172-31-36-230:~$ sudo update-alternatives --config java
There is 1 choice for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority  Status
  ----
* 0            /usr/lib/jvm/java-11-openjdk-amd64/bin/java    1111     auto mode
  1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java    1111     manual mode

Press <enter> to keep the current choice[*], or type selection number:
ubuntu@ip-172-31-36-230:~$ echo ""
ubuntu@ip-172-31-36-230:~$ echo "export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64" >> ~/.bashrc
ubuntu@ip-172-31-36-230:~$ echo "export PATH=\$JAVA_HOME/bin:\$PATH" >> ~/.bashrc
ubuntu@ip-172-31-36-230:~$ source ~/.bashrc
ubuntu@ip-172-31-36-230:~$ echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64
ubuntu@ip-172-31-36-230:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-36-230:~$ ls
ubuntu@ip-172-31-36-230:~$ mkdir sample_java_program
ubuntu@ip-172-31-36-230:~$ cd sample_java_program/
ubuntu@ip-172-31-36-230:~/sample_java_program$ vim hello.java
ubuntu@ip-172-31-36-230:~/sample_java_program$ cat hello.java
// HelloWorld.java
public class HelloWorld {
    public static void main(String[] args) {
        // Prints "Hello, World!" to the terminal window
        System.out.println("Hello, World!");
    }
}
ubuntu@ip-172-31-36-230:~/sample_java_program$ javac hello.java
hello.java:2: error: class HelloWorld is public, should be declared in a file named HelloWorld.java
public class HelloWorld {
      ^
1 error
ubuntu@ip-172-31-36-230:~/sample_java_program$ mv hello.java HwllowWorld.java
ubuntu@ip-172-31-36-230:~/sample_java_program$ mv HwllowWorld.java HelloWorld.java
ubuntu@ip-172-31-36-230:~/sample_java_program$ javac HelloWorld.java
ubuntu@ip-172-31-36-230:~/sample_java_program$ java HelloWorld
Hello, World!
ubuntu@ip-172-31-36-230:~/sample_java_program$
```

Fig. 4.03: Verifying the result by compiling and running a simple java program

5. L5 - Create two AWS EC2 Ubuntu Instances to establish SSH Connection and SCP the files from one Instance to another instance

Ans.

1. Launch yet another EC2 ubuntu instance
 - a. Launch yet another ubuntu EC2 instance from aws dashboard other than what we launched in the L1 task
 - b. Here the instance we made in the L1 task will act as server instance
 - c. The instance that we will launch will now act as client instance
2. Connect to Server(first) EC2 instance and copy it's public key:
 - a. Connect to the server(first) EC2 instance
 - b. Login as the newly added user previously in L1 using command:
``su - newusername``
 - c. After login change directory to ``.ssh`` using command:
``cd .ssh/``
 - d. Copy the public key we earlier generated in task L1 during SSH key pair generation using command:
``cat id_ecdsa.pub``
Above command will display the public key. Copy manually
3. Connect to Client(second) EC2 instance and create the authorized key:
 - a. Connect to Client(second) EC2 instance
 - b. Create a new user using command:
``sudo useradd username -s /bin/bash -m -d /home/username``
 - c. Also add password to the user using command:
``sudo passwd username``
 - d. Change user to the newly created user using command:
``su - username``
 - e. Create a new ``.ssh`` special directory using command:
``mkdir .ssh``
 - f. Change directory to ``.ssh/`` using command:
``cd .ssh/``
 - g. Create and open the ``.ssh/authorized_keys`` file using command:
``vi authorized_keys``
 - h. Paste the public key in the text editor, save and exit using
``Shift+i`` = insert mode
``Esc`` = To escape out of insert mode

`:wq = to save and exit`

- i. Change permission of the 'ssh' directory using command:

`chmod 600 /home/username/.ssh/*`

4. Test authorized key SSH connection from server to client instance:

- a. Logout of client session
- b. Login as new user in the server instance or using command in server instance:

`su - username`

- c. SSH from server to client instance using command:

`ssh username@public-ipv4-address-of-client-instance`

- d. If not sure then enter below command to know the current user

`whoami`

5. SCP file from server instance to client instance:

- a. Logout of client session using command

`exit`

- b. Change directory to '/home/username' using command

`cd /home/devopsadmin`

- c. Create a new file using command:

`touch filename1.txt`

- d. Copy the file from server instance to client instance using scp command:

`scp /path/to/file/in/server/instance username@public-ipv4-of-instance:/home/userdirectory`

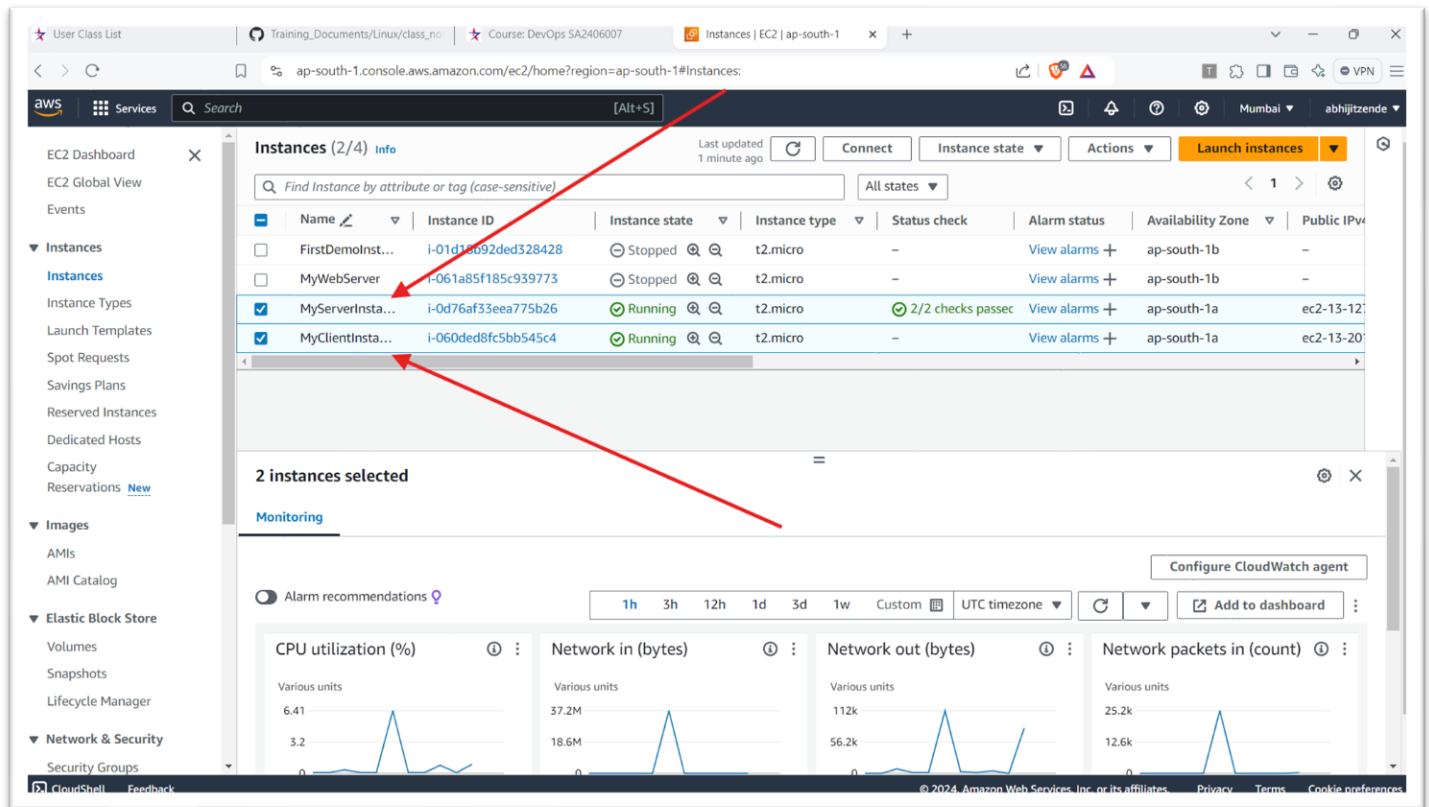


Fig. 5.01: Launch 2 EC2 instance

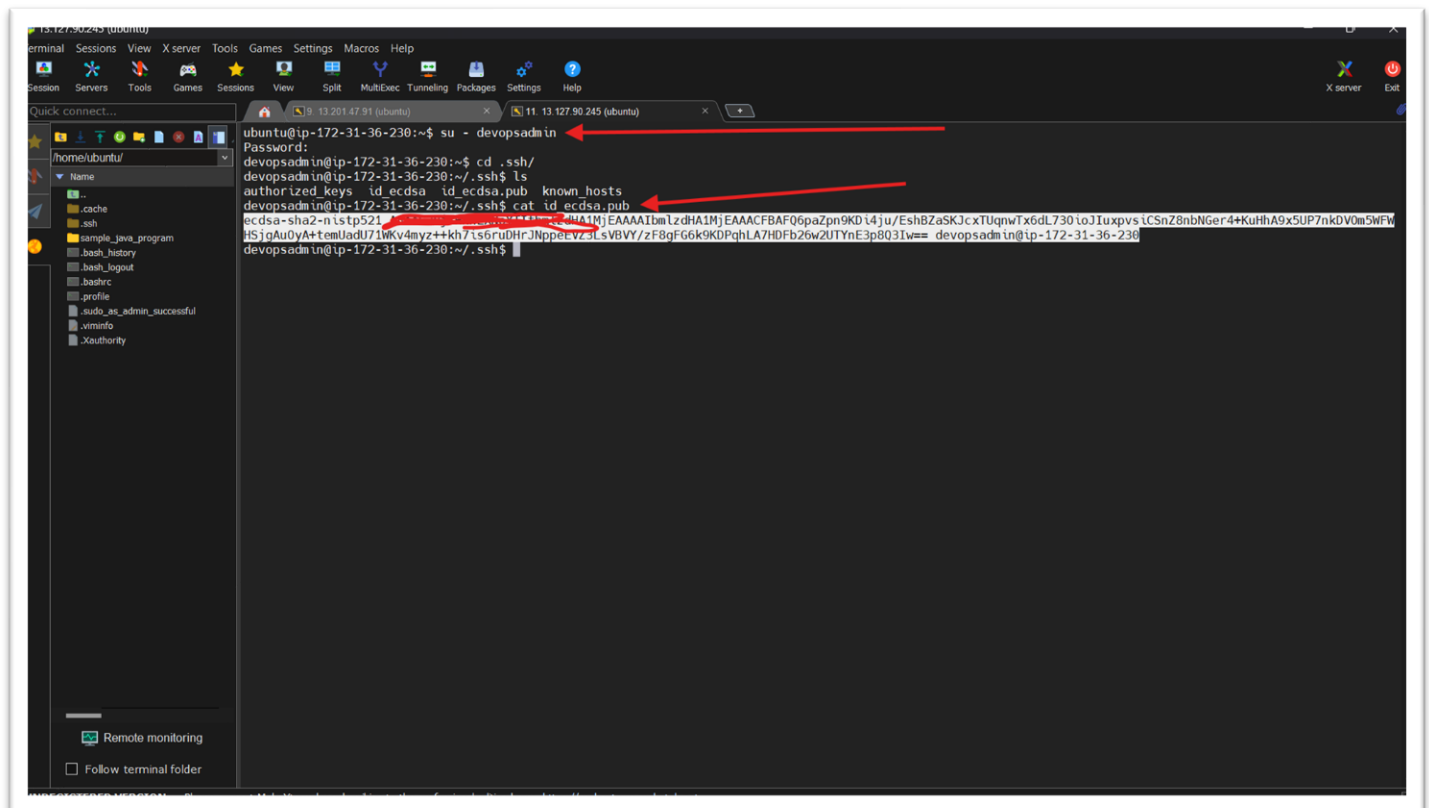


Fig. 5.02: Copy public key from server instance

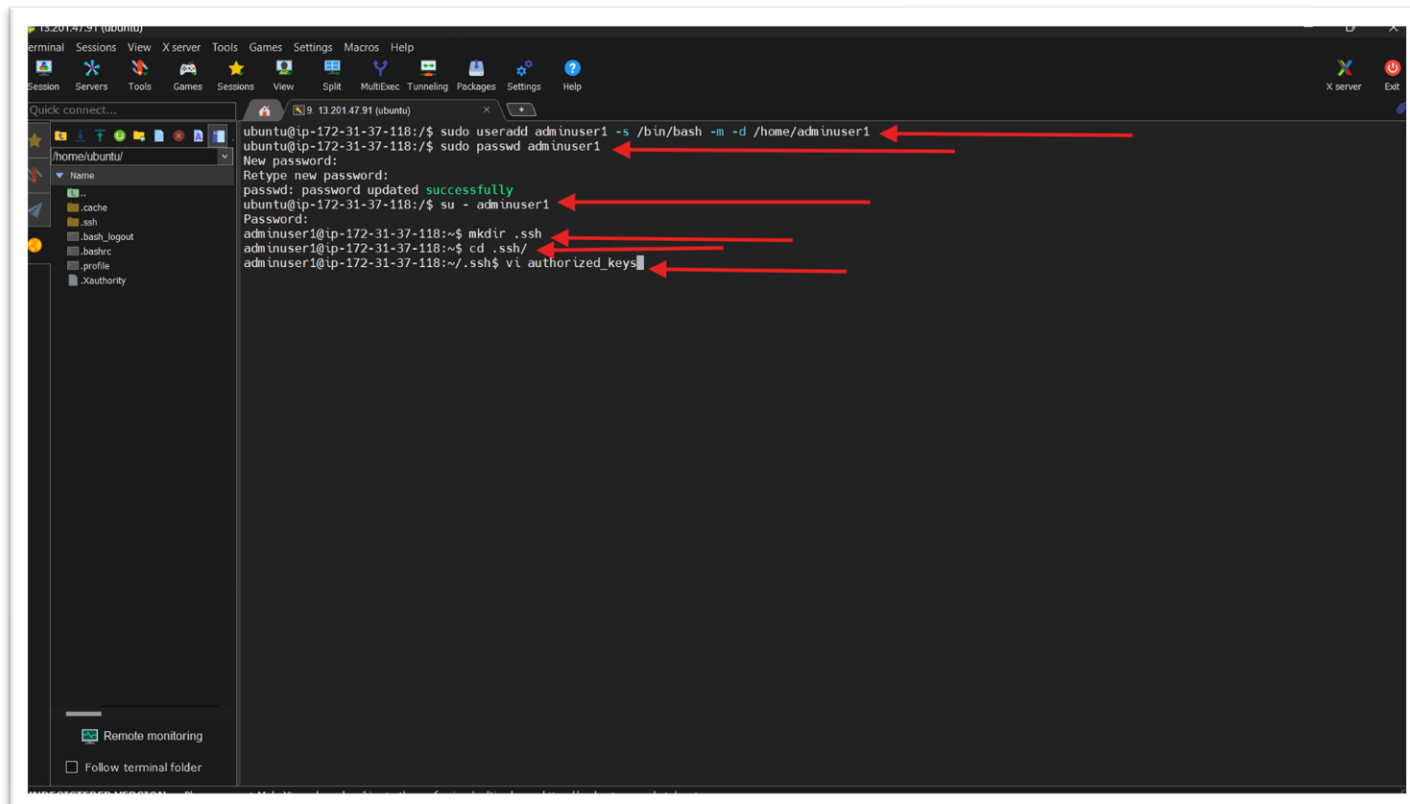


Fig. 5.03: Create new user in the client instance

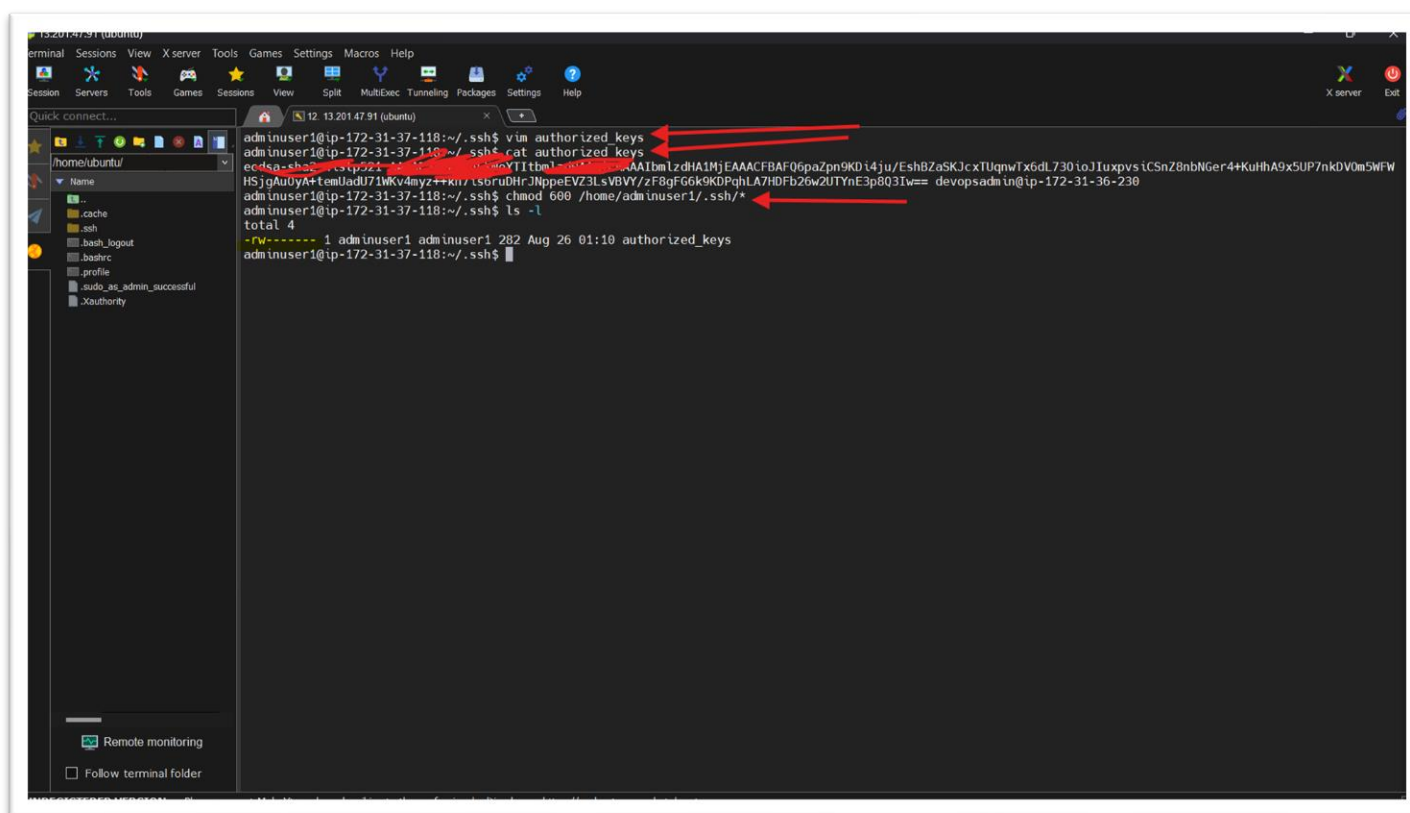


Fig. 5.04: Paste public key of server instance into authorized_keys of client instance user

The terminal window shows a user logging into a server instance. The user runs `su - devopsadmin` to switch to the `devopsadmin` user. Then, they run `ssh adminuser1@13.201.47.91` to establish an SSH connection to the client instance. The terminal displays the SSH warning about the host's fingerprint, the user's confirmation to continue, and the system information for the client instance. The system information shows the system load, memory usage, and the list of available updates. The user then runs `whoami` to verify the connection, and the output is `adminuser1`.

```
ubuntu@ip-172-31-36-230:~$ su - devopsadmin
devopsadmin@ip-172-31-36-230:~$ ssh adminuser1@13.201.47.91
The authenticity of host '13.201.47.91 (13.201.47.91)' can't be established.
ED25519 key fingerprint is SHA256:mkUp6wF9v6u7n6J+MzvCYwqHWxAc8pFPhZbe4WuW0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.201.47.91' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Aug 26 01:15:01 UTC 2024

System load:  0.0      Processes:    118
Usage of /:   23.0% of 6.71GB   Users logged in: 1
Memory usage: 22%      IPv4 address for enX0: 172.31.37.118
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

adminuser1@ip-172-31-37-118:~$ whoami
adminuser1
adminuser1@ip-172-31-37-118:~$
```

Fig. 5.05: Test and verify the SSH connection through `authorized_keys`

The terminal window shows the user logging out of the client instance and then running `scp` to copy a file from the server instance to the client instance. The user runs `exit` to log out of the client instance. Then, they run `cd /home/devopsadmin/` to navigate to the `/home/devopsadmin/` directory. They then run `touch mywebappfile1.txt` to create a new file. Next, they run `ls` to list the files in the directory. The output shows `mywebappfile1.txt` and `userdirectory`. Then, they run `scp /home/devopsadmin/mywebappfile1.txt adminuser1@13.201.47.91:/home/adminuser1/` to copy the file to the client instance. The terminal displays the progress of the file transfer. The user then runs `ssh adminuser1@13.201.47.91` to establish an SSH connection to the client instance. The terminal displays the SSH warning about the host's fingerprint, the user's confirmation to continue, and the system information for the client instance. The user then runs `whoami` to verify the connection, and the output is `adminuser1`. They then run `cd /home/adminuser1/` to navigate to the `/home/adminuser1/` directory. Finally, they run `ls` to list the files in the directory, and the output shows `mywebappfile1.txt`.

```
adminuser1@ip-172-31-37-118:~$ whoami
adminuser1
adminuser1@ip-172-31-37-118:~$ exit
Logout
Connection to 13.201.47.91 closed.
devopsadmin@ip-172-31-36-230:~$ cd /home/devopsadmin/
devopsadmin@ip-172-31-36-230:~$ touch mywebappfile1.txt
devopsadmin@ip-172-31-36-230:~$ ls
mywebappfile1.txt  userdirectory  userfile.txt
devopsadmin@ip-172-31-36-230:~$ scp /home/devopsadmin/mywebappfile1.txt adminuser1@13.201.47.91:/home/adminuser1/
mywebappfile1.txt
devopsadmin@ip-172-31-36-230:~$ ssh adminuser1@13.201.47.91
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Aug 26 01:19:18 UTC 2024

System load:  0.0      Processes:    119
Usage of /:   23.0% of 6.71GB   Users logged in: 1
Memory usage: 21%      IPv4 address for enX0: 172.31.37.118
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Aug 26 01:15:02 2024 from 13.127.90.245
adminuser1@ip-172-31-37-118:~$ whoami
adminuser1
adminuser1@ip-172-31-37-118:~$ cd /home/adminuser1/
adminuser1@ip-172-31-37-118:~$ ls
mywebappfile1.txt
adminuser1@ip-172-31-37-118:~$
```

Fig. 5.06: Copy actual files from server instance to client instance using `scp` command

6. L6 - Write a Linux Shell Script to Install Git, JDK, Maven in EC2 Ubuntu Instance

Ans.

1. Create a Shell Script:

On the first EC2 instance, create a new script file:

```
`vim install_tools.sh`
```

2. Add Commands to the Script:

Add the following content:

```
``
```

```
#!/bin/bash
```

```
# Update package list
```

```
sudo apt update -y
```

```
# Install Git
```

```
sudo apt install git -y
```

```
# Install OpenJDK
```

```
sudo apt install openjdk-11-jdk -y
```

```
# Install Maven
```

```
sudo apt install maven -y
```

```
# Verify installations
```

```
git --version
```

```
java -version
```

```
mvn -version
```

```
``
```

3. Make the Script Executable:

Save the file and make the script executable:

```
`chmod +x install_tools.sh`
```

4. Run the Script:

Execute the script to install the required tools:

```
`./install_tools.sh`
```


5. Verify Installations:

Check the installed versions of Git, Java, and Maven:

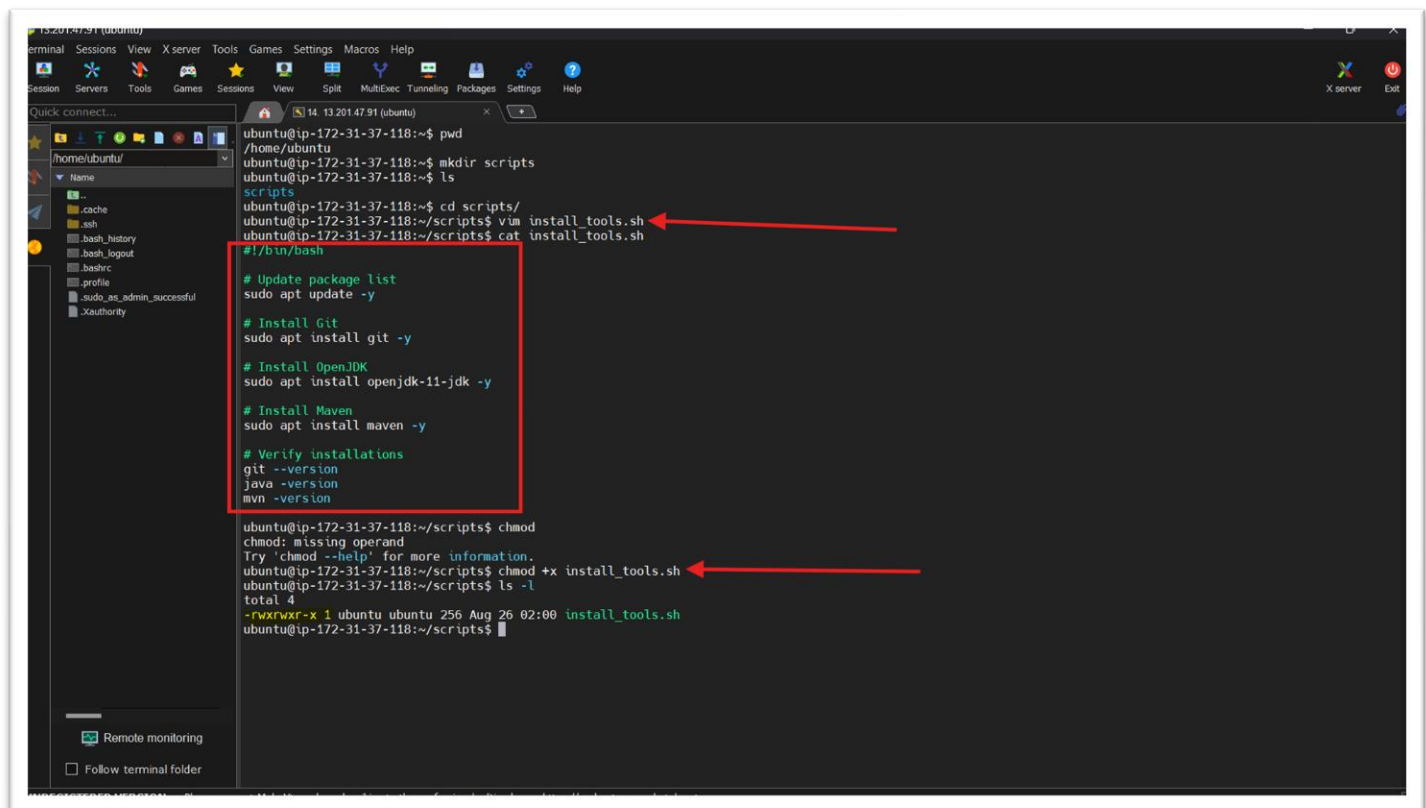
```

git --version

java -version

mvn -version

```



The screenshot shows a terminal window with the following commands and output:

```
ubuntu@ip-172-31-37-118:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-37-118:~$ mkdir scripts
ubuntu@ip-172-31-37-118:~$ ls
scripts
ubuntu@ip-172-31-37-118:~$ cd scripts/
ubuntu@ip-172-31-37-118:~/scripts$ vim install_tools.sh
ubuntu@ip-172-31-37-118:~/scripts$ cat install_tools.sh

#!/bin/bash

# Update package list
sudo apt update -y

# Install Git
sudo apt install git -y

# Install OpenJDK
sudo apt install openjdk-11-jdk -y

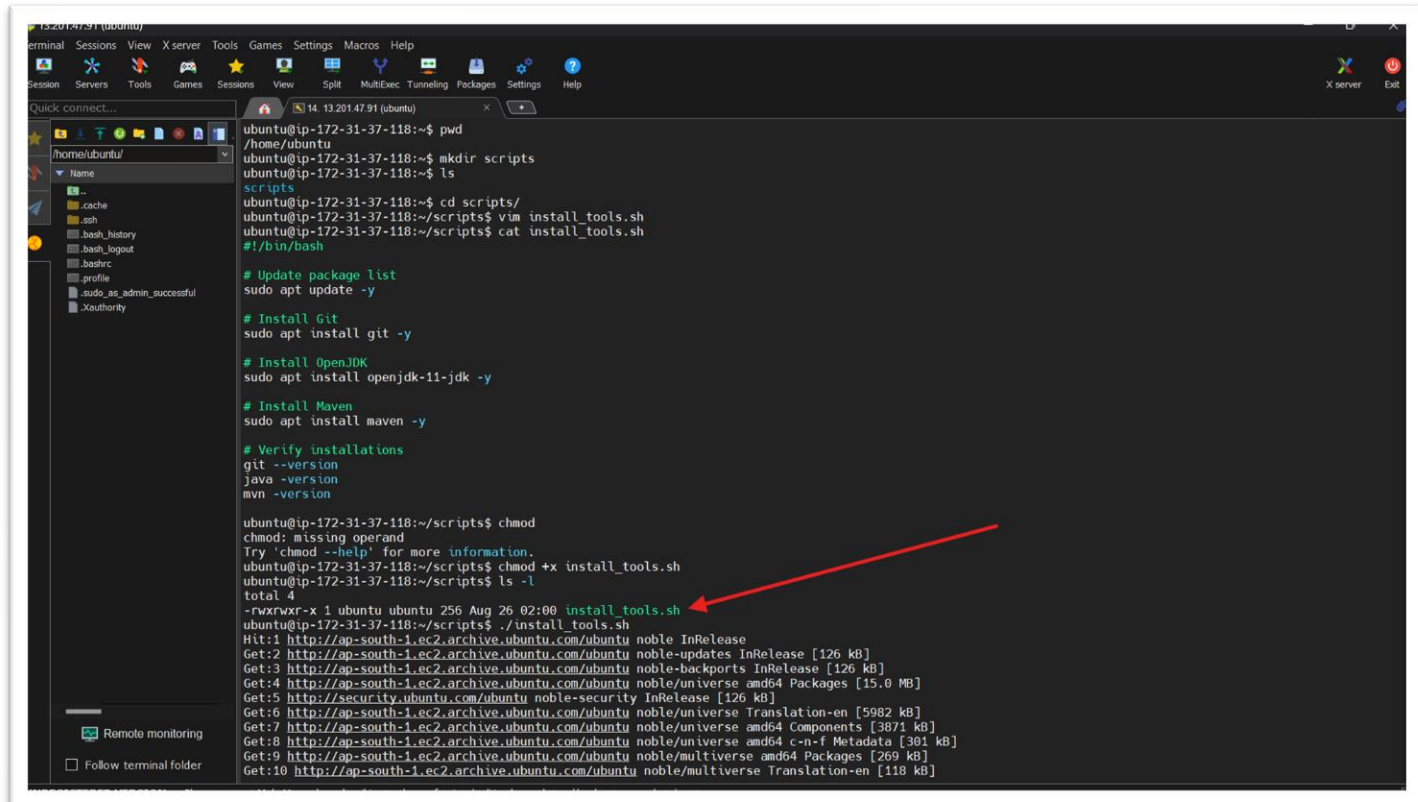
# Install Maven
sudo apt install maven -y

# Verify installations
git --version
java -version
mvn -version

ubuntu@ip-172-31-37-118:~/scripts$ chmod
chmod: missing operand
Try 'chmod --help' for more information.
ubuntu@ip-172-31-37-118:~/scripts$ chmod +x install_tools.sh
ubuntu@ip-172-31-37-118:~/scripts$ ls -l
total 4
-rwxrwxr-x 1 ubuntu ubuntu 256 Aug 26 02:00 install_tools.sh
ubuntu@ip-172-31-37-118:~/scripts$
```

Two red arrows point to the `vim install_tools.sh` and `chmod +x install_tools.sh` commands in the terminal output.

Fig. 6.01: Writing script



```
ubuntu@ip-172-31-37-118:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-37-118:~$ mkdir scripts
ubuntu@ip-172-31-37-118:~$ ls
scripts
ubuntu@ip-172-31-37-118:~$ cd scripts/
ubuntu@ip-172-31-37-118:~/scripts$ vim install_tools.sh
ubuntu@ip-172-31-37-118:~/scripts$ cat install_tools.sh
#!/bin/bash

# Update package list
sudo apt update -y

# Install Git
sudo apt install git -y

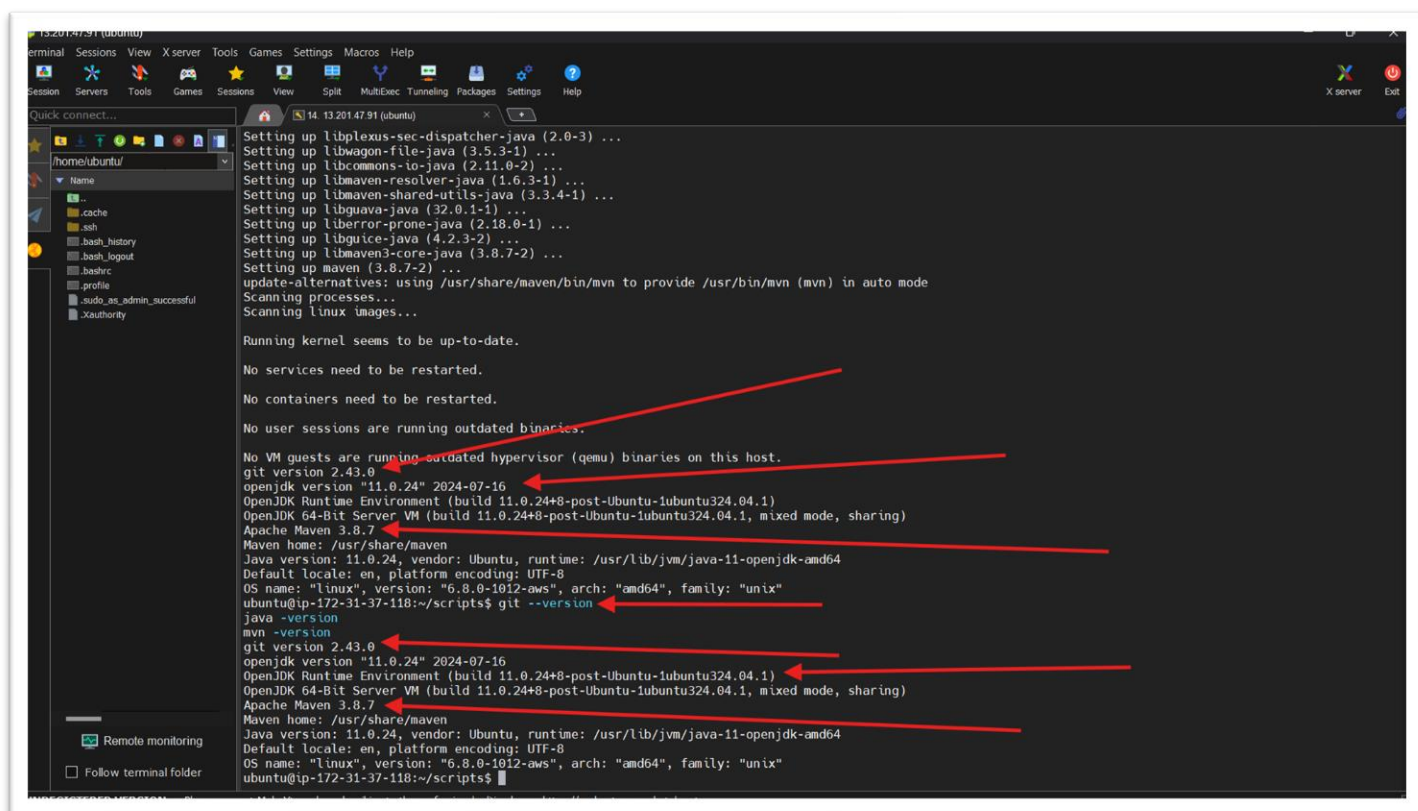
# Install OpenJDK
sudo apt install openjdk-11-jdk -y

# Install Maven
sudo apt install maven -y

# Verify installations
git --version
java -version
mvn -version

ubuntu@ip-172-31-37-118:~/scripts$ chmod
chmod: missing operand
Try 'chmod --help' for more information.
ubuntu@ip-172-31-37-118:~/scripts$ chmod +x install_tools.sh
ubuntu@ip-172-31-37-118:~/scripts$ ls -l
total 4
-rwxrwxr-x 1 ubuntu ubuntu 256 Aug 26 02:00 install_tools.sh
ubuntu@ip-172-31-37-118:~/scripts$ ./install_tools.sh
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.9 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
```

Fig. 6.02: Running script



```
Setting up libplexus-sec-dispatcher-java (2.0-3) ...
Setting up libwagon-file-java (3.5.3-1) ...
Setting up libcommons-io-java (2.11.0-2) ...
Setting up libmaven-resolver-java (1.6.3-1) ...
Setting up libmaven-shared-utils-java (3.3.4-1) ...
Setting up libguava-java (32.0.1-1) ...
Setting up liberror-prone-java (2.18.0-1) ...
Setting up libguice-java (4.2.3-2) ...
Setting up libmaven3-core-java (3.8.7-2) ...
Setting up maven (3.8.7-2) ...
update-alternatives: using /usr/share/maven/bin/mvn to provide /usr/bin/mvn (mvn) in auto mode
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
git version 2.43.0
openjdk version "11.0.24" 2024-07-16
OpenJDK Runtime Environment (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1)
OpenJDK 64-Bit Server VM (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1, mixed mode, sharing)
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 11.0.24, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1012-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-37-118:~/scripts$ git --version
git version 2.43.0
java -version
openjdk version "11.0.24" 2024-07-16
OpenJDK Runtime Environment (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1)
OpenJDK 64-Bit Server VM (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1, mixed mode, sharing)
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 11.0.24, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1012-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-37-118:~/scripts$ mvn -version
mvn version 3.8.7
openjdk version "11.0.24" 2024-07-16
OpenJDK Runtime Environment (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1)
OpenJDK 64-Bit Server VM (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1, mixed mode, sharing)
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 11.0.24, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1012-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-37-118:~/scripts$
```

Fig. 6.03: Verifying installation