Expert Cloud Consulting

Enhance Optimise & Scale

ASCP GPUonCLOUD Pvt Ltd

“Expert Cloud Consulting” -

SOP | Operating Systems and Networking Basics

09.August.2024

version 1.0

—

Contributed by Dhanshri Partil
Approved by   Akshay Shide(In Review)
Expert Cloud Consulting
Office #811, Gera Imperium Rise,
Hinjewadi Phase-II Rd, Pune, India – 411057

“Expert Cloud Consulting”
Setup AWS Linux Server on EC2 [Title,18, Arial]

1.0 Contents [Heading3,14, Arial]

1.0 Contents [Heading3,14, Arial] 1
2.0 General Information: [Heading3,14, Arial] 2
2.1 Document Jira/ Github Ticket(s) [Heading4,12, Arial] 2
2.2 Document Purpose 2
2.3 Document Revisions 2
2.4 Document References 2
3.0 Document Overview: 4
4.0 Steps / Procedure 5
4.1 : Setup the ubuntu server for .net core application 5
4.2: Choose an Amazon Machine Image (AMI) 5
4.3: Key-Pair Configuration 6
4.4: Network settings 6
4.3.1: VPC Configuration: 6
4.3.2: Security Group Configuration 7
4.4: Launch Instance 7
4.5: SSH Configuration 8
4.6: Install SSM Agent on Server 9

2.0 General Information:

2.1 Document Jira/ Github Ticket(s)

Ticket(s) Name	Url
Setup the linux Server on EC2 **[Normal text,10, Arial]**	Jira / github url

2.2 Document Purpose

This manual lays out the processes and guidelines for setting up the Ubuntu linux operating system for the .Net core application on aws EC2 instance. **[Normal text,10, Arial, Justify Alignment]**

2.3 Document Revisions

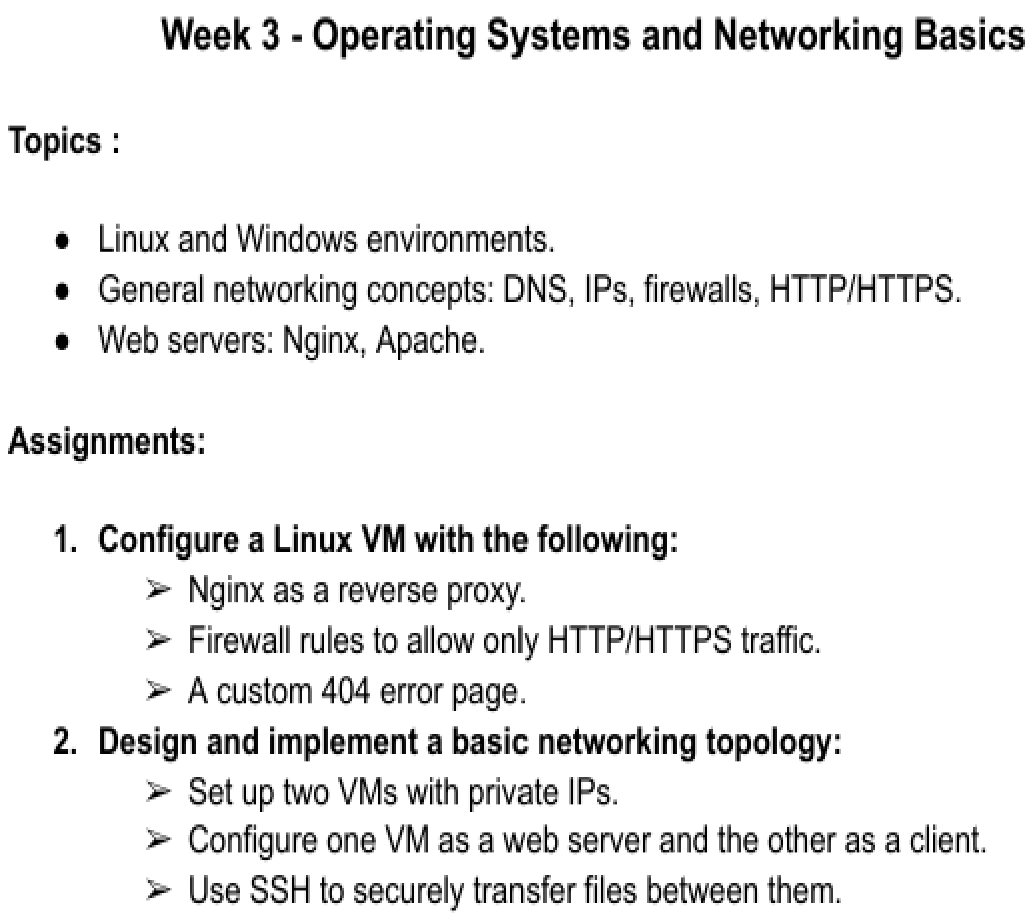
Date	Version	Contributor(s)	Approver(s)	Section(s)	Change(s)
09/Aug/2024	1.0	Atul Kumbhar	Atul Kumbhar	All Sections	New Document Created

2.4 Document References

The following artifacts are referenced within this document. Please refer to the original documents for additional information.

Date	Document	Filename / Url
2021	Setup AWS Linux Server	https://linux.how2shout.com/how-to-create-a-ubuntu-linux-aws-ec2-instance-on-amazon-cloud/
2020	How to Create a Linux 20.04 Server on AWS EC2	https://medium.com/nerd-for-tech/how-to-create-a-ubuntu-20-04-server-on-aws-ec2-elastic-cloud-computing-5b423b5bf635

2022	Running Linux Desktop on an AWS EC2 instance	https://ubuntu.com/tutorials/ubuntu-desktop-aws#1-overview
2022	Install linux on ec2	https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html



Week 3 - Operating Systems and Networking Basics

Topics :

- Linux and Windows environments.
- General networking concepts: DNS, IPs, firewalls, HTTP/HTTPS.
- Web servers: Nginx, Apache.

Assignments:

1. **Configure a Linux VM with the following:**
   - ➢ Nginx as a reverse proxy.
   - ➢ Firewall rules to allow only HTTP/HTTPS traffic.
   - ➢ A custom 404 error page.
2. **Design and implement a basic networking topology:**
   - ➢ Set up two VMs with private IPs.
   - ➢ Configure one VM as a web server and the other as a client.
   - ➢ Use SSH to securely transfer files between them.

3.0 Document Overview:

Amazon Elastic Compute Cloud (EC2) is a popular computing service that allows users to create a virtual machine using various available Linux and applications Images. It is provided by Amazon Cloud with a complete infrastructure to host commercial applications on Linux virtual machines. In short, it is a Cloud service to create virtual servers.

In this document we'll be going through the steps of setting up an ubuntu linux server on aws EC2.

4. Step/ Procedure

4.1. Nginx as a reverse proxy

We will use Nginx as a reverse Proxy for jenkins.
Jenkins runs on port 8080(default), Nginx runs on port 80(default).

4.1.1. Installation of nginx

```
sudo apt update
```

4.1.2. Start and Enable NGINX Service

```
sudo systemctl start nginx
sudo systemctl enable nginx
```

4.1.3. Check NGINX Status

```
sudo systemctl status nginx
```

4.1.4. Test NGINX Web Server

```
http://<your_server_public_ip>
```

4.1.5. Installation of jenkins

4.1.5.1. Install Java

```
sudo apt update
sudo apt install fontconfig openjdk-21-jre
java -version
openjdk version "21.0.3" 2024-04-16
OpenJDK Runtime Environment (build 21.0.3+11-Debian-2)
OpenJDK 64-Bit Server VM (build 21.0.3+11-Debian-2, mixed
mode, sharing)
```

4.1.5.2. Install Jenkins

```
sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \
 https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
 /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```

4.1.5.3. Start/ enable jenkins

```
Systemctl start jenkins
Systemctl enable jenkins
```

4.1.5.4. Check jenkins

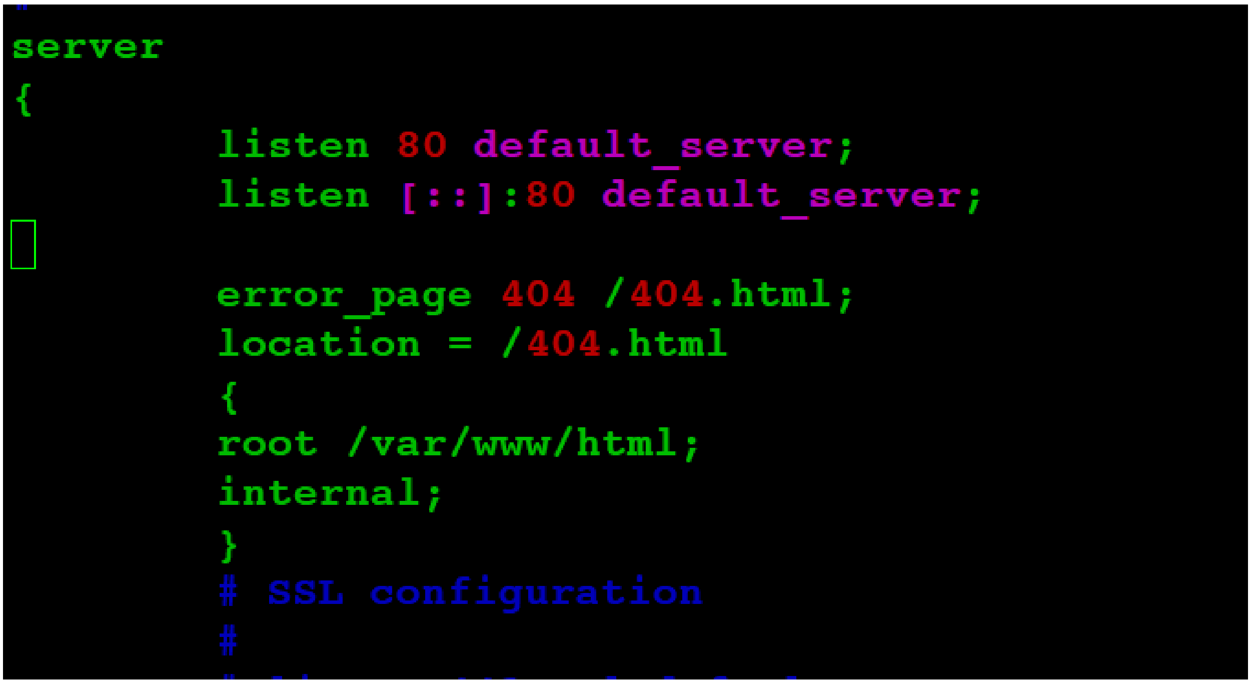
```
Systemctl status jenkins
```

4.1.5.5. Test Web Server
http://<your_public_ip>:8080

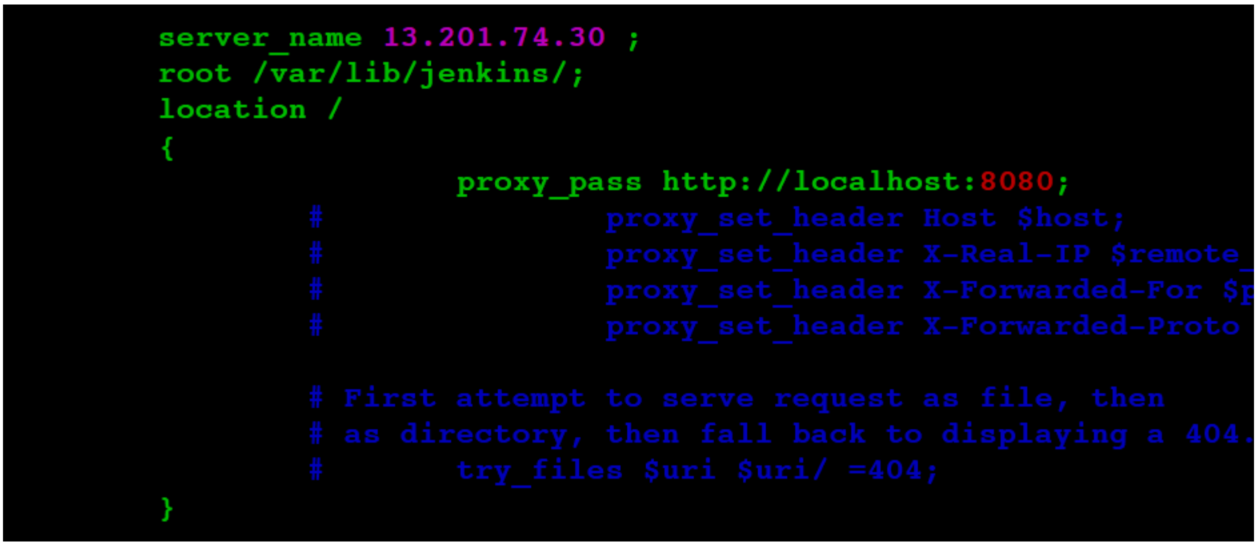
4.1.6. Configure the Default file of site-available in Nginx
cd /etc/nginx/site-available
Vim Default

4.1.6.1. The content of the file is:

```
server
{
    listen 80 default_server;
    listen [::]:80 default_server;

    error_page 404 /.htaccess/404.html;
    location = /.htaccess/404.html
    {
    root /var/www/html;
    internal;
    }
        server_name 13.201.74.30 ;
    root /var/lib/jenkins/;
    location /
    {
        proxy_pass http://localhost:8080;
        #           proxy_set_header Host $host;
        #           proxy_set_header X-Real-IP $remote_addr;
        #           proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        #           proxy_set_header X-Forwarded-Proto $scheme;

        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        #     try_files $uri $uri/ =404;
        error_page 502 503 504 = /404.html;
    }
    location = /404.html
    {
    root /var/www/html/.htaccess;
    internal;
    }
}
```



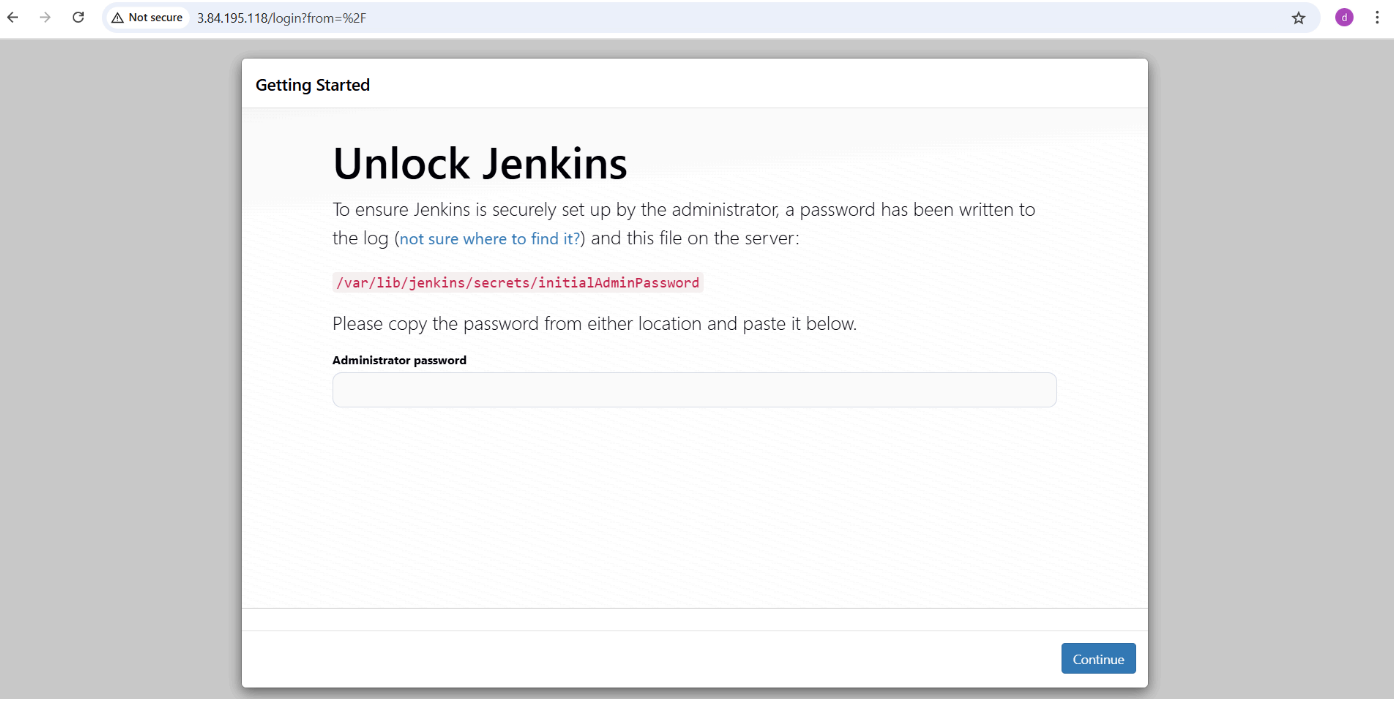
```
server
{
        listen 80 default_server;
        listen [::]:80 default_server;

        error_page 404 /404.html;
        location = /404.html
        {
        root /var/www/html;
        internal;
        }
        # SSL configuration
        #
```



```
        server_name 13.201.74.30 ;
        root /var/lib/jenkins/;
        location /
        {
                        proxy_pass http://localhost:8080;
                #               proxy_set_header Host $host;
                #               proxy_set_header X-Real-IP $remote_
                #               proxy_set_header X-Forwarded-For $p
                #               proxy_set_header X-Forwarded-Proto

                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                #       try_files $uri $uri/ =404;
        }
```

4.1.7. Restart Nginx

Systemctl restart Nginx



Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

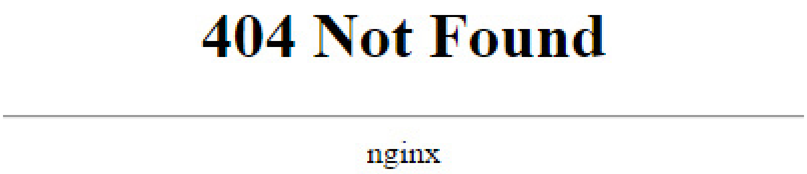
Please copy the password from either location and paste it below.

Administrator password

4.2. **Add Custom 404 Error Page**

404 error - page not found
404 error indicates that your web server is working, but it cannot locate the specific page or resource requested by the user.

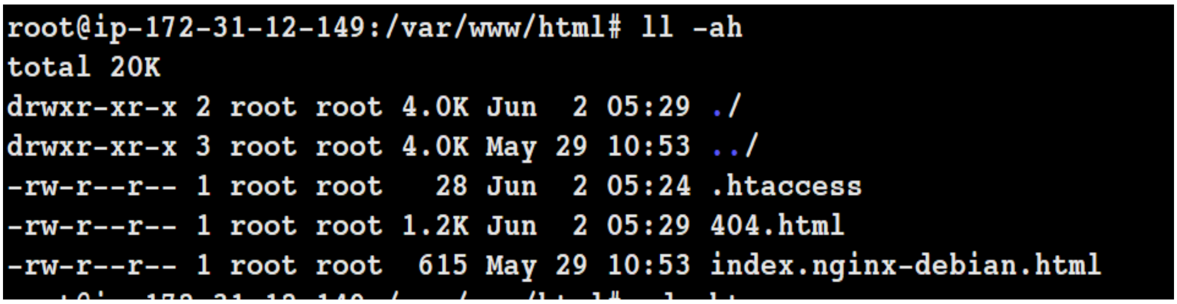
Default 404 error page



404 Not Found

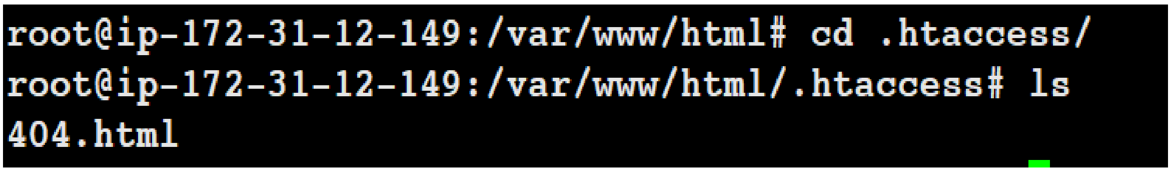
nginx

4.2.1. Create/ go to .htaccess directory
cd /var/www/html/
ll -ah
-------or —-------
mkdir .htaccess



```
root@ip-172-31-12-149:/var/www/html# ll -ah
total 20K
drwxr-xr-x 2 root root 4.0K Jun  2 05:29 ./
drwxr-xr-x 3 root root 4.0K May 29 10:53 ../
-rw-r--r-- 1 root root   28 Jun  2 05:24 .htaccess
-rw-r--r-- 1 root root 1.2K Jun  2 05:29 404.html
-rw-r--r-- 1 root root  615 May 29 10:53 index.nginx-debian.html
```

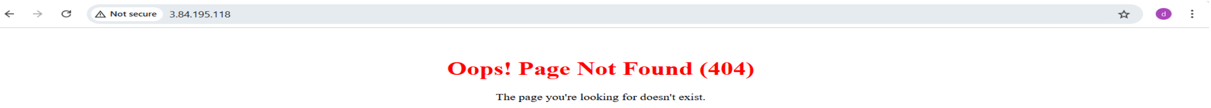
4.2.2. Create a custom 404 error HTML file



```
root@ip-172-31-12-149:/var/www/html# cd .htaccess/
root@ip-172-31-12-149:/var/www/html/.htaccess# ls
404.html
```

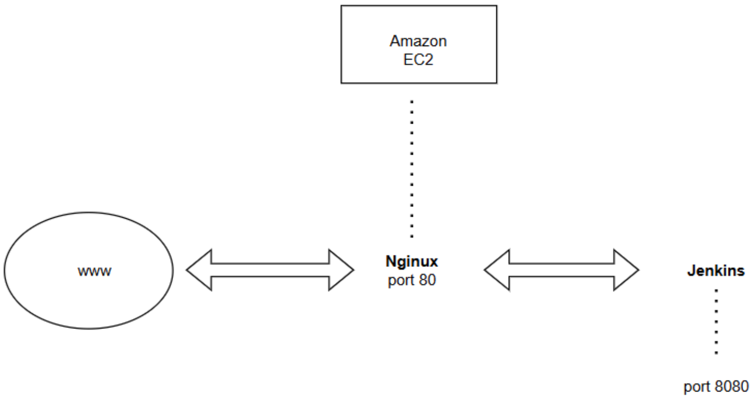
```
<!DOCTYPE html>
<html>
 <head><title>404 Not Found</title></head>
 <body style="text-align:center; margin-top:50px;">
 <h1 style="color:red;">Oops! Page Not Found (404)</h1>
 <p>The page you're looking for doesn't exist.</p>
</body>
</html>
```

4.2.3. Restart nginx
systemctl restart nginx



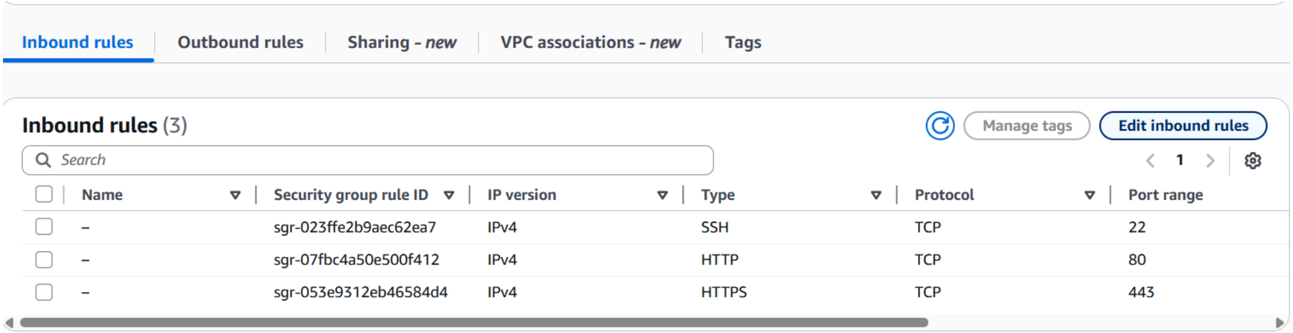
Oops! Page Not Found (404)

The page you're looking for doesn't exist.



4.3. Inbound rules to allow only HTTP/HTTPS traffic.

It simplifies setting up Inbound rules for allowing or denying connections.



Inbound rules (3)

□	Name	Security group rule ID	IP version	Type	Protocol	Port range
□	–	sgr-023ffe2b9aec62ea7	IPv4	SSH	TCP	22
□	–	sgr-07fbc4a50e500f412	IPv4	HTTP	TCP	80
□	–	sgr-053e9312eb46584d4	IPv4	HTTPS	TCP	443

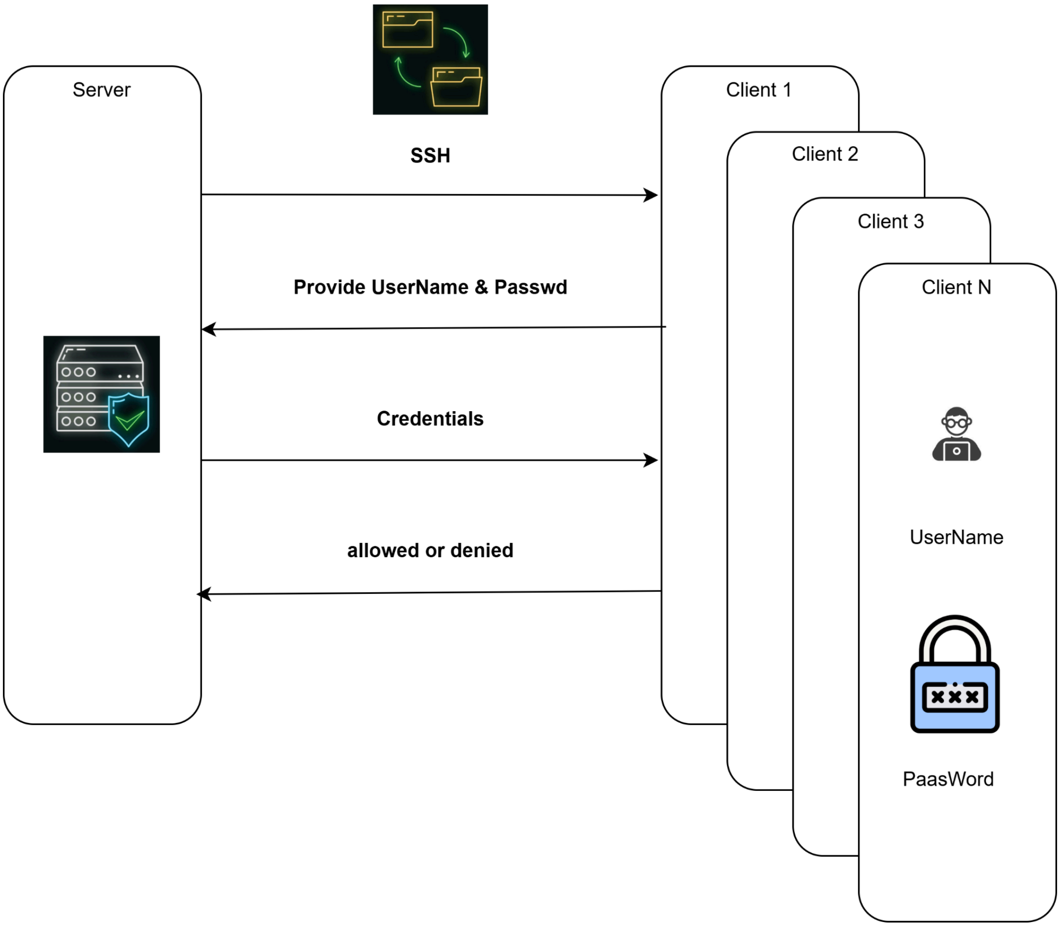
Inbound firewall rules are essential for network security because they control which external traffic is allowed to enter a network, protecting it from malicious connections, malware, and other threats. By specifying which incoming traffic is permitted, inbound rules prevent unauthorized access and ensure only legitimate connections reach internal resource

4.4. Use SSH to transfer files between 2 servers securely.

To transfer files using scp (Secure Copy Protocol) over SSH, explaining both **password-based** and **key-based authentication** methods.

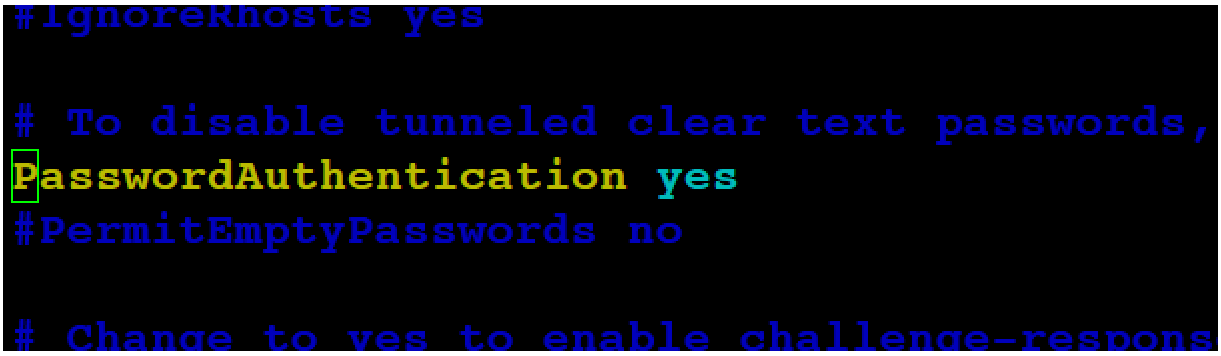
Situation: We have 2 EC2 servers, one acts as a DB server and the other acts as a developer server. Both servers are in the same VPC.

4.4.1. Password-based file transfer (on EC2 Instances)



NOTE: Perform 4.5.1.1 and 4.5.1.2 on the db server and the developer server.

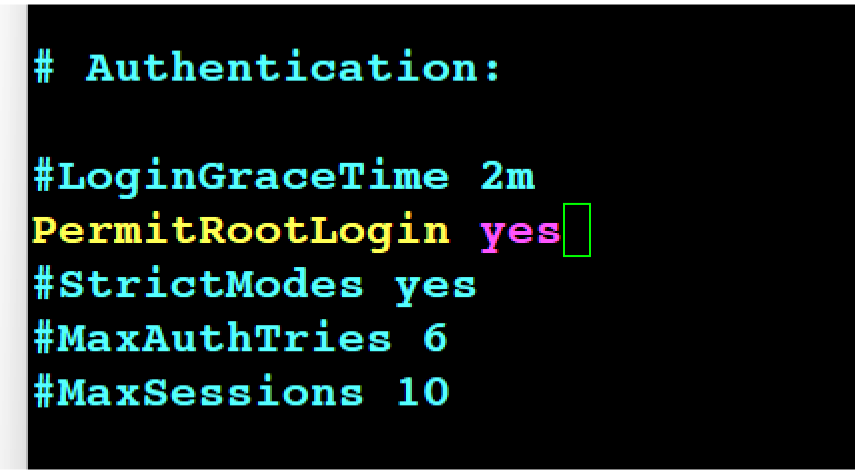
4.4.1.1. Allow Password authentication in both servers
vim /etc/ssh/sshd_config
Enable “PasswordAuthentication yes”



```
#IgnoreRhosts yes

# To disable tunneled clear text passwords,
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-respons
```

4.4.1.2. Allow root login
NOTE: In AWS cloud server, root login by SSH is prohibited due to security reasons, and SSH to root is not best practice.

vim /etc/ssh/sshd_config



```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

4.4.1.3. Create/add username and password to OS
To access root user:

su root
Passwd root

4.4.1.4. Check wether your OS have
/etc/ssh/sshd_config.d/60-cloudimg-settings.conf OR
Any files in "/etc/ssh/sshd_config.d"

ls /etc/ssh/sshd_config.d/
vim /etc/ssh/sshd_config.d/60-cloudimg-settings.conf

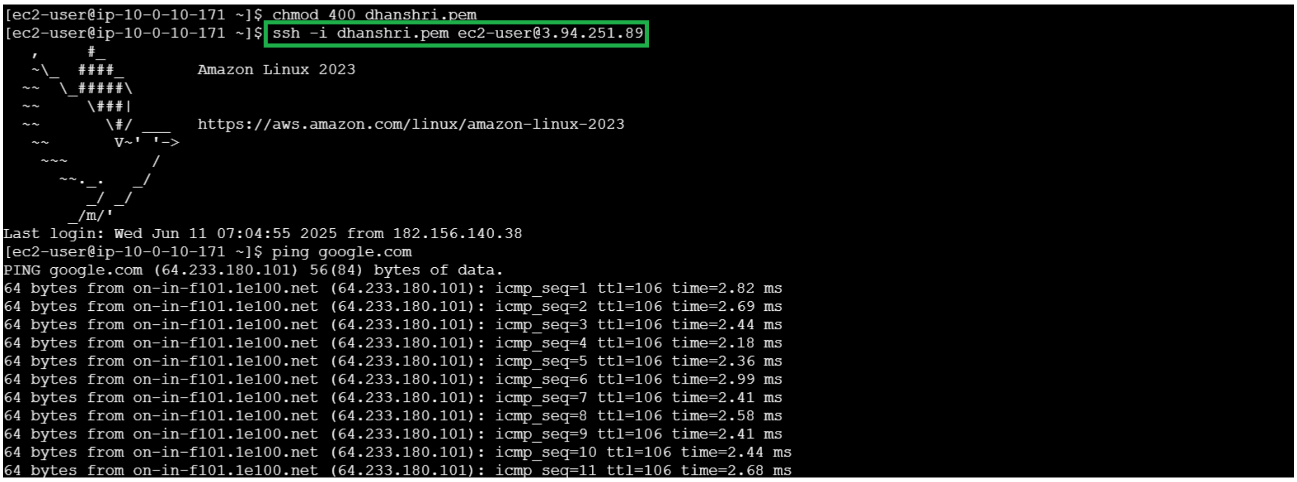
Enable PasswordAuthentication
PasswordAuthentication yes

4.4.1.5. Restart the SSH service
systemctl restart ssh
-------or-------
systemctl restart sshd

4.4.1.6. Command to do SSH using password

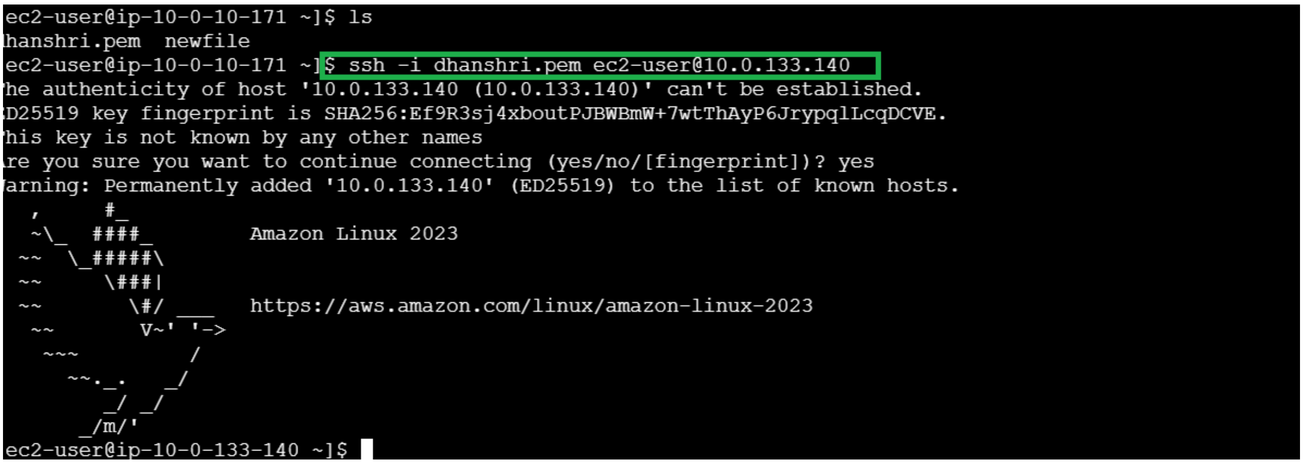
Case 1: Both servers are in the same VPN
ssh <user_name>@<private_IP>

Case 2: servers are in different VPN or exposed to the internet or have public IPs.
ssh <user_name>@<public_IP>



```
[ec2-user@ip-10-0-10-171 ~]$ chmod 400 dhanshri.pem
[ec2-user@ip-10-0-10-171 ~]$ ssh -i dhanshri.pem ec2-user@3.94.251.89
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Jun 11 07:04:55 2025 from 182.156.140.38
[ec2-user@ip-10-0-10-171 ~]$ ping google.com
PING google.com (64.233.180.101) 56(84) bytes of data.
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=1 ttl=106 time=2.82 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=2 ttl=106 time=2.69 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=3 ttl=106 time=2.44 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=4 ttl=106 time=2.18 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=5 ttl=106 time=2.36 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=6 ttl=106 time=2.99 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=7 ttl=106 time=2.41 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=8 ttl=106 time=2.58 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=9 ttl=106 time=2.41 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=10 ttl=106 time=2.44 ms
64 bytes from on-in-f101.1e100.net (64.233.180.101): icmp_seq=11 ttl=106 time=2.68 ms
```

SSH using public Key using Passwd



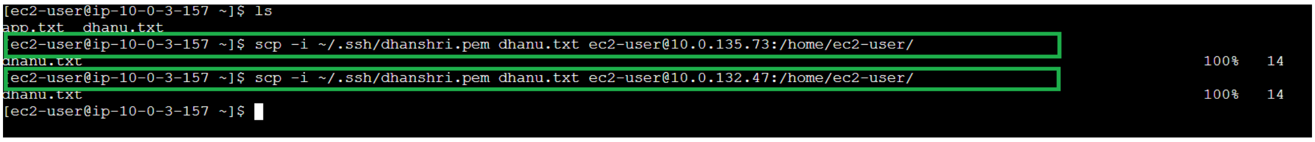
```
ec2-user@ip-10-0-10-171 ~]$ ls
hanshri.pem  newfile
ec2-user@ip-10-0-10-171 ~]$ ssh -i dhanshri.pem ec2-user@10.0.133.140
he authenticity of host '10.0.133.140 (10.0.133.140)' can't be established.
D25519 key fingerprint is SHA256:Ef9R3sj4xboutPJBWBmW+7wtThAyP6JrypqlLcqDCVE.
his key is not known by any other names
re you sure you want to continue connecting (yes/no/[fingerprint])? yes
arning: Permanently added '10.0.133.140' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
ec2-user@ip-10-0-133-140 ~]$ 
```

SSH using Private IP using Passwd

4.4.1.7. Use SCP to file transfer

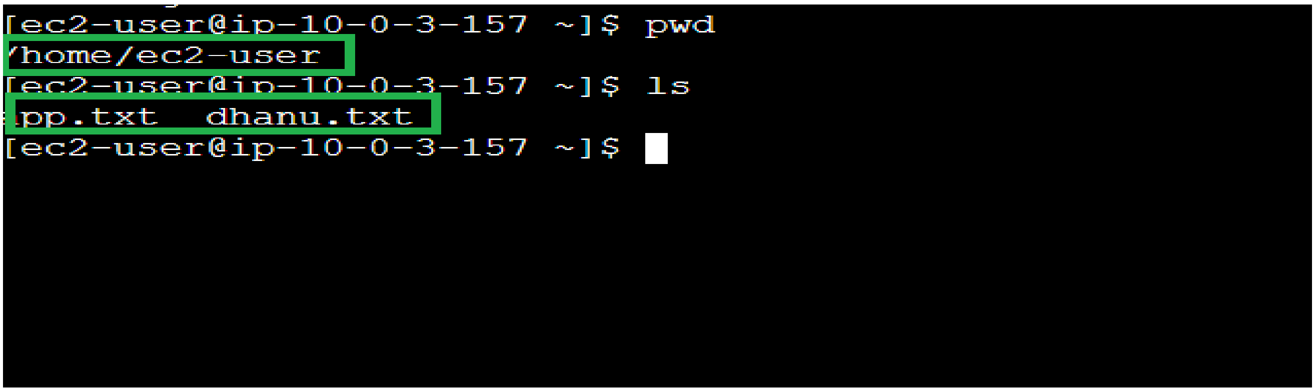
Command:

scp <dir_of_file>   <username>@<ip_addr>:<destination_dir>



```
[ec2-user@ip-10-0-3-157 ~]$ ls
app.txt  dhanu.txt
[ec2-user@ip-10-0-3-157 ~]$ scp -i ~/.ssh/dhanshri.pem dhanu.txt ec2-user@10.0.135.73:/home/ec2-user/
dhanu.txt                                                                                          100%   14
[ec2-user@ip-10-0-3-157 ~]$ scp -i ~/.ssh/dhanshri.pem dhanu.txt ec2-user@10.0.132.47:/home/ec2-user/
dhanu.txt                                                                                          100%   14
[ec2-user@ip-10-0-3-157 ~]$ 
```

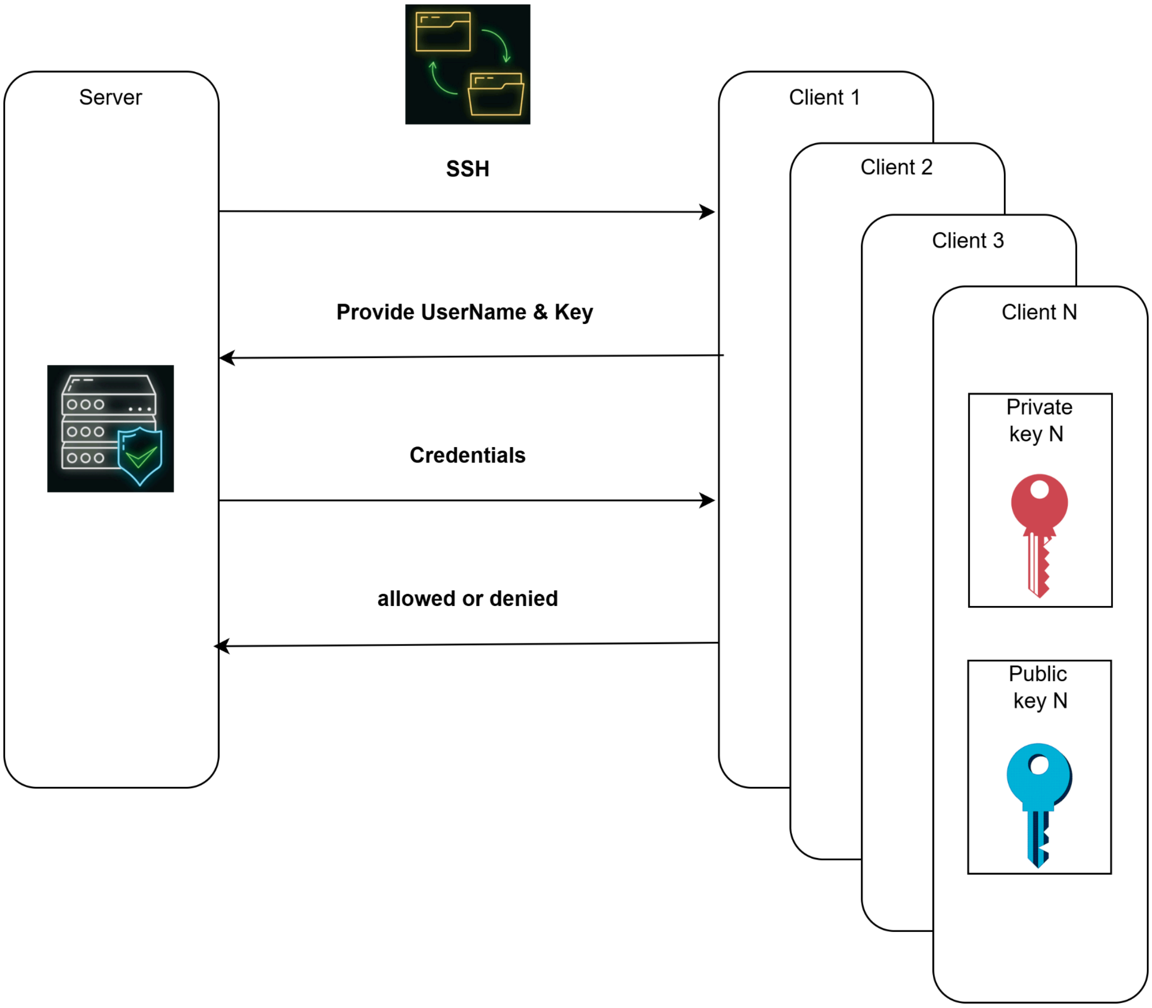
Executed the scp command in the developer os



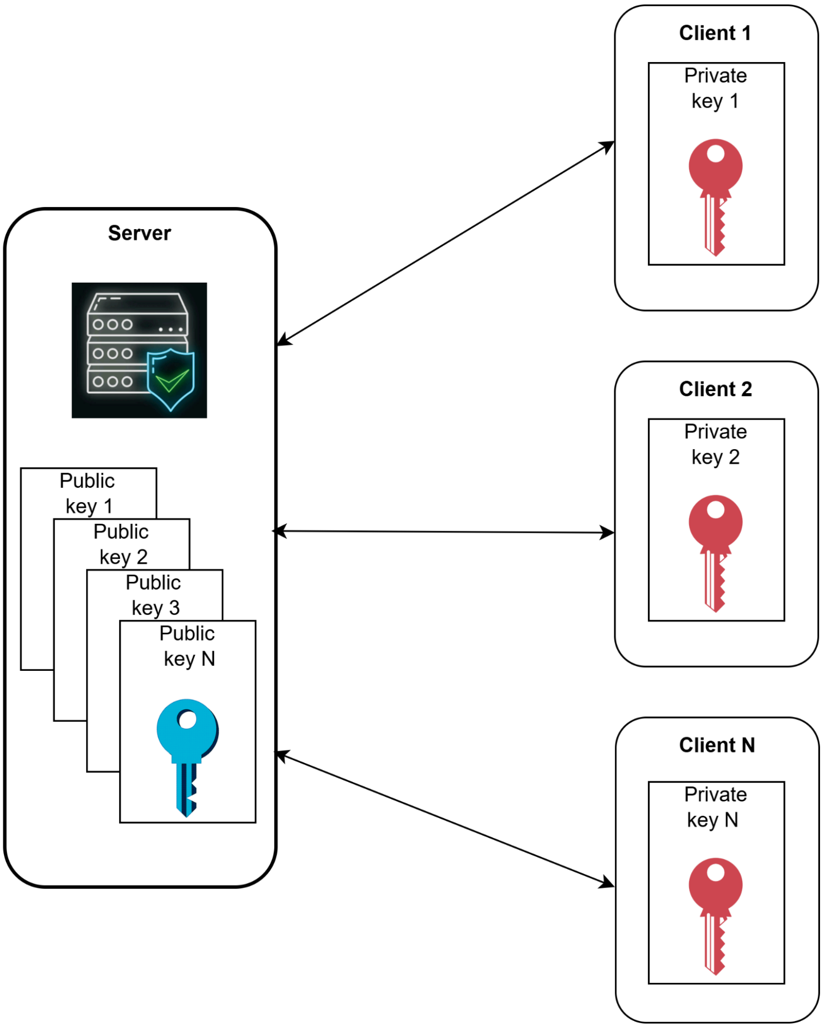
```
[ec2-user@ip-10-0-3-157 ~]$ pwd
/home/ec2-user
[ec2-user@ip-10-0-3-157 ~]$ ls
pp.txt   dhanu.txt
[ec2-user@ip-10-0-3-157 ~]$ 
```

File is transfer to DB os.

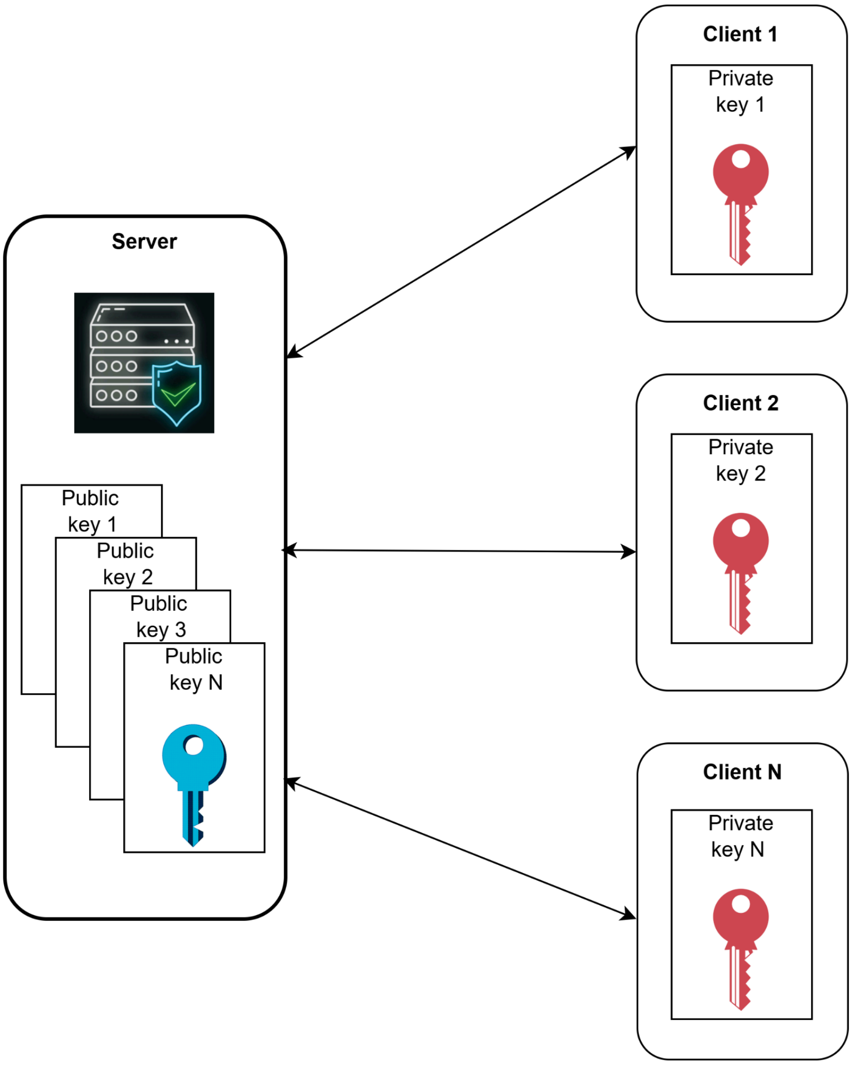
4.4.2. Key-based file transfer (on EC2 Instances)



Every user / client need to create key-pair.



All Public key of Clients is stored in the Server (DB server),
According to the above diagram, only the Client OS can transfer files to the server, but the server cannot transfer files to the Client because the Client server has the private key.



4.5.

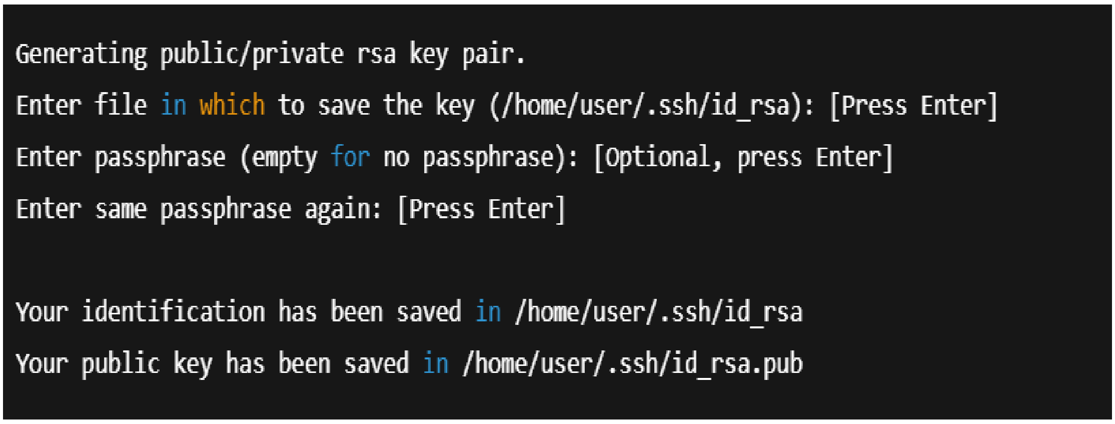
Situation: We have 2 EC2 servers, one acts as a DB server and the other acts as a developer server. Both servers are in the same VPC.
The DB server has only a Private IP.
To transfer the file, we are using port 2020.

User name of DB server - dbuser
Os - linux
Hostname of DB server - dbserver
Ssh key - dbkey, dbkey.pub

User name of the developer server - devuser
Os - amazon linux
Hostname of developer server - devserver
Ssh key - devkey, devkey.pub

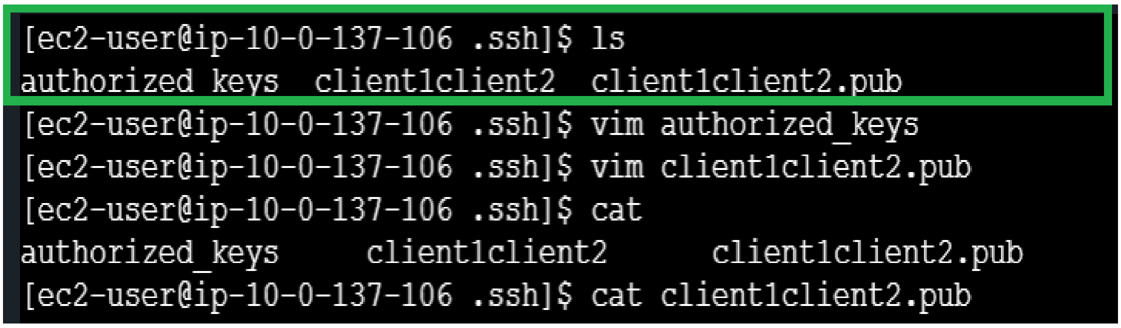
4.5.1. **Secure SSH authentication** without a password

Ssh-keygen



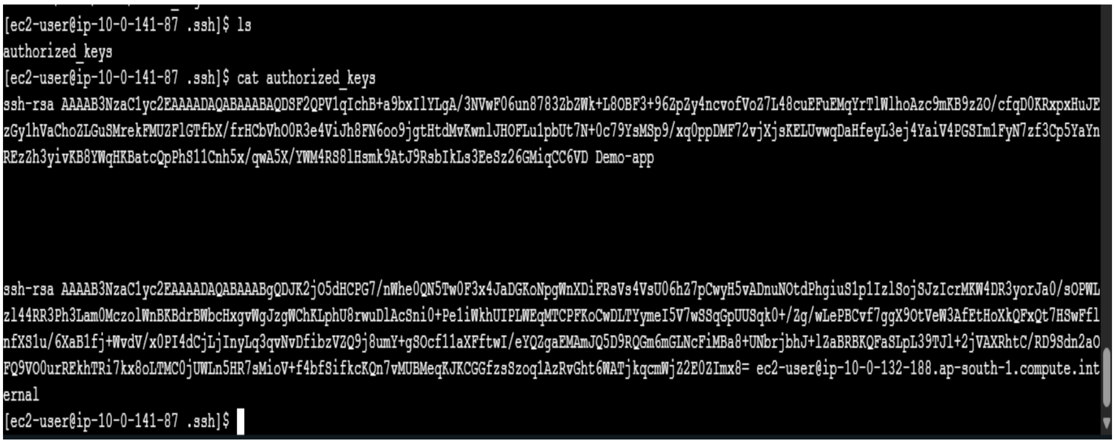
```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa): [Press Enter]
Enter passphrase (empty for no passphrase): [Optional, press Enter]
Enter same passphrase again: [Press Enter]

Your identification has been saved in /home/user/.ssh/id_rsa
Your public key has been saved in /home/user/.ssh/id_rsa.pub
```

Ls



```
[ec2-user@ip-10-0-137-106 .ssh]$ ls
authorized keys  client1client2  client1client2.pub
[ec2-user@ip-10-0-137-106 .ssh]$ vim authorized_keys
[ec2-user@ip-10-0-137-106 .ssh]$ vim client1client2.pub
[ec2-user@ip-10-0-137-106 .ssh]$ cat
authorized_keys     client1client2      client1client2.pub
[ec2-user@ip-10-0-137-106 .ssh]$ cat client1client2.pub
```

Vim authorized_keys
Cat authorized_keys



```
[ec2-user@ip-10-0-141-87 .ssh]$ ls
authorized_keys
[ec2-user@ip-10-0-141-87 .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDSF2QPV1qIchB+a9bxIlYLgA/3NVwF06un8783ZbZWk+L8OBF3+96ZpZy4ncvofVoZ7L48cuEFuEMqYrTlWlhoAzc9mKB9zZO/cfqD0KRxpxHuJE
zGy1hVaChoZLGuSMrekFMUZFlGTfbX/frHCbVhO0R3e4ViJh8FN6oo9jgtHtdMvKwnlJHOFLu1pbUt7N+0c79YsMSp9/xq0ppDMF72vjXjsKELUvwqDaHfeyL3ej4YaiV4PGSIm1FyN7zf3Cp5YaYn
REzZh3yivKB8YWqHKBatcQpPhS11Cnh5x/qwA5X/YWM4RS8lHsmk9AtJ9RsbIkLs3EeSz26GMiqCC6VD Demo-app



ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDJK2jO5dHCPG7/nWhe0QN5Tw0F3x4JaDGKoNpgWnXDiFRsVs4VsU06hZ7pCwyH5vADnuNOtdPhgiuS1p1IzlSojSJzIcrMKW4DR3yorJa0/sOPWL
zl44RR3Ph3Lam0MczolWnBKBdrBWbcHxgvWgJzgWChKLphU8rwuDlAcSni0+Pe1iWkhUIPLWEqMTCPFKoCwDLTYymeI5V7wSSqGpUUSqk0+/Zg/wLePBCvf7ggX9OtVeW3AfEtHoXkQFxQt7HSwFfl
nfXS1u/6XaB1fj+WvdV/x0PI4dCjLjInyLq3qvNvDfibzVZQ9j8umY+gSOcf11aXFftwI/eYQZgaEMAmJQ5D9RQGm6mGLNcFiMBa8+UNbrjbhJ+lZaBRBKQFaSLpL39TJl+2jVAXRhtC/RD9Sdn2aO
FQ9VO0urREkhTRi7kx8oLTMC0jUWLn5HR7sMioV+f4bfSifkcKQn7vMUBMeqKJKCGGfzsSzoq1AzRvGht6WATjkqcmWjZ2E0ZImx8= ec2-user@ip-10-0-132-188.ap-south-1.compute.int
ernal
[ec2-user@ip-10-0-141-87 .ssh]$ 
```

4.6 To **change ownership** of a file or directory in Linux

sudo adduser demo

sudo passwd demo

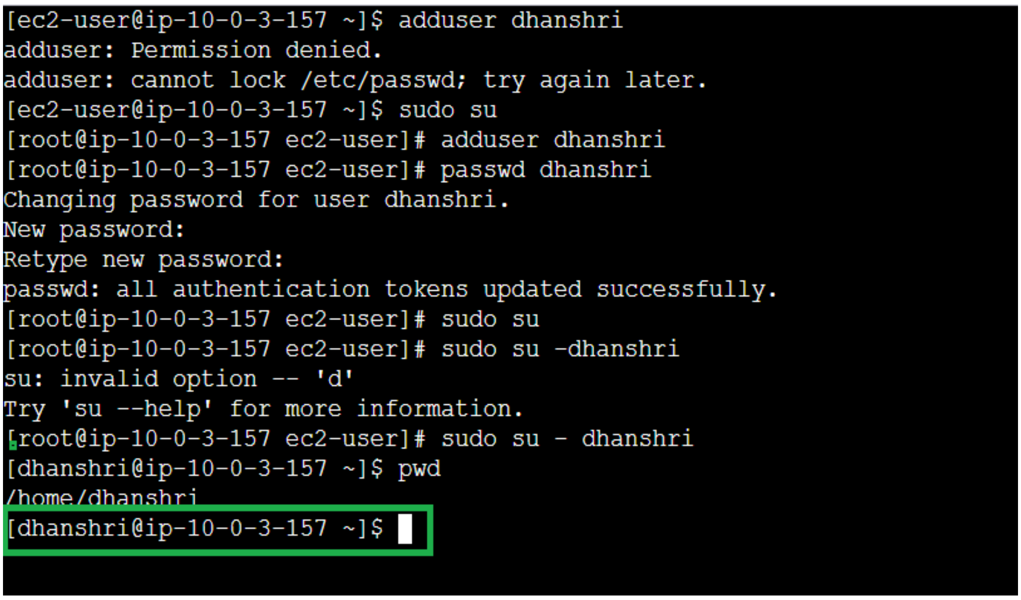
sudo usermod -aG wheel demo

sudo mkdir /home/demo/.ssh

sudo cp /home/ec2-user/.ssh/authorized_keys /home/demo/.ssh/
sudo chown -R demo:demo /home/demo/.ssh
sudo chmod 700 /home/demo/.ssh
sudo chmod 600 /home/demo/.ssh/authorized_keys

sudo chown demo:demo /home/ec2-user/app.txt

sudo mv /home/ec2-user/app.txt /home/demo/



```
[ec2-user@ip-10-0-3-157 ~]$ adduser dhanshri
adduser: Permission denied.
adduser: cannot lock /etc/passwd; try again later.
[ec2-user@ip-10-0-3-157 ~]$ sudo su
[root@ip-10-0-3-157 ec2-user]# adduser dhanshri
[root@ip-10-0-3-157 ec2-user]# passwd dhanshri
Changing password for user dhanshri.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-10-0-3-157 ec2-user]# sudo su
[root@ip-10-0-3-157 ec2-user]# sudo su -dhanshri
su: invalid option -- 'd'
Try 'su --help' for more information.
[root@ip-10-0-3-157 ec2-user]# sudo su - dhanshri
[dhanshri@ip-10-0-3-157 ~]$ pwd
/home/dhanshri
[dhanshri@ip-10-0-3-157 ~]$
```