

“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 Purpose	2
2.2 Document Revisions	2
2.3 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
VPC Configuration:	6
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 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]**

2.2 Document Revisions

Date	Version	Contributor(s)	Approver(s)	Section(s)	Change(s)
2025	1.0	Dhanshri patil	Akshay shinde	All Sections	New Document Created

2.3 Document References

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

Date	Document	Filename / Url
2025	Setup AWS Linux Server	https://linux.how2shout.com/how-to-create-a-ubuntu-linux-aws-ec2-instance-on-amazon-cloud/
2025	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

2025	Running Linux Desktop on an AWS EC2 instance	https://ubuntu.com/tutorials/ubuntu-desktop-aws#1-overview
2025	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.



4.0 Step/ Procedure

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).

1.Installation of nginx

- sudo yum update
- sudo yum install nginx-y

2.Start and Enable NGINX Service

- sudo systemctl start nginx
- sudo systemctl enable nginx

3.Check NGINX Status

- sudo systemctl status nginx

4.Test NGINX Web Server

http://<your_server_public_ip>

Install Jenkins

1. Update the System

- sudo yum update-y

2. Install Java (OpenJDK 17)

Amazon Linux 2023 does not support amazon-linux-extras , so use yum:

- sudo dnf install java-17-amazon-corretto-y

3. Add Jenkins Repository

- sudo wget -O /etc/yum.repos.d/jenkins.repo \https://pkg.jenkins.io/redhat-stable/jenkins.repo
- sudo rpm--import \https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

4. Install Jenkins

- sudo yum install jenkins-y

5. Start and Enable Jenkins

- sudo systemctl enable jenkins
- sudo systemctl start Jenkins

6. Test NGINX Web Server

http://<your_ip>

7. Configure NGINX as a Reverse Proxy



Edit the default config file:

- `sudo nano /etc/nginx/nginx.conf`

Update the server block as follows:

```
server {
    listen 80 default_server;
    server_name _;
    root
    /usr/share/nginx/html;
    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;
    }
    error_page 404 /custom_404.html;
    location = /custom_404.html {
        root /usr/share/nginx/html;
        internal;
    }
    error_page 502 503 504 /custom_404.html;
}
```

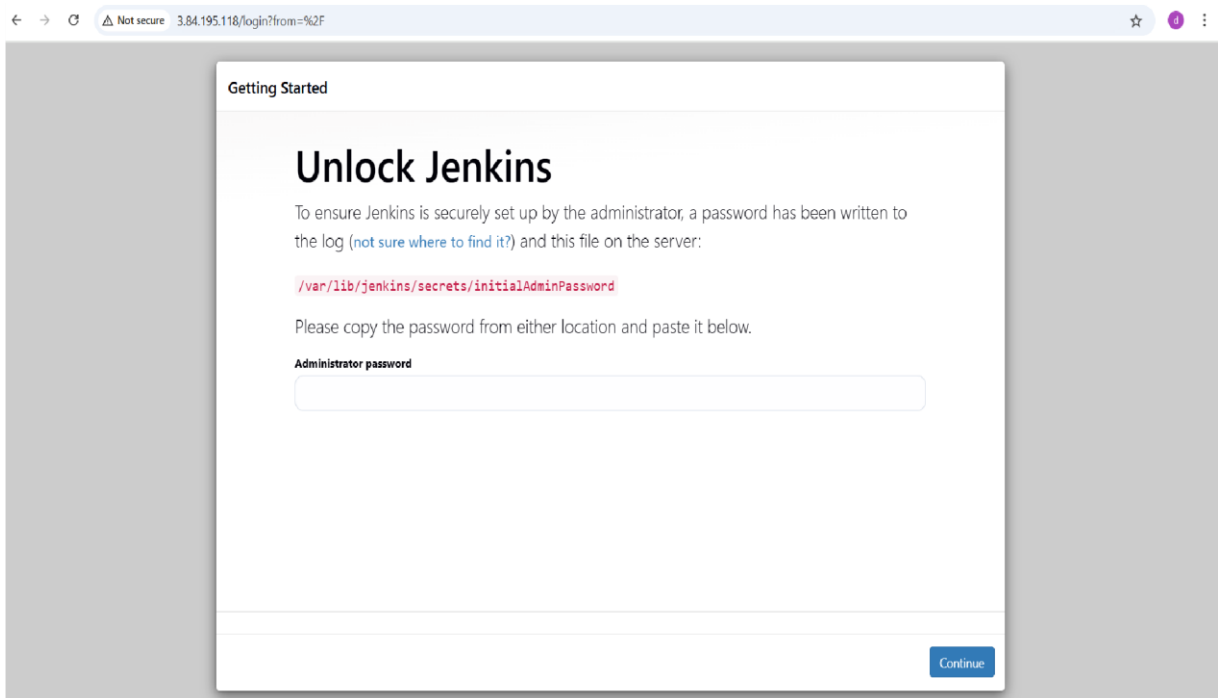
```
server
{
    listen 80 default_server;
    listen [::]:80 default_server;

    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_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;
}
```





Restart Nginx

- Systemctl restart nginx

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



Create/ go to .htaccess directory

- /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
```

Create a custom 404 error HTML file

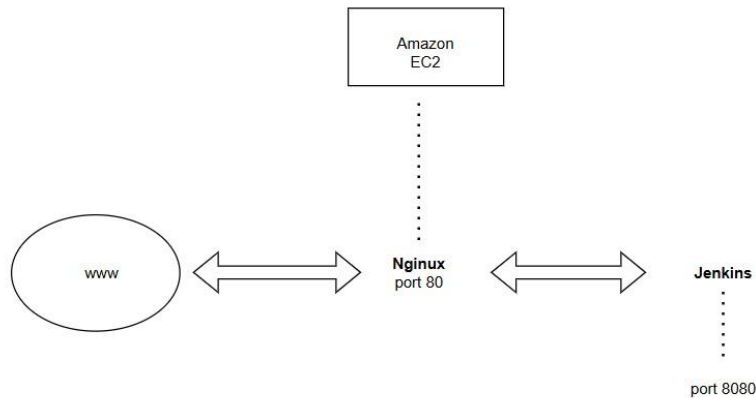
```
root@ip-172-31-12-149:/var/www/html# cd .htaccess/
root@ip-172-31-12-149:/var/www/html/.htaccess# ls
404.html
```

```
<!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>
```

Restart Nginx

- systemctl restart nginx





Inbound rules to allow only HTTP/HTTPS traffic.

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

Inbound rules								
Inbound rules Outbound rules Sharing - new VPC associations - new Tags								
Inbound rules (3) Manage tags Edit inbound rules								
<input type="text" value="Search"/>								
<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range		
<input type="checkbox"/>	-	sgr-023ffe2b9aec62ea7	IPv4	SSH	TCP	22		
<input type="checkbox"/>	-	sgr-07fbc4a50e500f412	IPv4	HTTP	TCP	80		
<input type="checkbox"/>	-	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

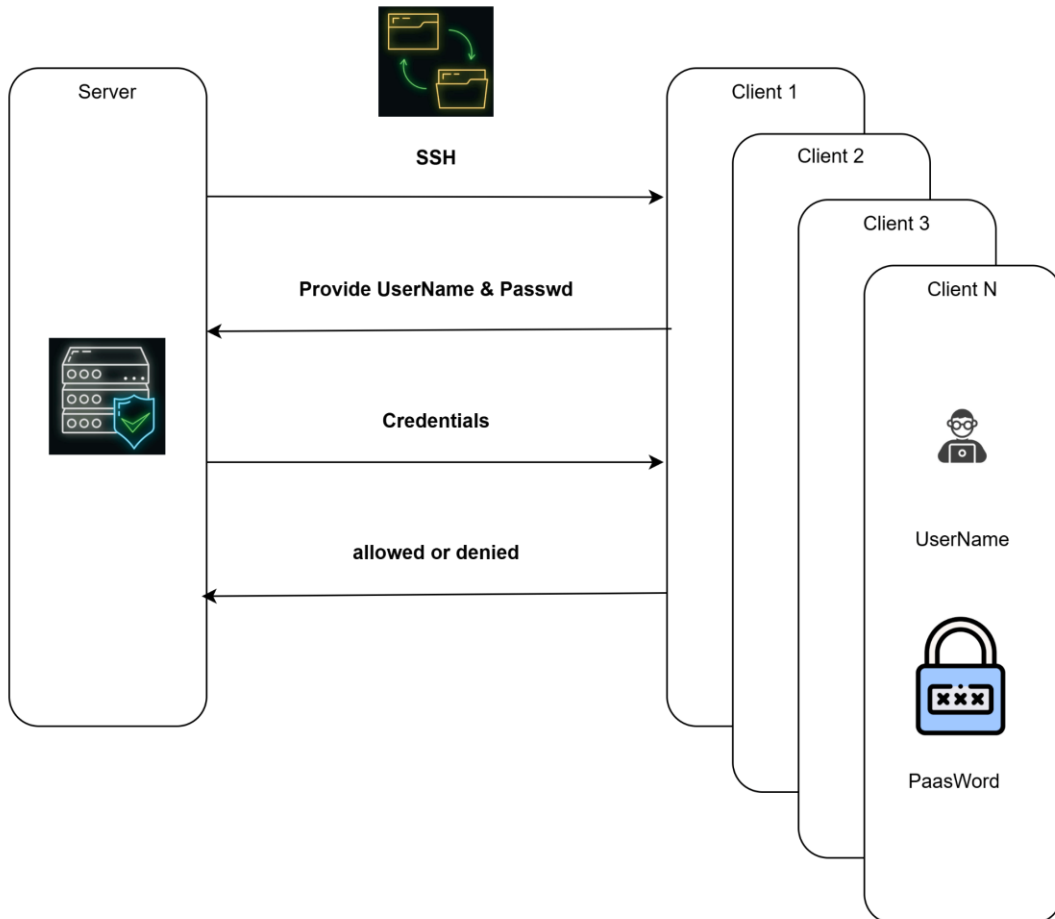
Use SSH to transfer files between 2 servers securely.

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.

Password-based file transfer (on EC2 Instances)



NOTE: Perform and on the db server and the developer server

Allow Password authentication in both servers

- vim /etc/ssh/sshd_config
- Enable "PasswordAuthentication yes"

```
#IgnoreRhosts yes

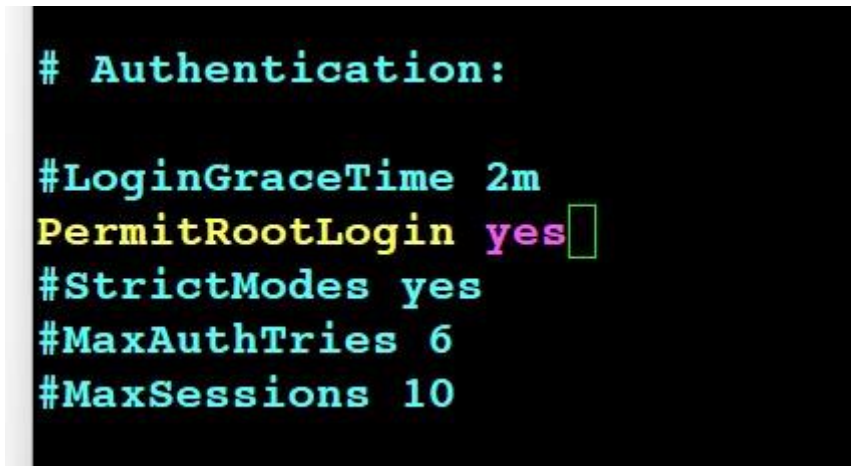
# To disable tunneled clear text passwords,
PasswordAuthentication yes
#PermitEmptyPasswords no

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

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
```

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

- su root
- Passwd root

Check whether 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
```

Restart the SSH service

- systemctl restart ssh
- or-----
- systemctl restart sshd

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>



[illegible]

SSH using public Key using Passwd

```
ec2-user@ip-10-0-10-171 ~]$ ls
dhanshri.pem newfile
ec2-user@ip-10-0-10-171 ~]$ ssh -i dhanshri.pem ec2-user@10.0.133.140
The authenticity of host '10.0.133.140 (10.0.133.140)' can't be established.
ED25519 key fingerprint is SHA256:Ef9R3sj4xboutPJBWBmW+7wtThAyP6JrypglLcqDCVE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: 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

Use SCP to file transfer Command:

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

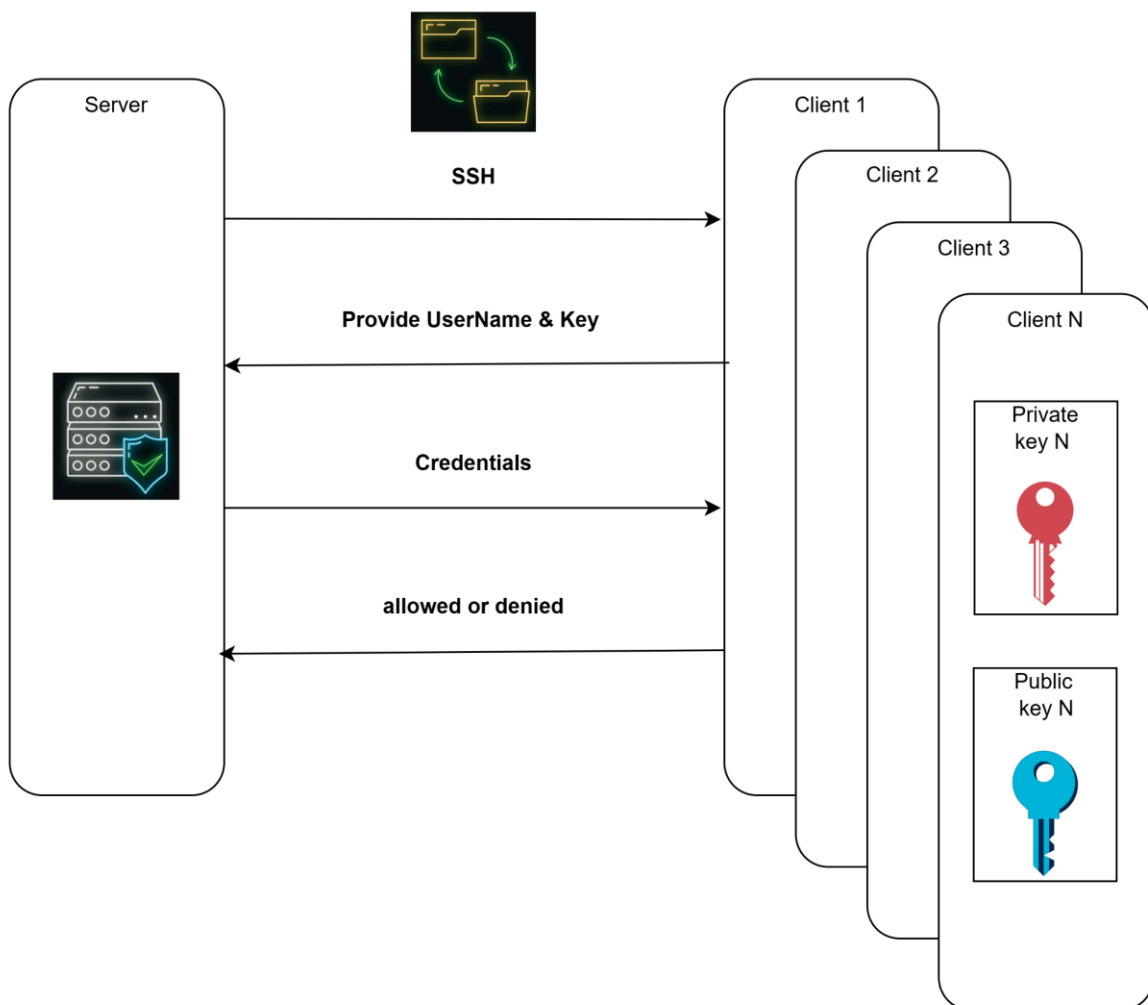
```
[ec2-user@ip-10-0-3-157 ~]$ ls  
app.txt dhanu.txt  
[ec2-user@ip-10-0-3-157 ~]$ scp -i ~/.ssh/dhanshri.pem dhanu.txt ec2-user@10.0.135.73:/home/ec2-user/  
dhanu.txt 100% 14  
[ec2-user@ip-10-0-3-157 ~]$ scp -i ~/.ssh/dhanshri.pem dhanu.txt ec2-user@10.0.132.47:/home/ec2-user/  
dhanu.txt 100% 14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

Executed the scp command in the developer os

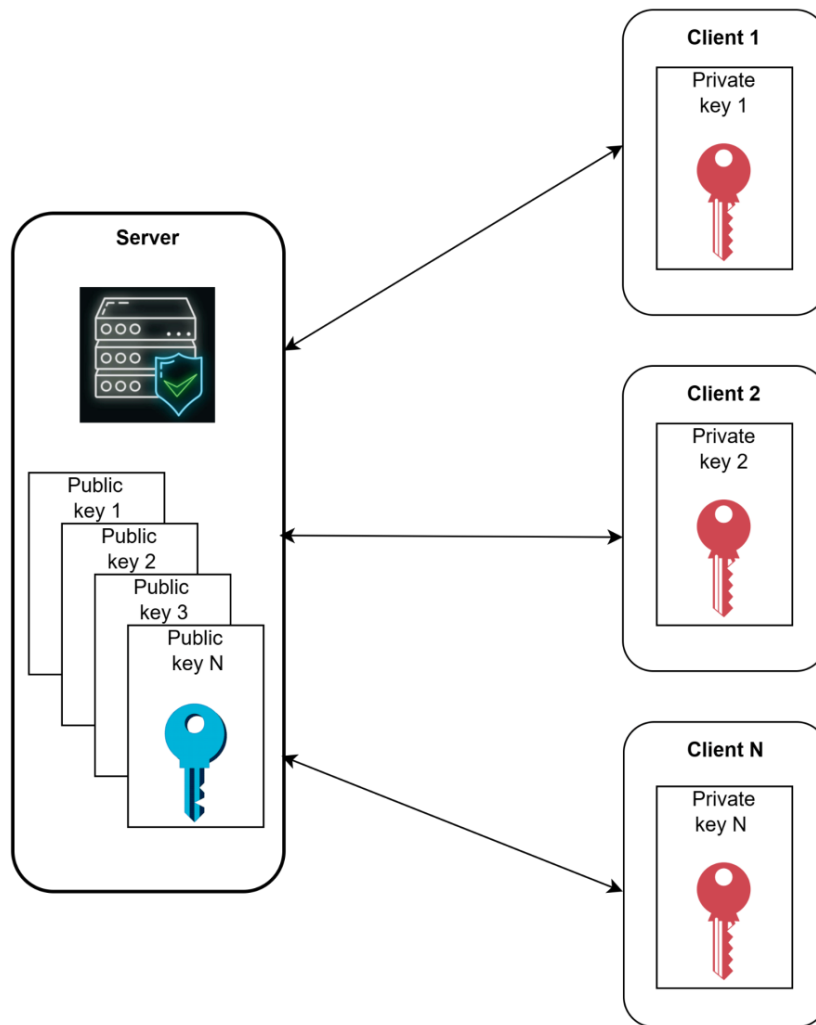
```
[ec2-user@ip-10-0-3-157 ~]$ pwd
/home/ec2-user
[ec2-user@ip-10-0-3-157 ~]$ ls
app.txt  dhanu.txt
[ec2-user@ip-10-0-3-157 ~]$
```

File is transfer to DB os.

Key-based file transfer (on EC2 Instances)

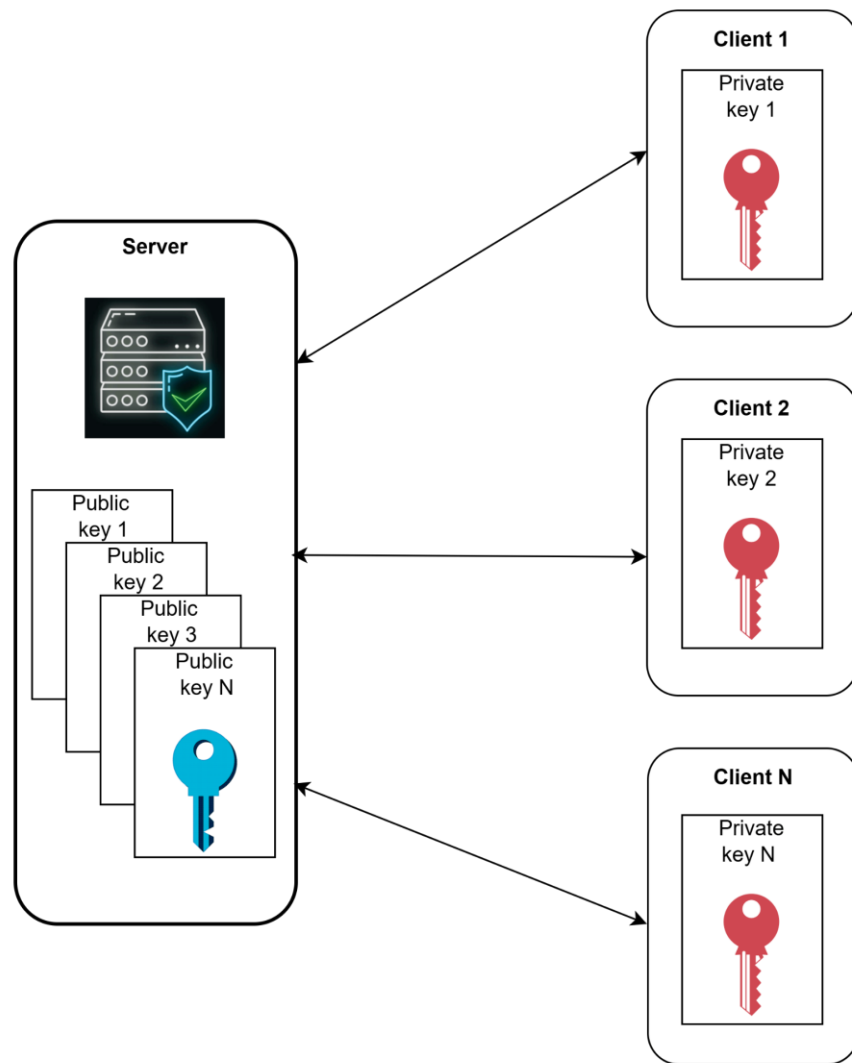


Every user / client need to create key-pair.



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.



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

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 AAAAB3NzaC1yc2EAAAADAQABAAQDSF2QPv1qIchB+a9bx1LYLgA/3NVwF06un87832bZWk+L80BF3+96Zpzy4ncvofVo27L48cuEFuEMqYrTlWlhoAzc9mKB9zZO/cfqD0KRxpHxHuJE
zGylhVaCho2LGuSMrekFMUZFlGTFbX/frHCbVh00R3e4ViJh8FN6oo9jgtHtdMvKwnlJHOFu1pbUt7N+0c79YsMSp9/xq0ppDMF72vjXjsKELUvwqDaHfeyL3ej4YaiV4PGSImlFyN7zf3Cp5YaYn
REzZh3yivKB8YWqHKBatcQpPhs11Cnh5x/qwA5X/YWM4RS8lHsmk9AtJ9RsbIkLs3EeS26GMiqCC6VD Demo-app

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDJK2j05dHCPG7/nWhe0QN5Tw0F3x4JaDGKoNpgWnXDIFrsVs4VsU06hZ7pCwyH5vADnuNotdPhgiuS1p1Iz1Soj8JzIcrMKW4DR3yorJa0/sOPWL
z144RR3Ph3Lam0MczolWnBKBDrBWbchXgvWgJzgWChKLphU8rwdlAcSni0+Pe1iWkhUIPLWEqMTCPPKoCwDLTYymeI5V7wSSqGpUUSqk0+/Zg/wLePBCvf7ggX9OtVeW3AFetHoXkQFxt7HSwFfl
nfXS1u/6XaB1fj+WvdV/x0PI4dCjLjInyLq3qvNvDfibzVZQ9j8umY+gSOcf11aXfftwI/eYQ2gaEMAmJQ5D9RQGM6mGLNcFiMBa8+UNbrjbhJ+lZaBRBKQFaSLpL39TJ1+2jVAXRhtC/RD9Sdn2a0
FQ9V00urREkhTri7kx8oLTMCOjUWLn5HR7sMioV+f4bfsifkcKQn7vMUBMeqKJKCGGfzsSzoqlA2RvGht6WATjKqcmWjZ2E0ZImx8= ec2-user@ip-10-0-132-188.ap-south-1.compute.int
ernal
[ec2-user@ip-10-0-141-87 .ssh]$
```



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/
- sudo su - 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 ~]$
```

