|  |
| --- |
| **“Expert Cloud Consulting” -**  **Linux VM and Networking Topology**  25.August.2025  Contributed by M Bindu Sri Santhi  Approved by Rushi Wagh(In Review)  Expert Cloud Consulting  Office #333, Gera Imperium Rise,  Hinjewadi Phase-II Rd, Pune, India – 411057 |

“Expert Cloud Consulting”

Linux VM and Networking Topology

### 1.0 Contents

Contents

[1.0 Contents 1](#_Toc207034165)

[2.0 General Information: 2](#_Toc207034166)

[3.0 Document Overview: 3](#_Toc207034167)

[4.0 Steps / Procedure 8](#_Toc207034168)

[5.0 Annexure - I 16](#_Toc207034169)

****

### 

### 2.0 General Information:

#### 2.1 Document Jira/ Github Ticket(s)

|  |  |
| --- | --- |
| **Ticket(s) Name** | **Url** |
| Learn AWS and servers | chatgpt |
|  |  |

#### 2.2 Document Purpose

This document provides step-by-step instructions and best practices for setting up the Ubuntu Linux operating system on AWS EC2 instances, specifically for deploying and running applications such as file transfers between private VMs and other related tasks.

#### 2.3 Document Revisions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Contributor(s)** | **Approver(s)** | **Section(s)** | **Change(s)** |
| 25/Aug/2025 | 1.0 | M Bindu | M Bindu | All Sections | New Document Created. File transfer between private VMs on AWS EC2 |
|  |  |  |  |  |  |

#### 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** |
| 2022 | Create Linux server on EC2 | <https://www.youtube.com/watch?v=dO1p_Tn4fOc&t=260s> |
| 2021 | Install Nginx and learned nginx reverse proxy | <https://www.youtube.com/watch?v=B62QSbPhh1s> |
| 2022 | Installed Jenkins and Configured Nginx as a reverse proxy for Jenkins | <https://www.rosehosting.com/blog/how-to-install-jenkins-on-ubuntu-24-04/?srsltid=AfmBOoqgiuHv9dzwvo3QgUjz1cmLClRZYfoaLh34iBzQk__SuFVJuvnw> |
| 2022 | Transfer between 2 VMs | <https://www.youtube.com/watch?v=3rROB-2r0bQ> |

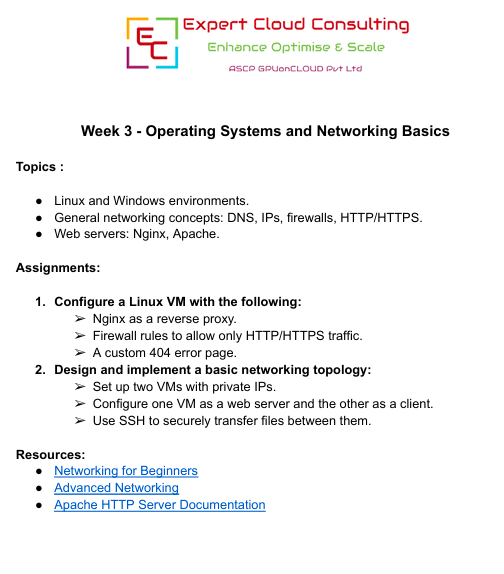
### 

### 3.0 Document Overview:

Amazon Elastic Compute Cloud (EC2) is a cloud computing service that allows users to create and manage virtual machines

This document demonstrates the step-by-step process of setting up Ubuntu Linux servers on AWS EC2, including tasks such as:

* Launching public and private EC2 instances
* Configuring SSH access with key pairs
* Transferring files securely between private virtual machines using SCP



**What is AWS ?**

AWS (Amazon Web Services) is **Amazon’s cloud computing platform** that provides on-demand IT resources over the internet — like servers, storage, databases, networking, analytics, and more — without needing to buy and maintain physical hardware.

Think of AWS as **renting powerful computers and software tools from Amazon**, but you only pay for what you use.

**Key Points:**

* **Cloud-based**: Everything runs on Amazon’s global data centers.
* **Pay-as-you-go**: No large upfront costs; you’re billed for the hours, storage, or data you use.
* **Scalable**: You can start small and scale up to millions of users without changing hardware.
* **Global**: Data centers in multiple regions worldwide for faster performance and reliability.



**Popular AWS Services:**

| **Category** | **Example Services** | **Purpose** |
| --- | --- | --- |
| Compute | **EC2** | Virtual servers to run apps. |
| Storage | **S3** | Store and retrieve files. |
| Databases | **RDS** | Manage relational or NoSQL databases. |
| Networking | **VPC** | Networking, DNS, and traffic routing. |

**EC2:**

EC2 stands for **Elastic Compute Cloud**. It is one of the core services of **AWS (Amazon Web Services)** and is used to run virtual servers in the cloud. Let me explain it step by step in simple terms.

**What EC2 is**

* EC2 is like renting a computer from Amazon’s cloud data center.
* Instead of buying a physical server, you can launch a virtual server (called an **instance**) that runs your applications.
* You can choose the operating system, CPU, memory, storage, and network settings according to your needs

**Key Features of EC2**

* **Elasticity**: You can quickly scale up (more resources) or scale down (fewer resources) based on demand.
* **Variety of instance types**: EC2 provides different types of instances optimized for **compute**, **memory**, **storage**, or **GPU-intensive tasks**.
* **Pay-as-you-go**: You only pay for what you use, like electricity for your server.
* **Secure**: You can control access using **security groups** and **key pairs** (SSH keys for Linux or password for Windows).

**Components**

1. **Instance** – Your virtual server.
2. **AMI (Amazon Machine Image)** – Preconfigured template with OS and software.
3. **Key Pair** – For securely connecting to your instance.
4. **Security Groups** – Firewall rules for controlling inbound/outbound traffic.
5. **Elastic IP** – Public IP that stays fixed even if you stop/restart the instance.

**VPC:**

A **VPC** stands for **Virtual Private Cloud** in AWS.

**What VPC is**

* A VPC is like your **own private network in the AWS cloud**.
* It lets you control your **IP addresses, subnets, routing, and security** for resources like EC2 instances, databases, and load balancers.
* Think of it as a **data center inside the cloud**, but fully virtual and under your control.

**. Key Features of VPC**

* **Isolation**: Resources inside your VPC are isolated from other users’ resources.
* **Subnets**: You can divide your VPC into smaller networks called **subnets** (public or private).
  + **Public subnet** → Can access the internet.
  + **Private subnet** → Cannot directly access the internet.
* **Security**: Use **security groups** and **network ACLs** to control traffic.
* **Customizable IP ranges**: You choose your own CIDR block (e.g., 10.0.0.0/16) for your VPC.

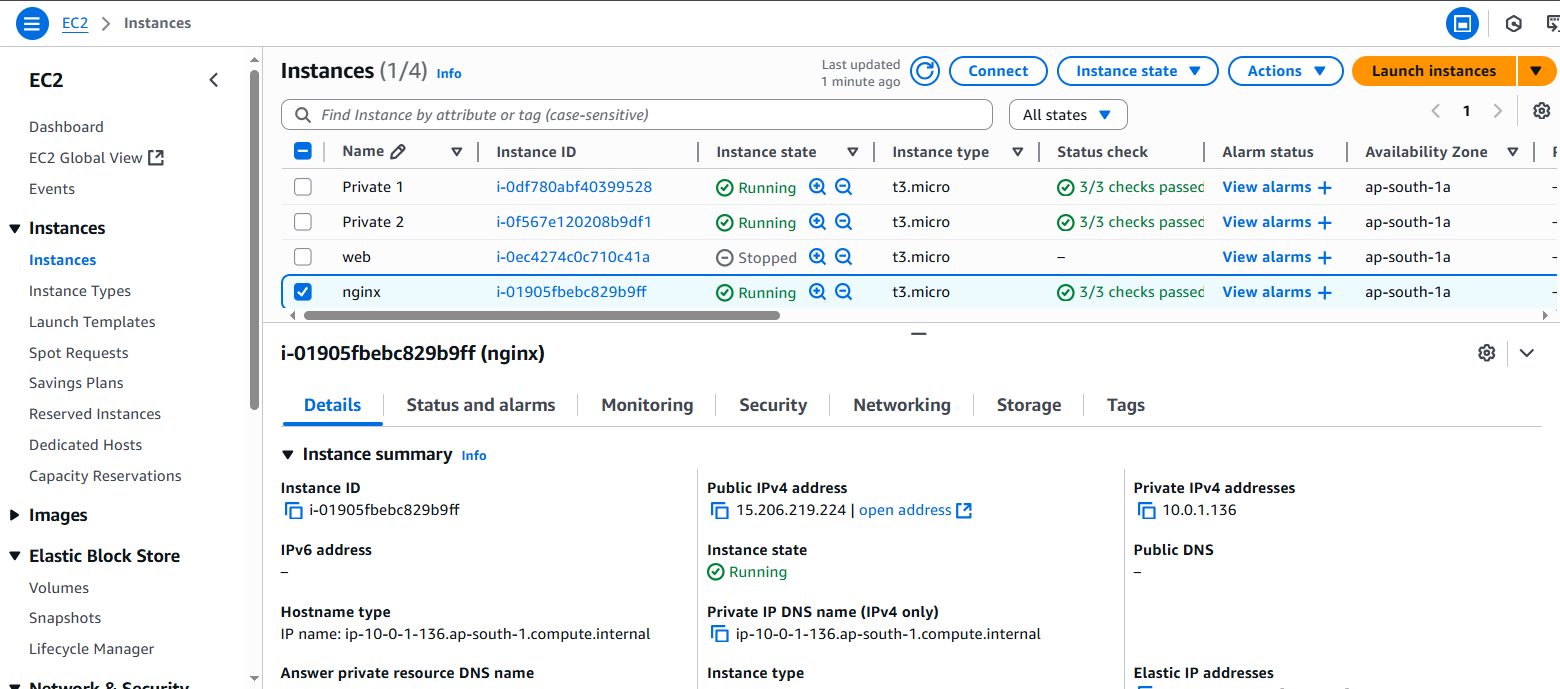
**Components of VPC**

1. **Subnets** – Smaller network segments inside your VPC.
2. **Internet Gateway** – Allows resources in public subnets to access the internet.
3. **Route Tables** – Direct traffic between subnets and outside networks.
4. **Security Groups** – Virtual firewall for your instances.
5. **NAT Gateway/Instance** – Lets private instances access the internet for updates without exposing them.

### 4.0 Steps / Procedure

#### 4.1 : Installed Nginx

For installing Nginx and reverse proxy I installed public VM named as Nginx



Updated OS:

Sudo apt update

Installed Nginx:

Sudo apt install Nginx

Welcome page of Nginx

A screenshot of a computer

AI-generated content may be incorrect.

#### 4.2: Installed Jenkins

Installed Jenkins

By default Jenkins is running on port 8080

A screenshot of a computer

AI-generated content may be incorrect.

#### 4.3: Configured Nginx as a reverse proxy for Jenkins

Configured Nginx as a reverse proxy for Jenkins

A screenshot of a computer

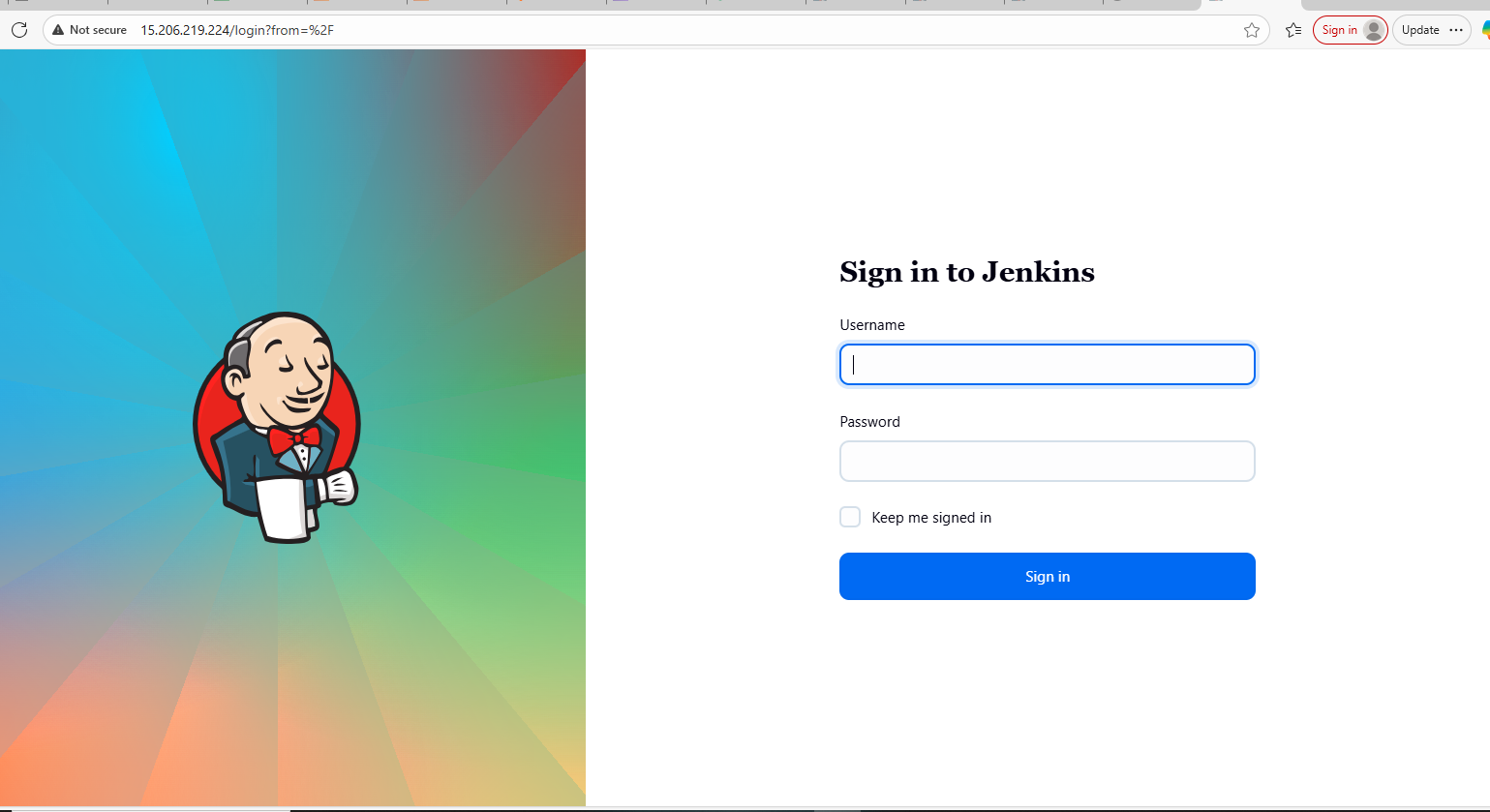
AI-generated content may be incorrect.

Now, Jenkins is running on port 80. So, reverse proxy worked here

4.4 Jenkins page:

A screenshot of a computer

AI-generated content may be incorrect.



##### 4.5: Custom 404 Error page :

I created Custom 404 error page with using html page

A screenshot of a computer

AI-generated content may be incorrect.

##### 4.4.2: Security Group Configuration

Set up two VMs with private IPs.

One is private1 VM:

A screenshot of a computer

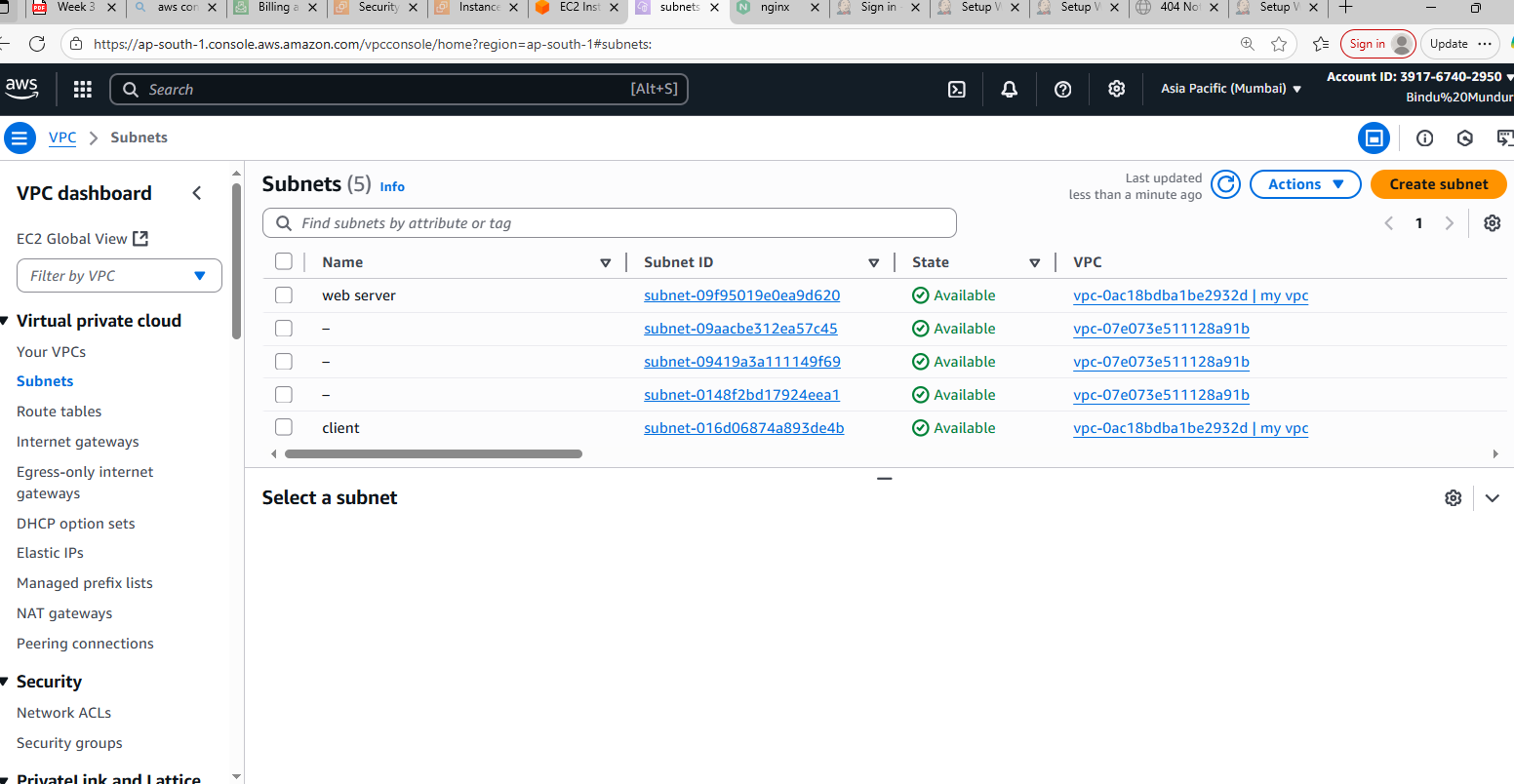
AI-generated content may be incorrect.

Second one is private2 VM :

A screenshot of a computer

AI-generated content may be incorrect.

Created 2 subnets also in same VPC



#### 4.5: Named one VM as a web server and the other as a client.

Named them as web server and client

A screenshot of a computer

AI-generated content may be incorrect.

#### 4.5: SSH to securely transfer files between them.

Using SSH, I transfer files, Keys between them using this command :

First, I located my SSH keys on the public VM at ~/.ssh/.

find ~ -name "id\_ed25519\*"

Then,I used **scp (secure copy)** to transfer both the private key and public key from the public VM to the private VM:

scp ~/.ssh/id\_ed25519 ~/.ssh/id\_ed25519.pub [ubuntu@10.0.2.45:~/.ssh/](mailto:ubuntu@10.0.2.45:~/.ssh/)

A screenshot of a computer

AI-generated content may be incorrect.

### 

### 5.0 Annexure - I

I installed **Nginx** on a Linux VM and configured it as a **web server**. I also created a **custom 404 error page** to handle invalid requests and improve user experience. During this process, I learned how Nginx works as a **reverse proxy**, allowing requests to be forwarded to backend services while serving static content efficiently. To implement the reverse proxy, I installed **Jenkins** on port 8080 as the backend service and configured Nginx to route incoming HTTP requests to Jenkins. This setup helped me understand key concepts such as server configuration, port mapping, handling high traffic, and ensuring secure and smooth communication between the client and backend services. Overall, the task enhanced my practical knowledge of web server management, reverse proxy setup, and real-world application deployment.