**Starting with Linux: Easy Server Setup**

In the world of cloud computing and server management, creating and managing a Linux server is a fundamental skill. Whether you’re setting up a personal project or managing a production environment, understanding the process from start to finish is crucial. This will walk you through the essential steps to create a Linux server and connect to it.

**Step 1: Creating a Linux Server**

***1.1 Set Up a Virtual Machine (VM)***

First, you’ll need to set up a Virtual Machine (VM). You can do this using cloud providers such as AWS, Azure, or Google Cloud. During the VM creation process, specify the Linux distribution you wish to use. (e.g., Ubuntu, CentOS, or Amazon Linux).

***1.2 Install Linux***

Once your VM is up, proceed with installing your preferred Linux distribution. Follow the installation prompts to set up your system, including setting a root password and configuring the network settings.

**Step 2: Authentication Methods**

***2.1 What You Know***This method involves using a username and password, common in private systems.

***2.2 What You Have***Tokens like RSA keys or Multi-Factor Authentication (MFA) are often used in public systems.

***2.3 What You Are***Biometric authentication, such as fingerprints or retina scans, is employed in highly secure environments.



Even if you know the server’s IP address, access is impossible without proper authentication. Only users with the correct private key can connect.

**Step 3: Understanding Ports and Services**

Understanding ports and services is key to managing your Linux server effectively. **In a server, there are total 65,536 ports (i.e., 0 – 65,535)**

***SSH (Port 22)***: Used for secure server access.  
***HTTP (Port 80) & HTTPS (Port 443)***: Used for web services.  
**MySQL (Port 3306)**: Used for database services.

**Step 4: Generating an SSH Key Pair and Importing It to AWS EC2**

***4.1 Install Git Bash***We need client to connect to the Linux sever. For connecting ssh server, we have many clients like gitbash, putty, mobaxterm etc., Here we Download and install Git Bash on our Windows machine.

***4.2 Create an SSH Key Pair***Use the following command to create your SSH key pair:

**ssh-keygen -f filename**

**Example:** ssh-keygen -f linux-key

This generates two files: **linux-key.pem** (private key) and **linux-key.pub** (public key).

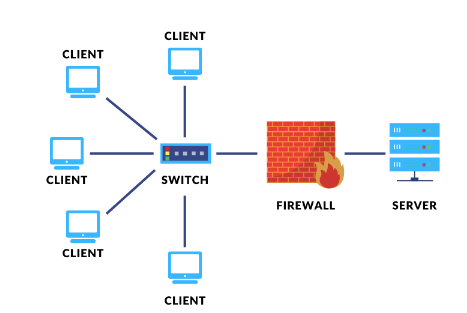


***4.3 View and Import Your Public Key to AWS EC2***

View the contents of your public key by opening **linux-key.pub** in a text editor. Then, log into your AWS account, navigate to the EC2 dashboard, and import your public key under “***Key Pairs***.”

**Step 5: Configuring the Firewall and Security Groups**

***5.1 Understanding the Firewall***Think of your firewall as the gatekeeper of your server. It ensures that only authorized requests can reach your server, blocking any unauthorized access attempts.

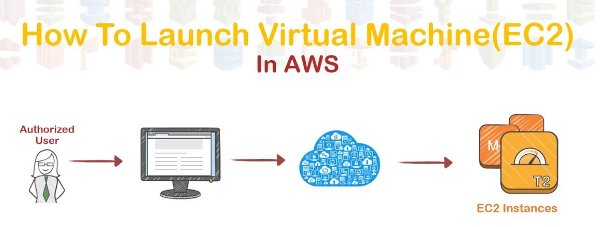


***5.2 Setting Up a Security Group***

* Open AWS and go to the security group.
* Create a security group named **allow-ssh**` to allow ***SSH*** connections on ***port 22***. Set the inbound rule to allow traffic from the internet (`***CIDR block: 0.0.0.0/0***`). You can further restrict access to specific IP addresses for added security.

**Step 6: Launching and Connecting to Your Linux Server**

***6.1 Launching the Server***Navigate to the EC2 dashboard, click “***Launch Instances***,” and configure your server:



* ***Instance name***: Choose a descriptive name (e.g., “***Linux***”).
* ***OS***: Select “Amazon Linux 2023 AMI” (similar to CentOS or ***RedHat***)
* ***Instance type***: Choose ***t2.micro*** *or* ***t3.micro***.
* ***Key pair***: Select the key pair you created (e.g., “***linux-key***”).
* ***Security group***: Choose the **allow-ssh**` group.

Click “***Launch Instance***” and find your server’s IP address in the Instances table.

***6.2 Connecting to the Server***Once your server is up, use the following command to connect:

**ssh -i linux-key.pem ec2-user@server\_ip\_address**

**Conclusion**

Creating and managing a Linux server involves a series of steps that require careful attention to detail. From generating SSH keys to understanding firewall rules, each step is crucial to ensuring a secure and efficient server environment. Armed with these basics, you can confidently manage your Linux server and execute essential commands to maintain it.