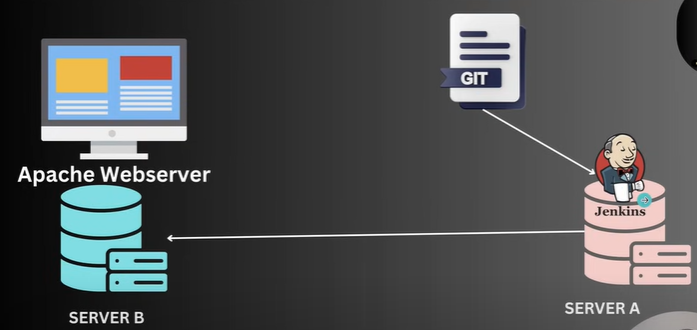
**Project Steps for Setting Up Jenkins, Ansible, and Apache Server with Automated Deployment**

****

1. **Launch Instances**

* Launch two instances: designate one as the master and the other as node1.

1. **Install Jenkins on the Master**

* SSH into the master instance.
* Update the package repository:

sudo apt update

* Install Jenkins
* wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
* sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
* sudo apt update

sudo apt install Jenkins

* Start Jenkins service
* sudo systemctl start jenkins

sudo systemctl enable Jenkins

1. **Install Publish Over SSH Plugin on Jenkins**

* Access Jenkins web interface at http://<master\_ip>:8080.
* Go to "Manage Jenkins" -> "Manage Plugins".
* Install the "Publish Over SSH" plugin.
* Generate SSH keys on the master node:

ssh-keygen -t rsa

* Copy the public key to node1:

ssh-copy-id -i ~/.ssh/id\_rsa.pub <node1\_user>@<node1\_ip>

* Configure the plugin:
  + Go to "Manage Jenkins" -> "Configure System".
  + Under "Publish over SSH", add node1's IP address and the private key (id\_rsa).

1. **Install Apache Server on Master**

* SSH into the master instance.
* Install Apache server:
* sudo apt update
* sudo apt install apache2
* sudo systemctl start apache2

sudo systemctl enable apache2

1. **Install Ansible on the Master Node**

* SSH into the master instance.
* Install Ansible:
* sudo apt update

sudo apt install ansible

* Configure node1 in the Ansible inventory:

sudo nano /etc/ansible/hosts

* Add the following:
* [webservers]

node1 ansible\_host=<node1\_ip> ansible\_user=<node1\_user>

1. **Create Jenkins User on Node1 and Setup SSH Access**

* SSH into node1.
* Create a Jenkins user
* sudo adduser jenkins
* Set up passwordless SSH for the Jenkins user:
* sudo mkdir /home/jenkins/.ssh
* sudo cp ~/.ssh/authorized\_keys /home/jenkins/.ssh/

sudo chown -R jenkins:jenkins /home/jenkins/.ssh

* Grant sudo access to Jenkins user:

sudo visudo

* Add the following line:

jenkins ALL=(ALL) NOPASSWD:ALL

1. **Write Ansible Playbook for Deployment**

* Create an Ansible playbook on the master node:

---

- name: Install Apache2, start service, copy file, and restart Apache2

hosts: webserver

become: true

tasks:

- name: Update the apt package index

apt:

update\_cache: yes

- name: Install Apache2 package

apt:

name: apache2

state: present

- name: Start and enable Apache2 service

service:

name: apache2

state: started

enabled: yes

- name: Copy file from Jenkins workspace to remote server

copy:

src: /var/lib/jenkins/workspace/website-update/index.html # Source file on the master node

dest: /var/www/html/index.html # Destination file on the remote node

owner: root

group: root

mode: '0644'

- name: Restart Apache2 service

service:

name: apache2

state: restarted

* Save this playbook as deploy.yml

1. **Create Jenkins Job to Deploy Website**

* In Jenkins, create a new job.
* Configure the job to use Git as the source code management.
* Set the repository URL to your GitHub repository.
* Add a build step to invoke the Ansible playbook

ansible-playbook /path/to/deploy.yml --ask-become-pass

1. **Set Up SCM Polling in Jenkins**

* In the Jenkins job configuration, enable "Poll SCM" and set the schedule to poll every minute:

\* \* \* \* \*

**10. Configure Email Notifications for Build Failures**

* In the Jenkins job configuration, go to the "Post-build Actions" section.
* Add "E-mail Notification" and configure it to send emails on build failures.

**11. Access the Deployed Website**

* Copy the IP address of node1.
* Paste the IP address into a web browser to access the newly deployed website.

