

CERTIFICATION PROJECT

SOLUTIONS

Capstone Project: DevOps Lifecycle Implementation

Name: Pradeepa k

Email: kpradeepacharya0@gmail.com

Location: Kasaragod, Kerala, India

Education: MCA (Master of Computer Applications)

Introduction

1.1 Introduction:

1.2 Project Title: Implementation of a DevOps Lifecycle for Abode Software's Web Application.

Abode Software, a company specializing in web applications, currently uses manual processes for development and deployment. This approach leads to inefficiencies, delays, and potential errors. To improve these processes, the company needs to implement an automated DevOps lifecycle. This project aims to streamline and automate software installation, build and test processes, and deployment using modern tools and practices. The goal is to enhance efficiency, reliability, and speed in delivering updates and new features.

1.3 Problem Statement

Exiting system: Abode Software currently relies on manual processes for development and deployment of their web application. The application code is hosted on GitHub, but there is no automated system in place for building, testing, and deploying the code. This results in inefficiencies, inconsistencies, and delays in delivering updates to production.

1.4 Issues:

- 1. Manual Software Installation:** The installation and configuration of necessary software on EC2 instances are done manually, which is time-consuming and prone to errors.
- 2. Lack of Automated Build and Testing:** There is no automated mechanism to trigger builds and tests based on code commits, leading to delayed and inconsistent testing and deployment.
- 3. Inefficient Code Containerization:** The application is not currently containerized, making deployment complex and inconsistent.
- 4. No Defined Git Workflow:** There is no established Git workflow to manage different branches (e.g., master for production and develop for staging), resulting in uncoordinated development and deployment processes.

1.5 Requirements:

-> **Automate Software Installation:** Use a configuration management tool (Ansible) to automate the installation and configuration of necessary software on EC2 instances.

-> **Implement Git Workflow:** Establish a Git workflow with master and develop branches, and configure AWS CodeBuild to automatically trigger builds and tests based on commits:

- For the master branch: Test the code and push to production if tests pass.
- For the develop branch: Test the code without pushing to production.

-> **Containerize the Application:** Use Docker to containerize the application code, utilizing the pre-built container hshar/webapp and ensuring that the Dockerfile is built on each GitHub push. The application code should reside in /var/www/html.

-> **Define Jenkins Pipeline:** Create a Jenkins pipeline with the following jobs:

- **Job1:** Build
- **Job2:** Test
- **Job3:** Deploy to production (for master branch commits)

1.6 Objective:

The objective of this project is to implement a fully automated DevOps lifecycle for Abode Software's web application. The lifecycle will include software installation and configuration, implementation of a Git workflow, automated builds and testing, containerization of the application, and deployment processes. The implementation will leverage industry-standard tools such as Ansible, Git, AWS CodeBuild, Docker, and Jenkins.

1.7 Scope:

1.7.1 Configuration Management:

Utilize a configuration management tool (Ansible) to install and configure the necessary software on the designated machines (EC2 instances). This includes setting up environments for the master node and slave nodes.

1.7.2 Git Workflow Implementation:

Establish a Git workflow with two primary branches: master and develop.

Define branch-specific policies to ensure that commits to these branches trigger appropriate build and deployment actions.

1.7.3 Automated Build and Testing with CodeBuild:

Integrate AWS CodeBuild to automatically trigger builds based on commits to the master or develop branches.

For the master branch:

Test the application.

If successful, push the tested code to production.

For the develop branch:

Test the application.

Do not push to production; this branch is for staging and development purposes only.

1.7.4 Containerization with Docker:

Containerize the application code using Docker.

Utilize a pre-built container (hshar/webapp) for the application.

Ensure that the Dockerfile is built and the container is created every time there is a push to GitHub.

The application code should be placed in the directory /var/www/html within the container.

1.7.5 Jenkins Pipeline Setup:

Define a Jenkins pipeline with the following jobs:

Job1: Build: Compile and build the application code.

Job2: Test: Run automated tests to ensure code quality and functionality.

Job3: Prod: Deploy the tested application to production if it passes all tests (triggered by commits to the master branch).

1.8 Tools and Platforms Used:

AWS Cloud: Provides the infrastructure for deploying and managing the application and services, including EC2 instances for hosting and running the DevOps tools.

Jenkins: Used for automating the build, test, and deployment processes through a continuous integration and continuous deployment (CI/CD) pipeline.

Docker: Facilitates containerization of the application, ensuring consistency across different environments and simplifying deployment.

Docker Hub: A registry for storing and managing Docker images, used to pull pre-built containers and push custom images.

GitHub: Hosts the application code and version control, integrating with other tools for automated builds and deployments based on code changes.

Ansible: Manages configuration and automation of software installation and setup on the EC2 instances, streamlining environment provisioning and maintenance.

SOLUTION

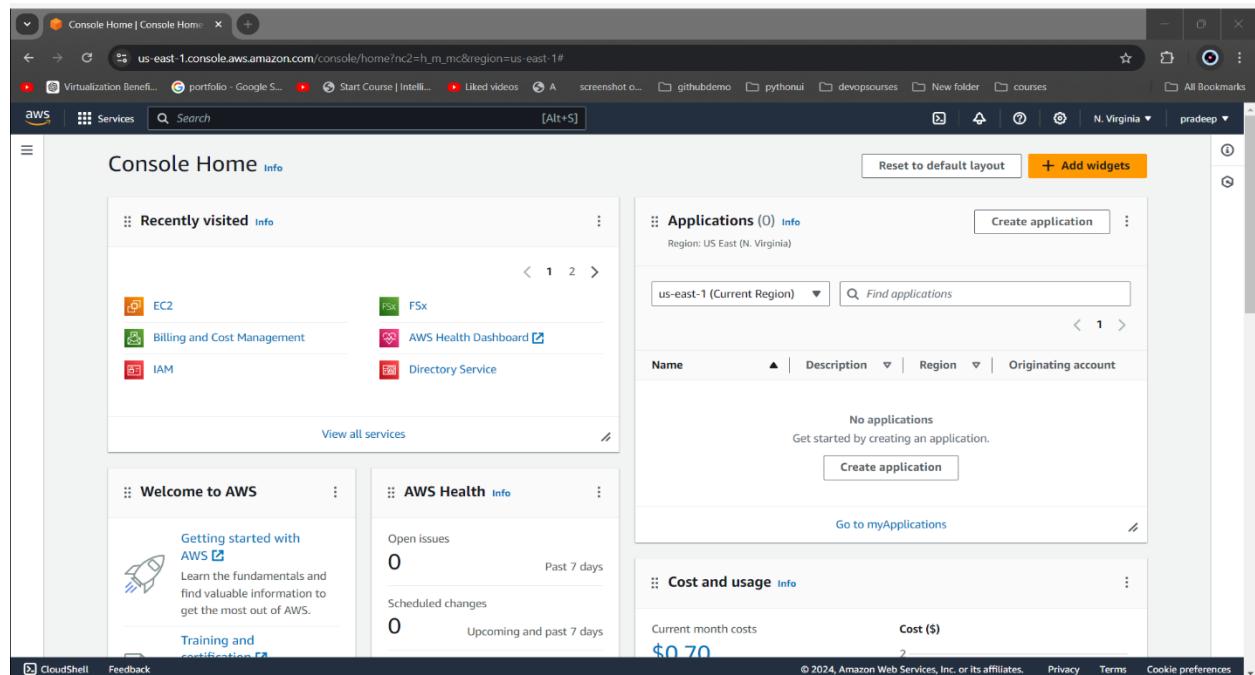
1. AWS Environment is a setup:

- Launch EC2 instances for Jenkins, Ansible, and application servers.
- Install basic software on the instances (e.g., Docker, Git, Jenkins, Ansible).

As we are utilizing AWS, the next step is to launch instances. We'll be setting up an instance for Jenkins and Ansible.

In addition, we need to provision two instances as Jenkins slave nodes to handle job execution.

First, we'll configure the instances, followed by launching the instance designated for Jenkins and Ansible.



Click on 'Launch Instance' to begin creating a new instance.

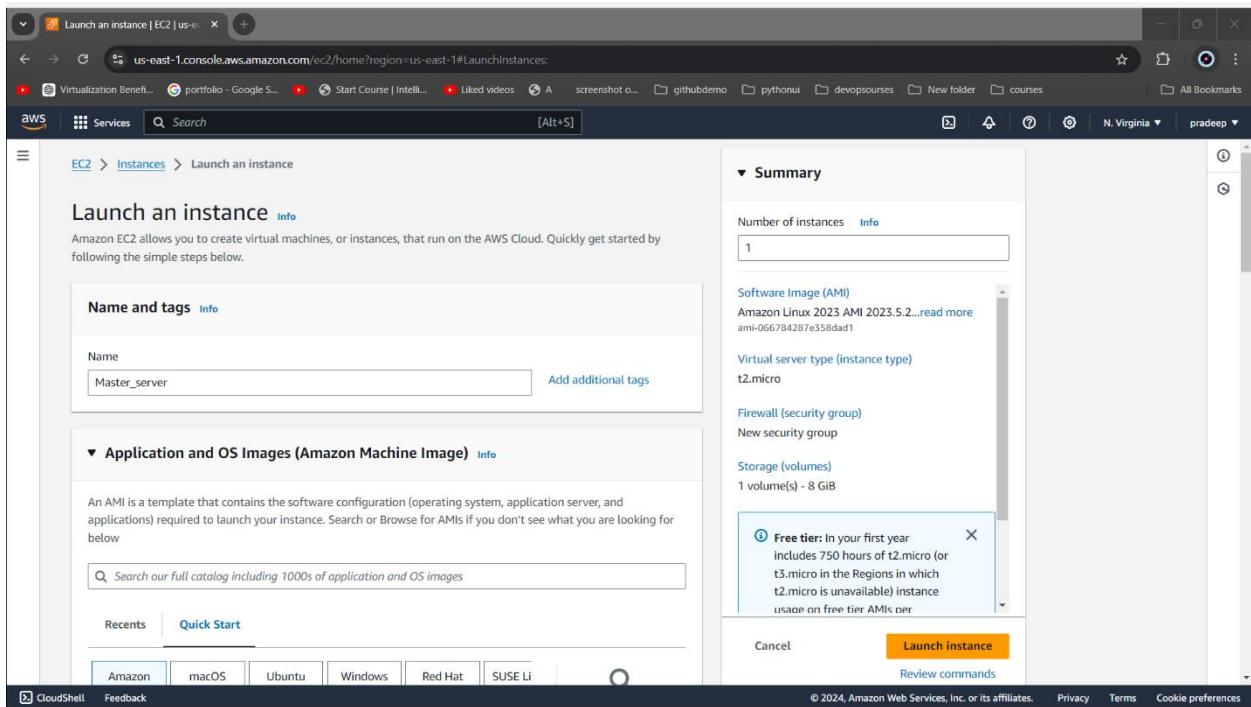
The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area has a 'Resources' section with various metrics (Instances running: 0, Auto Scaling Groups: 0, Capacity Reservations: 0, etc.). Below it is a 'Launch instance' section with a large orange 'Launch instance' button. To the right are sections for Service health (AWS Health Dashboard), Zones, and Account attributes. A sidebar on the right displays 'EC2 Free Tier' information and offers.

The screenshot shows the 'Launch an instance' wizard. It starts with a 'Launch an instance' step, followed by a 'Name and tags' step where 'e.g. My Web Server' is entered. Then it moves to an 'Application and OS Images (Amazon Machine Image)' step, which includes a search bar and a 'Quick Start' tab. On the right, a 'Summary' panel shows the configuration: 1 instance, Amazon Linux 2023.5.1 AMI, t2.micro instance type, New security group, and 1 volume(s) - 8 GiB. A tooltip for the free tier is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month.' At the bottom are 'Cancel', 'Launch instance', and 'Review commands' buttons.

We can now observe that several instance attributes require configuration.

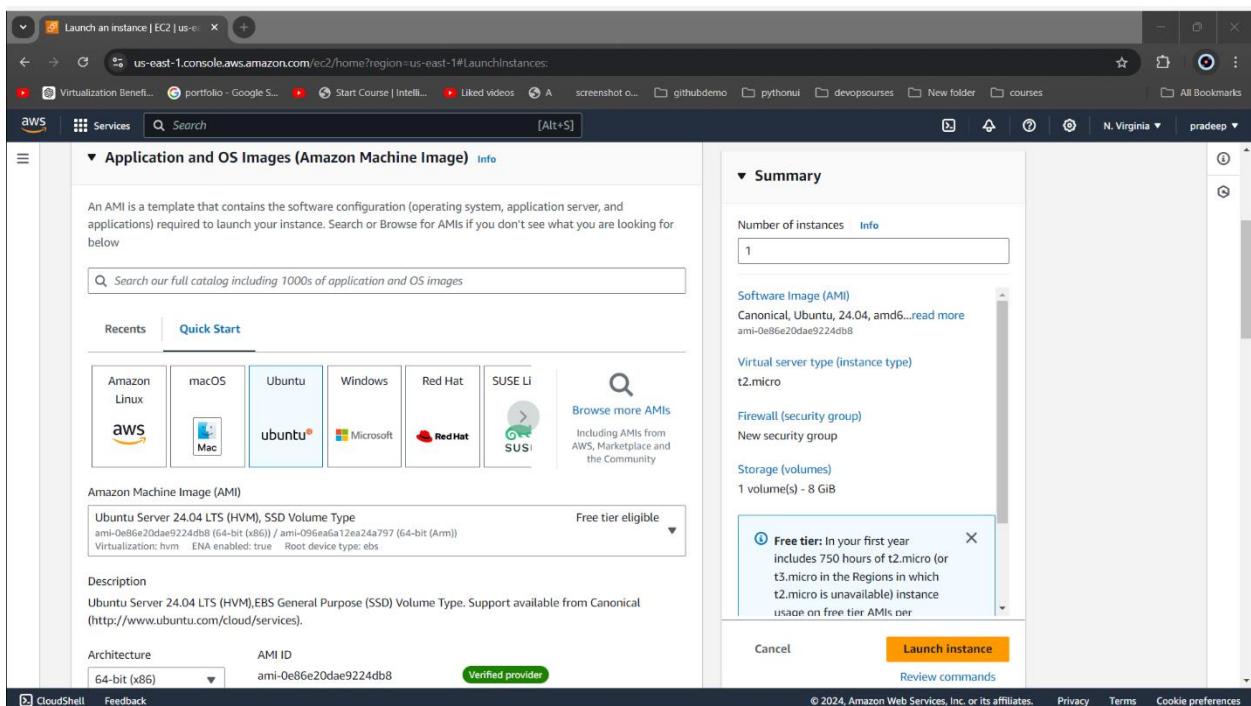
The first step is to provide a name for the EC2 instance.

Assign the name 'Master Server' to the instance.



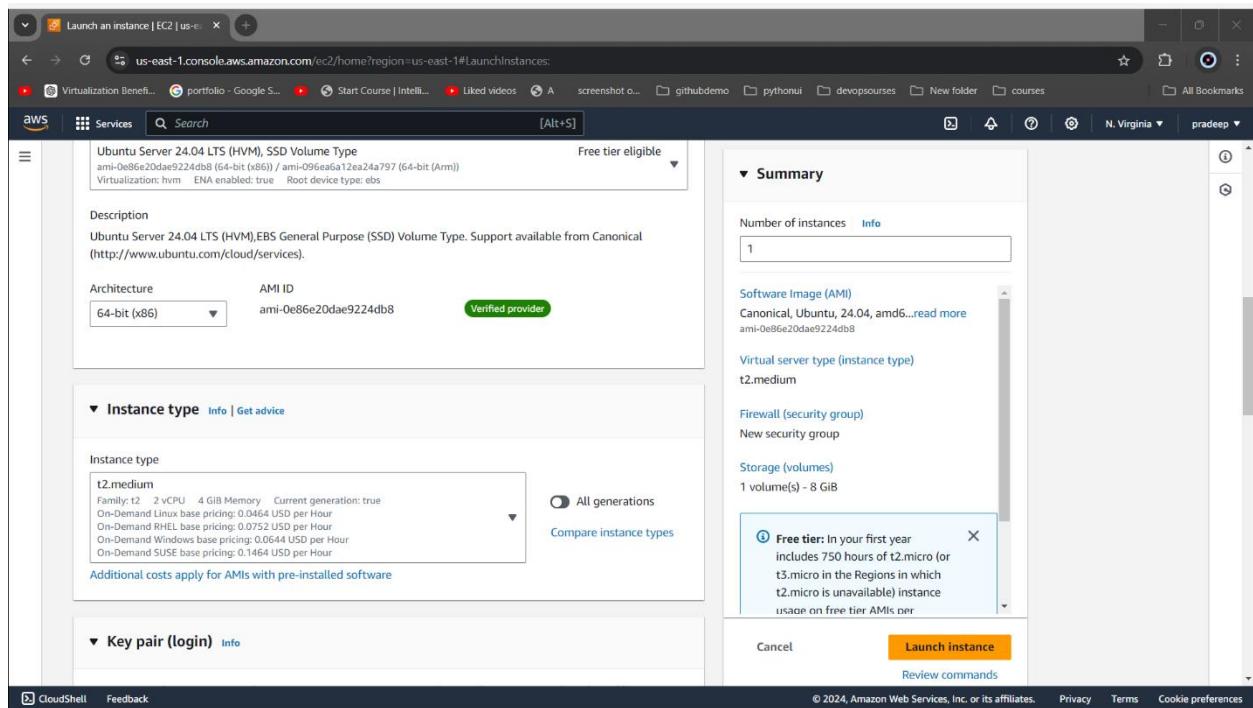
The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. In the 'Name and tags' section, the 'Name' field contains 'Master_server'. The 'Summary' panel on the right shows a configuration for 1 instance of the Amazon Linux 2023 AMI (ami-066784287e358dad1) in a t2.micro instance type, associated with a new security group, and using 1 volume(s) - 8 GiB. A tooltip for the 'Free tier' is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per account per year.' A prominent orange 'Launch instance' button is at the bottom right.

Specify the AMI as Ubuntu for this instance.

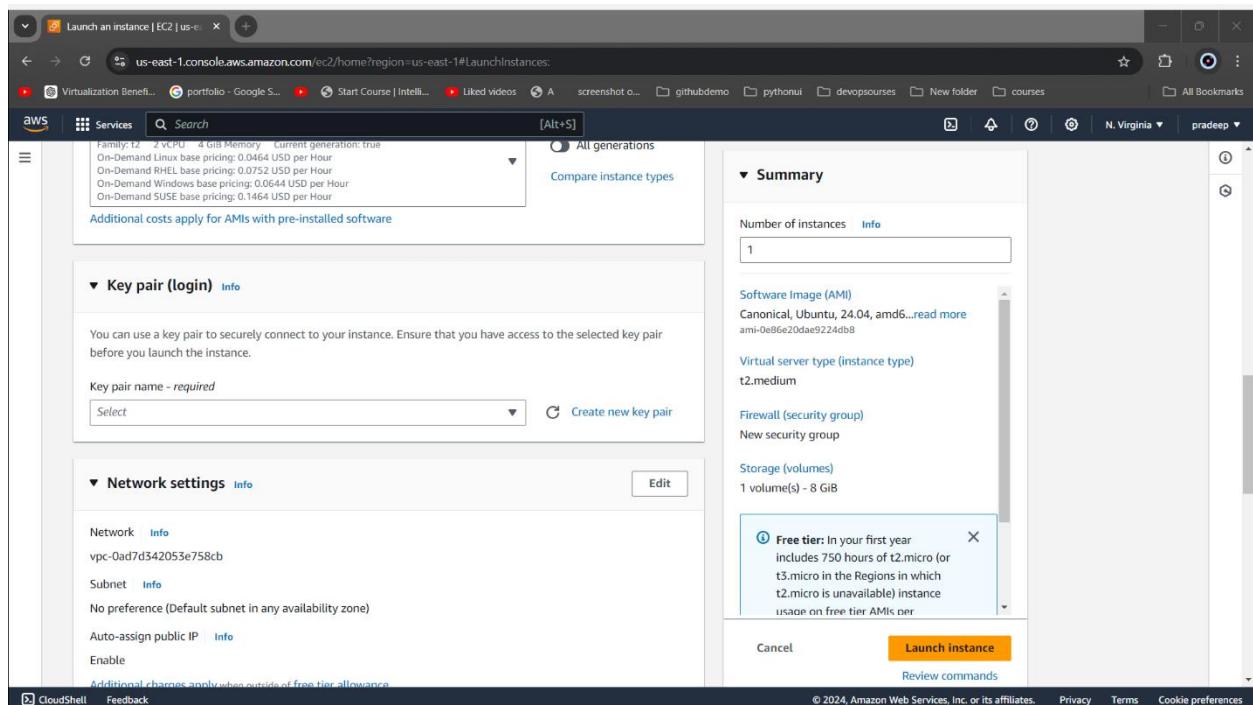


The screenshot shows the 'Launch an instance' wizard with the 'Application and OS Images (Amazon Machine Image)' section active. Under the 'Recent' tab, 'Ubuntu' is selected. The 'Summary' panel on the right shows the same configuration as the previous screenshot, but with 'Canonical, Ubuntu, 24.04, amd64' selected as the Software Image (AMI). A tooltip for the 'Free tier' is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per account per year.' A prominent orange 'Launch instance' button is at the bottom right.

Next, we need to specify the instance type, which will be t2.medium.

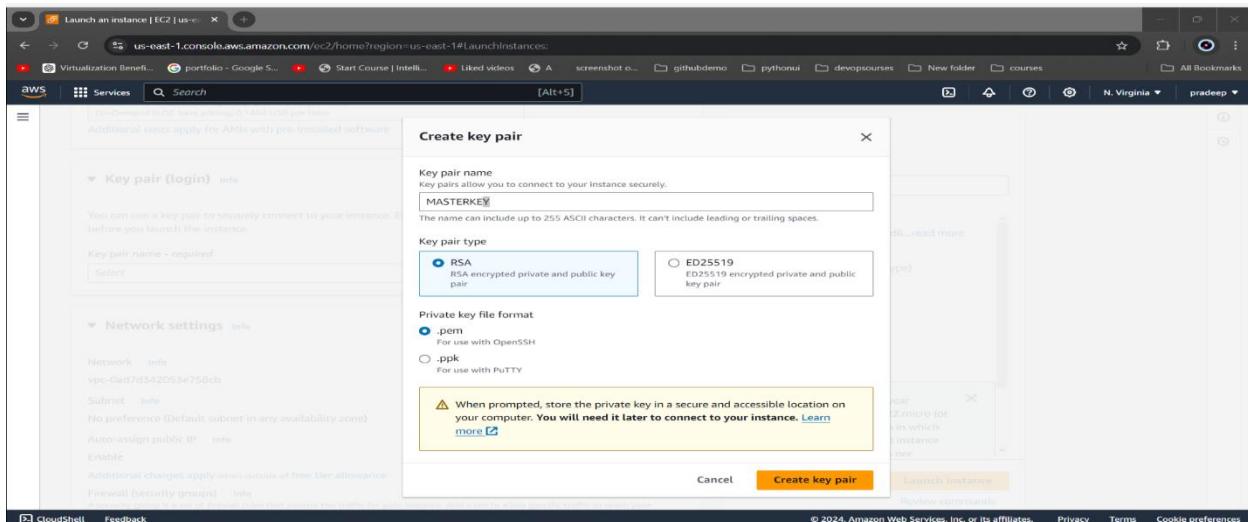


Generate a new key pair for this instance and associate it with the instance.



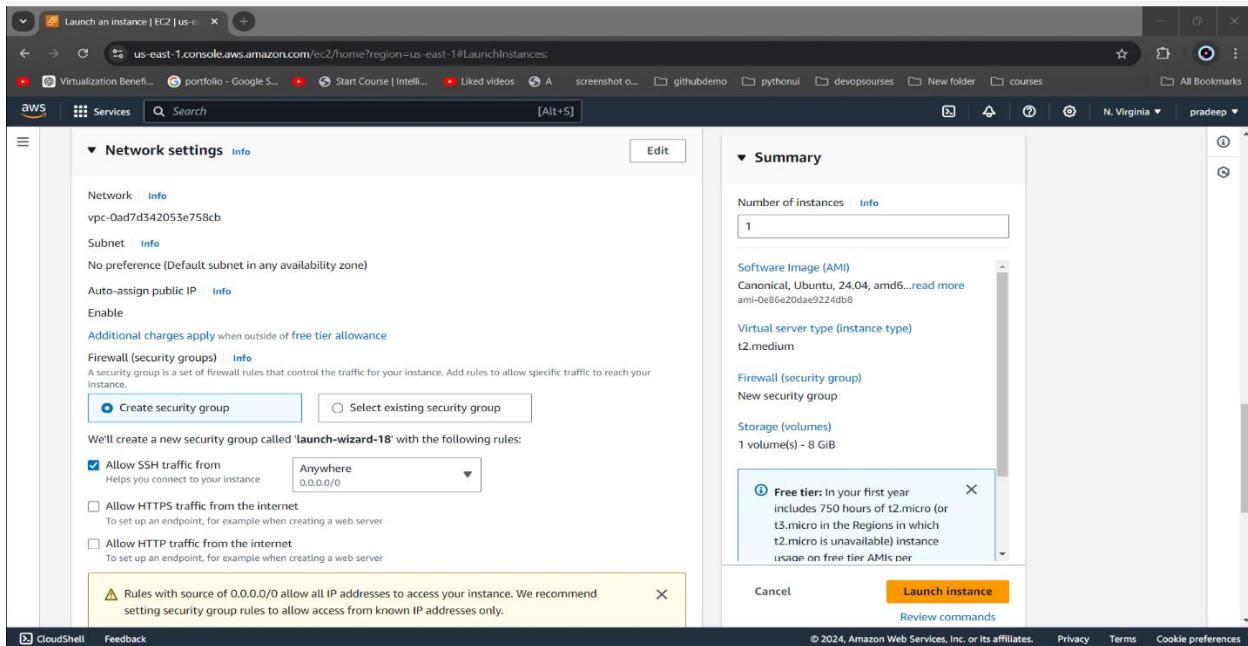
To create a new EC2 instance, select the 'Create New Key Pair' option. A window will pop up where you'll need to enter the necessary details, and the key pair file type will be .pem.

Name: MASTERKEY

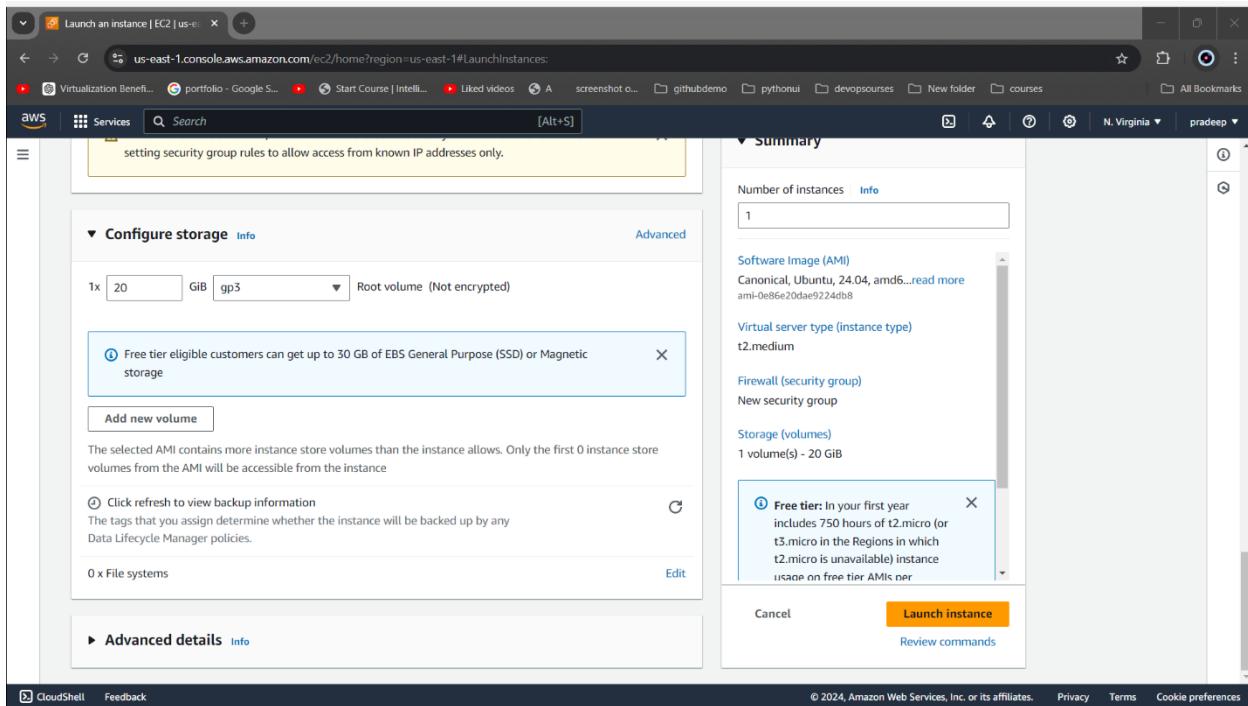


Select 'Create Key Pair' to generate a new key pair.

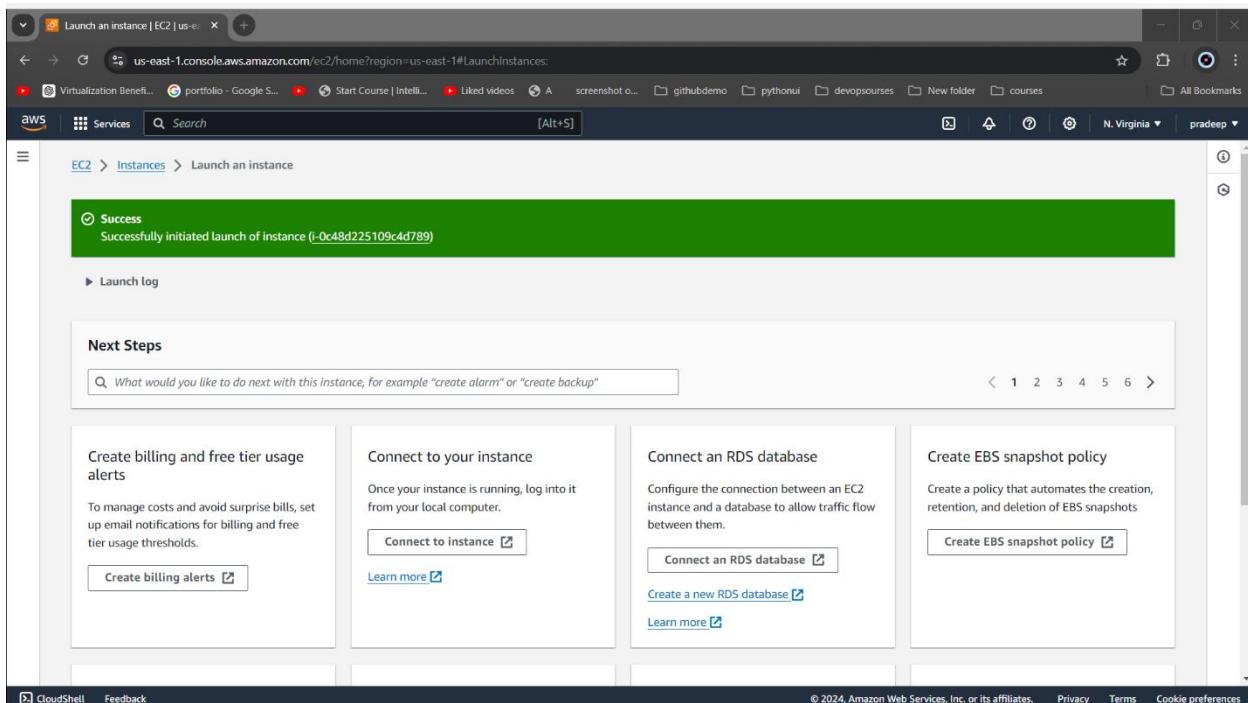
Leave the network settings at their default values, as no modifications are necessary.



Set the storage size to 20 GB.



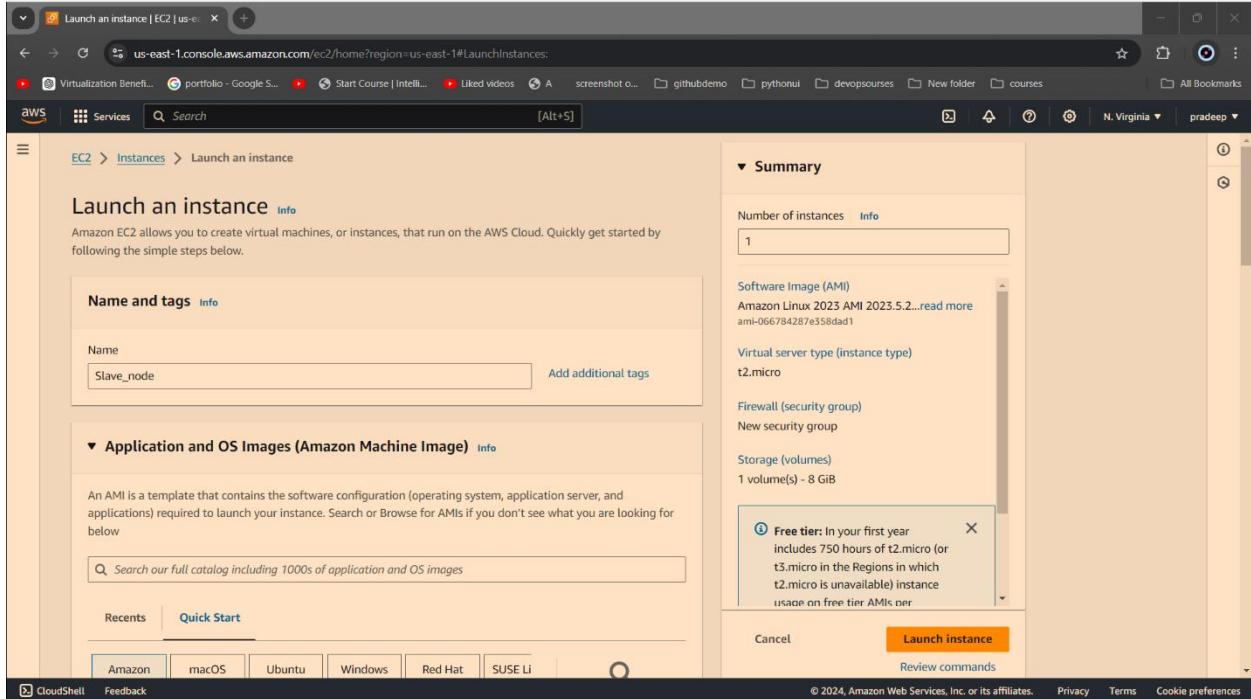
Select 'Launch Instance' to complete the creation of the instance.



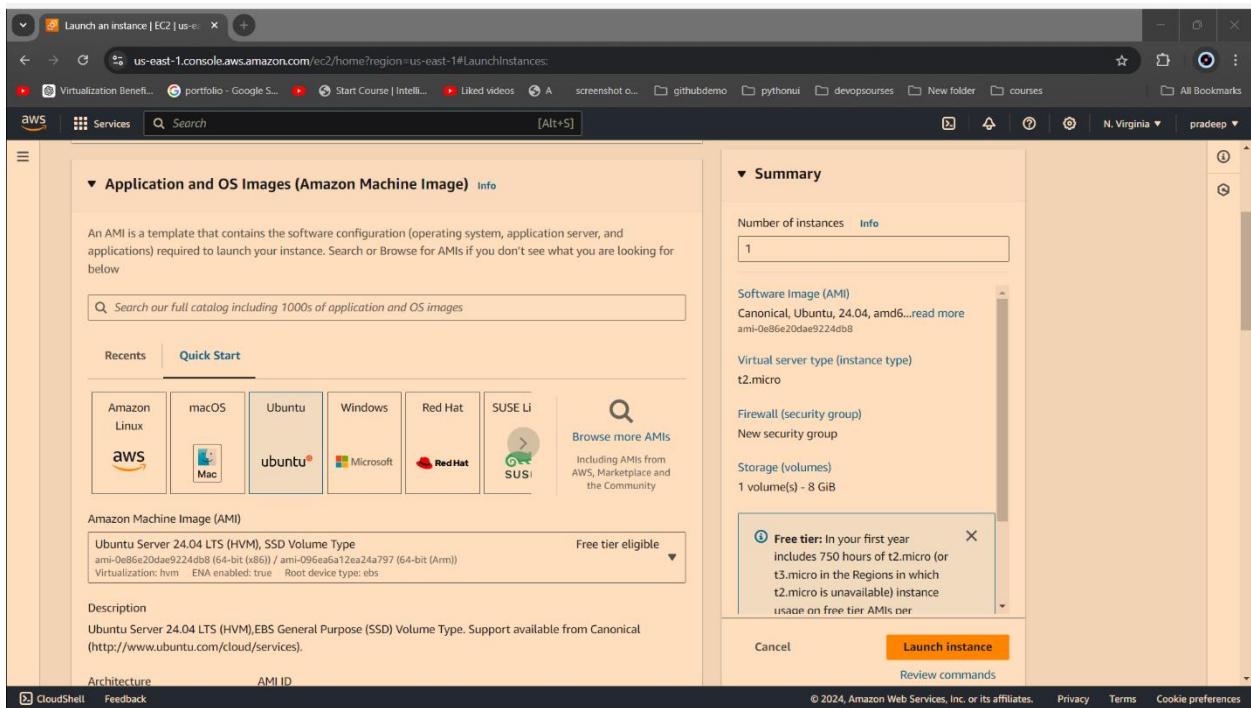
The instance has been successfully created and is currently running.

The next step is to launch two EC2 instances as slave nodes.

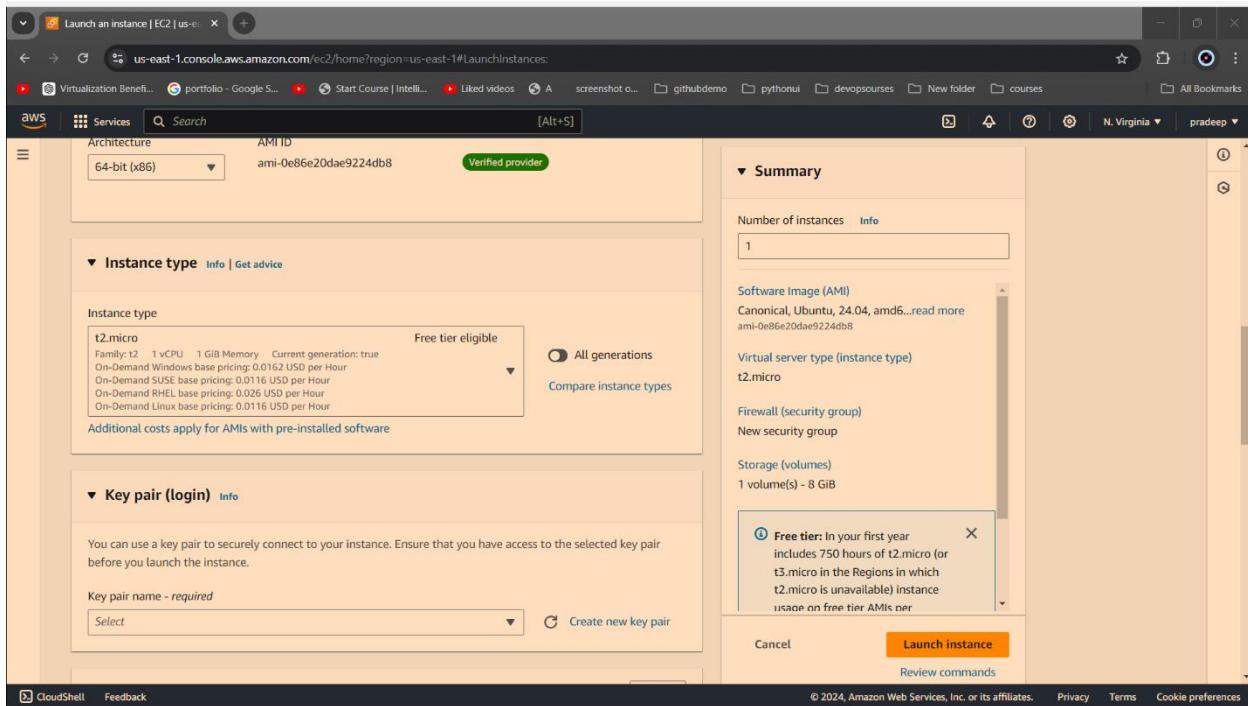
Assign the name 'Slave Nodes' to the EC2 instances. We can always edit the names after the instances are launched.



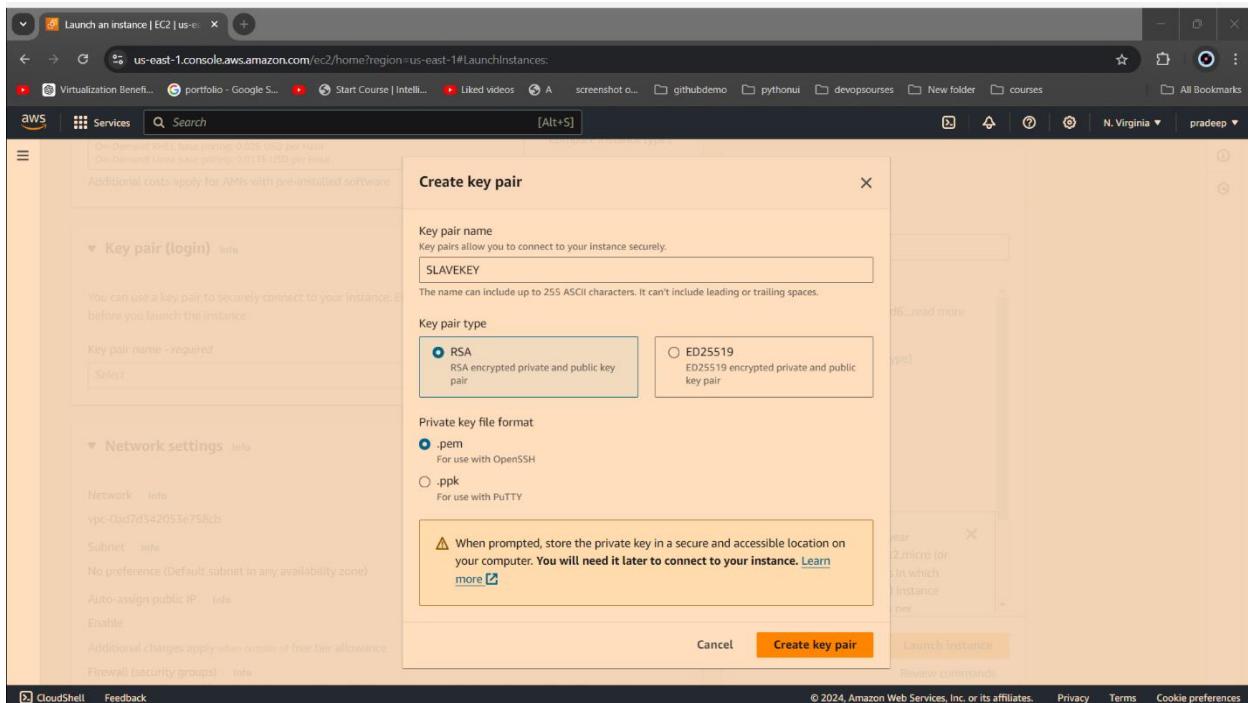
Choose Ubuntu as the AMI image.

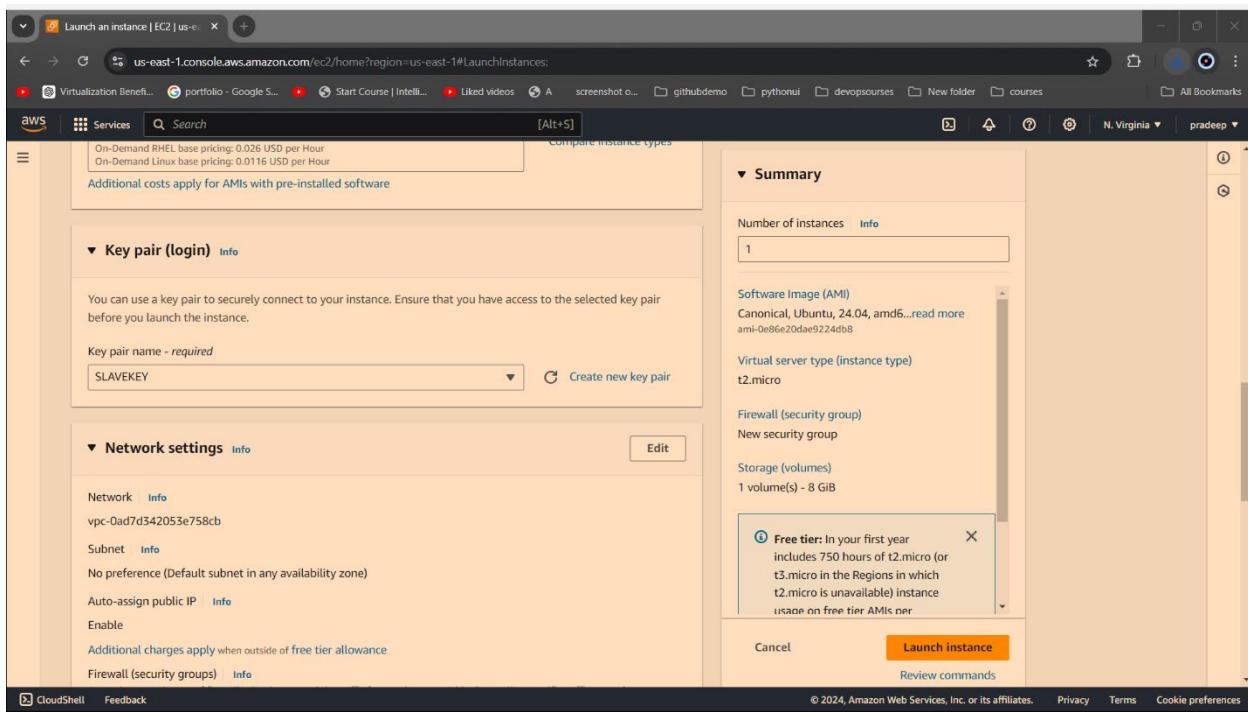


Select t2.micro as the instance type.

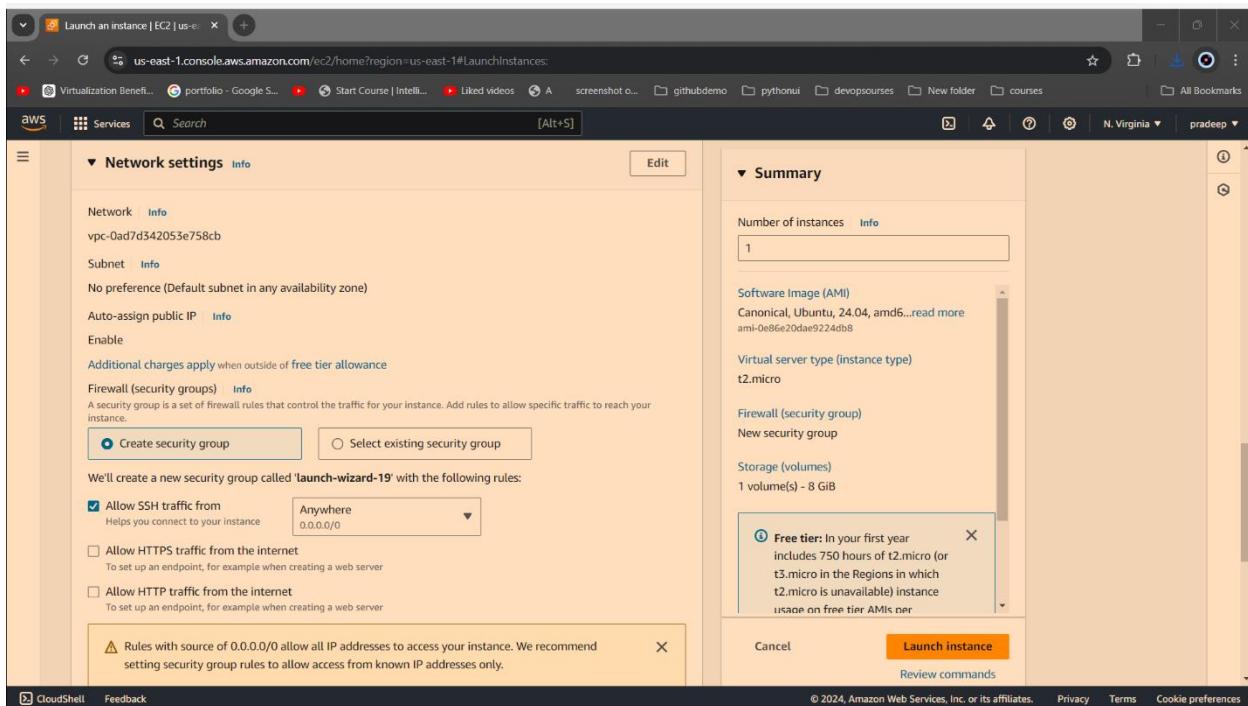


Generate a new key pair.

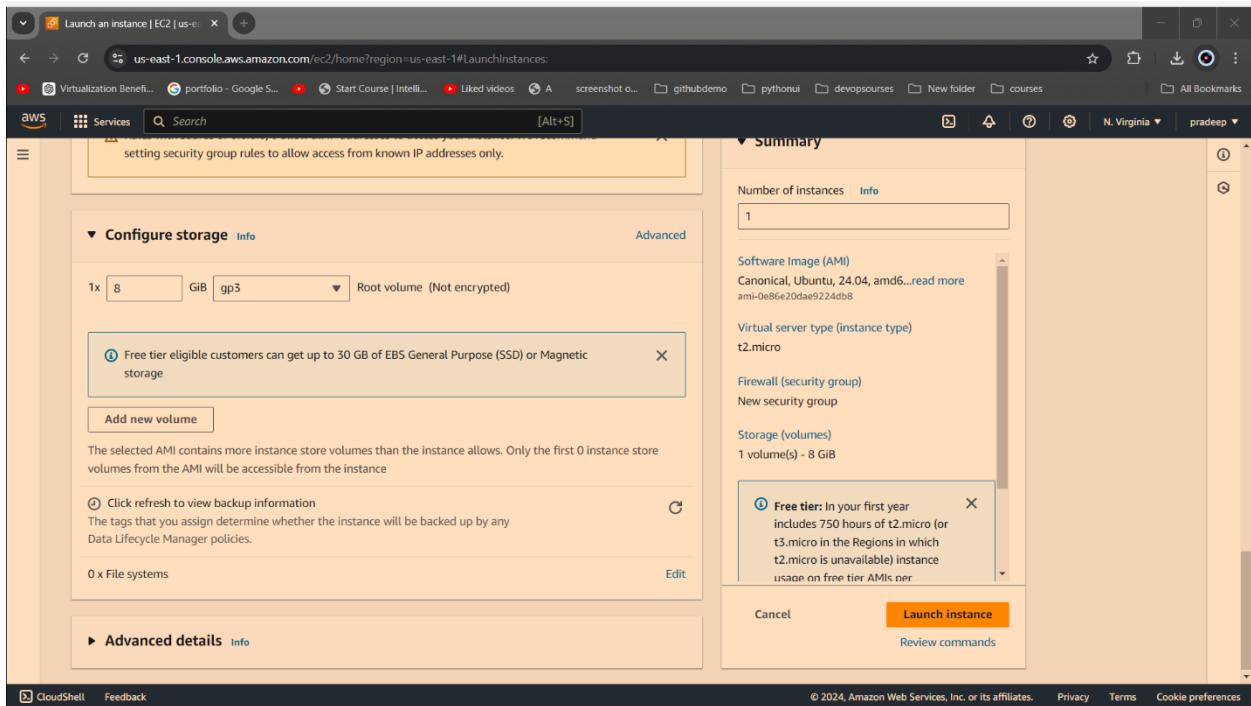




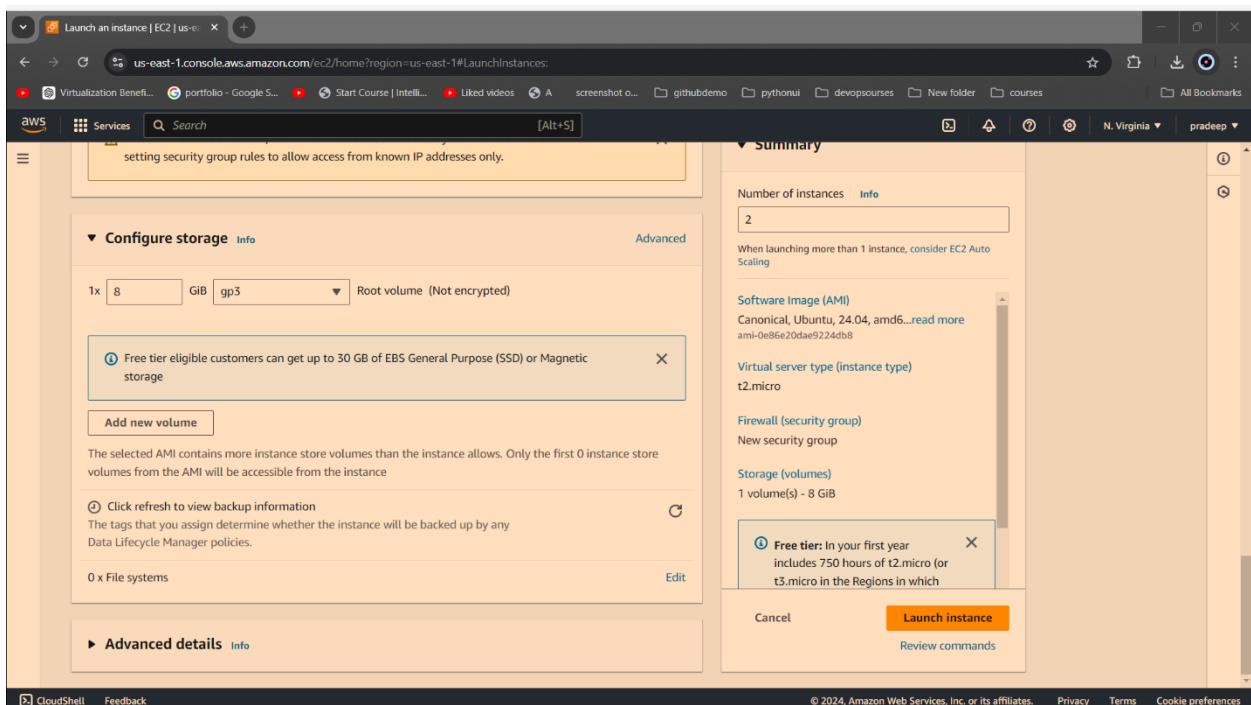
Leave the network settings at their default configuration.



Leave the storage size at the default setting.



Set the number of instances to 2.



Click 'Launch Instance' to create the EC2 instance.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Slave_Node	i-076061495bb4039d5	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-44-20
Master_server	i-0c48d225109c4d789	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1a	ec2-18-23
Slave_node	i-0cd0f347cb719fc81	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-3-87-2

The instances are successfully created. Rename them to 'slave_node' to 'manage_node1' and 'manage_node2' as needed.

The screenshot shows the AWS EC2 Instances page with four instances selected: Controll_node, Manage_Node1, and Manage_Node2. The left sidebar and main content area are similar to the previous screenshot, but the instance names have been renamed. The monitoring dashboard shows CPU utilization, network in/out, and network packets for each instance.

The configuration of the EC2 instances is finished.

2. Configuration Management with Ansible:

Ansible Setup:

→ Master_Node:

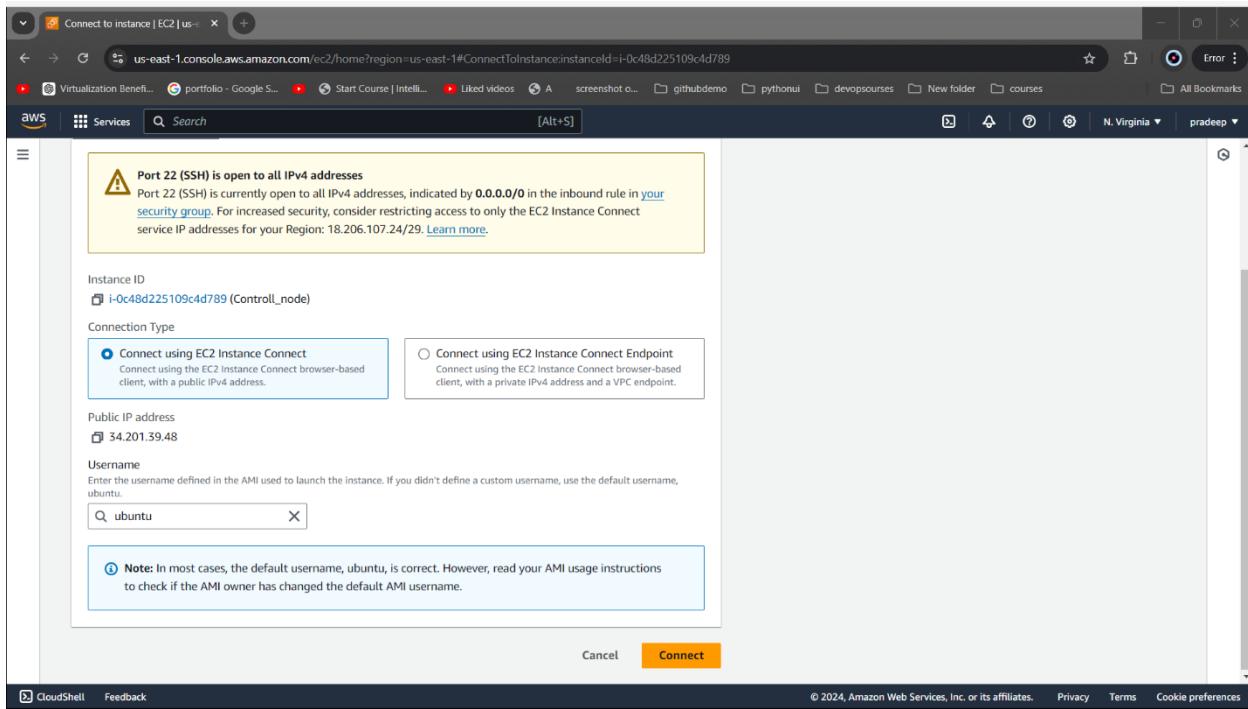
- Install Ansible on the control node EC2 instance.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Instances, Images, and Elastic Block Store. The main area displays a table titled 'Instances (3/3) Info' with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. Three instances are selected: 'Controll_node' (t2.medium), 'Manage_Node1' (t2.micro), and 'Manage_Node2' (t2.micro). Below the table, a message says '3 instances selected'. Further down, there's a monitoring section with CPU utilization, Network in, Network out, and Network packets in metrics. At the bottom right, there are links for 'Configure CloudWatch Metrics' and 'Add to CloudWatch Metrics'.

First, let's establish a connection to the master node.

Choose the master node, click 'Connect,' and then proceed by selecting 'Connect' again.

This screenshot shows the same AWS EC2 Instances page as the previous one, but with a modal window open over it. The modal title is 'Successfully initiated starting of i-0c48d225109c4d789,i-076061495bb4039d5,i-0cd0f347cb719fc81'. It contains a table with the same three instances. The 'Controll_node' row has its 'Connect' button highlighted. Below the table, there's a detailed view for the 'Controll_node' instance, showing its Public IPv4 address (34.201.59.48), Instance state (Running), and Private IP DNS name (ip-172-31-85-119.ec2.internal). There are also sections for Public IPv4 addresses and Public IPv4 DNS.



Connection to the master node established successfully.

```
Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.
https://ubuntu.com/ubuntu/pro

Expanded Security Maintenance for Applications is not enabled.

84 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

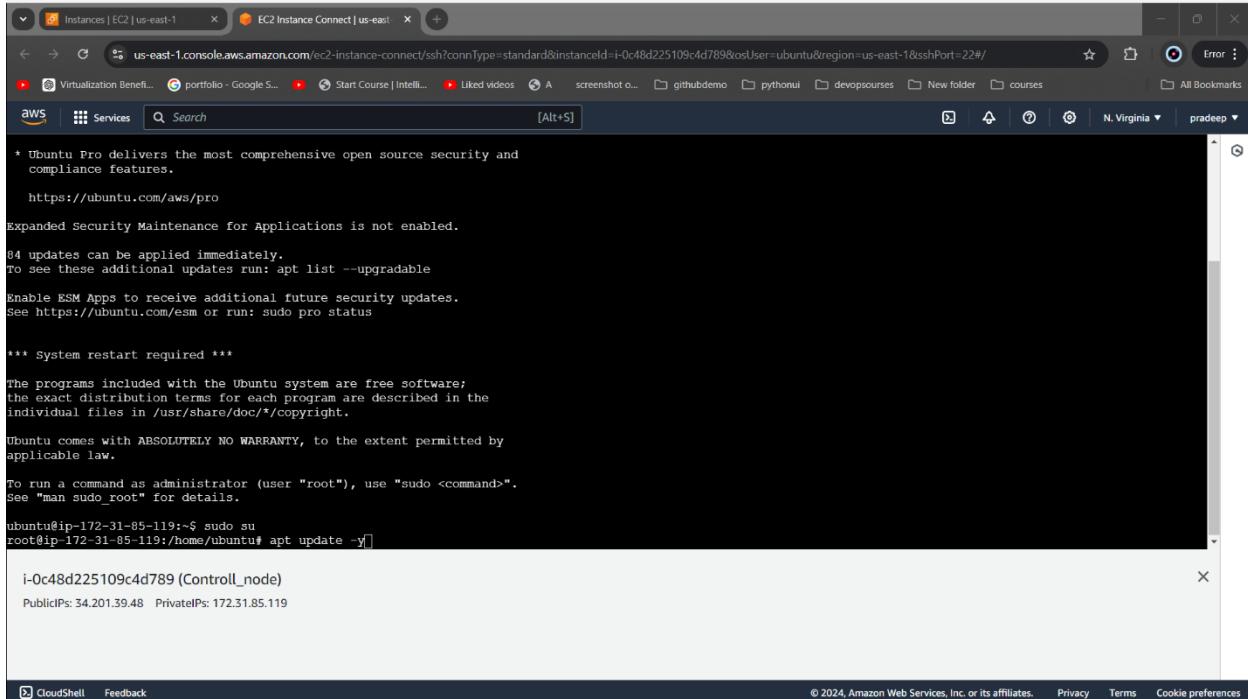
ubuntu@ip-172-31-85-119:~$
```

i-0c48d225109c4d789 (Controll_node)
PublicIP: 34.201.39.48 PrivateIP: 172.31.85.119

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

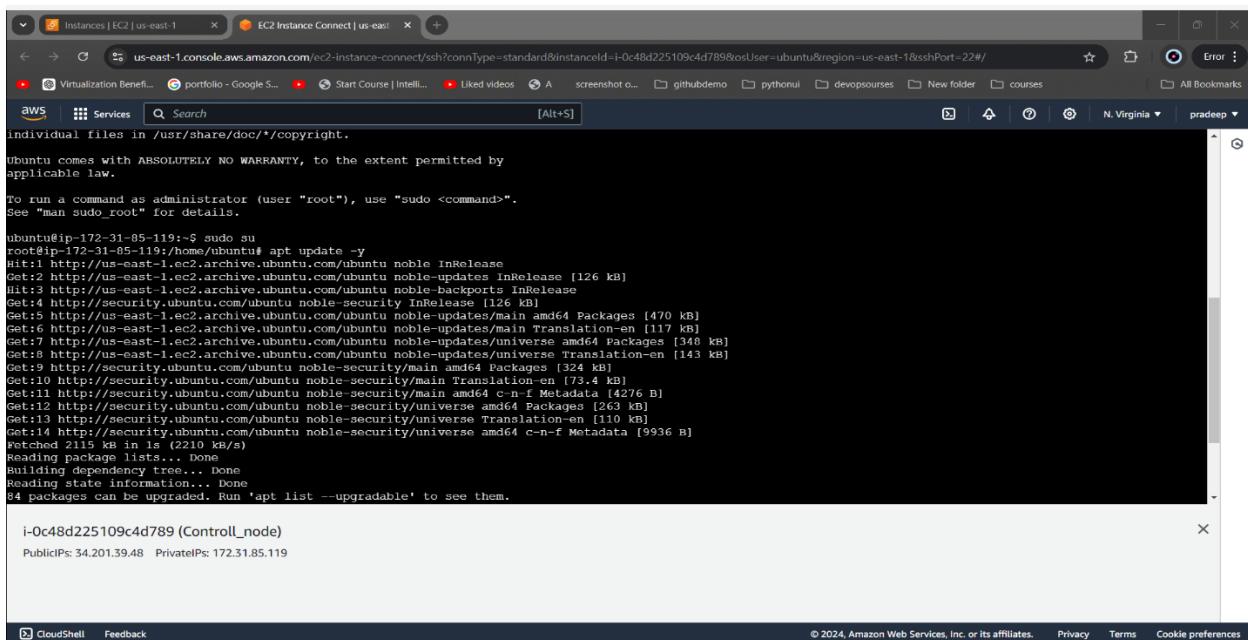
- Execute the command `apt update -y` next:

The `apt update` command is used to update the local package index on a system that uses APT (Advanced Package Tool) as its package manager. When you run `apt update`, it refreshes the list of available packages and their versions from the repositories configured on your system.



A screenshot of a web browser window titled "Instances | EC2 | us-east-1" and "EC2 Instance Connect | us-east-1". The address bar shows "us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu®ion=us-east-1&sshPort=22#/" and the tab title is "EC2 Instance Connect | us-east-1". The browser interface includes a navigation bar with links like "Virtualization Benefits", "portfolio - Google S...", "Start Course | Intelli...", "Liked videos", "A", "screenshot o...", "githubdemo", "pythonui", "devopsources", "New folder", "courses", and "All Bookmarks". Below the navigation bar is a search bar with "aws" and "Services" selected. The main content area is a terminal window showing the output of the `apt update -y` command. The output includes messages about security features, upgrade availability, and a required system restart. It also shows the command being run and the resulting update process. At the bottom of the terminal window, it says "i-0c48d225109c4d789 (Controll_node)" and "PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119". The footer of the browser window includes "CloudShell Feedback" and copyright information: "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

```
* Ubuntu Pro delivers the most comprehensive open source security and compliance features.  
https://ubuntu.com/aws/pro  
Expanded Security Maintenance for Applications is not enabled.  
84 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
*** System restart required ***  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-172-31-85-119:~$ sudo su  
root@ip-172-31-85-119:/home/ubuntu# apt update -y  
i-0c48d225109c4d789 (Controll_node)  
PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119
```



A screenshot of a web browser window titled "Instances | EC2 | us-east-1" and "EC2 Instance Connect | us-east-1". The address bar shows "us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu®ion=us-east-1&sshPort=22#/" and the tab title is "EC2 Instance Connect | us-east-1". The browser interface includes a navigation bar with links like "Virtualization Benefits", "portfolio - Google S...", "Start Course | Intelli...", "Liked videos", "A", "screenshot o...", "githubdemo", "pythonui", "devopsources", "New folder", "courses", and "All Bookmarks". Below the navigation bar is a search bar with "aws" and "Services" selected. The main content area is a terminal window showing the detailed output of the `apt update -y` command. The output lists numerous package updates, including security patches and translations, from various repositories like "noble-security", "main", "universe", and "multiverse". The terminal window shows the progress of the download and extraction of these packages. At the bottom, it indicates "2115 kB in 1s (2210 kB/s)", "Reading package lists... Done", "Building dependency tree... Done", "Reading state information... Done", and "84 packages can be upgraded. Run 'apt list --upgradable' to see them.". The footer of the browser window includes "CloudShell Feedback" and copyright information: "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

```
individual files in /usr/share/doc/*copyright.  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-172-31-85-119:~$ sudo su  
root@ip-172-31-85-119:/home/ubuntu# apt update -y  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [470 kB]  
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [117 kB]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [346 kB]  
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [143 kB]  
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [324 kB]  
Get:10 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [73.4 kB]  
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4276 B]  
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [263 kB]  
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [110 kB]  
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [9936 B]  
Fetched 2115 kB in 1s (2210 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
84 packages can be upgraded. Run 'apt list --upgradable' to see them.  
i-0c48d225109c4d789 (Controll_node)  
PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119
```

- Visit this site for instructions on installing Ansible on the controller node.

https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu

The screenshot shows a web browser window with three tabs open: "Instances | EC2 | us-east-1", "EC2 Instance Connect | us-east-1", and "Installing Ansible on specific operating systems | docs.ansible.com". The main content area is titled "Installing Ansible on Ubuntu". It contains a sidebar with links like "Installing Ansible on Ubuntu", "Configuring Ansible", "Ansible Porting Guides", "USING ANSIBLE", "CONTRIBUTING TO ANSIBLE", and "EXTENDING ANSIBLE". The main content area has two sections: "Installing Ansible on Ubuntu" and "Installing Ansible on Debian". The "Installing Ansible on Ubuntu" section includes a note about PPA availability and a command example:

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```

Below this, there's a "Note" section with a warning about older Ubuntu distributions and a table comparing Debian and Ubuntu versions:

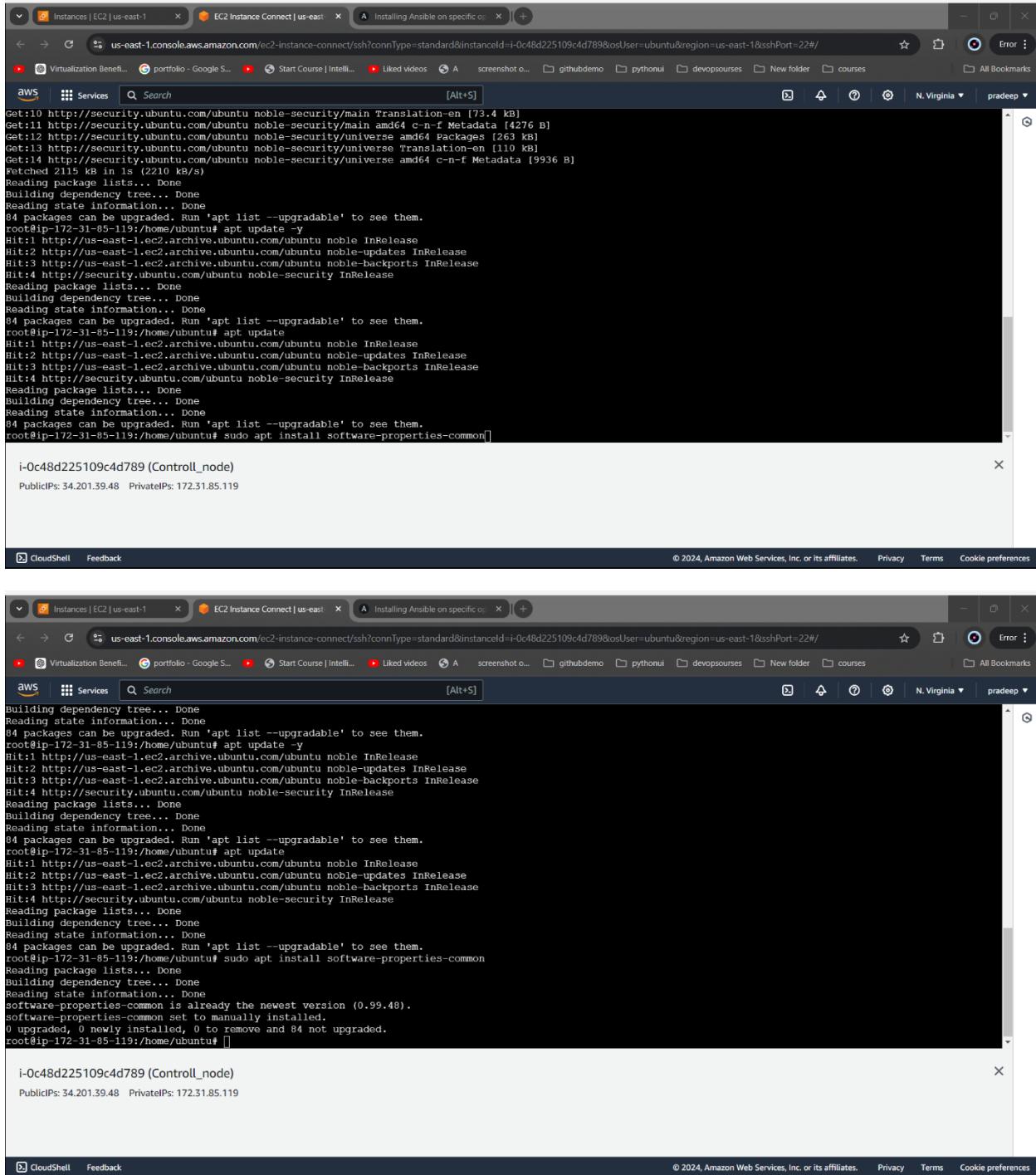
Debian	Ubuntu	UBUNTU_CODENAME
Debian 12 (Bookworm)	>->	Jammy
Debian 11 (Bullseye)	>->	Focal

- sudo apt update
- sudo apt install software-properties-common
- sudo add-apt-repository --yes --update ppa:ansible/ansible
- sudo apt install ansible

To install Ansible, run the required commands. As Ansible is a third-party tool, its repository needs to be added to our package manager.

After executing the apt update command, proceed with running the remaining commands.

→ sudo apt install software-properties-common

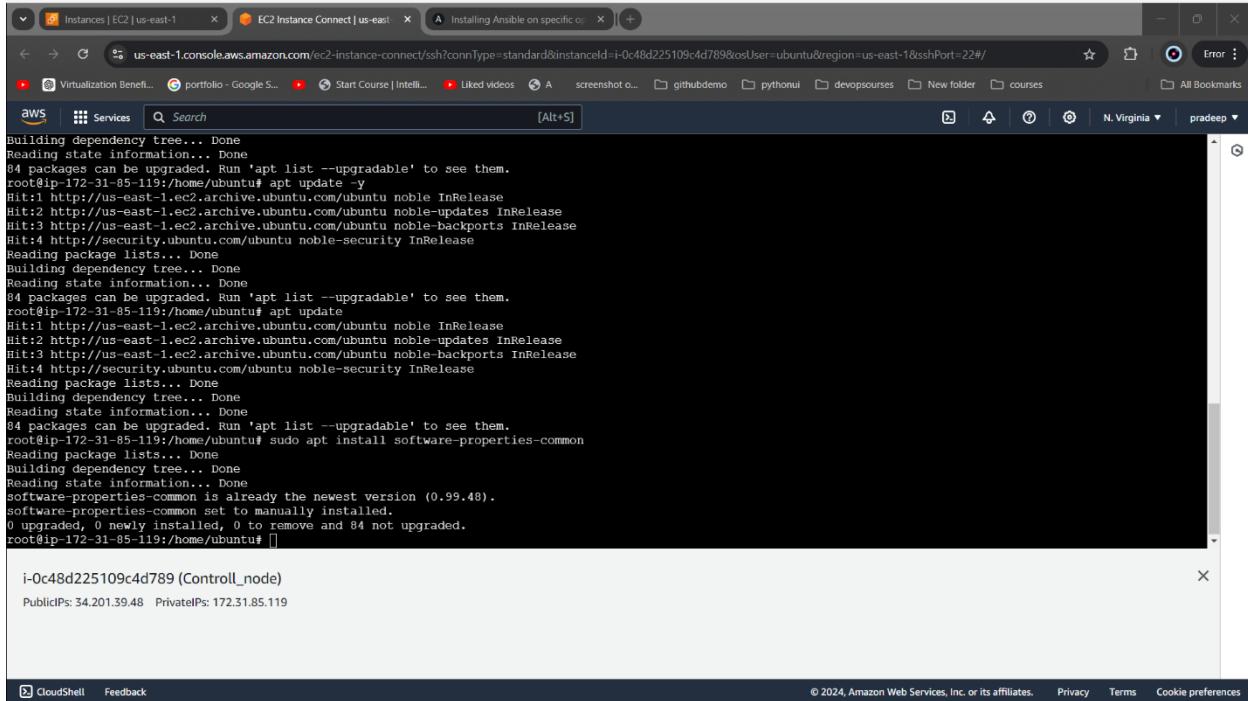


```
Get:10 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [73.4 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4276 B]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [263 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [110 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [9936 B]
Fetched 2115 kB in 1s (2210 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
84 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-85-119:/home/ubuntu# apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
84 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-85-119:/home/ubuntu# sudo apt install software-properties-common[]

i-0c48d225109c4d789 (Controll_node)
Public IPs: 34.201.39.48 Private IPs: 172.31.85.119

CloudShell Feedback
```

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```
building dependency tree... Done
Reading state information... Done
84 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-85-119:/home/ubuntu# apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
84 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-85-119:/home/ubuntu# sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
84 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-85-119:/home/ubuntu# sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.48).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 84 not upgraded.
0 upgraded, 0 newly installed, 0 to remove and 84 not upgraded.
root@ip-172-31-85-119:/home/ubuntu# []

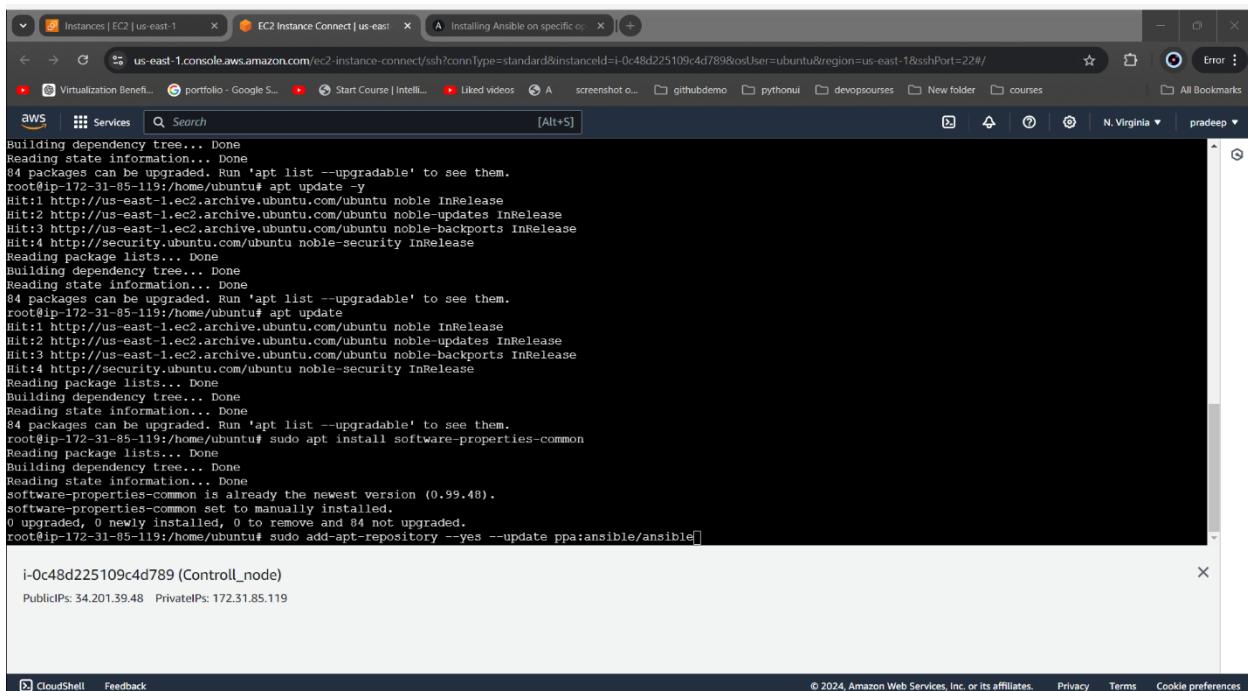
i-0c48d225109c4d789 (Controll_node)
Public IPs: 34.201.39.48 Private IPs: 172.31.85.119

CloudShell Feedback
```

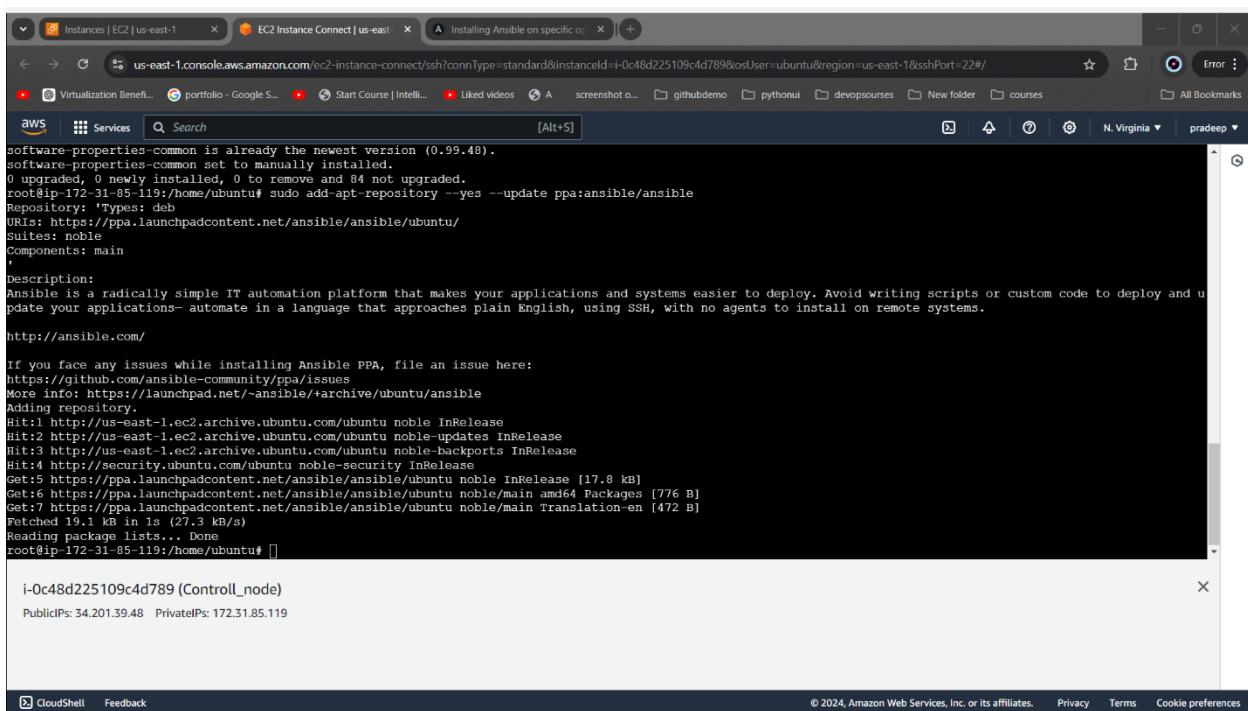
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

This command adds the Ansible repository to our package manager.

→ sudo add-apt-repository --yes --update ppa:ansible/ansible



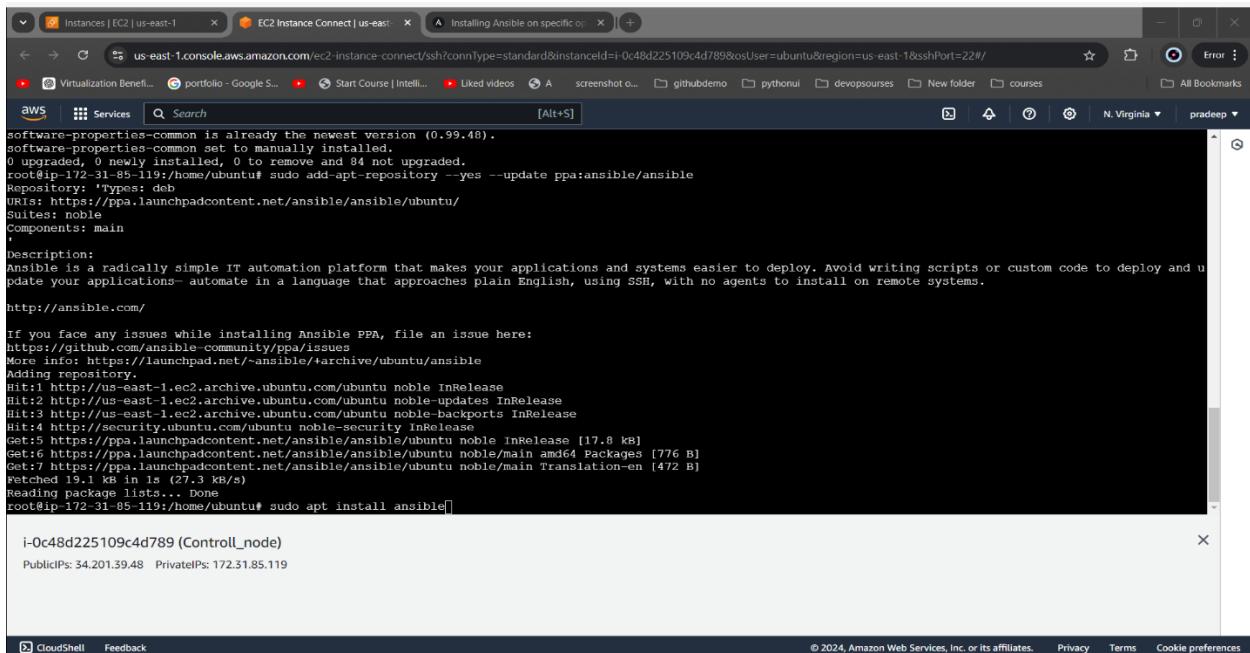
```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | A Installing Ansible on specific op... | +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu&region=us-east-1&sshPort=22/#  
Virtualization Benefits portfolio - Google S... Start Course | Intelli... Liked videos A screenshot o... githubdemo pythonui devopsources New folder courses All Bookmarks N. Virginia pradeep  
aws Services Search [Alt+S]  
Building dependency tree... Done  
Reading state information... Done  
84 packages can be upgraded. Run 'apt list --upgradable' to see them.  
root@ip-172-31-85-119:/home/ubuntu# apt update -y  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
84 packages can be upgraded. Run 'apt list --upgradable' to see them.  
root@ip-172-31-85-119:/home/ubuntu# apt update  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
84 packages can be upgraded. Run 'apt list --upgradable' to see them.  
root@ip-172-31-85-119:/home/ubuntu# sudo apt install software-properties-common  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
software-properties-common is already the newest version (0.99.48).  
software-properties-common set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 84 not upgraded.  
root@ip-172-31-85-119:/home/ubuntu# sudo add-apt-repository --yes --update ppa:ansible/ansible  
i-0c48d225109c4d789 (Controll_node)  
Public IPs: 34.201.39.48 Private IPs: 172.31.85.119  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```



```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | A Installing Ansible on specific op... | +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu&region=us-east-1&sshPort=22/#  
Virtualization Benefits portfolio - Google S... Start Course | Intelli... Liked videos A screenshot o... githubdemo pythonui devopsources New folder courses All Bookmarks N. Virginia pradeep  
aws Services Search [Alt+S]  
software-properties-common is already the newest version (0.99.48).  
software-properties-common set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 84 not upgraded.  
root@ip-172-31-85-119:/home/ubuntu# sudo add-apt-repository --yes --update ppa:ansible/ansible  
Repository: 'Types: deb  
URIs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/  
Suites: noble  
Components: main  
'  
Description:  
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications—automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.  
http://ansible.com/  
If you face any issues while installing Ansible PPA, file an issue here:  
https://github.com/ansible-community/ansible/issues  
More info: https://launchpad.net/~ansibles/+archive/ubuntu/ansible  
Adding repository.  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Get:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/noble InRelease [17.8 kB]  
Get:6 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/noble/main amd64 Packages [776 B]  
Get:7 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/noble/main Translation-en [472 B]  
Fetched 19.1 kB in 1s (27.3 kB/s)  
Reading package lists... Done  
root@ip-172-31-85-119:/home/ubuntu#  
i-0c48d225109c4d789 (Controll_node)  
Public IPs: 34.201.39.48 Private IPs: 172.31.85.119  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Execute this command to install Ansible.

→ sudo apt install ansible



```
root@ip-172-31-85-119:/home/ubuntu# sudo apt install ansible
[sudo] password for root: 
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  software-properties-common
  software-properties-common set to manually installed.
  0 upgraded, 0 newly installed, 0 to remove and 84 not upgraded.
root@ip-172-31-85-119:/home/ubuntu# sudo add-apt-repository --yes --update ppa:ansible/ansible
Repository: *types: deb
URLs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/
Suites: noble
Components: main
.
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

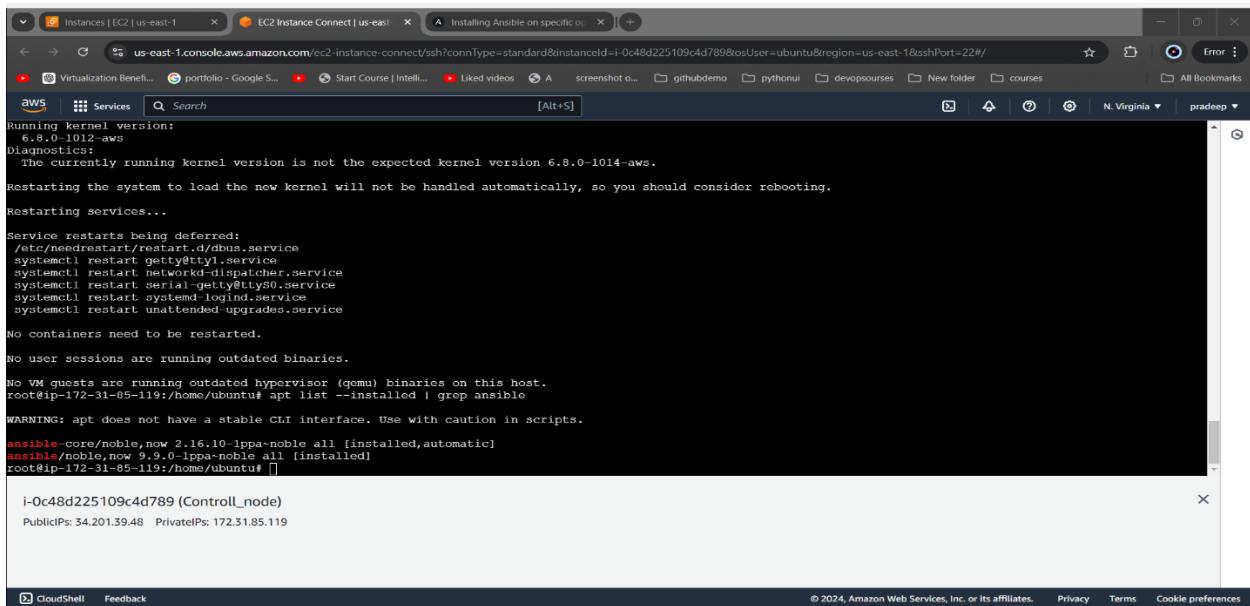
http://ansible.com/
If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~archive/ubuntu/ansible
Add repository:
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ubuntu/noble InRelease [17.8 kB]
Get:6 https://ppa.launchpadcontent.net/ansible/ubuntu/noble/main amd64 Packages [776 B]
Get:7 https://ppa.launchpadcontent.net/ansible/ubuntu/noble/main Translation-en [472 B]
Fetched 19.1 kB in 1s (27.3 kB/s)
Reading package lists...
Done
root@ip-172-31-85-119:/home/ubuntu# sudo apt install ansible

i-0c48d225109c4d789 (Controll_node)
PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119
```

Ansible installation was successful.

Verify the installation success by running the following command.

To check if Ansible is installed: -> execute apt list --installed | grep ansible.



```
root@ip-172-31-85-119:/home/ubuntu# apt list --installed | grep ansible
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

ansible-core/noble,now 2.16.10-1ppa-noble all [installed,automatic]
ansible/noble,now 9.9.0-1ppa-noble all [installed]
root@ip-172-31-85-119:/home/ubuntu#
```

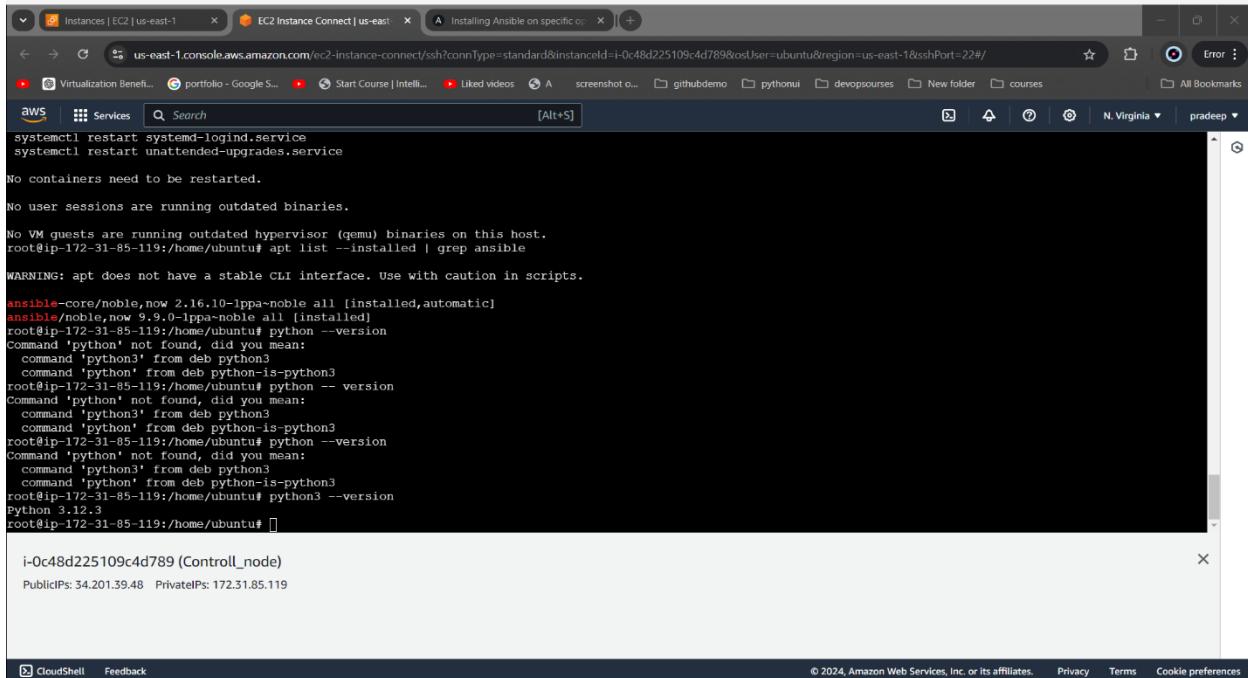
It is confirmed that Ansible is successfully installed.

Verify that Python and SSH are installed on all the nodes.

Controller Node: Ensure SSH client and Python are installed.

Python:

→ Python3 –version

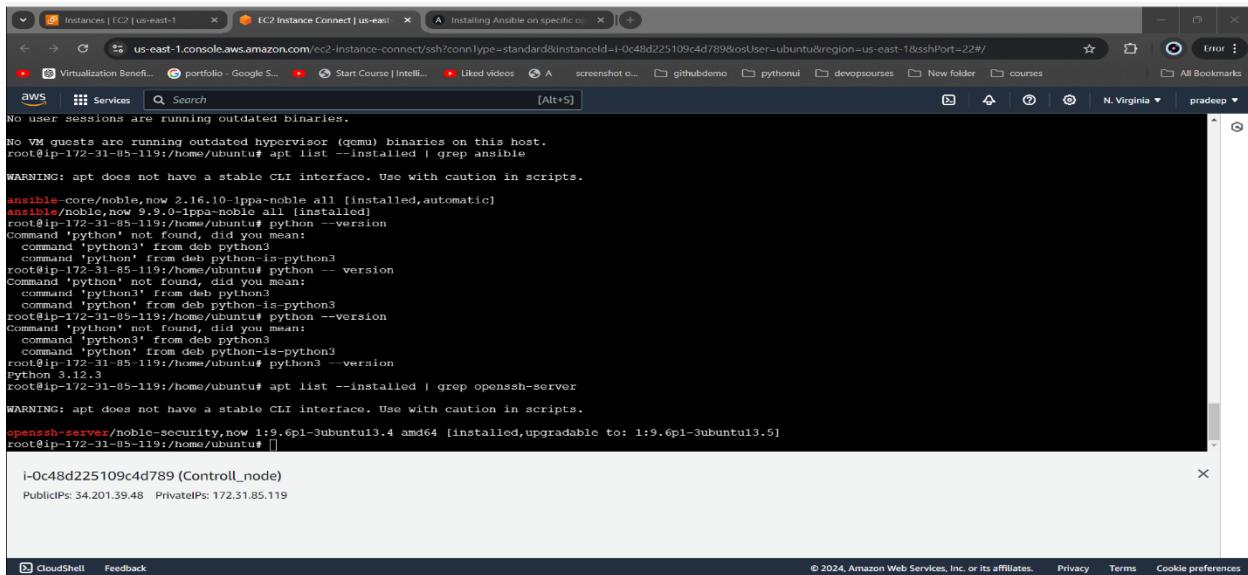


```
Instances | EC2 | us-east-1 EC2 Instance Connect | us-east-1 A Installing Ansible on specific o ... + 
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu&region=us-east-1&sshPort=22#/
Virtualization Benefits portfolio - Google Search Start Course | Intelli... Liked videos A screenshot o... githubdemo python devopsources New folder courses All Bookmarks N. Virginia pradeep
aws Services Search [Alt+S]
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-85-119:/home/ubuntu# apt list --installed | grep ansible
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

ansible-core/noble,now 2.16.10-1ppa-noble all [installed,automatic]
ansible/noble,now 9.9.0-1ppa-noble all [installed]
root@ip-172-31-85-119:/home/ubuntu# python --version
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
root@ip-172-31-85-119:/home/ubuntu# python --version
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
root@ip-172-31-85-119:/home/ubuntu# python3 --version
Python 3.12.3
root@ip-172-31-85-119:/home/ubuntu# i-0c48d225109c4d789 (Controll_node)
PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Ssh:

→ Apt list –installed | grep open ssh-server



```
Instances | EC2 | us-east-1 EC2 Instance Connect | us-east-1 A Installing Ansible on specific o ... + 
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu&region=us-east-1&sshPort=22#/
Virtualization Benefits portfolio - Google Search Start Course | Intelli... Liked videos A screenshot o... githubdemo python devopsources New folder courses All Bookmarks N. Virginia pradeep
aws Services Search [Alt+S]
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-85-119:/home/ubuntu# apt list --installed | grep ansible
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

ansible-core/noble,now 2.16.10-1ppa-noble all [installed,automatic]
ansible/noble,now 9.9.0-1ppa-noble all [installed]
root@ip-172-31-85-119:/home/ubuntu# python --version
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
root@ip-172-31-85-119:/home/ubuntu# python --version
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
root@ip-172-31-85-119:/home/ubuntu# python --version
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
root@ip-172-31-85-119:/home/ubuntu# python3 --version
Python 3.12.3
root@ip-172-31-85-119:/home/ubuntu# apt list --installed | grep openssh-server
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

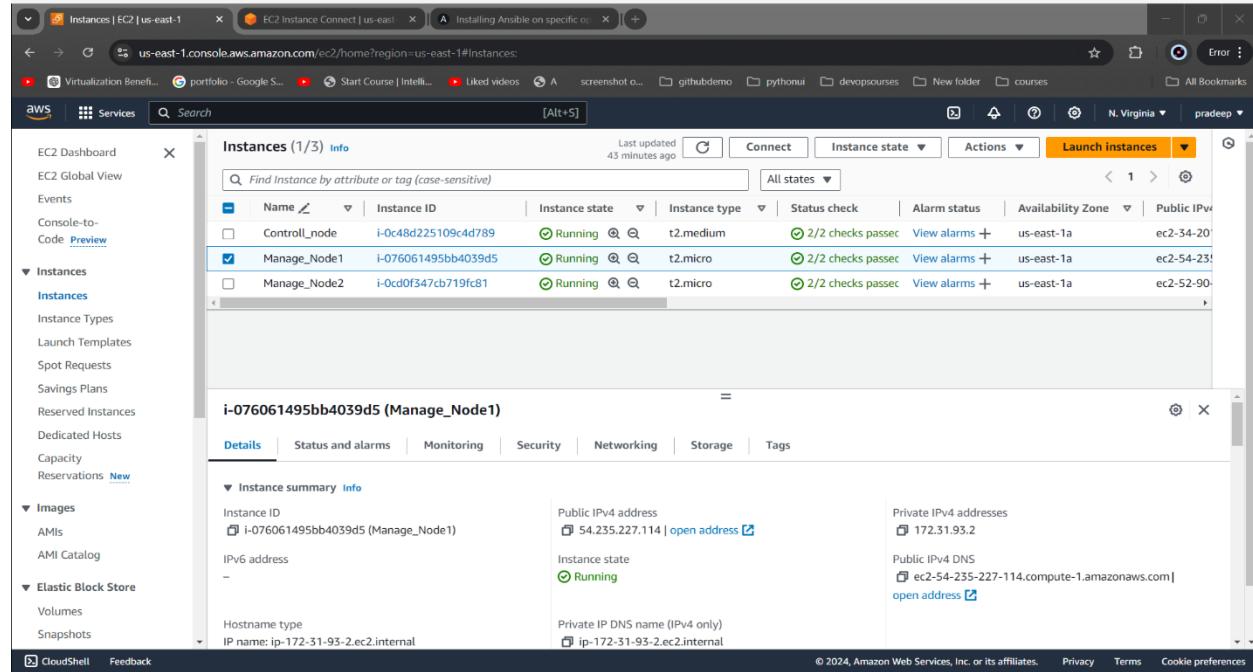
openssh-server/noble-security,now 1:9.6p1-3ubuntu13.4 amd64 [installed,upgradable to: 1:9.6p1-3ubuntu13.5]
root@ip-172-31-85-119:/home/ubuntu# i-0c48d225109c4d789 (Controll_node)
PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Verification shows that both Python and SSH are installed on the controller node.

Managed Nodes: Ensure SSH server and Python are installed.

Establish connections to both 'Manage Node 1' and 'Manage Node 2'.

Manage Node 1:

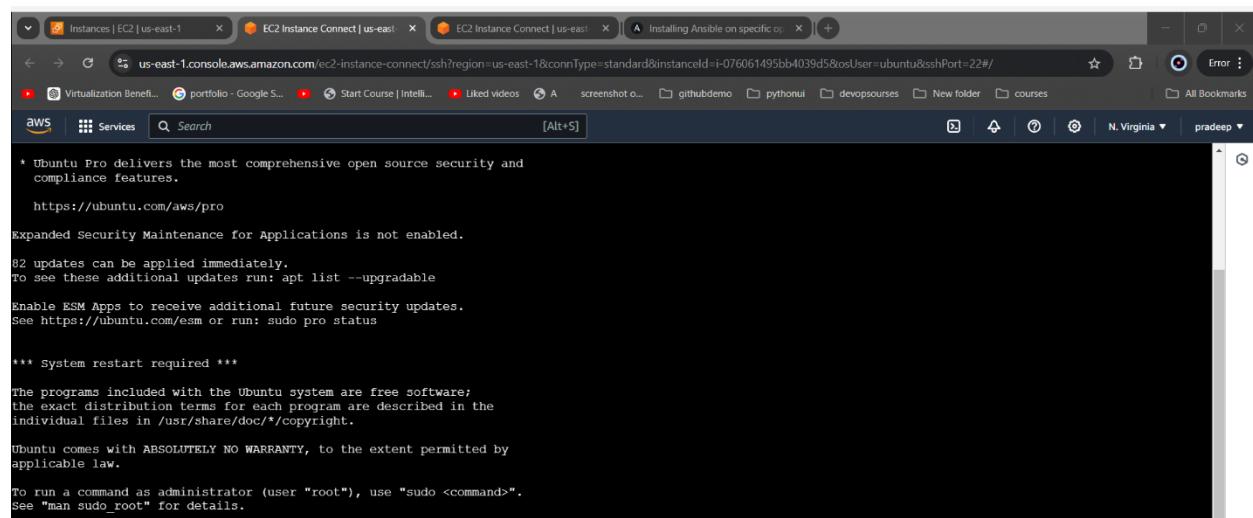


The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 address
Controll_node	i-0c48d225109c4d789	Running	t2.medium	2/2 checks passed	View alarms	us-east-1a	ec2-34-20-
Manage_Node1	i-076061495bb4039d5	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-54-23-
Manage_Node2	i-0cd0f347cb719fc81	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-52-90-

Details for Manage_Node1:

- Instance ID: i-076061495bb4039d5 (Manage_Node1)
- Public IPv4 address: 54.235.227.114 | [open address](#)
- Instance state: Running
- Private IP DNS name (IPv4 only): ip-172-31-93-2.ec2.internal
- Private IPv4 addresses: 172.31.93.2
- Public IPv4 DNS: ec2-54-23-227-114.compute-1.amazonaws.com | [open address](#)



The CloudShell terminal shows the following output:

```
* Ubuntu Pro delivers the most comprehensive open source security and compliance features.
https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

82 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo root" for details.

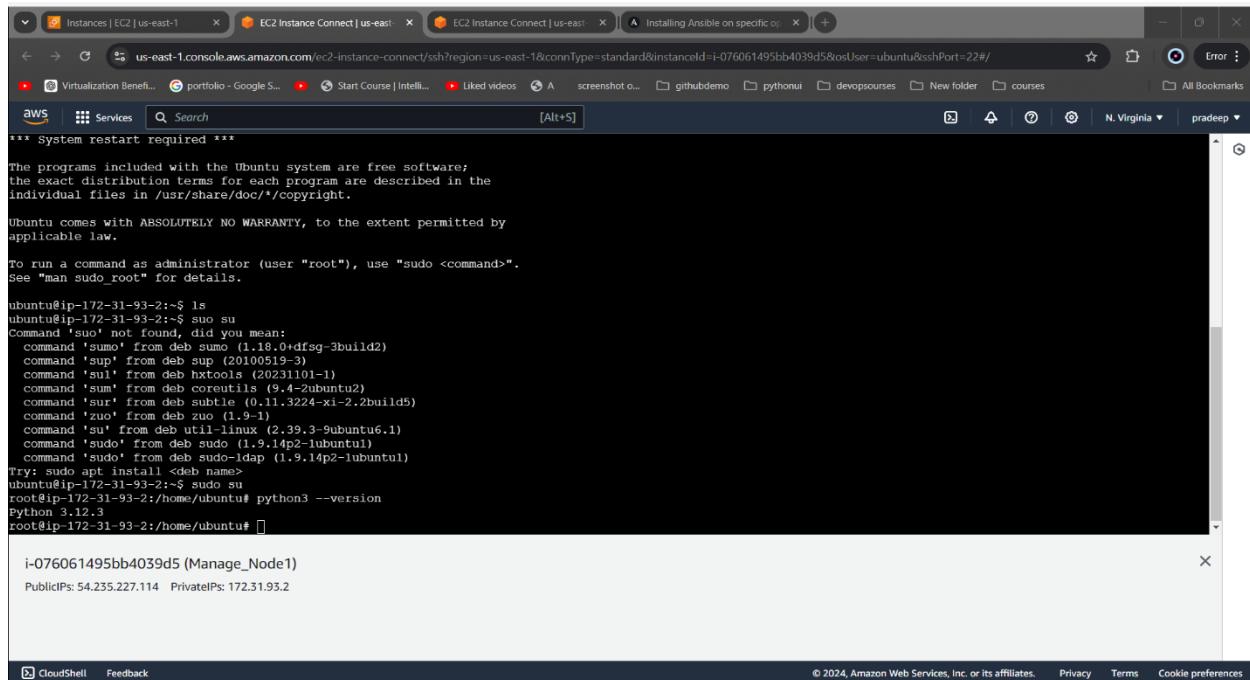
ubuntu@ip-172-31-93-2:~$ ls
ubuntu@ip-172-31-93-2:~$ 
```

CloudShell status: i-076061495bb4039d5 (Manage_Node1)
Public IPs: 54.235.227.114 Private IPs: 172.31.93.2

Connation successfully.

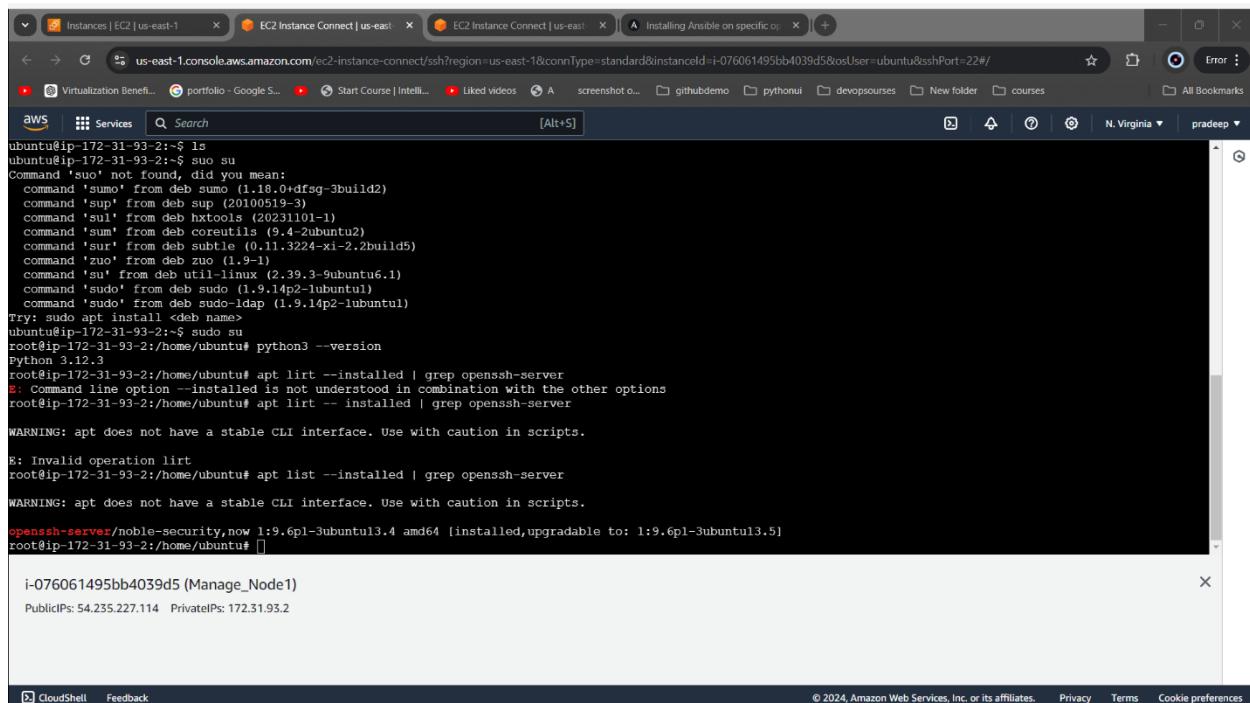
Use these commands to check if SSH and Python are installed.

→ Python3 –version



```
Instances | EC2 | us-east-1 EC2 Instance Connect | us-east-1 EC2 Instance Connect | us-east-1 A Installing Ansible on specific o ... Error
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-076061495bb4039d5&osUser=ubuntu&sshPort=22#
Virtualization Benefit... portfolio - Google S... Start Course | Intelli... Liked videos screenshot o... githubdemo python1 devopsources New folder courses All Bookmarks AWS Services Search [Alt+S]
*** System restart required ***
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-93-2:~$ ls
ubuntu@ip-172-31-93-2:~$ sudo su
Command 'su' not found, did you mean:
  command 'sumo' from deb sumo (1.18.0+dfsg-3build2)
  command 'sup' from deb sup (20100519-3)
  command 'sul' from deb htools (20231101-1)
  command 'sum' from deb coreutils (9.4-2ubuntu2)
  command 'sur' from deb subtle (0.11.3224-xi-2.2build5)
  command 'zuo' from deb zuo (1.9-1)
  command 'su' from deb util-linux (2.39.3-9ubuntu6.1)
  command 'sudo' from deb sudo (1.9.14p2-lubuntu1)
  command 'sudo' from deb sudo-ldap (1.9.14p2-lubuntu1)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-93-2:~$ sudo su
root@ip-172-31-93-2:/home/ubuntu# python3 --version
Python 3.12.3
root@ip-172-31-93-2:/home/ubuntu# []
i-076061495bb4039d5 (Manage_Node1)
PublicIP: 54.235.227.114 PrivateIP: 172.31.93.2
```

→ Apt list –installed | grep open ssh-server



```
Instances | EC2 | us-east-1 EC2 Instance Connect | us-east-1 EC2 Instance Connect | us-east-1 A Installing Ansible on specific o ... Error
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-076061495bb4039d5&osUser=ubuntu&sshPort=22#
Virtualization Benefit... portfolio - Google S... Start Course | Intelli... Liked videos screenshot o... githubdemo python1 devopsources New folder courses All Bookmarks AWS Services Search [Alt+S]
ubuntu@ip-172-31-93-2:~$ ls
ubuntu@ip-172-31-93-2:~$ sudo su
Command 'su' not found, did you mean:
  command 'sumo' from deb sumo (1.18.0+dfsg-3build2)
  command 'sup' from deb sup (20100519-3)
  command 'sul' from deb htools (20231101-1)
  command 'sum' from deb coreutils (9.4-2ubuntu2)
  command 'sur' from deb subtle (0.11.3224-xi-2.2build5)
  command 'zuo' from deb zuo (1.9-1)
  command 'su' from deb util-linux (2.39.3-9ubuntu6.1)
  command 'sudo' from deb sudo (1.9.14p2-lubuntu1)
  command 'sudo' from deb sudo-ldap (1.9.14p2-lubuntu1)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-93-2:~$ sudo su
root@ip-172-31-93-2:/home/ubuntu# python3 --version
Python 3.12.3
root@ip-172-31-93-2:/home/ubuntu# apt llist --installed | grep openssh-server
E: Command line option --installed is not understood in combination with the other options
root@ip-172-31-93-2:/home/ubuntu# apt llist -- installed | grep openssh-server
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
E: Invalid operation llist
root@ip-172-31-93-2:/home/ubuntu# apt list --installed | grep openssh-server
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
openssh-server/noble-security,now 1:9.6p1-3ubuntul3.4 amd64 [installed,upgradable to: 1:9.6p1-3ubuntul3.5]
root@ip-172-31-93-2:/home/ubuntu# []
i-076061495bb4039d5 (Manage_Node1)
PublicIP: 54.235.227.114 PrivateIP: 172.31.93.2
```

Manage Node 2:

Use the same command to verify the installation on Manage Node 2.

Establish a connection to Manage Node 2:

The screenshot shows the AWS CloudShell interface. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Instances, Images, and Elastic Block Store. The main area displays the 'Instances (1/3) Info' table with one row for 'Manage_Node2'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. Below the table, a detailed view for 'i-Ocd0f347cb719fc81 (Manage_Node2)' is shown, including sections for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is selected, showing fields like Instance ID, Public IPv4 address (52.90.114.136), Instance state (Running), and Private IP DNS name (ip-172-31-93-253.ec2.internal).

The screenshot shows the AWS CloudShell terminal window. The terminal output is as follows:

```
* Ubuntu Pro delivers the most comprehensive open source security and compliance features.
https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

109 updates can be applied immediately.
30 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-93-253:~$ sudo su
root@ip-172-31-93-253:/home/ubuntu# [REDACTED]

i-Ocd0f347cb719fc81 (Manage_Node2)
PublicIPs: 52.90.114.136 PrivateIPs: 172.31.93.253
```

Successfully connected.

➔ Python3 –version:

```
109 updates can be applied immediately.  
30 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/<copyright>.br/>  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-172-31-93-253:~$ sudo su  
root@ip-172-31-93-253:/home/ubuntu# python3 --version  
Python 3.12.3  
root@ip-172-31-93-253:/home/ubuntu#  
  
i-0cd0f347cb719fc81 (Manage_Node2)  
Public IPs: 52.90.114.136 Private IPs: 172.31.93.253
```

→ Apt list –installed | grep open ssh-server

```
command 'pt' from deb tcllib (1.21+dfsg-1)  
command 'sptk' from deb sptk (3.9-3)  
command 'opt' from deb llvm (1:17.0-58-exp1)  
command 'srt' from deb python3-psr1 (1.0.1-3)  
See 'snap info <snapname>' for additional versions.  
root@ip-172-31-93-253:/home/ubuntu# spt list --installed | grep openssh-server  
Command 'spt' not found, did you mean:  
  command 'spot' from snap spot (0.4.1)  
  command 'smt' from snap smt (0.2)  
  command 'sqit' from deb python3-sqit (0.8.0-8)  
  command 'gpt' from deb gpt (1.1-7)  
  command 'sptk' from deb sptk (3.9-3)  
  command 'apt' from deb apt (2.7.6)  
  command 'srt' from deb python3-psr1 (1.0.1-3)  
  command 'pt' from deb tcllib (1.21+dfsg-1)  
  command 'spc' from deb supercat (0.5.7-1)  
  command 'sit' from deb sit (0.0.git20140301-6ubuntu0.24.04.1)  
  command 'opt' from deb llvm (1:17.0-58-exp1)  
  command 'spd' from deb spd (1.3.0-1.1)  
  command 'st' from deb stterm (0.9-1)  
  command 'dpt' from deb pkg-perl-tools (0.76)  
  command 'ppt' from deb bsdgames (2.17-30)  
See 'snap info <snapname>' for additional versions.  
root@ip-172-31-93-253:/home/ubuntu# apt list --installed | grep openssh-server  
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.  
openssh-server/noble-security,now 1:9.6+1-3ubuntu13.4 amd64 [installed,upgradable to: 1:9.6+1-3ubuntu13.5]  
root@ip-172-31-93-253:/home/ubuntu#  
  
i-0cd0f347cb719fc81 (Manage_Node2)  
Public IPs: 52.90.114.136 Private IPs: 172.31.93.253
```

All nodes have Python and SSH installed.

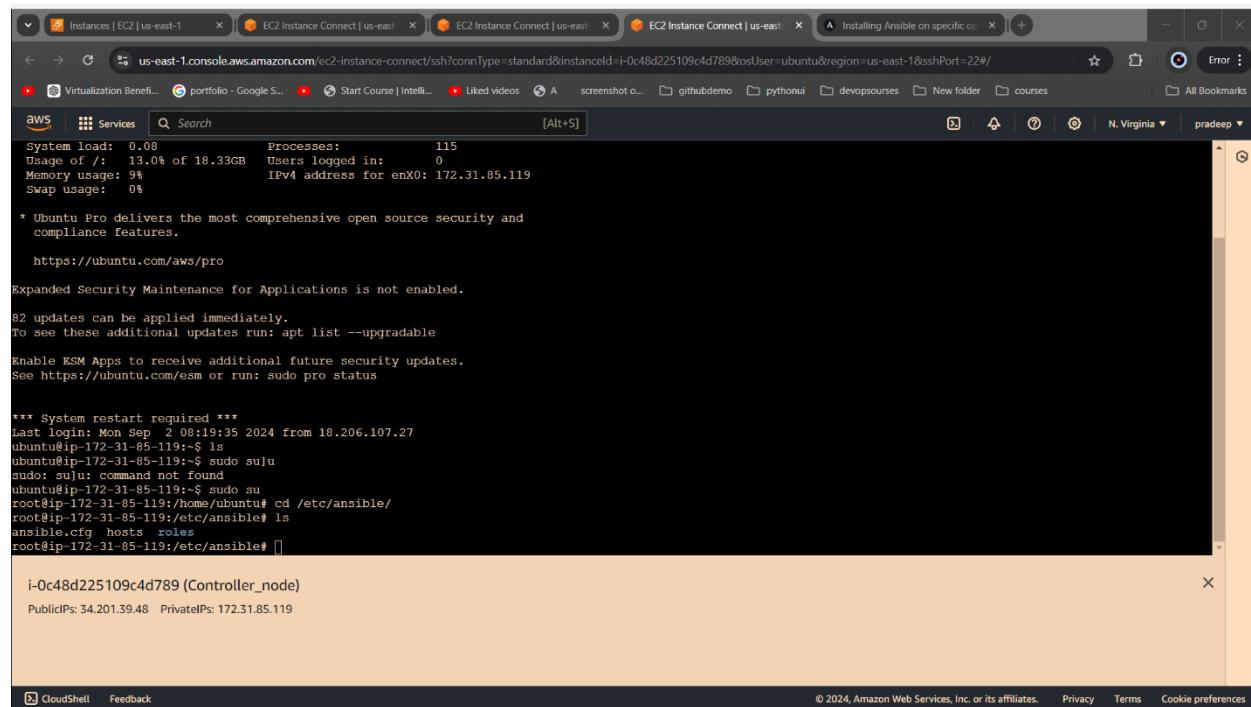
When connecting the manage node to the controller node via SSH, the server IP, username, password, or private key must be configured.

In AWS, we use passwordless SSH, which requires using the private key.

Ansible supports both password-based SSH and passwordless SSH.

We'll create an inventory file to specify the IP addresses.

Upon installing Ansible, a directory /etc/ansible is created, often referred to as Ansible's root directory.



The screenshot shows a CloudShell terminal window with the following output:

```
System load: 0.08 Processes: 115
Usage of /: 13.0% of 18.33GB Users logged in: 0
Memory usage: 9% IPv4 address for enX0: 172.31.85.119
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

82 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Mon Sep 2 08:19:35 2024 from 18.206.107.27
ubuntu@ip-172-31-85-119:~$ ls
ubuntu@ip-172-31-85-119:~$ sudo su
sudo: su: command not found
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# cd /etc/ansible/
root@ip-172-31-85-119:/etc/ansible# ls
ansible.cfg hosts roles
root@ip-172-31-85-119:/etc/ansible#
```

Below the terminal, the CloudShell interface is visible with the following details:

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119

In the Ansible root directory, there are three files, including a file named hosts. This is the inventory file where we need to configure the IP addresses of the manage nodes.



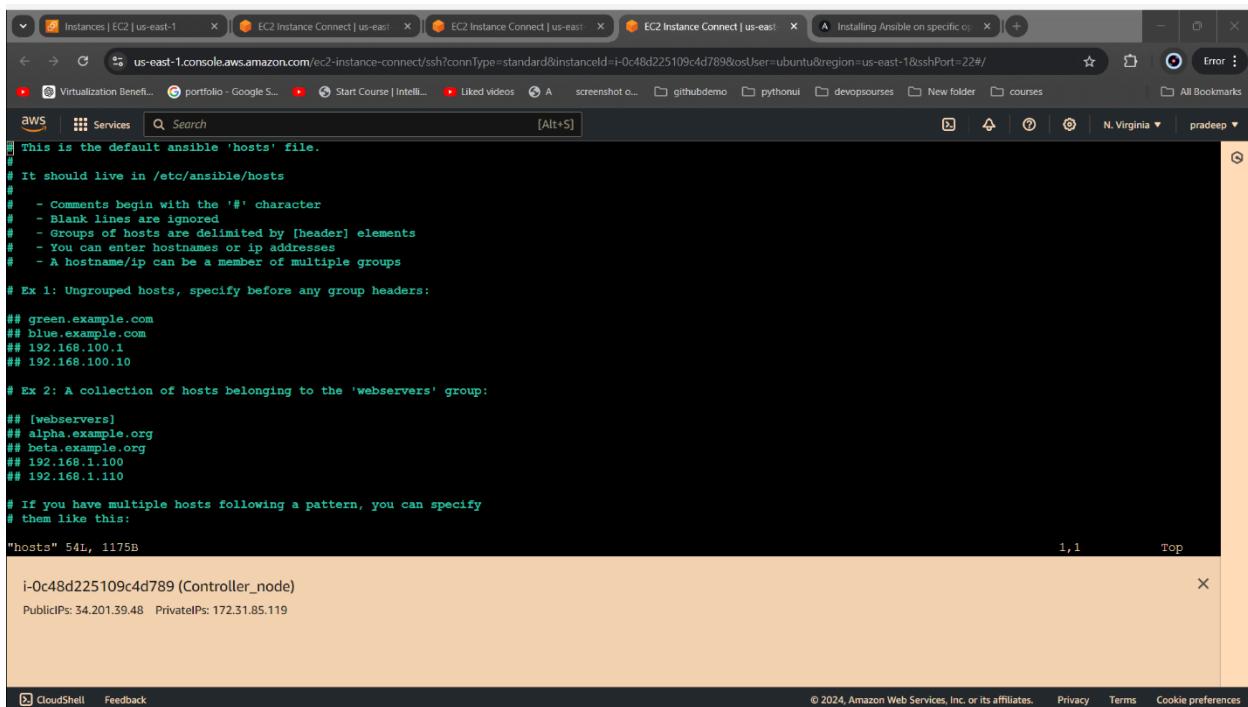
The screenshot shows a CloudShell terminal window with the following output:

```
*** System restart required ***
Last login: Mon Sep 2 08:19:35 2024 from 18.206.107.27
ubuntu@ip-172-31-85-119:~$ ls
ubuntu@ip-172-31-85-119:~$ sudo su
sudo: su: command not found
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# cd /etc/ansible/
root@ip-172-31-85-119:/etc/ansible# ls
ansible.cfg hosts roles
root@ip-172-31-85-119:/etc/ansible#
```

Below the terminal, the CloudShell interface is visible with the following details:

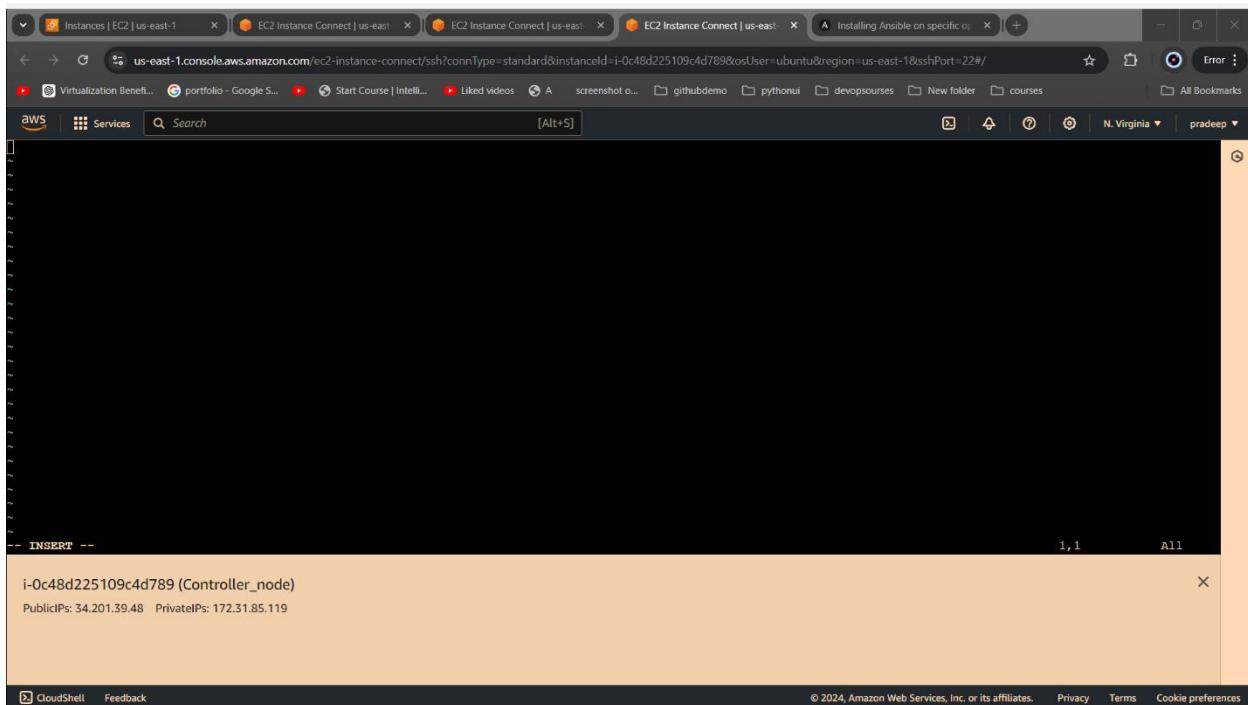
i-0c48d225109c4d789 (Controller_node)
PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119

Use Vim to open the hosts file for editing.



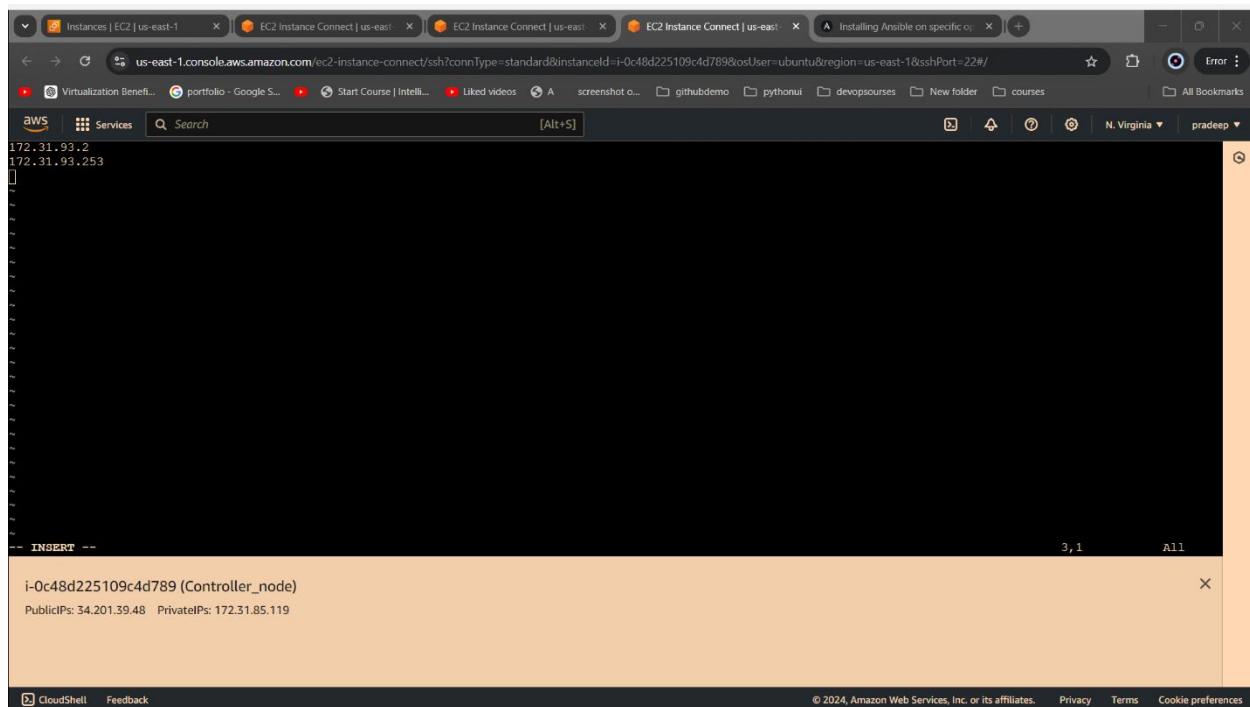
```
This is the default ansible 'hosts' file.  
# It should live in /etc/ansible/hosts  
  
# Comments begin with the '#' character  
# Blank lines are ignored  
# Groups of hosts are delimited by [header] elements  
# You can enter hostnames or ip addresses  
# A hostname/ip can be a member of multiple groups  
  
# Ex 1: Ungrouped hosts, specify before any group headers:  
  
## green.example.com  
## blue.example.com  
## 192.168.100.1  
## 192.168.100.10  
  
# Ex 2: A collection of hosts belonging to the 'webservers' group:  
  
## [webservers]  
## alpha.example.org  
## beta.example.org  
## 192.168.1.100  
## 192.168.1.110  
  
# If you have multiple hosts following a pattern, you can specify  
# them like this:  
  
"hosts" 54L, 1175B  
  
i-0c48d225109c4d789 (Controller_node)  
PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119
```

Remove all existing content.



```
-- INSERT --  
  
i-0c48d225109c4d789 (Controller_node)  
PublicIPs: 34.201.39.48 PrivateIPs: 172.31.85.119
```

Next, enter the IP addresses of the manage nodes in this file.



The screenshot shows a browser window with multiple tabs open. The active tab is titled "Instances | EC2 | us-east-1". The URL is "us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu®ion=us-east-1&sshPort=22#/".

The content of the terminal window shows the following:

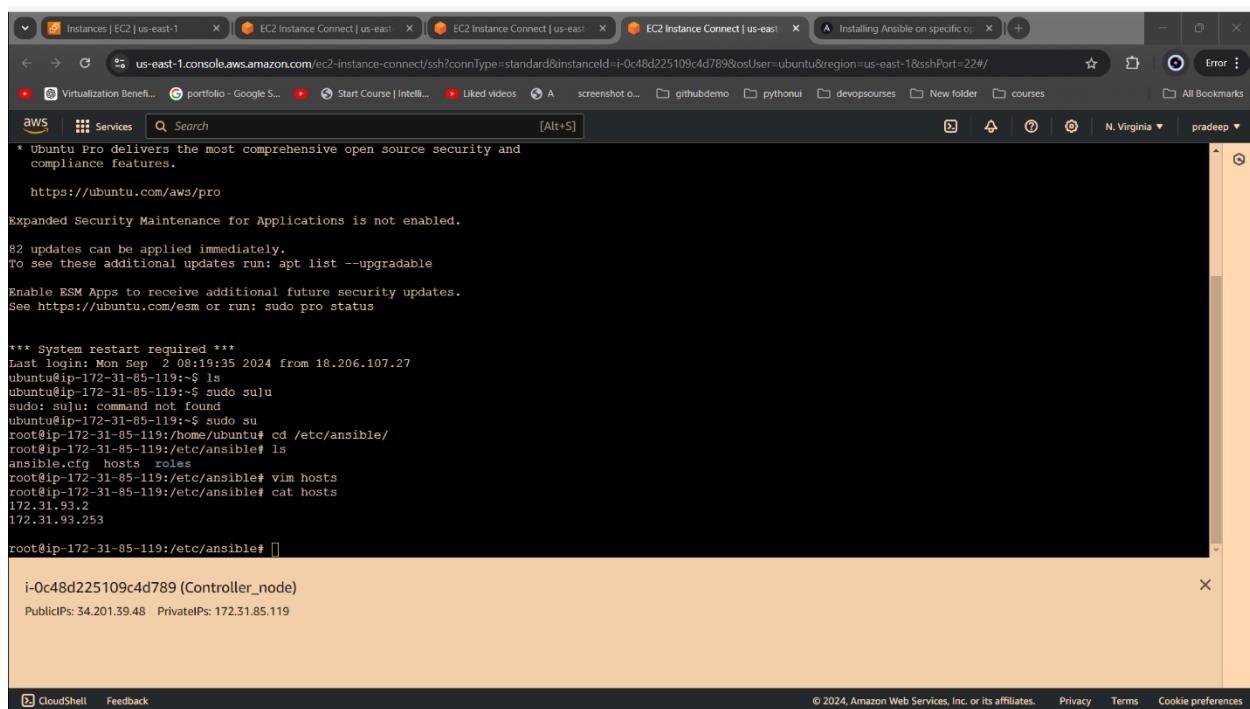
```
172.31.93.2
172.31.93.253
```

Below this, there is a modal dialog box with the title "i-0c48d225109c4d789 (Controller_node)". It displays the following information:

i-0c48d225109c4d789 (Controller_node)
Public IPs: 34.201.39.48 Private IPs: 172.31.85.119

At the bottom of the terminal window, there are links for "CloudShell" and "Feedback".

Save your changes to the file.



The screenshot shows a browser window with multiple tabs open. The active tab is titled "Instances | EC2 | us-east-1". The URL is "us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu®ion=us-east-1&sshPort=22#/".

The content of the terminal window shows the following:

```
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

82 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Mon Sep 2 08:19:35 2024 from 18.206.107.27
ubuntu@ip-172-31-85-119:~$ ls
ubuntu@ip-172-31-85-119:~$ sudo su
sudo: su: command not found
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# cd /etc/ansible/
root@ip-172-31-85-119:/etc/ansible# ls
ansible.cfg hosts roles
root@ip-172-31-85-119:/etc/ansible# vim hosts
root@ip-172-31-85-119:/etc/ansible# cat hosts
172.31.93.2
172.31.93.253

root@ip-172-31-85-119:/etc/ansible#
```

Below this, there is a modal dialog box with the title "i-0c48d225109c4d789 (Controller_node)". It displays the following information:

i-0c48d225109c4d789 (Controller_node)
Public IPs: 34.201.39.48 Private IPs: 172.31.85.119

At the bottom of the terminal window, there are links for "CloudShell" and "Feedback".

To view the nodes in the inventory file, run the command.

→ ansible all --list-hosts.

```
root@ip-172-31-85-119:/etc/ansible# ansible all --list-hosts
hosts (2):
  172.31.93.2
  172.31.93.253
root@ip-172-31-85-119:/etc/ansible# 

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 18.204.194.62 PrivateIPs: 172.31.85.119

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Create a host group in the inventory file and assign the IP addresses to it.

```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | Start Course | Intellipaat |
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu&region=us-east-1&sshPort=22#/ 
Virtualization Benefits portfolio - Google Slides Start Course | Intellipaat Liked videos A screenshot GitHub demo Python devopsourses New folder courses All Bookmarks N. Virginia pradeep

AWS Services Search [Alt+S] 
[slave Host]
172.31.93.2
172.31.93.253

-- INSERT --
i-0c48d225109c4d789 (Controller_node)
PublicIPs: 18.204.194.62 PrivateIPs: 172.31.85.119

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

The IP addresses are now grouped under a single host group named slave_host.

To view the IP addresses in this host group, run the following command.

→ ansible slave-host --list-hosts

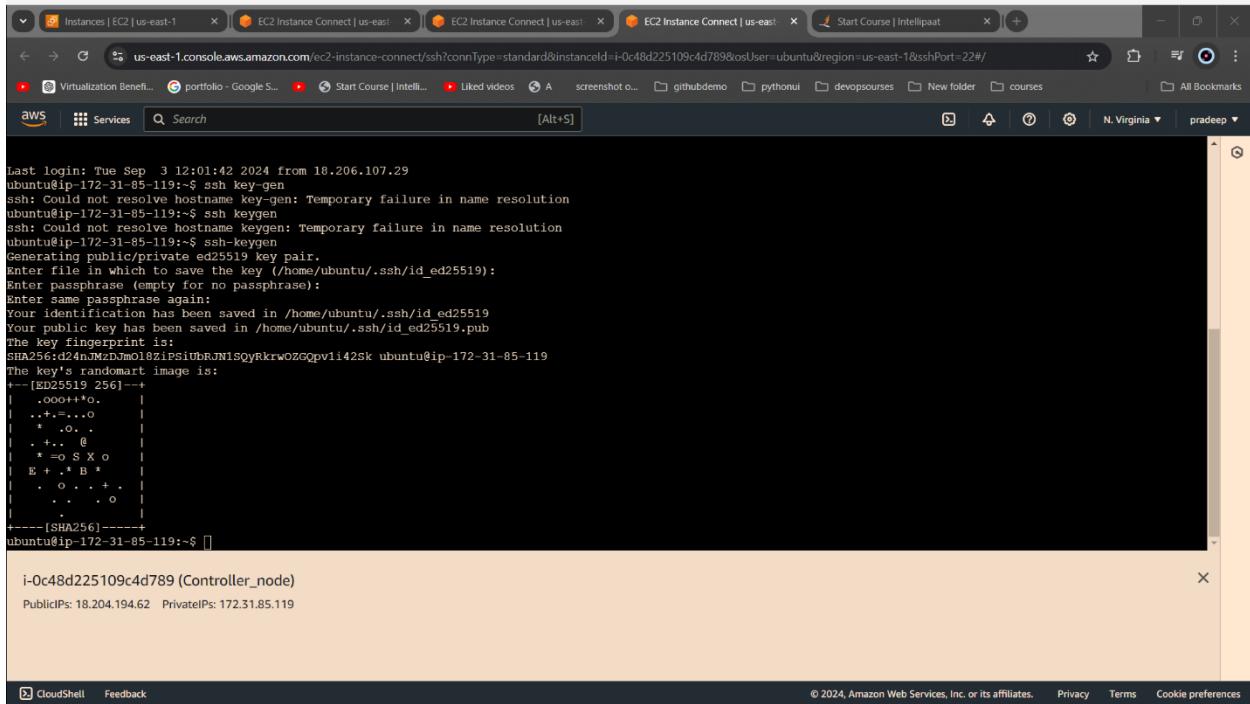
```
hosts (0):
root@ip-172-31-85-119:/etc/ansible# ansible slave_host --list-hosts
hosts (2):
  172.31.93.2
  172.31.93.253
root@ip-172-31-85-119:/etc/ansible# 
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 18.204.194.62 PrivateIPs: 172.31.85.119

After providing the IP address for the SSH connection, we need to provide either a password or a private key. In AWS, we use passwordless SSH, so we'll use a private key. Run the ssh-keygen command to generate the SSH key.

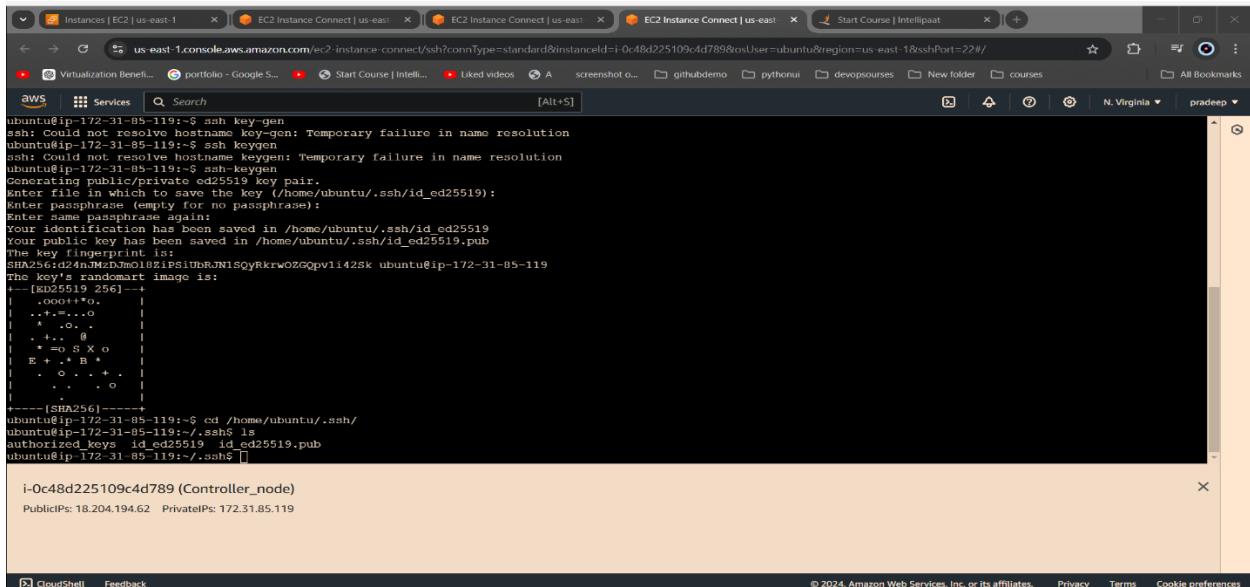
→ Ssh keygen



```
Last login: Tue Sep 3 12:01:42 2024 from 18.206.107.29
ubuntu@ip-172-31-85-119:~$ ssh-keygen
ssh: Could not resolve hostname key-gen: Temporary failure in name resolution
ubuntu@ip-172-31-85-119:~$ ssh-keygen
ssh: Could not resolve hostname keygen: Temporary failure in name resolution
ubuntu@ip-172-31-85-119:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519.
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:d24nMzDm0l8ZlP5iUbRJN1S0yRkrwoZGQpvli42sk ubuntu@ip-172-31-85-119
The key's randomart image is:
[ED25519 256]-
[SHA256]-
ubuntu@ip-172-31-85-119:~$ [REDACTED]

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 18.204.194.62 PrivateIPs: 172.31.85.119
```

Inside /home/ubuntu/.ssh, we can find the key.

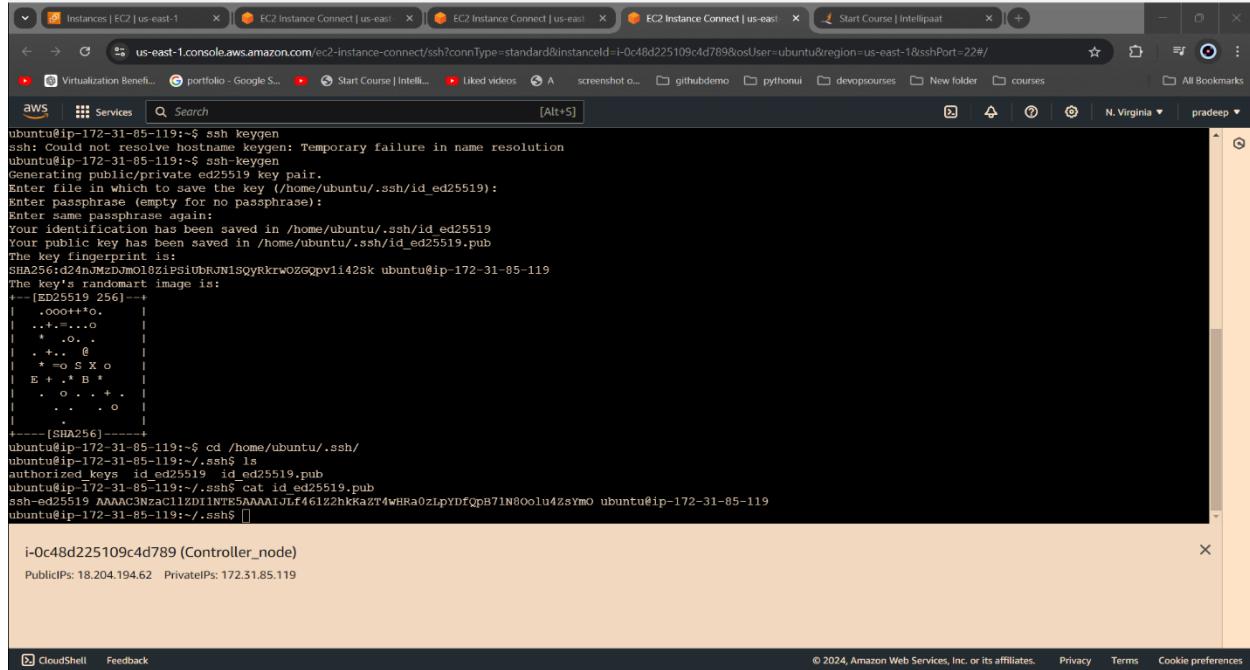


```
ubuntu@ip-172-31-85-119:~$ ssh-keygen
ssh: Could not resolve hostname key-gen: Temporary failure in name resolution
ubuntu@ip-172-31-85-119:~$ ssh-keygen
ssh: Could not resolve hostname keygen: Temporary failure in name resolution
ubuntu@ip-172-31-85-119:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519.
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:d24nMzDm0l8ZlP5iUbRJN1S0yRkrwoZGQpvli42sk ubuntu@ip-172-31-85-119
The key's randomart image is:
[ED25519 256]-
[SHA256]-
ubuntu@ip-172-31-85-119:~$ cd /home/ubuntu/.ssh/
ubuntu@ip-172-31-85-119:~/ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
ubuntu@ip-172-31-85-119:~/ssh$ [REDACTED]

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 18.204.194.62 PrivateIPs: 172.31.85.119
```

Private and public keys have both been generated.

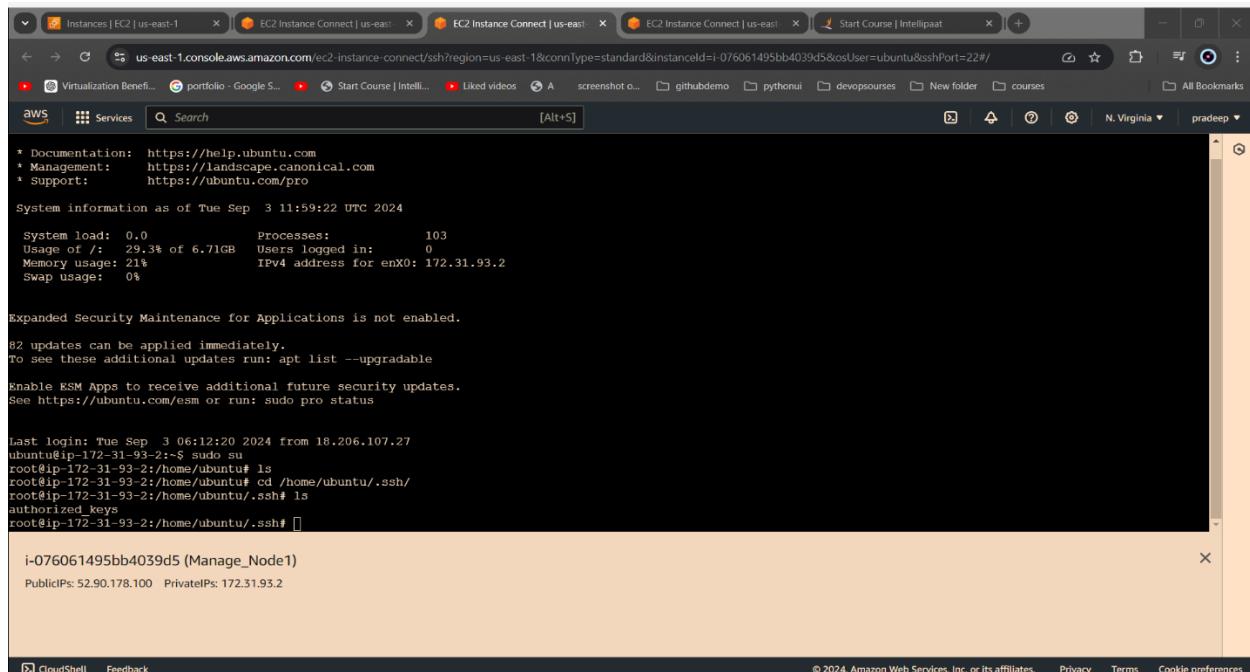
Copy the public key from the control node and add it to /home/ubuntu/.ssh/authorized_keys on the manage nodes.



```
ubuntu@ip-172-31-85-119:~$ ssh-keygen
ssh: Could not resolve hostname keygen: Temporary failure in name resolution
ubuntu@ip-172-31-85-119:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:d4nMxDmo1BzIPsi0URRN15QyRkwOZGqpVii425k ubuntu@ip-172-31-85-119
The key's randomart image is:
----[ED25519 256]----
.0000++*o.
+.+...o.
.+.o.
.+.+
* =o S X o
E + .* B *
. o . + .
. . . o
----[SHA256]-----
ubuntu@ip-172-31-85-119:~$ cd /home/ubuntu/.ssh
ubuntu@ip-172-31-85-119:~/ssh$ ls
authorized_keys id_ed25519 id_ed25519.pub
ubuntu@ip-172-31-85-119:~/ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIEJL461Z2hkKaZT4wHRAoZlpYDfOpB71N8Oolu4zsYMO ubuntu@ip-172-31-85-119
ubuntu@ip-172-31-85-119:~/ssh$
```

Copy the public key to the manage nodes.

Manage node1:



```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Tue Sep 3 11:59:22 UTC 2024
system load: 0.0      Processes:          103
Usage of /: 29.3% of 6.71GB   Users logged in: 0
Memory usage: 21%      IPv4 address for enX0: 172.31.93.2
Swap usage: 0%

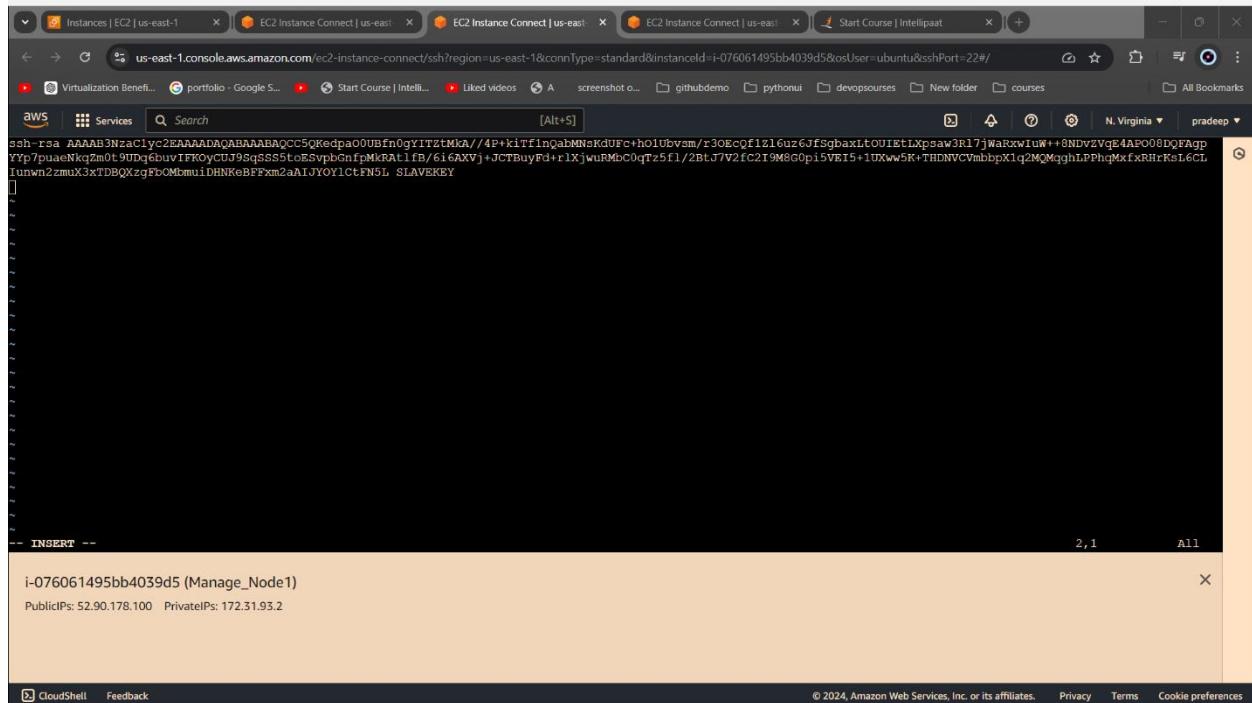
Expanded Security Maintenance for Applications is not enabled.
82 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Sep 3 06:12:20 2024 from 18.206.107.27
ubuntu@ip-172-31-93-2:~$ sudo su
root@ip-172-31-93-2:~# /home/ubuntu# ls
root@ip-172-31-93-2:~# /home/ubuntu# cd /home/ubuntu/.ssh/
root@ip-172-31-93-2:~# /home/ubuntu/.ssh# ls
authorized_keys
root@ip-172-31-93-2:~# /home/ubuntu/.ssh#
```

```
i-076061495bb4039d5 (Manage_Node1)
PublicIPs: 52.90.178.100 PrivateIPs: 172.31.93.2
```

Use Vim to open the file for editing.



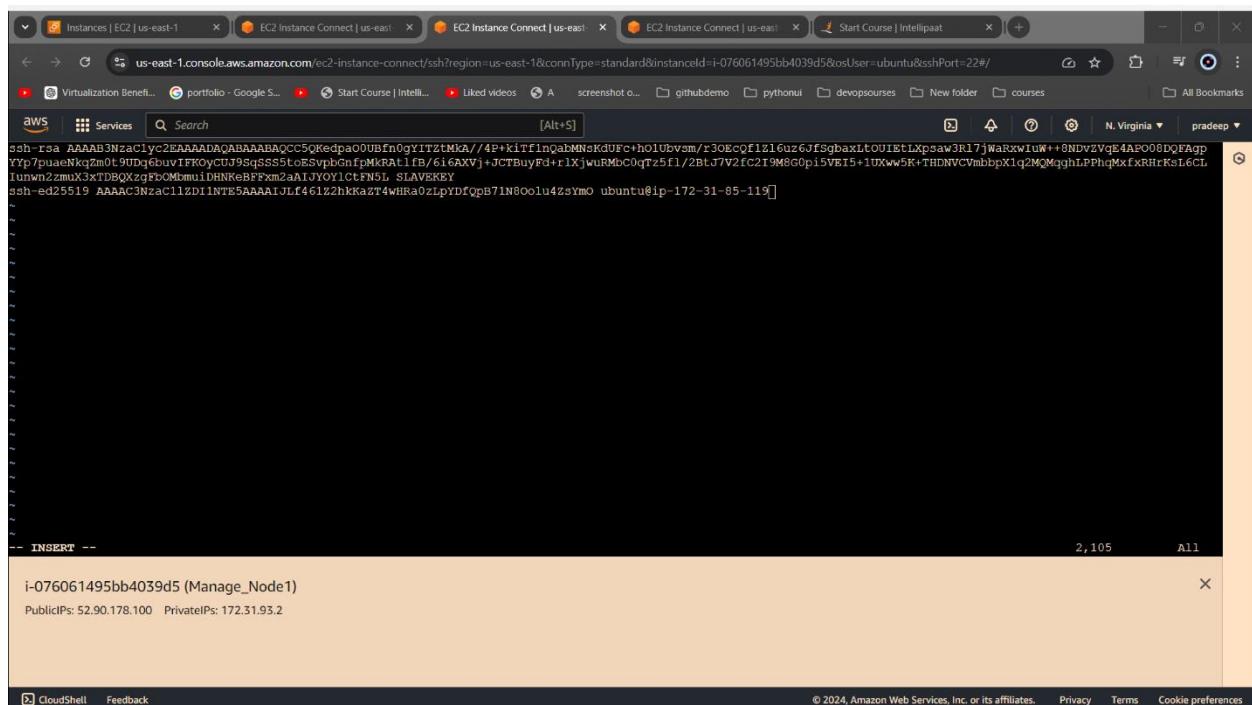
```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCC5QKedpa00UBfn0gYITztMKA//4P+kiTf1nQabMNkDUFc+h0lUbvsm/r30EcQf1z16uz6JfSgbaxLt0U1EtLxpsaw3R17jWaRxwIuW++0NDvZVqE4APo00DQFagpYYp7puaeNkjZm0t9UDq6buVfKoyCJ95qSS5t0ESvpbGnfpMKRat1fb/6i6AXVj+JCTBuyFd+r1XjwuMbc0qtz5f1/2BtJ7V2fc2I9M8G0p15VE15+1UXww5K+THDNVCVmmbbpX1q2MQMqghLPPhqMxfxRHrRsL6CLiunwn2zmlX3xTDBQXzgFBcMmuuiDHNkeBFFxm2aAIJYOF1CtEN5L SLAVEKEY
```

-- INSERT --

i-076061495bb4039d5 (Manage_Node1)

PublicIPs: 52.90.178.100 PrivateIPs: 172.31.93.2

Copy and paste the public key from the controller node into the authorized_keys file on manage node 1.



```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCC5QKedpa00UBfn0gYITztMKA//4P+kiTf1nQabMNkDUFc+h0lUbvsm/r30EcQf1z16uz6JfSgbaxLt0U1EtLxpsaw3R17jWaRxwIuW++0NDvZVqE4APo00DQFagpYYp7puaeNkjZm0t9UDq6buVfKoyCJ95qSS5t0ESvpbGnfpMKRat1fb/6i6AXVj+JCTBuyFd+r1XjwuMbc0qtz5f1/2BtJ7V2fc2I9M8G0p15VE15+1UXww5K+THDNVCVmmbbpX1q2MQMqghLPPhqMxfxRHrRsL6CLiunwn2zmlX3xTDBQXzgFBcMmuuiDHNkeBFFxm2aAIJYOF1CtEN5L SLAVEKEY
```

-- INSERT --

i-076061495bb4039d5 (Manage_Node1)

PublicIPs: 52.90.178.100 PrivateIPs: 172.31.93.2

```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | Start Course | Intellipaat | +
```

Memory usage: 21% IPv4 address for enx0: 172.31.93.2
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
82 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

```
Last login: Tue Sep 3 06:12:20 2024 from 18.206.107.27  
ubuntu@ip-172-31-93-2:~$ sudo su  
root@ip-172-31-93-2:/home/ubuntu# ls  
root@ip-172-31-93-2:/home/ubuntu# cd /home/ubuntu/.ssh/  
root@ip-172-31-93-2:/home/ubuntu/.ssh# ls  
authorized_keys  
authorized_keys  
root@ip-172-31-93-2:/home/ubuntu/.ssh# vim authorized_keys  
root@ip-172-31-93-2:/home/ubuntu/.ssh# vim authorized_keys  
root@ip-172-31-93-2:/home/ubuntu/.ssh# cat authorized_keys  
ssh-rsa AAAQAB3NzaC1yc2EAAQABAAAQc9Qredpa00Bn0qfrrZtMKA/+4P+kITf1nQabNNsKdUFc+hO1GbVsm/r30RcQf1216uz6JfSgbaxLtOUTEtLXpsaw3R17jWaRxwTuW++8NDvzVqE4AP008DQFAdpYfp7puaeKKq2m0t9UDqbuVfrOyc0J9sgss5tob5vhGnfpMkratifh/6i6AXVJ+JCTBuuyFd+r1XjwuMc0grzr5f1/2BtJ7v2Fc219M0G0p15VE15+1uXWW5K+THDNVCVmzbpbX1q2MgghiLPPPhqMxfxRIRksL6CLrunwN2zmlX3xTDBQXzjpbM6miiDHNRkBFFFxm2a1tJYO1Ctn51 SIAYERKEY  
ssh-ed25519 AAAAC3NzaC1ZD3DTE5AAAQJlf461ZzhkKaZT4wHRAoZLpYDfQpB71N8oolu4ZsyMo ubuntu@ip-172-31-85-119  
root@ip-172-31-93-2:/home/ubuntu/.ssh# ]
```

i-076061495bb4039d5 (Manage_Node1)
PublicIPs: 52.90.178.100 PrivateIPs: 172.31.93.2

CloudShell Feedback

Repeat the process on manage node 2.

```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | Start Course | Intellipaat | +
```

https://ubuntu.com/aws/pro
Expanded Security Maintenance for Applications is not enabled.
82 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

```
*** System restart required ***  
Last login: Tue Sep 3 06:13:51 2024 from 18.206.107.28  
ubuntu@ip-172-31-93-253:~$ sudo su  
root@ip-172-31-93-253:/home/ubuntu# /home/ubuntu/.ssh/  
bashi: /home/ubuntu/.ssh/: Is a directory  
root@ip-172-31-93-253:/home/ubuntu# cd .  
root@ip-172-31-93-253:/home/ubuntu#  
root@ip-172-31-93-253:/home/ubuntu# cd ..  
root@ip-172-31-93-253:/home# dc..  
dc..: command not found  
root@ip-172-31-93-253:/home# cd ..  
root@ip-172-31-93-253:# cd ..  
root@ip-172-31-93-253:# sudo su  
root@ip-172-31-93-253:# cd /home/ubuntu/.ssh  
root@ip-172-31-93-253:/home/ubuntu/.ssh# ls  
authorized_keys  
root@ip-172-31-93-253:/home/ubuntu/.ssh# vim authorized_keys]
```

i-0cd0f347cb719fc81 (Manage_Node2)
PublicIPs: 54.163.183.136 PrivateIPs: 172.31.93.253

CloudShell Feedback

The screenshot shows a terminal window in the AWS CloudShell interface. The user has run the command `ssh -t i-0cd0f347cb719fc81 'cat > /etc/ssh/slavekey'`. The terminal output shows the contents of the file being written:

```
-----  
-- INSERT --  
i-0cd0f347cb719fc81 (Manage_Node2)  
PublicIPs: 54.163.183.136 PrivateIPs: 172.31.93.253
```

This screenshot shows the continuation of the previous terminal session. The user has run the command `ssh -t i-0cd0f347cb719fc81 'cat > /etc/ssh/slavekey'`. The terminal output shows the contents of the file being written, which is significantly longer than in the first screenshot.

```
-----  
-- INSERT --  
i-0cd0f347cb719fc81 (Manage_Node2)  
PublicIPs: 54.163.183.136 PrivateIPs: 172.31.93.253
```

```
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Tue Sep 3 06:13:51 2024 from 18.206.107.28
ubuntu@ip-172-31-93-253:~$ sudo su
root@ip-172-31-93-253:/home/ubuntu# /home/ubuntu/.ssh/
bash: /home/ubuntu/.ssh/: Is a directory
root@ip-172-31-93-253:/home/ubuntu# cd .
root@ip-172-31-93-253:/home/ubuntu# root@ip-172-31-93-253:/home/ubuntu# cd ..
root@ip-172-31-93-253:/home/ubuntu# dc..
dc: command not found
root@ip-172-31-93-253:/home# cd ..
root@ip-172-31-93-253:# cd ..
root@ip-172-31-93-253:# sudo su
root@ip-172-31-93-253:# cd /home/ubuntu/.ssh
root@ip-172-31-93-253:/home/ubuntu/.ssh# ls
authorized_keys
root@ip-172-31-93-253:/home/ubuntu/.ssh# vim authorized_keys
root@ip-172-31-93-253:/home/ubuntu/.ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAQAAQAAQCC5QKedpa00Ubfm0gYTztkMKA/4P+kiTflnQabMNskdUfc+h01Ubvsm/r30EcQf1z16uz6JfSgbaxLtOUTEtLxpsaw3R17jWaRxwTuW++8NDv2VqE4APO08DQFAGp
Yp7puaeNkq2m0t9UDqcbv1FKOyC0J9sgSS5tobSvpbGnfpMkrat1fb/gi6AxVj+JCTBuyFd+r1xjwuRMc0p15Vb15+1Uxww5K+THDNVCmbbpX1q2MqMgghLPPhqMxfRirksL6CL
Unwn2zmlX3xTDBQXzgPbMemuilDHNRkBFFxm2AIJYO1ctFN51 SLAVEKEY
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIEJlf461ZzhkKaZT4wHRAoZlpYDfQpB71N8Oolu4ZsyMo ubuntu@ip-172-31-93-253:/home/ubuntu/.ssh# 
```

i-0cd0f347cb719fc81 (Manage_Node2)

PublicIPs: 54.163.183.136 PrivateIPs: 172.31.93.253

The public key has been successfully copied to manage node 2.

```
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
see https://ubuntu.com/esm or run: sudo pro status

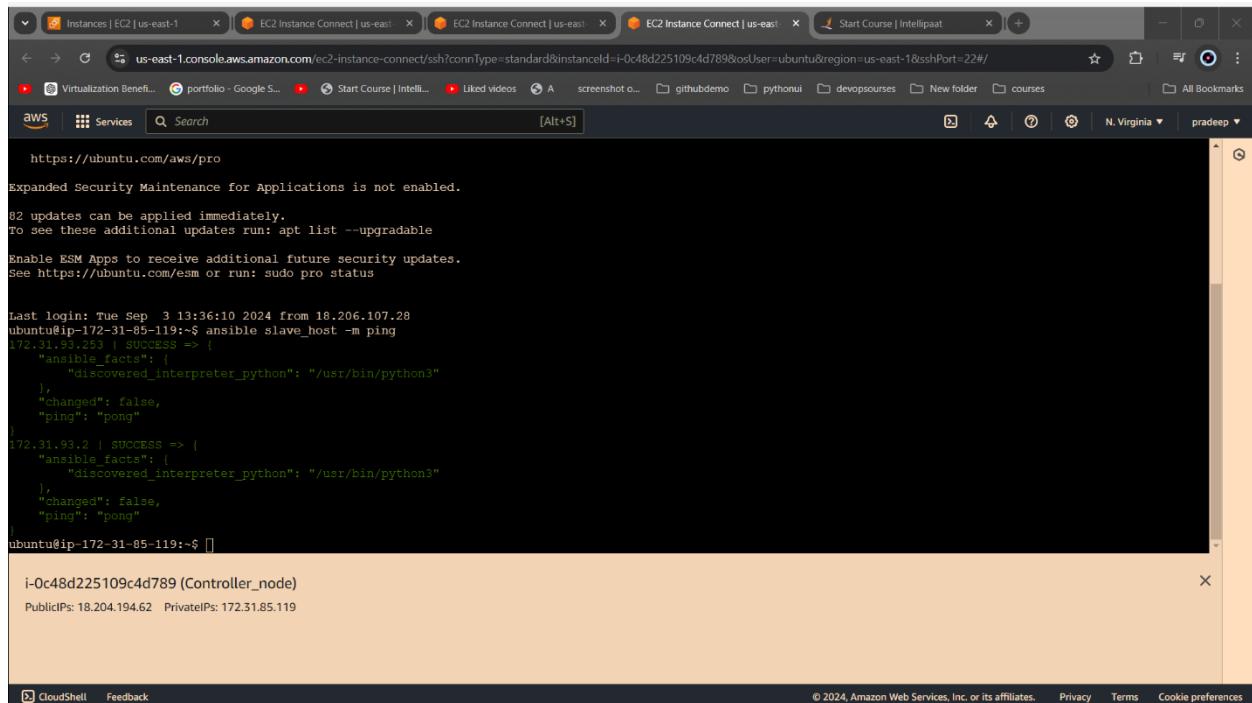
*** System restart required ***
Last login: Tue Sep 3 06:13:51 2024 from 18.206.107.28
ubuntu@ip-172-31-93-253:~$ sudo su
root@ip-172-31-93-253:/home/ubuntu# /home/ubuntu/.ssh/
bash: /home/ubuntu/.ssh/: Is a directory
root@ip-172-31-93-253:/home/ubuntu# cd .
root@ip-172-31-93-253:/home/ubuntu# root@ip-172-31-93-253:/home/ubuntu# cd ..
root@ip-172-31-93-253:/home# dc..
dc: command not found
root@ip-172-31-93-253:/home# cd ..
root@ip-172-31-93-253:# cd ..
root@ip-172-31-93-253:# sudo su
root@ip-172-31-93-253:# cd /home/ubuntu/.ssh
root@ip-172-31-93-253:/home/ubuntu/.ssh# ls
authorized_keys
root@ip-172-31-93-253:/home/ubuntu/.ssh# vim authorized_keys
root@ip-172-31-93-253:/home/ubuntu/.ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAQAAQAAQCC5QKedpa00Ubfm0gYTztkMKA/4P+kiTflnQabMNskdUfc+h01Ubvsm/r30EcQf1z16uz6JfSgbaxLtOUTEtLxpsaw3R17jWaRxwTuW++8NDv2VqE4APO08DQFAGp
Yp7puaeNkq2m0t9UDqcbv1FKOyC0J9sgSS5tobSvpbGnfpMkrat1fb/gi6AxVj+JCTBuyFd+r1xjwuRMc0p15Vb15+1Uxww5K+THDNVCmbbpX1q2MqMgghLPPhqMxfRirksL6CL
Unwn2zmlX3xTDBQXzgPbMemuilDHNRkBFFxm2AIJYO1ctFN51 SLAVEKEY
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIEJlf461ZzhkKaZT4wHRAoZlpYDfQpB71N8Oolu4ZsyMo ubuntu@ip-172-31-93-253:/home/ubuntu/.ssh# 
```

i-0cd0f347cb719fc81 (Manage_Node2)

PublicIPs: 54.163.183.136 PrivateIPs: 172.31.93.253

Use this command to verify the connection between the controller node and the manage node.

→ Ansible slave_host -m ping



```
https://ubuntu.com/aws/pro
Expanded Security Maintenance for Applications is not enabled.
82 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Sep  3 13:36:10 2024 from 18.206.107.28
ubuntu@ip-172-31-85-119:~$ ansible slave_host -m ping
172.31.93.253 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
172.31.93.2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-85-119:~$ []

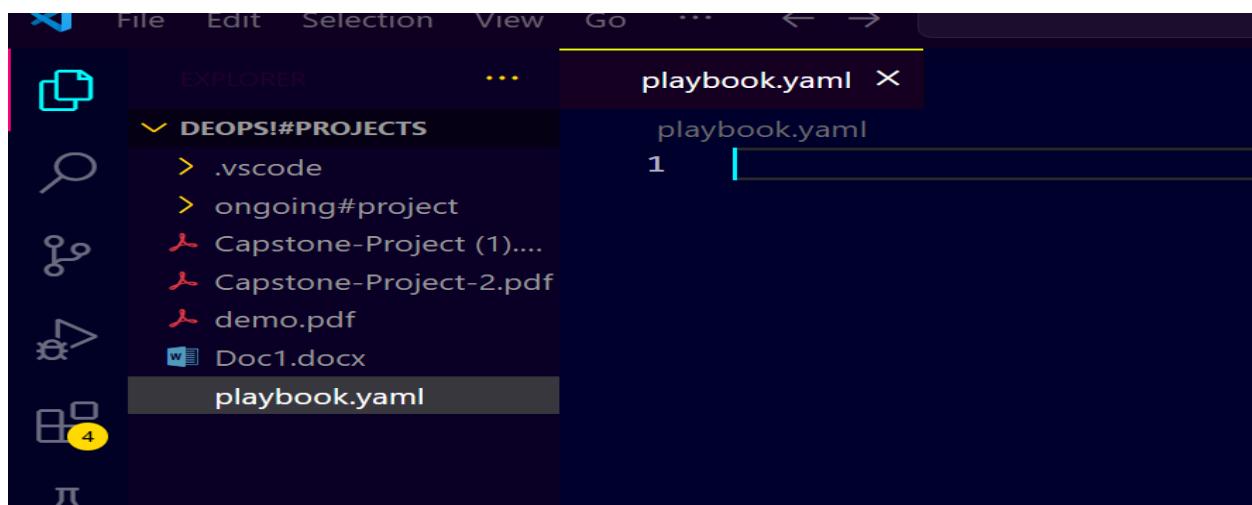
i-0c48d225109c4d789 (Controller_node)
PublicIPs: 18.204.194.62 PrivateIPs: 172.31.85.119
```

Successfully connected to manage nodes.

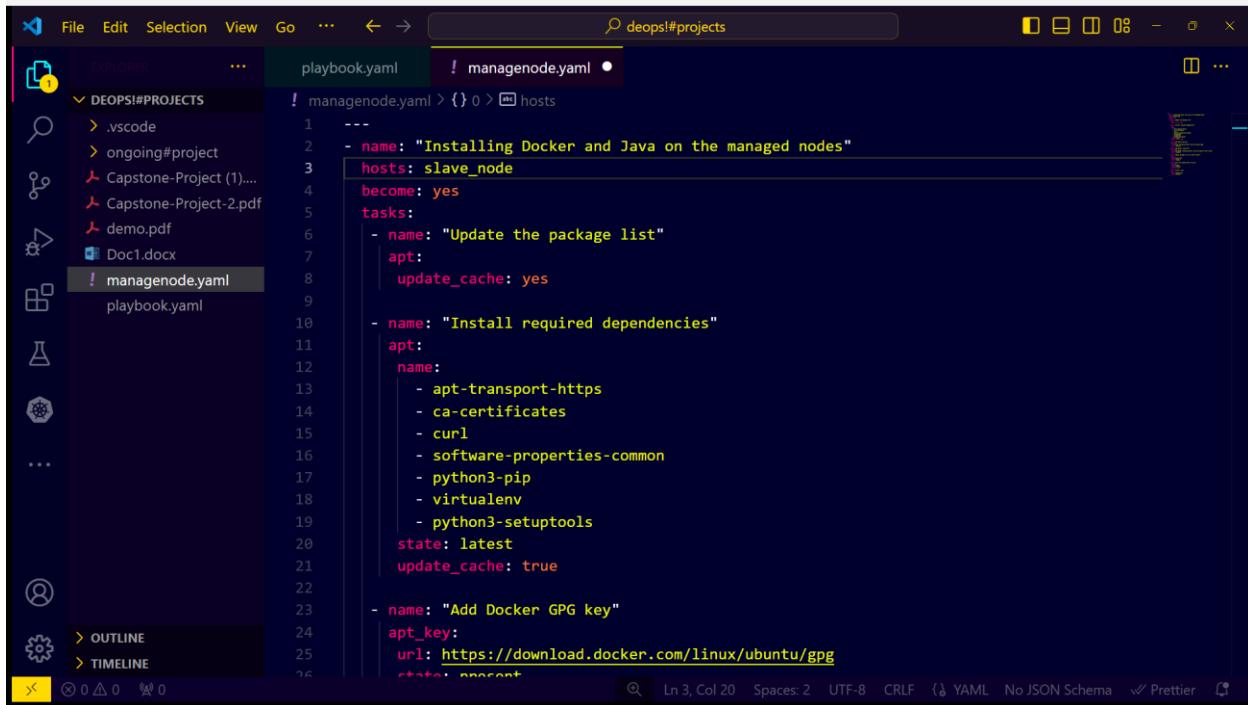
After successfully installing Ansible and connecting the manage nodes to the controller, the next step is to write a playbook that installs packages on the manage nodes and Jenkins on the controller node. The playbook will ensure all packages are properly configured.

We will use a text editor, such as Visual Studio Code, to write the playbook.

Create a new file called `playbook.yml` in Visual Studio Code for writing the Ansible playbook.



Creating a playbook to install Docker and Java on the manage nodes.



The screenshot shows the Visual Studio Code interface with the 'managenode.yaml' file selected in the Explorer sidebar. The main editor area displays the following Ansible YAML code:

```
---
- name: "Installing Docker and Java on the managed nodes"
  hosts: slave_node
  become: yes
  tasks:
    - name: "Update the package list"
      apt:
        update_cache: yes

    - name: "Install required dependencies"
      apt:
        name:
          - apt-transport-https
          - ca-certificates
          - curl
          - software-properties-common
          - python3-pip
          - virtualenv
          - python3-setuptools
        state: latest
        update_cache: true

    - name: "Add Docker GPG key"
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
```

```
---
- name: "Installing Docker and Java on the managed nodes"
  hosts: slave_node
  become: yes
  tasks:
    - name: "Update the package list"
      apt:
        update_cache: yes

    - name: "Install required dependencies"
      apt:
        name:
          - apt-transport-https
          - ca-certificates
          - curl
          - software-properties-common
          - python3-pip
          - virtualenv
          - python3-setuptools
        state: latest
        update_cache: true

    - name: "Add Docker GPG key"
      apt_key:
```

```
url: https://download.docker.com/linux/ubuntu/gpg
state: present

- name: "Add Docker repository"
  apt_repository:
    repo: deb https://download.docker.com/linux/ubuntu focal stable
    state: present

- name: "Update package list and install Docker"
  apt:
    update_cache: yes
    name: docker-ce
    state: latest

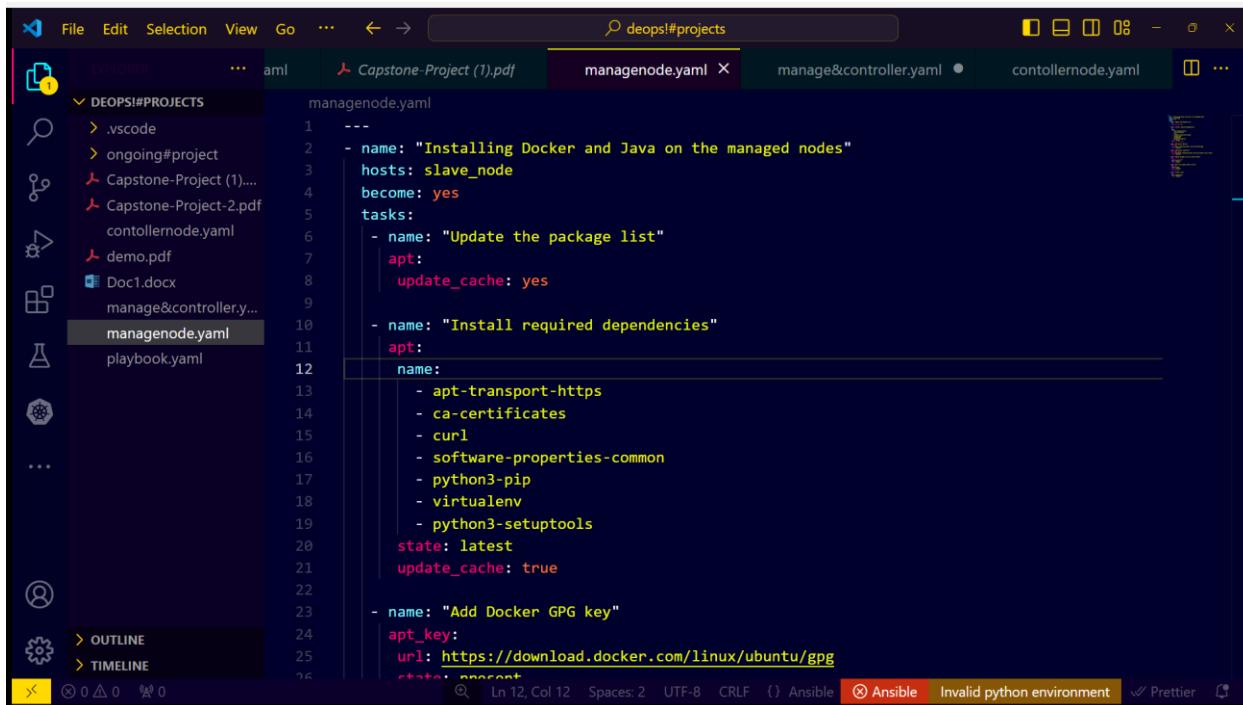
- name: "Start and enable Docker service"
  systemd:
    name: docker
    state: started
    enabled: yes

- name: "Install Java"
  apt:
    name: default-jdk
    state: present
...

```

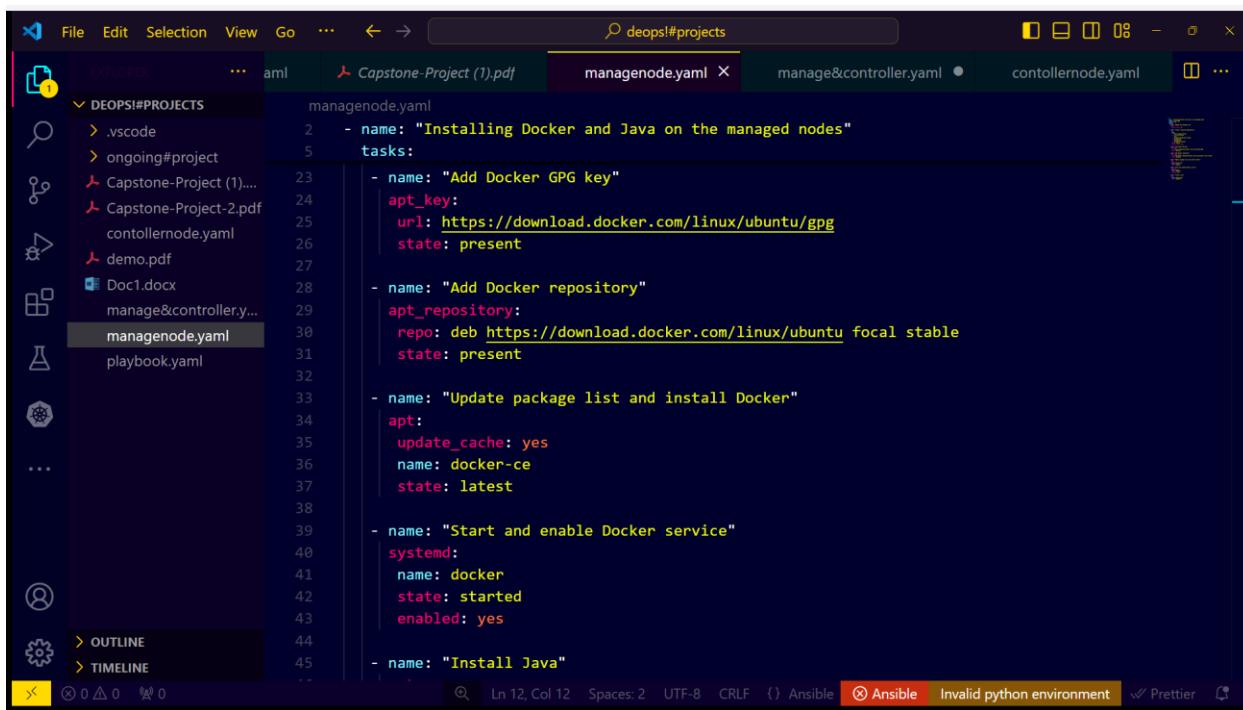
The playbook is configured to install Java and Docker on the manage nodes. We had earlier created a host group called `slave_node` in the inventory file, which includes all the slave nodes. In the playbook, we reference this host group in the hosts module.

Create a playbook to install Jenkins, Java, Docker, and Git on the controller node:



The screenshot shows the VS Code interface with the 'managenode.yaml' file open in the center editor tab. The file contains Ansible YAML code for managing a slave node. It includes tasks for updating the package list, installing required dependencies (apt), adding a Docker GPG key, and installing Docker. The code is color-coded for syntax highlighting.

```
1 ---  
2 - name: "Installing Docker and Java on the managed nodes"  
3 hosts: slave_node  
4 become: yes  
5 tasks:  
6   - name: "Update the package list"  
7     apt:  
8       update_cache: yes  
9  
10  - name: "Install required dependencies"  
11    apt:  
12      name:  
13        - apt-transport-https  
14        - ca-certificates  
15        - curl  
16        - software-properties-common  
17        - python3-pip  
18        - virtualenv  
19        - python3-setuptools  
20        state: latest  
21        update_cache: true  
22  
23  - name: "Add Docker GPG key"  
24    apt_key:  
25      url: https://download.docker.com/linux/ubuntu/gpg  
26      state: present  
27  
28  - name: "Add Docker repository"  
29    apt_repository:  
30      repo: deb https://download.docker.com/linux/ubuntu focal stable  
31      state: present  
32  
33  - name: "Update package list and install Docker"  
34    apt:  
35      update_cache: yes  
36      name: docker-ce  
37      state: latest  
38  
39  - name: "Start and enable Docker service"  
40    systemd:  
41      name: docker  
42      state: started  
43      enabled: yes  
44  
45  - name: "Install Java"  
46
```



The screenshot shows the same VS Code interface with the 'managenode.yaml' file open. In this version, the 'apt' task has been expanded to include Java installation, and the 'apt_repository' task has been removed. The code is color-coded for syntax highlighting.

```
1 ---  
2 - name: "Installing Docker and Java on the managed nodes"  
3 hosts: slave_node  
4 become: yes  
5 tasks:  
6   - name: "Update the package list"  
7     apt:  
8       update_cache: yes  
9  
10  - name: "Install required dependencies"  
11    apt:  
12      name:  
13        - apt-transport-https  
14        - ca-certificates  
15        - curl  
16        - software-properties-common  
17        - python3-pip  
18        - virtualenv  
19        - python3-setuptools  
20        state: latest  
21        update_cache: true  
22  
23  - name: "Add Docker GPG key"  
24    apt_key:  
25      url: https://download.docker.com/linux/ubuntu/gpg  
26      state: present  
27  
28  - name: "Install Java"  
29    apt:  
30      update_cache: yes  
31      name: oracle-java8-installer  
32      state: latest  
33  
34  - name: "Start and enable Docker service"  
35    systemd:  
36      name: docker  
37      state: started  
38      enabled: yes  
39  
40  - name: "Install Java"  
41    apt:  
42      update_cache: yes  
43      name: oracle-java8-installer  
44      state: latest  
45
```

The screenshot shows a dark-themed code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, ..., Back, Forward, Search bar (deops#projects), and window control buttons.
- Sidebar:** Explorer, Outline, Timeline, and other icons.
- File List:** DEOPS#PROJECTS folder containing .vscode, ongoing#project, Capstone-Project (1).pdf, Capstone-Project-2.pdf, contollernode.yaml, demo.pdf, Doc1.docx, manage&controller.y..., managenode.yaml (highlighted with a yellow box), and playbook.yaml.
- Code Editor:** The managenode.yaml file is open, showing Ansible YAML code. It includes tasks for installing Docker, enabling it as a service, and installing Java.
- Bottom Bar:** Search, Ln 12, Col 12, Spaces: 2, UTF-8, CRLF, Ansible, Ansible status (Invalid python environment), and Prettier.

```
managenode.yaml
  - name: "Installing Docker and Java on the managed nodes"
    tasks:
      - name: "Start and enable Docker service"
        systemd:
          name: docker
          state: started
          enabled: yes
      - name: "Install Java"
        apt:
          name: default-jdk
          state: present
```

```
---
```

- name: "Installing Docker and Java on the managed nodes"
 - hosts: slave_node
 - become: yes
 - tasks:
 - name: "Update the package list"
 - apt:
 - update_cache: yes
 - name: "Install required dependencies"
 - apt:
 - name:
 - apt-transport-https
 - ca-certificates
 - curl
 - software-properties-common
 - python3-pip
 - virtualenv
 - python3-setuptools
 - state: latest
 - update_cache: true
 - name: "Add Docker GPG key"
 - apt_key:

```

url: https://download.docker.com/linux/ubuntu/gpg
state: present

- name: "Add Docker repository"
  apt_repository:
    repo: deb https://download.docker.com/linux/ubuntu focal stable
    state: present

- name: "Update package list and install Docker"
  apt:
    update_cache: yes
    name: docker-ce
    state: latest

- name: "Start and enable Docker service"
  systemd:
    name: docker
    state: started
    enabled: yes

- name: "Install Java"
  apt:
    name: default-jdk
    state: present
...

```

Create a file named playbook1.yaml on the controller node and paste the YAML code that will run on the manage nodes.

```

aws | Services | Search [Alt+S]
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1014-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Sat Sep 7 11:14:48 UTC 2024
System load: 0.0 Processes: 114
Usage of /: 13.2% of 18.33GB Users logged in: 0
Memory usage: 5% IPv4 address for enx0: 172.31.85.119
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

82 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Sep 7 11:07:45 2024 from 18.206.107.29
ubuntu@ip-172-31-85-119:~$ █

i-0c48d225109c4d789 (Controller_node)
Public IPs: 35.174.13.126 Private IPs: 172.31.85.119

```

Create a file named playbook1.yaml on the controller node and paste the YAML code that will run on the manage nodes.

```
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1014-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Sat Sep 7 11:14:40 UTC 2024

System load: 0.0          Processes:           114
Usage of /: 13.2% of 18.33GB   Users logged in: 0
Memory usage: 5%          IPv4 address for enX0: 172.31.85.119
Swap usage: 0B

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

02 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

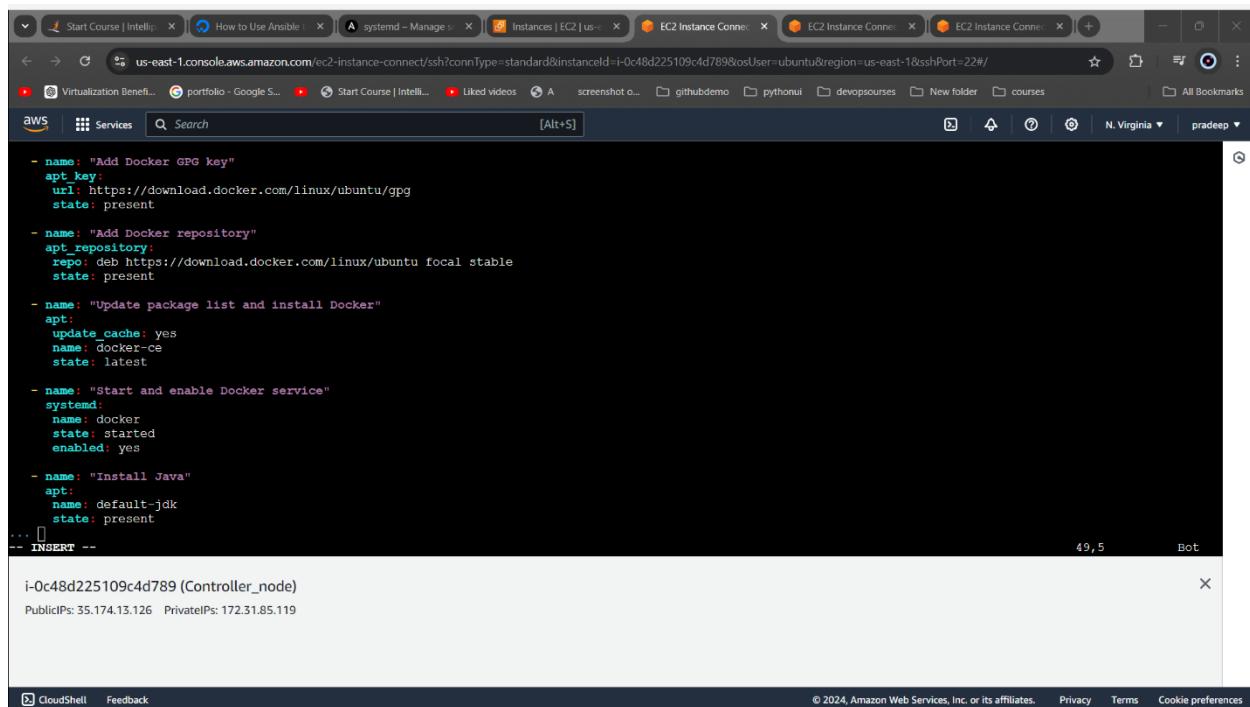
Last login: Sat Sep 7 11:07:45 2024 from 18.206.107.29
ubuntu@ip-172-31-85-119:~$ vim playbook1.yaml[

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119
```

```
-- INSERT --
i-0c48d225109c4d789 (Controller_node)
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119
```

Opened the file with the Vim text editor.

Copy the code to the playbook.yaml file.



```
- name: "Add Docker GPG key"
  apt_key:
    url: https://download.docker.com/linux/ubuntu/gpg
  state: present

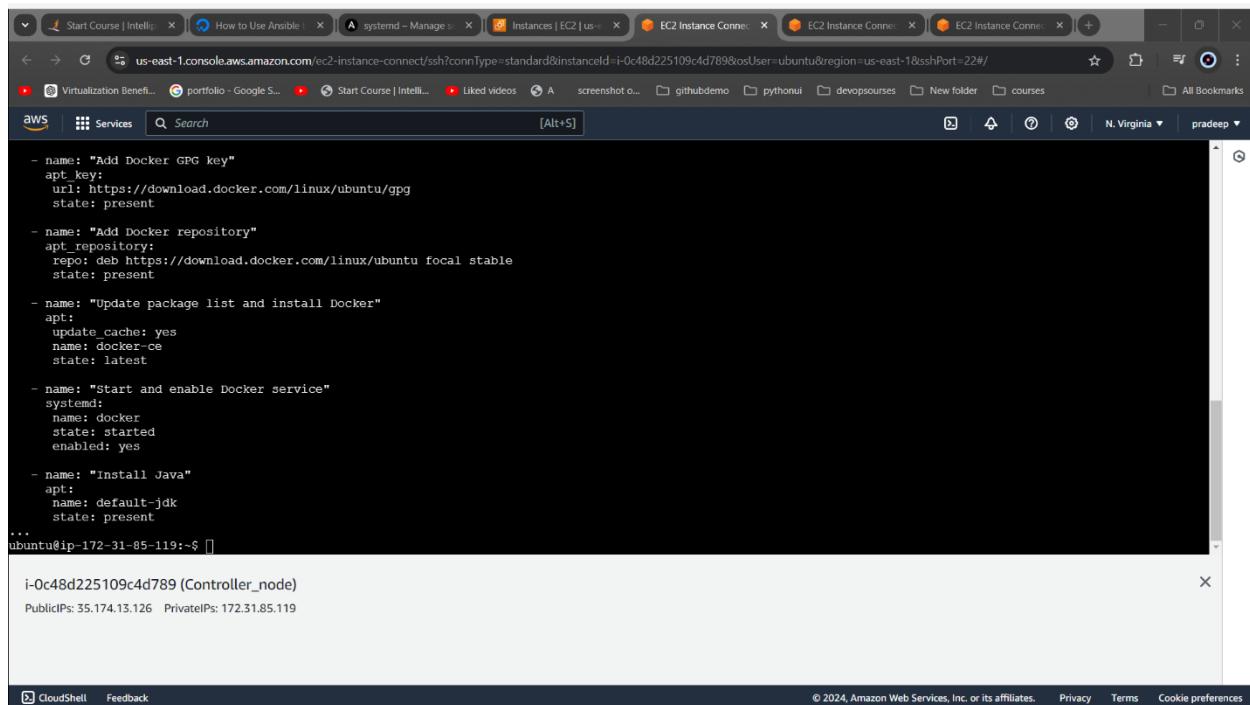
- name: "Add Docker repository"
  apt_repository:
    repo: deb https://download.docker.com/linux/ubuntu focal stable
  state: present

- name: "Update package list and install Docker"
  apt:
    update_cache: yes
  name: docker-ce
  state: latest

- name: "Start and enable Docker service"
  systemd:
    name: docker
    state: started
    enabled: yes

- name: "Install Java"
  apt:
    name: default-jdk
    state: present
...
-- INSERT --
i-0c48d225109c4d789 (Controller_node)
Public IPs: 35.174.13.126 Private IPs: 172.31.85.119
```

Save the file and check for errors by compiling it.



```
- name: "Add Docker GPG key"
  apt_key:
    url: https://download.docker.com/linux/ubuntu/gpg
  state: present

- name: "Add Docker repository"
  apt_repository:
    repo: deb https://download.docker.com/linux/ubuntu focal stable
  state: present

- name: "Update package list and install Docker"
  apt:
    update_cache: yes
  name: docker-ce
  state: latest

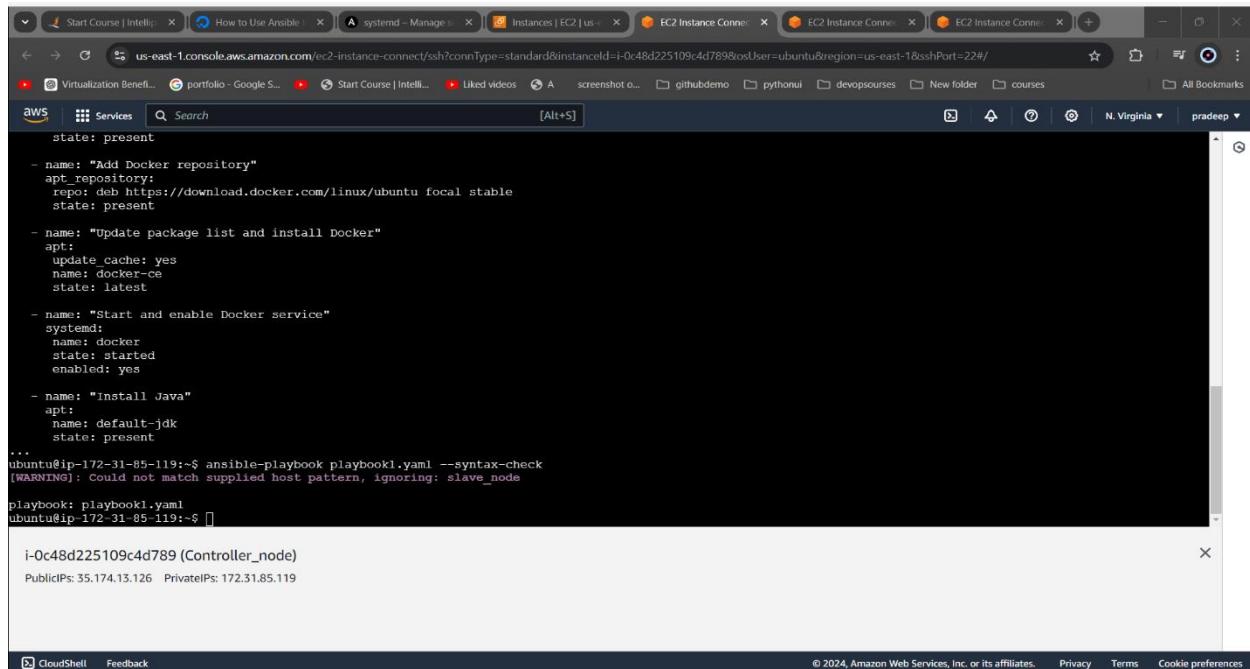
- name: "Start and enable Docker service"
  systemd:
    name: docker
    state: started
    enabled: yes

- name: "Install Java"
  apt:
    name: default-jdk
    state: present
...
ubuntu@ip-172-31-85-119:~$ []
i-0c48d225109c4d789 (Controller_node)
Public IPs: 35.174.13.126 Private IPs: 172.31.85.119
```

File is saved.

→ Syntax check:

Ansible-playbook playbook1.yaml --syntax-check

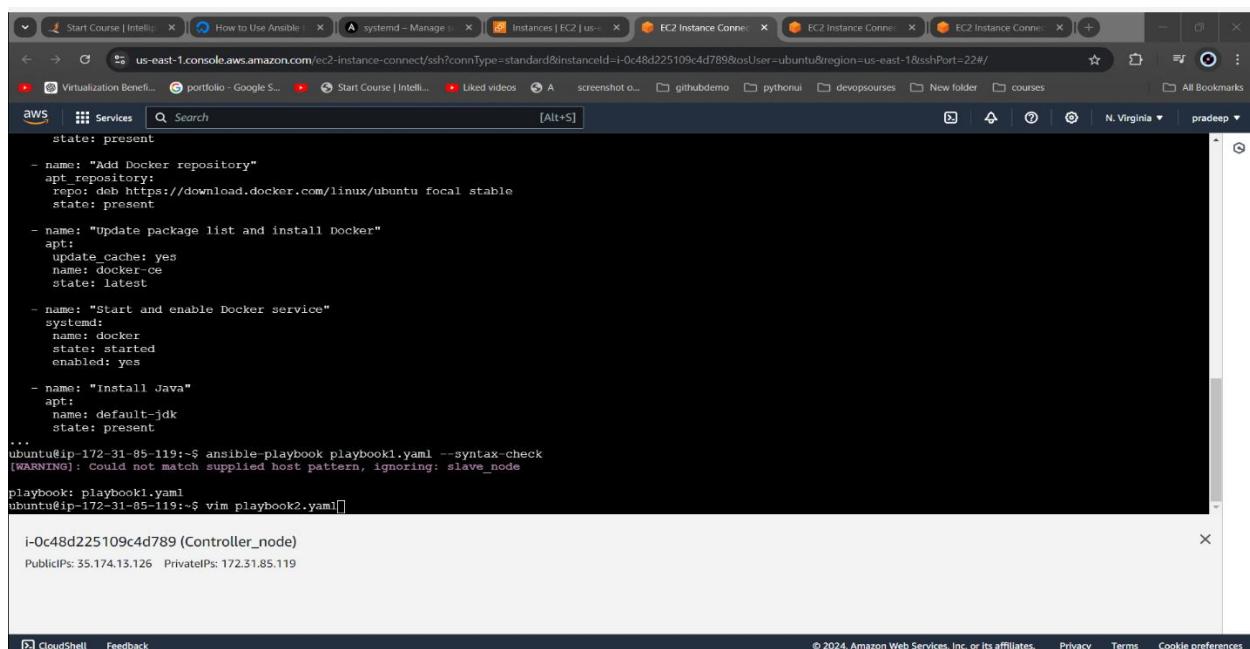


```
Start Course | IntelliJ x How to Use Ansible x A systemctl - Manage x Instances | EC2 | us- x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu&region=us-east-1&sshPort=22#/  
Virtualization Benefit... portfolio - Google S... Start Course | IntelliJ... Liked videos A screenshot o... githubdemo pythonui devopsourses New folder courses All Bookmarks N. Virginia pradeep  
aws Services Search [Alt+S]  
state: present  
- name: "Add Docker repository"  
  apt_repository:  
    repo: deb https://download.docker.com/linux/ubuntu focal stable  
  state: present  
- name: "Update package list and install Docker"  
  apt:  
    update_cache: yes  
    name: docker-ce  
    state: latest  
- name: "Start and enable Docker service"  
  systemd:  
    name: docker  
    state: started  
    enabled: yes  
- name: "Install Java"  
  apt:  
    name: default-jdk  
    state: present  
...  
ubuntu@ip-172-31-85-119:~$ ansible-playbook playbook1.yaml --syntax-check  
[WARNING]: Could not match supplied host pattern, ignoring: slave_node  
playbook: playbook1.yaml  
ubuntu@ip-172-31-85-119:~$  
i-0c48d225109c4d789 (Controller_node)  
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

No syntax errors were detected.

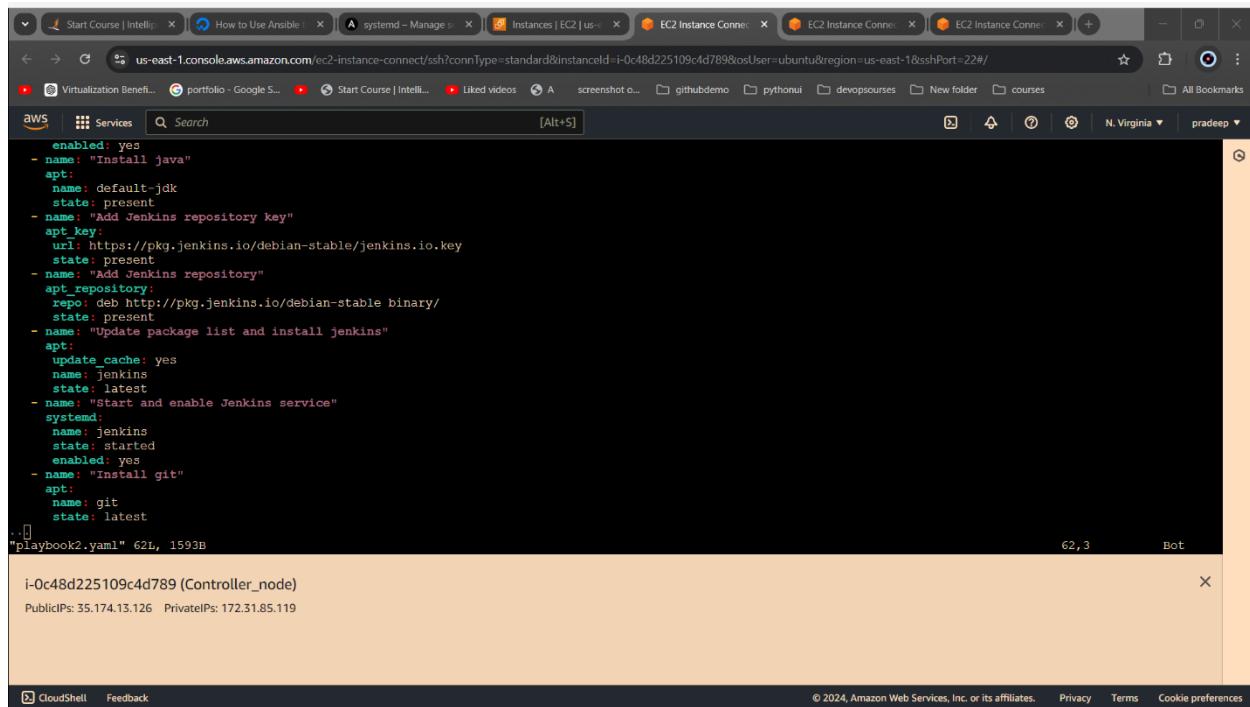
Create a file named playbook2.yaml and check its syntax for errors.

→ Vim playbook2.yaml



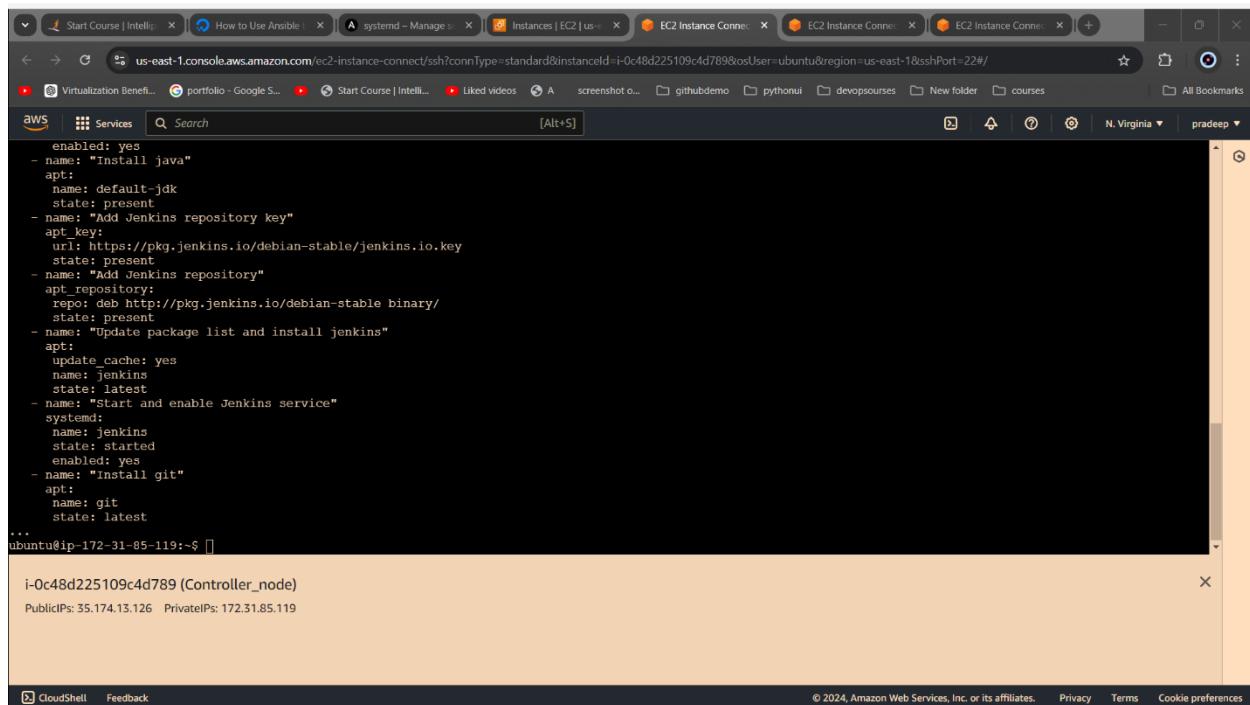
```
Start Course | IntelliJ x How to Use Ansible x A systemctl - Manage x Instances | EC2 | us- x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x EC2 Instance Connect x +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu&region=us-east-1&sshPort=22#/  
Virtualization Benefit... portfolio - Google S... Start Course | IntelliJ... Liked videos A screenshot o... githubdemo pythonui devopsourses New folder courses All Bookmarks N. Virginia pradeep  
aws Services Search [Alt+S]  
state: present  
- name: "Add Docker repository"  
  apt_repository:  
    repo: deb https://download.docker.com/linux/ubuntu focal stable  
  state: present  
- name: "Update package list and install Docker"  
  apt:  
    update_cache: yes  
    name: docker-ce  
    state: latest  
- name: "Start and enable Docker service"  
  systemd:  
    name: docker  
    state: started  
    enabled: yes  
- name: "Install Java"  
  apt:  
    name: default-jdk  
    state: present  
...  
ubuntu@ip-172-31-85-119:~$ vim playbook2.yaml  
ubuntu@ip-172-31-85-119:~$ ansible-playbook playbook1.yaml --syntax-check  
[WARNING]: Could not match supplied host pattern, ignoring: slave_node  
playbook: playbook1.yaml  
ubuntu@ip-172-31-85-119:~$  
i-0c48d225109c4d789 (Controller_node)  
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Paste the code intended to run on the controller node.



A screenshot of the AWS CloudShell interface. The terminal window displays an Ansible playbook named "playbook2.yaml". The playbook installs Java, adds Jenkins repository keys, adds Jenkins repositories, updates package lists, installs Jenkins, and installs Git. The terminal shows the execution of the playbook on an EC2 instance with ID i-0c48d225109c4d789, which has public IP 35.174.13.126 and private IP 172.31.85.119. The terminal window includes standard AWS CloudShell controls like CloudShell and Feedback buttons at the bottom.

```
aws | Services | Search [Alt+S]
[...]
enabled: yes
- name: "Install java"
  apt:
    name: default-jdk
    state: present
- name: "Add Jenkins repository key"
  apt_key:
    url: https://pkg.jenkins.io/debian-stable/jenkins.io.key
    state: present
- name: "Add Jenkins repository"
  apt_repository:
    repo: deb http://pkg.jenkins.io/debian-stable binary/
    state: present
- name: "Update package list and install jenkins"
  apt:
    update_cache: yes
    name: jenkins
    state: latest
- name: "Start and enable Jenkins service"
  systemd:
    name: jenkins
    state: started
    enabled: yes
- name: "Install git"
  apt:
    name: git
    state: latest
...
"playbook2.yaml" 62L, 1593B
62,3 Bot
i-0c48d225109c4d789 (Controller_node)
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119
[...]
CloudShell Feedback
```



A second screenshot of the AWS CloudShell interface, showing the same Ansible playbook execution. The terminal output is identical to the first screenshot, indicating successful installation of Jenkins and Git on the EC2 instance. The terminal window includes standard AWS CloudShell controls like CloudShell and Feedback buttons at the bottom.

```
aws | Services | Search [Alt+S]
[...]
enabled: yes
- name: "Install java"
  apt:
    name: default-jdk
    state: present
- name: "Add Jenkins repository key"
  apt_key:
    url: https://pkg.jenkins.io/debian-stable/jenkins.io.key
    state: present
- name: "Add Jenkins repository"
  apt_repository:
    repo: deb http://pkg.jenkins.io/debian-stable binary/
    state: present
- name: "Update package list and install jenkins"
  apt:
    update_cache: yes
    name: jenkins
    state: latest
- name: "Start and enable Jenkins service"
  systemd:
    name: jenkins
    state: started
    enabled: yes
- name: "Install git"
  apt:
    name: git
    state: latest
...
ubuntu@ip-172-31-85-119:~$ []
i-0c48d225109c4d789 (Controller_node)
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119
[...]
CloudShell Feedback
```

```
aws | Services | Search [Alt+S]
[...]
- name: "Install java"
  apt:
    name: default-jdk
    state: present
- name: "Add Jenkins repository key"
  apt_key:
    url: https://pkg.jenkins.io/debian-stable/jenkins.io.key
    state: present
- name: "Add Jenkins repository"
  apt_repository:
    repo: deb http://pkg.jenkins.io/debian-stable binary/
    state: present
- name: "Update package list and install jenkins"
  apt:
    update_cache: yes
    name: Jenkins
    state: latest
- name: "Start and enable Jenkins service"
  systemd:
    name: Jenkins
    state: started
    enabled: yes
- name: "Install git"
  apt:
    name: git
    state: latest
...
ubuntu@ip-172-31-85-119:~$ ansible-playbook playbook2.yaml --syntax-check
i-0c48d225109c4d789 (Controller_node)
Public IPs: 35.174.13.126 Private IPs: 172.31.85.119
```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Syntax checking is complete, and no errors were found.

To execute the playbook, use the command below:

→ Ansible_playbook playbook1

```
aws | Services | Search [Alt+S]
[...]
name: default-jdk
state: present
- name: "Add Jenkins repository key"
  apt_key:
    url: https://pkg.jenkins.io/debian-stable/jenkins.io.key
    state: present
- name: "Add Jenkins repository"
  apt_repository:
    repo: deb http://pkg.jenkins.io/debian-stable binary/
    state: present
- name: "Update package list and install jenkins"
  apt:
    update_cache: yes
    name: Jenkins
    state: latest
- name: "Start and enable Jenkins service"
  systemd:
    name: Jenkins
    state: started
    enabled: yes
- name: "Install git"
  apt:
    name: git
    state: latest
...
ubuntu@ip-172-31-85-119:~$ ansible-playbook playbook2.yaml --syntax-check
playbook: playbook2.yaml
ubuntu@ip-172-31-85-119:~$ ansible-playbook playbook1.yaml
i-0c48d225109c4d789 (Controller_node)
Public IPs: 35.174.13.126 Private IPs: 172.31.85.119
```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Installing package:

The screenshot shows a terminal window in the AWS CloudShell interface. The terminal output is as follows:

```
Expanded Security Maintenance for Applications is not enabled.  
02 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Sat Sep 7 14:49:00 2024 from 10.206.107.28  
ubuntu@ip-172-31-85-119:~$ ls  
playbook1.yaml playbook2.yaml  
ubuntu@ip-172-31-85-119:~$ vim playbook1.yaml  
ubuntu@ip-172-31-85-119:~$ ansible-playbook playbook1.yaml  
  
PLAY [Installing Docker and Java on the managed nodes] *****  
TASK [Gathering Facts] *****  
ok: [172.31.93.2]  
ok: [172.31.93.253]  
  
TASK [Update the package list] *****  
changed: [172.31.93.2]  
changed: [172.31.93.253]  
  
TASK [Install required dependencies] *****  
ok: [172.31.93.253]  
  
i-0c48d225109c4d789 (Controller_node)  
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119
```

Playbook 1 was executed successfully.

The screenshot shows a terminal window in the AWS CloudShell interface. The terminal output is as follows:

```
TASK [Install required dependencies] *****  
changed: [172.31.93.2]  
changed: [172.31.93.253]  
  
TASK [Add Docker GPG key] *****  
changed: [172.31.93.253]  
changed: [172.31.93.2]  
  
TASK [Add Docker repository] *****  
changed: [172.31.93.2]  
changed: [172.31.93.253]  
  
TASK [Update package list and install Docker] *****  
changed: [172.31.93.253]  
changed: [172.31.93.2]  
  
TASK [Start and enable Docker service] *****  
ok: [172.31.93.253]  
ok: [172.31.93.2]  
  
TASK [Install Java] *****  
changed: [172.31.93.2]  
changed: [172.31.93.253]  
  
PLAY RECAP *****  
172.31.93.2 : ok=8 changed=6 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0  
172.31.93.253 : ok=8 changed=6 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0  
  
ubuntu@ip-172-31-85-119:~$
```

Manage node1:

Docker is installed:

To check -> sudo systemctl status docker

```
Start Course | IntelliJ IDEA
How to Use Ansible ...
systemd - Manage ...
Instances | EC2 | us-east-1 ...
EC2 Instance Connect ...
EC2 Instance Connect ...
EC2 Instance Connect ...
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-076061495bb4039d5&osUser=ubuntu&sshPort=22#
Virtualization Benefits portfolio - Google Sheets Start Course | IntelliJ IDEA Liked videos A screenshot of ... GitHub demo Python devopsources New folder courses All Bookmarks N. Virginia pradeep

AWS Services Search [Alt+S]
Management: https://landscape.canonical.com
Support: https://ubuntu.com/pro

System information as of Sat Sep 7 15:36:10 UTC 2024

System load: 0.0 Processes: 108
Usage of /: 55.5% of 6.71GB Users logged in: 1
Memory usage: 28% IPv4 address for enX0: 172.31.93.2
Swap usage: 0%

Ubuntu Pro delivers the most comprehensive open source security and compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

85 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Sep 7 15:20:38 2024 from 172.31.85.119
ubuntu@ip-172-31-93-2:~$ ls
ubuntu@ip-172-31-93-2:~$ sudo su
root@ip-172-31-93-2:/home/ubuntu# systemctl status docker

i-076061495bb4039d5 (Manage_Node1)
PublicIPs: 100.24.19.138 PrivateIPs: 172.31.93.2

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Check java installed or not:

→ Java –version

```
Start Course | IntelliJ IDEA
How to Use Ansible ...
systemd - Manage ...
Instances | EC2 | us-east-1 ...
EC2 Instance Connect ...
EC2 Instance Connect ...
EC2 Instance Connect ...
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-076061495bb4039d5&osUser=ubuntu&sshPort=22#
Virtualization Benefits portfolio - Google Sheets Start Course | IntelliJ IDEA Liked videos A screenshot of ... GitHub demo Python devopsources New folder courses All Bookmarks N. Virginia pradeep

AWS Services Search [Alt+S]
Sep 07 15:20:30 ip-172-31-93-2 systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 07 15:20:30 ip-172-31-93-2 dockerd[16005]: time="2024-09-07T15:20:30.817328281Z" level=info msg="Starting up"
Sep 07 15:20:30 ip-172-31-93-2 dockerd[16005]: time="2024-09-07T15:20:30.828077207Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so not using /etc/resolv.conf"
Sep 07 15:20:30 ip-172-31-93-2 dockerd[16005]: time="2024-09-07T15:20:31.103474949Z" level=info msg="Loading containers: start."
Sep 07 15:20:30 ip-172-31-93-2 dockerd[16005]: time="2024-09-07T15:20:31.103501515Z" level=info msg="loading container 3ab5c7d"
Sep 07 15:20:31 ip-172-31-93-2 dockerd[16005]: time="2024-09-07T15:20:31.648012416Z" level=info msg="Docker daemon" commit=3ab5c7d containerd-snapshotter=false storage-driver=overlay2
Sep 07 15:20:31 ip-172-31-93-2 dockerd[16005]: time="2024-09-07T15:20:31.649819388Z" level=info msg="Daemon has completed initialization"
Sep 07 15:20:31 ip-172-31-93-2 dockerd[16005]: time="2024-09-07T15:20:31.744949463Z" level=info msg="API listen on /run/docker.sock"
Sep 07 15:20:31 ip-172-31-93-2 systemd[1]: Started docker.service - Docker Application Container Engine.

root@ip-172-31-93-2:/home/ubuntu#
root@ip-172-31-93-2:/home/ubuntu# apt list --installed | grep java17
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

root@ip-172-31-93-2:/home/ubuntu# systemctl status java17
Unit java17.service could not be found.
root@ip-172-31-93-2:/home/ubuntu# java -version
openjdk version "17.0.4" 2024-04-16
OpenJDK 64-Bit Server VM (build 17.0.4+7-Ubuntu-1ubuntu224.04)
root@ip-172-31-93-2:/home/ubuntu#
```

Image node 2:

The screenshot shows a terminal window within an AWS CloudShell interface. The terminal displays system information for an Ubuntu 22.04 LTS instance. It includes details like system load (0.08), memory usage (27%), swap usage (0%), and network configuration (IPv4 address 172.31.93.253). The terminal also shows a command history with the last command being a system status check.

```
System information as of Sat Sep 7 15:42:25 UTC 2024
System load: 0.08 Processes: 107
Usage of /: 55.3% of 6.71GB Users logged in: 0
Memory usage: 27% IPv4 address for enx0: 172.31.93.253
Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.
https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

87 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Sep 7 15:20:38 2024 from 172.31.85.119
ubuntu@ip-172-31-93-253:~$ ls
ubuntu@ip-172-31-93-253:~$ i-0cd0f347cb719fc81 (Manage_Node2)
PublicIPs: 3.86.102.160 PrivateIPs: 172.31.93.253
```

→ Systemctl status docker

The screenshot shows a terminal window within an AWS CloudShell interface. The terminal displays system information for an Ubuntu 22.04 LTS instance. It includes details like system load (0.08), memory usage (27%), swap usage (0%), and network configuration (IPv4 address 172.31.93.253). The terminal also shows a command history with the last command being a system status check. A specific command, systemctl status docker, is highlighted in the terminal window.

```
System information as of Sat Sep 7 15:42:25 UTC 2024
System load: 0.08 Processes: 107
Usage of /: 55.3% of 6.71GB Users logged in: 0
Memory usage: 27% IPv4 address for enx0: 172.31.93.253
Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.
https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

87 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Sep 7 15:20:38 2024 from 172.31.85.119
ubuntu@ip-172-31-93-253:~$ ls
ubuntu@ip-172-31-93-253:~$ systemctl status docker
i-0cd0f347cb719fc81 (Manage_Node2)
PublicIPs: 3.86.102.160 PrivateIPs: 172.31.93.253
```

```

Last login: Sat Sep  7 15:20:38 2024 from 172.31.85.119
ubuntu@ip-172-31-93-253:~$ ls
ubuntu@ip-172-31-93-253:~$ systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Sun 2024-09-07 15:20:31 UTC; 22min ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Main PID: 6217 (dockerd)
   Tasks: 8
    Memory: 41.5M (peak: 91.8M)
      CPU: 344ms
     CGroup: /system.slice/docker.service
             └─6217 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 07 15:20:30 ip-172-31-93-253 systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.319650697Z" level=info msg="Starting up"
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.329140073Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, using /etc/resolv.conf"
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.547413673Z" level=info msg="Loading containers: start."
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.882709420Z" level=info msg="Loading containers: done."
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.916560826Z" level=info msg="Docker daemon" commit=3ab5c7d containerd-snapshotter=false
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.916855735Z" level=info msg="Daemon has completed initialization"
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.999395400Z" level=info msg="API listen on /run/docker.sock"
Sep 07 15:20:31 ip-172-31-93-253 systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)

i-0cd0f347cb719fc81 (Manage_Node2)
Public IPs: 3.86.102.160 Private IPs: 172.31.93.253

```

Check Java installed or not :

→ Java –version

```

Last login: Sat Sep  7 15:20:38 2024 from 172.31.85.119
ubuntu@ip-172-31-93-253:~$ ls
ubuntu@ip-172-31-93-253:~$ systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Sun 2024-09-07 15:20:31 UTC; 22min ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Main PID: 6217 (dockerd)
   Tasks: 8
    Memory: 41.5M (peak: 91.8M)
      CPU: 344ms
     CGroup: /system.slice/docker.service
             └─6217 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 07 15:20:30 ip-172-31-93-253 systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.319650697Z" level=info msg="Starting up"
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.329140073Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, using /etc/resolv.conf"
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.547413673Z" level=info msg="Loading containers: start."
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.882709420Z" level=info msg="Loading containers: done."
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.916560826Z" level=info msg="Docker daemon" commit=3ab5c7d containerd-snapshotter=false
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.916855735Z" level=info msg="Daemon has completed initialization"
Sep 07 15:20:31 ip-172-31-93-253 dockerd[6217]: time="2024-09-07T15:20:31.999395400Z" level=info msg="API listen on /run/docker.sock"
Sep 07 15:20:31 ip-172-31-93-253 systemd[1]: Started docker.service - Docker Application Container Engine.

java --version
openjdk 21.0.4 2024-07-16
OpenJDK Runtime Environment (build 21.0.4+7-Ubuntu-1ubuntu224.04)
OpenJDK 64-Bit Server VM (build 21.0.4+7-Ubuntu-1ubuntu224.04, mixed mode, sharing)
ubuntu@ip-172-31-93-253:~$ 

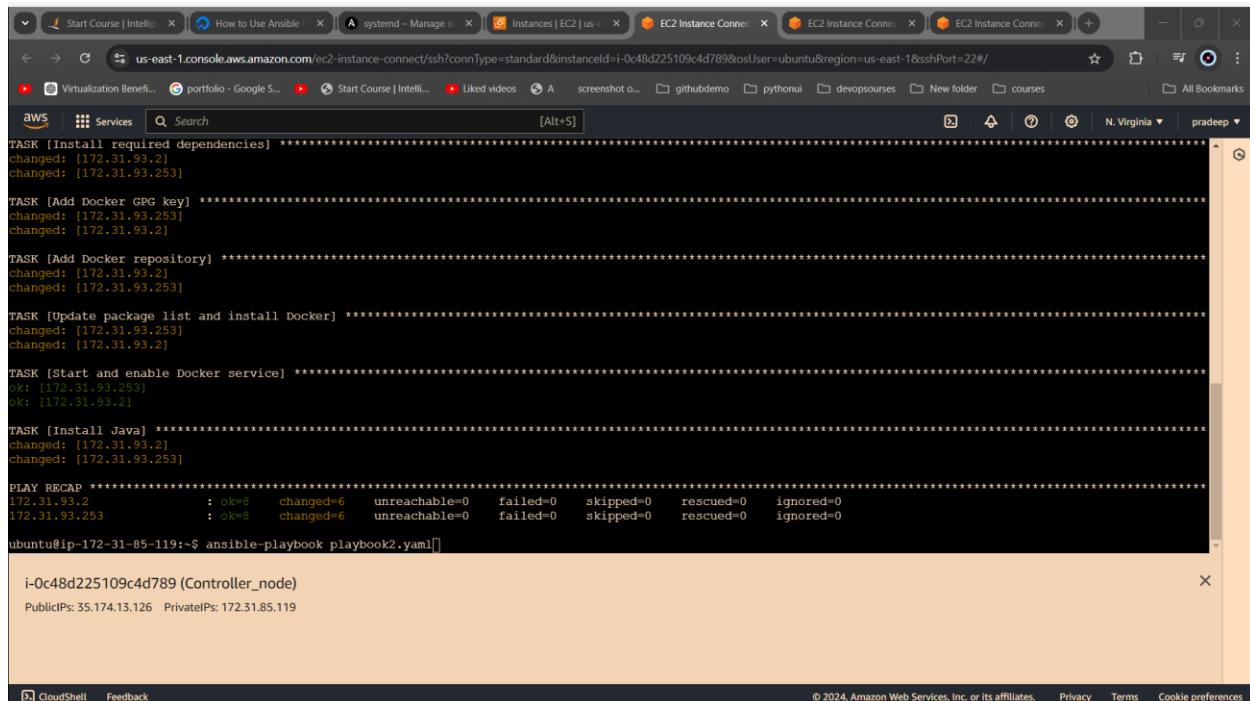
i-0cd0f347cb719fc81 (Manage_Node2)
Public IPs: 3.86.102.160 Private IPs: 172.31.93.253

```

Playbook 1 executed successfully, and the required packages were installed on both manage nodes.

Now, execute playbook2.

→ Ansible-playbook playbook2.yaml



```
TASK [Install required dependencies] *****
changed: [172.31.93.2]
changed: [172.31.93.253]

TASK [Add docker GPG key] *****
changed: [172.31.93.253]
changed: [172.31.93.2]

TASK [Add Docker repository] *****
changed: [172.31.93.2]
changed: [172.31.93.253]

TASK [Update package list and install Docker] *****
changed: [172.31.93.253]
changed: [172.31.93.2]

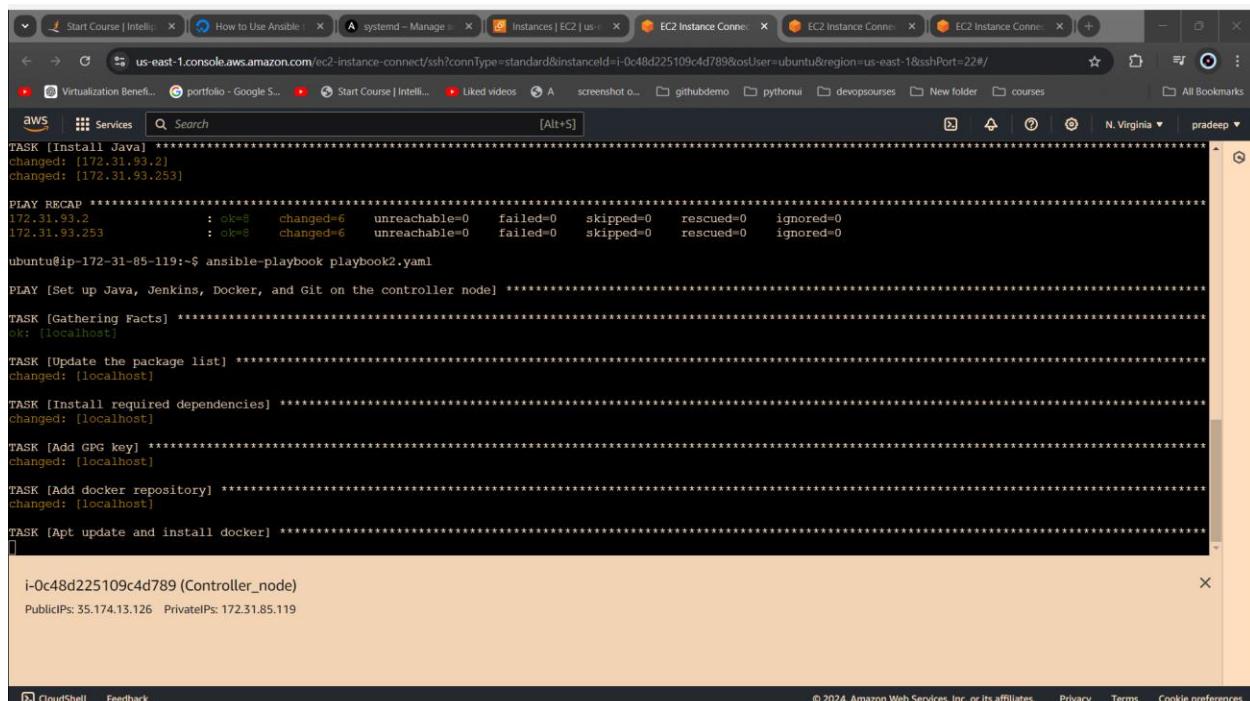
TASK [Start and enable Docker service] *****
ok: [172.31.93.253]
ok: [172.31.93.2]

TASK [Install Java] *****
changed: [172.31.93.2]
changed: [172.31.93.253]

PLAY RECAP *****
172.31.93.2      : ok=8    changed=6   unreachable=0    failed=0    skipped=0   rescued=0   ignored=0
172.31.93.253   : ok=8    changed=6   unreachable=0    failed=0    skipped=0   rescued=0   ignored=0

ubuntu@ip-172-31-85-119:~$ ansible-playbook playbook2.yaml[]

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119
```



```
TASK [Install Java] *****
changed: [172.31.93.2]
changed: [172.31.93.253]

PLAY RECAP *****
172.31.93.2      : ok=8    changed=6   unreachable=0    failed=0    skipped=0   rescued=0   ignored=0
172.31.93.253   : ok=8    changed=6   unreachable=0    failed=0    skipped=0   rescued=0   ignored=0

ubuntu@ip-172-31-85-119:~$ ansible-playbook playbook2.yaml

PLAY [Set up Java, Jenkins, Docker, and Git on the controller node] *****
TASK [Gathering Facts] *****
ok: [localhost]

TASK [Update the package list] *****
changed: [localhost]

TASK [Install required dependencies] *****
changed: [localhost]

TASK [Add GPG key] *****
changed: [localhost]

TASK [Add docker repository] *****
changed: [localhost]

TASK [Apt update and install docker] []
[]

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119
```

```
TASK [Apt update and install docker] ****
:ok: [localhost]

TASK [Start and Enable docker service] ****
:ok: [localhost]

TASK [Install java] ****
:ok: [localhost]

TASK [Add Jenkins repository key] ****
changed: [localhost]

TASK [Add Jenkins repository] ****
changed: [localhost]

TASK [Update package list and install jenkins] ****
changed: [localhost]

TASK [Start and enable Jenkins service] ****
:ok: [localhost]

TASK [Install git] ****
:ok: [localhost]

PLAY RECAP ****
localhost : ok=13    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-85-119:~$ 

i-0c48d225109c4d789 (Controller_node)
Public IPs: 35.174.13.126 Private IPs: 172.31.85.119
```

Playbook 2 executed successfully, installing Jenkins, Docker, and Java on the controller node.

➔ Systemctl status jenkins:

```
TASK [Apt update and install docker] ****
:ok: [localhost]

TASK [Start and Enable docker service] ****
:ok: [localhost]

TASK [Install java] ****
:ok: [localhost]

TASK [Add Jenkins repository key] ****
changed: [localhost]

TASK [Add Jenkins repository] ****
changed: [localhost]

TASK [Update package list and install jenkins] ****
changed: [localhost]

TASK [Start and enable Jenkins service] ****
:ok: [localhost]

TASK [Install git] ****
:ok: [localhost]

PLAY RECAP ****
localhost : ok=13    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# systemctl status jenkins

i-0c48d225109c4d789 (Controller_node)
Public IPs: 35.174.13.126 Private IPs: 172.31.85.119
```

```
Start Course How to Use Instances | EC2 Instance Connect EC2 Instance DevOps CloudShell Feedback
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu&region=us-east-1&sshPort=22/
Virtualization Benefits portfolio - Google Slides Start Course | IntelliJ IDEA Liked videos A screenshot of... githubdemo pythonui devopsources New folder courses All Bookmarks AWS Services Search [Alt+S]
TASK [Install git] ****
localhost ok: [localhost]

PLAY RECAP ****
localhost : ok=13    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-09-07 15:59:07 UTC; 3min 11s ago
       Main PID: 16811 (java)
          Tasks: 44 (limit: 4676)
        Memory: 573.7M (peak: 580.6M)
          CPU: 12.784s
         CGroup: /system.slice/jenkins.service
                   └─16811 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Sep 07 15:59:05 ip-172-31-85-119 jenkins[16811]: 4763ca0b06704830b8d0ae0e7f161b73
Sep 07 15:59:05 ip-172-31-85-119 jenkins[16811]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Sep 07 15:59:05 ip-172-31-85-119 jenkins[16811]: ****
Sep 07 15:59:07 ip-172-31-85-119 jenkins[16811]: 2024-09-07 15:59:07.903+0000 [id=56]      INFO    h.m.DownloadService$Downloadable#load: Obtained the update
Sep 07 15:59:07 ip-172-31-85-119 jenkins[16811]: 2024-09-07 15:59:07.905+0000 [id=56]      INFO    hudson.util.Retriger#start: Performed the action check update
Sep 07 15:59:07 ip-172-31-85-119 jenkins[16811]: 2024-09-07 15:59:07.928+0000 [id=40]      INFO    jenkins.InitReactorRunner$1#onAttained: Completed initialization
Sep 07 15:59:07 ip-172-31-85-119 jenkins[16811]: 2024-09-07 15:59:07.944+0000 [id=31]      INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
Sep 07 15:59:07 ip-172-31-85-119 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
lines 1-20/20 (END)
```

Jenkins is installed and it is running.

→ Systemctl status docker

```

root@ip-172-31-85-119:/home/ubuntu#
root@ip-172-31-85-119:/home/ubuntu# systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Sat 2024-09-07 15:05:59 UTC; 3min ago
TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 5876 (dockerd)
      Tasks: 9
     Memory: 21.2M (peak: 21.6M)
        CPU: 364ms
       CGroup: /system.slice/docker.service
               └─5876 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 07 15:30:58 ip-172-31-85-119 systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 07 15:30:58 ip-172-31-85-119 dockerd[5876]: time="2024-09-07T15:30:58.724252865Z" level=info msg="Starting up"
Sep 07 15:30:58 ip-172-31-85-119 dockerd[5876]: time="2024-09-07T15:30:58.727756103Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so"
Sep 07 15:30:58 ip-172-31-85-119 dockerd[5876]: time="2024-09-07T15:30:58.867932893Z" level=info msg="Loading containers: start."
Sep 07 15:30:59 ip-172-31-85-119 dockerd[5876]: time="2024-09-07T15:30:59.164577894Z" level=info msg="Loading containers: done."
Sep 07 15:30:59 ip-172-31-85-119 dockerd[5876]: time="2024-09-07T15:30:59.188159507Z" level=info msg="Docker daemon" commit=3abbc7d containerd-snapshotter=false st
Sep 07 15:30:59 ip-172-31-85-119 dockerd[5876]: time="2024-09-07T15:30:59.188267305Z" level=info msg="Daemon has completed initialization"
Sep 07 15:30:59 ip-172-31-85-119 dockerd[5876]: time="2024-09-07T15:30:59.242347469Z" level=info msg="API listen on /run/docker.sock"
Sep 07 15:30:59 ip-172-31-85-119 systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119

```

Docker is present in controller node.

→ Java – version

```

root@ip-172-31-85-119:/home/ubuntu# java --version
openjdk 21.0.4 2024-07-16
OpenJDK Runtime Environment (build 21.0.4+7-Ubuntu-1ubuntu224.04)
OpenJDK 64-Bit Server VM (build 21.0.4+7-Ubuntu-1ubuntu224.04, mixed mode, sharing)
root@ip-172-31-85-119:/home/ubuntu#

```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119

→ Git –version

```

ubuntu@ip-172-31-85-119:~$ playbook1.yaml playbook2.yaml
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# git --version
git version 2.43.0
root@ip-172-31-85-119:/home/ubuntu#

```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 35.174.13.126 PrivateIPs: 172.31.85.119

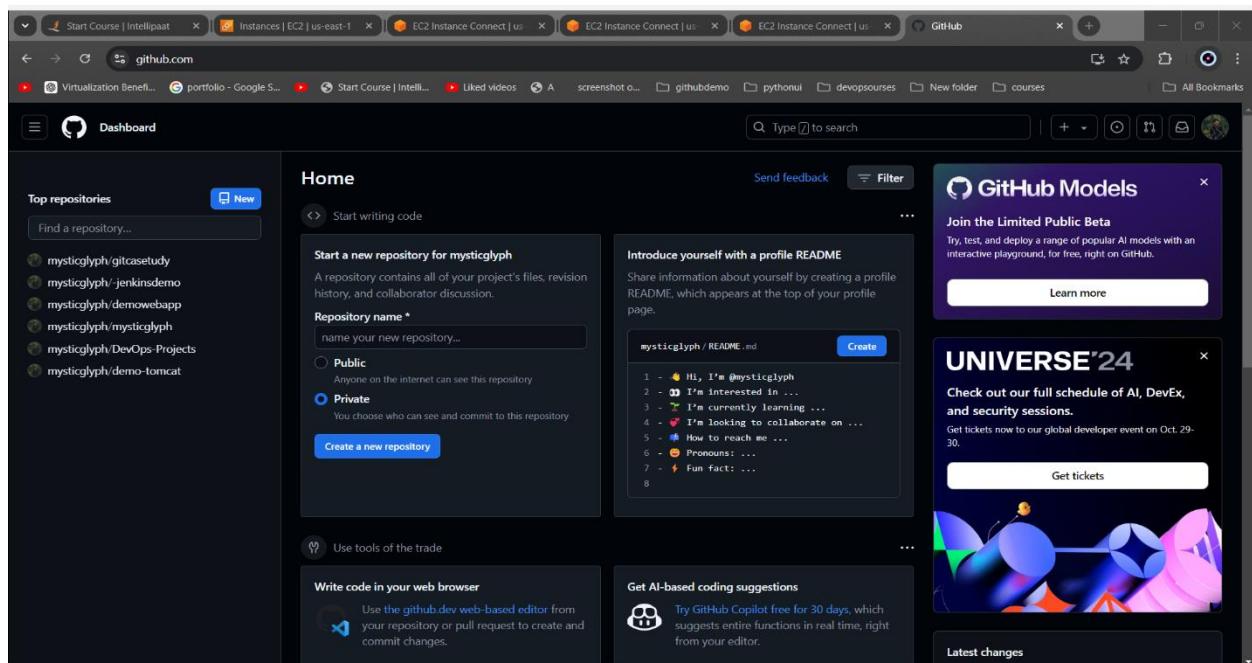
All required packages have been installed using Ansible.

IMPLEMENTING GIT WORKFLOW:

To implement Git flow, we'll first create a GitHub repository, then set up a local repository. We'll initialize Git, create two branches—master and develop—and connect the local repository to the remote one, pushing both branches. This must be done before configuring Jenkins, as GitHub will be integrated with Jenkins. Once done, we'll proceed with configuring Jenkins and setting up its dashboard.

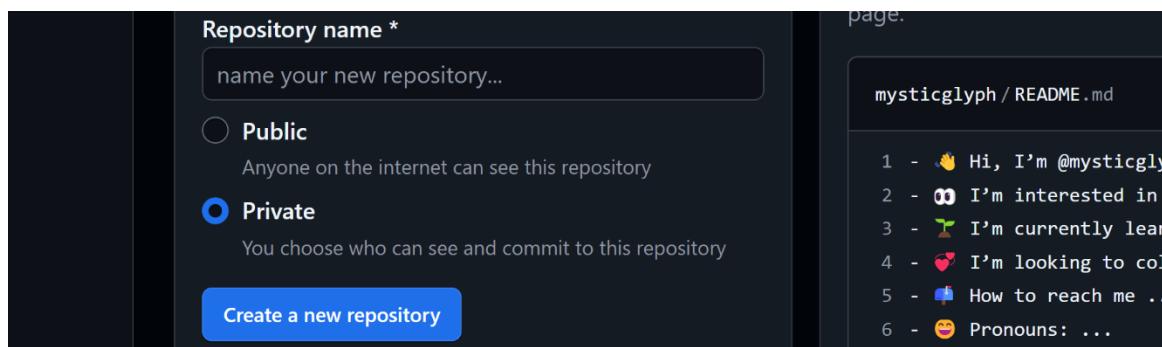
I'll be creating the Git repository in this GitHub account.

→ Github ID: mysticglyph



Let's go ahead and create a repository in this GitHub and name it capstoneproject1.

Click on create new repository:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner * mysticglyph / **Repository name *** Capstoneproject1 Capstoneproject1 is available.

Great repository names are short and memorable. Need inspiration? How about [solid-meme](#) ?

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

[Add a README file](#) This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore [.gitignore template: None](#)

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner * mysticglyph / **Repository name *** Capstoneproject1 Capstoneproject1 is available.

Great repository names are short and memorable. Need inspiration? How about [solid-meme](#) ?

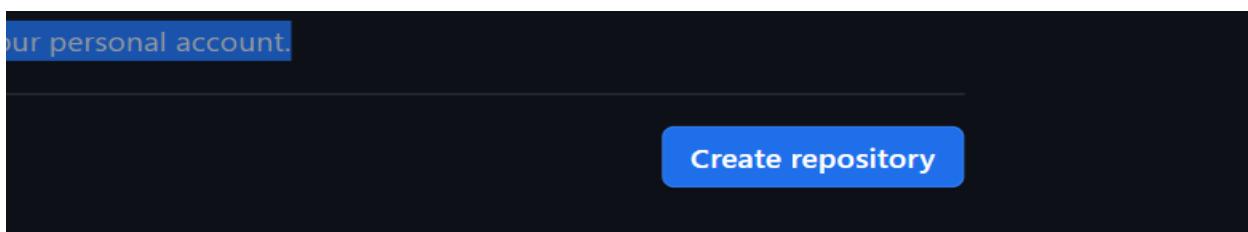
Description (optional)

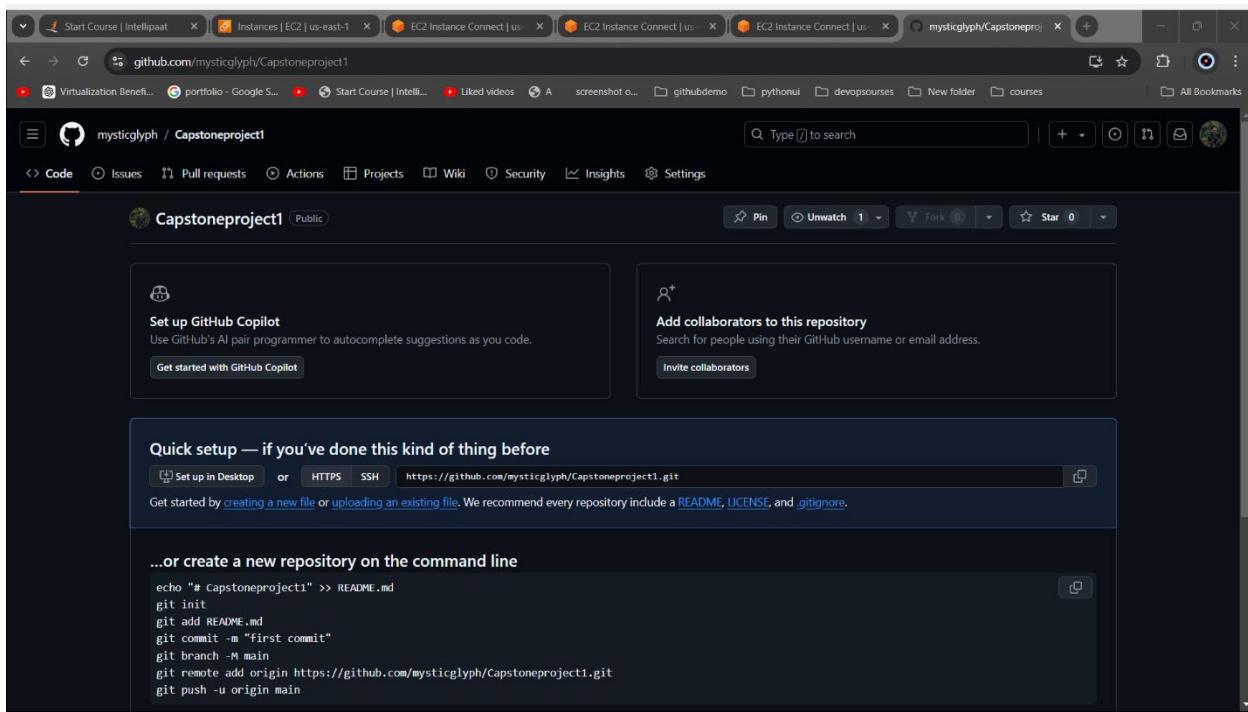
Implementing DevOps Lifecycle

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

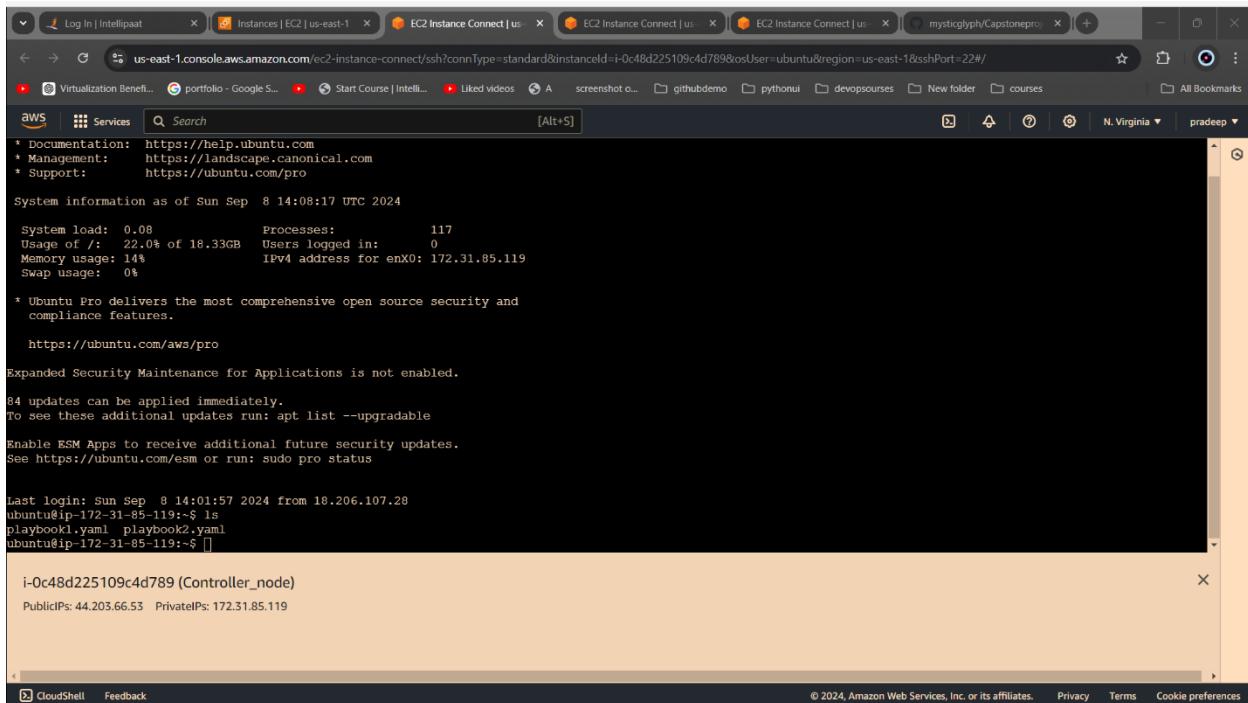
After naming the repository capstoneproject1 and setting it as a public repository, click 'Create Repository'.





Repository is created successfully.

Create a directory named `git_repo` on your local system and initialize Git within it.



Creating a directory:

→ Mkdir gitrepo

```
ubuntu@ip-172-31-85-119:~$ sudo su
Command 'sudo' not found, but there are 15 similar ones.
ubuntu@ip-172-31-85-119:~$ sudo su
Command 'sudo' not found, did you mean:
  command 'sudo' from deb sudo (1:9.14p2-1ubuntu1)
  command 'sudo' from deb sudo-ldap (1:9.14p2-1ubuntu1)
  command 'tudu' from deb tudu (0.10.4-2)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# ls
playbook1.yaml playbook2.yaml
root@ip-172-31-85-119:/home/ubuntu# mkdir gitrepo

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119
```

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
root@ip-172-31-85-119:/home/ubuntu# mkdir gitrepo
root@ip-172-31-85-119:/home/ubuntu# ls
gitrepo playbook1.yaml playbook2.yaml
root@ip-172-31-85-119:/home/ubuntu#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

Succesfully created.

Enter the directory and initialize Git using the git init command.

→ Cd gitrepo

```
root@ip-172-31-85-119:/home/ubuntu# ls
gitrepo playbook1.yaml playbook2.yaml
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
root@ip-172-31-85-119:/home/ubuntu/gitrepo#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

→ Git init

```
root@ip-172-31-85-119:/home/ubuntu# mkdir gitrepo
root@ip-172-31-85-119:/home/ubuntu# ls
gitrepo playbook1.yaml playbook2.yaml
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ubuntu/gitrepo/.git/
root@ip-172-31-85-119:/home/ubuntu/gitrepo#
```

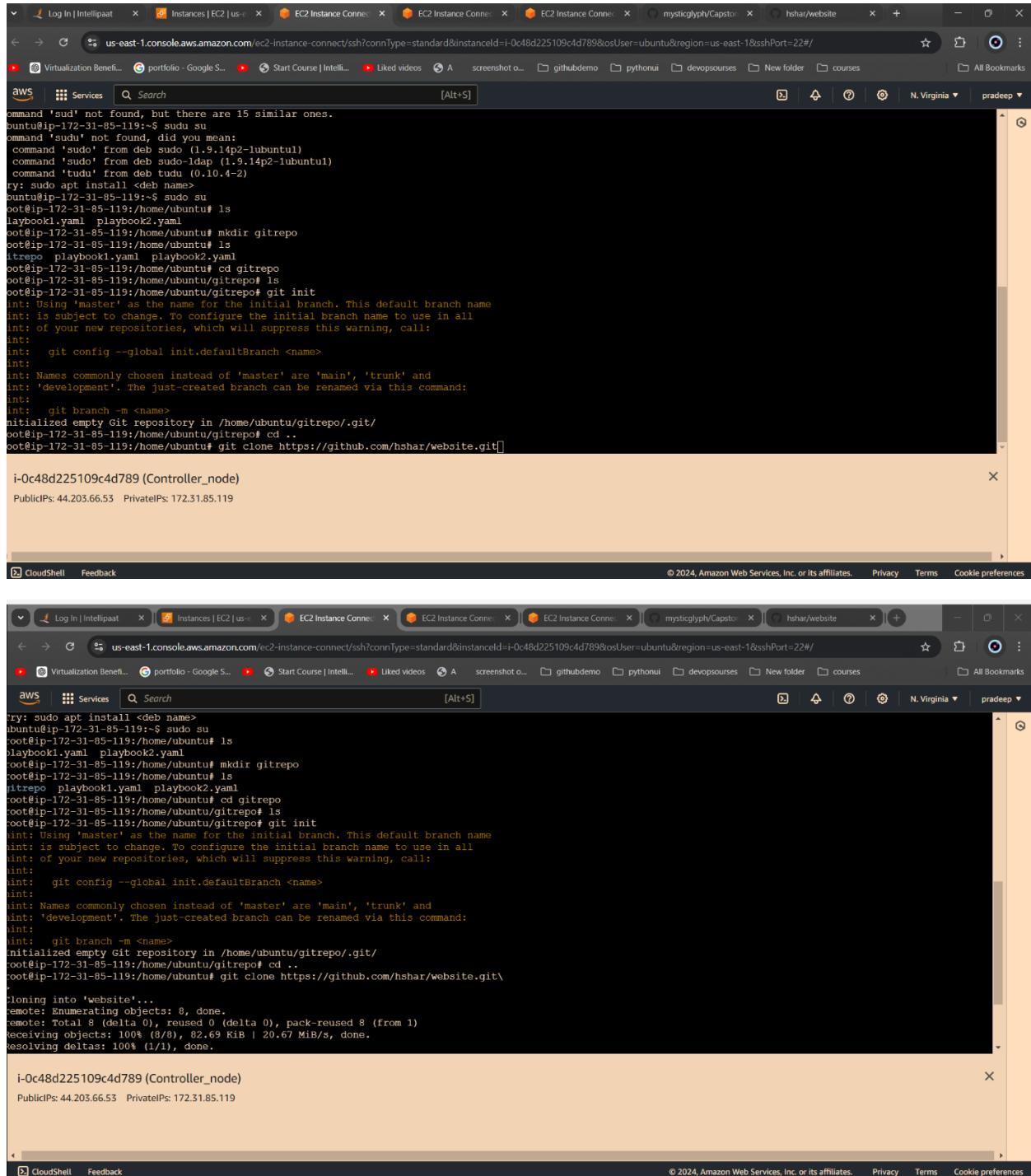
i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

The directory has been successfully initialized with Git.

Now, clone the repository from <https://github.com/hshar/website.git> to the local machine.

→ Git clone <https://github.com/hshar/website.git>



```
command 'sudo' not found, but there are 15 similar ones.
ubuntu@ip-172-31-85-119:~$ sudo su
command 'sudo' not found, did you mean:
 command 'sudo' from deb sudo (1.9.14p2-1ubuntu1)
 command 'sudo' from deb sudo-ldap (1.9.14p2-1ubuntu1)
 command 'tudu' from deb tudu (0.10.4-2)
try: sudo apt install <deb name>
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:~/home/ubuntu# ls
playbook1.yaml playbook2.yaml
root@ip-172-31-85-119:~/home/ubuntu# mkdir gitrepo
root@ip-172-31-85-119:~/home/ubuntu# ls
gitrepo playbook1.yaml playbook2.yaml
root@ip-172-31-85-119:~/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:~/home/ubuntu/gitrepo# ls
root@ip-172-31-85-119:~/home/ubuntu/gitrepo# git init
int: Using 'master' as the name for the initial branch. This default branch name
int: is subject to change. To configure the initial branch name to use in all
int: of your new repositories, which will suppress this warning, call:
int:
int:   git config --global init.defaultBranch <name>
int:
int: Names commonly chosen instead of 'master' are 'main', 'trunk' and
int: 'development'. The just-created branch can be renamed via this command:
int:
int:   git branch -m <name>
initialized empty Git repository in /home/ubuntu/gitrepo/.git/
root@ip-172-31-85-119:~/home/ubuntu/gitrepo# cd ..
root@ip-172-31-85-119:~/home/ubuntu# git clone https://github.com/hshar/website.git
i-0c48d225109c4d789 (Controller_node)
PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```



```
try: sudo apt install <deb name>
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:~/home/ubuntu# ls
playbook1.yaml playbook2.yaml
root@ip-172-31-85-119:~/home/ubuntu# mkdir gitrepo
root@ip-172-31-85-119:~/home/ubuntu# ls
gitrepo playbook1.yaml playbook2.yaml
root@ip-172-31-85-119:~/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:~/home/ubuntu/gitrepo# ls
root@ip-172-31-85-119:~/home/ubuntu/gitrepo# git init
int: Using 'master' as the name for the initial branch. This default branch name
int: is subject to change. To configure the initial branch name to use in all
int: of your new repositories, which will suppress this warning, call:
int:
int:   git config --global init.defaultBranch <name>
int:
int: Names commonly chosen instead of 'master' are 'main', 'trunk' and
int: 'development'. The just-created branch can be renamed via this command:
int:
int:   git branch -m <name>
initialized empty Git repository in /home/ubuntu/gitrepo/.git/
root@ip-172-31-85-119:~/home/ubuntu/gitrepo# cd ..
root@ip-172-31-85-119:~/home/ubuntu# git clone https://github.com/hshar/website.git\

cloning into 'website'...
remote: Enumerating objects: 8, done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 8 (from 1)
receiving objects: 100% (8/8), 82.69 KiB | 20.67 MiB/s, done.
resolving deltas: 100% (1/1), done.

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Cloning the repo is successful.

```
root@ip-172-31-85-119:/home/ubuntu# ls
gitrepo playbook1.yaml playbook2.yaml website
root@ip-172-31-85-119:/home/ubuntu# ls
gitrepo playbook1.yaml playbook2.yaml website
root@ip-172-31-85-119:/home/ubuntu#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

```
root@ip-172-31-85-119:/home/ubuntu# ls
gitrepo playbook1.yaml playbook2.yaml website
root@ip-172-31-85-119:/home/ubuntu# ls
gitrepo playbook1.yaml playbook2.yaml website
root@ip-172-31-85-119:/home/ubuntu# cd website
root@ip-172-31-85-119:/home/ubuntu/website# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/website#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

After cloning, two files are visible in the repository.

First Ensure you are in the correct directory [/home/ubuntu/website] and then Copy the contents to the gitrepo directory:

→ cp -r * /home/ubuntu/gitrepo/

```
root@ip-172-31-85-119:/home/ubuntu# cd website
root@ip-172-31-85-119:/home/ubuntu/website# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/website# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/website#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

```
See https://ubuntu.com/csm or run! sudo apt status

Last login: Sun Sep  9 14:09:18 2024 from 18.206.107.27
ubuntu@ip-172-31-85-119:~$ ls
gitrepo playbook1.yaml playbook2.yaml website
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# cd website
root@ip-172-31-85-119:/home/ubuntu/website# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/website# cp -r * /home/ubuntu/gitrepo/
root@ip-172-31-85-119:/home/ubuntu/website#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

Successfully copied to gitrepo.

```
ubuntu@ip-172-31-85-119:~$ ls
gitrepo playbook1.yaml playbook2.yaml website
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# cd website
root@ip-172-31-85-119:/home/ubuntu/website# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/website# cp -r * /home/ubuntu/gitrepo/
root@ip-172-31-85-119:/home/ubuntu/website# cd ..
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

Now, we will link the local repository to the remote repository.

➔ Git remote add [Capstoneproject1 https://github.com/mysticglyph/Capstoneproject1.git](https://github.com/mysticglyph/Capstoneproject1.git)

First, navigate to the git_repo directory in the local repository, and then connect it to the remote repository.

```
gitrepo playbook1.yaml playbook2.yaml website
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# cd website
root@ip-172-31-85-119:/home/ubuntu/website# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/website# cp -r * /home/ubuntu/gitrepo/
root@ip-172-31-85-119:/home/ubuntu/website# cd ..
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

We inside gitrepo directory.

```
root@ip-172-31-85-119:/home/ubuntu/website# cp -r * /home/ubuntu/gitrepo/
root@ip-172-31-85-119:/home/ubuntu/website# cd ..
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git remote add Capstoneproject1 https://github.com/mysticglyph/Capstoneproject1.git
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

```
root@ip-172-31-85-119:/home/ubuntu/website# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/website# cp -r * /home/ubuntu/gitrepo/
root@ip-172-31-85-119:/home/ubuntu/website# cd ..
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git remote add Capstoneproject1 https://github.com/mysticglyph/Capstoneproject1.git
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git remote -v
Capstoneproject1      https://github.com/mysticglyph/Capstoneproject1.git (fetch)
Capstoneproject1      https://github.com/mysticglyph/Capstoneproject1.git (push)
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

The local repository has been successfully linked to the remote repository.

To push the files to the remote repository:

1. Add the files to the staging area.

→ Git add .

```
root@ip-172-31-85-119:/home/ubuntu/website# cd ..
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git add .
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

```
root@ip-172-31-85-119:/home/ubuntu/website# cd ..
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git add .
root@ip-172-31-85-119:/home/ubuntu/gitrepo#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

2. Commit the changes.

→ Git commit -m"files are added"

```
root@ip-172-31-85-119:/home/ubuntu/website# cd ..
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git add .
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git commit -m"files are added"
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

```
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git add .
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git commit -m"files are added"
[master (root-commit) dc5c9e2] files are added
 committer: root <root@ip-172-31-85-119.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

 git config --global --edit

After doing this, you may fix the identity used for this commit with:

 git commit --amend --reset-author

2 files changed, 8 insertions(+)
 create mode 100644 images/github3.jpg
 create mode 100644 index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

3. Push the changes to GitHub.

→ git push origin master

```
System load: 0.08      Processes: 117
Usage of /: 22.0% of 18.33GB   Users logged in: 0
Memory usage: 13%          IPv4 address for enx0: 172.31.85.119
Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

xpanded Security Maintenance for Applications is not enabled.

4 updates can be applied immediately.
o see these additional updates run: apt list --upgradable

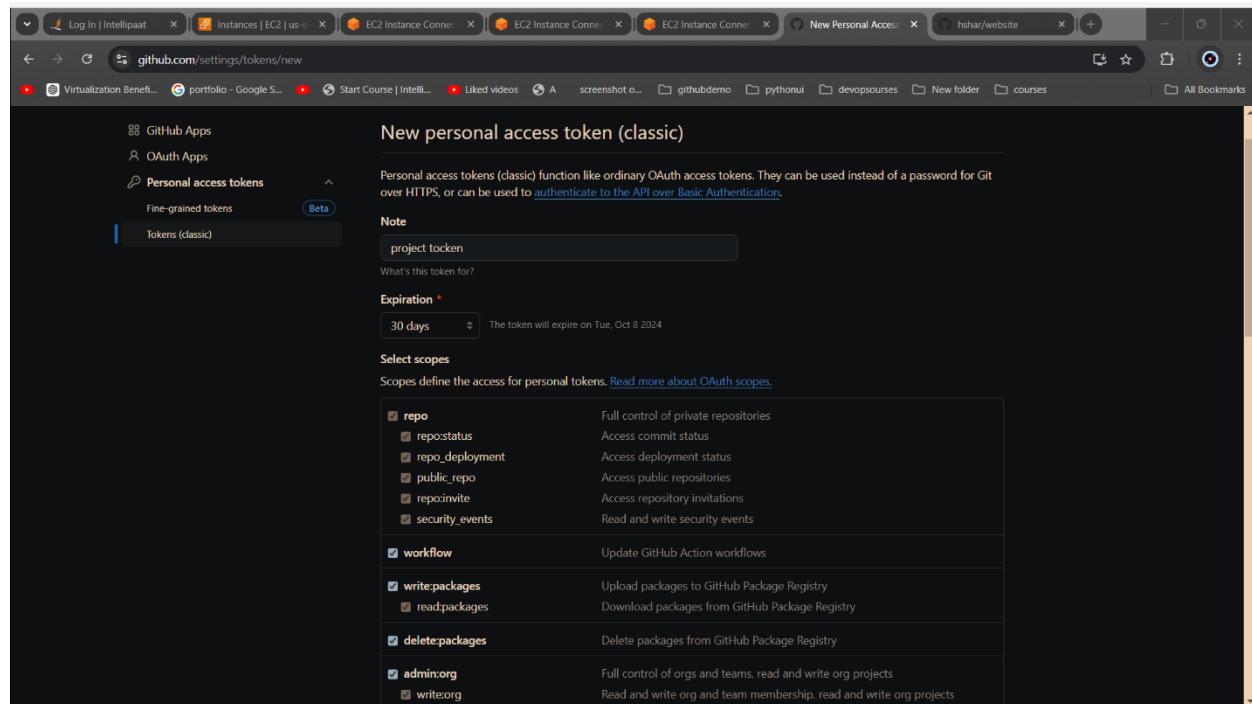
nable ESM Apps to receive additional future security updates.
ee https://ubuntu.com/esm or run: sudo pro status

ast login: Sun Sep  8 15:39:17 2024 from 18.206.107.28
untu@ip-172-31-85-119:~$ ls
itrepo playbook1.yaml playbook2.yaml website
untu@ip-172-31-85-119:~$ sudo su
ot@ip-172-31-85-119:/home/ubuntu# cd gitrepo
ot@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
ot@ip-172-31-85-119:/home/ubuntu/gitrepo# git push origin master

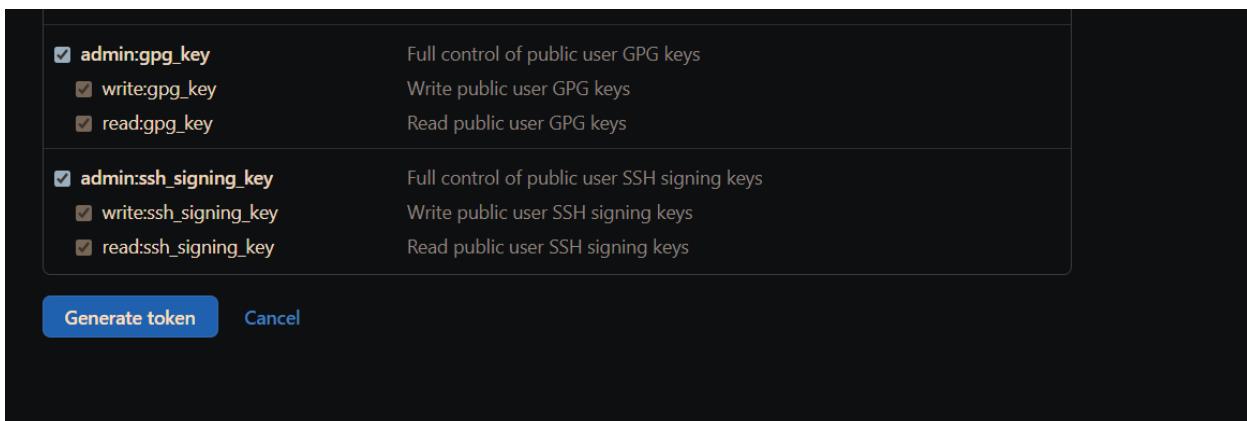
i-0c48d225109c4d789 (Controller_node)
Public IPs: 44.203.66.53 Private IPs: 172.31.85.119
```

Next, generate a token to authorize pushing the files to the repository.

Settings->developers setting->personal access tokens->token[classic]->generate new classic token->classic token.



Create the token with the name project_token and proceed to generate it.



The screenshot shows the GitHub 'Personal access tokens (classic)' list. The sidebar on the left includes 'Hub Apps', 'Auth Apps', and 'Personal access tokens' (with 'Granular tokens' and 'Tokens (classic)' options). The main area displays two tokens:

- casestudy** — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:packages
Last used within the last 3 weeks
⚠️ This token has no expiration date.
Delete
- sample2** — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:packages
Last used within the last 4 weeks
Delete

A note at the top of the list says: 'Selected are included in other scopes. Only the minimum set of necessary scopes has been saved.'

```
ubuntu@ip-172-31-85-119:~$ ls
gitrepo playbook1.yaml playbook2.yaml website
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push origin master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git remote -v
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (fetch)
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (push)
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push origin master
fatal: 'master' does not appear to be a git repository
fatal: could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push Capstoneproject1 master
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com': []

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119
```

```
gitrepo playbook1.yaml playbook2.yaml website
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# cd gitrepo
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push origin master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git remote -v
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (fetch)
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (push)
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push origin master
fatal: 'master' does not appear to be a git repository
fatal: could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push Capstoneproject1 master
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com': []

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119
```

The screenshot shows a CloudShell terminal window with several tabs open at the top. The active tab displays a terminal session on an EC2 instance. The user has run a series of commands to clone a GitHub repository, switch branches, and push changes. The output shows the repository URL, the branch being pushed to ('master'), and the successful completion of the push operation.

```
Please make sure you have the correct access rights  
and the repository exists.  
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git remote -v  
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (fetch)  
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (push)  
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push origin master  
fatal: 'origin' does not appear to be a git repository  
fatal: Could not read from remote repository.  
  
Please make sure you have the correct access rights  
and the repository exists.  
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push Capstoneproject1 master  
Username for 'https://github.com': mysticglyph  
Password for 'https://mysticglyph@github.com':  
Enumerating objects: 5, done.  
Counting objects: 100% (5/5), done.  
Delta compression using up to 2 threads  
Compressing objects: 100% (4/4), done.  
Writing objects: 100% (5/5), 82.38 KiB | 16.48 MiB/s, done.  
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/mysticglyph/capstoneproject1.git  
 * [new branch]      master -> master  
root@ip-172-31-85-119:/home/ubuntu/gitrepo# []  
  
i-0c48d225109c4d789 (Controller_node)  
PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119
```

The files have been successfully pushed to the master branch.

Now, we will create a new branch named develop.

- Command for creating a branch
- Git branch branch_name

The screenshot shows a CloudShell terminal window. The user has navigated to their local repository directory and run the command to create a new branch named 'develop'. The terminal output shows the creation of the branch and the current status of the repository.

```
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push Capstoneproject1 master  
Username for 'https://github.com': mysticglyph  
Password for 'https://mysticglyph@github.com':  
Enumerating objects: 5, done.  
Counting objects: 100% (5/5), done.  
Delta compression using up to 2 threads  
Compressing objects: 100% (4/4), done.  
Writing objects: 100% (5/5), 82.38 KiB | 16.48 MiB/s, done.  
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/mysticglyph/Capstoneproject1.git  
 * [new branch]      master -> master  
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls  
images index.html  
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git branch develop
```

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

The screenshot shows a CloudShell terminal window. The user has run the command to create a new branch named 'develop'. The terminal output shows the creation of the branch and the current status of the repository.

```
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git branch develop
```

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

To switch branch

> Git checkout branch name

```
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git branch develop
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git checkout develop
Switched to branch 'develop'
root@ip-172-31-85-119:/home/ubuntu/gitrepo#
```

i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119

Switched to develop branch

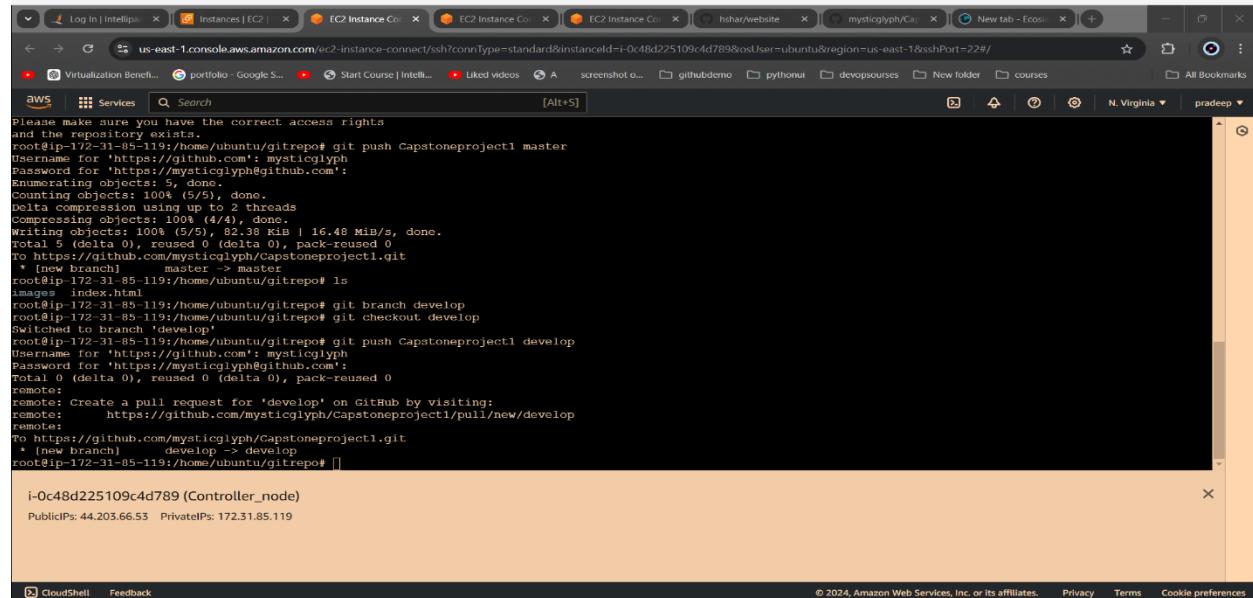
Push the branch to the GitHub.

➔ Git push origin develop

```
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git branch develop
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git checkout develop
Switched to branch 'develop'
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push Capstoneproject1 develop
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com':
```

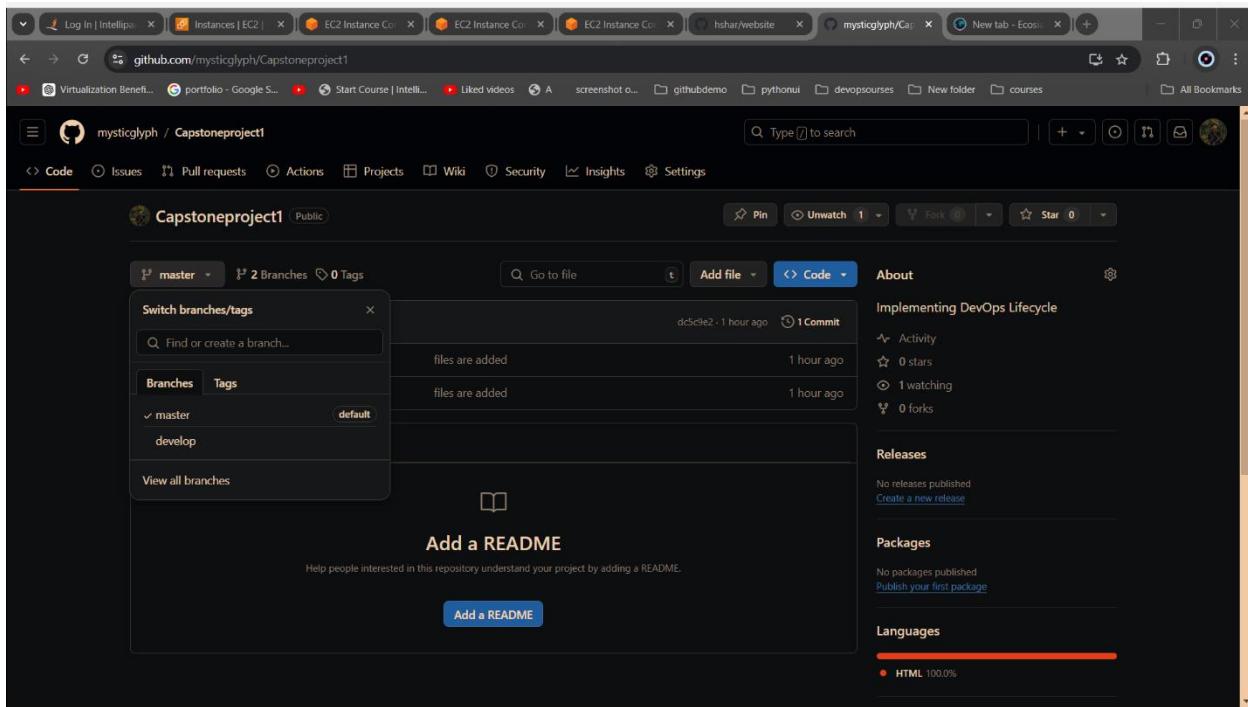
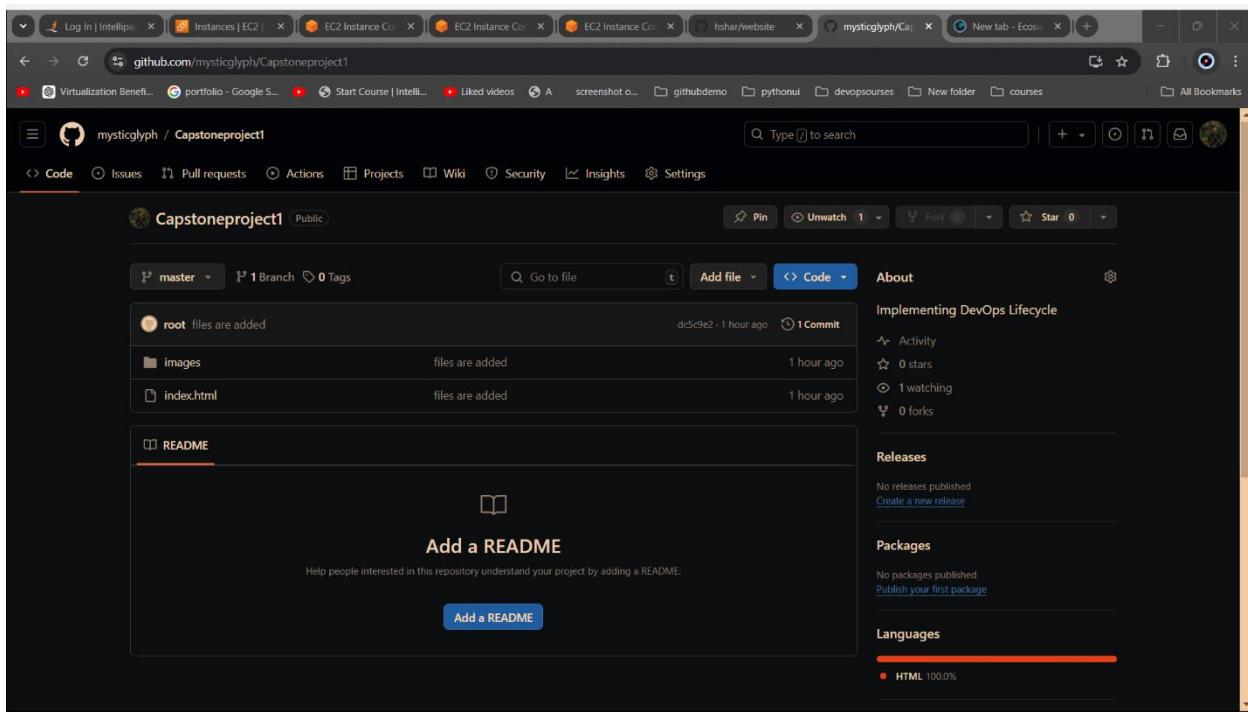
i-0c48d225109c4d789 (Controller_node)

PublicIPs: 44.203.66.53 PrivateIPs: 172.31.85.119



```
Please make sure you have the correct access rights
and the repository exists.
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push Capstoneproject1 master
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 16.48 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mysticglyph/capstoneproject1.git
 * [new branch]      master -> master
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
images index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git branch develop
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git checkout develop
Switched to branch 'develop'
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push Capstoneproject1 develop
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/mysticglyph/capstoneproject1/pull/new/develop
remote:
To https://github.com/mysticglyph/capstoneproject1.git
 * [new branch]      develop -> develop
root@ip-172-31-85-119:/home/ubuntu/gitrepo#
```

Branch is added.



The repository on GitHub is now successfully implemented with both master and develop branches.

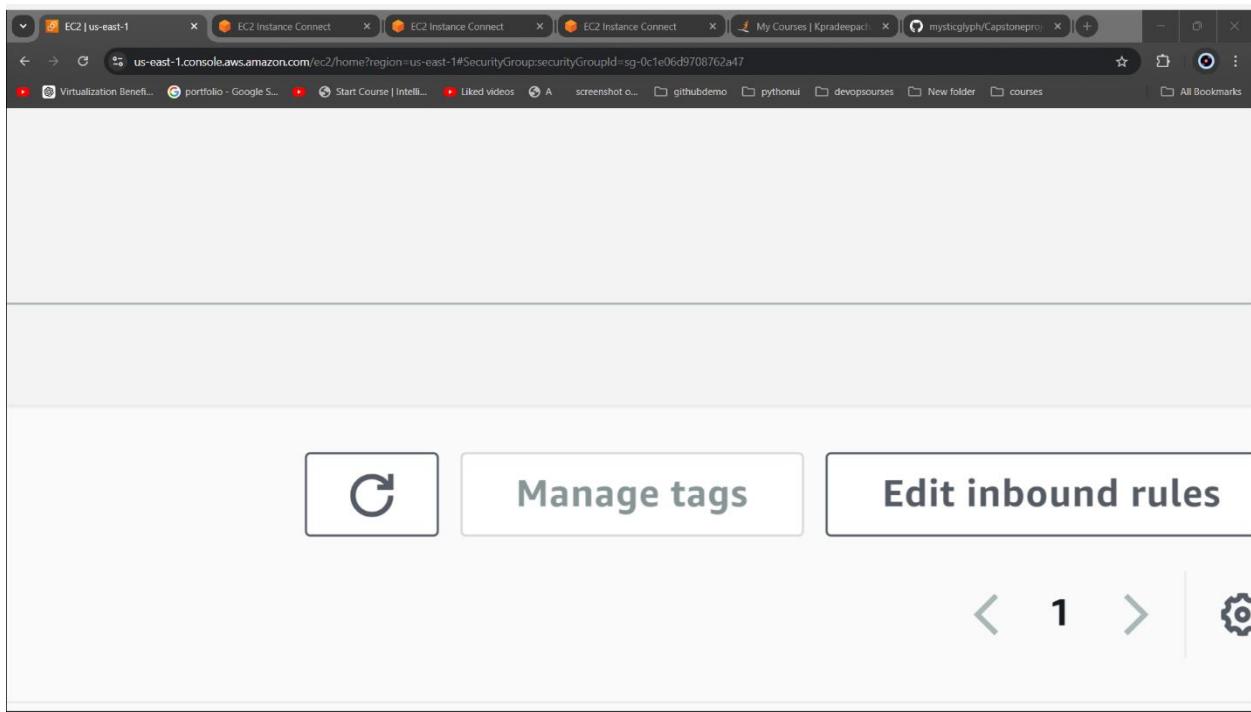
Now, let's proceed with the Jenkins setup:

First, open port 8080 in the security group. Then, access the Jenkins user interface by navigating to http://<ip_address>:8080 using the instance's IP address.

The screenshot shows the AWS EC2 Instances page. On the left, there is a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. The main area displays a table of instances. One instance, 'Controller_node' (i-0c48d225109c4d789), is selected and highlighted with a blue border. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. Below the table, a detailed view for 'Controller_node' is shown. It includes sections for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details section, it shows Instance ID (i-0c48d225109c4d789), Public IPv4 address (44.211.204.202), Instance state (Running), and Private IP4 addresses (172.31.85.119). Other details like IPv6 address, Hostname type, and Private IP DNS name are also listed. At the bottom right of the page, there are links for CloudShell, Feedback, and a copyright notice for 2024, Amazon Web Services, Inc. or its affiliates.

Configuring the security group of the controller node to allow traffic on port 8080.

The screenshot shows the AWS Security Groups page. The left sidebar has the same navigation as the previous EC2 page. The main area shows a security group named 'sg-0c1e06d9708762a47 - launch-wizard-18'. It has a 'Details' section with fields for Security group name (launch-wizard-18), Security group ID (sg-0c1e06d9708762a47), Description (launch-wizard-18 created 2024-08-31T10:48:20.943Z), Owner (626130759947), VPC ID (vpc-0ad7d342053e758cb), and two rule counts (Inbound rules: 1, Outbound rules: 1). Below this is an 'Inbound rules' section with one entry: sgr-0f2cc62fa8396241e, IPv4, SSH, TCP, Port range 22. The bottom of the page includes CloudShell, Feedback, and standard footer links.



Goto edits inbound rules.

A screenshot of the AWS EC2 'Edit inbound rules' wizard, step 18. The URL in the address bar is 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-0c1e06d9708762a47'. The page shows the 'Edit inbound rules' section with one rule listed. The rule details are: Security group rule ID 'sgr-0f2cc62fa8396241e', Type 'SSH', Protocol 'TCP', Port range '22', Source 'Custom', and Description 'optional'. Below the table, a warning message states: '⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' At the bottom right, there are 'Cancel', 'Preview changes', and a highlighted 'Save rules' button.

Add port 8080.

The screenshot shows the 'Edit inbound rules' step of the 'ModifyInboundSecurityGroupRules' wizard. It lists two rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0f2cc62fa8396241e	SSH	TCP	22	Custom	0.0.0.0/0
-	Custom TCP	TCP	8080	Anywh...	0.0.0.0/0

A warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." There are 'Cancel', 'Preview changes', and 'Save rules' buttons at the bottom.

After saving the rules, open Chrome and attempt to access Jenkins.

The screenshot shows the 'Details' page for the security group 'sg-0c1e06d9708762a47'. The 'Inbound rules' tab is selected, displaying the two rules previously configured:

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0f2cc62fa8396241e	IPv4	SSH	TCP	22
-	sgr-07193e6be50ec8cbe	IPv4	Custom TCP	TCP	8080

AWS EC2 Dashboard - Instances (1/3) - Controller_node

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Controller_node	i-0c48d225109c4d789	Running	t2.medium	2/2 checks passed	View alarms	us-east-1a	ec2-44-211-204-202.compute-1.amazonaws.com
Manage_Node1	i-076061495bb4039d5	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-3-85-8-119.ec2.internal
Manage_Node2	i-0cd0f347cb719fc81	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-44-211-204-202.compute-1.amazonaws.com

i-0c48d225109c4d789 (Controller_node)

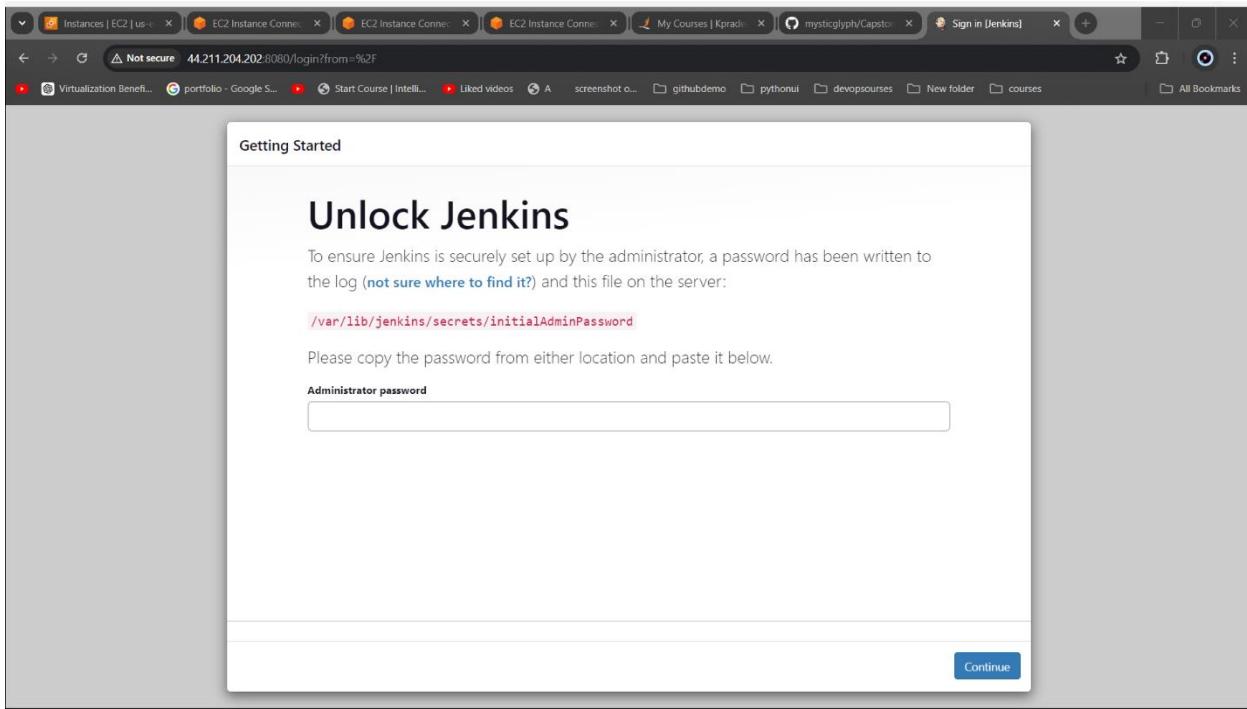
Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0c48d225109c4d789 (Controller_node)	44.211.204.202 open address	172.31.85.119
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-44-211-204-202.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-172-31-85-119.ec2.internal	ip-172-31-85-119.ec2.internal	

Instances | EC2 | us-east-1 | 44.211.204.202:8080

Virtualization Benefits | portfolio - Google Slides | Start Course | IntelliJ IDEA



The Jenkins user interface is now accessible. We need to configure the Jenkins user and install the necessary plugins.

To unlock Jenkins and retrieve the initial admin password, follow these steps:

Find the Initial Admin Password:

- Run the following command to display the contents of the initialAdminPassword file:
- Cat /var/lib/jenkins/secrets/initialAdminPassword

Run the command `sudo cat /var/lib/jenkins/secrets/initialAdminPassword` on the Jenkins server to retrieve the password, and copy it to the user interface for Jenkins setup.

```

Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
Main PID: 526 (java)
Tasks: 44 (limit: 4676)
Memory: 518.0M (peak: 520.1M)
CPU: 20.479s
CGroup: /system.slice/jenkins.service
└─526 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

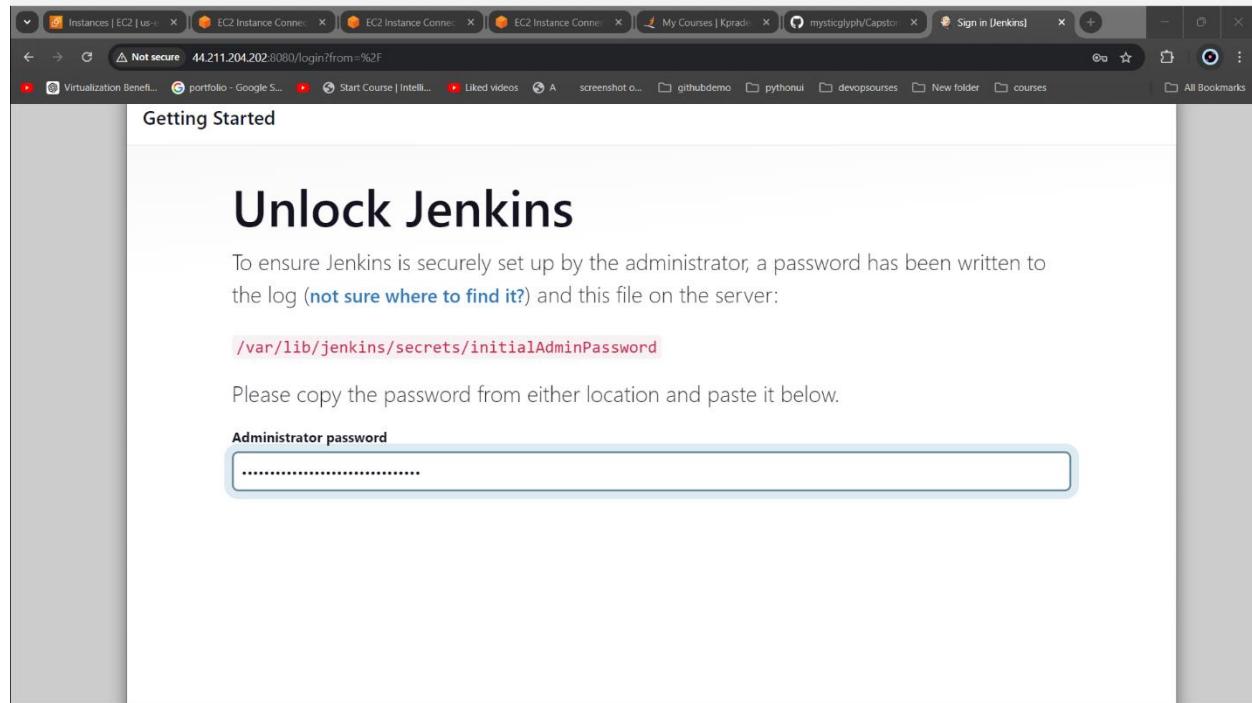
Sep 13 08:32:40 ip-172-31-85-119 jenkins[526]: 4763ca0b06704830b8d0ae0e7f161b73
Sep 13 08:32:40 ip-172-31-85-119 jenkins[526]: ****
Sep 13 08:32:40 ip-172-31-85-119 jenkins[526]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Sep 13 08:32:40 ip-172-31-85-119 jenkins[526]: ****
Sep 13 08:32:40 ip-172-31-85-119 jenkins[526]: ****
Sep 13 08:32:40 ip-172-31-85-119 jenkins[526]: ****
Sep 13 08:32:45 ip-172-31-85-119 jenkins[526]: 2024-09-13 08:32:45.139+0000 [id:37]
Sep 13 08:32:45 ip-172-31-85-119 jenkins[526]: INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
Sep 13 08:32:45 ip-172-31-85-119 jenkins[526]: 2024-09-13 08:32:45.164+0000 [id:31] INFO hudson.lifecycle.Lifecycle$OnReady: Jenkins is fully up and ready
Sep 13 08:32:45 ip-172-31-85-119 jenkins[526]: 2024-09-13 08:32:45.228+0000 [id:56] INFO h.m.DownloadService$DownloadableFileLoad: obtained the updated file
Sep 13 08:32:45 ip-172-31-85-119 jenkins[526]: 2024-09-13 08:32:45.233+0000 [id:56] INFO hudson.util.Retrier$Start: Performed the action check update
-
-
-
-
-
ubuntu@ip-172-31-85-119:~$ i-0c48d225109c4d789 (Controller_node)
PublicIPs: 44.211.204.202 PrivateIPs: 172.31.85.119

```

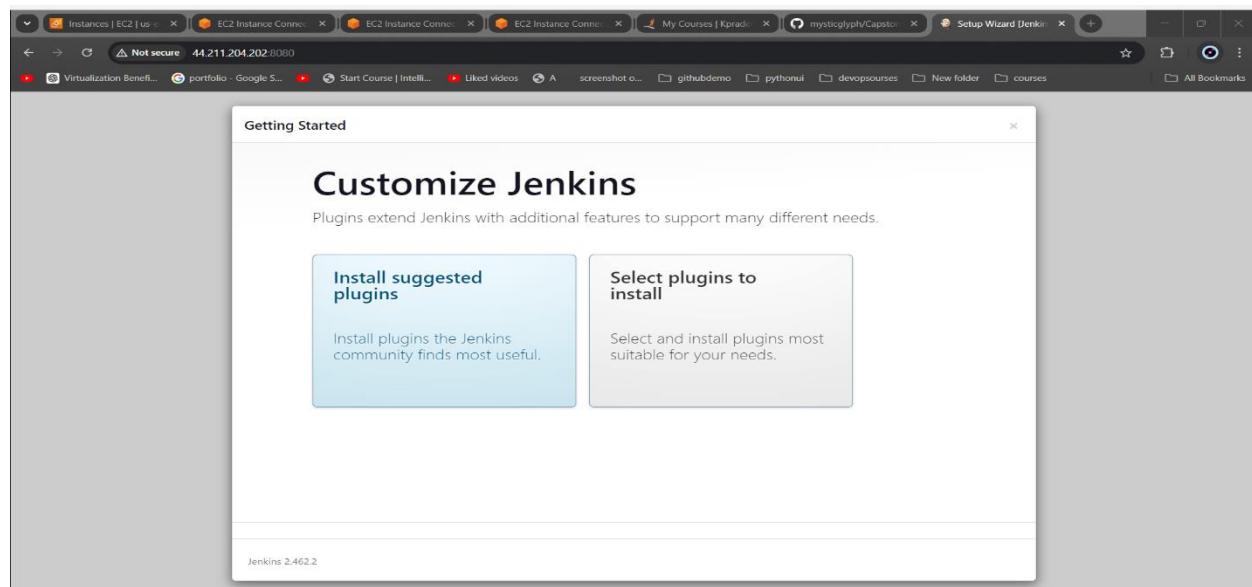
```
ubuntu@ip-172-31-85-119:~$ sudo su
root@ip-172-31-85-119:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
4763ca0b06704830b8d0ae0e7f161b73
root@ip-172-31-85-119:/home/ubuntu#
```

```
i-0c48d225109c4d789 (Controller_node)
PublicIPs: 44.211.204.202 PrivateIPs: 172.31.85.119
```

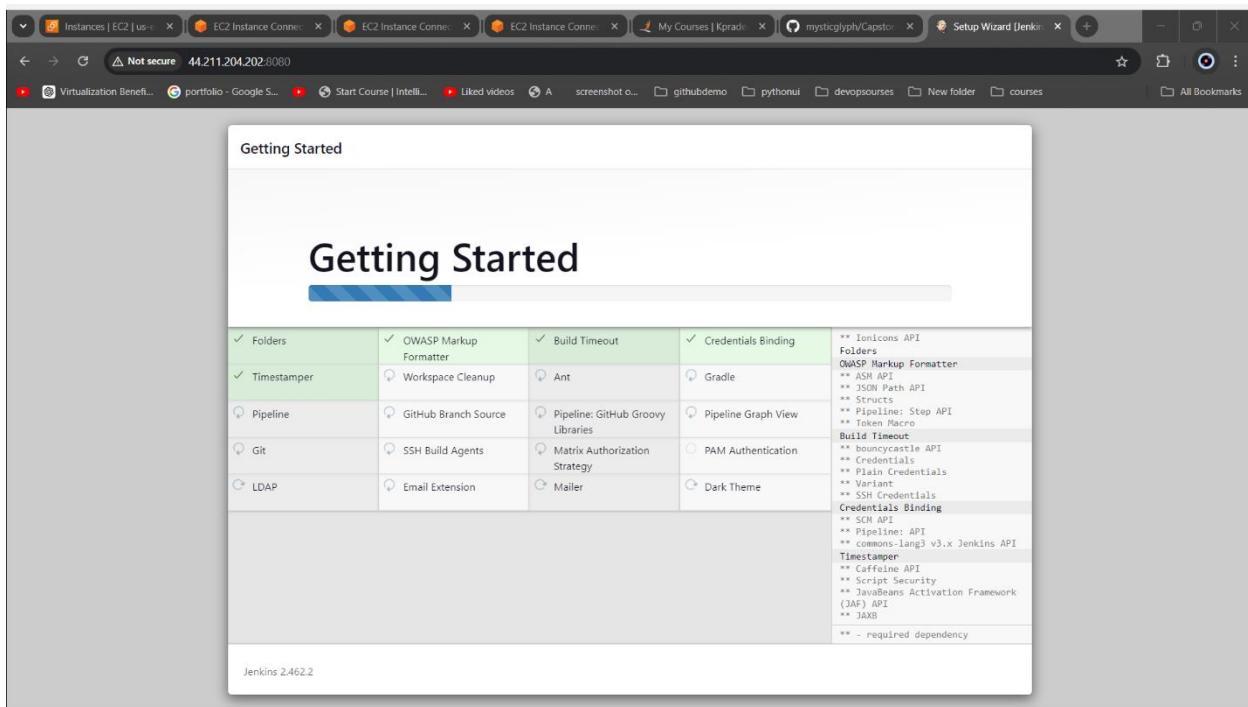
Paste the copied password from the local server into the Jenkins user interface.



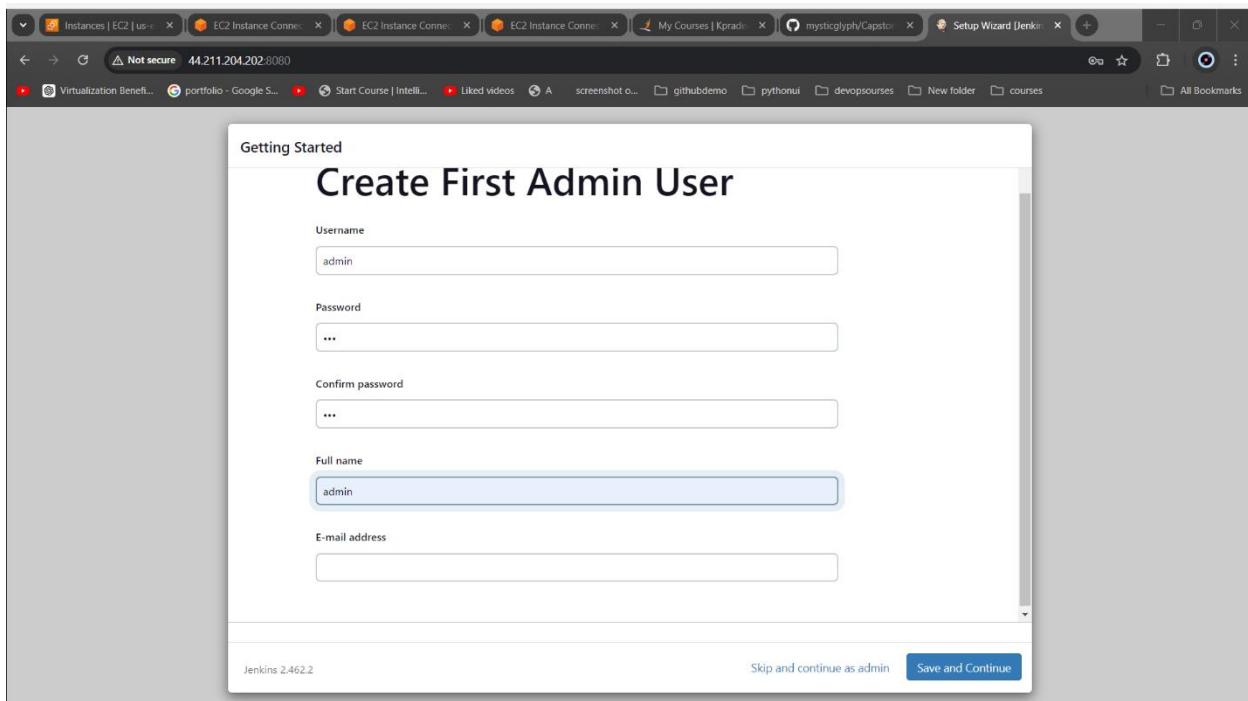
Click on continue.



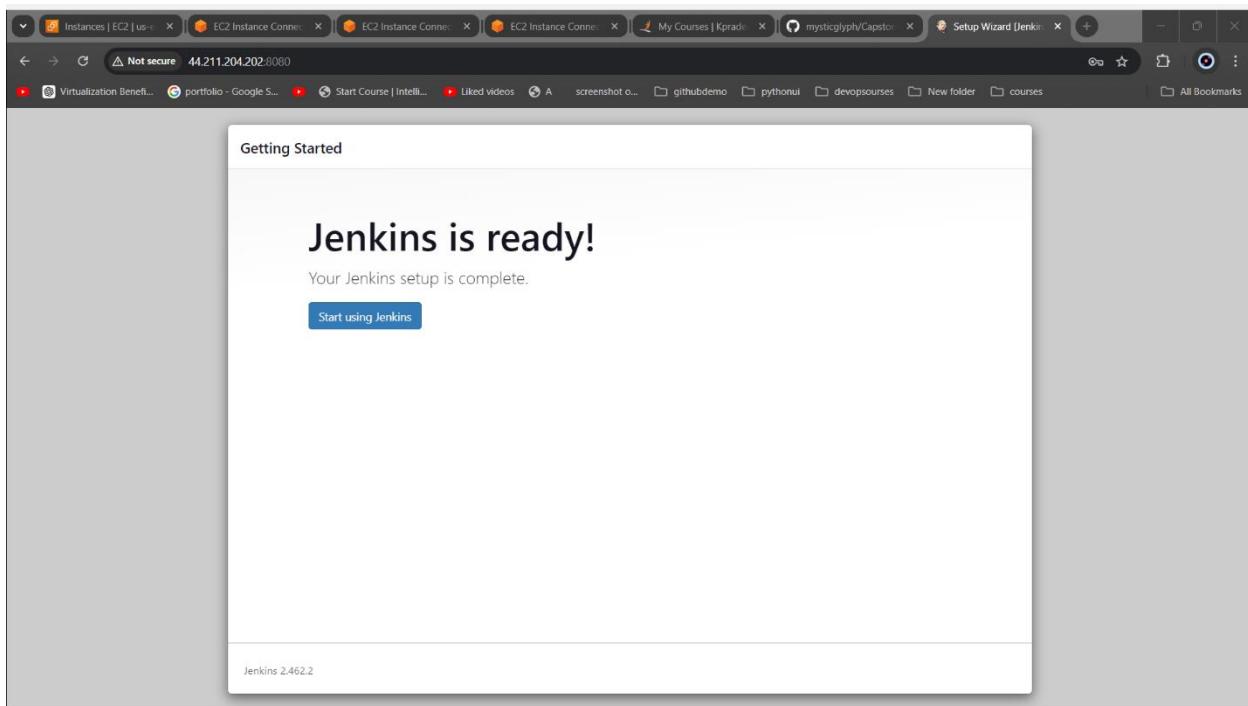
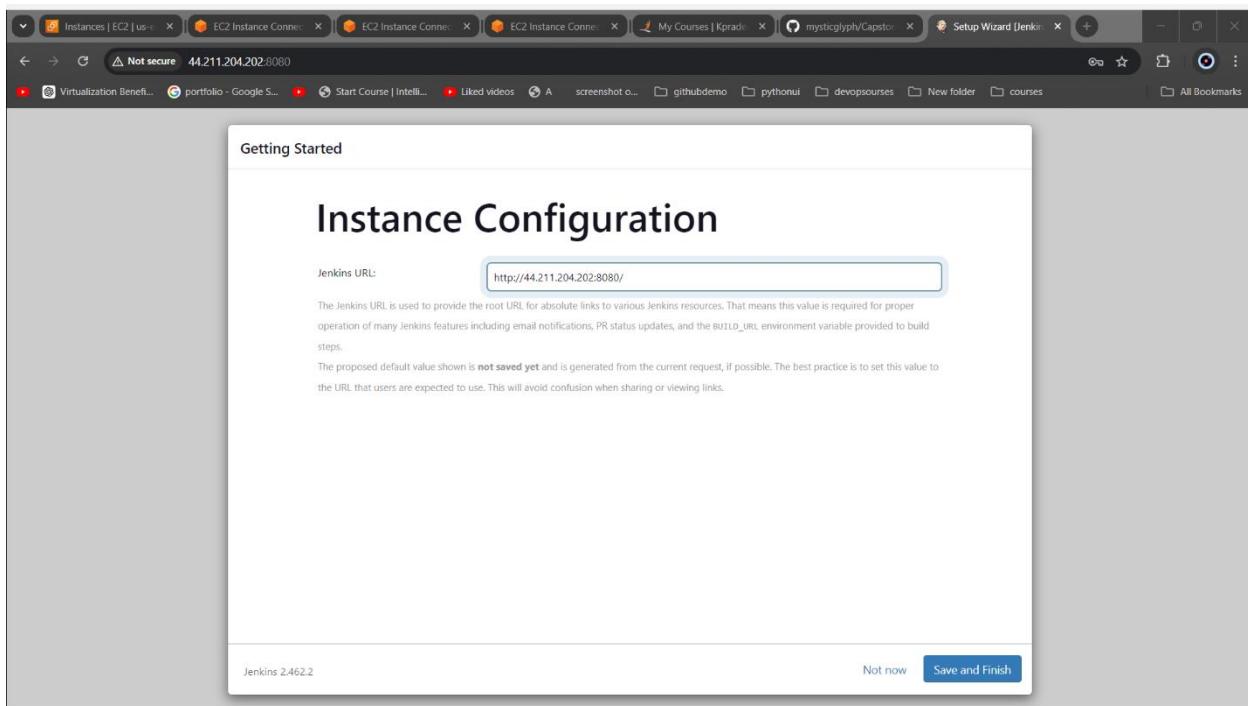
We need to install suggested plugins:



Enter the required user details to set up the admin account



Save and continue.



Click on start using jenkins.

The screenshot shows the Jenkins dashboard at the URL <http://44.211.204.202:8080>. The page title is "Jenkins". On the left, there's a sidebar with links for "New item", "Build History", "Manage Jenkins", and "My Views". The main content area has a heading "Welcome to Jenkins!" with a sub-instruction: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." Below this, there are sections for "Start building your software project" (with "Create a job" and a plus sign button), "Set up a distributed build" (with "Set up an agent" and a computer icon, and "Configure a cloud" with a cloud icon), and "Learn more about distributed builds" (with a question mark icon). At the bottom right, it says "REST API" and "Jenkins 2.462.2".

The Jenkins interface is set up and ready. We can now start creating jobs and assigning tasks.

Add a Jenkins slave node and configure it to handle specific jobs.

Go to Manage Jenkins -> Manage Nodes and Clouds -> New Node to add nodes to the Jenkins server.

The screenshot shows the "Manage Jenkins" page at the URL <http://44.211.204.202:8080/manage>. The title is "Manage Jenkins". A message at the top says "Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)." There are three buttons: "Set up agent", "Set up cloud", and "Dismiss". The main area is divided into sections: "System Configuration" (with "System", "Nodes", "Tools", "Clouds", and "Plugins" items), "Security" (with "Security", "Users", "Credentials", and "Credential Providers" items), and a search bar at the top right labeled "Search settings".

The screenshot shows the Jenkins 'Nodes' page. At the top, there are tabs for 'Dashboard', 'Manage Jenkins', and 'Nodes'. The main area is titled 'Nodes' and contains a table with one row. The row details a 'Built-in Node' named 'Data obtained', which is a 'Linux (amd64)' machine. It shows 'In sync' status, 14.27 GiB of free disk space, 0 B of free swap space, 14.27 GiB of free temp space, and a response time of 0ms. Below the table, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (2 idle). A legend indicates icons for Slave (S), Master (M), and Label (L). At the bottom right, there are links for 'REST API' and 'Jenkins 2.462.2'.

Click on New Node to add and configure the new slave node.

Configure the first slave node with the name “prod” and connect it to Jenkins.

The screenshot shows the 'New node' configuration page. The URL is '44.211.204.202:8080/manage/computer/new'. The page has a 'Node name' input field containing 'prod'. Under 'Type', the 'Permanent Agent' option is selected. A tooltip explains that this adds a plain, permanent agent to Jenkins. At the bottom is a 'Create' button.

The screenshot shows the Jenkins 'New node' creation interface. At the top, there are several tabs and a search bar. Below the header, the page title is 'New node'. The main form has a 'Node name' field containing 'prod'. Under the 'Type' section, 'Permanent Agent' is selected. A descriptive text explains that this adds a plain, permanent agent to Jenkins. At the bottom of the form is a blue 'Create' button.

Click on create.

The screenshot shows the Jenkins 'createItem' configuration page. It includes fields for 'Name' (set to 'prod'), 'Description' (empty), 'Plain text' (disabled), 'Number of executors' (set to 1), 'Remote root directory' (empty, with a red error message 'Remote directory is mandatory'), and 'Labels' (empty). A 'Save' button is located at the bottom of the form.

Specify the remote root directory, which is the Jenkins home directory on the slave node.

The screenshot shows the Jenkins 'Create Item' form for a new node. The URL is <http://44.211.204.202:8080/manage/computer/createtitem>. The form fields are as follows:

- Description**: A large text area with the placeholder "Plain text" and a "Preview" link.
- Number of executors**: A dropdown menu set to "1".
- Remote root directory**: A text input field containing "/home/ubuntu/jenkins/".
- Labels**: A text input field containing "prod".
- Usage**: A dropdown menu set to "Use this node as much as possible".
- Launch method**: A dropdown menu set to "Launch agent by connecting it to the controller".

The screenshot shows the Jenkins 'Create Item' form for a new node. The URL is <http://44.211.204.202:8080/manage/computer/createtitem>. The form fields are as follows:

- Number of executors**: A dropdown menu set to "1".
- Remote root directory**: A text input field containing "/home/ubuntu/jenkins/".
- Labels**: A text input field containing "prod".
- Usage**: A dropdown menu set to "Use this node as much as possible".
- Launch method**: A dropdown menu set to "Launch agent by connecting it to the controller".

Label will be prod.

Usage:

The screenshot shows the Jenkins 'Create Item' form for a new node configuration. The URL in the browser is `44.211.204.202:8080/manage/computer/createItem`. The form fields are as follows:

- Plain text:** Preview
- Number of executors:** 1
- Remote root directory:** /home/ubuntu/jenkins/
- Labels:** prod
- Usage:** Use this node as much as possible
- Launch method:** Launch agent by connecting it to the controller
- Availability:** (dropdown menu)
- Save** button

Launch method:

The screenshot shows the Jenkins 'Create Item' form for a new node configuration, identical to the previous one except for the 'Launch method' field. The URL in the browser is `44.211.204.202:8080/manage/computer/createItem`. The form fields are as follows:

- Plain text:** Preview
- Number of executors:** 1
- Remote root directory:** /home/ubuntu/jenkins/
- Labels:** prod
- Usage:** Use this node as much as possible
- Launch method:** Launch agents via SSH
- Host:** (dropdown menu)
- Save** button

The screenshot shows the Jenkins configuration interface for managing nodes. The 'Usage' field is set to 'Use this node as much as possible'. The 'Launch method' is set to 'Launch agents via SSH'. Under 'Host', there is a dropdown menu showing 'none'. A red error message says 'The selected credentials cannot be found'. The 'Host Key Verification Strategy' is set to 'Known hosts file Verification Strategy'. The 'Availability' field is set to 'Keep this agent online as much as possible'. At the bottom is a blue 'Save' button.

To connect through SSH, configure the host details such as the IP address and credentials.

Copying private IP:

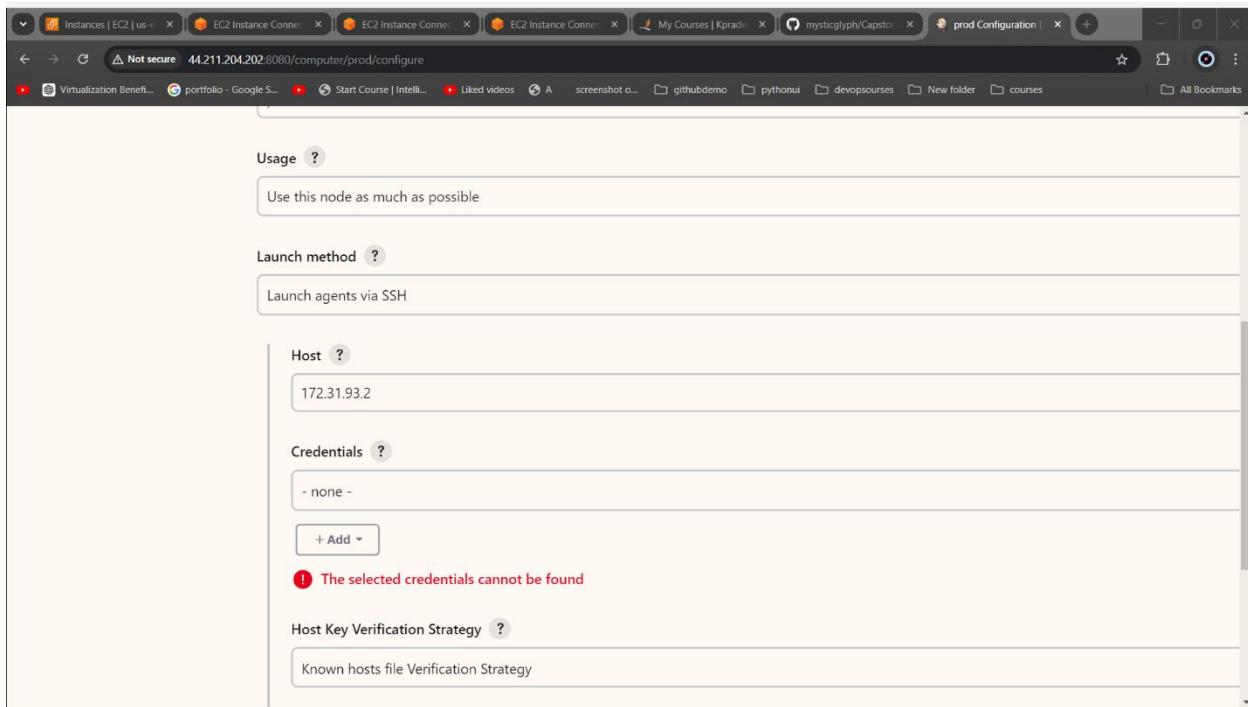
The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations (New), Images (selected), AMIs, and AMI Catalog. The main area shows a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 address
Controller_node	i-0c4bd225109c4d789	Running	t2.medium	2/2 checks passed	View alarms	us-east-1a	ec2-44-21-
Manage_Node1	i-076061495bb4039d5	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-3-85-8-
Manage_Node2	i-0cd0f347cb719fc81	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-44-21-

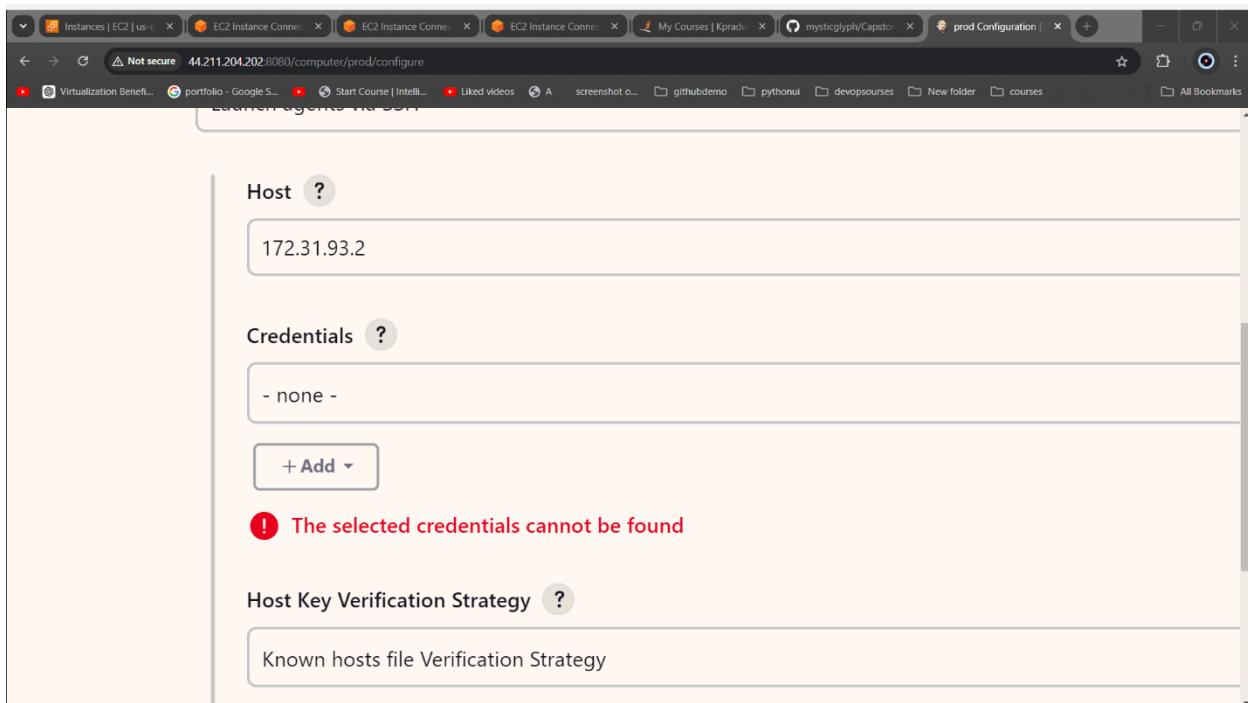
Below the table, the details for 'Manage_Node1' are expanded:

- Details:** Instance ID: i-076061495bb4039d5 (Manage_Node1)
- Instance summary:**
 - Instance ID: i-076061495bb4039d5 (Manage_Node1)
 - Public IPv4 address: 3.85.86.170 [open address]
 - Instance state: Running
 - Private IP DNS name (IPv4 only): ip-172-31-93-2.ec2.internal
- Images:** Private IPv4 addresses: 172.31.93.2
- Elastic Block Store:** Public IPv4 DNS: ec2-3-85-86-170.compute-1.amazonaws.com [open address]

Host:



Credentials:



Add credentials:

Kind:

Ssh and private key:

Jenkins Credentials Provider: Jenkins

Domain: Global credentials (unrestricted)

Kind: SSH Username with private key

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: (empty)

Description: (empty)

Username: (empty)

Jenkins Credentials Provider: Jenkins

Kind: SSH Username with private key

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: prod

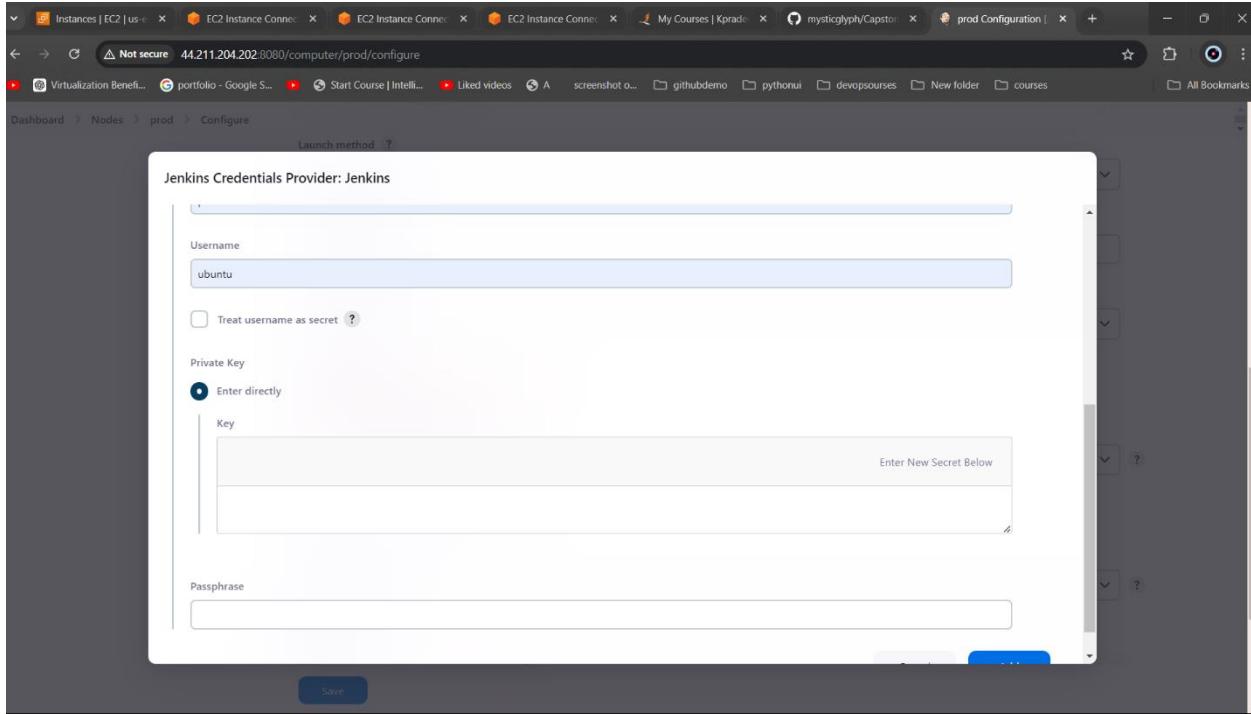
Description: prod

Username: ubuntu

Treat username as secret

Private Key: (empty)

We need to provide the private key:



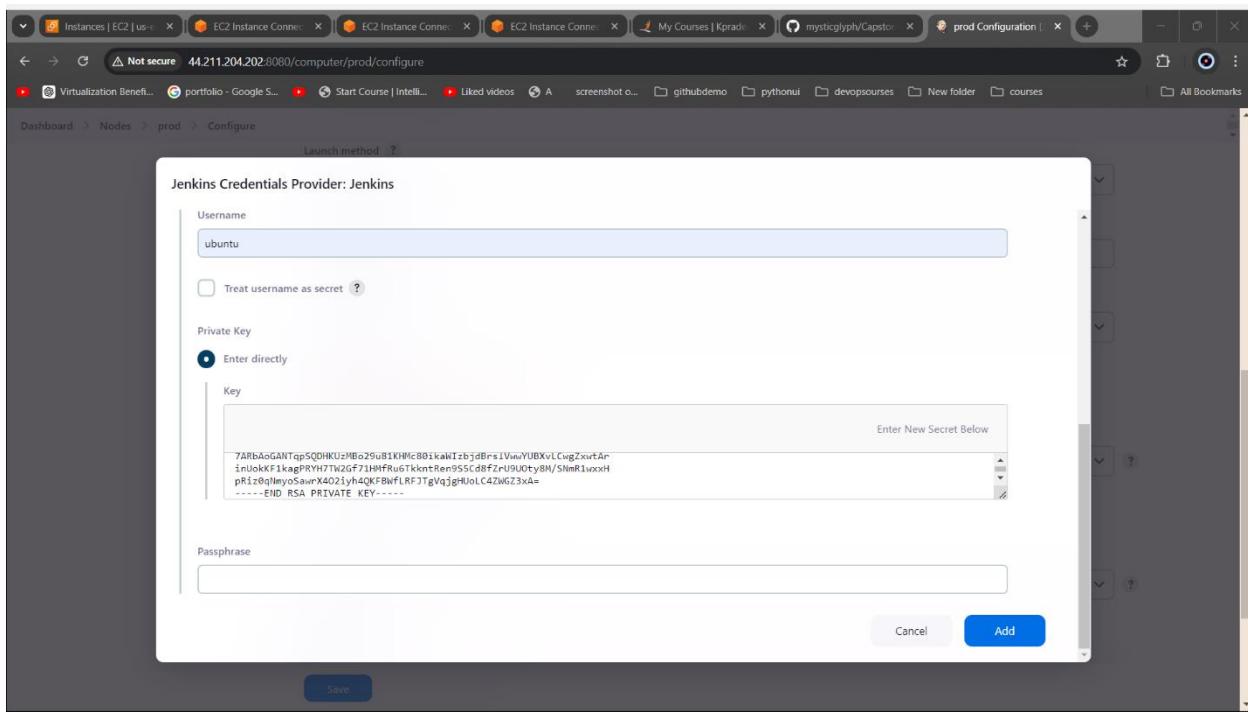
We need to provide the private key,

Paste the content of the key pair from the slave node into the private key section.

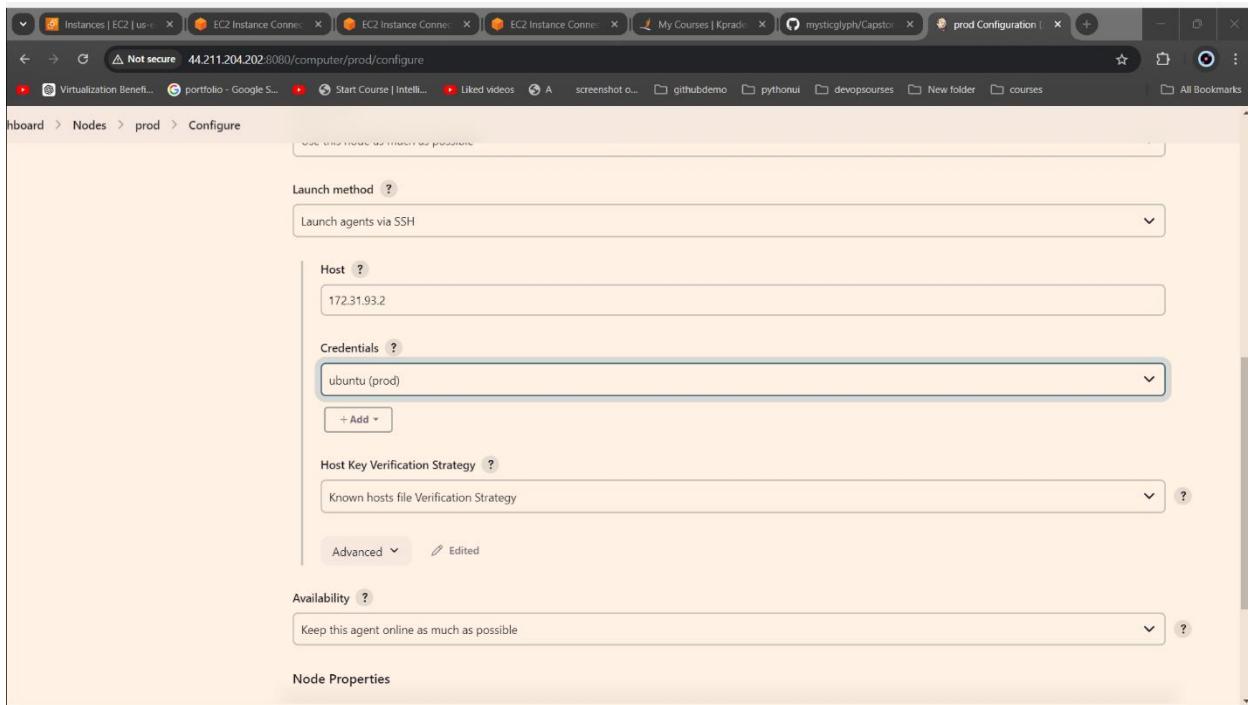
```
--BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEaguCnnawjtFAX59IGCE2bTJAP/+D/pIk39Z0GmzDbCnVBXpo
TtVG77Jv69zhHEH9WZers+iX0oG2ssT1cBLS16bGsN0Ze41mkccCLlvvvDQ72Va
h0ADztpA0BQ1KwGKe6bmjZKmZtlfVA6um7ryvSjsg1cfUgkkubaBEr6Wxp36TJ
EQLZXwf+ougF1Y/iQkwbshFxq5V48LkTGwtKk8+x5f9gbSe1dnwtiPTPBtKyUVR
Of7VF8MOSvkxwzVQ1zM26v9atjEDkoTSzz4ajMX8UR6yrc+giyLp8J9s5r198Uuw
UF84BWzjG5rogxzSngRRcZtmgCCWDmJQrRTeSwIDAQABAoIBAH9I1Nh1NxjgEHOr
215mrZe/Asy+vtbLangkETjVBN+8egJyaNGoAdfq2ovl9zs9qYa4aCkoiwTGHV
wbbsBgvhLnDDGgA4uPluyKues8yY45002qmRFt48ieWNZTzgPjrrfJzi7H4cjjbo
eoNqc2tX7qk6HwPxqlgVNniw0ubnmjmj05U8f6pXDp1dDV29hC6/vjcfwdsOG+H
lGxEii/8+PQsaw/9nOo1lWJsxsPH+/IF2AgGapojCKDXm0bz39hShhhAs2C73V7
I3djejkQ003XjiocnGub4V2vNoUr+7VMWkfqpFQRD014bMhk9bejv09gLZxi7Ym
c1EoPOECgYE4HkWkCtn67yj8pejM6nzK16YmLDrXFvdItXntcJ3rpSAFvh0XktER
sU9t06ev+r05TzL+LuQ8QDGmbv1qz3R/UsZjM+u12v1Y1hj864hbKmzvQjd6BzE
Jxie1y2REAMFz9F7YDP8a9apHI8v45xcr/eAPXFaQF6NECaaGiXT2DECgYEAlUDr
NUjbGwy0cWuwMHfxman9tEmspk+3imdhFeoZrcvLuBo/sriiAZLyoiHe5pu1bz
Yo/Cf1Knkyo8+bogIF3y5thLfdu2ENUMVm/3Rd/2T89kK16o/dr5iaZazlw7qXM
UN79iifu7z9qPothcJ9d4RxvBPs vAPUBHxq+zscGyBmbpriZMofbKKtyugHch6F
GgJ9WHRizIFnRy9F7M6bLxCqY/ iTUk3kjkuadbiiosOWwSmvTvJR17cs1sMQ+r+
CM3QqcybgVdzTxe76zdhshPfsVQNs5xAbnZMFPfzjc8r1sGebsBs6w255MzUN2
gQiiASHgEcVpQ1Q67lwXYQKBgCG2npeLE0Q1sbbf1maR4CWTr+MUTIJHBKukUYCZ
MLE8hg+NLQmDl1+l1f4kXY/ ztm dv cugmXqouBYrLa2SNgmf6I0/RgsM+6vPwJZ
beov0xbt+mh+bi0zxtGorB10npYfKbq18dpw4I91fpG4Fqnb18S2Y71XPqljtnFg
7ARbAoGANtqpsQDHKUzMB029u81KMc80ikawIzbjdBrsv1vwvYUBxvLCwgZxwtAr
inUokKF1kagPRYH7TW2Gf71HMfRu6TkknRen9S5Cd8fZrU9UOty8M/SNmR1wxXH
pRiz0qNmyoSawrx4A02iyh4QKFbfLRFJTgVqjgHUoLc4ZnGZ3xA=
--END RSA PRIVATE KEY-----

```

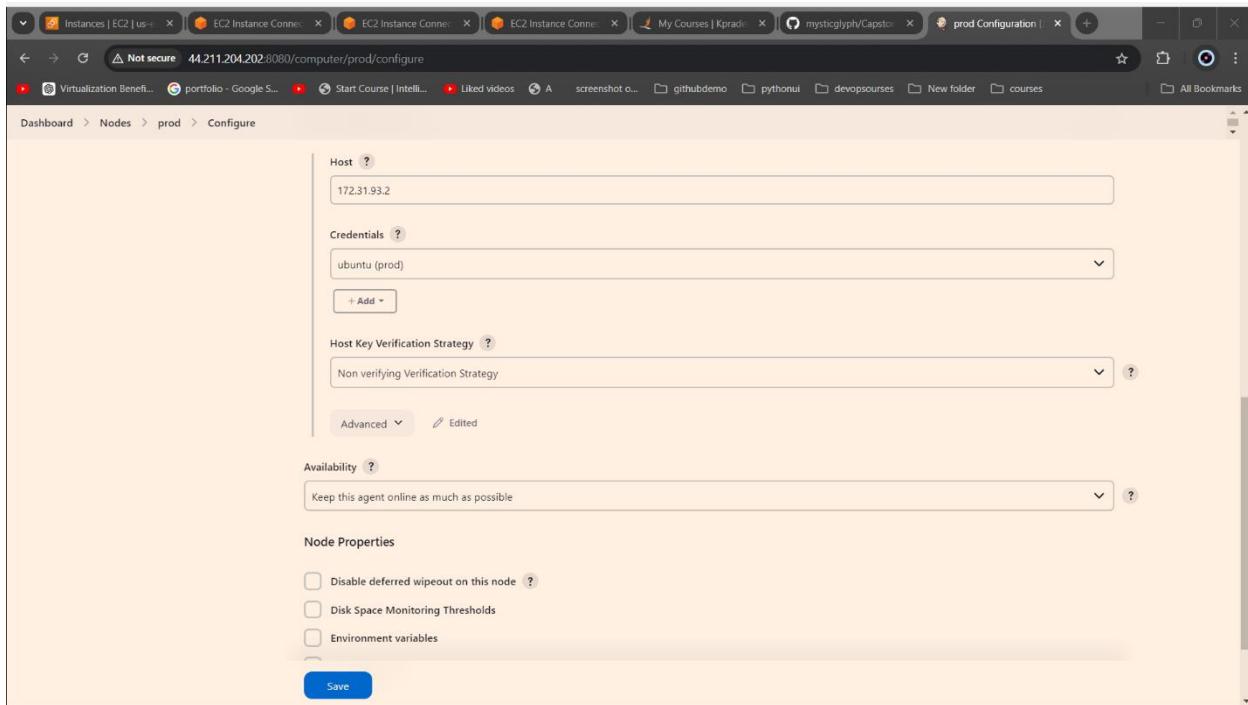
The terminal window title is "SLAVEKEY". The content of the terminal shows a long RSA PRIVATE KEY block, starting with "--BEGIN RSA PRIVATE KEY--" and ending with "--END RSA PRIVATE KEY--". The key itself is a very long string of characters.



After saving the configurations, click Save to apply the changes.



Host Key Verification Strategy:



And click on save:

We need to run this command in slave1:

→ sudo chown ubuntu:ubuntu /home/ubuntu/jenkins/

changes the ownership of the /home/ubuntu/jenkins/ directory to the user ubuntu and the group ubuntu

```
* Ubuntu Pro delivers the most comprehensive open source security and compliance features.
https://ubuntu.com/aws/pro
Expanded Security Maintenance for Applications is not enabled.
// updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Sep 13 09:27:03 2024 from 18.206.107.27
ubuntu@ip-172-31-93-2:~$ sudo su
root@ip-172-31-93-2:/home/ubuntu# cd /home/ubuntu/jenkins/
bash: cd: /home/ubuntu/jenkins/: No such file or directory
root@ip-172-31-93-2:/home/ubuntu# cd /home/ubuntu/
root@ip-172-31-93-2:/home/ubuntu# ls
root@ip-172-31-93-2:/home/ubuntu# ls
root@ip-172-31-93-2:/home/ubuntu# ls
root@ip-172-31-93-2:/home/ubuntu# mkdir jenkins
root@ip-172-31-93-2:/home/ubuntu# ls
jenkins
root@ip-172-31-93-2:/home/ubuntu# cd jenkins
root@ip-172-31-93-2:/home/ubuntu/jenkins# cd ..
root@ip-172-31-93-2:/home/ubuntu# chown ubuntu:ubuntu /home/ubuntu/jenkins/[]

i-076061495bb4039d5 (Manage_Node1)
PublicIPs: 3.85.86.170 PrivateIPs: 172.31.93.2
```

```

root@ip-172-31-93-2:/home/ubuntu# cd jenkins
root@ip-172-31-93-2:/home/ubuntu/jenkins# cd ..
root@ip-172-31-93-2:/home/ubuntu# chown ubuntu:ubuntu /home/ubuntu/jenkins/
root@ip-172-31-93-2:/home/ubuntu#

```

i-076061495bb4039d5 (Manage_Node1)

Public IPs: 3.85.86.170 Private IPs: 172.31.93.2

The prod node is now successfully connected:

```

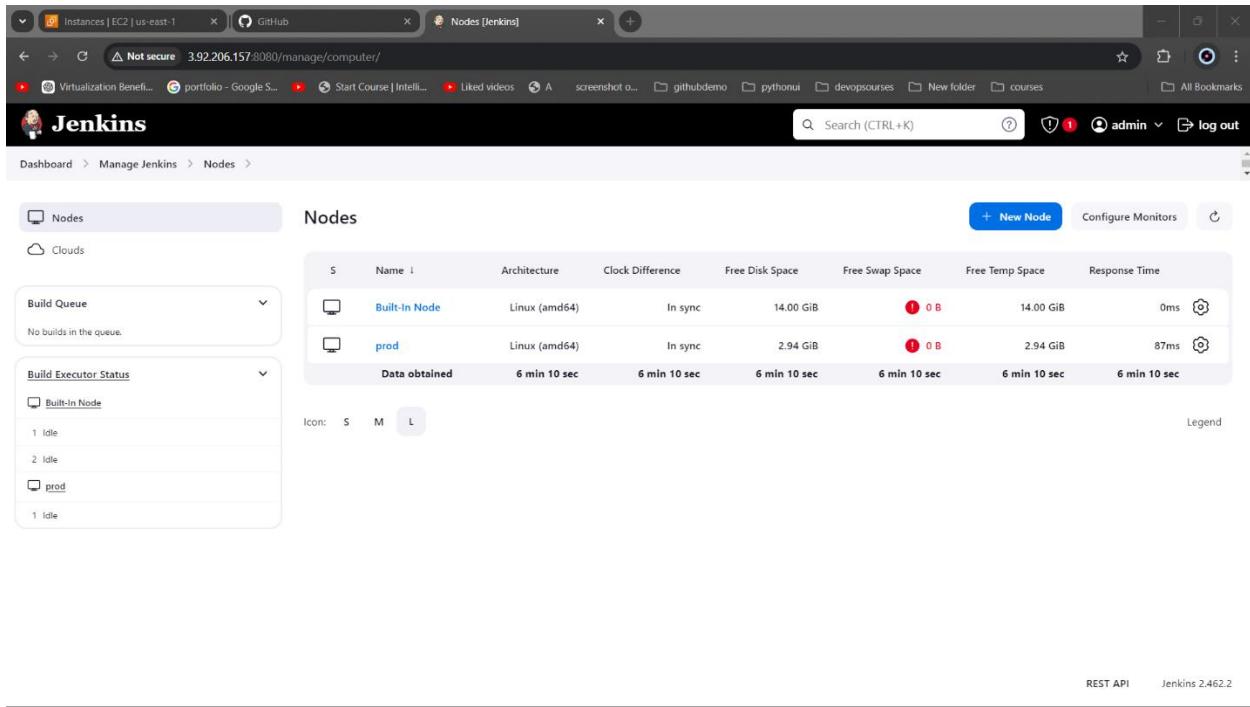
XDG_SESSION_ID=41
XDG_SESSION_TYPE=tty
_='
Checking Java version in the PATH
openjdk version "21.0.4" 2024-07-16
OpenJDK Runtime Environment (build 21.0.4+7-Ubuntu-ubuntu224.04)
OpenJDK 64-Bit Server VM (build 21.0.4+7-Ubuntu-ubuntu224.04, mixed mode, sharing)
[09/13/24 13:07:07] [SSH] Checking java version of /home/ubuntu/jenkins/jdk/bin/java
Couldn't figure out the Java version of /home/ubuntu/jenkins/jdk/bin/java
bash: line 1: /home/ubuntu/jenkins/jdk/bin/java: No such file or directory

[09/13/24 13:07:07] [SSH] Checking java version of java
[09/13/24 13:07:07] [SSH] java -version returned 21.0.4.
[09/13/24 13:07:07] [SSH] Starting sftp client.
[09/13/24 13:07:07] [SSH] Copying latest remoting.jar...
[09/13/24 13:07:08] [SSH] Copied 1,393,029 bytes.
Expanded the channel window size to 4MB
[09/13/24 13:07:08] [SSH] Starting agent process: cd "/home/ubuntu/jenkins" && java -jar remoting.jar -workDir /home/ubuntu/jenkins -jar-cache /home/ubuntu/jenkins/remoting/jarCache
Sep 13, 2024 1:07:08 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Sep 13, 2024 1:07:08 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/jenkins/remoting
<==[JENKINS REMOTING CAPACITY]==>channel started
Remoting version: 3248.3250.v3277a_8e88c9b_
Launcher: SSHLauncher
Communication Protocol: Standard in/out
This is a Unix agent
Agent successfully connected and online

```

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	14.01 GiB	0 B	14.01 GiB	0ms
	prod	Linux (amd64)	In sync	2.95 GiB	0 B	2.95 GiB	53ms
	Data obtained	1 min 39 sec	1 min 39 sec	1 min 39 sec	1 min 39 sec	1 min 39 sec	1 min 39 sec

We are done connecting slave node 1. Now, we need to connect slave node 2 by following the same steps as we did for connecting slave node 1.



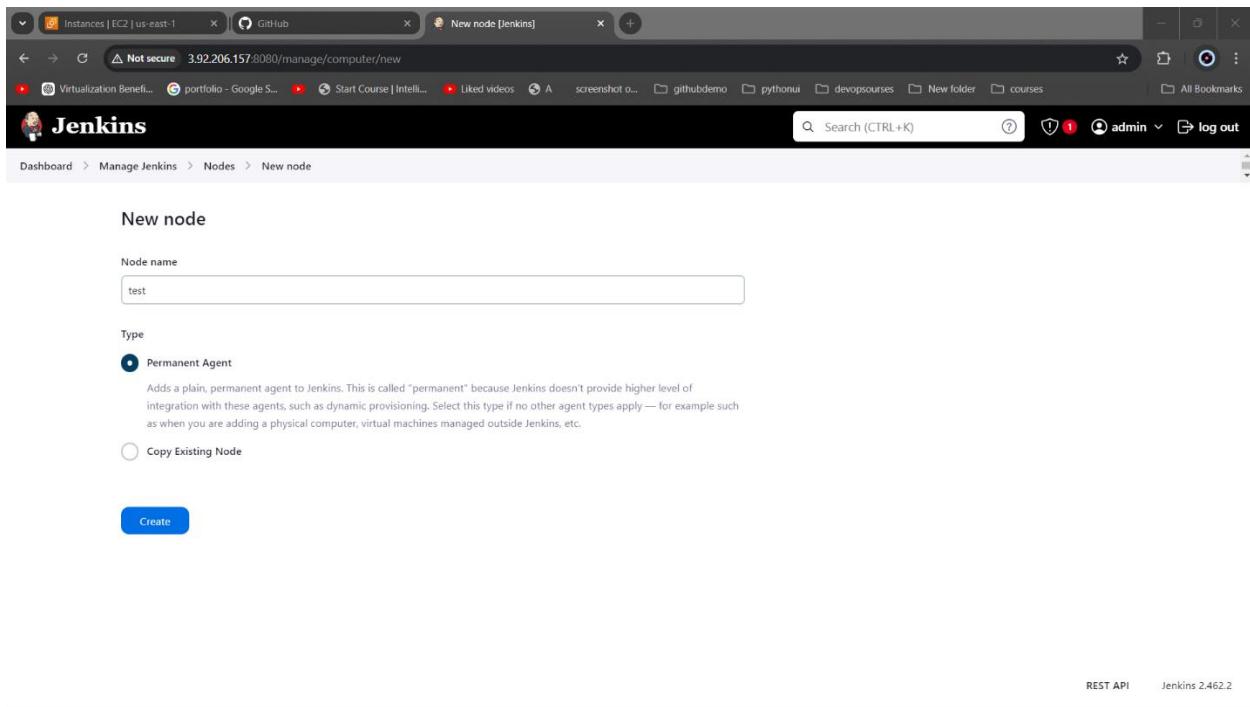
The screenshot shows the Jenkins 'Nodes' page. On the left, there are two expandable sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status'. The 'Build Executor Status' section shows one 'Built-In Node' and one 'prod' node, both of which are listed as 'Idle'. The main table lists the following nodes:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	14.00 GiB	0 B	14.00 GiB	0ms
	prod	Linux (amd64)	In sync	2.94 GiB	0 B	2.94 GiB	87ms
	Data obtained	6 min 10 sec	6 min 10 sec	6 min 10 sec	6 min 10 sec	6 min 10 sec	6 min 10 sec

At the bottom right of the table area, there is a 'Legend' button. At the very bottom of the page, there are links for 'REST API' and 'Jenkins 2.462.2'.

Provide the name of the node.

Select the Permanent Agent option.



The screenshot shows the 'New node' creation form. The 'Node name' field contains 'test'. The 'Type' section has a radio button selected for 'Permanent Agent'. A tooltip explains that this adds a plain, permanent agent to Jenkins. Below the type selection, there is another radio button for 'Copy Existing Node'. At the bottom of the form is a blue 'Create' button. The bottom right of the page shows 'REST API' and 'Jenkins 2.462.2'.

We need to set up a home directory on slave node 2 and specify the root path when connecting the Jenkins node.

The screenshot shows two tabs open in a browser:

- Instances | EC2 | us-east-1**: A CloudShell window displaying terminal output. The user is connected via SSH to an EC2 instance (ip-172-31-93-253). They run `sudo su` and `apt update`. The terminal shows a warning about legacy trusted.gpg keyring. Then, they create a directory `/home/ubuntu/jenkins` and change into it.
- EC2 Instance Connect**: A Jenkins configuration page for a node named "test". The "Configure" tab is selected. The "Description" field contains "test". Under "Build Executor Status", the "Number of executors" is set to 1. The "Remote root directory" is set to `/home/ubuntu/jenkins/`. The "Labels" field contains "test". In the "Usage" dropdown, the option "Use this node as much as possible" is selected. A "Save" button is at the bottom.

Provide connection method as ssh:

Providing private ip:

Launch method ?

Launch agents via SSH

Host ?

172.31.93.253

Credentials ?

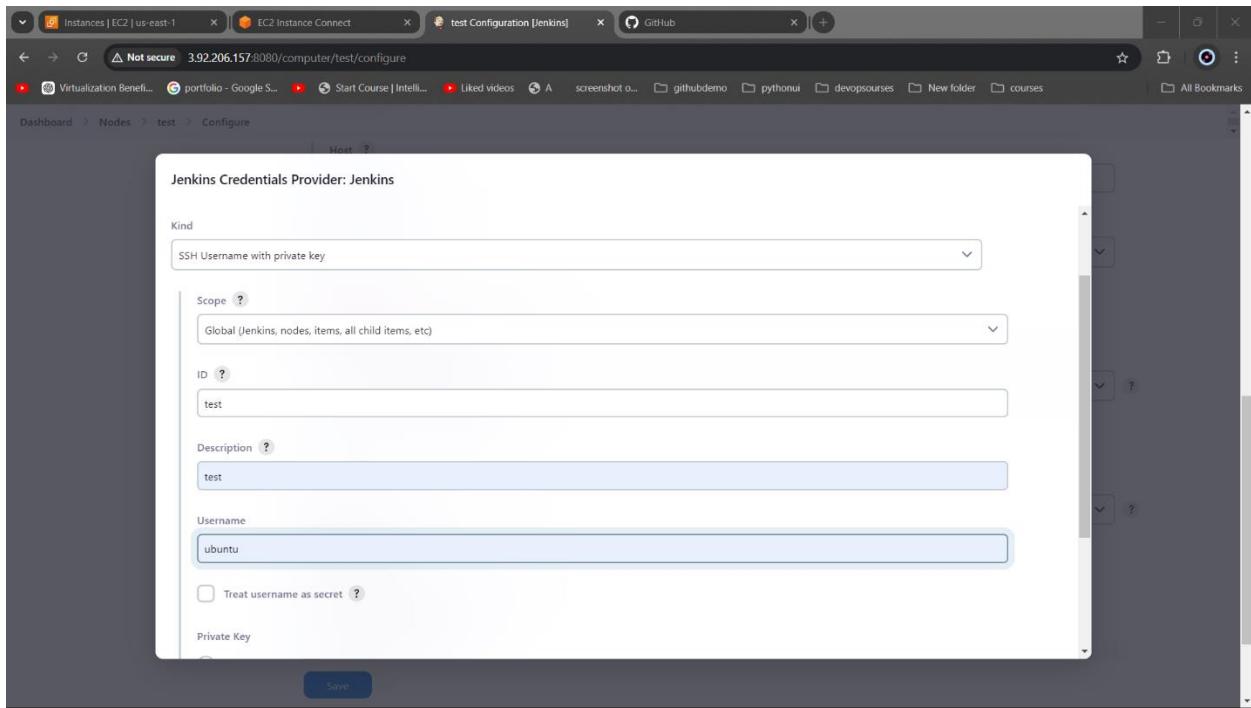
- none -

+ Add ▾

-



We need to provide credentials by adding them for the node and then selecting them.



Jenkins Credentials Provider: Jenkins

Kind: SSH Username with private key

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: test

Description: test

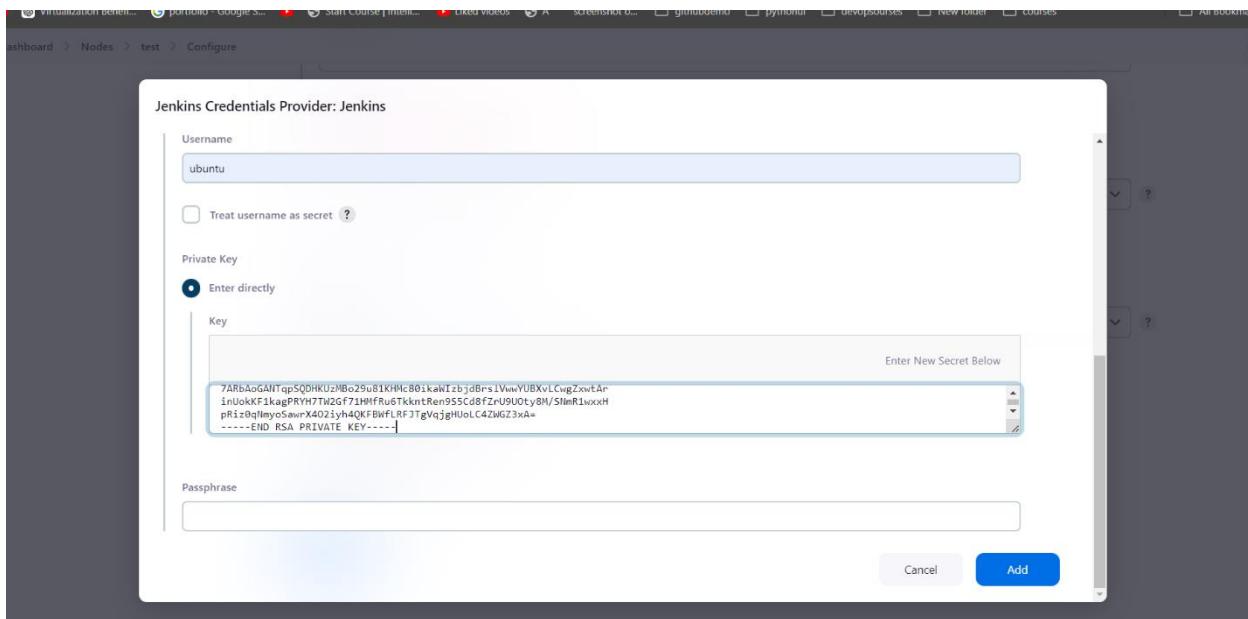
Username: ubuntu

Treat username as secret

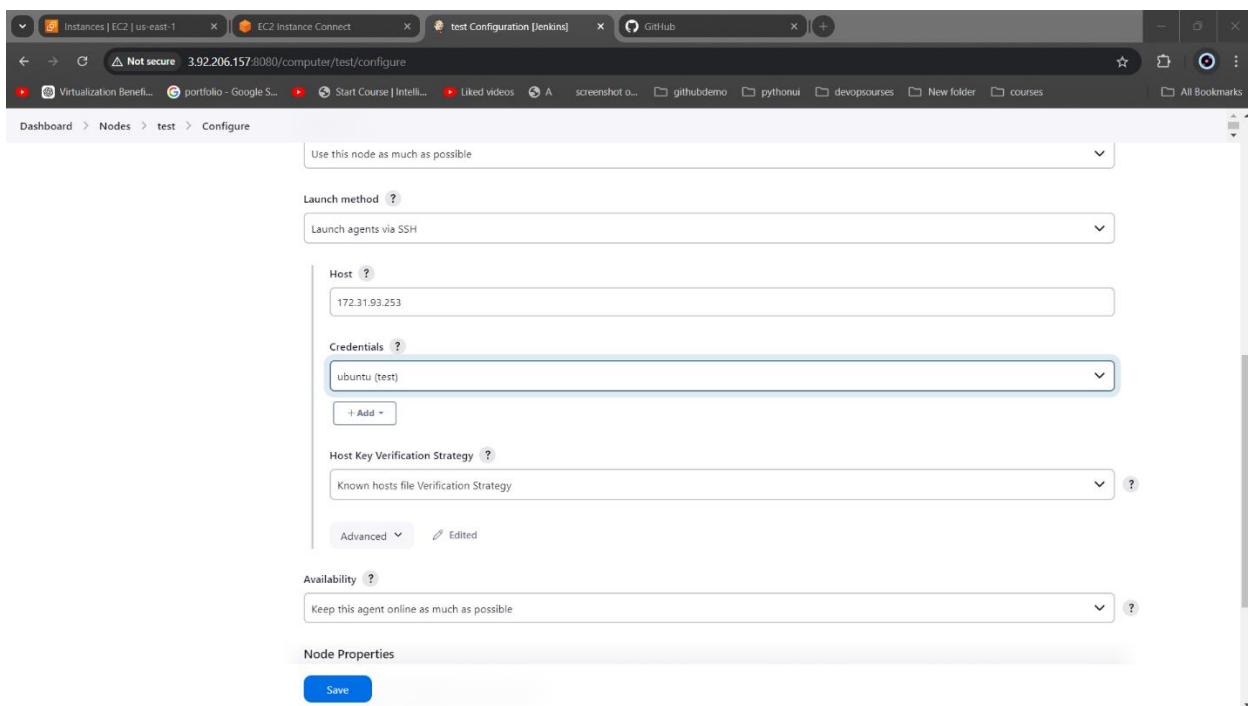
Private Key

Save

This screenshot shows the Jenkins Credentials Provider dialog for creating a new credential. The 'Kind' is set to 'SSH Username with private key'. The 'Scope' is set to 'Global'. The 'ID' is 'test', and the 'Description' is 'test'. The 'Username' is 'ubuntu'. A checkbox for 'Treat username as secret' is unchecked. There is a 'Private Key' section at the bottom which is currently empty. A 'Save' button is at the bottom right.



Save the credentials.



Host Key Verification Strategy.

The screenshot shows the AWS CloudShell interface with multiple tabs open. The active tab is titled 'test Configuration [Jenkins]'. The content of the tab shows the configuration of a slave node. The 'Host' field is set to '172.31.93.253'. The 'Credentials' field is set to 'ubuntu (test)'. The 'Host Key Verification Strategy' dropdown is set to 'Non verifying Verification Strategy'. Under 'Node Properties', there are two unchecked checkboxes: 'Disable deferred wipeout on this node' and 'Disk Space Monitoring Thresholds'. A blue 'Save' button is at the bottom right. The browser's address bar shows the URL '3.92.206.157:8080/computer/test/configure'.

First, run the required command on slave node 2, and then proceed to launch the node.

→ Sudo chown ubuntu:ubuntu /home/ubuntu/jenkins/

The screenshot shows the AWS CloudShell terminal window. The command 'sudo chown ubuntu:ubuntu /home/ubuntu/jenkins/' is being typed and executed. The terminal output shows the user is root on an Ubuntu 20.04 instance. It lists package updates from the 'focal' repository, installs Docker, and creates the Jenkins directory. The command is completed successfully with a status of 0. The AWS CloudShell interface includes a navigation bar with tabs like 'CloudShell', 'Feedback', and 'Cookie preferences'.

```
Last login: Fri Sep 13 13:43:14 2024 from 18.206.107.28
ubuntu@ip-172-31-93-253:~$ sudo su
root@ip-172-31-93-253:/home/ubuntu# apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [502 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [366 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Fetched 1200 kB in 1s (913 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
107 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://download.docker.com/linux/ubuntu/dists/focal/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
root@ip-172-31-93-253:/home/ubuntu# mkdir /home/ubuntu/jenkins/
root@ip-172-31-93-253:/home/ubuntu# ls
jenkins
root@ip-172-31-93-253:/home/ubuntu# chown ubuntu:ubuntu /home/ubuntu/jenkins/ []

i-0cd0f347cb719fc81 (Manage_Node2)
Public IPs: 54.159.162.91 Private IPs: 172.31.93.253
```

```

Instances | EC2 | us-east-1 EC2 Instance Connect test Configuration [Jenkins] GitHub
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0cd0f347cb719fc81&osUser=ubuntu&region=us-east-1&sshPort=22/
Virtualization Benefits portfolio - Google Search Start Course | IntelliJ... Liked videos A screenshot o... githubdemo pythonu devopsourses New folder courses All Bookmarks N. Virginia pradeep
aws Services Search [Alt+S]
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Sep 13 13:43:14 2024 from 18.206.107.28
ubuntu@ip-172-31-93-253:~$ sudo su
root@ip-172-31-93-253:/home/ubuntu# apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Get:6 https://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [502 kB]
Get:7 https://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [366 kB]
Get:8 https://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:9 https://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Fetched 1200 kB in 1s (913 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
107 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://download.docker.com/linux/ubuntu/dists/focal/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section
  in apt-key(8) for details.
root@ip-172-31-93-253:/home/ubuntu# mkdir /home/ubuntu/jenkins/
root@ip-172-31-93-253:/home/ubuntu# ls
jenkins
root@ip-172-31-93-253:/home/ubuntu# chown ubuntu:ubuntu /home/ubuntu/jenkins/
root@ip-172-31-93-253:/home/ubuntu# []

```

i-0cd0f347cb719fc81 (Manage_Node2)
 PublicIPs: 54.159.162.91 PrivateIPs: 172.31.93.255

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Save the node and check the connection:

Not secure 3.92.206.157:8080/computer/test/log

Dashboard > Nodes > test > Log

```

Checking Java version in the PATH
openjdk version "21.0.4" 2024-07-16
OpenJDK Runtime Environment (build 21.0.4+7-Ubuntu-ubuntu224.04)
OpenJDK 64-Bit Server VM (build 21.0.4+7-Ubuntu-ubuntu224.04, mixed mode, sharing)
[09/14/24 06:32:02] [SSH] Checking java version of /home/ubuntu/jenkins/jdk/bin/java
Couldn't figure out the Java version of /home/ubuntu/jenkins/jdk/bin/java
bash: line 1: /home/ubuntu/jenkins/jdk/bin/java: No such file or directory

[09/14/24 06:32:02] [SSH] Checking java version of java
[09/14/24 06:32:02] [SSH] java -version returned 21.0.4.
[09/14/24 06:32:02] [SSH] Starting sftp client...
[09/14/24 06:32:02] [SSH] Copying latest remoting.jar...
[09/14/24 06:32:02] [SSH] Copied 1,393,029 bytes.
Expanded the channel window size to 4MB
[09/14/24 06:32:02] [SSH] Starting agent process: cd "/home/ubuntu/jenkins" && java -jar remoting.jar -workDir /home/ubuntu/jenkins -jar-cache /home/ubuntu/jenkins/remoting/jarCache
Sep 14, 2024 6:32:02 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Sep 14, 2024 6:32:03 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/jenkins/remoting
<==[JENKINS REMOTING CAPACITY]==>channel started
Remoting version: 3248.3250.v3277a_8e88c9b_
Launcher: SSHLauncher
Communication Protocol: Standard in/out
This is a Unix agent
Agent successfully connected and online

```

REST API Jenkins 2.462.2

Node is successfully connected and then it is in online.

With both nodes online and connected, the next step is to create jobs and allocate them to these nodes.

The screenshot shows the Jenkins Nodes page. At the top, there are tabs for Instances | EC2 | us-east-1, EC2 Instance Connect, Nodes [Jenkins], and GitHub. The Nodes tab is active. The main content area is titled "Nodes". It displays a table with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. There are three rows:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	14.00 GiB	0 B	14.00 GiB	0ms
	prod	Linux (amd64)	In sync	2.94 GiB	0 B	2.94 GiB	36ms

Below the table, there is a section titled "Build Executor Status" which lists "Built-In Node" with 2 idle instances and "prod" with 1 idle instance. A legend at the bottom indicates icons for Slave (S), Master (M), and Label (L).

REST API Jenkins 2.462.2

The screenshot shows the Jenkins Dashboard page. At the top, there are tabs for Instances | EC2 | us-east-1, EC2 Instance Connect, Dashboard [Jenkins], and GitHub. The Dashboard tab is active. The main content area is titled "Welcome to Jenkins!". It includes a message: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." Below this, there is a "Start building your software project" section with a "Create a job" button. On the left side, there is a sidebar with links: + New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. The sidebar also shows sections for Build Queue and Build Executor Status, which are identical to the Nodes page.

' Click on create new job:

Creating docker file and pushing it to docker hub:

```
FROM ubuntu
RUN apt update -y && apt install -y apache2 git
RUN rm -rf /var/www/html/*
RUN git clone https://github.com/hshar/website.git /var/www/html
RUN mv /var/www/html/website/* /var/www/html && rm -rf /var/www/html/website
ENTRYPOINT apachectl -D FOREGROUND
```

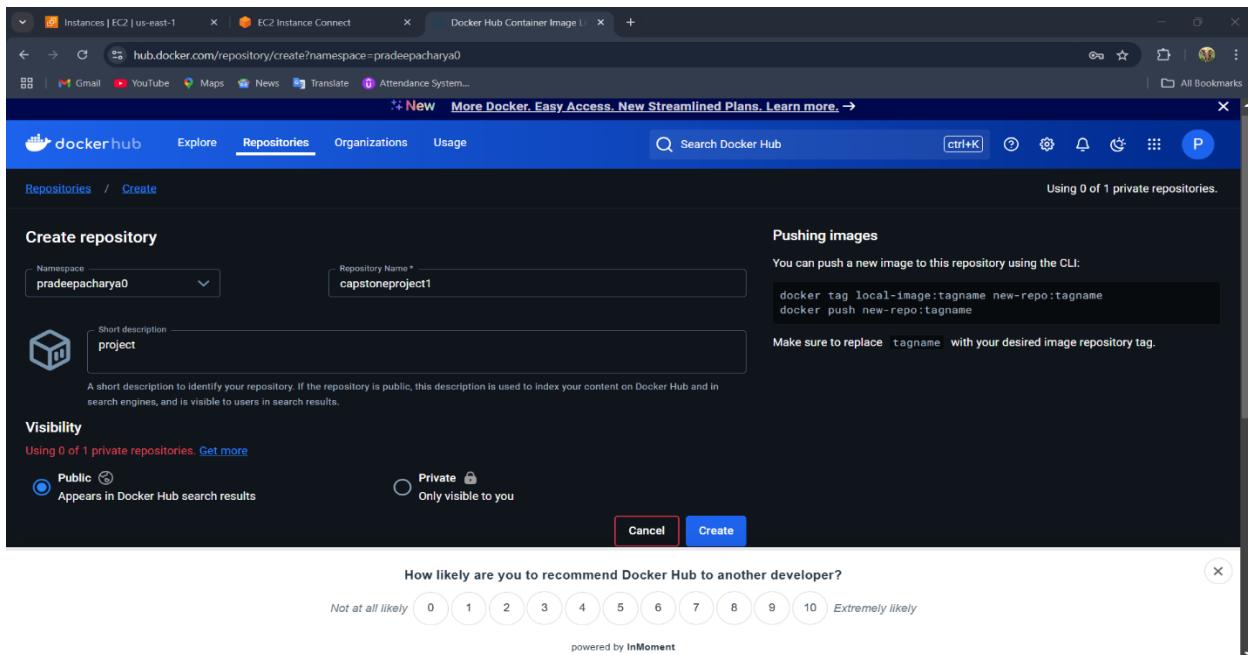
To start, we should create a Dockerfile on the local host before uploading it to Docker Hub.

The screenshot shows a web browser window with the Docker Hub interface. The address bar shows 'hub.docker.com'. The main navigation bar includes 'Explore', 'Repositories' (which is selected), 'Organizations', and 'Usage'. A search bar at the top right says 'Search Docker Hub'. Below the search bar, there's a dropdown menu set to 'pradeepacharya0' and a 'Create repository' button. The main content area displays four repositories under the namespace 'pradeepacharya0': 'jenkins_casestudy', 'dockerassignment3', 'maven', and 'newfile'. Each repository card shows a star icon (0 stars), a date ('Last pushed: 3 months ago' or '4 months ago'), a public status, and a 'Scout inactive' badge. To the right of the repository list, there's a sidebar with options like 'Create An Org', 'Create and manage access to your r...', and a survey for 'How likely are you to recommend Docker Hub to another developer?' with a scale from 0 to 10.

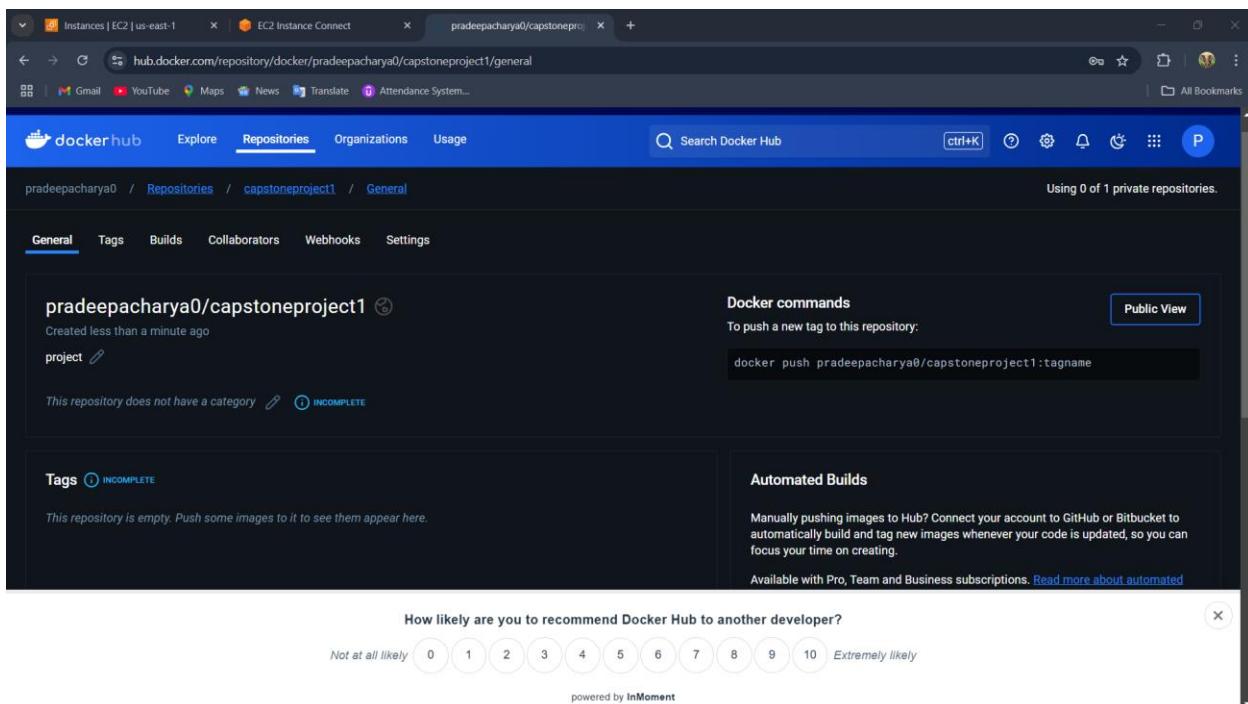
Now we need to create a Dockerfile named CapstoneProject1.

The screenshot shows a web browser window with the Docker Hub interface. The address bar shows 'hub.docker.com/repository/create?namespace=pradeepacharya0'. The main navigation bar includes 'Explore', 'Repositories' (selected), 'Organizations', and 'Usage'. A search bar at the top right says 'Search Docker Hub'. Below the search bar, there's a 'Create repository' button. The main content area has a 'Create repository' form. In the 'Namespace' dropdown, 'pradeepacharya0' is selected. In the 'Repository Name' field, 'capstoneproject1' is entered. Below the name field, there's a 'Short description' input with the placeholder 'project'. A note below the description says: 'A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.' Under the 'Visibility' section, the 'Public' radio button is selected, with the note 'Appears in Docker Hub search results'. The 'Private' radio button is also present. At the bottom of the form are 'Cancel' and 'Create' buttons. To the right of the form, there's a 'Pushing images' section with instructions: 'You can push a new image to this repository using the CLI:' followed by 'docker tag local-image:tagname new-repo:tagname' and 'docker push new-repo:tagname'. It also says 'Make sure to replace tagname with your desired image repository tag.' Below the instructions is a survey for 'How likely are you to recommend Docker Hub to another developer?' with a scale from 0 to 10. The footer of the page says 'powered by InMoment'.

"Click 'Create' to set up a new Docker repository.



The repository was created successfully.



Now we need to create a Dockerfile on the local host and then push it to Docker Hub.

```
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1015-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Wed Nov 13 15:42:06 UTC 2024

System load: 0.89 Processes: 128
Usage of /: 22.5% of 18.33GB Users logged in: 0
Memory usage: 17% IPv4 address for enX0: 172.31.85.119
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

99 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

last login: Fri Nov 1 15:15:56 2024 from 18.206.107.29
ubuntu@ip-172-31-85-119:~$ ls
gitrepo playbook1.yaml playbook2.yaml website
ubuntu@ip-172-31-85-119:~$ ||

i-0c48d225109c4d789 (Controller_node)
PublicIP: 54.92.243.102 PrivateIPs: 172.31.85.119
```

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Wed Nov 13 15:42:06 UTC 2024

System load: 0.89 Processes: 128
Usage of /: 22.5% of 18.33GB Users logged in: 0
Memory usage: 17% IPv4 address for enX0: 172.31.85.119
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

99 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

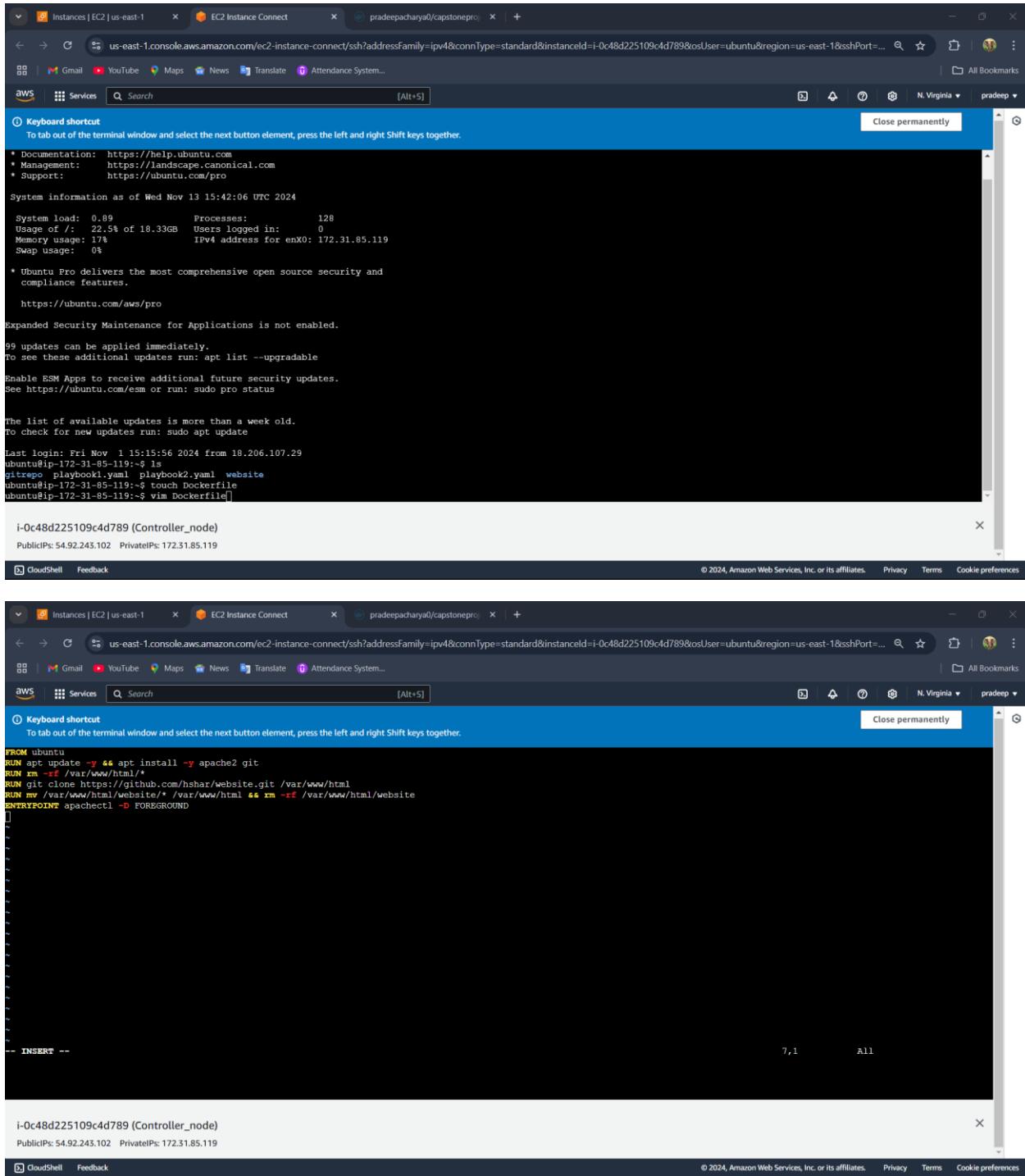
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Nov 1 15:15:56 2024 from 18.206.107.29
ubuntu@ip-172-31-85-119:~$ ls
gitrepo playbook1.yaml playbook2.yaml website
ubuntu@ip-172-31-85-119:~$ touch Dockerfile []

i-0c48d225109c4d789 (Controller_node)
PublicIP: 54.92.243.102 PrivateIPs: 172.31.85.119
```

→ Touch dockerfile

After creating a file called Dockerfile, open it in Vim and add the required Dockerfile instructions.



The screenshot shows a web browser window with two tabs. The active tab is titled "Instances | EC2 | us-east-1" and shows a terminal session on an AWS EC2 instance. The terminal output includes:

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Wed Nov 13 15:42:06 UTC 2024
System load: 0.89 Processes: 128
Usage of /: 22.5% of 18.33GB Users logged in: 0
Memory usage: 17% IPv4 address for enX0: 172.31.85.119
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

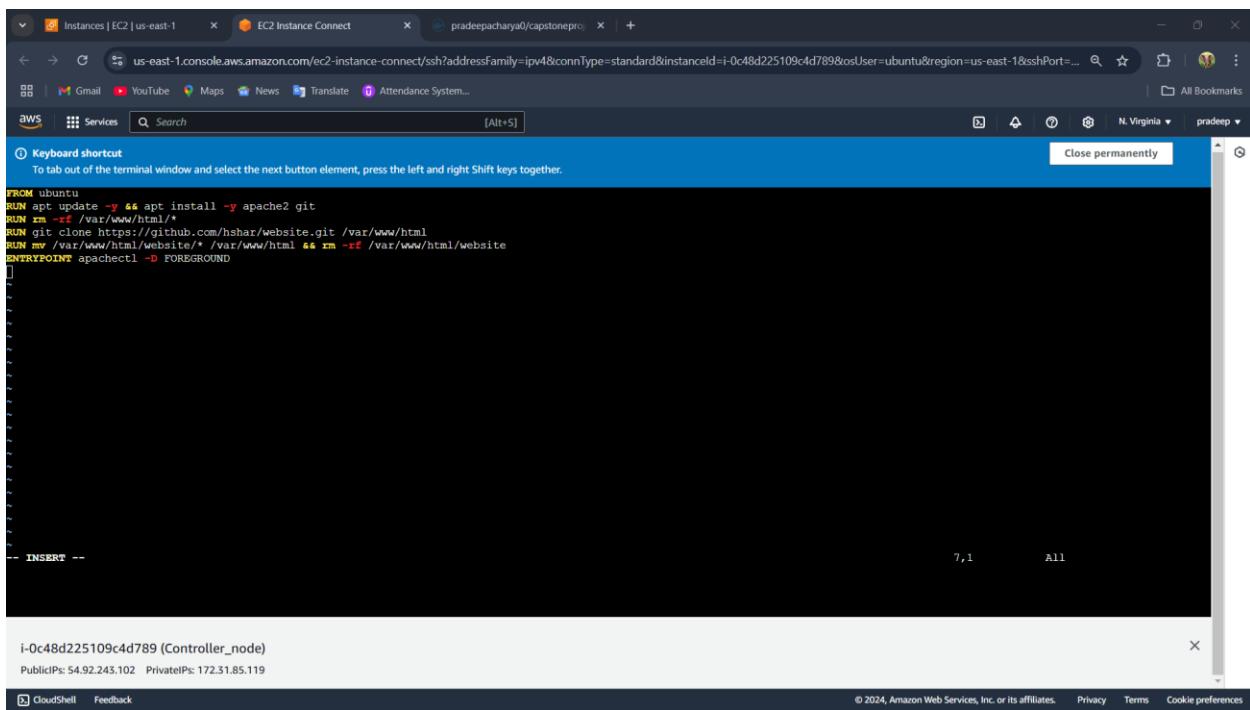
99 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Nov 1 15:15:56 2024 from 10.206.107.29
ubuntu@ip-172-31-85-119:~$ ls
gitrepo playbook1.yaml playbook2.yaml website
ubuntu@ip-172-31-85-119:~$ touch Dockerfile
ubuntu@ip-172-31-85-119:~$ vim Dockerfile[]

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 54.92.243.102 PrivateIPs: 172.31.85.119
```

The second tab is titled "EC2 Instance Connect". Below the tabs, the browser navigation bar shows "us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?addressFamily=ipv4&connType=standard&instanceId=i-0c48d225109c4d789&osUser=ubuntu®ion=us-east-1&sshPort=...". The bottom of the browser window shows standard links like CloudShell, Feedback, and cookie preferences.



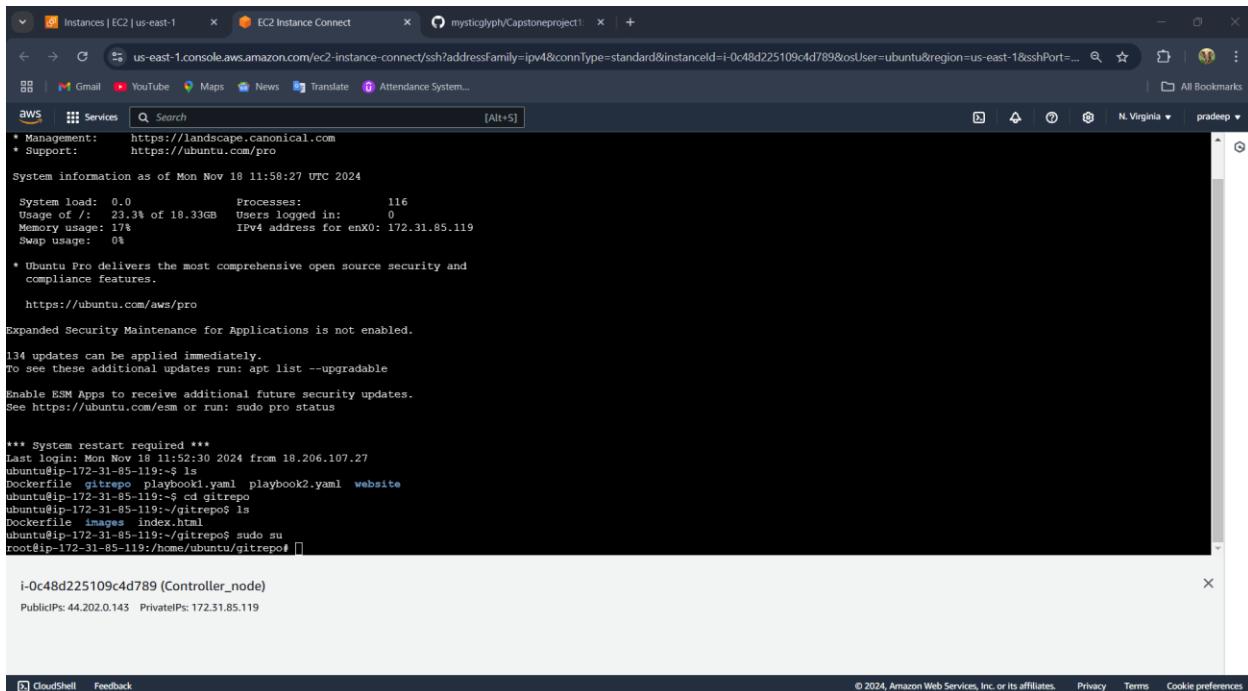
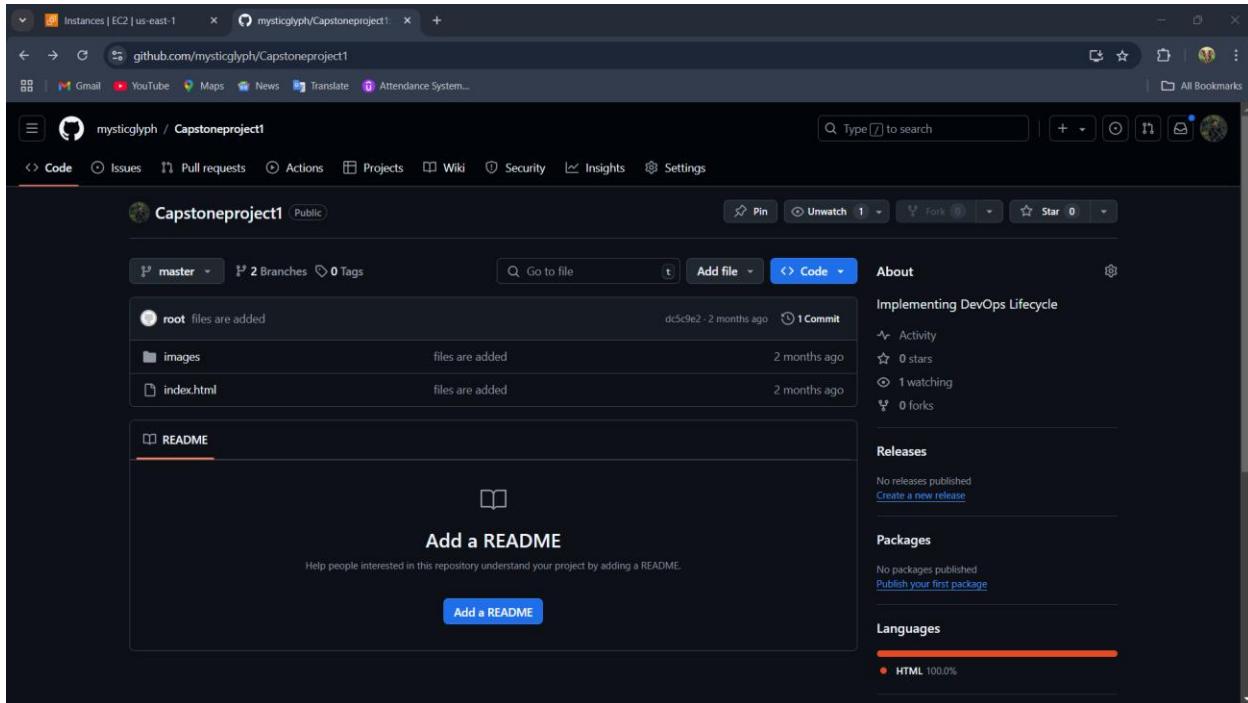
The second screenshot shows the same terminal session after the Dockerfile has been edited. The terminal output now includes the Dockerfile content:

```
FROM ubuntu
RUN apt update -y && apt install -y apache2 git
RUN rm -rf /var/www/html/*
RUN git clone https://github.com/hshar/website.git /var/www/html
RUN mv /var/www/html/website/* /var/www/html && rm -rf /var/www/html/website
ENTRYPOINT apachectl -D FOREGROUND
```

The bottom of the browser window shows standard links like CloudShell, Feedback, and cookie preferences.

Save the Dockerfile and then push it to the Git hub repository.

The Dockerfile must be pushed to the pre-existing GitHub repository.



We need to ensure that the Git repository is connected to our local repository. Now, we need to add and push the Dockerfile to GitHub.

```
*** System restart required ***
Last login: Mon Nov 18 11:52:30 2024 from 18.206.107.27
ubuntu@ip-172-31-85-119:~$ ls
Dockerfile  gitrepo  playbook1.yaml  playbook2.yaml  website
ubuntu@ip-172-31-85-119:~$ cd gitrepo
ubuntu@ip-172-31-85-119:~/gitrepo$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-85-119:~/gitrepo$ sudo su
root@ip-172-31-85-119:/home/ubuntu/gitrepo$ ls
Dockerfile  images  index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo$ git remote -v
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (fetch)
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (push)
root@ip-172-31-85-119:/home/ubuntu/gitrepo$ [REDACTED]
```

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 44.202.0.143 PrivateIPs: 172.31.85.119

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- ➔ Add all the file to staging area.
- ➔ Git add .

```
Instances | EC2 | us-east-1 | EC2 Instance Connect | mysticglyph/Capstoneproject1 | +
```

Usage of /: 23.3% of 18.33GB Users logged in: 0
Memory usage: 17% IPv4 address for enX0: 172.31.85.119
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and compliance features.

<https://ubuntu.com/aws/pro>

Expanded Security Maintenance for Applications is not enabled.

134 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

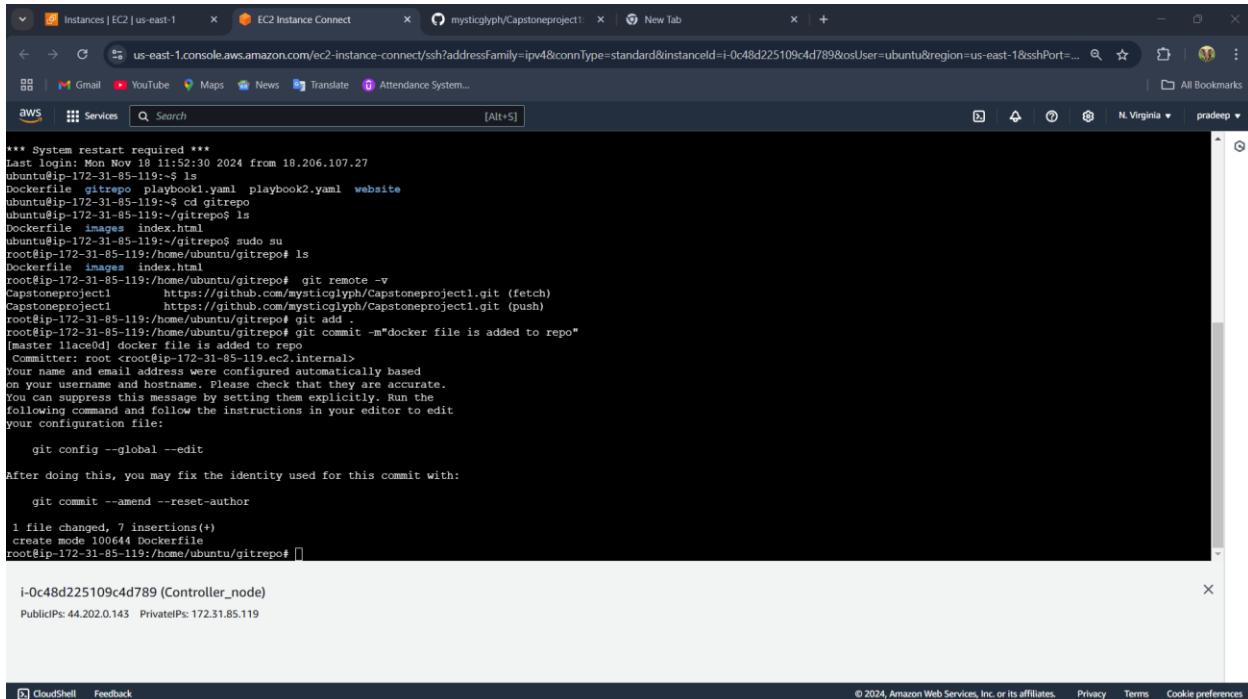
```
*** System restart required ***
Last login: Mon Nov 18 11:52:30 2024 from 18.206.107.27
ubuntu@ip-172-31-85-119:~$ ls
Dockerfile  gitrepo  playbook1.yaml  playbook2.yaml  website
ubuntu@ip-172-31-85-119:~$ cd gitrepo
ubuntu@ip-172-31-85-119:~/gitrepo$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-85-119:~/gitrepo$ sudo su
root@ip-172-31-85-119:/home/ubuntu/gitrepo$ ls
Dockerfile  images  index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo$ git remote -v
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (fetch)
Capstoneproject1  https://github.com/mysticglyph/Capstoneproject1.git (push)
root@ip-172-31-85-119:/home/ubuntu/gitrepo$ git add .
root@ip-172-31-85-119:/home/ubuntu/gitrepo$ [REDACTED]
```

i-0c48d225109c4d789 (Controller_node)
PublicIPs: 44.202.0.143 PrivateIPs: 172.31.85.119

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Commit the changes.

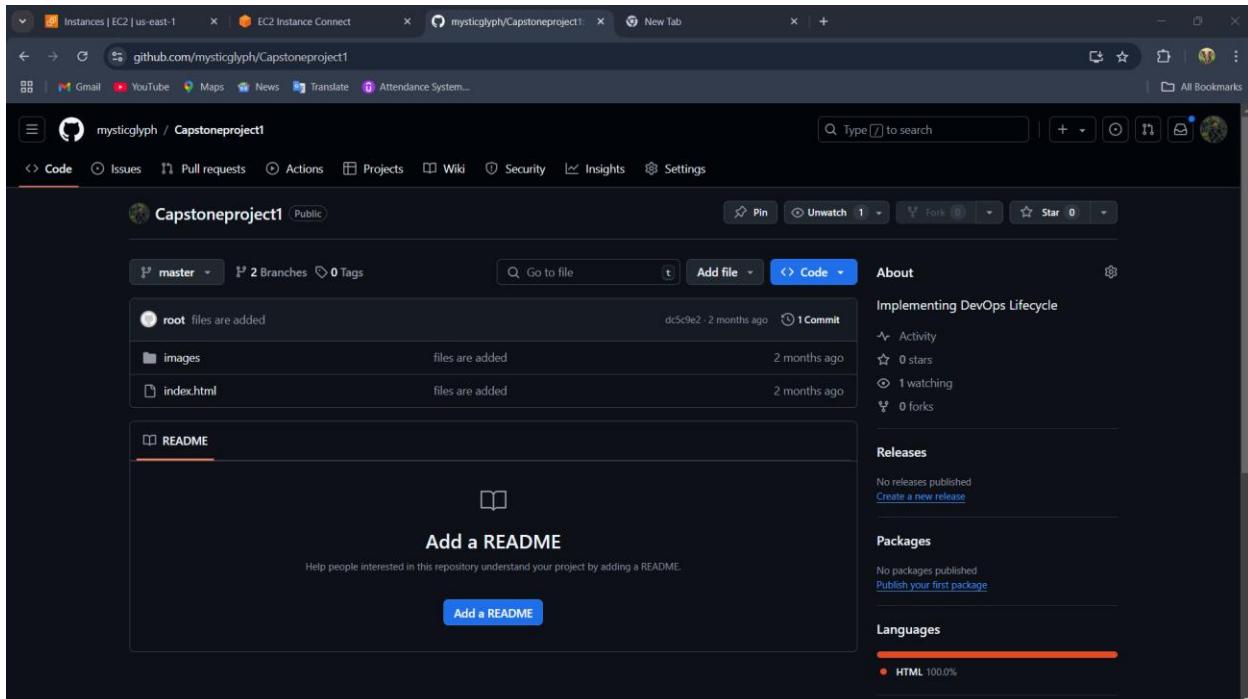
Git commit -m "docker file is added to repo".



```
*** System restart required ***
Last login: Mon Nov 18 11:52:30 2024 from 18.206.107.27
ubuntu@ip-172-31-85-119:~$ ls
Dockerfile  gitrepo  playbook1.yaml  playbook2.yaml  website
ubuntu@ip-172-31-85-119:~$ cd gitrepo
ubuntu@ip-172-31-85-119:/home/ubuntu/gitrepo$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-85-119:/home/ubuntu/gitrepo$ sudo su
root@ip-172-31-85-119:/home/ubuntu/gitrepo# ls
Dockerfile  images  index.html
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git remote -v
Capstoneproject1  https://github.com/mysticglyph/capstoneproject1.git (fetch)
Capstoneproject1  https://github.com/mysticglyph/capstoneproject1.git (push)
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git add .
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git commit -m"docker file is added to repo"
[master 11ace0d] docker file is added to repo
Committer: root<root@ip-172-31-85-119.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
git config --global --edit
After doing this, you may fix the identity used for this commit with:
git commit --amend --reset-author
1 file changed, 7 insertions(+)
create mode 100644 Dockerfile
root@ip-172-31-85-119:/home/ubuntu/gitrepo# 
i-0c48d225109c4d789 (Controller_node)
Public IPs: 44.202.0.143 Private IPs: 172.31.85.119
```

Now, we need to push the file to the Git repository.

Before pushing to GitHub, we need to generate a token from GitHub, which will be used as a password during the push process.

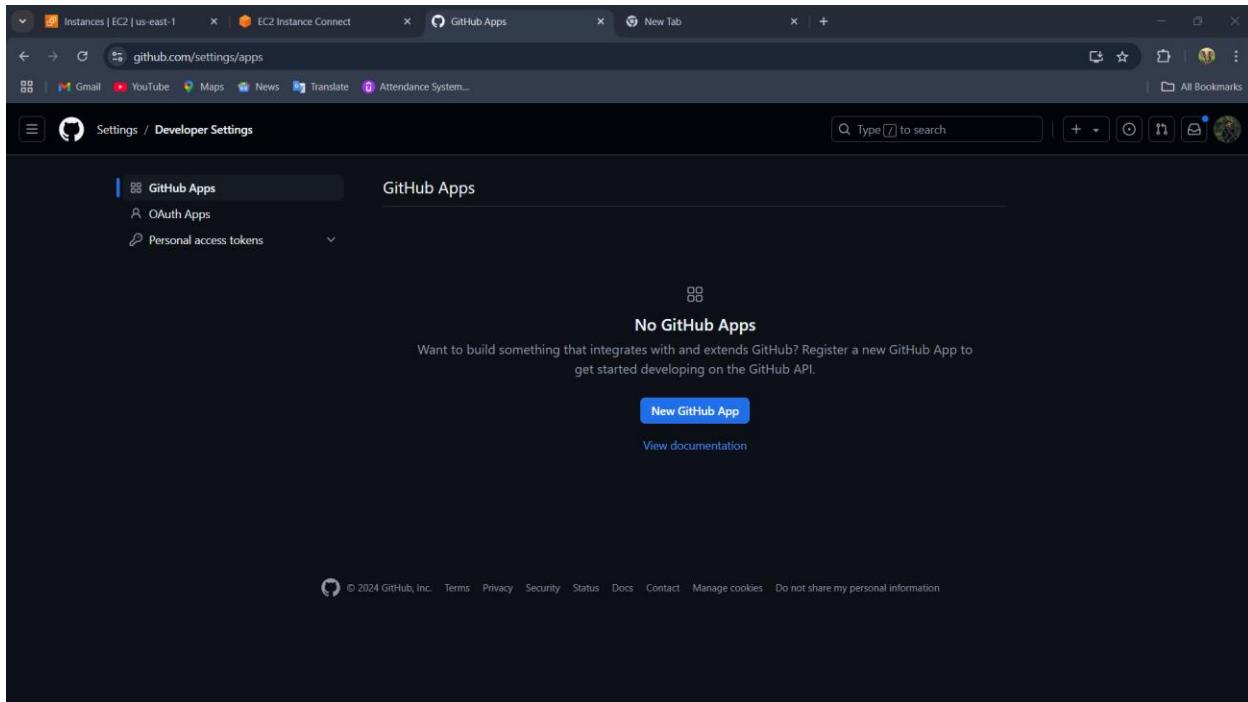


Goto -> settings:

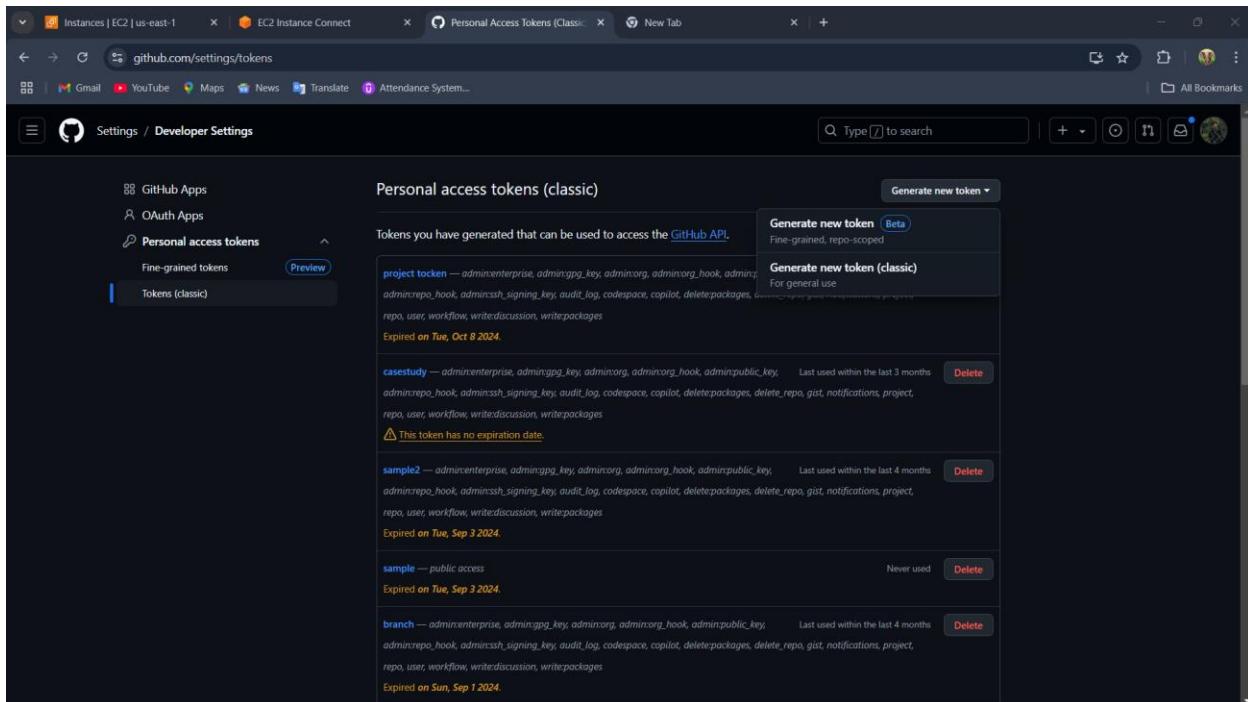
The screenshot shows a GitHub repository named 'Capstoneproject1'. The 'Code' tab is selected, displaying a list of files: 'root' (files added), 'images' (files added), and 'index.html' (files added). A 'README' section is present with a button to 'Add a README'. The sidebar on the right shows the user's profile ('mysticlyph pradeep acharya') and various GitHub features: Your profile, Your repositories, Your Copilot, Your projects, Your stars, Your gists, Your organizations, Your enterprises, Your sponsors, Try Enterprise, Feature preview, Settings, GitHub Docs, GitHub Support, GitHub Community, and Sign out.

Developer's settings:

The screenshot shows the 'Your profile' settings page. On the left, there are several sections: Security, Code security, Integrations, Applications, Scheduled reminders, Archives, Security log, Sponsorship log, and Developer settings. The main area contains fields for linking social profiles, company information, location, and a checkbox for 'Display current local time'. Below this is the 'Contributions & activity' section, which includes checkboxes for 'Make profile private and hide activity' (which hides contributions and activity from the GitHub profile) and 'Include private contributions on my profile' (which shows private contributions in the contribution graph and activity overview). At the bottom of this section is a 'Update preferences' button.



Personal access tokens -> classic tokens:



Generate new token:

The screenshot shows the GitHub 'Personal Access Tokens (classic)' page. At the top right, there is a 'Generate new token' button with a dropdown menu. The menu has two items: 'Generate new token (Beta)' (marked as 'Fine-grained, repo-scoped') and 'Generate new token (classic)' (marked as 'For general use'). Below the menu, a table lists tokens. One token is shown in detail: 'capstones1' (Last used within the last 3 months), with a 'Delete' button.

Token	Last used	Action
capstones1	Within the last 3 months	Delete

The screenshot shows the GitHub 'New personal access token (classic)' creation page. On the left, a sidebar lists 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens' (with 'Tokens (classic)' selected). The main area is titled 'New personal access token (classic)'. It includes fields for 'Note' (containing 'capstones1'), 'Expiration' (set to '30 days'), and a 'Select scopes' section. The 'Select scopes' section lists various GitHub API permissions:

Scope	Description
<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repository	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo_invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry

Select all the scopes define the access for personal tokens.

The screenshot shows the GitHub 'Personal access tokens' section. A note 'capstones1' is entered in the 'Note' field. The 'Expiration' dropdown is set to '90 days'. Under 'Select scopes', several options are checked: repo, workflow, write:packages, delete:packages, admin:org, write:discussion, admin:enterprise, audit_log, codespace, copilot, project, admin:gpg_key, and admin:ssh_signing_key. The 'repo' scope includes sub-options like repostatus, repo_deployment, public_repo, repoinit, and security_events. The 'write:packages' scope includes readpackages. The 'delete:packages' scope includes deletepackages. The 'admin:org' scope includes writeorg. The 'write:discussion' scope includes read:discussion. The 'admin:enterprise' scope includes manage_runners:enterprise, manage_billing:enterprise, readenterprise, and scimenterprise. The 'audit_log' scope includes readaudit_log. The 'codespace' scope includes codespace:secrets. The 'copilot' scope includes manage_billing:copilot. The 'project' scope includes read:project. The 'admin:gpg_key' scope includes writegpg_key and readgpg_key. The 'admin:ssh_signing_key' scope includes writessh_signing_key and readssh_signing_key.

This screenshot shows the same GitHub token creation interface with more expanded scopes listed. The expanded list includes additional enterprise-related scopes: enterprise, manage_entitlements, manage_billing, and manage_entitlements. It also includes organization-related scopes: org, org:settings, and org:members. The expanded list for 'write:discussion' includes manage_discussions. The expanded list for 'admin:enterprise' includes manage_entitlements, manage_billing, and manage_entitlements. The expanded list for 'audit_log' includes readaudit_log. The expanded list for 'codespace' includes codespace:secrets. The expanded list for 'copilot' includes manage_billing:copilot. The expanded list for 'project' includes read:project. The expanded list for 'admin:gpg_key' includes writegpg_key and readgpg_key. The expanded list for 'admin:ssh_signing_key' includes writessh_signing_key and readssh_signing_key.

Click on Generate token:

The screenshot shows the GitHub 'Personal access tokens (classic)' page. On the left, there's a sidebar with 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens' (which is expanded). Under 'Personal access tokens', there are two sections: 'Tokens (classic)' (selected) and 'Preview'. A message at the top says, 'Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.' Below this, a note says, 'Tokens you have generated that can be used to access the [GitHub API](#).'. A warning message in a box says, 'Make sure to copy your personal access token now. You won't be able to see it again!' Below this, three tokens are listed:

- gph_HC71W4qCBtTyF3GMauVj11t0RIGZ0FBz1C** (Copied icon) - Last used within the last 3 months. Scopes: project_token — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key. Expires on Tue, Oct 8 2024.
- project_token** — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key. Last used within the last 3 months. Scopes: admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, delete_packages, delete_repo, gist, notifications, project, repo, user, workflow, write_discussion, write_packages.
- casedstudy** — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key. Last used within the last 3 months. Scopes: admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, delete_packages, delete_repo, gist, notifications, project, repo, user, workflow, write_discussion, write_packages. A warning icon says, '⚠ This token has no expiration date.'
- sample2** — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key. Last used within the last 4 months. Scopes: admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, delete_packages, delete_repo, gist, notifications, project, repo, user, workflow, write_discussion, write_packages. A warning icon says, '⚠ This token has no expiration date.'

The token has been successfully generated.

Now, we can proceed with the git push operation.

→ Git push

```
To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

root@ip-172-31-85-119:/home/ubuntu/gitrepo# sudo au
sudo: au: command not found
root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream Capstoneproject1 master

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

root@ip-172-31-85-119:/home/ubuntu/gitrepo# git push --set-upstream Capstoneproject1 master
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression objects: 100% (3/3), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 494 bytes | 494.00 KiB/s, done.
Total 3 delta 0, reused 0 (delta 0), pack-reused 0
To https://github.com/mysticglyph/Capstoneproject1.git
 * [new branch] master -> master
branch 'master' set up to track 'Capstoneproject1/master'.

```

The Dockerfile has been successfully added to our capstone project repository. We can now use the repository for defining jobs.

```
FROM ubuntu
RUN apt update -y & apt install -y apache2 git
RUN rm -rf /var/www/html/*
RUN git clone https://github.com/mysticglyph/Capstoneproject1.git /var/www/html
RUN mv /var/www/html/Capstoneproject1/* /var/www/html
ENTRYPOINT apache2ctl -D FOREGROUND
```

1. Job1 : build
2. Job2 : test
3. Job3 : prod

Start by creating the build job, and then move on to creating the remaining jobs.

The screenshot shows the Jenkins dashboard at the URL <http://3.87.42.83:8080>. The main heading is "Welcome to Jenkins!". Below it, a message says: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." A "Create a job" button is visible. On the left, there's a sidebar with links like "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views". Below the sidebar, there are two sections: "Build Queue" (which says "No builds in the queue") and "Build Executor Status" (which lists "Built-In Node" with entries for "prod" and "test", both marked as "Idle").

This screenshot shows the same Jenkins dashboard after a refresh. The main heading is now "Welcome". The "Create a job" button is highlighted with a light gray background. The rest of the interface is identical to the previous screenshot, including the sidebar links and the "Build Queue" and "Build Executor Status" sections.

Click on new item.

New Item

Enter an item name

> This field cannot be empty, please enter a valid name

Select an item type

- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

A name must be assigned to the job, and the job type should be selected as 'Freestyle.' Once the job is created, it needs to be configured accordingly.

New Item

Enter an item name

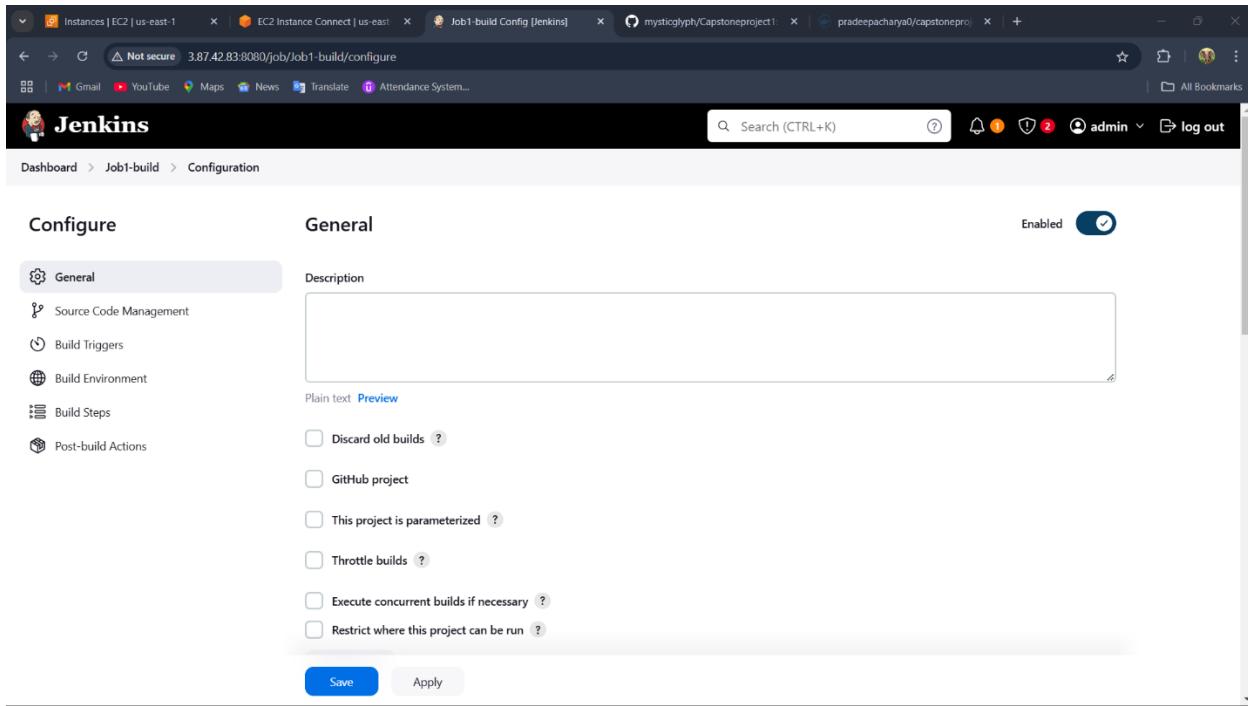
Job1-build

Select an item type

- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Click on "ok"



The screenshot shows the Jenkins configuration interface for a job named "Job1-build". The "General" tab is selected. On the left, there's a sidebar with options like General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main area has a "Description" field which is currently empty. Below it are several checkboxes for build-related settings: "Discard old builds", "GitHub project", "This project is parameterized", "Throttle builds", "Execute concurrent builds if necessary", and "Restrict where this project can be run". At the bottom are "Save" and "Apply" buttons.

The job is created successfully. Now, configure it to meet the project's requirements.

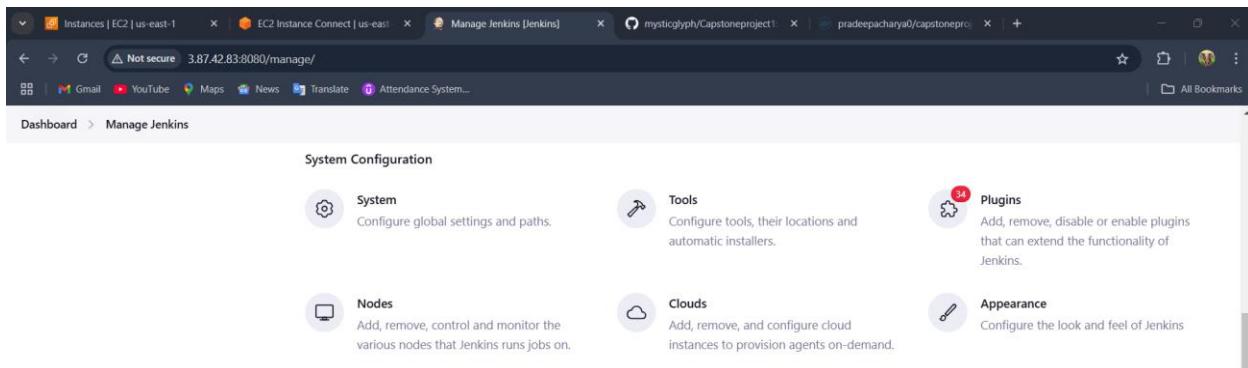
Downloading and installing essential plugins is a prerequisite for configuring the job.

Installing plugins:

This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.

→ **CloudBees Docker Build and Publish.**

Manage Jenkins -> plugins.



The screenshot shows the "System Configuration" section of the Jenkins management interface. It includes links for "System", "Tools", "Nodes", "Clouds", "Plugins", and "Appearance". The "Plugins" link is highlighted with a red circle containing the number 34, indicating available updates or new plugins.

The screenshot shows the Jenkins CloudBees Docker Build and Publish plugin page. At the top, there's a navigation bar with links like Blog, Success Stories, Contributor Spotlight, Documentation, Plugins, Community, Subprojects, Security, About, and Download. Below the navigation is the plugin's title, "CloudBees Docker Build and Publish". A "Documentation" tab is selected. Other tabs include Releases, Issues, Dependencies, and Health Score. A button for "How to install" is visible. The main content area displays the plugin's version (1.4.0), release date (2 years ago), and compatibility (Requires Jenkins 2.263.4). It also shows that it's installed on 3.14% of controllers. A progress bar indicates its popularity. A "View detailed version information" link is present. To the right, sections for "Links" (GitHub, Open issues, Report an issue, Javadoc) and "Labels" (Build Tools, docker) are shown. A "Maintainers" section is also listed.

The screenshot shows the Jenkins Manage Jenkins > Plugins page. The search bar at the top is set to "docker". The results list several plugins:

- docker-build-step** 2.12 (Build Tools, docker)
This plugin allows to add various docker commands to your job as build steps.
Warning: This plugin version may not be safe to use. Please review the following security notices:
 - CSRF vulnerability and missing permission check6 mo 1 day ago
- CloudBees Docker Build and Publish** 1.4.0 (Build Tools, docker)
This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.
2 yr 3 mo ago
- Amazon ECR** 1.136.v914ea_5948634 (aws)
This plugin generates Docker authentication token from Amazon Credentials to access Amazon ECR.
This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.
5 mo 0 days ago
- Docker Compose Build Step** 1.0 (Docker Compose plugin for Jenkins)
6 yr 4 mo ago

The screenshot shows the Jenkins 'Plugins' management interface. On the left, a sidebar lists 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress'. The 'Download progress' item is highlighted with a grey background. The main content area is titled 'Download progress' and shows a 'Preparation' section with three items: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success', each accompanied by a blue circular icon with a dot. Below this is a table of plugin updates:

Plugin	Status
Authentication Tokens API	Pending
Docker Commons	Pending
JavaMail API	Pending
SSH server	Pending
CloudBees Docker Build and Publish	Pending
Loading plugin extensions	Pending

At the bottom of the page, there are two links: 'Go back to the top page' (with a note '(you can start using the installed plugins right away)') and 'Restart Jenkins when installation is complete and no jobs are running' (with a checkbox input).

REST API Jenkins 2.46.2

This screenshot is identical to the one above, showing the Jenkins 'Plugins' management interface with the 'Download progress' section highlighted. The preparation steps and the list of pending plugin updates are the same.

REST API Jenkins 2.46.2

Plugin is downloaded.

Downloading docker—build-step plugin:

This plugin allows adding various Docker commands into your Jenkins Freestyle job as a build step.

The screenshot shows the Jenkins plugin page for 'docker-build-step'. At the top, there's a warning message about an unresolved security vulnerability: 'CSRF vulnerability and missing permission check'. Below this, the plugin's name 'Docker Build Step Plugin' is displayed, along with its description: 'This plugin allows adding various Docker commands into your Jenkins Freestyle job as a build step.' A 'Plugin Information' section follows, containing links to Documentation, Releases, Issues, Dependencies, and Health Score. To the right, there's a summary box for 'Version: 2.12': 'Released: 6 months ago', 'Requires Jenkins 2.387.3', and 'ID: docker-build-step'. It also states 'Installed on 4.94% of controllers' with a progress bar and a link to 'View detailed version information'. Further down are sections for 'Links' (GitHub, Open issues (Jira), Report an issue (Jira), Extension Points, Javadoc) and 'Labels' (Build Tools, docker). A 'Maintainers' section is partially visible at the bottom.

Plugin currently support following commands:

- commit changes in specified container
- create new container from image
- create image from Dockerfile
- create exec command
- kill container(s)
- pull image from a repository
- push image to a repository
- remove container(s)
- remove all containers
- restart container(s)
- start container(s)
- stop container(s)
- stop all containers
- start/stop all containers created from specified image.

The screenshot shows the Jenkins plugin manager interface. The left sidebar has tabs for 'Updates' (34), 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. The main area displays a search bar with 'docker' and a list of available plugins:

- Docker API** 3.4.0-94.v65ced49b_a_7d5: Pipeline, DevOps, Deployment, docker. Last updated 6 mo 6 days ago. Status: Available.
- docker-build-step** 2.12: Build Tools, docker. Last updated 6 mo 1 day ago. Status: Available. A warning message states: "Warning: This plugin version may not be safe to use. Please review the following security notices:
 - CSRF vulnerability and missing permission check"
- Amazon ECR** 1.136.v9f14ea_5948634: aws. Last updated 5 mo 0 days ago. Status: Available. A message states: "This plugin generates Docker authentication token from Amazon Credentials to access Amazon ECR."

The screenshot shows the Jenkins plugin manager interface with the 'Download progress' tab selected. The left sidebar has tabs for 'Updates' (34), 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. The main area displays a 'Download progress' section:

Preparation

- Checking internet connectivity
- Checking update center connectivity

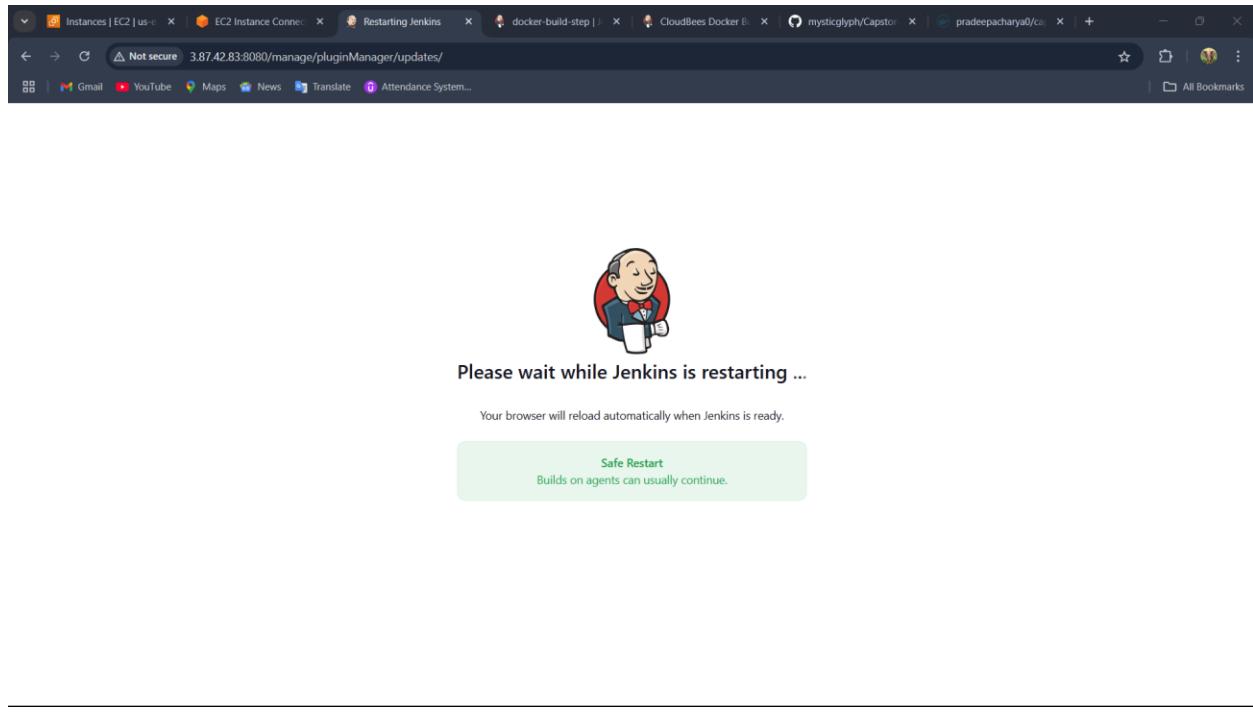
Download progress

Plugin	Status
Javadoc	Pending
JSch dependency	Pending
Maven Integration	Pending
Apache HttpComponents Client 5.x API	Pending
Docker API	Pending
docker-build-step	Pending
Loading plugin extensions	Pending

→ [Go back to the top page](#)
(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

REST API Jenkins 2.462.2



Plug in is installed.

A screenshot of the Jenkins Plugins page at 3.87.42.83:8080/manage/pluginManager/updates/. The top navigation bar shows "Jenkins" and the user "admin". The page has two main sections: "Plugins" on the left with links for "Updates", "Available plugins", "Installed plugins", and "Advanced settings"; and "Download progress" on the right. Under "Download progress", there is a link to "Go back to the top page" with the note "(you can start using the installed plugins right away)". At the bottom, there is a checkbox for "Restart Jenkins when installation is complete and no jobs are running". The footer includes links for "REST API" and "Jenkins 2.462.2".

Necessary plugins have been successfully installed and are ready to be utilized in the jobs.

We must now configure the first job to handle code building, specifying its execution on Manage Node 1.

The screenshot shows the Jenkins dashboard at the URL 3.87.42.83:8080. The 'Job1-build' project is listed in the main table with the following details:

S	W	Name	Last Success	Last Failure	Last Duration
...	...	Job1-build	N/A	N/A	N/A

On the left sidebar, under 'Build Executor Status', there is a table showing build nodes:

Node	Status
Built-In Node	Idle
prod	Idle
test	Idle

Let's begin the job configuration process.

The screenshot shows the configuration page for the 'Job1-build' project at the URL 44.203.122.180:8080/job/job1-build/. The left sidebar includes options like 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. The 'Configure' option is highlighted. The main content area displays the 'Job1-build' configuration with sections for 'Build History' (which shows 'No builds') and 'Permalinks'.

Click on the configuration option to start the setup.

Now we need to mention the slave node where this job need to execute.

The screenshot shows the Jenkins configuration page for a job named 'Job1-build'. In the 'General' section, under 'Label Expression', the value 'test' is entered. This indicates that the job will only be executed on nodes labeled 'test'.

Under Source Code Management, choose the 'Git' option.

The screenshot shows the Jenkins configuration page for the same job. In the 'Source Code Management' section, the 'Git' option is selected. A red error message 'Please enter Git repository.' is displayed above the 'Repository URL' field, which is currently empty.

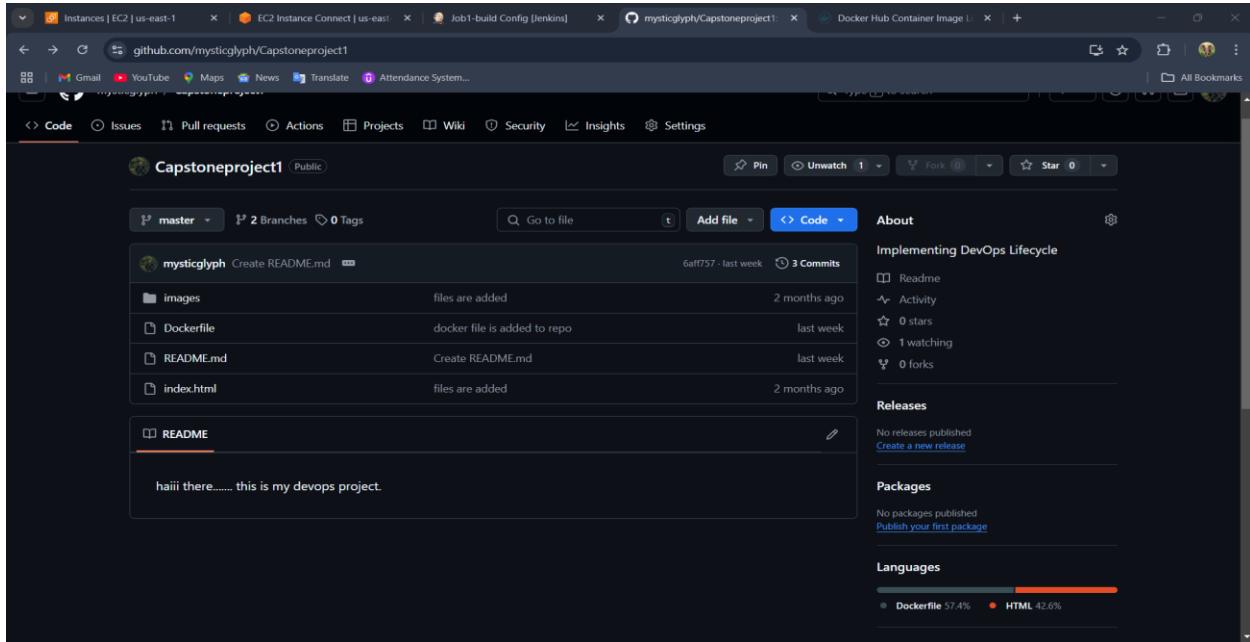
The screenshot shows the Jenkins job configuration interface for a job named "Job1-build". The "Source Code Management" section is open, specifically the "Git" tab. The "Repository URL" field contains the value `https://github.com/mysticglyph/Capstoneproject1.git`. The "Credentials" dropdown is set to "- none -". There is an "Advanced" button and a "Save" button at the bottom.

Because the Git repository is public, providing credentials is not necessary.

We need to configure the branch so that commits made to it result in an automatic trigger.

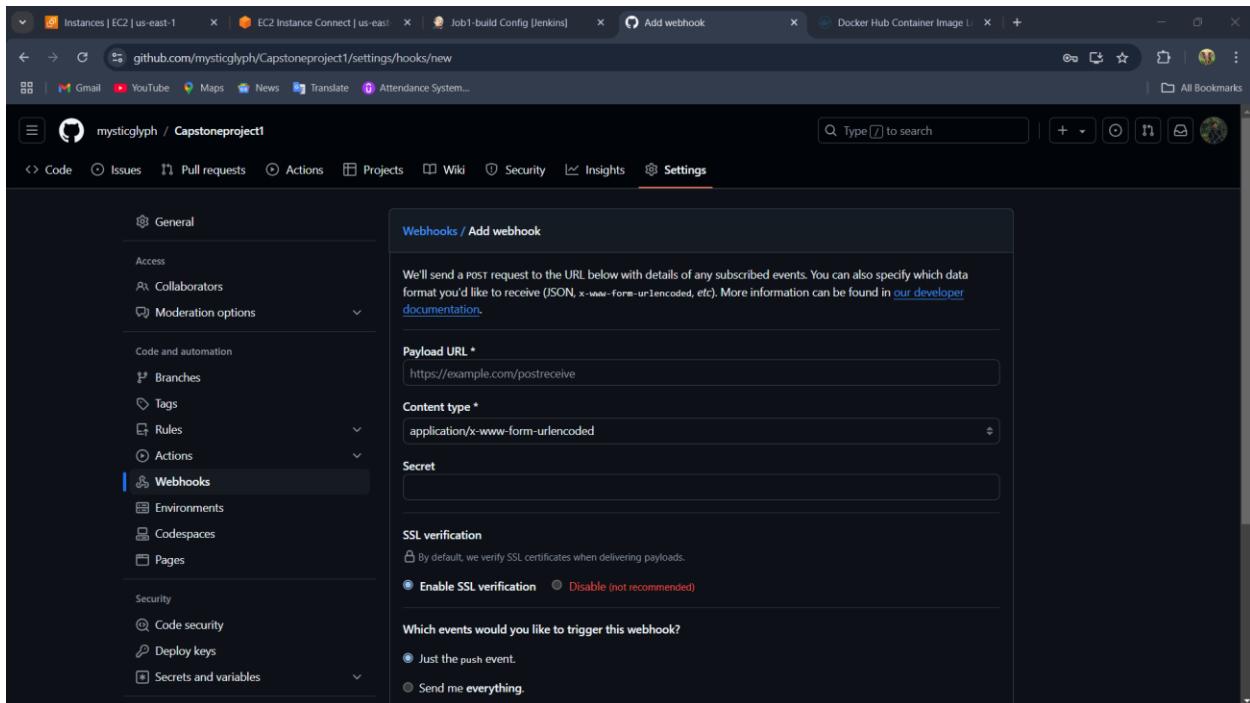
The screenshot shows the Jenkins job configuration interface for a job named "Job1-build". The "Branches to build" section is open, showing a "Branch Specifier" field containing `*/master`. There is an "Add Branch" button and a "Repository browser" dropdown set to "(Auto)".

Creating a webhook in the GitHub repository is required before proceeding.

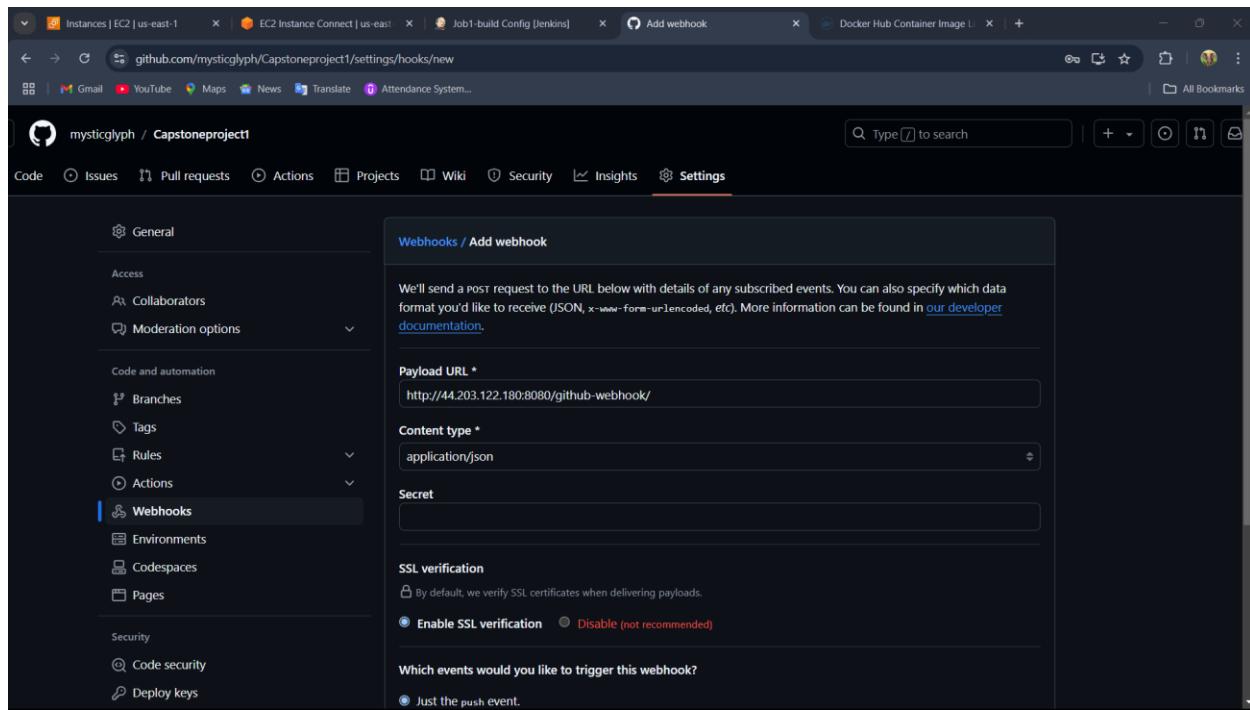


The screenshot shows a GitHub repository named 'Capstoneproject1'. The repository has 2 branches (master and dev) and 0 tags. The README file contains the text: "haiii there..... this is my devops project." The repository has 3 commits from 'mysticglyph' and 0 stars. It also includes sections for About, Releases, Packages, and Languages.

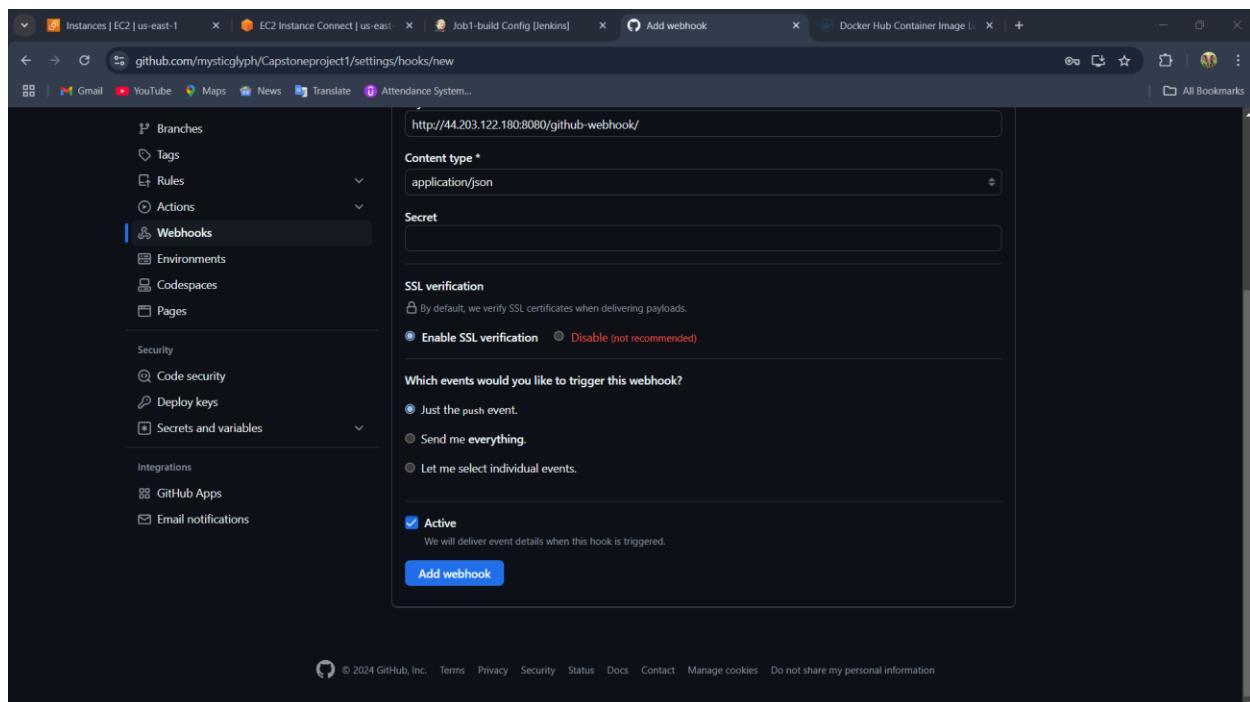
Settings -> webhooks -> add webhooks



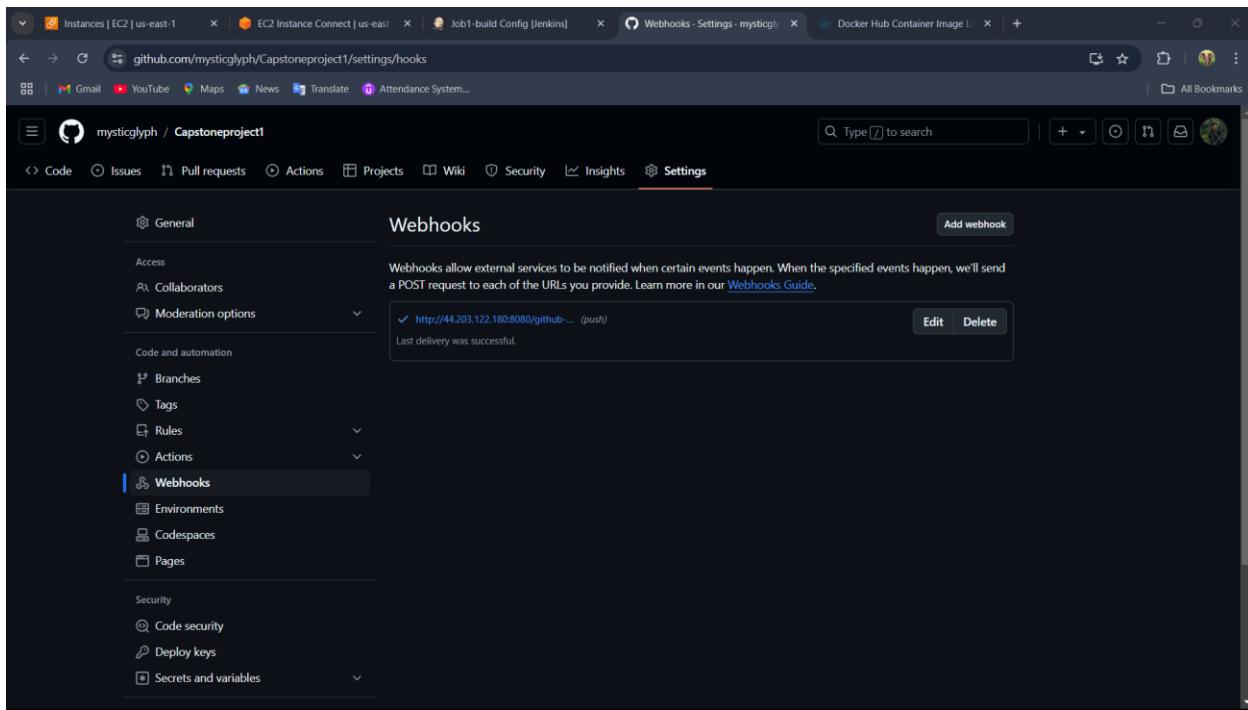
The screenshot shows the GitHub Settings page for the 'Capstoneproject1' repository. The left sidebar is expanded to show the 'Webhooks' section under 'Code and automation'. The main panel displays the 'Add webhook' form. The 'Payload URL' field is set to 'https://example.com/postreceive', 'Content type' is set to 'application/x-www-form-urlencoded', and 'SSL verification' is enabled. The 'Which events would you like to trigger this webhook?' section has 'Just the push event.' selected.



"Specify 'application/json' as the Content-Type for the webhook payload.



Click on Add webhook.



The webhook has been created successfully and is now ready to use.

Select the webhook trigger option within the job configuration settings.

A screenshot of a web browser showing the Jenkins job configuration page for 'Job1-build Config'. In the 'Build Triggers' section, the 'GitHub hook trigger for GITScm polling' checkbox is checked. Other options like 'Trigger builds remotely' and 'Build periodically' are unchecked. Below this, the 'Build Environment' section is visible, containing three unchecked checkboxes for workspace deletion, secret use, and timestamp addition.

Now, we need to select the option for build steps.

In the build steps, we need to choose the "Execute Shell" option.

The screenshot shows the AWS Lambda configuration interface. The left sidebar has sections: General, Source Code Management, Build Triggers, **Build Environment**, Build Steps, and Post-build Actions. The 'Build Environment' section is selected. A modal window titled 'Build Environment' is open, listing various options under a 'Filter' dropdown. The 'Execute shell' option is highlighted. Other options include Docker Build and Publish, Execute Docker command, Execute Windows batch command, Invoke Ant, Invoke Gradle script, Invoke top-level Maven targets, Run with timeout, and Set build status to "pending" on GitHub commit. At the bottom of the modal is a button labeled 'Add build step ^'.

The screenshot shows the AWS Lambda configuration interface in a browser window. The URL is 54.158.125.236:8080/job/job1-build/configure. The left sidebar has sections: General, Source Code Management, Build Triggers, Build Environment, **Build Steps**, and Post-build Actions. The 'Build Steps' section is selected. A modal window titled 'Build Steps' is open, showing a single 'Execute shell' step. The 'Command' field is empty. Below it is an 'Advanced' dropdown and a 'Save' button at the bottom. The 'Post-build Actions' section below it has a 'Add post-build action ^' button.

In the "Execute Shell" option, add commands to build the Docker image, start the Docker container, and map port 80 on your host to port 80 in the container.

The screenshot shows a web browser window with multiple tabs open, including 'Instances | EC2', 'EC2 Instance Co...', 'Job1-build Config', 'YouTube', 'Webhooks - Set...', and 'ChatGPT'. The main content area is titled 'Configuration' under 'Job1-build'. On the left, a sidebar lists 'Configure' sections: General, Source Code Management, Build Triggers, **Build Environment**, Build Steps, and Post-build Actions. The 'Build Environment' section is currently selected. In the main pane, there are several configuration options: 'Map port 80 for persistent build status' (unchecked), 'Terminate a build if it's stuck' (unchecked), and 'With Ant' (unchecked). Below these is a 'Build Steps' section containing a single 'Execute shell' step. The 'Execute shell' step has a 'Command' field containing the following two lines of code:

```
sudo docker build . -t testing
sudo docker run -it -d -p 80:80 testing
```

Below the command field are 'Advanced' and 'Add build step' dropdown menus. At the bottom of the configuration page are 'Save' and 'Apply' buttons.

Once the job configuration is saved, proceed to run the job.

The screenshot shows a continuation of the configuration interface. It features a 'Post-build Actions' section with a 'Add post-build action' dropdown menu. Below this are 'Save' and 'Apply' buttons. The background shows the same 'Build Environment' configuration from the previous screenshot.

The screenshot shows a browser window with multiple tabs open. The active tab is 'Job1-build [Jenkins]' at the URL 54.158.125.236:8080/job/Job1-build/. The page title is 'Jenkins'. The main content area is titled 'Job1-build' and contains the following sections:

- Status**
- Permalinks**
- Build History**: Shows 'No builds'.
- REST API** and **Jenkins 2.462.2** are listed at the bottom right.

To build the job, we need to click on the "Build" option.

This screenshot shows the same Jenkins Job1-build dashboard as the previous one, but with a different visual style. The background is white, and the navigation links are displayed in a dark grey bar. The main content area has a light grey background. The sections and options are identical to the first screenshot.

The screenshot shows the Jenkins Job1-build dashboard. On the left, a sidebar lists project management options: Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, and Rename. The main area displays the job status as "Job1-build" with a green checkmark icon. Below this is a "Permalinks" section listing recent builds. A "Build History" card shows two builds: #2 (Dec 22, 2024, 4:00PM) and #1 (Dec 22, 2024, 3:01PM). At the bottom of the card are links for "Atom feed for all" and "Atom feed for failures".

The job has been successfully executed.

The screenshot shows an AWS CloudShell session titled "Instances | EC2 | us-east-1 | console.aws.amazon.com". The session output shows the following terminal session:

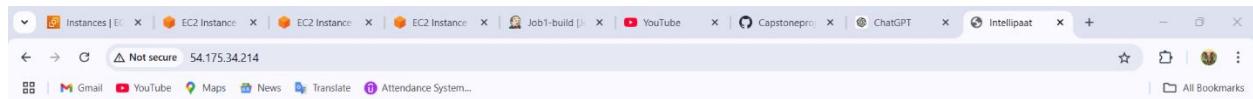
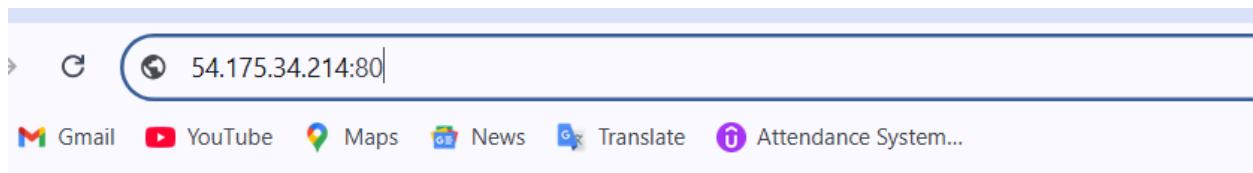
```

Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
231 updates can be applied immediately.
81 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Dec 22 16:46:38 2024 from 18.206.107.28
ubuntu@ip-172-31-93-253:~$ sudo su
root@ip-172-31-93-253:/home/ubuntu# ls
jenkins
root@ip-172-31-93-253:/home/ubuntu# cd jenkins
bash: cd: jenkins: No such file or directory
root@ip-172-31-93-253:/home/ubuntu# cd jenkins
bash: cd: jenkins: No such file or directory
root@ip-172-31-93-253:/home/ubuntu# cd jenkins
root@ip-172-31-93-253:/home/ubuntu/jenkins# ls
remote  remoting_jar  workspace
root@ip-172-31-93-253:/home/ubuntu/jenkins# cd workspace
root@ip-172-31-93-253:/home/ubuntu/jenkins/workspace# ls
Job1-build
root@ip-172-31-93-253:/home/ubuntu/jenkins/workspace# cd Job1-build
bash: cd: job1-build: No such file or directory
root@ip-172-31-93-253:/home/ubuntu/jenkins/workspace# cd Job1-build
root@ip-172-31-93-253:/home/ubuntu/jenkins/workspace/Job1-build# ls
nodevfile  README.md  images  index.html
root@ip-172-31-93-253:/home/ubuntu/jenkins/workspace/Job1-build#

```

Below the terminal output, the session summary shows "i-Ocd0f347cb719fc81 (Manage_Node1)" and "PublicIPs: 54.175.34.214 PrivateIPs: 172.31.93.253". The footer includes links for CloudShell, Feedback, and cookie preferences.



GitHub

As observed, the first job executed successfully. When a commit is made to the master branch, it triggers testing and then pushes the changes to production.

So now, we need to create Job 2 and configure it accordingly.

New Item

Enter an item name

job2

Select an item type

 **Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 **Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

 **Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

Copy from



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in origin repository.



Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

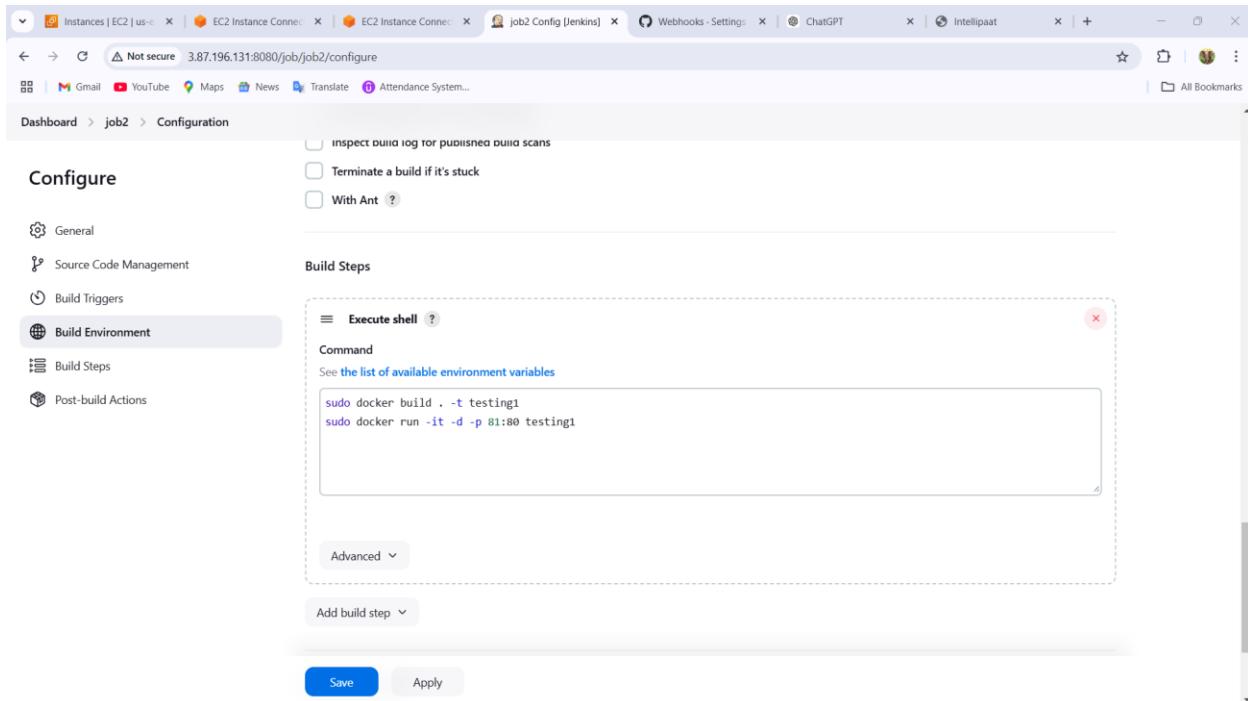
If you want to create a new item from other existing, you can use this option:

Copy from

Job1-build

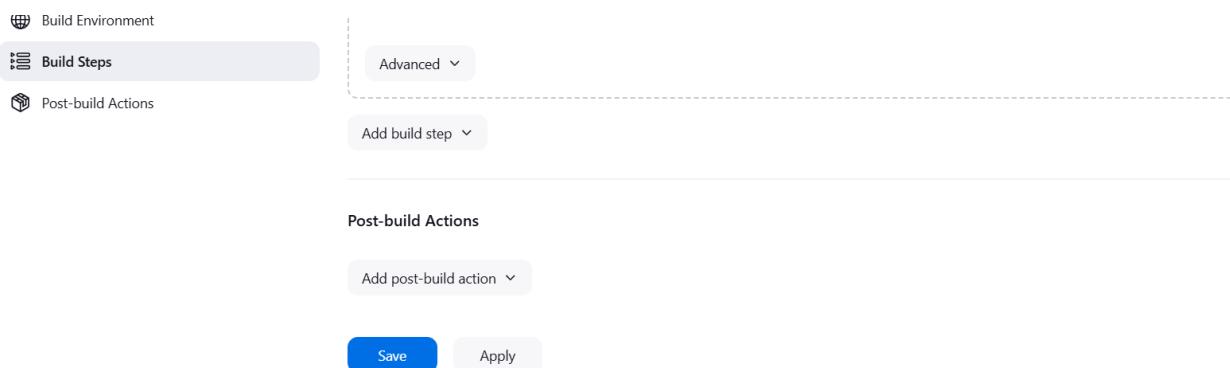
OK

The same modifications must be reflected in the Docker build command. The port number needs to be updated.



A screenshot of a web browser displaying the Jenkins job configuration page for 'job2'. The URL is 3.87.196.131:8080/job/job2/configure. The left sidebar shows navigation options: Dashboard, job2, Configuration, General, Source Code Management, Build Triggers, Build Environment (selected), Build Steps, and Post-build Actions. The main content area is titled 'Configure' and shows 'Build Steps'. A single step is listed: 'Execute shell'. The 'Command' field contains the following Docker commands:
`sudo docker build . -t testing1
sudo docker run -it -d -p 81:80 testing1`

The rest of the configuration settings will remain as they are.



A screenshot of the Jenkins job configuration page for 'job2'. The left sidebar shows 'Build Environment' (selected), 'Build Steps' (selected), and 'Post-build Actions'. The main content area shows the 'Build Steps' section with an 'Advanced' dropdown and an 'Add build step' button. Below it is the 'Post-build Actions' section with an 'Add post-build action' button. At the bottom are 'Save' and 'Apply' buttons.

Save the job configurations.

The screenshot shows the Jenkins interface for a job named 'job2'. The top navigation bar includes links for Instances, EC2 Instance Connect, Webhooks, ChatGPT, and Intellipaat. The main content area displays the 'Status' tab for 'job2'. On the left, a sidebar lists options like Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, and Rename. The central area shows a 'Build History' section with a 'trend' dropdown set to 'No builds'. At the bottom, there are links for Atom feed for all and Atom feed for failures.

The job has been configured successfully.

This screenshot shows the same Jenkins interface for 'job2' after configuration. The 'Status' tab now displays a green checkmark icon. The 'Build History' section shows a list of four successful builds: 'Last build (#1), 35 sec ago', 'Last stable build (#1), 35 sec ago', 'Last successful build (#1), 35 sec ago', and 'Last completed build (#1), 35 sec ago'. The date and time of the last build are also listed as 'Dec 25, 2024, 7:42 AM'. The rest of the interface remains consistent with the first screenshot.

Job execution was completed successfully.

We now need to update our security groups to include port number 81 for Job 2

Inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-0ab0422d73755b793	SSH	TCP	22	Custom	<input type="text"/> <input type="button" value="Delete"/>
sgr-05c0d6be0ad061840	HTTP	TCP	80	Custom	<input type="text"/> <input type="button" value="Delete"/>
sgr-0651b6994eef7cad	Custom TCP	TCP	81	Anywhere	<input type="text"/> <input type="button" value="Delete"/>

Add rule

⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

Save the changes.

Now access Job 2 from the browser using the IP address.

→ <ip>:81

Instances | EC2 | us-east-1

EC2 Instance Connect | us-

EC2 Instance Connect

→

Gmail YouTube Maps News Translate Attendance System...



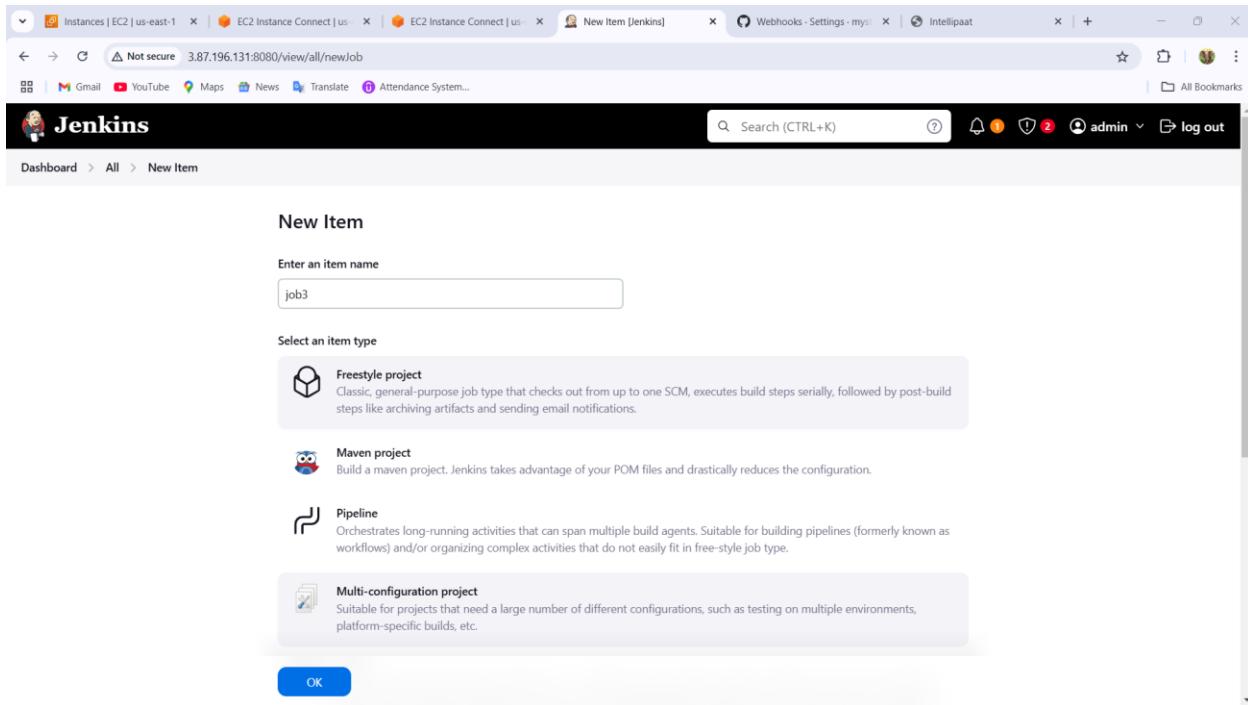
Hello world!



GitHub

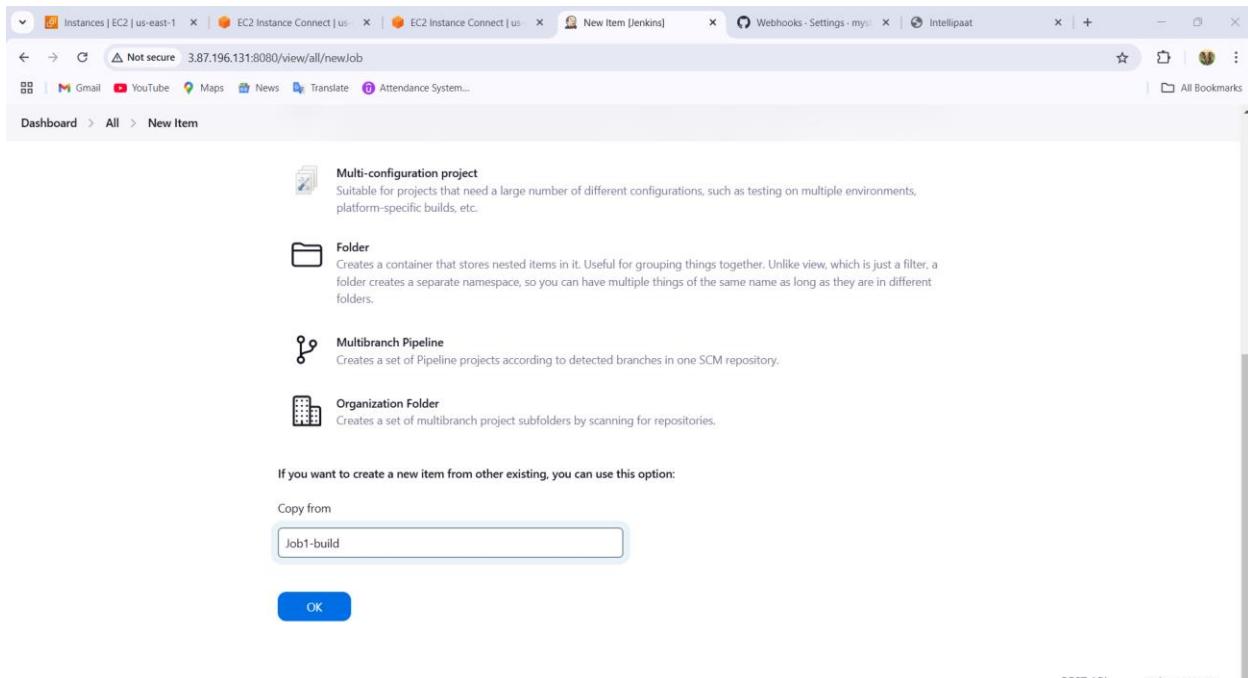
Job 2 has been successfully executed and is designed to test the product on commits to the develop branch, without deploying to production.

Job 3 must be created and configured to deploy changes to production after a commit is made to the master branch, with testing handled in Job 1.



The screenshot shows the Jenkins 'New Item' creation interface. In the top navigation bar, there are several tabs: Instances | EC2 | us-east-1, EC2 Instance Connect | us..., EC2 Instance Connect | us..., New Item [Jenkins], Webhooks - Settings - my..., and Intellipaat. Below the tabs, the URL is 3.87.196.131:8080/view/all/newJob. The main content area has a title 'New Item'. A search bar says 'Search (CTRL+K)'. On the right, there are icons for notifications, security, and user admin. The page shows the 'job3' name in the item name input field. Under 'Select an item type', there are five options: 'Freestyle project' (selected), 'Maven project', 'Pipeline', 'Multi-configuration project', and 'Folder'. Each option has a brief description. At the bottom is a blue 'OK' button.

Copy from



This screenshot shows the 'Copy from' section of the Jenkins 'New Item' creation page. It lists four items: 'Multi-configuration project', 'Folder', 'Multibranch Pipeline', and 'Organization Folder', each with a brief description. Below this, a note says 'If you want to create a new item from other existing, you can use this option:'. A 'Copy from' input field contains 'Job1-build'. At the bottom is a blue 'OK' button. The footer of the page includes links for 'REST API' and 'Jenkins 2.462.2'.

Save the job

Configure the job and then proceed to save the configuration.

Now we need to restrict the job to run only on the prod node, which is managed by Node 2.

The screenshot shows the Jenkins job configuration page for 'job3'. In the 'General' section, the 'Restrict where this project can be run' checkbox is checked, and the 'Label Expression' field contains 'prod'. A note below states: 'Label prod matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.' Below this, the 'Source Code Management' section is set to 'None'. At the bottom are 'Save' and 'Apply' buttons.

Now, we need to change the name of the Docker image and container from 'testing' to 'production' and save the job.

The screenshot shows the Jenkins job configuration page for 'job3'. In the 'Build Steps' section, there is one step named 'Execute shell' with the command:

```
sudo docker build . -t product  
sudo docker run -it -d -p 80:80 product
```

. Below this, the 'Post-build Actions' section is empty. At the bottom are 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins configuration interface for a job named 'job3'. The 'Build Steps' section is selected. Under 'Command', the following Docker commands are defined:

```
sudo docker build . -t product
sudo docker run -it -d -p 80:80 product
```

Below the command section, there is an 'Advanced' dropdown and a 'Post-build Actions' section with a 'Save' button.

Apply and save the job.

The screenshot shows the Jenkins dashboard for the 'job3' project. The 'Status' tab is selected. The 'Permalinks' section includes links for 'Build History', 'trend', 'Atom feed for all', and 'Atom feed for failures'. Below this, it says 'No builds'.

Job is successfully created and now job build is required.

GitHub Hook Log

Rename

Build History

trend

Filter...

#1

Dec 25, 2024, 8:28 AM

Atom feed for all Atom feed for failures

Build is successful.

Jenkins

Search (CTRL+K)

admin log out

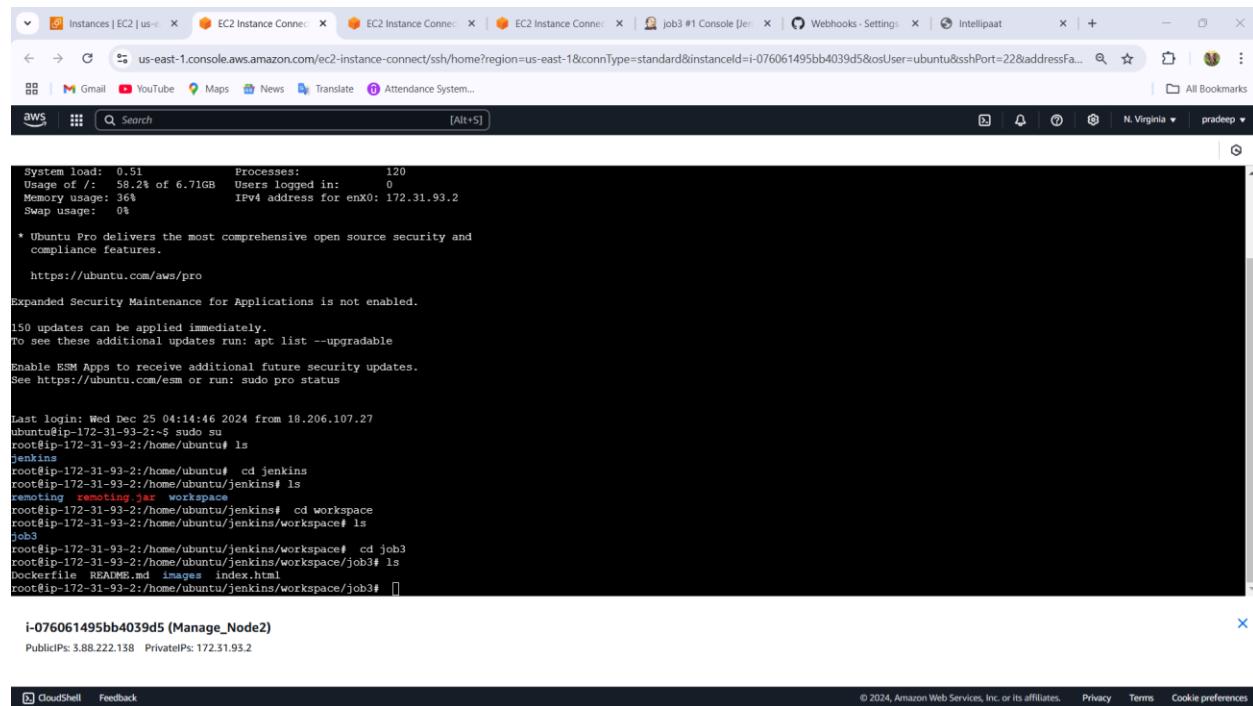
Dashboard > job3 > #1 > Console Output

Console Output

Status Download Copy View as plain text

Started by user admin
Running as SYSTEM
Building remotely on prod in workspace /home/ubuntu/jenkins/workspace/job3
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/mysticlyph/Capstoneproject1.git
> git init /home/ubuntu/jenkins/workspace/job3 # timeout=10
Fetching upstream changes from https://github.com/mysticlyph/Capstoneproject1.git
> git --version # timeout=10
> git -version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/mysticlyph/Capstoneproject1.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/mysticlyph/Capstoneproject1.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 4cb814096c2bab7a2656f5c67f422ceaf758a813 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 4cb814096c2bab7a2656f5c67f422ceaf758a813 # timeout=10
Commit message: "new Dockerfile update"
First time build. Skipping changelog.
[job3] \$ /bin/sh -xe /tmp/jenkins3149187914969165743.sh
+ sudo docker build . -t product
#0 building with "default" instance using docker driver

Check if the job has been executed correctly on the server 'Manage Node 2'.



```
System load: 0.51      Processes: 120
Usage of /: 58.2% of 6.71GB  Users logged in: 0
Memory usage: 36%          IPv4 address for enx0: 172.31.93.2
Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.
https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

150 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

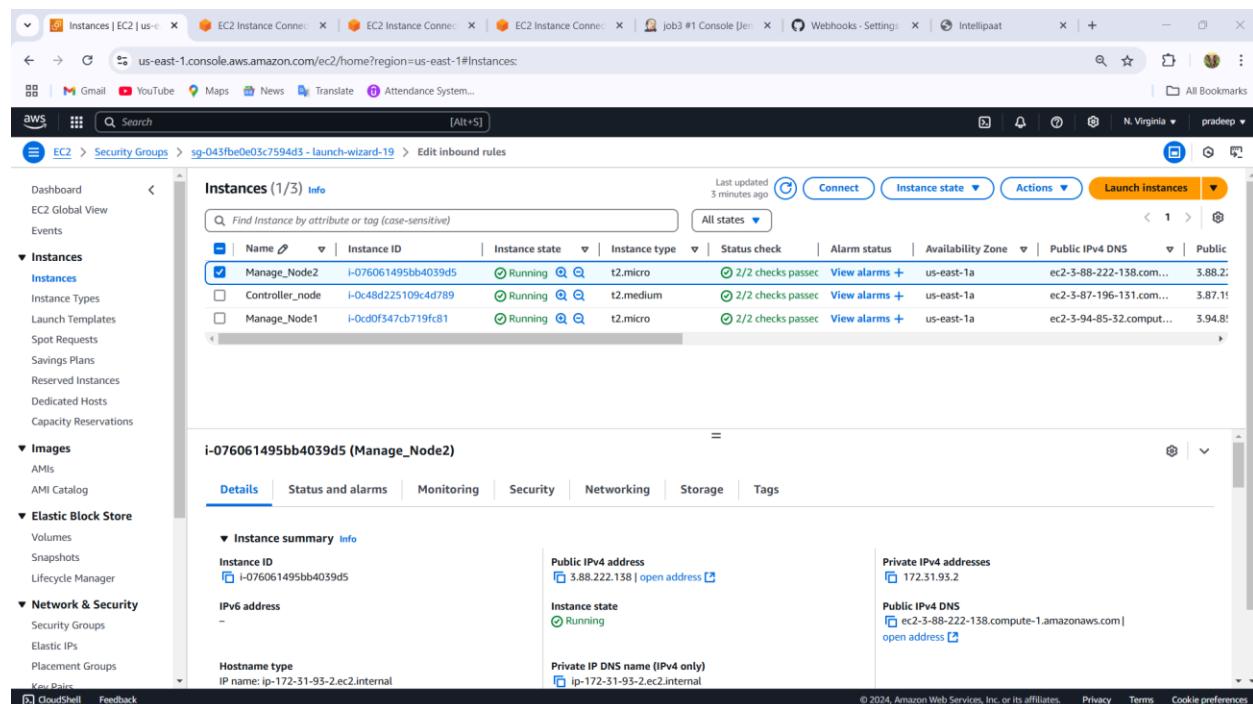
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Wed Dec 25 04:14:46 2024 from 18.206.107.27
ubuntu@ip-172-31-93-2:~$ sudo su
root@ip-172-31-93-2:~/home/ubuntu#
jenkins
root@ip-172-31-93-2:~/home/ubuntu# cd jenkins
root@ip-172-31-93-2:~/home/ubuntu/jenkins# ls
remoting remotng_jar workspace
root@ip-172-31-93-2:~/home/ubuntu/jenkins# cd workspace
root@ip-172-31-93-2:~/home/ubuntu/jenkins/workspace# ls
job3
root@ip-172-31-93-2:~/home/ubuntu/jenkins/workspace# cd job3
root@ip-172-31-93-2:~/home/ubuntu/jenkins/workspace/job3# ls
Dockerfile REAME.md images index.html
root@ip-172-31-93-2:~/home/ubuntu/jenkins/workspace/job3# 
```

i-076061495bb4039d5 (Manage_Node2)
PublicIPs: 3.88.222.138 PrivateIPs: 172.31.93.2

Job is exicuted successfully.

Now, we need to open port 80 in the security group of the Manage Node server.



Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
Manage_Node2	i-076061495bb4039d5	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-3-88-222-138.com...	3.88.2;
Controller_node	i-0c48c225109cd789	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1a	ec2-3-87-196-131.com...	3.87.1!
Manage_Node1	i-0cd0f347cb719fc81	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-3-94-85-32.comput...	3.94.8!

i-076061495bb4039d5 (Manage_Node2)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID i-076061495bb4039d5	Public IPv4 address 3.88.222.138 open address	Private IPv4 addresses 172.31.93.2
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-88-222-138.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-93-2.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-93-2.ec2.internal	

i-076061495bb4039d5 (Manage_Node2)

Security

Inbound rules

Name	Protocol	Port range	Source
sgr-0dc63993dbd24ef43	TCP	22	0.0.0.0/0

Goto → security groups.

sg-04c9cf18f50f96388 - launch-wizard-20

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0dc63993dbd24ef43	IPv4	SSH	TCP	22	0.0.0.0/0

→ Edit in-boundrules.

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0dc63993d8d24ef43	SSH	TCP	22	Custom	

[Add rule](#)

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

Click on 'Add Rule' and then add port 80.

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0dc63993d8d24ef43	SSH	TCP	22	Custom	
-	Custom TCP	TCP	80	Anywh...	

[Add rule](#)

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

The screenshot shows the AWS Security Groups console. A green success message at the top states: "Inbound security group rules successfully modified on security group (sg-04c9cf18f50f96388 | launch-wizard-20) > Details". Below this, the security group details are shown: name (sg-04c9cf18f50f96388), ID (sg-04c9cf18f50f96388), description (launch-wizard-20 created 2024-08-31T13:51:04.904Z), owner (626130759947), VPC ID (vpc-0ad7d342053a758cb). The Inbound rules section shows two entries:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-081f559f518310571	IPv4	HTTP	TCP	80	0.0.0.0/0	
sgr-0dc65995df8d24ef43	IPv4	SSH	TCP	22	0.0.0.0/0	

Port has been successfully added. Now, we need to attempt accessing the container through the browser

The screenshot shows a web browser with the URL 3.88.222.138:80. The page is currently blank, indicating that the connection is not yet successful or the page has not loaded.

The screenshot shows a web browser displaying the GitHub homepage. The title "Hello world!" is visible above the GitHub logo, which features the iconic black cat icon. The page content includes the GitHub logo and the text "GitHub".

Now we can confirm that a commit made to the master branch will trigger a push to production.

Project Conclusion: DevOps Lifecycle Implementation for Abode Software

In this project, I successfully implemented a complete DevOps lifecycle for Abode Software, a product-based company. The primary objective was to streamline the development and deployment process using industry-standard DevOps practices. This included configuring infrastructure, setting up Git workflows, automating builds with AWS CodeBuild, containerizing the application using Docker, and defining Jenkins pipelines for continuous integration and delivery (CI/CD).

The project involved the following key tasks:

Configuration Management: Automated the installation of necessary software using Ansible, ensuring a consistent environment across all nodes.

Git Workflow: Implemented a branching strategy with master and develop branches to manage code versions efficiently.

Automated Builds: Integrated AWS CodeBuild to automatically trigger builds and tests based on commits to specific branches.

Containerization: Used Docker to containerize the application, making deployments faster and more reliable.

CI/CD Pipeline: Created a multi-stage Jenkins pipeline with jobs for building, testing, and deploying the application.

Through this project, I gained hands-on experience with core DevOps tools like Git, Jenkins, Docker, AWS CodeBuild, and Ansible. The implementation of an automated CI/CD pipeline enhanced the development process by reducing manual intervention, increasing productivity, and ensuring faster delivery of high-quality software.

The project also helped me improve my troubleshooting skills, as I faced challenges related to configuration errors, containerization issues, and automating workflows, which I successfully resolved. Overall, this project was a significant learning experience that demonstrated my ability to apply DevOps principles in a real-world scenario.

