

Case study

Integration of Devops tools with Jenkins

NAME: PRADEEP A K

EMAIL: KPRADEEPACHARYA0@GMAIL.COM

You have been hired as a DevOps Engineer in XYZ Software Company. They want to implement a CI/CD pipeline in their company. You have been asked to implement this lifecycle as fast as possible. As this is a product-based company, their product is available on this GitHub link.

<https://github.com/hshar/website.git>

Following are the specifications of the Continuous integration:

1. Git Workflow has to be implemented
2. Code Build should automatically be triggered once commit is made to master branch
or develop branch.

If commit is made to master branch, build and publish website on port 82. If commit is made to develop branch, just build the product, do not publish.

3. Create a pipeline for the above tasks.
4. Create a container with Ubuntu and apache installed in it and use that container to
build the code and the code should be on '/var/www/html'.

Introduction

This case study focuses on implementing a Continuous Integration (CI) pipeline using Jenkins for a product hosted on GitHub. The objective is to establish an efficient CI workflow that integrates with the existing Git repository. Key tasks include implementing a Git workflow, automating builds triggered by commits to specific branches, and creating a Docker container with Ubuntu and Apache to manage the build process. Commits to the master branch will trigger a full build and deployment, while commits to the develop branch will result in a build only, without deployment.

Project Background

The product, hosted on GitHub, is regularly updated by a team of developers. However, the current process of building and deploying code is manual, which leads to several issues:

- Slow Development: Manually building and deploying code takes a lot of time, slowing down the development process.
- Inconsistent Environments: Developers sometimes face problems because the build environments aren't always the same.
- Human Errors: Manual processes are more likely to have mistakes, which can cause bugs and other issues.

To solve these problems, a Continuous Integration and Continuous Deployment (CI/CD) pipeline using Jenkins is being implemented. This will automate the build, test, and deployment processes, making the workflow faster, more consistent, and less prone to errors.

Objectives

The primary objectives of this case study are:

- Implement Git Workflow: Establish a structured Git workflow that integrates with Jenkins, ensuring that code changes are managed effectively across different branches.
- Automate Build Triggers: Configure Jenkins to automatically trigger builds based on commits to specific branches:
- Master Branch: Trigger a full build and deploy the product on port 82.
- Develop Branch: Trigger a build without deploying the product.
- Create a Jenkins Pipeline: Develop a Jenkins pipeline that automates the entire process, from code integration to testing and deployment.

Containerize the Build Environment: Use Docker to create a container with Ubuntu and Apache, which will serve as the build environment. Ensure that the code is built and placed in the /var/www/html directory within the container.

Resources Used

The following tools and technologies were utilized in the implementation of the CI/CD pipeline:

- Version control system used to manage and track changes in the codebase, ensuring effective collaboration among developers.
- AWS: Cloud platform used to host the Jenkins server and other necessary infrastructure for the CI/CD pipeline.
- Jenkins: Automation server used to create and manage the CI/CD pipeline, automating the build, test, and deployment processes.
- Docker: Containerization tool used to create a consistent and isolated build environment, ensuring that the code is built and deployed in a controlled manner.

Implementation Steps:

Setup EC2 Instance:

The screenshot shows the AWS EC2 Dashboard for the us-east-1 region. The left sidebar includes links for EC2 Global View, Events, Console-to-Code, Instances, Images, and Elastic Block Store. The main content area displays a summary of resources: 0 Instances (running), 0 Auto Scaling Groups, 0 Dedicated Hosts, 0 Elastic IPs, 0 Instances, 6 Key pairs, 0 Load balancers, 0 Placement groups, 30 Security groups, 0 Snapshots, and 0 Volumes. Below this is a 'Launch instance' section with a prominent orange 'Launch instance' button. To the right, there's a 'Service health' section showing 'This service is operating normally.' and a 'Zones' table. On the far right, there's a sidebar for 'EC2 Free Tier' offers, 'Offer usage (monthly)', and 'Account attributes'.

Creating a Jenkins sever:

Name and tags:

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console. In the 'Name and tags' step, the instance name is set to 'jenkins server'. The 'Application and OS Images (Amazon Machine Image)' section is expanded, showing a search bar and a grid of OS icons. The 'Summary' section on the right shows one instance being launched with an Amazon Linux 2023.5.2 AMI, t2.micro instance type, and a single 8 GiB volume. 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'. The 'Launch instance' button is highlighted.

Application and OS Images (Amazon Machine Image):

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console, specifically the 'Application and OS Images (Amazon Machine Image)' step. It displays a catalog of OS images, including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. The 'Ubuntu' image is selected. The 'Summary' section on the right shows one instance being launched with a Canonical, Ubuntu, 24.04 LTS AMI, t2.micro instance type, and a single 8 GiB volume. 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'. The 'Launch instance' button is highlighted.

Instance type:

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console. The 'Summary' step is selected. Configuration details include:

- Description:** Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
- Architecture:** 64-bit (x86)
- AMI ID:** ami-04a81a99f5ec58529 (Verified provider)
- Instance type:** t2.medium (selected)
 - Family: t2
 - 2 vCPU
 - 4 GB Memory
 - Current generation: true
 - On-Demand Linux base pricing: 0.0464 USD per Hour
 - On-Demand RHEL base pricing: 0.0752 USD per Hour
 - On-Demand Windows base pricing: 0.0644 USD per Hour
 - On-Demand SUSE base pricing: 0.1464 USD per Hour
- Additional costs apply for AMIs with pre-installed software**
- Key pair (login):** (Info | Get advice)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
- Summary:** Number of instances: 1
- Software Image (AMI):** Canonical, Ubuntu, 24.04 LTS, ...read more
- Virtual server type (instance type):** t2.medium
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB
- 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.
- Buttons:** Cancel, Launch instance, Review commands

Key pair (login):

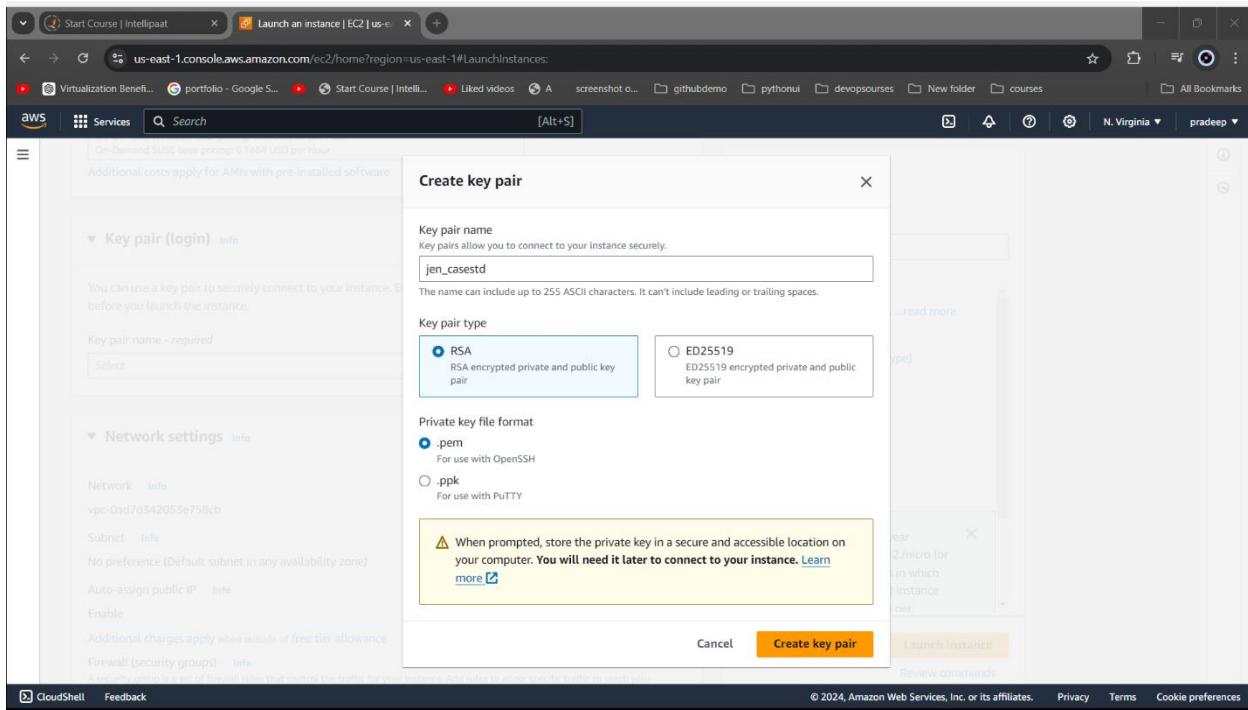
-> click on Create new key pair.

The screenshot shows the 'Create key pair' dialog box. The fields are:

- Key pair name:** Enter key pair name (text input field)
- Key pair type:**
 - RSA: RSA encrypted private and public key pair
 - ED25519: ED25519 encrypted private and public key pair
- Private key file format:**
 - .pem: For use with OpenSSH
 - .ppk: For use with PuTTY
- Warning message:** When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)
- Buttons:** Cancel, Create key pair

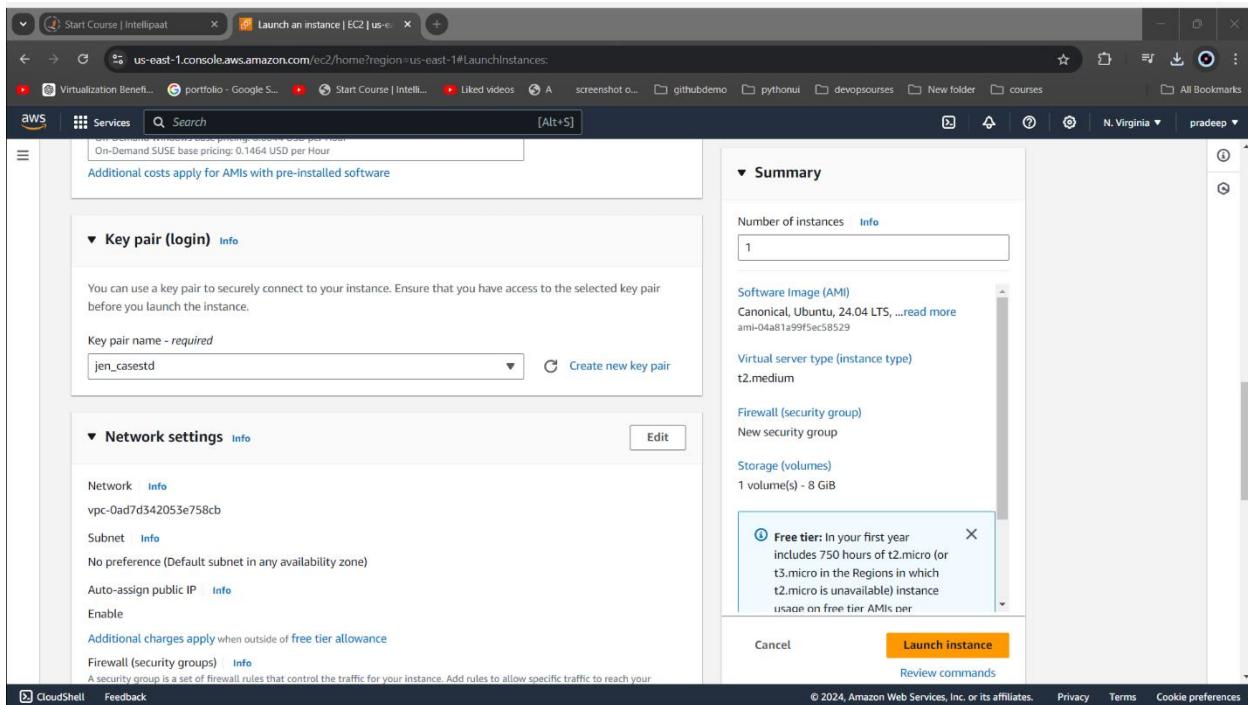
Create key pair:

Key pair name:



Private key file format:

-> pem.



Keypair is created.

Network settings is set to be default:

The screenshot shows the 'Network settings' step of the EC2 instance launch wizard. It includes fields for Network (vpc-0ad7d342053e758cb), Subnet (No preference), Auto-assign public IP (Enable), Firewall (Create security group selected), and security group rules for SSH, HTTPS, and HTTP traffic. A note at the bottom suggests setting security group rules for known IP addresses.

Configure storage:
storage is set to be 20 gb.

The screenshot shows the 'Configure storage' step of the EC2 instance launch wizard. It specifies 1x 20 GiB gp3 volume as the root volume (Not encrypted). A note indicates that free tier eligible customers can get up to 30 GB of EBS storage. A warning message states that only the first 0 instance store volumes from the AMI will be accessible.

Launching the instance.

The screenshot shows the AWS EC2 Instances launch page. A green success message at the top states: "Success Successfully initiated launch of instance (i-01bda222ee0e85ac7)". Below this, there's a "Next Steps" section with several options:

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database
- Create EBS snapshot policy

Each step has a corresponding button or link. At the bottom, there are links for CloudShell, Feedback, and a footer with copyright information.

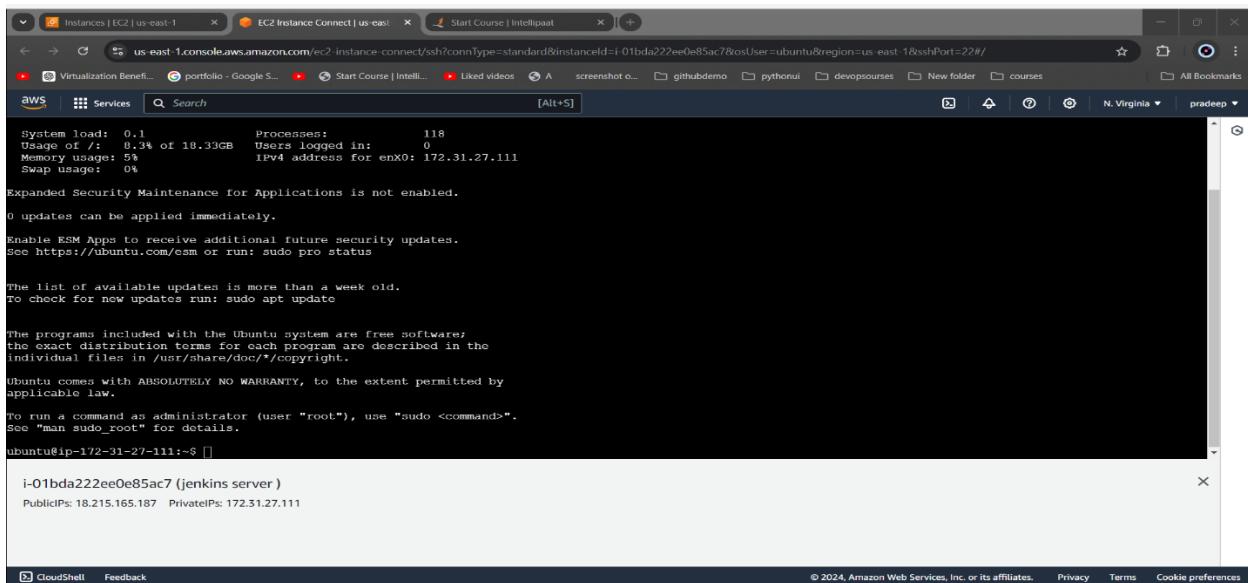
The screenshot shows the AWS EC2 Instances dashboard. On the left, a sidebar lists various services like EC2 Dashboard, EC2 Global View, Events, and Instances. The Instances section is expanded, showing a table of instances. One instance is listed: "jenkins server" (Instance ID: i-01bda222ee0e85ac7), which is currently "Running".

Jenkins server in created.

So, the creation of the EC2 instance for Jenkins is complete. Now, Jenkins needs to be installed, and the Jenkins dashboard needs to be set up."

Connecting to Jenkins server:

Select the jenkins server - > connect.



```
System load: 0.1      Processes:          118
Usage of /: 8.3% of 18.33GB  Users logged in: 0
Memory usage: 5%        IPv4 address for enx0: 172.31.27.111
Swap usage: 0%         

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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-172-31-27-111:~$ []

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187  PrivateIPs: 172.31.27.111
```

To install and set up a Jenkins server, follow these steps:

Install Java:

Jenkins requires Java to run, so you need to install it first.

Commands:

→ Sudo apt update -y

```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | Start Course | Intellipaat | Debian Jenkins Packages | + | - | ○ | × | [Alt+S]

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda22ee0e85ac7&osUser=ubuntu&region=us-east-1&sshPort=22#/
```

Usage of /: 8.3% of 10.33GB Users logged in: 0
Memory usage: 5% IPv4 address for enx0: 172.31.27.111
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
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

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-172-31-27-111:~$ ls
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# apt update -y
```

i-01bda22ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | Start Course | Intellipaat | Debian Jenkins Packages | + | - | ○ | × | [Alt+S]

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda22ee0e85ac7&osUser=ubuntu&region=us-east-1&sshPort=22#/
```

```
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [86.7 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [5716 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [321 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [135 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [12.7 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [241 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [47.0 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [416 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.3 kB]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB]
Get:45 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:46 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1016 B]
Get:47 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:48 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [1116 B]
Get:49 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:50 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [1116 B]
Fetched 28.3 MB in 5s (5964 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-27-111:/home/ubuntu#
```

i-01bda22ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

→ Sudo apt install openjdk-11-jre

```
Instances | EC2 | us-east-1 EC2 Instance Connect | us-east-1 Start Course | Intellipaat Debian Jenkins Packages + Instances | EC2 | us-east-1 us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda222ee0e85ac7&osUser=ubuntu&region=us-east-1&sshPort=22/# Virtualization Benefit... portfolio - Google S... Start Course | Intellipaat Liked videos A screenshot o... githubdemo pythonui devopsources New folder courses All Bookmarks AWS Services Search [Alt+S] Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [86.7 kB] Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [5716 B] Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [321 kB] Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [135 kB] Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB] Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [12.7 kB] Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [241 kB] Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [47.0 kB] Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [416 B] Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB] Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B] Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B] Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B] Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B] Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B] Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.3 kB] Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB] Get:45 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB] Get:46 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1016 B] Get:47 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B] Get:48 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B] Get:49 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B] Get:50 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B] Fetched 28.3 MB in 5s (5964 kB/s) Reading package lists... Done Building dependency tree... Done Reading state information... Done 51 packages can be upgraded. Run 'apt list --upgradable' to see them. root@ip-172-31-27-111:/home/ubuntu# apt install openjdk-11-jre -y] i-01bda222ee0e85ac7 (jenkins server) PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

```
Instances | EC2 | us-east-1 EC2 Instance Connect | us-east-1 Start Course | Intellipaat Debian Jenkins Packages + Instances | EC2 | us-east-1 us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda222ee0e85ac7&osUser=ubuntu&region=us-east-1&sshPort=22/# Virtualization Benefit... portfolio - Google S... Start Course | Intellipaat Liked videos A screenshot o... githubdemo pythonui devopsources New folder courses All Bookmarks AWS Services Search [Alt+S] Adding debian:UCA_Global_G2_Root.pem Adding debian:USERTrust_ECC_Certification_Authority.pem Adding debian:USERTrust_RSA_Certification_Authority.pem Adding debian:Xamp_Global_CA_Root.pem Adding debian:cortSIGN_ROOT_CA.pem Adding debian:cortSIGN_Root_CA_G2.pem Adding debian:cortSIGN_Root_CA_G3.pem Adding debian:eSsign_Root_CA_2017.pem Adding debian:epKT_Root_Certification_Authority.pem Adding debian:emSign_ECC_Root_CA_G3.pem Adding debian:emSign_ECC_Root_CA_G3.pem Adding debian:emSign_Root_CA_G1.pem Adding debian:emSign_Root_CA_G1.pem Adding debian:virus_ECC_Root_CA.pem Adding debian:virus_ROOT_CA.pem done Setting up openjdk-11-jre:amd64 (11.0.24+8-1ubuntu3~24.04.1) ... Scanning processes... Scanning linux images... Running kernel seems to be up-to-date. No services need to be restarted. 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-172-31-27-111:/home/ubuntu# [] i-01bda222ee0e85ac7 (jenkins server) PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

Add Jenkins Repository:

We need to add the Jenkins repository to install the latest version.

We can use the “pkg.jenkins.io” this site for installation for repo.

The screenshot shows a browser window with multiple tabs. The active tab displays the directory index for the root path on the pkg.jenkins.io website. The page title is "Index of /". The table lists files and directories with their names, last modified dates, sizes, and descriptions. Notable entries include "debian-stable/" and "openSUSE-stable/".

Name	Last modified	Size	Description
debian-rc/	2016-04-07 04:19	-	
debian-stable-rc/	2016-02-04 19:46	-	
debian-stable/	2024-08-07 10:16	-	
debian/	2024-08-07 11:19	-	
openSUSE-rc/	2016-04-07 04:21	-	
openSUSE-stable-rc/	2016-02-04 19:48	-	
openSUSE-stable/	2024-08-07 10:16	-	
openSUSE/	2024-08-07 11:19	-	
redhat-rc/	2016-04-07 04:20	-	
redhat-stable-rc/	2016-02-04 19:47	-	
redhat-stable/	2024-08-07 10:16	-	
redhat/	2024-08-07 11:19	-	
war/	2020-04-16 16:01	-	
windows/	2020-04-16 16:02	-	

We are using ubuntu so it is from Debian family of linux. And we can need select the option called.

[-> “debian-stable/”](#)

The screenshot shows the "Jenkins Debian Packages" page on the pkg.jenkins.io website. The page title is "Jenkins Debian Packages". It contains instructions for adding the repository to a system:

- This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system (for the Weekly Release Line):

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

Then add a Jenkins apt repository entry:

```
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/" | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

Update your local package index, then finally install Jenkins:

```
sudo apt-get update
sudo apt-get install fontconfig openjdk-17-jre
sudo apt-get install jenkins
```

The apt packages were signed using this key:

```
pub rsa4096 2023-03-27 [SC] [expires: 2026-03-26]
63667EE74BBA1F0A08A698725BA31D57EFS975CA
uid Jenkins Project
sub rsa4096 2023-03-27 [E] [expires: 2026-03-26]
```

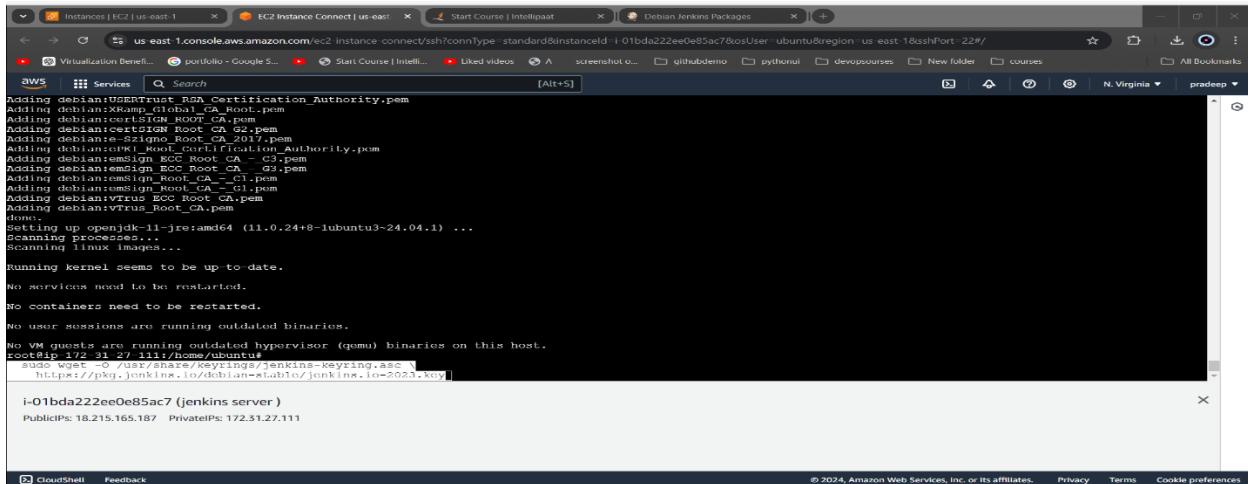
This is the Debian package repository of Jenkins to automate installation and upgrade.

→ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

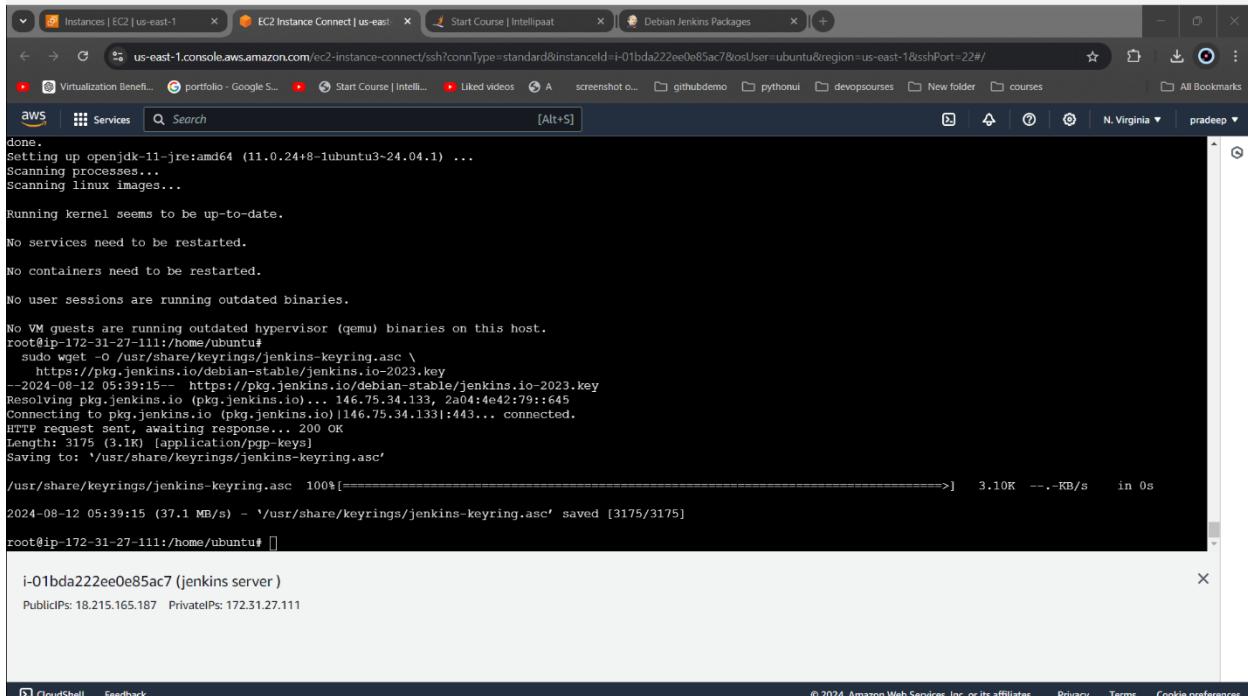
**sudo wget -O /usr/share/keyrings/jenkins-keyring.asc **

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

This command is essential for adding the Jenkins package repository key to the system's keyring, enabling the system to verify the authenticity of the Jenkins packages during installation or updates.



```
Adding debian:USERTrust RSA_Certification_Authority.pem
Adding debian:XRM_Root_CA.pem
Adding debiandLTM_Root_CA.pem
Adding debian:certSIGN Root CA G2.pem
Adding debian:eSign_Root_CA_2017.pem
Adding debian:emSign_ECC_Root_CA_G3.pem
Adding debian:emSign_ECC_Root_CA_G3.pem
Adding debian:emSign_Root_CA_G1.pem
Adding debian:virus_ECC_Root_CA.pem
Adding debian:virus_Root_CA.pem
done.
Setting up openjdk-11-jre:amd64 (11.0.24+8~ubuntu3~24.04.1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
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-172-31-27-111:/home/ubuntu# sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```



```
done.
Setting up openjdk-11-jre:amd64 (11.0.24+8~ubuntu3~24.04.1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
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-172-31-27-111:/home/ubuntu# sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2024-08-12 05:39:15-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:79::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc 100%[=====] 3.10K --.-KB/s   in 0s
2024-08-12 05:39:15 (37.1 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]
root@ip-172-31-27-111:/home/ubuntu#
```

Then add a Jenkins apt repository entry:

→ echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \

```
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
Instances | EC2 | us-east-1 EC2 Instance Connect | us-east-1 Start Course | Intellipaat Debian Jenkins Packages [Alt+S]

Running kernel seems to be up-to-date.
No services need to be restarted.
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-172-31-27-111:/home/ubuntu# sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2024-08-12 05:39:15-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:79::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc 100%[=====] 3.10K --.-KB/s in 0s

2024-08-12 05:39:15 (37.1 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

root@ip-172-31-27-111:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null

i-01bda22ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

```
Instances | EC2 | us-east-1 EC2 Instance Connect | us-east-1 Start Course | Intellipaat Debian Jenkins Packages [Alt+S]

Running kernel seems to be up-to-date.
No services need to be restarted.
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-172-31-27-111:/home/ubuntu# sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2024-08-12 05:39:15-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:79::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc 100%[=====] 3.10K --.-KB/s in 0s

2024-08-12 05:39:15 (37.1 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

root@ip-172-31-27-111:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null

root@ip-172-31-27-111:/home/ubuntu# 
```

Update your local package index, then finally install Jenkins:

➔ sudo apt-get update

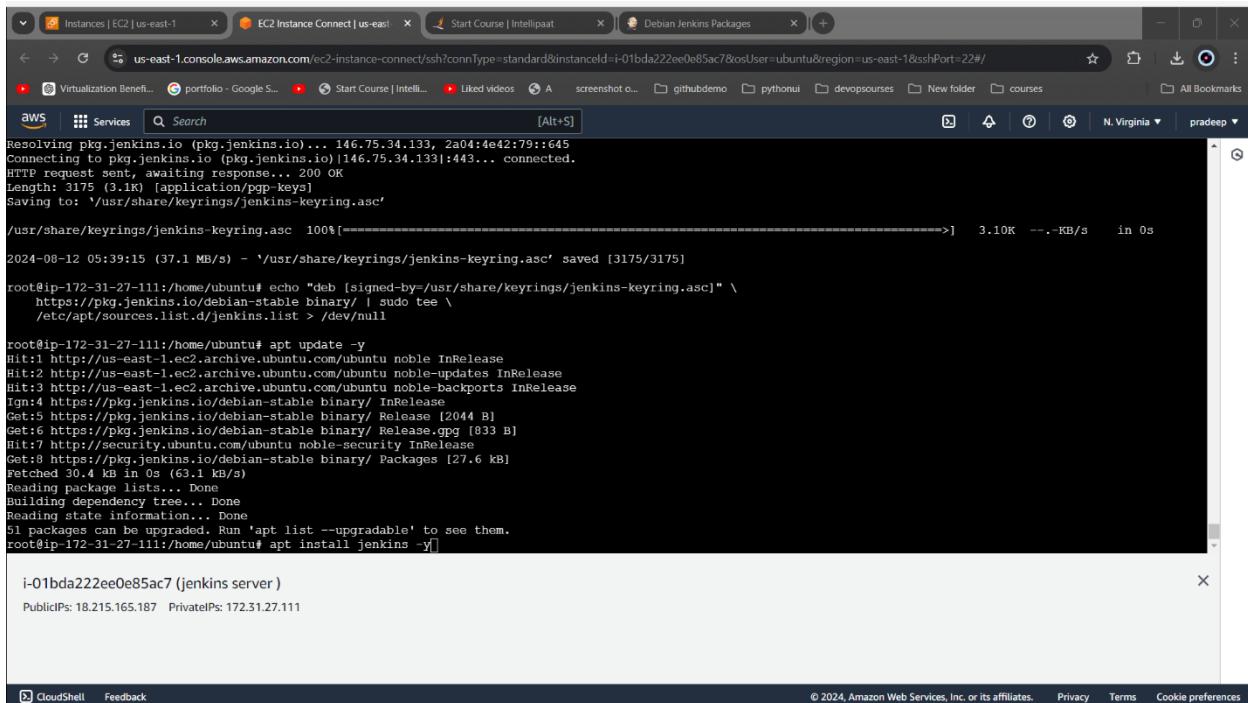
```
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
root@ip-172-31-27-111:/home/ubuntu# sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \  
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key  
--2024-08-12 05:39:15 -- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key  
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:79::645  
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 3175 (3.1K) [application/pgp-keys]  
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'  
  
/usr/share/keyrings/jenkins-keyring.asc 100%[=====] 3.10K --.-KB/s in 0s  
2024-08-12 05:39:15 (37.1 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]  
root@ip-172-31-27-111:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \  
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \  
/etc/apt/sources.list.d/jenkins.list > /dev/null  
root@ip-172-31-27-111:/home/ubuntu# apt update -y  
  
i-01bda222ee0e85ac7 (jenkins server)  
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

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

```
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:79::645  
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 3175 (3.1K) [application/pgp-keys]  
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'  
  
/usr/share/keyrings/jenkins-keyring.asc 100%[=====] 3.10K --.-KB/s in 0s  
2024-08-12 05:39:15 (37.1 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]  
root@ip-172-31-27-111:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \  
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \  
/etc/apt/sources.list.d/jenkins.list > /dev/null  
root@ip-172-31-27-111:/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  
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease  
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]  
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]  
Hit:7 http://security.ubuntu.com/ubuntu noble-security InRelease  
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [27.6 kB]  
Fetched 30.4 kB in 0s (63.1 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
51 packages can be upgraded. Run 'apt list --upgradable' to see them.  
root@ip-172-31-27-111:/home/ubuntu#  
  
i-01bda222ee0e85ac7 (jenkins server)  
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

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

→ sudo apt-get install jenkins



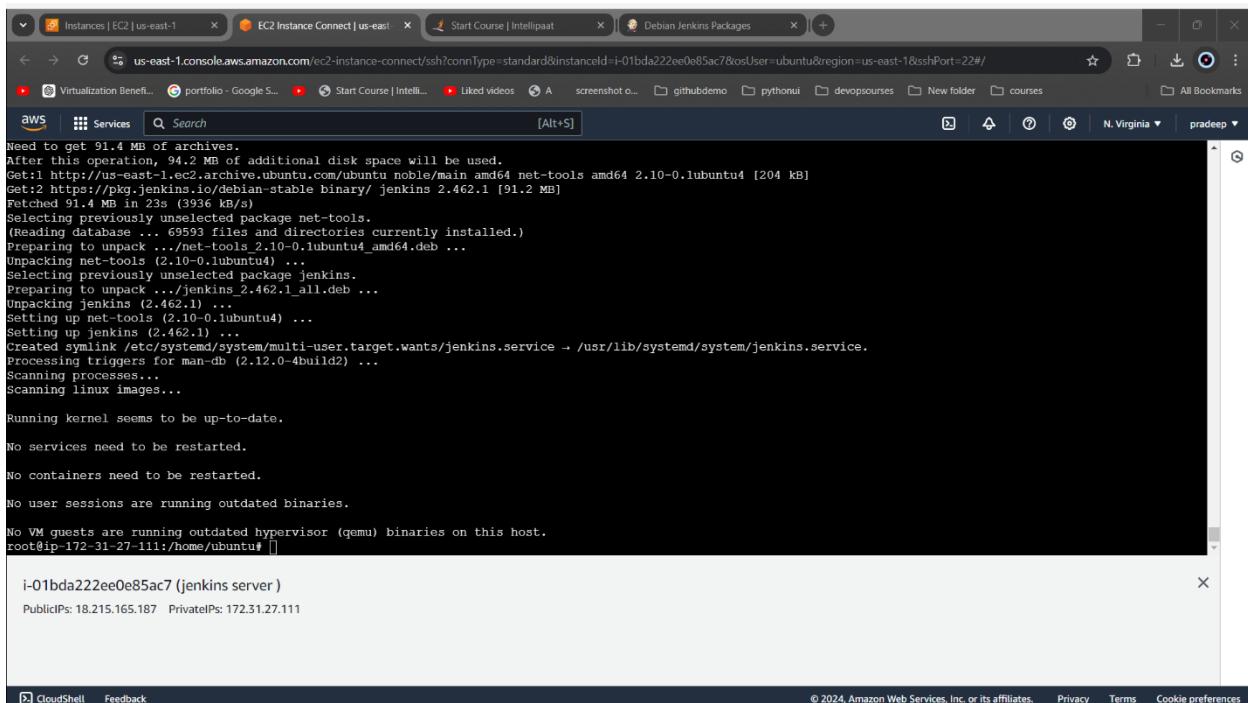
```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | Start Course | Intellipaat | Debian Jenkins Packages | + | - | o | x |
```

Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:79::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)[146.75.34.133]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc 100%[=====] 3.10K --.-KB/s in 0s
2024-08-12 05:39:15 (37.1 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]
root@ip-172-31-27-111:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null

root@ip-172-31-27-111:/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
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [27.6 kB]
Fetched 30.4 kB in 0s (63.1 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-27-111:/home/ubuntu# apt install jenkins -y

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111



```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | Start Course | Intellipaat | Debian Jenkins Packages | + | - | o | x |
```

Need to get 91.4 MB of archives.
After this operation, 94.2 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 net-tools amd64 2.10-0.lubuntu4 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.462.1 [91.2 MB]
Patched 91.4 MB in 23s (3936 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 69593 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-0.lubuntu4_amd64.deb ...
Unpacking net-tools (2.10-0.lubuntu4) ...
selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.462.1_all.deb ...
Unpacking Jenkins (2.462.1) ...
Setting up net-tools (2.10-0.lubuntu4) ...
Setting up jenkins (2.462.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
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-172-31-27-111:/home/ubuntu#

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

Start Jenkins:

Start the Jenkins service.

→ sudo systemctl start jenkins

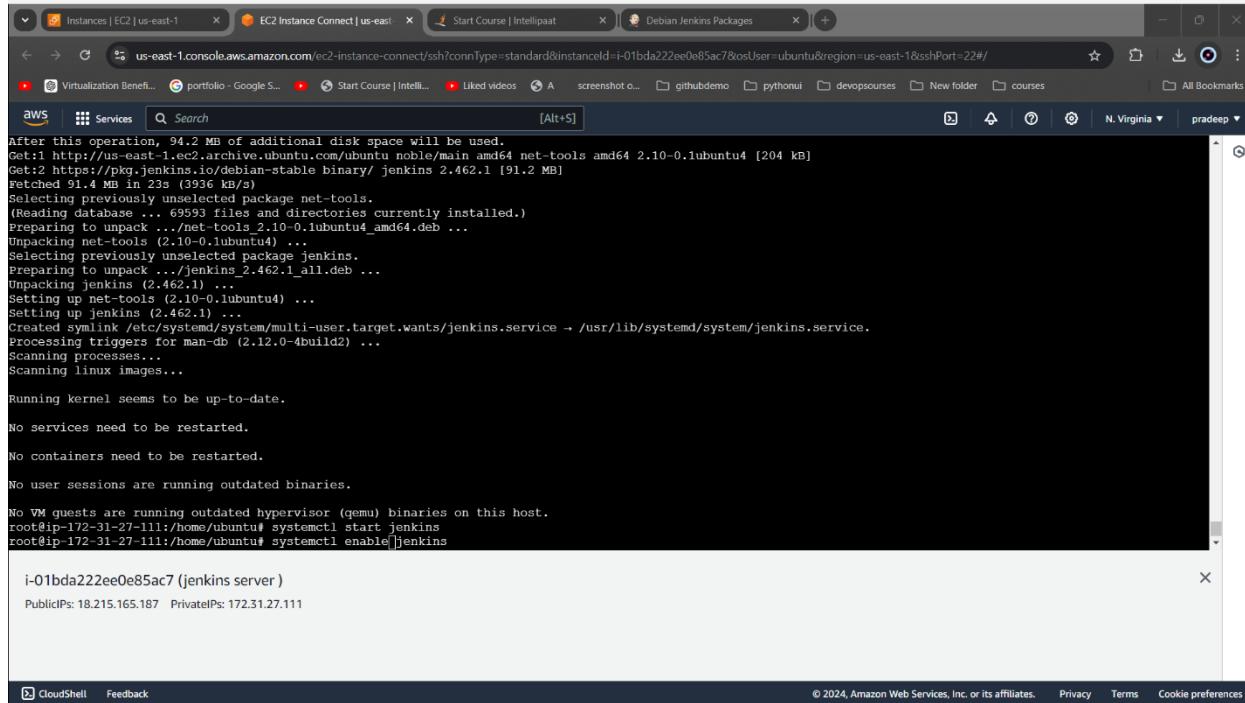
```
Need to get 91.4 MB of archives.  
After this operation, 94.2 MB of additional disk space will be used.  
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 net-tools amd64 2.10-0.1ubuntu4 [204 kB]  
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.462.1 [91.2 MB]  
Fetched 91.4 MB in 23s (3936 kB/s)  
Selecting previously unselected package net-tools.  
(Reading database ... 69593 files and directories currently installed.)  
Preparing to unpack .../net-tools_2.10-0.1ubuntu4_amd64.deb ...  
Unpacking net-tools (2.10-0.1ubuntu4) ...  
selecting previously unselected package jenkins.  
Preparing to unpack .../jenkins_2.462.1_all.deb ...  
Unpacking jenkins (2.462.1) ...  
Setting up net-tools (2.10-0.1ubuntu4) ...  
Setting up jenkins (2.462.1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.  
Processing triggers for man-db (2.12.0-4build2) ...  
Scanning processes...  
Scanning linux images...  
Running kernel seems to be up-to-date.  
No services need to be restarted.  
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-172-31-27-111:/home/ubuntu# systemctl start jenkins  
  
i-01bda222ee0e85ac7 (jenkins server)  
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

```
After this operation, 94.2 MB of additional disk space will be used.  
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 net-tools amd64 2.10-0.1ubuntu4 [204 kB]  
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.462.1 [91.2 MB]  
Fetched 91.4 MB in 23s (3936 kB/s)  
Selecting previously unselected package net-tools.  
(Reading database ... 69593 files and directories currently installed.)  
Preparing to unpack .../net-tools_2.10-0.1ubuntu4_amd64.deb ...  
Unpacking net-tools (2.10-0.1ubuntu4) ...  
Selecting previously unselected package jenkins.  
Preparing to unpack .../jenkins_2.462.1_all.deb ...  
Unpacking jenkins (2.462.1) ...  
Setting up net-tools (2.10-0.1ubuntu4) ...  
Setting up jenkins (2.462.1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.  
Processing triggers for man-db (2.12.0-4build2) ...  
Scanning processes...  
Scanning linux images...  
Running kernel seems to be up-to-date.  
No services need to be restarted.  
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-172-31-27-111:/home/ubuntu# systemctl start jenkins  
root@ip-172-31-27-111:/home/ubuntu#  
  
i-01bda222ee0e85ac7 (jenkins server)  
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

Enable Jenkins to Start on Boot:

To ensure Jenkins starts on boot, enable the service.

→ sudo systemctl enable jenkins



```
After this operation, 94.2 MB of additional disk space will be used.
Get:1 https://us-east-1.ec2.archive.ubuntu.com/ubuntu main amd64 net-tools amd64 2.10-0.1ubuntu4 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.462.1 [91.2 MB]
Fetched 91.4 MB in 23s (3936 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 69593 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-0.1ubuntu4_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.462.1_all.deb ...
Unpacking jenkins (2.462.1)
Setting up net-tools (2.10-0.1ubuntu4) ...
Setting up jenkins (2.462.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

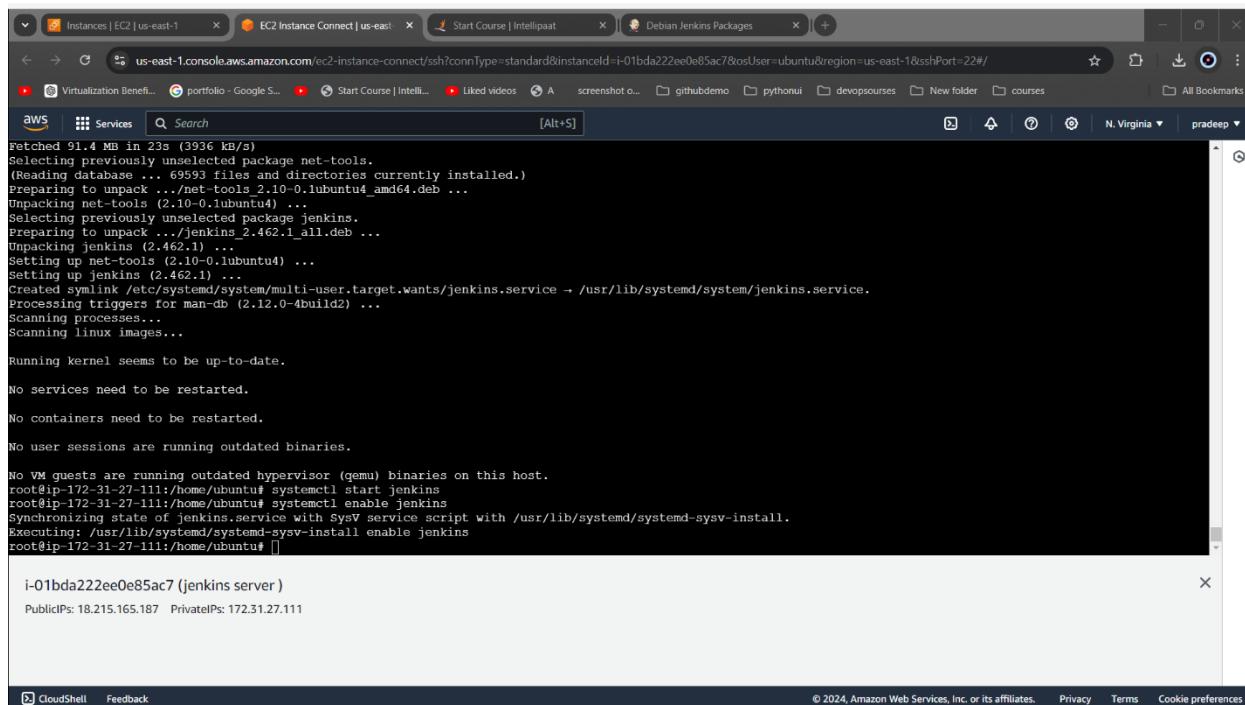
No services need to be restarted.

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-172-31-27-111:/home/ubuntu# systemctl start jenkins
root@ip-172-31-27-111:/home/ubuntu# systemctl enable jenkins

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```



```
Fetched 91.4 MB in 23s (3936 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 69593 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-0.1ubuntu4_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.462.1_all.deb ...
Unpacking jenkins (2.462.1)
Setting up net-tools (2.10-0.1ubuntu4) ...
Setting up jenkins (2.462.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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-172-31-27-111:/home/ubuntu# systemctl start jenkins
root@ip-172-31-27-111:/home/ubuntu# systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
root@ip-172-31-27-111:/home/ubuntu# []

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

Adjust Firewall:

If you have a firewall running, you'll need to allow traffic on port 8080 in security group.

Instances (1/1) **Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
jenkins server	i-01bda222ee0e85ac7	Running	t2.medium	Initializing	View alarms +	us-east-1b	ec2-18-21...

i-01bda222ee0e85ac7 (jenkins server)

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

Inbound rules

→ Inbound rules -> Edit Inbound rules Inbound rules->add rule.

ModifyInboundSecurityGroupRules | EC2 Instance Connect | us-east-1 | Start Course | Intellipaat | Debian Jenkins Packages

EC2 > Security Groups > sg-00b74eeb67787ec66 - launch-wizard-10 > Edit inbound rules

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-020923269bdb1ce77	SSH	TCP	22	Custom	0.0.0.0/0 X
-	Custom TCP	TCP	8080	Anywh...	0.0.0.0/0 X

Add rule

⚠ Rules with source of 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 EC2 Security Groups page. A green success message at the top states: "Inbound security group rules successfully modified on security group (sg-00b74eeb67787ec66 | launch-wizard-10) Details". Below this, the security group details are shown for "sg-00b74eeb67787ec66 - launch-wizard-10". The details table includes:

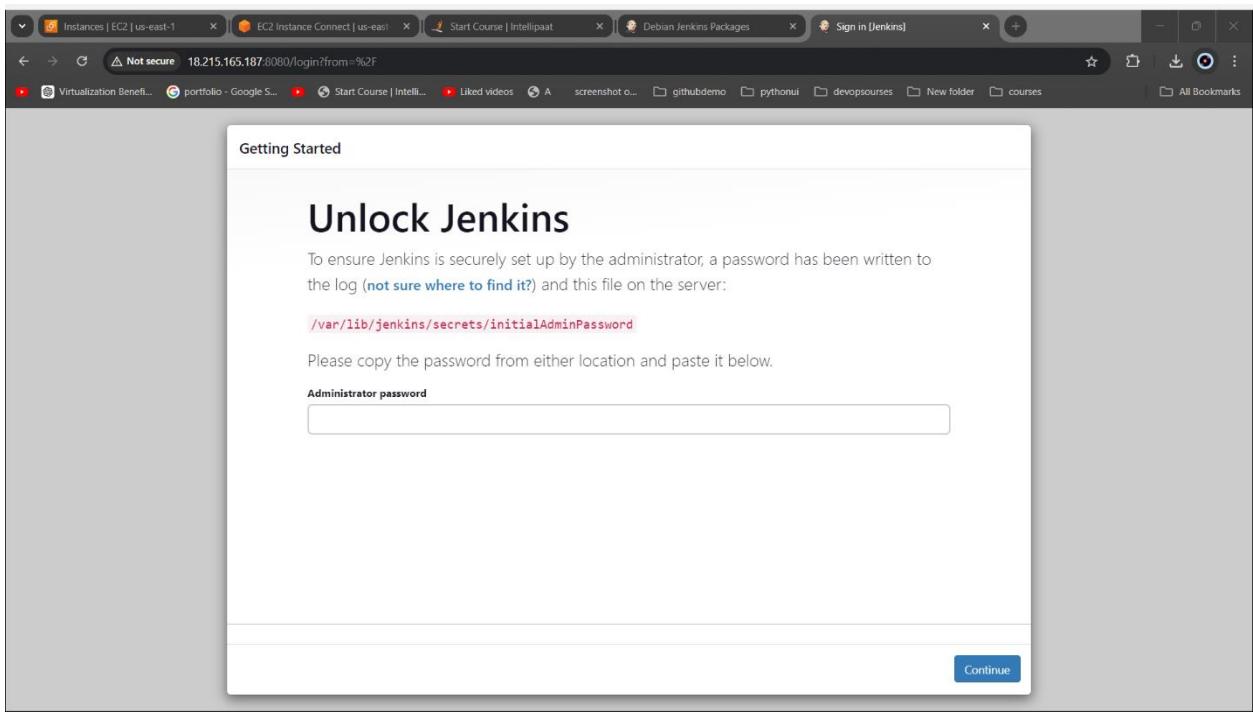
Security group name	Security group ID	Description	VPC ID
launch-wizard-10	sg-00b74eeb67787ec66	launch-wizard-10 created 2024-08-11T09:50:00.369Z	vpc-0ad7d342053e758cb
Owner	626130759947	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry

The "Inbound rules" tab is selected, showing two rules. The first rule is a default rule allowing all traffic from 0.0.0.0/0 to port 22. The second rule is a custom rule allowing port 22 from 18.215.165.187. Both rules have "Allow" status and "Edit inbound rules" buttons.

Access Jenkins:

Jenkins runs on port 8080 by default. You can access it by going to http://server_ip:8080.

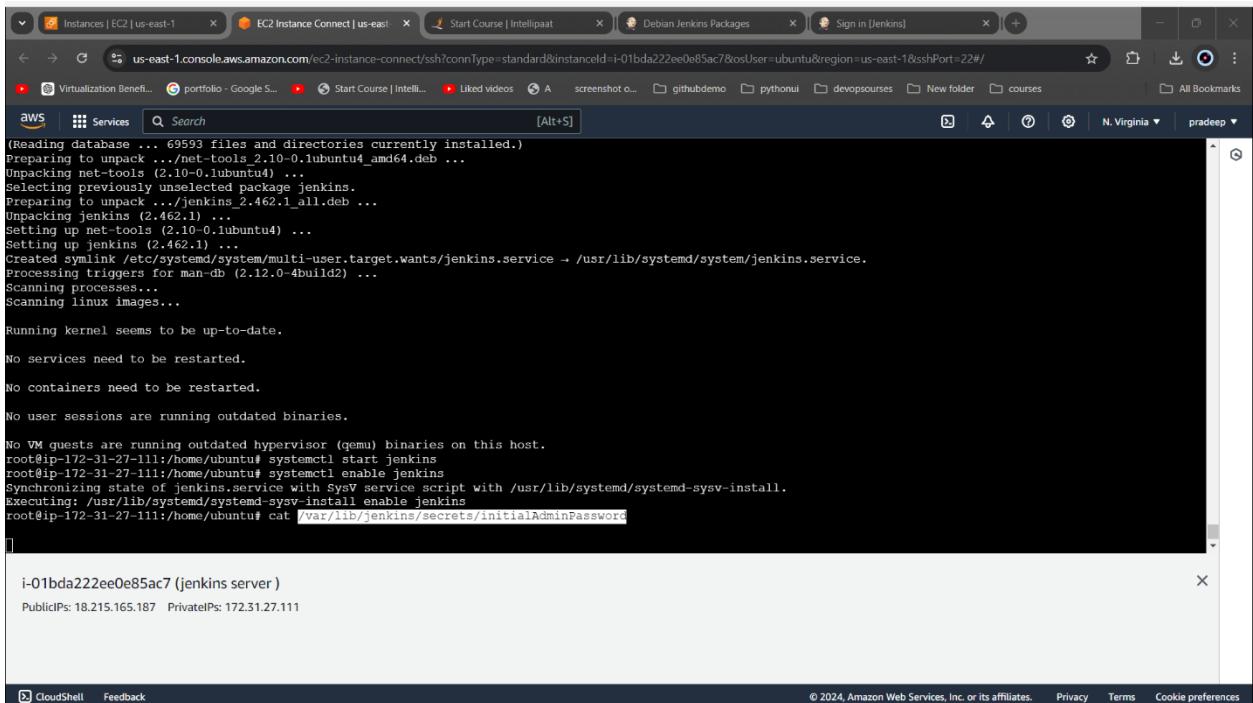
The screenshot shows a browser window displaying the Jenkins dashboard. The URL in the address bar is "18.215.165.187:8080". The dashboard features a central "Dashboard" section with various project cards and a "Build Status" summary. The bottom of the screen shows the standard browser navigation and search bar.



Unlock Jenkins:

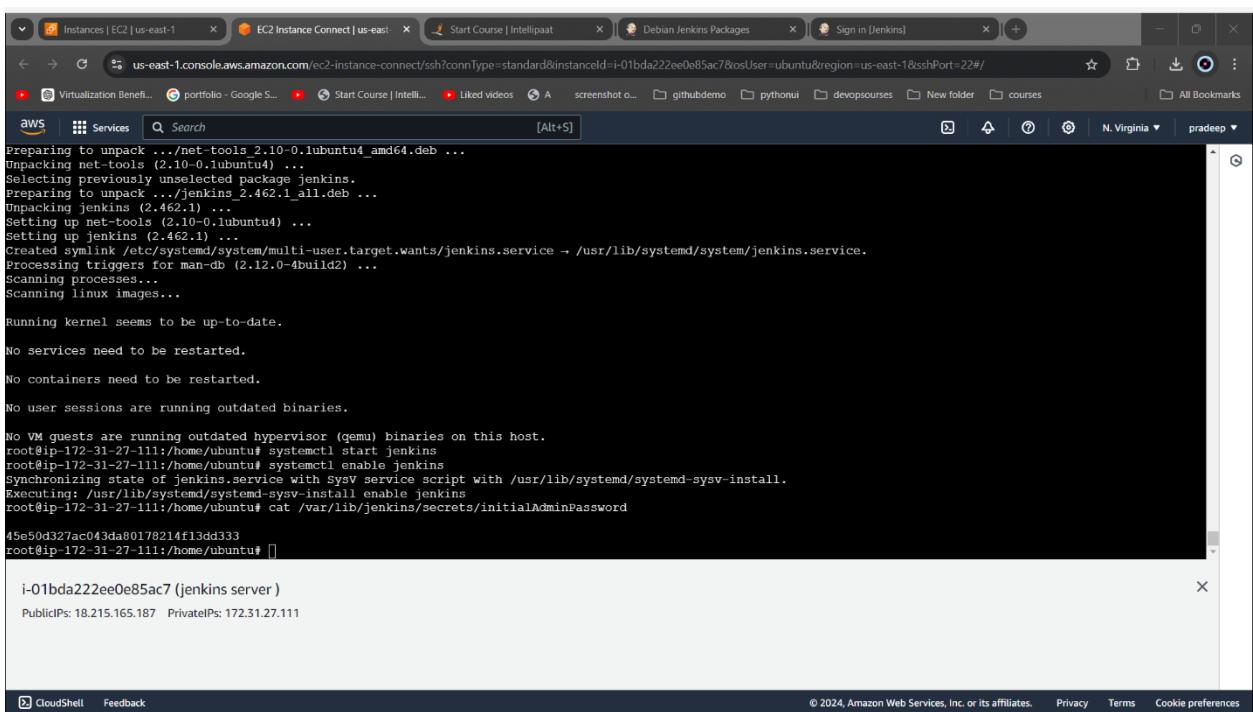
When we first access Jenkins, it will ask for an initial administrative password. We can find this password by running:

➔ sudo cat /var/lib/jenkins/secrets/initialAdminPassword



```
(Reading database ... 69593 files and directories currently installed.)  
Preparing to unpack .../net-tools_2.10-0.1ubuntu4_amd64.deb ...  
Unpacking net-tools (2.10-0.1ubuntu4) ...  
Selecting previously unselected package jenkins.  
Preparing to unpack .../jenkins_2.462.1_all.deb ...  
Unpacking jenkins (2.462.1) ...  
Setting up net-tools (2.10-0.1ubuntu4) ...  
Setting up jenkins (2.462.1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.  
Processing triggers for man-db (2.12.0-4build2) ...  
Scanning processes...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
No services need to be restarted.  
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-172-31-27-111:/home/ubuntu# systemctl start jenkins  
root@ip-172-31-27-111:/home/ubuntu# systemctl enable jenkins  
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins  
root@ip-172-31-27-111:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
```

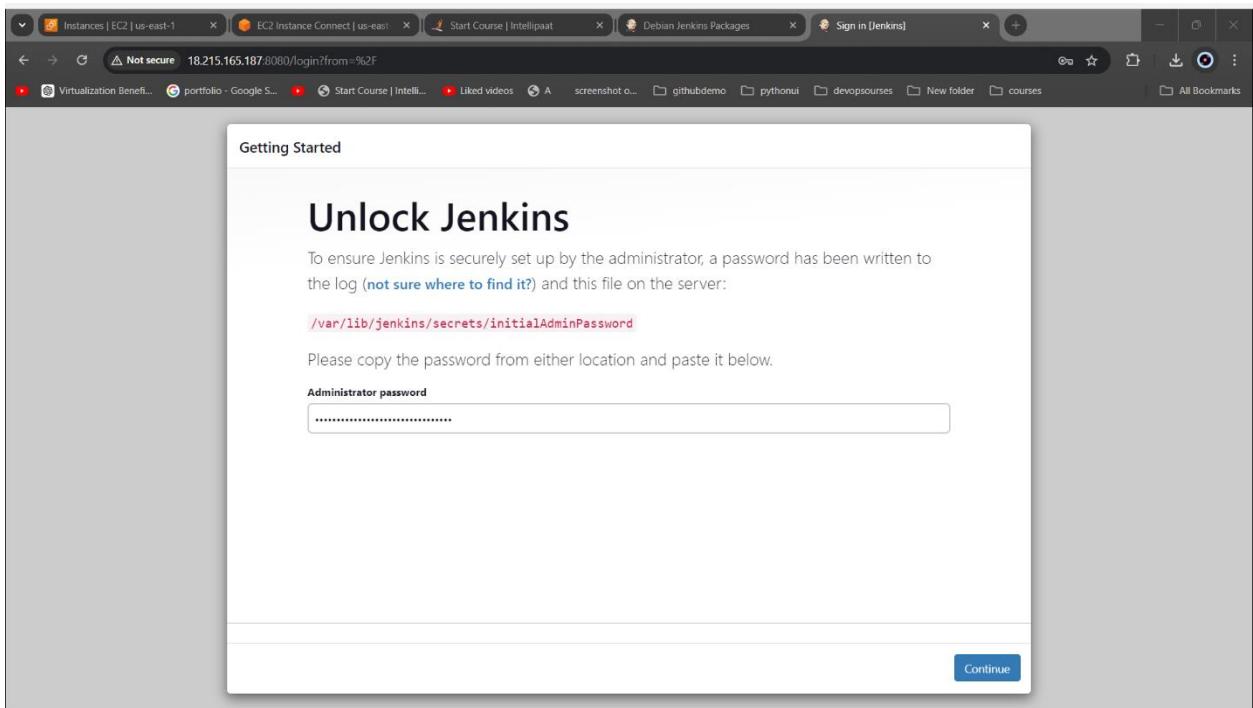
i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111



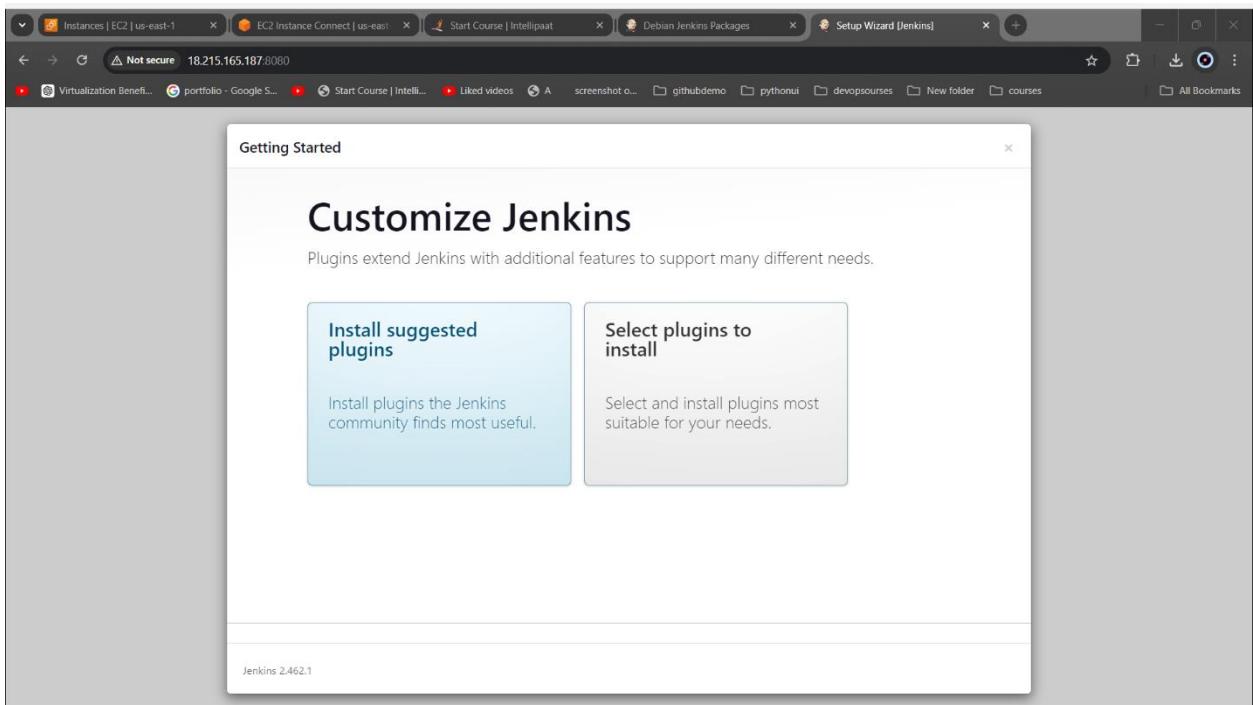
```
Preparing to unpack .../net-tools_2.10-0.1ubuntu4_amd64.deb ...  
Unpacking net-tools (2.10-0.1ubuntu4) ...  
Selecting previously unselected package jenkins.  
Preparing to unpack .../jenkins_2.462.1_all.deb ...  
Unpacking jenkins (2.462.1) ...  
Setting up net-tools (2.10-0.1ubuntu4) ...  
Setting up jenkins (2.462.1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.  
Processing triggers for man-db (2.12.0-4build2) ...  
Scanning processes...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
No services need to be restarted.  
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-172-31-27-111:/home/ubuntu# systemctl start jenkins  
root@ip-172-31-27-111:/home/ubuntu# systemctl enable jenkins  
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins  
root@ip-172-31-27-111:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword  
45e50d327ac043da80178214f13dd333  
root@ip-172-31-27-111:/home/ubuntu#
```

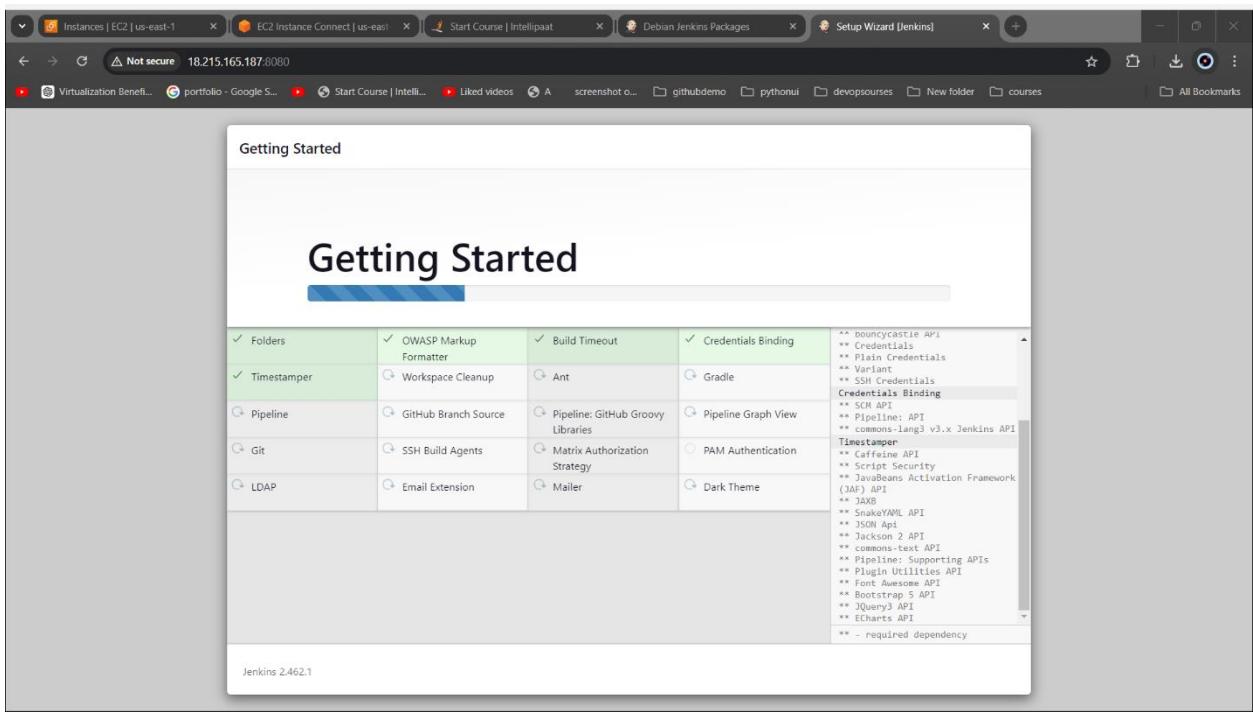
i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

Copy the password, paste it into the Jenkins interface

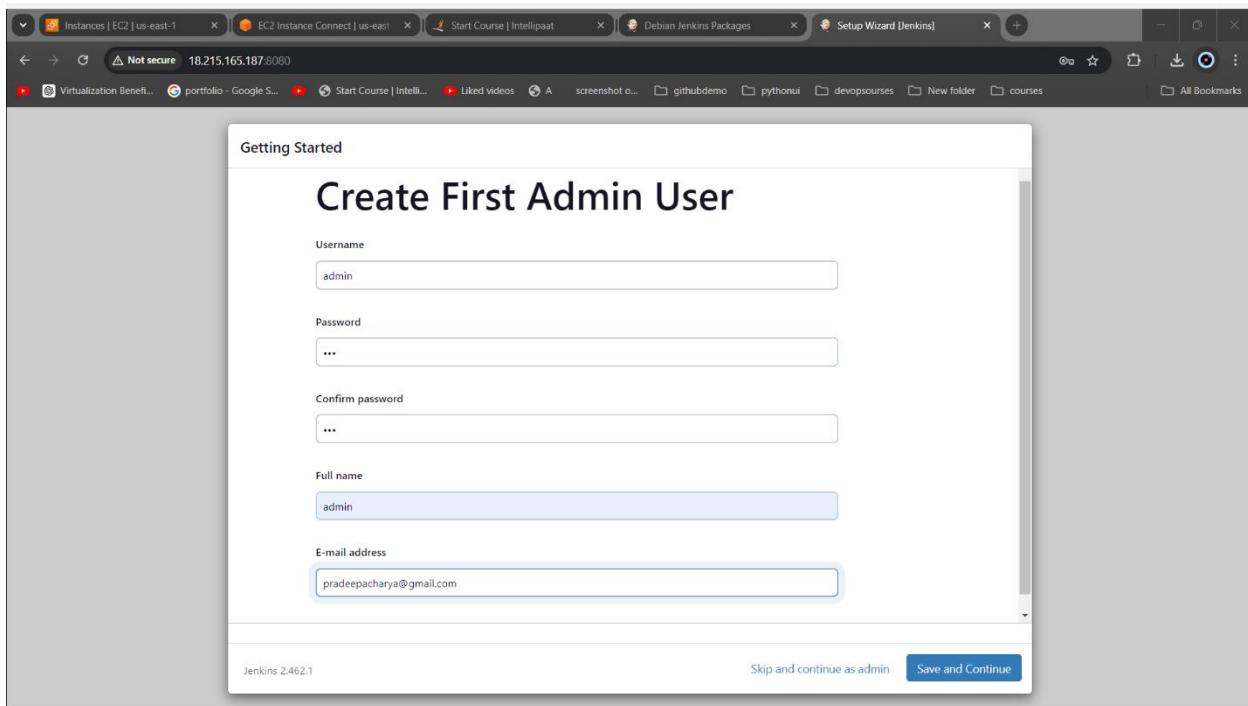


Choose customized jenkins:

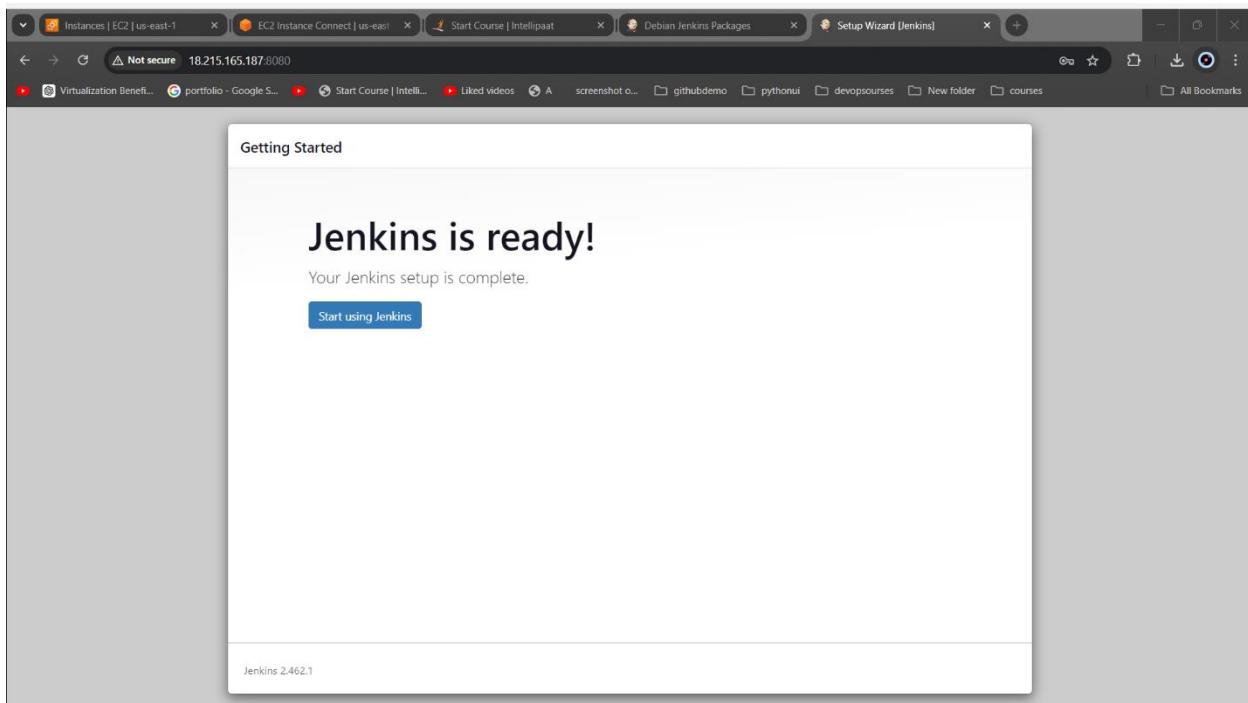
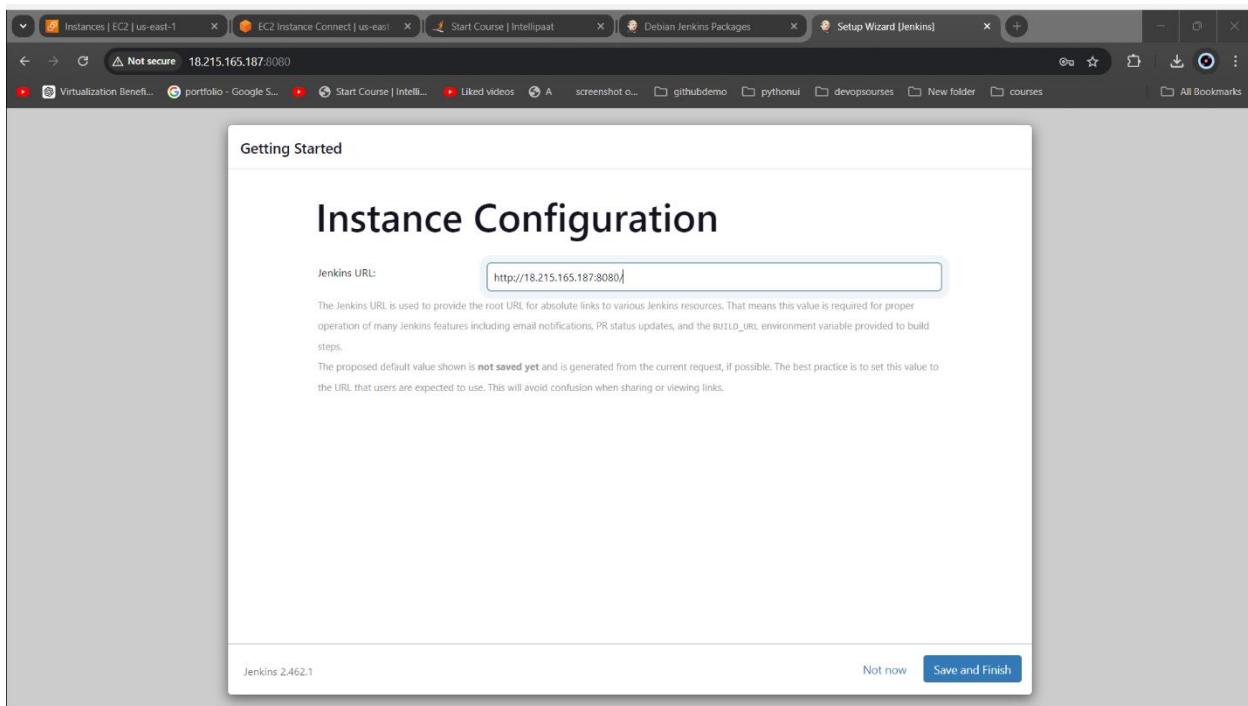




Creating admin user:



Save and continue.



Jenkins server up and running.

Implement the Git Workflow for continuous integration (CI) in the product hosted at the GitHub repository.

Creating a git repo in git hub:

The screenshot shows the GitHub Home page. At the top left, there's a 'Top repositories' section with a search bar and a 'New' button. Below it, a list of repositories owned by 'mysticlyph' includes 'jenkinsdemo', 'demowebapp', 'mysticlyph/mysticlyph', 'mysticlyph/DevOps Projects', and 'mysticlyph/demo-tomcat'. In the center, there's a 'Start a new repository for mysticlyph' form. To its right, a 'Introduce yourself with a profile README' section shows a sample README.md file with various profile details. On the right side of the main content area, there are two banners: 'GitHub Models' (Join the Limited Public Beta) and 'UNIVERSE'24' (29-30 October). Below the main content, there are sections for 'Use tools of the trade' (github.dev editor), 'Get AI-based coding suggestions' (GitHub Copilot), and 'Latest changes'.

The screenshot shows the 'Create a new repository' form. At the top, it says 'Create a new repository' and provides a link to 'Import a repository...'. It includes a note about repository names being short and memorable, with a suggestion for 'miniature-octo-fortnight'. Below this, there are fields for 'Owner' (set to 'mysticlyph') and 'Repository name *'. A description field is present, though empty. Under 'Visibility', the 'Public' option is selected, with a note that anyone can see the repository. There's also a 'Private' option. The 'Initialize this repository with:' section contains a checkbox for 'Add a README file', which is unchecked. A note explains that this is where to write a long project description. Below this, there's a 'Add .gitignore' section with a dropdown menu set to 'None'. A note about choosing ignore files is provided. Finally, there's a 'Choose a license' section.

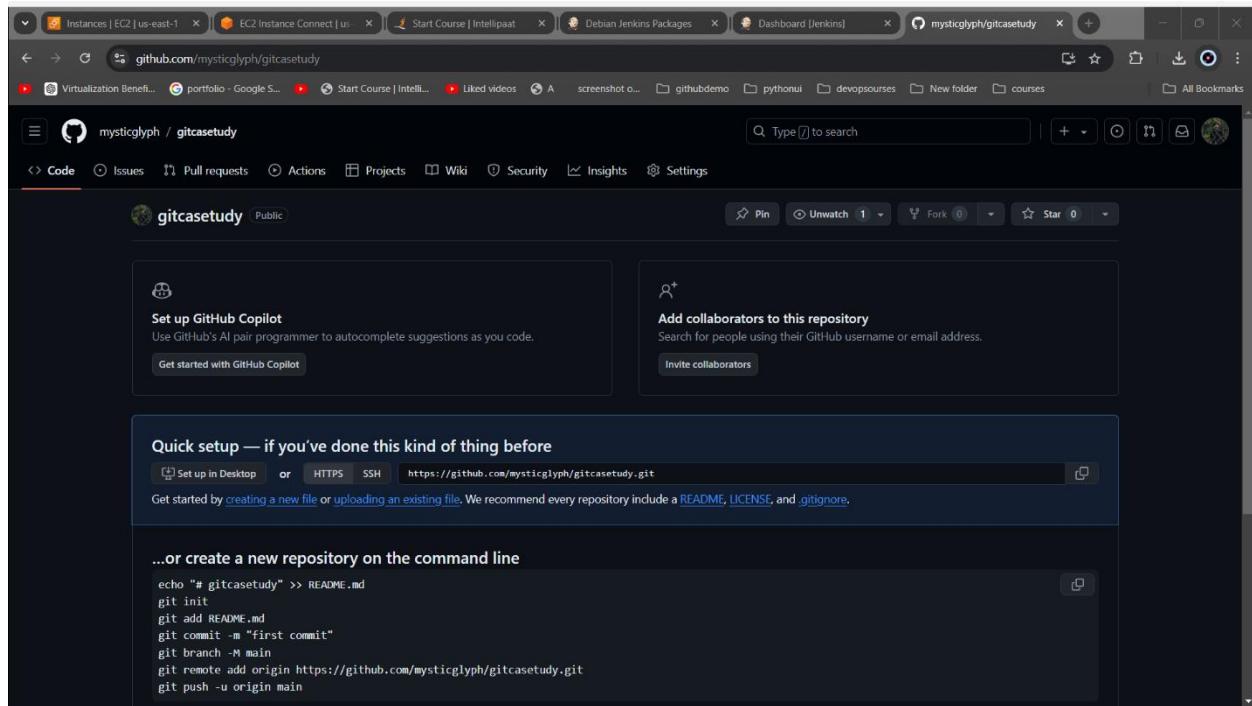
The screenshot shows the GitHub 'New repository' creation interface. The repository name is set to 'gitcasestudy'. The 'Owner' dropdown is set to 'mysticglyph'. The repository is marked as 'Public'. The 'Description' field is empty. Under 'Initialize this repository with:', the 'Add a README file' checkbox is checked. The 'Add .gitignore' section shows a dropdown menu set to 'None'. A note at the bottom says 'Choose which files not to track from a list of templates.' A 'Create repository' button is visible at the bottom right.

Repository is public:

This screenshot shows the same GitHub 'New repository' page as above, but with a license selected. The 'Choose a license' dropdown is set to 'Apache-2.0'. A note at the bottom says 'A license tells others what they can and can't do with your code.' A 'Create repository' button is visible at the bottom right.

Click on create repo.

Repo is created:



Install git in local Machin:

```
Preparing to unpack .../net-tools_2.10-0.1ubuntu4_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.462.1_all.deb ...
Unpacking jenkins (2.462.1) ...
Setting up net-tools (2.10-0.1ubuntu4) ...
Setting up jenkins (2.462.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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-172-31-27-111:/home/ubuntu# systemctl start jenkins
root@ip-172-31-27-111:/home/ubuntu# systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
root@ip-172-31-27-111:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
45e59d327ac043da80178214f13dd333
root@ip-172-31-27-111:/home/ubuntu# apt install git []

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

→ apt install git

```
Instances | EC2 | us-east-1 X EC2 Instance Connect | us-... X Start Course | Intellia... X Debian Jenkins Packages X Dashboard [Jenkins] X mysticglyph/gitcasestudy X + - ○ ×
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda222ee0e85ac7&osUser=ubuntu&region=us-east-1&sshPort=22#/
Virtualization Benefit... portfolio - Google S... Start Course | Intellia... Liked videos A screenshot o... githubdemo pythonui devopsources New folder courses All Bookmarks AWS Services Search [Alt+S]
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
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-172-31-27-111:/home/ubuntu# systemctl start jenkins
root@ip-172-31-27-111:/home/ubuntu# systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
root@ip-172-31-27-111:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
45e50d327ac043da80178214f13dd333
root@ip-172-31-27-111:/home/ubuntu# apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.1).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 29 not upgraded.
root@ip-172-31-27-111:/home/ubuntu# i-01bda222ee0e85ac7 (jenkins server )
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

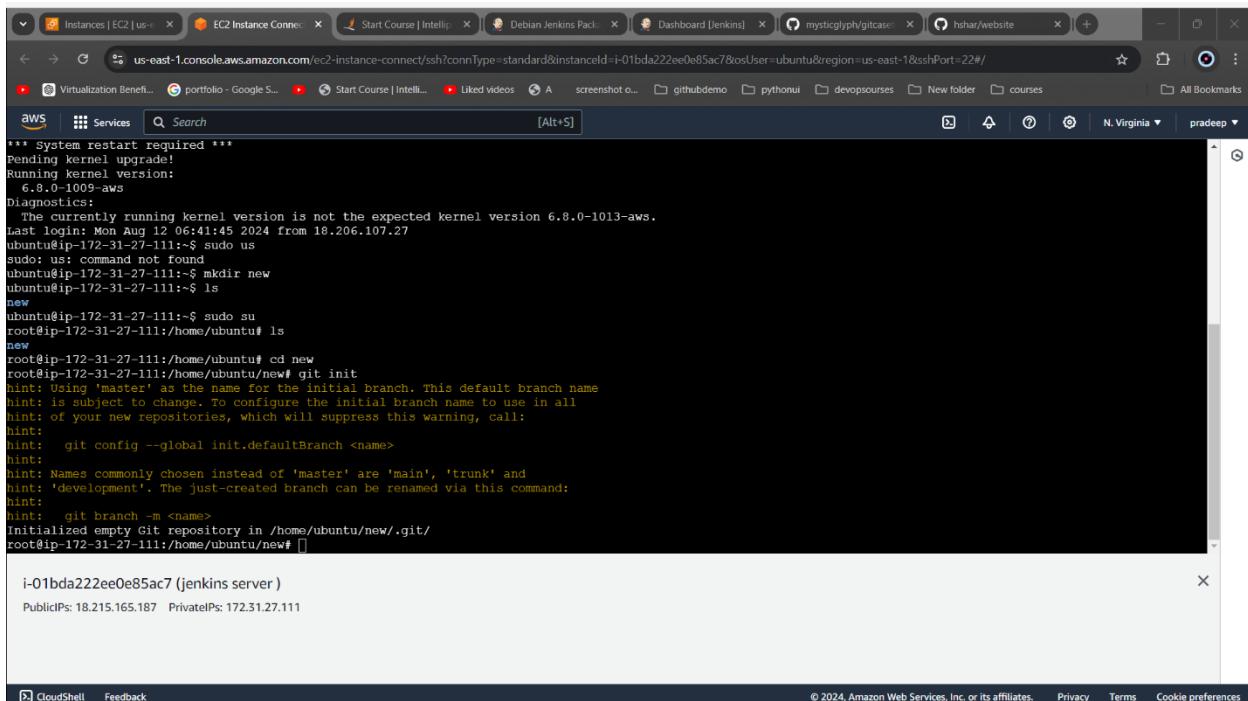
Create a folder to initializes local git repo:

→ mkdir new

```
Instances | EC2 | us-east-1 X EC2 Instance Connect | us-... X Start Course | Intellia... X Debian Jenkins Packages X Dashboard [Jenkins] X mysticglyph/gitcasestudy X lshar/website X + - ○ ×
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda222ee0e85ac7&osUser=ubuntu&region=us-east-1&sshPort=22#/
Virtualization Benefit... portfolio - Google S... Start Course | Intellia... Liked videos A screenshot o... githubdemo pythonui devopsources New folder courses All Bookmarks AWS Services Search [Alt+S]
Memory usage: 21% IPv4 address for enX0: 172.31.27.111
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
25 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 ***
Pending kernel upgrade!
Running kernel version:
 6.8.0-1009-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1013-aws.
Last login: Mon Aug 12 06:41:45 2024 from 18.206.107.27
ubuntu@ip-172-31-27-111:~$ sudo us
sudo: us: command not found
ubuntu@ip-172-31-27-111:~$ mkdir new
ubuntu@ip-172-31-27-111:~$ ls
new
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu#
new
root@ip-172-31-27-111:/home/ubuntu# i-01bda222ee0e85ac7 (jenkins server )
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Initializing git

→ git init



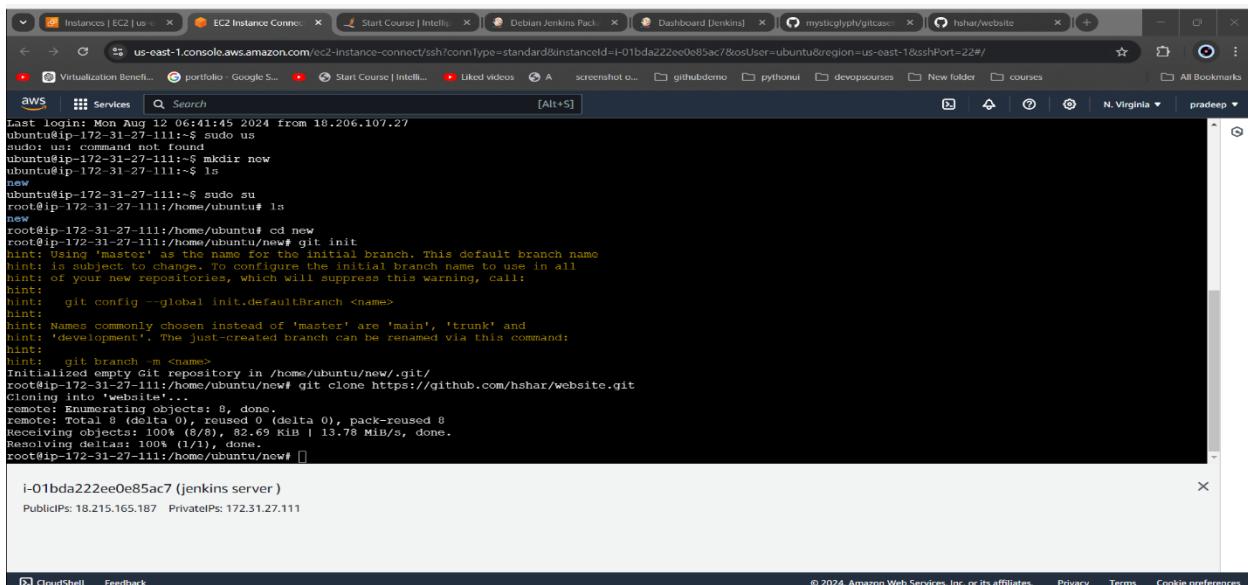
```
*** system restart required ***
Pending kernel upgrade!
Running kernel version:
 6.8.0-1009-aws
Diagnostics:
The currently running kernel version is not the expected kernel version 6.8.0-1013-aws.
last login: Mon Aug 12 06:41:45 2024 from 18.206.107.27
ubuntu@ip-172-31-27-111:~$ sudo su
sudo: us: command not found
ubuntu@ip-172-31-27-111:~$ mkdir new
ubuntu@ip-172-31-27-111:~$ ls
new
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu#
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# 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/new/.git/
root@ip-172-31-27-111:/home/ubuntu/new# 
```

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

Clone the Repository:

Start by cloning the GitHub repository to the local machine:

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



```
Last login: Mon Aug 12 06:41:45 2024 from 18.206.107.27
ubuntu@ip-172-31-27-111:~$ sudo su
sudo: us: command not found
ubuntu@ip-172-31-27-111:~$ mkdir new
ubuntu@ip-172-31-27-111:~$ ls
new
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu#
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# 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/new/.git/
root@ip-172-31-27-111:/home/ubuntu/new# git clone https://github.com/hshar/website.git
Cloning into 'website'...
remote: Enumerating objects: 0, done.
remote: Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), 82.69 KiB | 13.78 MiB/s, done.
Resolving deltas: 100% (1/1), done.
root@ip-172-31-27-111:/home/ubuntu/new# 
```

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

```

sudo: command not found
ubuntu@ip-172-31-27-111:~$ mkdir new
ubuntu@ip-172-31-27-111:~$ ls
new
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# ls
new
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# 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/new/.git/
root@ip-172-31-27-111:/home/ubuntu/new# git clone https://github.com/hshar/website.git
Cloning into 'website'...
remote: Enumerating objects: 8, done.
receiving objects: 100% (8/8), 82.69 KiB | 13.78 MiB/s, done.
Resolving deltas: 100% (1/1), done.
root@ip-172-31-27-111:/home/ubuntu/new# ls
website
root@ip-172-31-27-111:/home/ubuntu/new# 

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

```

To copy the contents of the website directory to /home/ubuntu/new,:

→ cp -r /home/ubuntu/new/website/* /home/ubuntu/new/

```

Usage of /: 15.5% of 18.33GB  Users logged in: 0
Memory usage: 21%          IPv4 address for enX0: 172.31.27.111
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

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

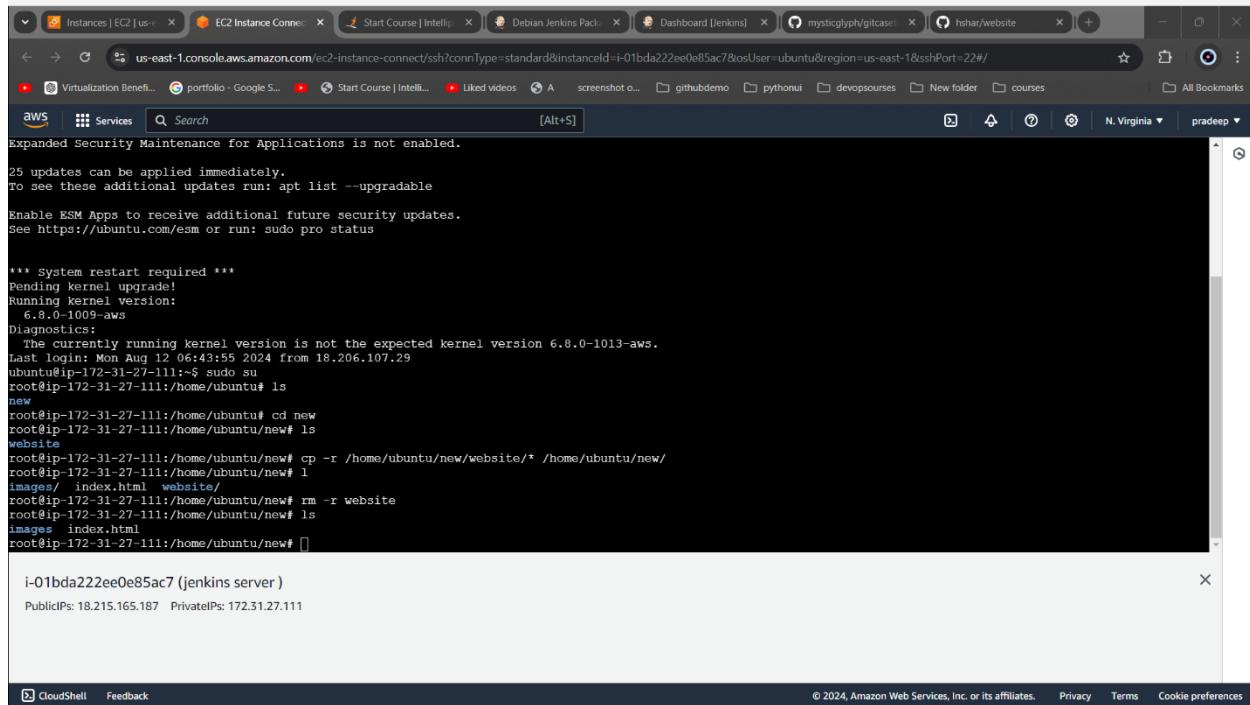
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Pending kernel upgrade!
Running kernel version:
  6.8.0-1009-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1013-aws.
Last login: Mon Aug 12 06:43:55 2024 from 18.206.107.29
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# ls
new
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
website
root@ip-172-31-27-111:/home/ubuntu/new# cp -r /home/ubuntu/new/website/* /home/ubuntu/new/
[]

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

```

Removing the website folder cloned from git repo:

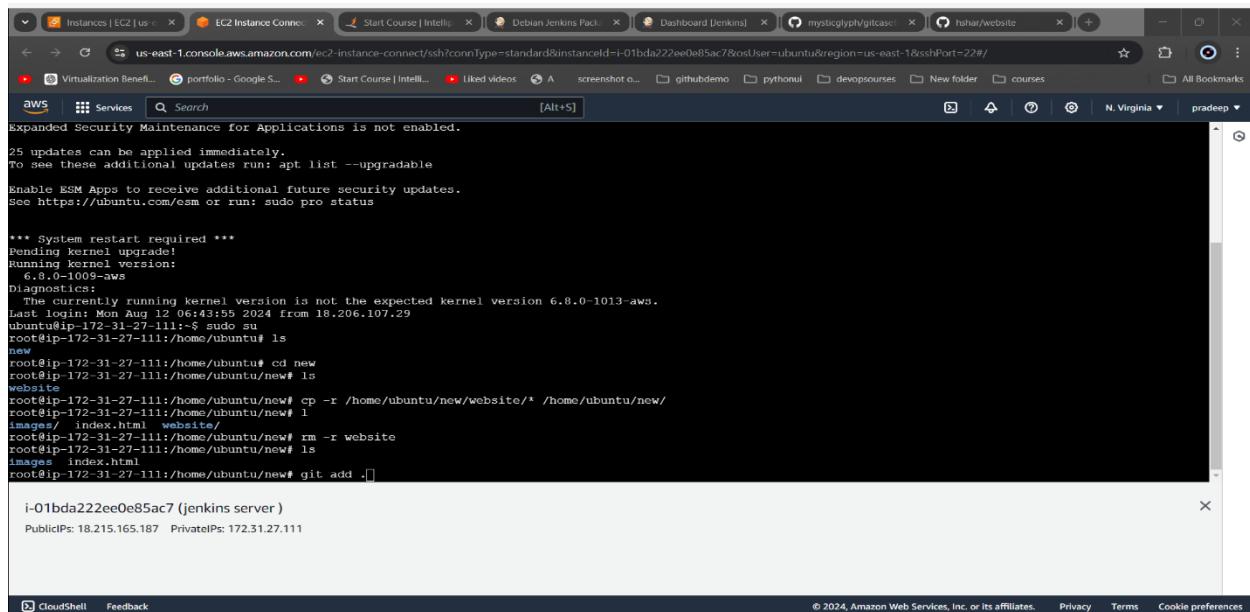


```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda222ee0e85ac7&osUser=ubuntu&region=us-east-1&sshPort=22#  
Virtualization Benefits portfolio - Google S... Start Course | Intelli... Liked videos A screenshot o... githubdemo pythonl devopsources New folder courses All Bookmarks AWS Services Search [Alt+S]  
Expanded Security Maintenance for Applications is not enabled.  
25 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 ***  
Pending kernel upgrade!  
Running kernel version:  
6.8.0-1009-aws  
Diagnostics:  
The currently running kernel version is not the expected kernel version 6.8.0-1013-aws.  
Last login: Mon Aug 12 06:43:55 2024 from 18.206.107.29  
ubuntu@ip-172-31-27-111:~$ sudo su  
root@ip-172-31-27-111:/home/ubuntu# ls new  
root@ip-172-31-27-111:/home/ubuntu# cd new  
root@ip-172-31-27-111:/home/ubuntu/new# ls website  
root@ip-172-31-27-111:/home/ubuntu/new# cp -r /home/ubuntu/new/* /home/ubuntu/new/  
root@ip-172-31-27-111:/home/ubuntu/new# rm -r website  
root@ip-172-31-27-111:/home/ubuntu/new# ls images index.html  
root@ip-172-31-27-111:/home/ubuntu/new#  
root@ip-172-31-27-111:/home/ubuntu/new#  
i-01bda222ee0e85ac7 (jenkins server)  
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

To push the contents of the new directory to a newly created GitHub repository:

Add the Files to the Staging Area:

- Add all the files to the Git staging area:
- Git add .



```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda222ee0e85ac7&osUser=ubuntu&region=us-east-1&sshPort=22#  
Virtualization Benefits portfolio - Google S... Start Course | Intelli... Liked videos A screenshot o... githubdemo pythonl devopsources New folder courses All Bookmarks AWS Services Search [Alt+S]  
Expanded Security Maintenance for Applications is not enabled.  
25 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 ***  
Pending kernel upgrade!  
Running kernel version:  
6.8.0-1009-aws  
Diagnostics:  
The currently running kernel version is not the expected kernel version 6.8.0-1013-aws.  
Last login: Mon Aug 12 06:43:55 2024 from 18.206.107.29  
ubuntu@ip-172-31-27-111:~$ sudo su  
root@ip-172-31-27-111:/home/ubuntu# ls new  
root@ip-172-31-27-111:/home/ubuntu# cd new  
root@ip-172-31-27-111:/home/ubuntu/new# ls website  
root@ip-172-31-27-111:/home/ubuntu/new# cp -r /home/ubuntu/new/* /home/ubuntu/new/  
root@ip-172-31-27-111:/home/ubuntu/new# rm -r website  
root@ip-172-31-27-111:/home/ubuntu/new# ls images index.html  
root@ip-172-31-27-111:/home/ubuntu/new# git add .  
root@ip-172-31-27-111:/home/ubuntu/new#  
i-01bda222ee0e85ac7 (jenkins server)  
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

```

25 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 ***
Pending kernel upgrade!
Running kernel version:
  6.8.0-1009-aws
Diagnostics:
The currently running kernel version is not the expected kernel version 6.8.0-1013-aws.
Last login: Mon Aug 12 06:43:55 2024 from 18.206.107.29
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# ls
new
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
website
root@ip-172-31-27-111:/home/ubuntu/new# cp -r /home/ubuntu/new/website/* /home/ubuntu/new/
root@ip-172-31-27-111:/home/ubuntu/new# 1
images/ index.html website/
root@ip-172-31-27-111:/home/ubuntu/new# rm -r website
root@ip-172-31-27-111:/home/ubuntu/new# ls
images/ index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# [REDACTED]

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

```

Commit the Files:

→ git commit -m "Initial commit with copied content".

```

root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
website
root@ip-172-31-27-111:/home/ubuntu/new# cp -r /home/ubuntu/new/website/* /home/ubuntu/new/
root@ip-172-31-27-111:/home/ubuntu/new# 1
images/ index.html website/
root@ip-172-31-27-111:/home/ubuntu/new# rm -r website
root@ip-172-31-27-111:/home/ubuntu/new# ls
images/ index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "Initial commit with copied content"
[master (root-commit) 463a7bd] Initial commit with copied content
[Committer: root <root@ip-172-31-27-111.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-172-31-27-111:/home/ubuntu/new# [REDACTED]

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

```

Connecting the remote repo to local repo :

→ git remote add origin remote repo url

The screenshot shows a GitHub repository page titled "gitcasetudy". It features two main sections: "Set up GitHub Copilot" and "Add collaborators to this repository". Below these, a large blue box contains instructions for "Quick setup — if you've done this kind of thing before". It provides two options: "...or create a new repository on the command line" and "...or push an existing repository from the command line". Each option includes a code snippet and a copy icon.

```
echo "# gitcasetudy" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/mysticglyph/gitcasetudy.git
git push -u origin main
```

```
git remote add origin https://github.com/mysticglyph/gitcasetudy.git
git branch -M main
```

The screenshot shows an AWS CloudShell terminal window. The user is connected via SSH to an EC2 instance. They have cloned a GitHub repository named "gitcasetudy" into a directory called "new". They then navigate to this directory and run several commands to set up the repository. These include creating files, committing changes, and adding the remote origin. The terminal output shows the creation of files like "index.html" and "images/index.html", and the execution of "git add .", "git commit", and "git remote add origin https://github.com/mysticglyph/gitcasetudy.git". Finally, they run "git push" to upload their changes to the GitHub repository.

```
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
website
root@ip-172-31-27-111:/home/ubuntu/new# cp -r /home/ubuntu/new/* /home/ubuntu/new/
root@ip-172-31-27-111:/home/ubuntu/new# 1
images/ index.html website
root@ip-172-31-27-111:/home/ubuntu/new# rm -r website
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "Initial commit with copied content"
[master (root-commit) 463a7bd] Initial commit with copied content
Committer: root <root@ip-172-31-27-111.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-172-31-27-111:/home/ubuntu/new# git remote add origin https://github.com/mysticglyph/gitcasetudy.git
```

```
i-01bda222ee0e85ac7 (jenkins server)
```

```
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

Instances | EC2 | us... EC2 Instance Connect Start Course | Intelli... Debian Jenkins Pack... Dashboard [Jenkins] mysticglyph/gitcase hshar/website

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda22ee0e85ac7&osUser=ubuntu®ion=us-east-1&sshPort=22#

Virtualization Benefit... portfolio - Google S... Start Course | Intelli... Liked videos A screenshot o... githubdemo pythonui devopsources New folder courses All Bookmarks N. Virginia pradeep

Services Search [Alt+S]

```
root@ip-172-31-27-111:/home/ubuntu/new# cp -r /home/ubuntu/new/* /home/ubuntu/new/
root@ip-172-31-27-111:/home/ubuntu/new# ls
images/ index.html website/
root@ip-172-31-27-111:/home/ubuntu/new# rm -r website
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "Initial commit with copied content"
[master (root-commit) 463a7bd] Initial commit with copied content
Committer: root <root@ip-172-31-27-111.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-172-31-27-111:/home/ubuntu/new# git remote add origin https://github.com/mysticglyph/gitcasestudy.git
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new#
```

i-01bda22ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

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

Connected to remote repo:

Instances | EC2 | us... EC2 Instance Connect Start Course | Intelli... Debian Jenkins Pack... Dashboard [Jenkins] mysticglyph/gitcase hshar/website

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda22ee0e85ac7&osUser=ubuntu®ion=us-east-1&sshPort=22#

Virtualization Benefit... portfolio - Google S... Start Course | Intelli... Liked videos A screenshot o... githubdemo pythonui devopsources New folder courses All Bookmarks N. Virginia pradeep

Services Search [Alt+S]

```
root@ip-172-31-27-111:/home/ubuntu/new# rm -r website
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "Initial commit with copied content"
[master (root-commit) 463a7bd] Initial commit with copied content
Committer: root <root@ip-172-31-27-111.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-172-31-27-111:/home/ubuntu/new# git remote add origin https://github.com/mysticglyph/gitcasestudy.git
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git remote -v
origin https://github.com/mysticglyph/gitcasestudy.git (fetch)
origin https://github.com/mysticglyph/gitcasestudy.git (push)
root@ip-172-31-27-111:/home/ubuntu/new#
```

i-01bda22ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111

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

Push the Files to GitHub:

-> git push origin -u master

```
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# ls
new
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git push gitcasetudy
fatal: 'gitcasetudy' 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-172-31-27-111:/home/ubuntu/new# git push origin gitcasetudy
error: src refspec gitcasetudy does not match any
! [remote rejected] HEAD <https://github.com/mysticglyph/gitcasetudy.git>
root@ip-172-31-27-111:/home/ubuntu/new# git push -u origin gitcasetudy
error: src refspec gitcasetudy does not match any
! [remote rejected] HEAD <https://github.com/mysticglyph/gitcasetudy.git>
root@ip-172-31-27-111:/home/ubuntu/new# git push -u origin main
error: src refspec main does not match any
! [remote rejected] HEAD <https://github.com/mysticglyph/gitcasetudy.git>
root@ip-172-31-27-111:/home/ubuntu/new# git remote -v
origin https://github.com/mysticglyph/gitcasetudy.git (fetch)
origin https://github.com/mysticglyph/gitcasetudy.git (push)
root@ip-172-31-27-111:/home/ubuntu/new# git branch
* master
root@ip-172-31-27-111:/home/ubuntu/new# git push -u origin master
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com': []

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

```
Please make sure you have the correct access rights
and the repository exists.
root@ip-172-31-27-111:/home/ubuntu/new# git push origin gitcasetudy
error: src refspec gitcasetudy does not match any
! [remote rejected] HEAD <https://github.com/mysticglyph/gitcasetudy.git>
root@ip-172-31-27-111:/home/ubuntu/new# git push -u origin gitcasetudy
error: src refspec gitcasetudy does not match any
! [remote rejected] HEAD <https://github.com/mysticglyph/gitcasetudy.git>
root@ip-172-31-27-111:/home/ubuntu/new# git push -u origin main
error: src refspec main does not match any
! [remote rejected] HEAD <https://github.com/mysticglyph/gitcasetudy.git>
root@ip-172-31-27-111:/home/ubuntu/new# git remote -v
origin https://github.com/mysticglyph/gitcasetudy.git (fetch)
origin https://github.com/mysticglyph/gitcasetudy.git (push)
root@ip-172-31-27-111:/home/ubuntu/new# git branch
* master
root@ip-172-31-27-111:/home/ubuntu/new# git push -u origin master
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com':
Enumerating objects: 100% (5/5), 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), 82.40 KiB | 16.48 MiB/s, done.
total 5 (delta 0), reused 0 (delta 0), pack-reused 0
ro https://github.com/mysticglyph/gitcasetudy.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
root@ip-172-31-27-111:/home/ubuntu/new# []

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 18.215.165.187 PrivateIPs: 172.31.27.111
```

Successfully pushed to remote repo:

creating develop branch:

- git checkout -b branch_name

- git checkout -b develop

```

End of Japan/  End of Japan/  Instances | EC2 | us-east-1  EC2 Instance Connect | us-east-1  My Courses | Kpradeepacharya/  +
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda22ee0e85ac7&osUser=ubuntu&region=us-east-1&sshPort=22#/
System load: 0.0 Processes: 113
Usage of /: 15.5% of 10.33GB Users logged in: 0
Memory usage: 16% IPv4 address for enX0: 172.31.27.111
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
25 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: Mon Aug 12 17:15:57 2024 from 18.206.107.29
ubuntu@ip-172-31-27-111:~$ ls
ubuntu@ip-172-31-27-111:~$ cd new
ubuntu@ip-172-31-27-111:/home/ubuntu/new$ ls
index.html
ubuntu@ip-172-31-27-111:/home/ubuntu/new$ git checkout -b develop
[]

i-01bda22ee0e85ac7 (jenkins server )
PublicIPs: 52.72.213.179 PrivateIPs: 172.31.27.111

```

```
System load: 0.0 Processes: 113
Usage of /: 15.5% of 18.33GB Users logged in: 0
Memory usage: 16% IPv4 address for enX0: 172.31.27.111
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

25 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: Mon Aug 12 17:15:57 2024 from 18.206.107.29
ubuntu@ip-172-31-27-111:~$ ubuntu@ip-172-31-27-111:~$ ls
new
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git checkout -b develop
Switched to a new branch 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# git branch
* develop
  master
root@ip-172-31-27-111:/home/ubuntu/new# 

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 52.72.213.179 PrivateIPs: 172.31.27.111
```

Created.

Push the new develop branch to GitHub:

→ git push origin develop

```
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

25 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: Mon Aug 12 17:15:57 2024 from 18.206.107.29
ubuntu@ip-172-31-27-111:~$ ubuntu@ip-172-31-27-111:~$ ls
new
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git checkout -b develop
Switched to a new branch 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# git branch
* develop
  master
root@ip-172-31-27-111:/home/ubuntu/new# git push origin develop
Username for 'https://github.com': []

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 52.72.213.179 PrivateIPs: 172.31.27.111
```

```
Expanded Security Maintenance for Applications is not enabled.  
25 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: Mon Aug 12 17:15:57 2024 from 18.206.107.29  
ubuntu@ip-172-31-27-111:~$ ls  
new  
ubuntu@ip-172-31-27-111:~$ sudo su  
root@ip-172-31-27-111:/home/ubuntu# cd new  
root@ip-172-31-27-111:/home/ubuntu/new# ls  
images index.html  
root@ip-172-31-27-111:/home/ubuntu/new# git checkout -b develop  
Switched to a new branch 'develop'  
root@ip-172-31-27-111:/home/ubuntu/new# git branch  
* develop  
  master  
root@ip-172-31-27-111:/home/ubuntu/new# ls  
images index.html  
root@ip-172-31-27-111:/home/ubuntu/new# git push origin develop  
Username for 'https://github.com': mysticglyph  
Password for 'https://mysticglyph@github.com': []  
  
i-01bda222ee0e85ac7 (jenkins server)  
Public IPs: 52.72.213.179 Private IPs: 172.31.27.111
```

```
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Mon Aug 12 17:15:57 2024 from 18.206.107.29  
ubuntu@ip-172-31-27-111:~$ ls  
new  
ubuntu@ip-172-31-27-111:~$ sudo su  
root@ip-172-31-27-111:/home/ubuntu# cd new  
root@ip-172-31-27-111:/home/ubuntu/new# ls  
images index.html  
root@ip-172-31-27-111:/home/ubuntu/new# git checkout -b develop  
Switched to a new branch 'develop'  
root@ip-172-31-27-111:/home/ubuntu/new# git branch  
* develop  
  master  
root@ip-172-31-27-111:/home/ubuntu/new# ls  
images index.html  
root@ip-172-31-27-111:/home/ubuntu/new# git push origin 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/gitcasestudy/pull/new/develop  
remote:  
To https://github.com/mysticglyph/gitcasestudy.git  
 * [new branch]  develop -> develop  
root@ip-172-31-27-111:/home/ubuntu/new# []  
  
i-01bda222ee0e85ac7 (jenkins server)  
Public IPs: 52.72.213.179 Private IPs: 172.31.27.111
```

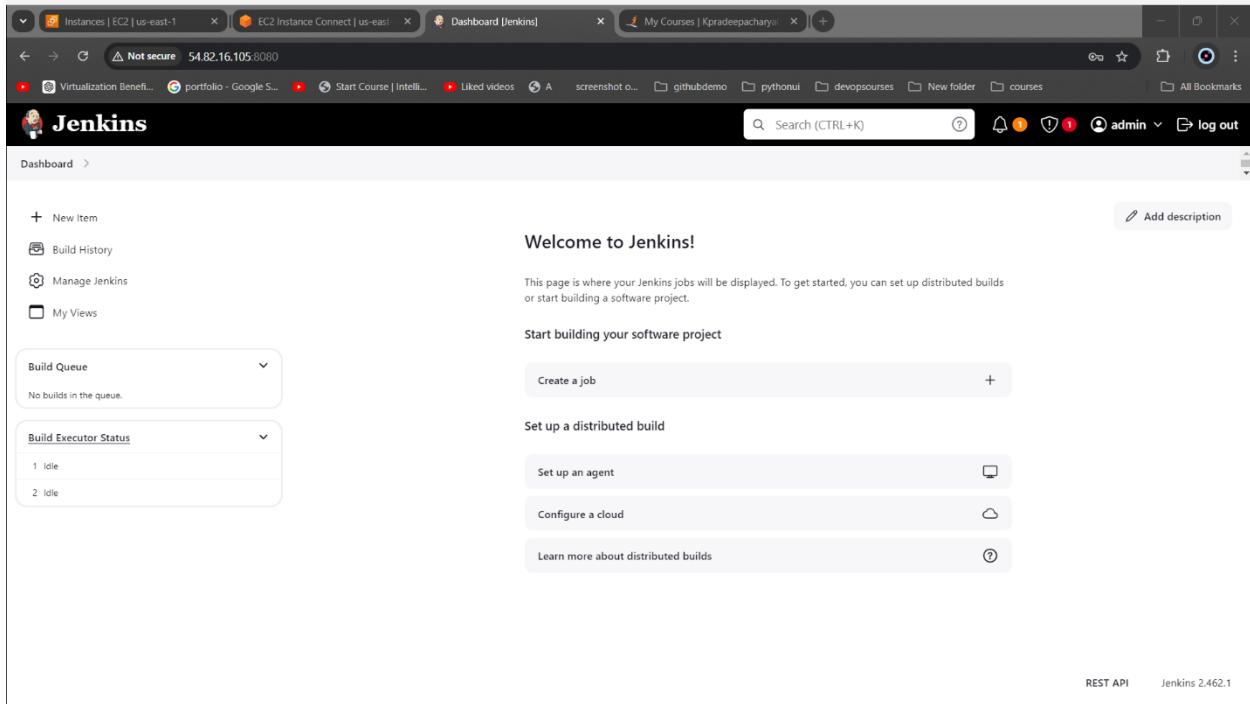
The screenshot shows a web browser window with multiple tabs open. The active tab is for a GitHub repository named 'gitcasetudy' at the URL github.com/mysticglyph/gitcasetudy/tree/develop. The repository is public and contains one branch ('develop'), two files ('index.html' and 'images'), and one commit ('root'). The repository has 0 stars, 0 forks, and 1 watching. There is a 'Contribute' button and a 'README' section with a link to add a README file.

Successfully created.

Git Workflow has implemented.

Task: We now need to create a job that will automatically trigger a code build whenever a commit is made to the master or develop branch

Let's create two separate jobs: one called "master-job" and another called "develop-job".

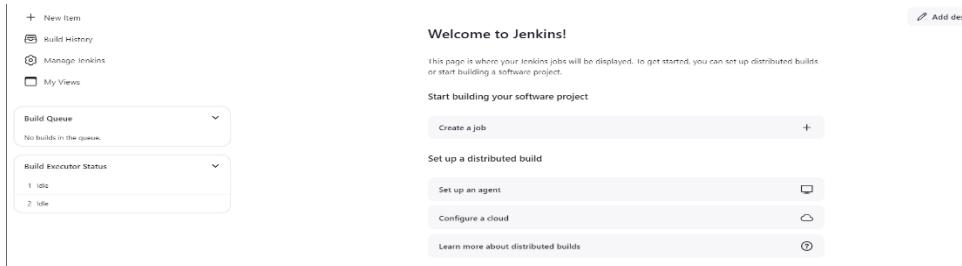


The screenshot shows the Jenkins dashboard at <http://54.82.16.105:8080>. The main heading is "Welcome to Jenkins!". Below it, a message says: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." There are two main sections: "Start building your software project" and "Set up a distributed build". Under "Start building your software project", there is a button labeled "Create a job" with a plus sign. Under "Set up a distributed build", there are three buttons: "Set up an agent" (with a monitor icon), "Configure a cloud" (with a cloud icon), and "Learn more about distributed builds" (with a question mark icon). On the left side, there are links for "New item", "Build History", "Manage Jenkins", and "My Views". A sidebar shows "Build Queue" (No builds in the queue) and "Build Executor Status" (1 idle, 2 idle).

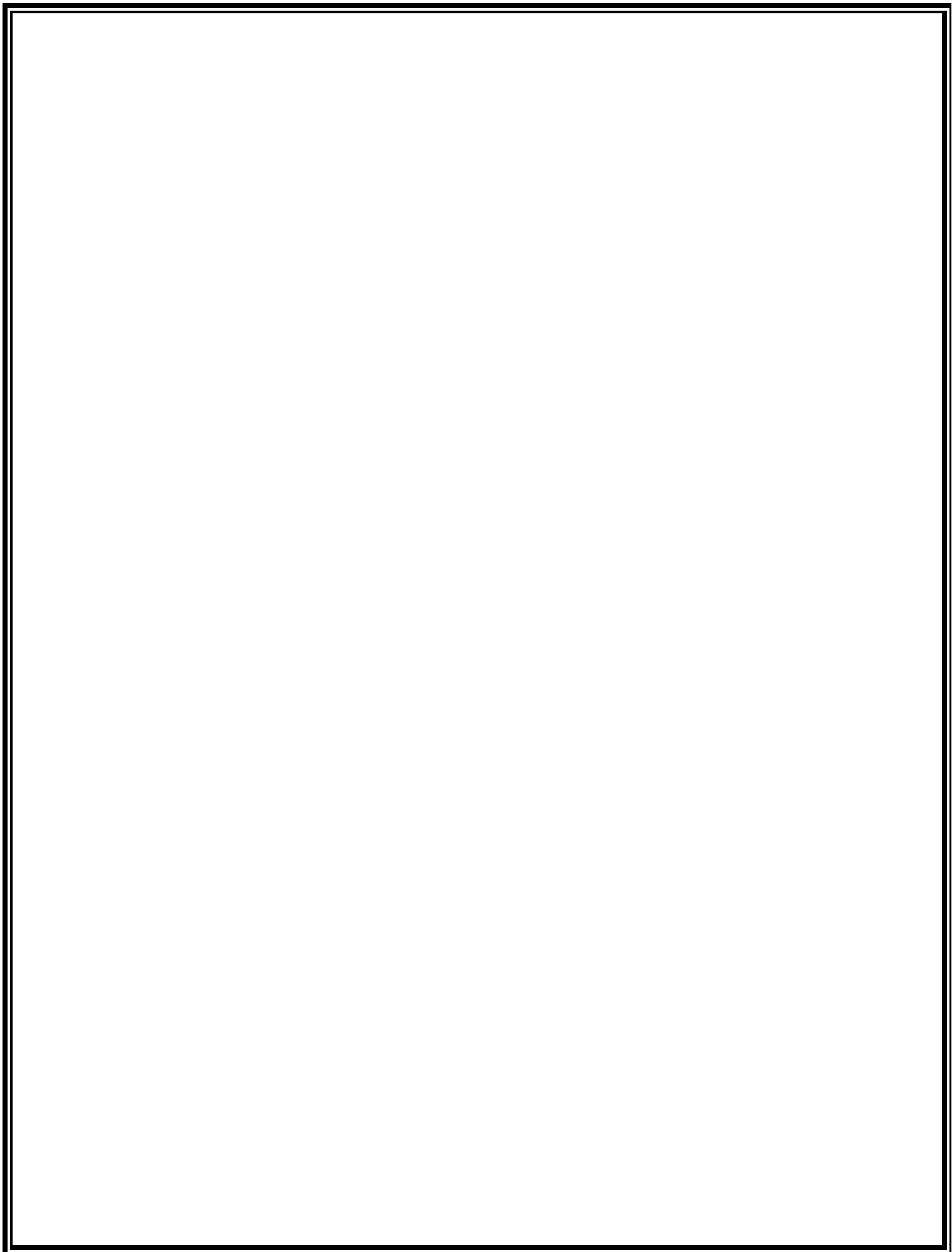
Creating a Master-Job:

The objective of this job, named Master Job, is to automatically trigger a code build whenever a commit is made to the master branch. If a commit is made to the master branch, the job will build and publish the website accordingly.

Click on new item or create a job for creating new job:



This screenshot is identical to the one above, showing the Jenkins dashboard with the "Welcome to Jenkins!" page. It includes the same sections: "Build Queue" (No builds in the queue), "Build Executor Status" (2 idle), and "Start building your software project" with its various sub-options. The overall layout and content are the same as the first screenshot.



Enter an item name and select the job type:

New Item

Enter an item name

mater-job

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.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

Click ok.

Configure

General

Enabled

Description

Plain text [Preview](#)

Discard old builds [?](#)

GitHub project

This project is parameterized [?](#)

Throttle builds [?](#)

Execute concurrent builds if necessary [?](#)

Advanced [▼](#)

Save Apply

Job is created.

In order to trigger the job when a push event occurs, we need to create a webhook in GitHub.

gitcasetudy

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 2 Branches 0 Tags

root Initial commit with copied content 463a7bd · 5 days ago 1 Commit

images Initial commit with copied content 5 days ago

index.html Initial commit with copied content 5 days ago

README

Add a README

No description, website, or topics provided.

Activity 0 stars 1 watching 0 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

Languages HTML 100.0%

Goto settings:

General

Access Collaborators Moderation options

Branches Tags Rules Actions Webhooks Environments Codespaces Pages

Repository name gitcasetudy Rename

Template repository

Require contributors to sign off on web-based commits

Default branch master

Social preview

Upload an image to customize your repository's social media preview. Images should be at least 640x320px (1280x640px for best display). Download template

select the option -> webhook

The screenshot shows the GitHub Settings page for the repository 'gitcasetudy'. The left sidebar has 'Webhooks' selected under 'Code and automation'. The main content area is titled 'Webhooks' and contains a brief description: 'Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#)'. A 'Add webhook' button is located at the top right of this section.

Add webhook:

The screenshot shows the 'Add webhook' form within the GitHub Settings page for the repository 'gitcasetudy'. The left sidebar has 'Webhooks' selected. The main form area is titled 'Webhooks / Add webhook' and contains the following fields:

- Payload URL ***:
- Content type ***:
- Secret**:
- SSL verification**:
 - Enable SSL verification
 - Disable (not recommended)
- Which events would you like to trigger this webhook?**
 - Just the push event.
 - Send me everything.

Payload URL:

We need to provide the Jenkins URL in the webhook configuration.

Content type:

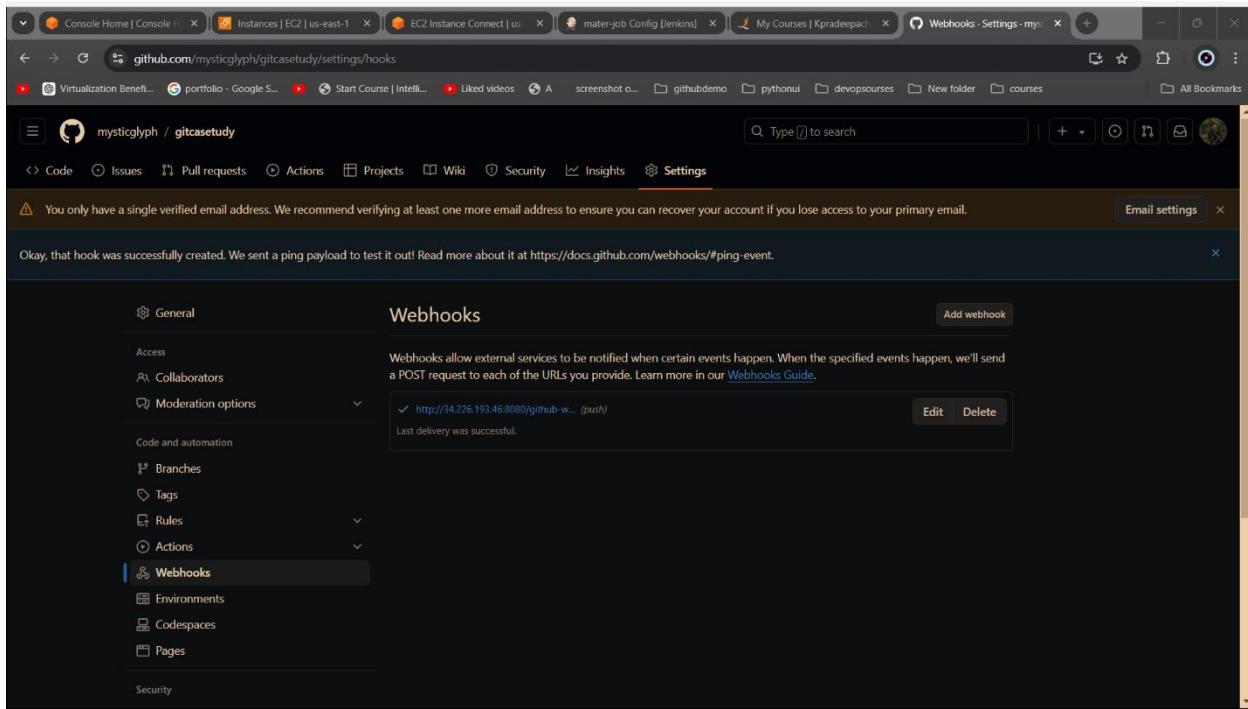
Application/json

The screenshot shows the GitHub settings interface for adding a webhook. On the left, a sidebar lists various GitHub features like General, Access, Collaborators, and Webhooks (which is currently selected). The main panel is titled "Webhooks / Add webhook". It contains a note about sending POST requests with event details. The "Payload URL" field is populated with "http://34.226.193.46:8080/github-webhook/". The "Content type" dropdown is set to "application/json". There is also a "Secret" input field and an "SSL verification" section with a radio button for "Enable SSL verification" (which is selected) and "Disable (not recommended)". Below these, there's a section for triggering events with options for "Just the push event.", "Send me everything.", and "Let me select individual events." A large blue "Add webhook" button is at the bottom.

Which events would you like to trigger this webhook:

push

This screenshot continues from the previous one, showing the event selection step. The "Just the push event." option is selected. Below it, there are three other options: "Send me everything.", "Let me select individual events.", and "Active" (which is checked). A note below the "Active" checkbox states, "We will deliver event details when this hook is triggered." At the bottom is a large blue "Add webhook" button.



Webhook is created.

After setting up the webhook, the next step is to install Docker on the local EC2 instance. Once Docker is installed, start the Docker service. Following this, download the necessary plugins to create a Docker image and push it to Docker Hub. After successfully pushing the image, install a plugin to create a Docker container using the image stored in Docker Hub. Finally, deploy a Docker container using the pushed image.

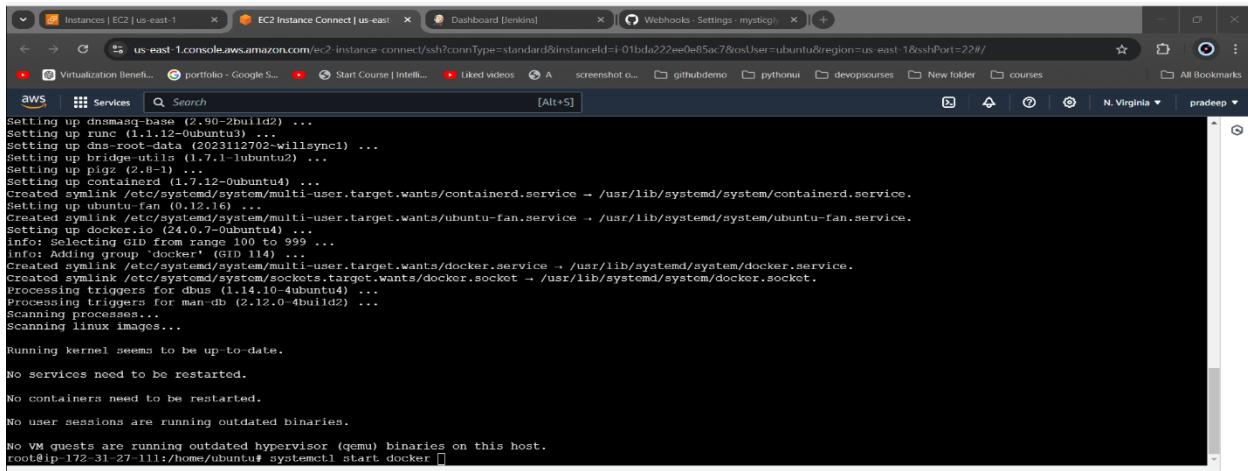
Installing docker in local host:

-> sudo apt install docker.io -y

A screenshot of an AWS CloudShell terminal window titled 'Webhooks - Settings - mystic...'. The terminal shows system information for an Ubuntu Pro instance. It then displays the output of the command 'sudo apt install docker.io -y', which includes the download and installation process. The terminal ends with the prompt 'i-01bda222ee0e85ac7 (jenkins server)'.

Start docker:

→ `systemctl start docker`



```
Setting up dnsmasq-base (2.90-2build2) ...
Setting up runit (1.1.12-0ubuntu3) ...
Setting up dns-root-data (2023112702-willsync1) ...
Setting up bridge-utils (1.7.1-1ubuntu2) ...
Setting up pigz (2.8-1) ...
Setting up containerd (1.7.12-0ubuntu4) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.7-0ubuntu4) ...
Processing triggers for dbus (1.14.10-4ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

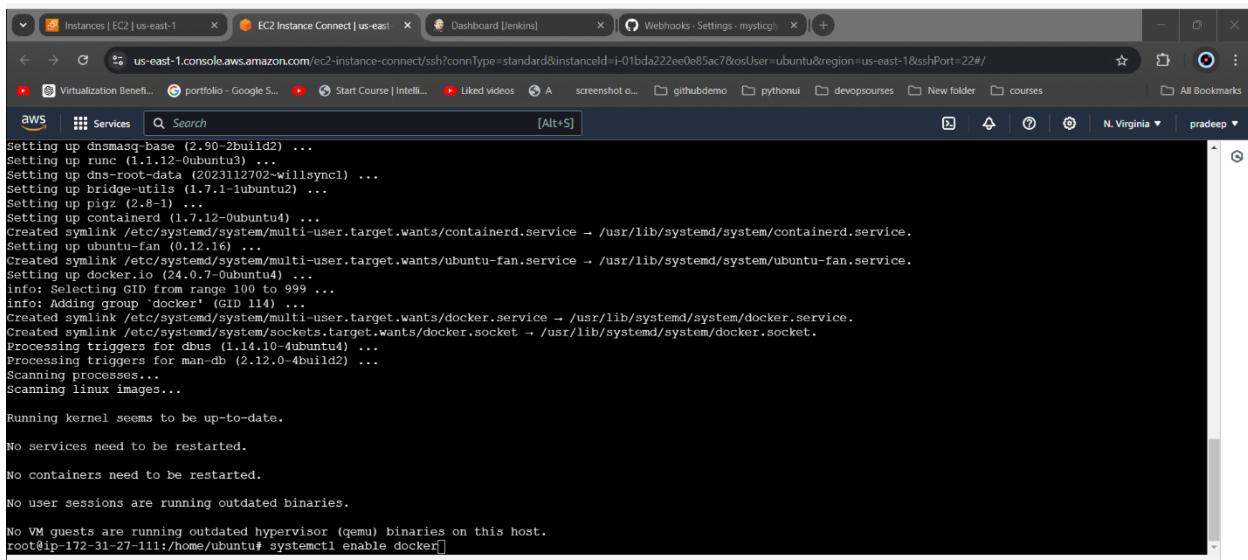
No services need to be restarted.

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-172-31-27-111:/home/ubuntu# systemctl start docker
```

→ `systemctl enable docker`



```
Setting up dnsmasq-base (2.90-2build2) ...
Setting up runit (1.1.12-0ubuntu3) ...
Setting up dns-root-data (2023112702-willsync1) ...
Setting up bridge-utils (1.7.1-1ubuntu2) ...
Setting up pigz (2.8-1) ...
Setting up containerd (1.7.12-0ubuntu4) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.7-0ubuntu4) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group 'docker' (GID 114) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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-172-31-27-111:/home/ubuntu# systemctl enable docker
```



```
running kernel seems to be up-to-date.

No services need to be restarted.

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-172-31-27-111:/home/ubuntu# systemctl enable docker
root@ip-172-31-27-111:/home/ubuntu#
```

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

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Available plugins - Plugins" and has the URL "54.242.24.146:8080/manage/pluginManager/available". The page content is as follows:

Docker Pipeline 580.vc0c340686b_54

pipeline DevOps Deployment docker

Build and use Docker containers from pipelines.

CloudBees Docker Build and Publish 1.4.0

Build Tools docker

This plugin enables building Dockerfile based projects, as well as publishing of

Amazon ECR 1.136.v914ea_5948634

aws

Installing this plugin:

The screenshot shows the Jenkins "Manage Jenkins > Plugins" page. The "Download progress" section is highlighted. The page content is as follows:

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SSH server Pending

CloudBees Docker Build and Publish 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 2.462.1

The screenshot shows the Jenkins 'Plugins' management interface. On the left, there's a sidebar with links: 'Updates' (with a red exclamation mark), 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress' (which is selected). The main content area is titled 'Download progress' and shows the following status:

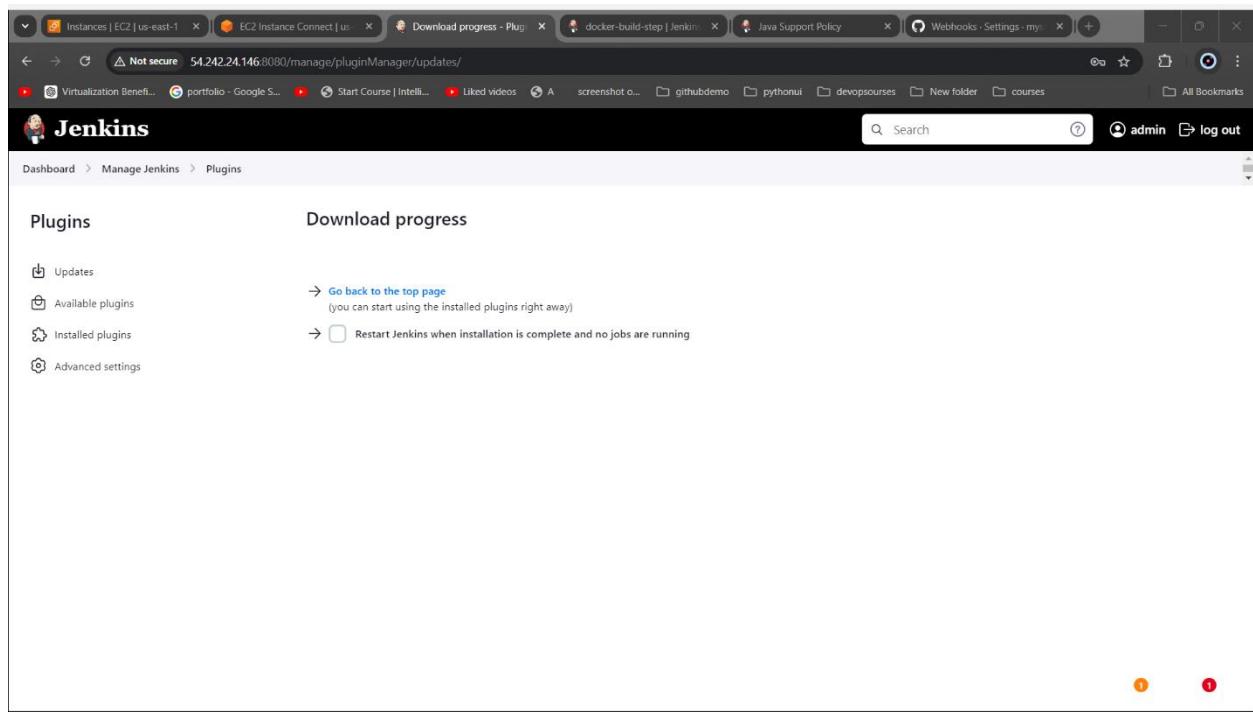
Category	Status
Preparation	<ul style="list-style-type: none">Checking internet connectivityChecking update center connectivitySuccess
SSH server	Pending
CloudBees Docker Build and Publish	Pending
Loading plugin extensions	Pending

Below this, 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 checked checkbox).

At the bottom right of the page, it says 'REST API' and 'Jenkins 2.462.1'.

The screenshot shows the Jenkins restart page. It features a large 'Jenkins' logo at the top. Below it, the text 'Please wait while Jenkins is restarting ...' is displayed. A small note below states 'Your browser will reload automatically when Jenkins is ready.' At the bottom, there is a green button labeled 'Safe Restart' with the subtext 'Builds on agents can usually continue.'

Plugin is installing.



Plugin is installed.

The screenshot shows the Jenkins Dashboard at <http://54.242.24.146:8080>. The main area displays a table of build items. One item, 'mater-job', is listed with the status 'N/A' under 'Last Success' and 'Last Failure'. The dashboard also includes sections for 'Build Queue' (empty) and 'Build Executor Status' (2 idle). The top navigation bar shows tabs for 'Instances | EC2 | us-east-1', 'EC2 instance Connect | us-east-1', 'Dashboard [Jenkins]', and 'Webhooks - Settings - mysticly'. The bottom right corner indicates 'REST API' and 'Jenkins 2.462.1'.

→ manage Jenkins :

The screenshot shows the 'Manage Jenkins' page at <http://54.242.24.146:8080/manage>. The main header says 'Manage Jenkins'. A prominent message states: 'Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)'. Below this are buttons for 'Set up agent', 'Set up cloud', and 'Dismiss'. A red warning box highlights: 'Java 11 end of life in Jenkins' and 'You are running Jenkins on Java 11, support for which will end on or after Sep 30, 2024. Refer to [the documentation](#) for more details.' It includes 'More Info' and 'Ignore' buttons. The page is divided into sections: 'System Configuration' (with 'System', 'Nodes', 'Tools', 'Clouds', and 'Plugins' sub-sections), 'Security', 'Credentials', and 'Credential Providers'. The left sidebar has links for 'New Item', 'Build History', 'Manage Jenkins' (which is selected), and 'My Views'. The top navigation bar is identical to the dashboard.

→ plugins:

The screenshot shows the Jenkins plugin manager interface. The left sidebar has 'Updates' selected. A search bar at the top right says 'Search (CTRL+K)'. Below it, a table lists several available updates:

Name	Released	Installed
commons-lang3 v3.x Jenkins API	3 days 4 hr ago	3.14.0-76.vda_5591261cfef
ECharts API	3 days 2 hr ago	5.5.0-1
Font Awesome API	3 days 2 hr ago	6.5.2-1
Mina SSHD API :: Common	5 days 19 hr ago	2.13.1-117.v2f1a_b_66ff91d
Mina SSHD API :: Core	5 days 19 hr ago	2.13.1-117.v2f1a_b_66ff91d
Pipeline: API		

→ available plugins:

The screenshot shows the Jenkins plugin manager interface. The left sidebar has 'Available plugins' selected. A search bar at the top right says 'Search (CTRL+K)'. Below it, a table lists several available plugins:

Install	Name	Released
<input type="checkbox"/>	Pipeline: REST API	9 mo 21 days ago
<input type="checkbox"/>	Pipeline: Stage View	9 mo 21 days ago
<input type="checkbox"/>	Oracle Java SE Development Kit Installer	12 days ago
<input type="checkbox"/>	Command Agent Launcher	12 days ago
<input type="checkbox"/>	JSch dependency	7 mo 18 days ago
<input type="checkbox"/>	SSH server	2 mo 10 days ago

Downloading docker—build-step plugin:

The screenshot shows a browser window with multiple tabs open. The active tab is 'plugins.jenkins.io/docker-build-step/'. The page title is 'docker-build-step'. A warning message at the top states: 'The Jenkins project announced an unresolved security vulnerability affecting the current version of this plugin (why?): CSRF vulnerability and missing permission check'. Below this, there's a 'How to install' button. To the right, it says 'Version: 2.12' was released 3 months ago for Jenkins 2.387.3. It also indicates that 5.31% of controllers use this plugin. There are sections for 'Plugin Information', 'Commands' (listing 17 items), 'Links' (GitHub, Jira, JavaDoc), 'Labels' (Build Tools, docker), and 'Maintainers' (vjuranek). A 'View detailed version information' link is also present.

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
- start exec command

The screenshot shows the Jenkins plugin manager interface. The left sidebar has tabs for 'Updates', 'Available plugins' (which is selected), 'Installed plugins', and 'Advanced settings'. The main area is titled 'Plugins' and contains a search bar with 'docker' typed in. Below the search bar is a message: 'Search for available Docker commands from pipelines.' A large orange button labeled 'Install' is at the top right. The results list includes:

- Docker API** 3.3.6-90.ve7c5c7535ddd (Library plugins [for use by other plugins] docker) - Last updated 2 mo 18 days ago.
- docker-build-step** 2.12 (Build Tools docker) - Last updated 2 mo 23 days ago. Includes a warning: 'Warning: This plugin version may not be safe to use. Please review the following security notices:' with a link to 'CSRF vulnerability and missing permission check'.
- CloudBees Docker Build and Publish** 1.4.0 (Build Tools docker) - Last updated 1 yr 12 mo ago.
- Amazon ECR** 1.136.v914ea_5948634 (aws) - Last updated 1 mo 22 days ago.
- Docker Compose Build Step** 1.0 (Docker Compose plugin for Jenkins) - Last updated 6 yr 1 mo ago.

The screenshot shows the Jenkins plugin manager interface with the 'Download progress' tab selected. The left sidebar has tabs for 'Updates', 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. The main area is titled 'Download progress' and shows the status of plugin updates. It includes a 'Preparation' section with a bulleted list:

- Checking internet connectivity
- Checking update center connectivity
- Success

Below this is a table of download progress for various plugins:

Plugin	Status
Authentication Tokens API	Success
Docker Commons	Pending
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

At the bottom, there are two links:

- 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

At the very bottom right, it says 'REST API' and 'Jenkins 2.462.1'.

Plugin is downloaded:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "Download progress - Plugin" and has the URL "54.242.24.146:8080/manage/pluginManager/updates/". The page content is titled "Download progress" and includes a sidebar with links for "Updates", "Available plugins", "Installed plugins", and "Advanced settings". Below the sidebar, there are two informational messages with arrows: one pointing to "Go back to the top page" (you can start using the installed plugins right away) and another pointing to a checkbox labeled "Restart Jenkins when installation is complete and no jobs are running". At the bottom right of the page, it says "REST API" and "Jenkins 2.462.1".

Writing and Pushing the Dockerfile to GitHub Repository:

We need to write a Dockerfile that defines the instructions to create a Docker container. Once the Dockerfile is complete, it should be pushed to the GitHub repository. This allows the Dockerfile to be used for automated builds and deployments as part of the CI/CD pipeline.

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

Adding Dockerfile to staging area:

→ git add .

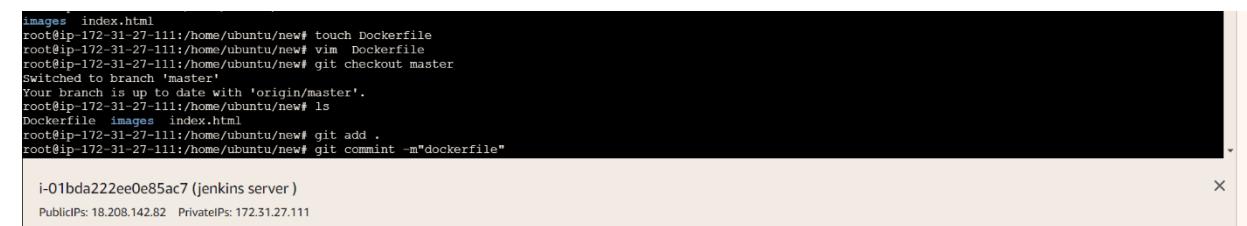


```
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# cd new
bash: cd: new: No such file or directory
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# touch Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# vim Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .

i-01bda222ee0e85ac7 (jenkins server )
PublicIPs: 18.208.142.82 PrivateIPs: 172.31.27.111

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

Committing the changes:



```
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# touch Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# vim Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m"dockerfile"

i-01bda222ee0e85ac7 (jenkins server )
PublicIPs: 18.208.142.82 PrivateIPs: 172.31.27.111
```

A screenshot of the AWS CloudShell interface. The terminal window shows a root session on an EC2 instance. The user has run several commands to edit a Dockerfile, check it out, add it to the repository, and commit it. The commit message is "jenkins server". The terminal also displays the public and private IP addresses of the instance. At the bottom, there are links for CloudShell and Feedback, along with standard AWS footer links for Privacy, Terms, and Cookie preferences.

```
root@ip-172-31-27-111:/home/ubuntu/new# vim Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m"dockerfile"
git: 'commit' is not a git command. See 'git --help'.

The most similar command is
    commit
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "dockerfile"
[master 7f50b47] dockerfile
Committer: root <root@ip-172-31-27-111.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

i-01bda222ee0e85ac7 (jenkins server)

PublicIPs: 18.208.142.82 PrivateIPs: 172.31.27.111
```

Changes successfully committed.

A second screenshot of the AWS CloudShell interface, identical to the first one. It shows the same sequence of commands being run on the EC2 instance, resulting in a successful git commit of the Dockerfile changes. The commit message is "jenkins server". The terminal output at the bottom includes the instance's public and private IP addresses.

```
root@ip-172-31-27-111:/home/ubuntu/new# vim Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m"dockerfile"
git: 'commit' is not a git command. See 'git --help'.

The most similar command is
    commit
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "dockerfile"
[master 7f50b47] dockerfile
Committer: root <root@ip-172-31-27-111.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

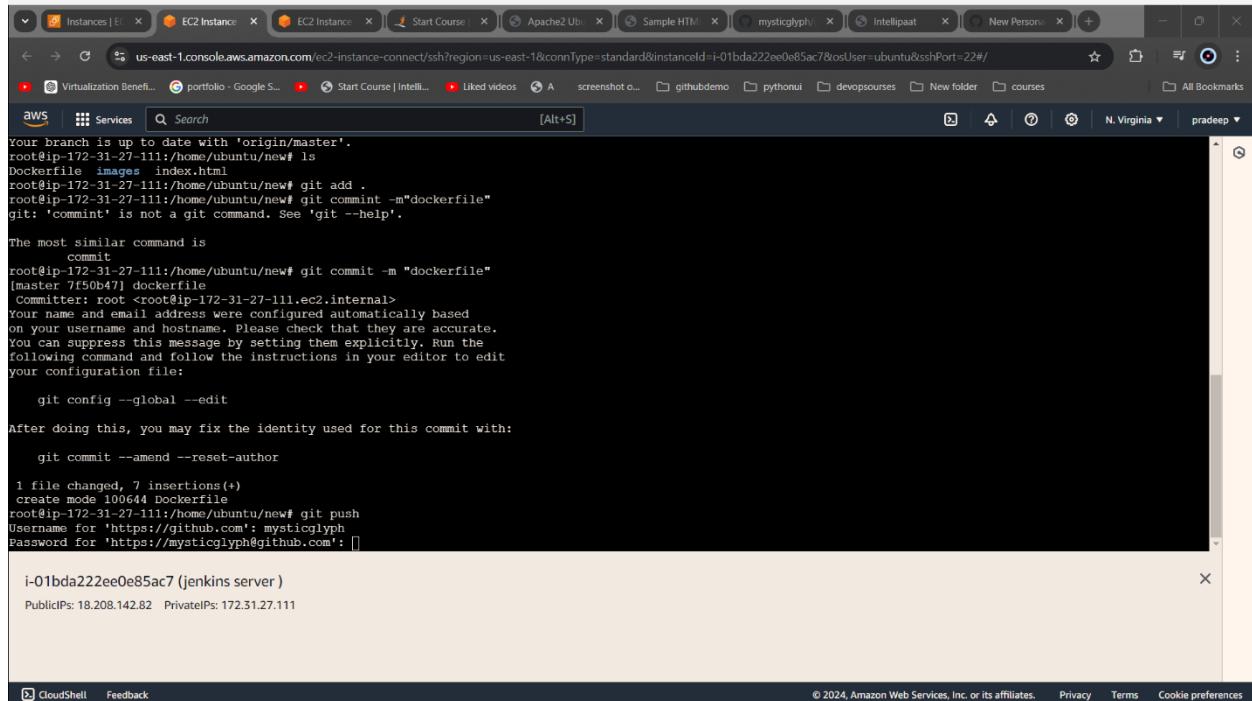
1 file changed, 7 insertions(+)
create mode 100644 Dockerfile

i-01bda222ee0e85ac7 (jenkins server)

PublicIPs: 18.208.142.82 PrivateIPs: 172.31.27.111
```

Pushing to remote repo:

→ git push



```
Your branch is up to date with 'origin/master'.
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile  images  index.html
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "dockerfile"
git: 'commit' is not a git command. See 'git --help'.

The most similar command is
  commit
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "dockerfile"
[master 7f50b47] dockerfile
  Committer: root <root@ip-172-31-27-111.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

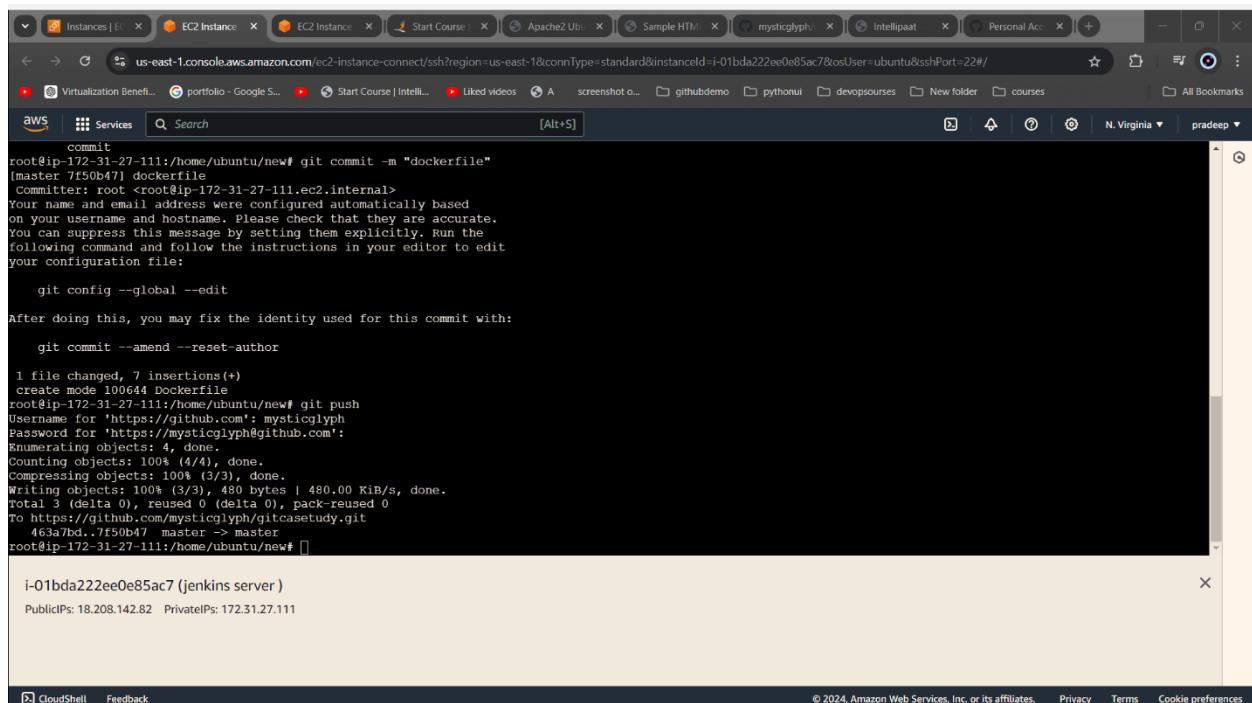
After doing this, you may fix the identity used for this commit with:

  git commit --amend --reset-author

  1 file changed, 7 insertions(+)
  create mode 100644 Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# git push
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com': []

i-01bda222ee0e85ac7 (jenkins server )
PublicIPs: 18.208.142.82 PrivateIPs: 172.31.27.111
```

Successfully pushed to master branch:



```
commit
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "dockerfile"
[master 7f50b47] dockerfile
  Committer: root <root@ip-172-31-27-111.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-172-31-27-111:/home/ubuntu/new# git push
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 480 bytes | 480.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mysticglyph/gitcasestudy.git
  463a7bd..7f50b47  master -> master
root@ip-172-31-27-111:/home/ubuntu/new# []

i-01bda222ee0e85ac7 (jenkins server )
PublicIPs: 18.208.142.82 PrivateIPs: 172.31.27.111
```

Instances | EC2 Instance Connect | EC2 Instance | Start Course | Apache2 Ubuntu | Sample HTML | mysticglyph/ | Intellipaat | Personal Account

github.com/mysticglyph/gitcasetudy

Virtualization Benefits | portfolio - Google Slides | Start Course | IntelliJ | Liked videos | A screenshot of | githubdemo | pythonui | devopsources | New folder | courses | All Bookmarks

mysticglyph / gitcasetudy

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

You only have a single verified email address. We recommend verifying at least one more email address to ensure you can recover your account if you lose access to your primary email.

Email settings

gitcasetudy Public

master 2 Branches 0 Tags

root dockerfile 7/15/04 7 - 7 minutes ago 2 Commits

images Initial commit with copied content 2 weeks ago

Dockerfile dockerfile 7 minutes ago

index.html Initial commit with copied content 2 weeks ago

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Dockerfile 57.8% HTML 42.2%

Dockerfile is pushed to remote repo.

We pushed the Dockerfile to our master branch; we need to do the same for our develop branch as well.

→ Git checkout develop

Instances | EC2 Instance Connect | us-east-1 | mysticglyph/gitcasetudy

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01bda22ee0e85ac7&osUser=ubuntu®ion=us-east-1&sshPort=22#

Virtualization Benefits | portfolio - Google Slides | Start Course | IntelliJ | Liked videos | A screenshot of | githubdemo | pythonui | devopsources | New folder | courses | All Bookmarks

aws Services Search [Alt+S]

N. Virginia pradeep

System information as of Sat Aug 24 05:19:40 UTC 2024

```
System load: 0.15 Processes: 111
Usage of /: 17.5% of 18.33GB Users logged in: 0
Memory usage: 50% IPv4 address for enX0: 172.31.27.111
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.

25 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 Aug 23 13:10:25 2024 from 18.206.107.29
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# ls
new
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# git checkout develop
Switched to branch 'develop'
root@ip-172-31-27-111:/home/ubuntu/new#
```

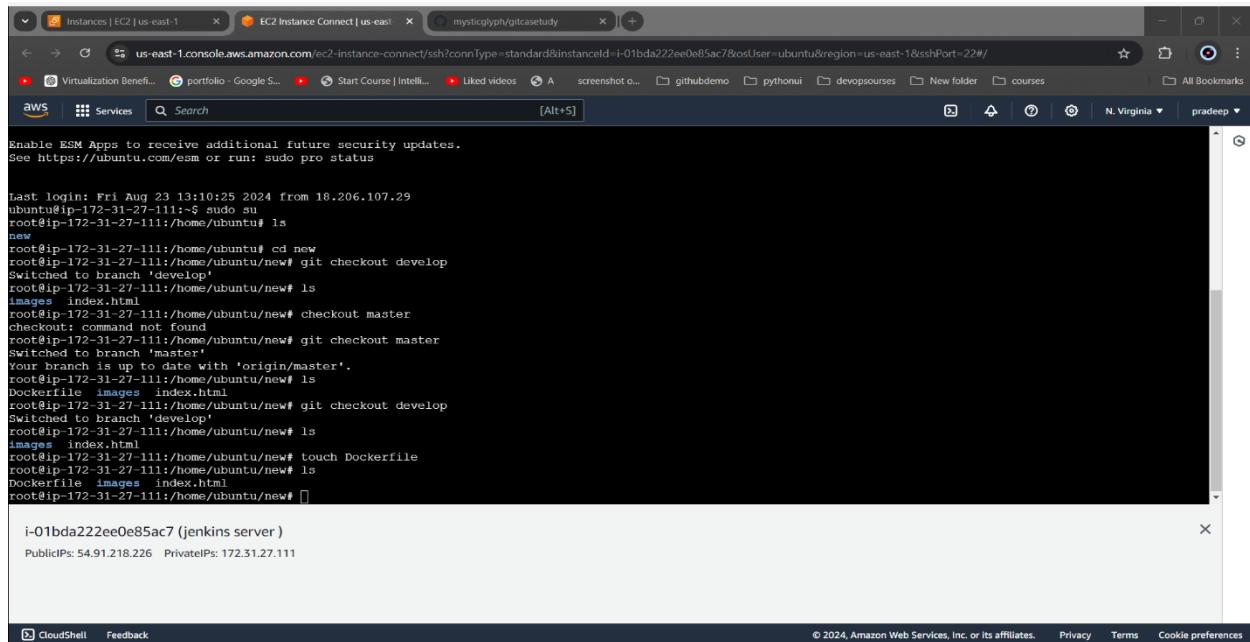
i-01bda22ee0e85ac7 (jenkins server)

PublicIPs: 54.91.218.226 PrivateIPs: 172.31.27.111

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

Creating dockerfile:

→ Touch dockerfile



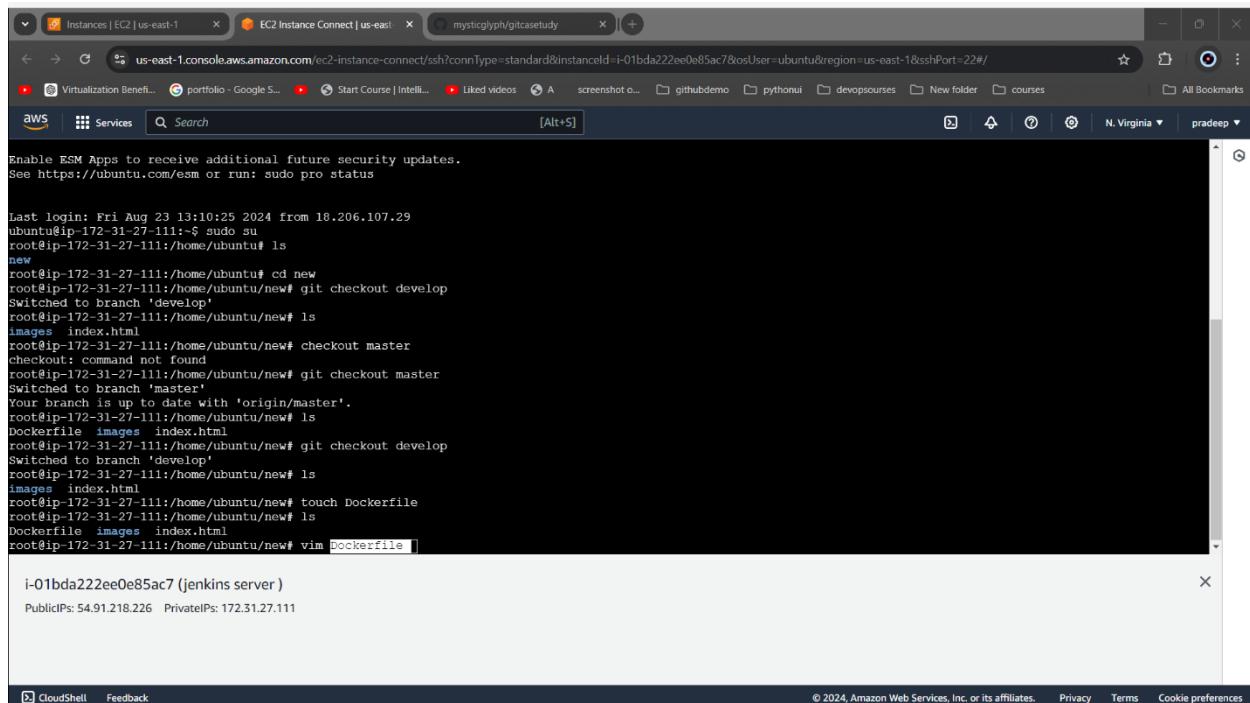
The screenshot shows a terminal window within an AWS CloudShell interface. The terminal output is as follows:

```
Last login: Fri Aug 23 13:10:25 2024 from 18.206.107.29
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# ls
new
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# git checkout develop
Switched to branch 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# checkout master
checkout: command not found
root@ip-172-31-27-111:/home/ubuntu/new# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git checkout develop
Switched to branch 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# touch Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# vim Dockerfile
```

i-01bda222ee0e85ac7 (jenkins server)

Public IPs: 54.91.218.226 Private IPs: 172.31.27.111

Open dockerfile with vim texteditor:

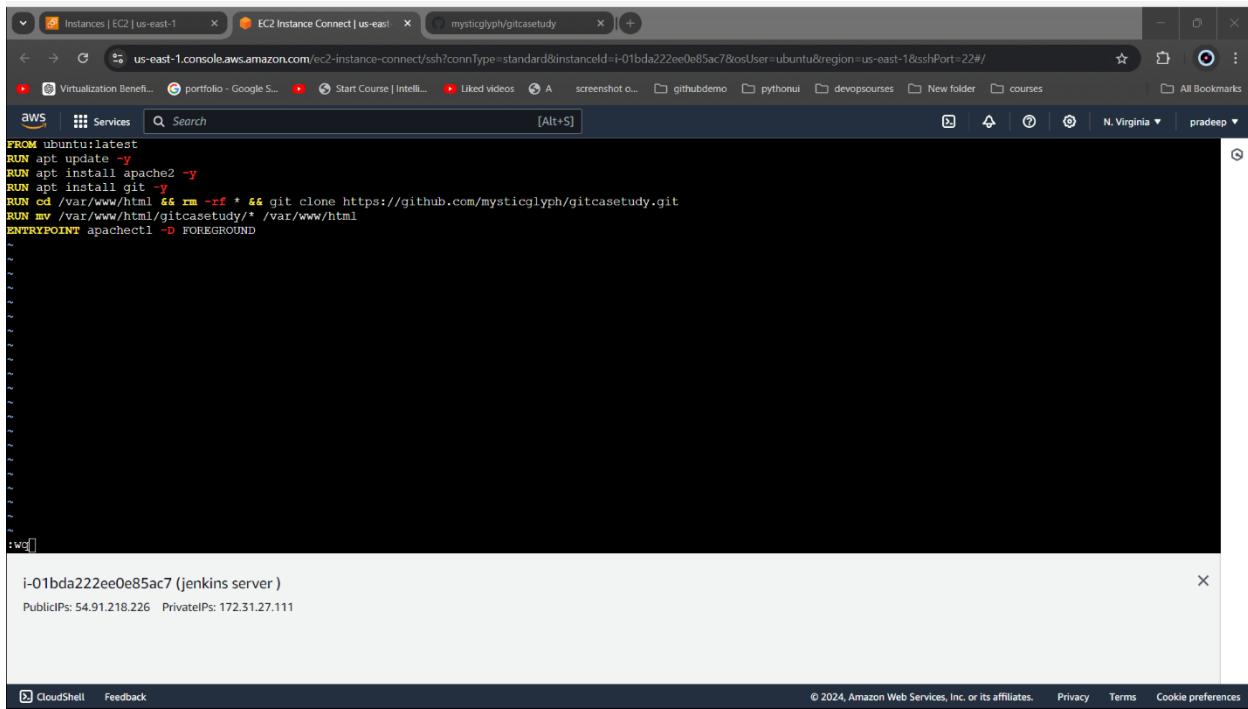


The screenshot shows a terminal window within an AWS CloudShell interface. The terminal output is as follows:

```
Last login: Fri Aug 23 13:10:25 2024 from 18.206.107.29
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# ls
new
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# git checkout develop
Switched to branch 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# checkout master
checkout: command not found
root@ip-172-31-27-111:/home/ubuntu/new# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git checkout develop
Switched to branch 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# touch Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# vim Dockerfile
```

i-01bda222ee0e85ac7 (jenkins server)

Public IPs: 54.91.218.226 Private IPs: 172.31.27.111



A screenshot of the AWS CloudShell interface. The terminal window shows a Dockerfile with several lines of code. The file starts with a FROM instruction for Ubuntu:latest, followed by RUN commands for apt update, apt install apache2, apt install git, and cd /var/www/html. It then uses rm -rf * to remove existing files, git clone https://github.com/mysticglyph/gitcasestudy.git to download a new repository, and mv to move the cloned directory to /var/www/html. The final line is ENTRYPOINT apachectl -D FOREGROUND. Below the terminal, a status bar displays the instance ID (i-01bda222ee0e85ac7), Jenkins server status, and public/private IP addresses (54.91.218.226 and 172.31.27.111). The bottom navigation bar includes CloudShell, Feedback, and links to AWS Privacy, Terms, and Cookie preferences.

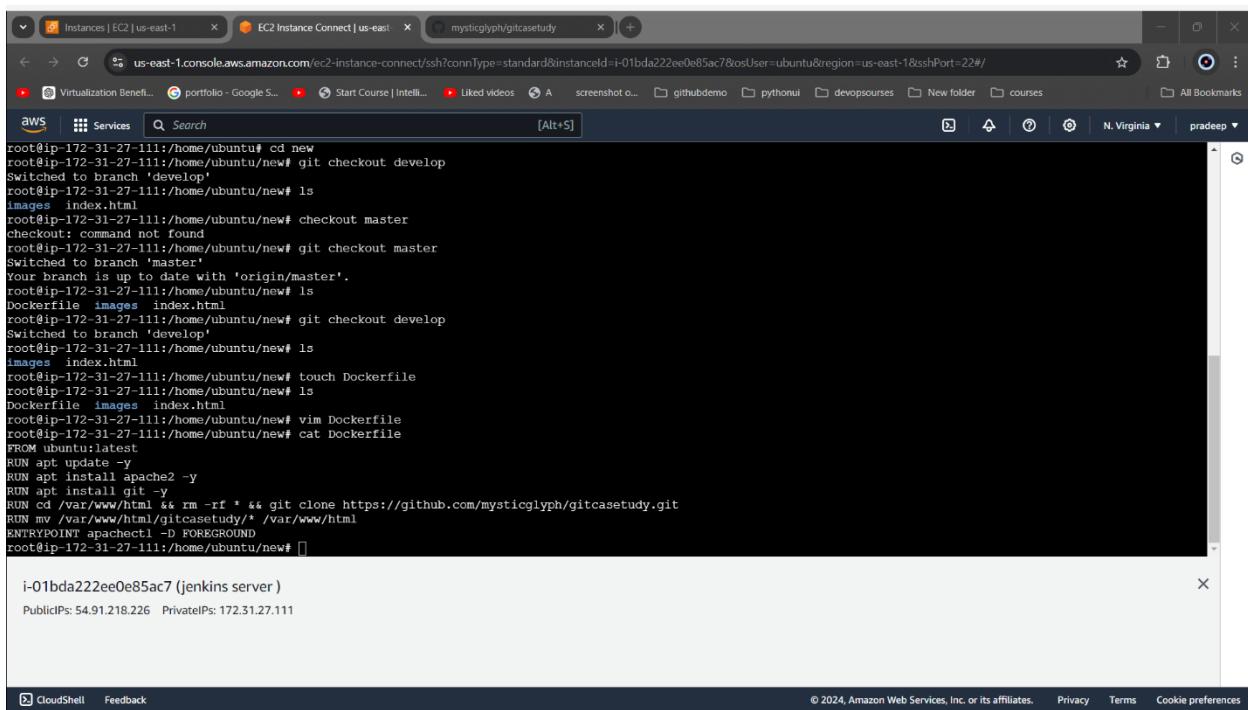
```
FROM ubuntu:latest
RUN apt update -y
RUN apt install apache2 -y
RUN apt install git -y
RUN cd /var/www/html && rm -rf * && git clone https://github.com/mysticglyph/gitcasestudy.git
RUN mv /var/www/html/gitcasestudy/* /var/www/html
ENTRYPOINT apachectl -D FOREGROUND
```

:wq! []

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 54.91.218.226 PrivateIPs: 172.31.27.111

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

Save the docker file.



A screenshot of the AWS CloudShell interface, similar to the previous one but with more content in the terminal window. The terminal shows a root shell on an Ubuntu 22.04 LTS system. The user has run several commands to switch branches (develop and master), touch files (Dockerfile, index.html), and edit the Dockerfile using vim. The Dockerfile now includes a FROM instruction for ubuntu:latest, RUN apt update and apt install commands for apache2 and git, and the previously seen git cloning and directory setup. The status bar at the bottom remains the same, showing the instance ID, Jenkins server status, and network information. The bottom navigation bar is also identical.

```
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# git checkout develop
Switched to branch 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# checkout master
checkout: command not found
root@ip-172-31-27-111:/home/ubuntu/new# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git checkout develop
Switched to branch 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# ls
images index.html
root@ip-172-31-27-111:/home/ubuntu/new# touch Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# vim Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# cat Dockerfile
FROM ubuntu:latest
RUN apt update -y
RUN apt install apache2 -y
RUN apt install git -y
RUN cd /var/www/html && rm -rf * && git clone https://github.com/mysticglyph/gitcasestudy.git
RUN mv /var/www/html/gitcasestudy/* /var/www/html
ENTRYPOINT apachectl -D FOREGROUND
root@ip-172-31-27-111:/home/ubuntu/new#
```

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 54.91.218.226 PrivateIPs: 172.31.27.111

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

Add the file to the staging area, commit the changes, and then push the file to the develop branch.

Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# vim Dockerfile
root@ip-172-31-27-111:/home/ubuntu/new# cat Dockerfile
FROM ubuntu:latest
RUN apt update -y
RUN apt install apache2 -y
RUN apt install git -y
RUN cd /var/www/html && rm -rf * && git clone https://github.com/mysticglyph/gitcasestudy.git
RUN mv /var/www/html/gitcasestudy/* /var/www/html
ENTRYPOINT apachectl -D FOREGROUND
root@ip-172-31-27-111:/home/ubuntu/new# git add .
root@ip-172-31-27-111:/home/ubuntu/new# git commit -m "dockerfile in develop branch"
[develop 2087746] dockerfile in develop branch
Committer: root <root@ip-172-31-27-111.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-172-31-27-111:/home/ubuntu/new#

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 54.91.218.226 PrivateIPs: 172.31.27.111

Added to staging area and changes are committed.

Pushing to remote repo:

→ Git push

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 Aug 24 05:19:41 2024 from 18.206.107.27
ubuntu@ip-172-31-27-111:~\$ ls
new
ubuntu@ip-172-31-27-111:~\$ sudo su
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git checkout develop
Already on 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git push
fatal: The current branch develop has no upstream branch.
To push the current branch and set the remote as upstream, use
git push --set-upstream origin develop

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
root@ip-172-31-27-111:/home/ubuntu/new# git push origin develop
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com': []

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 54.91.218.226 PrivateIPs: 172.31.27.111

A screenshot of a web browser window titled "Instances | EC2 | us-east-1" and "EC2 Instance Connect | us-east-1". The address bar shows "mysticglyph/gitcasestudy". The terminal window displays the following command-line session:

```
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# cd new
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git checkout develop
Already on 'develop'
root@ip-172-31-27-111:/home/ubuntu/new# ls
Dockerfile images index.html
root@ip-172-31-27-111:/home/ubuntu/new# git push
fatal: The current branch develop has no upstream branch.
To push the current branch and set the remote as upstream, use

  git push --set-upstream origin develop

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

root@ip-172-31-27-111:/home/ubuntu/new# git push origin develop
Username for 'https://github.com': mysticglyph
Password for 'https://mysticglyph@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (3/3), 493 bytes | 493.00 KiB/s, done.
total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mysticglyph/gitcasestudy.git
  453a7bd..2087746 develop -> develop
root@ip-172-31-27-111:/home/ubuntu/new#
```

Below the terminal, the message "i-01bda222ee0e85ac7 (jenkins server)" and "Public IPs: 54.91.218.226 Private IPs: 172.31.27.111" is displayed.

A screenshot of a GitHub repository page for "gitcasestudy" at the "develop" branch. The URL is "github.com/mysticglyph/gitcasestudy/tree/develop". The page shows the following details:

- Code**: develop branch (selected), 2 Branches, 0 Tags
- Actions**: This branch is 1 commit ahead of master.
- Contributors**: Compare & pull request
- Commits**:
 - root dockerfile in develop branch (2087746, 4 minutes ago)
 - images Initial commit with copied content (2 weeks ago)
 - Dockerfile dockerfile in develop branch (4 minutes ago)
 - index.html Initial commit with copied content (2 weeks ago)
- README**: A file icon is shown.
- About**: No description, website, or topics provided.
- Activity**: 0 stars, 1 watching, 0 forks.
- Releases**: No releases published. Create a new release.
- Packages**: No packages published. Publish your first package.
- Languages**: A progress bar indicates 0% completion.

Push is successful.

We need to create a Docker repository for building the image and publishing it. We'll use Docker Hub for publishing.

So we will create docker repository:

The screenshot shows a browser window with multiple tabs open. The active tab is 'Repositories | Docker Hub'. The URL is 'hub.docker.com/repositories/pradeepacharya0'. The page displays a list of Docker repositories owned by 'pradeepacharya0'. The repositories listed are:

- pradeepacharya0 / dockerassignment3
- pradeepacharya0 / maven
- pradeepacharya0 / newfile
- pradeepacharya0 / jenkins
- pradeepacharya0 / staticapp

Each repository entry includes a star icon (0), a pull request icon (2 or 1), a public status icon, and a 'Scout inactive' status indicator. A 'Create repository' button is located at the top right of the list area.

Creating directory:

The screenshot shows a browser window with multiple tabs open. The active tab is 'Repositories | Docker Hub'. The URL is 'hub.docker.com/repository/create?namespace=pradeepacharya0'. The page displays a form for creating a new repository. The 'Namespace' dropdown is set to 'pradeepacharya0'. The 'Repository Name' field contains 'jenkins_casestudy'. The 'Short description' field contains 'directory for jenkins case study'. The 'Visibility' section shows 'Public' is selected, with a note that it appears in Docker Hub search results. The 'Pushing images' section provides instructions for pushing images to the repository using the CLI:

```
docker tag local-image:tagname new-repo:tagname  
docker push new-repo:tagname
```

A note says to replace 'tagname' with the desired image repository tag.

At the bottom, there are 'Cancel' and 'Create' buttons.

Directory is created.

The screenshot shows a Docker Hub repository page. The repository name is 'pradeepacharya0/jenkilns_casestudy'. It has a single tag named 'General'. The Docker commands section contains the command 'docker push pradeepacharya0/jenkilns_casestudy:tagname'. There is a 'Public View' button. The 'Tags' section is labeled 'INCOMPLETE' and says 'This repository is empty. Push some images to it to see them appear here.' The 'Automated Builds' section says 'Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.' It also mentions 'Available with Pro, Team and Business subscriptions' and a 'Read more about automated builds' link. An 'Upgrade' button is present. At the bottom, there is a 'Repository overview' section labeled 'INCOMPLETE'.

Now we need to configure our master job. The main objective of the master node is to automatically trigger a code build whenever a commit is made to the master or develop branch. If a commit is made to the master branch, the build should publish the website on port 82.

The screenshot shows the Jenkins dashboard. On the left, there are navigation links: '+ New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Under 'Build Queue', it says 'No builds in the queue.' Under 'Build Executor Status', it shows '1 Idle' and '2 Idle'. In the center, there is a table for the 'mater-job' job. The columns are 'S' (Status), 'W' (Work), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The 'Name' column shows 'mater-job' with a sun icon. The 'Last Success' and 'Last Failure' columns both show 'N/A'. The 'Last Duration' column shows 'N/A'. Below the table, there is a 'Icon:' dropdown with options 'S', 'M', and 'L'. At the bottom right, there are links for 'REST API' and 'Jenkins 2.462.1'.

The screenshot shows the Jenkins configuration interface for a job named "mater-job". The "General" tab is selected. The "Enabled" checkbox is checked. The "Description" field is empty. Under "Advanced", there are several optional checkboxes: "Discard old builds", "GitHub project", "This project is parameterized", "Throttle builds", and "Execute concurrent builds if necessary". The "Save" and "Apply" buttons are at the bottom.

General:

Git project:

The screenshot shows the Jenkins configuration interface for a job named "mater-job". The "General" tab is selected. The "GitHub project" checkbox is checked. The "Project url" field is empty. Under "Advanced", there are three optional checkboxes: "This project is parameterized", "Throttle builds", and "Execute concurrent builds if necessary". The "Save" and "Apply" buttons are at the bottom.

The screenshot shows the Jenkins job configuration interface for a job named 'mater-job'. The 'General' section is selected. Under 'Source Code Management', the 'GitHub project' option is checked, and the 'Project url' field contains the value 'https://github.com/mysticglyph/gitcasestudy.git'. There are several other optional checkboxes like 'Discard old builds', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. At the bottom, there are 'Save' and 'Apply' buttons.

Please provide the GitHub URL.

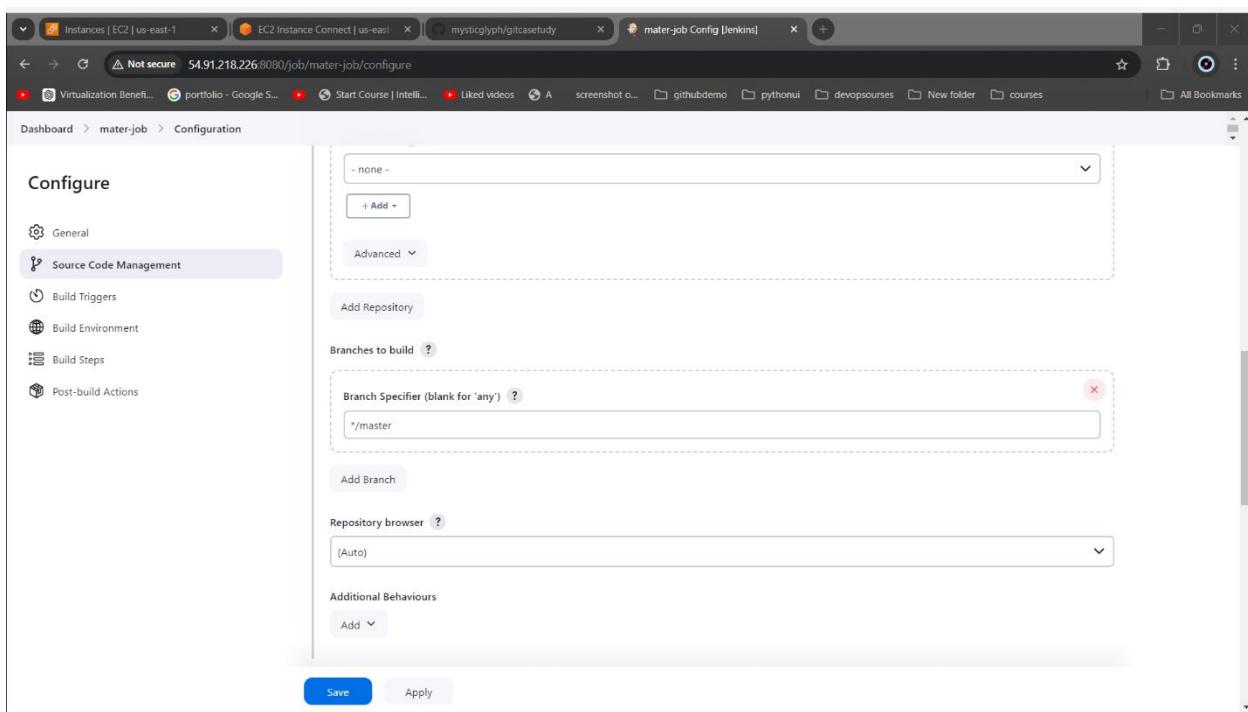
Source Code Management:

Repository URL:

The screenshot shows the Jenkins job configuration interface for a job named 'mater-job'. The 'General' section is selected. Under 'Source Code Management', the 'Git' option is selected. In the 'Repositories' section, the 'Repository URL' field contains 'https://github.com/mysticglyph/gitcasestudy.git'. The 'Credentials' dropdown is set to 'none'. There is an 'Add Repository' button and a 'Branches to build' section at the bottom. At the bottom, there are 'Save' and 'Apply' buttons.

We don't need to provide credentials because our GitHub repository is public.

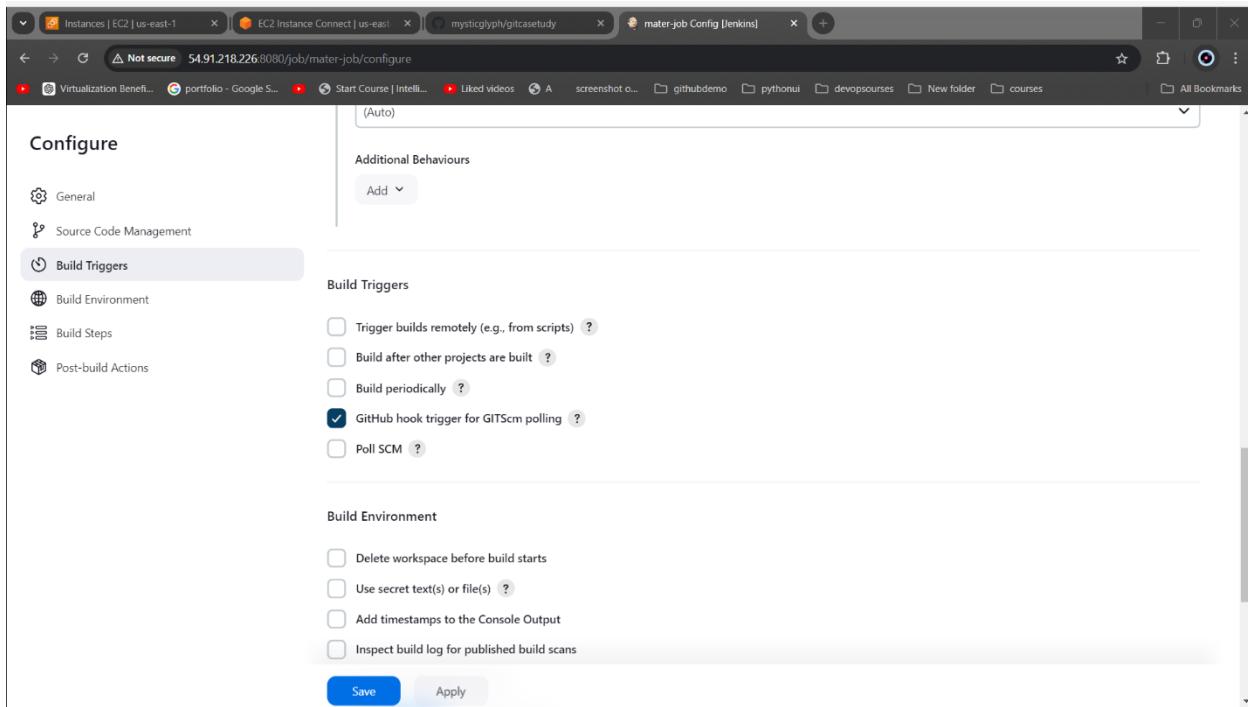
Branches to build is master branch:



The screenshot shows the Jenkins job configuration page for a job named "mater-job". The left sidebar has "Source Code Management" selected. In the main area, under "Branches to build", the "Branch Specifier" field contains the value "*/master". There is also an "Add Branch" button and a "Repository browser" dropdown set to "(Auto)".

Build Triggers:

GitHub hook trigger for GITScm polling:



The screenshot shows the Jenkins job configuration page for a job named "mater-job". The left sidebar has "Build Triggers" selected. In the main area, under "Build Triggers", the "GitHub hook trigger for GITScm polling" checkbox is checked. Other options like "Trigger builds remotely" and "Poll SCM" are unchecked.

Build Steps:

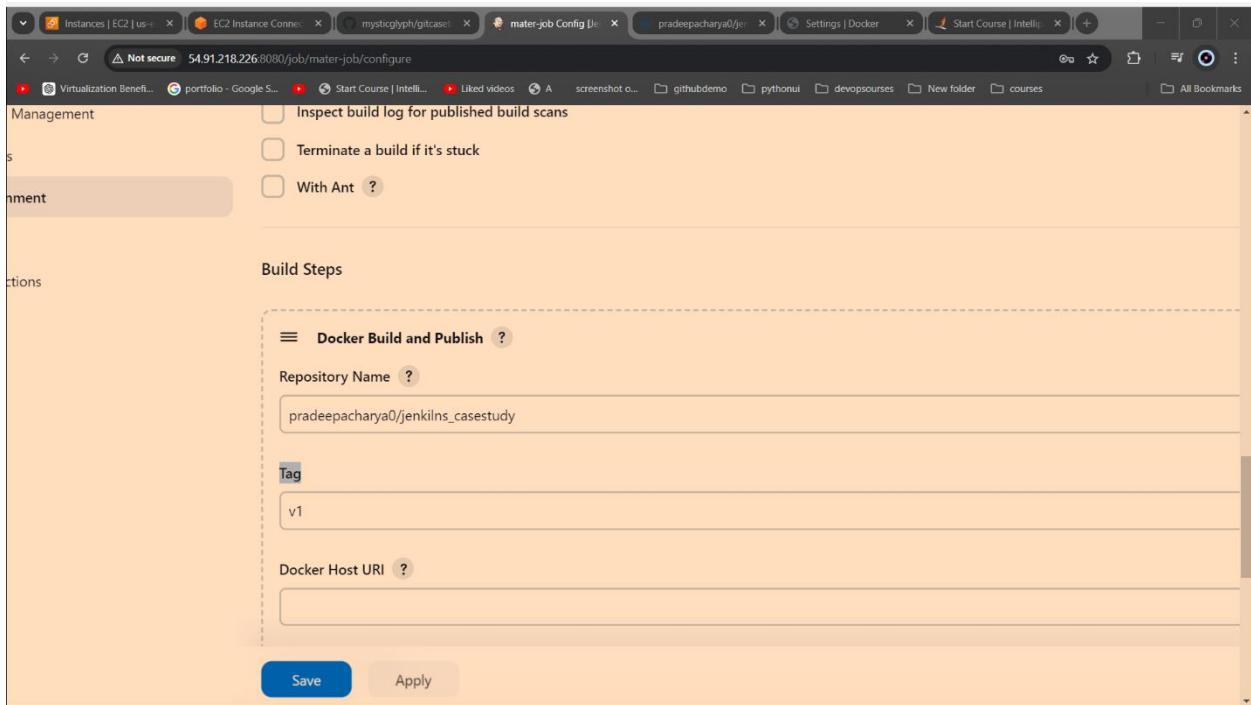
The screenshot shows the Jenkins configuration interface for a job named 'mater-job'. The 'Build Environment' section is open, displaying a list of build steps. The 'Docker Build and Publish' option is highlighted with a light blue background. Other options listed include 'Execute Docker command', 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. A 'Filter' input field is present at the top of the list.

We need to select Docker Build and Publish.

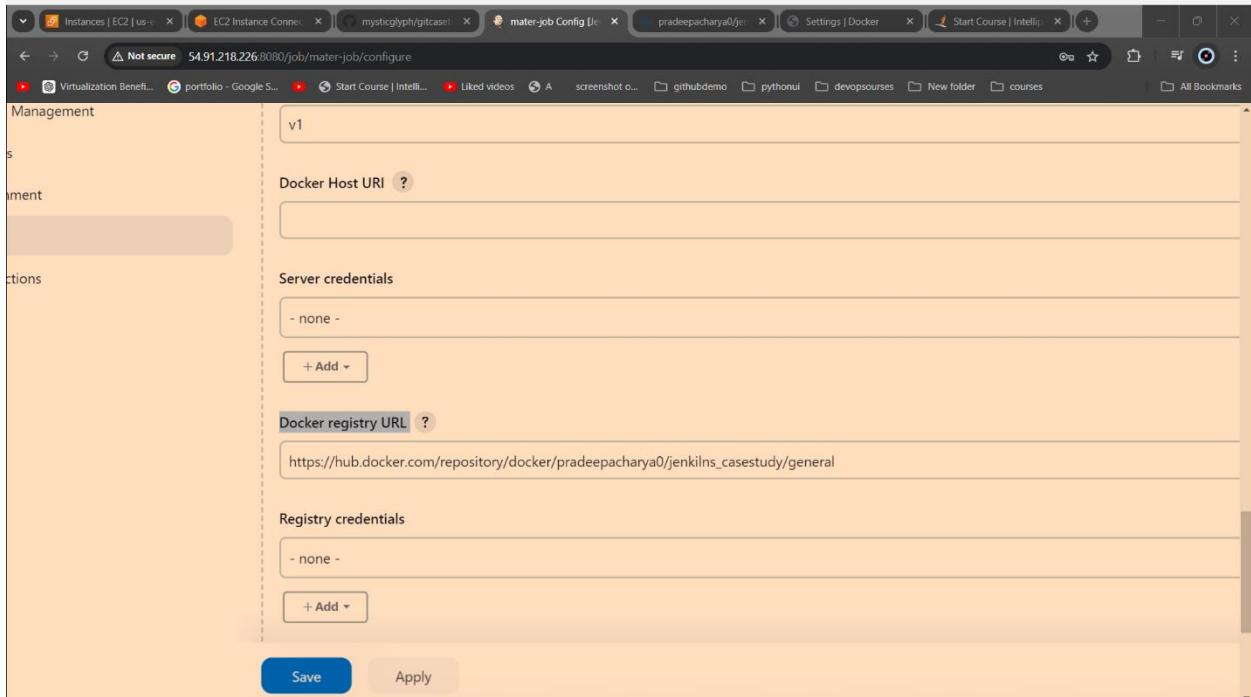
Repository Name:

The screenshot shows the Jenkins configuration interface for a job named 'mater-job'. The 'Build Steps' section is open, showing a single step named 'Docker Build and Publish'. Under this step, the 'Repository Name' field is populated with the value 'pradeepacharya0/jenkilns_casestudy'. The 'Tag' field contains 'v1'. The 'Docker Host URI' field is empty. To the left of the build steps, there is a sidebar with various management options like 'Use secret text(s) or file(s)', 'Add timestamps to the Console Output', 'Inspect build log for published build scans', 'Terminate a build if it's stuck', and 'With Ant'.

Tag:

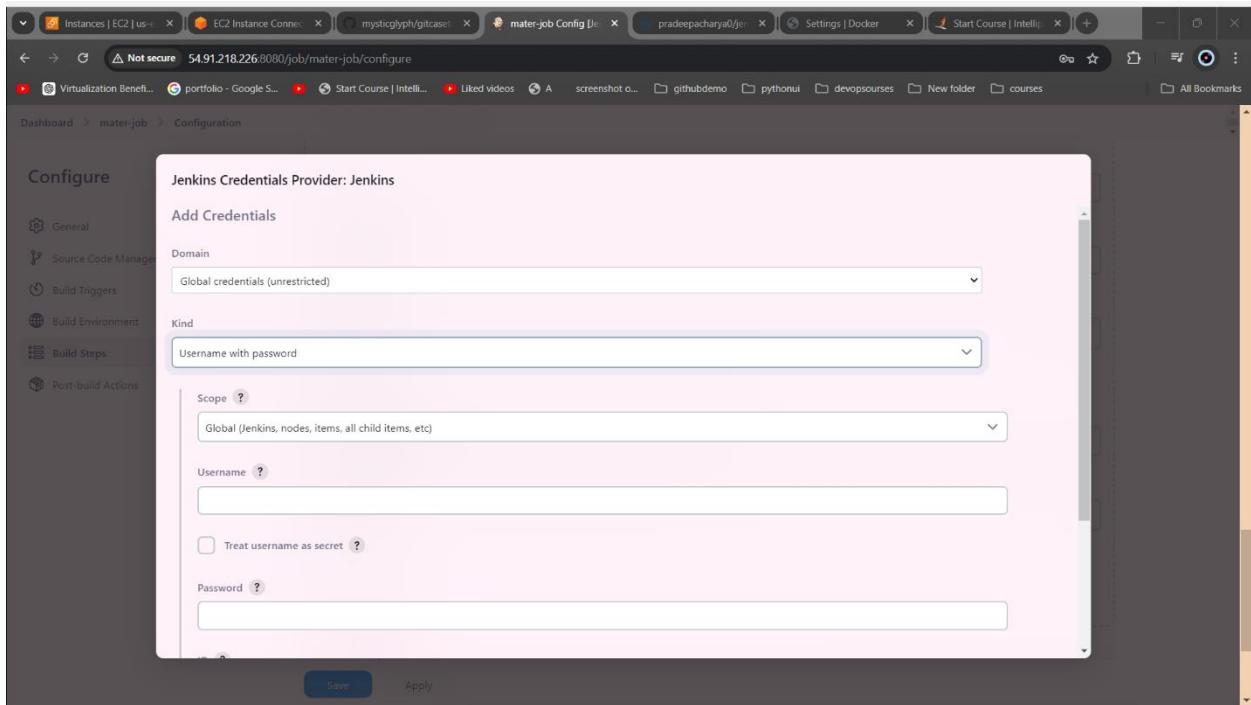
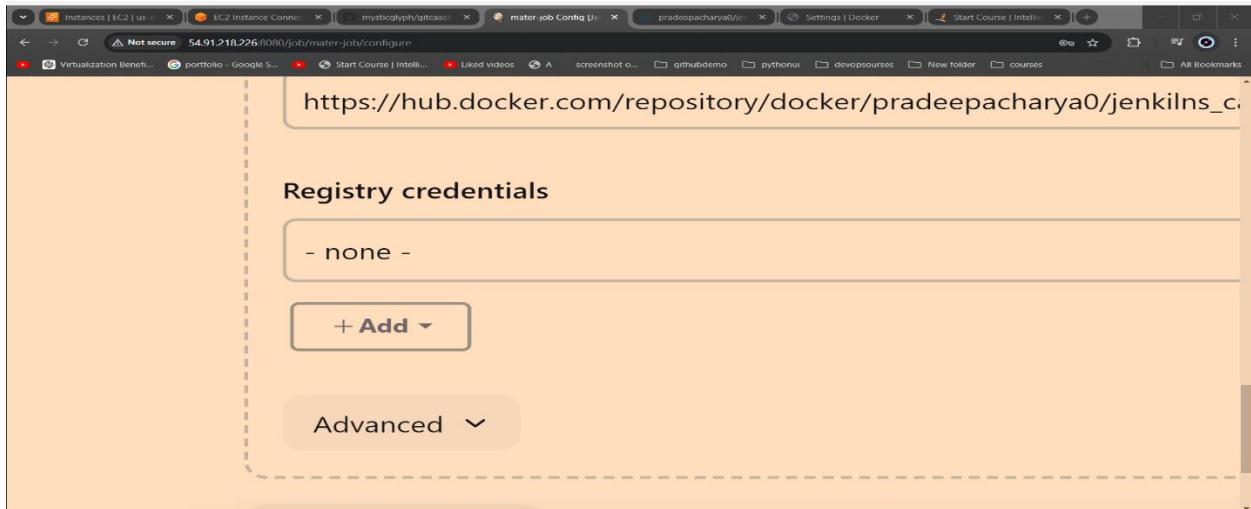


Docker registry URL:



Registry credentials:

In order to add credentials, proceed with the 'Add' option.



Kind:

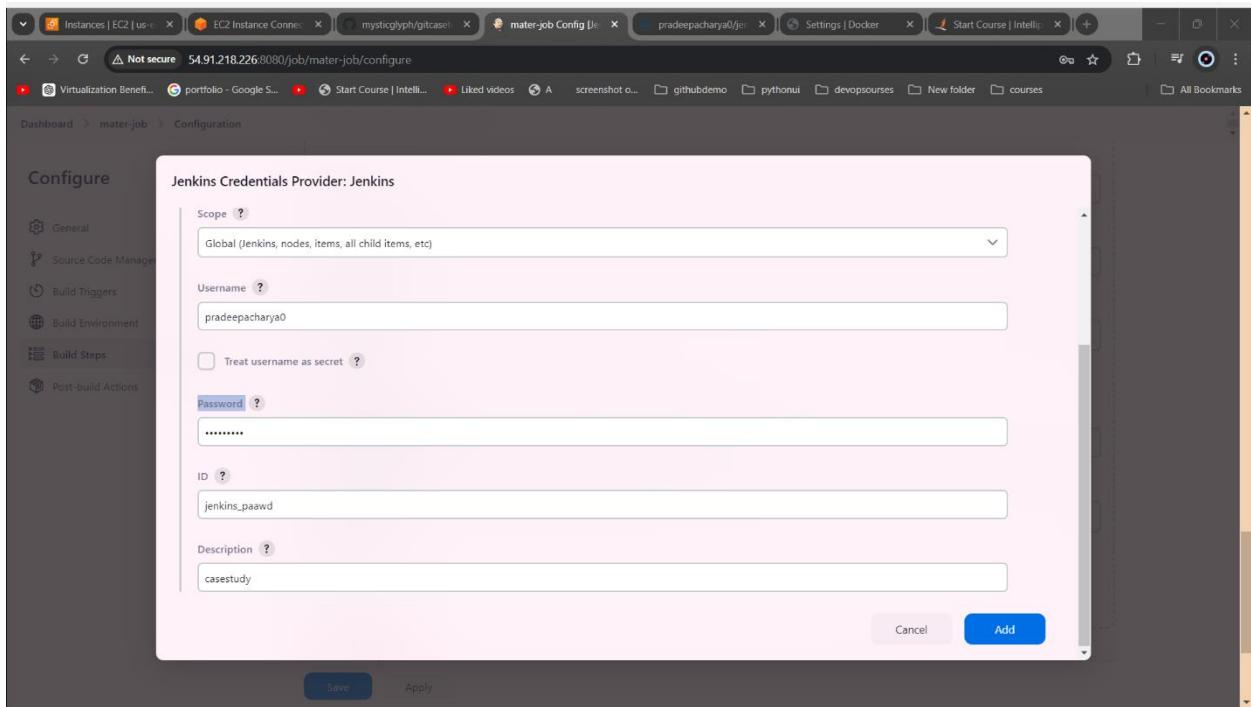
Username and password:

The screenshot shows the Jenkins 'Add Credentials' dialog. On the left, there's a sidebar with icons for General, Source Code Manage, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main area has a title 'Add Credentials'. Under 'Kind', 'Username with password' is selected. Below it, 'Scope' is set to 'Global (Jenkins, nodes, items, all child items, etc)'. There are fields for 'Username' and 'Password', with a checkbox 'Treat username as secret'.

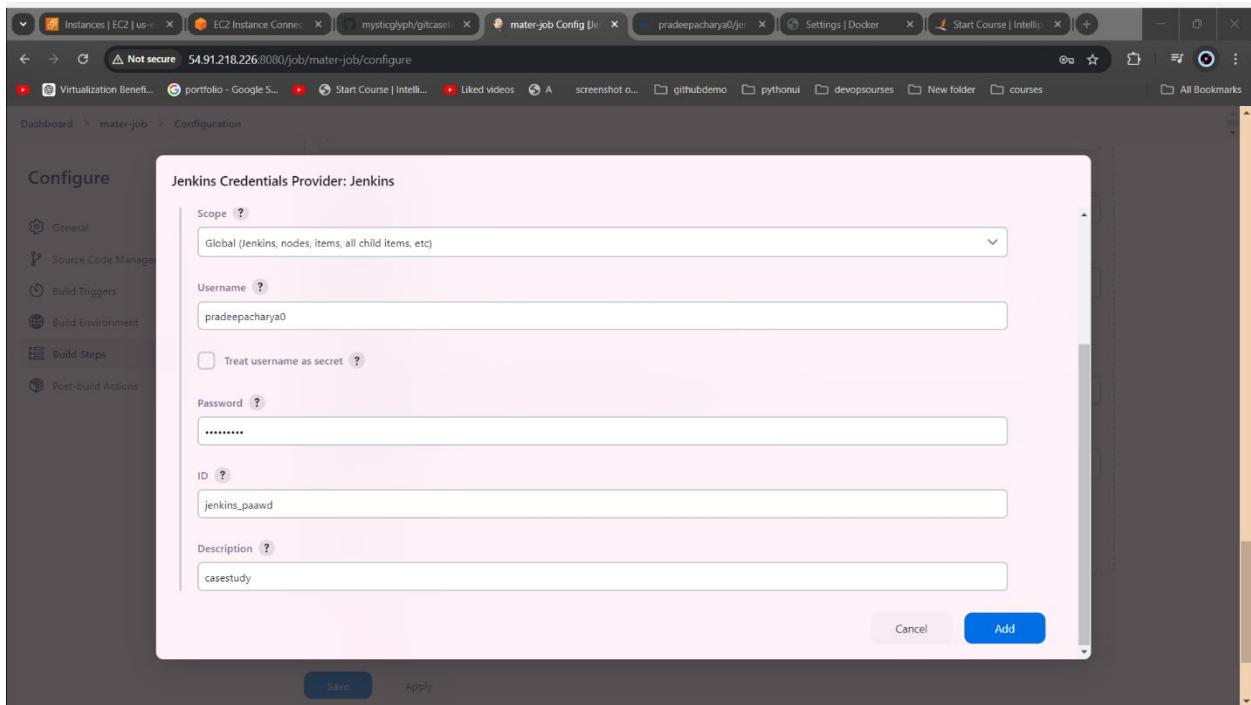
Username:

The screenshot shows the Jenkins 'Configure' dialog for a Jenkins Credentials Provider. The sidebar on the left includes 'General', 'Source Code Manage', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The main dialog is titled 'Jenkins Credentials Provider: Jenkins'. It shows a 'Scope' dropdown set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains 'pradeepacharya0'. Other fields include 'Password' (with a redacted value), 'ID' (set to 'jenkins_paawd'), and 'Description' (set to 'casestudy'). At the bottom are 'Save', 'Apply', and 'Cancel' buttons, with a prominent blue 'Add' button.

Password:



Description and id:



Registry credentials:

Choose the credentials that have been created.

The screenshot shows the Jenkins job configuration page for a job named "mater-job". The "Configure" section is open, specifically the "Build Steps" section. Under "Docker registry URL", the value is set to `https://hub.docker.com/repository/docker/pradeepacharya0/jenkilns_casestudy/general`. Under "Registry credentials", there is a dropdown menu containing the entry `pradeepacharya0/******** (casestudy)`. A button labeled "+ Add" is available to add more credentials. Below these fields is an "Advanced" dropdown. At the bottom of the configuration section is a "Save" button.

This screenshot shows the Jenkins job configuration page for the same "mater-job" job. The "Configure" section is open, and the "Build Steps" tab is selected. The "Docker registry URL" field contains the value `https://hub.docker.com/repository/docker/pradeepacharya0/jenkilns_casestudy/general`. The "Registry credentials" dropdown also contains the entry `pradeepacharya0/******** (casestudy)`. Below these fields is an "Advanced" dropdown. At the bottom of the configuration section are "Save" and "Apply" buttons. The footer of the page indicates "REST API" and "Jenkins 2.462.1".

And save it.

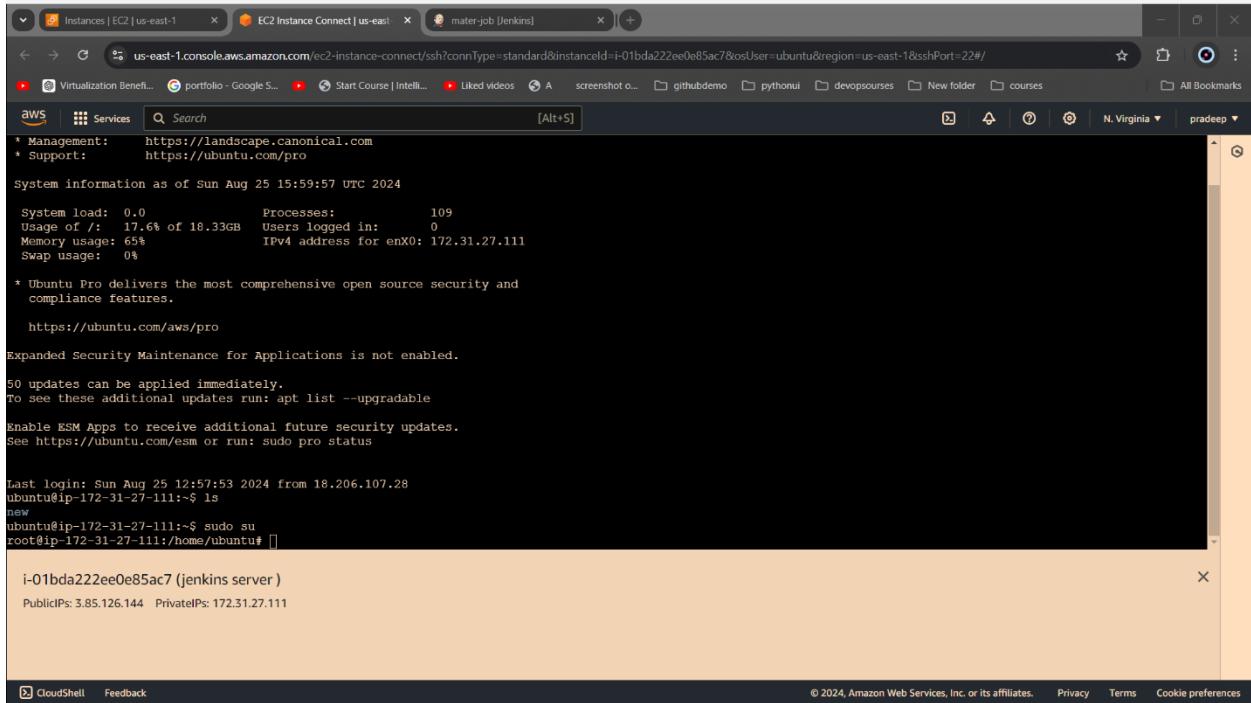
The screenshot shows the Jenkins interface for the 'mater-job' project. The left sidebar contains links for Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, GitHub, and Rename. The main area is titled 'mater-job' and includes a 'Permalinks' section with a 'Search (CTRL+K)' bar. Below it is a 'Build History' section with a 'trend' dropdown set to 'trend'. A message 'No builds' is displayed. At the bottom are links for 'Atom feed for all' and 'Atom feed for failures'. The footer shows 'REST API' and 'Jenkins 2.462.1'.

Job have been configured.

In order to execute the job, we need to choose the 'Build Now' option in the dashboard.

This screenshot is identical to the one above, showing the Jenkins dashboard for the 'mater-job' project. The 'Build Now' link in the sidebar is highlighted with a yellow box, indicating it has been selected. The rest of the interface and content are the same as the first screenshot.

There is one more thing to do: we need to add the Jenkins user to the Docker group because the Jenkins user doesn't have the privileges to execute Docker commands. By adding the Jenkins user to the Docker group, Jenkins will be able to run Docker commands.



A screenshot of an AWS CloudShell terminal window. The terminal shows system information for an Ubuntu 22.04 LTS instance. It includes details like system load (0.0), memory usage (65%), and swap usage (0%). It also lists processes (109) and users logged in (0). A note about Ubuntu Pro security features is present. The terminal then transitions to a root shell, showing the user's public and private IP addresses. At the bottom, there are links for CloudShell, Feedback, and cookie preferences.

```
System information as of Sun Aug 25 15:59:57 UTC 2024

System load: 0.0 Processes: 109
Usage of /: 17.6% of 18.33GB Users logged in: 0
Memory usage: 65% IPv4 address for enX0: 172.31.27.111
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.

50 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

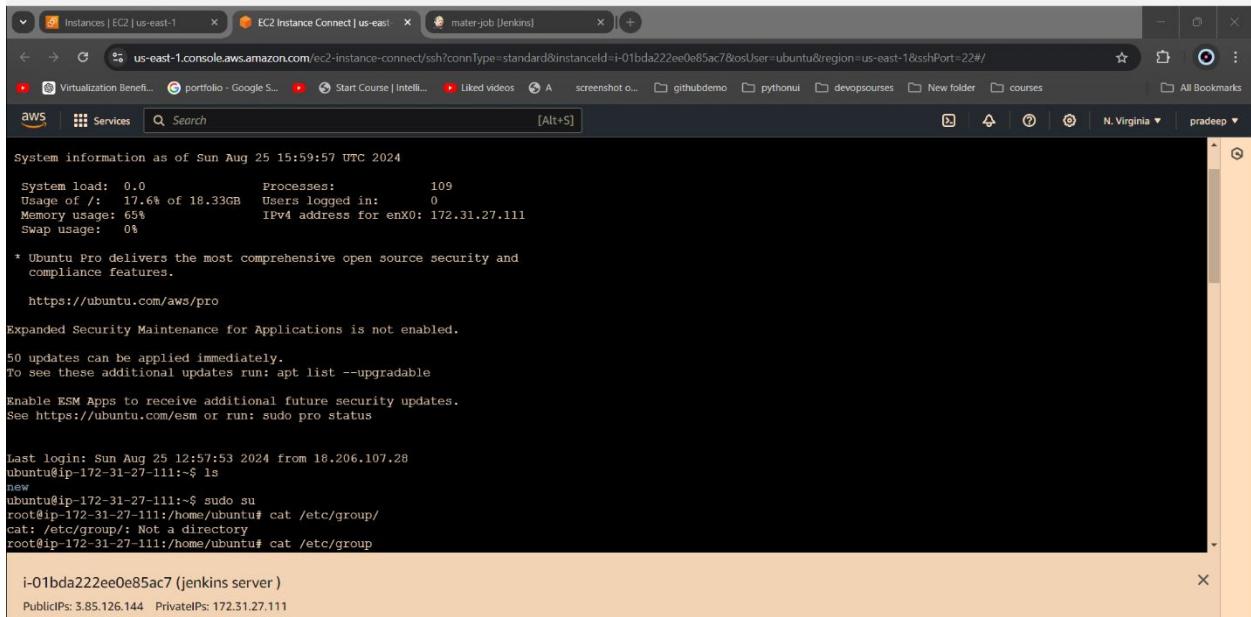
Last login: Sun Aug 25 12:57:53 2024 from 18.206.107.28
ubuntu@ip-172-31-27-111:~$ ls
new
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# []

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 3.85.126.144 PrivateIPs: 172.31.27.111

CloudShell Feedback
```

In order to check how many group are there we can run the command:

→ **Cat /etc/group**



A screenshot of an AWS CloudShell terminal window, identical to the previous one but showing the result of the 'cat /etc/group' command. The output lists several groups, including 'root', 'dialout', 'audio', 'cdrom', 'sudo', and 'docker'. The 'docker' group is specifically mentioned as not being a directory. The terminal shows the user's public and private IP addresses at the bottom.

```
System information as of Sun Aug 25 15:59:57 UTC 2024

System load: 0.0 Processes: 109
Usage of /: 17.6% of 18.33GB Users logged in: 0
Memory usage: 65% IPv4 address for enX0: 172.31.27.111
Swap usage: 0%

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

  https://ubuntu.com/aws/pro

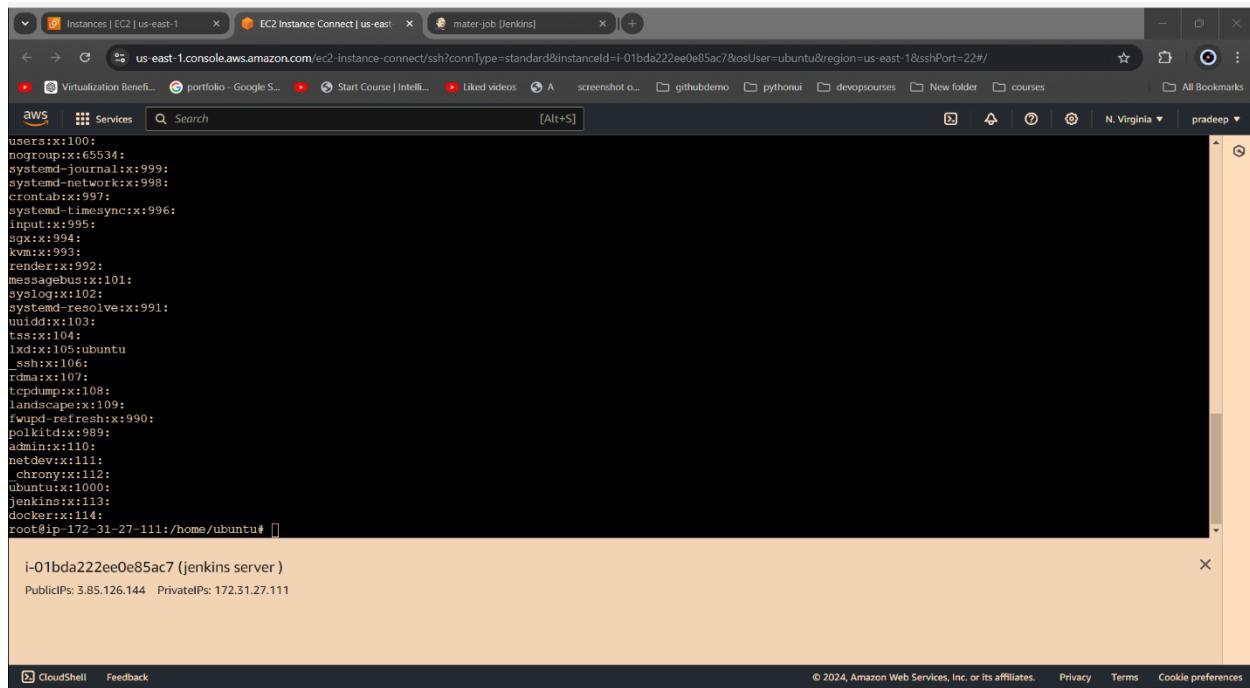
Expanded Security Maintenance for Applications is not enabled.

50 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: Sun Aug 25 12:57:53 2024 from 18.206.107.28
ubuntu@ip-172-31-27-111:~$ ls
new
ubuntu@ip-172-31-27-111:~$ sudo su
root@ip-172-31-27-111:/home/ubuntu# cat /etc/group/
cat: /etc/group/: Not a directory
root@ip-172-31-27-111:/home/ubuntu# cat /etc/group

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 3.85.126.144 PrivateIPs: 172.31.27.111
```



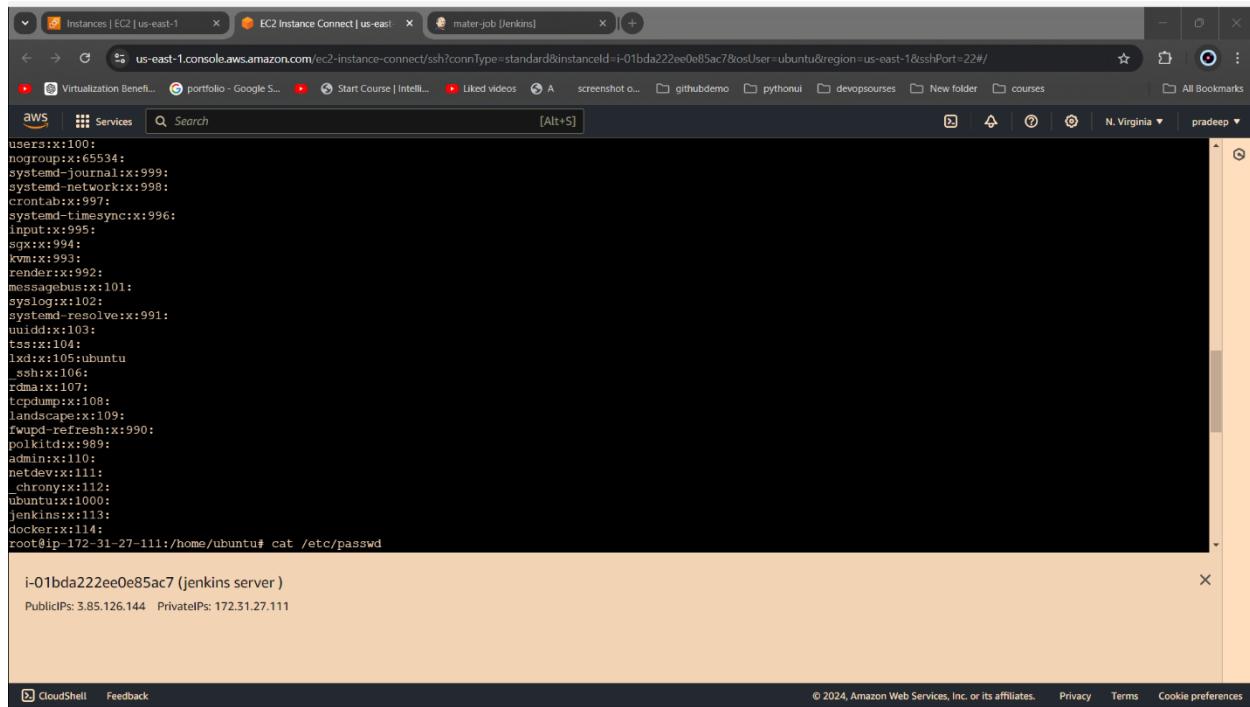
```
aws Services Search [Alt+S]
users:x:100:
nogroup:x:65534:
systemd-journal:x:999:
systemd-network:x:998:
crontab:x:997:
systemd-timesync:x:996:
input:x:995:
sgx:x:994:
kvm:x:993:
render:x:992:
messagebus:x:101:
syslog:x:102:
systemd-resolve:x:991:
uuidd:x:103:
tss:x:104:
lxde:x:105:ubuntu
_ssh:x:106:
rdmax:x:107:
tcpdump:x:108:
landscape:x:109:
fwupd-refresh:x:990:
polkitd:x:989:
admin:x:110:
netdev:x:111:
_chrony:x:112:
ubuntu:x:1000:
jenkins:x:113:
docker:x:114:
root@ip-172-31-27-111:/home/ubuntu#
```

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 3.85.126.144 PrivateIPs: 172.31.27.111

Now, it is evident that there is a group called 'docker'.

Identifying whether the Jenkins user is present or not.

→ Cat /etc/passwd



```
aws Services Search [Alt+S]
users:x:100:
nogroup:x:65534:
systemd-journal:x:999:
systemd-network:x:998:
crontab:x:997:
systemd-timesync:x:996:
input:x:995:
sgx:x:994:
kvm:x:993:
render:x:992:
messagebus:x:101:
syslog:x:102:
systemd-resolve:x:991:
uuidd:x:103:
tss:x:104:
lxde:x:105:ubuntu
_ssh:x:106:
rdmax:x:107:
tcpdump:x:108:
landscape:x:109:
fwupd-refresh:x:990:
polkitd:x:989:
admin:x:110:
netdev:x:111:
_chrony:x:112:
ubuntu:x:1000:
jenkins:x:113:
docker:x:114:
root@ip-172-31-27-111:/home/ubuntu# cat /etc/passwd
```

i-01bda222ee0e85ac7 (jenkins server)
PublicIPs: 3.85.126.144 PrivateIPs: 172.31.27.111

The screenshot shows a web browser interface with two tabs: "Instances | EC2 us-east-1" and "EC2 Instance Connect | us-east-1". The main content area displays a terminal window with the following text:

```
news:ix:19:news:/var/spool/news:/usr/sbin/nologin
uucp:ix:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:ix:13:13:proxy:/bin:/usr/sbin/nologin
www-data:ix:33:33:www-data:/var/www:/usr/sbin/nologin
backup:ix:34:34:backup:/var/backups:/usr/sbin/nologin
list:ix:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:ix:39:39:ircd:/run/ircd:/usr/sbin/nologin
apt:ix:42:65534:/noneexistent:/usr/sbin/nologin
nobody:ix:65534:65534:nobody:/noneexistent:/usr/sbin/nologin
systemd-network:ix:998:998:system Network Management:/:/usr/sbin/nologin
systemd-timesync:ix:996:996:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:ix:100:65534:DHCP Client Daemon,,:/usr/lib/dhcpcd:/bin/false
messagebus:ix:101:101:/noneexistent:/usr/sbin/nologin
syslog:ix:102:102:/noneexistent:/usr/sbin/nologin
systemd-resolve:ix:991:991:systemd Resolver:/:/usr/sbin/nologin
uuid:ix:103:103:/run/uuid:/usr/sbin/nologin
tss:ix:104:104:TPM software stack,,,:/var/lib/tpm:/bin/false
sshd:ix:105:65534:/run/sshd:/usr/sbin/nologin
pollinate:ix:106:1:/var/cache/pollinate:/bin/false
tcpdump:ix:107:108:/noneexistent:/usr/sbin/nologin
landscape:ix:108:109:/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:ix:990:990:firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
polkitd:ix:989:989:User for polkitd:/:/usr/sbin/nologin
ec2-instance-connect:ix:109:65534:/noneexistent:/usr/sbin/nologin
_chrony:ix:110:112:chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
ubuntu:ix:1000:ubuntu:/home/ubuntu:/bin/bash
jenkins:ix:111:113:Jenkins,,,:/var/lib/jenkins:/bin/bash
dnsmasq:ix:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
root@ip-172-31-27-111:/home/ubuntu#
```

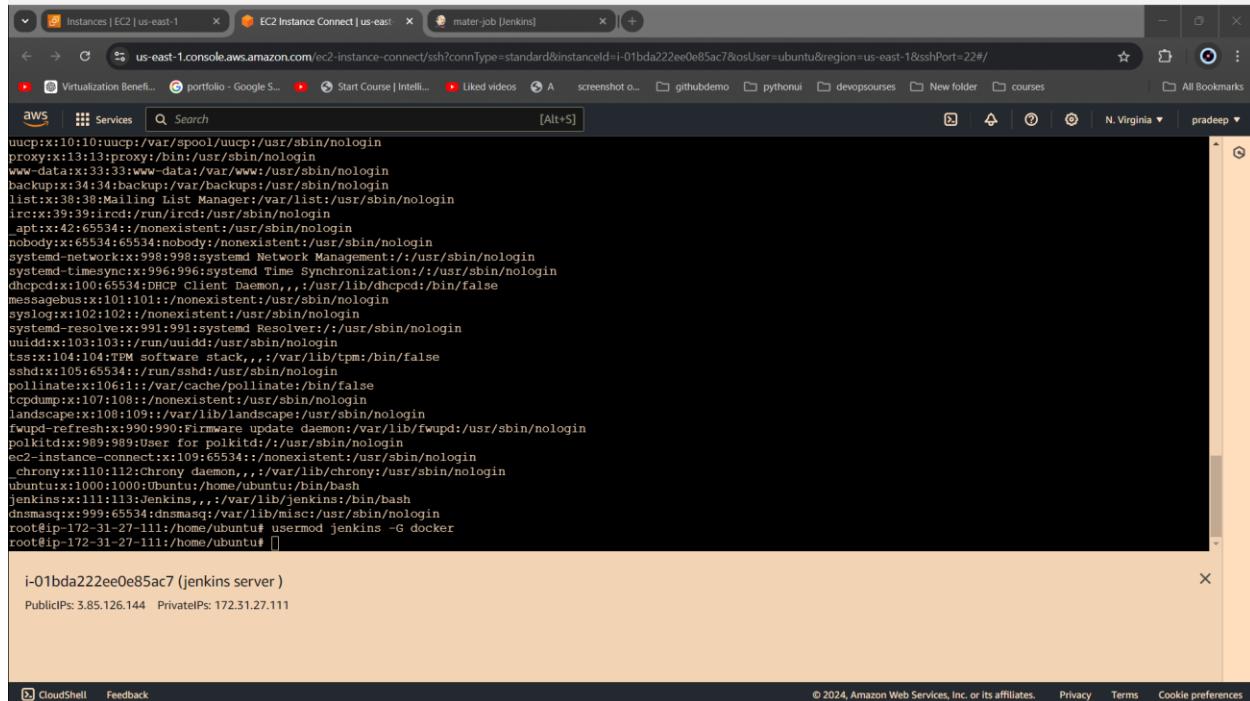
Below the terminal window, the text "i-01bda222ee0e85ac7 (jenkins server)" is displayed, followed by "Public IPs: 3.85.126.144 Private IPs: 172.31.27.111".

At the bottom of the browser window, there are links for "CloudShell", "Feedback", "© 2024, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

The Jenkins user is present on the server.

Now we need to add the Jenkins user to the Docker group. To do that, we need to run a command.

→ Sudo usermod jenkins -G docker

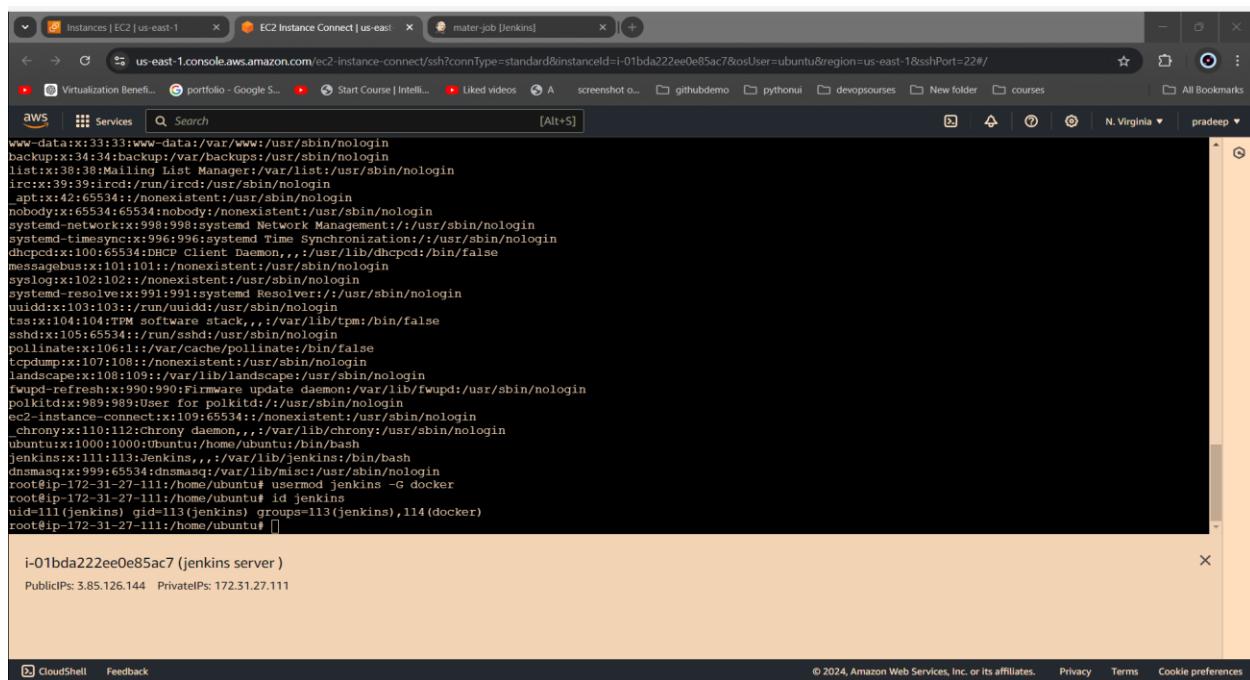


```
usermod: user 'jenkins' does not exist
root@ip-172-31-27-111:/home/ubuntu# usermod jenkins -G docker
root@ip-172-31-27-111:/home/ubuntu#
```

Jenkins is added to docker group.

In order to check that we need to run a command.

→ Id jenkins



```
root@ip-172-31-27-111:/home/ubuntu# id jenkins
uid=113(jenkins) gid=113(jenkins) groups=113(jenkins),114(docker)
root@ip-172-31-27-111:/home/ubuntu#
```

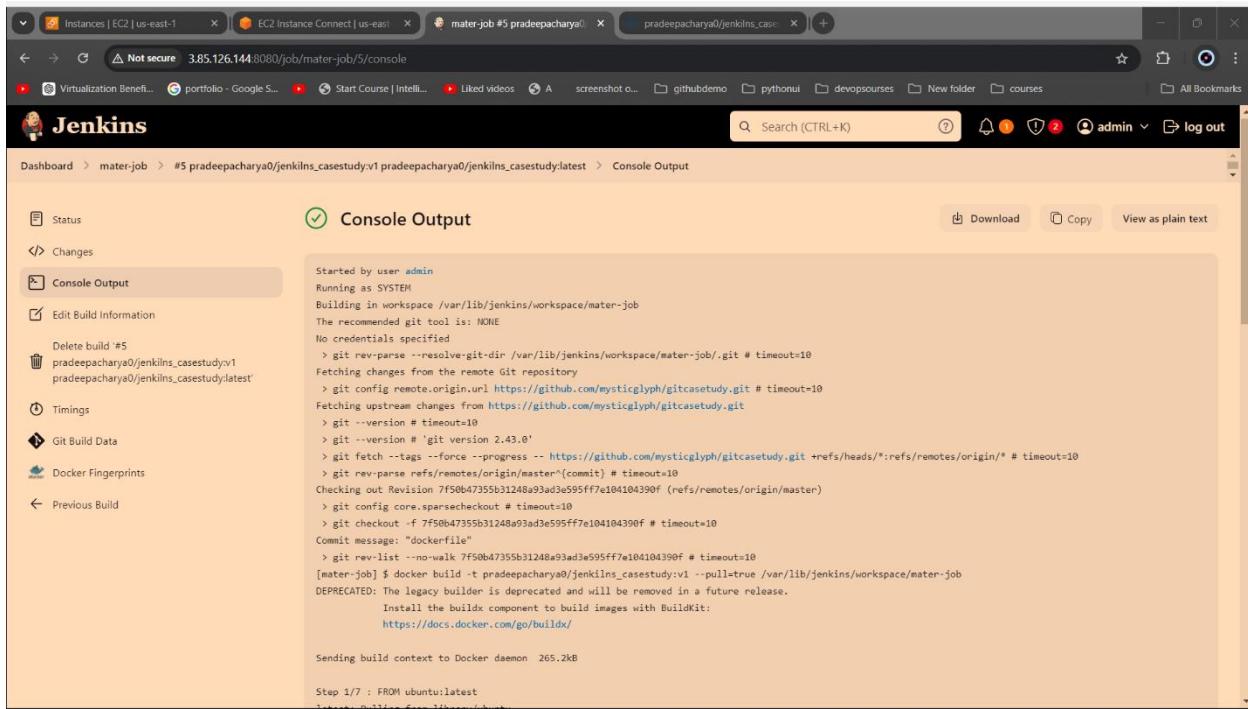
Now Jenkins can run Docker commands.

Now build the job and test if it executes according to the requirements.

The screenshot shows the Jenkins master-job dashboard. On the left, there's a sidebar with options like Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, and Rename. The main area is titled "mater-job" and contains a "Permalinks" section with a list of recent builds. Below that is a "Build History" section with a table showing the latest build. The table has columns for "Status", "Build #", "Duration", and "Last Result". The latest build is shown as successful (#5 pradeepacharya0/jenkilns_casestudy:v1 pradeepacharya0/jenkilns_casestudy:latest Aug 25, 2024, 4:37 PM).

Job is build successfully.

The screenshot shows the details of the latest build (#5 pradeepacharya0/jenkilns_casestudy:v1 pradeepacharya0/jenkilns_casestudy:latest). The page includes a sidebar with options like Status, Changes, Console Output, Edit Build Information, Delete build, Timings, Git Build Data, Docker Fingerprints, and Previous Build. The main content area displays the build summary, which includes the start time (Aug 25, 2024, 4:37:43 PM), duration (Started 1 min 4 sec ago Took 15 sec), and a list of tasks: Started by user admin, This run spent: 4 ms waiting, 15 sec build duration, 15 sec total from scheduled to completion. It also shows the git repository information: Revision: 7f50b47355b31248a93ad3e595ff7e104104390f, Repository: <https://github.com/mysticlyph/gitcasestudy.git>, and refs/remotes/origin/master. A note at the bottom says </> No changes.

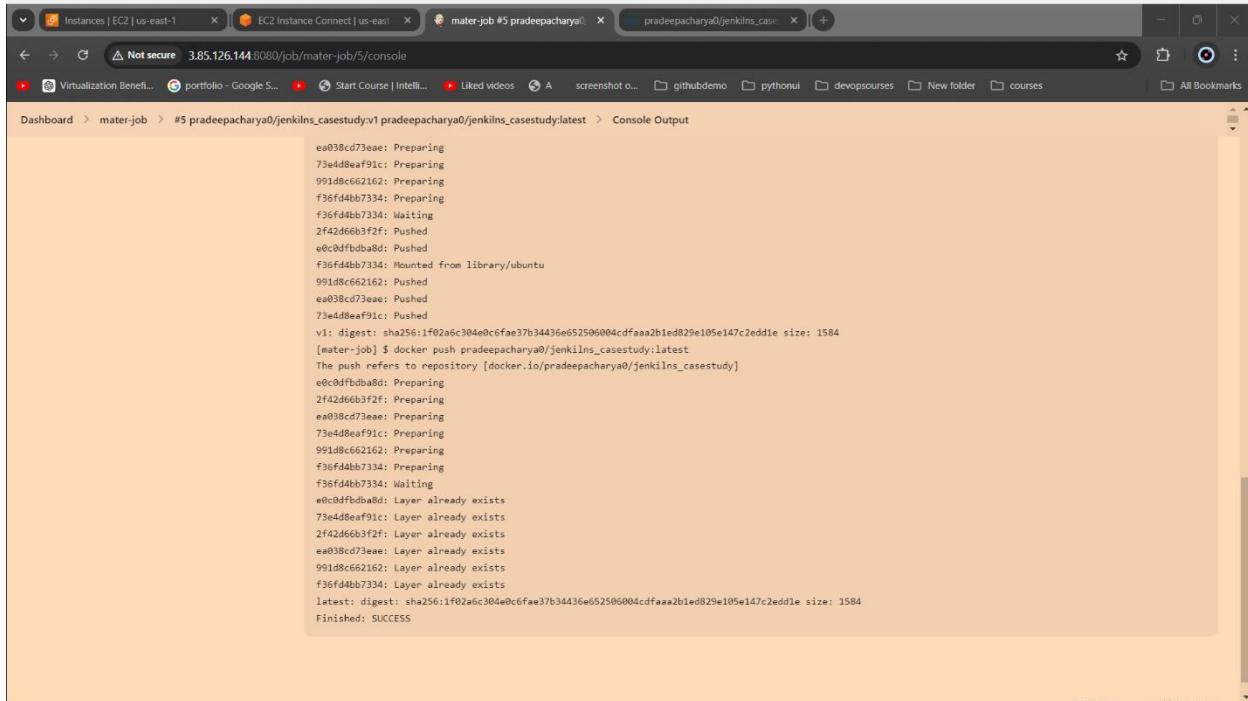


The screenshot shows the Jenkins console output for build #5 of the 'mater-job' project. The build was started by user 'admin' and is running as SYSTEM. It is building in workspace /var/lib/jenkins/workspace/mater-job. The recommended git tool is NONE, and no credentials are specified. The build is cloning from https://github.com/mysticglyph/gitcasestudy.git with a timeout of 10 seconds. It is fetching upstream changes from https://github.com/mysticglyph/gitcasestudy.git. The build version is 2.43.0'. It is pulling the 'master' branch and pushing to 'origin/master'. A commit message 'dockerfile' is present. The build is checking out Revision 7f50b4735b31248a93ad3e595ff7e104104390f. It is installing the buildx component to build images with BuildKit. The Docker context is being sent to the daemon at 265.2kB. The first step, 'FROM ubuntu:latest', is in progress.

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/mater-job
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/mater-job/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/mysticglyph/gitcasestudy.git # timeout=10
Fetching upstream changes from https://github.com/mysticglyph/gitcasestudy.git
> git --version # timeout=10
> git -v
> git fetch --tags --force --progress -- https://github.com/mysticglyph/gitcasestudy.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 7f50b4735b31248a93ad3e595ff7e104104390f (refs/remotes/origin/master)
Commit message: "dockerfile"
> git config core.sparsecheckout # timeout=10
> git checkout -f 7f50b4735b31248a93ad3e595ff7e104104390f # timeout=10
Commit message: "dockerfile"
> git rev-list --no-walk 7f50b4735b31248a93ad3e595ff7e104104390f # timeout=10
[mater-job] $ docker build -t pradeepacharya0/jenkilns_casestudy:v1 --pull=true /var/lib/jenkins/workspace/mater-job
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 265.2kB

Step 1/7 : FROM ubuntu:latest
```



The screenshot shows the Jenkins console output for build #5 of the 'mater-job' project. The build has completed successfully. The logs show the steps taken to build the Docker image, including pulling the 'ubuntu' base image, creating layers for the build context, and pushing the final image to the repository. The final digest is sha256:1f02a6c304e0c6fae37b34436e652506004cdfaaa2b1ed829e105e147c2edde. The size of the image is 1584 bytes.

```
ea038cd73ea: Preparing
73ed4d8eaef91c: Preparing
991d8c662162: Preparing
f36fd4bb7334: Preparing
f36fd4bb7334: Waiting
2f42d66b5f2f: Pushed
e0c0dfdbba8d: Pushed
f36fd4bb7334: Mounted from library/ubuntu
991d8c662162: Pushed
ea038cd73ea: Pushed
73ed4d8eaef91c: Pushed
v1: digest: sha256:1f02a6c304e0c6fae37b34436e652506004cdfaaa2b1ed829e105e147c2edde size: 1584
[mater-job] $ docker push pradeepacharya0/jenkilns_casestudy:latest
The push refers to repository [docker.io/pradeepacharya0/jenkilns_casestudy]
e0c0dfdbba8d: Preparing
2f42d66b5f2f: Preparing
ea038cd73ea: Preparing
73ed4d8eaef91c: Preparing
991d8c662162: Preparing
f36fd4bb7334: Preparing
f36fd4bb7334: Waiting
e0c0dfdbba8d: Layer already exists
73ed4d8eaef91c: Layer already exists
2f42d66b5f2f: Layer already exists
ea038cd73ea: Layer already exists
991d8c662162: Layer already exists
f36fd4bb7334: Layer already exists
latest: digest: sha256:1f02a6c304e0c6fae37b34436e652506004cdfaaa2b1ed829e105e147c2edde size: 1584
Finished: SUCCESS
```

The screenshot shows a DockerHub repository page for the user pradeepacharya0. The repository name is jenkilns_casestudy. The page displays basic repository information, including tags (latest, v1), automated build options, and a repository overview section.

Docker commands
To push a new tag to this repository:
`docker push pradeepacharya0/jenkilns_casestudy:tagname`

Tags
This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
latest	🐧	Image	2 minutes ago	3 minutes ago
v1	🐧	Image	2 minutes ago	3 minutes ago

[See all](#)

Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. [Read more about automated builds](#).

[Upgrade](#)

Repository overview INCOMPLETE
An overview describes what your image does and how to run it. It displays in [the public view of your repository](#) once you have pushed some content.
[Add overview](#)

Image is published in dockerhub.

In order to build the develop job, create a new Jenkins node, connect the node to the Jenkins server, and create the develop job. This job should be executed on the new node.

Creating a node called “develop”:

The screenshot shows the AWS EC2 "Launch an instance" wizard. The user is selecting the "Amazon Linux 2023.5.2" AMI and choosing the "t2.micro" instance type. A tooltip for the "Free tier" is visible, stating: "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".

Summary

Number of instances:

Software Image (AMI): Amazon Linux 2023.5.2... [read more](#)

ami-066784287e358dad1

Virtual server type (instance type): t2.micro

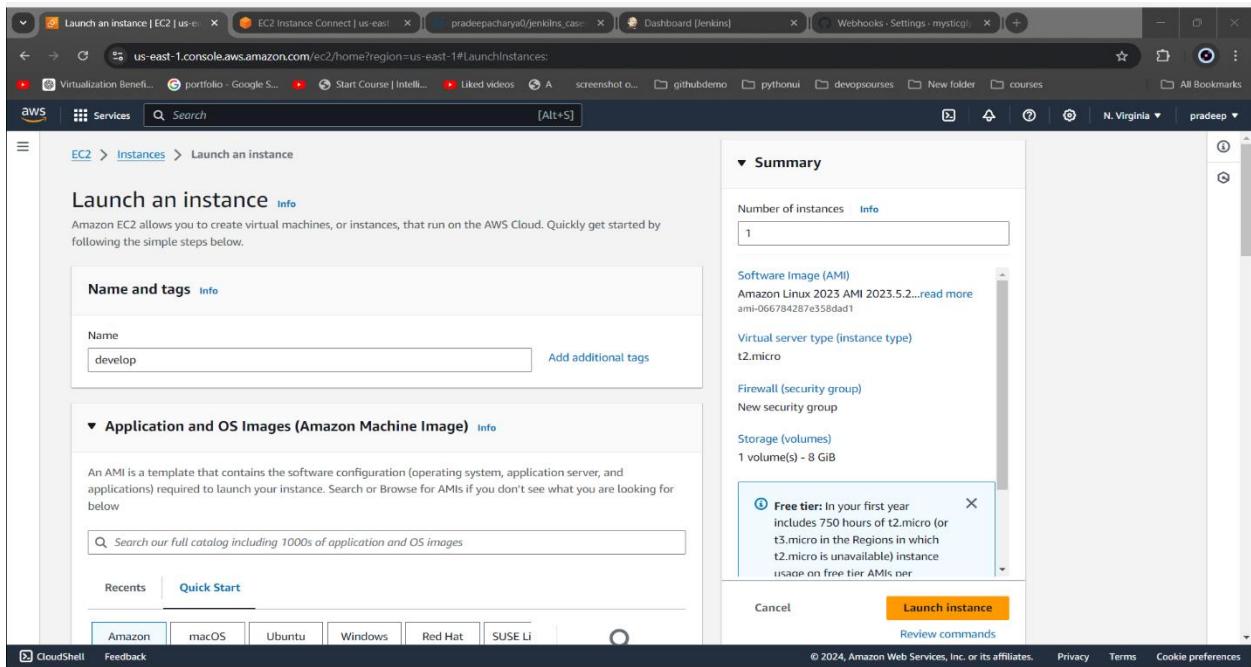
Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

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

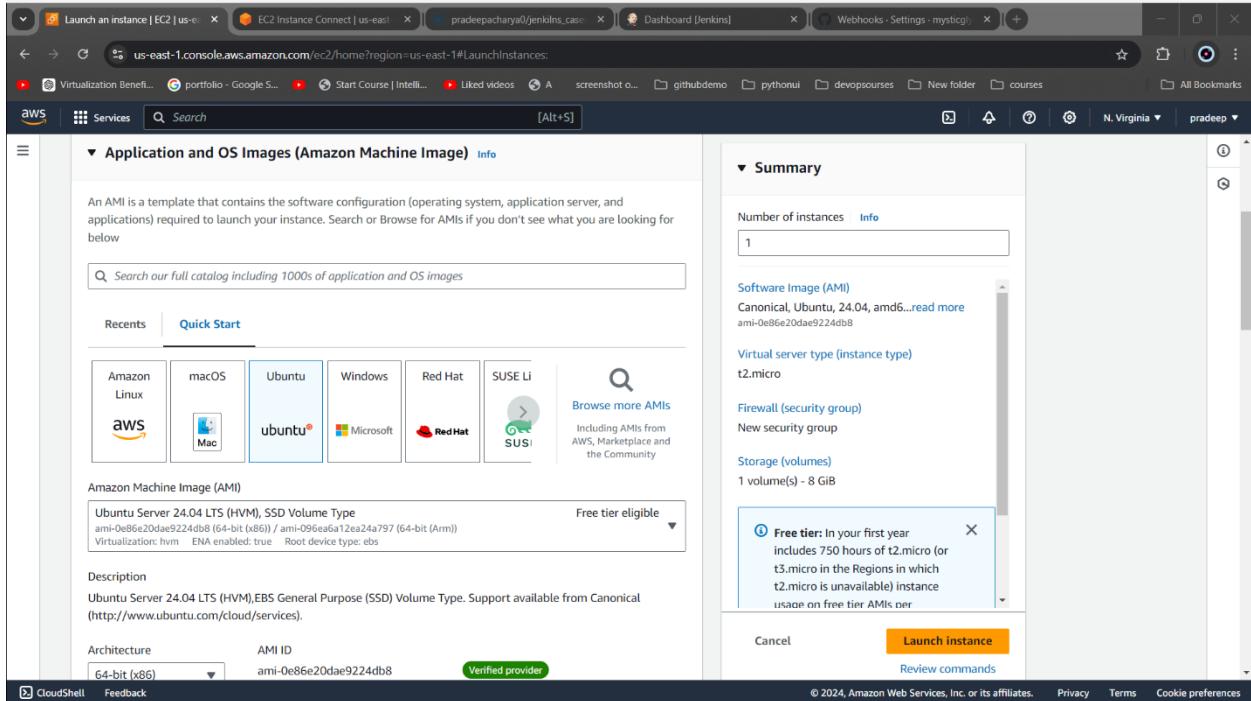
[Launch instance](#)

Instance name:



The screenshot shows the 'Launch an instance' page in the AWS EC2 console. In the 'Name and tags' section, the 'Name' field contains 'develop'. 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'.

AMI:



The screenshot shows the 'Application and OS Images (Amazon Machine Image)' page in the AWS EC2 console. The 'Ubuntu' tab is selected. 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'. The 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' AMI is highlighted.

Instance type:

The screenshot shows the AWS EC2 Launch Instances page. In the 'Instance type' section, a t2.micro instance is selected. A tooltip for 'Free tier eligible' indicates that it includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) usage on free tier AMIs per year. The 'Summary' section shows 1 instance being launched. The 'Software Image (AMI)' is set to Canonical, Ubuntu, 24.04, amd64. The 'Virtual server type (instance type)' is t2.micro. Under 'Key pair (login)', a key pair named 'newjenkins' is selected. The 'Network settings' section is partially visible at the bottom.

key pair:

The screenshot shows the AWS EC2 Launch Instances page. The 'Key pair (login)' section is expanded, showing the message: 'You can use a key pair to securely connect to your instance. Ensure that you have access to before you launch the instance.' A dropdown menu for 'Key pair name - required' has 'newjenkins' selected. The 'Network settings' section is partially visible at the bottom.

Default Network settings:

The screenshot shows the AWS EC2 'Launch instances' wizard. In the 'Network settings' section, it lists a VPC (vpc-0ad7d342053e758cb) and a subnet (No preference). It includes fields for creating a new security group ('Create security group') or selecting an existing one ('Select existing security group'). Under 'Additional charges apply' (when outside of free tier allowance), there are three checkboxes: 'Allow SSH traffic from Anywhere (0.0.0.0/0)', 'Allow HTTPS traffic from the internet', and 'Allow HTTP traffic from the internet'. A note below states: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' In the 'Summary' section, it shows 1 instance, the Canonical Ubuntu 24.04 AMI, t2.micro instance type, and a new security group. A tooltip for the 'Free tier' indicates usage on free tier AMIs per year. The 'Launch instance' button is highlighted.

Default Configure storage:

The screenshot shows the AWS EC2 'Launch instances' wizard. In the 'Configure storage' section, it shows 1x 8 GiB gp3 root volume (Not encrypted). A tooltip indicates that free-tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Below this, it says 'The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance'. It also says 'Click refresh to view backup information' and '0 x File systems'. In the 'Advanced details' section, there is an 'Edit' link. In the 'Summary' section, it shows 1 instance, the Canonical Ubuntu 24.04 AMI, t2.micro instance type, and a new security group. A tooltip for the 'Free tier' indicates usage on free tier AMIs per year. The 'Launch instance' button is highlighted.

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

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
i-01bda22ee0e85ac7	Terminated	t2.micro	-	-	View alarms	us-east-1b	-
i-00ce4e0c3301b9b3	Terminated	t2.micro	-	-	View alarms	us-east-1c	-
i-06736b488b6699751	Terminated	t2.micro	-	-	View alarms	us-east-1e	-
jenkins_server	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-34-22-	-
develop	i-0225cc1a209633578	Running	t2.micro	Initializing	View alarms	us-east-1a	ec2-52-90-
i-00a92e80675ce28b4	Terminated	t2.micro	-	-	View alarms	us-east-1a	-

Below the table, the details for the selected instance 'i-0225cc1a209633578 (develop)' are shown. The 'Details' tab is selected, displaying information such as Instance ID, Public IPv4 address (52.90.199.141), Instance state (Running), and Private IP DNS name (ip-172-31-94-94.ec2.internal).

Connect to develop instance and install java and docker:

The screenshot shows the AWS CloudShell terminal connected to the 'develop' instance. The terminal output is as follows:

```

Memory usage: 4%   IPv4 address for eth0: 10.10.10.2
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

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

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-172-31-94-94:~$ 1
ubuntu@ip-172-31-94-94:~$ ls
ubuntu@ip-172-31-94-94:~$ sudo su
root@ip-172-31-94-94:/home/ubuntu# []

```

After the terminal session ends, a summary message is displayed:

```

i-0225cc1a209633578 (develop)
PublicIPs: 52.90.199.141  PrivateIPs: 172.31.94.94

```

Connected.

Installing docker:

→ Apt install docker.io -y

```
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [463 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [114 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [7192 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [337 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [142 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [13.6 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [280 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [54.8 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 kB]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.3 kB]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1016 B]
Get:45 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:46 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:47 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:48 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 28.6 MB in 4s (6453 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
86 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-94-94:/home/ubuntu# apt install docker.io -y
```

i-0225cc1a209633578 (develop)
PublicIPs: 52.90.199.141 PrivateIPs: 172.31.94.94

Docker started :

```
Setting up pigz (2.8-1) ...
Setting up containerd (1.7.12-0ubuntu4.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.7-0ubuntu0.1) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group 'docker' (GID 113) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

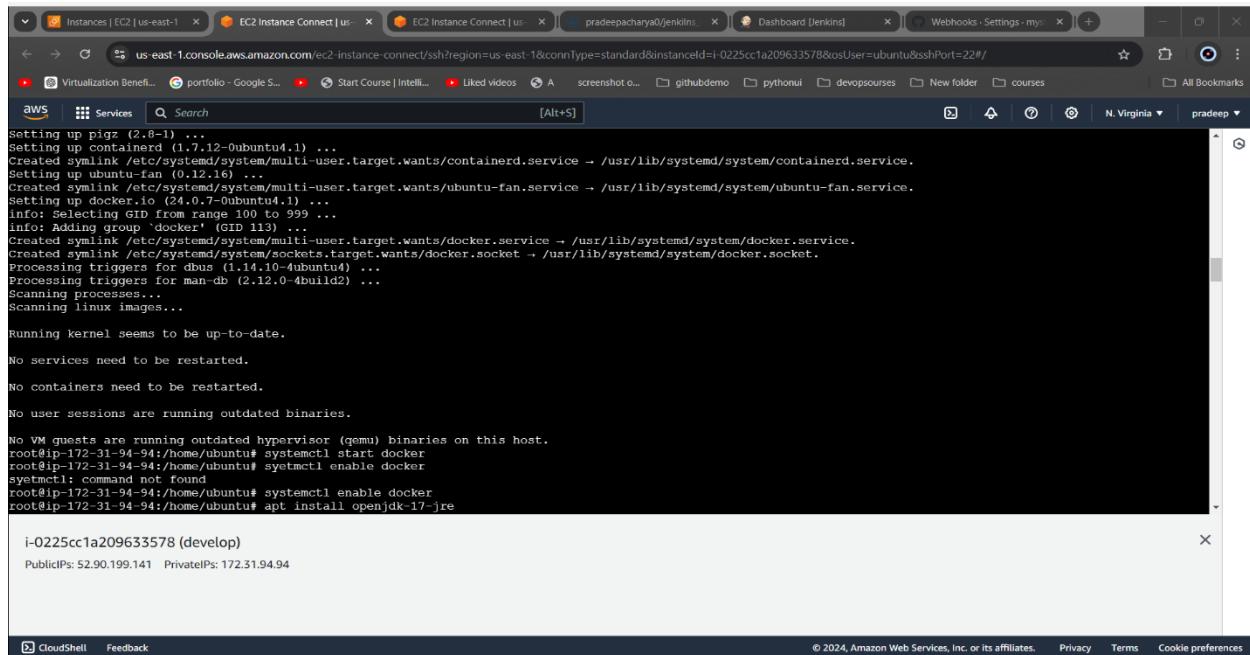
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-94-94:/home/ubuntu# systemctl start docker
root@ip-172-31-94-94:/home/ubuntu# systemctl enable docker
systemctl: command not found
root@ip-172-31-94-94:/home/ubuntu# systemctl enable docker
root@ip-172-31-94-94:/home/ubuntu# 
```

i-0225cc1a209633578 (develop)
PublicIPs: 52.90.199.141 PrivateIPs: 172.31.94.94

Installing java-17:

→ Apt install openjdk-17-jre



```
Setting up pigz (2.8-1) ...
setting up containerd (1.7.12-0ubuntu1.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.7-0ubuntu0.1) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group 'docker' (GID 113) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created socket /lib/systemd/system/sockets.hangit.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

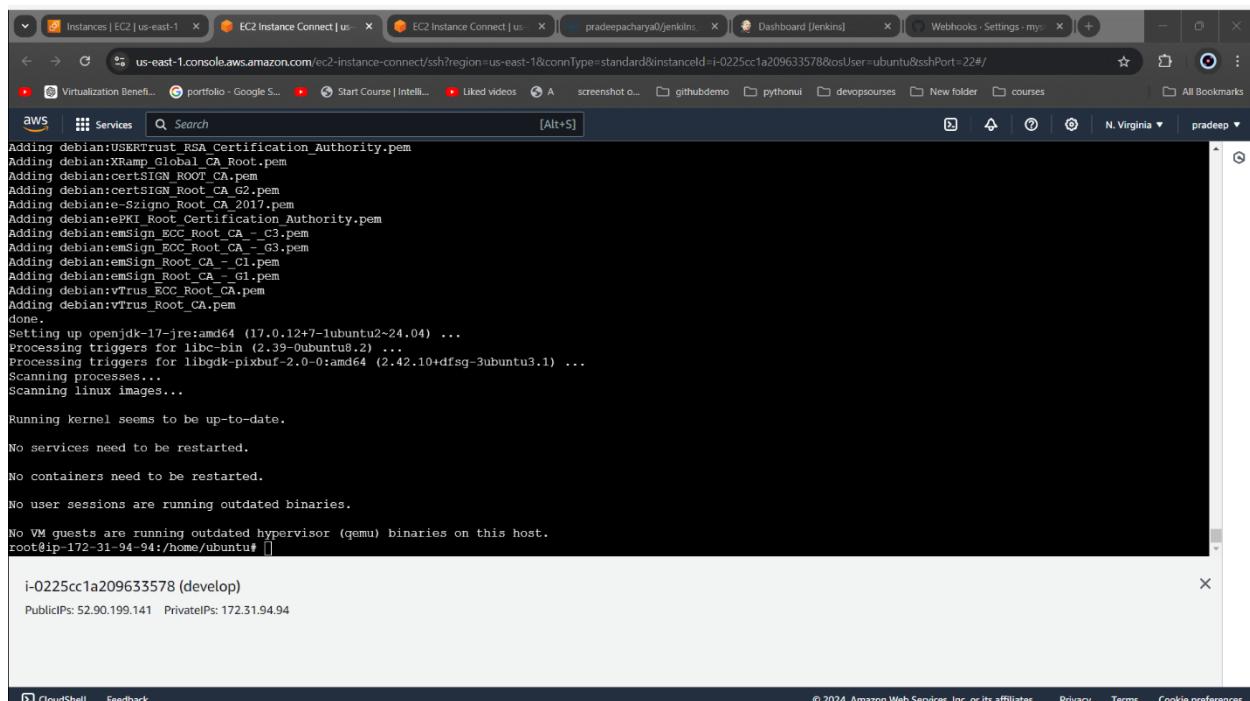
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-94-94:/home/ubuntu# systemctl start docker
root@ip-172-31-94-94:/home/ubuntu# systemctl enable docker
Systemctl: command successful.
root@ip-172-31-94-94:/home/ubuntu# apt install openjdk-17-jre
i-0225cc1a209633578 (develop)

Public IPs: 52.90.199.141 Private IPs: 172.31.94.94
```

Installed:



```
Adding debian:USERTrust RSA Certification Authority.pem
Adding debian:X509 Global CA Root.pem
Adding debian:crtSIGN ROOT CA.pem
Adding debian:crtSIGN Root CA G2.pem
Adding debian:crtSIGN Root CA G3.pem
Adding debian:e-Sign Root CA_2017.pem
Adding debian:epK1 Root Certification_Authority.pem
Adding debian:emSign ECC Root CA _C3.pem
Adding debian:emSign ECC Root CA _G3.pem
Adding debian:emSign Root CA _C1.pem
Adding debian:emSign Root CA _G1.pem
Adding debian:virus ECC Root CA.pem
Adding debian:virus_ROOT_CA.pem
done.
Setting up openjdk-17-jre:amd64 (17.0.12+7-lubuntu2~24.04) ...
Processing triggers for libc-bin (2.39-0ubuntu0.2) ...
Processing triggers for libgdk-pixbuf-2.0-0:amd64 (2.42.10+dfsg-3ubuntu3.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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-172-31-94-94:/home/ubuntu#
```

Connect the node to jenkins:

The screenshot shows the Jenkins dashboard with a single job listed: 'master_job'. The last success was 24 min ago, and the last failure was 26 min ago. The job's URL is [#4 pradeepacharya0/jenkins_casestudy/v1](#). The Jenkins version is 2.462.1.

→ Manage jenkins-> node -> add new node

The screenshot shows the 'Manage Jenkins' page under 'System Configuration'. It includes sections for 'System', 'Tools', 'Plugins', 'Security', 'Users', 'Credentials', and 'Clouds'. A warning message is displayed: 'CSRF vulnerability and missing permission check (no fix available)'. The Jenkins version is 2.462.1.

The screenshot shows the Jenkins 'Nodes' page. At the top, there are tabs for 'Instances | EC2 | us-east-1', 'EC2 Instance Connect | us...', 'pradeepacharya0/jenkins...', 'Webhooks - Settings - my...', and 'Nodes [Jenkins]'. The main content area has a title 'Nodes' and a table with one row:

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-in Node	Linux (amd64)	In sync	3.73 GiB	0 B	3.73 GiB	0ms
	Data obtained		27 min	27 min	27 min	27 min	27 min

On the left, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). On the right, there is a 'Legend' section and links for 'REST API' and 'Jenkins 2.462.1'.

In order to add new node click on new node:

Node name and type:

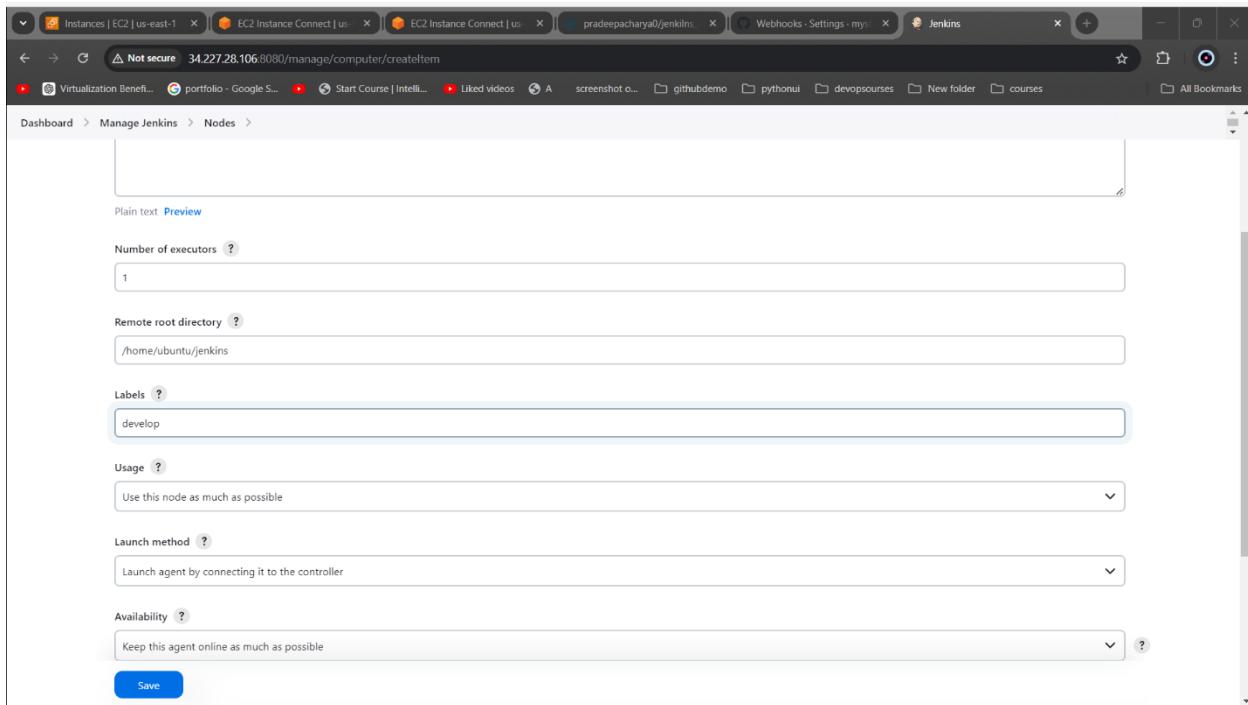
The screenshot shows the 'New node' configuration page. The URL is '34.227.28.106:8080/manage/computer/new'. The page has a title 'New node' and a form with the following fields:

- Node name:** A text input field containing 'develop'.
- Type:** A radio button group where 'Permanent Agent' is selected.
- Description:** A small text area with placeholder text about permanent agents.
- Create:** A blue 'Create' button at the bottom.

At the bottom right, there are links for 'REST API' and 'Jenkins 2.462.1'.

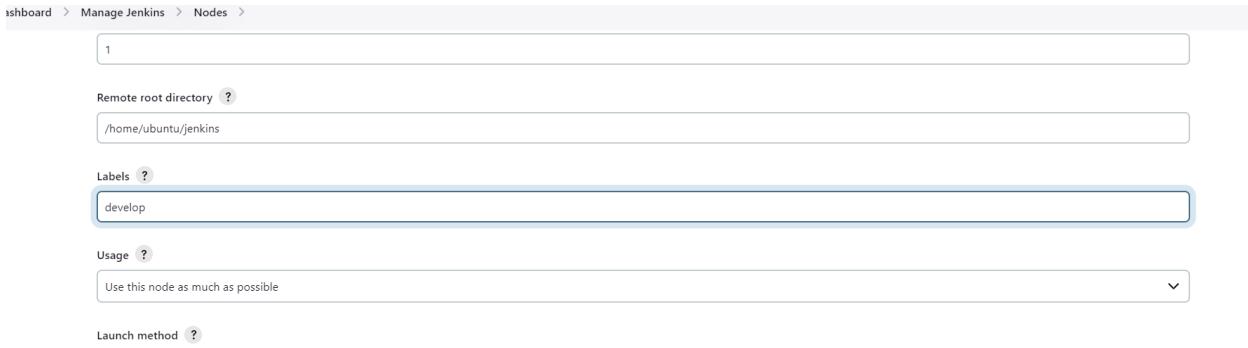
Here in remote root directory we need to provide the jenkins home directory in develop_node.

/home/ubuntu/jenkins



The screenshot shows the Jenkins management interface for creating a new node. The 'Remote root directory' field is highlighted and contains the value '/home/ubuntu/jenkins'. Other fields shown include 'Number of executors' (1), 'Labels' (develop), 'Usage' (Use this node as much as possible), 'Launch method' (Launch agent by connecting it to the controller), and 'Availability' (Keep this agent online as much as possible). A 'Save' button is at the bottom.

Labels:

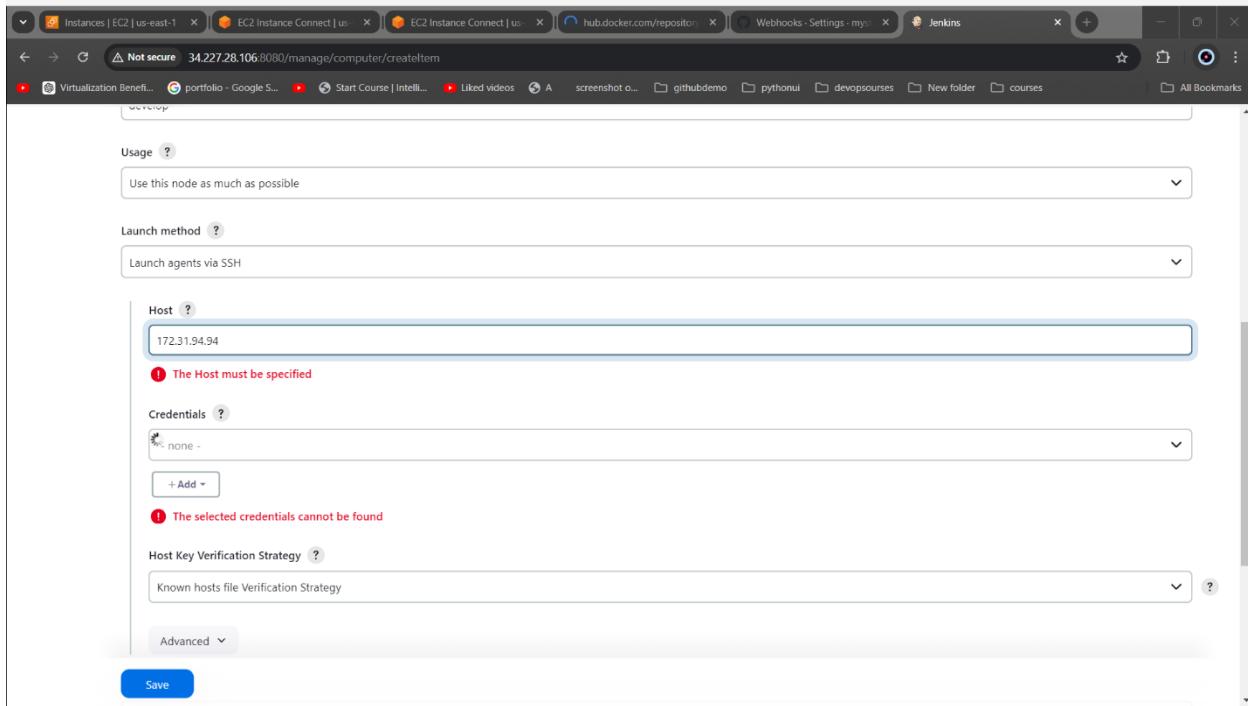


The screenshot shows the Jenkins management interface for creating a new node. The 'Labels' field is highlighted and contains the value 'develop'. Other fields shown include 'Number of executors' (1), 'Remote root directory' (/home/ubuntu/jenkins), 'Usage' (Use this node as much as possible), and 'Launch method' (Launch agent by connecting it to the controller).

Launch method:

Via ssh :

In host we need to provide the private IP address:



Now we need to provide credentials so we need to create credentials:

Adding credentials:

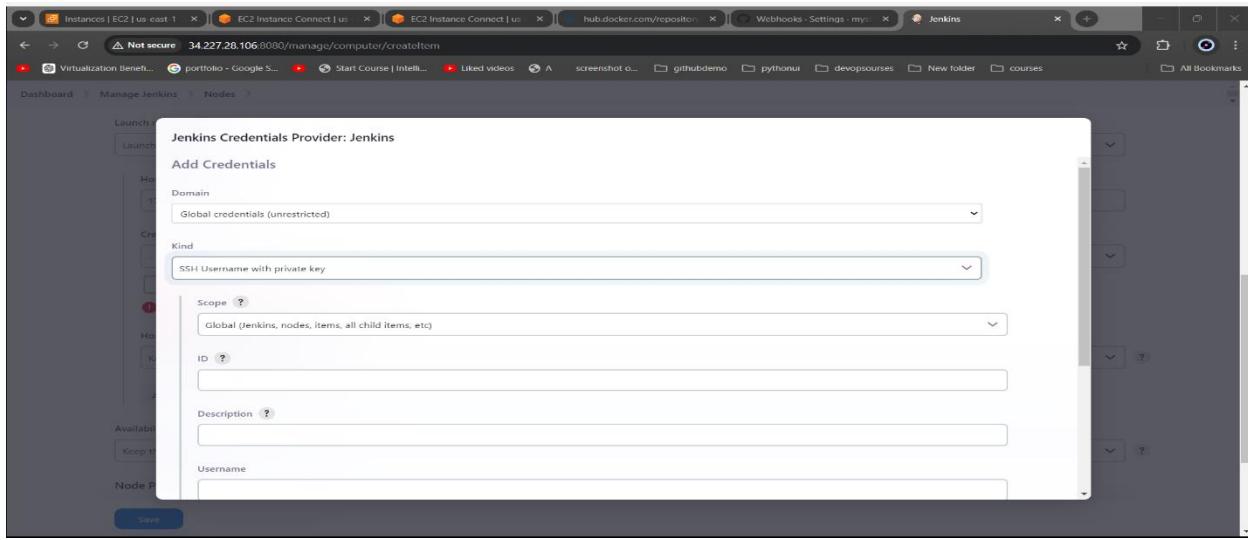
172.31.94.94

Credentials ?

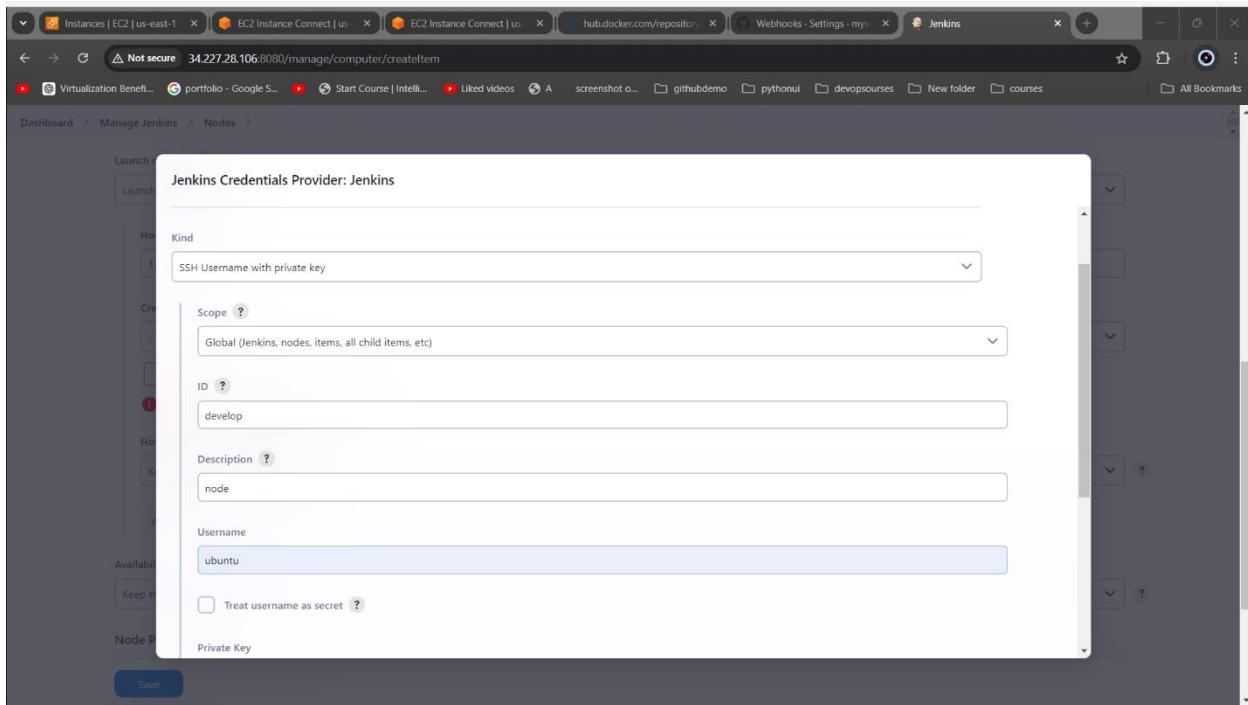
- none -

+ Add ▾

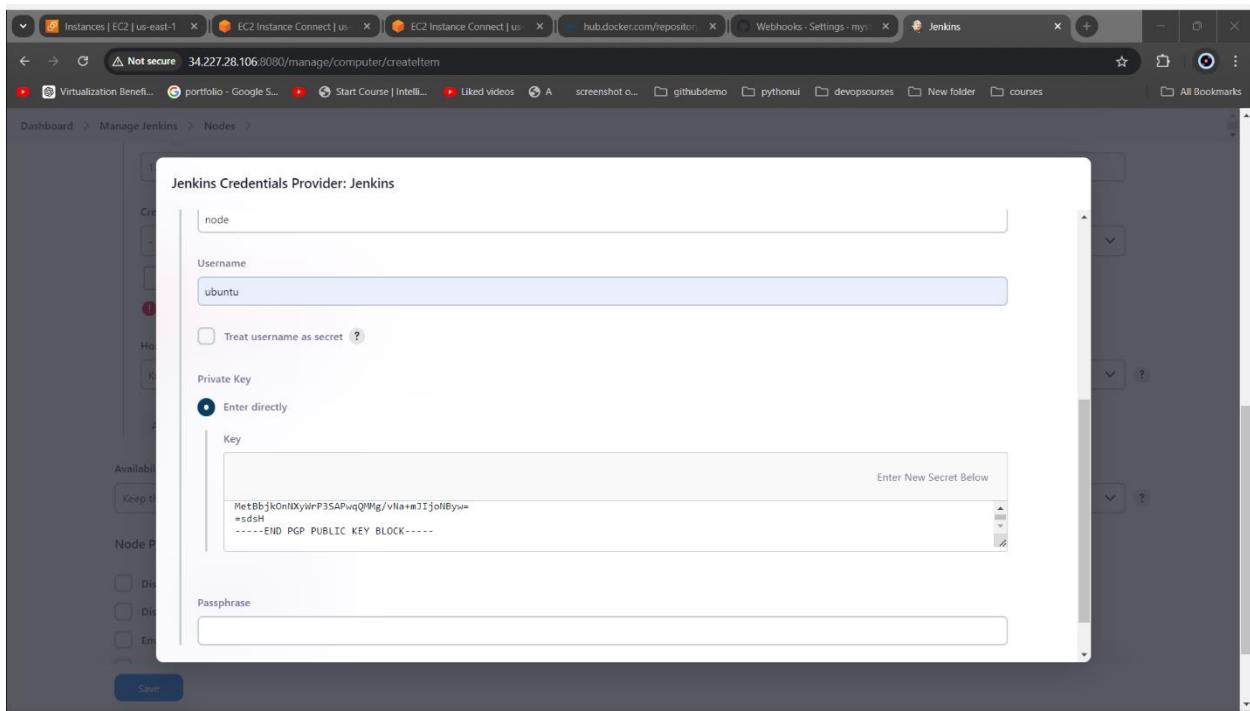
! The selected credentials cannot be found



Providing id,description,etc,username.

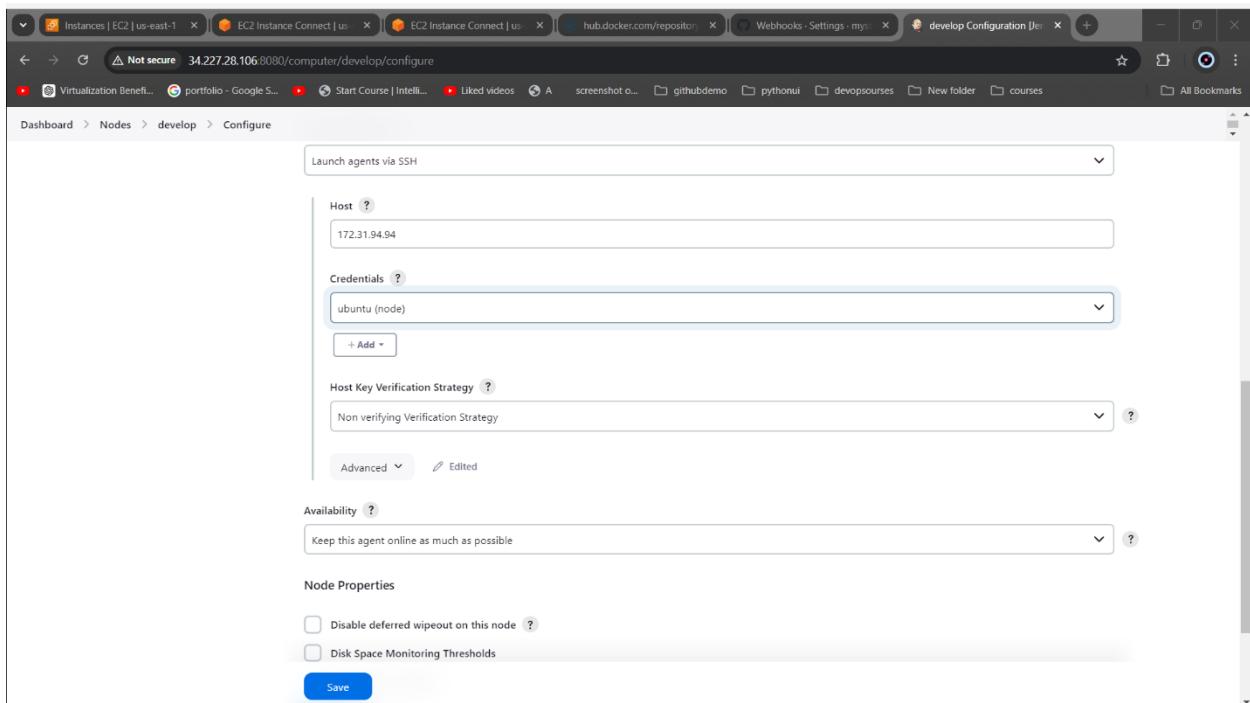


Private Key:



Private key is added.

Selecting the credentials:



Host Key Verification Strategy:

Host Key Verification Strategy: Non verifying Verification Strategy

Credentials: pradeepacharya0/***** (casestd)

Availability: Keep this agent online as much as possible

Node Properties:

- Disable deferred wipeout on this node
- Disk Space Monitoring Thresholds
- Environment variables
- Tool Locations

Save

And save.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.23 GiB	0 B	3.23 GiB	0ms
	develop	Linux (amd64)	In sync	4.17 GiB	0 B	4.17 GiB	67ms
	Data obtained	4 min 7 sec	4 min 7 sec	4 min 7 sec	4 min 7 sec	4 min 7 sec	4 min 7 sec

Nodes

New Node

Configure Monitors

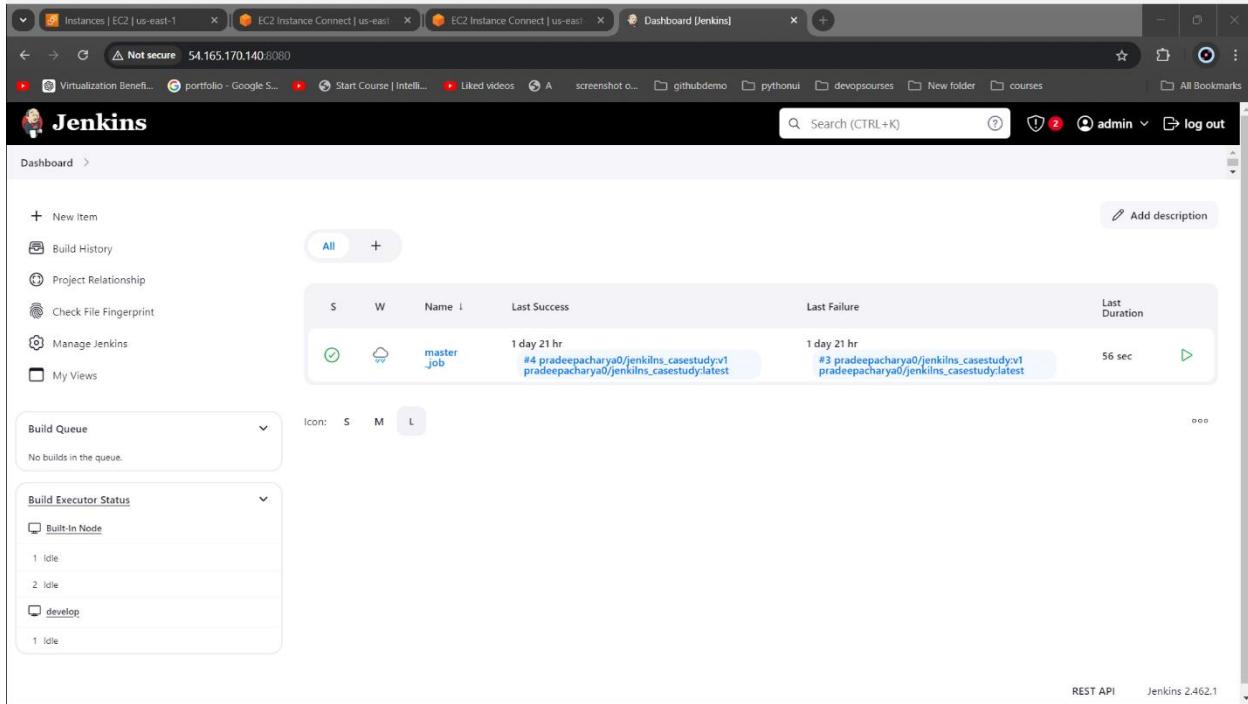
Legend

REST API Jenkins 2.462.1

Node is connected.

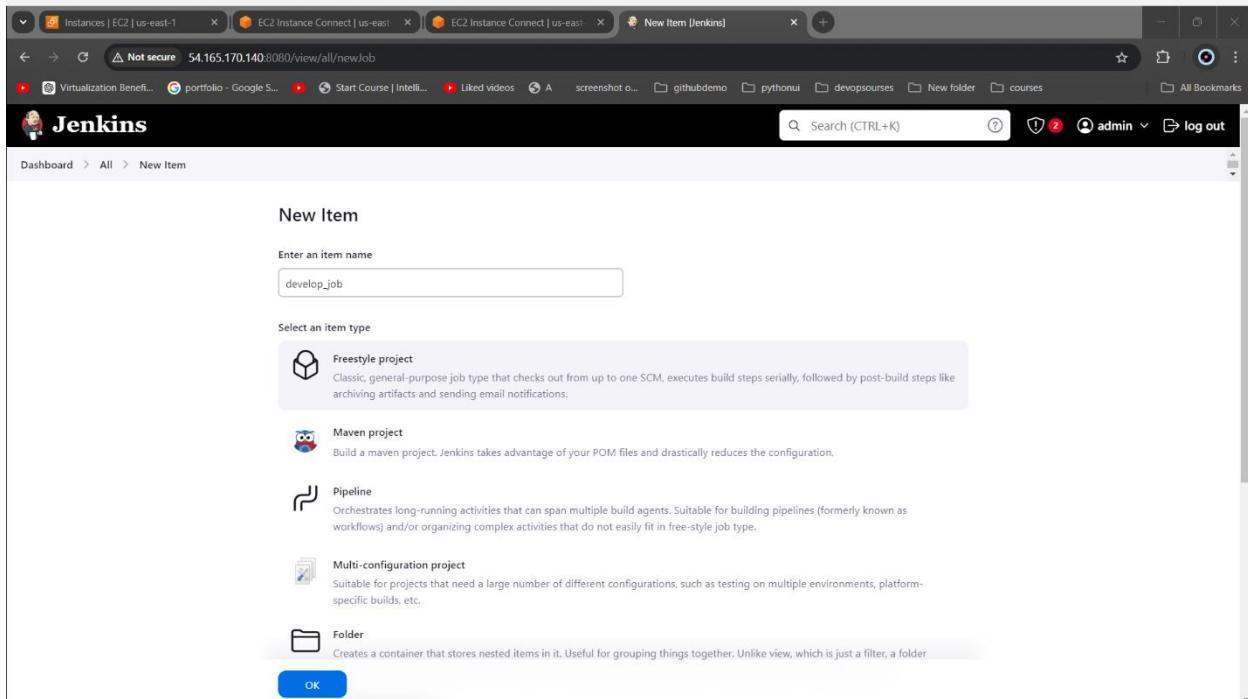
In order to do this, if a commit is made to the develop branch, just build the product without publishing it. Let's create a job for this.

NAME : develop



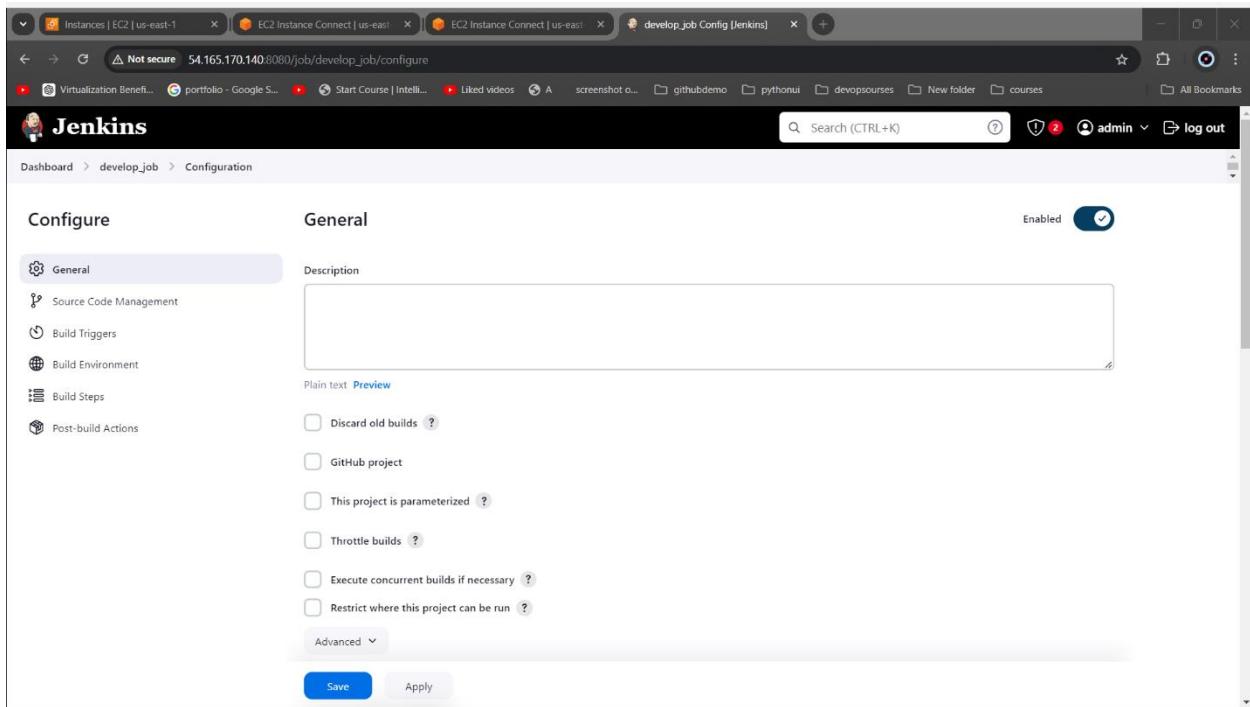
The screenshot shows the Jenkins dashboard at <http://54.165.170.140:8080>. The 'master_job' configuration is listed under the 'All' tab. It has a status icon (green checkmark), a build history icon (cloud with dots), and a name 'master_job'. The 'Last Success' was 1 day 21 hr ago, build #4. The 'Last Failure' was 1 day 21 hr ago, build #3. The 'Last Duration' was 56 sec. Below the main table, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (one 'develop' node idle).

Name:develop _job and the type will be free style:



The screenshot shows the 'New Item' dialog in Jenkins. In the 'Enter an item name' field, 'develop_job' is typed. Under 'Select an item type', the 'Freestyle project' option is selected, described as a classic, general-purpose job type. Other options shown include 'Maven project', 'Pipeline', 'Multi-configuration project', and 'Folder'. A blue 'OK' button is at the bottom.

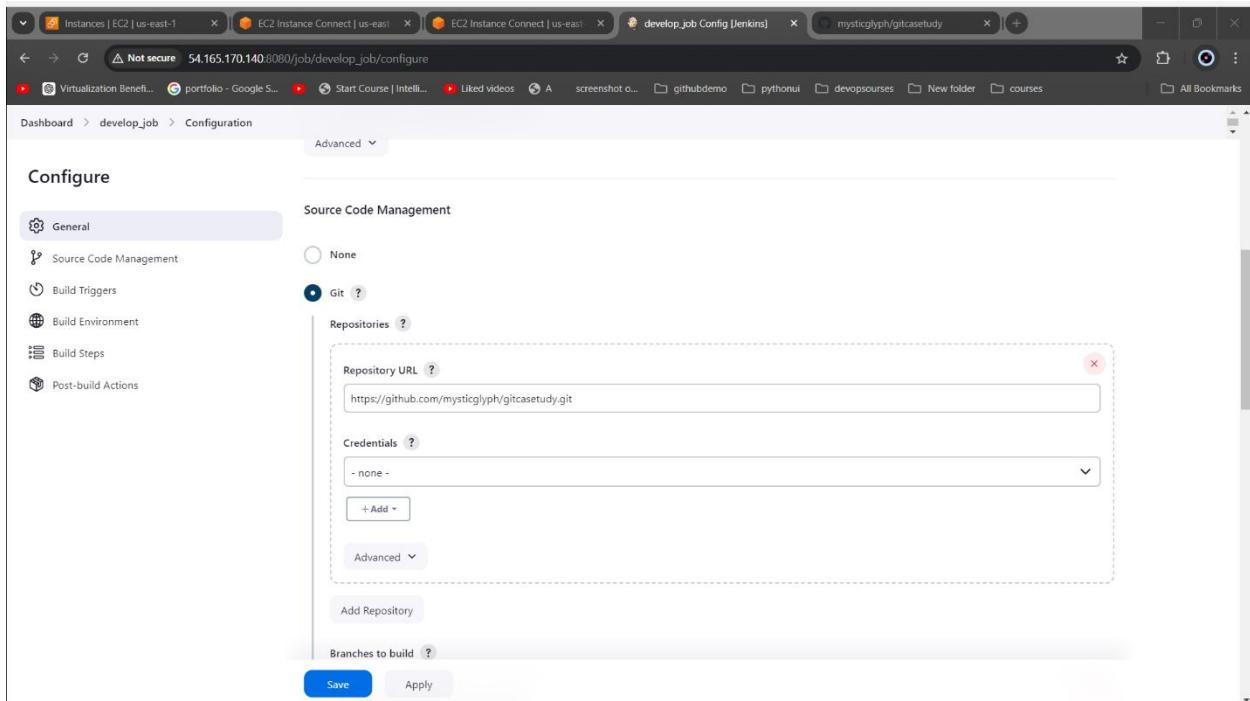
Job is created:



The screenshot shows the Jenkins configuration interface for a job named "develop_job". The "General" tab is selected. The "Enabled" switch is turned on. The "Description" field is empty. Under "Post-build Actions", several options are listed: "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.

Source Code Management:

Git



The screenshot shows the Jenkins configuration interface for the "develop_job" job, specifically under the "Source Code Management" section. The "Git" option is selected. The "Repository URL" is set to "https://github.com/mysticglyph/gitcasestudy.git". The "Credentials" dropdown is set to "- none -". There is an "Add Repository" button at the bottom. At the very bottom are "Save" and "Apply" buttons.

Branch Specifier:

The screenshot shows the Jenkins job configuration page for 'develop_job'. In the 'Branches to build' section, the 'Branch Specifier' field contains the value '/develop'. This indicates that the Jenkins job will build any code pushed to the 'develop' branch of the repository.

Build Triggers:

The screenshot shows the Jenkins job configuration page for 'develop_job'. In the 'Build Triggers' section, the 'GitHub hook trigger for GITScm polling' option is selected, indicated by a checked checkbox. This means the Jenkins job will automatically trigger a build whenever there are changes pushed to the repository via GitHub.

Select execute docker command:

Restrict where this project can be run:

This screenshot shows the Jenkins job configuration page for a job named 'develop_job'. In the 'Restriction' section, the checkbox for 'Restrict where this project can be run' is checked. Below it, the 'Label Expression' field contains 'develop'. A note below the field states: 'Label develop matches 1 node. Permissions or other restrictions provided by plugins may further reduce the available nodes.'

Source Code Management

Build Steps:

In order to build we will use exixute shell:

This screenshot shows the Jenkins job configuration page for the 'develop_job'. The 'Build Environment' step is selected in the sidebar. A dropdown menu is open, listing various build steps: Docker Build and Publish, Execute Docker command, Execute Windows batch command, Execute shell, 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 dropdown, there is an 'Add build step' button.

Docker command for build a image:

The screenshot shows the Jenkins job configuration page for 'develop_job'. In the 'Build Steps' section, there is a single 'Execute shell' step defined with the command:

```
sudo docker build . -t myimage
```

Below the build steps, there is a 'Post-build Actions' section with a 'Save' button.

Job is executed successfully:

The screenshot shows the Jenkins job console output for 'develop_job #3'. The output displays the Docker build logs:

```
Install the buildx component to build images with BuildKit:  
https://docs.docker.com/go/buildx/  
  
Sending build context to Docker daemon 265.2kB  
  
Step 1/7 : FROM ubuntu:latest  
--> eddfef74c41f8  
Step 2/7 : RUN apt update -y  
--> Using cache  
--> 2f109362de03  
Step 3/7 : RUN apt install apache2 -y  
--> Using cache  
--> 849960288883  
Step 4/7 : RUN apt install git -y  
--> Using cache  
--> 5a89d34f748b  
Step 5/7 : RUN cd /var/www/html && rm -rf * && git clone https://github.com/mysticglyph/gitcasestudy.git  
--> Using cache  
--> 343bf9746884  
Step 6/7 : RUN mv /var/www/html/gitcasestudy/* /var/www/html  
--> Using cache  
--> ab8500ac6b9c  
Step 7/7 : ENTRYPOINT apachectl -D FOREGROUND  
--> Using cache  
--> dc03e925da3a  
Successfully built dc03e925da3a  
Successfully tagged mynewimage:latest  
Finished: SUCCESS
```

The screenshot shows a web browser window with multiple tabs open. The active tab is 'develop_job [Jenkins]' at the URL 54.165.170.140:8080/job/develop_job/. The page title is 'Jenkins' with a checkmark icon. The main content area is titled 'develop.job'. On the left, there's a sidebar with links like Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, and Rename. The 'Status' section shows a green checkmark and the text 'develop.job'. Below it is a 'Permalinks' section with a link to 'Add description'. The central part of the page is titled 'Build History' with a dropdown menu set to 'trend'. It lists four builds:

- Last build (#3), 1 min 42 sec ago
- Last stable build (#3), 1 min 42 sec ago
- Last successful build (#3), 1 min 42 sec ago
- Last failed build (#1), 2 hr 28 min ago
- Last unsuccessful build (#1), 2 hr 28 min ago
- Last completed build (#3), 1 min 42 sec ago

At the bottom of the build history list, there are two links: 'Atom feed for all' and 'Atom feed for failures'. The browser's address bar shows the URL http://54.165.170.140:8080/job/develop_job/.

Now we need to create a container with Ubuntu and Apache installed in it. Then, create a new job.

The screenshot shows the Jenkins dashboard with two active jobs listed:

Name	Last Success	Last Failure	Last Duration
develop_job	8 min 9 sec #3	2 hr 35 min #1	0.63 sec
master_job	1 day 23 hr #4 pradeepacharya0/jenkins_casestudy:v1 pradeepacharya0/jenkins_casestudy:latest	1 day 23 hr #3 pradeepacharya0/jenkins_casestudy:v1 pradeepacharya0/jenkins_casestudy:latest	56 sec

Creating container:

Name of the container and type of the container:

The screenshot shows the 'New Item' creation dialog. The item name is 'DEPOLY' and the item type is selected as 'Freestyle project'.

Other options shown in the list:

- Maven project
- Pipeline
- Multi-configuration project
- Folder

Job is created.

The screenshot shows the Jenkins configuration interface for a job named 'DEPOLY'. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The 'General' tab is selected. The main panel has a title 'General' and a status 'Enabled' with a checked checkbox. A 'Description' field is empty. Below it is a 'Plain text' preview area containing several checkboxes for build options: '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.

Configuring the job:

We are using a pug in for creating a docker whis is previously downloaded.

→ Pug in: docker compose build step

So . in build step:

The screenshot shows the Jenkins configuration interface for the 'Build Environment' section of the 'DEPOLY' job. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', and 'Build Steps'. The 'Build Environment' tab is selected. A 'Filter' dropdown is open, showing a list of available build steps: 'Docker Build and Publish', 'Execute Docker command', 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. Below the filter is an 'Add build step' button. Further down is a 'Post-build Actions' section with an 'Add post-build action' button.

We will use execute shell:

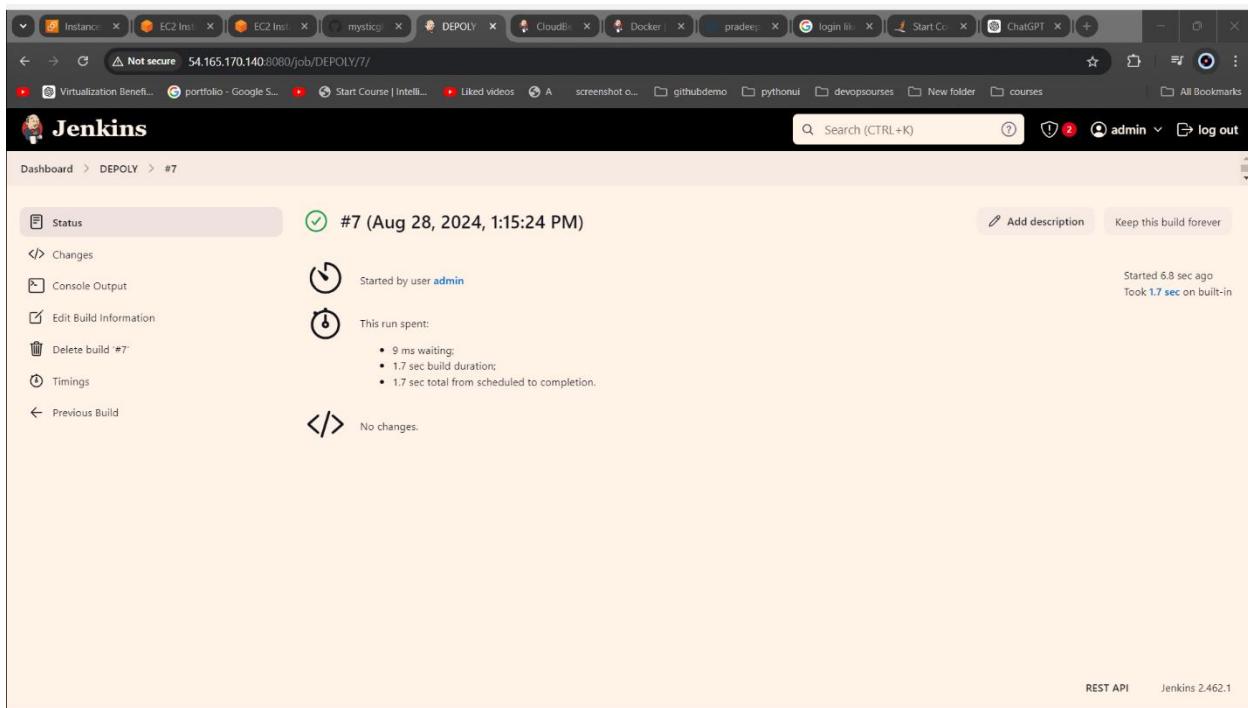
The screenshot shows a Jenkins job configuration page. The 'Execute Shell' build step is selected, displaying a shell script command. The command is as follows:

```
count=$(docker ps | grep mycontainer | wc -l )
if[ $count -eq 1 ]
then
    docker stop mycontainer
    docker rm mycontainer
    sudo docker run -d --name mycontainer -p 80:80 pradeepacharya0/jenkilns_casestudy
```

Below the command, there is an 'Advanced' section with a dropdown menu labeled 'Add build step'. At the bottom of the screen, there are 'Save' and 'Apply' buttons.

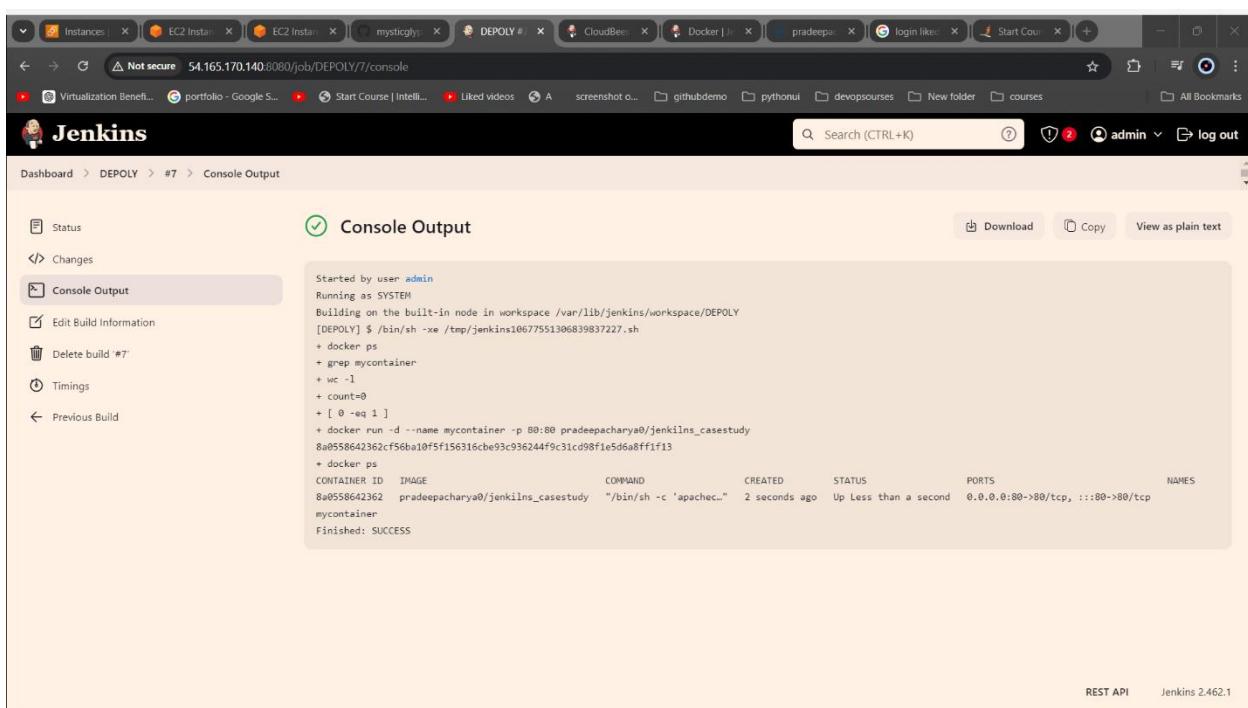
```
count=$(docker ps | grep mycontainer | wc -l )
if[ $count -eq 1 ]
then
    docker stop mycontainer
    docker rm mycontainer
    sudo docker run -d --name mycontainer -p 80:80 pradeepacharya0/jenkilns_casestudy
else
    sudo docker run -d --name mycontainer -p 80:80 pradeepacharya0/jenkilns_casestudy
fi
docker ps
```

this will be the command for creating a container.



The screenshot shows the Jenkins job details for '#7 (Aug 28, 2024, 1:15:24 PM)'. The job was started by user 'admin' and took 1.7 seconds on built-in. The build status is successful. The console output section is collapsed.

Job executed successfully.



The screenshot shows the Jenkins job details for '#7 (Aug 28, 2024, 1:15:24 PM)'. The job was started by user 'admin' and took 1.7 seconds on built-in. The build status is successful. The console output section is expanded, showing the command history and Docker logs.

```
Started by user admin
Running as SYSTEM
Building on the built-in node in workspace /var/lib/jenkins/workspace/DEPOLY
[DEPOLY] $ /bin/sh -xe /tmp/jenkins10677551306839837227.sh
+ docker ps
+ grep mycontainer
+ wc -l
+ count=0
+ [ 0 -eq 1 ]
+ docker run -d --name mycontainer -p 80:80 pradeepacharya0/jenkilns_casestudy
8a0558642362cf56ba10f5f156316be93c936244f9c31cd98file5d6a8ff1f13
+ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8a0558642362 pradeepacharya0/jenkilns_casestudy "/bin/sh -c 'apache'" 2 seconds ago Up Less than a second 0.0.0.0:80->80/tcp, :::80->80/tcp
mycontainer
Finished: SUCCESS
```

```

at Jenkins Main ClassLoader//org.eclipse.jetty.server.ServerConnector.openAcceptChannel(ServerConnector.java:344)
at Jenkins Main ClassLoader//org.eclipse.jetty.server.ServerConnector.open(ServerConnector.java:304)
at Jenkins Main ClassLoader//org.eclipse.jetty.server.Server.lambda$doStart$0(Server.java:402)
at java.base/java.util.stream.ForEachOps$ForEachOp$ofRef.accept(ForEachOps.java:183)
at java.base/java.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.java:197)
at java.base/java.util.stream.ReferencePipeline$2$1.accept(ReferencePipeline.java:179)
at java.base/java.util.Spliterators$ArraySpliterator.forEachRemaining(Spliterators.java:992)
at java.base/java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:509)
at java.base/java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:499)
at java.base/java.util.stream.ForEachOps$ForEachOp.evaluateSequential(ForEachOps.java:150)
at java.base/java.util.stream.ForEachOps$ForEachOp$ofRef.evaluateSequential(ForEachOps.java:173)
at java.base/java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)
at java.base/java.util.stream.ReferencePipeline$2$1.evaluate(ReferencePipeline.java:156)
at Jenkins Main ClassLoader//org.eclipse.jetty.server.Server.lambda$doStart$0(Server.java:398)
at Jenkins Main ClassLoader//org.eclipse.jetty.util.component.AbstractLifeCycle.start(AbstractLifeCycle.java:93)
at Jenkins Main ClassLoader/Winstone.Launcher.<init>(Launcher.java:205)
Caused: java.io.IOException: Failed to start Jetty
at Jenkins Main ClassLoader/Winstone.Launcher.<init>(Launcher.java:209)
at Jenkins Main ClassLoader/Winstone.Launcher.main(Launcher.java:495)
at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)
at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.base/java.lang.reflect.Method.invoke(Method.java:569)
at executable.Main.main(Main.java:35)
root@ip-172-31-89-21:/home/ubuntu# sudo systemctl restart jenkins
root@ip-172-31-89-21:/home/ubuntu# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9a0558642362 pradeepacharya0/jenkins_casestudy "/bin/sh -c 'apache_..." 2 minutes ago Up 2 minutes 0.0.0.0:80->80/tcp, :::80->80/tcp mycontainer
root@ip-172-31-89-21:/home/ubuntu#

```

i-05526c5b82a086990 (jenkins_server)
Public IPs: 54.165.170.140 Private IPs: 172.31.89.21

We can see that in server container is created.

Post-build Actions:

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Docker registry URL ?
pradeepacharya0/***** (casestd)

Registry credentials
+ Add

Advanced
Add build step

Post-build Actions
Add post-build action

Save Apply REST API

Build other project:

The screenshot shows the Jenkins configuration interface for the 'master_job'. The left sidebar has 'Post-build Actions' selected. The main area displays the 'Build other projects' action. Under 'Projects to build', there is a text input field containing 'develop_job'. Below it, three radio button options are shown: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. At the bottom are 'Save' and 'Apply' buttons.

This screenshot is identical to the one above, showing the Jenkins configuration for the 'master_job'. The 'Post-build Actions' section is open, and the 'Build other projects' action is configured to trigger on a stable build for the project 'develop_job'.

Save.

case study is completed

Conclusion

The Jenkins CI/CD pipeline implementation for the given case study successfully integrates essential DevOps tools and practices to streamline software development and deployment. By setting up automated Git workflows, the pipeline ensures code quality and consistency through triggered builds on every commit to the master or develop branches. Additionally, the containerization approach, leveraging an Ubuntu environment with Apache, facilitates efficient code building and deployment. This solution not only enhances operational efficiency but also supports the organization's goal of delivering reliable and scalable software products.