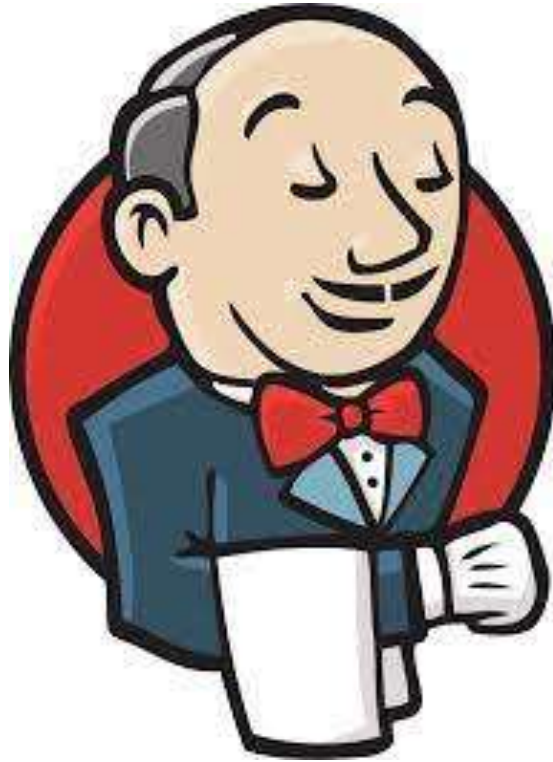


Module 6: Jenkins Assignment



Module 6: Jenkins Assignment - 1

Tasks To Be Performed:

1. Trigger a pipeline using Git when push on develop branch
2. Pipeline should pull Git content to a folder

Launching 3 Ec2 Instance

aws

Search

[Alt+S]

Asia Pacific (Mumbai)

prakash24

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents

My AMIs

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

Debian

debian

Search

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

ami-09b0a86a2c84101e1 (64-bit (x86)) / ami-0a87daabd88e93b1f (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

▼ Summary

Number of instances Info

3

When launching more than 1 instance, [consider EC2 Auto Scaling](#)

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

ami-09b0a86a2c84101e1

Virtual server type (instance type)

t2.micro

Firewall (security group)

default

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, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million

×

Cancel

Launch instance

[Preview code](#)

3 Ec2 Instance one master & 2 slave nodes

aws

Search

[Alt+S]

Asia Pacific (Mumbai)

prakash24

EC2

EC2 Image Builder

S3

IAM

VPC

CloudFormation

Lambda

Cognito

API Gateway

CloudWatch

Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Instances (1/3) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

Instance state = running

Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Av
<input checked="" type="checkbox"/>	jenkins-master	i-0a935b1d36e673239	Running	t2.micro	Initializing	View alarms +	ap
<input type="checkbox"/>	jenkins-slave1	i-02bcee21b2de5562a	Running	t2.micro	Initializing	View alarms +	ap
<input type="checkbox"/>	jenkins-slave2	i-0ee3b1129d630fba2	Running	t2.micro	Initializing	View alarms +	ap

i-0a935b1d36e673239 (jenkins-master)

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

Instance summary Info

Instance ID

i-0a935b1d36e673239

IPv6 address

-

Hostname type

IP name: ip-172-31-3-219.ap-south-1.compute.internal

Answer private resource DNS name

IPv4 (A)

Public IPv4 address

13.126.147.60 | open address

Instance state

Running

Private IP DNS name (IPv4 only)

ip-172-31-3-219.ap-south-1.compute.internal

Instance type

t2.micro

Private IPv4 addresses

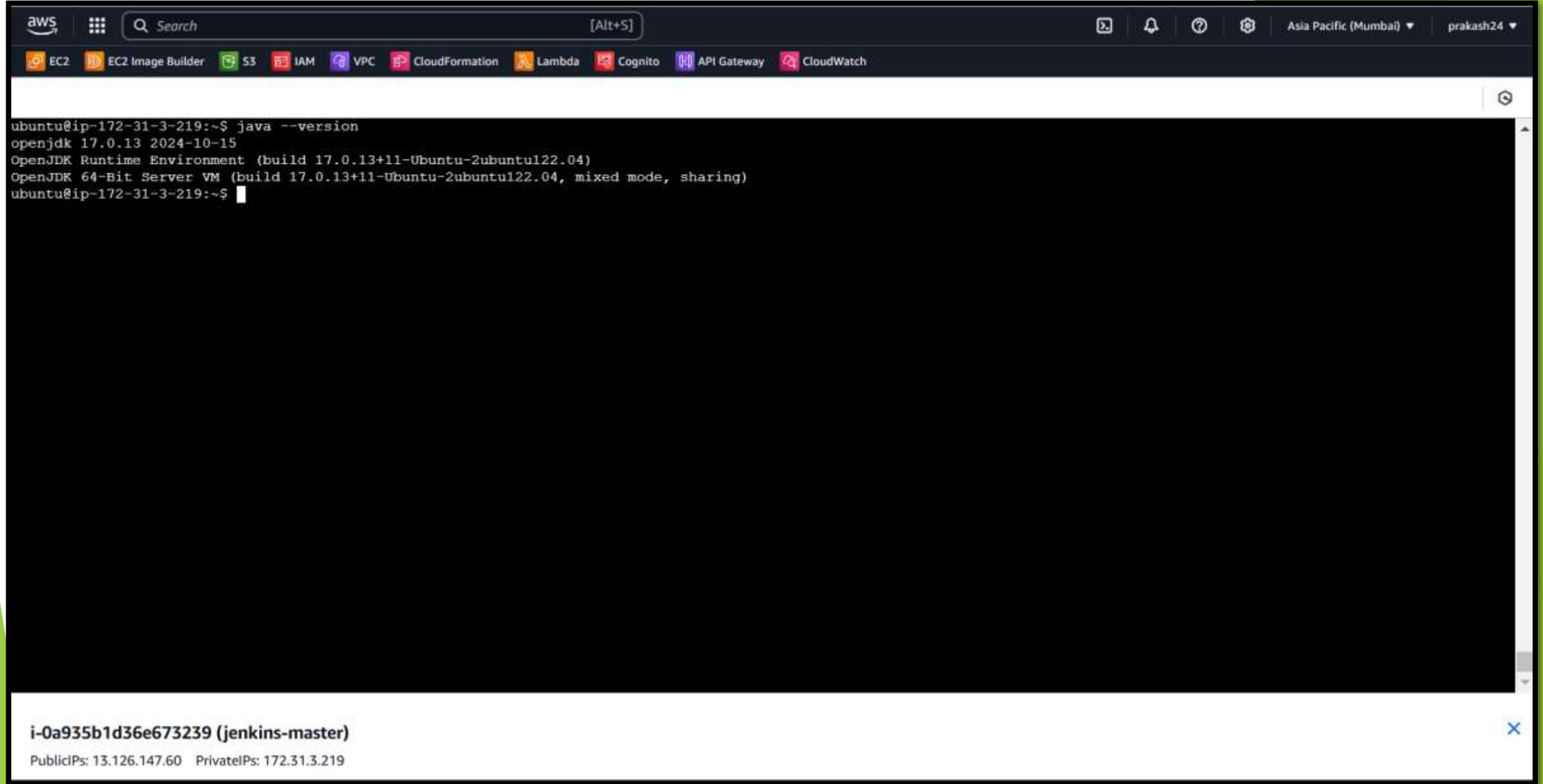
172.31.3.219

Public IPv4 DNS

ec2-13-126-147-60.ap-south-1.compute.amazonaws.com | open address

Elastic IP addresses

Ec2 Instance one master nodes



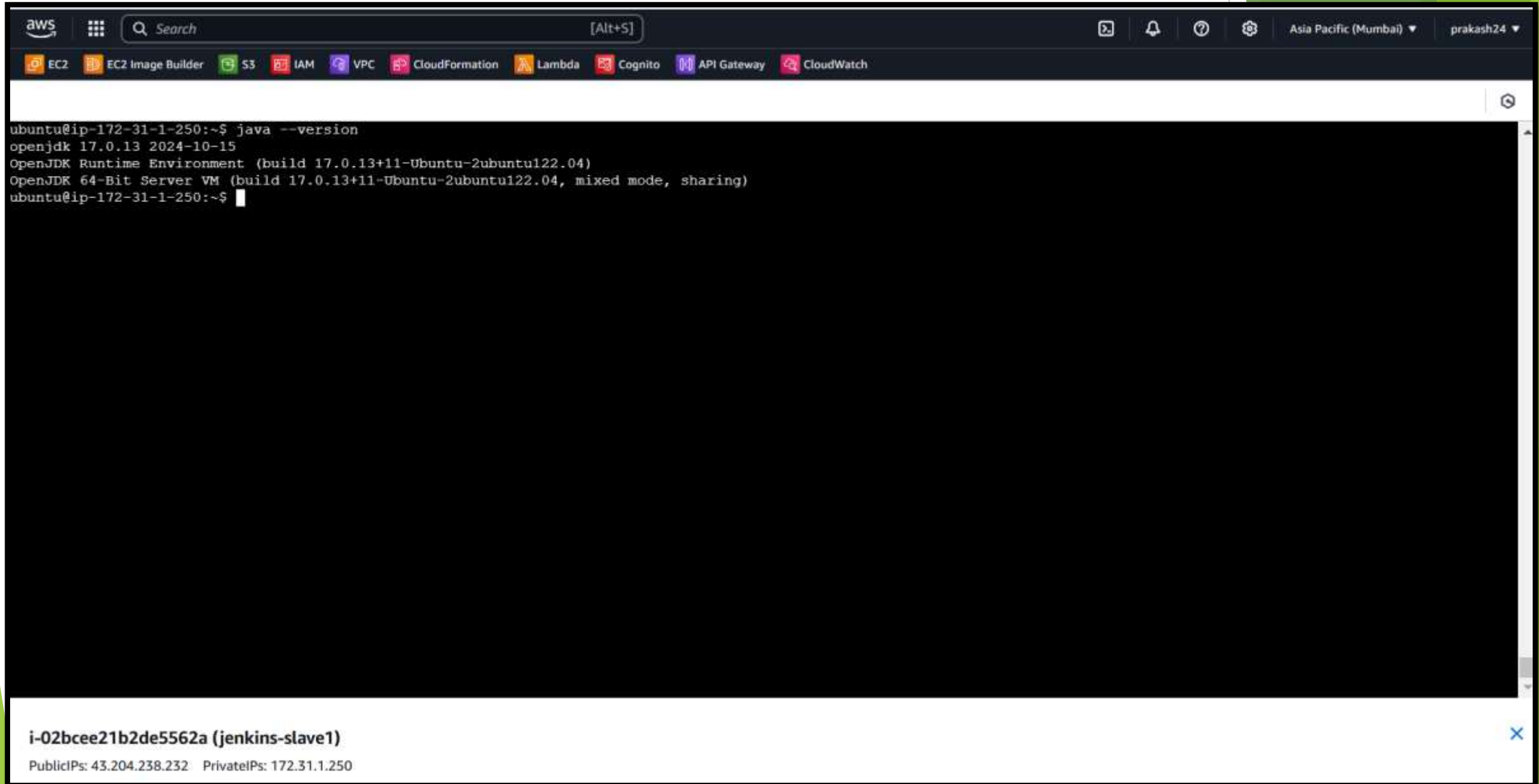
The screenshot shows the AWS Management Console interface with a terminal window open. The terminal displays the output of the command `java --version` executed on an Ubuntu instance. The output indicates that OpenJDK 17.0.13 is installed, with the runtime environment and 64-bit server VM details. The instance ID `i-0a935b1d36e673239` is visible at the bottom, along with its public and private IP addresses.

```
aws
[Alt+S]
EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

ubuntu@ip-172-31-3-219:~$ java --version
openjdk 17.0.13 2024-10-15
OpenJDK Runtime Environment (build 17.0.13+11-Ubuntu-2ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.13+11-Ubuntu-2ubuntu122.04, mixed mode, sharing)
ubuntu@ip-172-31-3-219:~$
```

i-0a935b1d36e673239 (jenkins-master)
PublicIPs: 13.126.147.60 PrivateIPs: 172.31.3.219

Ec2 Instance Jenkins-slave1



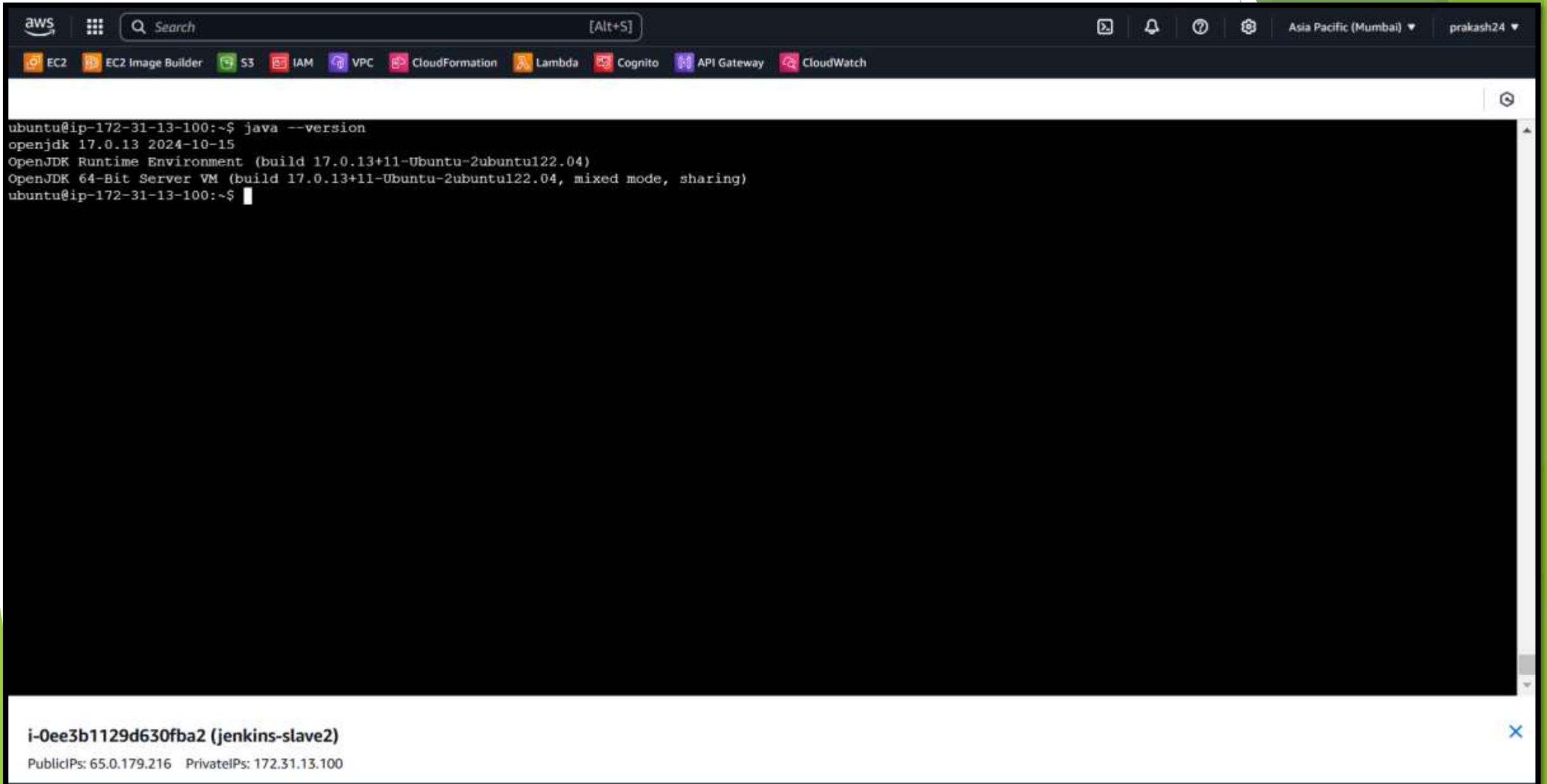
The screenshot displays the AWS Management Console interface. At the top, the navigation bar includes the AWS logo, a search bar, and a list of services: EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The region is set to Asia Pacific (Mumbai) and the user is 'prakash24'.

The main content area shows a terminal window for an Ubuntu instance. The terminal output is as follows:

```
ubuntu@ip-172-31-1-250:~$ java --version
openjdk 17.0.13 2024-10-15
OpenJDK Runtime Environment (build 17.0.13+11-Ubuntu-2ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.13+11-Ubuntu-2ubuntu122.04, mixed mode, sharing)
ubuntu@ip-172-31-1-250:~$
```

At the bottom of the console, the instance details are shown: **i-02bcee21b2de5562a (jenkins-slave1)**. Below the instance name, the public and private IP addresses are listed: Public IPs: 43.204.238.232, Private IPs: 172.31.1.250.

Ec2 Instance Jenkins-slave2



The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and a list of services including EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The region is set to Asia Pacific (Mumbai) and the user is 'prakash24'. Below the navigation bar is a terminal window for an EC2 instance named 'i-0ee3b1129d630fba2 (jenkins-slave2)'. The terminal shows the command 'java --version' being executed, with the following output:

```
ubuntu@ip-172-31-13-100:~$ java --version
openjdk 17.0.13 2024-10-15
OpenJDK Runtime Environment (build 17.0.13+11-Ubuntu-2ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.13+11-Ubuntu-2ubuntu122.04, mixed mode, sharing)
ubuntu@ip-172-31-13-100:~$
```

At the bottom of the terminal window, the instance ID 'i-0ee3b1129d630fba2 (jenkins-slave2)' is displayed, along with its public and private IP addresses: 'PublicIPs: 65.0.179.216 PrivateIPs: 172.31.13.100'.



Search

[Alt+S]



Asia Pacific (Mumbai) ▼

prakash24 ▼

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch



GNU nano 6.2 jenkins.sh

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

[Read 7 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo	M-A Set Mark	M-] To Bracket	M-Q Previous
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^/ Go To Line	M-E Redo	M-6 Copy	^Q Where Was	M-W Next

i-0a935b1d36e673239 (jenkins-master)

PublicIPs: 13.126.147.60 PrivateIPs: 172.31.3.219



Jenkins landing page

← → ↻ ⚠ Not secure 13.126.147.60:8080/login?from=%2F ☆ | 📄 ⬇️ ⓘ

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue

Installing all suggest plugin in Jenkins

← → ↺ ⚠ Not secure 13.126.147.60:8080 ☆ 📁 ⬇️ ⓘ

Getting Started


Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	🔔 Credentials Binding	** Ionicons API
🔔 Timestampers	🔔 Workspace Cleanup	🔔 Ant	🔔 Gradle	Folders
🔔 Pipeline	🔔 GitHub Branch Source	🔔 Pipeline: GitHub Groovy Libraries	🔔 Pipeline Graph View	OWASP Markup Formatter
🔔 Git	🔔 SSH Build Agents	🔔 Matrix Authorization Strategy	🔔 PAM Authentication	** ASM API
🔔 LDAP	🔔 Email Extension	🔔 Mailer	🔔 Dark Theme	** JSON Path API
				** Struts
				** Pipeline: Step API
				** Token Macro
				Build Timeout
				** bouncycastle API
				** Credentials


** - required dependency


Jenkins 2.479.3


Home page of Jenkins


 **Jenkins**

Search (CTRL+K)





 1


 raj


 log out

Dashboard >

 New Item

 Build History

 Manage Jenkins

 My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

Add description

Welcome to Jenkins!

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.

Start building your software project

Create a job

Set up a distributed build


Set up an agent

Configure a cloud

Learn more about distributed builds

Setting up Prod node in Jenkins

← → ↻ ⚠ Not secure 13.126.147.60:8080/computer/new ☆ 📁 ⬇️ 📄 ⋮

 **Jenkins** 🔍 Search (CTRL+K) ? 🛡️ 1 👤 raj ⌵ 🚪 log out

Dashboard > Nodes > New node

New node

Node name

Prod

Type

☒ Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create

REST API Jenkins 2.479.3

Setting up Remote root directory for Prod node in Jenkins

← → ↻ ⚠ Not secure 13.126.147.60:8080/computer/createItem ☆ 📄 ⬇️ p ⋮

Dashboard > Nodes >

Number of executors ?

1

Remote root directory ?

/home/ubuntu/jenkins/

Labels ?

Usage ?

Use this node as much as possible ▼

Launch method ?

Launch agent by connecting it to the controller ▼

Availability ?

Keep this agent online as much as possible ▼ ?

Save

Connecting Prod node with Slave1 Ec2 instance in Jenkins

Dashboard > Nodes >

Launch method ?

Launch agents via SSH

Host ?

172.31.1.250

Credentials ?

ubuntu

+ Add

Host Key Verification Strategy ?


Non verifying Verification Strategy

Advanced





Availability ?

Save


Prod node connected and online


 **Jenkins**

Search (CTRL+K)

   raj  log out

Dashboard > Nodes >


 Nodes


 Clouds

Build Queue

No builds in the queue.

Build Executor Status


 Built-In Node 0/2







 Prod 0/1

Nodes

+ New Node

Configure Monitors



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.72 GiB	 0 B	4.72 GiB	0ms 
	Prod	Linux (amd64)	In sync	5.20 GiB	 0 B	5.20 GiB	58ms 
Data obtained		33 sec	33 sec	33 sec	33 sec	33 sec	33 sec

Icon:

S

M

L

Legend


REST API

Jenkins 2.479.3

Setting up a job job-assignment-1

The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is `13.126.147.60:8080/job/job-assignment-1/configure`. The Jenkins logo and name are in the top left, and a search bar is in the top right. The breadcrumb navigation shows `Dashboard > job-assignment-1 > Configuration`. On the left, the 'Configure' section is active, with a sidebar listing options: General (selected), Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main area is titled 'General' and shows a toggle switch for 'Enabled' which is turned on. Below this is a 'Description' text area. At the bottom, there are four unchecked checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', and 'Throttle builds'. 'Save' and 'Apply' buttons are at the very bottom.


← → ↻ ⚠ Not secure 13.126.147.60:8080/job/job-assignment-1/configure ☆ | 📁 | ⬇️ | 📄 | ⋮







 **Jenkins** 🔍 Search (CTRL+K) ? 🛡️ 1 👤 raj ▾ 🚪 log out

Dashboard > job-assignment-1 > Configuration

Configure

General



Enabled 

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

Description

Plain text [Preview](#)

- ☐ Discard old builds ?
- ☐ GitHub project
- ☐ This project is parameterized ?
- ☐ Throttle builds ?

Restricting job-assignment-1 to Run Prod Node

← → ↻ ⚠ Not secure 13.126.147.60:8080/job/job-assignment-1/configure ☆ 📁 ⬇️ 👤 ⋮

Dashboard > job-assignment-1 > Configuration

Configure

- ⚙ General
- 🔗 Source Code Management
- 🕒 Build Triggers
- 🌐 Build Environment
- 📋 Build Steps
- 📦 Post-build Actions

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

☒ Restrict where this project can be run ?

Label Expression ?

Prod

Label **Prod** matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

Save

Apply

Connecting GitHub repo to job-assignment-1 to Run Prod Node

github.com/ratreraj/Proj1

ratreraj / Proj1

Type / to search

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

Proj1Public

PinUnwatch1Fork0Star0

develop2 Branches0 Tags

Go to file

Add file

Code

About

for Jenkins assignment 1

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Ubuntu dev commit

da4fc81 · 2 months ago

2 Commits

developfile

dev commit

2 months ago

masterfile

this is my first commit

2 months ago

README

Add a README

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

Add a README

Configuring GitHub repo & branch for job-assignment-1

←

→

↻

⚠ Not secure

13.126.147.60:8080/job/job-assignment-1/configure

🔍

☆

📄

⬇

Ⓜ

⋮

Dashboard > job-assignment-1 > Configuration

Configure

⚙ General

🔑 Source Code Management

🕒 Build Triggers

🌐 Build Environment

☰ Build Steps

📦 Post-build Actions

Repository URL ?

https://github.com/ratreraj/Proj1.git

Credentials ?

- none -

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/develop

Add Branch

Repository browser ?

Save

Apply

Setting up build triggers on git changes for job-assignment-1

The screenshot shows the Jenkins web interface for configuring a job named 'job-assignment-1'. The browser address bar indicates the URL is '13.126.147.60:8080/job/job-assignment-1/configure'. The breadcrumb navigation shows 'Dashboard > job-assignment-1 > Configuration'. On the left sidebar, the 'Build Triggers' option is selected and highlighted. The main content area is titled 'Configure' and contains two sections: 'Build Triggers' and 'Build Environment'. In the 'Build Triggers' section, the checkbox for 'GitHub hook trigger for GITScm polling' is checked, while the others are unchecked. The 'Build Environment' section contains several unchecked checkboxes. At the bottom, there are 'Save' and 'Apply' buttons.

Dashboard > job-assignment-1 > Configuration

Configure

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

Build Triggers

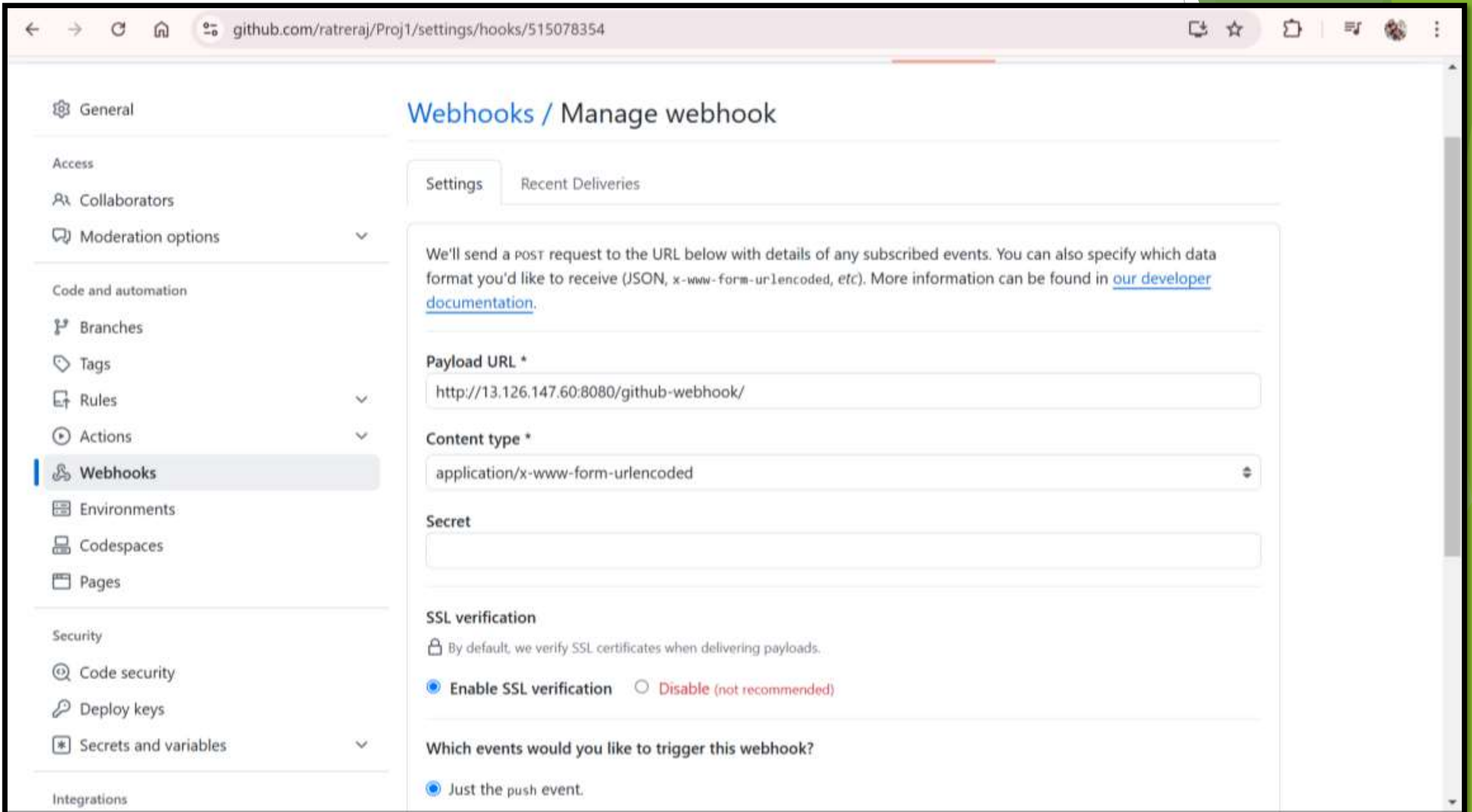
- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

Build Environment

- ☐ Delete workspace before build starts
- ☐ 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
- ☐ With Ant ?

Save Apply

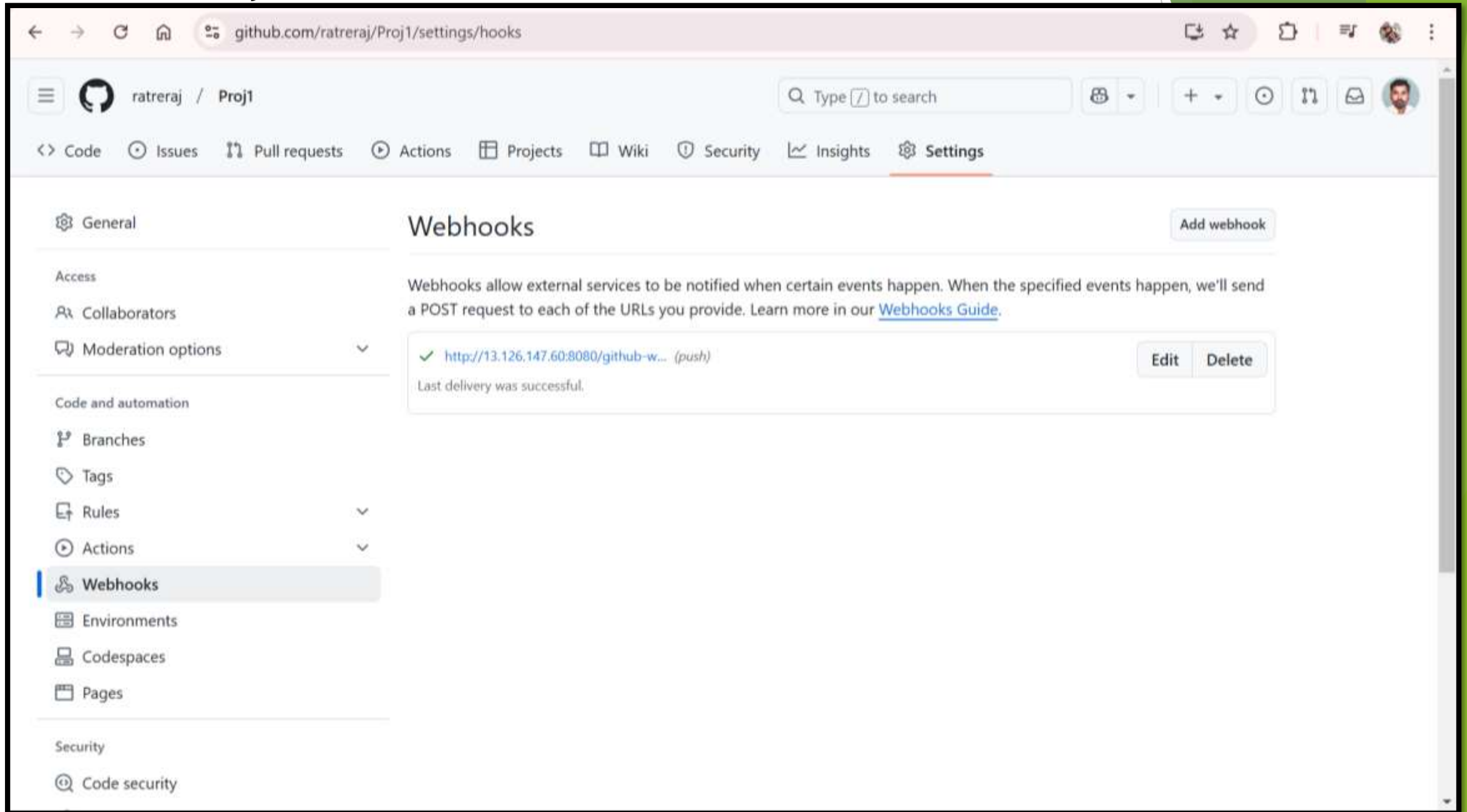
Setting a webhooks github



The screenshot shows the GitHub webhooks settings page for a repository. The browser address bar displays `github.com/ratreraj/Proj1/settings/hooks/515078354`. On the left sidebar, the 'Webhooks' option is selected under the 'Code and automation' section. The main content area is titled 'Webhooks / Manage webhook' and has two tabs: 'Settings' (active) and 'Recent Deliveries'. The 'Settings' tab contains the following configuration options:

- General:** A text block explaining that a POST request will be sent to the specified URL with details of subscribed events. It also mentions the ability to specify a data format (JSON, x-www-form-urlencoded, etc.) and provides a link to the [developer documentation](#).
- Payload URL *:** A text input field containing `http://13.126.147.60:8080/github-webhook/`.
- Content type *:** A dropdown menu set to `application/x-www-form-urlencoded`.
- Secret:** An empty text input field for a secret key.
- SSL verification:** A section with a lock icon and the text 'By default, we verify SSL certificates when delivering payloads.' Below this are two radio buttons: **Enable SSL verification** (selected) and **Disable (not recommended)**.
- Which events would you like to trigger this webhook?:** A section with a single radio button option: **Just the push event.**

Webhook delivery is successful



The screenshot shows the GitHub web interface for a repository named 'Proj1' by user 'ratreraj'. The 'Settings' tab is selected, and the 'Webhooks' section is active in the left sidebar. The main content area displays the 'Webhooks' configuration. A single webhook is listed with a green checkmark icon, indicating a successful delivery. The URL is 'http://13.126.147.60:8080/github-w...' and the event is '(push)'. Below the URL, it states 'Last delivery was successful.' To the right of the webhook entry are 'Edit' and 'Delete' buttons. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The left sidebar lists various repository settings like General, Access, Collaborators, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Webhooks (highlighted), Environments, Codespaces, Pages, Security, and Code security.

github.com/ratreraj/Proj1/settings/hooks

ratreraj / Proj1

Type [7] to search

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Webhooks

Add webhook

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security

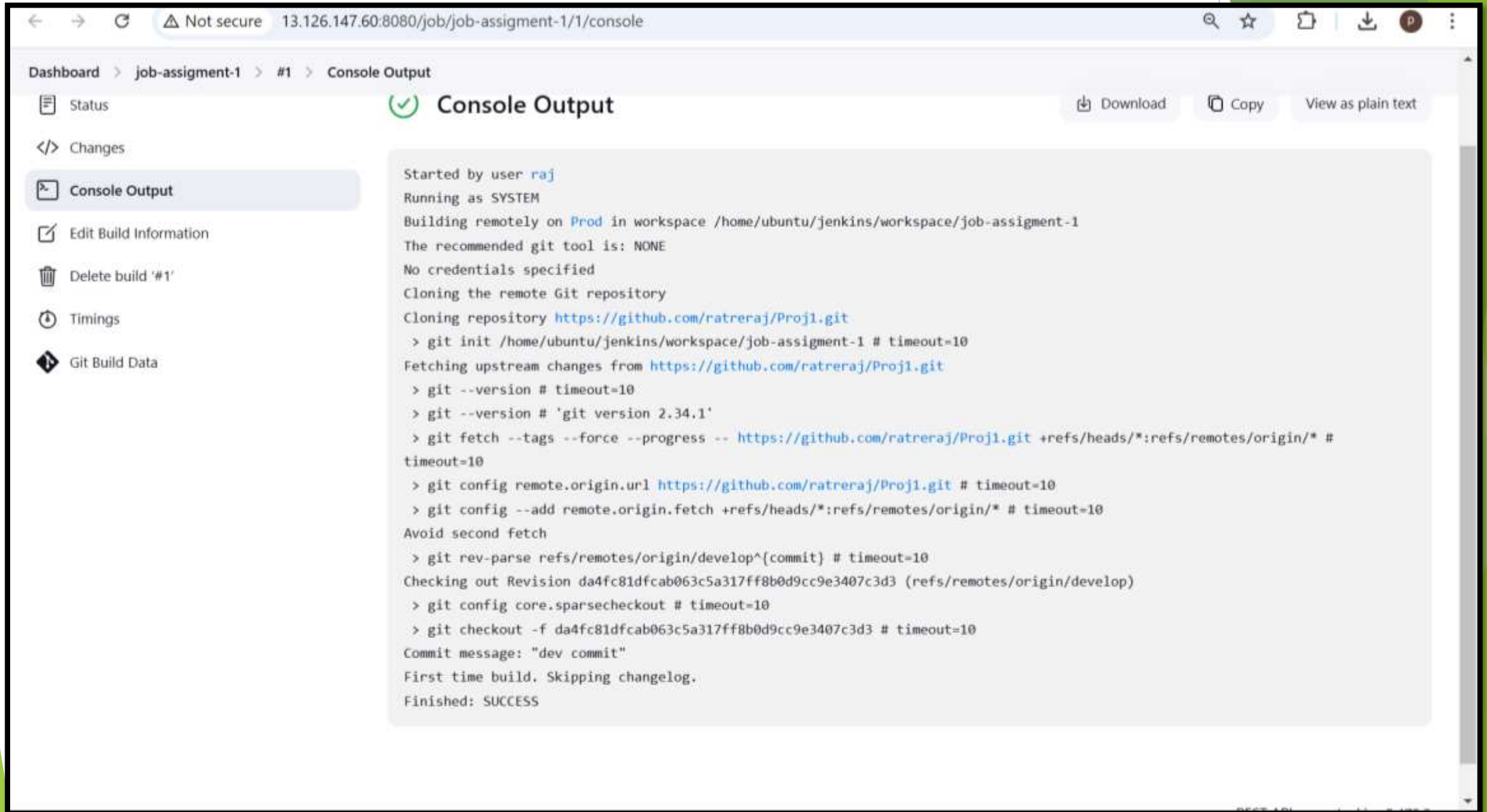
Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

✓ http://13.126.147.60:8080/github-w... (push)

Last delivery was successful.

Edit Delete

Run the job-assignment-1 we can see the console output successful



The screenshot shows the Jenkins web interface for a job named 'job-assignment-1'. The left sidebar contains navigation links: Status, Changes, Console Output (selected), Edit Build Information, Delete build '#1', Timings, and Git Build Data. The main area displays the 'Console Output' for build #1, which is successful. The output text shows the build process starting with user 'raj', running as 'SYSTEM', and cloning a repository from 'https://github.com/ratreraaj/Proj1.git'. It includes several git commands and their outputs, such as 'git init', 'git fetch', 'git config', and 'git checkout'. The build concludes with the message 'Finished: SUCCESS'.

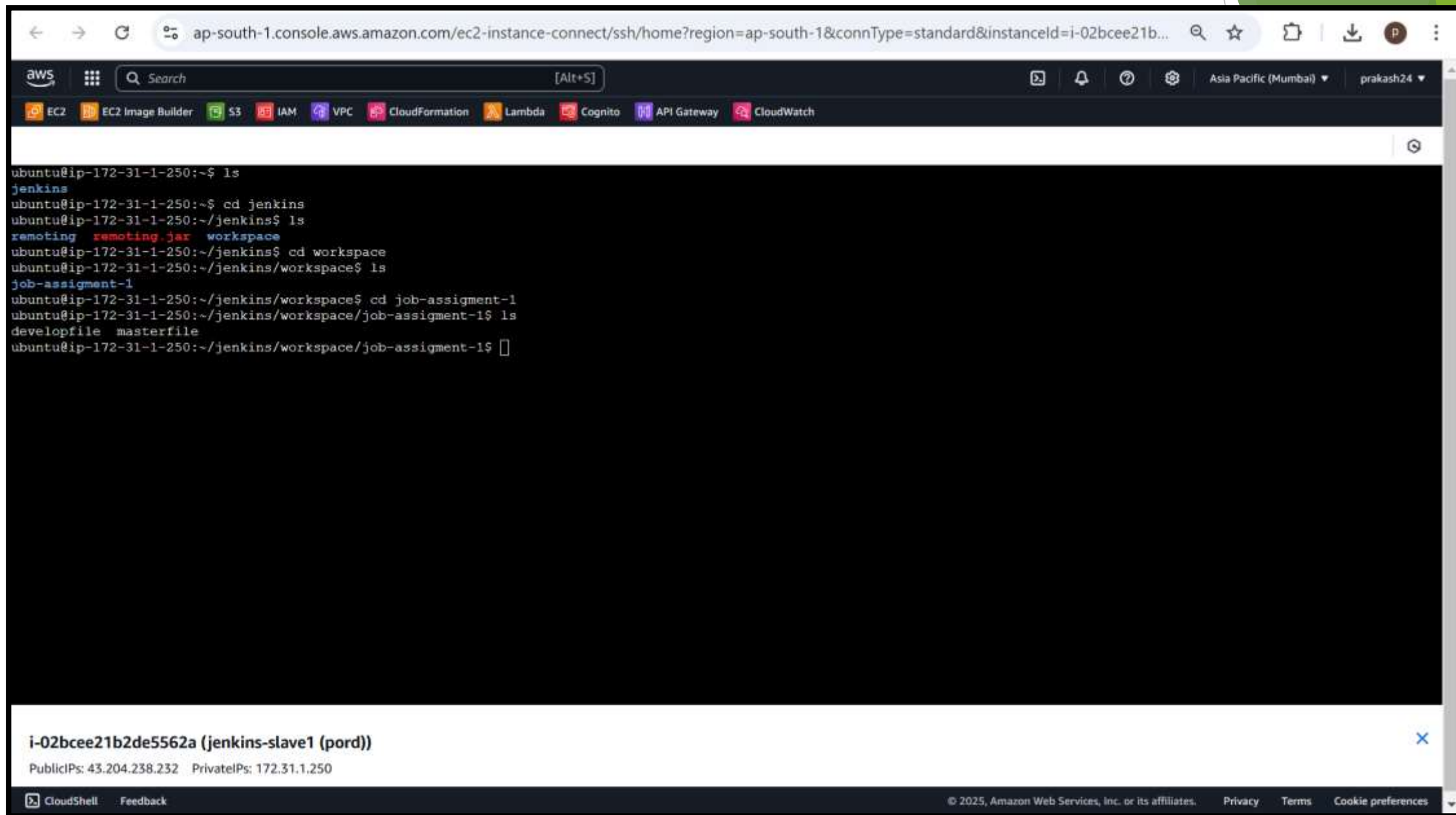
Dashboard > job-assignment-1 > #1 > Console Output

Console Output

Download Copy View as plain text

```
Started by user raj
Running as SYSTEM
Building remotely on Prod in workspace /home/ubuntu/jenkins/workspace/job-assignment-1
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/ratreraaj/Proj1.git
> git init /home/ubuntu/jenkins/workspace/job-assignment-1 # timeout=10
Fetching upstream changes from https://github.com/ratreraaj/Proj1.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/ratreraaj/Proj1.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git config remote.origin.url https://github.com/ratreraaj/Proj1.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/develop^{commit} # timeout=10
Checking out Revision da4fc81dfcab063c5a317ff8b0d9cc9e3407c3d3 (refs/remotes/origin/develop)
> git config core.sparsecheckout # timeout=10
> git checkout -f da4fc81dfcab063c5a317ff8b0d9cc9e3407c3d3 # timeout=10
Commit message: "dev commit"
First time build. Skipping changelog.
Finished: SUCCESS
```

We can see the files present in workspace folder

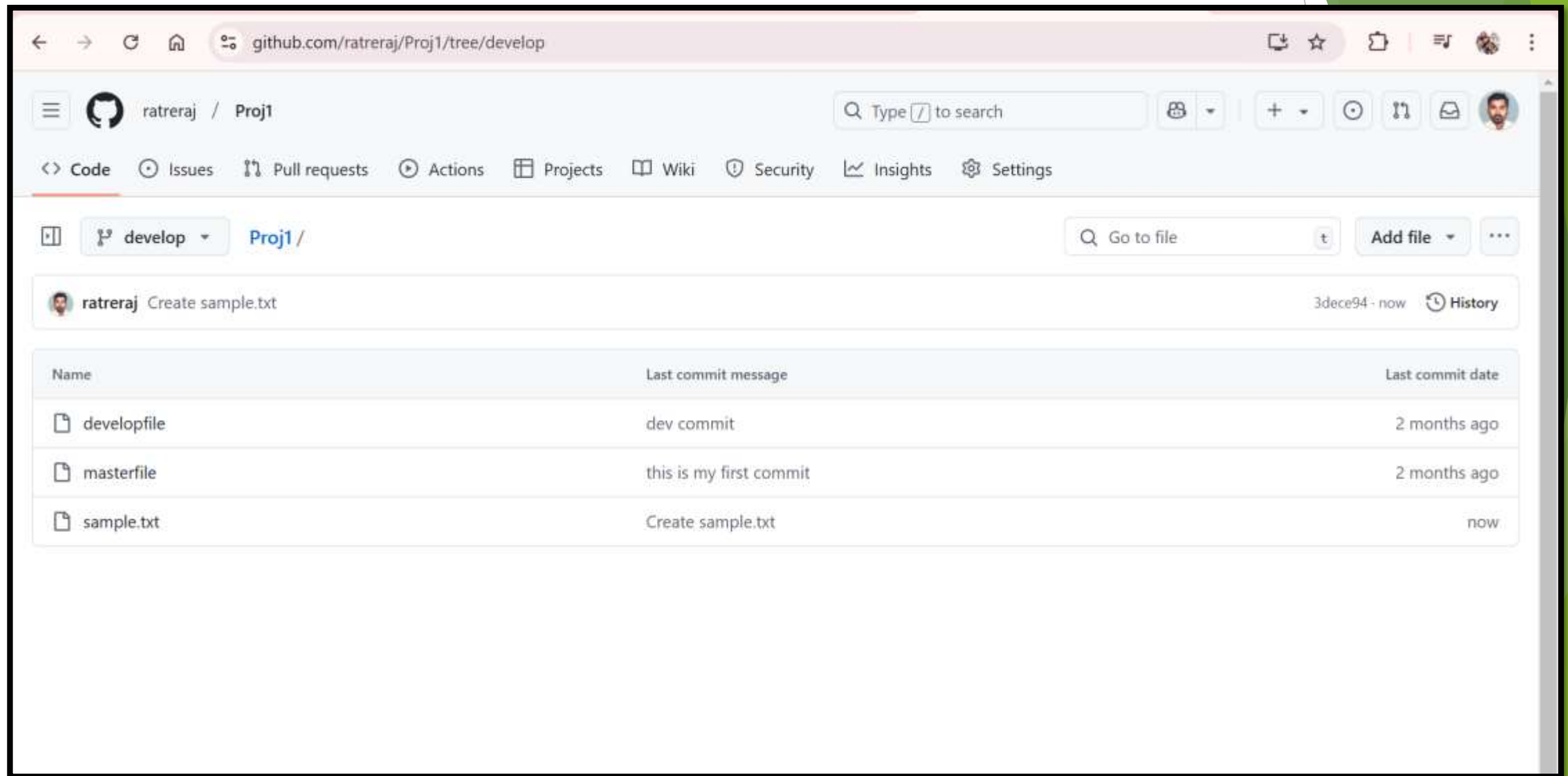


The screenshot shows an AWS CloudShell terminal window. The browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-02bcee21b...`. The AWS console header is visible with the search bar and navigation links for EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user is logged in as 'prakash24' in the 'Asia Pacific (Mumbai)' region.

```
ubuntu@ip-172-31-1-250:~$ ls
jenkins
ubuntu@ip-172-31-1-250:~$ cd jenkins
ubuntu@ip-172-31-1-250:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-1-250:~/jenkins$ cd workspace
ubuntu@ip-172-31-1-250:~/jenkins/workspace$ ls
job-assignment-1
ubuntu@ip-172-31-1-250:~/jenkins/workspace$ cd job-assignment-1
ubuntu@ip-172-31-1-250:~/jenkins/workspace/job-assignment-1$ ls
developfile  masterfile
ubuntu@ip-172-31-1-250:~/jenkins/workspace/job-assignment-1$
```

At the bottom of the terminal window, the instance information is displayed: `i-02bcee21b2de5562a (jenkins-slave1 (pord))`. Below this, the public and private IP addresses are listed: `PublicIPs: 43.204.238.232 PrivateIPs: 172.31.1.250`. The footer of the terminal shows the 'CloudShell' logo, a 'Feedback' link, and the copyright notice: '© 2025, Amazon Web Services, Inc. or its affiliates.' along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

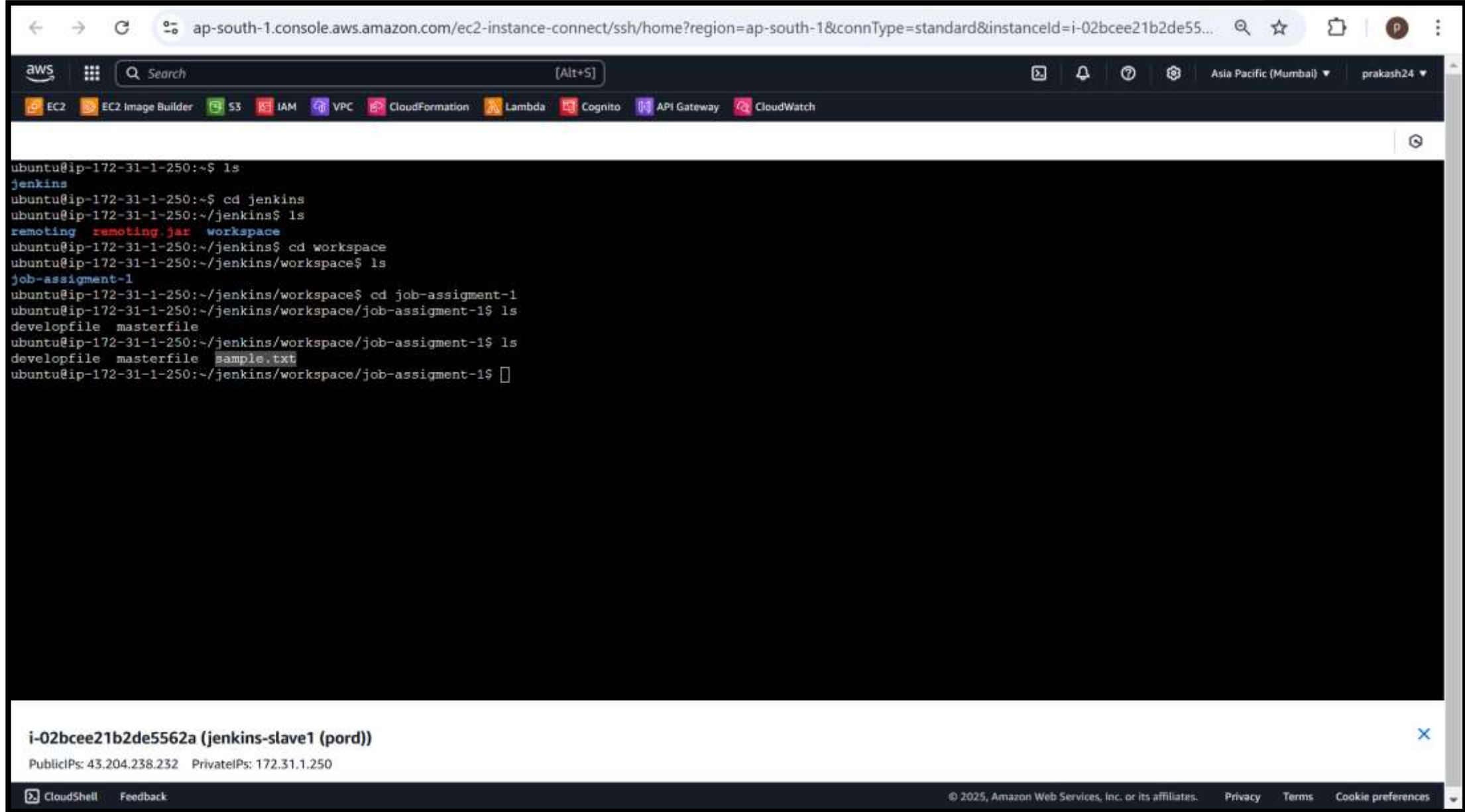
Added new file sample.txt in develop branch



The screenshot shows the GitHub interface for the repository 'ratreraj / Proj1'. The 'develop' branch is selected. The commit history table is displayed, showing the following commits:

Name	Last commit message	Last commit date
developfile	dev commit	2 months ago
masterfile	this is my first commit	2 months ago
sample.txt	Create sample.txt	now

We can see the sample.txt file in workspace after job run



The screenshot shows the AWS CloudShell interface. At the top, the browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-02bcee21b2de55...`. Below the address bar is the AWS navigation bar with a search bar and various service icons (EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, CloudWatch). The main area is a terminal window with a black background and white text. The terminal shows the following commands and output:

```
ubuntu@ip-172-31-1-250:~$ ls
jenkins
ubuntu@ip-172-31-1-250:~$ cd jenkins
ubuntu@ip-172-31-1-250:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-1-250:~/jenkins$ cd workspace
ubuntu@ip-172-31-1-250:~/jenkins/workspace$ ls
job-assignment-1
ubuntu@ip-172-31-1-250:~/jenkins/workspace$ cd job-assignment-1
ubuntu@ip-172-31-1-250:~/jenkins/workspace/job-assignment-1$ ls
developfile  masterfile
ubuntu@ip-172-31-1-250:~/jenkins/workspace/job-assignment-1$ ls
developfile  masterfile  sample.txt
ubuntu@ip-172-31-1-250:~/jenkins/workspace/job-assignment-1$
```

At the bottom of the terminal window, a status bar shows the instance ID `i-02bcee21b2de5562a (jenkins-slave1 (pord))` and its IP addresses: `PublicIPs: 43.204.238.232 PrivateIPs: 172.31.1.250`. The footer of the CloudShell interface includes the `CloudShell` logo, a `Feedback` link, and copyright information: `© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences`.

Module 6: Jenkins Assignment - 2

Tasks To Be Performed:

1. Add 2 nodes to Jenkins master
2. Create 2 jobs with the following jobs:
 - a. Push to test
 - b. Push to prod
3. Once a push is made to test branch, copy Git files to test server
4. Once a push is made to master branch, copy Git files to prod server

3 Ec2 Instance one master & 2 slave nodes

The screenshot displays the AWS Management Console for the 'ap-south-1' region. The 'Instances' page shows three running EC2 instances: 'my-jenkins-master', 'my-jenkins-node-pord', and 'my-jenkins-node-test'. All instances are of type 't2.micro' and have a status of 'Running'. The console also shows a sidebar with navigation options like Dashboard, EC2 Global View, Events, and various services like IAM, VPC, and CloudFormation.

Instances (3) [Info](#) Last updated 14 minutes ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[All states](#)

[Instance state = running](#) [Clear filters](#)

<input type="checkbox"/>	Name ✎	Instance ID	Instance state	Instance type	Status check	Alarm status	Av
<input type="checkbox"/>	my-jenkins-master	i-0fccfceaba1d1b4a	Running	t2.micro	2/2 checks passed	View alarms +	ap
<input type="checkbox"/>	my-jenkins-node-pord	i-0d757eb8e00b231a3	Running	t2.micro	2/2 checks passed	View alarms +	ap
<input type="checkbox"/>	my-jenkins-node-test	i-0b789f5f693e55b01	Running	t2.micro	2/2 checks passed	View alarms +	ap

Select an instance

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Configured two worker node Prod & Test

←

→

↺


Not secure 13.233.22.78:8080/computer/

☆

📁

👤

⋮

 **Jenkins**

🔍 Search (CTRL+K)

🛡️ 1

👤 raj

🚪 log out

Dashboard > Nodes >

🖥️ Nodes

☁️ Clouds

Build Queue

No builds in the queue.

Build Executor Status

🖥️ Built-In Node0/2

🖥️ Prod0/1

🖥️ Test0/1

Nodes

+ New Node

Configure Monitors

🔄

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
🖥️	Built-In Node	Linux (amd64)	In sync	4.72 GiB	❗ 0 B	4.72 GiB	0ms ⚙️
🖥️	Prod	Linux (amd64)	In sync	5.20 GiB	❗ 0 B	5.20 GiB	4ms ⚙️
🖥️	Test	Linux (amd64)	In sync	5.21 GiB	❗ 0 B	5.21 GiB	24ms ⚙️
Data obtained		58 ms	58 ms	52 ms	54 ms	52 ms	53 ms

Icon:

S

M

L

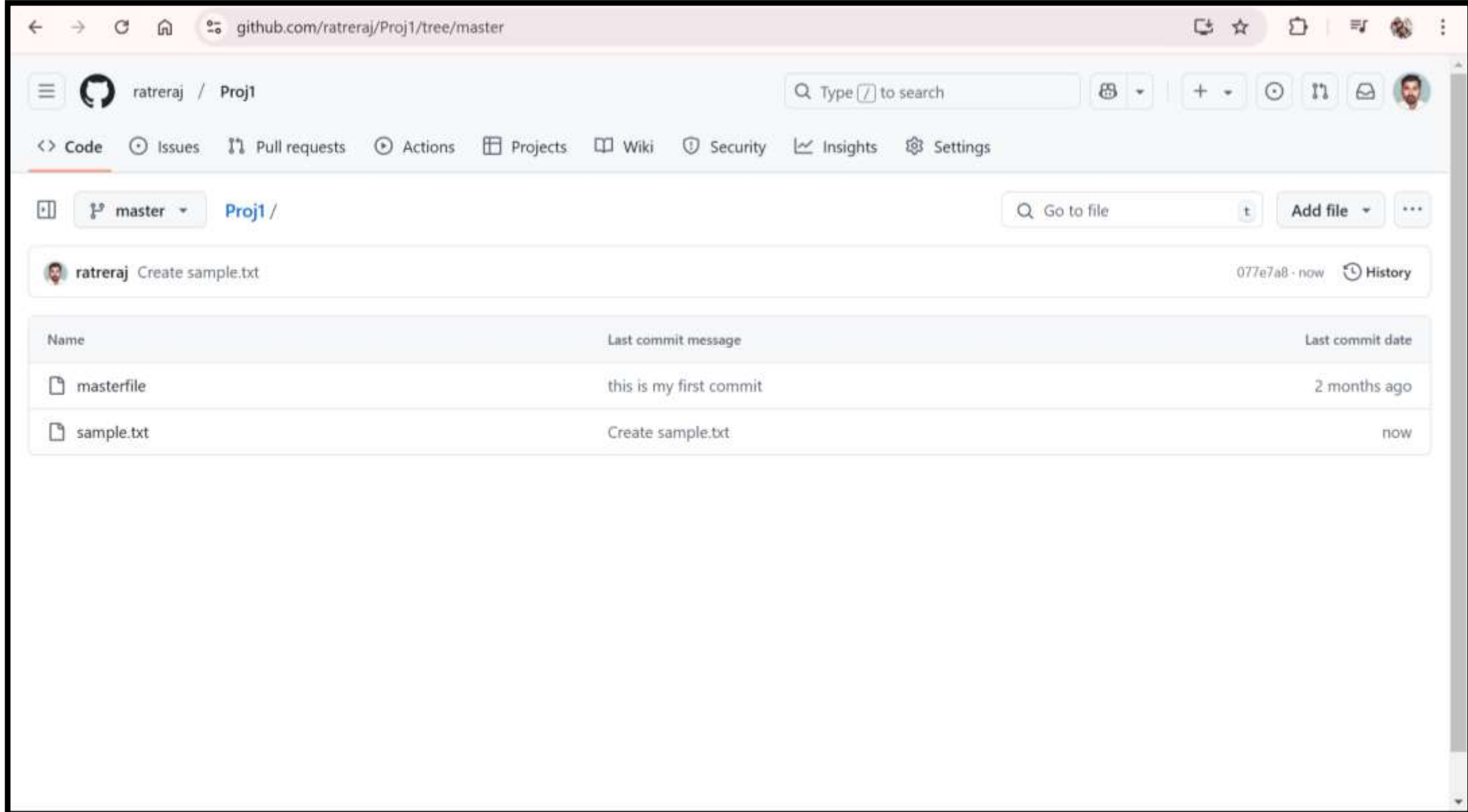
Legend

13.233.22.78:8080

REST API

Jenkins 2.479.3

Master branch for Prod Node & Prod Pipeline



The screenshot shows the GitHub interface for the repository 'ratreraj / Proj1'. The browser address bar indicates the URL 'github.com/ratreraj/Proj1/tree/master'. The repository name 'ratreraj / Proj1' is displayed at the top, along with a search bar and navigation icons. Below the repository name, a navigation bar includes links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected, and the 'master' branch is chosen from a dropdown menu. A 'Go to file' search bar and an 'Add file' button are also visible. The commit history is shown as a table with the following data:

Name	Last commit message	Last commit date
masterfile	this is my first commit	2 months ago
sample.txt	Create sample.txt	now

Test branch for Test Node & Test Pipeline

github.com/ratreraj/Proj1/tree/test

ratreraj / Proj1

Type / to search

+

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

test ▾

Proj1 /

Go to file t

Add file ▾

⋮

ratreraj Create sample2.txt

74286f9 · now History

This branch is 1 commit ahead of, 1 commit behind master .

Contribute ▾

Name	Last commit message	Last commit date
masterfile	this is my first commit	2 months ago
sample2.txt	Create sample2.txt	now

Configured two pipeline Push to prod & Push to test

The screenshot shows the Jenkins Dashboard interface. The browser address bar indicates a 'Not secure' connection to '13.233.22.78:8080'. The dashboard includes a sidebar with navigation links: 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main content area displays a table of pipeline builds. The table has columns for status (S), warnings (W), name, last success, last failure, and last duration. Two pipelines are listed: 'Push to prod' and 'Push to test'. Both show a successful status (green checkmark) and a warning icon (yellow sun). The 'Push to prod' build took 1 min 11 sec and is build #2. The 'Push to test' build took 21 sec and is build #3. Below the table, there is a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section showing the status of three executors: 'Built-In Node' (0/2), 'Prod' (0/1), and 'Test' (0/1). The footer of the dashboard shows 'REST API' and 'Jenkins 2.479.3'.

Dashboard >

+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Add description

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	Push to prod	1 min 11 sec #2	N/A	0.91 sec
✓	☀	Push to test	21 sec #3	N/A	7.4 sec

Build Queue

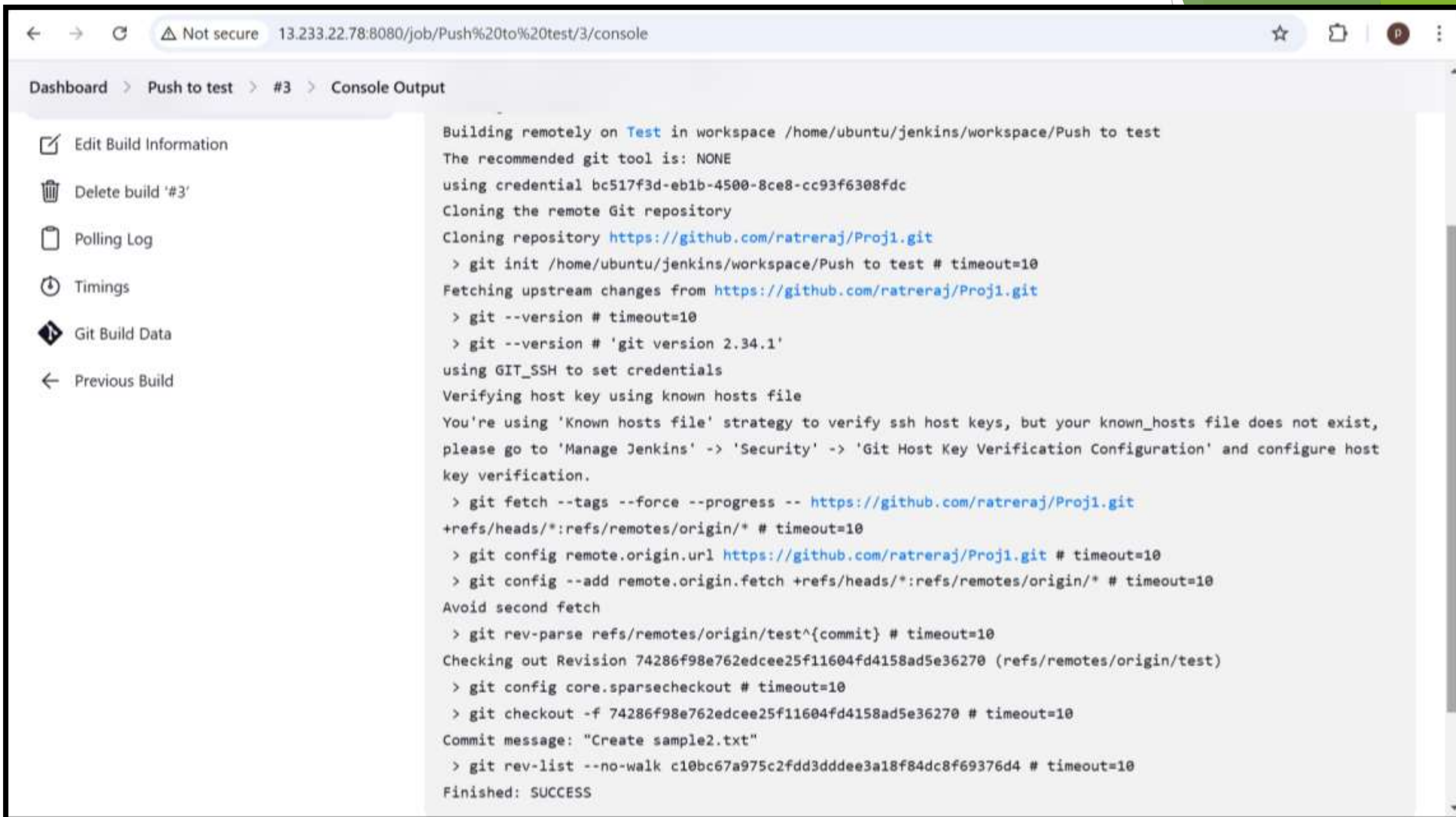
No builds in the queue.

Build Executor Status

- Built-In Node 0/2
- Prod 0/1
- Test 0/1

REST API Jenkins 2.479.3

Push to test pipeline successfully run

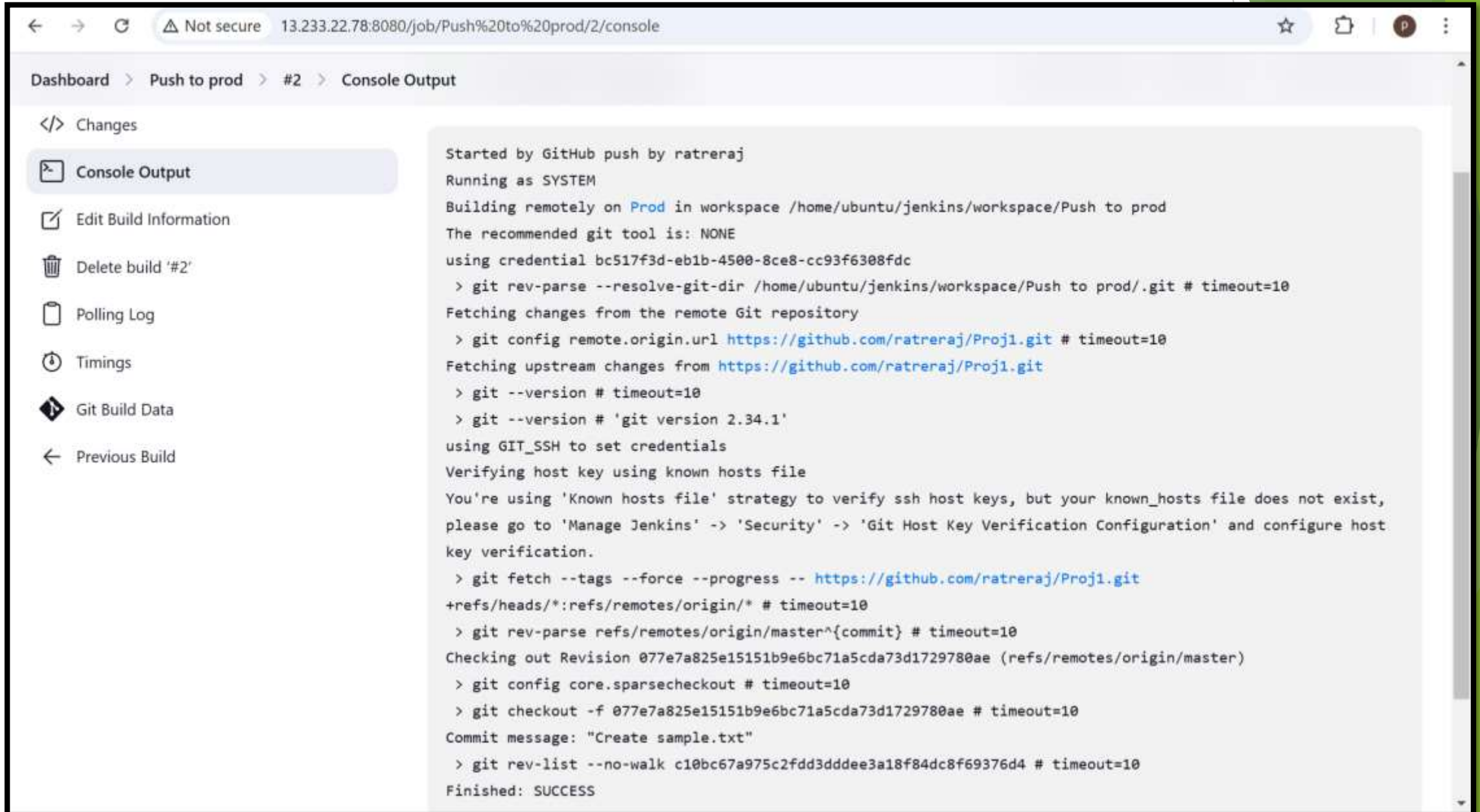


The screenshot shows the Jenkins web interface for a build named 'Push to test' (build #3). The left sidebar contains navigation links: 'Edit Build Information', 'Delete build '#3'', 'Polling Log', 'Timings', 'Git Build Data', and 'Previous Build'. The main area displays the console output of the build, which details the process of cloning a repository, setting up git, and checking out a specific revision.

Dashboard > Push to test > #3 > Console Output

```
Building remotely on Test in workspace /home/ubuntu/jenkins/workspace/Push to test
The recommended git tool is: NONE
using credential bc517f3d-eb1b-4500-8ce8-cc93f6308fdc
Cloning the remote Git repository
Cloning repository https://github.com/ratreraaj/Proj1.git
> git init /home/ubuntu/jenkins/workspace/Push to test # timeout=10
Fetching upstream changes from https://github.com/ratreraaj/Proj1.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_SSH to set credentials
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist,
please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host
key verification.
> git fetch --tags --force --progress -- https://github.com/ratreraaj/Proj1.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/ratreraaj/Proj1.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/test^{commit} # timeout=10
Checking out Revision 74286f98e762edcee25f11604fd4158ad5e36270 (refs/remotes/origin/test)
> git config core.sparsecheckout # timeout=10
> git checkout -f 74286f98e762edcee25f11604fd4158ad5e36270 # timeout=10
Commit message: "Create sample2.txt"
> git rev-list --no-walk c10bc67a975c2fdd3dddee3a18f84dc8f69376d4 # timeout=10
Finished: SUCCESS
```

Push to prod pipeline successfully run



The screenshot shows the Jenkins web interface for a pipeline named 'Push to prod', build #2. The left sidebar contains navigation links: 'Changes', 'Console Output' (selected), 'Edit Build Information', 'Delete build '#2'', 'Polling Log', 'Timings', 'Git Build Data', and 'Previous Build'. The main area displays the console output of the build, which shows a successful deployment to production.

Dashboard > Push to prod > #2 > Console Output

</> Changes

Console Output

Edit Build Information

Delete build '#2'

Polling Log

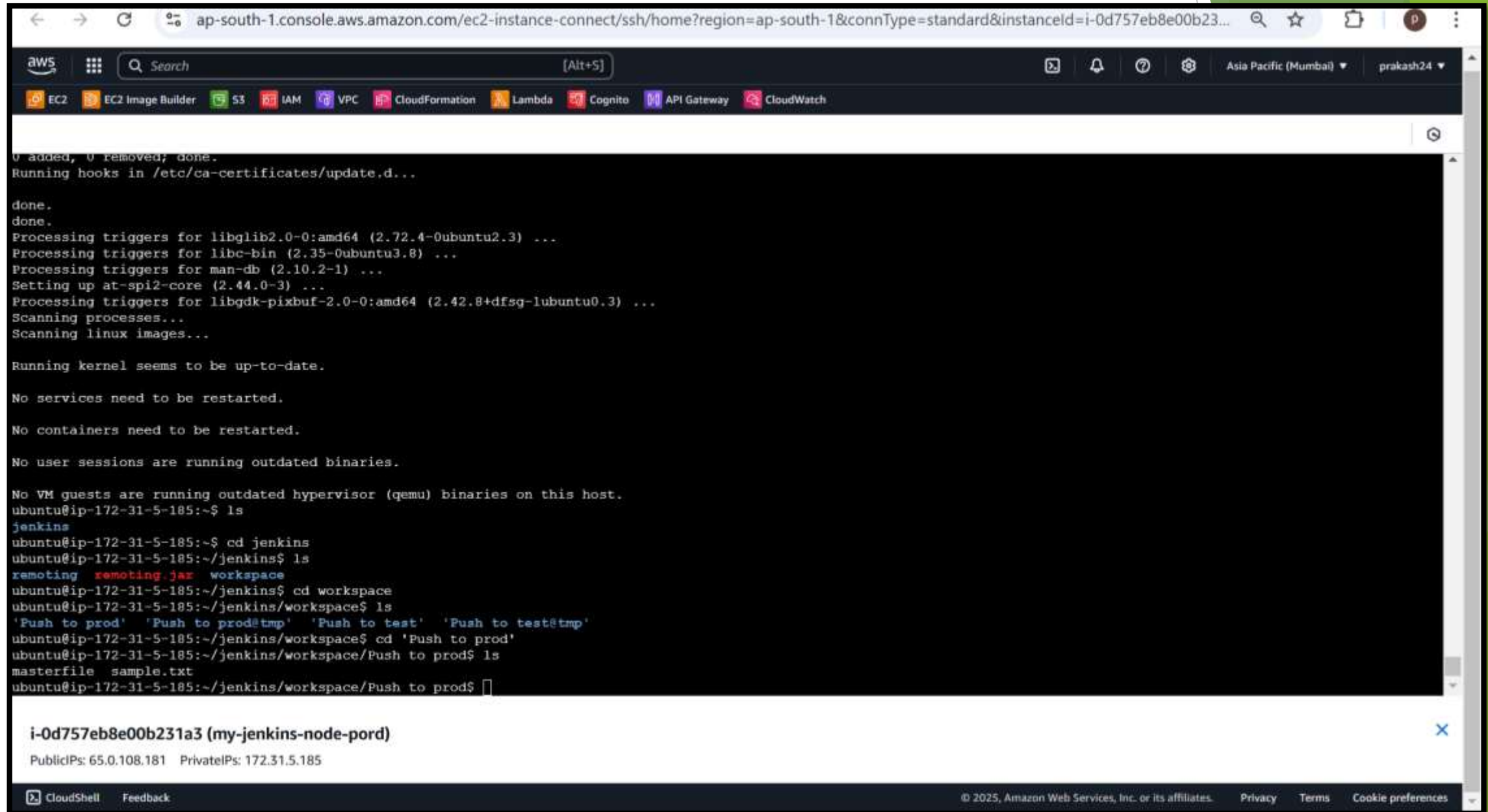
Timings

Git Build Data

Previous Build

```
Started by GitHub push by ratreraj
Running as SYSTEM
Building remotely on Prod in workspace /home/ubuntu/jenkins/workspace/Push to prod
The recommended git tool is: NONE
using credential bc517f3d-eb1b-4500-8ce8-cc93f6308fdc
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/Push to prod/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ratreraj/Proj1.git # timeout=10
Fetching upstream changes from https://github.com/ratreraj/Proj1.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_SSH to set credentials
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist,
please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host
key verification.
> git fetch --tags --force --progress -- https://github.com/ratreraj/Proj1.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 077e7a825e15151b9e6bc71a5cda73d1729780ae (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 077e7a825e15151b9e6bc71a5cda73d1729780ae # timeout=10
Commit message: "Create sample.txt"
> git rev-list --no-walk c10bc67a975c2fdd3ddddee3a18f84dc8f69376d4 # timeout=10
Finished: SUCCESS
```


We can see the files pull from master branch in workspace folder



The screenshot shows the AWS Management Console interface for an EC2 instance. The terminal window displays the output of a Jenkins installation script. The script performs various checks and configurations, including processing triggers for different packages, setting up at-spi2-core, and scanning processes. It also checks the kernel, services, containers, and user sessions. Finally, it lists the contents of the workspace folder, showing files like 'remoting.jar' and 'workspace'. The terminal output is as follows:

```
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
done.
Processing triggers for libgl1-mesa-dev:amd64 (2.72.4-0ubuntu2.3) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
Processing triggers for man-db (2.10.2-1) ...
Setting up at-spi2-core (2.44.0-3) ...
Processing triggers for libgdk-pixbuf-2.0-0:amd64 (2.42.8+dfsg-1ubuntu0.3) ...
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.

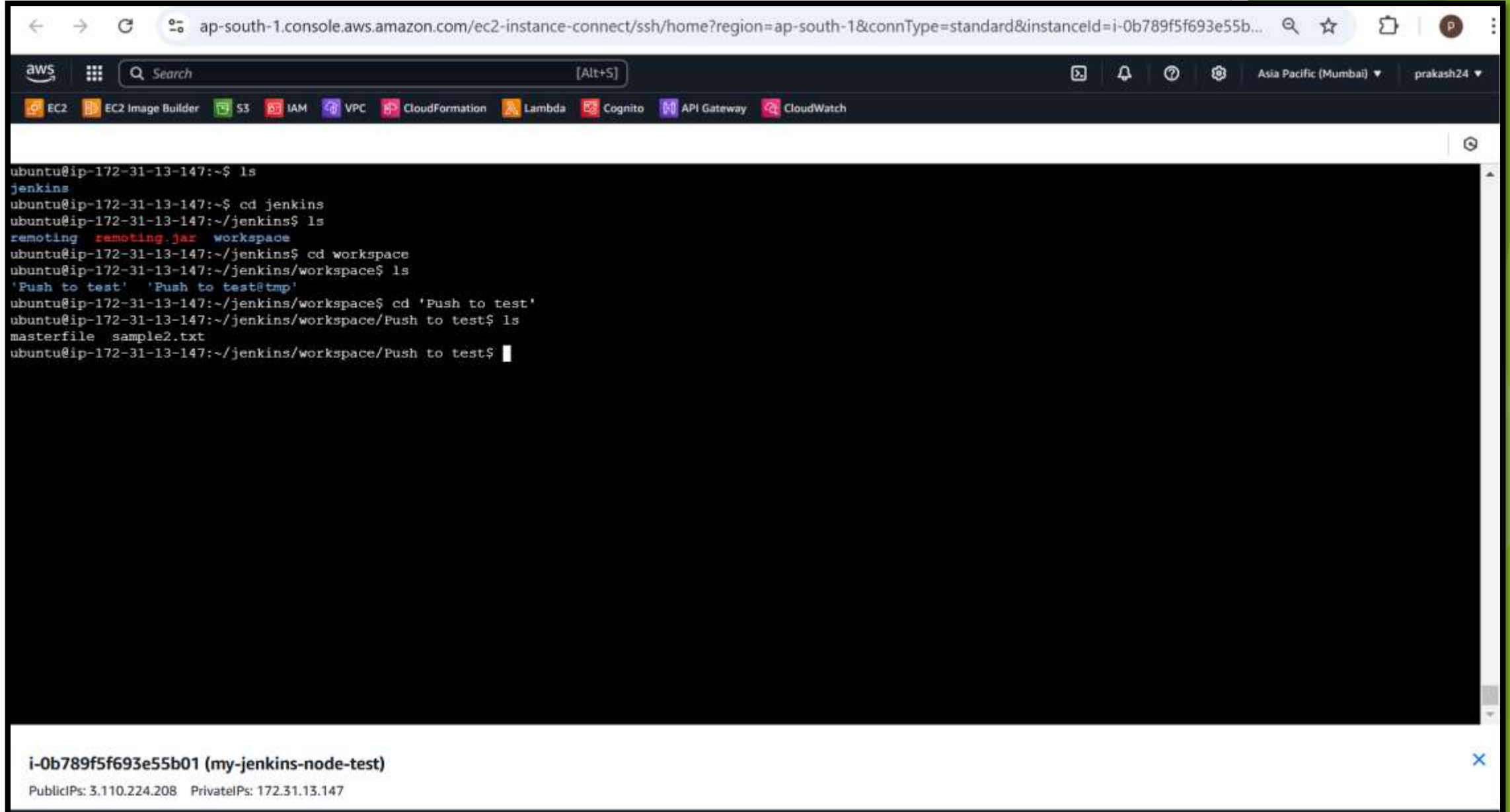
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-5-185:~$ ls
jenkins
ubuntu@ip-172-31-5-185:~$ cd jenkins
ubuntu@ip-172-31-5-185:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-5-185:~/jenkins$ cd workspace
ubuntu@ip-172-31-5-185:~/jenkins/workspace$ ls
'Push to prod' 'Push to prod@tmp' 'Push to test' 'Push to test@tmp'
ubuntu@ip-172-31-5-185:~/jenkins/workspace$ cd 'Push to prod'
ubuntu@ip-172-31-5-185:~/jenkins/workspace/Push to prod$ ls
masterfile  sample.txt
ubuntu@ip-172-31-5-185:~/jenkins/workspace/Push to prod$
```

Below the terminal output, the instance details are shown:

i-0d757eb8e00b231a3 (my-jenkins-node-pord)
PublicIPs: 65.0.108.181 PrivateIPs: 172.31.5.185

The bottom of the console shows the CloudShell logo, a Feedback link, and the copyright notice: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

We can see the files pull from test branch in workspace folder



The screenshot shows the AWS Management Console interface for an EC2 instance. The terminal window displays the following commands and output:

```
ubuntu@ip-172-31-13-147:~$ ls
jenkins
ubuntu@ip-172-31-13-147:~$ cd jenkins
ubuntu@ip-172-31-13-147:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-13-147:~/jenkins$ cd workspace
ubuntu@ip-172-31-13-147:~/jenkins/workspace$ ls
'Push to test'  'Push to test@tmp'
ubuntu@ip-172-31-13-147:~/jenkins/workspace$ cd 'Push to test'
ubuntu@ip-172-31-13-147:~/jenkins/workspace/Push to test$ ls
masterfile  sample2.txt
ubuntu@ip-172-31-13-147:~/jenkins/workspace/Push to test$
```

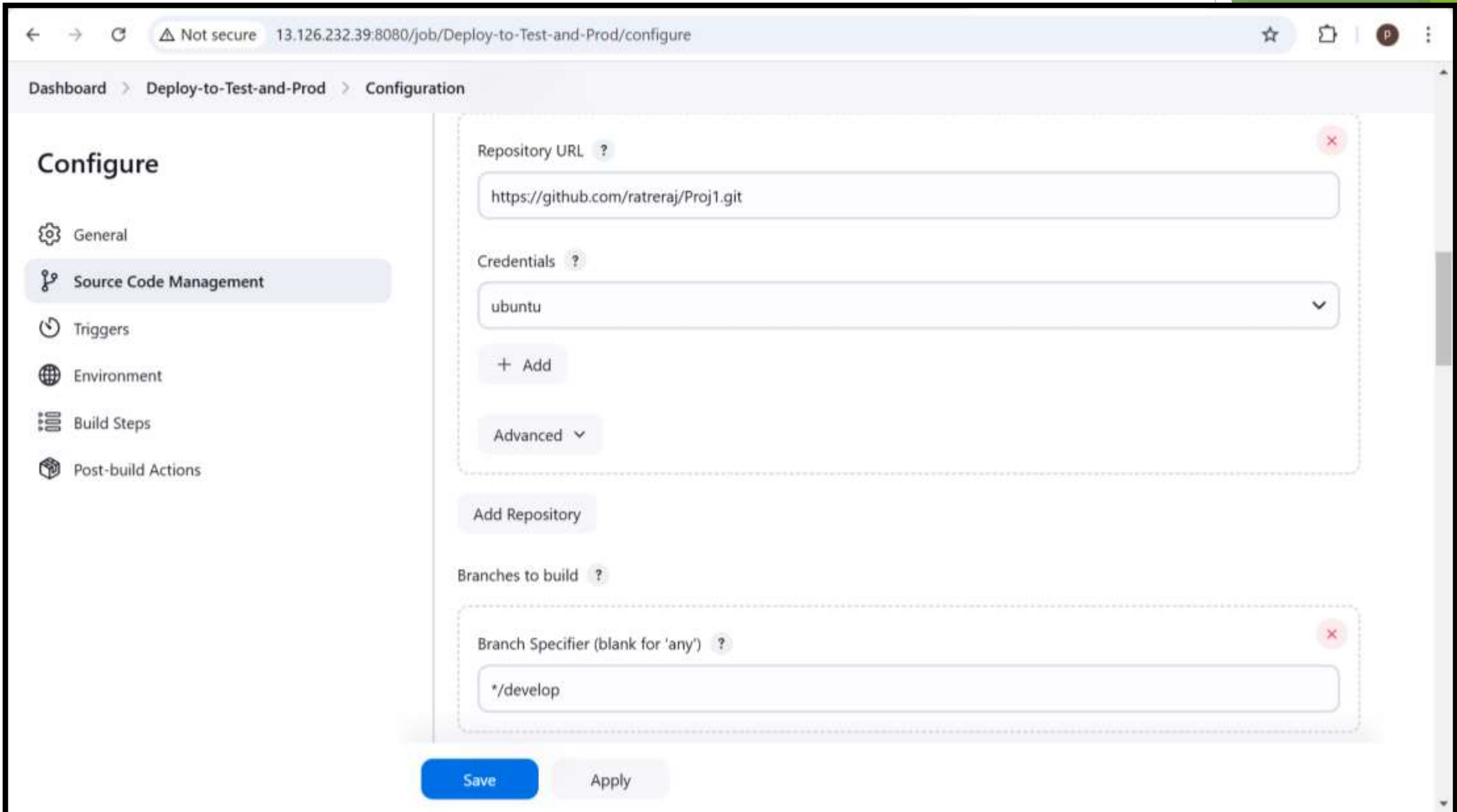
The instance ID is **i-0b789f5f693e55b01 (my-jenkins-node-test)**. The public IP is 3.110.224.208 and the private IP is 172.31.13.147.

Module 6: Jenkins Assignment - 3

Tasks To Be Performed:

1. Create a pipeline in Jenkins
2. Once push is made to “develop” a branch in Git, trigger job “test”. This will copy Git files to test node
3. If test job is successful, then prod job should be triggered
4. Prod jobs should copy files to prod node

Configured the pipeline “Deploy-to-test-and-prod” when code push to develop branch



The screenshot shows the Jenkins configuration interface for a pipeline named "Deploy-to-Test-and-Prod". The browser address bar indicates the URL is `13.126.232.39:8080/job/Deploy-to-Test-and-Prod/configure`. The left sidebar contains a "Configure" section with the following options: General, Source Code Management (selected), Triggers, Environment, Build Steps, and Post-build Actions. The main configuration area is titled "Configure" and includes the following fields:

- Repository URL**: `https://github.com/ratreraaj/Proj1.git`
- Credentials**: `ubuntu`
- + Add** button
- Advanced** dropdown menu
- Add Repository** button
- Branches to build**: `*/develop`

At the bottom of the configuration area, there are two buttons: **Save** and **Apply**.

Configured the pipeline “Deploy-to-test-and-prod” to trigger a Push-to-prod pipeline after stable build

The screenshot shows the Jenkins configuration interface for a pipeline named "Deploy-to-Test-and-Prod". The browser address bar indicates the URL is `13.126.232.39:8080/job/Deploy-to-Test-and-Prod/configure`. The left sidebar contains navigation links: "Dashboard", "Deploy-to-Test-and-Prod", and "Configuration". The main content area is titled "Configure" and includes a sidebar with configuration categories: "General", "Source Code Management", "Triggers", "Environment", "Build Steps", and "Post-build Actions". The "Post-build Actions" section is active, showing a description: "Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs." A dashed box highlights the "Build other projects" action. It is configured with "Projects to build" set to "Push-to-prod" and the trigger condition "Trigger only if build is stable" selected. Below this, there are "Add post-build action" and "Add build step" buttons. At the bottom, there are "Save" and "Apply" buttons. The footer of the page shows "REST API" and "Jenkins 2.493".

Dashboard > Deploy-to-Test-and-Prod > Configuration

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Configure

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

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Build other projects ?

Projects to build

Push-to-prod

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

Add post-build action

Save Apply

REST API Jenkins 2.493

Configured shell script to copy all file from Test Node to Prod Node

The screenshot shows the Jenkins web interface for configuring a job named 'Push-to-prod'. The left sidebar contains a 'Configure' section with a 'Build Steps' tab selected. The main area is titled 'Execute shell' and contains a text editor with a shell script. The script defines source and destination paths, validates the source path remotely using ssh, and then uses rsync to copy files to the destination. The script includes error handling with echo and exit commands. At the bottom of the configuration area are 'Save' and 'Apply' buttons.

Dashboard > Push-to-prod > Configuration

Configure

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

Execute shell ?

Command

[See the list of available environment variables](#)

```
#!/bin/bash

# Define source and destination paths
source_path="ubuntu@ip-172-31-8-37:/home/ubuntu/jenkins/workspace/Deploy-to-Test-and-Prod/"
destination_path="/home/ubuntu/jenkins/workspace/Push-to-prod/"

# Validate source path remotely
ssh ubuntu@ip-172-31-8-37 "test -d /home/ubuntu/jenkins/workspace/Deploy-to-Test-and-Prod"
if [ $? -ne 0 ]; then
    echo "Error: Source path '$source_path' does not exist. Exiting."
    exit 1
fi

# Execute rsync
rsync -avz --delete "$source_path" "$destination_path" -e "ssh -o StrictHostKeyChecking=no"
if [ $? -ne 0 ]; then
    echo "Error: rsync failed. Exiting with non-zero exit code."
    exit 1
fi

echo "rsync successful."
```

Save Apply

“Deploy-to-test-and-prod” run successfully & Downstream project Push-to prod

The screenshot displays the Jenkins web interface for a job named "Deploy-to-Test-and-Prod". The browser address bar shows the URL "13.126.232.39:8080/job/Deploy-to-Test-and-Prod/". The Jenkins logo and name are in the top left, and the user "raj" is logged in, as shown in the top right. The breadcrumb navigation indicates the path "Dashboard > Deploy-to-Test-and-Prod >".

On the left sidebar, the "Status" tab is selected, showing options like "Changes", "Workspace", "Build Now", "Configure", "Delete Project", "GitHub Hook Log", and "Rename".

The main content area shows the job "Deploy-to-Test-and-Prod" with a green checkmark icon. Below the job name, there is a section for "Downstream Projects" with a link to "Push-to-prod". A "Permalinks" section lists several build links:

- [Last build \(#12\), 3 min 12 sec ago](#)
- [Last stable build \(#12\), 3 min 12 sec ago](#)
- [Last successful build \(#12\), 3 min 12 sec ago](#)
- [Last failed build \(#3\), 58 min ago](#)
- [Last unsuccessful build \(#3\), 58 min ago](#)
- [Last completed build \(#12\), 3 min 12 sec ago](#)

At the bottom left, a "Builds" section shows a list of recent builds:

Build Number	Time	Status
#12	9:18 AM	Success
#11	9:14 AM	Success
#10	9:03 AM	Success
#9	8:56 AM	Success

“Push-to prod” run successfully

The screenshot shows the Jenkins web interface. The browser address bar indicates the URL `13.126.232.39:8080/job/Push-to-prod/12/console`. The Jenkins logo and name are in the top left. The top right shows a search icon, a shield with a red exclamation mark, a user profile for 'raj', and a 'log out' button. The breadcrumb navigation shows 'Dashboard > Push-to-prod > #12 > Console Output'. On the left sidebar, 'Console Output' is selected. The main area displays the console output for build #12, which is successful. The output text is as follows:

```
Started by upstream project "Deploy-to-Test-and-Prod" build number 12
originally caused by:
  Started by user raj
Running as SYSTEM
Building remotely on Prod in workspace /home/ubuntu/jenkins/workspace/Push-to-prod
The recommended git tool is: NONE
using credential 4b4cf88f-520e-45f2-badc-3f90dd52bbfc
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/Push-to-prod/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ratreraaj/Proj1.git # timeout=10
Fetching upstream changes from https://github.com/ratreraaj/Proj1.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_SSH to set credentials
Verifying host key using known hosts file
> git fetch --tags --force --progress -- https://github.com/ratreraaj/Proj1.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 077e7a825e15151b9e6bc71a5cda73d1729780ae (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 077e7a825e15151b9e6bc71a5cda73d1729780ae # timeout=10
Commit message: "Create sample.txt"
> git rev-list --no-walk 077e7a825e15151b9e6bc71a5cda73d1729780ae # timeout=10
```


File present in develop branch


github.com/ratreraj/Proj1/tree/develop

develop Proj1 /

Go to file

Add file

...












 **ratreraj** Create tyyiooo.txt

8823973 · 5 minutes ago

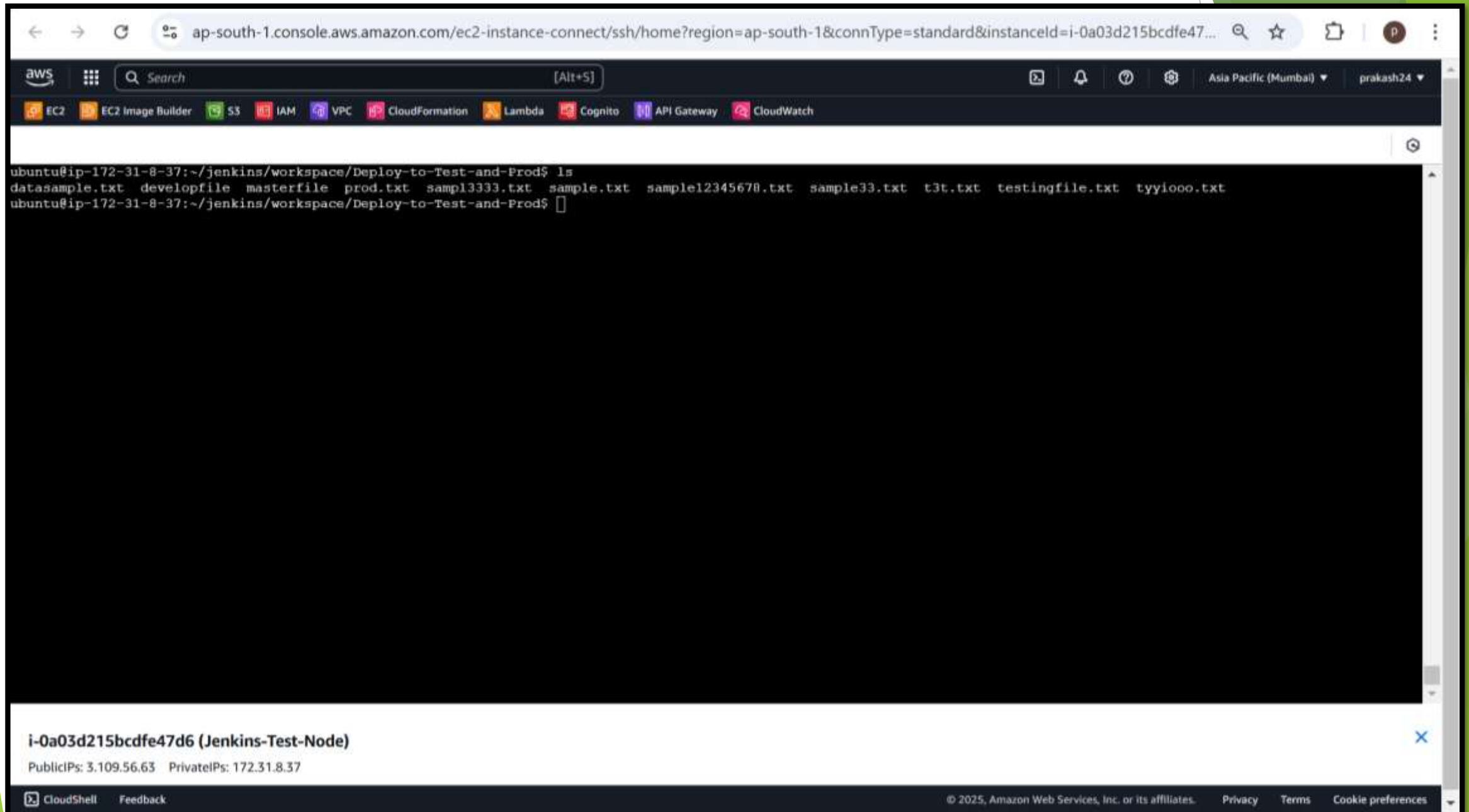
History

This branch is 12 commits ahead of, 1 commit behind master.

Contribute

Name	Last commit message	Last commit date
 datasample.txt	Create datasample.txt	5 hours ago
 developfile	dev commit	2 months ago
 masterfile	this is my first commit	2 months ago
 prod.txt	Create prod.txt	16 hours ago
 sampl3333.txt	Create sampl3333.txt	20 minutes ago
 sample.txt	Create sample.txt	19 hours ago
 sample12345678.txt	Create sample12345678.txt	14 minutes ago
 sample33.txt	Create sample33.txt	4 hours ago
 t3t.txt	Create t3t.txt	9 minutes ago
 testingfile.txt	Update testingfile.txt	4 hours ago
 tyyiooo.txt	Create tyyiooo.txt	5 minutes ago

File present in develop branch is pulled in Test Node after “Deploy-to-test-and-prod”

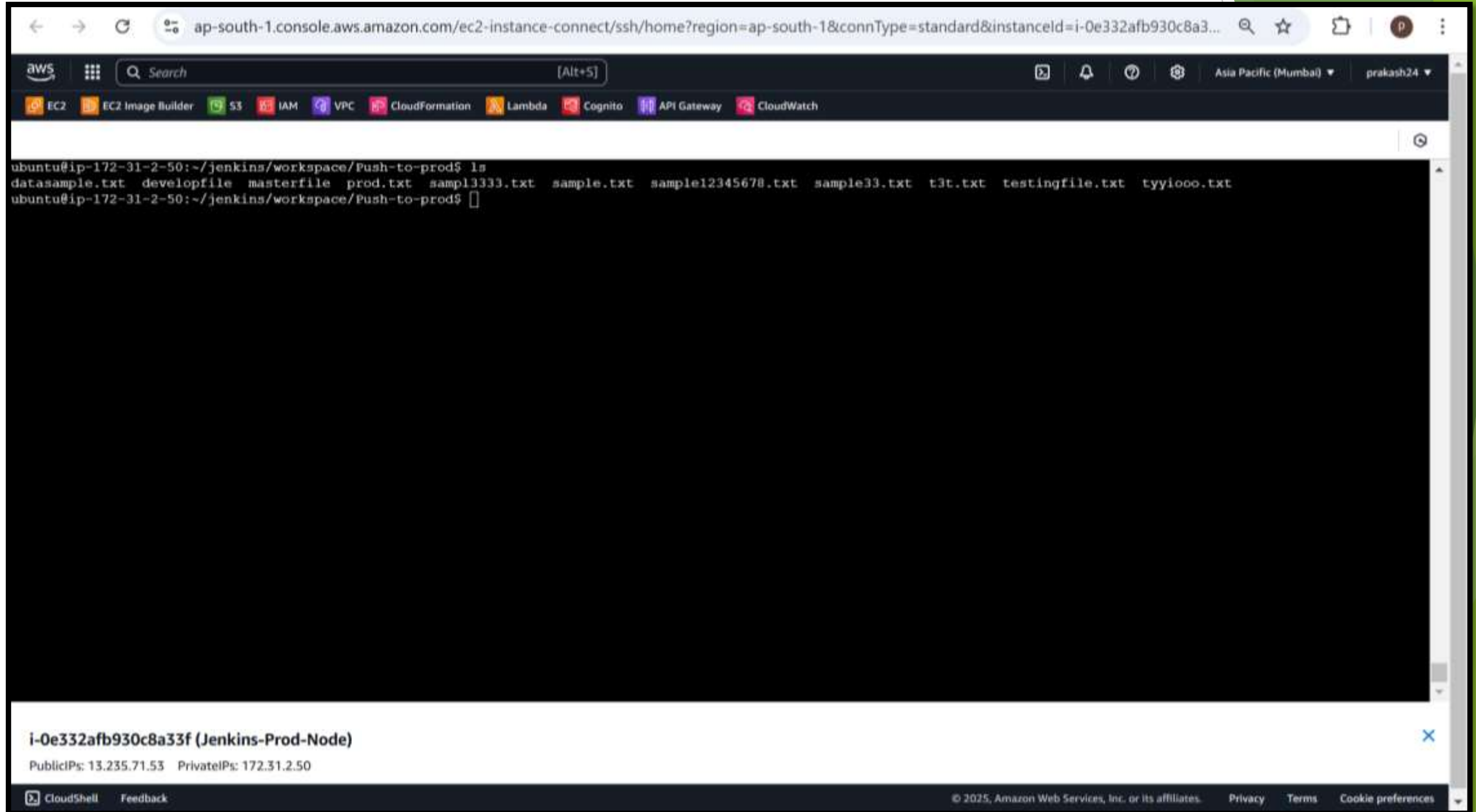


The screenshot shows the AWS CloudShell interface. The browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0a03d215bcdfe47...`. The AWS navigation bar at the top includes the AWS logo, a search bar, and a list of services: EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user's profile is 'prakash24' in the 'Asia Pacific (Mumbai)' region.

The terminal window shows a shell session on an Ubuntu instance. The prompt is `ubuntu@ip-172-31-8-37:~/jenkins/workspace/Deploy-to-Test-and-Prod$`. The user enters `ls`, and the output lists the following files: `datasample.txt developfile masterfile prod.txt sampl3333.txt sample.txt sample12345678.txt sample33.txt t3t.txt testingfile.txt tyyl000.txt`. The prompt returns to `ubuntu@ip-172-31-8-37:~/jenkins/workspace/Deploy-to-Test-and-Prod$`.

At the bottom of the terminal window, the instance details for `i-0a03d215bcdfe47d6 (Jenkins-Test-Node)` are shown, including Public IPs: 3.109.56.63 and Private IPs: 172.31.8.37. The footer of the page includes 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for Privacy, Terms, and Cookie preferences.

Same file copied Test Node after “Push-to-prod”



The screenshot shows the AWS CloudShell interface. The browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0e332afb930c8a3...`. The AWS console header includes the AWS logo, a search bar, and navigation links for EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user's profile is 'prakash24' in the 'Asia Pacific (Mumbai)' region.

The terminal window shows the following command and output:

```
ubuntu@ip-172-31-2-50:~/jenkins/workspace/Push-to-prod$ ls
datasample.txt  developfile  masterfile  prod.txt  sampl3333.txt  sample.txt  sample12345678.txt  sample33.txt  t3t.txt  testingfile.txt  tyyiooo.txt
ubuntu@ip-172-31-2-50:~/jenkins/workspace/Push-to-prod$
```

Below the terminal, the instance details for `i-0e332afb930c8a33f (Jenkins-Prod-Node)` are shown, including PublicIPs: 13.235.71.53 and PrivateIPs: 172.31.2.50.

The footer of the CloudShell interface includes links for CloudShell, Feedback, and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for Privacy, Terms, and Cookie preferences.