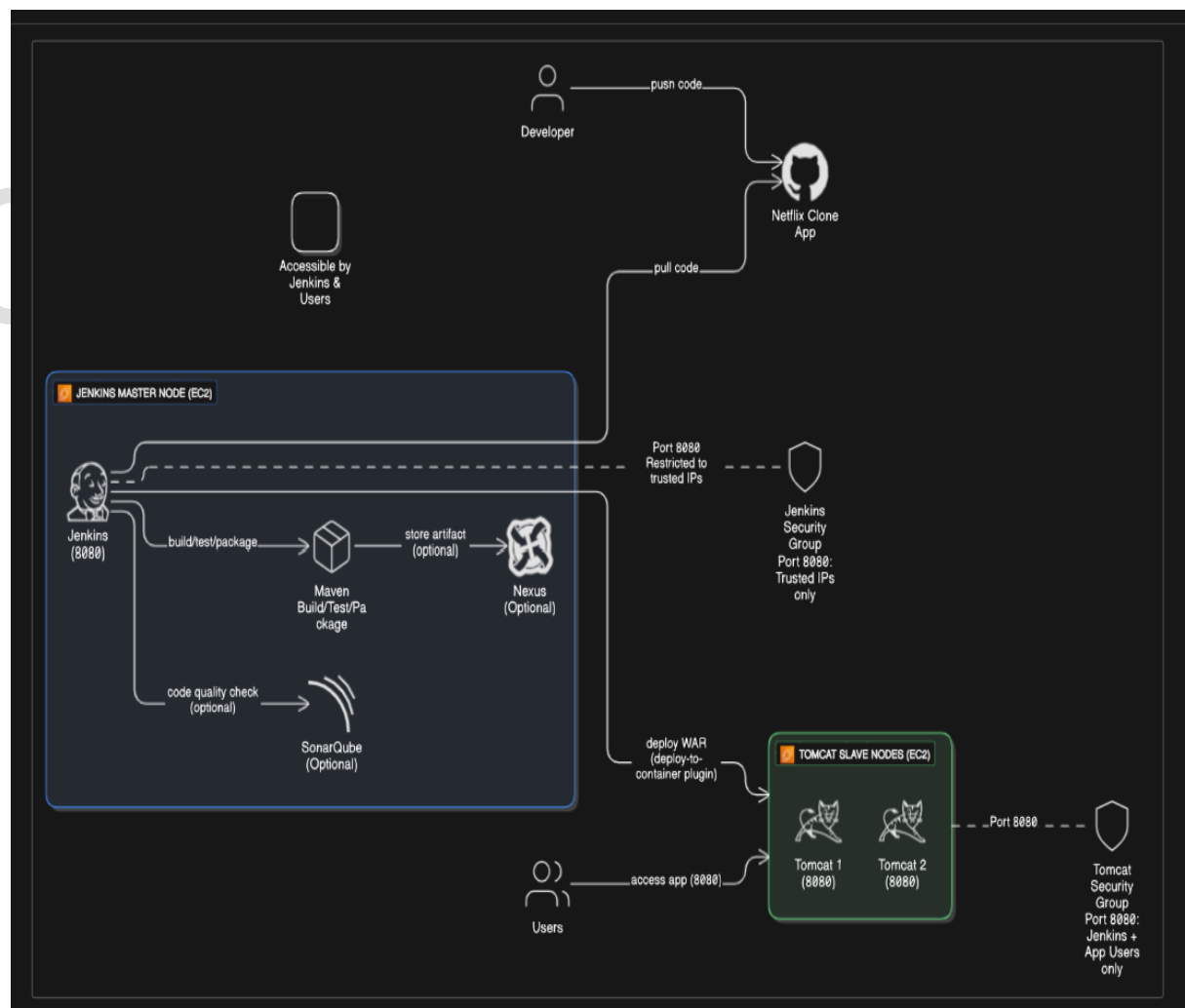


Netflix Jenkins CI/CD Pipeline Setup with Tomcat Server:

This task involves setting up a complete CI/CD pipeline using Jenkins to automate the deployment of a Netflix clone application. The solution is designed to follow industry-standard practices by utilizing Jenkins as the automation server. Jenkins is installed on a main node, while deployment targets are configured with Apache Tomcat to serve the application.

The pipeline fetches source code from a Git repository, compiles and tests the application using Maven, and packages it as a WAR file. The final stage automatically deploys the WAR file to Tomcat servers on the target nodes. This setup enables seamless, automated, and scalable deployment of applications in a DevOps environment, promoting faster release cycles, consistency, and reduced human error.



- Step 1: Sign into the AWS Console.
- Step 2: Navigate to EC2.
- Step 3: Create 1 instance – for Installation of Jenkins.
- Step 4: Name → AMI- Amazon Linux (choose Amazon Linux 2 AMI) → Instance Type :t2.micro → Create a key pair (.pem /.ppk) [if .ppk choosen then later convert it to .pem file using putty Gen] → Security Group – Allow all security/ Allow All TCP [Allow these ports- 22, 8080, 80, 50000) → Click on Launch Instance.

[NOTE: Allowing All Security & All TCP -: It's strongly discouraged in production environments due to serious security risks. It is always a best practise to allow the ports which is required.]

The screenshot displays the AWS Management Console's 'Launch an instance' wizard. The wizard is currently at the 'Application and OS Images (Amazon Machine Image)' step. The selected AMI is 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type'. The instance type is 't2.micro'. A 'Create key pair' modal is open, showing 'netflix' as the key pair name, 'RSA' as the key pair type, and '.pem' as the private key file format. The modal also includes a warning to store the private key securely.

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 **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-03edbe403ec8522ed (64-bit (x86)) / ami-06d5bdfada20e525f (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Amazon Linux 2 Kernel 5.10 AMI 2.0.20250428.0 x86_64 HVM gp2

Architecture **AMI ID** **Publish Date** **Username** **Verified provider**

64-bit (x86) ami-03edbe403ec8522ed 2025-04-25 ec2-user

Instance type [Info](#) [Get advice](#)

Instance type

t2.micro
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

Additional costs apply for AMIs with pre-installed software

On-Demand Linux base pricing: 0.0124 USD per Hour On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure

Key pair name - required

Select

Network settings [Info](#)

Network [Info](#)

vpc-05ef0cd1f2b4621b

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

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

⚠ 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](#)

[Cancel](#) [Create key pair](#)

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2...[read more](#)
ami-062f0cc54dbf0bf1

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

Network settings

VPC - required Info

vpc-05ef0dc1f2b4621b (default) ↻

Subnet Info

No preference ↻ [Create new subnet](#)

Auto-assign public IP Info

Enable ↻

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Common security groups Info

Select security groups

ALL sg-079dd7a9f5f90a71 ✕

VPC: vpc-05ef0dc1f2b4621b

Security groups that you add or remove here will be added to or removed from all your network interfaces.

[Compare security group rules](#)

Configure storage

Advanced

1x 8 GiB gp2 ↻ Root volume, Not encrypted

Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage

[Add new volume](#)

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

Summary

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)

ami-03edbe403ec8522ed

Virtual server type (instance type)

t2.micro

Firewall (security group)

ALL

Storage (volumes)

1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

Step 5: Select the jenkins instance → click on Connect.

EC2

Dashboard

EC2 Global View

Events

Instances

Instances (1/2) Info

Last updated less than a minute ago

[Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
Netfix Jenkins	i-0b5778bd90bf14298	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-3-110-16
linux	i-055cf8c52ada27be8	Running	t2.micro	2/2 checks pass	View alarms +	ap-south-1b	ec2-3-111-58

EC2

Instances

i-0b5778bd90bf14298

Connect to instance

Connect to instance Info

Connect to your instance i-0b5778bd90bf14298 (Netfix Jenkins) using any of these options

EC2 Instance Connect **Session Manager** **SSH client** **EC2 serial console**

Instance ID

i-0b5778bd90bf14298 (Netfix Jenkins)

Connection Type

☒ Connect using EC2 Instance Connect

Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ Connect using EC2 Instance Connect Endpoint

Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IPv4 address

3.110.162.42

IPv6 address

-

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user ✕

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

[Cancel](#) [Connect](#)

```

Last login: Mon May 5 09:13:03 2025 from ec2-13-233-177-4.ap-south-1.compute.amazonaws.com
Amazon Linux 2
AL2 End of Life is 2026-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-34-15 ~]$

```

i-0b5778bd90bf14298 (Netfix Jenkins)

PublicIPs: 3.110.162.42 PrivateIPs: 172.31.34.15

Step 6: Commands:

- `sudo -i`
- use the below link to install Jenkins
https://github.com/confideoit/Confideo-All_Setup/blob/main/jenkins.sh
- Copy the entire script and paste in the terminal.
- Don't forget to start the service

```

aws Search [Alt+S] Asia Pacific (Mumbai) santhosh @ 9781-5017-6450
java-17-amazon-corretto.x86_64 1:17.0.15+6-1.amzn2.1 jenkins.noarch 0:2.504.1-1.1
Dependency Installed:
  dejavu-sans-mono-fonts.noarch 0:2.33-6.amzn2      dejavu-serif-fonts.noarch 0:2.33-6.amzn2      java-17-amazon-corretto-headless.x86_64 1:17.0.15+6-1.amzn2.1
  libXt.x86_64 0:1.1.5-3.amzn2.0.2
Complete!
[root@ip-172-31-34-15 ~]# systemctl start jenkins.service
[root@ip-172-31-34-15 ~]# systemctl status jenkins.service
● Jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2025-05-05 09:17:52 UTC; 3s ago
     Main PID: 5824 (java)
    CGroup: /system.slice/jenkins.service
            └─5824 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

May 05 09:17:44 ip-172-31-34-15.ap-south-1.compute.internal jenkins[5824]: 957ba4c50alf4537b32afeal261bd987
May 05 09:17:44 ip-172-31-34-15.ap-south-1.compute.internal jenkins[5824]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
May 05 09:17:44 ip-172-31-34-15.ap-south-1.compute.internal jenkins[5824]: *****
May 05 09:17:44 ip-172-31-34-15.ap-south-1.compute.internal jenkins[5824]: *****
May 05 09:17:52 ip-172-31-34-15.ap-south-1.compute.internal jenkins[5824]: 2025-05-05 09:17:52.848+0000 [id=32] INFO jenkins.InitReactorRunner$...ization
May 05 09:17:52 ip-172-31-34-15.ap-south-1.compute.internal jenkins[5824]: 2025-05-05 09:17:52.871+0000 [id=24] INFO hudson.lifecycle.Lifecycle...running
May 05 09:17:52 ip-172-31-34-15.ap-south-1.compute.internal systemd[1]: Started Jenkins Continuous Integration Server.
May 05 09:17:54 ip-172-31-34-15.ap-south-1.compute.internal jenkins[5824]: 2025-05-05 09:17:54.552+0000 [id=48] INFO h.m.DownloadService$Downlo...staller
May 05 09:17:54 ip-172-31-34-15.ap-south-1.compute.internal jenkins[5824]: 2025-05-05 09:17:54.553+0000 [id=48] INFO hudson.util.Retrier$start:...empt #1
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-34-15 ~]#
  
```

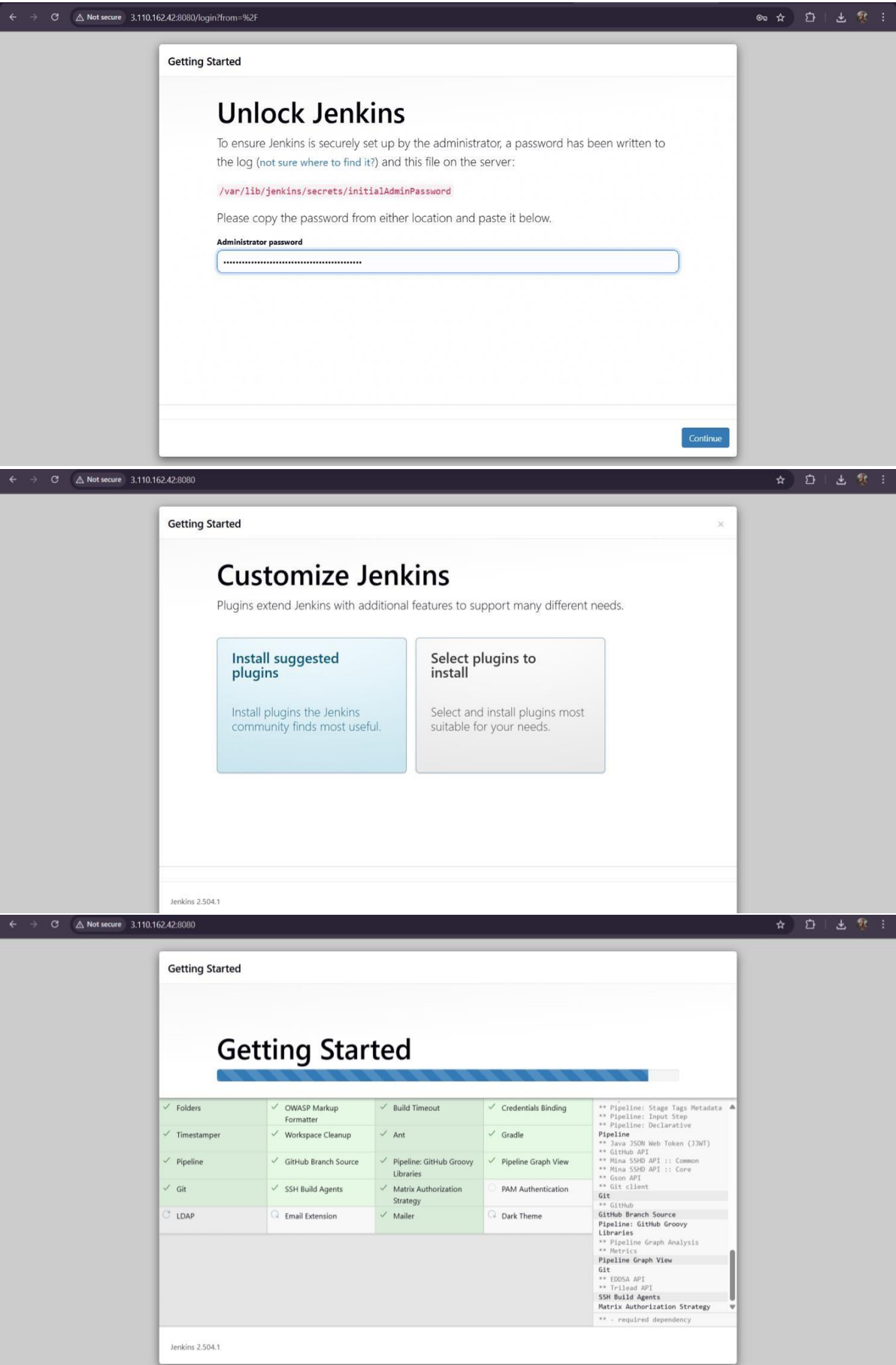
i-0b5778bd90bf14298 (Netfix Jenkins)
PublicIPs: 3.110.162.42 PrivateIPs: 172.31.34.15

Step 7: After, the service get started copy the IP of your Jenkins instance and paste in the browser with port-8080 (e.g: 3.110.162.42:8080).

Step 8: Paste the IP – Jenkins Sign-in page will appear.

Step 9: Copy the `/var/lib/jenkins/secrets/initialAdminPassword` which is present in the jenkins sign-in page

- Command:
 - `cat /var/lib/jenkins/secrets/initialAdminPassword`
- You will get a code copy and paste it in the Administrator password slot.



The image displays three sequential screenshots of the Jenkins 2.504.1 installation wizard, accessed via a web browser at 3.110.162.42:8080. The browser's address bar indicates the connection is 'Not secure'.

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.504.1 Skip and continue as admin [Save and Continue](#)

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.504.1 Not now [Save and Finish](#)

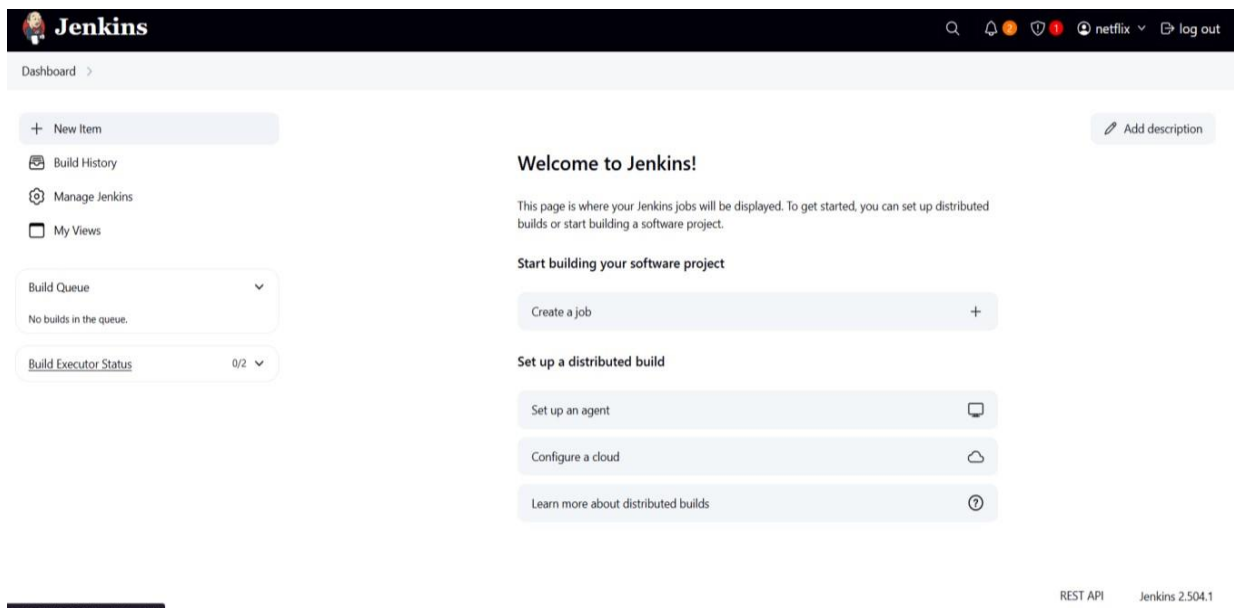
Getting Started

Jenkins is ready!

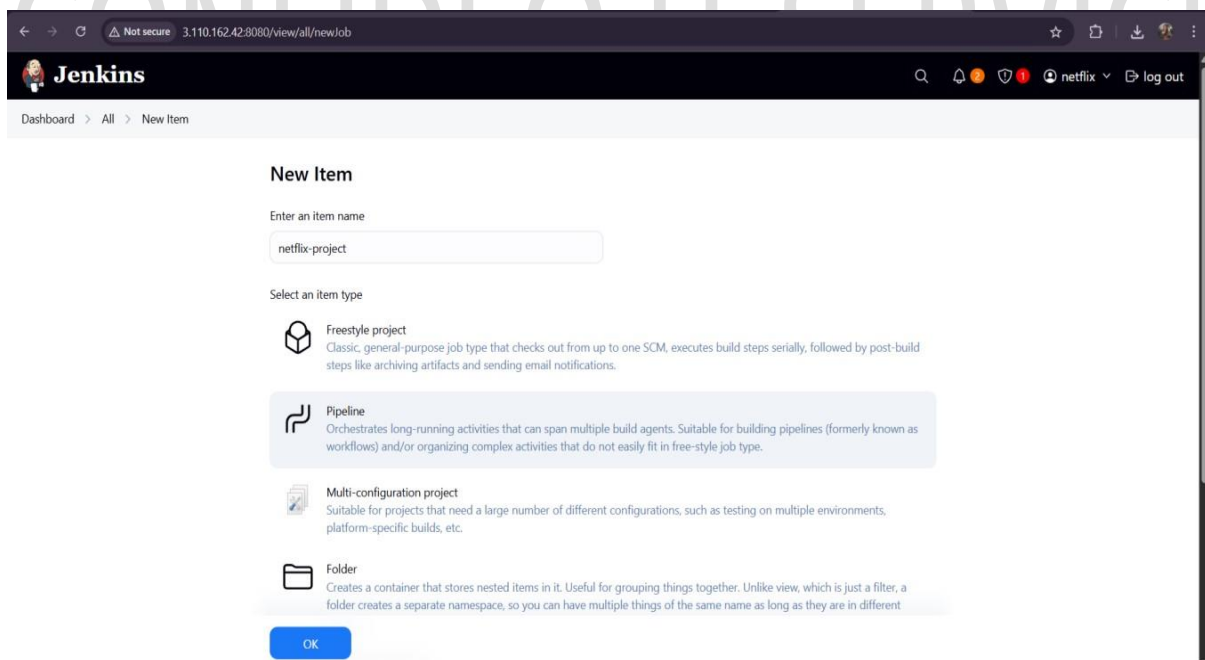
Your Jenkins setup is complete.

[Start using Jenkins](#)

Jenkins 2.504.1



Step 10: Click on New Item → Enter the name of the item → choose Pipeline → Click on OK.



Step 11: Now Create 2 instances – for Installation of Apache Tomcat.

Step 12: Give 2 in the Number of Instances → Name → AMI- Amazon Linux (choose Amazon Linux 2 AMI) → Instance Type – t2.micro → use the existing key pair → Use the Existing Security Group → Click on Launch Instance.

Launch an instance

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

Name
Slave1-Netflix

Add additional tags

Application and OS Images (Amazon Machine Image)

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.

Search our full catalog including 1000s of application and OS images

RecentsQuick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Free tier eligible

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-03e8be403ec8522ed (64-bit x86) / ami-06d5b0ffada20e525f (64-bit ARM)
Virtualization: true ENA-enabled: true Root device type: ebs

Description
Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Amazon Linux 2 Kernel 5.10 AMI 2.0.20250428.0 x86_64 HVM gp2

Architecture
64-bit (x86)

AMI ID
ami-03e8be403ec8522ed

Publish Date
2025-04-25

Username
ec2-user

Verified provider

Instance type

Free tier eligible

t2.micro
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations
Compare instance types

Key pair (login)

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.

Key pair name - required
netflix

Create new key pair

Network settings

VPC - required
vpc-05ef0cdc1f2b4621b (default)
172.31.0.0/16

Subnet
No preference

Create new subnet

Auto-assign public IP
Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group
Select existing security group

Common security groups
Select security groups
ALL sg-079dd7a97f5f90a71
VPC: vpc-05ef0cdc1f2b4621b

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Configure storage

Advanced

1x 8 GiB gp2 Root volume, Not encrypted

Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

Edit

Advanced details

Summary

Number of instances
2

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-062f0cc54dbfd8ef1

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Cancel Launch instance
Preview code

Summary

Number of instances
2

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-03e8be403ec8522ed

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Cancel Launch instance
Preview code

Summary

Number of instances
2

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-03e8be403ec8522ed

Virtual server type (instance type)
t2.micro

Firewall (security group)
ALL

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

Cancel Launch instance
Preview code

Summary

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-03e8be403ec8522ed

Virtual server type (instance type)
t2.micro

Firewall (security group)
ALL

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

Cancel Launch instance
Preview code

Instances (4)

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find instance by attribute or tag (case-sensitive) All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
<input type="checkbox"/>	LINUX	i-055cf8c52ada27be8	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-3-111-58-
<input type="checkbox"/>	Netflix Jenkins	i-0b5778bd90bf14298	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-3-110-16-
<input type="checkbox"/>	Slave1-Netflix	i-05ce976bd420fb9f	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13-235-8-
<input type="checkbox"/>	Slave2-Netflix	i-088cd546f690b1286	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13-127-2-

Select an instance

Step 13: Select the 1st Apache Instance → click on connect.

The screenshot shows the AWS Management Console for the 'ap-south-1' region. Under the 'Instances' section, the instance 'Slave1-Netfli' (ID: i-05ce976bd420f6b9f) is selected. The 'Connect' button is visible. Below the console, a terminal window shows the command to connect to the instance:

```
aws ssm start-session --target i-05ce976bd420f6b9f
```

Step 14: Commands:

- `sudo -i`
- `yum install git java-1.8.0-openjdk maven -y`
- `sudo yum install java-17-amazon-corretto -y`
- `vim tomcat.sh`

Inside vim tomcat.sh paste the script:

https://github.com/confideoit/Confideo-All_Setup/blob/main/tomcat.sh

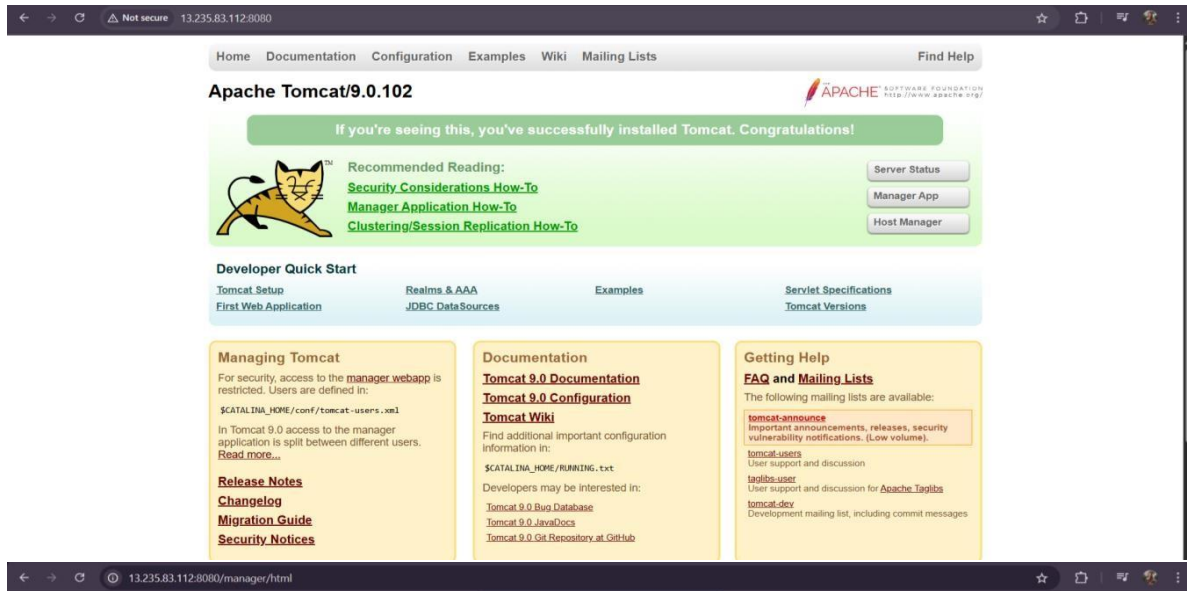
- `sh tomcat.sh`

The screenshot shows the AWS Management Console for the 'ap-south-1' region. Under the 'Instances' section, the instance 'Slave1-Netfli' (ID: i-05ce976bd420f6b9f) is selected. The terminal window shows the output of the 'sh tomcat.sh' command, which includes the installation of Java and Maven, and the setup of the Tomcat web server.

```

apache-tomcat-9.0.102/webapps/manager/WEB-INF/jsp/sessionDetail.jsp
apache-tomcat-9.0.102/webapps/manager/WEB-INF/jsp/sessionList.jsp
apache-tomcat-9.0.102/webapps/manager/WEB-INF/web.xml
apache-tomcat-9.0.102/webapps/manager/cas/manager.css
apache-tomcat-9.0.102/webapps/manager/images/asf-logo.svg
apache-tomcat-9.0.102/webapps/manager/images/tomcat.svg
apache-tomcat-9.0.102/webapps/manager/index.jsp
apache-tomcat-9.0.102/webapps/manager/status.xsd
apache-tomcat-9.0.102/webapps/manager/xform.xsl
apache-tomcat-9.0.102/bin/ciphers.sh
apache-tomcat-9.0.102/bin/configtest.sh
apache-tomcat-9.0.102/bin/digest.sh
apache-tomcat-9.0.102/bin/makebase.sh
apache-tomcat-9.0.102/bin/setclasspath.sh
apache-tomcat-9.0.102/bin/shutdown.sh
apache-tomcat-9.0.102/bin/startup.sh
apache-tomcat-9.0.102/bin/tool-wrapper.sh
apache-tomcat-9.0.102/bin/version.sh
Using CATALINA_BASE: /root/apache-tomcat-9.0.102
Using CATALINA_HOME: /root/apache-tomcat-9.0.102
Using CATALINA_TMPDIR: /root/apache-tomcat-9.0.102/temp
Using JRE_HOME: /root/apache-tomcat-9.0.102/bin/bootstrap.jar:/root/apache-tomcat-9.0.102/bin/tomcat-juli.jar
Using CLASSPATH:
tomcat started.
[ec2-user@ip-172-31-3-235 ~]$
  
```

- Step 15: Now, check that the tomcat is installed properly or not.
- Step 16: Copy the instance public ip and paste it.
- Step 17: Once the page appears → click on manager app → user=tomcat & pw=confideo123.



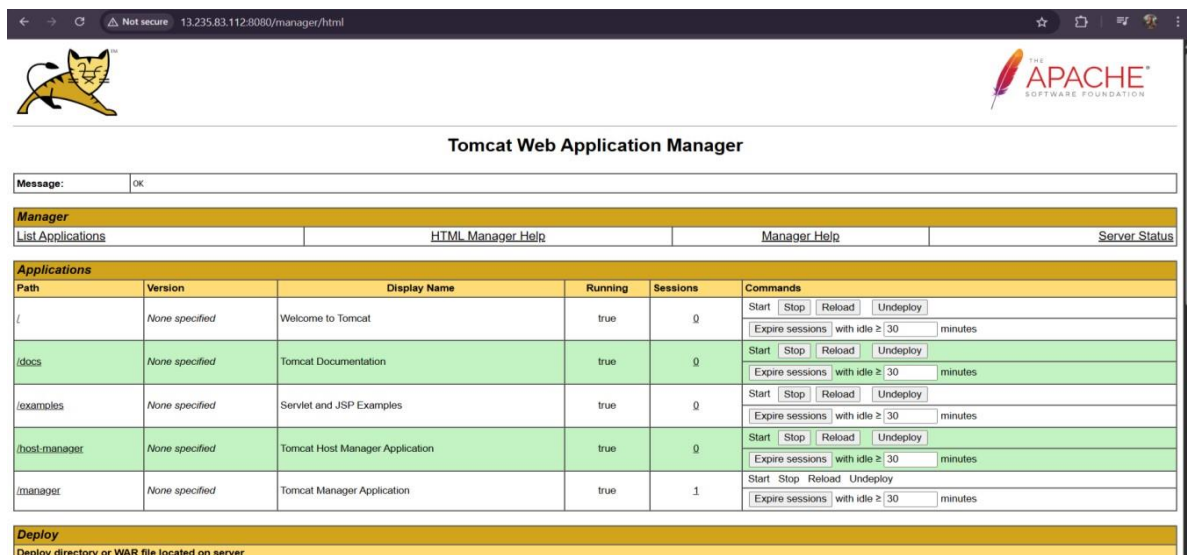
Sign in

http://13.235.83.112:8080

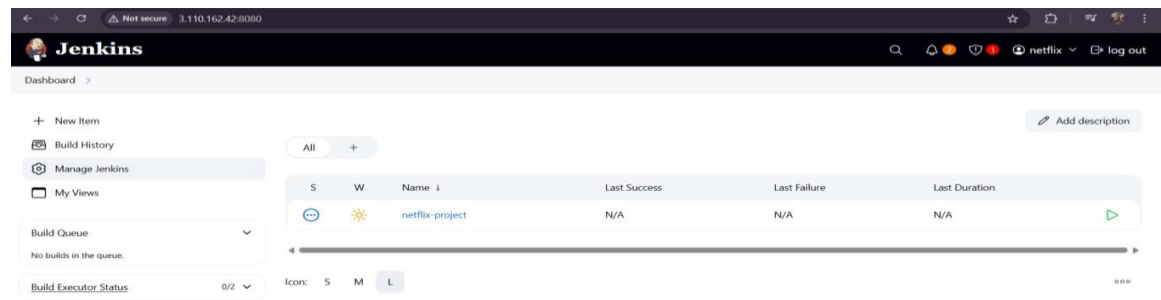
Your connection to this site is not private

Username:

Password:



- Step 18: Repeat from step-14 to step-17 for another Instance.
- Step 19: Now login to Jenkins Console.
- Step 20: From Dashboard → Click on manage Jenkins → nodes → click on new node → give node name → select permanent agent → Save
- In the Host add the private-ip.



Jenkins Dashboard

Build Queue: No builds in the queue.

Build Executor Status: 0/2

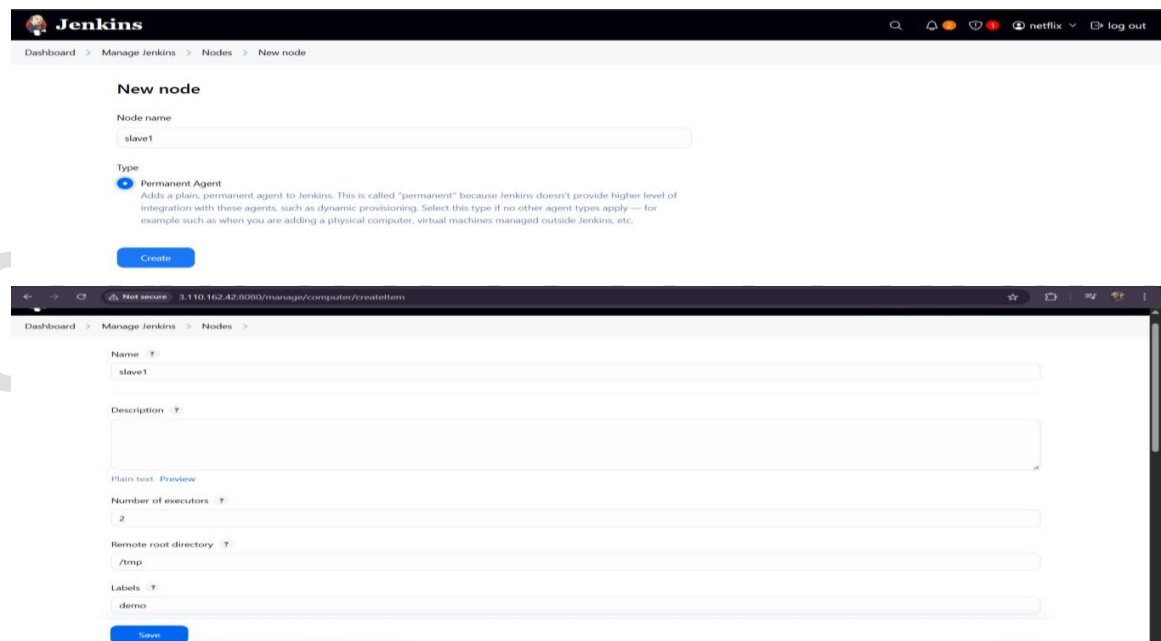
S	W	Name	Last Success	Last Failure	Last Duration
		netflix-project	N/A	N/A	N/A



Jenkins Nodes

+ New Node

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	5.25 GiB	0 B	5.25 GiB	0ms
	Data obtained	37 min	37 min	37 min	37 min	37 min	37 min

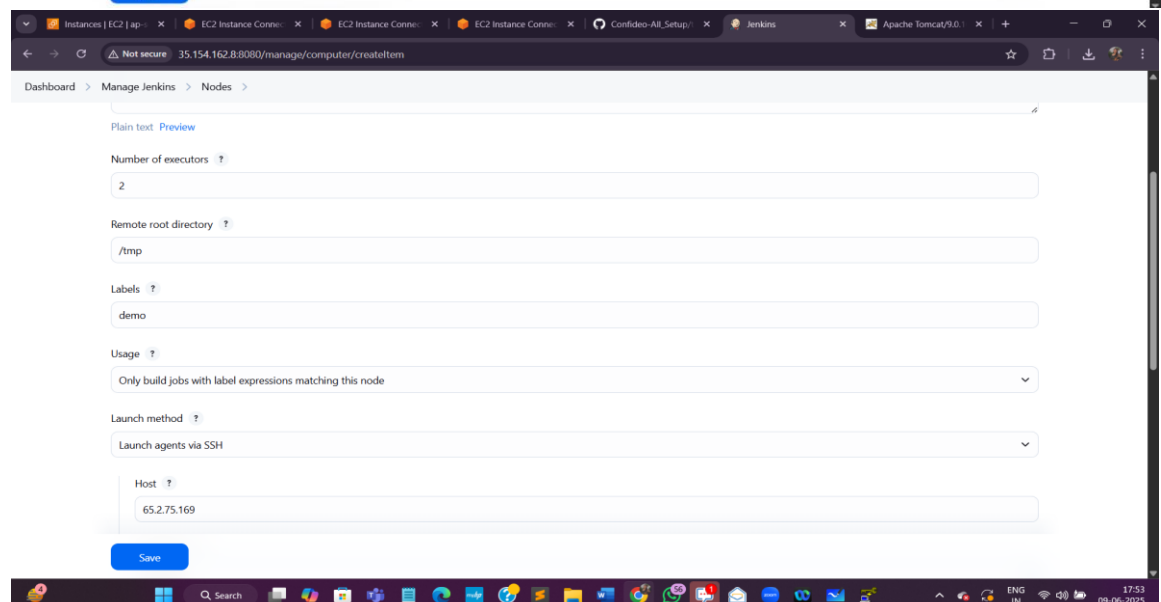


Jenkins New node

Node name: slave1

Type: Permanent Agent

Create



Jenkins Node configuration

Name: slave1

Description:

Number of executors: 2

Remote root directory: /tmp

Labels: demo

Usage: Only build jobs with label expressions matching this node

Launch method: Launch agents via SSH

Host: 65.2.75.169

Save

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain
Global credentials (unrestricted)

Kind
SSH Username with private key

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

ID

Description

Username

Jenkins Credentials Provider: Jenkins

Username
ec2-user

☐ Treat username as secret

Private Key
☒ Enter directly

Key
Enter New Secret Below
-----BEGIN RSA PRIVATE KEY-----
pF+61kTDpCNZD923oxuG/11+Q8g/kjuih3aThY693tX0PwJYkX8hyAs/utisPn68ka
J/6p3P1HJec3YJccSH0ebBONAFcACpgS2N7N2XLre7gellar+01om1ba6
-----END RSA PRIVATE KEY-----

Passphrase

Cancel Add

Jenkins

Dashboard

+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue
No builds in the queue.

Build Executor Status

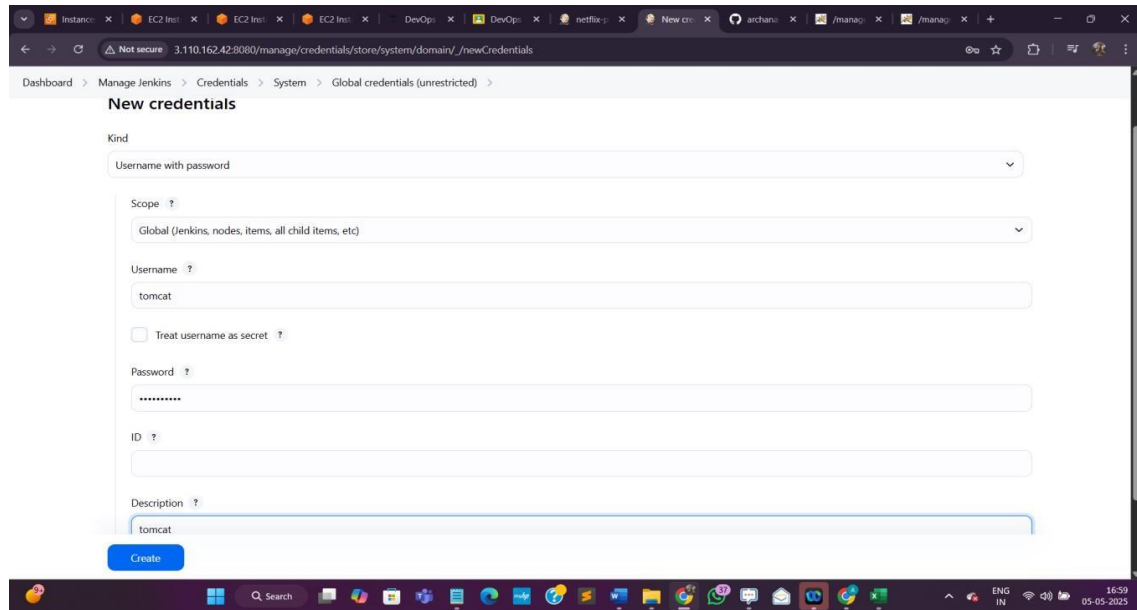
Built-in Node 0/2

slave1 0/2

S	W	Name	Last Success	Last Failure	Last Duration
...	☀	netflix-project	N/A	N/A	N/A

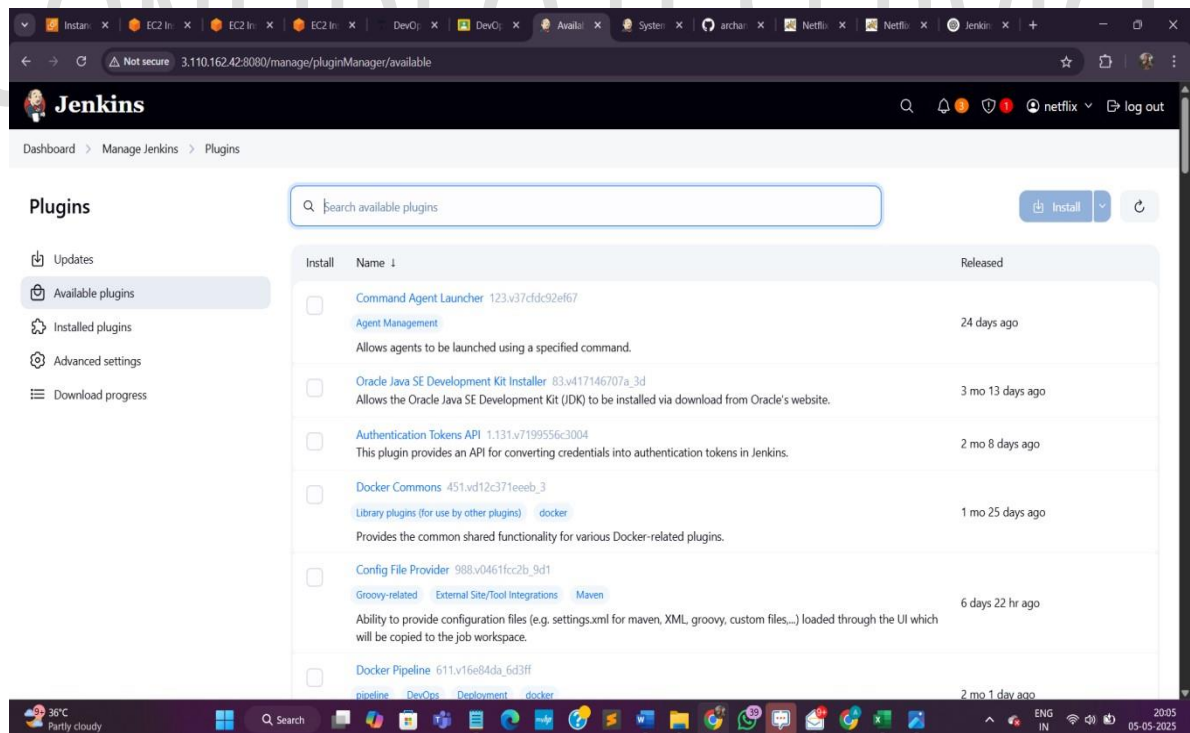
Step 21: Now, Add the Tomcat Credentials

Step 22: Click on manage Jenkins → Credentials → Global → Add credentials → give the tomcat username & password → create .



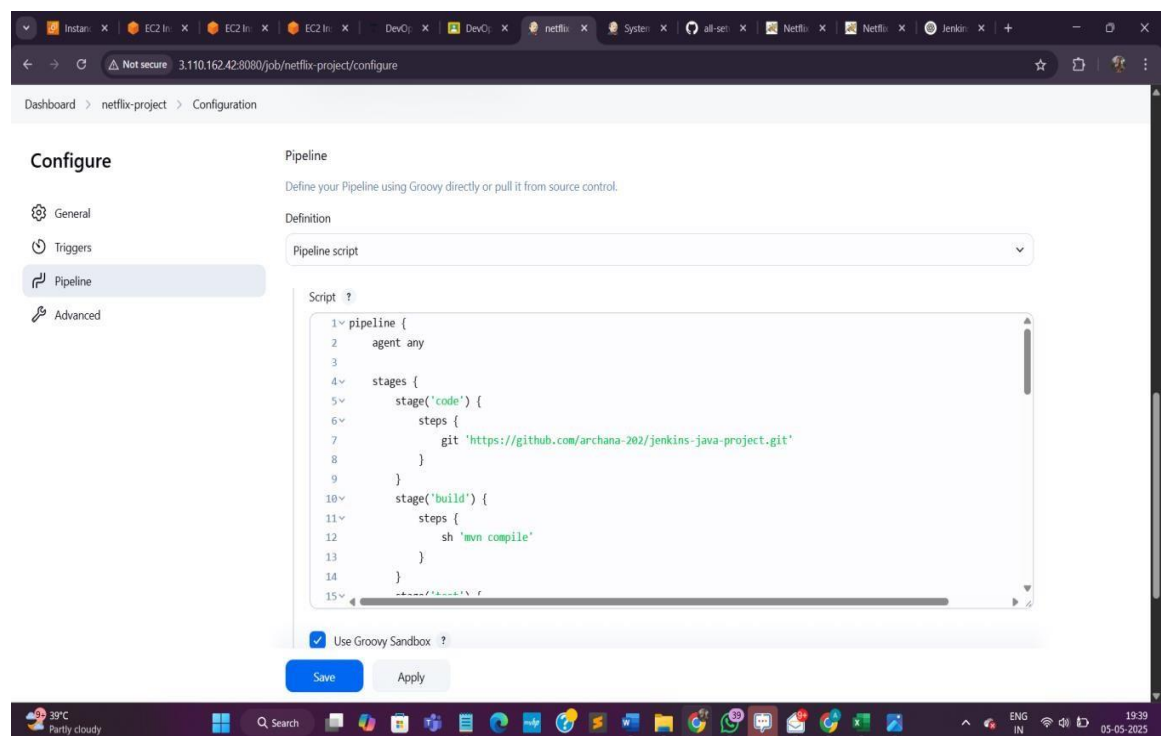
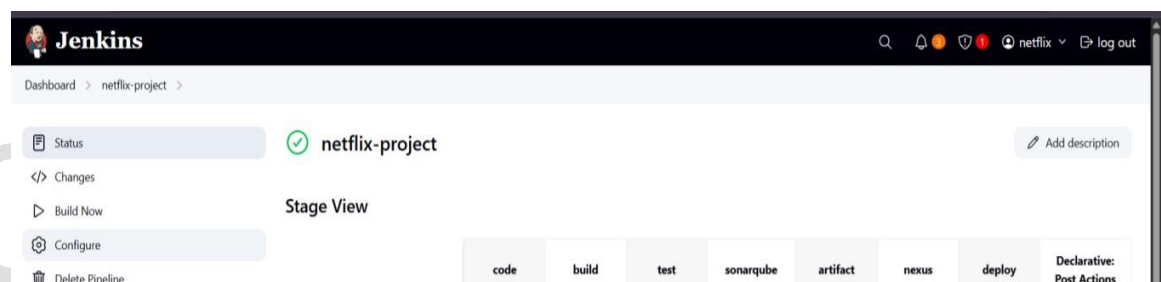
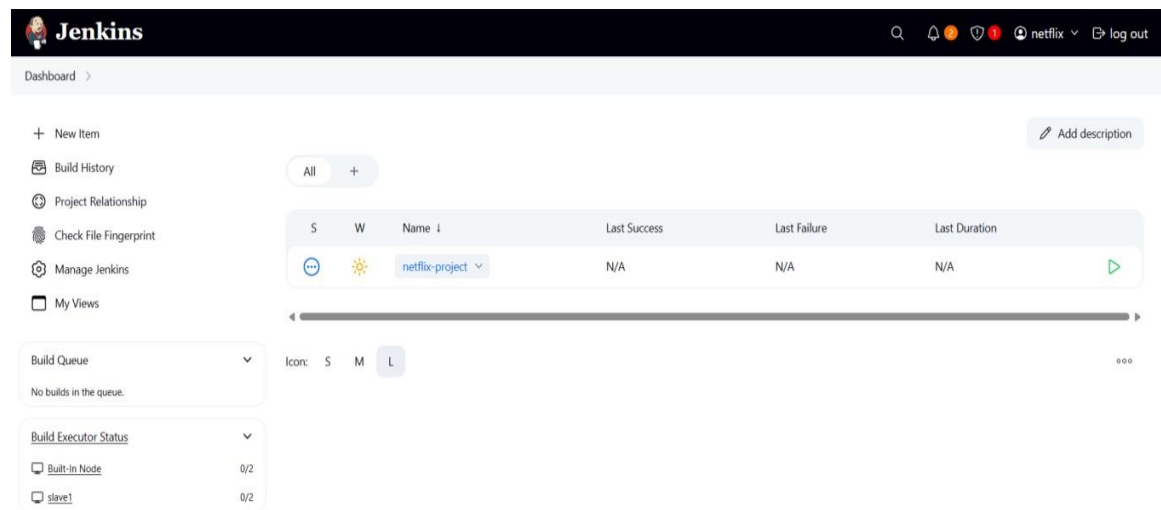
Step 23: Now add the pulg-in

Step 24: Click on Manage Jenkins → Pulgins → select Available Pulgins → select deploy to container & pipeline-stage view.



Step 25: Now, Its time to Add the pipeline.

Step 26: Click on the project → Click on Configure → Scroll down to Pipeline → and paste the script.



SCRIPT:

```
pipeline {
  agent any

  stages {
    stage('code') {
      steps {
        git branch: 'main', url: 'https://github.com/confideoit/Confideo-Netflix-Jenkins-ec2.git'
      }
    }
    stage('build') {
      steps {
        sh 'mvn compile'
      }
    }
    stage('test') {
      steps {
        sh 'mvn test'
      }
    }
    stage('sonarqube') {
      steps {
        echo "my code is tested"
      }
    }
    stage('artifact') {
      steps {
        sh 'mvn clean package'
      }
    }
    stage('nexus') {
      steps {
        echo "my artifact is stored on nexus"
      }
    }
    stage('deploy') {
      steps {
        deploy adapters: [
          tomcat9(credentialsId: '540b11a8-4021-4b9f-a99b-23aceb2217e4', path: '', url:
            'http://3.110.117.33:8080/')
        ],
        contextPath: 'netflix',
        war: 'target/NETFLIX-1.2.2.war'
      }
    }
    post {
```

```
always {  
  echo "my code deployed"  
}  
}  
}
```

Changes required at your end in the script is:

Replace the github url.

Step 27: To generate GITHUB URL → Click on pipeline syntax → select GIT → add the git hub url → click on generate pipeline script → copy and paste that in the script

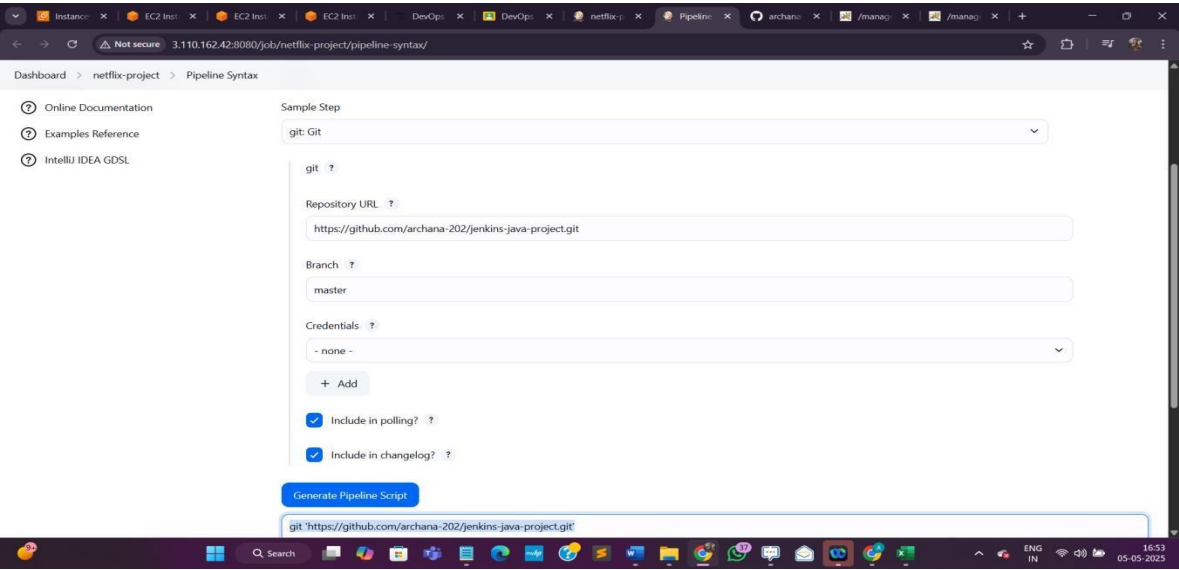
```
stage('code') {  
  steps {  
    git 'https://github.com/archana-202/jenkins-java-project.git'  
  }  
}
```

change the credential id to your generated credential id and the URL of the slave instance

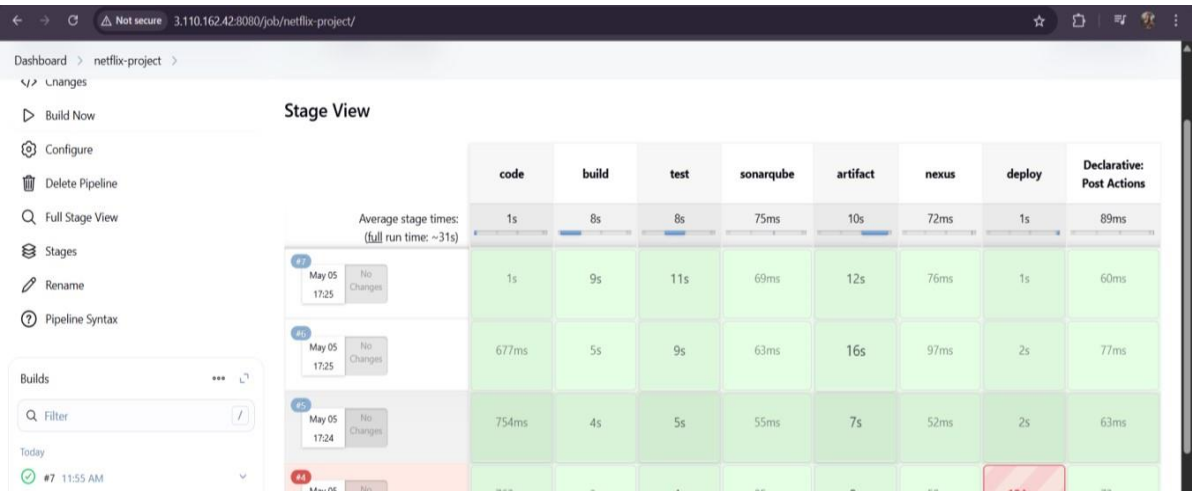
```
stage('deploy') {  
  steps {  
    deploy  
    adapters: [  
      tomcat9(credentialsId: 'ed4edddd-04a5-4027-9a0e-f5f59b6bcd86', path: '', url:  
        'http://13.235.83.112:8080/')  
    ],  
    contextPath: 'netflix',  
    war: 'target/NETFLIX-  
1.2.2.war'
```

Step 29: From github also fork this repo.

<https://github.com/confideoit/Confideo-Netflix-Jenkins-ec2.git>



Step 28: In the End, Click on Build Now.



-----X-----