TECHNOLOGY

Computing

Caltech | Center for Technology & Management Education

Configure AWS for Front End

Powerd by simplilearn

TECHNOLOGY

Computing

Caltech | Center for Technology & Management Education

Set Up AWS for Angular Apps

Powerd by simplilearn

You Already Know

Before we begin, let's recall what we have covered till now:

Agile

Git

SQL

Angular

HTML

CSS

JavaScript

Core Java

You Already Know

Before we begin, let's recall what we have covered till now:

JDBC

JSP

Servlets

MongoDB

Maven

You Already Know

Before we begin, let's recall what we have covered till now:

JUnit

Spring

Spring Boot

Webservices

Microservices

You Already Know

Before we begin, let's recall what we have covered till now:

Docker

Jenkins

AWS

Recap

Front-End Back-End Communication

- Used HTTP Client in Angular to communicate with Java back end
- Implemented communication for both admin and end-user projects

Jenkins Pipeline

- Used Jenkins build automation server to build angular apps and Java back end
- Created Pipeline with multiple stages

Docker

- Used Docker to build images for Angular apps and Java back end
- Used Jenkins to dockerize the Angular apps and Java back end with Jenkins

A Day in the Life of a Full Stack Developer

As a full stack web developer, our key role is to develop both client and server software.

- Angular and Node can be used to build front end of the web page.

- Spring Boot, Java, and MySQL or MongoDB can be used to build at the back end.

A Day in the Life of a Full Stack Developer

Bob needs to configure AWS for front end. He brainstorms a bit and finds a solution.

Let me use Jenkins, Docker, and AWS to build CI/CD Pipelines, containerize the apps, and finally host them to AWS EC2 instance.

In this lesson, we will host the angular apps on EC2 instance, use Jenkins and Docker as DevOps tools, and help Bob complete his task effectively and quickly.

Learning Objectives

By the end of this lesson, you will be able to:

- Create and configure EC2 instance on AWS
- Create S3 bucket
- Use Jenkins on EC2 for CI/CD
- Integrate Docker in Jenkins to build and release images as containers on EC2

Create and Configure EC2 in AWS

AWS

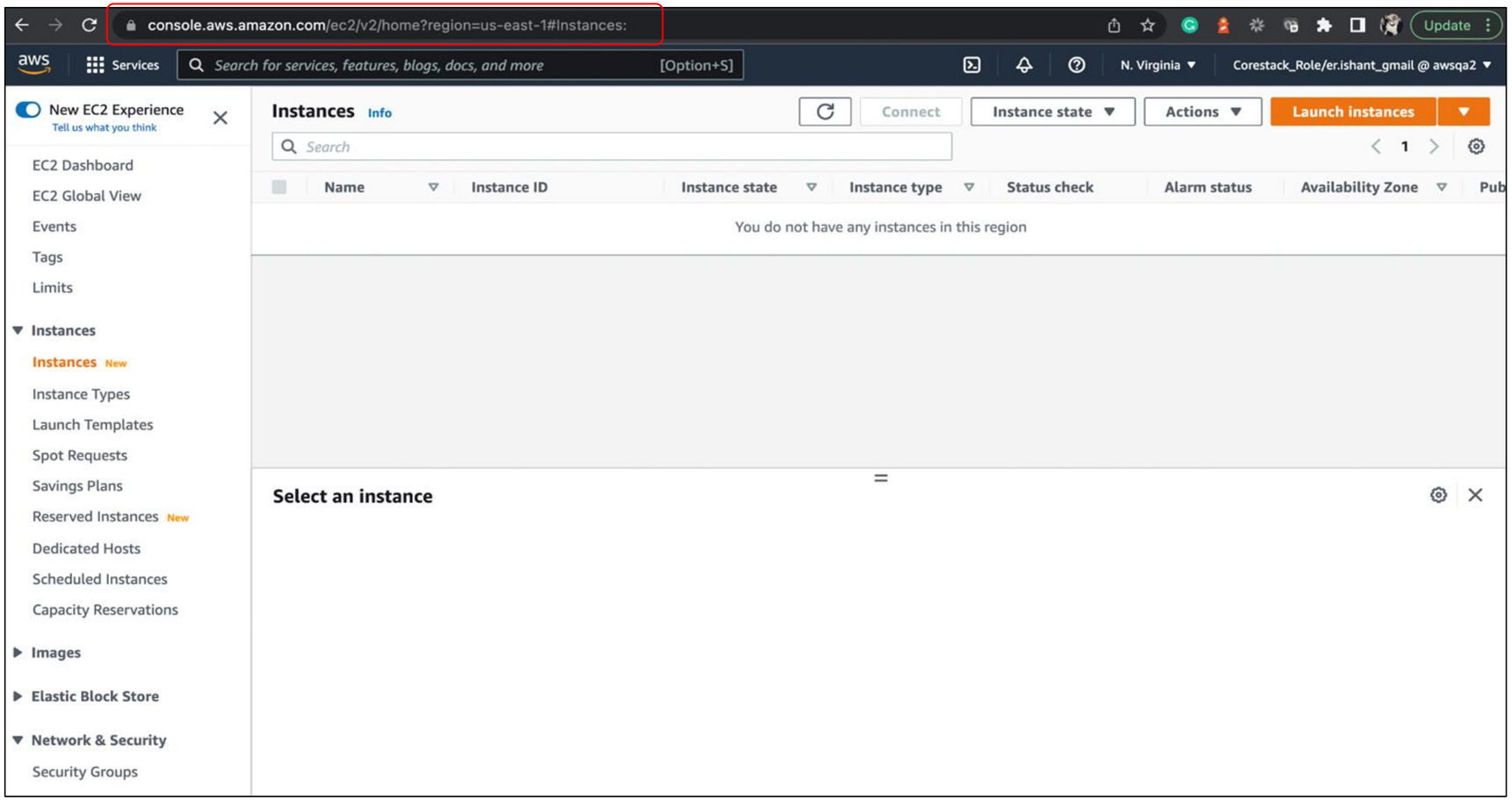
Amazon Web Services (AWS) is a cloud platform, which offers more than 200 fully featured services.

Angular web apps on EC2 instance is deployed using Jenkins and Docker.

.

Create an EC2 Instance

Open the **EC2 dashboard.**



Launch Instance

Launch a new Instance.

console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
FrontEndEC2	i-08bc6e53201548e00	Running	t2.micro	Initializing	No alarms	us-east-1d

Use the AWS Cloud Shell to perform tasks.

Create service group and key pair to access the instance from terminal.

Instance: i-08bc6e53201548e00 (FrontEndEc2)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID: i-08bc6e53201548e00 (FrontEndEc2)

Public IPv4 address: 3.88.180.84 | open address

Private IPv4 addresses: 172.31.28.55

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-88-180-84.compute-1.amazonaws.com | open address

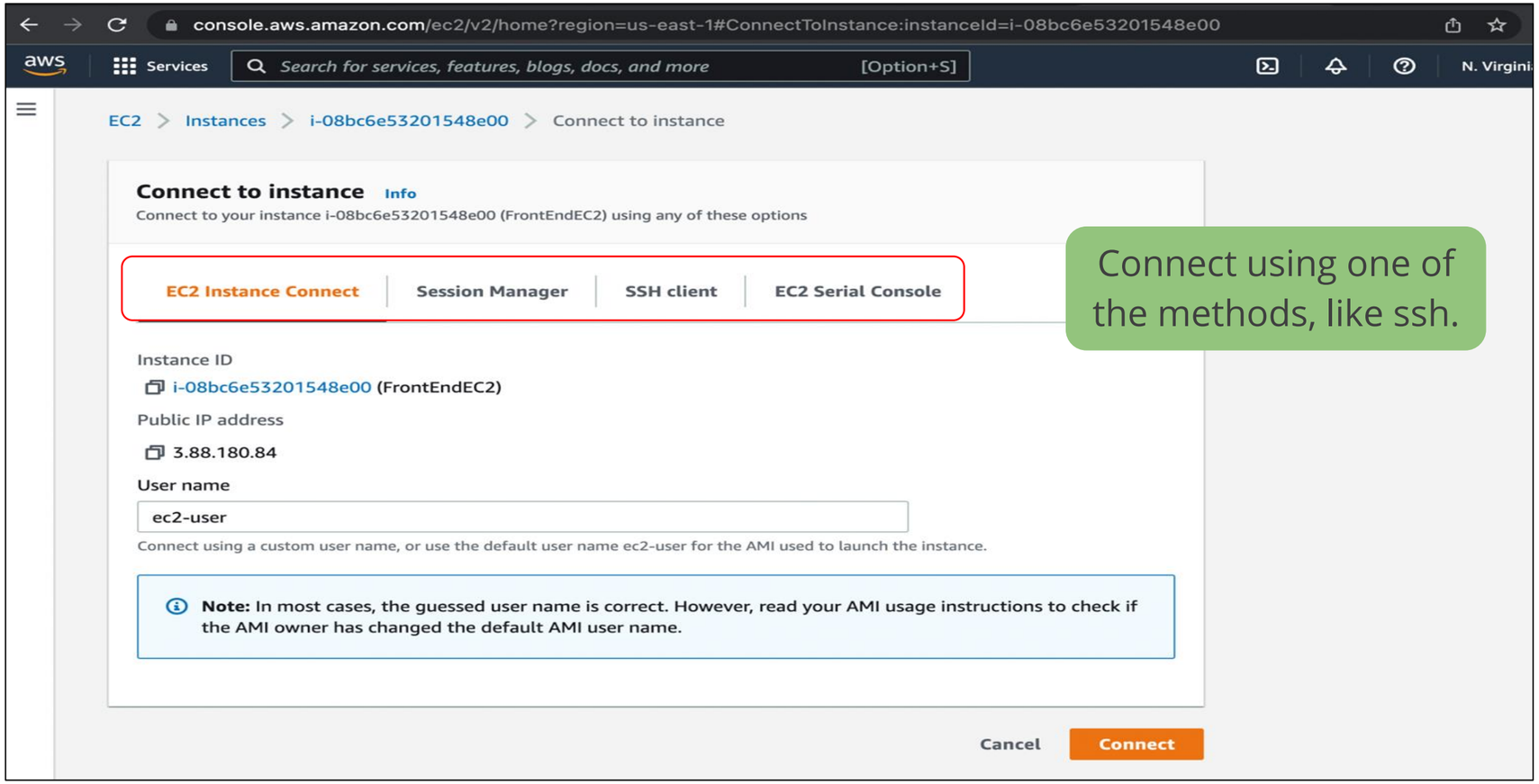
Hostname type: IP name: ip-172-31-28-55.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-28-55.ec2.internal

Answer private resource DNS name: –

Connect to EC2 Instance

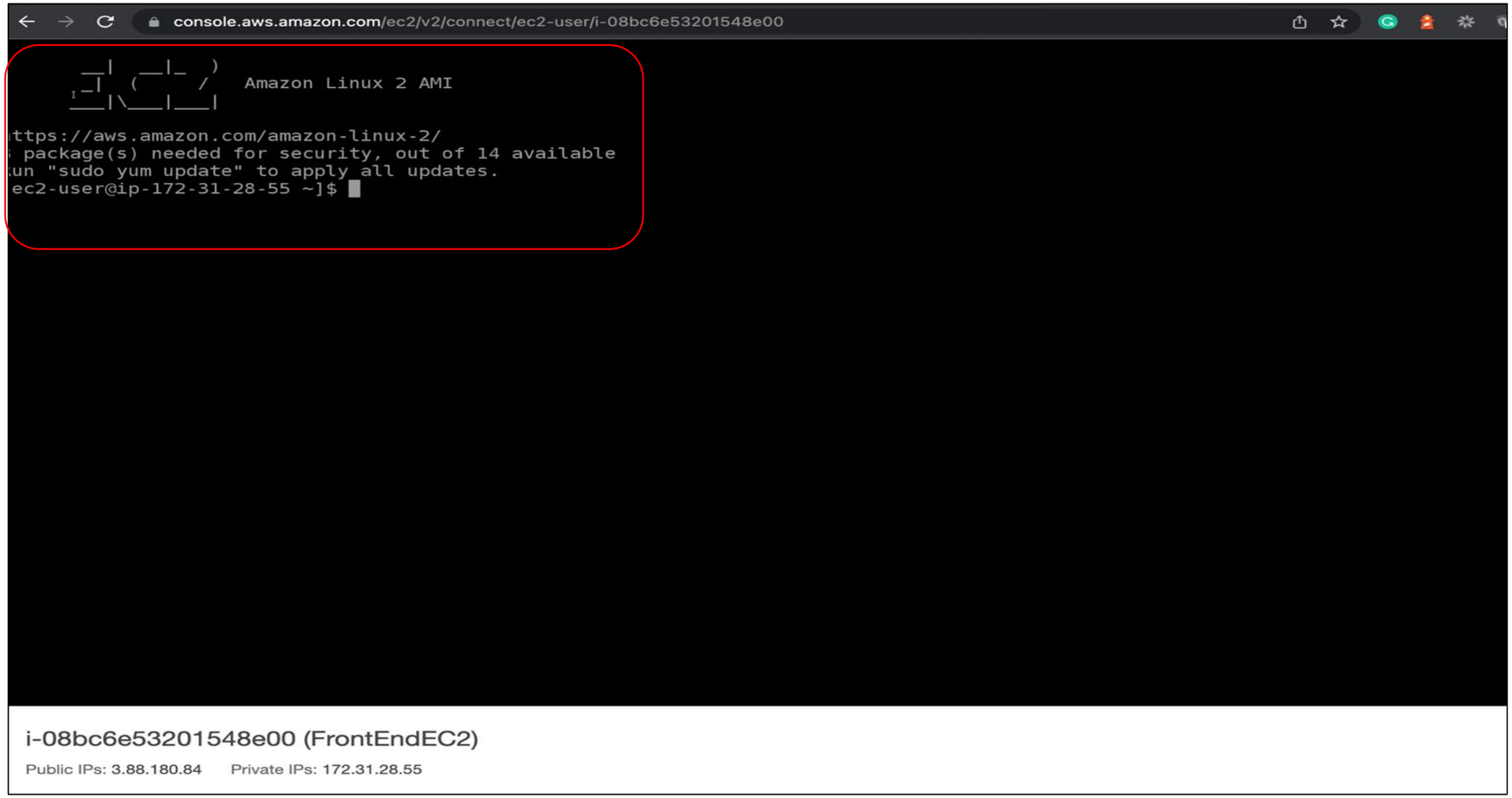
Connect through AWS Cloud Shell, which is the first option on clicking connect that is **EC2 Instance Connect.**



Connect using one of the methods, like ssh.

Launch the EC2 Shell and Configure the Tools

Install the tools and software required to deploy Angular apps and configure Node, Git, Docker, and Jenkins.



Configure Node on EC2

Configure the Jenkins Pipeline Project by passing GitHub repository URL.

Reference link on AWS Documentation: https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/setting-up-node-on-ec2-instance.html

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-08bc6e53201548e00

```
Last login: Sun Feb 27 10:56:27 2022 from ec2-18-206-107-25.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
8 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-28-55 ~]$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 13226  100 13226    0     0   456k      0 --:--:-- --:--:-- --:--:--  461k
=> Downloading nvm as script to '/home/ec2-user/.nvm'

=> Appending nvm source string to /home/ec2-user/.bashrc
=> Appending bash_completion source string to /home/ec2-user/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"  # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"  # This loads nvm bash_completion
[ec2-user@ip-172-31-28-55 ~]$ . ~/.nvm/nvm.sh
[ec2-user@ip-172-31-28-55 ~]$ nvm install node
Downloading and installing node v17.6.0...
Downloading https://nodejs.org/dist/v17.6.0/node-v17.6.0-linux-x64.tar.xz...
################################################################################################################################################# 100.0
Computing checksum with sha256sum
Checksums matched!
Now using node v17.6.0 (npm v8.5.1)
Creating default alias: default -> node (-> v17.6.0)
[ec2-user@ip-172-31-28-55 ~]$
```

Configure Angular CLI on EC2

Add Angular CLI to run the Angular project.

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b2d99ad1267484be

```
Last login: Sun Feb 27 19:47:23 2022 from ec2-18-206-107-25.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-87-187 ~]$ npm install -g @angular/cli

added 188 packages, and audited 189 packages in 16s

23 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New patch version of npm available! 8.5.1 -> 8.5.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.5.2
npm notice Run npm install -g npm@8.5.2 to update!
npm notice
[ec2-user@ip-172-31-87-187 ~]$
```

Configure Git on EC2 Instance

Configure Git.

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-08bc6e53201548e00

```
[ec2-user@ip-172-31-28-55 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-28-55 ~]$ sudo yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package git-2.32.0-1.amzn2.0.1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-28-55 ~]$ git version
git version 2.32.0
[ec2-user@ip-172-31-28-55 ~]$
```

Configure Docker on EC2 Instance

Install docker for containerizing the Angular apps.

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b2d99ad1267484be

```
[ec2-user@ip-172-31-87-187 ~]$ sudo yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
208 packages excluded due to repository priority protections
Package docker-20.10.7-5.amzn2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-87-187 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2022-02-27 20:53:28 UTC; 17min ago
     Docs: https://docs.docker.com
  Process: 3052 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
  Process: 3039 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Main PID: 3057 (dockerd)
    Tasks: 7
   Memory: 42.7M
   CGroup: /system.slice/docker.service
           └─3057 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.581813062Z" level=info msg="[graphdriver
Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.594013561Z" level=warning msg="Your kern
Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.594040114Z" level=warning msg="Your kern
Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.594919996Z" level=info msg="Loading cont
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.247258940Z" level=info msg="Default brid
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.415246900Z" level=info msg="Loading cont
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.560515372Z" level=info msg="Docker daemo
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.562805066Z" level=info msg="Daemon has c
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.689510365Z" level=info msg="API listen o
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-87-187 ~]$
```

Install Docker : sudo yum install docker

Configure Docker on EC2 Instance

Add ec2-user to docker group, which can execute Docker commands without using sudo and exit the terminal. Re-login to make the change effective.
Command: sudo usermod -a -G docker ec2-user

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b2d99ad1267484be

```
[ec2-user@ip-172-31-87-187 ~]$ sudo yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
208 packages excluded due to repository priority protections
Package docker-20.10.7-5.amzn2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-87-187 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2022-02-27 20:53:28 UTC; 17min ago
     Docs: https://docs.docker.com
  Process: 3052 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
  Process: 3039 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Main PID: 3057 (dockerd)
    Tasks: 7
   Memory: 42.7M
   CGroup: /system.slice/docker.service
           └─3057 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.581813062Z" level=info msg="[graphdriver
Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.594013561Z" level=warning msg="Your kern
Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.594040114Z" level=warning msg="Your kern
Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.594919996Z" level=info msg="Loading cont
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.247258940Z" level=info msg="Default brid
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.415246900Z" level=info msg="Loading cont
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.560515372Z" level=info msg="Docker daemo
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.562805066Z" level=info msg="Daemon has c
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.689510365Z" level=info msg="API listen o
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-87-187 ~]$
```

Configure Docker on EC2 Instance

Enable docker service : **sudo systemctl enable docker**
Start docker service : **sudo systemctl start docker**

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b2d99ad1267484be

```
[ec2-user@ip-172-31-87-187 ~]$ sudo yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
208 packages excluded due to repository priority protections
Package docker-20.10.7-5.amzn2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-87-187 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2022-02-27 20:53:28 UTC; 17min ago
     Docs: https://docs.docker.com
  Process: 3052 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
  Process: 3039 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Main PID: 3057 (dockerd)
    Tasks: 7
   Memory: 42.7M
   CGroup: /system.slice/docker.service
           └─3057 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.581813062Z" level=info msg="[graphdriver
Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.594013561Z" level=warning msg="Your kern
Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.594040114Z" level=warning msg="Your kern
Feb 27 20:53:27 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:27.594919996Z" level=info msg="Loading cont
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.247258940Z" level=info msg="Default brid
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.415246900Z" level=info msg="Loading cont
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.560515372Z" level=info msg="Docker daemo
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.562805066Z" level=info msg="Daemon has c
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 27 20:53:28 ip-172-31-87-187.ec2.internal dockerd[3057]: time="2022-02-27T20:53:28.689510365Z" level=info msg="API listen o
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-87-187 ~]$
```

Check the docker service.

Configure Jenkins on EC2 Instance

Configure Jenkins on EC2 Instance. **Install Java** and use these commands.

Install JDK 8 : **sudo yum update –y or sudo yum install java-1.8.0-openjdk**.

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b2d99ad1267484be

```
[ec2-user@ip-172-31-87-187 ~]$ sudo yum install jenkins -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
208 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package jenkins.noarch 0:2.319.3-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                          Arch                          Version                          Repository
================================================================================================================================
Installing:
 jenkins                          noarch                        2.319.3-1.1                      jenkins

Transaction Summary
================================================================================================================================
Install  1 Package

Total download size: 69 M
Installed size: 69 M
Downloading packages:
jenkins-2.319.3-1.1.noarch.rpm                                                                                  |  69 MB  00:00:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.319.3-1.1.noarch
  Verifying  : jenkins-2.319.3-1.1.noarch

Installed:
  jenkins.noarch 0:2.319.3-1.1

Complete!
[ec2-user@ip-172-31-87-187 ~]$
```

Configure Jenkins on EC2 Instance

Install JDK 11 – Java : **sudo amazon-linux-extras install java-openjdk11**.

Install JDK 11 Development – javac : **sudo yum install java-11-openjdk-devel**.

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b2d99ad1267484be

```
[ec2-user@ip-172-31-87-187 ~]$ sudo yum install jenkins -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
208 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package jenkins.noarch 0:2.319.3-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                          Arch                          Version                          Repository
================================================================================================================================
Installing:
 jenkins                          noarch                        2.319.3-1.1                      jenkins

Transaction Summary
================================================================================================================================
Install  1 Package

Total download size: 69 M
Installed size: 69 M
Downloading packages:
jenkins-2.319.3-1.1.noarch.rpm                                                                                  |  69 MB  00:00:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.319.3-1.1.noarch
  Verifying  : jenkins-2.319.3-1.1.noarch

Installed:
  jenkins.noarch 0:2.319.3-1.1

Complete!
[ec2-user@ip-172-31-87-187 ~]$
```

Configure Jenkins on EC2 Instance

Configure Java Version 8 or 11 : **sudo alternatives --config java**.

Read OS name : **cat /etc/os-release**

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b2d99ad1267484be

```
[ec2-user@ip-172-31-87-187 ~]$ sudo yum install jenkins -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
208 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package jenkins.noarch 0:2.319.3-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                          Arch                          Version                          Repository
================================================================================================================================
Installing:
 jenkins                          noarch                        2.319.3-1.1                      jenkins

Transaction Summary
================================================================================================================================
Install  1 Package

Total download size: 69 M
Installed size: 69 M
Downloading packages:
jenkins-2.319.3-1.1.noarch.rpm                                                                                  |  69 MB  00:00:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.319.3-1.1.noarch
  Verifying  : jenkins-2.319.3-1.1.noarch

Installed:
  jenkins.noarch 0:2.319.3-1.1

Complete!
[ec2-user@ip-172-31-87-187 ~]$
```

Configure Jenkins on EC2 Instance

Jenkins Installation document: [https://www.jenkins.io/doc/book/installing/linux/].
Follow the instructions.

sudo amazon-linux-extras install epel
sudo yum-config-manager --enable epel
sudo yum install daemonize -y

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b2d99ad1267484be

```
[ec2-user@ip-172-31-87-187 ~]$ sudo yum install jenkins -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
208 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package jenkins.noarch 0:2.319.3-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                    Arch                    Version                    Repository
================================================================================================================================
Installing:
 jenkins                    noarch                  2.319.3-1.1                jenkins

Transaction Summary
================================================================================================================================
Install  1 Package

Total download size: 69 M
Installed size: 69 M
Downloading packages:
jenkins-2.319.3-1.1.noarch.rpm                                                                          |  69 MB  00:00:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.319.3-1.1.noarch
  Verifying  : jenkins-2.319.3-1.1.noarch

Installed:
  jenkins.noarch 0:2.319.3-1.1

Complete!
[ec2-user@ip-172-31-87-187 ~]$
```

Configure Jenkins on EC2 Instance

```
sudo yum install jenkins -y
sudo systemctl daemon-reload
sudo systemctl enable jenkins
sudo systemctl start jenkins
sudo systemctl status jenkins
```

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b2d99ad1267484be

```
[ec2-user@ip-172-31-87-187 ~]$ sudo yum install jenkins -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
208 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package jenkins.noarch 0:2.319.3-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                          Arch                          Version                          Repository
================================================================================================================================
Installing:
 jenkins                          noarch                        2.319.3-1.1                      jenkins

Transaction Summary
================================================================================================================================
Install  1 Package

Total download size: 69 M
Installed size: 69 M
Downloading packages:
jenkins-2.319.3-1.1.noarch.rpm                                                                                  |  69 MB  00:00:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.319.3-1.1.noarch
  Verifying  : jenkins-2.319.3-1.1.noarch

Installed:
  jenkins.noarch 0:2.319.3-1.1

Complete!
[ec2-user@ip-172-31-87-187 ~]$
```

Configure Jenkins on EC2

Edit the inbound rules of security group associated with EC2 instance.

console.aws.amazon.com/ec2/v2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-02326a39c3dd3f1a6

Services | Search for services, features, blogs, docs, and more [Option+S] | N. Virginia | Corestack_Role/er.ishant_gmail @ aws

EC2 > Security Groups > sg-02326a39c3dd3f1a6 - launch-wizard-1 > Edit inbound rules

Edit inbound rules Info

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

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0786ae8ec9f00898a	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-0d6217835c899c230	Custom TCP	TCP	8080	Custom	0.0.0.0/0		Delete

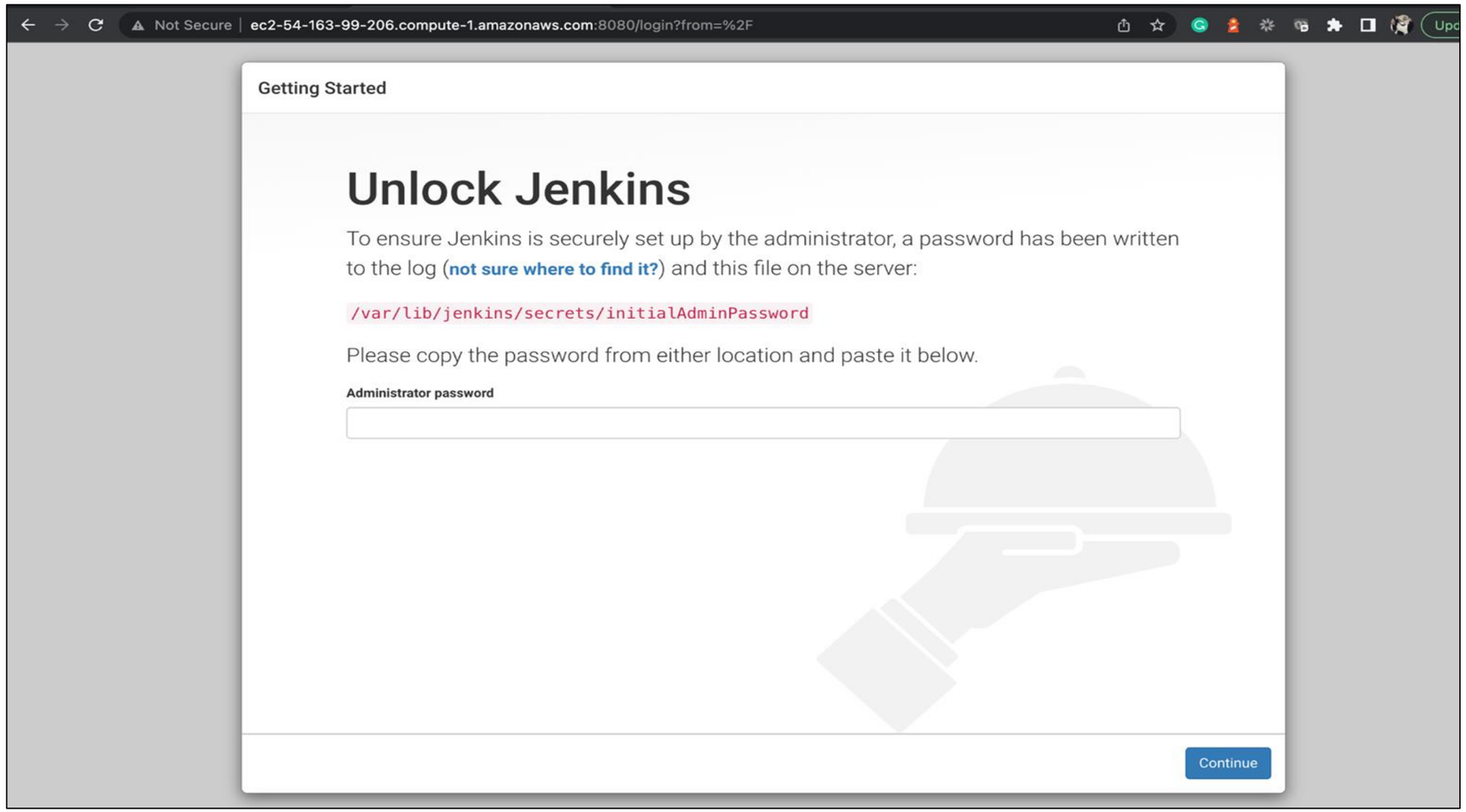
Add rule

Open port 8080 so that Jenkins runs.

Cancel | Preview changes | Save rules

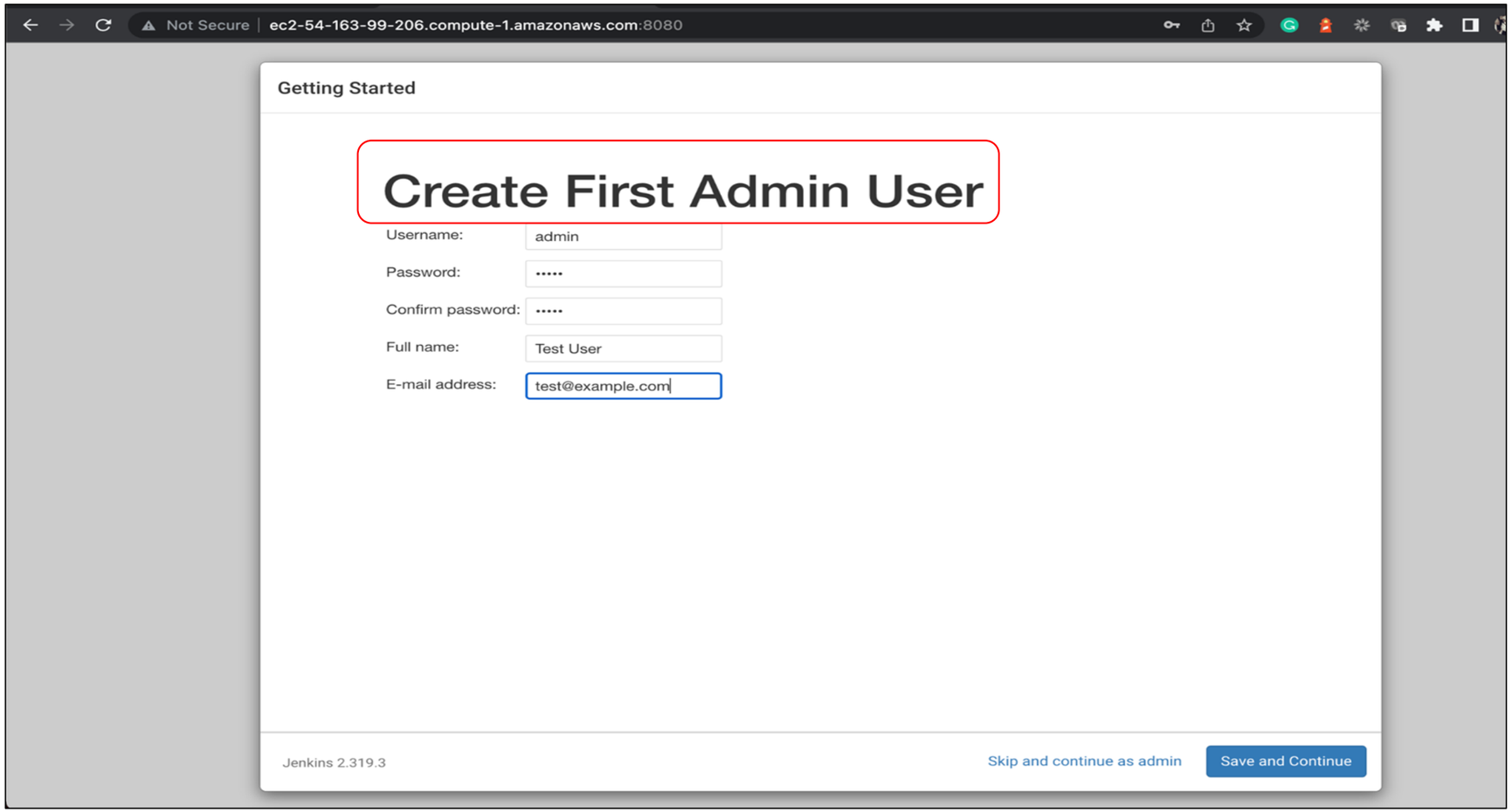
Access Jenkins on Port 8080 of EC2

From here, proceed by **logging** into the Jenkins with the initial password.



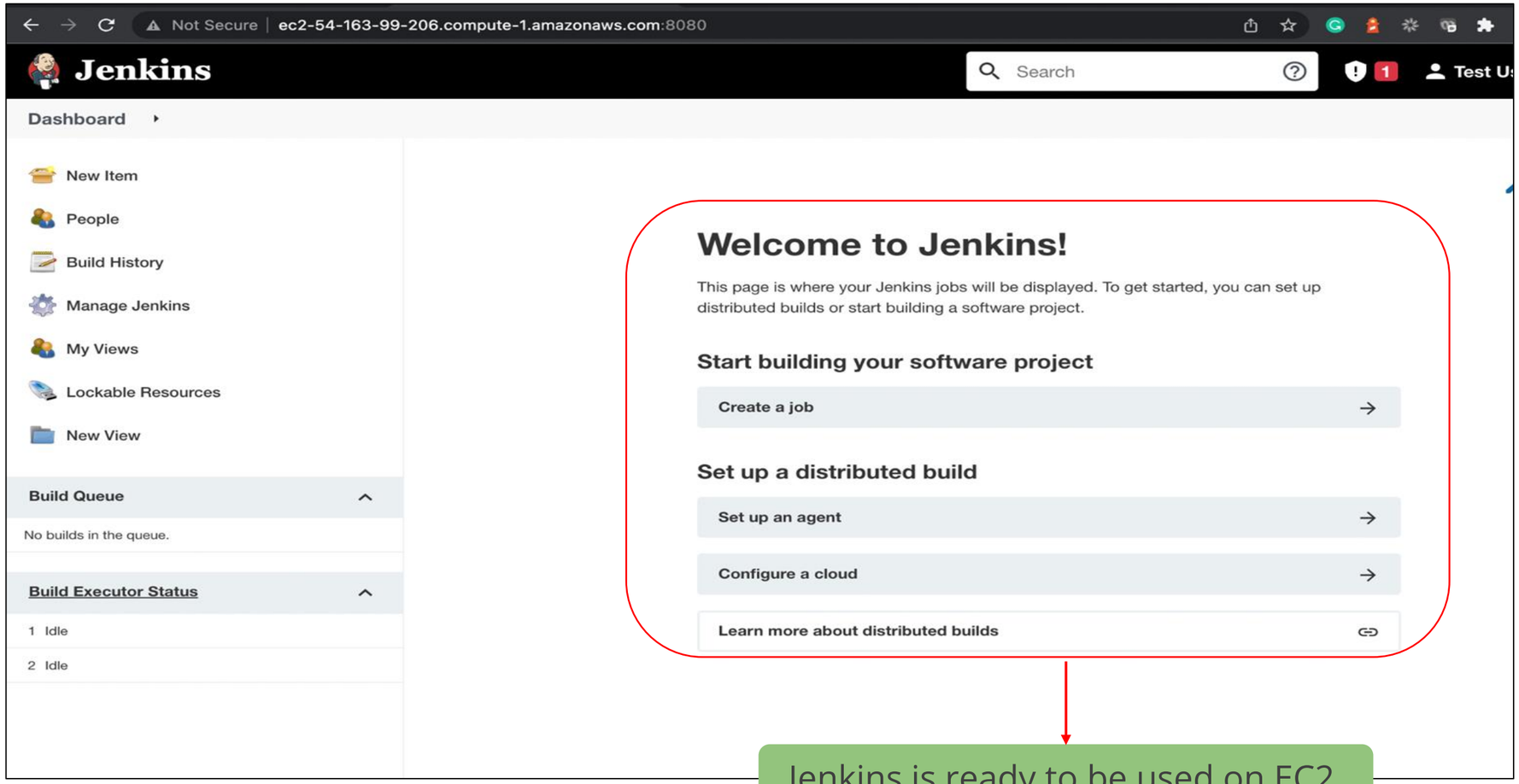
Access Jenkins on Port 8080 of EC2

Create a **new user.**



Configure Jenkins on EC2

Install desired plugins, e.g., NodeJS Plugin to be used for npm command on EC2.

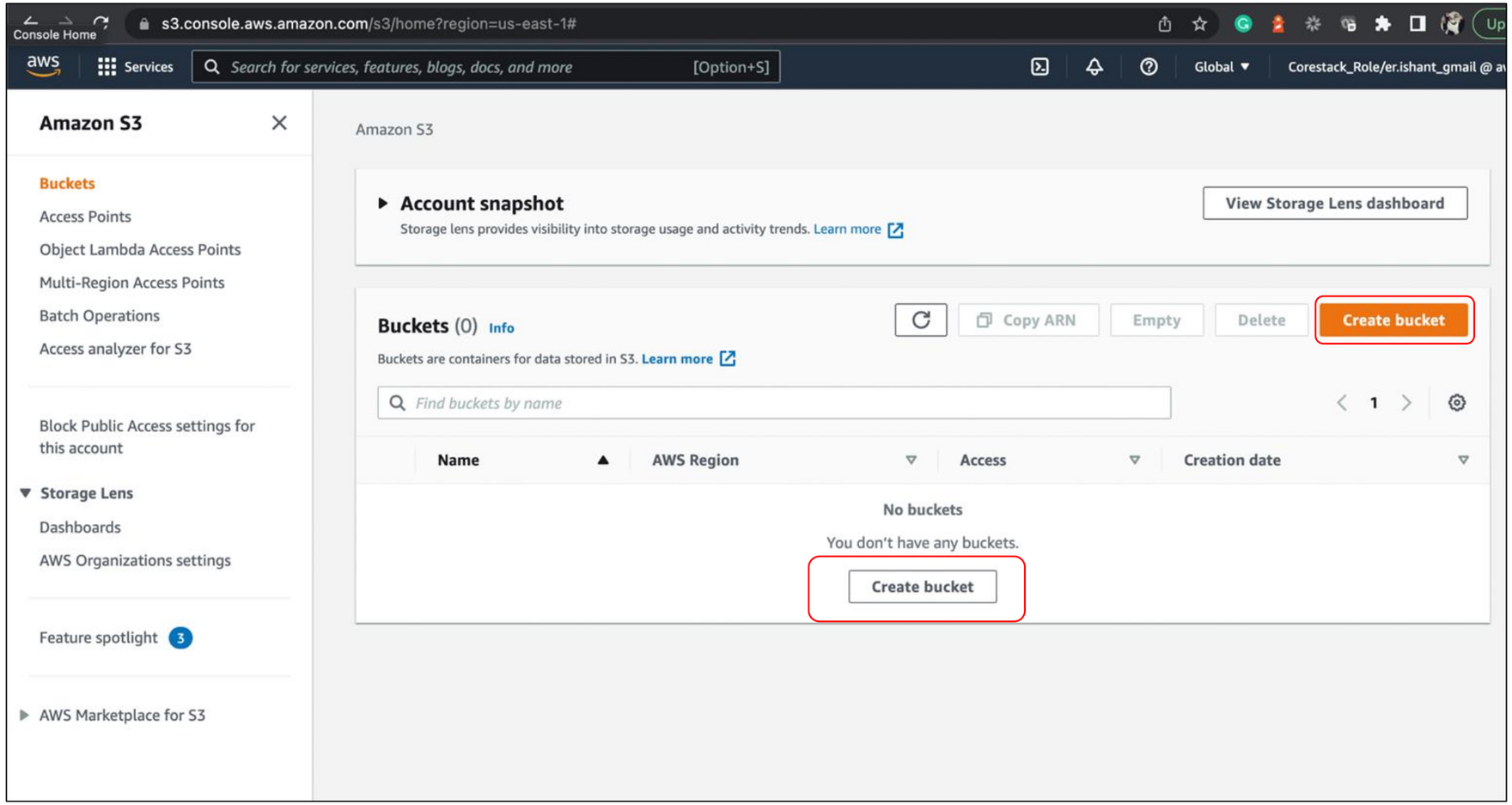


Jenkins is ready to be used on EC2.

Create AWS S3 Bucket

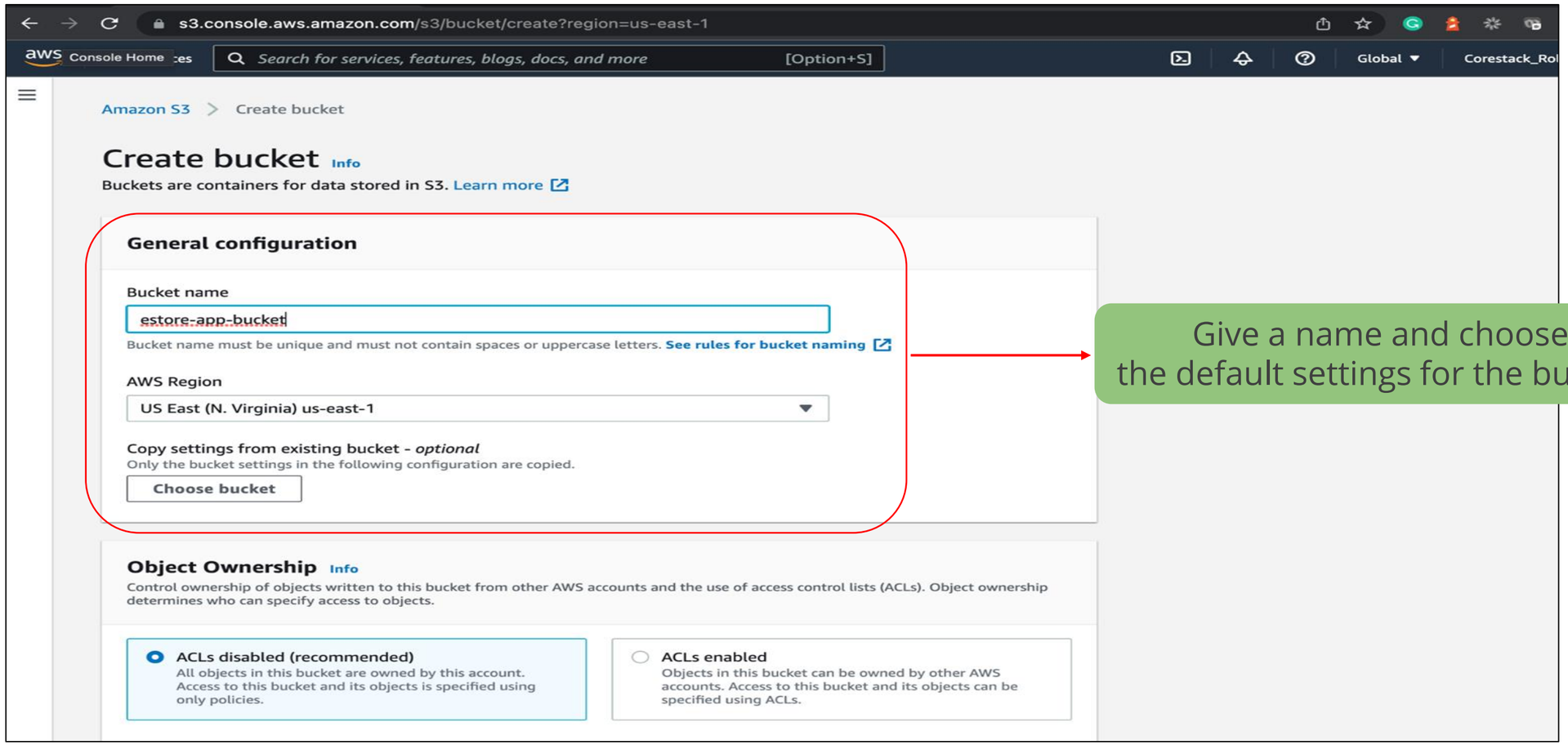
Create S3 Bucket

From the S3 dashboard, click on **Create bucket.**



Create S3 Bucket

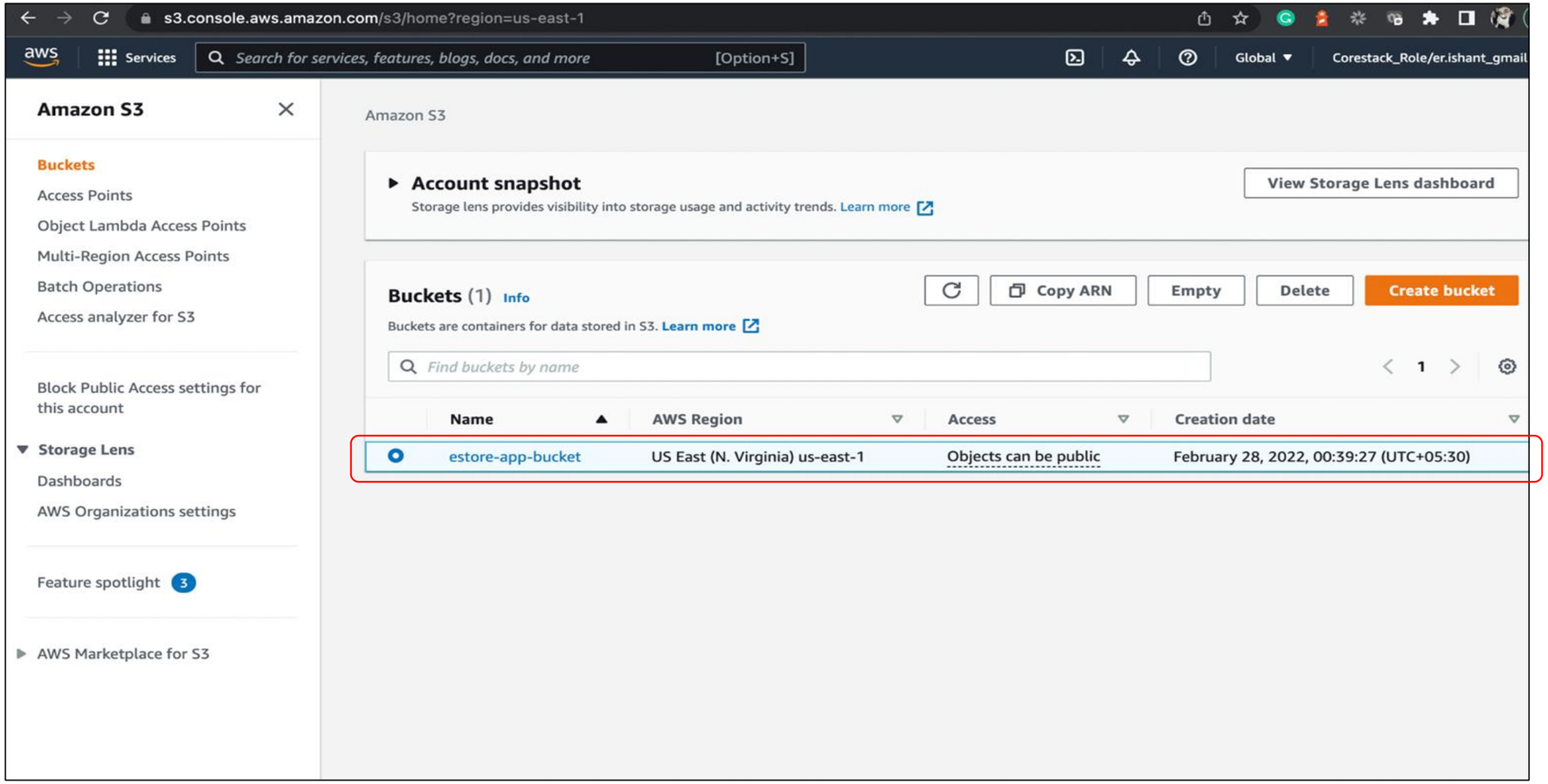
Make bucket accessible to public for the images to be read or written by Angular apps.



Give a name and choose the default settings for the bucket.

Create S3 Bucket

The bucket is **configured successfully.**



TECHNOLOGY

Computing

Caltech | Center for Technology & Management Education

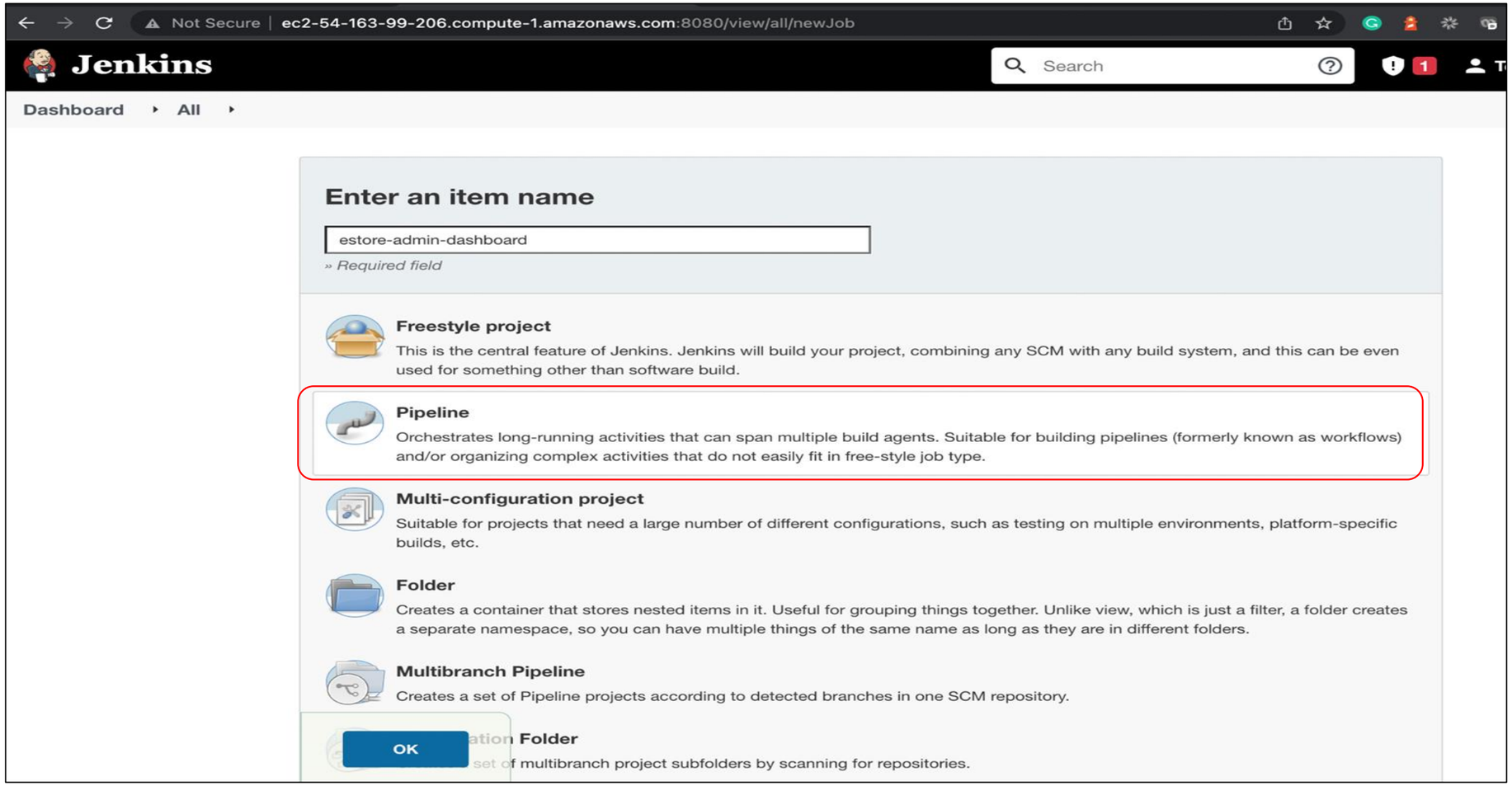
Build and Deploy with Jenkins Docker

Powerd by simplilearn

Set Up Jenkins for the Angular Web-Admin Project

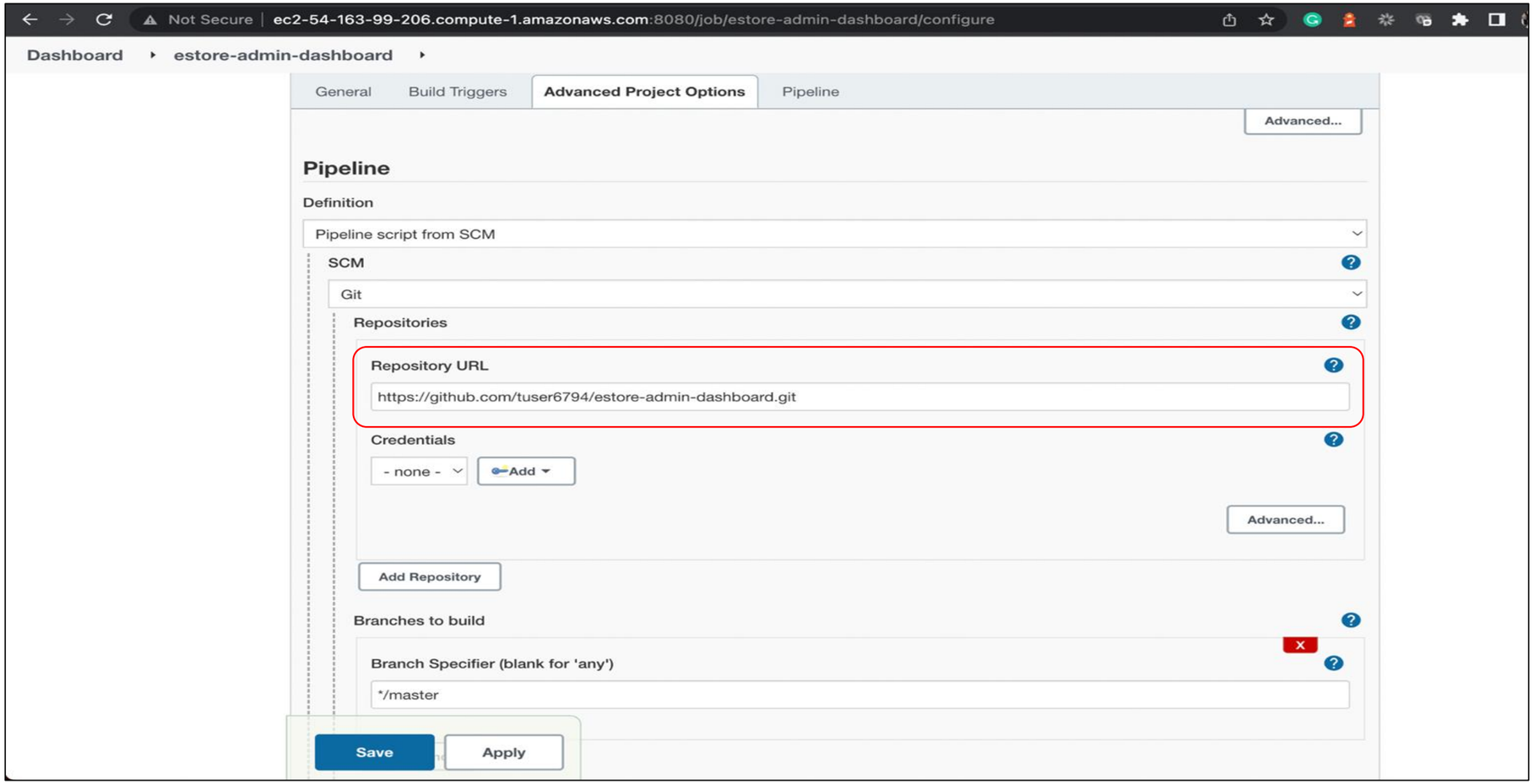
Create a Jenkins Pipeline Project for the Web-Admin Project

Create a new project in Jenkins of type **Pipeline.**



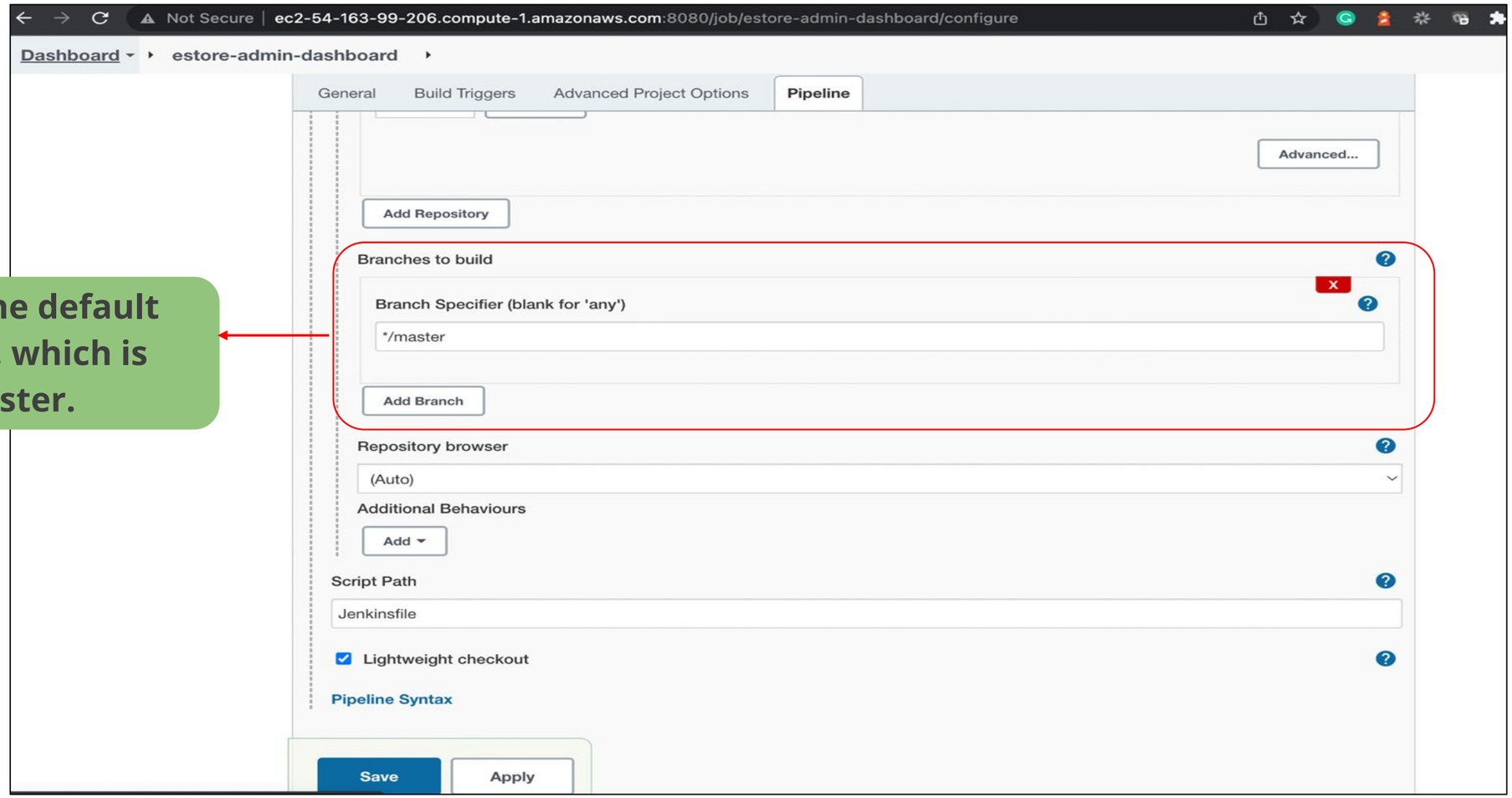
Configure Jenkins Pipeline SCM for the Web-Admin Project

Configure the Jenkins Pipeline Project by passing **GitHub repository URL**.



Configure Jenkinsfile in Jenkins for the Web-Admin Project

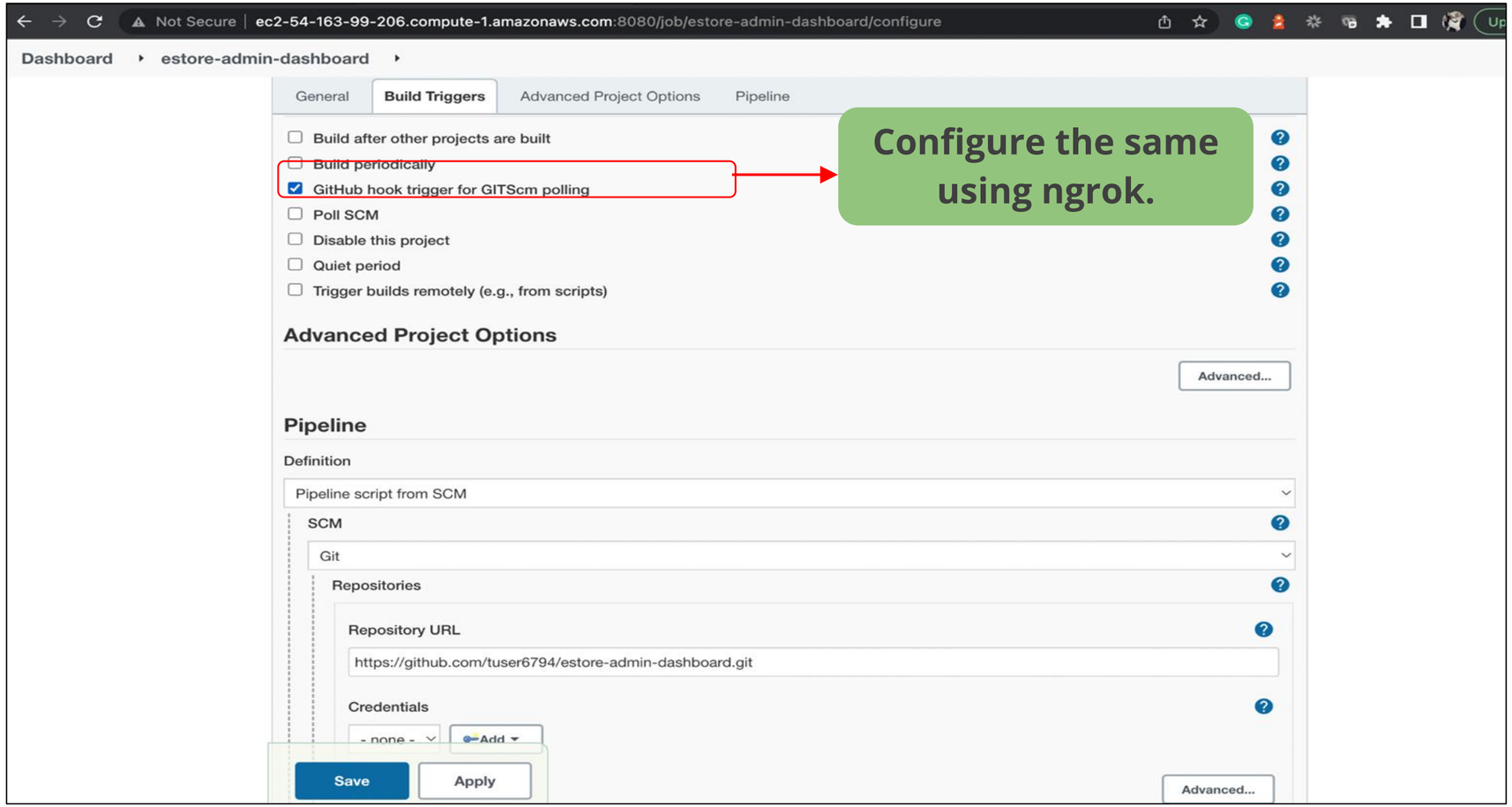
Script Path contains the Jenkinsfile, which is created in the root directory of our project.



Select the default branch, which is master.

Configure GitHub Hook Trigger for the Web-Admin Project

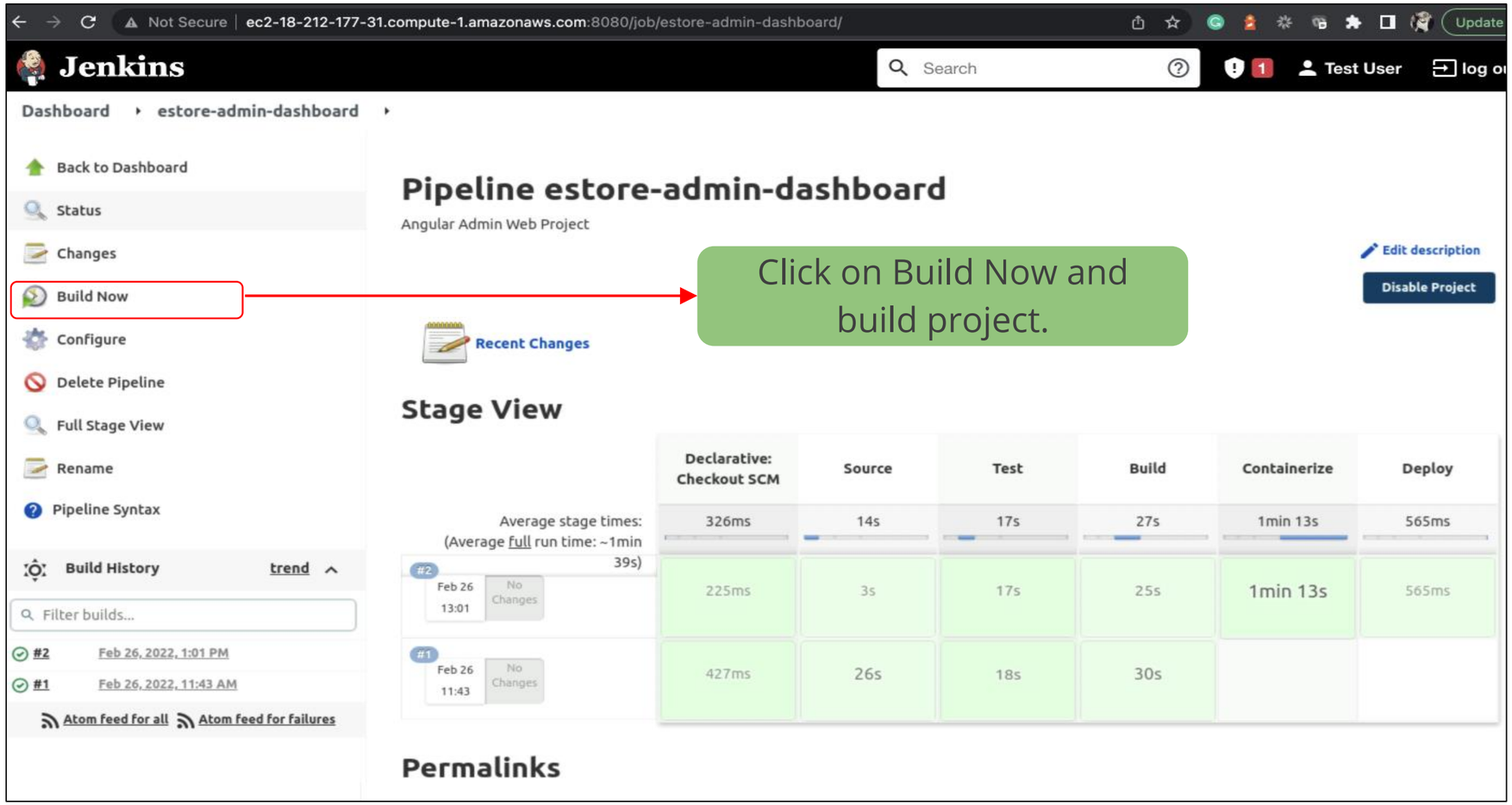
Configure GitHub trigger for GITScm. Polling and option will work when Jenkins is running with a proper URL instead of localhost.



Configure the same using ngrok.

Run the Jenkins Pipeline Project for the Web-Admin Project

Notice stages appearing as mentioned in the Jenkinsfile.
Source > Test > Build > Containerize > Deploy

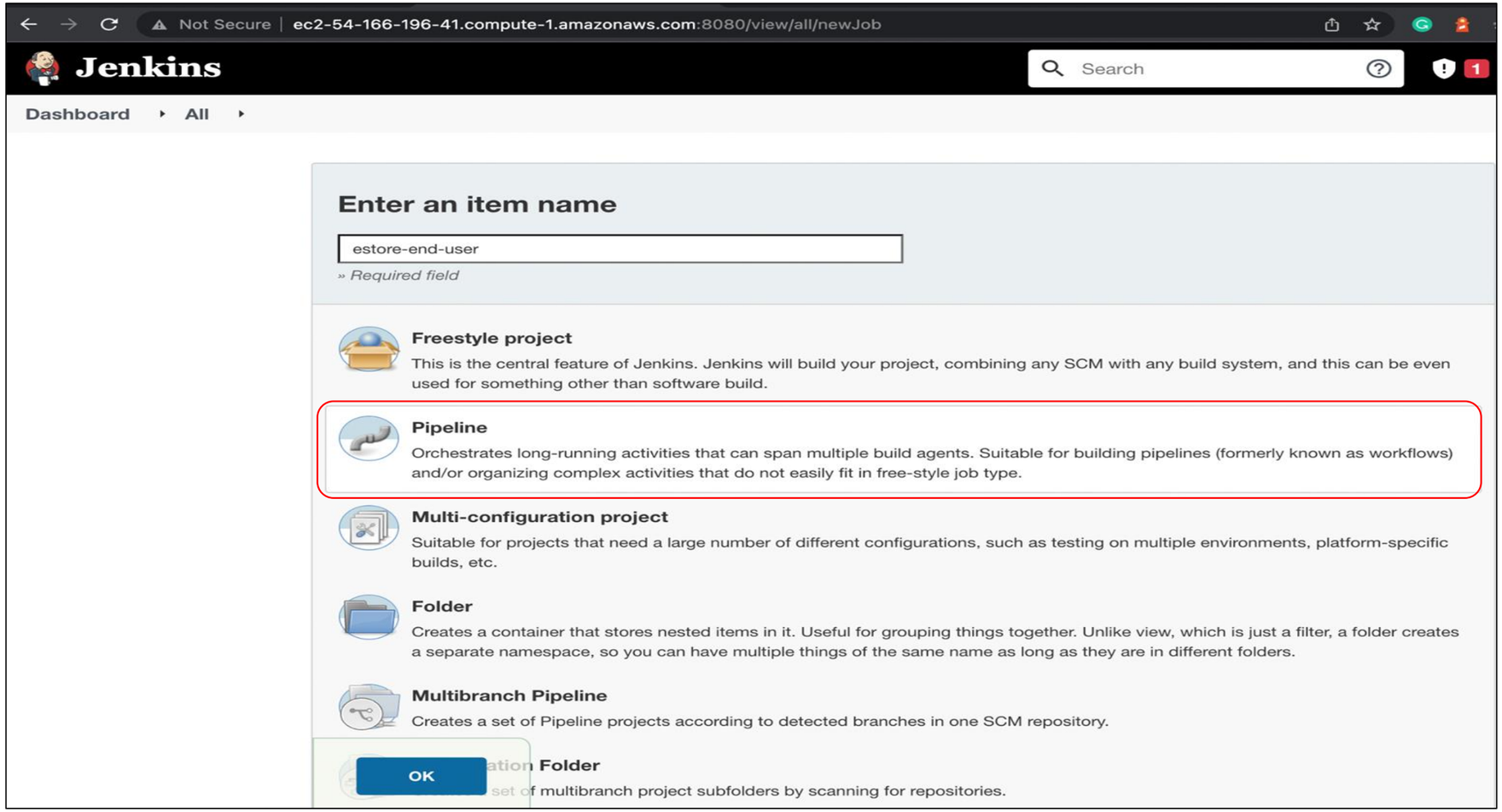


Click on Build Now and build project.

Set Up Jenkins for the Angular End-User Project

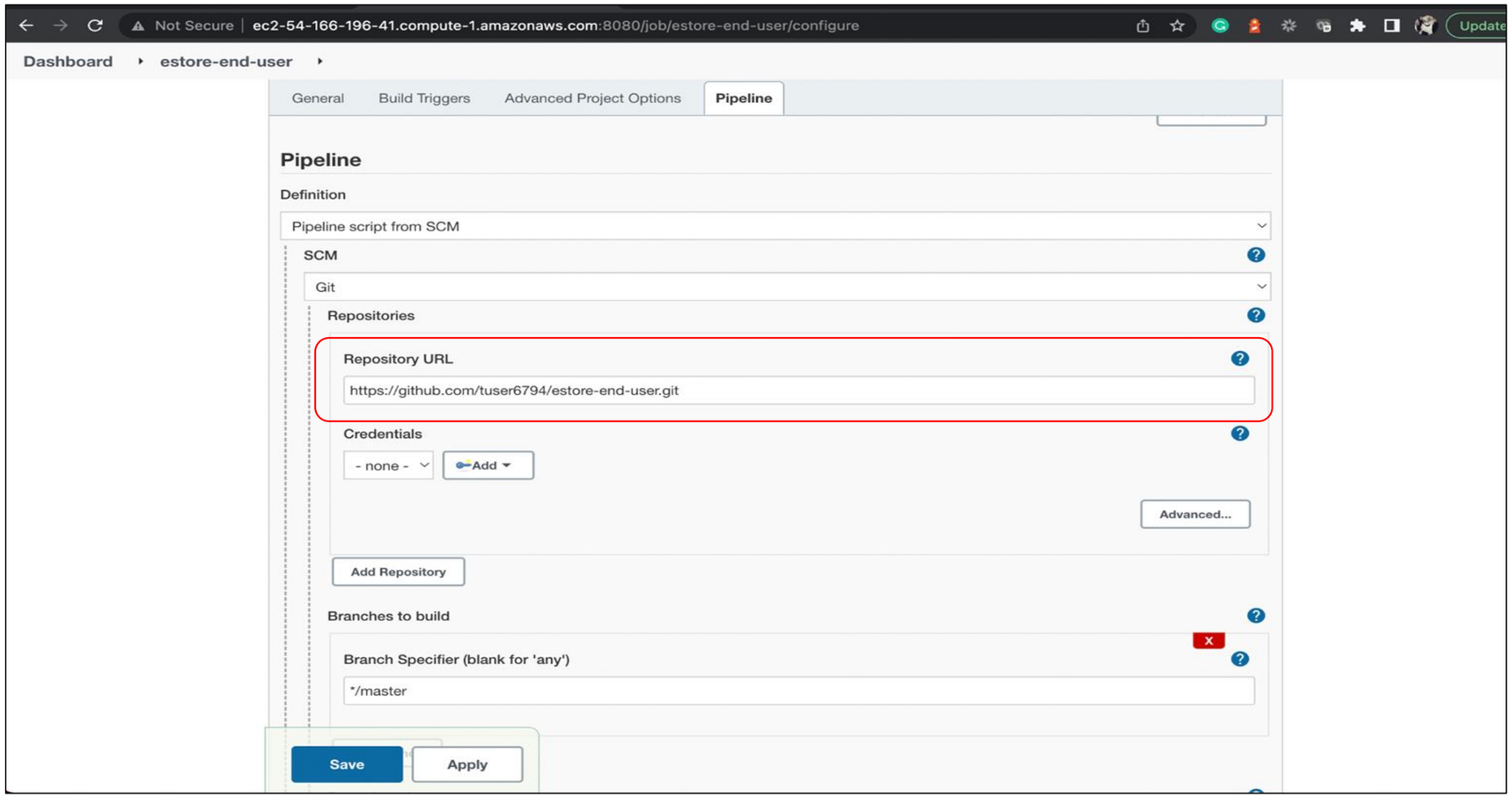
Create a Jenkins Pipeline Project for the End-User Project

Create a new project in Jenkins of type **Pipeline**.



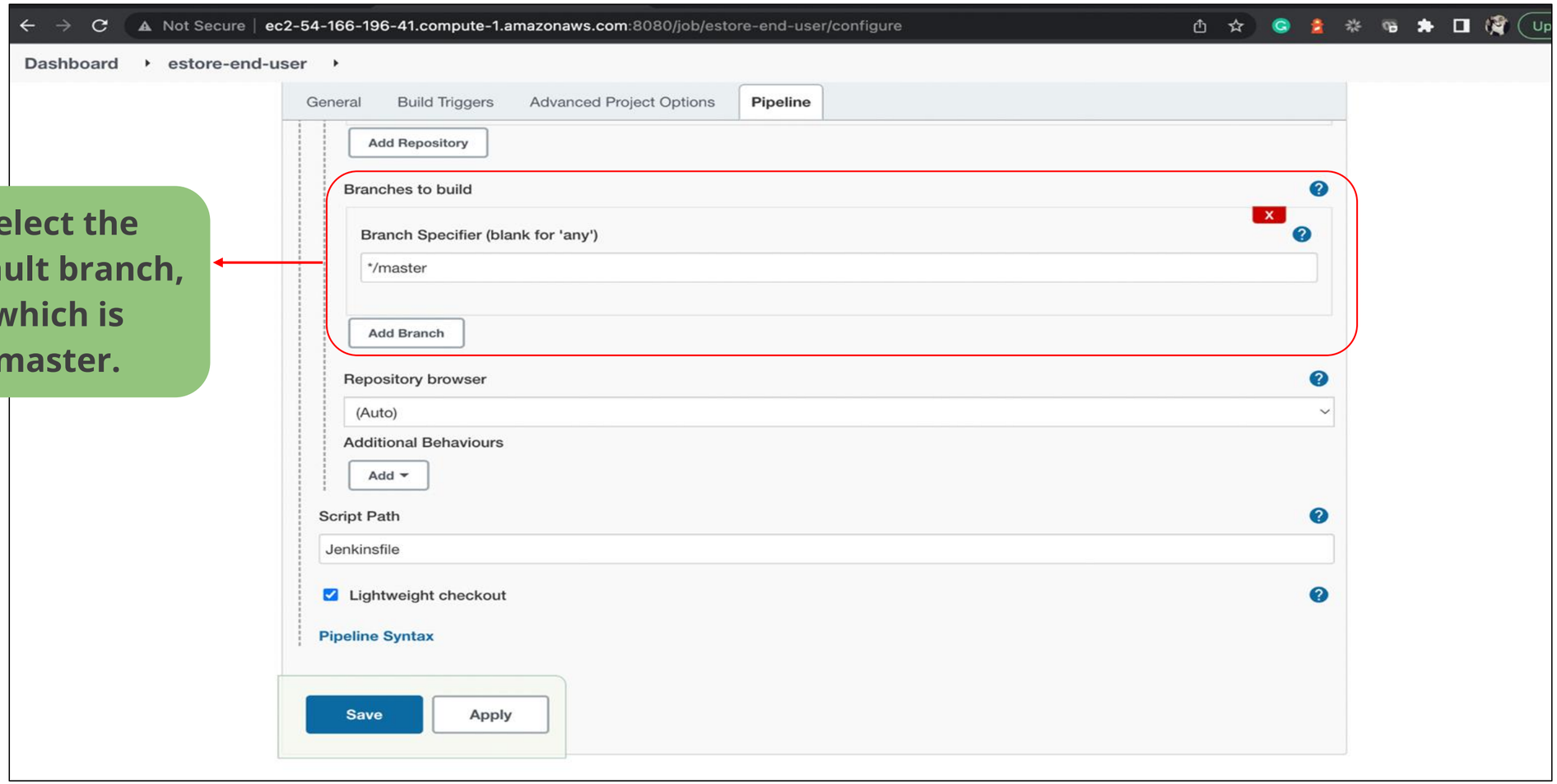
Configure Jenkins Pipeline SCM for the End-User Project

Configure the Jenkins Pipeline Project by passing **GitHub repository URL**.



Configure Jenkinsfile in Jenkins for the End-User Project

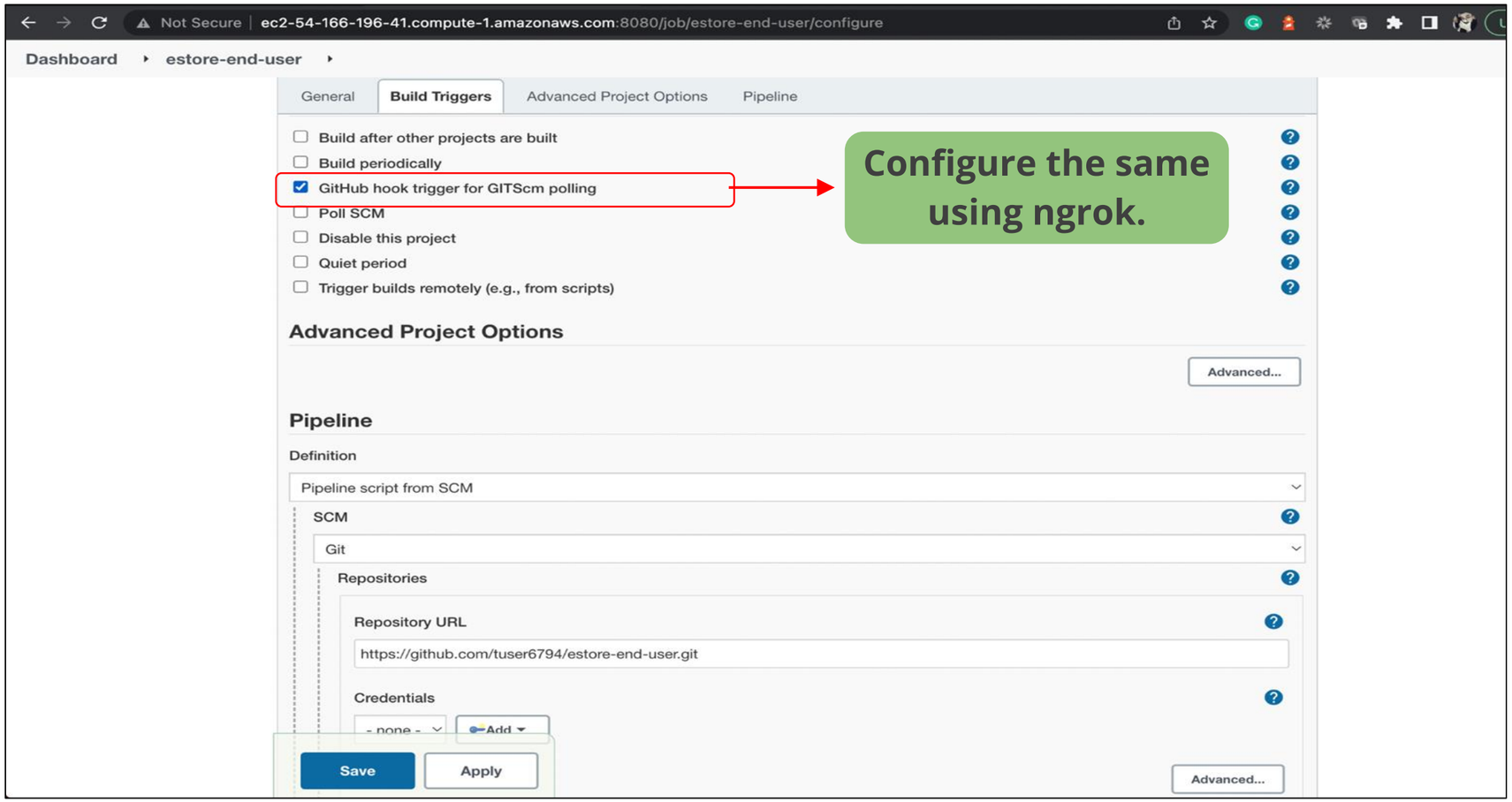
Script Path contains the Jenkinsfile, which is created in the root directory of our project.



Select the default branch, which is master.

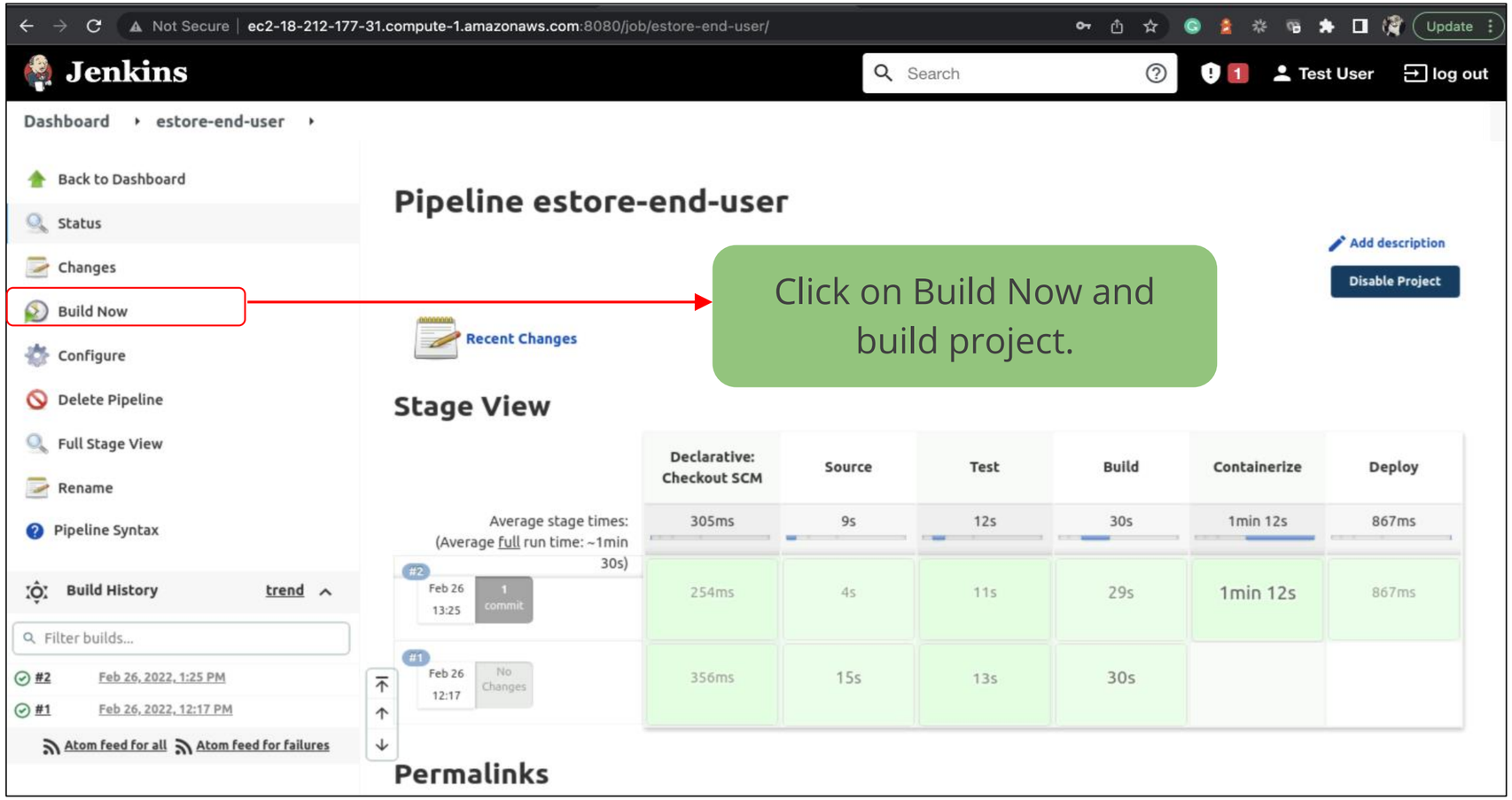
Optional: GitHub Hook Trigger for the End-User Project

Configure **GitHub trigger** for GITScm. Polling and option will work when Jenkins is running with a proper URL instead of localhost.



Run the Jenkins Pipeline Project for the End-User Project

Notice stages appearing as mentioned in the Jenkinsfile.
Source > Test > Build > Containerize > Deploy



Click on Build Now and build project.

Key Takeaways

- EC2 instance is set up in AWS.
- Tools and dev kits are configured on EC2 instance.
- Jenkins Pipeline is built to dockerize the Angular apps.
- Angular apps are made live.

Before the Next Class

You have successfully completed this session.

For our next end user module discussion, you should:

- Review AWS services
- Explore steps to create EC2 and S3 buckets
- Explore how to connect to EC2 instance
- Review Jenkins
- Review Docker commands

What’s Next?

Now we have finished our classes and design Pattern for the back-end project with respect to end-user module.

In our next live session, we will:

- See how to create EC2 Instance on AWS
- See how to work with RDS MySQL
- Deploy Java back end on EC2
- Use Jenkins and Docker as DevOps tools on EC2