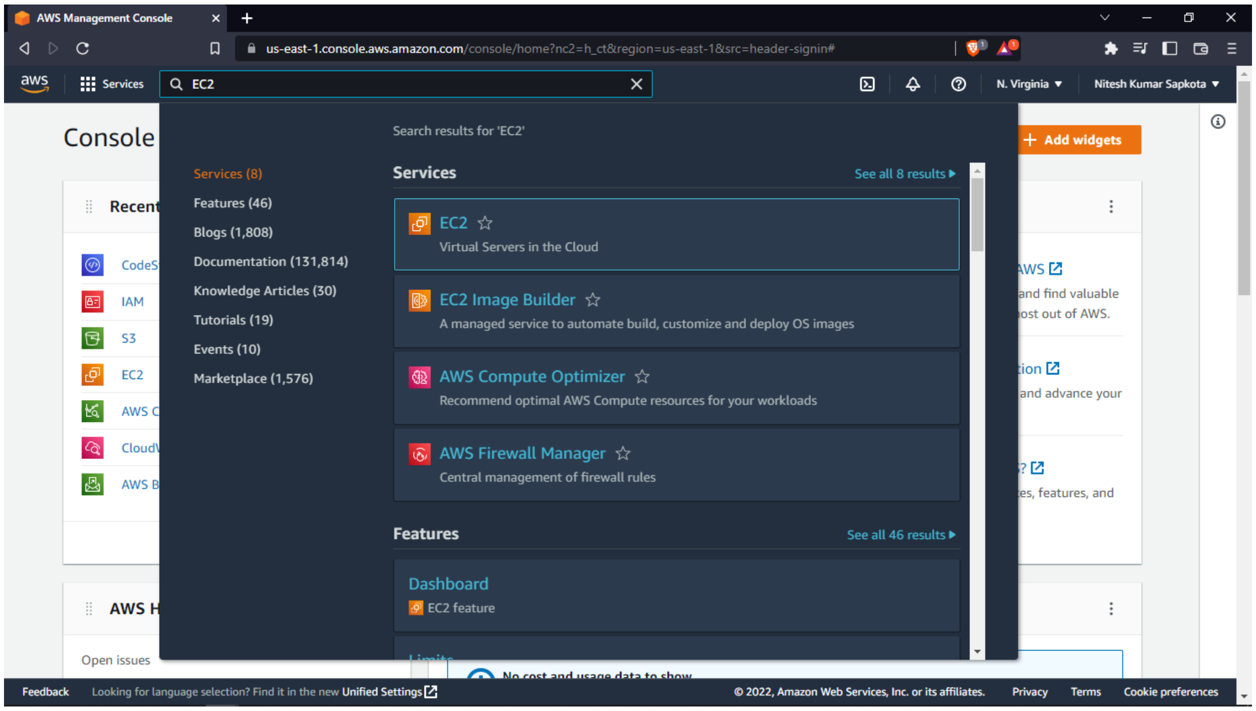
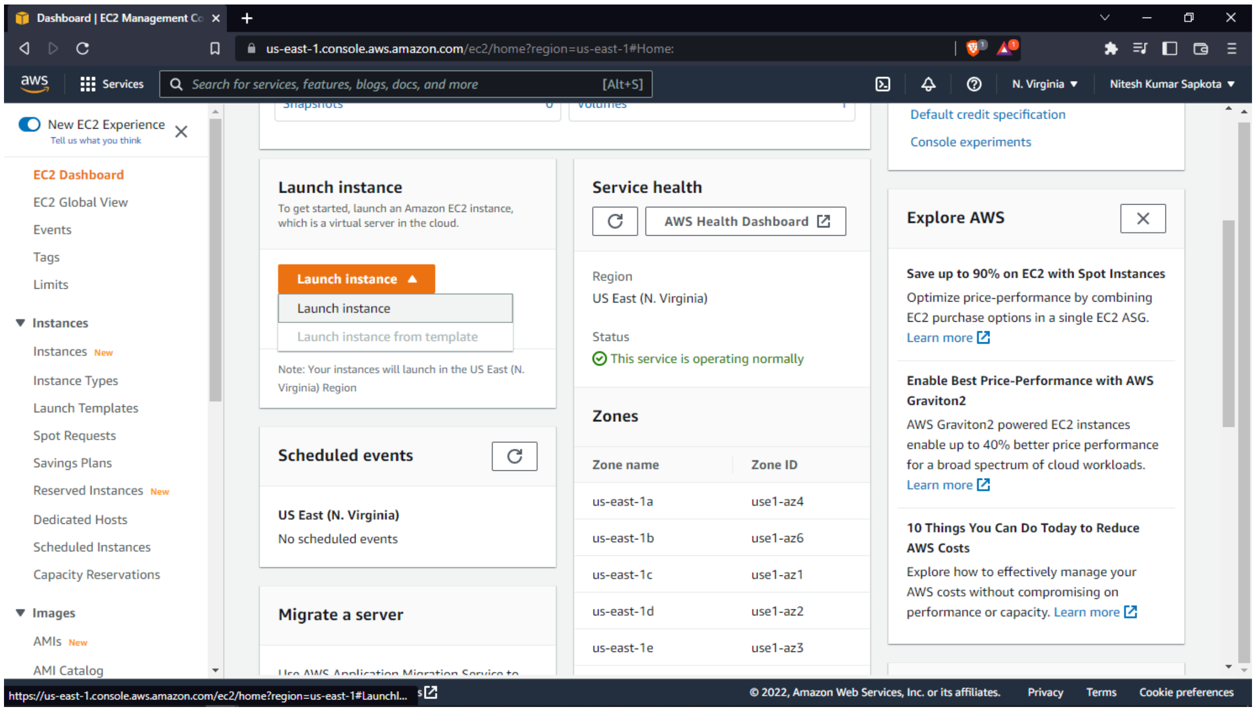
Setting Up EC2 machines: Ubuntu

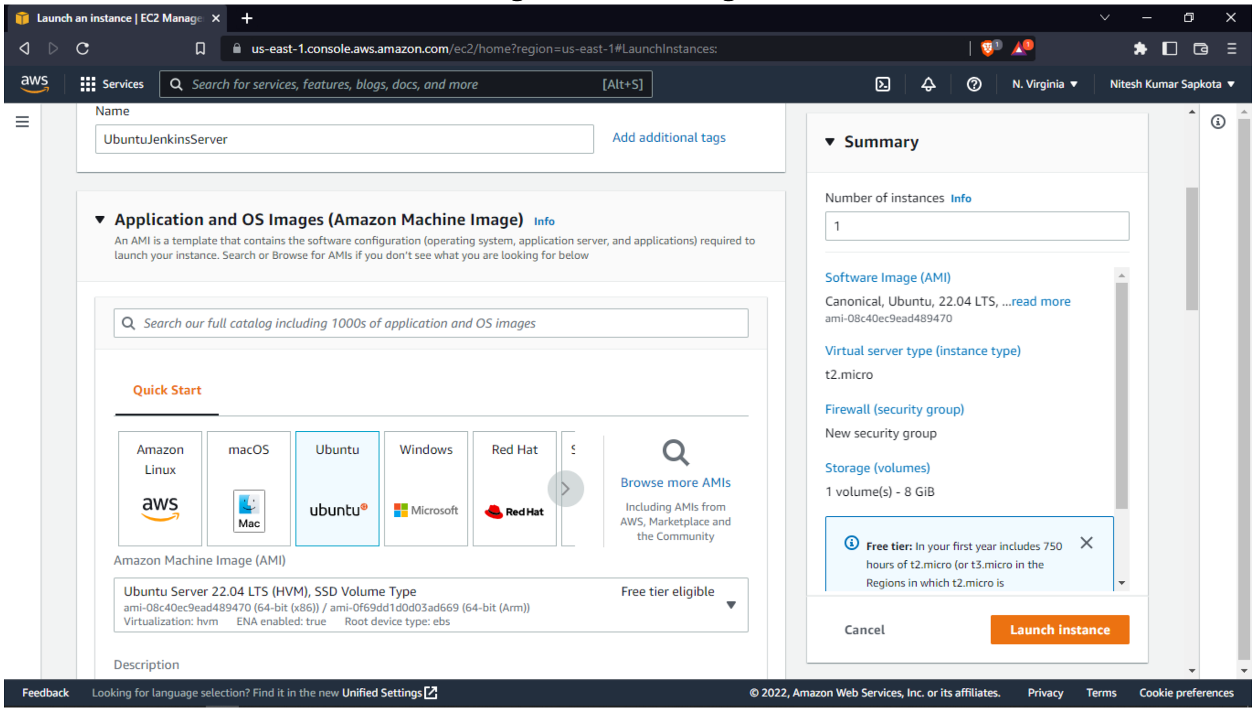
1. Sign up to aws.amazon.com
2. On Services Search for EC2 and Select EC2



3. Select Launch Instance::


Dashboard | EC2 Management Co
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home:
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
Nitesh Kumar Sapkota
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
AMI Catalog
Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance
Launch instance
Launch instance from template
Note: Your instances will launch in the US East (N. Virginia) Region
Scheduled events
US East (N. Virginia)
No scheduled events
Migrate a server
Service health
AWS Health Dashboard
Region
US East (N. Virginia)
Status
This service is operating normally
Zones
Zone name
Zone ID
us-east-1a use1-az4
us-east-1b use1-az6
us-east-1c use1-az1
us-east-1d use1-az2
us-east-1e use1-az3
Default credit specification
Console experiments
Explore AWS
Save up to 90% on EC2 with Spot Instances
Optimize price-performance by combining EC2 purchase options in a single EC2 ASG.
Learn more
Enable Best Price-Performance with AWS Graviton2
AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad spectrum of cloud workloads.
Learn more
10 Things You Can Do Today to Reduce AWS Costs
Explore how to effectively manage your AWS costs without compromising on performance or capacity. Learn more
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Launchl...
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

4. Name the EC2 machine :: UbuntuJenkinsServer
5. Select Ubuntu Linux with default settings of free tier eligible.



Name: UbuntuJenkinsServer

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

Quick Start: Ubuntu

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-08c40ec9ead489470 (64-bit (x86)) / ami-0f69dd1d0d03ad669 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more
ami-08c40ec9ead489470

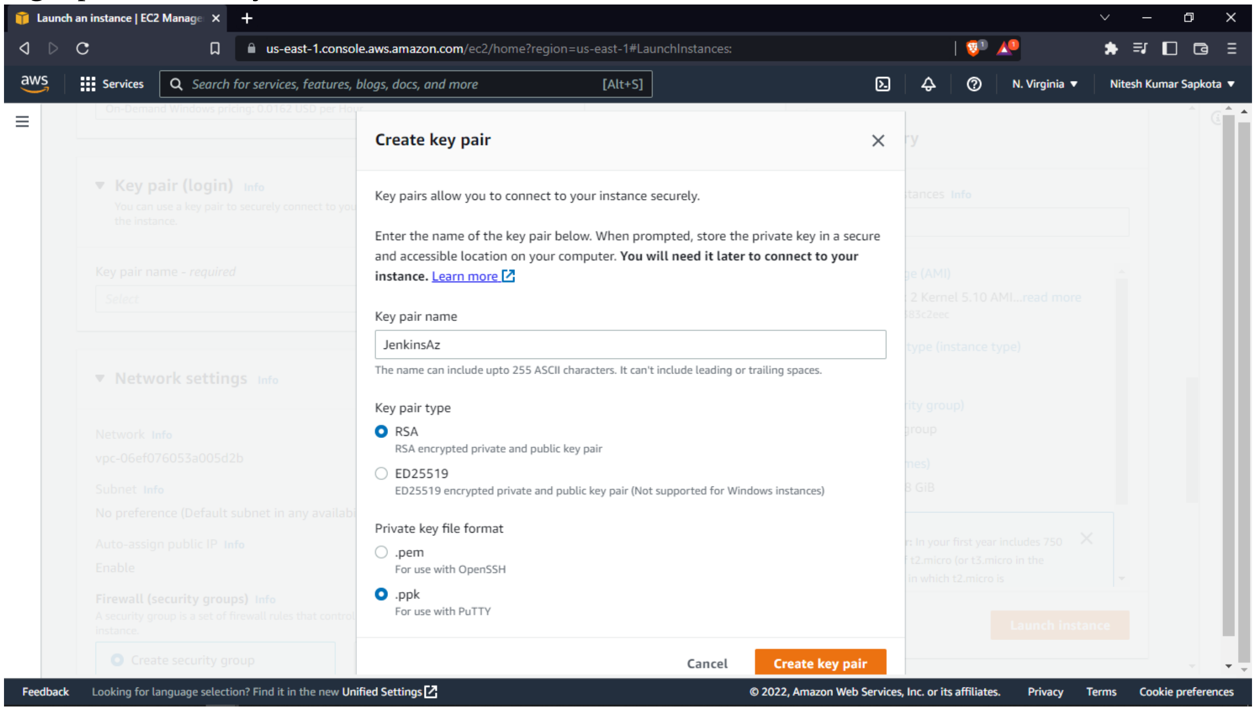
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is

6. Scroll down and find KeyPair Login:: select Create New Key Pair Named it as JenkinsUbuntu and select pair type RSA and key file format with .ppk to easily use this login pair with Putty.



Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. 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

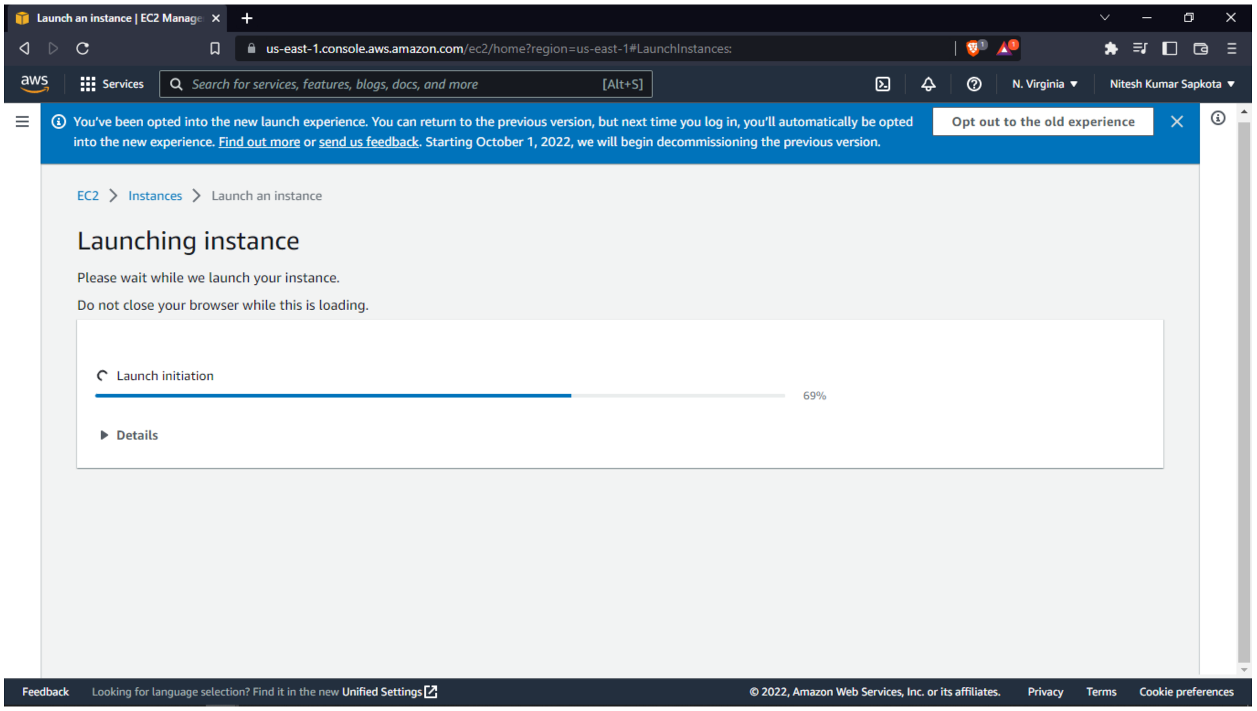
Key pair name: JenkinsAz

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

Key pair type: [✓ RSA] ED25519
- RSA: RSA encrypted private and public key pair
- ED25519: ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format: .pem [✓ .ppk]
- .pem: For use with OpenSSH
- .ppk: For use with PuTTY

7. Download Key Pair and Store it in a secure place.
8. Leave everything default and select Launch Instance.


Launch an instance | EC2 Manage

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

Services | Search for services, features, blogs, docs, and more [Alt+S] | N. Virginia | Nitesh Kumar Sapkota

You've been opted into the new launch experience. You can return to the previous version, but next time you log in, you'll automatically be opted into the new experience. Find out more or send us feedback. Starting October 1, 2022, we will begin decommissioning the previous version.

Opt out to the old experience

EC2 > Instances > Launch an instance

Launching instance

Please wait while we launch your instance.

Do not close your browser while this is loading.

Launch initiation 69%

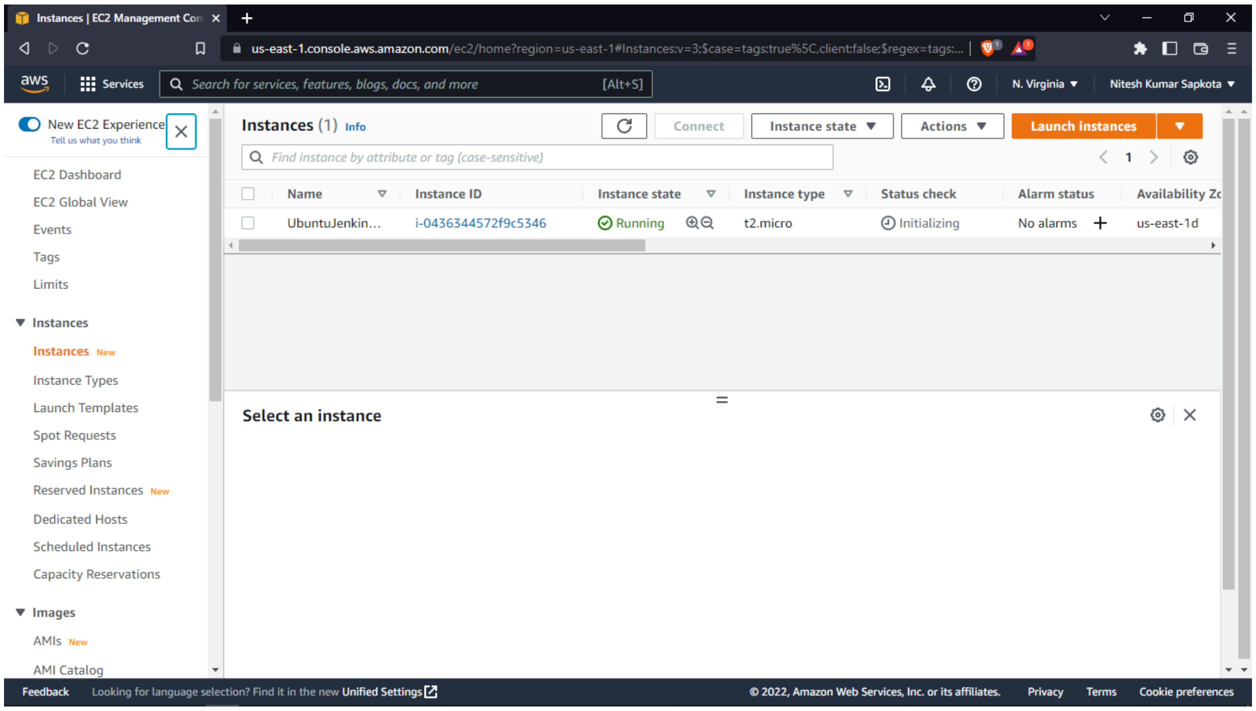
Details

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

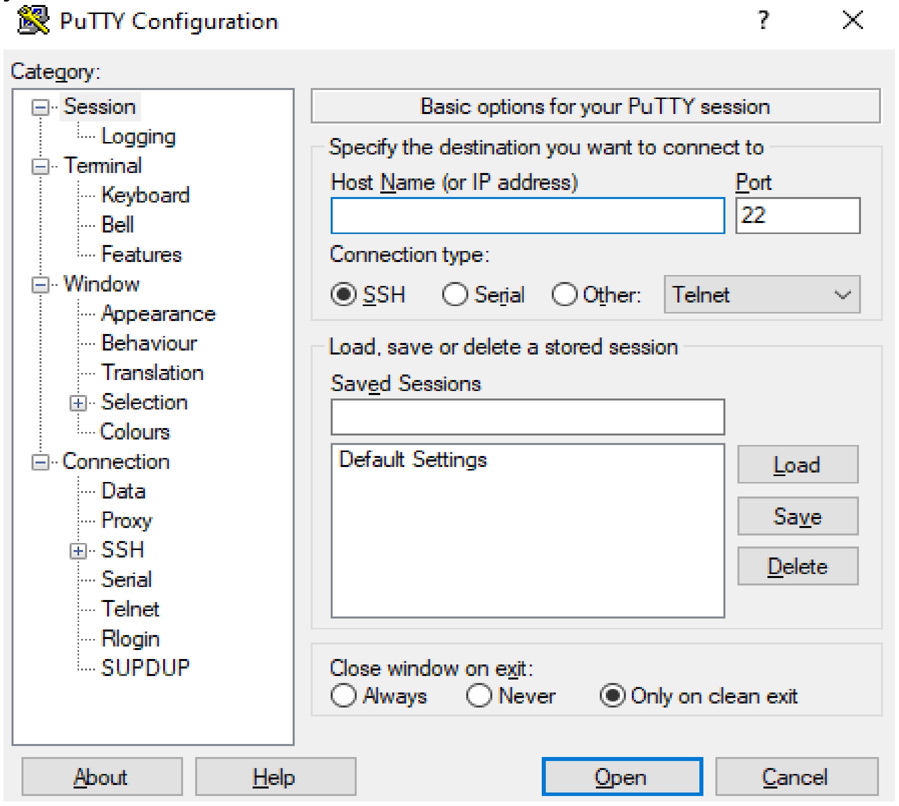
Connecting to EC2 Server Using Putty:

After successful creation of instance we could look all instances status on EC2. The running one in we just created.



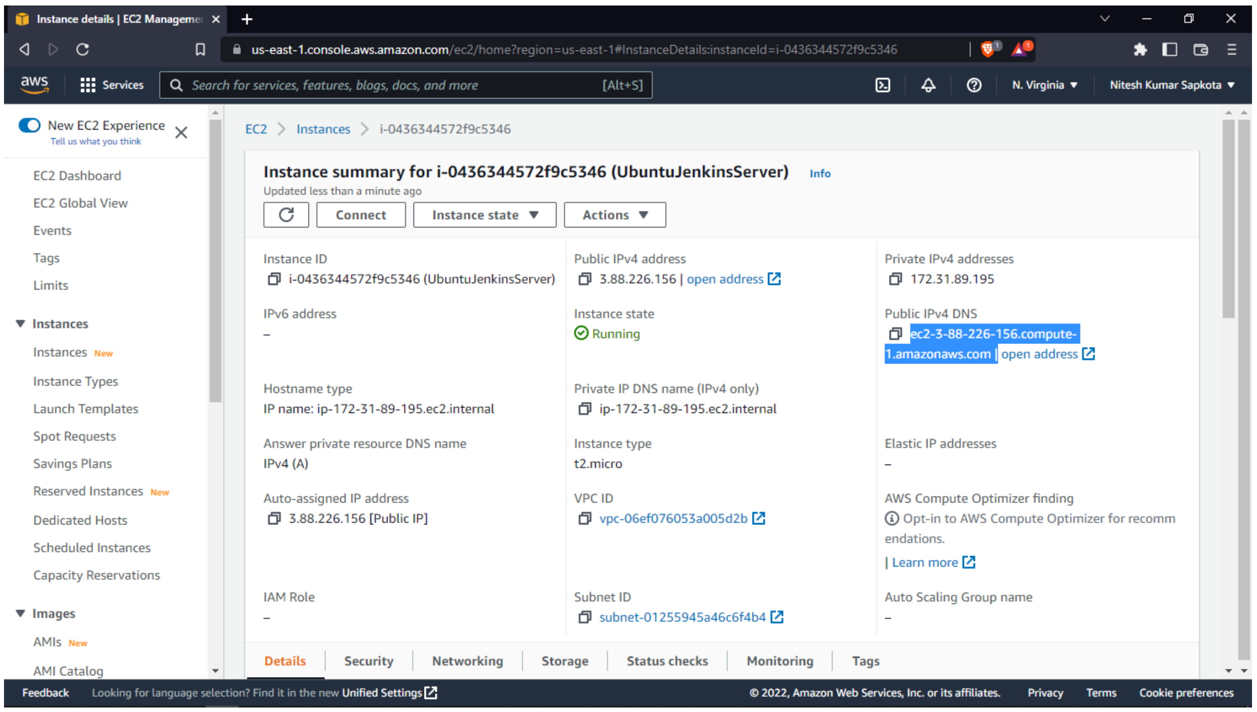
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zo
UbuntuJenkin...	i-0436344572f9c5346	Running	t2.micro	Initializing	No alarms	us-east-1d

1. Download Putty from https://www.putty.org/
2. Open Putty



Specify the destination you want to connect to
Host Name (or IP address):
Port: 22
Connection type: [✓ SSH] Serial Other: Telnet
Saved Sessions:
Default Settings
Close window on exit: Always Never [✓ Only on clean exit]

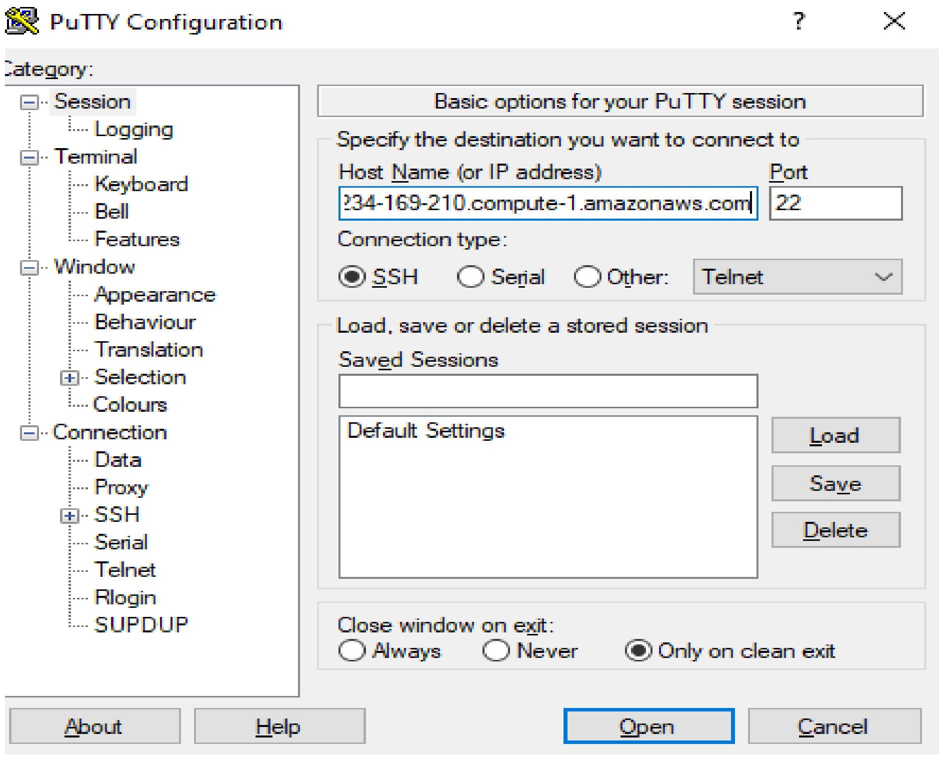
3. On Your EC2 instances Click on instance Id:
4. Select public Ipv4 DNS and Copy



Instance summary for i-0436344572f9c5346 (UbuntuJenkinsServer) Info
Updated less than a minute ago

Instance ID i-0436344572f9c5346 (UbuntuJenkinsServer)	Public IPv4 address 3.88.226.156 | open address	Private IPv4 addresses 172.31.89.195
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-3-88-226-156.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-89-195.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-89-195.ec2.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 3.88.226.156 [Public IP]	VPC ID vpc-06ef076053a005d2b	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-01255945a46c6f4b4	Auto Scaling Group name –

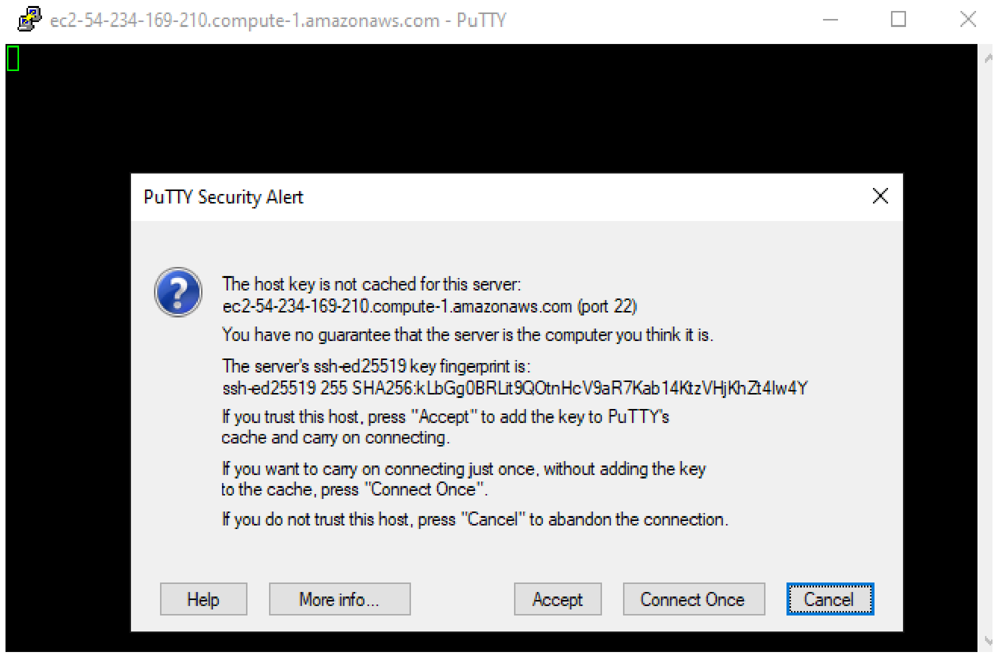
5. Paste It on Host Name on Putty:



Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address): ?34-169-210.compute-1.amazonaws.com
Port: 22
Connection type: [✓ SSH] Serial Other: Telnet
Load, save or delete a stored session
Saved Sessions
Default Settings
Close window on exit: Always Never [✓ Only on clean exit]

6. On the category sidebar Expand Connection and SSH . Select Auth and Browse the .ppk file downloaded while creating a EC2 instance.

7. Click on Open:



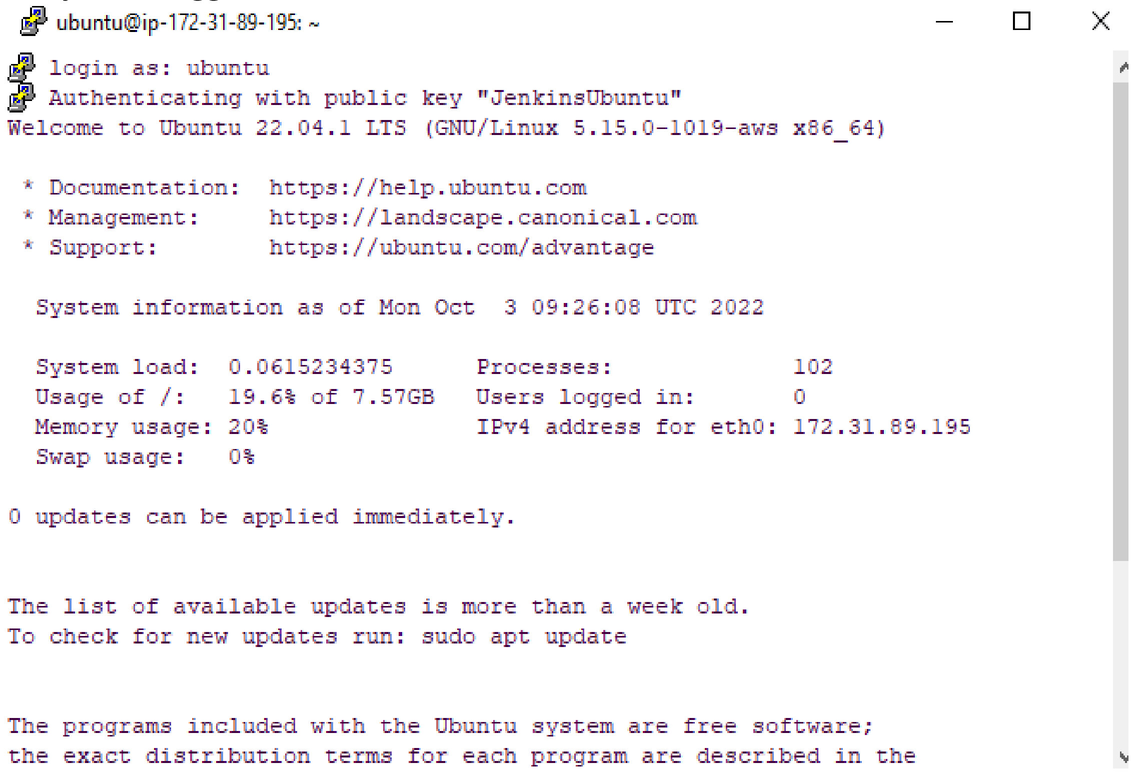
ec2-54-234-169-210.compute-1.amazonaws.com - PuTTY

PuTTY Security Alert

The host key is not cached for this server:
ec2-54-234-169-210.compute-1.amazonaws.com (port 22)
You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:kLbGg0BRLit9QOtnHcV9aR7Kab14KtzVHjKhZt4lw4Y
If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".
If you do not trust this host, press "Cancel" to abandon the connection.

Click on accept

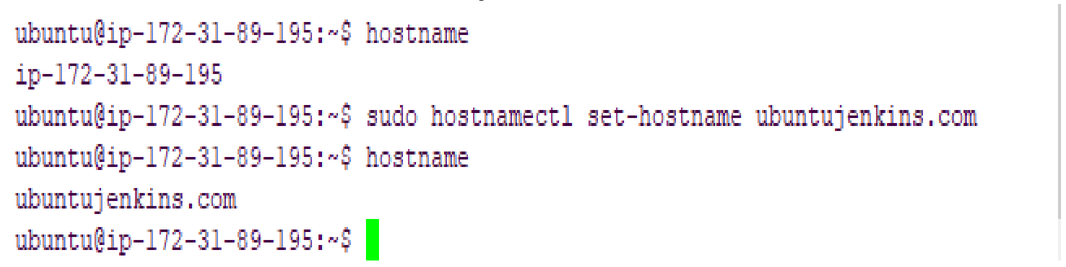
9. Login as ubuntu
10. Now you are logged in:



ubuntu@ip-172-31-89-195: ~

```
login as: ubuntu
Authenticating with public key "JenkinsUbuntu"
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

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

  System information as of Mon Oct  3 09:26:08 UTC 2022

  System load:  0.0615234375      Processes:             102
  Usage of /:   19.6% of 7.57GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 172.31.89.195
  Swap usage:   0%

0 updates can be applied immediately.


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


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
```

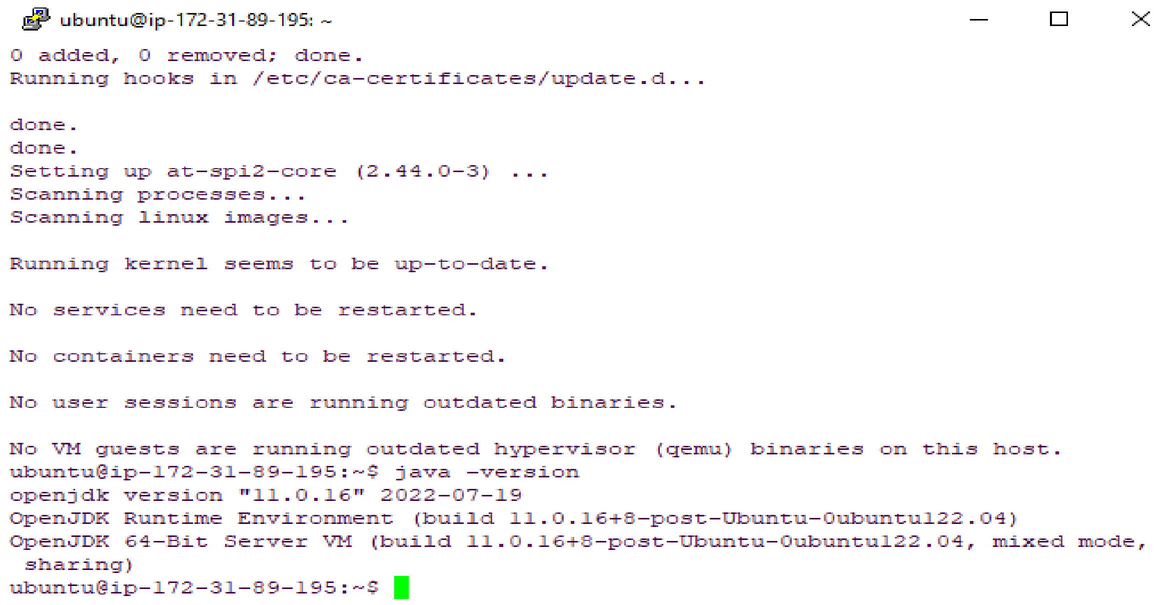
Change Hostname to Ubuntu.jenkins.com
Sudo hostnamectl set-hostname Ubuntu.jenkins.com



```
ubuntu@ip-172-31-89-195:~$ hostname
ip-172-31-89-195
ubuntu@ip-172-31-89-195:~$ sudo hostnamectl set-hostname ubuntujenkins.com
ubuntu@ip-172-31-89-195:~$ hostname
ubuntujenkins.com
ubuntu@ip-172-31-89-195:~$
```

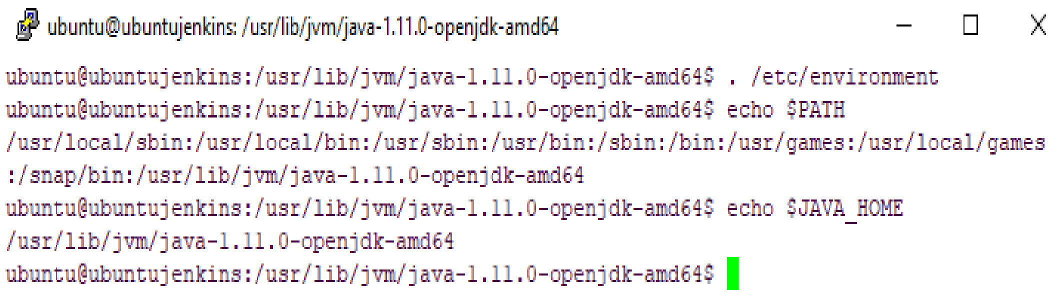
Installing Jenkins:

1. Installing java
   Sudo apt update
   Sudo apt install openjdk-11-jre
   Java –version



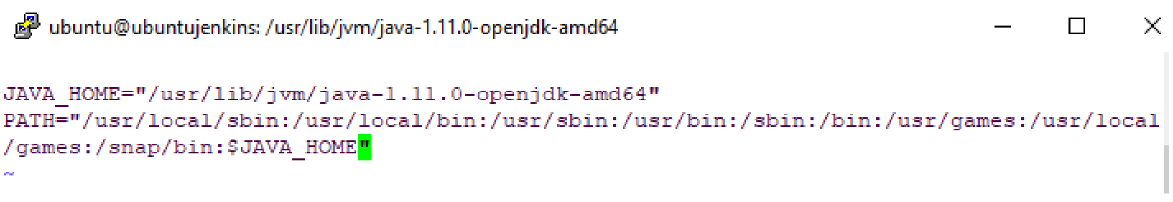
```
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done.
Setting up at-spi2-core (2.44.0-3) ...
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-172-31-89-195:~$ java -version
openjdk version "11.0.16" 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu122.04, mixed mode,
 sharing)
ubuntu@ip-172-31-89-195:~$
```

2. Add java jdk to path
   On /etc/environment



ubuntu@ubuntujenkins: /usr/lib/jvm/java-1.11.0-openjdk-amd64

```
ubuntu@ubuntujenkins:/usr/lib/jvm/java-1.11.0-openjdk-amd64$ . /etc/environment
ubuntu@ubuntujenkins:/usr/lib/jvm/java-1.11.0-openjdk-amd64$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
:/snap/bin:/usr/lib/jvm/java-1.11.0-openjdk-amd64
ubuntu@ubuntujenkins:/usr/lib/jvm/java-1.11.0-openjdk-amd64$ echo $JAVA_HOME
/usr/lib/jvm/java-1.11.0-openjdk-amd64
ubuntu@ubuntujenkins:/usr/lib/jvm/java-1.11.0-openjdk-amd64$
```


ubuntu@ubuntujenkins: /usr/lib/jvm/java-1.11.0-openjdk-amd64

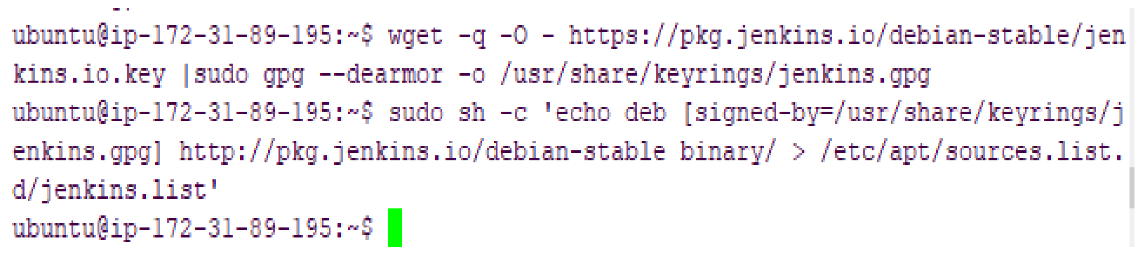
```
JAVA_HOME="/usr/lib/jvm/java-1.11.0-openjdk-amd64"
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local
/games:/snap/bin:$JAVA_HOME"
~
```

3. To install Jenkins:
4. First, add the repository key to your system:

   wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |sudo gpg --dearmor -o /usr/share/keyrings/jenkins.gpg

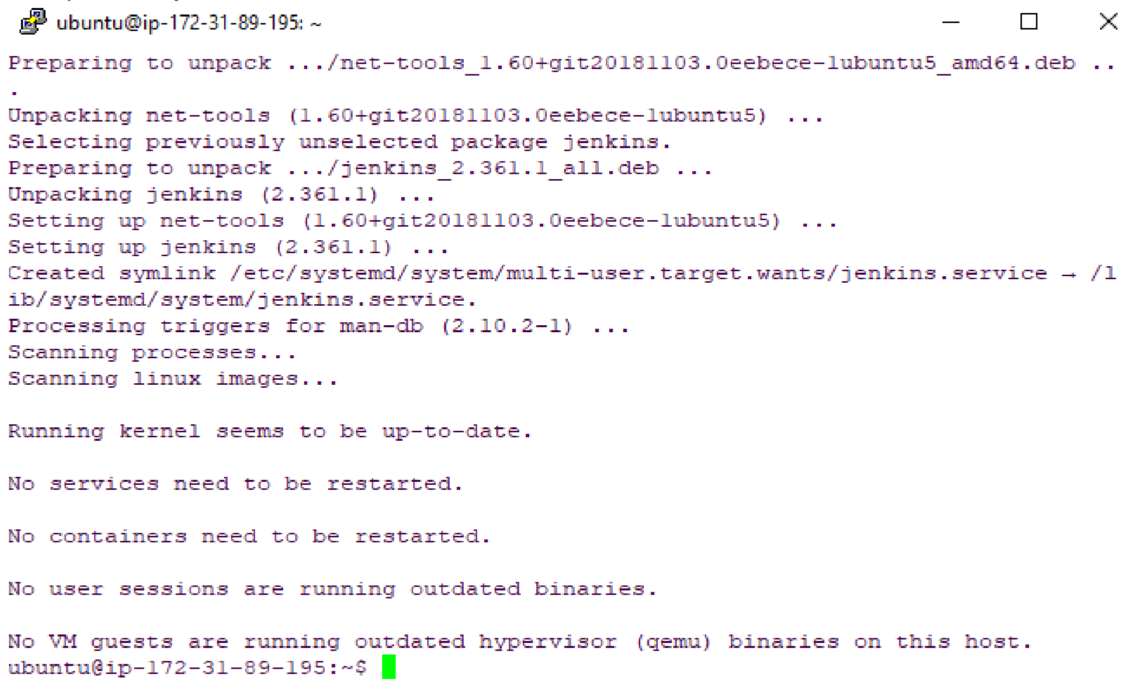
   Then:

   Append the Debian package repository address to the server's sources.list:
   sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg] http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'



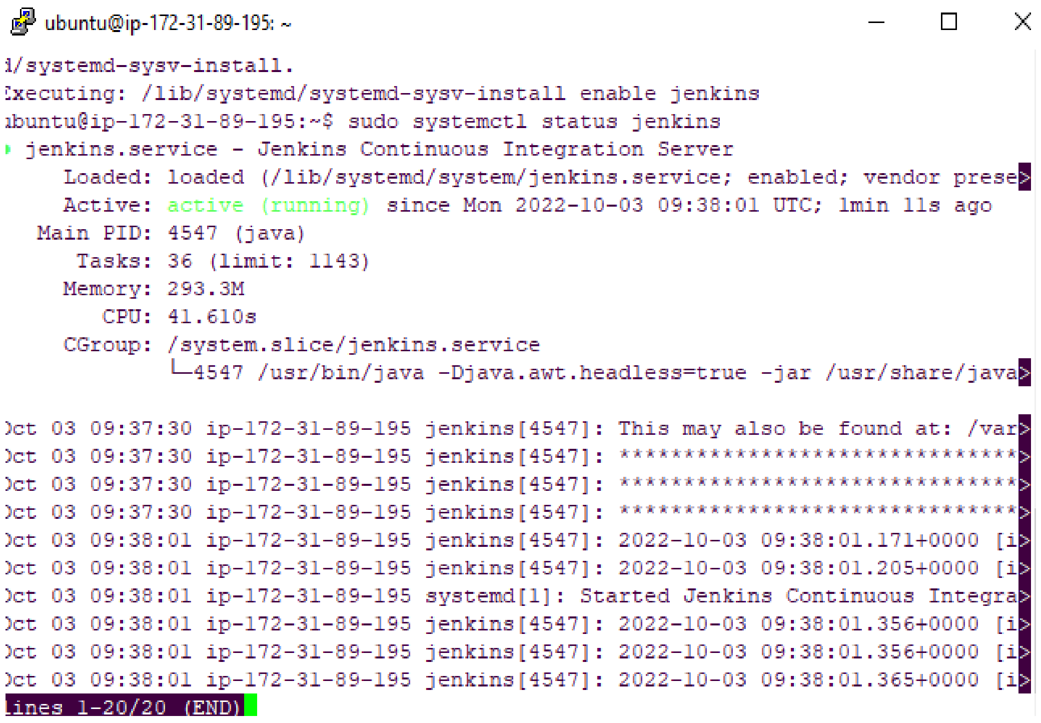
```
ubuntu@ip-172-31-89-195:~$ wget -q -O - https://pkg.jenkins.io/debian-stable/jen
kins.io.key |sudo gpg --dearmor -o /usr/share/keyrings/jenkins.gpg
ubuntu@ip-172-31-89-195:~$ sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/j
enkins.gpg] http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.
d/jenkins.list'
ubuntu@ip-172-31-89-195:~$
```

5. Update the packages and install jenkins :
   sudo apt update
   sudo apt install jenkins


ubuntu@ip-172-31-89-195: ~

```
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ..
.
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.361.1_all.deb ...
Unpacking jenkins (2.361.1) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.361.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /l
ib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-89-195:~$
```

6. Start Jenkins with
   sudo systemctl start Jenkins
   Sudo systemctl enable Jenkins → to start automatically when the system in up.
   Check status of Jenkins with:
   Sudo systemctl status jenkins



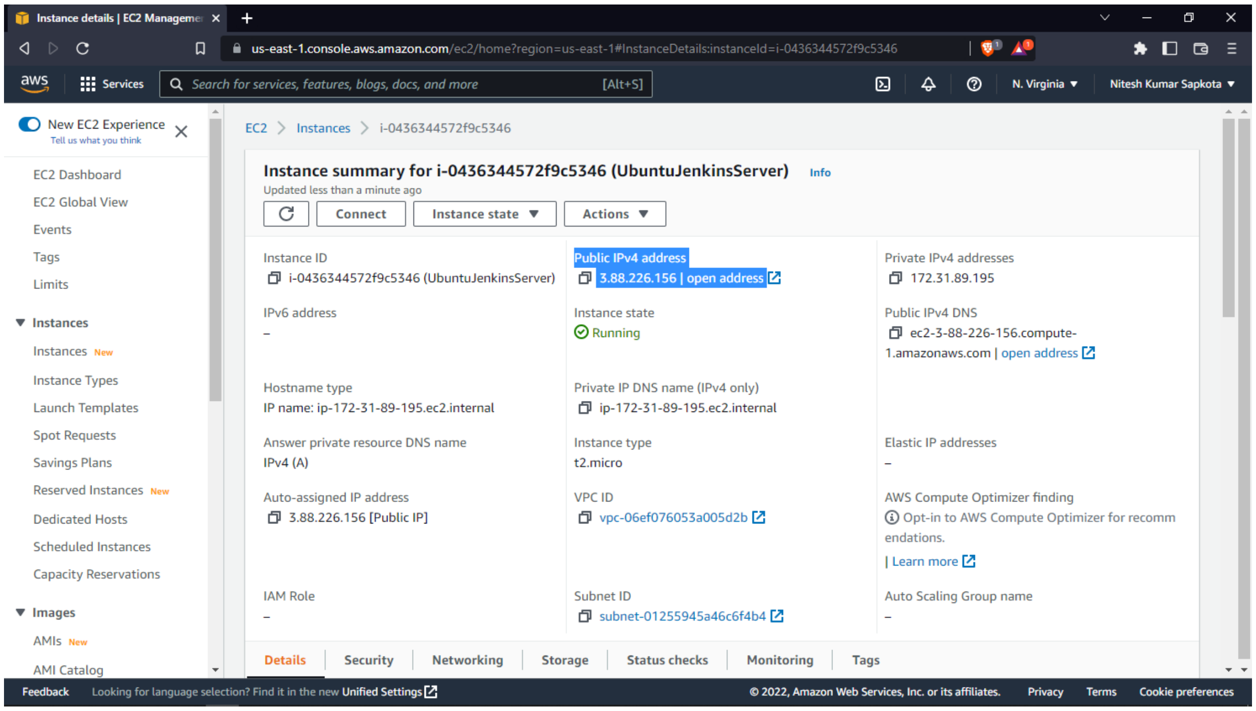
ubuntu@ip-172-31-89-195: ~

```
d/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
ubuntu@ip-172-31-89-195:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor prese>
     Active: active (running) since Mon 2022-10-03 09:38:01 UTC; 1min 11s ago
   Main PID: 4547 (java)
      Tasks: 36 (limit: 1143)
     Memory: 293.3M
        CPU: 41.610s
     CGroup: /system.slice/jenkins.service
             └─4547 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java>

Oct 03 09:37:30 ip-172-31-89-195 jenkins[4547]: This may also be found at: /var>
Oct 03 09:37:30 ip-172-31-89-195 jenkins[4547]: *******************************>
Oct 03 09:37:30 ip-172-31-89-195 jenkins[4547]: *******************************>
Oct 03 09:37:30 ip-172-31-89-195 jenkins[4547]: *******************************>
Oct 03 09:38:01 ip-172-31-89-195 jenkins[4547]: 2022-10-03 09:38:01.171+0000 [i>
Oct 03 09:38:01 ip-172-31-89-195 jenkins[4547]: 2022-10-03 09:38:01.205+0000 [i>
Oct 03 09:38:01 ip-172-31-89-195 systemd[1]: Started Jenkins Continuous Integra>
Oct 03 09:38:01 ip-172-31-89-195 jenkins[4547]: 2022-10-03 09:38:01.356+0000 [i>
Oct 03 09:38:01 ip-172-31-89-195 jenkins[4547]: 2022-10-03 09:38:01.356+0000 [i>
Oct 03 09:38:01 ip-172-31-89-195 jenkins[4547]: 2022-10-03 09:38:01.365+0000 [i>
lines 1-20/20 (END)
```

Reference for Jenkins installation on Ubuntu 20.04::

https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-22-04

Now Jenkins runs on port 8080 so we could browse port 8080 with ip



Instance details | EC2 Management

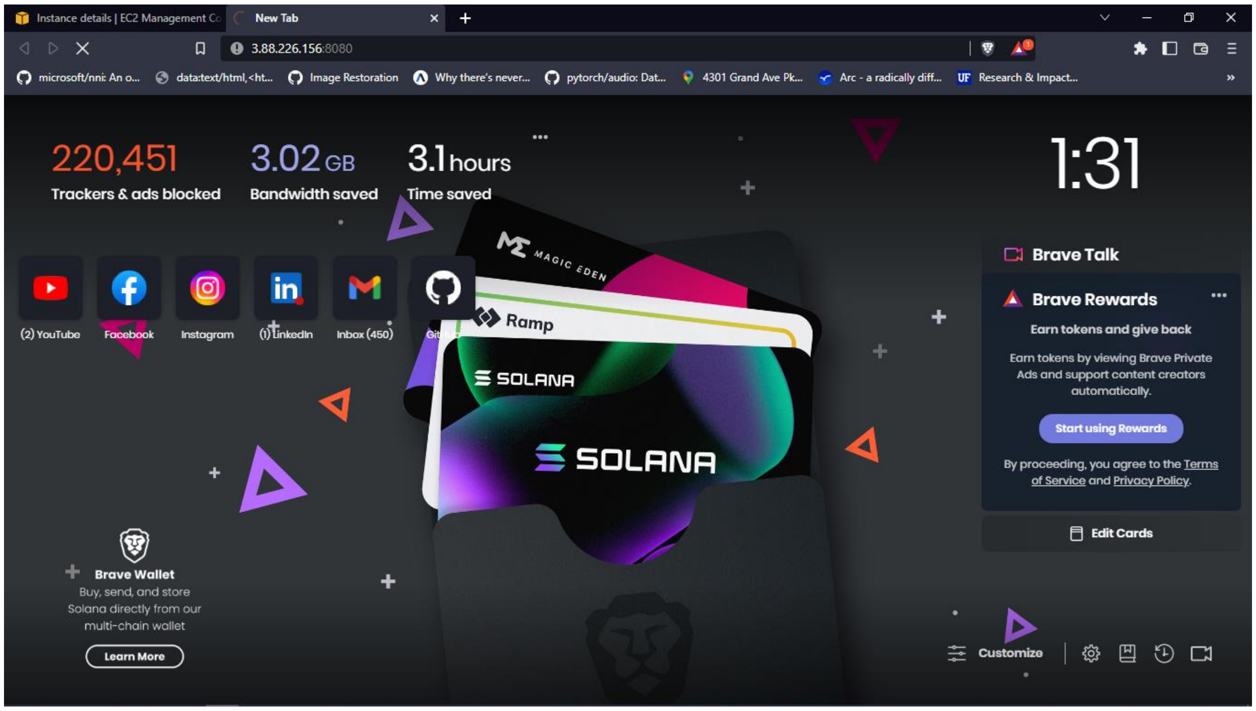
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-0436344572f9c5346

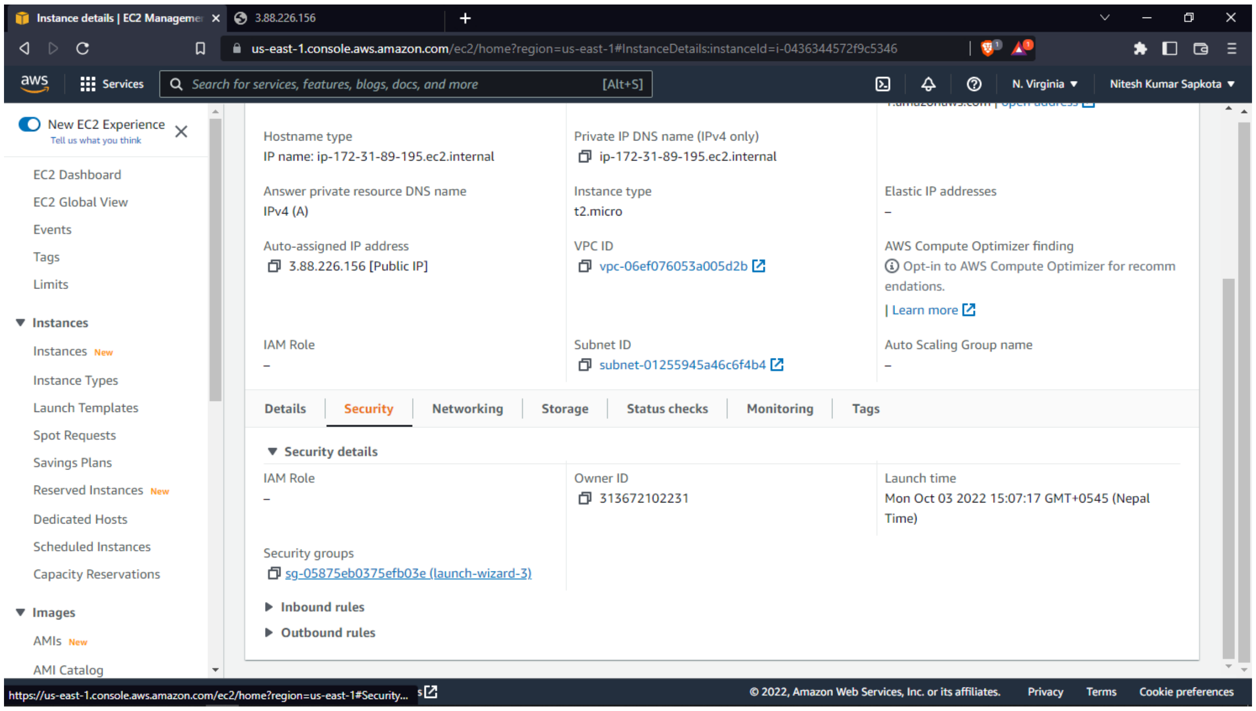
EC2 > Instances > i-0436344572f9c5346

Instance summary for i-0436344572f9c5346 (UbuntuJenkinsServer) Info
Updated less than a minute ago

Instance ID i-0436344572f9c5346 (UbuntuJenkinsServer)	Public IPv4 address 3.88.226.156 | open address	Private IPv4 addresses 172.31.89.195
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-3-88-226-156.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-89-195.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-89-195.ec2.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 3.88.226.156 [Public IP]	VPC ID vpc-06ef076053a005d2b	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-01255945a46c6f4b4	Auto Scaling Group name –

It doesnot return any response as of default security group inbound rules so we need to at an inbound rule to open 8080 TCP port that can be access from anywhere 0.0.0.0.




Hostname type
IP name: ip-172-31-89-195.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-89-195.ec2.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
3.88.226.156 [Public IP]

VPC ID
vpc-06ef076053a005d2b

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

IAM Role
–

Subnet ID
subnet-01255945a46c6f4b4

Auto Scaling Group name
–

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

Security details

IAM Role
–

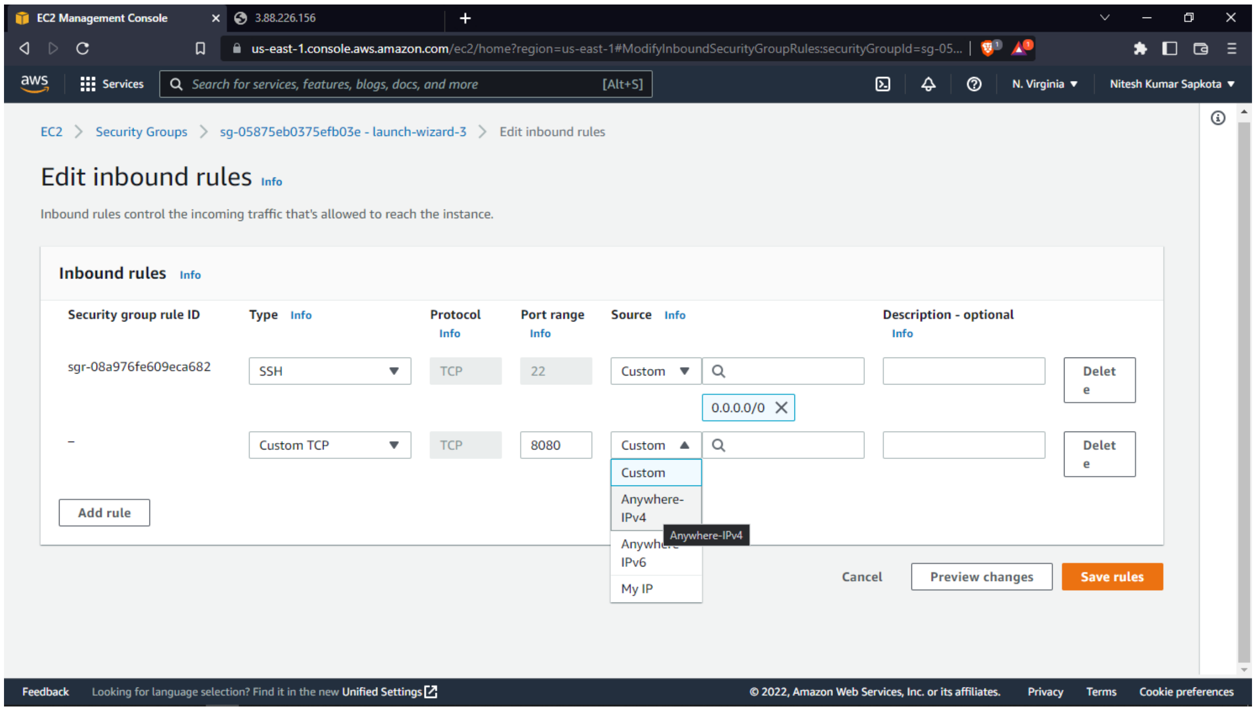
Owner ID
313672102231

Launch time
Mon Oct 03 2022 15:07:17 GMT+0545 (Nepal Time)

Security groups
sg-05875eb0375efb03e (launch-wizard-3)

Inbound rules
Outbound rules

Select Security on instances and select security groups by scrolling down.


EC2 > Security Groups > sg-05875eb0375efb03e - launch-wizard-3 > Edit inbound rules

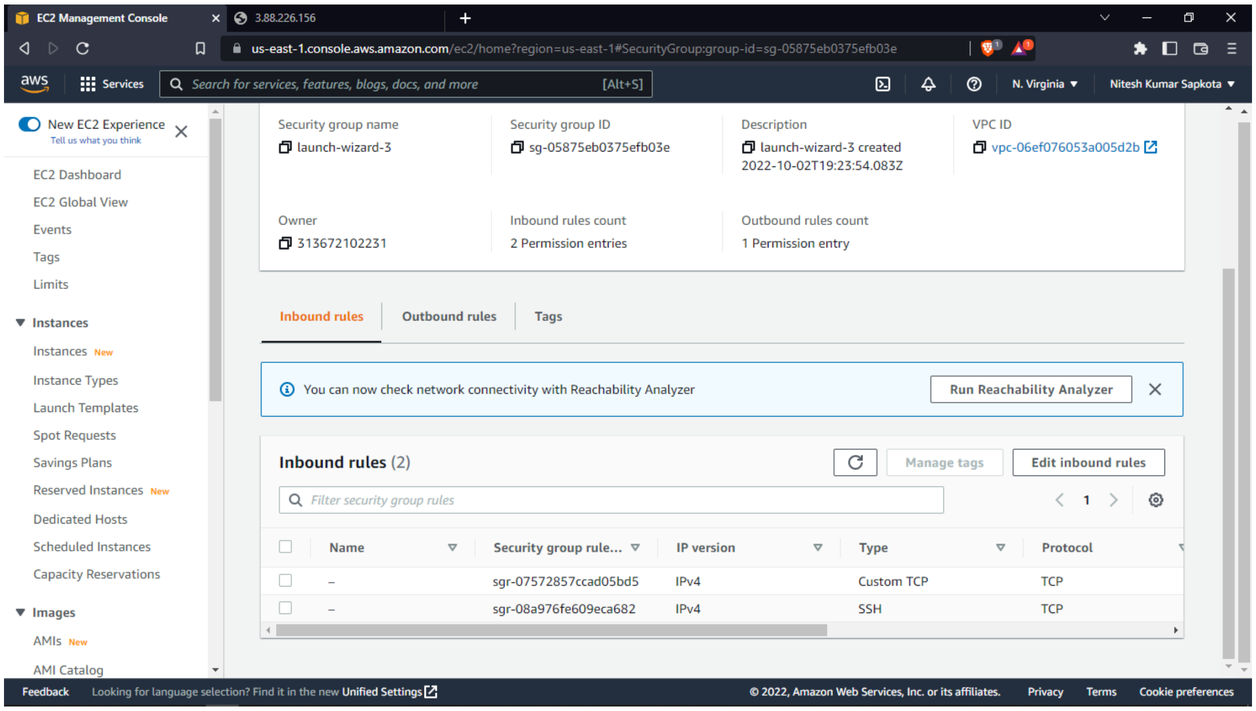
Edit inbound rules Info

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

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-08a976fe609eca682	SSH	TCP	22	Custom 0.0.0.0/0	
–	Custom TCP	TCP	8080	Custom	

Save Rules


Security group name: launch-wizard-3

Security group ID: sg-05875eb0375efb03e

Description: launch-wizard-3 created 2022-10-02T19:23:54.083Z

VPC ID: vpc-06ef076053a005d2b

Owner: 313672102231

Inbound rules count: 2 Permission entries

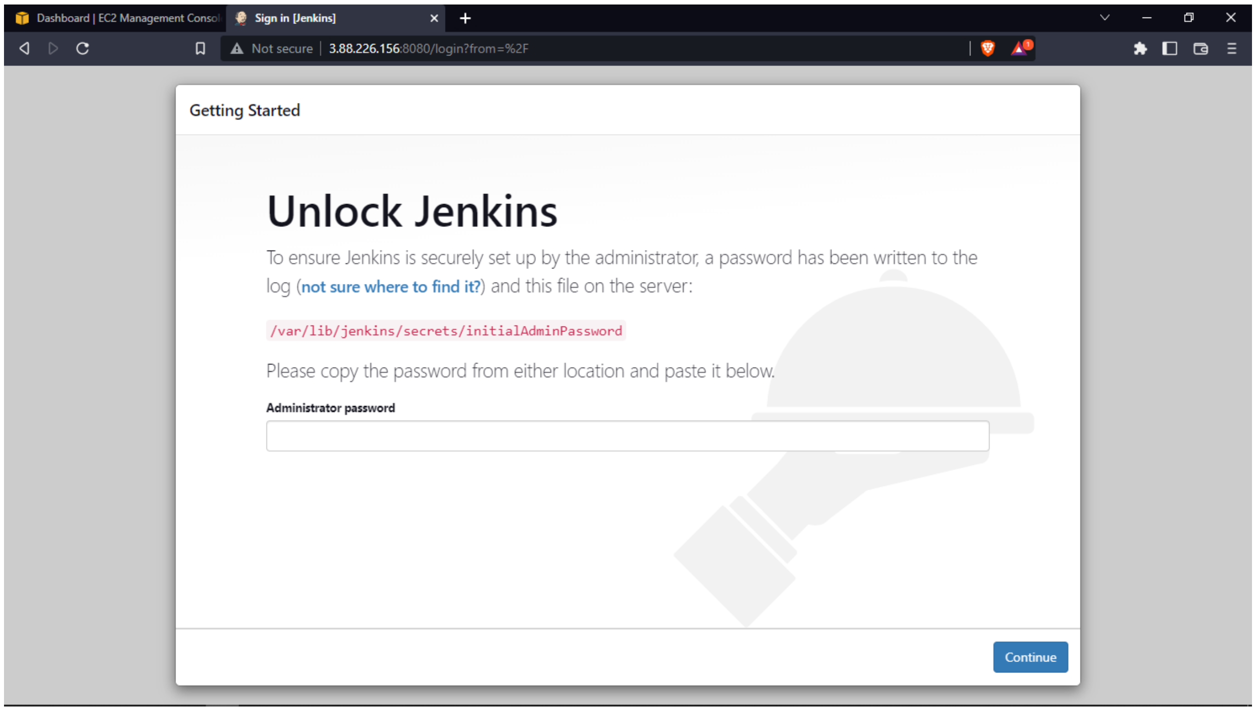
Outbound rules count: 1 Permission entry

Inbound rules (2)

Name	Security group rule...	IP version	Type	Protocol
–	sgr-07572857ccad05bd5	IPv4	Custom TCP	TCP
–	sgr-08a976fe609eca682	IPv4	SSH	TCP

Now browse:: Ip:8080

Jenkins is Up and running::


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:

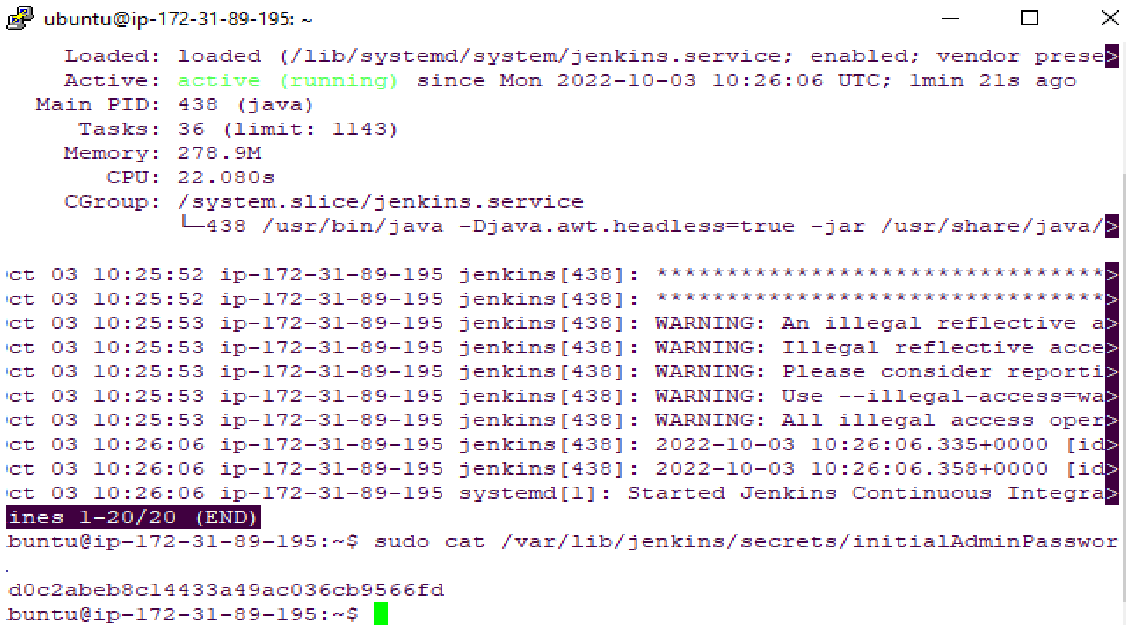
```
/var/lib/jenkins/secrets/initialAdminPassword
```

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

Administrator password

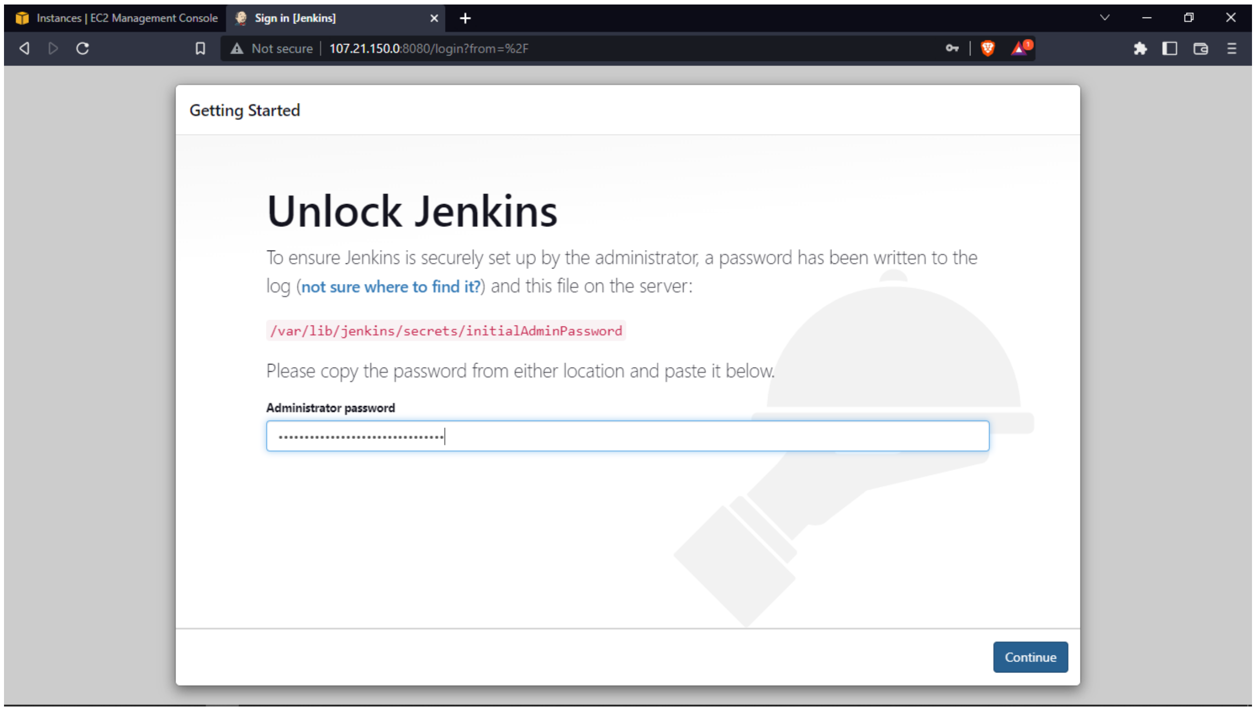
To get Initial Password::

Sudo cat `/var/lib/jenkins/secrets/initialAdminPassword`



```
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor prese>
     Active: active (running) since Mon 2022-10-03 10:26:06 UTC; 1min 21s ago
   Main PID: 438 (java)
      Tasks: 36 (limit: 1143)
     Memory: 278.9M
        CPU: 22.080s
     CGroup: /system.slice/jenkins.service
             └─438 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/>

ct 03 10:25:52 ip-172-31-89-195 jenkins[438]: ******************************>
ct 03 10:25:52 ip-172-31-89-195 jenkins[438]: ******************************>
ct 03 10:25:53 ip-172-31-89-195 jenkins[438]: WARNING: An illegal reflective a>
ct 03 10:25:53 ip-172-31-89-195 jenkins[438]: WARNING: Illegal reflective acce>
ct 03 10:25:53 ip-172-31-89-195 jenkins[438]: WARNING: Please consider reporti>
ct 03 10:25:53 ip-172-31-89-195 jenkins[438]: WARNING: Use --illegal-access=wa>
ct 03 10:25:53 ip-172-31-89-195 jenkins[438]: WARNING: All illegal access oper>
ct 03 10:26:06 ip-172-31-89-195 jenkins[438]: 2022-10-03 10:26:06.335+0000 [id>
ct 03 10:26:06 ip-172-31-89-195 jenkins[438]: 2022-10-03 10:26:06.358+0000 [id>
ct 03 10:26:06 ip-172-31-89-195 systemd[1]: Started Jenkins Continuous Integra>
ines 1-20/20 (END)
buntu@ip-172-31-89-195:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPasswor

d0c2abeb8c14433a49ac036cb9566fd
buntu@ip-172-31-89-195:~$
```

2d0c2abeb8c14433a49ac036cb9566fd



Instances | EC2 Management Console
Sign in [Jenkins]
Not secure
107.21.150.0:8080/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:

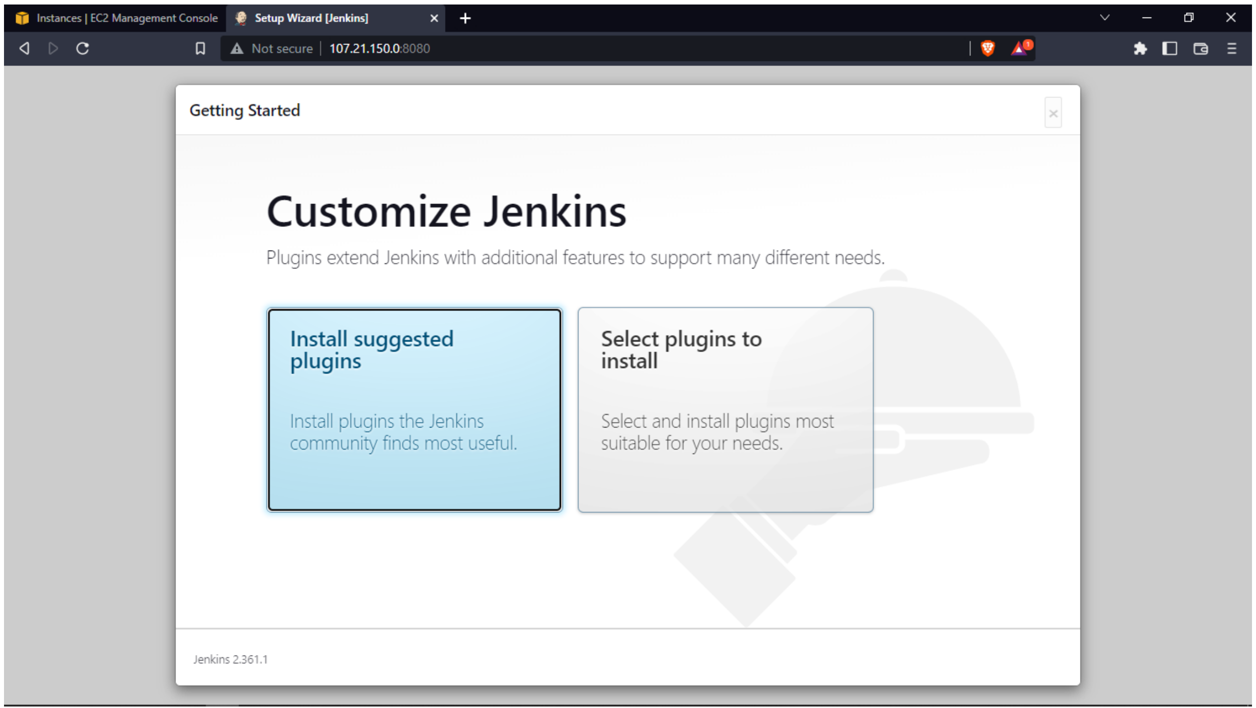
`/var/lib/jenkins/secrets/initialAdminPassword`

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

Administrator password

································

Select install suggested plugins



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

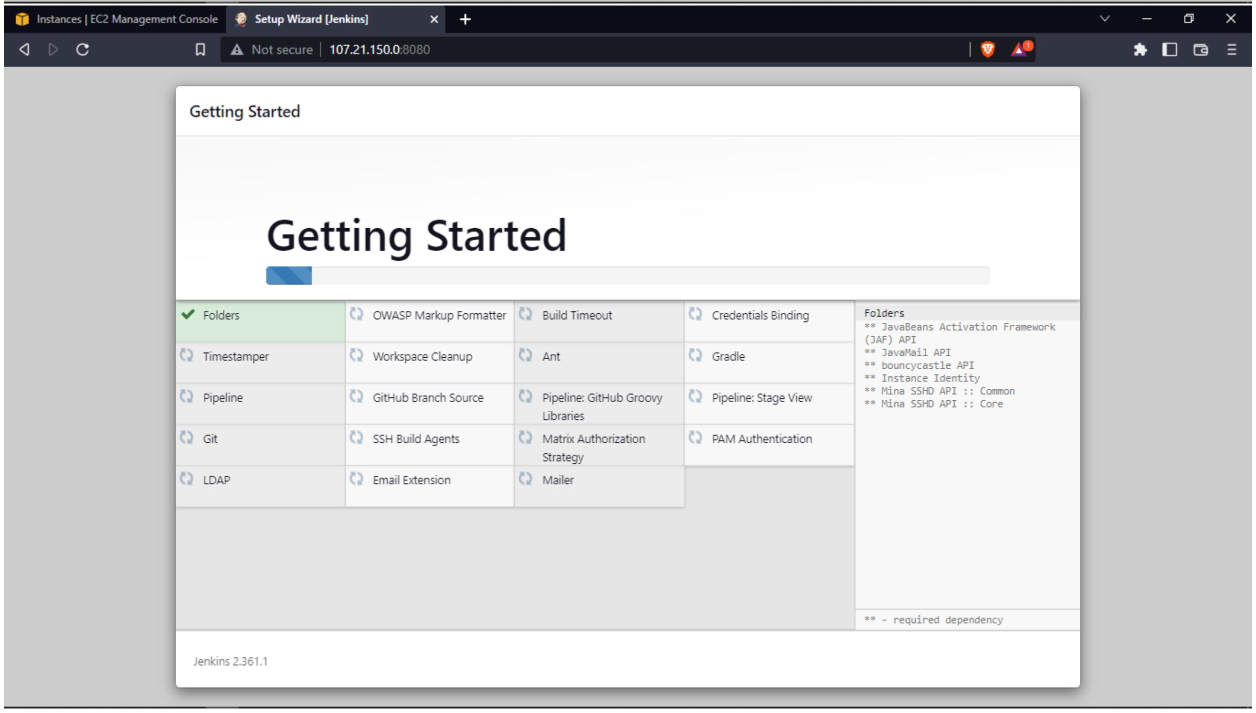
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.361.1



Getting Started

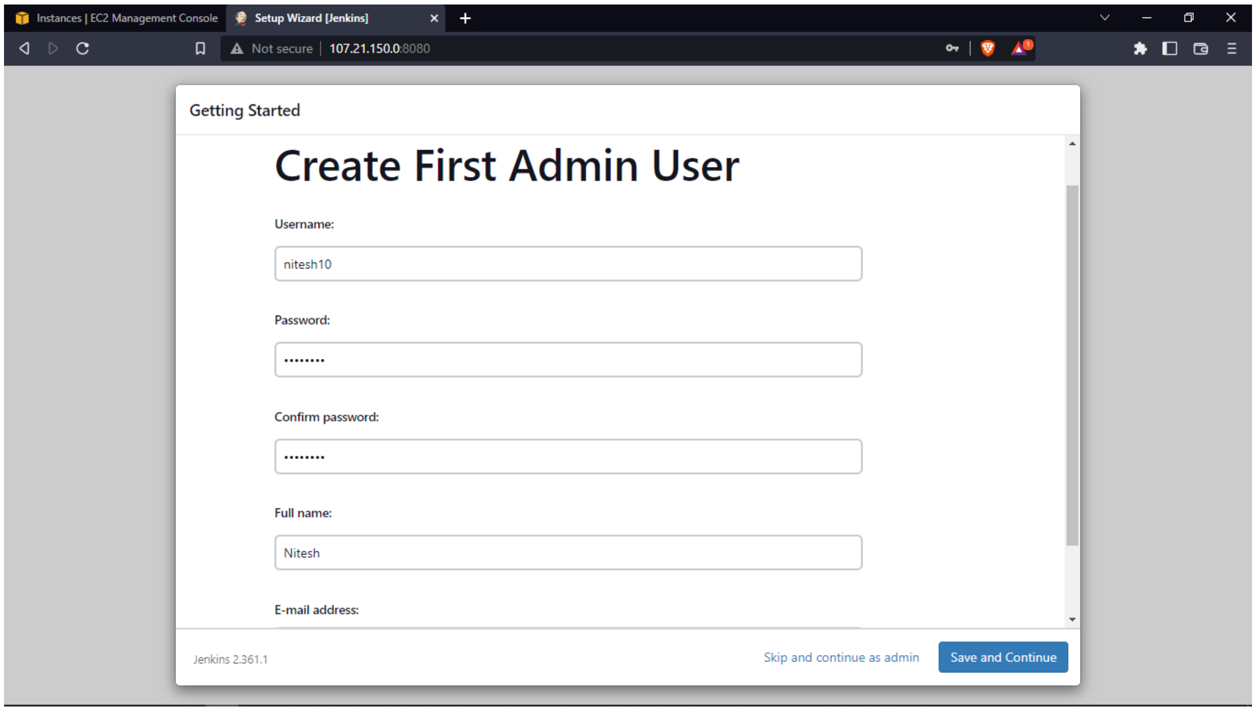
Getting Started

✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

```
Folders
** JavaBeans Activation Framework
(JAF) API
** JavaMail API
** bouncycastle API
** Instance Identity
** Mina SSHD API :: Common
** Mina SSHD API :: Core

** - required dependency
```

Jenkins 2.361.1

Create first admin user with your credentials:



Getting Started

Create First Admin User

Username: nitesh10

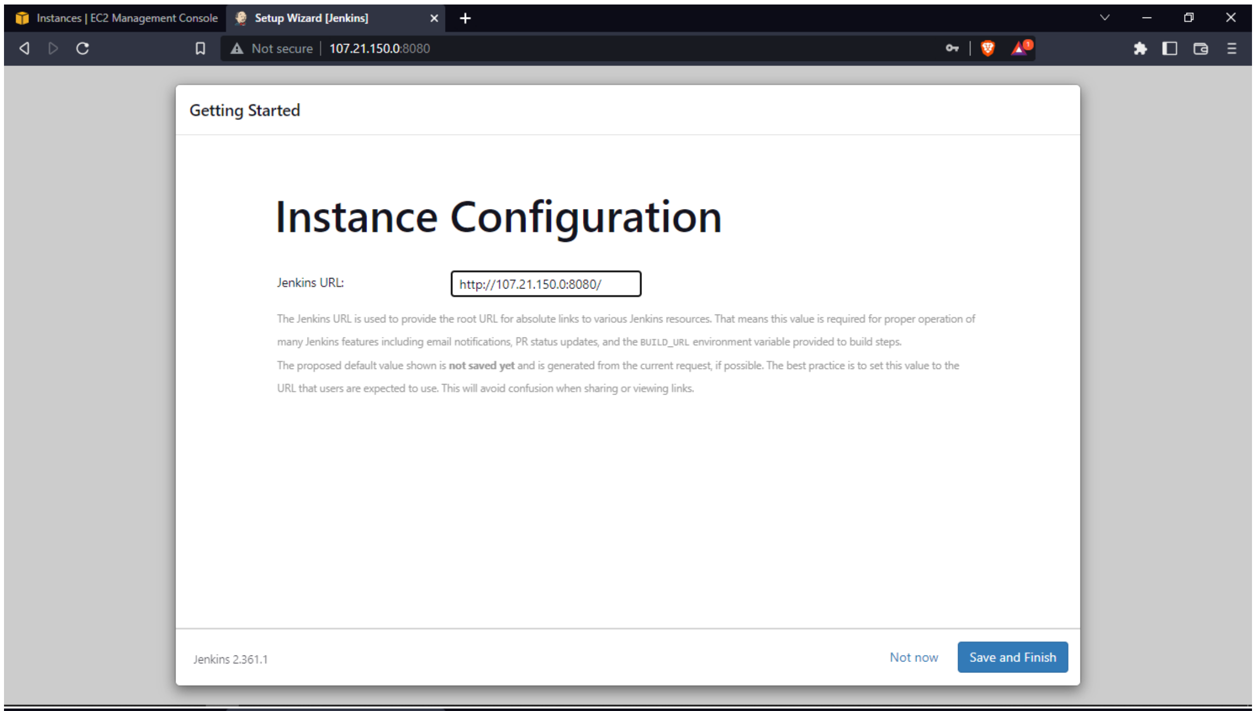
Password: ••••••••

Confirm password: ••••••••

Full name: Nitesh

E-mail address:

Jenkins 2.361.1


Instances | EC2 Management Console | Setup Wizard [Jenkins]

Not secure | 107.21.150.0:8080

Getting Started

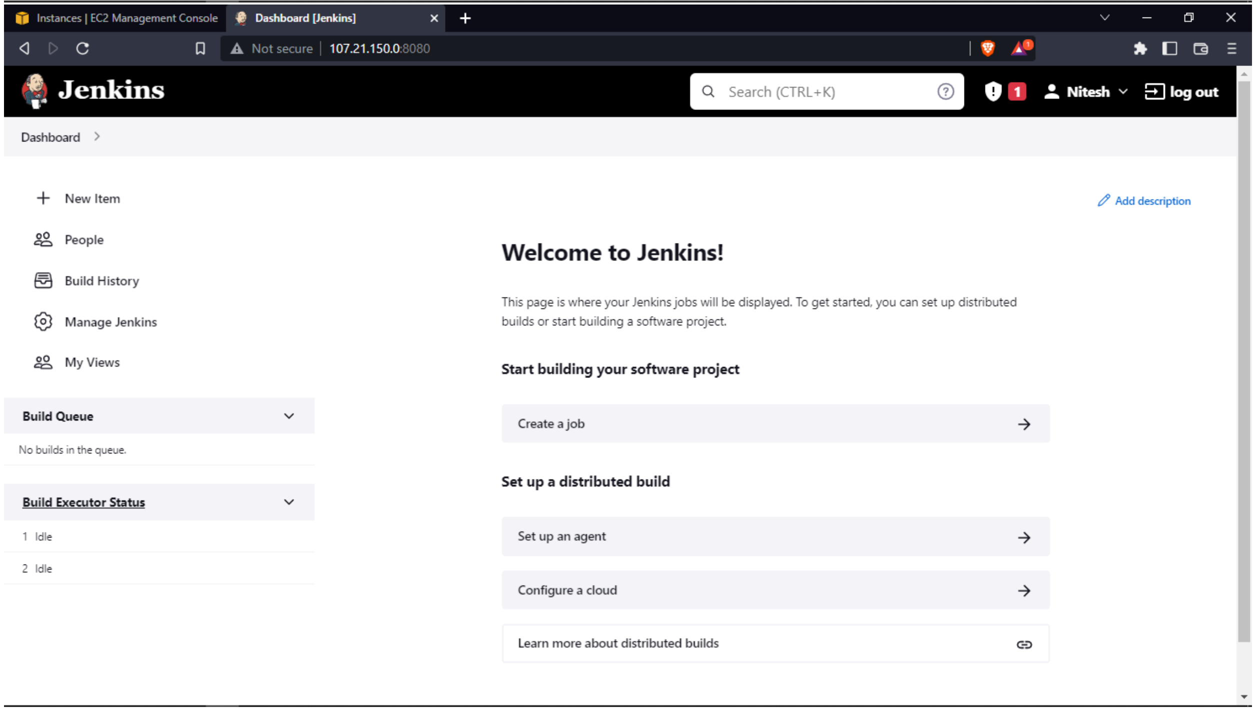
Instance Configuration

Jenkins URL: http://107.21.150.0:8080/

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 2.361.1


Instances | EC2 Management Console | Dashboard [Jenkins]

Not secure | 107.21.150.0:8080

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

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

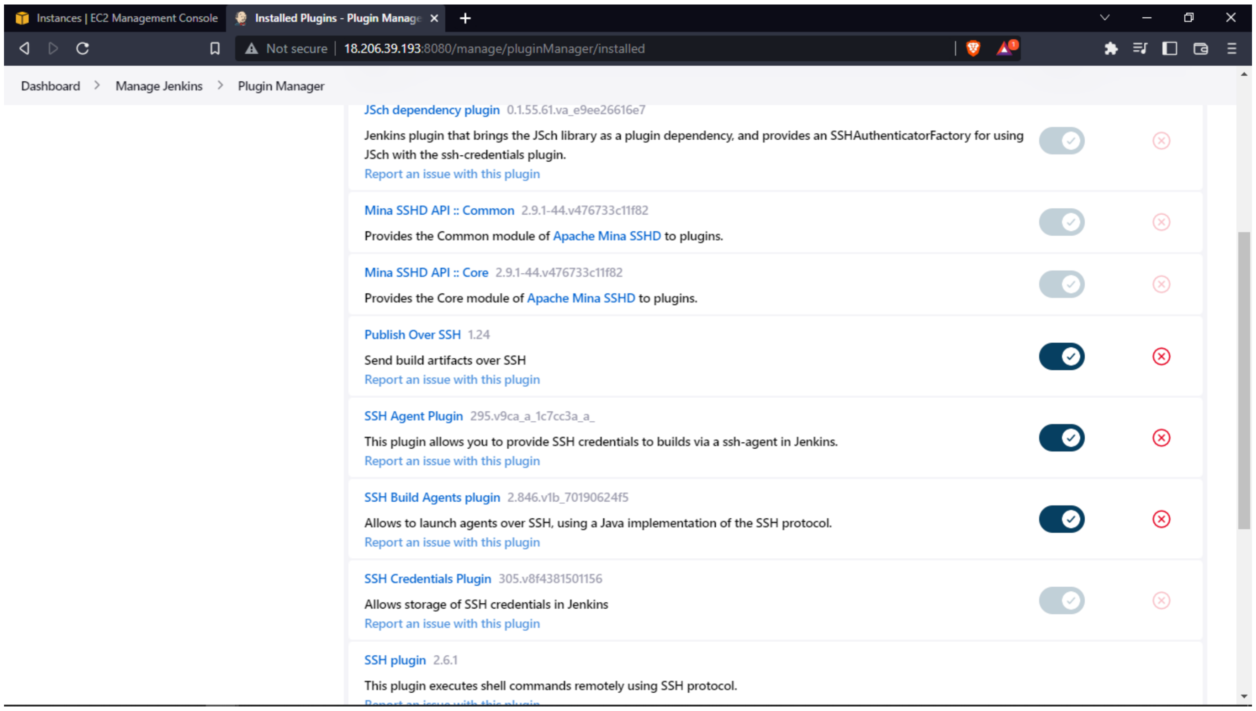
Learn more about distributed builds

Installing Plugins on Jenkins:

Navigate to Manage Jenkins:

Go to Manage Plugins:

Install Publish over SSH and SSH agent Plugin.



Instances | EC2 Management Console | Installed Plugins - Plugin Manager

Not secure | 18.206.39.193:8080/manage/pluginManager/installed

Dashboard > Manage Jenkins > Plugin Manager

JSch dependency plugin 0.1.55.61.va_e9ee26616e7
Jenkins plugin that brings the JSch library as a plugin dependency, and provides an SSHAuthenticatorFactory for using JSch with the ssh-credentials plugin.
Report an issue with this plugin

Mina SSHD API :: Common 2.9.1-44.v476733c11f82
Provides the Common module of Apache Mina SSHD to plugins.

Mina SSHD API :: Core 2.9.1-44.v476733c11f82
Provides the Core module of Apache Mina SSHD to plugins.

Publish Over SSH 1.24
Send build artifacts over SSH
Report an issue with this plugin

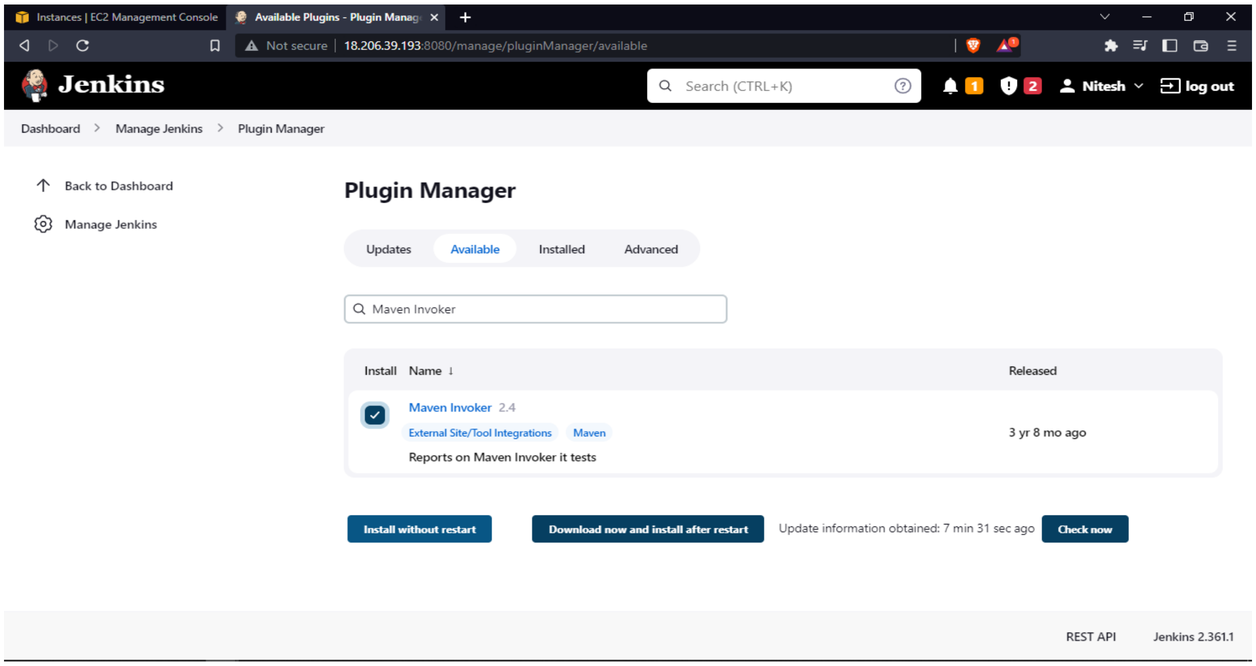
SSH Agent Plugin 295.v9ca_a_1c7cc3a_a_
This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.
Report an issue with this plugin

SSH Build Agents plugin 2.846.v1b_70190624f5
Allows to launch agents over SSH, using a Java implementation of the SSH protocol.
Report an issue with this plugin

SSH Credentials Plugin 305.v8f4381501156
Allows storage of SSH credentials in Jenkins
Report an issue with this plugin

SSH plugin 2.6.1
This plugin executes shell commands remotely using SSH protocol.

Search for Maven Invoker Plugin.



Instances | EC2 Management Console | Available Plugins - Plugin Manager

Not secure | 18.206.39.193:8080/manage/pluginManager/available

Jenkins — Search (CTRL+K) — Nitesh — log out

Dashboard > Manage Jenkins > Plugin Manager

Back to Dashboard
Manage Jenkins

Plugin Manager

Updates | Available | Installed | Advanced

Maven Invoker

Install	Name ↓	Released
✓	Maven Invoker 2.4 — External Site/Tool Integrations, Maven — Reports on Maven Invoker it tests	3 yr 8 mo ago

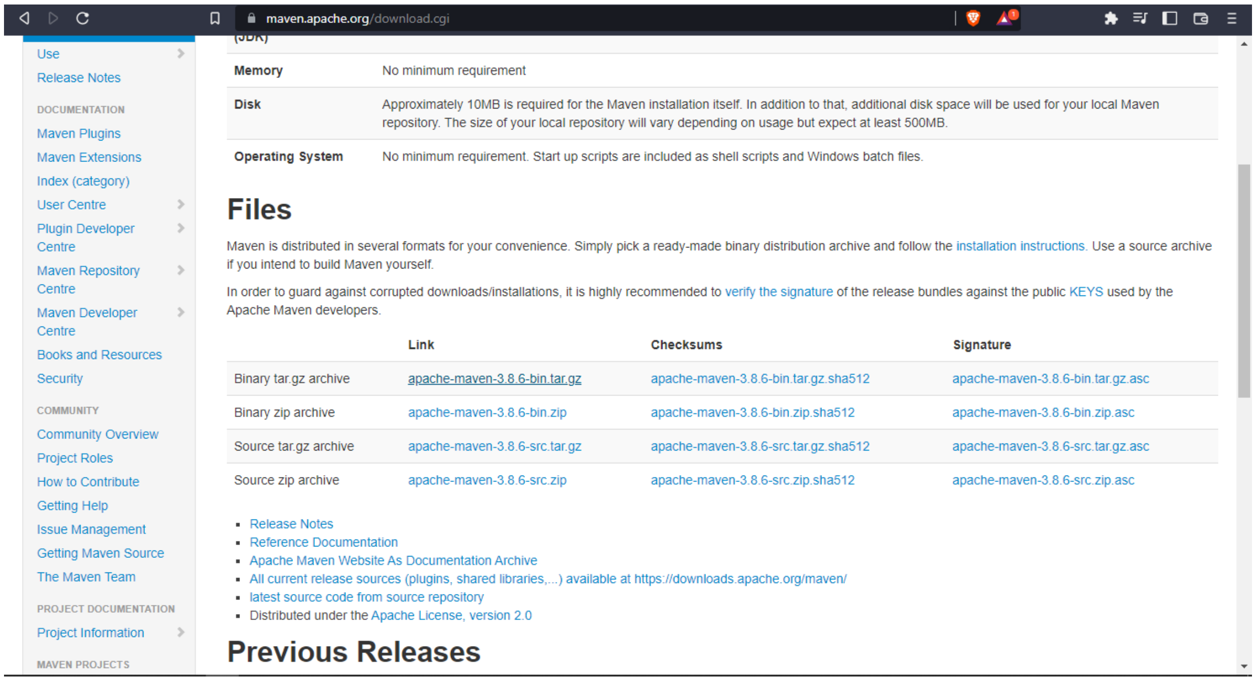
Install without restart | Download now and install after restart | Update information obtained: 7 min 31 sec ago | Check now

REST API — Jenkins 2.361.1

Click on Install without restart.

Installing Maven on our UbuntuJenkins server.

1. Go to https://maven.apache.org/download.cgi
   Wget .tar.gz link



maven.apache.org/download.cgi

Use
Release Notes
DOCUMENTATION
Maven Plugins
Maven Extensions
Index (category)
User Centre
Plugin Developer Centre
Maven Repository Centre
Maven Developer Centre
Books and Resources
Security
COMMUNITY
Community Overview
Project Roles
How to Contribute
Getting Help
Issue Management
Getting Maven Source
The Maven Team
PROJECT DOCUMENTATION
Project Information
MAVEN PROJECTS

Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts and Windows batch files.

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

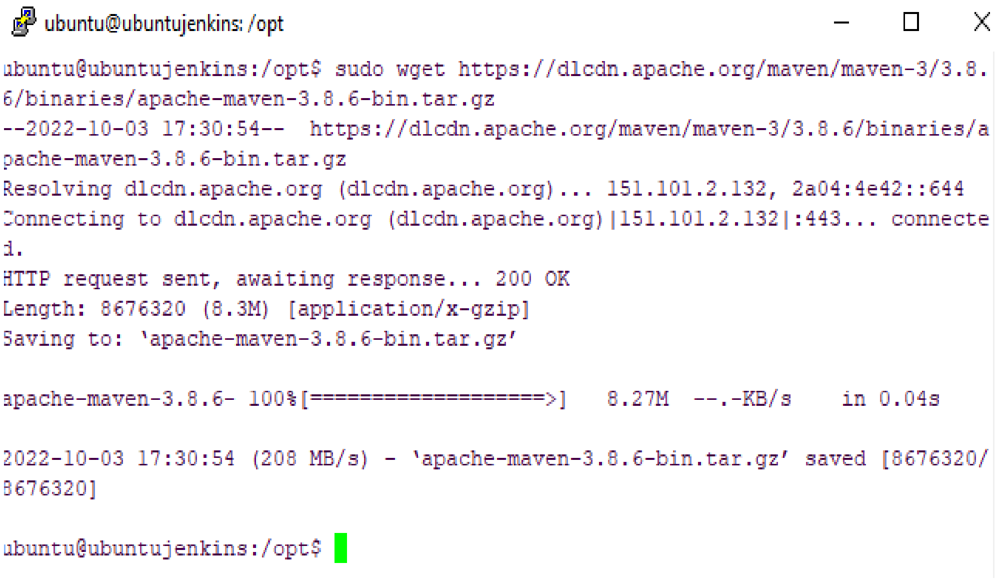
In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.8.6-bin.tar.gz	apache-maven-3.8.6-bin.tar.gz.sha512	apache-maven-3.8.6-bin.tar.gz.asc
Binary zip archive	apache-maven-3.8.6-bin.zip	apache-maven-3.8.6-bin.zip.sha512	apache-maven-3.8.6-bin.zip.asc
Source tar.gz archive	apache-maven-3.8.6-src.tar.gz	apache-maven-3.8.6-src.tar.gz.sha512	apache-maven-3.8.6-src.tar.gz.asc
Source zip archive	apache-maven-3.8.6-src.zip	apache-maven-3.8.6-src.zip.sha512	apache-maven-3.8.6-src.zip.asc

- Release Notes
- Reference Documentation
- Apache Maven Website As Documentation Archive
- All current release sources (plugins, shared libraries,...) available at https://downloads.apache.org/maven/
- latest source code from source repository
- Distributed under the Apache License, version 2.0

Previous Releases

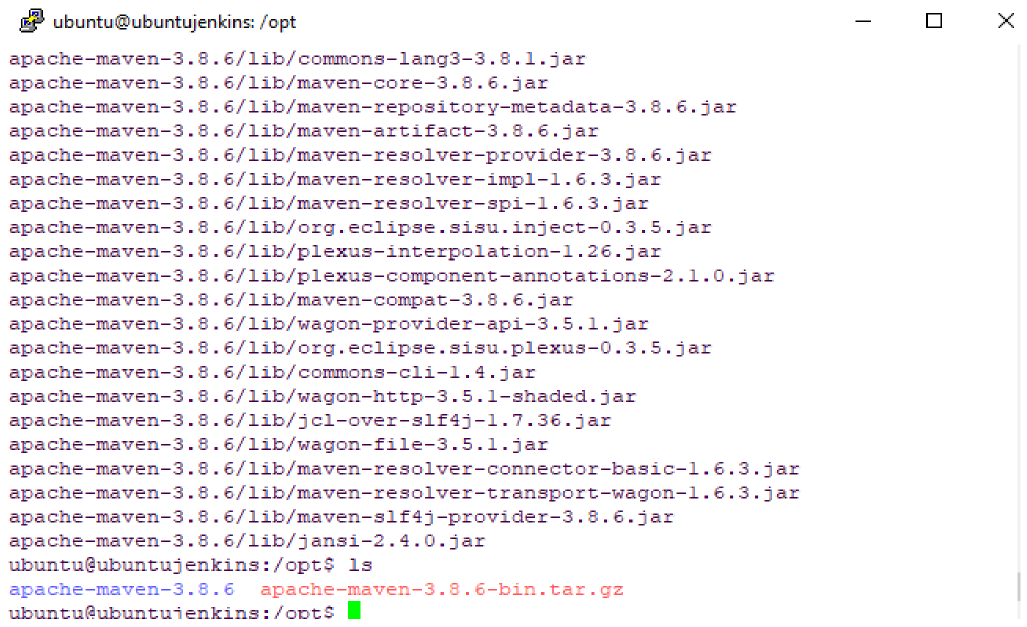
   Copy .tar.gz link
   Sudo wget link..
   Cd /opt



ubuntu@ubuntujenkins: /opt

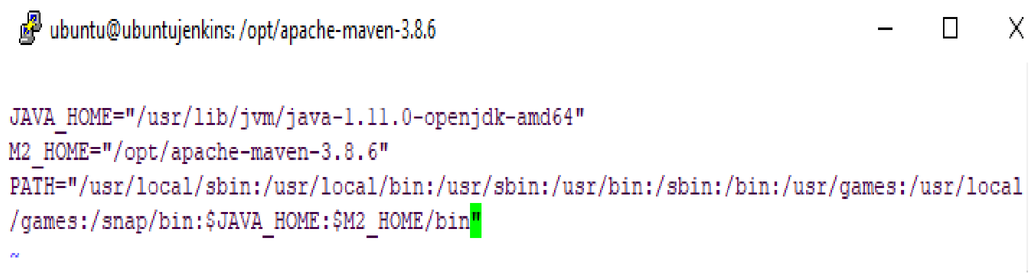
```
ubuntu@ubuntujenkins:/opt$ sudo wget https://dlcdn.apache.org/maven/maven-3/3.8.
6/binaries/apache-maven-3.8.6-bin.tar.gz
--2022-10-03 17:30:54--  https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/a
pache-maven-3.8.6-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 8676320 (8.3M) [application/x-gzip]
Saving to: 'apache-maven-3.8.6-bin.tar.gz'

apache-maven-3.8.6- 100%[===================>]   8.27M  --.-KB/s    in 0.04s

2022-10-03 17:30:54 (208 MB/s) - 'apache-maven-3.8.6-bin.tar.gz' saved [8676320/
8676320]

ubuntu@ubuntujenkins:/opt$
```

   Extract tar.gz file



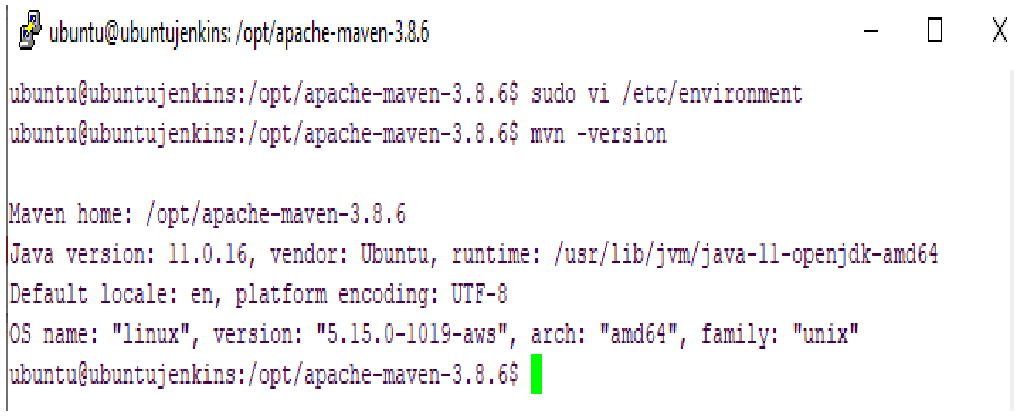
```
ubuntu@ubuntujenkins: /opt
apache-maven-3.8.6/lib/commons-lang3-3.8.1.jar
apache-maven-3.8.6/lib/maven-core-3.8.6.jar
apache-maven-3.8.6/lib/maven-repository-metadata-3.8.6.jar
apache-maven-3.8.6/lib/maven-artifact-3.8.6.jar
apache-maven-3.8.6/lib/maven-resolver-provider-3.8.6.jar
apache-maven-3.8.6/lib/maven-resolver-impl-1.6.3.jar
apache-maven-3.8.6/lib/maven-resolver-spi-1.6.3.jar
apache-maven-3.8.6/lib/org.eclipse.sisu.inject-0.3.5.jar
apache-maven-3.8.6/lib/plexus-interpolation-1.26.jar
apache-maven-3.8.6/lib/plexus-component-annotations-2.1.0.jar
apache-maven-3.8.6/lib/maven-compat-3.8.6.jar
apache-maven-3.8.6/lib/wagon-provider-api-3.5.1.jar
apache-maven-3.8.6/lib/org.eclipse.sisu.plexus-0.3.5.jar
apache-maven-3.8.6/lib/commons-cli-1.4.jar
apache-maven-3.8.6/lib/wagon-http-3.5.1-shaded.jar
apache-maven-3.8.6/lib/jcl-over-slf4j-1.7.36.jar
apache-maven-3.8.6/lib/wagon-file-3.5.1.jar
apache-maven-3.8.6/lib/maven-resolver-connector-basic-1.6.3.jar
apache-maven-3.8.6/lib/maven-resolver-transport-wagon-1.6.3.jar
apache-maven-3.8.6/lib/maven-slf4j-provider-3.8.6.jar
apache-maven-3.8.6/lib/jansi-2.4.0.jar
ubuntu@ubuntujenkins:/opt$ ls
apache-maven-3.8.6  apache-maven-3.8.6-bin.tar.gz
ubuntu@ubuntujenkins:/opt$
```

Let’s Add Maven to path:: i.e M2_HOME=path to Maven in /etc/environment



```
ubuntu@ubuntujenkins: /opt/apache-maven-3.8.6

JAVA_HOME="/usr/lib/jvm/java-1.11.0-openjdk-amd64"
M2_HOME="/opt/apache-maven-3.8.6"
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local
/games:/snap/bin:$JAVA_HOME:$M2_HOME/bin"
~
```

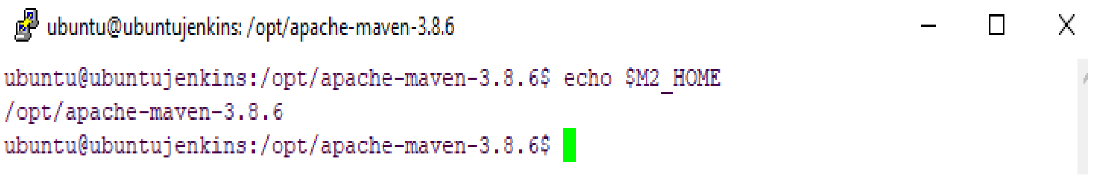
Refresh environment file with sudo . /etc/environment



```
ubuntu@ubuntujenkins: /opt/apache-maven-3.8.6
ubuntu@ubuntujenkins:/opt/apache-maven-3.8.6$ sudo vi /etc/environment
ubuntu@ubuntujenkins:/opt/apache-maven-3.8.6$ mvn -version

Maven home: /opt/apache-maven-3.8.6
Java version: 11.0.16, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.15.0-1019-aws", arch: "amd64", family: "unix"
ubuntu@ubuntujenkins:/opt/apache-maven-3.8.6$
```

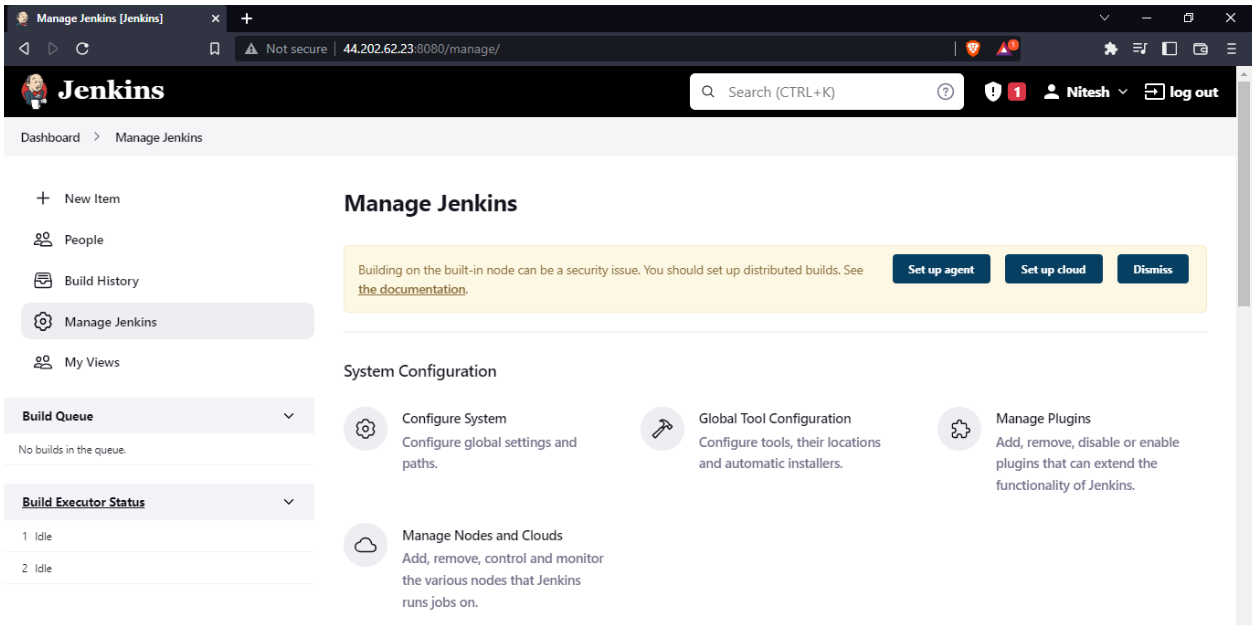
Configuring Maven in Jenkins:

1. Copy the maven path



```
ubuntu@ubuntujenkins: /opt/apache-maven-3.8.6

ubuntu@ubuntujenkins:/opt/apache-maven-3.8.6$ echo $M2_HOME
/opt/apache-maven-3.8.6
ubuntu@ubuntujenkins:/opt/apache-maven-3.8.6$
```

2. On Jenkins Select Manage Jenkins



Manage Jenkins [Jenkins]

Not secure | 44.202.62.23:8080/manage/

Jenkins

Search (CTRL+K) Nitesh log out

Dashboard > Manage Jenkins

New Item
People
Build History
Manage Jenkins
My Views

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

Set up agent | Set up cloud | Dismiss

System Configuration

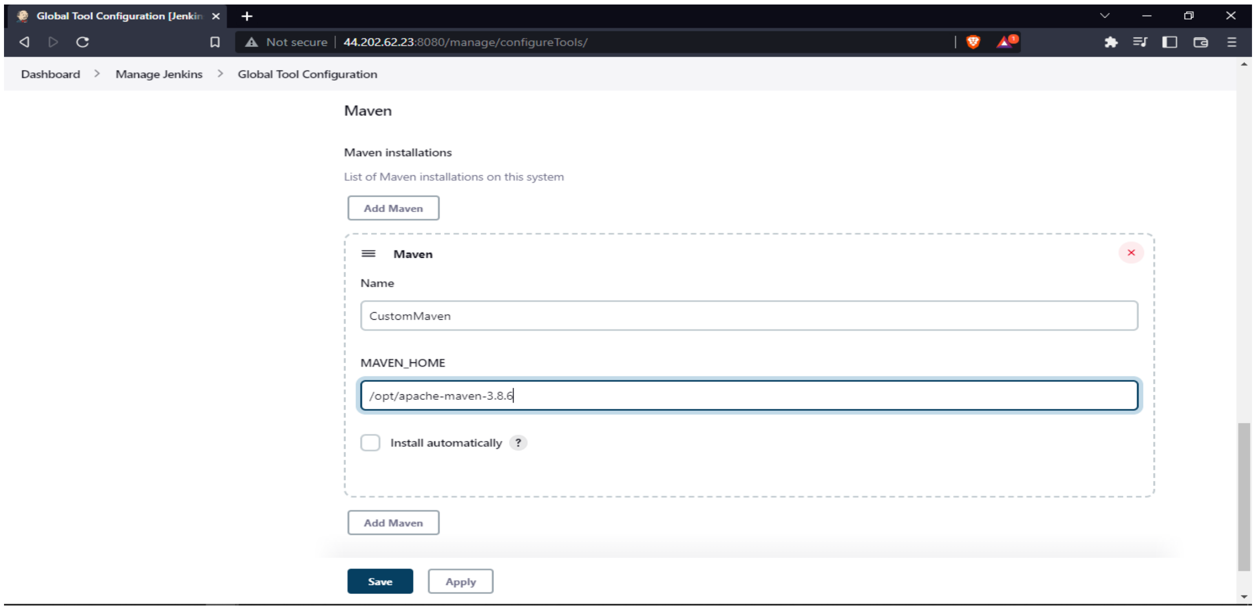
Configure System
Configure global settings and paths.

Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

3. Go to Global Tool Configuration, Scroll down to Maven Installation and Add new .
   Give the name and paste the path you copied I have given name as CustomMaven.



Global Tool Configuration [Jenkins]

Not secure | 44.202.62.23:8080/manage/configureTools/

Dashboard > Manage Jenkins > Global Tool Configuration

Maven

Maven installations

List of Maven installations on this system

Add Maven

Maven

Name

CustomMaven

MAVEN_HOME

/opt/apache-maven-3.8.6

Install automatically ?

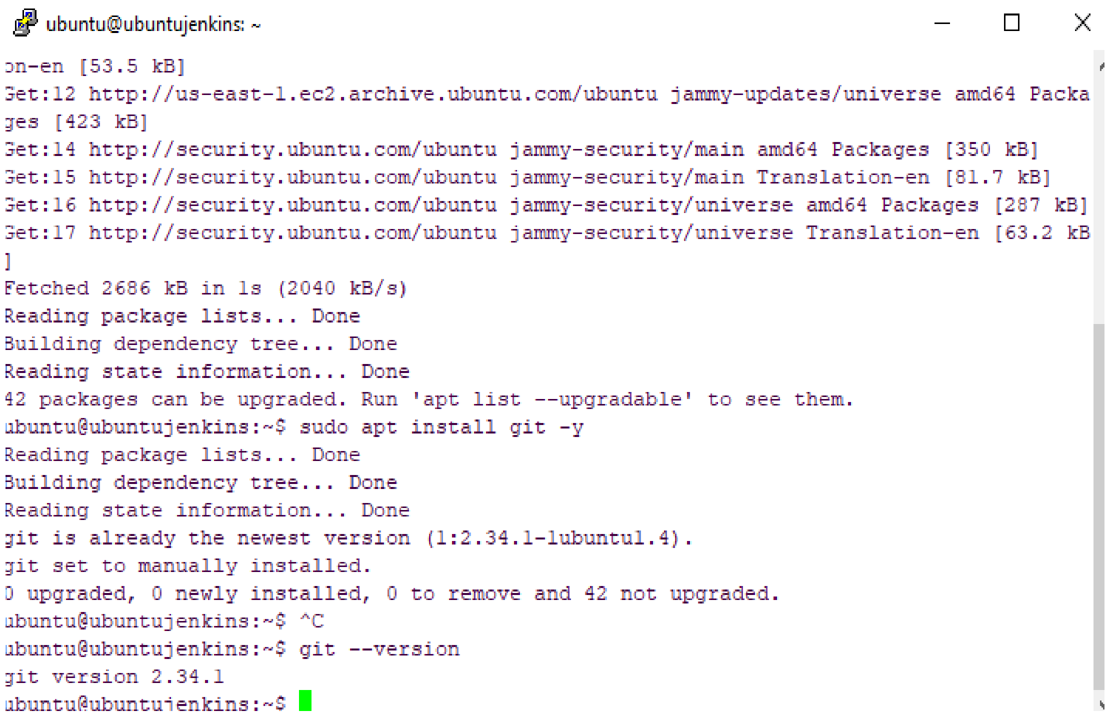
Add Maven

Save | Apply

4. Click on save.

Installing Git on UbuntuJenkinsServer

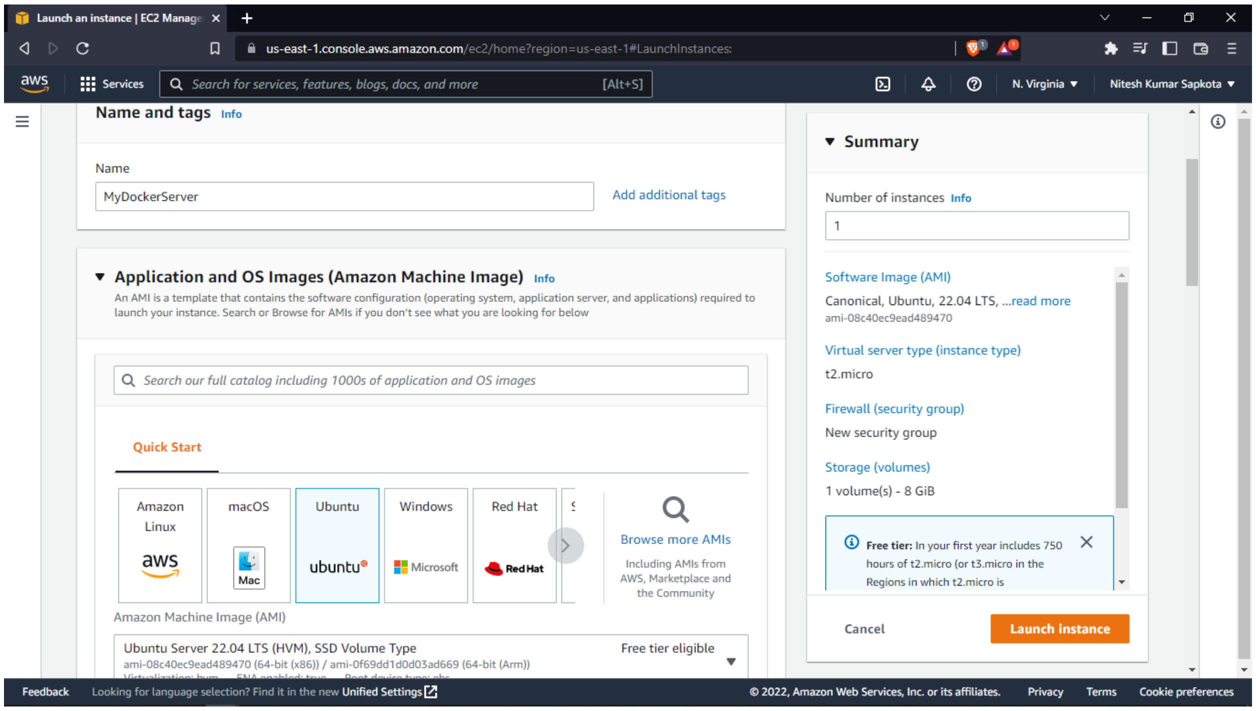
1. Sudo apt update
2. Sudo apt install git –y
3. Git –version



```
ubuntu@ubuntujenkins: ~
on-en [53.5 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packa
ges [423 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [350 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [81.7 kB]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [287 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [63.2 kB
]
Fetched 2686 kB in 1s (2040 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
42 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ubuntujenkins:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.4).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.
ubuntu@ubuntujenkins:~$ ^C
ubuntu@ubuntujenkins:~$ git --version
git version 2.34.1
ubuntu@ubuntujenkins:~$
```

Setting Docker Server

Creating AWS EC2 Docker Machine:



Name and tags Info

Name: MyDockerServer

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

Quick Start: Amazon Linux, macOS, Ubuntu, Windows, Red Hat

Amazon Machine Image (AMI): Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-08c40ec9ead489470 (64-bit (x86)) / ami-0f69dd1d0d03ad669 (64-bit (Arm))

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more
ami-08c40ec9ead489470

Virtual server type (instance type): t2.micro

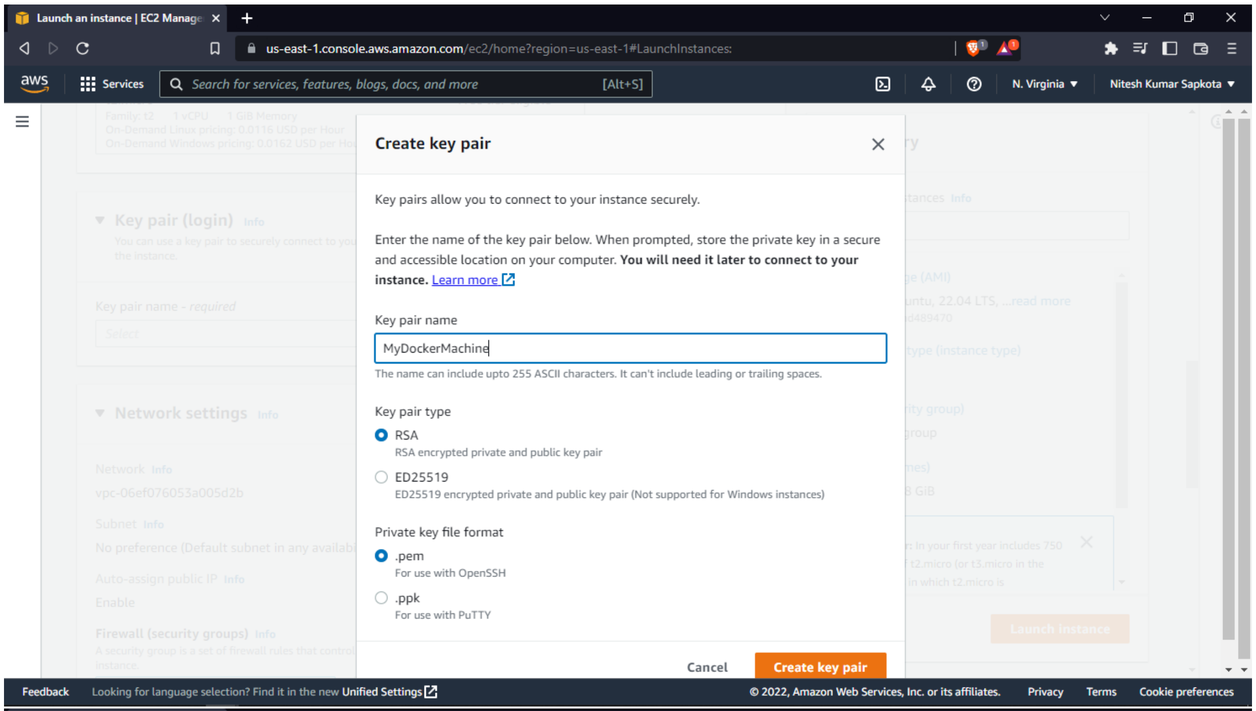
Firewall (security group): New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is

Cancel | Launch instance

Create a new key pair with .pem



Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. 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

Key pair name

MyDockerMachine

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

Key pair type

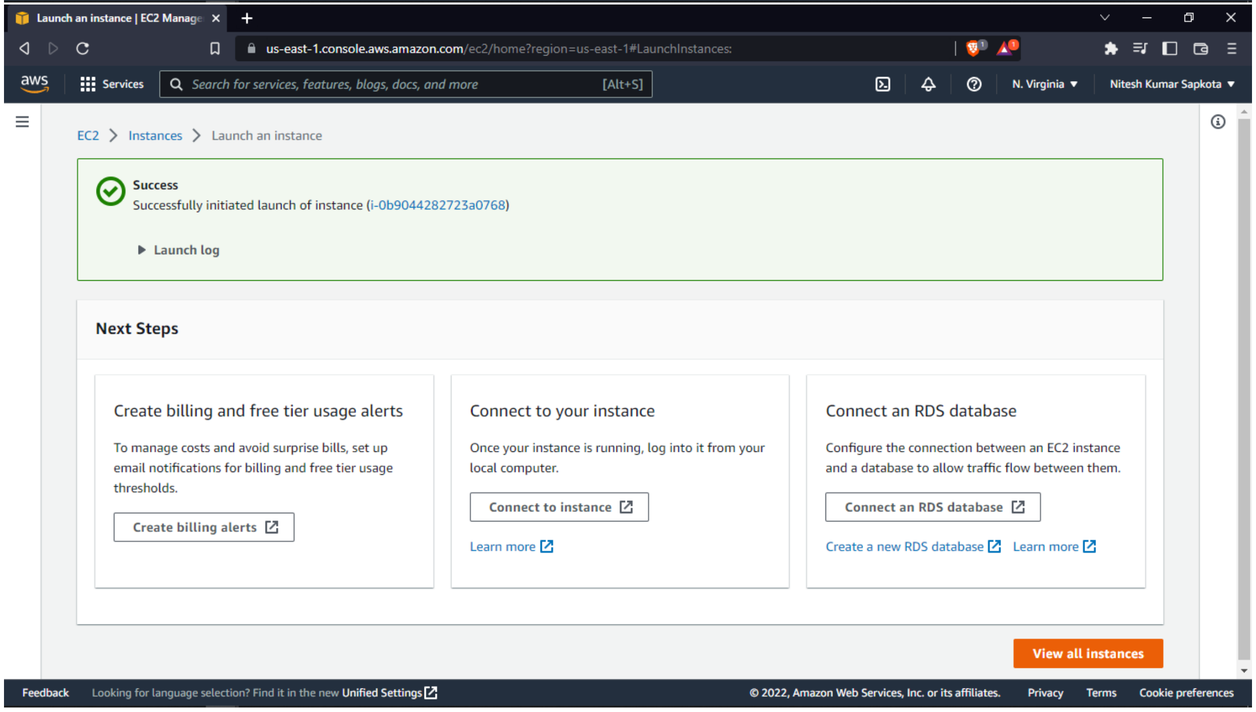
[✓ RSA] RSA encrypted private and public key pair

ED25519 ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

[✓ .pem] For use with OpenSSH

.ppk For use with PuTTY



EC2 > Instances > Launch an instance

Success

Successfully initiated launch of instance (i-0b9044282723a0768)

Launch log

Next Steps

Create billing and free tier usage alerts

To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

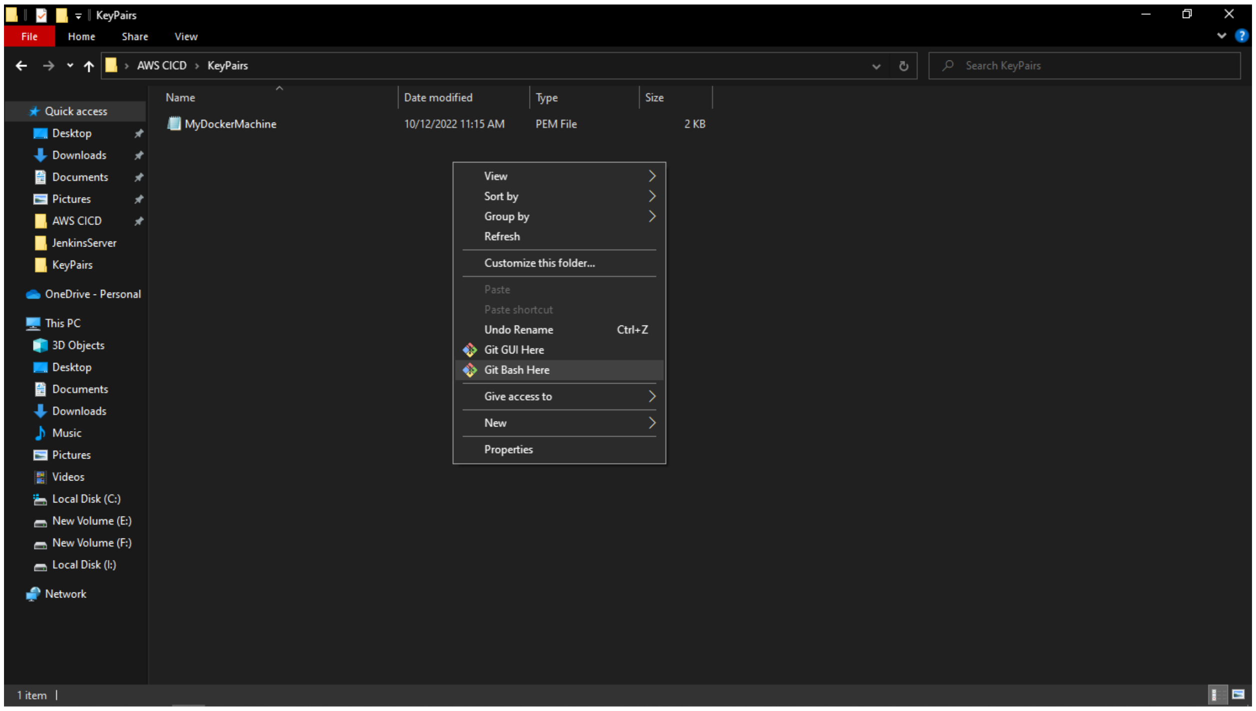
Connect to your instance

Once your instance is running, log into it from your local computer.

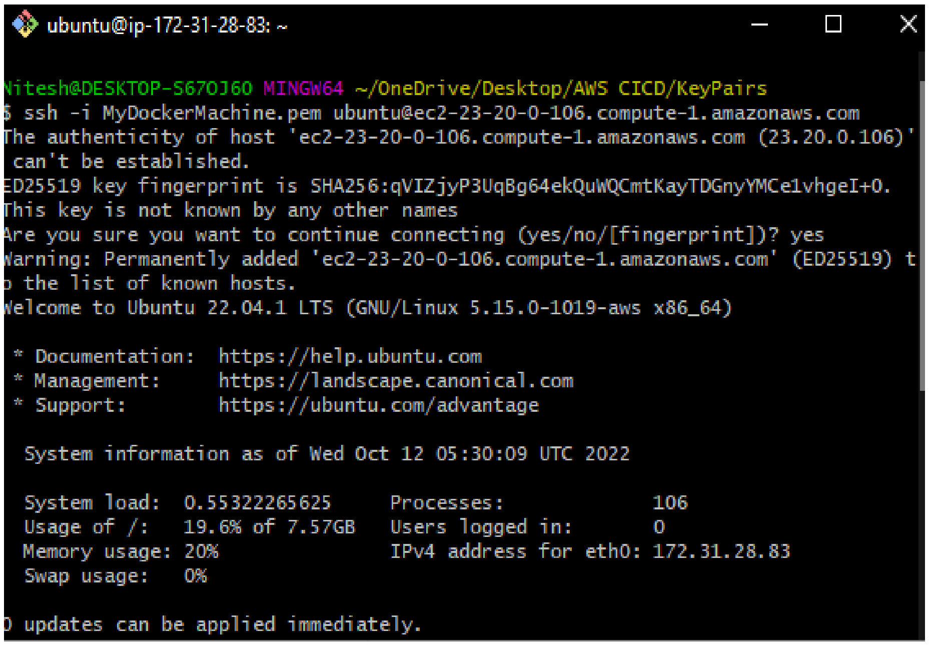
Connect an RDS database

Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Let’s Connect using Git Bash Terminal:



For Connection :: ssh –i MyDockerMachine.pem ubuntu@publicPv4



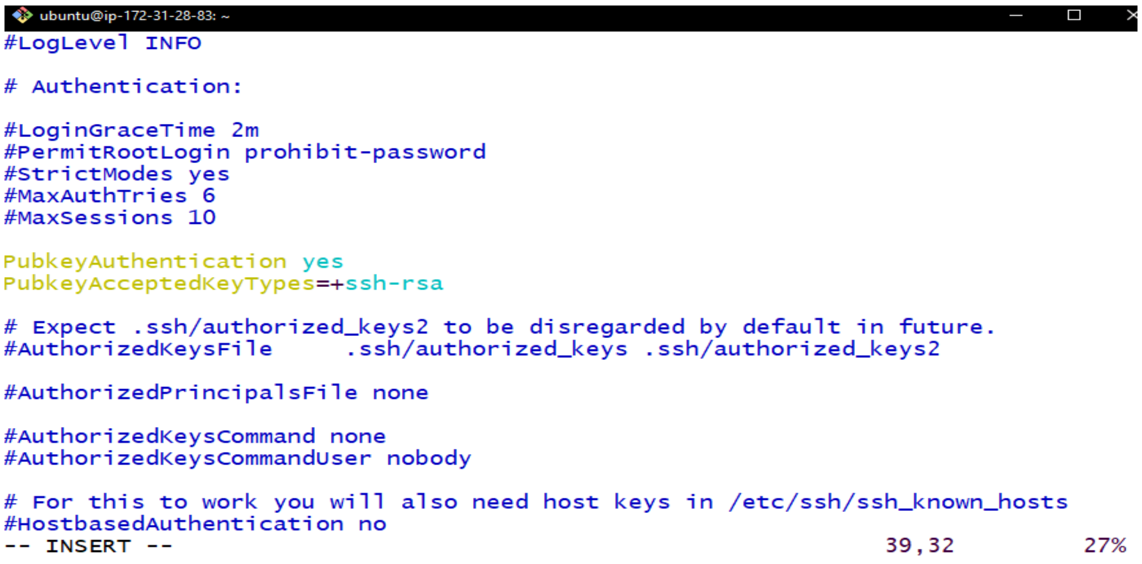
```
Nitesh@DESKTOP-S670J60 MINGW64 ~/OneDrive/Desktop/AWS CICD/KeyPairs
$ ssh -i MyDockerMachine.pem ubuntu@ec2-23-20-0-106.compute-1.amazonaws.com
The authenticity of host 'ec2-23-20-0-106.compute-1.amazonaws.com (23.20.0.106)'
 can't be established.
ED25519 key fingerprint is SHA256:qVIZjyP3UqBg64ekQuWQCmtKayTDGnyYMCe1vhgeI+0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-23-20-0-106.compute-1.amazonaws.com' (ED25519) t
o the list of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

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

  System information as of Wed Oct 12 05:30:09 UTC 2022

  System load:  0.55322265625     Processes:             106
  Usage of /:   19.6% of 7.57GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 172.31.28.83
  Swap usage:   0%

0 updates can be applied immediately.
```

By default it doesnot accept public key authentication and rsa type of key so add

PubkeyAuthentication yes

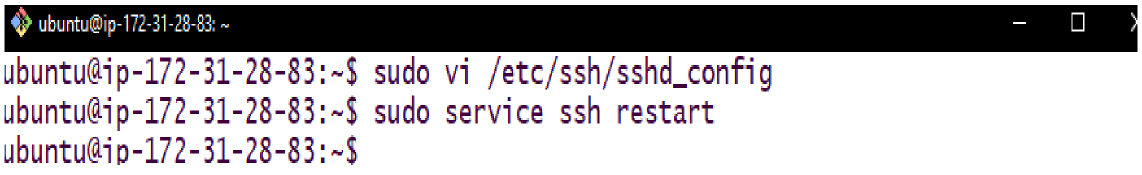
PubkeyAcceptedKeyTypes=+ssh-rsa

On /etc/ssh/sshd_config


```
ubuntu@ip-172-31-28-83: ~
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes
PubkeyAcceptedKeyTypes=+ssh-rsa

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
-- INSERT --                                                 39,32         27%
```

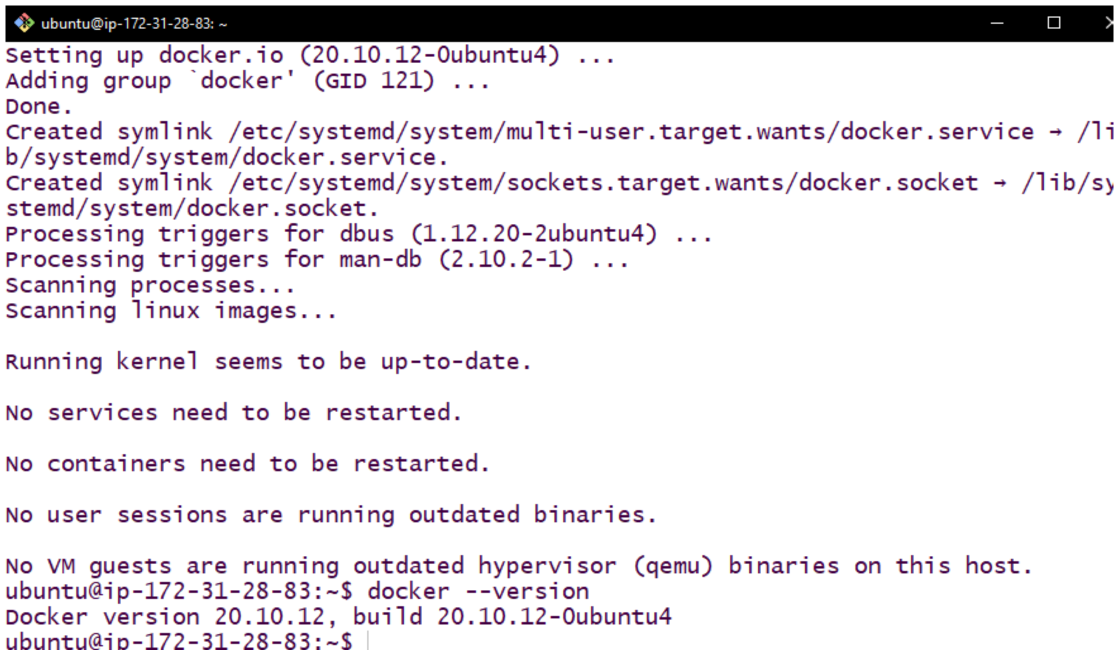
Let's restart ssh service with

Sudo service ssh restart


```
ubuntu@ip-172-31-28-83: ~
ubuntu@ip-172-31-28-83:~$ sudo vi /etc/ssh/sshd_config
ubuntu@ip-172-31-28-83:~$ sudo service ssh restart
ubuntu@ip-172-31-28-83:~$
```

Let's Install Docker on this machine.

sudo apt update
sudo apt install docker.io


```
ubuntu@ip-172-31-28-83: ~
Setting up docker.io (20.10.12-0ubuntu4) ...
Adding group `docker' (GID 121) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /li
b/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/sy
stemd/system/docker.socket.
Processing triggers for dbus (1.12.20-2ubuntu4) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-28-83:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
ubuntu@ip-172-31-28-83:~$
```

Let’s create a directory in home:;

Sudo mkdir project

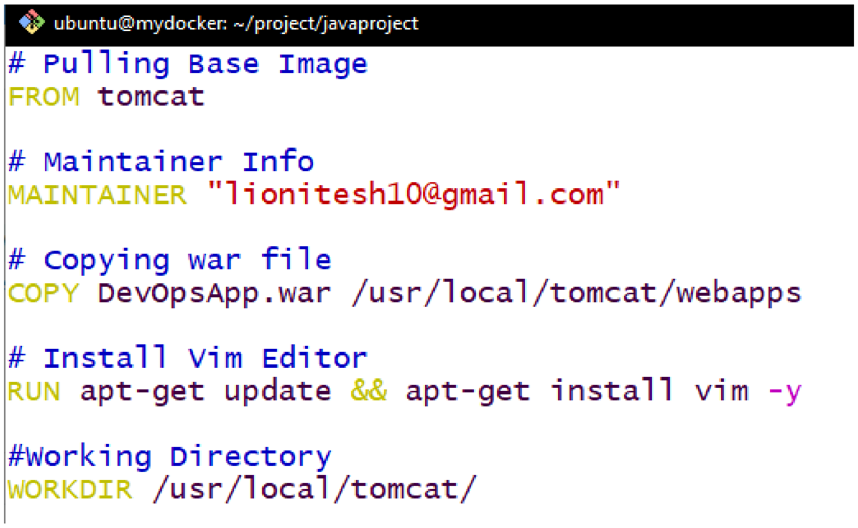
Cd project

Sudo mkdir javaproject

Cd javaproject

Create a Dockerfile inside

Sudo vi Dockerfile



ubuntu@mydocker: ~/project/javaproject

```
# Pulling Base Image
FROM tomcat

# Maintainer Info
MAINTAINER "lionitesh10@gmail.com"

# Copying war file
COPY DevOpsApp.war /usr/local/tomcat/webapps

# Install Vim Editor
RUN apt-get update && apt-get install vim -y

#Working Directory
WORKDIR /usr/local/tomcat/
```

FROM tomcat

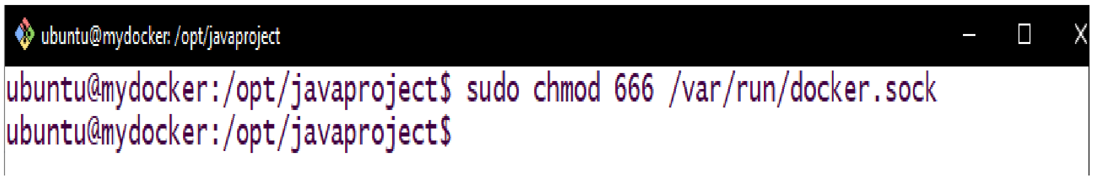
COPY DevOpsApp.war /usr/local/tomcat/webapaps

RUN apt-get update && apt-get install vim – y

WORKDIR /usr/local/tomcat

Change permission of docker.sock to run daemon without sudo privileges.

Sudo chmod 666 /var/run/docker.sock

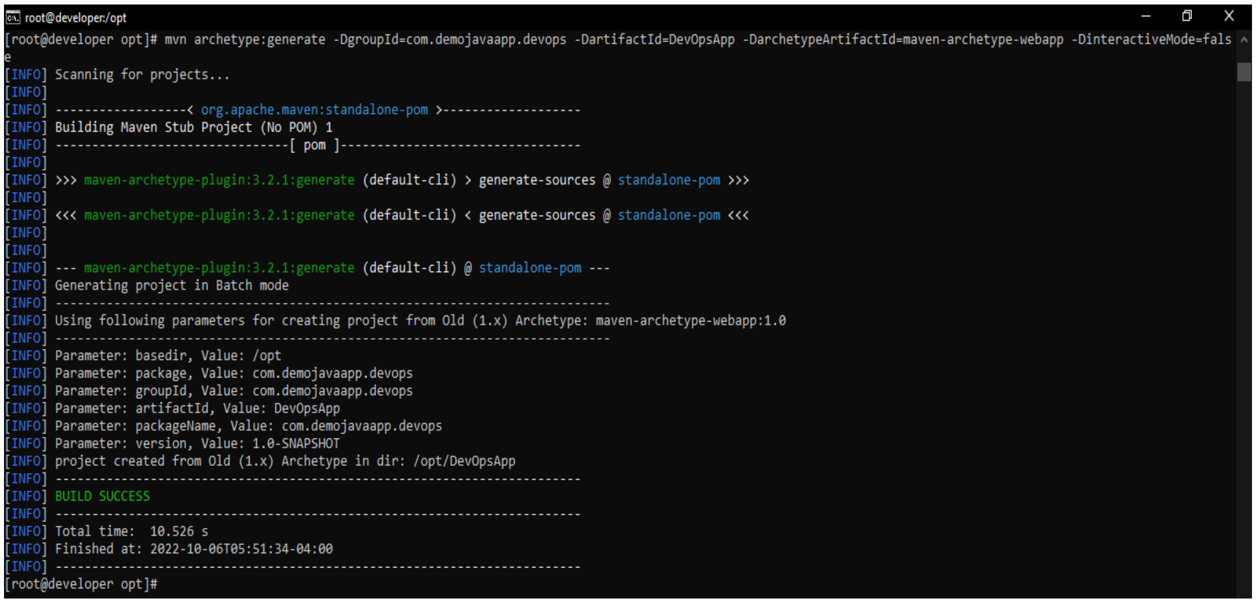


ubuntu@mydocker: /opt/javaproject

```
ubuntu@mydocker:/opt/javaproject$ sudo chmod 666 /var/run/docker.sock
ubuntu@mydocker:/opt/javaproject$
```

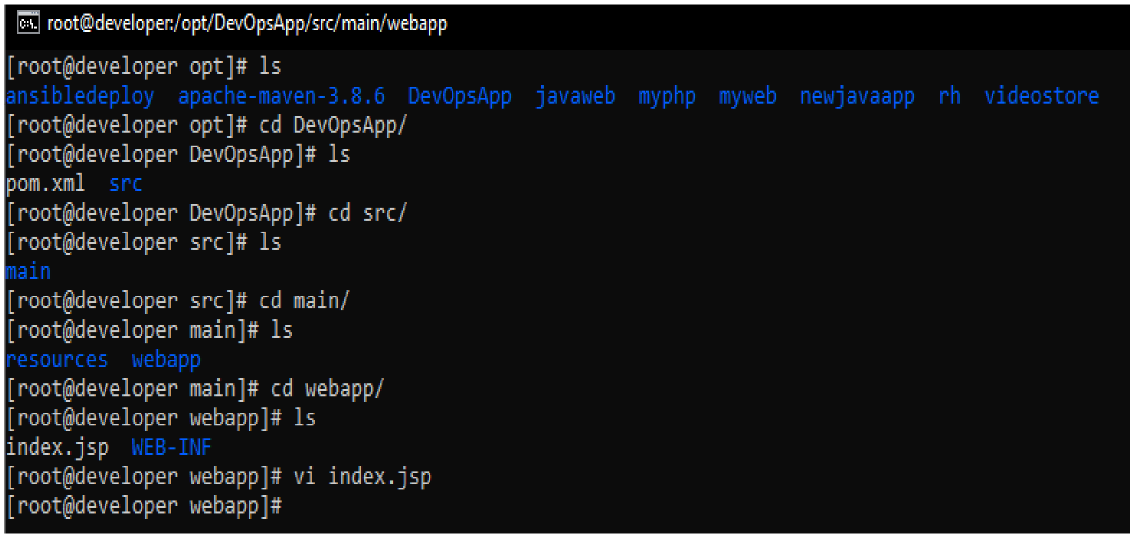
Now , lets create a Maven web project on your own machine using following command.

mvn archetype:generate -DgroupId=com.demojavaapp.devops -DartifactId=DevOpsApp -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false

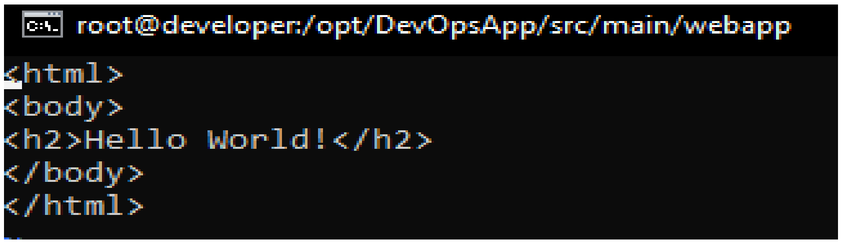


```
root@developer/opt
[root@developer opt]# mvn archetype:generate -DgroupId=com.demojavaapp.devops -DartifactId=DevOpsApp -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=fals
e
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< org.apache.maven:standalone-pom >-------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] --------------------------------[ pom ]---------------------------------
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO]
[INFO] --- maven-archetype-plugin:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: basedir, Value: /opt
[INFO] Parameter: package, Value: com.demojavaapp.devops
[INFO] Parameter: groupId, Value: com.demojavaapp.devops
[INFO] Parameter: artifactId, Value: DevOpsApp
[INFO] Parameter: packageName, Value: com.demojavaapp.devops
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /opt/DevOpsApp
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  10.526 s
[INFO] Finished at: 2022-10-06T05:51:34-04:00
[INFO] ------------------------------------------------------------------------
[root@developer opt]#
```

Let's navigate inside DevOpsApp/src/main/webapp and watch index.jsp



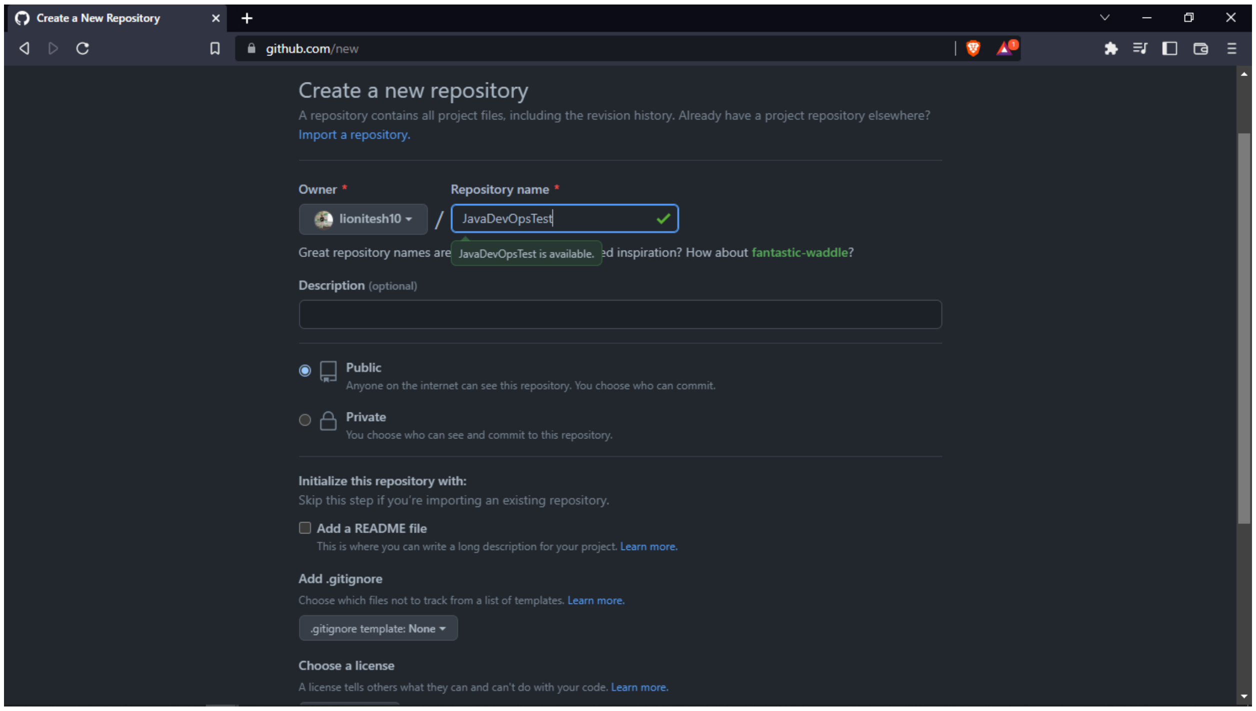
```
root@developer:/opt/DevOpsApp/src/main/webapp
[root@developer opt]# ls
ansibledeploy  apache-maven-3.8.6  DevOpsApp  javaweb  myphp  myweb  newjavaapp  rh  videostore
[root@developer opt]# cd DevOpsApp/
[root@developer DevOpsApp]# ls
pom.xml  src
[root@developer DevOpsApp]# cd src/
[root@developer src]# ls
main
[root@developer src]# cd main/
[root@developer main]# ls
resources  webapp
[root@developer main]# cd webapp/
[root@developer webapp]# ls
index.jsp  WEB-INF
[root@developer webapp]# vi index.jsp
[root@developer webapp]#
```



```
root@developer:/opt/DevOpsApp/src/main/webapp
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

Pushing project in github.

Create a new github repo.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * lionitesh10 / Repository name * JavaDevOpsTest

Great repository names are JavaDevOpsTest is available. ed inspiration? How about fantastic-waddle?

Description (optional)

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

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

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file — This is where you can write a long description for your project. Learn more.

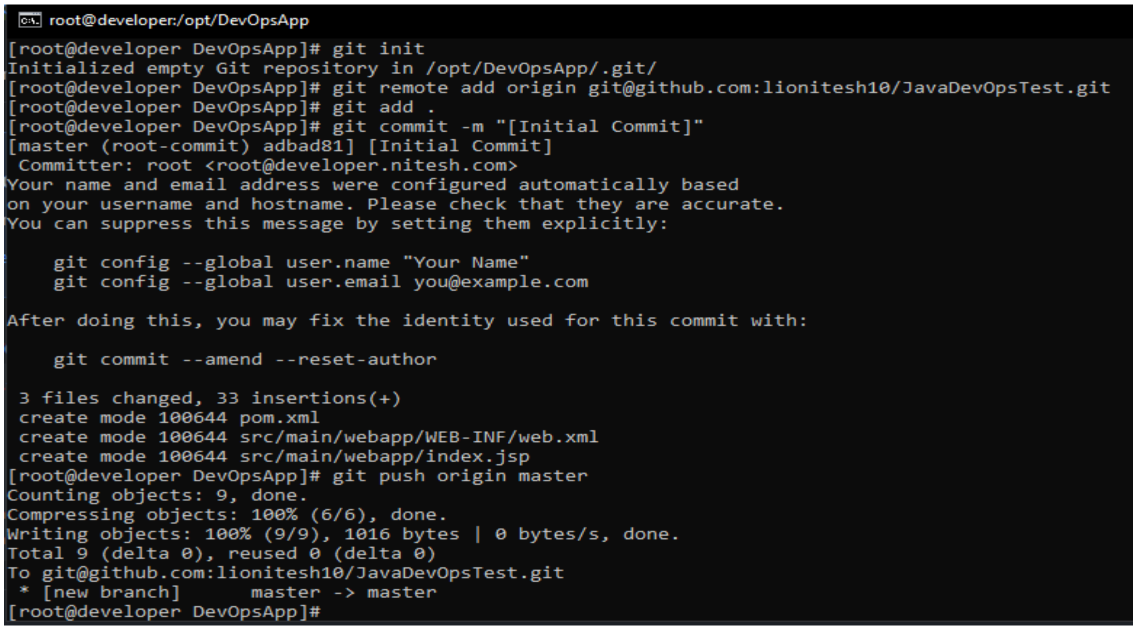
Add .gitignore — Choose which files not to track from a list of templates. Learn more.

.gitignore template: None

Choose a license — A license tells others what they can and can't do with your code. Learn more.

Git Commands to Push be inside DevOpsApp folder.

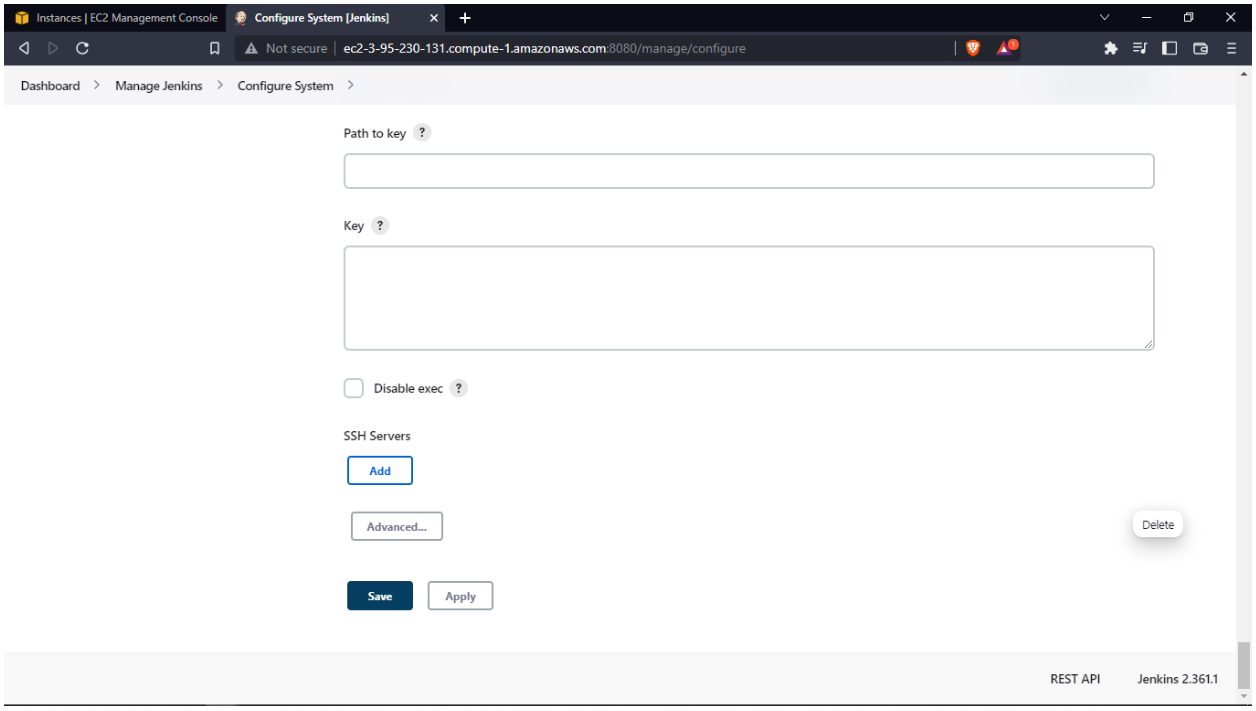
Git init
Git remote add origin git@github.com:repourl
Git add .
Git commit –m "[Message]"
Git push origin master



```
root@developer:/opt/DevOpsApp
[root@developer DevOpsApp]# git init
Initialized empty Git repository in /opt/DevOpsApp/.git/
[root@developer DevOpsApp]# git remote add origin git@github.com:lionitesh10/JavaDevOpsTest.git
[root@developer DevOpsApp]# git add .
[root@developer DevOpsApp]# git commit -m "[Initial Commit]"
[master (root-commit) adbad81] [Initial Commit]
 Committer: root <root@developer.nitesh.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

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

    git commit --amend --reset-author

 3 files changed, 33 insertions(+)
 create mode 100644 pom.xml
 create mode 100644 src/main/webapp/WEB-INF/web.xml
 create mode 100644 src/main/webapp/index.jsp
[root@developer DevOpsApp]# git push origin master
Counting objects: 9, done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 1016 bytes | 0 bytes/s, done.
Total 9 (delta 0), reused 0 (delta 0)
To git@github.com:lionitesh10/JavaDevOpsTest.git
 * [new branch]      master -> master
[root@developer DevOpsApp]#
```

Configuring SSH server in Jenkins

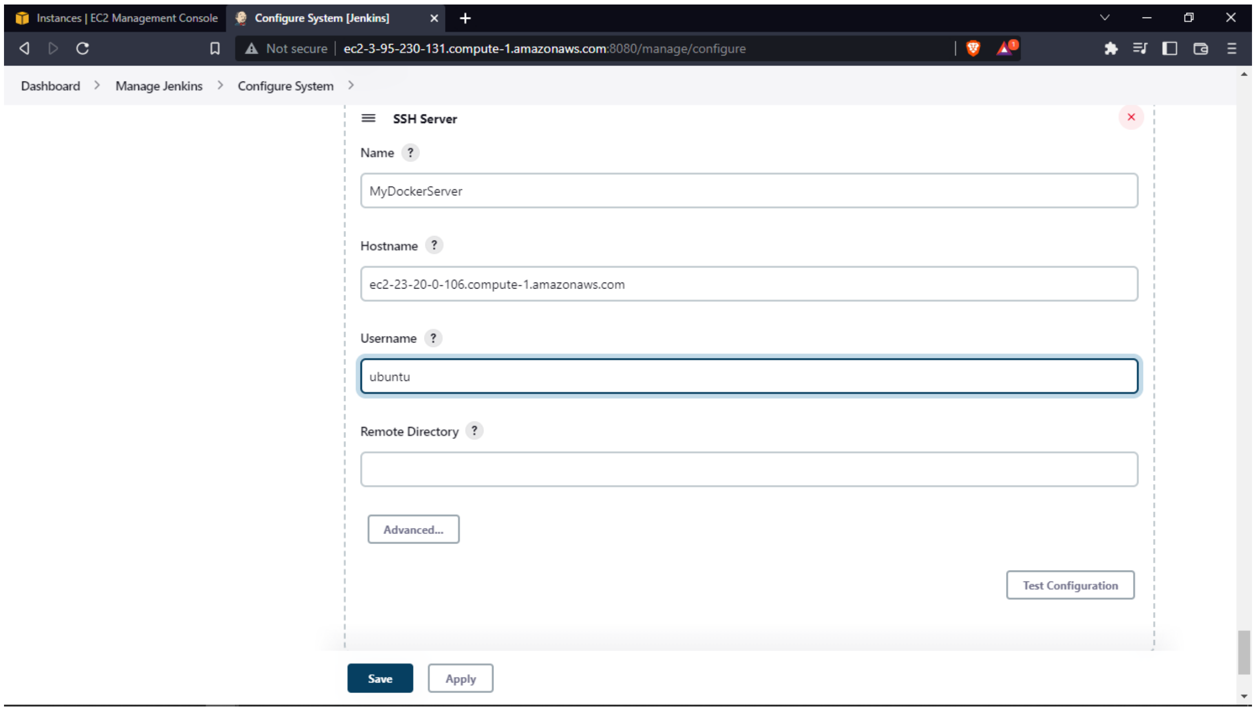
Navigate to Manage Jenkins and Configure System and scroll down to SSH Servers. Click on Add.



Give Name:: MyDockerServer

Hostname:: public IPv4 of MyDockerServer or DNS.

Username:: ubuntu


Instances | EC2 Management Console
Configure System [Jenkins]
Not secure
ec2-3-95-230-131.compute-1.amazonaws.com:8080/manage/configure
Dashboard
Manage Jenkins
Configure System
SSH Server
Advanced...
Test Configuration
Save
Apply

Name: MyDockerServer

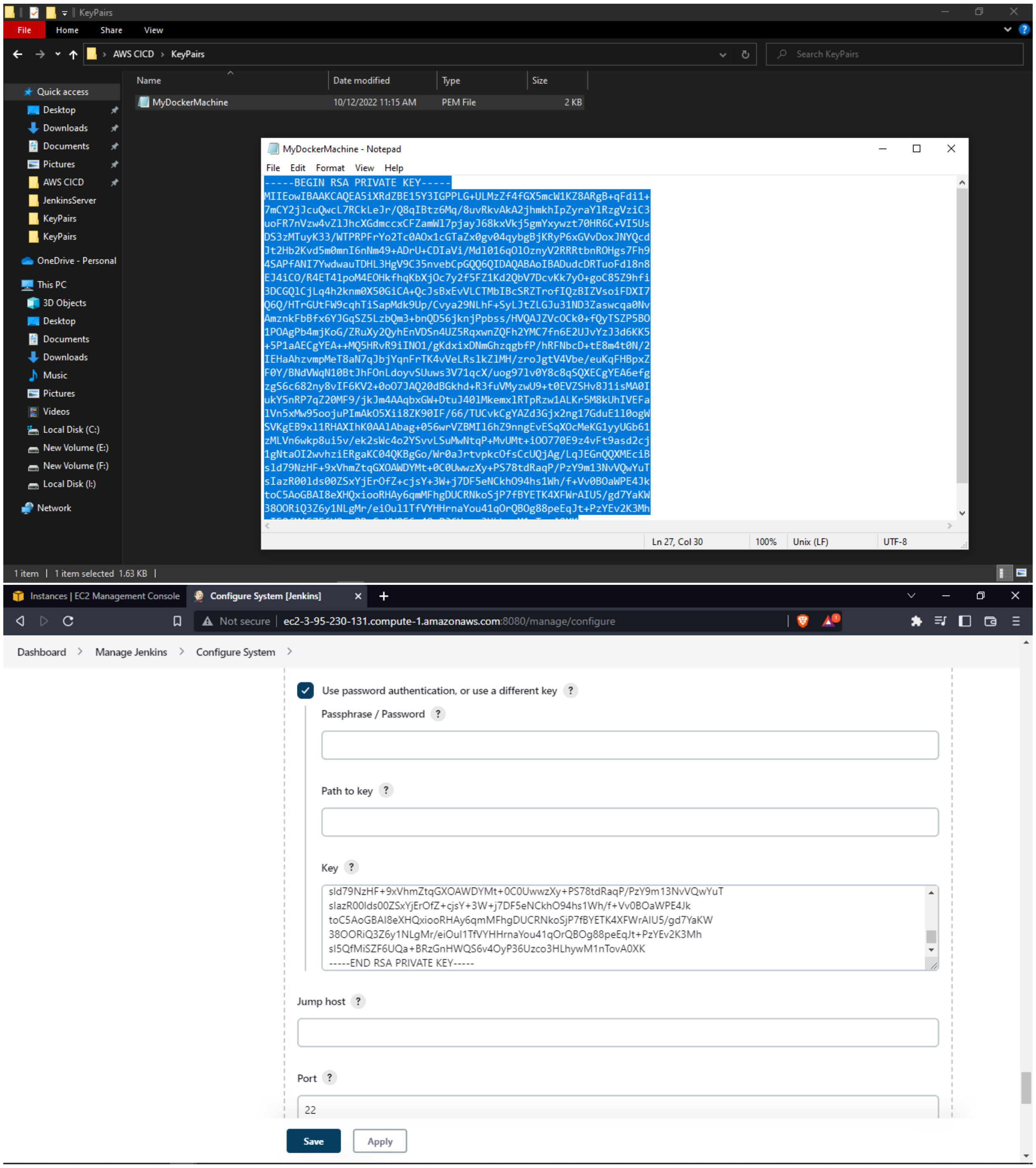
Hostname: ec2-23-20-0-106.compute-1.amazonaws.com

Username: ubuntu

Remote Directory:

Click on Advanced and Check Use password authentication or a different key.

Open the PEM key with notepad or any other editors copy the content and paste it into the key.



Name	Date modified	Type	Size
MyDockerMachine	10/12/2022 11:15 AM	PEM File	2 KB

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA5iXRdZBE15Y3IGPPLG+ULMzZf4fGX5mcW1KZ8ARgB+qFdi1+
7mCY2jJcuQwcL7RCkLeJr/Q8qIBtz6Mq/8uvRkvAkA2jhmkhIpZyraY1RzgVziC3
uoFR7nVzw4vZlJhcXGdmccxCFZamWl7pjayJ68kxVkj5gmYxywzt70HR6C+VI5Us
DS3zMTuyK33/WTPRPFrYo2Tc0AOx1cGTaZx0gv04qybgBjKRyP6xGVvDoxJNYQcd
Jt2Hb2Kvd5m0mnI6nNm49+ADrU+CDIaVi/Md1016qO1OznyV2RRRtbnROHgs7Fh9
4SAPfANI7YwdwauTDHL3HgV9C35nvebCpGQQ6QIDAQABAoIBADudcDRTuoFdl8n8
EJ4iCO/R4ET4lpoM4EOHkfhqKbXjOc7y2f5FZ1Kd2QbV7DcvKk7yO+goC85Z9hfi
3DCGQlCjLq4h2knm0X50GiCA+QcJsBxEvVLCTMbIBcSRZTrofIQzBIZVsoiFDXI7
Q6Q/HTrGUtFW9cqhTiSapMdk9Up/Cvya29NLhF+SyLJtZLGJu31ND3Zaswcqa0Nv
AmznkFbBfx6YJGqSZ5LzbQm3+bnQD56jknjPpbss/HVQAJZVcOCk0+fQyTSZP5BO
1POAgPb4mjKoG/ZRuXy2QyhEnVDSn4UZ5RqxwnZQFh2YMC7fn6E2UJvYzJ3d6KK5
+5P1aAECgYEA++MQ5HRvR9iINO1/gKdxixDNmGhzqgbfP/hRFNbcD+tE8m4t0N/2
IEHaAhzvmpMeT8aN7qJbjYqnFrTK4vVeLRslkZlMH/zroJgtV4Vbe/euKqFHBpxZ
F0Y/BNdVWqN10BtJhFOnLdoyvSUuws3V71qcX/uog97lv0Y8c8qSQXECgYEA6efg
zgS6c682ny8vIF6KV2+0oO7JAQ20dBGkhd+R3fuVMyzwU9+t0EVZSHv8J1isMA0I
ukY5nRP7qZ20MF9/jkJm4AAqbxGW+DtuJ40lMkemxlRTpRzw1ALKr5M8kUhIVEFa
lVn5xMw95oojuPImAkO5Xii8ZK90IF/66/TUCvkCgYAZd3Gjx2ng17GduE1l0ogW
SVKgEB9xl1RHAXIhK0AAlAbag+056wrVZBMIl6hZ9nngEvESqXOcMeKG1yyUGb61
zMLVn6wkp8ui5v/ek2sWc4o2YSvvLSuMwNtqP+MvUMt+iO0770E9z4vFt9asd2cj
1gNtaOI2wvhziERgaKC04QKBgGo/Wr0aJrtvpkcOfsCcUQjAg/LqJEGnQQXMEciB
sld79NzHF+9xVhmZtqGXOAWDYMt+0C0UwwzXy+PS78tdRaqP/PzY9m13NvVQwYuT
sIazR00lds00ZSxYjErOfZ+cjsY+3W+j7DF5eNCkhO94hs1Wh/f+Vv0BOaWPE4Jk
toC5AoGBAI8eXHQxiooRHAy6qmMFhgDUCRNkoSjP7fBYETK4XFWrAIU5/gd7YaKW
38OORiQ3Z6y1NLgMr/eiOul1TfVYHHrnaYou41qOrQBOg88peEqJt+PzYEv2K3Mh
```

Use password authentication, or use a different key ✓

Passphrase / Password

Path to key

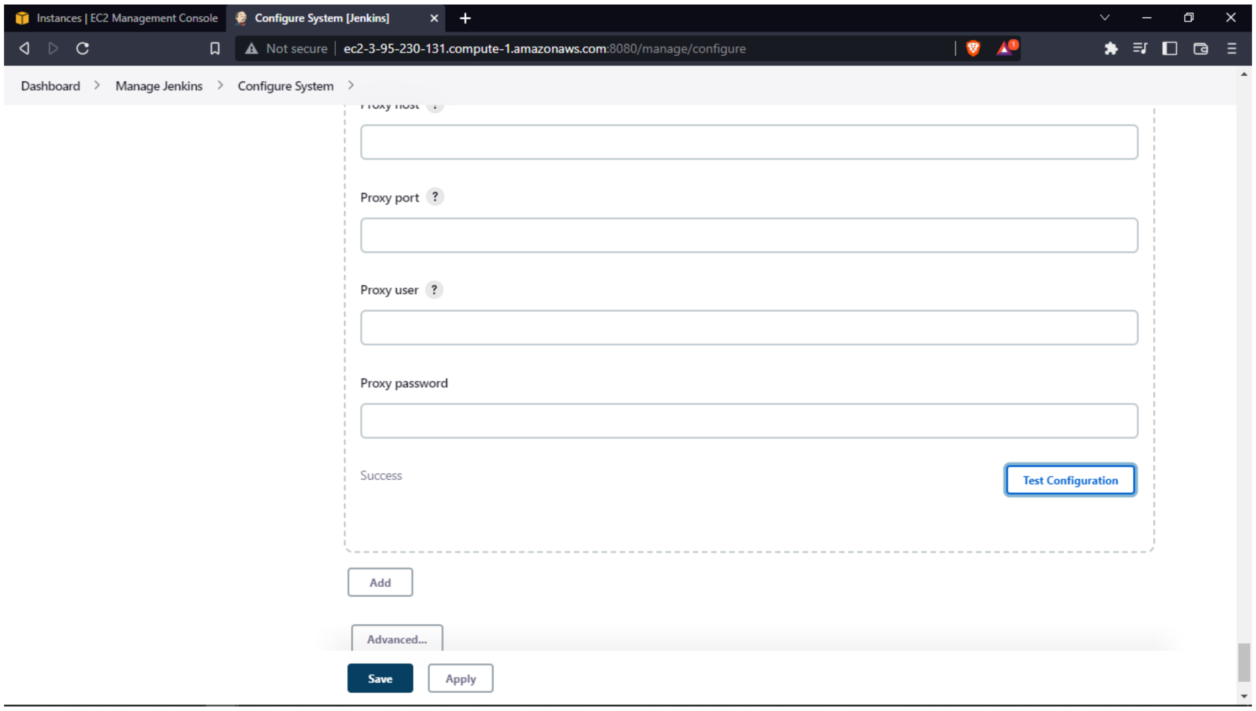
Key

```
sld79NzHF+9xVhmZtqGXOAWDYMt+0C0UwwzXy+PS78tdRaqP/PzY9m13NvVQwYuT
sIazR00lds00ZSxYjErOfZ+cjsY+3W+j7DF5eNCkhO94hs1Wh/f+Vv0BOaWPE4Jk
toC5AoGBAI8eXHQxiooRHAy6qmMFhgDUCRNkoSjP7fBYETK4XFWrAIU5/gd7YaKW
38OORiQ3Z6y1NLgMr/eiOul1TfVYHHrnaYou41qOrQBOg88peEqJt+PzYEv2K3Mh
sI5QfMiSZF6UQa+BRzGnHWQS6v4OyP36Uzco3HLhywM1nTovA0XK
-----END RSA PRIVATE KEY-----
```

Jump host

Port

22



Click on Test Configuration. Can see success.

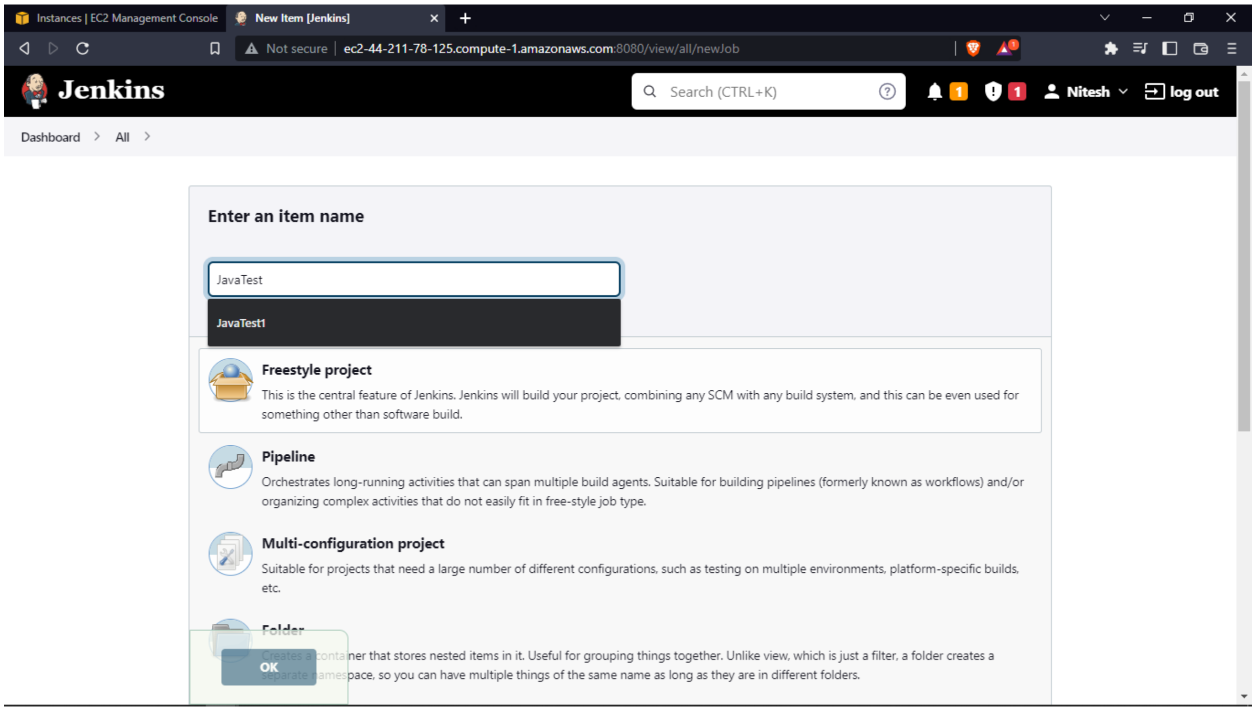
Click on Save

Now Creating Pipeline::

Our Pipeline :: Developer -Push→ Git→Jenkins-Pull :: (Build with Maven)→Send artifacts to DockerServer→Run Docker Container.

So Let's create a new job Click on New Item on Left.

Lets give the name as JavaTest1. Select Freestyle project and Click Ok.



Instances | EC2 Management Console | New Item [Jenkins]

Not secure | ec2-44-211-78-125.compute-1.amazonaws.com:8080/view/all/newJob

Jenkins | Search (CTRL+K) | Nitesh | log out

Dashboard > All >

Enter an item name

JavaTest

JavaTest1

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

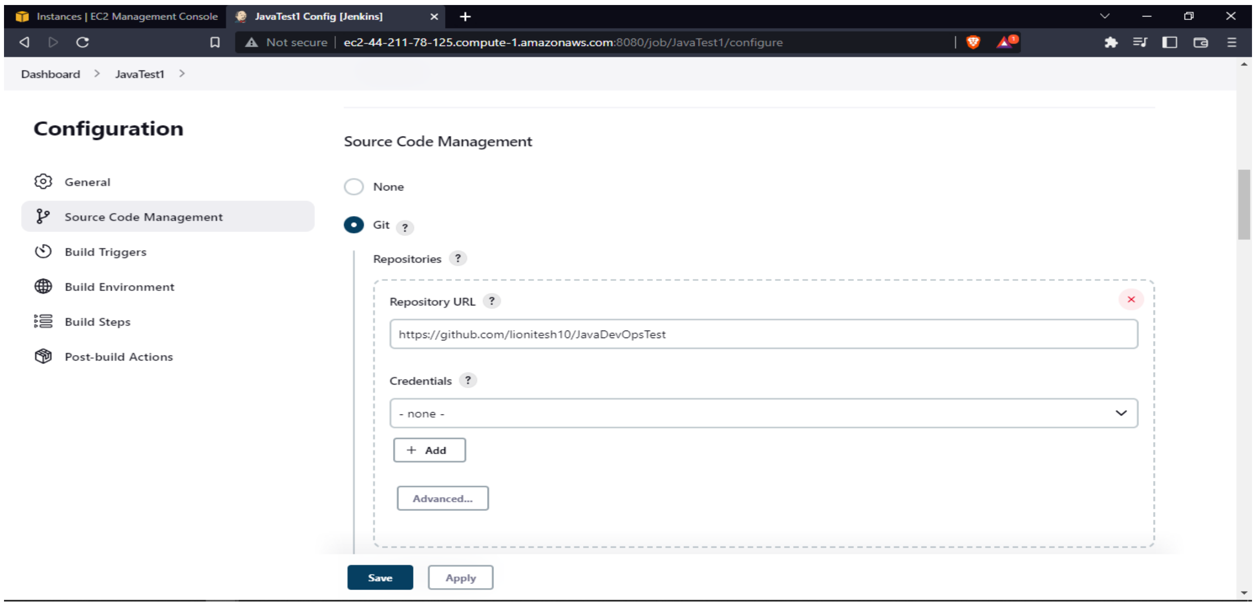
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Scroll to Source Code Management and on Repository url paste the github Url of the project.



Instances | EC2 Management Console | JavaTest1 Config [Jenkins]

Not secure | ec2-44-211-78-125.compute-1.amazonaws.com:8080/job/JavaTest1/configure

Dashboard > JavaTest1 >

Configuration

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

Source Code Management

None [✓ Git]

Repositories

Repository URL: https://github.com/lionitesh10/JavaDevOpsTest

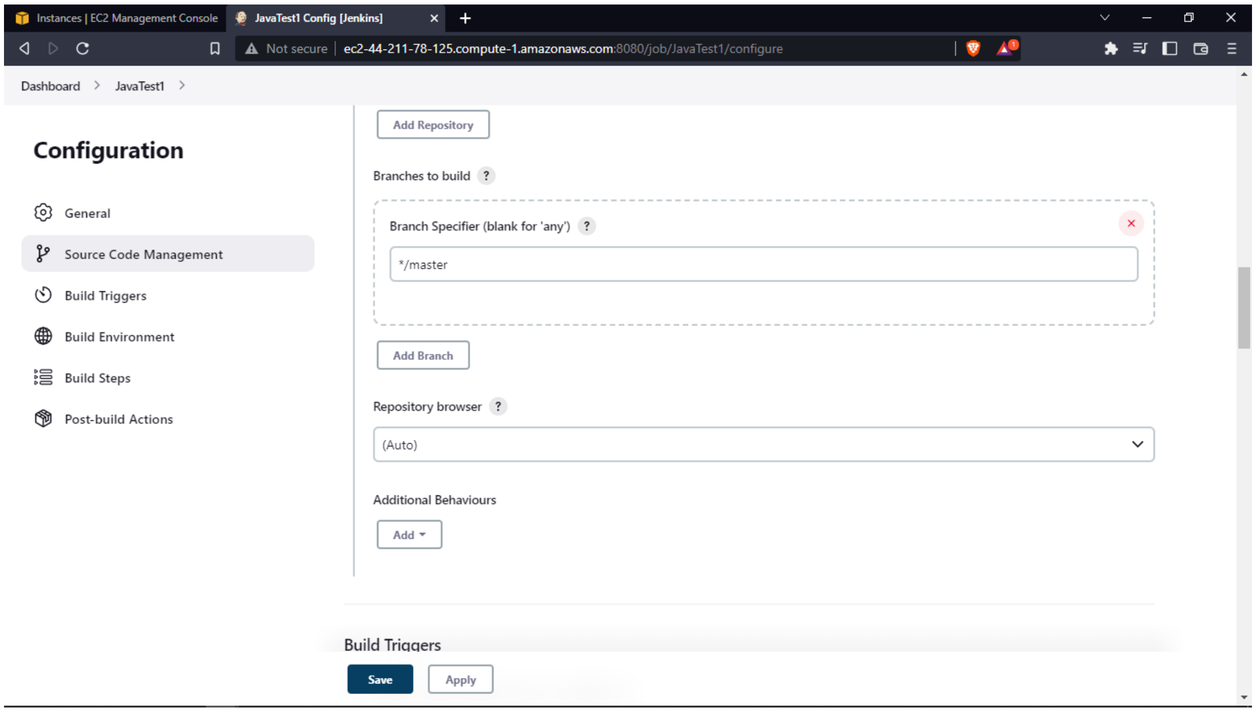
Credentials: - none -

Add

Advanced...

Save | Apply

Branches to Build Select */master by default.



Configuration

Branches to build

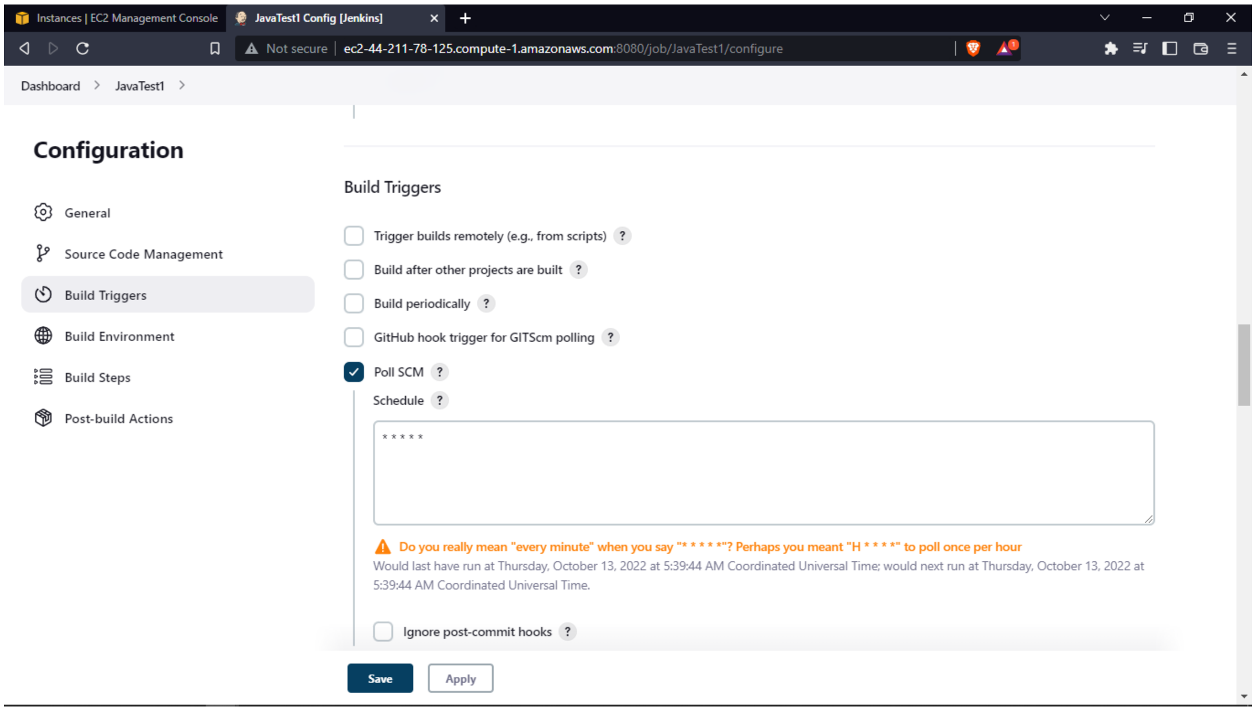
Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

Additional Behaviours

Build Triggers

On build triggers check on Poll SCM Enter * * * * * . It polls SCM every minute for change.



Build Triggers

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

Schedule

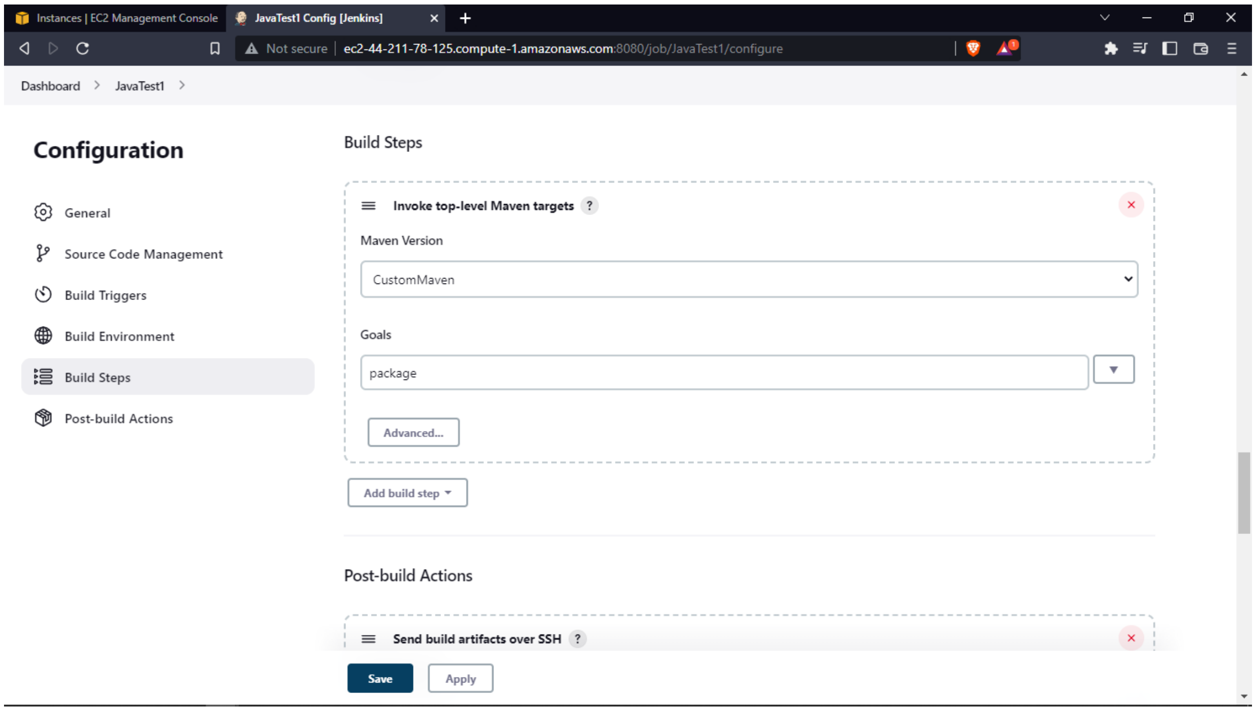
```
* * * * *
```

Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Thursday, October 13, 2022 at 5:39:44 AM Coordinated Universal Time; would next run at Thursday, October 13, 2022 at 5:39:44 AM Coordinated Universal Time.

- [ ] Ignore post-commit hooks

On Build Steps Select Maven Version as CustomMaven and enter goal as package as package command compiles and packages maven web application into war file.


Dashboard > JavaTest1 >

Configuration

Build Steps

Invoke top-level Maven targets

Maven Version: CustomMaven

Goals: package

Post-build Actions

Send build artifacts over SSH

On Post Build Actions Select Send Build Artifacts over SSH.

Select MyDockerServer you previously created.

On Transfers::

Source files::

Source file resides under /var/lib/Jenkins/workspace/{Job Name}

By default its path is there and after package war file is build inside target folder.

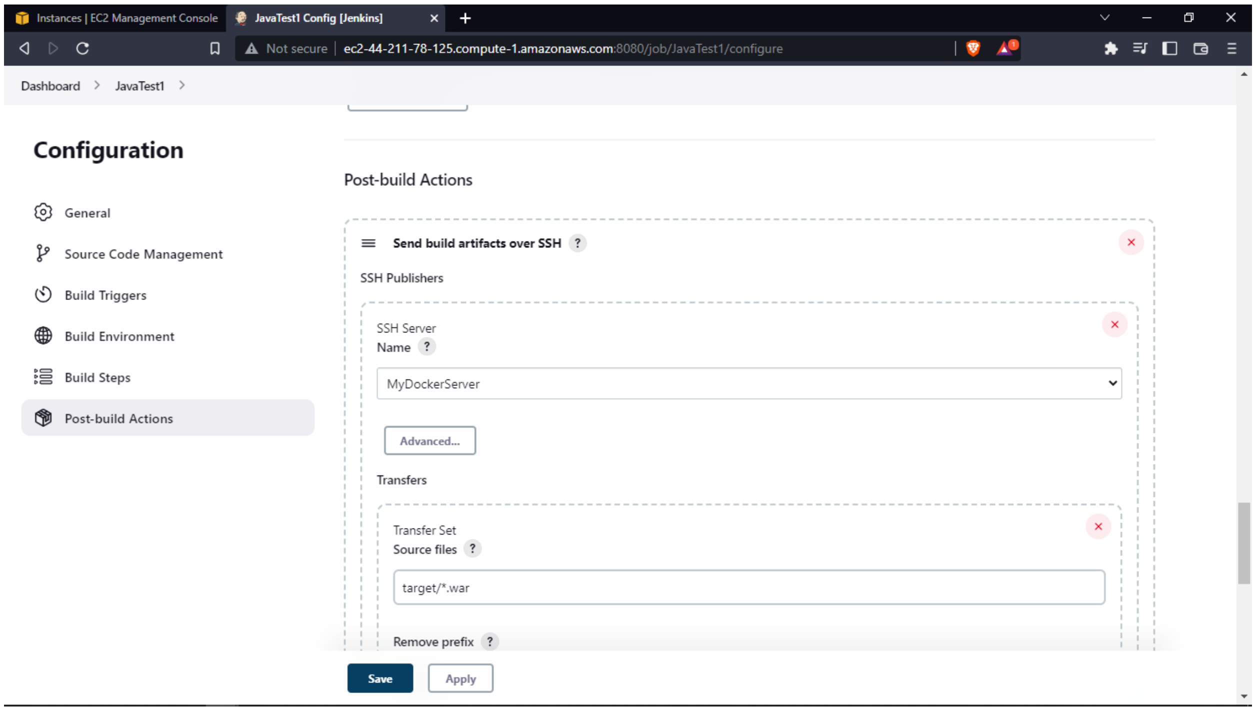
Source Files: target/*.war

Remove Prefix : target

Remote Directory: //home//ubuntu//project//javaproject

Exec Command::

```
docker stop mytomcatserver
docker rm mytomcatserver
docker rmi customtomcatimage
cd /home/ubuntu/project/javaproject
docker build -t customtomcatimage .
docker run -d --name mytomcatserver -p 8050:8080 customtomcatimage
```



Post-build Actions

Send build artifacts over SSH

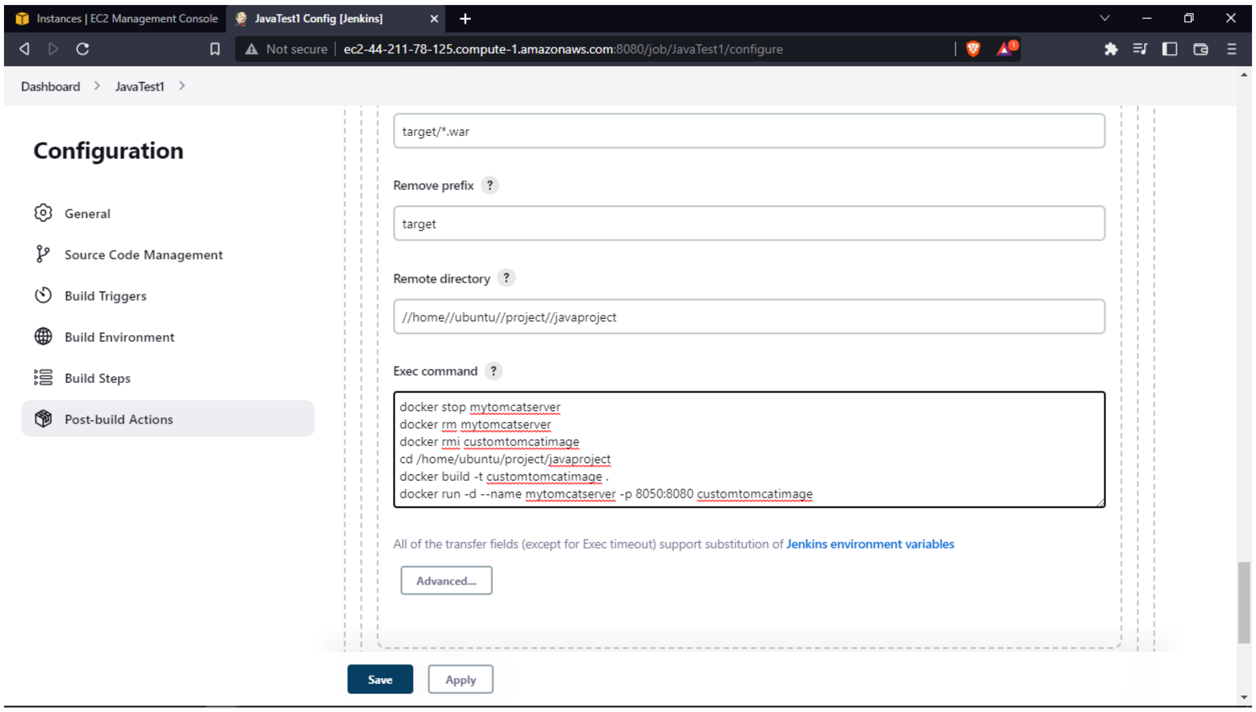
SSH Publishers

SSH Server
Name: MyDockerServer

Transfers

Transfer Set
Source files: target/*.war

Remove prefix



target/*.war

Remove prefix: target

Remote directory: //home//ubuntu//project//javaproject

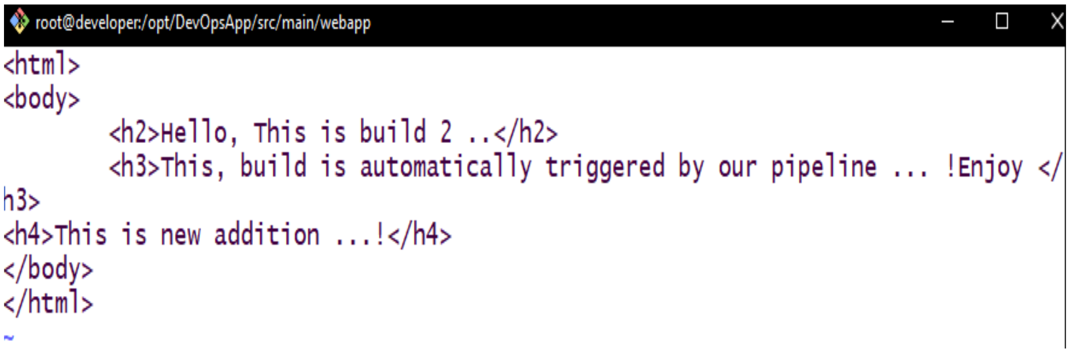
Exec command:

```
docker stop mytomcatserver
docker rm mytomcatserver
docker rmi customtomcatimage
cd /home/ubuntu/project/javaproject
docker build -t customtomcatimage .
docker run -d --name mytomcatserver -p 8050:8080 customtomcatimage
```

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

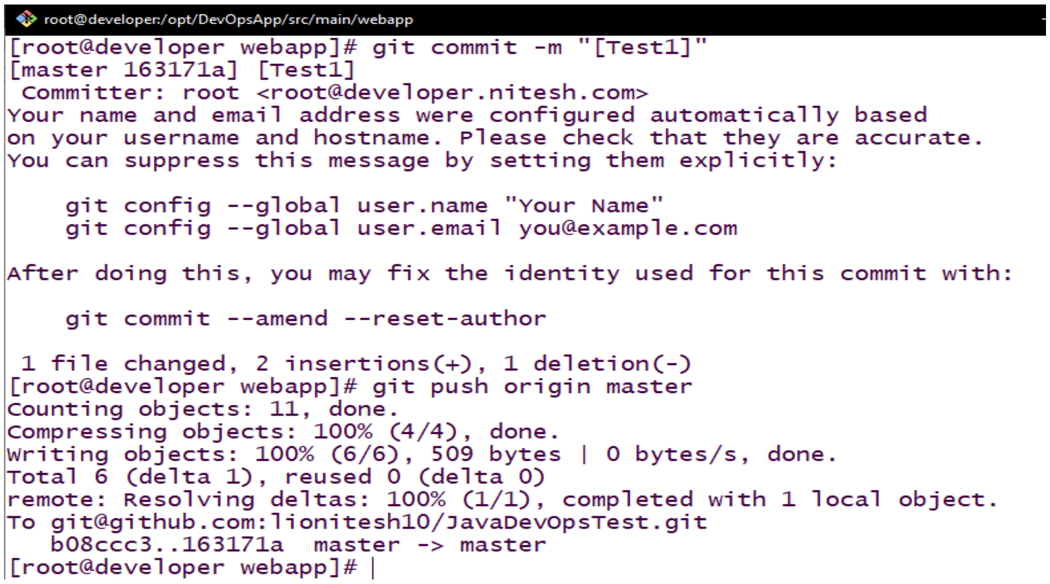
Click on Save.

Now lets make change in code.

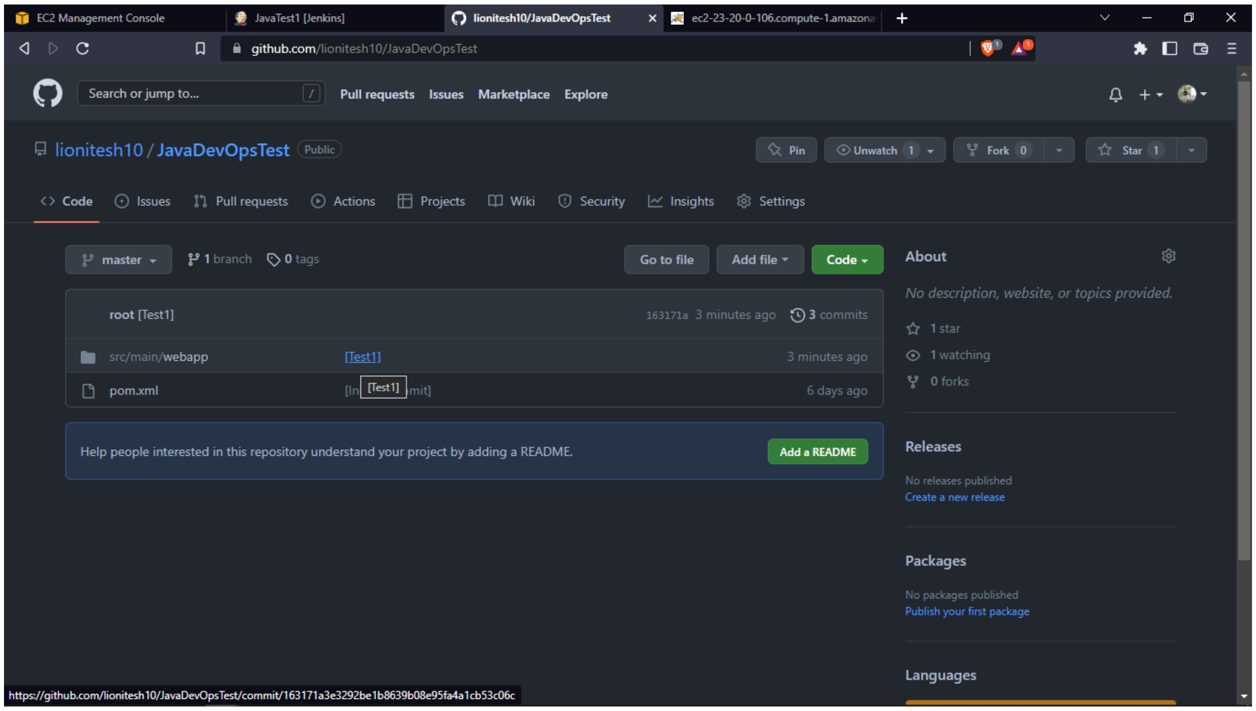


```
<html>
<body>
        <h2>Hello, This is build 2 ..</h2>
        <h3>This, build is automatically triggered by our pipeline ... !Enjoy </
h3>
<h4>This is new addition ...!</h4>
</body>
</html>
~
```

Commit and Push the changes:



```
[root@developer webapp]# git commit -m "[Test1]"
[master 163171a] [Test1]
 Committer: root <root@developer.nitesh.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

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

    git commit --amend --reset-author

 1 file changed, 2 insertions(+), 1 deletion(-)
[root@developer webapp]# git push origin master
Counting objects: 11, done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 509 bytes | 0 bytes/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To git@github.com:lionitesh10/JavaDevOpsTest.git
   b08ccc3..163171a  master -> master
[root@developer webapp]# |
```



root [Test1] 163171a 3 minutes ago 3 commits

src/main/webapp [Test1] 3 minutes ago

pom.xml [In[Test1]mit] 6 days ago

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

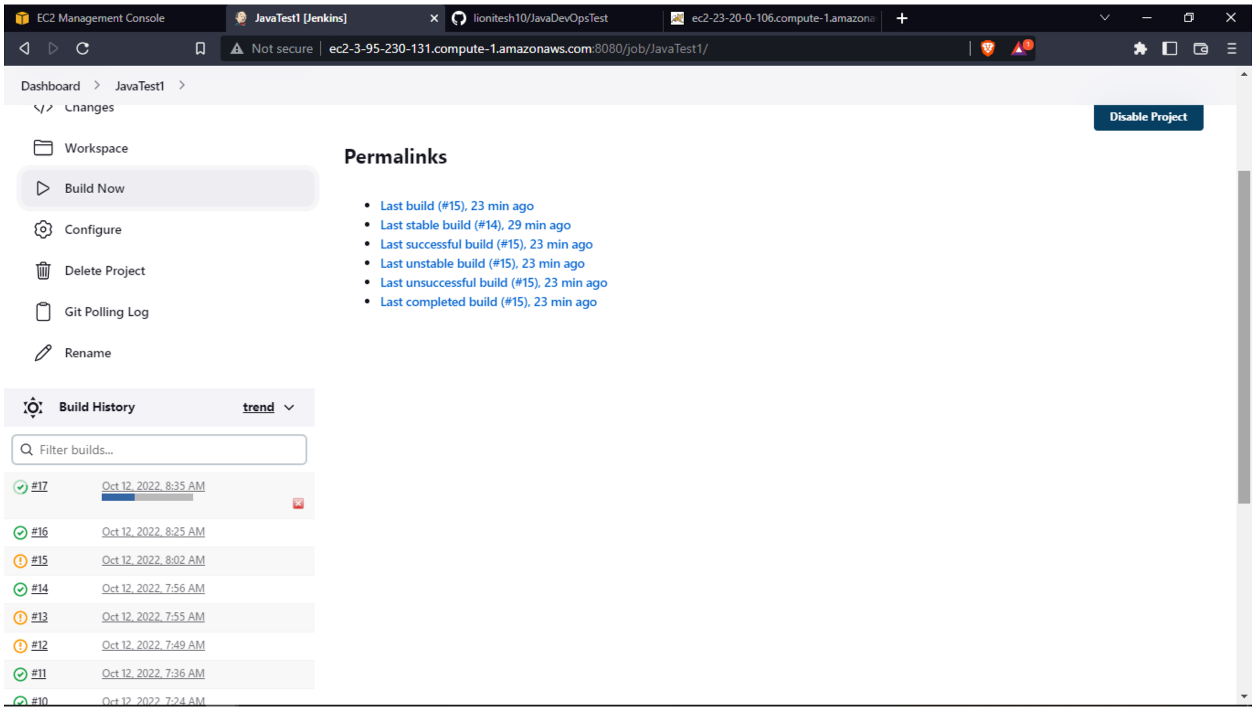
No description, website, or topics provided.

No releases published
Create a new release

No packages published
Publish your first package

https://github.com/lionitesh10/JavaDevOpsTest/commit/163171a3e3292be1b8639b08e95fa4a1cb53c06c

You can see your build progress starting after a minute.



Permalinks

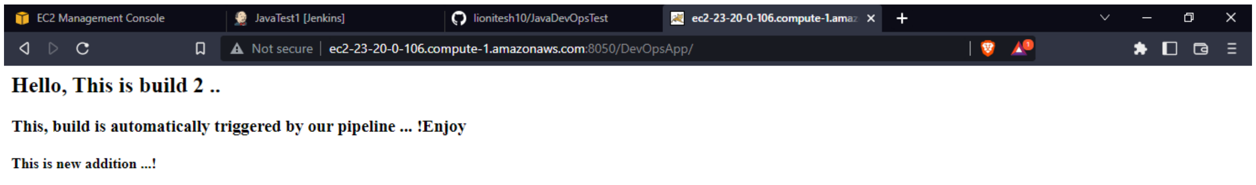
- Last build (#15), 23 min ago
- Last stable build (#14), 29 min ago
- Last successful build (#15), 23 min ago
- Last unstable build (#15), 23 min ago
- Last unsuccessful build (#15), 23 min ago
- Last completed build (#15), 23 min ago

#17	Oct 12, 2022, 8:35 AM
#16	Oct 12, 2022, 8:25 AM
#15	Oct 12, 2022, 8:02 AM
#14	Oct 12, 2022, 7:56 AM
#13	Oct 12, 2022, 7:55 AM
#12	Oct 12, 2022, 7:49 AM
#11	Oct 12, 2022, 7:36 AM
#10	Oct 12, 2022, 7:24 AM

Open the TCP PORT 8050 on Security Groups from anywhere 0.0.0.0

Then Navigate to DockerMachine IPv4 url:8050 /DevOpsApp/

You can see your result.


EC2 Management Console
JavaTest1 [Jenkins]
lionitesh10/JavaDevOpsTest
ec2-23-20-0-106.compute-1.amaz
Not secure
ec2-23-20-0-106.compute-1.amazonaws.com:8050/DevOpsApp/

Hello, This is build 2 ..

This, build is automatically triggered by our pipeline ... !Enjoy

This is new addition ...!

After the build succeed you can see the war file in dockermachine /home/Ubuntu/project/javaproject/

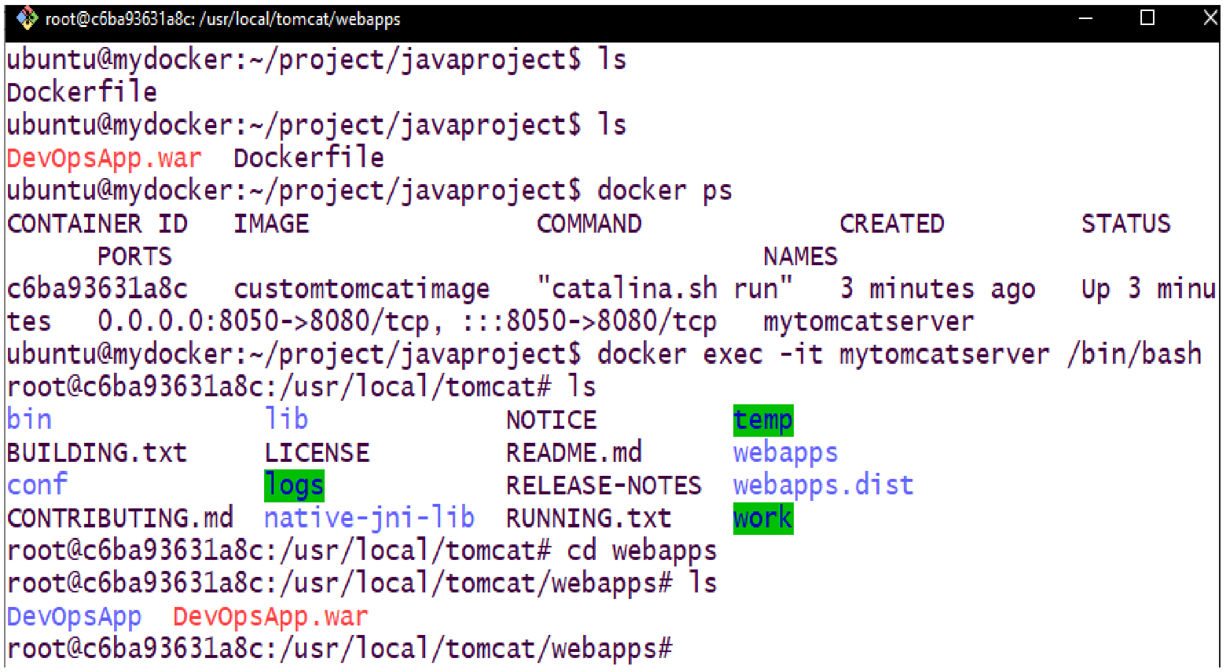
Before there was only Dockerfile and after build we have DevOpsApp.war.

We can check docker containers which are running we see out mytomcatserver

We can navigate into container using

Docker exec –it mytomcatserver /bin/bash

We could navigate into webapps and we see our .war file DevOpsApp.war.



```
root@c6ba93631a8c: /usr/local/tomcat/webapps
ubuntu@mydocker:~/project/javaproject$ ls
Dockerfile
ubuntu@mydocker:~/project/javaproject$ ls
DevOpsApp.war  Dockerfile
ubuntu@mydocker:~/project/javaproject$ docker ps
CONTAINER ID   IMAGE                COMMAND             CREATED         STATUS
      PORTS                                     NAMES
c6ba93631a8c   customtomcatimage   "catalina.sh run"   3 minutes ago   Up 3 minu
tes   0.0.0.0:8050->8080/tcp, :::8050->8080/tcp   mytomcatserver
ubuntu@mydocker:~/project/javaproject$ docker exec -it mytomcatserver /bin/bash
root@c6ba93631a8c:/usr/local/tomcat# ls
bin              lib             NOTICE         temp
BUILDING.txt     LICENSE         README.md      webapps
conf             logs            RELEASE-NOTES  webapps.dist
CONTRIBUTING.md  native-jni-lib  RUNNING.txt    work
root@c6ba93631a8c:/usr/local/tomcat# cd webapps
root@c6ba93631a8c:/usr/local/tomcat/webapps# ls
DevOpsApp  DevOpsApp.war
root@c6ba93631a8c:/usr/local/tomcat/webapps#
```