



Installing Jenkins on Ec2 Ubuntu

Introduction

This Documentation provides step by step guide for installing Jenkins on an EC2 instance running Ubuntu allowing you to leverage its capabilities to streamline your development workflow.

Prerequisites for Installing Jenkins on AWS EC2

Before proceeding with the installation of Jenkins on an AWS EC2 instance, ensure you have the following prerequisites in place

- ✓ AWS Account
- ✓ EC2 Instance
- ✓ Java Runtime Environment (JRE)
- ✓ SSH Key Pair
- ✓ Basic of AWS Services

What is Jenkins ?

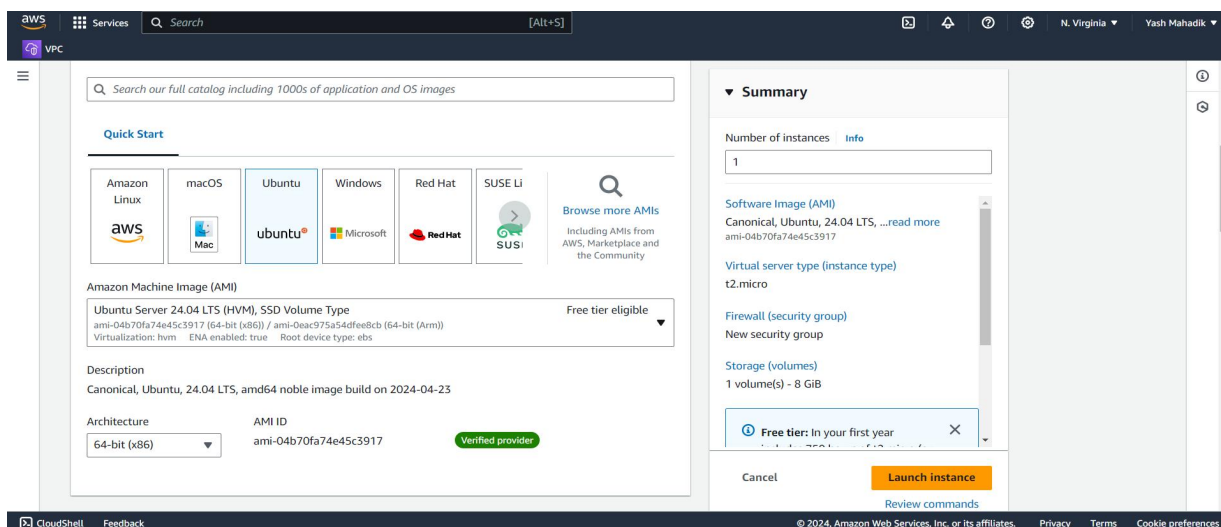
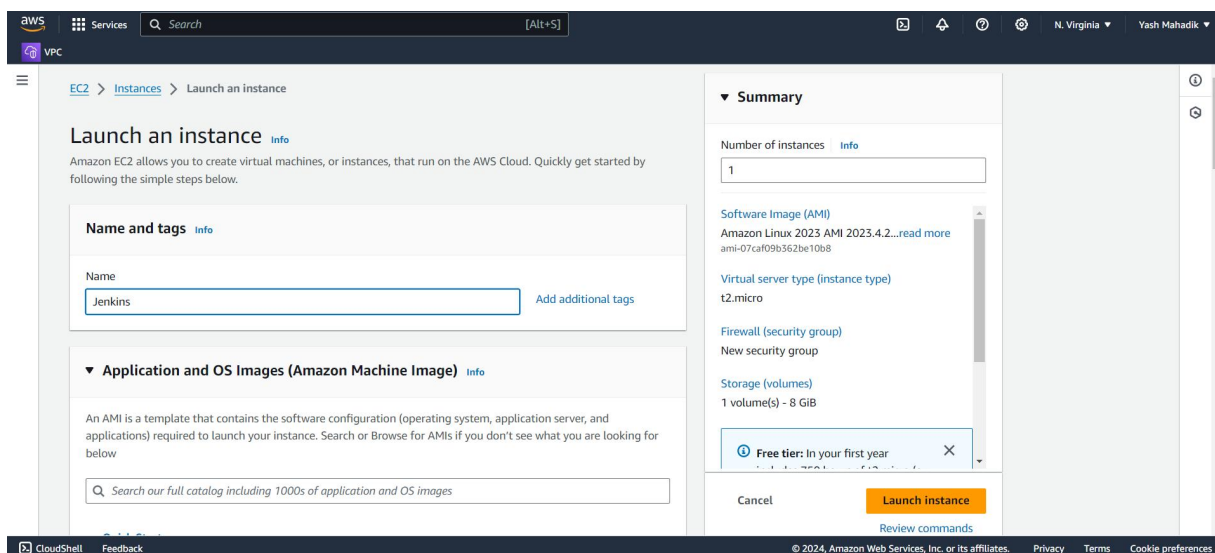
Jenkins is an open-source automation server that helps to automate various stages of software development. It is primarily used for Continuous Integration and Continuous Delivery/Deployment (CI/CD) pipelines, which allow developers to quickly build, test, and deploy their code to production.

What is AWS Cloud ?

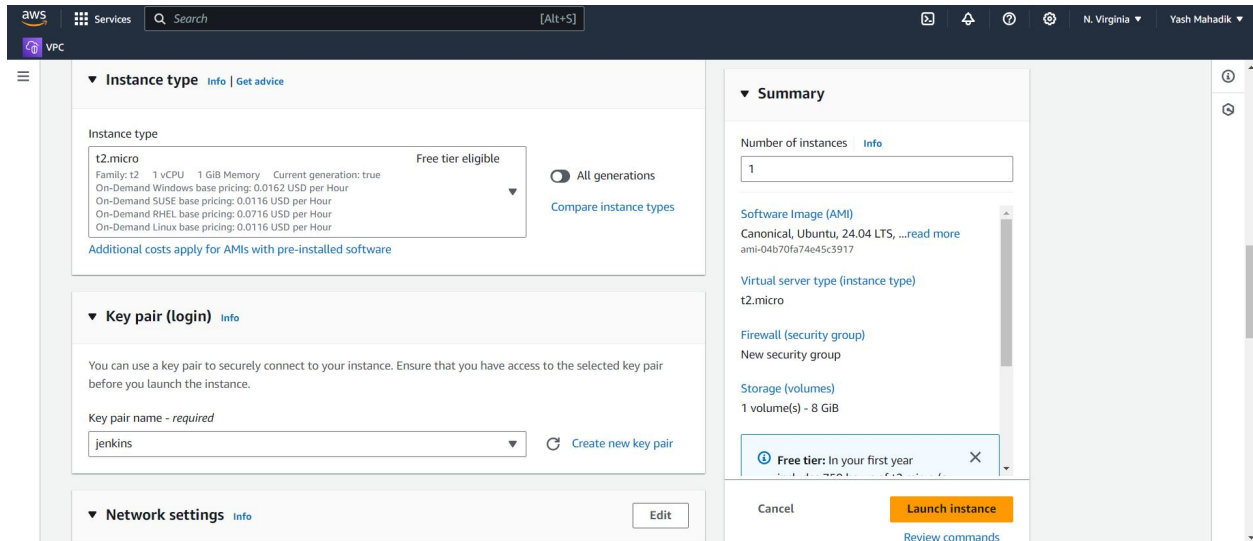
AWS (Amazon Web Services) is a cloud computing platform that provides a range of services for developers and businesses to build and deploy applications on the cloud. AWS has a wide range of services, including computing, storage, database, networking, and machine learning.

A step-by-step guide to installing Jenkins on an AWS EC2 Ubuntu 22.04 instance:

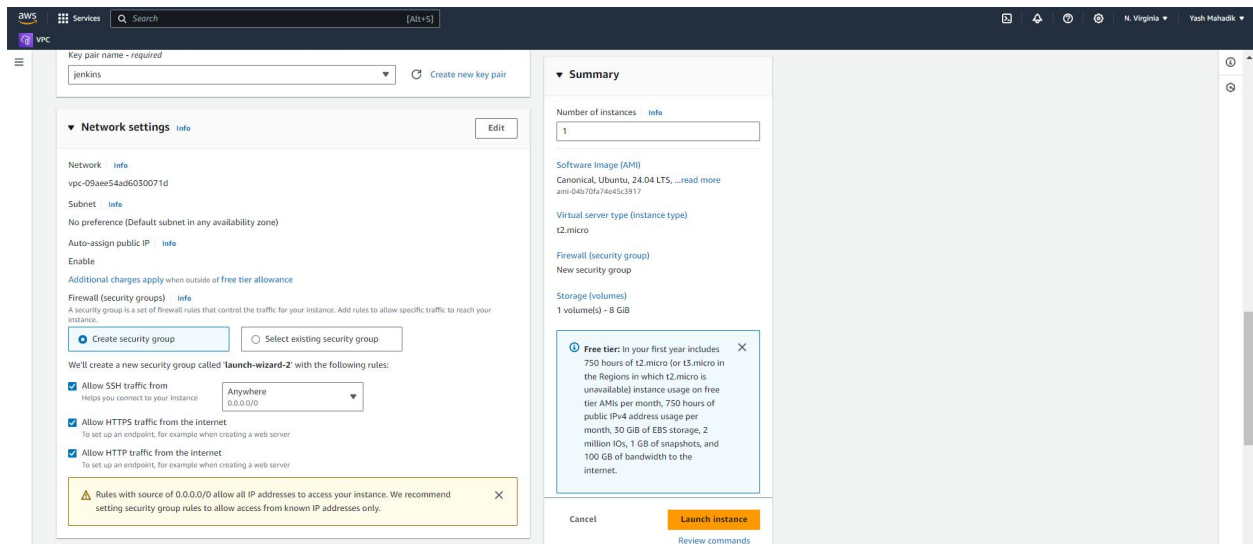
1) To First Login into your AWS account and to Launch Ec2 instance by giving the name of the instances as Jenkins-demo.



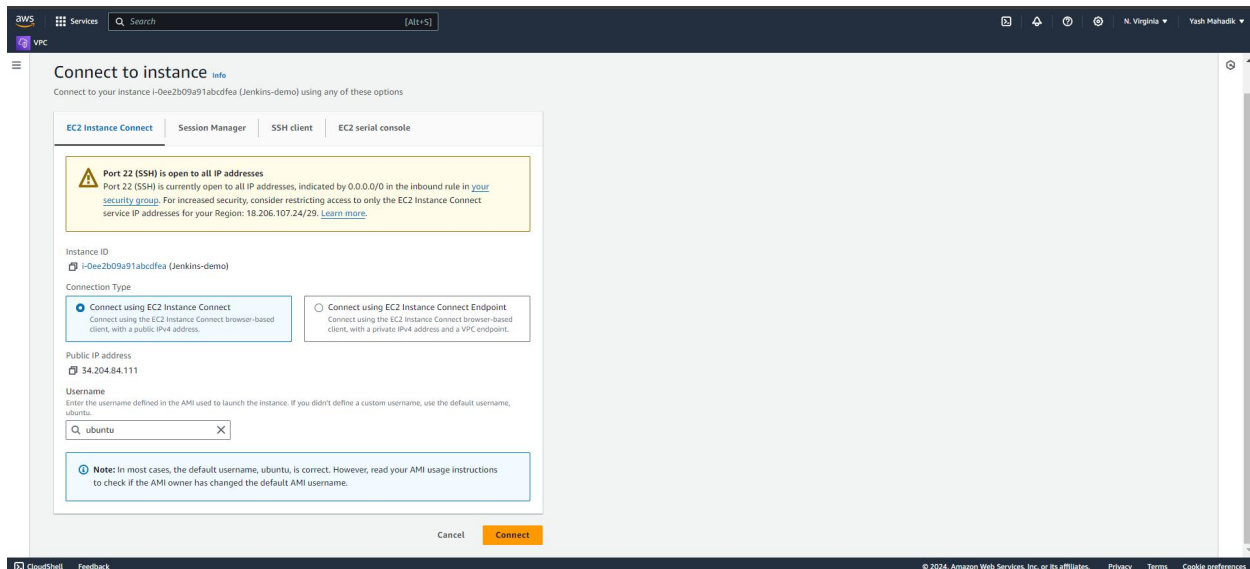
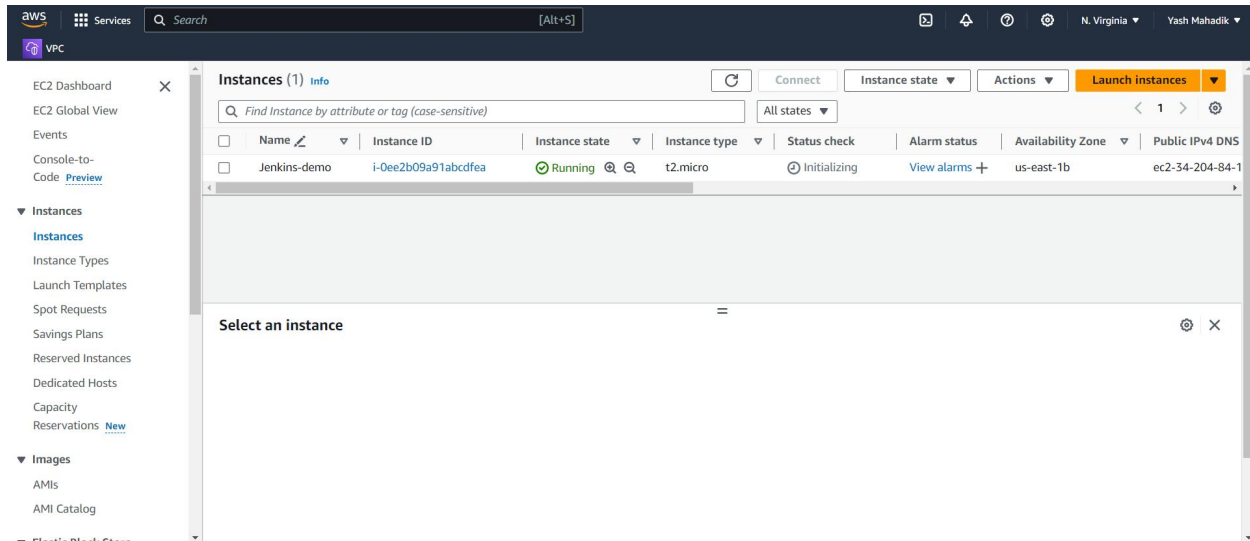
2) Use instance type as **t2.micro** which is **free tier** and create a **key pair**. You can select the appropriate instance type based on the requirements of your project.



3) In network section just check box ☒ on Allow HTTPS traffic from the internet and Click on Launch instance.

















4) Once the instance is created click on Instance and click on connect to open the terminal.



```
sudo apt-get update
sudo apt-get install openjdk-8-jdk
```

6) Search pkg.jenkins.io in Url and select Debian-stable.

Name	Last modified	Size	Description
 debian-rc/	2016-04-07 04:19	-	
 debian-stable-rc/	2016-02-04 19:46	-	
 debian-stable/	2024-04-17 15:03	-	
 debian/	2024-05-07 14:44	-	
 openuse-rc/	2016-04-07 04:21	-	
 openuse-stable-rc/	2016-02-04 19:48	-	
 openuse-stable/	2024-04-17 15:03	-	
 openuse/	2024-05-07 14:44	-	
 redhat-rc/	2016-04-07 04:20	-	
 redhat-stable-rc/	2016-02-04 19:47	-	
 redhat-stable/	2024-04-17 15:03	-	
 redhat/	2024-05-07 14:44	-	
 war/	2020-04-16 16:01	-	
 windows/	2020-04-16 16:02	-	

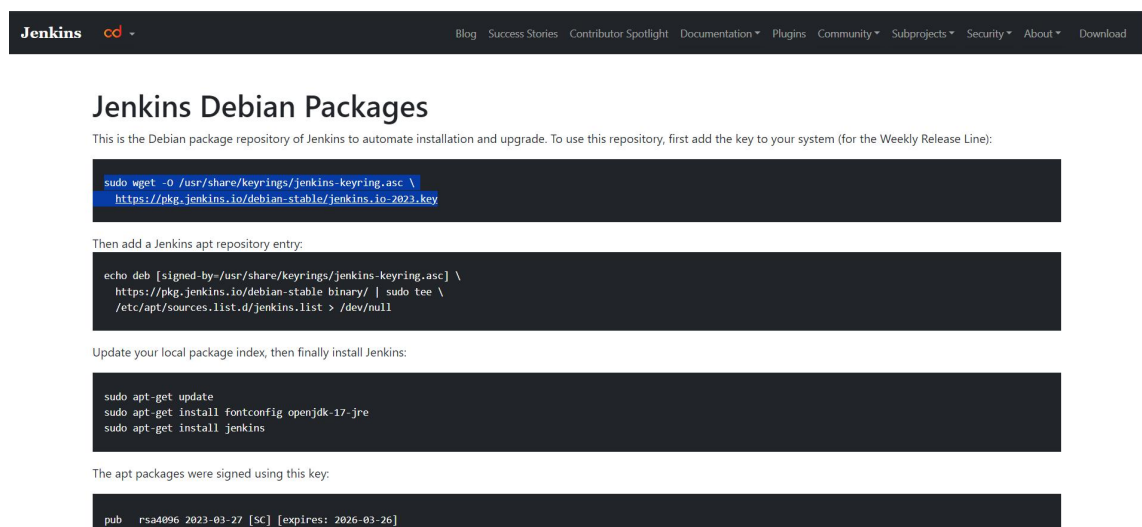
7) Copy each command and paste it on the terminal for installing jenkins.

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

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

Update your local package index, then finally install Jenkins:

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



The screenshot shows the Jenkins website's 'Debian Packages' page. The header includes the Jenkins logo and navigation links like 'Blog', 'Success Stories', 'Contributor Spotlight', 'Documentation', 'Plugins', 'Community', 'Subprojects', 'Security', 'About', and 'Download'. The main heading is 'Jenkins Debian Packages'. Below it, a text block states: 'This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system (for the Weekly Release Line):'. This is followed by a terminal code block containing two commands: `sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \` and `https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key`. The next section says 'Then add a Jenkins apt repository entry:' followed by another terminal block with three lines: `echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \`, `https://pkg.jenkins.io/debian-stable binary/ | sudo tee \`, and `/etc/apt/sources.list.d/jenkins.list > /dev/null`. The following section says 'Update your local package index, then finally install Jenkins:' followed by a terminal block with three lines: `sudo apt-get update`, `sudo apt-get install fontconfig openjdk-17-jre`, and `sudo apt-get install jenkins`. The final section says 'The apt packages were signed using this key:' followed by a terminal block showing a GPG key fingerprint: `pub rsa4096 2023-03-27 [SC] [expires: 2026-03-26]`.

8) Jenkins gets installed on the terminal.

```

aws
Services Search [Alt+S]
VPC
ubuntu@ip-172-31-41-241:~$ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2024-05-09 06:49:20-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.30.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)[146.75.30.133]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

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

2024-05-09 06:49:20 (36.2 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

ubuntu@ip-172-31-41-241:~$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
ubuntu@ip-172-31-41-241:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [26.7 kB]
Fetched 29.6 kB in 1s (41.8 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-41-241:~$ sudo apt-get install fontconfig openjdk-17-jre
Reading package lists... Done
Building dependency tree... Done

```

9) Once Jenkins is installed, start & enable the Jenkins service using the following command:

```

sudo systemctl start jenkins

sudo systemctl status jenkins

sudo systemctl enable jenkins

```

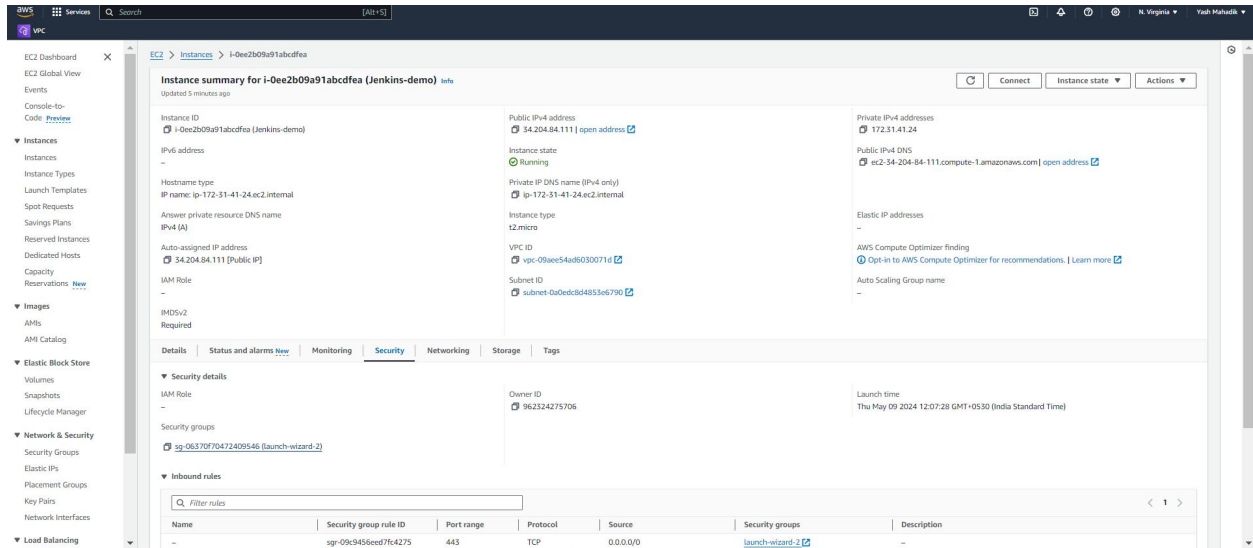
```

aws
Services Search [Alt+S]
VPC
ubuntu@ip-172-31-41-241:~$ java -version
openjdk version "17.0.11" 2024-04-16
OpenJDK Runtime Environment (build 17.0.11+9-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 17.0.11+9-Ubuntu-1, mixed mode, sharing)
ubuntu@ip-172-31-41-241:~$ sudo systemctl start jenkins
ubuntu@ip-172-31-41-241:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
ubuntu@ip-172-31-41-241:~$ sudo systemctl status jenkins
● Jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-05-09 06:52:35 UTC; 2min 46s ago
     Main PID: 4189 (java)
       Tasks: 39 (limit: 1130)
      Memory: 316.4M (peak: 352.2M)
         CPU: 46.907s
      CGroup: /system.slice/jenkins.service
              └─4189 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

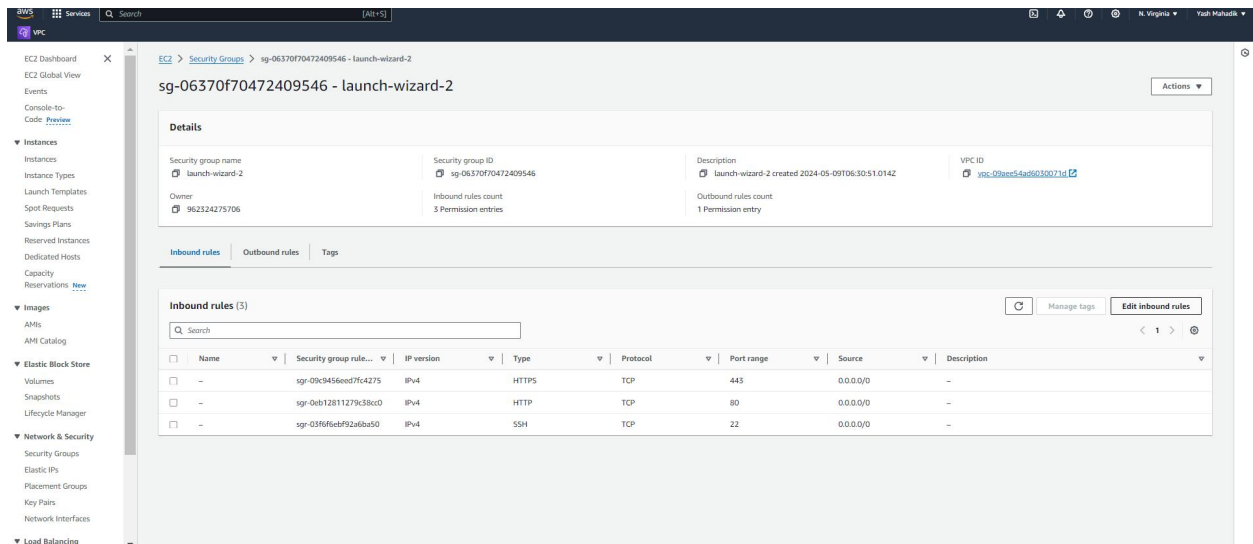
May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: cbc7b39f20b54efc9f0b46ae4d8bd5e0
May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: *****
May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: *****
May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: *****
May 09 06:52:35 ip-172-31-41-24 jenkins[4189]: 2024-05-09 06:52:35.899+0000 [id=31] INFO Jenkins.InitReactorRunner$1.onAttained: Completed initialization
May 09 06:52:35 ip-172-31-41-24 jenkins[4189]: 2024-05-09 06:52:35.947+0000 [id=24] INFO hudson.lifecycle.Lifecycle.onReady: Jenkins is fully up and running
May 09 06:52:35 ip-172-31-41-24 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
May 09 06:52:36 ip-172-31-41-24 jenkins[4189]: 2024-05-09 06:52:36.074+0000 [id=47] INFO hudson.DownloadService$Downloadable$load: Obtained the updated data file for hudson
May 09 06:52:36 ip-172-31-41-24 jenkins[4189]: 2024-05-09 06:52:36.075+0000 [id=47] INFO hudson.util.Retrier$start: Performed the action check updates server successful
y at the attempt #1
ubuntu@ip-172-31-41-241:~$

```


10) As we know Jenkins runs on port 8080, so we need to add a security group.



11) In the inbound section select edit inbound rules.



12) Add a new section with port range 8080 and click on save rules.

Edit inbound rules

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

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	Actions
sg-09c945eed7fc4275	HTTPS	TCP	443	Custom	0.0.0.0/0	Delete
sg-0eb12811279c38c0	HTTP	TCP	80	Custom	0.0.0.0/0	Delete
sg-03f6f9eb92a6ba50	SSH	TCP	22	Custom	0.0.0.0/0	Delete
-	Custom TCP	TCP	8080	Anywhere-IPv4	0.0.0.0/0	Delete

[Add rule](#)

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

sg-06370f70472409546 - launch-wizard-2

Details

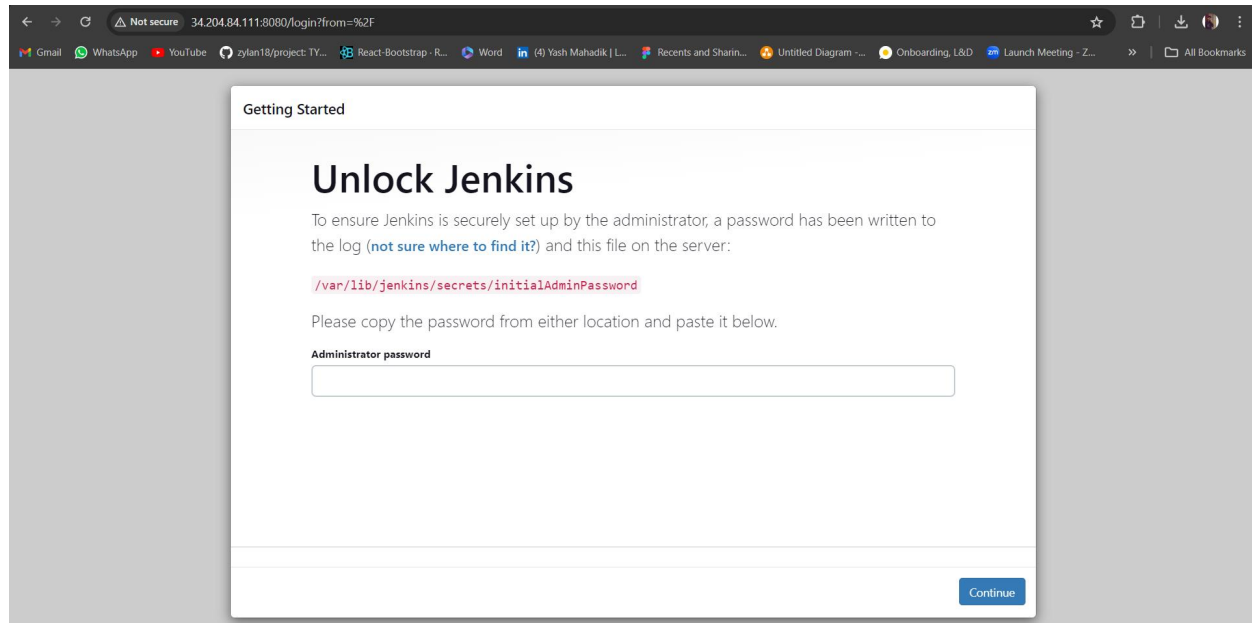
Security group name launch-wizard-2	Security group ID sg-06370f70472409546	Description launch-wizard-2 created 2024-05-09T06:30:51.014Z	VPC ID vpc-0baecf4ad6030071a
Owner 962324275706	Inbound rules count 4 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules (4)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sg-09c945eed7fc4275	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
-	sg-0eb12811279c38c0	IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	sg-05180f15ede78320	IPv4	Custom TCP	TCP	8080	0.0.0.0/0	-
-	sg-03f6f9eb92a6ba50	IPv4	SSH	TCP	22	0.0.0.0/0	-

[Manage tags](#) [Edit inbound rules](#)

13) We can see port 8080 is running and we can get start with Jenkins. Finally, you can configure Jenkins by accessing the Jenkins web interface using the public IP address of your EC2 instance and the default port 8080 (e.g., `http://<EC2_Public_IP>:8080`).

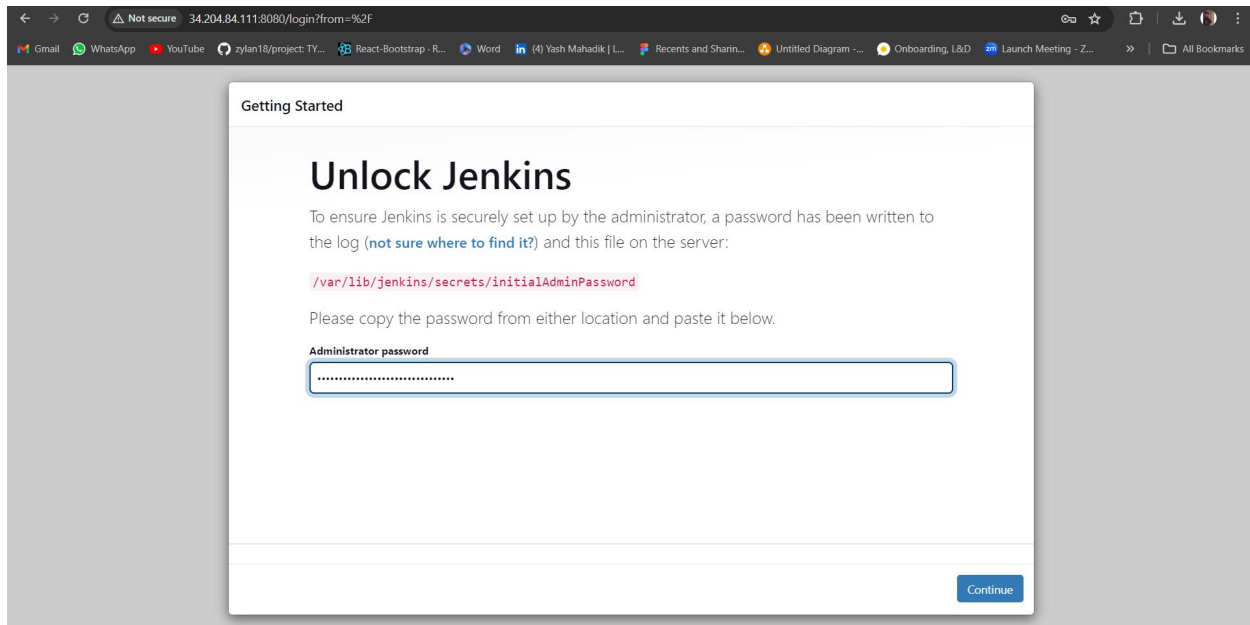


14) Copy the text and run `sudo cat <text you copied>` on terminal, a secret key will be generated and copy that key and paste in into you Jenkins as Administrator password

```
aws
Services
[Alt+S]
VPC
ubuntu@ip-172-31-41-24:~$ java -version
openjdk version "17.0.11" 2024-04-16
OpenJDK Runtime Environment (build 17.0.11+9-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 17.0.11+9-Ubuntu-1, mixed mode, sharing)
ubuntu@ip-172-31-41-24:~$ sudo systemctl start jenkins
ubuntu@ip-172-31-41-24:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
ubuntu@ip-172-31-41-24:~$ sudo systemctl status jenkins
jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-05-09 06:52:35 UTC; 2min 46s ago
     Main PID: 4189 (java)
       Tasks: 39 (limit: 1130)
      Memory: 216.4M (peak: 352.2M)
         CPU: 46.907s
    CGroup: /system.slice/jenkins.service
            └─4189 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: cbc7b39f20b54efc9f0b46ae4d8bd5e0
May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: *****
May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: *****
May 09 06:51:59 ip-172-31-41-24 jenkins[4189]: *****
May 09 06:52:35 ip-172-31-41-24 jenkins[4189]: 2024-05-09 06:52:35.899+0000 [id=31] INFO jenkins.InitReactorRunner$1:onAttained: Completed initialization
May 09 06:52:35 ip-172-31-41-24 jenkins[4189]: 2024-05-09 06:52:35.947+0000 [id=24] INFO hudson.lifecycle.Lifecycle$onReady: Jenkins is fully up and running
May 09 06:52:35 ip-172-31-41-24 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
May 09 06:52:36 ip-172-31-41-24 jenkins[4189]: 2024-05-09 06:52:36.074+0000 [id=47] INFO h.m.DownloadService$Downloadable$load: Obtained the updated data file for hudson.tasks.Maven.MavenIns
trinsics
May 09 06:52:36 ip-172-31-41-24 jenkins[4189]: 2024-05-09 06:52:36.075+0000 [id=47] INFO hudson.util.Retrier$start: Performed the action check updates server successfully at the attempt #1
ubuntu@ip-172-31-41-24:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
cbc7b39f20b54efc9f0b46ae4d8bd5e0
ubuntu@ip-172-31-41-24:~$
```

15) Add secret key to Administrator password



Getting Started

Unlock Jenkins

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

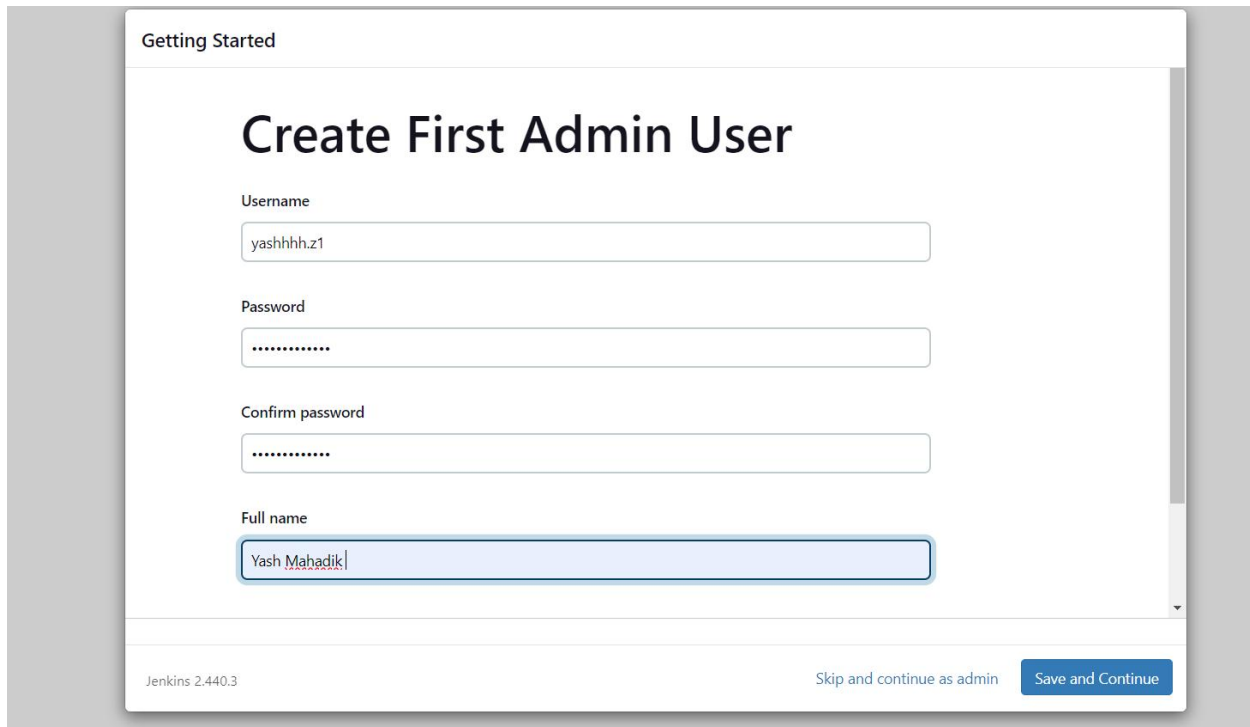
```
/var/lib/jenkins/secrets/initialAdminPassword
```

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

Administrator password

Continue

16) Create Admin user and click on save and continue.



Getting Started

Create First Admin User

Username

yashhhh.z1

Password

Confirm password

Full name

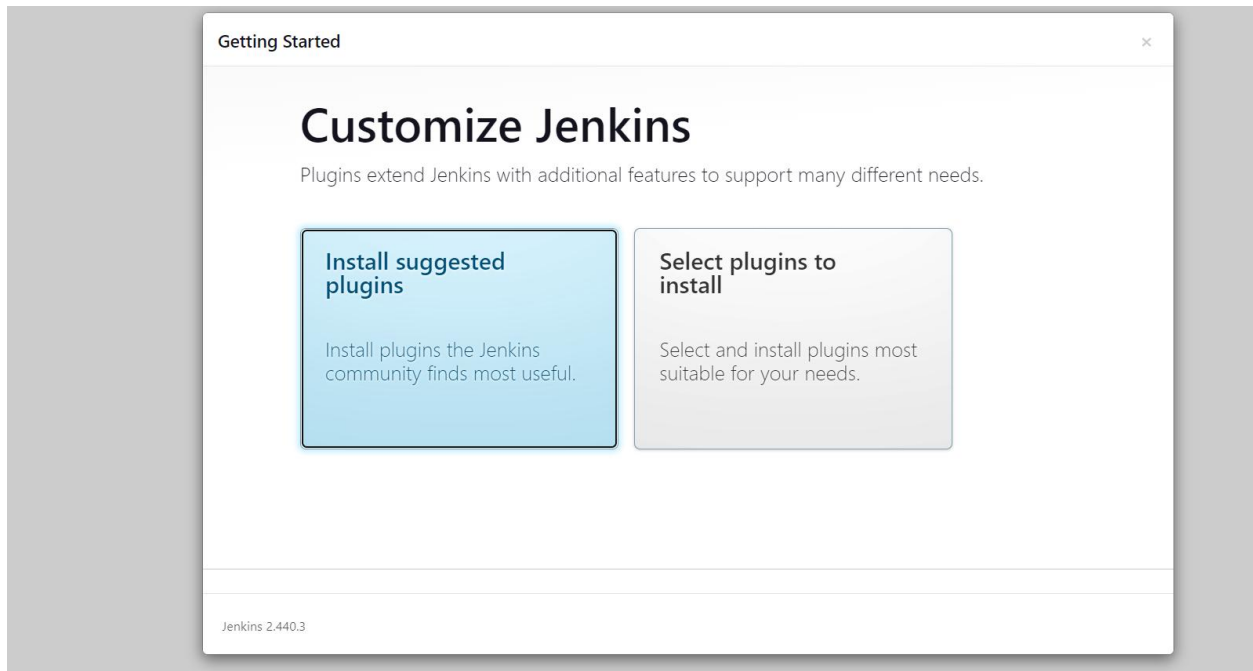
Yash Mahadik

Jenkins 2.440.3

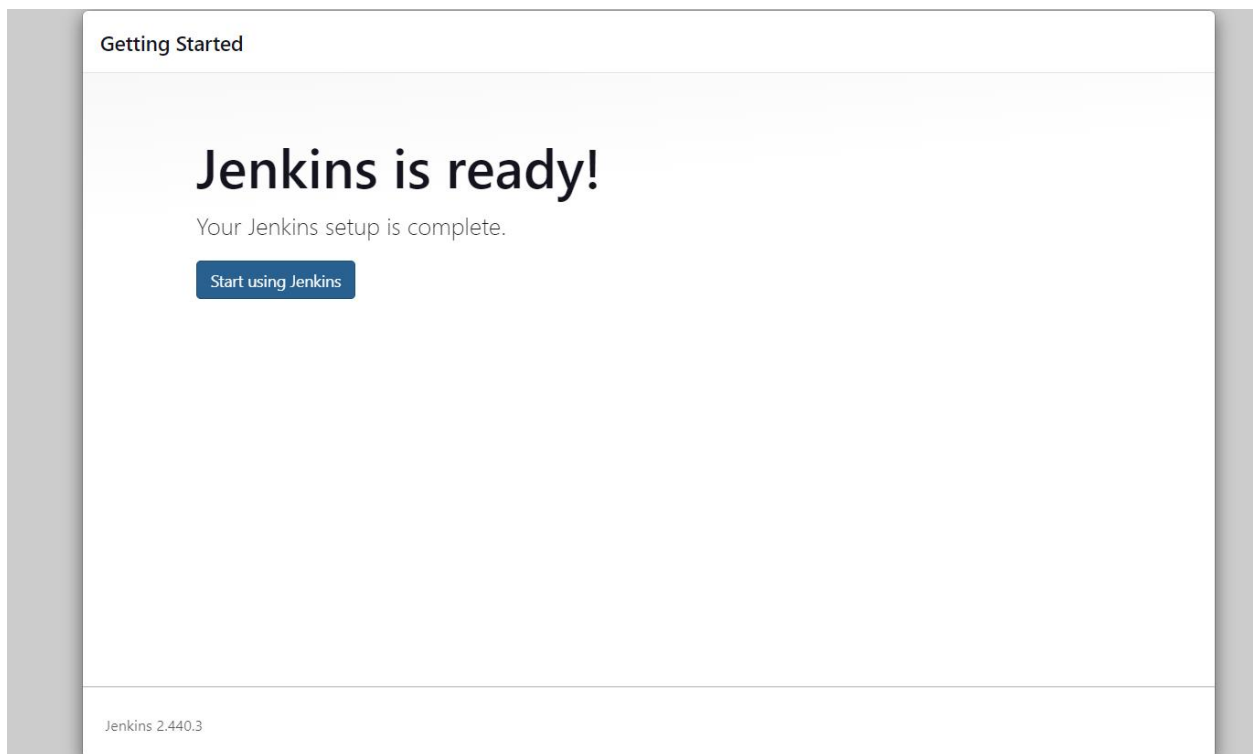
Skip and continue as admin

Save and Continue

17) Install necessary plugins.



18) Jenkins is Ready! Start using Jenkins



Conclusion: Installing Jenkins on an EC2 Ubuntu instance provides a robust foundation for implementing CI/CD pipelines, fostering collaboration among development teams, and accelerating software delivery. By following the documentation and incorporating best practices, users can leverage Jenkins to enhance the quality, reliability, and efficiency of their software development process.