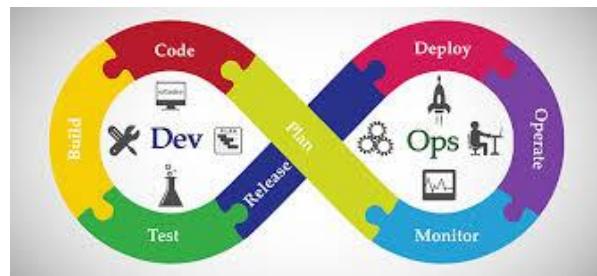


**A PROJECT REPORT ON**  
**DEPLOYING OF DOCKER IMAGE ON**  
**EC2 INSTANCE**  
**COURSE**  
**DEVOPS WITH AWS**

**Submitted by**  
**SIRI CHANDANA.BANDA**  
**Under The Guidance Of**  
**VAMSI BYRAMALA (TRAINER)**  
**ASHOK REKHA (MENTOR)**



Greatcoder Training institute in Madhapur Hyderabad

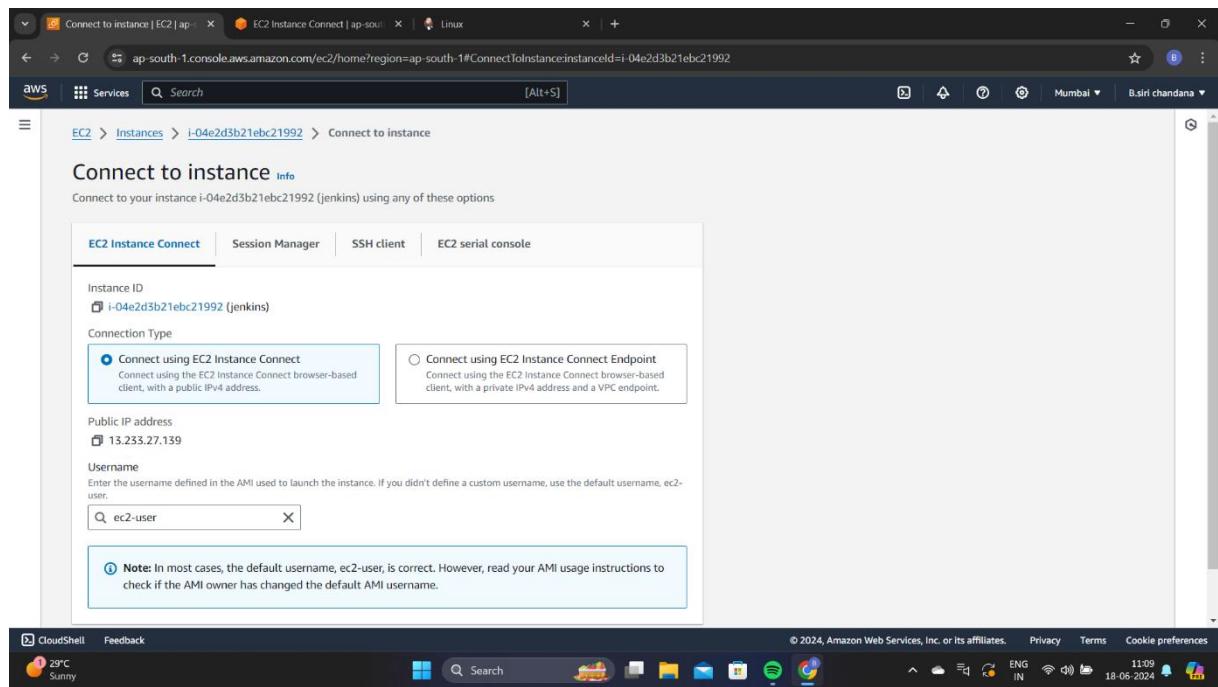
#201,2nd floor,Above Building,Beside Indian

Bank Madhapur, Hyderabad-500081

Land Mark: Beside Karachi Bakery, lane, Hyderabad Telangana.

## To Launch an Instance:

- ❖ Open the Amazon Ec2 console at <https://console.aws.amazon.com/ec2>
- ❖ From the Ec2 console dashboard
- ❖ We should Launch Instance
- ❖ Under the name and tags we should give Instance name
- ❖ Under Application and Os image we should select Amazon Linux (AWS) OS
- ❖ Under Amazon Machine Image (AMI) we should select version Amazon Linux 2(AMI)
- ❖ Under Instance type we should select t2.micro (free tier eligible)
- ❖ Under Key Pair (login) we should select key pair
- ❖ Click On Launch Instance
- ❖ Instance will be launched



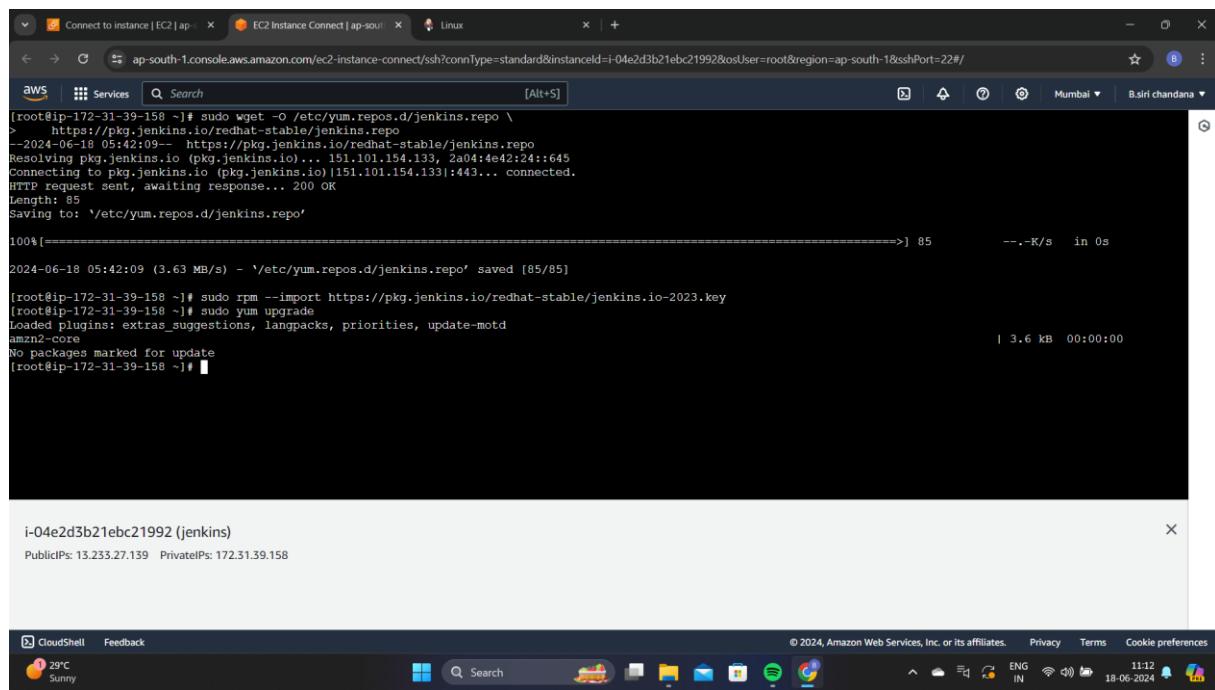
- ❖ Connect Instance

## Jenkins Installation and Configure:

- Jenkins Installation (search in google)
- Select install jenkins
- Select Linux
- Click on [Red Hat/Alma/Rocky](#)
- Under Long Term Support release

## Run These Commands:

- o Long term Support Release:
- sudo wget -O /etc/yum.repos.d/jenkins.repo \
- sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
- sudo yum upgrade
- #Add required dependencies for the jenkins package
- Yum list | grep "java" (select java-17)
- Yum install java-17-amazon-corretto.x86\_64
- Sudo yum install jenkins
- Systemctl start jenkins
- Systemctle enable jenkins



```
[root@ip-172-31-39-158 ~]# sudo wget -O /etc/yum.repos.d/jenkins.repo \
> https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-06-18 05:42:09-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.154.133, 2a04:4e42:24::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.154.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

100%[=====] 85 --.-R/s   in 0s

2024-06-18 05:42:09 (3.63 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[root@ip-172-31-39-158 ~]# sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@ip-172-31-39-158 ~]# sudo yum upgrade
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update
[root@ip-172-31-39-158 ~]#
```

i-04e2d3b21ebc21992 (jenkins)  
PublicIPs: 15.233.27.139 PrivateIPs: 172.31.39.158



CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences  
29°C Sunny ENG IN 11:12 18-06-2024

```
[root@ip-172-31-39-158 ~]# yum list | grep "java"
aether-javadoc.noarch      1.13.1-13.amzn2      amzn2-core
ant-antunit-javadoc.noarch 1.2-10.amzn2        amzn2-core
ant-contrib-javadoc.noarch  1.0-0.23.b3.amzn2    amzn2-core
ant-javadoc.noarch         1.9.16-1.amzn2.0.1   amzn2-core
ant-javamail.noarch        1.9.16-1.amzn2.0.1   amzn2-core
antlr-javadoc.noarch      2.7.7-30.amzn2.0.2  amzn2-core
aopalliance-javadoc.noarch 1.0-8.1.amzn2       amzn2-core
apache-commons-beanutils-javadoc.noarch 1.8.3-15.amzn2 amzn2-core
apache-commons-cli-javadoc.noarch 1.2-13.amzn2    amzn2-core
apache-commons-codec-javadoc.noarch 1.8-7.amzn2    amzn2-core
apache-commons-collections-javadoc.noarch apache-commons-collections-testframework-javadoc.noarch
apache-commons-compress-javadoc.noarch 1.5-4.amzn2.0.1 amzn2-core
apache-commons-configuration-javadoc.noarch apache-commons-daemon-javadoc.noarch 1.0.13-7.amzn2  amzn2-core
apache-commons-dbcp-javadoc.noarch 1.4-17.amzn2    amzn2-core
apache-commons-digester-javadoc.noarch 1.8.1-19.amzn2 amzn2-core
apache-commons-exec-javadoc.noarch 1.1-11.amzn2    amzn2-core
apache-commons-io-javadoc.noarch 1:2.4-12.amzn2.0.1 amzn2-core
apache-commons-jexl-javadoc.noarch 2.1.1-9.amzn2    amzn2-core
apache-commons-jxpath-javadoc.noarch 1.3-20.amzn2   amzn2-core
apache-commons-lang-javadoc.noarch 2.6-15.amzn2   amzn2-core
apache-commons-lang3-javadoc.noarch 3.1-9.amzn2    amzn2-core
apache-commons-logging-javadoc.noarch 1.1.2-7.amzn2 amzn2-core
apache-commons-net-javadoc.noarch 3.2-8.amzn2     amzn2-core
apache-commons-pool-javadoc.noarch 1.6-9.amzn2    amzn2-core
apache-commons-validator-javadoc.noarch 1.4.0-8.amzn2 amzn2-core
```

i-04e2d3b21ebc21992 (jenkins)  
Public IPs: 13.233.27.139 Private IPs: 172.31.39.158



```
[root@ip-172-31-39-158 ~]# yum list | grep "java"
java-1.7.0-openjdk-accessibility.x86_64 1:1.7.0.321-2.6.28.2.amzn2.0.2
java-1.7.0-openjdk-demo.x86_64          1:1.7.0.321-2.6.28.2.amzn2.0.2
java-1.7.0-openjdk-devel.x86_64          1:1.7.0.321-2.6.28.2.amzn2.0.2
java-1.7.0-openjdk-headless.x86_64       1:1.7.0.321-2.6.28.2.amzn2.0.2
java-1.7.0-openjdk-javadoc.noarch        1:1.7.0.321-2.6.28.2.amzn2.0.2
java-1.7.0-openjdk-src.x86_64           1:1.7.0.321-2.6.28.2.amzn2.0.2
java-1.8.0-openjdk.x86_64               1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-accessibility.x86_64 1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-accessibility-debug.x86_64 1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-debug.x86_64          1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-demo.x86_64          1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-demo-debug.x86_64    1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-devel.x86_64          1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-devel-debug.x86_64    1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-headless.x86_64       1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-headless-debug.x86_64 1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-javadoc.noarch        1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-javadoc-debug.noarch 1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-javadoc-zip.noarch   1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-javadoc-zip-debug.noarch 1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-src.x86_64           1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-1.8.0-openjdk-src-debug.x86_64     1:1.8.0.412.b08-1.amzn2.0.1 amzn2-core
java-11-amazon-corretto.x86_64          1:11.0.23+9-1.amzn2 amzn2-core
java-11-amazon-corretto-headless.x86_64 1:11.0.23+9-1.amzn2 amzn2-core
java-11-amazon-corretto-javadoc.x86_64  1:11.0.23+9-1.amzn2 amzn2-core
java-17-amazon-corretto.x86_64          1:17.0.11+9-1.amzn2.1 amzn2-core
java-17-amazon-corretto-devel.x86_64    1:17.0.11+9-1.amzn2.1 amzn2-core
java-17-amazon-corretto-headless.x86_64  1:17.0.11+9-1.amzn2.1 amzn2-core
```

i-04e2d3b21ebc21992 (jenkins)  
Public IPs: 13.233.27.139 Private IPs: 172.31.39.158



```
[root@ip-172-31-39-158 ~]# yum install java java-17-amazon-corretto.x86_64 -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package java-17-amazon-corretto.x86_64 1:17.0.11+9-1.amzn2.1 will be installed
--> Processing Dependency: java-17-amazon-corretto-headless(x86-64) = 1:17.0.11+9-1.amzn2.1 for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libX11 for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXi for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXinerama for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXt for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXrandr for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXtst for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: giflib for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Running transaction check
--> Package giflib.x86_64 0:4.1.6-9.amzn2.0.2 will be installed
--> Processing Dependency: libICE.so.6() (64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
--> Processing Dependency: libSM.so.6() (64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
-->> Package java-17-amazon-corretto-headless.x86_64 1:17.0.11+9-1.amzn2.1 will be installed
--> Processing Dependency: jpackage-utils for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: fontconfig for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-fonts for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-serif-fonts for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-mono-fonts for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: alsalib for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
-->> Package libX11.x86_64 0:1.6.7-3.amzn2.0.5 will be installed
```

i-04e2d3b21ebc21992 (jenkins)

PublicIPs: 13.233.27.139 PrivateIPs: 172.31.39.158



```
[root@ip-172-31-39-158 ~]# java -version
java version "17.0.11+9-1.amzn2.1"
Java(TM) SE Runtime Environment "17.0.11+9-1.amzn2.1"
Java HotSpot(TM) 64-Bit Server VM "17.0.11+9-1.amzn2.1"

[root@ip-172-31-39-158 ~]# yum install java java-17-amazon-corretto.x86_64 -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package java-17-amazon-corretto.x86_64 1:17.0.11+9-1.amzn2.1 will be installed
--> Processing Dependency: java-17-amazon-corretto-headless(x86-64) = 1:17.0.11+9-1.amzn2.1 for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libX11 for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXi for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXinerama for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXt for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXrandr for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: libXtst for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: giflib for package: 1:java-17-amazon-corretto-17.0.11+9-1.amzn2.1.x86_64
--> Running transaction check
--> Package giflib.x86_64 0:4.1.6-9.amzn2.0.2 will be installed
--> Processing Dependency: libICE.so.6() (64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
--> Processing Dependency: libSM.so.6() (64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
-->> Package java-17-amazon-corretto-headless.x86_64 1:17.0.11+9-1.amzn2.1 will be installed
--> Processing Dependency: jpackage-utils for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: fontconfig for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-fonts for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-serif-fonts for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-mono-fonts for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
--> Processing Dependency: alsalib for package: 1:java-17-amazon-corretto-headless-17.0.11+9-1.amzn2.1.x86_64
-->> Package libX11.x86_64 0:1.6.7-3.amzn2.0.5 will be installed
```

i-04e2d3b21ebc21992 (jenkins)

PublicIPs: 13.233.27.139 PrivateIPs: 172.31.39.158



```
[root@ip-172-31-39-158 ~]# yum install jenkins -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package jenkins.noarch 0:2.452.2-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
| Package           | Arch   | Version      | Repository | Size |
|-----|-----|-----|-----|-----|
| Installing:     |        |             |            |       |
| jenkins          | noarch | 2.452.2-1.1 | jenkins    | 89 M |
|-----|-----|-----|-----|-----|
| Transaction Summary |          |              |            |       |
|-----|-----|-----|-----|-----|
| Install 1 Package |          |              |            |       |
|-----|-----|-----|-----|-----|
| Total download size: 89 M |          |              |            |       |
| Installed size: 89 M |          |              |            |       |
| Downloading packages: |          |              |            |       |
| jenkins-2.452.2-1.1.noarch.rpm |          |              |            |       |
|-----|-----|-----|-----|-----|
| Running transaction check |          |              |            |       |
| Running transaction test |          |              |            |       |
| Transaction test succeeded |          |              |            |       |
|-----|-----|-----|-----|-----|
| Running transaction |          |              |            |       |
| . Installing : jenkins-2.452.2-1.1.noarch |          |              |            |       |
|-----|-----|-----|-----|-----|
1/1
```

i-04e2d3b21ebc21992 (jenkins)  
PublicIPs: 13.233.27.139 PrivateIPs: 172.31.39.158



```
Transaction Summary
Install 1 Package

Total download size: 89 M
Installed size: 89 M
Downloading packages:
jenkins-2.452.2-1.1.noarch.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.452.2-1.1.noarch
  Verifying : jenkins-2.452.2-1.1.noarch
1/1
1/1

Installed:
  jenkins.noarch 0:2.452.2-1.1

Complete!
[root@ip-172-31-39-158 ~]# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
[root@ip-172-31-39-158 ~]# systemctl start jenkins
[root@ip-172-31-39-158 ~]# systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-39-158 ~]#
```

i-04e2d3b21ebc21992 (jenkins)  
PublicIPs: 13.233.27.139 PrivateIPs: 172.31.39.158



- ❖ We should edit the inbound rules
- ❖ We should give port number (8080-8083) AnyWhere-Ipv4
- ❖ Save rules

The screenshot shows the 'Edit inbound rules' step of the AWS EC2 ModifyInboundSecurityGroup wizard. It lists two rules:

- SSH**: Type: SSH, Protocol: TCP, Port range: 22, Source: Custom, Description: (empty)
- Custom TCP**: Type: Custom TCP, Protocol: TCP, Port range: 8080 - 8083, Source: Anywh..., Description: (empty)

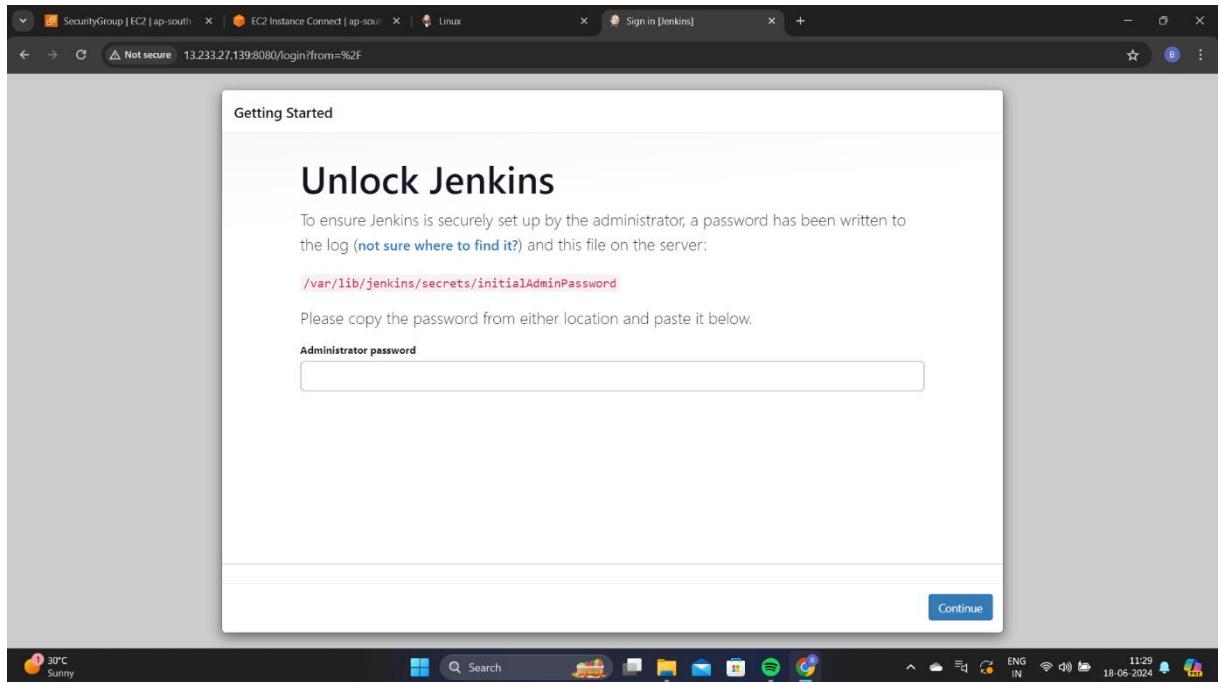
A yellow warning box at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." Buttons for 'Cancel', 'Preview changes', and 'Save rules' are visible.

- ❖ Copy the PublicIPs from instance
- ❖ Paste in google PublicIPs:8080(port number)

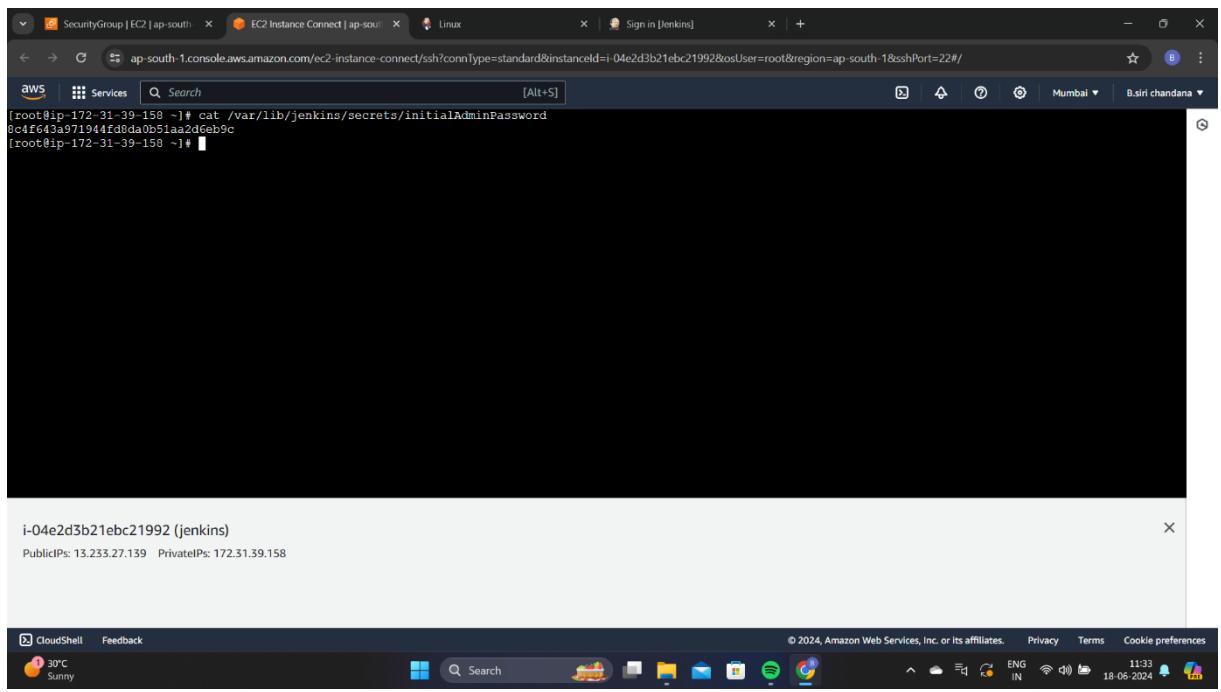
The screenshot shows an AWS CloudShell terminal window. The command `curl -s http://checkip.amazonaws.com` was run, and the output was:

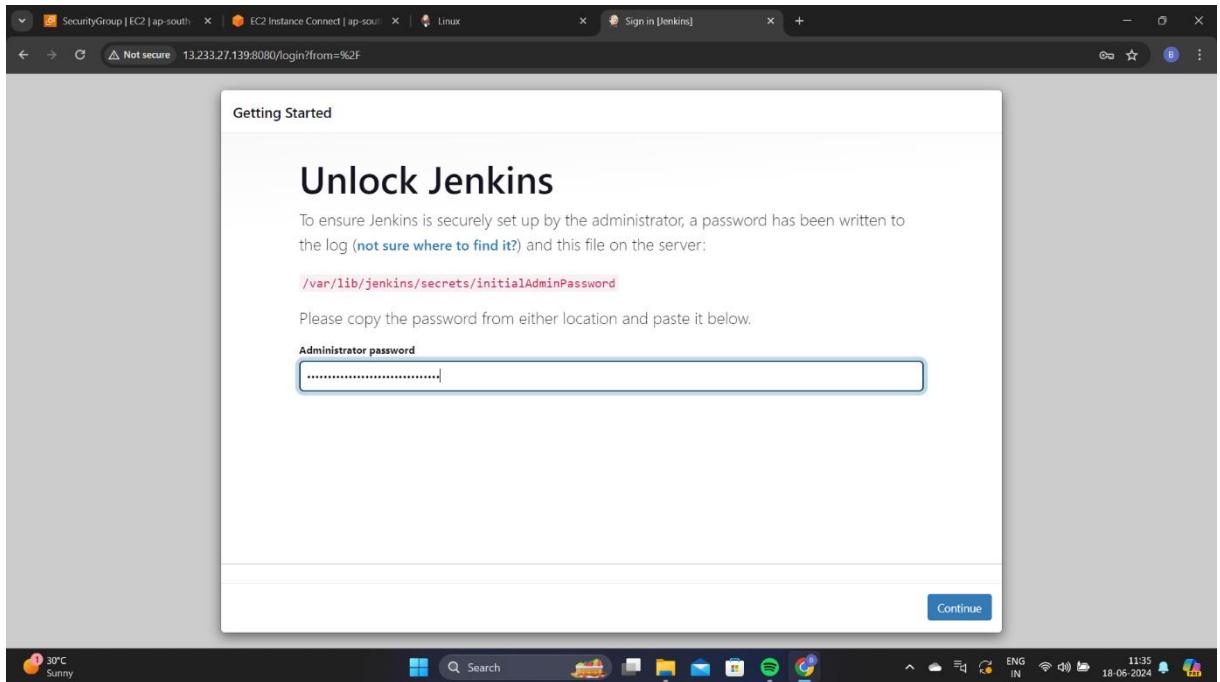
```
i-04e2d3b21ebc21992 (jenkins)
PublicIPs: 13.235.27.139 PrivateIPs: 172.31.39.158
```

The terminal window has a dark theme and includes a status bar at the bottom with system icons like battery level, signal strength, and date/time.

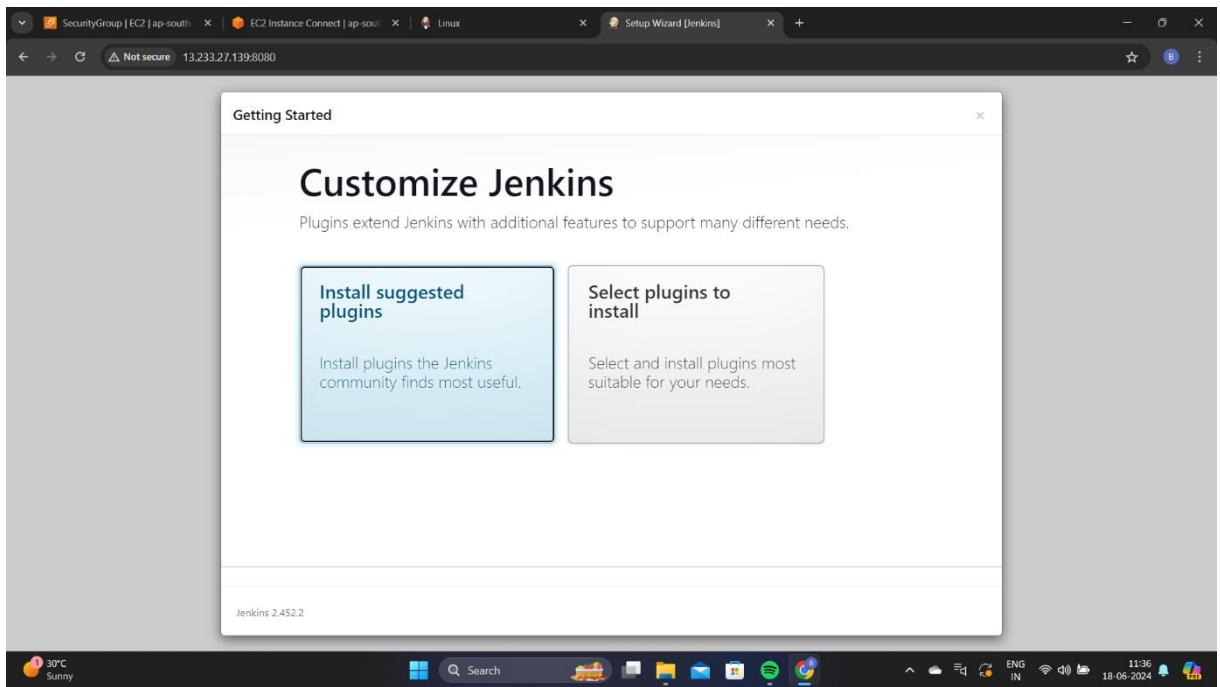


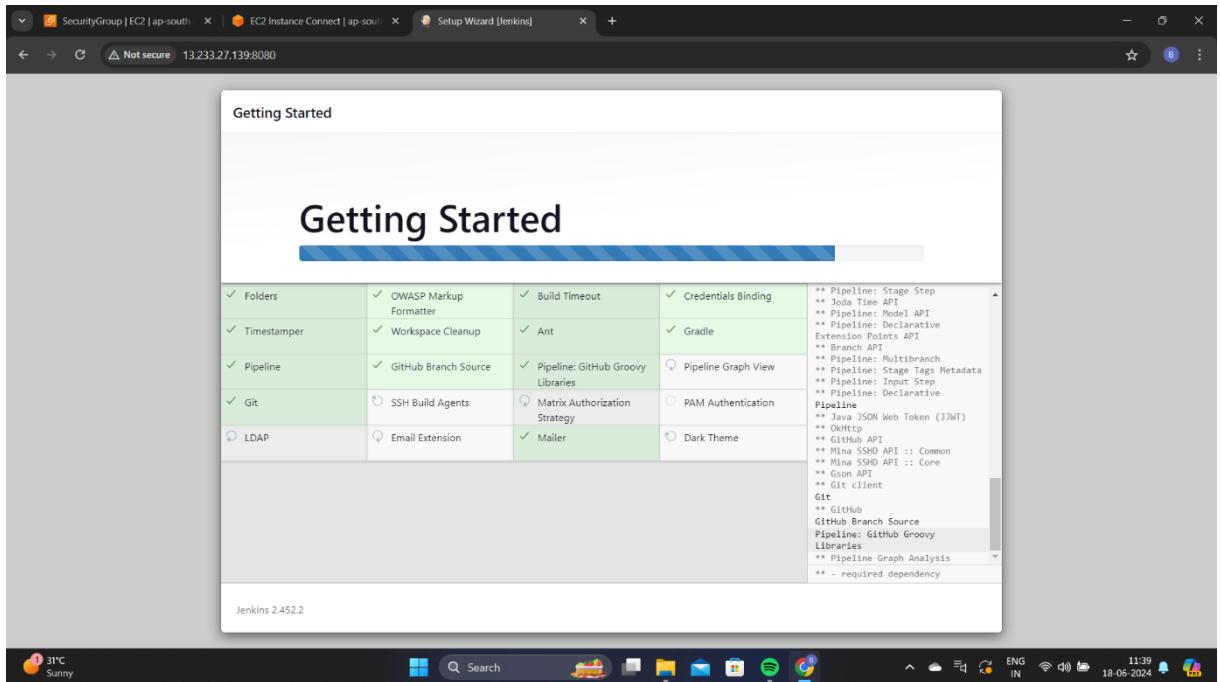
- ❖ Copy this path </var/lib/jenkins/secrets/initialAdminPassword>
- ❖ Cat </var/lib/jenkins/secrets/initialAdminPassword>
- ❖ Cat command used to show the default password for jenkins



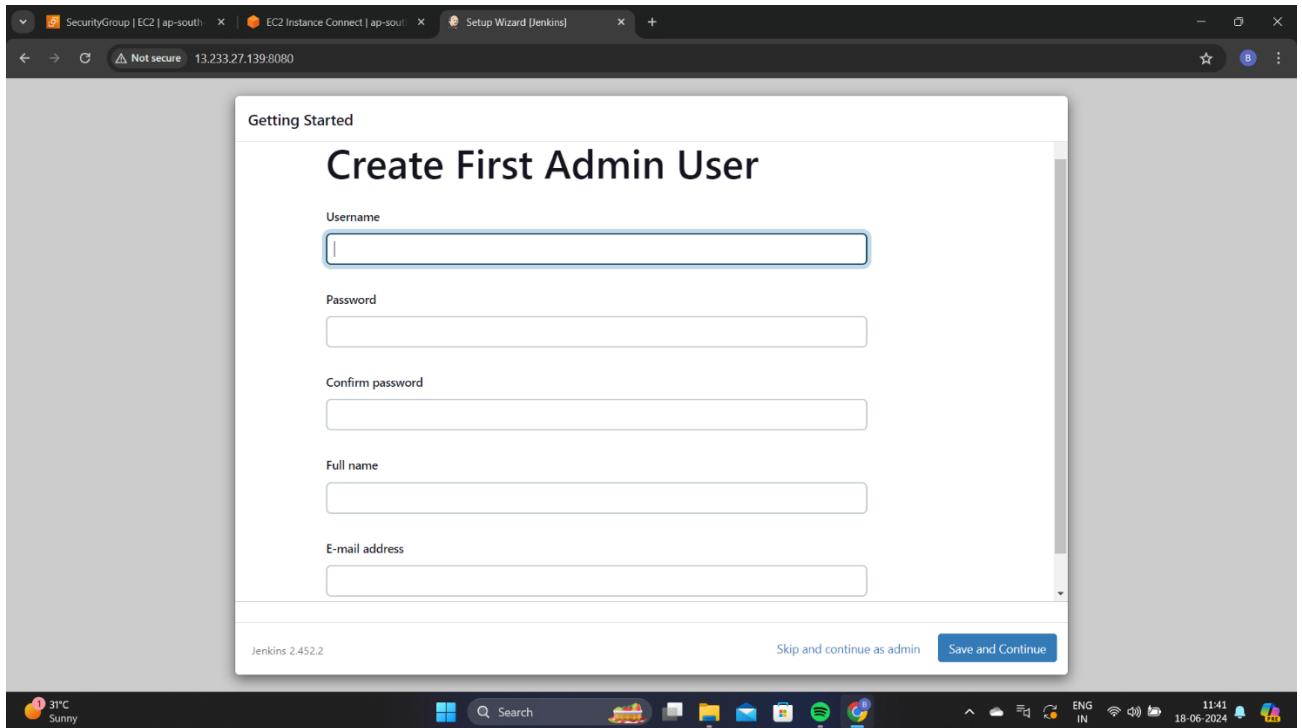


❖ Click on continue

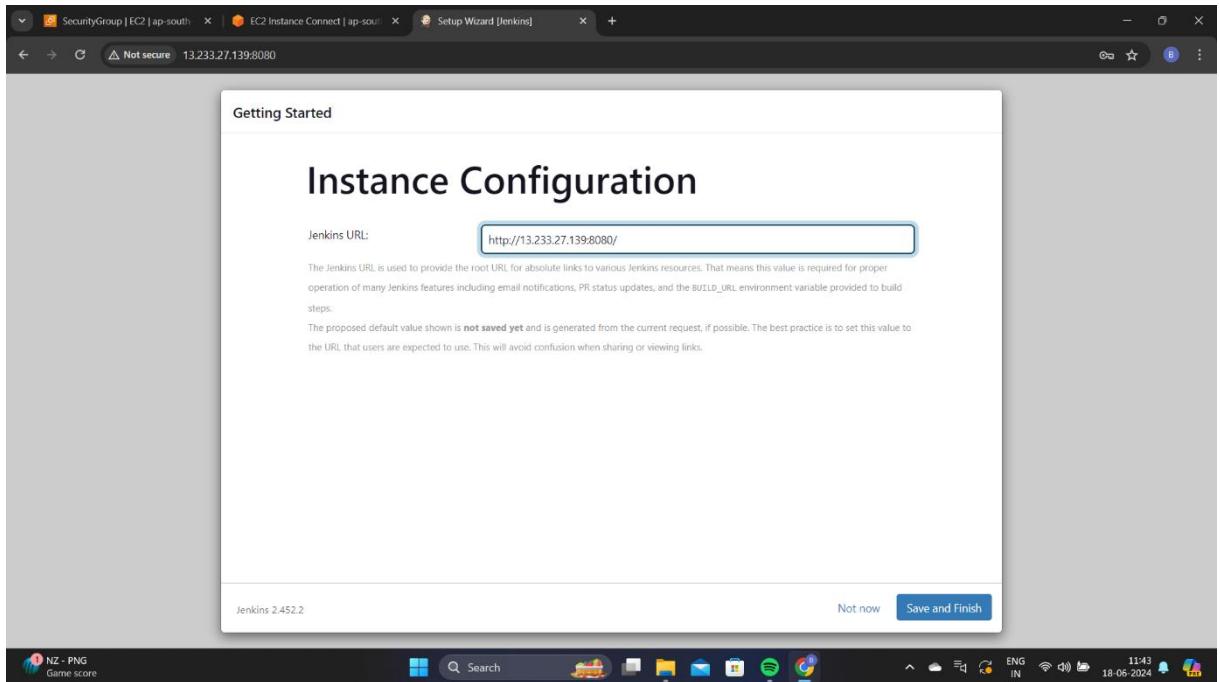




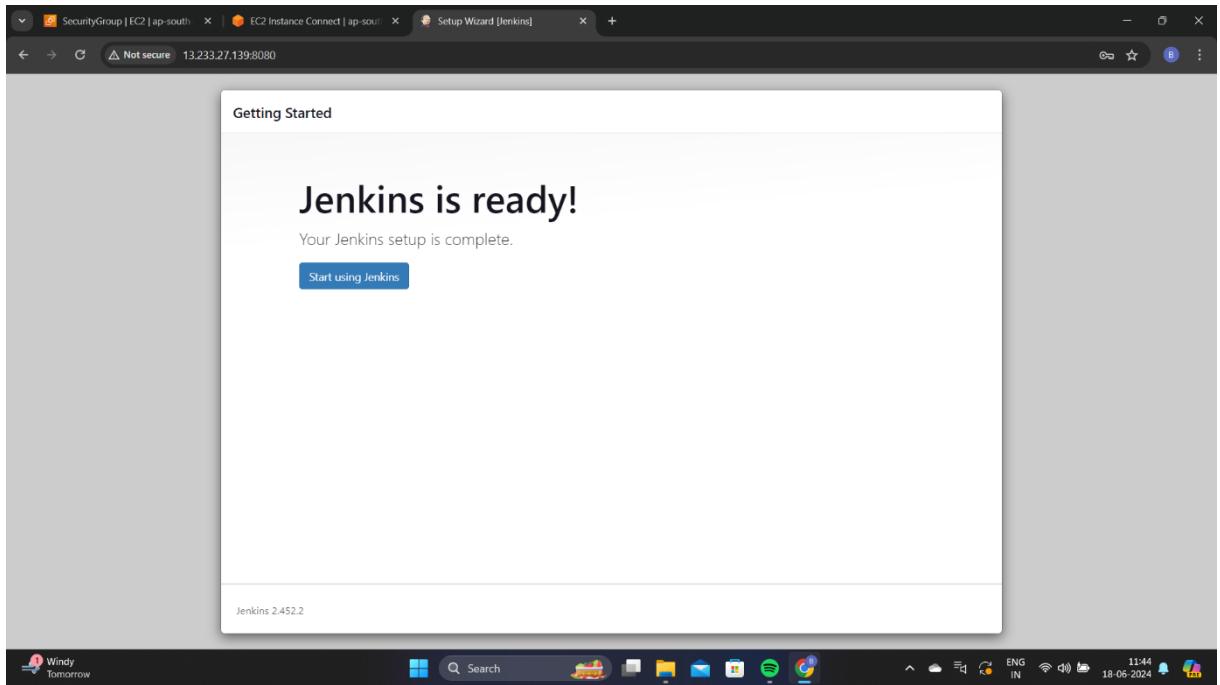
- ❖ Jenkins is getting started



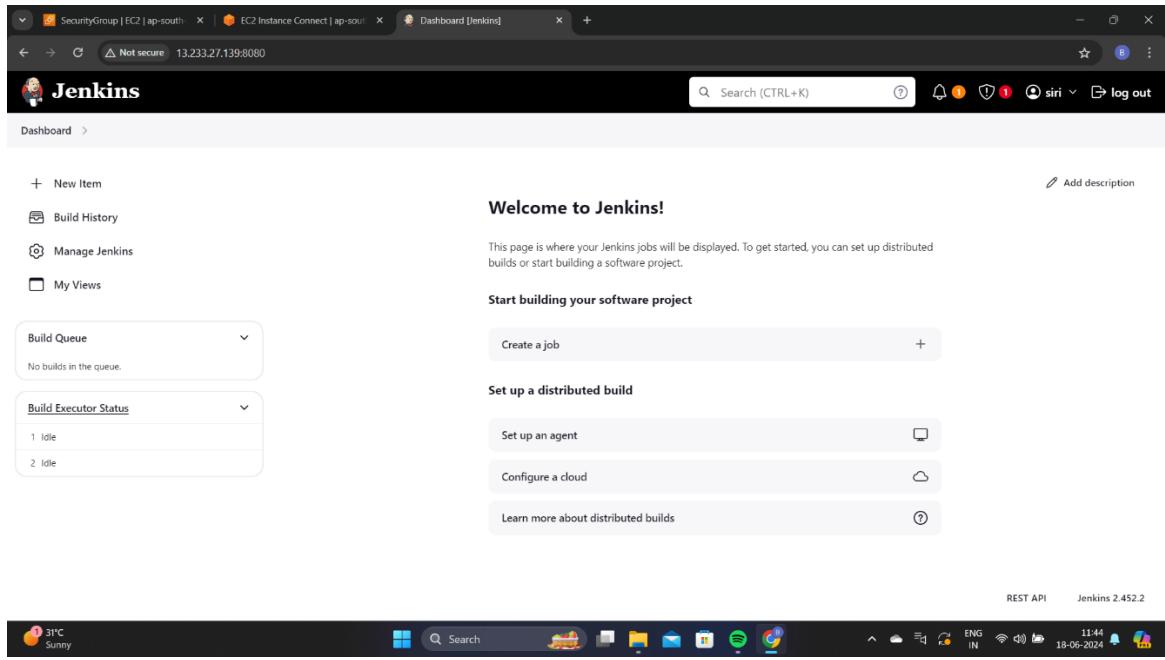
- ❖ Create First Admin User details
- ❖ Click Save and Continue



❖ Click on save and continue



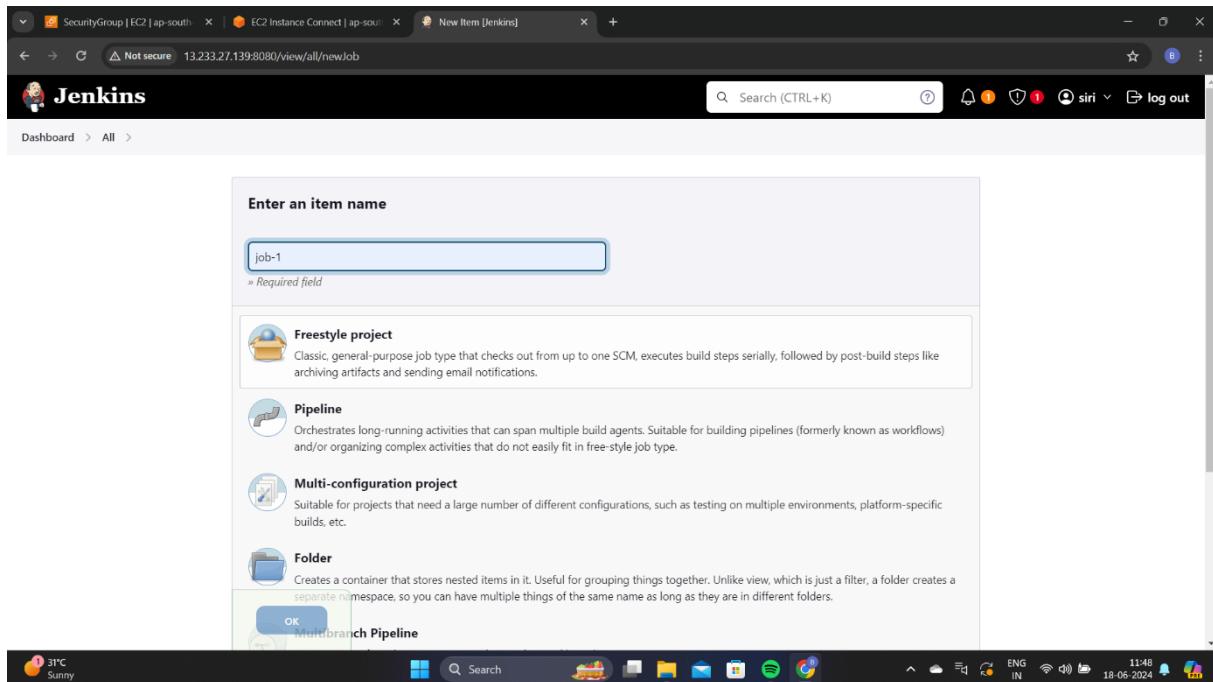
❖ Click on start using jenkins



- ❖ Jenkins dashboard will open

## Create a Job

- ❖ Click on New Item
- ❖ Enter an item name (job-1)
- ❖ Select Free Style Project
- ❖ Click on Ok



The screenshot shows the Jenkins configuration interface for a job named 'job-1'. The 'General' tab is selected. On the left, a sidebar lists options: General (selected), Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'General' section contains a 'Description' field (empty) and several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. A 'Save' button is at the bottom. The top navigation bar shows tabs for SecurityGroup, EC2, and job-1 Config [Jenkins]. The status bar at the bottom indicates it's an 'Not secure' connection.

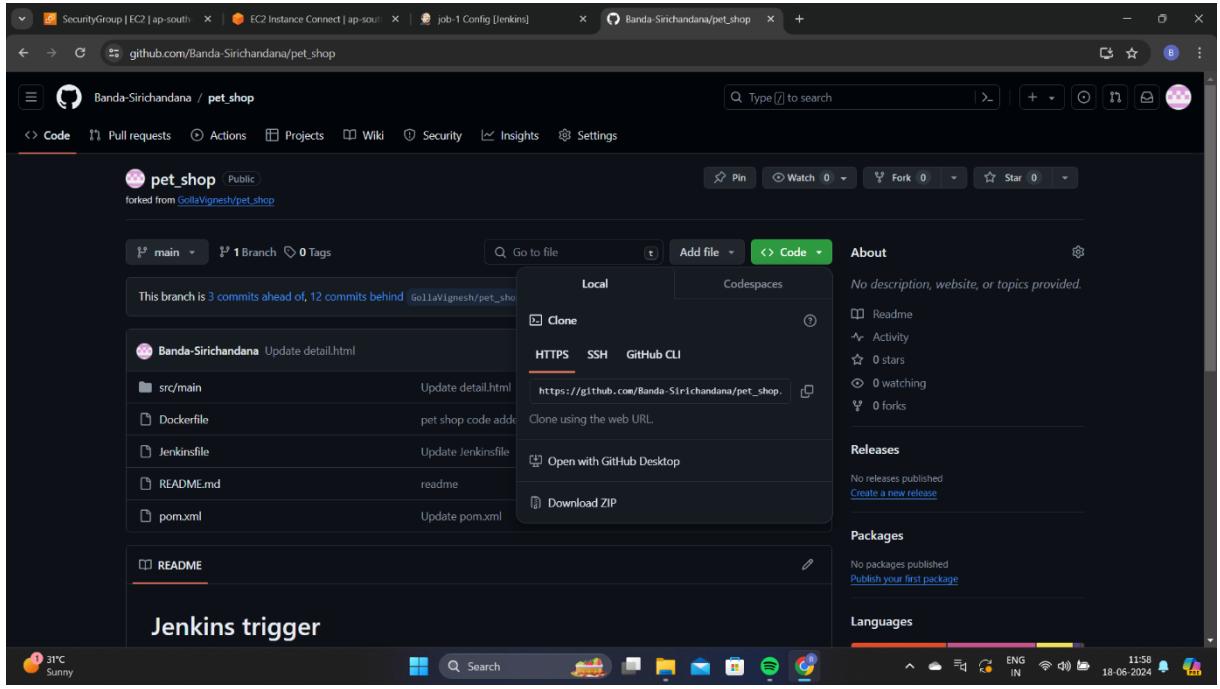
## ❖ Configure the Job

### Configure of Job:

#### ❖ We have to select GIT

The screenshot shows the 'Source Code Management' section of the Jenkins configuration page. Under 'Source Code Management', the 'Git' option is selected. The 'Repositories' section has a 'Repository URL' field which is empty and has a red error message: 'Please enter Git repository.' Below it is a 'Credentials' dropdown menu set to '- none -'. A '+ Add' button is available for adding new credentials. The 'Save' and 'Apply' buttons are at the bottom. The status bar at the bottom indicates it's an 'Not secure' connection.

- ❖ We should give Repository Url
- ❖ Copy the Url from GitHub
- ❖ Pet\_shop repository choosing from my GitHub account



- ❖ Copy the link from the code [https://github.com/Banda-Sirichandana/pet\\_shop.git](https://github.com/Banda-Sirichandana/pet_shop.git)
- ❖ Paste the link in Repository Url (job-1 configure)
- ❖ Install Git in server (yum install git -y)
- ❖ Change the branch specifier to main branch
- ❖ No need of any credentials
- ❖ Click on save

```

root@ip-172-31-39-158 ~# yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl-Git = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Git) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
--> Package git-core-doc.noarch 0:2.40.1-1.amzn2.0.3 will be installed
--> Package perl-Git.noarch 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.40.1-1.amzn2.0.3.noarch
--> Package perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Package           Arch      Version            Repository      Size
Installing:
git              x86_64   2.40.1-1.amzn2.0.3   amzn2-core      54 k

i-04e2d3b21ebc21992 (jenkins)

PublicIPs: 13.233.27.139 PrivateIPs: 172.31.39.158

```



**Configure**

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

**Repositories ?**

Repository URL ?  
https://github.com/Banda-Sirichandana/pet\_shop.git

Credentials ?  
- none -

Advanced ▾

Add Repository

**Branches to build ?**

Branch Specifier (blank for 'any') ?  
\*/main

Save Apply

- ❖ After saving of project click on build
- ❖ The project will be build
- ❖ After building of project we get build successfull

**job-1**

**Permalinks**

- Last build (#1), 40 sec ago
- Last stable build (#1), 40 sec ago
- Last successful build (#1), 40 sec ago
- Last completed build (#1), 40 sec ago

**Build History**

#1 Jun 18, 2024, 6:35 AM

Atom feed for all Atom feed for failures

REST API Jenkins 2.452.2

**Console Output**

```
Started by user siri
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/job-1
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Banda-Sirichandana/pet_shop.git
> git init /var/lib/jenkins/workspace/job-1 # timeout=10
Fetching upstream changes from https://github.com/Banda-Sirichandana/pet_shop.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/Banda-Sirichandana/pet_shop.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/Banda-Sirichandana/pet_shop.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision ae25ca8232d96ebf25c5074467b84b739cfa5bc0 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f ae25ca8232d96ebf25c5074467b84b739cfa5bc0 # timeout=10
Commit message: "Update detail.html"
First time build. Skipping changelog.
Finished: SUCCESS
```

BSE smicap +0.82%

ENG IN 12:36 18-06-2024

- ❖ Go to Jenkins dashboard
- ❖ Click on Manage Jenkins
- ❖ Click on Plugins
- ❖ Click on Available Plugins
- ❖ Search for Docker plugin
- ❖ Select Docker and Cloudbuild Build and Publish
- ❖ Click on install

The screenshot shows the Jenkins Manage Jenkins interface. The 'Manage Jenkins' link is highlighted. The 'System Configuration' section is expanded, showing options for Tools, Nodes, Clouds, and Plugins. The Plugins section is described as adding, removing, or enabling plugins to extend Jenkins functionality.

The screenshot shows the Jenkins Manage Jenkins interface with the 'Plugins' section selected. The 'Updates' link is highlighted. A message at the bottom states 'No updates available'.



The screenshot shows the Jenkins Plugins management interface. The left sidebar has tabs for Updates, Available plugins (which is selected), Installed plugins, Advanced settings, and Download progress. A search bar at the top right contains the query "docker". Below it, a table lists several Docker-related plugins:

	Name	Version	Category	Last Updated
<input checked="" type="checkbox"/>	Docker	1.6.2	Cloud Providers   Cluster Management   docker	14 days ago
<input type="checkbox"/>	Docker Commons	439.va_3cb_0a_6a_fb_29	Library plugins (for use by other plugins)   docker	11 mo ago
<input type="checkbox"/>	Docker Pipeline	580.vc0c340686b_54	pipeline   DevOps   Deployment   docker	27 days ago
<input type="checkbox"/>	Docker API	3.3.6-90.ve7c5c7535ddd	Library plugins (for use by other plugins)   docker	15 days ago
<input type="checkbox"/>	docker-build-step	2.12	Build Tools   docker	

At the bottom of the screen, there is a Windows taskbar with various icons and system status information.

This screenshot shows the same Jenkins Plugins management interface, but with a different set of search results for "docker". The table lists:

	Name	Version	Category	Last Updated
<input checked="" type="checkbox"/>	CloudBees Docker Build and Publish	1.4.0	Build Tools   docker	1 yr 9 mo ago
<input type="checkbox"/>	Amazon ECR	1.114.yfd2243062ff	aws	1 yr 4 mo ago
<input type="checkbox"/>	Docker Compose Build Step	1.0	Docker Compose plugin for Jenkins	5 yr 11 mo ago
<input type="checkbox"/>	Docker Slaves	1.0.7	docker	6 yr 11 mo ago
<input type="checkbox"/>	CloudBees Docker Custom Build Environment	1.7.3	docker	6 yr 4 mo ago
<input type="checkbox"/>	JFrog	1.5.0	.NET Development   Maven   npm   Deployment   docker	9 mo 21 days ago

The Windows taskbar at the bottom is identical to the one in the first screenshot.

The screenshot shows the Jenkins interface with the 'Manage Jenkins > Plugins' path selected. On the left, there's a sidebar with options like 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress'. The 'Download progress' option is highlighted. The main content area is titled 'Download progress' and includes a 'Preparation' section with a bulleted list: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Below this is a table of plugin download status:

Plugin	Status
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
Token Macro	Success
Build Timeout	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success

- ❖ Go to Manage Jenkins
- ❖ Click on Tools

The screenshot shows the Jenkins 'Manage Jenkins' page. At the top, it says 'You are running Jenkins on Amazon Linux 2. Jenkins stopped supporting Amazon Linux 2 as of 2023-11-16. Please upgrade to a supported operating system.' Below this, there are several configuration sections:

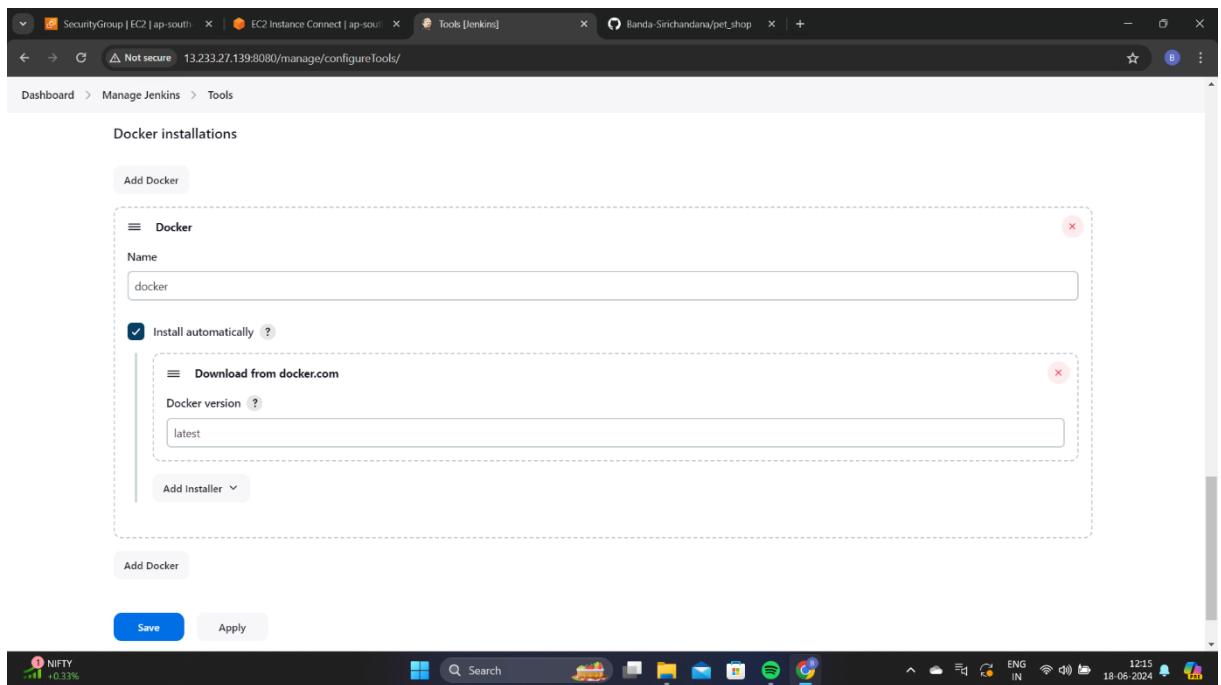
- System Configuration**: Includes links for 'System' (Configure global settings and paths), 'Tools' (Configure tools, their locations and automatic installers), 'Nodes' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), 'Clouds' (Add, remove, and configure cloud instances to provision agents on-demand), and 'Appearance' (Configure the look and feel of Jenkins).
- Security**: Includes links for 'Security' (Secure Jenkins; define who is allowed to access/use the system), 'Credentials' (Configure credentials), 'Users' (Configure users), and 'Credential Providers' (Configure the credential providers and types).

The screenshot shows the Jenkins 'Tools' configuration page. Under 'Maven Configuration', there are two dropdown menus: 'Default settings provider' set to 'Use default maven settings' and 'Default global settings provider' set to 'Use default maven global settings'. Below this is a 'JDK installations' section with a 'Add JDK' button. Under 'Git installations', there are 'Save' and 'Apply' buttons. The browser address bar shows the URL: 13.233.27.139:8080/manage/configureTools/.

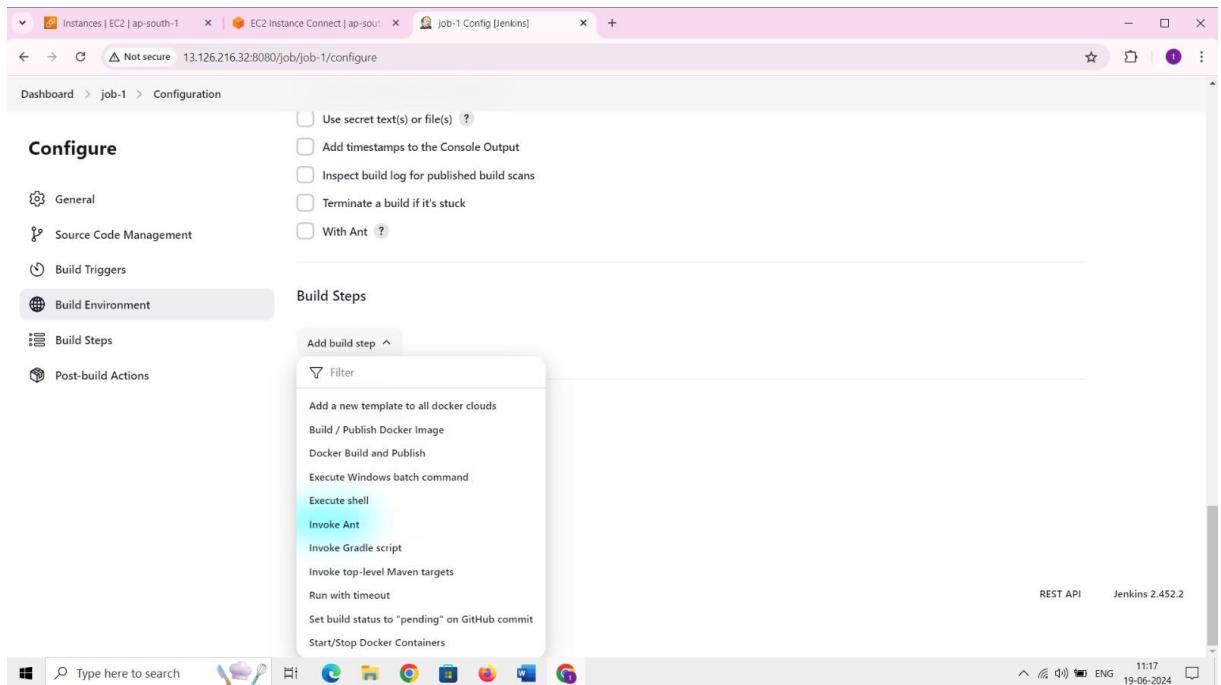
❖ Click on Add Maven

The screenshot shows the Jenkins 'Maven installations' configuration page. A modal dialog titled 'Add Maven' is open. It has a 'Name' field containing 'maven', a checked 'Install automatically' checkbox, and an 'Install from Apache' section with a 'Version' dropdown set to '3.9.8'. There is also an 'Add Installer' button. At the bottom of the dialog are 'Add Maven', 'Save', and 'Apply' buttons. The browser address bar shows the URL: 13.233.27.139:8080/manage/configureTools/.

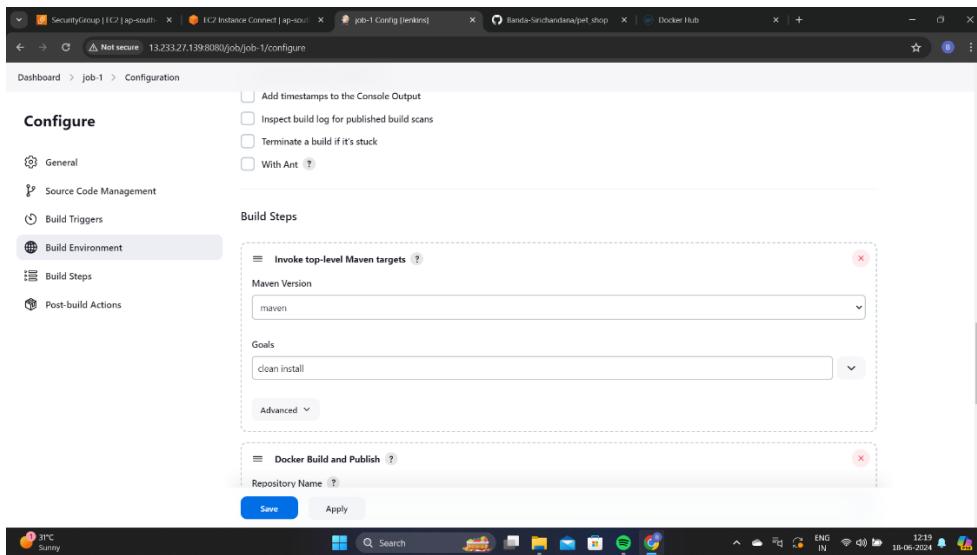
- ❖ Click on Add Docker
- ❖ Select checkbox Installautomatically
- ❖ Click on Add Installer
- ❖ Select Download from docker.com



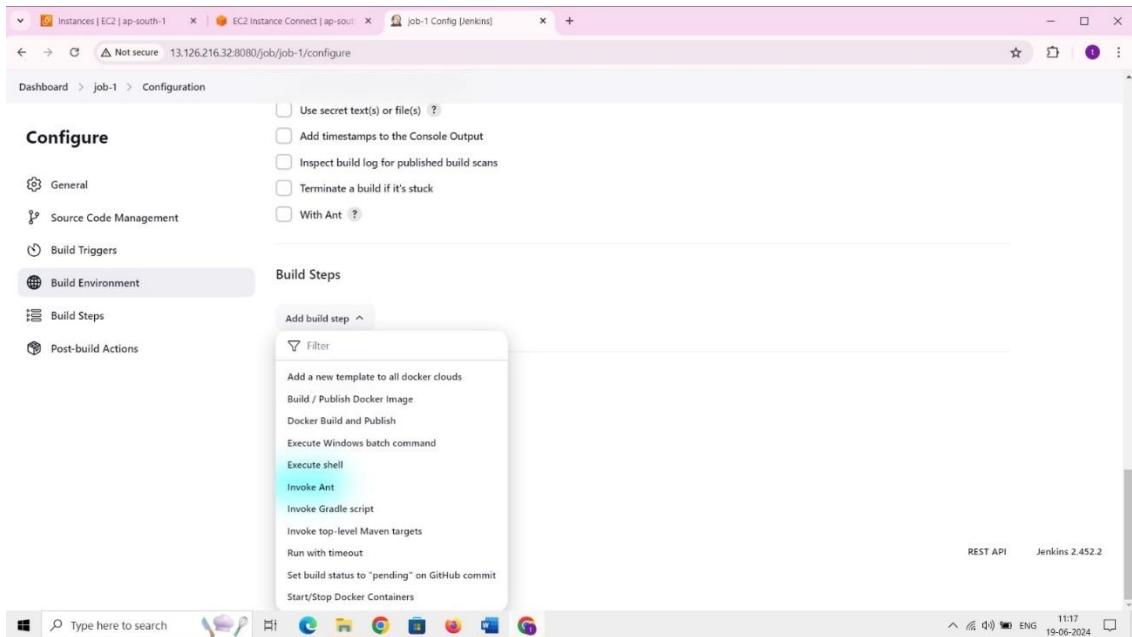
- ❖ Click on Save
- ❖ Go to Jenkins Dashboard
- ❖ Configure the Job1
- ❖ Click on Add build step



- ❖ Select Invoke top-level Maven targets
- ❖ Click on Maven Version
- ❖ Select Maven
- ❖ Click on Goals
- ❖ We should give Clean Install



- ❖ Click on Add build step
- ❖ Select Docker Build and Publish



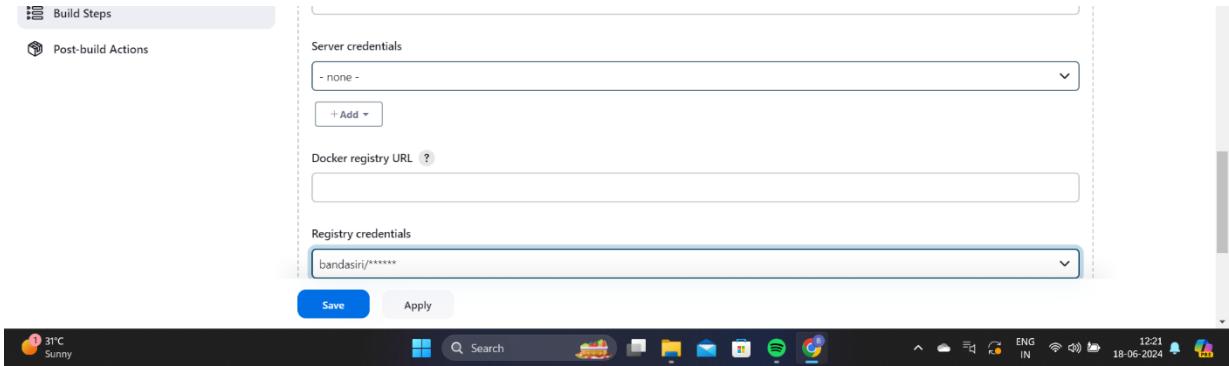
The screenshot shows the Jenkins job configuration page for 'job-1'. The 'Build Steps' section is selected. Under 'Docker Build and Publish', the 'Repository Name' is set to 'bandasiri'. There is a 'Tag' field, a 'Docker Host URI' field, and a 'Server credentials' dropdown set to '- none -'. A 'Registry credentials' section is also present.

- ❖ We should give DockerHub username/image name at Repository Name
- ❖ We should give Registry Credentials to Docker
- ❖ Click on Add
- ❖ Select Jenkins

The screenshot shows the 'Jenkins Credentials Provider: Jenkins' configuration dialog. It has fields for 'Kind' (set to 'Username with password'), 'Scope' (set to 'Global (Jenkins, nodes, items, all child items, etc)'), 'Username' (set to 'bandasiri'), and 'Password' (a masked input field). There is also a 'Treat username as secret' checkbox and an 'ID' field. At the bottom are 'Save' and 'Apply' buttons.

- ❖ We have to give Username as DockerHub username
- ❖ We have to give password as DockerHub password

❖ Click on save



- ❖ In server we have to Install docker (yum install docker -y)
- ❖ First we have Start Docker using command Systemctl start docker
- ❖ To check wheather the docker is in running state or off state we use command systemctl status docker
- ❖ If the docker is in off state then use systemctl enable --now docker

```
(root@ip-172-31-39-158 ~]# yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.25-1.amzn2.0.4 will be installed
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: libcgroup >= 0.40.rcl-5.15 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: pigz for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.7.11-1.amzn2.0.1 will be installed
--> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Package runc.x86_64 0:1.1.11-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

-----
```

Package	Arch	Version	Repository	Size
<b>Installing:</b>				
docker	x86_64	20.10.25-1.amzn2.0.4	amzn2extra-docker	43 M
<b>Installing for dependencies:</b>				
containerd	x86_64	1.7.11-1.amzn2.0.1	amzn2extra-docker	30 M
libcgroup	x86_64	0.41-21.amzn2	amzn2-core	66 k
pigz	x86_64	2.3.4-1.amzn2.0.1	amzn2-core	81 k
runc	x86_64	1.1.11-1.amzn2	amzn2extra-docker	3.0 M

i-04e2d3b21ebc21992 (jenkins)  
PublicIPs: 13.233.27.139 PrivateIPs: 172.31.39.158

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences AUT - FRA Game score 12:12 18-06-2024

```

Instances | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | Tejashwini-Gannamaneni/live01 | Docker Hub | job-1 Config [Jenkins] | - | □ | X
← → ⌂ ⌂ ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0ad0480ac0efb2e31&osUser=ec2-user&sshPort=22#/
star 🌟 📁 🌐 🌐 Mumbai Tejashwini ▾
aws Services Search [Alt+S]
(5/5): runc-1.1.11-1.amzn2.x86_64.rpm
Total 76 MB/s | 77 MB 00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : runc-1.1.11-1.amzn2.x86_64 1/5
  Installing : containerd-1.7.11-1.amzn2.0.1.x86_64 2/5
  Installing : libcgroup-0.41-21.amzn2.x86_64 3/5
  Installing : pigz-2.3.4-1.amzn2.0.1.x86_64 4/5
  Installing : docker-20.10.25-1.amzn2.0.4.x86_64 5/5
  Verifying : runc-1.1.11-1.amzn2.x86_64 1/5
  Verifying : containerd-1.7.11-1.amzn2.0.1.x86_64 2/5
  Verifying : libcgroup-0.41-21.amzn2.x86_64 3/5
  Verifying : runc-1.1.11-1.amzn2.x86_64 4/5
  Verifying : docker-20.10.25-1.amzn2.0.4.x86_64 5/5

Installed:
  docker.x86_64 0:20.10.25-1.amzn2.0.4

Dependency Installed:
  containerd.x86_64 0:1.7.11-1.amzn2.0.1           libcgroup.x86_64 0:0.41-21.amzn2             pigz.x86_64 0:2.3.4-1.amzn2.0.1           runc.x86_64 0:1.1.11-1.amzn2

Complete!
[root@ip-172-31-5-107 ~]# systemctl enable --now docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[root@ip-172-31-5-107 ~]# chmod 666 /var/run/docker.sock
[root@ip-172-31-5-107 ~]#

```

i-0ad0480ac0efb2e31 (jenkins)  
PublicIPs: 13.232.222.213 PrivateIPs: 172.31.5.107

## ❖ Give permissions to docker.sock By Using command chmod

```

SecurityGroup | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | job-1 #2 bandasiri/petshop Cor | Banda-Sirichandana/pet_shop | Docker Hub | - | + | 
← → ⌂ ⌂ ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-04e2d3b21ebc21992&osUser=root&region=ap-south-1&sshPort=22#/
star 🌟 📁 🌐 🌐 Mumbai B.siri chandana ▾
aws Services Search [Alt+S]
(root@ip-172-31-39-158 ~)# chmod 666 /var/run/docker.sock
(root@ip-172-31-39-158 ~)#

```

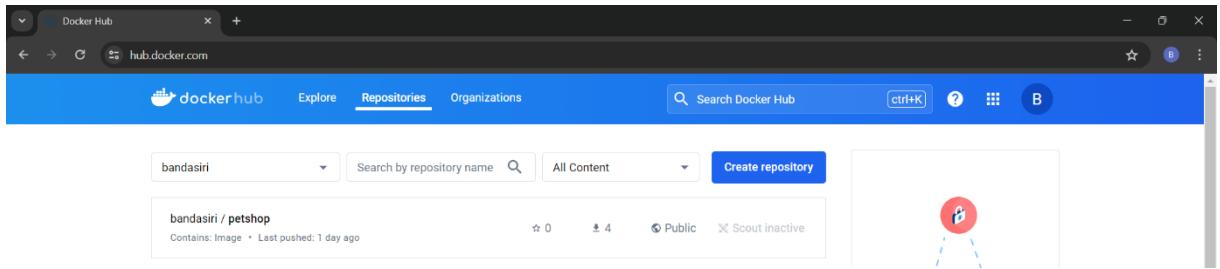
i-04e2d3b21ebc21992 (jenkins)  
PublicIPs: 13.233.27.139 PrivateIPs: 172.31.39.158

- ❖ Go to Jenkins Dashboard
- ❖ Select the Job-1
- ❖ Click on Build Now

The screenshot shows the Jenkins interface for job-1. The top navigation bar includes tabs for SecurityGroup, EC2, EC2 Instance Connect, job-1 [Jenkins], Banda-Sirichandana/pet\_shop, Docker Hub, and a log out button. The main content area displays the job-1 status with a green checkmark icon and the text "job-1". A sidebar on the left contains links for Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. Below the sidebar is a "Build History" section showing three builds: #3 (Jun 18, 2024, 6:54 AM), #2 (Jun 18, 2024, 6:52 AM), and #1 (Jun 18, 2024, 6:55 AM). The "Permalinks" section lists recent builds. The bottom of the screen shows a Windows taskbar with icons for weather (32°C, sunny), search, file explorer, mail, and browser, along with system status indicators.

The screenshot shows the Jenkins interface for job-1's console output. The top navigation bar and sidebar are identical to the previous screenshot. The main content area displays the "Console Output" tab with a green checkmark icon and the text "Console Output". The log output shows the build process starting with user "siri" running as SYSTEM, cloning from a GitHub repository, and performing a Maven clean install. The log concludes with a successful build message: "Build was successful (id: 3, version: 1)" and "Build step 'Maven Build' marked build as success". The bottom of the screen shows a Windows taskbar with icons for weather (32°C, sunny), search, file explorer, mail, and browser, along with system status indicators.

- ❖ Build Success
- ❖ Open DockerHub account
- ❖ After build success Automatically image pushes to my DockerHub Account



## Configuration of Build Triggers:

- ❖ Open Jenkins Dashboard
- ❖ Select Job-1
- ❖ Click on Configure
- ❖ Now we have to select Build Trigger
- ❖ Click on GitHub hook trigger GITscm polling

A screenshot of a web browser showing the Jenkins configuration page for a job named 'job-1'. The URL is '13.233.27.139:8080/job/job-1/configure'. The left sidebar shows 'Configure' sections: General, Source Code Management (with 'Build Triggers' selected), Build Environment, Build Steps, and Post-build Actions. The 'Build Triggers' section contains several checkboxes:

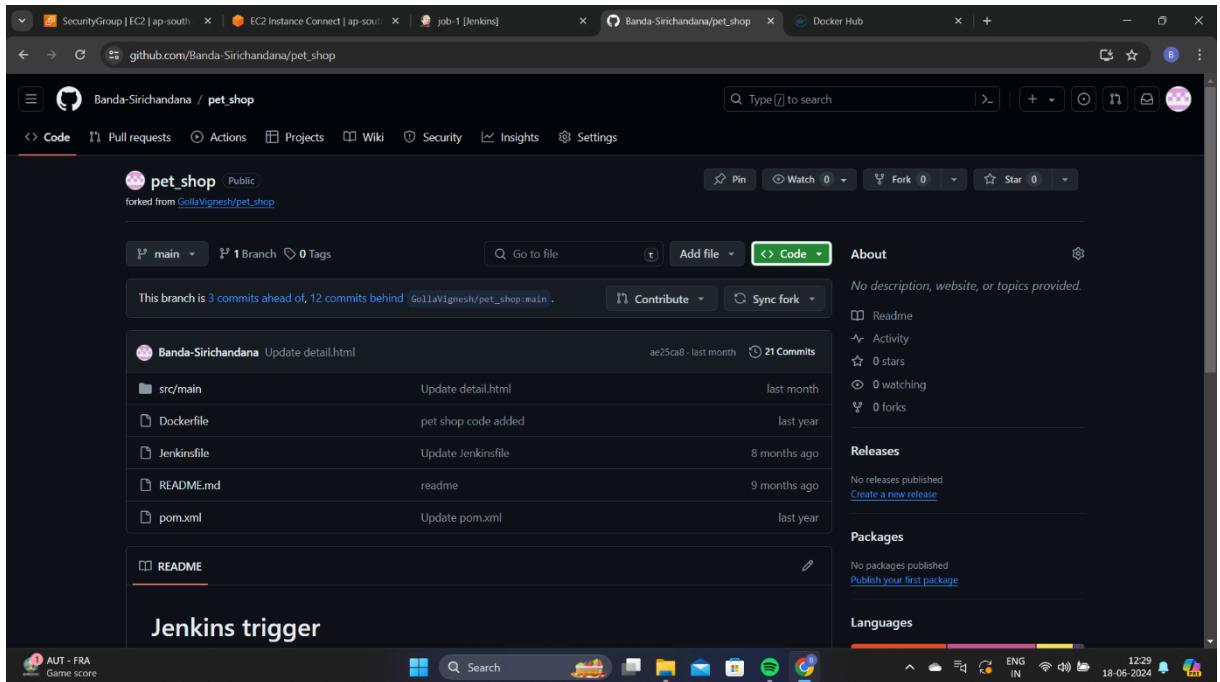
- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

The 'Build Environment' section below has several checkboxes:

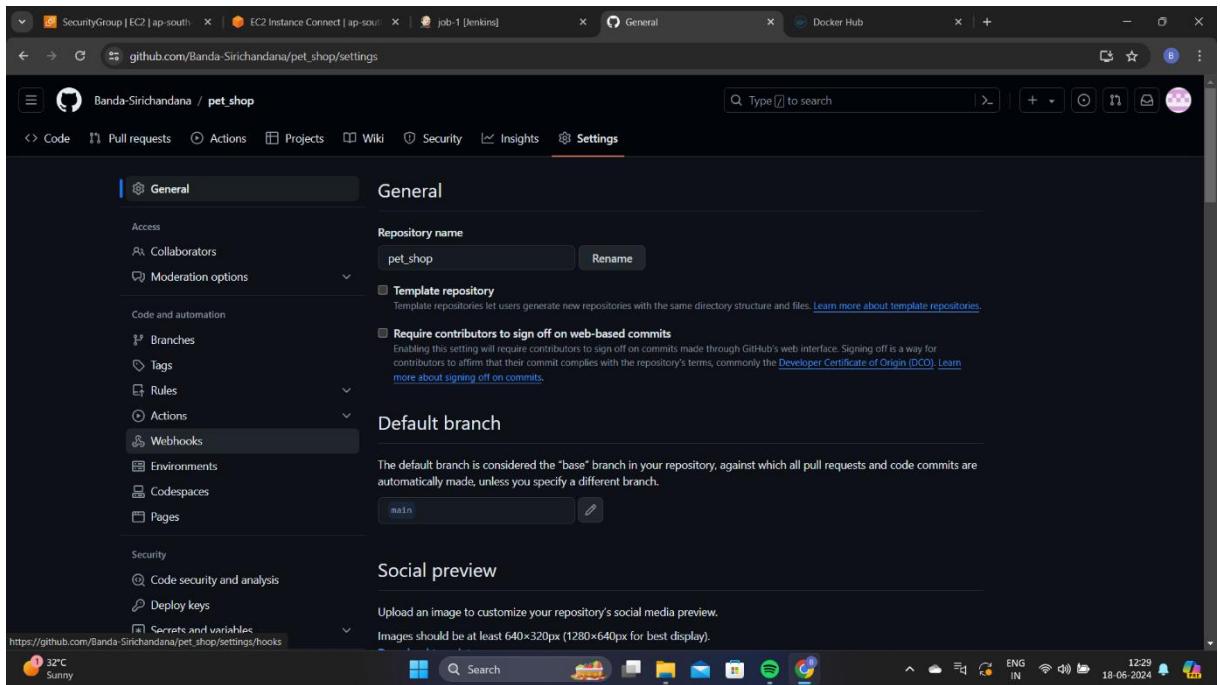
- Delete workspace before build starts
- Use secret text(s) or file(s) ?
- Add timestamps to the Console Output
- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant ?

At the bottom, there are 'Save' and 'Apply' buttons. The system tray at the bottom shows various icons and a battery level of 0.13%.

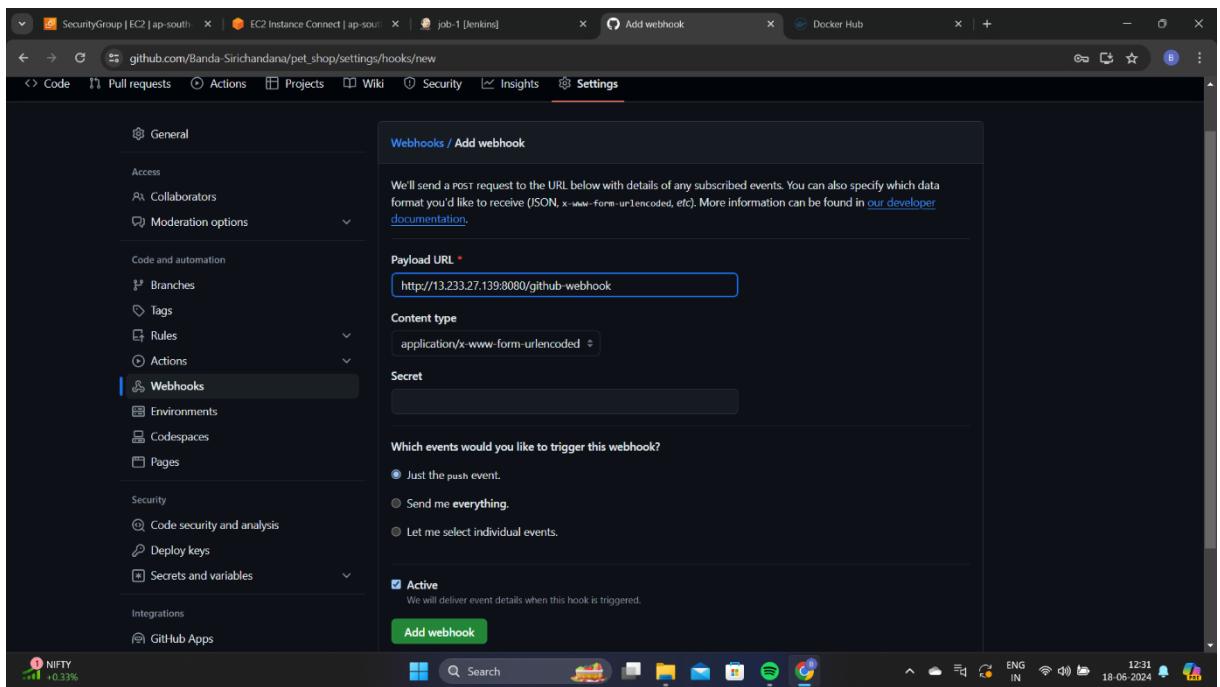
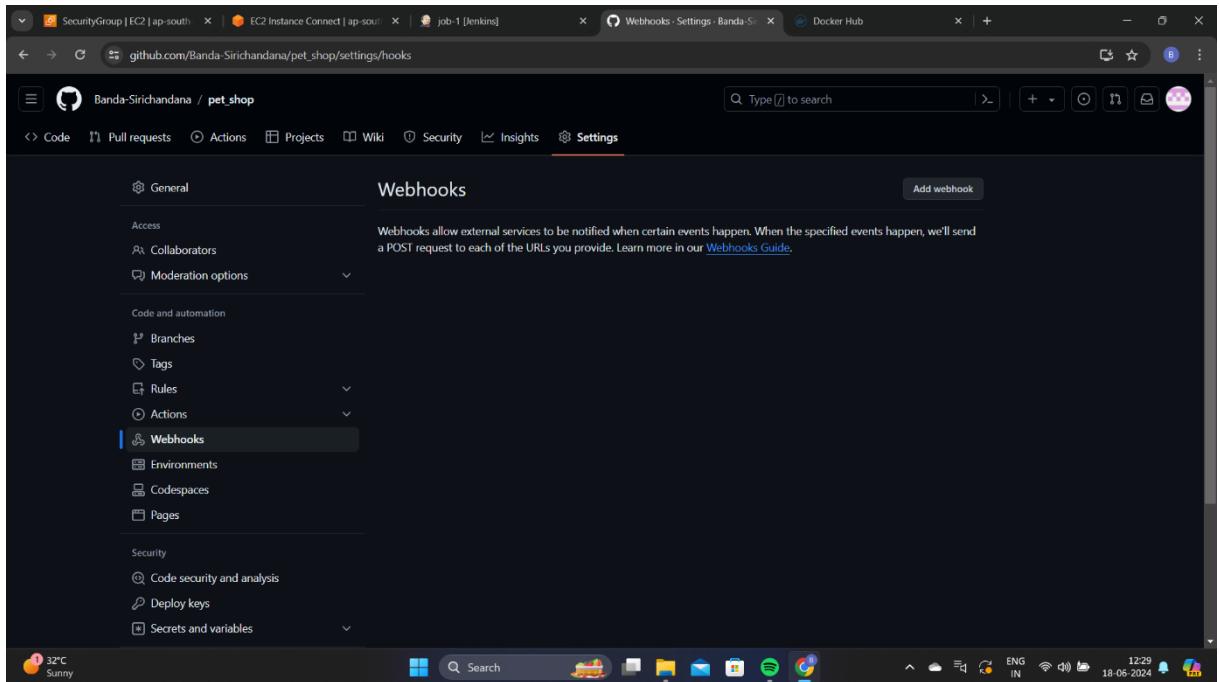
- ❖ Click on save
- ❖ Open GitHub Account
- ❖ Select Repository Pet\_shop



## ❖ Click on Settings

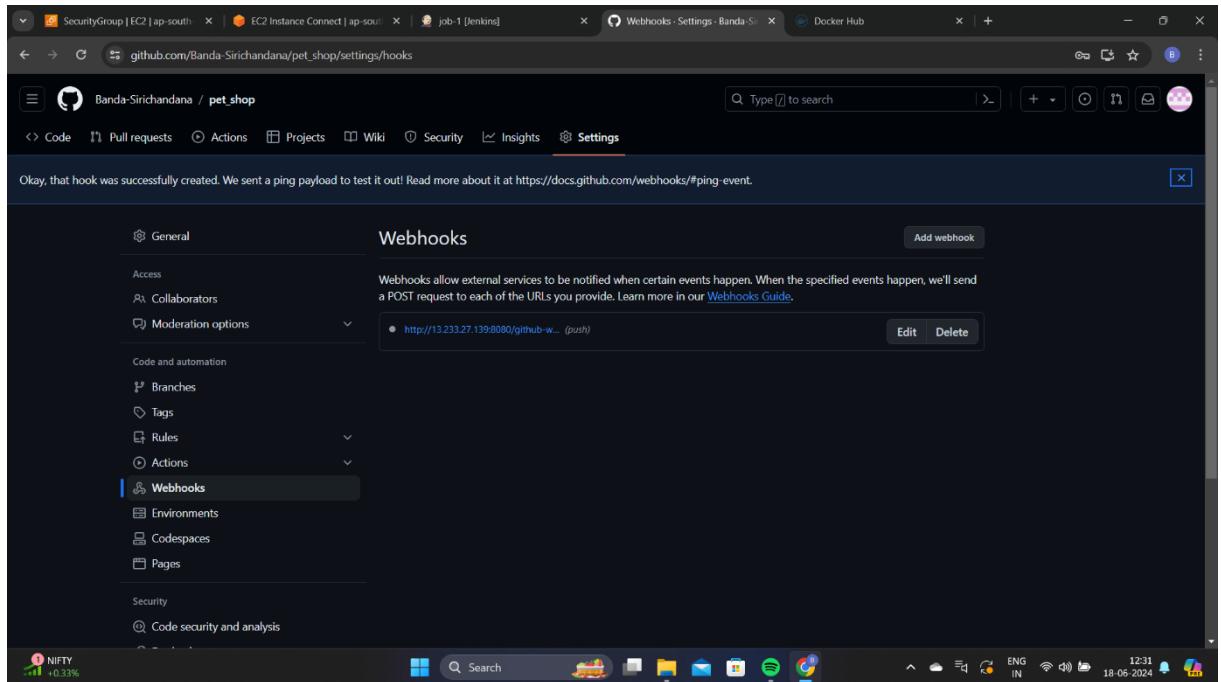


- ❖ Select Webhooks
- ❖ Click on Add Webhook
- ❖ We should give Payload Url

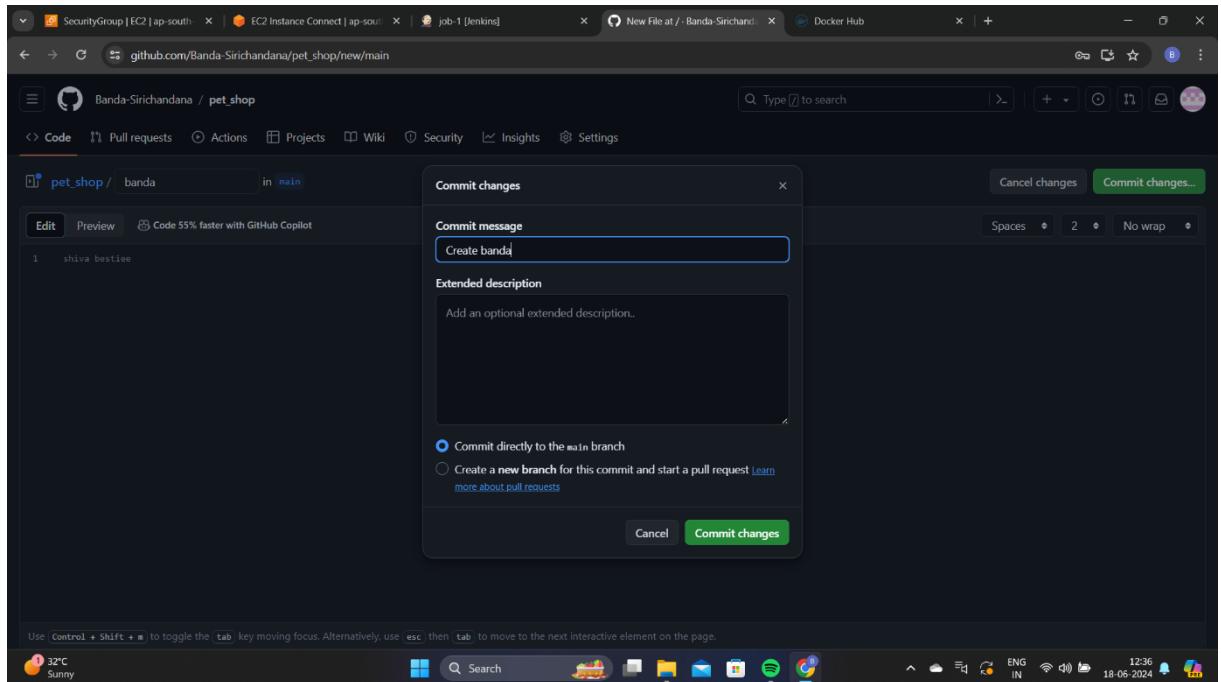


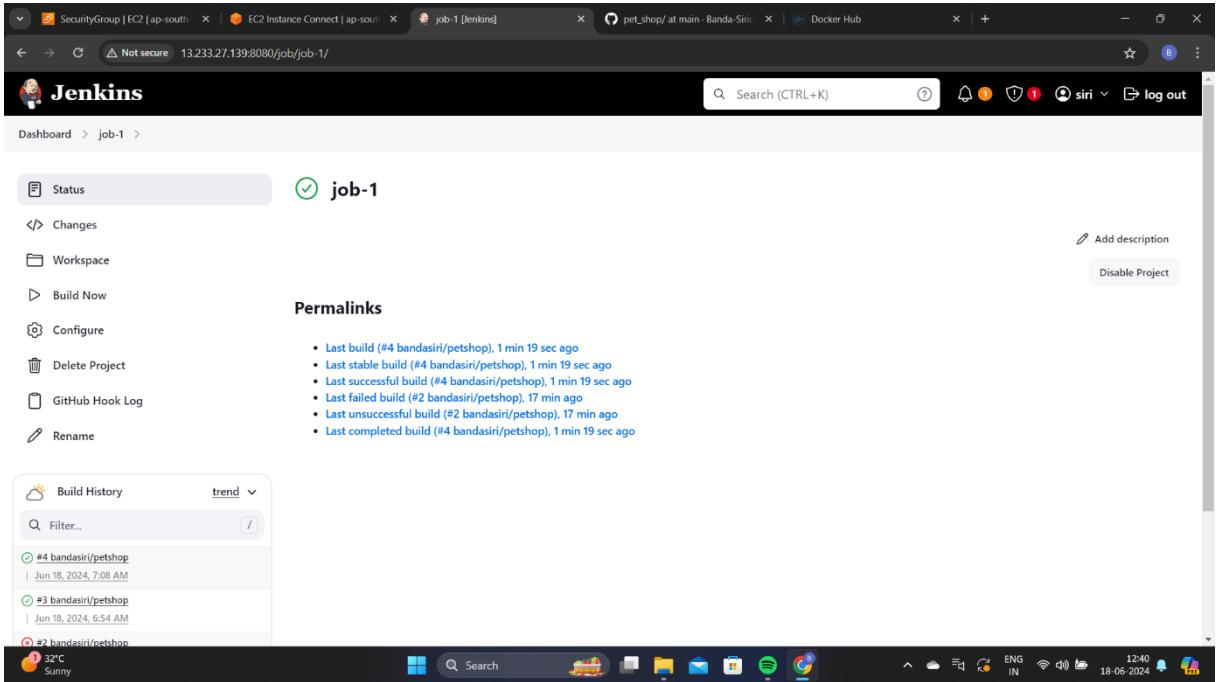
## ➤ Payload Url

- ❖ http://jenkins server PublicIPs:8080/github-webhook/
- ❖ click on Add Webhook

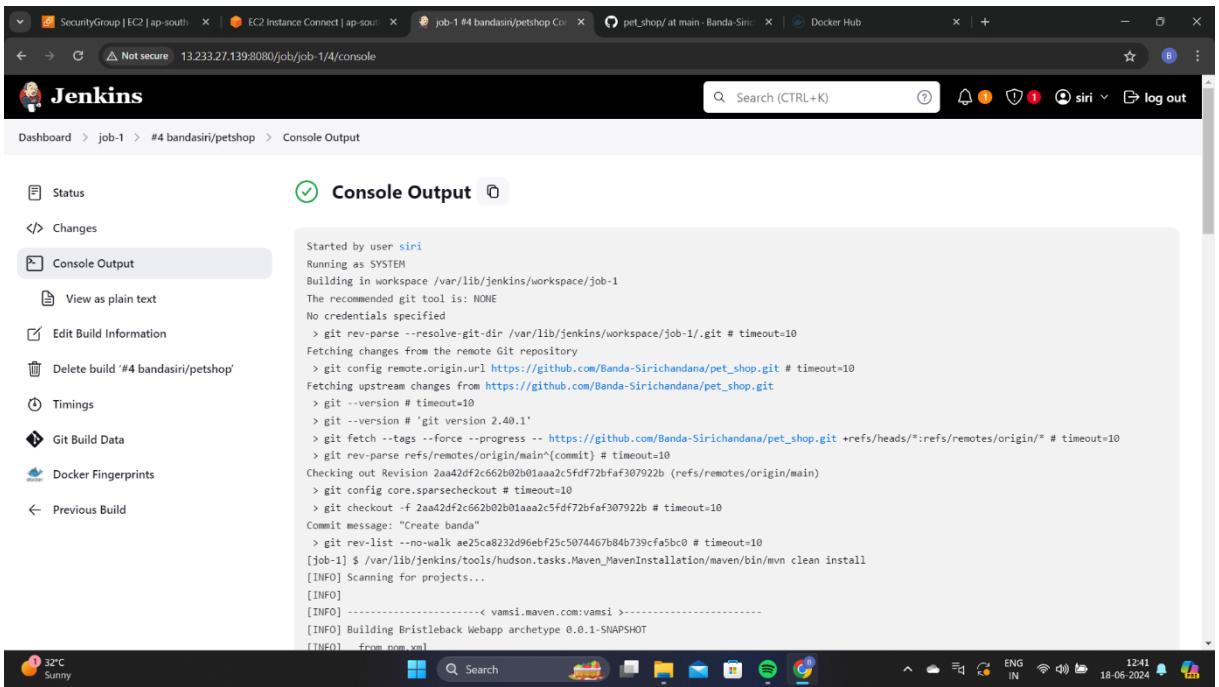


- ❖ Now we can do some changes in code
- ❖ After doing changes in code
- ❖ Click on commit changes
- ❖ It will build the job-1 automatically in jenkins





The screenshot shows the Jenkins interface for job-1. The top navigation bar includes tabs for SecurityGroup, EC2, job-1 [Jenkins], pet\_shop at main · Banda-Sirichandana/petshop, Docker Hub, and a Not secure warning. The main content area displays the job status as "job-1" with a green checkmark. A sidebar on the left lists options like Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, and Rename. To the right, a "Permalinks" section lists recent builds, and a "Build History" panel shows three builds: #4 (Jun 18, 2024, 7:08 AM), #3 (Jun 18, 2024, 6:54 AM), and #2 (Jun 18, 2024, 6:53 AM). The bottom status bar shows system information including weather (32°C, Sunny) and network connectivity.



The screenshot shows the Jenkins interface for job-1's console output. The top navigation bar is identical to the previous screenshot. The main content area displays the "Console Output" for build #4, which was started by user siri. The output log shows the git clone process, Maven clean install command, and the building of Bristleback Webapp archetype 0.0.1-SNAPSHOT. The bottom status bar shows system information including weather (32°C, Sunny) and network connectivity.

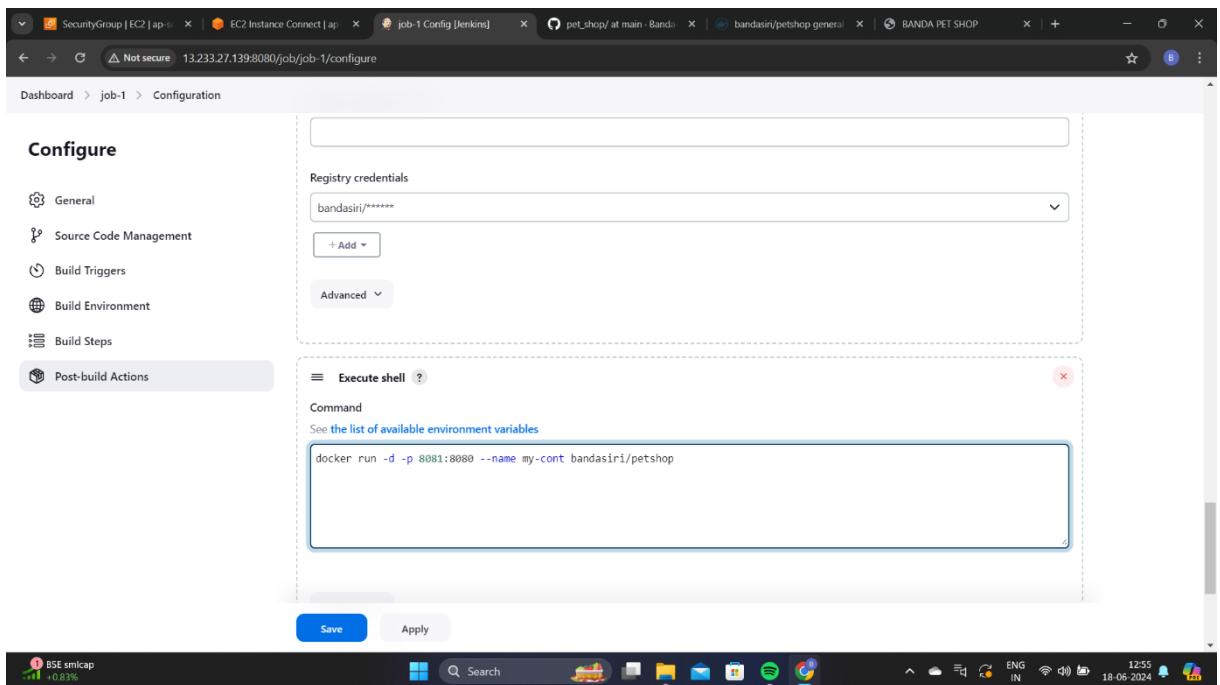
## Deployment of an Application:

- ❖ Click on Configure(job-1)
- ❖ Click on add Build step
- ❖ Select Execute shell
- ❖ We want to select this after maven
- ❖ docker rm -f my-cont

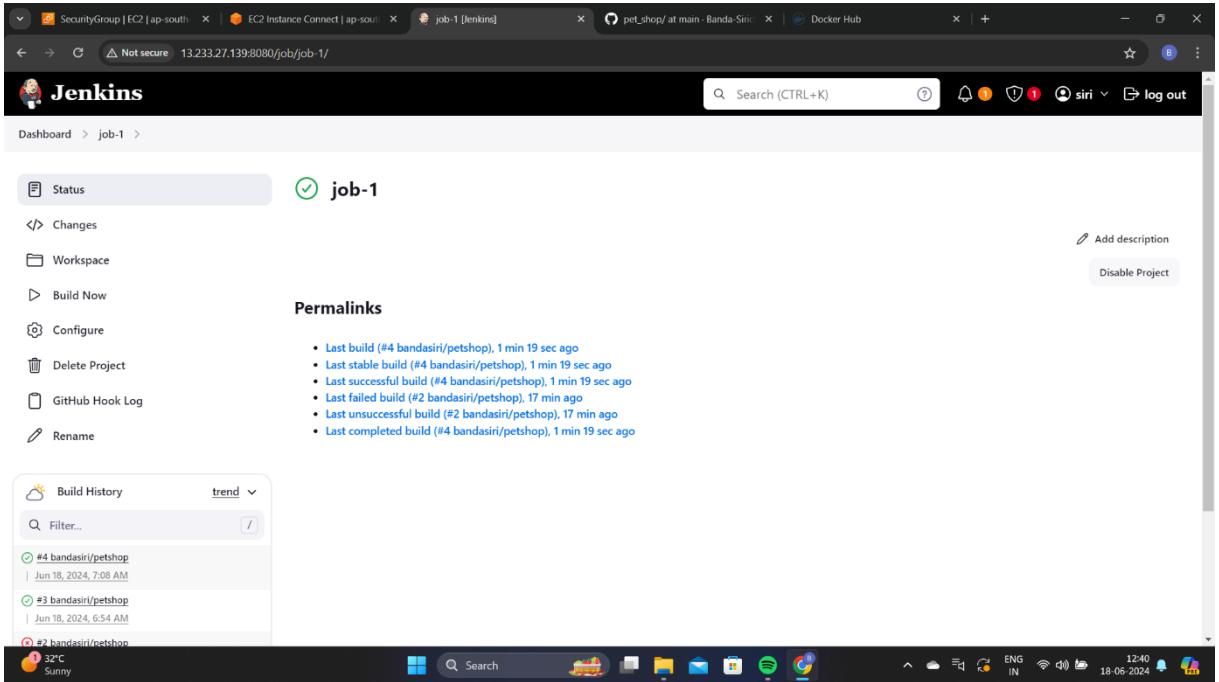
when we relaunch the application, it will remove previous container

- docker rmi bandasiri/petshop

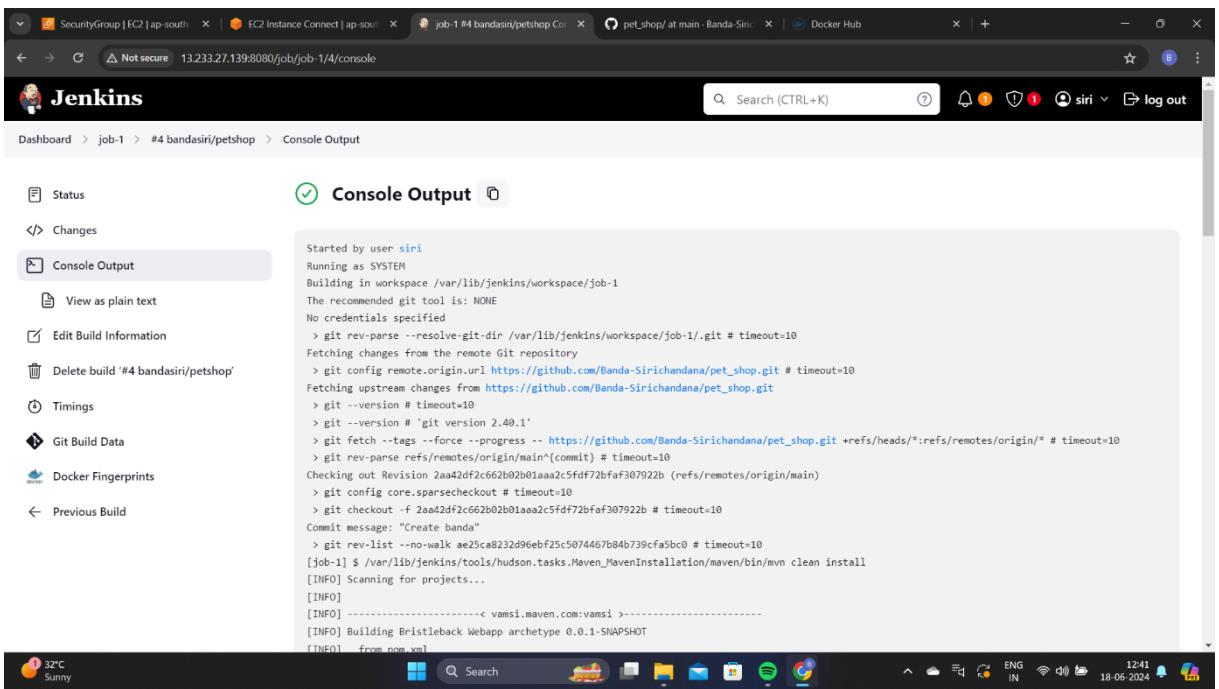
when we relaunch the application, it will remove previous image



- ❖ We should run command  
docker run -d -p 8081:8080 --name my-cont bandasiri/petshop
- ❖ Click on save
- ❖ It will build application automatically

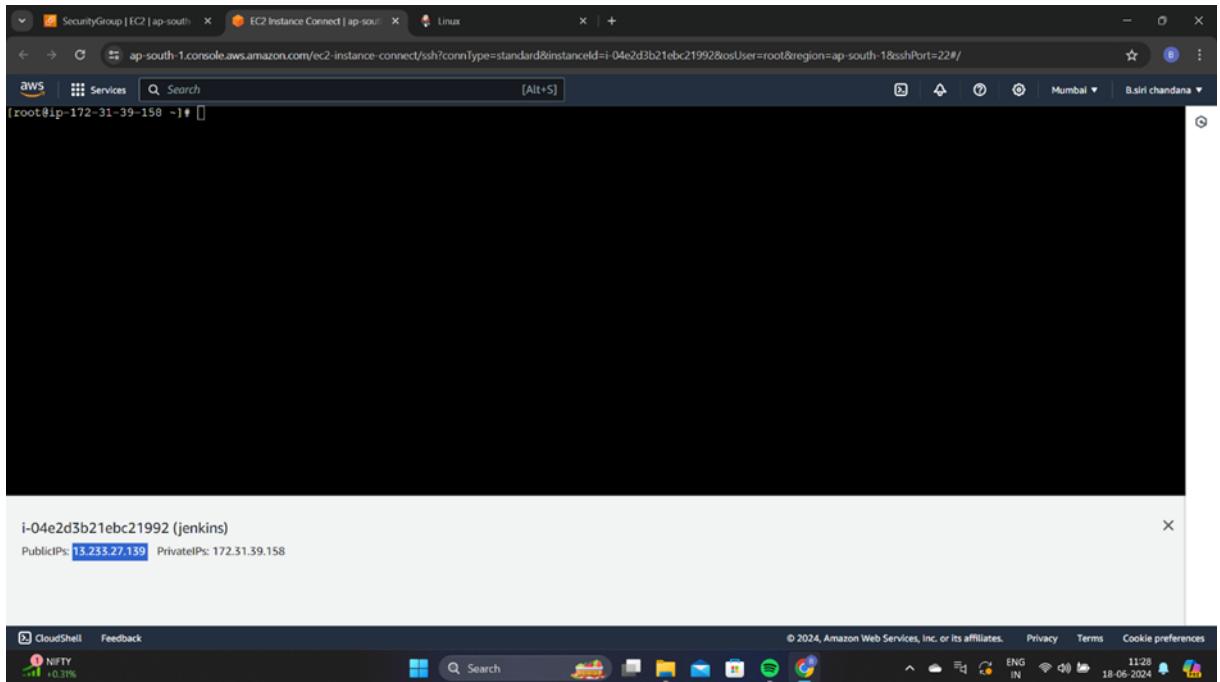


The screenshot shows the Jenkins interface for a job named 'job-1'. The top navigation bar includes tabs for 'SecurityGroup | EC2 | ap-south' and 'EC2 Instance Connect | ap-south'. The main title is 'job-1 [Jenkins]'. Below the title, the URL is '13.233.27.139:8080/job/job-1'. The page has a header with a Jenkins logo, a search bar, and a 'log out' button. On the left, a sidebar lists options: Status (selected), Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, and Rename. To the right, a 'Permalinks' section lists recent builds. A 'Build History' card shows the last four builds: #4 (green), #3 (green), and #2 (red). The system tray at the bottom shows a weather icon (32°C, Sunny) and various system icons.

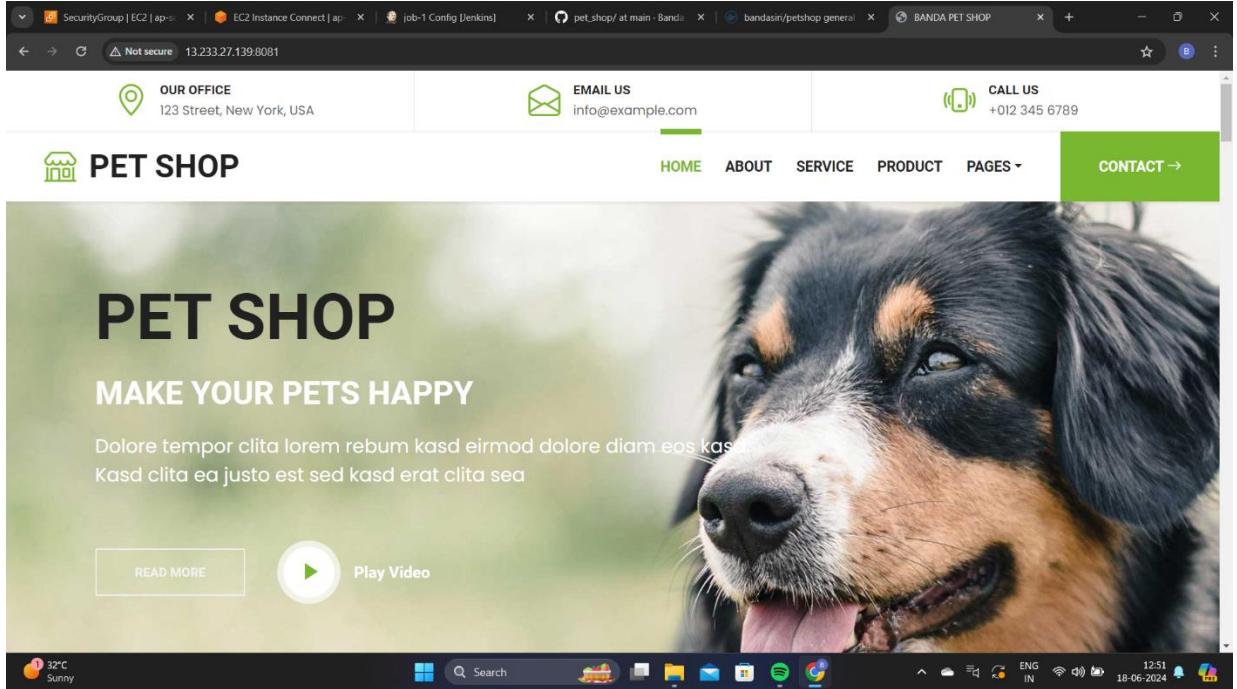


The screenshot shows the Jenkins interface for a specific build of 'job-1', labeled '#4 bandsari/petshop'. The top navigation bar and URL are identical to the previous screenshot. The main title is 'job-1 #4 bandsari/petshop [Console Output]'. The sidebar on the left has 'Console Output' selected. The central area displays the 'Console Output' for build #4, showing the command-line logs of the build process. The logs include commands like 'git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/job-1/.git # timeout=10', 'git config remote.origin.url https://github.com/Banda-Sirichandana/pet\_shop.git # timeout=10', and Maven installation commands. The system tray at the bottom shows a weather icon (32°C, Sunny) and various system icons.

❖ Copy the PublicIPs from jenkin server



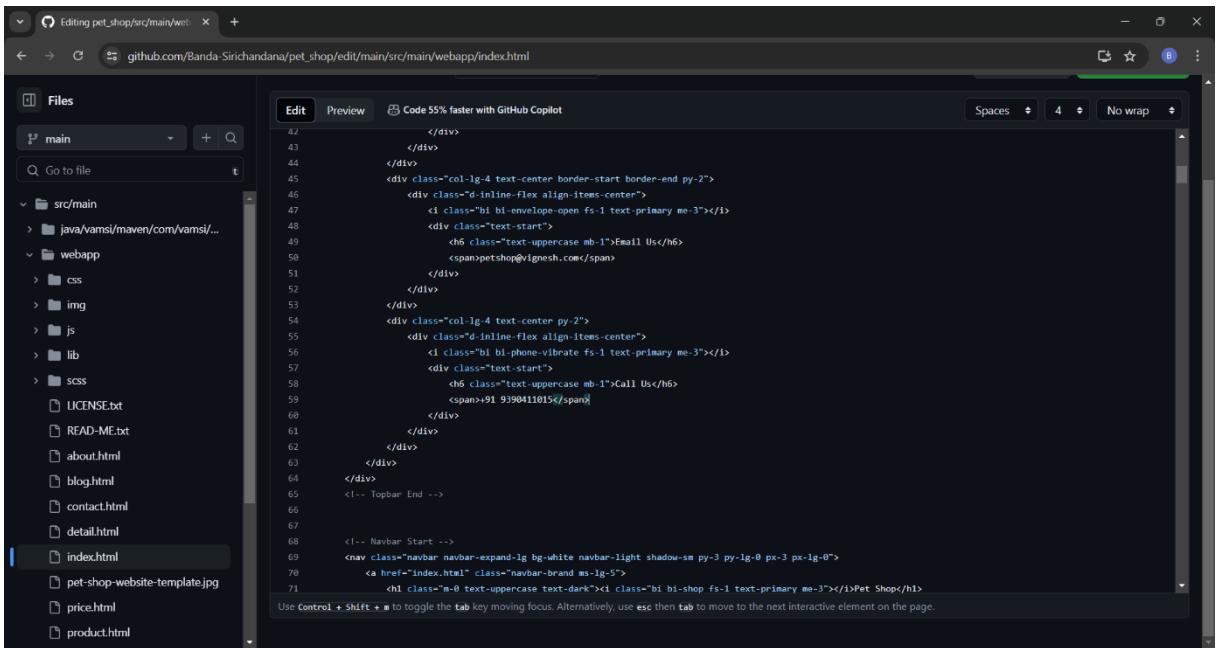
- ❖ Paste in google PublicIPs:8081
- ❖ Our Application will be Live



- ❖ This is our application
- ❖ Now I want to edit email and call us
- ❖ In my present application
- ❖ Email us [info@email.com](mailto:info@email.com)

- ❖ Call us = +0123456789
- ❖ Now I want to update my application as
- ❖ Email us = [petshop@vignesh.com](mailto:petshop@vignesh.com)
- ❖ Call us = 9390411015

- ❖ Open github account
- ❖ Click on repository (petshop)
- ❖ Edit your code at
- ❖ Src
- ❖ Main
- ❖ Webapps



The screenshot shows a browser window with the URL [github.com/Banda-Sirichandana/pet\\_shop/edit/main/src/main/webapp/index.html](https://github.com/Banda-Sirichandana/pet_shop/edit/main/src/main/webapp/index.html). The page title is "Editing pet\_shop/src/main/webapp/index.html". The left sidebar shows a file tree for the project structure:

```

Files
main
src/main
  java/vamsi/maven/com/vamsi/...
webapp
  css
  img
  js
  lib
  scss
    LICENSE.txt
    README.txt
    about.html
    blog.html
    contact.html
    detail.html
    index.html
    pet-shop-website-template.jpg
    price.html
    product.html

```

The right pane displays the content of the `index.html` file. The code includes sections for contact information (Email Us, Phone Number) and a navigation bar.

```

42      </div>
43      </div>
44      <div class="col-lg-4 text-center border-start border-end py-2">
45        <div class="d-inline-flex align-items-center">
46          <i class="bi bi-envelope-open fs-1 text-primary me-3"></i>
47          <div class="text-start">
48            <h6 class="text-uppercase mb-1">Email Us</h6>
49            <span>petshop@vignesh.com</span>
50          </div>
51        </div>
52      </div>
53      <div class="col-lg-4 text-center py-2">
54        <div class="d-inline-flex align-items-center">
55          <i class="bi bi-phone-vibrate fs-1 text-primary me-3"></i>
56          <div class="text-start">
57            <h6 class="text-uppercase mb-1">Call Us</h6>
58            <span>+91 9390411015</span>
59          </div>
60        </div>
61      </div>
62      </div>
63    </div>
64  </div>
65  <!-- Topbar End -->
66
67
68  <!-- Navbar Start -->
69  <nav class="navbar navbar-expand-lg bg-white navbar-light shadow-sm py-3 py-lg-0 px-3 px-lg-0">
70    <a href="index.html" class="navbar-brand ms-1">
71      <h1 class="me-0 text-uppercase text-dark"><i class="bi bi-shop fs-1 text-primary me-3"></i>Pet Shop</h1>

```

A GitHub Copilot icon is visible above the code editor. The status bar at the bottom of the editor indicates: "Code 55% faster with GitHub Copilot".

The screenshot shows a GitHub commit dialog box. The commit message is "Update index.html". The "Commit directly to the main branch" option is selected. The "Commit changes" button is highlighted in green.

- ❖ Click on commit changes
- ❖ Automatically it will build job 1
- ❖ Refresh the page
- ❖ Automatically changes has done
- ❖ Updated application

