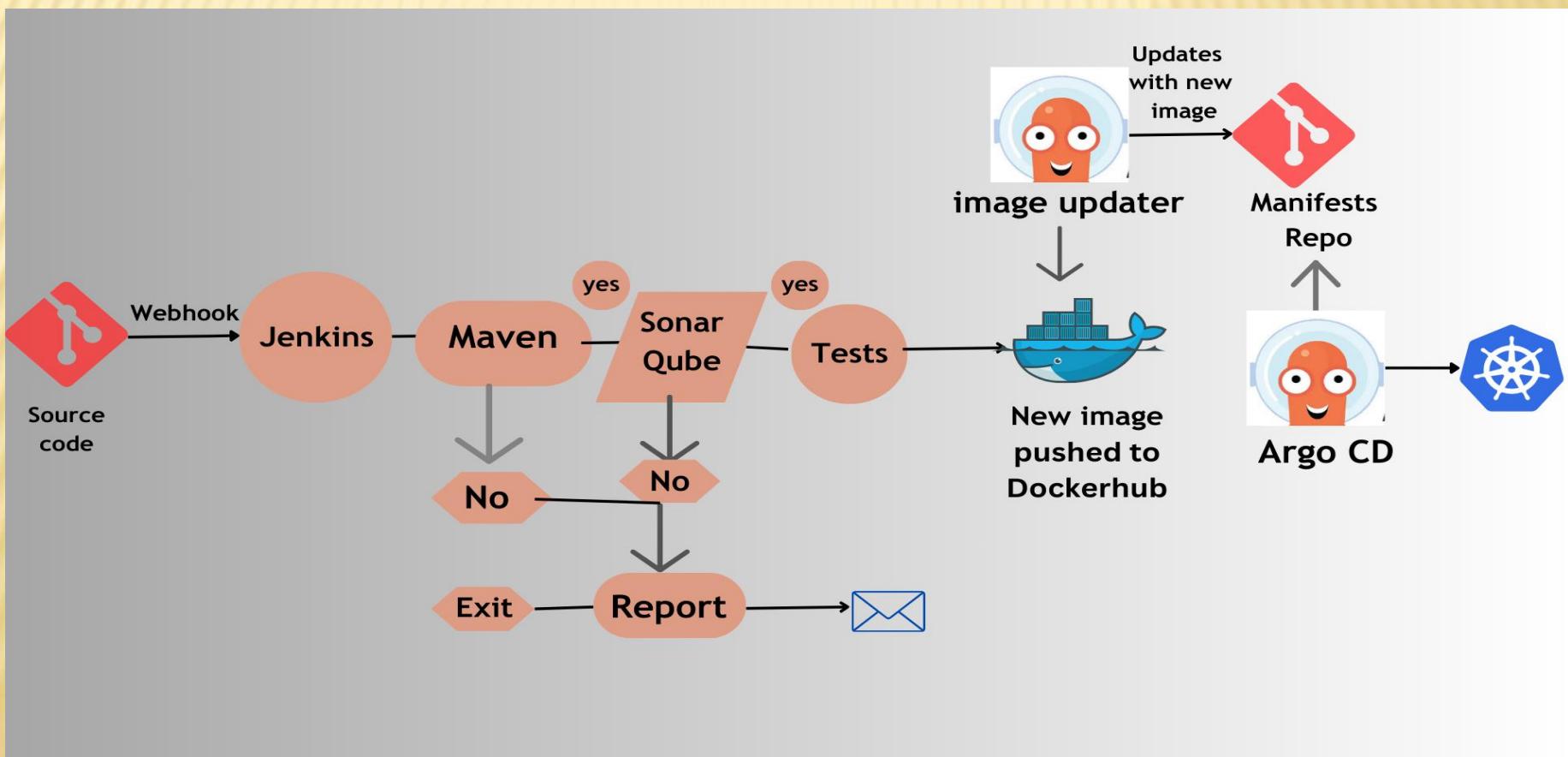


JENKINS END TO END CICD IMPLEMENTAION

Goal:

Build a spring boot based Java application using Maven and deploy this application on kubernetes using CICD

We accomplish this by adhering to the procedure outlined below, which integrates CI/CD implementation at each stage



Activities in this project:

- ✓ Build a Java application using Maven including configuration & installation
- ✓ Setup a sonar server locally and send the report of code execution(static code analysis) to sonarqube
- ✓ Build a docker image for the artifact and push it to dockerhub
- ✓ Automatically update the manifest repository or source code repository using a shell script
- ✓ Deploy the manifest automatically to a Kubernetes cluster using Argo CD

Step1:

Set up an EC2 instance in your AWS account, utilizing Ubuntu with an instance type of t2.large to effectively utilize resources such as 2 CPUs and 8GB of memory, which are sufficient for this project

Checkout below video by Abhishek Veermalla , if you have any doubts on how to create EC2 instance.

https://youtu.be/MkIRh1mi8Ms?si=ZXbIEOUZ3xw_jQZp

The screenshot shows the AWS Management Console with the EC2 service selected. The breadcrumb navigation indicates the user is at the 'Launch an instance' step. The main content area is titled 'Launch an instance' with a 'Info' link. A descriptive text explains that Amazon EC2 allows creating virtual machines on the AWS Cloud. Below this, the 'Name and tags' section is visible, featuring a 'Name' input field containing 'ultimate-cicd-demo' and a 'Add additional tags' button.

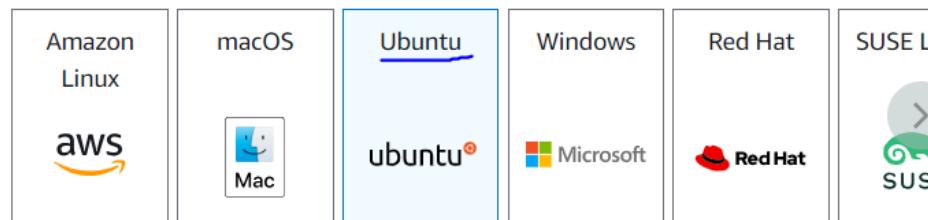


▼ Application and OS Images (Amazon Machine Image) [Info](#)

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

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

Quick Start



[Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

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

Instance type

t2.large

Family: t2 2 vCPU 8 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.1208 USD per Hour
On-Demand RHEL base pricing: 0.1528 USD per Hour
On-Demand SUSE base pricing: 0.1928 USD per Hour
On-Demand Linux base pricing: 0.0928 USD per Hour

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

Checkout below github repo url where you can find code for this project

Note: Its created by Abhishek veermalla - creator of this project)

<https://github.com/iam-veeramalla/Jenkins-Zero-To-Hero>

Step2:

1) Lets move to Jenkins part

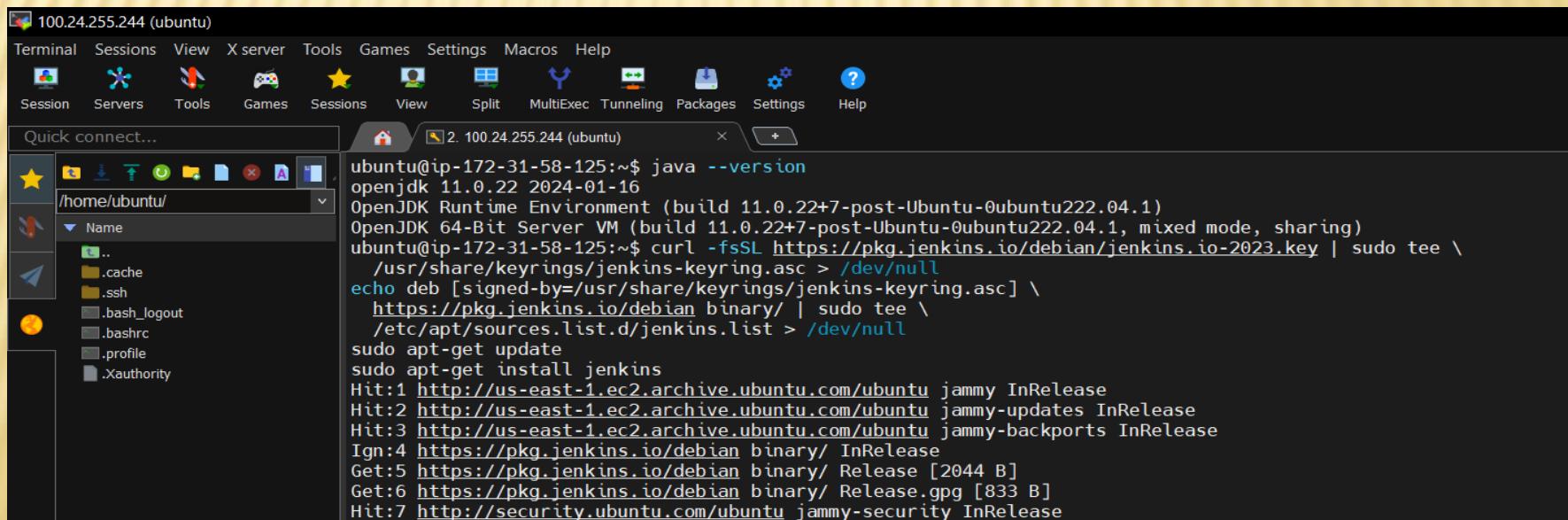
Login to EC2 instance using ssh

command: ssh -i /Downloads/sakeena-aws-pem-file.pem ubuntu@ 100.24.255.244 (public ip)

2) Install java as a prerequisite for Jenkins

sudo apt update

sudo apt install openjdk-11-jre



```
ubuntu@ip-172-31-58-125:~$ java --version
openjdk 11.0.22 2024-01-16
OpenJDK Runtime Environment (build 11.0.22+7-post-Ubuntu-0ubuntu222.04.1)
OpenJDK 64-Bit Server VM (build 11.0.22+7-post-Ubuntu-0ubuntu222.04.1, mixed mode, sharing)
ubuntu@ip-172-31-58-125:~$ curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \
    /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
    https://pkg.jenkins.io/debian binary/ | sudo tee \
    /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian binary/ InRelease
Get:5 https://pkg.jenkins.io/debian binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu jammy-security InRelease
```

- ❖ By default Jenkins server will be started by port 8080, when you try to access localhost 8080 it will not be started
so we have to configure the inbound traffic rules on EC2 instance(security rules which prevents the incoming traffic unless we enable the port for the EC2 instance)

The screenshot shows the AWS Management Console interface for managing security group inbound rules. The top navigation bar includes the AWS logo, Services (dropdown), a search bar with placeholder 'Search' and keyboard shortcut '[Alt+S]', and various status icons. The region is set to N. Virginia, and the character encoding is Ansi.

The current path in the breadcrumb navigation is: EC2 > Security Groups > sg-0c0d250ecb11885cc - launch-wizard-1 > Edit inbound rules.

Edit inbound rules

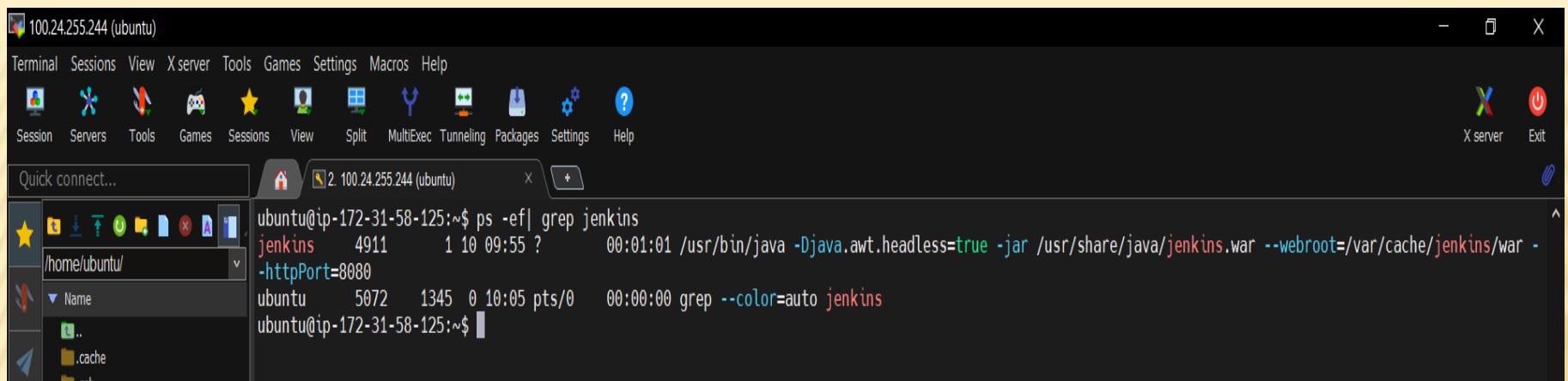
Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0d82861e0c0675cbc	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0/0 (X)
-	All traffic	All	All	Anyw...	<input type="text"/> 0.0.0.0/0 (X)

Add rule

we have selected All traffic here ,but in real time environment restrict it as per your requirement.

To check if Jenkins is running use below
ps -ef | grep Jenkins



Login to Jenkins using your ec2 instance ip address
eg: <http://<instance ip address>:8080/>

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`

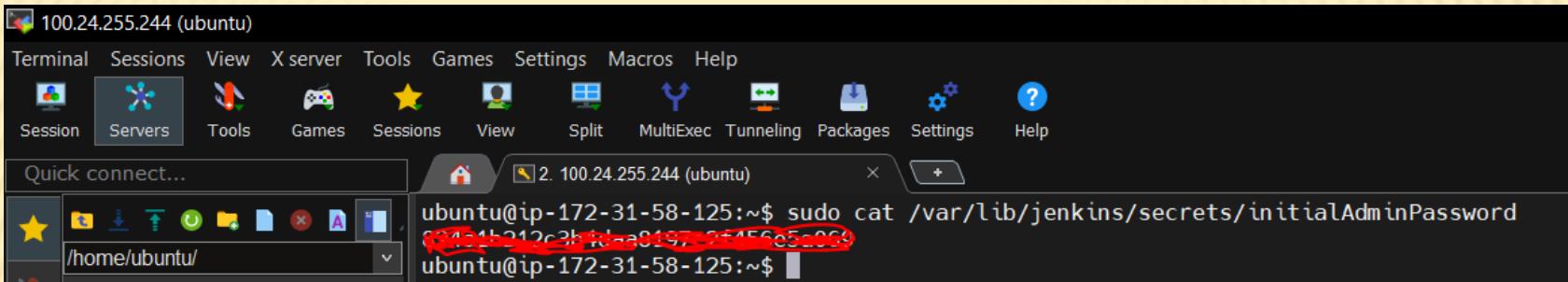
Please copy the password from either location and paste it below.

Administrator password

Continue

Type here to search ENG 1543 37°C Sunny 24-03-2024

Cat the password as below or you can set username and password by your own



A screenshot of a terminal window titled "100.24.255.244 (ubuntu)". The window has a dark theme with white text. At the top, there is a menu bar with options: Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, and Help. Below the menu bar is a toolbar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. The main area of the terminal shows a command line session. The user has run the command "sudo cat /var/lib/jenkins/secrets/initialAdminPassword". The output of this command is visible, though some characters are redacted with a large red marker. The path "/home/ubuntu/" is also visible at the bottom of the terminal window.

- ❖ Login to Jenkins using the password you got in above command
now you are ready to use the Jenkins
- ❖ Now click on new item and select pipeline and click on ok
- ❖ Jenkins allows you to write your code in two ways , you can write under script section in Jenkins or
you can put your Jenkins file in git repo where your source code is available
- ❖ Usually in real time, organizations will manage the Jenkins file in same folder where your
application code is available i.e., in Git repo



Search (CTRL+K)



Dashboard > All >

Enter an item name

ultimate-demo

» Required field



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



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

branch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

we have to give the Jenkins file path in your Jenkins configuration as below

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

<https://github.com/sakeena19/Jenkins-Zero-To-Hero>



Credentials ?



Branch Specifier (blank for 'any') ?

*/main



Add Branch

Repository browser ?

(Auto)



Additional Behaviours

Add ▾

Script Path ?

java-maven-sonar-argocd-helm-k8s/spring-boot-app/JenkinsFile

Lightweight checkout ?

Pipeline Syntax

Save

Apply



The purpose of Jenkins file is to execute all the other tasks of the continuous integration.

➤ We now need to install the Docker Pipeline plugin, which already includes Maven, located at the following path

Dashboard -> manage Jenkins -> Plugins -> Available plugins -> Docker pipeline -> install without restart

➤ To install sonar server, install sonar service under below path

Dashboard -> manage Jenkins -> Plugins -> Available Plugins -> SonarQubeScanner -> install without restart

➤ To install sonar sever on ec2 instance

use apt install unzip to install unzip package before downloading sonarqube

sudo su - sonarqube -> to get in to sonarqube folder

unzip * -> To extract sonarqube folder

Checkout below Readme file where you have steps to install sonarqube

<https://github.com/iam-veeramalla/Jenkins-Zero-To-Hero/blob/main/java-maven-sonar-argocd-helm-k8s/spring-boot-app/README.md>

By default sonar will be started on port 9000

100.24.255.244 (ubuntu)

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

2 100.24.255.244 (ubuntu)

```
ubuntu@ip-172-31-58-125:~$ sudo su -
root@ip-172-31-58-125:~# adduser sonarqube
Adding user `sonarqube' ...
Adding new group `sonarqube' (1001) ...
Creating home directory /home/sonarqube' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for sonarqube
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []

Is the information correct? [Y/n] y
root@ip-172-31-58-125:~# sudo su - sonarqube
sonarqube@ip-172-31-58-125:~$ wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.4.0.54424.zip
unzip *
--2024-03-24 11:02:34-- https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.4.0.54424.zip
Resolving binaries.sonarsource.com (binaries.sonarsource.com)... 99.84.191.75, 99.84.191.87, 99.84.191.23, ...
Connecting to binaries.sonarsource.com (binaries.sonarsource.com)|99.84.191.75|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 287666040 (274Mi) [binary/octet-stream]
Saving to: 'sonarqube-9.4.0.54424.zip'

sonarqube-9.4.0.54424.zip          100%[=====] 274.34M  88.0MB/s    in 3.1s

2024-03-24 11:02:37 (88.0 MB/s) - 'sonarqube-9.4.0.54424.zip' saved [287666040/287666040]

Command 'unzip' not found, but can be installed with:
apt install unzip
Please ask your administrator.
sonarqube@ip-172-31-58-125:~$
```

100.24.255.244 (ubuntu)

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

2 100.24.255.244 (ubuntu)

```
sonarqube@ip-172-31-58-125:~$ cd sonarqube-9.4.0.54424/
sonarqube@ip-172-31-58-125:~/sonarqube-9.4.0.54424$ cd ../
sonarqube@ip-172-31-58-125:~$ chmod -R 755 /home/sonarqube/sonarqube-9.4.0.54424
sonarqube@ip-172-31-58-125:~$ chown -R sonarqube:sonarqube /home/sonarqube/sonarqube-9.4.0.54424
sonarqube@ip-172-31-58-125:~$ cd sonarqube-9.4.0.54424/
sonarqube@ip-172-31-58-125:~/sonarqube-9.4.0.54424$ ls
COPYING bin conf data dependency-license.json elasticsearch extensions lib logs temp web
sonarqube@ip-172-31-58-125:~/sonarqube-9.4.0.54424$ cd bin
sonarqube@ip-172-31-58-125:~/sonarqube-9.4.0.54424/bin$ ls
jsw-license linux-x86-64 macosx-universal-64 windows-x86-64
sonarqube@ip-172-31-58-125:~/sonarqube-9.4.0.54424/bin$ cd linux-x86-64/
sonarqube@ip-172-31-58-125:~/sonarqube-9.4.0.54424/bin/linux-x86-64$ ./sonar.sh start
Starting SonarQube...
Started SonarQube.
sonarqube@ip-172-31-58-125:~/sonarqube-9.4.0.54424/bin/linux-x86-64$
```

➤ To Authenticate Jenkins with sonar follow below:

Go to sonarqube -> my account-> security ->give token name as Jenkins and generate token

The screenshot shows the SonarQube interface at the URL 100.24.255.244:9000/account/security/. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. A red banner at the top states: "You're running a version of SonarQube that is past end of life. Please upgrade to a supported version immediately." Below this, the user is identified as "Administrator". The main content area is titled "Tokens" and contains instructions about using tokens for security. A "Generate Tokens" section has a text input field for "Enter Token Name" and a "Generate" button. A success message box displays: "New token 'jenkins' has been created. Make sure you copy it now, you won't be able to see it again!" with a "Copy" button and a redacted token value. A table lists the token details:

Name	Last use	Created	Action
jenkins	Never	March 24, 2024	Revoke

Copy the token, go to Jenkins -> click on Manage Jenkins ->credentials ->system ->global credentials -> add credentials -> add sonarqube authentication here

Not secure 100.24.255.244:8080/manage/credentials/store/system/domain/_/newCredentials

Jenkins

Search (CTRL+K) ? 🔍

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind

Secret text

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

Secret
.....

ID ?
sonarqube

Description ?

Create

This screenshot shows the Jenkins 'New credentials' configuration page. The 'Kind' field is set to 'Secret text'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Secret' field contains several redacted dots. The 'ID' field is filled with 'sonarqube'. The 'Description' field is empty. At the bottom left is a blue 'Create' button.

☐ Install Docker on your ec2-instance

command: sudo apt install docker.io

Grant permissions to all users like Jenkins, ubuntu etc to avoid any permission issues

command: usermod -aG docker jenkins
usermod -aG docker ubuntu
systemctl restart docker -> to restart docker

- Go back and restart your jenkins(its a good practice to restart jenkins when any plugin is installed or any activity done from our end)

eg: <http://100.24.255.244:8080/restart>

- Now We have to configure docker credentials and github credentials inside jenkins
- To configure dockerhub cred:

Dashboard=>Manage jenkins -> Credentials -> system-> global credentials->adduser

Username will be your dockerhub username

ID is based on the name we have provided in the jenkins file as per the jenkins credential store

- To configure github cred:

To get access key -> go to your github account -> settings -> developer settings ->personal access tokens->Tokens(classic) ->generate new token ->generate new token(classic)-> copy token to your jenkins dashboard in same path(Dashboard=>Manage jenkins -> Credentials -> system-> global credentials)

Not secure 100.24.255.244:8080/manage/credentials/store/system/domain/_/newCredentials

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind

Username with password

Scope ?

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

Username ?

sakeena19

Treat username as secret ?

Password ?

.....

ID ?

docker-cred

Description ?

Create

Jenkins

Search (CTRL+K)

admin log out

Not secure 100.24.255.244:8080/manage/credentials/store/system/domain/_/newCredentials

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind

Secret text

Scope ?

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

Secret

.....

ID ?

github

Description ?

Create

Windows Taskbar

Type here to search

System tray icons

36°C Partly sunny

18:11

ENG

24-03-2024

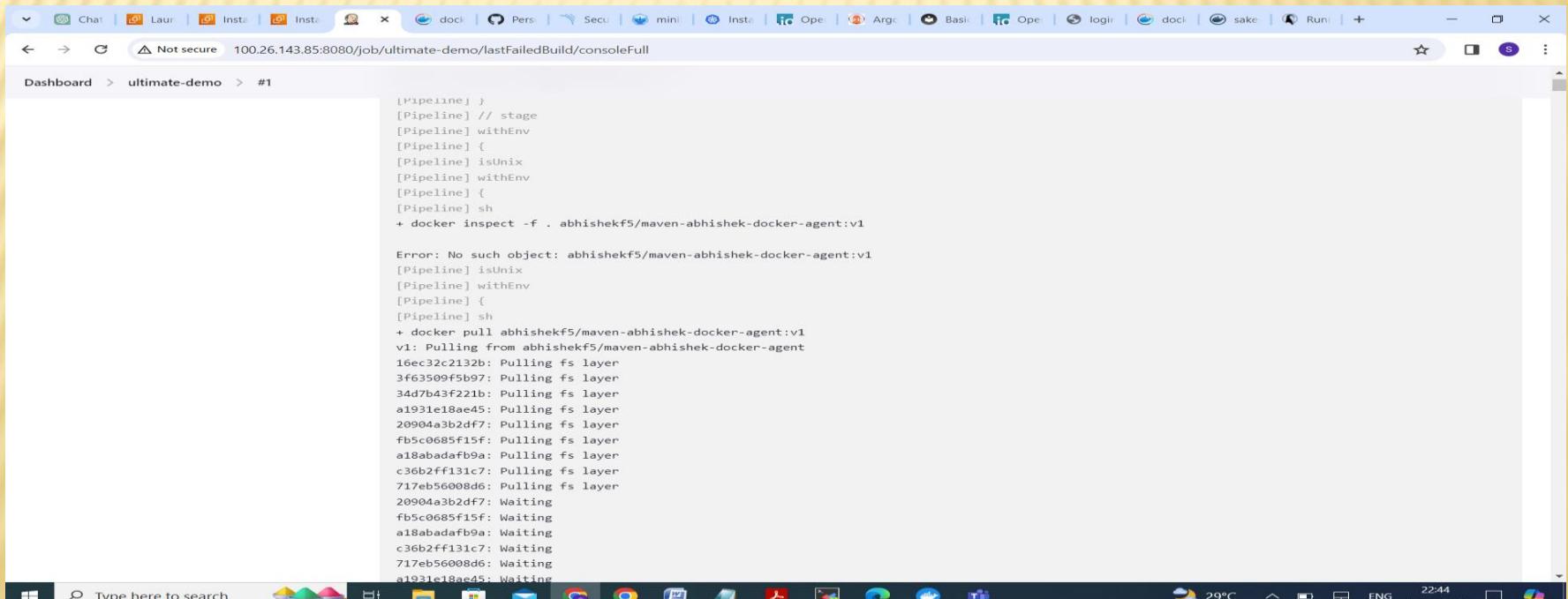
Now restart jenkins as we did above configuration to get the changes reflected

Run build and check the console output :

Things we automated via Jenkins like below:

- Create Container
- Checkout SCM
- Should push dockerhub image
- Should create sonarqube report
- Should update the image on dockerhub with build number automatically (shell script to update the manifest folder)

Stage1: Creating container, Jenkins has to start creating a container and executing the container



The screenshot shows a browser window displaying the Jenkins console output for a failed build. The URL in the address bar is `100.26.143.85:8080/job/ultimate-demo/lastFailedBuild/consoleFull`. The page title is "Dashboard > ultimate-demo > #1". The console output shows a Jenkins pipeline script attempting to inspect and pull a Docker image, but failing due to a missing object error.

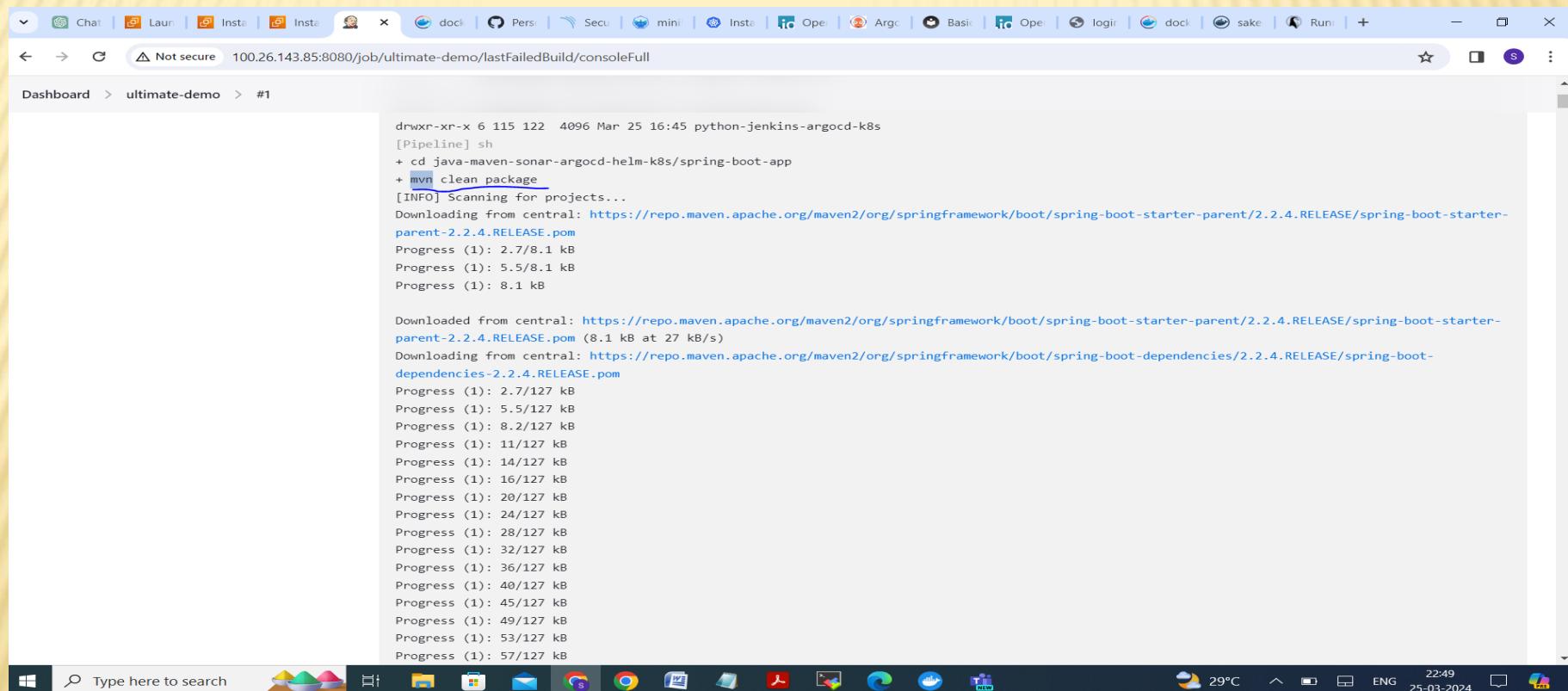
```
[Pipeline]
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker inspect -f . abhishekf5/maven-abhishek-docker-agent:v1

Error: No such object: abhishekf5/maven-abhishek-docker-agent:v1
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker pull abhishekf5/maven-abhishek-docker-agent:v1
v1: Pulling from abhishekf5/maven-abhishek-docker-agent
16ec32c2132b: Pulling fs layer
3f63509f5fb9: Pulling fs layer
34d7b43f221b: Pulling fs layer
a1931e18ae45: Pulling fs layer
20904a3b2df7: Pulling fs layer
fb5c0685f15f: Pulling fs layer
a18abadaafb9a: Pulling fs layer
c36b2ff131c7: Pulling fs layer
717eb56008d6: Pulling fs layer
20904a3b2df7: Waiting
fb5c0685f15f: Waiting
a18abadaafb9a: Waiting
c36b2ff131c7: Waiting
717eb56008d6: Waiting
a1931e18ae45: Waiting
```

Stage 2:

Maven will create the jar file

here maven target(mvn clean package) is downloading a lot of packages and its dependencies and it will create a jar file, This step is important as all other steps based on creation of this jarfile



The screenshot shows a browser window displaying the Jenkins console output for a job named "ultimate-demo". The URL is 100.26.143.85:8080/job/ultimate-demo/lastFailedBuild/consoleFull. The console output shows the execution of a shell script (drwxr-xr-x 6 115 122 4096 Mar 25 16:45 python-jenkins-argocd-k8s). It then navigates to a directory (cd java-maven-sensor-argocd-helm-k8s/spring-boot-app) and runs the command mvn clean package. The output indicates that Maven is scanning for projects and then downloading various Spring Boot dependencies from the central repository. The progress bar shows the download of the spring-boot-starter-parent-2.2.4.RELEASE.pom file at 27 kB/s. The Jenkins interface includes a dashboard, navigation links, and a status bar at the bottom.

```
drwxr-xr-x 6 115 122 4096 Mar 25 16:45 python-jenkins-argocd-k8s
[Pipeline] sh
+ cd java-maven-sensor-argocd-helm-k8s/spring-boot-app
+ mvn clean package
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.2.4.RELEASE/spring-boot-starter-parent-2.2.4.RELEASE.pom
Progress (1): 2.7/8.1 kB
Progress (1): 5.5/8.1 kB
Progress (1): 8.1 kB

Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.2.4.RELEASE/spring-boot-starter-parent-2.2.4.RELEASE.pom (8.1 kB at 27 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.2.4.RELEASE/spring-boot-dependencies-2.2.4.RELEASE.pom
Progress (1): 2.7/127 kB
Progress (1): 5.5/127 kB
Progress (1): 8.2/127 kB
Progress (1): 11/127 kB
Progress (1): 14/127 kB
Progress (1): 16/127 kB
Progress (1): 20/127 kB
Progress (1): 24/127 kB
Progress (1): 28/127 kB
Progress (1): 32/127 kB
Progress (1): 36/127 kB
Progress (1): 40/127 kB
Progress (1): 45/127 kB
Progress (1): 49/127 kB
Progress (1): 53/127 kB
Progress (1): 57/127 kB
```

Not secure 100.26.143.85:8080/job/ultimate-demo/lastFailedBuild/consoleFull

Dashboard > ultimate-demo > #1

```
demo| 9/31 | ^ | x
```

Downloading from central: <https://repo.maven.apache.org/maven2/org/springframework/security/spring-security-bom/5.2.1.RELEASE/pom>
Progress (1): 4.1/5.4 kB
Progress (1): 5.4 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/org/springframework/security/spring-security-bom/5.2.1.RELEASE/pom> (5.4 kB at 319 kB/s)
Downloading from central: <https://repo.maven.apache.org/maven2/org/springframework/session/spring-session-bom/Corn-RELEASE/pom>
Progress (1): 2.8 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/org/springframework/session/spring-session-bom/Corn-RELEASE/pom> (2.8 kB at 252 kB/s)
[INFO]
[INFO] -----< com.abhishek:spring-boot-demo >-----
[INFO] Building spring-boot-demo 1.0
[INFO] -----[jar]-----
Downloading from central: <https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-maven-plugin/2.2.4.RELEASE/spring-boot-maven-plugin-2.2.4.RELEASE/pom>
Progress (1): 4.1/5.0 kB
Progress (1): 5.0 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-maven-plugin/2.2.4.RELEASE/spring-boot-maven-plugin-2.2.4.RELEASE/pom> (5.0 kB at 414 kB/s)
Downloading from central: <https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-tools/2.2.4.RELEASE/spring-boot-tools-2.2.4.RELEASE/pom>
Progress (1): 1.8 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-tools/2.2.4.RELEASE/spring-boot-tools-2.2.4.RELEASE/pom> (1.8 kB at 176 kB/s)
Downloading from central: <https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-parent/2.2.4.RELEASE/spring-boot-parent-2.2.4.RELEASE/pom>

Type here to search

Hig... 22:50 25-03-2024 ENG

Stage 3: mvn sonar:sonar will be executed and pushes the report to sonarqube

Not secure 100.26.143.85:8080/job/ultimate-demo/lastFailedBuild/consoleFull

Dashboard > ultimate-demo > #1

```
[Pipeline] sh
+ cd java-maven-sonar-argocd-helm-k8s/spring-boot-app
+ mvn sonar:sonar -Dsonar.login=**** -Dsonar.host.url=http://100.26.143.85:9000
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.5.2/maven-install-plugin-2.5.2.pom
Progress (1): 2.7/6.4 kB
Progress (1): 5.5/6.4 kB
Progress (1): 6.4 kB

Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.5.2/maven-install-plugin-2.5.2.pom (6.4 kB at 25 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/25/maven-plugins-25.pom
Progress (1): 2.7/9.6 kB
Progress (1): 5.5/9.6 kB
Progress (1): 8.2/9.6 kB
Progress (1): 9.6 kB

Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/25/maven-plugins-25.pom (9.6 kB at 637 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/24/maven-parent-24.pom
Progress (1): 2.7/37 kB
Progress (1): 5.5/37 kB
Progress (1): 8.2/37 kB
Progress (1): 11/37 kB
Progress (1): 14/37 kB
Progress (1): 16/37 kB
Progress (1): 19/37 kB
Progress (1): 22/37 kB
Progress (1): 25/37 kB
```

Not secure 100.26.143.85:9000/projects

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

You're running a version of SonarQube that is past end of life. Please upgrade to a supported version immediately. [Learn More](#)

My Favorites All

Search by project name or key

Create Project

1 projects

Perspective: Overall Status Sort by: Name

spring-boot-demo Passed

Last analysis: 31 minutes ago

Bugs: 0 A Vulnerabilities: 0 A Hotspots Reviewed: - A Code Smells: 0 A Coverage: 0.0% Duplications: 0.0% Lines: 79 XML, Java

Filters

Quality Gate: Passed (1) Failed (0)

Bugs: A (1) B (0) C (0) D (0) E (0)

Vulnerabilities: A (1) B (0) C (0) D (0) E (0)

Security: A (1) B (0) C (0) D (0) E (0)

Security Review: A (≥ 80%) (1) B (70% - 80%) (0) C (50% - 70%) (0)

1 of 1 shown

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 9.4 (build 54424) - LGPL v3 - Community - Documentation - Plugins - Web API

29°C ENG 25-03-2024 22:54

Stage 4:

Docker image has to be created which will be configured in your jenkins file with the same name, image should get created on docker hub

The screenshot shows a browser window displaying the Docker Hub interface. The URL in the address bar is `hub.docker.com/r/sakeena19/ultimate-cicd/tags`. The page title is `sakeena19/ultimate-cicd`. The repository was created by [sakeena19](#) and was updated 1 minute ago.

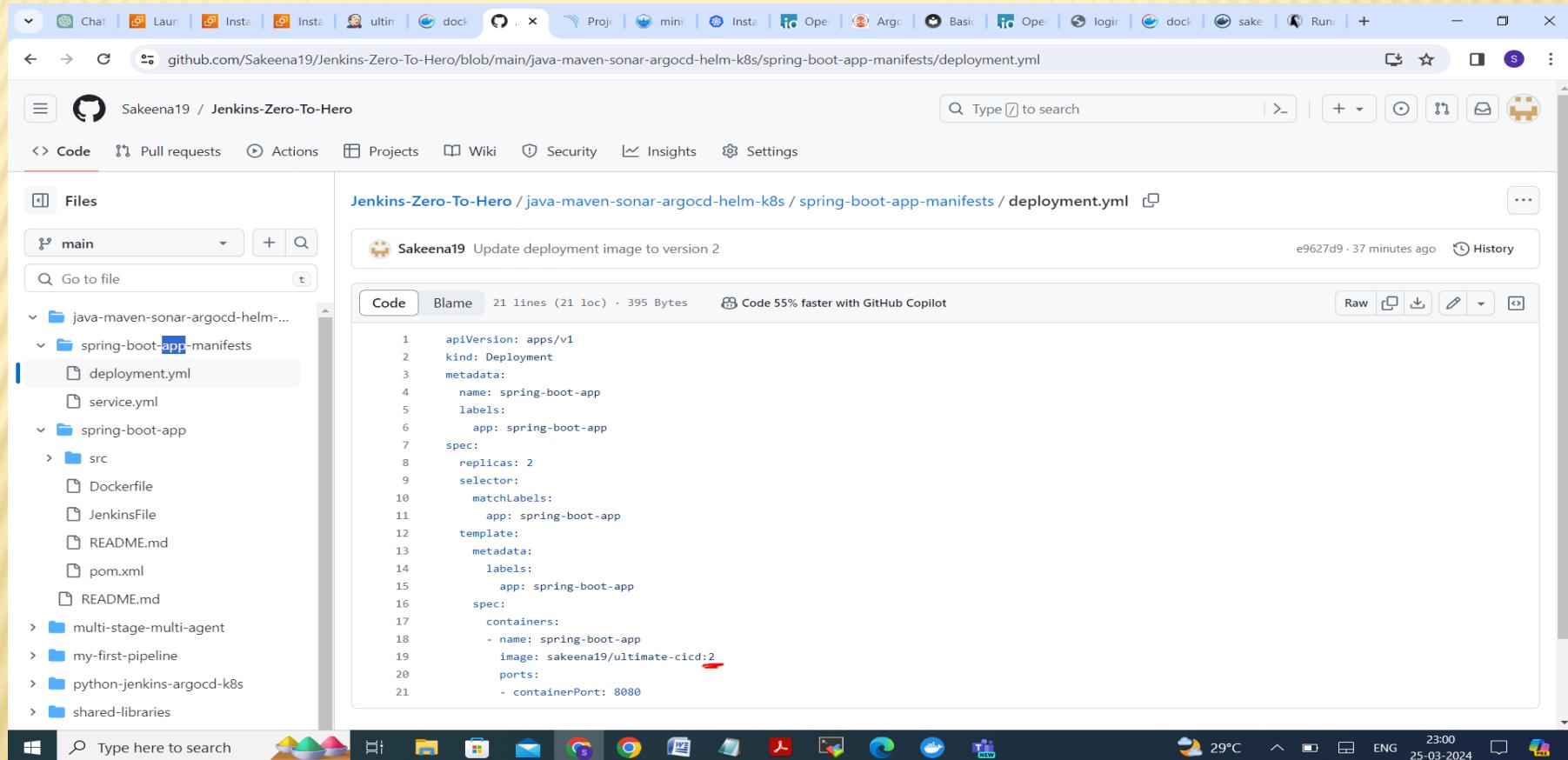
The main content area displays two tags for the repository:

- Tag: 2**
Last pushed a minute ago by [sakeena19](#)
Digest: [48ce8016499f](#) OS/ARCH: linux/amd64 Last pull: --- Compressed Size: 67.31 MB
- Tag: 8**
Last pushed 12 hours ago by [sakeena19](#)
Digest: [48ce8016499f](#) OS/ARCH: linux/amd64 Last pull: --- Compressed Size: 67.31 MB

At the bottom of the screen, the Windows taskbar is visible, showing the Start button, a search bar with the text "Type here to search", and various pinned icons for apps like File Explorer, Edge, and File History. The system tray shows the date (25-03-2024), time (22:24), battery level (29%), and language (ENG).

Stage 5:

Final step using shell script to update the manifest folder



The screenshot shows a GitHub repository page for "Jenkins-Zero-To-Hero". The left sidebar displays a file tree with several folders and files, including "main", "java-maven-sonar-argocd-helm-k8s", "spring-boot-app-manifests", "deployment.yaml", "service.yml", "spring-boot-app", "src", "Dockerfile", "JenkinsFile", "README.md", "pom.xml", "README.md", "multi-stage-multi-agent", "my-first-pipeline", "python-jenkins-argocd-k8s", and "shared-libraries". The main content area shows the "deployment.yaml" file with the following content:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: spring-boot-app
  labels:
    app: spring-boot-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: spring-boot-app
  template:
    metadata:
      labels:
        app: spring-boot-app
    spec:
      containers:
        - name: spring-boot-app
          image: sakeena19/ultimate-cicd:2
          ports:
            - containerPort: 8080
```

A commit by "Sakeena19" titled "Update deployment image to version 2" is visible, with the commit ID "e9627d9" and a timestamp of "37 minutes ago". A GitHub Copilot note indicates "Code 55% faster with GitHub Copilot". The GitHub interface includes a search bar, navigation icons, and a Windows taskbar at the bottom.

Lets move on to CD part now,

To install kubectl follow below documentation as per your OS

<https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>

```
ubuntu@ip-172-31-58-125:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload   Total Spent   Left  Speed
100 138 100 138  0     0  1378      0 --::-- --::-- --::-- 1380
100 47.4M 100 47.4M  0     0  85.5M      0 --::-- --::-- --::-- 133M
ubuntu@ip-172-31-58-125:~$ echo "$(cat kubectl.sha256)" kubectl | sha256sum --check
cat: kubectl.sha256: No such file or directory
sha256sum: 'standard input': no properly formatted SHA256 checksum lines found
ubuntu@ip-172-31-58-125:~$ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
ubuntu@ip-172-31-58-125:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload   Total Spent   Left  Speed
100 138 100 138  0     0  1330      0 --::-- --::-- --::-- 1339
100  64 100  64  0     0  374      0 --::-- --::-- --::-- 374
ubuntu@ip-172-31-58-125:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload   Total Spent   Left  Speed
100 138 100 138  0     0  1417      0 --::-- --::-- --::-- 1422
100  64 100  64  0     0  390      0 --::-- --::-- --::-- 390
ubuntu@ip-172-31-58-125:~$ echo "$(cat kubectl.sha256)" kubectl | sha256sum --check
kubectl: OK
ubuntu@ip-172-31-58-125:~$ █
```

To install minikube use below doc

<https://minikube.sigs.k8s.io/docs/start/>

Note: follow steps and documentation based in your OS

The screenshot shows a MobaXterm window with two terminal sessions. Session 1 (tab 1) is titled '100.24.255.244 (ubuntu)' and shows the command `curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64` being run. Session 2 (tab 2) is titled '7. 100.24.255.244 (ubuntu)' and shows the output of the curl command, which includes a progress bar for file download and a warning about Docker space usage.

```
ubuntu@ip-172-31-58-125:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time      Time     Current
          Dload  Upload Total Spent   Left Speed
100 89.3M  100 89.3M    0     0  105M      0 --:--:-- --:--:-- --:--:-- 105M
ubuntu@ip-172-31-58-125:~$ minikube start
* minikube v1.32.0 on Ubuntu 22.04 (xen/amd64)
* Automatically selected the docker driver. Other choices: ssh, none
* Using Docker driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.28.3 preload ...
  > preloaded-images-k8s-v18-v1...: 403.35 MiB / 403.35 MiB 100.00% 69.03 M
  > gcr.io/k8s-minikube/kicbase...: 453.88 MiB / 453.90 MiB 100.00% 61.43 M
* Creating docker container (CPUs=2, Memory=2200MB) ...
X Docker is nearly out of disk space, which may cause deployments to fail! (90% of capacity). You can pass '--force' to skip this check.
* Suggestion:

  Try one or more of the following to free up space on the device:
  1. Run "docker system prune" to remove unused Docker data (optionally with "-a")
  2. Increase the storage allocated to Docker for Desktop by clicking on:
    Docker icon > Preferences > Resources > Disk Image Size
  3. Run "minikube ssh -- docker system prune" if using the Docker container runtime
* Related issue: https://github.com/kubernetes/minikube/issues/9024

* Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
- Generating certificates and keys ...
- Booting up control plane ...
- Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
ubuntu@ip-172-31-58-125:~$
```

Below the terminals, the MobaXterm status bar displays 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>'. The Windows taskbar at the bottom shows various pinned icons and the system tray with the date and time.

To install argo CD using operators(which will manage lifecycle of kubernetes controllers)
<https://operatorhub.io/operator/argocd-operator>

Run all commands sequentially as mentioned in doc to installed argocd

OperatorHub.io



Argo CD

0.8.0 provided by Argo CD Community

Install on Kubernetes

1. Install Operator Lifecycle Manager (OLM), a tool to help manage the Operators running on your cluster.

```
$ curl -sL https://github.com/operator-framework/operator-lifecycle-manager/releases/download/v0.27.0/install.sh | bash -s v0.27.0
```

2. Install the operator by running the following command:

```
$ kubectl create -f https://operatorhub.io/install/argocd-operator.yaml
```

[What happens when I execute this command?](#)

This Operator will be installed in the "operators" namespace and will be usable from all namespaces in the cluster.

3. After install, watch your operator come up using next command.

```
$ kubectl get csv -n operators
```

To use it, checkout the custom resource definitions (CRDs) introduced by this operator to start using it.

100.24.255.244 (ubuntu)

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Sessions View Split Multitab Tunneling Packages Settings Help

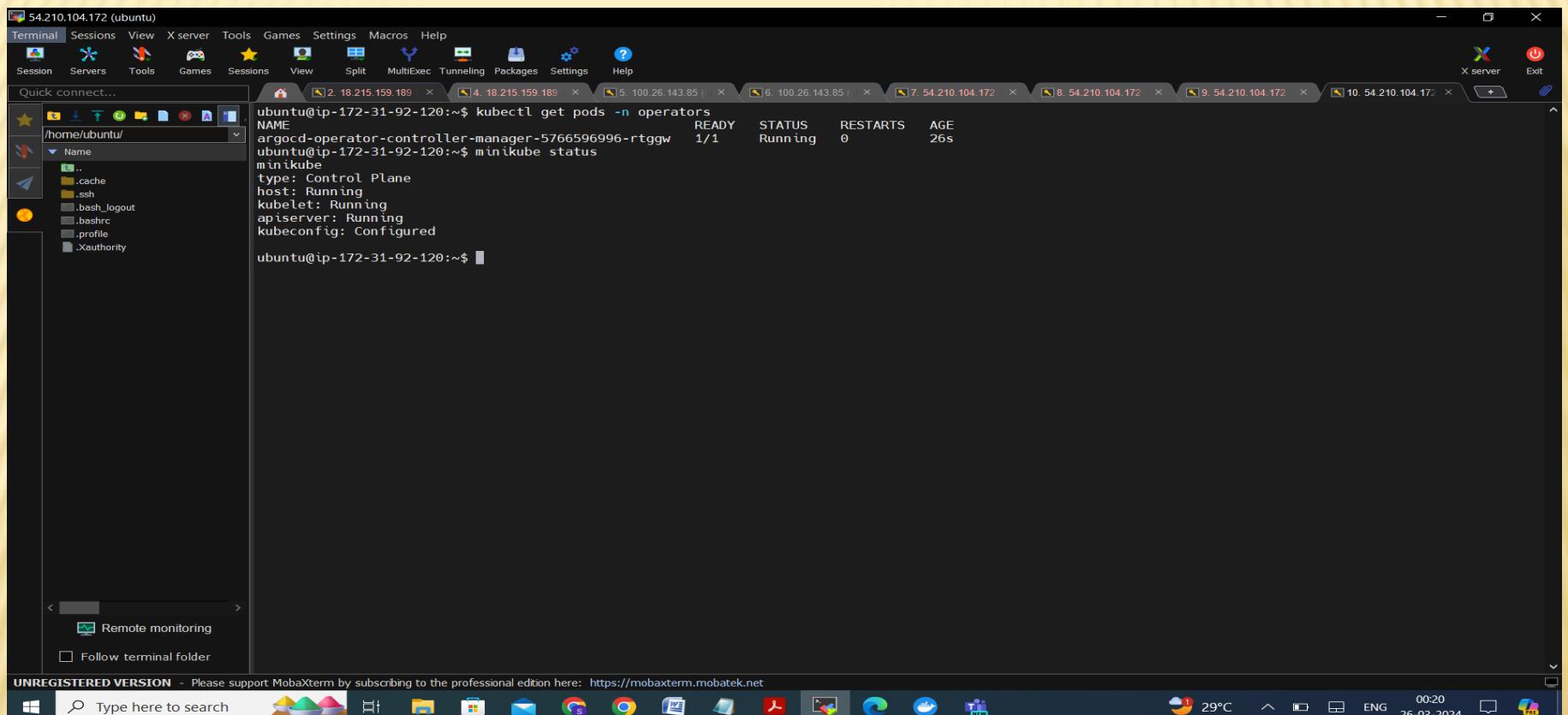
Quick connect...

ubuntu@ip-172-31-58-125:~\$ curl -sL https://github.com/operator-framework/operator-lifecycle-manager/releases/download/v0.27.0/install.sh | bash -s v0.27.0

```
customresourcedefinition.apiextensions.k8s.io/catalogsources.coreos.com created
customresourcedefinition.apiextensions.k8s.io/clusterserviceversions.coreos.com created
customresourcedefinition.apiextensions.k8s.io/installplans.coreos.com created
customresourcedefinition.apiextensions.k8s.io/operators.coreos.com created
customresourcedefinition.apiextensions.k8s.io/operatorconditions.coreos.com created
customresourcedefinition.apiextensions.k8s.io/operatorgroups.coreos.com created
customresourcedefinition.apiextensions.k8s.io/operators.coreos.com created
customresourcedefinition.apiextensions.k8s.io/operators.coreos.com created
customresourcedefinition.apiextensions.k8s.io/subscriptions.coreos.com created
customresourcedefinition.apiextensions.k8s.io/subscriptions.coreos.com condition met
customresourcedefinition.apiextensions.k8s.io/clusterserviceversions.coreos.com condition met
customresourcedefinition.apiextensions.k8s.io/installplans.coreos.com condition met
customresourcedefinition.apiextensions.k8s.io/operators.coreos.com condition met
customresourcedefinition.apiextensions.k8s.io/operatorconditions.coreos.com condition met
customresourcedefinition.apiextensions.k8s.io/operatorgroups.coreos.com condition met
customresourcedefinition.apiextensions.k8s.io/operators.coreos.com condition met
customresourcedefinition.apiextensions.k8s.io/subscriptions.coreos.com condition met
namespace/olm created
namespace/operators created
serviceaccount/operator-serviceaccount created
clusterrole.rbac.authorization.k8s.io/system:controller:operator-lifecycle-manager created
clusterrolebinding.rbac.authorization.k8s.io/olm-operator-binding-olm created
olmconfig.operators.coreos.com/custer created
deployment.apps/catalog-operator created
deployment.apps/catalog-operator-edit created
clusterrole.rbac.authorization.k8s.io/aggregate-olm-edit created
clusterrole.rbac.authorization.k8s.io/aggregate-olm-view created
operatorgroup.operators.coreos.com/global-operators created
operatorgroup.operators.coreos.com/operators created
clusterserviceversion.operators.coreos.com/packageserver created
catalogsource.operators.coreos.com/operatorhubio-catalog created
Waiting for deployment "olm-operator" rollout to finish: 0 of 1 updated replicas are available...
deployment "olm-operator" successfully rolled out
catalog-operator successfully rolled out
Package server phase: Installing
Package server phase: Succeeded
deployment "packageserver" successfully rolled out
ubuntu@ip-172-31-58-125:~$
```

Remote monitoring

To check whether your Argocd operator is installed use below command
kubectl get pods -n operators
minikube status



The screenshot shows a MobaXterm window titled "54.210.104.172 (ubuntu)". It has multiple tabs at the top, including "Terminal", "Sessions", "View", "X server", "Tools", "Games", "Settings", "Macros", and "Help". Below the tabs are icons for "Session", "Servers", "Tools", "Games", "Sessions", "View", "Split", "MultiExec", "Tunneling", "Packages", "Settings", and "Help". On the right side of the window, there are icons for "X server" and "Exit".

The main terminal window displays the following command output:

```
ubuntu@ip-172-31-92-120:~$ kubectl get pods -n operators
NAME                           READY   STATUS    RESTARTS   AGE
argocd-operator-controller-manager-5766596996-rtggw   1/1     Running   0          26s

ubuntu@ip-172-31-92-120:~$ minikube status
minikube
  type: Control Plane
  host: Running
  kubelet: Running
  apiserver: Running
  kubeconfig: Configured

ubuntu@ip-172-31-92-120:~$
```


At the bottom of the terminal window, there is a message: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

The taskbar at the bottom of the screen includes icons for the Start button, search bar, File Explorer, Mail, Google Chrome, Microsoft Edge, File Manager, Task View, and a system tray with a battery icon, temperature (29°C), language (ENG), date (00:20 26-03-2024), and a clock icon.

Use below documentation to get started with argocd implementation

<https://argocd-operator.readthedocs.io/en/latest/usage/basics/>

❖ To create argocd controller , create vi file with below example(mentioned in above doc)

eg: apiVersion: argoproj.io/v1alpha1

kind: ArgoCD

metadata:

 name: example-argocd

 labels:

 example: basic

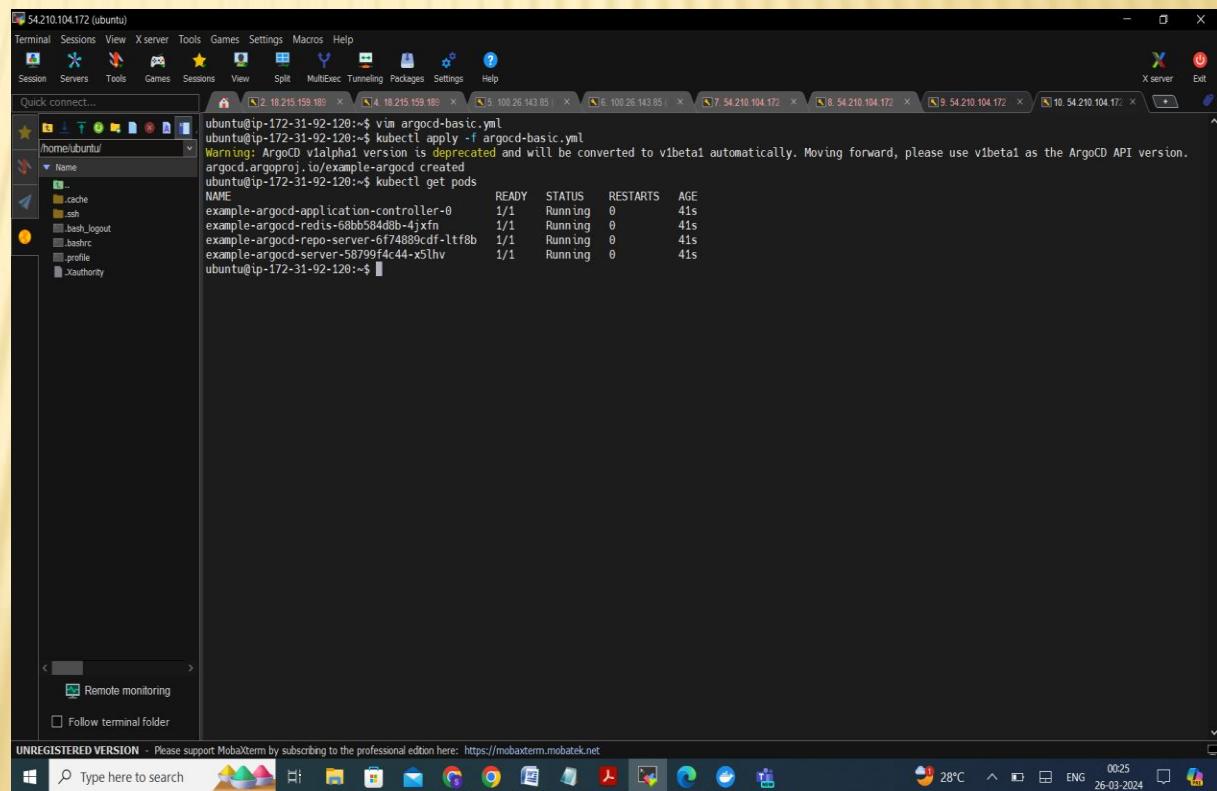
spec: {}

❖ Run below commands to get the argocd workloads being created

Commands:

kubectl apply -f argocd-basic.yml

kubectl get pods (to check status)



The screenshot shows a MobaXterm window titled '54.210.104.172 (ubuntu)'. The terminal session displays the following commands and output:

```
ubuntu@ip-172-31-92-120:~$ vim argocd-basic.yml
Warning: ArgoCD v1alpha1 version is deprecated and will be converted to v1beta1 automatically. Moving forward, please use v1beta1 as the ArgoCD API version.
argocd.argoproj.io/example-argocd created
ubuntu@ip-172-31-92-120:~$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
example-argocd-application-controller-0   1/1     Running   0          41s
example-argocd-redis-68bb584d8b-jxfn      1/1     Running   0          41s
example-argocd-repo-server-6f74889cdf-ltf8b 1/1     Running   0          41s
example-argocd-server-5b799f4c44-x5lhv     1/1     Running   0          41s
ubuntu@ip-172-31-92-120:~$
```

The terminal window also shows a sidebar with session history and a message at the bottom: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

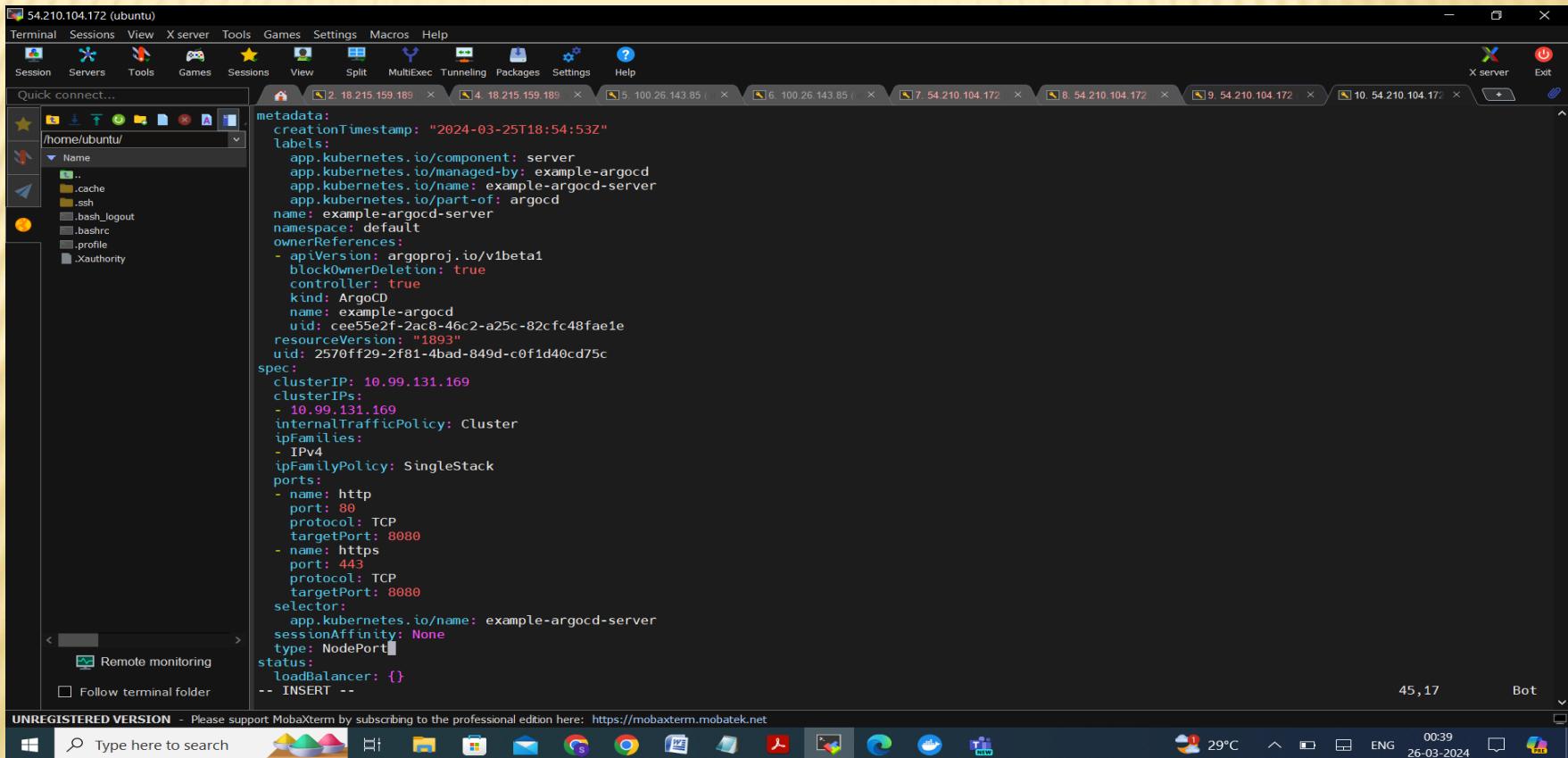
❖ Execute below command which is responsible to run the argocd ui on your browser

Kubectl get svc

kubectl edit svc example-argocd-server (to change the type from ClusterIP to NodePort)

❖ Below command can also be passed directly:

```
kubectl patch svc example-argocd-server -n argocd -p '{"spec": {"type": "NodePort"}}'
```



```
54.210.104.172 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
metadata:
  creationTimestamp: "2024-03-25T18:54:53Z"
labels:
  app.kubernetes.io/component: server
  app.kubernetes.io/managed-by: argocd
  app.kubernetes.io/name: example-argocd-server
  app.kubernetes.io/part-of: argocd
  name: example-argocd-server
  namespace: default
  ownerReferences:
    - apiVersion: argoproj.io/v1beta1
      blockOwnerDeletion: true
      controller: true
      kind: ArgocD
      name: argocd
      uid: cee5e2f-2ac8-46c2-a25c-82cf48fae1e
      resourceVersion: "1893"
      uid: 2570ff29-2f81-4bad-849d-c0f1d40cd75
spec:
  clusterIP: 10.99.131.169
  clusterIPs:
    - 10.99.131.169
  internalTrafficPolicy: Cluster
  ipFamilies:
    - IPv4
  ipFamilyPolicy: SingleStack
  ports:
    - name: http
      port: 80
      protocol: TCP
      targetPort: 8080
    - name: https
      port: 443
      protocol: TCP
      targetPort: 8080
  selector:
    app.kubernetes.io/name: example-argocd-server
  sessionAffinity: None
  type: NodePort
  status:
    loadBalancer: {}
-- INSERT --
45,17 Bot
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search 29°C ENG 00:39 26-03-2024
```

The screenshot shows a terminal window titled "54.210.104.172 (ubuntu)" running on a MobaXterm interface. The terminal displays the output of several commands related to Kubernetes services:

```
ubuntu@ip-172-31-92-120:~$ kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
example-argocd-metrics  NodePort  10.100.109.58 <none>       8082:30820/TCP  12m
example-argocd-redis   ClusterIP 10.109.114.15  <none>       6379/TCP     12m
example-argocd-repo-server  ClusterIP 10.97.240.134 <none>       8081/TCP,8084/TCP 12m
example-argocd-server   ClusterIP 10.99.131.169 <none>       80/TCP,443/TCP  12m
example-argocd-server-metrics  ClusterIP 10.109.16.141 <none>       8083/TCP     12m
kubernetes          ClusterIP  10.96.0.1    <none>       443/TCP      23m
ubuntu@ip-172-31-92-120:~$ kubectl edit svc example-argocd-server
service/example-argocd-server edited
ubuntu@ip-172-31-92-120:~$ kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
example-argocd-metrics  NodePort  10.100.109.58 <none>       8082:30820/TCP  14m
example-argocd-redis   ClusterIP 10.109.114.15  <none>       6379/TCP     14m
example-argocd-repo-server  ClusterIP 10.97.240.134 <none>       8081/TCP,8084/TCP 14m
example-argocd-server   NodePort  10.99.131.169 <none>       80:30688/TCP,443:32327/TCP 14m
example-argocd-server-metrics  ClusterIP 10.109.16.141 <none>       8083/TCP     14m
kubernetes          ClusterIP  10.96.0.1    <none>       443/TCP      26m
ubuntu@ip-172-31-92-120:~$
```

The terminal also shows a "Remote monitoring" icon and a "Follow terminal folder" checkbox at the bottom. The status bar at the bottom of the window indicates "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net".

To execute it on browser , use minikube feature by executing below command

minikube service list -> lists all the services exposed by minikube

minikube service example-argocd-server -> to expose specific service

The screenshot shows a MobaXterm window with multiple terminal sessions and a file explorer. The terminal sessions are:

- Session 1: 2.18.215.199.189
- Session 2: 4.18.215.199.189
- Session 3: 5.100.26.143.85
- Session 4: 6.100.26.143.85
- Session 5: 7.54.210.104.172
- Session 6: 8.54.210.104.172
- Session 7: 9.54.210.104.172
- Session 8: 10.54.210.104.172

The current terminal session (Session 1) displays the output of the command `minikube service list`, which lists various Kubernetes services and their URLs. It also shows the output of `kubectl get pods`, listing several pods in the default namespace.

NAMESPACE	NAME	TARGET PORT	URL
default	example-argocd-metrics	metrics/8882	http://192.168.49.2:30829
default	example-argocd-redis	No node port	
default	example-argocd-repo-server	No node port	
default	example-argocd-server	http/80 https/443	http://192.168.49.2:30688 http://192.168.49.2:32327
default	example-argocd-server-metrics	No node port	
default	kubernetes	No node port	
kube-system	kube-dns	No node port	
olm	operatorhubio-catalog	No node port	
olm	packageserver-service	No node port	
operators	argocd-operator-controller-manager-metrics-service	No node port	
operators	argocd-operator-controller-manager-service	No node port	
operators	argocd-operator-webhook-service	No node port	

ubuntu@ip-172-31-92-120:~\$ kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
example-argocd-application-controller-0	1/1	Running	0	20m
example-argocd-redis-68bb584d8b-4jxfn	1/1	Running	0	20m
example-argocd-repo-server-6f74889cdf-ltf8b	1/1	Running	0	20m
example-argocd-server-5879ff4c44-x5lhv	1/1	Running	0	20m

ubuntu@ip-172-31-92-120:~\$

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

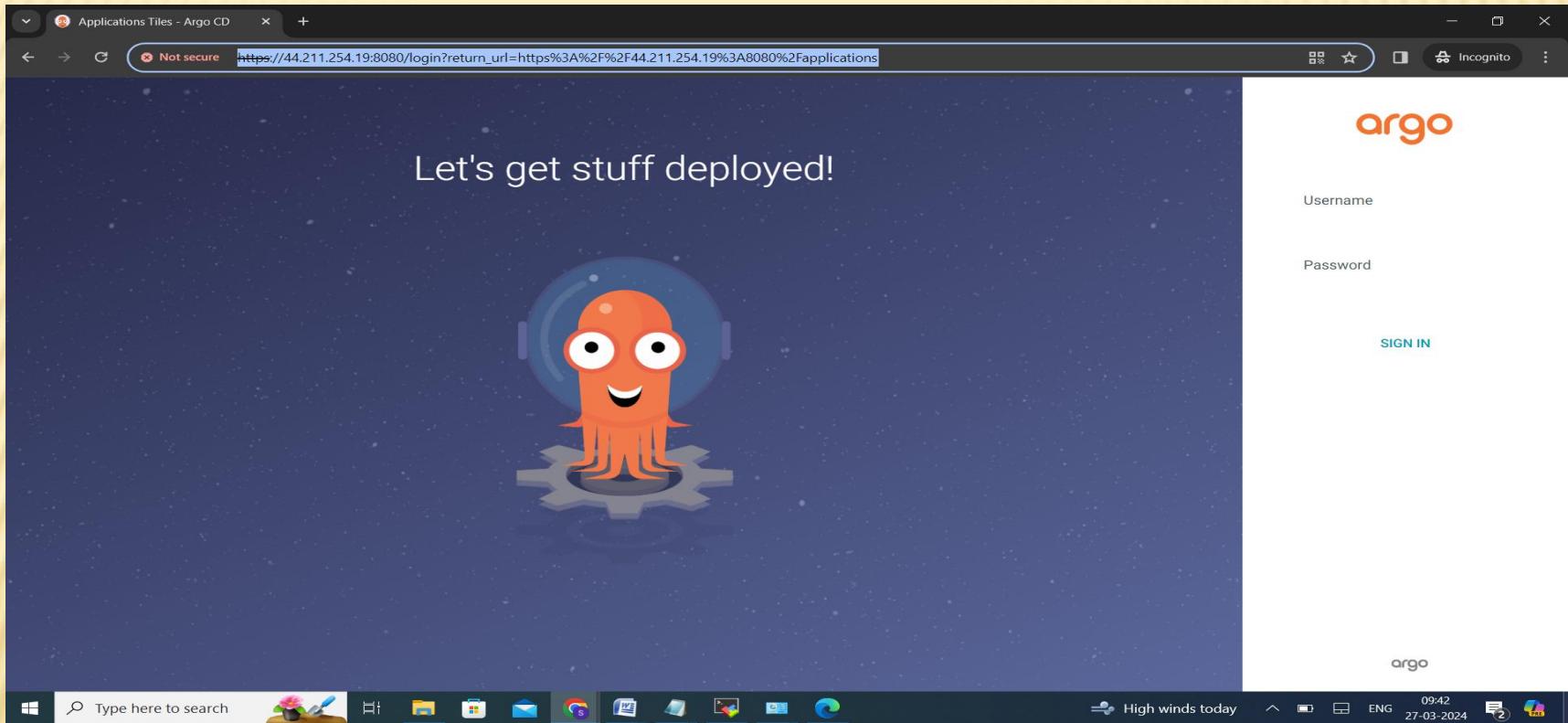
Windows taskbar icons include: Start, Search, File Explorer, Mail, Google Chrome, Microsoft Edge, File Manager, Paint 3D, Task View, Taskbar settings, and a system tray showing 29°C, ENG, 00:45, 26-03-2024.

In case if tunneling is not happening use below command

```
kubectl port-forward svc/example-argocd-server -n default --address 0.0.0.0 8080:443
```

Tunneling in Minikube enables access to Kubernetes cluster services from the host machine by forwarding traffic through a secure channel. It facilitates seamless interaction with applications and services running within the cluster for development and testing purposes

Now you should be able to login argocd ui on your browser



Username is admin by default

And for password , use below command

```
kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d  
; echo
```

Click on create application tab and start creating your first application😊

Applications Tiles - Argo CD

Not secure https://44.211.254.19:8080/applications?new=%7B"apiVersion"%3A"argoproj.io%2Fv1alpha1"%2C"kind"%3A"Application"%2C"metadata"%3A%7B"name"%3A"test"%7D%2C... ☆

Incognito

CREATE CANCEL

argo v2.10.4+5d63a5

Applications Settings User Info Documentation

Application + NEW

GENERAL EDIT AS YAML

Application Name: test

Project Name: default

SYNC POLICY: Automatic

PRUNE RESOURCES ⓘ
 SELF HEAL ⓘ
 SET DELETION FINALIZER ⓘ

SYNC OPTIONS

SKIP SCHEMA VALIDATION
 PRUNE LAST
 RESPECT IGNORE DIFFERENCES AUTO-CREATE NAMESPACE
 APPLY OUT OF SYNC ONLY
 SERVER-SIDE APPLY

PRUNE PROPAGATION POLICY: foreground

Type here to search

28°C Sunny 10:02 27-03-2024 ENG

Applications Tiles - Argo CD

Not secure https://44.211.254.19:8080/applications?new=%7B"apiVersion"%3A"argoproj.io%2Fv1alpha1"%2C"kind"%3A"Application"%2C"metadata"%3A%7B"name"%3A"test"%7D%2C... ☆ Incognito :

argo v2.10.4+f5d63a5

Application CREATE CANCEL

+ NEW

SOURCE

Repository URL
https://github.com/Sakeena19/Jenkins-Zero-To-Hero

GIT ▾

Revision
HEAD

Branches ▾

Path
java-maven-sonar-argocd-helm-k8s/spring-boot-app-manifests

DESTINATION

Cluster URL
https://kubernetes.default.svc

URL ▾

Namespace
default

Type here to search

Windows Start button

Taskbar icons: File Explorer, Mail, Google Chrome, Microsoft Edge, File Manager, Paint, Task View, Taskbar settings

System tray: Weather (28°C Sunny), Battery, Network, Language (ENG), Date (27-03-2024), Notifications (2)

Click on create, you will see that application will be automatically created using kubernetes cluster as argocd will fetch your application info from github repo

Applications Tiles - Argo CD

Not secure https://44.211.254.19:8080/applications?showFavorites=false&proj=&sync=&autoSync=&health=&namespace=&cluster=&labels=

Incognito

Applications

+ NEW APP SYNC APPS REFRESH APPS Search applications... /

APPLICATIONS TILES

Sort: name ▾ Items per page: 10 ▾

argo v2.10.4+f5d63a5

Applications

Settings

User Info

Documentation

Favorites Only

SYNC STATUS

- Unknown 0
- Synced 1
- OutOfSync 0

HEALTH STATUS

- Unknown 0
- Progressing 0
- Suspended 0
- Healthy 1
- Degraded 0
- Missing 0

test

Project: default

Labels:

Status: Healthy Synced

Repository: https://github.com/Sakeena19/Jenkins-...

Target Ref: HEAD

Path: java-maven-sonar-argocd-helm-k8s/sprin...

Destination: in-cluster

Namespace: default

Created: 03/27/2024 10:04:03 (a few seconds ago)

Last Sync: 03/27/2024 10:04:04 (a few seconds ago)

SYNC REFRESH DELETE

Type here to search

28°C Sunny 10:04 27-03-2024 ENG

test - Application Details Tree

Not secure https://44.211.254.19:8080/applications/argocd/test?view=tree&resource=

APPLICATION DETAILS TREE

arg v2.10.4+5d63a5

Applications / Q test

DETAILS DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

APP HEALTH Healthy

Sync Status Synced to HEAD (e9627d9)

Last Sync Sync OK to e9627d9

Auto sync is enabled.

Author: Sakeena19 <shaiksaheena19@gmail.com> - Comment: Update deployment image to version 2

Succeeded a few seconds ago (Wed Mar 27 2024 10:04:52 GMT+0530)

Author: Sakeena19 <shaiksaheena19@gmail.com> - Comment: Update deployment image to version 2

NAME

KINDS

SYNC STATUS

HEALTH STATUS

spring-boot-app-service

spring-boot-app-service-ck222

spring-boot-app

spring-boot-app-98f568686

spring-boot-app-981

spring-boot-app-service ep

spring-boot-app-service-ck222 endpointslice

spring-boot-app-98f568686 rs

spring-boot-app-981 pod

spring-boot-app-981 pod

test svc

deploy

Type here to search

28°C Sunny

ENG 27-03-2024

You can use below command to check whether the application is deployed

kubectl get deploy