

DevOps

Reddit Clone App Deployment with DevSecOps | Ingress

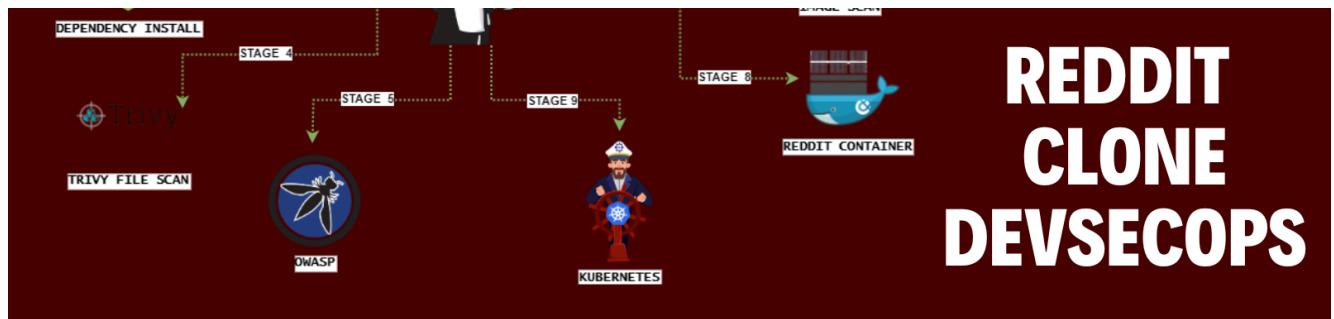


mrclocloudbook.com · 8 January 2024

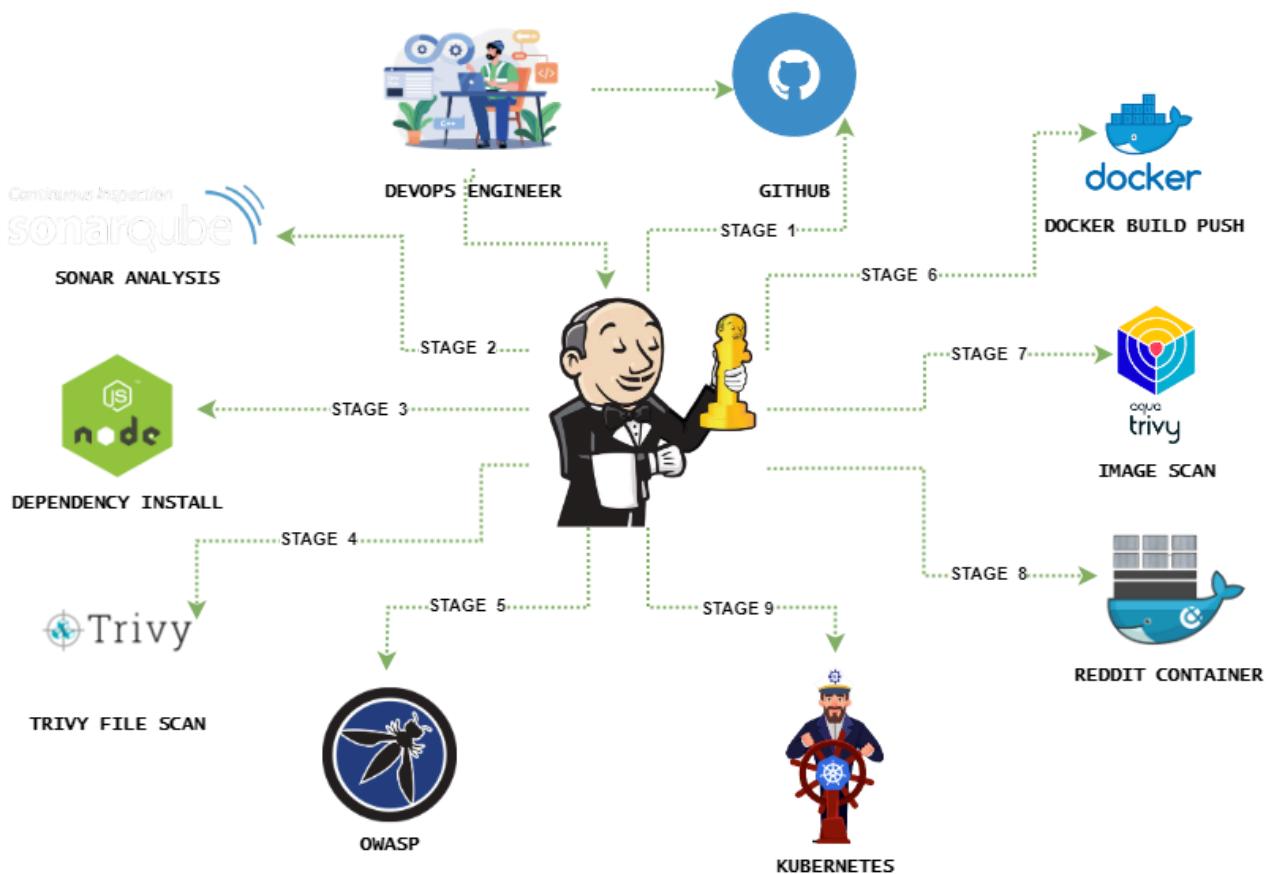
Mr Cloud Book

Home Blog DevSecOps Contact About Me Testimonials

Search Blogs



Hello friends, we will be deploying a React Js Reddit-clone. We will be using Jenkins as a CICD tool and deploying our application on a Docker container. I Hope this detailed blog is useful.



[CLICK HERE FOR GITHUB REPO](#)

Steps:-

Step 1 – Launch an Ubuntu(22.04) T2 Large Instance

Step 2 – Install Jenkins, Docker and Trivy. Create a Sonarqube Container using Docker.

Step 3 – Install Plugins like JDK, Sonarqube Scanner, Nodejs, and OWASP Dependency Check.

Step 4 – Create a Pipeline Project in Jenkins using a Declarative Pipeline

Step 5 – Install OWASP Dependency Check Plugins

Step 6 – Docker Image Build and Push

Step 7 – Deploy the image using Docker

Step 8 – Terminate the AWS EC2 Instances.

Now, let's get started and dig deeper into each of these steps:-

Contents [hide]

[STEP1:Launch an Ubuntu\(22.04\) T2 Large Instance](#)
[Step 2 – Install Jenkins, Docker and Trivy](#)
[2A – To Install Jenkins](#)
[2B – Install Docker](#)
[2C – Install Trivy](#)
[Step 3 – Install Plugins like JDK, Sonarqube Scanner, NodeJs, OWASP Dependency Check](#)
[3A – Install Plugin](#)
[3B – Configure Java and Nodejs in Global Tool Configuration](#)
[3C – Create a Job](#)
[Step 4 – Configure Sonar Server in Manage Jenkins](#)
[Step 5 – Install OWASP Dependency Check Plugins](#)
[Step 6 – Docker Image Build and Push](#)
[Step 8 – Kuberentes Setup](#)
[Kubectl is to be installed on Jenkins also](#)
[Part 1 -----Master Node-----](#)
[-----Worker Node-----](#)
[Part 2 -----Both Master & Node -----](#)
[Part 3 ----- Master -----](#)
[-----Worker Node-----](#)
[STEP9:Access from a Web browser with](#)
[Step 10: Terminate instances.](#)
[Complete Pipeline](#)

STEP1:Launch an Ubuntu(22.04) T2 Large Instance

Launch an AWS T2 Large Instance. Use the image as Ubuntu. You can create a new key pair or use an existing one. Enable HTTP and HTTPS settings in the Security Group and open all ports (not best case to open all ports but just for learning purposes it's okay).

Instances (1) Info										
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 Dl	Actions	Launch instances
<input type="checkbox"/>	CI-CD	i-065c10200537a1eee	Running	t2.large	2/2 checks passed	No alarms	ap-south-1a	ec2-52-66-14-	Actions	Launch instances

Step 2 – Install Jenkins, Docker and Trivy

2A – To Install Jenkins

Connect to your console, and enter these commands to install Jenkins

```
vi jenkins.sh #run in root
```



```
#!/bin/bash
sudo apt update -y
#sudo apt upgrade -y
wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public
echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/api/debian stable main" | sudo tee /etc/apt/sources.list.d/adoptium.list > /dev/null
sudo apt update -y
sudo apt install temurin-17-jdk -y
/usr/bin/java --version
curl -fsSL https://pkg.jenkins.io/debian-stable/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-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
sudo systemctl start jenkins
sudo systemctl status jenkins
```



```
sudo chmod 777 jenkins.sh
./jenkins.sh
```



Once Jenkins is installed, you will need to go to your AWS EC2 Security Group and open Inbound Port 8080, since Jenkins works on Port 8080.

Now, grab your Public IP Address



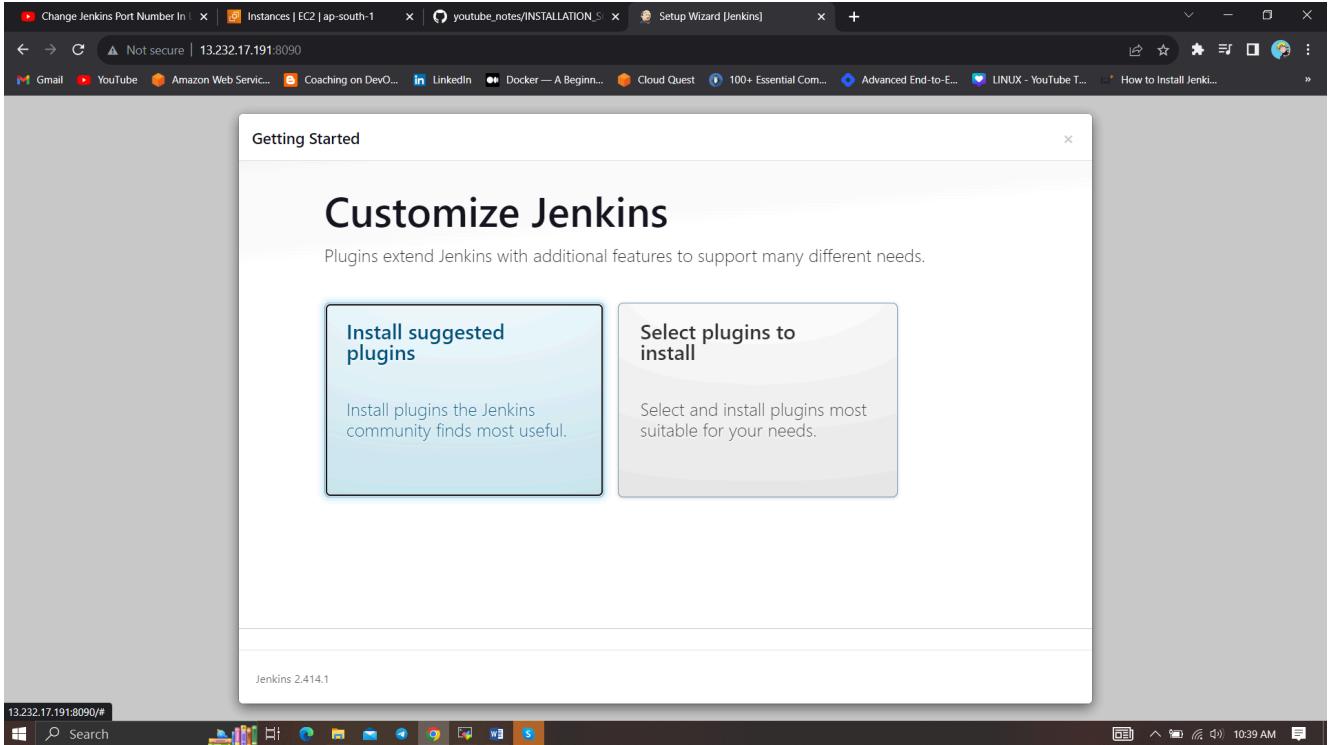
EC2 Public IP Address:8080

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

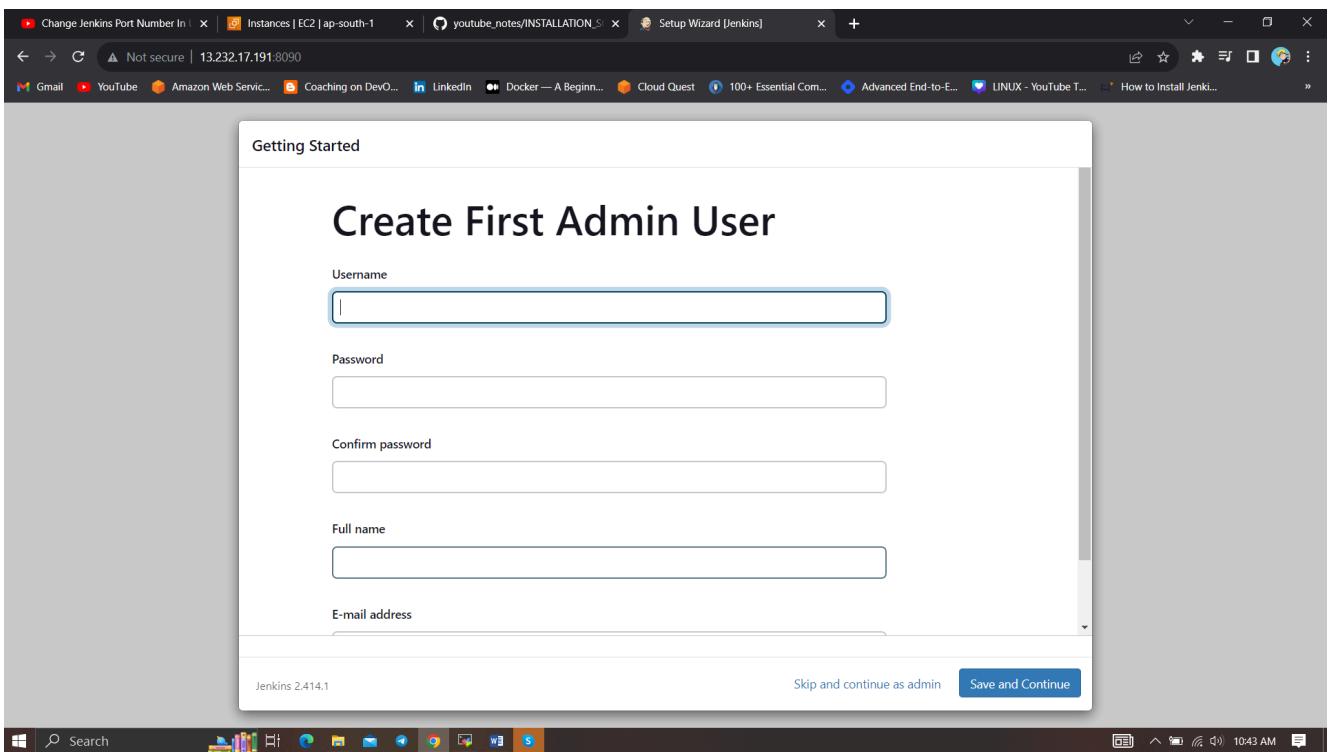


A screenshot of a Microsoft Edge browser window. The address bar shows the URL '13.232.17.191:8090/login?from=%2F'. The main content is a 'Getting Started' page titled 'Unlock Jenkins'. It instructs the user to copy the password from '/var/lib/jenkins/secrets/initialAdminPassword'. A text input field labeled 'Administrator password' is provided for pasting the password. A blue 'Continue' button is at the bottom right. The browser's taskbar at the bottom shows various pinned and open tabs.

Unlock Jenkins using an administrative password and install the suggested plugins.



Jenkins will now get installed and install all the libraries.



Create a user click on save and continue.

Jenkins Getting Started Screen.

Jenkins

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Set up a distributed build

Create a job →

Set up an agent →

Configure a cloud →

Learn more about distributed builds ↗

2B – Install Docker

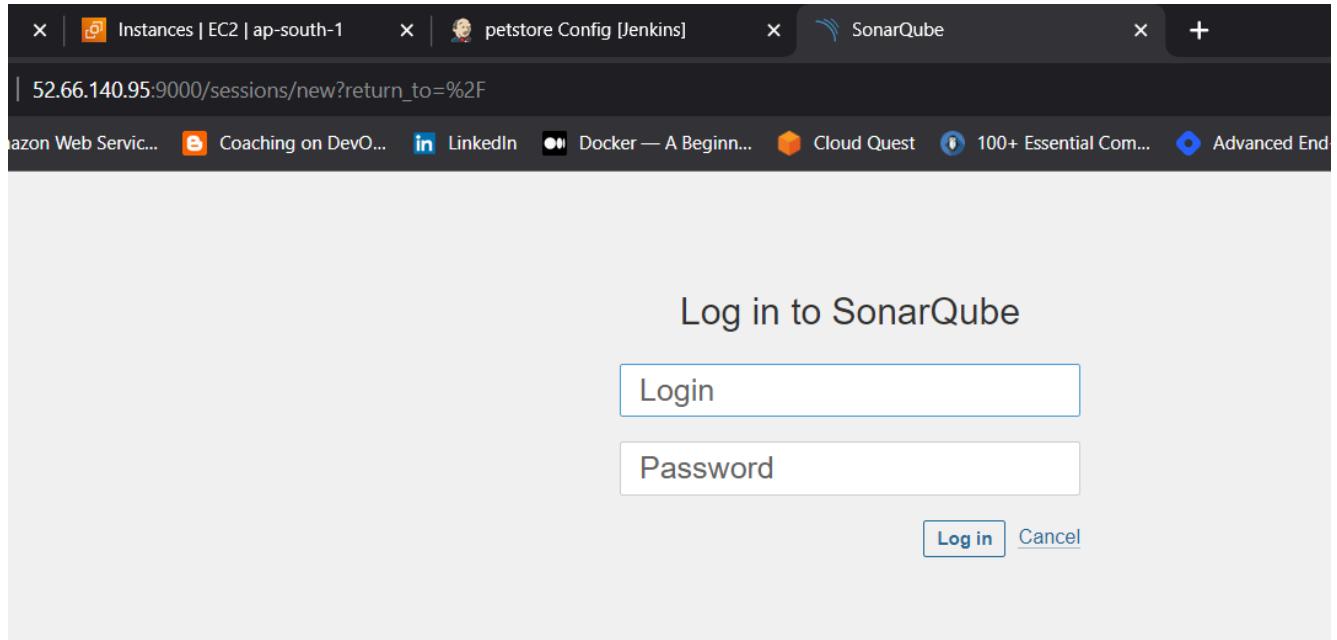
```
sudo apt-get update
sudo apt-get install docker.io -y
sudo usermod -aG docker $USER      #my case is ubuntu
newgrp docker
sudo chmod 777 /var/run/docker.sock
```

After the docker installation, we create a sonarqube container (Remember to add 9000 ports in the security group).

```
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

```
ubuntu@ip-172-31-42-253:~$ sudo chmod 777 /var/run/docker.sock
ubuntu@ip-172-31-42-253:~$ docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
44ba2882fb8eb: Pull complete
2cabec57ffa36: Pull complete
c20481384b6a: Pull complete
bf7b17ee74f8: Pull complete
38617faac714: Pull complete
706f20f58f5e: Pull complete
65a29568c257: Pull complete
Digest: sha256:1a118f8ab960d6c3d4ea8b4455a5a6560654511c88a6816f1603f764d5dcc77c
Status: Downloaded newer image for sonarqube:lts-community
4b66c96bf9ad3d62289436af7f752fd804993892d0ca5065e2f2e32301b50139
ubuntu@ip-172-31-42-253:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4b66c96bf9ad sonarqube:lts-community "/opt/sonarqube/dock..." 9 seconds ago Up 5 seconds 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp sonar
ubuntu@ip-172-31-42-253:~$
```

Now our sonarqube is up and running



Enter username and password, click on login and change password



```
username admin
password admin
```



Instances | EC2 | ap-south-1 x petstore Config [Jenkins] x SonarQube x +

6.140.95.9000/account/reset_password

Web Servic... Coaching on DevO... LinkedIn Docker — A Beginn... Cloud Quest 100+ Essential Com... Advanced End-to-E...

Update your password

This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Update

Update New password, This is Sonar Dashboard.

← → C Not secure | 52.66.140.95:9000/projects/create

Gmail YouTube Amazon Web Service... Coaching on DevO... LinkedIn Docker — A Beginn... Cloud Quest 100+ Essential Com... Advanced End-to-E... LINUX - YouTube T... How to Install Jenk...

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects... A

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform. First, you need to set up a DevOps platform configuration.

From Azure DevOps Set up global configuration	From Bitbucket Server Set up global configuration	From Bitbucket Cloud Set up global configuration	From GitHub Set up global configuration	From GitLab Set up global configuration
--	--	---	--	--

2C – Install Trivy



```
vi trivy.sh
```



```
sudo apt-get install wget apt-transport-https gnupg lsb-release -y  
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor > /usr/share/keyrings/trivy.gpg  
echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb/ /" | sudo tee /etc/apt/sources.list.d/trivy.list  
sudo apt-get update  
sudo apt-get install trivy -y
```



Next, we will log in to Jenkins and start to configure our Pipeline in Jenkins

Step 3 – Install Plugins like JDK, Sonarqube Scanner, NodeJs, OWASP Dependency Check

3A – Install Plugin

Goto Manage Jenkins → Plugins → Available Plugins →

Install below plugins

1 → Eclipse Temurin Installer (Install without restart)

2 → SonarQube Scanner (Install without restart)

3 → NodeJs Plugin (Install Without restart)

Plugins

Available plugins

Install	Name	Released
<input checked="" type="checkbox"/>	Eclipse Temurin installer 1.5 Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net	11 mo ago
<input checked="" type="checkbox"/>	SonarQube Scanner 2.15 <small>External Site/Tool Integrations Build Reports</small> This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	9 mo 19 days ago

Install Name ↓ Released

<input checked="" type="checkbox"/>	NodeJS 1.6.1 <small>npm</small> NodeJS Plugin executes NodeJS script as a build step.	1 mo 2 days ago
-------------------------------------	---	-----------------

3B – Configure Java and Nodejs in Global Tool Configuration

Goto Manage Jenkins → Tools → Install JDK(17) and NodeJs(16) → Click on Apply and Save

Add JDK

JDK

Name: jdk17

Install automatically

Install from adoptium.net

Version: jdk-17.0.8.1+1

Add Installer

NodeJS

Name: node16

Install automatically ?

Install from nodejs.org

Version: NodeJS 16.2.0

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail
 Force 32bit architecture

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax: 'packageName@version'

3C – Create a Job

create a job as Reddit Name, select pipeline and click on ok.

Step 4 – Configure Sonar Server in Manage Jenkins

Grab the Public IP Address of your EC2 Instance, Sonarqube works on Port 9000, so <Public IP>:9000. Goto your Sonarqube Server. Click on Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Administration

Configuration ▾ Security ▾ Projects ▾ System Marketplace

General ▾ **Users** Groups Global Permissions Permission Templates

Find i

click on update Token

SCM Accounts	Last connection	Groups	Tokens
A Administrator admin	< 1 hour ago	sonar-administrators sonar-users	

Update Tokens

Create a token with a name and generate

Tokens of Administrator

Generate Tokens

Name	Expires in
Enter Token Name	30 days
<input type="button" value="Generate"/>	

New token "Jenkins" has been created. Make sure you copy it now, you won't be able to see it again!

Copy :squ_21d162904c1c72cf8b39665f96480185c99dc2f9

Name	Type	Project	Last use	Created	Expiration
Jenkins	User		Never	September 8, 2023	October 8, 2023
					Revoke

copy Token

Goto Jenkins Dashboard → Manage Jenkins → Credentials → Add Secret Text. It should look like this

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

New credentials

Kind: Secret text

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

Secret: POST THE TOKEN HERE

ID: Sonar-token

Description: Sonar-token

Create

You will see this page once you click on create

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description	
Sonar-token	sonar	Secret text	sonar	

Now, go to Dashboard → Manage Jenkins → System and Add like the below image.

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables Enable injection of SonarQube server configuration as build environment variables

SonarQube installations

List of SonarQube installations

Name	<input type="text" value="sonar-server"/>	X
Server URL	Default is http://localhost:9000 <input type="text" value="http://13.232.17.191:9000"/>	
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled. <input type="text" value="Sonar-token"/>	
<input style="margin-right: 10px;" type="button" value="Add"/> <input style="background-color: #0072bc; color: white; border-radius: 5px; padding: 5px; margin-right: 10px;" type="button" value="Save"/> <input type="button" value="Apply"/>		

Click on Apply and Save

The Configure System option is used in Jenkins to configure different server

Global Tool Configuration is used to configure different tools that we install using Plugins

We will install a sonar scanner in the tools.

SonarQube Scanner installations**SonarQube Scanner****Name**

Install automatically ?

Install from Maven Central**Version**

In the Sonarqube Dashboard add a quality gate also

Administration-> Configuration->Webhooks

The screenshot shows the SonarQube administration interface. The top navigation bar has a red box around the 'Administration' tab. Below it, the 'Configuration' tab is selected. On the left, a sidebar menu has a red box around the 'Webhooks' option. The main content area displays a table of users, with one user listed: 'Administrator admin'. The table includes columns for SCM Accounts, Last connection, Groups, and Tokens. A 'Create User' button is located in the top right corner of the user table area.

Click on Create

The screenshot shows the SonarQube 'Webhooks' configuration page. The top navigation bar has a red box around the 'Administration' tab. Below it, the 'Configuration' tab is selected. The main content area is titled 'Webhooks' and contains a brief description: 'Webhooks are used to notify external services when a project analysis is done. An HTTP POST request including a JSON payload is sent to each of the provided URLs. Learn more in the [Webhooks documentation](#)'. To the right of the description is a large 'Create' button, which is highlighted with a red box.

Add details

```
#in url section of quality gate
<http://jenkins-public-ip:8080>/sonarqube-webhook/
```

Create Webhook

All fields marked with * are required

Name *
jenkins

URL *
http://43.204.36.242:8090/sonarqube-webhook/

Server endpoint that will receive the webhook payload, for example:
"http://my_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example:
"https://myLogin:myPassword@my_server/foo"

Secret
[Empty]

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header.

Create **Cancel**

Get the most out of SonarQube! Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. **Learn More** **Dismiss**

Let's go to our Pipeline and add the script in our Pipeline Script.

```
pipeline{
    agent any
    tools{
        jdk 'jdk17'
        nodejs 'node16'
    }
    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
    stages {
        stage('clean workspace'){
            steps{
                cleanWs()
            }
        }
        stage('Checkout from Git'){
            steps{
                git branch: 'main', url: 'https://github.com/Aj7Ay/redd'
            }
        }
        stage("Sonarqube Analysis"){
            steps{
                withSonarQubeEnv('sonar-server') {
                    sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=RedditClone -Dsonar.sources=. -Dsonar.java.binaries=target -Dsonar.host.url=http://43.204.36.242:8090/sonarqube-webhook/ '''
                }
            }
        }
    }
}
```

```

-Dsonar.projectKey=Reddit ''
}

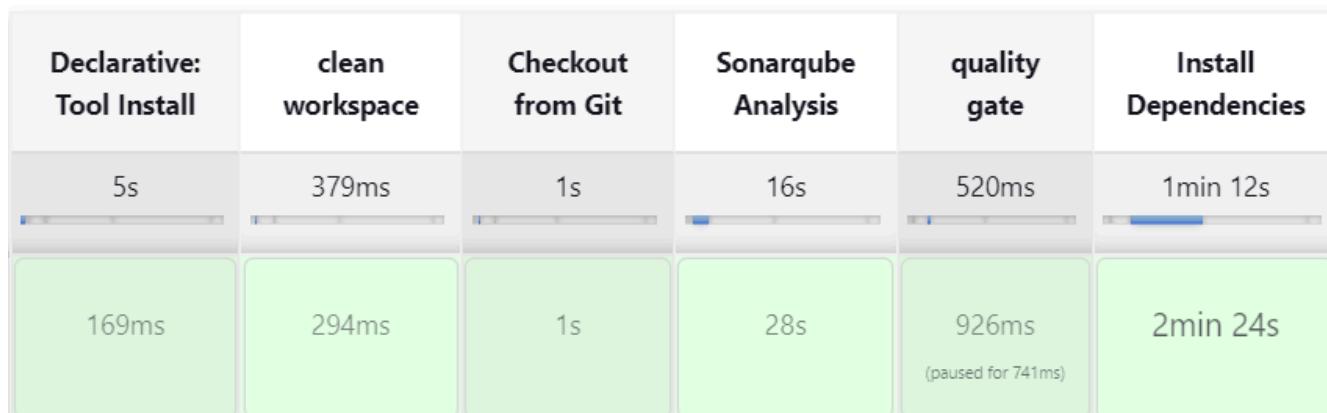
}

stage("quality gate"){
    steps {
        script {
            waitForQualityGate abortPipeline: false, credential:
        }
    }
}

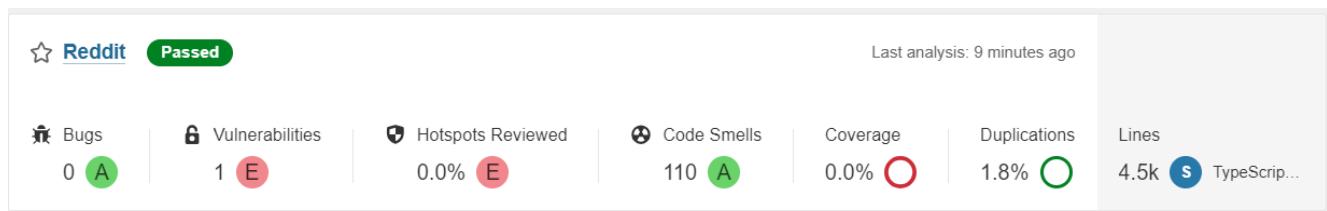
stage('Install Dependencies') {
    steps {
        sh "npm install"
    }
}
}
}

```

Click on Build now, you will see the stage view like this



To see the report, you can go to Sonarqube Server and go to Projects.



You can see that there are 4.5k lines it scanned. To see a detailed report, you can go to

issues.

Step 5 – Install OWASP Dependency Check Plugins

Goto Dashboard → Manage Jenkins → Plugins → OWASP Dependency-Check. Click on it and install it without restart.

The screenshot shows the Jenkins Plugins page. On the left, there are navigation links: Updates, Available plugins (which is selected), Installed plugins, Advanced settings, and Download progress. The main area is titled "Plugins" and shows a search bar with "Search available plugins". A list of available plugins is displayed, with "OWASP Dependency-Check 5.4.2" highlighted. This plugin is categorized under Security, DevOps, Build Tools, and Build Reports. It has a release date of "8 days 17 hr ago". A description below the plugin states: "This plug-in can independently execute a Dependency-Check analysis and visualize results. Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities." An "Install" button is visible at the top right of the plugin's card.

First, we configured the Plugin and next, we had to configure the Tool

Goto Dashboard → Manage Jenkins → Tools →

The screenshot shows the Jenkins Tools page. The top navigation bar includes "Dashboard", "Manage Jenkins", and "Tools". Below this, the page title is "Dependency-Check installations". There is a "Add Dependency-Check" button. A configuration section for "Dependency-Check" is shown, with a "Name" field containing "DP-Check" and an "Install automatically" checkbox checked. A dashed box encloses a "Install from github.com" section, which includes a "Version" field with "dependency-check 6.5.1" and an "Add Installer" button. The entire configuration is enclosed in a dashed box.

Click on Apply and Save here.

Now go configure → Pipeline and add this stage to your pipeline and build.

```
stage('OWASP FS SCAN') {  
    steps {  
        dependencyCheck additionalArguments: '--scan ./ --disable-scan-known-vulnerabilities'  
        dependencyCheckPublisher pattern: '**/dependency-check-report.html'  
    }  
}  
stage('TRIVY FS SCAN') {  
    steps {  
        sh "trivy fs . > trivyfs.txt"  
    }  
}
```



The stage view would look like this,



You will see that in status, a graph will also be generated and Vulnerabilities.

Dependency-Check Results

Severity Distribution			
5	8	3	
File Name	Vulnerability	Severity	Weakness
+ css-what:3.4.2	OSSINDEX CVE-2022-21222	High	CWE-1333
+ ejs:3.1.8	OSSINDEX CVE-2023-29827	High	CWE-74
+ json5:1.0.1	NVD CVE-2022-46175	High	CWE-1321
+ jsonpointer:5.0.1	NVD CVE-2022-4742	Critical	CWE-1321
+ nth-check:1.0.2	NVD CVE-2021-3803	High	CWE-1333
+ parseurl:1.3.3	NVD CVE-2022-0722	High	CWE-200
+ parseurl:1.3.3	NVD CVE-2022-2216	Critical	CWE-918
+ parseurl:1.3.3	NVD CVE-2022-2217	Medium	CWE-79
+ parseurl:1.3.3	NVD CVE-2022-2218	Medium	CWE-79
+ parseurl:1.3.3	NVD CVE-2022-2900	Critical	CWE-918

Step 6 – Docker Image Build and Push

We need to install the Docker tool in our system, Goto Dashboard → Manage Plugins → Available plugins → Search for Docker and install these plugins

Docker

Docker Commons

Docker Pipeline

Docker API

docker-build-step

and click on install without restart

The screenshot shows the Jenkins Plugins page with a search bar for "docker". The results list three plugins:

- Docker 1.5**: Version 1.5, released 3 days 15 hr ago. This plugin integrates Jenkins with Docker.
- Docker Commons**: Version 439.va_3cb_0a_6a_fb_29, released 1 mo 29 days ago. Provides the common shared functionality for various Docker-related plugins.
- Docker Pipeline**: Version 572.v950f58993843, released 27 days ago. Build and use Docker containers from pipelines.
- Docker API**: Version 3.3.1-79.v20b_53427e041, released 3 mo 4 days ago. This plugin provides docker-java API for other plugins.

Now, goto Dashboard → Manage Jenkins → Tools →

The screenshot shows the Jenkins Tools configuration page under "Docker installations". A new configuration is being added with the following details:

- Name**: docker
- Install automatically**: checked
- Download from docker.com**
- Docker version**: latest
- Add Installer**: dropdown menu

Add DockerHub Username and Password under Global Credentials

Kind

Username with password

Scope ?

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

Username ?

sevenajay

 Treat username as secret ?

Password ?

.....

ID ?

docker

Description ?

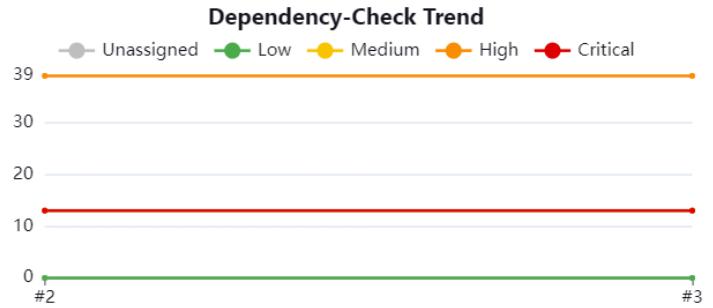
docker

Create**Add this stage to Pipeline Script**

```
stage("Docker Build & Push"){
    steps{
        script{
            withDockerRegistry(credentialsId: 'docker', toolName
                sh "docker build -t reddit ."
                sh "docker tag reddit sevenajay/reddit:latest"
                sh "docker push sevenajay/reddit:latest"
            }
        }
    }
}
stage("TRIVY"){
    steps{
        sh "trivy image sevenajay/reddit:latest > trivy.txt"
    }
}
```



You will see the output below, with a dependency trend.



Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN	Docker Build & Push	TRIVY
3s	366ms	1s	19s	451ms	1min 20s	2min 1s	16s	3min 9s	4s
154ms	341ms	1s	25s	315ms	1min 36s	2min 31s	23s	3min 9s	4s

When you log in to Dockerhub, you will see a new image is created

Now Run the container to see if the app coming up or not by adding the below stage

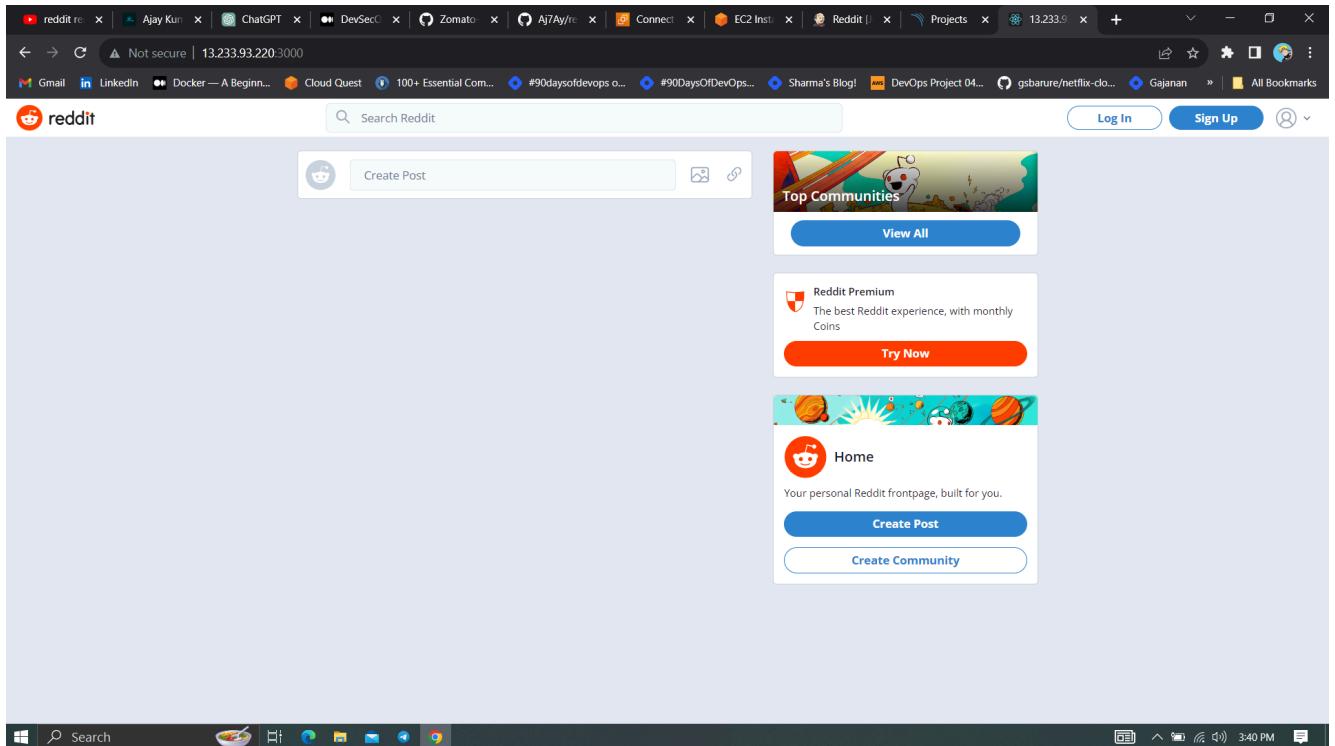
```
stage('Deploy to container'){
    steps{
        sh 'docker run -d --name reddit -p 3000:3000 sevenajay/reddit'
    }
}
```

stage view

Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN	Docker Build & Push	TRIVY	Deploy to container
144ms	284ms	1s	25s	410ms	1min 47s	2min 43s	23s	2min 7s	36s	789ms
146ms	251ms	1s	26s	305ms	1min 36s	2min 35s	23s	1min 50s	2min 8s	1s

<Jenkins-public-ip:3000>

You will get this output



Step 8 – Kuberenetes Setup

Connect your machines to Putty or Mobaxtreme

Take-Two Ubuntu 20.04 instances one for k8s master and the other one for worker.

Install Kubectl on Jenkins machine also.

Kubectl is to be installed on Jenkins also

```
sudo apt update
sudo apt install curl
curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/releas...
```

```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl  
kubectl version --client
```

Part 1 ----Master Node-----

```
sudo hostnamectl set-hostname K8s-Master
```



-----Worker Node-----

```
sudo hostnamectl set-hostname K8s-Worker
```



Part 2 -----Both Master & Node -----

```
sudo apt-get update  
sudo apt-get install -y docker.io  
sudo usermod -aG docker Ubuntu  
newgrp docker  
sudo chmod 777 /var/run/docker.sock  
sudo curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo tee /etc/apt/sources.list.d/kubernetes.list <<EOF  
deb https://apt.kubernetes.io/ kubernetes-xenial main  
EOF  
sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo snap install kube-apiserver
```



Part 3 ----- Master -----



```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
# in case your in root exit from it and run below commands
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master
```



-----Worker Node-----



```
sudo kubeadm join <master-node-ip>:<master-node-port> --token <token> --
```



Copy the config file to Jenkins master or the local file manager and save it

```
cat: c:\> no such file or directory
ubuntu@ip-172-31-40-131:~/kube$ cat config
apiVersion: v1
clusters:
- cluster:
  certIFICATE-authority-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT1d3GoycW91b1drNTh35FpMS1T2kupSp0xRj0dnXpHeHh0zGz28UFNNT3NFt2kdnY3L2dz0WVJSEhxK0R2UzBFsFRC13pq0HBk0fFa5jYzblwdqFtpCmxsWp1dRteWxX0VXkEeNW0VppcUlyWt10TlnqFceWvha2d2krcnyky9TTmzvNk02T1hTcDjycktWlzJa5n0kdn91Y1YMERJ1T02W15T19kV1U0e1Y1nn1kCcMz2NmzUlmTuSj3DM1oYwXbxmhu5e93A1V1XBmenczzc1j1bgpoY2Y4d4TfN0n1xEnHoeXlojy0rRmSVTHYSUo2QW0N0KFckJn1TzUk1C0WY4RJ0JU0URBlUgjUyTJnRExEWREZ1Fx0kJtaWWRGvNtRlCjVucE9BaVd0CwduJdNkRNQpVY0k0md0VknSVReakFN2dwcimXSmjzTbVs2dWeck1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzKzVzCm9vZvFw1M12RLJD0WvEc090SDNCNSHryTHN1Sc91J23hWmJ3WjFXWVhmmDdyUlkwd1pCNWNYn1KzRHSUs2R0wKL0kvFbra0jPUetYk1J5mY5L3DVWEX0x05NhHpv2VXyJyU0FraBDU3h0e1Vz0UV5sjkycn1oR3N3V1ljMgo1TC9VbmRpJUX0Uelcge54UEhle1NS5TfXSwBx0UzwdWSek5MYT2EeJu0zV1YeXwLy6RDC35Eh6ahrN2VNCnLnmRpaeJkRvcxK3pUnkLjyWjVj0zXzS1gbxStqTfVy0lRUv20rVvgcncveWx6aBWSMUKZ3BzZfRdwNDT00wC10tR0tR5E1EN0tUJrjk1D0vRFLs0tLS0k
  server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    cLIENT-certificate-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT1d3GoycW91b1drNTh35FpMS1T2kupSp0xRj0dnXpHeHh0zGz28UFNNT3NFt2kdnY3L2dz0WVJSEhxK0R2UzBFsFRC13pq0HBk0fFa5jYzblwdqFtpCmxsWp1dRteWxX0VXkEeNW0VppcUlyWt10TlnqFceWvha2d2krcnyky9TTmzvNk02T1hTcDjycktWlzJa5n0kdn91Y1YMERJ1T02W15T19kV1U0e1Y1nn1kCcMz2NmzUlmTuSj3DM1oYwXbxmhu5e93A1V1XBmenczzc1j1bgpoY2Y4d4TfN0n1xEnHoeXlojy0rRmSVTHYSUo2QW0N0KFckJn1TzUk1C0WY4RJ0JU0URBlUgjUyTJnRExEWREZ1Fx0kJtaWWRGvNtRlCjVucE9BaVd0CwduJdNkRNQpVY0k0md0VknSVReakFN2dwcimXSmjzTbVs2dWeck1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzKzVzCm9vZvFw1M12RLJD0WvEc090SDNCNSHryTHN1Sc91J23hWmJ3WjFXWVhmmDdyUlkwd1pCNWNYn1KzRHSUs2R0wKL0kvFbra0jPUetYk1J5mY5L3DVWEX0x05NhHpv2VXyJyU0FraBDU3h0e1Vz0UV5sjkycn1oR3N3V1ljMgo1TC9VbmRpJUX0Uelcge54UEhle1NS5TfXSwBx0UzwdWSek5MYT2EeJu0zV1YeXwLy6RDC35Eh6ahrN2VNCnLnmRpaeJkRvcxK3pUnkLjyWjVj0zXzS1gbxStqTfVy0lRUv20rVvgcncveWx6aBWSMUKZ3BzZfRdwNDT00wC10tR0tR5E1EN0tUJrjk1D0vRFLs0tLS0k
  server:
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    cLIENT-certificate-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT1d3GoycW91b1drNTh35FpMS1T2kupSp0xRj0dnXpHeHh0zGz28UFNNT3NFt2kdnY3L2dz0WVJSEhxK0R2UzBFsFRC13pq0HBk0fFa5jYzblwdqFtpCmxsWp1dRteWxX0VXkEeNW0VppcUlyWt10TlnqFceWvha2d2krcnyky9TTmzvNk02T1hTcDjycktWlzJa5n0kdn91Y1YMERJ1T02W15T19kV1U0e1Y1nn1kCcMz2NmzUlmTuSj3DM1oYwXbxmhu5e93A1V1XBmenczzc1j1bgpoY2Y4d4TfN0n1xEnHoeXlojy0rRmSVTHYSUo2QW0N0KFckJn1TzUk1C0WY4RJ0JU0URBlUgjUyTJnRExEWREZ1Fx0kJtaWWRGvNtRlCjVucE9BaVd0CwduJdNkRNQpVY0k0md0VknSVReakFN2dwcimXSmjzTbVs2dWeck1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzKzVzCm9vZvFw1M12RLJD0WvEc090SDNCNSHryTHN1Sc91J23hWmJ3WjFXWVhmmDdyUlkwd1pCNWNYn1KzRHSUs2R0wKL0kvFbra0jPUetYk1J5mY5L3DVWEX0x05NhHpv2VXyJyU0FraBDU3h0e1Vz0UV5sjkycn1oR3N3V1ljMgo1TC9VbmRpJUX0Uelcge54UEhle1NS5TfXSwBx0UzwdWSek5MYT2EeJu0zV1YeXwLy6RDC35Eh6ahrN2VNCnLnmRpaeJkRvcxK3pUnkLjyWjVj0zXzS1gbxStqTfVy0lRUv20rVvgcncveWx6aBWSMUKZ3BzZfRdwNDT00wC10tR0tR5E1EN0tUJrjk1D0vRFLs0tLS0k
  server:
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    cLIENT-certificate-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT1d3GoycW91b1drNTh35FpMS1T2kupSp0xRj0dnXpHeHh0zGz28UFNNT3NFt2kdnY3L2dz0WVJSEhxK0R2UzBFsFRC13pq0HBk0fFa5jYzblwdqFtpCmxsWp1dRteWxX0VXkEeNW0VppcUlyWt10TlnqFceWvha2d2krcnyky9TTmzvNk02T1hTcDjycktWlzJa5n0kdn91Y1YMERJ1T02W15T19kV1U0e1Y1nn1kCcMz2NmzUlmTuSj3DM1oYwXbxmhu5e93A1V1XBmenczzc1j1bgpoY2Y4d4TfN0n1xEnHoeXlojy0rRmSVTHYSUo2QW0N0KFckJn1TzUk1C0WY4RJ0JU0URBlUgjUyTJnRExEWREZ1Fx0kJtaWWRGvNtRlCjVucE9BaVd0CwduJdNkRNQpVY0k0md0VknSVReakFN2dwcimXSmjzTbVs2dWeck1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzKzVzCm9vZvFw1M12RLJD0WvEc090SDNCNSHryTHN1Sc91J23hWmJ3WjFXWVhmmDdyUlkwd1pCNWNYn1KzRHSUs2R0wKL0kvFbra0jPUetYk1J5mY5L3DVWEX0x05NhHpv2VXyJyU0FraBDU3h0e1Vz0UV5sjkycn1oR3N3V1ljMgo1TC9VbmRpJUX0Uelcge54UEhle1NS5TfXSwBx0UzwdWSek5MYT2EeJu0zV1YeXwLy6RDC35Eh6ahrN2VNCnLnmRpaeJkRvcxK3pUnkLjyWjVj0zXzS1gbxStqTfVy0lRUv20rVvgcncveWx6aBWSMUKZ3BzZfRdwNDT00wC10tR0tR5E1EN0tUJrjk1D0vRFLs0tLS0k
  server:
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    cLIENT-certificate-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT1d3GoycW91b1drNTh35FpMS1T2kupSp0xRj0dnXpHeHh0zGz28UFNNT3NFt2kdnY3L2dz0WVJSEhxK0R2UzBFsFRC13pq0HBk0fFa5jYzblwdqFtpCmxsWp1dRteWxX0VXkEeNW0VppcUlyWt10TlnqFceWvha2d2krcnyky9TTmzvNk02T1hTcDjycktWlzJa5n0kdn91Y1YMERJ1T02W15T19kV1U0e1Y1nn1kCcMz2NmzUlmTuSj3DM1oYwXbxmhu5e93A1V1XBmenczzc1j1bgpoY2Y4d4TfN0n1xEnHoeXlojy0rRmSVTHYSUo2QW0N0KFckJn1TzUk1C0WY4RJ0JU0URBlUgjUyTJnRExEWREZ1Fx0kJtaWWRGvNtRlCjVucE9BaVd0CwduJdNkRNQpVY0k0md0VknSVReakFN2dwcimXSmjzTbVs2dWeck1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzKzVzCm9vZvFw1M12RLJD0WvEc090SDNCNSHryTHN1Sc91J23hWmJ3WjFXWVhmmDdyUlkwd1pCNWNYn1KzRHSUs2R0wKL0kvFbra0jPUetYk1J5mY5L3DVWEX0x05NhHpv2VXyJyU0FraBDU3h0e1Vz0UV5sjkycn1oR3N3V1ljMgo1TC9VbmRpJUX0Uelcge54UEhle1NS5TfXSwBx0UzwdWSek5MYT2EeJu0zV1YeXwLy6RDC35Eh6ahrN2VNCnLnmRpaeJkRvcxK3pUnkLjyWjVj0zXzS1gbxStqTfVy0lRUv20rVvgcncveWx6aBWSMUKZ3BzZfRdwNDT00wC10tR0tR5E1EN0tUJrjk1D0vRFLs0tLS0k
  server:
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    cLIENT-certificate-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT1d3GoycW91b1drNTh35FpMS1T2kupSp0xRj0dnXpHeHh0zGz28UFNNT3NFt2kdnY3L2dz0WVJSEhxK0R2UzBFsFRC13pq0HBk0fFa5jYzblwdqFtpCmxsWp1dRteWxX0VXkEeNW0VppcUlyWt10TlnqFceWvha2d2krcnyky9TTmzvNk02T1hTcDjycktWlzJa5n0kdn91Y1YMERJ1T02W15T19kV1U0e1Y1nn1kCcMz2NmzUlmTuSj3DM1oYwXbxmhu5e93A1V1XBmenczzc1j1bgpoY2Y4d4TfN0n1xEnHoeXlojy0rRmSVTHYSUo2QW0N0KFckJn1TzUk1C0WY4RJ0JU0URBlUgjUyTJnRExEWREZ1Fx0kJtaWWRGvNtRlCjVucE9BaVd0CwduJdNkRNQpVY0k0md0VknSVReakFN2dwcimXSmjzTbVs2dWeck1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzKzVzCm9vZvFw1M12RLJD0WvEc090SDNCNSHryTHN1Sc91J23hWmJ3WjFXWVhmmDdyUlkwd1pCNWNYn1KzRHSUs2R0wKL0kvFbra0jPUetYk1J5mY5L3DVWEX0x05NhHpv2VXyJyU0FraBDU3h0e1Vz0UV5sjkycn1oR3N3V1ljMgo1TC9VbmRpJUX0Uelcge54UEhle1NS5TfXSwBx0UzwdWSek5MYT2EeJu0zV1YeXwLy6RDC35Eh6ahrN2VNCnLnmRpaeJkRvcxK3pUnkLjyWjVj0zXzS1gbxStqTfVy0lRUv20rVvgcncveWx6aBWSMUKZ3BzZfRdwNDT00wC10tR0tR5E1EN0tUJrjk1D0vRFLs0tLS0k
  server:
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    cLIENT-certificate-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT1d3GoycW91b1drNTh35FpMS1T2kupSp0xRj0dnXpHeHh0zGz28UFNNT3NFt2kdnY3L2dz0WVJSEhxK0R2UzBFsFRC13pq0HBk0fFa5jYzblwdqFtpCmxsWp1dRteWxX0VXkEeNW0VppcUlyWt10TlnqFceWvha2d2krcnyky9TTmzvNk02T1hTcDjycktWlzJa5n0kdn91Y1YMERJ1T02W15T19kV1U0e1Y1nn1kCcMz2NmzUlmTuSj3DM1oYwXbxmhu5e93A1V1XBmenczzc1j1bgpoY2Y4d4TfN0n1xEnHoeXlojy0rRmSVTHYSUo2QW0N0KFckJn1TzUk1C0WY4RJ0JU0URBlUgjUyTJnRExEWREZ1Fx0kJtaWWRGvNtRlCjVucE9BaVd0CwduJdNkRNQpVY0k0md0VknSVReakFN2dwcimXSmjzTbVs2dWeck1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzKzVzCm9vZvFw1M12RLJD0WvEc090SDNCNSHryTHN1Sc91J23hWmJ3WjFXWVhmmDdyUlkwd1pCNWNYn1KzRHSUs2R0wKL0kvFbra0jPUetYk1J5mY5L3DVWEX0x05NhHpv2VXyJyU0FraBDU3h0e1Vz0UV5sjkycn1oR3N3V1ljMgo1TC9VbmRpJUX0Uelcge54UEhle1NS5TfXSwBx0UzwdWSek5MYT2EeJu0zV1YeXwLy6RDC35Eh6ahrN2VNCnLnmRpaeJkRvcxK3pUnkLjyWjVj0zXzS1gbxStqTfVy0lRUv20rVvgcncveWx6aBWSMUKZ3BzZfRdwNDT00wC10tR0tR5E1EN0tUJrjk1D0vRFLs0tLS0k
  server:
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    cLIENT-certificate-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT1d3GoycW91b1drNTh35FpMS1T2kupSp0xRj0dnXpHeHh0zGz28UFNNT3NFt2kdnY3L2dz0WVJSEhxK0R2UzBFsFRC13pq0HBk0fFa5jYzblwdqFtpCmxsWp1dRteWxX0VXkEeNW0VppcUlyWt10TlnqFceWvha2d2krcnyky9TTmzvNk02T1hTcDjycktWlzJa5n0kdn91Y1YMERJ1T02W15T19kV1U0e1Y1nn1kCcMz2NmzUlmTuSj3DM1oYwXbxmhu5e93A1V1XBmenczzc1j1bgpoY2Y4d4TfN0n1xEnHoeXlojy0rRmSVTHYSUo2QW0N0KFckJn1TzUk1C0WY4RJ0JU0URBlUgjUyTJnRExEWREZ1Fx0kJtaWWRGvNtRlCjVucE9BaVd0CwduJdNkRNQpVY0k0md0VknSVReakFN2dwcimXSmjzTbVs2dWeck1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzKzVzCm9vZvFw1M12RLJD0WvEc090SDNCNSHryTHN1Sc91J23hWmJ3WjFXWVhmmDdyUlkwd1pCNWNYn1KzRHSUs2R0wKL0kvFbra0jPUetYk1J5mY5L3DVWEX0x05NhHpv2VXyJyU0FraBDU3h0e1Vz0UV5sjkycn1oR3N3V1ljMgo1TC9VbmRpJUX0Uelcge54UEhle1NS5TfXSwBx0UzwdWSek5MYT2EeJu0zV1YeXwLy6RDC35Eh6ahrN2VNCnLnmRpaeJkRvcxK3pUnkLjyWjVj0zXzS1gbxStqTfVy0lRUv20rVvgcncveWx6aBWSMUKZ3BzZfRdwNDT00wC10tR0tR5E1EN0tUJrjk1D0vRFLs0tLS0k
  server:
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    cLIENT-certificate-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT1d3GoycW91b1drNTh35FpMS1T2kupSp0xRj0dnXpHeHh0zGz28UFNNT3NFt2kdnY3L2dz0WVJSEhxK0R2UzBFsFRC13pq0HBk0fFa5jYzblwdqFtpCmxsWp1dRteWxX0VXkEeNW0VppcUlyWt10TlnqFceWvha2d2krcnyky9TTmzvNk02T1hTcDjycktWlzJa5n0kdn91Y1YMERJ1T02W15T19kV1U0e1Y1nn1kCcMz2NmzUlmTuSj3DM1oYwXbxmhu5e93A1V1XBmenczzc1j1bgpoY2Y4d4TfN0n1xEnHoeXlojy0rRmSVTHYSUo2QW0N0KFckJn1TzUk1C0WY4RJ0JU0URBlUgjUyTJnRExEWREZ1Fx0kJtaWWRGvNtRlCjVucE9BaVd0CwduJdNkRNQpVY0k0md0VknSVReakFN2dwcimXSmjzTbVs2dWeck1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzKzVzCm9vZvFw1M12RLJD0WvEc090SDNCNSHryTHN1Sc91J23hWmJ3WjFXWVhmmDdyUlkwd1pCNWNYn1KzRHSUs2R0wKL0kvFbra0jPUetYk1J5mY5L3DVWEX0x05NhHpv2VXyJyU0FraBDU3h0e1Vz0UV5sjkycn1oR3N3V1ljMgo1TC9VbmRpJUX0Uelcge54UEhle1NS5TfXSwBx0UzwdWSek5MYT2EeJu0zV1YeXwLy6RDC35Eh6ahrN2VNCnLnmRpaeJkRvcxK3pUnkLjyWjVj0zXzS1gbxStqTfVy0lRUv20rVvgcncveWx6aBWSMUKZ3BzZfRdwNDT00wC10tR0tR5E1EN0tUJrjk1D0vRFLs0tLS0k
  server:
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    cLIENT-certificate-data: LS0tLS1CRUdJTBDRVJUSUZJ00FURS0tLS0tCK1JSURCVEND0WlUyZ0F3SUJBZ0lJR0xxLzWe69Yy1V3RFZSktyWkLodnN0QVFTEJR0XdGVEVJUJF0ExVJUJK0XhNS2EzVmleEp1WhSbGN60WVGdzb5TxbNU1EZ3d0akkzTVRSYUZ3MhpnkE1TRVR6d0SqTxLNFeJhTlVeApPekFSmd0vKjB7VDb0xW0lWeWtVtJbaWf1E32d2fa1J1BMEd0J3FHU01lM0RRRBJUvB0RTRJ0K3D0hZ0V1CkFyS1JBU1N3TlHcb02T3hT3RjYmlzZkDvcyDzJ6RjvB10Yj00828MV0TE1bVzEkpyd0TvYU5o2h2013V082yj1wHFhFk0dM12G0mfCMn05bl1052fcF1bd2p2lcJ1cT
```

copy it and save it in documents or another folder save it as secret-file.txt

Note: create a secret-file.txt in your file explorer save the config in it and use this at the kubernetes credential section.

Install Kubernetes Plugin, Once it's installed successfully

Install	Name	Released
<input checked="" type="checkbox"/>	Kubernetes Credentials 0.11 kubernetes credentials Common classes for Kubernetes credentials	9 days 16 hr ago
<input checked="" type="checkbox"/>	Kubernetes Client API 6.8.1-224.vd388fca_4db_3b kubernetes Library plugins (for use by other plugins) Kubernetes Client API plugin for use by other Jenkins plugins.	9 days 17 hr ago
<input checked="" type="checkbox"/>	Kubernetes 4029.v5712230ccb_f8 Cloud Providers Cluster Management kubernetes Agent Management This plugin integrates Jenkins with Kubernetes	9 days 15 hr ago
<input checked="" type="checkbox"/>	Kubernetes CLI 1.12.1 kubernetes Configure kubectl for Kubernetes	8 days 22 hr ago

goto manage Jenkins -> manage credentials -> Click on Jenkins global -> add credentials

final step to deploy on the Kubernetes cluster



```

stage('Deploy to kubernets'){
    steps{
        script{
            withKubeConfig(caCertificate: '', clusterName: '', {
                sh 'kubectl apply -f deployment.yml'
                sh 'kubectl apply -f service.yml'
                sh 'kubectl apply -f ingress.yml'
            })
        }
    }
}

```



stage view

Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN	Docker Build & Push	TRIVY	Deploy to container	Deploy to kubernets
132ms	264ms	1s	25s	295ms	1min 49s	2min 38s	23s	1min 51s	1min 35s	1s	2s
133ms	261ms	1s	25s	284ms	1min 51s	2min 46s	23s	1min 23s	1min 52s	1s	1s

In the Kubernetes cluster give this command



```

kubectl get all
kubectl get svc #use anyone

```



```

ubuntu@ip-172-31-40-131:~$ kubectl get all
NAME                               READY   STATUS    RESTARTS   AGE
pod/petshop-768578655f-kzcd9     1/1     Running   0          43s

NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)      AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP     58m
service/petshop     LoadBalancer 10.104.122.152  <pending>    80:30699/TCP 21m

NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/petshop 1/1       1           1           43s

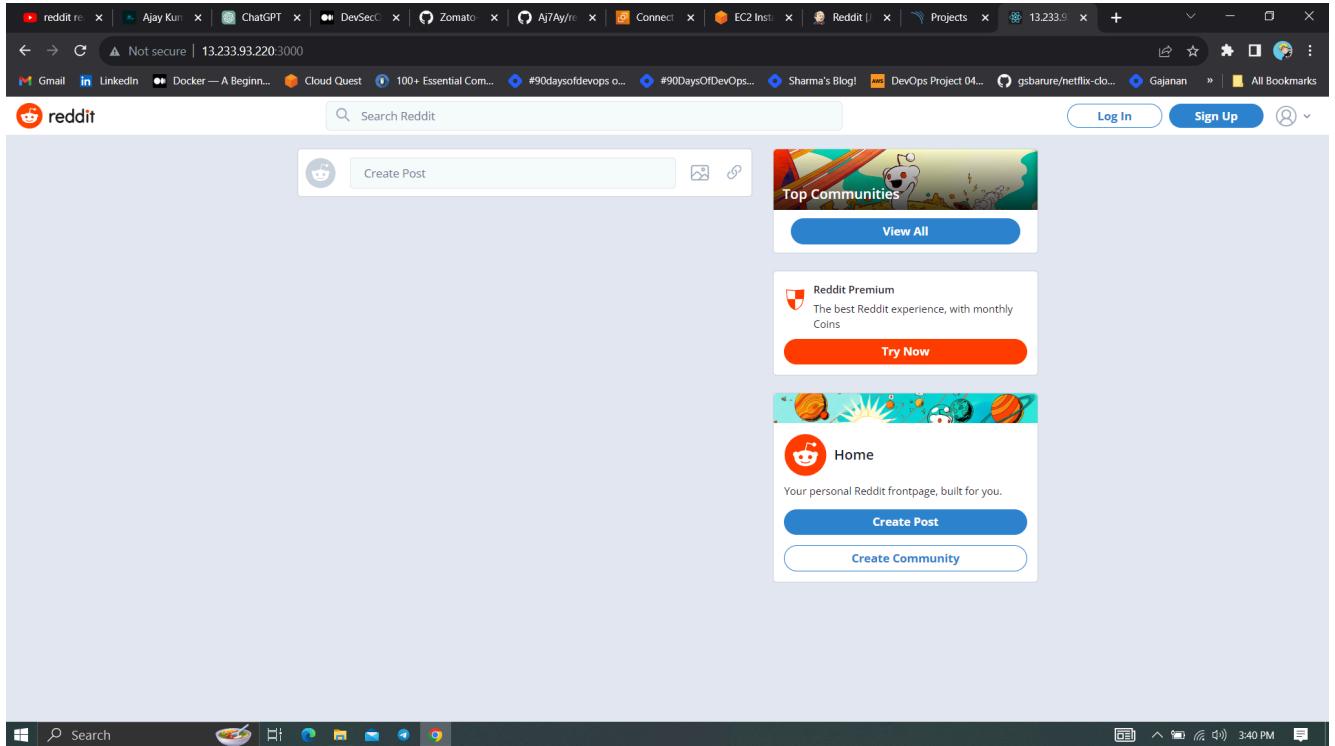
NAME                    DESIRED   CURRENT   READY   AGE
replicaset.apps/petshop-768578655f 1         1         1       43s
ubuntu@ip-172-31-40-131:~$ 

```

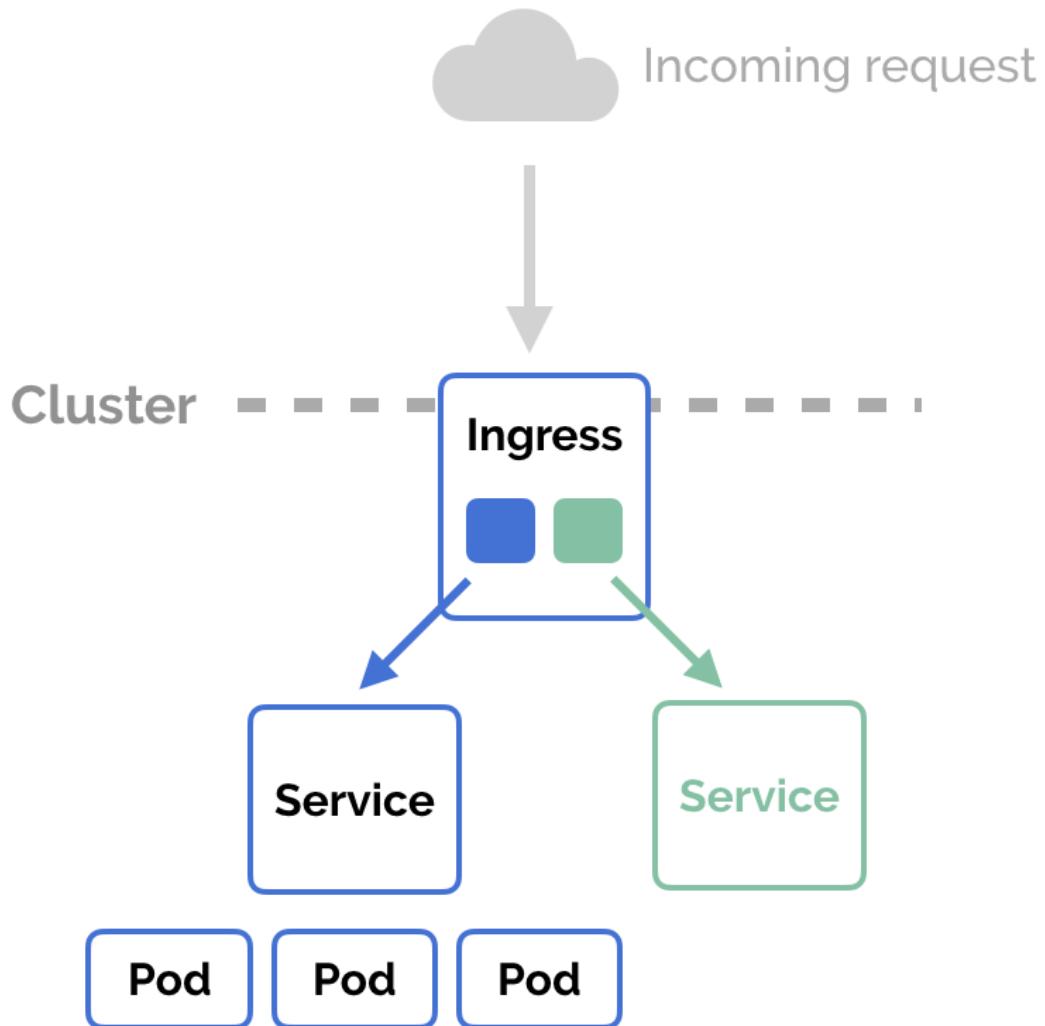
STEP9:Access from a Web browser with

<public-ip-of-slave:service port>

output:



Ingress



1. Go to the browser and type in URL `http://<Public_IP>:3000` or edit the `/etc/hosts` and add the (localhost domain.com/test) to access via URL.

You have successfully Deployed a Reddit Copy on Kubernetes with Ingress Enabled.



Step 10: Terminate instances.

Complete Pipeline

```
pipeline{  
    agent any  
    tools{
```

```
        jdk 'jdk17'
        nodejs 'node16'
    }
environment {
    SCANNER_HOME=tool 'sonar-scanner'
}
stages {
    stage('clean workspace'){
        steps{
            cleanWs()
        }
    }
    stage('Checkout from Git'){
        steps{
            git branch: 'main', url: 'https://github.com/sejal1011/i'
        }
    }
    stage("Sonarqube Analysis"){
        steps{
            withSonarQubeEnv('sonar-server') {
                sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.proje
-Dsonar.projectKey=Reddit '''
            }
        }
    }
    stage("quality gate"){
        steps {
            script {
                waitForQualityGate abortPipeline: false, credential:
            }
        }
    }
    stage('Install Dependencies') {
        steps {
            sh "npm install"
        }
    }
    stage('OWASP FS SCAN') {
        steps {
            dependencyCheck additionalArguments: '--scan ./ --disab
            dependencyCheckPublisher pattern: '**/dependency-check-i
        }
    }
    stage('TRIVY FS SCAN') {
        steps {

```

```
sh "trivy fs . > trivyfs.txt"
    }
}
stage("Docker Build & Push"){
    steps{
        script{
            withDockerRegistry(credentialsId: 'docker', toolName
                sh "docker build -t reddit ."
                sh "docker tag reddit sevenajay/reddit:latest "
                sh "docker push sevenajay/reddit:latest "
            }
        }
    }
}
stage("TRIVY"){
    steps{
        sh "trivy image sevenajay/reddit:latest > trivy.txt"
    }
}
stage('Deploy to container'){
    steps{
        sh 'docker run -d --name reddit -p 3000:3000 sevenajay/reddit'
    }
}
stage('Deploy to kubernets'){
    steps{
        script{
            withKubeConfig(caCertificate: '', clusterName: '', contextName: '')
                sh 'kubectl apply -f deployment.yml'
                sh 'kubectl apply -f service.yml'
                sh 'kubectl apply -f ingress.yml'
            }
        }
    }
}
}
```



Ajay Kumar Yegireddi is a DevSecOps Engineer and System Administrator, with a passion for sharing real-world DevSecOps projects and tasks. **Mr. Cloud Book**, provides hands-on tutorials and practical insights to help others master DevSecOps tools and workflows. Content is designed to bridge the gap between development, security, and operations, making complex concepts easy to understand for both beginners and professionals.

Comments

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.



I'm not a robot

reCAPTCHA
Privacy - Terms[Post Comment](#)

Uncategorized

How to Automate Incident Response : How Q Developer Helped Me Automate a Daily Pain Point

22 July 2025

AI

How to Run Docker Model Runner on Ubuntu 24.04

11 July 2025

AI, DevOps

How to Install docker-ai on Ubuntu 24.04

15 June 2025

Upskill with Ajay: DevSecOps Mastery

Join Mr Cloud book to master DevSecOps through real-world projects. Learn CI/CD, security integration, automation, and more, gaining hands-on skills for industry-level challenges.



Important Links

[Privacy Policy](#)[Terms & Conditions](#)[Contact](#)

Resources

[Blog](#)

YouTube Channel

© 2024 · Powered by [Mr Cloud Book](#)

[Follow Us on YouTube](#)