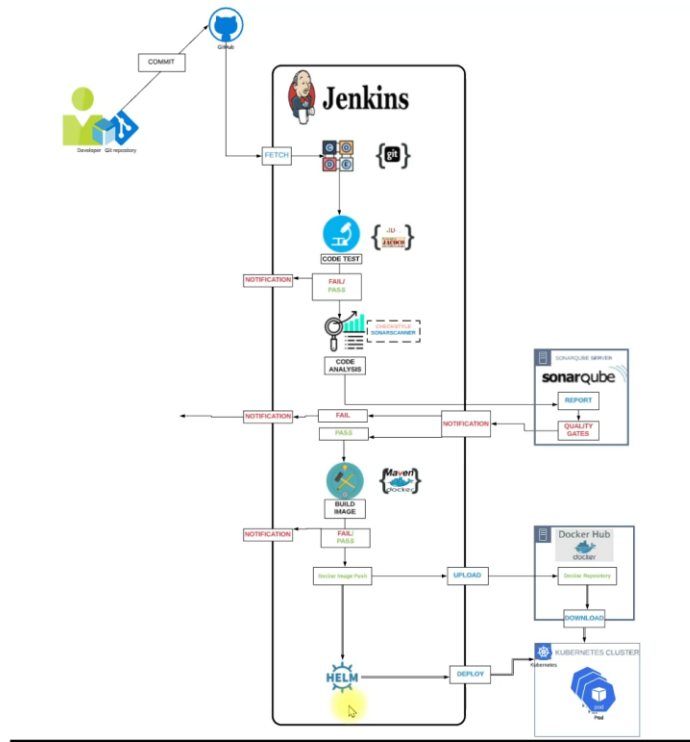


CICD with docker and Kubernetes



This is a CICD pipeline project, when a Developer make a code change and commit to git hub, Jenkins Pipeline will get triggered (Jenkins poll for GitHub changes every minute) and pull the application source code, Docker file, helm charts from GitHub. Jenkins will perform Code Unit test, Code analysis using sonar Qube and the report will be generated. Quality Gates is configured in Sonar qube which helps to limit the pipeline flow based on the number of bugs.

If Quality gate check is passed, docker image will be built and pushed to docker hub. Following the pipeline, Helm charts will be called, It will deploy kubernetes cluster which in turn pull images from Dockerhub, launches the PODS and host the application.

FLOW OF EXECUTION

1. Continuous Integration Setup
 - a. Jenkins, Sonarqube & Nexus (Continuous Integration Project)
2. Dockerhub account (Containerization Project)
3. Store Dockerhub credentials in Jenkins
4. Setup Docker Engine in Jenkins
5. Install Plugins in Jenkins
 - a. Docker-pipeline
 - b. Docker
 - c. Pipeline utility
6. Create Kubernetes Cluster with Kops
7. Install Helm in Kops VM
8. Create Helm Charts
9. Test Charts in K8s Cluster in test namespace..

FLOW OF EXECUTION

10. Add Kops VM as Jenkins Slave
11. Create Pipeline code [Declarative]
12. Update Git Repository with
 - a. Helm Charts
 - b. Dockerfile
 - c. Jenkinsfile (Pipeline code)
13. Create Jenkins job for Pipeline
14. Run & Test the job.

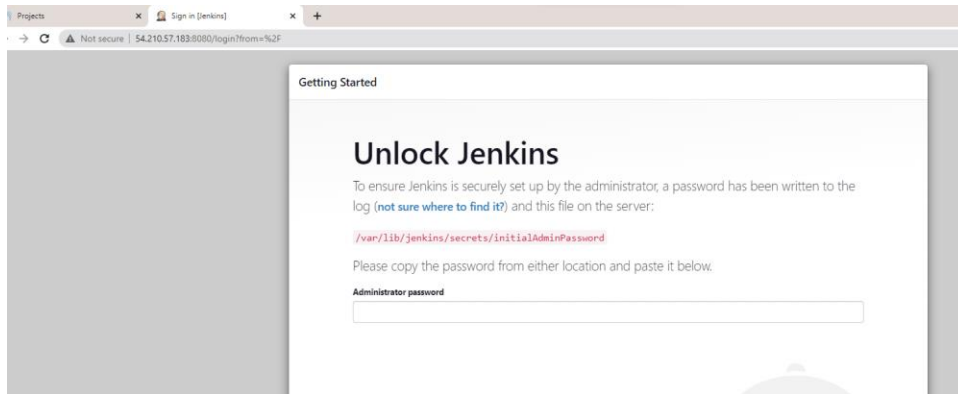
1. Launch the Jenkins server, ubuntu t2 small and refer user data from repo [Jenkins CI Pipeline/jenkins-setup.sh at main · satzwebio/Jenkins CI Pipeline · GitHub](#)

And update the sec group as below,

The screenshot shows the AWS Management Console interface for editing inbound rules of a security group. The breadcrumb navigation at the top reads: EC2 > Security Groups > sg-0021e3468b71e795b - jenkins-sg > Edit inbound rules. The main heading is "Edit inbound rules" with a link to "info". Below this, a note states: "Inbound rules control the incoming traffic that's allowed to reach the instance." The "Inbound rules" section contains a table with the following columns: Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. There are three rules listed:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sg-048f8d2fbd5794e2	Custom TCP	TCP	8080	Anywhere...	
sg-03eade604ad7cac2	SSH	TCP	22	Custom	
-	Custom TCP	TCP	80	Anywhere...	

At the bottom of the table, there is an "Add rule" button. At the bottom right of the console, there are buttons for "Cancel", "Preview changes", and "Save rules".



Get admin pwd from and login jenkins

2. Launch sonar server, ubuntu t2 medium and refer user data from repo [Jenkins_CI_Pipeline/sonar-setup.sh](https://github.com/satzwebio/Jenkins_CI_Pipeline/blob/main/Jenkins_CI_Pipeline/sonar-setup.sh) at main · satzwebio/Jenkins_CI_Pipeline · GitHub

Update the security groups as below,

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

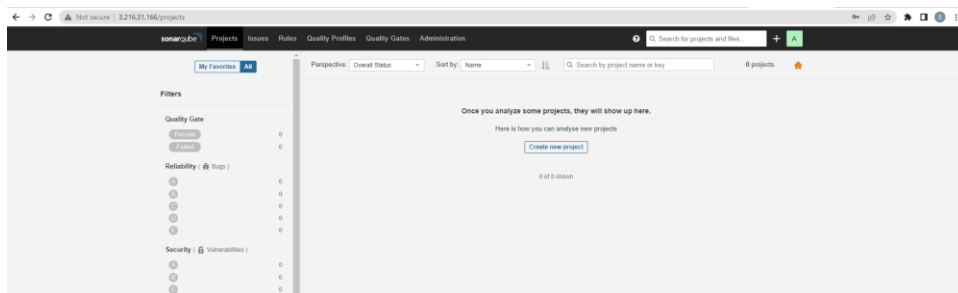
Inbound rules (5)

Filter security group rules

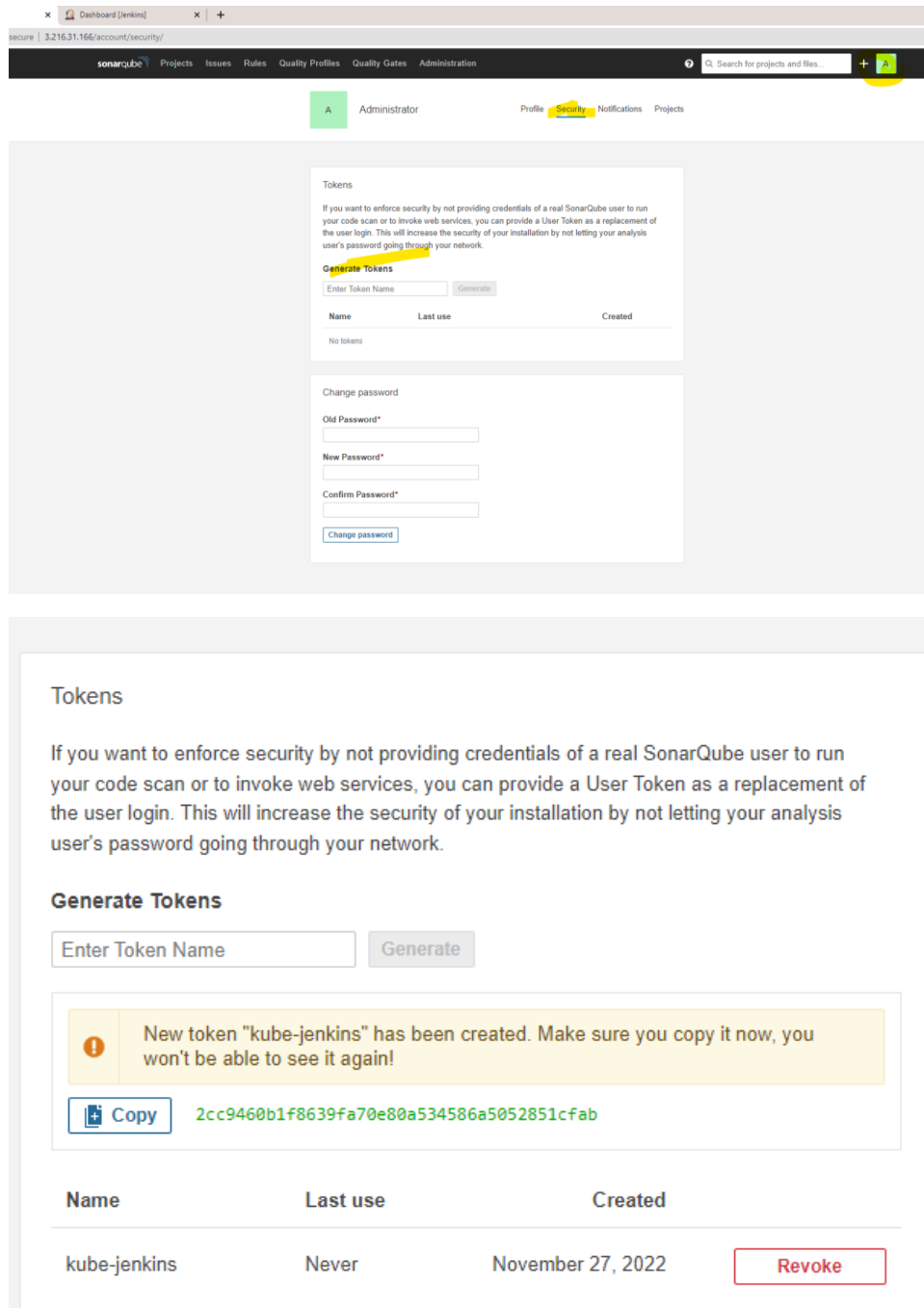
	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgp-04653b8de6c70du9	IPv4	SSH	TCP	22	122.178.75.86/32	-
<input type="checkbox"/>	-	sgp-016b1c2282496c2...	IPv4	Custom TCP	TCP	9000	0.0.0.0/0	-
<input type="checkbox"/>	-	sgp-08f91b56938ffdfcb	IPv4	HTTP	TCP	80	0.0.0.0/0	-

Launch instance, login ssh and check systemctl status sonarqube;

Check in port 80 or 9000, username is both admin



3. Generate new token in Sonar



Tokens

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Enter Token Name

Name	Last use	Created
No tokens		

Change password

Old Password*

New Password*

Confirm Password*

Tokens

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Enter Token Name

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

2cc9460b1f8639fa70e80a534586a5052851cfab

Name	Last use	Created	
kube-jenkins	Never	November 27, 2022	<input type="button" value="Revoke"/>

Token: 2cc9460b1f8639fa70e80a534586a5052851cfab

4. Install the below plugins on Jenkins,

Plugin Manager

Updates

Available

Installed

Advanced

Q

Nexus artif

Install

Name

↓

✓

SonarQube Scanner

2.15

External Site/Tool Integrations

Build Reports

This plugin allows an easy integration of [SonarQube](#), the open source platform for Continuous Inspection of code quality.

✓

Build Timestamp

1.0.3

Build Wrappers

This plugin adds BUILD_TIMESTAMP to Jenkins variables and system properties.

This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

✓

Pipeline Maven Integration

1235.v2db_ddd9f797b

pipeline

Maven

This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The sele be configured and prepended to the path.

✓

Pipeline Utility Steps

2.14.0

pipeline

Build Tools

Miscellaneous

Utility steps for pipeline jobs.

✓

Nexus Artifact Uploader

2.14

Artifact Uploaders

This plugin to upload the artifact to Nexus Repository.

Install without restart

Download now and install after restart

Update information obtained: 23 min ago

Check now

5. In Jenkins, configure sonar qube server, Mention name as sonar-pro, Enter Sonarqube private ip, add sonar qube authentication token.

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

sonar-pro

Server URL

Default is http://localhost:9000

172.31.1.127

Server authentication token


SonarQube authentication token. Mandatory when anonymous access is disabled.


kube-sonar-token

+ Add

Advanced...

Add SonarQube

 **Jenkins Credentials Provider: Jenkins**

 **Add Credentials**

Domain

Global credentials (unrestricted)

Kind

Secret text

Scope

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

Secret

.....

ID

kube-sonar-token

Description

kube-sonar-token

Add

Cancel

6. Update All traffic from Sonar security group to Jenkins and similarly allow all traffic from Jenkins security group to Sonar.

7. Create new credentials in Jenkins, this is to add docker hub creds,

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

New credentials

Kind

Username with password

Scope ?

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

Username ?

satzweb

☐ Treat username as secret ?

Password ?

ID ?

dockerhub

Description ?

dockerhub

Create

8. Install Docker engine on Jenkins, refer <https://docs.docker.com/engine/install/ubuntu/>

```
ubuntu@ip-172-31-53-191:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Sun 2022-11-27 10:01:52 UTC; 41s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 6758 (dockerd)
      Tasks: 9
     Memory: 28.9M
        CPU: 468ms
    CGroup: /system.slice/docker.service
            └─6758 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont

Nov 27 10:01:51 ip-172-31-53-191 dockerd[6758]: time="2022-11-27T10:01:51.62785>
Nov 27 10:01:51 ip-172-31-53-191 dockerd[6758]: time="2022-11-27T10:01:51.62802>
Nov 27 10:01:51 ip-172-31-53-191 dockerd[6758]: time="2022-11-27T10:01:51.71630>
Nov 27 10:01:52 ip-172-31-53-191 dockerd[6758]: time="2022-11-27T10:01:52.17845>
Nov 27 10:01:52 ip-172-31-53-191 dockerd[6758]: time="2022-11-27T10:01:52.32802>
Nov 27 10:01:52 ip-172-31-53-191 dockerd[6758]: time="2022-11-27T10:01:52.39016>
Nov 27 10:01:52 ip-172-31-53-191 dockerd[6758]: time="2022-11-27T10:01:52.39034>
Nov 27 10:01:52 ip-172-31-53-191 systemd[1]: Started Docker Application Contain>
Nov 27 10:01:52 ip-172-31-53-191 dockerd[6758]: time="2022-11-27T10:01:52.42771>
```

9. As a root add Jenkins user to the Jenkins group, and then reboot the machine.

```
ubuntu@ip-172-31-53-191:~$ sudo -i
root@ip-172-31-53-191:~# su - jenkins
jenkins@ip-172-31-53-191:~$ docker images
Got permission denied while trying to connect to the Docker daemon socket at uni
x:///var/run/docker.sock: Get "http://%2Frun%2Fdocker.sock/v1.24/images/js
on": dial unix /var/run/docker.sock: connect: permission denied
jenkins@ip-172-31-53-191:~$ exit
logout
root@ip-172-31-53-191:~# usermod -aG docker jenkins
root@ip-172-31-53-191:~# id jenkins
uid=114(jenkins) gid=121(jenkins) groups=121(jenkins),999(docker)
root@ip-172-31-53-191:~# su - jenkins
jenkins@ip-172-31-53-191:~$ Docker images
Command 'docker' not found, did you mean:
  command 'docker' from snap docker (20.10.17)
  command 'docker' from deb docker.io (20.10.12-0ubuntu4)
  command 'docker' from deb podman-docker (3.4.4+ds1-1ubuntu1)
See 'snap info <snapname>' for additional versions.
jenkins@ip-172-31-53-191:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    feb5d9fea6a5  14 months ago  13.3kB
jenkins@ip-172-31-53-191:~$

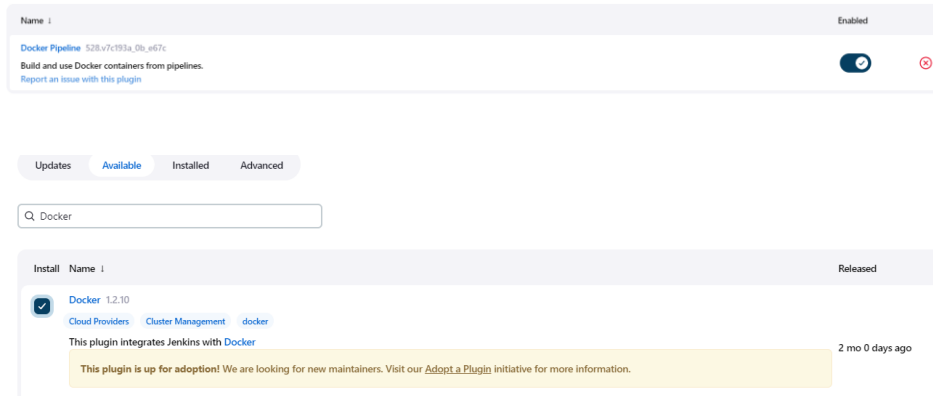
logout
root@ip-172-31-53-191:~# reboot
root@ip-172-31-53-191:~# Connection to 54.210.57.183 closed by remote host.
Connection to 54.210.57.183 closed.

satzw@LAPTOP-C4RG1671 MINGW64 ~/Downloads
$
```


10. Install Below plugin on Jenkins, and install without restart

Docker

Docker Pipeline



11. In KOPS VM download HELM

Get the coplink from <https://github.com/helm/helm/releases>. Check the os Linux amd64

```
ubuntu@ip-172-31-93-244: /tmp/linux-amd64
ubuntu@ip-172-31-93-244:~/kube-app$ cd /tmp/
ubuntu@ip-172-31-93-244: /tmp$ wget https://get.helm.sh/helm-v3.10.2-linux-amd64.tar.gz
--2022-11-27 10:16:18-- https://get.helm.sh/helm-v3.10.2-linux-amd64.tar.gz
Resolving get.helm.sh (get.helm.sh)... 152.195.19.97, 2606:2800:11f:1cb7:261b:1f9c:2074:3c
Connecting to get.helm.sh (get.helm.sh)|152.195.19.97|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14564021 (14M) [application/x-tar]
Saving to: 'helm-v3.10.2-linux-amd64.tar.gz'

helm-v3.10.2-linux-am 100%[=====>] 13.89M 6.01MB/s in 2.3s

2022-11-27 10:16:21 (6.01 MB/s) - 'helm-v3.10.2-linux-amd64.tar.gz' saved [14564021/14564021]

ubuntu@ip-172-31-93-244: /tmp$ tar xzvf helm-v3.10.2-linux-amd64.tar.gz
linux-amd64/
linux-amd64/helm
linux-amd64/LICENSE
linux-amd64/README.md
ubuntu@ip-172-31-93-244: /tmp$ cd linux-amd64/
ubuntu@ip-172-31-93-244: /tmp/linux-amd64$ ls
LICENSE README.md helm
ubuntu@ip-172-31-93-244: /tmp/linux-amd64$ sudo mv helm /usr/local/bin/helm
ubuntu@ip-172-31-93-244: /tmp/linux-amd64$ helm --help
The Kubernetes package manager
```

From here the steps are option. YOu can clone the <https://github.com/imranvisualpath/cicd-kube-docker.git>

12. Get into KOPS Vm, and clone the repository

```
Last login: Sun Nov 27 09:23:24 2022 from 122.178.75.86
ubuntu@ip-172-31-53-191:~$ git clone https://github.com/devopshydc1ub/vprofile-p
project.git
Cloning into 'vprofile-project'...
remote: Enumerating objects: 2620, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 2620 (delta 0), reused 13 (delta 0), pack-reused 2605
Receiving objects: 100% (2620/2620), 75.53 MiB | 29.12 MiB/s, done.
Resolving deltas: 100% (995/995), done.
ubuntu@ip-172-31-53-191:~$ ls
vprofile-project
ubuntu@ip-172-31-53-191:~$ cd vprofile-project/
ubuntu@ip-172-31-53-191:~/vprofile-project$ git checkout vp-docker
Branch 'vp-docker' set up to track remote branch 'vp-docker' from 'origin'.
Switched to a new branch 'vp-docker'
ubuntu@ip-172-31-53-191:~/vprofile-project$ ls
Docker-app  Docker-web  ansible  helm      pom.xml
Docker-db   README.md   compose  kubernetes  src
ubuntu@ip-172-31-53-191:~/vprofile-project$
```

13. Copy all the files from vprofile to Cidc directory created on home, remove unwanted files.

```
ubuntu@ip-172-31-53-191:~$ mkdir cidc-kube-app
ubuntu@ip-172-31-53-191:~$ cd vprofile-project/
ubuntu@ip-172-31-53-191:~/vprofile-project$ ls
Docker-app  Docker-web  ansible  helm      pom.xml
Docker-db   README.md   compose  kubernetes  src
ubuntu@ip-172-31-53-191:~/vprofile-project$ cp -r * ../cidc-kube-app
ubuntu@ip-172-31-53-191:~/vprofile-project$ cd ..
ubuntu@ip-172-31-53-191:~$ cd cidc-kube-app/
ubuntu@ip-172-31-53-191:~/cidc-kube-app$ ls
LS: command not found
ubuntu@ip-172-31-53-191:~/cidc-kube-app$ ls
Docker-app  Docker-web  ansible  helm      pom.xml
Docker-db   README.md   compose  kubernetes  src
ubuntu@ip-172-31-53-191:~/cidc-kube-app$ mv Docker-app/Dockerfile .
ubuntu@ip-172-31-53-191:~/cidc-kube-app$ ls
Docker-app  Docker-web  README.md  compose  kubernetes  src
Docker-db   Dockerfile  ansible    helm      pom.xml
ubuntu@ip-172-31-53-191:~/cidc-kube-app$ rm -rf Docker-app Docker-db Docker-web
compose ansible
```

Remove Helm directory as well.

```
Docker-app  Dockerfile  README.md  helm  kubernetes  pom.
ubuntu@ip-172-31-93-244:~/cidc-kube-app$ rm -rf Docker-app
ubuntu@ip-172-31-93-244:~/cidc-kube-app$ ls
Dockerfile  README.md  helm  kubernetes  pom.xml  src
ubuntu@ip-172-31-93-244:~/cidc-kube-app$ rm -rf helm
ubuntu@ip-172-31-93-244:~/cidc-kube-app$
```

14. Create a helm directory and create helm charts. Then move all the deployment, service files to helm/templates directory.

```
ubuntu@ip-172-31-93-244: ~/cicd-kube-app
ubuntu@ip-172-31-93-244:~/cicd-kube-app$ mkdir helm
ubuntu@ip-172-31-93-244:~/cicd-kube-app$ cd helm
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm$ helm create vprofilecharts
Creating vprofilecharts
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm$ cd vprofilecharts/
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm/vprofilecharts$ ls
Chart.yaml  charts  templates  values.yaml
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm/vprofilecharts$ cd templates/
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm/vprofilecharts/templates$ ls
NOTES.txt  deployment.yaml  ingress.yaml  serviceaccount.yaml
_helpers.tpl  hpa.yaml  service.yaml  tests
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm/vprofilecharts/templates$ rm -rf*
rm: invalid option -- '*'
Try 'rm --help' for more information.
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm/vprofilecharts/templates$ rm -rf *
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm/vprofilecharts/templates$ ls
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm/vprofilecharts/templates$ cd ../../
ubuntu@ip-172-31-93-244:~/cicd-kube-app/helm$ cd ..
ubuntu@ip-172-31-93-244:~/cicd-kube-app$ ls
Dockerfile  README.md  helm  kubernetes  pom.xml  src
ubuntu@ip-172-31-93-244:~/cicd-kube-app$ cp kubernetes/vpro-app/* helm/vprofilecharts/templates/
ubuntu@ip-172-31-93-244:~/cicd-kube-app$ ls helm/vprofilecharts/templates/
app-secret.yml  mc-CIP.yml  rmq-CIP-service.yml  vproapp-service.yml  vprodbdep.yml
db-CIP.yml      mcdep.yml   rmq-dep.yml         vproappdep.yml
```

Vim Vproappdep.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: vproapp
  labels:
    app: vproapp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: vproapp
  template:
    metadata:
      labels:
        app: vproapp
    spec:
      containers:
        - name: vproapp
          image: {{.Values.appimage}}
          ports:
            - name: vproapp-port
              containerPort: 8080
      initContainers:
        - name: init-mysql
          image: busybox
"vproappdep.yml" 29L, 699B
```

Just for testing we created a test namespace and launched the stacks. It creates deployment, service stc...

```

ubuntu@ip-172-31-93-244:~/cicd-kube-app$ helm install --namespace test vprofile-stack helm/vprofilecharts --set appimage=imranvisualpath/vproappdock:9
NAME: vprofile-stack
LAST DEPLOYED: Sun Nov 27 11:15:04 2022
NAMESPACE: test
STATUS: deployed
REVISION: 1
TEST SUITE: None
ubuntu@ip-172-31-93-244:~/cicd-kube-app$ helm list --namespace test
NAME                NAMESPACE    REVISION    UPDATED                               S
TATUS  CHART
vprofile-stack test      1           2022-11-27 11:15:04.286339778 +0000 UTCd
eployed vprofilecharts-0.1.0 1.16.0
ubuntu@ip-172-31-93-244:~/cicd-kube-app$ kubectl get all --namespace test
NAME                                READY    STATUS    RESTARTS   AGE
pod/vproapp-5d56b6469d-1kcscc       1/1      Running   0           42s
pod/vprodb-77668447fc-jvlpw         1/1      Running   0           42s
pod/vpromc-7db9bfbd6d-cthkcd        1/1      Running   0           42s
pod/vpromq01-5bd75bf4bc-p6jfd       1/1      Running   0           42s

NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP    PORT(S)          A
GE
service/vproapp-service             LoadBalancer        100.70.207.153  <pending>      80:31312/TCP     4

```

To uninstall the stacks,

```

ubuntu@ip-172-31-93-244:~/cicd-kube-app$ helm delete vprofile-stack --namespace test
release "vprofile-stack" uninstalled
ubuntu@ip-172-31-93-244:~/cicd-kube-app$

```

Create a prod namespace:

```

release "vprofile-stack" uninstalled
ubuntu@ip-172-31-93-244:~/cicd-kube-app$ kubectl create namespace prod
namespace/prod created
ubuntu@ip-172-31-93-244:~/cicd-kube-app$

```

Create jenkinsfile

Refer our repo : <https://github.com/satzwebio/cicd-kube-docker.git>

In KOPS VM, create jenkins-slave directory and install jdk

```

cicd-kube-docker vprofile project
ubuntu@ip-172-31-32-194:~$ mkdir jenkins-slave
ubuntu@ip-172-31-32-194:~$ sudo apt install openjdk-8-jdk -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-8-jdk is already the newest version (8u272-b10-0ubuntu1~20.04).
0 upgraded, 0 newly installed, 0 to remove and 59 not upgraded.
ubuntu@ip-172-31-32-194:~$

```

In addition perform below,

```
sudo apt remove openjdk-8-jdk -y
```

```
sudo apt purge openjdk-8-jdk -y
```

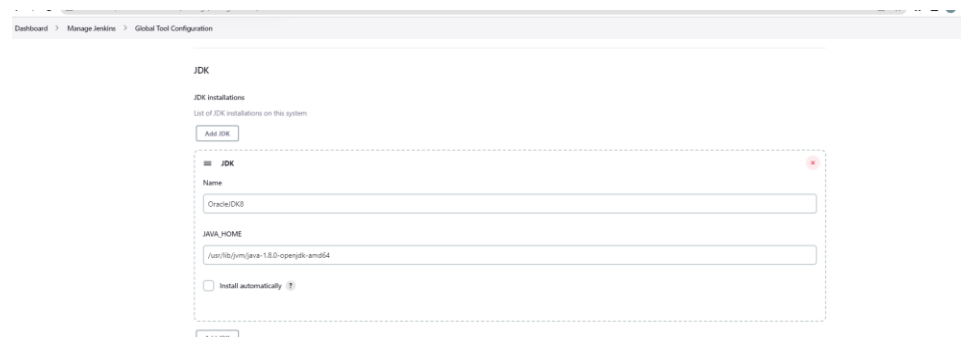
```
sudo apt update
```

```
sudo apt install openjdk-11-jdk -y
```

One more Java package need to be installed for Jenkins to access

```
ubuntu@ip-172-31-20-8:~$ sudo apt install openjdk-8-jdk -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-8-jdk is already the newest version (8u342-b07-0ubuntu1~20.04).
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
ubuntu@ip-172-31-20-8:~$ java -version
openjdk version "11.0.16" 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)
ubuntu@ip-172-31-20-8:~$ sudo -i
root@ip-172-31-20-8:~# ls /usr/lib/jvm
java-1.11.0-openjdk-amd64  java-1.8.0-openjdk-amd64  java-11-openjdk-amd64  java-8-openjdk-amd64  openjdk-11
root@ip-172-31-20-8:~# /usr/lib/jvm/java-1.8.0-openjdk-amd64
```

And specify the path in Jenkins,



Then add MAVEN,

Maven

Maven installations

List of Maven installations on this system

[Add Maven](#)

Maven

Name

MAVEN3

☒ Install automatically ?

Install from Apache

Version

3.8.6

[Add Installer](#)

[Add Maven](#)

[Save](#) [Apply](#)

Create opt/jenkins-slave directory and give ubuntu permission.

So jenkins master will have an agent in this directory.

Jenkins will ssh into this VM with ubuntu user, so ubuntu should own this directory

```
ubuntu@ip-172-31-93-244:~$ sudo mkdir /opt/jenkins-slave
ubuntu@ip-172-31-93-244:~$ sudo chown ubuntu:ubuntu /opt/jenkins-slave -R
```

Update security group, allow ssh from Jenkins to KOPS vm,

EC2 > **Security Groups** > **sg-0e10d4883c3acff - launch-wizard-2** > **Edit inbound rules**

Edit inbound rules


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

Inbound rules	Type	Protocol	Port range	Source	Description - optional
sg-09c5b407d761c7d09	SSH	TCP	22	Custom	122.178.75.46/32
-	SSH	TCP	22	Custom	122.178.75.46/32

[Add rule](#)

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

In Jenkins, create a new node

 **Jenkins**

Dashboard > Manage Jenkins > Nodes >

↑ Back to Dashboard

⚙️ Manage Jenkins

+

 New Node

☁️ Configure Clouds

⚙️ Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

New node

Node name

kops

Type

•

 Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide high dynamic provisioning. Select this type if no other agent types apply — for example such as when you are managed outside Jenkins, etc.

Create

Name ?

kops

Description ?

Number of executors ?

1

Remote root directory ?

/opt/jenkins-slave

Labels ?

KOPS

Usage ?

Only build jobs with label expressions matching this node

Launch method ?

Launch agents via SSH

Host ?

172.31.93.244

Usage ?
Only build jobs with label expressions matching this node

Launch method ?
Launch agents via SSH

Host ?
172.31.93.244

Credentials ?
- none -
+ Add

Host Key Verification Strategy ?
Known hosts file Verification Strategy ?
Advanced...

Availability ?
Keep this agent online as much as possible ?

Node Properties

☐ Disable deferred wipeout on this node ?
☐ Force reconnect

Add credentials, enter KOPS private key

Domain
Global credentials (unrestricted)

Kind
SSH Username with private key

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

ID ?
kops-login

Description ?

Username
ubuntu

☐ Treat username as secret ?

Private Key
Enter directly

Select the cred and host verification strategy

Credentials ?
ubuntu
+ Add

Host Key Verification Strategy ?
Non-verifying Verification Strategy ?
Advanced...

Availability ?

\

[Dashboard](#) > [Manage Jenkins](#) > [Global Tool Configuration](#)

Create a new project in sonar qube,

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

Search for projects and files...

vprofile-repo

master

OverviewIssuesSecurity HotspotsMeasuresCodeActivityProject SettingsProject Info

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

1

Provide a token

Generate a token

Enter a name for your token

Generate

Use existing token

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your [user account](#).

2

Run analysis on your project

In project setting under webhook, create below..

Provide private ip of Jenkins.

vprofile-repo

master

OverviewIssuesSecurity HotspotsMeasuresCodeActivityProject SettingsProject Information

Last analysis had 2 warnings

November 27, 2022 at 6:11 PM

Version 1.0

Webhooks

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](#).

Create


Name	URL	Secret?	Last delivery
jenkins-webhook	http://172.31.53.191:8080/sonarqube-webhook	No	Never

Create a pipeline project in jenkins,


Dashboard > All >

Enter an item name


» Required field

 **Freestyle project**


This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can

 **Pipeline**


Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as 1

 **Multi-configuration project**


Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform

 **Folder**

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a fol
are in different folders.

 **Multibranch Pipeline**

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

 **Organization Folder**

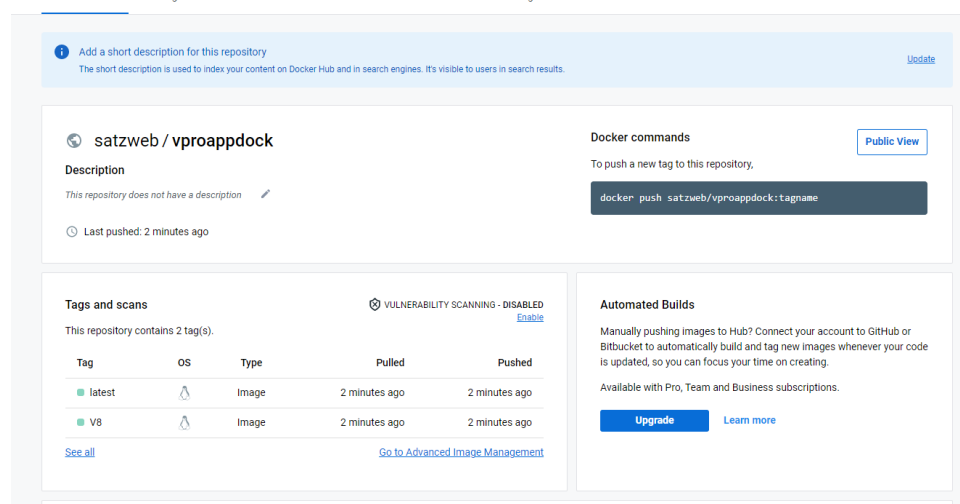
Creates a set of multibranch project subfolders by scanning for repositories.

OK

Pipeline finished successfully.



In the docker hub, we can see the images,



We can see the unused images are removed from the docker hub

```

[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Remove Unused docker image)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker rmi satzweb/vproappdock:V8
Untagged: satzweb/vproappdock:V8
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Kubernetes Deploy)
[Pipeline] node
Running on kops in /opt/jenkins-slave/workspace/kube-cicd
[Pipeline] {
[Pipeline] checkout

```

Can see the helm is installing the required stack,

```

[Pipeline] sh
+ helm upgrade --install --force vprofile-stack helm/vprofilecharts --set appimage=satzweb/vproappdock:V8 --namespace prod
Release "vprofile-stack" does not exist. Installing it now.
NAME: vprofile-stack
LAST DEPLOYED: Sun Nov 27 12:53:22 2022
NAMESPACE: prod
STATUS: deployed
REVISION: 1
TEST SUITE: None
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }

```

```

ubuntu@ip-172-31-93-244: ~
ubuntu@ip-172-31-93-244:~$ helm list --namespace prod
NAME                NAMESPACE    REVISION    UPDATED                               STATUS    CHART
vprofile-stack      prod          1           2022-11-27 12:53:22.501203262 +0000 UTC    deployed vprofilecharts-0.1.0
vprofile-stack      prod          1           2022-11-27 12:53:22.501203262 +0000 UTC    1.16.0
ubuntu@ip-172-31-93-244:~$ kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
vproapp-d65c4d4f8-476dd             1/1     Running   0            4h9m
vprodb-8d5b8f99d-wg6j5              1/1     Running   0            4h56m
vpromc-7db9bfbd6d-kwsqc             1/1     Running   0            4h9m
vpromq01-5bd75bf4bc-wg6v4           1/1     Running   0            4h9m

```

```

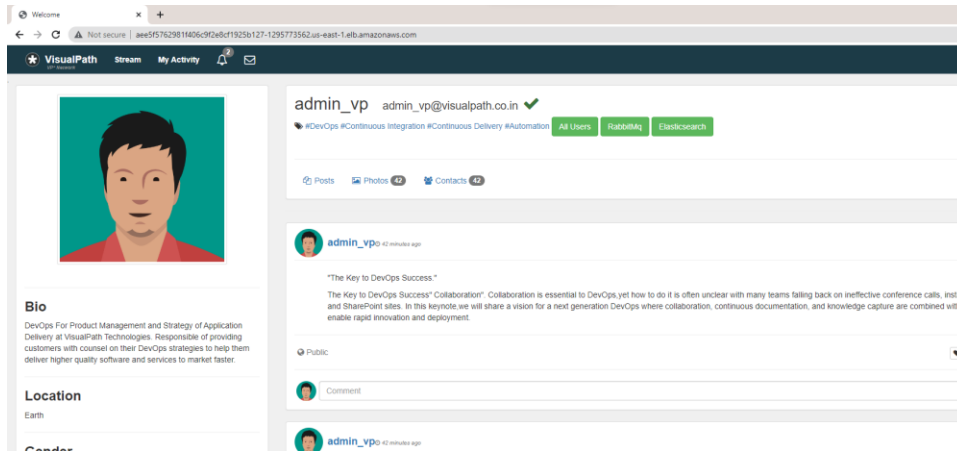
ubuntu@ip-172-31-93-244: ~
ubuntu@ip-172-31-93-244:~$ kubectl get pod --namespace prod
NAME                                READY    STATUS    RESTARTS    AGE
vproapp-866dc8bd79-2qdr9            1/1     Running   0            9m10s
vprodb-77668447fc-zsgcw             1/1     Running   0            9m10s
vpromc-7db9bfbd6d-7pkqh            1/1     Running   0            9m10s
vpromq01-5bd75bf4bc-bhcfp           1/1     Running   0            9m10s
ubuntu@ip-172-31-93-244:~$ kubectl describe pod vproapp-866dc8bd79-2qdr9 --namespace prod
Name:                               vproapp-866dc8bd79-2qdr9
Namespace:                          prod
Priority:                             0
Service Account:                     default
Node:                                i-08c036cad52606513/172.20.59.115
Start Time:                          Sun, 27 Nov 2022 12:53:22 +0000
Labels:                              app=vproapp
                                      pod-template-hash=866dc8bd79
Annotations:                         <none>
Status:                              Running
IP:                                  100.96.2.11
IPs:                                 IP: 100.96.2.11
Controlled By:                       ReplicaSet/vproapp-866dc8bd79
Init Containers:
  init-mysql:
    Container ID:  containerd://69832ba99c5011c6e723f58d0e96b53b2493f53c3e6ddf9af92a37fe980

```

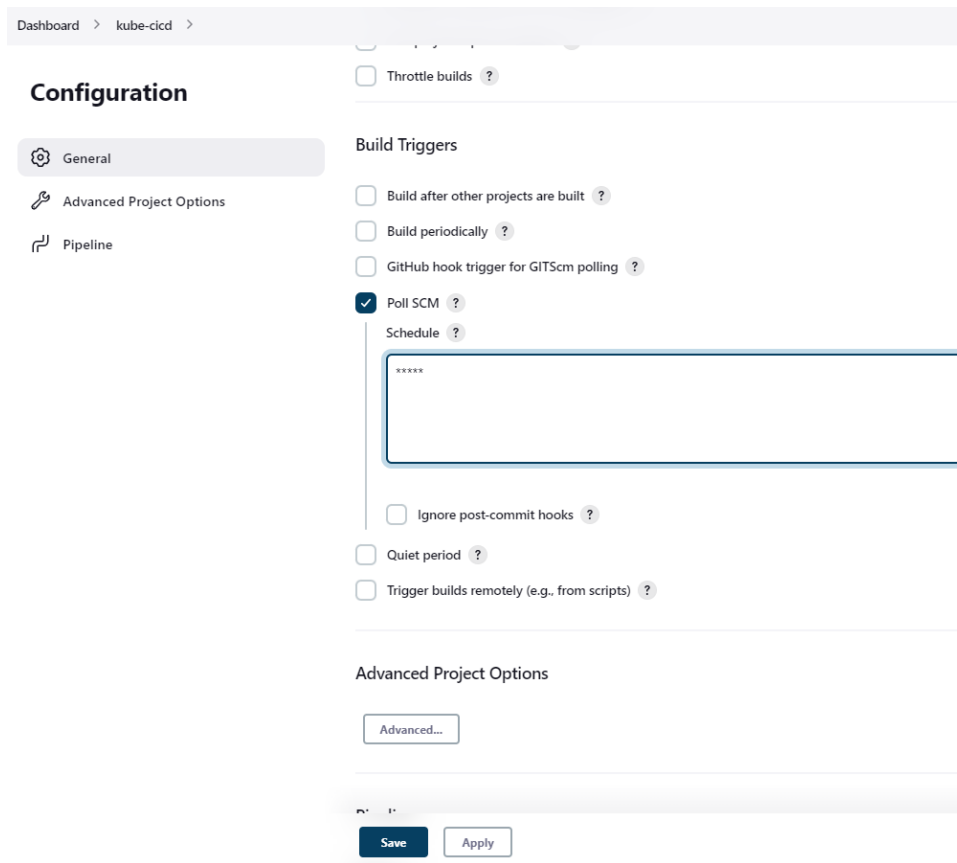
```

ubuntu@ip-172-31-93-244: ~
ubuntu@ip-172-31-93-244:~$ kubectl get svc --namespace prod
NAME                TYPE                CLUSTER-IP    EXTERNAL-IP    AGE
vproapp-service     LoadBalancer      100.69.111.252    aee5f5762981f406c9f2e8cf1925b127-1295773562.us-east-1.elb.amazonaws.com    10m
vprocache01         ClusterIP           100.64.126.101    <none>          10m
vprodb              ClusterIP           100.65.154.112    <none>          10m
vpromq01            ClusterIP           100.71.113.32     <none>          10m

```



Instead of Build manually in jenkins, we can update it to Poll SCM, so whenever a change in git repo, a new build will get triggered and deploy the stack to kuberntes.




```
ubuntu@ip-172-31-93-244: ~  
ubuntu@ip-172-31-93-244:~$ kops delete cluster --name kops.satzwebio.com --state=s3://sa  
tz-kops-bucket --yes  
w1127 13:11:06.361863 10850 aws.go:2250] (new) cluster tag not found on volume:vol-039  
fe83f2eed5ccf1  
TYPE NAME ID  
autoscaling-config master-us-east-1a.masters.kops.satzwebio.com 1t-02d9f92858d24  
cbf8  
autoscaling-config nodes-us-east-1a.kops.satzwebio.com 1t-04a792daf8cd4  
0a41  
autoscaling-group master-us-east-1a.masters.kops.satzwebio.com master-us-east-1  
a.masters.kops.satzwebio.com  
autoscaling-group nodes-us-east-1a.kops.satzwebio.com nodes-us-east-1a  
.kops.satzwebio.com  
dhcp-options kops.satzwebio.com dopt-05513f795f4  
373891  
iam-instance-profile masters.kops.satzwebio.com masters.kops.satzwebio.com
```