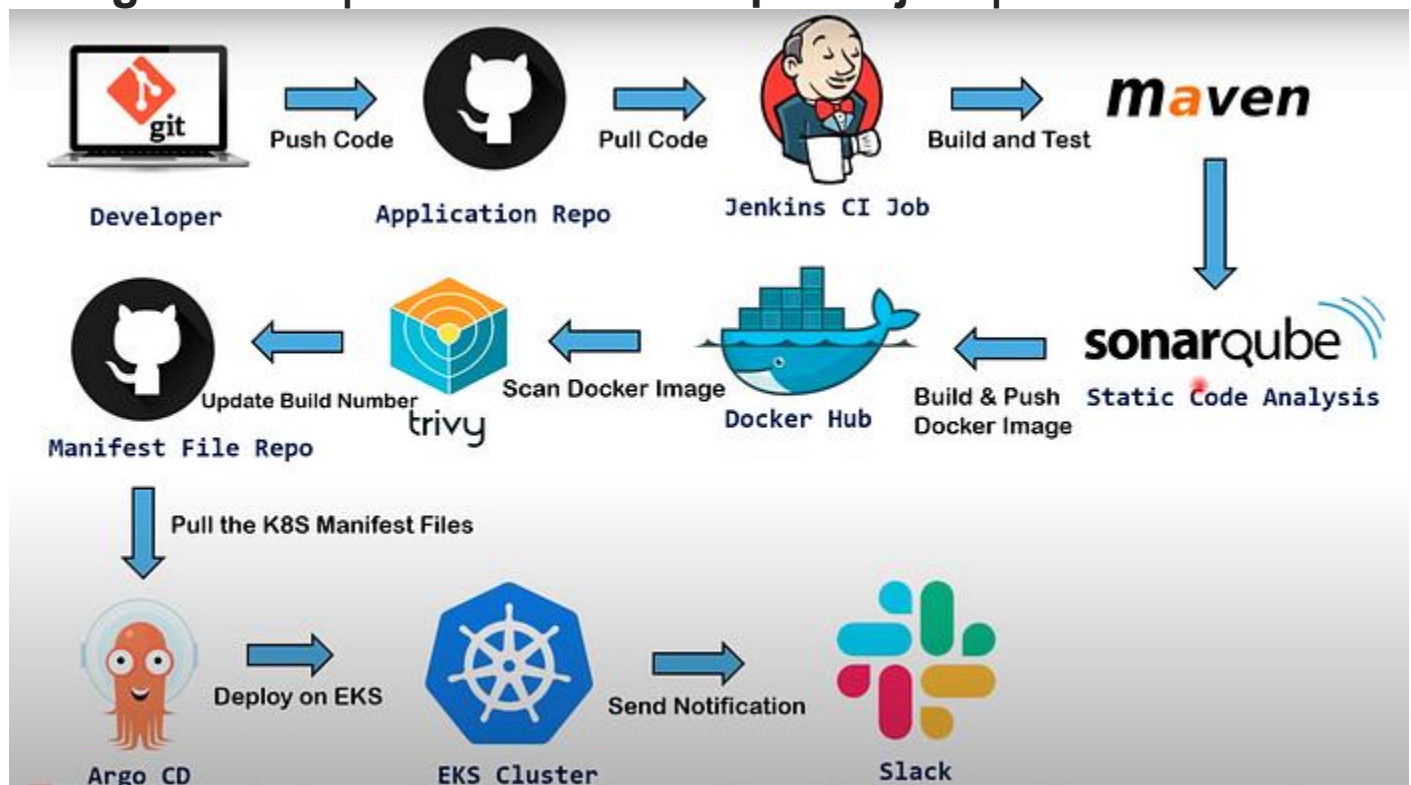


Deploy Register website

Real Time DevOps Project | Deploy to Kubernetes Using Jenkins | End to End DevOps Project | CI/CD

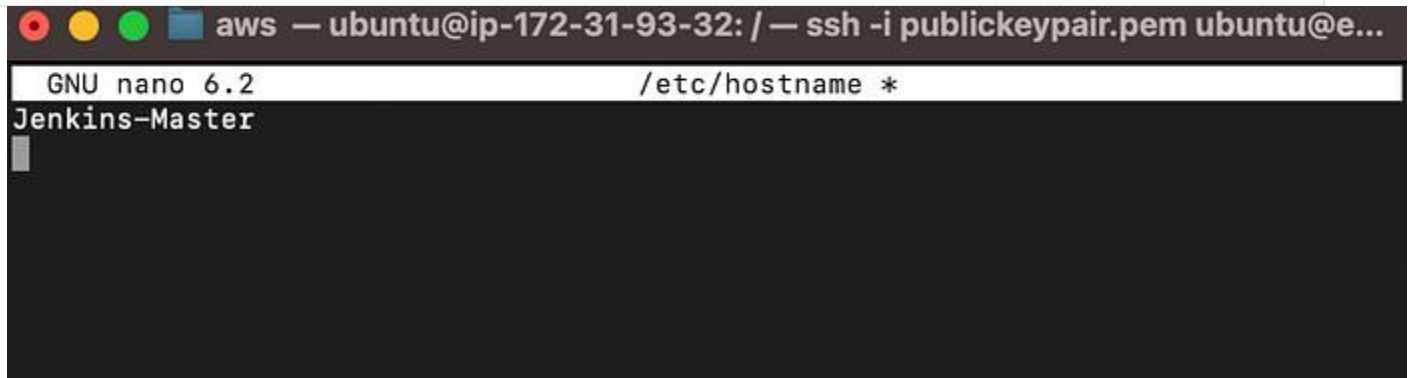


Step 1: Server 1- Jenkins-Master

Make one server.

Change the hostname IP address to name.

```
sudo nano /etc/hostname
```



Once we update the name, we will reboot our system.

```
sudo init 6
```

Install Jenkins on your server, and as you know, Jenkins can not run without JAVA. So, JAVA installation is a must.

For installation, read my DevOps tools installation from the below link.

[Jenkins Installation](#)

Now you have to uncomment two lines in sshd_config, So write a command

```
sudo nano /etc/ssh/sshd_config
```

```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none
```

Reload service

```
sudo service sshd reload
```

Generate SSH key

```
ssh-keygen
```

Copy the ssh pub key

```
ubuntu@Jenkins-Master:~$ pwd
/home/ubuntu
ubuntu@Jenkins-Master:~$ cd .ssh/
ubuntu@Jenkins-Master:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@Jenkins-Master:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQDRd4dZzufck1SQRtsrRgbWTMJe1cB4aVdUJ2PptH4ILrqHtS+r5FSD1q5vawdI
CMV7uJEQZ/7Yi7LrIk3ts089mBuxmRMY+PoXf0nnSYkIinRR3cGd52aZoY7TBQs1Zt8Ilry7BgItsA54E/7ycYxA0nZyCcnQmSh
YdpcY2WQCq8FweJa0wWNhev2ypcuAFpIJQqzmTj7rs7F+JNEJ45L3dTvTvwnYv4GDfoVhIIN1GNxTVbIovwvDGn9aBvXUTfrOcQS
P+zT9Fk6RwQs960BnUqcxN3Ds9/+hcwToSWSjFpXdrWhM1S1VTQueCNXNbYA6khKA7d8d6c0y/pTiiPsaDjCG3xcfbk6fWqQqQGQ
Exj/csr1goCBRpuFtqTKugqqyBnEWIiNebNh/Dj76Eil/YBZboE5JVNFxi4be8u91AHldbNgIGUtqbqnbxZ6/gidF2EinmgyAt2N
iyB3ETv3HpTENEo/EYpw9Y/ktUmRQJPHUtH1Z0Bfc= ubuntu@Jenkins-Master
ubuntu@Jenkins-Master:~/.ssh$
```

Open Jenkins with server IP :8080

In Jenkins, open Dashboard > Manage Jenkins > Nodes

Under Node > Built in Node > Configure

Dashboard > Nodes > Built-In Node > Configure

Status

Configure

Build History

Load Statistics

Script Console

Build Executor Status

1 idle

2 idle

Number of executors ?

0

Labels ?

Usage ?

Use this node as much as possible

Node Properties

☐ Disable deferred wipeout on this node ?

☐ Environment variables

Save

Save and go to Dashboard again.

Dashboard > Manage Jenkins > Nodes > New node

New node

Node name

Type



Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because it provides a higher level of integration with these agents, such as dynamic provisioning and node types apply — for example such as when you are adding a physical node outside Jenkins, etc.

Create

Name ?

Jenkins-Agent

Description ?

Jenkins-Agent



Plain text [Preview](#)

Number of executors ?

1

Remote root directory ?

/home/ubuntu

Labels ?

Jenkins-Agent

Usage ?

Use this node as much as possible

Launch method ?

Launch agents via SSH

Host ?

172.31.80.4

Jenkins-Agent Server private IP Address

Credentials ?

ubuntu (Jenkins-Agent)

Check below image, you will get credentials

+ Add

Host Key Verification Strategy ?

Non verifying Verification Strategy

Advanced

Availability ?

Keep this agent online as much as possible

Save

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

SSH Username with private key

Scope ?

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

ID ?

Jenkins-Agent

Description ?

Jenkins-Agent

Username

Username

ubuntu

☐ Treat username as secret ?

Private Key

☒ Enter directly

Key

Add Jenkins-Master Private SSH key

Enter New Secret Below

```
-----BEGIN OPENSSH PRIVATE KEY-----  
b3B1bnNzaC1rZXktdjEAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn  
NhAAAAAwEAAQAAAEFA0XehWc7n3JJUkEbbK0YG1kzCxtXAeGLXVCdi6bR+CC66h7Uyo+RU
```


Passphrase


Add

Cancel

Start new pipeline

Test-Pipeline
» 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 be even used for something other than software build.

 **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.

Dashboard > Test-Pipeline > Configuration

Configure

General
Advanced Project Options
Pipeline

Pipeline

Definition
Pipeline script

Script ?

```
1 - pipeline {  
2   agent any  
3  
4   stages {  
5     stage('Hello') {  
6       steps {  
7         echo 'Hello World'  
8       }  
9     }  
10  }  
11 }  
12
```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

Save Apply

Test Jenkins

Build now.

Dashboard > Manage Jenkins > Plugins

-> Maven Integration in Jenkins.

→ Pipeline Maven Integration

→ Eclipse Temurin Installer

Add Maven

≡ Maven

Name

Maven3

! Required

☒ Install automatically ?

≡ Install from Apache

Version

3.9.6

Add Installer ▾

Add Maven

Save

Apply

JDK installations

Add JDK

≡ JDK

Name

Java17

☒ Install automatically ?

≡ Install from adoptium.net ?

Version ?

jdk-17.0.8.1+1 ▼

Add Installer ▼

Save and start add-in (git-hub)

New credentials

Kind

Username with password

Scope ?

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

Username ?

mihirmodi2561

Github ID

☐ Treat username as secret ?

Password ?

.....

Github Token

ID ?

github

Description ?

Create

Step 2: Server 2- Jenkins-Agent

Like Jenkins-Master, we must make another Jenkins-Agent server and change the **hostname**.

Install JAVA from DevOps tools installation → [JAVA](#)
[INSTALLATION](#).

Install Docker from DevOps tools installation → [Docker](#)
[INSTALLATION](#).

Give a rights docker to **Jenkins-Agent user**.

```
sudo usermod -aG docker $USER
```

Restart server:

```
sudo init 6
```

Now you have to uncomment two lines in sshd_config, So write a command

```
sudo nano /etc/ssh/sshd_config
```

```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none
```

Reload service

```
sudo service sshd reload
```

Paste the ssh pub key.

```

[ubuntu@Jenkins-Agent:~$ pwd
/home/ubuntu
[ubuntu@Jenkins-Agent:~$ cd .ssh/
[ubuntu@Jenkins-Agent:~/.ssh$ ls
authorized_keys
[ubuntu@Jenkins-Agent:~/.ssh$ nano authorized_keys
[ubuntu@Jenkins-Agent:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCSNcbvVWjiSTZfkf+pHY47dSnwD8ZxDdABA6NM/2eFzQID1rSATTNYaIvXr
zqTWt/nYvcdk7PVM/cbpgIZQmV4WmiC2dxt2LM4shR9S5kMeF5pzo03437ym/woAwAB498c7KfAhWXNdeDDpt5I0nm434j0ss
HmKzZbPGiTLHmbP6UCR9pZXohj60TJLUnF81SHbqr5o1ReiqbJfDvdfoe7ONhhGsZuGHsm5trPYtVWqNFwY28Yli01x31PMBs
NuDuYiXhakAT8BSMU1ISgYGzqwVD2bkrB/S1YTuc25fahWWjTJvxQ3+A4IK0Wdlr0efW4tRyGh6wjBf publickeypair

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDRd4dZzufck1SQRtsrRgBWTMJe1cB4aVdUJ2PptH4ILrqHtS+r5FSD1q5va
SmCMV7uJEQZ/7Yi7LrIk3ts089mBuxmRMY+PoXf0nnSYkIinRR3cGd52aZoY7TBQs1Zt8Ilry7BgItsA54E/7ycYxA0nZyCc
SHdFYdpcY2WQCq8FweJaOwWNhev2ypcuAFpIJQmzTj7rs7F+JNEJ45L3dTvTvwNyyv4GDfoVhIIN1GNxTVbIovvvdGn9aBvXU
OcQSdiP+zt9Fk6RwQs960BnUqcXN3Ds9/+hewToSWSjFpXdrWhM1S1VTQueCNXNbYA6khKA7d8d6c0y/pTiiPsaDjCG3xcFBk
qQqQGCLTEj/csr1goCBRpuFtqTKugqyBnEWIiNebNh/Dj76Eil/YBZboE5JVNFxi4be8u91AHldbnGlGUtbqnbxZ6/gidF
nmgyAt2N0oiiyB3ETv3HpTENEo/EYpw9Y/ktUmRQJPHUtHlZ0Bfc= ubuntu@Jenkins-Master
[ubuntu@Jenkins-Agent:~/.ssh$ █

```

Step: Create a Jenkins File in the GitHub repo.

Github link:- <https://github.com/mihirmodi2561/register-app>

Step 4: New Server SONAR QUBE

SONAR QUBE INSTALLTION

Once you have installed Sonar Qube, start integrating with Jenkins.

My Account > Security > Generate Token

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

?

Search for projects...

A

A

Administrator

ProfileSecurityNotificationsProjects

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

Name

Type

Expires in

Enter Token Name

Select Token Type

30 days

Generate

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

Copy

sqa_1205b7972888cc05757a88c67e6a8b678f379c

Name	Type	Project	Last use	Created	Expiration
Jenkins-sonarqu...	Global		Never	December 22, 2023	-

Revoke

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

New credentials

Kind

Secret text

Scope ?

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

Secret

.....

Sonar Qube Token

ID ?

Sonar-Qube

Description ?

Sonar-Qube

Create

Add This 3 Plugin in Jenkins

Install	Name ↓
<input checked="" type="checkbox"/>	<p>SonarQube Scanner 2.16.1</p> <p>External Site/Tool Integrations Build Reports</p> <p>This plugin allows an easy integration of SonarQube, the open source platform code quality.</p>
<input checked="" type="checkbox"/>	<p>Sonar Quality Gates 1.3.1</p> <p>Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis are (Quality Gates status is different than "Passed")</p> <p>Warning: This plugin version may not be safe to use. Please review the following:</p> <ul style="list-style-type: none"> Credentials transmitted in plain text
<input checked="" type="checkbox"/>	<p>Quality Gates 2.5</p> <p>Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (Quality Gates status is different than "Passed")</p> <p>Warning: This plugin version may not be safe to use. Please review the following:</p> <ul style="list-style-type: none"> Credentials transmitted in plain text

Once you installed, Now you have to set up in Jenkins system.

Manage Jenkins > system

Name

sonar-qube-server

Server URL

Default is http://localhost:9000

172.31.92.63:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

Sonar-Qube

+ Add ▼

Advanced ▼

SonarQube Scanner installations

Add SonarQube Scanner

☰ SonarQube Scanner

Name

SonarQube-scanner

! Required

☒ Install automatically ?

☰ Install from Maven Central

Version

SonarQube Scanner 5.0.1.3006

Add Installer ▾

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

Administration

Configuration Security Projects System Marketplace

General Settings
Encryption
Webhooks

to notify external services when a project analysis is done. An HTTP POST request including a JSON payload is sent
and URLs. Learn more in the [Webhooks documentation](#).

Create

Jenkins-master private IP/sonarqube-webhook

Name	URL	Secret?	Last delivery
sonarqube-webhook	http://172.31.93.32:8080/sonarqube-webhook	No	December 22, 2023 at 8:05 PM

+

Add Plugins for Docker

Updates

Available plugins

Installed plugins

Advanced settings

docker



Docker 1.5

Cloud Providers Cluster Management docker

This plugin integrates Jenkins with Docker



Docker Commons 439.va_3cb_0a_6a_fb_29

Library plugins (for use by other plugins) docker

Provides the common shared functionality for various Docker-related plugins.



Docker Pipeline 572.v950f58993843

pipeline DevOps Deployment docker

Build and use Docker containers from pipelines.



Docker API 3.3.4-86.v39b_a_5ede342c

Library plugins (for use by other plugins) docker

This plugin provides docker-java API for other plugins.

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



docker-build-step 2.10

Build Tools docker

This plugin allows to add various docker commands to your job as build steps.



CloudBees Docker Build and Publish 1.4.0

Build Tools docker

This plugin enables building Dockerfile based projects, as well as publishing of the built the docker registry.

Run Jenkins script

```

pipeline{
  agent { label 'Jenkins-Agent' }
  tools{
    jdk 'Java17'
    maven 'Maven3'
  }
  environment{
    APP_NAME = "register-app-ci"
    RELEASE = "1.0.0"
    DOCKER_USER = "mihirmodi2561"
    DOCKER_PASS = 'DockerHub'
    IMAGE_NAME = "${DOCKER_USER}" + "/" + "${APP_NAME}"
    IMAGE_TAG = "${RELEASE}-${BUILD_NUMBER}"
  }
  stages{
    stage("Cleanup Worksapce"){
      steps{
        cleanWs()
      }
    }
    stage("Check out from SCM"){
      steps{
        git credentialsId: 'github', url:
'https://github.com/mihirmodi2561/Register-app-ci-cd'
      }
    }
    stage("Build Application"){
      steps{
        sh "mvn clean package"
      }
    }
    stage("Test Application"){
      steps{
        sh "mvn test"
      }
    }
    stage("SonarQube Analysis"){
      steps{
        script{
          withSonarQubeEnv(credentialsId: 'Sonar-Qube'){
            sh "mvn sonar:sonar"
          }
        }
      }
    }
    stage("Quality check"){
      steps{
        script{
          waitForQualityGate abortPipeline: false, credentialsId:
'Sonar-Qube'
        }
      }
    }
  }
}

```

```

    }
    stage("Build & Push Docker Image"){
        steps{
            script{
                docker.withRegistry('', DOCKER_PASS){
                    docker_image = docker.build "${IMAGE_NAME}"
                }

                docker.withRegistry('', DOCKER_PASS){
                    docker_image.push("${IMAGE_TAG}")
                    docker_image.push('latest')
                }
            }
        }
    }
    stage("Trivy scan"){
        steps{
            script{
                sh(' docker run -v
/var/run/docker.sock:/var/run/docker.sock aquasec/trivy image
mihirmodi2561/register-app-ci:latest --no-progress --scanner vuln --exit-code
0 --severity HIGH, CRITICAL --format table')
            }
        }
    }
    stage('Clean Artifacts'){
        steps{
            script{
                sh "docker rmi ${IMAGE_NAME}:${IMAGE_TAG}"
                sh "docker rmi ${IMAGE_NAME}:latest"
            }
        }
    }
}
}
}

```

Setup Bootstrap EKS Server.

Install AWS Cli on the above EC2

```

sudo su
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
apt install unzip,    $ unzip awscliv2.zip
sudo ./aws/install

```

```
pip3 install --user awscli
sudo ln -s $HOME/.local/bin/aws /usr/bin/aws
aws --version
```

Installing kubectl

```
sudo su
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.1/2023-04-19/bin/linux/amd64/kubectl
ll , $ chmod +x ./kubectl //Gave executable permissions
mv kubectl /bin //Because all our executable files are in /bin
kubectl version --output=yaml
```

Installing eksctl

```
curl --silent --location
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname
-s)_amd64.tar.gz" | tar xz -C /tmp
cd /tmp
$ ll
$ sudo mv /tmp/eksctl /bin
$ eksctl version
```

Setup Kubernetes using eksctl

```
eksctl create cluster --name virtualtechbox-cluster \
--region ap-south-1 \
--node-type t2.small \
--nodes 3 \
```

check the node from below.

```
kubectl get nodes
```

Add IAM role to EKS-bootstrap

Add Permission of AdministratorAccess

after creating the IAM Role, add the IAM role to the EKS-bootstrap server.

ArgoCD Installation on EKS Cluster and Add EKS Cluster to ArgoCD

First, create a namespace

```
kubectl create namespace argocd
```

Next, let's apply the yaml configuration files for ArgoCd

```
kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
```

Now, we can view the pods created in the ArgoCD namespace.

```
kubectl get pods -n argocd
```

To interact with the API Server we need to deploy the CLI:

```
curl --silent --location -o /usr/local/bin/argocd
https://github.com/argoproj/argo-cd/releases/download/v2.4.7/argocd-linux-
amd64
chmod +x /usr/local/bin/argocd
```

Expose argocd-server

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

```
Last login: Mon Dec 25 02:23:19 2023 from 35.147.33.158
(ubuntu@EKS-Bootstrap:~)$ kubectl cluster-info
Kubernetes control plane is running at https://A12CF1EE5466651D85C86A841000AB82.gr7.ap-south-1.eks.amazonaws.com
CoreDNS is running at https://A12CF1EE5466651D85C86A841000AB82.gr7.ap-south-1.eks.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
(ubuntu@EKS-Bootstrap:~)$ kubectl patch svc -n argocd -p '{"spec": {"type": "LoadBalancer"}}'
service/argocd-server patched
(ubuntu@EKS-Bootstrap:~)$ kubectl get svc -n argocd
```

NAME	AGE	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
argocd-applicationset-controller	61m	ClusterIP	10.100.46.221	<none>	7000/TCP,8080/TCP
argocd-dex-server	61m	ClusterIP	10.100.41.166	<none>	5556/TCP,5557/TCP
argocd-metrics	61m	ClusterIP	10.100.249.201	<none>	8082/TCP
argocd-notifications-controller-metrics	61m	ClusterIP	10.100.81.182	<none>	9001/TCP
argocd-redis	61m	ClusterIP	10.100.203.68	<none>	6379/TCP
argocd-repo-server	61m	ClusterIP	10.100.68.162	<none>	8081/TCP,8084/TCP
argocd-server	61m	LoadBalancer	10.100.83.124	ad786358ac7274798b0bd9e42a450c46-1852836236.ap-south-1.elb.amazonaws.com	80:32731/TCP,443:32257/TCP
argocd-server-metrics	61m	ClusterIP	10.100.179.25	<none>	8083/TCP

Wait about 2 minutes for the LoadBalancer creation

```
kubectl get svc -n argocd
```

```

ubuntu@EKS-Bootstrap:~$ kubectl get secret argocd-initial-admin-secret -n argocd -o yaml
apiVersion: v1
data:
  password: aG5UNnZTY0FNZF1XR0VZNg==
kind: Secret
metadata:
  creationTimestamp: "2023-12-25T01:50:47Z"
  name: argocd-initial-admin-secret
  namespace: argocd
  resourceVersion: "8075"
  uid: d1857854-6ea7-4a7f-a6ae-f424ee02d6f0
type: Opaque
ubuntu@EKS-Bootstrap:~$ echo aG5UNnZTY0FNZF1XR0VZNg== | base64 --decode
hnT6vScAMdYWGEY6ubuntu@EKS-Bootstrap:~$

```

Get a password and decode it.

```

kubectl get secret argocd-initial-admin-secret -n argocd -o yaml
echo WXVpLUg2LWxoWjRkSHFmSA== | base64 --decode

```

login to ArgoCD from CLI

```

argocd login ad786358ac7274798b0bd9e42a450c46-1852836236.ap-south-1.elb.amazonaws.com --username admin

```

```

ubuntu@EKS-Bootstrap:~$ echo aG5UNnZTY0FNZF1XR0VZNg== | base64 --decode
hnT6vScAMdYWGEY6ubuntu@EKS-Bootstrap:~$ echo ad786358ac7274798b0bd9e42a450c46-1852836236.ap-south-1.elb.amazonaws.com | base64 --decode
ad786358ac7274798b0bd9e42a450c46-1852836236.ap-south-1.elb.amazonaws.com
ubuntu@EKS-Bootstrap:~$ argocd login ad786358ac7274798b0bd9e42a450c46-1852836236.ap-south-1.elb.amazonaws.com --username admin
WARNING: server certificate had error: x509: certificate is valid for localhost, argocd-server, argocd-server.argocd, argocd-server.argocd.svc, argocd-server.svc, cluster.local, not ad786358ac7274798b0bd9e42a450c46-1852836236.ap-south-1.elb.amazonaws.com. Proceed insecurely (y/n)? y
Password:
'admin:login' logged in successfully
Context 'ad786358ac7274798b0bd9e42a450c46-1852836236.ap-south-1.elb.amazonaws.com' updated
ubuntu@EKS-Bootstrap:~$ kubectl config get-contexts
CURRENT  NAME                                     CLUSTER                                     AUTHINFO
*         i-0cf29193e810db4c4@virtualtechbox-cluster.ap-south-1.eksctl.io  virtualtechbox-cluster.ap-south-1.eksctl.io  i-0cf29193e810db4c4@virtualtechbox-cluster.ap-south-1.eksctl.io
ubuntu@EKS-Bootstrap:~$ argocd cluster add i-08b9d0ff0409f48e7@virtualtechbox-cluster.ap-south-1.eksctl.io --name virtualtechbox-eks-cluster
FATA[0000] Context i-08b9d0ff0409f48e7@virtualtechbox-cluster.ap-south-1.eksctl.io does not exist in kubeconfig

```

```

argocd cluster list

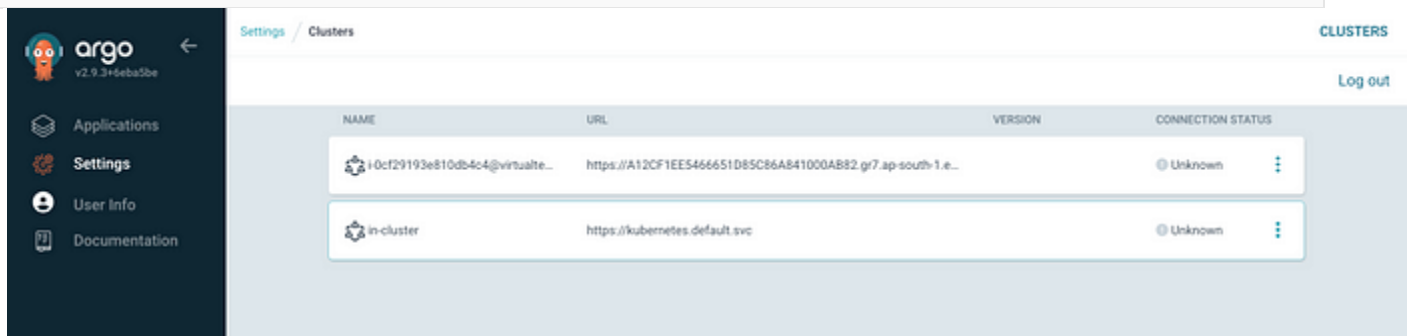
```

Below command will show the EKS cluster


```
kubectl config get-contexts
```

Add the above EKS cluster to ArgoCD with the below command

```
argocd cluster add i-0cf29193e810db4c4@virtualtechbox-cluster.ap-south-1.eksctl.io  
kubectl get svc
```



argo → setting → repositories → connect repo with github repo

argo application healthy.

argo application healthy.

New user Register for DevOps Learning at Vir

Please fill in this form to create an account.

Enter Name	<input type="text" value="Enter Full Name"/>
Enter mobile	<input type="text" value="Enter moible number"/>
Enter Email	<input type="text" value="Enter Email"/>
Password	<input type="password" value="Enter Password"/>
Repeat Password	<input type="password" value="Repeat Password"/>

By creating an account you agree to our [Terms & Privacy](#).

Already have an account? [Sign in](#).

Thank You

Website launched sueccsfully.

Thank you..