Launch an ec2
 -cicddemo
 -ubuntu
 -t2 large 2cpu 8gb mem
 -keypair
--> [Launch]

//Run app locally,
Clone the repo locally
Springboot app
Open readme.md
=>install maven
-> mvn clean package
-build scucces
-no need to run jar file
-run as docker image
-docker build -t cicd-Pipeline:v1
-docker run -d -p 8010:8080 cicd-Pipeline:v1
-Acces on local host 8010
-Deployed.

=>ssh into ec2
-sudo apt update
- sudo apt install openjdk-11-jdk
- java --version
=> Install jenkins
-curl ...
-echo....
-sudo apt-get update
-sudo apt-get install jenkins
=> SECURITY GROUP
-all traffic- anywhere.
#check how jenkins is running.
$ ps -ef | grep jenkins
-access on public_ip:8080

---JENKINS----
-New item
-name= ultimate-demo
-pipeline
-okay
-configuration
-pipeline from scm
-Jenkins file inside ur repo
-copy url
-main
-script path
-select->ightweight checkout

//Use docker as agent.
-lessen the config.
[Note:
.Have to do lot of configuration.
.lot of resource usage unnecessarily
.download all dependencies.
Advantage...
.as pipeline is created.
.it take the job of creating a docker bcontainer.
.all stages successfull.
.it gets deleted.
.resources freed up.
.so very useful.

So if ur using docker as agent.
1.Docker plugin
2.Choose the docker image wisely

Maven image already in container.

Install sonar-server to attach ec2
Commands on github readme.

//diff between useradd & adduser.
go back to root user.

Root@ apt install unzip
$adduser sonarqube
$ sudo su -
$ sudo su - sonarqube
$ls
$unzip *
$cd
$ chmod -R
$cd
$ ./sonar.sh
Access public_ip:9000
Name=admin
Password=admin
Myaccoutn
Security
Generate token
Name= jenkins
Copy the token
Got to jenkins server.
Manage jenkins
Credentials

System
Global credentials
Add credentials
Secret text
Name=sonarqube
=> sonarqube config. Done.

__Configuration of tools___

Plug-Ins
1.docker pipeline ✔️
2.sonarqube scanner ✔️

Install docker on ec2
Copy cmds from repo.
Go back and restart jenkins
Searchbar public_ip:8080/restart.
Password.
Copy the
Keep me signed in.

U need k8s and argo cd.

U can install locally. Apple hai na😒
We cant.

Run a minikube cluster.
Go to minikube page.
Kubernetes controllers lifecycle controllled by operators.
Got to operatorhub.io
Search for ArgoCD
-Olm (Operator Life cycleManager) curl url.
-kubectl create -f.....
-kubectl get svc -n operators
-kubectl get pods -n operators

//mvn clean package
No need to push to artifactory
Use into docker image
And push into docker registry.
//mvn clean install
Push ur archive to any artifactory

Go to jenkins
-Manage jenkins
-credentials

-System
-global credentials
-add
 -username with password
    Name=
     Password=
    Id= docker-cred
  - secret text
  - id =github
  //go to github
  Settings
  Developer settings
  Token
  Generate a classic token
  Name=jenkins
  Generate token
  Copy token.

Restart jenkins

Replace the sonar url

Check kubectl get pods -n operators
   -->Running

BUILD NOW.
-downloading docker image.
-Checkout
-Build and test
-statuc code analysis
-Build and push docker
-Update deployment file done ✔️
-Success 🙌

Refresh the github

Go to sonarqube
Springboot demo project done.

C-I part is done.

Now CD ... on ArgoCD.
Now create the argocd controller.
Got to ArgoCD operators management
Usahe basics
Copy example as is.
Vim argocd-basic.yml
Paste.

Kubectl apply -f argocd-basic.yml
Kubectl get pods
Now all the argocd workloads are getting ready.
//Go to argo cd UI and pull the latest image from git repo and And onto the k8s cluster using the cd process..

Now we want those containers to run on browser also.. so..
$ kubectl get svc
$kubectl edit svc example-argocd-server
//Change the type.. Cluster_Ip to node port.
$Kubectl get svc
Notice the change to nodeport.

Now to execute on browser..
Minikube offers u a service..it will generate a url to access by port forwarding.

$minikube service argocd-server
$ minikube service list
    -->U will get urls.
$ kubectl get pods.
Copy the url..
Ul get access to argocd.
Username= admin
//get
$ kubectl get secret
In argocd cluster
$kubectl edit secret example-argocd-cluser
Copy the password.
//k8s secrets are not in plain texts
$echo ___password ___= | base64 -dķ
Ull get a "password"% dont copy the % symbol
Password= paste it here.
Logged into ArgoCd

-Create application
-name= test
-project name= default
-sync =automatic
-Source url = github.com/Veeramalla/jenkins-zero-to-hero
-revison= HEAD
-path of the deployment file
-cluster name= https://kubernetes.default.svc
-namespace default
--> create.
Click on sync

$kubectl get deploy
Deployment done

$kubectl get pods
Pods also running

How to check is healthy.

A malicious user :-

Kubectl edit deploy Springboot-app
Update image version from 1 to 2
So it should fail.
Ill save it. It got edited.

Now refresh the argocd.
It will immediately catch the difference.
And it throw an error.
Currently "OUTOFSYNC" STATUS
Difference tab.
It will highlight the changes in the file.
It will automatically fiz it to version:'1'
It will automatically heal it.

Perfectly DONE. ✔️...