Tasks:
create an environment for a sample python application website hosting with the help of AWS eksctl and use ECR for storing images and use GitHub for SCM

use Jenkins groove pipeline for automated deployments end to end flow

prerequisite:-
APPLICATION:- python
SOURCE CODE MANAGEMENT:- Github
Docker:- build the image
AWS ECR:- to storage the image into ecr
Deploy:- AWS eksctl cluster
CI & CD Tool: Jenkins
The flow of CI/CD:- USE above all the services and create CI/CD Jenkins groove pipeline.

Prerequisites:

1. Java
2. Jenkins
3. Aws CLI
4. Kubectl
5. eksctl

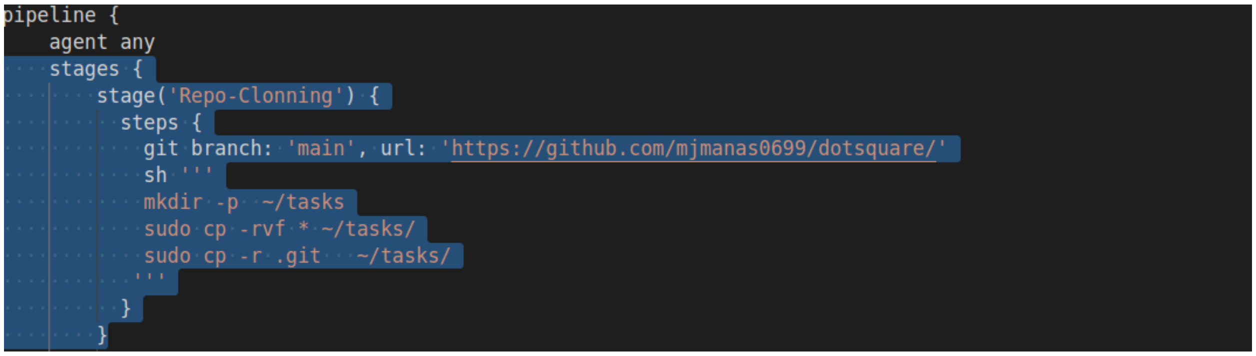
Let's start :

First, I have installed Jenkins after installing it and installed the following plugins:
1: Pipeline
2: git

Github repo: https://github.com/mjmanas0699/dotsquare

I Have created 5 stages in Jenkinsfile. I will start explaining one by one what I did in these individual stages

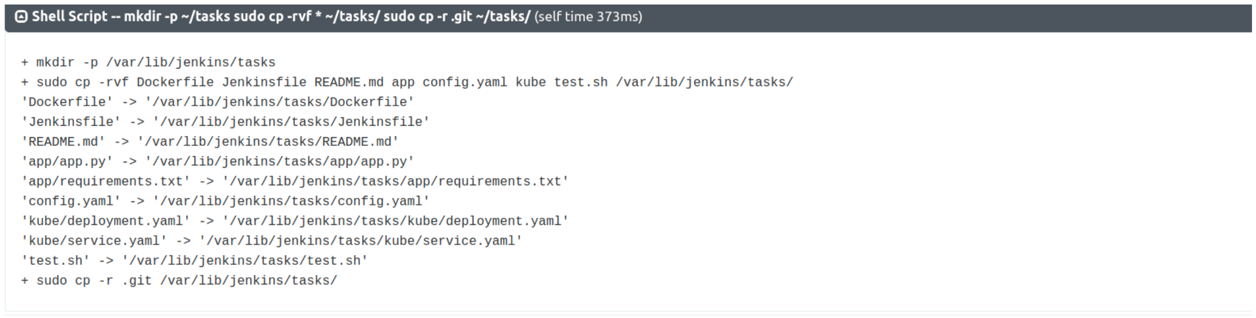
Stage 1:



```
pipeline {
    agent any
    stages {
        stage('Repo-Clonning') {
          steps {
            git branch: 'main', url: 'https://github.com/mjmanas0699/dotsquare/'
            sh '''
            mkdir -p  ~/tasks
            sudo cp -rvf * ~/tasks/
            sudo cp -r .git   ~/tasks/
           '''
          }
        }
```

So in stage 1 I'm cloning the repo which I mentioned above and moving the content of that repo into a different folder so firstly I created the folder with mkdir command on the home directory of the Jenkins user and then moved the data from the workspace folder to ~/tasks directory.
Here I used "Sudo" I have given Jenkins user permission with visudo to perform the task with Sudo privileges

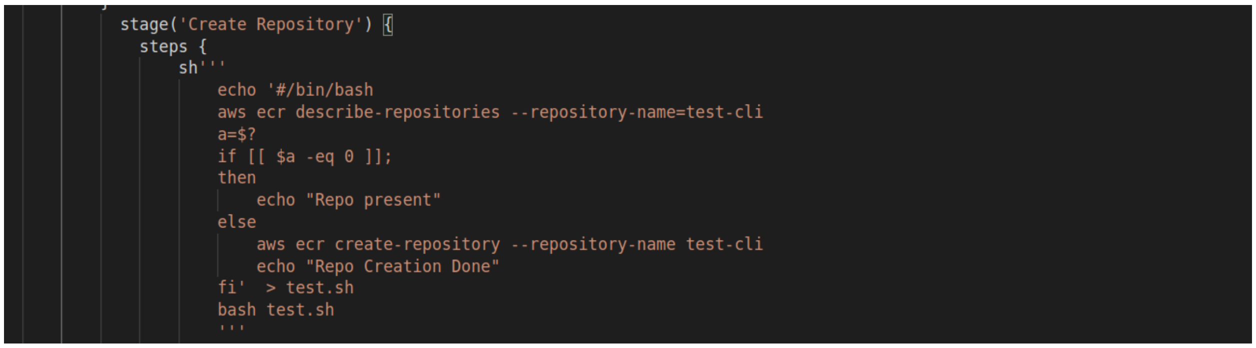
Result of stage 1:



Shell Script -- mkdir -p ~/tasks sudo cp -rvf * ~/tasks/ sudo cp -r .git ~/tasks/ (self time 373ms)

```
+ mkdir -p /var/lib/jenkins/tasks
+ sudo cp -rvf Dockerfile Jenkinsfile README.md app config.yaml kube test.sh /var/lib/jenkins/tasks/
'Dockerfile' -> '/var/lib/jenkins/tasks/Dockerfile'
'Jenkinsfile' -> '/var/lib/jenkins/tasks/Jenkinsfile'
'README.md' -> '/var/lib/jenkins/tasks/README.md'
'app/app.py' -> '/var/lib/jenkins/tasks/app/app.py'
'app/requirements.txt' -> '/var/lib/jenkins/tasks/app/requirements.txt'
'config.yaml' -> '/var/lib/jenkins/tasks/config.yaml'
'kube/deployment.yaml' -> '/var/lib/jenkins/tasks/kube/deployment.yaml'
'kube/service.yaml' -> '/var/lib/jenkins/tasks/kube/service.yaml'
'test.sh' -> '/var/lib/jenkins/tasks/test.sh'
+ sudo cp -r .git /var/lib/jenkins/tasks/
```

Stage2:

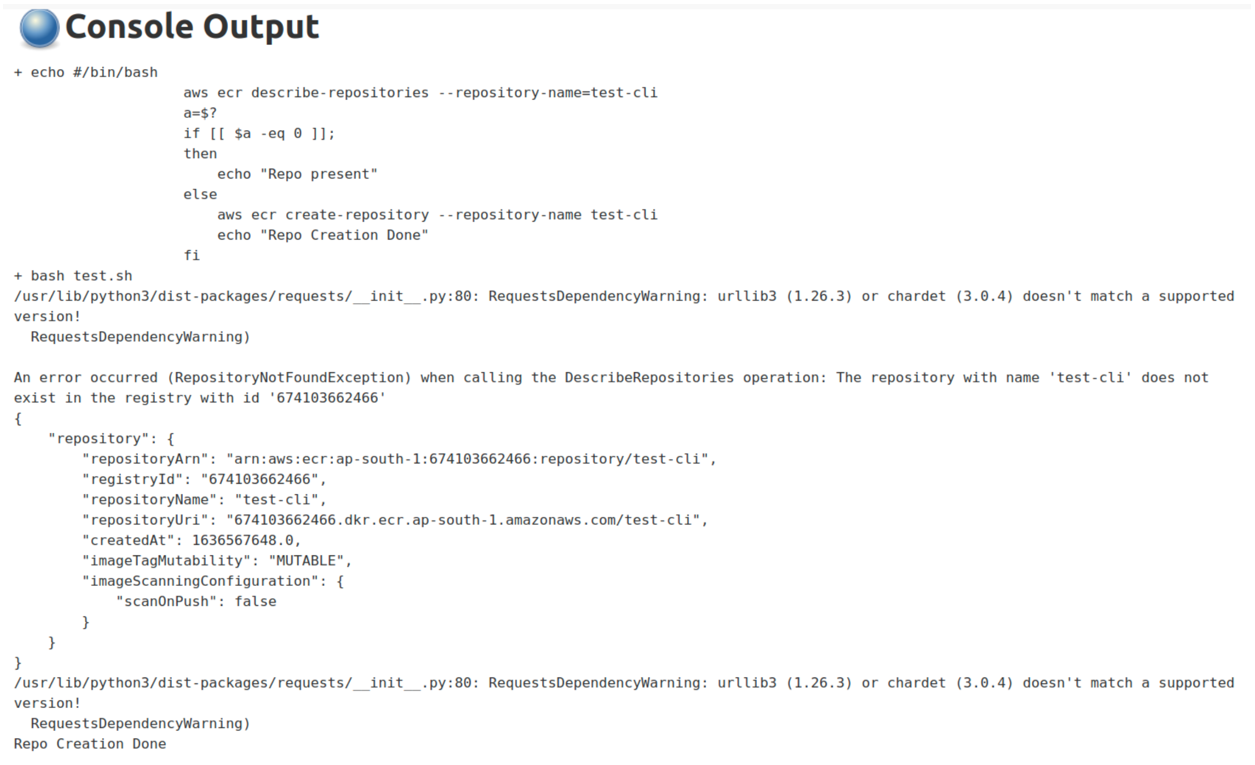


```
        stage('Create Repository') {
          steps {
              sh'''
                  echo '#/bin/bash
                  aws ecr describe-repositories --repository-name=test-cli
                  a=$?
                  if [[ $a -eq 0 ]];
                  then
                      echo "Repo present"
                  else
                      aws ecr create-repository --repository-name test-cli
                      echo "Repo Creation Done"
                  fi'  > test.sh
                  bash test.sh
                  '''
```

So, in stage 2 I'm creating the Repository on the AWS through the AWS CLI command utility. I'm giving the predefined name. So in the first command, I'm checking whether the repository is available or not if this isn't available so the command status code would be a non-zero number

And if it is non zero number so it will create the repo on AWS. I had preconfigured the **AWS configure** command

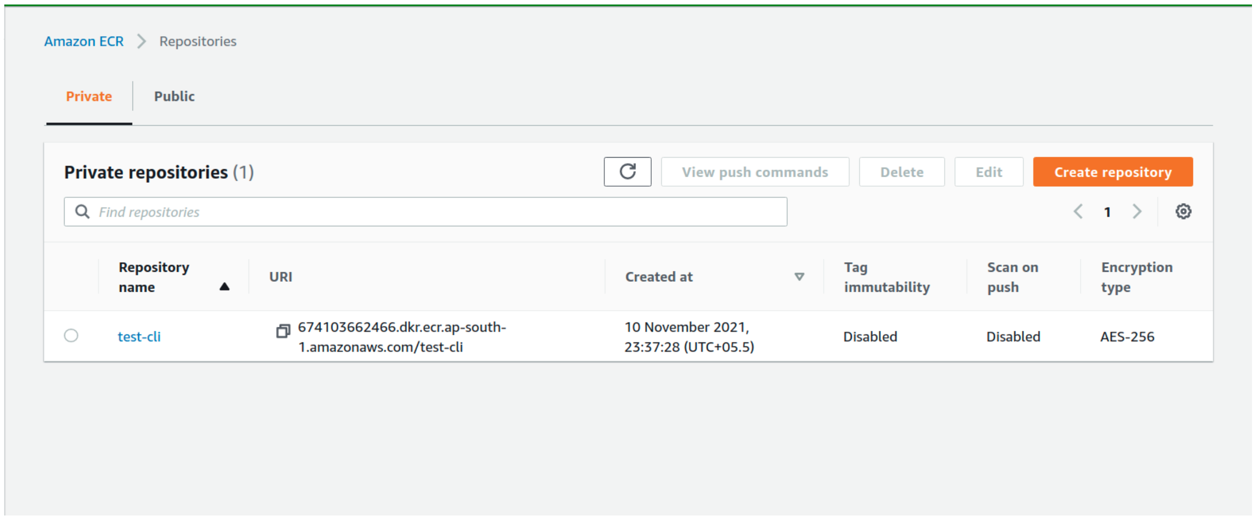
Result of stage2:



Console Output

```
+ echo #/bin/bash
                    aws ecr describe-repositories --repository-name=test-cli
                    a=$?
                    if [[ $a -eq 0 ]];
                    then
                        echo "Repo present"
                    else
                        aws ecr create-repository --repository-name test-cli
                        echo "Repo Creation Done"
                    fi
+ bash test.sh
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match a supported
version!
  RequestsDependencyWarning)

An error occurred (RepositoryNotFoundException) when calling the DescribeRepositories operation: The repository with name 'test-cli' does not
exist in the registry with id '674103662466'
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:ap-south-1:674103662466:repository/test-cli",
        "registryId": "674103662466",
        "repositoryName": "test-cli",
        "repositoryUri": "674103662466.dkr.ecr.ap-south-1.amazonaws.com/test-cli",
        "createdAt": 1636567648.0,
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        }
    }
}
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match a supported
version!
  RequestsDependencyWarning)
Repo Creation Done
```

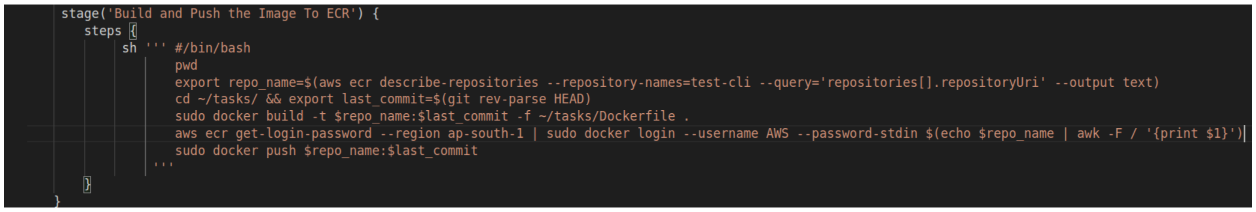
Here is the output of the AWS console of the ECR private repo which is created by Jenkins



Private repositories (1)

Repository name	URI	Created at	Tag immutability	Scan on push	Encryption type
test-cli	674103662466.dkr.ecr.ap-south-1.amazonaws.com/test-cli	10 November 2021, 23:37:28 (UTC+05.5)	Disabled	Disabled	AES-256

Stage 3:



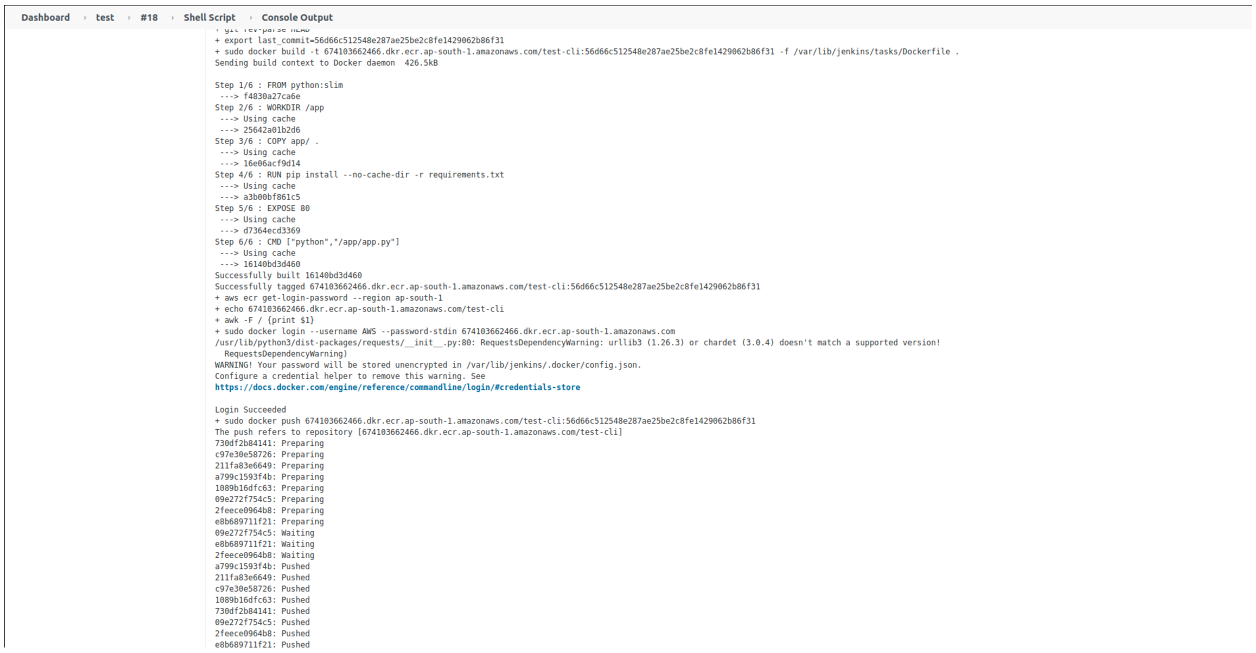
```
stage('Build and Push the Image To ECR') {
    steps {
        sh ''' #/bin/bash
            pwd
            export repo_name=$(aws ecr describe-repositories --repository-names=test-cli --query='repositories[].repositoryUri' --output text)
            cd ~/tasks/ && export last_commit=$(git rev-parse HEAD)
            sudo docker build -t $repo_name:$last_commit -f ~/tasks/Dockerfile .
            aws ecr get-login-password --region ap-south-1 | sudo docker login --username AWS --password-stdin $(echo $repo_name | awk -F / '{print $1}')
            sudo docker push $repo_name:$last_commit
           '''
    }
}
```

So, in stage3 I'm Build and Push the Image To AWS ECR. In the first command, I'm exporting a variable **repo_name** that will contain the repository name which was created in stage2. In the 2nd command going inside the ~/tasks dir and run the **git rev-parse HEAD** command and take the output of this command in a variable **last_commit.** This command will list the last commit id. In the 3rd command, I'm building the image with the docker command and passing the variable name so all the time a new image is built it will use a new commit id so the conflict won't occur and it is easy to find in GitHub if we assign commit id to the docker image and in the last 2 commands it is login and pushing that image to ECR

Dockerfile:

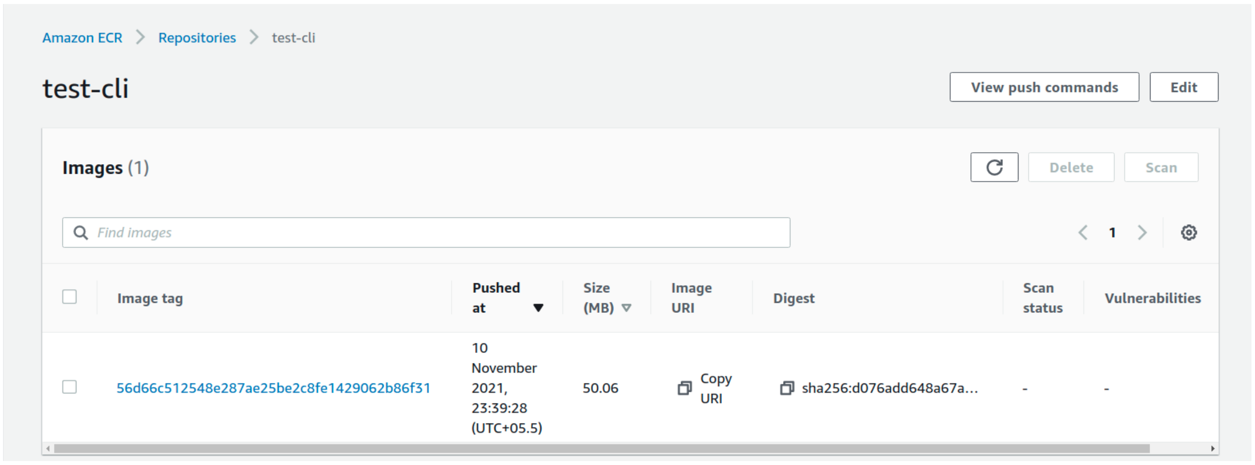
```
FROM python:slim

WORKDIR /app

COPY app/ .

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 80

CMD ["python","/app/app.py"]
```

Result of stage 3:



```
+ git rev-parse HEAD
+ export last_commit=56d66c512548e287ae25be2c8fe1429062b86f31
+ sudo docker build -t 674103662466.dkr.ecr.ap-south-1.amazonaws.com/test-cli:56d66c512548e287ae25be2c8fe1429062b86f31 -f /var/lib/jenkins/tasks/Dockerfile .
Sending build context to Docker daemon  426.5kB

Step 1/6 : FROM python:slim
 ---> f4830a27ca6e
Step 2/6 : WORKDIR /app
 ---> Using cache
 ---> 25642a01b2d6
Step 3/6 : COPY app/ .
 ---> Using cache
 ---> 16e06acf9d14
Step 4/6 : RUN pip install --no-cache-dir -r requirements.txt
 ---> Using cache
 ---> a3b00bf861c5
Step 5/6 : EXPOSE 80
 ---> Using cache
 ---> d7364ecd3369
Step 6/6 : CMD ["python","/app/app.py"]
 ---> Using cache
 ---> 16140bd3d460
Successfully built 16140bd3d460
Successfully tagged 674103662466.dkr.ecr.ap-south-1.amazonaws.com/test-cli:56d66c512548e287ae25be2c8fe1429062b86f31
+ aws ecr get-login-password --region ap-south-1
+ echo 674103662466.dkr.ecr.ap-south-1.amazonaws.com/test-cli
+ awk -F / {print $1}
+ sudo docker login --username AWS --password-stdin 674103662466.dkr.ecr.ap-south-1.amazonaws.com
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
+ sudo docker push 674103662466.dkr.ecr.ap-south-1.amazonaws.com/test-cli:56d66c512548e287ae25be2c8fe1429062b86f31
The push refers to repository [674103662466.dkr.ecr.ap-south-1.amazonaws.com/test-cli]
730df2b84141: Preparing
c97e30e58726: Preparing
211fa83e6649: Preparing
a799c1593f4b: Preparing
1089b16dfc63: Preparing
09e272f754c5: Preparing
2feece0964b8: Preparing
e8b689711f21: Preparing
09e272f754c5: Waiting
e8b689711f21: Waiting
2feece0964b8: Waiting
a799c1593f4b: Pushed
211fa83e6649: Pushed
c97e30e58726: Pushed
1089b16dfc63: Pushed
730df2b84141: Pushed
09e272f754c5: Pushed
2feece0964b8: Pushed
e8b689711f21: Pushed
```

Result of AWS console of pushed image

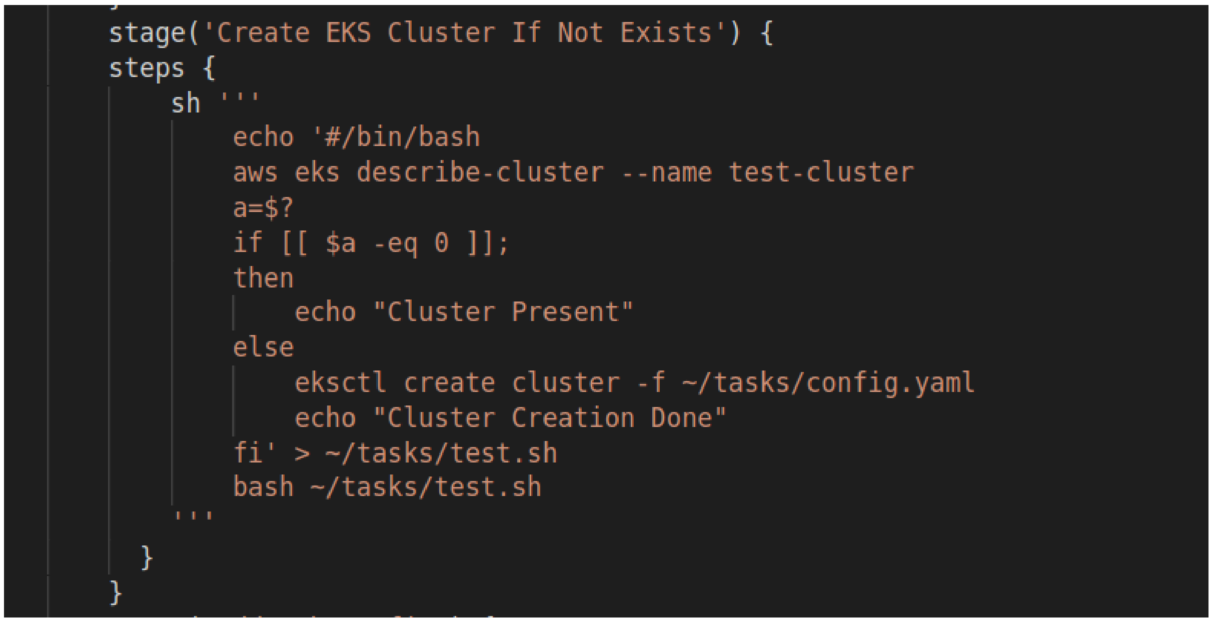


test-cli

Images (1)

	Image tag	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities
☐	56d66c512548e287ae25be2c8fe1429062b86f31	10 November 2021, 23:39:28 (UTC+05.5)	50.06	Copy URI	sha256:d076add648a67a...	-	-

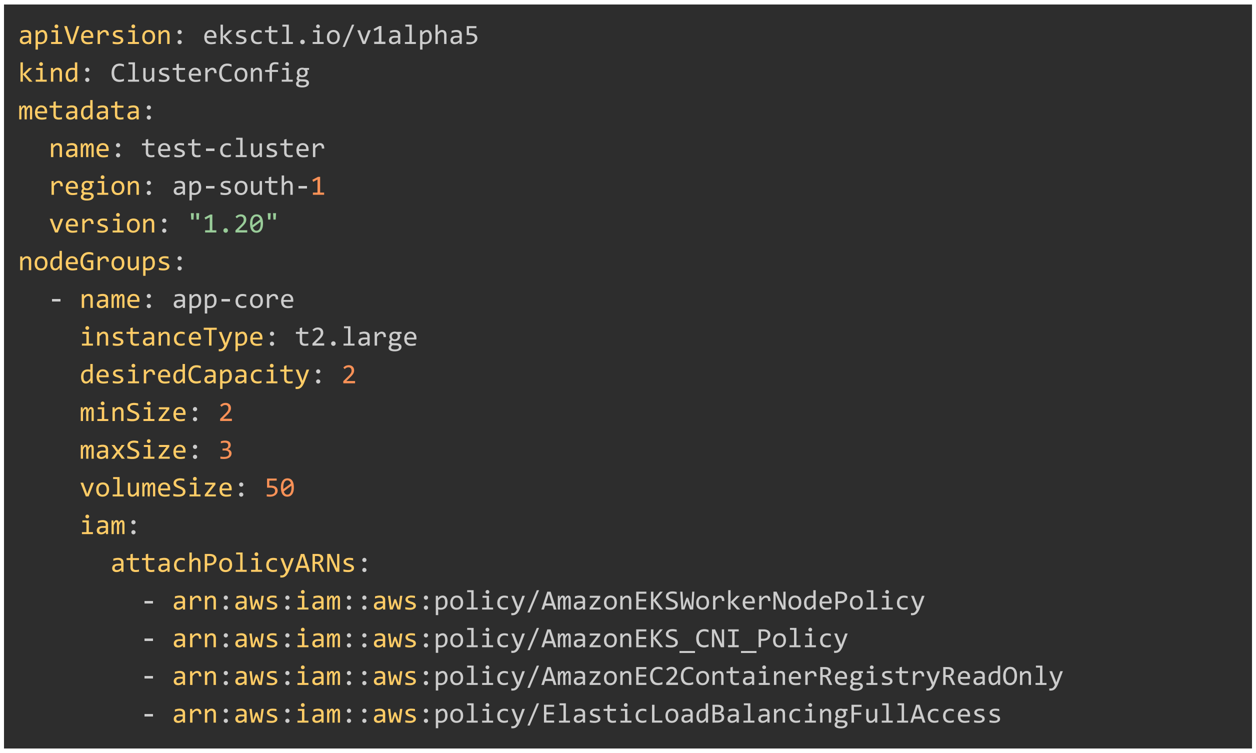
Stage 4:



```
stage('Create EKS Cluster If Not Exists') {
steps {
    sh '''
        echo '#/bin/bash
        aws eks describe-cluster --name test-cluster
        a=$?
        if [[ $a -eq 0 ]];
        then
            echo "Cluster Present"
        else
            eksctl create cluster -f ~/tasks/config.yaml
            echo "Cluster Creation Done"
        fi' > ~/tasks/test.sh
        bash ~/tasks/test.sh
    '''
  }
}
```

In stage 4 I'm checking the condition if the cluster exists or not if this does not exist so it will directly call the config file of the EKS with eksctl command and start creating the cluster

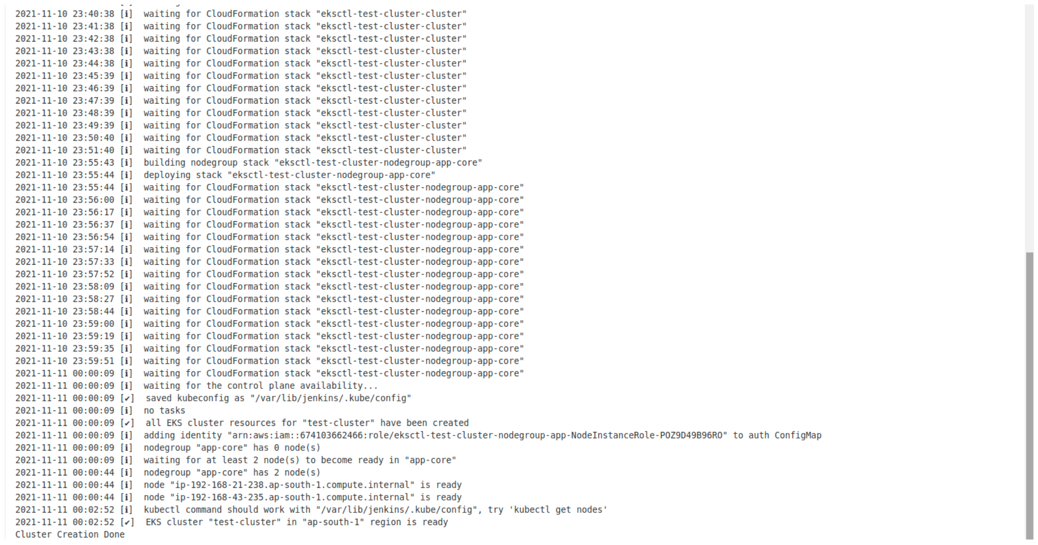
Config file:



```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: test-cluster
  region: ap-south-1
  version: "1.20"
nodeGroups:
  - name: app-core
    instanceType: t2.large
    desiredCapacity: 2
    minSize: 2
    maxSize: 3
    volumeSize: 50
    iam:
      attachPolicyARNs:
        - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
        - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
        - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
        - arn:aws:iam::aws:policy/ElasticLoadBalancingFullAccess
```

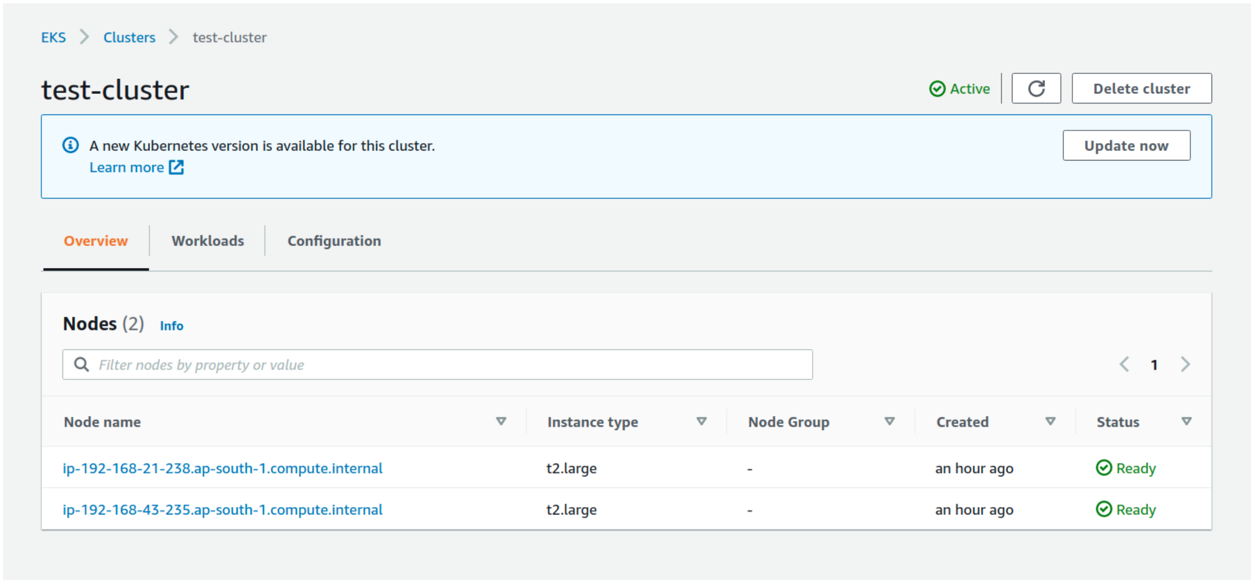
This is the ClusterConfig file of the eksctl in the metadata I'm giving the name of the cluster and in the node groups I'm defining the type of instance and the capacities and in the iam I'm attaching the roles which will be directly attached to the nodes so that they can pull the images from the ECR and also able get access of Elastic Load Balancing

So here is the output of cluster creation:



```
2021-11-10 23:40:38 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:41:38 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:42:38 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:43:38 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:44:38 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:45:39 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:46:39 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:47:39 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:48:39 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:49:39 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:50:40 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:51:40 [i]  waiting for CloudFormation stack "eksctl-test-cluster-cluster"
2021-11-10 23:55:43 [i]  building nodegroup stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:55:44 [i]  deploying stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:55:44 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:56:00 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:56:17 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:56:37 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:56:54 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:57:14 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:57:33 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:57:52 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:58:09 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:58:27 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:58:44 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:59:00 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:59:19 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:59:35 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-10 23:59:51 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-11 00:00:09 [i]  waiting for CloudFormation stack "eksctl-test-cluster-nodegroup-app-core"
2021-11-11 00:00:09 [i]  waiting for the control plane availability...
2021-11-11 00:00:09 [✓]  saved kubeconfig as "/var/lib/jenkins/.kube/config"
2021-11-11 00:00:09 [i]  no tasks
2021-11-11 00:00:09 [✓]  all EKS cluster resources for "test-cluster" have been created
2021-11-11 00:00:09 [i]  adding identity "arn:aws:iam::674103662466:role/eksctl-test-cluster-nodegroup-app-NodeInstanceRole-POZ9D49B96RO" to auth ConfigMap
2021-11-11 00:00:09 [i]  nodegroup "app-core" has 0 node(s)
2021-11-11 00:00:09 [i]  waiting for at least 2 node(s) to become ready in "app-core"
2021-11-11 00:00:44 [i]  nodegroup "app-core" has 2 node(s)
2021-11-11 00:00:44 [i]  node "ip-192-168-21-238.ap-south-1.compute.internal" is ready
2021-11-11 00:00:44 [i]  node "ip-192-168-43-235.ap-south-1.compute.internal" is ready
2021-11-11 00:02:52 [i]  kubectl command should work with "/var/lib/jenkins/.kube/config", try 'kubectl get nodes'
2021-11-11 00:02:52 [✓]  EKS cluster "test-cluster" in "ap-south-1" region is ready
Cluster Creation Done
```

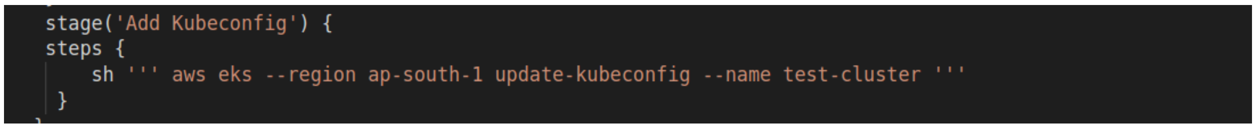
Here is the output of the AWS console of the EKS cluster which was created by eksctl command in the Jenkins



A new Kubernetes version is available for this cluster.

Node name	Instance type	Node Group	Created	Status
ip-192-168-21-238.ap-south-1.compute.internal	t2.large	-	an hour ago	✓ Ready
ip-192-168-43-235.ap-south-1.compute.internal	t2.large	-	an hour ago	✓ Ready

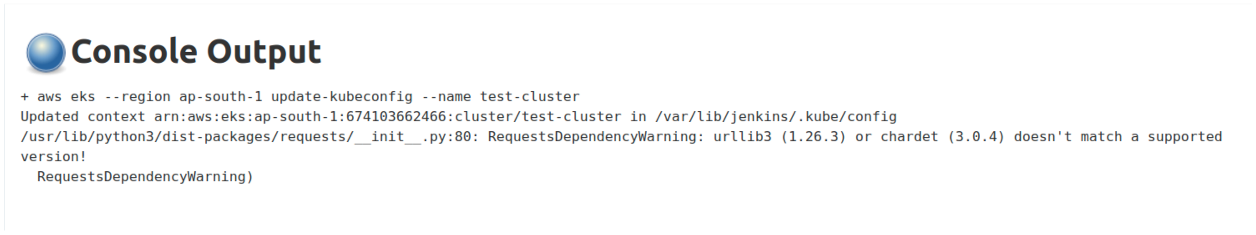
Stage5 :



```
stage('Add Kubeconfig') {
steps {
    sh ''' aws eks --region ap-south-1 update-kubeconfig --name test-cluster '''
 }
```

So in stage 5 I'm hitting the **aws eks update-kubeconfig** command for update kubeconfig os Jenkins user so that Jenkins user can able to connect via kubectl utility to Kubernetes cluster and perform the further tasks

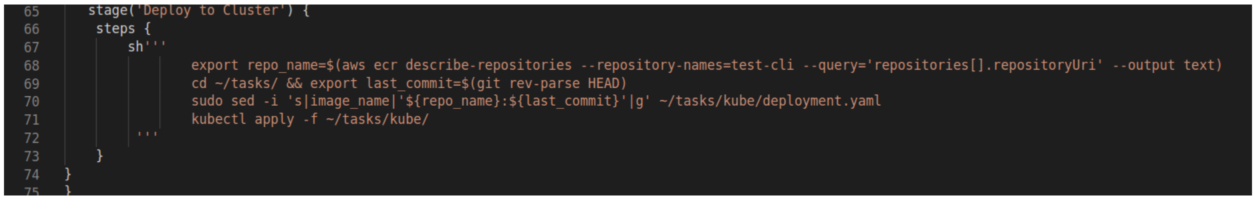
Here is the output of stage 5



Console Output

```
+ aws eks --region ap-south-1 update-kubeconfig --name test-cluster
Updated context arn:aws:eks:ap-south-1:674103662466:cluster/test-cluster in /var/lib/jenkins/.kube/config
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match a supported
version!
  RequestsDependencyWarning)
```

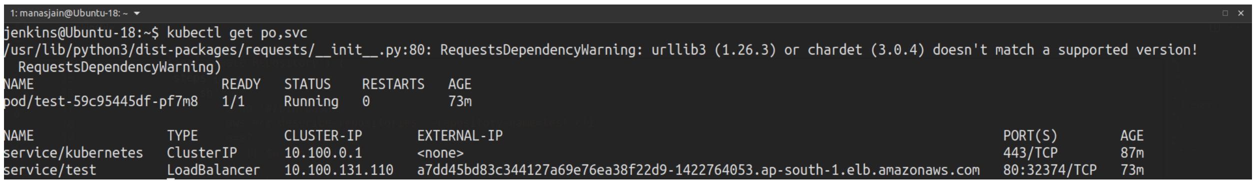
Final Stage:



```
    stage('Deploy to Cluster') {
     steps {
         sh'''
                export repo_name=$(aws ecr describe-repositories --repository-names=test-cli --query='repositories[].repositoryUri' --output text)
                cd ~/tasks/ && export last_commit=$(git rev-parse HEAD)
                sudo sed -i 's|image_name|'${repo_name}:${last_commit}'|g' ~/tasks/kube/deployment.yaml
                kubectl apply -f ~/tasks/kube/
          '''
     }
}
}
```

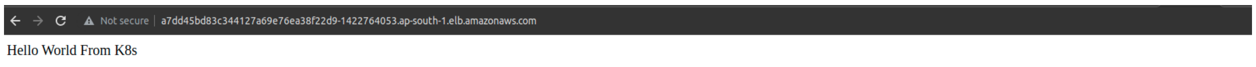
In the final stage of Jenkinsfile, I'm exporting the same variables which I was exported earlier for repo and last commit id. So in the third step of shell, I'm using the sed command so it can manipulate the values in the existing Kubernetes objects file it will find and replace the value of the image name according to the repo and last commit id so that Kubernetes can pull them. Because it has been already pushed by stage 3

The output of the final stage :



```
1: manasjain@Ubuntu-18: ~
jenkins@Ubuntu-18:~$ kubectl get po,svc
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
NAME                      READY   STATUS    RESTARTS   AGE
pod/test-59c95445df-pf7m8   1/1     Running   0          73m

NAME                 TYPE           CLUSTER-IP      EXTERNAL-IP                                                                PORT(S)        AGE
service/kubernetes   ClusterIP      10.100.0.1      <none>                                                                     443/TCP        87m
service/test         LoadBalancer   10.100.131.110  a7dd45bd83c344127a69e76ea38f22d9-1422764053.ap-south-1.elb.amazonaws.com   80:32374/TCP   73m
```

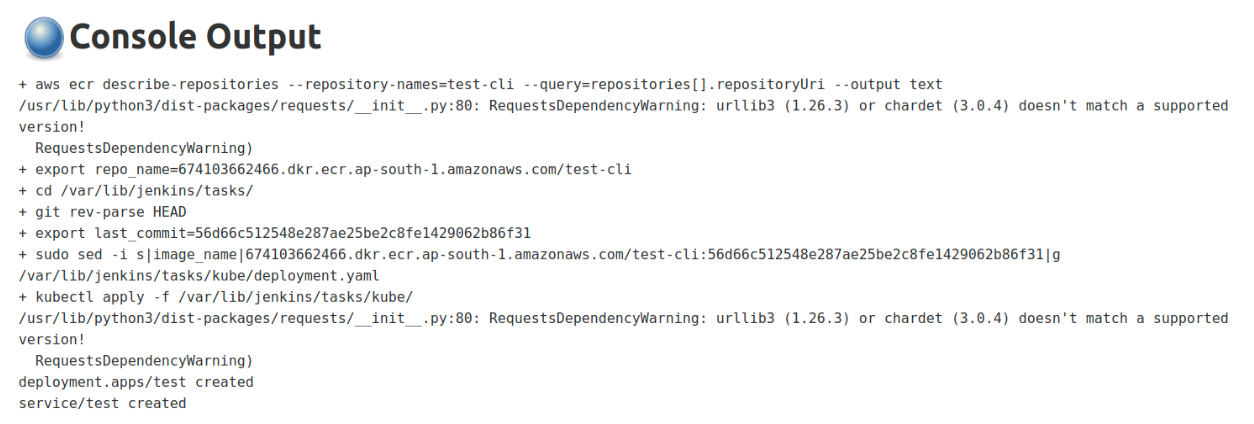
Webpage output of the application which is accessible with the load balancer



Not secure | a7dd45bd83c344127a69e76ea38f22d9-1422764053.ap-south-1.elb.amazonaws.com

Hello World From K8s

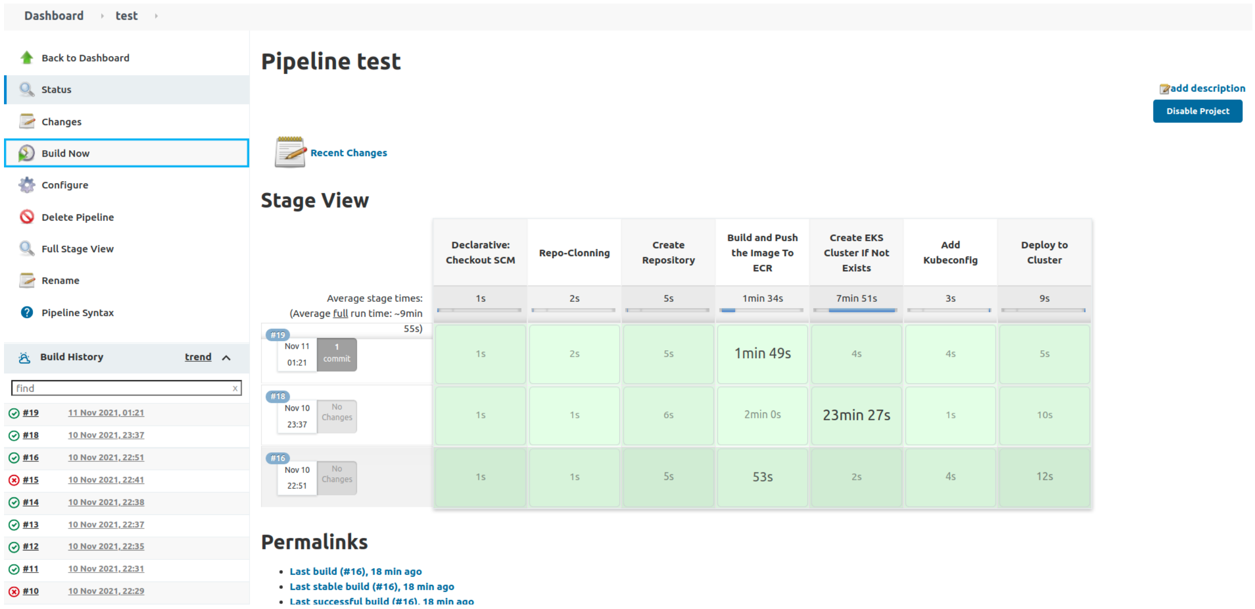
Result of Final stage



Console Output

```
+ aws ecr describe-repositories --repository-names=test-cli --query=repositories[].repositoryUri --output text
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match a supported
version!
  RequestsDependencyWarning)
+ export repo_name=674103662466.dkr.ecr.ap-south-1.amazonaws.com/test-cli
+ cd /var/lib/jenkins/tasks/
+ git rev-parse HEAD
+ export last_commit=56d66c512548e287ae25be2c8fe1429062b86f31
+ sudo sed -i s|image_name|674103662466.dkr.ecr.ap-south-1.amazonaws.com/test-cli:56d66c512548e287ae25be2c8fe1429062b86f31|g
/var/lib/jenkins/tasks/kube/deployment.yaml
+ kubectl apply -f /var/lib/jenkins/tasks/kube/
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match a supported
version!
  RequestsDependencyWarning)
deployment.apps/test created
service/test created
```

Here is the Final View of the Jenkins Pipeline. The difference b/w build 18 and 19 is in the 18 it created the cluster that's why it took around 23 min to complete the creation of the cluster and in the 19 the cluster already existed so it just skips that step that moves forward to perform further tasks



Dashboard › test ›

Back to Dashboard
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Rename
Pipeline Syntax

Build History trend

✓	#19	11 Nov 2021, 01:21
✓	#18	10 Nov 2021, 23:37
✓	#16	10 Nov 2021, 22:51
✗	#15	10 Nov 2021, 22:41
✓	#14	10 Nov 2021, 22:38
✓	#13	10 Nov 2021, 22:37
✓	#12	10 Nov 2021, 22:35
✓	#11	10 Nov 2021, 22:31
✗	#10	10 Nov 2021, 22:29

Pipeline test

add description
Disable Project

Recent Changes

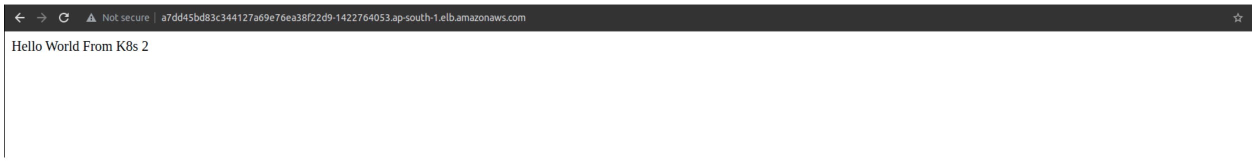
Stage View

	Declarative: Checkout SCM	Repo-Clonning	Create Repository	Build and Push the Image To ECR	Create EKS Cluster If Not Exists	Add Kubeconfig	Deploy to Cluster
Average stage times: (Average full run time: ~9min 55s)	1s	2s	5s	1min 34s	7min 51s	3s	9s
#19 Nov 11 01:21 1 commit	1s	2s	5s	1min 49s	4s	4s	5s
#18 Nov 10 23:37 No Changes	1s	1s	6s	2min 0s	23min 27s	1s	10s
#16 Nov 10 22:51 No Changes	1s	1s	5s	53s	2s	4s	12s

Permalinks

- Last build (#16), 18 min ago
- Last stable build (#16), 18 min ago
- Last successful build (#16), 18 min ago

Build 19 output did change in the code which is automatically updated by Jenkins



Not secure | a7dd45bd83c344127a69e76ea38f22d9-1422764053.ap-south-1.elb.amazonaws.com

Hello World From K8s 2

Thanks for giving me this task. It helped me learn lots of new things

Manas Jain
+91-8114496703