CDD Project Documentation – 2100030084 Akhil

Create 3 EC2 Instances in AWS

—

Rename servers as JenkinsServer, AnsibleServer, ControllerVM(Kubernetes server)

Jenkin Server Setup

—

Connect to Jenkins server from terminal as root user using ssh
Set password to the root user
Install java-17 ,jenkins , Git

Ansible Server Setup

—

Connect to jenkin server from terminal as root user using ssh command
Set password to the root user
Install ansible and docker

ControllerVM setup

—

Connect to jenkin server from terminal as root user using ssh with .pem key
Set password to the root user
Install Kubectl and kops
sudo su
vim /etc/ssh/sshd_config
passwd root
systemctl restart sshd

Install Kubectl

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux
/amd64/kubectl
chmod +x ./kubectl
sudo mv ./kubectl /bin/kubectl
```

Create an IAM user/role with Route53, EC2, IAM and S3 full access

Attach IAM role to EC2 instance

Install kops

```
curl -LO https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag_name | cut -d '"' -f 4)/kops-linux-amd64

chmod +x kops

sudo mv kops /bin/kops
```

Create a Route53 private hosted zone

create an S3 bucket : aws s3 mb s3://clusters.devops.cdd.in

Expose environment variable:
export KOPS_STATE_STORE=s3://clusters.devops.cdd.in

Create sshkeys before creating cluster: ssh-keygen

Create kubernetes cluster definitions on S3 bucket:

kops create cluster --cloud=aws --zones=ap-south-1a --name=dev.k8s.project.com --dns-zone=project.com --dns private

Create kubernetes cluser :kops update cluster dev.k8s.project.com --yes --admin

Validate your cluster : kops validate cluster --wait 10m

To list nodes: `kubectl get nodes`

Establish Keyless communication between jenkin server to ansible and ansible to controllerVM(Kubernetes Server)

—----------

From jenkin server to Ansible:

in jenkin Server

Generate key: ssh-keygen

Vim /.ssh/id_rsa.pub

Copy that key from that file and paste it in the authorized_keys file in ansible server

The same process should be repeated for ansible server, ControllerVM

Continuous Integration with Jenkins

Create following files:

Create a docker file and push into the git repository

Source Code management configurations

Pulling soucecode from my git repository

Create webhook b/w jenkins and git

Jenkins Configuration
Create new user

Go to Manage Jenkins then Plugins
Then go to available Plugins

Install publish over ssh plugin

Configure ssh server
Set ansible and jenkin server hosts with their respective private ips and give correct credentials.
Once check the server configurations success or not.

Add build Configurations for Jenkins
Configuring jenkin server to copy Dockerfile to /opt directory in ansible from jenkins

Configuring ansible server to create a docker image and push into my docker repository

Add another build configuration for Ansible:

Using these commands to build and push docker image automatically via Ansible

cd /opt

docker image build -t $JOB_NAME:v1.$BUILD_ID.

docker image tag $JOB_NAME:v1.$BUILD_ID docky9/$JOB_NAME:v1.$BUILD_ID

docker image tag $JOB_NAME:v1.$BUILD_ID docky9/$JOB_NAME:latest

docker image push docky9/$JOB_NAME:V1.$BUILD_ID

docker image push docky9/$JOB_NAME:latest

docker image rmi $JOB_NAME:V1.$BUILD_ID docky9/$JOB_NAME:v1.$BUILD_ID docky9/$JOB_NAME:latest

Add post build Configurations:
Run ansible playbook to deploy my application in kubernetes cluster

“ansible-playbook /opt/ansible.yml”
This command runs the ansible.yml and automates the task

Build Creation

Jenkins Build Console Output

15.206.72.248:8080/job/cddproj/16/console

Jenkins

Search (⌘+K) akhil log out

Dashboard > cddproj > #16 > Console Output

- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete build '#16'
- Git Build Data
- Previous Build

Console Output

```
Started by user akhil
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/cddproj
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/cddproj/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/git-user-9/cdd-react.git # timeout=10
Fetching upstream changes from https://github.com/git-user-9/cdd-react.git
 > git --version # timeout=10
 > git --version # 'git version 2.40.1'
 > git fetch --tags --force --progress -- https://github.com/git-user-9/cdd-react.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 7f1c84e599bcc1c4452b62f3ecd1afaa97c1a01e (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 7f1c84e599bcc1c4452b62f3ecd1afaa97c1a01e # timeout=10
Commit message: "Update Dockerfile - luxory"
 > git rev-list --no-walk 7f1c84e599bcc1c4452b62f3ecd1afaa97c1a01e # timeout=10
SSH: Connecting from host [ip-172-31-36-40.ap-south-1.compute.internal]
SSH: Connecting with configuration [Jenkins] ...
SSH: EXEC: completed after 801 ms
SSH: Disconnecting configuration [Jenkins] ...
SSH: Transferred 0 file(s)
Build step 'Send files or execute commands over SSH' changed build result to SUCCESS
SSH: Connecting from host [ip-172-31-36-40.ap-south-1.compute.internal]
SSH: Connecting with configuration [Ansible] ...
SSH: Creating session: username [root], hostname [172.31.42.140], port [22]
SSH: Connecting session ...
SSH: Connected
SSH: Opening exec channel ...
SSH: EXEC: channel open
SSH: EXEC: STDOUT/STDERR from command [cd /opt
docker image build -t cddproj:v1.16 .
docker image tag cddproj:v1.16 docky9/cddproj:v1.16
```

15.206.72.248:8080/job/cddproj/16/console

Dashboard > cddproj > #16 > Console Output

```
d8ad9445f151: Preparing
5498e8c22f69: Preparing
5f70bf18a086: Layer already exists
90678cf22d1e: Layer already exists
a8ae4c8000a4: Layer already exists
5498e8c22f69: Layer already exists
d8ad9445f151: Layer already exists
v1.16: digest: sha256:193c1eecf646bec138cce255f833af51a19acc117482c6767471d9741445863c size: 1369
The push refers to repository [docker.io/docky9/cddproj]
a8ae4c8000a4: Preparing
5f70bf18a086: Preparing
90678cf22d1e: Preparing
d8ad9445f151: Preparing
5498e8c22f69: Preparing
5498e8c22f69: Layer already exists
d8ad9445f151: Layer already exists
a8ae4c8000a4: Layer already exists
90678cf22d1e: Layer already exists
5f70bf18a086: Layer already exists
latest: digest: sha256:193c1eecf646bec138cce255f833af51a19acc117482c6767471d9741445863c size: 1369
Untagged: cddproj:v1.16
Untagged: docky9/cddproj:v1.16
Untagged: docky9/cddproj:latest
SSH: EXEC: completed after 9,406 ms
SSH: Disconnecting configuration [Ansible] ...
SSH: Transferred 0 file(s)
SSH: Connecting from host [ip-172-31-36-40.ap-south-1.compute.internal]
SSH: Connecting with configuration [Ansible] ...
SSH: EXEC: completed after 4,205 ms
SSH: Disconnecting configuration [Ansible] ...
SSH: Transferred 0 file(s)
Finished: SUCCESS
```

REST API Jenkins 2.440.2

Checking kubectl pods

```
cddproj-6f69d7fb46-db8h6   0/1     ContainerCreating   0          5s
cddproj-6f69d7fb46-lg96x   0/1     ContainerCreating   0          5s
[root@ip-172-31-36-63 opt]# kubectl get all
NAME                                   READY   STATUS              RESTARTS   AGE
pod/cddproj-6f69d7fb46-db8h6   0/1     ContainerCreating   0          8s
pod/cddproj-6f69d7fb46-lg96x   0/1     ContainerCreating   0          8s

NAME                 TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/cddproj      LoadBalancer   100.71.177.151   <pending>     8080:31200/TCP   3m55s
service/kubernetes   ClusterIP      100.64.0.1       <none>        443/TCP          179m

NAME                       READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/cddproj    0/2     2            0           3m55s

NAME                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/cddproj-6f69d7fb46    2         2         0       3m55s
[root@ip-172-31-36-63 opt]#
[root@ip-172-31-36-63 opt]#
[root@ip-172-31-36-63 opt]#
[root@ip-172-31-36-63 opt]#
[root@ip-172-31-36-63 opt]#
[root@ip-172-31-36-63 opt]#
[root@ip-172-31-36-63 opt]#
[root@ip-172-31-36-63 opt]#
[root@ip-172-31-36-63 opt]# kubectl get all
NAME                                   READY   STATUS    RESTARTS   AGE
pod/cddproj-6f69d7fb46-db8h6   1/1     Running   0          6m21s
pod/cddproj-6f69d7fb46-lg96x   1/1     Running   0          6m21s

NAME                 TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/cddproj      LoadBalancer   100.71.177.151   <pending>     8080:31200/TCP   10m
service/kubernetes   ClusterIP      100.64.0.1       <none>        443/TCP          3h5m

NAME                       READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/cddproj    2/2     2            2           10m

NAME                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/cddproj-6f69d7fb46    2         2         2       10m
[root@ip-172-31-36-63 opt]#
```

i-056b73c4cb112db12 (Controller VM)

PublicIPs: 13.232.125.6 PrivateIPs: 172.31.36.63

Running My application

From Slave Node - nodes-ap-south-1b.clusters.devops.cdd.in

Node public IP address - 65.1.248.85:31200

Port = 31200