Run a Stateless Application Using a Deployment

You need to have a Kubernetes cluster, and the kubectl command-line tool must be configured to communicate with your cluster.

The GoldenAMI has already been configured to communicate the Kubernetes Cluster. You can confirm the same by login into your Instance terminal and running the below command.

$ kubectl config view *# Show Merged kubeconfig settings.*

1. Deploy an application (ex. nginx) on the kubernetes cluster

   $ kubectl run <your-app-name> --replicas=2 --image=nginx:latest --port=80

The preceding command creates a Deployment object and an associated ReplicaSet object. The ReplicaSet has Two Pods, each of which runs the **nginx** application.

2. Display information about the Deployment:

   $ kubectl get deployments <your-app-name>

   $ kubectl describe deployments <your-app-name>

3. Display information about your ReplicaSet objects:

   $ kubectl get replicasets <your-app-name>
   $ kubectl describe replicasets <your-app-name>

4. Create a **Service object** that exposes the deployment:

   $ kubectl expose deployment <your-app-name> --type=NodePort --name=<your-service-name>

5. Display information about the Service:
   $ kubectl get services <your-service-name>
   NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
   arshad NodePort 100.66.70.157 <none> 80:**32587**/TCP 3m

6. Display detailed information about the Service:

   $ kubectl describe services <my-service>

7. Check the Node details where the POD has been deployed
   $ kubectl get pod -o wide | grep <your-app-name>

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
arshad-5fdff48b48-7r4pg	1/1	Running	0	2m	100.96.4.43	**ip-172-20-55-125.ec2.internal**
arshad-5fdff48b48-gg7j4	1/1	Running	0	2m	100.96.4.44	ip-172-20-55-125.ec2.internal
rajni-deployment-d667	1/1	Running	0	3h	100.96.4.13	ip-172-20-55-125.ec2.internal

ip-172-20-55-125.ec2.internal is the Worker node where the app has been deployed

8. Login to the **AWS Console** and Copy the **Public IP** of the NODE where the POD has been deployed to access the application.

Access the application as shown below

http://<node-public-ip>:NodePort ##see step 5 to check the port

Example

http://18.208.206.161:32587/

Cleaning up

To delete the Service, enter this command:

$ kubectl delete services <service-name>

To delete the Deployment, the ReplicaSet, and the Pods that are running the NGINX application, enter this command:

$ kubectl delete deployment <app-name>