**KUBERNETES**

**NEED FOR KUBERNETES:**

**Chart

Description automatically generated with medium confidence**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated with low confidence**

**CLUSTER:**

Kubernetes manages your virtual servers.

Amazon calls these virtual servers as Elastic Compute Cloud (EC2)

Azure calls them Virtual Machines

Google Cloud calls them Compute Engines

Kubernetes calls them Nodes

So, Kubernetes manages Nodes. To manage 1000s of such nodes, Kubernetes uses managers or Master Nodes.

Diagram

Description automatically generated

So, a Cluster is a combination of nodes and master nodes.

**Create a Kubernetes cluster:**

Go to Kubernetes Engine service in Console and then click Create Cluster. It will take some time to create.

**Connecting to Kubernetes Cluster:**

After creating the cluster, we need to connect to it. Go to your cluster.

**Graphical user interface, text, application, email

Description automatically generated**

Then Click on Connect.

**Graphical user interface, text

Description automatically generated**

This will give a command. Copy this command.

**Graphical user interface, text, application, email

Description automatically generated**

To Execute this command, we will use Cloud Shell. Click on Activate Shell to open Cloud Shell.

**Graphical user interface, application

Description automatically generated**

Then click to open Cloud Shell in new window:

**Graphical user interface, text, application

Description automatically generated**

Now we can execute commands in this shell.

Graphical user interface, text, application

Description automatically generated

Paste the previously copied command here. This will connect with our cluster.

**Connect to cluster:**

gcloud container clusters get-credentials ldeepak-cluster --zone us-central1-c --project clear-wind-342111

**Check Kubernetes version:**

kubectl version

**Create Deployment:**

kubectl create deployment hello-world-rest-api --image=in28min/hello-world-rest-api:0.0.1.RELEASE

**Exposing the Deployment:**

kubectl expose deployment hello-world-rest-api --type=LoadBalancer --port=8080

**Accessing the Exposed endpoint:**

Go to Services & Ingress:

**Graphical user interface, application

Description automatically generated**

Click on the Endpoint:

**Graphical user interface, text, application, email

Description automatically generated**

This will open it in new browser window:

**Graphical user interface, text, application, website

Description automatically generated**

**Here you can access API at** <http://35.239.100.38:8080/hello-world>

**Graphical user interface, text, application, website

Description automatically generated**

**get command:**

**kubectl get <thing>**

**thing={events, pods, replicasets, deployments, services, etc...}**

**explain command:**

**kubectl explain <thing>**

**thing={events, pods, replicasets, deployments, services, etc...}**

**PODS:**

Pod is a collection of containers that can run on a host.

Chart, treemap chart

Description automatically generated

**kubectl get pods:**

****

**kubectl describe pod <pod-name>:**

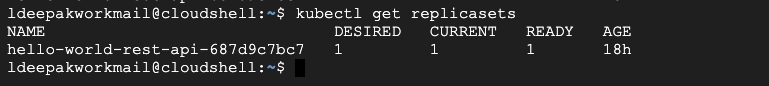
**Text

Description automatically generated**

**REPLICASETS:**

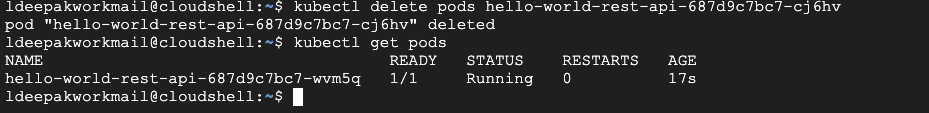
Ensures that a specified number of pod replicas are running at any given time.

**kubectl get replicasets:**



We can see we have one pod available. So even if we delete this pod, replicaset will ensure that one pod is available, so it will create another pod.

**kubectl delete pods <pod-id>:**



To have some more pods, we can scale our deployment to create more pods.

**kubectl scale deployment <pod-name> --replicas=<desired replica count>**

**Text

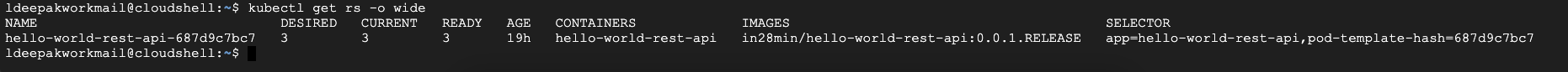
Description automatically generated**

**DEPLOYMENT:**

A deployment allows you to describe an application’s life cycle, such as which images to use for the app, the number of pods there should be, and the way in which they should be updated.

**Kubectl get pods -o wide:**

Gives more info about the replica set. Like which image it is tied to.



To set a new image (i.e., changing the application to new version)

**Kubectl set image deployment <deployment-name> <container-name>=<image-name>**