Assignment and Exam Content

Kubernetes Engine

Always Delete your Cloud Resources to Avoid $$ Charges.

Kubernetes lab

Kubernetes Lab Contains – Three major areas below to say complete Lab ☺

A

Launch Kubernetes Cluster

Choose different configurations to Launch Cluster.

B

Deploy Application

Once the cluster is ready , you can deploy Sample Application Bundle. Available to deploy.

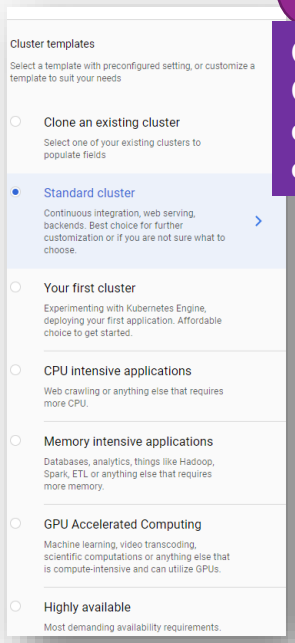
3

Manage Application on Kubernetes Cluster.

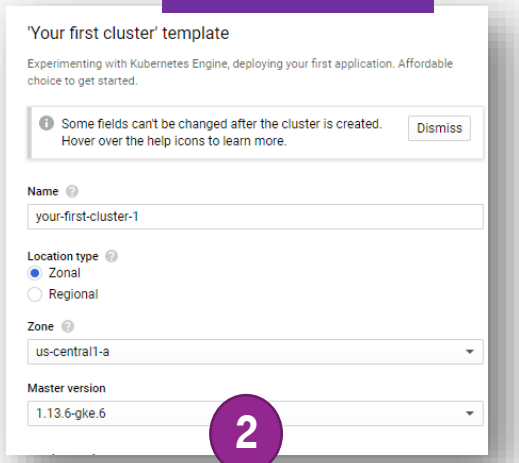
Monitor Cluster , Log , Add Node Pool etc.

A

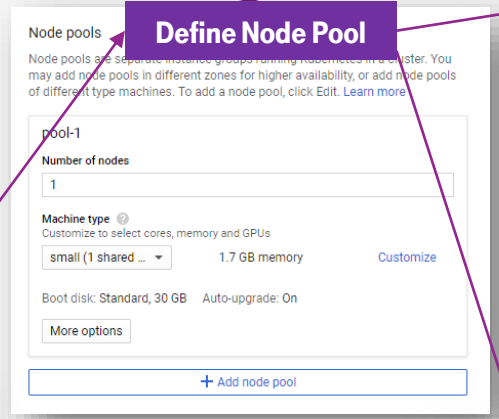
Kubernetes Engine Creations – Steps in Summary



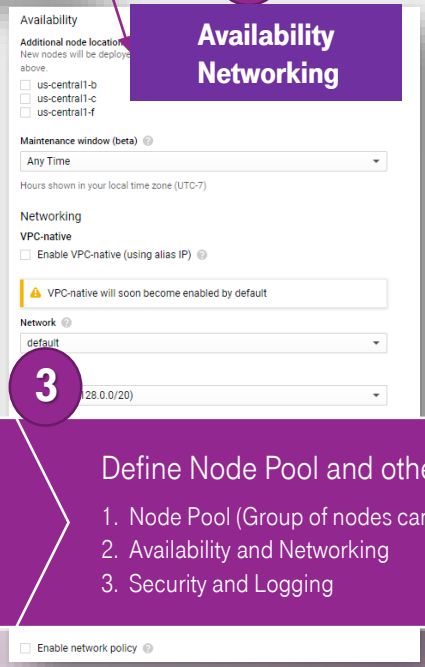
1 **Cluster Creation Choose Options or define your own configuration**



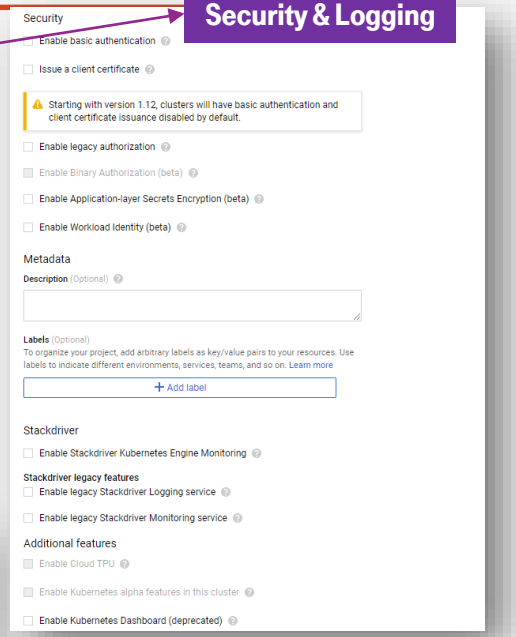
2 **Cluster location and Master Node**



3 **Define Node Pool**



3.1 **Availability Networking**



3.2 **Security & Logging**

1

Choose from predefined template

- Google Provides Multiple template for you to use /reconfigure it while using it.
- You can define your own configuration.
- Or you can clone existing cluster.

2

Cluster Location and Master Node

- Configuration – regional /zonal cluster location for it.
- Provide Master Node Configuration.

3

Define Node Pool and other configurations

1. Node Pool (Group of nodes can be managed separately)
2. Availability and Networking
3. Security and Logging

A

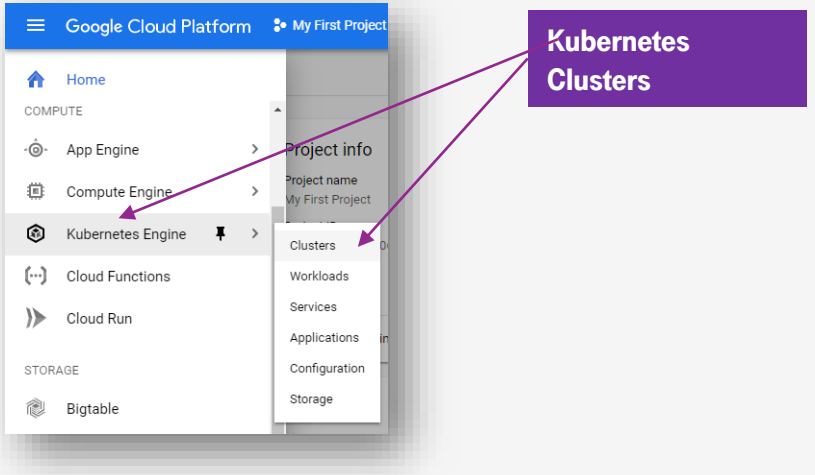
1 Cluster Selection

Choose from predefined template
- Google Provides Multiple template for you to use /reconfigure it while using it.
- You can define your own configuration.
- Or you can clone existing cluster.

Cluster Location and Master Node
- Configuration – regional /zonal cluster location for it.
- Provide Master Node Configuration.

Define Node Pool and other configurations
1. Node Pool (Group of nodes can be managed separately)
2. Availability and Networking
3. Security and Logging

1 Go To -> COMPUTE -> Kubernetes Engine-> Cluster



2 There are multiple Templates available to choose.

e.g. CPU Intensive, Memory Intensive, GPU Accelerations or Generic High Availability

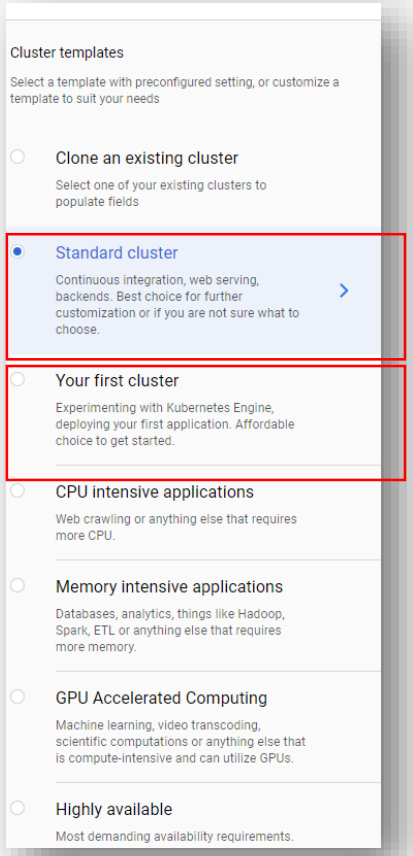
The only change between configurations are

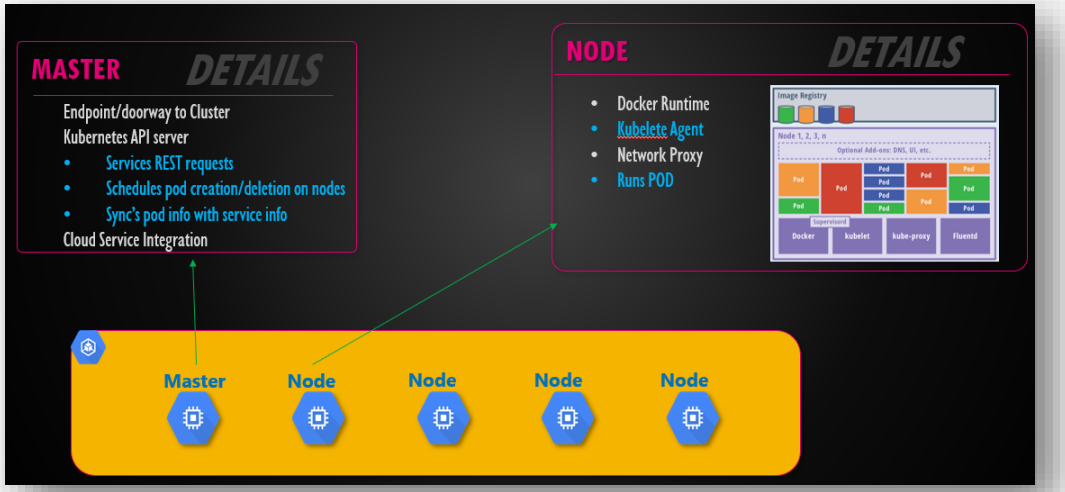
1. Node Configurations – CPU, Memory size
2. No of Nodes
3. HTTP Load Balancer
4. Logging and Monitoring

You can change/adjust all configurations in all options.

For Current lab

Choose Standard or Your first Cluster and make sure that you have only one node in node pool





A

2 Cluster and Master Configuration

Choose from predefined template
- Google Provides Multiple template for you to use /reconfigure it while using it.
- You can define your own configuration.
- Or you can clone existing cluster.

Cluster Location and Master Node
- Configuration – regional /zonal cluster location for it.
- Provide Master Node Configuration.

Define Node Pool and other configurations
1. Node Pool (Group of nodes can be managed separately)
2. Availability and Networking
3. Security and Logging

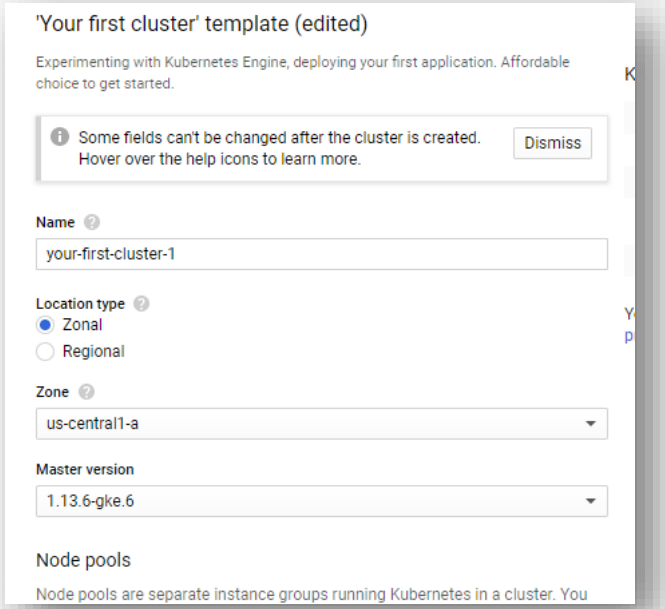
2 Cluster and Master Configurations

Name - : Choose Any name

Locations : You can say Zonal

Zone : Choose Your preferred Zone.

Master Version : You can choose different Available Kubernetes versions. You can leave default.



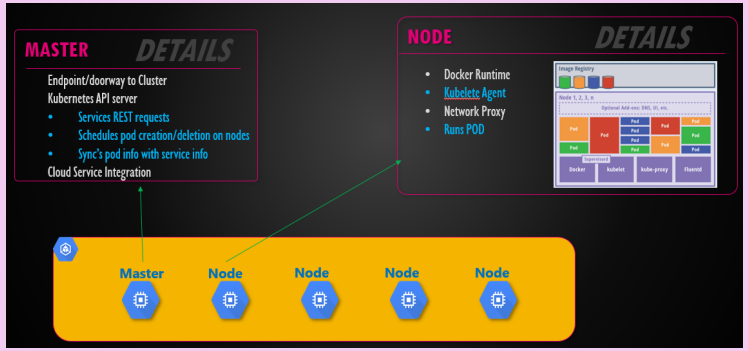
Things to remember

1. Kubernetes works in Master and Slave mode.

2. You can have Master Configuration as well as Nodes configurations

3. You need to have one Master and at least one Slave Node.

4. Cluster Master is endpoint to cluster.



A

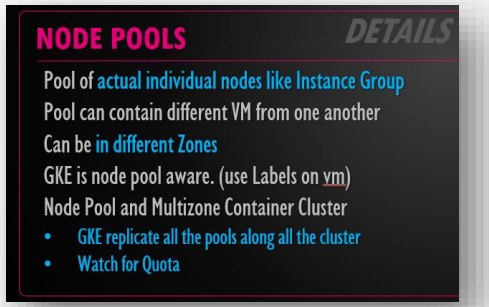
2 Node Pool and Availability

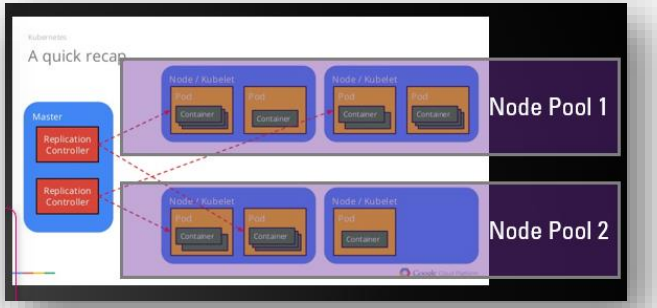
Choose from predefined template
- Google Provides Multiple template for you to use /reconfigure it while using it.
- You can define your own configuration.
- Or you can clone existing cluster.

Cluster Location and Master Node
- Configuration – regional /zonal cluster location for it.
- Provide Master Node Configuration.

Define Node Pool and other configurations
1. Node Pool (Group of nodes can be managed separately)
2. Availability and Networking
3. Security and Logging

2 Node pools are separate instance groups running Kubernetes in a cluster. You may add node pools in different zones for higher availability, or add node pools of different type machines.





For Current lab

Enter Number of Node : 1

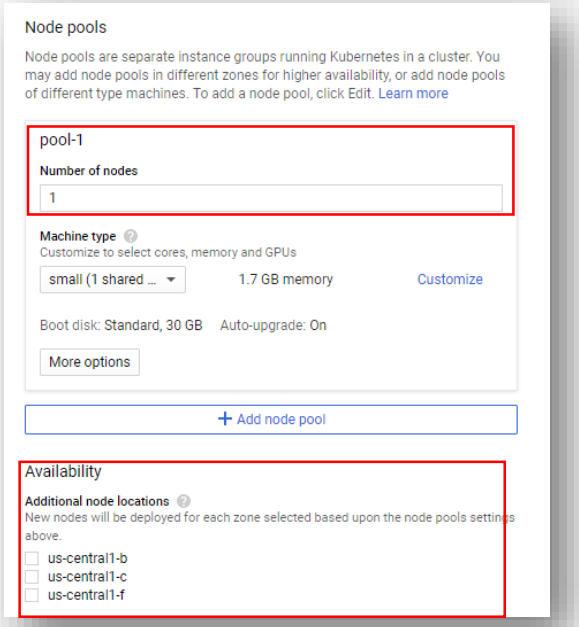
Go to Availability : Additional Node Locations

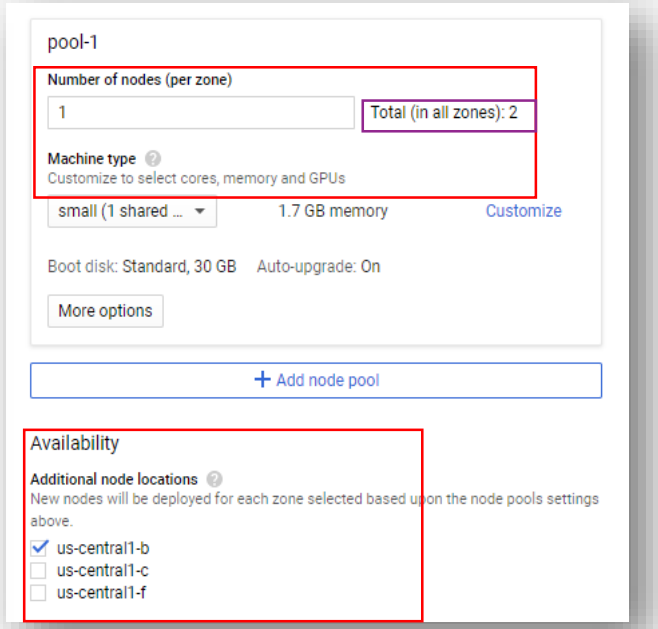
Click checkbox for Multiple Zone

observe Label Changes and Total Nodes numbers

$$$ Impacting

Do not choose any Zone (in Additional Node Locations) , GCP Charges based on number of nodes.





A

2 Networking, Security, Logging

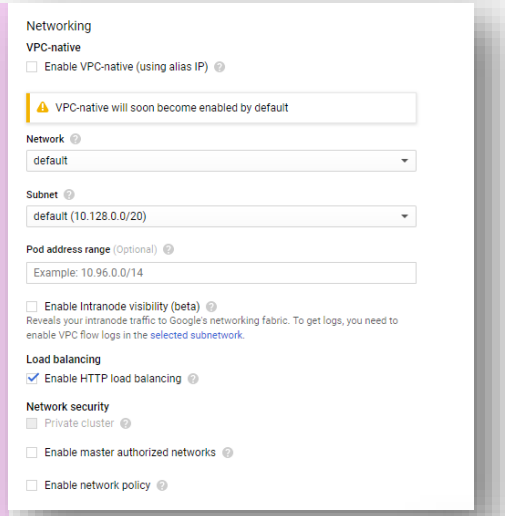
Choose from predefined template
- Google Provides Multiple template for you to use /reconfigure it while using it.
- You can define your own configuration.
- Or you can clone existing cluster.

Cluster Location and Master Node
- Configuration – regional /zonal cluster location for it.
- Provide Master Node Configuration.

Define Node Pool and other configurations
1. Node Pool (Group of nodes can be managed separately)
2. Availability and Networking
3. Security and Logging

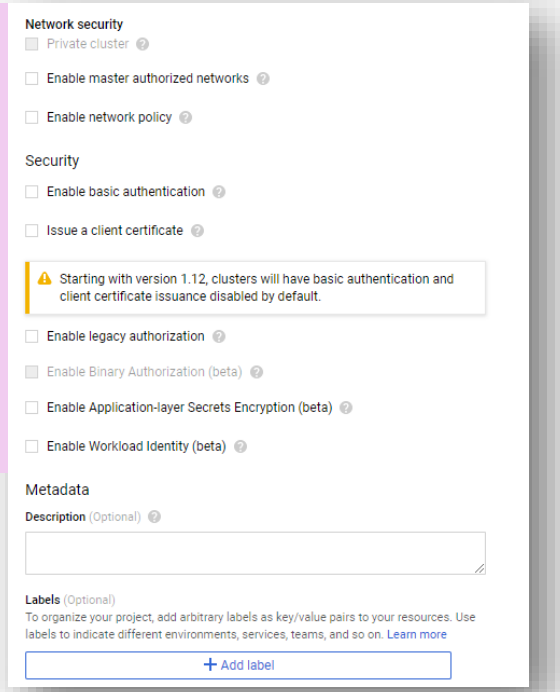
Networking

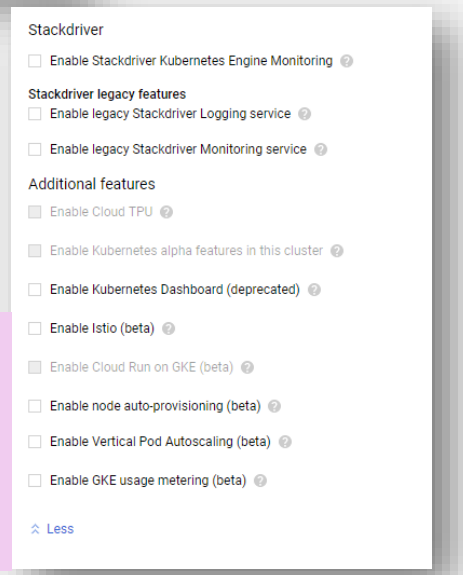
- You can choose default network or you can choose your custom Network.
- If you want you can have POD ip Address range
- Internode Visibility will provide internode communications logs.. _ You don't need that for this demo.
- You can enable Load Balancing.
- **KEEP DEFAULT**



Security

- You have option to choose Basic authentication (User and Password) Not recommended.
- You can have Client Certificate
- You can have legacy (old school) or choose RBAC (RBAC recommended)
- You can have Application level secrets stored in etcd
- **KEEP DEFAULT**





Logging

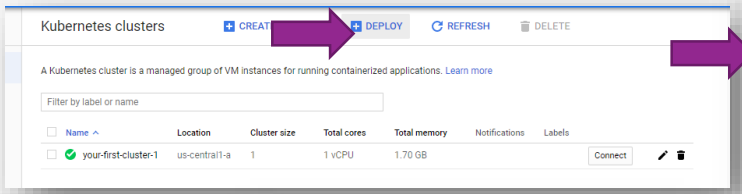
- You can enable Stackdriver logging for Kubernetes Cluster
- You can also optionally have Istio and Other beta features .
- **KEEP DEFAULT**

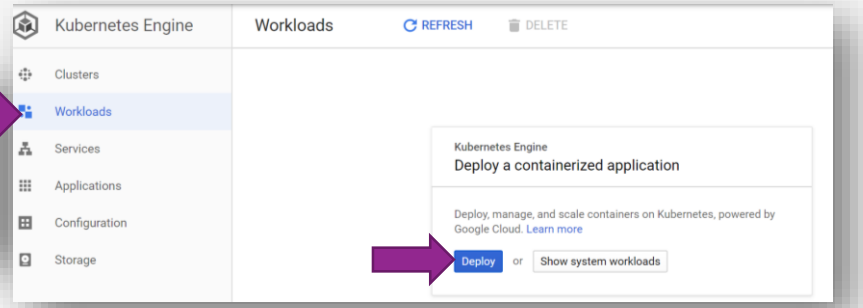
B

1

Deploy Application

1 Use Sample Application to deploy. There are two ways to start deployment process shows in below snippets 1. Cluster Home page or Workloads.



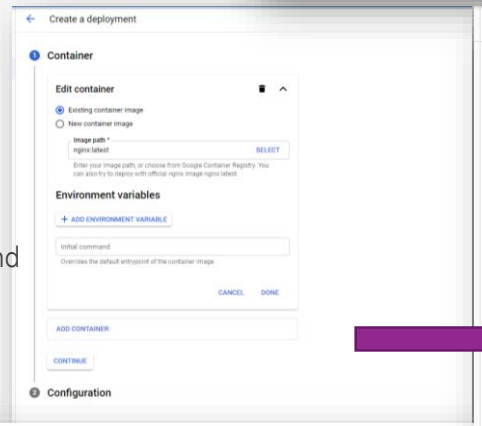


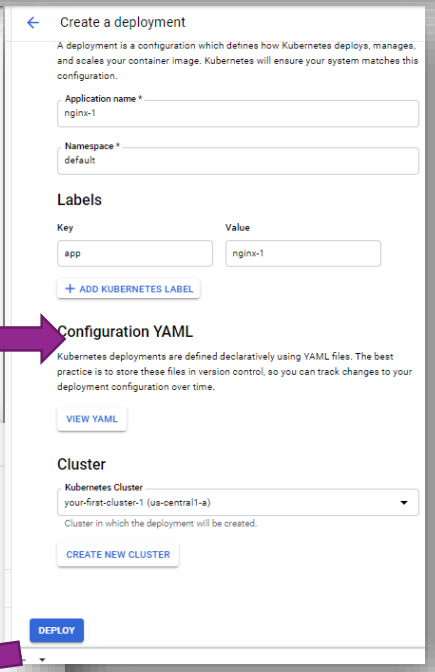
2

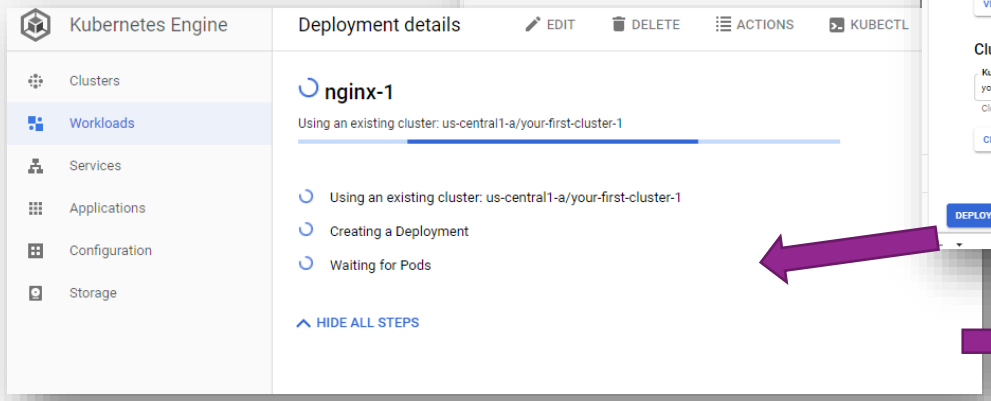
1. Use Existing Image
2. Click Continue
3. Choose Application Name

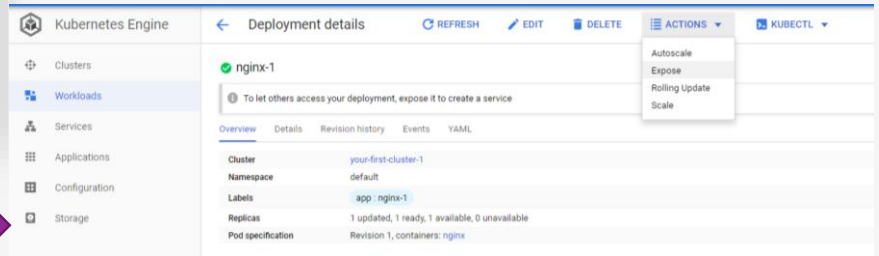
Click Create

4. Expose Service -> go to Workloads , Click on Service and go to actions Expose.









2 **Alternatively – You can get sample Image from**

https://cloud.google.com/kubernetes-engine/docs/quickstart

Using Command Line

See next Page

B

1

Create Cluster and Deploy Application – CLI (gcloud)

1

Set default Configurations

Set Project

$ gcloud config set project *[project-id]*

Set Zone

$ gcloud config set compute/zone *[Your Default Zone]*

Set Region

$ gcloud config set compute/region *[Your default Region]*

Sample output

$ gcloud config set project tokyo-data-243419

Updated property [core/project].

$ gcloud config set compute/zone us-central1

Updated property [compute/zone].

$ gcloud config set compute/region us-central1-a

Updated property [compute/region].

2

Create Cluster

$ gcloud container clusters create gcptraince–num-nodes=1

Alternatively you can choose different options to create cluster. E.g.

$ gcloud container clusters create [CLUSTER_NAME] –image-type ubuntu

$ gcloud container clusters upgrade –image-type cos [CLUSTER_NAME] \

[–node-pool [POOL_NAME]]

Auto Scaling

$ gcloud container clusters create [CLUSTER_NAME] –num-nodes 1 \

–enable-autoscaling –min-nodes 1 –max-nodes 3

You can enable autoscaling for existing cluster

$ gcloud container clusters update [CLUSTER_NAME] –enable-autoscaling \

–min-nodes 1 –max-nodes 10 –zone [COMPUTE_ZONE] –node-pool default-pool

Sample output

$ gcloud container clusters create gcptraince –num-nodes=1

Creating cluster gcptraince in us-central1-a... Cluster is being health-checked (master is healthy)...done.

Created [https://container.googleapis.com/v1/projects/tokyo-data-243419/zones/us-central1-a/clusters/gcptraince].

To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/us-central1-a/gcptraince?project=tokyo-data-243419

kubeconfig entry generated for gcptraince.

NAME	LOCATION	MASTER_VERSION	MASTER_IP	MACHINE_TYPE	NODE_VERSION	NUM_NODES	STATUS
gcptraince	us-central1-a	1.12.8-gke.6	35.202.63.71	n1-standard-1	1.12.8-gke.6	1	RUNNING

B

1

Create Cluster and Deploy Application – CLI (gcloud)

1

Get Authentications to connect to Cluster using kubectl.

$ gcloud container clusters get-credentials [Your Cluster]

Sample output

$ gcloud container clusters get-credentials gcptrainee

Fetching cluster endpoint and auth data.

kubeconfig entry generated for gcptrainee.

2

Deploy and run application

$ kubectl run hello-server –image gcr.io/google-samples/hello-app:1.0 –port 8080

Sample output

$ kubectl run hello-server –image gcr.io/google-samples/hello-app:1.0 –port 8080

kubectl run –generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version. Use kubectl run –generator=run-pod/v1 or kubectl create instead.

deployment.apps/hello-server created

Export Service

$ kubectl expose deployment hello-server –type LoadBalancer

$ kubectl expose deployment hello-server –type LoadBalancer \

> –port 80 –target-port 8080

service/hello-server exposed

.

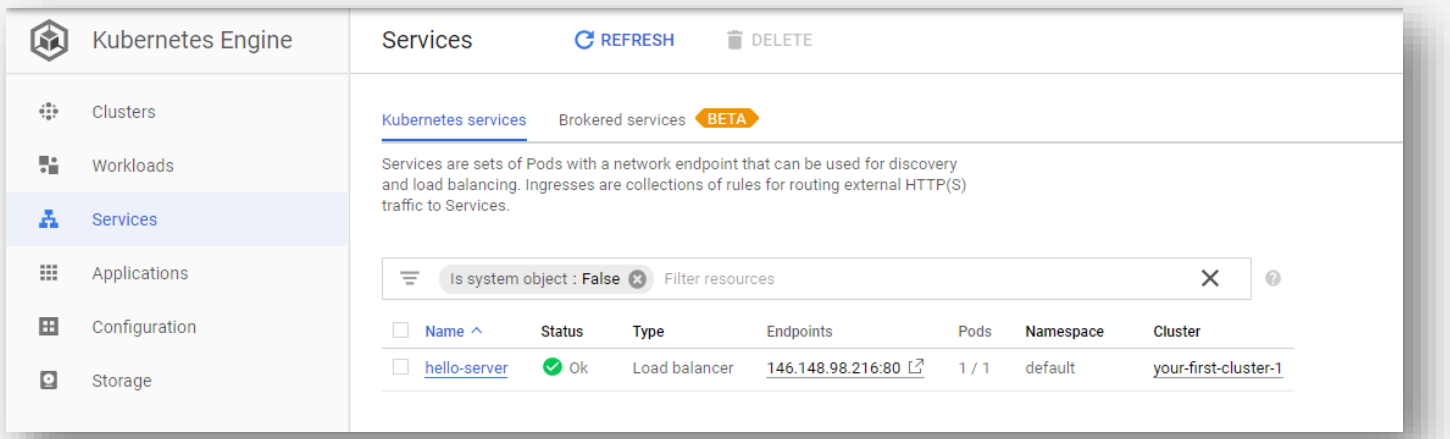
View Application

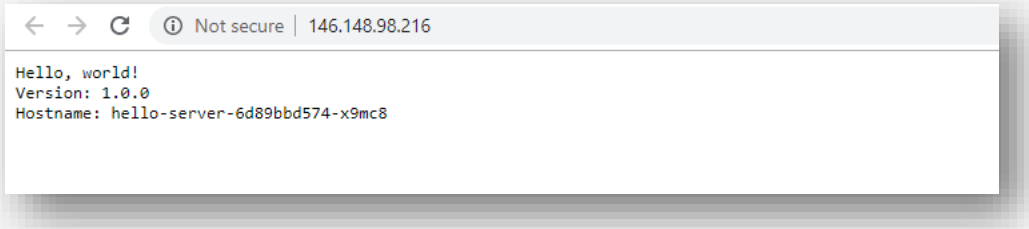
Go To Service

COMPUTE -> KUBERNETES ENGINE -> SERVICES

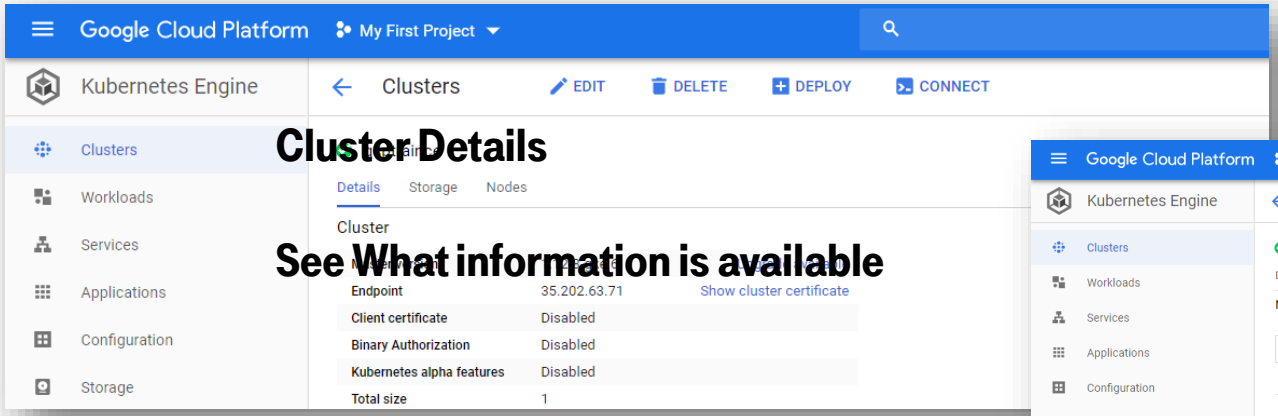
Click on Endpoint

- You should see





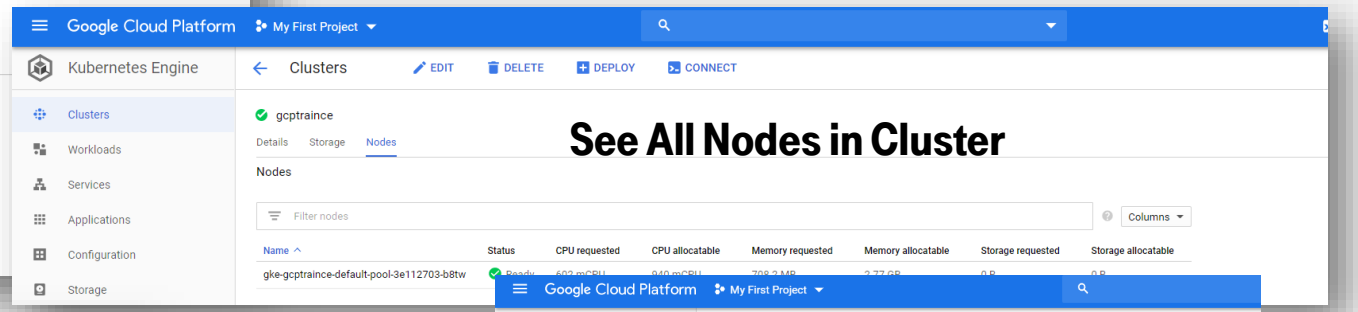
Now Browse Different Screens on Console and Try different CLIS



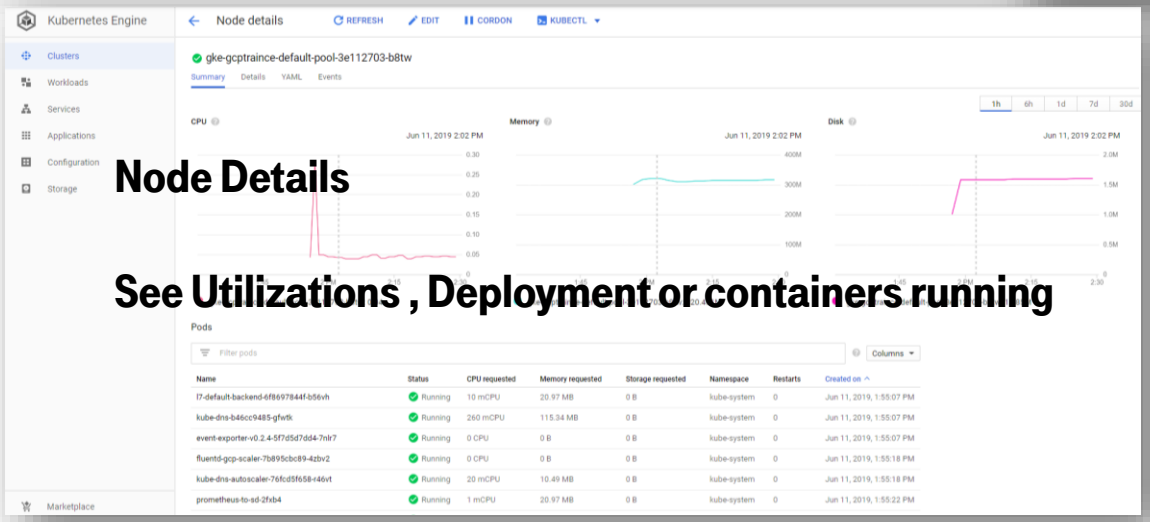
Cluster Details

See What information is available

Delete Your Cluster after work

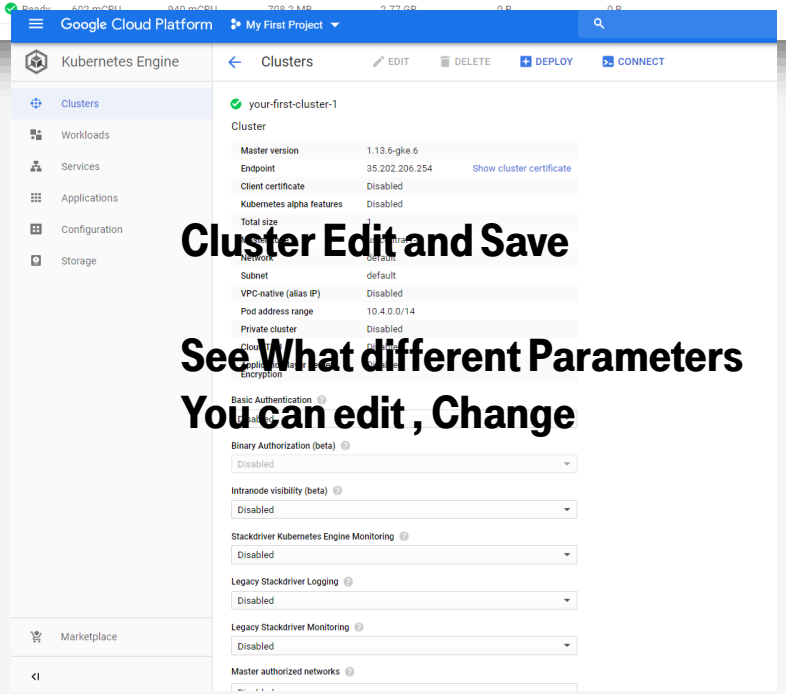


See All Nodes in Cluster



Node Details

See Utilizations , Deployment or containers running



Cluster Edit and Save

See What different Parameters

You can edit , Change

Details Also Available https://cloud.google.com/kubernetes-engine/docs/tutorials/hello-app

Deploy : Docker Container Application - Important for Exam

1. Config base setup

Your earlier default setup can be used -> project, Region , Zone

2. Sample App

Use Google Sample Code

$ git clone https://github.com/GoogleCloudPlatform/kubernetes-engine-samples cd kubernetes-engine-samples/hello-app

3. Build Docket Image

$ docker build -t gcr.io/${PROJECT_ID}/hello-app:v1 .

$ docker images

4. Get Authorization for Docker and Push Image to Container Registry

$ gcloud auth configure-docker

$ docker push gcr.io/${PROJECT_ID}/hello-app:v1

5. Create Cluster

Your earlier default setup can be used -> project, Region , Zone

6. Get Cluster Credentials for kubctl Command

$ gcloud container clusters get-credentials *[your Cluster]*

7. Deploy Application

$ kubectl run hello-web --image=gcr.io/${PROJECT_ID}/hello-app:v1 --port 8080

8. Get the information on Deployed Pods

$ kubectl get pods

9. Expose Service

$ kubectl expose deployment hello-web --type=LoadBalancer --port 80 --target-port 8080

Get IP address of service (http://External IP) to access application - Output of following command
$ kubectl get service

10. Adjust Different Parameters

Scale Your Application (Not Cluster – For Cluster you have to use gcloud command)

$ kubectl scale deployment hello-web --replicas=3

Check – if application has now 3 pods running

$ kubectl get deployment hello-web

$ kubectl get pods

New Version Deployment

$ docker build -t gcr.io/${PROJECT_ID}/hello-app:v2 .

Delete Your Cluster after work

Important For Exam

- How to Get Current Configurations of Cluster
- Node pool and Configurations
- Availability & Autoscaling Configurations
- Deployment, Service, POD Monitoring
- Secrets Management (remember etcd)
- How to get Cluster to latest version of Kubernetes
- Kubectl vs gcloud Commands. – Purposes. -> Example all Cluster Management is done by gcloud , And Application deployment and management like scaling , exposing, Versioning etc is done by kubectl.

Delete Your Cluster after work

End of KUBERNETES ENGINE lab

Always Delete your Cloud Resources to Avoid $$ Charges.