## Useful commands

cd myapp

docker build -t acloud-guru-ch1-myapp:blue .

docker tag acloud-guru-ch1-myapp:blue gcr.io/ravi-gcloud01-user01-prj01/myapp:blue

docker push gcr.io/ravi-gcloud01-user01-prj01/myapp:blue

watch kubectl get pods

> gcloud container clusters get-credentials my-cluster-1 --zone=us-central1-a

Fetching cluster endpoint and auth data.

kubeconfig entry generated for my-cluster-1.

> gcloud container clusters list

NAME LOCATION MASTER\_VERSION MASTER\_IP MACHINE\_TYPE NODE\_VERSION NUM\_NODES STATUS

my-cluster-1 us-central1-a 1.14.10-gke.24 34.71.146.111 g1-small 1.14.10-gke.24 1 RUNNING

> kubectl kubectl.1.13 kubectl.1.14 kubectl.1.15 kubectl.1.16 kubectl.1.17 kubectx kubens

> kubectl get nodes

NAME STATUS ROLES AGE VERSION

gke-my-cluster-1-default-pool-2db91e44-c4lh Ready <none> 70m v1.14.10-gke.24

> kubectl run nginx --image=nginx

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/nginx created

> kubectl get nodes

NAME STATUS ROLES AGE VERSION

gke-my-cluster-1-default-pool-2db91e44-c4lh Ready <none> 74m v1.14.10-gke.24

> kubectl get services

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.32.0.1 <none> 443/TCP 75m

> kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-7db9fccd9b-rf894 1/1 Running 0 78s

> kubectl expose deployment nginx --port=80 --type=LoadBalancer

service/nginx exposed

kubectl get events

kubectl describe pod <name of pod>

To deploy from yaml directly

kubectl apply -f <yaml file with deployment/service>

kubectl describe deployment <name of deployment>

kubectl delete pod <pod name>

kubectl get pods

kubectl get pods -o=custom-columns=NODE:.spec.nodeName,NAME:.metadata.name

-o is to modify the output and add extra columns

Exclude nodes from scheduling any pods further:

kubectl taint nodes <node name> key=value:NoSchedule

Untaint the node:

kubectl taint nodes <node name> key:NoSchedule-

To roll-out an update, use the same command, use --record to record this command so that this can be seen in roll-out history:

kubectl apply -f <yaml file with deployment> --record

kubectl rollout status deployment.v1.apps/<Deployment.metadata.name>

kubectl rollout history deployment.v1.apps/<Deployment.metadata.name>

Undo the deployment to the last successful Deployment:

kubectl rollout undo deployment.v1.apps/<Deployment.metadata.name>

Get the logs:

kubectl logs pod/<pod-name>

kubectl logs --selector run=<label>

Build/push the images to GCR without using Docker commands:

> gcloud builds submit --tag gcr.io/<project>/<artifact>/<version>

----------------------------------------------

## Random Tips

GcePersistentDisk can only be accessed by one pod so updating the pod could cause an issue as a new pod will try

to access the disk. This can be prevented by adding below property to the Deployment file,

spec:

Strategy:

type: Recreate

Pod identifier = podname.default.svc.cluster.local

----------------------------------------------

## ACloudGuru

### Volume Lab

Create both the volumes first.

> kubectl get pvc

Store a secret:

> kubectl create secret generic(//type) mysql(//name) --from-literal=password(//KeyName)=MYSQLPASSWORD(//literalStringValue)

Deploy mysql Deployment

Deploy mysql Service

Cleanup:

Delete wordpress' Service, Deployment, VolumeClaim

And repeat for Mysql

----------------------------------------------

### Secrets

To obfuscat the secret data

Can be used as ENV Vars or Volumes

Encoded, not encrypted

Encryption with KMS in Beta mode

ConfigMaps:

Decouple the configuration data from the container

Can be created from files, directories and literal key-value pairs

Can be used as ENV vars or Volumes

### Accessing external services (Endpoints)

Create Service (ClusterIP, LB) with no selector (so no connection with Pod) and

K8s looks for an Endpoint with the same name.

Endpoints map external services to Service objects.

Discovery via internal DNS

Endpoints can map to multiple IPs as well (then round robin)

ExternalName within Service spec doesn't require Endpoints Object

SideCars

Containers running along with our apps.

Userful for Proxies.

-------------------------------------------------

### Lab: Maintaining a Service with Unhealthy Pods

Enable Cloud SQL APIs and spin up an SQL instance, and get the full instance name.

Proxy Docker image: cloudsql-docker/gce-proxy

Service Accounts = non human user accounts that can be used by the applications

> gcloud iam service-accounts create cloudsqlproxy

bind it with the role and project

> gcloud projects add-iam-policy-binding tim-acloud-guru<project-name> --member serviceaccount:cloudsqlproxy@tim-acloud-guru.iam.gserviceaccount.com<name of the service account> --role roles/cloudsql.client<roles>

Create a key that SA can use:

> gloud iam service-accounts keys create ./sqlproxy.json --iam-acount cloudsqlproxy@tim-acloud-guru.iam.gserviceaccount.com

Create a Secret out of that file.

> kubectl create secret generic cloudsql generic cloudsql-instance-credentials --from-file=credentials.json=./sqlproxy.json

Watch progress of the pods using,

> kubectl watch <deployment-name>

Set the timit-limit to quite the update if readiness probes are failing:

> kubectl patch deployments.v1.apps/myapp-deployment -p '{"spec":{"progressDeadlineSeconds":120}}'

> kubectl rollout status deployment.v1.apps/myapp-deployment

--------------------------------------------------

### Deployment Patterns

Rolling updates

Gradually replace pods

Specify threshold for failed pods

Canary deployments

Combine multiple Deployments with a single Service

Small subset of traffic will be routed

(can be automated with the tool called Spinnaker)

BlueGreen deployments

Maintain 2 versions of your application deployment

Switch the traffic with the Service selector

-----------------------------------------------------

### Autoscaling

HorizontalPodAutoscaler

based on CPU and Memory

custome application metrix supported

stackdriver metrix also supported

VerticalPodAutoscaler

CPU and RAM

updateMode: auto

restarts the pod

Node-pool autoscaler

## COURSERA

### WEEK-3

-----------------

#### Resources Hierarchy basics:

Cloud Resources -> Projects -> product A (Folders) -> Team A (Folders) -> Dept. A (Folders) -> Company (Org. Node)

Resources -> Projects -> Folders -> Org. Node

Tip: project does not have to reside in a folder in order to have a parent.

In this instance, Project 5 inherits directly from the org node, example.com.

Project ID (Chosen by us) and number (assigned by gcp) - unique and immutable

Name - doesn't need to be unique, also mutable, chosen by us

Resources inherit the policy from parents.

Roles: Primitive, Predefined, Custom

Primitive Roles:

Coarse-grained

Owner, Editor, Viewer, Billing Admin

Predefined Roles:

Fine-grained collection of roles. ex. InstanceAdmin for ComputeEngine

GSuites customer can edit the users from Admin console.

Non GSuites customer can do the same in Cloud Identity Domain.

APIs Explorer - interactive tool lets us try Google APIs using a browser.

Client Libraries:

Cloud Client Libraries - community owned

Google API Client Libraries - Open source

Billing:

Billing is must to use GCP services

To change the billing of a project, you must be an owner of the project and Billing Admin on the billing account

Setting Budget doesn't cap API usage.

CLI:

CloudShell can be accessed via Browser, it comes with SDK preinstalled.

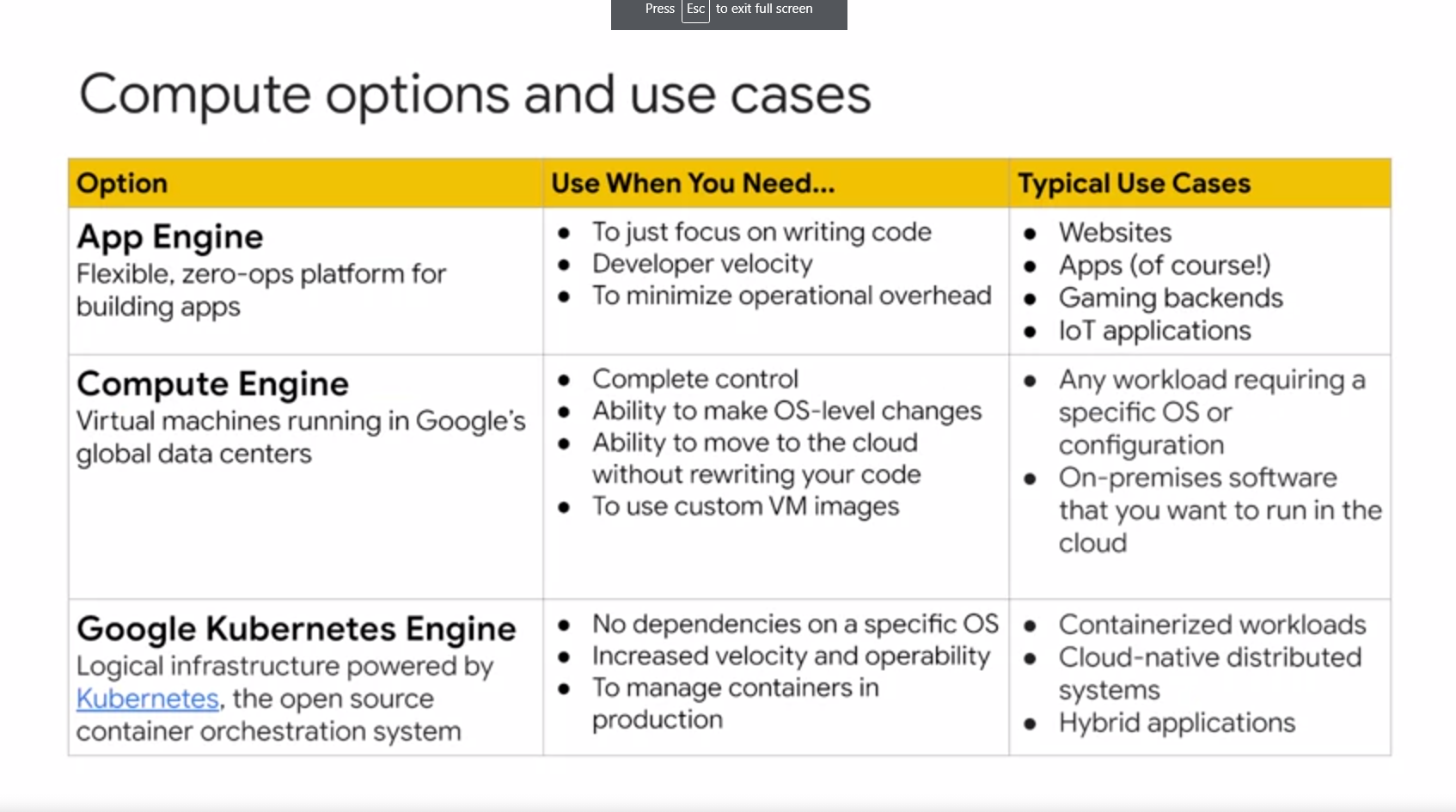
SDK includes - gcloud, gsutil (cloud storage), bq (bigQuery)

### WEEK-4

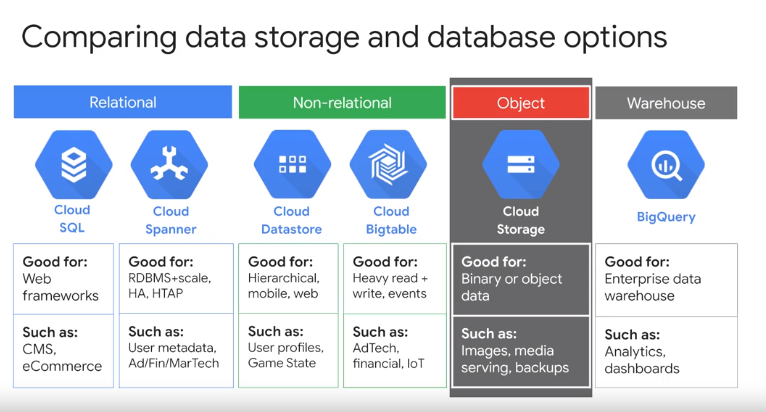
------------------

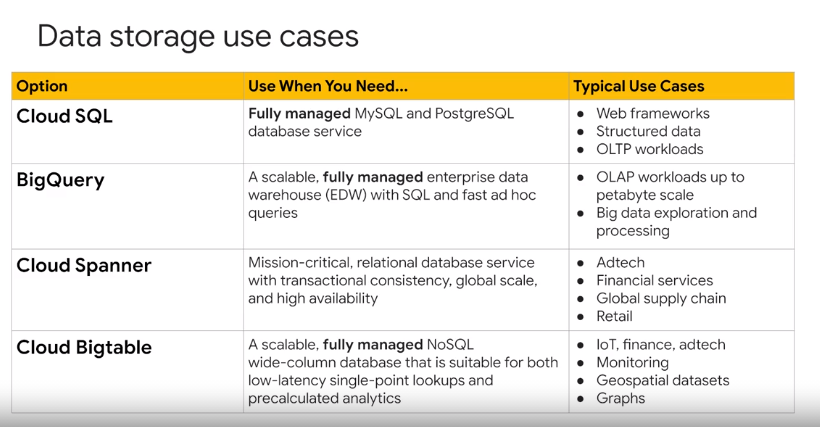
#### Pricing Calculator

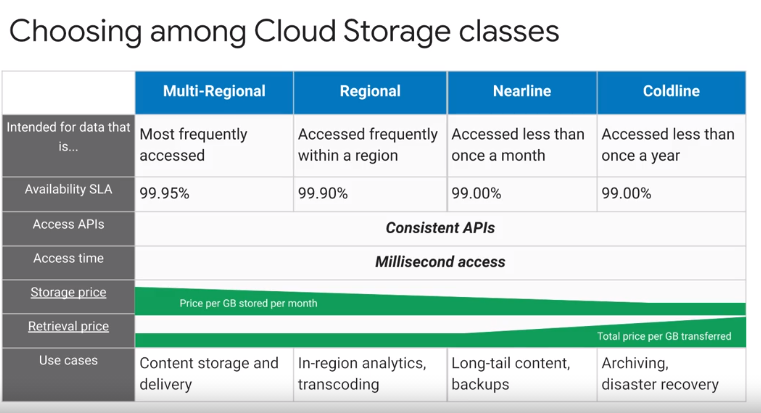
#### Planning the Compute Engine

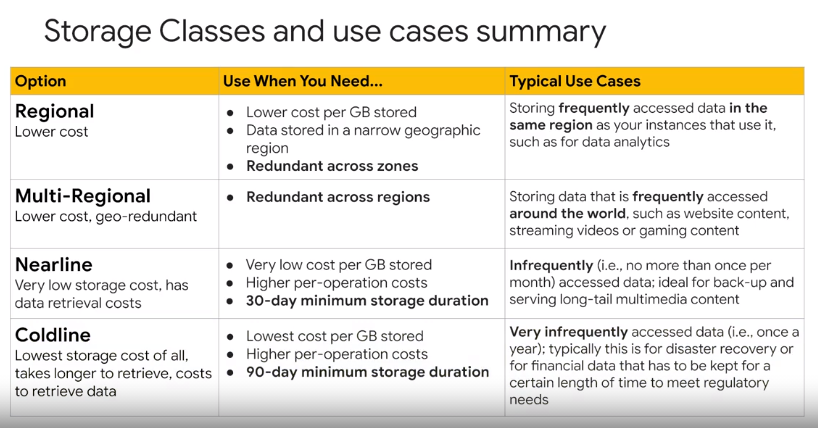


#### Planning the Data Storage







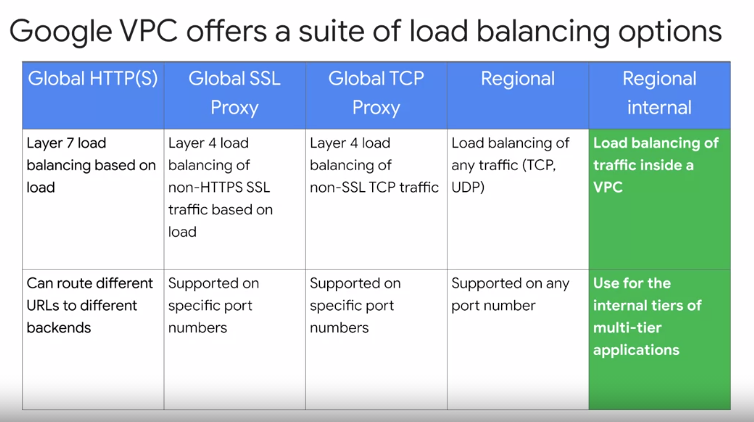


All storage options cost the fees per GB per month.

Multi-regional is the costliest and Coldline is the cheapest option.

Nearline incurs the access fees per GB and Coldline incurs the higher access fees per GB

#### Planning the Network Resources



Global (1,2,3) - also provides IPv6 termination, Balancing the load across the regions, must use premium network tier

Regional (4,5) - IPv4 termination, Balancing the load within the region, Standard network tier

External (1,2,3,4)

Internal (5)

===================================

### QWIKLABS QUESTS

1. Monitoring and Logging (Stackdriver Fundamentals Quest)   
   <https://www.qwiklabs.com/quests/35>