## 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**

**Global Flags:**

* **--help, -h**
* **--project <projectId>**
* **--account <Account>**
* **--filter (alternative of grep)**
* **--format (can choose json, yaml, csv. Can pipe (“I”) json to jq commands for further processing.**
* **--quiet, -q**

**Config Properties:**

* Values entered once and used by any command, however, command can override them.
* **> gcloud config set <property> <value>**
* **> gcloud config get-value <property>**
* **> gcloud config unset <property>**
* Used very often for account, project, region and zone. Below properties can be set,
  + **core/account or account**
  + **core/project or project**
  + **compute/region**
  + **compute/zone**

**To check the current project and account:**

* **> gcloud config list**
* **> gcloud config get-value project**

**Managing the configuration:**

* Interactive workflow to set the common properties in a config with **> gcloud init**
* To list all the configurations we have **> glcoud config configuration list**
* To list all the properties from an active configuration **> gcloud config list**
* To create a new configuration **> gcloud config configuration create <name>**
* To activate the configuration **> gcloud config configuration activate <name>**
  + **IS\_ACTIVE** column says if the configuration is active or not.
* To use a particular configuration for only one command then use a flag **> --configuration <name>**
* To list all the properties from an inactive configuration,
  + **> gcloud config configuration describe <configName>**
  + **> gcloud --configuration=<name> config list**
* The settings we use in glcoud is different from the settings we use in the Console.

**To check the services/api available/enabled for the current project:**

Project gets some default list of services enabled ex. Storage, bigQuery, logging, monitoring etc.

* **> gcloud services list**
* **> gcloud services list --enabled (this is a default flag)**
* **> gcloud services list --available**
* **> gcloud services list --available | grep compute**
* **> gcloud services list -h**
* **> gcloud projects list --filter='name:production\_v1'**
* **> gcloud services list --project <project-id>**
* **> gcloud services enable api-name**
* If an API is not enabled you will get **HTTPError 403**

**Filter:**

* **> gcloud topic filters**

**> 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>**

**> watch kubectl get pods ??**

**> kubectl get pod**s

**> 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>**

**> kubectl config get-contexts**

kubectl config get-contexts displays a **list of contexts as well as the clusters that use them.**

**> kubectl config get-contexts -o=name**

**> kubectl config use-context <name>**

GCloud Commands Documentation:

<https://cloud.google.com/sdk/gcloud/reference>

GCloud Commands Cheatsheet:

<https://cloud.google.com/sdk/docs/cheatsheet>

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

## 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 obfuscate 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 time-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

custom application metrics supported

stackdriver metrics 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 customers can edit the users from the Admin console.

Non GSuites customers 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 the Budget doesn't cap API usage.

CLI:

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

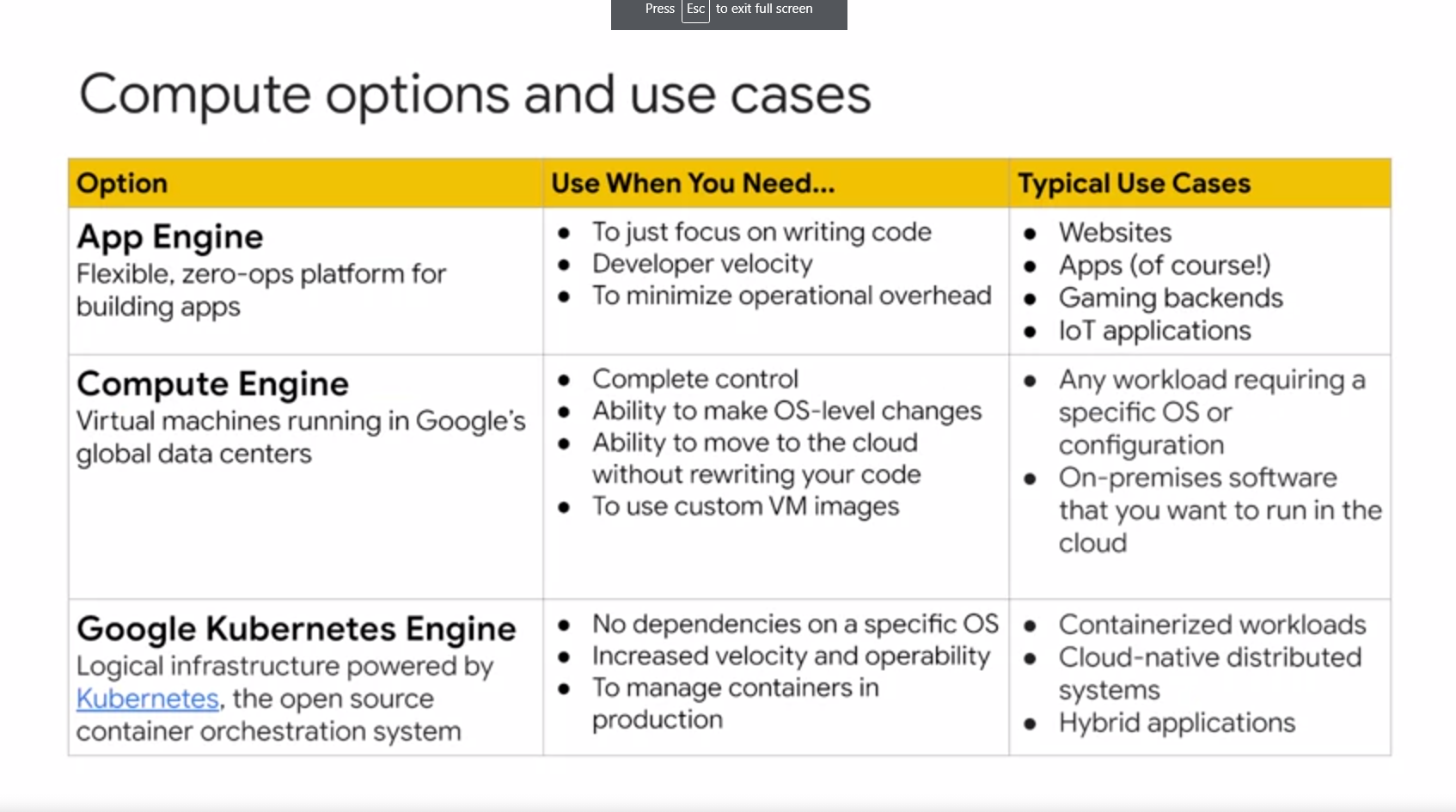
SDK includes - gcloud, gsutil (cloud storage), bq (bigQuery)

### WEEK-4

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

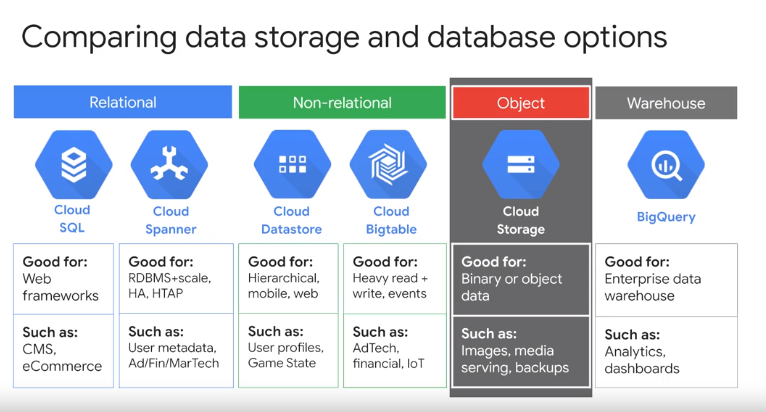
#### Pricing Calculator

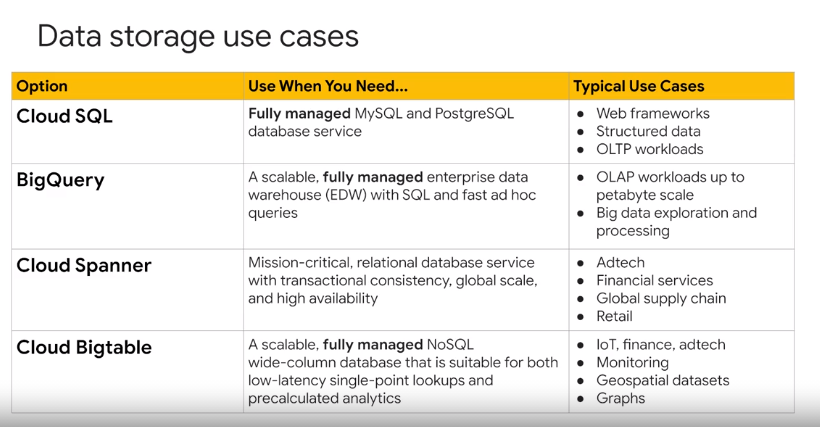
#### Planning the Compute Engine

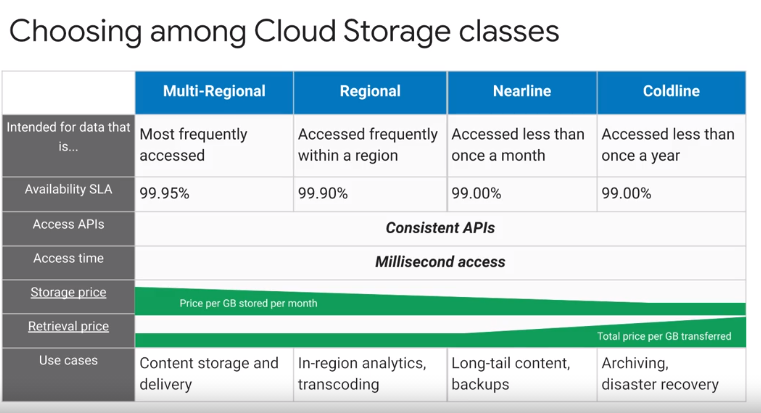


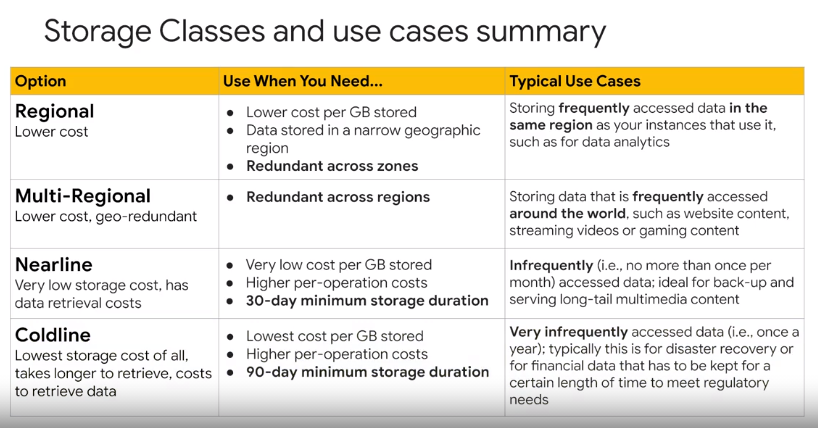
* Mobile or HTML application that does the heavy lifting on the client, need to store complex noSQL documents, need client APIs for mobiles then use “**Firebase**”
* Serverless (AppEngine is also serverless), or building a system based more on events than user interactions then use “Cloud Functions” - Javascript functions that run on Node.js, gets exposed as Micro-services, can call 100s of APIs

#### Planning the Data Storage









##### General Points

* Cloud SQL is regional while CloudSpanner is Global and horizontally scalable.
* Cloud DataStore also known as FireStore.
* Command for BigQuery is bq

##### GCS

* All storage options cost the fees per GB per month.
* Multi-regional is the costliest and Coldline is the cheapest option. (Archive is even cheaper)
* Nearline incurs the access fees per GB and Coldline incurs the higher access fees per GB
* To share an object publicly, give the “allUsers” identity a Reader permission.
* Commands:
  + List all the content > **gsutil ls gs://bucket-name/\*\***
  + **-a** flag shows the archived files too (after enabling versioning)
  + Create a bucket **> gsutil mb -l <region> <bucket-name>**
  + Get the bucket labels **> gsutil label get gs://<bucket-name>/**
  + Add a new bucket label **> gsutil label ch -l “newLabel:newValue” gs://<bucket-name>**
  + Versioning can only be done through the command line.
  + **gsutil versioning gs://bucket-name**
  + Set versioning **> gsutil versioning set on bs://bucket-name**
  + To copy the local files to the bucket **> gsutil cp file-name gs://bucket-name**
  + Use **\*\*** after bucket name as regEx to copy all the bucket content from one bucket to another, now one caveat is that the folder hierarchy will not be maintained here unlike **-r flag.**
  + Now, the permissions in the new bucket will not be overwritten by the permission of the old bucket. To make an object publicly available through command,  
    **> gsutil acl ch -u AllUsers:R gs://bucket-name/file-name**

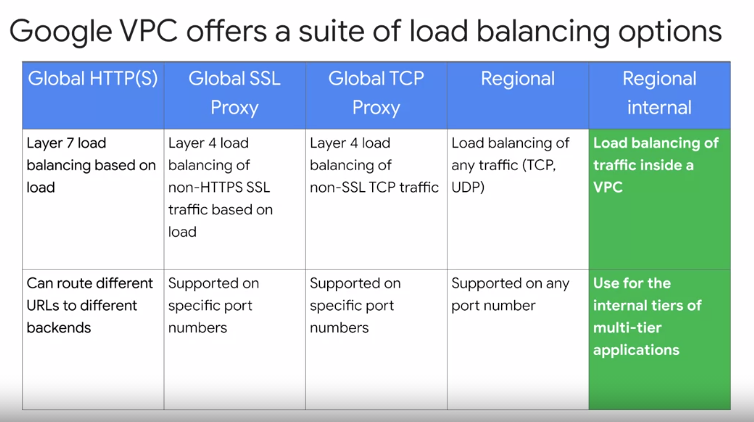
##### Cloud BigTable

* Low latency, high availability, wide column, not serverless.
* In BigTable, Node stores the metadata about the data.
* **BigTable storage system is known as Colossus**, it is a successor to the Google File System.

##### Cloud MemoryStore

* Caching Service, Low latency, high availability
* Redis
  + Redis can be connected from App Engine, GCE, Cloud Function, Kubernetes Clusters
  + Scales as needed, upto 30 GB storage and 12 Gbps network bandwidth
  + Basic Tier and Standard Tier
* Memcached
  + Distributed in-memory key-value store
  + Need to provide number of nodes, memory and CPU
  + Max 20 nodes with same configuration
  + 1 to 32 CPUs, 1 to 256 GB memory per node, and max 5 TB in an instance.

#### 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)

**Regional is the Network load balancer and doesn’t support SSL or TCP Proxies.**

#### Lab: Setting up Network and HTTP LoadBalancers

* A startup script to be used by every virtual machine instance to setup Nginx server upon startup
* An instance template to use the startup script
* A target pool - allows a single access point to all the instances in a group and is necessary for load balancing
* A managed instance group using the instance template (similar to ASG and ASG template)

Network Load Balancer - regional, allows to balance the traffic on protocols such as HTTP, TCP, UDP (SMTP Traffic), allows to inspect network packets.

Create “forwarding-rules” in order to create Network LB.

HTTP(s) Load Balancer -

Components:

* Http-health-check health check service
* Set named-ports of MIG, traffic will be forwarded to these ports
* Backend-service: takes the protocol and this will make use of the http-health-check.
  + Apply this to MIG (Add the instance group into the backend service)
* Create Url-maps
  + Attach this to Backend-service
* Create a target HTTP-LB-Proxy to route requests to your URL map
* Create Global forwarding Rule HTTP-Content-Rule to handle and route incoming requests.

It sends traffic to HTTP or HTTPS proxy depending on the IP, protocol and port specified. (Similar to Listener Rule). It doesn’t support multiple ports.

Forwarding-rules for Network Load balancer and Global forwarding rule for HTTP(s) LB both can be created with,

**Gcloud compute forwarding-rules create**… command. And all the forwarding-rules can be listed with below command.

**gcloud compute forwarding-rules list**

#### Networking VPC and VPN: Udemy Section-19/20/21

##### VPC

* VPC has by default all subnets created with distinct IP ranges and 4 firewall rules.
* **Shared VPC:** Share the resources between VPCs under the same Organization. However, projects don’t have to be under the Organization but in this case creating an organization is necessary.
* **VPC Network Peering:** Share the resources between VPCs of different organizations. (low level networking option). Traffic is routed through BGP.

##### VPN

* VPN lets you connect your Google Compute Engine resources to your own private network, through IPSec protocol (encrypted)
* 2 options
  + High Availability (HA): 99.99%, supports dynamic routing
  + Classic VPN: supports dynamic and static routing, no high availability
* Throughput upto 3 Gbps

##### Cloud Interconnect

* Lets you connect your onPrem with Google Cloud
* High level service, comes with SLA
* Minimum 10 Gbps upto 100 Gbps,
* Speed with partner, minimum 50 Mbps to 10 Gbps

##### Cloud DNS

* Zones are the containers of the DNS records (suffix part)
* When you create a DNS record, 2 records are created as below,
  + Type NS - Stores the information about NameServer
  + Type SOA - Authority Record
* Record A - to associate the IP with DNS

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

### WEEK-5

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

#### GCE

* **> gcloud compute instances list**
* **> gcloud compute instances create myvm (asks for the zone name)**
* **> gcloud compute instances delete myvm**
* Default machine type **n1-standard-1** and default zone **us-west1-a**
* To see the available machine types **> gcloud compute machine-types list**
* **F1-micro** is a free tier machine-type.But it is limited to the **US region and the zone should NOT be Virginia.**
* **> gcloud compute machine-types list --filter=”NAME:f1-micro”**
* **> gcloud compute machine-types list --filter=”NAME:f1-micro AND ZONE~us-west”**
* **> gcloud compute instances create --machine-type=f1-micro myvm**
* SSH to the box:
  + **> gcloud compute ssh <vm-name>** this command creates a ssh key and lets you login to the vm.
* To find the metadata about the vm, curl to the metadata.google.internal/computeMetadata/v1/ api is being used.
  + **> curl -H “Metadata-flavor:Google” metadata.google.internal/computeMetadata/v1/**
  + **Output:** instance/, oslogin/, project/
  + Ssh key information sent to this vm using below url:
  + metadata.google.internal/computeMetadata/v1**/project/attributes/ssh-keys**
  + Service account details are present under **instance/service-accounts/default/**
  + Above command return a few further entities like aliases, email, identity, scopes, token
  + So that means, to find out a service account being used/created by the vm this endpoint can be used > curl -H “Metadata-Flavor:Google” metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/email
* We can’t shut down the instance from within the instance after doing ssh because the default service-account gets Read only access to that instance by default.
* Deleting the Unmanaged instance group doesn’t delete all the instances as opposed to the managed MIG.
* MIG provides the feature of Autoscaling also.

#### App Engine

* Requirements: only specific versions of Java, Python, Go, PHP, Node.js, Ruby
* Runs apps only in preconfigured containers.
* Autoscales, requires minimal configuration.
* Sandbox constraints:
  + All requests time out at 60 sec
  + Limits on third party softwares
  + No writing to local files
* AppEngine Flexible
  + Allows custom containers and hence custom health checks
  + Apps run in regional managed instance groups, not zonal managed instance groups.
* Update the app-engine-python environment
  + **Gcloud components install app-engine-python**
* For deployment, provide a yaml file and test script: **gcloud app deploy app.yaml**
* To retrieve the endpoint: **gcloud app browse**
* 3 ways to split the traffic between multiple versions of an application,
  + IP addres, HTTP Cookie, Random selection
  + HTTP Cookie: GOOGAPPUID Cookie contains hash value which can be used to determine the version.
* App.yaml is the default configuration file name.

#### Cloud Functions

* Written in Javascript (Node.js), Python and Go
* Events (GCP specific) can call them asynchronously or you can use HTTP calls to call them synchronously.
* Code can be provided using an inline editor, from the bucket etc.
* Code from the GCS Bucket: format must be zip, default file name (if not given) Main.zip
* Default memory allocated to functions is 256 MB.
* CloudShell command to create a Cloud Function which will be triggered on GCS Bucket event, source code is also stored inside a bucket:

**gcloud functions deploy hello\_gcs --runtime=python37 \ --trigger-resource=<bucket\_name> \**

**--trigger-event=google.storage.object.finalize \**

**--source=gs://<zip\_location\_in\_a\_bucket>**

#### Data Solutions

* Cloud DataStore: Terabytes+, 1 MB/entity
* Cloud BigTable: Petabytes, approx. 10MB/Cell or 100MB/row
* Cloud SQL: Terabytes, relational SQL for OLTP, Regional
* Cloud Spanner: Petabytes. 10,240 MB/row, relational SQL for OLTP, Global, Scalable
* Cloud Storage: Max 5 TB/Object
* BigQuery: 10 MB/row, supports relational SQL for OLAP

#### Cloud Launcher (GCP MarketPlace)

* Deploy functional software without extra configuration.
* Only pay for underlying GCP resources.

#### Deployment Manager

* Infrastructure management service, automates the creation and management of Google Cloud Resources.
* Resources, Type/Type-provider, base type, composite type
* **[API].[version].[Resource]**
* Templates are called Deployment Manager Templates.

#### Lab: Deployment Manager and Cloud Monitoring Dashboard and Alert

* <https://github.com/GoogleCloudPlatform/deploymentmanager-samples>

### WEEK-6: Managing Resources (Ensuring successful operation)

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

#### Managing Compute Engine Resources

* Public images are free to use except some premium images.
* Public images are updated monthly by Google.
* Boot disk gets created of the same size as the image. Resizing is possible.
* Custom images are created from boot-disks etc., need to pay storage charges for them.
* Image Family: group similar images together, always pointing to the most recent image.
* A Custom Image can be created from a PD, a snapshot of a PD, another image, another image from another project, a compressed RAW image.
* Snapshots are stored in multiple locations.
* Can be shared across projects.
* **Commands:** **gcloud compute snapshots list/describe**.
* Snapshots - for data, as opposed to Images which are primarily used to create Boot disks.
* Snapshots are stored in the multi-region that is geographically closest to the location of your PD.
* Snapshots are incremental.
* Accessing the Snapshot from a different region will incur the Network Cost.
* **Preemptible Instance:** saves 80% cost, GCP can shutdown the instance at any time so your workload must be able to recover from where it left last time. Instances last 24 hours or less.

#### Managing Kubernetes Resources

* in GKE, load balancers are Network LBs
* By default Clusters are created with Compute Engine type n1-standard-1 and size 3, disk size per Node 100 GB.
* To create the Cluster programmatically, the **command** being used is “gcloud container clusters create <cluster-name>” as opposed to kubectl command.
* Using GKE over plain Kubernetes provides some additional GCP specific Node level functionalities like LoadBalancing, Scaling, Auto healing, Auto upgrades, Node Pool instance marking etc.

#### Managing AppEngine

* Can be Resident (manual scaling) or Dynamic (automatic scaling) instance.
* Basic scaling uses Dynamic instances. It creates instances when the application receives requests. It shuts down all instances when the application is idle. This is ideal for intermittent work or work is driven by user activity.
* Manual, basic, and automatically scaling instances startup differently.
* When you start a manual scaling instance, App Engine immediately sends a /\_ah/start request to each instance.
* When you start an instance of a basic scaling service, App Engine allows it to accept traffic, but the /\_ah/start request is not sent to an instance until it receives its first user request. Multiple basic scaling instances are only started as necessary, in order to handle increased traffic.
* Automatically scaling instances do not receive any /\_ah/start request.
* When an instance responds to the /\_ah/start request with an HTTP status code of 200–299 or 404, it is considered to have successfully started and can handle additional requests.

#### Managing Data Solution

* GCS: Conditions and action.
* If multiple rules have their conditions satisfied simultaneously, Cloud Storage performs an action based on following rules:
  + The Delete action takes precedence over any SetStorageClass action.
  + The SetStorageClass action that switches the object to the storage class with the lowest at-rest storage pricing takes precedence.

#### Managing Network Resources

* Each subnet can have max 5 secondary IP ranges (for alias IP).
* Longest subnet mask /29 (8 IP addresses). Broadest is /16.
* Google reserves 4 IP addresses.
* IP range can be replaced (if no instance uses it), extended but cannot be shrunk.

#### Managing Alerts

* Conditions, Notifications, Documentation.
* Authorization: IAM role must be one of - Monitoring Admin, Editor or Project Owner.

### Week-7: Managing IAM

#### Managing IAM

* In Cloud IAM, you grant access to member, member can be of the following types:
  + Google Account, Service Account, Google Group, G Suite Domain, Cloud Identity Domain.
* Predefined roles contain permissions.

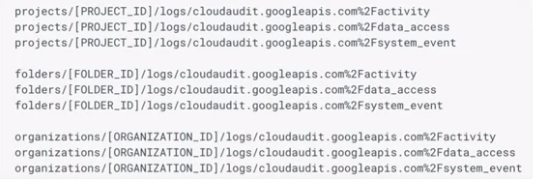
#### Custom Roles

* Must possess iam.roles.create permission on our userAccount or Role. By default the owner of a Project or Organization has this permission.
* Users who are not owners, including organization admins must be assigned either the organization role administrator role, or the IAM role administrator role.
* Predefined role - Hexagone icon, custom role - Factory icon.
* Can be created from existing Roles as well.

#### Service Accounts

* **Access Scopes**: no longer required for setting VM permissions, but still required for configuring instances to act as Service Accounts.
* Scope takes the form of a URL - **googleapis.com/auth/**bigquery.insertdata
* VMs can only perform operations that are allowed by the Scope and roles assigned to the Service account. (in case of conflict, least access will take precedence)
* VM needs to be restarted if you need to change the scope.
* Permission to impersonate the service account is provided by any role that includes the **iam.serviceAccounts.actAs** permission.
* To directly impersonate a service account, the role **iam.serviceAccountTokenCreator** is required which contains 4 permissions. **(iam.serviceAccounts.getAccessToken)**
* You can create upto **100 User-managed service accounts** and **10 user-managed keys per account**.
* You can disable the use of the user-managed key by applying policy constraint at Project, Folder or Organization.
* Google Cloud services create the user-managed service accounts which are called **Default service accounts**. They are automatically granted the Editor Role on your project.
* Some Google Cloud services need access to your resources so that they can act on your behalf. To meet this need, Google creates and manages service accounts for many Google Cloud services. These service accounts are known as **Google-managed service accounts**.
* You can grant the Service Account User role (**roles/iam.serviceAccountUser**) to a user for a project and that will give the user access to all service accounts in the project, including service accounts that may be created in the future.

#### Viewing Audit Logs

* Logs are written for Projects, Folders and Organization.
* Four types of Audit logs are kept for the projects and services:
  + Admin Activity, System Events, Policy Denied, Data Access
* IAM role **Logging/Logs Viewer** or **Project/Viewer** is required for viewing the first 3 types of logs, they are written by default.
* IAM role **Logging/Private Logs Viewer** or **Project/Owner** is required for viewing the first 4th type of logs, they are NOT written by default because they can be quite large.
* Can be seen inside StackDriver → Logging → Logs
* Every audit log entry in Cloud Logging is an object of type **LogEntry**. What distinguishes an audit log entry from other log entries is the **protoPayload field**; this field contains an **AuditLog object** that stores the audit logging data.
* Format: cloudaudit.googleapids.com%2fsystem\_event (in case of a particular service, the subdomain will be replaced with the service name)
  + 

#### Udemy: Section-6: IAM 23, 24, 27, 28

##### Primitive Roles

Those are the roles before the existence of IAM. Not recommended because of broad level access.

##### Policies

Collection of stmts which defines which user has access to what resources.

*Roles assigned to Identities while Policies assigned to Resources.*

*Policies take effect in 60 seconds to 7 minutes.*

#### Lab: Site Reliability Troubleshooting with Cloud Monitoring APM [ACE]

Github Repositories:

https://github.com/GoogleCloudPlatform/training-data-analyst/tree/master/blogs/microservices-demo-1

<https://github.com/GoogleCloudPlatform/microservices-demo>

## Udemy/ACloudGuru/Google-Docs

### Cloud Run and Anthos

* Cloud Run: Compute Service to run the stateless containers
* Either Managed Service or Run on Anthos
* Managed Service
  + Pay per use
  + 1000 services per region, 2 GB memory per container instance, 2 vCPUs per container instance
  + Container isolated using gVisor
  + “Service” is the main abstraction of computing (there might be multiple containers but they represent the same Service).
  + Revision is the deployment of a Service. It consists of a specific image and configuration.
  + Container instances run Revisions.
  + Container instances can receive up to 80 requests at the same time, and are possible to reduce to 1.
* Anthos
  + Application management platform based on Kubernetes.
  + Anthos workloads can be moved across various infrastructure (AWS, GKE, OnPrem)

### Cloud Dataproc

Managed Hadoop and Spark service. For lift and shift the on-prem Hadoop cluster, and you want to use a managed service.

### Cloud Dataflow

Stream processing and batch processing platform. For doing ETL (extraction transformation load) work.

If you have a lot of data in Cloud Storage and you want to do transformation to it and load them to BigQuery.

### BigData and IoT

* Cloud TensorFlow
* Cloud **Vision** API (OCR) - pay per image
* Cloud **Speech** API - pay per 15s of audio
* Cloud **Natural Language API** - analyzes text, charged per request of 1000 characters
* Cloud **Translation** API - charged per character processed
* **DialogFlow** - chatbot, charged per request
* Cloud **Video Intelligence API**
* Cloud **Job Discovery**
* **Data Catalog** - fully managed and scalable **metadata management service** that empowers organizations to discover quickly, understand, and manage all their data. It offers a simple and easy-to-use search interface for data discovery, a flexible and powerful cataloguing system for capturing both technical and business metadata, and a strong security and compliance foundation with Cloud Data Loss Prevention (DLP) and Cloud Identity and Access Management (IAM) integrations.
  + The service **automatically ingests technical metadata for BigQuery and Cloud Pub/Sub.**

## QWIKLABS QUESTS

1. Monitoring and Logging (Stackdriver Fundamentals Quest)   
   <https://www.qwiklabs.com/quests/35>
2. Anthos: Service Mesh  
   <https://www.qwiklabs.com/quests/100?catalog_rank=%7B%22rank%22%3A45%2C%22num_filters%22%3A2%2C%22has_search%22%3Afalse%7D>
3. Serverless Firebase  
   <https://www.qwiklabs.com/quests/153?catalog_rank=%7B%22rank%22%3A42%2C%22num_filters%22%3A2%2C%22has_search%22%3Afalse%7D>

## Udemy - things to go through

Section-8: Advanced Virtual Machine Configuration

Section-12: Intro to AppEngine

Section-13: Cloud Functions

Section-14: Cloud Run and Anthos

Section-15: Review of Compute Options

Section-16: Storage

77

Section-18: Overview of Big Data Services

Section-19: Networking

Section-20: Networking

Section-21: Networking

Section-23: Monitoring

## ACloudGuru - things to go through

Chapter-7: Basic Services

Chapter-8: Basic Services

Chapter-9: Scaling

Chapter-10: Security (if required)

Chapter-11: Networking (if required)

Load Balancing

Configuring Firewall Rules

Chapter-14: Services Breadth

From Data Transfer…

## Important Links

* <https://cloud.google.com/sdk/docs/cheatsheet>
* <https://cloud.google.com/sdk/gcloud/reference>
* <https://cloud.google.com/docs/enterprise/best-practices-for-enterprise-organizations>
* <https://cloud.google.com/solutions/best-practices-for-operating-containers>
* <https://cloud.google.com/iam/docs/understanding-roles>
* https://cloud.google.com/products/calculator/
* <https://cloud.google.com/bigquery/docs/best-practices-costs#materialize_query_results_in_stages>
* <https://cloud.google.com/iam/docs/understanding-roles#cloud-spanner-roles>
* <https://cloud.google.com/compute/docs/access/iam#compute.storageAdmin>
* <https://cloud.google.com/vpc/docs/shared-vpc>
* <https://cloud.google.com/sdk/gcloud/reference/beta/compute/instance-groups/managed/rolling-action>
* <https://cloud.google.com/sdk/gcloud/reference/app/versions/list>
* <https://cloud.google.com/sdk/gcloud/reference/compute/instance-templates/create>
* <https://cloud.google.com/sdk/gcloud/reference/app/deploy>
* <https://cloud.google.com/billing/docs/concepts>
* <https://cloud.google.com/sdk/gcloud/reference/auth/activate-service-account>
* <https://cloud.google.com/sdk/gcloud/reference/auth>
* <https://github.com/ddneves/awesome-gcp-certifications>
* <https://sathishvj.medium.com/writing-and-passing-the-google-cloud-associate-engineer-certification-a60c2f6d99c2>

## Udemy - practice tests

* You can make use of the Cloud Dataflow connector for Cloud Spanner (https://cloud.google.com/spanner/docs/dataflow-connector) and Dataflow Connector for Cloud Bigtable (https://cloud.google.com/bigtable/docs/hbase-dataflow-java) to retrieve data from these sources, but **you can’t use Dataflow SQL to restrict this to specific users**. Dataflow SQL natively only works when reading data from Pub/Sub topics, Cloud Storage file sets, and BigQuery tables.
* Set up a Cloud Dataflow job to read data from Cloud Storage and Cloud BigTable for the specific gamer. - not as efficient as using BigQuery. Also, **BigQuery is faster. Dataflow expects the CSV files in the CloudStorage.**
* Also, it is possible to set up a Cloud Dataproc Cluster to run a Hadoop job to join up data from Cloud Storage and Cloud BigTable for the specific gamer - not right but possible.
* **The Datastore emulator** provides local emulation of the production Datastore environment. You can use the emulator to develop and test your application locally. Can be installed using **gcloud components install**.
* **App Engine is regional**, which means the infrastructure that runs your apps is located in a specific region and is managed by Google to be redundantly available across all the zones within that region. **You cannot change an app's region after you set it.   
  Therefore, the only way to have an app run in another region is by creating a new project and targeting the app engine to run in the required region**
* **Activity logs** - can be restricted down to users and further filtered by Data Access as activity but can’t be filtered at granular level like bucket names, operation type etc.
* **Stackdriver Trace** runs on Linux in the following environments: Compute Engine, Google Kubernetes Engine (GKE), App Engine flexible environment, App Engine standard environment. **Not on GCS.**
* **Redis**
  + Redis can be connected from App Engine, GCE, Cloud Function, Kubernetes Clusters.
  + **Cloud Memorystore** is a fully managed in-memory data store service for Redis built on scalable, secure, and highly available infrastructure managed by Google.
  + You can also specify **eviction (maxmemory) policies** to restrict the rest of processes to 2GB or the reverse proxy to 30GB or both; you can select a suitable maxmemory policy to handle scenarios when memory is full.
* **Data Access audit logs** do not record the data-access operations on resources that are publicly shared (available to All Users or All Authenticated Users), or that can be accessed without logging into Google Cloud. In GCS, they are not enabled by default.
* **Bigtable** provides out of the box support for time series data.
* **BigQuery**
  + BigQuery offers **flat-rate pricing** for customers who prefer a **stable monthly cost** for queries rather than paying the on-demand price per TB of data processed. You enrol in flat-rate pricing by purchasing slot commitments, measured in BigQuery slots. Slot commitments **start at 500 slots, and the price starts from $10000**.
  + If your capacity demands exceed your committed capacity, BigQuery will queue up slots, and you will not be charged additional fees.
  + On-demand price **per TB of data processed**.
  + Once your data is loaded into BigQuery, you are charged for storing it.
  + BigQuery **supports querying Cloud Storage data in several formats such as CSV, JSON, AVRO**, etc. You do this by creating a Big Query external table that points to a Cloud Storage data source (bucket). This solution works out of the box, involves minimal effort, minimal cost, and is quick.
  + You can copy the data from local but they have to be in a structured format.
  + Projects using the **different billing accounts** can still export their billing data to **the same BigQuery Dataset.**
  + **roles/bigquery.jobUser role** lets your (App engine service) **create and run jobs including "query jobs"** but doesn't give access to read data, i.e. query the data directly from the datasets.
  + **roles/bigquery.dataViewer role** lets you **query and view the data from the datasets**.
* **AppEngine**
  + **> gcloud app deploy app.yaml**
  + **> gcloud app browse**
  + Lists all the services deployed **> gcloud app versions list**
  + Lists all the service minus the ones which don’t serve any traffic **> gcloud app versions list --hide-no-traffic**
  + To **migrate** to new version of app   
    > **gcloud app versions migrate v3 --service="pt-createOrder"**
  + You can **roll back to a previous version in the app engine GCP console**. Go back to the list of versions and check the box next to the version that you want to receive all traffic and click the MAKE DEFAULT button located above the list. Traffic immediately switches over to the selected version.
  + ENV var for using AppEngine Flexible > **env: flex**
  + **--no-promote** flag is used when deploying an app but don’t want it to serve the traffic.
  + To deploy an app to a non-default project, pass **--project flag**, however, if won’t create a new project if not exisitng.
  + **AppEngine standard is serverless, but AppEngine Flex is not as it runs on Compute Engine VMs. They are already within the VPC and can use Cloud VPN.**
  + **Logging** - In App Engine Standard, you should log using the **App Engine SDK** and the connection to Stackdriver (i.e. agent installation and configuration) is handled automatically for you. AppEngine sends **requests and applications both logs** to Stackdriver natively.
  + **In Pricing Calculator** - you can enter the number of instances, Outgoing Network Traffic, amount of GCS storage. But you **can’t enter the Operating System**.
* **GCS**
  + GCS - **Object composition** can be used to upload the object chunks in parallel.
  + gsutil mb -l us-west2 -s coldline --retention 10y gs://logs-archive
  + We can have **gsutil use an encryption key by using the -o** top-level flag: -o "GSUtil:encryption\_key=[KEY\_RESOURCE]".
  + As an alternative to the -o top-level flag, gsutil can also use an encryption key if **.boto** configuration is modified to specify the encryption key.
  + Gsutil command accepts the **Httpmethod using -m flag** - gsutil signurl -m PUT -d 4h <JSON Key File> gs://<bucket>/\*\*
  + Coldline can also be used for at most quarterly retrieval.
  + **Object Lifecycle Management does not rewrite** an object when changing its storage class. When an object is transitioned to Nearline Storage, Coldline Storage, or Archive Storage using the SetStorageClass feature, any subsequent early deletion and associated charges are based on the original creation time of the object, regardless of when the storage class changed.
  + If however, the change of storage class is **done manually using a rewrite,** the creation time of the objects is the new creation time since they are rewritten. In such a case, you would need to apply a lifecycle delete action accordingly.
  + **Static Website:**Create a bucket whose name matches the CNAME you created for your domain. For example, if you added a CNAME record pointing www.example.com to c.storage.googleapis.com., then create a bucket with the name "[www.example.com](http://www.example.com)
  + You can **control the storage type for the non-current version** also.
  + **To change the region, a new bucket must be created** as the region can not be modified after a bucket is created.
  + **Data cannot be transitioned from Multi-Regional to Regional** through Lifecycle Management; that would change the location
* **Deployment Manager** changes can be previewed using --preview flag.
  + **Command** example   
    > **gcloud deployment-manager deployments create my-gcp-ace-cluster --config cluster.yaml**
  + Also supports **update** action.
  + **Resource command,  
    > gcloud deployment-manager resources <describe/list>**
  + Create an **instance template,**

**>** gcloud compute **instance-templates** create templateName **--metadata-from-file** **startup-script=/script/install\_app.sh**

* + Similarly, to use the file from the web, a **startup-script-url** param is being used.
* To **patch the application running on MIG** without downtime,
  + > gcloud compute **instance-groups** <managed group name> **rolling-action start-update** --max-surge 1 --max-unavailable 0
  + The supported  **rolling-action** are replace, restart, start-update and stop-proactive-update.
  + **If MIG keeps failing to start a VM**,   
    we need to ensure that the instance **template is valid**, **disks.autoDelete is turned on**, and that there are **no existing persistent disks with the same name as VM instance**.
  + MIG comes with an **auto healing feature** based on Health check, this health check is different to Load balancer health check.
* **GCE**
  + **--boot-disk-auto-delete is enabled by default**. It enables automatic deletion of boot disks when the instances are deleted. Use **--no-boot-disk-auto-delete** to disable.
  + While creating an instance through the console, the scopes of the SA can not be modified when using a non-default service account. Only possible when using default SA.
  + To list all the VPCs  
    **> gcloud compute networks list**
  + To list all the instances with internal and external IPs  
    **> gcloud compute instances list**
  + An **ephemeral IP** **address is the public IP** Address assigned to a compute instance. An ephemeral external IP address is an IP address that doesn't persist beyond the life of the resource. When you create an instance or forwarding rule without specifying an IP address, the resource is automatically assigned an ephemeral external IP address.
  + **You can promote this ephemeral address into a Static IP address**, but this will be an external IP address and not an internal one.
  + > **gcloud compute addresses create** to reserve static internal or external IP addresses.
  + **> gcloud compute regions list**
  + **> gcloud compute images list.**
  + **OS Login** lets you use Compute Engine IAM roles to manage SSH access to Linux instances efficiently and is an alternative to manually managing instance access by adding and removing SSH keys in the metadata. Before you can manage instance access using IAM roles, you must enable the OS Login feature by setting a metadata key-value pair in your project or your instance's metadata: **enable-oslogin=TRUE**. After you enable OS Login on one or more instances in your project, those instances accept connections only from user accounts that have the necessary IAM roles in your project or organization. There are two predefined roles.
    - **roles/compute.osLogin, which does not grant administrator permissions**
    - **roles/compute.osAdminLogin, which grants administrator permissions**
  + You can **generate Windows passwords** using either the Google Cloud Console or the gcloud command-line tool. Alternatively, you can generate passwords programmatically with API commands, but all these methods assume that you have an existing windows instance.  
    **>** **gcloud compute reset-windows-password <windows-instance>**
  + **Set the default zone and region for in the project meta-data,**
  + **> gcloud compute project-info add-metadata \**  
    --metadata google-compute-default-region=europe-west1,google-compute-default-zone=europe-west1-b
* **Cloud Run for Anthos** is the only serverless GCP offering that supports WebSockets.
  + When using the fully managed environment, **Cloud Run on GKE is serverless**.
* **GKE Load Balancer** can be set as internal, making the service limited to VPC for the resources living in the same region.
  + Another alternative could be Cloud Armor to set up a whitelist policy of incoming traffic from Compute Engine but expensive.
  + externalTrafficPolicy - helps to manage the external traffic but pods will stay exposed to the internet.
* **Cloud Transfer** Service is used for transferring massive amounts (in the range of petabytes of data) of data to the cloud.
* **IAM**
  + **gcloud iam roles copy command** to copy the IAM roles from the Development GCP project to the Staging GCP project.
  + To create a new role that will apply to **all current and future projects created in your organization**, create that role for your organization (don’t create it within the project as you won’t be able to promote it to the organization)
  + **Compute Engine Admin Role** grants full control of all Compute Engine resources; including networks, load balancing, service accounts etc.
  + **Compute Engine Instance Admin** Role grants full control of Compute Engine instances, instance groups, disks, snapshots, and images.
  + To allow gcloud (and other tools in Cloud SDK) to use service account credentials to make requests, create JSON keys for the service account and execute   
    **gcloud auth activate-service-account --key-file [KEY\_FILE]**
  + **roles/iam.securityReviewer** provides permissions to **list all resources and Cloud IAM policies**. Note that list permissions only allow you to list but **not view** resources.
  + **roles/viewer - primitive role -** provides permissions for read-only actions that do not affect the state.
  + **roles/browser** provides read access to browse the hierarchy for a project, including the folder, organization, and Cloud IAM policy.
  + **roles/iam.roleAdmin** provides access to all **custom roles** in the project.
  + **roles/iam.roleViewer** provides read access to all **custom roles** in the project.
  + **roles/accessapproval.approver** is an Access Approval Approver role and provides the ability to view or act on access approval requests and view configuration.
  + **Billing**
    - To **change the billing of a project**, you must be an **owner of the project and Billing Admin** on the billing account
    - **roles/billing.Viewer**
    - **roles/billing.Admin**
    - **roles/billing.User**  
      This role has very restricted permissions, so you can grant it broadly, typically in combination with Project Creator. These two roles allow a user to create new projects linked to the billing account on which the role is granted.
    - **roles/billing.ProjectManager**This role allows a user to **attach the project to the billing account**, but does not grant any rights over resources. Project Owners can use this role to allow someone else to manage the billing for the project without granting them resource access.
    - **Billing Account Administrator role** enables a user to view the spend and **set budget alerts.**
  + **Custom Roles Lifecycle Stages**
    - When setting support levels for permissions in custom roles, you can set to one of **SUPPORTED, TESTING or NOT\_SUPPORTED**.
    - **SUPPORTED** -The permission is fully supported in custom roles.
    - **TESTING** - The permission is being tested to check its compatibility with custom roles. You can include the permission in custom roles, but you might see unexpected behaviour. Such permissions are not recommended for production use.
    - Since we want the role to be **suitable for production use, we need "SUPPORTED"** and not "TESTING".
    - In terms of role stage, the stage transitions from **ALPHA --> BETA --> GA**
  + To get the **list of all the project owners,**   
    > **gcloud projects get-iam-policy <project\_id> --filter=”binding.role:roles/owner”**
  + **Organization Admin** role provides permissions to get and list projects but **not shutdown** projects.
  + The **primitive Project Owner** role provides permissions to delete projects.
    - All billing and traffic serving stops
    - You lose access to the project
    - The owners of the project will be notified and can stop the deletion within 30 days, The project will be scheduled to be deleted after 30 days. However, some resources may be deleted much earlier.
  + **Default GCE Service Account:** [**PROJECT\_NUMBER-compute@developer.gserviceaccount.com**](mailto:PROJECT_NUMBER-compute@developer.gserviceaccount.com)
  + **Non-default Service Account:** **SERVICE\_ACCOUNT\_NAME@PROJECT\_ID.iam.gserviceaccount.com**
  + **Default App Engine service account:**[**PROJECT\_ID@appspot.gserviceaccount.com**](mailto:PROJECT_ID@appspot.gserviceaccount.com)
  + To list the name of the active account, **> gcloud auth list**
* **Login Mechanisms**
  + **you can't set the credentials in SAML** integration to always use a particular account. The authentication is carried out by IdP such as GSuite or a third-party identity provider.
  + **Cloud Identity catalogue**If you have a third-party IdP, you can still configure SSO for third-party apps in the Cloud Identity catalogue. User authentication occurs in the third-party IdP, and Cloud Identity manages the cloud apps.  
    To use Cloud Identity for SSO, **your users need Cloud Identity accounts**. They sign in through your third-party IdP or using a password on their Cloud Identity accounts.
  + As a G Suite or Cloud Identity administrator, the **password vaulted apps service** enables you to manage access to some of the apps that **don't support federation** and that are available to users on the User Dashboard. The password vaulted apps service saves login credential sets for applications and assigns those credential sets to users through group association.
  + **Secure LDAP** enables authentication, authorization, and user/group lookups for LDAP-based apps and IT infrastructure.   
    Secure LDAP **uses the same user directory for both SaaS and LDAP-based applications**, so people can use the same Cloud Identity credentials they use to log in to services like G Suite and other SaaS apps as they do to log into traditional applications. Applications and IT infrastructure that use LDAP can be configured to leverage Cloud Identity’s secure LDAP service instead of an existing legacy identity system—end-users don't have to change how they access their apps.
  + For GCP, **you need identities either in G Suite or Google Cloud Identity**. **Google Cloud Directory Sync (GCDS)** is a tool that enables you to synchronize users, groups, and other data **from an Active Directory/LDAP service to their Google Cloud domain directory.**
  + Use **Cloud Identity Aware Proxy (IAP)** to enable **SSH tunnels to the VMs** and add the third-party team as a tunnel user. IAP lets you,
    - Control access to your cloud-based and on-premises applications and VMs running on Google Cloud
    - Verify user identity and use context to determine if a user should be granted access
    - **Work from untrusted networks without the use of a VPN**
    - Implement a zero-trust access model
* **GKE**
  + **> gcloud container clusters update** - to specify the node locations.
  + **>** gcloud container clusters **update [CLUSTER\_NAME] --enable-autoscaling** --min-nodes=1 --max-nodes=10
  + **> gcloud container clusters resize** **my-gcp-ace-proj-1 --node-pool my-gcp-ace-primary-node-pool --num-nodes 20** - to specify the node locations.
  + To scale a Replica Set named 'target-replicaset', scale command is used,  
    **> kubectl scale --replicas=5**
  + **> kubectl label pods target-pod environment=production**
  + In GKE, **DaemonSets** manage groups of replicated Pods and adhere to a one-Pod-per-node model, either across the entire cluster or a subset of nodes. (HPA not applicable here)
  + Node **“auto-upgrade”** - enabled by default - helps you keep the nodes in your cluster up to date with the cluster master version when your master is updated on your behalf.
  + Node **“auto-repair”** - helps you keep the nodes in your cluster in a healthy, running state. When enabled, GKE makes periodic checks on the health state of each node in your cluster.
  + With **kubectl apply**, you can apply the configuration from a single file or multiple files comma separated or even a complete directory.
  + When you create an **Ingress object**, the GKE Ingress controller creates a Google Cloud HTTP(S) Load Balancer and configures it according to the information in the Ingress and its associated Services.
  + To deploy an **external load balancer in Ingress**, you use the **gce ingress class**. (For an internal load balancer, you would use **gce-internal ingress** class.)
  + **One zonal cluster per billing account is free**.
  + **GKE cluster management fees do not apply to Anthos GKE clusters.**
  + **Node auto-provisioning** automatically manages a set of **node-pools** on the user's behalf.
  + When you **enable GKE Sandbox** on a node pool, a **sandbox is created for each Pod** running on a node in that node pool. Also, nodes running sandboxed Pods are prevented from accessing other Google Cloud services or cluster metadata. Each sandbox uses its user-space kernel.
  + You can configure **node selector on the pods** with **key: cloud.google.com/gke-accelerator** and **value: <Ex. nvidia-tesla-k80>** .
  + To **reduce the cost of a fault-tolerant job** running on Compute Engine - GKE’s autoscaler is very smart and always tries to **first scale the node pool with cheaper VMs. In this case, it scales up the preemptible node pool.** The GKE autoscaler then scales up the default node pool—but only if no preemptible VMs are available.
  + The **ReplicaSet automatically creates the pods** to maintain a desired state (
* **Binary Authorization** is a deploy-time security control that ensures only trusted container images are deployed on Google Kubernetes Engine (GKE).
* **Container Analysis** is a service that provides vulnerability scanning and metadata storage for software artefacts.
* **Build/push the images to GCR without using Docker commands**
  + > gcloud builds submit **--tag** gcr.io/<project>/<artifact>/<version>
  + > gcloud builds submit **--config**=[CONFIG\_FILE] **--gcs-log-dir**=[GCS\_LOG\_LOCATION] [SOURCE]  
    --config - to supply the config in an yaml file.  
    [SOURCE] is the location of the source to build. The location can be a directory on a local disk or a gzipped archive file (.tar.gz) in Google Cloud Storage.
* **VPC**
  + The default VPC comes with **port 3389 open to the public**.
  + You can configure the **Serverless VPC** by creating a Connector.
  + Biggest CIDR range is 24-bit block **10.0.0.0/8** (16777216 IP Addresses)
  + You can use **Cloud Router Custom Route Advertisement to announce the Restricted Google APIs IP addresses through Cloud Router to your on-premises network.** The Restricted Google APIs IP range is **199.36.153.4/30**. While this is technically a public IP range, Google does not announce it publicly. This IP range is only accessible to hosts that can reach your Google Cloud projects through internal IP ranges, such as through a Cloud VPN or Cloud Interconnect connection.
  + To access VPC control services, hosts must send their requests to the **restricted.googleapis.com** domain name instead of \*.googleapis.com. The restricted.googleapis.com domain resolves to a VIP (virtual IP address) range 199.36.153.4/30
  + We need to create a **static route whose destination is 199.36.153.4/30** and whose next hop is the **default Internet gateway**.
  + If you require access to other Google APIs and services that aren't supported by VPC Service Controls, you can use **199.36.153.8/30 (private.googleapis.com)**.
  + **Default network tags** - default-allow-icmp, default-allow-internal, default-allow-RDP and default-allow-ssh
* **Load Balancer**
  + Google HTTP(S) Load Balancing has native support for the **WebSocket protocol** when you use HTTP or HTTPS, not HTTP/2, as the protocol to the backend.
  + The load balancer also supports **Session affinity**.
* **Cloud SQL**
  + **To set up the failover instance, configure High Availability (HA)** for Cloud SQL and Create a Failover replica in the same region but in a different zone.
  + by enabling **Binary logging**, we can enable **point-in-time recovery.**
* The simplest way to launch a **Jenkins server** is from GCP Marketplace.
* **Stackdriver**
  + **Groups** provide a mechanism for alerting on the behaviour of a set of resources, rather than on individual resources.
  + You can **monitor** resources of different projects in a **single Stackdriver account by creating a Stackdriver workspace.** A Stackdriver workspace is a tool for monitoring resources contained in one or more Google Cloud projects or AWS accounts. Each Workspace can have (**linked**) between 1 and 100 monitored projects, including Google Cloud projects and AWS accounts.
  + **Log entries** are held in Stackdriver Logging for a limited time known as the retention period - which is **30 days (default configuration).** After that, the entries are deleted. To keep log entries longer, you need to export them outside of Stackdriver Logging by configuring log sinks.
  + **Supported sink destinations** for exported log entries are **Cloud Storage, BigQuery, and Pub/Sub**.
* **Domain restriction constraint** can be used in organization policies to limit resource sharing based on domain.