1 - Setting up a cloud solution environment

1.1 - Setting up cloud projects and accounts

1.1.1 - Creating projects

https://cloud.google.com/sdk/gcloud/reference/projects/create

1.1.2 - Assigning users to pre-defined IAM roles within a project

https://cloud.google.com/sdk/gcloud/reference/projects/add-iam-policy-binding

1.1.3 - Linking users to G Suite identities

https://support.google.com/cloudidentity/answer/7319251?hl=en

1.1.4 - Enabling APIs within projects

https://cloud.google.com/service-management/enable-disable
https://cloud.google.com/sdk/gcloud/reference/services/
https://cloud.google.com/sdk/gcloud/reference/services/list
https://cloud.google.com/sdk/gcloud/reference/services/enable
https://cloud.google.com/sdk/gcloud/reference/services/disable

1.1.5 - Provisioning one or more Stackdriver accounts

https://cloud.google.com/monitoring/accounts/

1.2 - Managing billing configuration

1.2.1 - Creating one or more billing accounts

https://cloud.google.com/billing/docs/how-to/manage-billing-account

1.2.2 - Linking projects to a billing account

https://cloud.google.com/sdk/gcloud/reference/beta/billing/projects/link

1.2.3 - Establishing billing budgets and alerts

https://cloud.google.com/billing/docs/how-to/budgets

1.2.4 - Setting up billing exports to estimate daily/monthly charges

https://cloud.google.com/billing/docs/how-to/export-data-file
https://cloud.google.com/billing/docs/how-to/export-data-bigquery
https://cloud.google.com/billing/docs/how-to/bq-examples

1.3 - Installing and configuring the command line interface (CLI), specifically the Cloud SDK (e.g., setting the default project).

Install:
https://hub.docker.com/r/google/cloud-sdk/~/dockerfile/
https://cloud.google.com/sdk/docs/quickstart-linux

To set the project property in the core section, run:

```
$ gcloud config set project myProject
```

To set the zone property in the compute section, run:

```
$ gcloud config set compute/zone asia-east1-b
```

2 - Planning and configuring a cloud solution

2.1 - Planning and estimating GCP product use using the Pricing Calculator

https://cloud.google.com/products/calculator/
Google Cloud Platform on a shoestring budget (Google I/O '18)

2.2 - Planning and configuring compute resources

2.2.1 - Selecting appropriate compute choices for a given workload

Deciding between Compute Engine, Container Engine, App Engine and more (Google Cloud Next '17)

2.2.1.1 - Compute Engine

And Compute Engine is, basically, kind of everything else, or even all of those things if you want. It's VM.
So you have full control to do whatever you need to do to connect things together.
It's also a really good fit for existing systems.

2.2.1.2 - Kubernetes Engine

Container Engine is a system of containers working together to solve your problems.

2.2.1.3 - App Engine

Get to know Google App Engine
App Engine is focused on making your web code run extremely well.
It's optimized for that. And it's code first kind of thinking.

2.2.2 - Using preemptible VMs and custom machine types as appropriate

```
1. // CREATE INSTANCE WITH 4 vCPUs and 5 GB MEMORY
2. gcloud compute instances create my-vm --custom-cpu 4 --custom-memory 5
3.
4. // ENABLE PREEMPTIBLE OPTION
5. gcloud compute instances create my-vm --zone us-central1-b --preemptible
```

2.3 - Planning and configuring data storage options

From blobs to relational tables: Where do I store my Data? (Google Cloud Next '17)

2.3.1 - Product choice

2.3.1.1 - Cloud SQL

https://cloud.google.com/sql/docs/
Cloud SQL is a fully-managed database service that makes it easy to set up, maintain, manage, and administer your relational databases on Google Cloud Platform.

2.3.1.2 - BigQuery

https://cloud.google.com/bigquery/
A fast, highly scalable, cost-effective and fully-managed enterprise data warehouse for analytics at any scale

BigQuery is Google's serverless, highly scalable, low cost enterprise data warehouse designed to **make all your data analysts productive**. Because there is no infrastructure to manage, **you can focus on analyzing data to find meaningful insights** using familiar SQL and you don't need a database administrator. BigQuery **enables you to analyze all your data** by creating a logical data warehouse over managed, columnar storage as well as data from object storage, and spreadsheets. BigQuery makes it easy to securely**share insights within your organization and beyond** as datasets, queries, spreadsheets and reports. BigQuery allows organizations

to**capture and analyze data in real-time** using its powerful streaming ingestion capability **so that your insights are always current**. BigQuery **is free** for up to 1TB of data analyzed each month and 10GB of data stored.

2.3.1.3 - Cloud Spanner

https://cloud.google.com/spanner/
The first horizontally scalable, strongly consistent, relational database service

Cloud Spanner: The best of the relational and non-relational worlds

	CLOUD SPANNER	TRADITIONAL RELATIONAL	TRADITIONAL NON-RELATIONAL
Schema	✓ **Yes**	✓ Yes	✗ No
SQL	✓ **Yes**	✓ Yes	✗ No
Consistency	✓ **Strong**	✓ Strong	✗ Eventual
Availability	✓ **High**	✗ Failover	✓ High
Scalability	✓ **Horizontal**	✗ Vertical	✓ Horizontal
Replication	✓ **Automatic**	↻ Configurable	↻ Configurable

2.3.1.4 - Cloud Bigtable

https://cloud.google.com/bigtable/

2.3.2 - Choosing storage options

https://cloud.google.com/storage/docs/gsutil/commands/mb
https://cloud.google.com/storage/docs/storage-classes

2.3.2.1 - Regional

Storing frequently accessed in the same region as your Google Cloud DataProc or Google Compute Engine instances that use it, such as for data analytics.

2.3.2.2 - Multi-regional

Storing data that is frequently accessed ("hot" objects) around the world, such as serving website content, streaming videos, or gaming and mobile applications.

2.3.2.3 - Nearline

Data you do not expect to access frequently (i.e., no more than once per month). Ideal for back-up and serving long-tail multimedia content.

2.3.2.4 - Coldline

Data you expect to access infrequently (i.e., no more than once per year). Typically this is for disaster recovery, or data that is archived and may or may not be needed at some future time.

StorageClass	Characteristics	Use Cases	Price (per GB per month)**
Multi-Regional Storage	• >99.99% typical monthly availability • 99.95% availability SLA* • Geo-redundant	Storing data that is frequently accessed ("hot" objects) around the world, such as serving website content, streaming videos, or gaming and mobile applications.	$0.026
Regional Storage	• 99.99% typical monthly availability • 99.9% availability SLA* • Lower cost per GB stored • Data stored in a narrow geographic region • Redundant across availability zones	Storing frequently accessed in the same region as your Google Cloud DataProc or Google Compute Engine instances that use it, such as for data analytics.	$0.02
Nearline Storage	• 99.9% typical monthly availability • 99.0% availability SLA* • Very low cost per GB stored • Data retrieval costs • Higher per-operation costs • 30-day minimum storage duration	Data you do not expect to access frequently (i.e., no more than once per month). Ideal for back-up and serving long-tail multimedia content.	$0.01
Coldline Storage	• 99.9% typical monthly availability • 99.0% availability SLA* • Lowest cost per GB stored • Data retrieval costs • Higher per-operation costs • 90-day minimum storage duration	Data you expect to access infrequently (i.e., no more than once per year). Typically this is for disaster recovery, or data that is archived and may or may not be needed at some future time.	$0.007

2.4 - Planning and configuring network resources

2.4.1 - Differentiating load balancing options

https://cloud.google.com/files/internal-load-balancing-tutorial-slides.pdf
https://cloud.google.com/compute/docs/load-balancing/internal/

2.4.2 - Identifying resource locations in a network for availability

2.4.3 - Configuring Cloud DNS

https://cloud.google.com/dns/quickstart

3 - Deploying and implementing a cloud solution

3.1 - Deploying and implementing Compute Engine resources

3.1.1 - Launching a compute instance using Cloud Console and Cloud SDK (gcloud)

https://cloud.google.com/sdk/gcloud/reference/compute/instances/create

3.1.1.1 - assign disks

```
[--disk=[auto-delete=AUTO-DELETE],[boot=BOOT],[device-name=DEVICE-NAME],[mode=MODE],[name=NAME]]
```

https://cloud.google.com/sdk/gcloud/reference/compute/instances/attach-disk

3.1.1.2 - availability policy

3.1.1.3 - SSH keys

3.1.2 - Creating an autoscaled managed instance group using an instance template

https://cloud.google.com/compute/docs/autoscaler/#managed_instance_groups
https://cloud.google.com/compute/docs/instance-groups/
https://cloud.google.com/compute/docs/instance-templates/

You can create two types of managed instance groups:

- A zonal managed instance group, which contains instances from the same zone.
- A regional managed instance group, which contains instances from multiple zones across the same region.

3.1.3 - Generating/uploading a custom SSH key for instances

https://cloud.google.com/compute/docs/instances/adding-removing-ssh-keys

3.1.4 - Configuring a VM for Stackdriver monitoring and logging

https://cloud.google.com/logging/docs/agent/installation

3.1.5 - Assessing compute quotas and requesting increases

https://console.cloud.google.com/iam-admin/quotas?project=xxx
https://cloud.google.com/compute/quotas
https://cloud.google.com/appengine/quotas
https://cloud.google.com/pubsub/quotas
https://cloud.google.com/bigquery/quotas
https://cloud.google.com/bigquery/docs/custom-quotas
https://cloud.google.com/functions/quotas
https://cloud.google.com/datastore/pricing
https://cloud.google.com/deployment-manager/pricing-and-quotas
https://cloud.google.com/monitoring/quotas
https://cloud.google.com/logging/quotas
https://cloud.google.com/endpoints/docs/openapi/quotas-configure

3.1.6 - Installing the Stackdriver Agent for monitoring and logging

https://cloud.google.com/monitoring/agent/install-agent

3.2 - Deploying and implementing Kubernetes Engine resources

Kubernetes Design and Architecture

3.2.1 - Deploying a Kubernetes Engine cluster

https://cloud.google.com/kubernetes-engine/docs/how-to/creating-a-container-cluster
https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture

3.2.2 - Deploying a container application to Kubernetes Engine using pods

```
gcloud config set container/cluster [CLUSTER_NAME]
gcloud container clusters get-credentials [CLUSTER_NAME]
```

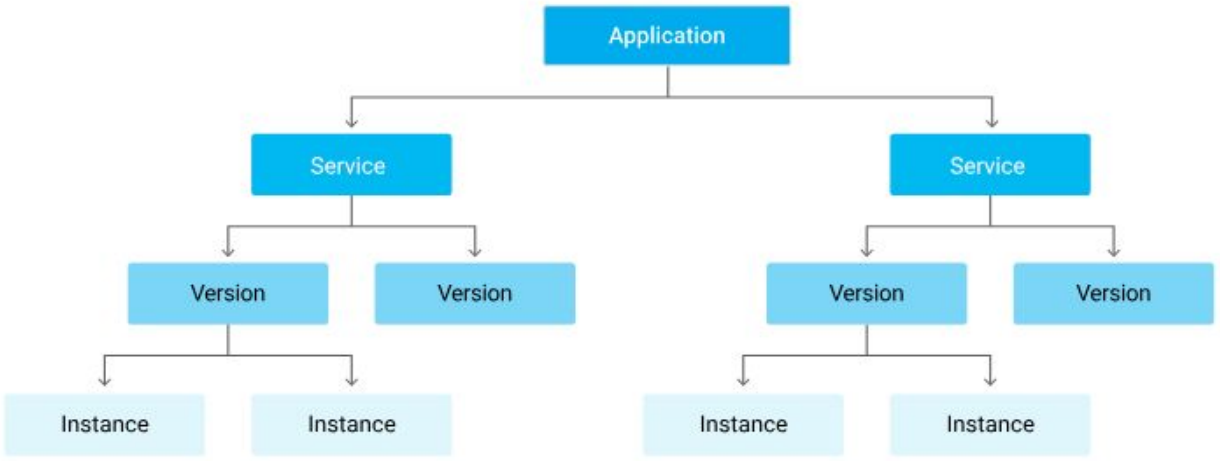
3.2.3 - Configuring Kubernetes Engine application monitoring and logging

https://kubernetes.io/docs/tasks/debug-application-cluster/logging-stackdriver/

```
gcloud beta container clusters update --logging-service=none CLUSTER
kubectl get ds fluentd-gcp-v2.0 --namespace kube-system -o yaml > fluentd-gcp-ds.yaml
kubectl replace -f fluentd-gcp-ds.yaml
```

3.3 - Deploying and implementing App Engine and Cloud Functions resources

3.3.1 - Deploying an application to App Engine



3.3.1.1 - scaling configuration

https://cloud.google.com/appengine/docs/standard/python/how-instances-are-managed

The scaling type you assign to a service determines the whether its instances are resident or dynamic:

- Auto scaling services use dynamic instances.
- Manual scaling services use resident instances.
- Basic scaling services use dynamic instances.

Manual scaling

A service with manual scaling use resident instances that continuously run the specified number of instances irrespective of the load level. This allows tasks such as comple initializations and applications that rely on the state of the memory over time.

Automatic scaling

Auto scaling services use dynamic instances that get created based on request rate, response latencies, and other application metrics. However, if you specify a number of minimum idle instances, that specified number of instances run as resident instances while any additional instances are dynamic.

Basic Scaling

A service with basic scaling use dynamic instances. Each instance is created when the application receives a request. The instance will be turned down when the app becomes idle. Basic scaling is ideal for work that is intermittent or driven by user activity.

3.3.1.2 - versions

The recommended approach is to remove the version element from your app.yaml file and instead, use a command-line flag to specify your version ID:

https://cloud.google.com/appengine/docs/admin-api/deploying-apps

```
gcloud app deploy -v [YOUR_VERSION_ID]
appcfg.py update -V [YOUR_VERSION_ID]
```

3.3.1.3 - traffic splitting

https://cloud.google.com/sdk/gcloud/reference/app/services/set-traffic

3.3.2 - Deploying a Cloud Function that receives Google Cloud events

3.3.2.1 - Cloud Pub/Sub events

https://cloud.google.com/functions/docs/tutorials/pubsub

3.3.2.2 - Cloud Storage object change notification events

https://cloud.google.com/functions/docs/calling/storage

3.4 - Deploying and implementing data solutions

3.4.1 - Initializing data systems with products

3.4.1.1 - Cloud SQL

3.4.1.2 - Cloud Datastore

3.4.1.3 - Cloud Bigtable

3.4.1.4 - BigQuery

3.4.1.5 - Cloud Spanner

3.4.1.6 - Cloud Pub/Sub

3.4.1.7 - Cloud Dataproc

3.4.1.8 - Cloud Storage

3.4.2 - Loading data

3.4.2.1 - Command line upload

3.4.2.2 - API transfer

https://cloud.google.com/storage/transfer/reference/rest/

3.4.2.3 - Import / export

https://cloud.google.com/sql/docs/mysql/import-export/
https://cloud.google.com/sql/docs/postgres/import-export/importing
https://cloud.google.com/sql/docs/postgres/import-export/
https://cloud.google.com/datastore/docs/export-import-entities

3.4.2.4 - load data from Cloud Storage

3.4.2.5 - streaming data to Cloud Pub/Sub

https://cloud.google.com/pubsub/docs/quickstart-cli

3.5 - Deploying and implementing networking resources

3.5.1 - Creating a VPC with subnets

3.5.1.1 - Custom-mode VPC

3.5.1.2 - Shared VPC

3.5.2 - Launching a Compute Engine instance with custom network configuration

https://cloud.google.com/sdk/gcloud/reference/compute/networks/subnets/create

3.5.2.1 - Internal-only IP address

`[--enable-private-ip-google-access]`
Enable/disable access to Google Cloud APIs from this subnet for instances without a public ip address.

3.5.2.2 - Google private access

https://cloud.google.com/sdk/gcloud/reference/compute/instances/create
`[--private-network-ip=PRIVATE_NETWORK_IP]`

3.5.2.3 - Static external and private IP address

`[--address=ADDRESS | --no-address]`

3.5.2.4 - network tags

`[--tags=TAG,[TAG,…]]`

3.5.3 - Creating ingress and egress firewall rules for a VPC

https://cloud.google.com/sdk/gcloud/reference/compute/firewall-rules/
`[--direction=DIRECTION]`
If direction is NOT specified, then default is to apply on incoming traffic. For incoming traffic, it is NOT supported to specify destination-ranges; For outbound traffic, it is NOT supported to specify source-ranges or source-tags.
For convenience, 'IN' can be used to represent ingress direction and 'OUT' can be used to represent egress direction.
DIRECTION must be one of: **INGRESS**, **EGRESS**, **IN**, **OUT**.

3.5.3.1 - IP subnets

`[--source-ranges=CIDR_RANGE,[CIDR_RANGE,…]]`
A list of IP address blocks that are allowed to make inbound connections that match the firewall rule to the instances on the network. The IP address blocks must be specified in CIDR format: http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing.
If neither --source-ranges nor --source-tags are specified, --source-ranges defaults to 0.0.0.0/0, which means that the rule applies to all incoming connections from inside or outside the network. If both --source-ranges and --source-tags are specified, the rule matches if either the range of the source matches --source-ranges or the tag of the source matches --source-tags.
If neither --source-ranges nor --source-tags is provided, then this flag will default to 0.0.0.0/0, allowing all sources. Multiple IP address blocks can be specified if they are separated by commas.
`[--destination-ranges=CIDR_RANGE,[CIDR_RANGE,…]]`
The firewall rule will apply to traffic that has destination IP address in these IP address block list. The IP address blocks must be specified in CIDR format:http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing.
If --destination-ranges is NOT provided, then this flag will default to 0.0.0.0/0, allowing all destinations. Multiple IP address blocks can be specified if they are separated by commas.

3.5.3.2 - Tags

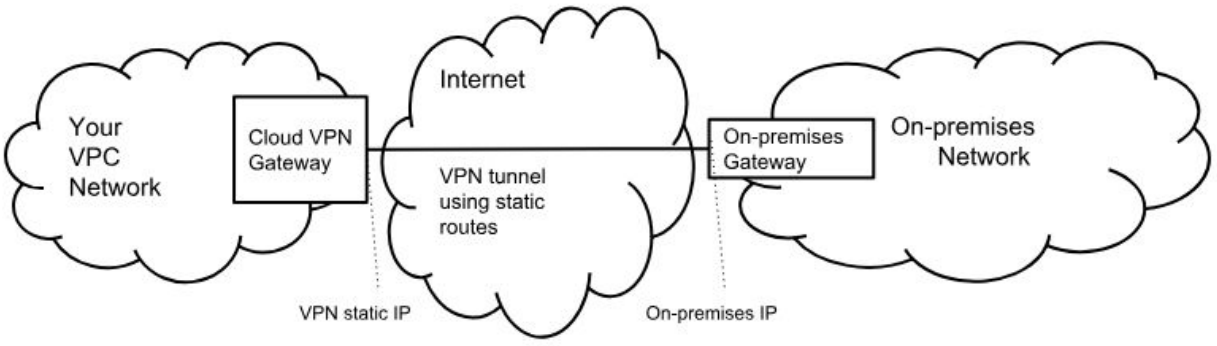
`[--source-tags=TAG,[TAG,…]]`
A list of instance tags indicating the set of instances on the network to which the rule applies if all other fields match. If neither --source-ranges nor --source-tags are specified, --source-ranges defaults to 0.0.0.0/0, which means that the rule applies to all incoming connections from inside or outside the network.
If both --source-ranges and --source-tags are specified, an inbound connection is allowed if either the range of the source matches --source-ranges or the tag of the source matches --source-tags.
Tags can be assigned to instances during instance creation.
If source tags are specified then neither a source nor target service account can also be specified.
`[--target-tags=TAG,[TAG,…]]`
A list of instance tags indicating the set of instances on the network which may accept inbound connections that match the firewall rule. If both target tags and target service account are omitted, all instances on the network can receive inbound connections that match the rule.
Tags can be assigned to instances during instance creation.
If target tags are specified then neither a source nor target service account can also be specified.

3.5.3.3 - Service accounts

`[--source-service-accounts=EMAIL,[EMAIL,…]]`
The email of a service account indicating the set of instances on the network which match a traffic source in the firewall rule.
If a source service account is specified then neither source tags nor target tags can also be specified.
`[--target-service-accounts=EMAIL,[EMAIL,…]]`
The email of a service account indicating the set of instances to which firewall rules apply. If both target tags and target service account are omitted, the firewall rule is applied to all instances on the network.
If a target service account is specified then neither source tag nor target tags can also be specified.

3.5.4 - Creating a VPN between a Google VPC and an external network using Cloud VPN

This diagram shows a simple VPN connection between your Cloud VPN gateway and your on-premises VPN gateway.



VPN diagram (click to enlarge)

https://cloud.google.com/vpn/docs/concepts/overview
https://cloud.google.com/vpn/docs/how-to/creating-vpns

3.5.5 - Creating a load balancer to distribute application network traffic to an application

3.5.5.1 - Global HTTP(S) load balancer
https://cloud.google.com/compute/docs/load-balancing/http/
3.5.5.2 - Global SSL Proxy load balancer
https://cloud.google.com/compute/docs/load-balancing/tcp-ssl/
3.5.5.3 - Global TCP Proxy load balancer
https://cloud.google.com/compute/docs/load-balancing/tcp-ssl/tcp-proxy
3.5.5.4 - Regional Network load balancer
https://cloud.google.com/compute/docs/load-balancing/network/
3.5.5.5 - Regional Internal load balancer
https://cloud.google.com/solutions/internal-load-balancing-haproxy

3.6 - Deploying a Solution using Cloud Launcher

3.6.1 - Browsing Cloud Launcher catalog and viewing solution details

https://console.cloud.google.com/launcher
3.6.2 - Deploying a Cloud Launcher marketplace solution

https://console.cloud.google.com/launcher

3.7 - Deploying an Application using Deployment Manager

3.7.1 - Developing Deployment Manager templates to automate deployment of an application

https://github.com/GoogleCloudPlatform/deploymentmanager-samples
3.7.2 - Launching a Deployment Manager template to provision GCP resources and configure an application automatically

VM with Startup Script
Creates a VM with user specified disks attached to it.
Cloud Functions Example
Cloud Container Builder Example
GKE Cluster and Type
Custom IAM Role

4 - Ensuring successful operation of a cloud solution

4.1 - Managing Compute Engine resources

4.1.1 - Managing a single VM instance

4.1.1.1 - start
https://cloud.google.com/sdk/gcloud/reference/compute/instances/start
4.1.1.2 - stop
https://cloud.google.com/sdk/gcloud/reference/compute/instances/stop
4.1.1.3 - edit configuration

4.1.1.4 - delete an instance
https://cloud.google.com/sdk/gcloud/reference/compute/instances/delete
4.1.2 - SSH/RDP to the instance

https://cloud.google.com/compute/docs/instances/connecting-to-instance
4.1.3 - Attaching a GPU to a new instance and installing CUDA libraries

You can attach GPUs only to instances with a predefined machine type or custom machine type that you are able to create in a zone. GPUs are not supported on shared-core machine types or memory-optimized machine types.

https://cloud.google.com/compute/docs/gpus/add-gpus

- `[ACCELERATOR_COUNT]` is the number of GPUs that you want to add to your instance. See GPUs on Compute Engine for a list of GPU limits based on the machine type of your instance.
- `[ACCELERATOR_TYPE]` is the GPU model that you want to use. See GPUs on Compute Engine for a list of available GPU models.

4.1.4 - Viewing current running VM Inventory

4.1.4.1 - instance IDs
4.1.4.2 - details

4.1.5 - Working with snapshots

4.1.5.1 - create a snapshot from a VM

https://cloud.google.com/compute/docs/disks/create-snapshots
https://cloud.google.com/sdk/gcloud/reference/compute/disks/create

`--source-snapshot=SOURCE_SNAPSHOT`

A source snapshot used to create the disks. It is safe to delete a snapshot after a disk has been created from the snapshot. In such cases, the disks will no longer reference the deleted snapshot. To get a list of snapshots in your current project, run gcloud compute snapshots list. A snapshot from an existing disk can be created using the gcloud compute disks snapshot command. This flag is mutually exclusive with --image.
When using this option, the size of the disks must be at least as large as the snapshot size. Use --size to adjust the size of the disks.

4.1.5.2 - view snapshots

https://cloud.google.com/sdk/gcloud/reference/compute/snapshots/list

4.1.5.3 - delete a snapshot

https://cloud.google.com/sdk/gcloud/reference/compute/snapshots/delete

4.1.6 - Working with Images

4.1.6.1 - create an image from a VM or a snapshot

https://cloud.google.com/sdk/gcloud/reference/compute/images/create

4.1.6.2 - view images

https://cloud.google.com/sdk/gcloud/reference/compute/images/list

4.1.6.3 - delete an image

https://cloud.google.com/sdk/gcloud/reference/compute/images/delete

4.1.7 - Working with Instance Groups

https://cloud.google.com/compute/docs/instance-groups/
https://cloud.google.com/compute/docs/instance-groups/updating-managed-instance-groups

4.1.7.1 - set auto scaling parameters

https://cloud.google.com/compute/docs/autoscaler/

Managed instance groups and autoscaling
Managed instance groups support autoscaling so you can dynamically add or remove instances from a managed instance group in response to increases or decreases in load. You enable autoscaling and choose an autoscaling policy to determine how you want to scale. Applicable autoscaling policies include scaling based on CPU utilization, load balancing capacity, Stackdriver monitoring metrics, or by a queue-based workload like Google Cloud Pub/Sub. Because autoscaling requires adding and removing instances from a group, you can only use autoscaling with managed instance groups so the autoscaler can maintain identical instances. ***Autoscaling does not work on unmanaged instance groups***, which can contain heterogeneous instances.
For more information, read Autoscaling Groups of Instances.

4.1.7.2 - assign instance template

https://cloud.google.com/compute/docs/instance-templates/

4.1.7.3 - create an instance template

https://cloud.google.com/compute/docs/instance-templates/create-instance-templates
https://cloud.google.com/sdk/gcloud/reference/compute/instance-groups/managed/create

4.1.7.4 - remove instance group

https://cloud.google.com/sdk/gcloud/reference/compute/instance-groups/managed/delete

4.1.8 - Working with management interfaces

4.1.8.1 - Cloud Console
4.1.8.2 - Cloud Shell
4.1.8.3 - GCloud SDK

4.2 - Managing Kubernetes Engine resources

https://kubernetes.io/docs/reference/kubectl/cheatsheet/

4.2.1 - Viewing current running cluster inventory

4.2.1.1 - nodes

```
kubectl get nodes
```

4.2.1.2 - pods

```
kubectl get pods
```

4.2.1.3 - services

```
kubectl get services
```

4.2.2 - Browsing the container image repository and viewing container image details

4.2.3 - Working with nodes

https://cloud.google.com/kubernetes-engine/docs/how-to/resizing-a-container-cluster

4.2.3.1 - add a node

https://cloud.google.com/sdk/gcloud/reference/container/clusters/resize

```
gcloud container clusters resize [CLUSTER_NAME] \
--node-pool [NODE_POOL] \
    --size [SIZE]
```

4.2.3.2 - edit a node

4.2.3.3 - remove a node

4.2.4 - Working with pods

4.2.4.1 - add pods

4.2.4.2 - edit pods

4.2.4.3 - remove pods

4.2.5 - Working with services

4.2.5.1 - add a service

4.2.5.2 - edit a service

4.2.5.3 - remove a service

4.2.6 - Working with management interfaces

4.2.6.1 - Cloud Console
4.2.6.2 - Cloud Shell
4.2.6.3 - Cloud SDK

4.3 - Managing App Engine resources

4.3.1 - Adjusting application traffic splitting parameters

4.3.2 - Setting scaling parameters for autoscaling instances

4.3.3 - Working with management interfaces

4.3.3.1 - Cloud Console
4.3.3.2 - Cloud Shell
4.3.3.3 - Cloud SDK

4.4 - Managing data solutions

4.4.1 - Executing queries to retrieve data from data instances

4.4.1.1 - Cloud SQL
4.4.1.2 - BigQuery
4.4.1.3 - Cloud Spanner
4.4.1.4 - Cloud Datastore
4.4.1.5 - Cloud Bigtable
4.4.1.6 - Cloud Dataproc

4.4.2 - Estimating costs of a BigQuery query

https://cloud.google.com/bigquery/docs/estimate-costs

4.4.3 - Backing up and restoring data instances

4.4.3.1 - Cloud SQL
4.4.3.2 - Cloud Datastore
4.4.3.3 - Cloud Dataproc

4.4.4 - Reviewing job status in Cloud Dataproc or BigQuery

4.4.5 - Moving objects between Cloud Storage buckets

4.4.6 - Converting Cloud Storage buckets between storage classes

4.4.7 - Setting object lifecycle management policies for Cloud Storage buckets

4.4.8 - Working with management interfaces

4.4.8.1 - Cloud Console
4.4.8.2 - Cloud Shell
4.4.8.3 - Cloud SDK

4.5 - Managing networking resources

4.5.1 - Adding a subnet to an existing VPC

https://cloud.google.com/vpc/docs/using-vpc

Adding a new subnet to an existing VPC network
You can add a subnet to a region of an existing VPC network. The primary IP range of this new subnet cannot overlap the IP range of existing subnets in the current network, in peered VPC networks, or in on-premises networks connected via VPN or Interconnect.

You can optionally assign a secondary IP range to the subnet for use with Alias IP. The secondary IP range also cannot overlap the IP ranges of existing connected subnets.

CONSOLE GCLOUD

```
gcloud compute networks subnets create [SUBNET_NAME] \
--network [NETWORK] \
--range [IP_RANGE] \
[--secondary-range [RANGE_NAME]=[2ND_IP_RANGE]
```

where

[SUBNET_NAME] is the name of the new subnet you are creating
[NETWORK] is the name of the existing network where you are creating the new subnet.
[IP_RANGE] is the primary IP range of the subnet. Example: 192.168.0.0/20.
[2ND_RANGE_NAME] is the name of the secondary IP range you can optionally create.
[2ND_IP_RANGE] is the range of the secondary IP range you can optionally create. Example: 172.16.0.0/16.

4.5.2 - Expanding a CIDR block subnet to have more IP addresses

Expanding a subnet
You can expand the IP range of a subnet. You cannot shrink it.
Restrictions:

- The new subnet must not overlap with other subnets in the same VPC network in any region.
- The new subnet must stay inside the RFC 1918 address spaces.
- The new network range must be larger than the original, which means the prefix length value must be a smaller number.

- Auto mode subnets start with a /20 IP range. They can be expanded to a /16, but no larger.

```
gcloud compute networks subnets expand-ip-range [SUBNET_NAME] \
  --region [REGION] \
  --prefix-length [PREFIX_LENGTH]
```

[SUBNET_NAME] - the name of the subnet whose IP range you want to expand. You do not have to specify the network because the subnet and region together identify the network.
[REGION] - the region the subnet exists in.
[PREFIX_LENGTH] - the new numeric prefix length for the subnet. Must be smaller than the existing prefix length. For example, if the current subnet is a /24, the new prefix length must be 23 or smaller. This might change the first IP in the range. For example, if the original IP range was 10.128.131.0/24, specifying --prefix-length 20 sets the new IP range to 10.128.128.0/20.

4.5.3 - Reserving static external or internal IP addresses

4.5.3.1 - Reserving static external IP addresses

https://cloud.google.com/compute/docs/ip-addresses/reserve-static-external-ip-address

4.5.3.2 - Reserving static internal IP addresses

https://cloud.google.com/compute/docs/ip-addresses/reserve-static-internal-ip-address

4.5.4 - Working with management interfaces

4.5.4.1 - Cloud Console
4.5.4.2 - Cloud Shell
4.5.4.3 - Cloud SDK

4.6 - Monitoring and logging

4.6.1 - Creating Stackdriver alerts based on resource metrics

https://cloud.google.com/monitoring/custom-metrics/creating-metrics
Choosing a monitored resource type
Each of your metric's data points must include a monitored resource object. Points from different monitored resource objects are held in different time series.

You can use only the following monitored resource types in your custom metrics:

- **gce_instance** Google Compute Engine instance.
- **gke_container** Google Kubernetes Engine container.
- **dataflow_job** Dataflow job.
- **aws_ec2_instance** Amazon EC2 instance.
- **global** Anything else.

A common practice is to use the monitored resource object that represents the physical resource where your application code is running. This has several advantages:

- You get better performance.
- You avoid out-of-order data caused by multiple instances writing to the same time series.
- Your custom metric data can be grouped with other metric data from the same instance.

If none of the instance-related resource types are appropriate, use global. For example, Google App Engine users should use global because the resource type gae_app is not permitted in custom metrics.

4.6.2 - Creating Stackdriver custom metrics

https://cloud.google.com/monitoring/custom-metrics/
https://cloud.google.com/monitoring/api/v3/metrics-details

4.6.3 - Configuring log sinks to export logs to external systems

4.6.3.1 - on premises
4.6.3.2 - BigQuery

4.6.4 - Viewing and filtering logs in Stackdriver

4.6.5 - Viewing specific log message details in Stackdriver

4.6.6 - Using cloud diagnostics to research an application issue

4.6.6.1 - viewing Cloud Trace data
4.6.6.2 - using Cloud Debug to view an application point-in-time

4.6.7 - Viewing Google Cloud Platform status

4.6.8 - Working with management interfaces

4.6.8.1 - Cloud Console
4.6.8.2 - Cloud Shell
4.6.8.3 - Cloud SDK

5 - Configuring access and security

5.1 - Managing Identity and Access Management (IAM)

5.1.1 - Viewing account IAM assignments

https://cloud.google.com/sdk/gcloud/reference/projects/get-iam-policy
https://cloud.google.com/sdk/gcloud/reference/organizations/get-iam-policy

5.1.2 - Assigning IAM roles to accounts or Google Groups

https://cloud.google.com/sdk/gcloud/reference/projects/set-iam-policy
https://cloud.google.com/sdk/gcloud/reference/projects/add-iam-policy-binding
https://cloud.google.com/sdk/gcloud/reference/organizations/set-iam-policy
https://cloud.google.com/sdk/gcloud/reference/organizations/add-iam-policy-binding

5.1.3 - Defining custom IAM roles

https://cloud.google.com/iam/docs/creating-custom-roles
https://cloud.google.com/sdk/gcloud/reference/iam/roles/
https://cloud.google.com/sdk/gcloud/reference/iam/roles/create
https://cloud.google.com/iam/reference/rest/v1/projects.roles

5.2 - Managing service accounts

https://cloud.google.com/compute/docs/access/service-accounts

5.2.1 - Managing service accounts with limited scopes

Best practices
In general, Google recommends that each instance that needs to call a Google API should run as a service account with the minimum permissions necessary for that instance to do its job. In practice, this means you should configure service accounts for your instances with the following process:

1. Create a new service account rather than using the Compute Engine default service account.
2. Grant IAM roles to that service account for only the resources that it needs.
3. Configure the instance to run as that service account.
4. Grant the instance the `https://www.googleapis.com/auth/cloud-platform` scope.

Avoid granting more access than necessary and regularly check your service account permissions to make sure they are up-to-date.

5.2.2 - Assigning a service account to VM instances

https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-instances

5.2.3 - Granting access to a service account in another project

5.3 - Viewing audit logs for project and managed service

https://cloud.google.com/compute/docs/audit-logging
Cloud Audit Logging returns two types of logs:

Admin activity logs: Contains log entries for operations that *modify the configuration or metadata of a Compute Engine resource*. Any API call that *modifies a resource such as creation, deletion, updating, or modifying a resource using a custom verb fall into this category*.

Data access logs: Contains log entries for operations that perform read-only operations do not modify any data, such as get, list, and aggregated list methods. Unlike audit logs for other services, Compute Engine only has ADMIN_READ data access logs and do not generally offer DATA_READ and DATA_WRITE logs. This is because DATA_READ and DATA_WRITE logs are only used for services that store and manage user data such as Google Cloud Storage, Google Cloud Spanner, and Google Cloud SQL, which does not apply to Compute Engine. There is one exception to this rule: the **instance.getSerialPortOutput** does generate a DATA_READ log because the method reads data directly from the VM instance.