

Study Guide

Associate Cloud Engineer



Linux Academy



Cloud Assessments

Contents

Section 1: Setting up a Cloud Solution Environment	1
1.1 Setting up Cloud Projects and Accounts	1
1.2 Managing Billing Configuration	2
1.3 Installing and Configuring the Command Line Interface (CLI): Specifically the Cloud SDK	3
Section 2: Planning and Configuring a Cloud Solution	4
2.1 Planning and Estimating GCP Product Use Using the Pricing Calculator	4
2.2 Planning and Configuring Compute Resources	4
2.3 Planning and Configuring Data Storage Options	6
2.4 Planning and Configuring Network Resources	7
Section 3: Deploying and Implementing a Cloud Solution	8
3.1 Deploying and Implementing Compute Engine Resources	8
3.2 Deploying and implementing Kubernetes Engine resources	9
3.3 Deploying and Implementing App Engine and Cloud Functions Resources	10
3.5 Deploying and implementing networking resources. Tasks include:	11
3.6 Deploying a Solution Using Cloud Launcher	12
3.7 Deploying an Application Using Deployment Manager	12
Section 4: Ensuring successful operation of a cloud solution	13
4.2 Managing Kubernetes Engine Resources	15
4.3 Managing App Engine Resources	17
4.4 Managing data solutions	18
4.5 Managing Networking Resources	20
4.6 Monitoring and Logging	20
Section 5: Configuring access and security	23
5.1 Managing Identity and Access Management (IAM)	23
5.2 Managing Service Accounts	23
5.3 Viewing Audit Logs for Project and Managed Services	25

Section 1: Setting up a Cloud Solution Environment

1.1 Setting up Cloud Projects and Accounts

- Creating projects:
 - Projects are isolated boxes for Google Cloud resources. All resources belong to a project
 - Create projects in the Console or with the CLI
 - `gcloud projects create --organization=ORGANIZATION_ID [PROJECT_ID]`
- Assigning users to pre-defined IAM roles within a project:
 - Pre-defined roles make it easy to grant access to resources
 - `gcloud [GROUP] add-iam-policy-binding [RESOURCE-NAME] --member user:[USER-EMAIL]--role [ROLE-ID]`
 - Example: `gcloud projects add-iam-policy-binding PROJECT_ID --member user:bigquery.jobUser --role roles/bigquery.jobUser --member serviceAccount:SA_EMAIL`
- Linking users to G Suite identities:
 - Google requires users to exist in an IDP that they control. You can use:
 - Gmail accounts
 - Google group accounts
 - Service accounts
 - GSuite users
 - Google Identity users
 - You can link on-premises IDPs using Directory Sync

- Enabling APIs within projects:
 - You can enable APIs using the console or the CLI
 - Examples:
 - `gcloud services enable storage-component.googleapis.com`
 - `gcloud services enable bigtable.googleapis.com`
 - Find existing and enabled APIs:
 - `gcloud services list --available --sort-by="NAME"`
 - `gcloud services list --enabled --sort-by="NAME"`
- Provisioning one or more Stackdriver accounts:
 - Stackdriver can monitor one or more projects
 - Google recommends keeping Stackdriver in its own project if it's going to monitor more than one project

1.2 Managing Billing Configuration

- Creating one or more billing accounts:
 - Billing accounts are how you pay for the resources used in a project
- Linking projects to a billing account:
 - Once a billing account is created, it can be linked to one or more projects
 - `gcloud beta billing projects link PROJECT_ID --billing-account=ACCOUNT_ID [optional flags]`
- Establishing billing budgets and alerts:
 - You can set alerts based on a percentage of your monthly budget. If you hit the percentages, you'll be notified

- Setting up billing exports to estimate daily/monthly charges:
 - Billing exports allow you to export billing usage for reporting
 - Export formats are:
 - File (CSV, JSON)
 - BigQuery

1.3 Installing and Configuring the Command Line Interface (CLI): Specifically the Cloud SDK

Working with the Cloud SDK:

- The Cloud SDK is the command line interface used for managing Google Cloud. It consists of several components including, but not limited to:
 - `gcloud` - Google Cloud CLI
 - `gsutil` - Cloud Storage CLI
 - `bq` - BigQuery CLI
 - `kubectl` - Kubernetes CLI
- List off the existing components with: `gcloud components list`
- Install components with: `gcloud components install COMPONENT_ID`
- Remove components with: `gcloud components remove COMPONENT_ID`

Section 2: Planning and Configuring a Cloud Solution

2.1 Planning and Estimating GCP Product Use Using the Pricing Calculator

The price calculator is available at: <https://cloud.google.com/products/calculator/>

2.2 Planning and Configuring Compute Resources

Selecting appropriate compute choices for a given workload:

- Compute Engine
 - IaaS used for running virtual machines
 - Used when:
 - You need control over the OS
 - You need control over the CPU, GPU, SSDs, Memory, etc.
 - You need to lift-and-shift an application
 - You need batch processing with preemptible instances
- Kubernetes Engine
 - A managed version of Kubernetes.
 - Used when:
 - You're running complex containerized applications
 - You want to focus on containers as a unit of scale, deployment, and execution
 - You need portability

- App Engine
 - A highly scalable web platform
 - Used when:
 - You value development over ops
 - You need high availability
 - Portability isn't a concern
 - Your applications "speak HTTP"
 - You don't care about the underlying OS
- Cloud Functions
 - Execute code in response to events
 - Used when:
 - You need to run code in response to some cloud event
 - You can select the language and tools based on what's supported
 - Your code executes within the limits
 - You don't care about the underlying OS

Use preemptible VMs and custom machine types as appropriate. Preemptible VMs allow you to get a discounted rate on extra compute capacity

2.3 Planning and Configuring Data Storage Options

Considerations from data storage include:

- Product choice (e.g., Cloud SQL, BigQuery, Cloud Spanner, Cloud Bigtable)
 - Cloud SQL is used when you need a traditional SQL database
 - Spanner is used when you need a traditional SQL database that can scale horizontally
 - Bigtable is a sparse table, used when you have massive amounts of keyed data
 - BigQuery is a data warehouse service that can query massive data sets quickly
 - Datastore is a NoSQL database built on top of Bigtable
- Choosing cloud storage options
 - Cloud Storage has multiple storage classes
 - Multi-regional
 - Highest availability (99.95%)
 - Data replicated to multiple regions
 - Used for:
 - Content storage
 - Video
 - Multimedia
 - Regional
 - Second highest availability (99.95%)
 - Data replicated to multiple zones inside the single region
 - Used for:
 - Analytics data
 - Transcoding

- Compute intensive data processing storage
- Nearline
 - Stores infrequently used data - less than once per month
 - Used for:
 - Backup files
 - Rarely accessed documents
- Coldline
 - Stores infrequently used data - less than once per year
 - Used for:
 - Archive files
 - Disaster recovery

2.4 Planning and Configuring Network Resources

Differentiating load balancing options:

- Google offers global and regional load balancers. Selecting the option that's right for your application depends on the traffic type.
- Global load balancing options:
 - HTTP(S)
 - SSL Proxy
 - TCP Proxy
- Regional load balancing options:
 - Network TCP/UDP load balancing
 - Internal TCP/UDP load balancing

Section 3: Deploying and Implementing a Cloud Solution

3.1 Deploying and Implementing Compute Engine Resources

- Launching a compute instance using Cloud Console and Cloud SDK:
 - Launching an instance from the Console is as simple as filling out a form. Using the CLI requires a few commands
 - To create an instance for an image family:
 - `gcloud compute instances create development-server --machine-type n1-standard-2 --zone us-east1-d --image-family=ubuntu-1604-lts --subnet=big-data-dev-subnet --metadata-from-file ssh-keys=ssh_keys.txt --project=find-seller-big-data --image-project ubuntu-os-cloud`
- Creating an autoscaled managed instance group using an instance template:
 - Managed instance groups make it easy to dynamically start instances in response to scaling events such as CPU usage, HTTP requests, etc.
 - The instances are built based on a template that describes all of the required properties
- Generating/uploading a custom SSH key for instances:
 - SSH keys are set as either a project or instance metadata
- Configuring a VM for Stackdriver monitoring and logging:
 - To collect monitoring and logging data from an instance, you can add the monitoring and logging agents on the VM.
- Assessing compute quotas and requesting increases:
 - The APIs behind Google Cloud have limits.
 - To view your quotas, you can use the `gcloud compute project-info describe --project myproject` command
 - To request an increase contact support use: <https://console.cloud.google.com/iam-admin/quotas>

- Installing the Stackdriver Agent for monitoring and logging:
 - Monitoring agent: <https://cloud.google.com/monitoring/agent/install-agent>
 - Logging agent: <https://cloud.google.com/logging/docs/agent/installation>

3.2 Deploying and implementing Kubernetes Engine resources

- Deploying a Kubernetes Engine cluster:
 - Deploying a cluster can be done easily in the UI or with the CLI
 - Example:

```
gcloud beta container clusters create $PRODUCT_CLUSTER_NAME
-project $PROJECT_NAME
-zone $PROJECT_ZONE
-no-enable-basic-auth
-cluster-version "1.9.7-gke.3"
-machine-type "n1-standard-1"
-image-type "COS"
-disk-type "pd-standard"
-disk-size "100"
-num-nodes "3"
-enable-cloud-logging
-enable-cloud-monitoring
-network $SERVICES_NETWORK
-subnetwork $PRODUCT_SUBNET
-addons
HorizontalPodAutoscaling,HttpLoadBalancing,KubernetesDashboard
-enable-autoupgrade
-enable-autorepair
-service-account $SA_EMAIL
```

- Deploying a container application to Kubernetes Engine using pods:
 - Pods are a “collection” of containers that live together on the same host. They do everything together. They’re deployed together, scaled together, etc.
 - A pod can have only one container, which is the most common way to see Pods used
 - Pods can be created with the `kubectl run` command or defined in YAML and deployed with `kubectl apply`

- Configuring Kubernetes Engine application monitoring and logging:
 - Enabling logging and monitoring is done at the cluster level
 - It can be done during or after cluster creation

3.3 Deploying and Implementing App Engine and Cloud Functions Resources

- Deploying an application to App Engine
 - App Engine provides traffic splitting to perform:
 - A/B testing
 - Canary deployments
 - Rollbacks
 - App Engine also provides multiple scaling mechanisms
 - Automatic
 - Manual
 - Basic (Standard environments only)
 - Applications are deployed with the `gcloud app deploy app.yaml` command
- Deploying a Cloud Function that receives Google Cloud events
 - Create and deploy a function with the `gcloud functions deploy` command
 - Example:

```
gcloud beta functions deploy $FUNCTION_NAME \
  --entry-point=imageParser \
  --source=$SOURCE_LOCAL_FOLDER \
  --stage-bucket=$PRIVATE_ASSETS \
  --trigger-resource=$PUB_SUB_TOPIC \
  --trigger-event="google.pubsub.topic.publish" \
  --project=$PROJECT_NAME \
  --region=$PROJECT_REGION \
  --set-env-vars=BIGTABLE_INSTANCE_ID=$BIGTABLE_INSTANCE_
ID,BIGTABLE_TABLE_ID=$BIGTABLE_TABLE_ID,CLOUD_STORAGE_
BUCKET=$PUBLIC_ASSETS
```

3.5 Deploying and implementing networking resources.

Tasks include:

- Creating a VPC with subnets:
 - Create a VPC with the `gcloud compute networks create` command:
 - The `--subnet-mode` flag can be set to custom or automatic
 - Create a subnet with the `gcloud compute networks subnets create` command:
 - The range is specified in CIDR notation:
 - Example: `--range=10.29.0.0/24`
- Launching a Compute Engine instance with custom network configuration:
 - Specify the network interface at creation time by setting the `--subnet=subnet-name` flag
 - Specify a static external IP address with the `--address` flag
 - Specify a static internal IP address with the `--private-network-ip` flag
- Creating ingress and egress firewall rules for a VPC:
 - Create firewall rules with the `gcloud compute firewall-rules create` command
 - Example:

```
gcloud compute firewall-rules create "allow-ssh"  
-network custom-dev-network  
-allow tcp:22
```

- Creating a load balancer to distribute application network traffic to an application:
 - Creating a load balancer in the Console is the simplest option.
 - However you can also create the independent resources depending on the type of load balancer you're using.
 - <https://cloud.google.com/load-balancing/docs/how-to>

3.6 Deploying a Solution Using Cloud Launcher

- Browsing Cloud Launcher catalog and viewing solution details:
 - You can select applications from the Marketplace and quickly launch them in your project
 - Deployment Manager is used behind the scenes to configure resources
- Deploying a Cloud Launcher marketplace solution:
 - Once you find an application you can install it and walk through the settings forms.

3.7 Deploying an Application Using Deployment Manager

- Developing Deployment Manager templates to automate the deployment of an application:
 - Deployment Manager is Google's infrastructure as code service. It supports multiple templating options, including:
 - YAML
 - Jinja
 - Python
- Launching a Deployment Manager template to provision GCP resources and configure an application automatically:
 - Create a deployment with the `gcloud deployment-manager deployments create ad-service-deployment --config template-name.yaml` command
 - Update a deployment with the `gcloud deployment-manager deployments update ad-service-deployment --config template-name.yaml` command

Section 4: Ensuring successful operation of a cloud solution

4.1 Managing Compute Engine Resources

Tasks include:

- Managing a single VM instance
 - Start one or more instances with the `gcloud compute instances start INSTANCE_NAMES` command
 - Stop one or more instances with the `gcloud compute instances stop INSTANCE_NAMES` command
 - Update an instance with the `gcloud compute instances update INSTANCE_NAME` command
 - Delete one or more instances with the `gcloud compute instances delete INSTANCE_NAMES` command
- SSH/RDP to the instance:
 - Connect to a Linux instance via SSH:
 - Ensure port 22 is open to your IP address
 - Connect with the `gcloud compute ssh` command
 - Connect with third-party clients
 - Run the `gcloud compute instances list` command to get the external IP address of the instance you wish to connect into
 - Connect via the Console
 - SSH keys can be uploaded as either project or instance metadata
 - Connect to a Windows instance via RDP

- Attaching a GPU to a new instance and installing CUDA libraries:
 - Attach an accelerator to an instance with the `--accelerator type=nvidia-tesla-k80,count=1` flag
 - Example:

```
gcloud compute instances create gpu-instance-1 \
--machine-type n1-standard-2 --zone us-east1-d \
--accelerator type=nvidia-tesla-k80,count=1 \
--image-family ubuntu-1604-lts --image-project ubuntu-os-cloud \
--maintenance-policy TERMINATE --restart-on-failure \
--metadata startup-script='#!/bin/bash
echo "Checking for CUDA and installing."
#Check for CUDA and try to install.
if ! dpkg-query -W cuda-9-0; then
    curl -O http://developer.download.nvidia.com/compute/cuda/
repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_9.0.176-1_amd64.deb
    dpkg -i ./cuda-repo-ubuntu1604_9.0.176-1_amd64.deb
    apt-key adv --fetch-keys http://developer.download.nvidia.com/
compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
    apt-get update
    apt-get install cuda-9-0 -y
fi'
```

- Viewing current running VM Inventory:
 - List the virtual machines with the `gcloud compute instances list` command
 - View the details with the `gcloud compute instances describe INSTANCE_ID` command
- Working with snapshots:
 - Create a snapshot of a disk with the `gcloud compute disks snapshot DISK_NAME` command
 - Create a snapshot of a regional disk with the `gcloud beta compute disks snapshot DISK_NAME --region REGION` command
 - Delete a snapshot with the `gcloud compute disks delete DISK_NAME` command
- Working with Images:
 - Create a VM image with the `gcloud compute images create` command:
 - The `--source-disk` flag allows the image to be created from an existing disk
 - The `--source-image` and `--source-image-project` flags allow the image to be created from an existing image

- The `--source-snapshot` flag allows the image to be created from an existing snapshot
- The `--source-uri` flag allows the image to be created from a local image uploaded to Cloud Storage
- Working with Instance Groups:
 - Managed instance groups require a template:
 - A template defines how the instance will be created when it's dynamically added to the instance group
 - Create a template with the `gcloud compute instance-templates create` command
 - Create a managed instance group with the `gcloud compute instance-groups managed create` command:
 - Use the `--template` flag to specify the template
 - Example:

```
gcloud compute instance-groups managed create example-managed-  
instance-group -zone us-central1-a -template example-instance-  
template -size 1
```

4.2 Managing Kubernetes Engine Resources

- Viewing current running cluster inventory (nodes, pods, services):
 - List nodes with the `kubectl get nodes` command
 - List pods with the `kubectl get pods` command
 - List services with the `kubectl get services` command
 - View node details with the `kubectl describe nodes NODE_NAME` command
 - View pods details with the `kubectl describe pods POD_NAME` command
 - View services details with the `kubectl describe services SERVICE_NAME` command
- Browsing the container image repository and viewing container image details:
 - List images in the container registry with the `gcloud container images list` command:

- The `--repository` flag allows you to specify the registry
- View image details with the `gcloud container images describe IMAGE` command
- Working with nodes:
 - Add/Remove a node with the `gcloud container clusters resize NAME --size=SIZE` command
 - Upgrade the nodes in the cluster to use a different Kubernetes version with the `gcloud container clusters upgrade NAME` command
 - Use the `--cluster-version` flag to set the Kubernetes version
 - View cluster details with the `kubectl cluster-info` flag
- Working with pods:
 - Use the `kubectl run` command to create a pod in a deployment
 - Use the `kubectl create -f manifest.yaml` file to create a deployment based on a YAML manifest
 - Use the `kubectl apply -f manifest.yaml` file to create/update a deployment based on a YAML manifest
 - Use the `kubectl get deployments` command to list off the deployments
 - Use the `kubectl delete` command to delete a deployment
- Working with services:
 - Use the `kubectl create -f manifest.yaml` file to create a service based on a YAML manifest
 - Use the `kubectl apply -f manifest.yaml` file to create/update a service based on a YAML manifest
 - Use the `kubectl get services` command to list off the services
 - Use the `kubectl delete` command to delete a services

4.3 Managing App Engine Resources

- Adjusting application traffic splitting parameters:
 - To split traffic between multiple versions use the `gcloud app services set-traffic` command:
 - Use the `--split-by` flag to set the splitting mechanism:
 - IP Address
 - Cookie
 - Random
 - Use the `--splits` flag to distribute the traffic
 - Example: `gcloud app services set-traffic s1 --splits v2=.5,v1=.5 --split-by=random`
- Setting scaling parameters for autoscaling instances:
 - Automatic
 - `target_cpu_utilization`: Specifies the CPU usage threshold at which new instances will be started to handle traffic
 - `min_instances`: Specifies the minimum number of instances to keep running
 - `max_instances`: Specifies the maximum number of instances to allow
 - `min_pending_latency`: The minimum amount of time for a request to wait in the pending queue before starting another instance
 - `max_pending_latency`: The maximum amount of time for a request to wait in the pending queue before starting another instance
 - `max_concurrent_requests`: The number of concurrent requests instance can accept before starting a new instance
 - `min_idle_instances`: The number of instances to have running at all times to ease burst traffic latency
 - Manual

- `instances`: The number of instances to run
- Basic (Standard Environments)
- `max_instances`: The maximum number of instances to run
- `idle_timeout`: The amount of idle time before shutting down an instance

4.4 Managing data solutions

- Estimating costs of a BigQuery query:
 - Use the `--dry-run` flag to estimate the amount of data read
- Backing up and restoring data instances:
 - Cloud SQL
 - Create an on-demand backup with the `gcloud sql backups create --async --instance INSTANCE_NAME` command
 - Schedule a backup with the `gcloud sql instances patch INSTANCE_NAME --backup-start-time HH:MM` command
 - Disable a backup with the `gcloud sql instances patch INSTANCE_NAME --no-backup` command
 - See <https://cloud.google.com/sql/docs/mysql/backup-recovery/restoring> for details on restoring from backup
 - Bigtable
 - See <https://cloud.google.com/bigtable/docs/exporting-sequence-files#bigtable> for details on exporting Bigtable data
 - See <https://cloud.google.com/bigtable/docs/importing-sequence-files#import-table> for detail on restoring from backup
- Reviewing job status in Cloud Dataproc or BigQuery
 - Dataproc
 - List Dataproc jobs with the `gcloud dataproc jobs list` command
 - View job details with the `gcloud dataproc jobs describe JOB` command

- BigQuery
 - List jobs with the `bq ls` command
 - View job details `bq show -j JOB`
- Moving objects between Cloud Storage buckets using the `gsutil mv` command
- Converting Cloud Storage buckets between storage classes using the `gsutil rewrite -s STORAGE_CLASS gs://PATH_TO_OBJECT` command
- Setting object lifecycle management policies for Cloud Storage buckets
 - Use the `gsutil lifecycle set LIFECYCLE_CONFIG_FILE gs://BUCKET_NAME` command to set a lifecycle
 - The config file is a JSON file
 - Example:

```
{
  "lifecycle": {
    "rule": [
      {
        "action": {"type": "Delete"},
        "condition": {
          "age": 30,
          "isLive": true
        }
      },
      {
        "action": {"type": "Delete"},
        "condition": {
          "age": 10,
          "isLive": false
        }
      }
    ]
  }
}
```

4.5 Managing Networking Resources

- Adding a subnet to an existing VPC
 - Use the `gcloud beta compute networks subnets create` command to add a subnet to an existing VPC
 - Example:

```
gcloud beta compute networks subnets create  
ADSUBNET --project=PROJECT_NAME  
-network=SERVICES_NETWORK --region=PROJECT_REGION  
-range=10.28.0.0/24  
-enable-private-ip-google-access  
-enable-flow-logs
```

- Expanding a CIDR block subnet to have more IP addresses
 - Use the `gcloud compute networks subnets expand-ip-range` command to expand a subnet range
 - Use the `--prefix-length` flag to set the mask bit (The number that follows the IP address in CIDR notation. For example 10.28.0.0/24 the mask bit is 24)
- Reserving static external or internal IP addresses
 - Use the `gcloud compute addresses create` command to create an external IP address
 - Use the `gcloud compute instances create INSTANCE_NAME --private-network-ip IP_ADDRESS` command to set an internal static IP address when creating an instance

4.6 Monitoring and Logging

- Creating Stackdriver alerts based on resource metrics
 - Stackdriver allows alerts to be created based on the different metrics that are tracked for different services
 - Alert policies consist of:
 - Conditions: Some event that should be monitored, e.g., CPU utilization > 60% for 5 minutes

- Notifications:
 - Email
 - Console mobile app
 - PagerDuty
 - SMS
 - Hipchat
 - Campfire
 - Webhook
 - Slack
- Documentation (optional): Information to help resolve the problem
- Policy name
- Creating Stackdriver custom metrics:
 - A custom monitoring metric is created and set in code using a client library:
 - See <https://cloud.google.com/monitoring/custom-metrics/creating-metrics> for details
 - A custom logging metric allows data to be extracted from logs and used as a metric:
 - See <https://cloud.google.com/logging/docs/logs-based-metrics/> for details
- Configuring log sinks to export logs to external systems
 - Log sinks allow you to create a filter
 - Any log record that matches the filter will be exported to the sink destination
 - Destinations:
 - Cloud Storage
 - BigQuery

- Cloud Pub/Sub
- Viewing and filtering logs in Stackdriver
 - There are two forms of filters
 - Basic: Use the drop-downs to select services, logs, log-level, etc.
 - Advanced: Boolean expressions that can be combined to create more complex queries: * Allowed boolean operators: * AND * OR * NOT * \-
 - Comparison operators:
 - <= less than or equal
 - < : less than
 - = : equal
 - \> : greater than
 - \>= : greater than or equal
 - != : not equal
 - See <https://cloud.google.com/logging/docs/view/advanced-filters> for more detail
- Viewing Google Cloud Platform status:
 - See <https://status.cloud.google.com/>

Section 5: Configuring access and security

5.1 Managing Identity and Access Management (IAM)

- Viewing account IAM assignments:
 - Use the `gcloud projects get-iam-policy PROJECT_ID` command to list IAM assignments.
- Assigning IAM roles to accounts:
 - Use the `gcloud projects add-iam-policy-binding RESOURCE-NAME --member user:USER-EMAIL --role ROLE-ID` command to set roles for members of a project
 - Use the `gcloud organizations add-iam-policy-binding RESOURCE-NAME --member user:USER-EMAIL --role ROLE-ID` command to set roles for members of an organization
- Defining custom IAM roles:
 - Use the `gcloud iam roles create` command to create a custom role
 - The role settings can be specified in a YAML file or with CLI flags

5.2 Managing Service Accounts

- Managing service accounts with limited scopes:
 - Use the `gcloud iam service-accounts create SA-NAME --display-name "SA-DISPLAY-NAME"` command to create a service account
 - Use the `gcloud iam service-accounts list` command to list services accounts
 - Use the `gcloud iam service-accounts delete SA-NAME@PROJECT-ID.iam.gserviceaccount.com` command to delete a service account
- Assigning a service account to VM instances:
 - Use the `gcloud compute instances set-service-account INSTANCE_NAME` command to set a service account for an instance

- Granting access to a service account in another project:
 - Once you have a service account in one project, you can add the email address for that account as an IAM member in another project
 - Once the service account is set up as a member of another project, they can be granted roles like any other user

5.3 Viewing Audit Logs for Project and Managed Services

Viewing audit logs:

- Audit logs provide 3 forms of logs:
 - System events
 - Admin activity
 - Data access
- Each can be viewed using the activity tab on the dashboard or the log viewer
- You can also use the API or Cloud SDK to access the data programmatically