# GCP Major Commands for CLI
# (By Kavitha B)

List active account name command:

**gcloud auth list**

List project ID

**gcloud config list project**

RDP :

gcloud compute instances get-serial-port-output instance-1

Uid pwd setting

gcloud compute reset-windows-password [instance] --zone us-east1-b --user [username]

gcloud compute reset-windows-password instance-1 --zone us-east1-b --user admin

1. Set the region to `us-east1`

   ```
   gcloud config set compute/region us-east1
   ```

   Copied!

   content_copy

2. To view the project region setting, run the following command:

   ```
   gcloud config get-value compute/region
   ```

   Copied!

   content_copy

3. Set the zone to `us-east1-b` :

   ```
   gcloud config set compute/zone us-east1-b
   ```

   Copied!

   content_copy

4. To view the project zone setting, run the following command:

   ```
   gcloud config get-value compute/zone
   ```

1. Copy your project ID to your clipboard or text editor. The project ID is listed in 2 places:

- In the Cloud Console, on the Dashboard, under **Project info**. (Click **Navigation menu** (☰), and then click **Cloud overview > Dashboard**.)
- On the lab tab near your username and password.

2. In Cloud Shell, run the following `gcloud` command, to view the project id for your project:

```
gcloud config get-value project
```

Copied!

content_copy

3. In Cloud Shell, run the following `gcloud` command to view details about the project:

```
gcloud compute project-info describe --project $(gcloud config
get-value project)
```

## Setting environment variables

Environment variables define your environment and help save time when you write scripts that contain APIs or executables.

1. Create an environment variable to store your Project ID, replacing `<your_project_ID>` with the value for *name* from the `gcloud compute project-info describe` command you ran earlier:

```
export PROJECT_ID=$(gcloud config get-value project)
```

Copied!

content_copy

2. Create an environment variable to store your Zone, replacing `<your_zone>` with the value for *zone* from the `gcloud compute project-info describe` command you ran earlier:

```
export ZONE=$(gcloud config get-value compute/zone)
```

Copied!

content_copy

3. To verify that your variables were set properly, run the following commands:

```
echo -e "PROJECT ID: $PROJECT_ID\nZONE: $ZONE"
```

Copied!
content_copy
If the variables were set correctly, the echo commands will output your Project ID and Zone.

## Creating a virtual machine with the gcloud tool

Use the `gcloud` tool to create a new virtual machine (VM) instance.

1. To create your VM, run the following command:

```
gcloud compute instances create gcelab2 --machine-type e2-medium --zone $ZONE
```

Copied!
content_copy
**Output**:

```
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-04-326fae68bc3d/zones/us-east1-c/instances/gcelab2].
NAME      ZONE            MACHINE_TYPE   PREEMPTIBLE  INTERNAL_IP
EXTERNAL_IP   STATUS
gcelab2   us-east1-b      e2-medium                   10.128.0.2
34.67.152.90  RUNNING
```

**Command details**

- `gcloud compute` allows you to manage your Compute Engine resources in a format that's simpler than the Compute Engine API.
- `instances create` creates a new instance.

- `gcelab2` is the name of the VM.
- The `--machine-type` flag specifies the machine type as *e2-medium*.
- The `--zone` flag specifies where the VM is created.
- If you omit the `--zone` flag, the `gcloud` tool can infer your desired zone based on your default properties. Other required instance settings, such as `machine type` and `image`, are set to default values if not specified in the `create` command.

## Creating a virtual machine with the gcloud tool

Use the `gcloud` tool to create a new virtual machine (VM) instance.

1. To create your VM, run the following command:

```
gcloud compute instances create gcelab2 --machine-type e2-medium -
-zone $ZONE
```

- To open help for the `create` command, run the following command:

## gcloud compute instances create –help

## Explore gcloud commands

The `gcloud` tool offers simple usage guidelines that are available by adding the `–h` flag (for help) onto the end of any `gcloud` command.

1. Run the following command:

```
gcloud -h
```

Copied!

content_copy

You can access more verbose help by appending the `--help` flag onto a command or running the `gcloud help` command.

2. Run the following command:

   ```
   gcloud config --help
   ```

   Copied!

   content_copy

3. Run the following command:

   ```
   gcloud help config
   ```

   Copied!

   content_copy

   The results of the `gcloud config --help` and `gcloud help config` commands are equivalent. Both return long, detailed help.

   `gcloud` Global Flags govern the behavior of commands on a per-invocation level. Flags override any values set in SDK properties.

4. View the list of configurations in your environment:

   ```
   gcloud config list
   ```

   Copied!

   content_copy

5. To see all properties and their settings:

   ```
   gcloud config list --all
   ```

   Copied!

   content_copy

6. List your components:

```
gcloud components list
```

Copied!

content_copy

This command displays the gcloud components that are ready for you to use in this lab.

# Task 2. Filtering command line output

The gcloud CLI is a powerful tool for working at the command line. You may want specific information to be displayed.

1. List the compute instance available in the project:

```
gcloud compute instances list
```

List the gcelab2 virtual machine:

```
gcloud compute instances list --filter="name=('gcelab2')"
```

In the above command we have asked gcloud to only show the information matching the criteria i.e. a virtual instance name matching the criteria.

List the Firewall rules in the project:

```
gcloud compute firewall-rules list
```

List the Firewall rules for the default network:

```
gcloud compute firewall-rules list --filter="network='default'"
```

List the Firewall rules for the default network where the allow rule matches an ICMP rule:

```
gcloud compute firewall-rules list --filter="NETWORK:'default' AND
ALLOW:'icmp'"
```

# Task 3. Connecting to your VM instance

`gcloud compute` makes connecting to your instances easy. The `gcloud compute ssh` command provides a wrapper around SSH, which takes care of authentication and the mapping of instance names to IP addresses.

1. To connect to your VM with SSH, run the following command:

   ```
   gcloud compute ssh gcelab2 --zone $ZONE
   ```

# Task 3. Connecting to your VM instance

`gcloud compute` makes connecting to your instances easy. The `gcloud compute ssh` command provides a wrapper around SSH, which takes care of authentication and the mapping of instance names to IP addresses.

1. To connect to your VM with SSH, run the following command:

   ```
   gcloud compute ssh gcelab2 --zone $ZONE
   ```

1. To continue, type **Y**.

1. To leave the passphrase empty, press **ENTER** twice.

   **Note:** You have connected to the virtual machine created earlier in the lab. Did you notice how the command prompt changed?

   The prompt now says something similar to **sa_107021519685252337470@gcelab2**.

   - The reference before the @ indicates the account being used.
   - After the @ sign indicates the host machine being accessed.

2. Install `nginx` web server on to virtual machine:

   ```
   sudo apt install -y nginx
   ```

   Copied!

   content_copy

3. You don't need to do anything here, so to disconnect from SSH and exit the remote shell, run the following command:

   ```
   exit
   ```

# Task 4. Updating the Firewall

When using compute resources such as virtual machines, it's important to understand the associated firewall rules.

1. List the firewall rules for the project:

   ```
   gcloud compute firewall-rules list
   ```
1. From the above we can see we have two networks available.
   The `default` network is where our virtual machine `gcelab2` is located.

2. Try to access the nginx service running on the `gcelab2` virtual machine.

   **Note:** Communication with the virtual machine will fail as it does not have an appropriate firewall rule. Our **nginx** web server is expecting to communicate on **tcp:80**. To get communication working we need to:

   - Add a tag to the gcelab2 virtual machine
   - Add a firewall rule for http traffic
3. Add a tag to the virtual machine:

   ```
   gcloud compute instances add-tags gcelab2 --tags http-server,https-server
   ```

   Copied!

   content_copy

4. Update the firewall rule to allow:

```
gcloud compute firewall-rules create default-allow-http --
direction=INGRESS --priority=1000 --network=default --action=ALLOW
--rules=tcp:80 --source-ranges=0.0.0.0/0 --target-tags=http-server
```

Copied!

content_copy

5. List the firewall rules for the project:

```
gcloud compute firewall-rules list --filter=ALLOW:'80'
```

1. Verify communication is possible for http to the virtual machine:

```
curl http://$(gcloud compute instances list --filter=name:gcelab2
--format='value(EXTERNAL_IP)')
```

Copied!

content_copy

You will see the default `nginx` output.

# ask 5. Viewing the system logs

Viewing logs is essential to understanding the working of your project. Use gcloud to access the different logs available on Google Cloud.

1. View the available logs on the system:

```
gcloud logging logs list
```

View the logs that relate to compute resources:

```
gcloud logging logs list --filter="compute"
```

1. Read the logs related to the resource type of `gce_instance`:

   ```
   gcloud logging read "resource.type=gce_instance" --limit 5
   ```

   Copied!

   content_copy

2. Read the logs for a specific virtual machine:

   ```
   gcloud logging read "resource.type=gce_instance AND
   labels.instance_name='gcelab2'" --limit 5
   ```