

Assignment 1, Cloud Application Development

Put all deliverables into github repository in your profile. Share link to google form 24 hours before defense. Defend by explaining deliverables and answering questions.

Deliverables: report in pdf

Google form:

https://docs.google.com/forms/d/e/1FAIpQLSe0GyNdOYlvM1tX_I_CtlPod5jBf-ACLGdHYZq1gVZbUeBzlg/viewform?usp=sf_link

Exercise 1: Setting Up Google Cloud SDK

1. **Objective:** Install and configure the Google Cloud SDK on your local machine.
2. **Steps:**
 - Visit the Google Cloud SDK installation page.
 - Follow the instructions to download and install the SDK for your operating system.
 - After installation, run `gcloud init` to initialize the SDK and authenticate with your Google account.
 - Configure the default project and region.
 - Verify the installation by running `gcloud version` and `gcloud info`.
3. **Questions:**
 - What command did you use to authenticate with your Google account?

gcloud auth login

```
Do you want to continue (Y/n)? y

Go to the following link in your browser, and complete the sign-in prompts:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fsdk.cloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=TcOSLUPokxBWU5A4iXoKMUEMgc5uH6&prompt=consent&token_usage=remote&access_type=offline&code_challenge=4e-18bMU2IShK01m-YfSGSUY8y5Cb7Tqz3YdeoOvqLo&code_challenge_method=S256

Once finished, enter the verification code provided in your browser: 4/0AQ1Ed8zvyha9pf92T9fjDBrg21J98XZTl1dbslpp7EyflpTm3JTVijjYpYrHJ6HI_IywQ

You are now logged in as [k.sanzhq@gmail.com].
Your current project is [jovial-meridian-435717-r2]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
k_sanzhq@cloudshell:~ (jovial-meridian-435717-r2)$
```

- How did you set the default project?

gcloud config set project lab-work-1

```

gcloud projects set-iam-policy
gcloud config configurations activate
gcloud config configurations create
gcloud config configurations delete
gcloud config configurations describe
gcloud config configurations list
gcloud config configurations rename
gcloud resource-settings set-value

To search the help text of gcloud commands, run:
gcloud help -- SEARCH_TERMS
k_sanzhq@cloudshell:~ (jovial-meridian-435717-r2)$ gcloud config set project lab-work-1
WARNING: [k.sanzhq@gmail.com] does not have permission to access projects instance [lab-work-1]
(or it may not exist): The caller does not have permission. This command is authenticated as k
.sanzhq@gmail.com which is the active account specified by the [core/account] property
Are you sure you wish to set property [core/project] to lab-work-1?

Do you want to continue (Y/n)? y

Updated property [core/project].
k_sanzhq@cloudshell:~ (lab-work-1)$

```

- What information does the `gcloud info` command provide?

```

k_sanzhq@cloudshell:~ (lab-work-1)$ gcloud info
Google Cloud SDK [489.0.0]

Platform: [Linux, x86_64] uname_result(system='Linux', node='cs-267978145750-default', release='6.1.100+', version='#1 SMP PREEMPT_DYNAMIC Sat Aug 17 14:12:
26 UTC 2024', machine='x86_64')
Locale: ('en_US', 'UTF-8')
Python Version: [3.11.9 (main, Jul 27 2024, 03:07:42) [Clang 18.1.8 ]]
Python Location: [/usr/lib/google-cloud-sdk/platform/bundledpythonunix/bin/python3]
OpenSSL: [OpenSSL 3.0.14 4 Jun 2024]
Requests Version: [2.25.1]
urllib3 Version: [1.26.9]
Default CA certs file: [/usr/bin/./lib/google-cloud-sdk/lib/third_party/certifi/cacert.pem]
Site Packages: [Enabled]

Installation Root: [/usr/lib/google-cloud-sdk]
Installed Components:
  app-engine-python: [1.9.113]
  local-extract: [1.5.10]
  gcloud-crc32c: [1.0.0]
  alpha: [2024.08.16]
  package-go-module: [0.4.0]
  cloud-datastore-emulator: [2.3.1]
  skaffold: [2.13.1]
  gke-gcloud-auth-plugin: [0.5.9]
  app-engine-go: [1.9.76]
  bigtable: []
  kpt: [1.0.0-beta.50]
  gsutil: [5.30]
  cbt: [1.21.0]
  bundled-python3-unix: [3.11.9]
  app-engine-java: [2.0.29]
  minikube: [1.33.1]

```

Exercise 2: Exploring Cloud Shell

1. **Objective:** Familiarize yourself with the Google Cloud Shell environment.
2. **Steps:**
 - Open the Google Cloud Console and activate Cloud Shell.
 - Explore the environment by listing files and checking the available tools.
 - Run the command `gcloud config list` to see your current configuration.
 - Create a directory named `gcp-intro` and navigate into it.
 - Use the built-in code editor to create a simple `README.md` file describing your GCP project.
3. **Questions:**
 - What is the default home directory in Cloud Shell?

/home/k_sanzhq
 - What tools are pre-installed in Cloud Shell ?

gcloud CLI, bq, kubectl, docker, git, vim, emacs, nano, Python, pip, Node.js, npm, Go, Java, Ruby, PHP, MySQL client, PostgreSQL client, Terraform, Ansible, Maven, Gradle, OpenJDK, Cloud SQL Proxy, Helm, jq

- How can you open the built-in code editor in Cloud Shell?

We can click on the pencil icon (Open Editor) located in the topright corner of the Cloud Shell window

```
k_sanzhq@cloudshell:~ (lab-work-1)$ gcloud config list
[accessibility]
screen_reader = True
[component_manager]
disable_update_check = True
[compute]
gce_metadata_read_timeout_sec = 30
[core]
account = k.sanzhq@gmail.com
disable_usage_reporting = False
project = lab-work-1
[metrics]
environment = devshell

Your active configuration is: [cloudshell-23586]
k_sanzhq@cloudshell:~ (lab-work-1)$ mkdir gcp-intro
k_sanzhq@cloudshell:~ (lab-work-1)$ cd gcp_intro
-bash: cd: gcp_intro: No such file or directory
k_sanzhq@cloudshell:~ (lab-work-1)$ cd gcp-intro
k_sanzhq@cloudshell:~/gcp-intro (lab-work-1)$ cloudshell edit
usage: cloudshell edit-files [-h] filename [filename ...]
cloudshell edit-files: error: the following arguments are required: filename
k_sanzhq@cloudshell:~/gcp-intro (lab-work-1)$
```

Exercise 3: Managing Projects with Google Cloud SDK

1. **Objective:** Use Google Cloud SDK to manage projects.
2. **Steps:**
 - List all the projects associated with your Google account using `gcloud projects list`.
 - Create a new project with the command `gcloud projects create PROJECT_ID --name="My First GCP Project"`.
 - Set this new project as your default project.
 - Explore project metadata using `gcloud projects describe PROJECT_ID`.
 - Delete the project using `gcloud projects delete PROJECT_ID` after completing the exercise.

3. **Questions:**

- How do you list all projects associated with your account?

`gcloud projects list`

- What command is used to set a default project?

`gcloud config set project [project-id]`

- How do you describe project metadata?

gcloud projects describe [project-id]

Exercise 4: Using Cloud Shell for Basic Operations

1. **Objective:** Perform basic file and directory operations in Cloud Shell.

2. **Steps:**

- In Cloud Shell, create a directory structure that mimics a small project (e.g., `myproject/src`, `myproject/tests`, `myproject/docs`).
- Create a few files in these directories and use commands like `touch`, `nano`, `cat`, and `rm` to manipulate them.
- Write how to use `gsutil` to create a new Cloud Storage bucket and upload a file from your Cloud Shell environment.
- Verify the file upload by listing the contents of the bucket.

3. **Questions:**

- What command did you use to create the directory structure?

`mkdir cmd && mkdir internal && mkdir pkg`

- How did you upload a file to a Cloud Storage bucket?

`gsutil cp main.txt gs://my-bucket-lab-1`

- How can you list the contents of a Cloud Storage bucket?

`gsutil ls gs://my-bucket-lab-1`

Exercise 5: Automating Tasks with Shell Scripts in Cloud Shell

1. **Objective:** Write and execute a basic shell script in Cloud Shell.

2. **Steps:**

- In Cloud Shell, create a new shell script named `setup.sh` in your `gcp-intro` directory.
- The script should automate the creation of a new directory, a simple text file, and set up a basic Google Cloud configuration (e.g., set a default project).
- Make the script executable using `chmod +x setup.sh`.
- Run the script and verify that it performs the expected tasks.

3. **Questions:**

- What command did you use to make the script executable?

`chmod +x setup.sh`

- How did you ensure the script was executed correctly?

Checked if the new directory was created correctly and checked the contents of this new directory

- What steps did your script automate?
 - 1) Created a new directory with the command: `mkdir -p new_directory`
 - 2) Created a new text file with the following content: `echo "This is a sample file created by setup.sh" > new_directory/sample.txt`
 - 3) Set the default project: `gcloud config set project laboratory-work-1-435706`
 - 4) Indicated that the script has finished: `echo "Setup complete!"`