

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?

Answer: I used the **gcloud auth login** command.

- How did you set the default project?

Answer: I used the **gcloud config set project [PROJECT_ID]** command to set the default project and the **gcloud config set compute/region [REGION]** command to set default region.

- What information does the gcloud info command provide?

Answer: This command provides info about my OS, python version, location and path, system path, git version. Also about my account, default project and many other info.

```
C:\Users\azikkw>gcloud info
Google Cloud SDK [492.0.0]

Platform: [Windows, x86_64] uname_result(system='Windows', node='DESKTOP-T1GP8V3', release='10', version='10.0.22631', machine='AMD64')
Locale: ('Russian_Kazakhstan', '1251')
Python Version: [3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)]]
Python Location: [C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk\platform\bundledpython\python.exe]
OpenSSL: [OpenSSL 3.0.13 30 Jan 2024]
Requests Version: [2.25.1]
urllib3 Version: [1.26.9]
Default CA certs file: [C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk\bin\..\lib\third_party\certifi\cacert.pem]
Site Packages: [Disabled]

Installation Root: [C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk]
Installed Components:
  bq: [2.1.8]
  core: [2024.09.06]
  gcloud-crc32c: [1.0.0]
  gsutil: [5.30]
System PATH: [C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk\bin\..\bin;sdk;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\WIN
DOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Users\azikkw\AppData\Local\Pro
grams\Flutter\bin;C:\MinGW\bin;C:\Program Files\nodejs\;C:\Program Files\Docker\resources\bin;C:\Program Files\Git\cmd;C:\Program Files\Git\bin;C:\Pr
ogram Files (x86)\GnuWin32\bin;C:\Users\azikkw\AppData\Local\Programs\Python\Python310\Scripts\;C:\Users\azikkw\AppData\Local\Programs\Python\Python310\;C:\
Users\azikkw\AppData\Local\Microsoft\WindowsApps;C:\Users\azikkw\AppData\Local\Programs\Microsoft VS Code\bin;C:\Users\azikkw\AppData\Local\Programs\Flutter
\bin;C:\MinGW\bin;C:\Users\azikkw\AppData\Roaming\npm;C:\Program Files\JetBrains\WebStorm 2024.1.5\bin;C:\Users\azikkw\AppData\Local\Google\Cloud SDK\googl
e-cloud-sdk\bin]
Python PATH: [C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk\bin\..\lib\third_party;C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-c
loud-sdk\bin\..\lib;C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk\lib;C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk\plat
form\bundledpython\python311.zip;C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk\platform\bundledpython\DLLs;C:\Users\azikkw\AppData\Local\G
oogle\Cloud SDK\google-cloud-sdk\platform\bundledpython\Lib;C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk\platform\bundledpython]
Cloud SDK on PATH: [True]
Kubectl on PATH: [False]

Installation Properties: [C:\Users\azikkw\AppData\Local\Google\Cloud SDK\google-cloud-sdk\properties]
User Config Directory: [C:\Users\azikkw\AppData\Roaming\gcloud]
Active Configuration Name: [default]
Active Configuration Path: [C:\Users\azikkw\AppData\Roaming\gcloud\configurations\config_default]

Account: [azat.amenov@gmail.com]
Project: [my-first-gcpproject-2024]
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры".

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?
Answer: `/home/[USER_NAME]` directory.
- What tools are pre-installed in Cloud Shell?
Answer: `gcloud` CLI and tools; linux shell interpreters - `bash`, `sh`; version control tools: `git`; also - `docker`, `python`, `java`, `node.js`, etc.
- How can you open the built-in code editor in Cloud Shell?
Answer: Just clicked to the Open Editor button. But also we can use `cloudshell edit [FILE_PATH]`.

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?
Answer: using `gcloud projects list` command.
- What command is used to set a default project?
Answer: `gcloud config set project [PROJECT_ID]` command.
- How do you describe project metadata?

Answer: I used `gcloud projects describe [PROJECT_ID]` command to view metadata. On this image showed what describes project metadata:

```
C:\Users\azikkw>gcloud projects describe eventy-c5909
createTime: '2024-07-22T11:20:59.320543Z'
labels:
  firebase: enabled
lifecycleState: ACTIVE
name: eventy
projectId: eventy-c5909
projectNumber: '851322724573'
```

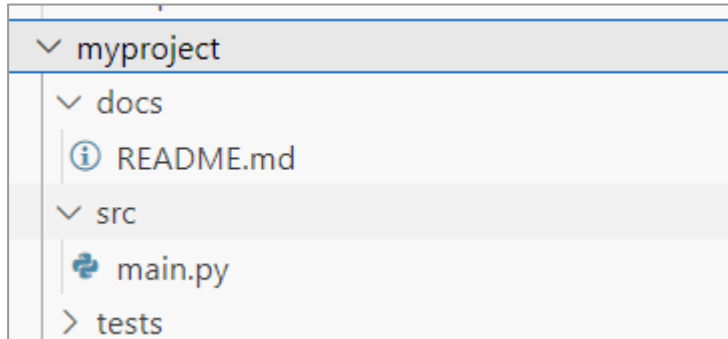
metadata consist of *createTime*, *labels*, *lifecycleState*, *name*, *id* and *number*.

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.



- 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?

Answer: using `mkdir -p myproject/src myproject/tests myproject/docs` command.

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

Answer: using `gsutil cp [FILE_PATH] gs://[BUCKET_NAME]` command.

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

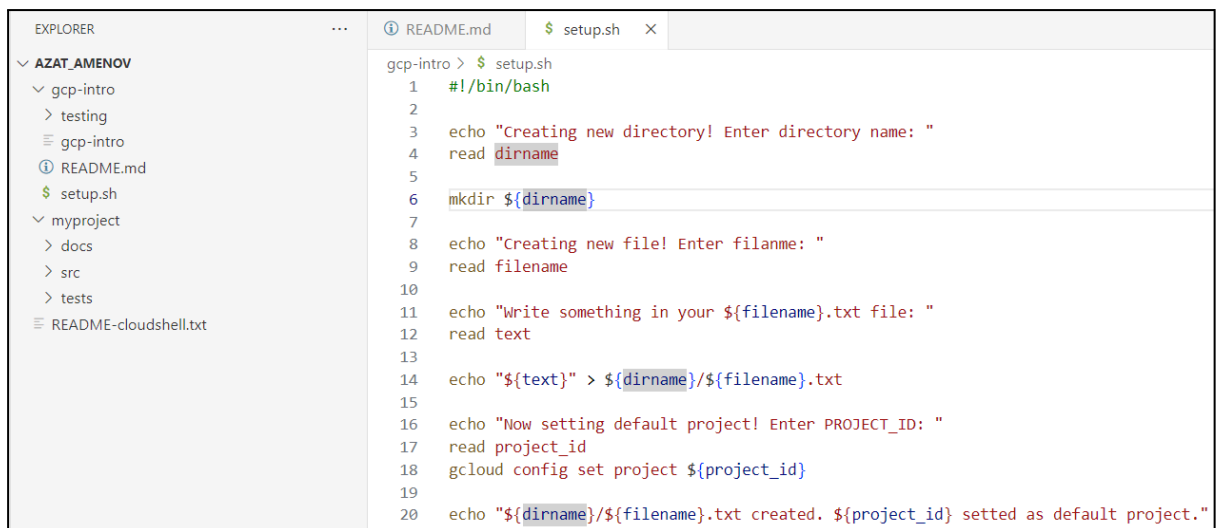
Answer: using `gsutil ls gs://[BUCKET_NAME]` command.

Exercise 5: Automating Tasks with Shell Scripts in Cloud Shell

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

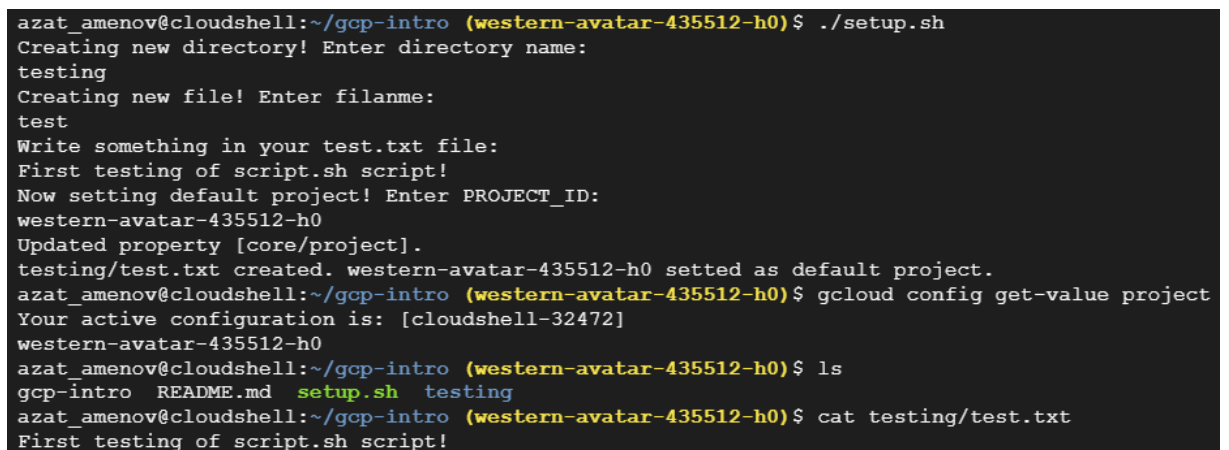
2. **Steps:**

- 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.



The screenshot shows the Cloud Shell Explorer interface. On the left, the file explorer shows the directory structure: AZAT_AMENOV > gcp-intro > testing > gcp-intro > README.md, setup.sh, myproject > docs, src, tests, and README-cloudshell.txt. The main editor shows the content of setup.sh, which is a shell script with 20 lines. The script starts with a shebang, prompts for a directory name, creates it, prompts for a filename, creates it, prompts for text to write to it, sets the default project, and finally confirms the creation of the file and the project setting.

```
gcp-intro > $ setup.sh
1  #!/bin/bash
2
3  echo "Creating new directory! Enter directory name: "
4  read dirname
5
6  mkdir ${dirname}
7
8  echo "Creating new file! Enter filename: "
9  read filename
10
11 echo "Write something in your ${filename}.txt file: "
12 read text
13
14 echo "${text}" > ${dirname}/${filename}.txt
15
16 echo "Now setting default project! Enter PROJECT_ID: "
17 read project_id
18 gcloud config set project ${project_id}
19
20 echo "${dirname}/${filename}.txt created. ${project_id} setted as default project."
```



The screenshot shows the Cloud Shell terminal output of running the setup.sh script. The user runs './setup.sh' in the ~/gcp-intro directory. The script prompts for a directory name ('testing'), a filename ('test'), and text to write to the file ('First testing of script.sh script!'). It then prompts for a project ID ('western-avatar-435512-h0') and sets it as the default project. Finally, it confirms the creation of the file and the project setting. The user then runs 'ls' to show the directory contents and 'cat testing/test.txt' to view the file content.

```
azat_amenov@cloudshell:~/gcp-intro (western-avatar-435512-h0)$ ./setup.sh
Creating new directory! Enter directory name:
testing
Creating new file! Enter filename:
test
Write something in your test.txt file:
First testing of script.sh script!
Now setting default project! Enter PROJECT_ID:
western-avatar-435512-h0
Updated property [core/project].
testing/test.txt created. western-avatar-435512-h0 setted as default project.
azat_amenov@cloudshell:~/gcp-intro (western-avatar-435512-h0)$ gcloud config get-value project
Your active configuration is: [cloudshell-32472]
western-avatar-435512-h0
azat_amenov@cloudshell:~/gcp-intro (western-avatar-435512-h0)$ ls
gcp-intro  README.md  setup.sh  testing
azat_amenov@cloudshell:~/gcp-intro (western-avatar-435512-h0)$ cat testing/test.txt
First testing of script.sh script!
```

3. **Questions:**

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

Answer: using `chmod +x setup.sh` command.

- How did you ensure the script was executed correctly?

Answer: by running it using `./setup.sh`, checking that created directory, file exist and viewed default project.

- What steps did your script automate?

Answer: my script creates a directory, file and set default project.