**Assignment 1. Cloud Application Development**

**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.

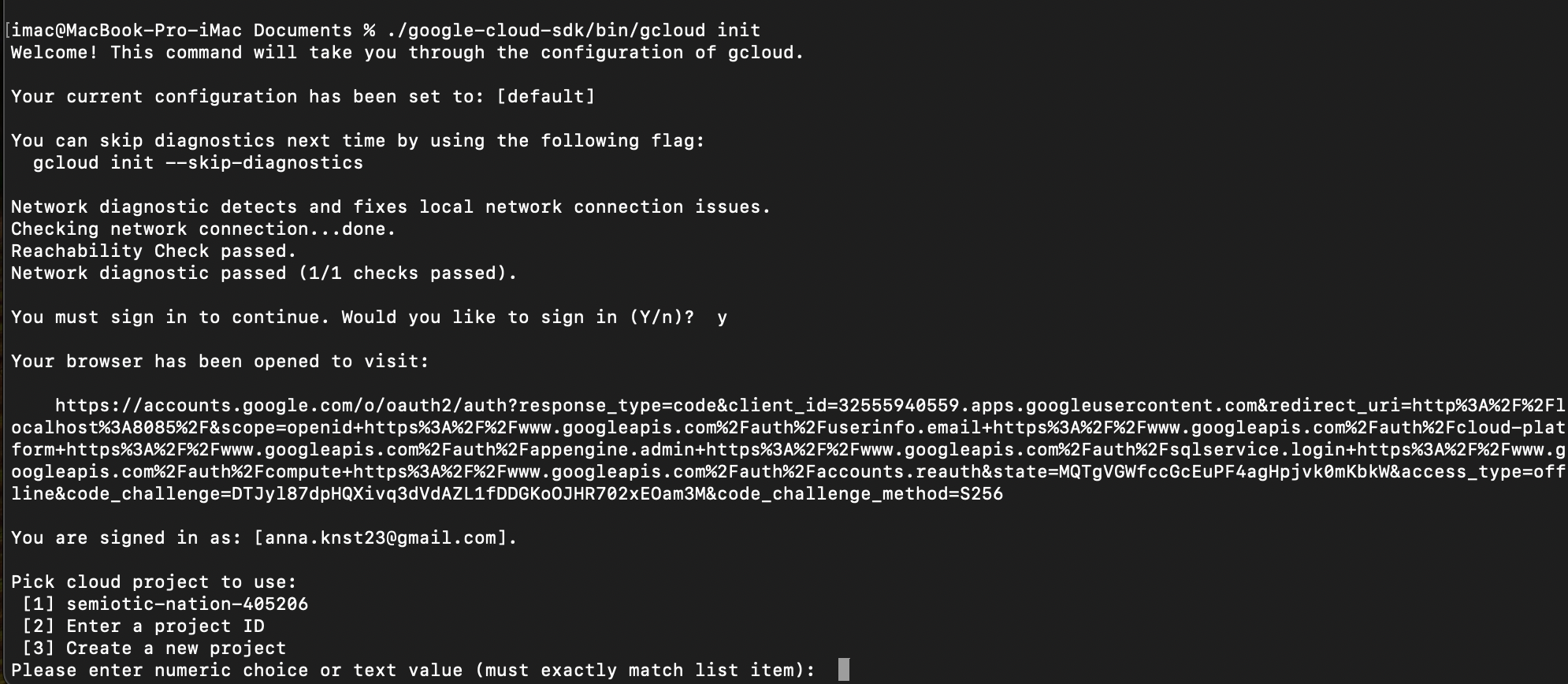
○ Configure the default project and region.

○ Verify the installation by running `gcloud version`and `gcloud info`.

3. Questions:

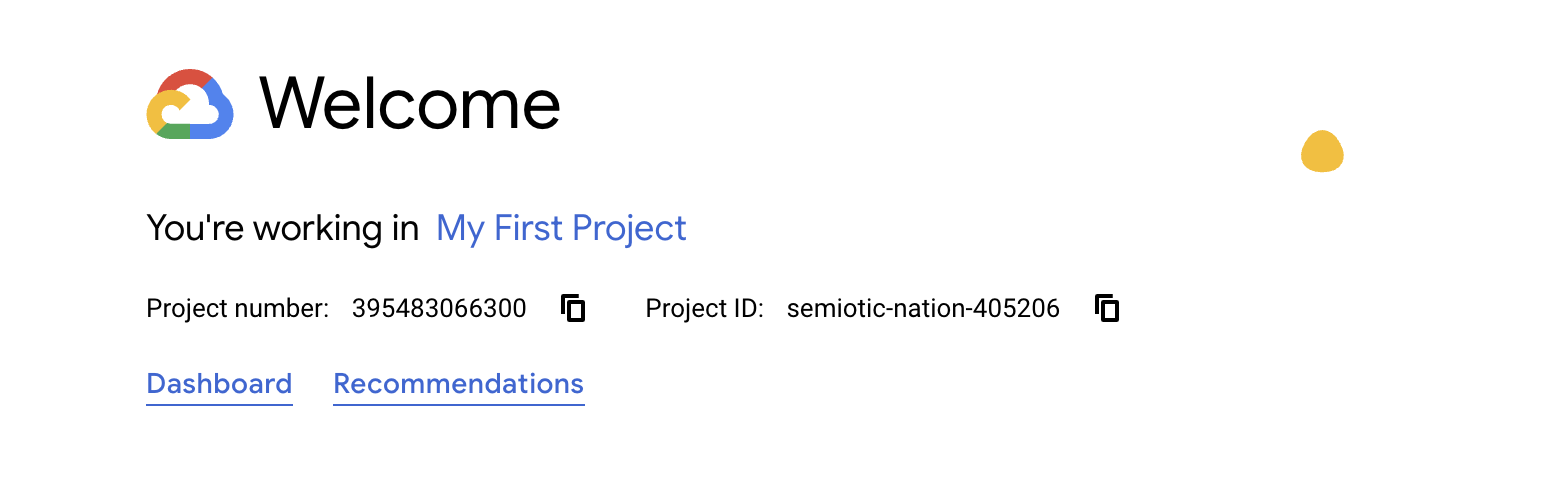
○ What command did you use to authenticate with your Google account?

- The command used to authenticate with your Google account is: `gcloud init`



○ How did you set the default project?

- The default project is set during the `gcloud init` process when prompted.



○ What information does the gcloud info command provide?

- The `gcloud info` command provides detailed information about your Google Cloud SDK installation and environment, including:

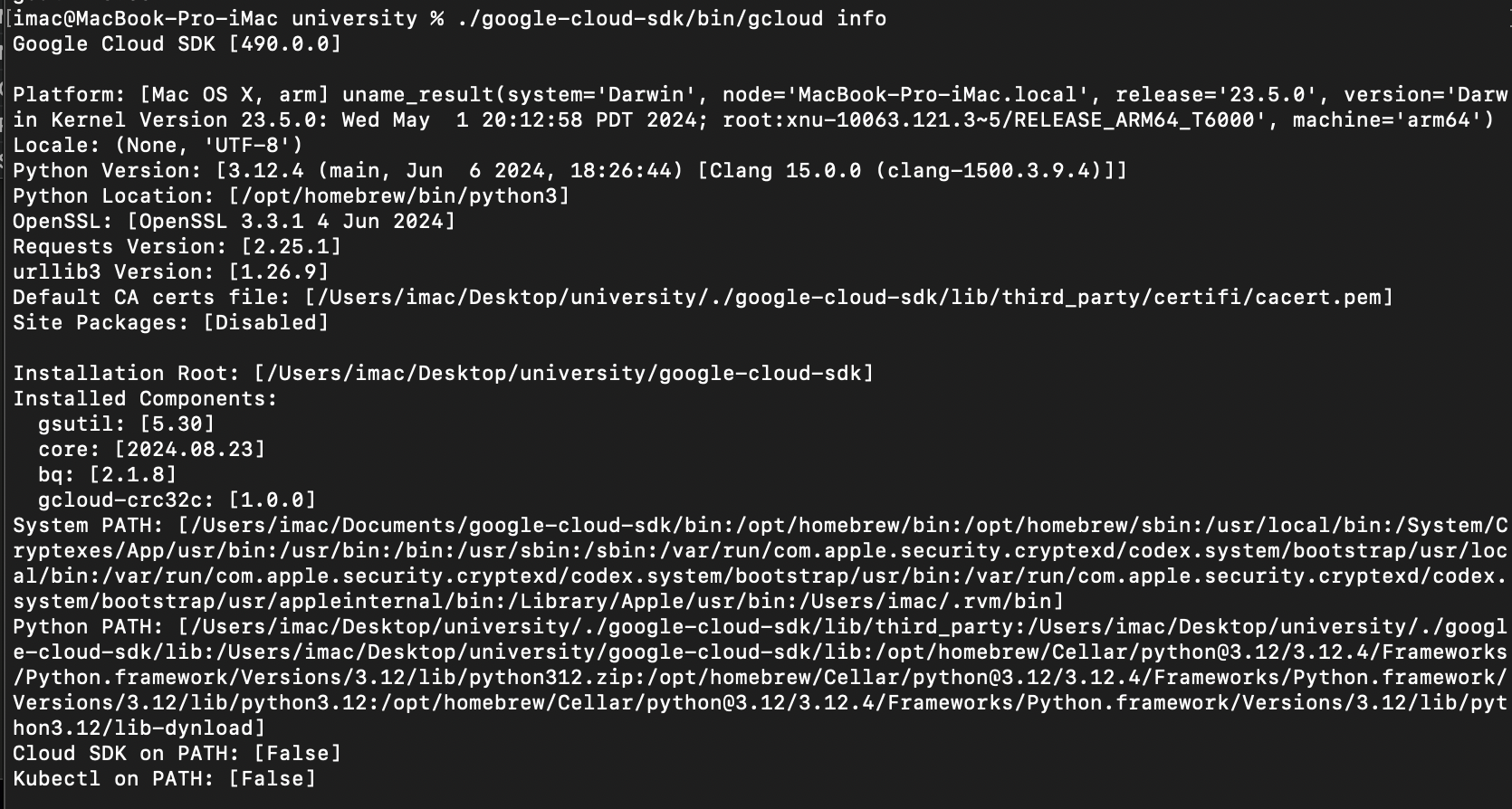
1.The version of the `gcloud` command-line tool.

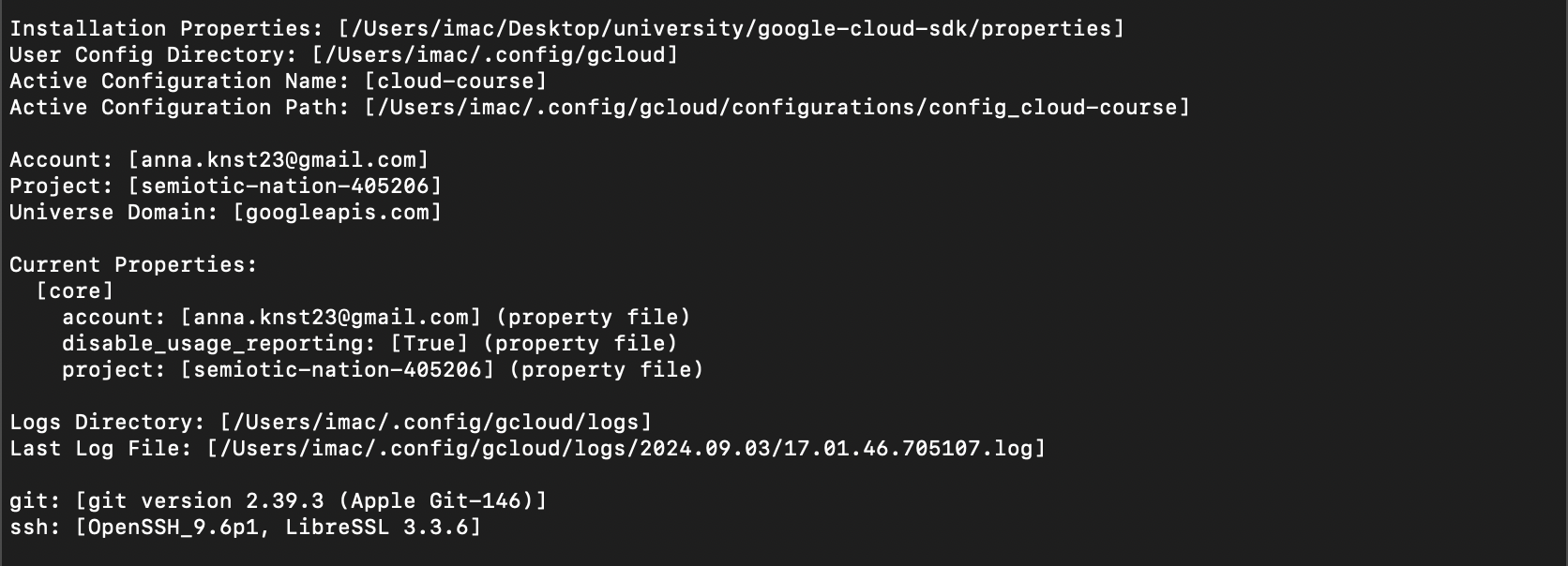
2.Installed components and their versions.

3.Configuration details such as the active project and account.

4.The current configuration and settings for the SDK.

5. Environment details, such as the operating system and Python version used by the SDK.





**Exercise 2: Exploring Cloud Shell**

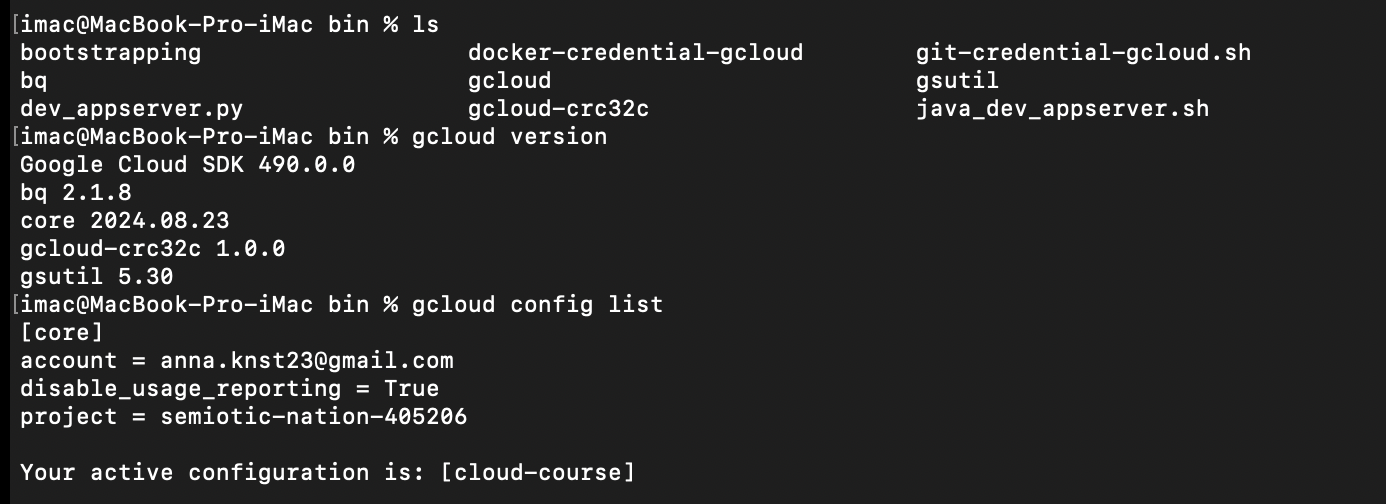
1. Objective: Familiarize yourself with the Google Cloud Shell environment.

2. Steps:

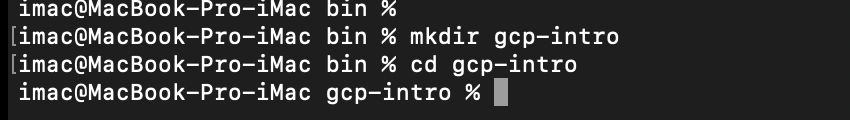
○ Open the Google Cloud Console and activate Cloud Shell.

○ Explore the environment by listing files and checking the available tools.

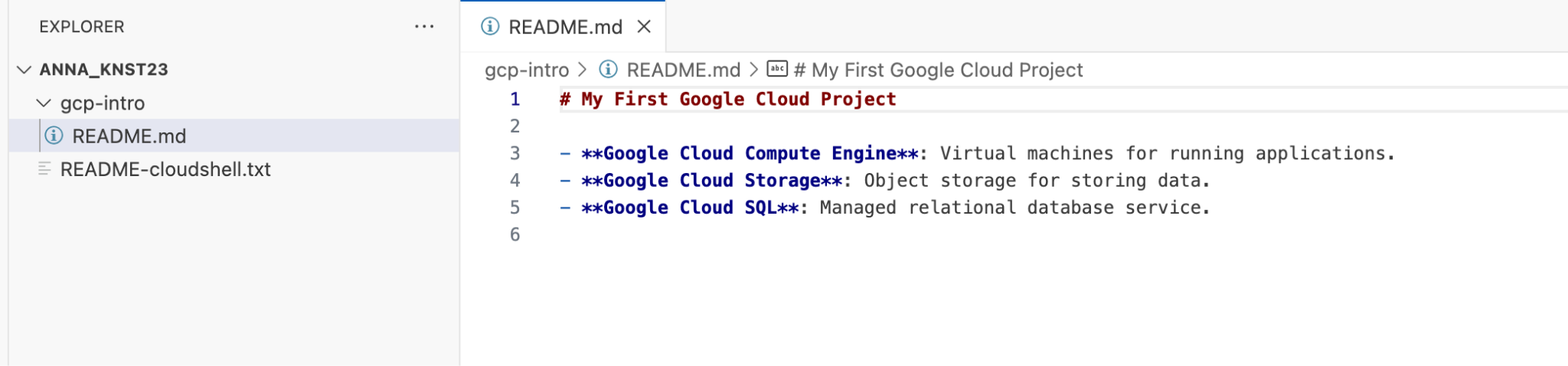
○ Run the command gcloud config list to see your current configuration.



○ 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?

* The default home directory in Cloud Shell is /home/<your-username>. When Cloud Shell will be opened in the first time, user is placed in this directory.

○ What tools are pre-installed in Cloud Shell?

* When you launch Cloud Shell, it automatically creates a virtual machine (VM) pre-installed with cloud management and system tools, including Python, Java, Node. js, SSH, version control system, and so on.

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

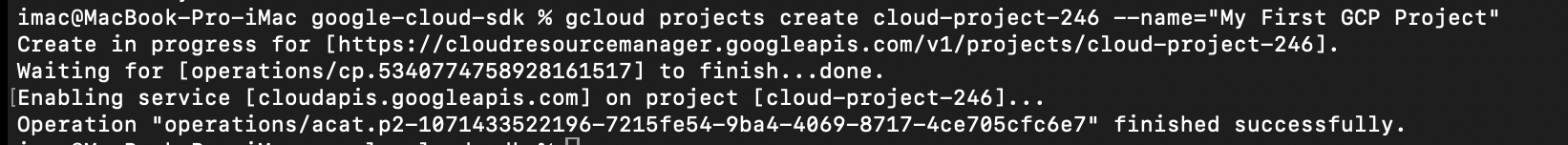
* To open the built-in code editor in Cloud Shell, click the **Editor** icon on the right side of the Cloud Shell window (it looks like a pencil). This opens a graphical code editor where you can create and edit files.

**Exercise 3: Managing Projects with Google Cloud SDK**

1. Objective: Use Google Cloud SDK to manage projects.

2. Steps:

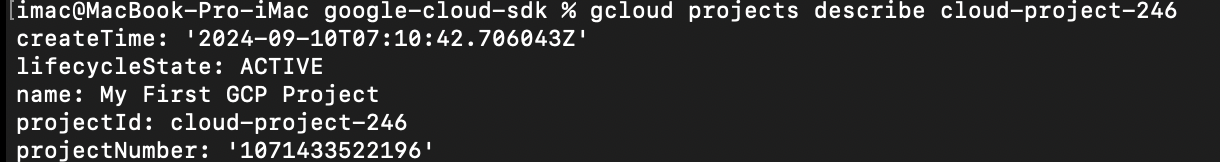
○ List all the projects associated with your Google account using gcloud projects list.

○ Create a new project with the command gcloud projects create PROJECT\_ID --name="My First GCP Project". 

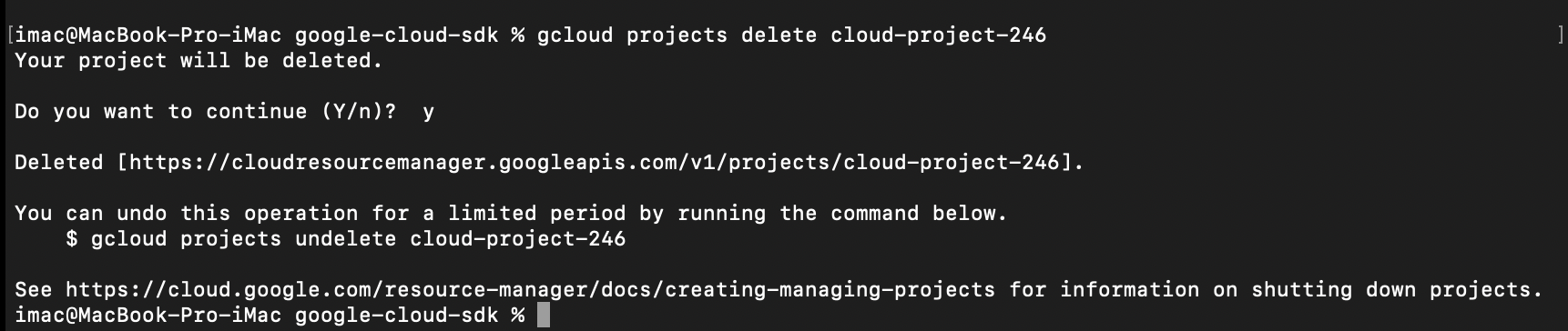
○ Set this new project as your default project.



○ 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?

* To list all projects associated with your Google account, should be used: gcloud projects list

○ What command is used to set a default project?

* To set a default project, should be used: gcloud config set project PROJECT\_ID

○ How do you describe project metadata?

* To describe project metadata, should be used: gcloud projects describe PROJECT\_ID

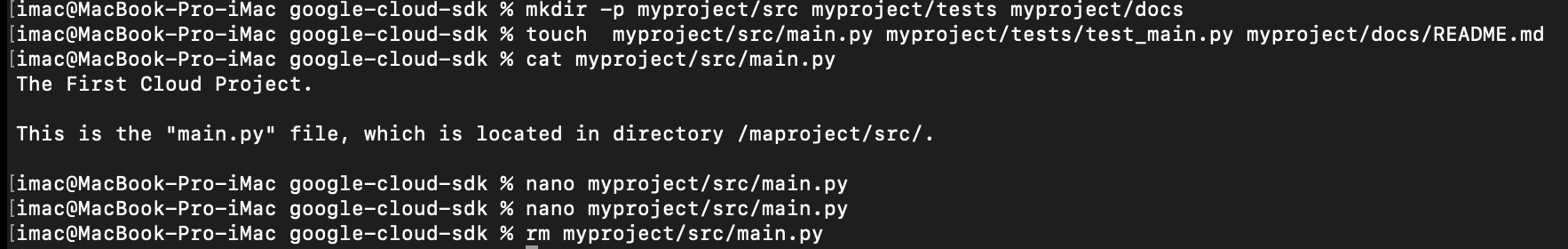
**Exercise 4: Using Cloud Shell for Basic Operations**

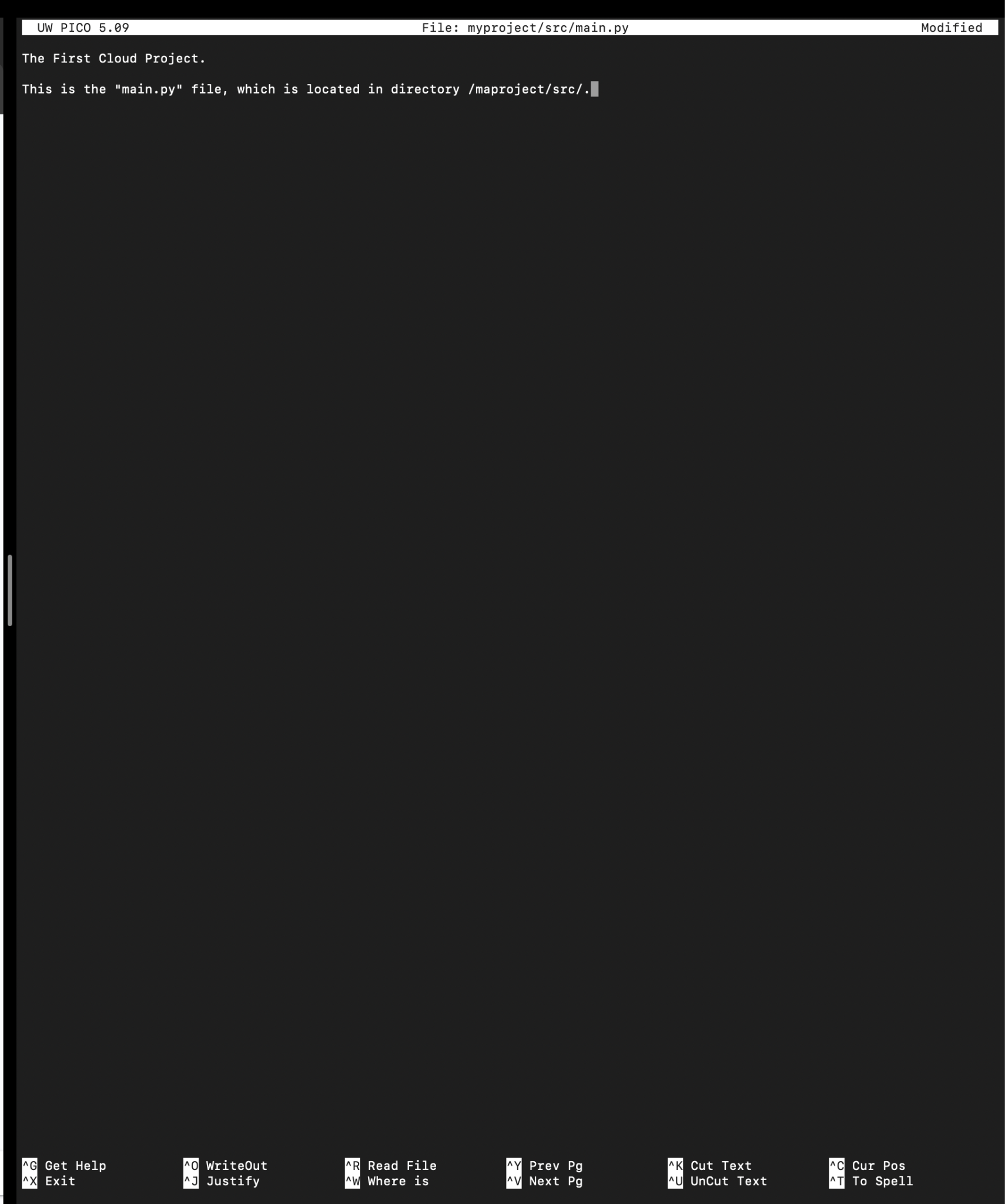
1. Objective: Perform basic file and directory operations in Cloud Shell.

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.

1. Create a New Cloud Storage Bucket: To create a new Cloud Storage bucket, it`s needed to use the gsutil mb (make bucket) command. The syntax is:

`gsutil mb -l [LOCATION] gs://[BUCKET\_NAME]`

-`[LOCATION]`will be replaced with the region or multi-region where the user wants he`s bucket to be (e.g., `us-central1`, `us-east`).

- `[BUCKET\_NAME]`will be replaced with a unique name for the user`s bucket (bucket names must be globally unique):

gsutil mb -l us-central1 gs://my-new-bucket/

2. Upload a File to the Bucket: To upload a file from your Cloud Shell environment to your newly created bucket, use the gsutil cp (copy) command:

`gsutil cp [FILE\_PATH] gs://[BUCKET\_NAME]/`

- `[FILE\_PATH]` will be replaced with the path to the file you want to upload.

- `[BUCKET\_NAME]` will be replaced with the user`s bucket name.

For example, there is a file named `example.txt` in user`s Cloud Shell:

gsutil cp example.txt gs://my-new-bucket/

This will upload the file `example.txt` to the bucket `my-new-bucket`.

○ Verify the file upload by listing the contents of the bucket.

It`s possible to verify that the file was uploaded by listing the contents of the bucket using the `gsutil ls` command:

`gsutil ls gs://[BUCKET\_NAME]/`

For example: gsutil ls gs://my-new-bucket/

This should display `example.txt` in the bucket.

3. Questions:

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

* The command used to create the directory structure is:  
  mkdir -p myproject/src myproject/tests myproject/docs. This command creates the directories myproject, myproject/src, myproject/tests, and myproject/docs.

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

* To upload a file to a Cloud Storage bucket, should be used: gsutil cp [LOCAL\_FILE\_PATH] gs://[BUCKET\_NAME]/

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

* To list the contents of a Cloud Storage bucket, should be used:  
  gsutil ls gs://[BUCKET\_NAME]/. For example:  
  gsutil ls gs://my-project-bucket-2024/. This command will list all files and directories within the specified bucket.

**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.

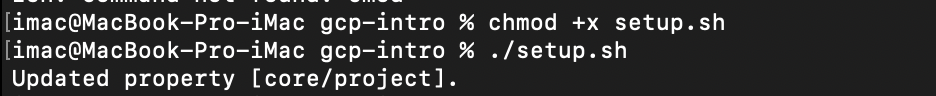
/

○ 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).

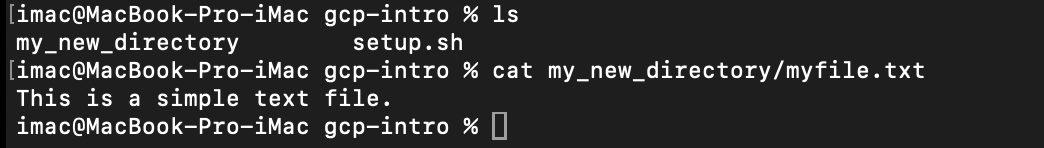


The echo command in the terminal is used to display text or variables on the screen.

○ 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?

* The command used was: chmod +x setup.sh

○ How did you ensure the script was executed correctly?

* To ensure the script was executed correctly, you verified that:
* The new directory my\_new\_directory was created.
* The text file myfile.txt contained the expected content.
* The Google Cloud configuration was updated with the correct project ID.

○ What steps did your script automate?

* The script automated the following tasks:
* Creation of a new directory named my\_new\_directory.
* Creation of a text file named myfile.txt inside the new directory with specific content.
* Setting the default Google Cloud project to the specified project ID.