**Assignment 1**

**Azhan Saleem**

**11635219**

**SUBMISSION REQUIREMENT #1: --) Provide a short paragraph to specify which operating system has been selected and to state that the student can use it effectively.**

I have chosen Windows as my preferred operating system, and I can confidently navigate its intricacies with a high level of proficiency. My knowledge encompasses the fundamental components of Windows, including drives, directories, folders, files, ownership status, and administrative privileges. I excel in basic operations, such as creating and managing directories, saving, and retrieving content, downloading, and installing software applications, and using terminals for command lines. Moreover, I am well-equipped to troubleshoot and resolve issues related to the Windows operating system and the host computing device. My research skills developed through tools like Google search, enable me to find solutions independently, and I am comfortable seeking technical support from relevant vendors and engaging in helpful discussions on technical forums. Overall, my effective use of Windows extends to both routine tasks and problem-solving scenarios, making it my OS of choice for various computing needs.

**PART II: Set Up Deep Learning Virtual Machine (VM) in GCP (20 Points)**

**Brief Report: Setting Up Remote Server on Google Cloud Platform (GCP)**

I successfully set up a remote server on Google Cloud Platform (GCP) by following the prescribed steps. The major milestones in this process included:

Google Cloud Console Login:

* Logged into the Google Cloud Console using my GCP credentials.

Compute Engine Configuration:

* Navigated to the "Compute Engine" section within the console.
* Initiated the creation of a new instance by clicking on "Create."
* Configured the VM by selecting an image containing the necessary deep learning frameworks and adjusting settings such as machine type and disk size.

VM Deployment:

* Launched the instance by clicking "Create" and patiently waited for the deployment process to complete.

**SCREENSHOT:**

A white screen with black text

Description automatically generated

**Google Cloud SDK Installation on Windows**

For this task, I installed the Google Cloud SDK on my Windows computer to enable interaction with Google Cloud Platform services.

*Here are the Steps:*

1. **Download and Run Installer:**
   * Visited the Google Cloud SDK download page for Windows.
   * Downloaded the installer executable.
   * Double-clicked the installer to run it.
2. **Configuration Choices:**
   * Selected installation location and added SDK components to the system PATH.
3. **Completion:**
   * Waited for the installation to finish.
   * Clicked "Finish" to complete the installation.
4. **Initialization:**
   * Opened the Command Prompt.
   * Ran **gcloud init** to initialize the SDK.
   * Logged in with Google Cloud credentials and configured settings.
5. **Verification:**
   * Checked successful installation with **gcloud --version** in the Command Prompt.

**SCREEN SHOT:**

**A computer screen with white text

Description automatically generatedA screenshot of a computer

Description automatically generated**

**A computer screen with white text

Description automatically generated**

**PART 3: Connect Explore Remote VM Using SSH (5 Points)**

**Introduction:**

In this session, I successfully connected to a remote virtual machine (VM) on Google Cloud Platform (GCP) and performed various tasks related to file system exploration and directory management.

1. **SSH Connection (Question 3.1)**: I initiated an SSH connection to the remote VM named tf2-keras-ann-vm using the Google Cloud SDK. The command used was:

bash

gcloud compute ssh is4920030@tf2-keras-ann-vm --project liquid-sylph-435002-u0 --zone us-west1-b

1. **Exploring the File System (Question 3.2)**: Upon establishing the SSH connection, I navigated through the file system of the remote VM. I used basic Linux commands to change to the home directory and list its contents, confirming the existing files and directories. Commands used:

bash

cd ~

ls -l

1. **Creating a Sub-Folder (Question 3.3)**: To organize my files, I created a new sub-folder named Reports under the Documents directory (which I created first) using the commands:

bash

mkdir Documents

mkdir Documents/Reports

1. **Changing the Current Directory (Question 3.4)**: After creating the Reports folder, I changed my working directory to the newly created folder:

bash

cd Documents/Reports

This allowed me to focus on tasks within the Reports directory

**SCREENSHOTS:**

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**PART 4: Start and Connect to Jupyter Notebook in Remote VM (20 Points)**

**Report: Guide to Setting Up and Using Jupyter Notebook on Remote Virtual Machine.**

**Introduction**:  
Today, I set up and utilized Jupyter Notebook on a Google Cloud remote virtual machine (VM). This report chronicles the steps I took to achieve this, aiming to harness the remote environment's computational power for data analysis and machine learning tasks.

**Step 1: Starting the Jupyter Notebook Server**:  
I began by opening an SSH terminal to connect to the remote VM. After successfully connecting, I navigated to my working directory and initiated the Jupyter Notebook server using the following command:

$ jupyter notebook --port=8888

This command started the Jupyter Notebook server on port 8888 and displayed a token required for authentication. The output included links similar to: <http://127.0.0.1:8888/?token=98cfcc554bda3ee61fe4be1a5b6e20c927ec0872f43e491f>

**Step 2: Forwarding Local Port to Remote Port**:  
Next, I opened the **Google Cloud SDK** terminal on my local machine to set up port forwarding. This step allows me to access the Jupyter Notebook interface running on the remote VM through my local browser. I used the following command to SSH into my VM and set up port forwarding from my local port 8000 to the remote VM’s port 8888:

gcloud compute ssh is4920030@tf2-keras-ann-vm --project liquid-sylph-435002-u0 --zone us-west1-b -- -L 8000:localhost:8888

This command establishes the SSH connection and forwards the local port (8000) to the remote VM’s port (8888).

**Step 3: Connecting to the Jupyter Notebook Server**:  
After setting up the infrastructure, I opened my web browser and entered the following URL in the address bar:

<http://localhost:8000>

The browser prompted me to provide a token for authentication. I copied the token displayed in the terminal output and pasted it into the prompt. The token looked something like this:

http://localhost:8888/?token=a93474e8e3aebe402a1efabd2c6d9c627250d00176705006

**Conclusion**:  
In conclusion, I successfully set up and connected to Jupyter Notebook on the remote Google Cloud VM using port forwarding. This setup allows me to efficiently use the VM’s computational resources for data analysis and machine learning projects from my local machine, providing a seamless remote working experience.

**SCREENSHOTS:**

**A screenshot of a phone

Description automatically generated**

**A computer screen with text and numbers

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**PART V: Create Service Account for GCP Project (20 Points)**

**Introduction:**

In this report, I'll outline the steps taken to create a service account for our Google Cloud Platform (GCP) project "liquid-sylph-435002-u0". This service account will enable us to securely access and manage resources within the project.

**Step 1: Accessing Google Cloud Console**

I started by opening the Google Cloud Console via my browser. In the search bar, I entered "Google Cloud Console", which directed me to the Google Cloud interface, where I could manage our GCP project.

**Step 2: Navigating to IAM & Admin Section**

After accessing the GCP Console, I selected our project: "My First Project" (Project ID: liquid-sylph-435002-u0). From the left-hand navigation menu, I clicked on "IAM & Admin" and then selected "Service Accounts." This is where the creation of a new service account began.

**Step 3: Creating the Service Account**

I clicked on the "Create Service Account" button to begin setting up the new service account. I provided a descriptive name for the account: "service-myfirstproject", which generated the following unique service account email within the project.

**Step 4: Assigning Roles**

Next, I assigned roles to the service account. Based on the specific permissions required for our project, I carefully selected the appropriate roles. In this case, I assigned the "Editor" role to grant comprehensive access to manage resources. Roles were manually configured to match the exact access needs of the service account, ensuring its functionality within the project.

**Step 5: Completing the Setup**

With the service account successfully created and roles assigned, the setup was completed. I did not create any keys at this stage, but this can be done later if needed for authentication purposes. The service account is now fully operational and ready to manage resources within our GCP project "liquid-sylph-435002-u0".

**Conclusion:**

In conclusion, by following these steps, we have successfully created and configured the service account named "service-myfirstproject" for our GCP project "liquid-sylph-435002-u0". This service account will facilitate secure access to and management of the resources within the project, allowing us to proceed with our tasks effectively.

**SCREENSHOTS:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**PART VI: Set Up Natural Language API for GCP Project (20 Points)**

**Report: Setting Up Service Account**

In this report, I will outline the steps I followed to set up a service account in Google Cloud Platform (GCP) and generate credentials for authentication. The service account was used for accessing various GCP services, particularly in a deep learning project.

**Step 1: Accessing the GCP Project Dashboard**

* Opened the Google Cloud Console by visiting the official Google Cloud Console.
* Selected the project **"My First Project"** from the top-left project selector, which corresponds to Project ID: liquid-sylph-435002-u0.
* Navigated to the project dashboard for an overview of available resources and configurations.

**Step 2: Creating the Service Account**

* Went to **IAM & Admin > Service Accounts** in the navigation menu.
* Clicked on **Create Service Account** and provided the service account name: service-acc-dl-tf-cnn.
* The service account email was generated: service-acc-dl-tf-cnn@liquid-sylph-435002-u0.iam.gserviceaccount.com.
* Continued by clicking **Create** and proceeded to the next step.

**Step 3: Granting Roles**

* After creating the service account, I assigned the necessary roles to ensure proper permissions. Specifically:
  + **Compute Admin**: To manage Compute Engine resources.
  + **Storage Admin**: To manage access to Cloud Storag e if needed.
* These roles were granted to ensure that the service account had the required access to GCP resources.

**Step 4: Generating the Key for the Service Account**

* Opened a command line terminal (CMD) on my local machine.
* Used the following command to generate a key for the service account and store it as a JSON file:

gcloud iam service-accounts keys create "C:\Users\azhan\Desktop\key.json" --iam-account [service-acc-dl-tf-cnn@liquid-sylph-435002-u0.iam.gserviceaccount.com](mailto:service-acc-dl-tf-cnn@liquid-sylph-435002-u0.iam.gserviceaccount.com) ‘

This created the key file key.json in the specified directory on my local system (Desktop).

**Step 5: Verifying the Key File and Configuration**

* Confirmed that the key file key.json was successfully created on my desktop with a unique identifier.
* This key will be used for authentication in future tasks requiring access to GCP services through the service account.

**Conclusion**

In conclusion, I successfully created and configured the service account service-acc-dl-tf-cnn for my GCP project, **"My First Project"** (Project ID: liquid-sylph-435002-u0). The key file was generated and stored locally for future use in interacting with GCP services. This setup will facilitate secure access to GCP resources and allow seamless integration with GCP services such as Compute Engine and the Natural Language API.

**SCREENSHOTS:**

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

**A screenshot of a computer screen

Description automatically generated**

**PART VII: Join a Group (10 Points)**

**Group Members: 1) Azhan Saleem**

**2) Yog Chaudhry**