**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 screenshot of a computer

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 screenshot of a computer program

Description automatically generated**

**PART 3: Connect Explore Remote VM Using SSH (10 Points)**

**Introduction:**

In this session, I successfully connected to a remote virtual machine (VM) on the 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 "azhan-5560" using the Google Cloud SDK. The command used was:

gcloud beta compute ssh azhan-5560 --project=marine-champion-413421 --zone=us-central1-a

**2. Exploring the File System (Question 3.2):**

Upon establishing the SSH connection, I navigated through the file system of the remote VM. Using basic Linux commands, I changed to the home directory and listed its contents. This helped me understand the existing files and directories on the remote VM.

**3. Creating a Sub-Folder (Question 3.3):**

To organize my files, I created a new sub-folder named "JPTR\_NTBK" under the home directory using the command:

mkdir JPTR\_NTBK

**4. Changing the Current Directory (Question 3.4):**

After creating the "JPTR\_NTBK" folder, I changed my working directory to the newly created folder:

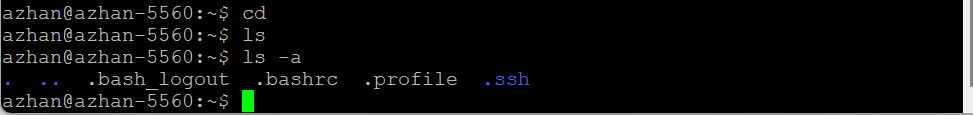
cd JPTR\_NTBK

This enabled me to focus on tasks within the specific directory.

**SCREENSHOTS:**

A screenshot of a computer screen

Description automatically generated





**PART 4: Start and Connect to Jupyter Notebook in Remote VM (30 Points)**

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

Introduction:

Today, I embarked on setting up and utilizing Jupyter Notebook on a remote virtual machine (VM). This report chronicles the steps I took to achieve this, aiming to harness the computational power of the remote environment for data analysis and coding tasks.

**Step 1: Starting the Jupyter Notebook Server:**

I began by opening an SSH terminal window to connect to the remote VM. After establishing the connection, I navigated to the directory where I wanted the Jupyter Notebook server to operate. With everything set, I executed the command:

$ jupyter notebook --port=8888

This command initiated the server on port 8888, ensuring that it's ready for action with a default token for authentication.

**Step 2: Forwarding Local Port to Remote Port:**

Switching gears, I opened a GCLOUD SDK terminal window to set up port forwarding. Using the provided command, I SSHed into the remote VM:

gcloud compute ssh us3238394@azhan-5560 --project=marine-champion-413421 --zone=us-central1-a -- -L 8000:localhost:8888

This command established a connection and forwarded my local port 8000 to port 8888 on the remote VM, paving the way for seamless interaction.

**Step 3: Connecting to the Jupyter Notebook Server:**

With the infrastructure in place, I fired up my web browser and entered one of the URLs provided:

http://localhost:8888/tree?token=[TOKEN]

http://127.0.0.1:8888/tree?token=[TOKEN]

I grabbed the token from the URL and pasted it into the login prompt on the Jupyter interface, ensuring secure access to the server.

**Conclusion:**

In conclusion, I successfully set up and connected to Jupyter Notebook on the remote VM. This setup promises efficient utilization of computational resources, enabling me to dive into data analysis and coding projects with ease.

**SCREENSHOTS:**

**A computer screen shot of a computer program

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**

**A computer screen with text and numbers

Description automatically generated**

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

**Report: Creating a Service Account for GCP Project**

**Introduction:**

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

**Step 1: Accessing Google Cloud Console**

To begin the process, I opened the Google Chrome browser and entered "Google Cloud Console" into the URL search text box. This directed me to the Google Cloud Console interface where I could manage our GCP project.

**Step 2: Creating the Service Account**

Once inside the Google Cloud Console, I navigated to the appropriate project for which we needed to create the service account. Then, I accessed the "IAM & Admin" section from the left-hand navigation menu and clicked on "Service accounts". This allowed me to initiate the creation of a new service account.

**Step 3: Providing Service Account Details**

In this step, I entered a descriptive name for the service account, which was "service-azhan5560". The unique email address assigned to this service account within the IAM of the GCP project is "service-azhan5560@marine-champion-413421.iam.gserviceaccount.com". This name and email address will help identify the purpose of the service account within our project. Clicking the "Create" button initiated the creation process.

**Step 4: Granting Roles**

After naming the service account, I was prompted to grant roles to it. I carefully selected the roles based on the specific permissions required for our project. Since no keys were associated with the service account at this point, I ensured that appropriate roles and permissions were assigned manually to enable its functionality within the project.

**Step 5: Completion**

With the service account created and roles assigned, we have successfully set up the necessary framework for its usage within our GCP project. However, we may need to create keys for authentication purposes in the future. The service account is currently enabled and ready for use in accessing and managing resources within the project.

**Conclusion:**

In conclusion, by following these steps, we have successfully created and configured a service account named "service-azhan5560" for our GCP project "marine-champion-413421". This account will facilitate secure access and management of resources within our project, enabling 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 taken to set up a service account in the Google Cloud Platform (GCP) to access the Natural Language API.

**Step 1: Access GCP Project Dashboard**

Accessed the GCP Project Dashboard by clicking on the Products and Services icon located in the top-left corner.

Navigated to the Cloud Overview and selected "Dashboard" from the drop-down menu.

**Step 2: Enable Natural Language API**

Located the APIs overview and clicked on "Go to APIs overview" to open the API page.

Searched for "Natural Language" and selected "Cloud Natural Language API".

Activated the API by clicking on "ENABLE".

**Step 3: Set Up Service Account of VM Instance**

Stopped the remote deep learning server (VM instance).

Accessed the information page of the VM instance by clicking on its name.

Clicked on "Edit" on the top menu bar.

Scrolled down to locate "Service Accounts".

Selected the previously created service account and saved the updated configuration.

**Step 4: Obtain Credentials of Service Account**

Navigated to IAM & Admin > Service Accounts page.

Clicked on the email of the service account to open its page.

Copied the "Email" of the service account to a Notepad file for reference.

**Step 5: Create Key File JSON of Service Account**

Opened an SSH connection to the remote deep learning server.

Executed the command gcloud iam service-accounts keys create ~/key.json --iam-account <email of service account> in the Home Directory.

Checked that the key file (key.json) was found in the home directory of the VM instance.

**Step 6: Verification and Conclusion**

Verified that the key file was successfully created and located in the home directory of the VM instance.

Confirmed the successful setup of the GCP Natural Language API.

Ensured that the service account was properly configured for accessing GCP systems.

**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**

**A screenshot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**PART VII: Join a Group (10 Points) TO-DO • Discuss with classmates to join a group.**

**Group members. Azhan Saleem(me), Erik Lorimor and Tarun hopefully.**