**ML USING PYTHON WITH CLOUD COMPUTING IN GOOGLE CLOUD PLATFORM**

**Guía inicial de preparación del entorno en la nube de Google**

**Nombre:** Rubén González Rodríguez.

**Pasos a seguir:**

1. **Creación de la cuenta de Google Cloud Platform (GCP):**
   1. Registrarse en GCP utilizando una cuenta de Google.
2. **Vinculación de medio de pago:**

Al crear una cuenta en GCP es posible acceder a un crédito de regalo por parte de Google de 200 USD, sin embargo, para poder obtenerlos es necesario agregar un medio de pago a la cuenta.

* 1. Ir a la sección **Billing > Payment method** y agregar como medio de pago una tarjeta de crédito activa y vigente.
  2. Hecho lo anterior se debe activar la **“cuenta de facturación”**, esto debido a que la nube de Google te permite utilizar algunos de los servicios sin tener la facturación activada, sin embargo, para acceder a toda la capacidad de la plataforma es necesario activarla.

**Nota:** Despreocuparse de lo anterior puesto que toda la facturación realizada se pagará con el saldo de regalo obtenido al registrarse.

1. **Creación de imagen con OS Linux y con la distribución de Python 3 de Anaconda:**

Esta imagen será el “template” del OS que se utilizará para levantar todas las VM que se creen, por lo que debe incluir también todas las herramientas necesarias de Data Science en Python.

**Crear y levantar una máquina virtual**

* 1. Ir a **Compute > Compute Engine** y seleccionar **“Create”** para crear una nueva VM**.**
  2. Asignar nombre de la vm (puede ser “vm1”).
  3. En **Boot Disk** click en **“change”** y seleccionar sistema operativo **Ubuntu** con la versión “20.04 LTS Minimal” (o la última versión LTS Minimal disponible).
  4. Click en **“Allow HTTP traffic”** para permitir funciones de red (conexión a internet) en la VM.
  5. Click en **“create”**.

**Nota:** Todas las opciones que no hayan sido mencionadas en los pasos anteriores se deben dejar tal cual se encuentran por defecto.

**Acceder a la VM mediante SSH y descargar la distribución de Python 3 de Anaconda**

1. Si todo lo anterior salió bien se debe observar en la sección “Compute Engine” lo siguiente:



El “check” verde indica que la VM arrancó correctamente y se encuentra en funcionamiento.

1. Hacer click en donde dice “SSH” para acceder mediante SSH a la VM. Se debe abrir una nueva ventana con la consola de Linux de la VM.
2. Descargar la distribución de Anaconda desde su repositorio mediante el comando

wget <https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86_64.sh>

El link corresponde a la última versión disponible al momento que se creó esta guía.

1. Esperar que se complete el proceso de descarga (ver barra de progreso).

**Instalar y configurar Python 3 utilizando la distribución de Anaconda descargada**

1. Instalar la distribución de Python con el comando

bash Anaconda3-2020.02-Linux-x86\_64.sh

**Nota:** El proceso de instalación requiere de aceptar todos los términos y seguir las instrucciones que se van indicando tales como, “presionar ENTER”, “responder yes”, etc. En otras palabras, debe responderse “yes” a todas las preguntas/indicaciones que vayan saltando en el camino hasta obtener el mensaje de instalación exitosa.

1. Una vez recibido el mensaje de instalación exitosa se debe activar dicha instalación mediante la ejecución del comando:

source ~/.bashrc

1. La verificación de la activación se lleva a cabo mediante la ejecución del comando

conda list

Ante lo cual debe recibirse como respuesta un listado de todos los paquetes y componentes que conforman la instalación.

1. Instalar el paquete “gcsfs”, el cual permite acceder a directorios alojados en Google Cloud Storage (GCS) desde Python, esto con el comando

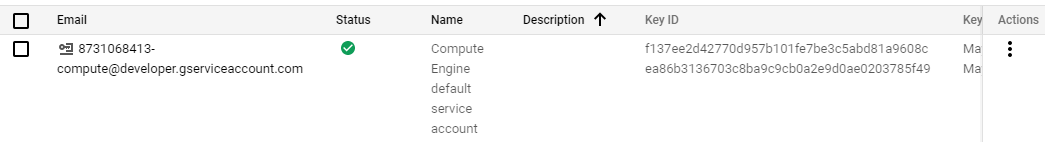
pip install gcsfs

1. Cerrar la ventana de la consola de la VM, ojo que esto NO apagará la máquina. El apagado debe hacerse desde la sección “Compute Engine” clickeando los 3 puntos que aparecen al lado de la palabra “SSH” que se utilizó para acceder a la VM y seleccionando **“Stop”**.
2. **Crear y configurar una “service account”:**

El entorno de GCP lo conforman numerosos servicios que interactúan entre sí mediante sus respectivas APIs. La manera de permitir que una VM acceda a estos servicios es mediante la autentificación de la identidad de la VM ante las APIs en cuestión utilizando una “service account”. Esta service account permite a las VM hacer “requests” a las APIs en función de los permisos otorgados a la VM, los cuales están establecidos en la service account.

* 1. Acceder a **Productos > IAM & Admin > Service accounts**.
  2. Por defecto GCP asigna una “default” service account, por lo que no hay necesidad de crear una nueva sino que se debe verificar que exista la cuenta “default”.

Si todo está bien debe verse algo parecido a lo siguiente:



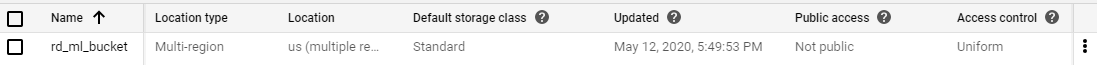
* 1. El campo **“Email”** es el usuario que identifica a la service account y se debe copiar para permitir que las VMs que usen esta service account puedan hacer requests a las APIs.

1. **Crear y configurar un “storage bucket”:**

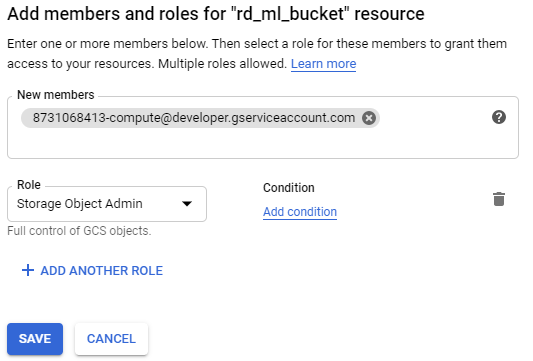
El “storage bucket” es uno de los varios servicios de almacenamiento incluídos en la nube de Google, este se caracteriza por funcionar como una bodega de almacenamiento de **objetos** y ofrece un bajo costo debido a que no cuenta con un File System, por lo que los objetos se depositan y extraen como un todo.

Para nuestro propósito, el bucket hará las veces de un “disco de red” al cual podrán acceder todas las VMs para leer y escribir datos, de manera que la información persista de manera independiente a la vida de las VMs.

1. Acceder a **Storage > Storage.**
2. Clickear en **“Create bucket”.**
3. Asignar un nombre al bucket.
4. Seleccionar el tipo de ubicación **“Multi-region”** y la ubicación en US (Estados Unidos).
5. Elegir el tipo de almacenamiento **“Standard”.**
6. Fijar el control de acceso en **“Uniform”.**
7. No modificar las opciones avanzadas y clickear en **“create”.**
8. Una vez creado el bucket volvera **Storage > Storage** y clickear en los 3 puntos para editar los permisos de acceso (**“edit bucket permissions”**)**.**



1. Clickear en **“add member”.**
2. Agregar como nuevo miembro al usuario de la service account **(Email copiado en el inciso 4 - C).**
3. En rol se debe seleccionar **Cloud Storage > Storage Object Admin** y clickear en **“save”**.



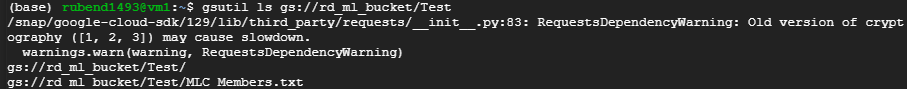
1. **Probar el entorno configurado:**

Se debe verificar que la VM creada previamente, cuya imagen servirá como template para todas las demás, puede acceder correctamente y levantar archivos desde el bucket.

* 1. Acceder a **Storage > Storage** y clickear en el bucket que se creó previamente.
  2. Clickear en **“create folder”** y crear un folder llamado **“Test”.**
  3. Acceder al folder creado y darle click en **“upload files”,** luego seleccionar y subir el archivo “csv” que contiene el DataFrame de prueba (alojado en el repositorio de GitHub) (<https://github.com/rdgonzalez14/ML_Scientist_Track/blob/master/CloudComputing_GCP_Docs/GuideN1_EnvSetup/MLC_Members.txt>).
  4. Esperar que se complete el proceso de carga.
  5. Acceder a **Compute > Compute Engine,** seleccionar la VM creada al inicio clickeando los 3 puntos y seleccionar **“start”.**
  6. Una vez iniciada la VM se debe clickear en “SSH” y abrir la ventana de comando de la VM.
  7. Desde la ventana de comandos de la VM se debe ingresar el comando

gsutil ls gs://nombre\_de\_mi\_bucket/Test

El comando **“gsutil”** es un comando integrado que viene incluido en GCP en todas las VMs que se crean y permite interactuar desde la VM en cuestión con la API de Google Storage (bucket). El resultado obtenido debe ser:



Que indica que se puede ver desde la VM el archivo “csv” subido previamente al bucket.

* 1. Para subir un archivo directamente al disco duro de la VM se debe clickear en el icono del engranaje de la ventana de comandos de dicha VM y seleccionar **“Upload file”** para luego elegir el archivo a pasar a la VM.
  2. Subir a la VM el script de Python alojado en el repositorio de GitHub (<https://github.com/rdgonzalez14/ML_Scientist_Track/blob/master/CloudComputing_GCP_Docs/GuideN1_EnvSetup/GCP_Test_Script.py>).
  3. Para ejecutar el script de Python en la VM utilizar el comando

python3 GCP\_Test\_Script.py

1. **Crear y guardar imagen del OS configurado en la VM inicial:**

Dado que la potencia de computo en la nube es ilimitada (casi), se hará necesario tener más de 1 VM funcionando a la vez, por lo que se debe buscar la manera de replicar todas las configuraciones realizadas en la VM inicial. Esto se logra creando una imagen de VM inicial y guardarla para utilizarla como base para la creación de cualquier otra VM.

* 1. Verificar que la VM esté apagada.
  2. Acceder a **Compute > Compute Engine > Images.**
  3. Click en **“create image”.**
  4. Asignar nombre a la imagen (“base-image”, por ejemplo).
  5. Seleccionar en **“Source”** la opción **“Disk”** y en **“Source disk”** seleccionar el nombre de la VM inicial utilizada para todas las configuraciones.
  6. Dejar el resto de opciones tal como están y dar click en **“create”.**
  7. Esperar que finalice el proceso, que puede tardar varios minutos en completarse.

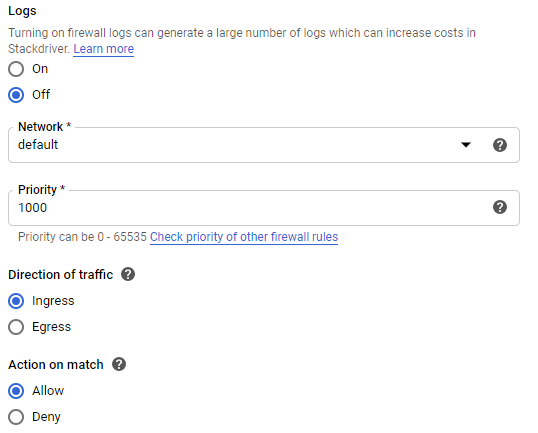
**Nota:** Ahora para crear nuevas máquinas virtuales se debe seleccionar la imagen creada por nosotros y no las predefinidas (como la usada en el inciso **3 - C**). Esto permitirá que todas las VM nuevas ya incluyan las configuraciones realizadas, es decir, instalación de Python 3 y configuración de acceso al bucket mediante la service account.

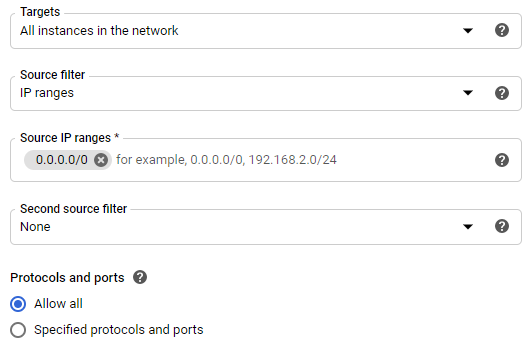
**Adicionales:**

1. **Configuración de firewall para las VMs:**

Por defecto, la configuración de firewall para todas las VMs que se crean en GCP permite la comunicación con todos los otros servicios de la plataforma mediante una IP de red interna. Sin embargo, la comunicación con dispositivos pertenecientes a redes externas se inicia bloqueada por defecto, por lo cual se debe configurar para poder interactuar con el mundo exterior. Esta configuración es una acción que se realiza una única vez y aplica para todas las VMs creadas en adelante.

1. Acceder a **Networking > VPC Network > Firewall rules.**
2. Clickear en **“Create firewall rule”.**
3. Asignar un nombre a la regla, este debe cumplir con los requisitos de la plataforma.
4. Configurar las propiedades con los valores que se muestran en las siguientes imágenes:





**Nota:** En el campo **“Source IP ranges”** debe ingresarse el valor de la imagen (0.0.0.0/0)!!!

1. Clickear en **“Create”**, lo que automáticamente creará y activará la regla de firewall.

Ahora todas las VMs creadas podrán interactuar con dispositivos conectados en redes externas como internet.

1. **Utilizar Jupyter Notebooks en las VMs:**

El ejemplo del **punto** **N°6** permite ejecutar scripts de Python con códigos ya terminados, sin embargo, si lo que se desea es implementar un código desde 0 en la VM, esto puede hacerse mediante la utilización de un Jupyter Notebook. Un punto a favor es que Jupyter ya viene incluido en la distribución de Anaconda, por lo que ya está instalado desde el **punto N°3**.

La ejecución de Jupyter para crear notebooks de Python en la VM es así:

1. Acceder mediante SSH a la consola de la VM (ver **punto N°3**).
2. Desde la consola, iniciar (ejecutar) jupyter con el comando

jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser

**Nota:** La opción **“ip”** establecida en **0.0.0.0** es un comodín que le indica que debe ejecutarse de forma pública para que sea accesible desde el exterior y a la vez toma el valor de la **ip externa** de la VM. El puerto a utilizar puede ser cualquier otro que se desee.

1. Desde una PC externa se puede acceder al Jupyter que está corriendo en la VM de GCP abriendo una ventana del navegador y escribiendo la dirección

http:// 34.69.160.176:8888/?token=88cc3cd0d00f72dbce56be73b7628b222d03b9860c6ffbcf

Donde la parte en **ROJO** corresponde a la **“ip externa”** de la VM de GCP, la parte en **AZUL** es el puerto designado y la parte en **VERDE** es un token de acceso asignado por Jupyter al ser ejecutado. Se muestran los valores referenciados en las siguientes imágenes:



