Hola, ayudame con este error de cloudshell:

"ERROR: (gcloud.services.enable) PERMISSION\_DENIED: Not found or permission denied for service(s):  .

Help Token: AeNz4PiycuMDWNTyIDleKcq2lIc8YOCKXGt85S8U1ZG\_GOSchoHpxyQQKCUZwNeIBWxPoMFokkVzIQ8ewkDx2wKHHKV-rb2AUcFmMXBoi-yveFwN. This command is authenticated as luistrocco@gmail.com which is the active account specified by the [core/account] property

- '@type': type.googleapis.com/google.rpc.PreconditionFailure

  violations:

  - subject: ?error\_code=220002&services=+

    type: googleapis.com

- '@type': type.googleapis.com/google.rpc.ErrorInfo

  domain: serviceusage.googleapis.com

  metadata:

    services: ' '

  reason: SERVICE\_CONFIG\_NOT\_FOUND\_OR\_PERMISSION\_DENIED"

This Cloud Shell error, **PERMISSION\_DENIED: Not found or permission denied for service(s):**, means your luistrocco@gmail.com account doesn't have the necessary permissions to enable the service(s) you're trying to use. The error message also indicates that the service might not exist or the service configuration isn't found.

**What's Happening**

* **Permissions Issue:** Your luistrocco@gmail.com account lacks the required IAM (Identity and Access Management) roles to perform the action.
* **Missing Service or Configuration:** It's possible the service you're trying to enable either doesn't exist, or there's an issue with its configuration. The blank space after services: suggests that the specific service you were trying to enable wasn't properly identified in the command you ran.

**How to Fix It**

Here's how you can troubleshoot and resolve this issue:

1. **Identify the Missing Service:**
   * First, you need to remember *which* Google Cloud service you were trying to enable. The error message shows services: ' ', meaning the service name wasn't captured. Review the gcloud command you ran just before getting this error. For example, were you trying to enable compute.googleapis.com or cloudbuild.googleapis.com?
2. **Verify Your Permissions:**
   * **Check Your Roles:** You need a role that grants permission to enable services. The most common roles for this are:
     + **Service Usage Admin** (roles/serviceusage.admin): This role allows you to enable and disable services.
     + **Owner** (roles/owner): This role grants full administrative access to all resources. (Use with caution as it's very broad).
   * **How to Check:**
     + Go to the Google Cloud Console.
     + Navigate to **IAM & Admin > IAM**.
     + Find your luistrocco@gmail.com account in the list and check the roles assigned to it.
   * **If You Don't Have the Role:** Ask an administrator of your Google Cloud project to grant your luistrocco@gmail.com account one of the roles mentioned above (preferably **Service Usage Admin**) on the project where you are trying to enable the service.
3. **Ensure Correct Service Name:**
   * Once you've identified the service you wanted to enable, double-check its exact name. Google Cloud services typically follow the format servicename.googleapis.com. For example, compute.googleapis.com for Compute Engine, or cloudbuild.googleapis.com for Cloud Build.
4. **Try Enabling the Service Again:**
   * Once your permissions are confirmed and you know the correct service name, try running the gcloud services enable command again in Cloud Shell.
   * **Example:** If you were trying to enable the Compute Engine API, the command would be:

Bash

gcloud services enable compute.googleapis.com

**In Summary**

The core of the problem is likely that your luistrocco@gmail.com account doesn't have the **Service Usage Admin** role, or you're not specifying the correct service name in your gcloud services enable command. Find the correct service name and have an administrator grant you the necessary permissions.

Let me know what service you were trying to enable, and I can provide more specific guidance!

y en cloudshell puedo cambiarme de usuario? estoy en luistrocco@gmail, pero podría cambiarme a 837138706172-compute@developer.gserviceaccount.com, dame el comando por favor

Ver razonamiento

Sí, definitivamente puedes cambiar de usuario en Cloud Shell y autenticarte con una cuenta de servicio (837138706172-compute@developer.gserviceaccount.com en tu caso).

Para hacer esto, necesitas el **archivo de clave JSON** asociado a esa cuenta de servicio. Este archivo es lo que usa gcloud para autenticarse como la cuenta de servicio.

Aquí te explico los pasos:

1. **Asegúrate de tener el archivo de clave JSON de la cuenta de servicio.**
   * Si no lo tienes, debes generarlo desde la Consola de Google Cloud:
     1. Ve a **IAM & Admin > Service Accounts**.
     2. Encuentra la cuenta de servicio 837138706172-compute@developer.gserviceaccount.com.
     3. Haz clic en los tres puntos verticales al lado de la cuenta de servicio y selecciona **Manage keys**.
     4. Haz clic en **ADD KEY > Create new key**.
     5. Selecciona **JSON** como el tipo de clave y haz clic en **Create**. Esto descargará el archivo JSON a tu máquina local.
   * Una vez que lo tengas, necesitas subirlo a tu Cloud Shell. Puedes hacerlo arrastrando y soltando el archivo en la ventana de Cloud Shell, o usando el menú "File" en la parte superior de Cloud Shell y seleccionando "Upload File". Guarda el archivo en un lugar seguro, por ejemplo, en tu directorio principal (e.g., ~/my-service-account-key.json).
2. **Activa la cuenta de servicio en Cloud Shell.** Una vez que el archivo JSON esté en tu Cloud Shell (supongamos que lo llamaste service-account-key.json y lo guardaste en tu directorio principal), usa el siguiente comando:

Bash

gcloud auth activate-service-account 837138706172-compute@developer.gserviceaccount.com --key-file=~/service-account-key.json --project=YOUR\_PROJECT\_ID

**Importante:**

* + Reemplaza ~/service-account-key.json con la ruta real a tu archivo JSON si lo guardaste en otro lugar.
  + Reemplaza YOUR\_PROJECT\_ID con el ID de tu proyecto de Google Cloud (por ejemplo, my-gcp-project-12345). La cuenta de servicio debe tener permisos en este proyecto.

1. **Verifica la cuenta activa.** Después de ejecutar el comando anterior, puedes verificar qué cuenta está activa con:

Bash

gcloud auth list

Verás que la cuenta de servicio 837138706172-compute@developer.gserviceaccount.com ahora está marcada con un asterisco (\*), indicando que es la cuenta activa.

**Notas Importantes:**

* **Seguridad del archivo de clave:** El archivo JSON contiene las credenciales completas de tu cuenta de servicio. Trátalo con la misma seguridad que tratarías una contraseña o una clave SSH. No lo compartas públicamente y bórralo de tu Cloud Shell una vez que ya no lo necesites, o asegúrate de que esté en un lugar seguro.
* **Permisos de la cuenta de servicio:** Asegúrate de que la cuenta de servicio (837138706172-compute@developer.gserviceaccount.com) tenga los roles IAM necesarios para realizar las operaciones que deseas. El error PERMISSION\_DENIED que mencionaste anteriormente podría seguir ocurriendo si la cuenta de servicio no tiene los permisos adecuados (por ejemplo, Service Usage Admin para habilitar servicios).

Este método es el estándar para usar cuentas de servicio en entornos donde no hay una sesión de navegador para autenticar un usuario regular.

me está diciendo que no encuentra el directorio y estoy en home, mira el error. QUé hago?:

ERROR: (gcloud.auth.activate-service-account) Unable to read file [./home/luistrocco/etl-lt-002-9c7d2f079b4f.json--project=etl-lt-002]: [Errno 2] No such file or directory: './home/luistrocco/etl-lt-002-9c7d2f079b4f.json--project=etl-lt-002'

El error No such file or directory (No existe tal archivo o directorio) indica que la ruta que le estás dando al archivo JSON de la clave de la cuenta de servicio es incorrecta.

Observa el error que te muestra: ./home/luistrocco/etl-lt-002-9c7d2f079b4f.json--project=etl-lt-002

El problema es que el --project=etl-lt-002 se ha pegado al nombre del archivo JSON. Esto significa que el sistema está buscando un archivo con un nombre muy largo que incluye --project=etl-lt-002 al final, y ese archivo no existe.

**La solución es simple: asegúrate de que haya un espacio entre el nombre del archivo JSON y el flag --project.**

El comando correcto debería ser:

Bash

gcloud auth activate-service-account 837138706172-compute@developer.gserviceaccount.com --key-file=./home/luistrocco/etl-lt-002-9c7d2f079b4f.json --project=etl-lt-002

**Verifica lo siguiente:**

1. **Espacio:** Asegúrate de que haya un espacio entre .json y --project.
2. **Ruta del archivo:** Confirma que el archivo etl-lt-002-9c7d2f079b4f.json realmente se encuentra en la ruta ./home/luistrocco/. Si lo subiste a tu directorio principal (~), entonces ~/etl-lt-002-9c7d2f079b4f.json o ./etl-lt-002-9c7d2f079b4f.json (si ya estás en /home/luistrocco/) deberían funcionar.

Intenta el comando de nuevo con el espacio correcto.

como borro un archivo repetido en cloudshell?

Claro, para borrar un archivo repetido en Cloud Shell, usarás el comando rm (remove). Aquí te explico cómo hacerlo de forma segura:

**Pasos para Borrar un Archivo**

1. **Lista los archivos para confirmar:** Primero, es buena idea usar ls para asegurarte de que ves el archivo duplicado y para verificar su nombre exacto y la ruta. Por ejemplo, si sospechas que tienes mi\_archivo.txt y mi\_archivo.txt.bak, podrías hacer:

Bash

ls -l mi\_archivo\*

Esto te mostrará todos los archivos que empiezan con mi\_archivo.

1. **Borra el archivo duplicado:** Una vez que estés seguro de cuál es el archivo que quieres eliminar, usa el comando rm seguido del nombre completo del archivo (incluyendo la extensión y cualquier parte adicional si es un duplicado con otro nombre).

Por ejemplo, si tienes mi\_informe.txt y mi\_informe\_copia.txt y quieres borrar la copia:

Bash

rm mi\_informe\_copia.txt

O, si tienes documento.pdf y documento.pdf.1 y quieres borrar la versión .1:

Bash

rm documento.pdf.1

**Consideraciones Importantes**

* **¡Cuidado con rm!** El comando rm borra archivos permanentemente. **No hay una papelera de reciclaje** en Cloud Shell. Una vez que borras algo con rm, no se puede recuperar fácilmente.
* **Comprueba la ruta:** Si el archivo no está en el directorio donde te encuentras actualmente, deberás especificar la ruta completa. Por ejemplo, si el archivo está en una subcarpeta llamada docs:

Bash

rm docs/mi\_informe\_copia.txt

* **Archivos con espacios en el nombre:** Si el nombre del archivo tiene espacios, debes encerrarlo entre comillas:

Bash

rm "Mi Archivo Duplicado.txt"

* **Confirmación interactiva (opcional):** Si quieres que rm te pida confirmación antes de borrar cada archivo, puedes usar la opción -i:

Bash

rm -i mi\_informe\_copia.txt

Te preguntará: rm: remove 'mi\_informe\_copia.txt'? y debes escribir y para sí o n para no.

Simplemente ten cuidado de escribir el nombre del archivo que realmente quieres eliminar. Si tienes dudas, puedes pegar el nombre completo del archivo que te aparece al hacer ls para asegurarte de no cometer un error.

estoy tratando de borrar este archi y no me deja jaja:

rm etl-lt-002-9c7d2f079b4f\_(1).json

-bash: syntax error near unexpected token `('

Jajaja, ¡entiendo perfectamente tu frustración! Ese error de sintaxis ocurre porque los paréntesis ( y ) tienen un significado especial en Bash (la shell que usa Cloud Shell). Cuando los usas directamente, Bash espera que sean parte de un comando o una estructura de control, no parte de un nombre de archivo.

**La Solución: Escapar los Caracteres Especiales o Usar Comillas**

Hay dos formas principales de decirle a Bash que trate los paréntesis como parte del nombre del archivo y no como un comando:

1. **Escapar los Caracteres (Usando \):** Puedes poner una barra invertida \ justo antes de cada carácter especial (en este caso, ( y )). Esto le dice a Bash que el siguiente carácter debe interpretarse literalmente.

Bash

rm etl-lt-002-9c7d2f079b4f\_\(1\).json

1. **Encerrar el Nombre del Archivo entre Comillas (Recomendado):** Esta es generalmente la forma más fácil y segura cuando los nombres de archivo contienen espacios, paréntesis, o cualquier otro carácter especial. Al poner todo el nombre del archivo entre comillas simples (') o dobles ("), le dices a Bash que trate todo lo que está dentro de las comillas como una sola cadena literal.

Bash

rm 'etl-lt-002-9c7d2f079b4f\_(1).json'

**¡Este es el método más recomendado para tu caso!** Es más limpio y menos propenso a errores.

**Qué hacer ahora**

Copia y pega el siguiente comando en tu Cloud Shell y ejecútalo:

Bash

rm 'etl-lt-002-9c7d2f079b4f\_(1).json'

Eso debería resolver el problema y borrar tu archivo. ¡Cuéntame si funciona!

ya he hecho de todo y cuando intento esto:

gcloud services enable \

    bigquery.googleapis.com \

    compute.googleapis.com \

    storage.googleapis.com \

    composer.googleapis.com \

    dataproc.googleapis.com \

    dataflow.googleapis.com \

    secretmanager.googleapis.com \

    sqladmin.googleapis.com \

    cloudfunctions.googleapis.com \

    cloudscheduler.googleapis.com \

    cloudresourcemanager.googleapis.com

me sigue diciendo:

ERROR: (gcloud.services.enable) PERMISSION\_DENIED: Not found or permission denied for service(s):  .

Help Token: AeNz4PibDvJjtzl0B3OMXyO4VG5Qb9SbGpd\_J73bYozoJK7RZpo\_aEi9VhOniyZUMfs9cyQcBZYnr4cHRFxaJ2PKECiBR5LF\_Iaq0R3etVZ3EltE. This command is authenticated as 837138706172-compute@developer.gserviceaccount.com which is the active account specified by the [core/account] property

- '@type': type.googleapis.com/google.rpc.PreconditionFailure

  violations:

  - subject: ?error\_code=220002&services=+

    type: googleapis.com

- '@type': type.googleapis.com/google.rpc.ErrorInfo

  domain: serviceusage.googleapis.com

  metadata:

    services: ' '

  reason: SERVICE\_CONFIG\_NOT\_FOUND\_OR\_PERMISSION\_DENIED

-bash: bigquery.googleapis.com: command not found

-bash: compute.googleapis.com: command not found

-bash: storage.googleapis.com: command not found

-bash: composer.googleapis.com: command not found

-bash: dataproc.googleapis.com: command not found

-bash: dataflow.googleapis.com: command not found

-bash: secretmanager.googleapis.com: command not found

-bash: sqladmin.googleapis.com: command not found

-bash: cloudfunctions.googleapis.com: command not found

-bash: cloudscheduler.googleapis.com: command not found

-bash: cloudresourcemanager.googleapis.com: command not found

Qué hagp?

¡Claro, entiendo tu frustración! Parece que hay dos problemas distintos en el error que estás viendo:

1. **PERMISSION\_DENIED**: Este sigue siendo el mensaje principal de Google Cloud. Significa que, aunque ya cambiaste a la cuenta de servicio 837138706172-compute@developer.gserviceaccount.com, esa cuenta **no tiene los permisos suficientes** en el proyecto actual para habilitar esos servicios.
2. **command not found**: Este es un error de Bash (tu terminal de Cloud Shell), no de Google Cloud. Ocurre porque la forma en que estás dividiendo el comando en varias líneas no es la correcta para gcloud. Las barras invertidas (\) al final de cada línea son correctas, pero parece haber caracteres invisibles o espacios después de la barra invertida que las anulan, haciendo que Bash interprete cada nombre de servicio como un comando separado. Además, parece que hay un carácter no imprimible al inicio de tu primera línea ( \ ).

Vamos a abordar ambos problemas.

**Solución al Error de Bash (command not found)**

El error command not found es el más fácil de corregir. Hay un pequeño problema con cómo estás estructurando las líneas. Asegúrate de que **NO HAYA NINGÚN ESPACIO DESPUÉS DE LA BARRA INVERTIDA (\)** en cada línea. Cualquier espacio después de la barra invertida anula su función y Bash intenta ejecutar la siguiente palabra como un nuevo comando.

También, he notado un espacio antes de la primera barra invertida en tu comando (\ ). Asegúrate de que la primera línea comience directamente con gcloud services enable y que la barra invertida esté inmediatamente al final de la línea si la usas para dividir el comando.

Intenta con este formato, copiando y pegando cuidadosamente:

Bash

gcloud services enable \

bigquery.googleapis.com \

compute.googleapis.com \

storage.googleapis.com \

composer.googleapis.com \

dataproc.googleapis.com \

dataflow.googleapis.com \

secretmanager.googleapis.com \

sqladmin.googleapis.com \

cloudfunctions.googleapis.com \

cloudscheduler.googleapis.com \

cloudresourcemanager.googleapis.com

**Consejo:** Para evitar problemas con los espacios, también puedes poner todos los servicios en una sola línea, aunque sea un poco larga:

Bash

gcloud services enable bigquery.googleapis.com compute.googleapis.com storage.googleapis.com composer.googleapis.com dataproc.googleapis.com dataflow.googleapis.com secretmanager.googleapis.com sqladmin.googleapis.com cloudfunctions.googleapis.com cloudscheduler.googleapis.com cloudresourcemanager.googleapis.com

**Solución al Error de Permisos (PERMISSION\_DENIED)**

Incluso si solucionas el error de Bash, el PERMISSION\_DENIED persistirá si la cuenta de servicio no tiene los roles adecuados.

La cuenta 837138706172-compute@developer.gserviceaccount.com es una **cuenta de servicio predeterminada de Compute Engine**. Por defecto, estas cuentas suelen tener el rol de **Editor** en el proyecto. El rol de **Editor** (roles/editor) **SÍ** debería permitirte habilitar servicios.

Si a pesar de ser Editor sigues viendo PERMISSION\_DENIED, podría haber algunas razones:

1. **Propagación de Permisos:** A veces, los cambios de permisos en Google Cloud pueden tardar unos minutos en propagarse completamente. Si acabas de asignar el rol, espera 5-10 minutos y vuelve a intentarlo.
2. **Conflicto de Políticas de la Organización:** En organizaciones grandes, puede haber políticas de la organización que restrinjan qué servicios se pueden habilitar o por quién, incluso si tienes el rol de Editor. Esto es menos común en proyectos personales.
3. **Error al autenticar el proyecto:** Asegúrate de que la cuenta de servicio esté autenticada en el **proyecto correcto** donde quieres habilitar los servicios. Cuando activaste la cuenta de servicio, usaste --project=YOUR\_PROJECT\_ID. Confirma que etl-lt-002 es el ID de proyecto correcto.

Puedes verificar el proyecto configurado actualmente en tu Cloud Shell con:

Bash

gcloud config get-value project

Si no es etl-lt-002, cámbialo con:

Bash

gcloud config set project etl-lt-002

**Si los problemas persisten después de corregir el comando y esperar:**

La acción más directa es verificar los permisos de la cuenta de servicio:

1. Ve a la **Consola de Google Cloud**.
2. Navega a **IAM & Admin > IAM**.
3. En la parte superior, asegúrate de que estás en el proyecto **etl-lt-002** (o el ID de tu proyecto).
4. Busca 837138706172-compute@developer.gserviceaccount.com en la lista de miembros.
5. Verifica los roles asignados a esta cuenta. Debería tener al menos el rol de **Editor** (roles/editor) o, idealmente, el rol de **Service Usage Admin** (roles/serviceusage.admin) si solo necesitas gestionar servicios.

Si la cuenta de servicio no tiene el rol de **Editor** o **Service Usage Admin**, o si hay roles que niegan explícitamente la capacidad de habilitar servicios, un usuario con el rol de **Propietario** (roles/owner) del proyecto deberá asignarle el rol adecuado a esa cuenta de servicio.

Intenta primero con el comando corregido para el error de Bash. Si el PERMISSION\_DENIED continúa, entonces los permisos de la cuenta de servicio son la clave.

También me pasa que me da error cuando le doy este comando para crear un cluster:

CLUSTER\_NAME="my-demo-cluster"

REGION="us-east1"

gcloud dataproc clusters create ${CLUSTER\_NAME} \

    --region ${REGION} \

    --num-workers=2 \

    --worker-machine-type=n1-standard-2 \

    --worker-boot-disk-size=50 \

    --master-machine-type=n1-standard-2 \

    --master-boot-disk-size=50 \

    --image-version=2.0-debian10 \

    --enable-component-gateway \

    --optional-components=JUPYTER \

    --initialization-actions=gs://goog-dataproc-initialization-actions-${REGION}/connectors/connectors.sh \

    --metadata bigquery-connector-version=1.2.0 \

    --metadata spark-bigquery-connector-version=0.21.0

me dice este error:

ERROR: (gcloud.dataproc.clusters.create) unrecognized arguments:

To search the help text of gcloud commands, run:

  gcloud help -- SEARCH\_TERMS

-bash: --region: command not found

-bash: --num-workers=2: command not found

-bash: --worker-machine-type=n1-standard-2: command not found

-bash: --worker-boot-disk-size=50: command not found

-bash: --master-machine-type=n1-standard-2: command not found

-bash: --master-boot-disk-size=50: command not found

-bash: --image-version=2.0-debian10: command not found

-bash: --enable-component-gateway: command not found

-bash: --optional-components=JUPYTER: command not found

-bash: --initialization-actions=gs://goog-dataproc-initialization-actions-us-east1/connectors/connectors.sh: No such file or directory

-bash: --metadata: command not found

-bash: --metadata: command not found

¡Uf! Parece que estás enfrentando el mismo problema de sintaxis de Bash que tenías al habilitar los servicios, pero esta vez con tu comando gcloud dataproc clusters create.

**Entendiendo el Error command not found**

El error -bash: --region: command not found y todos los subsiguientes command not found ocurren porque **hay caracteres invisibles o espacios después de las barras invertidas (\)** al final de cada línea.

Cuando Bash ve una barra invertida \ al final de una línea, espera que la siguiente línea sea una continuación del mismo comando. Sin embargo, si hay un espacio (o cualquier otro carácter invisible) después de la \, Bash lo interpreta como el final del comando en esa línea y luego intenta ejecutar la siguiente palabra (--region, --num-workers=2, etc.) como un nuevo comando, lo cual, obviamente, falla.

Además, el error unrecognized arguments: al inicio de la salida de gcloud también apunta a que la primera parte del comando (gcloud dataproc clusters create ${CLUSTER\_NAME}) se está ejecutando sola, y todo lo demás se está "perdiendo" para gcloud debido a esos caracteres extra.

**La Solución**

La clave es **eliminar cualquier espacio o carácter invisible después de cada barra invertida (\)** en tu comando.

Aquí tienes el comando correctamente formateado. Intenta copiar y pegar este bloque directamente en tu Cloud Shell. Es crucial que no haya espacios después de las barras invertidas al final de cada línea:

Bash

CLUSTER\_NAME="my-demo-cluster"

REGION="us-east1"

gcloud dataproc clusters create ${CLUSTER\_NAME} \

--region ${REGION} \

--num-workers=2 \

--worker-machine-type=n1-standard-2 \

--worker-boot-disk-size=50 \

--master-machine-type=n1-standard-2 \

--master-boot-disk-size=50 \

--image-version=2.0-debian10 \

--enable-component-gateway \

--optional-components=JUPYTER \

--initialization-actions=gs://goog-dataproc-initialization-actions-${REGION}/connectors/connectors.sh \

--metadata bigquery-connector-version=1.2.0 \

--metadata spark-bigquery-connector-version=0.21.0

**Consejo:** Si copiar y pegar sigue dándote problemas con estos caracteres invisibles, otra estrategia es **escribir el comando completo en una sola línea**. Es más largo, pero elimina el riesgo de los problemas con las barras invertidas:

Bash

CLUSTER\_NAME="my-demo-cluster"; REGION="us-east1"; gcloud dataproc clusters create ${CLUSTER\_NAME} --region ${REGION} --num-workers=2 --worker-machine-type=n1-standard-2 --worker-boot-disk-size=50 --master-machine-type=n1-standard-2 --master-boot-disk-size=50 --image-version=2.0-debian10 --enable-component-gateway --optional-components=JUPYTER --initialization-actions=gs://goog-dataproc-initialization-actions-${REGION}/connectors/connectors.sh --metadata bigquery-connector-version=1.2.0 --metadata spark-bigquery-connector-version=0.21.0

**Posibles Problemas Adicionales (Después de la Sintaxis)**

Una vez que arregles la sintaxis, el comando debería ejecutarse correctamente. Sin embargo, podrías encontrarte con otros errores que no son de sintaxis, como:

* **Permisos (PERMISSION\_DENIED):** Si tu cuenta activa (837138706172-compute@developer.gserviceaccount.com) no tiene los roles necesarios para crear clusters de Dataproc (como roles/dataproc.editor o roles/editor), seguirás viendo errores de permiso.
* **Recursos Agotados:** A veces, en ciertas regiones, es posible que no haya suficientes recursos de la máquina virtual que solicitas.
* **initialization-actions no encontrado:** El error -bash: --initialization-actions=...: No such file or directory también es un error de Bash por la misma razón de los espacios. Una vez que corrijas la sintaxis, Google Cloud verificará si el bucket gs://goog-dataproc-initialization-actions-us-east1 y el script connectors/connectors.sh existen y son accesibles. Si no, obtendrás un error diferente de Dataproc.

Intenta con el comando corregido y dime si eso soluciona el problema. Si aún así te da un PERMISSION\_DENIED o algún otro error específico de Google Cloud, entonces tendremos que investigar los permisos o la configuración del proyecto más a fondo.