# Assignment: Configure GCS Access in Databricks Using Secrets

## Step 1: Create a Secret Scope

databricks secrets create-scope gcs-secrets

## Step 2: Add GCS Credentials as Secrets

Replace the values below with the appropriate credentials from your GCP Service Account JSON key.  
  
databricks secrets put-secret gcs-secrets client-email

databricks secrets put-secret gcs-secrets private-key

databricks secrets put-secret gcs-secrets private-key-id

databricks secrets put-secret gcs-secrets project-id

## Step 3: Verify Secret Scope and Keys

List available secret scopes:  
databricks secrets list-scopes  
  
Sample output:  
Scope Backend Type  
gcs-secrets DATABRICKS  
  
List secrets under the gcs-secrets scope:  
databricks secrets list-secrets gcs-secrets  
  
Sample output:  
Key Last Updated Timestamp  
client-email 1748231452211  
private-key 1748231529388  
private-key-id 1748231585217  
project-id 1748231621493

A screenshot of a computer

AI-generated content may be incorrect.

## Use Secrets and Mount GCS Bucket in a Databricks Notebook

## Step 1: Validate Secret Scope Access in Notebook

dbutils.secrets.listScopes()  
dbutils.secrets.list("gcs-secrets")

## Step 2: Retrieve Secrets

client\_email = dbutils.secrets.get(scope="gcs-secrets", key="client-email")  
private\_key = dbutils.secrets.get(scope="gcs-secrets", key="private-key").replace("\\n", "\n")  
private\_key\_id = dbutils.secrets.get(scope="gcs-secrets", key="private-key-id")  
project\_id = dbutils.secrets.get(scope="gcs-secrets", key="project-id")

## Step 3: Prepare Mount Configuration

conf = {  
 "fs.gs.auth.service.account.enable": "true",  
 "fs.gs.auth.service.account.email": client\_email,  
 "fs.gs.auth.service.account.private.key": private\_key,  
 "fs.gs.auth.service.account.private.key.id": private\_key\_id,  
 "fs.gs.project.id": project\_id  
}

## Step 4: Mount the GCS Bucket

Replace <your-bucket-name> and <your-mount-name> with your actual GCS bucket and desired mount point name.  
  
dbutils.fs.mount(  
 source = "gs://<your-bucket-name>",  
 mount\_point = "/mnt/<your-mount-name>",  
 extra\_configs = conf  
)

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Read Data from the Mounted Bucket

Replace <path> and <your-file.csv> with the correct path and filename.  
  
df = spark.read.format("csv").load("/mnt/<your-mount-name>/<path>/<your-file.csv>")  
df.show()

## Unmount GCS Bucket (If Needed)

dbutils.fs.unmount("/mnt/<your-mount-name>")