SFTP to GCS Cloud Function Template

This template enables the automated transfer of files from an SFTP server to a Google Cloud Storage (GCS) bucket using a Google Cloud Function triggered via HTTP. It's ready for fast replication across different integrations by updating minimal configuration.

# What’s Included

* main.py: Cloud Function to fetch the latest file from SFTP and upload it to GCS.
* requirements.txt: Python dependencies.
* env\_vars.json: Non-sensitive environment variables.
* deploy.yml: GitHub Actions workflow to deploy the function securely using a service account and secrets. Must be placed in `.github/workflows` directory.

# How to Use This Template

## 1. Clone and Configure

Clone the repository and update the following:

Files:

* env\_vars.json – Environment variables, set variables as per your needs
* deploy.yml – Deployment configuration, set variables as per your needs and update `SFTP\_PASSWORD\_NAME` to match the name of your secret
* main.py – Cloud Function code
* requirements.txt – Python dependencies, do not modify unless you need to add additional dependencies

GitHub Actions secrets:

* GCP\_SA\_KEY – GitHub Actions secret, obtainable from GCP

## 2. Setup Google Cloud

### a. Enable APIs

Ensure these APIs are enabled in your GCP project:

* • Cloud Functions
* • Secret Manager
* • Cloud Build
* • Cloud Scheduler (optional)

### b. Create Secret for SFTP Password

GCP Secret Manager: create a new Secret with name `SFTP\_PASSWORD\_NAME` (example value) and the value of your SFTP password.

# 3. Required User Configuration

The following items must be configured before deploying your Cloud Function:

## 3.1 env\_vars.json

Update this file with values that reflect your target integration. Example:

{  
 "SFTP\_HOST": "your-sftp-host.example.com",  
 "SFTP\_PORT": "22",  
 "SFTP\_USERNAME": "your-username",  
 "ROOT\_FOLDER": "/path/to/files",  
 "GCS\_BUCKET\_NAME": "your-destination-bucket",  
 "GCP\_PROJECT\_NAME": "your-gcp-project-id"  
}

• Do not include passwords or secrets in this file.  
• Make sure the bucket and project exist in GCP.

## 3.2 deploy.yml

Located in `.github/workflows/deploy.yml`, this file defines how GitHub deploys your function. Update the following:

* • `project\_id`: Set this to your actual GCP Project ID
* • `--region`: Set the GCP region you want to deploy in
* • `--entry-point`: Confirm the function name matches your `main.py` function (default is `sftp\_to\_gcs`)
* • `--set-secrets`: Update the secret name (`SFTP\_PASSWORD=SFTP\_PASSWORD\_NAME:latest`) to match what you create in Secret Manager

## 3.3 GitHub Secrets

Navigate to your GitHub repository → Settings → Secrets → Actions. Add the following:

* • `GCP\_SA\_KEY`: Your GCP service account key JSON. Required for authentication and deployment.