📖 **04-composer.md**

Creating Cloud Composer Environment

This module includes all prerequisites for setting up the Cloud Composer Environment for the Serverless Spark Lab-

0. Prerequisites

1. GCP Project Details

Note the project number and project ID.
We will need this for the rest of the lab

2. IAM Roles needed to create Cloud Composer Environment

Grant the following permissions

- Composer Worker
- Dataproc Editor
- Service Account User

3. IAM Roles needed to upload and execute DAGs on the Cloud Composer Environment

Grant the following permissions

- Environment User and Storage Object Viewer
- Service Account User
- Storage Object Admin

4. Attach cloud shell to your project.

Open Cloud shell or navigate to shell.cloud.google.com
Run the below command to set the project in the cloud shell terminal:

```
gcloud config set project $PROJECT_ID
```

1. Declare variables

We will use these throughout the lab.
Run the below in cloud shells against the project you selected-

```
PROJECT_ID=#Project ID
COMPOSER_SA=composer-sa
```

```
COMPOSER_ENV=<your_composer_environment_name>
REGION=#Region to be used
VPC=#VPC Network Name
SUBNET=#Subnet with Private Google Access enabled
```

2. Create a Service Account for the Composer Environment

```
gcloud iam service-accounts create $COMPOSER_SA \
 --description="Service Account for Cloud Composer Environment" \
 --display-name "Cloud Composer SA"
```

3. Grant IAM Permissions for Composer Service Account

3.1.a. Composer Worker role for Composer Service Account

```
gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member serviceAccount:$COMPOSER_SA@$PROJECT_ID.iam.gserviceaccount.com --role roles/composer.worker
```

3.1.b. Dataproc Editor role for Composer Service Account

```
gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member serviceAccount:$COMPOSER_SA@$PROJECT_ID.iam.gserviceaccount.com --role roles/dataproc.editor
```

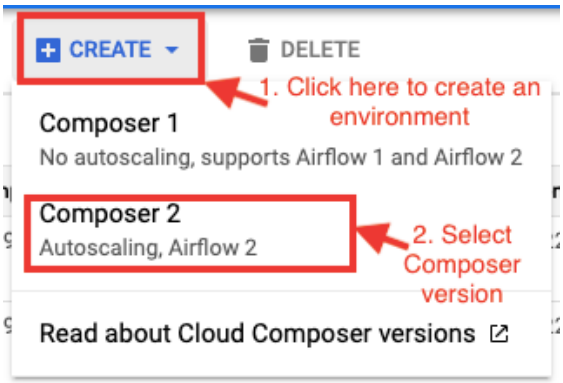
3.1.c. Service Account User role for Composer Service Account

```
gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member serviceAccount:$COMPOSER_SA@$PROJECT_ID.iam.gserviceaccount.com --role roles/iam.serviceAccountUser
```

4. Create a Composer Environment

4.1 Create a Composer Environment through the GCP console

Navigate to the Composer Service in your GCP project and click on **+CREATE>Composer 2**



Next, fill in the following values in the environment creation window :

- **Name** - A unique identifier for your environment
- **Location** - The region where you want to create the environment
- **Image Version** - Select the latest image version available
- **Service Account** - The Cloud Composer Service Account provided by the Admin
- **Network Configuration** - select the network and subnetwork with Private Google Access Enabled
- Next under **Web Server Network Access Control** select one of the below options:
  Allow access only from specific IP addresses and add all IP addresses which should have access to the Airflow UI **Allow access from all IP addresses**
- Next, click on **Create** to create the environment

4.2 Create a Composer environment through cloud shell

- To create a composer environment which will allow all IP addresses to access the Airflow web server execute the below command in cloud shell:

```
gcloud composer environments create $COMPOSER_ENV \
--location $REGION \
--environment-size small \
--service-account $COMPOSER_SA@$PROJECT_ID.iam.gserviceaccount.com \
--image-version composer-2.0.9-airflow-2.2.3 \
--network $VPC \
--subnetwork $SUBNET \
--web-server-allow-all
```

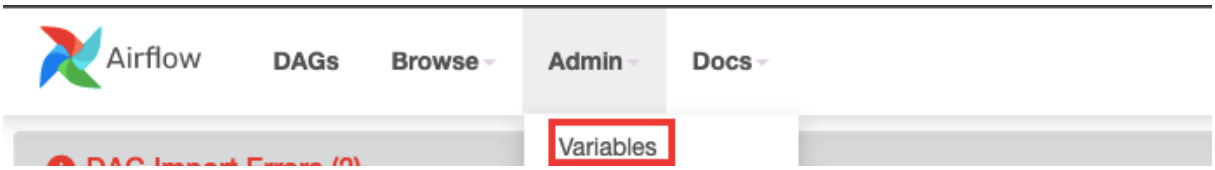
- Alternatively, to create a composer environment which will allow a specific list of IPv4 or IPv6 ranges to access the Airflow web server, execute the following command in cloud shell:

```
gcloud composer environments create $COMPOSER_ENV \
--location $REGION \
--environment-size small \
--service-account $COMPOSER_SA@$PROJECT_ID.iam.gserviceaccount.com \
--image-version composer-2.0.9-airflow-2.2.3 \
--network $VPC \
--subnetwork $SUBNET \
--web-server-allow-ip [description=<description>],[ip_range=<ip_address>]
```

Note: Here, `--web-server-allow-ip [description=<description>],[ip_range=<ip_address>]` is a repeatable flag and can be used to add multiple ip addresses.

5. Setup the Airflow DAG

- Open the file from the downloaded code repository at `customer_churn/00-scripts/variables.json` and edit the variables as per your environment
- Next, open the composer environment and navigate to **Environment Configuration>Airflow Web UI** to open the Airflow UI
- Once the Airflow UI opens, navigate to **Admin>Variables**



- Click on **Choose File** and select the file from the downloaded code repository at `customer_churn/00-scripts/variables.json`
- Click on **Import Variables**
- All the required variables will now be imported into Airflow