**Steps in Amazon Web Services (AWS)**

**Creating the terraform-en-1 user using the IAM service**

* Access the AWS console (<https://aws.amazon.com>) **and log in with your newly created account**. In the search bar, type IAM. In the Services section, click on IAM.
* Click on Users and then Add users, enter the name **terraform-en-1** and click Next to create a programmatic type user. A screenshot of a computer

  Description automatically generated
* After advancing, in **Set permissions**, click on the Attach existing policies directly button.
* A screenshot of a computer

  Description automatically generated
* Type **AmazonS3FullAccess** in **Search.**
* Select **AmazonS3FullAccess**

A screenshot of a computer

Description automatically generated4

* Click on **Next**
* Review all the details
* Click on **Create user**

**Creating the Access Key for the terraform-en-1 user using the IAM service**

* Access the **terraform-en-1** user

A screenshot of a computer

Description automatically generated

* Click on the **Security credentials** tab

A screenshot of a computer

Description automatically generated

* Navigate to the **Access keys** section
* Click on **Create access key**

A screenshot of a computer

Description automatically generated

* Select Command Line Interface (CLI) and **I understand the above recommendation and want to proceed to create an access key.**

A screenshot of a computer

Description automatically generated

* Click on **Next**.
* Click on **Create access key**

A screenshot of a computer

Description automatically generated

* Click on **Download .csv file**

A screenshot of a computer

Description automatically generated

* After the download finishes, click on Done.
* Once the download is complete, rename the **.csv** file to **key.csv**

**Steps in Google Cloud Platform (GCP)**

**Preparing the environment to run Terraform**

* Access the Google Cloud Console ([console.cloud.google.com](http://console.cloud.google.com/)) **and log in with your newly created account**
* Open the Cloud Shell
* A screenshot of a computer

  Description automatically generated
* Download the mission1.zip file in the Google Cloud shell using the wget command
* wget <https://tcb-public-events.s3.amazonaws.com/icp/mission1.zip>

**Result**

* A screenshot of a computer

  Description automatically generatedUpload the key.csv file to the Cloud Shell using the browser

**Step 1**

A screenshot of a computer

Description automatically generated**Step 2**

A screenshot of a computer

Description automatically generated**Step 3**

* A screenshot of a computer

  Description automatically generatedVerify if the mission1.zip and key.csv files are in the folder in the Cloud Shell using the command below
* ls

**Result**

* A black screen with white text

  Description automatically generatedExecute the file preparation commands:
* unzip mission1.zip
* mv key.csv mission1/en
* cd mission1/en
* chmod +x \*.sh

**Result**

* A black rectangular object with a red line

  Description automatically generatedExecute the commands below to prepare the AWS and GCP environment
* mkdir -p ~/.aws/
* touch ~/.aws/credentials\_multiclouddeploy
* ./aws\_set\_credentials.sh key.csv
* GOOGLE\_CLOUD\_PROJECT\_ID=$(gcloud config get-value project)
* gcloud config set project $GOOGLE\_CLOUD\_PROJECT\_ID
* Click on Authorize
* A screen shot of a computer

  Description automatically generatedExecute the command below to set the project in the Google Cloud Shell
* ./gcp\_set\_project.sh
* Execute the commands to enable the Kubernetes, Container Registry, and Cloud SQL APIs
* gcloud services enable containerregistry.googleapis.com
* gcloud services enable container.googleapis.com
* gcloud services enable sqladmin.googleapis.com
* gcloud services enable cloudresourcemanager.googleapis.com
* gcloud services enable serviceusage.googleapis.com
* gcloud services enable compute.googleapis.com
* gcloud services enable servicenetworking.googleapis.com --project=$GOOGLE\_CLOUD\_PROJECT\_ID

**Running Terraform to provision MultiCloud infrastructure in AWS and Google Cloud**

* + Execute the following commands to provision infrastructure resources

cd ~/mission1/en/terraform/

terraform init

terraform plan

terraform apply

**Attention**: The provisioning process can take between **15 to 25 minutes** to finish. Keep the **CloudShell** open during the process. If disconnected, click on **Reconnect** when the session expires (the session expires after **5 minutes** of inactivity by default)

**Appendix I - Destroying the environment and starting over**

In case you have encountered any problem/error and want to reset the environment to start over, follow the step-by-step instructions below to remove the entire MultiCloud environment.

**[Google Cloud] Delete VPC Peering**

A screenshot of a computer

Description automatically generated

**[Google Cloud] Delete remaining resources w/ Terraform - Cloud Shell**

cd ~/mission1/en/terraform/

terraform destroy

**Clean the Cloud Shell in AWS and Google Cloud**

**AWS**

cd ~

rm -rf mission\*

**Google Cloud**

cd ~

rm -rf mission\*

rm -rf .ssh

**Security Tips**

* + For production environments, it's recommended to use only the Private Network for database access.
  + Never provide public network access (0.0.0.0/0) to production databases. ⚠️

**By reaching this point, you have completed the implementation of the first part of the Hands-on Project and have implemented resources in a MultiCloud (AWS and Google Cloud) environment using Terraform!**

**Congratulations! 🚀🎉**