

Data Services Deployment Wizard: How-to Guide

Edited 2022-05-13

▼ On this page

Introduction

Before you start

Setup Wizard

Perform Single Service Deployment

- Step 1- Provide Environment Details
- Step 2- Store State Files in GCS bucket
- Step 3- Provide Git Repo Details
- Step 4- Choose a Service and Provide the Input Variables
- Step 5- View Logs and check the Deployment Status
- Step 6- View Terraform Code in the Git repo.
- Download the Terraform Code.

Fixing issues

Introduction

The aim of this guide is to help users get started with the Deployment of Individual Data Services using the wizard.

Deployment wizard is a turnkey deployment solution that enables a user to do the following-

- Setup the deployment wizard.
- Choose the deployment of services from various options
- Run deployment of services mentioned below parallelly from wizard
 1. Cloud Storage
 2. BigQuery Datasets
 3. Dataproc
 4. CloudSQL(MySQL and PostgreSQL)
 5. Cloud Composer
- Export Terraform templates to a repository like Github
- Run Terraform from deployment wizard
- Download Terraform templates for available services

Before you start

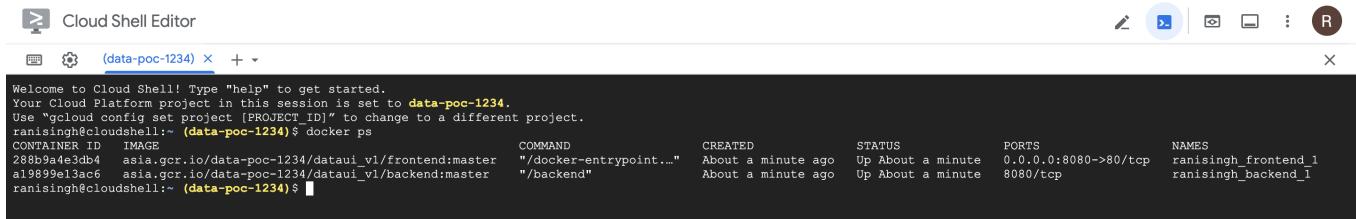
Make sure you have a GCP account and you have owner or equivalent access to a GCP organization. The solution runs on cloud shell and pulls all the information such as Organisation and project details. As a prerequisite make sure gcloud commands are running from the cloudshell.

As a sanity check just try to do a `gcloud organizations list` in cloudshell to make sure you have access.

Setup Wizard

Once in cloudshell run the following commands to bring up the wizard: `export LD_LIBRARY_PATH=/usr/local/lib shell gsutil cp gs://atc-data-acc/docker-compose.yaml ./; docker-compose up -d`

Point to Note: To check if the containers are running run the below command in Cloudshell.
`docker ps`



```

Cloud Shell Editor
(data-poc-1234) x + v

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to data-poc-1234.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
ranisingh@cloudshell:~ (data-poc-1234)$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS              PORTS          NAMES
288b9ade3db4   asia.gcr.io/data-poc-1234/dataui_v1/frontend:master   "/docker-entrypoint..."   About a minute ago   Up About a minute   0.0.0.0:8080->80/tcp   ranisingh_frontend_1
a1989e13ac6   asia.gcr.io/data-poc-1234/dataui_v1/backend:master    "/backend"               About a minute ago   Up About a minute   8080/tcp           ranisingh_backend_1
ranisingh@cloudshell:~ (data-poc-1234)$

```

If it's not running then redeploy the Wizard.

Perform Single Service Deployment

The wizard provides you option to select Data Services from the available list. Pick the one which is relevant for your use-case. Once selected, the wizard will help in deploying these services and push the terraform code to git repo.

Post the collection of the required details, you can choose 'Generate TF Code' to Download the TF Templates for each service. Once the Generate TF Code option has been selected for required services, Click the 'Download Template' to download a zip file which contains Terraform code for the selected services.

Here are the guided steps which shows the journey to Deploy Data Services using this wizard-

Step 1- Provide Environment Details

Provide details such as Organization and Project

Workload Deployment Wizard

Data Service Deployment

Environment Details

Choose an existing project or create one

Organizations *
List of organizations you are part of: rani.joonix.net

Projects List *
List of projects you are part of: data-poc-1234

Store State Files in GCS

Repository Details

GCP Cloud Storage

BigQuery Datasets

Dataproc

CloudSQL

Cloud Composer V2

Support

Step 2- Store State Files in GCS bucket

Provide details of Existing GCS bucket where state files will be stored. Click on Next.

Workload Deployment Wizard

Data Service Deployment

Environment Details

Store State Files in GCS

Existing Bucket *
datapoc-gcsbq

Folder Path prefix *
gcs1

Repository Details

GCP Cloud Storage

BigQuery Datasets

Dataproc

CloudSQL

Cloud Composer V2

Support

Every GCP customer automatically gets free support that includes support product documentation, community support, and support for billing issues. However, we recommend that enterprise

Step 3- Provide Git Repo Details

Provide details of Git repo where generated terraform code will be pushed. Either an existing Repo name can be provided or provide a new Repo name and the wizard will create it. Click on Next to proceed.

Workload Deployment Wizard

Data Service Deployment

Environment Details

Store State Files in GCS

Repository Details

Github

Github Username * raniksingh API Key * Repo Name * dataservices

Github repository dataservices created successfully

Next

GCP Cloud Storage

BigQuery Datasets

Dataproc

CloudSQL

Cloud Composer V2

Support

Step 4- Choose a Service and Provide the Input Variables

Choose a Service which you want to deploy and fill in the input variables. click on Deploy.

Workload Deployment Wizard

Environment Details

Store State Files in GCS

Repository Details

GCP Cloud Storage

New Bucket Name * atcdatatest Doc Link

Storage Class * Standard Storage Bucket Location * us-central

Service Account datapocatest@data-poc-1234.iam.gserviceaccount.com

Deploy Generate TF Code

Logs

BigQuery Datasets

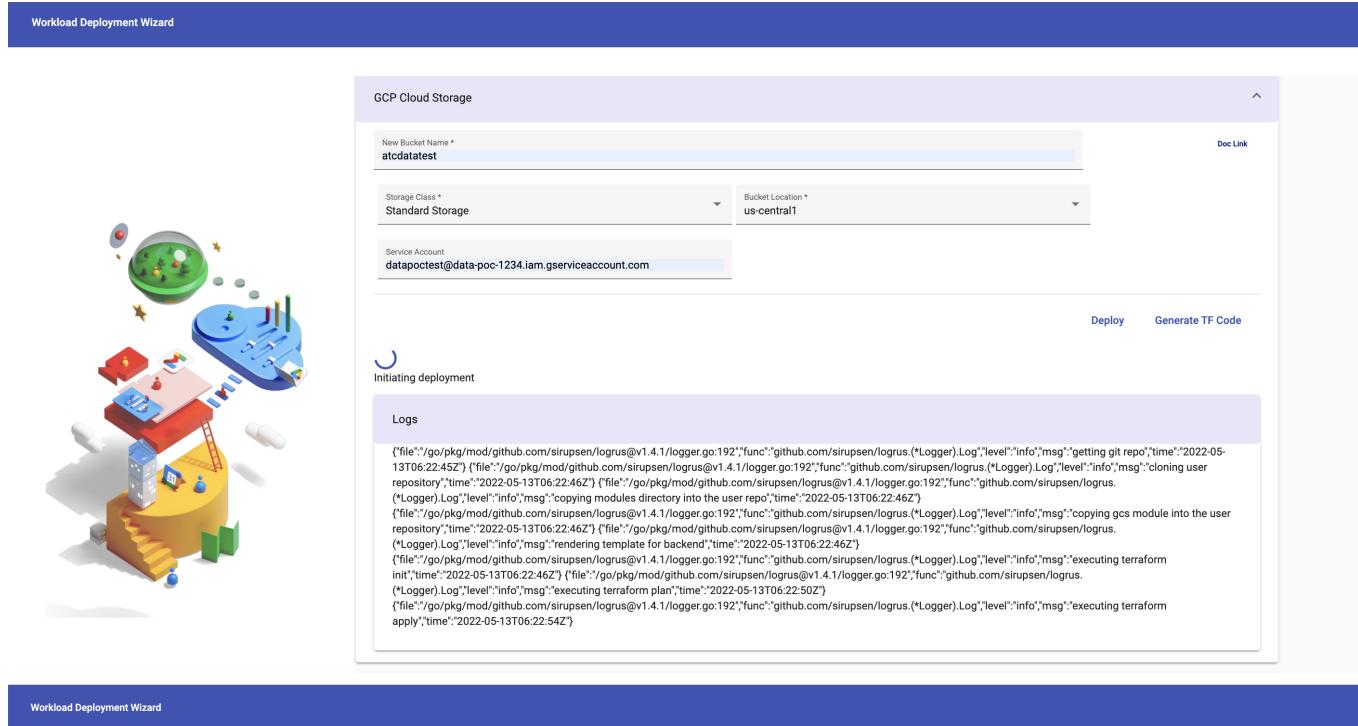
Dataproc

CloudSQL

Cloud Composer V2

Step 5- View Logs and check the Deployment Status

View the Deployment status and click on Logs button to view Logs.

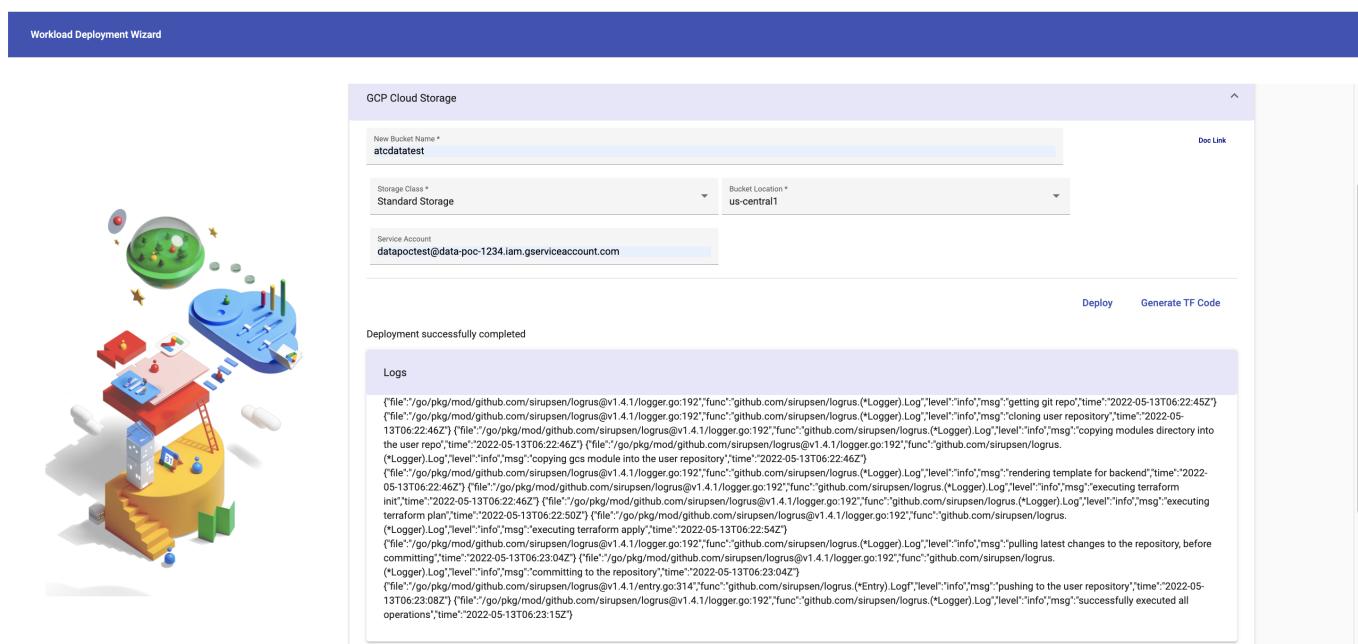


The screenshot shows the 'GCP Cloud Storage' section of the 'Workload Deployment Wizard'. The 'New Bucket Name' field contains 'atcdatatest'. The 'Storage Class' is set to 'Standard Storage' and the 'Bucket Location' is 'us-central1'. A service account 'datapocatest@data-poc-1234.iam.gserviceaccount.com' is selected. The 'Logs' section displays deployment logs, including messages about cloning user repositories, rendering templates, executing terraform init, and applying changes. The status bar at the bottom indicates 'Initiating deployment'.

```

Logs
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"getting git repo",time:"2022-05-13T06:22:45Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"cloning user repository",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"cloning user repository",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"copying modules directory into the user repo",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"copying gcs module into the user repository",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"copying modules directory into the user repository",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"executing terraform init",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"executing terraform init",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"executing terraform plan",time:"2022-05-13T06:22:50Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"executing terraform apply",time:"2022-05-13T06:22:54Z"}

```



The screenshot shows the 'GCP Cloud Storage' section of the 'Workload Deployment Wizard'. The 'New Bucket Name' field contains 'atcdatatest'. The 'Storage Class' is set to 'Standard Storage' and the 'Bucket Location' is 'us-central1'. A service account 'datapocatest@data-poc-1234.iam.gserviceaccount.com' is selected. The 'Logs' section displays deployment logs, including messages about cloning user repositories, rendering templates, executing terraform init, and applying changes. The status bar at the bottom indicates 'Deployment successfully completed'.

```

Logs
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"getting git repo",time:"2022-05-13T06:22:45Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"cloning user repository",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"cloning user repository",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"copying modules directory into the user repo",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"copying gcs module into the user repository",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"copying modules directory into the user repository",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"executing terraform init",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"executing terraform init",time:"2022-05-13T06:22:46Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"executing terraform plan",time:"2022-05-13T06:22:50Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"executing terraform apply",time:"2022-05-13T06:22:54Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"pulling latest changes to the repository before committing",time:"2022-05-13T06:23:04Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"committing to the repository",time:"2022-05-13T06:23:04Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/entry.go:314",func:"github.com/sirupsen/logrus.(Entry).Log",level:"info",msg:"pushing to the user repository",time:"2022-05-13T06:23:08Z")
(file:"/go/pkg/mod/github.com/sirupsen/logrus@v1.4.1/logger.go:192",func:"github.com/sirupsen/logrus.(*Logger).Log",level:"info",msg:"successfully executed all operations",time:"2022-05-13T06:23:15Z"}

```

Step 6- View Terraform Code in the Git repo.

View the Deployment status and click on Logs button to view Logs.

The screenshot shows a GitHub repository page for 'Raniksingh / dataservices'. The 'Code' tab is selected. At the top, it shows 'master' branch, '1 branch', '0 tags', and buttons for 'Go to file', 'Add file', and 'Code'. Below this, a commit history table is displayed:

ATC team Adding base files		55e99f2 2 minutes ago	⌚ 2 commits
modules	Adding base files	2 minutes ago	
service-gcs	Adding base files	2 minutes ago	
Readme.md	commit to init the repo	5 minutes ago	

Download the Terraform Code.

To download the Terraform code click on Generate TF Code for required service. Then at the end click on 'Download Generated TF Code'.

Workload Deployment Wizard

New Bucket Name * **atcdatatest**

Storage Class * Standard Storage

Bucket Location * us-central1

Service Account datapocetest@data-poc-1234.iam.gserviceaccount.com

Deploy Generate TF Code

Generating tf code

Logs

BigQuery Datasets

Dataproc

CloudSQL

Cloud Composer V2

Download Generated TF Code

Support

Every GCP customer automatically gets free support that includes support product documentation, community support, and support for billing issues. However, we recommend that enterprise customers sign up for a premium support plan, which offers one-on-one technical support with Google support engineers.

Workload Deployment Wizard

New Bucket Name * **atcdatatest**

Storage Class * Standard Storage

Bucket Location * us-central1

Service Account datapocetest@data-poc-1234.iam.gserviceaccount.com

Deploy Generate TF Code

TF Code generated successfully, you can now download the generated code using the download button

Logs

BigQuery Datasets

Dataproc

CloudSQL

Cloud Composer V2

Download Generated TF Code

Support

Every GCP customer automatically gets free support that includes support product documentation, community support, and support for billing issues. However, we recommend that enterprise customers sign up for a premium support plan, which offers one-on-one technical support with Google support engineers.

Fixing issues

In case of any queries or if you run into issues please reach out to gdc-atc@google.com