Preparing to install Data Fabric on TechZone AWS, Azure and IBM Cloud Environments

This document will walk you through the steps on how to prepare your environment to demonstrate the Data Fabric on TechZone. This will include various steps such as

How to request a demo environment, **Logging into your cloud of choice**, and **Step by Step guide for running the Automation**.

Goals for the Demo:

- Prepare your TechZone environment to demo Data Fabric on AWS, Azure and IBM Cloud which are supported with ROSA, ARO and ROKS OpenShift clusters.
- Configuration steps for Data Fabric including enabling license key and your first target environment to monitor.

Prerequisites:

- A valid IBM ID that can be used to access
  - TechZone https://techzone.ibm.com/
- A valid GitHub ID that can be used to create a repository in your own organization
  - GitHub https://github.com/
- Install a code editor, we recommend **VSCode**
  - VS Code https://code.visualstudio.com/
- Install a **Colima** a replacement for Docker Desktop https://github.com/abiosoft/colima
  - brew install Colima

Obtaining a License Key:

To use Data Fabric, you are required to install a license key. For Proof of Concepts, IBM Partners and IBMers can obtain it using the steps below.

Partners
For Partners follow these steps:

You must have your IBM entitlement API key to access images in the IBM Entitled Registry.

After you purchase Cloud Pak for Data, an entitlement API key for the software is associated with your My IBM account. You need this key to complete the Cloud Pak for Data installation. To obtain the entitlement key, complete the following steps:

1. Log in to Container software library on My IBM with the IBM ID and password that are associated with the entitled software.
2. On the Get entitlement key tab, select Copy key to copy the entitlement key to the clipboard.
3. Save the API key in a text file.

IBMers
IBMers, can download the entitlement key using these steps:

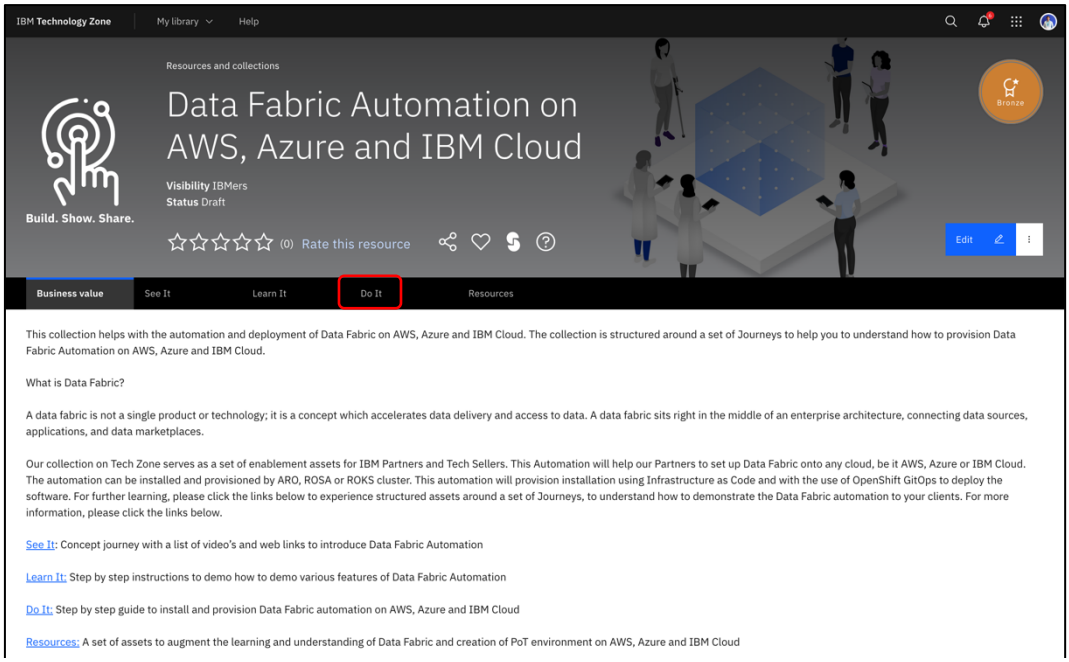
1. Log in to Container software library on My IBM with the IBM ID and password that are associated with the entitled software.
2. On the Get entitlement key tab, select Copy key to copy the entitlement key to the clipboard.
3. Save the API key in a text file.

For more info refer, https://www.ibm.com/docs/en/cloud-paks/cp-data/4.0?topic=tasks-obtaining-your-entitlement-api-key

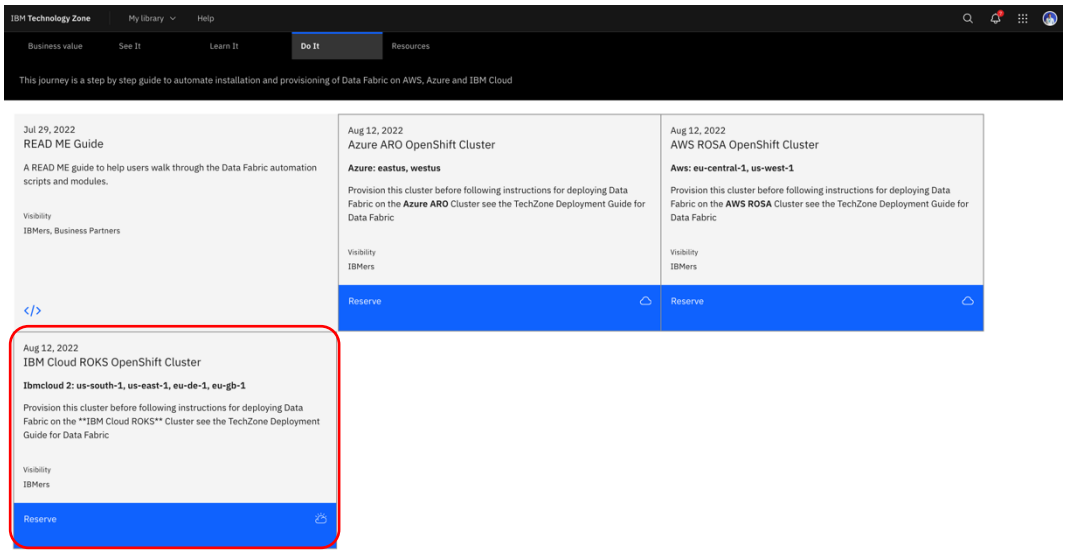
Requesting TechZone Environment

Demo Steps:

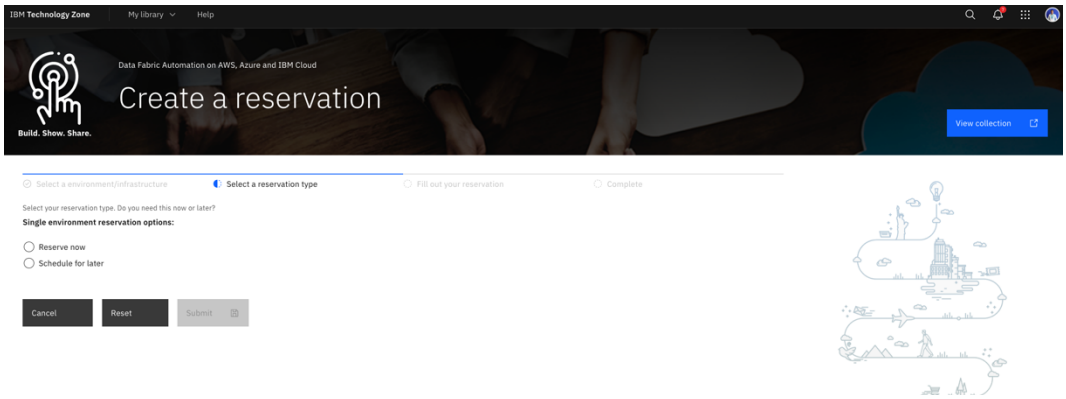
1. If you have not already done so, access the Tech Zone collection for Data Fabric Automation for AWS, Azure, and IBM Cloud
    a. https://techzone.ibm.com/collection/data-fabric-automation-on-aws-azure-and-ibm-cloud



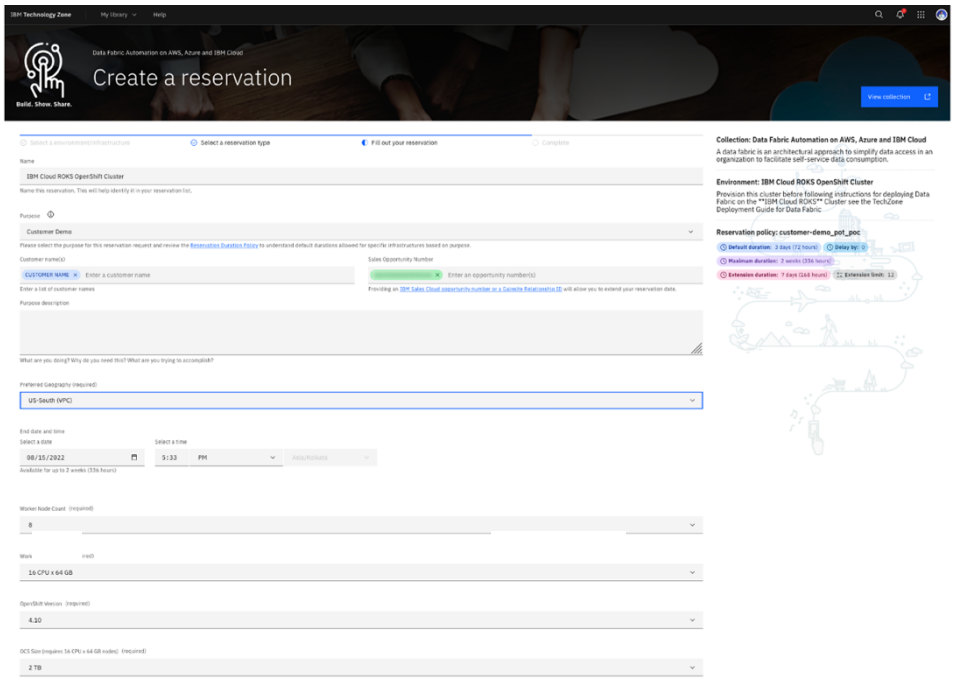
2. The first thing you need to do is to request an access to the demo environment. Click on **Do It** Tab



3. Click on the cloud type of your choice. For this guide we will use **IBM Cloud ROKS OpenShift Cluster** environment

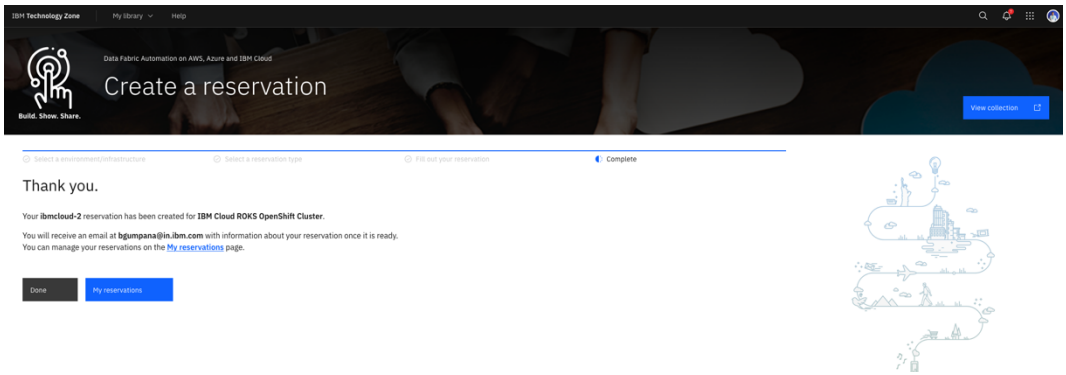


4. Click on the **Reserve now** radio button



5. Complete the Reservation Type form. Please make sure all the fields are complete, you may want to select **Customer Demo** as the Purpose of your request
    a. As these are real AWS, Azure, and IBM Cloud environment, you will need to use a real customer opportunity value from **IBM Sales Cloud** https://w3.ibm.com/w3publisher/ibm-sales-cloud

6. Enter the **Sales Opportunity Number** as you will not be able to proceed without a valid and live opportunity
7. Select your duration for the environment, and for the size for Data Fabric keep it as a **8 Worker Node Count** and **16 CPU x 64 GB** worker node flavor
8. Select the OCS Size as **2 TB**
9. Click on **Submit** once all the fields are entered correctly
10. Once the **Submit** has completed you will see the following screen with a THANK YOU message to confirm that your reservation is being processed

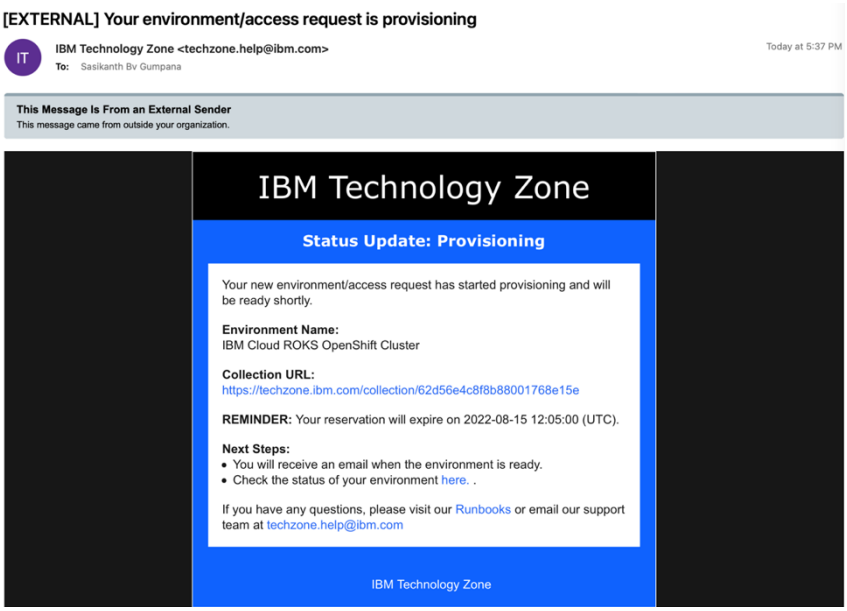


11. Check your email for confirmation.

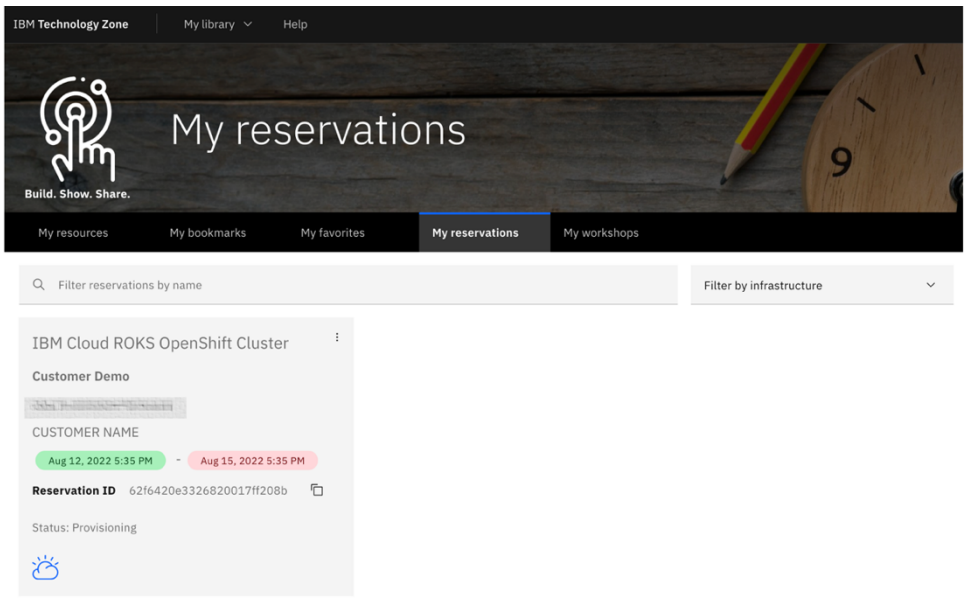
Receive Confirmation Email Log in for the First Time

Steps:

1. Once you have reserved the demo environment, you will receive the confirmation email.



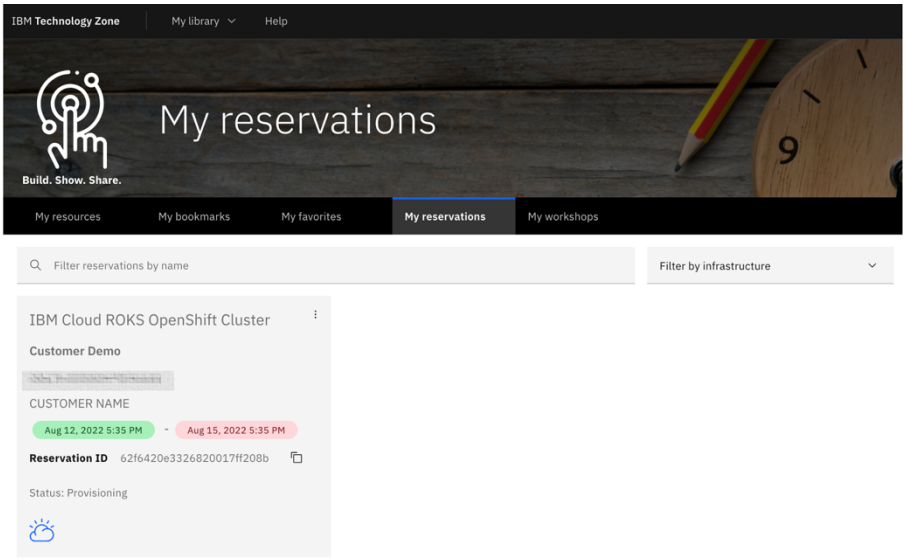
2. It is important that you read the email to follow the instructions on how to log into the Tech Zone Demo Account.
3. You will receive a second email once your reservation has been processed; Click on **My reservations** to see the status of your reservation.



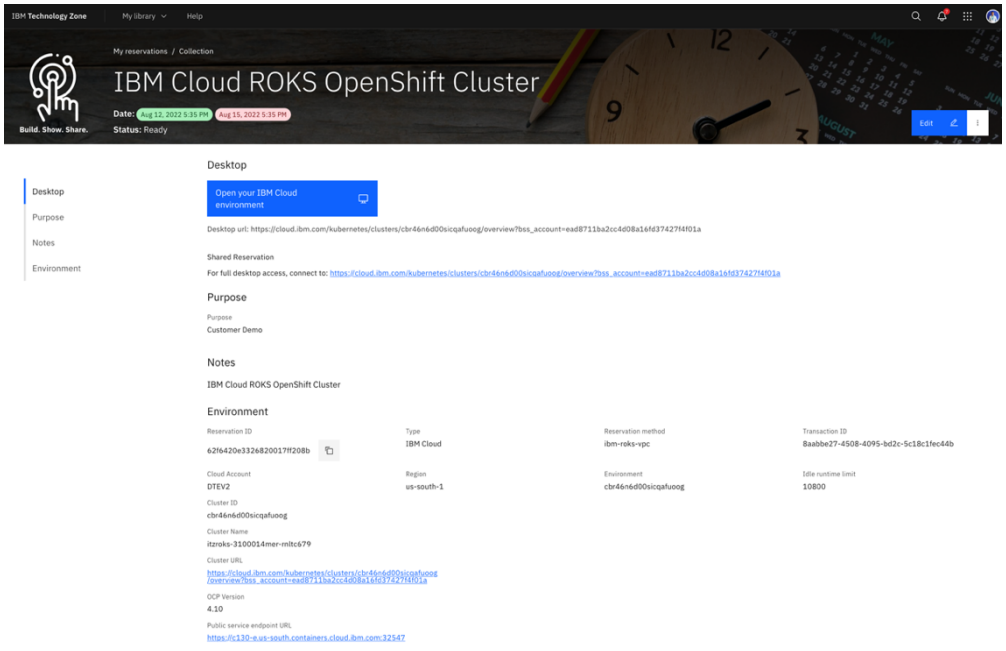
Log into your TechZone Environment

Steps:

1. Navigate to **TechZone** and **Your Reservations**, click on the reservation you created in the previous steps.

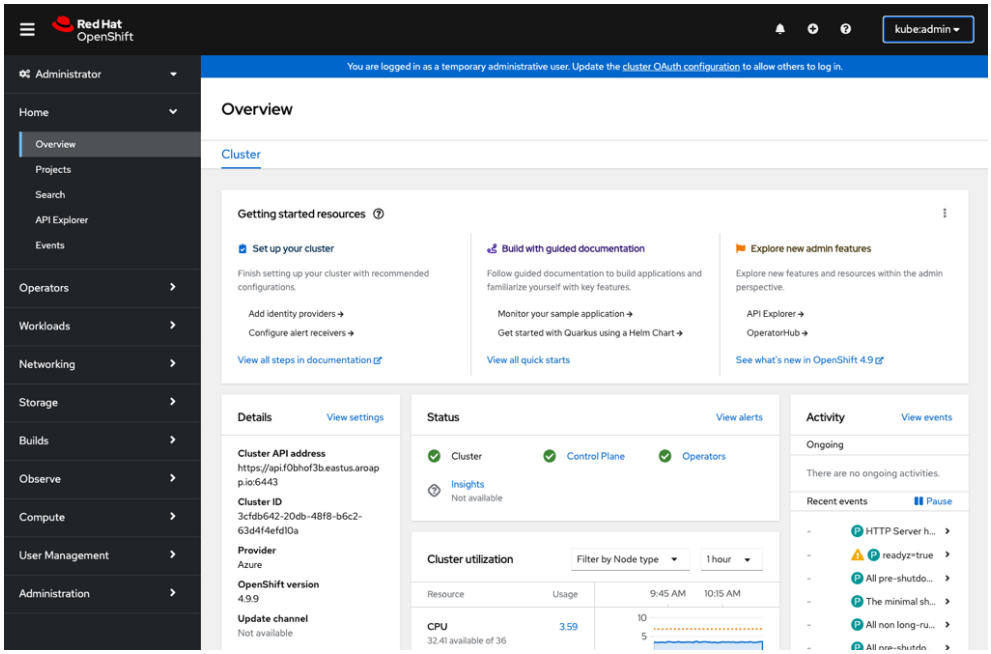


2. You will see the reservation view in detail, click on the **Open your IBM Cloud environment** button.



3. This will display the OpenShift login screen for the cluster, enter the user ID and password details from the previous reservation screen. Typically, it is **kubeadmin** for the user id and the system generated password.

4. Once you have logged in you will see the main OpenShift administration screen



5. You have successfully logged into your TechZone environment. You can now start the installation process

Installing Data Fabric into your TechZone environment

The installation process will use standard Terraform git repository that has been built using the modules you need to make Data Fabric installation consistent across the three cloud environments AWS, Azure, and IBM Cloud.

Steps:

1. First step is to clone the automation code to your local machine. Run this git command in your favorite command line shell.

   $ git clone https://github.com/IBM/automation-data-fabric

2. Navigate into the **automation-data-fabric** folder using your command line.
   a. The **README.md** has a comprehensive instruction on how to install this into cloud environments other than TechZone. This document focuses on getting it running in a TechZone requested environment.
3. The first step is setup your **credentials.properties** file. This will enable a secure access to your cluster.

   $ cp credentials.template credentials.properties
   $ code credentials.properties

   # Add the values for the Credentials to access the IBM Cloud
   # Instructions to access this information can be found in the README.MD
   # This is a template file and the ./launch.sh script looks for a file based on this template named credentials.properties
   TF_VAR_gitops_repo_username=
   TF_VAR_gitops_repo_token=
   TF_VAR_cluster_login_token=
   TF_VAR_server_url=

   # AWS Credentials are required to Create AWS S3 bucket and upload Datafiles to the S3 Bucket (https://github.com/IBM/automation-data-fabric/tree/main/610-datafabric-setup/terraform/Datafiles)
   TF_VAR_access_key=
   TF_VAR_secret_key=

4. You will need to populate these values. Add your **Git Hub** username and your Personal Access Token to **repo_username** and **repo_token**
5. From your **OpenShift console** click on top right menu and select **Copy login command** and click on **Display Token**

6. Copy the **API Token** value into the **login_token** value
7. Copy the **Server URL** into the **server_url** value, only the part starting with **https**
8. You need to make sure you are not running **Docker Desktop** as this is not currently allowed under the new terms and conditions. You will need to install **Colima** as an alternative

```
$ brew install colima
$ colima start
```

9. We are now ready to install Data Fabric, run the **launch.sh** command, make sure you are in the root of the **automation-Data Fabric** repository

```
$ ./launch.sh
Cleaning up old container: cli-tools-WljCg
Initializing container cli-tools-WljCg from
quay.io/cloudnativetoolkit/terraform:v1.1
Attaching to running container...
/terraform $
```

10. **Launch.sh** will download a container image that contains all the command line tools to enable easy installation of the software. Once the download is complete, , it will mount the local file system and exec into the container for you to start running commands from within this custom container.
11. Next step is to create a workspace to run the Terraform automation.
12. Run the command **setup-workspace.sh**

```
$ sudo ./setup-workspace.sh
```

13. The default **terraform.tfvars** file is symbolically linked to the new workspaces folder. This is to enable you to edit the file in your native operating system using editor of your choice.
14. Edit the default **terraform.tfvars** file to enable you to setup the GitOps parameters.
    $ sudo vi terraform.tfvars

    ##
    ######
    # Name: Data Fabric Terraform Variable File
    # Desc: Initial input variables to support installation of Data Fabric into the cloud provider of your choice
    ##
    ######

    ## gitops-ocp-datafabric_storage_class_name: Name of the block storage class to use - if multizone deployment then waitforfirstconsumer must be set on storageclass binding mode
    gitops-ocp-datafabric_storage_class_name="<your block storage on aws: gp2, on azure: managed-premium>"

    ## gitops-repo_host: The host for the git repository.
    gitops_repo_host="github.com"

    ## gitops-repo_type: The type of the hosted git repository (github or gitlab).
    gitops_repo_type="github"

    ## gitops-repo_org: The org/group where the git repository exists/will be provisioned.
    gitops_repo_org="<your gitorg - most likely your username>"

    ## gitops-repo_repo: The short name of the repository (i.e. the part after the org/group name)
    gitops_repo_repo="<repo name to create for git ops configuration>"

    ## gitops-cluster-config_banner_text: The text that will appear in the top banner in the cluster
    gitops-cluster-config_banner_text="Software Everywhere Data Fabric"

15. For IBM Cloud, you will need to run some automation to configure Storage for that environment If we are on Azure or AWS, change the storage_class_name value to **managed_premium** for Azure and other values for AWS.
16. You will see that the repo_type and repo_host are set to GitHub, you can change these to other Git Providers, like Git Hub Enterprise or GitLab.

17. For the repo_org value, set it to your default org name, or specific a custom org value. This is the organization that the GitOps Repository will be created in. Click on top right menu and select **Your Profile** to take you to your default organization.
18. Set the repo_repo value to a unique name for you to recognize as the place where the GitOps configuration is going to be placed before Data Fabric is installed into the cluster.
19. You can change the Banner text to something useful for you client project or demo.
20. Once in the container, check out the file system with **ls -al** command to see the workspace directory is now configured for you.
21. Navigate into the /workspaces/current folder, this is where your automation is configured.
22. Navigate into the **200** folder and run the following commands

    $ cd 200-openshift-gitops
    $ sudo terraform init

    $ sudo terraform apply --auto-approve
    .....
    $ Apply complete! Resources: 78 added, 0 changed, 0 destroyed.

23. This will kick off the automation for setting up the GitOps Operator into your **TechZone** cluster.

24. You can check the progress by looking at two places, first look in your git hub repository. You will see the git repository has been created based on the name you have provided. The Data Fabric install will populate this with information to let OpenShift GitOps install the software. The second place to look is the OpenShift console, Click **Workloads->Pods** and you will see the GitOps operator being installed.

25. Now GitOps is installed in the cluster. we have bound the git repository to OpenShift GitOps operator. We are now ready to populate this with some Software configuration that will cause OpenShift GitOps to install the software into the cluster. Navigate into **210** folder,

    a. If you are using **Techzone's IBM Cloud ROKS OpenShift Cluster (VPC), OCS** is already configured. You can skip this Step.

    b. If you are using on **AWS** run the Navigate into the **210** folder and run the following commands.

       $ cd 210-aws-portworx-storage
       $ sudo terraform init
       $ sudo terraform apply --auto-approve

    c. If you are using on **Azure** run the Navigate into the **210** folder and run the following commands.

       $ cd 210-azure-portworx-storage
       $ sudo terraform init
       $ sudo terraform apply --auto-approve

26. Navigate into the **300** folder and run the following commands, this will setup the Entitlement pull-secret in the cluster.

    $ cd 300-cloud-pak-for-data-entitlement
    $ sudo terraform init
    $ sudo terraform apply --auto-approve
    .....
    $ Apply complete! Resources: XX added, 0 changed, 0 destroyed.

27. Navigate into the **305** folder and run the following commands, this will install Cloud pak for Data in the cluster.

    $ cd 305-cloud-pak-for-data-foundation
    $ sudo terraform init
    $ sudo terraform apply --auto-approve
    .....
    $ Apply complete! Resources: XX added, 0 changed, 0 destroyed.

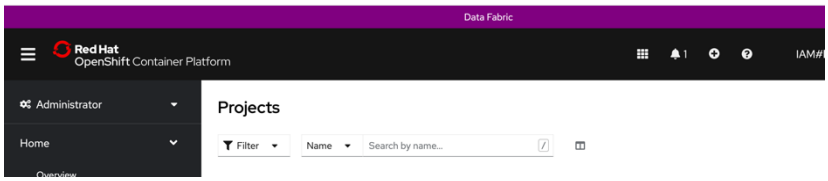
28. Navigate into the **600** folder and run the following commands, this will install services required for Data Fabric (**Watson Knowledge Catalog, Watson Studio, Watson Machine learning, Data Virtualization Service, Data Virtualization provision)** on top of Cloud pak for Data.

    cd 600-datafabric-services-odf
    $ sudo terraform init
    $ sudo terraform apply --auto-approve
    .....
    $ Apply complete! Resources: XX added, 0 changed, 0 destroyed.

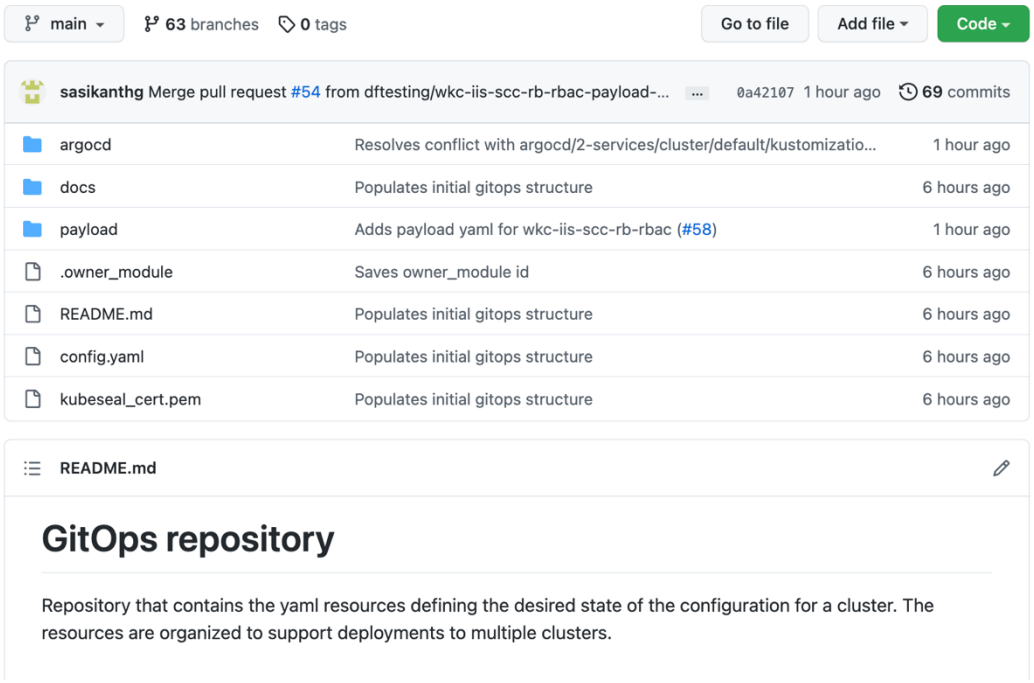
29. Navigate into the **610** folder and run the following commands, this will
    a. Create AWS S3 Bucket
    b. Upload Datafiles to AWS S3 Bucket
    c. Configure Data Fabric Solution on top of Cloud pak for Data

    cd 610-datafabric-demo
    $ sudo terraform init
    $ sudo terraform apply --auto-approve
    .....
    $ Apply complete! Resources: XX added, 0 changed, 0 destroyed.

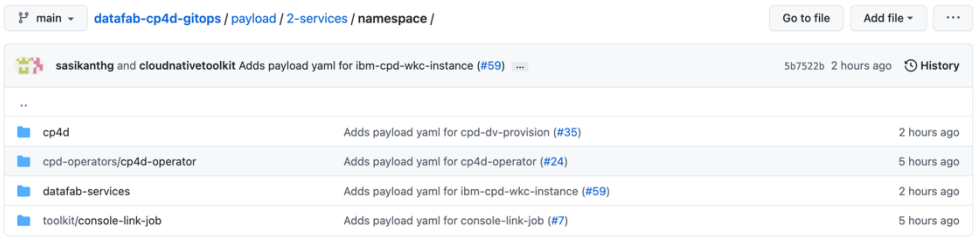
30. Once the installation has finished you will see a message from Terraform defining the state of the environment.
31. You will see the first change with a purple banner describing what was installed



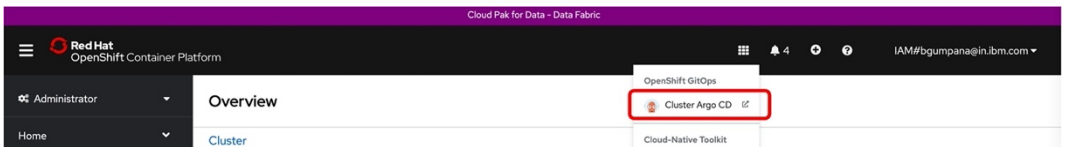
32. The next step is to validate everything installed correctly. Open your git repository where your git ops configuration was defined.



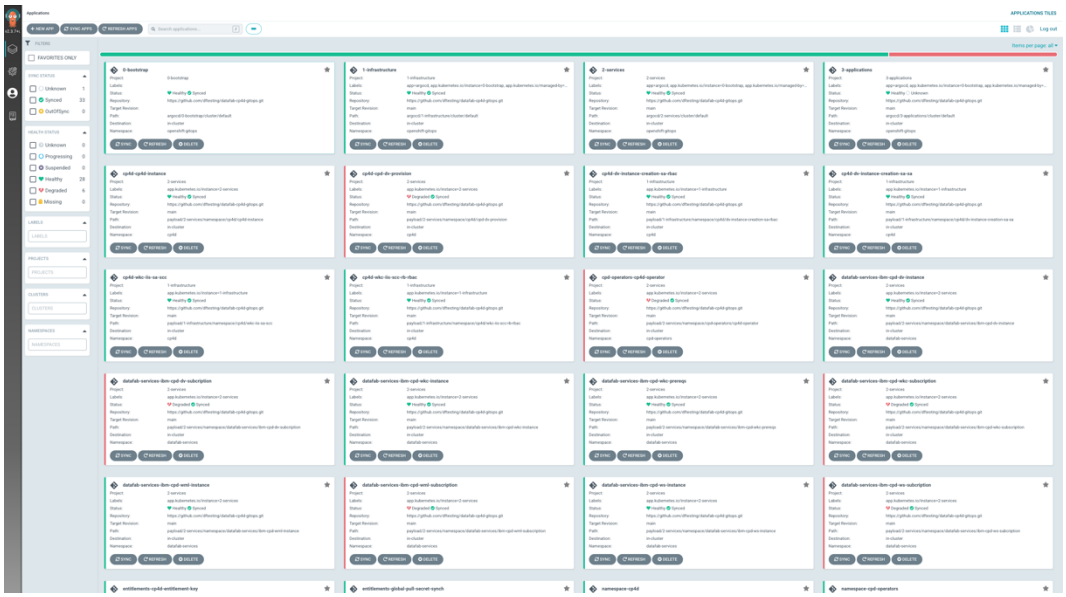
33. Check if the **payload** folder has been created with the correct definitions for GitOps. Navigate to the **payload/2-services/namespace/Data Fabric** folder and look at the content of the installation YAML files.



34. You should see the **Operator** CR definitions
35. Final Step is to Open up **Argo CD** (OpenShift GitOps) check if it is correctly configured, click on the Application menu 3x3 Icon on the header and select **Cluster Argo CD** menu item.



36. Complete the authorization with OpenShift and then narrow the filters by selecting the **CPD** or **Data Fabric** namespace.



37. This will show you the GitOps dashboard of the software you have installed using GitOps techniques
38. Click on **datafabric** tiles
39. You will see all the microservices that Data Fabric uses to install and their enablement state

THIS CONCLUDES THE GITOPS INSTALLATION STEPS

Data Fabric Solution after installation in TechZone Cluster

Now the Cloud Pak for Data, Services (WKC,WS, WML, DV, DV Provision) and Data Fabric configuration is completed.

Verification Steps:

1. In the **OpenShift** console navigate to the **Networking->Routes** and change the project to **CP4D,** you will see the route to launch **CPD Console**. Click on the URL to open **CPD Console.**
2. Login to CPD console with "**admin**" as username and get the password from **Workloads->Secrets** "**admin-user-details**" in CP4D namespace.
3. Review the below steps in CPD console to verify Data Fabric Setup & Configuration.

    a) Go to Hamburger menu -> Administration->Access Control. Check Data Scientist, Data Steward and Data Engineer users are created with their specific roles.
    b) Go to Hamburger menu -> Data -> Platform Connections -> aws-conn. Check and test the connection.
    c) Go to Hamburger menu -> Projects -> All Projects and check Data Fabric Analytics project is created.
    d) Go to Hamburger menu -> Services->Instances-> DV instance and check configuration for Virtualize files from DV and Create Join.
    e) Go to Hamburger menu -> Governance. Check Category, Classification, Business Term, Reference Data, Data Class, Data Protection Rule are created.
    f) Go to Hamburger menu -> Administration-> Catalogs. Check Connection to Catalog, Connected asset to Catalog and users added to Catalog
    g) Check Auto AI Experiment is created (Create Space, Create Pipeline, Create Data Asset, Create & Upload Attachment, Create Training and Deploy Model)

THIS CONCLUDES SETUP OF DATA FABRIC SOLUTION