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

This document will walk you through the steps of how to prepare your environment to demonstrate the Turbonomic on TechZone. This will include walking you through how to request a demo environment. Logging into your cloud of choice. Step by Step guide for running the Automation.

Goals for the Demo:

- Prepare your TechZone environment to demo Turbonomic AWS, Azure and IBM Cloud are supported with ROSA, ARO and ROKS OpenShift clusters.
- Configuration steps for Turbonomic 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 License Key:

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

Partners
For Partners follow these steps:

1. For PoCs/PoTs, Partners can download a license key from **PartnerWorld** https://www.ibm.com/partnerworld/public
2. You can search the software catalog for **M05C4EN IBM Turbonomic Application Resource Management On-Prem 8.4.6 for install on Kebernetes English**,
3. Download the package contains license file for Turbonomic, with a name similar to ***CP4MCM_IBM_ARM_OEM_Premier_License_July_2022.lic***
5. This file is covered by Turbonomic ARM P/N are currently available under IBM PPA terms and conditions

IBMers
For IBMers you can download a license key using these steps:

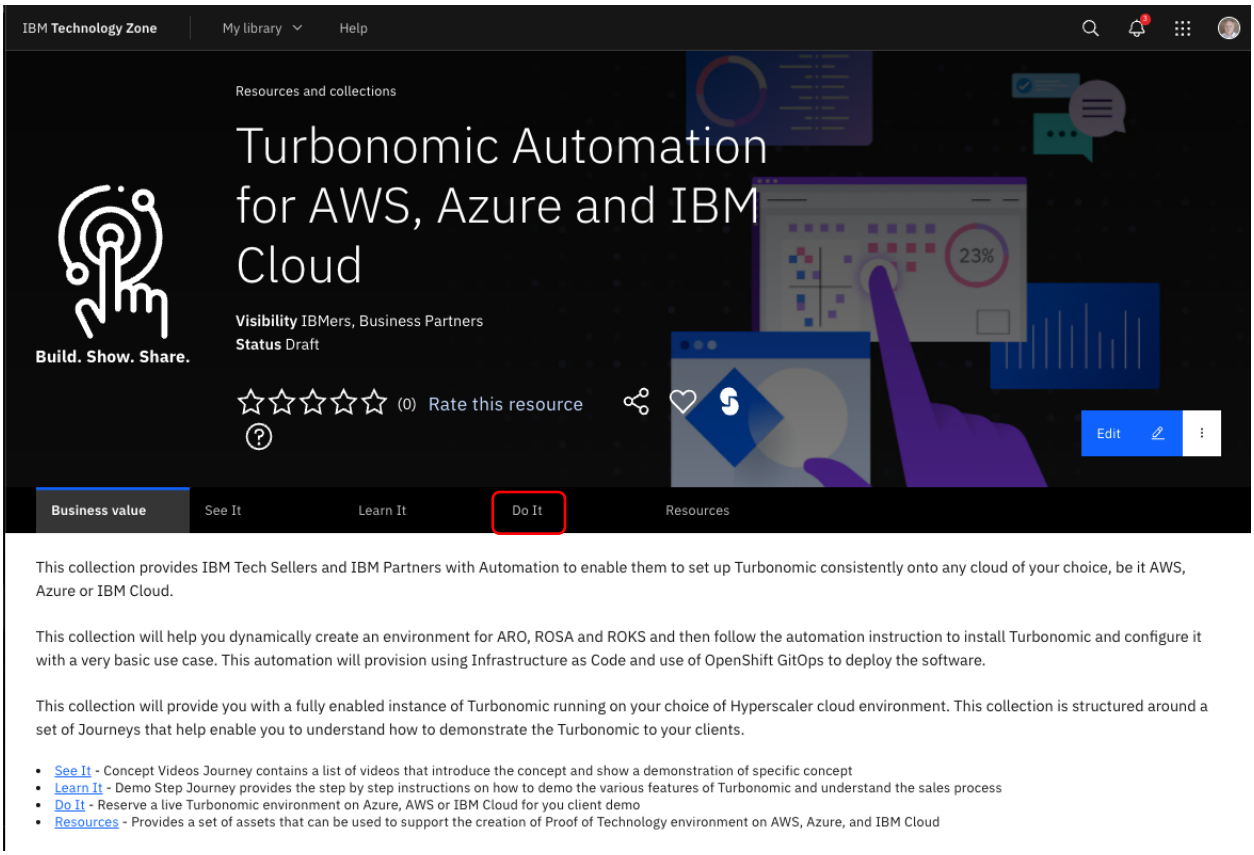
1. Go to XL Leverage https://w3-03.ibm.com/software/xl/download/ticket.wss
2. Search with keyword: turbonomic

3. Select the package **M05C4EN IBM Turbonomic Application Resource Management On-Prem 8.4.6 for install on Kebernetes English** and download
4. Extract this download package to get the turbonomic license key
   This package contains license file for turbonomic, with name similar to **CP4MCM_IBM_ARM_OEM_Premier_License_July_2022.lic**

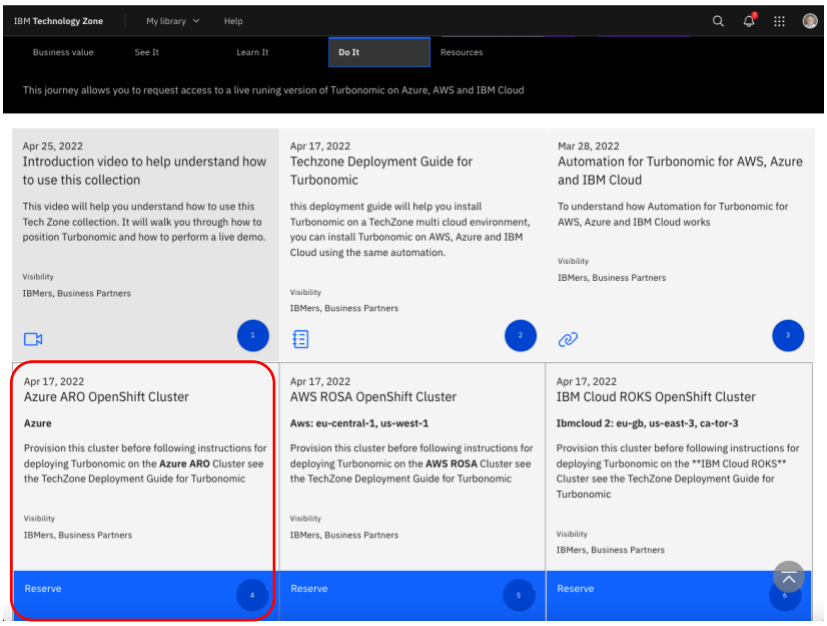
Requestion TechZone Environment

Demo Steps:

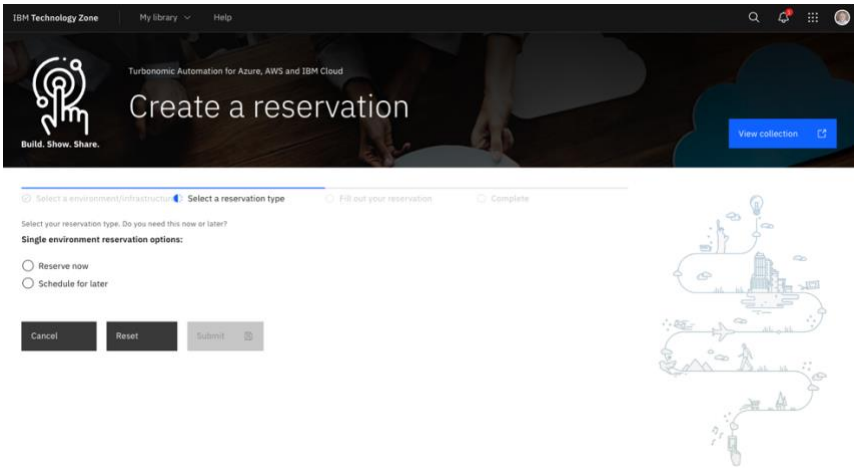
1. If you have not already done so, access the Tech Zone collection for Turbonomic Automation for AWS, Azure and IBM Cloud
    a. https://techzone.ibm.com/collection/turbonomic-automation-multicloud



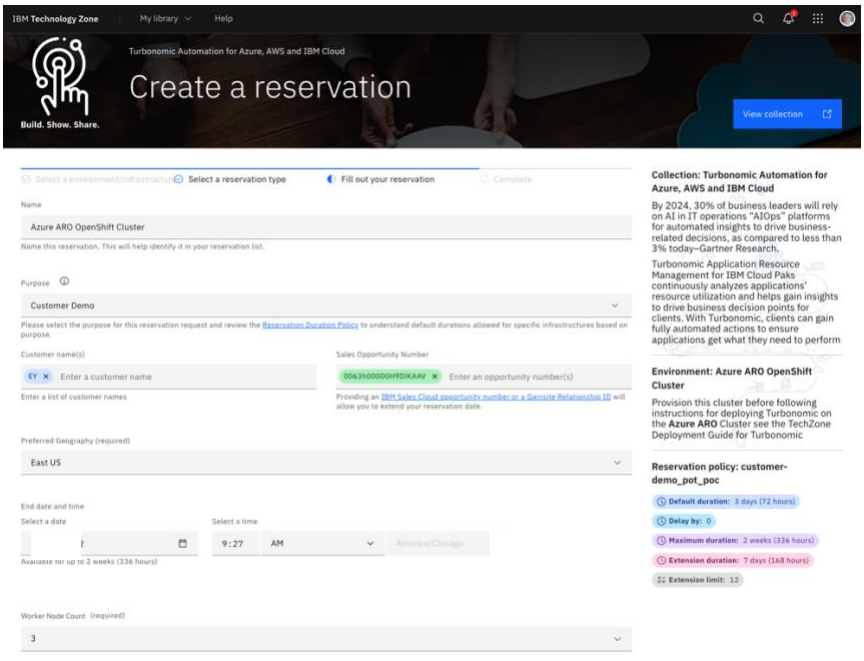
2. The first thing you need to do is request access to the demo environment. Click on **Do It** Tab



3. Click on the cloud type of your choice for this guide we will use **Azure ARO OpenShift Cluster** environment

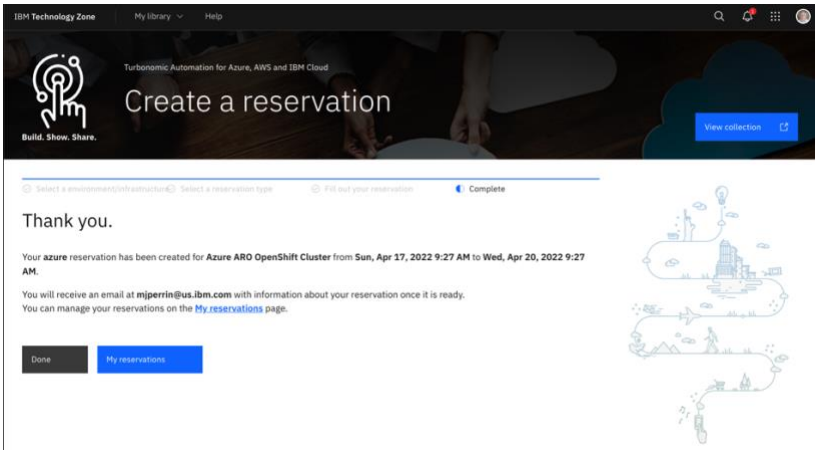


4. Click on the **Reserve now** radio button



5. Complete the Reservation Type form 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 environments 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** you will not be able to proceed unless this is a valid and live opportunity
7. Select your duration for the environment and the size for Turbonomic keep it as a **3 Worker Node Count** and **4 CPU x 16GB** worker node flavor
8. Click on **Submit** once all the fields are entered correctly
9. Once the **Submit** has completed you will see the following screen, this is confirming that your reservation is being processed

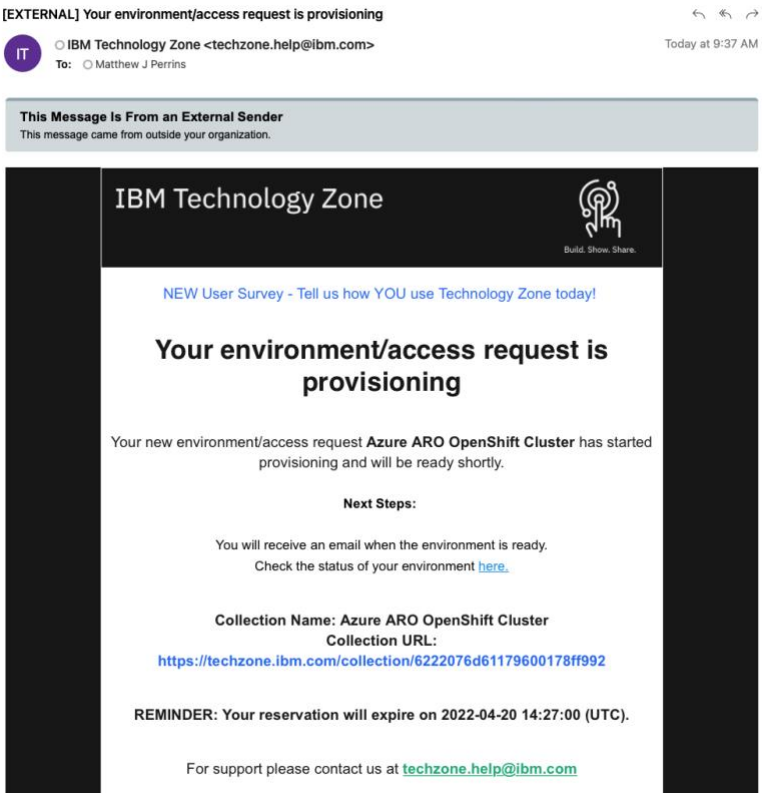


10. Check your email for confirmation email.

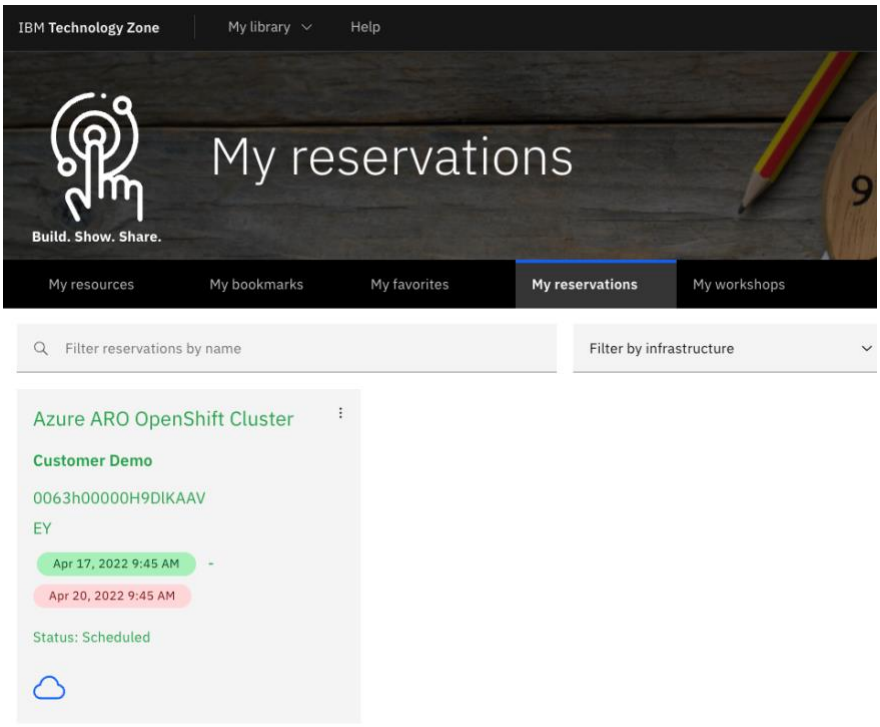
Receive Confirmation Email Log in for the First Time

Steps:

1. Once you have If you have completed the request for a demo environment you will receive a confirmation email.



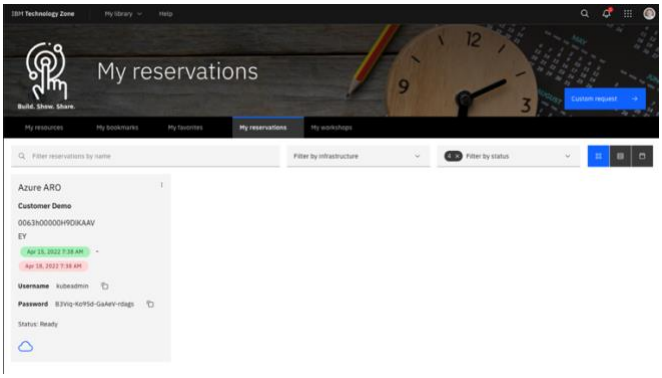
2. It is important you read the email as it contains instructions of how to log into the Tech Zone Demo Account.
3. You will receive a second email once your reservation has been processed; you click on **My reservations** to see the status of your reservation.


IBM Technology Zone
My library
Help
My reservations
Build. Show. Share.
My resources
My bookmarks
My favorites
My reservations
My workshops
Filter reservations by name
Filter by infrastructure
Azure ARO OpenShift Cluster
Customer Demo
0063h00000H9DIKAAV
EY
Apr 17, 2022 9:45 AM
-
Apr 20, 2022 9:45 AM
Status: Scheduled

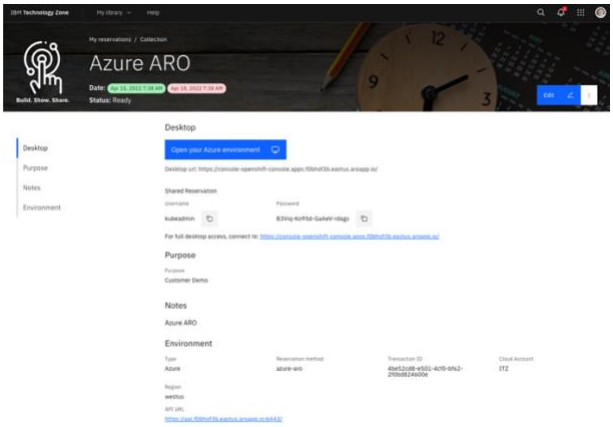
Log into your TechZone Environment

Steps:

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

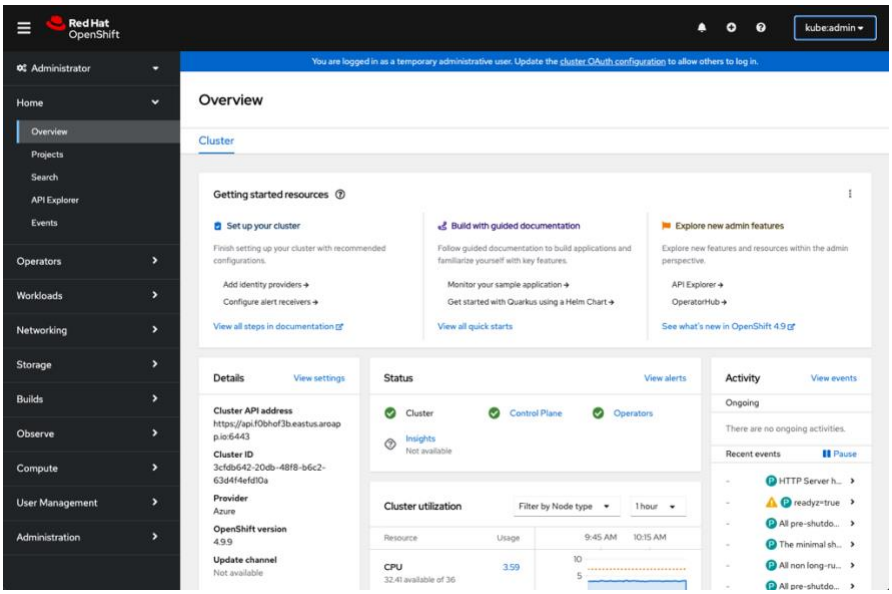


2. You will see the reservation detail's view, copy the **password** and click on the **Open your Azure environment** button.



3. This will display the OpenShift login screen for the cluster, enter the user ID and password details from the previous reservation screen. Typically its **kubeadmin** for the user id and the 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 Turbonomic into your TechZone environment

The installation process will use standard Terraform git repository that has been built using the modules you require to make Turbonomic 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-turbonomic

2. Navigate into the **automation-turbonomic** folder using your command line.
   a. The **README.md** has a comprehensive instruction on how to install this into other cloud environments than TechZone this document focuses in 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 credential.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=

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 you **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** this now not allowed under their new terms and conditions and you need to install **Colima** as an alternative

   $ brew install colima
   $ colima start

9. We are now ready to start install Turbonomic, run the **launch.sh** command, make sure you are in the root of the **automation-turbonimic** 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 that has downloaded 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**

    $ ./setup-workspace.sh

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

    ##
    # Name: Turbonomic Terraform Variable File

```
# Desc: Initial input variables to support installation of Turbonomic into the cloud provider of your choice
############################################################

## gitops-ocp-turbonomic_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-turbonomic_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 Turbonomic"
```

15. Change the storage_class_name value to **managed_premium** for Azure and other values for AWS. If we are on IBM Cloud you will need to run some automation to configure Storage for that environment
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 selection **Your Profile** this will take you to your default organization.
18. Set the repo_repo value to a unique name that you will recognize as the place where the GitOps configuration is going to be placed before Turbonomic 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 you will see that just the workspace directory is now configured for you.
21. Navigate into the /**workspaces/current** folder
22. Navigate into the **200** folder and run the following commands

```
$ cd 200-openshift-gitops
$ terraform init
$ 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, firstly look lets in your git hub repository. You will see the git repository has been created based on the name you provided. The Turbonomic 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.

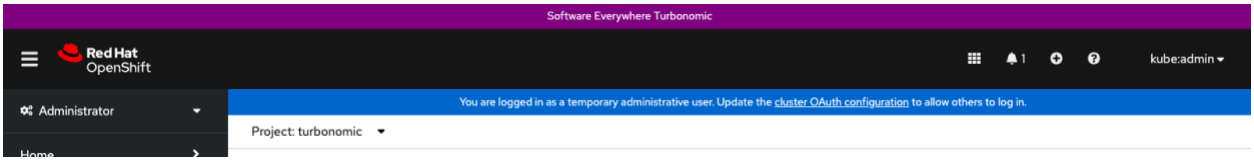
25. If you are using **IBM Cloud** run the Navigate into the **202** folder and run the following commands, this will configure the storage correctly for IBM Cloud. If you are installing on AWS or Azure you can skip this step and move to the **250** installation of Turbonomic.

    $ cd 202-turbonomic-ibmcloud-storage-class
    $ terraform init
    $ terraform apply --auto-approve

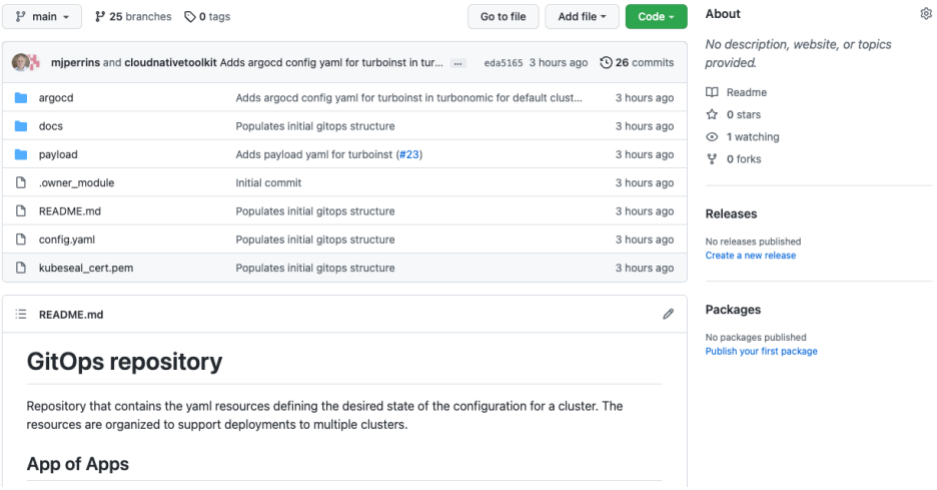
26. Now GitOps is installed in the cluster, and we have bound the git repository to OpenShift GitOps operator. We are now ready to populate this with some Software configuration that cause OpenShift GitOps to install the software into the cluster. Navigate into the **250** folder and run the following commands, this will install Turbonomic into the cluster.

    $ cd 250-turbonomic-multicloud
    $ terraform init
    $ terraform apply --auto-approve
    ……..
    $ Apply complete! Resources: 38 added, 0 changed, 0 destroyed.

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

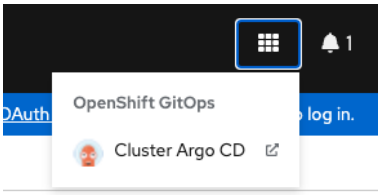


29. The next step is to validate everything installed correctly. Open your git repository where your git ops configuration was defined.

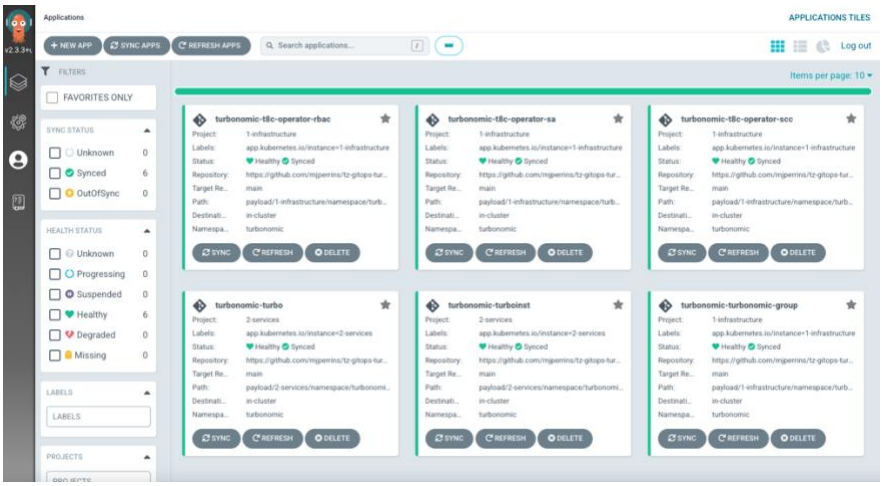


30. Check the **payload** folder has been created with the correct definitions for GitOps. Navigate to the **payload/2-services/namespace/turbonomic** folder and look at the content of the installation YAML files.

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



33. Complete the authorization with OpenShift and then narrow the filters by selecting the **Turbonomic** namespace.



34. This will show you the GitOps dashboard of the software you have installed using GitOps techniques
35. Click on **turbonomic-turboinst** tile
36. You will see all the microservices that Turbonomic uses to install and their enablement state

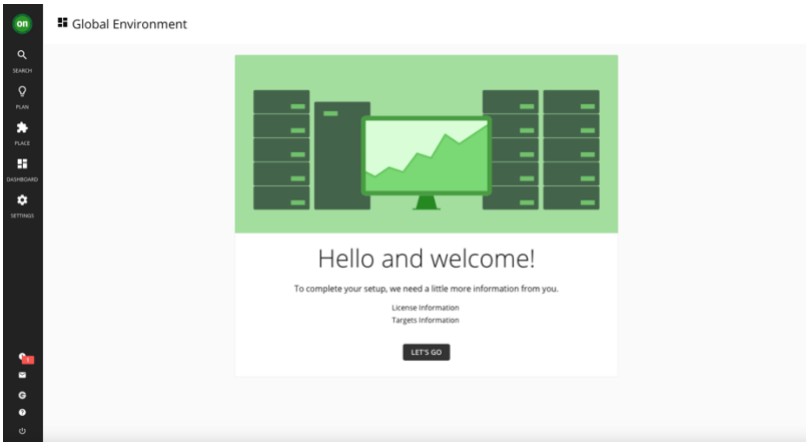
THIS CONCLUDES THE GITOPS INSTALLATION STEPS

Configuring Turbonomic after installation into TechZone

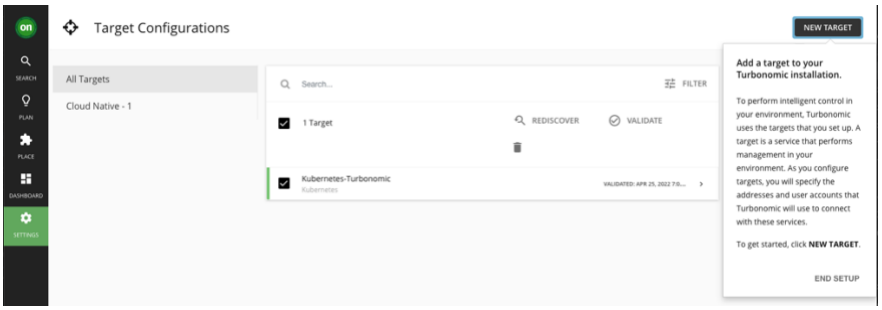
Now the installation process is complete it is now time to configure Turbonomic and load your downloaded license key.

Steps:

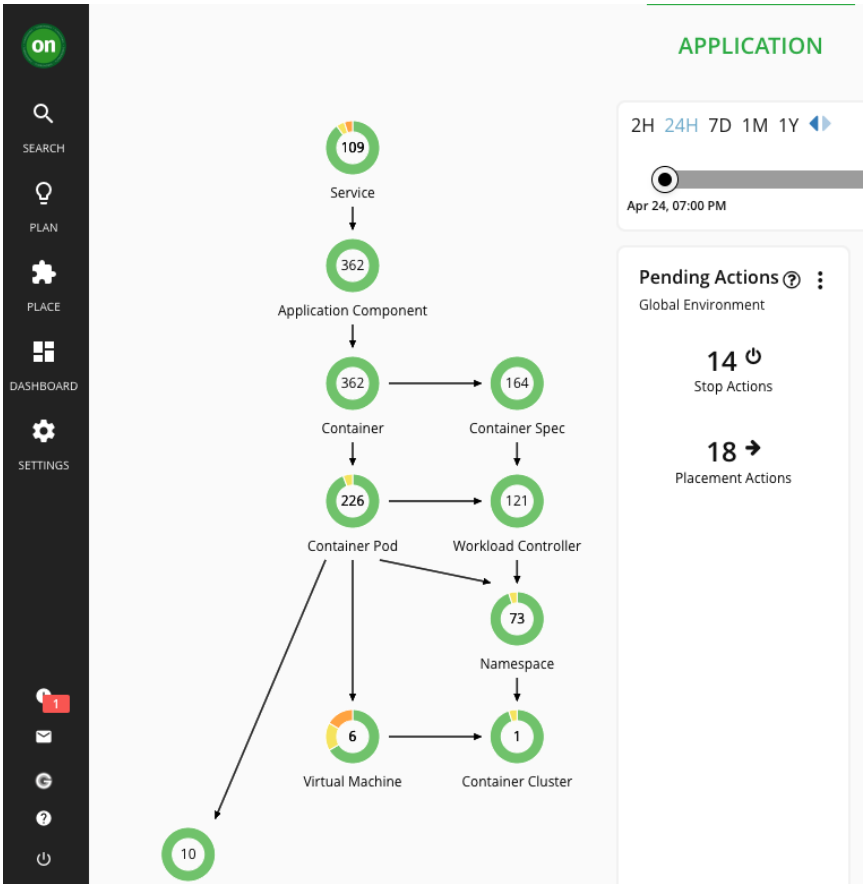
1. In the **OpenShift** console navigate to the **Networking->**Routes and change the project from to **turbonomic** you will see the route to launch dashboard for **Turbonomic**. Click on the Location URL to open **Turbonomic**
2. The first time you launch the dashboard it will ask you to define an **Administration** password. Enter your new password and confirm it. **Don't forget to store it in your password manager**
3. Once the account is created you will be greeted with the default screen.



4. Make sure you have downloaded the Tech Sales license key following instructions in the pre-requisites section at the front of this document.
5. Click on **Settings** on left menu, then click on **License** icon, click **Import license**
6. Drag you license key into the drop area and you will get a screen stating your has been added
7. Now we need to point Turbonomic at an environment for it to monitor
8. Click on the **Add Targets** button.



9. Click on **Kubernetes-Turbonomic** then **Validate** button let validation complete
10. Then click on the **On** icon at the top of the left menu you will see a monitor view of **Turbonomic**



THIS CONCLUDES SETUP OF TURBONOMIC