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

This document will walk you through the steps on how to prepare your environment to demonstrate the Turbonomic 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 Turbonomic AWS, Azure and IBM Cloud which 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

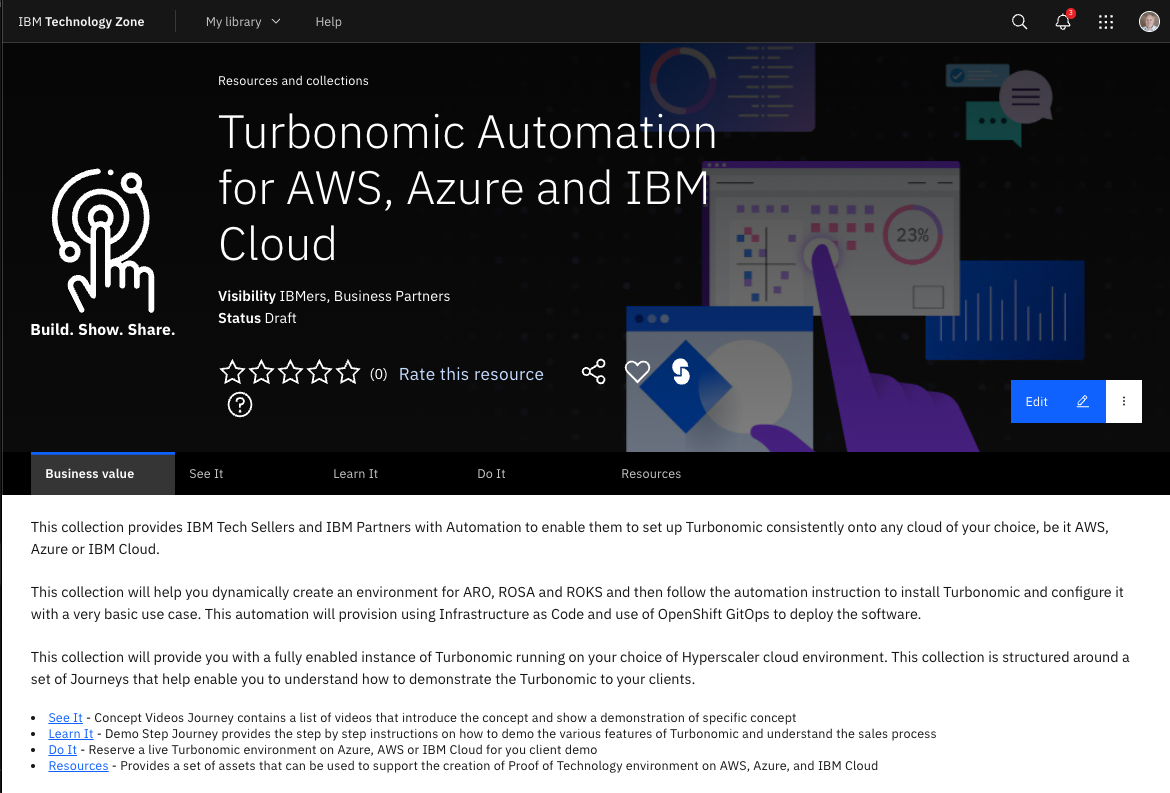
**Obtaining a License Key:**

To use Turbonomic, 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:  
  
1. For PoCs/PoTs, Partners can download a license key from **PartnerWorld** [https://www.ibm.com/partnerworld/public](https://w3-03.ibm.com/software/xl/download/ticket.wss)   
2. You can search the software catalog for **M05C4EN IBM Turbonomic Application Resource Management On-Prem 8.4.6 for install on Kubernetes English**,  
3. Download the package containing a 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 and are currently available under IBM PPA terms and conditions  
  
**IBMers**  
IBMers, 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 Kubernetes English** and download  
4. Extract this download package to get the turbonomic license key  
 This package contains license file for turbonomic, with a name similar to **CP4MCM\_IBM\_ARM\_OEM\_Premier\_License\_July\_2022.lic**

**Requesting TechZone Environment**

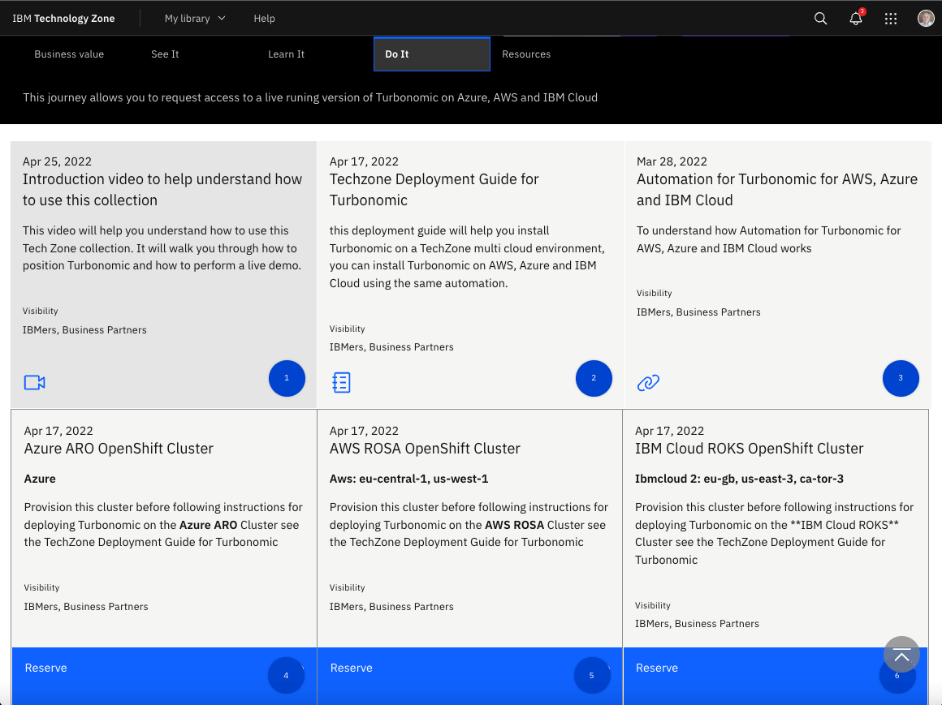
**Demo Steps:**

1. If you have not already done so, access the Tech Zone collection for Turbonomic Automation for AWS, Azure, and IBM Cloud
   1. [https://techzone.ibm.com/collection/turbonomic-automation-multicloud](file:///C:\Users\AnitaMathur\Downloads\a.%09https:\techzone.ibm.com\collection\turbonomic-automation-multicloud)



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

Tab



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

Graphical user interface, application

Description automatically generated

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

Graphical user interface, application

Description automatically generated

1. 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
   1. 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>
2. Enter the **Sales Opportunity Number** as you will not be able to proceed without a valid and live opportunity
3. Select your duration for the environment, and for the size for Turbonomic keep it as a **3 Worker Node Count** and **4 CPU x 16GB** worker node flavor
4. Click on **Submit** once all the fields are entered correctly
5. Once the **Submit** has completed you will see the following screen with a THANK YOU message to confirm that your reservation is being processed

Graphical user interface, text, application

Description automatically generated

1. 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.

Graphical user interface, application, Teams

Description automatically generated

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

Graphical user interface, website

Description automatically generated

**Log into your TechZone Environment**

**Steps:**

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

Graphical user interface

Description automatically generated

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

Graphical user interface, text, application

Description automatically generated

1. 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.
2. Once you have logged in you will see the main OpenShift administration screen

Graphical user interface, application, Teams

Description automatically generated

1. 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 need 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](file:///C:\Users\AnitaMathur\Downloads\git%20clone%20https:\github.com\IBM\automation-turbonomic)

1. Navigate into the **automation-turbonomic** folder using your command line.
   1. 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.
2. 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=

1. You will need to populate these values. Add your **Git Hub** username and your Personal Access Token to **repo\_username** and **repo\_token**
2. From you **OpenShift console** click on top right menu and select **Copy login command** and click on **Display Token**

Graphical user interface, application

Description automatically generated

1. Copy the **API Token** value into the **login\_token** value
2. Copy the **Server URL** into the **server\_url** value, only the part starting with **https**
3. 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

1. We are now ready to install Turbonomic, run the **launch.sh** command, make sure you are in the root of the **automation-turbonomic** 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 $

1. **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.
2. Next step is to create a workspace to run the Terraform automation.
3. Run the command **setup-workspace.sh**

$ ./setup-workspace.sh

1. 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 usineditoyour choice.
2. Edit the default **terraform.tfvars** file to 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"

1. 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
2. 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.
3. 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.
4. 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 Turbonomic is installed into the cluster.
5. You can change the Banner text to something useful for you client project or demo.
6. Once in the container, check out the file system with **ls -al** command to see the workspace directory is now configured for you.
7. Navigate into the /**workspaces/current** folder
8. 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.

1. This will kick off the automation for setting up the GitOps Operator into your **TechZone** cluster.
2. 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 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.
3. 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 on to the **250** installations of Turbonomic.

$ cd 202-turbonomic-ibmcloud-storage-class

$ terraform init

$ terraform apply --auto-approve

1. 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 will 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

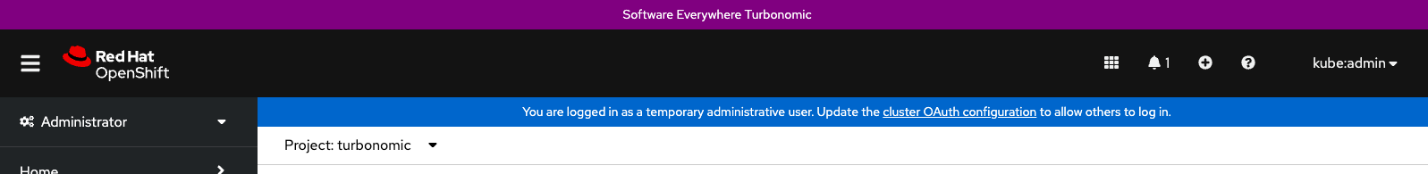
$ terraform init

$ terraform apply --auto-approve

………

$ Apply complete! Resources: 38 added, 0 changed, 0 destroyed.

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



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

` Graphical user interface, text, application, email

Description automatically generated

1. Check if 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.

Graphical user interface, text, application

Description automatically generated

1. You should see the **Operator** CR definitions
2. 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.

Graphical user interface, application

Description automatically generated

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

Graphical user interface, text, application, chat or text message

Description automatically generated

1. This will show you the GitOps dashboard of the software you have installed using GitOps techniques
2. Click on **turbonomic-turboinst** tile
3. 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 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.

Graphical user interface, application, table

Description automatically generated

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

Graphical user interface, text, application, Teams

Description automatically generated

1. Click on **Kubernetes-Turbonomic** then **Validate** button to complete the validation
2. Then click on the **On** icon at the top of the left menu to see a monitor view of **Turbonomic**

Graphical user interface

Description automatically generated

***THIS CONCLUDES SETUP OF TURBONOMIC***