



Preparing to install Turbonomic 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 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>
2. You can search the software catalog for **M07BWENIBM Turbonomic Application Resource Management On-Prem V8.6 for install on Kubernetes English**
3. Download the package containing a license file for Turbonomic, with a name like **TURBONOMIC-ARM.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:



Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud
IBM Ecosystem Engineering

1. Go to XL Leverage <https://w3-03.ibm.com/software/xl/download/ticket.wss>
2. Search with keyword: turbonomic
3. Select the package ***M07BWENIBM Turbonomic Application Resource Management On-Prem V8.6 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 like ***TURBONOMIC-ARM.lic***



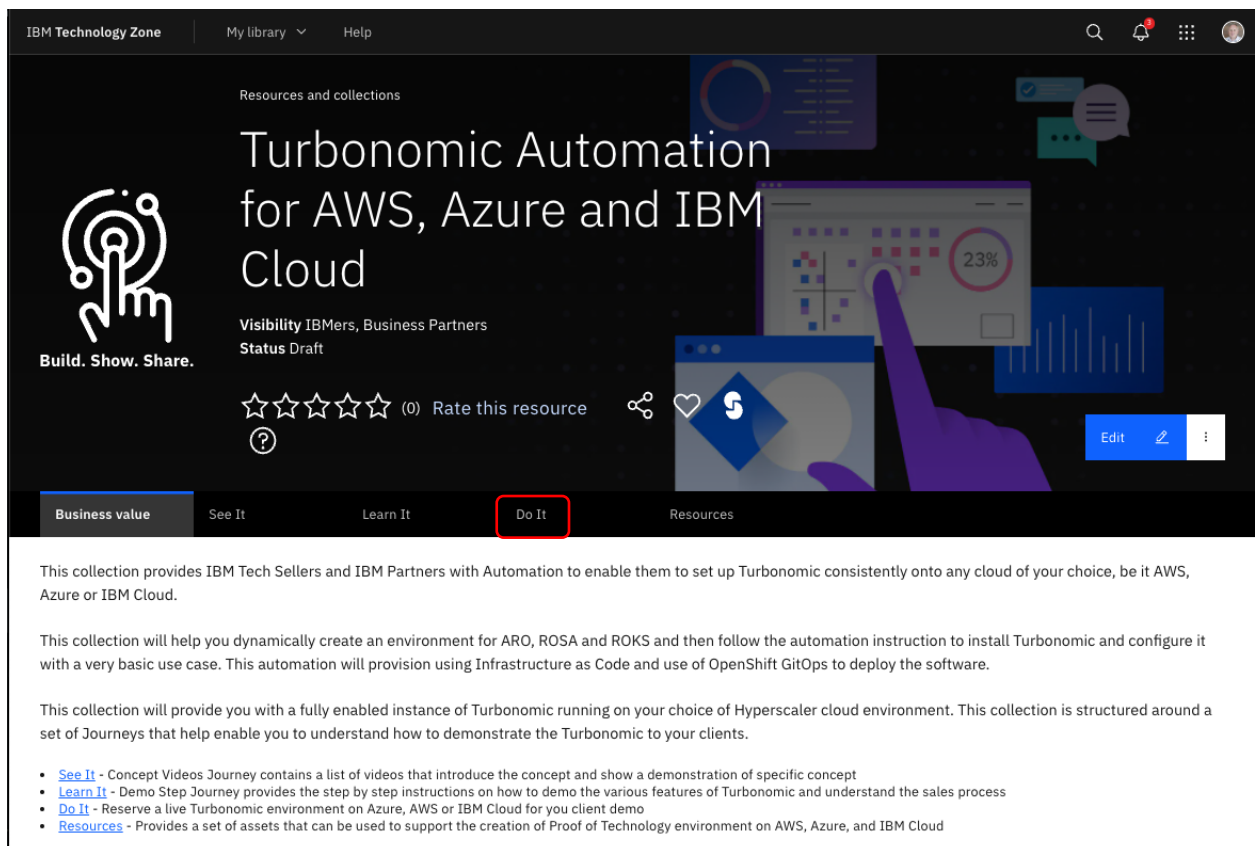
Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering

Requesting TechZone Environment

Demo Steps:

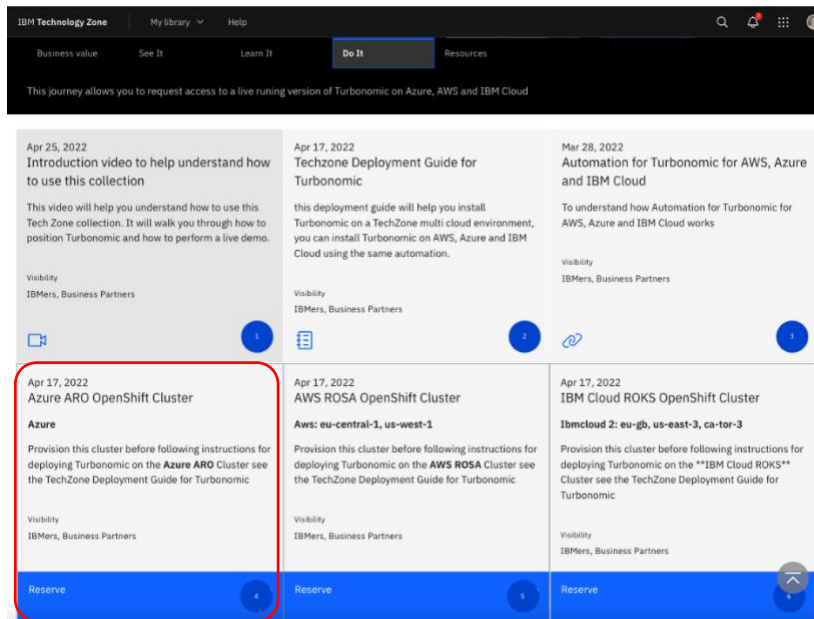
1. If you have not already done so, access the Tech Zone collection for Turbonomic Automation for AWS, Azure, and IBM Cloud
 - a. <https://techzone.ibm.com/collection/turbonomic-automation-multicloud>



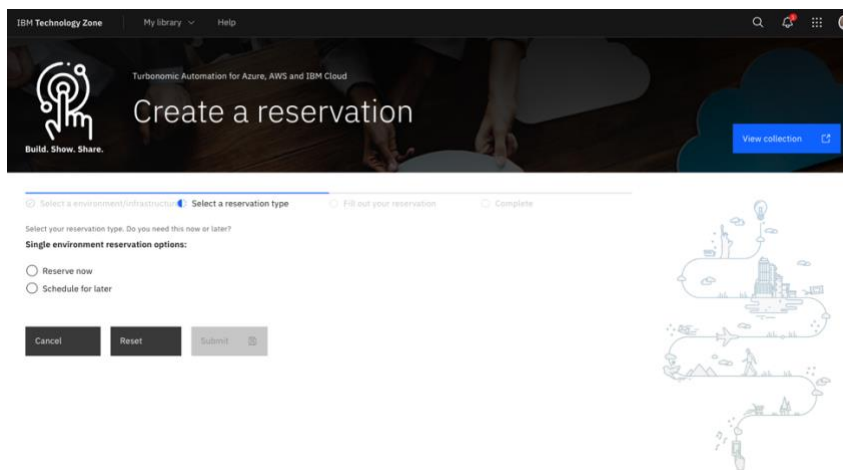
2. The first thing you need to do is to request an access to the demo environment. Click on **Do It** Tab



Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud IBM Ecosystem Engineering



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



Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering

The screenshot shows the 'Create a reservation' form in the IBM Technology Zone. The form is titled 'Create a reservation' and has a 'View collection' button. The form is divided into several sections: 'Name', 'Purpose', 'Customer name(s)', 'Sales Opportunity Number', 'Preferred Geography (required)', 'End date and time', 'Worker Node Count (required)', and 'Reservation policy: customer-demo_pot_poc'. The 'Name' field is filled with 'Azure ARO OpenShift Cluster'. The 'Purpose' dropdown is set to 'Customer Demo'. The 'Customer name(s)' field is empty. The 'Sales Opportunity Number' field is filled with '0063H00000HCHKAAY'. The 'Preferred Geography' dropdown is set to 'East US'. The 'End date and time' section shows a date of '9:27 AM' and a time of '9:27 AM'. The 'Worker Node Count' dropdown is set to '3'. The 'Reservation policy' section shows a 'Default duration' of '3 days (72 hours)', a 'Delay by' of '0', a 'Maximum duration' of '2 weeks (336 hours)', an 'Extension duration' of '7 days (168 hours)', and an 'Extension limit' of '12'.

- 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
 - 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>
- Enter the **Sales Opportunity Number** as you will not be able to proceed without a valid and live opportunity
- 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
- Click on **Submit** once all the fields are entered correctly
- Once the **Submit** has completed you will see the following screen with a THANK YOU message to confirm that your reservation is being processed

The screenshot shows the 'Thank you' confirmation screen in the IBM Technology Zone. The screen has a 'Thank you.' heading and a message: 'Your azure reservation has been created for Azure ARO OpenShift Cluster from Sun, Apr 17, 2022 9:27 AM to Wed, Apr 20, 2022 9:27 AM. You will receive an email at mijperrin@us.ibm.com with information about your reservation once it is ready. You can manage your reservations on the My reservations page.' There are two buttons: 'Done' and 'My reservations'.



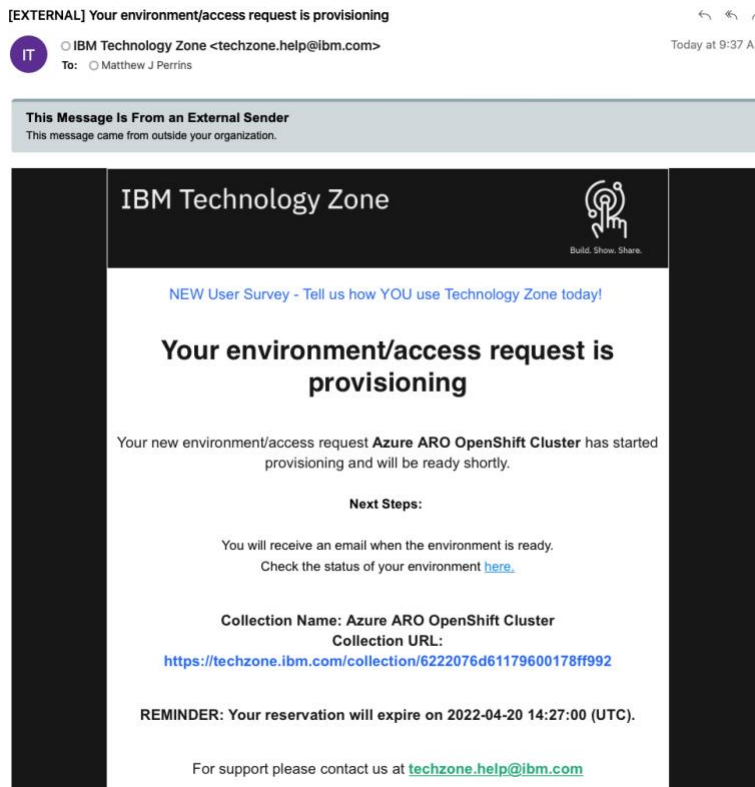
10. Check your email for confirmation.



Receive Confirmation Email Log in for the First Time

Steps:

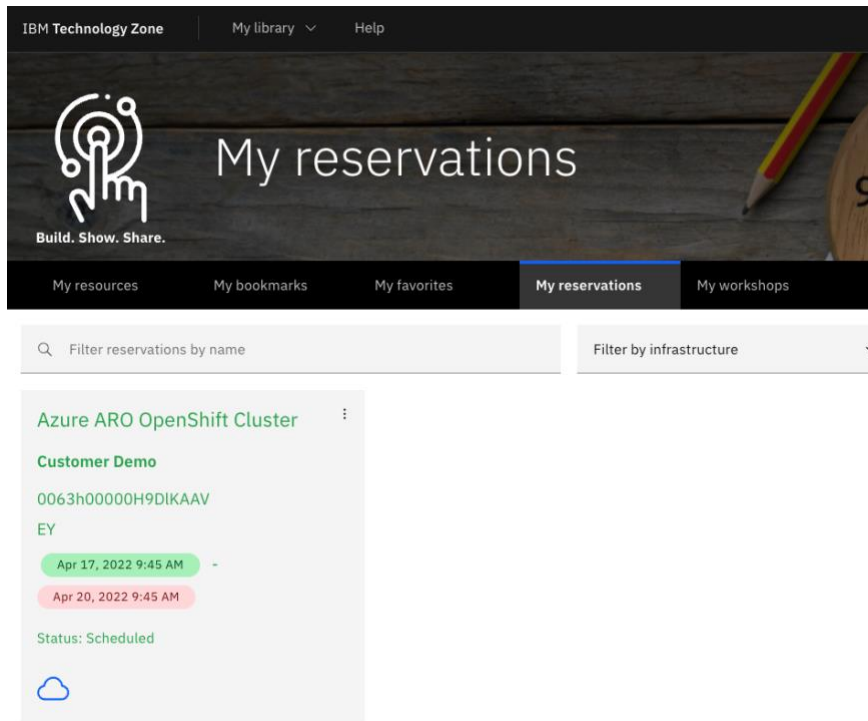
1. Once you have reserved the demo environment, you will receive the confirmation email.



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.



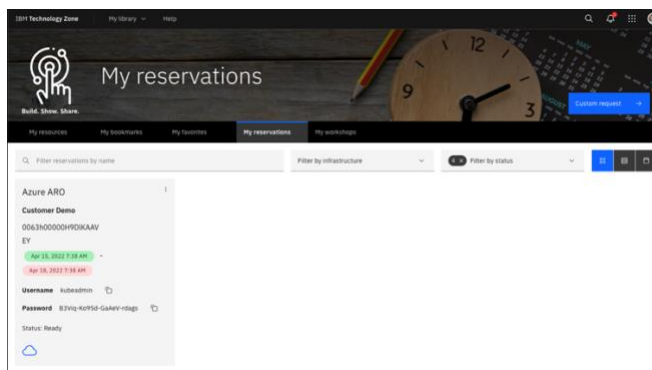
Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud IBM Ecosystem Engineering



Log into your TechZone Environment

Steps:

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

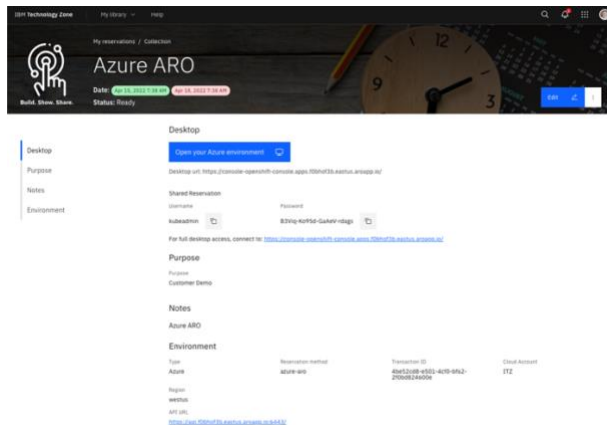


2. You will see the reservation view in detail, copy the **password** and click on the **Open your Azure environment** button.

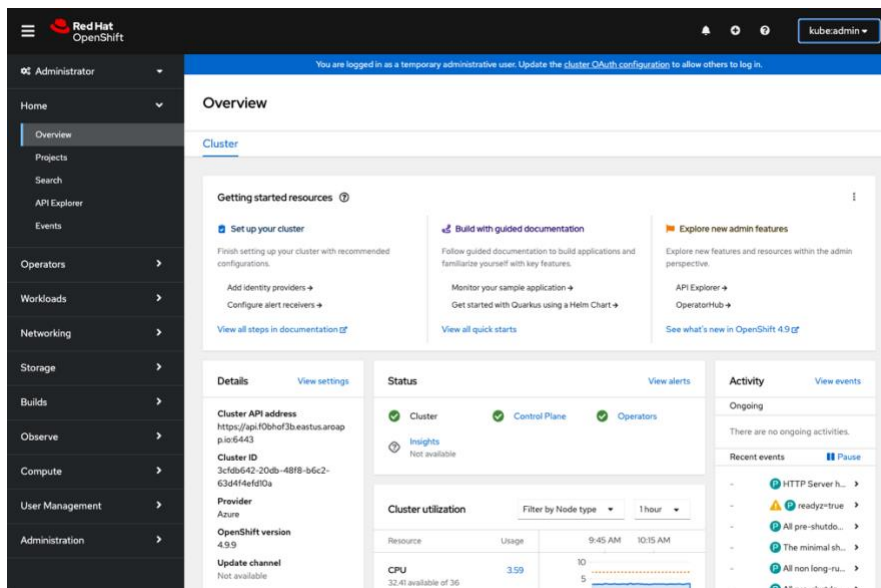


Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering



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 Turbonomic into your TechZone environment



Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering

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
```

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 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 OpenShift Environment  
## 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
```

```
## gitops_repo_host: The host for the git repository  
TF_VAR_gitops_repo_host=github.com  
## gitops_repo_username: The username of the user with access to the repository  
TF_VAR_gitops_repo_username=  
## gitops_repo_token: The personal access token used to access the repository  
TF_VAR_gitops_repo_token=
```

```
## TF_VAR_server_url: The url for the OpenShift api server  
TF_VAR_server_url=  
## TF_VAR_cluster_login_token: Token used for authentication to the api server  
TF_VAR_cluster_login_token=
```

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**



Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering

Your API token is

sha256~uSjFiiAvvc1TBGK4gRhVIbWknF5tVvVxEZ790yyTEno

Log in with this token

```
oc login --token=sha256~uSjFiiAvvc1TBGK4gRhVIbWknF5tVvVxEZ790yyTEno --server=https://api.hr9czz9.eastus.aroapp.io:6443
```

Use this token directly against the API

```
curl -H "Authorization: Bearer sha256~uSjFiiAvvc1TBGK4gRhVIbWknF5tVvVxEZ790yyTEno"
      "https://api.hr9czz9.eastus.aroapp.io:6443/apis/user.openshift.io/v1/users/~"
```

[Request another token](#)

[Logout](#)

6. Copy the **API Token** value into the **login_token** value
7. Set the **repo_username** value to your GitHub User Name
8. Copy the **Server URL** into the **server_url** value, only the part starting with **https**
9. 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
```

10. 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/cli-tools:v1.1
```

```
Attaching to running container...
```

```
/terraform $
```

11. **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.
12. Next step is to create a workspace to run the Terraform automation.
1. Run the command **setup-workspace.sh** you will need to provide the **-p** platform value which can be **azure** | **aws** or **ibm** then supply a prefix name

```
$ ./setup-workspace.sh -p azure -n tubro01
```

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 of your choice.
14. Edit the default **terraform.tfvars** file to enable you to setup the GitOps parameters.



Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering

```
#####
# 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
## <your block storage on aws: gp2, on azure: managed-premium, on ibm: ibmc-vpc-block-10iops-tier>
gitops-ocp-turbonomic_storage_class_name="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.
## your gitorg - if left blank the value will default to your username
gitops_repo_org="mjperrins"

## gitops-repo_repo: The short name of the repository (i.e. the part after the org/group name)
gitops_repo_repo="turbo-gitops-01"

## 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 will have already been set 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 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 to see 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, 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



Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering

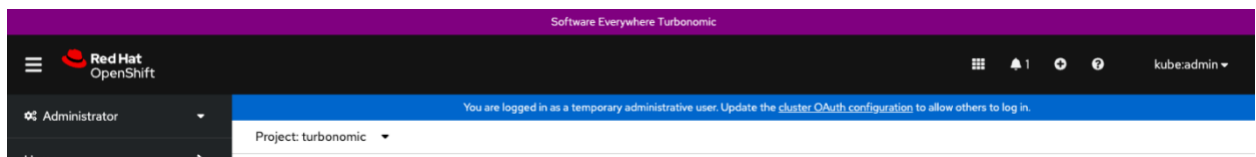
software. The second place to look is the OpenShift console, Click **Workloads->Pods** and you will see the GitOps operator being installed.

25. Now GitOps is installed in the cluster, 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.

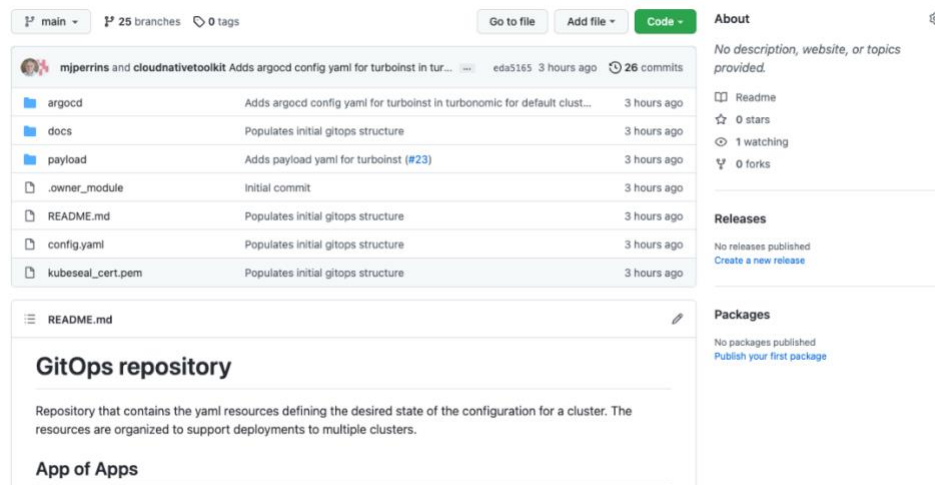
```
$ cd 250-turbonomic-multicloud
$ terraform init
$ terraform apply --auto-approve
.....
$ Apply complete! Resources: 38 added, 0 changed, 0 destroyed.
```

26. Once the installation has finished you will see a message from Terraform defining the state of the environment.

27. You will see the first change with a purple banner describing what was installed



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



29. 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.



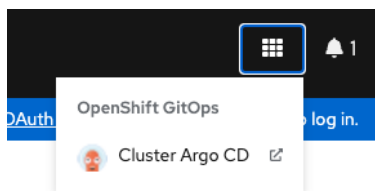
Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering

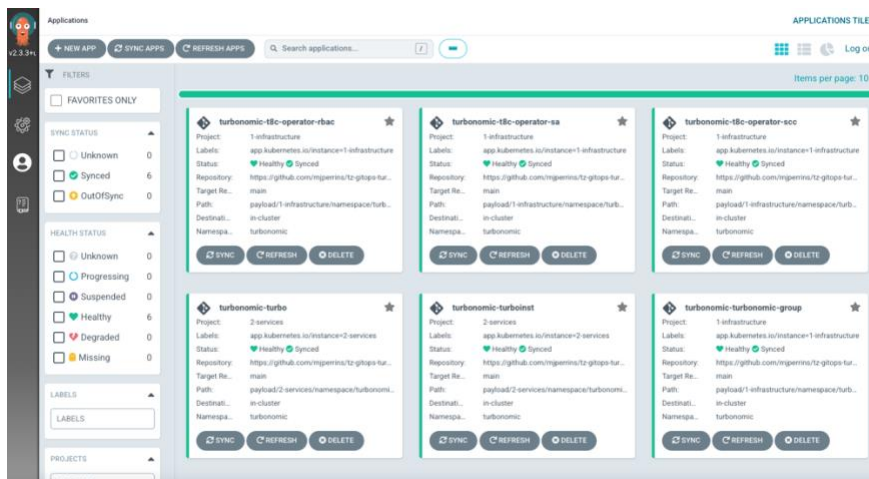


30. You should see the **Operator** CR definitions

31. 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.



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



33. This will show you the GitOps dashboard of the software you have installed using GitOps techniques

34. Click on **turbonomic-turboinst** tile

35. You will see all the microservices that Turbonomic uses to install and their enablement state

THIS CONCLUDES THE GITOPS INSTALLATION STEPS



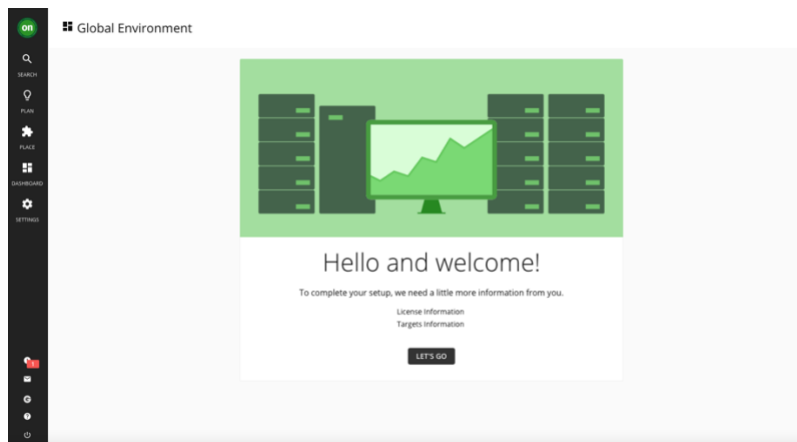
Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud IBM Ecosystem Engineering

Configuring Turbonomic after installation into TechZone

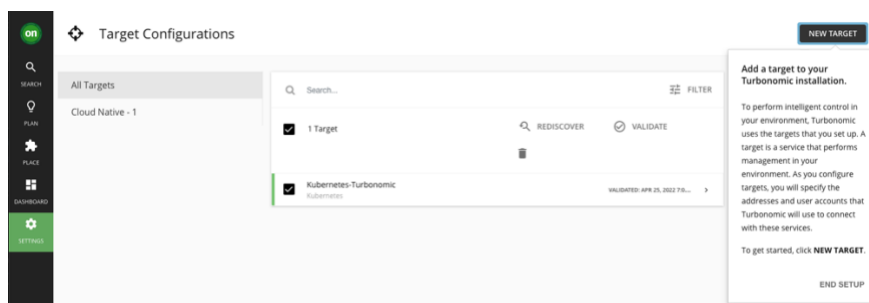
Now the installation process is complete it is time to configure Turbonomic and load your downloaded license key.

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.



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

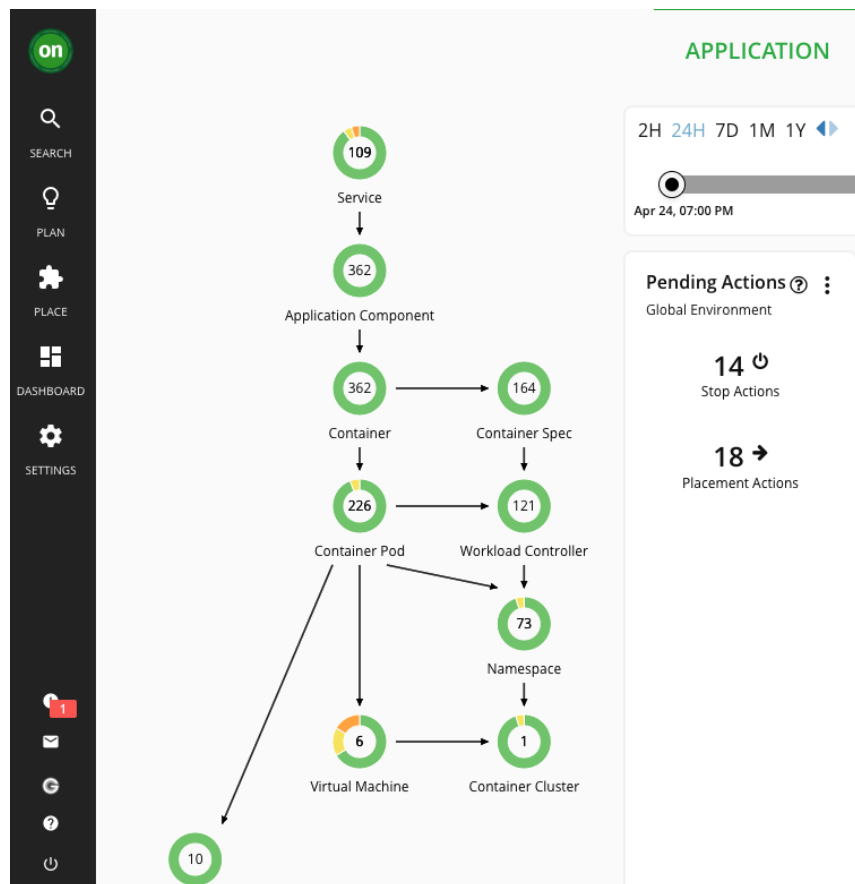


9. Click on **Kubernetes-Turbonomic** then **Validate** button to complete the validation
10. Then click on the **On** icon at the top of the left menu to see a monitor view of **Turbonomic**



Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering



THIS CONCLUDES SETUP OF TURBONOMIC



Post install steps for targets configuration

You can optionally follow the steps below to configure Turbonomic with certain pre-defined target environments. This would surface different kinds of action recommendations in Turbonomic console.

Note: Only Turbonomic deployed on AWS ROSA, Azure ARO and IBM ROKS clusters through TechZone is supported.

Pre-requisites:

- Enable aws, azure, vcenter, instana and udt probes also in addition to kubeturbo

If these probes are not enabled, navigate to xl-release.yaml file in your Github repo and update the file. Be careful about indentation while editing the yaml file. After making the changes in the file on Github, 'commit' the changes. ArgoCD running on the cluster will pick-up the changes and deploys the new probes.

<https://github.com/<Github-org>/<Github-repo>/tree/main/payload/2-services/namespace/<namespace>/turboinst/xl-release.yaml>

Below is an example:

```
apiVersion: charts.helm.k8s.io/v1
kind: XL
metadata:
  name: xl-release
  annotations:
    argocd.argoproj.io/sync-wave: "2"
spec:
  global:
    repository: turbonomic
    tag: 8.4.4
    storageClassName: managed-premium
    serviceAccountName: t8c-operator
  market:
    image:
      pullPolicy: IfNotPresent
      repository: docker.io/turbonomic
      tag: 8.4.4
      serviceAccountName: t8c-operator
  nginx:
    nginxIsPrimaryIngress: false
    httpsRedirect: false
  nginxingress:
    enabled: true
  openshiftingress:
    enabled: true
  aws:
    enabled: true
  azure:
    enabled: true
  vcenter:
    enabled: true
  instana:
    enabled: true
  udt:
    enabled: true
  kubeturbo:
    enabled: true
```

Wait for few minutes for the new probes to get enabled.



Steps:

1. Clone the Turbonomic-tz-target-config repository:

```
$ git clone https://github.com/cloud-native-toolkit/turbonomic-tz-target-config
```

2. Navigate to this cloned directory:

```
$ cd turbonomic-tz-target-config
```

3. Make sure you are logged into the OCP cluster as cluster admin

4. Change project to where Turbonomic is deployed:

```
$ oc project <turbo-project-name>
```

5. Run below command:

```
$ oc new-app -f turbonomic-tz-target-config.yaml -p TURBO_NAMESPACE=<Turbonomic-namespace> -p  
TURBO_ADMIN_USER=<Turbonomic-User> -p TURBO_PASSWORD=<Turbonomic-password>
```

Here, TURBO_NAMESPACE is where Turbonomic is deployed

TURBO_ADMIN_USER is the Turbonomic admin user

TURBO_PASSWORD is the password for the Turbonomic admin user

Wait for ~30 minutes for required actions to appear in Turbonomic console as shown in these screenshots.

Note: Actions suggested on AWS, Azure and vSphere resources cannot be executed as these are added as read-only targets. However actions suggested on the OCP cluster, where Robot-shop application has also been deployed, can be executed.



Turbonomic Automation Deployment Guide for AWS, Azure, and IBM Cloud

IBM Ecosystem Engineering

