# AEC Open Shift | Lab Guide

|  |
| --- |
| Conditions and Terms of Use |
| This course package is proprietary and confidential. Duplicating, reselling and/or distribution of course materials in their entirety or in part is not permitted without the expressed prior written consent of Spektra Systems, LLC. |

Contents

[AEC Open Shift | Lab Guide 1](#_Toc496905327)

[Workshop Architecture and Objective 2](#_Toc496905328)

[Workshop Architecture Before the Performing Labs 2](#_Toc496905329)

[Labs Objective 2](#_Toc496905330)

[Workshop Architecture after deploying ARM Template 2](#_Toc496905331)

[Lab 1: Introduction to Azure Portal 3](#_Toc496905332)

[Lab Overview` 3](#_Toc496905333)

[Prerequisites 3](#_Toc496905334)

[Time Estimate 4](#_Toc496905335)

[Exercise 1: Log into your Azure Portal 4](#_Toc496905336)

[Exercise 2: Verify access to the Subscription 6](#_Toc496905337)

[Lab 2: Deploying Open Shift cluster using ARM templates 9](#_Toc496905338)

[Lab Overview 9](#_Toc496905339)

[Prerequisites 9](#_Toc496905340)

[Time Estimate 9](#_Toc496905341)

[Exercise 1: Create an Azure AD Application 9](#_Toc496905342)

[Exercise 2: Create a Keyvault 14](#_Toc496905343)

[Exercise 3: Deploy Openshift Cluster using ARM Template 18](#_Toc496905344)

[Exercise 4: Configure Azure AD Authentication 23](#_Toc496905345)

[Exercise 5: Deploy a 2 Tier Node JS Application on Open Shift 39](#_Toc496905346)

**Workshop Architecture and Objective**

Workshop Architecture Before the Performing Labs

As part of Hand on Lab, you will get following details via email. Make a note of these details as these shall be leveraged throughout the lab exercise

* Azure Access: Azure Username and Password (Default Password, Change at first login)

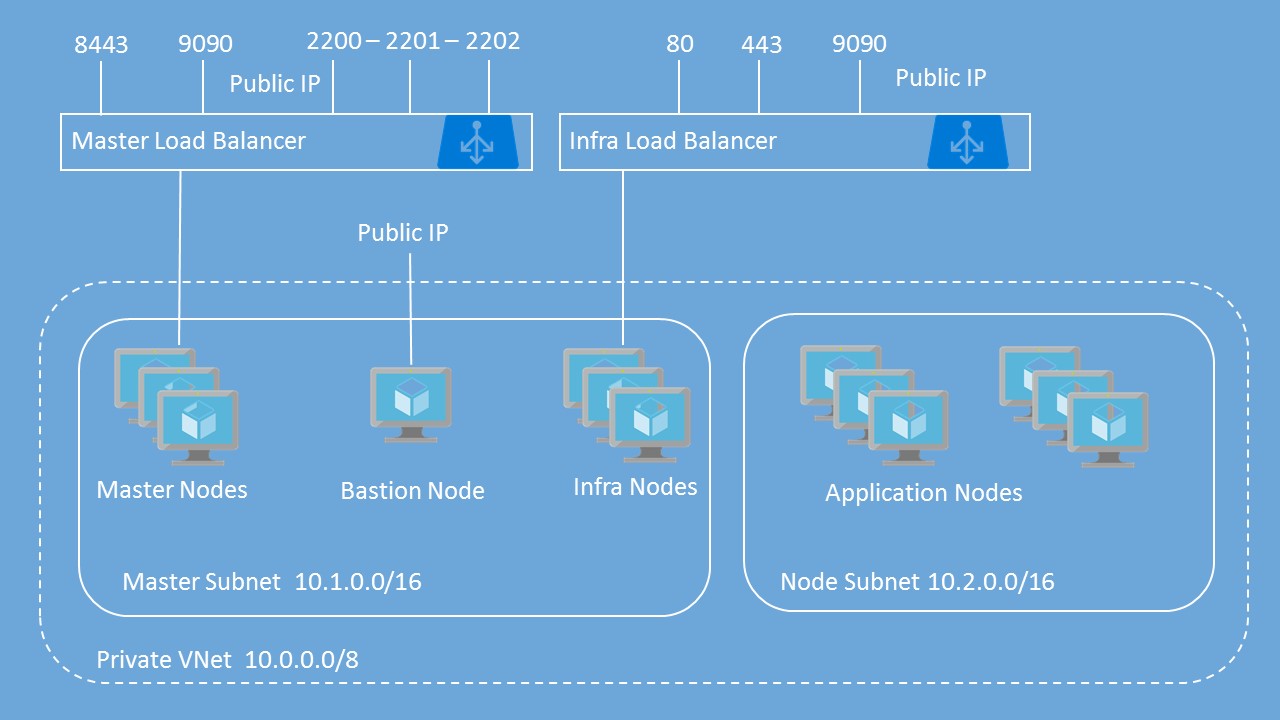
Labs Objective

During this lab, you will deploy Open Shift cluster on Azure and integrate Azure AD Authentication and Azure Container Registry into Open Shift. Detailed steps to achieve this is as follows.

* Get Familiar with Azure Portal and Ansible Tower UI
* Create an Azure AD Application for Authentication
* Create a key vault to store SSH Key
* Deploy Open Shift using ARM Template
* Configure Azure AD Authentication
* Deploy 2 Tier App on Open Shift

Workshop Architecture after deploying ARM Template

Following illustrates the architecture in your Azure deployment after completion of exercise’s part of workshop.



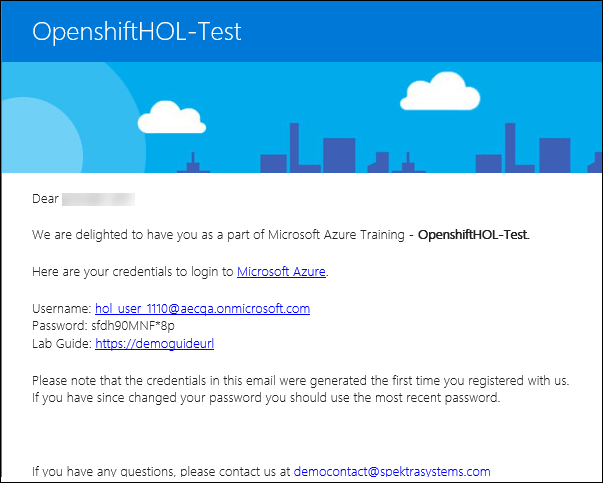
# **Lab 1: Introduction to Azure Portal**

## Lab Overview`

This lab will take you through Azure login and portal experience.

## Prerequisites

* Windows or a Mac machine with HTML5 supported browser such as Microsoft Edge, Internet Explorer, Chrome or Firefox
* You should have registered in the training portal <https://azuretraining.spektrasystems.com> and received the confirmation message with the credentials to login to the [Azure portal](https://portal.azure.com/).



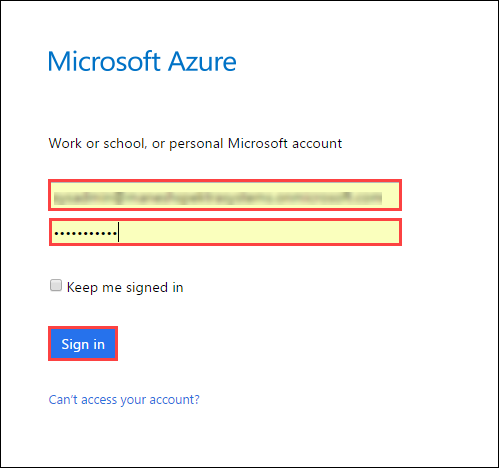
## Time Estimate

10 minutes

## Exercise 1: Log into your Azure Portal

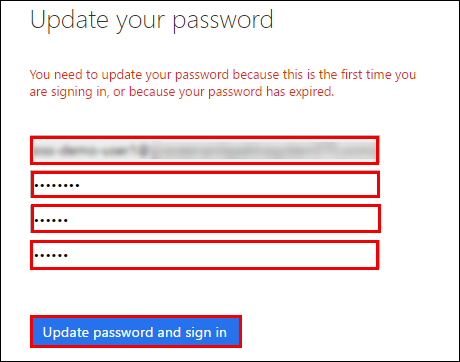
In this exercise, you will log into the Azure Portal using your Azure credentials.

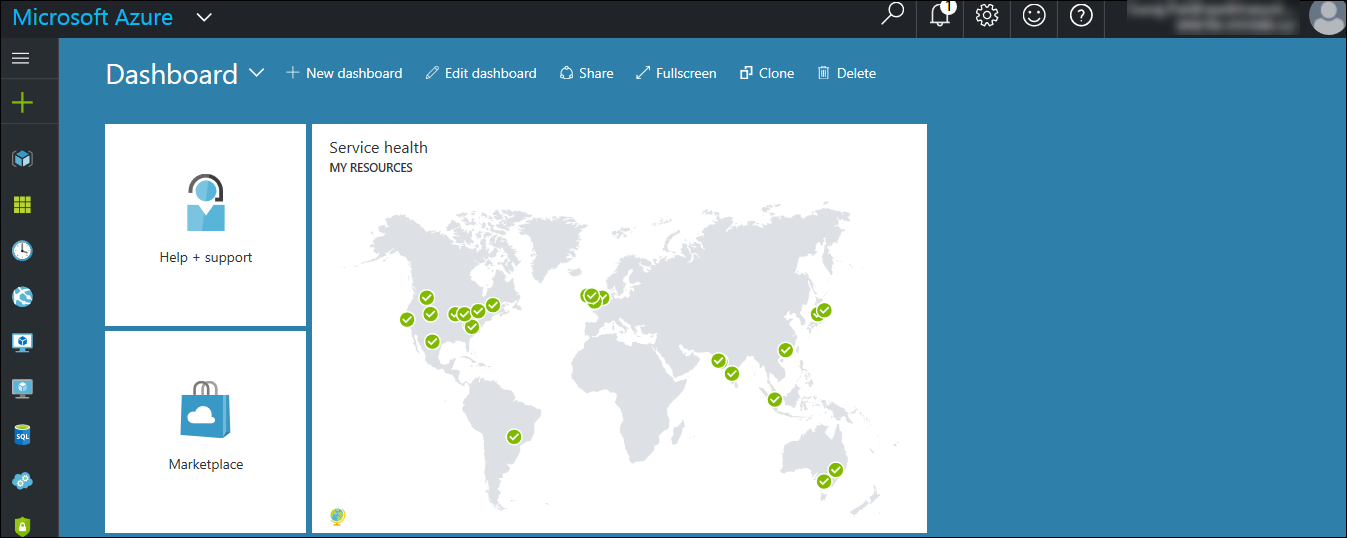
1. **Launch** a browser and **Navigate** to [**https://portal.azure.com**](https://portal.azure.com). Provide the credentials that you received via email. Clickon **Sign In**.



**Note** : At the first login, you may have to change the password, if asked for.

1. **Enter** a new **password**. Then select **Update password and sign in.**



1. Now, you will be directed to the Azure Dashboard  
     
   

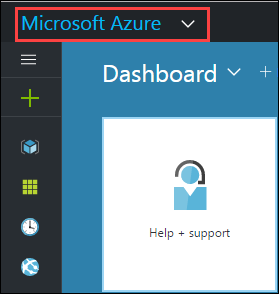
Exercise 2: Verify access to the Subscription

In this exercise, you will verify the type of role you are assigned in this Subscription.

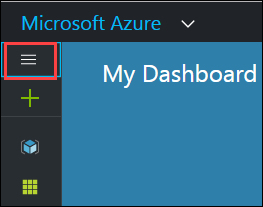
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.



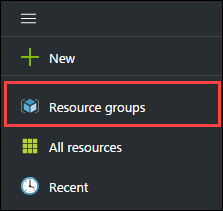
1. **Click** on **Microsoft Azure** at the top left corner of the screen,to view the Dashboard.



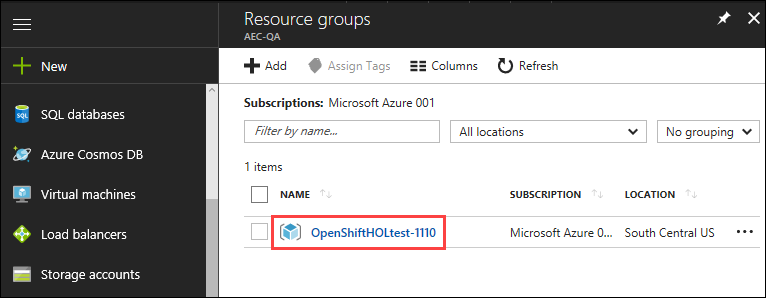
1. To toggle **show/hide** the Portal menu options with icon, **Click** on the **Show Menu** button.



1. **Click** on the **Resource groups** button in the **Menu** **navigation** bar to view the **Resource groups** blade.

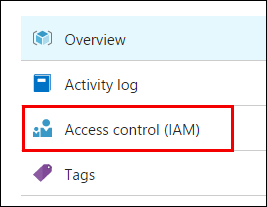
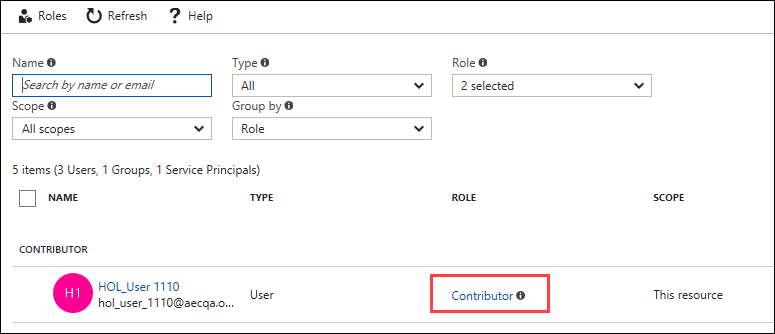


1. You will see a Resource Group which you have access to, **click** on it.



**Note**:

The Resource Group shown here is for demo purpose only. Actual name of the Resouce Group that you see may differ.

1. From the Resource Group blade that come up, **Select** the Access Control ( IAM ) which is on the left side of the blade.  
     
   
2. In the new blade that come up, you can see the **role** that is assigned to you.  
     
   

# **Lab 2: Deploying Open Shift cluster using ARM templates**

## Lab Overview

In this lab, you will learn how to deploy the Open Shift Cluster on Azure using ARM templates.

## Prerequisites

* Lab 1 must be completed

## Time Estimate

45 minutes

Exercise 1: Create an Azure AD Application

In this exercise, you will create an Azure AD App and retrieve the Client ID and Client secret values.

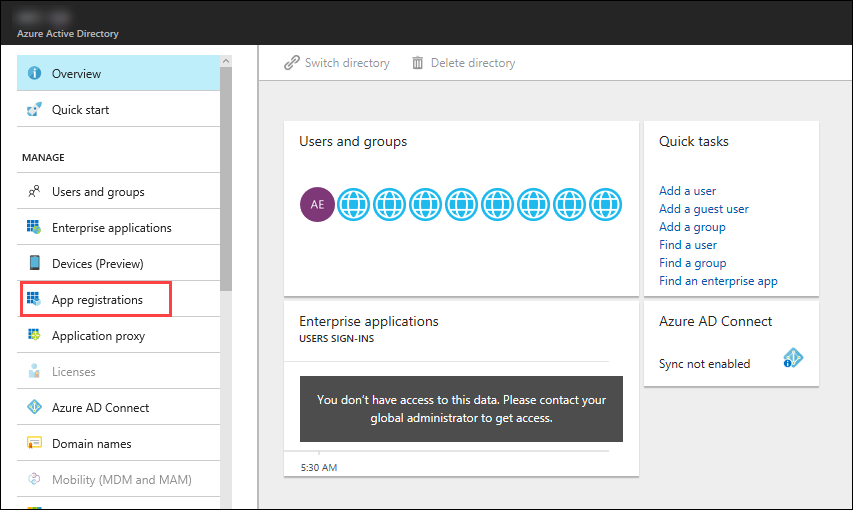
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.



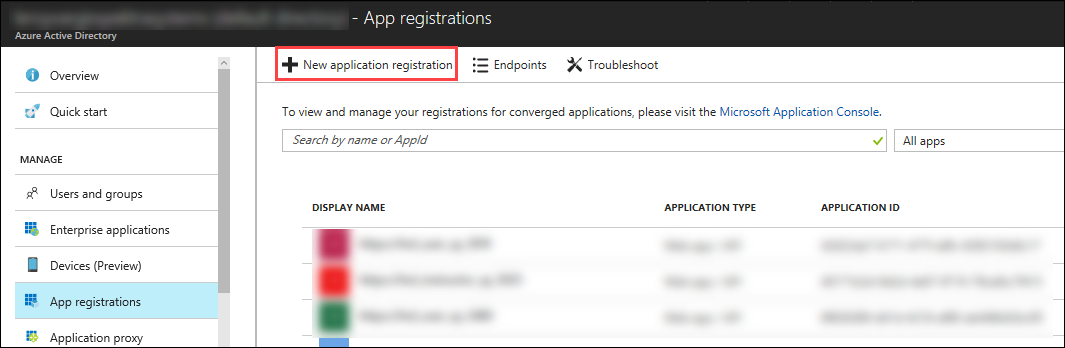
1. **Click** on the **Azure Active Directory** button in the **Menu** **navigation** bar to view the **Azure Active Directory** blade.



1. You will be directed to the Azure Active Directory blade, **click** on **App registrations.**



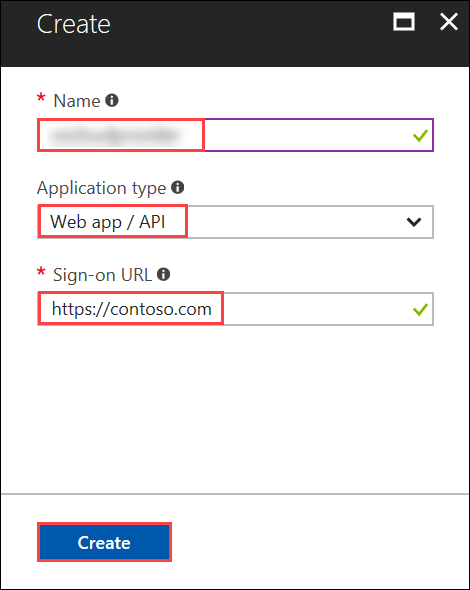
1. In the next blade, **click** on **New Application Registration on top of the blade.**

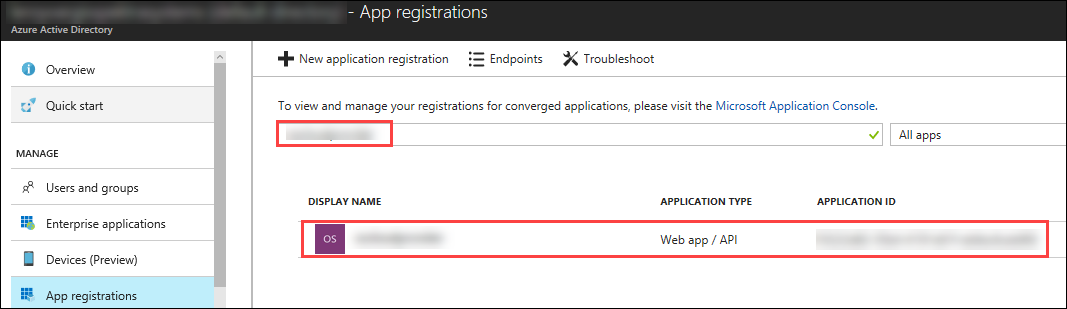


1. In the **Create** blade, **configure** as follows:

* Name: **(Provide a unique value)**
* Application type: **Web app/API**
* Sign-on URL: <https://contoso.com> **Note:** We will change this value later during the lab.

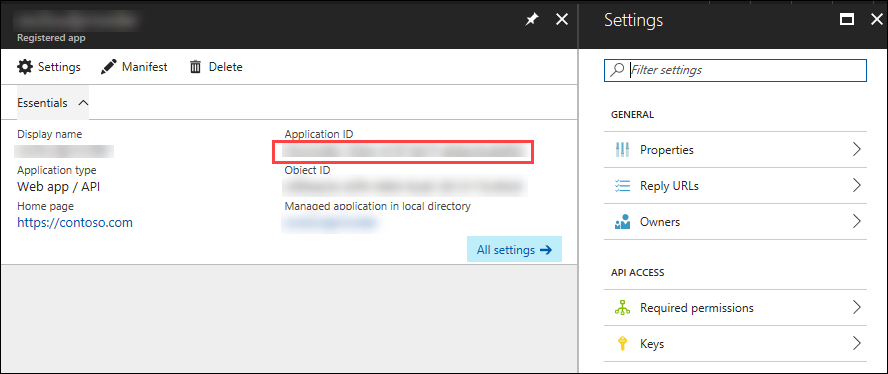
And then **click** on **Create.**



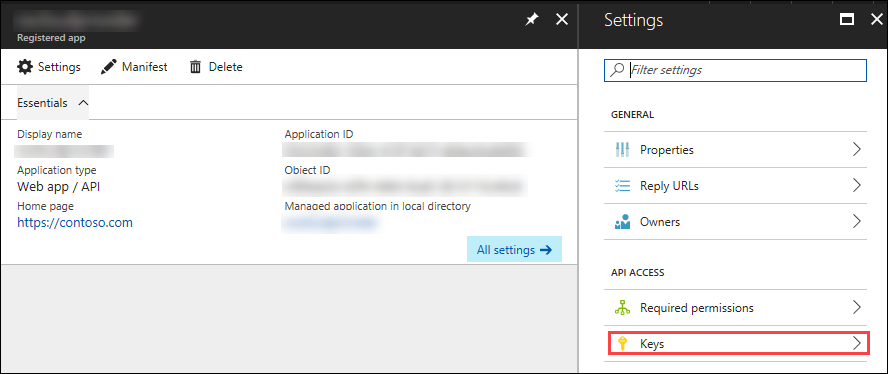
1. You will be redirected to the **App registrations** blade. You can check the app has been created by typing the App Name in the search field.  
   

If the app has been created, you can see it in the results as shown above.

1. Click on the app you created and you will be directed to the App blade.
2. Copy the Application Id and save it in a notepad or any text editor for later use.



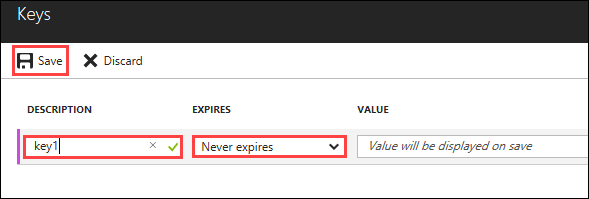
1. Now, **Click** on **Keys** in the settings blade.



1. In the **Keys** blade, **configure** as follows:

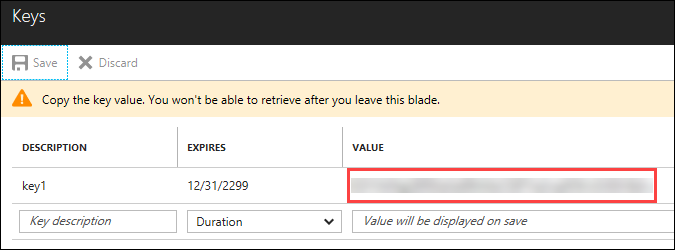
* Description: **key1**
* Expires: **Never expires**

And **Click** on **Save.**



1. After you click on save, the key value will be displayed which is the Client Secret.

Copy the value into the text editor where you saved the value of Application Id for later use.



Exercise 2: Create a Keyvault

In this exercise, you will configure Azure Bash Cloud Shell and create a Key vault in the existing resource group and store the SSH key inside the vault.

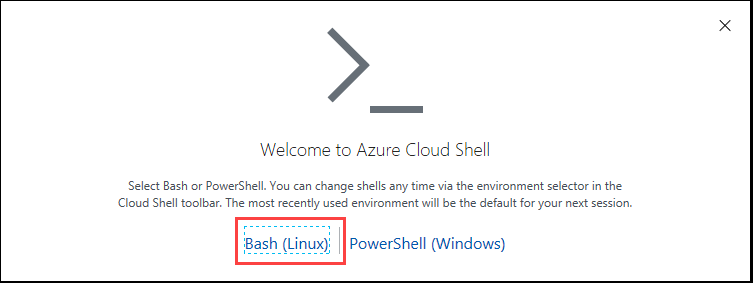
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.

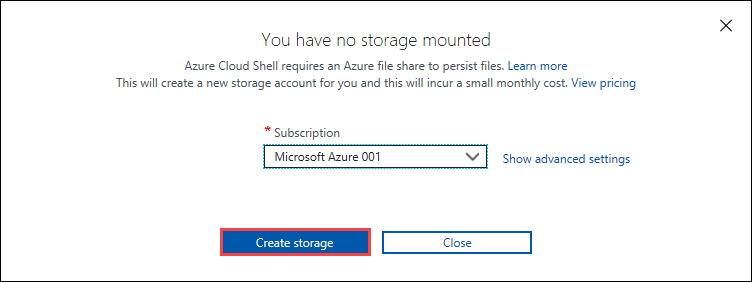


1. **Click** on **Cloud Shell** at the top right corner of the screen,to open the cloud shell.

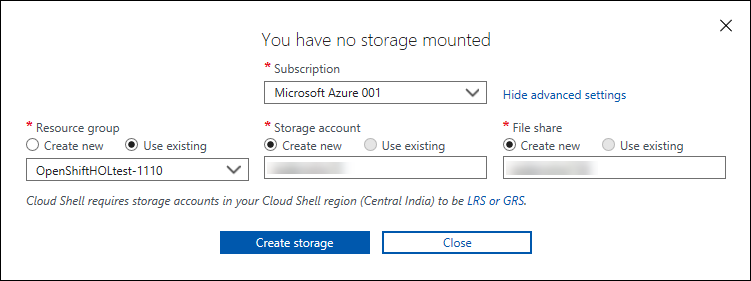


1. Then **Click** on **Bash ( Linux )**, and in the next page, **click** on **Show advanced settings**

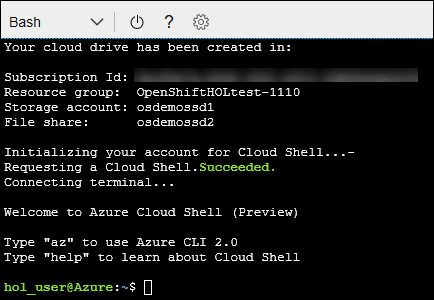




1. In the new blade, select the existing resource group, provide unique names under Create new(Storage account and File share) and **click** on **Create Storage.**



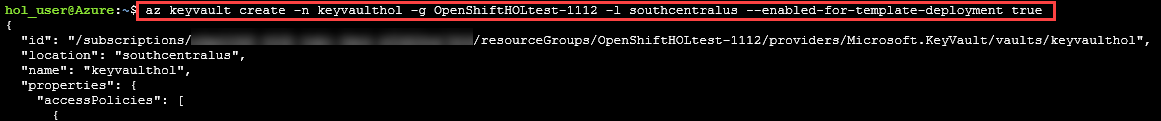
1. In a few minutes, the bash shell will come up.



1. Now execute the following command in the cloud shell to create a keyvault in the existing resource group.

az keyvault create -n **<uniquename>** -g **<ResourceGroup>** -l southcentralus --enabled-for-template-deployment true

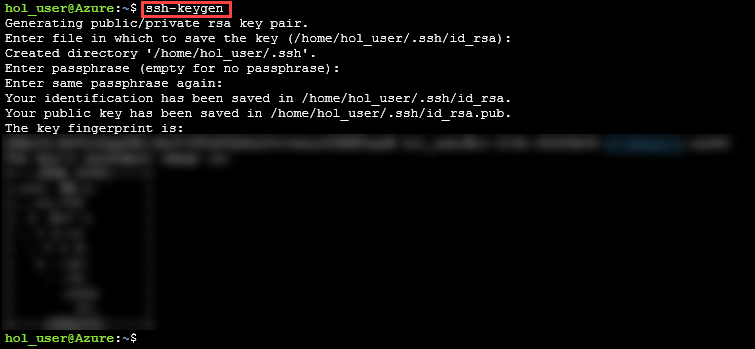
**Note**: Provide the existing Resource Group name and a unique name for keyvault in the above command when executing



1. Now execute the following command in the cloud shell to generate ssh key.

ssh-keygen

**Note**: Keep on clicking enter button until the key has been created.



1. Now execute the following command in the cloud shell to display the public ssh key. Copy the key into a text editor for later use.

cat .ssh/id\_rsa.pub

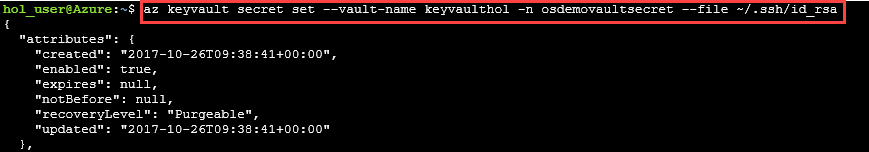
**Note**: You will need this key for later use.



1. Now execute the following command to store the generated key in the key vault.

az keyvault secret set --vault-name **<keyvaultname>** -n osdemovaultsecret --file ~/.ssh/id\_rsa

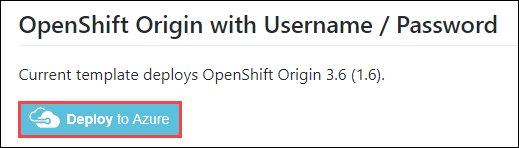
**Note**: Substitute for key vault name in the above command with the name of the keyvault created earlier when executing.



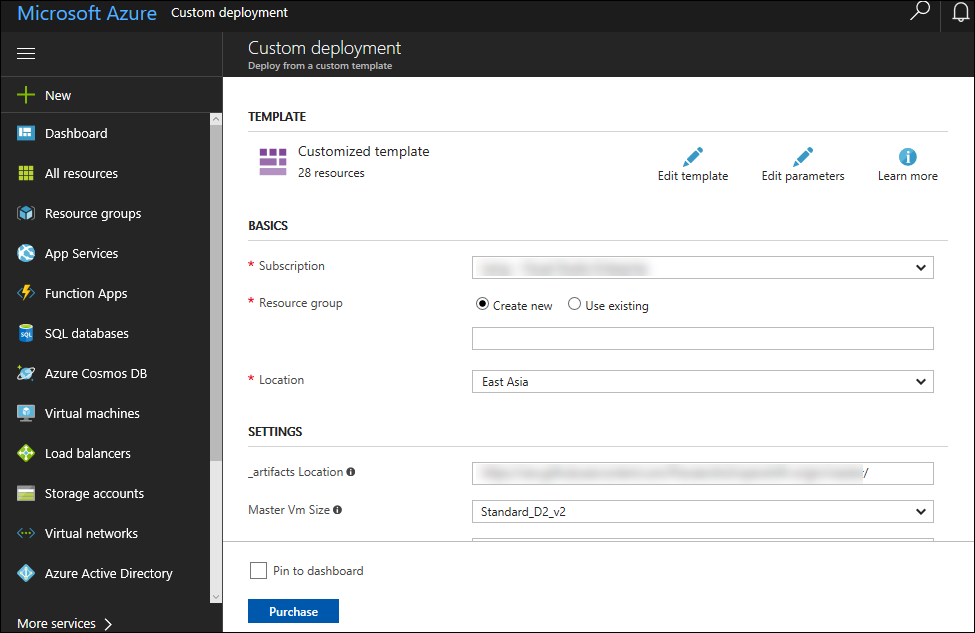
Exercise 3: Deploy Openshift Cluster using ARM Template

In this exercise, you will deploy the Openshift cluster on Azure using ARM Template .

1. **Launch** a browser and **Navigate** to <https://github.com/PraveenAnil/openshift-origin/blob/master/README.md>.
2. Now **click** on **Deploy to Azure** button and you will be redirected to the azure portal. If prompted **login** with the Microsoft Azure credentials you received via email.

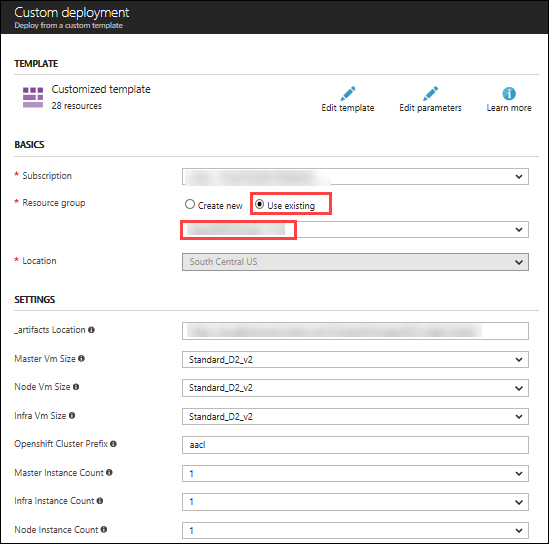


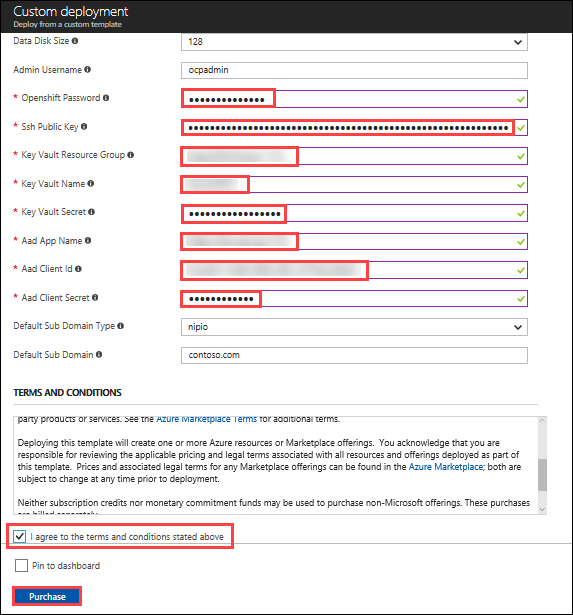
1. Now you will be directed to the Custom Deployment blade.



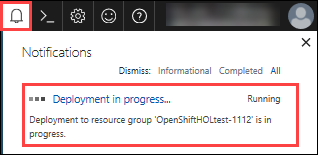
1. In the **Custom Deployment** blade, **configure** the settings as follows:

* Resource Group : Choose **Use existing** and scroll down to see the Resource Group which is already there)
* Openshift Password : **Provide a unique password**
* Ssh Public Key : **Provide the copied SSH key**
* Key Vault Resource Group : **Provide the existing resource group name**
* Key Vault Name : **Provide the Key Vault name you provided**
* Key Vault Secret : **osdemovaultsecret**
* Aad App Name : **Provide the name of the AD App you created**
* Aad Client Secret : **Provide the secret key of the AD App**

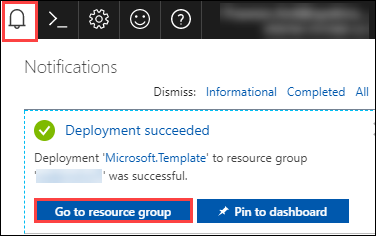




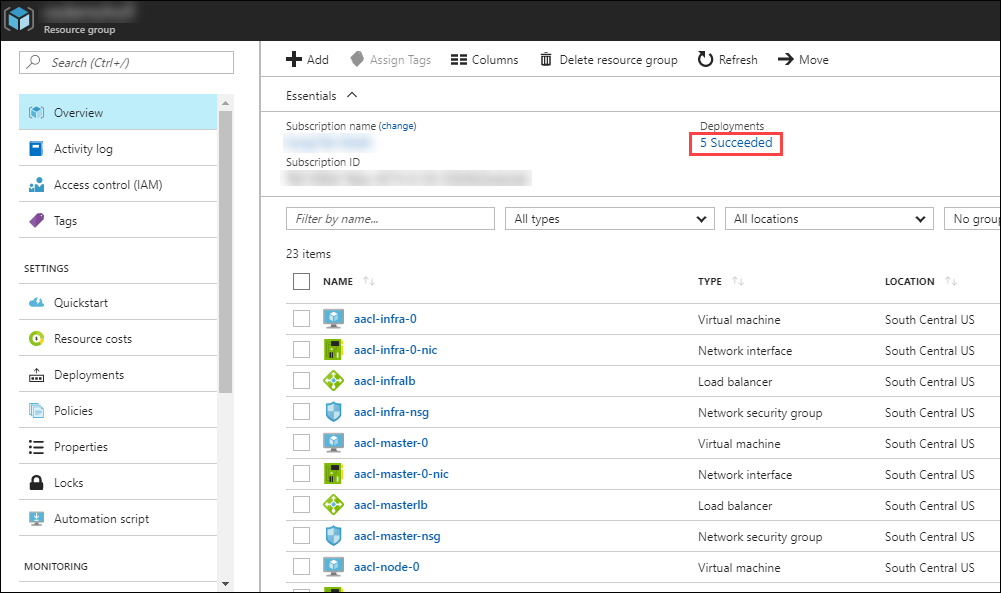
1. And then click on Purchase.
2. Once the deployment starts, you can see the progress in the notification bar at the top of the Azure portal.



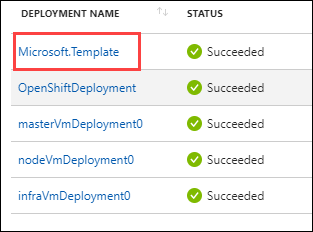
1. Once the deployment is complete, you can see it in the notifications tab as Deployment succeeded. Now, click on Go to resource group from the notifications tab.



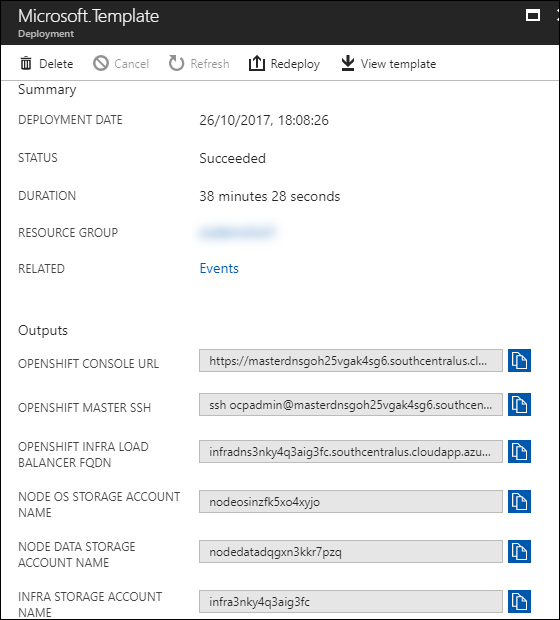
1. In the resource group blade that come up, you can see the deployments as Succeeded, click on that.



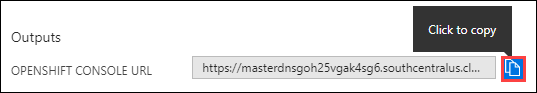
1. Select **Microsoft Template** from the new blade that come up.



1. From the new blade that come up, you can see the outputs of the deployment.

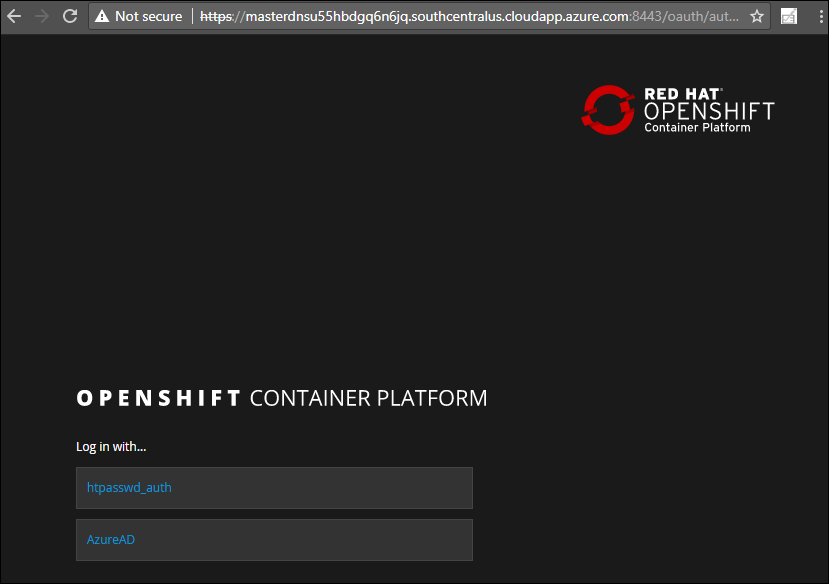


1. Copy the Openshift Console URL by clicking on Copy



1. To verify that the deployment is working, **Open** a new tab in the browser and paste the copied URL.

**Note:** Skip the certificate warning

1. Now you will be directed to the Openshift Console Login page.  
   

Note: If the above page comes up, then the deployment is working.

Exercise 4: Configure Azure AD Authentication

In this exercise, you will configure the AD App you created for Authentication into the Open Shift console.

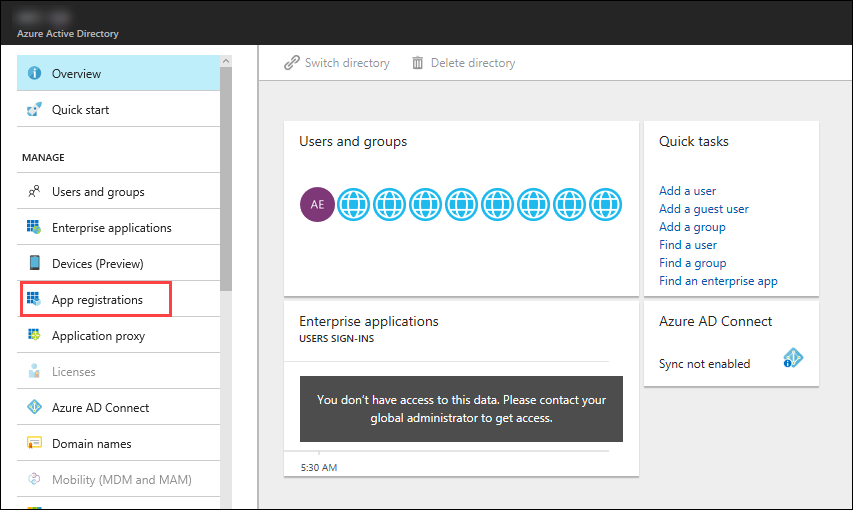
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.

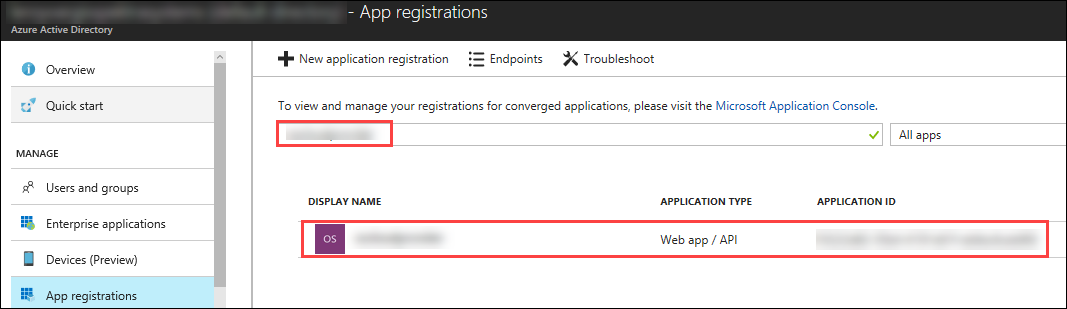


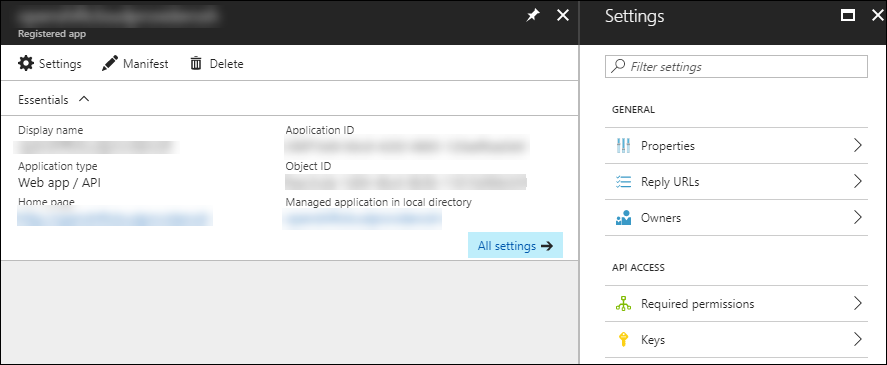
1. **Click** on the **Azure Active Directory** button in the **Menu** **navigation** bar to view the **Azure Active Directory** blade.



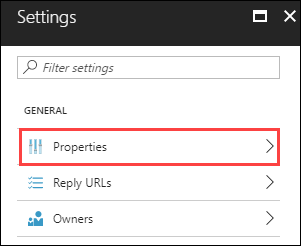
1. You will be directed to the Azure Active Directory blade, **click** on **App registrations.**



1. You will be redirected to the **App registrations** blade. You can search the App by typing the name of the App you created earlier, in the search field.  
   
2. Click on the app you created and you will be directed to the App blade.



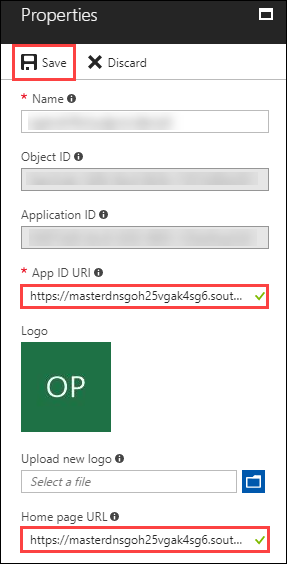
1. Now Click on Properties under Settings blade.



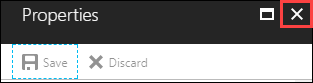
1. In the **Properties** blade, **edit** as follows:

* App ID URI: (Provide the Open Shift Console URI)
* Home Page URL type: (Provide the Open Shift Console URI)

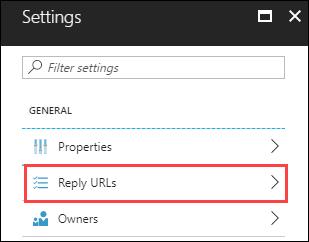
And then **click** on **Save.**



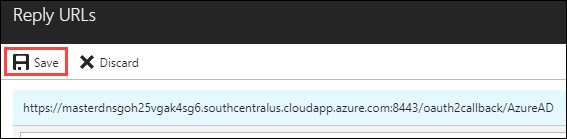
1. Once you save the properties, close the properties blade.



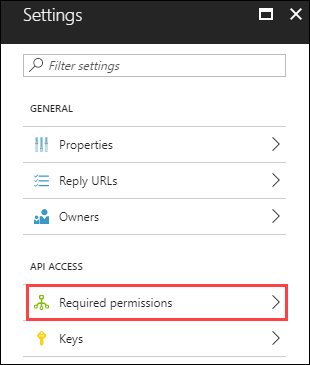
1. Then you will be redirected to the Settings Blade of AD App. Click on the Reply URLs



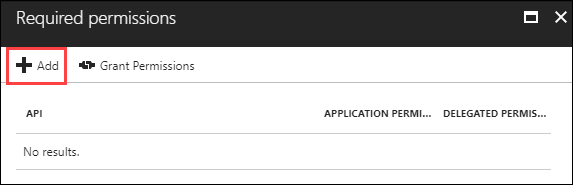
1. Now modify the openshift console url by removing the ‘console’ from the end and appending ‘oauth2callback/AzureAD’ to the url and provide it in the Reply URL blade that come up and then Click on Save.

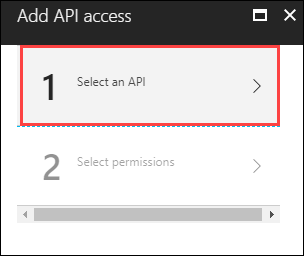


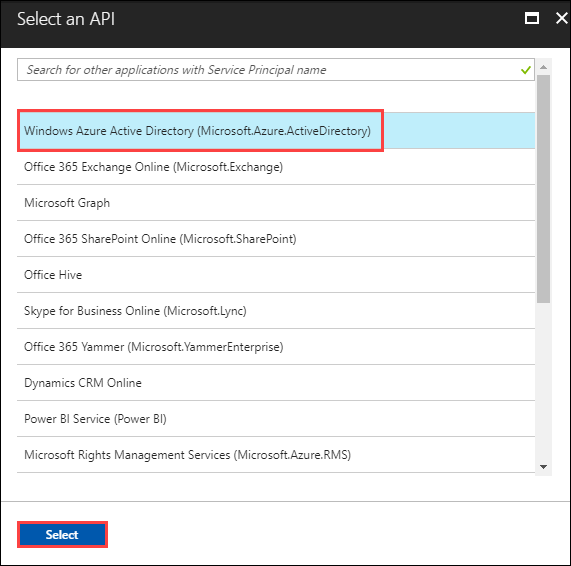
1. Now go back to the setting blade of the App and Click on Required permissions.



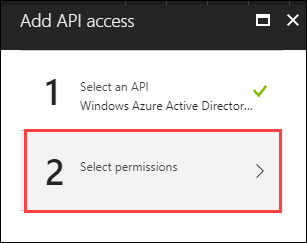
1. Click on +Add from the Required permissions blade that come up and from the new blade that come up, click on Select an API and then select Windows Azure Active Directory and click on **Select** button .



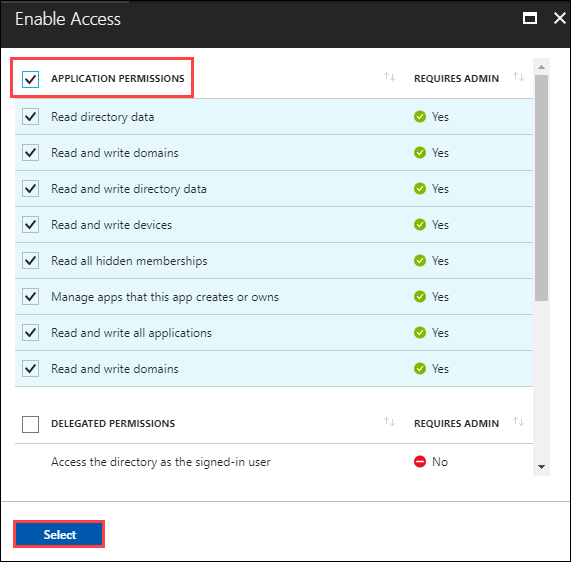




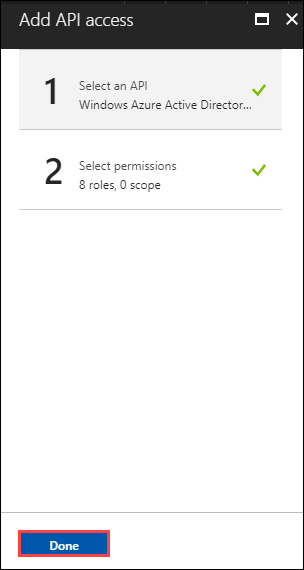
1. Now you will be directed to the Add API access blade. Click on Select permissions.



1. From the new Enable Access blade that come up, select Appliction Permissions and then Click on Select button at the bottom of the blade.

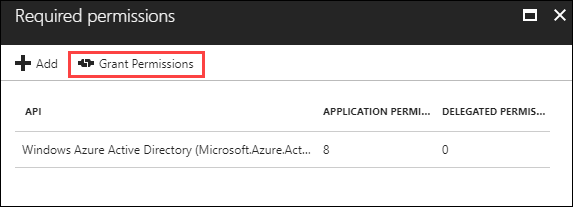


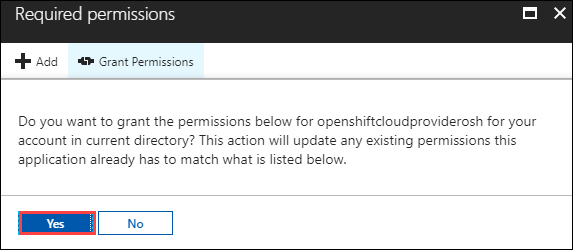
1. Now you will be directed to the API access blade, click on Done to add the permissions.



1. Once the permissions are added, you can will be directed to the Required permissions blade.

Click on Grant permissions and then click Yes.

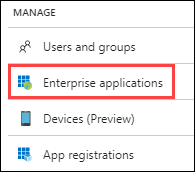




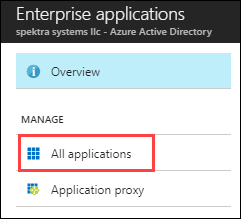
1. Now go back to the Active Directory blade by clicking on **Azure Active Directory** button in the **Menu** **navigation** bar.



1. Click on Enterprise Applications from the menu on the left side.



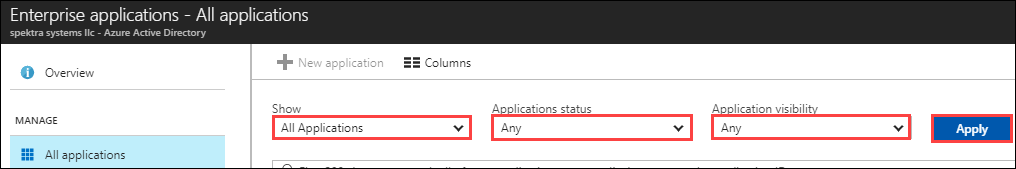
1. In the new blade that come up, click on **All applications**



1. In the new blade that come up, edit the filter as follows:

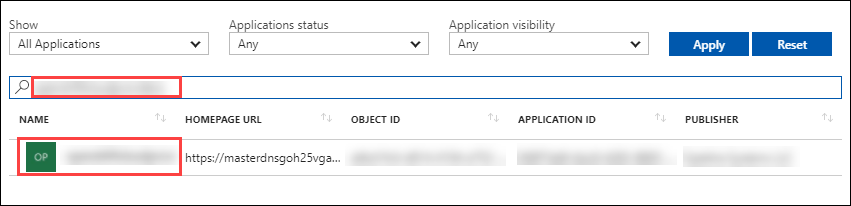
* Show: Select All Applications
* Application status: Any
* Application visibility: Any

And then **click** on **Apply.**

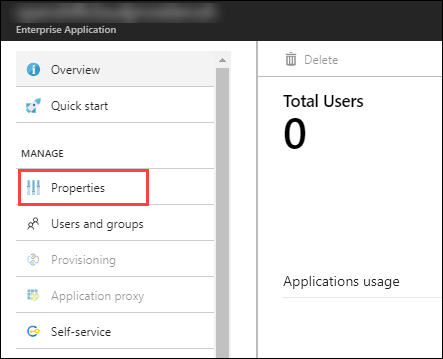


1. You can search the App by typing the name of the App you created earlier, in the search field.

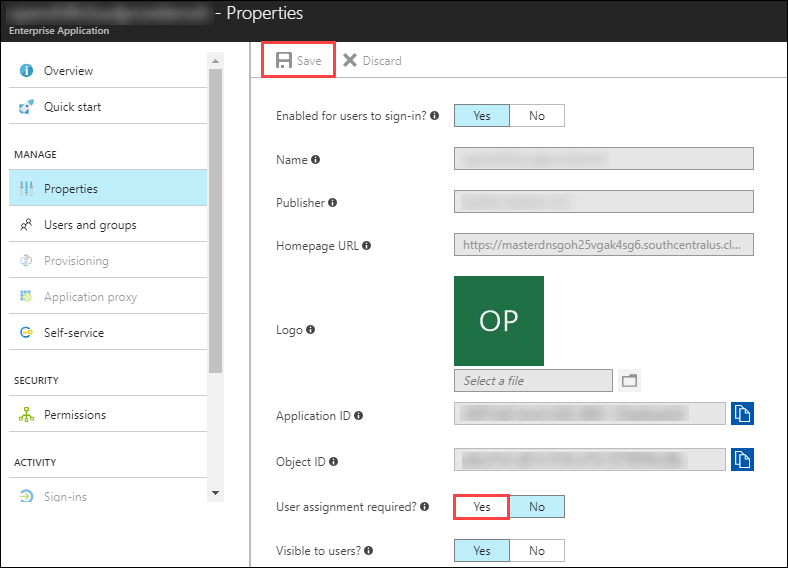
Select the App from the results.



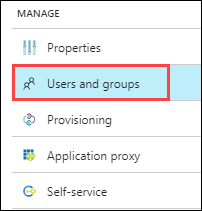
1. You will be redirected to the App Blade. Click on Properties under Manage section on the left side of the properties blade.



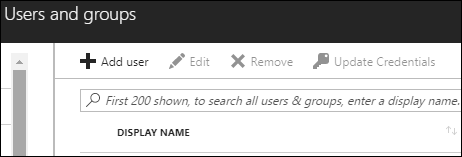
1. In the new blade that come up, select User Assignment Required and Click on Save.



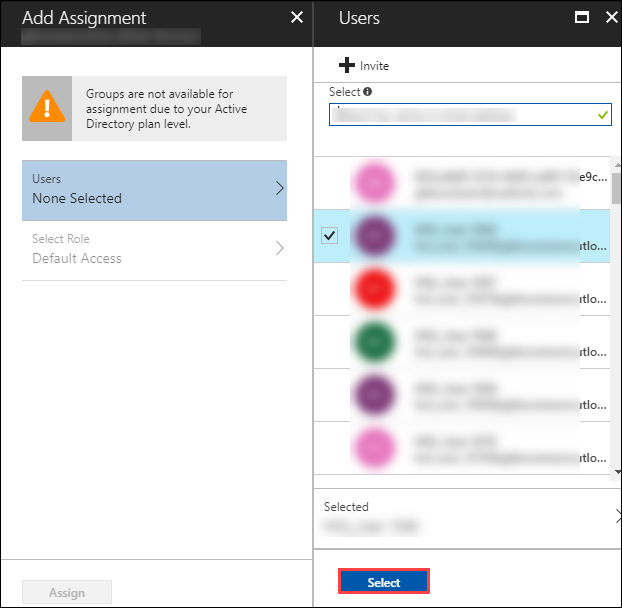
1. Now, click on **Users and groups** under Manage section on the left side of the App Blade.



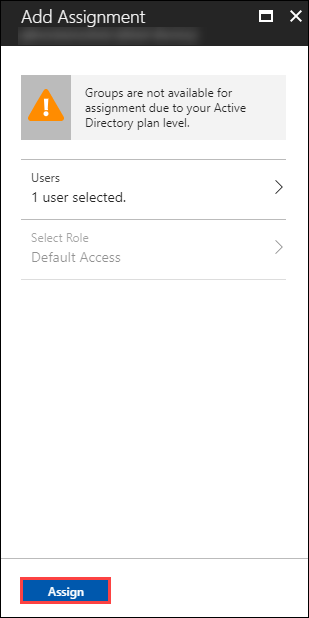
1. In the blade that come up, click on **+Add user** to assign a user to the app.



1. In the Add Assignment blade that come up, click on Users and then select the id with which you logged in to Azure portal and click on **Select**.

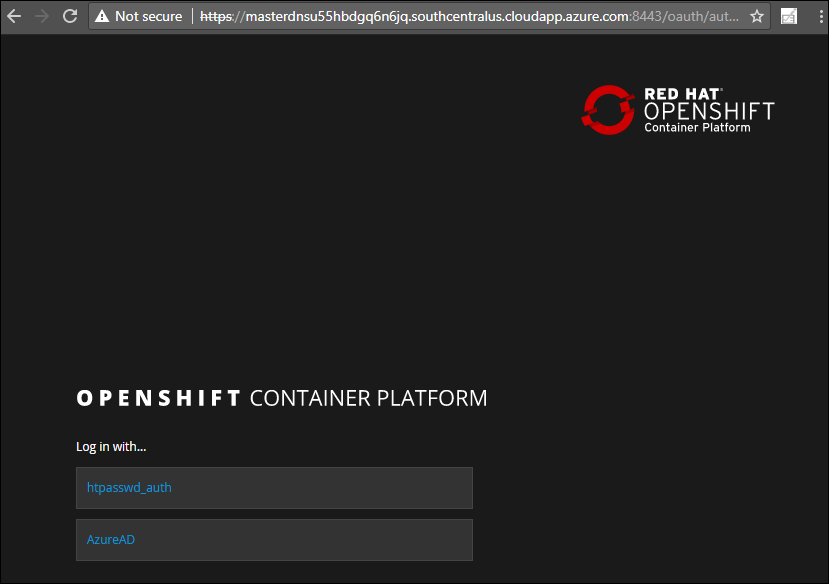


1. Now you will be redirected to the Add Assignment blade. Click on Assign to assign the user to the app.

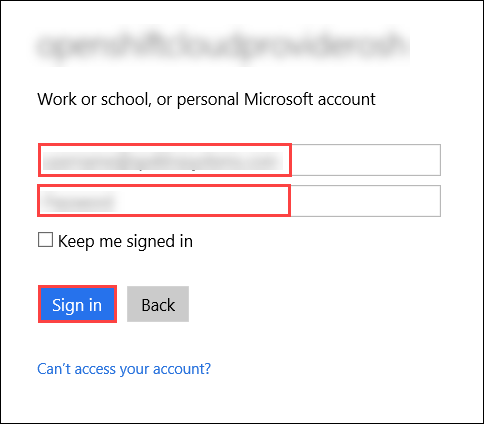


1. Now to verify that the user is able to authenticate to Openshift console via Azure AD, **Open** a new tab in the browser and paste the Openshift Console URL which you copied earlier.

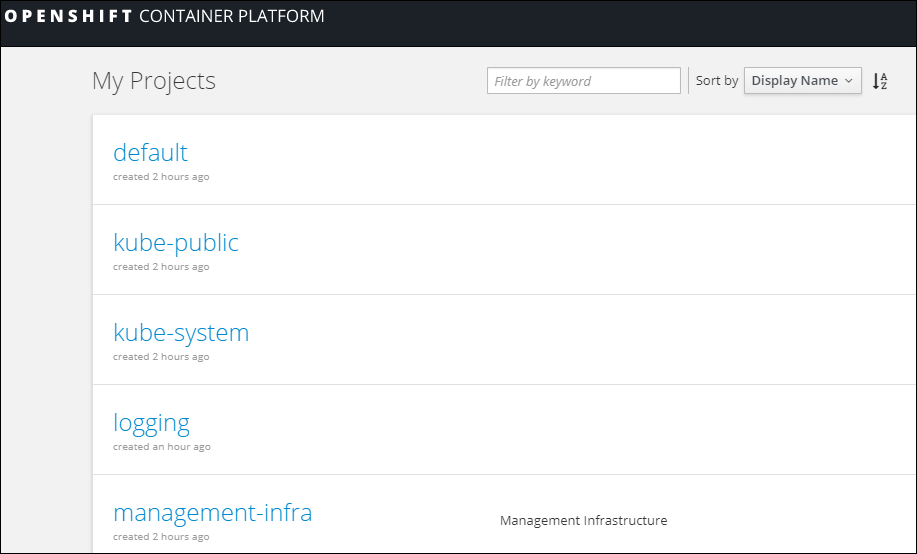
**Note:** Skip the certificate warning



1. Now click on AzureAD, you will be redirected to the Login Page. Provide the Azure credentials you received via email over there and click on Sign in.



1. Once the login is successful, you will be redirected to the Openshift console.



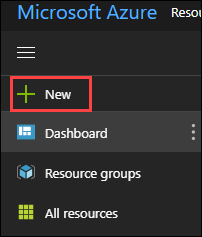
## Exercise 5: Deploy a 2 Tier Node JS Application on Open Shift

In this exercise, you will deploy a 2 tier Node.js app on Open Shift and configure it to use the DB on Azure.

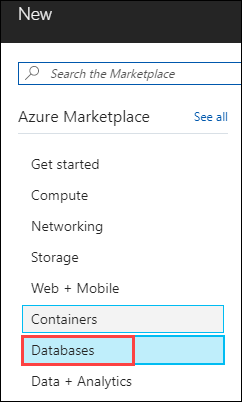
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.



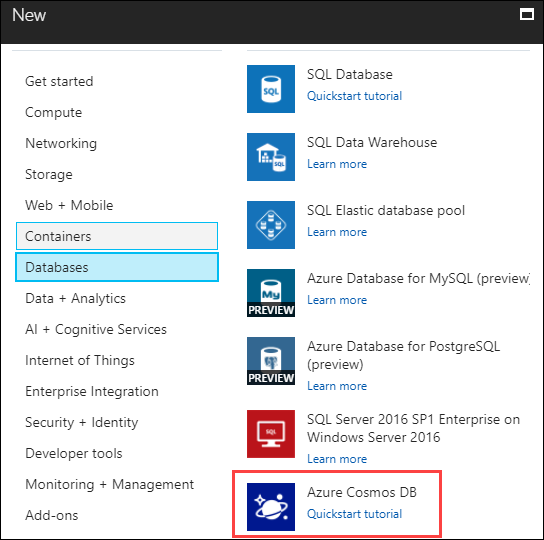
1. Click on +New on the left side of the Dashboard.



1. In the **New** blade that come up, Select **Databases.**

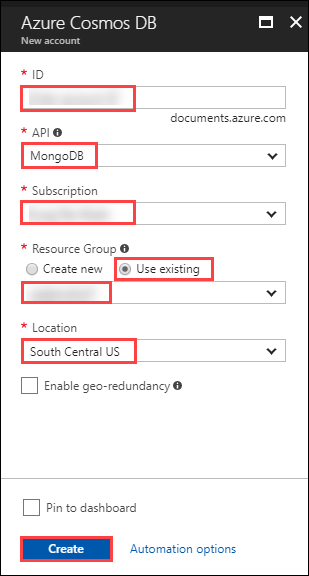


1. In the **Databases** blade appears. Select **Azure Cosmos DB.**



1. In the create blade that come up, configure the settings as follows:

* ID : **uniquename** (This name should be unique across Azure.)
* API : **MongoDB**
* Subscription : Select the existing subscription
* Resource Group : Choose **Use existing** and scroll down to see the Resource Group which is already there and select that)
* Location: **South Central US**

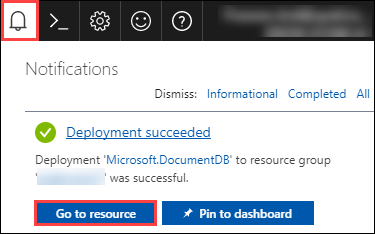


And then Click on Create.

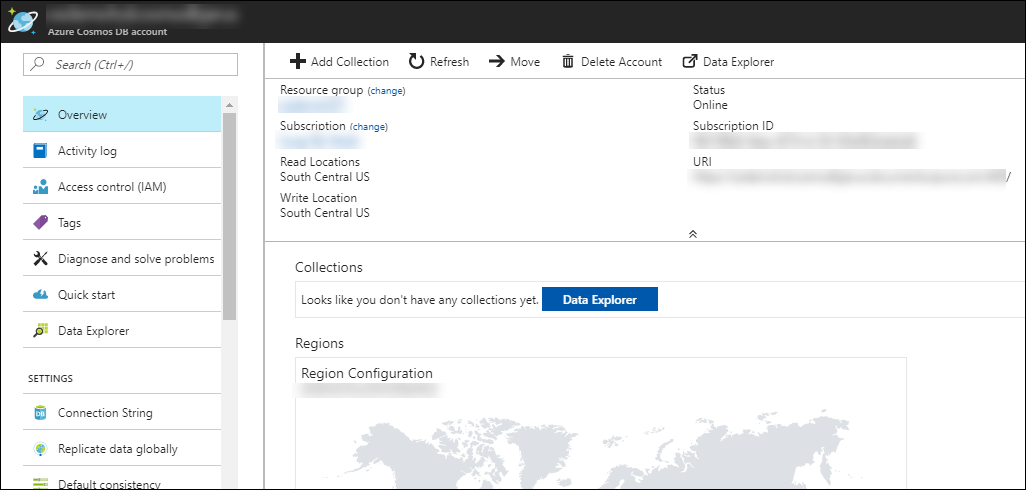
1. You can see the status of the deployment from the notifications tab on top of the page.



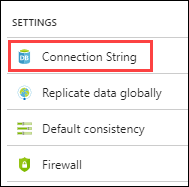
1. Once the deployment is successful, click on Go to resource from the notifications tab.



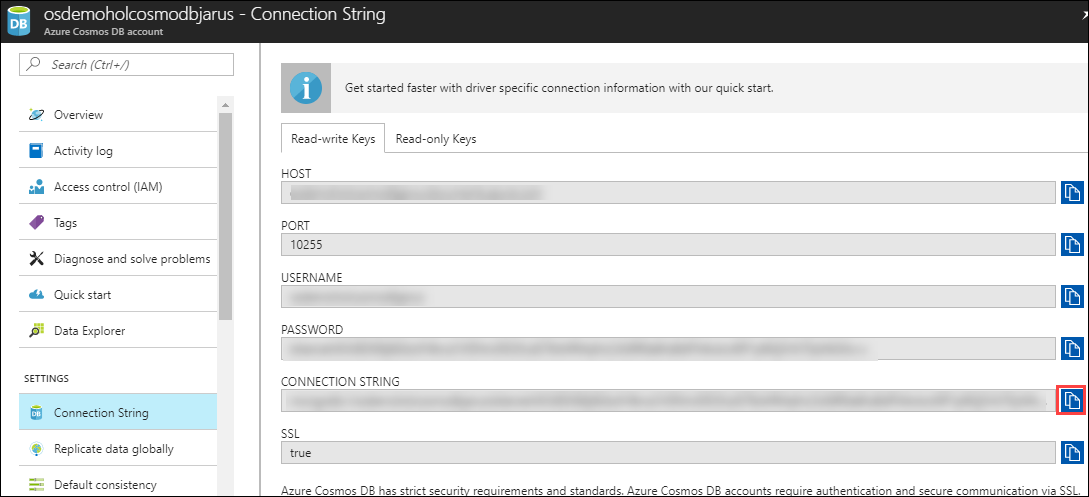
1. Now you will be directed to the deployed database.

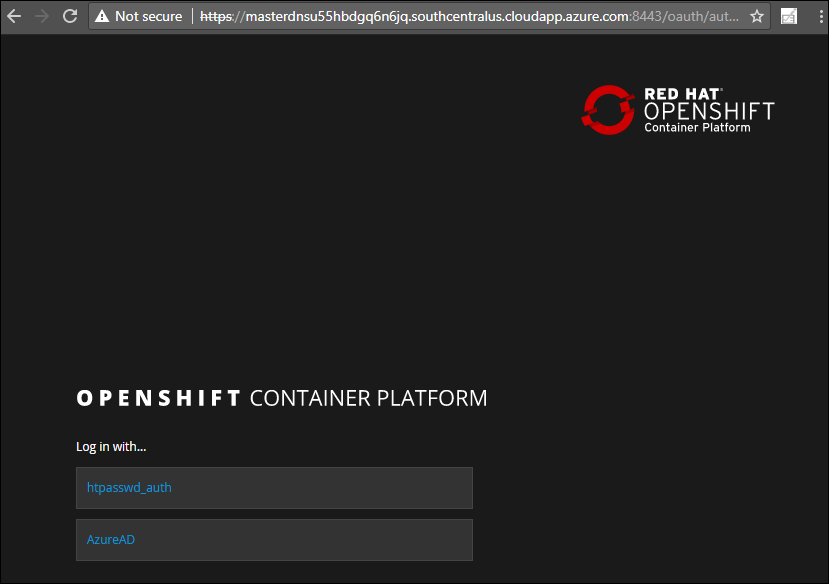
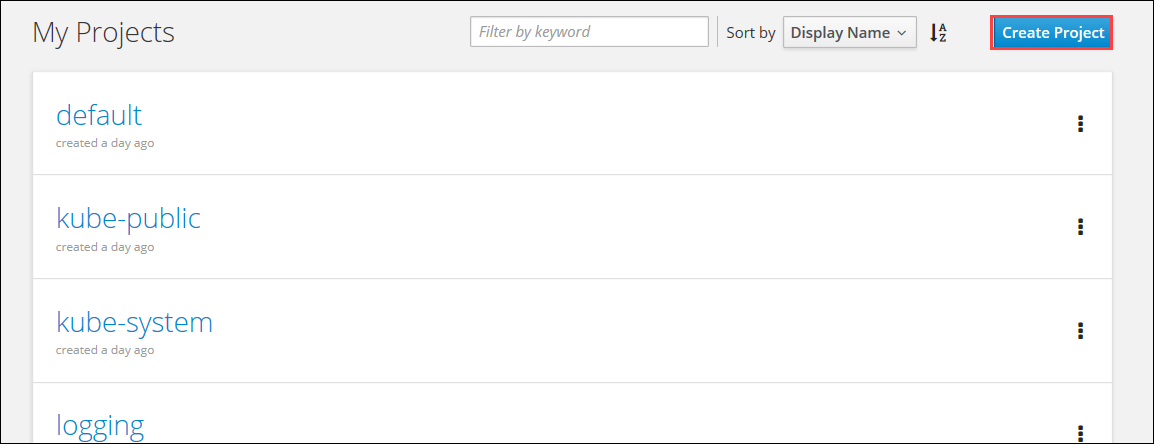


1. Now, click on Connection String under settings menu on the left side of the blade.



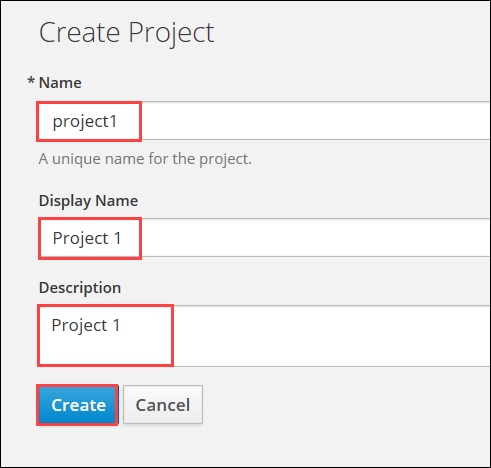
1. Now from the new blade that come up, copy the connection string for later use.



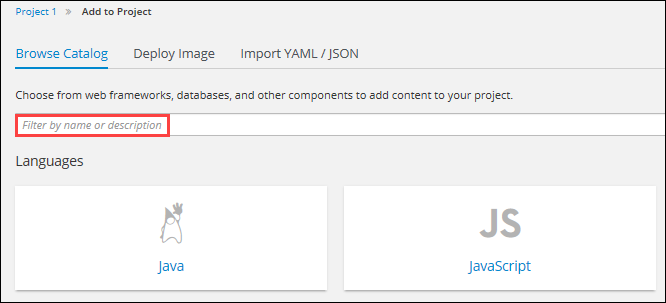
1. Now, open a new tab in a broswer and navigate to the Openshift console url. Login into the Openshift console using the credentials you received via email by Selecting AzureAD as authentication type.  
   
2. Once you are logged in to the console, click on Create Project   
     
   
3. **Enter** the **following** **details** into the Create Project Section, and click **Create**.

* *NAME:*  **project1**
* *DISPLAY NAME:* **Project 1**
* *DESCRIPTION:* **Project 1**

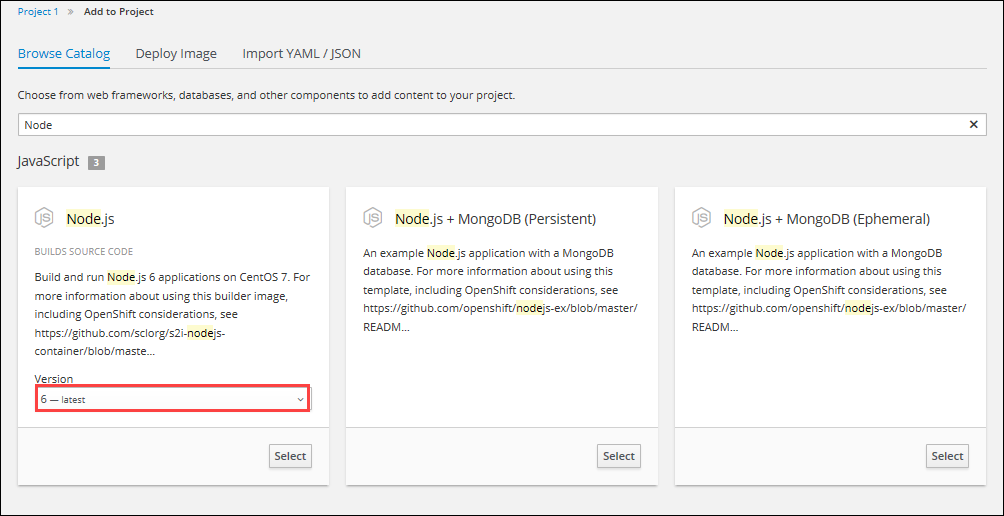
And then **Click Create.**



1. Now you will be redirected to the Add to Project section.

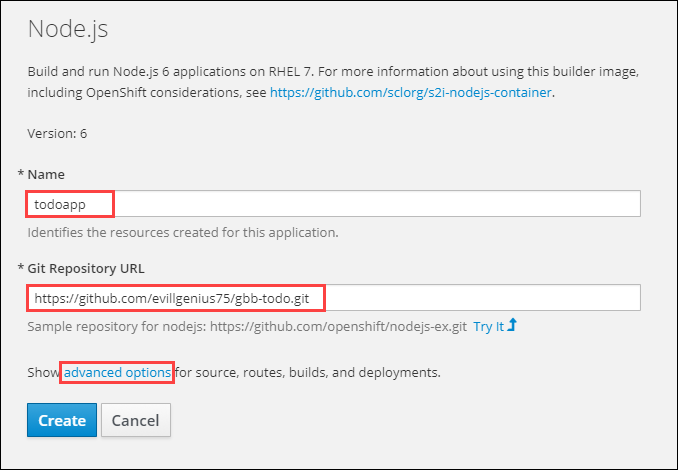


1. Search **Node** in the Filter name or description area and from the results that come up, change the Version of Node.js to 6-latest and then click on **Select**



1. Now in the new page that comes up, **Provide** the **following configuration** in the blade that come up.

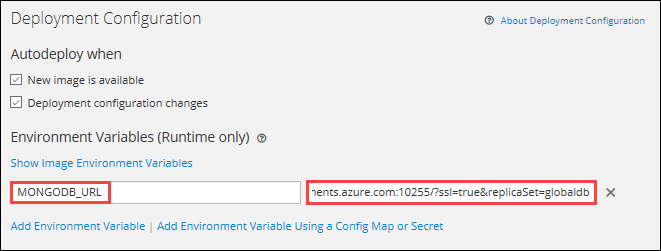
* Name: **todoapp**
* Git Repository URL: <https://github.com/evillgenius75/gbb-todo.git>



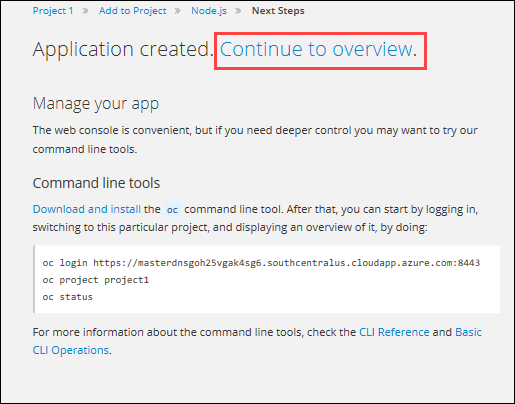
1. Now Click on Advanced settings and scroll down till you see Deployment Configuration section and provide the following:

* Name: **MONGODB\_URL**
* Value: (Provide the connection string value of Azure Cosmos DB which you created earlier)

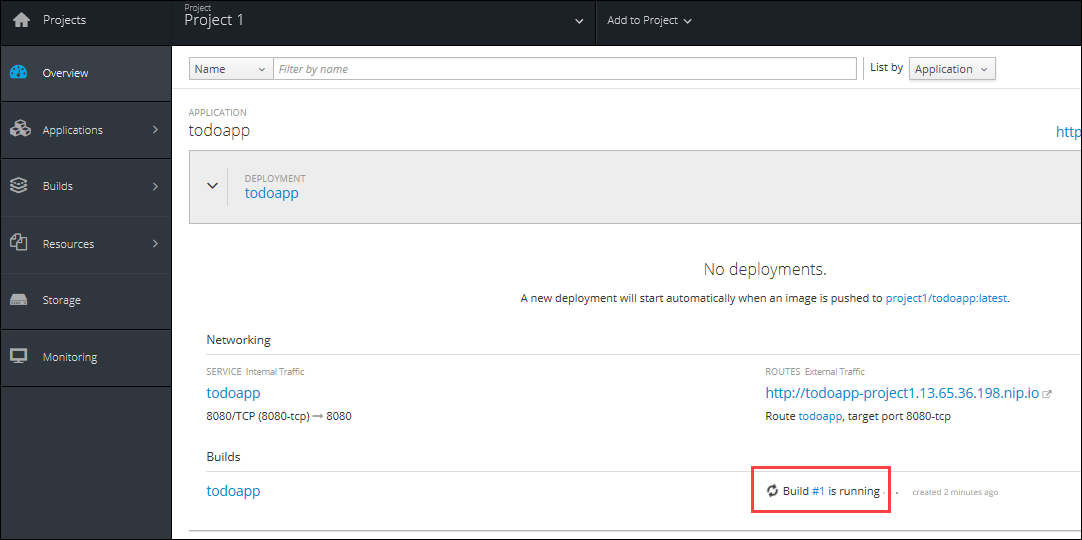
And then click on **Create.**



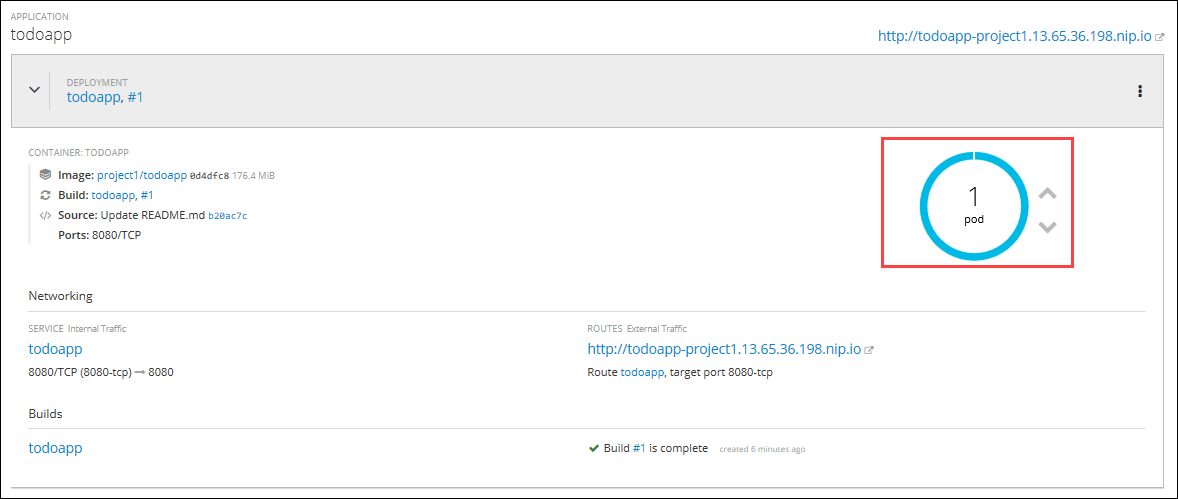
1. Once you click on Create, you will be redirected to Application Created page. Click on Continue to Overview page.



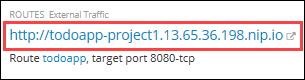
1. Now you will be directed to Project overview page and you can see that the build is running on the bottom right section of the page.

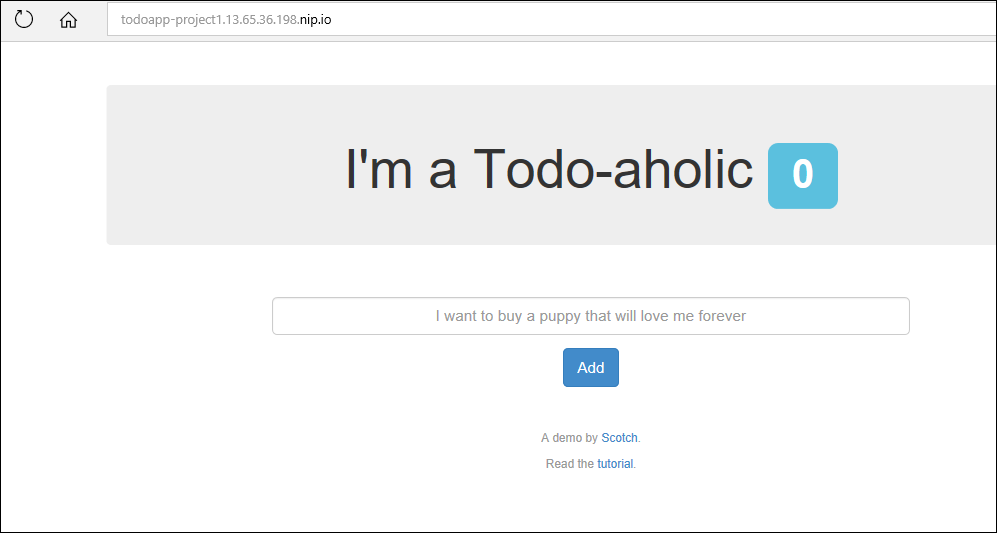


1. Wait until the build is completed and the colour of the circle surrounding the pod turn blue.



1. Once the build is completed and colour of the circle around pod turns blue, click on the url under Routes External Traffic.



1. You will be directed to the todoapp page.  
   
2. Now provide the following value in the text field above Add button and then click on Add Button.  
   **“Deployed a 2 Tier App on Openshift, check if the database is updated”**

