

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
- Workshop Architecture and Objective..... 1
 - Labs Objective 2
 - Workshop Architecture after deploying ARM Template 2
- Lab 1: Introduction to Azure Portal..... 3
 - Lab Overview` 3
 - Prerequisites..... 3
 - Time Estimate..... 4
 - Exercise 1: Log into your Azure Portal 4
 - Exercise 2: Verify access to the Subscription..... 5
- Lab 2: Deploying Open Shift cluster using ARM templates 8
 - Lab Overview 8
 - Prerequisites..... 8
 - Time Estimate..... 8
 - Exercise 1: Create an Azure AD Application 8
 - Exercise 2: Create a Keyvault 13
 - Exercise 3: Deploy Openshift Cluster using ARM Template 17
 - Exercise 4: Configure Azure AD Authentication 23
 - Exercise 5: Deploy a 2 Tier Node JS Application on Open Shift 36
 - Exercise 6: Integrate ACR with OpenShift 48

Workshop Architecture and Objective

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)

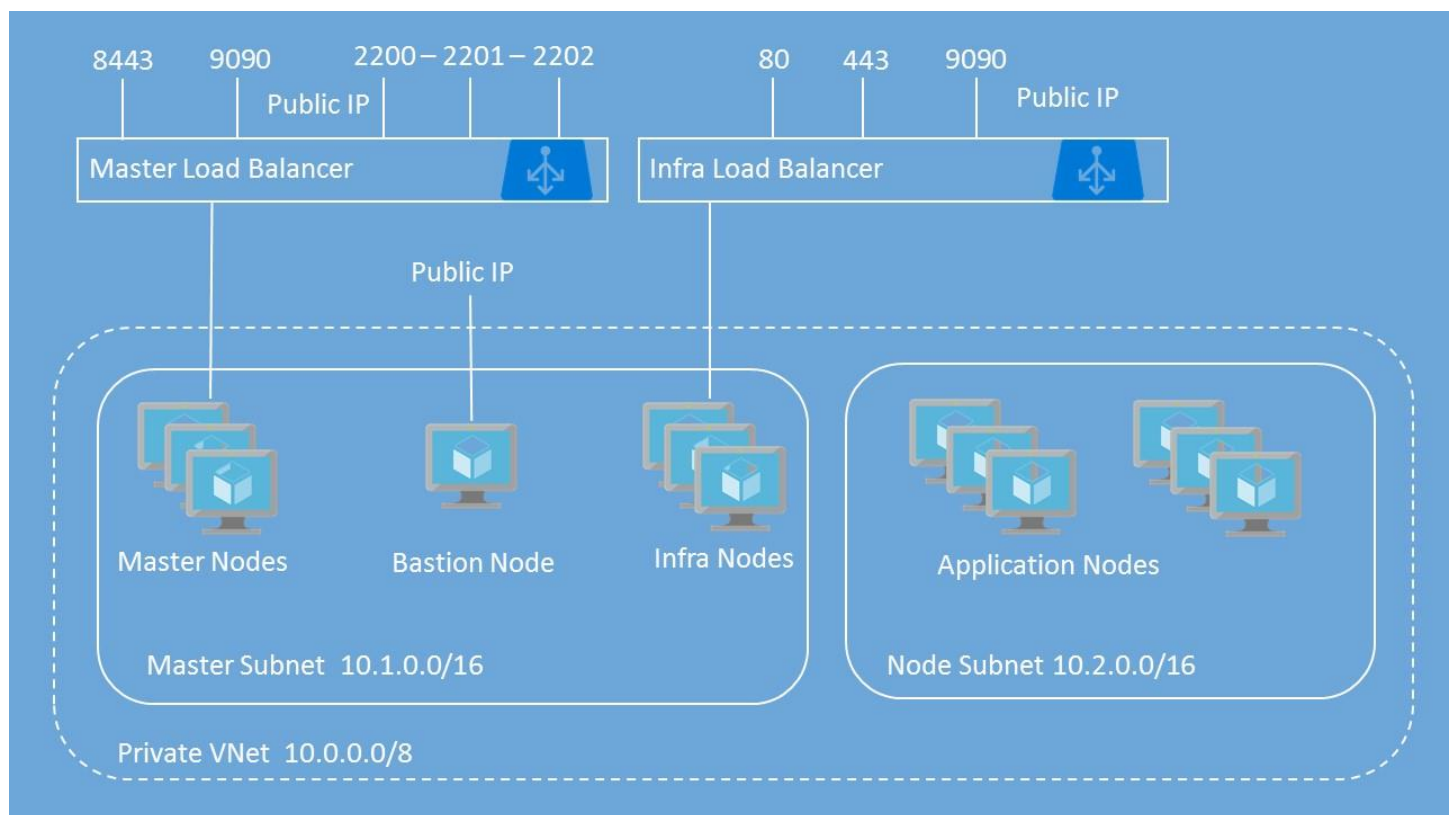
Labs Objective

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
- Integration of Azure Container Registry with Open Shift

Workshop Architecture after deploying ARM Template

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



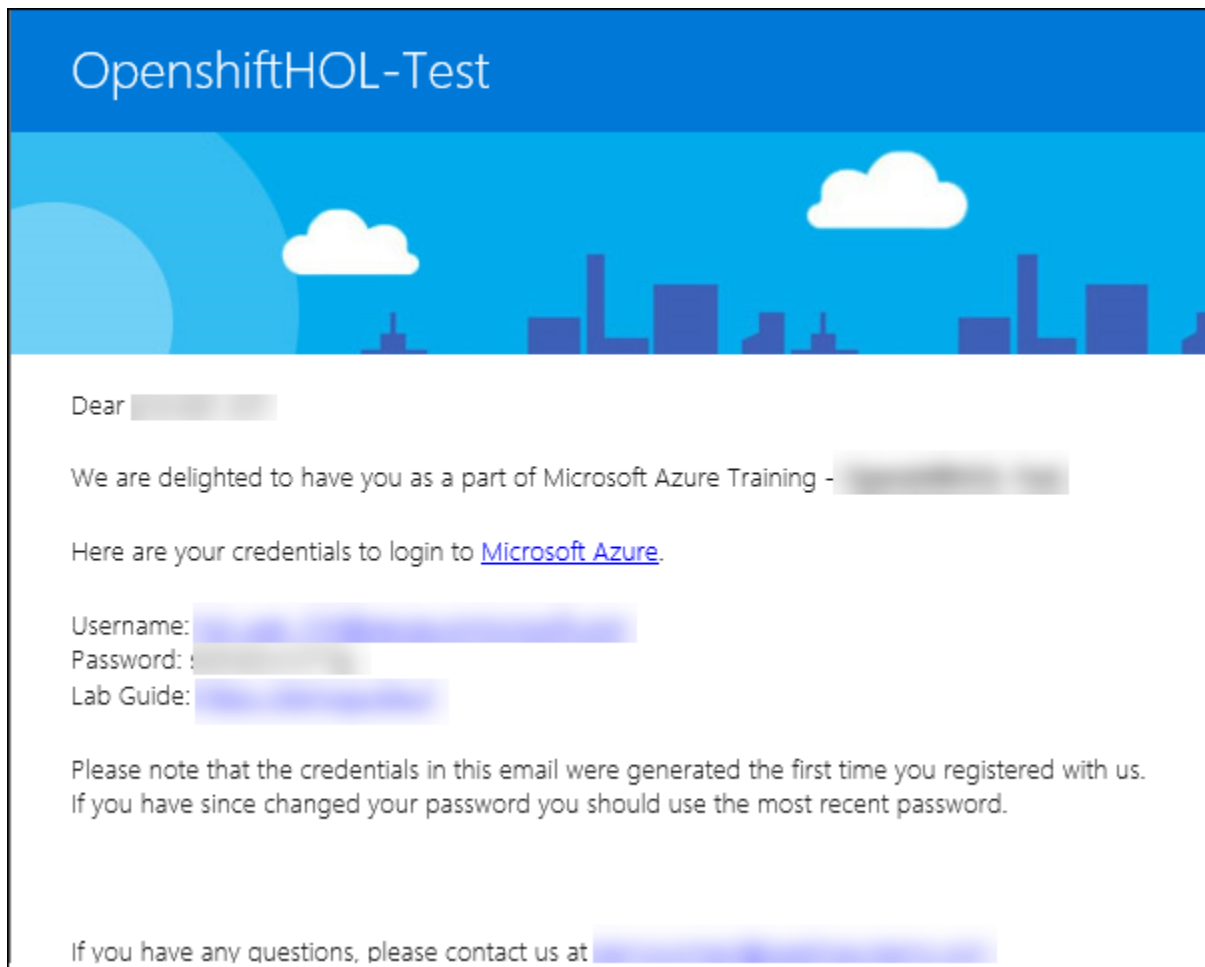
Lab 1: Introduction to Azure Portal

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](#).
- Red Hat Customer Portal login credentials so that the Azure instances can be registered with Red Hat Subscription Manager properly, and you must have enough OpenShift Container Platform entitlements to cover the chosen configuration.



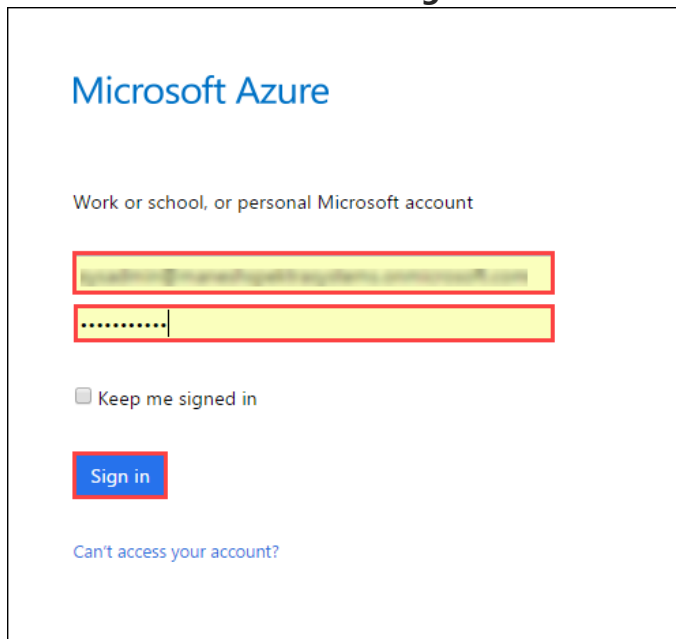
Time Estimate

10 minutes

Exercise 1: Log into your Azure Portal

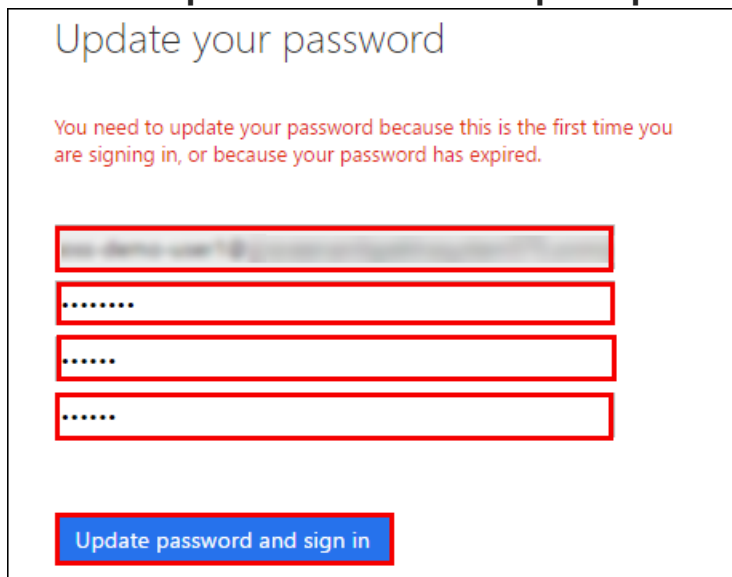
In this exercise, you will log into the Azure Portal using your Azure credentials.

1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. Provide the credentials that you received via email. Click on **Sign In**.

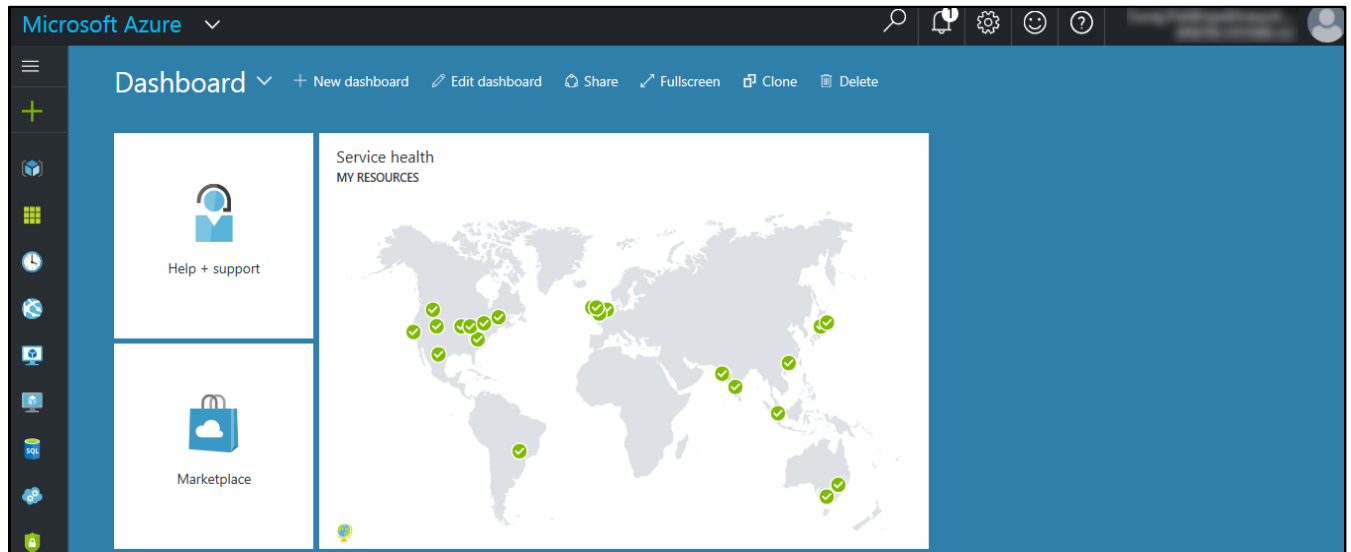


Note : At the first login, you may have to change the password, if asked for.

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



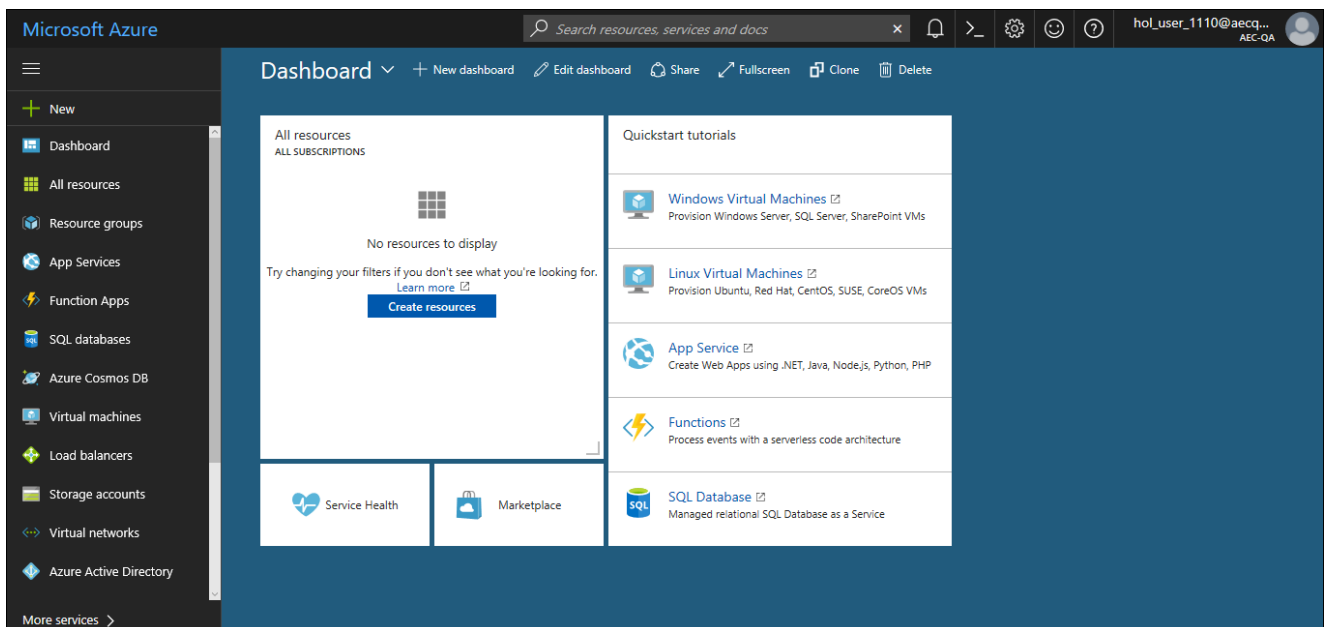
3. Now, you will be directed to the Azure Dashboard



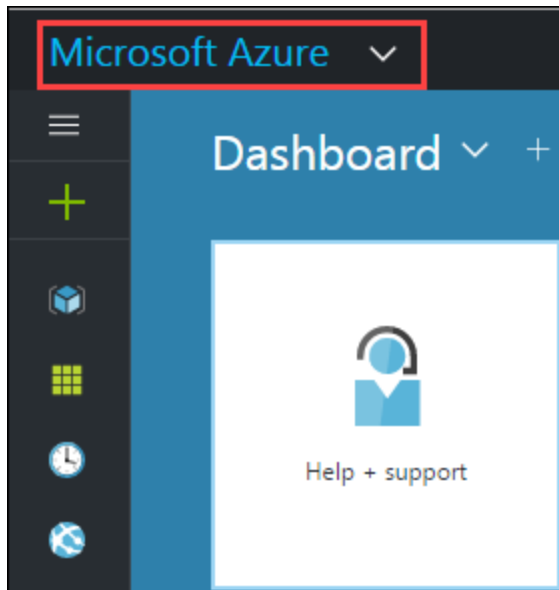
Exercise 2: Verify access to the Subscription

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

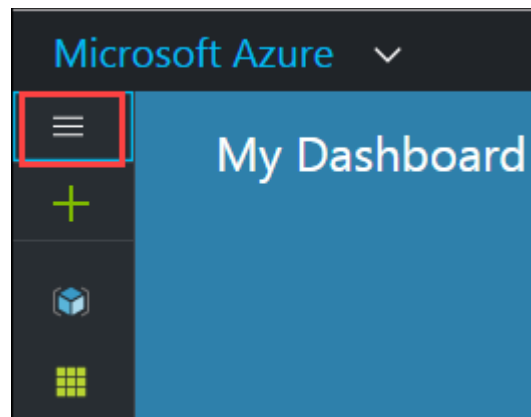
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.



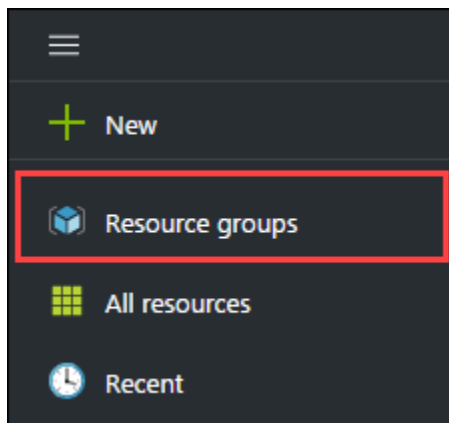
2. **Click** on **Microsoft Azure** at the top left corner of the screen, to view the Dashboard.



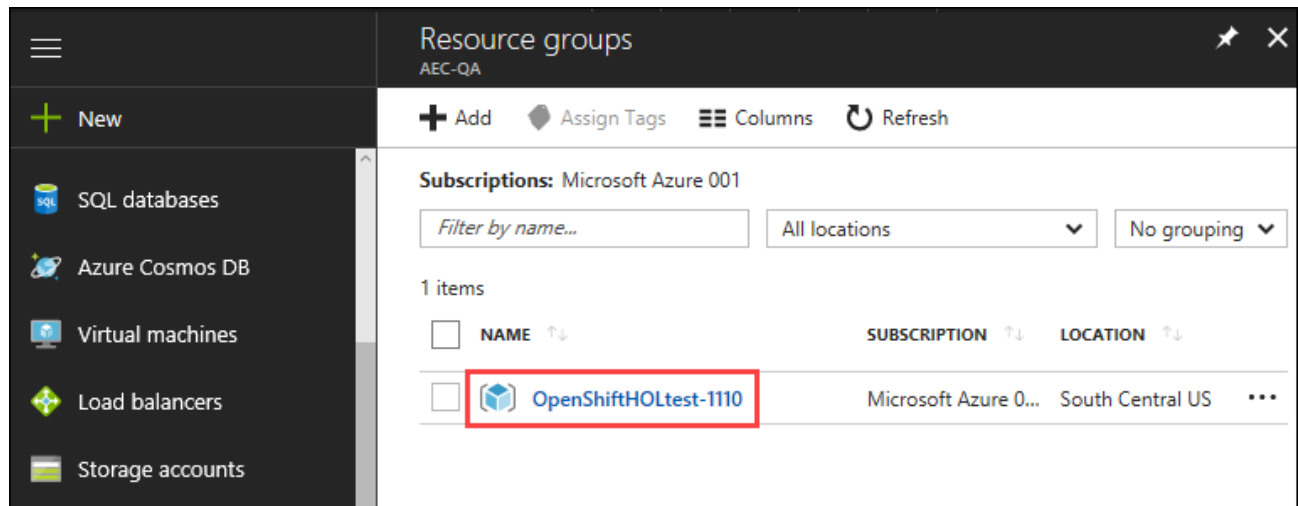
3. To toggle **show/hide** the Portal menu options with icon, **Click** on the **Show Menu** button.



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



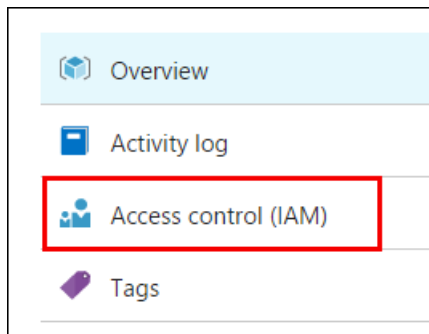
5. You will see a Resource Group which you have access to, **click** on it.



Note:

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

- From the Resource Group blade that come up, **Select** the Access Control (IAM) which is on the left side of the blade.



- In the new blade that come up, you can see the **role** that is assigned to you.

Roles
Refresh
Help


Name
Type
Role

All
2 selected

Scope
Group by

All scopes
Role

5 items (3 Users, 1 Groups, 1 Service Principals)

<input type="checkbox"/>	NAME	TYPE	ROLE	SCOPE
CONTRIBUTOR				
	HOL_User 1110 hol_user_1110@aecqa.o...	User	Contributor	This resource

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

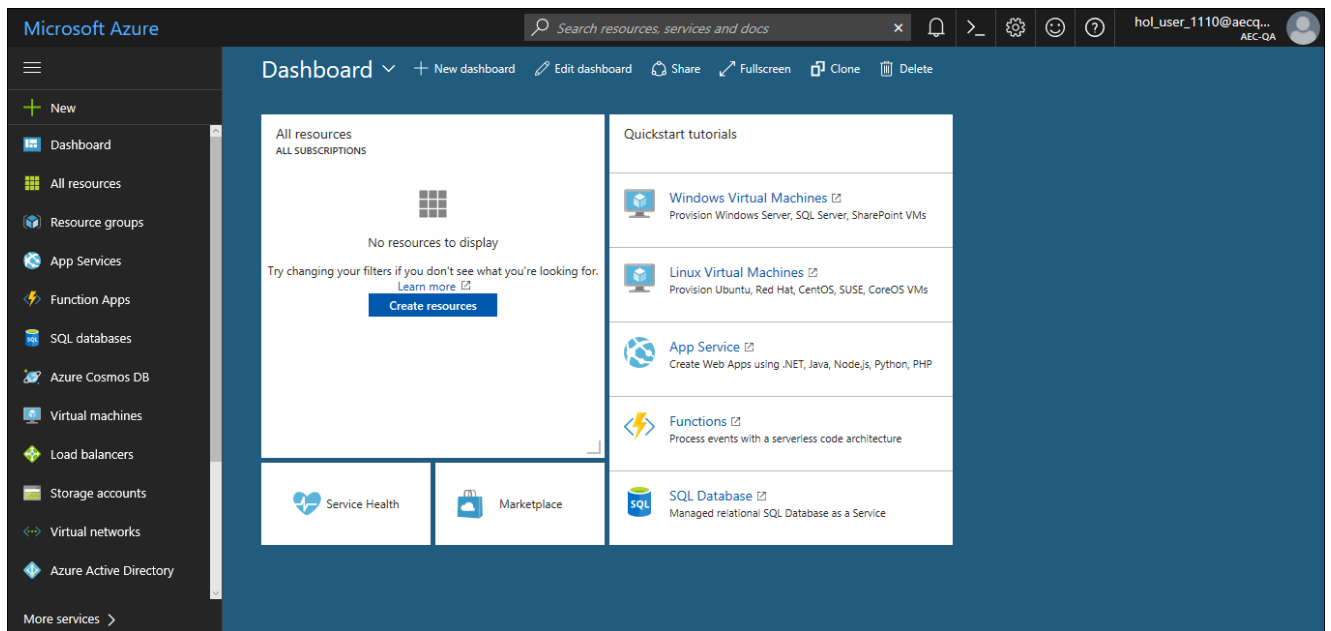
Time Estimate

120 minutes

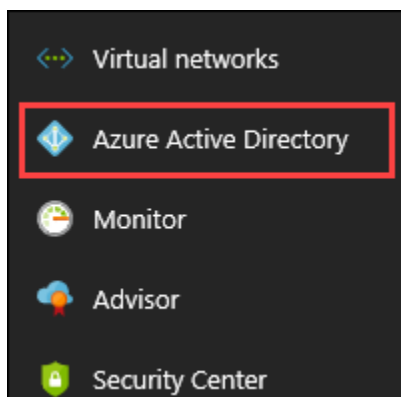
Exercise 1: Create an Azure AD Application

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

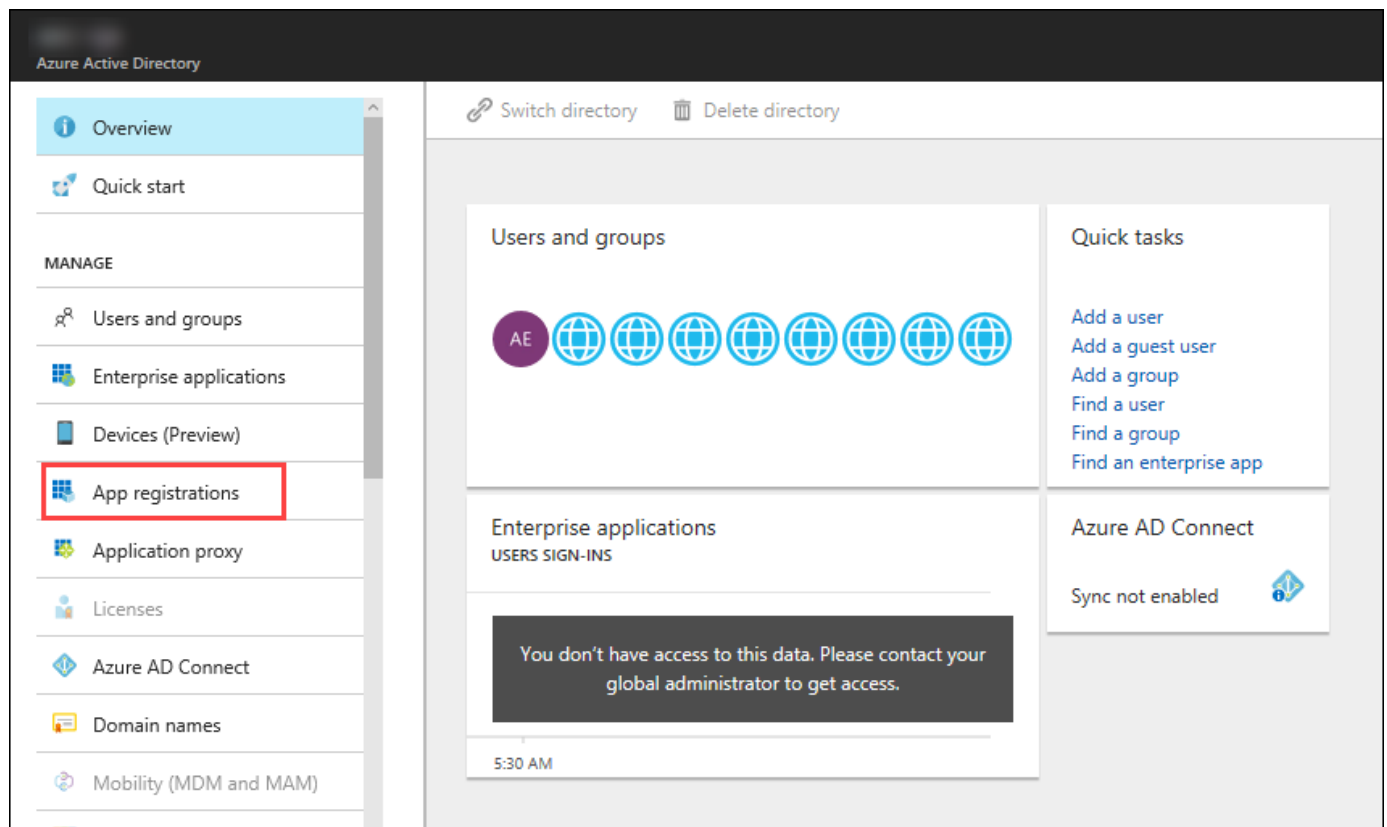
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.



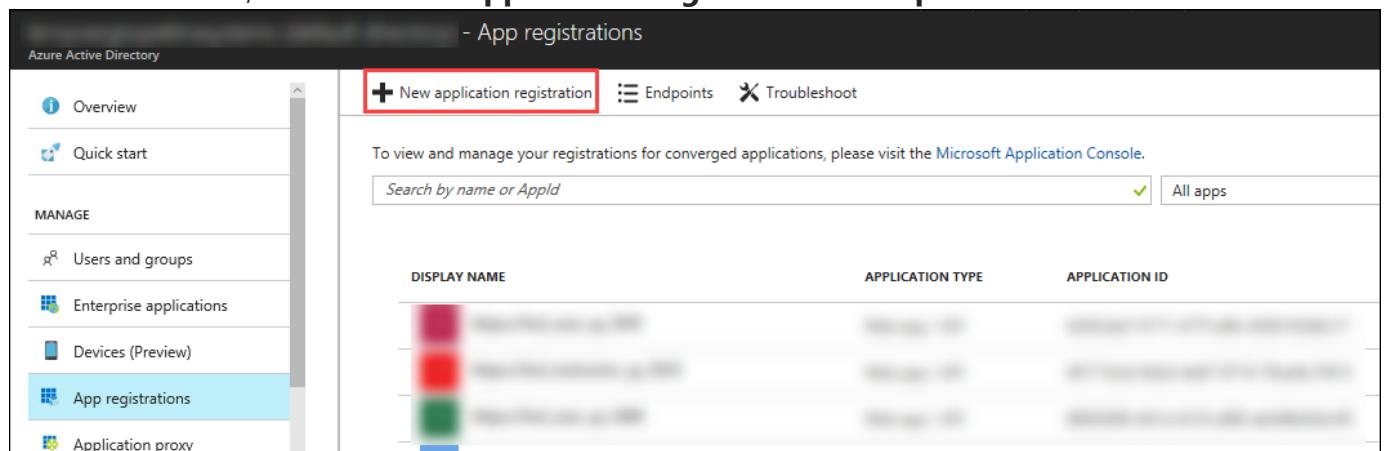
2. Click on the **Azure Active Directory** button in the **Menu navigation** bar to view the **Azure Active Directory** blade.



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



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



5. 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.**

Create

* Name ⓘ

Application type ⓘ

Web app / API

* Sign-on URL ⓘ

https://contoso.com

Create

- 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.

Azure Active Directory - App registrations

+ New application registration | Endpoints | Troubleshoot

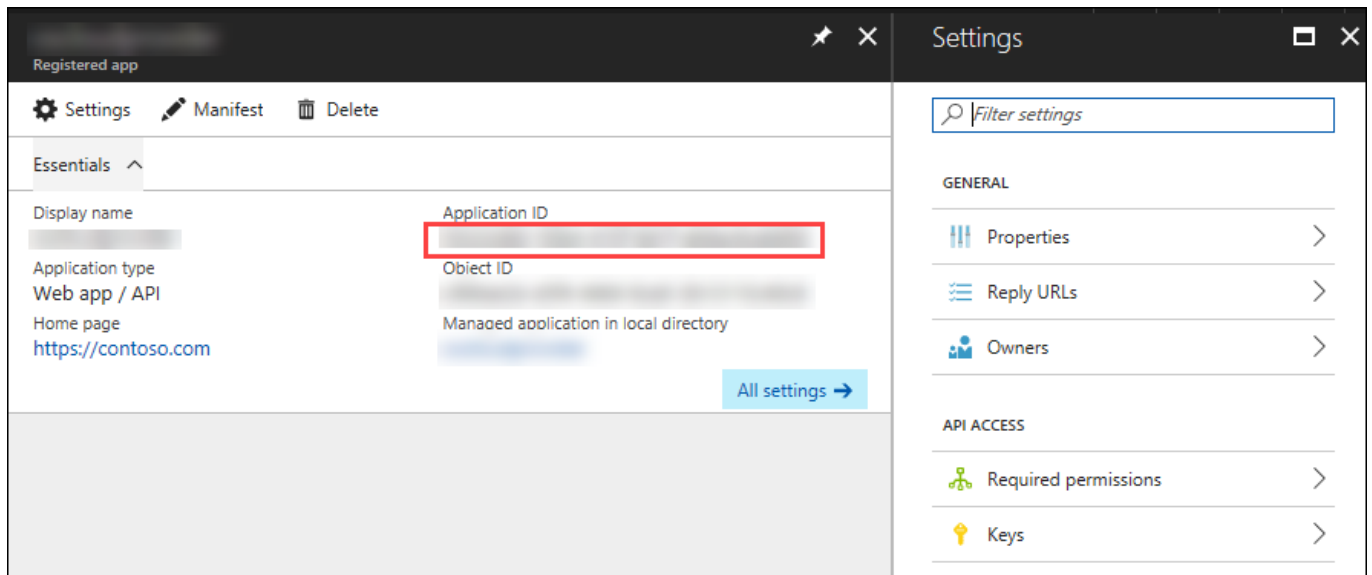
To view and manage your registrations for converged applications, please visit the [Microsoft Application Console](#).

Search: [App Name] ✓ All apps

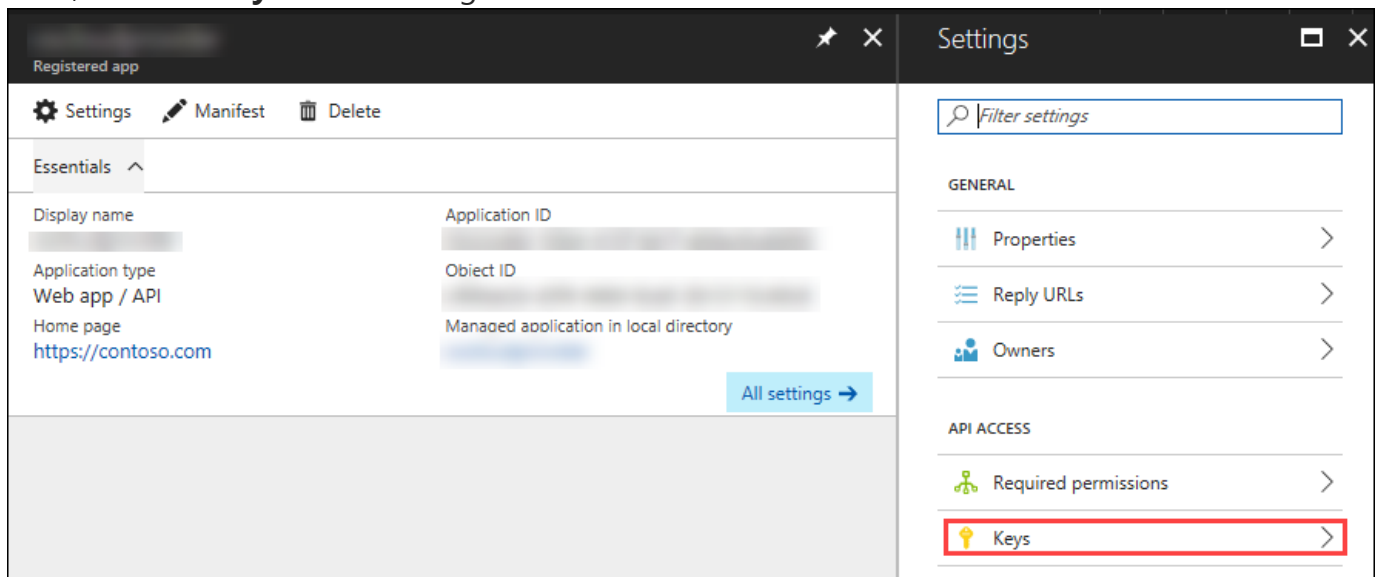
DISPLAY NAME	APPLICATION TYPE	APPLICATION ID
OS [App Name]	Web app / API	[Application ID]

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

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



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



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

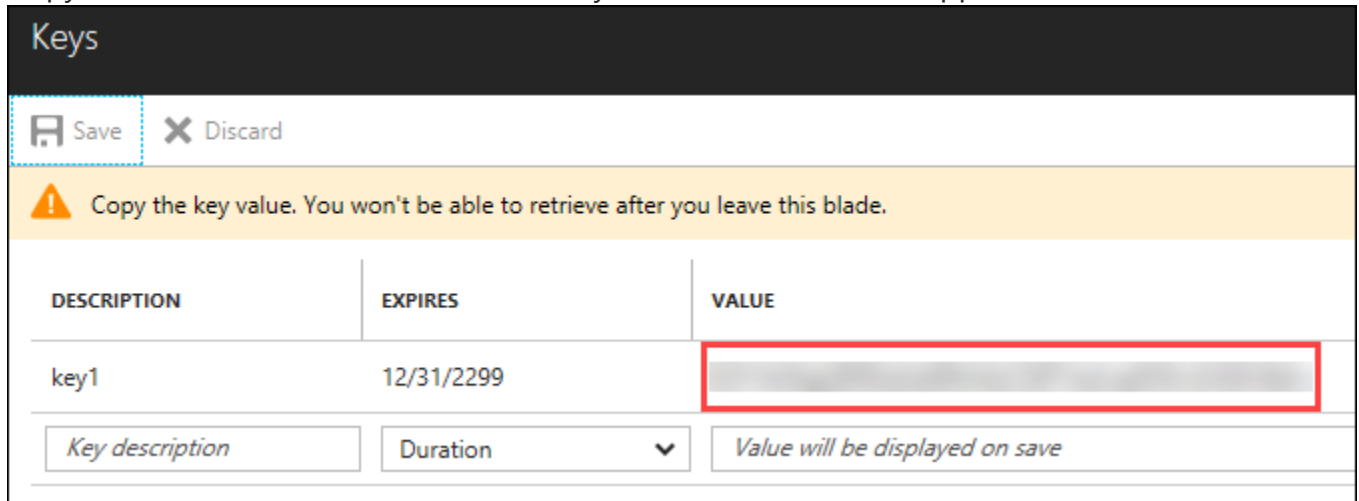
- Description: **key1**
- Expires: **Never expires**

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

The screenshot shows the 'Keys' blade in the Azure portal. The 'Save' button is highlighted with a red box. The 'DESCRIPTION' field contains 'key1', the 'EXPIRES' dropdown is set to 'Never expires', and the 'VALUE' field is empty.

DESCRIPTION	EXPIRES	VALUE
key1	Never expires	Value will be displayed on save

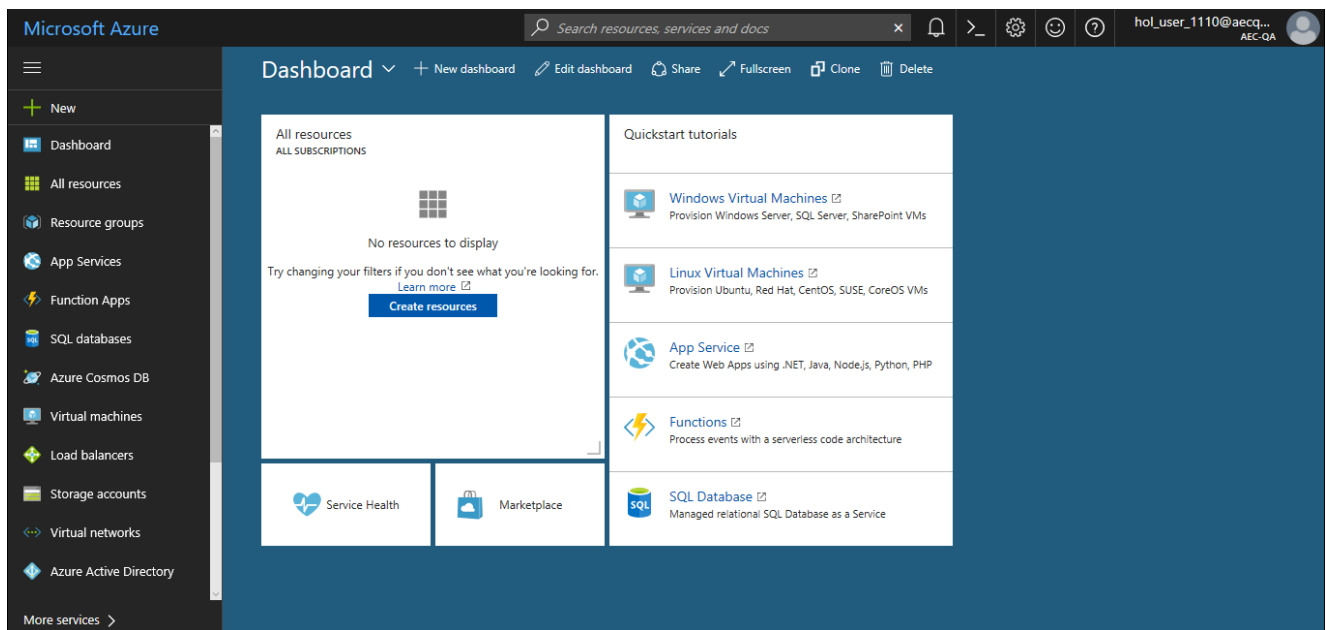
11. After you click on save, the key value will be displayed which is the Client Secret. Copy the value into the text editor where you saved the value of Application Id for later use.



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.

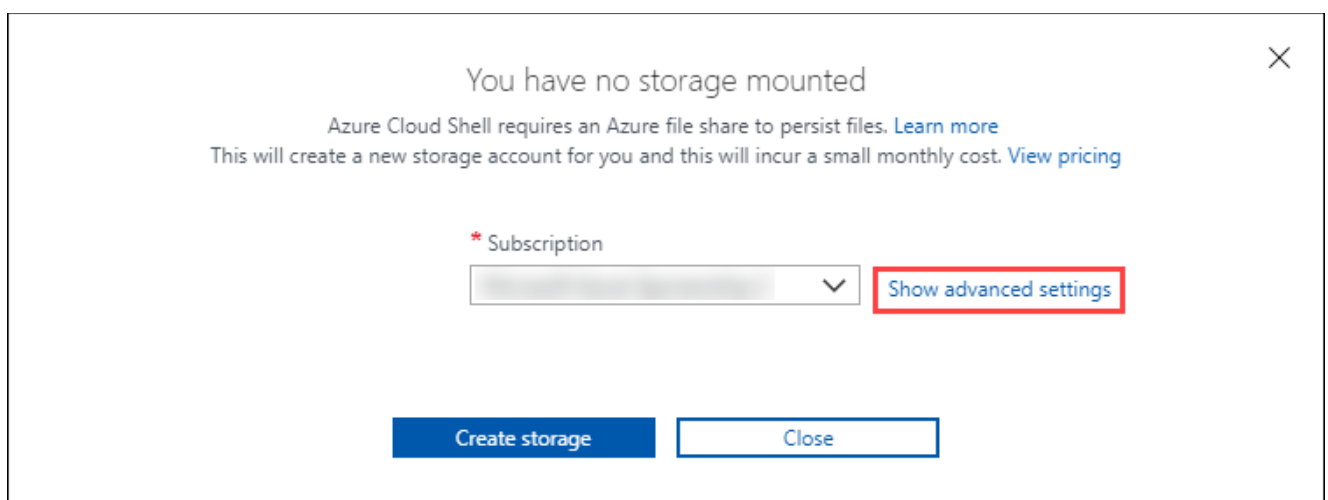
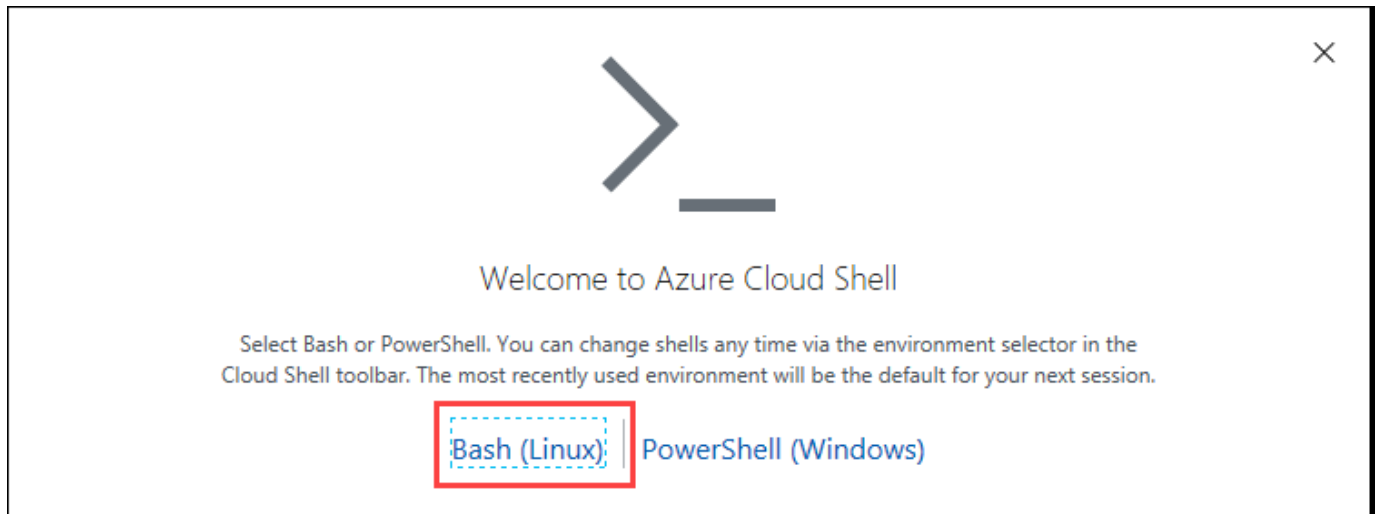
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.



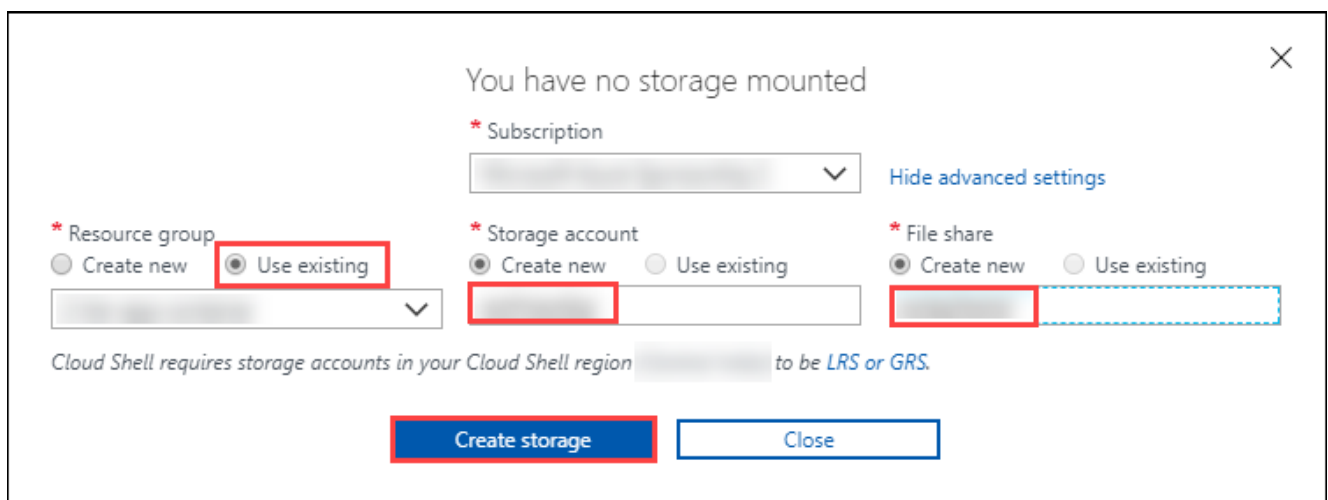
2. **Click** on **Cloud Shell** at the top right corner of the screen, to open the cloud shell.



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



4. 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**.



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

```
Bash
Your cloud drive has been created in:

Subscription Id: 
Resource group: OpenShiftHOLtest-1110
Storage account: osdemossd1
File share:      osdemossd2

Initializing your account for Cloud Shell...-
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell (Preview)

Type "az" to use Azure CLI 2.0
Type "help" to learn about Cloud Shell

hol_user@Azure:~$
```

6. Now execute the following command in the cloud shell to create a key vault in the existing resource group.

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

Note: Provide the existing Resource Group name, it's location and a unique name for key vault in the above command when executing

```
hol_user@Azure:~$ az keyvault create -n keyvaulthol -g OpenShiftHOLtest-1112 -l southcentralus --enabled-for-template-deployment true
{
  "id": "/subscriptions/.../resourceGroups/OpenShiftHOLtest-1112/providers/Microsoft.KeyVault/vaults/keyvaulthol",
  "location": "southcentralus",
  "name": "keyvaulthol",
  "properties": {
    "accessPolicies": [
      {

```

7. 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.

```
hol_user@Azure:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hol_user/.ssh/id_rsa):
Created directory '/home/hol_user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hol_user/.ssh/id_rsa.
Your public key has been saved in /home/hol_user/.ssh/id_rsa.pub.
The key fingerprint is:
```

8. 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: The copied SSH Key should be made into a single line. You will need this key for later use.

```
hol_user@Azure:~$ cat .ssh/id_rsa.pub
ssh-rsa
hol_user@Azure:~$
```

9. 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
```

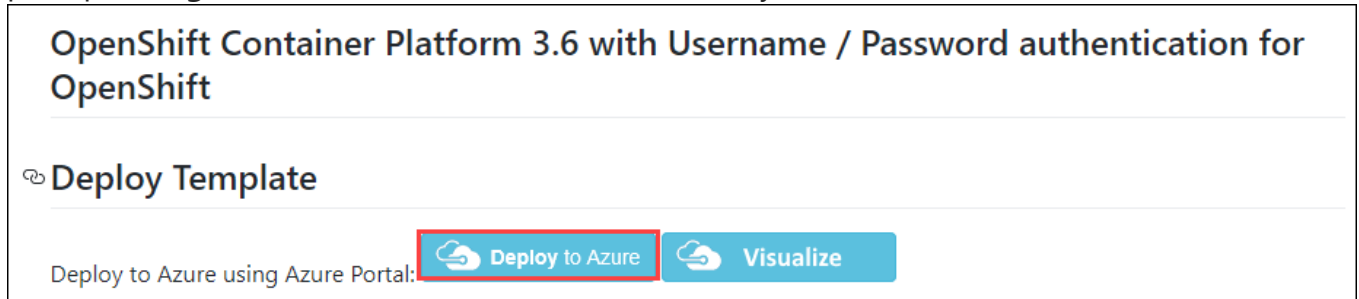
Note: Substitute for key vault name in the above command with the name of the keyvault created earlier when executing.

```
hol_user@Azure:~$ az keyvault secret set --vault-name keyvaulthol -n osdemovaultsecret --file ~/.ssh/id_rsa
{
  "attributes": {
    "created": "2017-10-26T09:38:41+00:00",
    "enabled": true,
    "expires": null,
    "notBefore": null,
    "recoveryLevel": "Purgeable",
    "updated": "2017-10-26T09:38:41+00:00"
  },
```


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/SpektraSystems/openshift-container-platform>
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.




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

Custom deployment

Deploy from a custom template

TEMPLATE



Customized template
36 resources

[Edit template](#)
[Edit parameters](#)
[Learn more](#)

BASICS

Subscription

Resource group

☒ Create new
☐ Use existing

Location

West US

SETTINGS

_artifacts Location ⓘ

Admin Username ⓘ

ocpadmin

Openshift Password ⓘ

Rhsm Username Password Or Activation Key ⓘ

usernamepassword

☐ Pin to dashboard

Purchase

4. 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**
- Rhsm Username or Org Id: **Provide the username of Redhat credentials**
- Rhsm Password or Org Id: **Provide the password of Redhat credentials**
- Rhsm Pool Id: **Provide the pool Id of Redhat OpenShift Subscription**
- Key Vault Name : **Provide the Key Vault name you provided**
- Key Vault Secret : **osdemovaultsecret**
- Aad Auth App Name : **Provide the name of the AD App you created**
- Aad Auth App Id : **Provide the Client ID of the AD App you created**


- Aad Auth Client Secret : **Provide the secret key of the AD App**


And accept the terms of conditions.


Custom deployment


Deploy from a custom template

TEMPLATE

 Customized template
36 resources

 Edit template

 Edit parameters

 Learn more

BASICS

* Subscription

* Resource group

☐ Create new

☒ Use existing

* Location

West US

SETTINGS

_artifacts Location ⓘ

Admin Username ⓘ

ocpadmin

* Openshift Password ⓘ

Rhsm Username Password Or Activation Key ⓘ

usernamepassword

Custom deployment
Deploy from a custom template

Rhsm Username Password Or Activation Key ①

* Rhsm Username Or Org Id ①

* Rhsm Password Or Activation Key ①

* Rhsm Pool Id ①

* Ssh Public Key ①

* Key Vault Name ①

* Key Vault Secret ①

* Aad Auth App Name ①

* Aad Auth Client Id ①

* Aad Auth Client Secret ①

TERMS AND CONDITIONS

[Azure Marketplace Terms](#) | [Azure Marketplace](#)

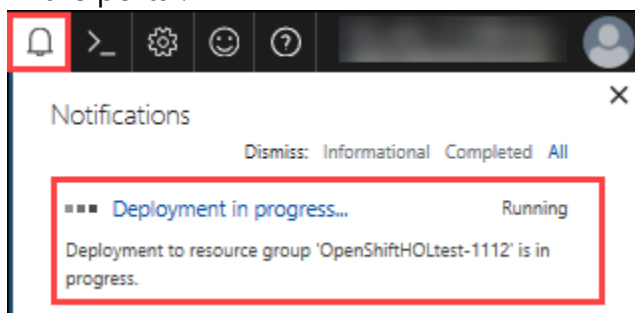
By clicking "Purchase," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

☒ I agree to the terms and conditions stated above

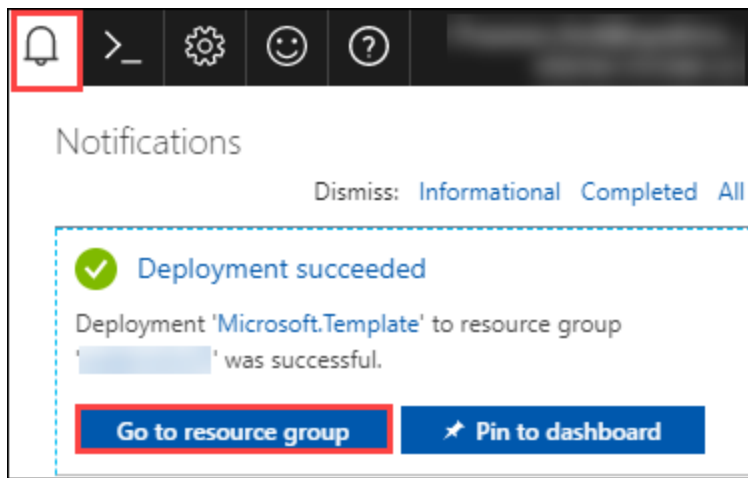
☐ Pin to dashboard

Purchase

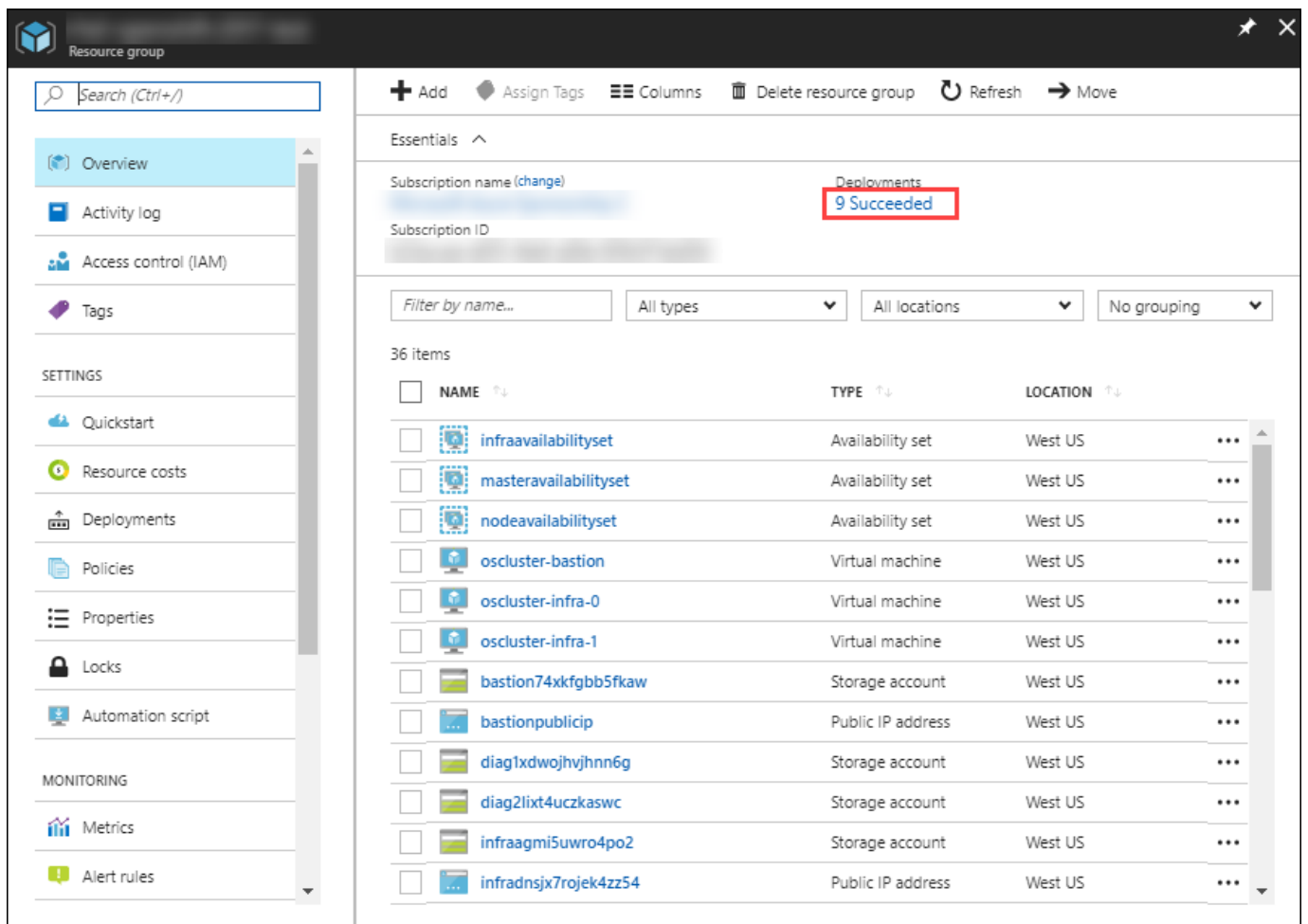
- And then **click** on **Purchase**.
- Once the deployment starts, you can see the progress in the notification bar at the top of the Azure portal.



- 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.



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



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

DEPLOYMENT NAME ↑↓	STATUS
Microsoft.Template	✓ Succeeded
OpenShiftDeployment	✓ Succeeded
masterVmDeployment0	✓ Succeeded
nodeVmDeployment0	✓ Succeeded
infraVmDeployment0	✓ Succeeded

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

Microsoft.Template
Deployment

Delete
Cancel
Refresh
Redeploy
View template

Summary

DEPLOYMENT DATE01/11/2017, 17:51:47
STATUSSucceeded
DURATION28 minutes 1 second
RESOURCE GROUP
RELATEDEvents

Outputs

OPENSIFT CONSOLE URL
https://masterdnsib2j2coakdzf6.westus.cloudapp.az...

BASTION DNS FQDN
bastiondns6gl4wxa3jame6.westus.cloudapp.azure.c...

OPENSIFT MASTER SSH
ssh ocpadmin@masterdnsib2j2coakdzf6.westus.clo...

OPENSIFT INFRA LOAD BALANCER FQDN
infradnsjx7rojek4zz54.westus.cloudapp.azure.com

NODE OS STORAGE ACCOUNT NAME
nodeosmr5s4pogr42hk

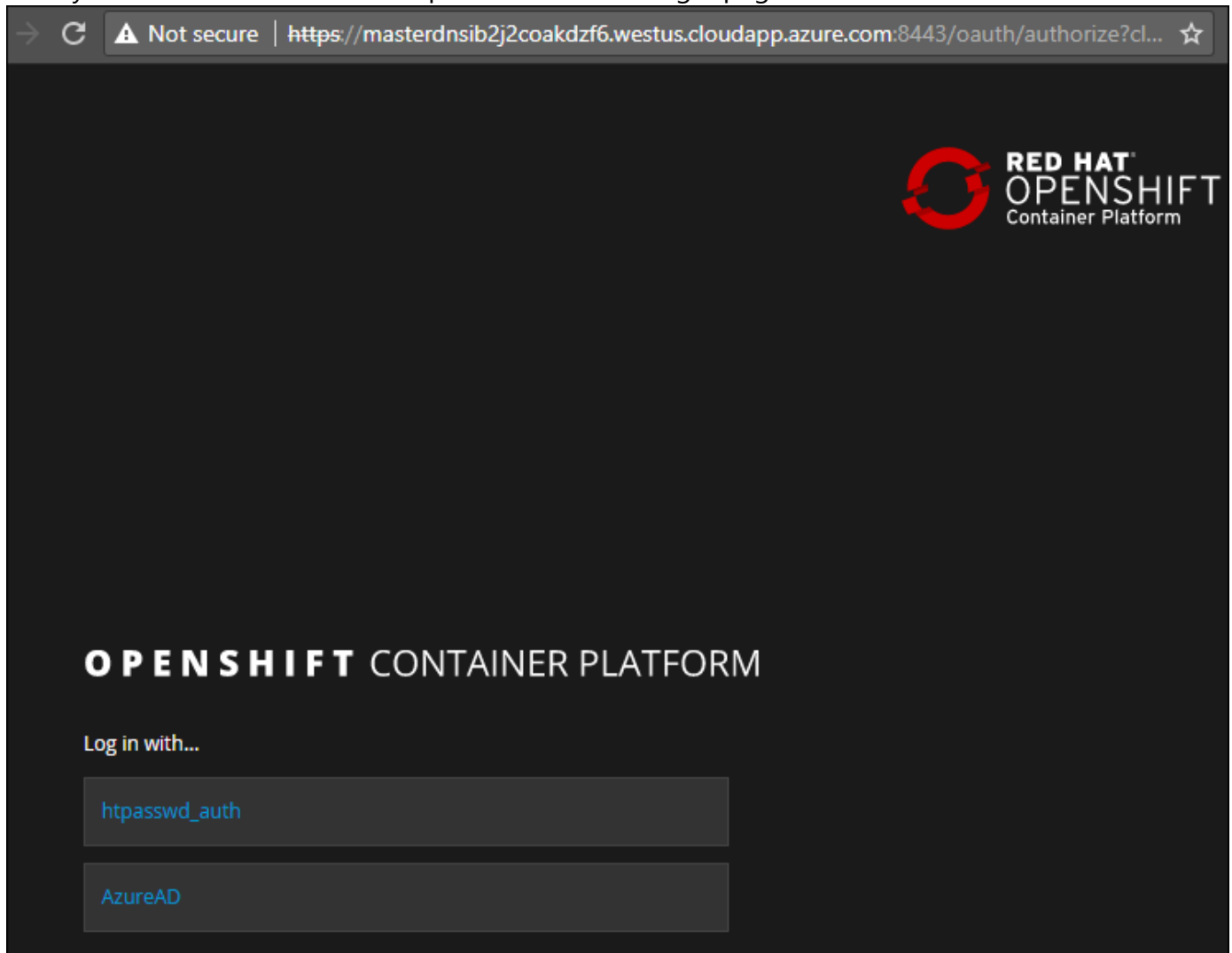
NODE DATA STORAGE ACCOUNT NAME
nodedataflo3vjkp4ou

11. Copy the Openshift Console URL, Bastion DNS FQDN and OpenShift Master SSH by clicking on Copy to a text editor

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

Note: Skip the certificate warning

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

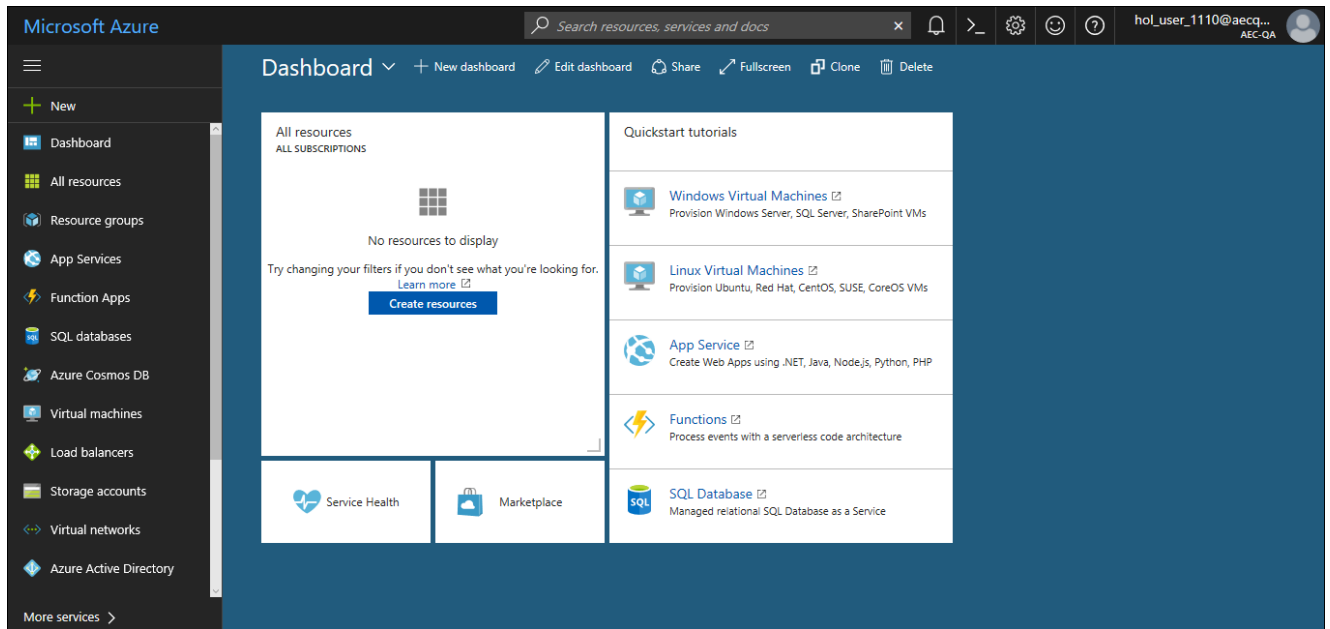


Note: If the above page comes up, then the deployment is working.

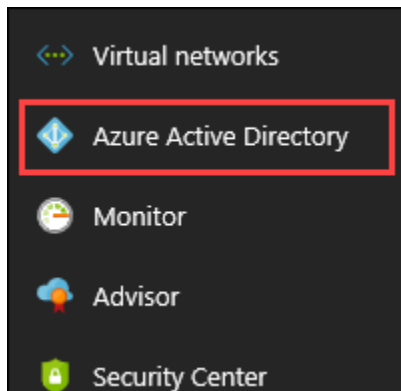
Exercise 4: Configure Azure AD Authentication

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

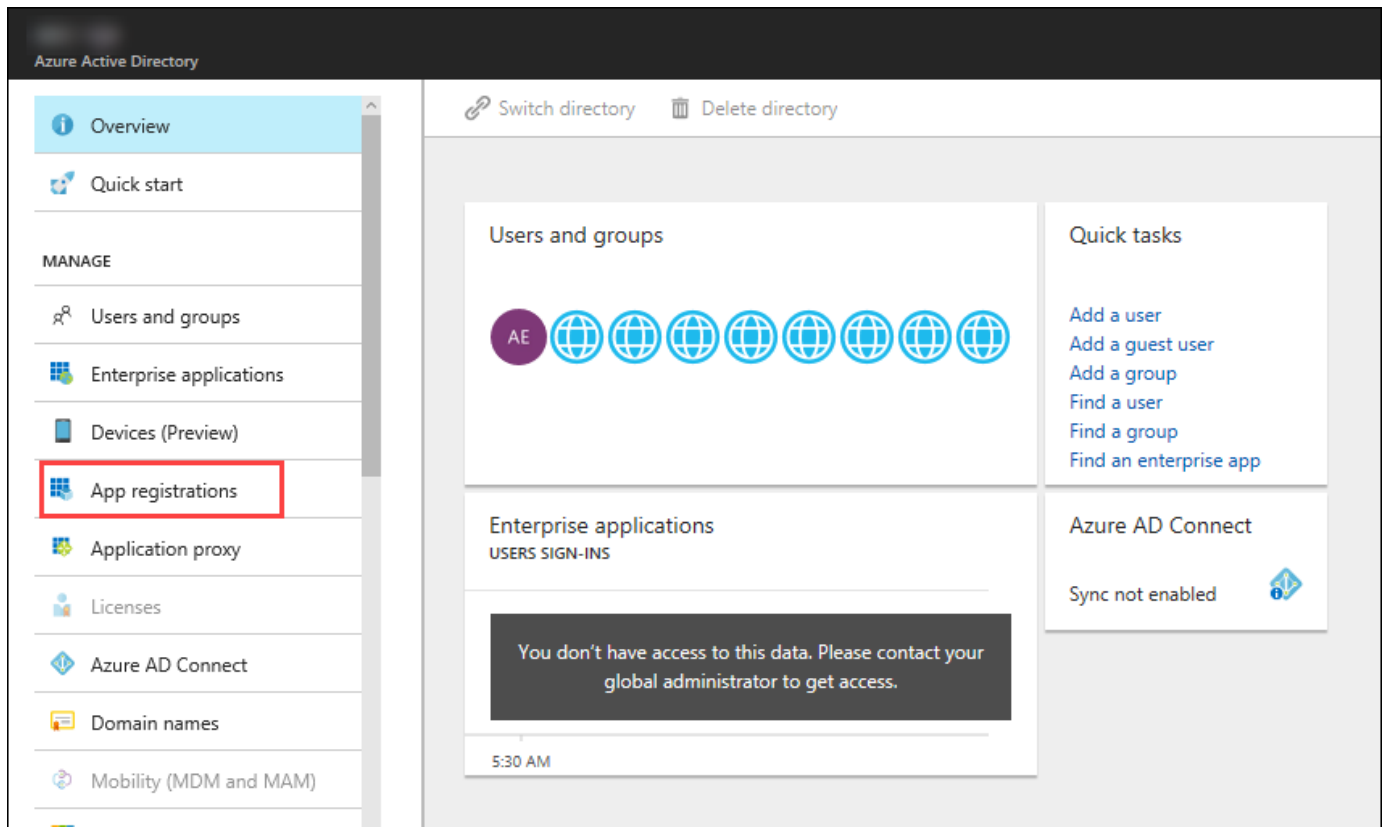
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.



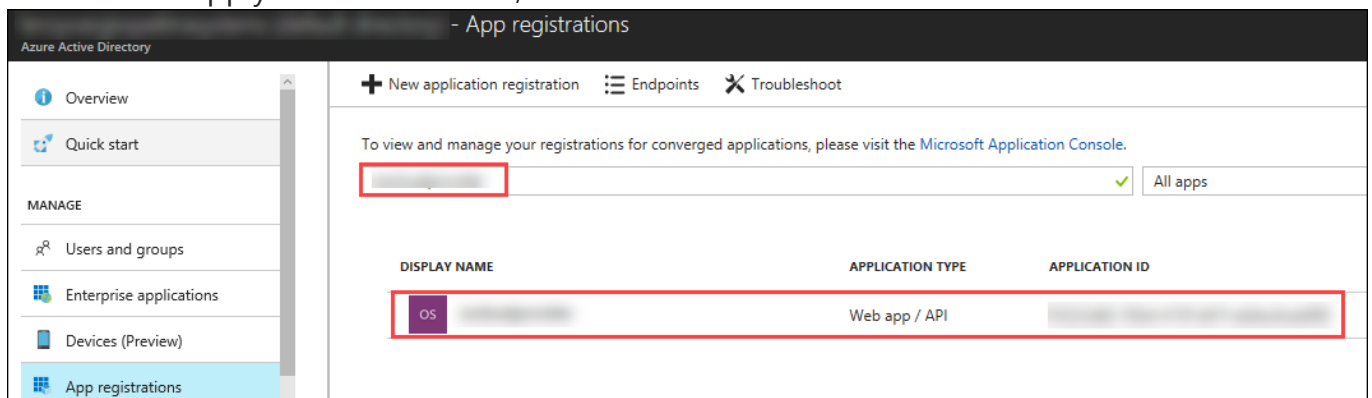
2. Click on the **Azure Active Directory** button in the **Menu navigation** bar to view the **Azure Active Directory** blade.



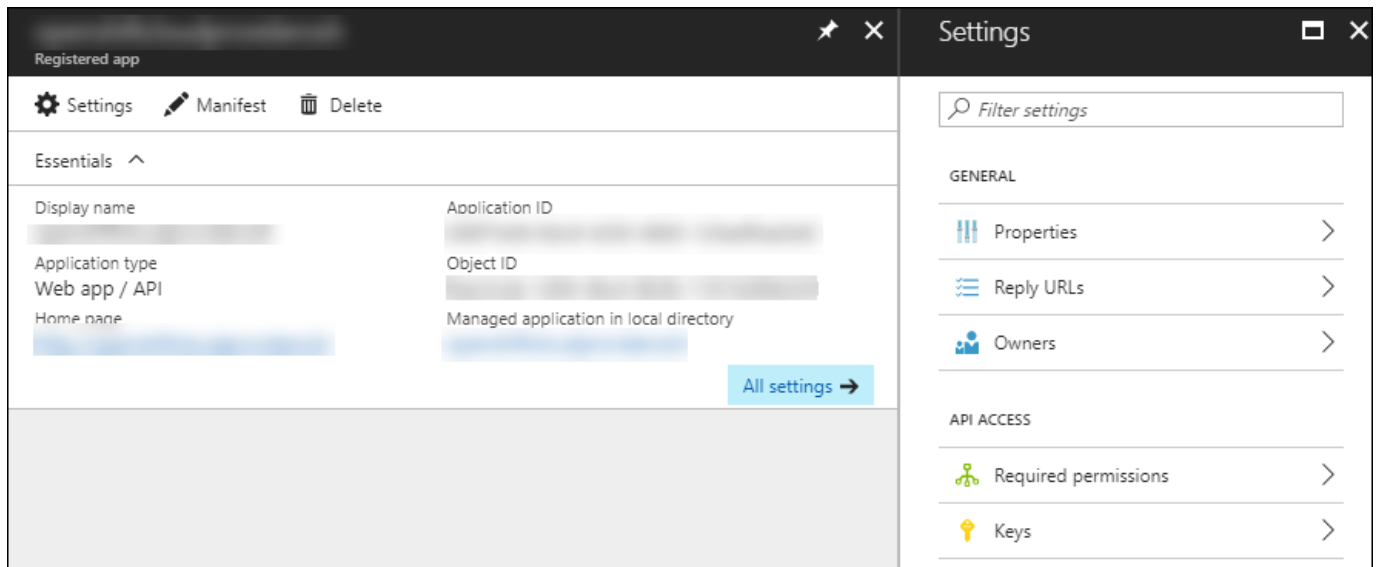
3. You will be directed to the Azure Active Directory blade, **click** on **App registrations**.



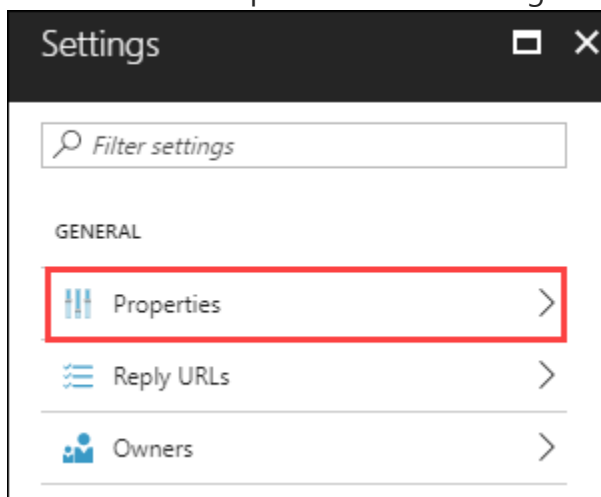
- You will be redirected to the **App registrations** blade. You can search the App by typing the name of the App you created earlier, in the search field.



- Click on the app you created and you will be directed to the App blade.



6. Now Click on Properties under Settings blade.



7. 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**.

Properties

Save Discard

* Name ⓘ

Object ID ⓘ

Application ID ⓘ

* App ID URI ⓘ

https://masterdnsgoh25vgak4sg6.sout... ✓

Logo

OP

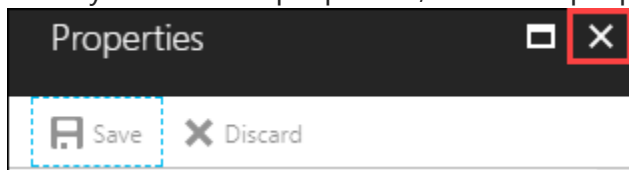
Upload new logo ⓘ

Select a file

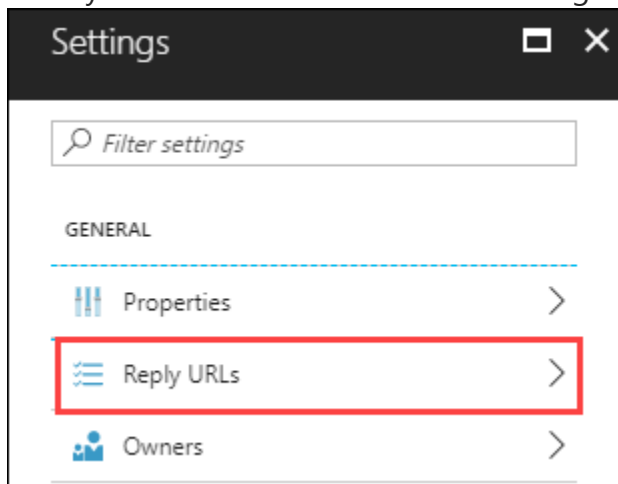
Home page URL ⓘ

https://masterdnsgoh25vgak4sg6.sout... ✓

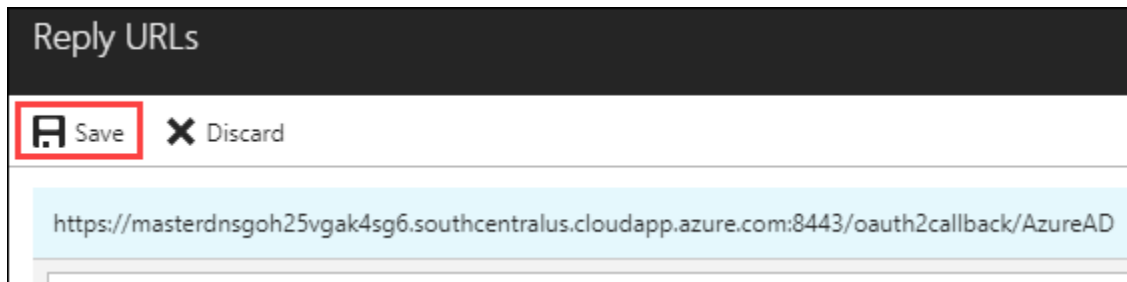
- Once you save the properties, close the properties blade.





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



10. 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.

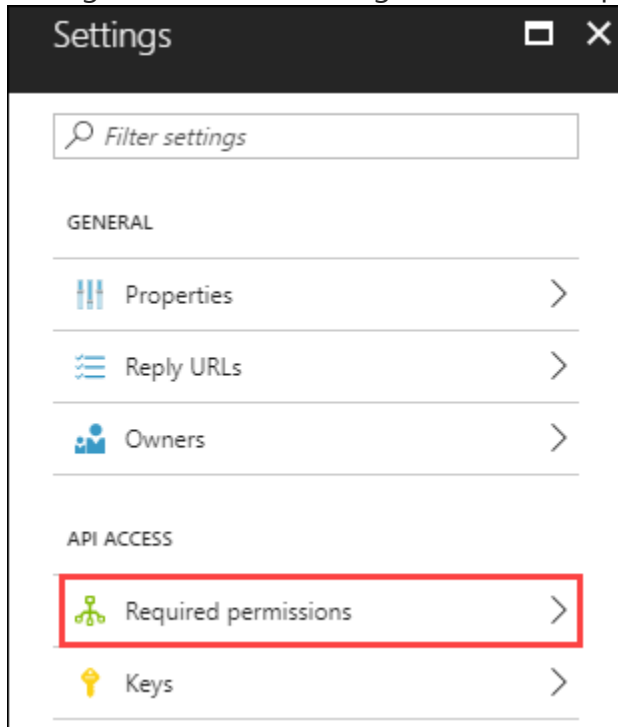


Reply URLs

 Save  Discard

`https://masterdmsgoh25vgak4sg6.southcentralus.cloudapp.azure.com:8443/oauth2callback/AzureAD`

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



Settings

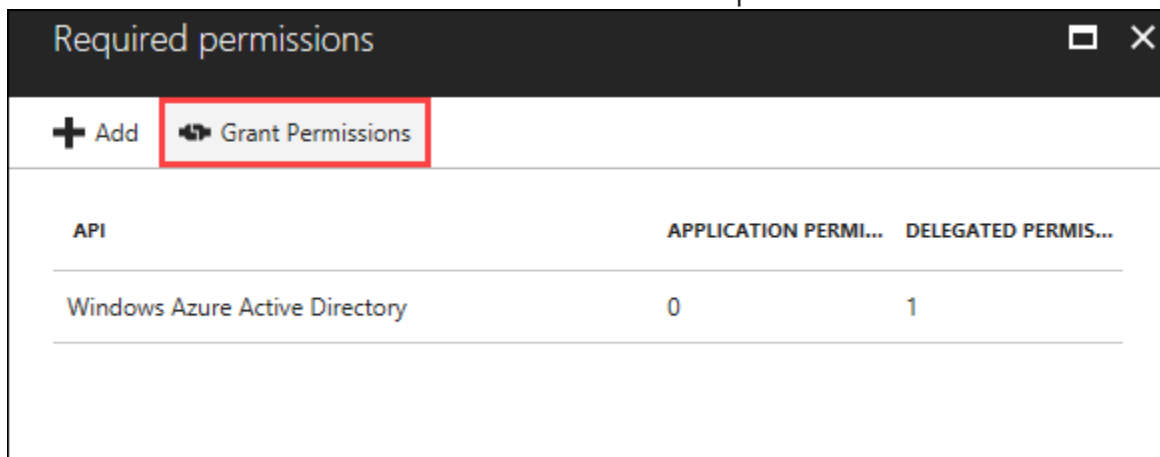
GENERAL

- Properties >
- Reply URLs >
- Owners >



API ACCESS

- Required permissions >**
- Keys >

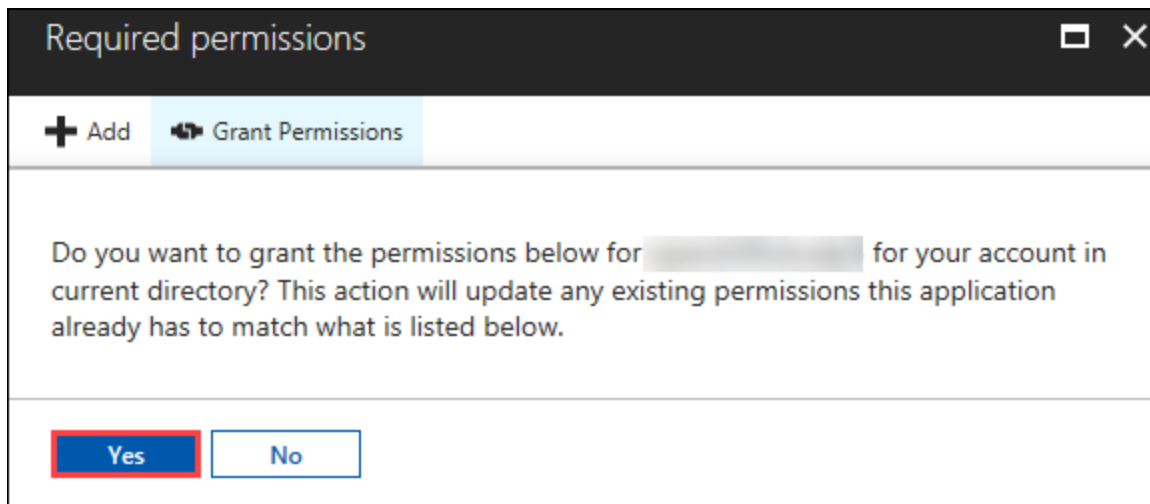
12. Click on Grant Permissions in the blade that come up and then **Click on Yes**



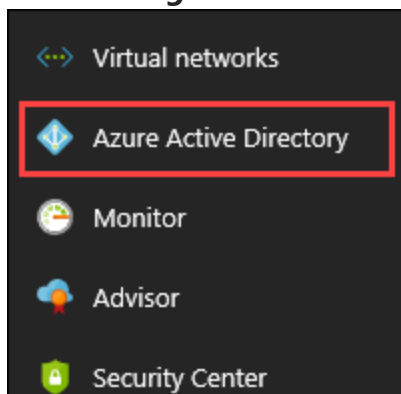
Required permissions

 Add  Grant Permissions

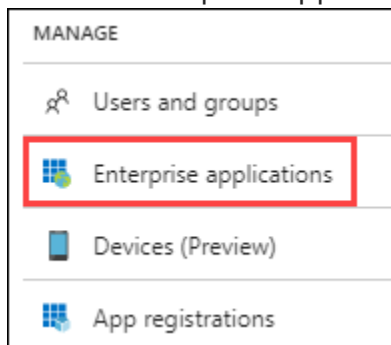
API	APPLICATION PERMI...	DELEGATED PERMIS...
Windows Azure Active Directory	0	1



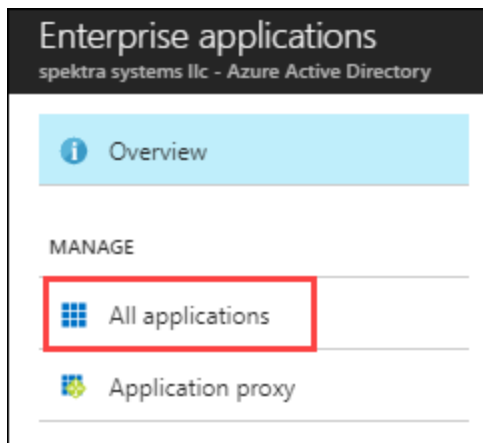
13. Now go back to the Active Directory blade by clicking on **Azure Active Directory** button in the **Menu navigation** bar.



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

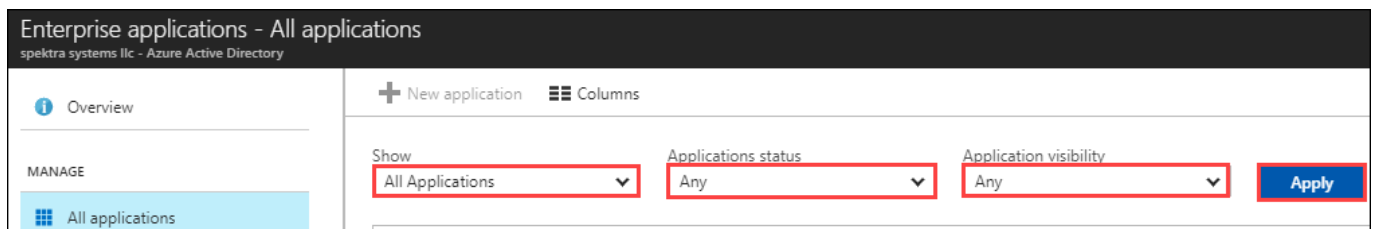


15. In the new blade that come up, click on **All applications**

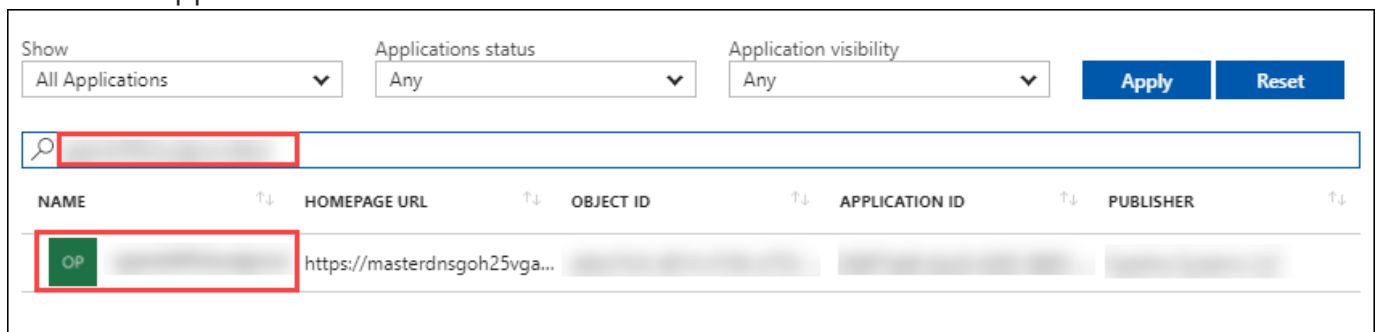


16. In the new blade that come up, edit the filter as follows:

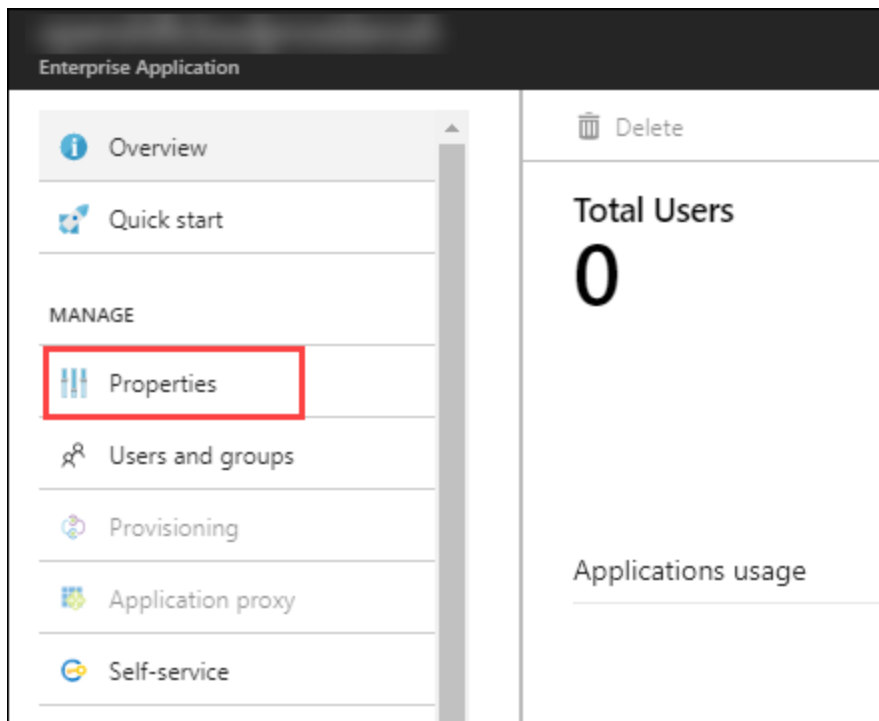
- Show: Select All Applications
 - Application status: Any
 - Application visibility: Any
- And then **click** on **Apply**.



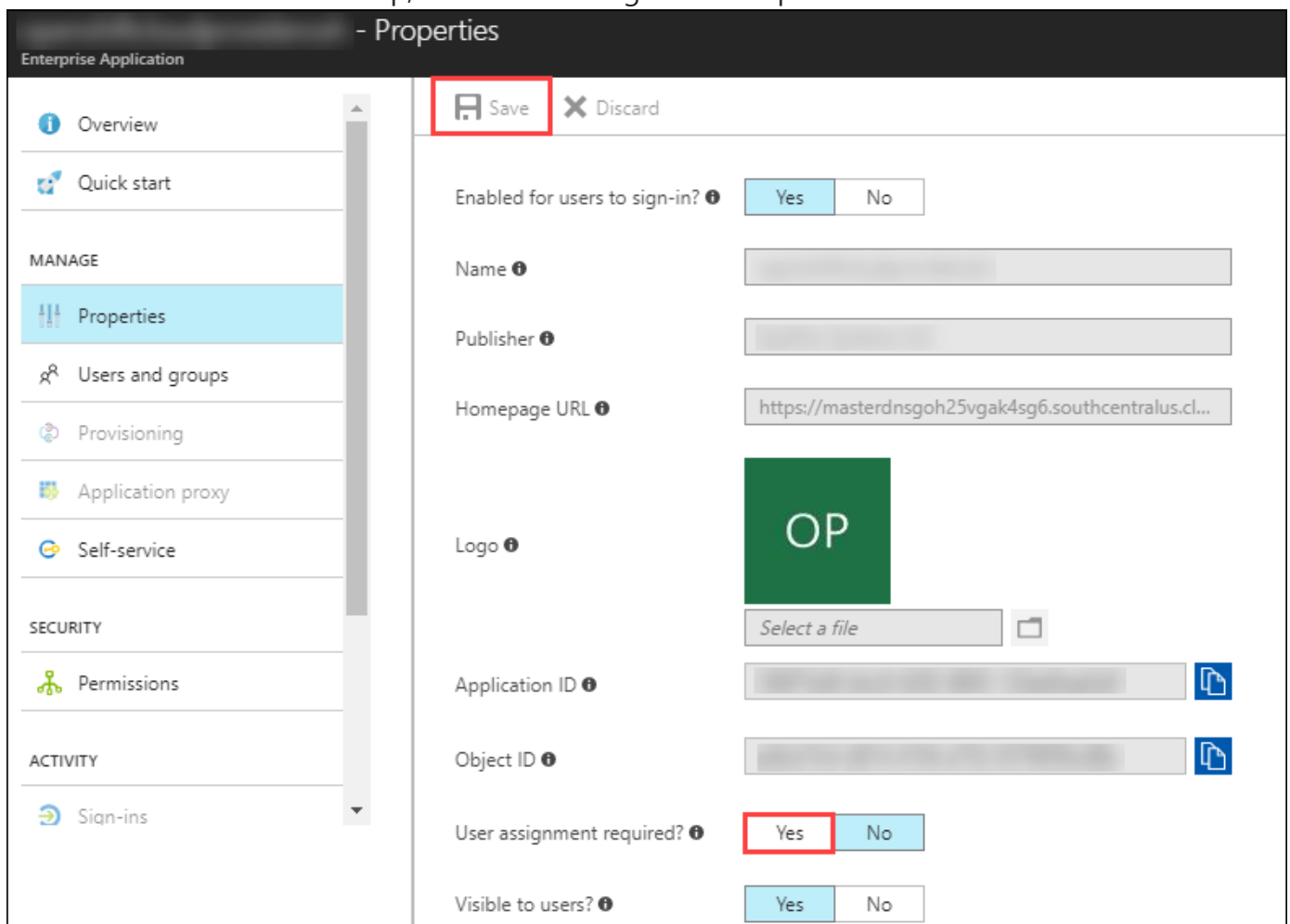
17. You can search the App by typing the name of the App you created earlier, in the search field. Select the App from the results.



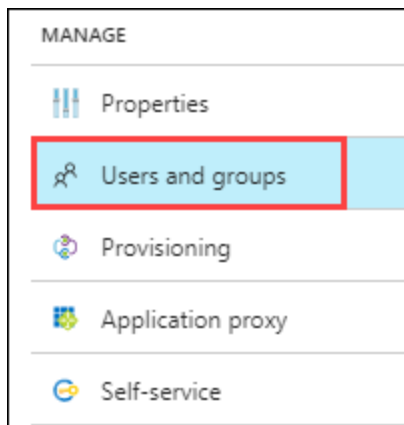
18. You will be redirected to the App Blade. Click on Properties under Manage section on the left side of the properties blade.



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

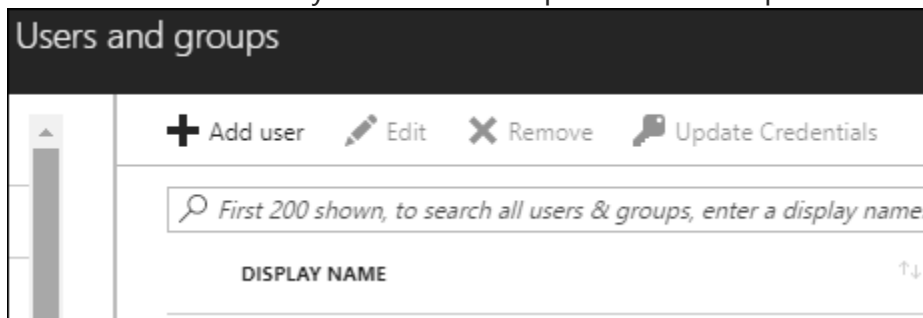


20. Now, click on **Users and groups** under Manage section on the left side of the App Blade.

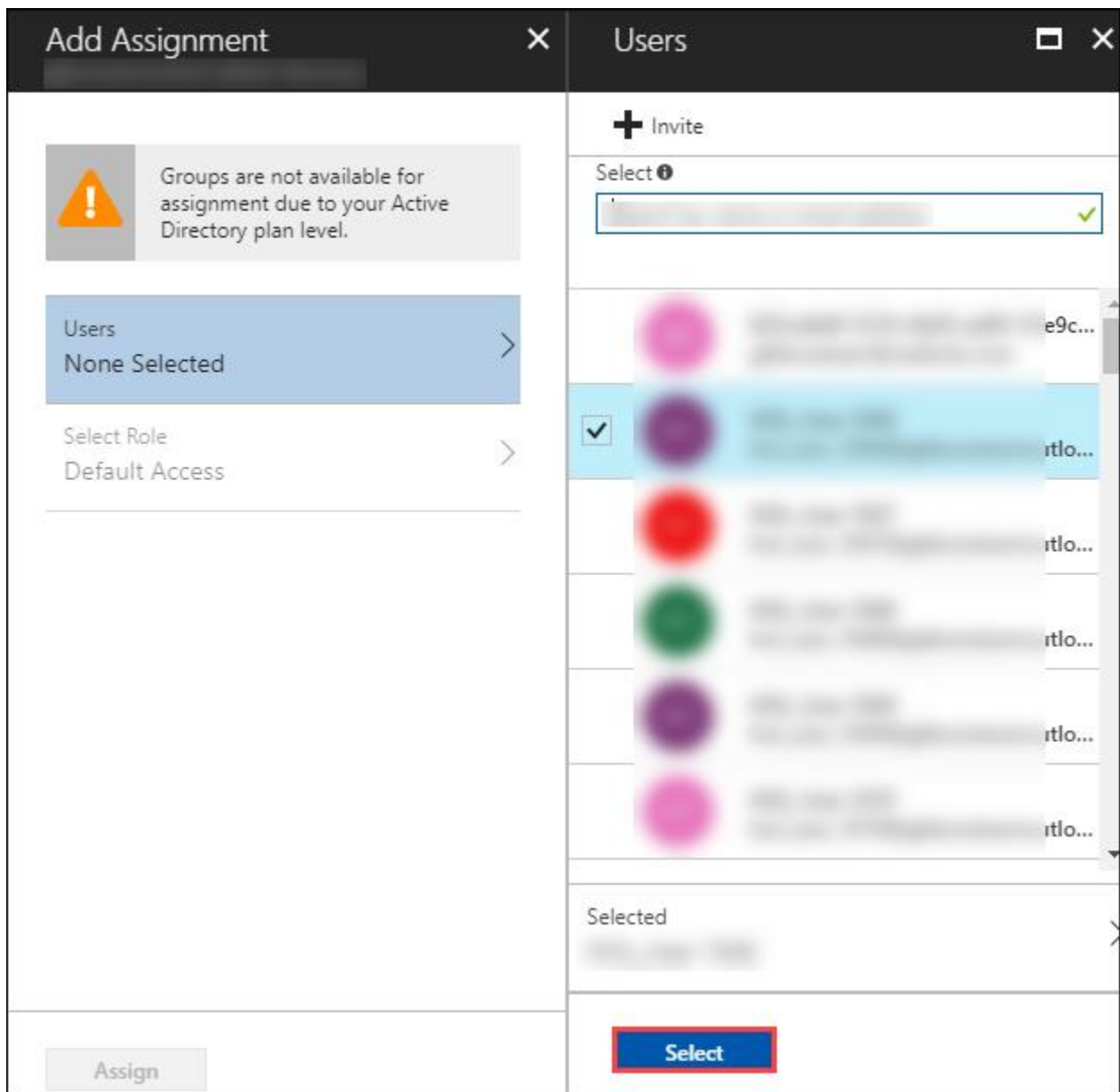


21. In the blade that come up, click on **+Add user** to assign a user to the app.

Note: If user is already added then skip next three steps.




22. 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**.



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

Add Assignment



Groups are not available for assignment due to your Active Directory plan level.

Users

1 user selected.

>

Select Role

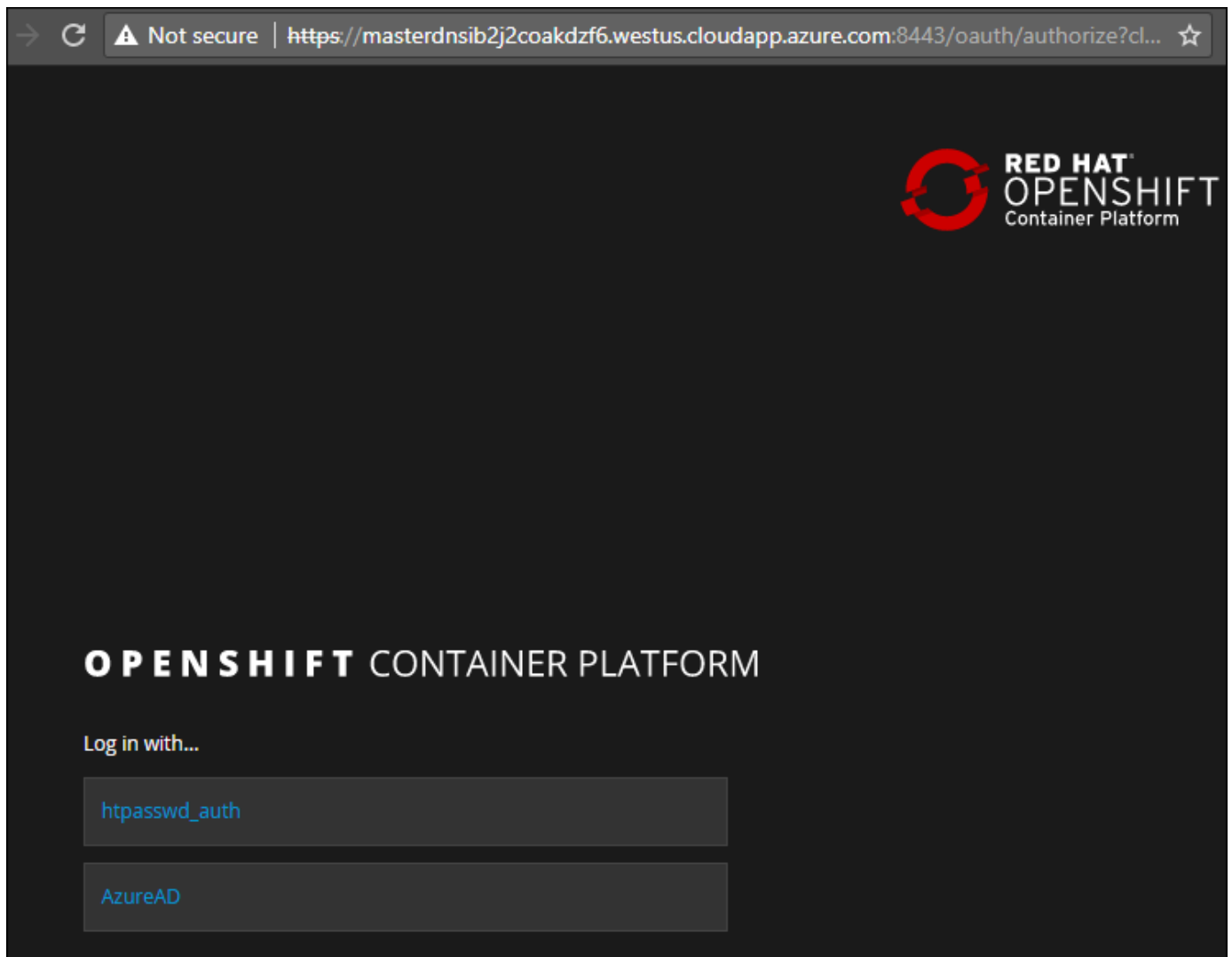
Default Access

>

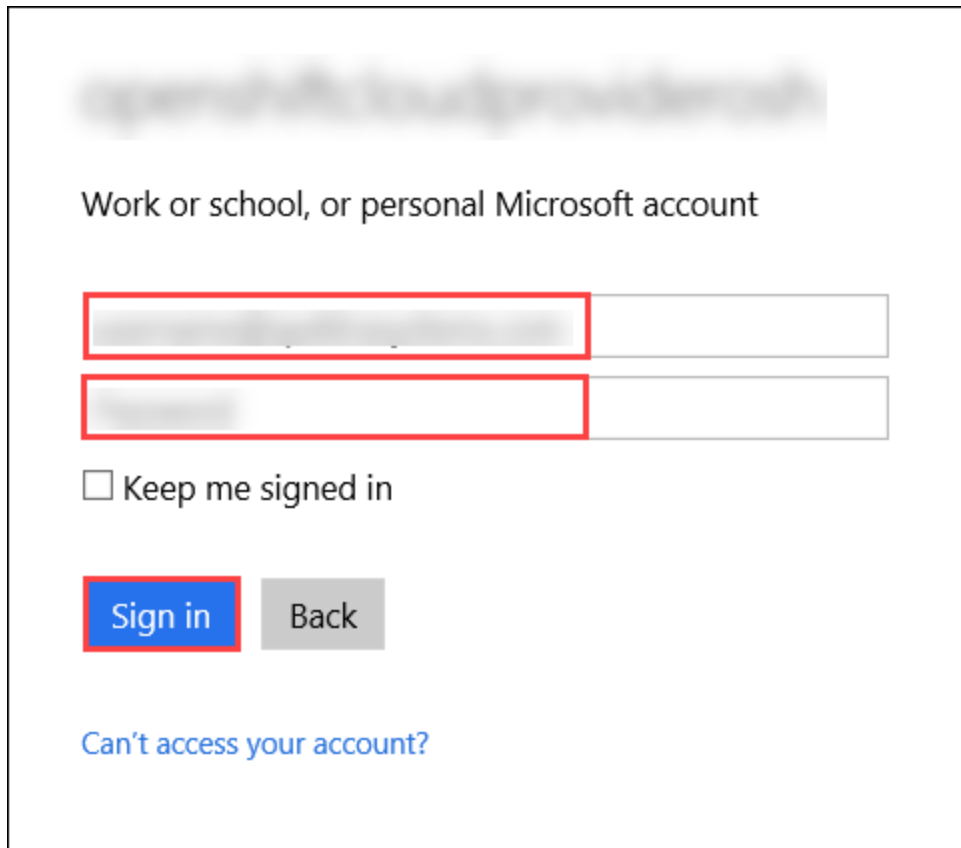
Assign

24. 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



25. 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.



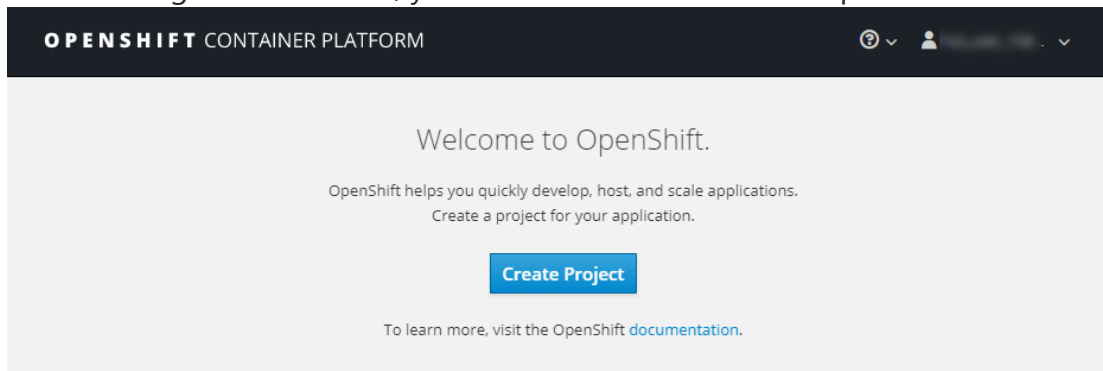
Work or school, or personal Microsoft account

☐ Keep me signed in

[Sign in](#) [Back](#)

[Can't access your account?](#)

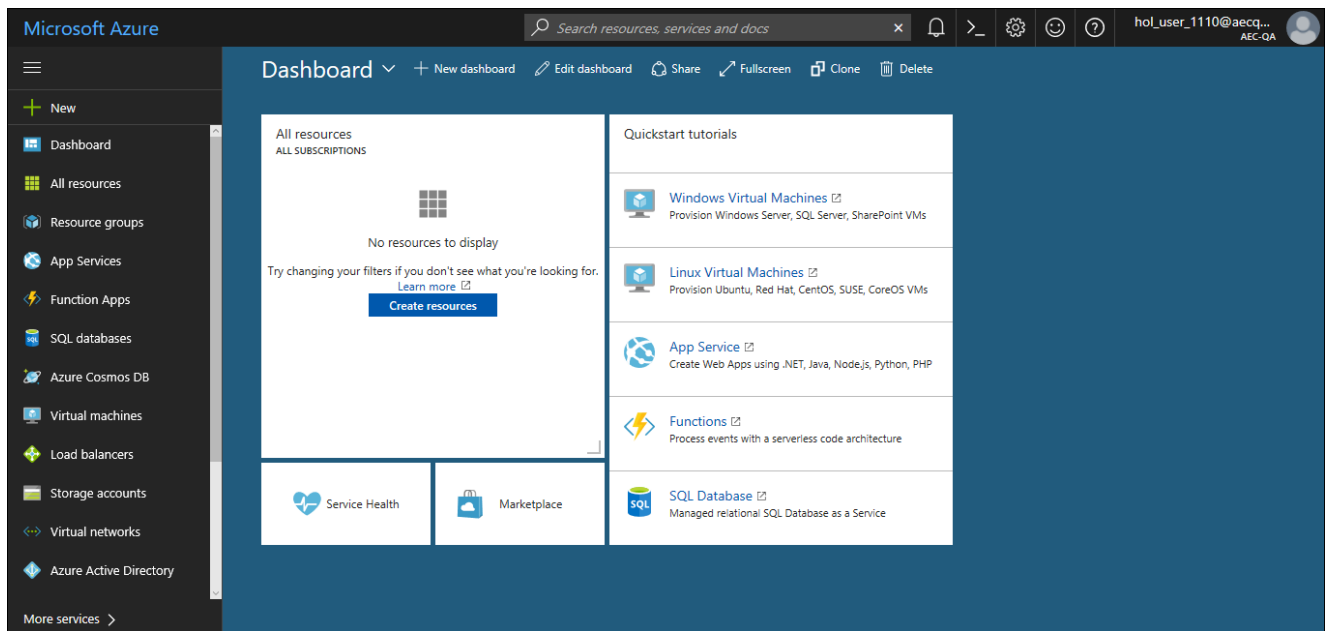
26. Once the login is successful, you will be redirected to the OpenShift console.



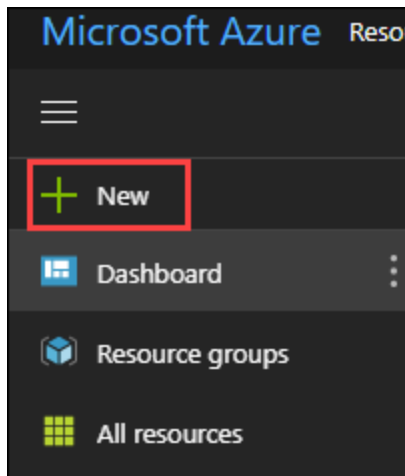
Exercise 5: Deploy a 2 Tier Node JS Application on Open Shift

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

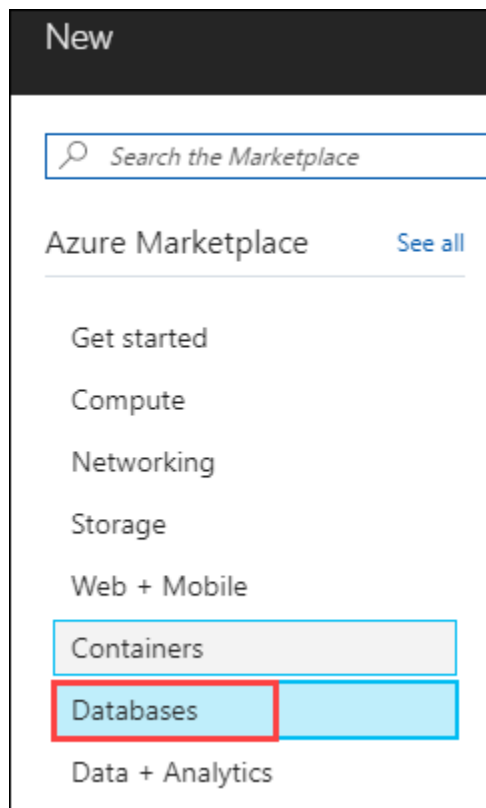
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.



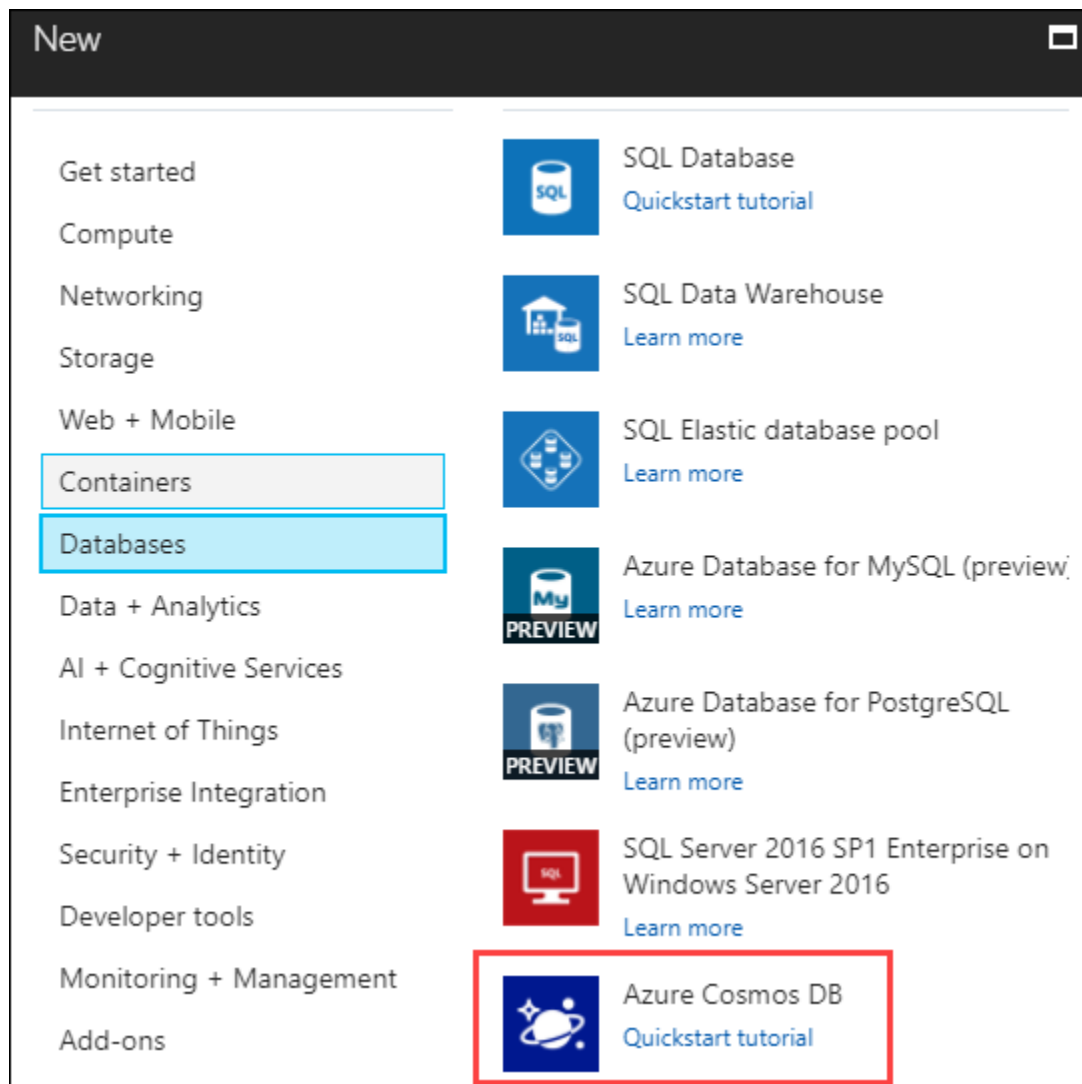
2. Click on + New on the left side of the Dashboard.



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



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



5. In the create blade that come up, configure the settings as follows:
- ID : **uniqueusername** (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**

Azure Cosmos DB
New account

* ID
[Redacted]

* API ⓘ
MongoDB

* Subscription
[Redacted]

* Resource Group ⓘ
☐ Create new
 ☒ Use existing
 [Redacted]

* Location
South Central US

☐ Enable geo-redundancy ⓘ

☐ Pin to dashboard

Create [Automation options](#)

And then Click on Create.

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



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

Notifications

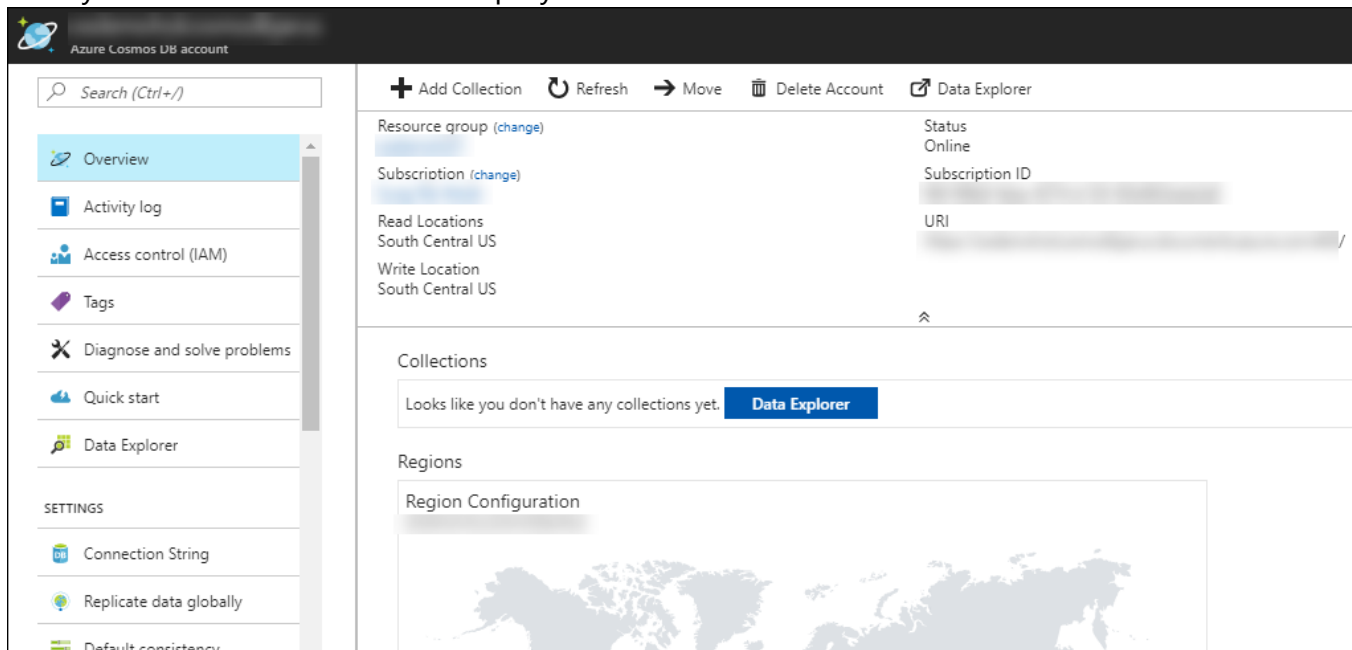
Dismiss: [Informational](#) [Completed](#) [All](#)

✓ [Deployment succeeded](#)

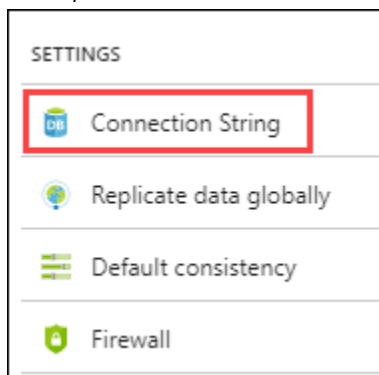
Deployment 'Microsoft.DocumentDB' to resource group '[Redacted]' was successful.

Go to resource **Pin to dashboard**

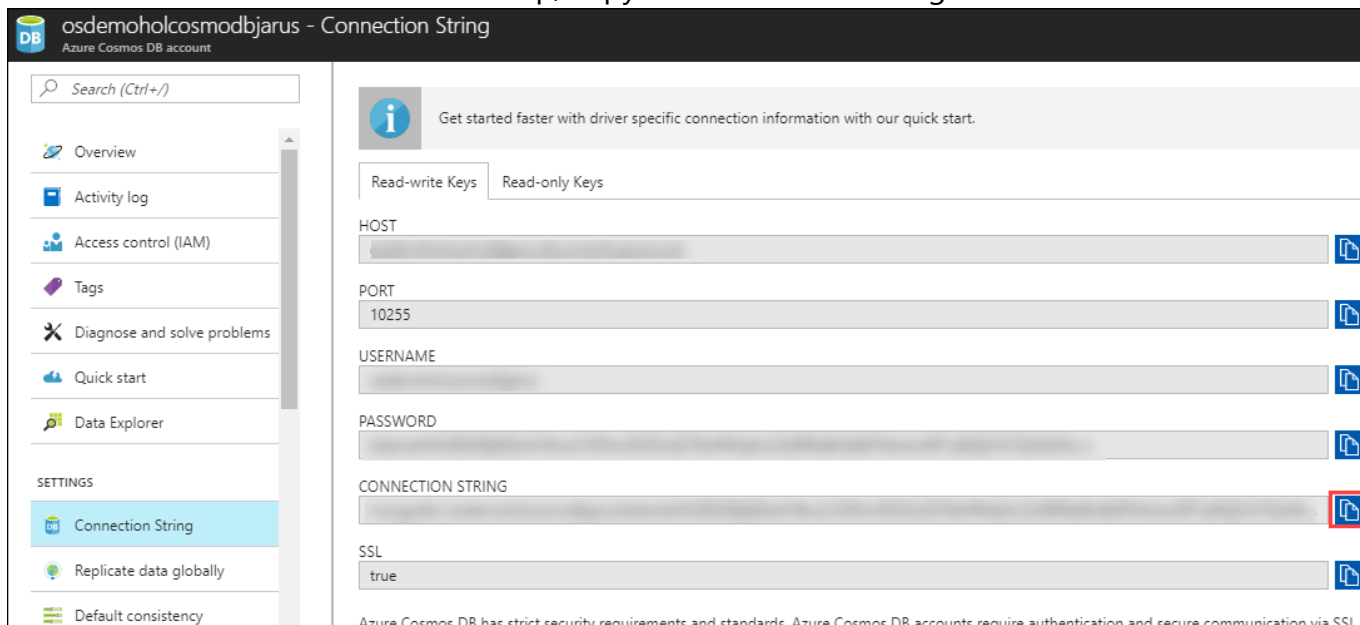
8. Now you will be directed to the deployed database.



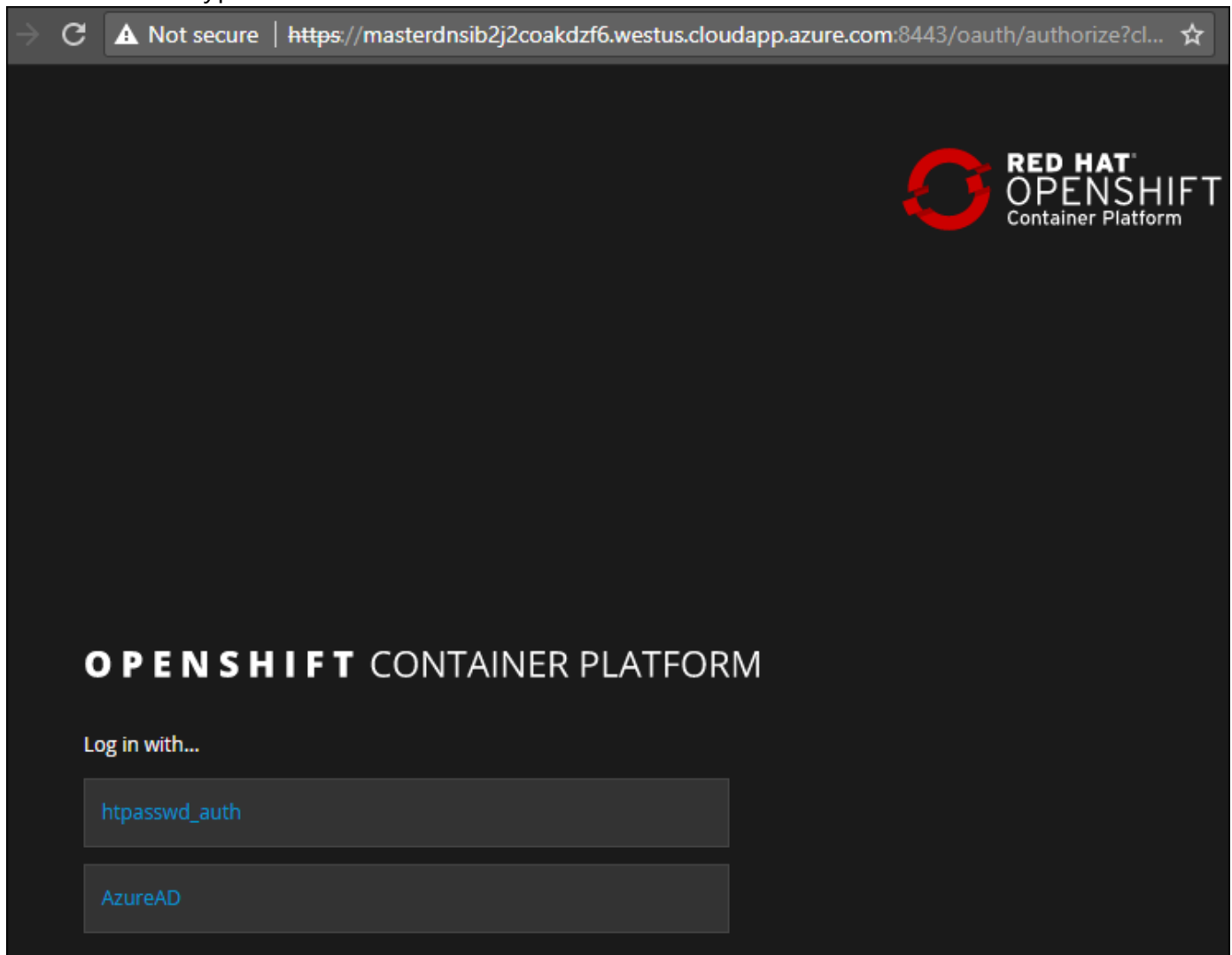
9. Now, click on Connection String under settings menu on the left side of the blade.



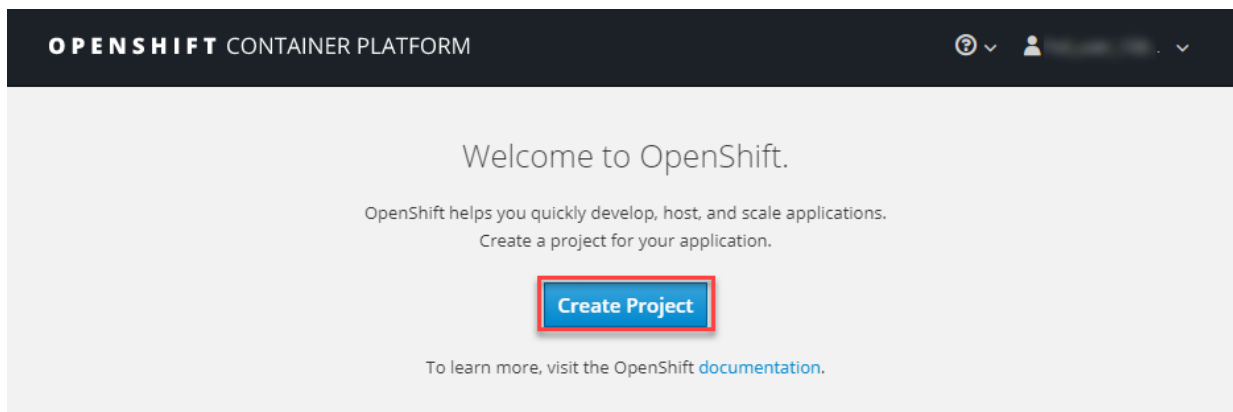
10. Now from the new blade that come up, copy the connection string for later use.



11. Now, open a new tab in a browser 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.



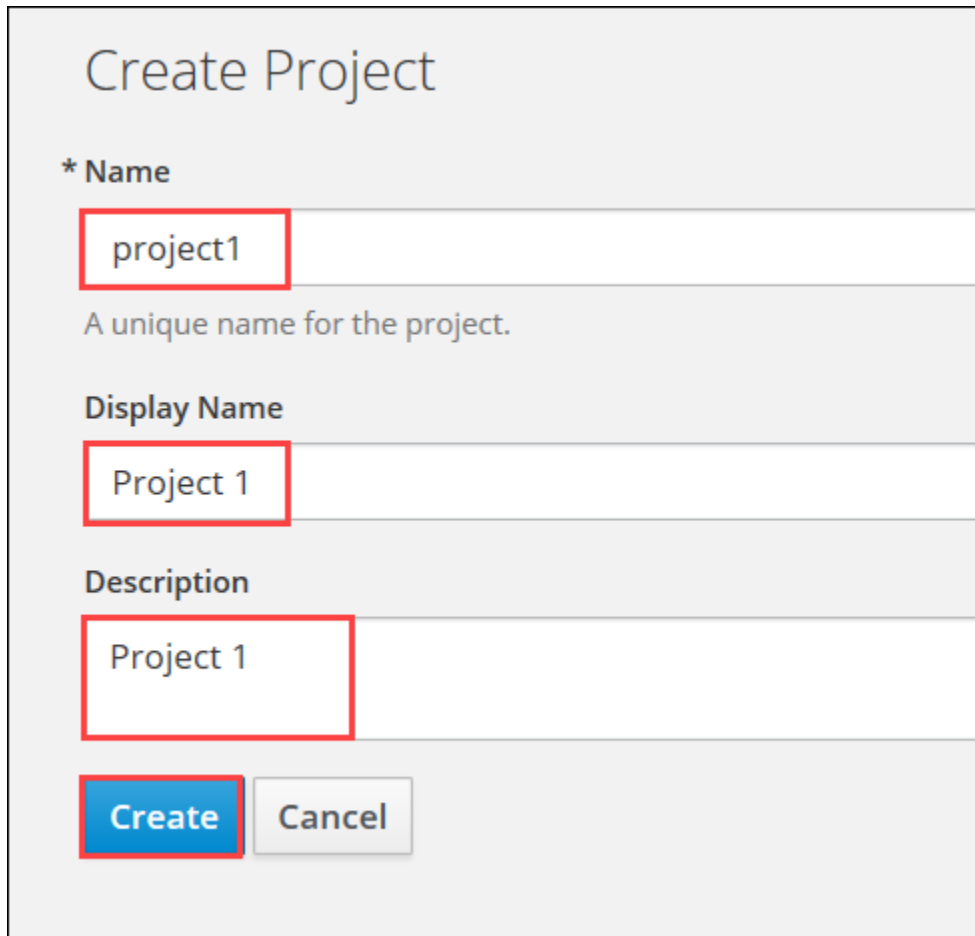
12. Once you are logged in to the console, click on Create Project



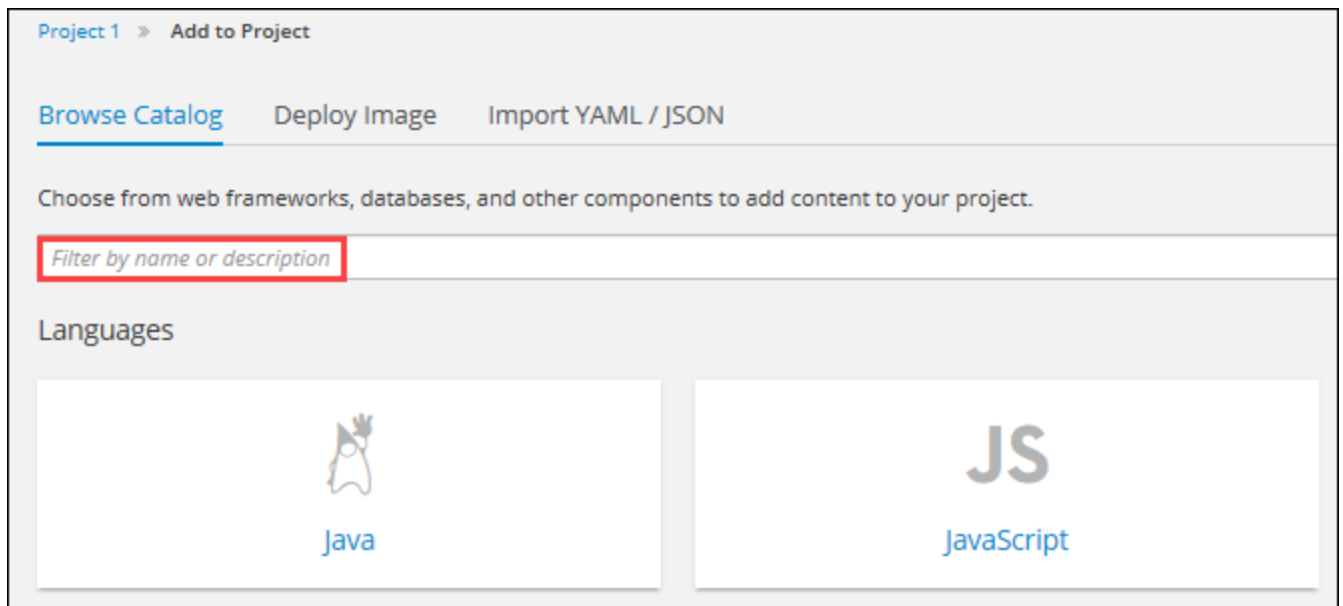
13. **Enter** the **following details** into the Create Project Section, and click **Create**.

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

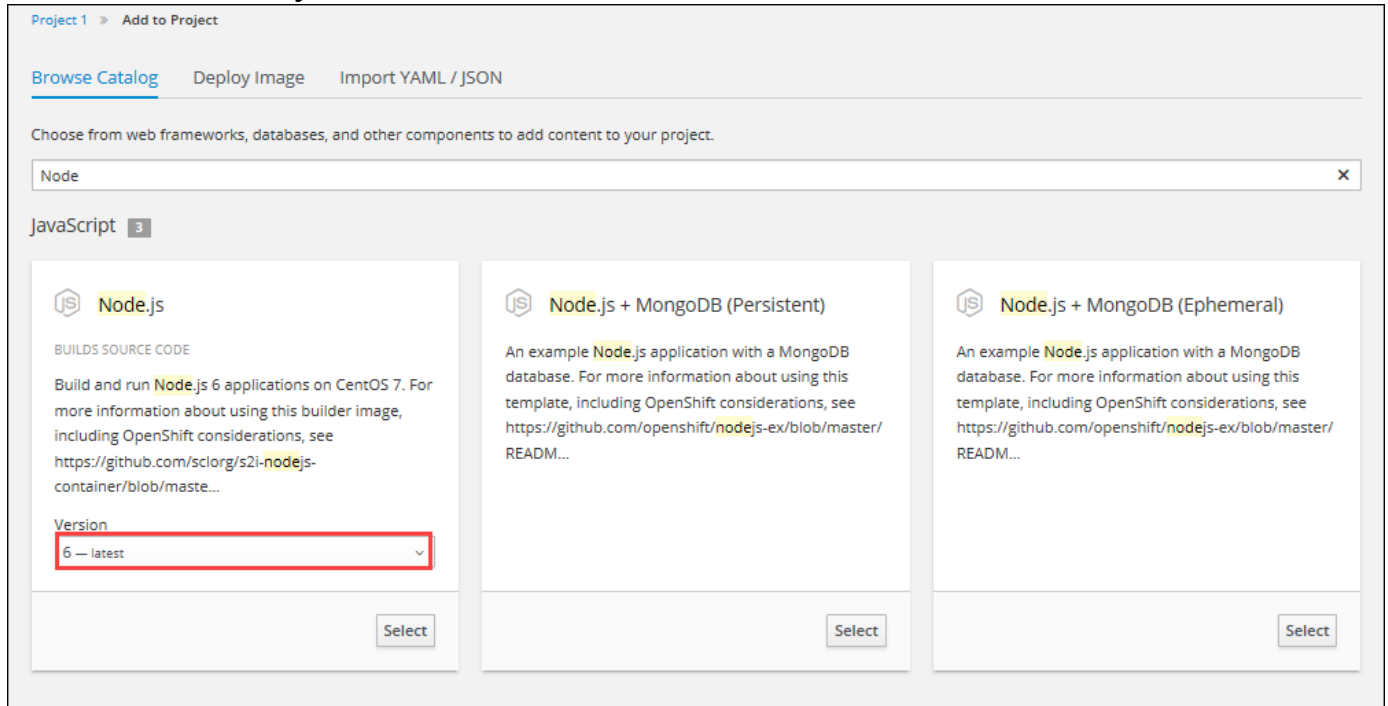
And then **Click Create**.

A screenshot of a 'Create Project' form. The form has a light gray background and a white border. At the top, the title 'Create Project' is displayed in a large, dark gray font. Below the title, there are three input fields, each with a label above it. The first field is labeled '* Name' and contains the text 'project1'. Below this field, a smaller, lighter gray text says 'A unique name for the project.' The second field is labeled 'Display Name' and contains the text 'Project 1'. The third field is labeled 'Description' and contains the text 'Project 1'. At the bottom of the form, there are two buttons: a blue 'Create' button and a gray 'Cancel' button. All three input fields and the 'Create' button are highlighted with a red rectangular border.

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



15. 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**



16. 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>

Node.js

Build and run Node.js 6 applications on RHEL 7. For more information about using this builder image, including OpenShift considerations, see <https://github.com/sclorg/s2i-nodejs-container>.

Version: 6

* Name

Identifies the resources created for this application.

* Git Repository URL

Sample repository for nodejs: <https://github.com/openshift/nodejs-ex.git> [Try It ↗](#)

Show [advanced options](#) for source, routes, builds, and deployments.

[Create](#) [Cancel](#)

17. 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**.

Deployment Configuration [About Deployment Configuration](#)

Autodeploy when

☒ New image is available

☒ Deployment configuration changes

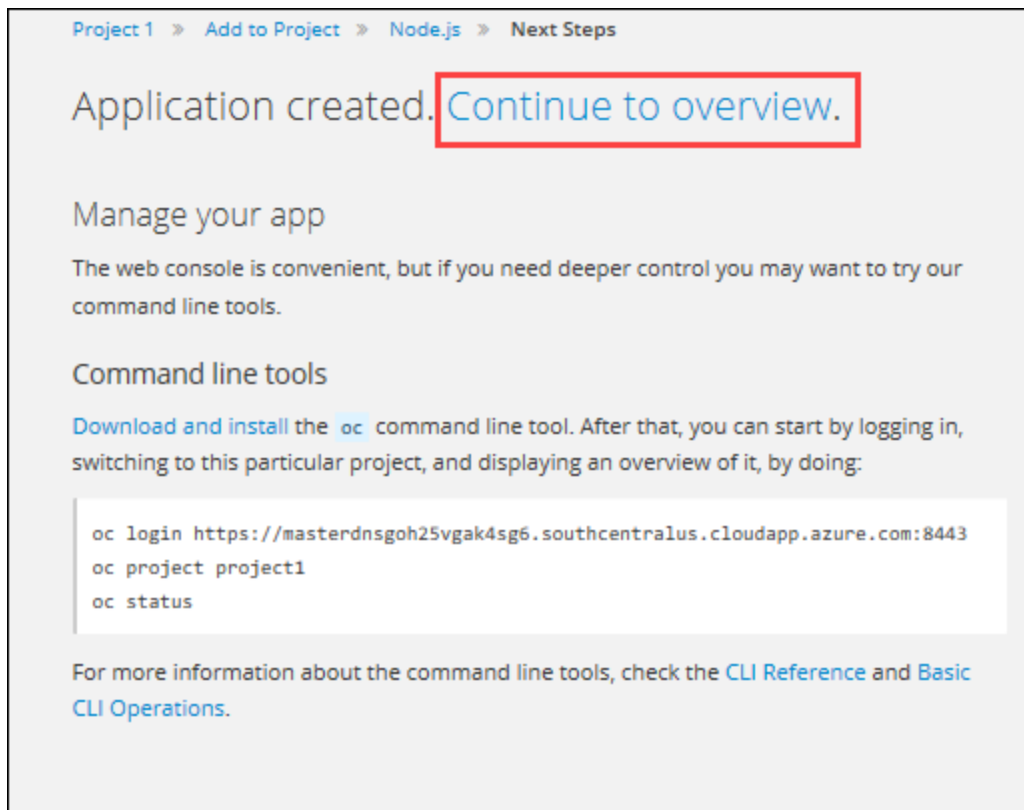
Environment Variables (Runtime only) ⓘ

[Show Image Environment Variables](#)

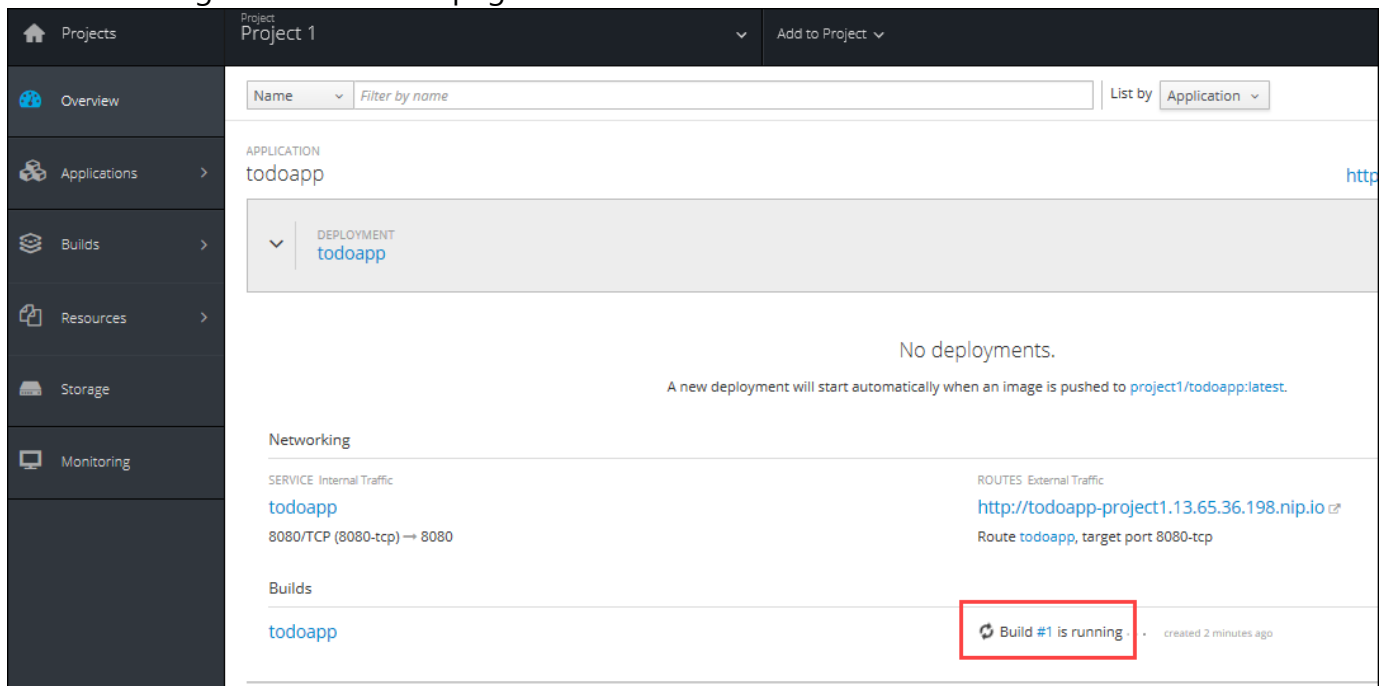
✕

[Add Environment Variable](#) | [Add Environment Variable Using a Config Map or Secret](#)

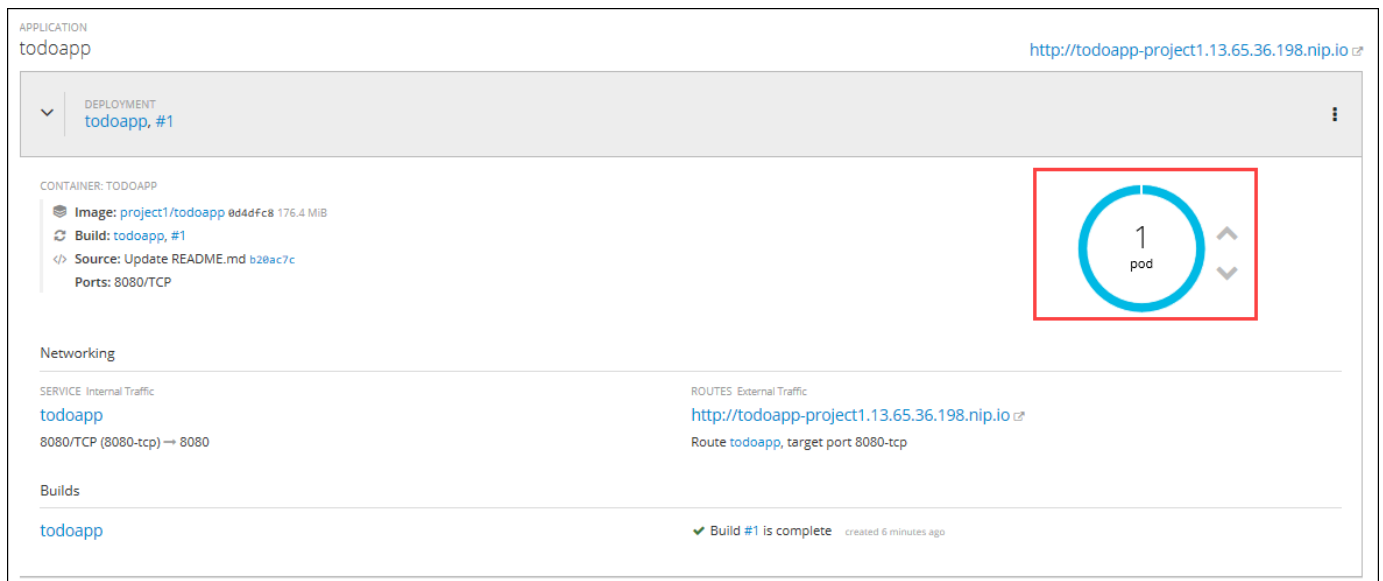
18. Once you click on Create, you will be redirected to Application Created page. Click on Continue to Overview page.



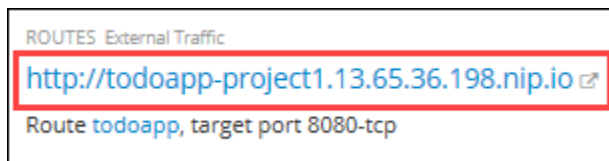
19. Now you will be directed to Project overview page and you can see that the build is running on the bottom right section of the page.



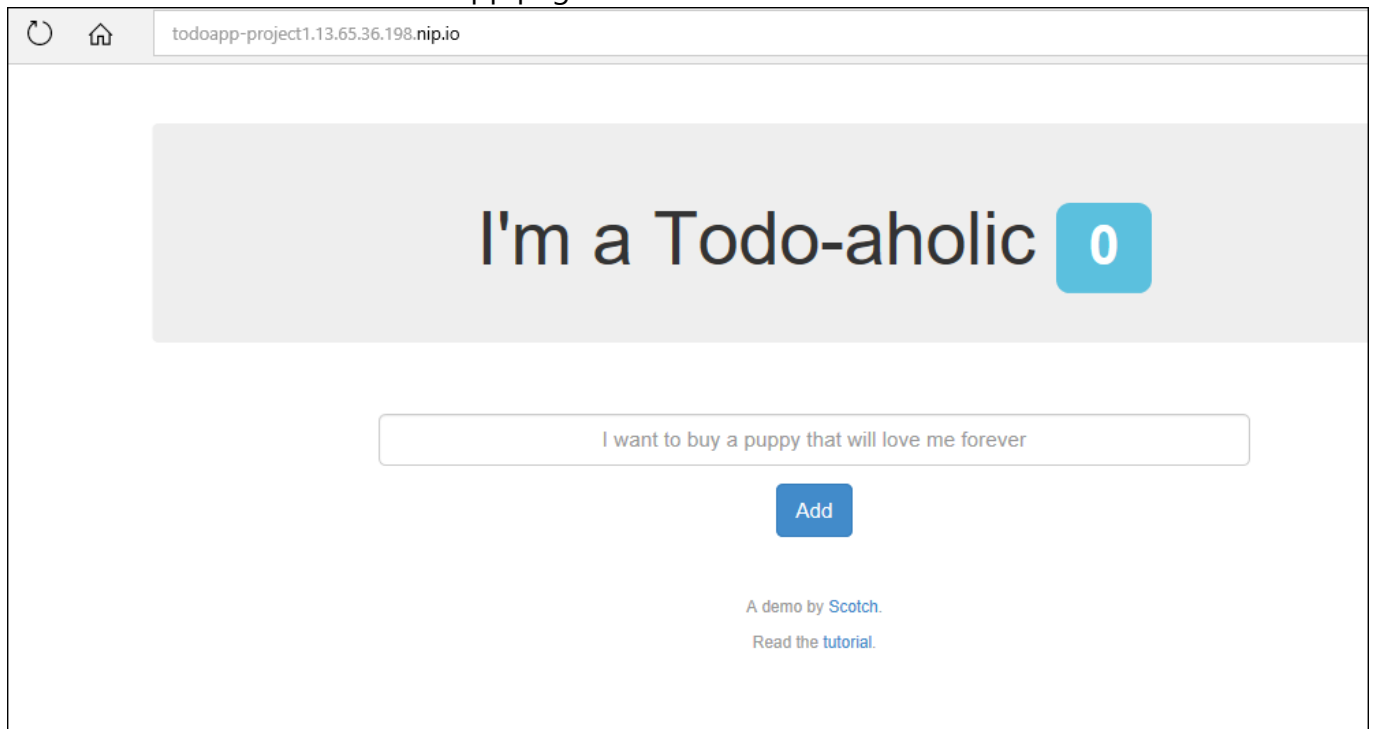
20. Wait until the build is completed and the colour of the circle surrounding the pod turn blue.



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



22. You will be directed to the todoapp page.



23. Now provide the following value in the text field above Add button and then click on Add Button.

“Deployed a 2 Tier Web App”

I'm a Todo-aholic

1

☐ Deployed a 2 Tier Web App

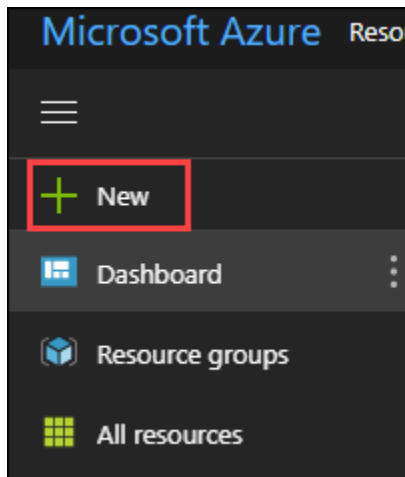
I want to buy a puppy that will love me forever

Add

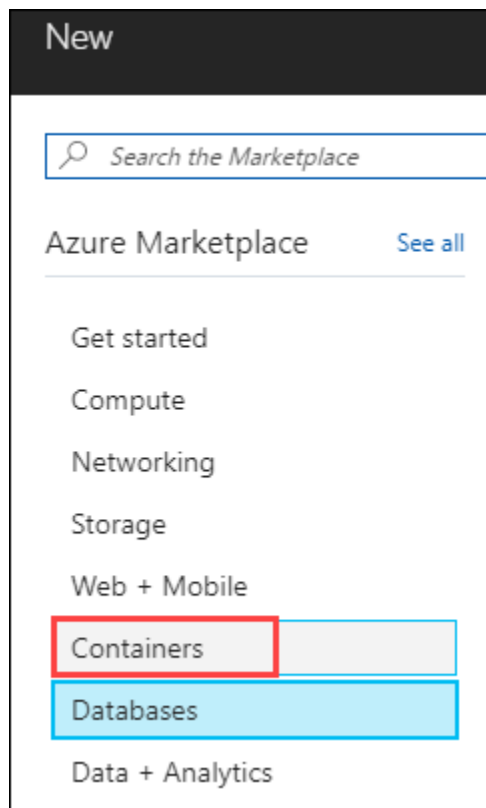
Exercise 6: Integrate ACR with OpenShift

In this exercise, you will deploy an Azure Container Registry and integrate it with Open Shift.

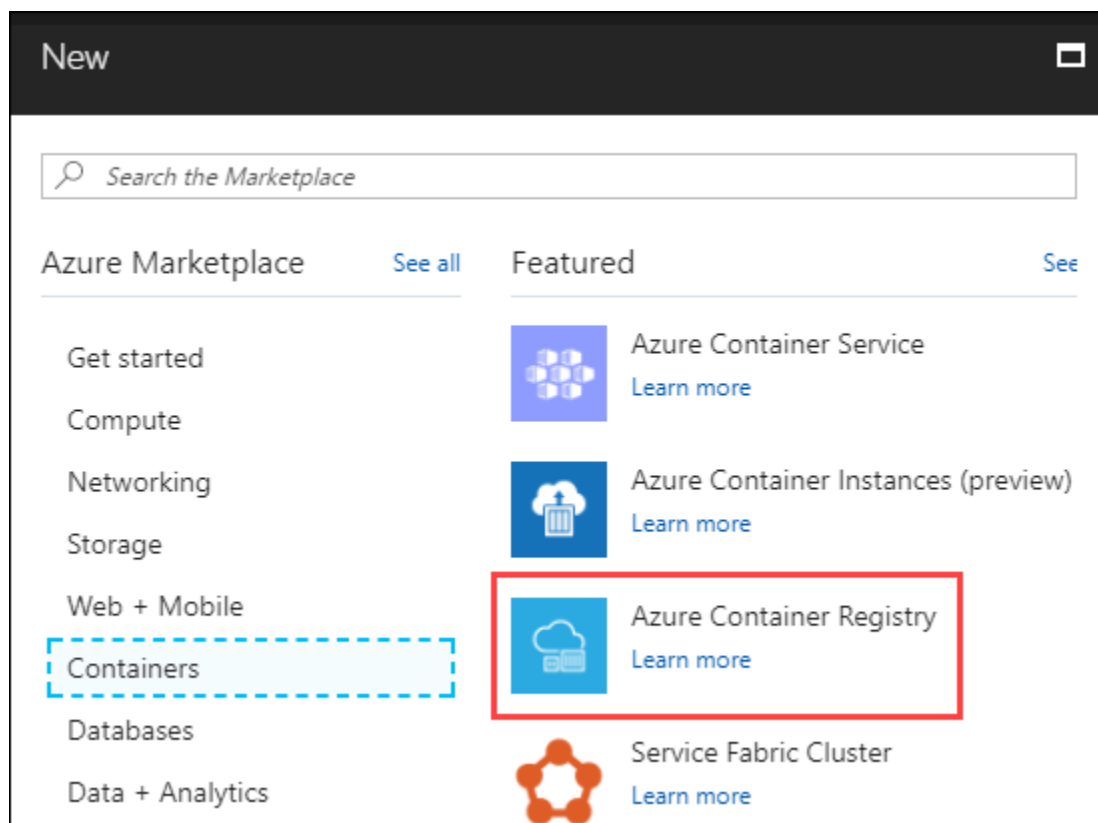
1. **Launch** a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.
2. Click on **+New** on the left side of the Dashboard.



3. In the **New** blade that come up, Select **Containers**.

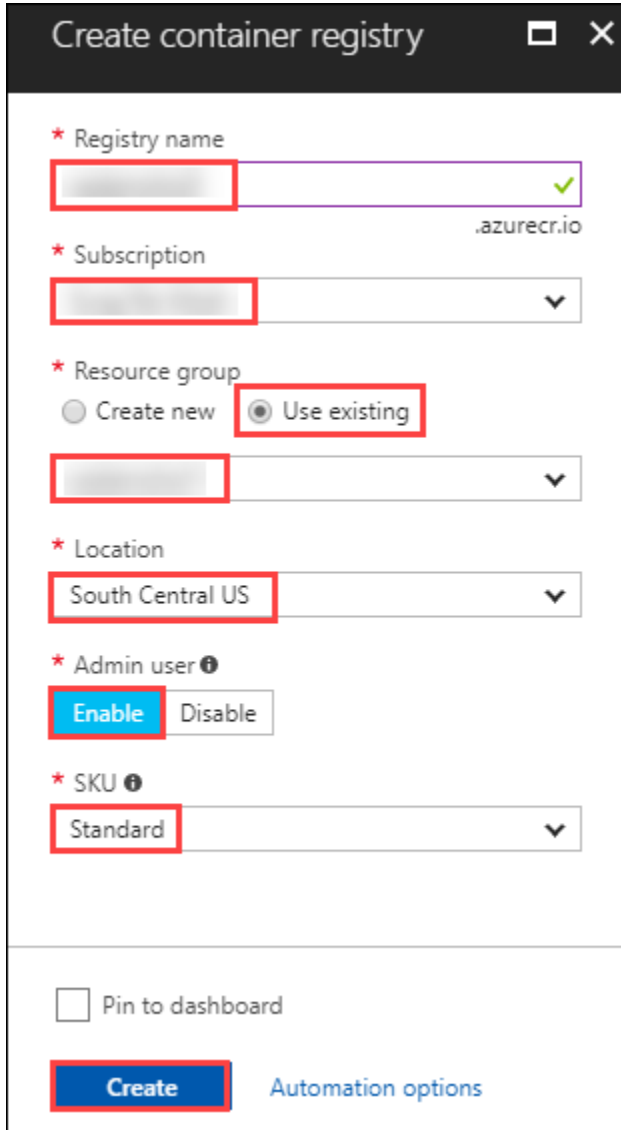


4. In the **Containers** blade appears. Select **Azure Container Registry**.



5. In the create blade that come up, configure the settings as follows:
- Registry name : **uniquename** (This name should be unique across Azure.)

- 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**
- Admin user: Select **Enable**
- SKU: Standard



Create container registry

* Registry name ✓ .azurecr.io

* Subscription

* Resource group ☐ Create new ☒ Use existing

* Location South Central US

* Admin user Enable Disable

* SKU Standard

☐ Pin to dashboard

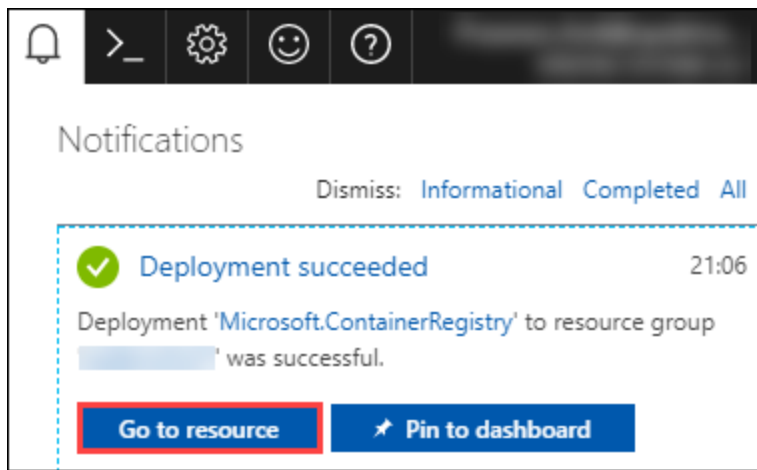
Create [Automation options](#)

And then **Click** on **Create**.

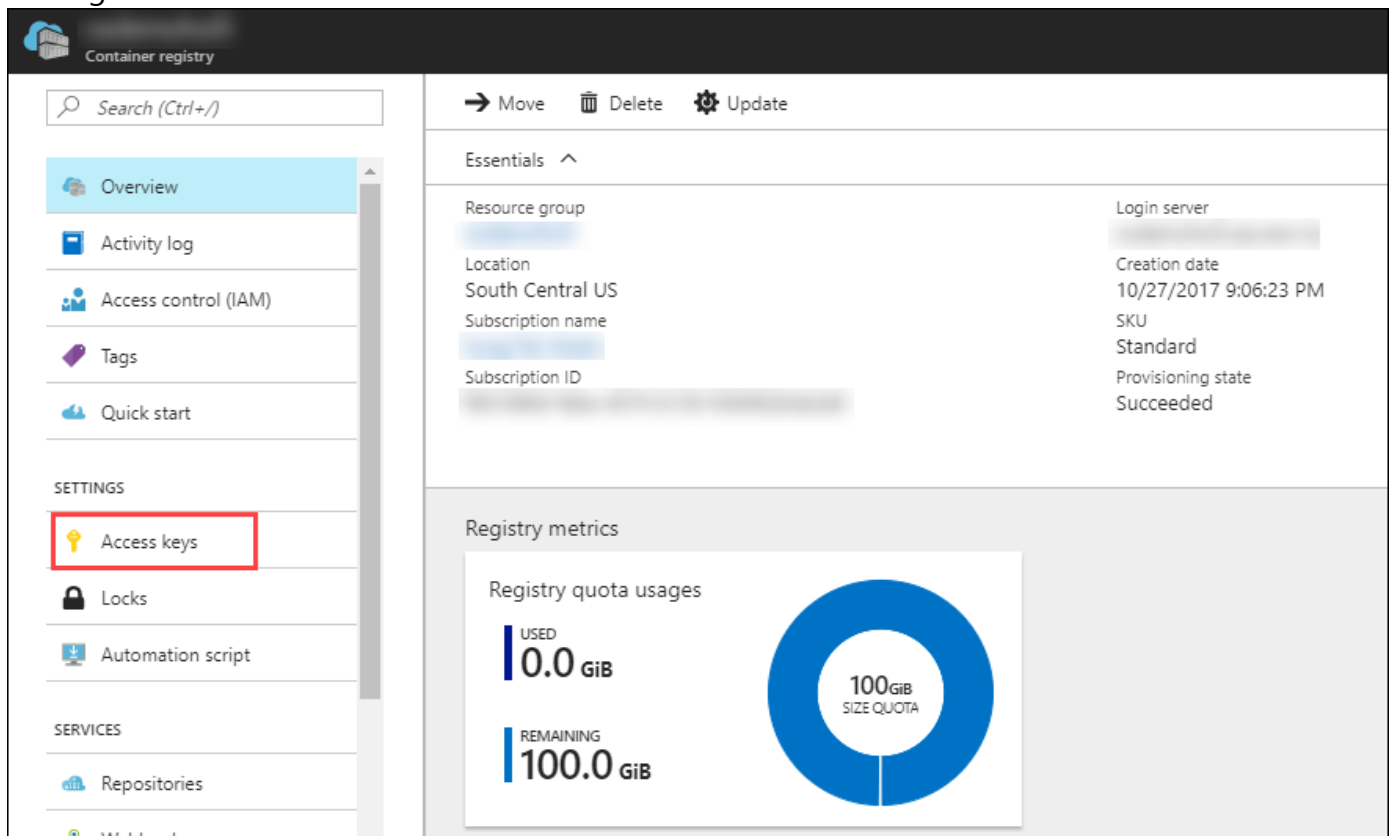
6. You can see the status of the deployment from the notifications tab on top of the page.



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



8. Now you will be directed to the deployed container registry. Click on the Access keys under Settings section which is on the left side of the blade.



9. Now you will be directed to the Access keys blade. Copy the Registry name, Login server, Username and password to a text editor for later use.

Container registry - Access keys

Search (Ctrl+/)

Overview

Activity log

Access control (IAM)

Tags

Quick start

SETTINGS

Access keys

Locks

Automation script

SERVICES

Repositories

Webhooks

Registry name

Login server

Admin user ⓘ


Enable Disable

Username

NAME	PASSWORD
password	
password2	

10. Now, open a new tab in a browser 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.

→ ↻ ⚠ Not secure | <https://masterdnsib2j2coakdzf6.westus.cloudapp.azure.com:8443/oauth/authorize?cl...> ☆

 **RED HAT**
OPENSHIFT
Container Platform

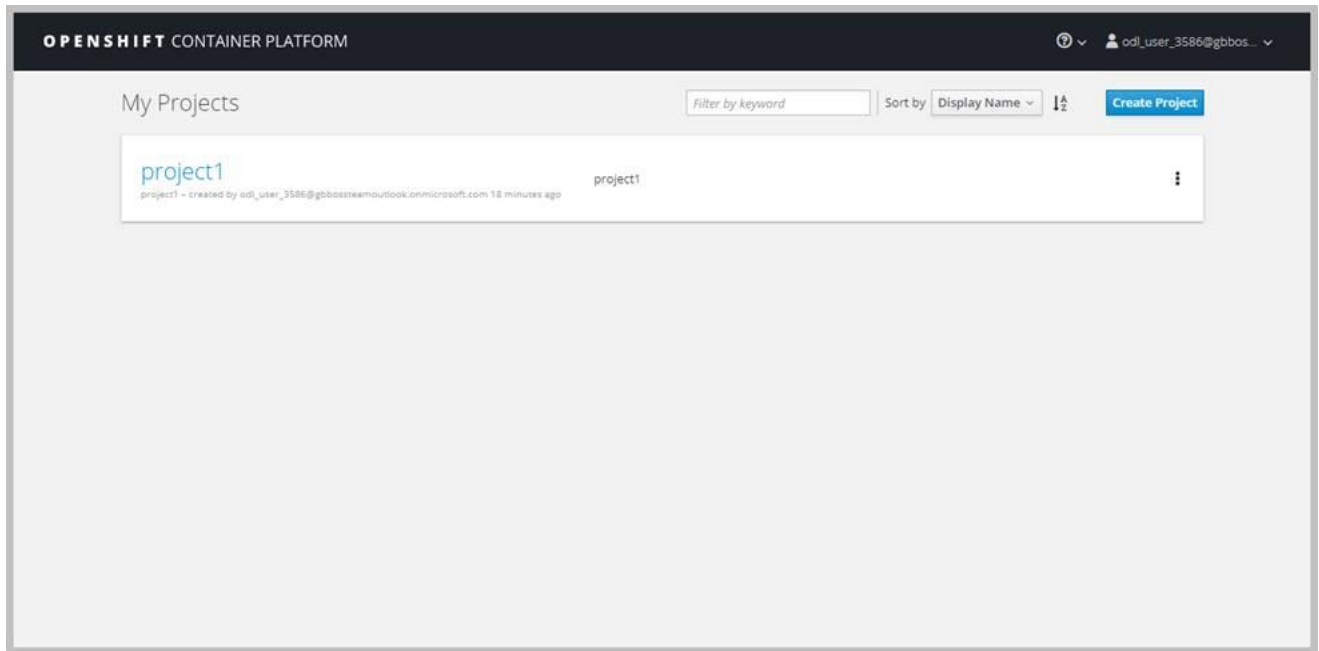
OPENSHIFT CONTAINER PLATFORM

Log in with...

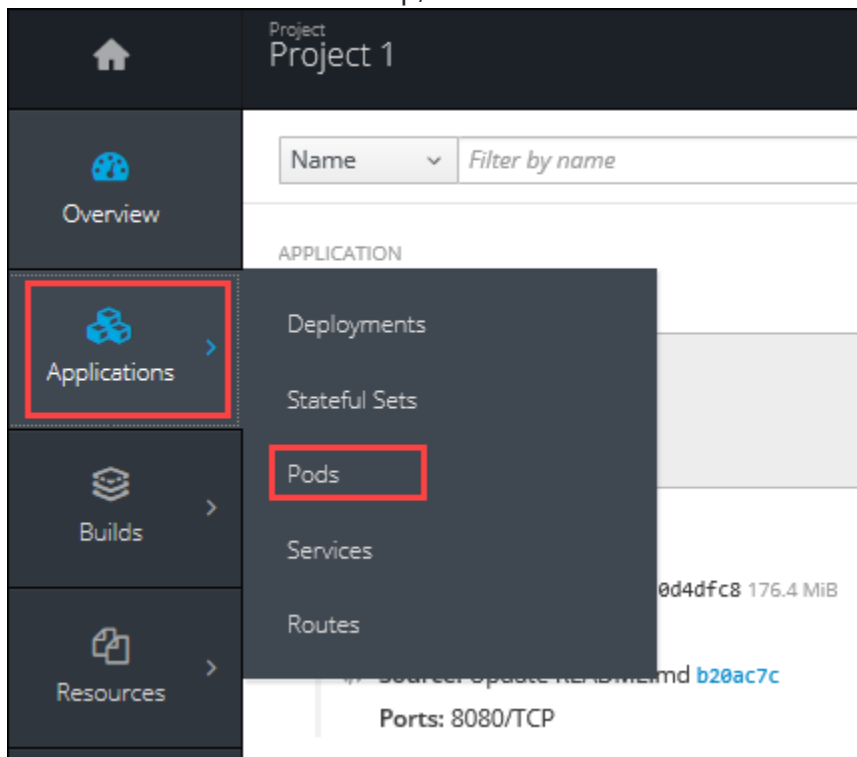
htpasswd_auth

AzureAD

11. Now you will be redirected to the My Projects page, where you will select the Project you created earlier.



12. Once you are in the project page, click on Applications from the left side of the menu and in the new menu that comes up, click on Pods.



13. Now you will be redirected to the Pods page. Click on the pod with status as running.

Project
Project 1


▼

Add to Project ▼

Pods [Learn More](#)

Filter by label

Add

Name	Status	Containers Ready	Containers
todoapp-1-5hgc3	 Running	1/1	1

14. Now you will be directed to the Details blade of the selected Pod.

Pods > todoapp-1-5hgc3

todoapp-1-5hgc3 created 3 days ago

app todoapp deployment todoapp-1 deploymentconfig todoapp

Details Environment Logs Terminal Events

Status

Status: Running

Deployment: todoapp, #1

IP:

Node: aaci-node-0 (10.2.0.4)

Restart Policy: Always

Container todoapp

State: Running since Oct 30, 2017 12:21:51 PM

Last State: Terminated at Oct 27, 2017 10:12:20 PM (Completed)

Ready: true

Restart Count: 1

15. From the details blade, copy the Private Ip Address of the node in which the Pod is running.

Status

Status: Running

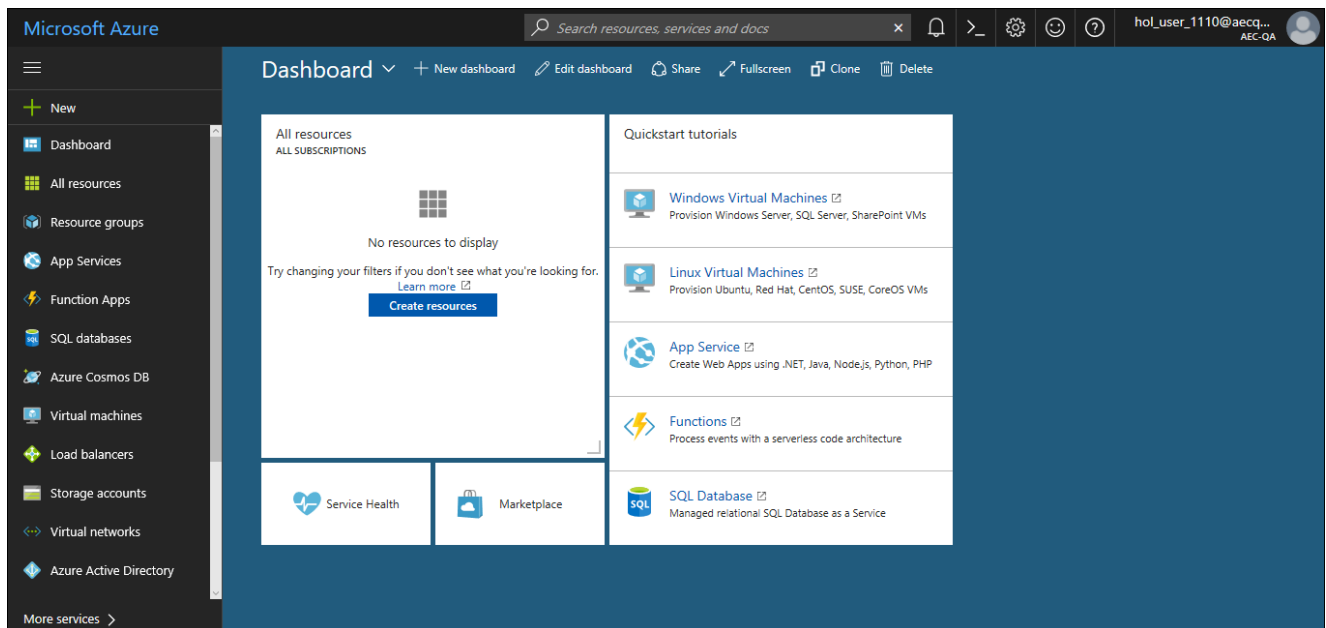
Deployment: todoapp, #1

IP:

Node: aaci-node-0 (10.2.0.4)

Restart Policy: Always

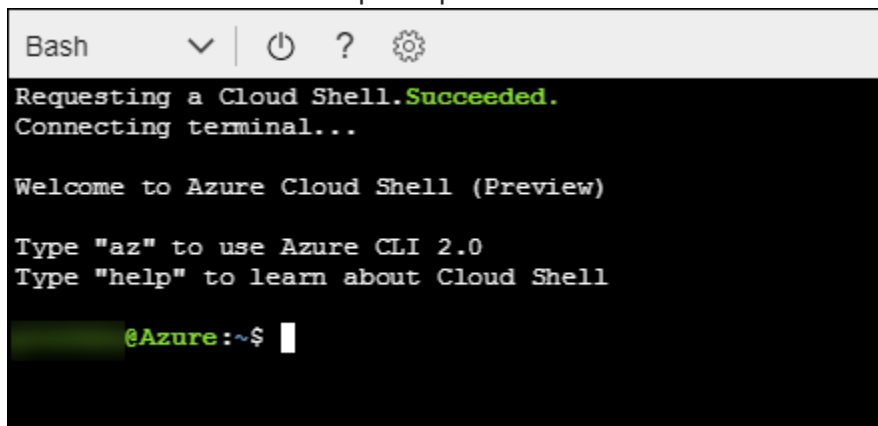
10. Now, **Open** a new tab in a browser and **Navigate** to <https://portal.azure.com>. **Login** with the Microsoft Azure credentials you received via email.



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



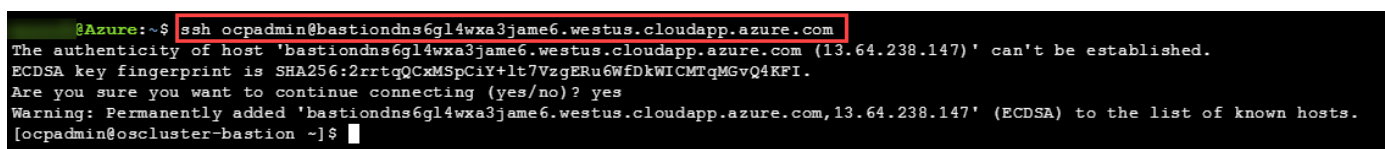
16. Now the bash shell will open up.



17. Now execute the following command. When prompted, type Yes and you will be logged in to the Openshift Master VM.

```
ssh ocpadmin@<copiedDNSNameofBastionVM>
```

Note: Substitute in the above command with the value of copied DNS Name of Bastion VM



18. Now execute the following command in the cloud shell to pull a docker image. Copy the key into a text editor for later use.

```
ssh ocpadmin@<copiedPrivateIpOfNode>
```

Note: Substitute in the above command with the value of copied Private IP Address of Node in which pod is running.

```
[ocpadmin@oscluster-bastion ~]$ ssh ocpadmin@10.2.0.4
Last login: Wed Nov  1 12:21:12 2017 from 10.1.0.6
[ocpadmin@oscluster-node-0 ~]$
```

19. Now execute the following command in the cloud shell to login in to root account.

```
sudo su -
```

```
[ocpadmin@oscluster-node-0 ~]$ sudo su -
[root@oscluster-node-0 ~]#
```

20. Now execute the following command in the cloud shell to check if the docker is installed and running.

```
docker -v
```

```
[root@oscluster-node-0 ~]# docker -v
Docker version 1.12.6, build 85d7426/1.12.6
[root@oscluster-node-0 ~]#
```

21. Now execute the following command in the cloud shell to display the list the docker images in the system.

```
docker images
```

```
[root@oscluster-node-0 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID
docker-registry.default.svc:5000/project1/todoapp	latest	713858a1d317
registry.access.redhat.com/rhsc1/nodejs-6-rhel7	<none>	96a89d880a8b
registry.access.redhat.com/openshift3/registry-console	v3.6	951d2b7d6a26
registry.access.redhat.com/openshift3/ose-sti-builder	v3.6.173.0.49	ecc2cacead49
registry.access.redhat.com/openshift3/ose-deployer	v3.6.173.0.49	030654868e6b
registry.access.redhat.com/openshift3/ose-pod	v3.6.173.0.49	99965fb98423

```
[root@oscluster-node-0 ~]#
```

22. From the displayed results, copy the Image name with todoapp in the end.

```
[root@oscluster-node-0 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID
docker-registry.default.svc:5000/project1/todoapp	latest	713858a1d317
registry.access.redhat.com/rhsc1/nodejs-6-rhel7	<none>	96a89d880a8b
registry.access.redhat.com/openshift3/registry-console	v3.6	951d2b7d6a26
registry.access.redhat.com/openshift3/ose-sti-builder	v3.6.173.0.49	ecc2cacead49
registry.access.redhat.com/openshift3/ose-deployer	v3.6.173.0.49	030654868e6b
registry.access.redhat.com/openshift3/ose-pod	v3.6.173.0.49	99965fb98423

```
[root@oscluster-node-0 ~]#
```

23. Now execute the following command in the cloud shell to tag the existing docker image.


```
docker tag <ImageName> <ACRLoginServerUri>/sample/todoapp
```

```
[root@oscluster-node-0 ~]# docker tag docker-registry.default.svc:5000/project1/todoapp .azurecr.io/sample/todoapp
[root@oscluster-node-0 ~]#
```

Note: Substitute for **ImageName** and **ACR Login Server** URI with the copied values in the above command

24. Now execute the following command in the cloud shell to login to docker registry. When prompted, enter the password for ACR you copied earlier

```
docker login <acrServerLoginServerUri> -u <ACRUsername>
```

Note: Substitute for ACR Login Server URI and Username in the above command

```
[root@oscluster-node-0 ~]# docker login .azurecr.io -u
Password:
Login Succeeded
[root@oscluster-node-0 ~]#
```

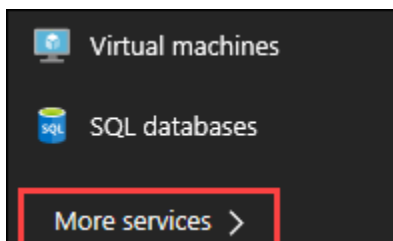
25. Now execute the following command in the cloud shell to push the tagged image to azure container Registry. Copy the key into a text editor for later use.

```
docker push <ACRLoginServerUri>/sample/todoapp
```

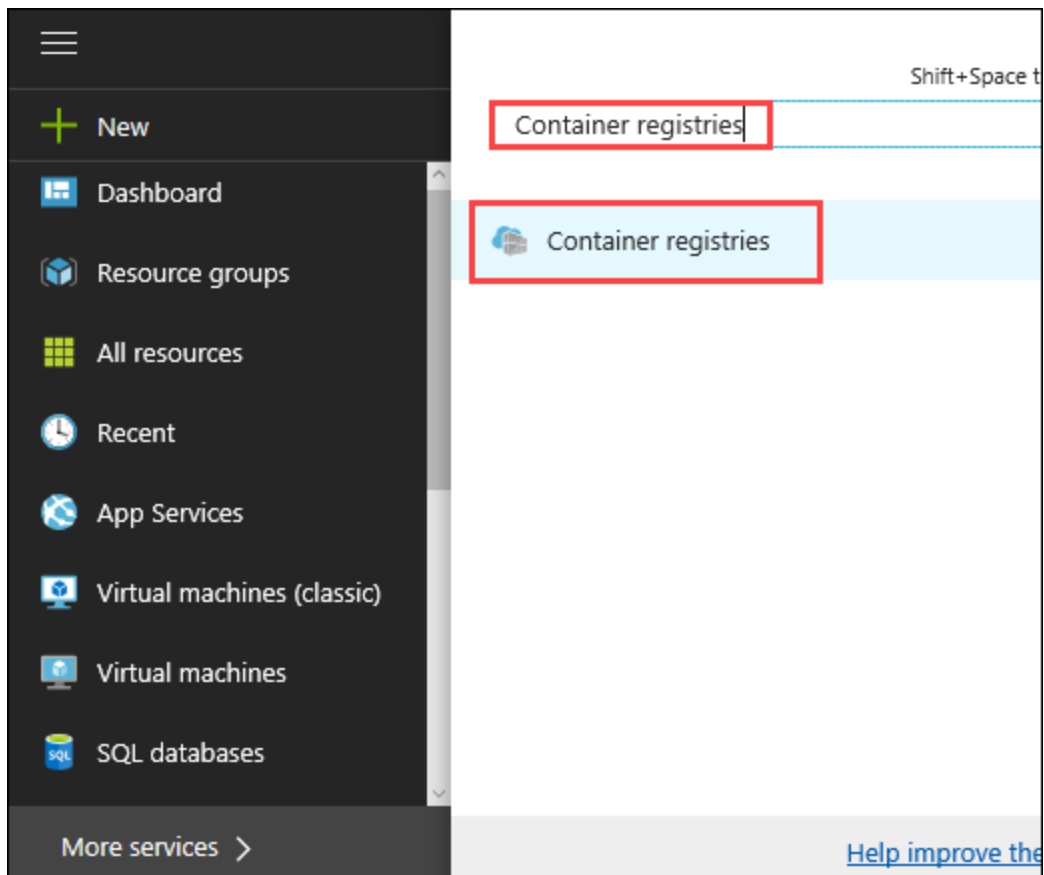
Note: Substitute for ACRLoginServerUri in the above command

```
[root@oscluster-node-0 ~]# docker push .azurecr.io/sample/todoapp
The push refers to a repository [.azurecr.io/sample/todoapp]
4187a9a39095: Pushed
23ba8d7e8e50: Pushed
a08242f55a1c: Pushed
67758eed08ef: Pushed
273d61014330: Pushed
1afb15ed6241: Pushed
latest: digest: sha256:2c31bb7d0aad41072b65616b09d513dfb1851cde013eaf320d33d9b0b2ea0e0a size: 1582
[root@oscluster-node-0 ~]#
```

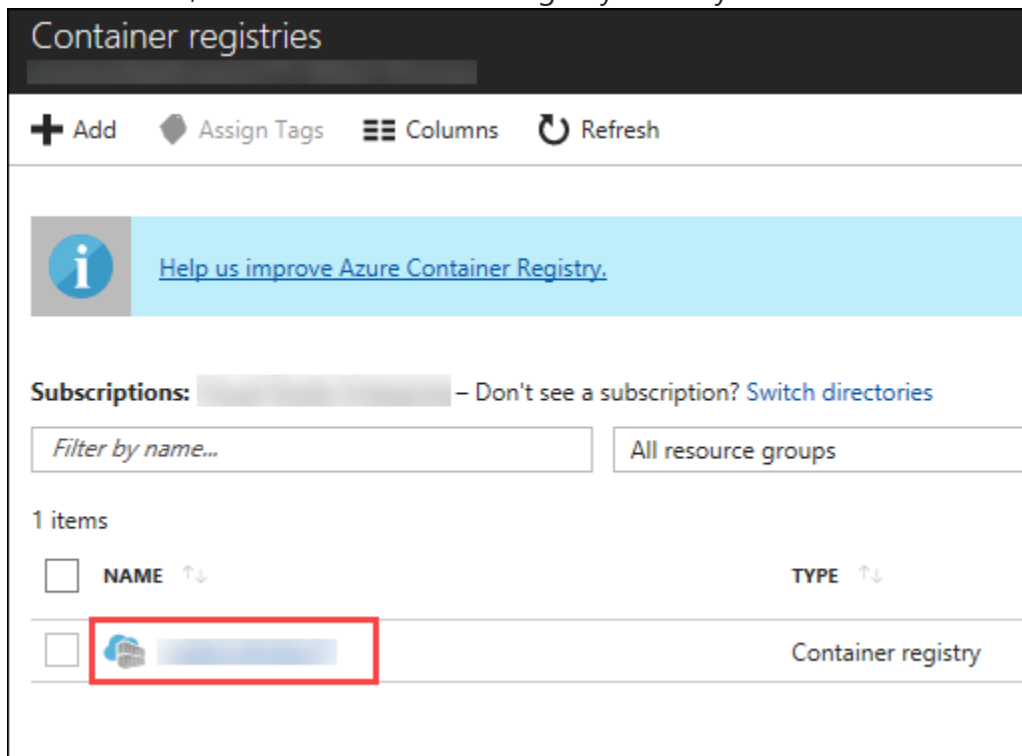
26. Once you have pushed the image to Azure Container Registry, **click** on **More services** on the left side of the menu on the dashboard.



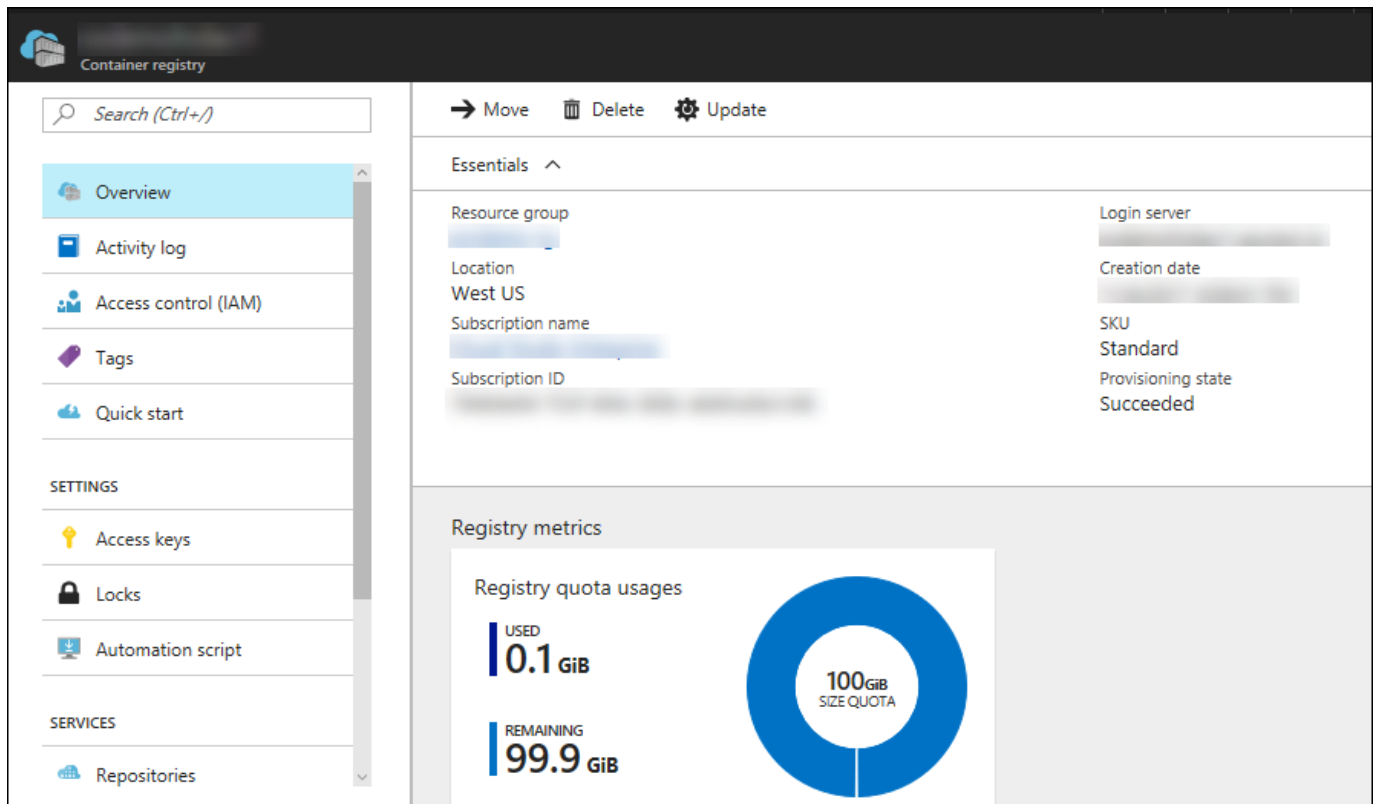
27. In the new blade that come up, search in the Filter box at the top "**Container registries**" and then Select "**Container Registries** from the search result.



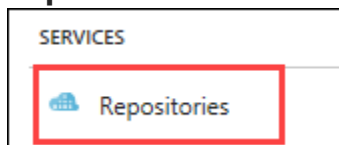
28. On the blade, **select** the Container Registry which you have created.



29. Now you will be directed to the Overview page of the container registry.



30. Now to check whether the image has been pushed to the repository, you can **click** on **Repositories** under Services on the menu on left side of the blade.



31. In the next blade that come up, if the push has been successful, you can see **sample/todapp** repository there.

