AEC Open Shift | Lab Guide

Contents

AEC Open Shift | Lab Guide 1
Workshop Architecture and Objective 2
- 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 6

Lab 2: Deploying Open Shift cluster using ARM templates 9
- Lab Overview 9
- Prerequisites 9
- Time Estimate 9
- Exercise 1: Create an Azure AD Application 9
- Exercise 2: Create a Keyvault 14
- Exercise 3: Deploy Openshift Cluster using ARM Template 18
- Exercise 4: Configure Azure AD Authentication 24

Lab 3: Deploying workload on Openshift 37
- Lab overview 37
- Prerequisites 37
- Time Estimate 37
- Exercise 1: Deploy a 2 Tier Node JS Application on Open Shift 38
- Exercise 2: Installing OpenShift CLI 43

Exercise 3: Deployment in OpenShift using CLI 45
Exercise 4: Create an App using Docker build 49
Lab 4: Integration of ACR with OpenShift........ 53
Exercise 1: Integrate ACR with OpenShift 53

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)

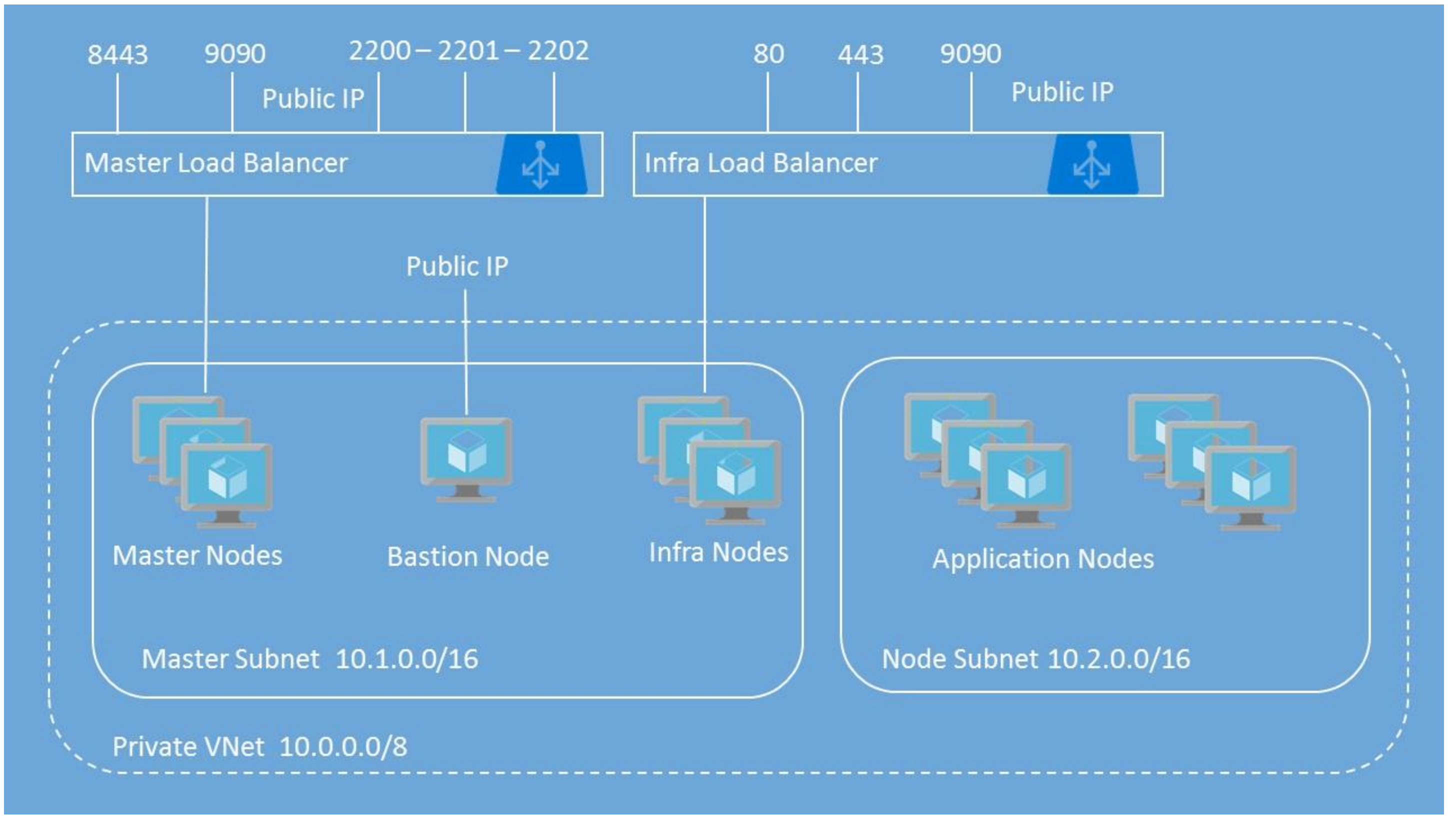
Labs Objective

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.



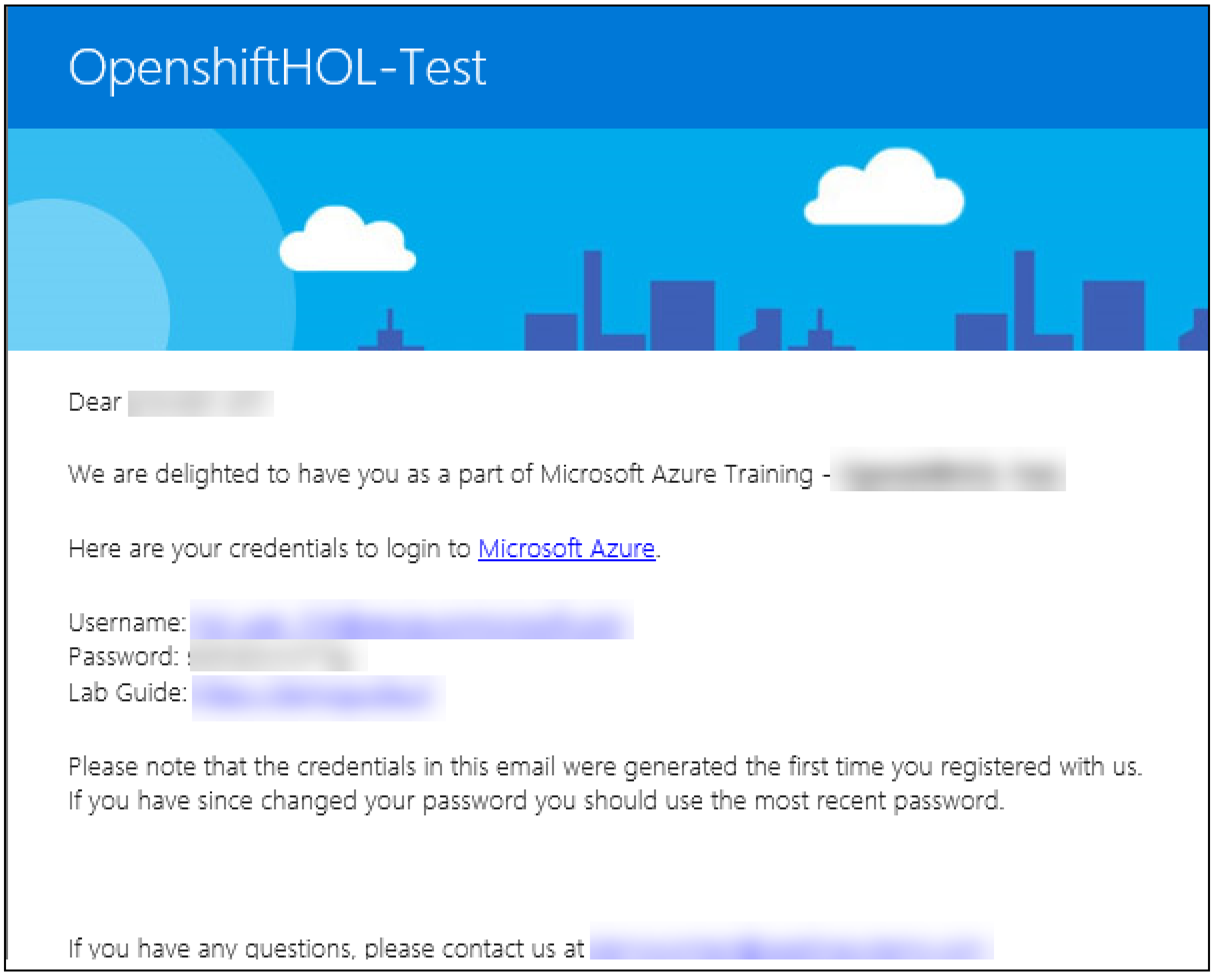
Lab 1: Introduction to Azure Portal

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.



OpenshiftHOL-Test

Dear [redacted]

We are delighted to have you as a part of Microsoft Azure Training - [redacted]

Here are your credentials to login to Microsoft Azure.

Username: [redacted]
Password: [redacted]
Lab Guide: [redacted]

Please note that the credentials in this email were generated the first time you registered with us.
If you have since changed your password you should use the most recent password.

If you have any questions, please contact us at [redacted]

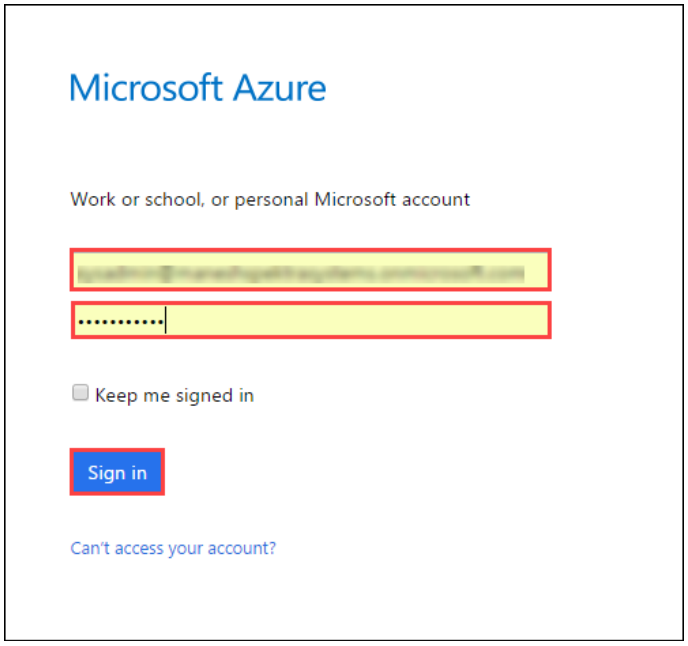
Time Estimate

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**.

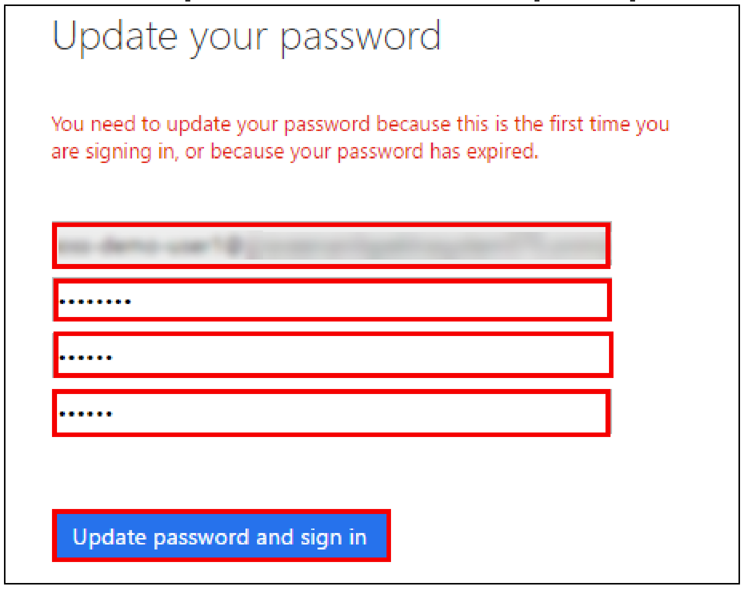


Microsoft Azure

Work or school, or personal Microsoft account

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

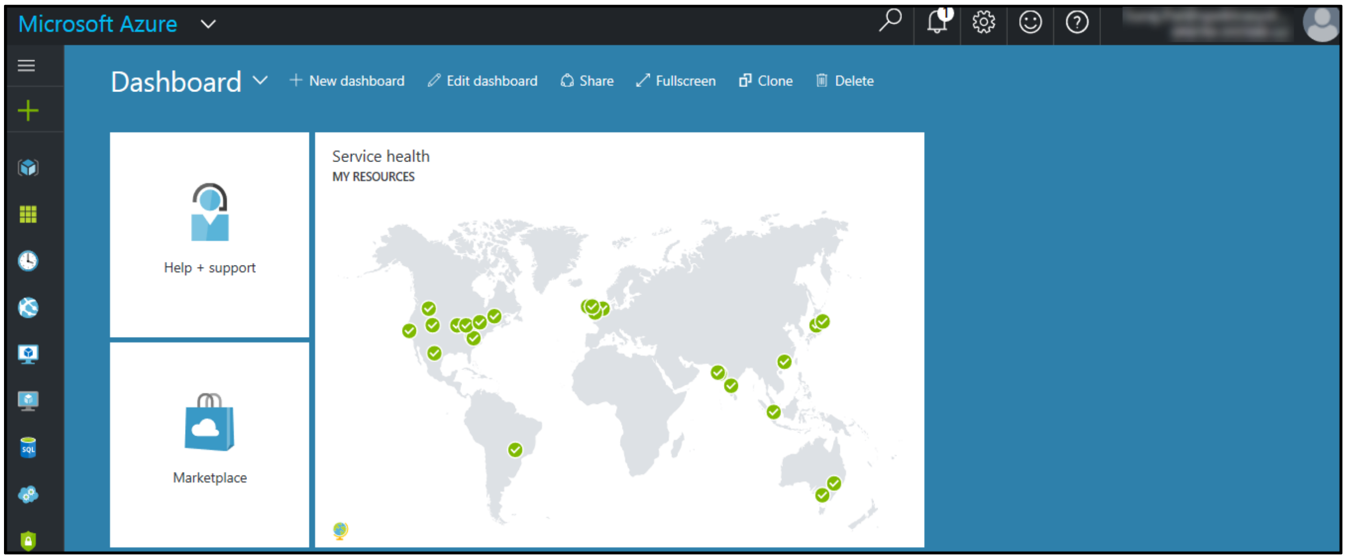
2. **Enter** a new **password**. Then select **Update password and sign in.**



Update your password

You need to update your password because this is the first time you are signing in, or because your password has expired.

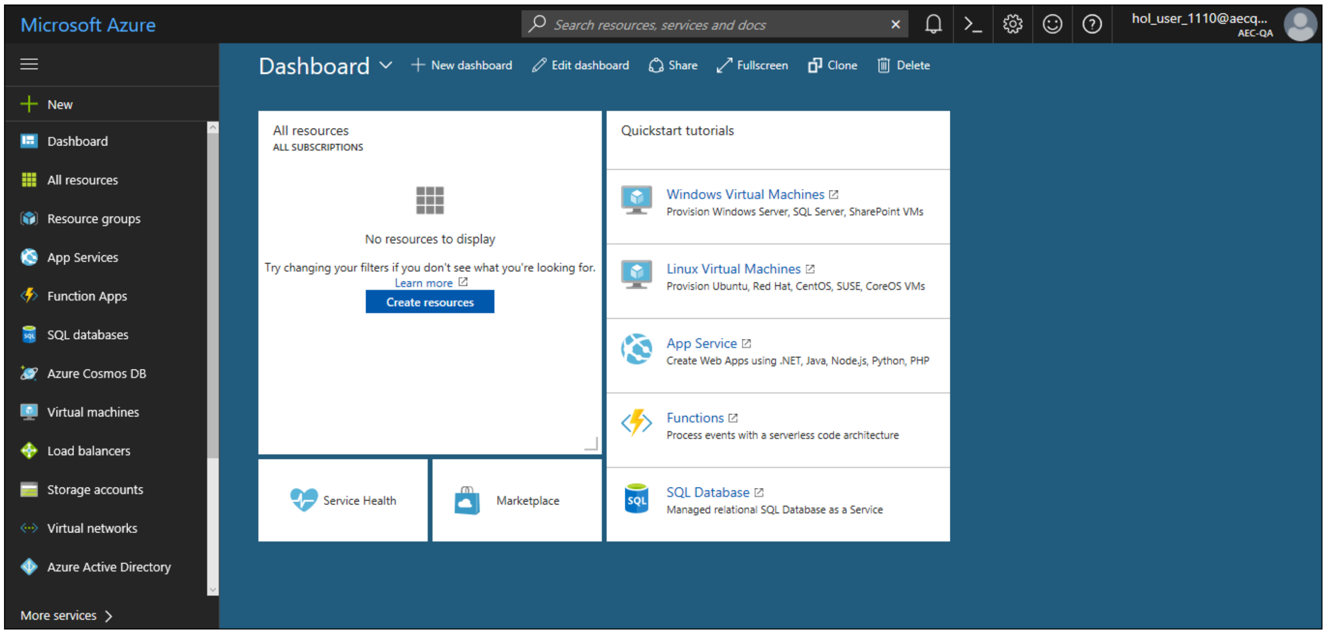
3. Now, you will be directed to the Azure Dashboard



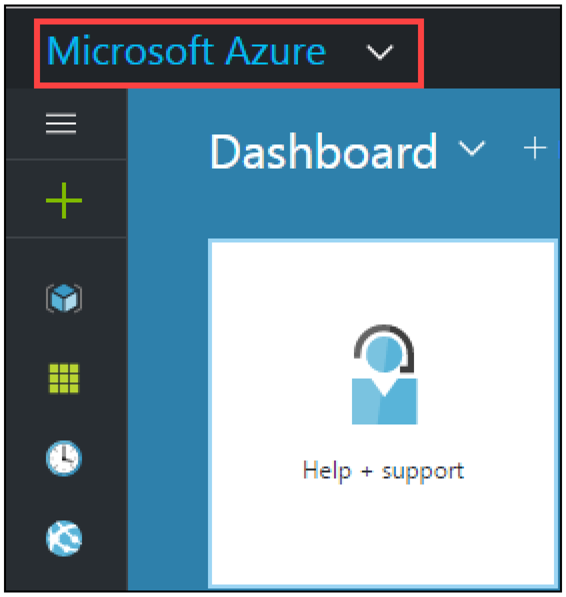
Exercise 2: Verify access to the Subscription

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

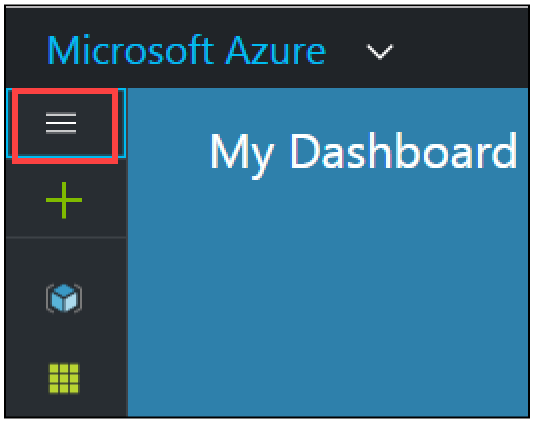
1. **Launch** a browser and **Navigate** to https://portal.azure.com. **Login** with the Microsoft Azure credentials you received via email.



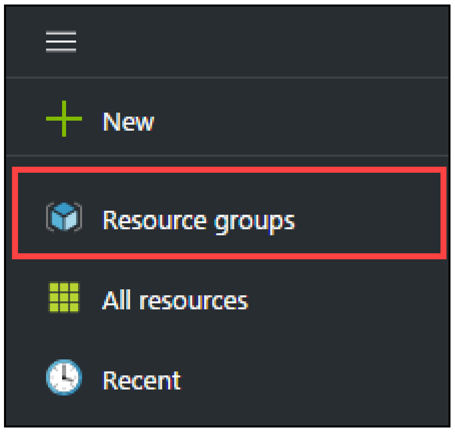
2. **Click** on **Microsoft Azure** at the top left corner of the screen, to view the Dashboard.



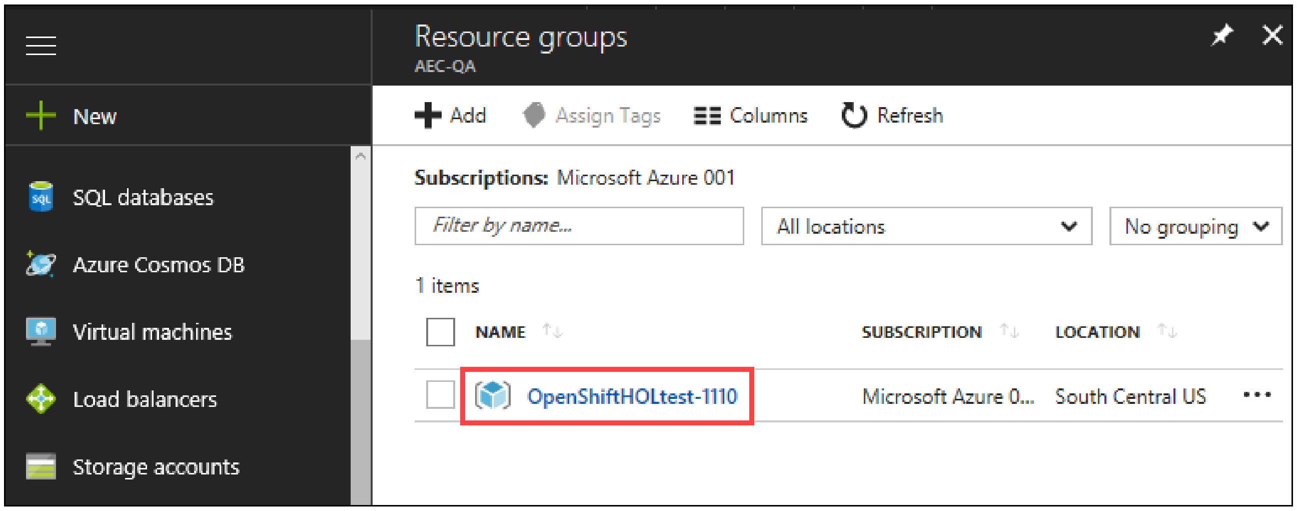
3. To toggle **show/hide** the Portal menu options with icon, **Click** on the **Show Menu** button.



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.


Resource groups
AEC-QA

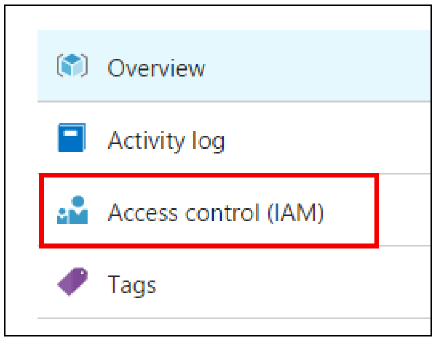
Subscriptions: Microsoft Azure 001

1 items

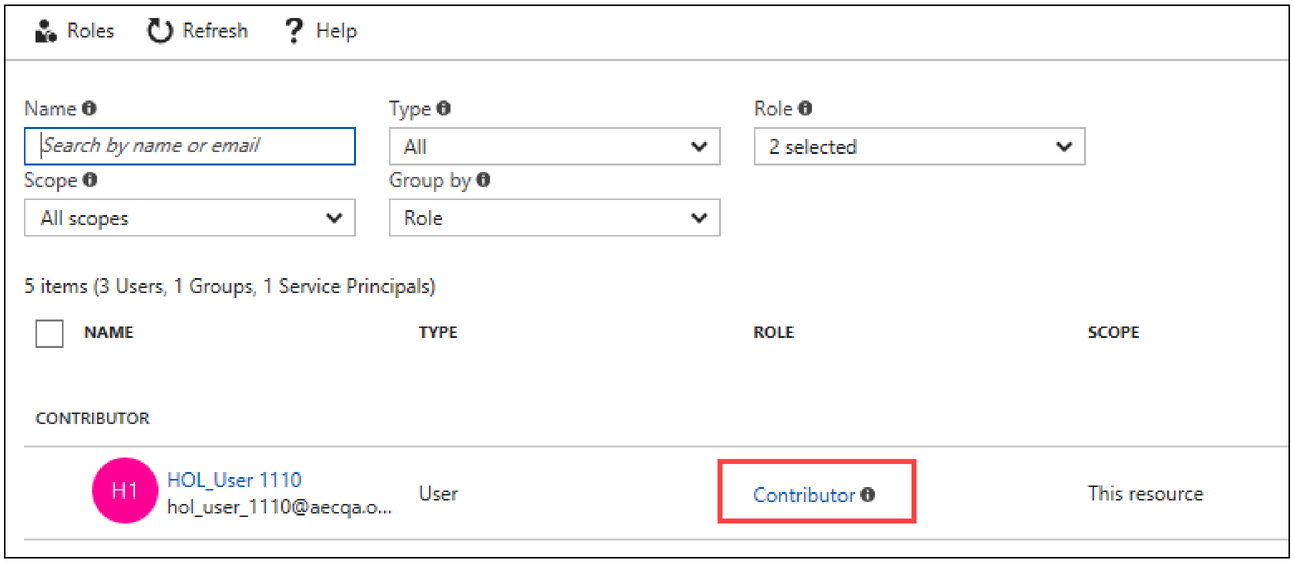
NAME	SUBSCRIPTION	LOCATION
OpenShiftHOLtest-1110	Microsoft Azure 0...	South Central US

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

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



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



Name: Search by name or email
Type: All
Role: 2 selected
Scope: All scopes
Group by: Role

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

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

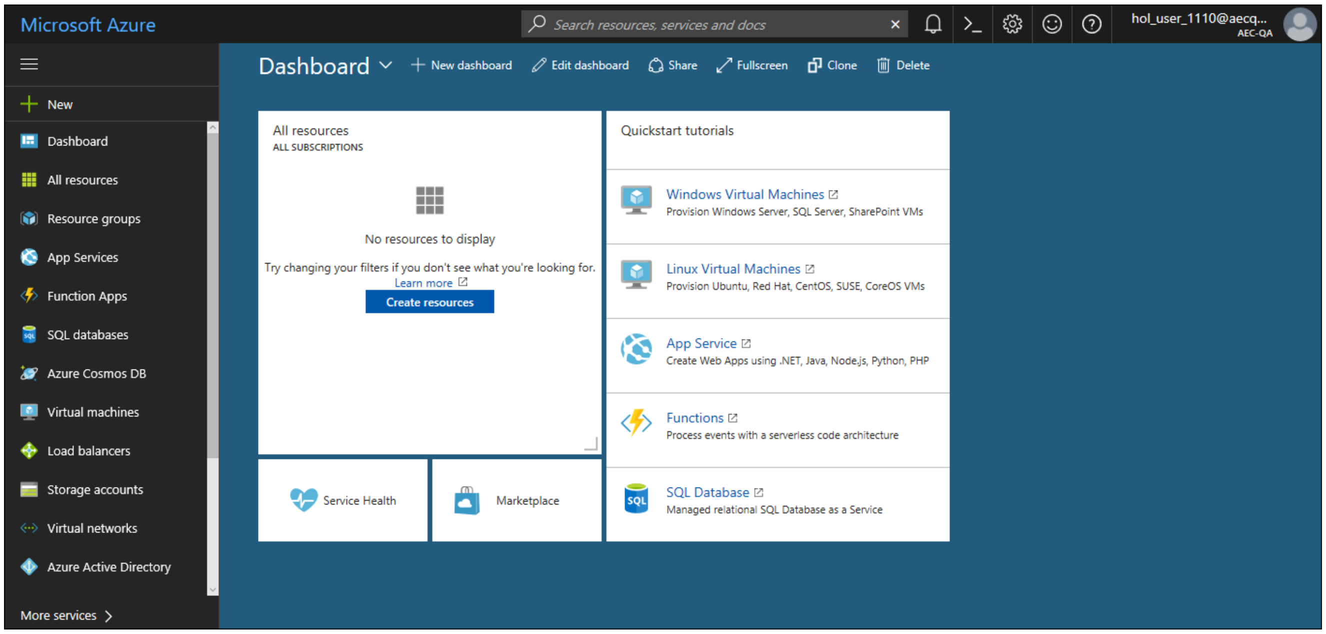
Time Estimate

120 minutes

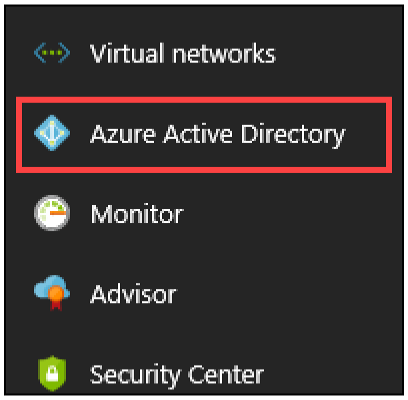
Exercise 1: Create an Azure AD Application

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

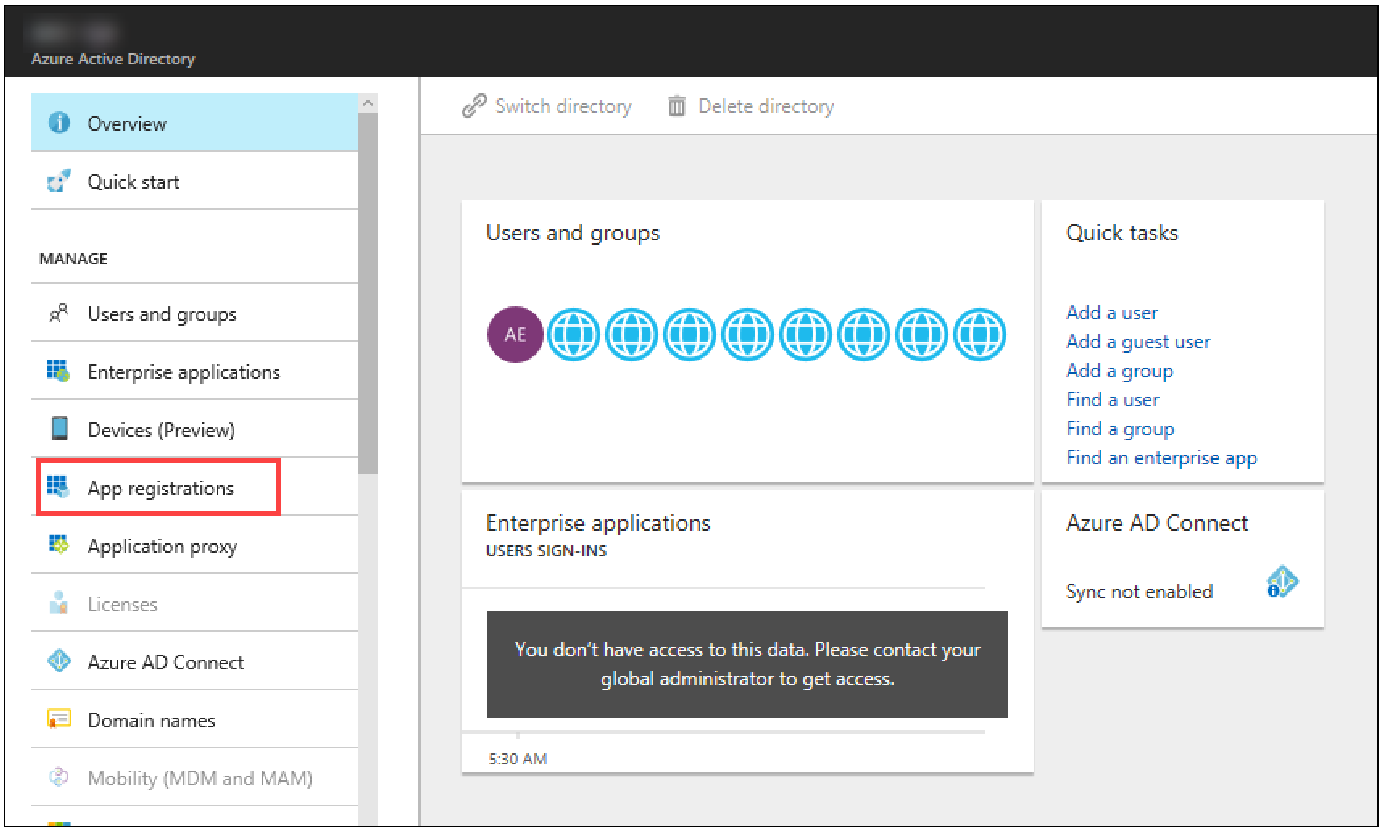
1. **Launch** a browser and **Navigate** to https://portal.azure.com. **Login** with the Microsoft Azure credentials you received via email.



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

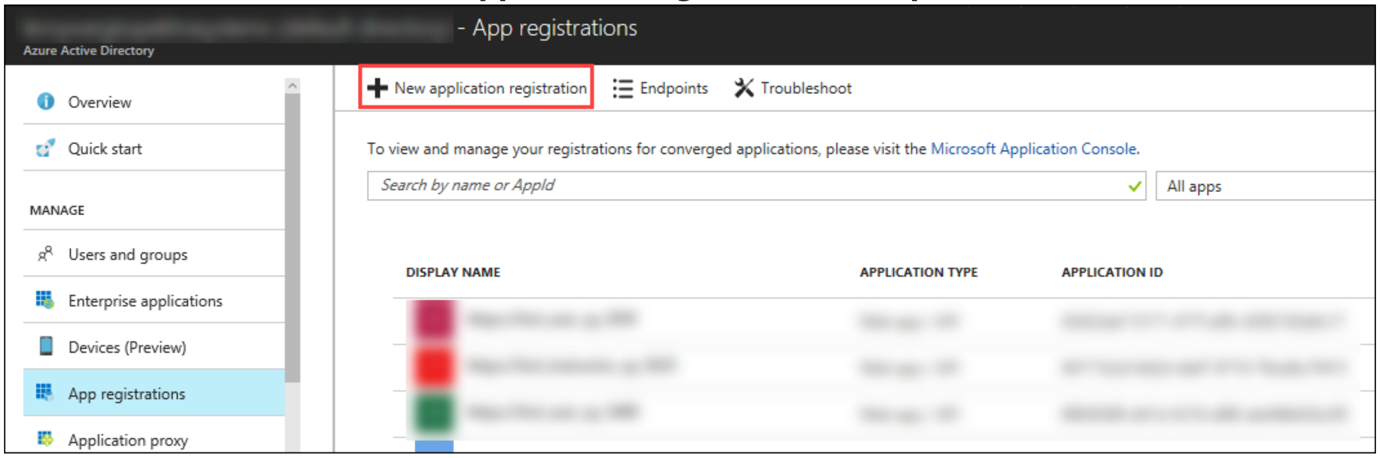


3. You will be directed to the Azure Active Directory blade, **click** on **App registrations.**



You don't have access to this data. Please contact your global administrator to get access.

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



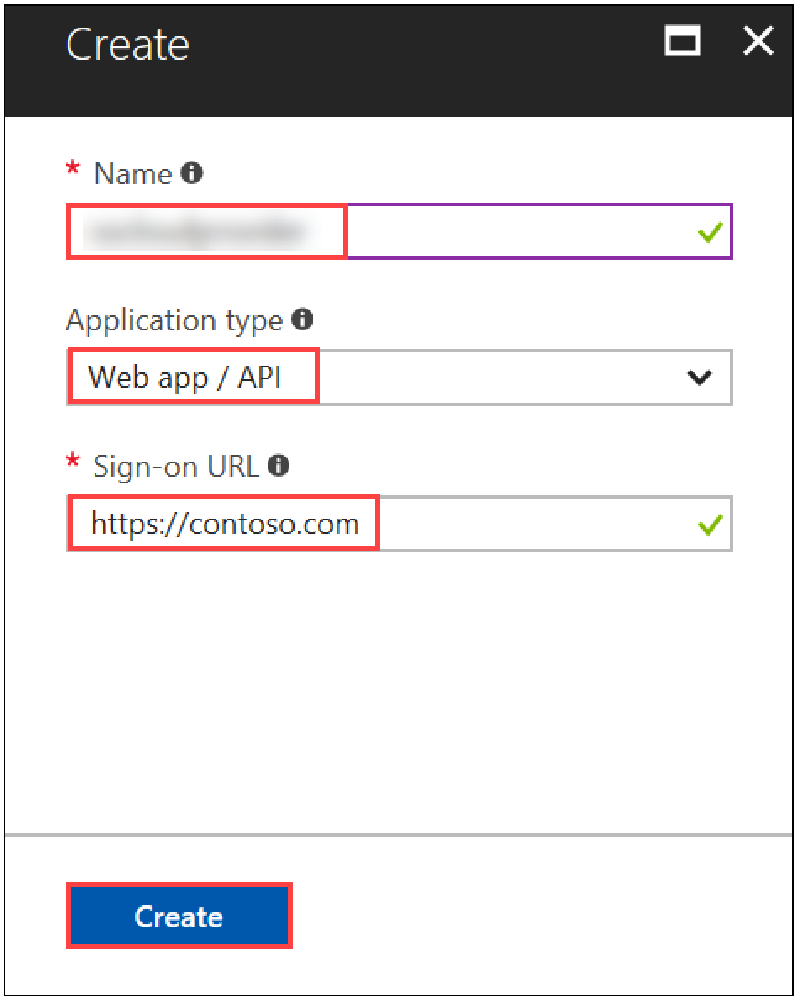
To view and manage your registrations for converged applications, please visit the Microsoft Application Console.

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

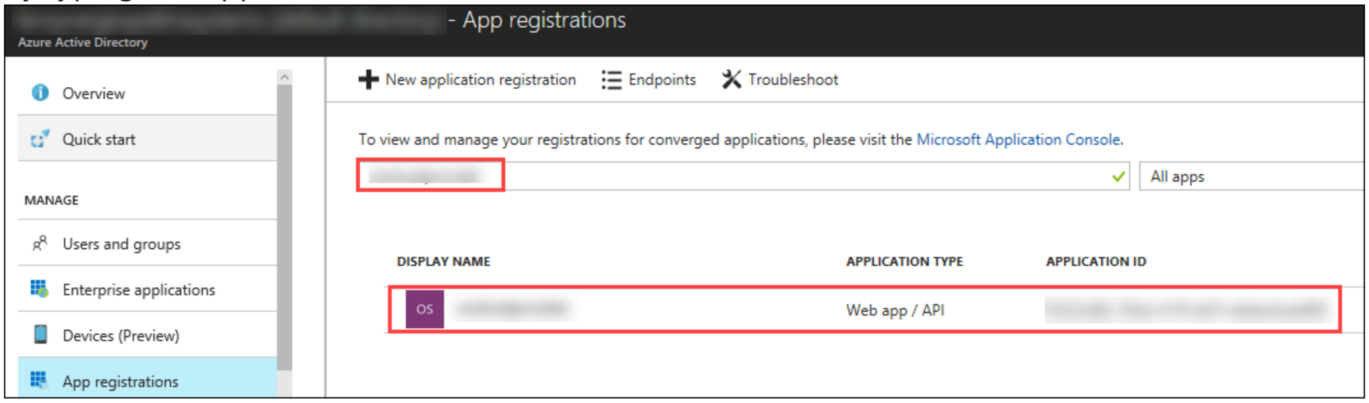
- Name: **(Provide a unique value)**
- Application type: **Web app/API**
- Sign-on URL: https://contoso.com

  Note: We will change this value later during the lab.

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



6. 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.

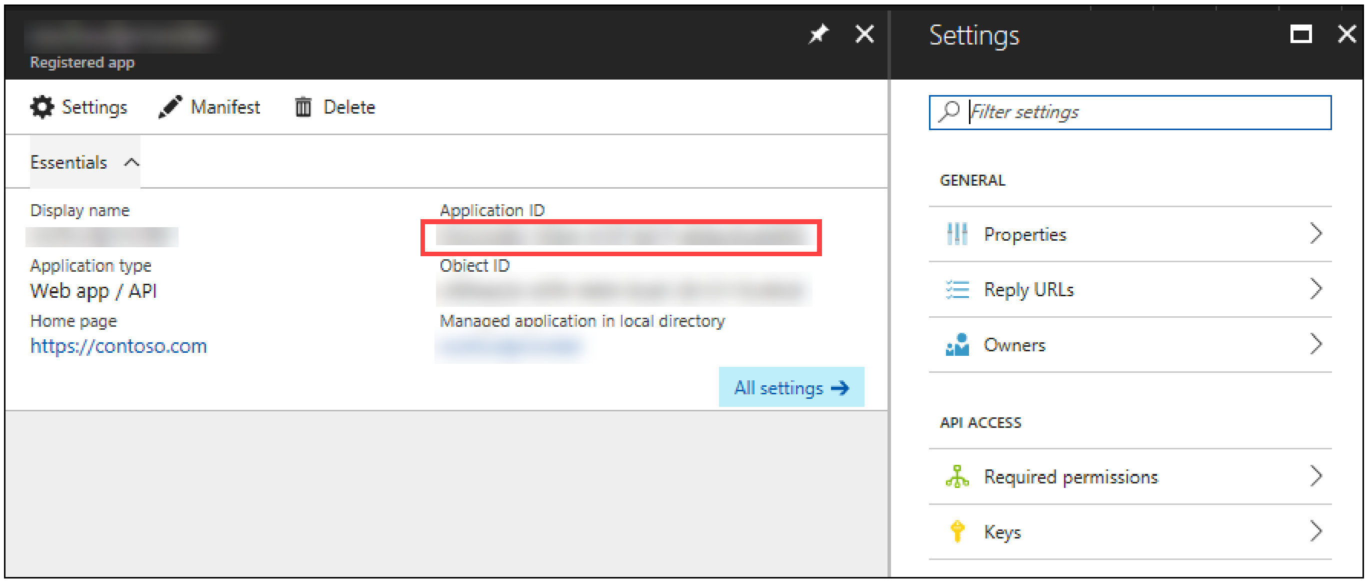


To view and manage your registrations for converged applications, please visit the Microsoft Application Console.

DISPLAY NAME	APPLICATION TYPE	APPLICATION ID
	Web app / API	

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

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


Registered app

Settings | Manifest | Delete

Essentials

Display name

Application type
Web app / API

Home page
https://contoso.com

Application ID

Object ID

Managed application in local directory

All settings →

Settings

Filter settings

GENERAL

Properties

Reply URLs

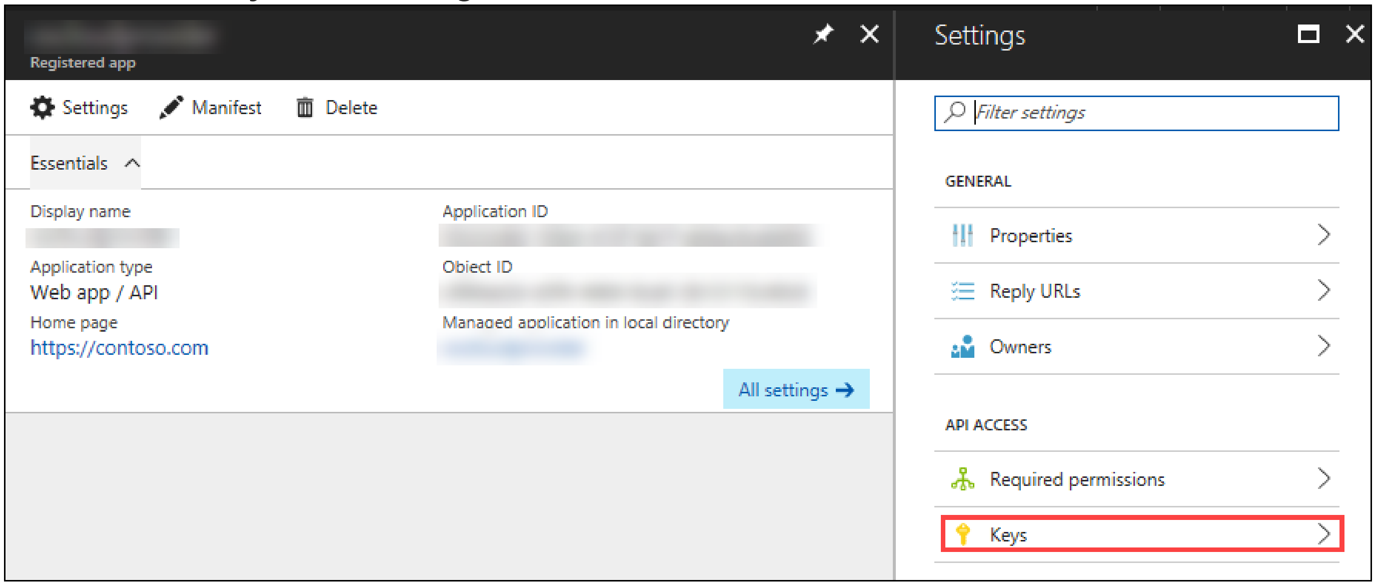
Owners

API ACCESS

Required permissions

Keys

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


Registered app

Settings | Manifest | Delete

Essentials

Display name

Application type
Web app / API

Home page
https://contoso.com

Application ID

Object ID

Managed application in local directory

All settings →

Settings

Filter settings

GENERAL

Properties

Reply URLs

Owners

API ACCESS

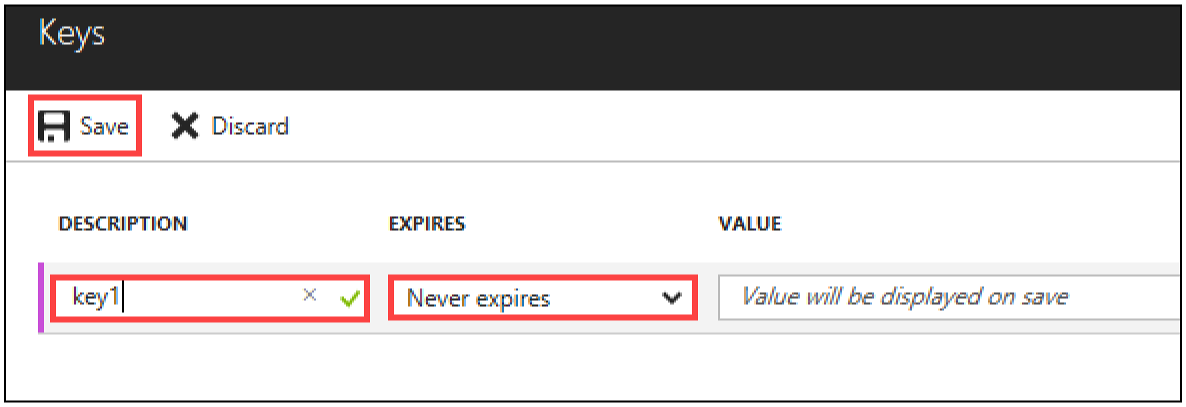
Required permissions

Keys

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

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

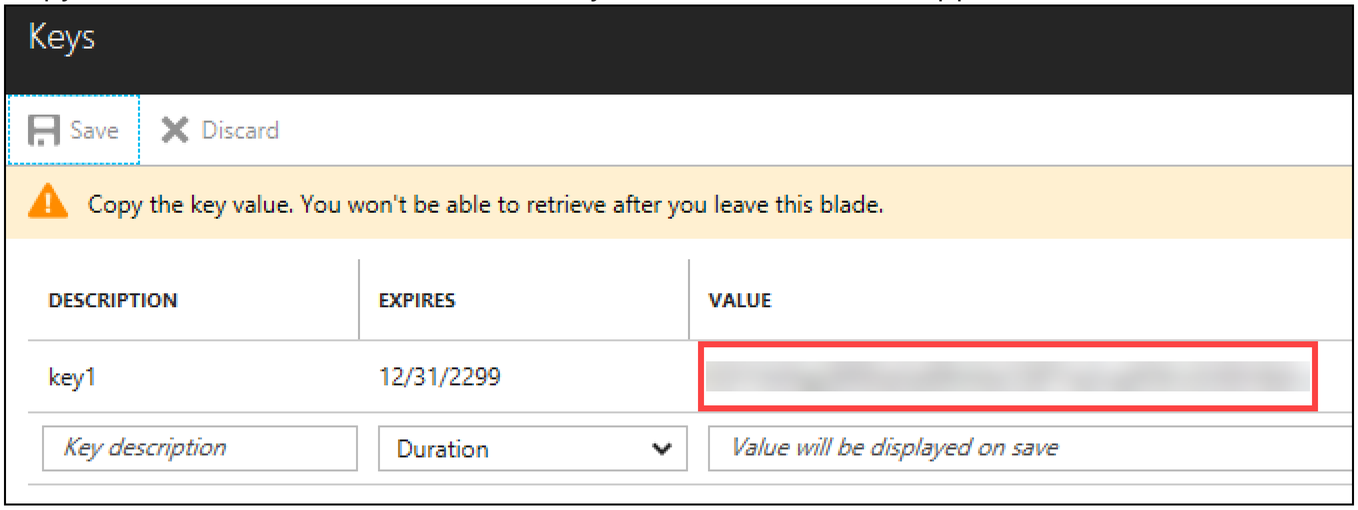
And **Click** on **Save.**


Keys

Save | Discard

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.


Keys

Save | Discard

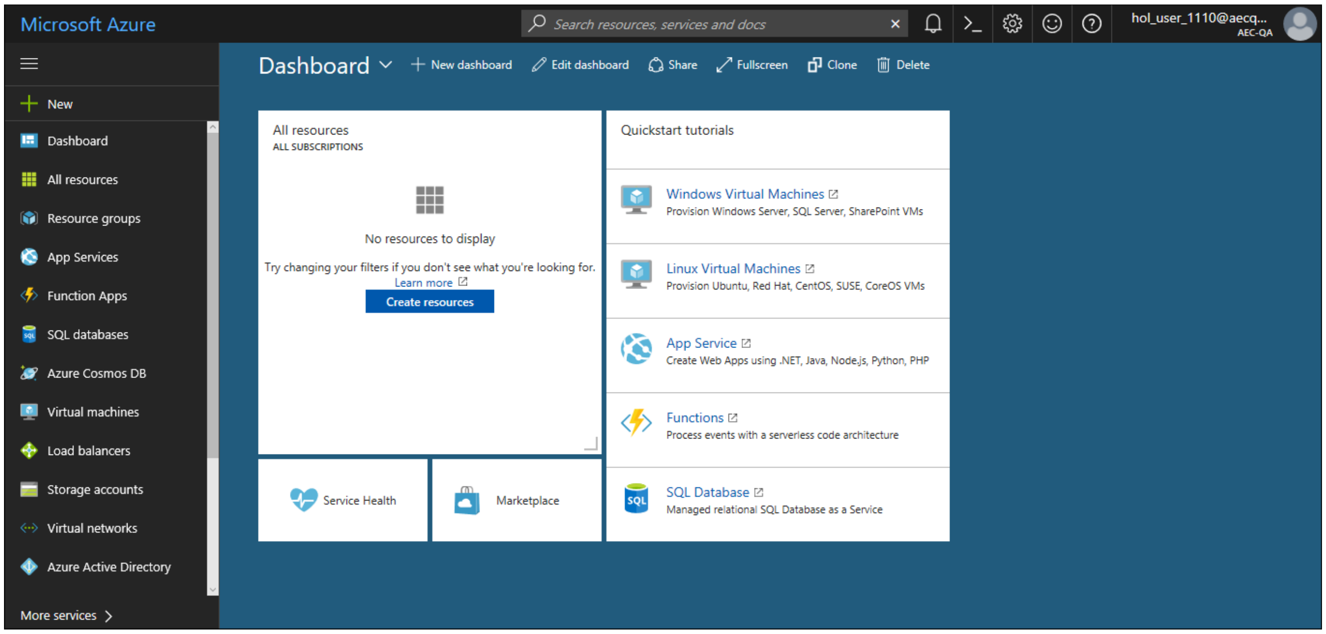
Copy the key value. You won't be able to retrieve after you leave this blade.

DESCRIPTION	EXPIRES	VALUE
key1	12/31/2299	[redacted]
Key description	Duration	Value will be displayed on save

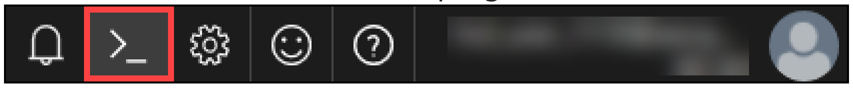
Exercise 2: Create a Keyvault

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.

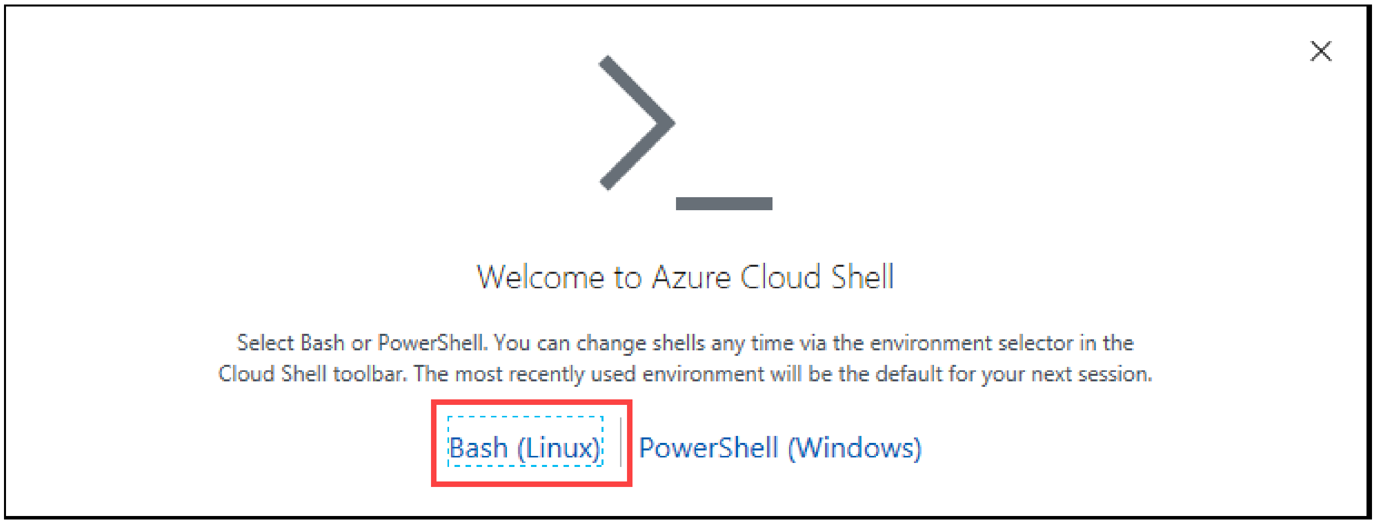
1. **Launch** a browser and **Navigate** to https://portal.azure.com. **Login** with the Microsoft Azure credentials you received via email.



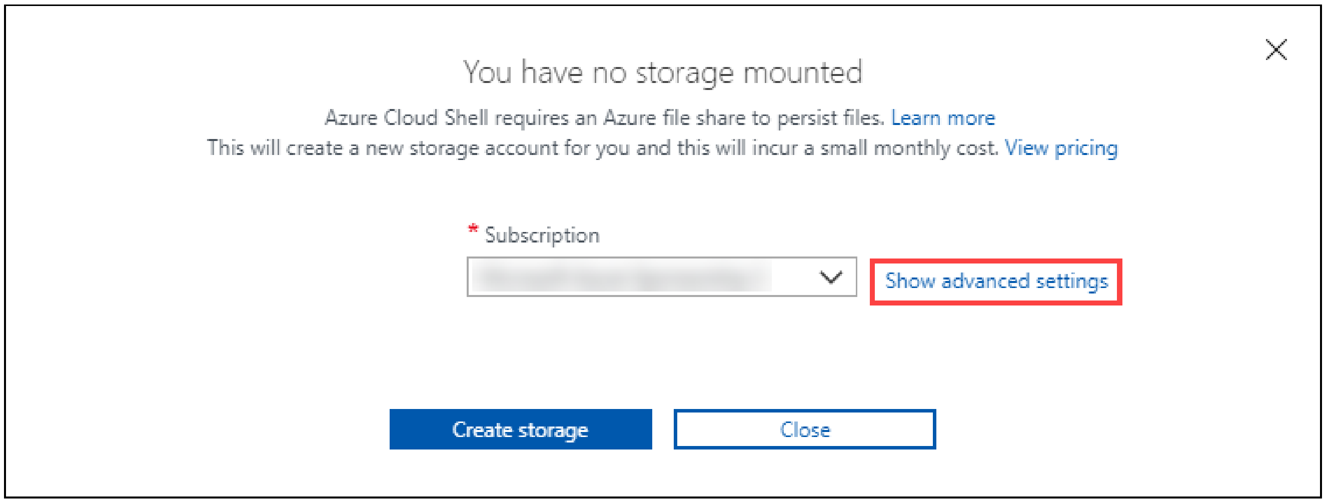
2. **Click** on **Cloud Shell** at the top right corner of the screen, to open the cloud shell.



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

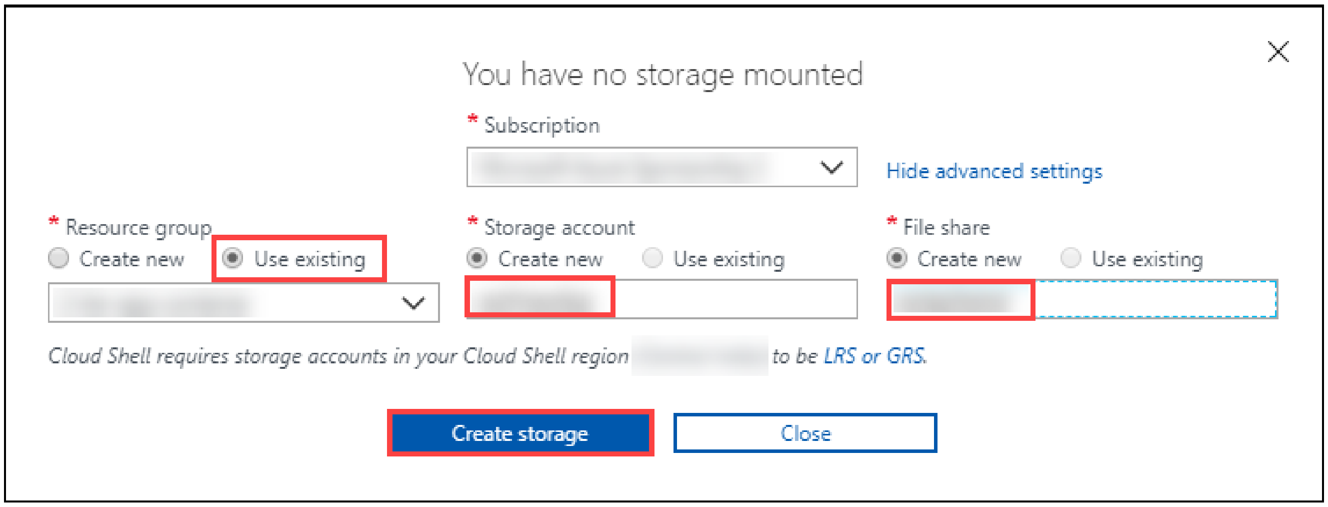

Welcome to Azure Cloud Shell

Select Bash or PowerShell. You can change shells any time via the environment selector in the Cloud Shell toolbar. The most recently used environment will be the default for your next session.


You have no storage mounted

Azure Cloud Shell requires an Azure file share to persist files. Learn more
This will create a new storage account for you and this will incur a small monthly cost. View pricing

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.**

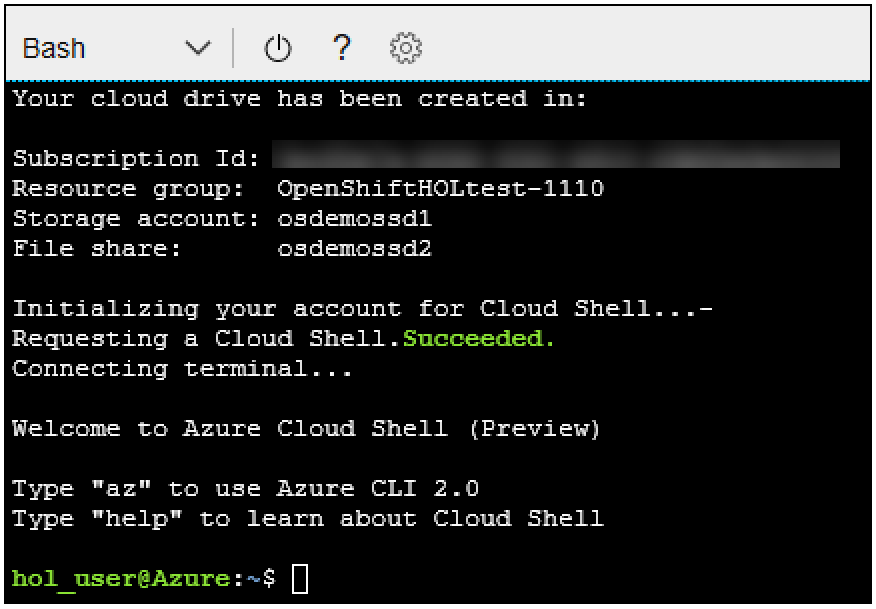

You have no storage mounted

* Subscription

* Resource group: Create new [✓ Use existing]
* Storage account: [✓ Create new] Use existing
* File share: [✓ Create new] Use existing

Cloud Shell requires storage accounts in your Cloud Shell region to be LRS or GRS.

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

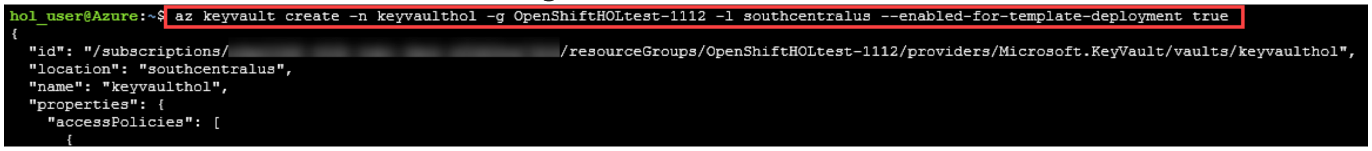


```
Your cloud drive has been created in:

Subscription Id: [redacted]
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

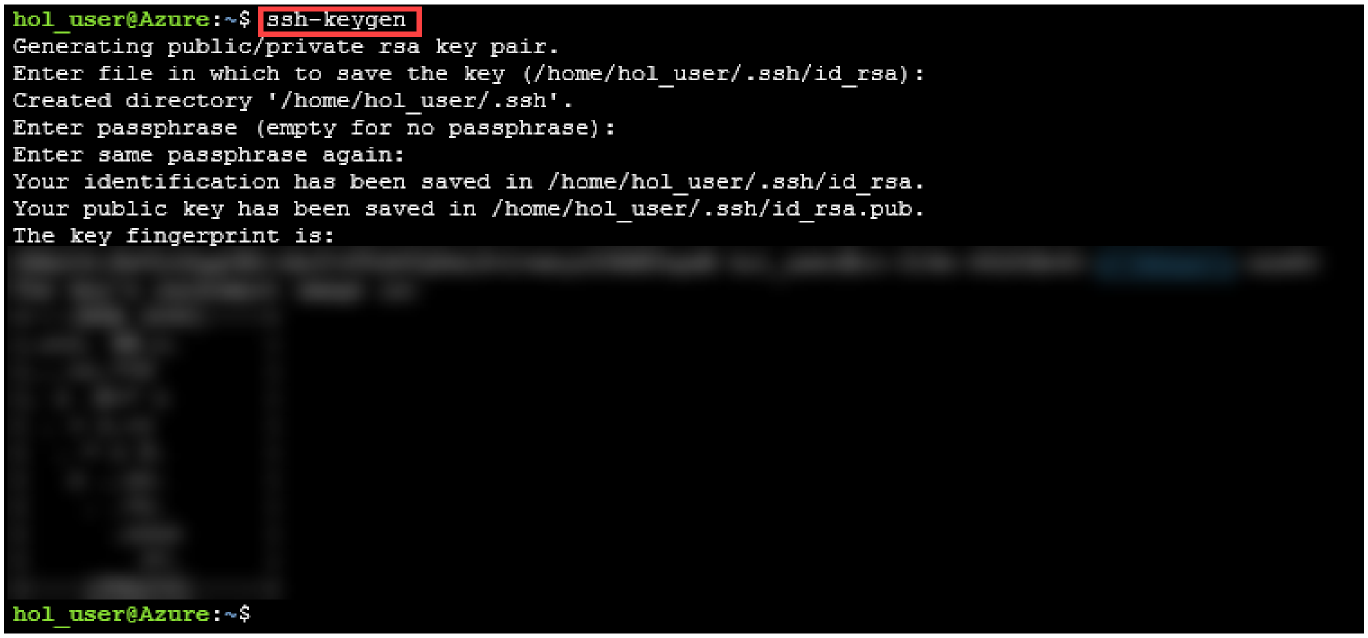


```
hol_user@Azure:~$ az keyvault create -n keyvaulthol -g OpenShiftHOLtest-1112 -l southcentralus --enabled-for-template-deployment true
{
  "id": "/subscriptions/[redacted]/resourceGroups/OpenShiftHOLtest-1112/providers/Microsoft.KeyVault/vaults/keyvaulthol",
  "location": "southcentralus",
  "name":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

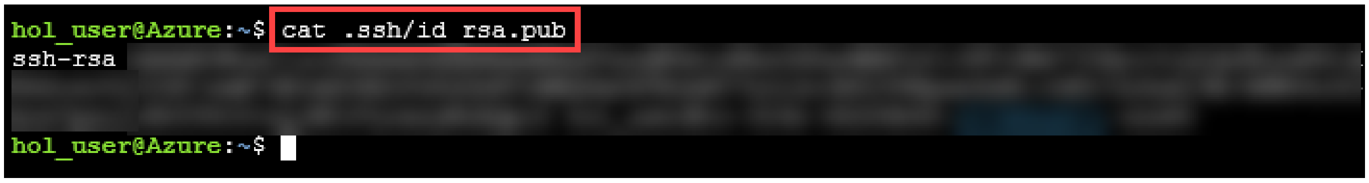


```
hol_user@Azure:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hol_user/.ssh/id_rsa):
Created directory '/home/hol_user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hol_user/.ssh/id_rsa.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

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.

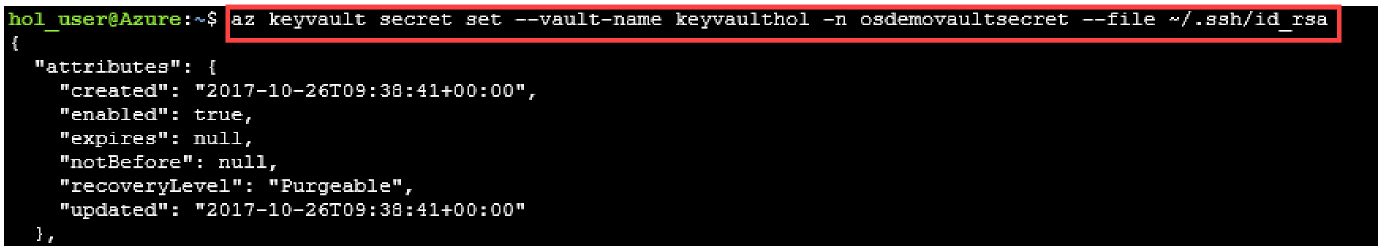


```
hol_user@Azure:~$ cat .ssh/id_rsa.pub
ssh-rsa [redacted]
hol_user@Azure:~$
```

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.



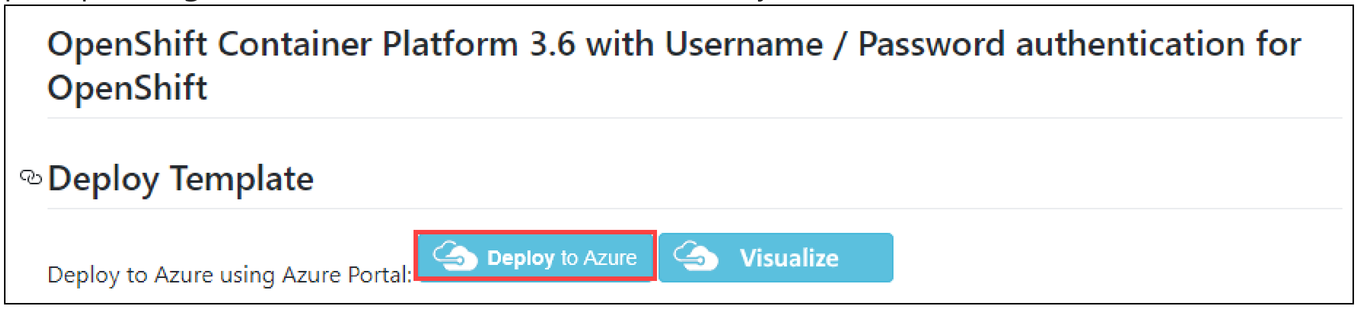
```
hol_user@Azure:~$ az keyvault secret set --vault-name keyvaulthol -n osdemovaultsecret --file ~/.ssh/id_rsa
{
  "attributes": {
    "created": "2017-10-26T09:38:41+00:00",
    "enabled": true,
    "expires": null,
    "notBefore": null,
    "recoveryLevel": "Purgeable",
    "updated": "2017-10-26T09:38:41+00:00"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

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.

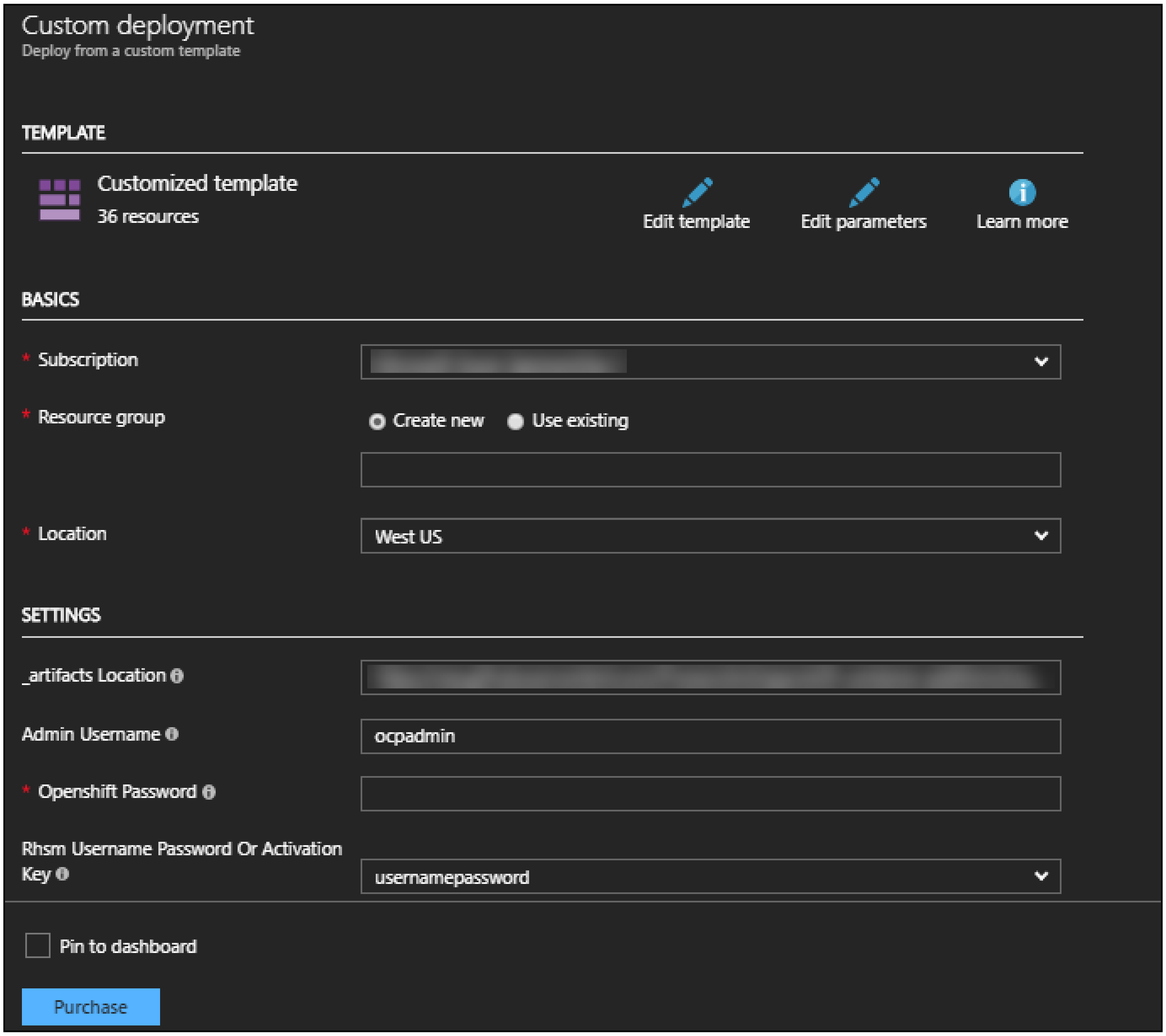


OpenShift Container Platform 3.6 with Username / Password authentication for OpenShift

Deploy Template

Deploy to Azure using Azure Portal:

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



Custom deployment
Deploy from a custom template

TEMPLATE

Customized template
36 resources

BASICS

* Subscription	[redacted]
* Resource group	Create new Use existing
* Location	West US

SETTINGS

_artifacts Location	[redacted]
Admin Username	ocpadmin
* Openshift Password	
Rhsm Username Password Or Activation Key	usernamepassword

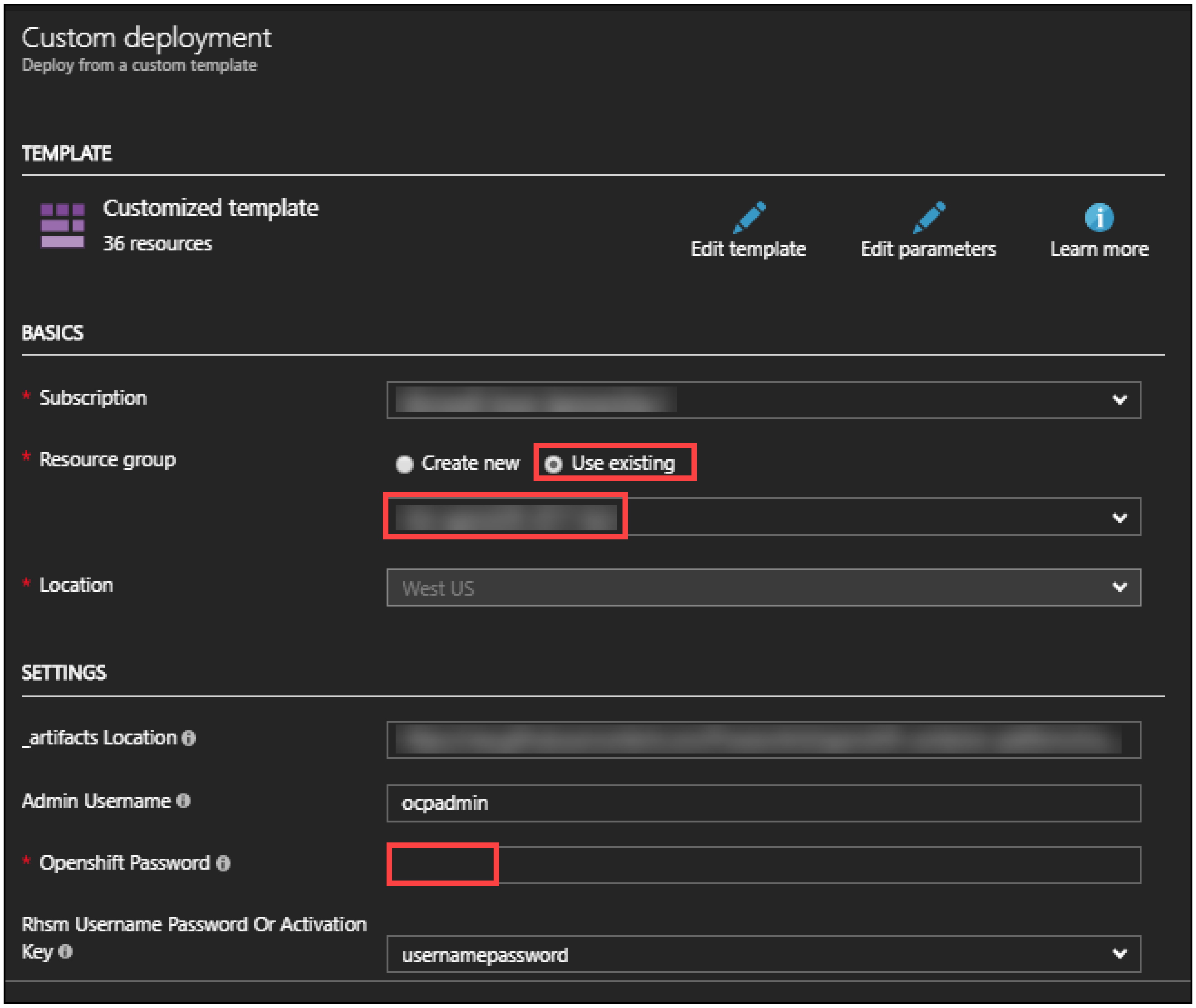
4. In the **Custom Deployment** blade, **configure** the settings as follows:

- 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

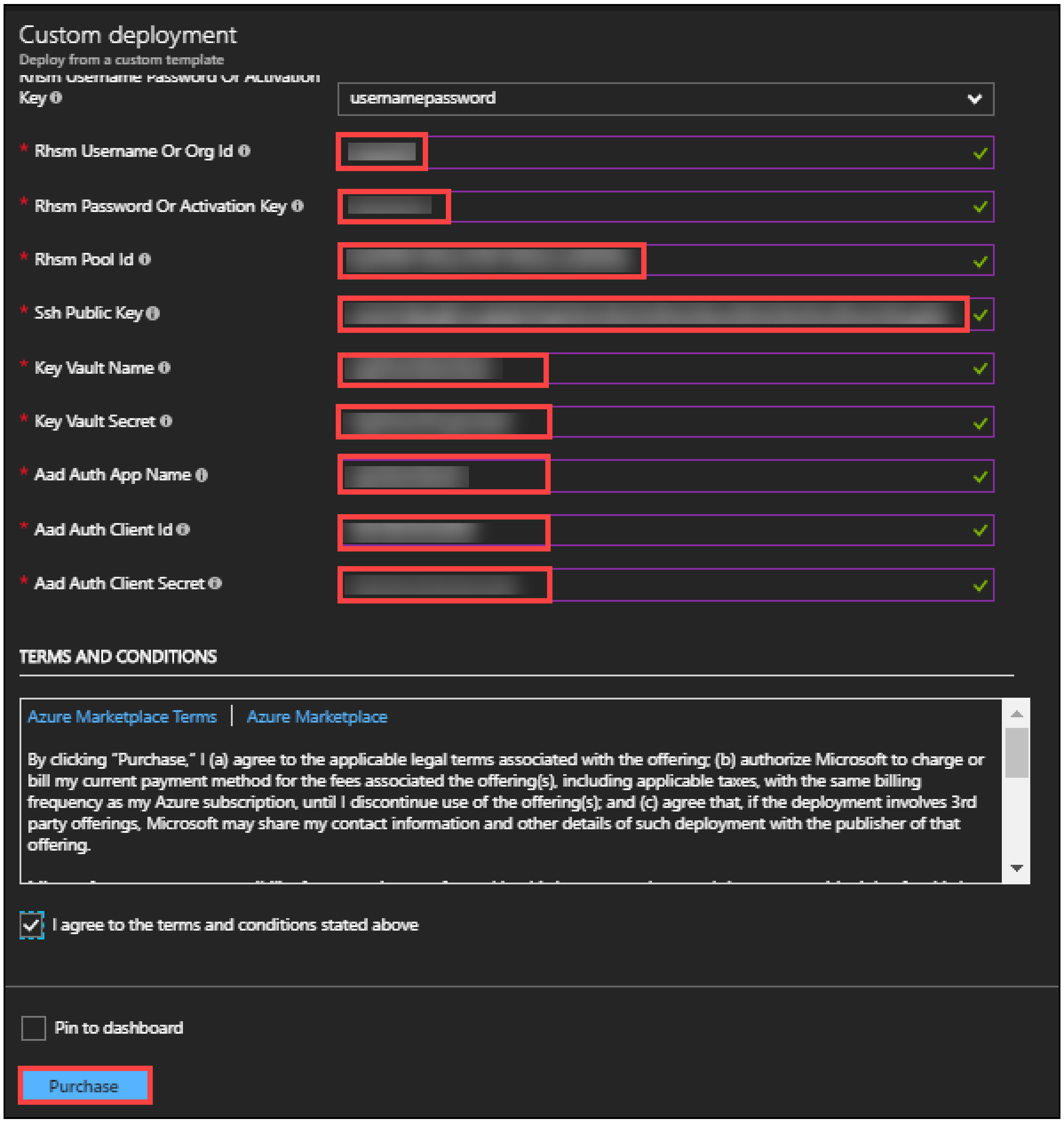
Customized template
36 resources

BASICS

* Subscription	[redacted]
* Resource group	Create new [✓ Use existing]
	[redacted]
* Location	West US

SETTINGS

_artifacts Location	[redacted]
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: usernamepassword

* Rhsm Username Or Org Id: [redacted]
* Rhsm Password Or Activation Key: [redacted]
* Rhsm Pool Id: [redacted]
* Ssh Public Key: [redacted]
* Key Vault Name: [redacted]
* Key Vault Secret: [redacted]
* Aad Auth App Name: [redacted]
* Aad Auth Client Id: [redacted]
* Aad Auth Client Secret: [redacted]

TERMS AND CONDITIONS

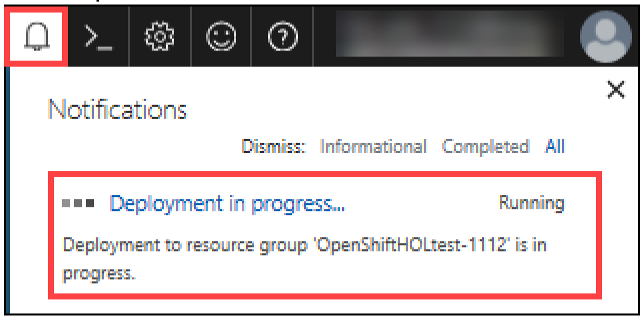
Azure Marketplace Terms | Azure Marketplace

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

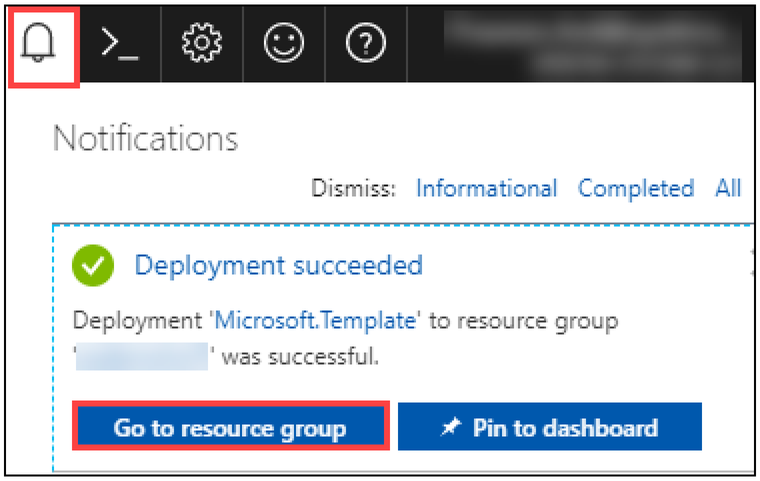
[ ] Pin to dashboard

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



Deployment in progress... Running
Deployment to resource group 'OpenShiftHOLtest-1112' is in progress.

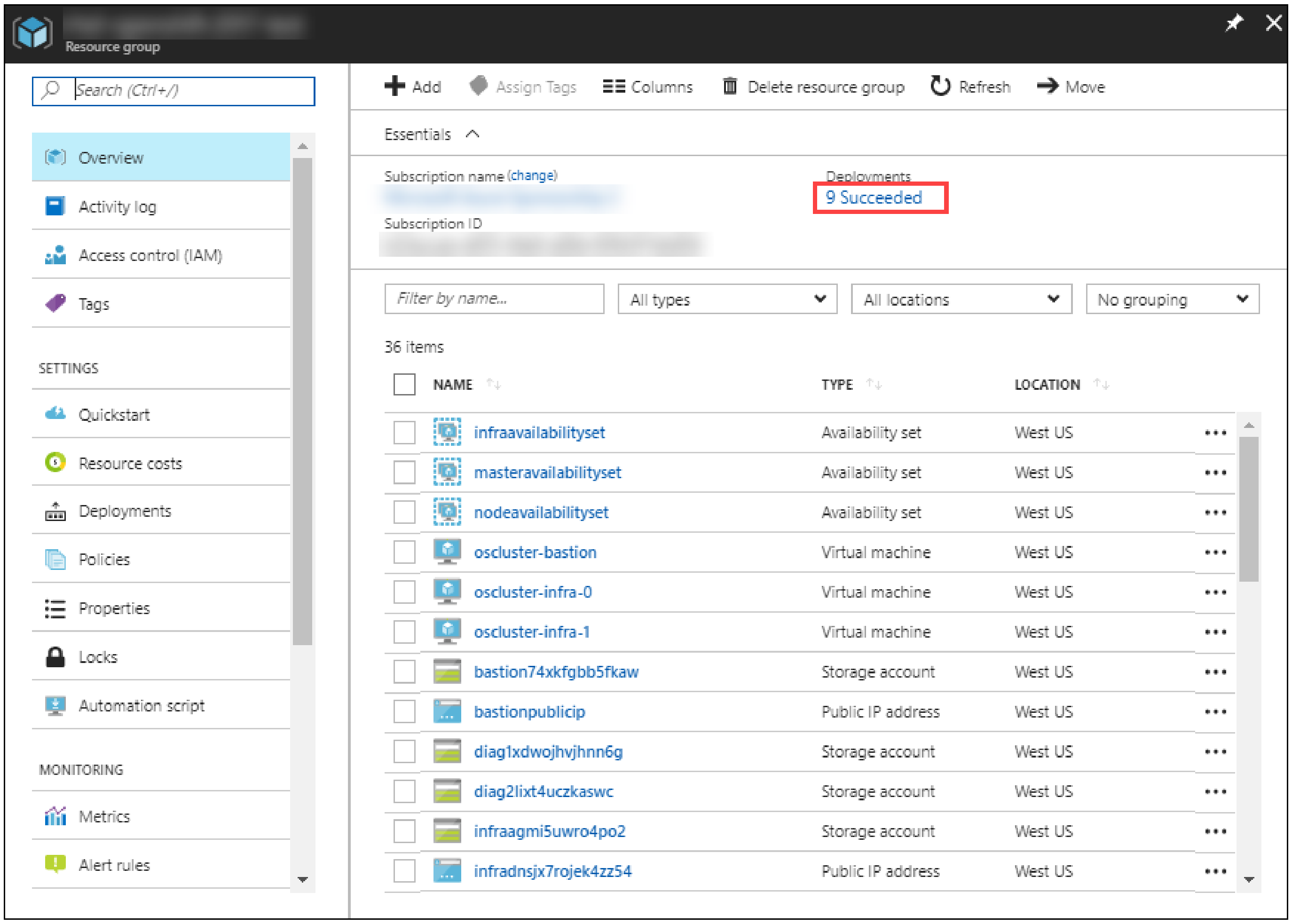
7. Once the deployment is complete, you can see it in the notifications tab as Deployment succeeded. Now, **click** on **Go to resource group** from the notifications tab.



Deployment succeeded

Deployment 'Microsoft.Template' to resource group ' ' was successful.

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



Subscription name (change)

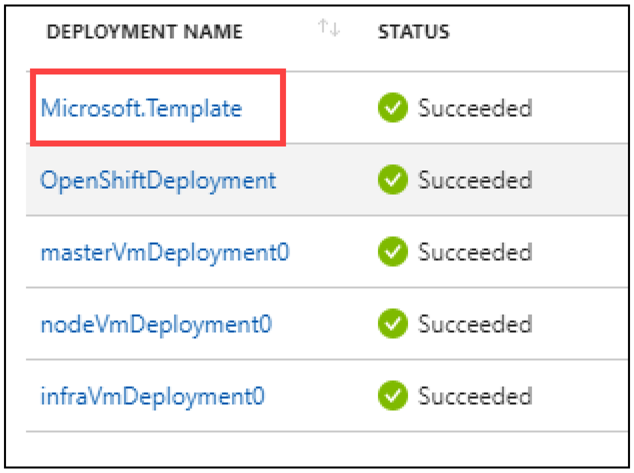
Subscription ID

Deployments
9 Succeeded

36 items

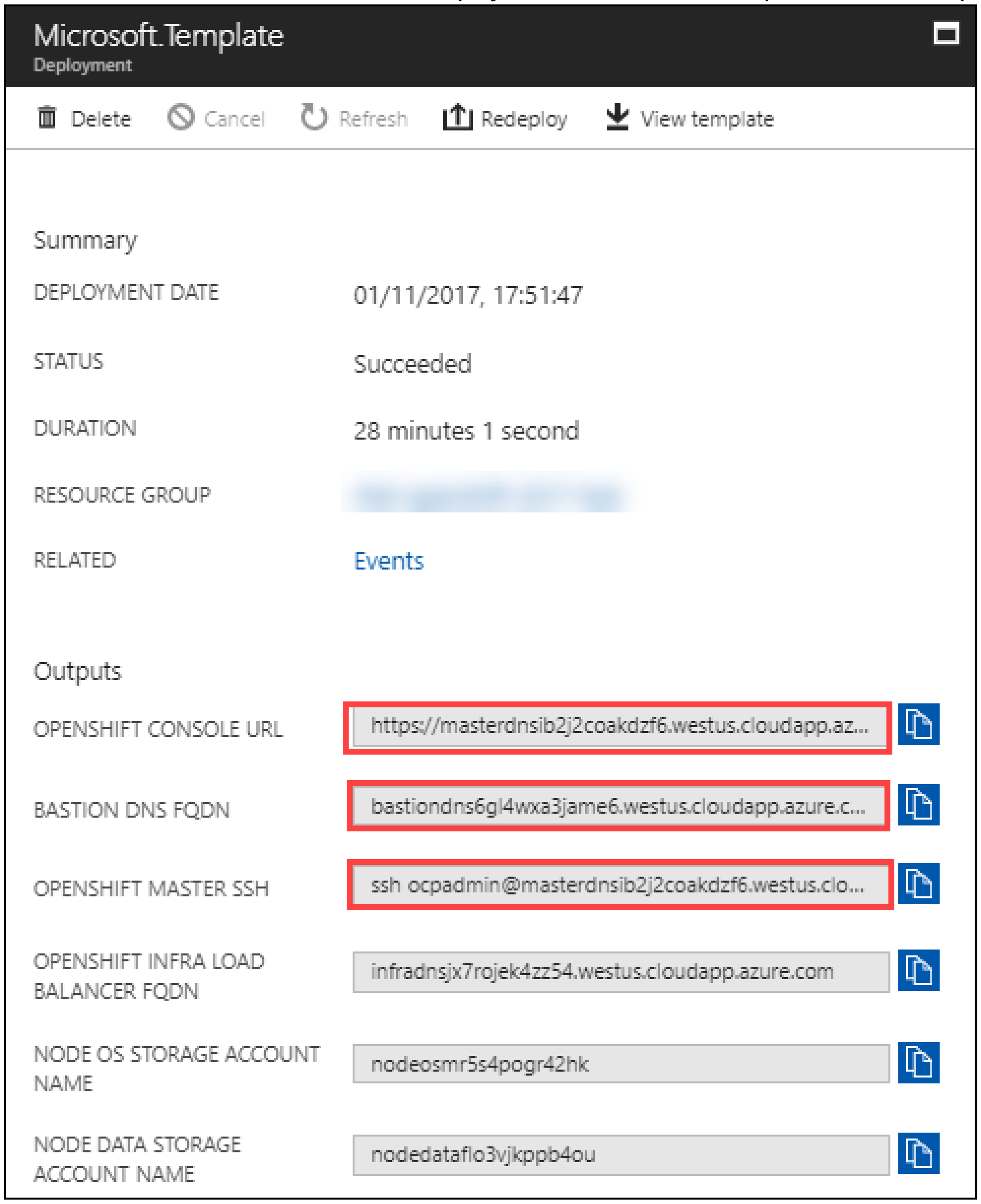
NAME	TYPE	LOCATION
infraavailabilityset	Availability set	West US
masteravailabilityset	Availability set	West US
nodeavailabilityset	Availability set	West US
oscluster-bastion	Virtual machine	West US
oscluster-infra-0	Virtual machine	West US
oscluster-infra-1	Virtual machine	West US
bastion74xkfgbb5fkaw	Storage account	West US
bastionpublicip	Public IP address	West US
diag1xdwojhvjhnn6g	Storage account	West US
diag2lixt4uczkaswc	Storage account	West US
infraagmi5uwro4po2	Storage account	West US
infradnsjx7rojek4zz54	Public IP address	West US

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

Summary

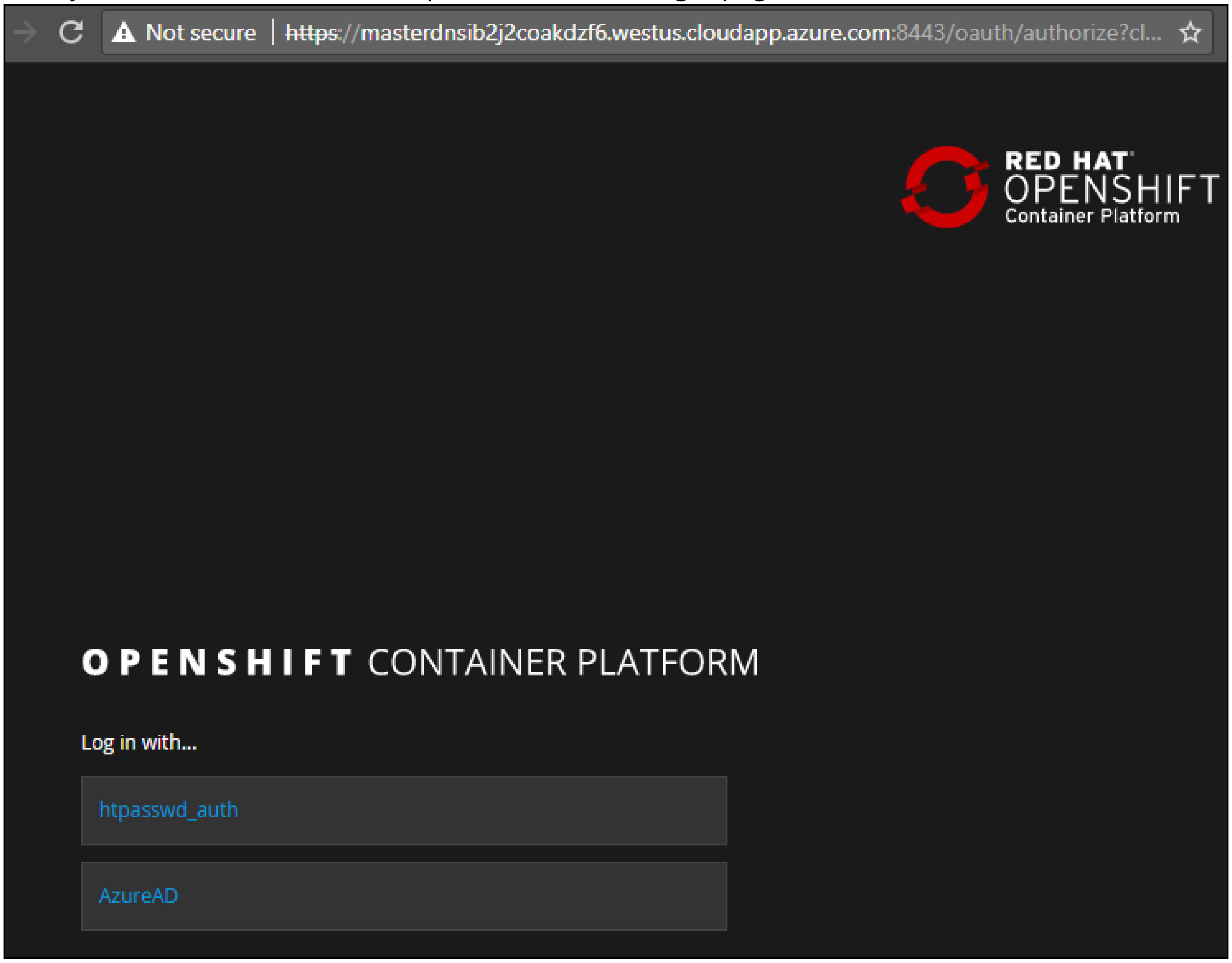
DEPLOYMENT DATE	01/11/2017, 17:51:47
STATUS	Succeeded
DURATION	28 minutes 1 second
RESOURCE GROUP	[illegible]
RELATED	Events

Outputs

OPENSHIFT CONSOLE URL	https://masterdnsib2j2coakdzf6.westus.cloudapp.az...
BASTION DNS FQDN	bastiondns6gl4wxa3jame6.westus.cloudapp.azure.c...
OPENSHIFT MASTER SSH	ssh ocpadmin@masterdnsib2j2coakdzf6.westus.clo...
OPENSHIFT INFRA LOAD BALANCER FQDN	infradnsjx7rojek4zz54.westus.cloudapp.azure.com
NODE OS STORAGE ACCOUNT NAME	nodeosmr5s4pogr42hk
NODE DATA STORAGE ACCOUNT NAME	nodedataflo3vjkppb4ou

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.

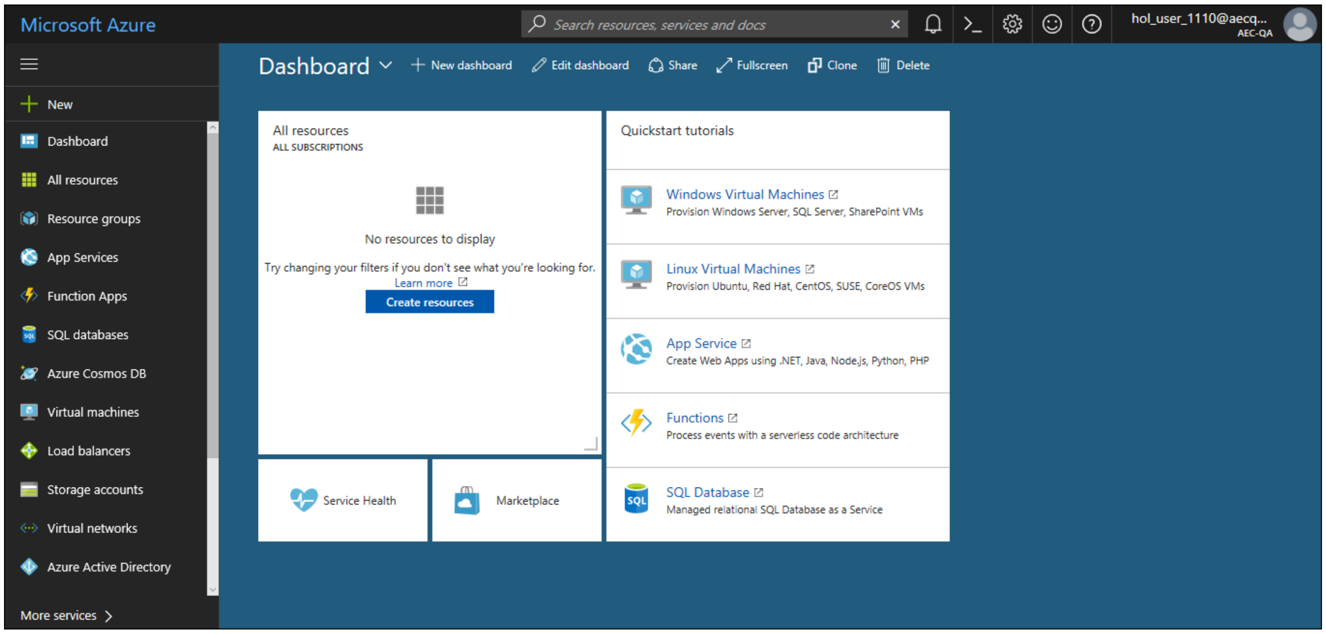

Not secure | https://masterdnsib2j2coakdzf6.westus.cloudapp.azure.com:8443/oauth/authorize?cl...

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.



No resources to display

Try changing your filters if you don't see what you're looking for.

Windows Virtual Machines
Provision Windows Server, SQL Server, SharePoint VMs

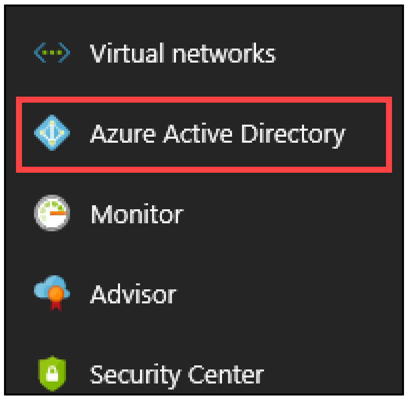
Linux Virtual Machines
Provision Ubuntu, Red Hat, CentOS, SUSE, CoreOS VMs

App Service
Create Web Apps using .NET, Java, Node.js, Python, PHP

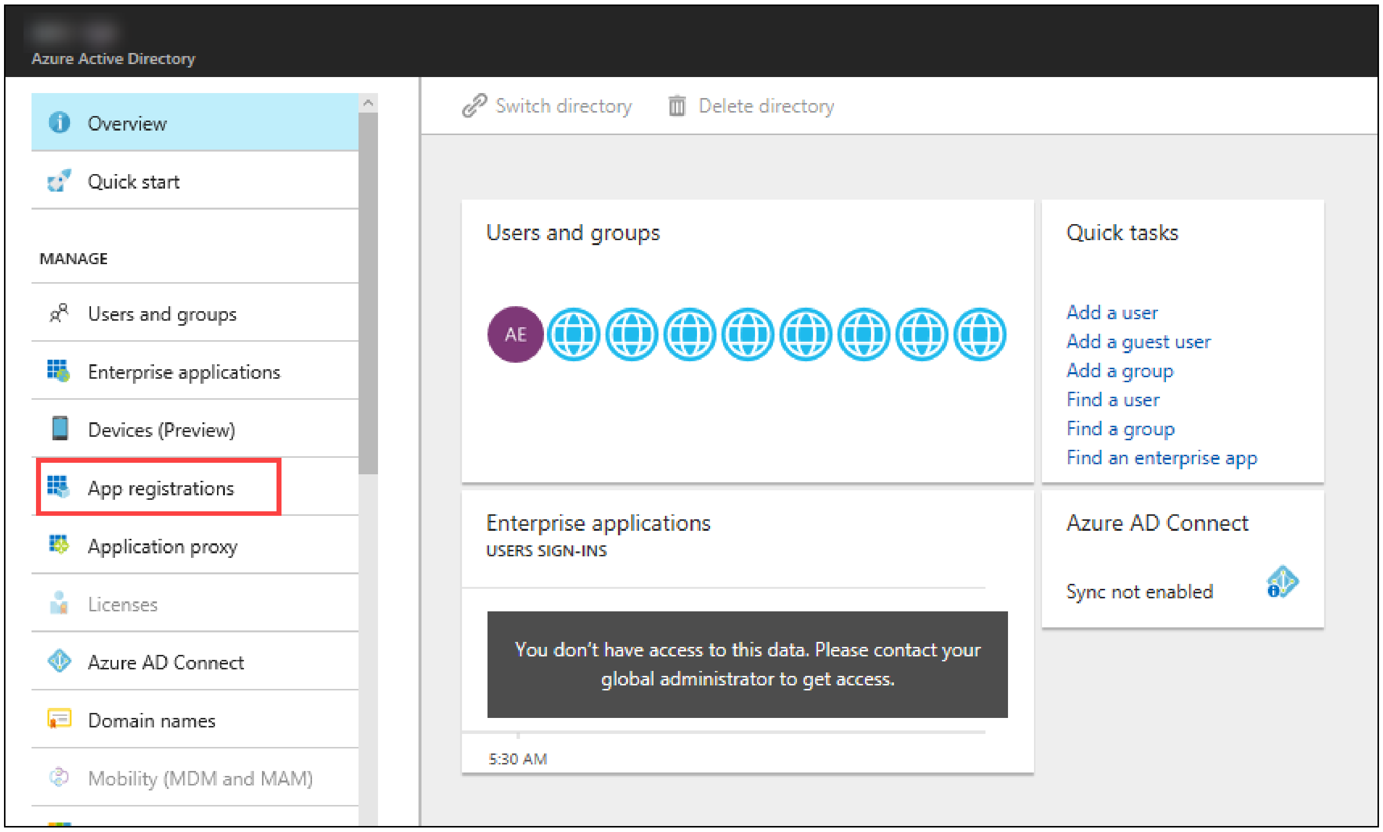
Functions
Process events with a serverless code architecture

SQL Database
Managed relational SQL Database as a Service

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

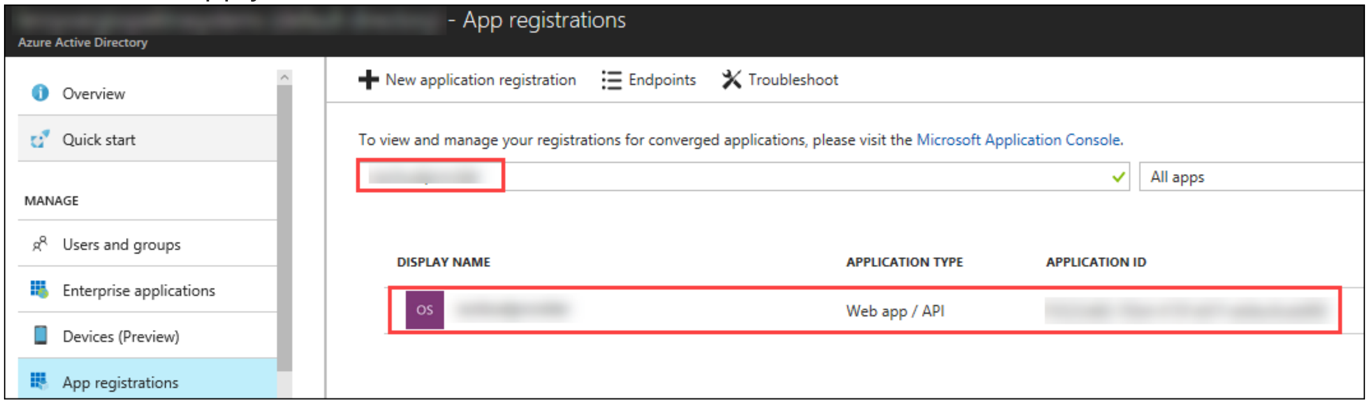


3. You will be directed to the Azure Active Directory blade, **click** on **App registrations.**



You don't have access to this data. Please contact your global administrator to get access.

4. 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.

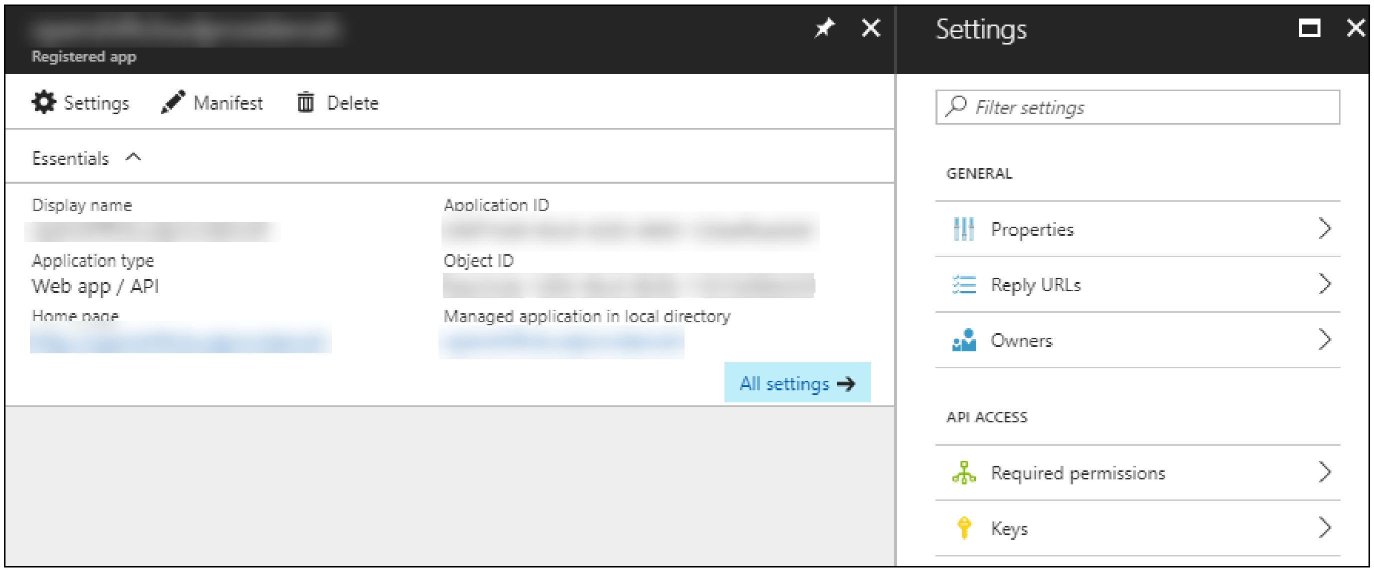


To view and manage your registrations for converged applications, please visit the Microsoft Application Console.

[redacted]

DISPLAY NAME	APPLICATION TYPE	APPLICATION ID
OS [redacted]	Web app / API	[redacted]

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



Display name

Application type
Web app / API

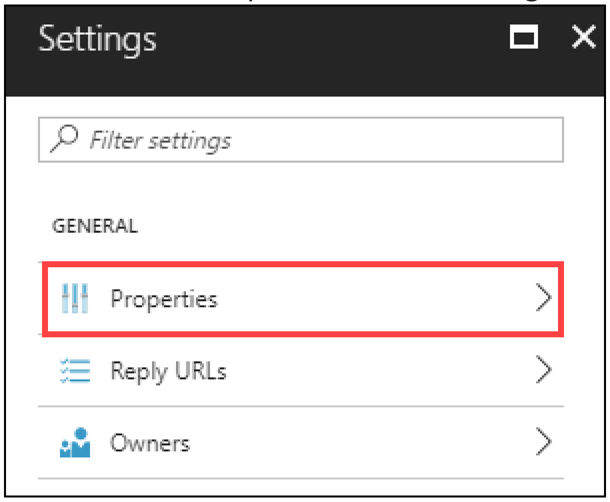
Home page

Application ID

Object ID

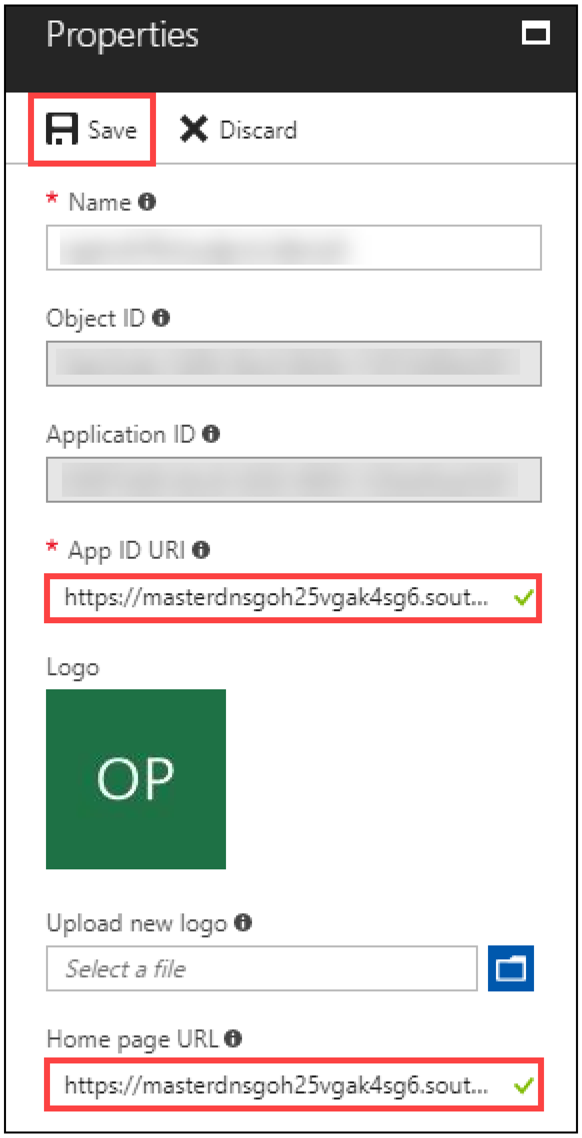
Managed application in local directory

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

* Name

Object ID

Application ID

* App ID URI

https://masterdnsgoh25vgak4sg6.sout...

Logo

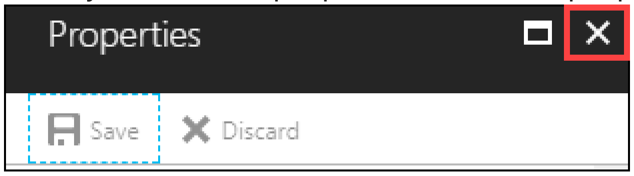
Upload new logo

Select a file

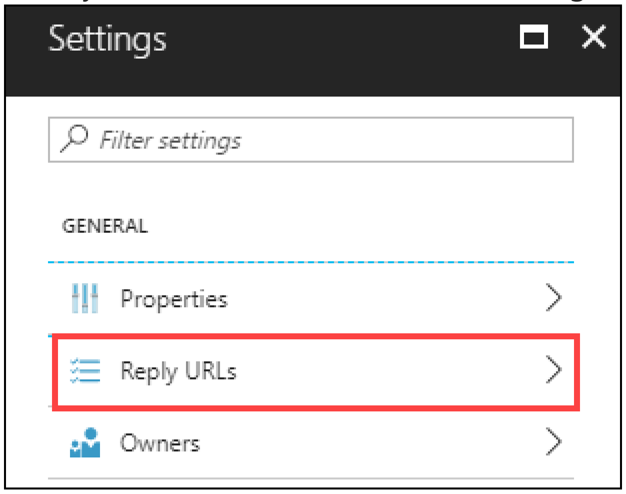
Home page URL

https://masterdnsgoh25vgak4sg6.sout...

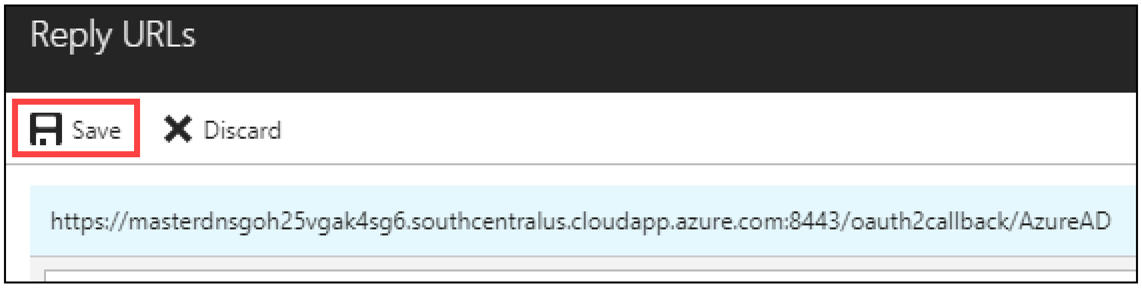
8. Once you save the properties, close the properties blade.



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

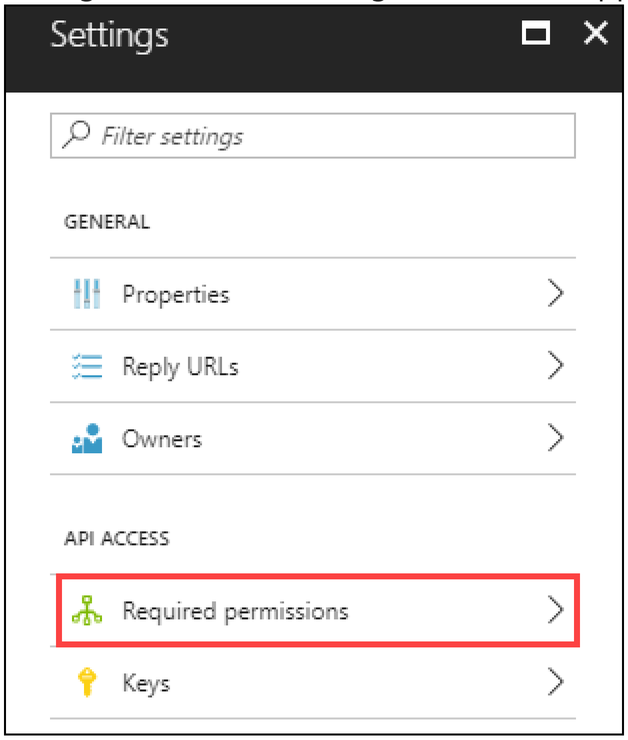


10. Now modify the openshift console url by removing the 'console' from the end and appending 'oauth2callback/AzureAD' to the url and provide it in the Reply URL blade that come up and then Click on Save.

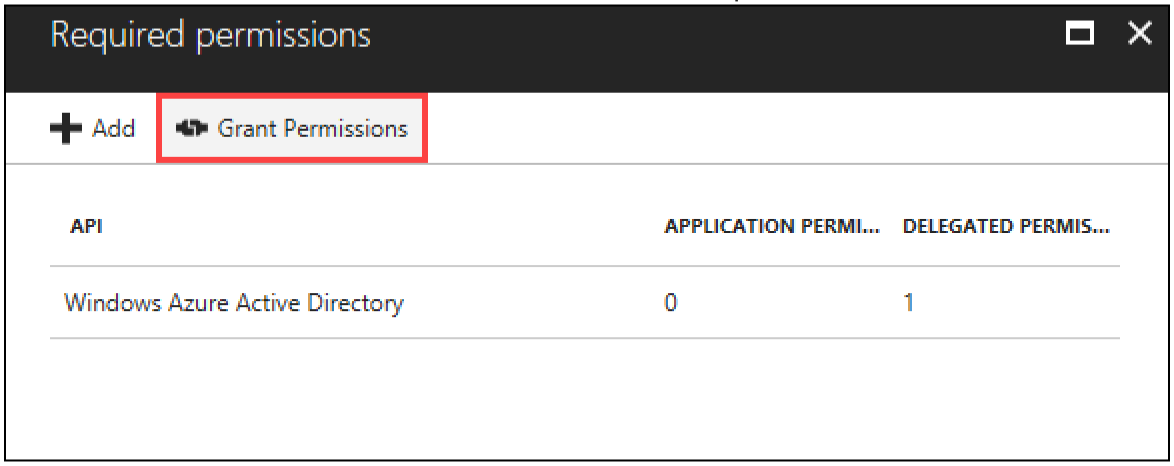


https://masterdnsgoh25vgak4sg6.southcentralus.cloudapp.azure.com:8443/oauth2callback/AzureAD

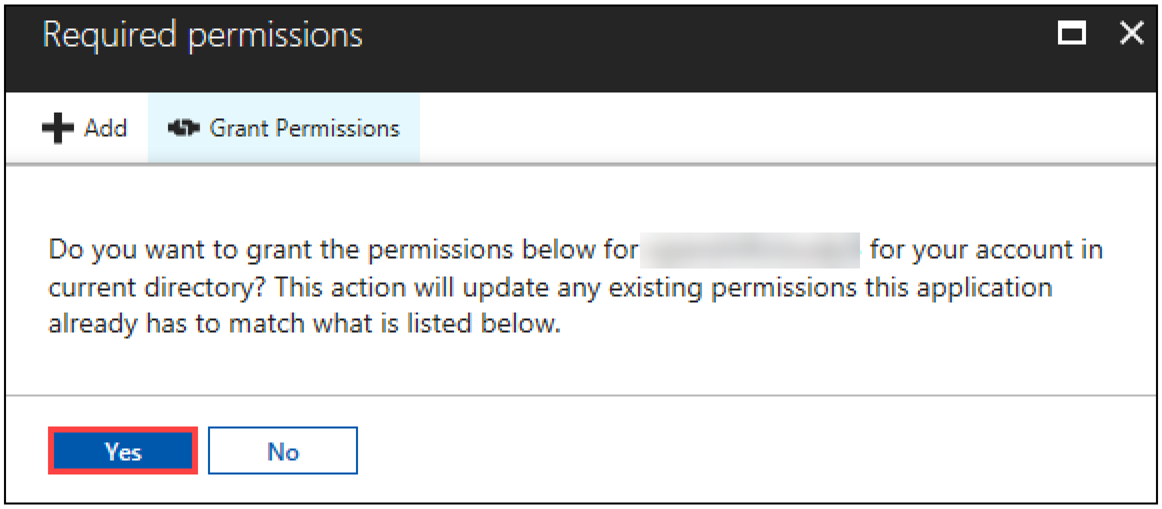
11. Now go back to the setting blade of the App and Click on Required permissions.



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



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

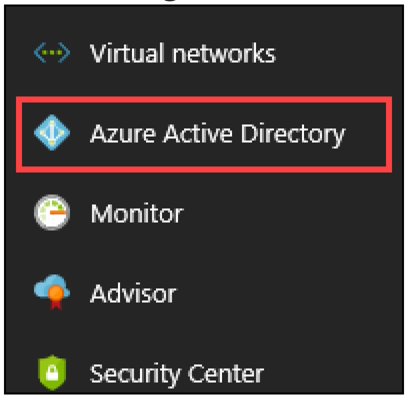

Required permissions

Add | Grant Permissions

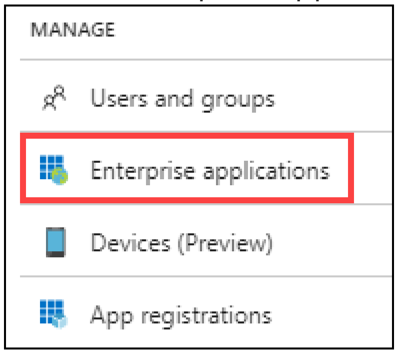
Do you want to grant the permissions below for [redacted] for your account in current directory? This action will update any existing permissions this application already has to match what is listed below.

Yes | No

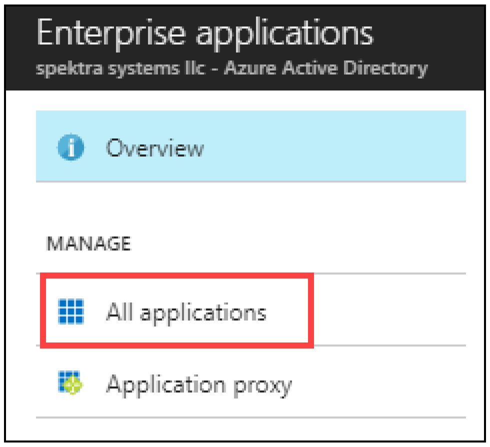
13. Now go back to the Active Directory blade by clicking on **Azure Active Directory** button in the **Menu navigation** bar.



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



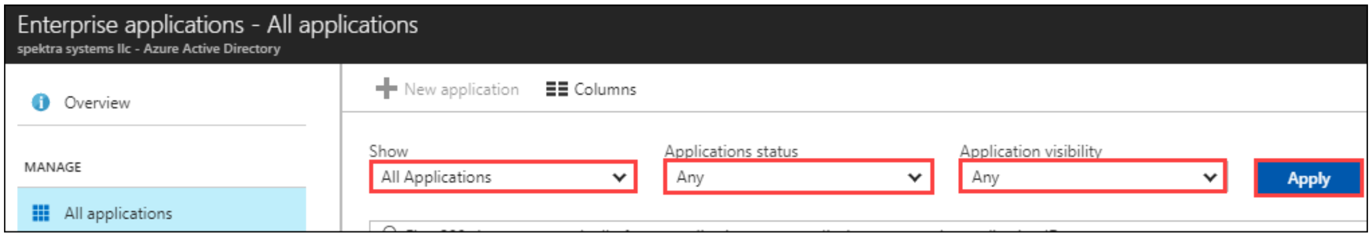
15. In the new blade that come up, click on **All applications**



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

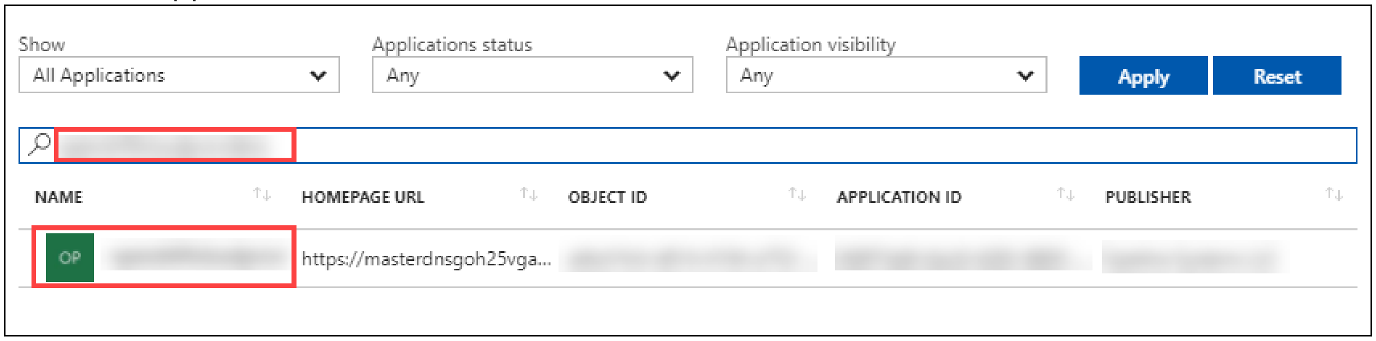
- Show: Select All Applications
- Application status: Any
- Application visibility: Any

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


Enterprise applications - All applications
spektra systems llc - Azure Active Directory
Overview
MANAGE
All applications
New application
Columns
Apply

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

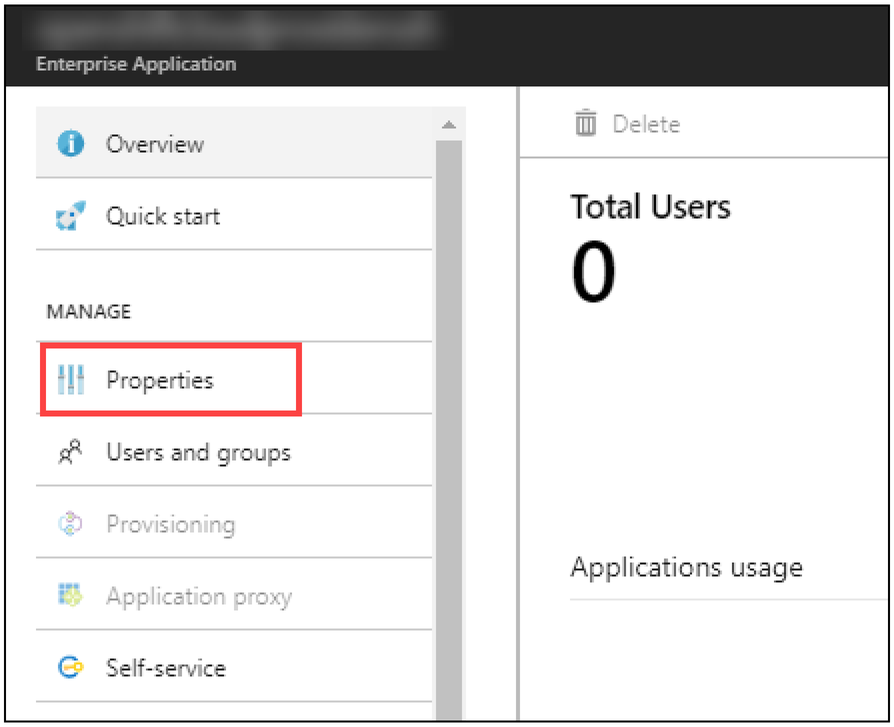
17. You can search the App by typing the name of the App you created earlier, in the search field. Select the App from the results.



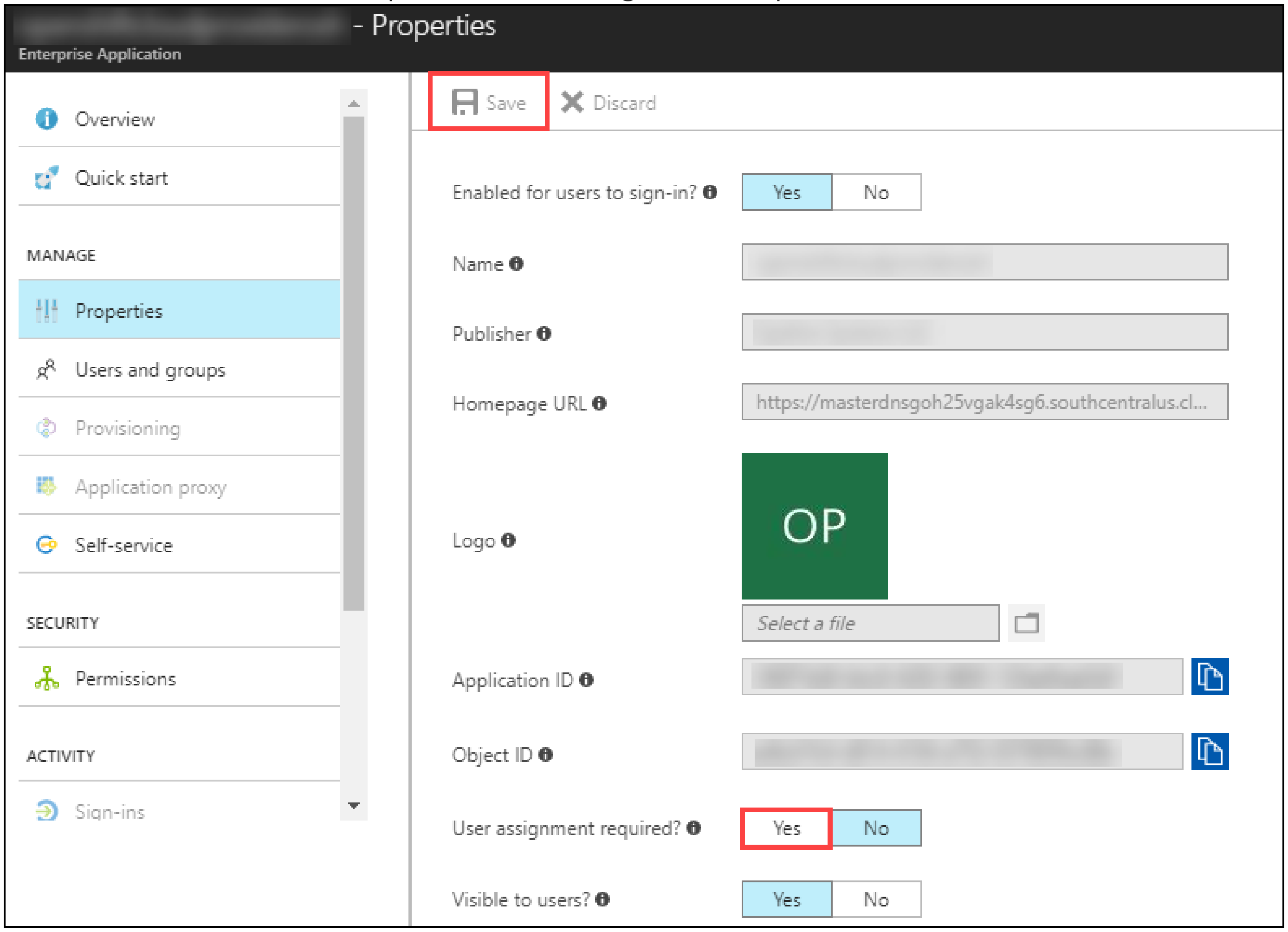
Show: All Applications
Applications status: Any
Application visibility: Any

NAME	HOMEPAGE URL	OBJECT ID	APPLICATION ID	PUBLISHER
OP [illegible]	https://masterdnsgoh25vga...	[illegible]	[illegible]	[illegible]

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.



Enabled for users to sign-in? [✓ Yes] No

Name [redacted]

Publisher [redacted]

Homepage URL https://masterdnsgoh25vgak4sg6.southcentralus.cl...

Logo OP

Select a file

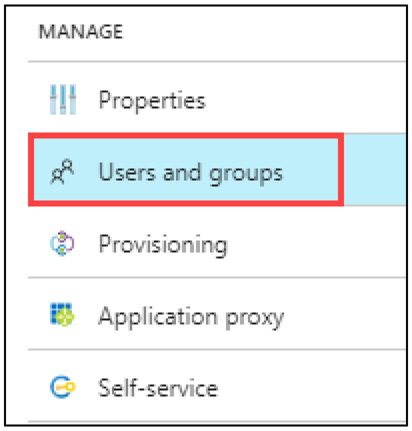
Application ID [redacted]

Object ID [redacted]

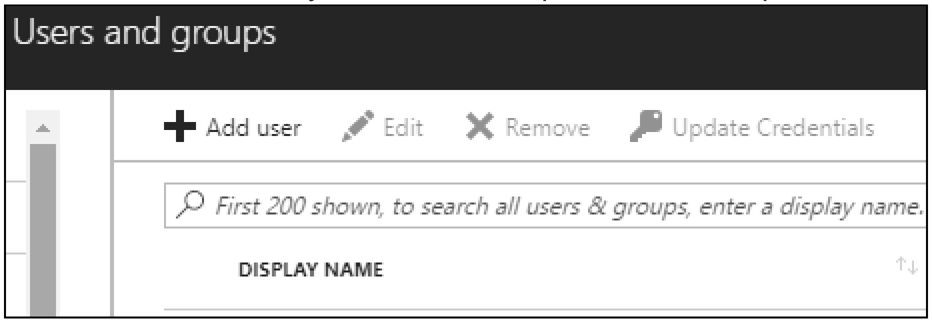
User assignment required? Yes [✓ No]

Visible to users? [✓ Yes] No

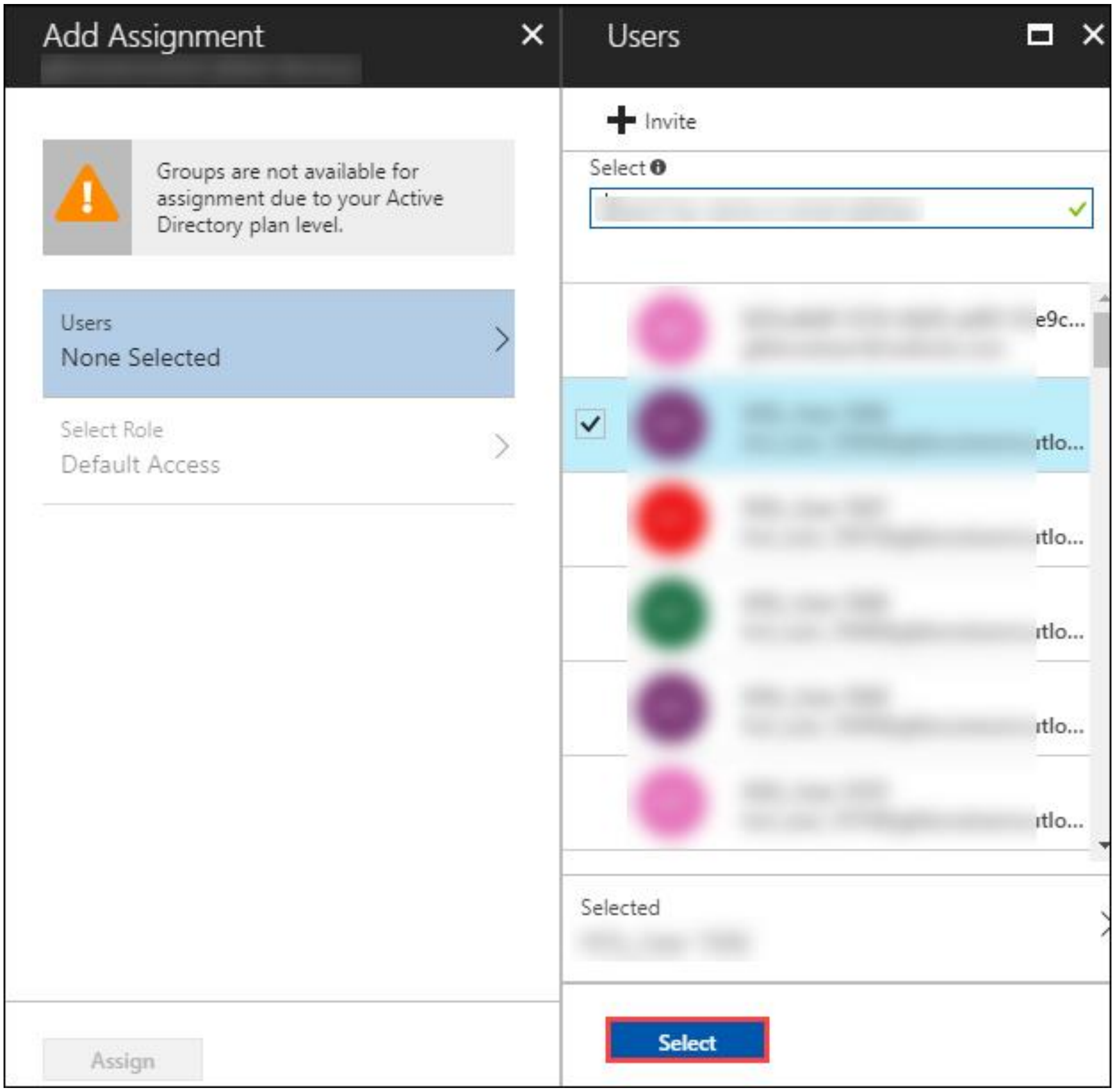
20. Now, click on **Users and groups** under Manage section on the left side of the App Blade.



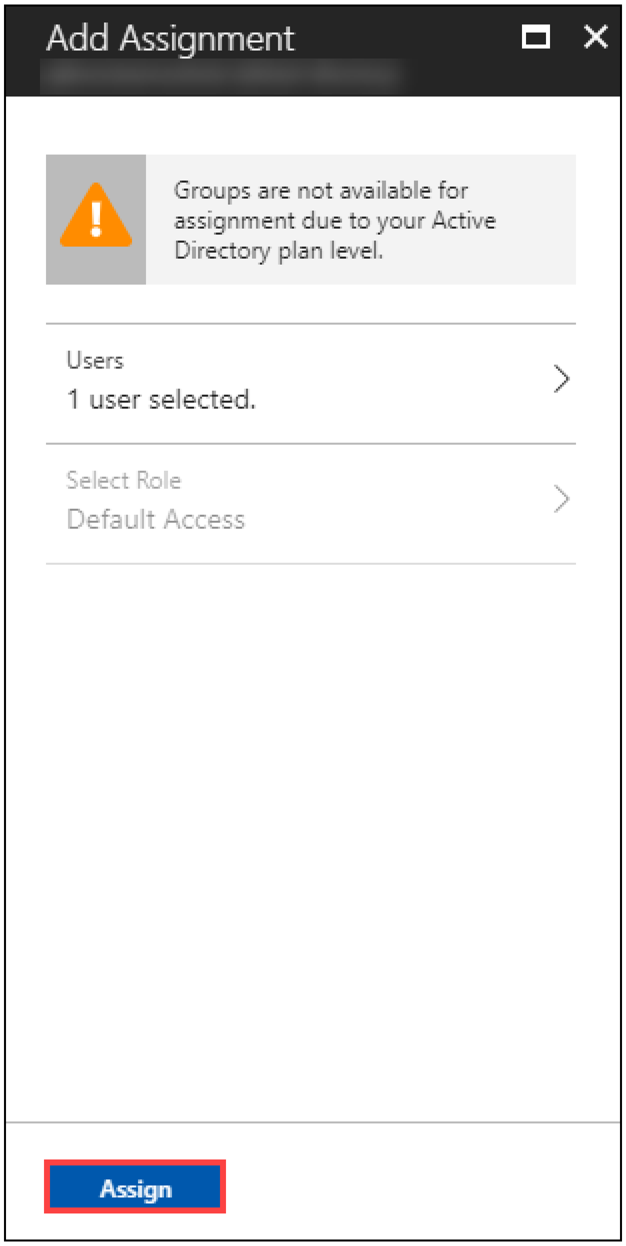
21. In the blade that come up, click on **+Add user** to assign a user to the app.
    Note: If user is already added then skip next three steps.


Users and groups
Add user
Edit
Remove
Update Credentials
First 200 shown, to search all users & groups, enter a display name.
DISPLAY NAME

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

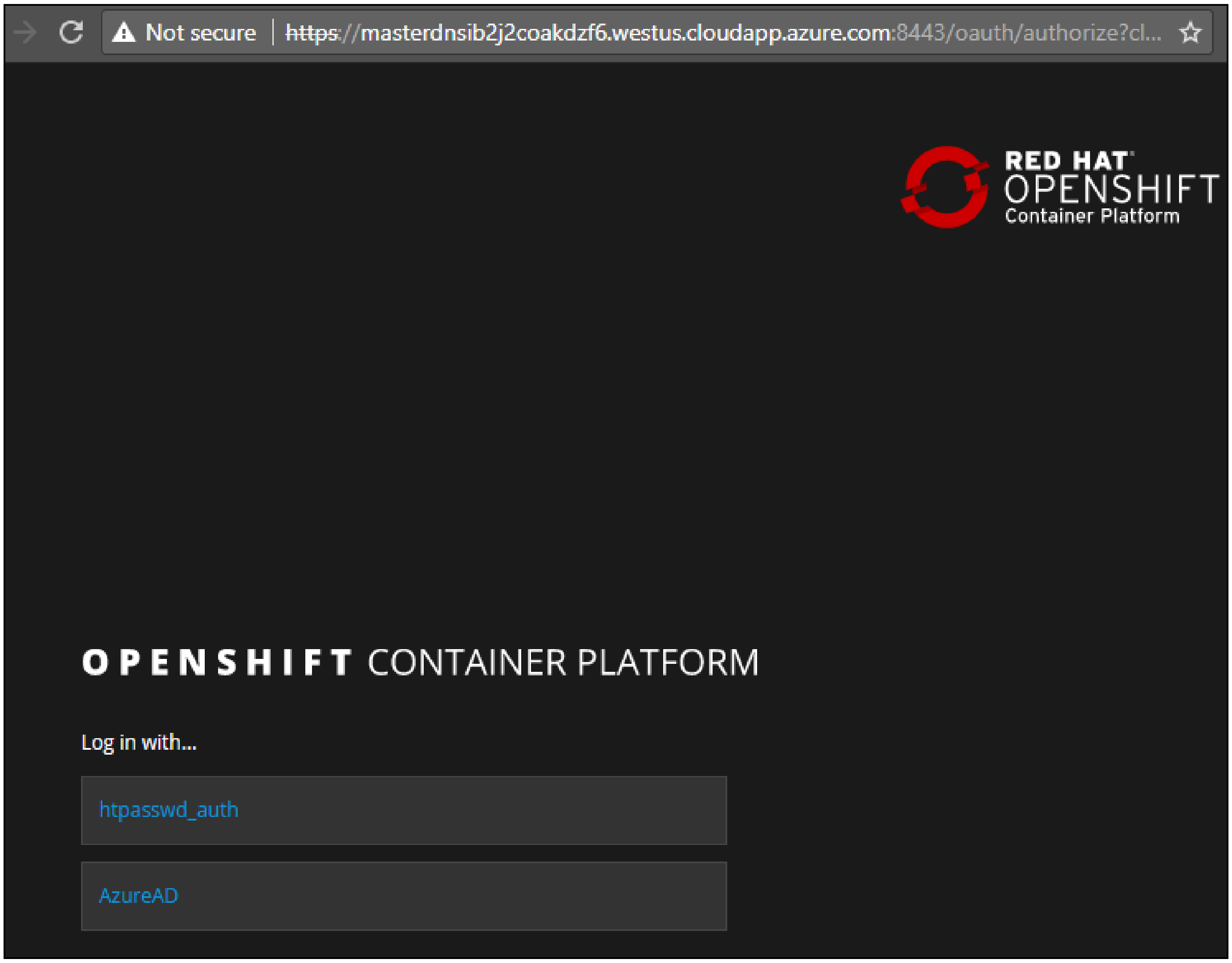
[redacted]

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

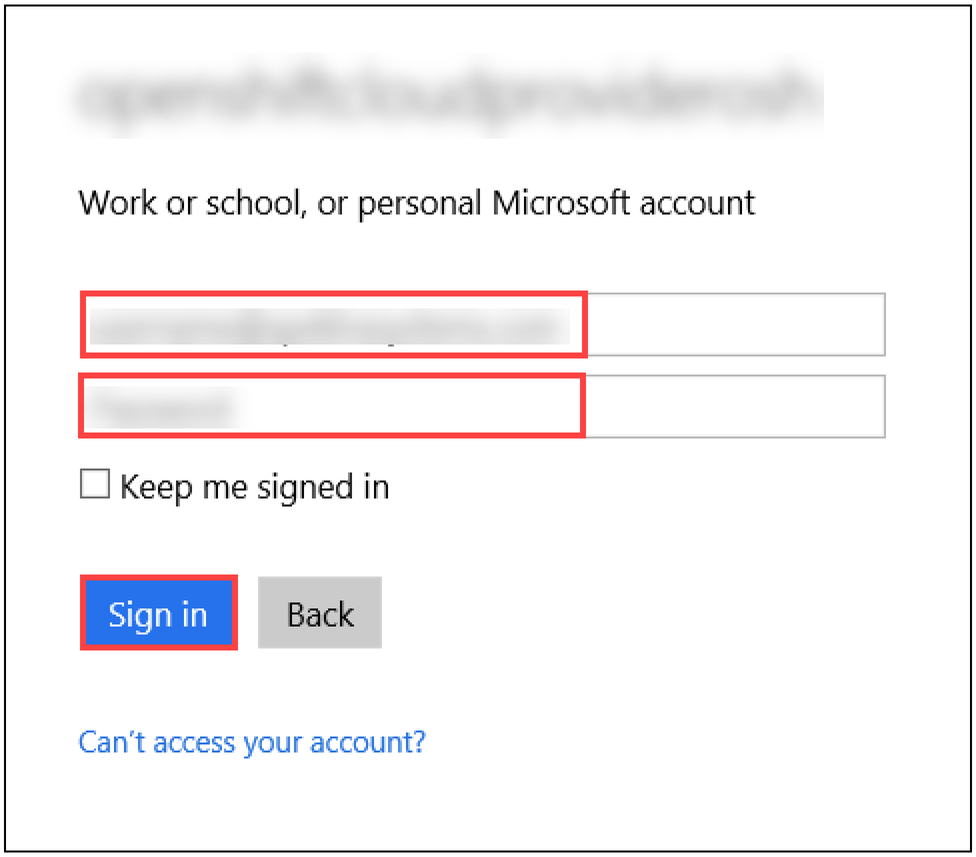
Users
1 user selected.

Select Role
Default Access

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


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

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

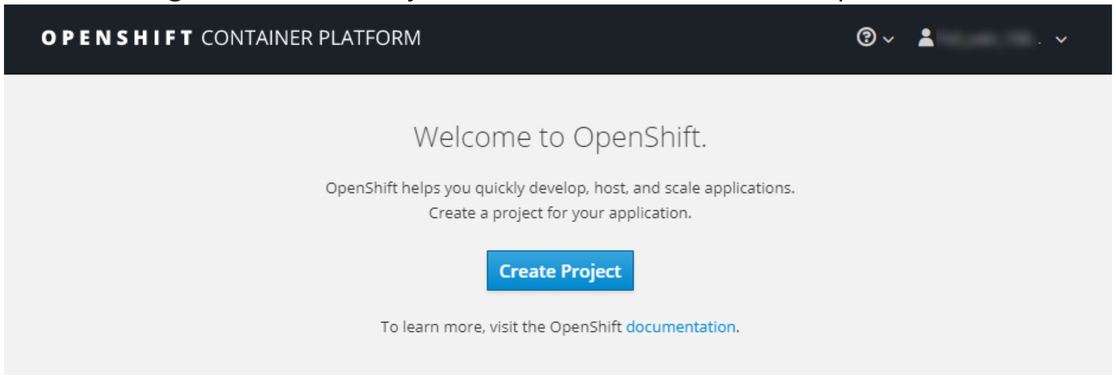
Keep me signed in

Sign in

Back

Can't access your account?

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


OPENSHIFT CONTAINER PLATFORM

Welcome to OpenShift.

OpenShift helps you quickly develop, host, and scale applications.
Create a project for your application.

Create Project

To learn more, visit the OpenShift documentation.

Lab 3: Deploying workload on Openshift

Lab overview

In this lab, we will deploy a workload on OpenShift.

Prerequisites

- Lab 1 must be completed

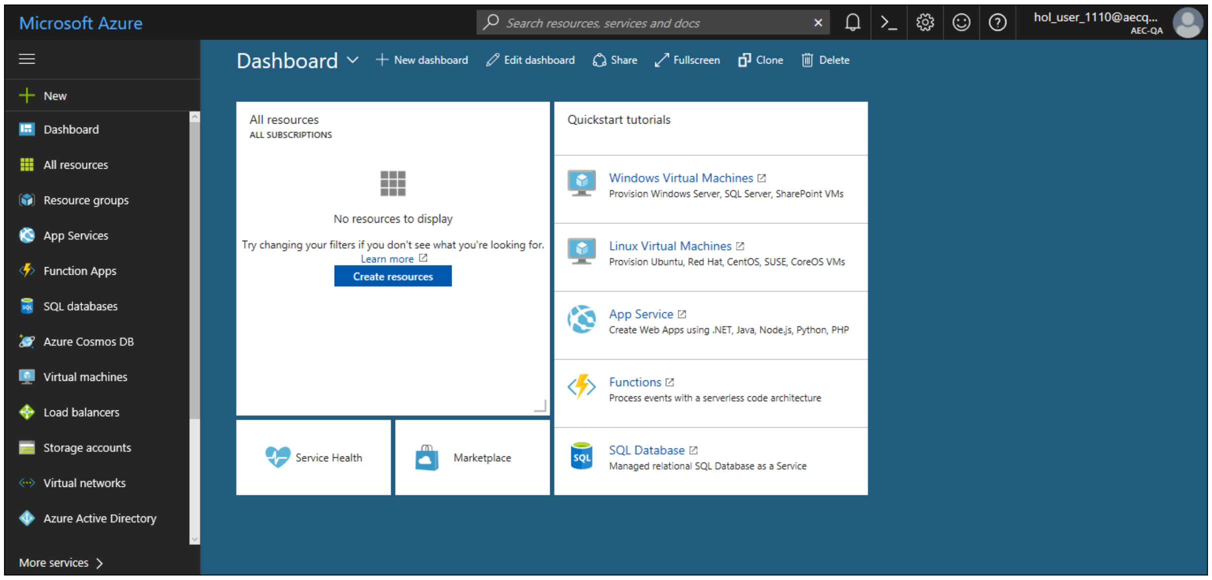
Time Estimate

45 minutes

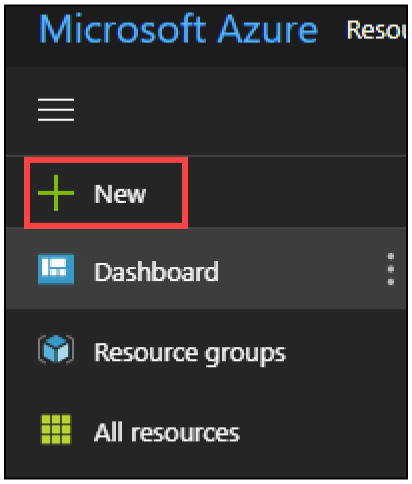
Exercise 1: Deploy a 2 Tier Node JS Application on Open Shift

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

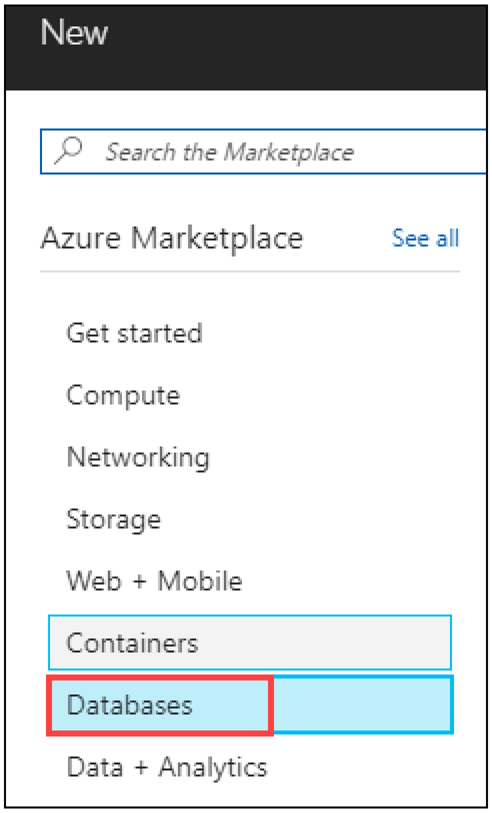
1. **Launch** a browser and **Navigate** to https://portal.azure.com. **Login** with the Microsoft Azure credentials you received via email.



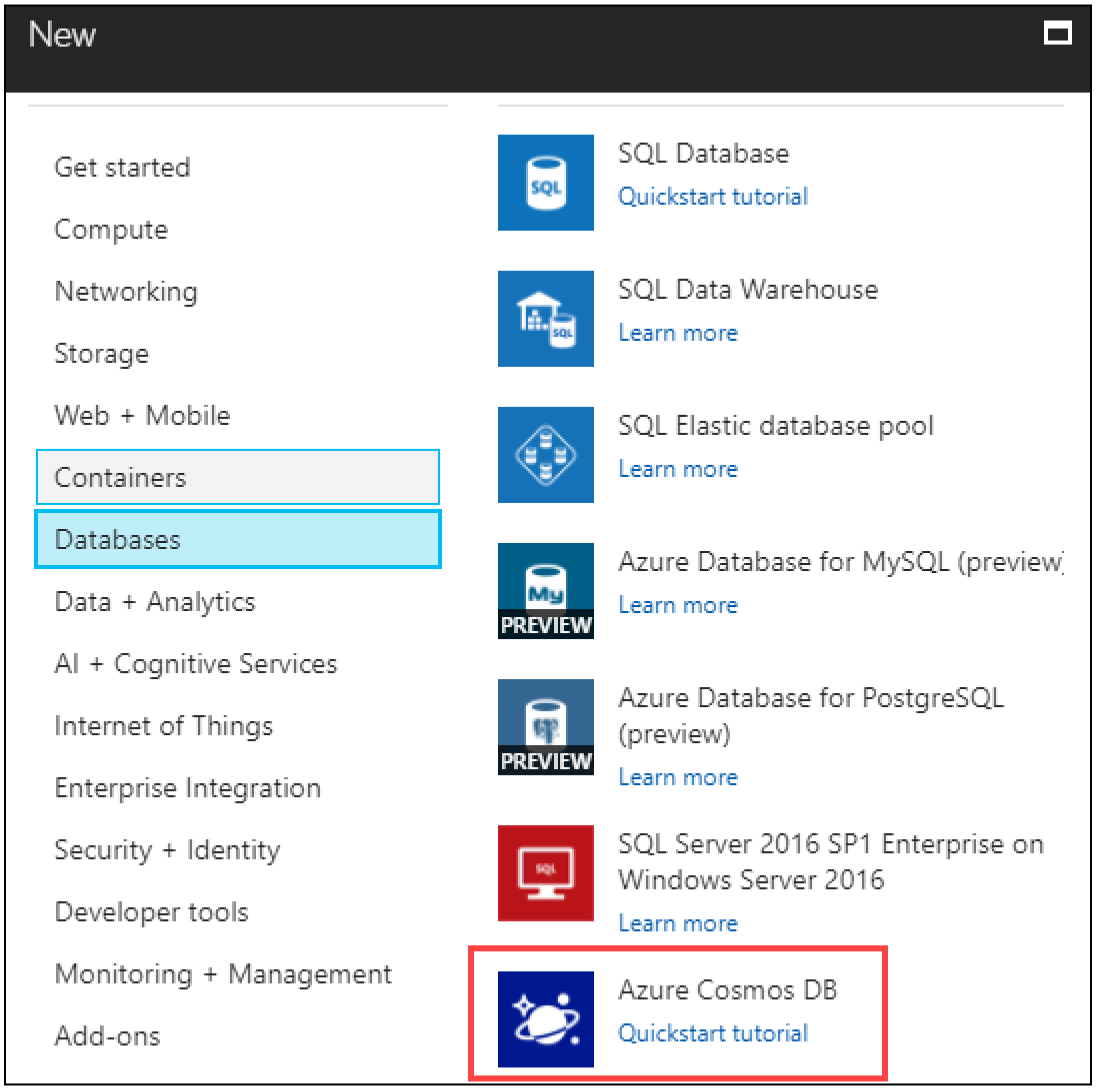
2. Click on +New on the left side of the Dashboard.



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



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


New

Get started
Compute
Networking
Storage
Web + Mobile
Containers
Databases
Data + Analytics
AI + Cognitive Services
Internet of Things
Enterprise Integration
Security + Identity
Developer tools
Monitoring + Management
Add-ons

SQL Database
Quickstart tutorial

SQL Data Warehouse
Learn more

SQL Elastic database pool
Learn more

Azure Database for MySQL (preview
Learn more

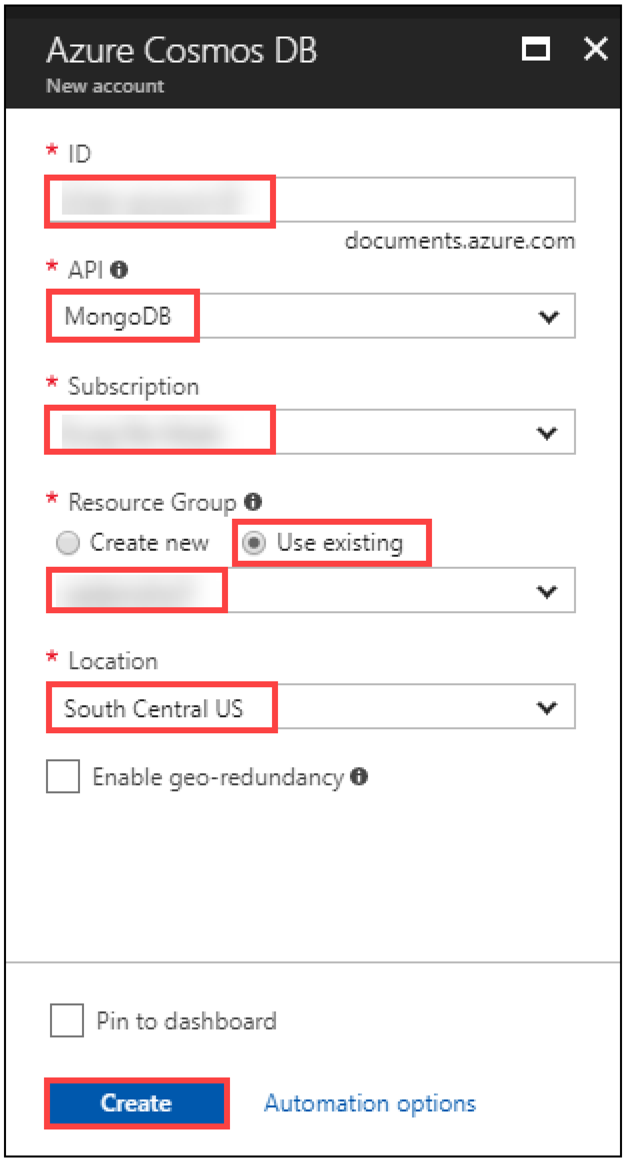
Azure Database for PostgreSQL (preview)
Learn more

SQL Server 2016 SP1 Enterprise on Windows Server 2016
Learn more

Azure Cosmos DB
Quickstart tutorial

5. 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**


Azure Cosmos DB
New account

* ID

documents.azure.com

* API: MongoDB

* Subscription

* Resource Group: ○ Create new [● Use existing]

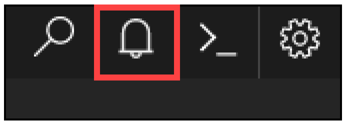
* Location: South Central US

☐ Enable geo-redundancy

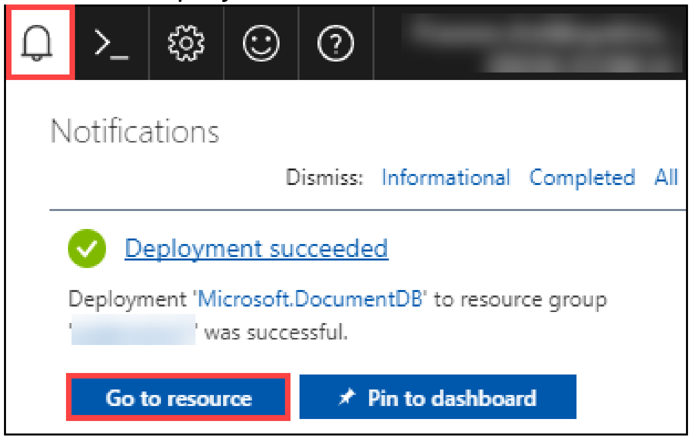
☐ Pin to dashboard

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.

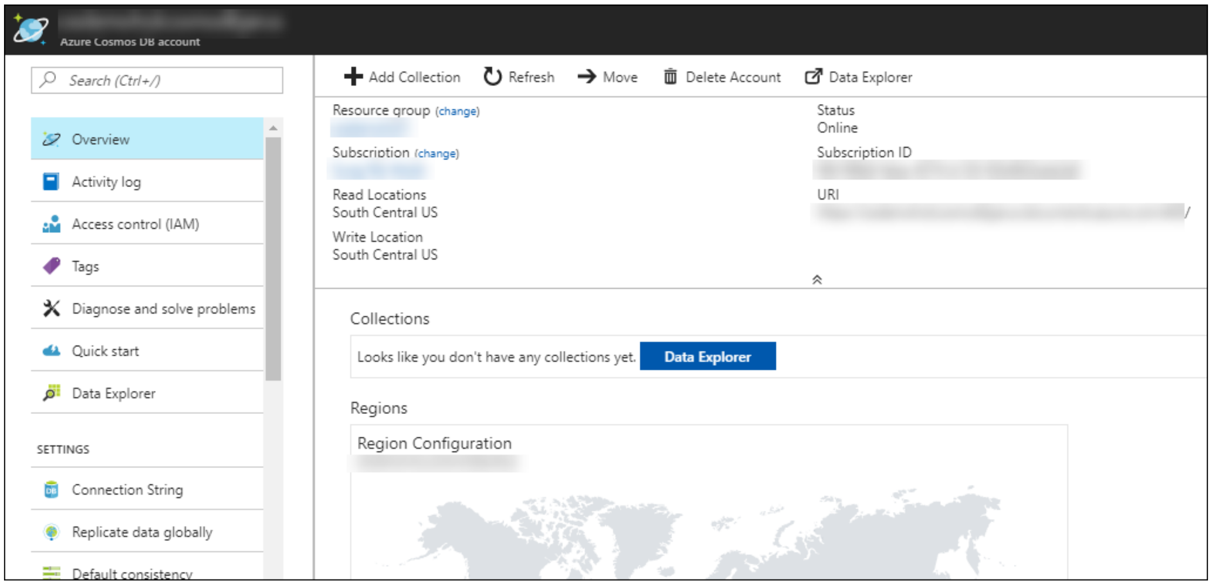

Notifications

Dismiss: Informational Completed All

Deployment succeeded

Deployment 'Microsoft.DocumentDB' to resource group ' ' was successful.

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



Resource group (change)

Subscription (change)

Read Locations
South Central US

Write Location
South Central US

Status
Online

Subscription ID

URI

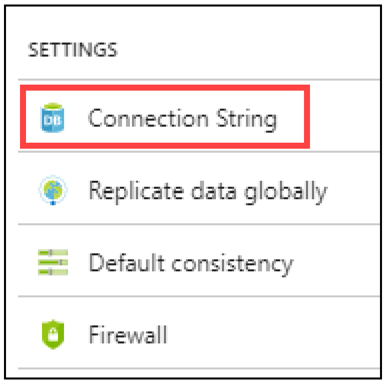
Collections

Looks like you don't have any collections yet. Data Explorer

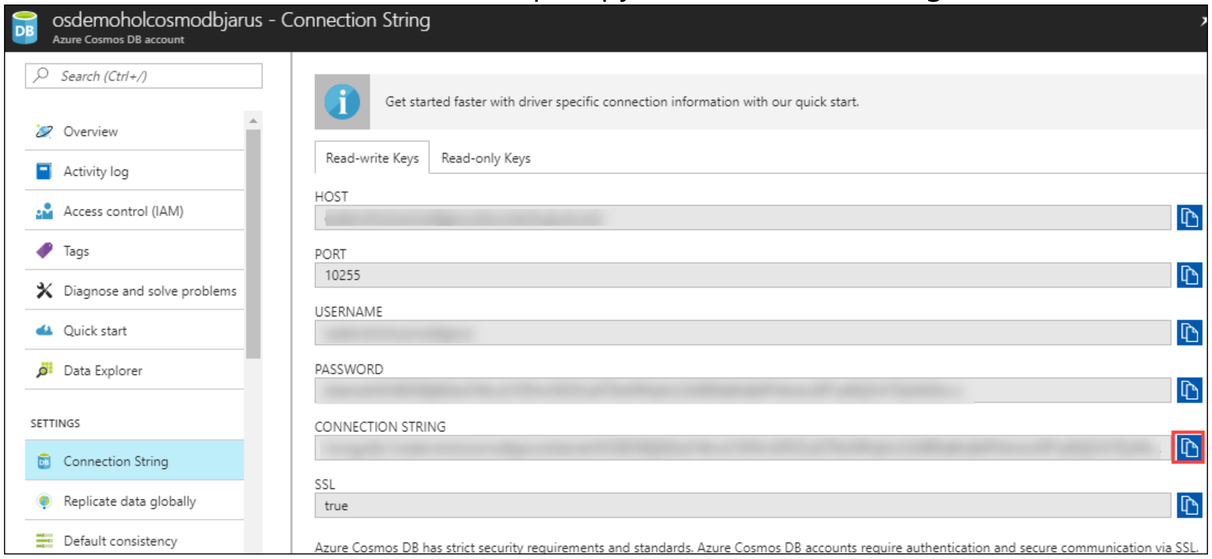
Regions

Region Configuration

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.



Get started faster with driver specific connection information with our quick start.

Read-write Keys | Read-only Keys

HOST

PORT
10255

USERNAME

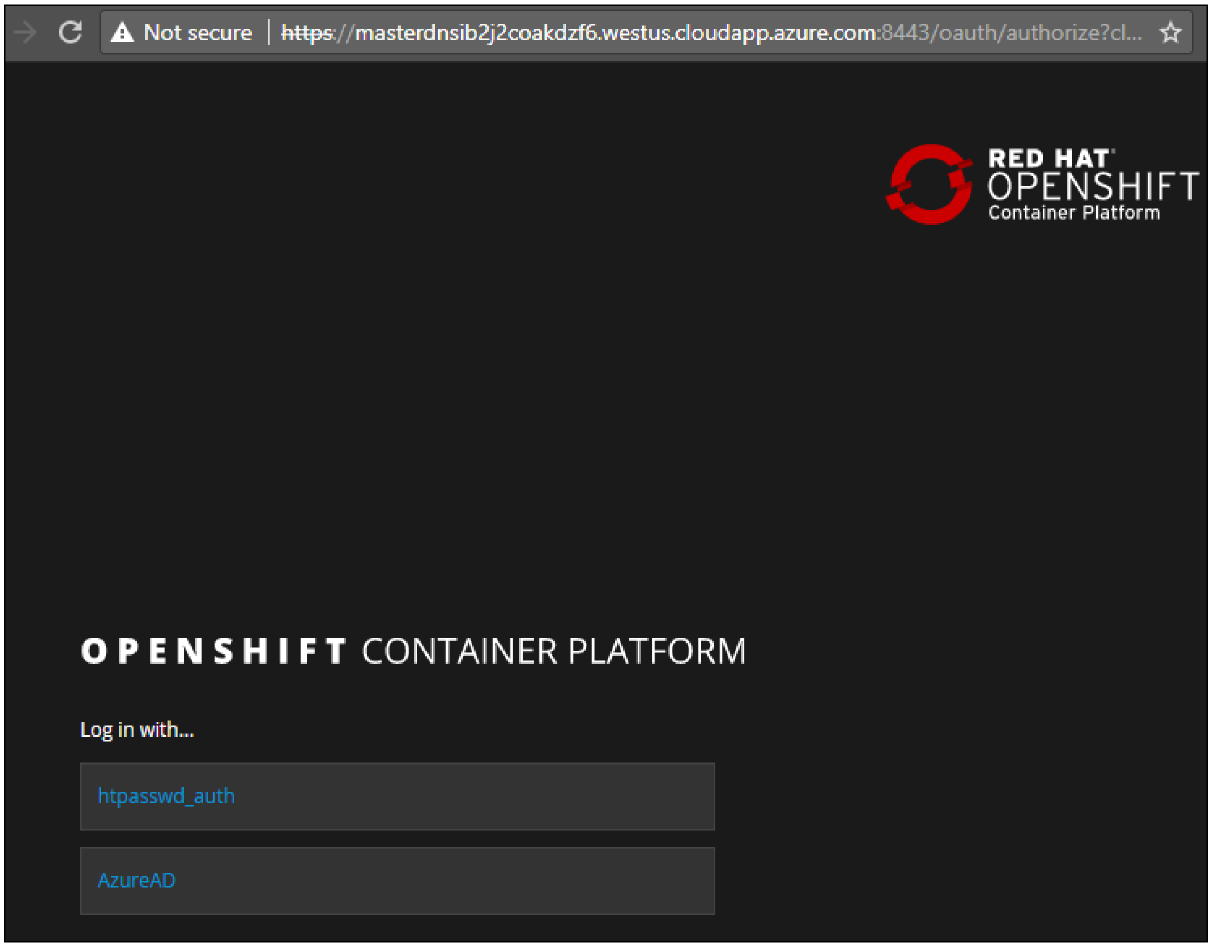
PASSWORD

CONNECTION STRING

SSL
true

Azure Cosmos DB has strict security requirements and standards. Azure Cosmos DB accounts require authentication and secure communication via SSL.

11. 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.


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

Exercise 2: Installing OpenShift CLI

COMMAND LINE INTERFACE

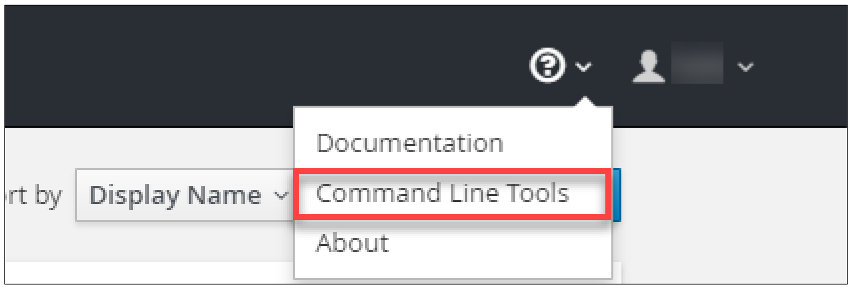
OpenShift ships with a feature rich web console as well as command line tools to provide users with a nice interface to work with applications deployed to the platform. The OpenShift tools are a single executable written in the Go programming language and is available for the following operating systems:

- Microsoft Windows
- Apple OS X
- Linux

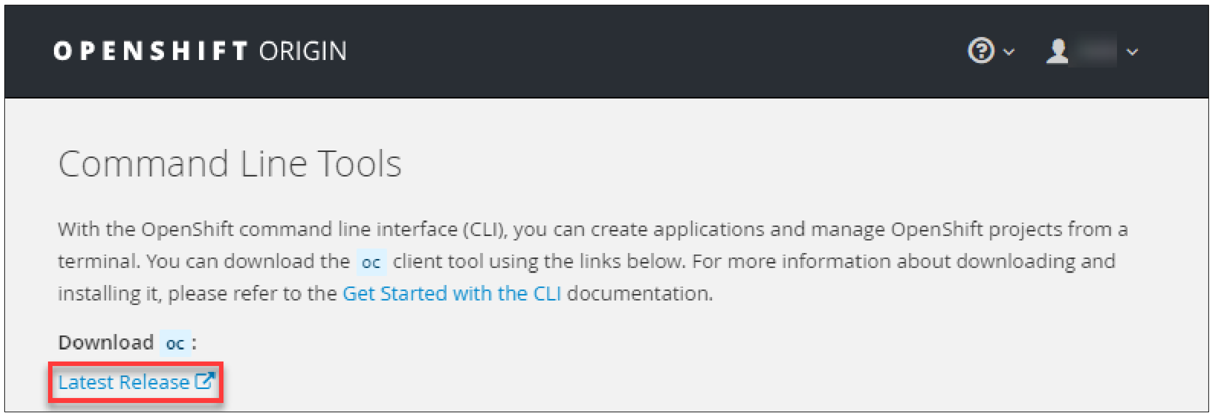
Installing the CLI

The easiest way to download the CLI is by accessing the **Command line tools** page on the web console.

1. Click on down arrow key as shown in below screenshot and click on **Command Line Tools.**



2. On **Command Line Tools** page, click on **Latest Release.**



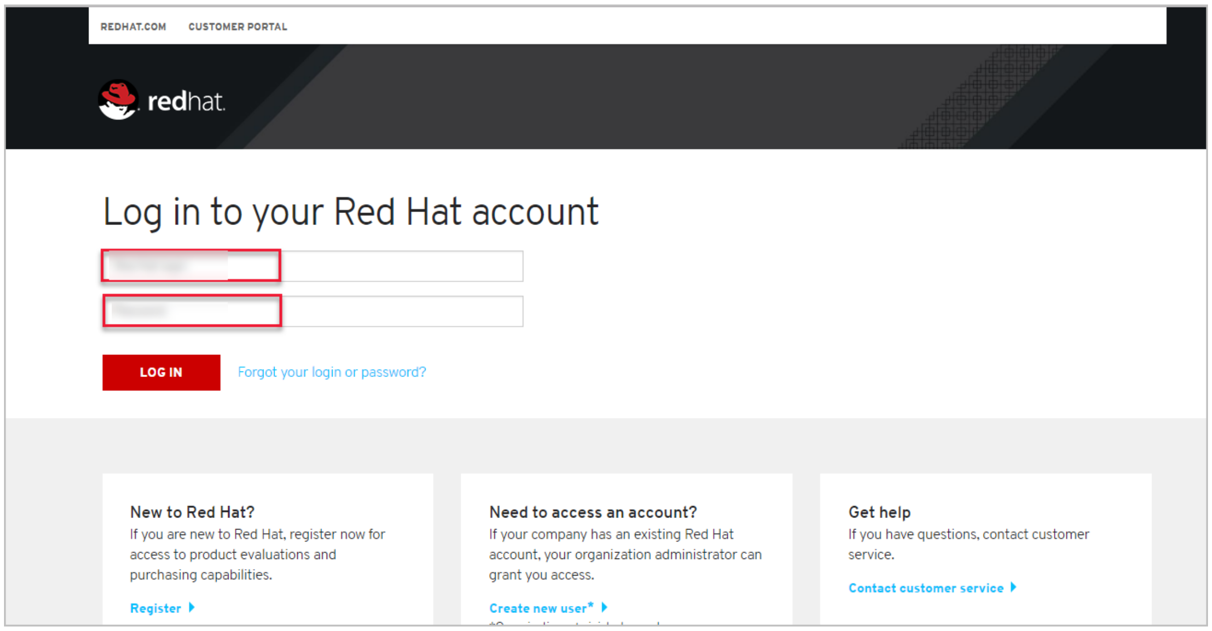
Command Line Tools

With the OpenShift command line interface (CLI), you can create applications and manage OpenShift projects from a terminal. You can download the `oc` client tool using the links below. For more information about downloading and installing it, please refer to the Get Started with the CLI documentation.

Download `oc` :

Latest Release

3. Now, you need to login in to your red hat account(one which has license for Openshift)



Log in to your Red Hat account

LOG IN

Forgot your login or password?

New to Red Hat?
If you are new to Red Hat, register now for access to product evaluations and purchasing capabilities.
Register

Need to access an account?
If your company has an existing Red Hat account, your organization administrator can grant you access.
Create new user*

Get help
If you have questions, contact customer service.
Contact customer service

4. scroll down and click on download.

5. Once the file has been downloaded, you will need to extract the contents of the same at below directories.

Windows:	***C:\OpenShift***
OS X:	***~/OpenShift***
Linux:	***~/OpenShift***

Windows: To extract a zip archive on windows, you will need a zip utility installed on your system. With newer versions of windows (greater than XP), this is provided by the operating system. Just right click on the downloaded file using file explorer and select to extract the contents to

OS X: Open a terminal window and change to the directory where you downloaded the file. Once you are in the directory, enter in the following command:

```
$ tar zxvf <File_Name>
```

Linux: Open a terminal window and change to the directory where you downloaded the file. Once you are in the directory, enter in the following command:

```
$ tar zxvf <File_Name>
```

6. Now you will need to add **oc** to your system's environment variable path:
Windows: Open Command prompt and run below command:

```
set PATH=%PATH%;C:\OpenShift
```

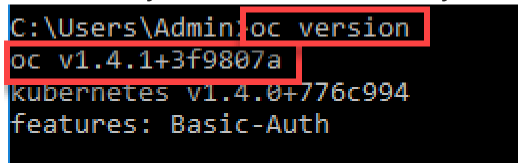
OS X: Open shell and run below command.

```
$ export PATH=$PATH:~/OpenShift
```

```
Linux: Open shell and run below command.
```

```
$ export PATH=$PATH:~/OpenShift
```

7. Now run below command on shell/command prompt to check the version of OpenShift client an to verify that it is successfully configured.



```
C:\Users\Admin>oc version
oc v1.4.1+3f9807a
kubernetes v1.4.0+776c994
features: Basic-Auth
```

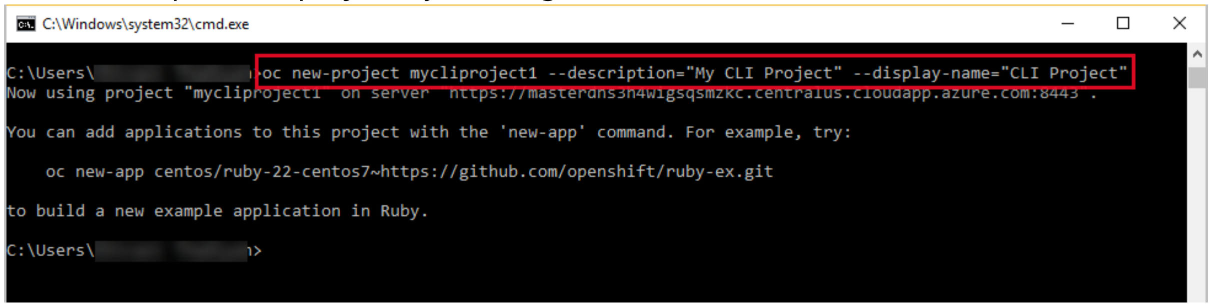
Exercise 3: Deployment in OpenShift using CLI

In this exercise, you will learn how to create a new project on OpenShift and how to create an application from an existing docker image**.**

1. Launch the command line and run below command and enter username and password as you have received in your lab mail.

```
oc login <URL of Openshift:8443>
```

2. Create an OpenShift project by running below command.



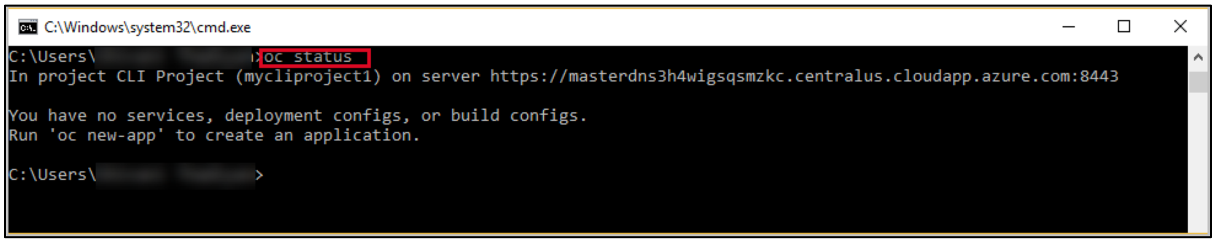
```
C:\Users\[redacted]>oc new-project mycliproject1 --description="My CLI Project" --display-name="CLI Project"
Now using project "mycliproject1" on server "https://masterdns3h4wigsqsmzkc.centralus.cloudapp.azure.com:8443".

You can add applications to this project with the 'new-app' command. For example, try:

    oc new-app centos/ruby-22-centos7~https://github.com/openshift/ruby-ex.git

to build a new example application in Ruby.

C:\Users\[redacted]>
```

3. Now you can see the project is created successfully.

```
oc get projects
```

4. You can also check the status of the project by running the following command.

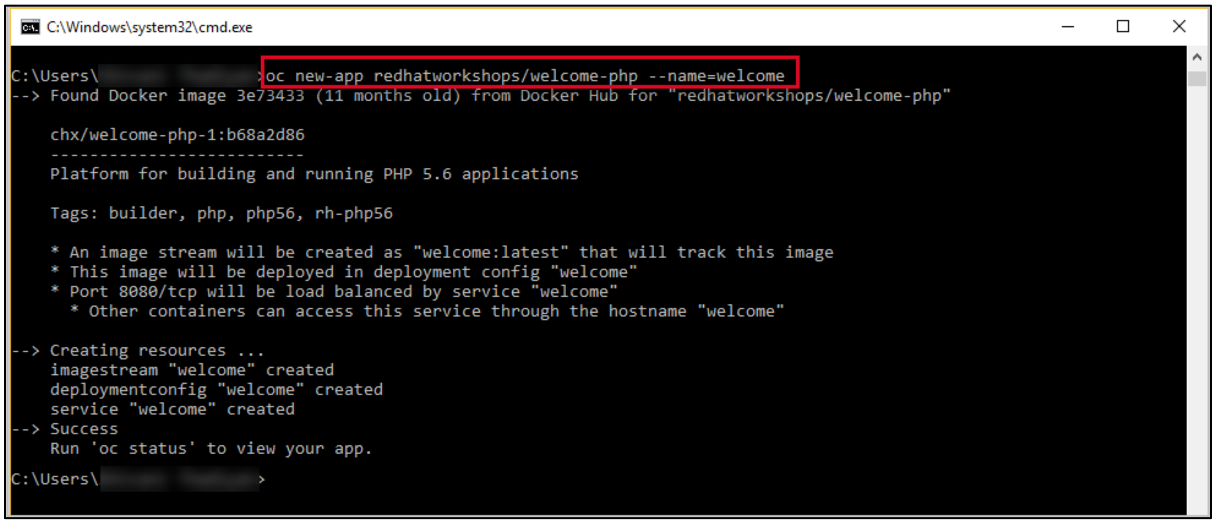
```
oc status
```



```
C:\Users\[redacted]>oc status
In project CLI Project (mycliproject1) on server https://masterdns3h4wigsqsmzkc.centralus.cloudapp.azure.com:8443

You have no services, deployment configs, or build configs.
Run 'oc new-app' to create an application.

C:\Users\[redacted]>
```

5. Create new application using below command

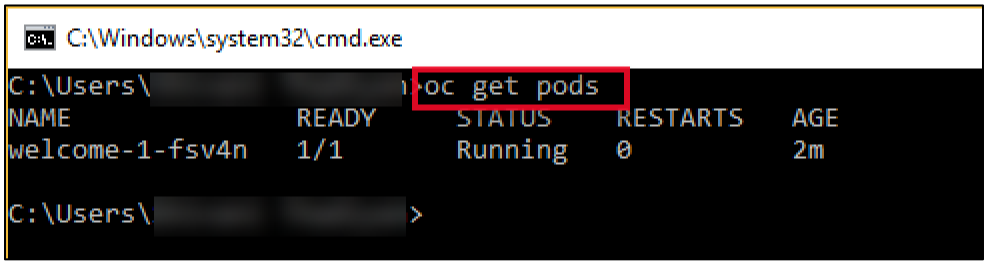
```
oc new-app redhatworkshops/welcome-php --name=welcome
```



```
C:\Users\[redacted]>oc new-app redhatworkshops/welcome-php --name=welcome
--> Found Docker image 3e73433 (11 months old) from Docker Hub for "redhatworkshops/welcome-php"

    chx/welcome-php-1:b68a2d86
    --------------------------
    Platform for building and running PHP 5.6 applications

    Tags: builder, php, php56, rh-php56

    * An image stream will be created as "welcome:latest" that will track this image
    * This image will be deployed in deployment config "welcome"
    * Port 8080/tcp will be load balanced by service "welcome"
      * Other containers can access this service through the hostname "welcome"

--> Creating resources ...
    imagestream "welcome" created
    deploymentconfig "welcome" created
    service "welcome" created
--> Success
    Run 'oc status' to view your app.
C:\Users\[redacted]>
```

6. The above command uses the docker image to deploy a docker container in a pod. you will notice that a deployed pod runs and it starts an application pod as shown below.

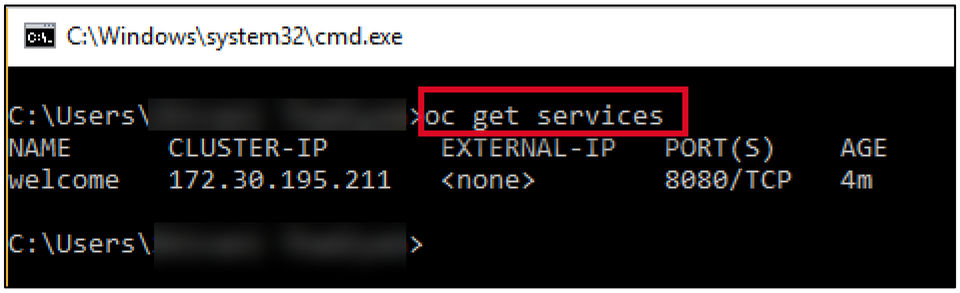
```
oc get pods
```



```
C:\Users\[redacted]>oc get pods
NAME                READY     STATUS    RESTARTS   AGE
welcome-1-fsv4n     1/1       Running   0          2m

C:\Users\[redacted]>
```

7. To view the list of services in the project, run the following command below

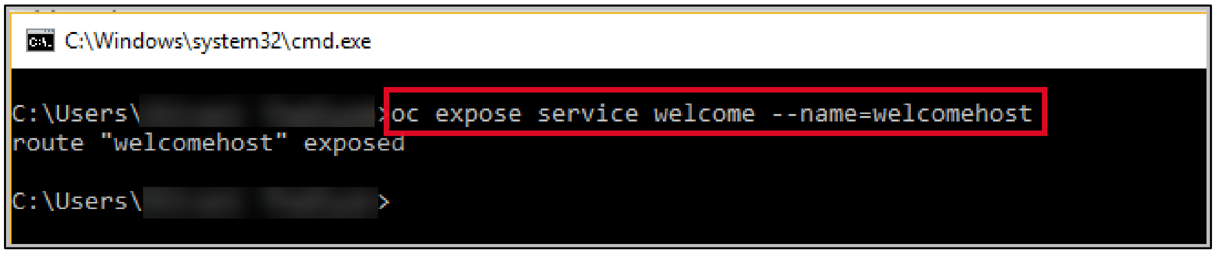
```
oc get services
```



```
C:\Users\[redacted]>oc get services
NAME      CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
welcome   172.30.195.211   <none>        8080/TCP   4m

C:\Users\[redacted]>
```

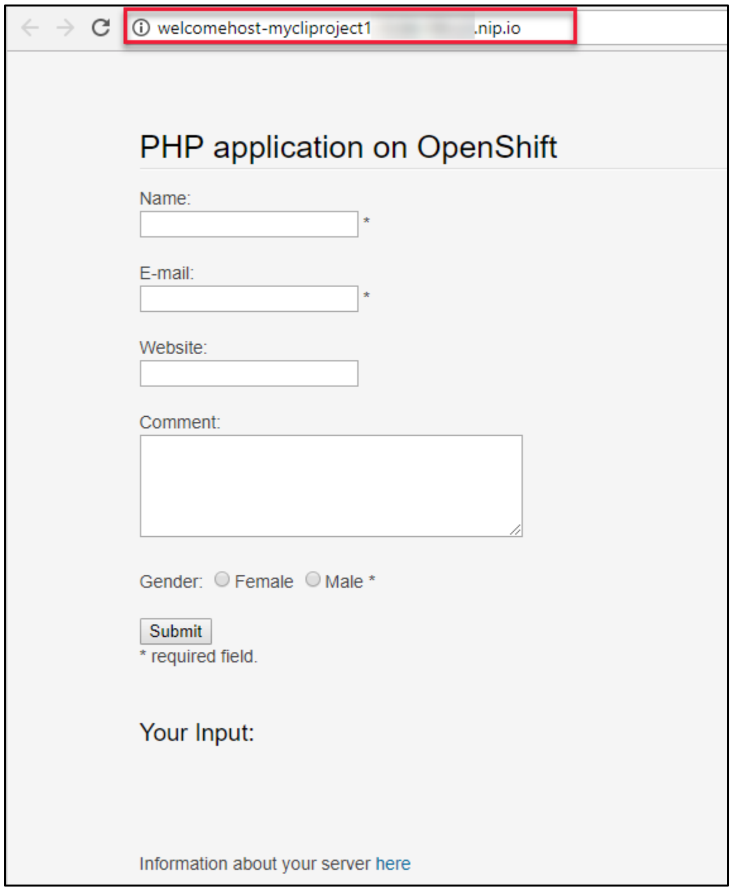
8. Now add a route to the service with the following command.

```
oc expose service welcome --name=welcome
```



```
C:\Users\[redacted]>oc expose service welcome --name=welcomehost
route "welcomehost" exposed

C:\Users\[redacted]>
```

9. Now go to your openshift platform and click on applications>hostname, you can access the application from the browser and see the result.


welcomehost-mycliproject1[redacted].nip.io

PHP application on OpenShift

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male *

Submit

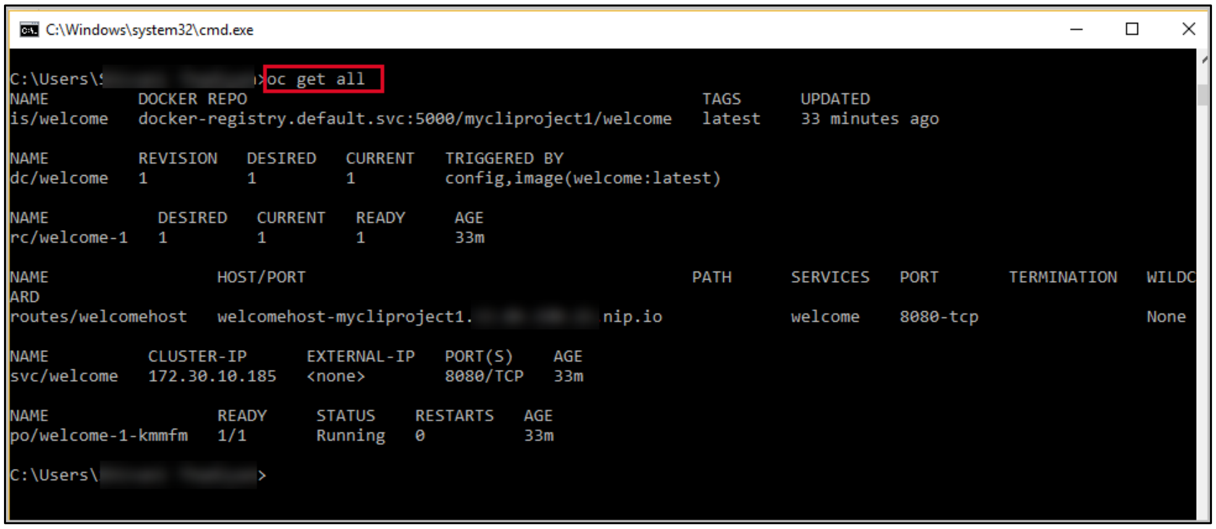
* required field.

Your Input:

Information about your server here

10. To view all the components that were created in your project, run he command is given below.

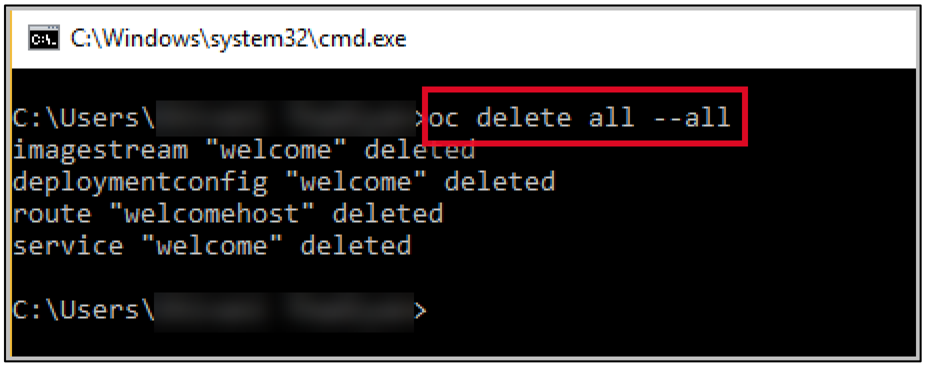
```
oc get all
```


C:\Windows\system32\cmd.exe

```
C:\Users\[redacted]>oc get all
NAME          DOCKER REPO                                              TAGS      UPDATED
is/welcome    docker-registry.default.svc:5000/mycliproject1/welcome   latest    33 minutes ago

NAME          REVISION   DESIRED   CURRENT   TRIGGERED BY
dc/welcome    1          1         1         config,image(welcome:latest)

NAME            DESIRED   CURRENT   READY     AGE
rc/welcome-1    1         1         1         33m

NAME                  HOST/PORT                                       PATH      SERVICES   PORT       TERMINATION   WILDC
ARD
routes/welcomehost    welcomehost-mycliproject1.[redacted]nip.io                 welcome    8080-tcp                 None

NAME          CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
svc/welcome   172.30.10.185    <none>        8080/TCP   33m

NAME                  READY     STATUS    RESTARTS   AGE
po/welcome-1-kmmfm    1/1       Running   0          33m

C:\Users\[redacted]>
```

11. Now you can delete all these components by running one command.

```
oc get all --all
```

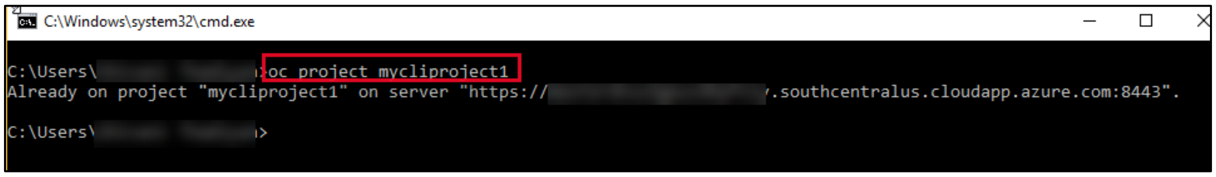


```
C:\Windows\system32\cmd.exe

C:\Users\[redacted]>oc delete all --all
imagestream "welcome" deleted
deploymentconfig "welcome" deleted
route "welcomehost" deleted
service "welcome" deleted

C:\Users\[redacted]>
```

Exercise 4: Create an App using Docker build

In this exercise, you will learn how to create an application from a Dockerfile. OpenShift takes Dockerfile as an input and generates your application docker image for you.

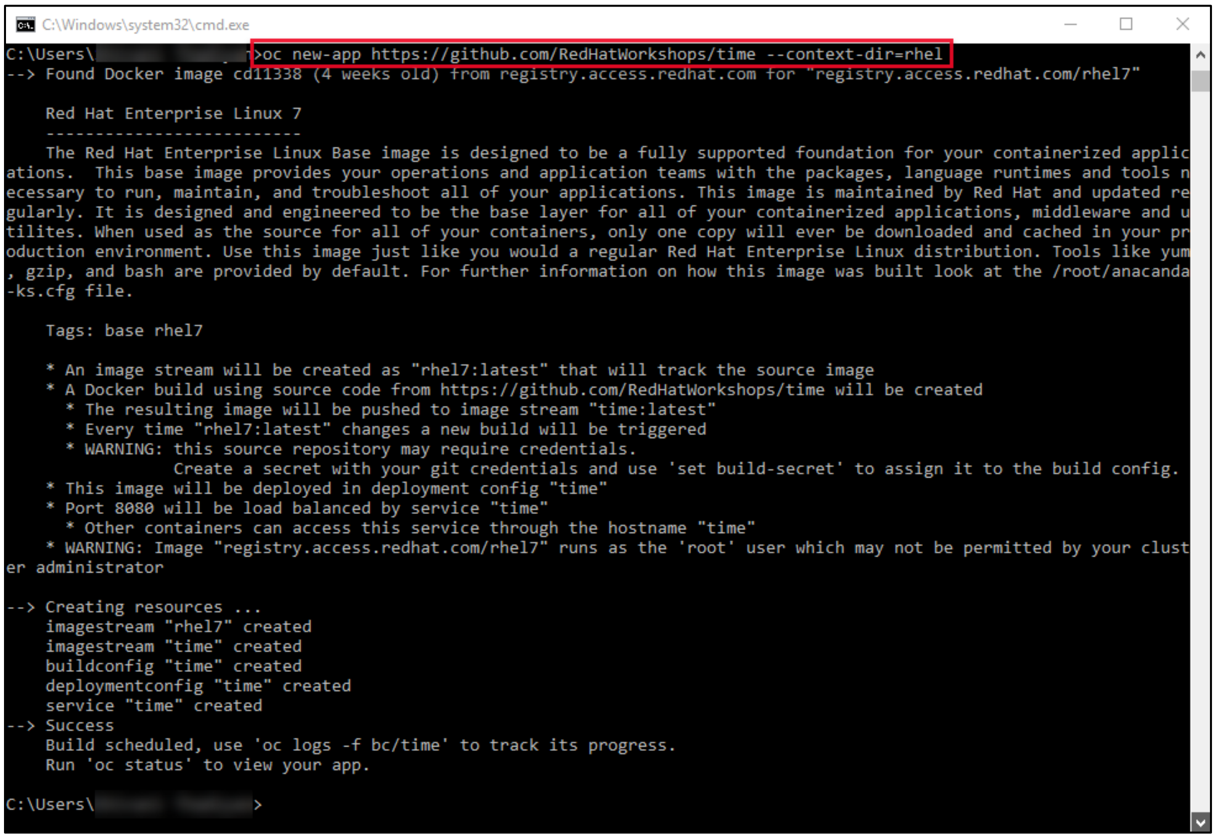
1. You can create a new project or use existing project that created in exercise 3. To make sure you have the existing project run the following command.



```
C:\Windows\system32\cmd.exe

C:\Users\[redacted]>oc project mycliproject1
Already on project "mycliproject1" on server "https://[redacted].southcentralus.cloudapp.azure.com:8443".

C:\Users\[redacted]>
```

2. Now, we are using the Dockerfile as the basis to create a docker image for application. Run the command is given below.

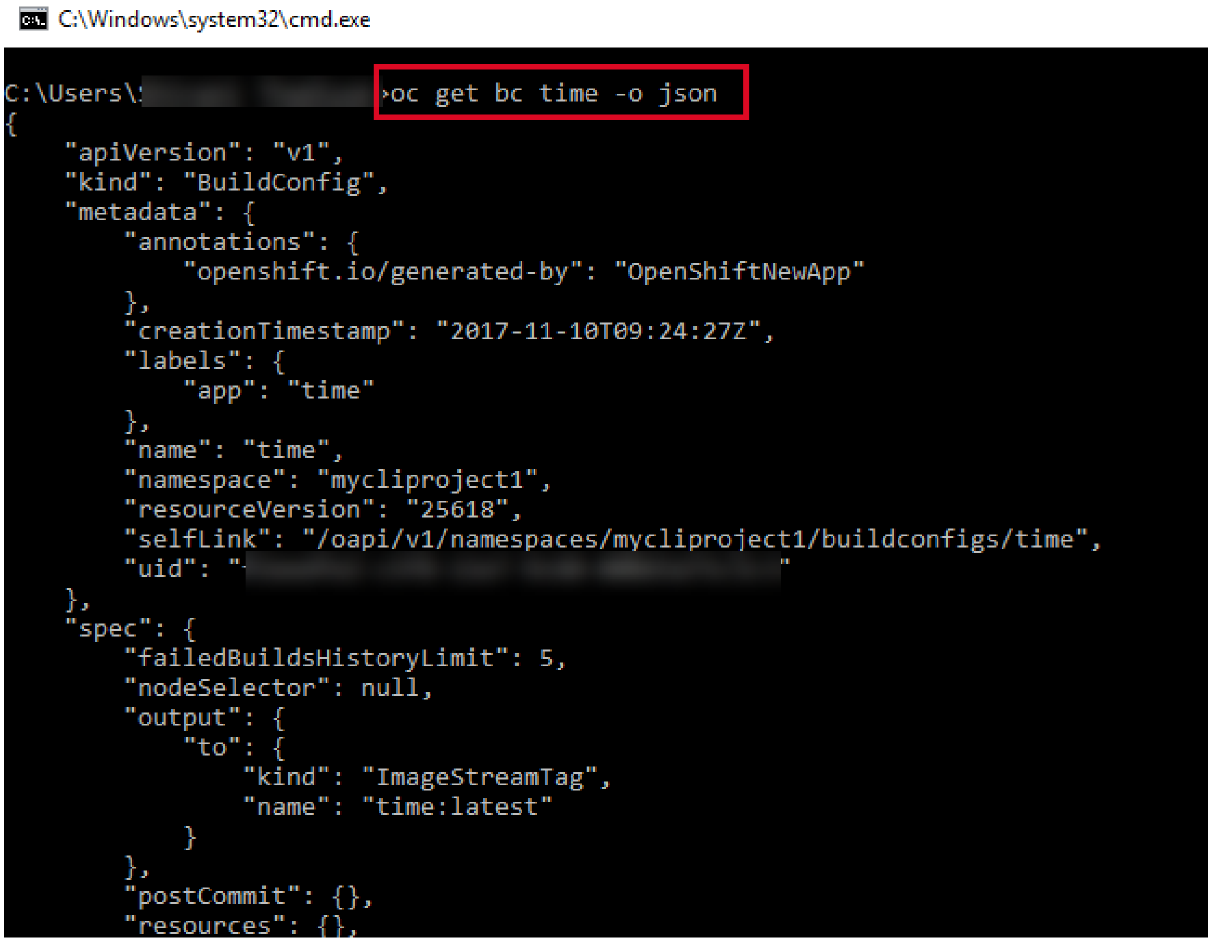
```
oc new-app https://github.com/RedHatWorkshops/time --context-dir=rhel
```



```
C:\Windows\system32\cmd.exe
C:\Users\[redacted]>oc new-app https://github.com/RedHatWorkshops/time --context-dir=rhel
--> Found Docker image cd11338 (4 weeks old) from registry.access.redhat.com for "registry.access.redhat.com/rhel7"

    Red Hat Enterprise Linux 7
    --------------------------
    The Red Hat Enterprise Linux Base image is designed to be a fully supported foundation for your containerized applic
ations.  This base image provides your operations and application teams with the packages, language runtimes and tools n
ecessary to run, maintain, and troubleshoot all of your applications. This image is maintained by Red Hat and updated re
gularly. It is designed and engineered to be the base layer for all of your containerized applications, middleware and u
tilites. When used as the source for all of your containers, only one copy will ever be downloaded and cached in your pr
oduction environment. Use this image just like you would a regular Red Hat Enterprise Linux distribution. Tools like yum
, gzip, and bash are provided by default. For further information on how this image was built look at the /root/anacanda
-ks.cfg file.

    Tags: base rhel7

    * An image stream will be created as "rhel7:latest" that will track the source image
    * A Docker build using source code from https://github.com/RedHatWorkshops/time will be created
      * The resulting image will be pushed to image stream "time:latest"
      * Every time "rhel7:latest" changes a new build will be triggered
      * WARNING: this source repository may require credentials.
                 Create a secret with your git credentials and use 'set build-secret' to assign it to the build config.
    * This image will be deployed in deployment config "time"
    * Port 8080 will be load balanced by service "time"
      * Other containers can access this service through the hostname "time"
    * WARNING: Image "registry.access.redhat.com/rhel7" runs as the 'root' user which may not be permitted by your clust
er administrator

--> Creating resources ...
    imagestream "rhel7" created
    imagestream "time" created
    buildconfig "time" created
    deploymentconfig "time" created
    service "time" created
--> Success
    Build scheduled, use 'oc logs -f bc/time' to track its progress.
    Run 'oc status' to view your app.

C:\Users\[redacted]>
```

3. Now, look at the buildconfig by running the command shown below.

```
oc get bc time -o json
```

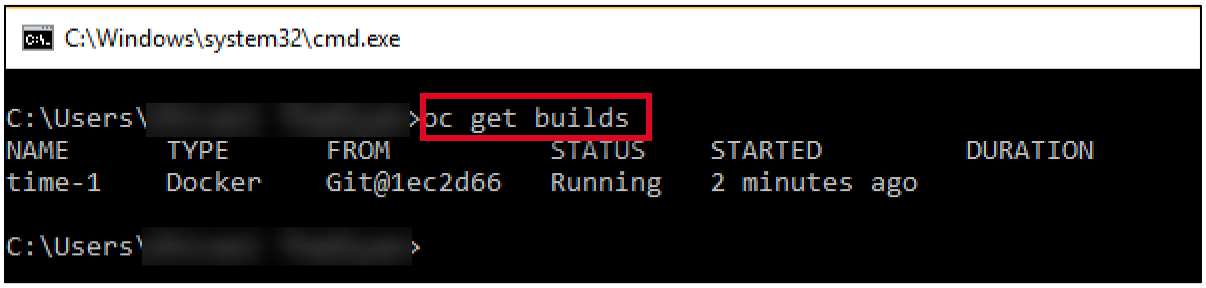


C:\Windows\system32\cmd.exe

```
C:\Users\[redacted]>oc get bc time -o json
{
    "apiVersion": "v1",
    "kind": "BuildConfig",
    "metadata": {
        "annotations": {
            "openshift.io/generated-by": "OpenShiftNewApp"
        },
        "creationTimestamp": "2017-11-10T09:24:27Z",
        "labels": {
            "app": "time"
        },
        "name": "time",
        "namespace": "mycliproject1",
        "resourceVersion": "25618",
        "selfLink": "/oapi/v1/namespaces/mycliproject1/buildconfigs/time",
        "uid": "[redacted]"
    },
    "spec": {
        "failedBuildsHistoryLimit": 5,
        "nodeSelector": null,
        "output": {
            "to": {
                "kind": "ImageStreamTag",
                "name": "time:latest"
            }
        },
        "postCommit": {},
        "resources": {},
```

4. To view the list of build, run command is given below.

```
oc get builds
```

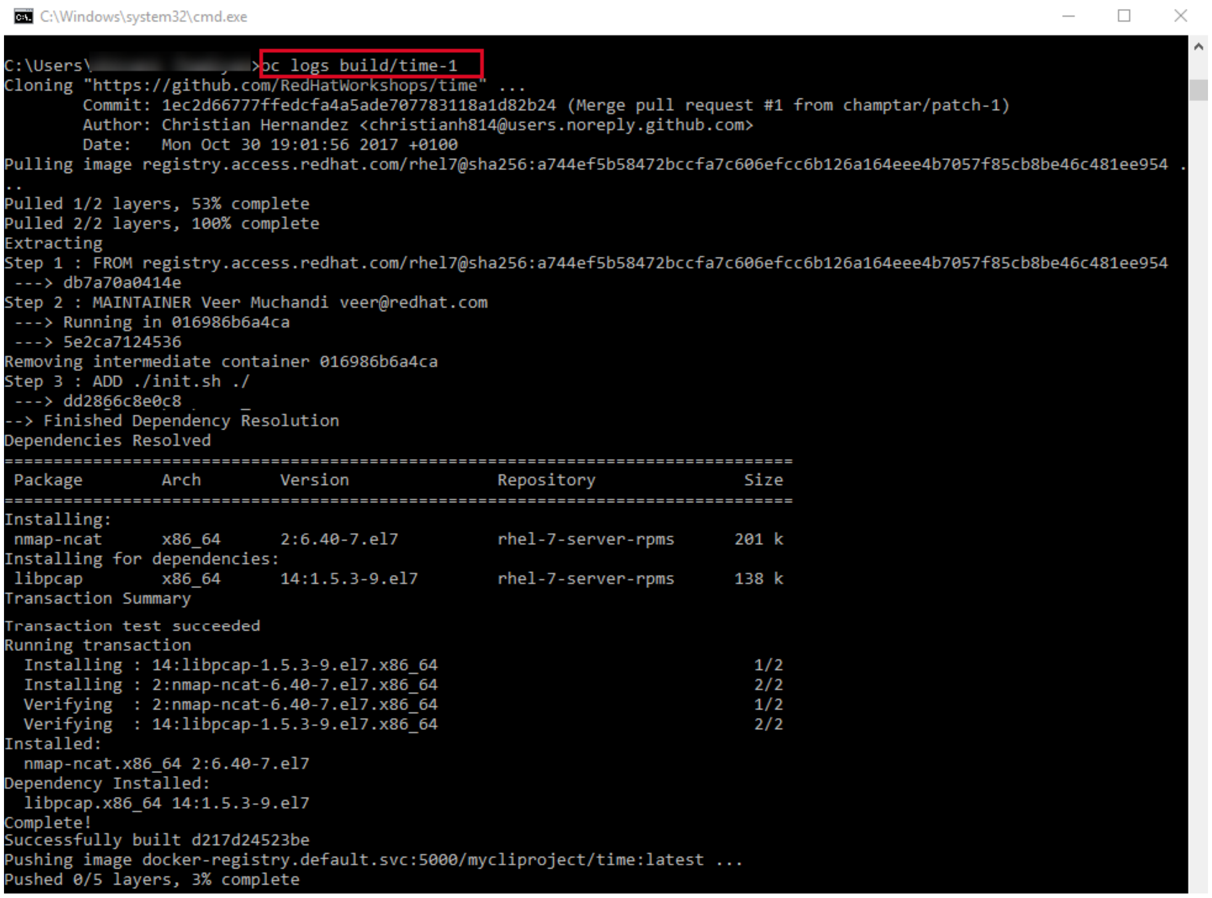


C:\Windows\system32\cmd.exe

```
C:\Users\[redacted]>oc get builds
NAME       TYPE      FROM          STATUS    STARTED          DURATION
time-1     Docker    Git@1ec2d66   Running   2 minutes ago

C:\Users\[redacted]>
```

5. Run the command as shown below to look at the build logs.

```
oc  logs build/time-1
```



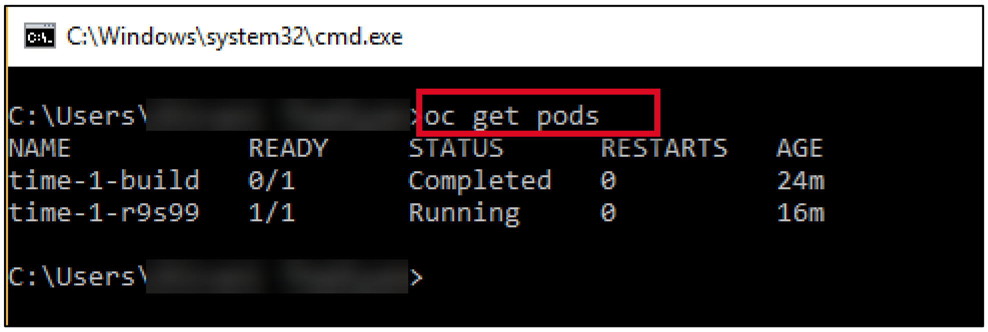
```
C:\Users\[redacted]>oc logs build/time-1
Cloning "https://github.com/RedHatWorkshops/time" ...
        Commit: 1ec2d66777ffedcfa4a5ade707783118a1d82b24 (Merge pull request #1 from champtar/patch-1)
        Author: Christian Hernandez <christianh814@users.noreply.github.com>
        Date:   Mon Oct 30 19:01:56 2017 +0100
Pulling image registry.access.redhat.com/rhel7@sha256:a744ef5b58472bccfa7c606efcc6b126a164eee4b7057f85cb8be46c481ee954 .
..
Pulled 1/2 layers, 53% complete
Pulled 2/2 layers, 100% complete
Extracting
Step 1 : FROM registry.access.redhat.com/rhel7@sha256:a744ef5b58472bccfa7c606efcc6b126a164eee4b7057f85cb8be46c481ee954
 ---> db7a70a0414e
Step 2 : MAINTAINER Veer Muchandi veer@redhat.com
 ---> Running in 016986b6a4ca
 ---> 5e2ca7124536
Removing intermediate container 016986b6a4ca
Step 3 : ADD ./init.sh ./
 ---> dd2866c8e0c8
--> Finished Dependency Resolution
Dependencies Resolved
================================================================================
 Package        Arch        Version               Repository              Size
================================================================================
Installing:
 nmap-ncat      x86_64      2:6.40-7.el7          rhel-7-server-rpms      201 k
Installing for dependencies:
 libpcap        x86_64      14:1.5.3-9.el7        rhel-7-server-rpms      138 k
Transaction Summary
Transaction test succeeded
Running transaction
  Installing : 14:libpcap-1.5.3-9.el7.x86_64                                1/2
  Installing : 2:nmap-ncat-6.40-7.el7.x86_64                                2/2
  Verifying  : 2:nmap-ncat-6.40-7.el7.x86_64                                1/2
  Verifying  : 14:libpcap-1.5.3-9.el7.x86_64                                2/2
Installed:
  nmap-ncat.x86_64 2:6.40-7.el7
Dependency Installed:
  libpcap.x86_64 14:1.5.3-9.el7
Complete!
Successfully built d217d24523be
Pushing image docker-registry.default.svc:5000/mycliproject/time:latest ...
Pushed 0/5 layers, 3% complete
```

6. Now we will deployment configuration by running the following command.

```
oc get dc -o json
{
    "apiVersion": "v1",
    "items": [
        {
            "apiVersion": "v1",
            "kind": "DeploymentConfig",
            "metadata": {
                "annotations": {
                    "openshift.io/generated-by": "OpenShiftNewApp"
                },
…………
…………
…………
                "creationTimestamp": "2017-11-
10T11:22:28Z",
                "generation": 3,
                "labels": {
    "metadata": {},
    "resourceVersion": "",
    "selfLink": ""
}
```

7. Now, you can get the list of pods, Run the following command is given below.

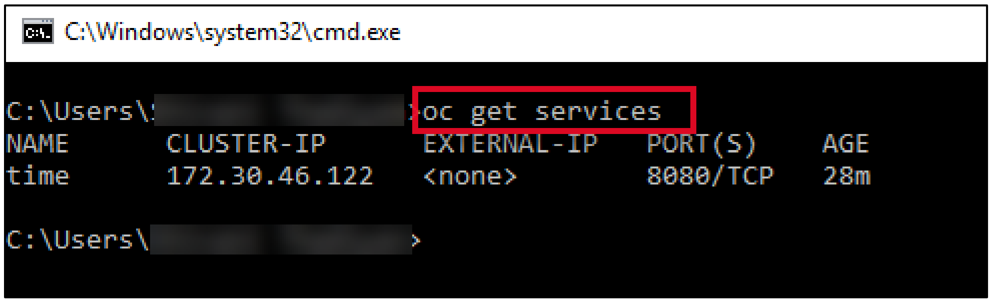
```
oc get pods
```



```
C:\Users\[redacted]>oc get pods
NAME           READY     STATUS      RESTARTS    AGE
time-1-build   0/1       Completed   0           24m
time-1-r9s99   1/1       Running     0           16m

C:\Users\[redacted]>
```

8. Now, add a route to expose that service, Run the following command is given below.

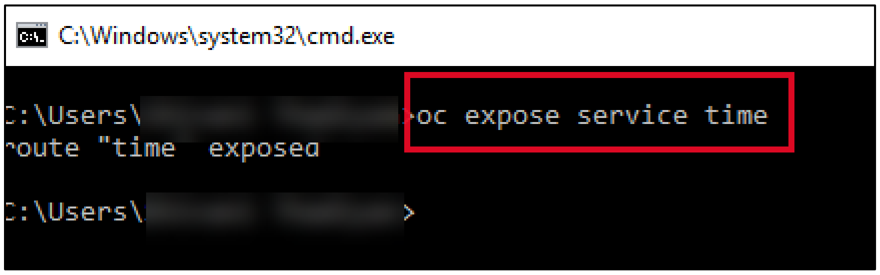
```
oc get services
```



```
C:\Users\[redacted]>oc get services
NAME     CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
time     172.30.46.122   <none>        8080/TCP   28m

C:\Users\[redacted]>
```

9. Now, we expose the service as a route.

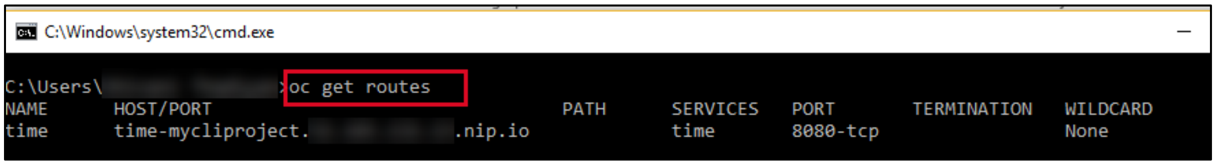
```
oc expose service time
```



```
:\Users\[redacted]>oc expose service time
oute "time" exposed

:\Users\[redacted]>
```

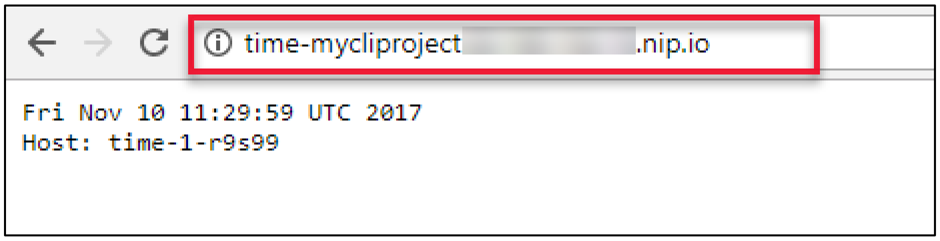
10. Now, we check the route is exposed.

```
oc get routes
```



```
C:\Users\[redacted]>oc get routes
NAME   HOST/PORT                            PATH   SERVICES   PORT       TERMINATION   WILDCARD
time   time-mycliproject.[redacted].nip.io          time       8080-tcp                 None
```

11. For run the application, copy the host/port and paste in browser and you can see the result.


time-mycliproject[redacted].nip.io

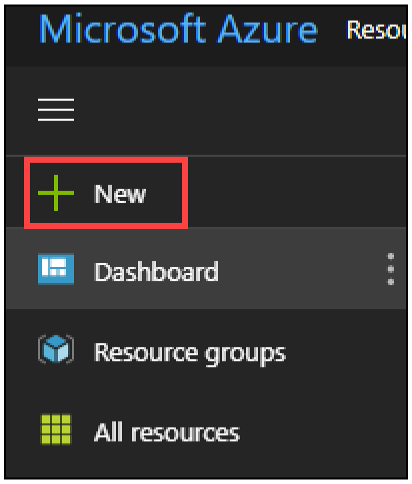
```
Fri Nov 10 11:29:59 UTC 2017
Host: time-1-r9s99
```

Lab 4: Integration of ACR with OpenShift

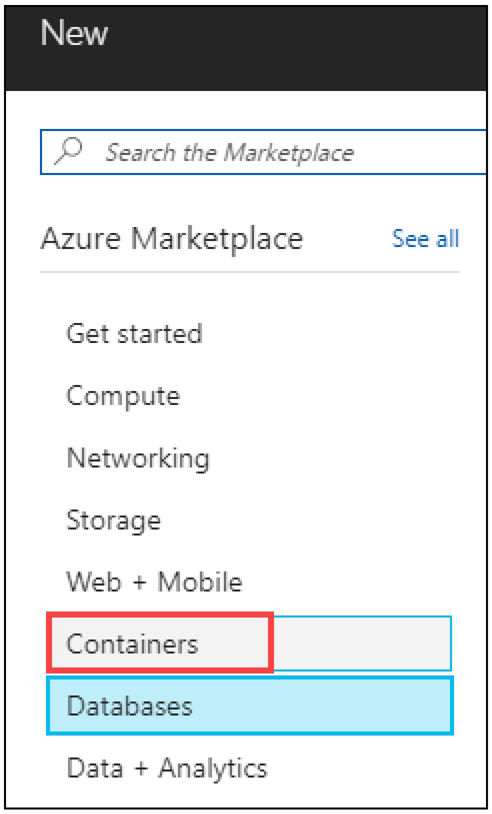
Exercise 1: Integrate ACR with OpenShift

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

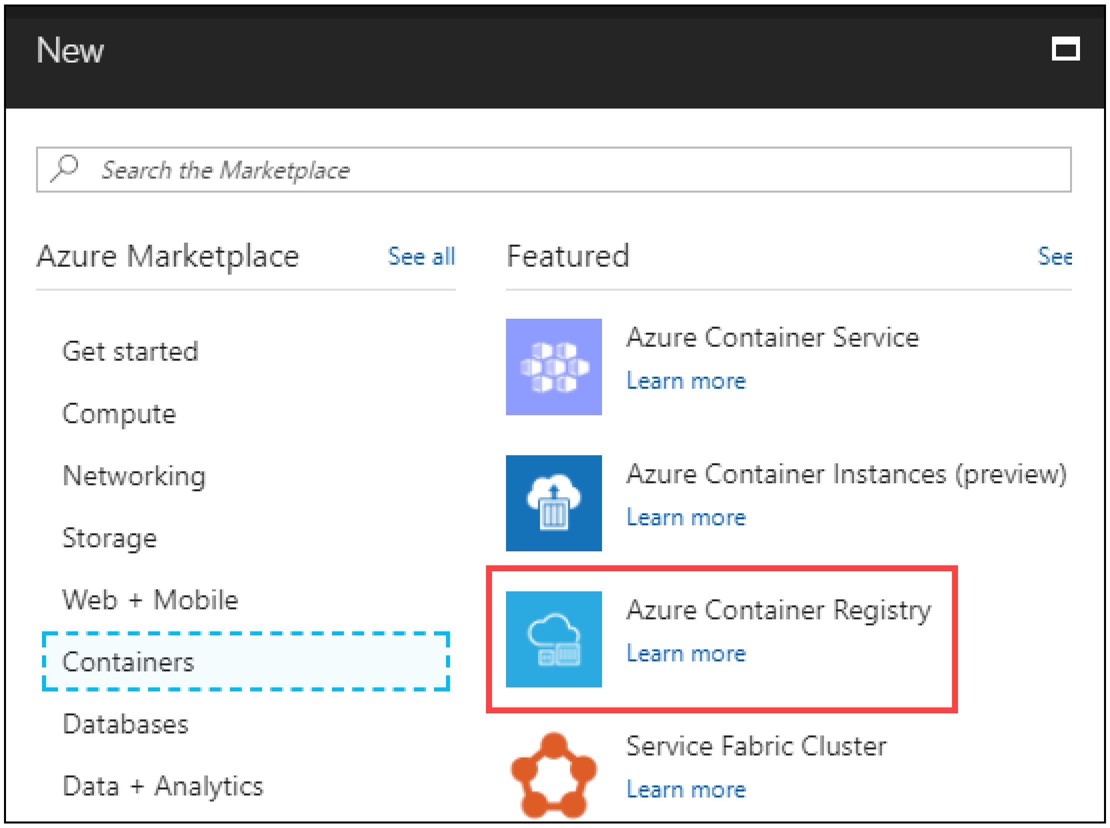
1. **Launch** a browser and **Navigate** to https://portal.azure.com. **Login** with the Microsoft Azure credentials you received via email.
2. Click on **+New** on the left side of the Dashboard.



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

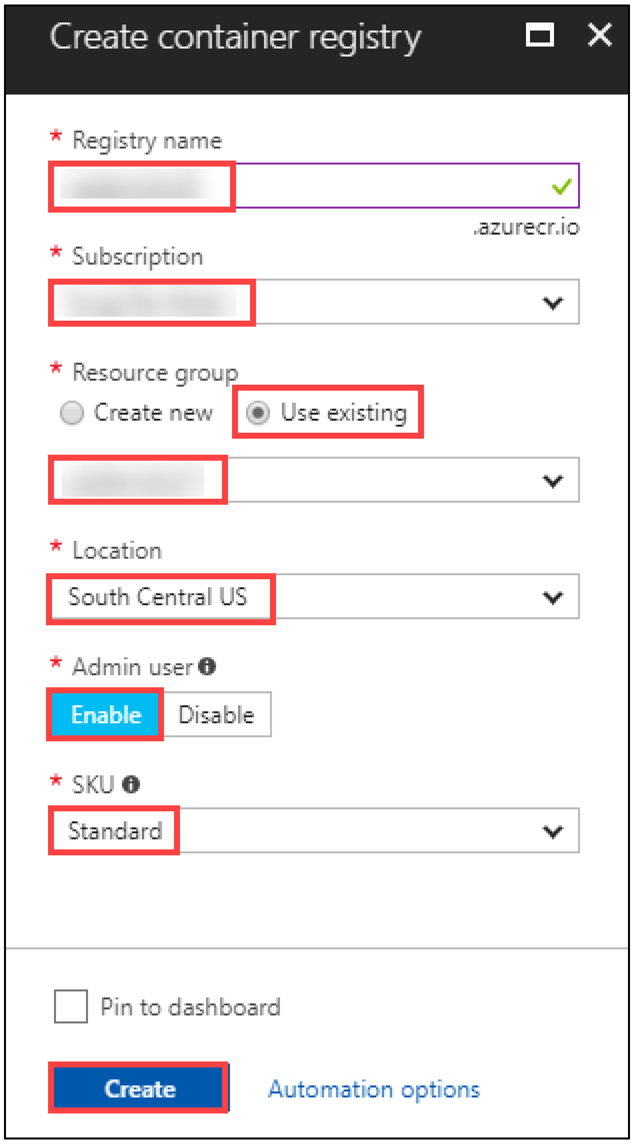


4. In the **Containers** blade appears. Select **Azure Container Registry.**



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

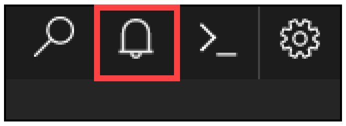
* SKU

Standard

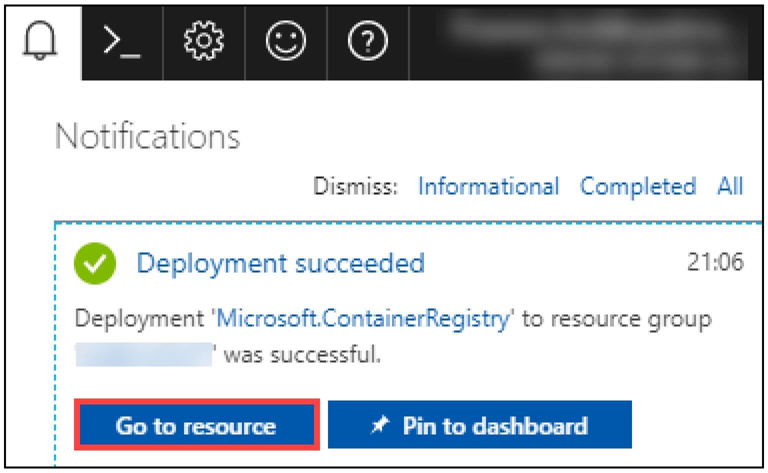
Pin to dashboard

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

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



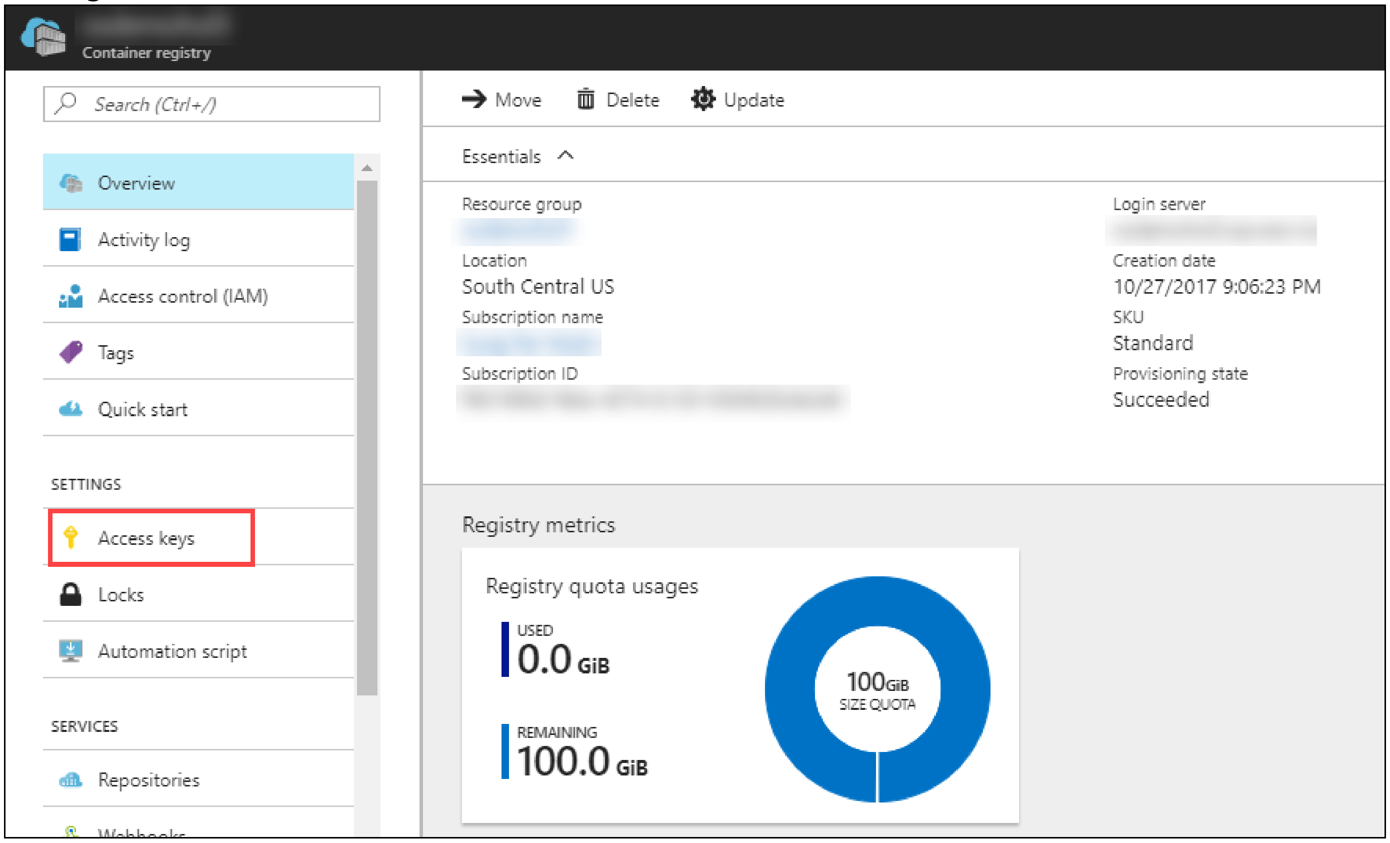
7. Once the deployment is successful, click on Go to resource from the notifications tab.



Deployment succeeded 21:06

Deployment 'Microsoft.ContainerRegistry' to resource group ' ' was successful.

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.



Resource group

Location
South Central US

Subscription name

Subscription ID

Login server

Creation date
10/27/2017 9:06:23 PM

SKU
Standard

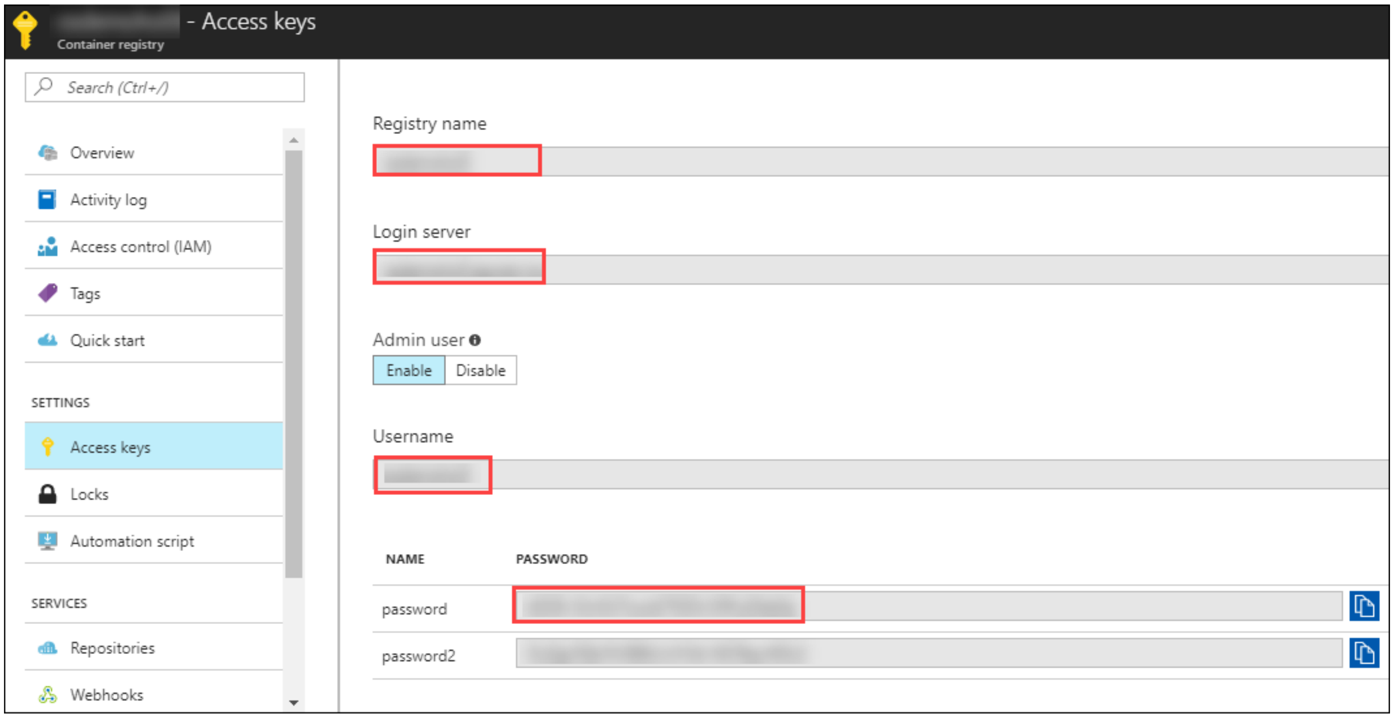
Provisioning state
Succeeded

USED
0.0 GiB

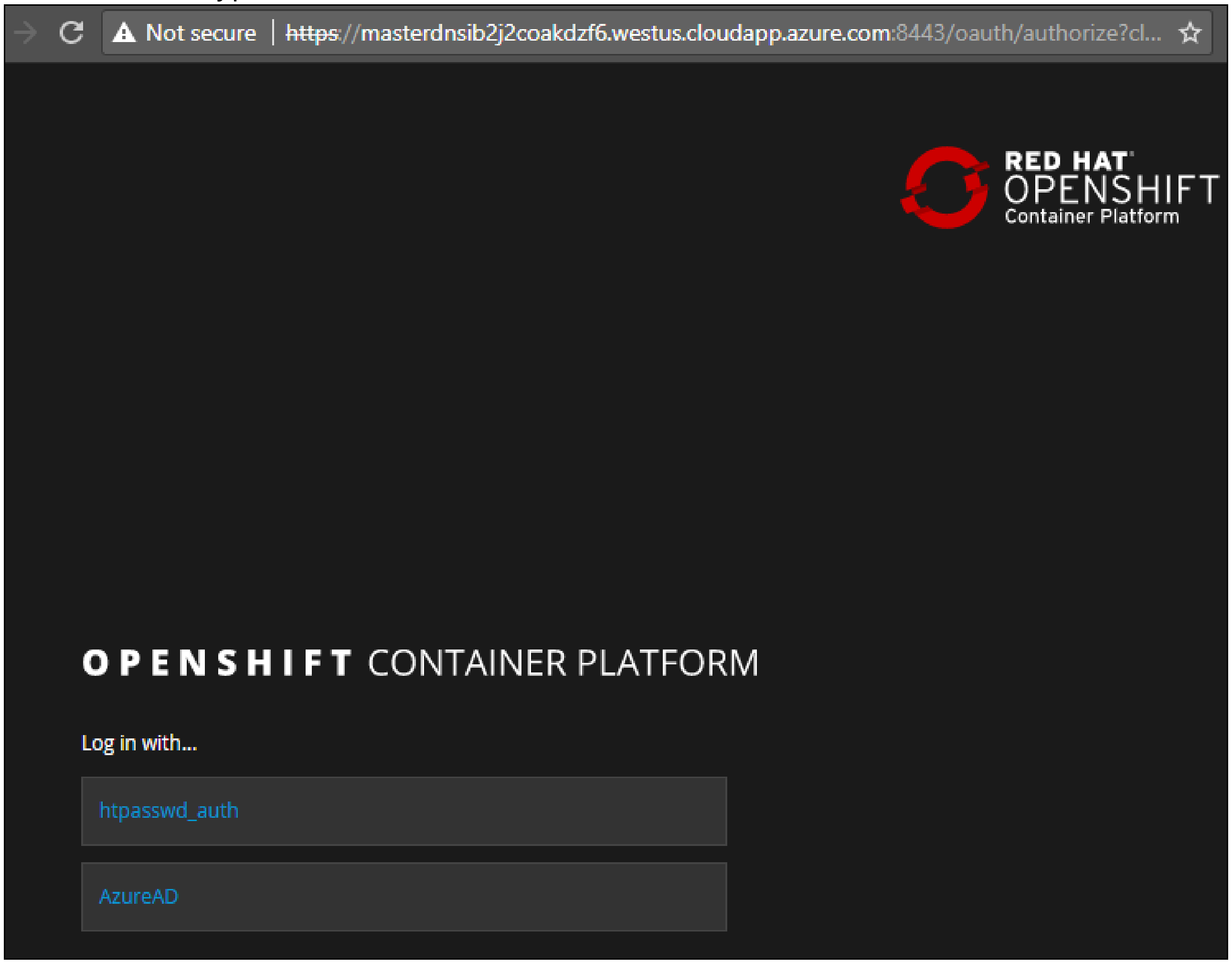
REMAINING
100.0 GiB

100GiB SIZE QUOTA

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.

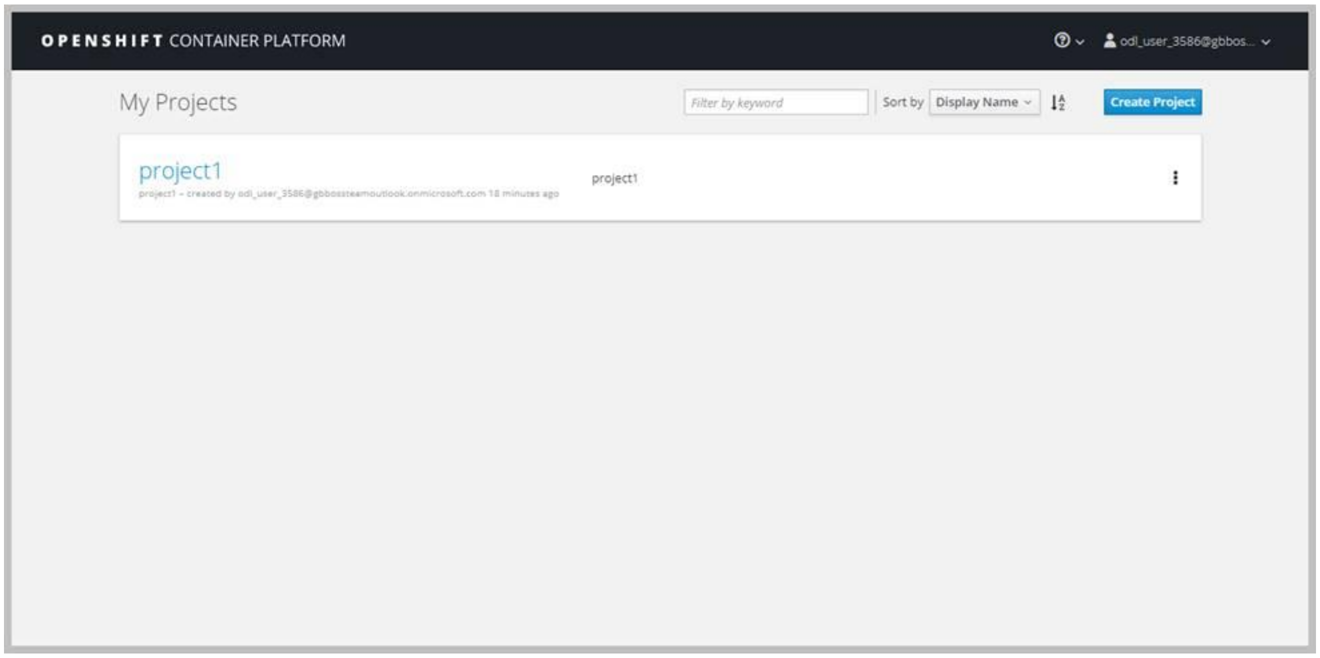


10. Now, open a new tab in a browser and navigate to the Openshift console url. Login into the Openshift console using the credentials you received via email by Selecting AzureAD as authentication type.

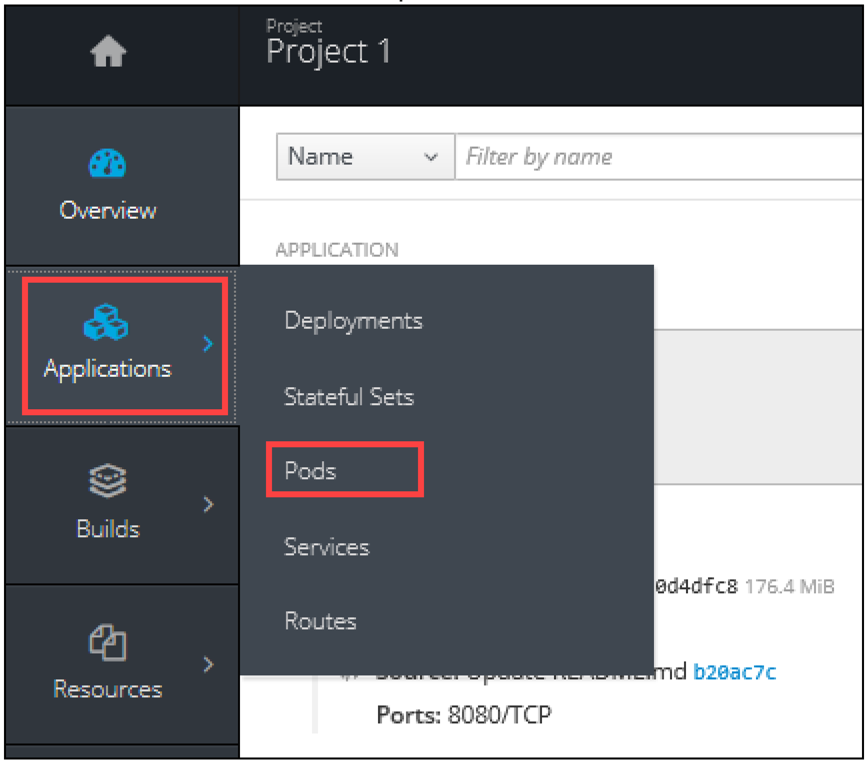


Not secure | ~~https~~://masterdnsib2j2coakdzf6.westus.cloudapp.azure.com:8443/oauth/authorize?cl...

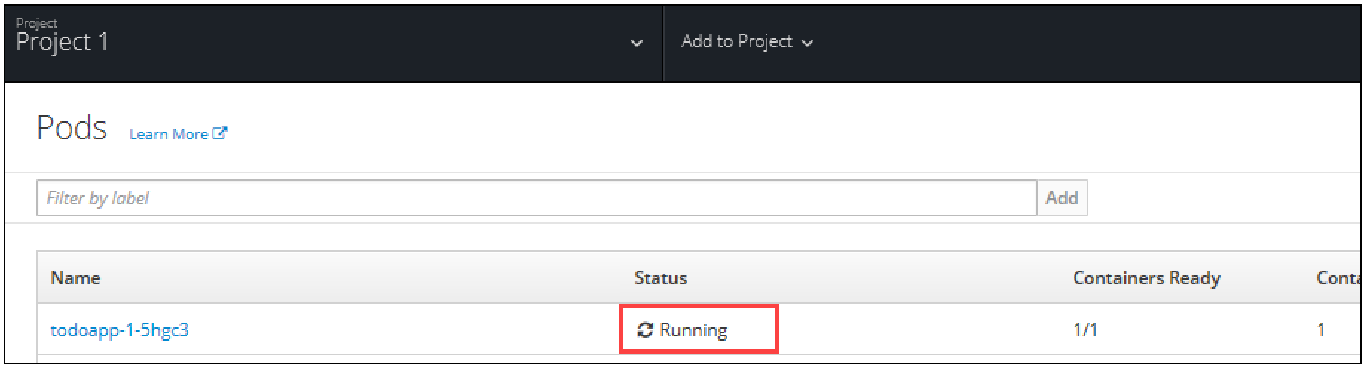
11. Now you will be redirected to the My Projects page, where you will select the Project you created earlier.


OPENSHIFT CONTAINER PLATFORM
odl_user_3586@gbbos...
My Projects
Filter by keyword
Sort by
Display Name
Create Project
project1
project1 – created by odl_user_3586@gbbossteamoutlook.onmicrosoft.com 18 minutes ago
project1

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.

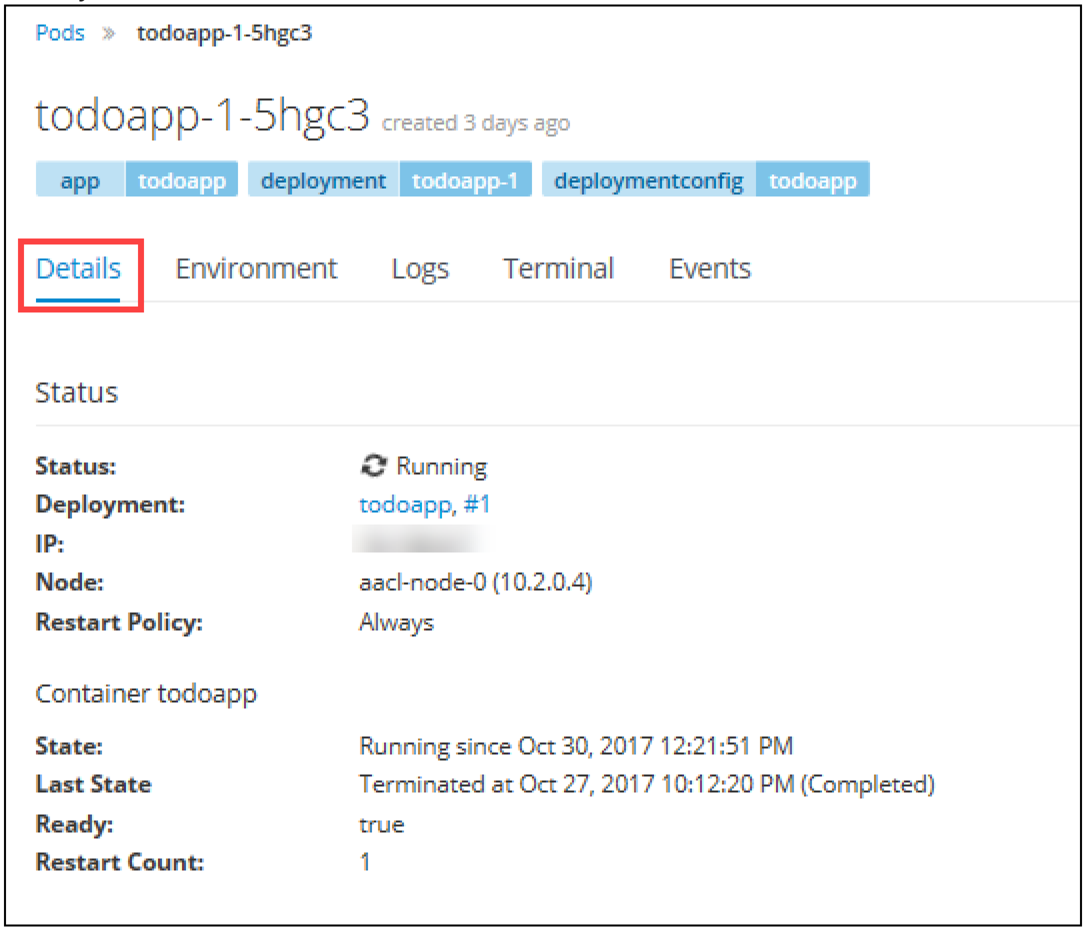

Project
Project 1
Add to Project

Pods Learn More

Filter by label Add

Name	Status	Containers Ready	Conta
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

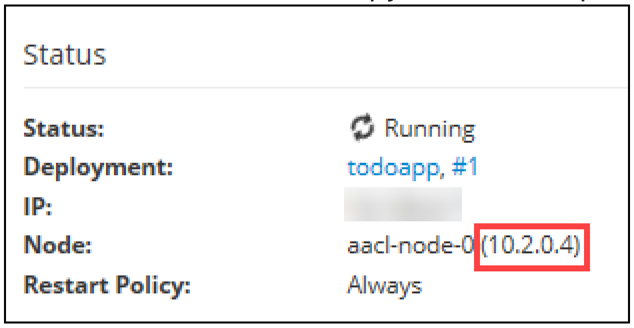
Status

Status:	Running
Deployment:	todoapp, #1
IP:	[redacted]
Node:	aacl-node-0 (10.2.0.4)
Restart Policy:	Always

Container todoapp

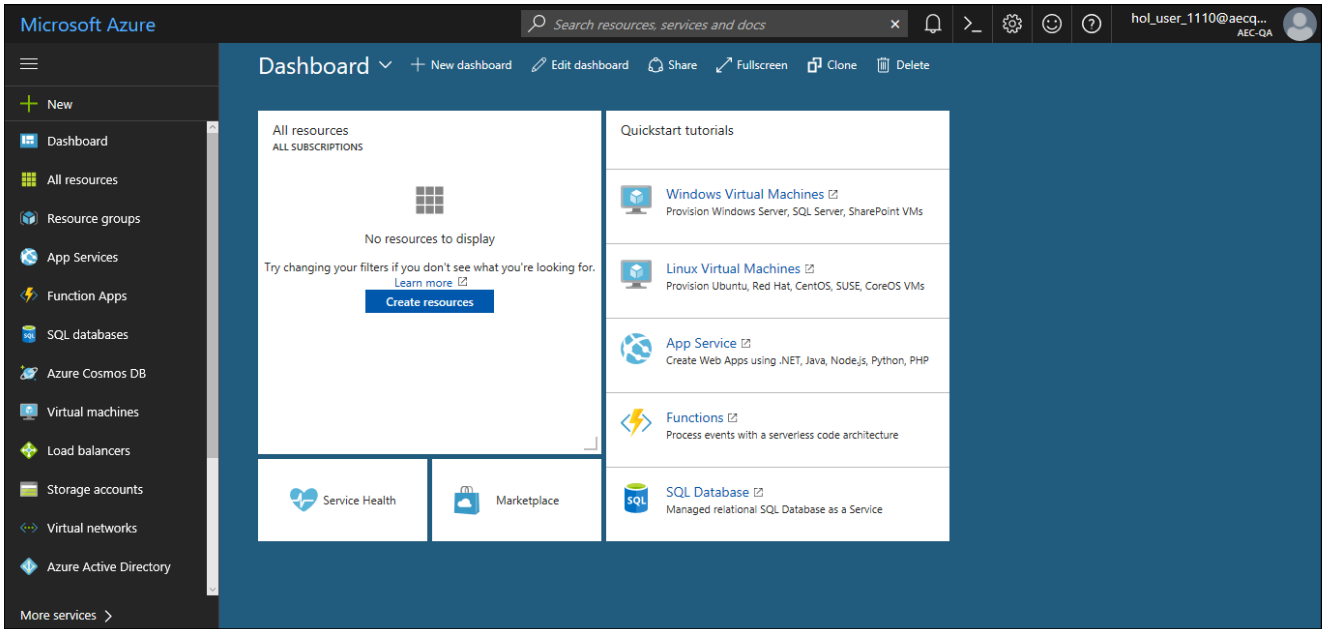
State:	Running since Oct 30, 2017 12:21:51 PM
Last State	Terminated at Oct 27, 2017 10:12:20 PM (Completed)
Ready:	true
Restart Count:	1

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

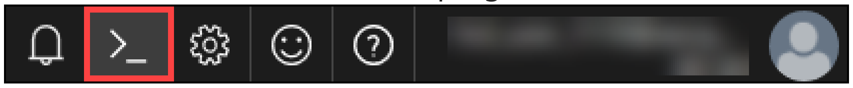

Status

Status:	Running
Deployment:	todoapp, #1
IP:	[redacted]
Node:	aacl-node-0 (10.2.0.4)
Restart Policy:	Always

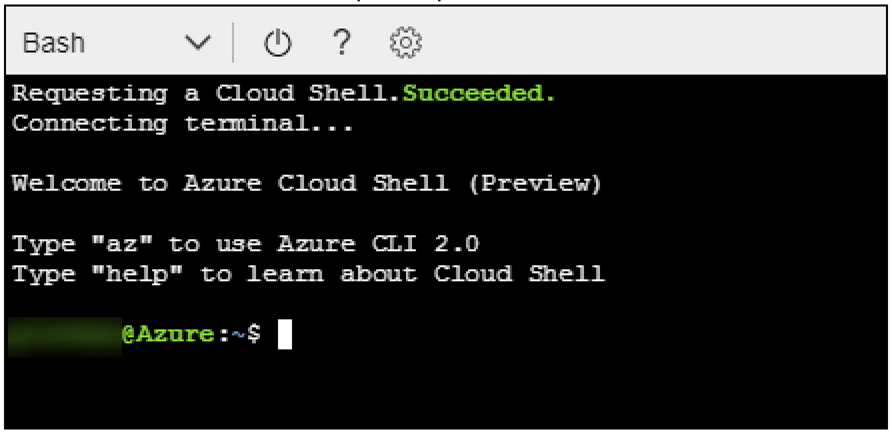
10. Now, **Open** a new tab in a browser and **Navigate** to https://portal.azure.com. **Login** with the Microsoft Azure credentials you received via email.



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.

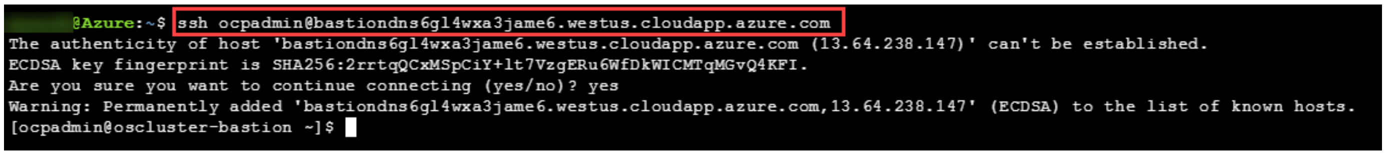


```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell (Preview)

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

[redacted]@Azure:~$
```

17. Now execute the following command. When promted, 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

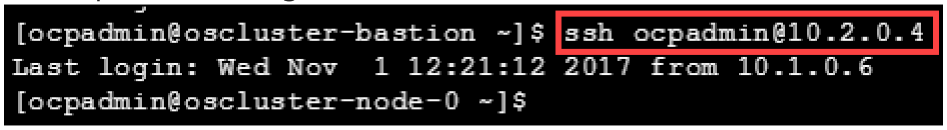


```
[redacted]@Azure:~$ ssh ocpadmin@bastiondns6gl4wxa3jame6.westus.cloudapp.azure.com
The authenticity of host 'bastiondns6gl4wxa3jame6.westus.cloudapp.azure.com (13.64.238.147)' can't be established.
ECDSA key fingerprint is SHA256:2rrtqQCxMSpCiY+lt7VzgERu6WfDkWICMTqMGvQ4KFI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'bastiondns6gl4wxa3jame6.westus.cloudapp.azure.com,13.64.238.147' (ECDSA) to the list of known hosts.
[ocpadmin@oscluster-bastion ~]$
```

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>
```

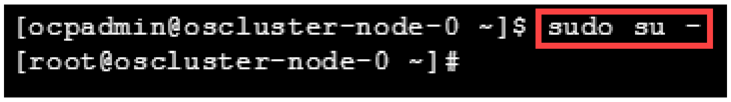
Note: Substitute in the above command with the value of copied Private IP Address of Node in which pod is running.



```
[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.

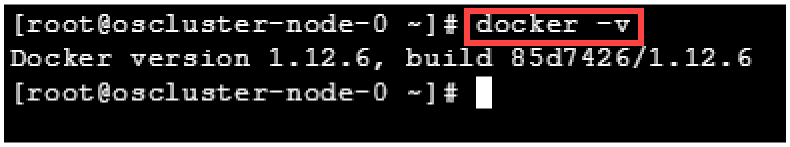
```
sudo su -
```



```
[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.

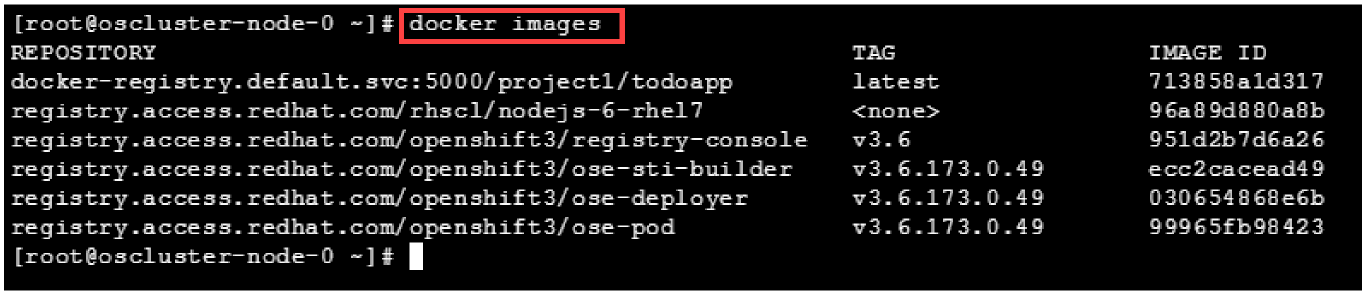
```
docker -v
```



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

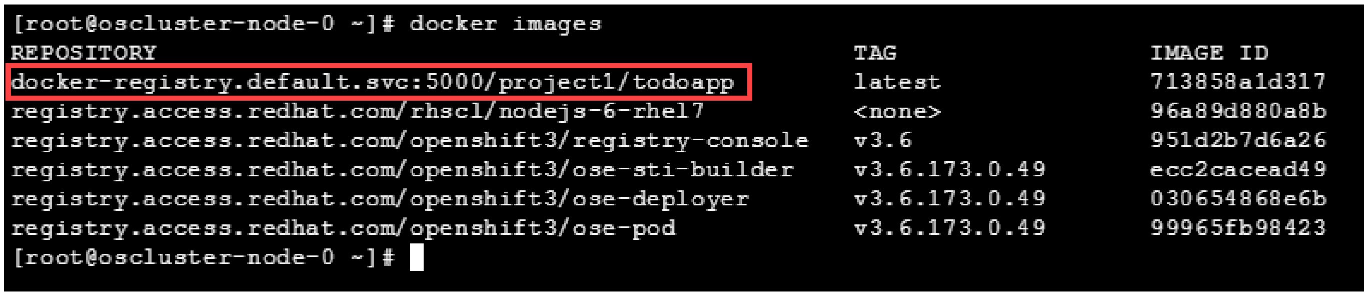
21. Now execute the following command in the cloud shell to display the list the docker images in the system.

```
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/rhscl/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 ~]#
```

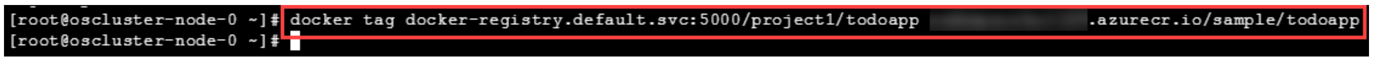
22. From the displayed results, copy the Image name with todoapp in the end.



```
[root@oscluster-node-0 ~]# docker images
REPOSITORY                                           TAG               IMAGE ID
docker-registry.default.svc:5000/project1/todoapp    latest            713858a1d317
registry.access.redhat.com/rhscl/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
```



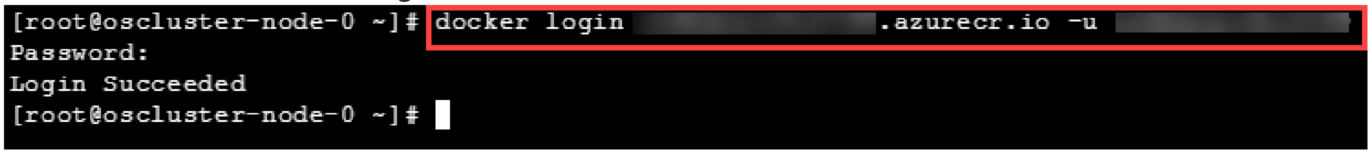
```
[root@oscluster-node-0 ~]# docker tag docker-registry.default.svc:5000/project1/todoapp [redacted].azurecr.io/sample/todoapp
[root@oscluster-node-0 ~]#
```

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

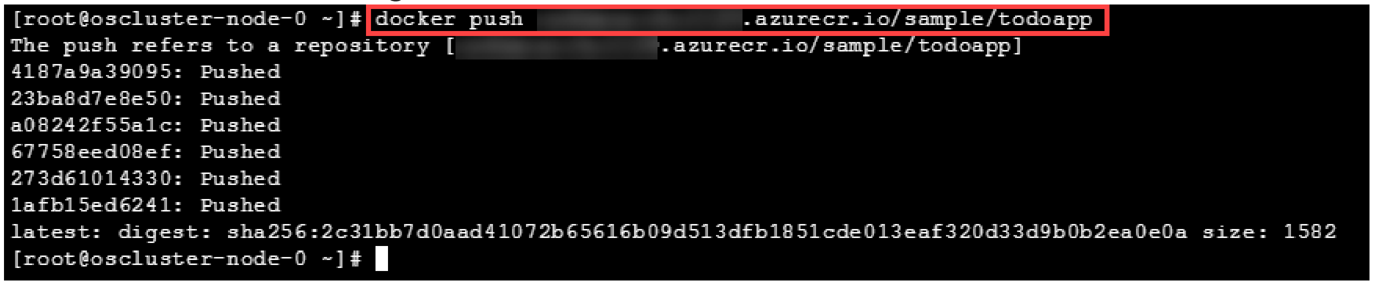


```
[root@oscluster-node-0 ~]# docker login [redacted].azurecr.io -u [redacted]
Password:
Login Succeeded
[root@oscluster-node-0 ~]#
```

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.

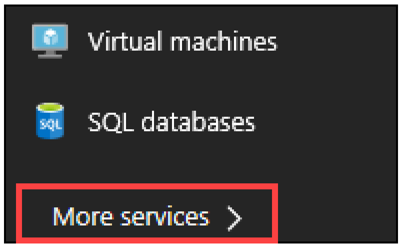
```
docker push <ACRLoginServerUri>/sample/todoapp
```

Note: Substitute for ACRLoginServerUri in the above command

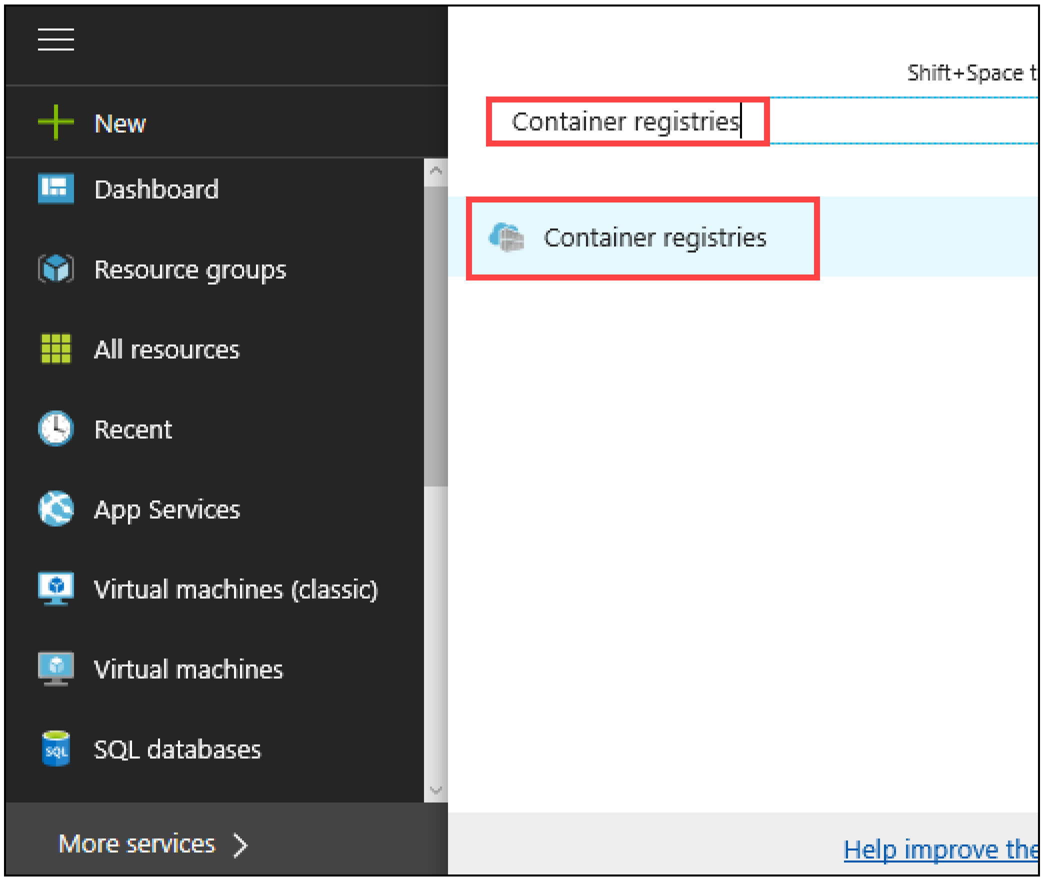


```
[root@oscluster-node-0 ~]# docker push [redacted].azurecr.io/sample/todoapp
The push refers to a repository [[redacted].azurecr.io/sample/todoapp]
4187a9a39095: Pushed
23ba8d7e8e50: Pushed
a08242f55a1c: Pushed
67758eed08ef: Pushed
273d61014330: Pushed
1afb15ed6241: Pushed
latest: digest: sha256:2c31bb7d0aad41072b65616b09d513dfb1851cde013eaf320d33d9b0b2ea0e0a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

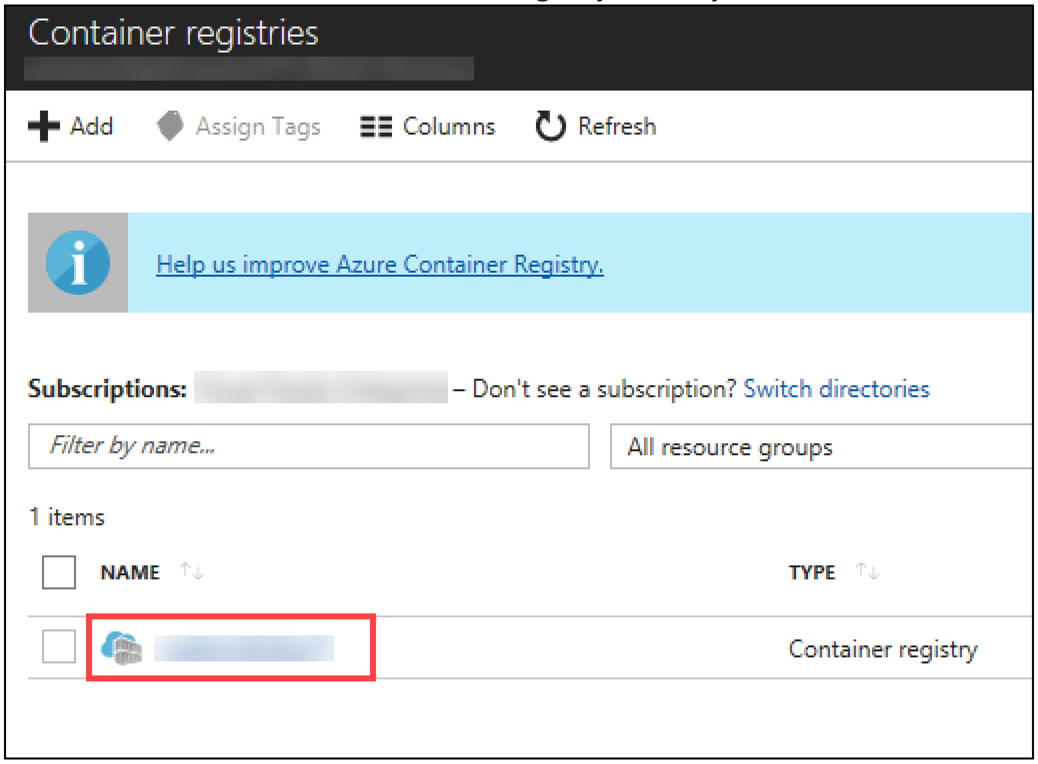
26. Once you have pushed the image to Azure Container Registry, **click** on **More services** on the left side of the menu on the dashboard.



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.



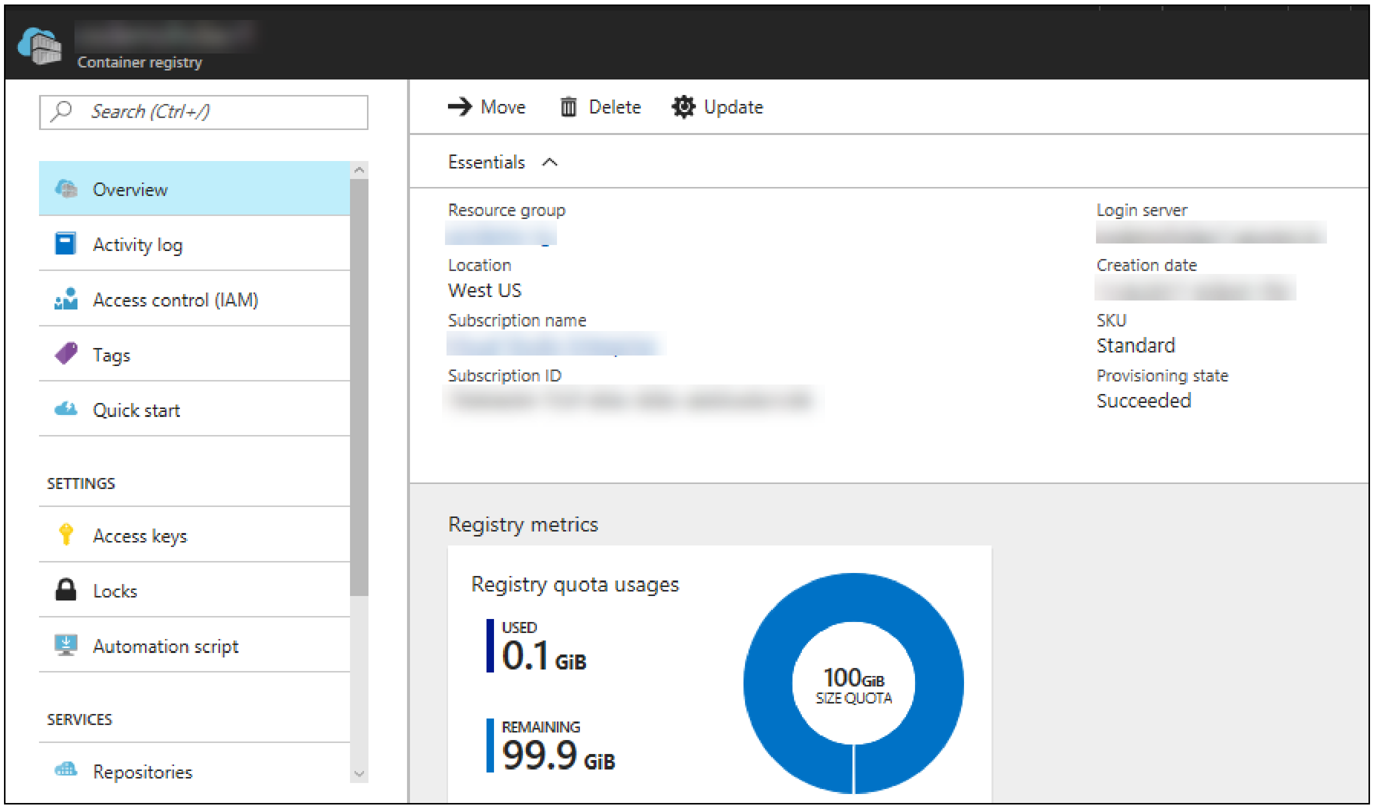
Subscriptions: [redacted] – Don't see a subscription? Switch directories

Filter by name... | All resource groups

1 items

NAME	TYPE
[redacted]	Container registry

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



Resource group
[redacted]

Location
West US

Subscription name
[redacted]

Subscription ID
[redacted]

Login server
[redacted]

Creation date
[redacted]

SKU
Standard

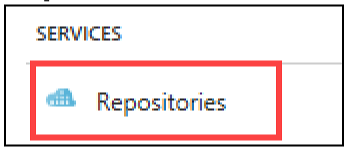
Provisioning state
Succeeded

USED
0.1 GiB

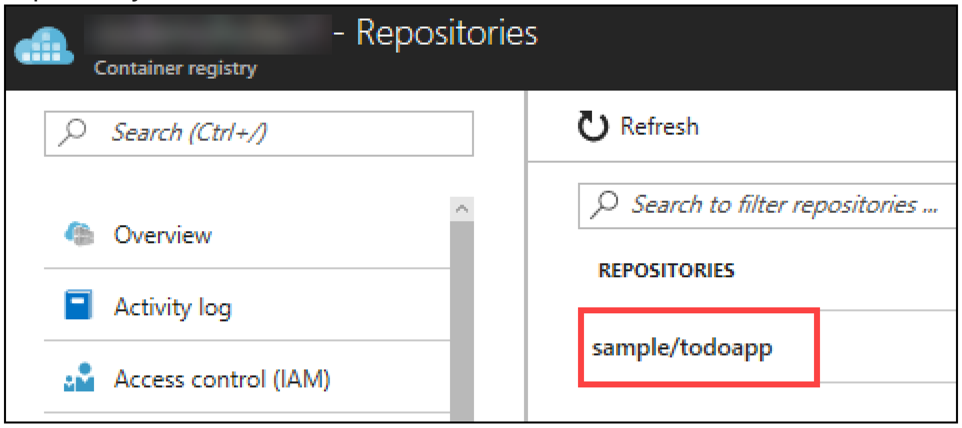
REMAINING
99.9 GiB

100GiB SIZE QUOTA

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.



[redacted] - Repositories
Container registry
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Refresh
Search to filter repositories ...
REPOSITORIES
sample/todoapp