Two-tier App migration on Azure Container: Lab Guide

Overview

This guide will help you in migrating On-premises two-tier WebApp with database to Windows Container and PaaS database on Azure.

Content

Lab 1: Getting Started with Azure

Lab Overview

In this lab, you will be deploying pre-requisite infrastructure which is simulation of on-premises two-tier webapp with MS SQL database.

Prerequisites

- Windows or a Mac machine with HTML5 supported browser such as Microsoft Edge, Internet Explorer, Chrome or Firefox.

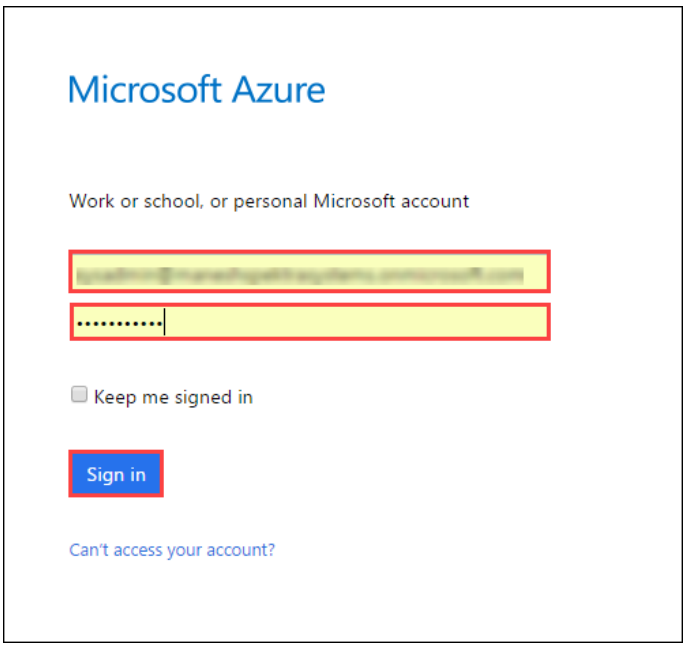
Time Estimate

20 minutes

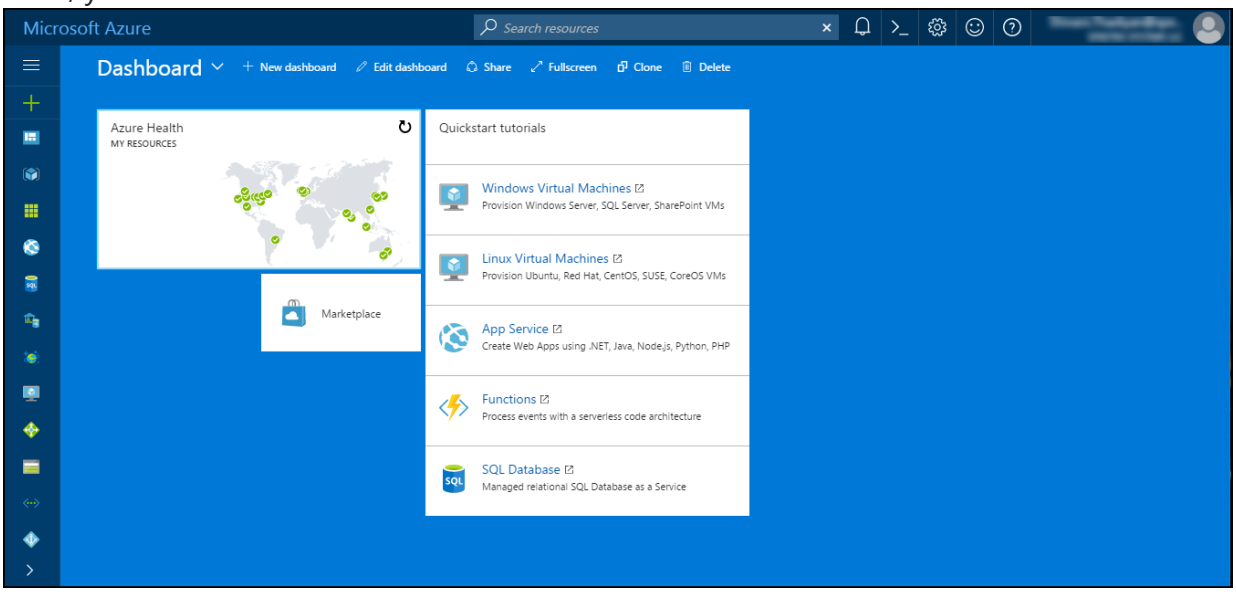
Exercise 1: Deploy Pre-requisite ON-preminses template

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 you Azure login credentials and click on **Sign In**.



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



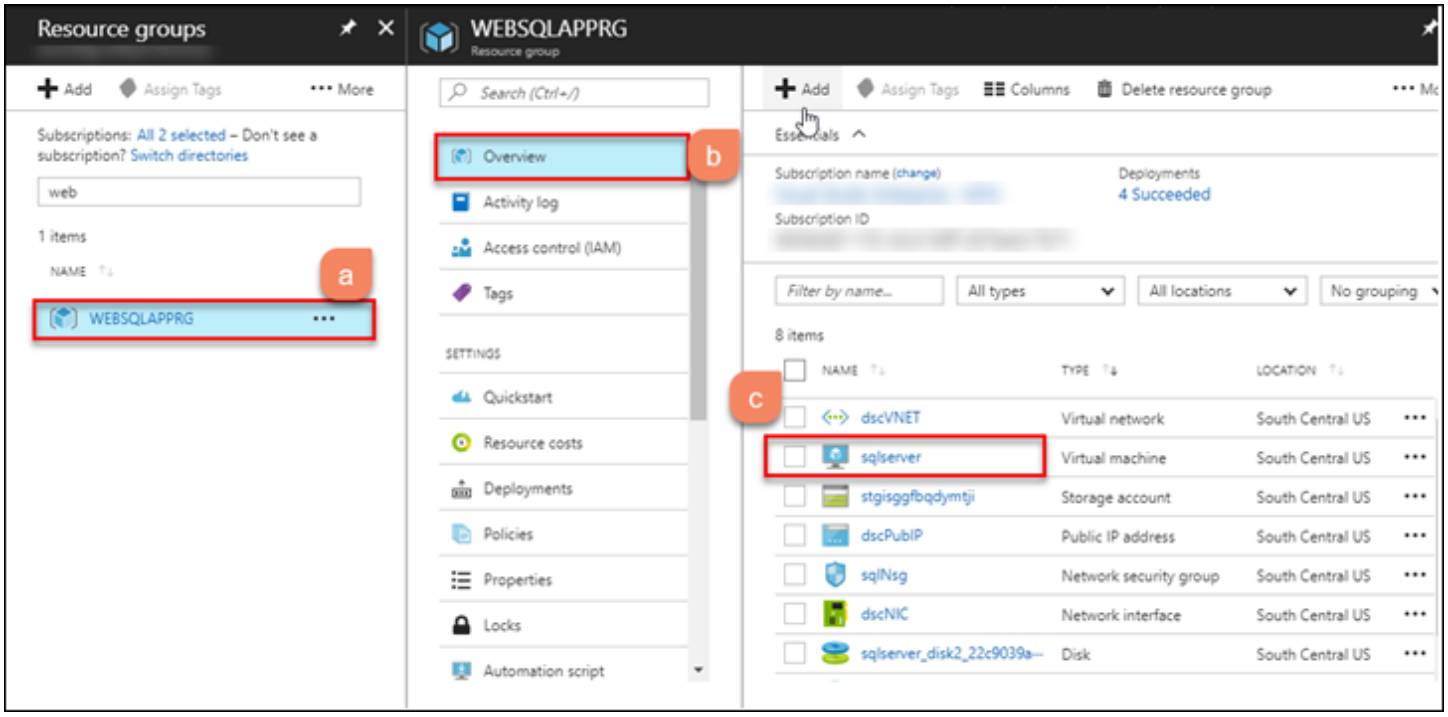
3. Open github url (https://github.com/wmhussain/two-tier-app-migration-containers) and click on deploy to azure to create pre-requiste simulation of on-premises two tier infrastructure.
    a. *ResourceGroup:* **OnPremisesRG (**any valid name**)**
    b. *Location:* **East US**(any location)
    c. *Virtual Machine Name:* **WebDbServer** (Any valid name)
    d. *Admin User:* **demouser** (any username of your choice)
    e. *Admin Password:* *<Valid Password>*

    Accepts terms and conditions and click on **Purchase** button.

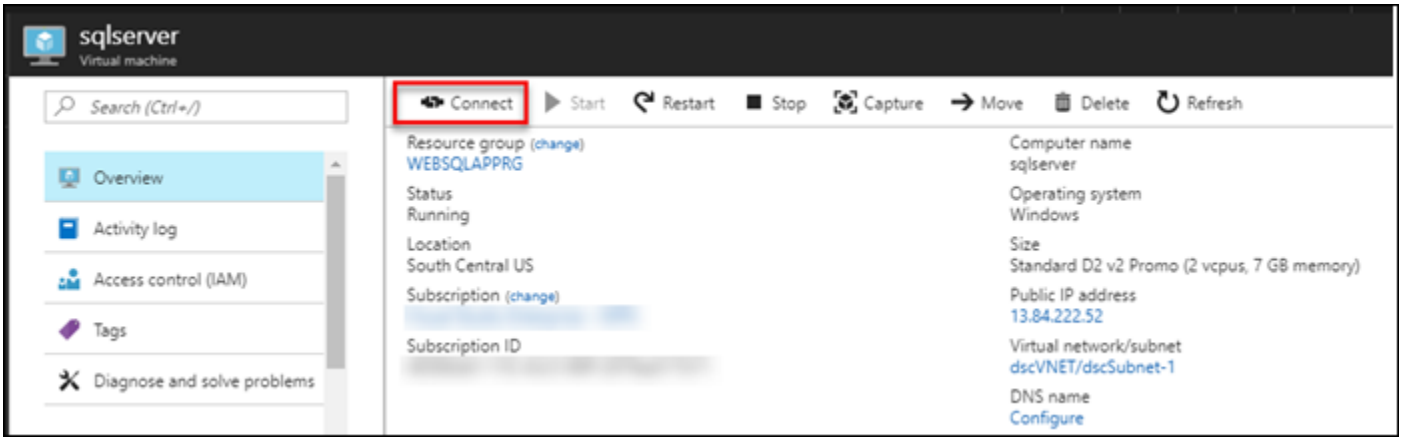
Exercise 2: Verify the DB and WebApp

In this exercise, we will login to VM which is pre-deployed as a part of lab. Login to VM with the credentials provided in mail.

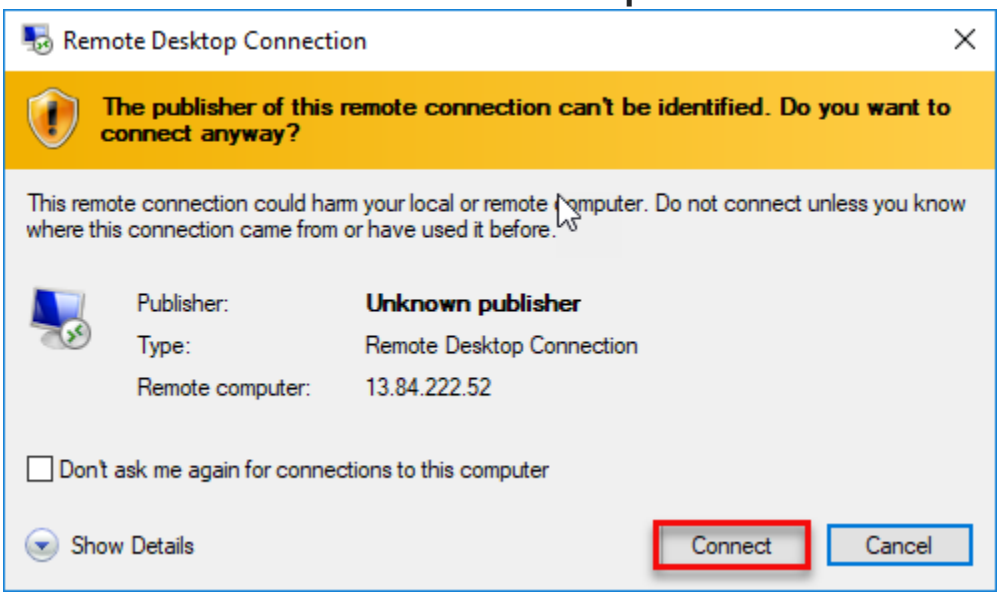
1. In Azure portal, click on Resource Group which contains the pre-deployed on-premises infrastructure then click on **Overview** tab and finally on VM.



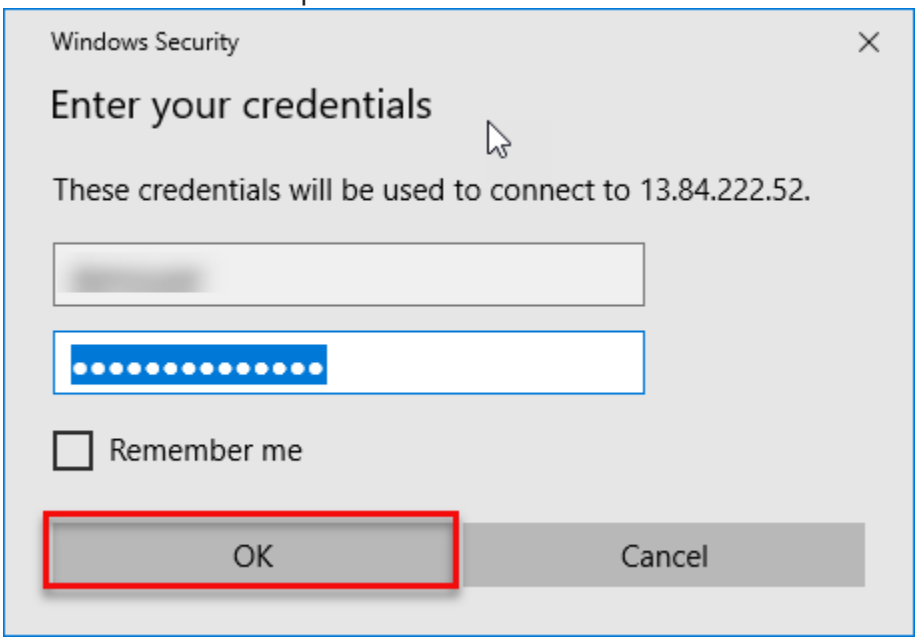
2. In overview section, click on **Connect** button. It will download an RDP file. Click on the same to connect the VM.



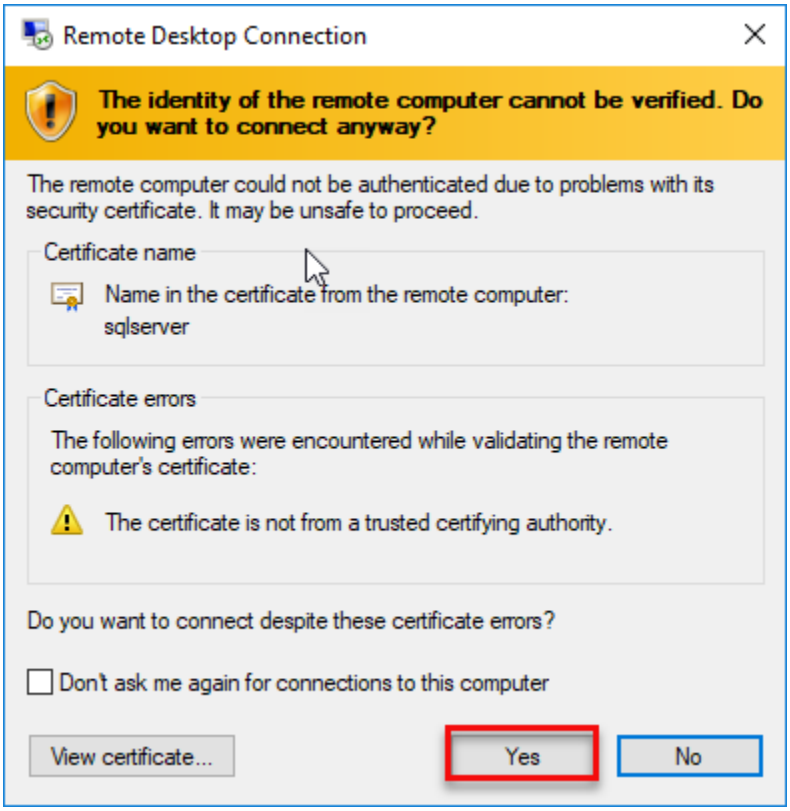
3. Click on **Connect** button in **Remote Desktop Connection** Wizard.



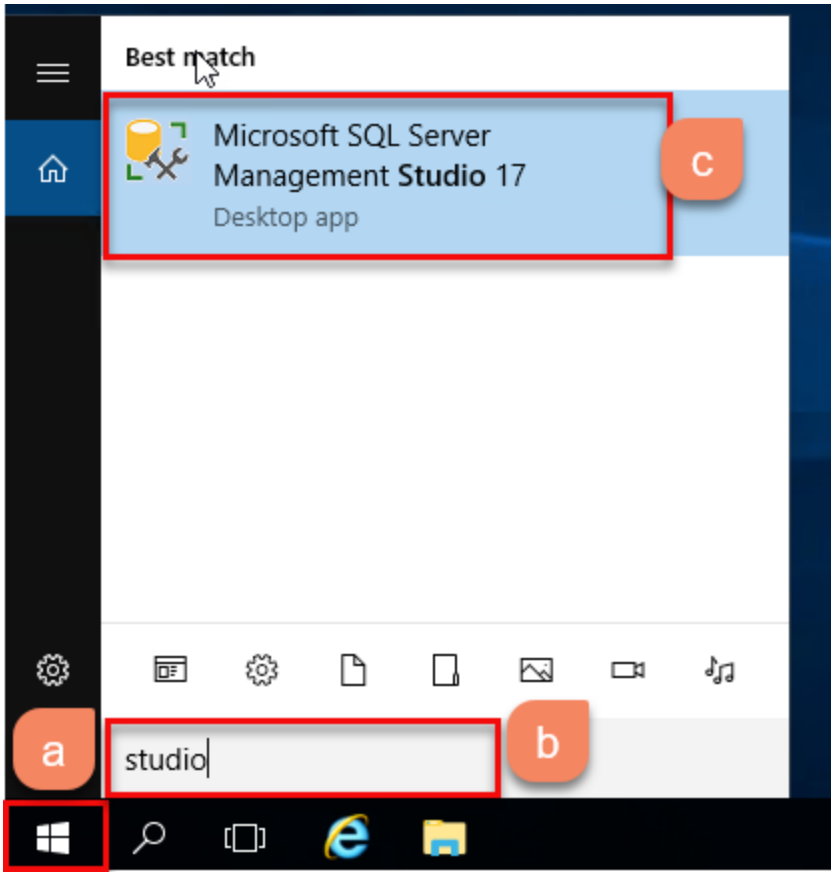
4. Enter the credentials provided in the mail to connect to VM.



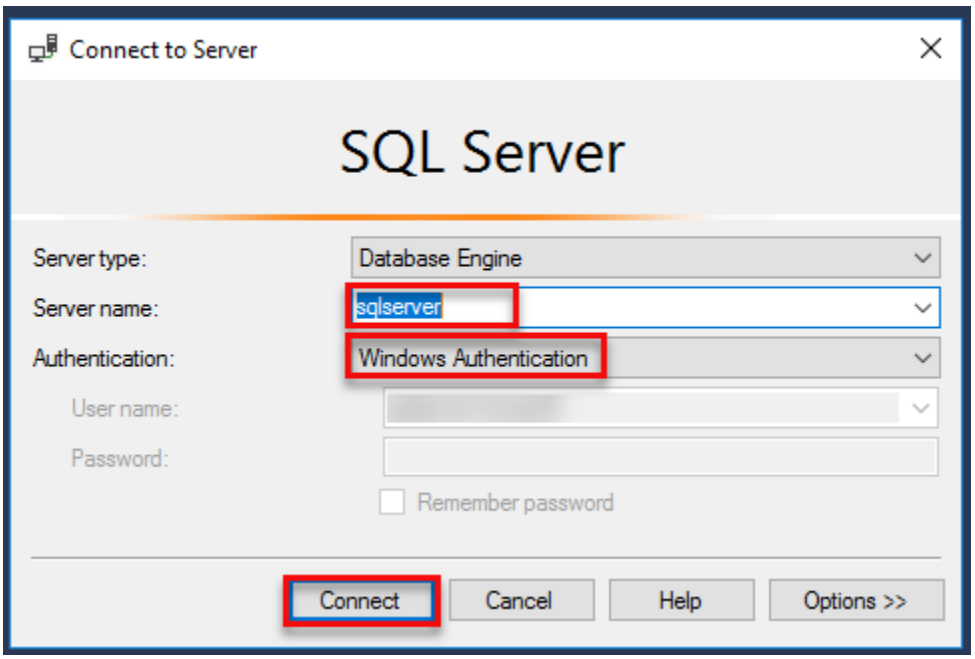
5. Click on **Yes** button.



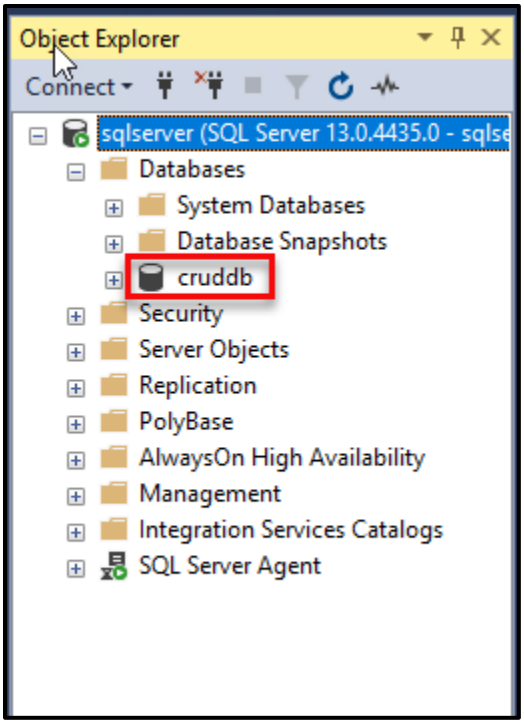
6. Once you are logged in the Windows OS, click on **start** button and search for '**Microsoft SQL Server management Studio**' and click on that as shown below.



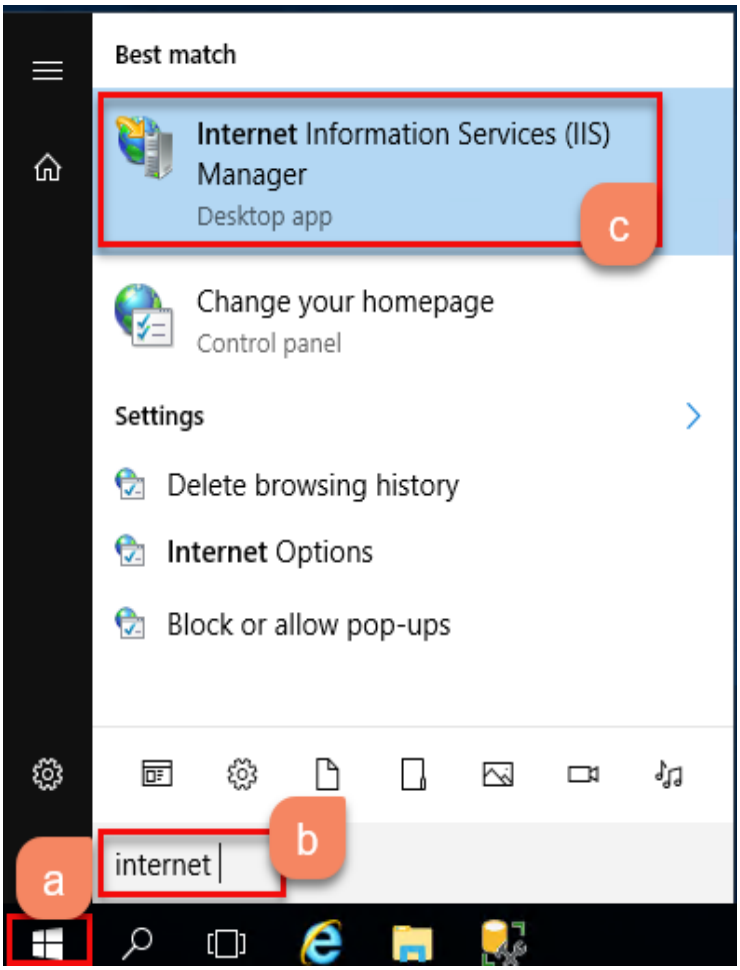
7. In the Connect to Server wizard, verify that Server name is set to *VM's name* and Authentication is **Windows Authentication**. Click **Connect** button to connect to database.



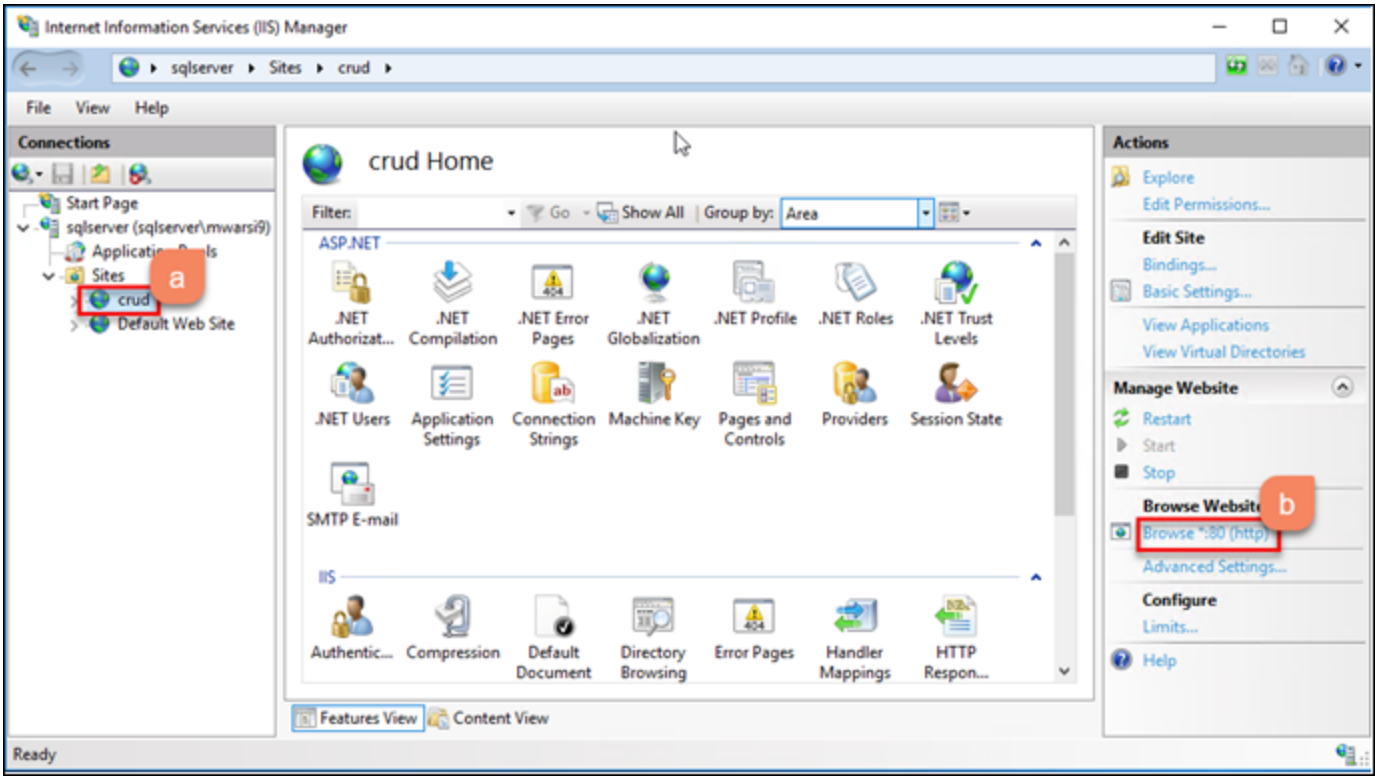
8. Once you are connected to database, you can verify that database named **cruddb** exists. Minimise the the studio.



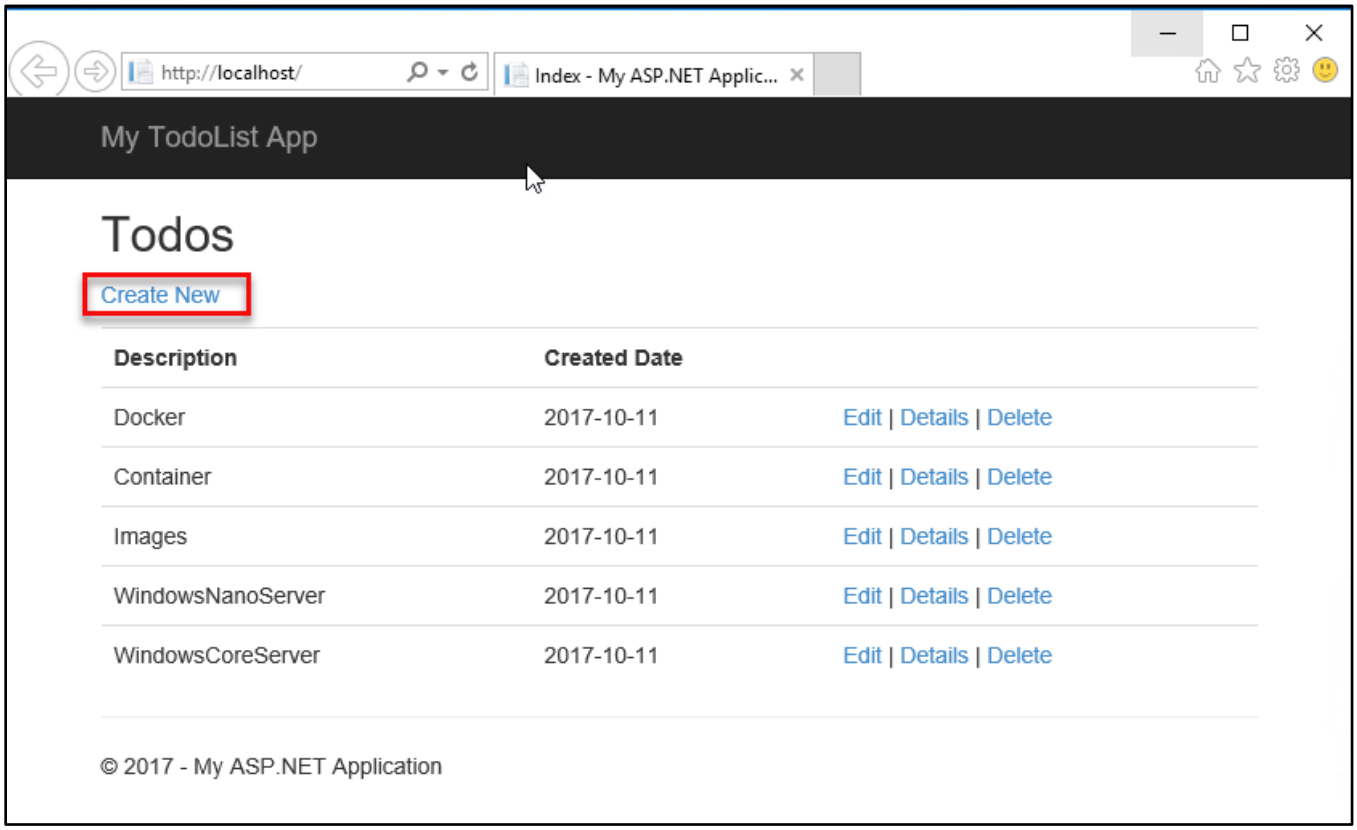
9. Click on **Start** button and search for *Internet Information Manager* and open the application.



10. In IIS Manager, expand the server and then Sites, you will see a site named as **crud.** Click on that site. And then click on browse site as shown below.



11. A web application will get open in IE. Create some record in that application as shown below.



12. This completes the verification of our on-premises two-tier web application. You can also verify the records in SQL database manually (Optional).

Lab 2: Two-tier WebApp migration to Container on Azure.

Lab Overview

In this lab, you will

- Verify the prerequisite Environment which is simulating the on-premises two-tier webapp.
- Verify the database and web-application.

Prerequisites

- Windows or a Mac machine with HTML5 supported browser such as Microsoft Edge, Internet Explorer, Chrome or Firefox.
- Lab 1 must be completed.

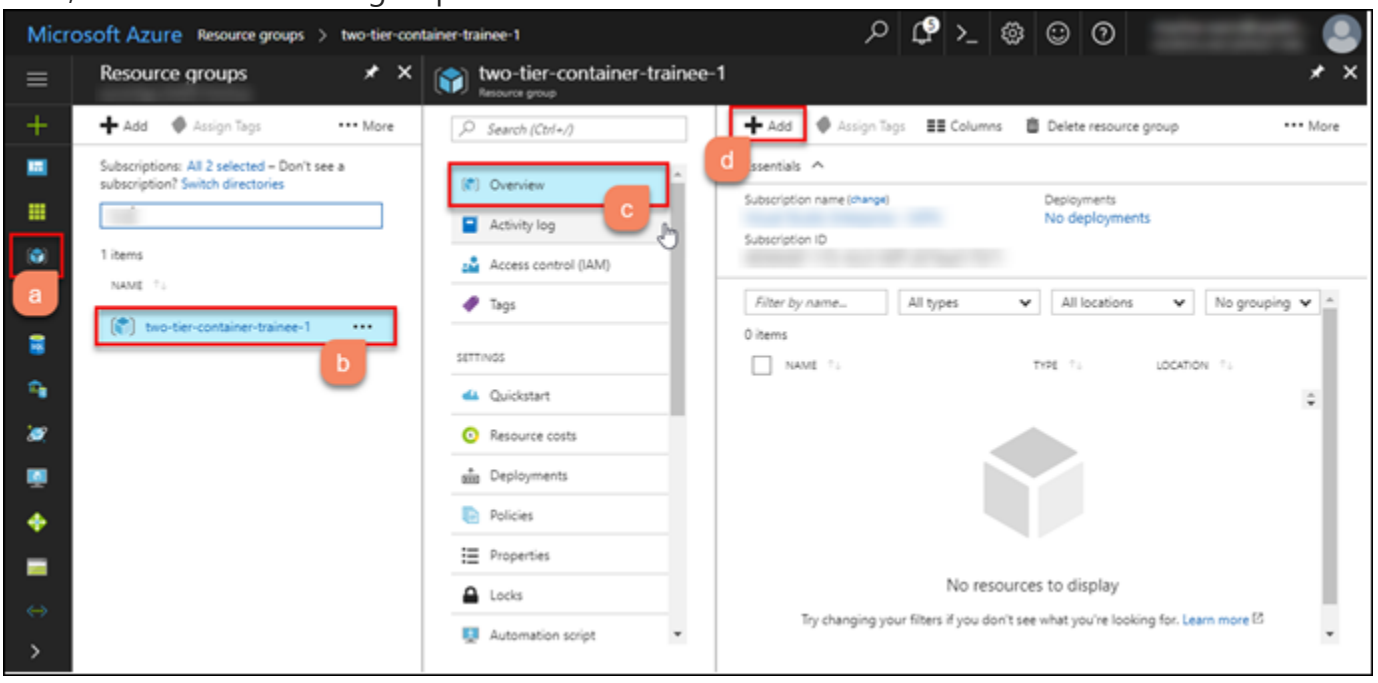
Time Estimate

20 minutes

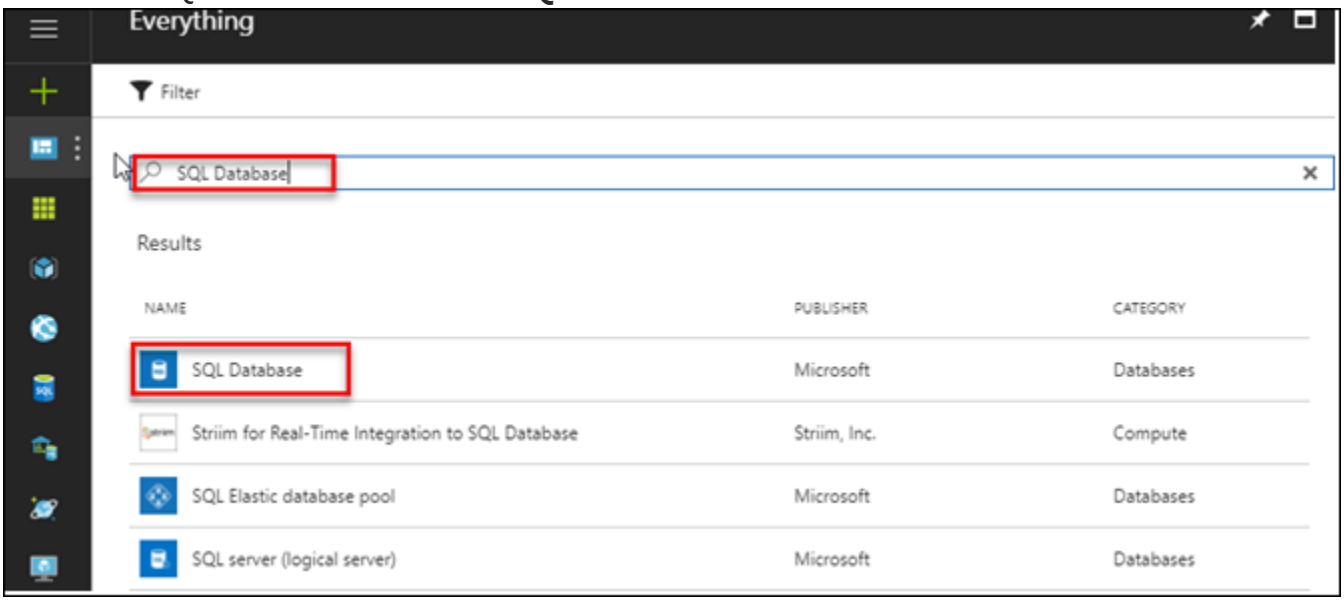
Exercise 1: Deploy Azure SQL DB for migration

In this exercise, we will deploy Azure SQL database which is required for migration of on-premises database to Azure.

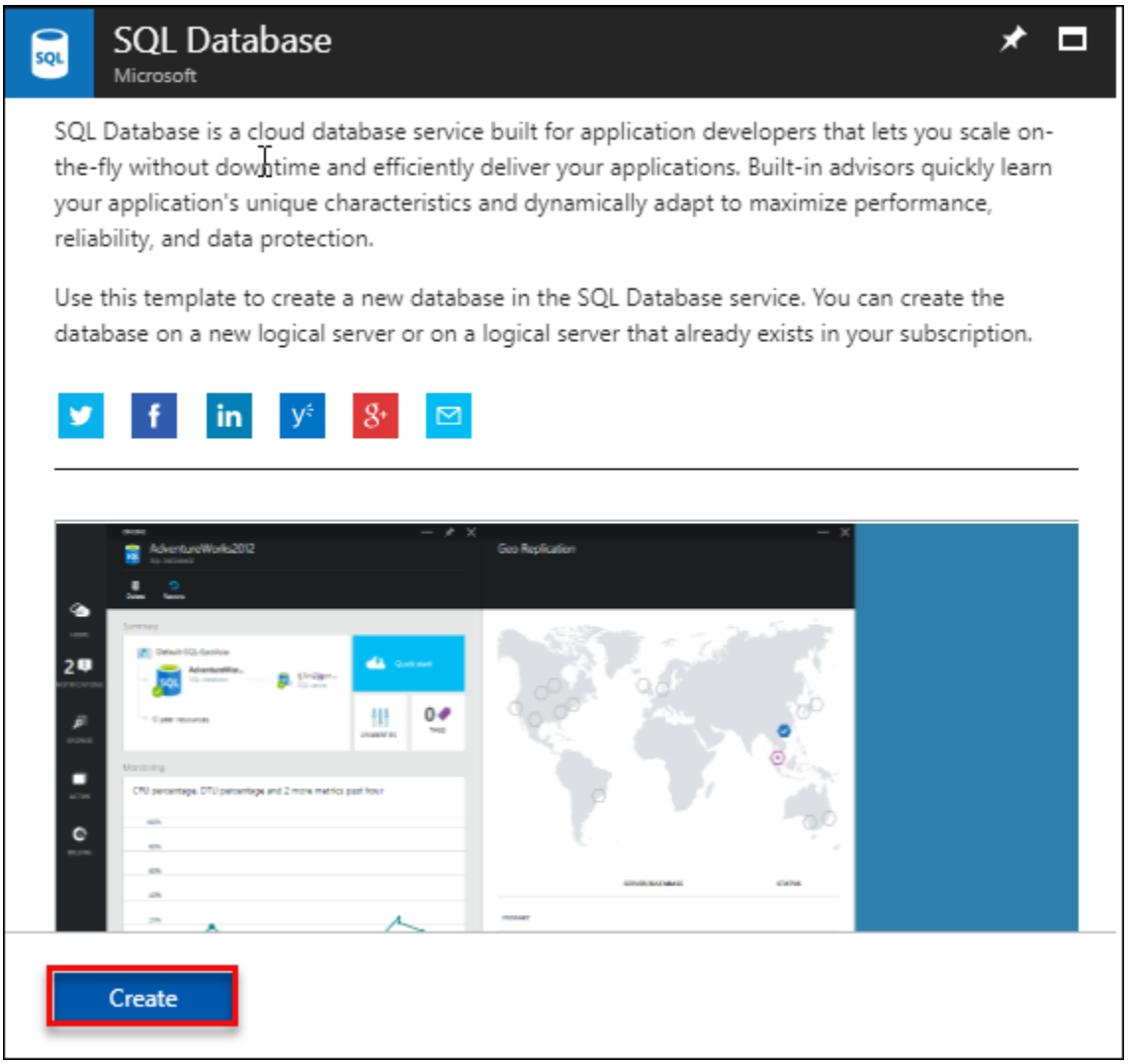
1. Open Azure portal, click on Resource group icon on the left, create a resource group.
    a. Resource Group Name : **two-tier-container-trainee-1**(any valid name)
    b. Location: **East US**(any valid location)
2. Now, select that resource group. Under the **Overview** section click on **Add** button.



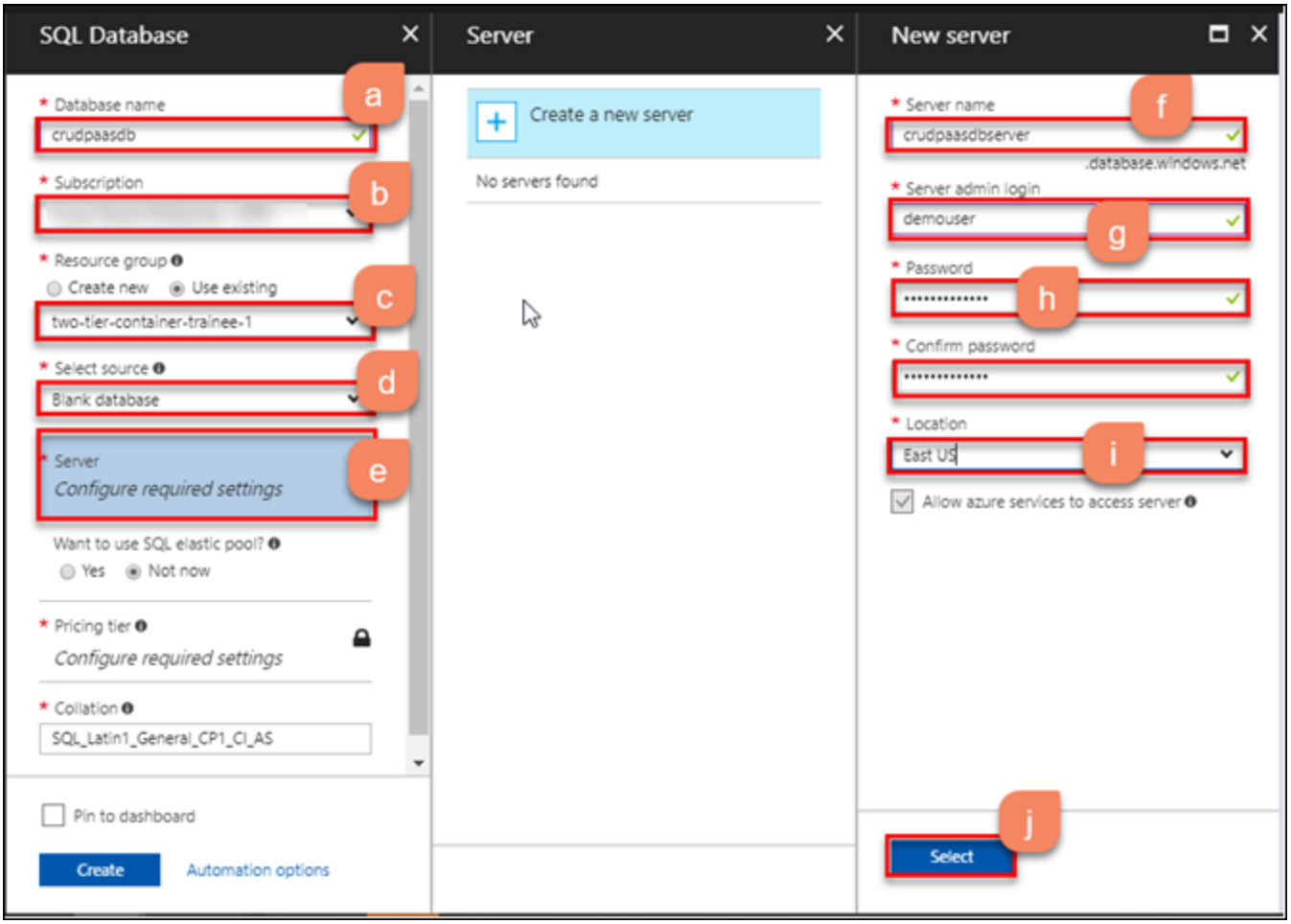
3. Search for *SQL Database* and select **SQL Database**



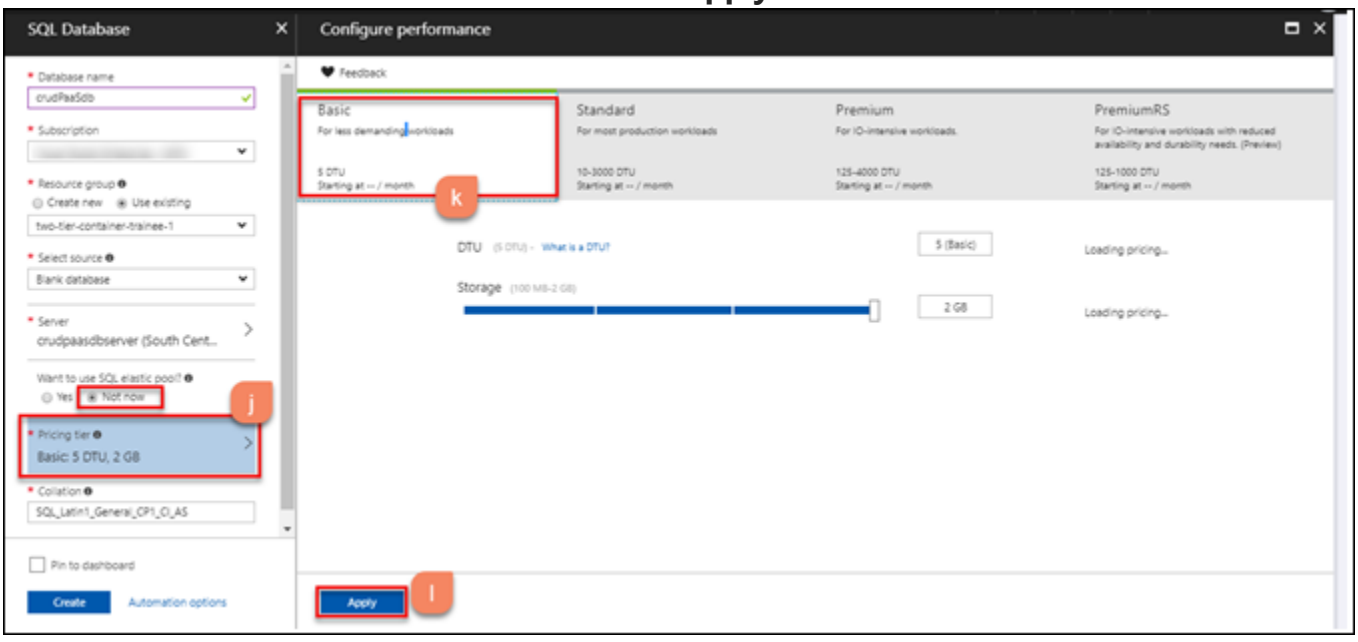
4. Click on **Create** button.



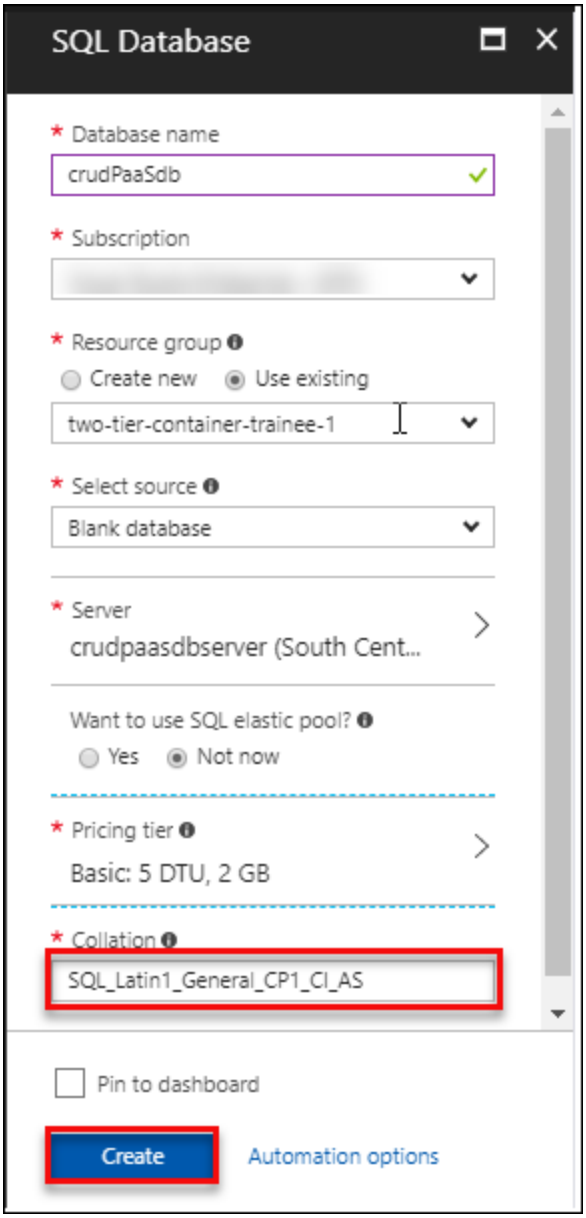
5. Populate the below parameters as shown below. Check for the unique value for server name (f) in Server creation wizard.



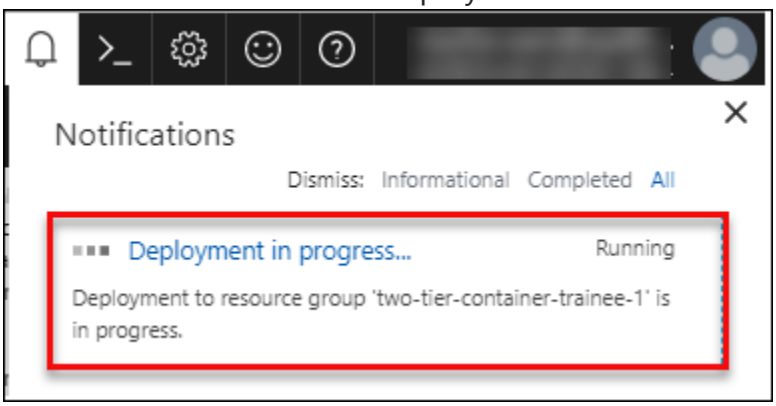
6. Select the size of database to be **Basic.** Click on **Apply.**



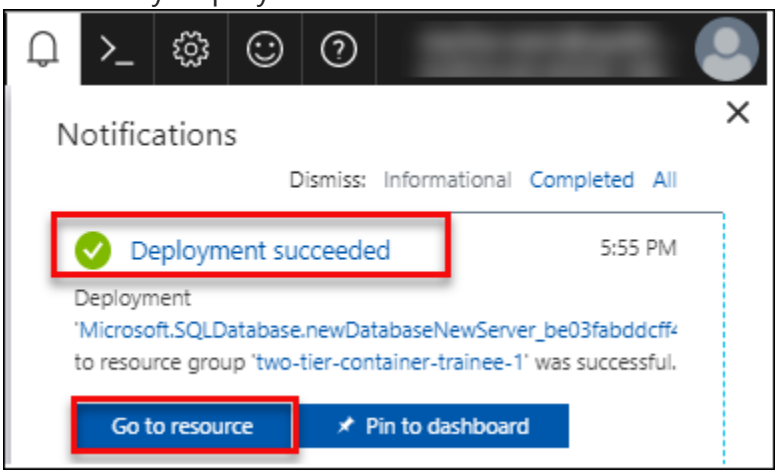
7. Leave the collation as default and click on **Create** button.



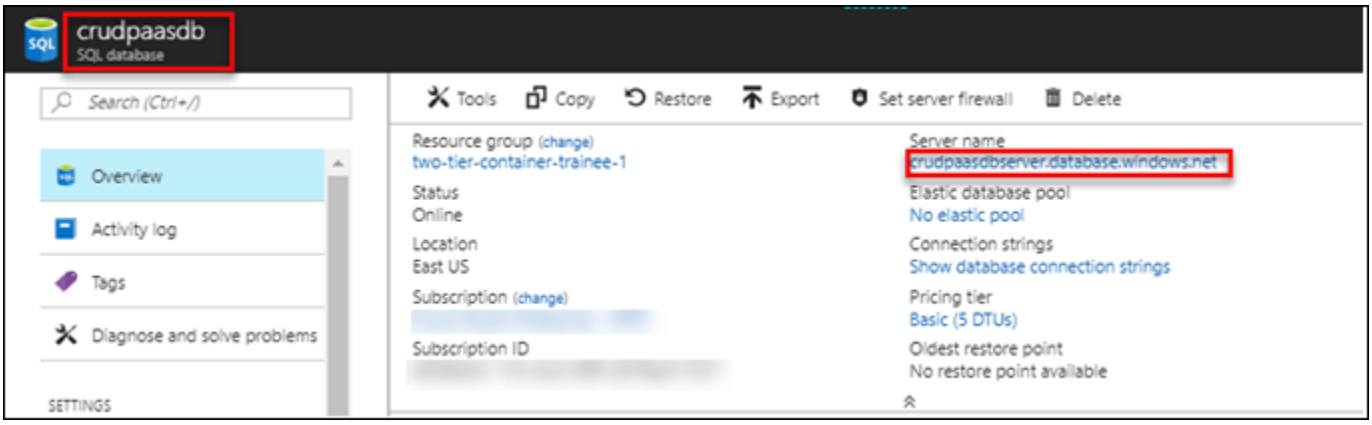
8. It will be validate and then deployment will start.



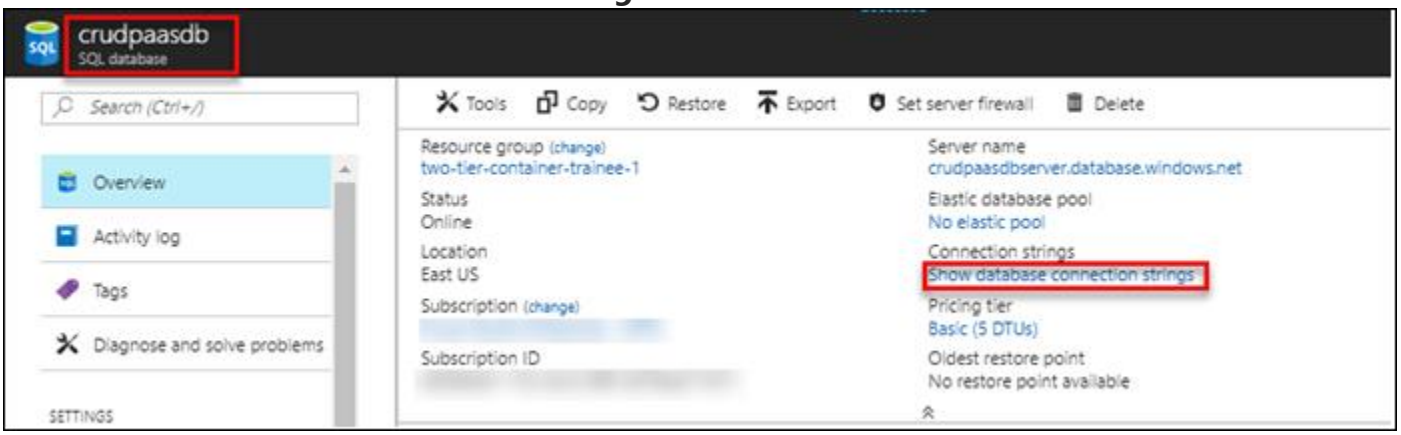
9. After deployment gets completed, click on **Go to resource** to verify that resource is successfully deployed.



10. You will see that database is created. Note down the server name to be used later.



11. Click on **Show Database Connection Strings** as shown below.



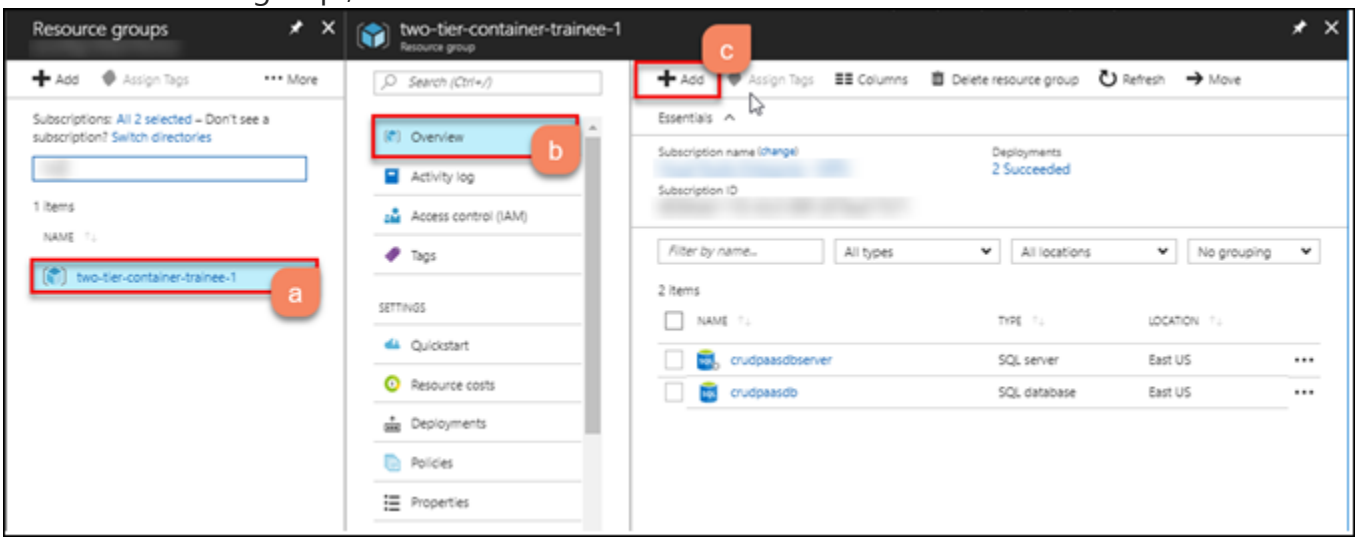
12. Copy the connection string in notepad to be used in database migration and modify the you're {your_username} and {your_password} with actual values you used while creating database in Lab2→ Exercise 1 → Step 4.

```
Server=tcp:cs2df6508fc98d4362.database.windows.net,1433;Initial
Catalog=cruddb;Persist Security Info=False;User
ID={your_username};Password={your_password};MultipleActiveResultSets=False;E
ncrypt=True;TrustServerCertificate=False;Connection Timeout=30;
```

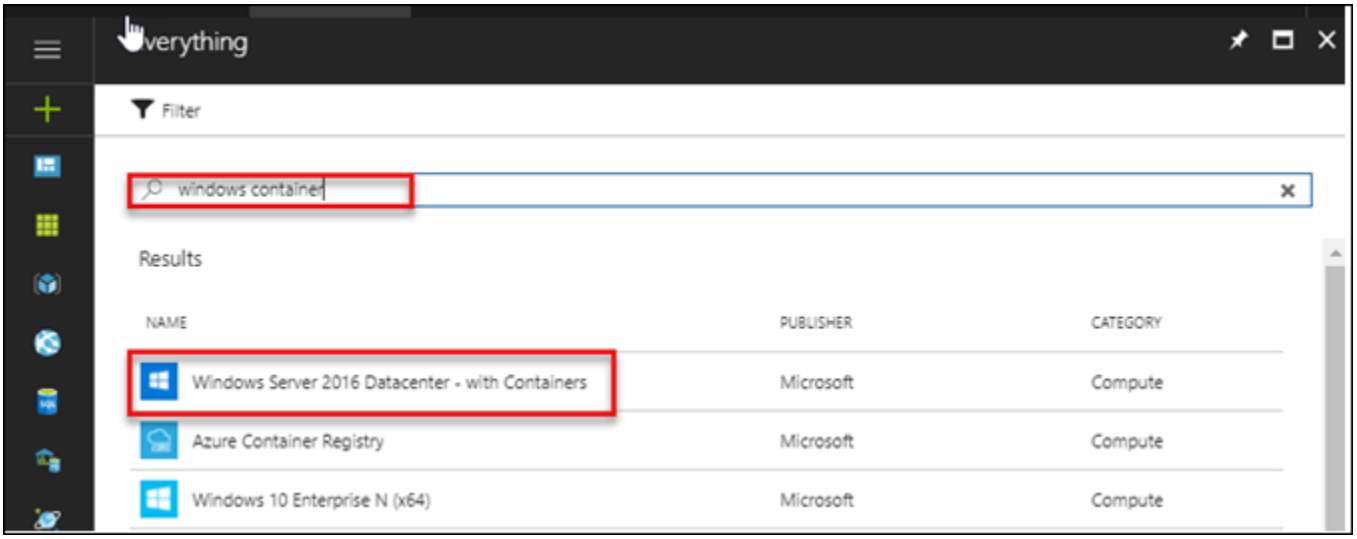
Exercise 2: Deploy Windows 2016 Server with Docker

In this exercise, we will create Windows server with Docker to migrate WebApp on container.

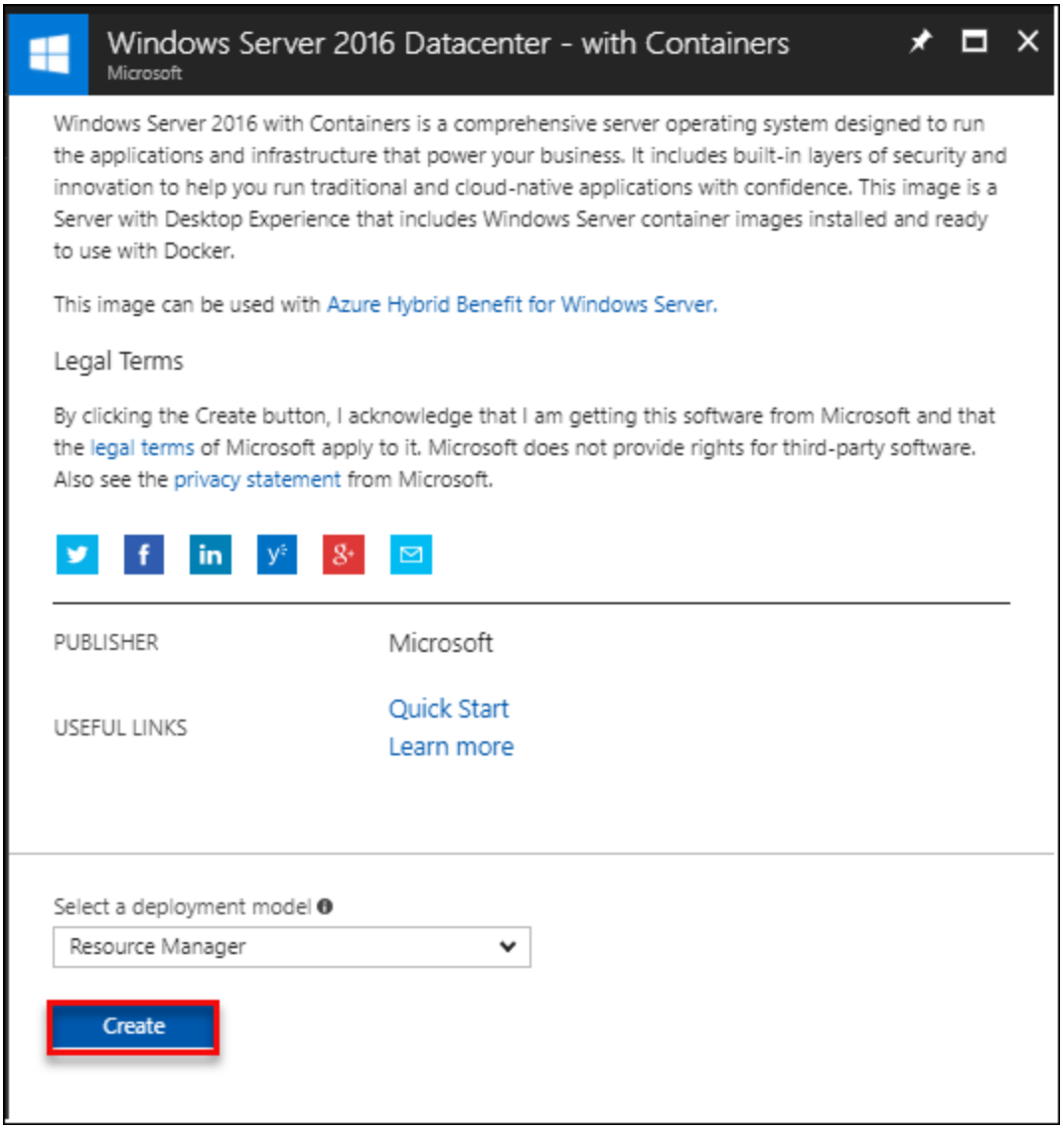
1. Click on resource group , click on **Add** button under **Overview** section.



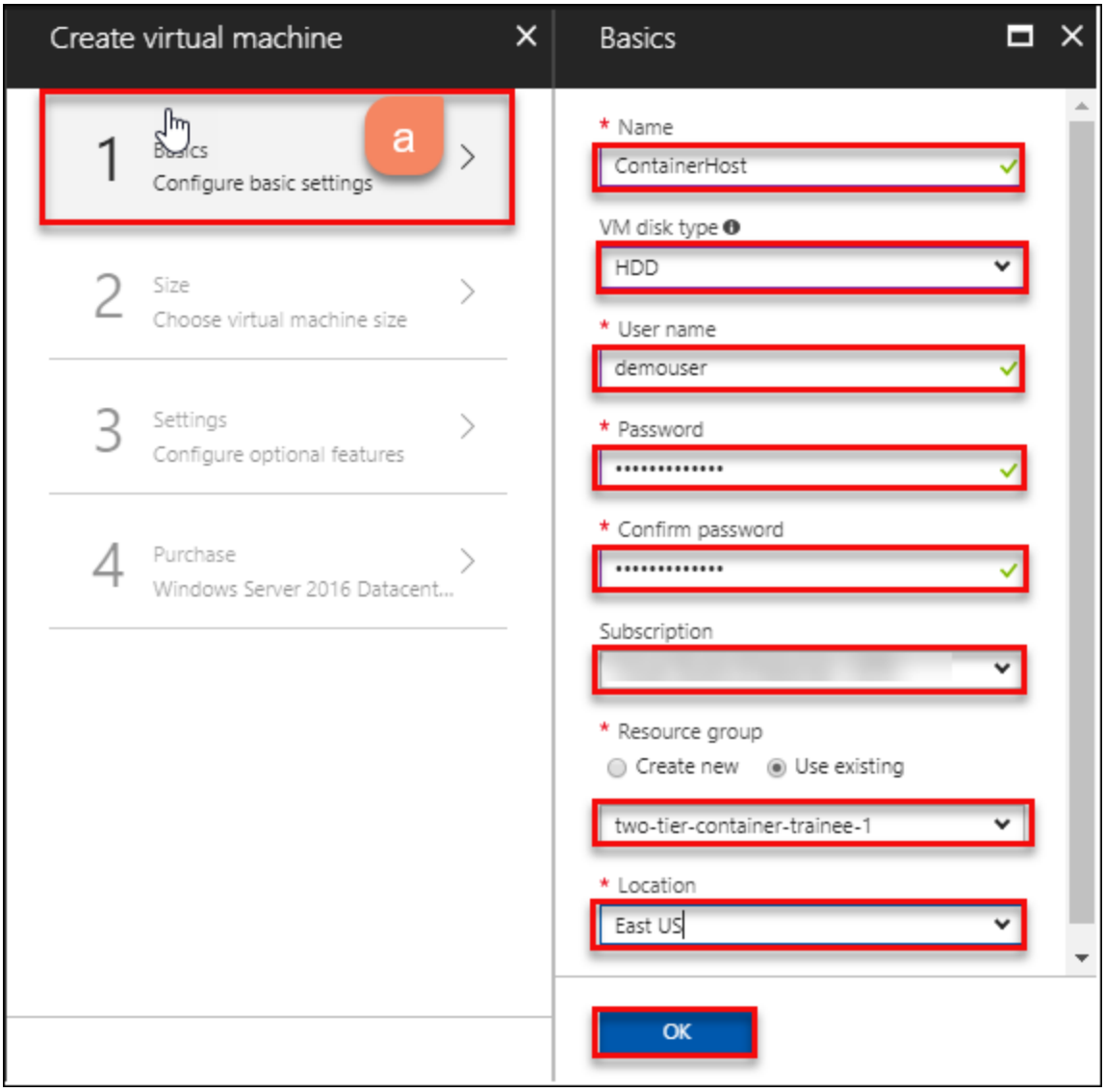
2. Search for *Windows Container* and then select **Windows Server 2016 Datacenter – with Container.**



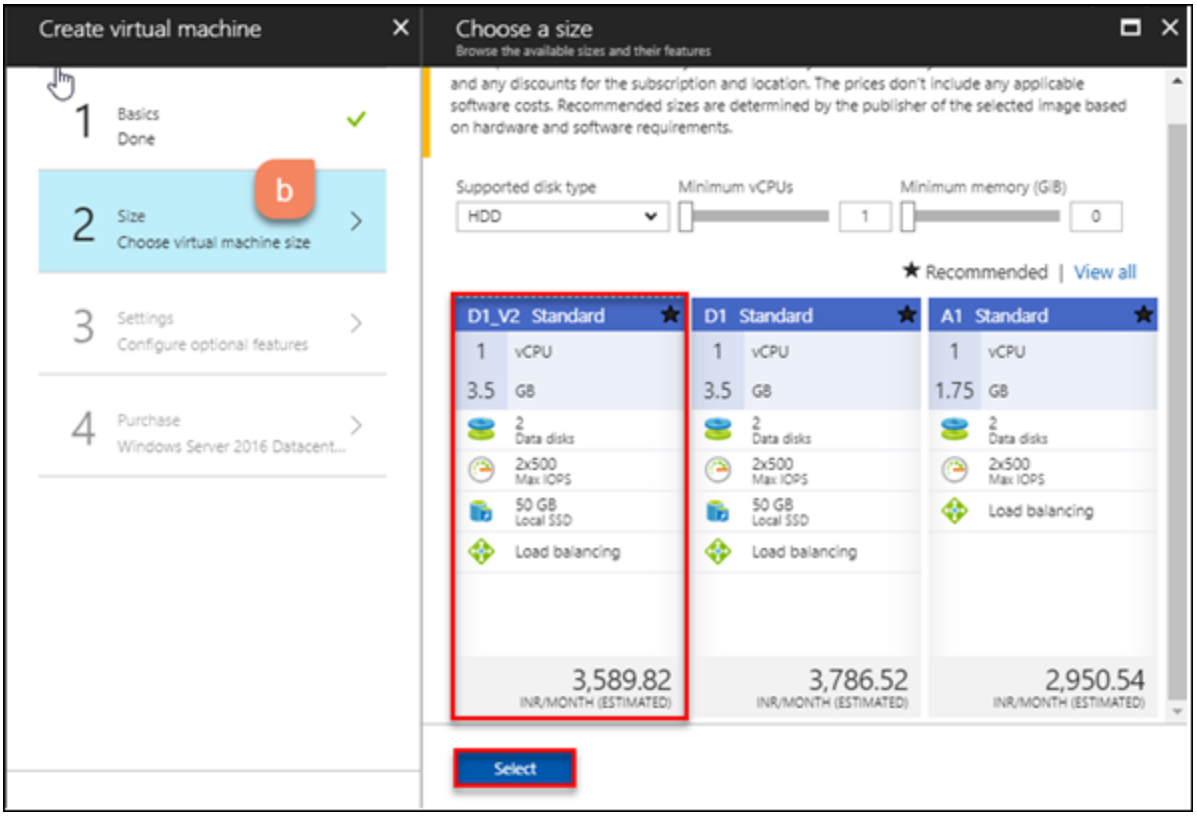
3. Click on **Create** button in the **Windows Server 2016 Datacenter – with Container** blade.



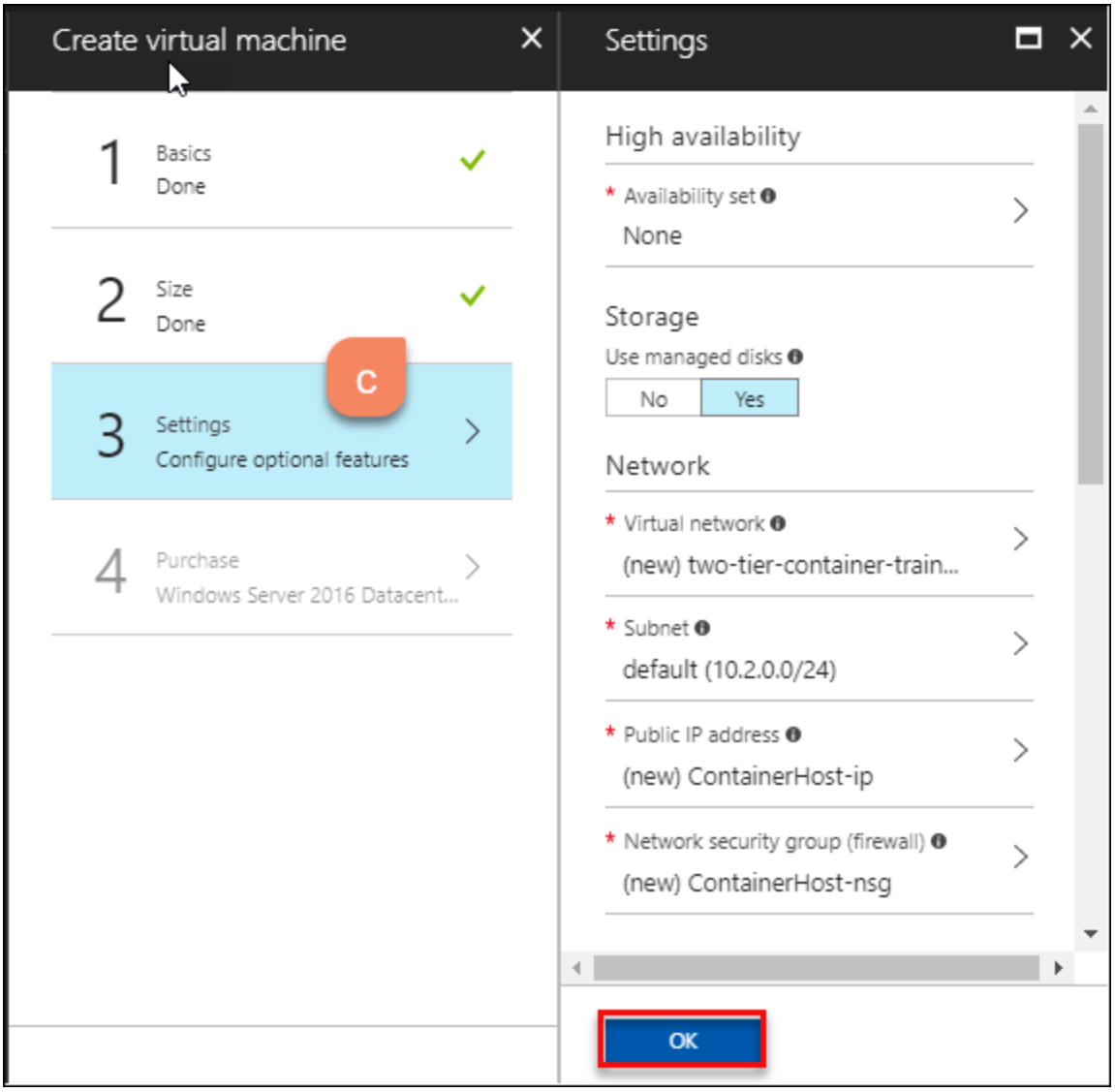
4. In **Basics** section, populate the parameters as shown below and clicn **OK**



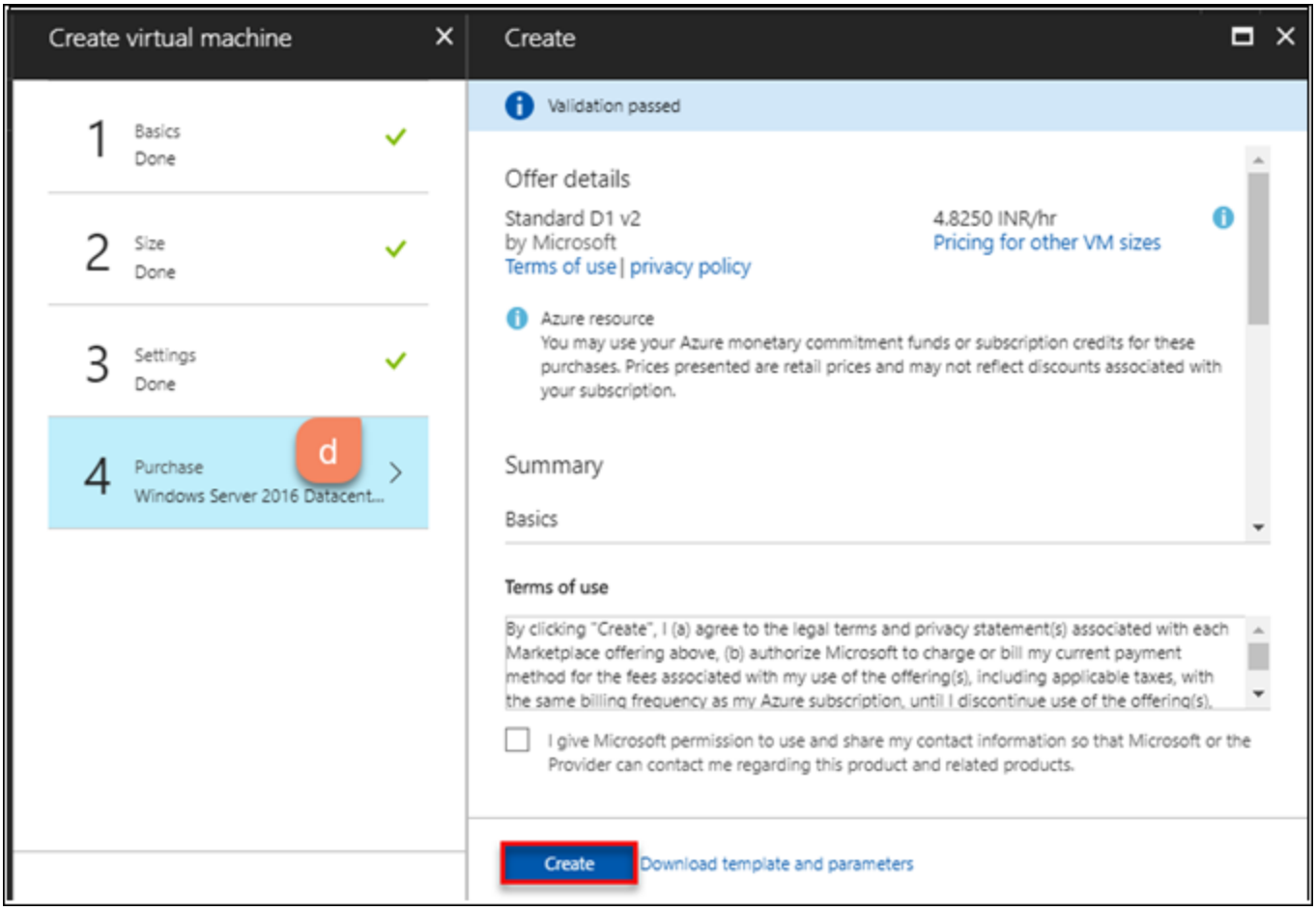
5. Select the size of the VM as **D1_V2 Standard** and click **Select.**



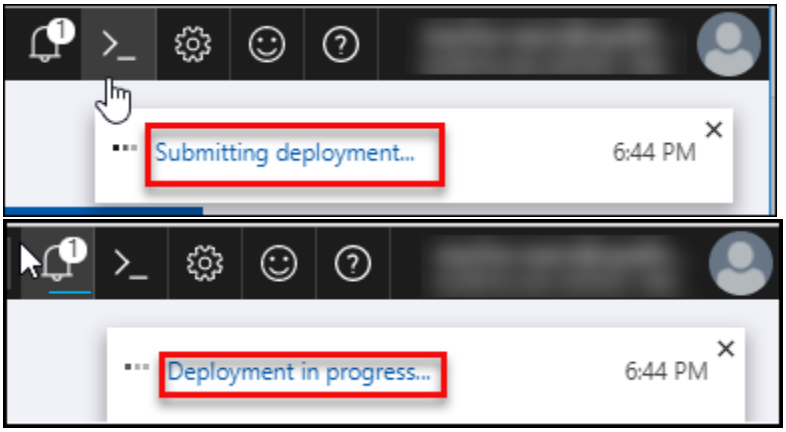
6. In **Settings** Blade, continue with defaults and click on **OK** button.



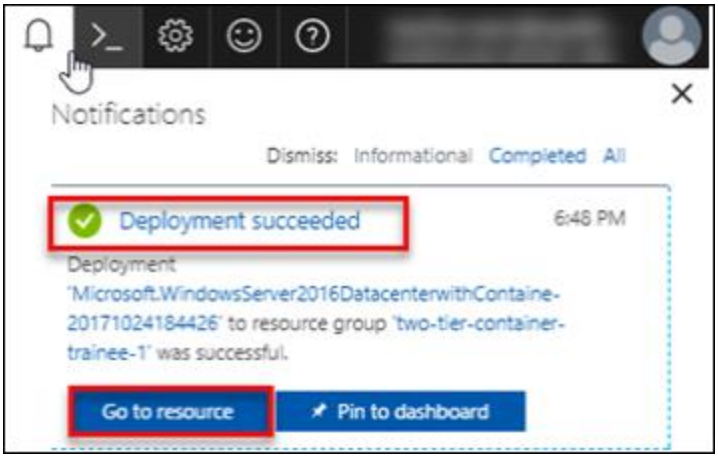
7. In Purchase Blade, click on **Create** button.



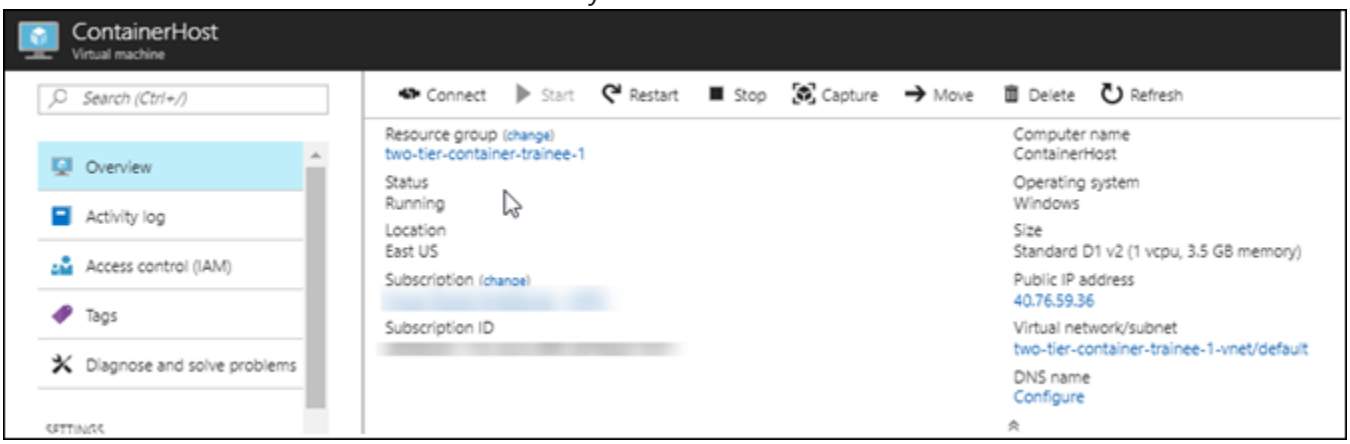
8. A notification will get raise for submitting the deployment and then deployment will get started.



9. Once deployment is successful, below notification will be raised. Click on **Go to resource** to validate the resource.



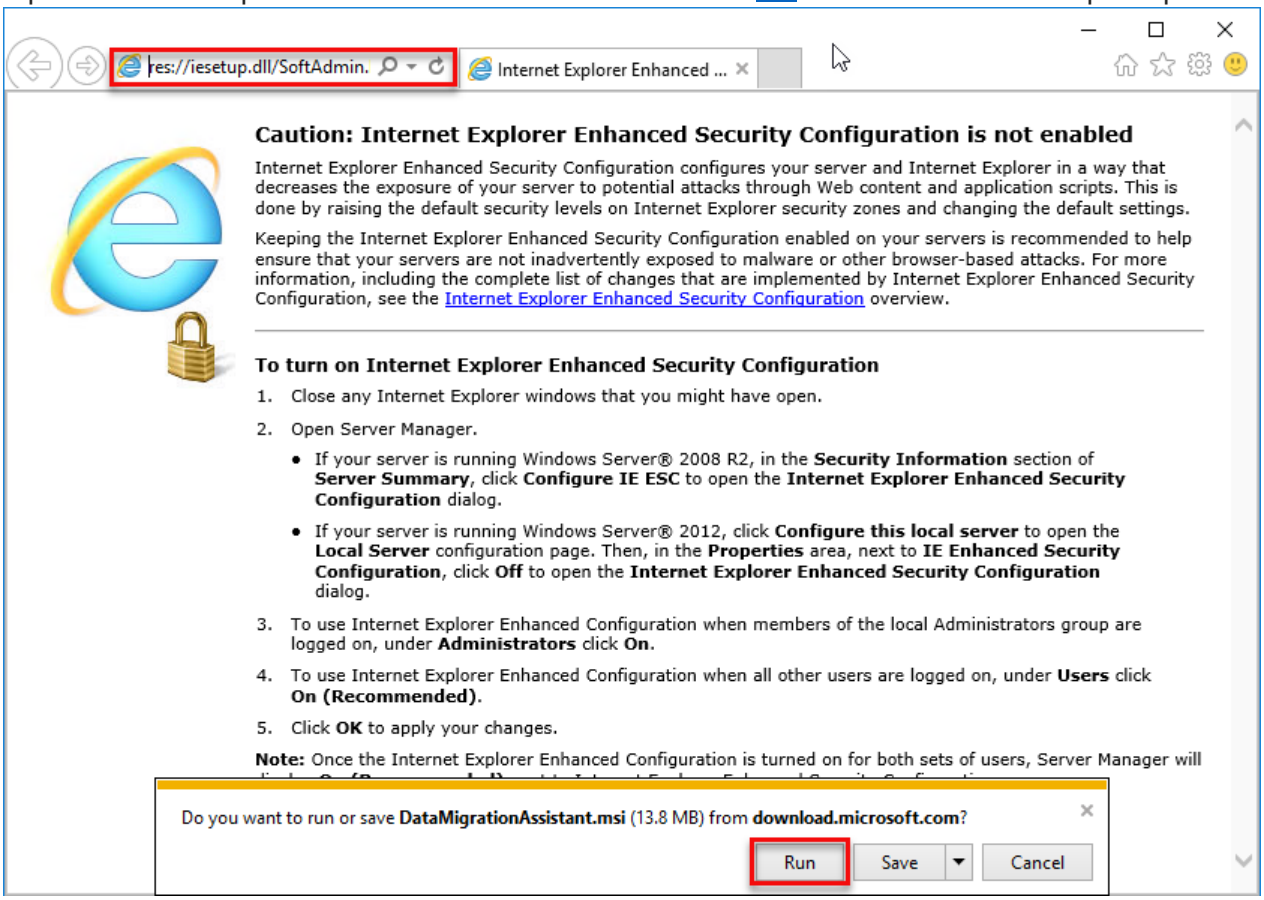
10. Windows VM with container is successfully installed.



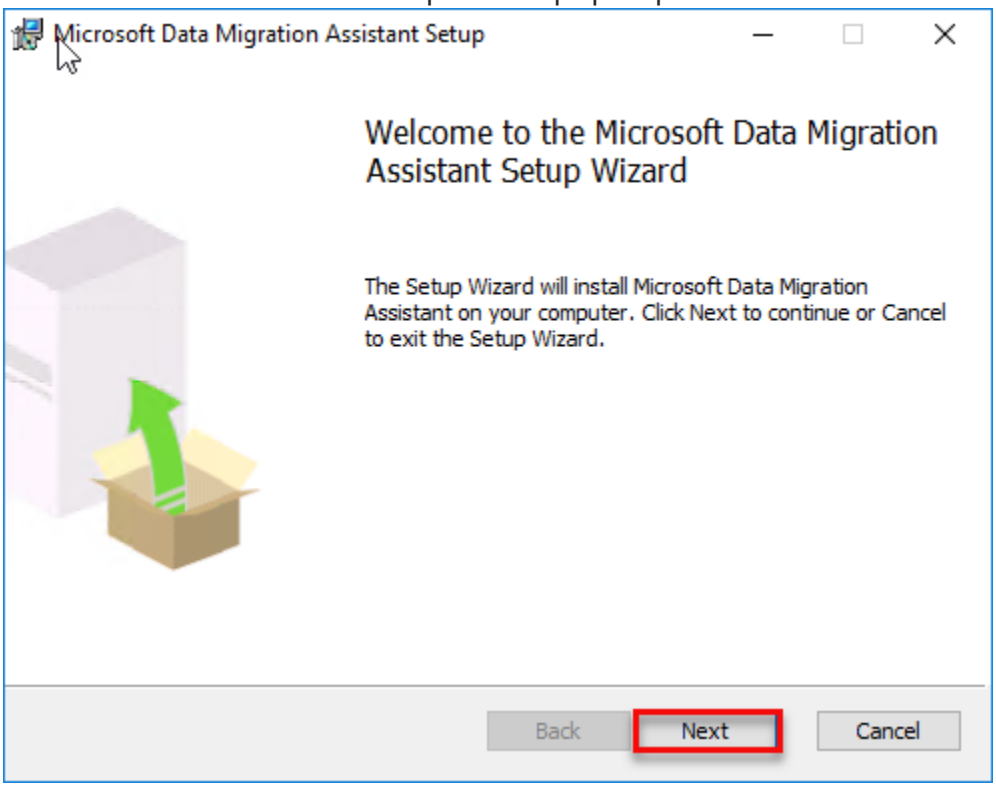
Exercise 3: Install DMA in On-premises VM and migrate database

In this exercise, we will login to on-premises VM and install database migration assistant.

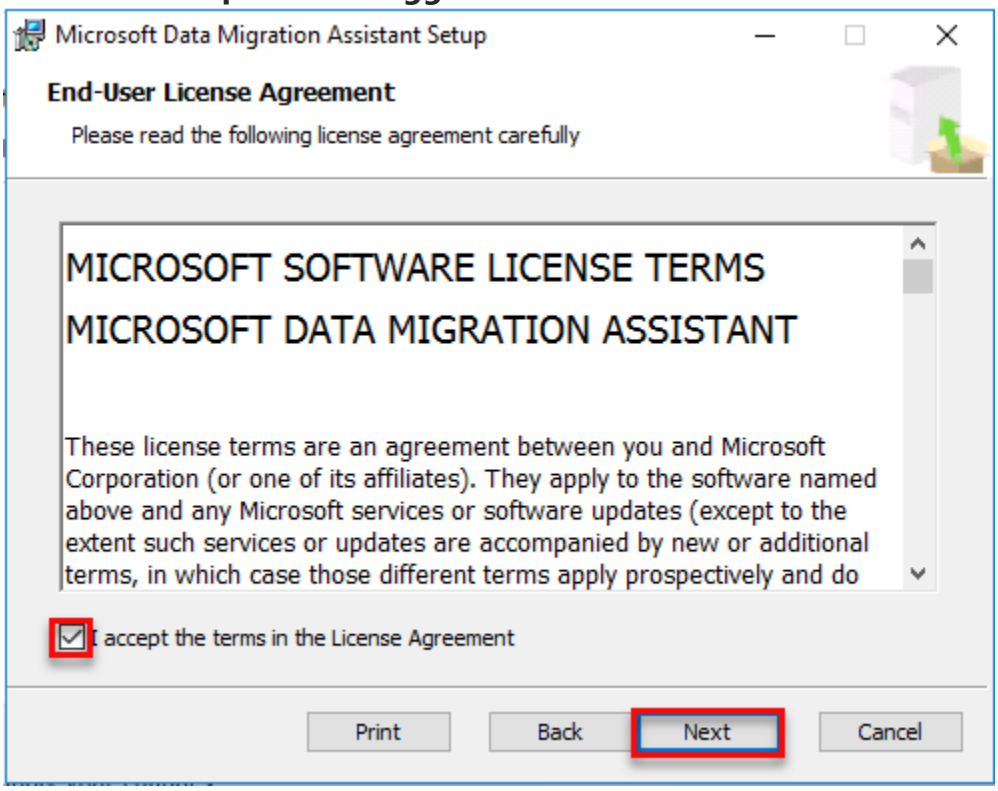
1. Login to On-premises VM if not already logged-in, follow Lab1 → Exercise 2 → step-1 to step-5.
2. Open Internet Explorer and download the DMA from Url. Click on **Run** when prompted.



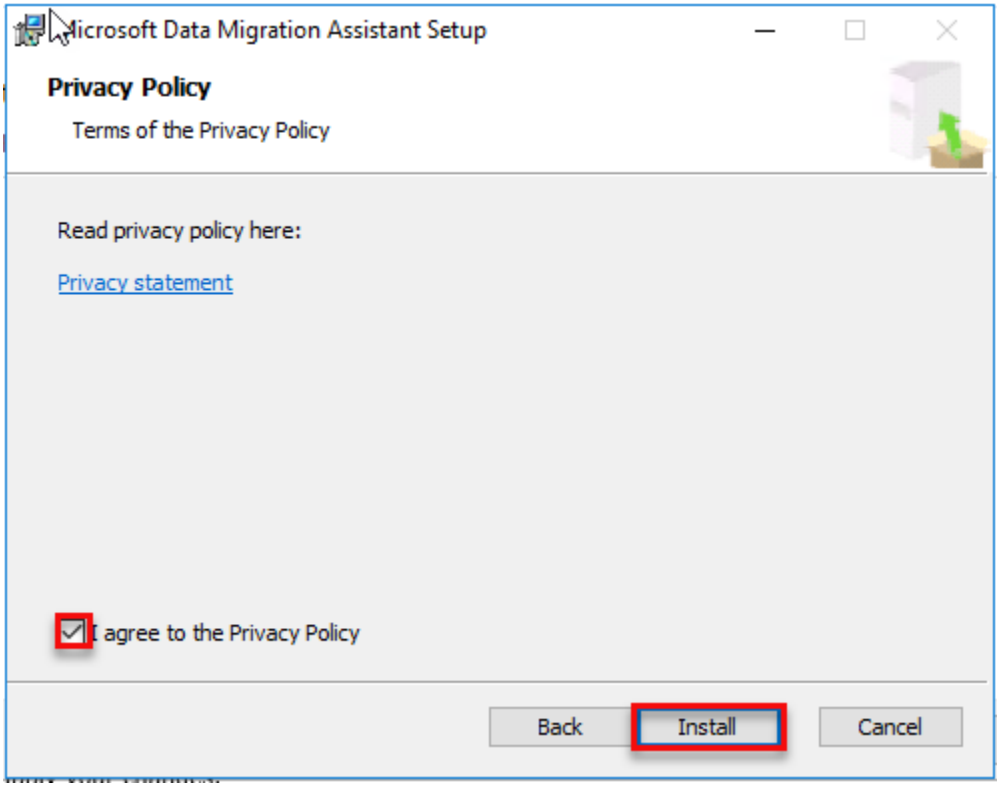
3. Click on **Next** when DMS setup wizard pops-up.



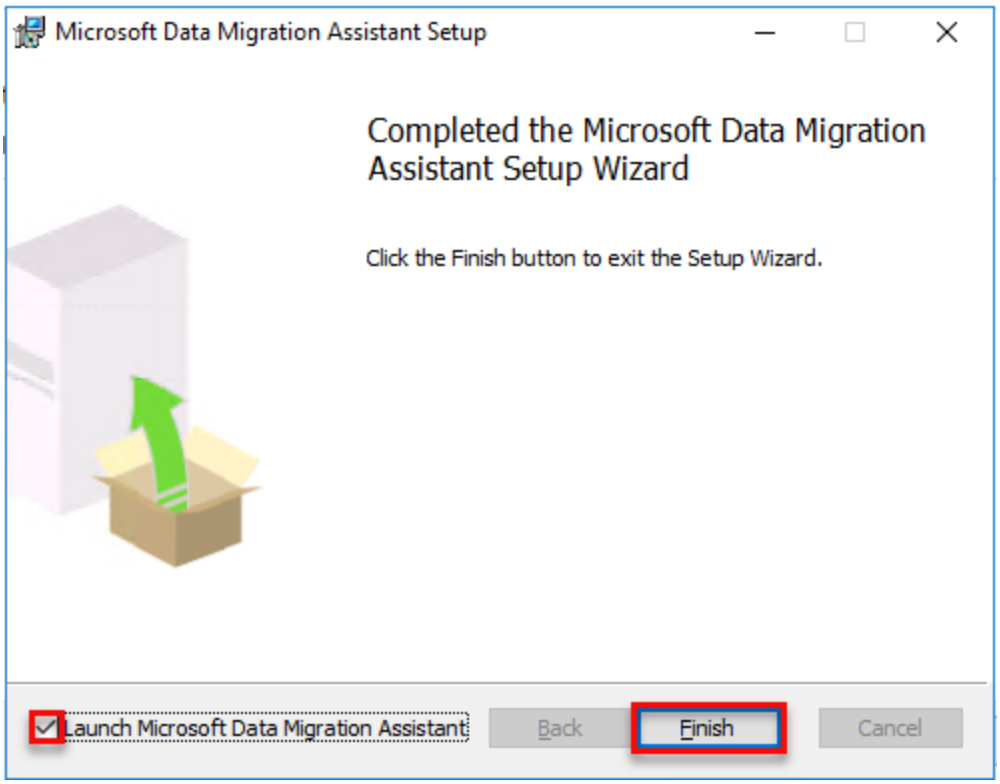
4. Check the **Accept License Aggreement** and click **Next.**



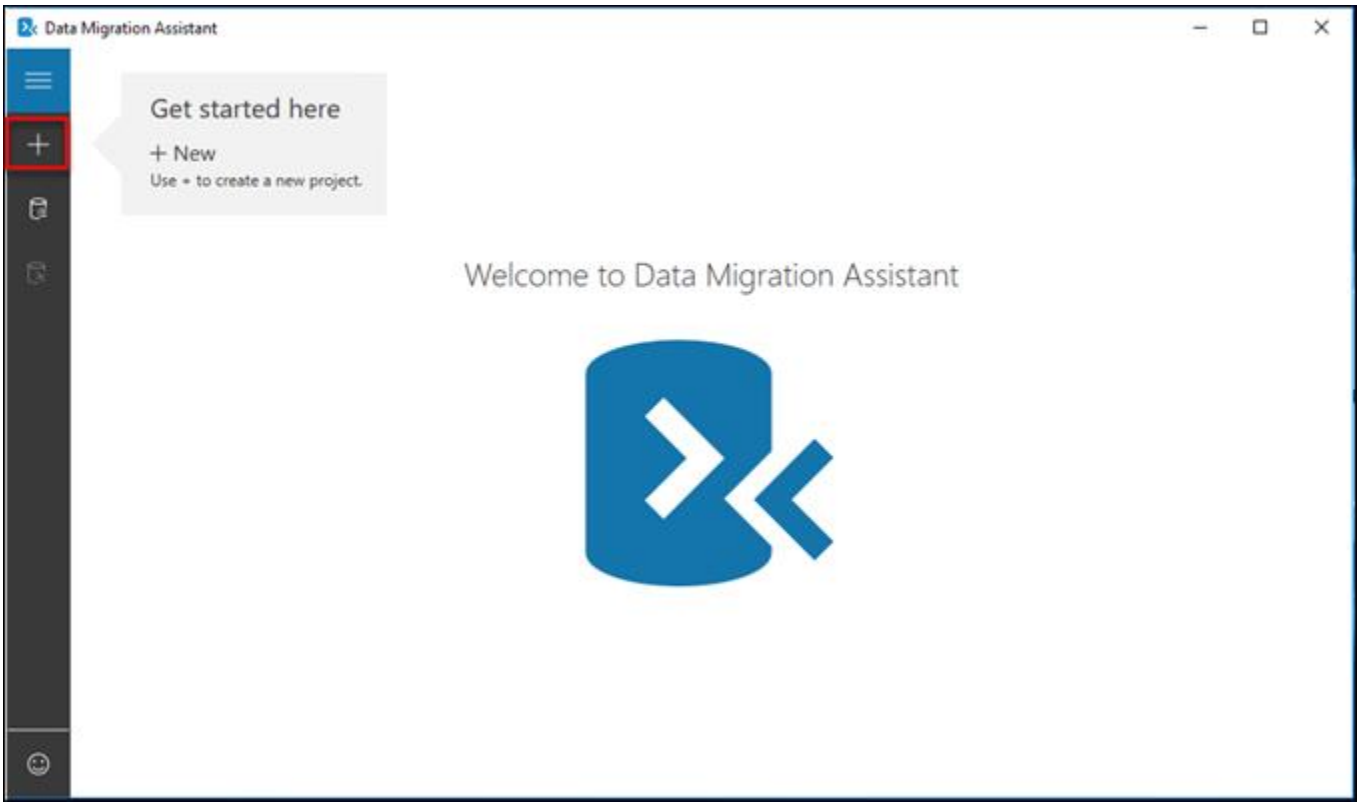
5. Check the **I agree to the privacy policy**, and click on **Install.**



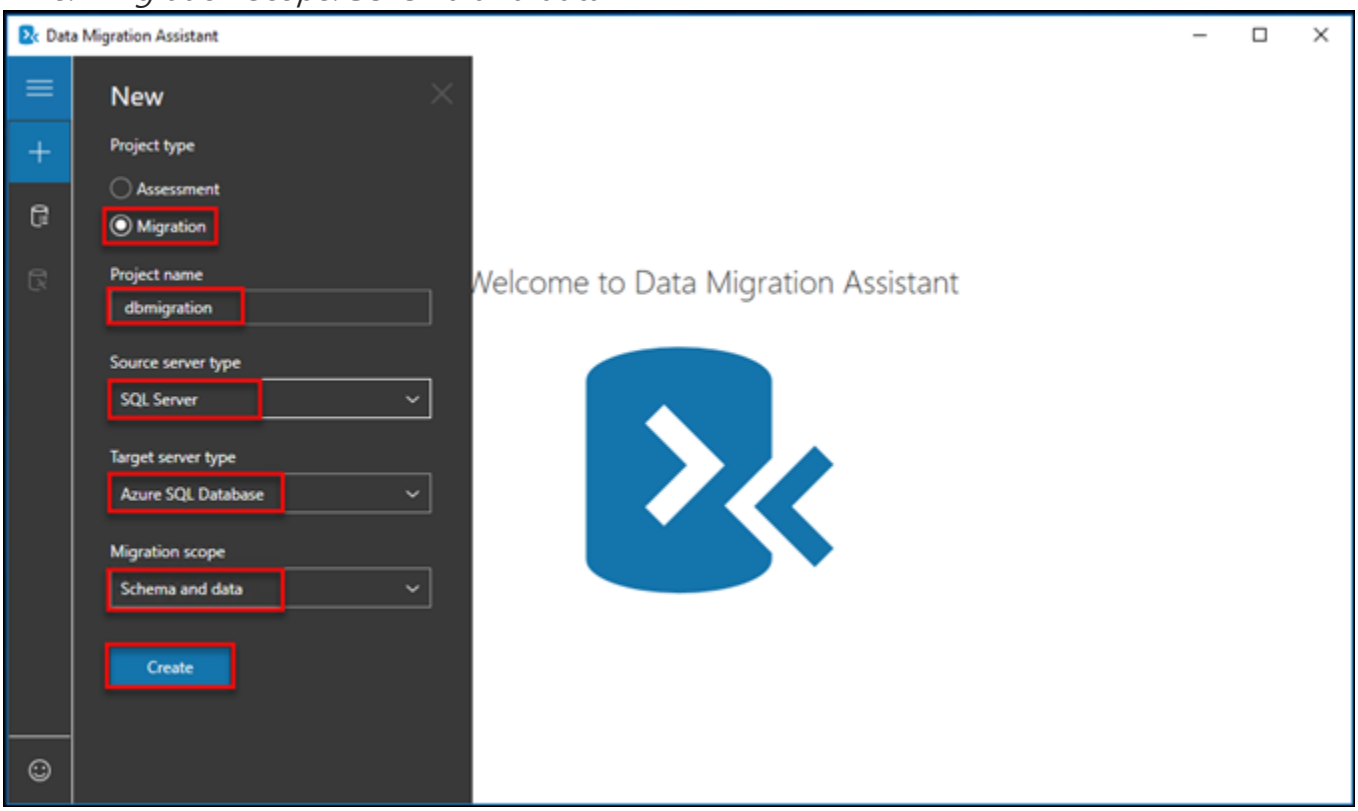
6. Once installation completes, check the **Launch Microsoft Data Migration Assistant** and click on **Finish.**



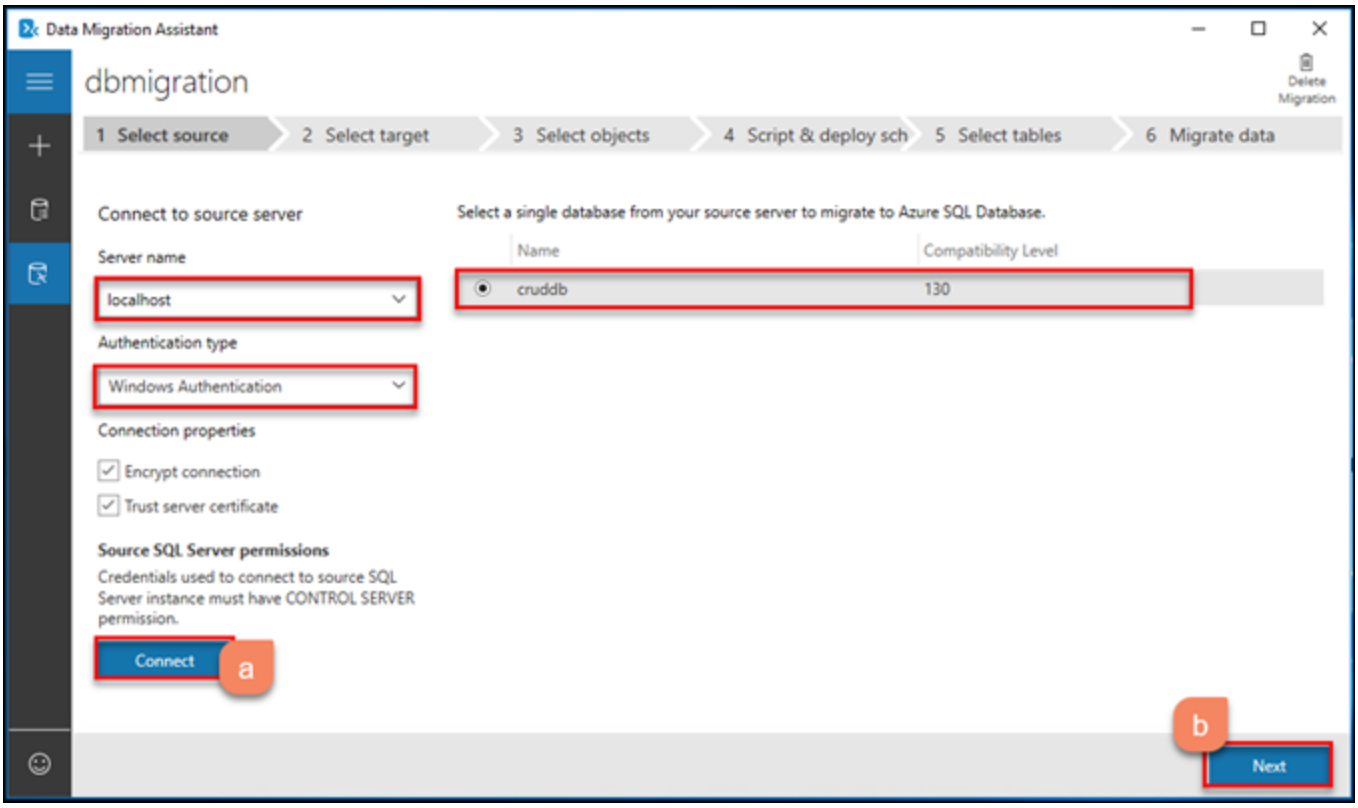
7. When Data Migration Assitant application launches, click on **+** button on the left to create a new project.



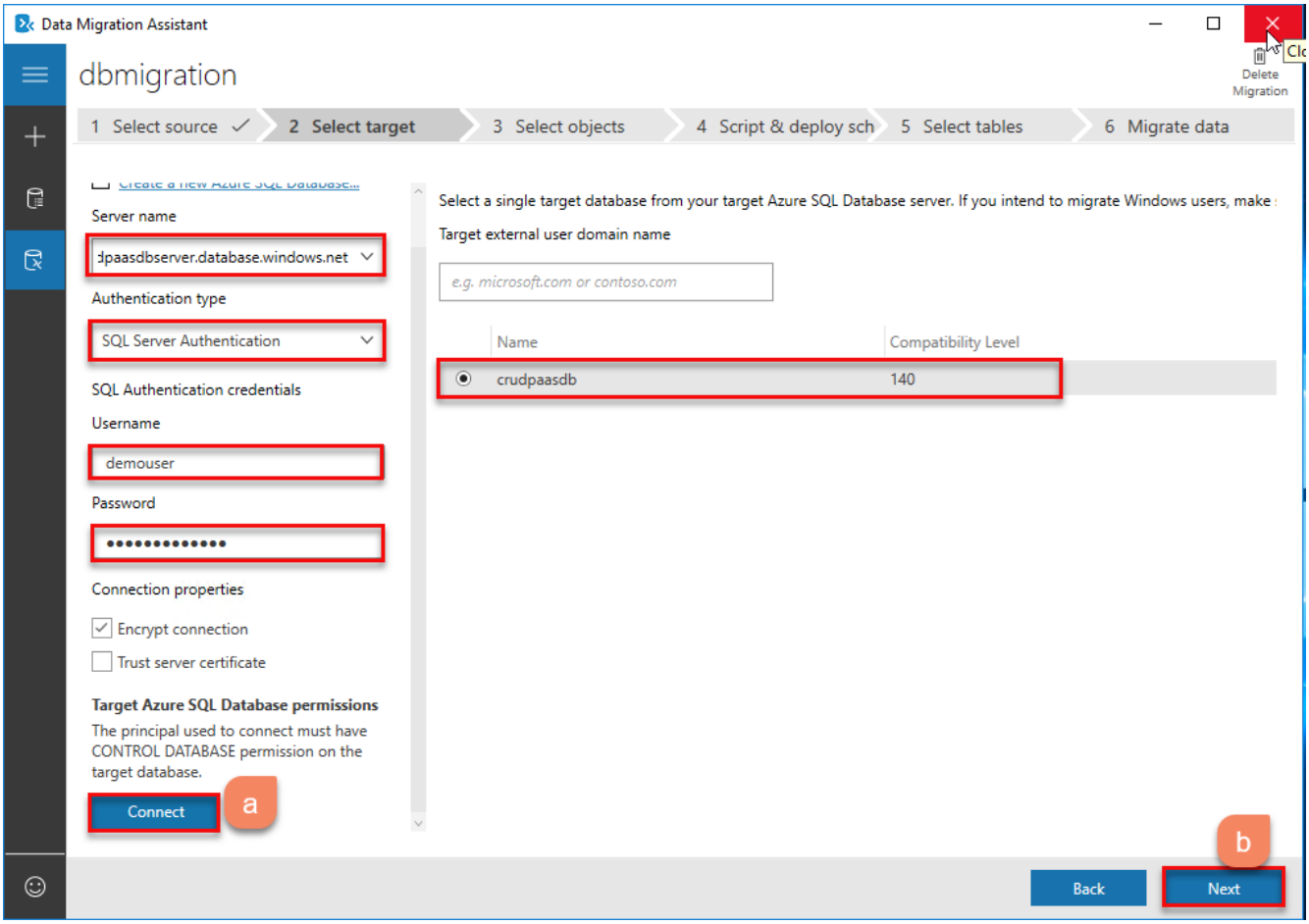
8. Populate the New Project with values provided below and click on **Create** button
    a. *Project Type:* **Migration**
    b. *Project Name: <any name>*
    c. *Source server type:* **SQL Server**
    d. *Target Server Type:* **Azure SQL Database**
    e. *Migration Scope:* **Schema and data**



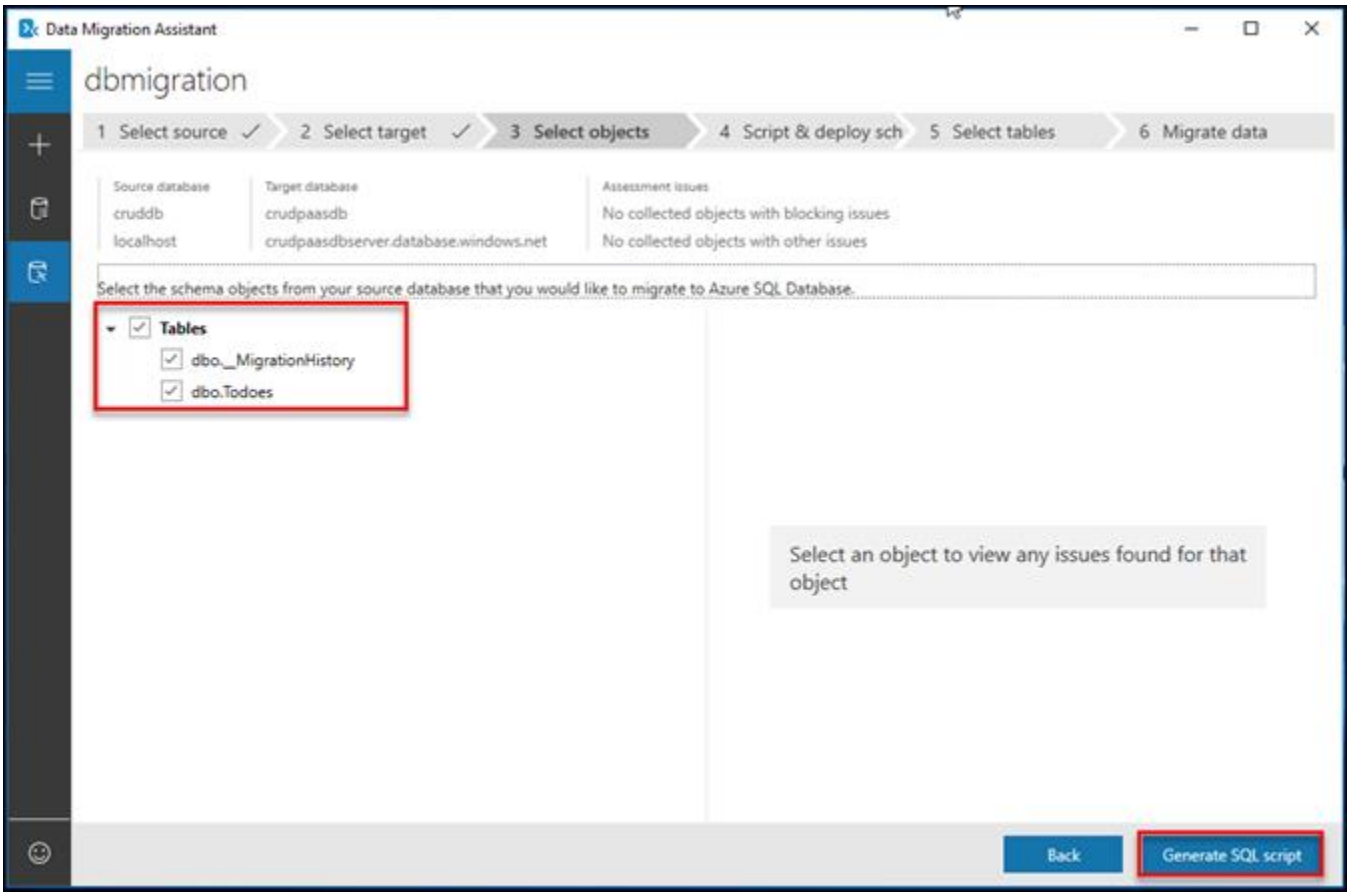
9. In the **Select Source**, populate below parameter and click **Connect** which will list down the database to be migrated, click **Next**
    a. *Server name:* **localhost** (as tool is installed on same server where database in installed)
    b. *Authentication type:* **Windows Authentication** (logged in user do have rights on database)



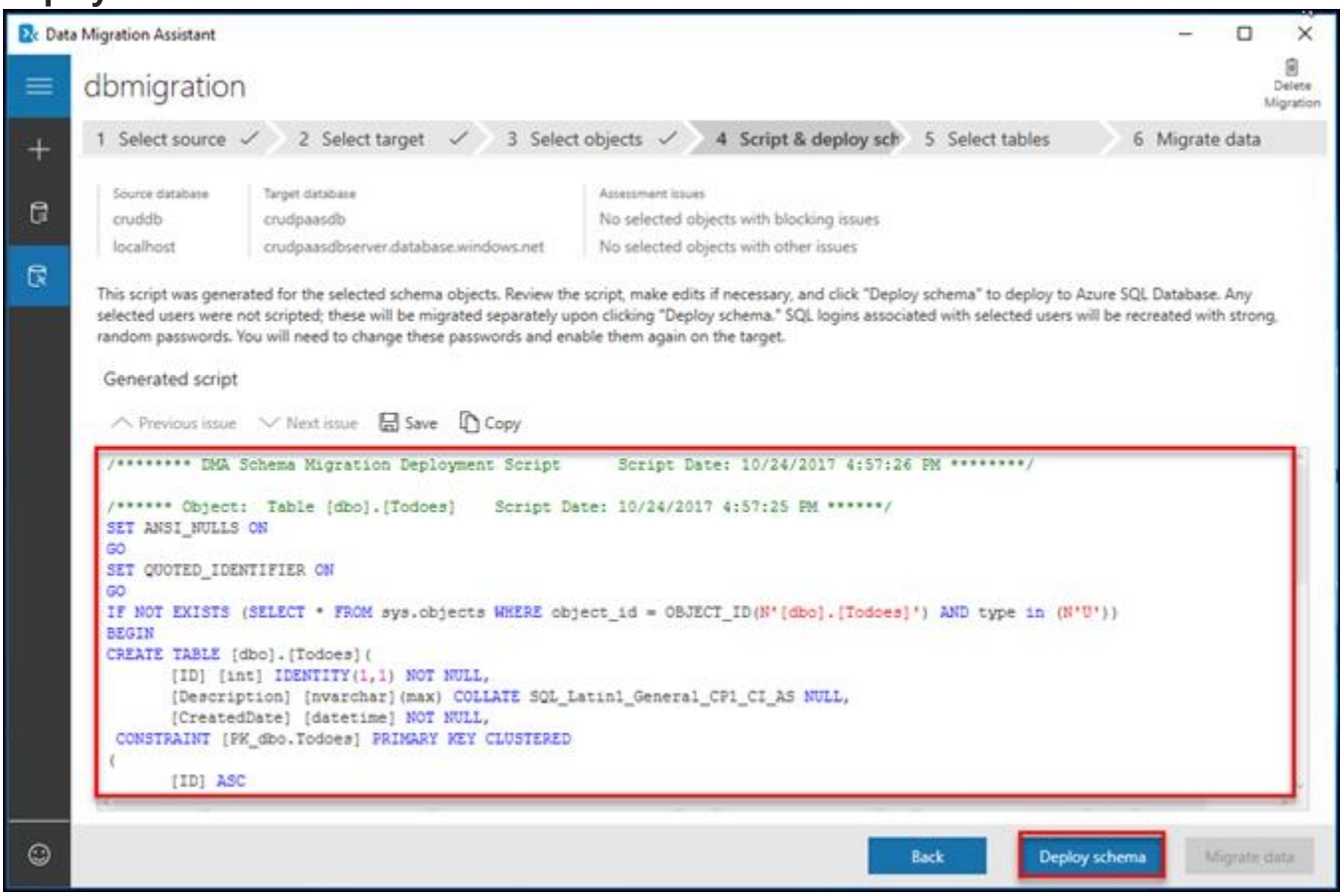
10. In the **Select target**, populate the below parameter and click **Connect** which will list down the database at destination. Select the databse and click **Next**
    a. Server Name: <<Exercise 2→ Step 9>
    b. Authentication type: **SQL Server Authentication**
    c. Username: <Lab2→ Exercise 1→ Step 4 → g>
    d. Password: < Lab2→ Exercise 1→ Step 4 → h >



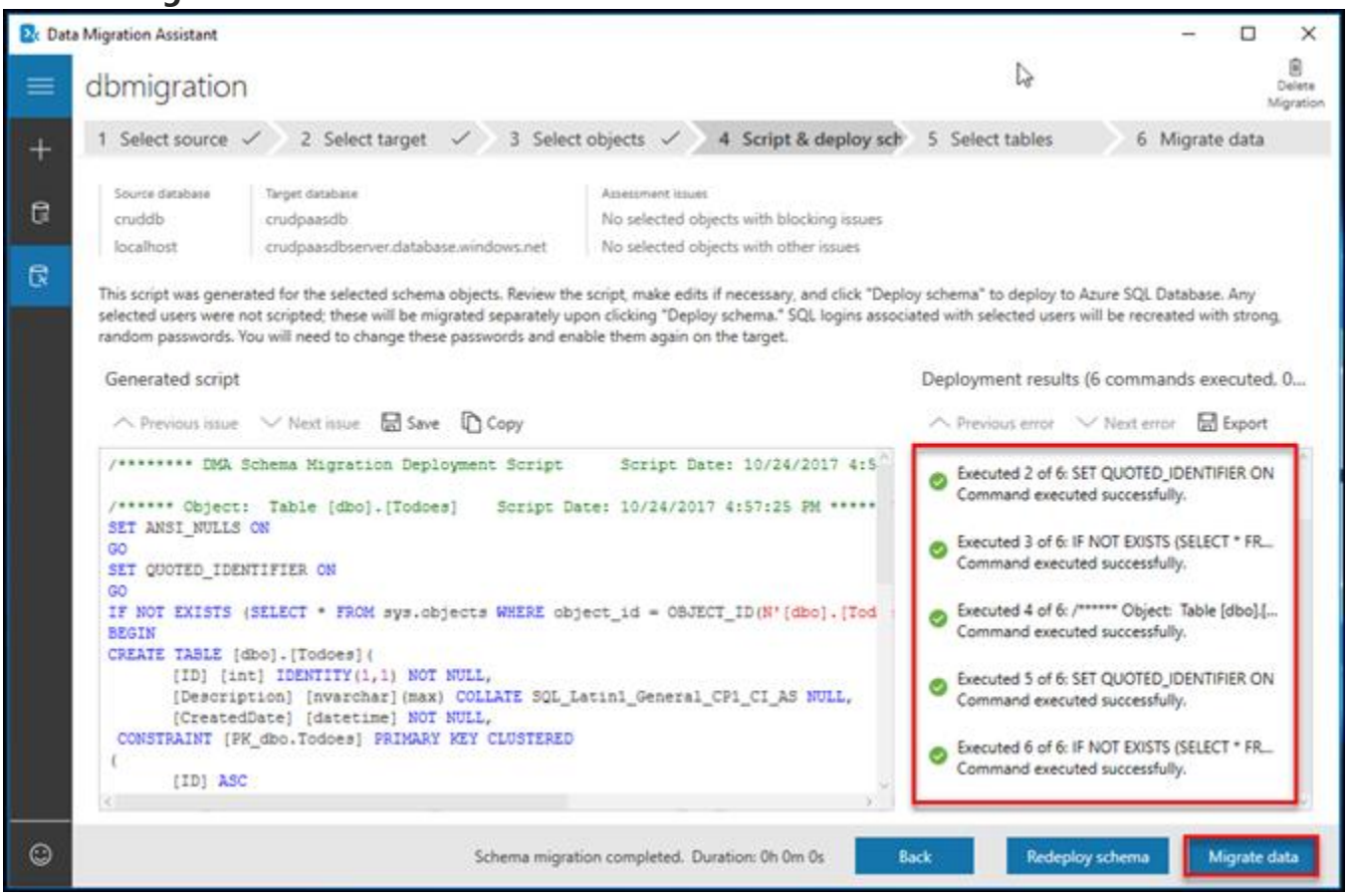
11. In **Select objects**, you will be provided with the tables which are in source database, here you can select/deselect table which you want to migrate to Azure PaaS database. Keep the default and click on **Generate SQL Script.**



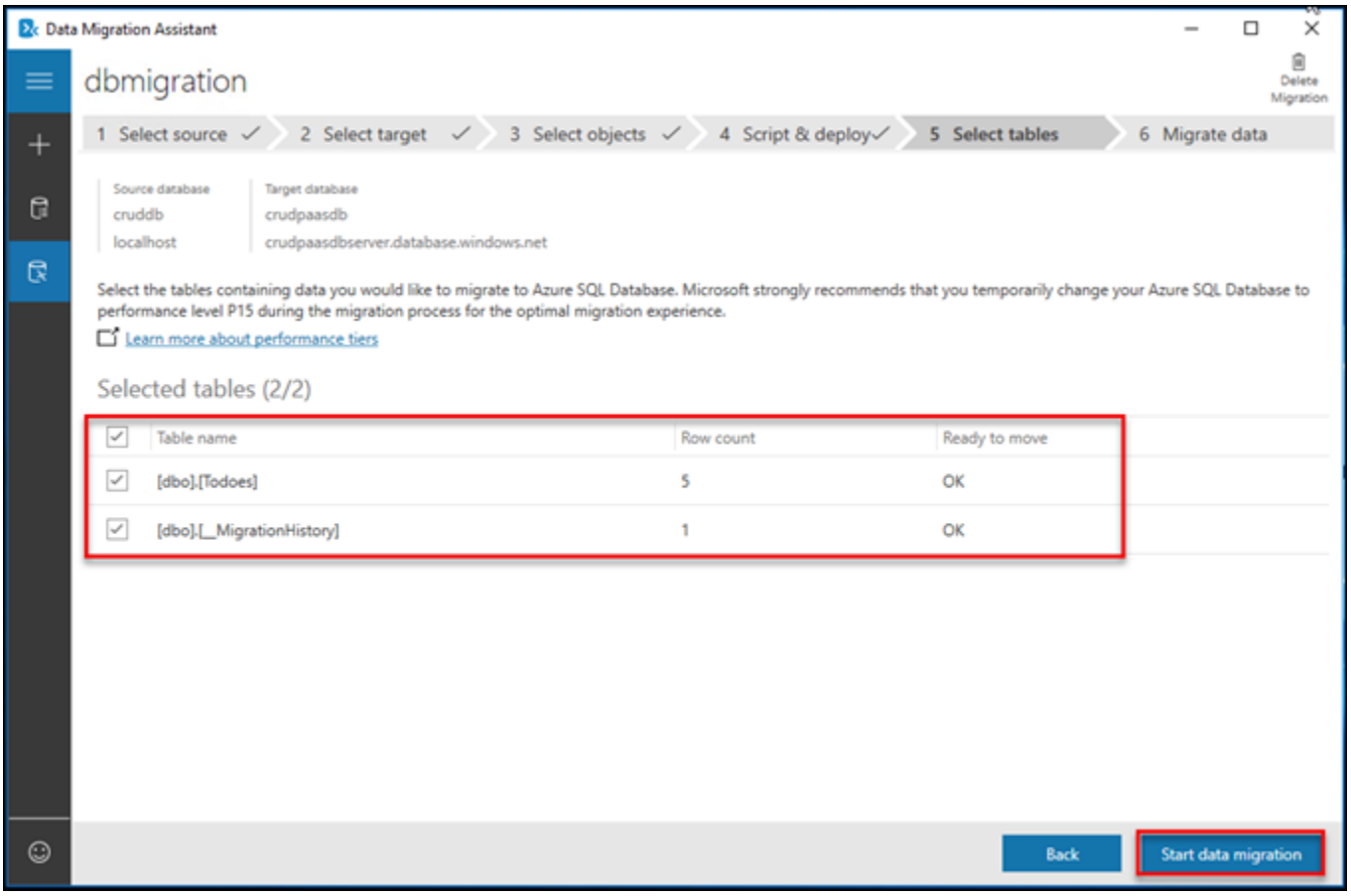
12. You will see an SQL script which will be used to create schema at destination database. Click on **Deploy schema** button.



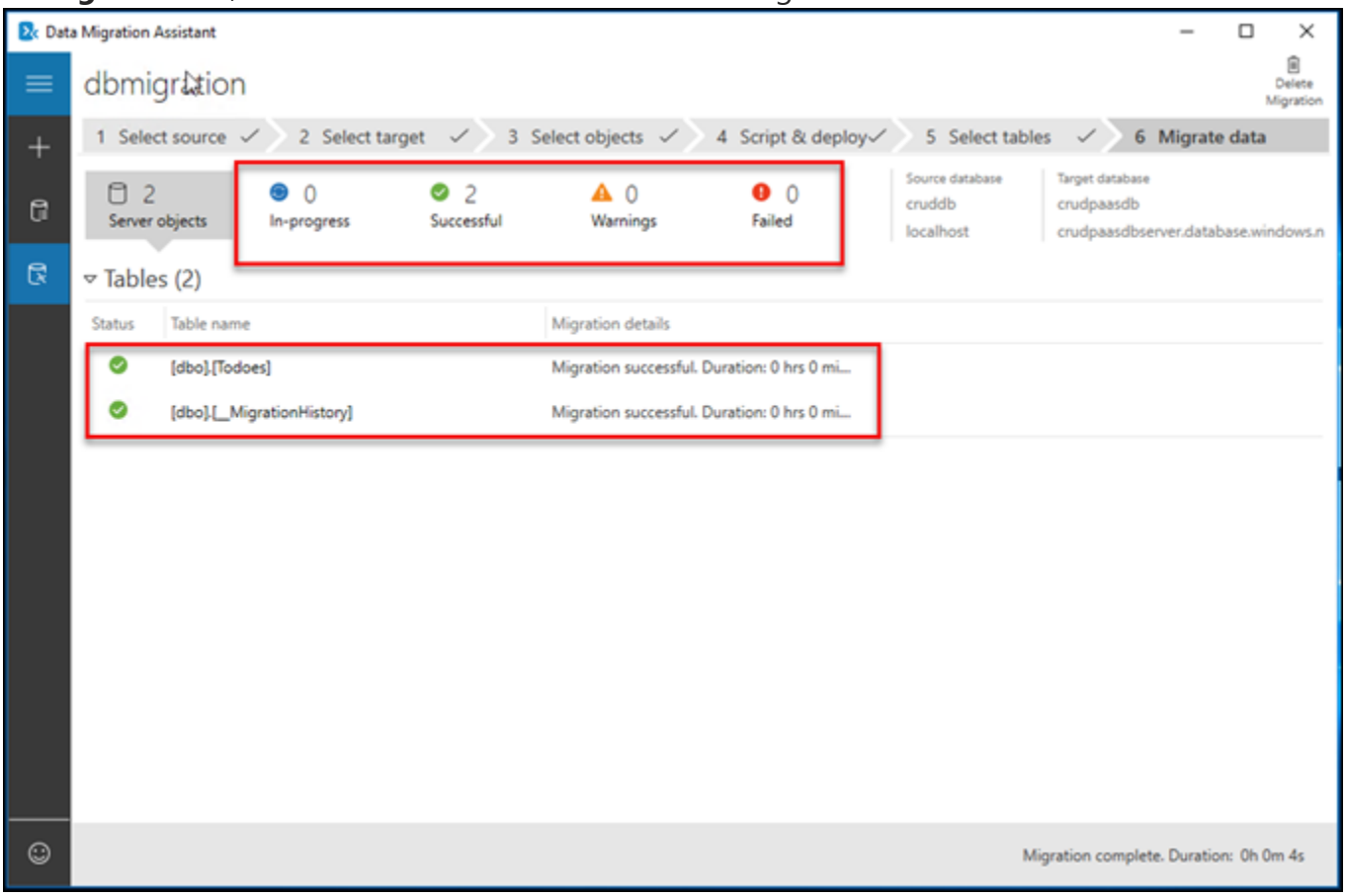
13. You will see the execution of script is done successfully completed in **Deployment results.** Click on **Migrate Data**.



14. In Select tables, you can choose which all table data to be migrated. Leave it default and click on **Start data migration.**



15. In **Migrate data**, You can see the final status of data migration.

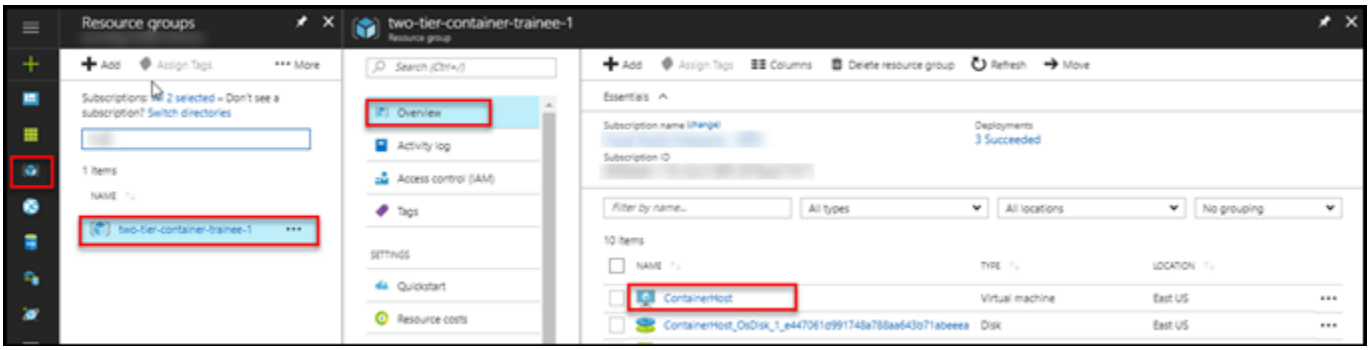


Now we have successfully migrated SQL database to Azure SQL database. Next, we we will move on to migrate webapp to containers.

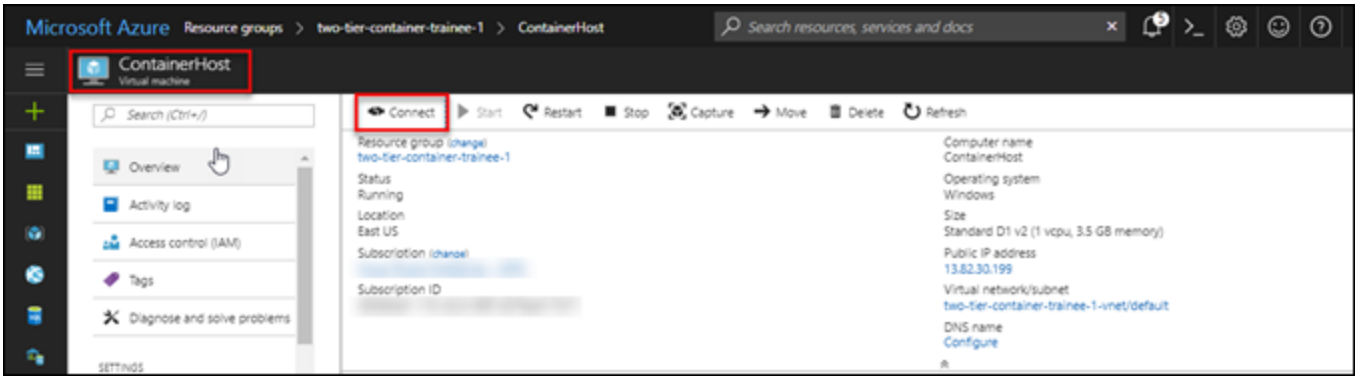
Exercise 4: Migrate WebApp to Container

In this exercise, we will login to VM which we created in Exercise 2 and migrate WebApp to container.

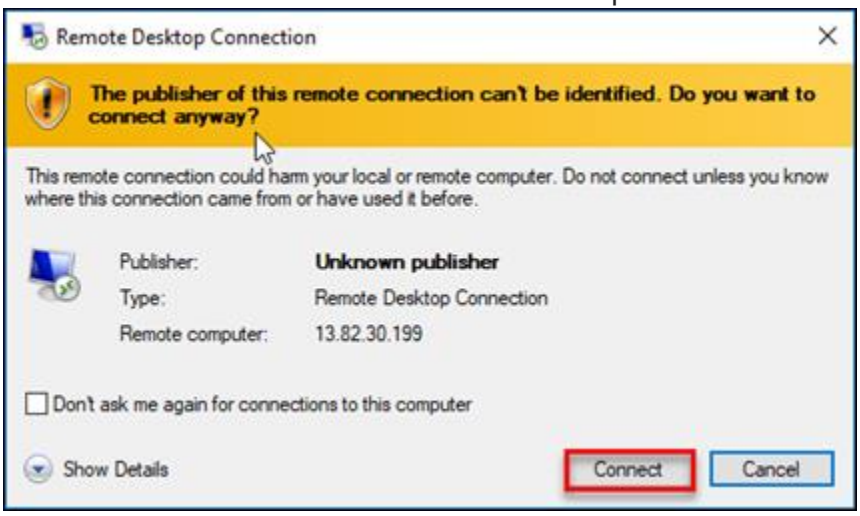
1. Click on resource group icon in Azure portal and select the resource group in which you have create DockerHost in Exercise 2. Under overview seciton, click on Container Host VM.



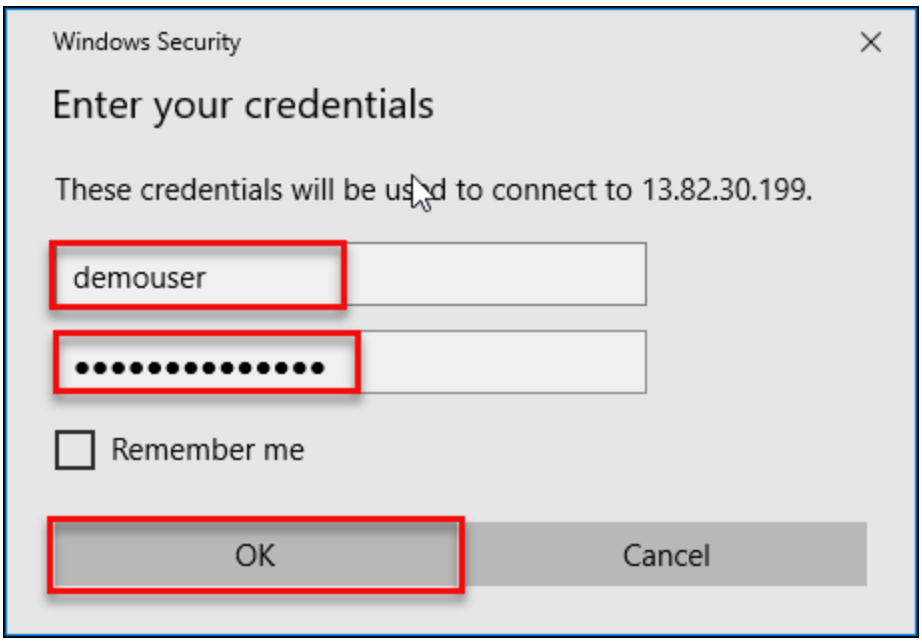
2. In overview section, click on **Connect** button. It will download an RDP file. Click on the same to connect the VM.



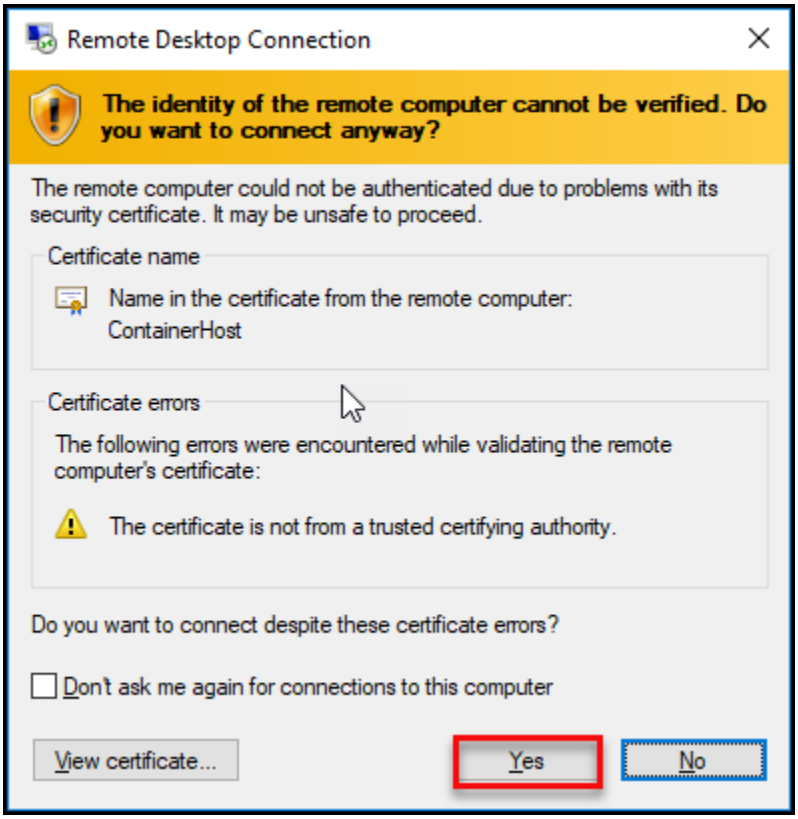
3. Click on **Connect** button in Remote Desktop Connection Wizard.



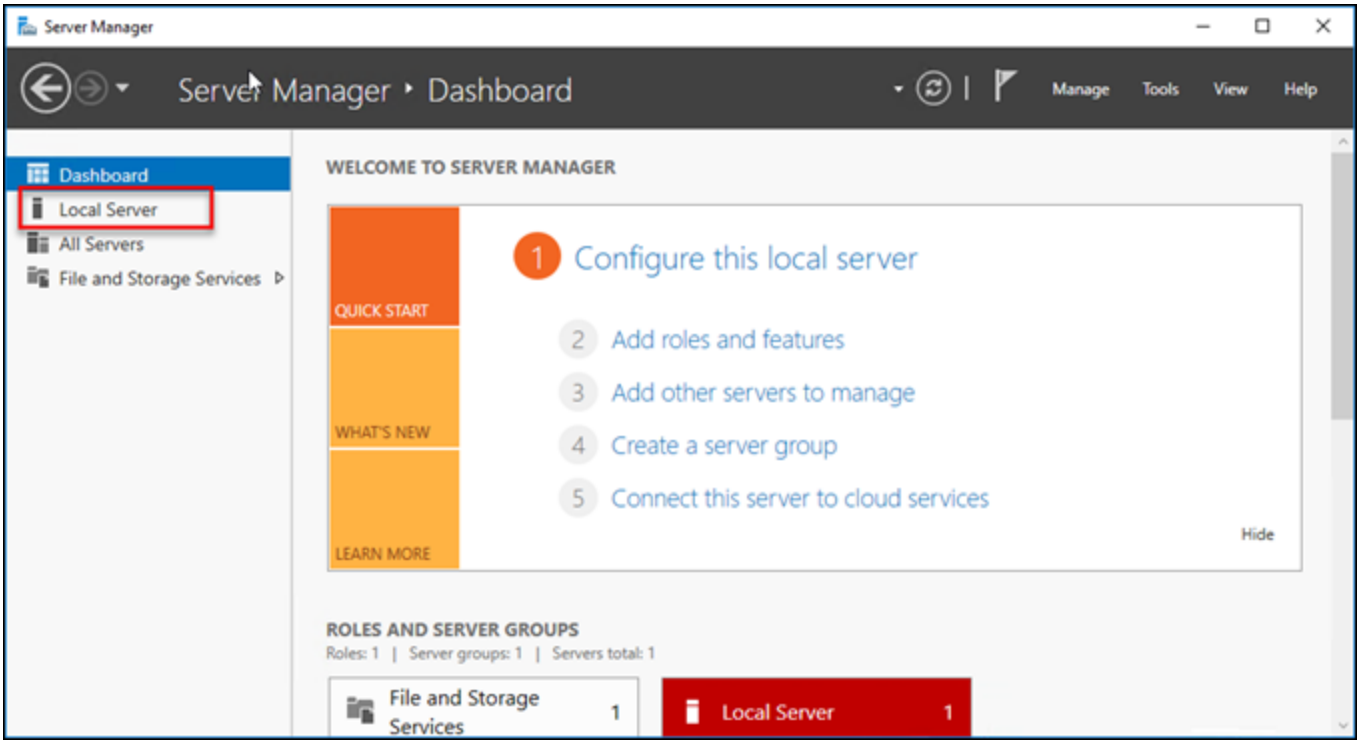
4. Enter the credentials provided in the mail to connect to VM.



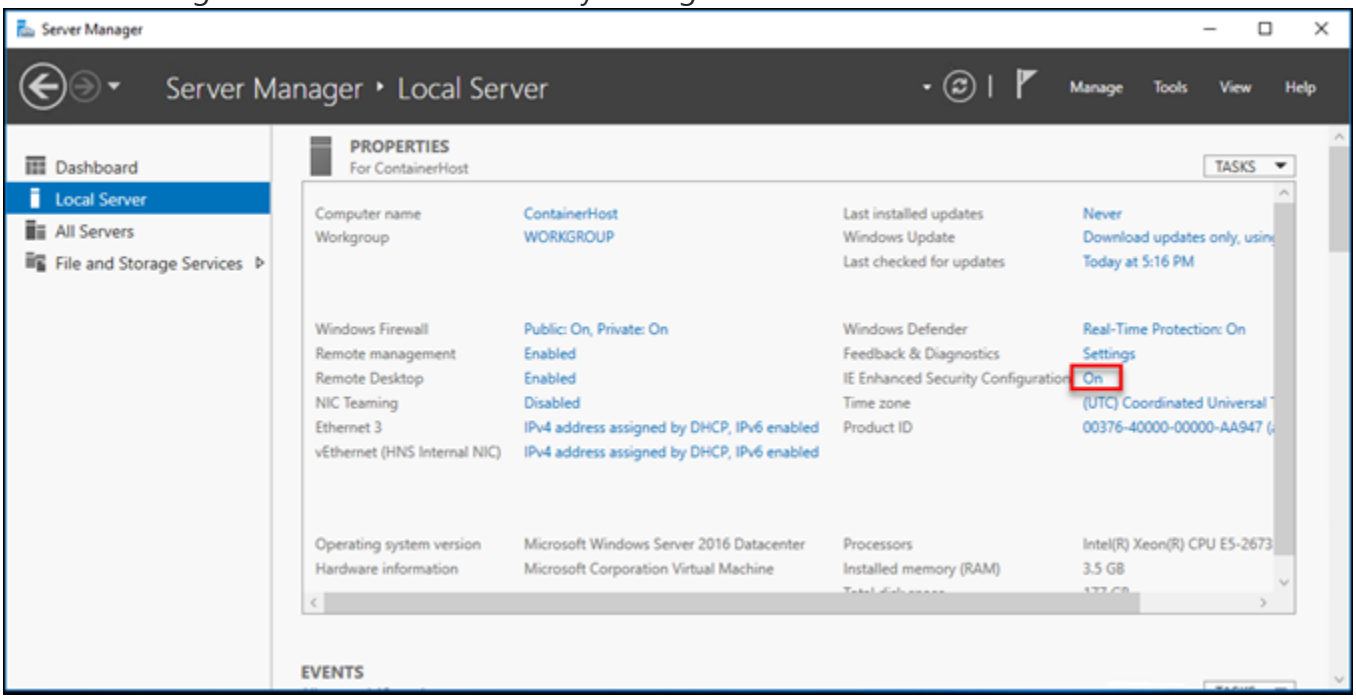
5. Click on **Yes** button.



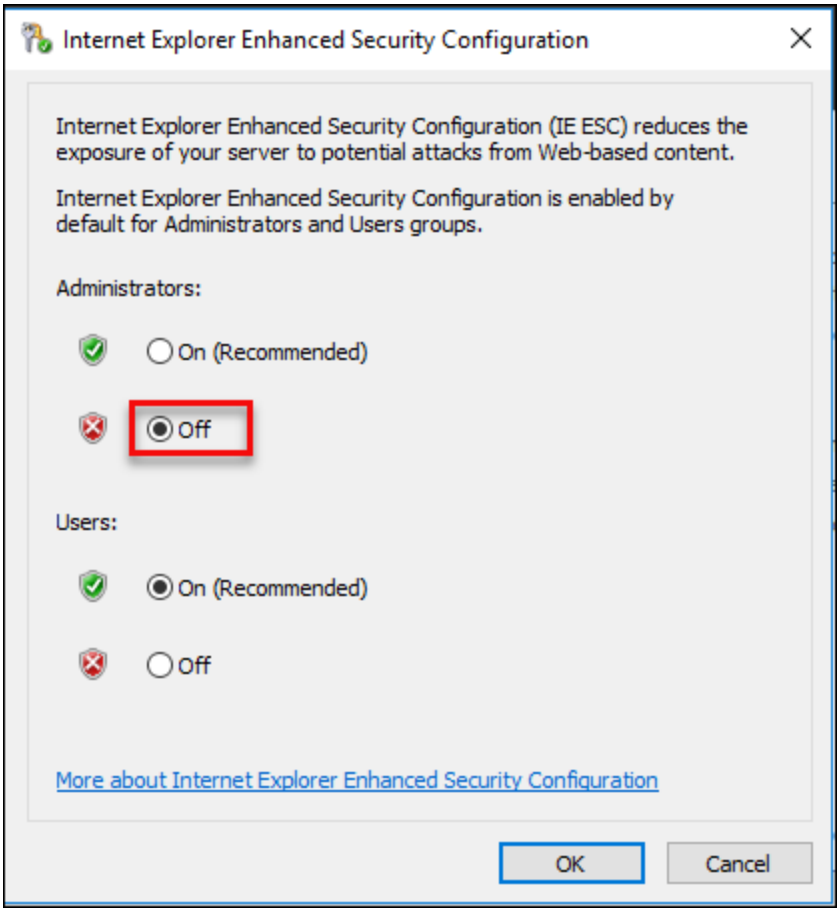
6. Once you are logged into server, you will see Server Manager dashboard. Click on **Local Server**.



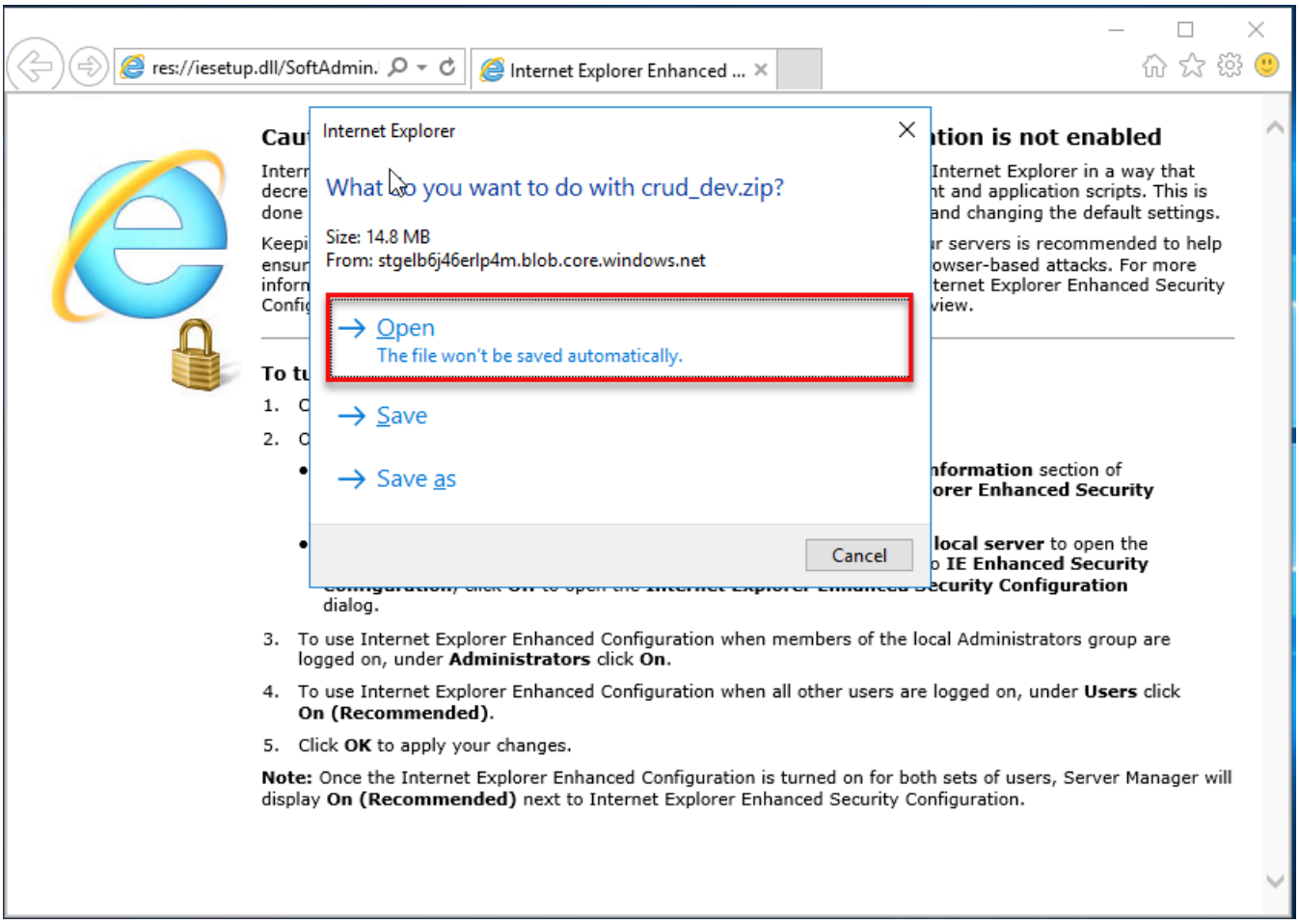
7. Click on **On** against IE Enhanced Security Configuration.



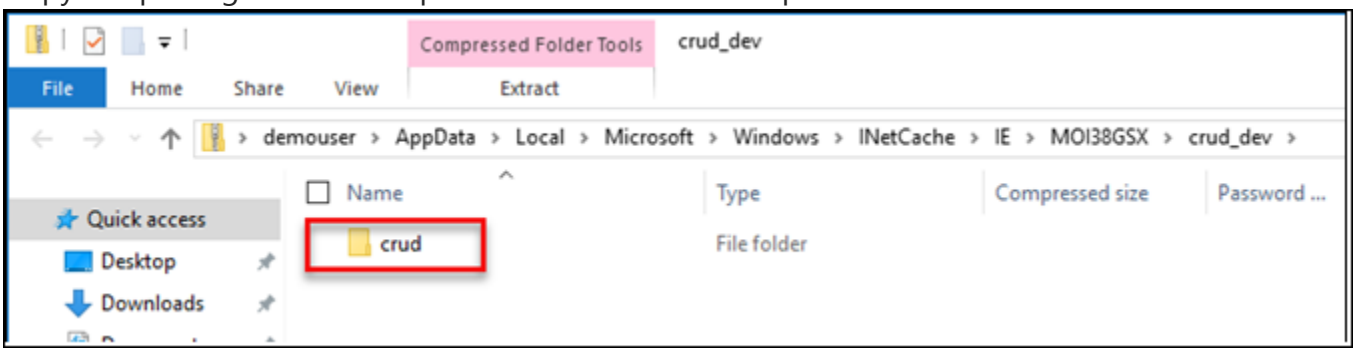
8. In IE Enhanced Security Configuration, select **Off** for **Administrators** and click on **OK**

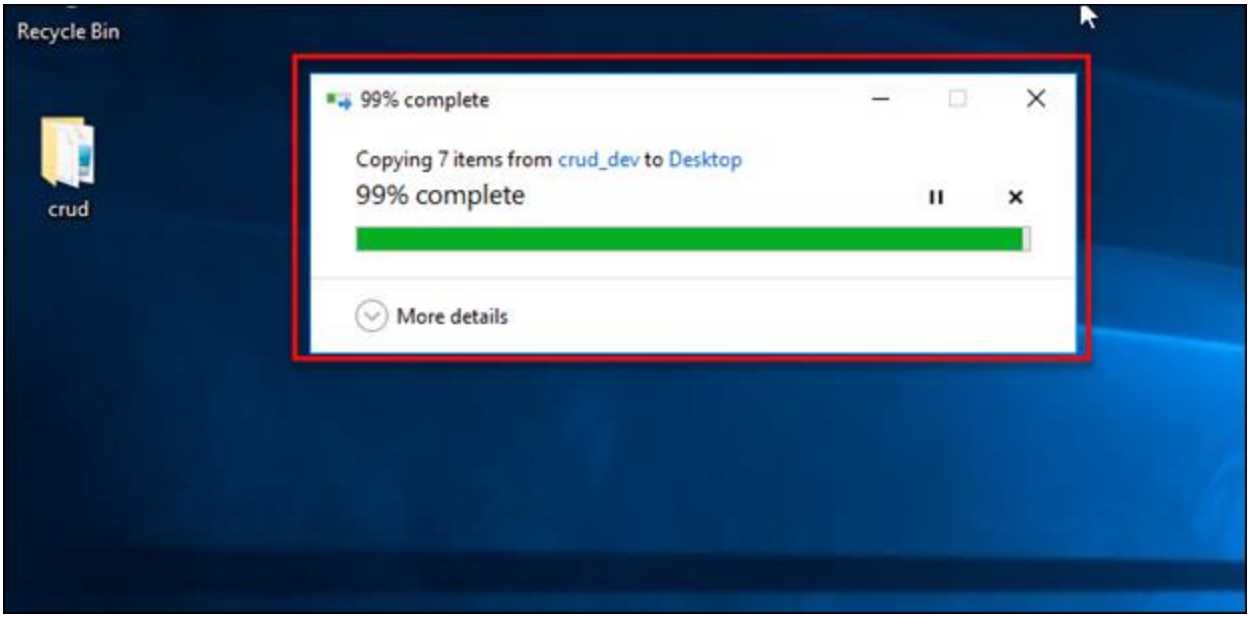


9. Download the package which contains the WebApp deployment package as well as Docker file from this url using IE browser and when prompted, click on **Open**.

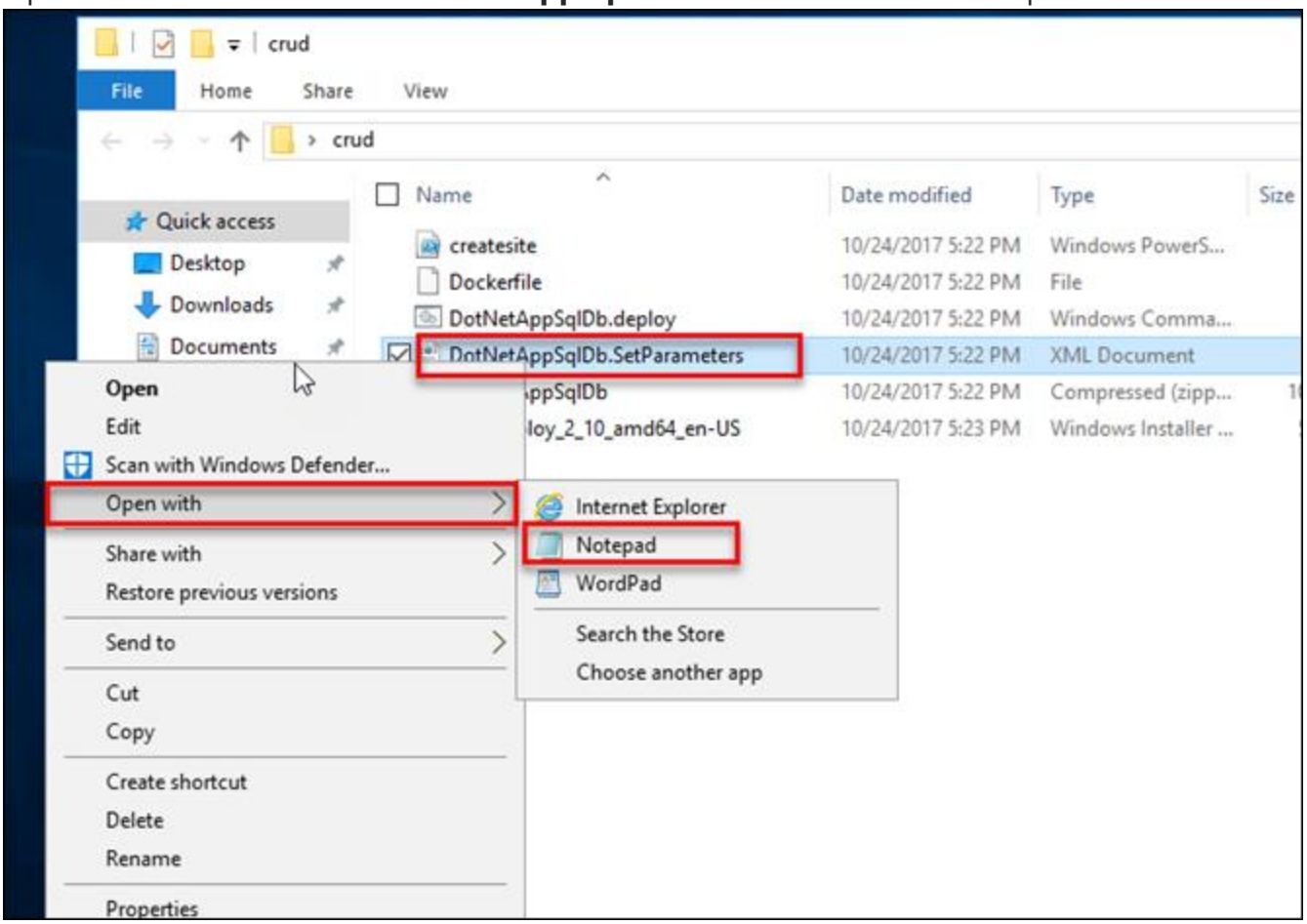


10. Copy the package folder and paste the same on desktop.

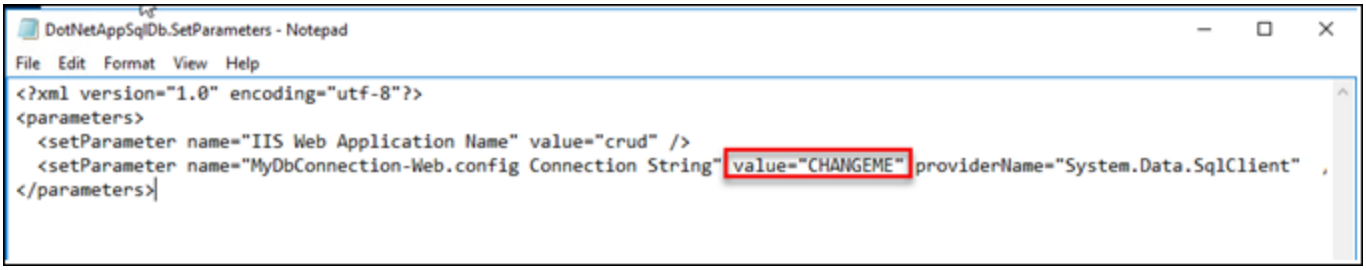


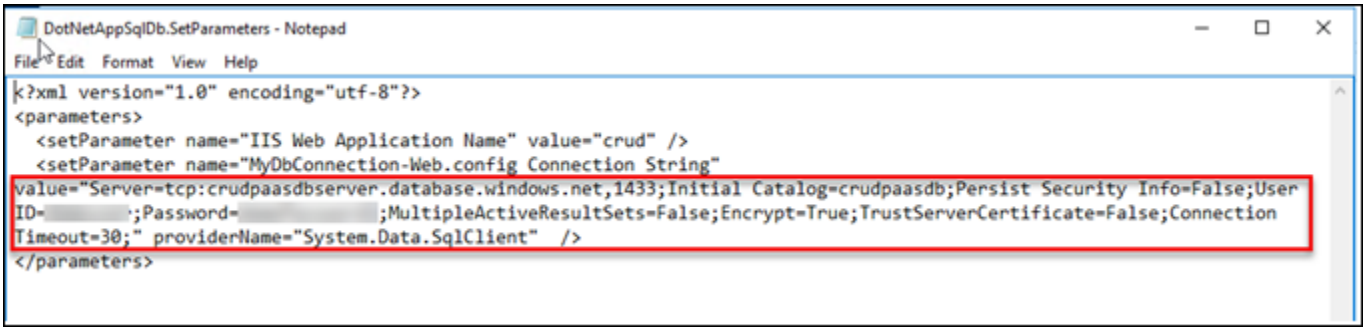


11. Open that folder and edit the **DotnetAppSqlDb.SetParamters** file in notepad as shown below.

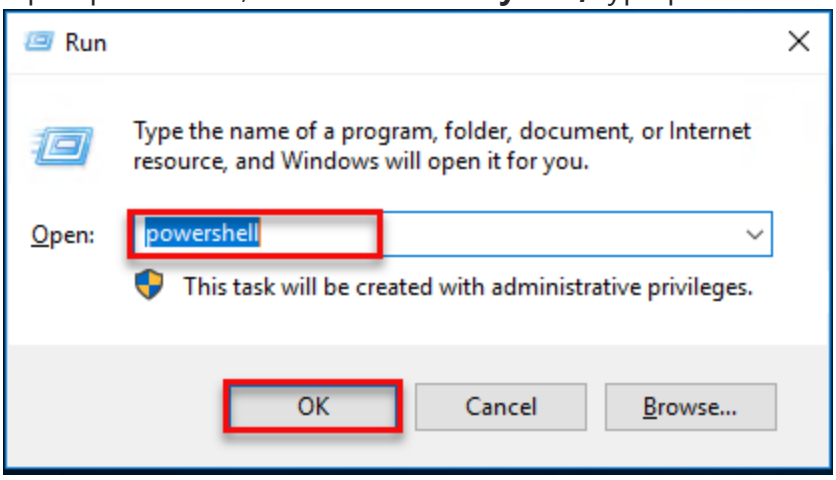


12. You need to change the keywork **CHANGEME** with the database conneciton string you copied in Exercise 2 → step 11.



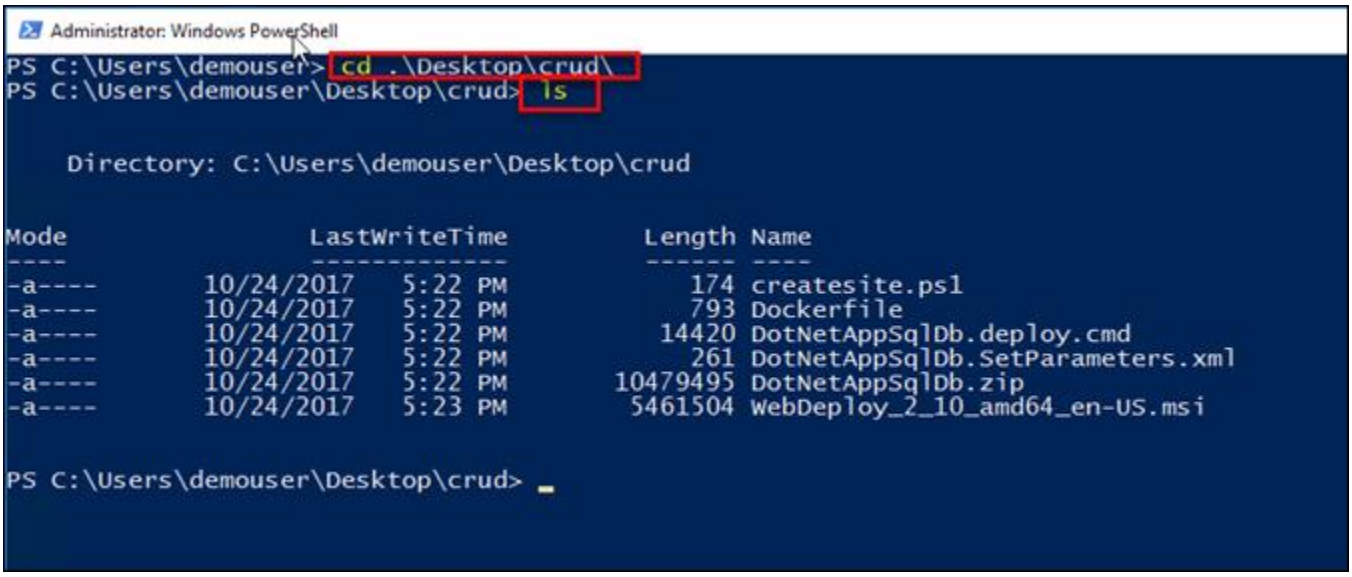


13. Open powershell, click **Window Key + r ,** type powershell and click **OK.**



14. Change the directory to the package folder you copied to desktop by running below command

```
cd .\Desktop\crud
```



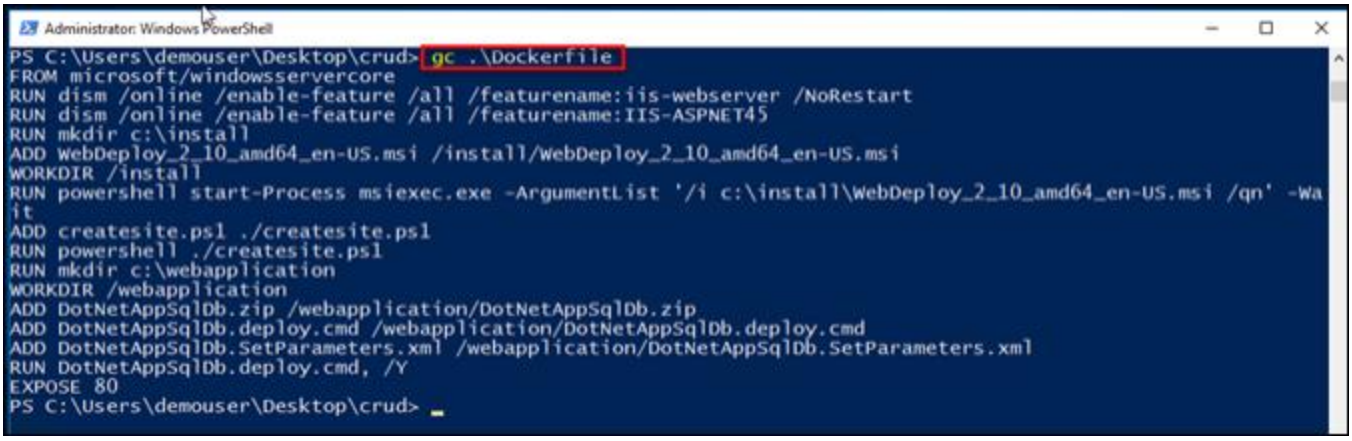
15. Run bleow command to read the content of Docker file.

```
gc .\Dockerfile
```

Content of file shown below:

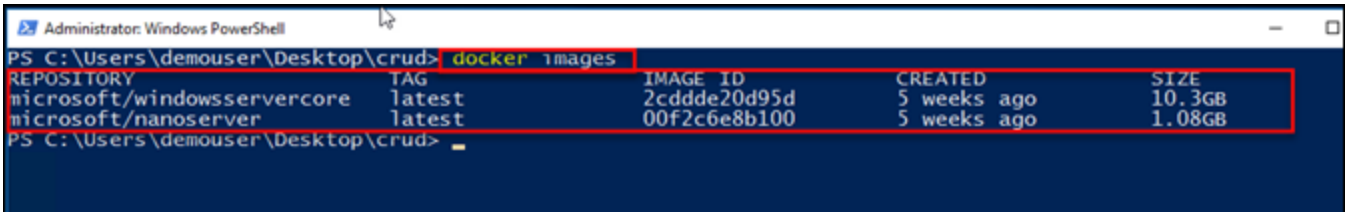
```
FROM microsoft/windowsservercore
RUN dism /online /enable-feature /all /featurename:iis-webserver /NoRestart
RUN dism /online /enable-feature /all /featurename:IIS-ASPNET45
RUN mkdir c:\install
ADD WebDeploy_2_10_amd64_en-US.msi /install/WebDeploy_2_10_amd64_en-US.msi
WORKDIR /install
RUN powershell start-Process msiexec.exe -ArgumentList '/i
c:\install\WebDeploy_2_10_amd64_en-US.msi /qn' -Wait
ADD createsite.ps1 ./createsite.ps1
RUN powershell ./createsite.ps1
RUN mkdir c:\webapplication
WORKDIR /webapplication
ADD DotNetAppSqlDb.zip /webapplication/DotNetAppSqlDb.zip
ADD DotNetAppSqlDb.deploy.cmd /webapplication/DotNetAppSqlDb.deploy.cmd
ADD DotNetAppSqlDb.SetParameters.xml
/webapplication/DotNetAppSqlDb.SetParameters.xml
RUN DotNetAppSqlDb.deploy.cmd, /Y
EXPOSE 80
```

```
Note:Explanation of Dockerfile
1. It is using windows server core container as base image.
2. Installed IIS-WebServer feature and IIS-ASPNET feature.
3. Installed MS WebDeploy software to deploy Website which is supplied as
   WebPackage from Visual Studio. Same package is used to deploy app in On-
   premises VM also.
4. Crearesite.ps1 is script is used to create website in IIS.
5. DotNetAppSqlDb.deploy.cmd uses MS webdeploy to deploy application in IIS.
```



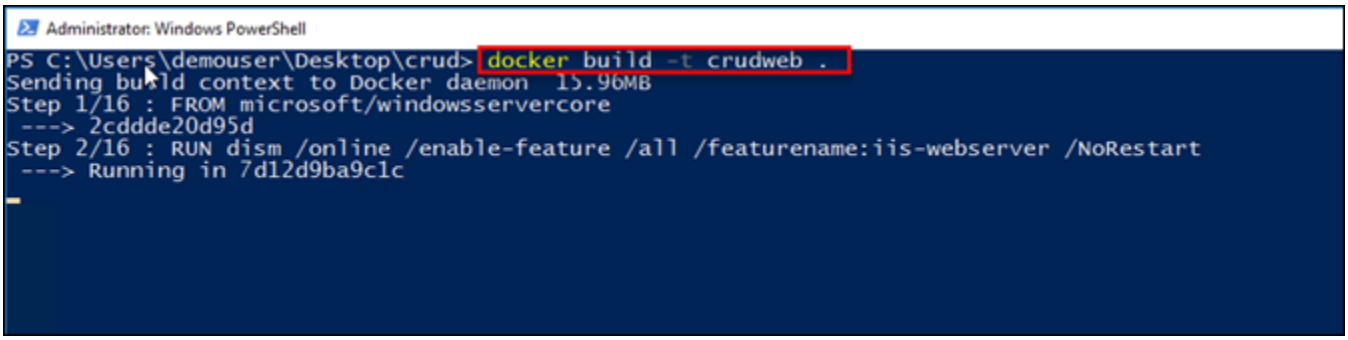
16. Run below command to list down the images which are already present in Docker Host.

```
docker images
```

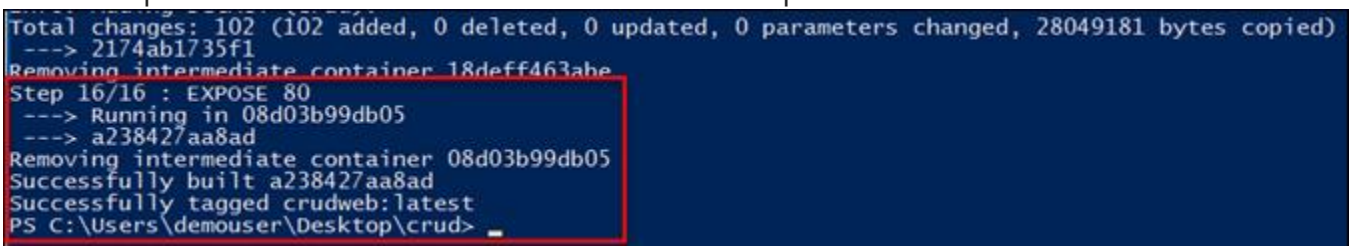


17. Run below command to build the image using the given Docker file.

```
docker build -t crudweb .
```



18. Above step will take 8 to 10 minutes for successful completion.



19. Run the below command again to see the list of images which will show newly created image.

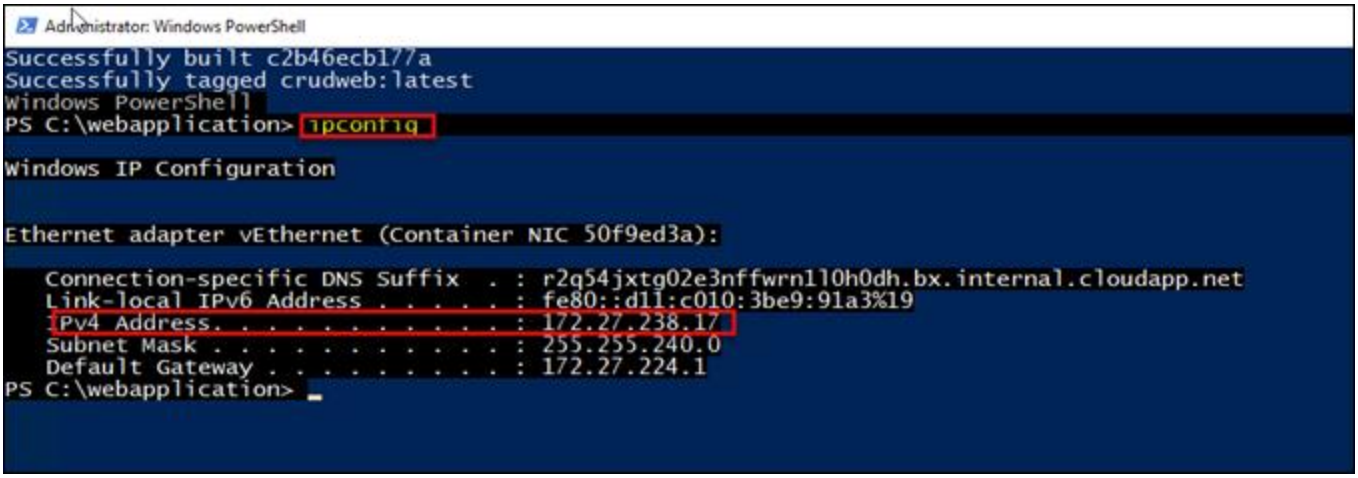
```
docker images
```

20. Run the below command to launch the container using the image which you created in above step.

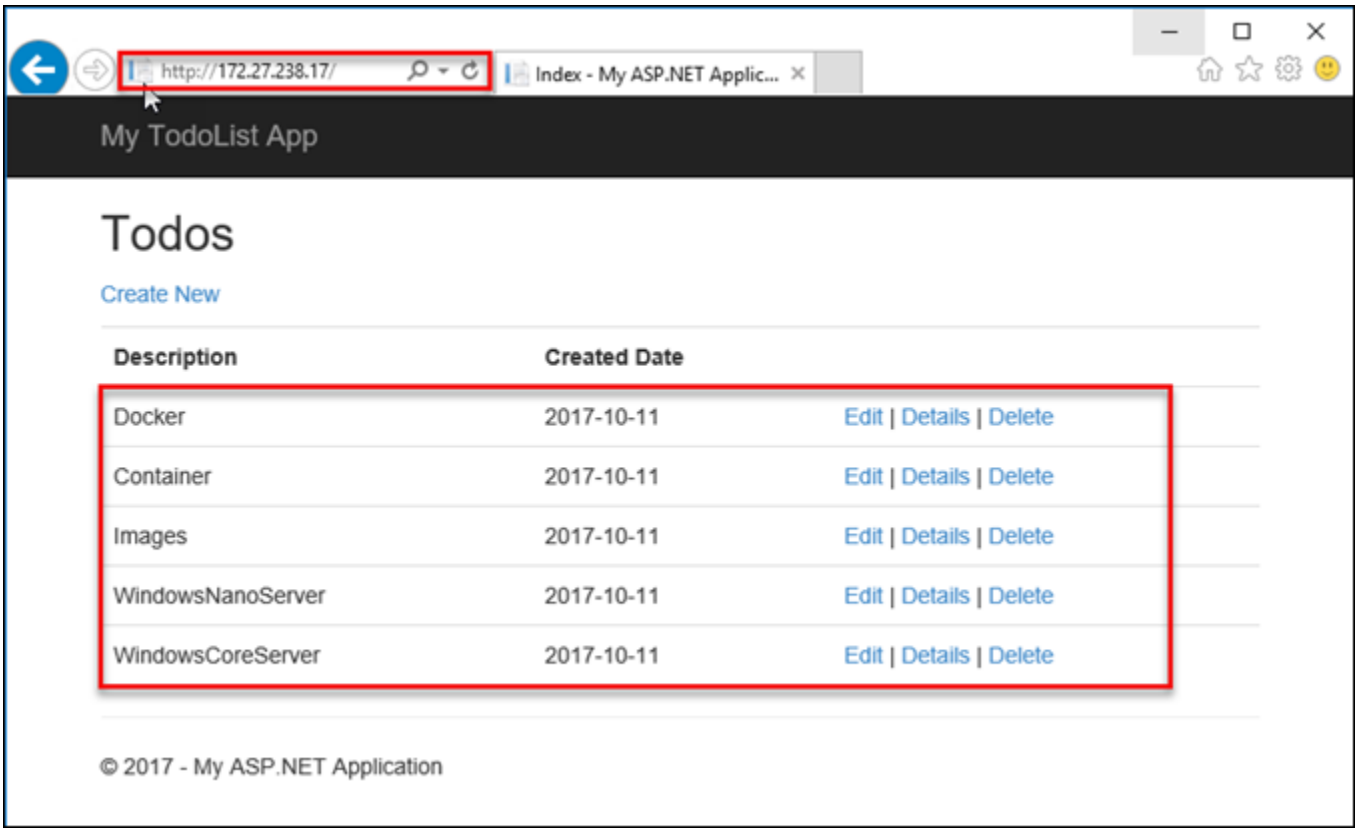
```
docker run -it crudweb powershell.exe
```

Once container is launched, run below command in Container Powershell to get the ip address.

```
ipconfig
```



21. Open the IP address from above step in browser inside docker host VM and verify that it does have the records which you created in your on-premises VM before migration.



*** This ends the lab.***