# Objective:

The purpose of this document is to demonstrate how to **migrate an on premise SQL Server based database to SQL Server running on an Azure VM.**

# Prerequisites:

* This demonstration uses an already built SQL Server database named Adventure Works. This database can be downloaded from [here](http://msftdbprodsamples.codeplex.com/releases/view/37304).
* SQL Server 2014 (Standard) has been used in this demo as on premise SQL Server
* Valid Azure Subscription

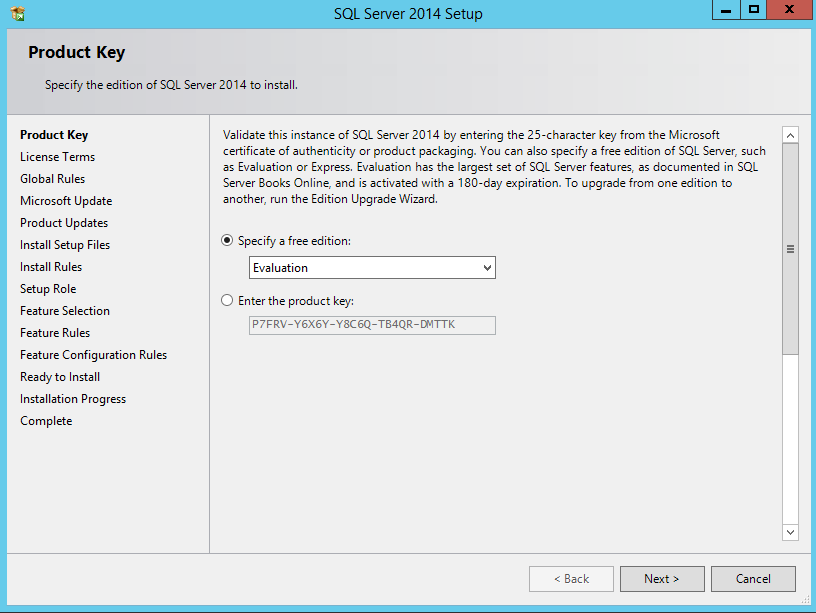
# Steps Overview:

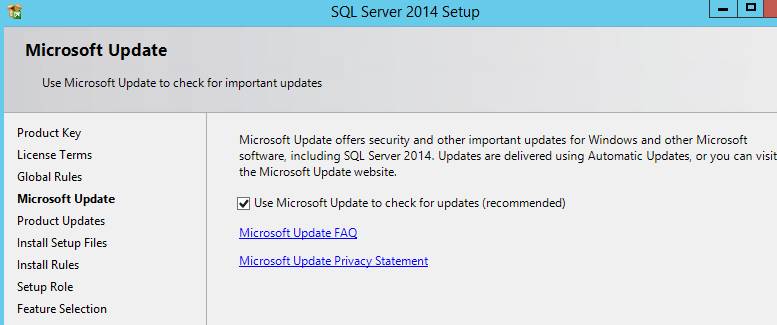
* Setting up On-Premise Database
* Setup SQL Server on an Azure VM
* Migrating database from on premise SQL Server to SQL Server on Azure VM

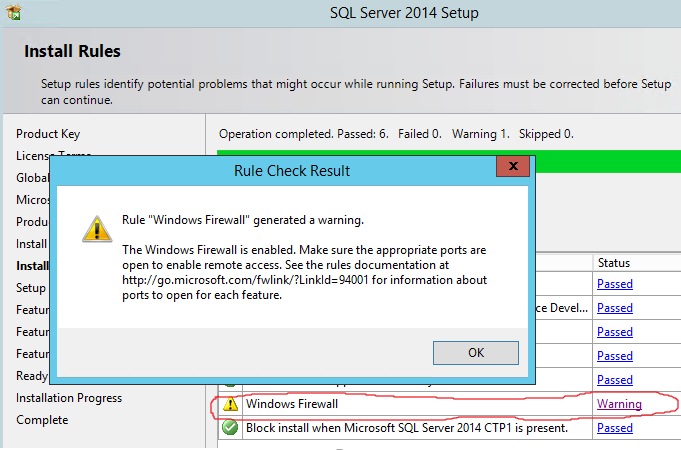
Note: There are many techniques to migrate database in this scenario. However we are using SQL Server Migration Wizard for the purpose of this demonstration.

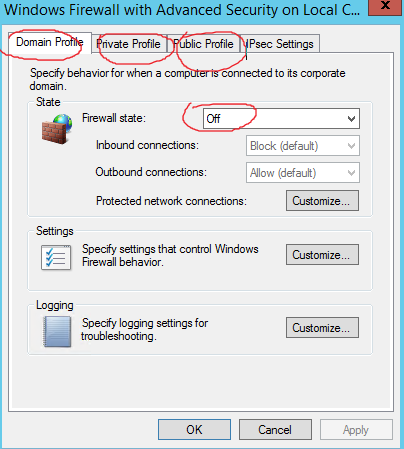
# Setting up On-Premise Database

* Make sure that SQL Server 2014 is installed on a local machine. We’ve used SQL Server 2014 Standard Edition for this demonstration.
* Screenshots of important steps during installation are shown below.

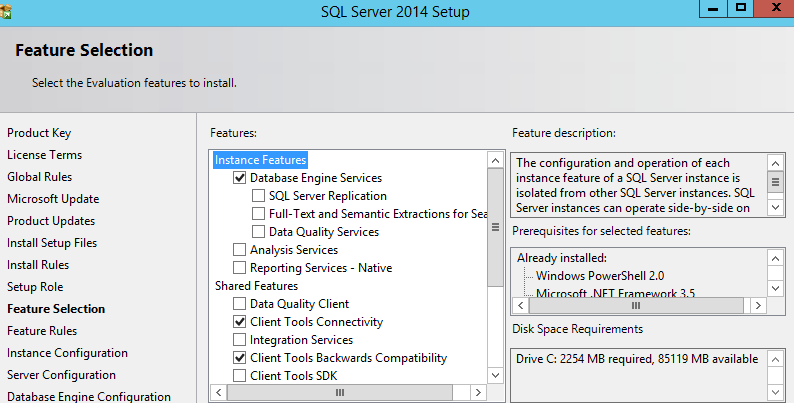


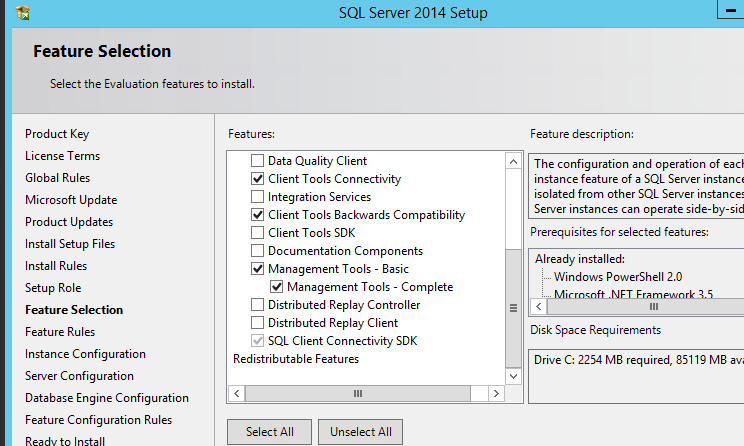


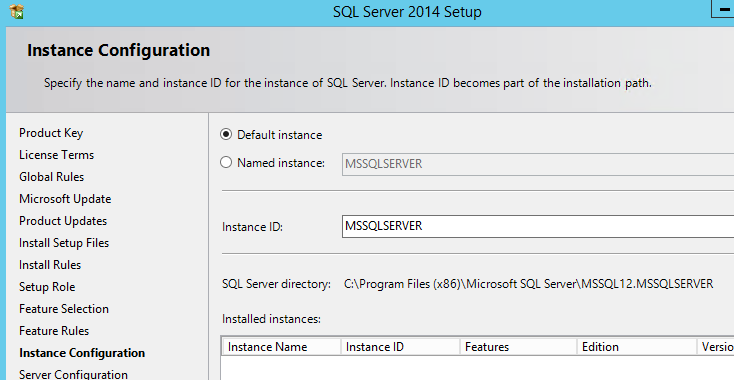


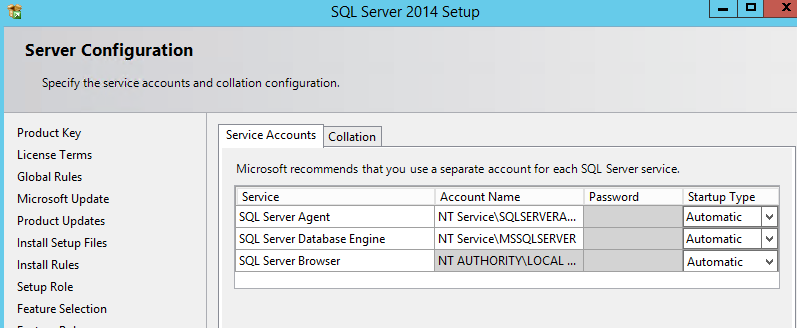


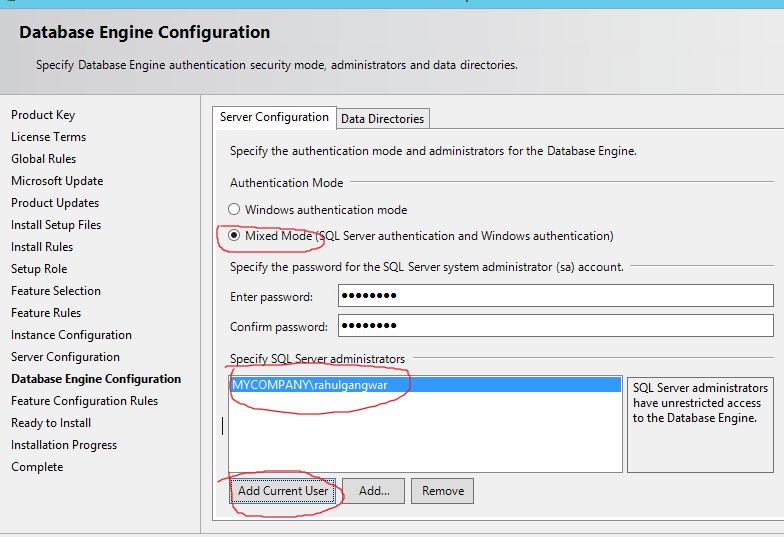
* You can either add exceptions to SQL ports (UDP/TCP) in host firewall or you can temporarily disable it. As shown above, I chose to disable it for demonstration. I’ve disabled it for ALL firewall profiles (Domain, Private & Public).



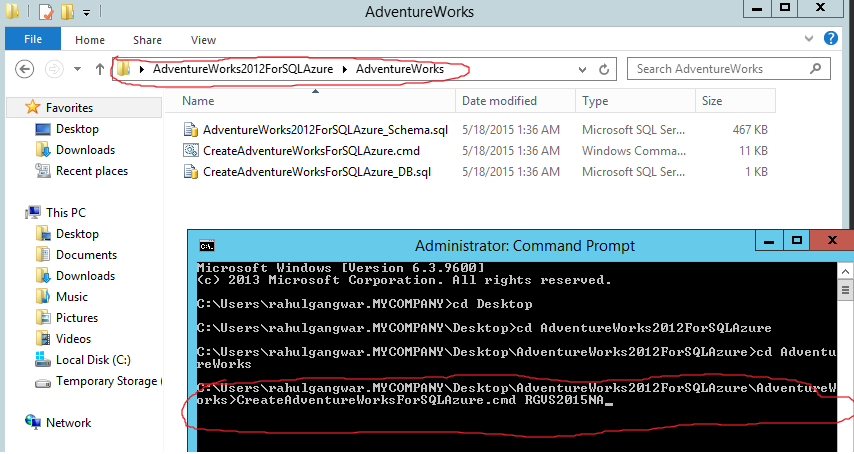




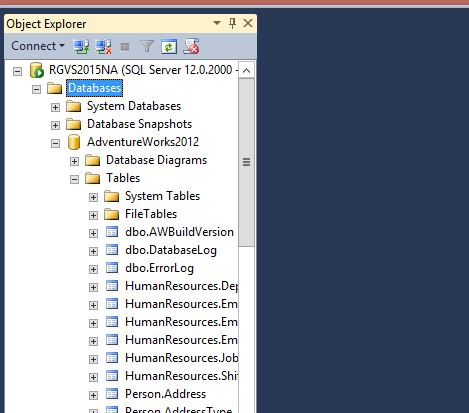




* Download and extract Adventure Works Database on local drive.
* Open command prompt in admin mode and navigate to “AdventureWorks” folder within the extracted folder.
* Run command **CreateAdventureWorksForSQLAzure.cmd** *servername* *username* *password* where *servername* is your local database server name (given all defaults were chosen during installation) and username/password is *<<sa>>* and *<<sa account’s password>>*. Recall that we created this user during installation. Screenshot shown below without username/password.



* After this you can verify the database successfully created from Management Studio. Make sure to verify this because no matter what error you get, the command prompt will show installation completion message at the end which is confusing. You need to scroll up to see exact error messages in case database creation failed.



* Congratulations! On premise DB is now successfully setup.

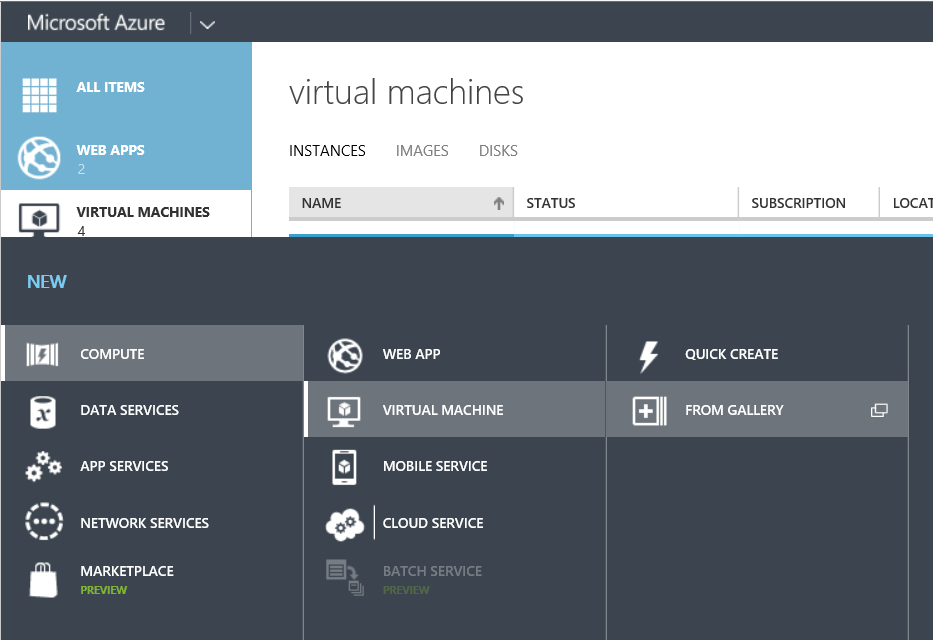
# Setup SQL Server on an Azure VM

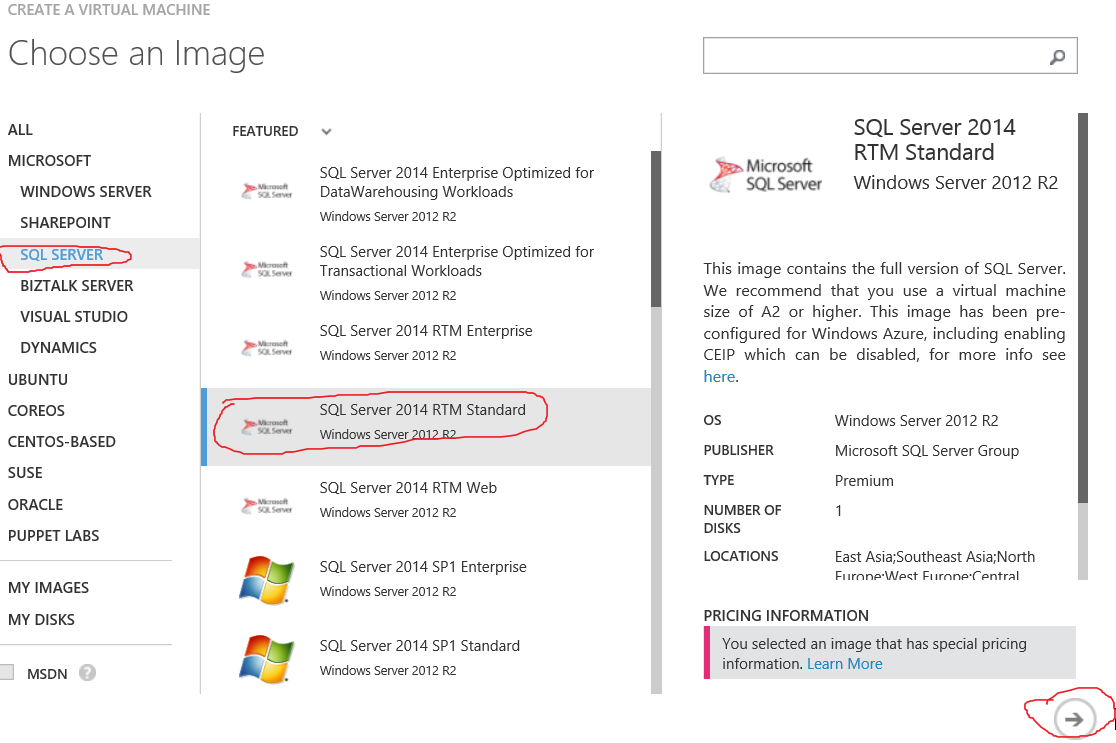
Below are the steps performed to set up a SQL Server VM on Azure.

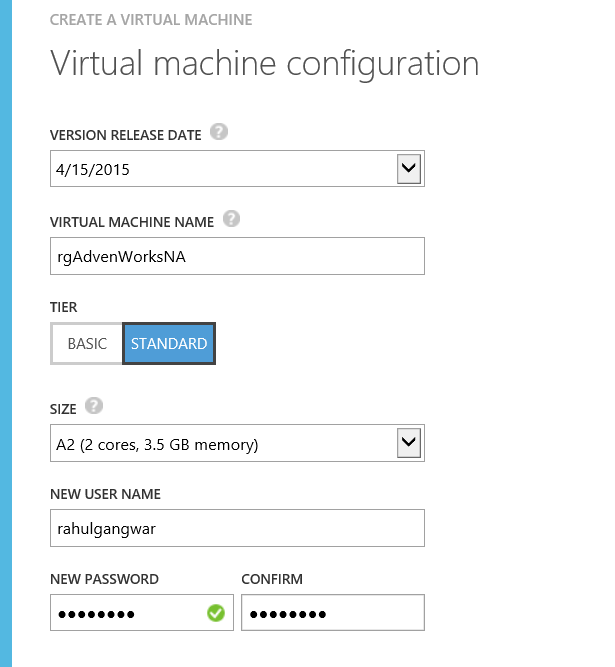
**Note:** The steps below show how to migrate your database to SQL Server VM on Azure by using the Azure Management Portal and the tool SQL Database Migration Wizard. If you have SQL Server 2014, you can have a similar migration completely within SQL Server Management Studio. See more details at: [SQL Server 2014 - Deploy a SQL Server Database to a Windows Azure Virtual Machine](https://msdn.microsoft.com/en-us/library/dn195938.aspx)

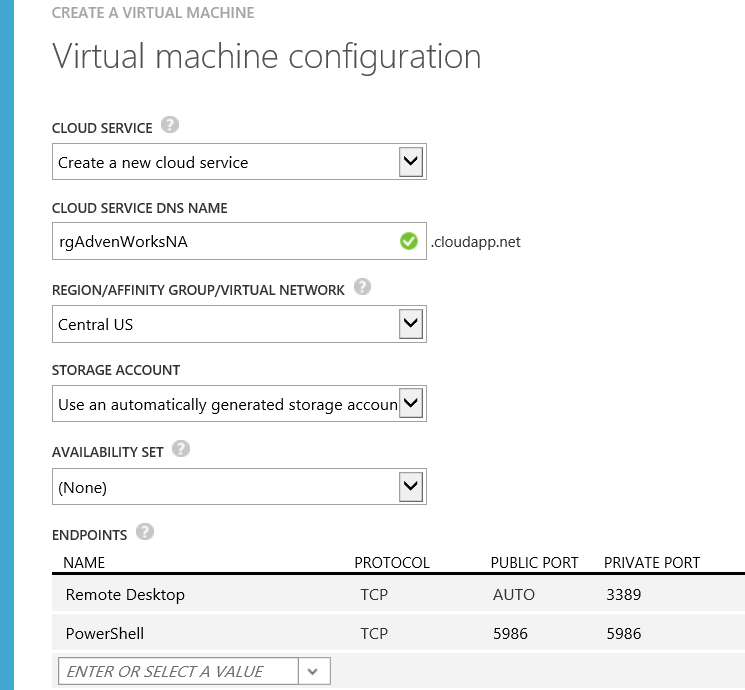
## Creating SQL VM

* Navigate to Azure portal at manage.windowsazure.com and select “Virtual Machines” from left menu and then following New 🡪 Compute 🡪 Virtual machine 🡪 From Gallery and then following the steps of wizard to select a SQL based VM as shown below.

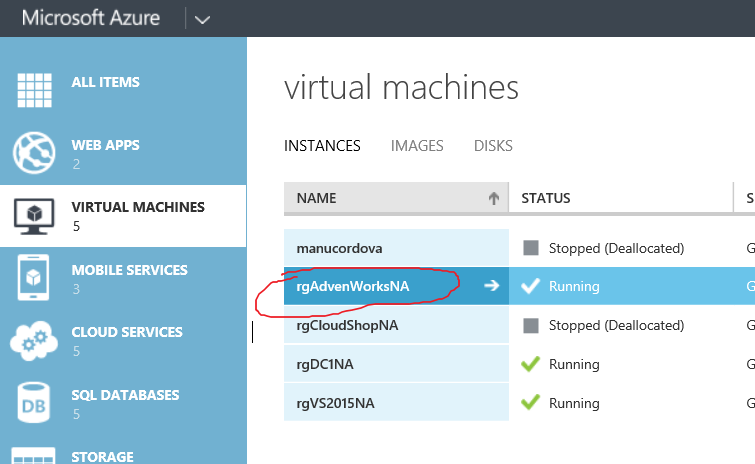




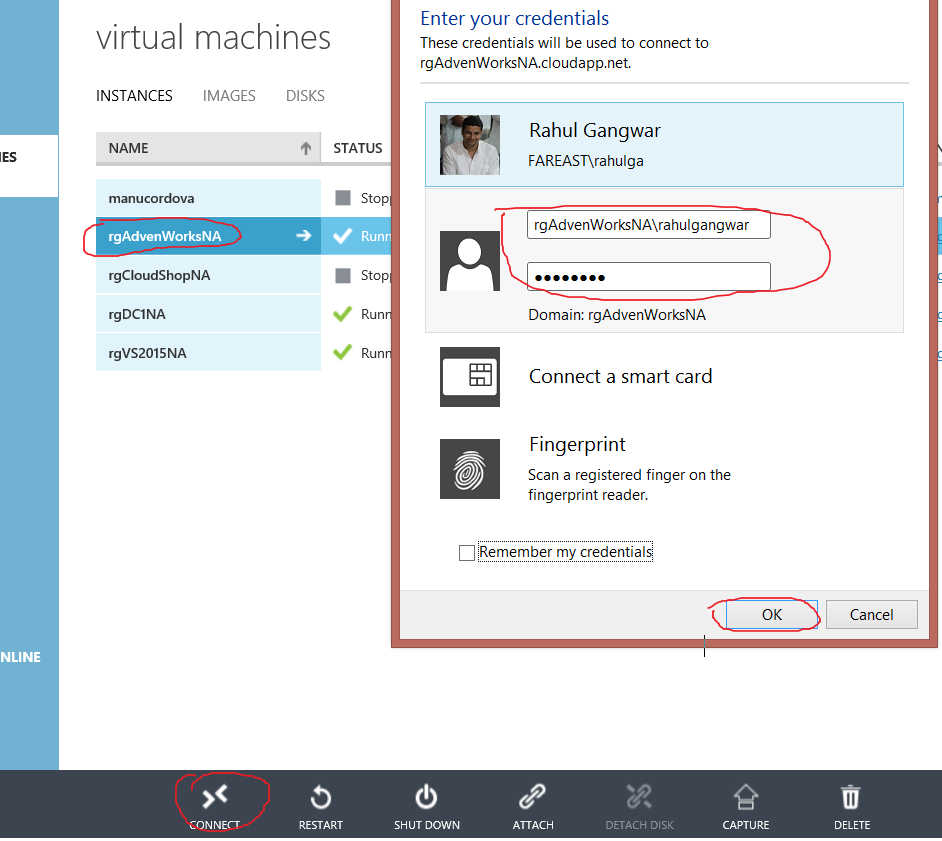




* Above steps should provision a new SQL Server VM on Azure as shown below.



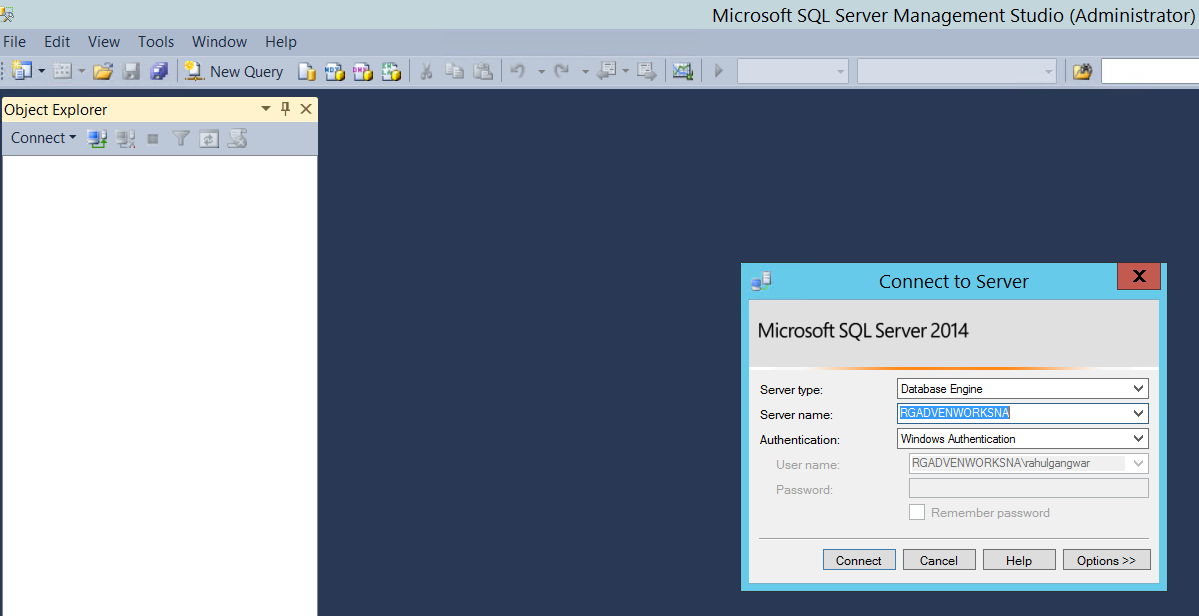
* SQL server is already installed on this VM. You can verify by first connecting to the VM using remote desktop as shown below.



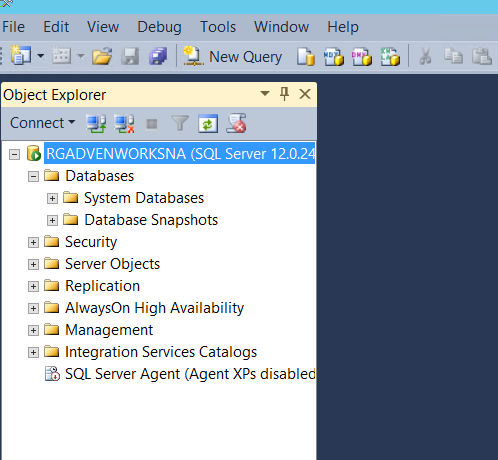
* Once connected, search for SQL Management Studio as shown below



* Login to default instance as shown below.

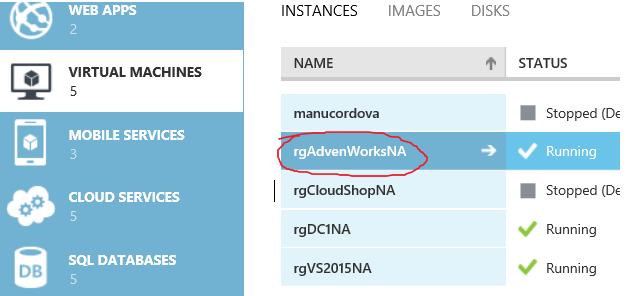


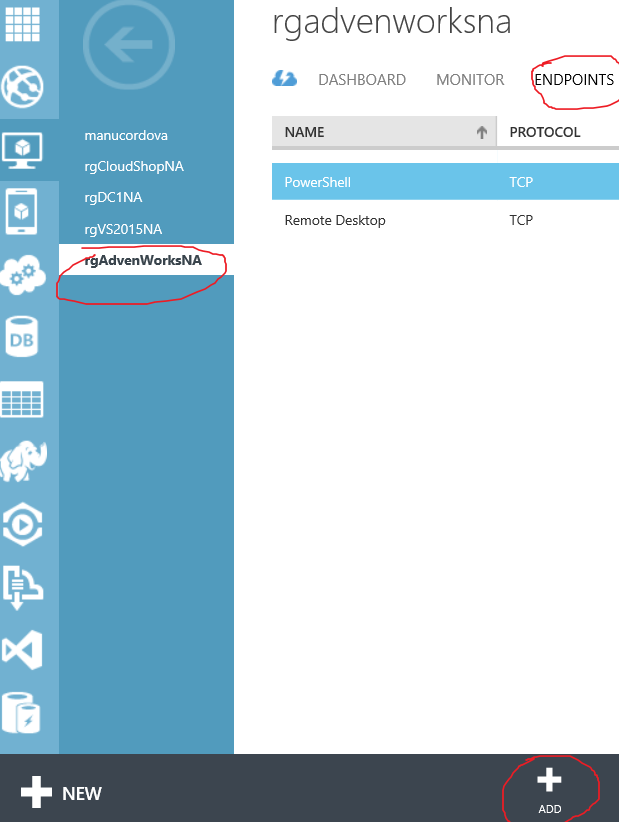
* We can see default databases under this instance as shown below.

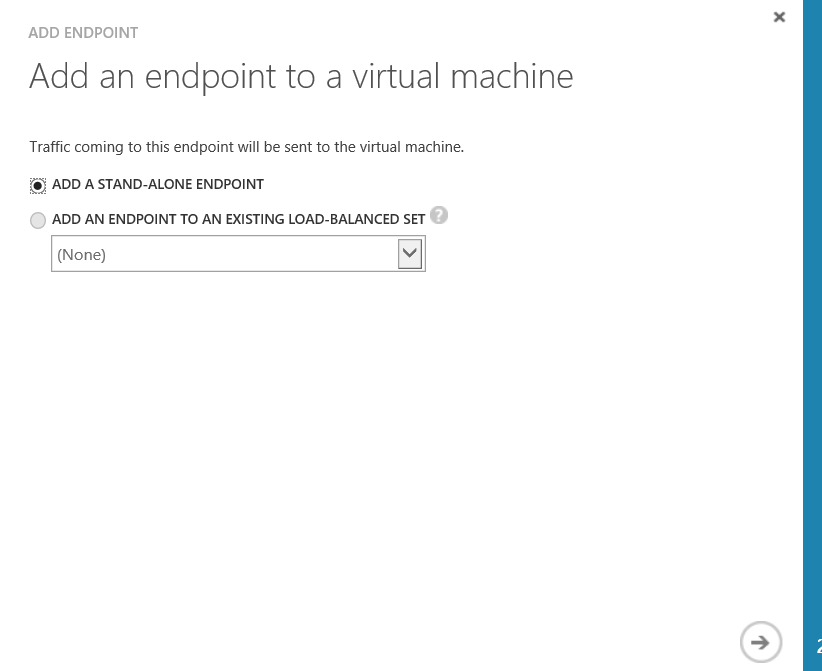


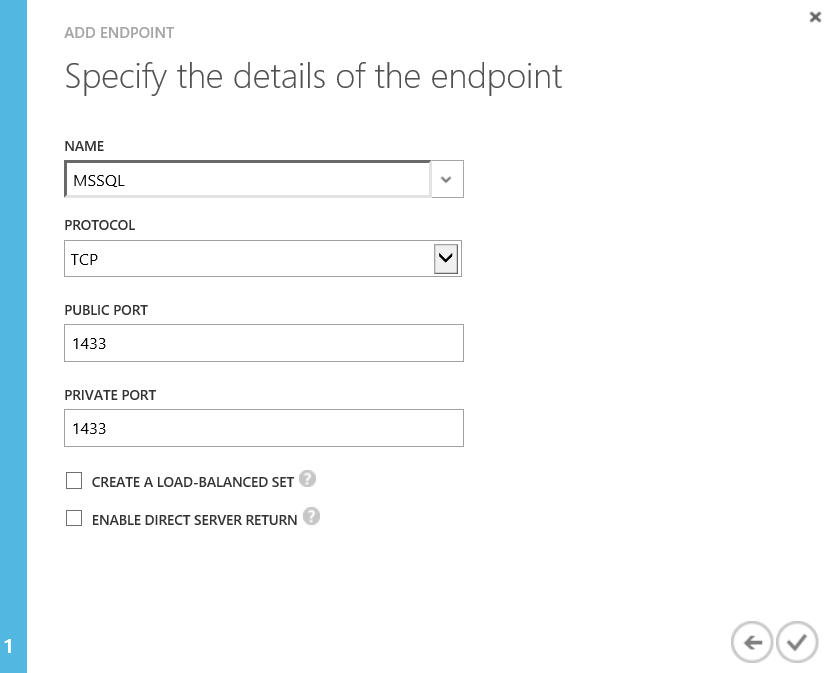
## Configuring Endpoints

* Now that SQL VM is created, we need to configure Azure endpoints to allow SQL connections from public internet.
* For this follow the below steps.

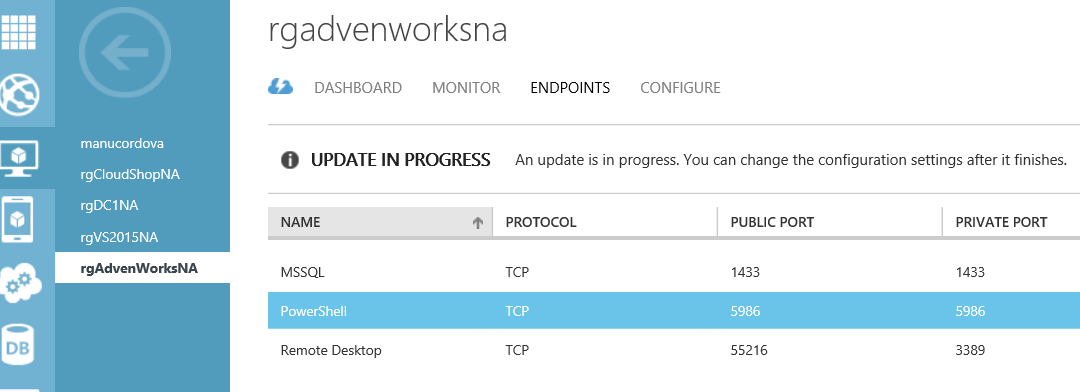








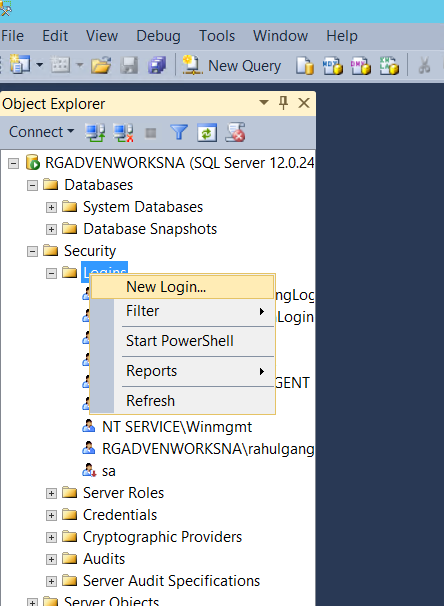
* Once above steps are completed, you must see a new endpoint created under the SQL VM a shown below.

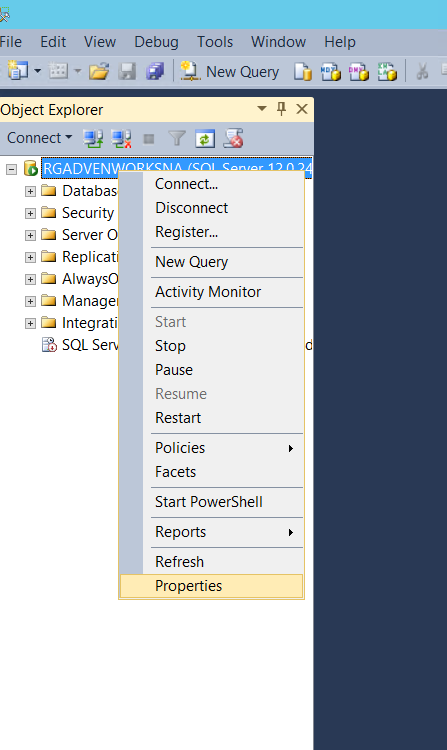


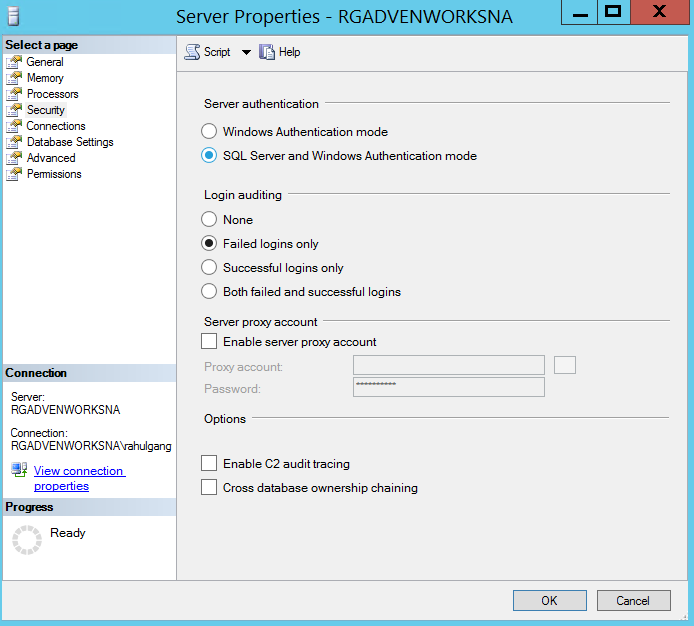
## Configuring SQL Server

### Enable Mixed Mode Authentication

* Out of the box when SQL Server is created, it is configured to use Windows Authentication. So we need to enable it to use mixed mode authentication. This can be done by following below steps.
* In SQL Server Management Studio login using windows authentication and then from Object Explorer, right-click the server, and then click **Properties**.
* On the **Security** page, under **Server authentication**, select the new server authentication mode, and then click **OK**.
* Restart the server. Steps shown below.

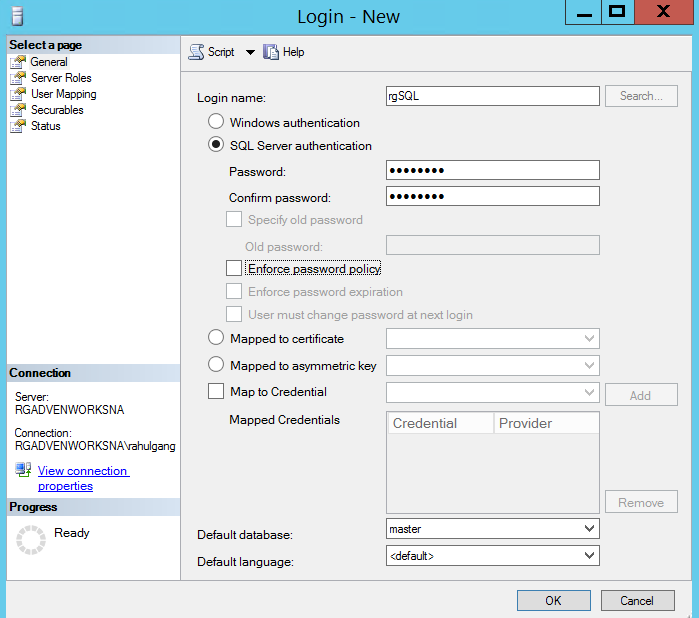


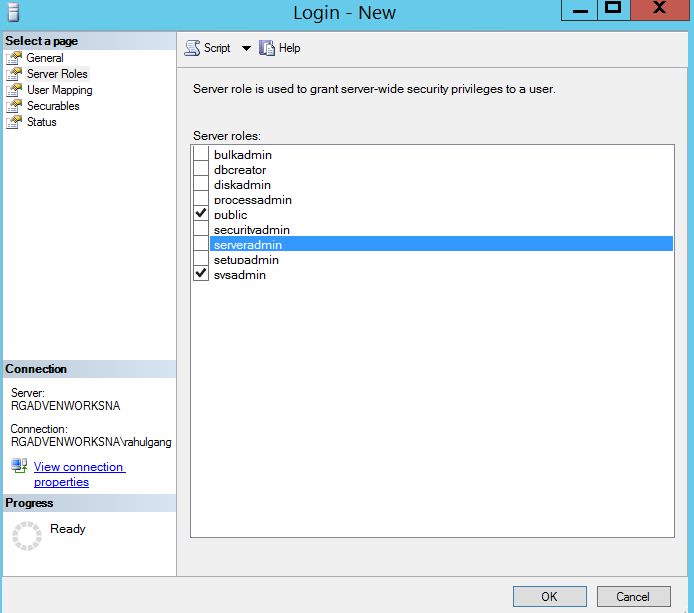




### Create a new SQL Server Login

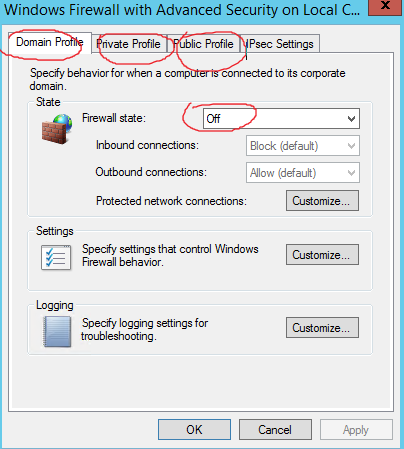
* Once we’ve enabled mixed mode authentication, we can either create a new SQL Server login or we can enable and reset an inbuilt account “sa”. In below steps we’ve created a new login named “rgSQL”. We’ll use this account to connect from migration wizard.





### Configure Firewall

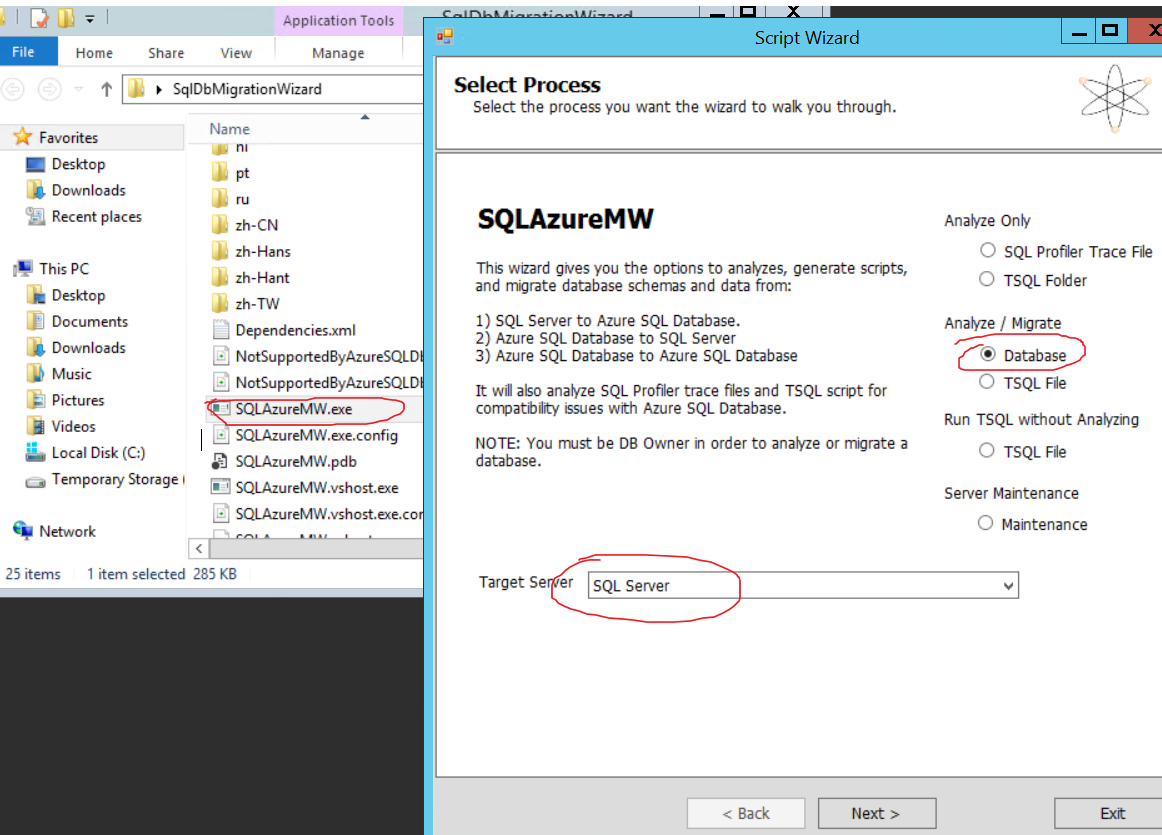
* We also need to enable inbound TCP port 1433 on Azure SQL VM. We can delete/disable this rule after we have completed migration. I have disabled the firewall for time being a shown below:

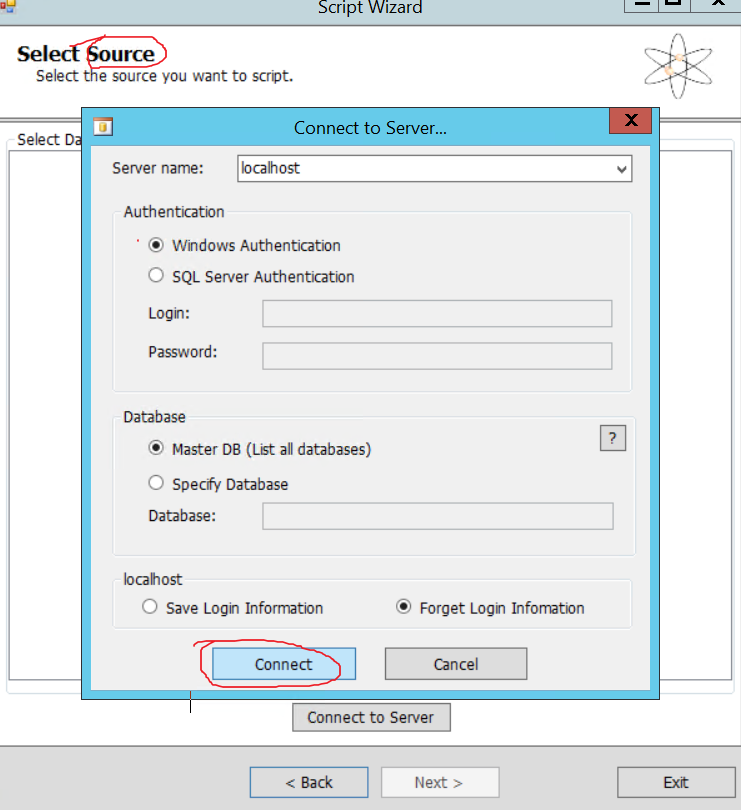


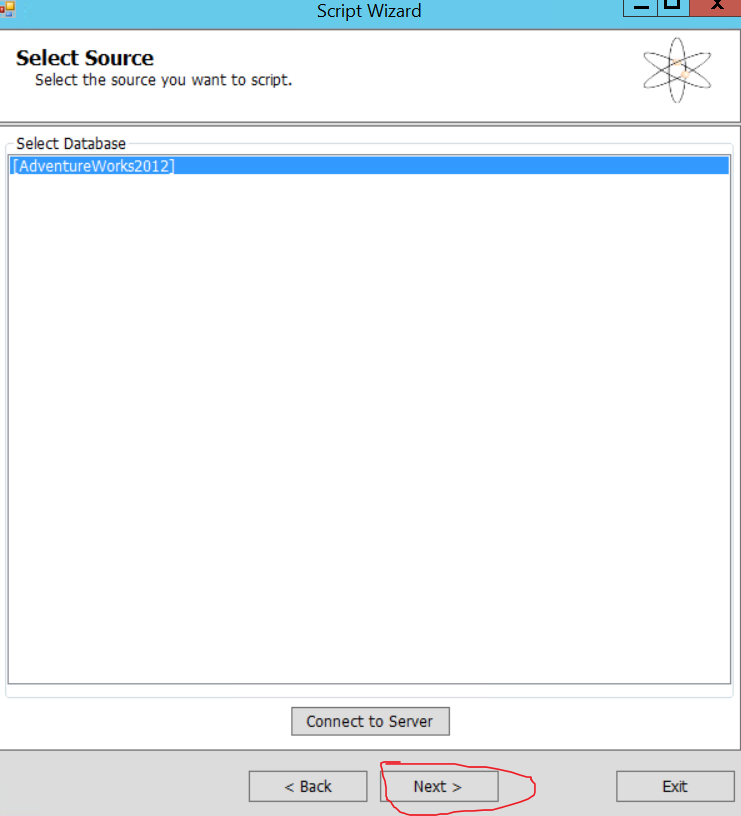
# Migrating database from on premise SQL Server to SQL Server on Azure VM

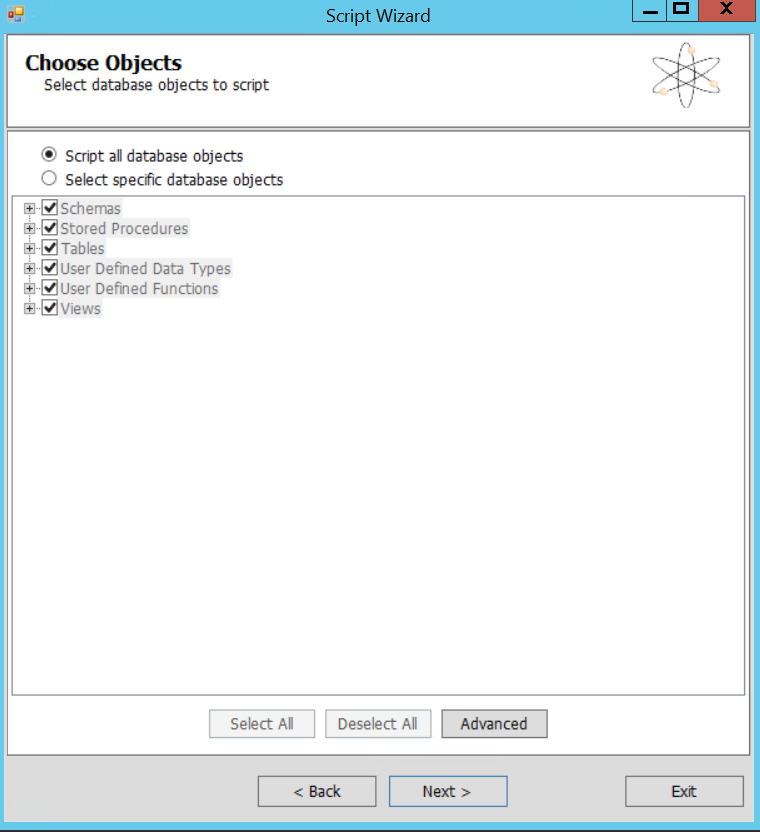
* Now that endpoints are configured and firewall setup, we are ready to migrate database.
* Run SQL Database Migration Wizard from on premise SQL VM and follow steps as shown below.

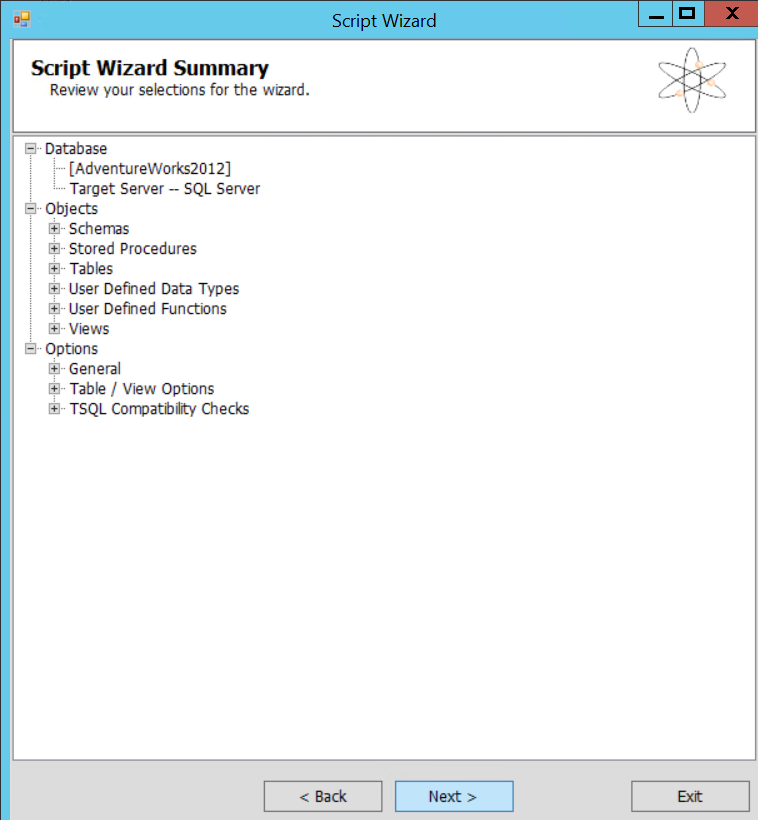
### Connect to source database (on premise server).

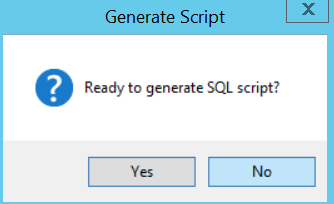


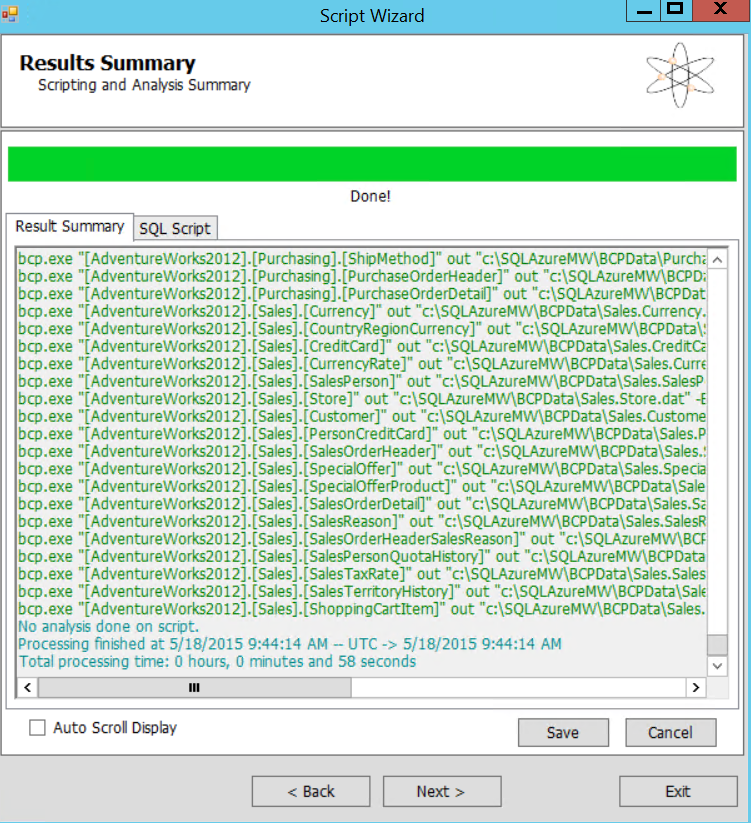








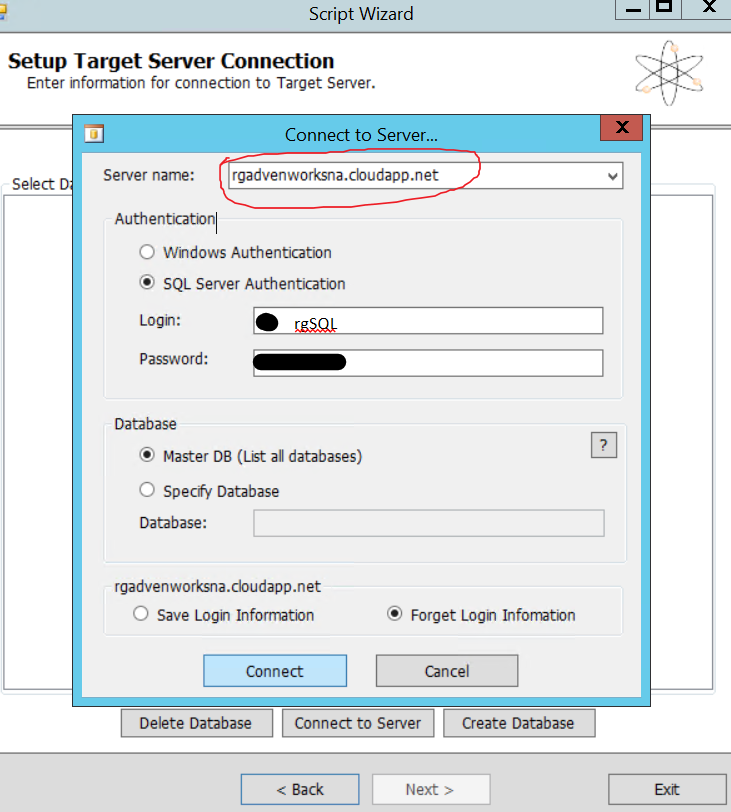




### Connect to target database (Azure SQL VM).

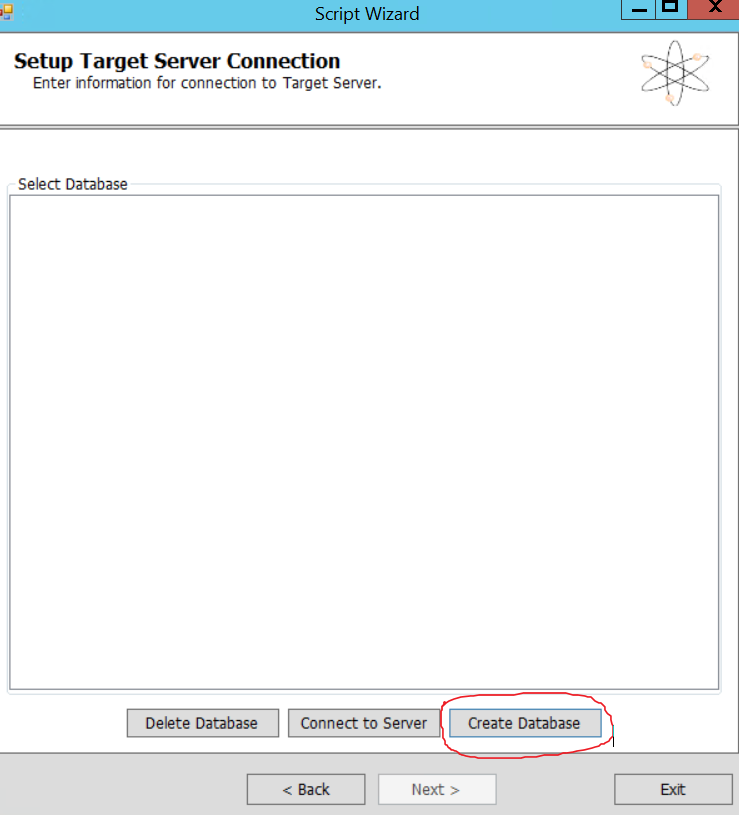
* You can obtain the name of target VM from portal as shown below. Here it is rgadvenworksna.cloudapp.net. Also remember to use credentials of newly created sql login in target database (here it is rgSQL).

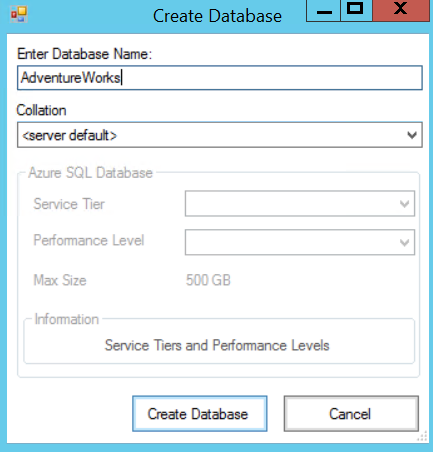


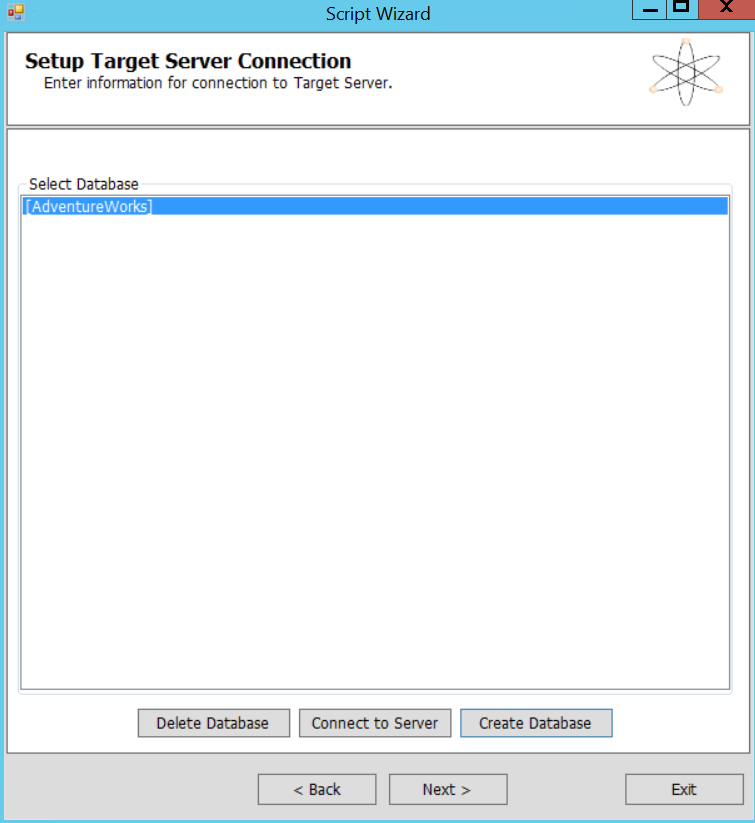


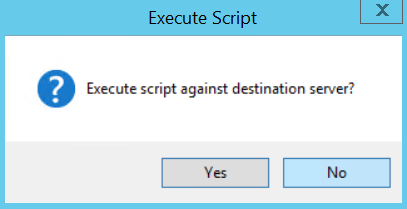
### Start Migration

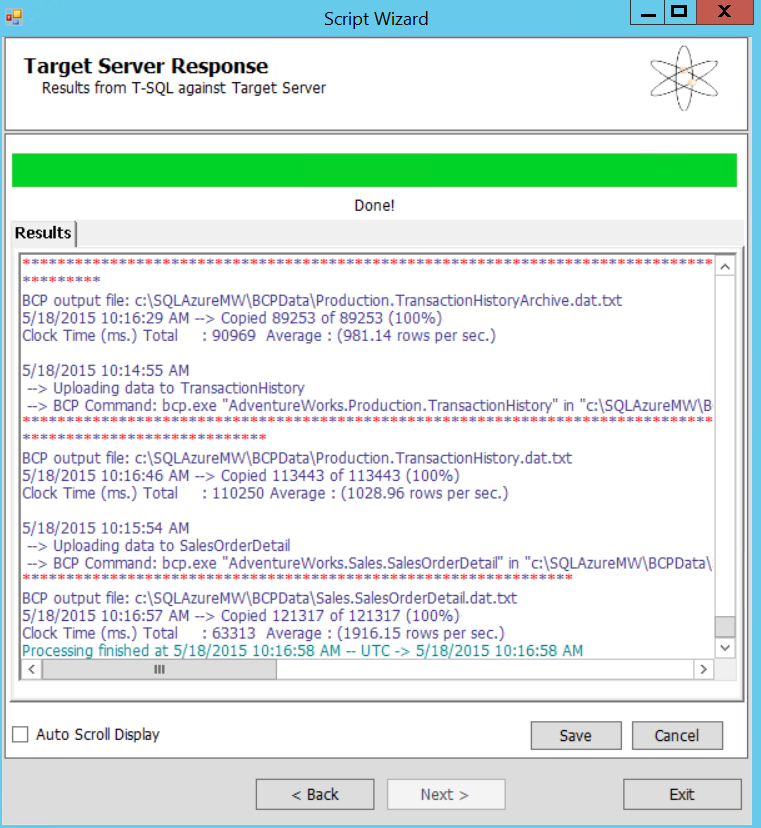
* Once connection with target database has been established, create a new database and start migration as shown below.





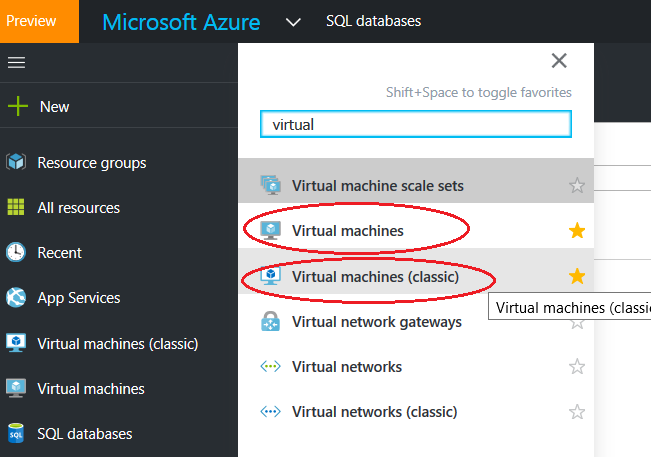






* Congratulations! You’ve successfully migrated database from on premise VM to SQL Server on Azure VM.

NOTE: For creating VM from the ARM portal, most of the settings remains same except which type of VM do you select. There are 2 types of VM you can find in ARM portal, one is VM Classic (same as you get from classic portal as discussed above), second Virtual Machine (Its ARM based)



Lab Completed!