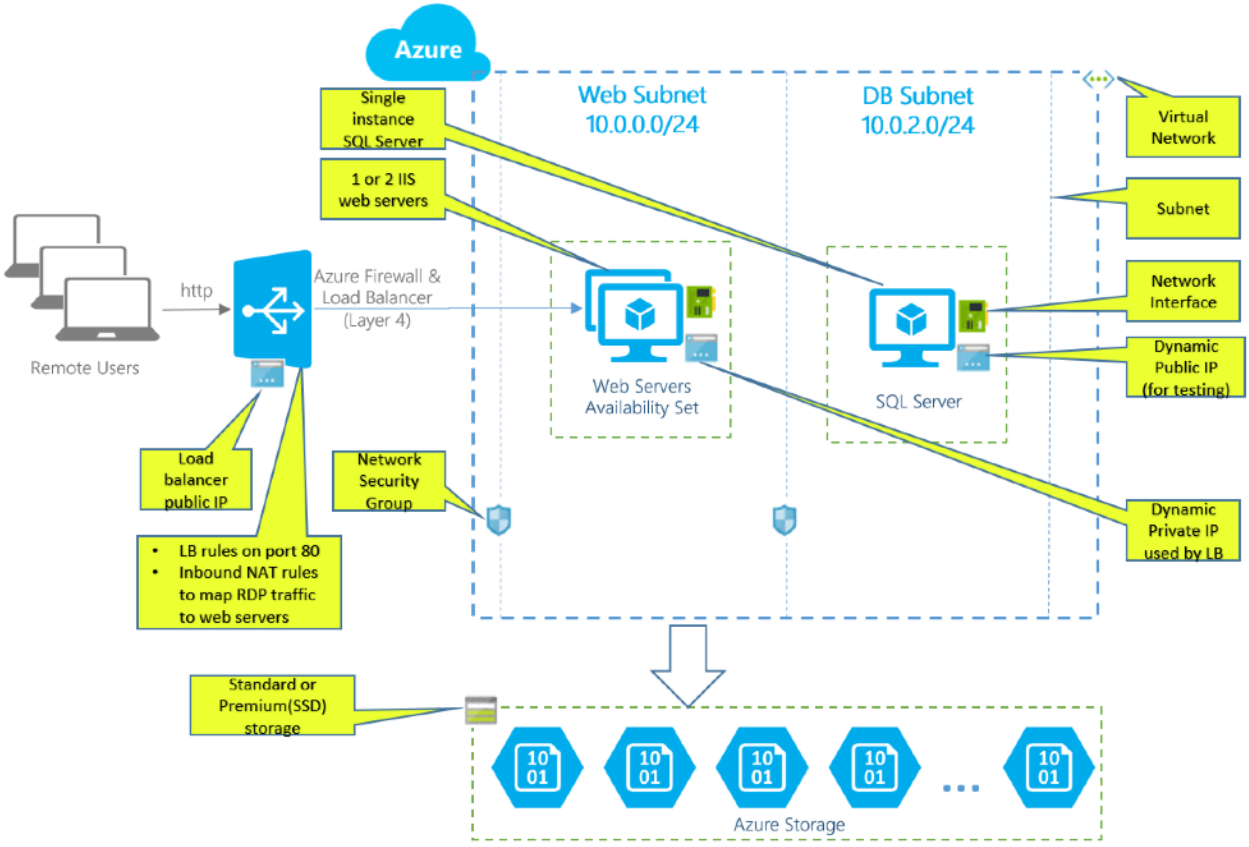
## Azure Resource Manager Template

### Step 1

Architecture diagram for ARM Template



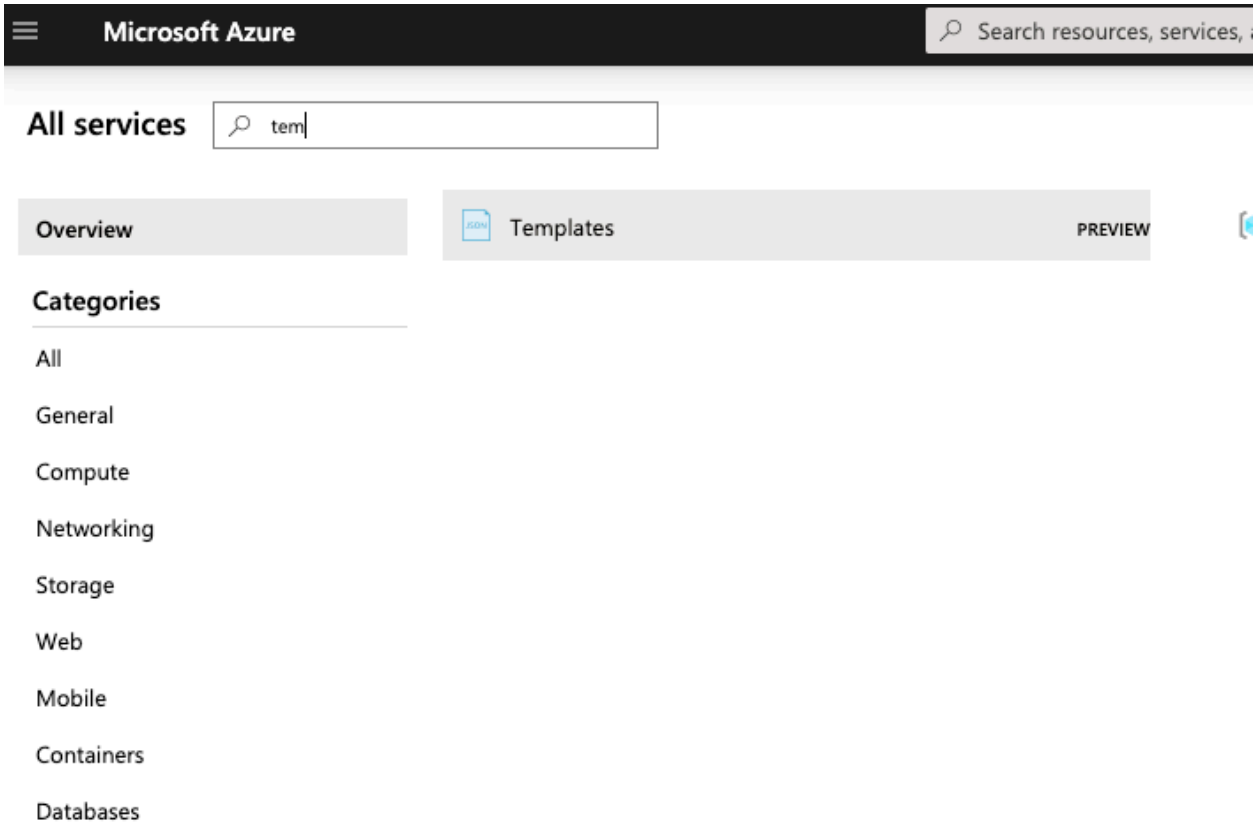

1 Availability Set for IIS servers.

1 Load balancer with NATing rules.

**Step 2**

The below **Deploy to Azure** button embeds an Azure ARM template which creates one or two Windows Server 2012R2 VM(s) with IIS configured using DSC. It also installs one SQL Server 2014 standard edition VM, a VNET with two subnets, NSG, loader balancer, NATing and probing rules.

**Go to All Services and write Templates**



**Step 3**

**General**

**Name:**

**Azure_Template_Username**

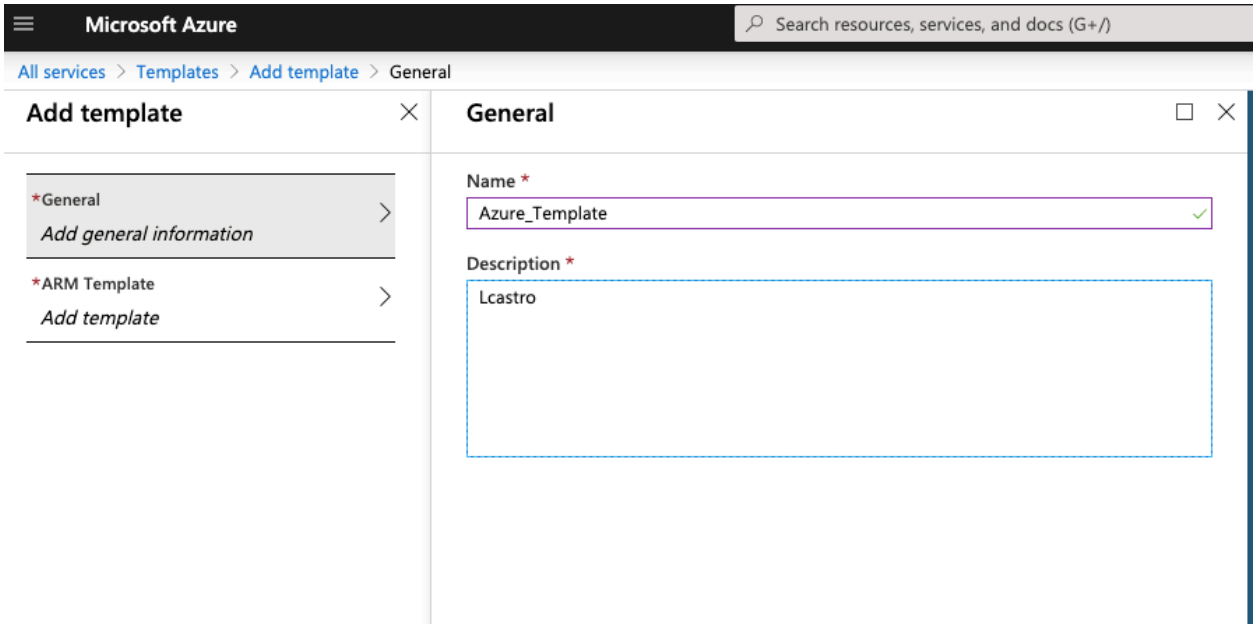
**Eg: Azure_Template_lcastro**

**Description:**

**Username**

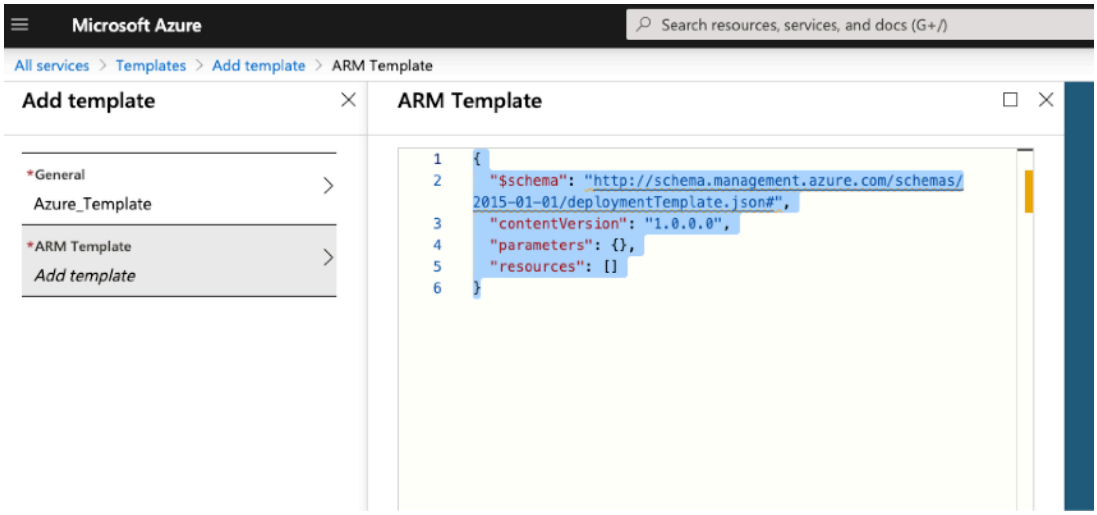
**Eg: Lcastro**

**Click Ok**

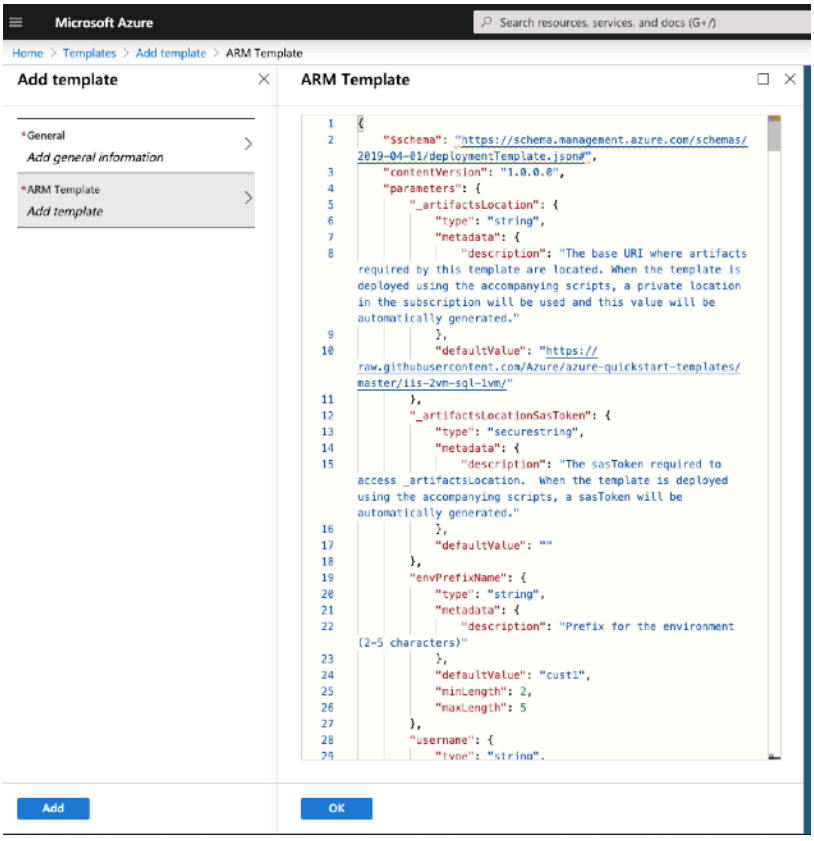


## ARM Template

### Delete everything inside ARM Template



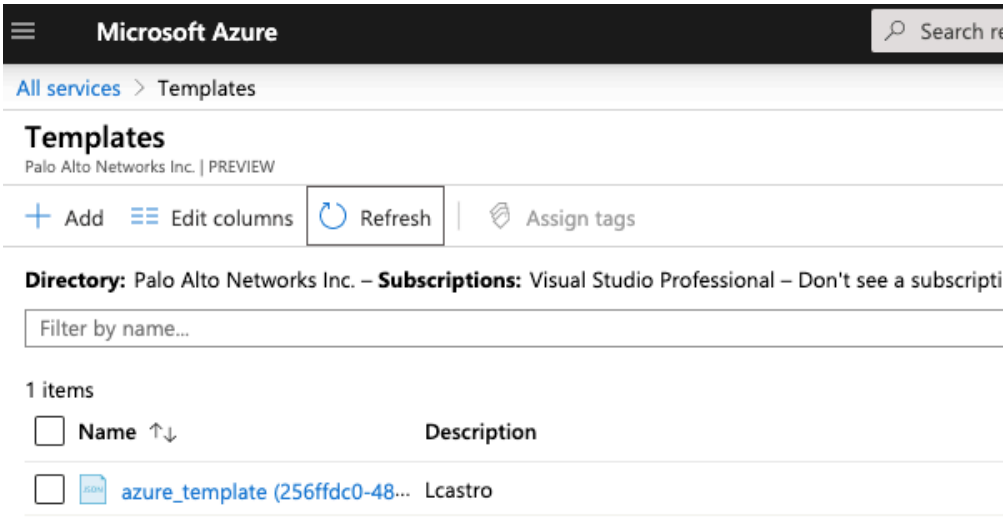

Go to GitHub and search for the **Azure_ARM.json File, copy and paste inside ARM Template Field**



**Click Ok and them Add**

**Agree Terms and Conditions > Purchase**


**Validate Template is loaded and Click on the Name**



**Add the following Parameters**

**Resource Name > Create New: username-ARM Eg: lcastro-ARM**

**Location:** Region designated by the instructions

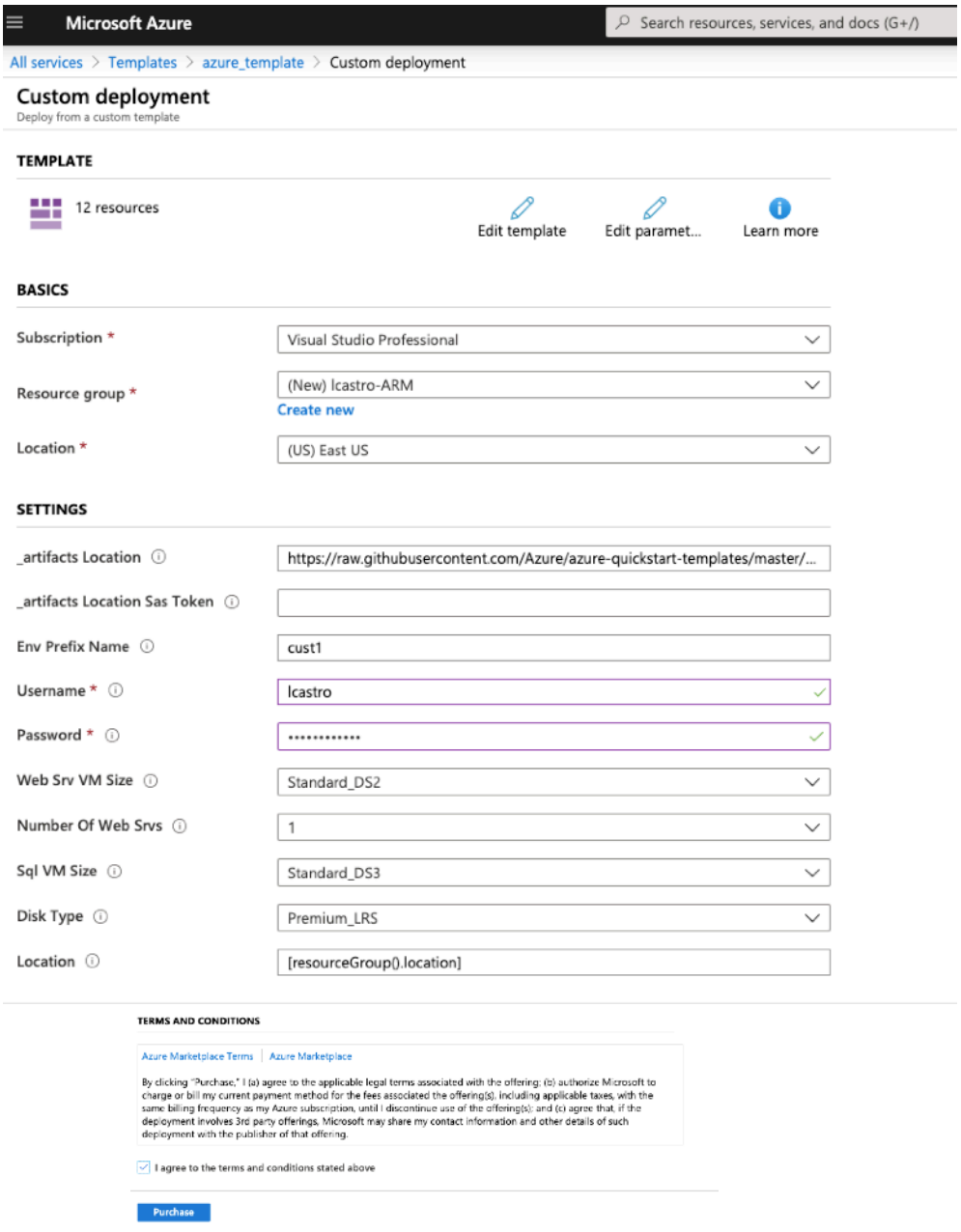
**Env Prefix Name:** First 5 letters of your username - Eg: lcast

**Username:** username Eg: lcastro

**Password:** Defined by each user

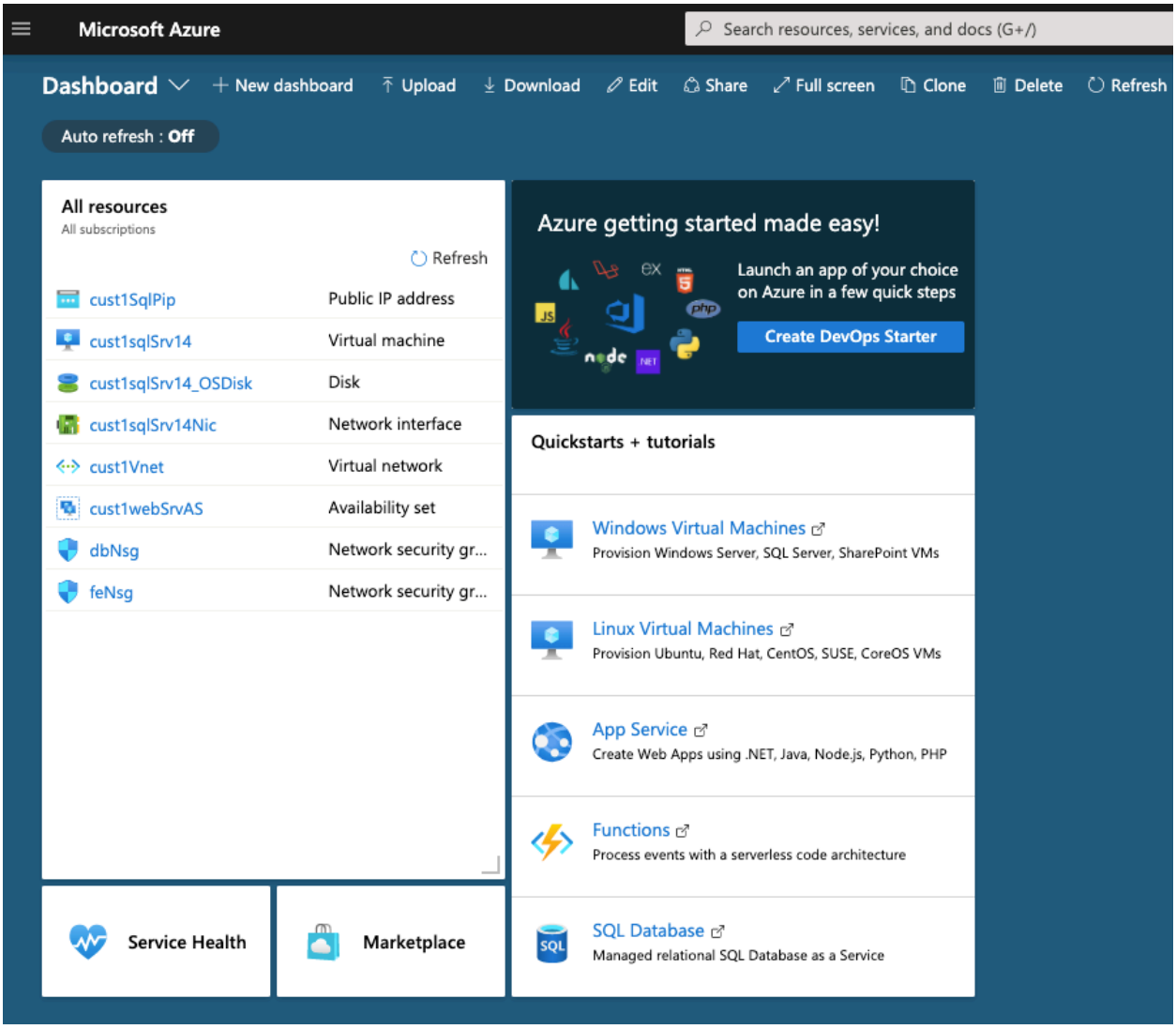
Agree Terms

**Click Purchase - Deployment start creating Assets**



## Step 4

## Go to Dashboard and review the assets created

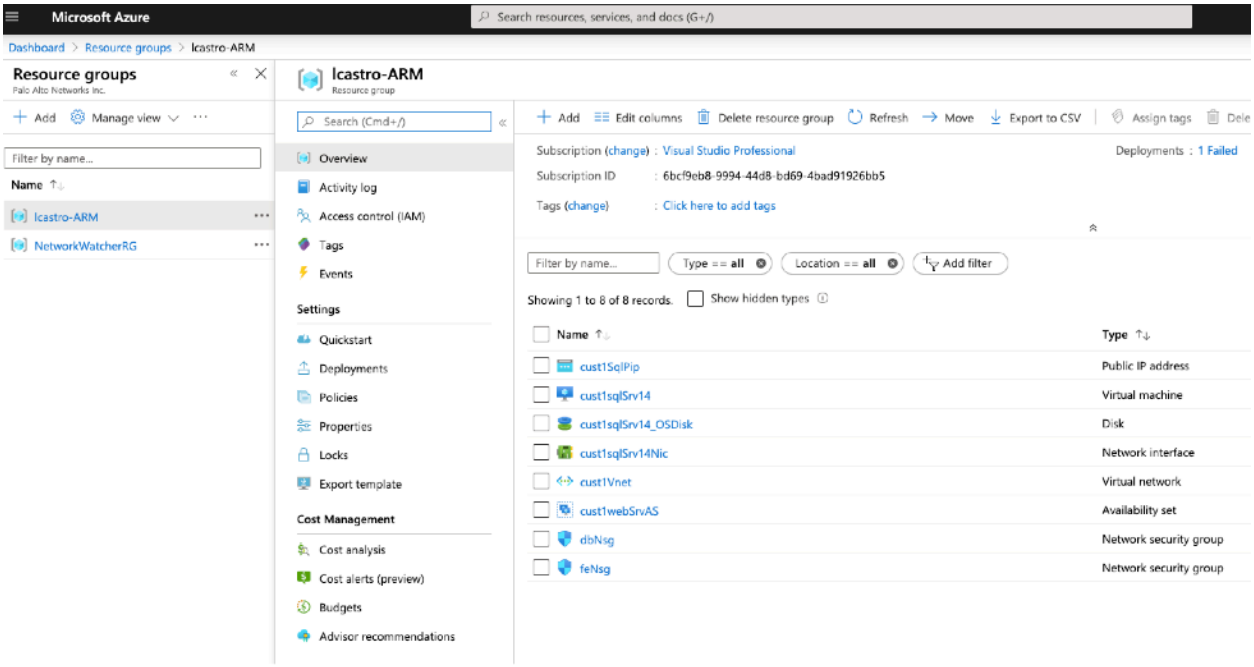


**Step 5**

Go to the Resource Group created under your name:

Once the Template Deployment succeeds, you will have WebServer with IIS Installed and SQL Server 2014 Standard deployed on a Virtual Network with 2 subnets with NSG rules and a Load Balancer with NATing rules.

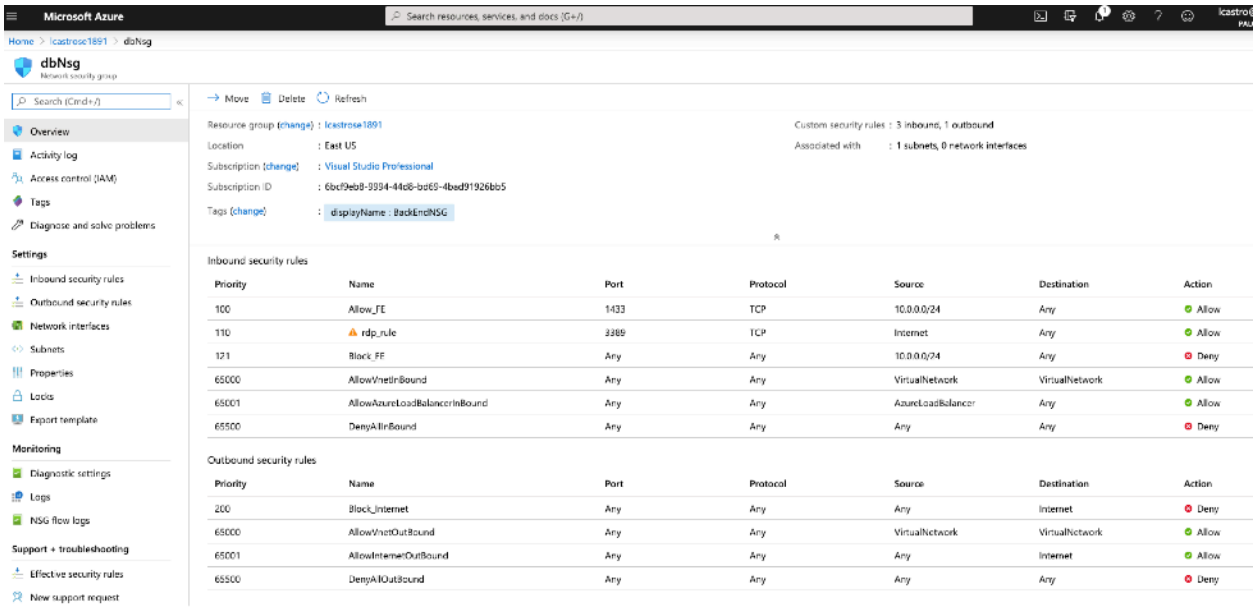
Click on the Resource Group Tile pinned on the dashboard and then click on each resource for more details.



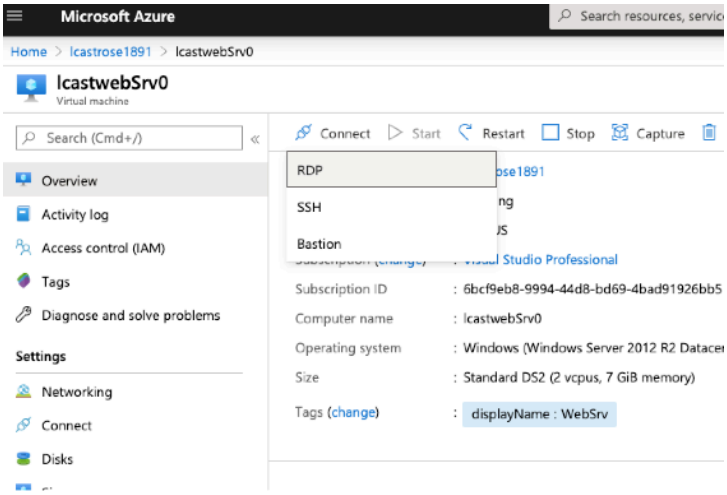
## Step 6

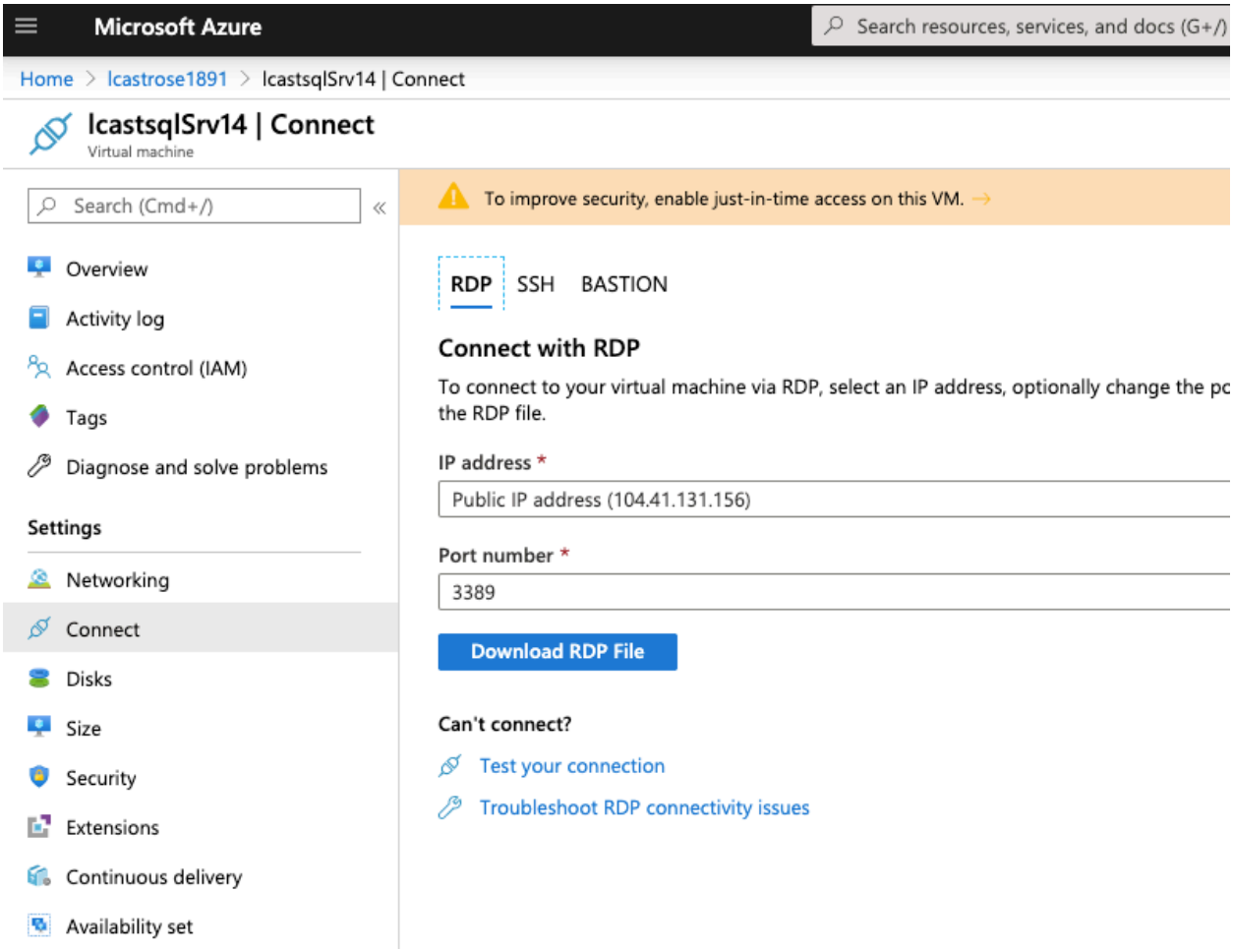
Now, the 2-tier architecture is created with all necessary resources, to check whether all the rules are applied we will deploy an ASP.NET application and a Sample Database(AdventureWorks2012).

Click on dbNsg which is a Network security group and delete the outbound Security rules (Prioirity - 200) - We do this step to download the below AdventureWorks2012 database.




Download RDPs for both Database server and Application server and login with the credentials and download the ASP.NET application content into app server and AdventureWorks2012 database into Database server.




Microsoft Azure
Search resources, services, and docs (G+/)
Home > lcastrose1891 > lcastsqlSrv14 | Connect
lcastsqlSrv14 | Connect
Virtual machine
Search (Cmd+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Networking
Connect
Disks
Size
Security
Extensions
Continuous delivery
Availability set
To improve security, enable just-in-time access on this VM.
RDP
SSH
BASTION
Connect with RDP
To connect to your virtual machine via RDP, select an IP address, optionally change the pc
the RDP file.
IP address *
Public IP address (104.41.131.156)
Port number *
3389
Download RDP File
Can't connect?
Test your connection
Troubleshoot RDP connectivity issues

**Step 7**

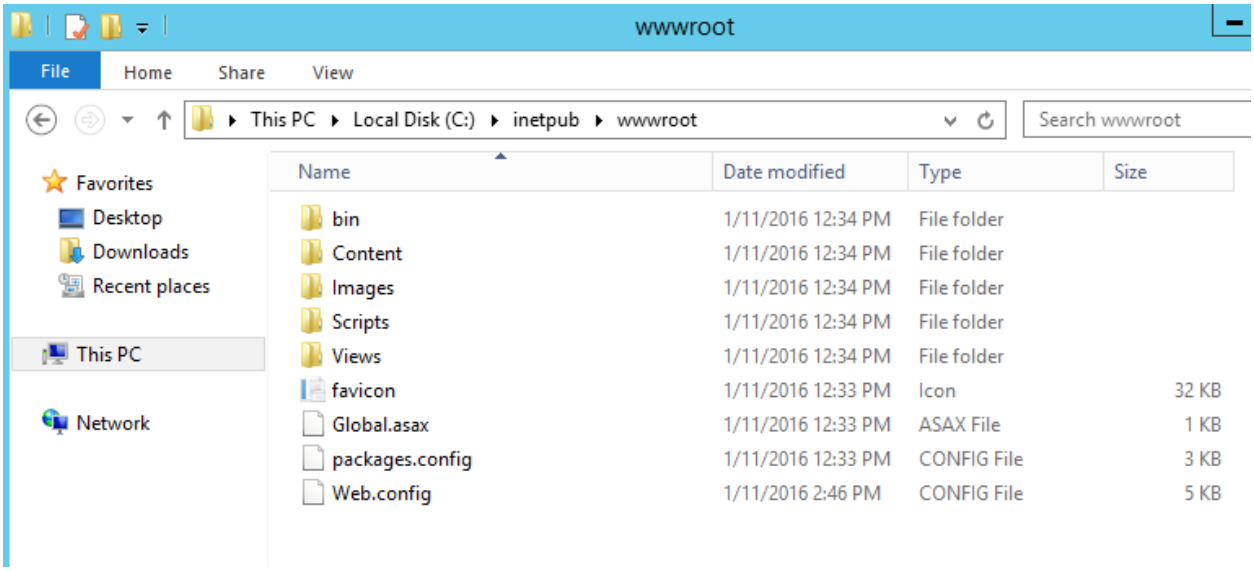
A sample ASP.NET Application content can be downloaded here:

https://github.com/lcastrose/AZURE-Training/blob/master/CloudShop.zip

Sample Database can be downloaded here :

https://dblcastro.s3.amazonaws.com/AdventureWorks2012.bak

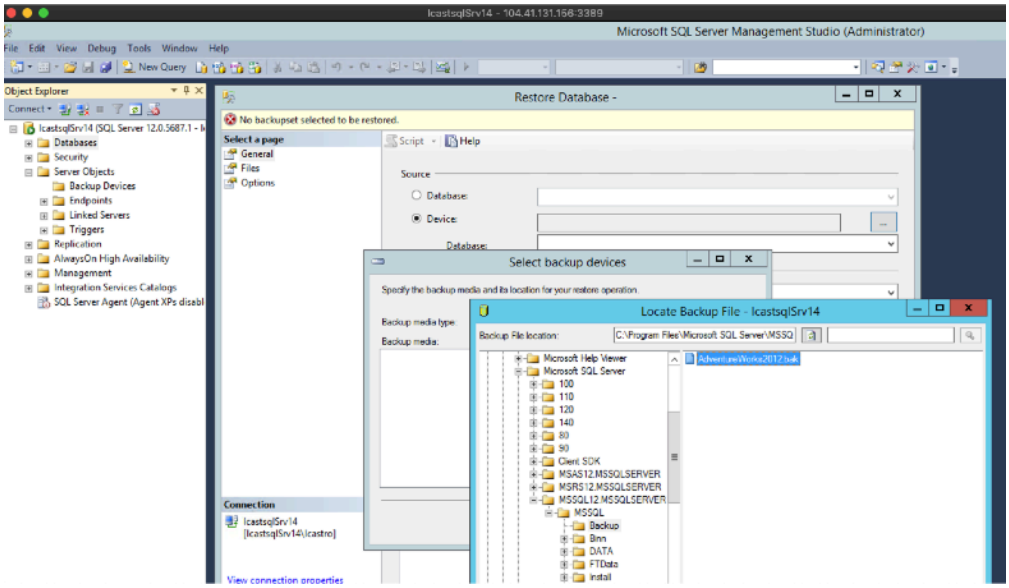
Once you download application content onto app server extract the .zip file and copy the content and past in C:\inetpub\wwwroot.

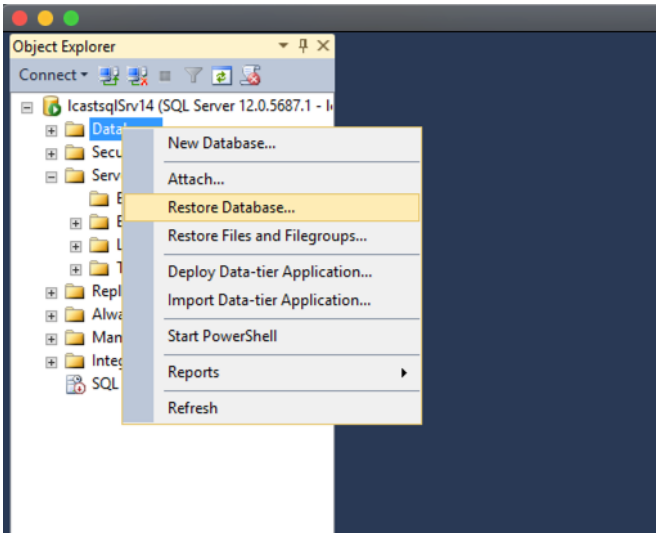


**Step 8**

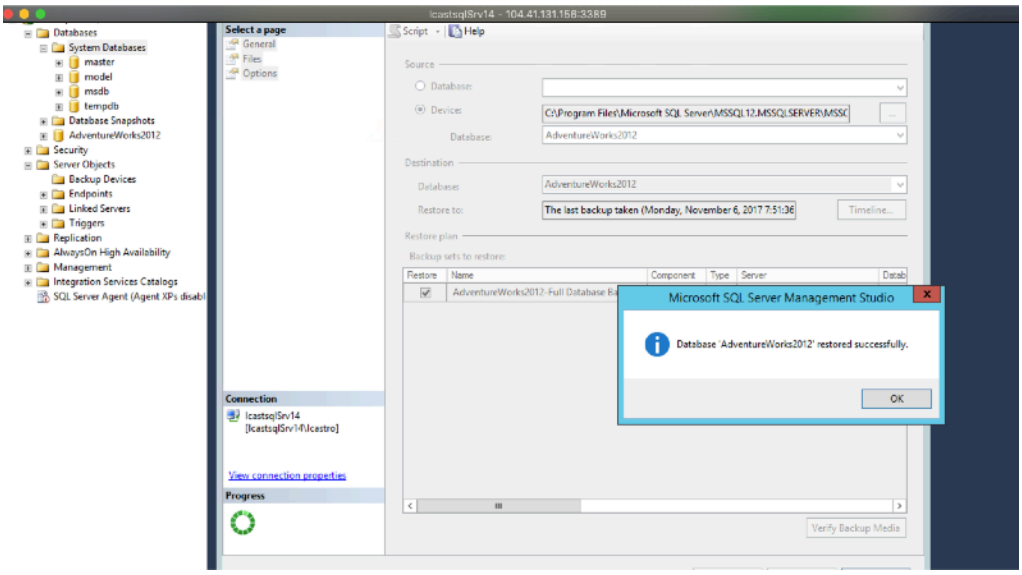
Open Database Server, and Open SQL Server Management Studio 2014 login with Windows Authentication for restoring the AdventureWorks2012 database.

Copy the .bak file to the Backup location "C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Backup" and click OK.



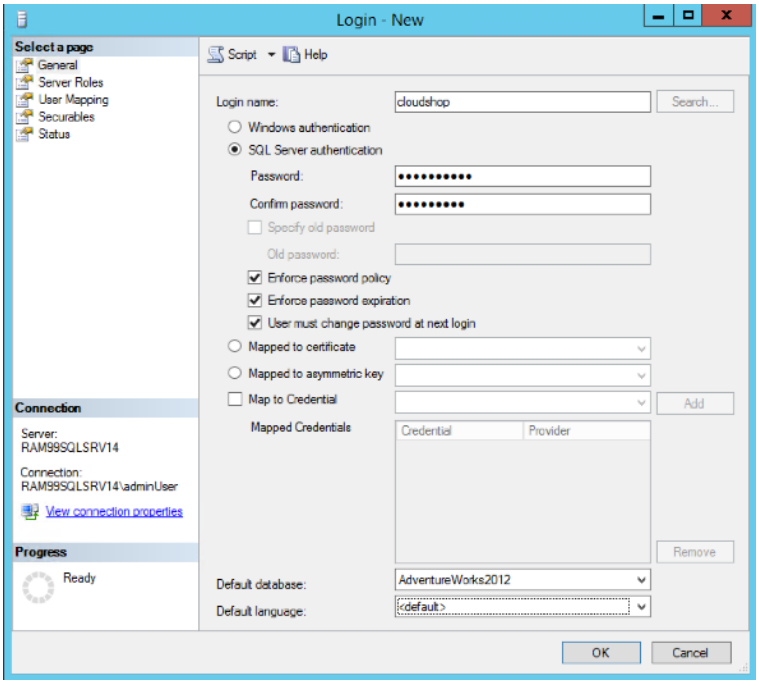



Restoring Backup database file

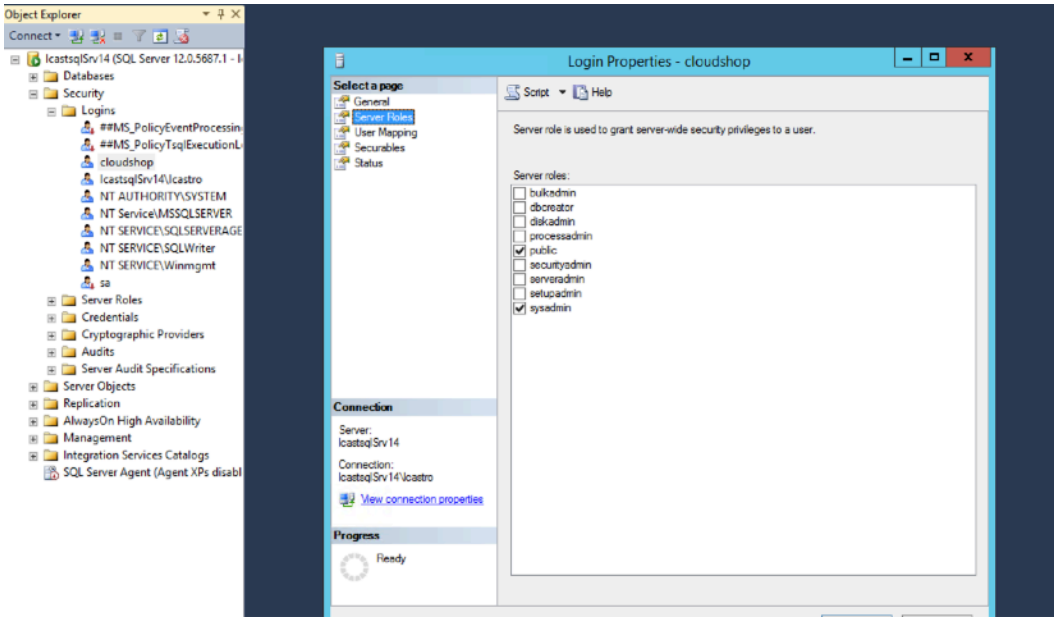



In object explorer go to Security section and Login subsection Right Click and New Login and create a user with SQL Server authentication and in the default database select as AdventureWorks2012

Create a login user

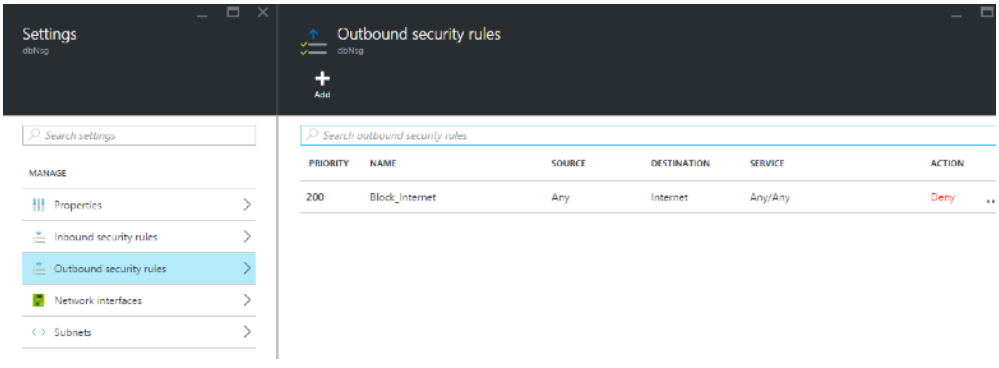


On Left side you have Server Roles -> Select public and sysadmin and check in User Mapping whether public is selected or not and click OK




So till now we have an application content in appserver and database in Database Server. Fianlly we have to setup the NSG Outbound rule which we have deleted earlier to Database Server through portal.

In Azure Portal Click on the resource group which we just created and in the resources click on dbNsg and click on All Settings and Outbound security rules and click on add and fill the details as below:




Adding a Outbound NSG rule

And Finally login into AppServer and open inetpub\wwwroot and Open Web.Config in notepad and replace the ConnectionString with the below code:

<add name="AdventureWorksEntities" connectionString="metadata=res://*/Models.AdventureWorks.csdl|res://*/Models.AdventureWorks.ssdl|res://*/Models.AdventureWorks.msl;provider=System.Data.SqlClient;provider connection string="Data Source=tcp:{SQL Server Private IP},1433;Initial Catalog=AdventureWorks2012;Uid={DB User};Password={Password Created};multipleactiveresultsets=True;App=EntityFramework"" providerName="System.Data.EntityClient"/>

<add name="DefaultConnection" connectionString="Data Source=tcp:{SQL Server Private IP},1433;initial catalog=AdventureWorks2012;Uid={DB User};Password={Password Created};MultipleActiveResultSets=True" providerName="System.Data.SqlClient"/>

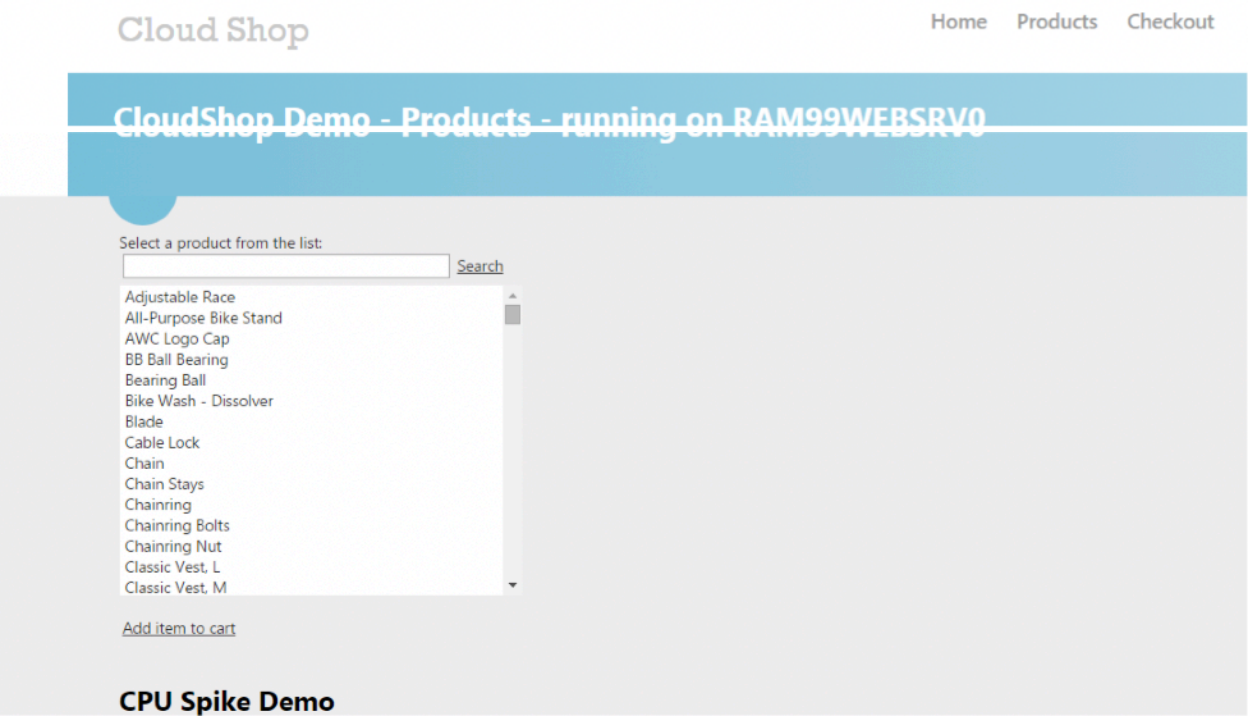
NOTE: Before replacing the connection string, do change the following text with respective values

1) Destination IP address : {Destination-Internal-IP} -> SQL Server Internal IP

2) User ID: {User-created-DbServer} -> User ID for SQL Authentication

3) Password: {password} -> Password for SQL Authentication

Now you can verify by copying the Loadbalancer IP address onto browser and you will see an asp.net application with data populating from the DB Server.



## Step 9

Remove Resource Group using Azure Portal