



Microsoft AZURE Step by Step Guide

By: Nitin pandit

This book also provides
the top 50 Microsoft Azure
Interview Questions and Answers



Microsoft Azure Step by Step Guide

This free book is provided by courtesy of [C# Corner](#) and Mindcracker Network and its authors. Feel free to share this book with your friends and co-workers. Please do not reproduce, republish, edit or copy this book.

Nitin Pandit

Software Engineer,
Microsoft.Net Consultant Noida
C# corner Delhi Chapter Lead
C# corner MVP

Amit Mishra

(Co-Author, Format Editor)

Index	Page No
Chapter 1 - Getting Started With Microsoft Azure	1-17
Chapter 2 - How To Host Your Web Apps On Azure Via FileZilla	18-31
Chapter 3 - Configuring A Custom Domain Name For An Azure Website	32-40
Chapter 4 - Virtual Machine In Microsoft Azure Step by Step - Part One	41-50
Chapter 5 - Creating A Virtual Disk Virtual Machine In Microsoft Azure Step by Step: Part Two	51-66
Chapter 6 - Use of SQL Azure In Visual Studio 2015 Step By Step Guide	67-98
Chapter 7 - Entity Framework With Microsoft Azure SQL	99-113
Chapter 8 - Introducing U-SQL- Make Big Data Processing Easy	114-117
Chapter 9 - Installation Of Azure SDK And Blob Storage In Azure: Part 1	118-139
Chapter 10 - Upload Image To Azure Blob Storage In ASP.NET: Part 2	140-163
Chapter 11 - Demo Project Azure Blob Storage With ASP.NET MVC 5.0: Part 3	164-204
Chapter 12 - Getting Started With Azure DocumentDB: Day 1	205-224
Chapter 13 - Querying In The Azure DocumentDB: Day Two	225-242
Chapter 14 - Manage Azure Mobile Services: Part One	243-259
Chapter 15 - How To Create A Mobile App In Microsoft Azure	260-277
Chapter 16 - Manage Database Settings Of Azure Mobile Services	278-281
Chapter 17 - Get Twitter Account Connected Into Microsoft Azure Blob Storage Live	282-333
Chapter 18 - Cloud Services Scheduler In Microsoft Azure	334-339
Chapter 19 - Azure Media Services Overview And Playing Media	340-352
Chapter 20 - How To Use API Apps In Azure	353-362
Chapter 21 - Manage Data In Microsoft Azure Machine Learning	363-389
Chapter 22 - Working With Azure RemoteApp	390-402
Chapter 23 - Microsoft Azure Service Bus	403-417

About Author

He is a Microsoft Professional having Master's degree in Computer Science. With over 5 years of experience his rich skill set includes designing, integrating, implementing, ORM with LINQ, WCF, MVC & managing IT/Web applications. He is an expert in C#.NET ADO.NET, LINQ to SQL, LINQ to EF, ASP.NET 2.0/3.x/4.0, WCF, MVC 5.0 (Razor), WPF and Silverlight, including client side technologies jQuery and AngularJS.

He has trained more than one lakh students and professionals as a speaker for workshops and AppFests conducted in more than 25 universities of North India.



Nitin Pandit

Software Engineer,
Microsoft.Net Consultant Noida
C# corner Delhi Chapter Lead
C# corner MVP

Getting Started With Microsoft Azure

I am going to start a new article for users who want to learn Microsoft Azure from the scratch. Let's start with a step by step approach.



Firstly, go to <http://Azure.microsoft.com/> and create a free Microsoft Azure Account.



The screenshot shows the Microsoft Azure homepage. At the top, there is a navigation bar with links for Apps, Bookmarks, Ajax Sample, ASP.NET Code Snippets, MVC Tutorial For Beginner, Articles Of The Day, C# Corner Annual, SALES 91-80-40103000, and a dropdown menu. Below the navigation bar, the title "Microsoft Azure" is displayed. A main banner features the text "Join us for AzureCon" and "Be the first to see what's next at this free, virtual event". A green button labeled "Learn more >" is located below the banner. At the bottom of the page, there is a call-to-action button labeled "Try for free >".

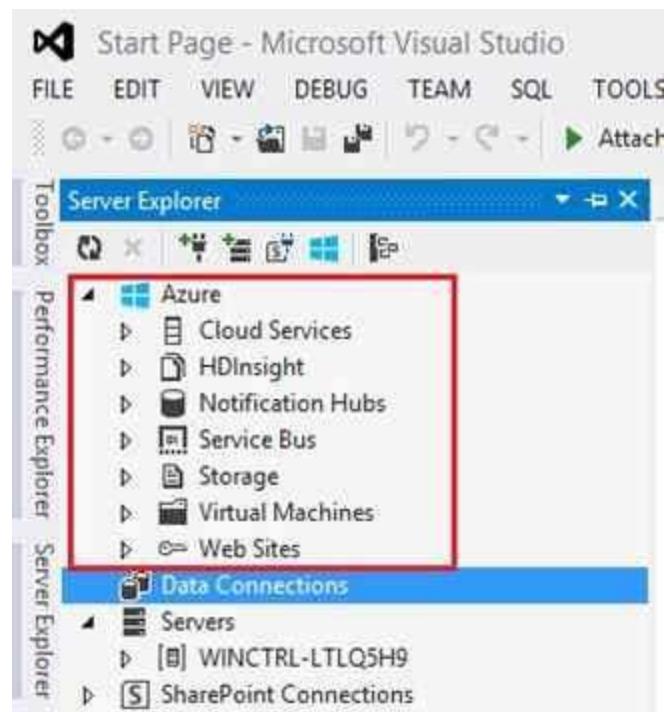
When you *log in* Microsoft Azure with any Microsoft Account like Outlook, Live or other, you will get an Account. After that go to <https://manage.windowsAzure.com/> and login, then click on portal link.

You can see the image in the following portal.

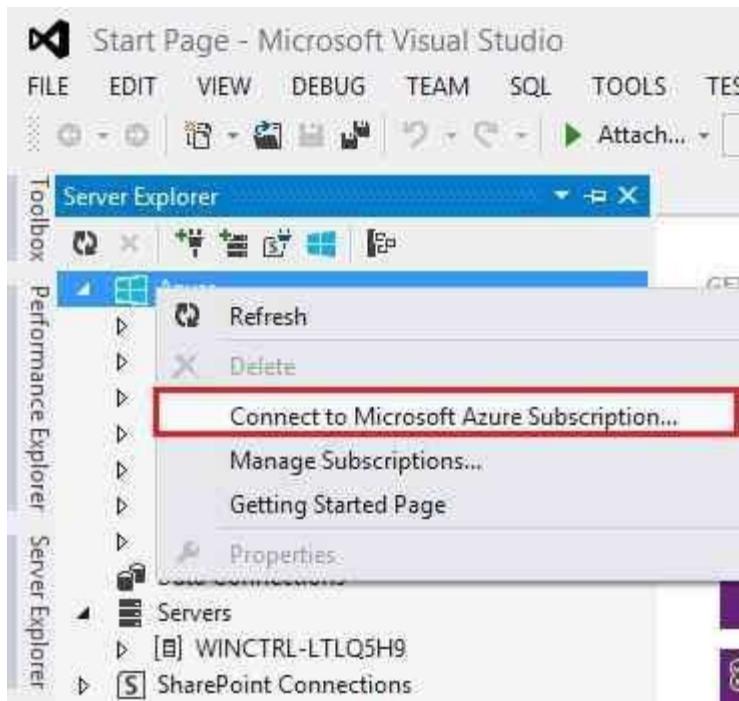


Now you can manage your Azure portal from web account or directly from your Visual Studio using Azure SDK. If you are using VS2012, download Azure SDK first.

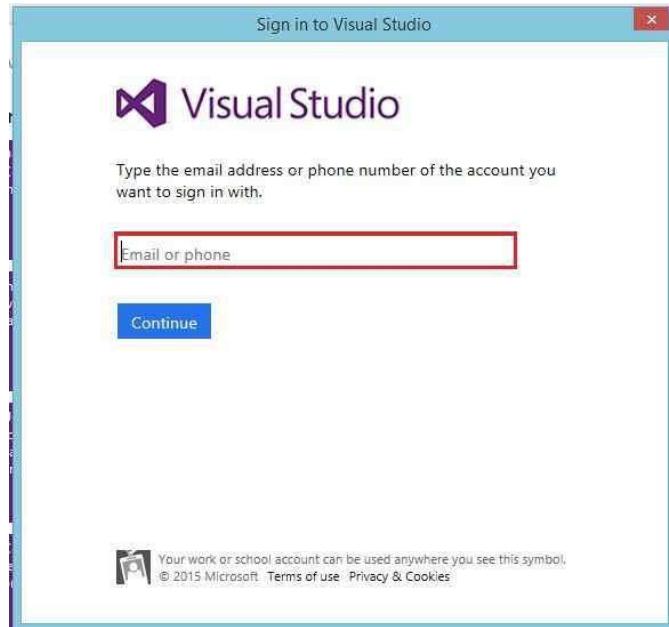
And after installing Azure SDK, your Visual Studio Server Explorer will be visible.



Right Click on SDK and click “**Connect to Microsoft Azure Subscription**”.



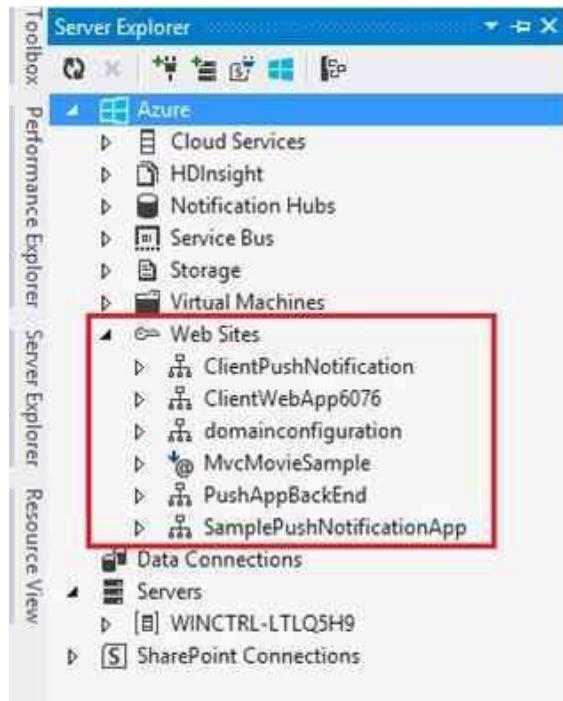
It will open a window to log in Microsoft Azure Account.



Enter your Microsoft Account Id on which you have Azure Subscription, then password details and press enter to login.



Now go to Azure SDK in Visual Studio Server Explorer and open *Web Sites*. Now you can see all the web sites created in your Azure Account.



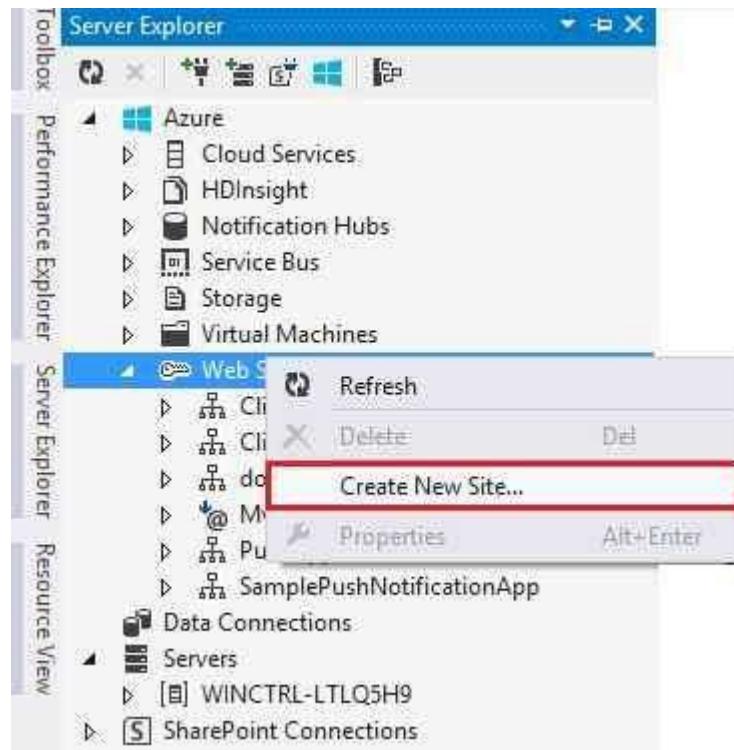
If you need to add or create a new website in your Azure Account, we have the following two ways:

1. By Microsoft Azure SDK in Visual Studio.
2. By Microsoft Azure Web Portal.

So let's see both one by one and step by step.

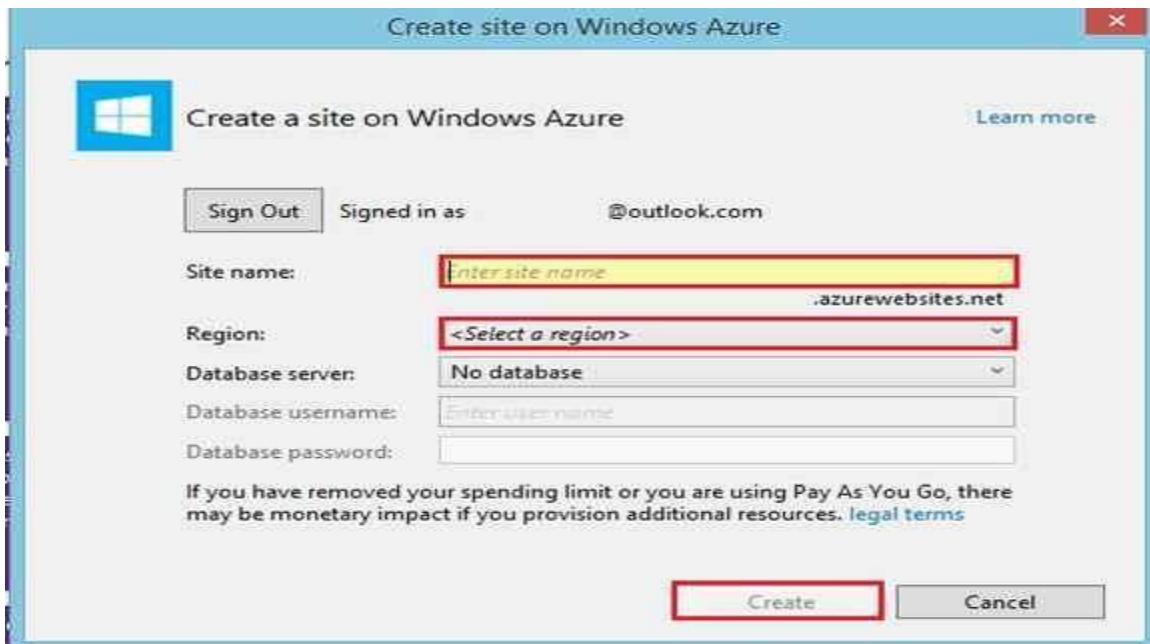
By Microsoft Azure SDK in Visual Studio:

Step - Go to *Azure SDK* in *Server Explorer* and right click on *Web Sites* and find the link “Create New Site”.



Now you'll see a window where you have to enter the following details:

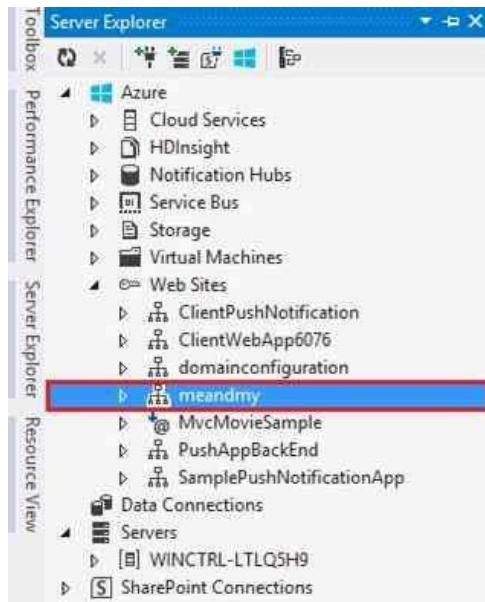
1. Site Name.
2. Select Region.
3. Database Server (if you have).



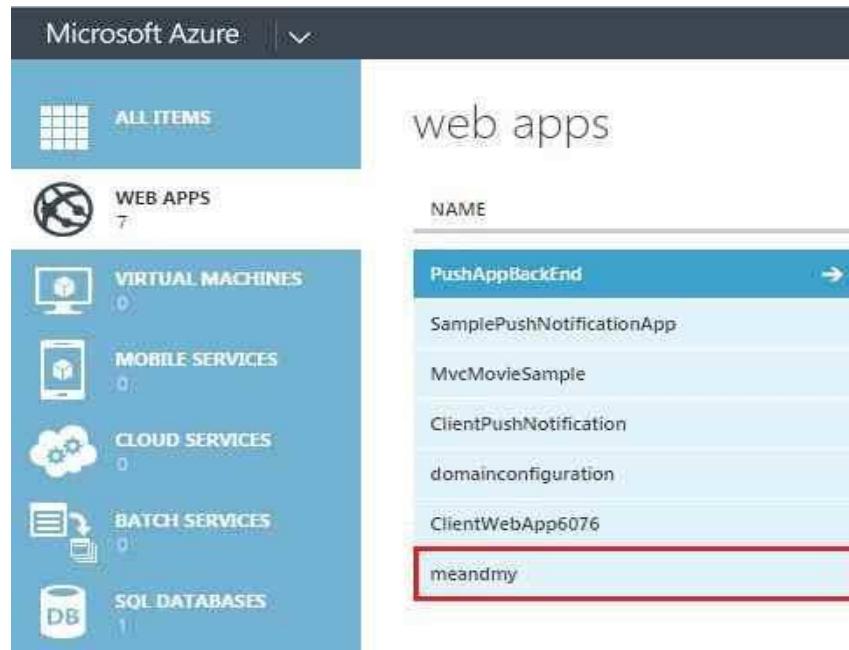
Let say I have entered my new site name “**meandmy**”, if the name is available it will allow me to create web site. Select a region like I selected East Asia. At last select database if you have any database on Azure for your web site.



When you click on create button, you can see your web site to SDK or Azure web portal.



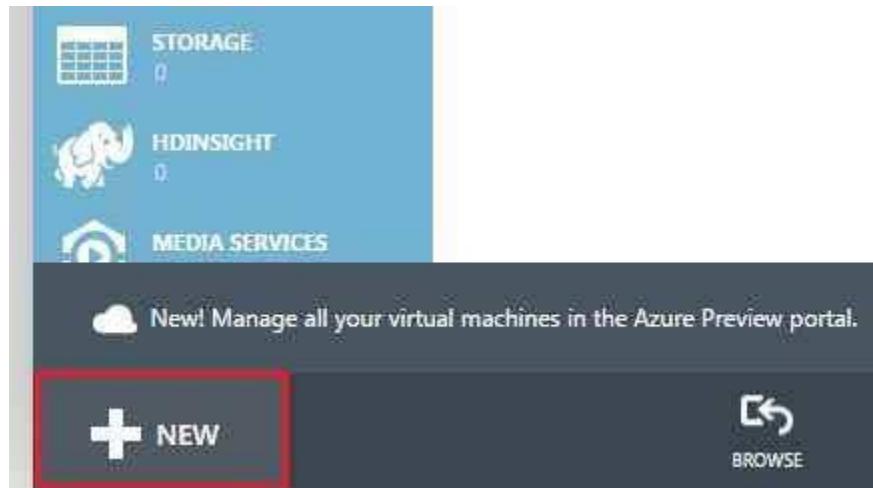
Now go to your Azure portal and refresh. After refreshing it, you can see the same site in your web portal created by SDK.



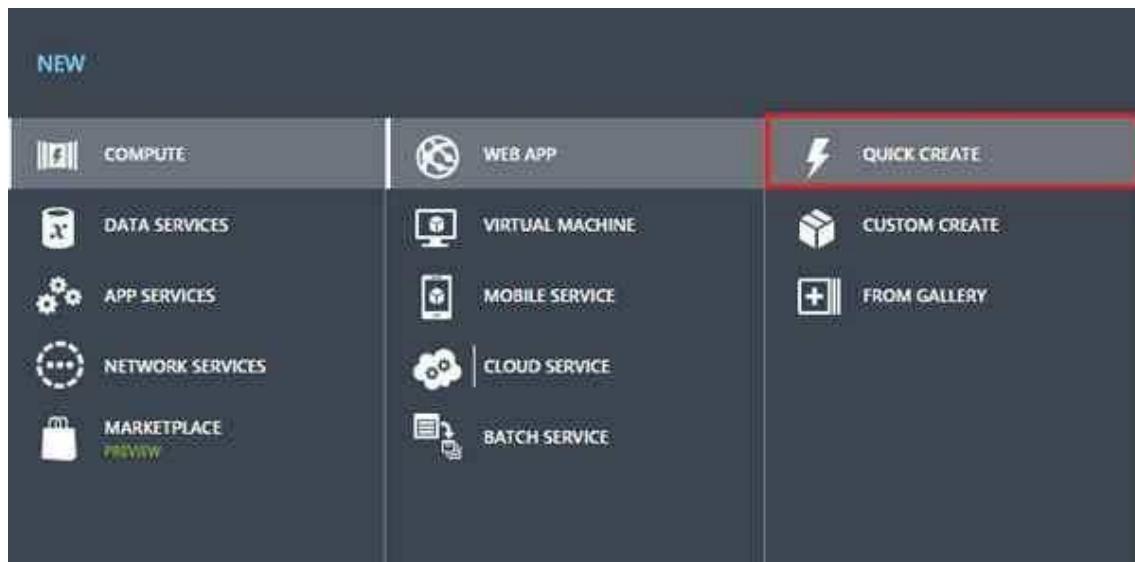
NAME
PushAppBackEnd
SamplePushNotificationApp
MvcMovieSample
ClientPushNotification
domainconfiguration
ClientWebApp6076
meandmy

By Microsoft Azure Web Portal

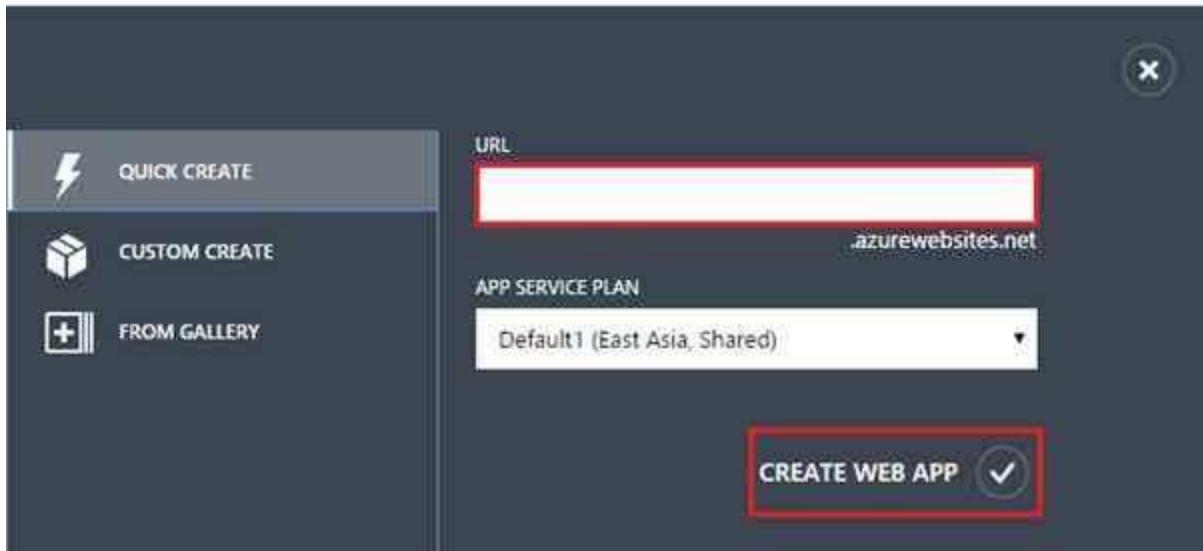
Now I am going to create web site from Web Portal, so let's go to web portal and see the NEW button on the left side bottom and click on it.



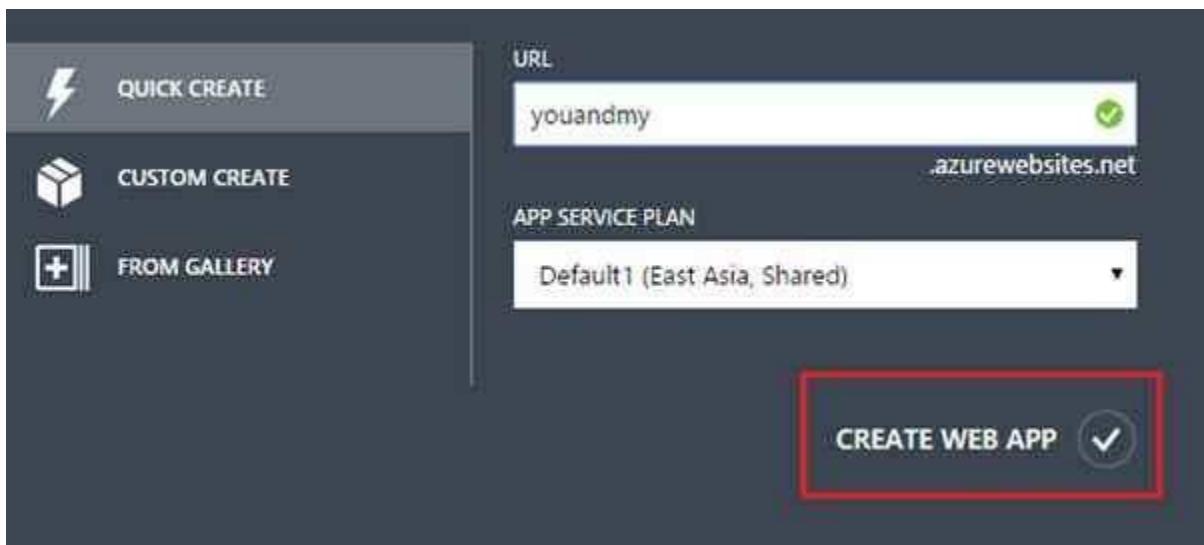
Now click *COMPUTE*, then *WEB APP* and *QUICK CREATE*:



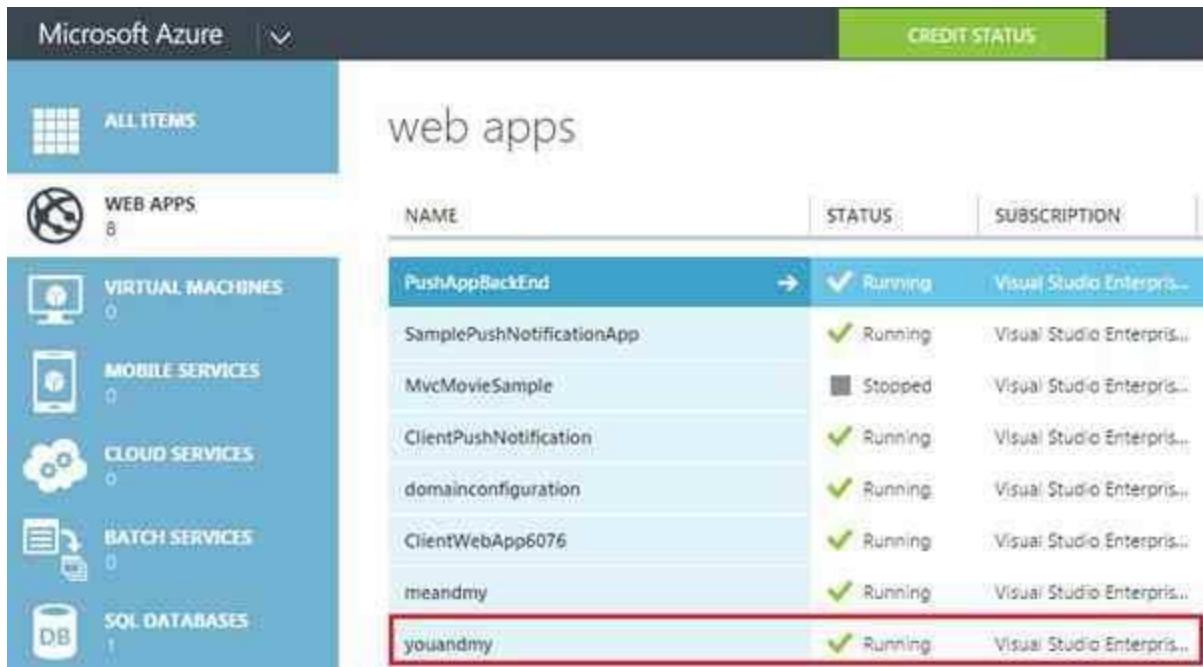
Now enter a URL for Website on which you need to host your web site and select *APP SERVICE PLAN* and lastly click *CREATE WEB APP*:



After that enter all the details,



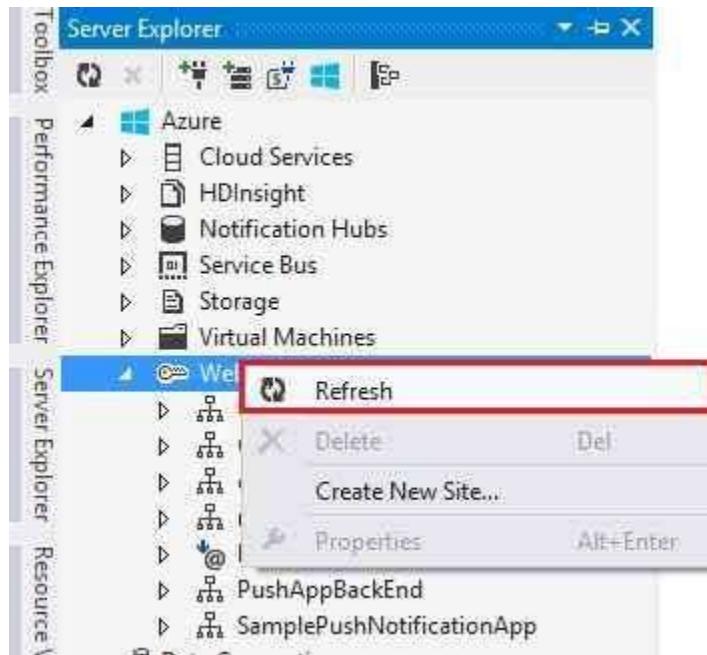
Now web site is ready to use.



The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with icons for All Items, Web Apps (selected), Virtual Machines, Mobile Services, Cloud Services, Batch Services, and SQL Databases. The main area is titled "web apps" and displays a table of running web applications:

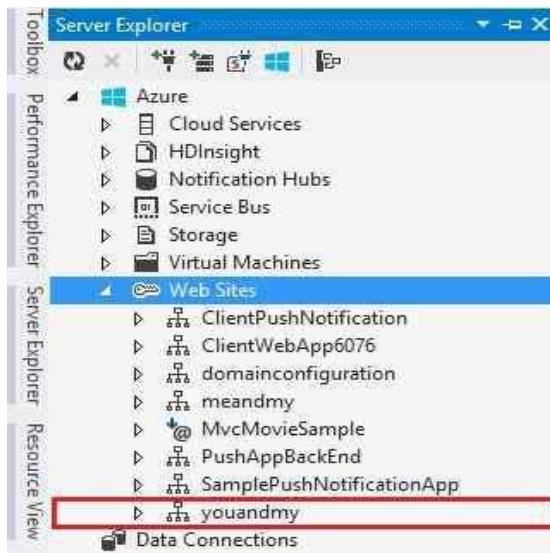
NAME	STATUS	SUBSCRIPTION
PushAppBackEnd	Running	Visual Studio Enterpr...
SamplePushNotificationApp	Running	Visual Studio Enterpr...
MvcMovieSample	Stopped	Visual Studio Enterpr...
ClientPushNotification	Running	Visual Studio Enterpr...
domainconfiguration	Running	Visual Studio Enterpr...
ClientWebApp6076	Running	Visual Studio Enterpr...
itmeandmy	Running	Visual Studio Enterpr...
youandmy	Running	Visual Studio Enterpr...

If you want to use this in SDK, open SDK and right click on *Refresh*.

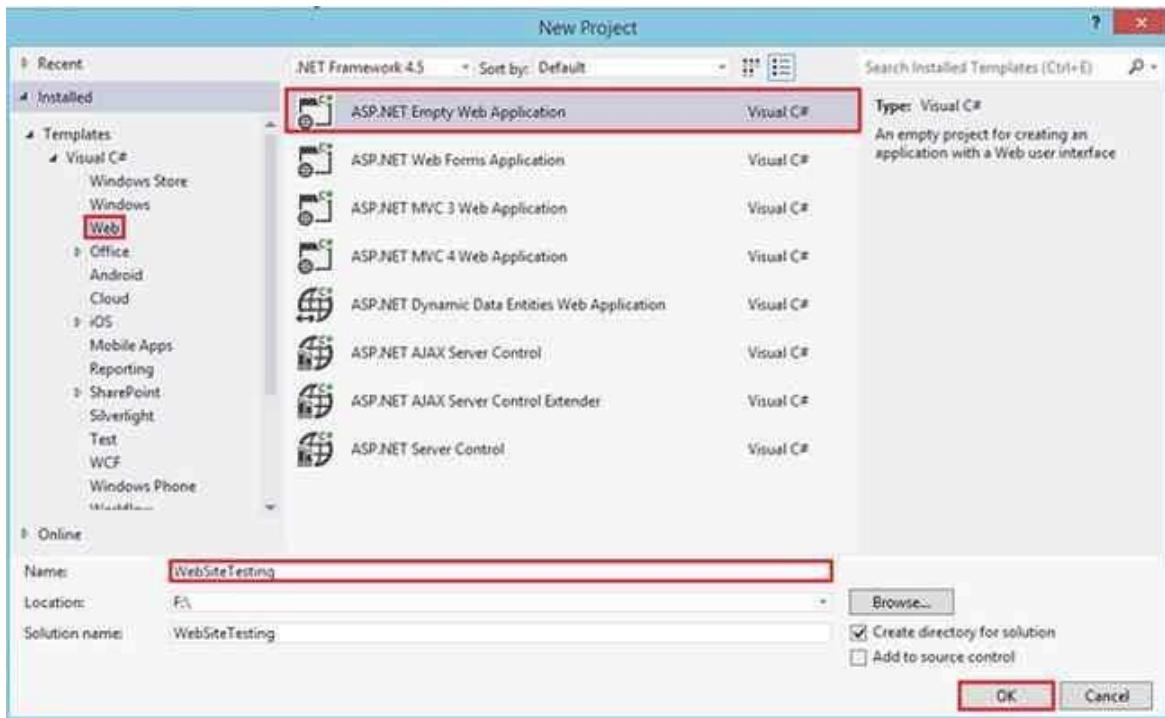


The screenshot shows the Server Explorer window in Visual Studio. The left pane lists Azure services: Cloud Services, HDInsight, Notification Hubs, Service Bus, Storage, and Virtual Machines. The right pane shows a list of web applications: PushAppBackEnd, SamplePushNotificationApp, and a selected item, "youandmy". A context menu is open over the "youandmy" item, with the "Refresh" option highlighted.

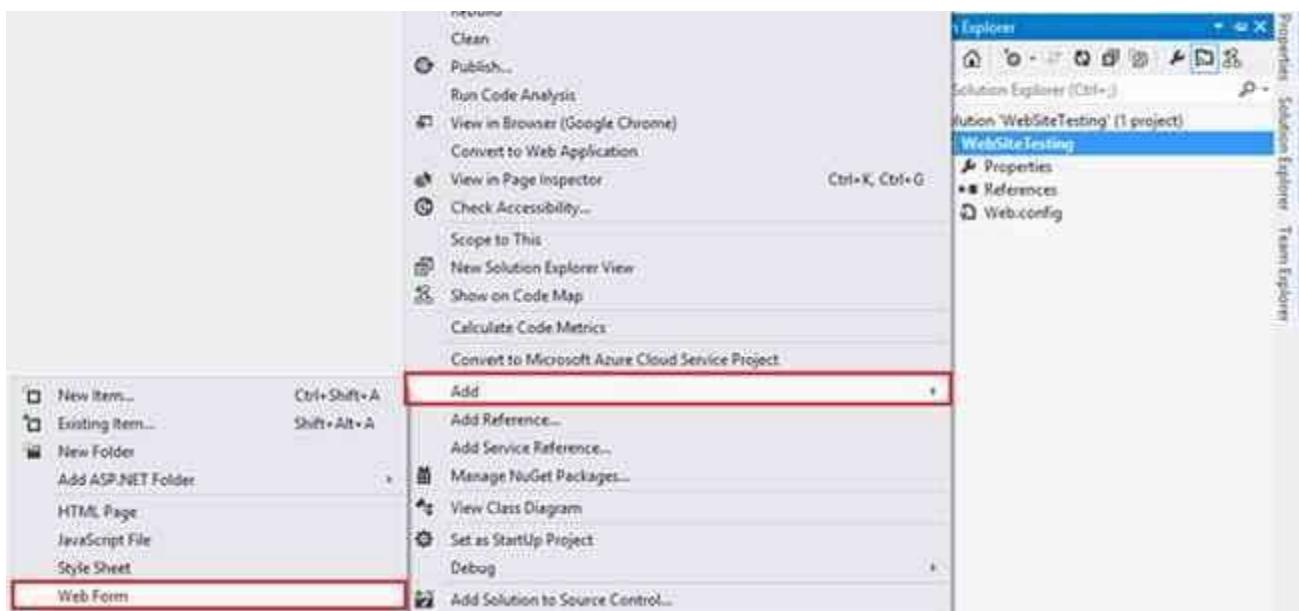
Now you can see the websites on SDK.



Create Web Project in Visual Studio to host on Web site that is already created on Azure Account. Open Visual Studio, *File*, and then *New Project* and select *Web*. After that click ASP.NET Empty Web Application and press *OK*.



Open Solution Explorer now and right click on Project, then add any Default page to View the Test Output.



Write any name for view page.



I just wrote a welcome message on page for testing.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h1>Welcome To My Web Site :)</h1>
        </div>
    </form>
</body>
</html>
```

If I run this web application on local system, the output will be the following:



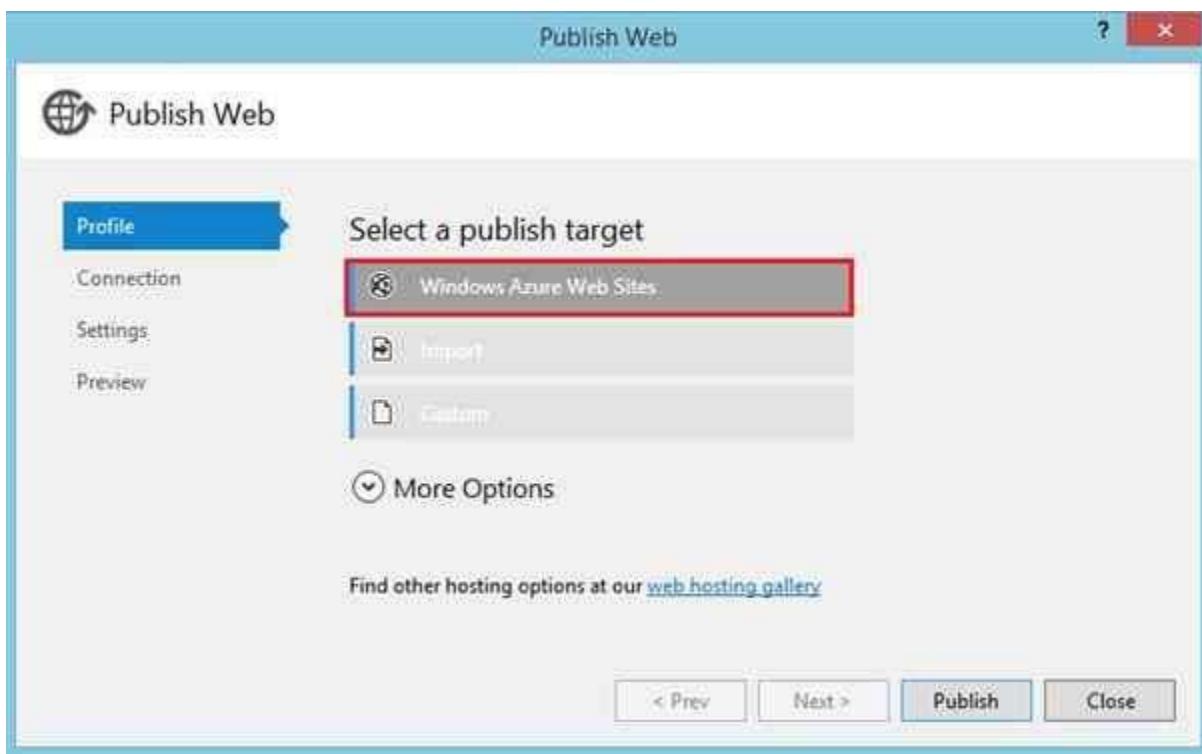
Welcome To My Web Site :)

Now I will upload this to web site in Azure.

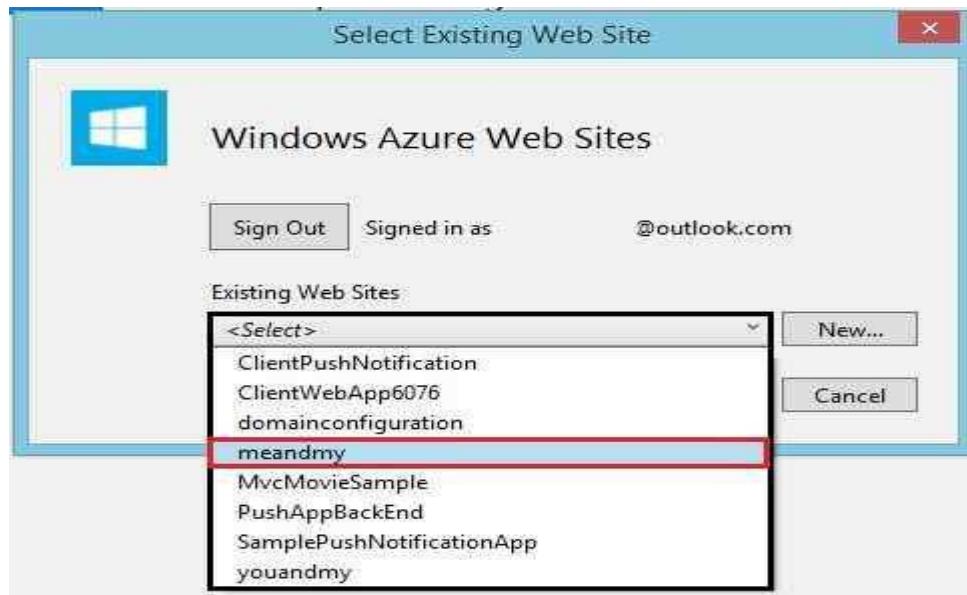
Firstly, right click on the project and find the link *Publish...*



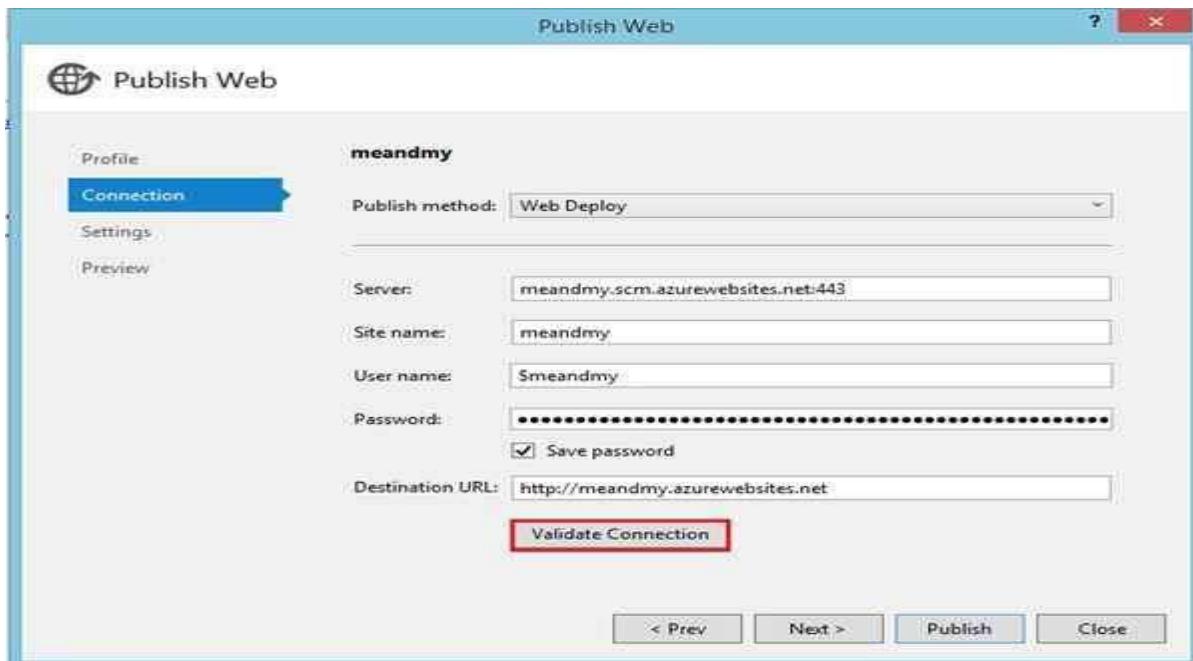
Select the publish target select type and if you want to publish on Microsoft Azure, select the first link in the following window and click *Publish* button.



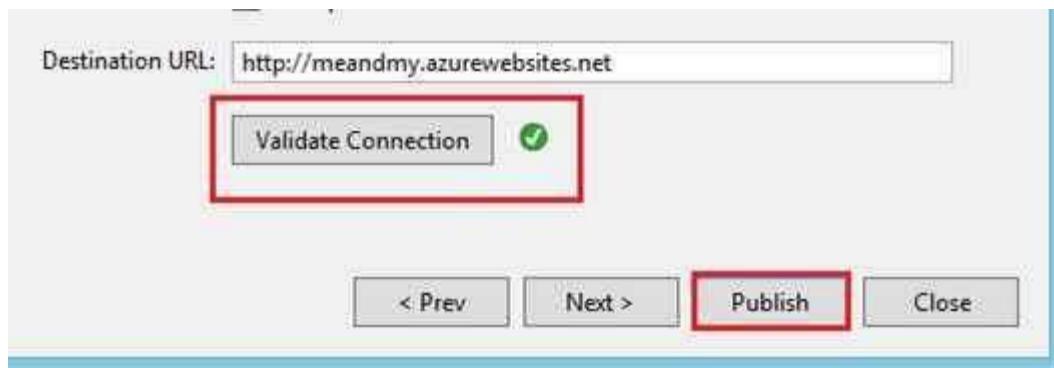
Now select your website name from the dropdown where you need to upload.



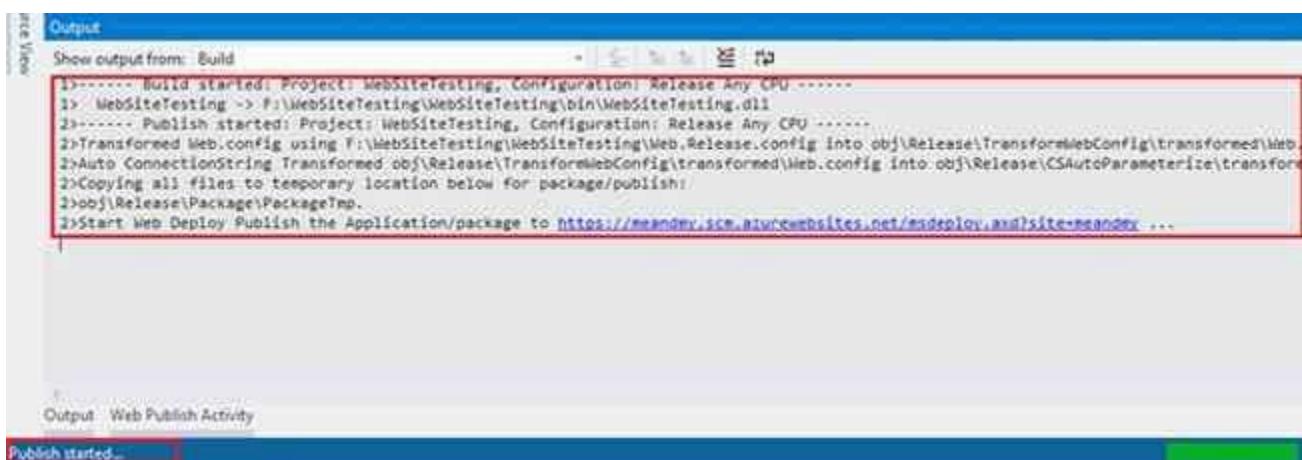
After selecting the website, it will download all the publish profiles from Azure and will fill all the details.



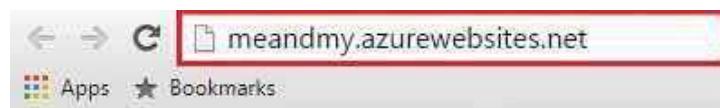
Click on *Validate Connection* button to check all the settings are valid or not. If all are valid, click *Publish* Button.



Now see the upload progress in output window.



After *Publish* succeeds it will directly open the URL in the Web Browser. Now you can check this link anywhere or any device like mobile.



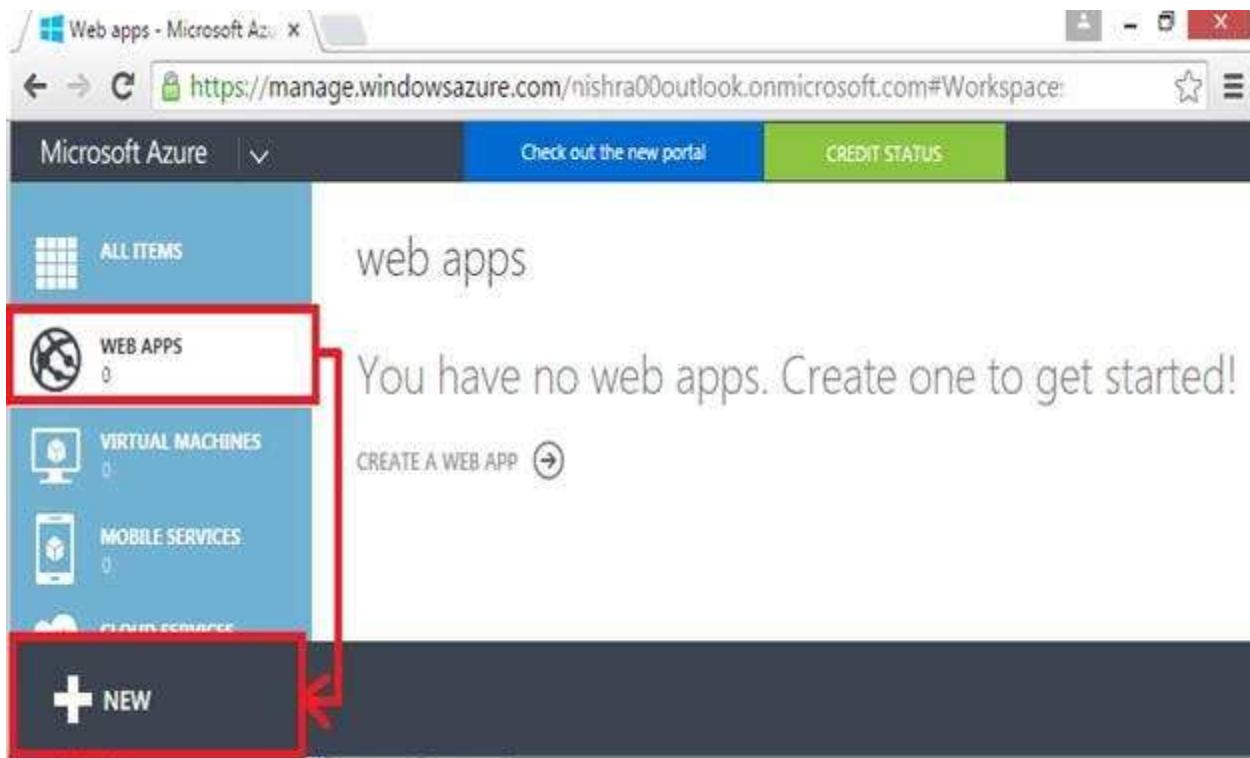
Welcome To My Web Site :)

How to Host Your Web Apps on Azure via FileZilla

In this chapter, I'm going to upload a website to the Microsoft Azure via FileZilla (FTP). FileZilla is a cross platform FTP application which is totally free software. We'll create a Web App then download and publish profile of that Web App.

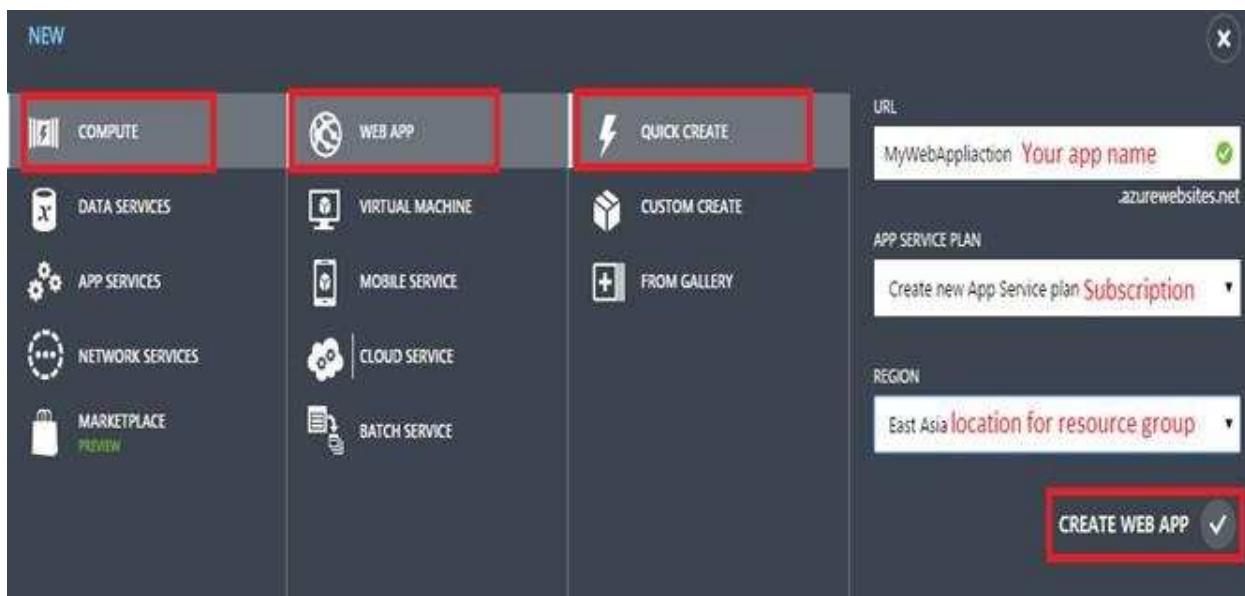
For creating a new Web App you must login into Azure.

Go to *WEB APPS*, and select “*NEW*” button at the bottom of the window.



Go to *COMPUTE, WEB APP, And QUICK CREATE*.

- Enter the name of your Web App.
- Enter your subscription.
- Select region/location of your Web App.
- Click on to “*CREATE WEB APP*”.



Wait for few seconds to complete processing, your WebApp is created and in running mode.

NAME	STATUS	SUBSCRIPTION	LOCATION	PRICING	URL
MyWebApplication	Running	Azure Pass	East Asia	Free	mywebapplication.azurewebsites.net

After creating our Web App on Azure, you can start with Visual Studio, write some text in to HTML page and that text will show in browser. We'll see how we can publish our website to Azure without Visual Studio.

So, create a simple application from Visual Studio and make some HTML text in to HTML page. Here's the screenshot,

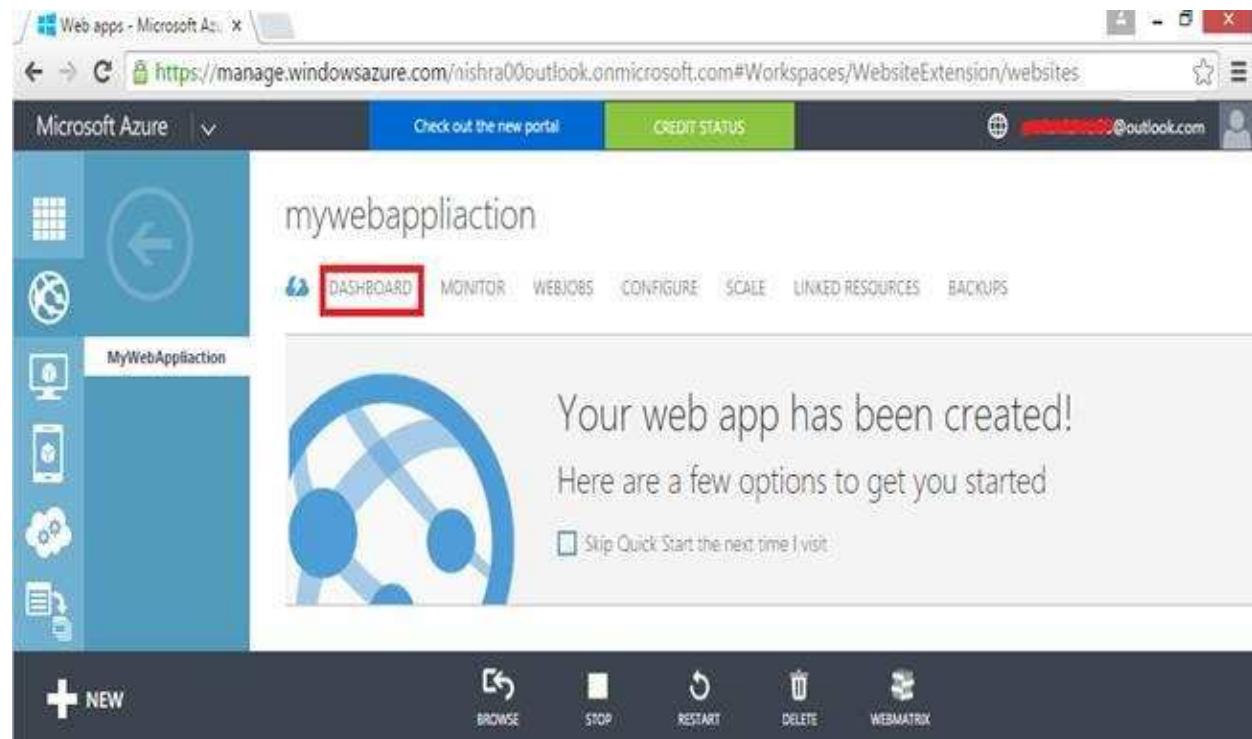
```
<!DOCTYPE html>
<html>
<head>
    <title>Hello Microsoft Azure</title>
    <meta charset="utf-8" />
</head>
<body>

    <hi>Hello Azure..I'm Nitin Pandit with Web App in Azure with FileZilla</hi>
    <p>So my Web App is created by Visual Studio 2013 and Microsoft Azure, would you like to join me?</p>
    <p>Cloud is so interesting thing.....</p>

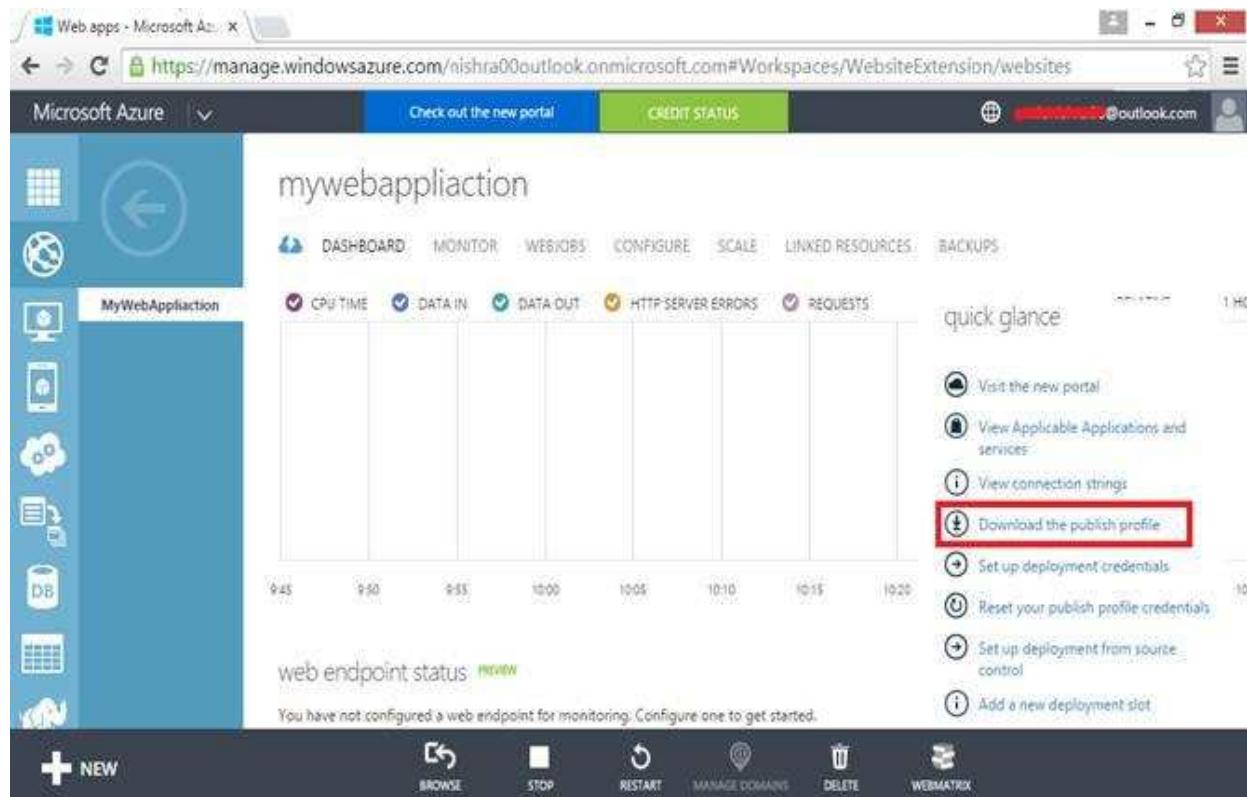
</body>
</html>
```

So, now our Web App and HTML text are completed, go back to your Web App.

Click on to your Existing Web App and select “DASHBOARD” to download the publish profile of your Web App.



In dashboard panel, click on to “Download the Publish Profile”.



The screenshot shows the Microsoft Azure Web Apps dashboard for a website named 'mywebappliaction'. The left sidebar lists various services: Storage, App Service, API Management, Functions, Logic Apps, Container Registry, and Database. The 'MyWebApplication' service is selected. The main area displays monitoring data for CPU TIME, DATA IN, DATA OUT, HTTP SERVER ERRORS, and REQUESTS over a time range from 9:45 to 10:20. Below this is a 'quick glance' section with several options, one of which, 'Download the publish profile', is highlighted with a red box. At the bottom, there are actions for BROWSE, STOP, RESTART, MANAGE DOMAINS, DELETE, and WEB MATRICES.

Right click on the downloadable file and open it with *Notepad* to collect all the FTP information.



In the following screenshot, you will get publishUrl (which is your host name), username and user password.

```

MyWebAppliaction.azurewebsites.net.PublishSettings - Notepad
File Edit Format View Help

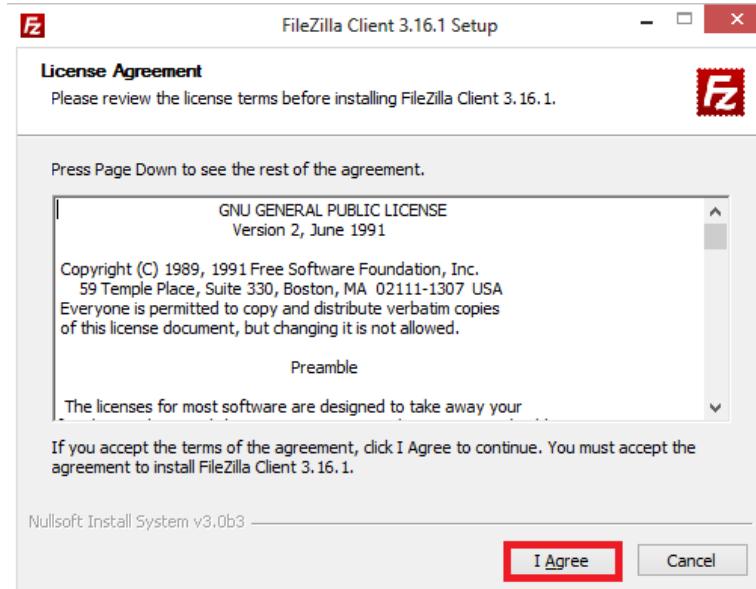
<publishData><publishProfile
profileName="MyWebAppliaction - Web Deploy"
publishMethod="MSDeploy"
publishUrl="mywebappliaction.scm.azurewebsites.net:443"
msdeploySite="MyWebAppliaction"
userName="$MyWebAppliaction"
userPWD="z60oS2iCBLYsfnSlDcd4WLdEZkiRzt7yrjZSfbkw5n0rGd8HYjuEWTFeTrov"
destinationAppUrl="http://mywebappliaction.azurewebsites.net"
SQLServerDBConnectionString="" mySQLDBConnectionString=""
hostingProviderForumLink=""
controlPanelLink="http://windows.azure.com"
webSystem="WebSites"><databases/></publishProfile><publishProfile
profileName="MyWebAppliaction - FTP" Host Name
publishMethod="FTP"
publishUrl="ftp://waws-prod-hk1-013.ftp.azurewebsites.windows.net/site/wwwroot"
ftpPassiveMode="True"
userName="MyWebAppliaction\MyWebAppliaction" User
userPWD="z60oS2iCBLYsfnSlDcd4WLdEZkiRzt7yrjZSfbkw5n0rGd8HYjuEWTFeTrov" Pwd
destinationAppUrl="http://mywebappliaction.azurewebsites.net"
SQLServerDBConnectionString="" mySQLDBConnectionString=""
hostingProviderForumLink=""
controlPanelLink="http://windows.azure.com" webSystem="WebSites"><databases/>
</publishProfile></publishData>

```

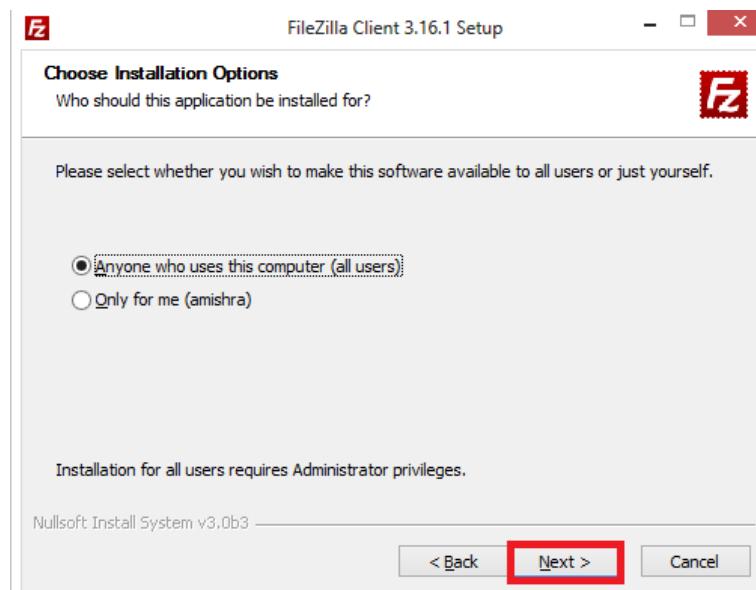
Now you will need to download FileZilla software from its official website. For more you can see the following screenshot,



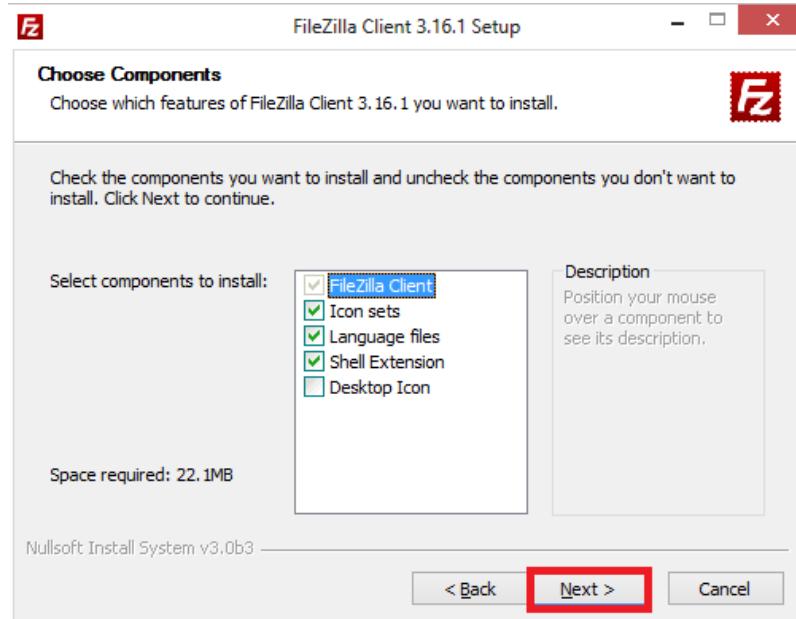
Accept the terms of license agreement by clicking on “*I Agree*”.



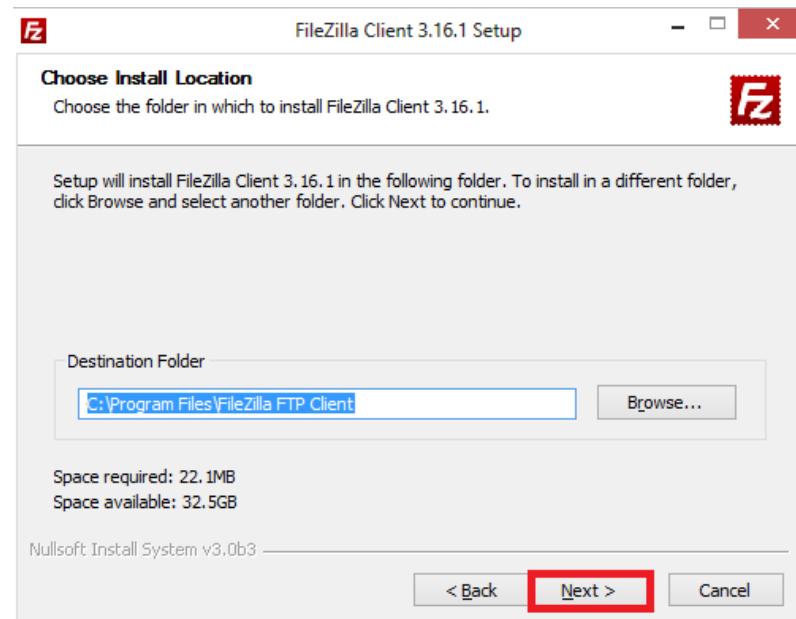
This will ask for administration of FileZilla whether you want for *Anyone* or only for you, so here I'm selecting all users, click on to next button.



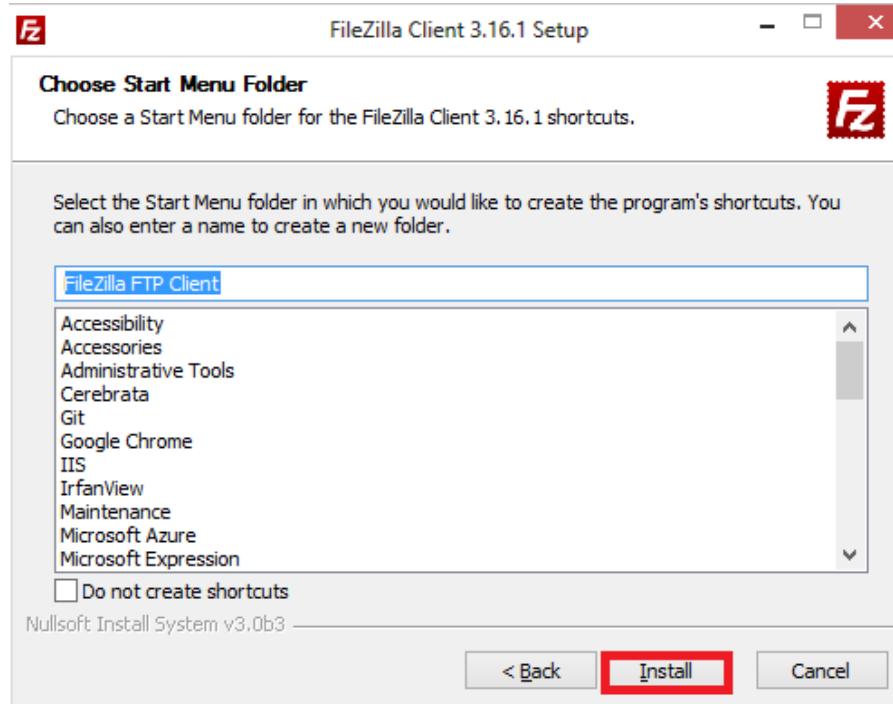
Next step of setup is selecting components of FileZilla which you want to install, so mark all the components and click on the next button.



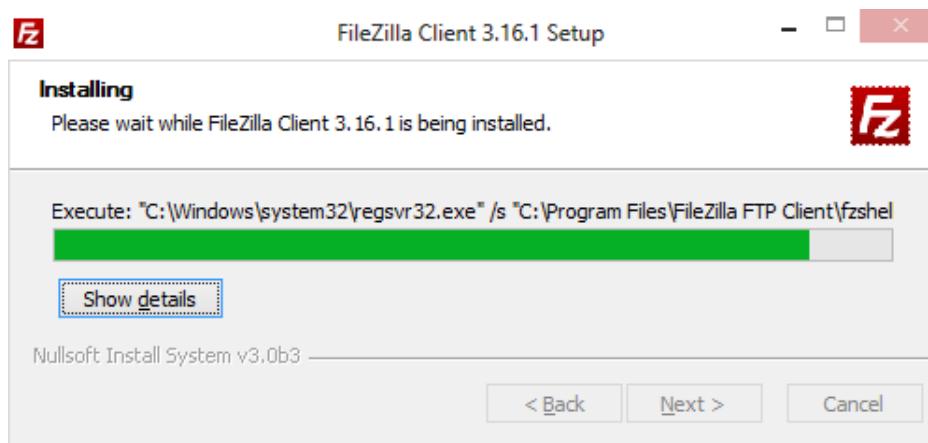
Select the path of installation, and click on the next button.



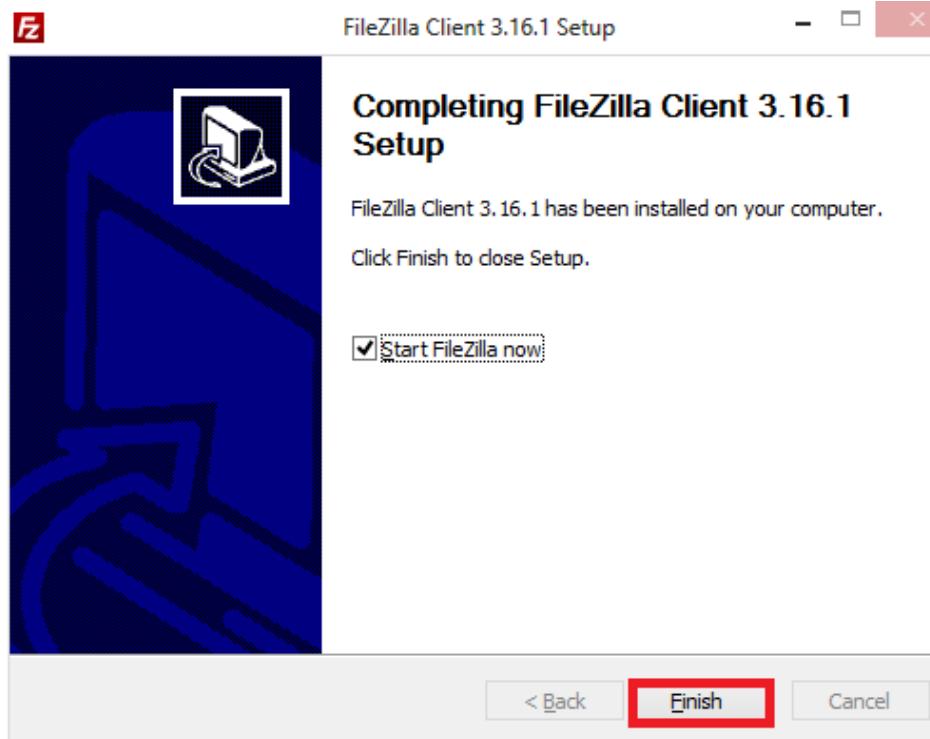
The following screenshot is asking for start menu and if you do not want to create a shortcut then check on "*Do not create shortcut*", and click on “*Install*” button.



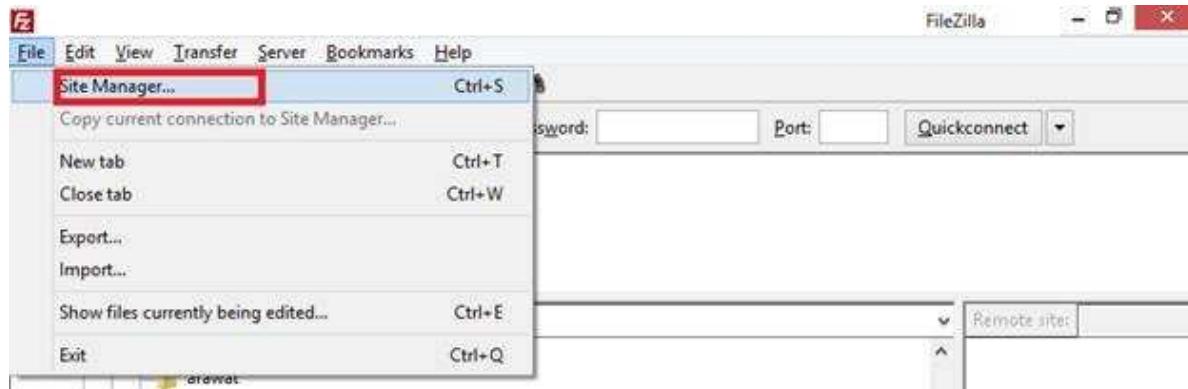
Your setup is being installed.



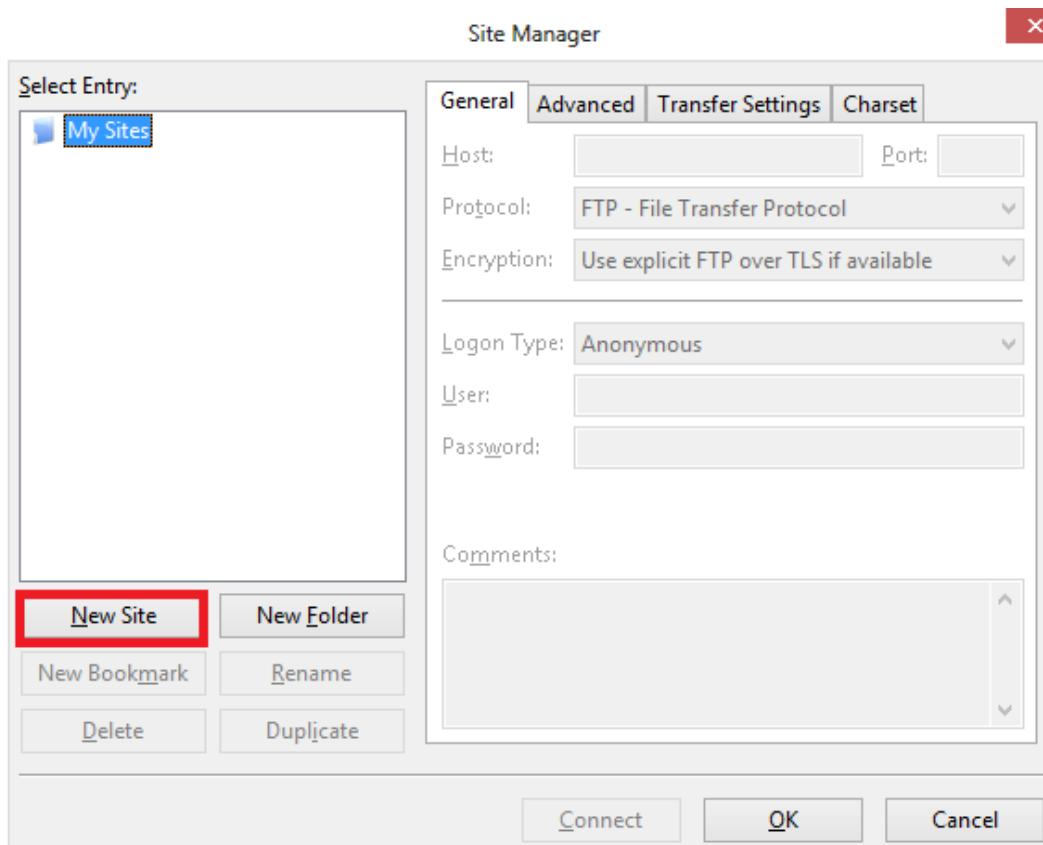
Click on to finish button to complete the process of setup.



In *File* menu select “Site Manager”.



Under the site manager, click on to “New Site”.



Give a name to your new site.

Under the *General* tab in the right side give the following values according to the following screenshot,

Host: Paste the hostname from publishUrl obtained from the published settings file above in Notepad.

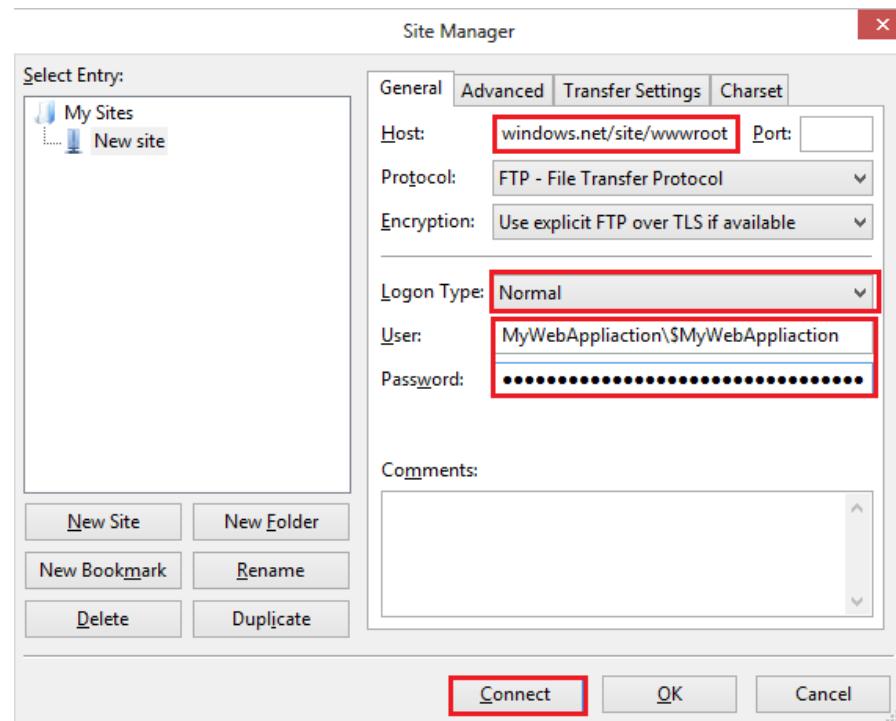
Logon Type: Set this to Normal.

User: Paste the userName obtained from the publish settings file above in Notepad.

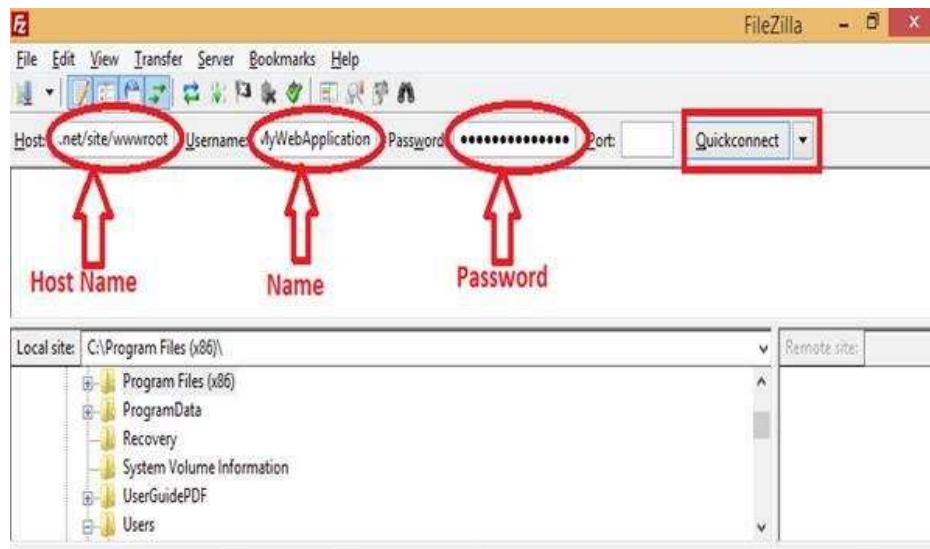
Password: Paste the userPWD obtained from the publish settings file above in Notepad.

Click on *Connect* to connect to the site over FTP.

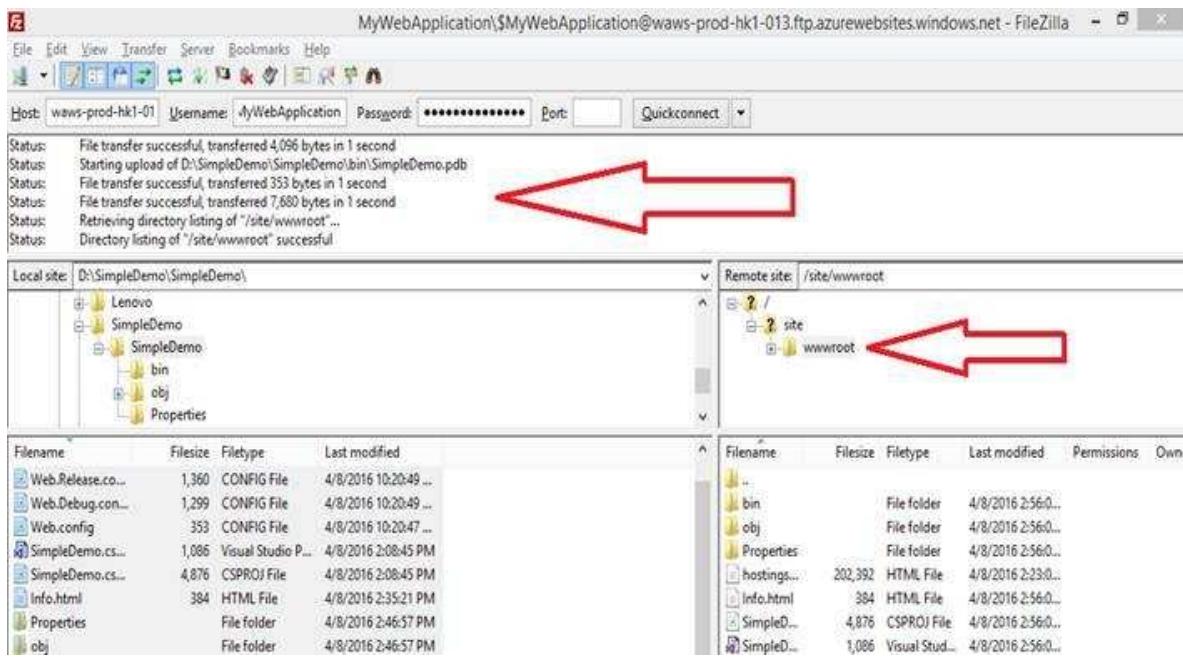
You will see two folders under the root: Logfiles and Site.



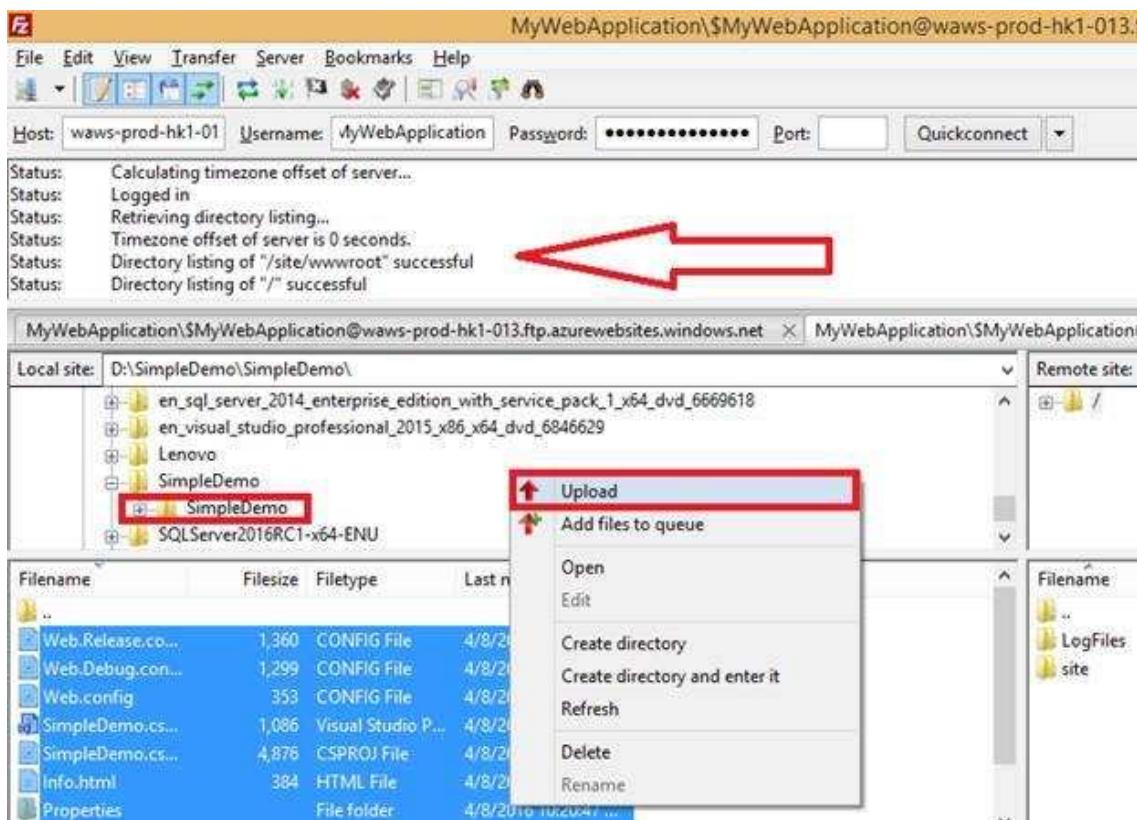
Or you can follow the following procedure,



After connecting to the server your status will be successful and *wwwroot* folder will get created in directory.

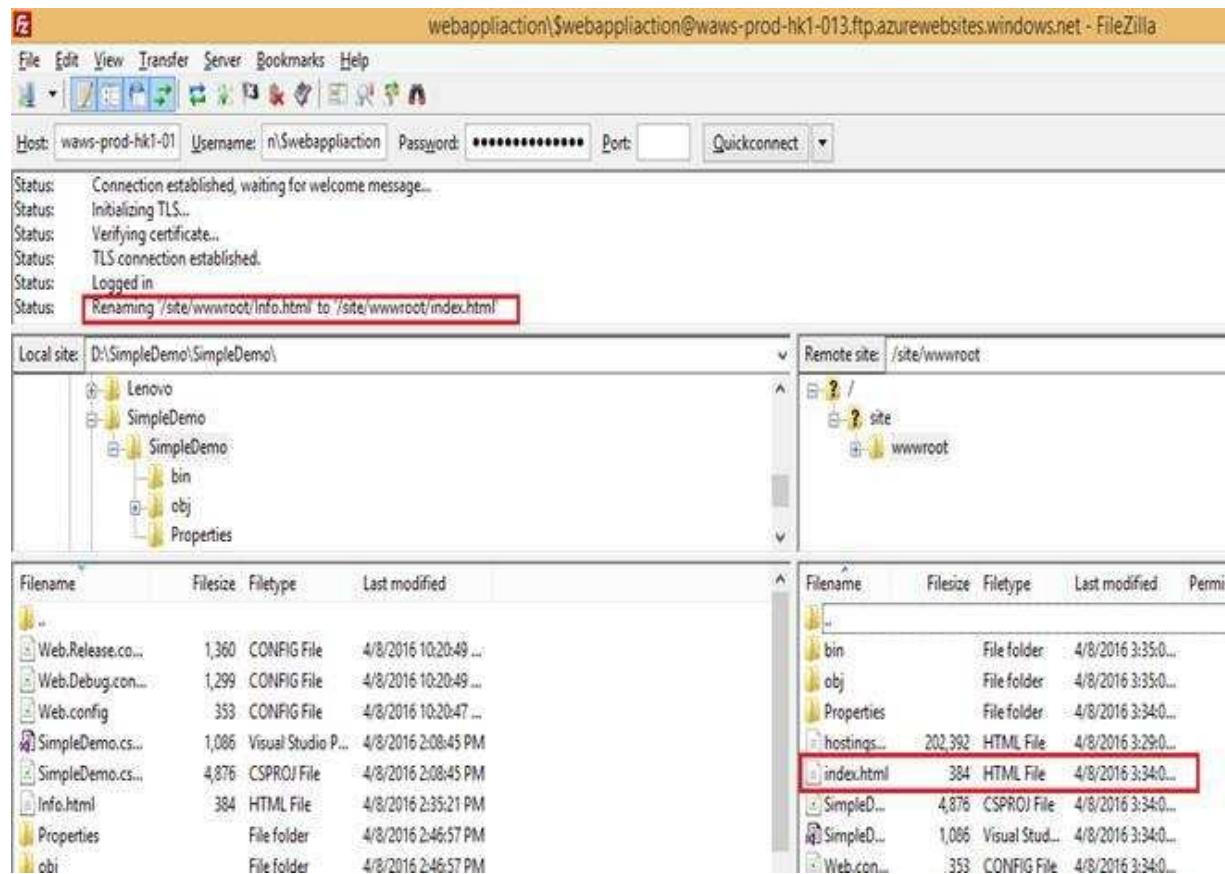


Now go to your Visual Studio project and select all the files of project, by clicking right, upload it.

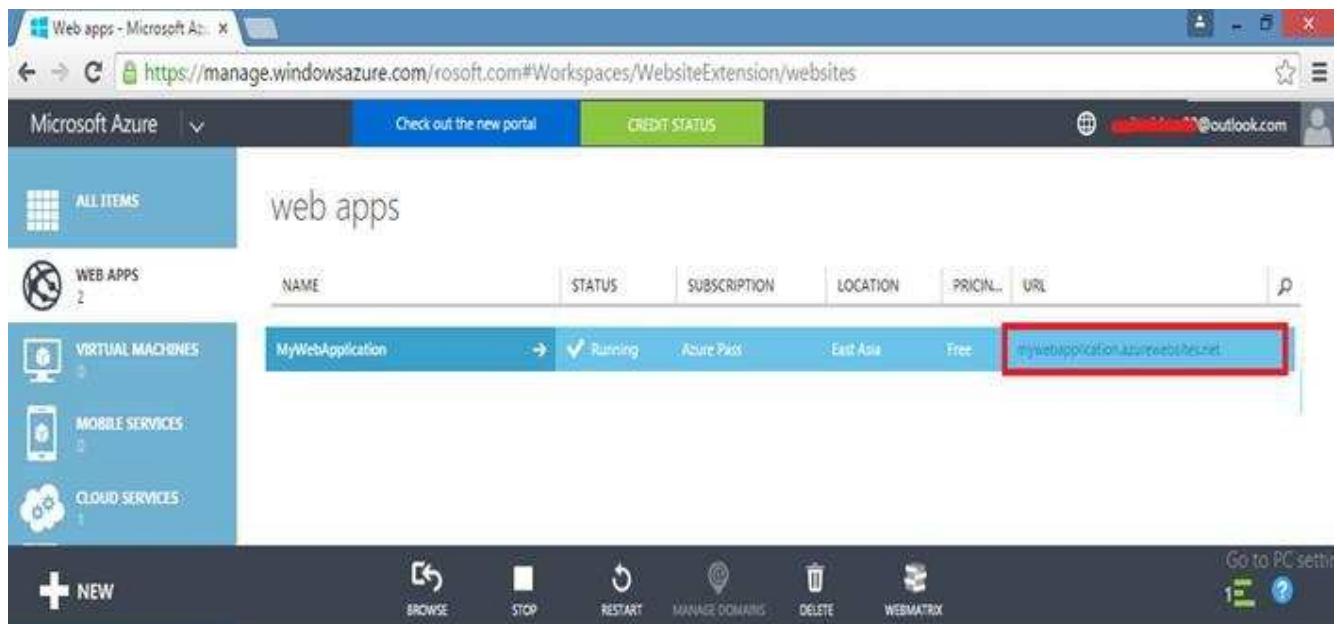


So, we created an html file with “info” named in my Visual Studio, change the name of file info.html to index.html. For more see the following screenshot.

We have a connection with the FTP Server on Azure Cloud and on our local PC.



If everything is completed successfully then go back to your Web App and click on it.

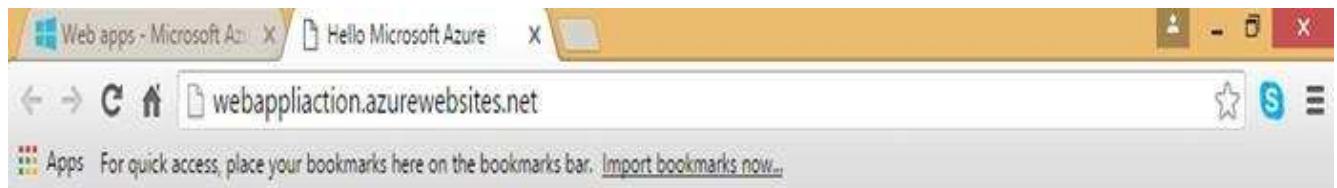


The screenshot shows the Microsoft Azure management portal for web apps. On the left sidebar, under 'WEB APPS', there is one item named 'MyWebApplication'. The main table displays the following details:

NAME	STATUS	SUBSCRIPTION	LOCATION	PRICING	URL
MyWebApplication	Running	Azure Pass	East Asia	Free	mywebapplication.azurewebsites.net

Below the table are several actions: BROWSE, STOP, RESTART, MANAGE DOMAINS, DELETE, and WEBMATRIX. A red box highlights the URL column.

So here I got my result, I have successfully deployed our HTML page. We can publish our website to Azure without Visual Studio.



The screenshot shows a web browser window with the address bar containing 'webappliation.azurewebsites.net'. The page content displays the text 'Hello Microsoft Azure'.

Hello Azure..I'm Nitin Pandit with Web App in Azure with FileZilla

So my Web App is created by Visual Studio 2013 and Microsoft Azure, would you like to join me?

Cloud is so interesting thing.....

I hope you enjoyed this chapter.

Configuring a Custom Domain Name for an Azure Website

A domain name is a URL which is responsible to access websites. Domain name identifies IP address. Domain name is an identification string which defines a scope of administrative autonomy, control within Internet. Any domain name locates Internet address for an URL at points to Internet and particular host server named.

For managing our custom domain name, you must have to login with Azure.

Firstly, we need to create a Web App, so go to Web Apps blade and click on “New” button in the bottom of the page.



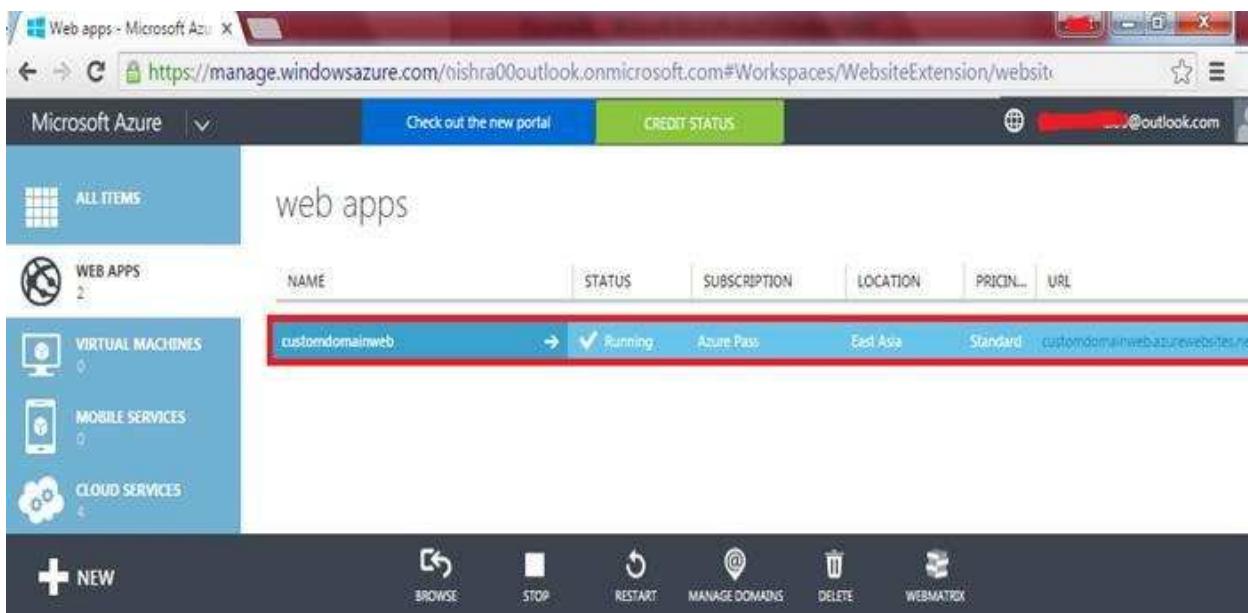
In compute section creates a Web App.

Give a name to your Web App; here I have given it “customdomainweb”.

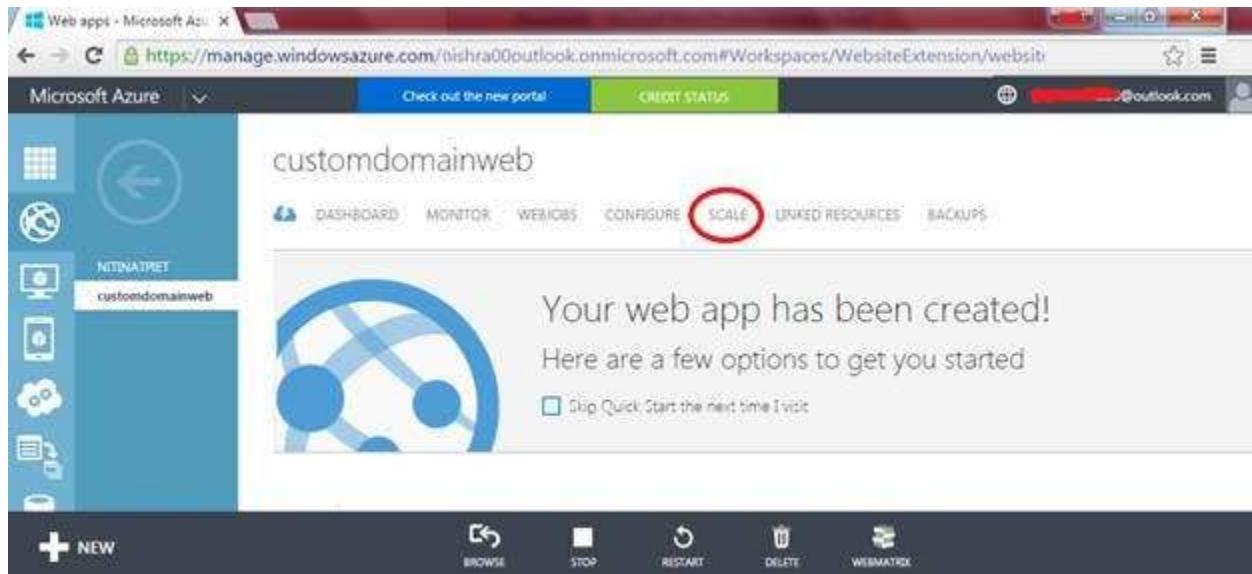
Select your Service Plan whether you want the location of your Web App and then click on to “*CREATE WEB APP*”.



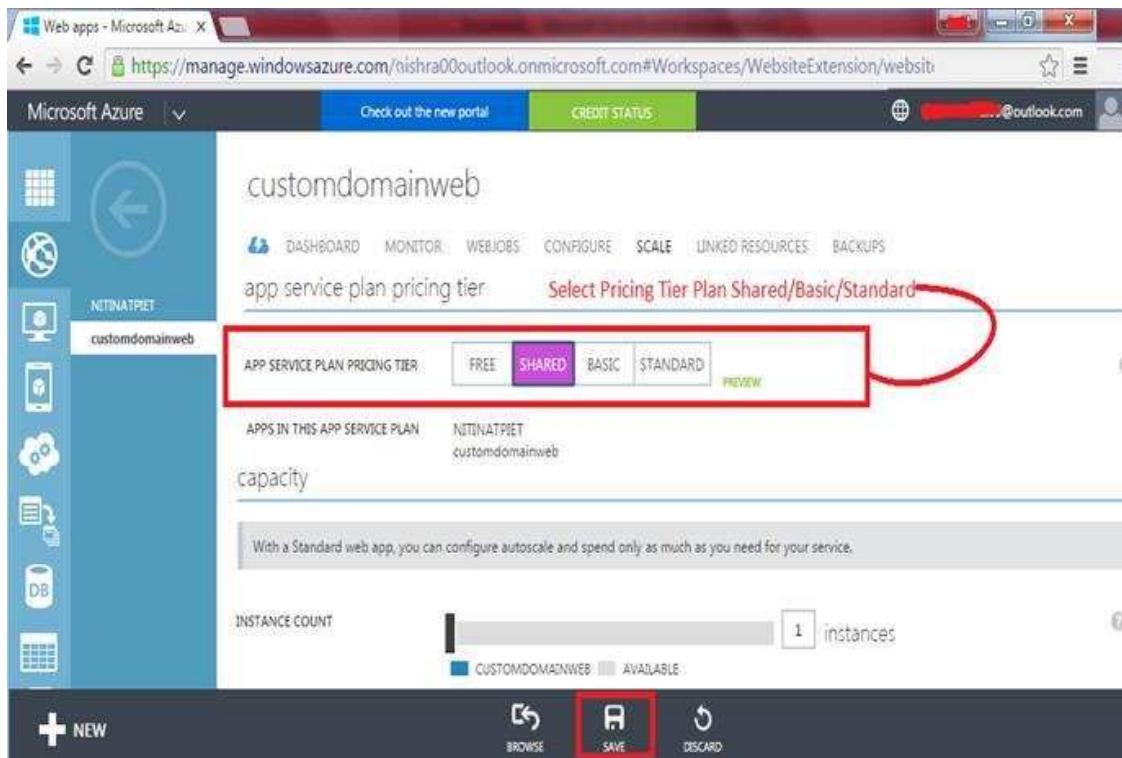
Web App will be successfully created, click on it for further.



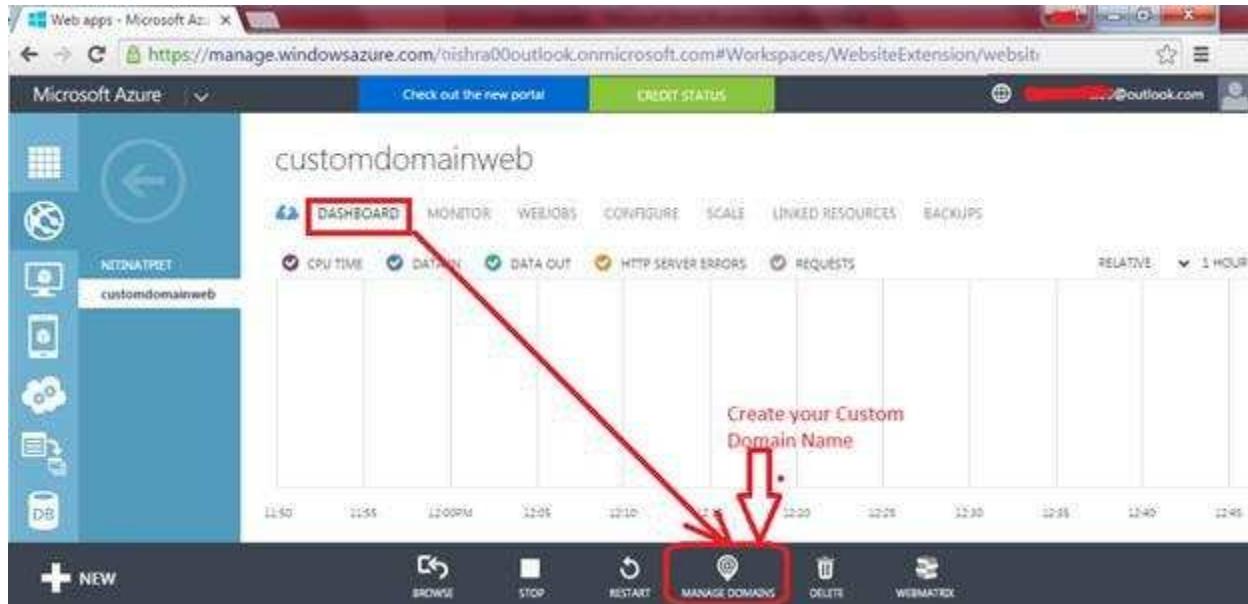
Click on “*SCALE*” panel to select the mode for your Web App.



In the following screenshot, for custom domain name we need to select app service plan pricing tier which is available in Shared, Basic and Standard modes for the Web App. Select *Shared* plan and save it.



Click on to “*Dashboard*” to manage custom domain, in the bottom of the window hit on “Manage Domains”.



So, my Web App has a domain and virtual IP Address, we need to map this IP Address to the specific domain which we can buy from the Godaddy website.

Put your domain name which you want to set, when you want to put your domain name it will mark as red because we do not map with specific domain name.

Here also we can see the IP Address and CNAME (Canonical Name) to map our particular website to specific domain name.

Manage custom domains

You can point custom domain names to your Windows Azure websites. Windows Azure must verify that you are authorized to configure the custom domain name to point to your Windows Azure website. To verify authorization, create a CNAME resource record with your DNS provider that points from either www.yourdomain.com to customdomainweb.azurewebsites.net, or from awverify.www.yourdomain.com to awverify.customdomainweb.azurewebsites.net. When using Traffic Manager, create a CNAME resource record with your DNS provider that points to the *.trafficmanager.net URL.

[Learn more about managing custom domains](#)

DOMAIN NAMES

customdomainweb.azurewebsites.net

4INSANIYAT.COM



THE IP ADDRESS TO USE WHEN YOU CONFIGURE A RECORDS

23.99.110.192

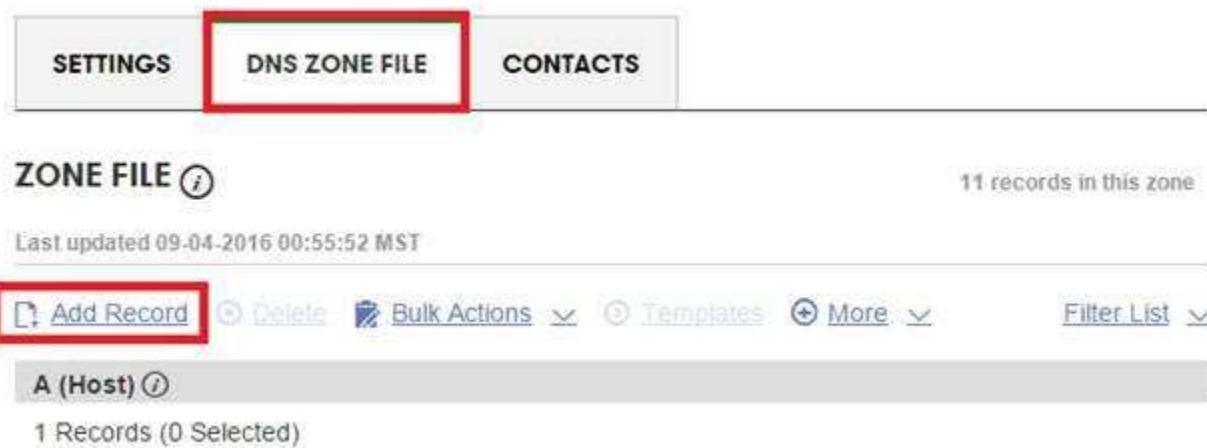
IP address will help to
configure a record

When you put your
domain name it will
given error



Log into your domain provider, here, I'm using Godaddy account, and go to “DNS Zone File” panel to make some changes.

We need to add a record of Host, click on to “Add Record”.



Type	Name	Value
A (Host)		23.99.110.192

A record, maps a domain name to an IP Address means *customdomainweb.com* mapped to my IP.

Record Type: Record Type would be A(Host) from the dropdown menu.

Host: In the Host panel give to @ symbol.

Points To: Enter IP Address of your Web App.

According to your choice give Time to Live to it.

Click on to finish button to add this record.

ADD ZONE RECORD

X

4INSANIYAT.COM

RECORD TYPE: * [View current](#)

A (Host) Record type would be host

HOST: * [\(i\)](#)

@ Host Name/Symbol

POINTS TO: * [\(i\)](#)

23.99.110.192 Points refers to IP Address

TTL: * [\(i\)](#) Give the Time to Live **SECONDS: ***

Custom

6000

[ADD ANOTHER](#)
[FINISH](#)
[Cancel](#)

A record will be created, for saving the change click on to *Save Changes*.



The screenshot shows the 'ZONE FILE' section of a DNS management interface. At the top, there's a message: 'Manage your DNS seamlessly on this Domain Details page.' and a link to 'Use classic DNS Manager'. Below this, the 'ZONE FILE' title has a '9 records in this zone' link. The main area shows an 'A (Host)' table with one record. The 'Actions' column for this record has a red box around it, and a large red arrow points upwards towards the 'Save Changes' button in the red bar at the top right.

Host	Points To	TTL	Actions
@	23.99.110.192	6000 seconds	<input checked="" type="checkbox"/>

Now we need to add CNAME, which maps a domain name to another domain name means `www.customdomainweb.com` mapped to `customdomainweb.Azurewebsites.net`.

To create a CNAME click on to “Add Record”.



The screenshot shows the 'Add Record' page for 'CName (Alias)'. The 'Add Record' button is highlighted with a red box. The table below shows three existing CNAME records: 'www' pointing to '@' with a TTL of 1 Hour, 'ftp' pointing to '@' with a TTL of 1 Hour, and 'email' pointing to 'email.secureserver.net' with a TTL of 1 Hour. The 'Actions' column for the 'email' record also has a red box around it.

Host	Points To	TTL	Actions
www	@	1 Hour	<input checked="" type="checkbox"/>
ftp	@	1 Hour	<input checked="" type="checkbox"/>
email	email.secureserver.net	1 Hour	<input checked="" type="checkbox"/>

Choose the CNAME from the dropdown menu.

An *awverify* record is used to determine whether our own domain is going to attempt to use something means CNAME record that maps from *awverify.customdomainweb.Azurewebsites.net* to *awverify*.

ADD ZONE RECORD

4INSANIYAT.COM

RECORD TYPE: *

[View current](#)

CNAME (Alias) [Create CNAME](#)

HOST: * ⓘ

awverify [Give to it Host Name](#)

POINTS TO: * ⓘ

awverify.customdomainweb.azurewebsites.net [Maps To](#)

TTL: * ⓘ Time to Live

Custom

SECONDS: *

6000

[ADD ANOTHER](#)

[FINISH](#)

[Cancel](#)

Add another CNAME points to map.

ADD ZONE RECORD

4INSANIYAT.COM

RECORD TYPE: *

[View current](#)

CNAME (Alias) [Create CNAME](#)

HOST: * ⓘ

awverify.www [Give to it Host Name](#)

POINTS TO: * ⓘ

awverify.customdomainweb.azurewebsites.net [Maps To](#)

TTL: * ⓘ Time to Live

Custom

SECONDS: *

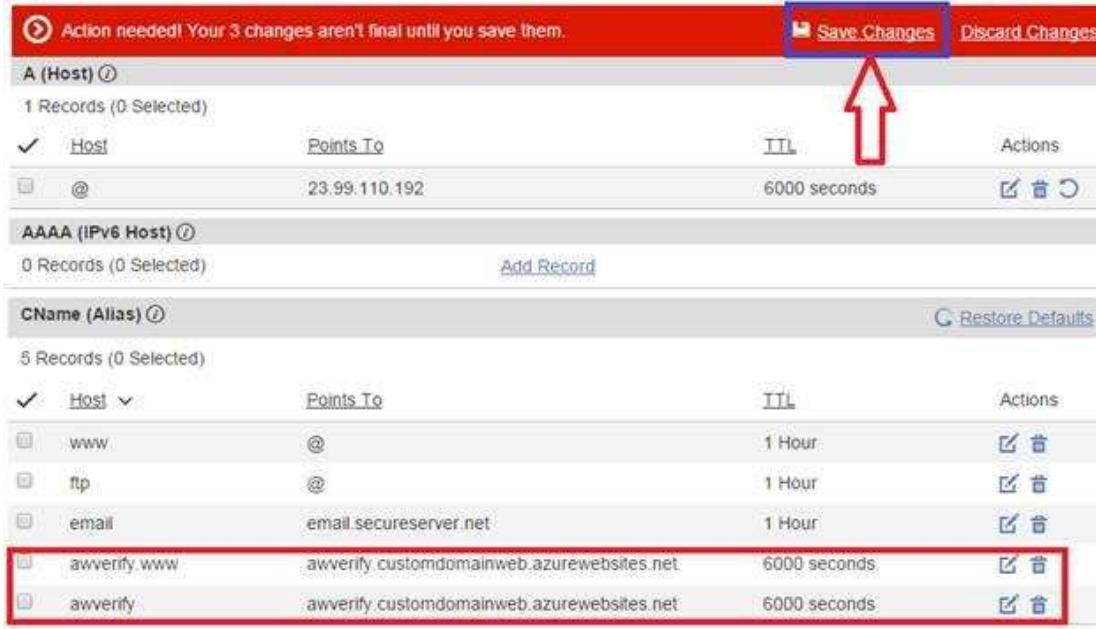
6000

[ADD ANOTHER](#)

[FINISH](#)

[Cancel](#)

Click on to save changes to save all the changes.



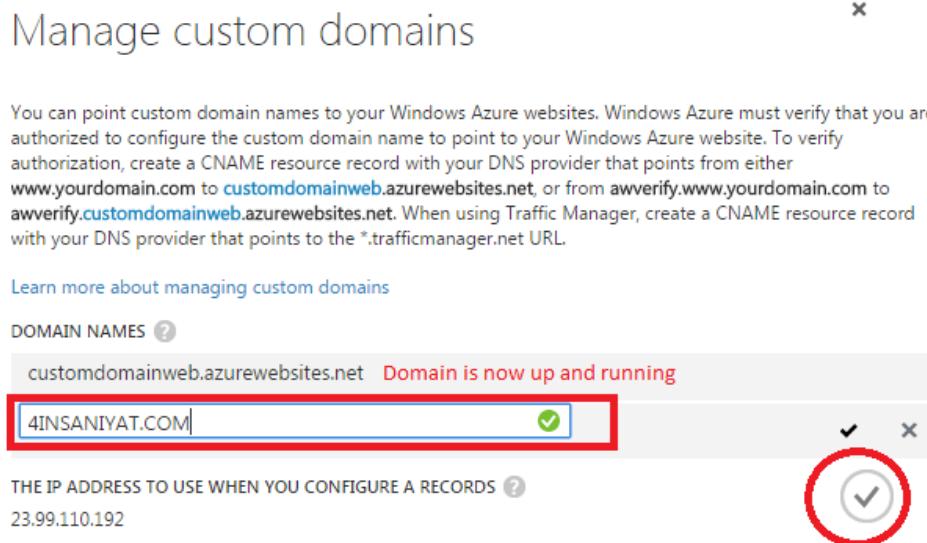
Action needed! Your 3 changes aren't final until you save them.

A (Host) <small>(1 Records (0 Selected))</small>			
<input checked="" type="checkbox"/> Host	Points To	TTL	Actions
<input type="checkbox"/> @	23.99.110.192	6000 seconds	<input checked="" type="checkbox"/> <input type="radio"/> <input type="button" value="Delete"/>

AAAA (IPv6 Host) <small>(0 Records (0 Selected))</small>			
		Add Record	

CName (Alias) <small>(5 Records (0 Selected))</small>			
<input checked="" type="checkbox"/> Host	Points To	TTL	Actions
<input type="checkbox"/> www	@	1 Hour	<input checked="" type="checkbox"/> <input type="radio"/> <input type="button" value="Delete"/>
<input type="checkbox"/> ftp	@	1 Hour	<input checked="" type="checkbox"/> <input type="radio"/> <input type="button" value="Delete"/>
<input type="checkbox"/> email	email.secureserver.net	1 Hour	<input checked="" type="checkbox"/> <input type="radio"/> <input type="button" value="Delete"/>
<input type="checkbox"/> awverify.www	awverify.customdomainweb.azurewebsites.net	6000 seconds	<input checked="" type="checkbox"/> <input type="radio"/> <input type="button" value="Delete"/>
<input type="checkbox"/> awverify	awverify.customdomainweb.azurewebsites.net	6000 seconds	<input checked="" type="checkbox"/> <input type="radio"/> <input type="button" value="Delete"/>

So here in the following screenshot put your domain name, it will be mark as green, means you have completed custom domain name.



Manage custom domains

You can point custom domain names to your Windows Azure websites. Windows Azure must verify that you are authorized to configure the custom domain name to point to your Windows Azure website. To verify authorization, create a CNAME resource record with your DNS provider that points from either `www.yourdomain.com` to `customdomainweb.azurewebsites.net`, or from `awverify.yourdomain.com` to `awverify.customdomainweb.azurewebsites.net`. When using Traffic Manager, create a CNAME resource record with your DNS provider that points to the `*.trafficmanager.net` URL.

[Learn more about managing custom domains](#)

DOMAIN NAMES (1)

customdomainweb.azurewebsites.net Domain is now up and running

4INSANIYAT.COM

THE IP ADDRESS TO USE WHEN YOU CONFIGURE A RECORDS (1)

23.99.110.192

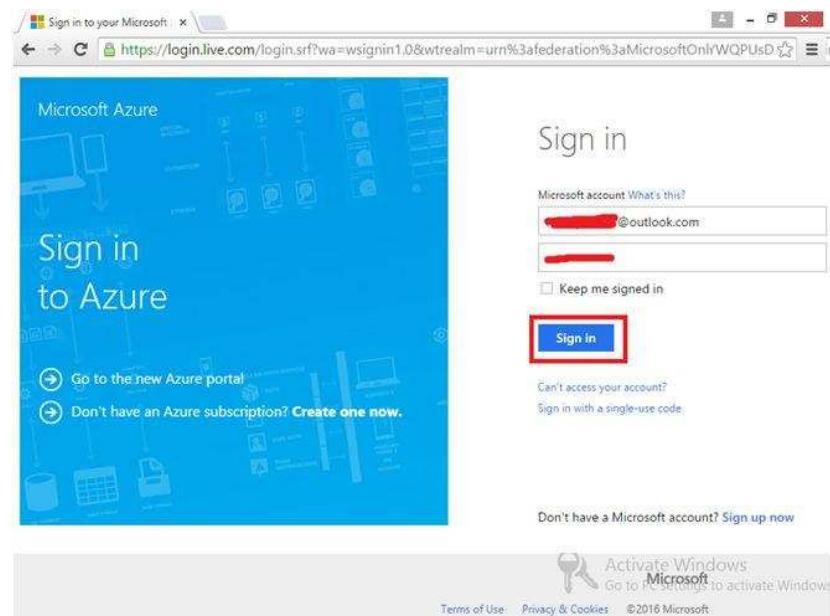
Just Click on button which is marked in circle and enjoy your website live on Azure with the domain name you want.

Virtual Machine in Microsoft Azure Step by Step

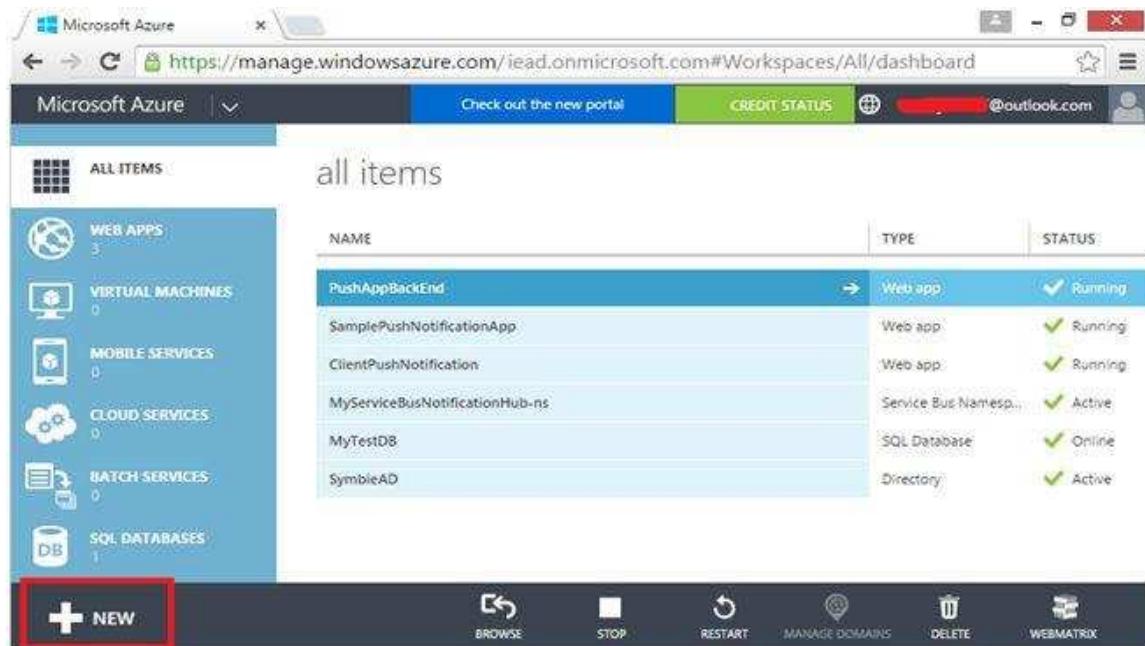
You can create a virtual machine in Azure using one of the following methods:

- Use **Quick Create** when you need only one virtual machine for your solution, or when you need to connect it to a virtual network.
- Use **From Gallery** when you need a solution that requires multiple virtual machines or a virtual machine with advanced settings.
- A virtual machine is a server in the cloud that you can control and manage. After you create a virtual machine in Azure, you can stop and restart it whenever you need to.
- When you create a virtual machine, it is automatically started for you. You'll know when a virtual machine is running because its status is Running. You can stop and restart a virtual machine at any time.
- Before you stop a virtual machine, you should make sure that applications running on the virtual machine are shut down appropriately. The applications that are running on a virtual machine are automatically shut down when the virtual machine is stopped.
- To stop a virtual machine, click its name, and then click Shutdown. When you stop the virtual machine, it shuts down just as your server in your office shuts down when you stop it. While the virtual machine is shutting down, its status is stopping. When the virtual machine is stopped, its status is stopped. If you set up load balancing for your cloud service, you should consider how stopping the virtual machine will affect your application.
- When a virtual machine is running and you want to restart it, click its name, and then click Restart. While the virtual machine is restarting, its status is Restarting. When the virtual machine is ready for you to use, its status is Running.

Step 1: Login to Microsoft Azure.



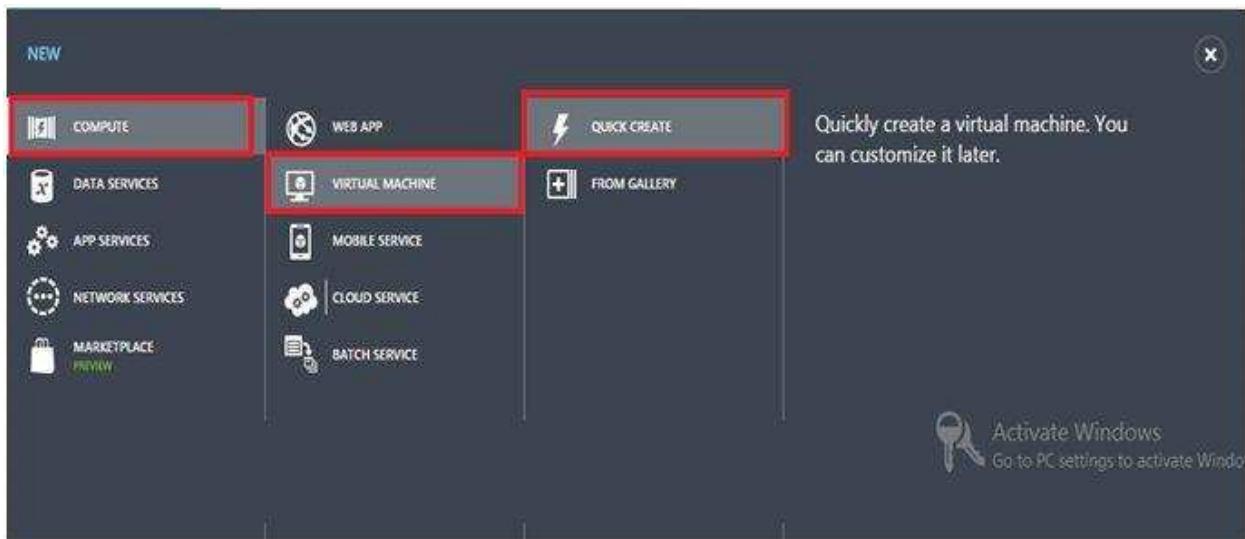
Step 2: By login in to Microsoft Azure, select "NEW" at the bottom of the window.



Name	Type	Status
PushAppBackEnd	Web app	Running
SamplePushNotificationApp	Web app	Running
ClientPushNotification	Web app	Running
MyServiceBusNotificationHub-ns	Service Bus Namesp...	Active
MyTestDB	SQL Database	Online
SymbileAD	Directory	Active

Step 3: Under "COMPUTE", select "Virtual Machine" and select "Quick Create".

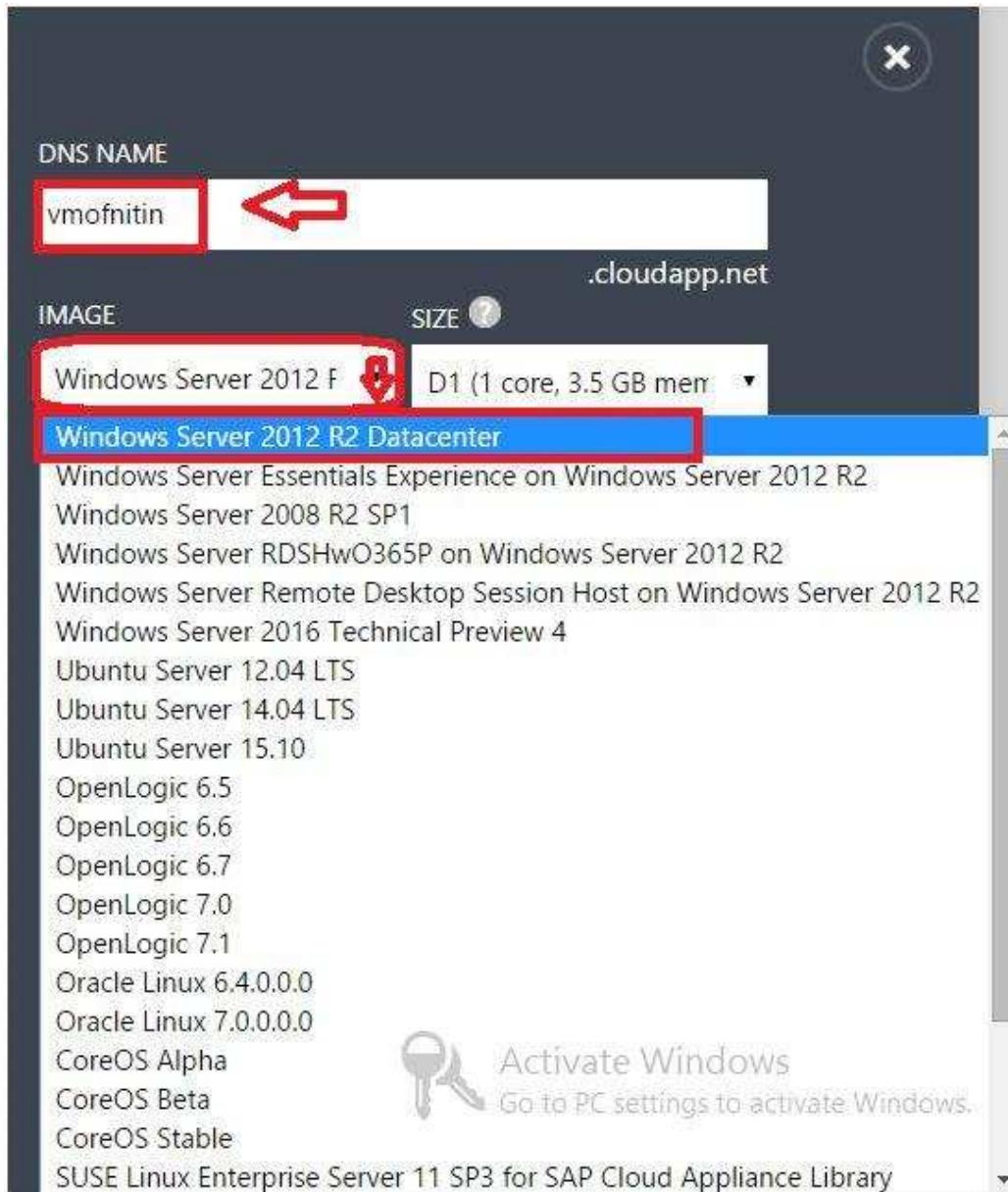
Here, I am going to create **Quick Create** when we need only one virtual machine for our solution.



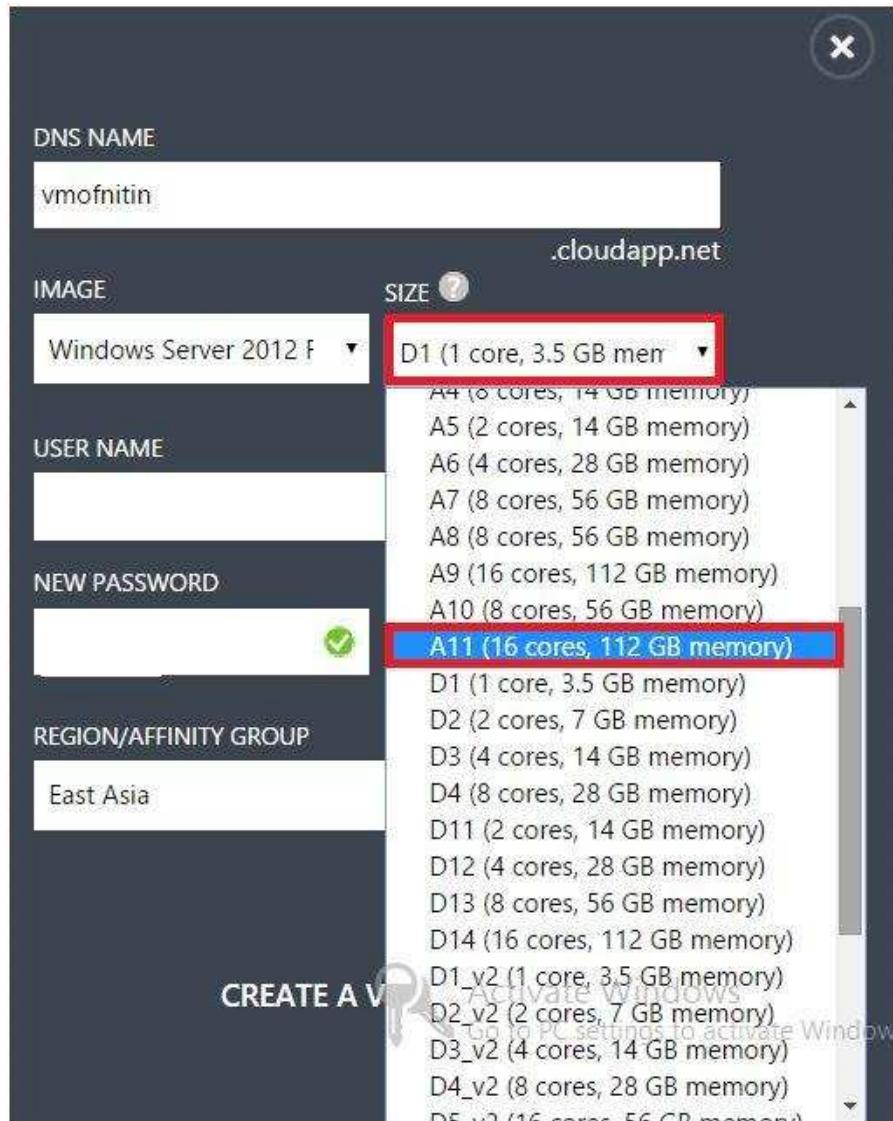
You will see the following figure,

DNS: The DNS name must contain between 3 and 15 characters. The name can contain only letters, numbers, and hyphens. The name must start with a letter and must end with a letter or a number.

Image: Choose an Operating System image from the drop down menu; here I want to show "**Windows Server 2012 R2 Datacenter.**"



Size: In size, you can give a size to your virtual machine from different sizes in the dropdown menu.

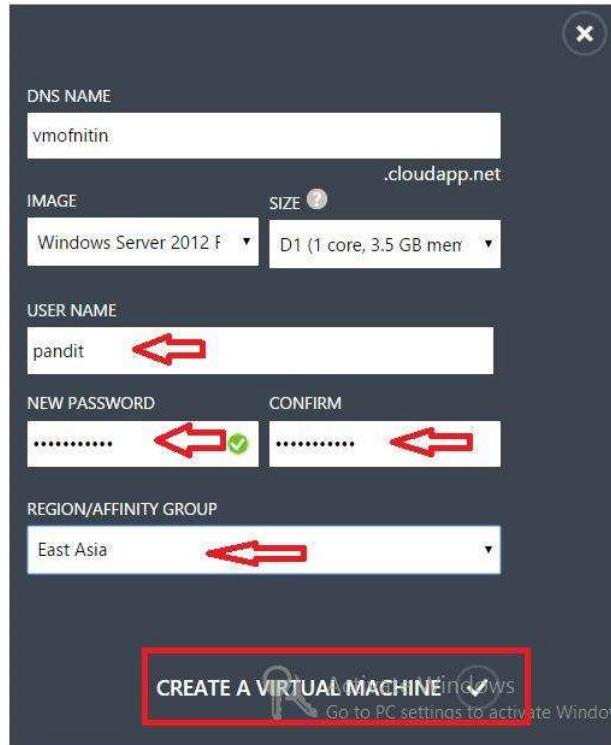


User Name: User Name is mandatory for the VM, it would be used at the time of credentials. Here I gave "pandit."

Password: Give the password for logging in to the virtual machine.

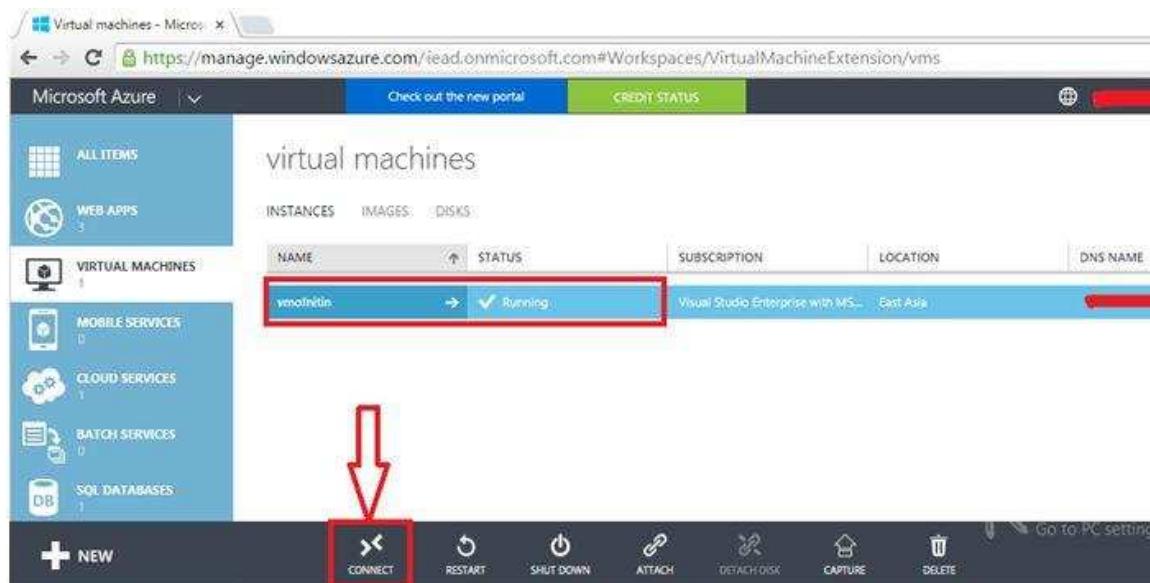
Region: I'm from India, so I am going to select "**East Asia.**"

Hit on check mark to create a virtual machine (shown in the rectangle).



After the creation of the virtual machine, we need to connect it where we can access it remotely. Select your virtual machine and click on "**Connect**" button.

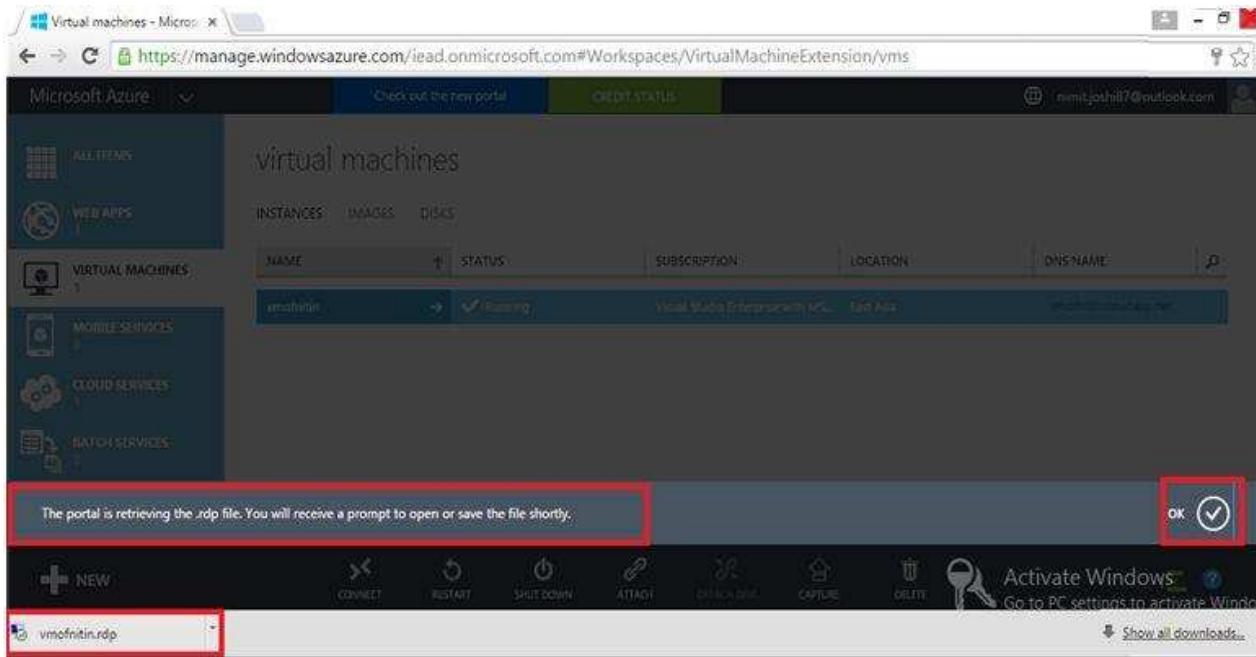
A file with extension ".rdp" will be downloaded.



NAME	STATUS	SUBSCRIPTION	LOCATION	DNS NAME
vmofnitin	Running	Visual Studio Enterprise with MS...	East Asia	vmofnitin

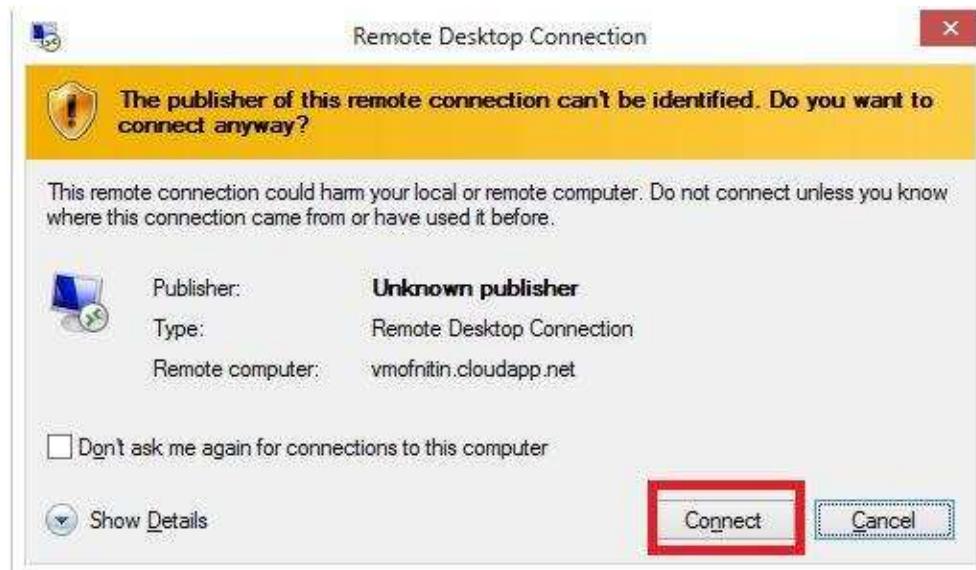
It will display a message for the file.

Open the downloaded RDP file.

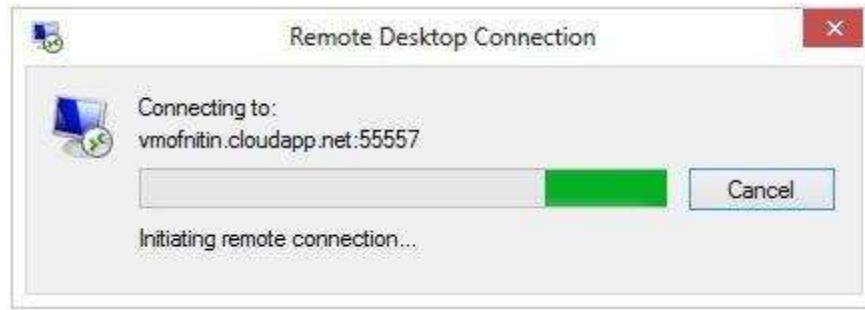


Go to the downloaded RDP file and open it. Remote Desktop is enabled by default.

Click on "Connect" button.



After clicking on connect button it will start the remote connection.



Now, we need to complete security by entering credentials, user name and password which we setup during creation of the virtual machine.

Click on "OK" button.



Click on "Yes" button.



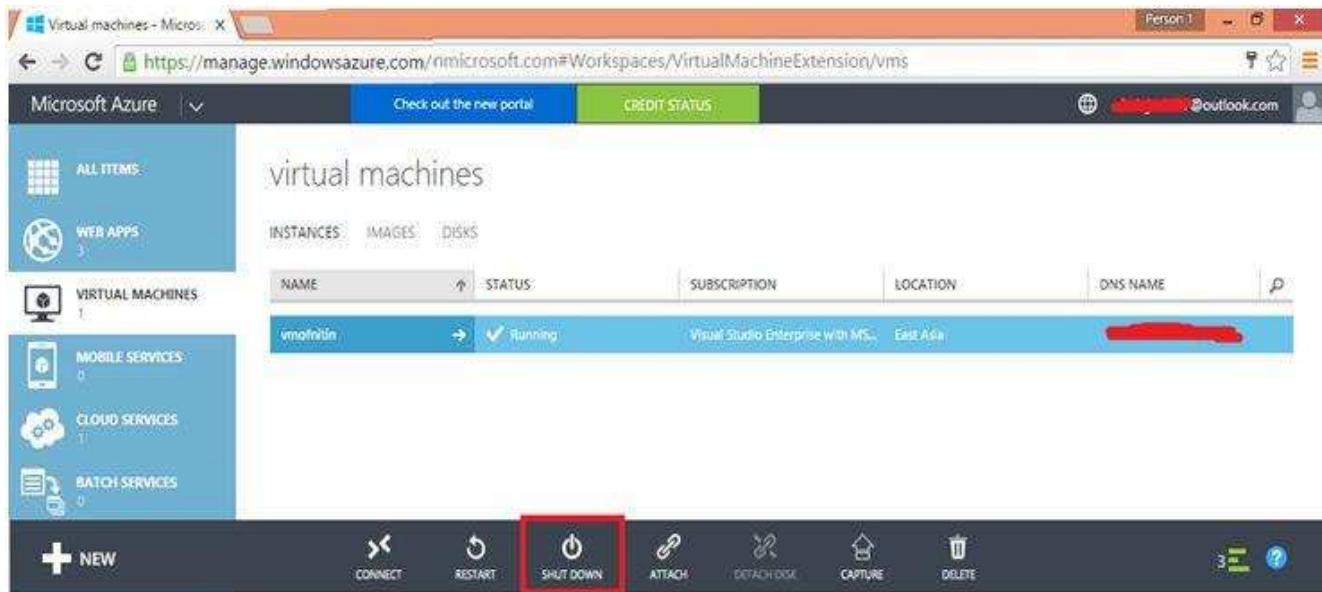
After entering your credentials, click on "yes" and you will be connected with your Virtual Machine.



Here is the **Dashboard**.



Make sure you don't forget to shut down your VM after completing your work. You can shut down it from **shutdown** option as in the following:



Creating a Virtual Disk Virtual Machine in Microsoft Azure Step by Step

In the previous chapter we discussed how we can create a virtual machine from “Quick Create” option in part one of the series.

Now, here I am going to create a virtual machine using the “**FROM GALLERY**” option. This is for when you need a solution that requires multiple virtual machines or a virtual machine with advanced settings.

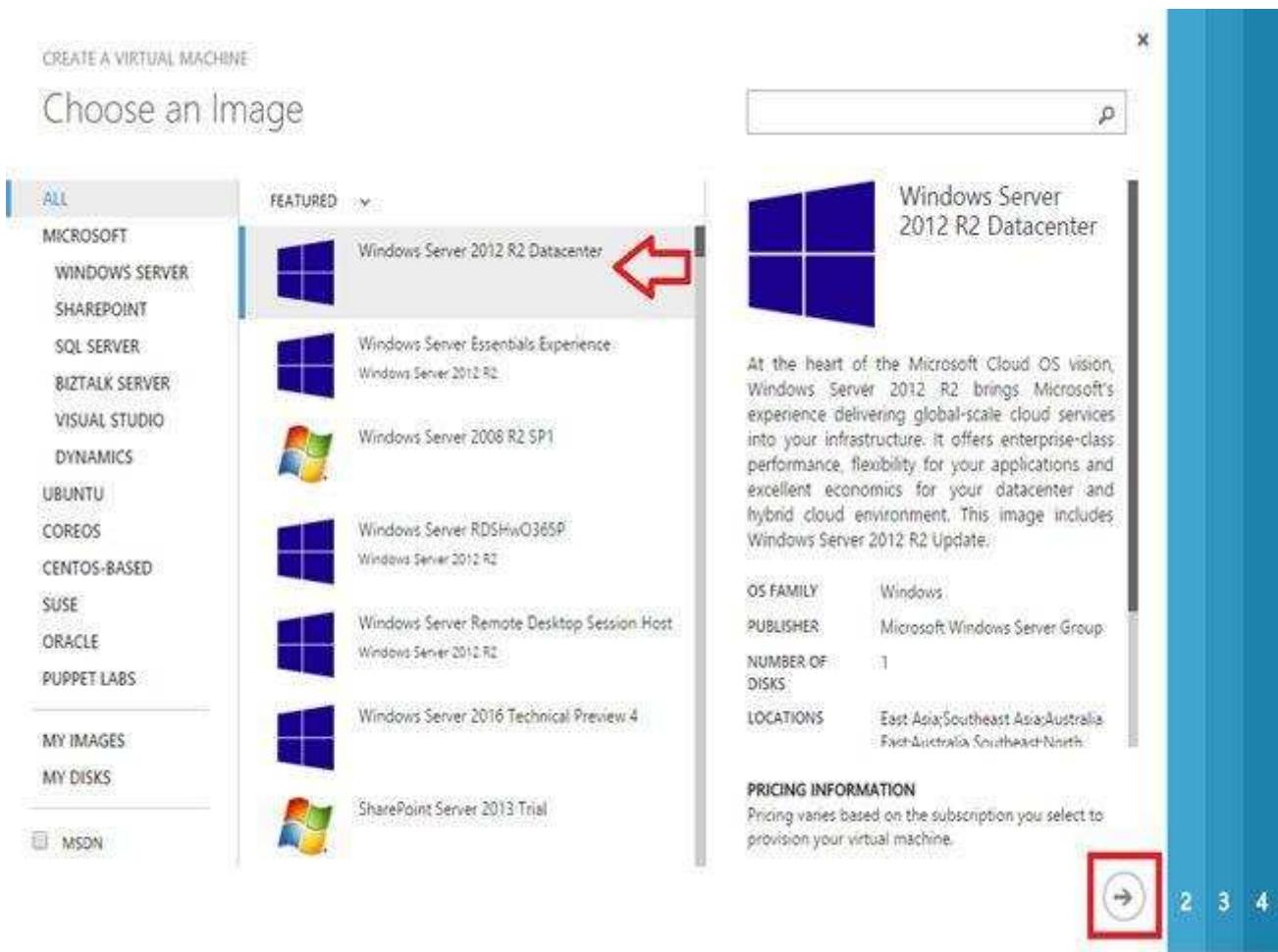
Select “**COMPUTE**” and “**VIRTUAL MACHINE**” and select “**FROM GALLERY**”.



Azure gallery provides Windows + Linux and their entire version. Select your desired operating system image and click next. I have selected **Windows Server 2012 R2 Datacenter**.

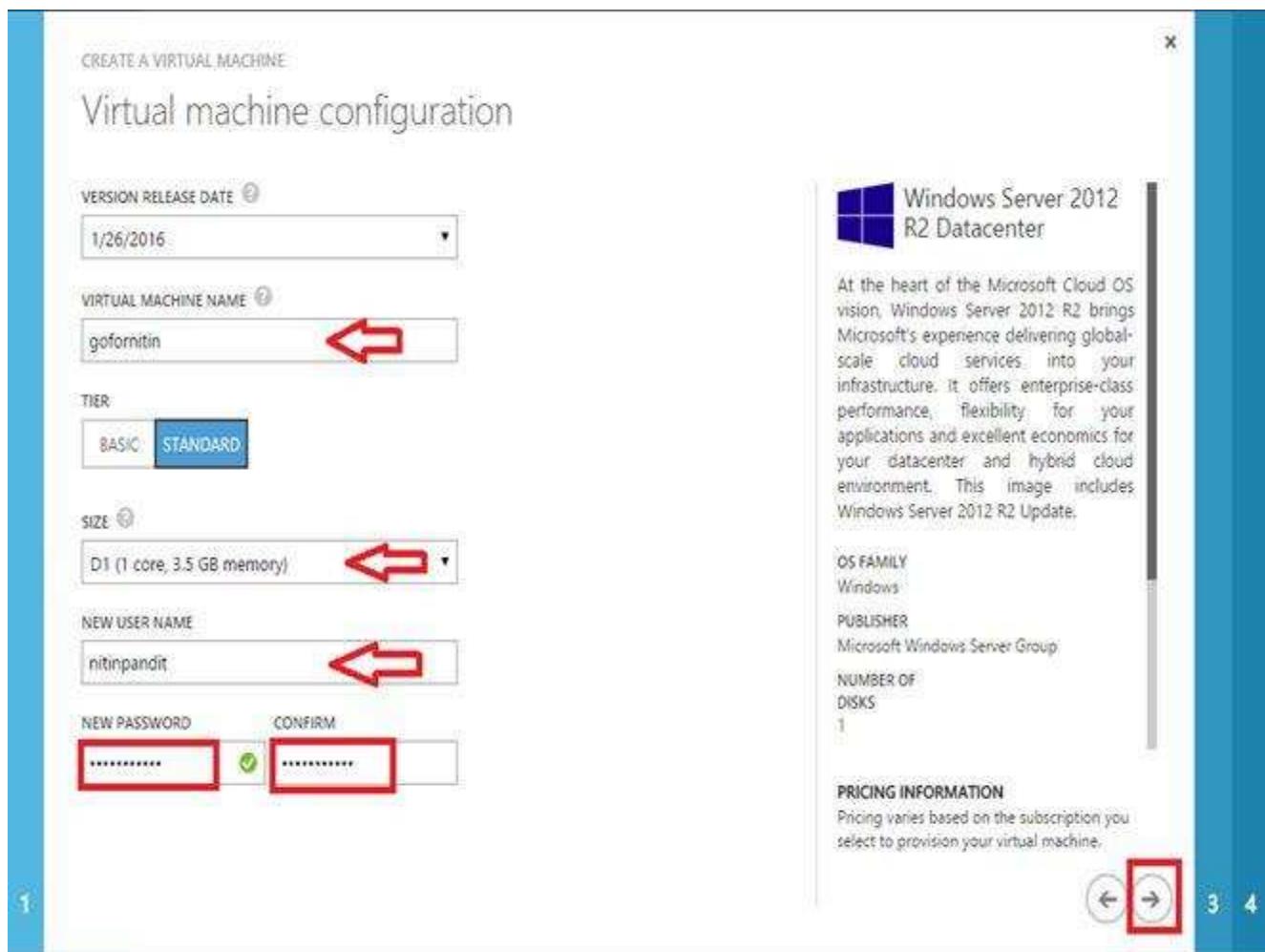
So select an image, whatever you want to show with your virtual machine.

Here, I'm going to select “**Windows Server 2012 R2 Datacenter**”.

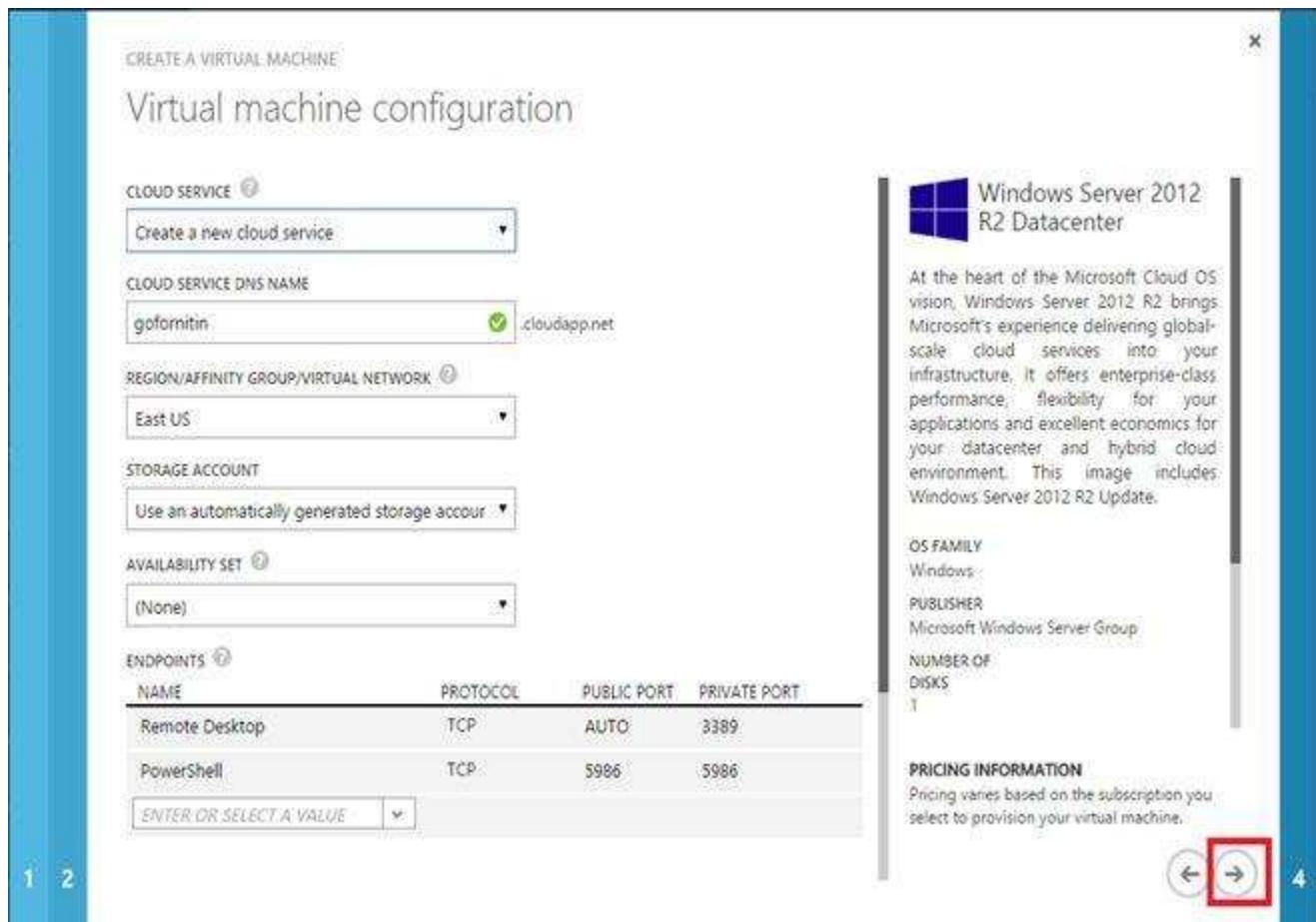


Give the Virtual Machine name, Tier, size, username and password.

Hit on Next (Shown in Rectangle at the bottom of the page).

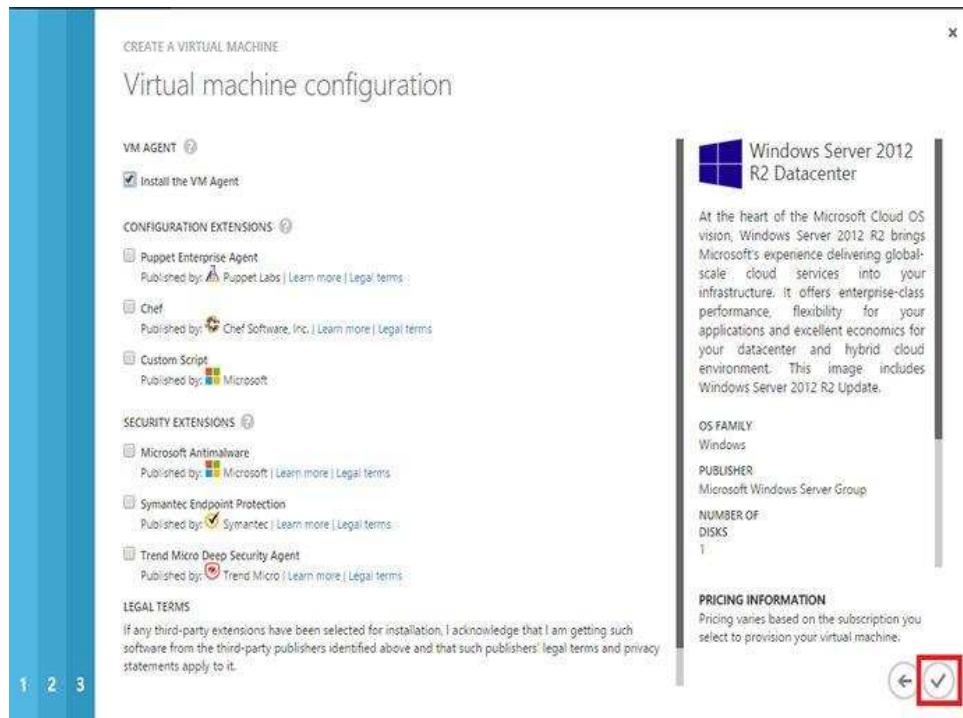


Here next is configuration window; all is set by default, so click next.

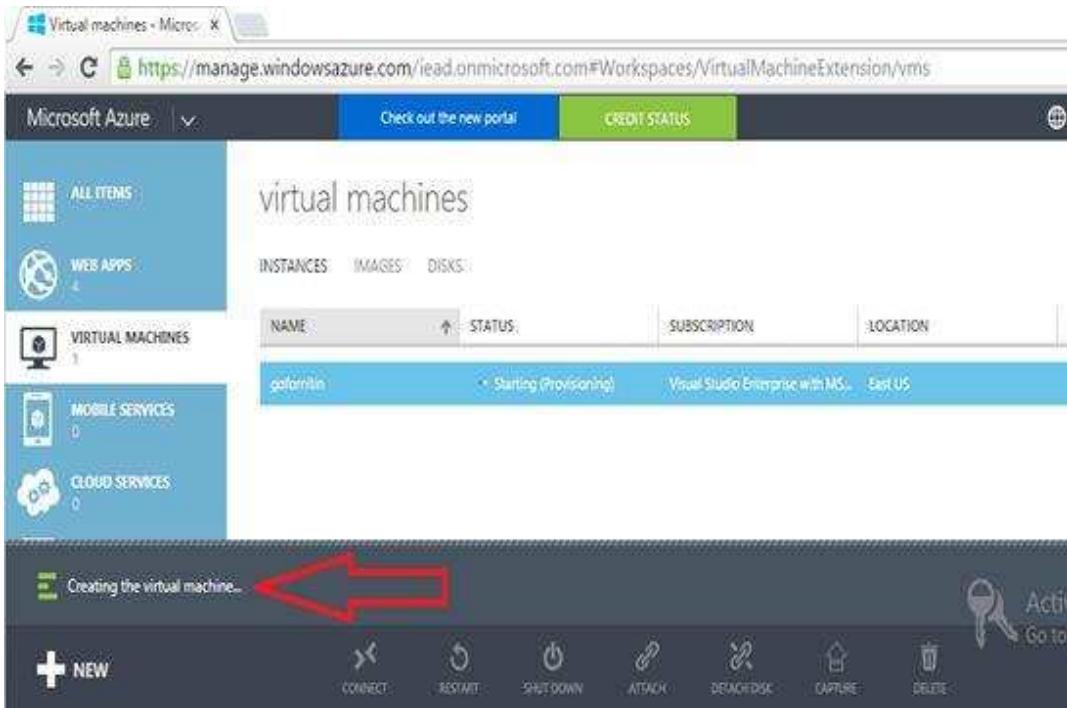


You can see the following window, it will ask for configuration extensions and security extensions, so here all is set by default.

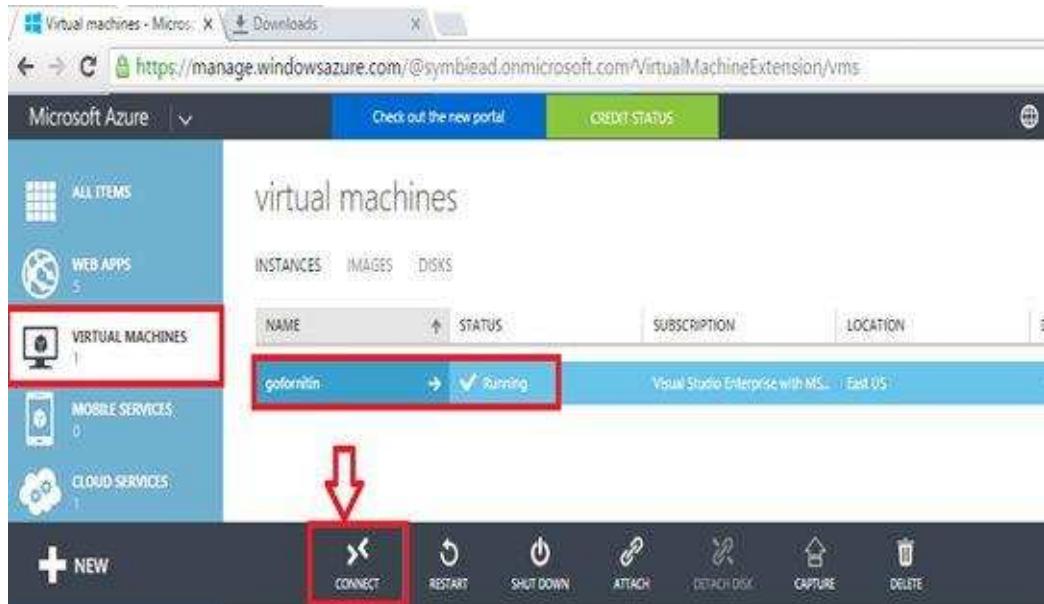
Click on to Next.



Here you can see, your virtual machine is creating, which is “**gofornitin**”.

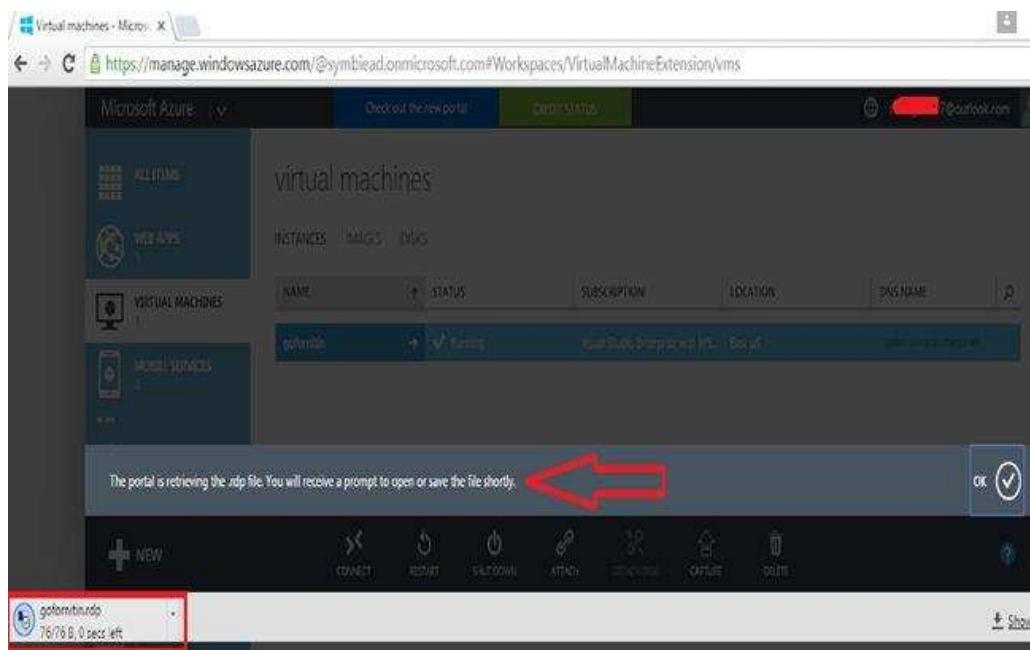


For using the virtual machine, click on “CONNECT” button.



So, here you can see portal retrieving a file with .rdp extension.

Click on file to open it.



Click “Connect” button.

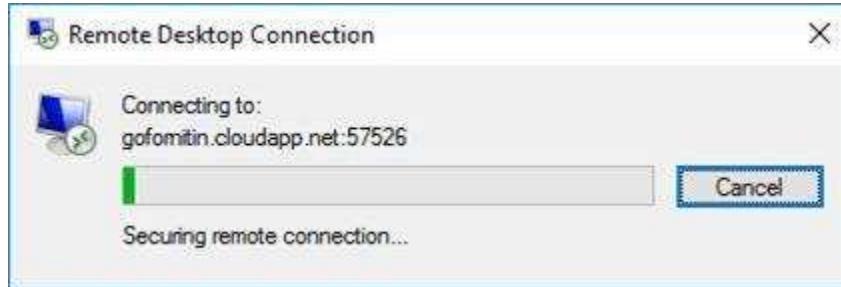


Enter your user name and password, my user name is “*nitinpandit*”.

Click on “OK” button.



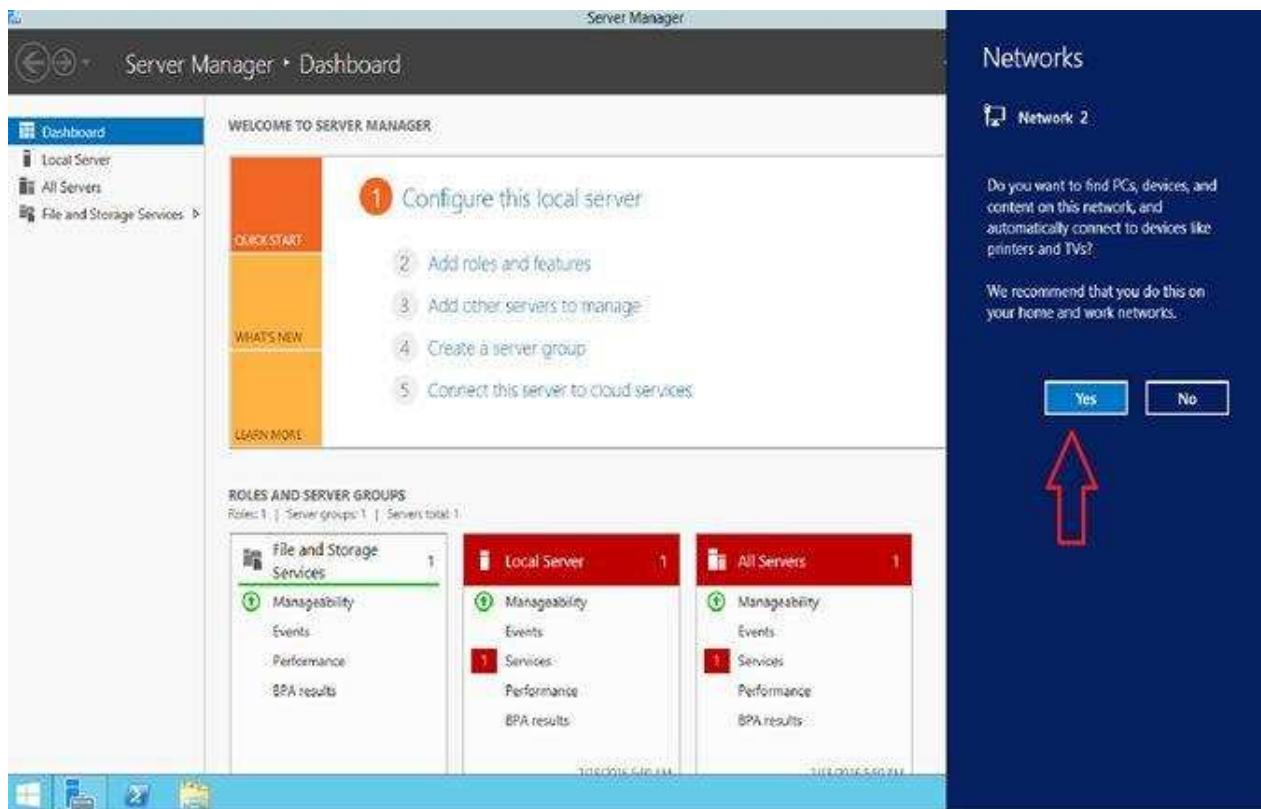
You can see I'm getting connecting to my virtual machine.



Click on “Yes” button.

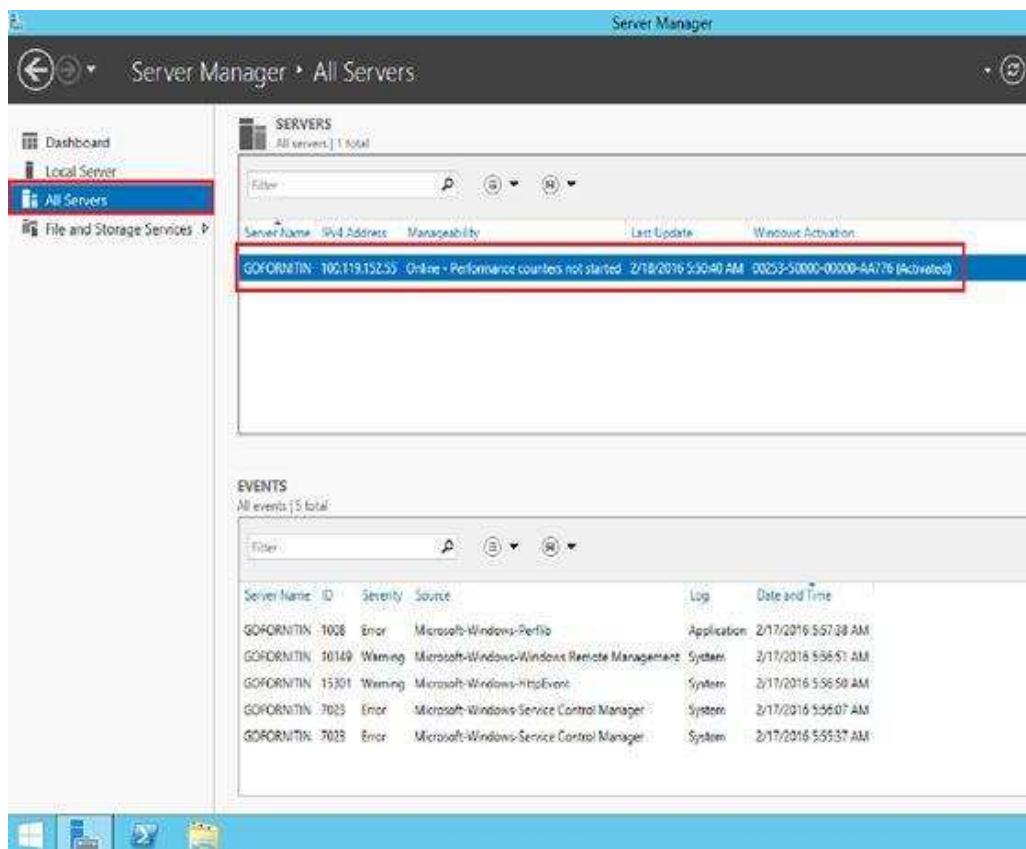


In the following screen shot, I'm successfully connected with my virtual machine; you can see **Dashboard of the Server Manager**.



Go to the “All Servers” and check the server name.

Click on to “All Servers”, you will find “**gofornitin**” server is available there.

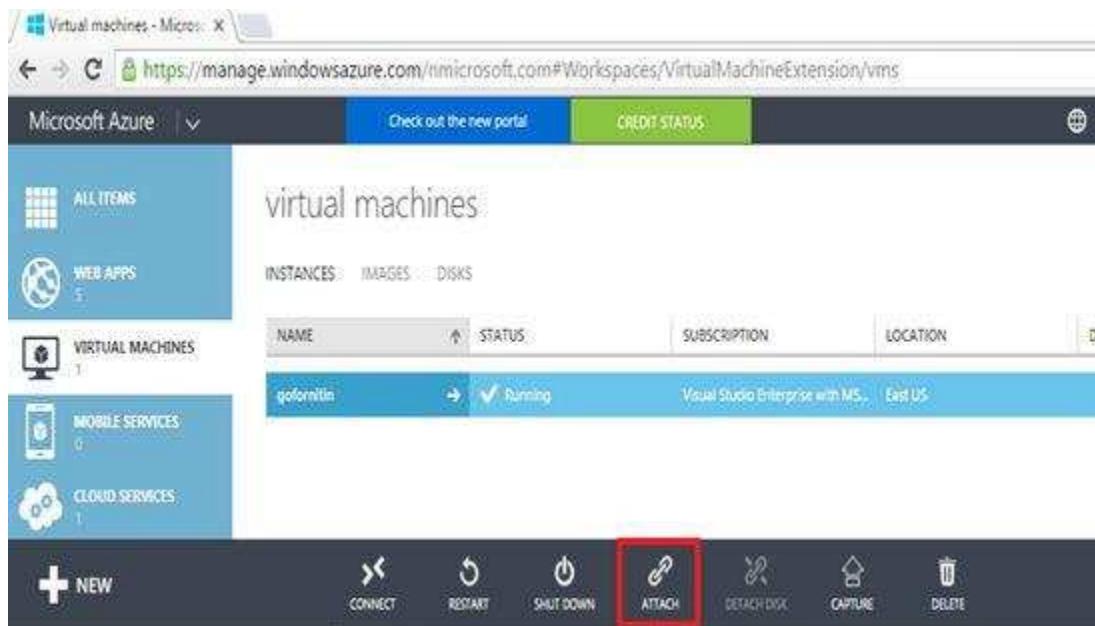


In the following screenshot, you can check your virtual machine's overall details, where you can see free space, host name and other related information of Virtual Machine.



Now, I'm going to add extra space in machine and an extra disk drive on it.

For this, select your appropriate Virtual Machine and click on “**ATTACH**” button.



It will ask to fill in the details for the empty disk.

It will take machine name (gofornitin), storage location, and file name by default.

Give the size for disk which you want to be created; here I have given it 15GB.

Click on the check button.

Attach an empty disk to the virtual machine

VIRTUAL MACHINE NAME
gofornitin

STORAGE LOCATION
<https://z0portalvhdsxmr60p51my7w.blob.core.windows.net>

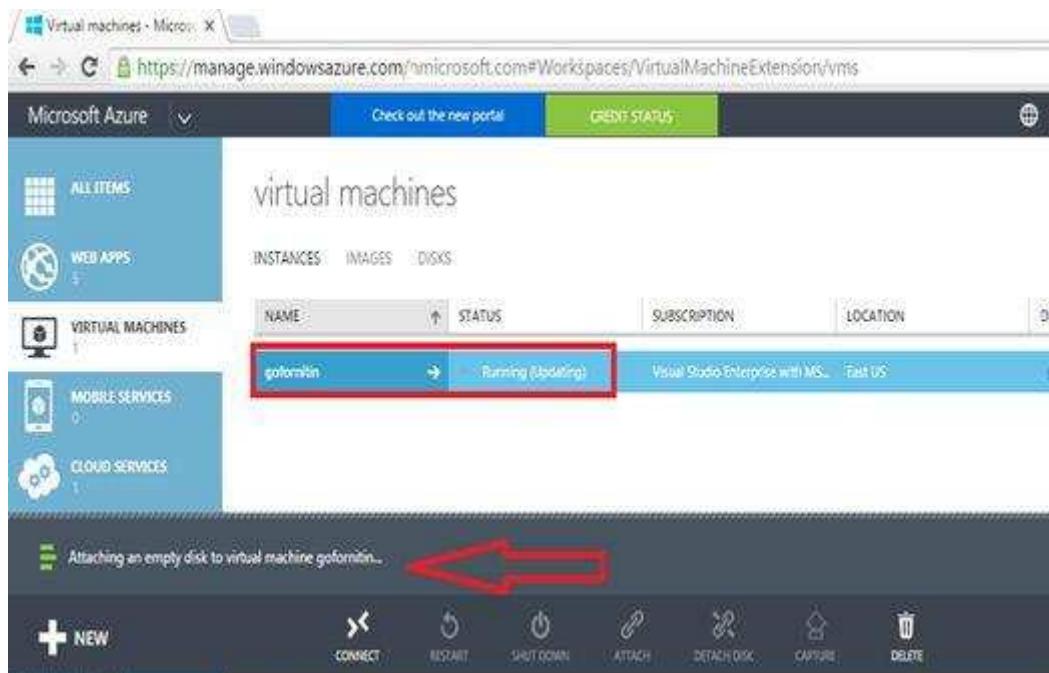
FILE NAME
gofornitin-gofornitin-0218-1

SIZE (GB)
15

HOST CACHE PREFERENCE
 NONE READ ONLY READ/WRITE 

It may take some time to create a new empty virtual machine.

Here you can see, our virtual machine is creating.



Virtual machines - Microsoft Azure

https://manage.windowsazure.com/#microsoft.com#Workspaces/VirtualMachineExtension/vms

Microsoft Azure Check out the new portal CLOUD STATUS

virtual machines

ALL ITEMS WEB APPS VIRTUAL MACHINES MOBILE SERVICES CLOUD SERVICES

INSTANCES IMAGES DISKS

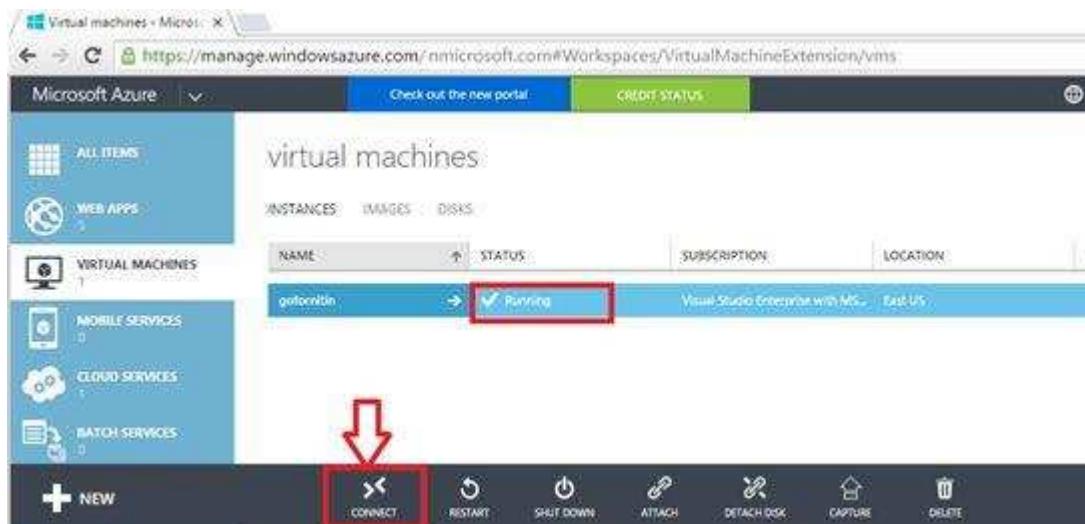
NAME	STATUS	SUBSCRIPTION	LOCATION
gofornitin	Running (Updating)	Visual Studio Enterprise with MS...	East US

Attaching an empty disk to virtual machine gofornitin...

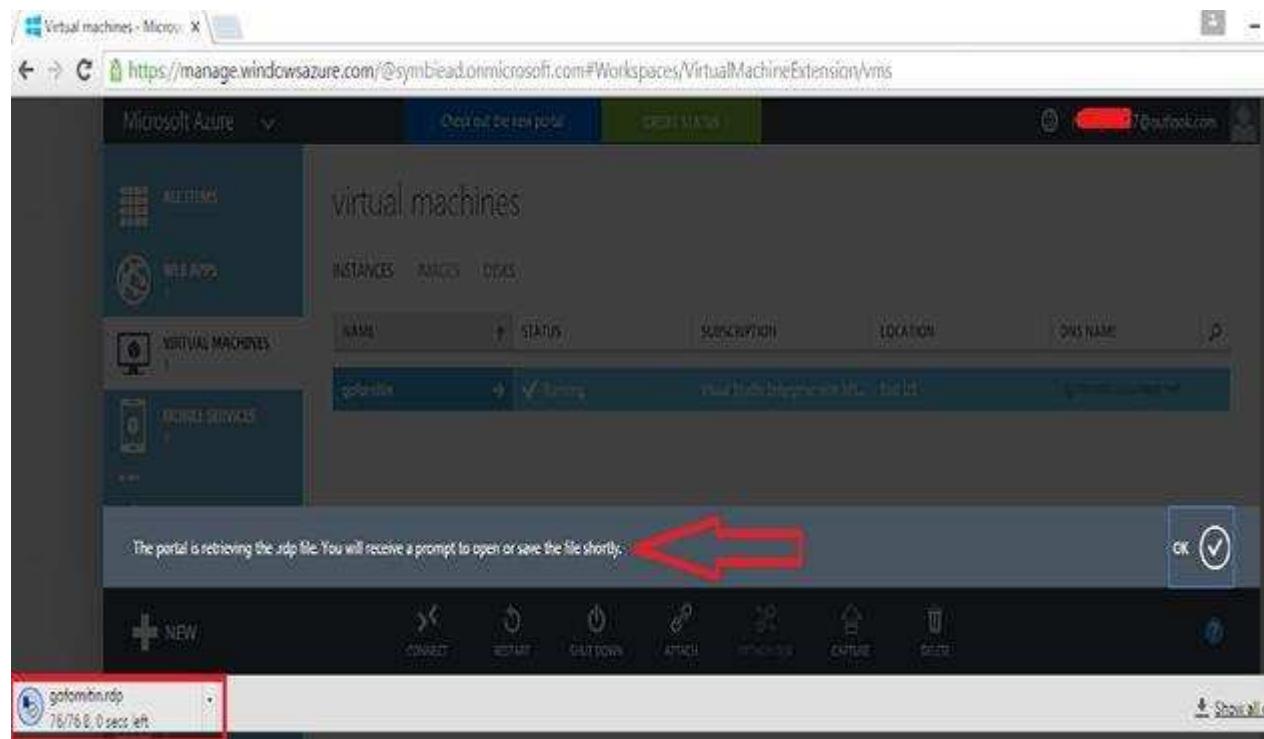
CONNECT RESTART SHUTDOWN ATTACH DETACH DISK CAPTURE DELETE NEW

So now, your virtual machine is ready for use and is in running mode.

Click on “**Connect**” button to retrieve the .rdp file.



Follow the last process, click on to downloaded .rdp file.

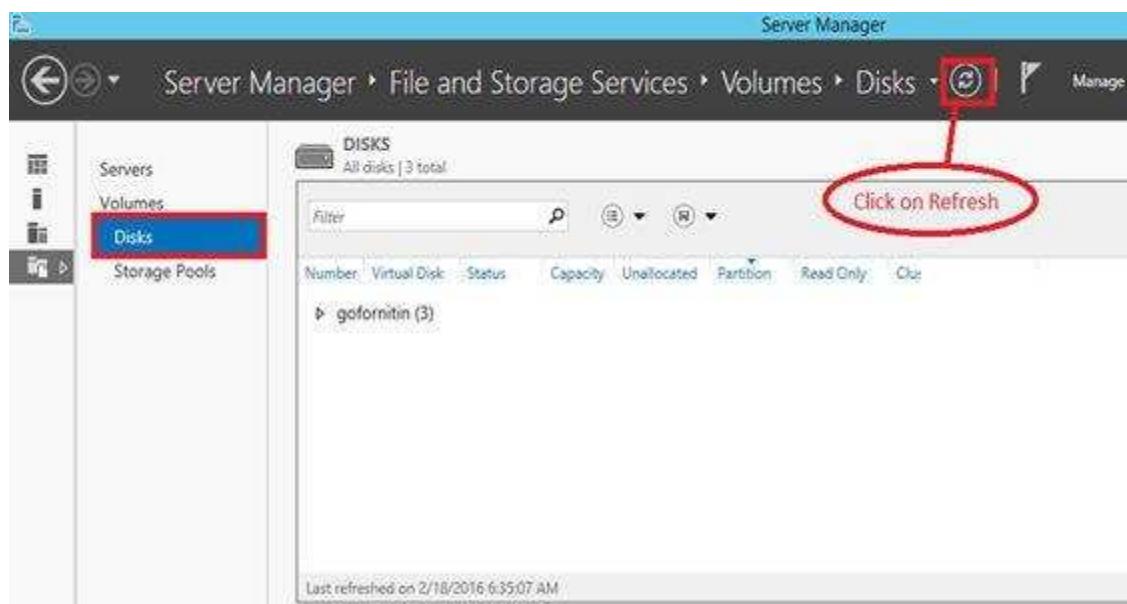


Now in the following screenshot, I'm connecting with the new Empty disk drive.

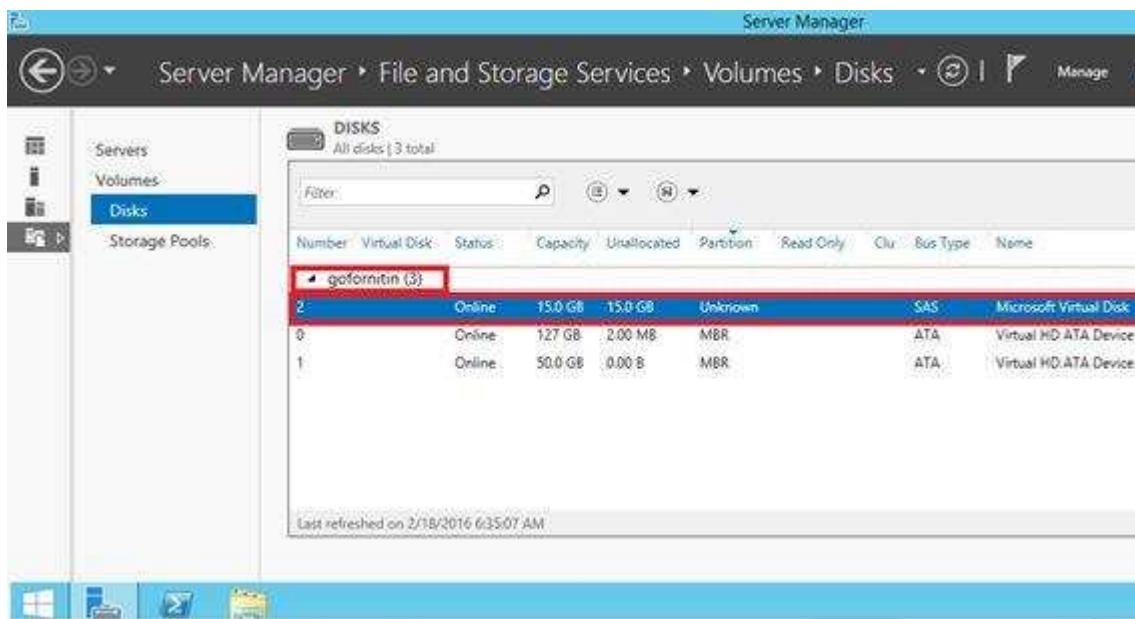
To check there is 15 GB disk in or not, simply go to the “File and Storage Services”.



Click on to “Disks” button and refresh the current page.

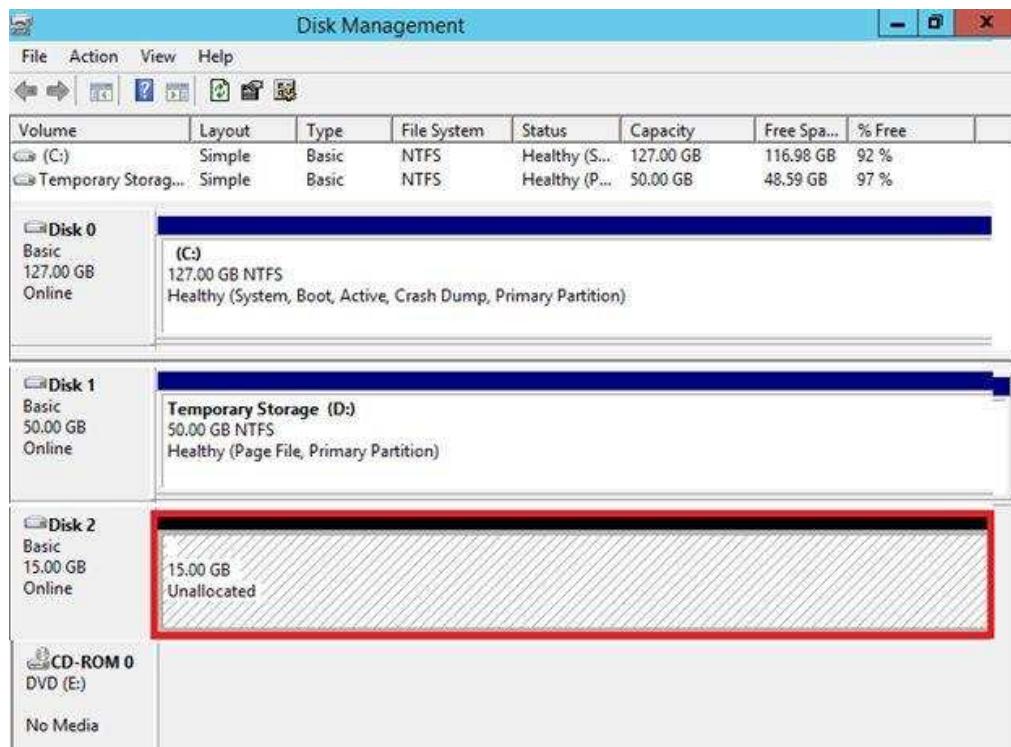


Click on to “gofornitin” and you can see our 15 GB virtual disk successfully added.

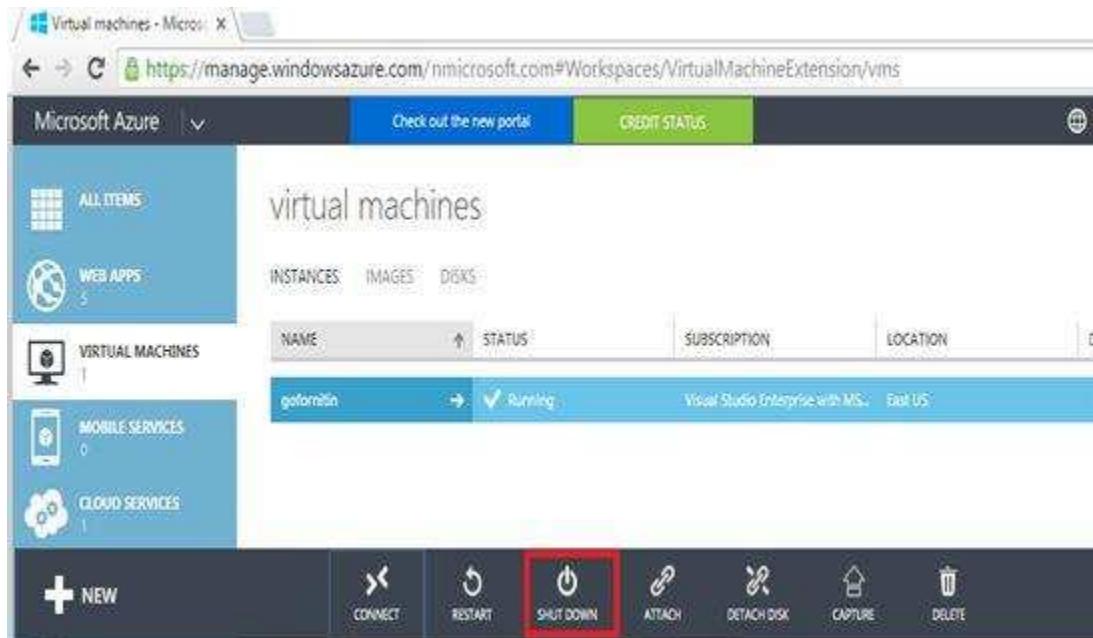


Let's check it to Disk Management.

There is also 15GB virtual disk available, and status of disk is online.



Make sure, click on “**Shut Down**” button to shut your virtual machine.



Use of SQL Azure in Visual Studio 2015 Step By Step Guide

In this we have to learn all about SQL Azure, it's very simple to use and also a powerful tool provided by Microsoft.



Microsoft made a revolution in Cloud History by Microsoft Azure. Microsoft does a lot of changes in Azure day by day providing best possibility for every user or developer. In Azure, we have lot of support for database such as NoSQL, SQL Server and more. In Microsoft Azure we can use SQL database by the following two ways:

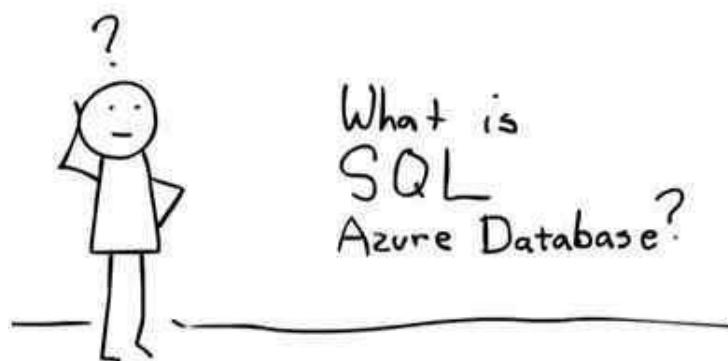
1. Microsoft SQL Server

We can use Microsoft SQL server in Microsoft Azure Virtual Machines (Azure VM) which is used as **IAAS (Infrastructure as a Service)** where we have full access by RDA (Remote Desktop Access) on this VM where we can use any Microsoft SQL Server.

2. SQL Azure

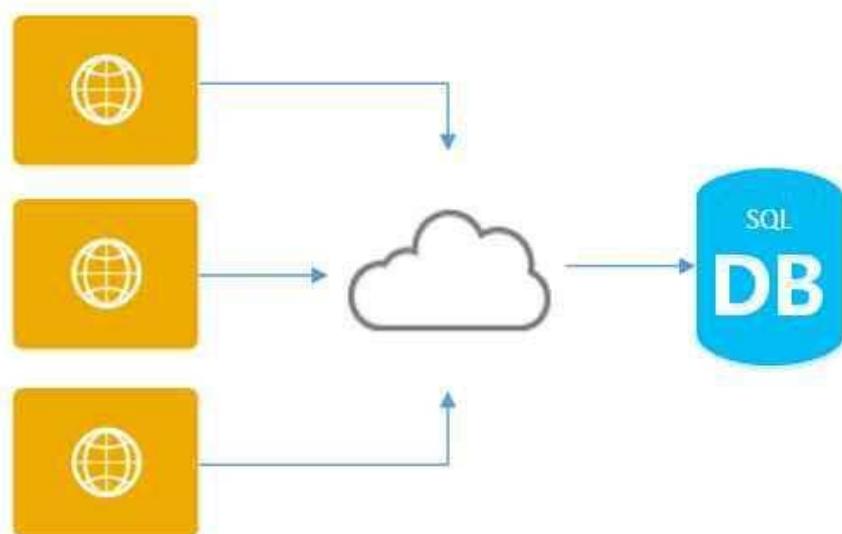
Sql Azure is the second option to use Sql Database on cloud which behaves like **PAAS** and it's very simple or easy to access by web portal of Microsoft Azure or in Visual Studio Azure SDK.

3.



What is SQL Azure Database?

In short, SQL Azure database is simply a way to get connected in Cloud Services where we can store our database into Cloud. Microsoft SQL Services and Microsoft SQL Data Services are now known as Microsoft SQL Azure and SQL Azure Database. Microsoft Azure is the best way to use **PAAS (Platform as a Service)** where we can host multiple databases on the same Account.



Microsoft SQL Azure has the same feature of SQL Server, i.e. high availability, scalability and security in the core.

Built-in Automatic Backup in Microsoft Azure SQL Database

Microsoft Azure SQL Database have a best feature, it automatically creates backups of every active database. Every hour a backup is taken and geo-replicated to enable the 1 hour recovery point objective (RPO) for Geo-Restore. Additionally, transaction log backups are taken every 5 minutes to enable Point in Time Restore.

Backup Storage

As per MSDN:

"Backup storage is the storage associated with your automated database backups that are used for Point in Time Restore and Geo-Restore. Azure SQL Database provides up to 200% of your maximum provisioned database storage of backup storage at no additional cost. For example, if you have a database in the Standard service tier with a provisioned size of 250 GB, you will be provided with 500 GB of backup storage at no additional charge. If your database exceeds the provided backup storage, you can choose to reduce the retention period by contacting Azure Support or pay for the extra backup storage billed at standard Read-Access Geographically Redundant Storage (RA-GRS) rate. For more information on RA-GRS billing, see Storage Pricing Details."

Now let's see how to create SQL Azure Step by Step

In your Azure Account Portal you have a link to use your Sql Databases on SQL Azure, so go to the Link.

Login now and go to portal Link where you can see Sql Database,



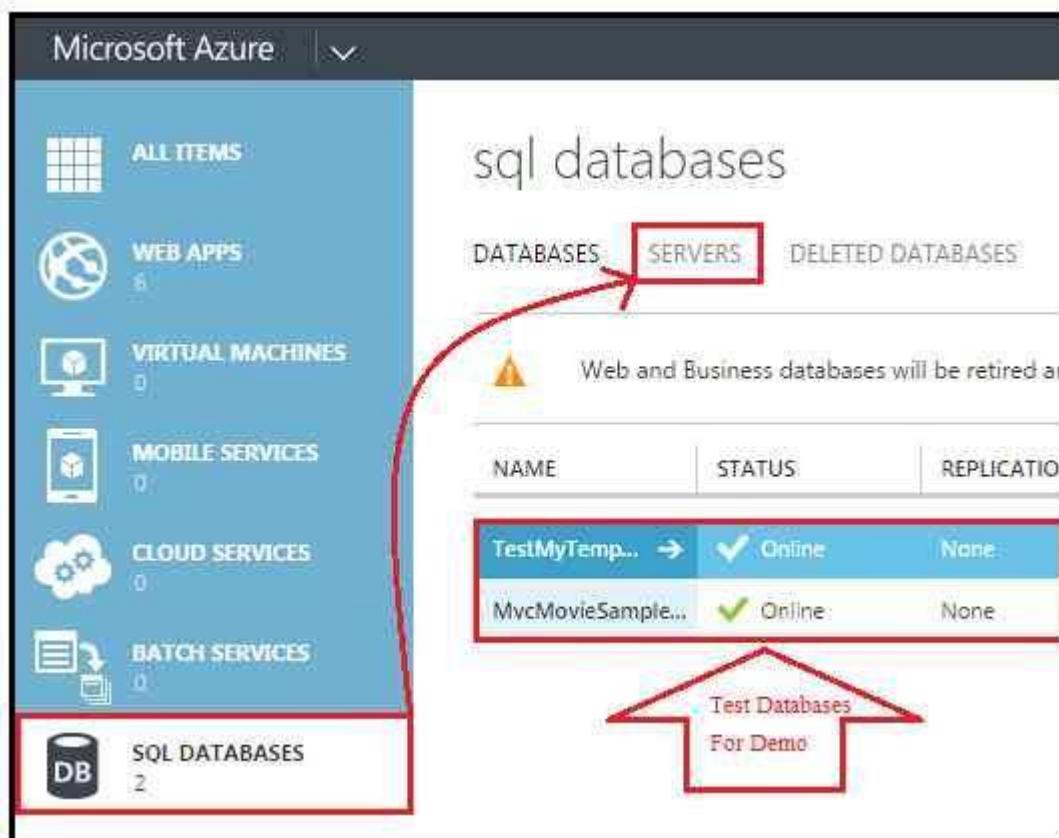
If you click on this section, you'll find all the already created databases and will get the link to go to servers, every database exist in any physical server, but for SQL Azure we don't need any hardware or software to use that because it's a **PAAS** on Cloud Services provided by Microsoft Azure but still you have to create a server where you want to use your datacenter or replica of your database. So first we want to create a server, now let's see how to create Data Server, but one thing you should remember, we don't have RDA (Remote Desktop Access) permission on this server we can use this database in our Visual Studio or in our Microsoft SQL Server Management Studio so that we can create table and can execute our .sql script file for database management. So finally in this article we'll learn all the things step by step which is divided into the following seven parts:

1. How to Create Servers in Microsoft Azure.
2. How to Configure Server for Client access by IP Addresses.
3. How to create Database on specified Server.
4. How to get the connection strings.
5. How to Use SQL Azure in Visual Studio.
6. How to Use SQL Azure Table for CRUD Operations in ADO.Net.
7. How to Use SQL Azure Table for CRUD Operations by Entity Framework.

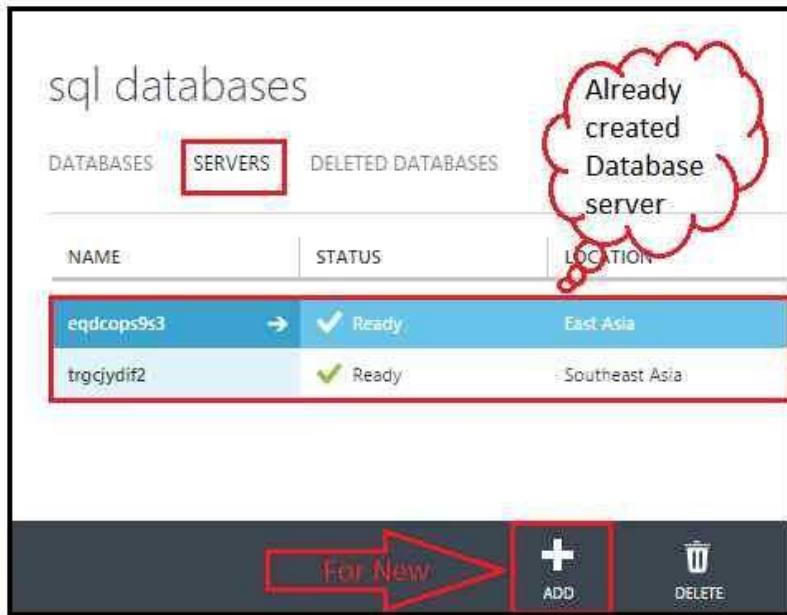
So I hope you'll enjoy this, now it's time to do everything by yourself, firstly login into your Microsoft Azure Account.

Now let's see How to Create Servers in Microsoft Azure

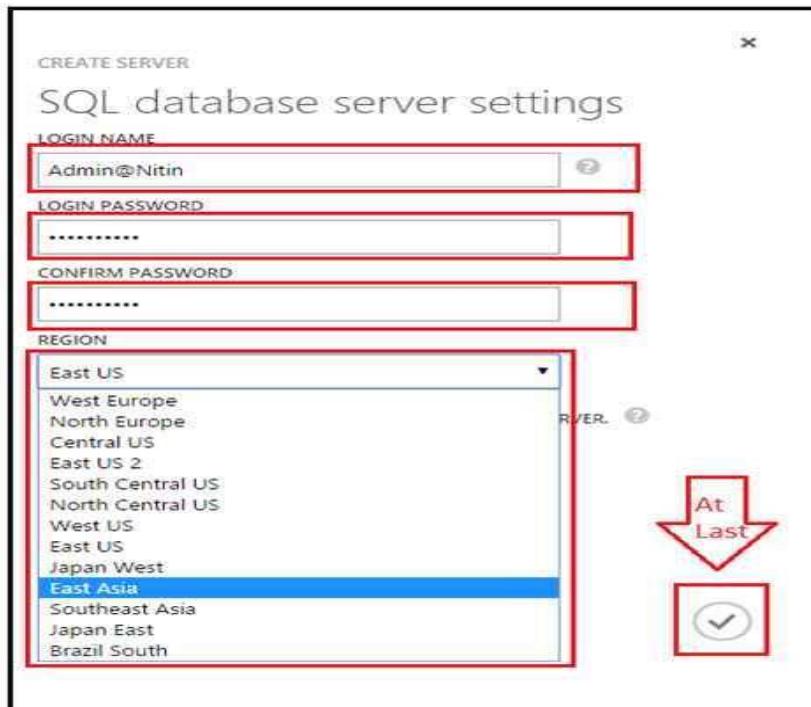
We need a server first to create any Sql DB to host on SQL Azure, so click on SQL Databases link in main menu and get the entire database already created and on the top there is a link. Servers find that link as I have shown in the following image:



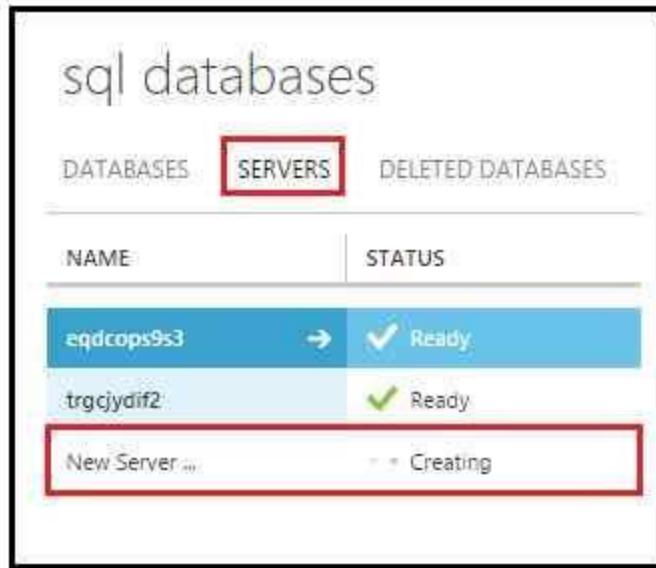
After clicking on Servers, you'll get the entire server which already running on the cloud service. Now the next step is find the button Add to add or create a new Server as in the following image.



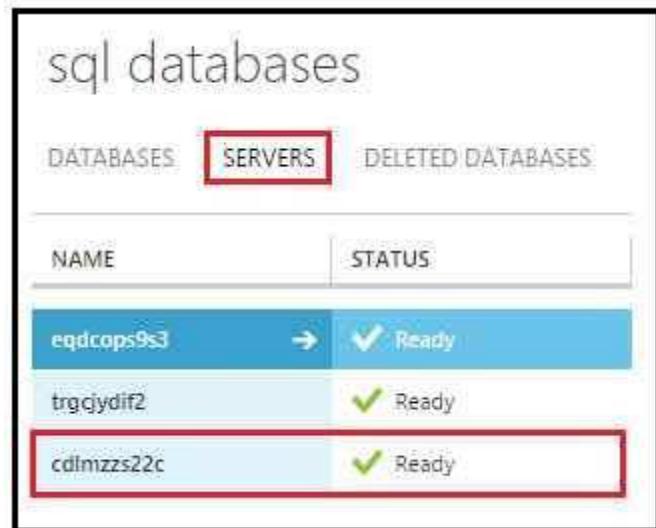
Now fill this form to set all the necessary details for a server like Login Name, Password, Confirm Password and region for selection of Datacenter and after filling all the information press enter to submit all the details and then that will create a server by a logical server name. It will be a combination of alphabetic and numeric digits which we have to use to create a database.



After submitting all the information about our server it will take a few seconds to create it.

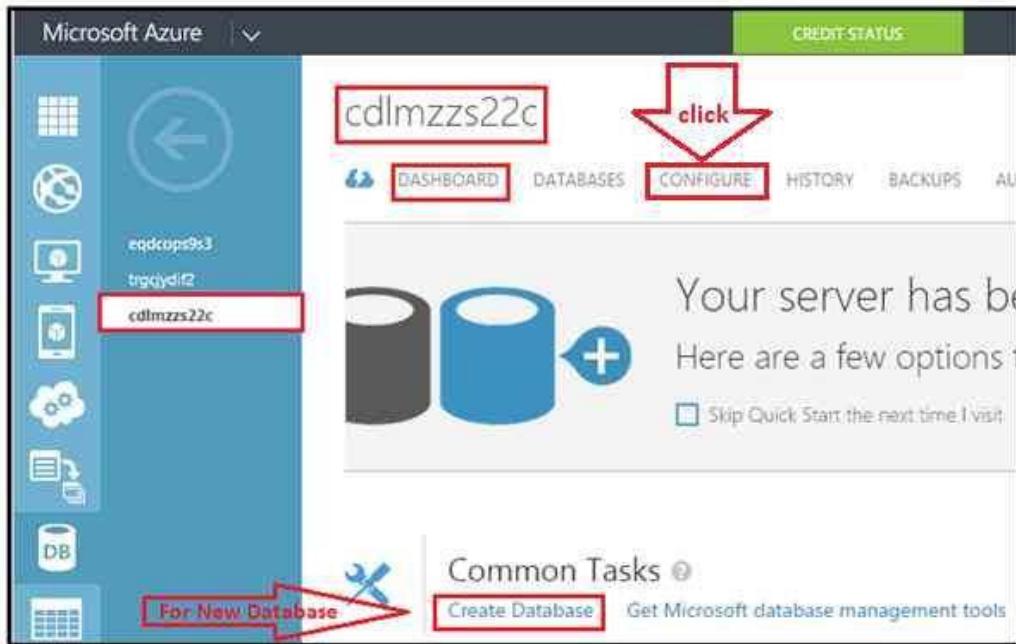


After that now server is ready to use with a logical name but if you use this server to host your database, you don't have permission to use on any client IP, so we need to configure this server for specified client IP addresses and go to the next step.

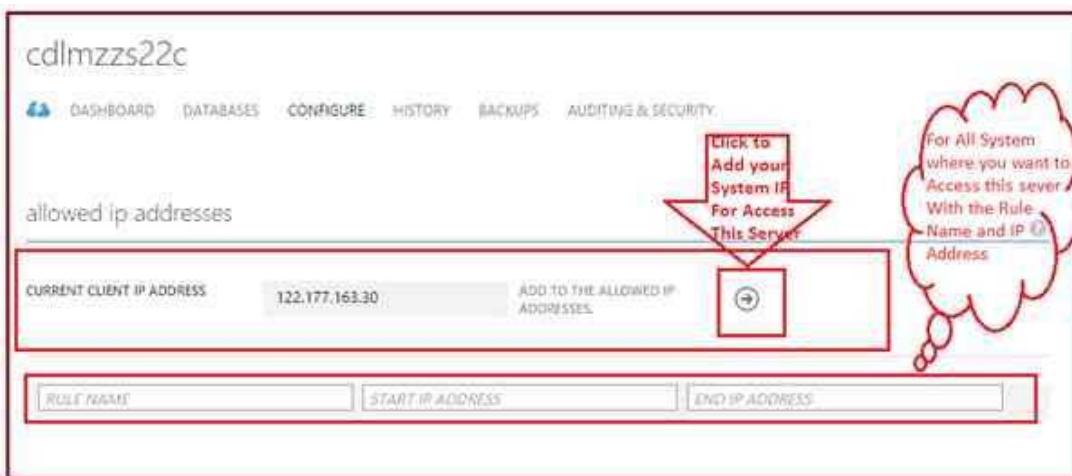


Now let's see How to Configure Server for Client access by IP Addresses-

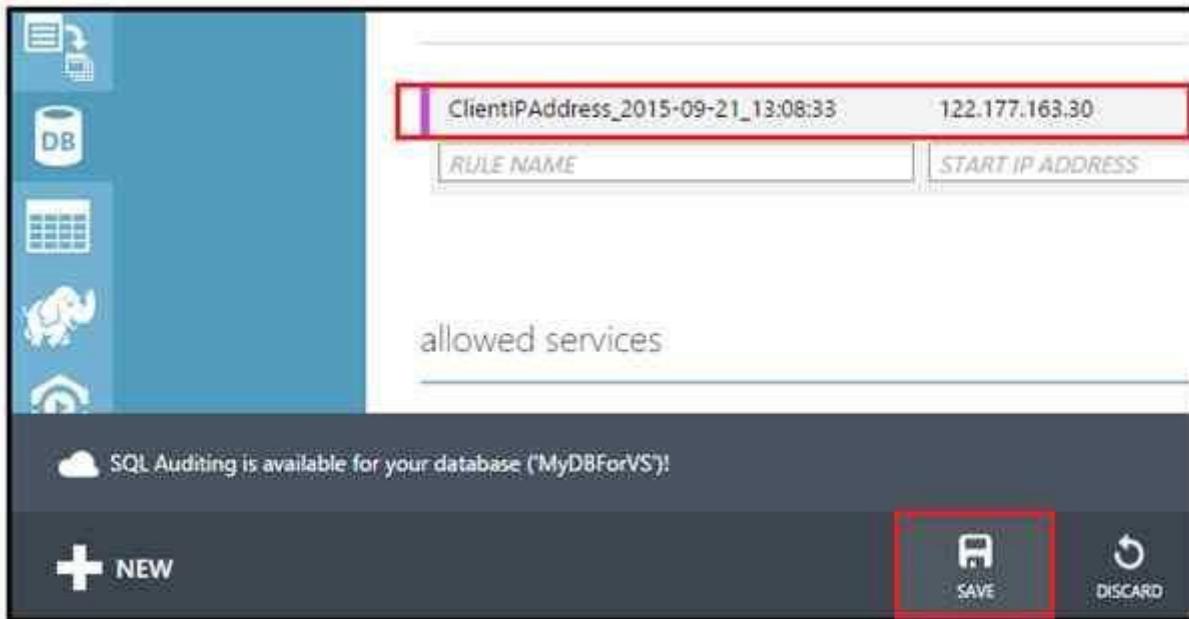
Every server is a must be configure for every client system IP address, so let's see, first click on the server logical name to open the server dashboard or configuration sections. Now open the configuration section, so click as per the following screenshot:



In this section there is an option to add details for every client where you need to login this server. Add all IP addresses with the rule name and you want to access the same system, so there is directly a link to add current system in the list as in the following image.



Now your system and all those systems are able to get connected with this server and those IP address is added in the list.



At last, just save the setting which have changed before leaving this section or creating new database.

Now let's see How to create Database on specified Server

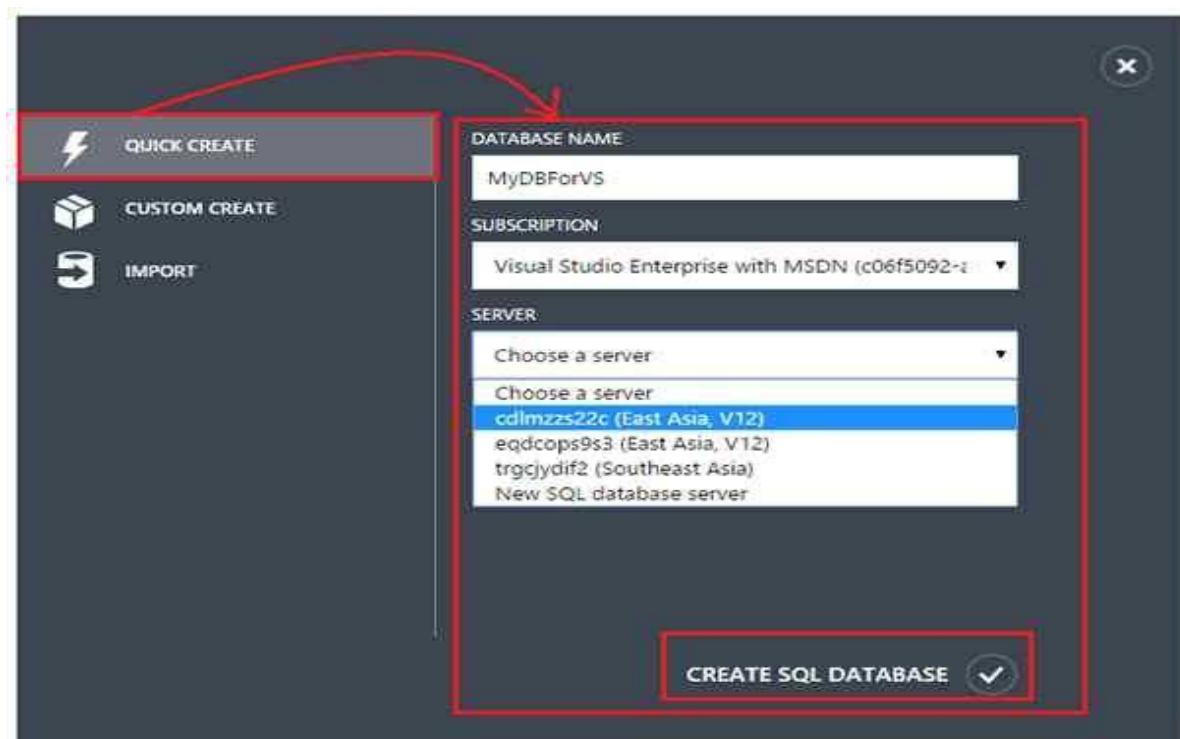
Now the time is to create a SQL Azure Database on cloud as PaaS, so now find the + New Button on the left side bottom in your Web portal of Azure and click that.



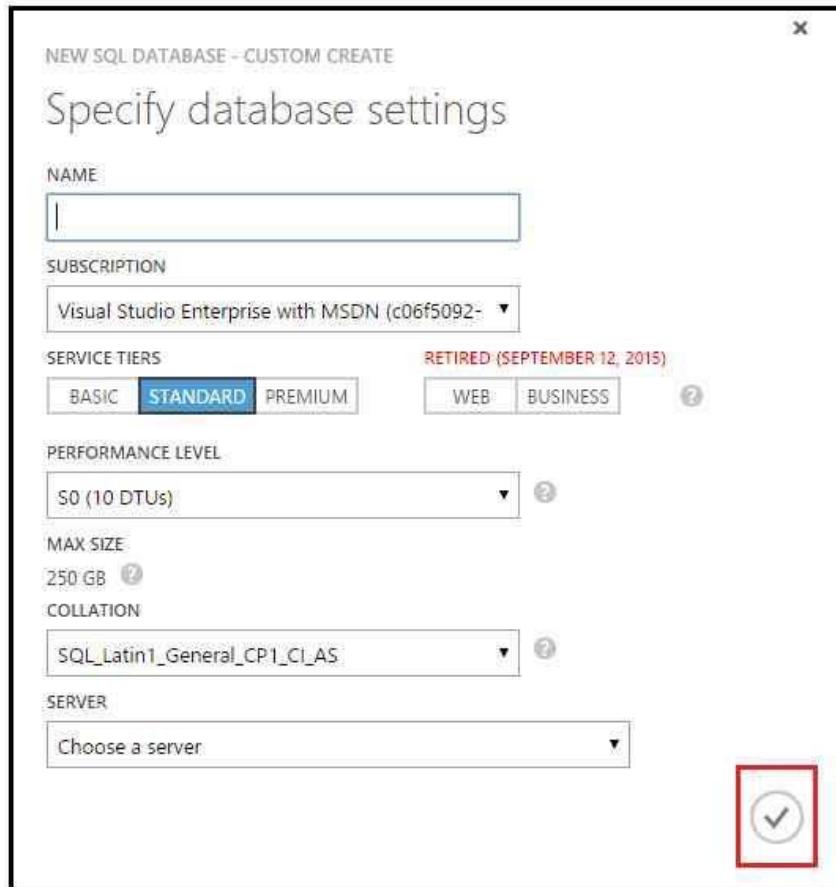
Now go to Data Services, then SQL Databases and Quick Create or Custom Create.



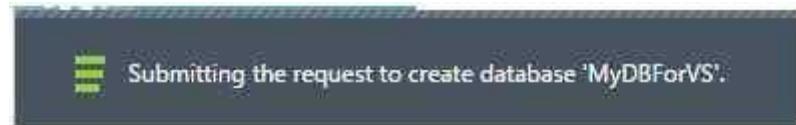
If you select Quick create, you have to enter only the name of database and select the server by the logical name of the server and click “**Create SQL Database**” button.



If you select Custom Create, you have many options to fill or select Database Name, Subscription Type, Max Size and Server Name and then click ok.



After submitting all the details it takes few seconds to be ready for use and you can see the progress in the Progress bar.

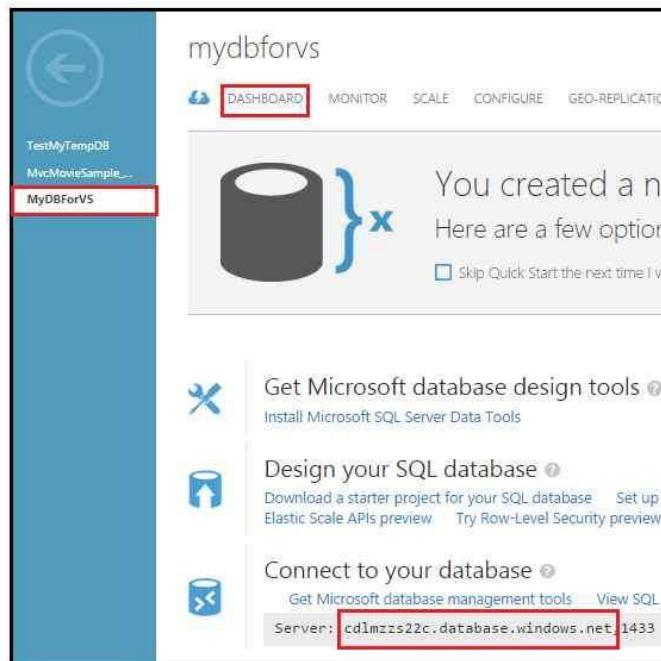


Now your server is ready to use from your Visual Studio or SQL Server Management Studio.



Now let's see How to get connection Strings

Now the most important part is if you want to connect this database to store your data as per need in any Application by any framework or technologies like in php or java and more. So the first thing which we need is Connections String to connect, so let's see how to get the connection string. Firstly, go to the Dashboard of the database.



In the Dashboard section there is an option in quick glance “**Show Connection Strings**” and below that there is the server name by which you have to connect in Visual Studio Server Explorer. So click on the link of Connection Strings.



Now you will see the window of connection string where we have all maximum connection string for every developer for every technology **ADO.Net**, **JDBC**, **PHP** and **ODBC**, so you can use it anywhere just add password in the connection string.

Connection Strings

ADO.NET:

```
Server=tcp:cdlmzzs22c.database.windows.net,1433;Database=MyDBForVS;User ID=Admin@Nitin@cdlmzzs22c;Password={your_password_here};Trusted_Connection=False;Encrypt=True;Connection Timeout=30;
```

ODBC:

```
Driver={SQL Server Native Client 10.0};Server=tcp:cdlmzzs22c.database.windows.net,1433;Database=MyDBForVS;Uid=Admin@Nitin@cdlmzzs22c;Pwd={your_password_here};Encrypt=yes;Connection Timeout=30;
```

PHP:

```
Server: cdlmzzs22c.database.windows.net,1433 \r\nSQL  
Database: MyDBForVS\r\nUser Name: Admin@Nitin\r\n\r\nPHP  
Data Objects(PDO) Sample Code:\r\ntry {$conn = new  
PDO ('sqlsrv:server =  
tcp:cdlmzzs22c.database.windows.net,1433; Database =  
MyDBForVS' , "Admin@Nitin\\","{your_password_here}");}in
```

JDBC:

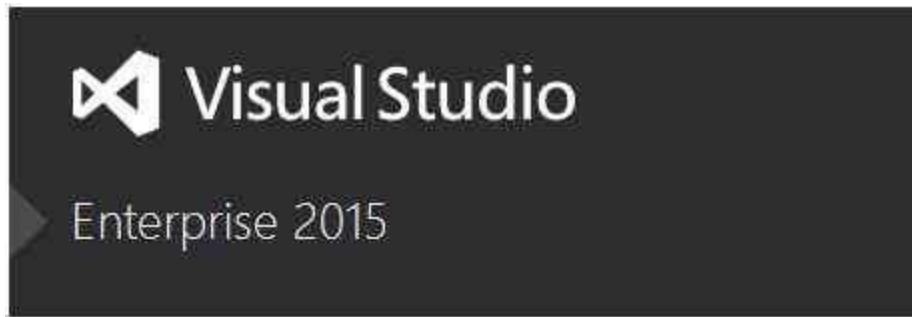
```
jdbc:sqlserver://cdlmzzs22c.database.windows.net:1433;database=MyDBForVS;user=Admin@Nitin@cdlmzzs22c;password={your_password_here};encrypt=true;hostNameInCertificate=*.database.windows.net;loginTimeout=30;
```

 Allow the connection in [firewall rules](#) 

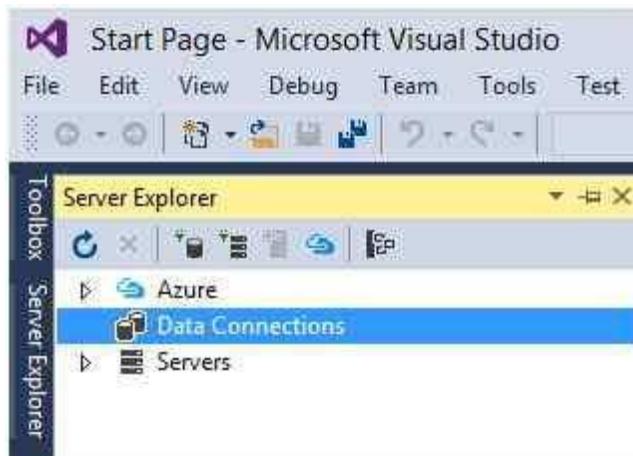


Now let's see How to Use SQL Azure in Visual Studio

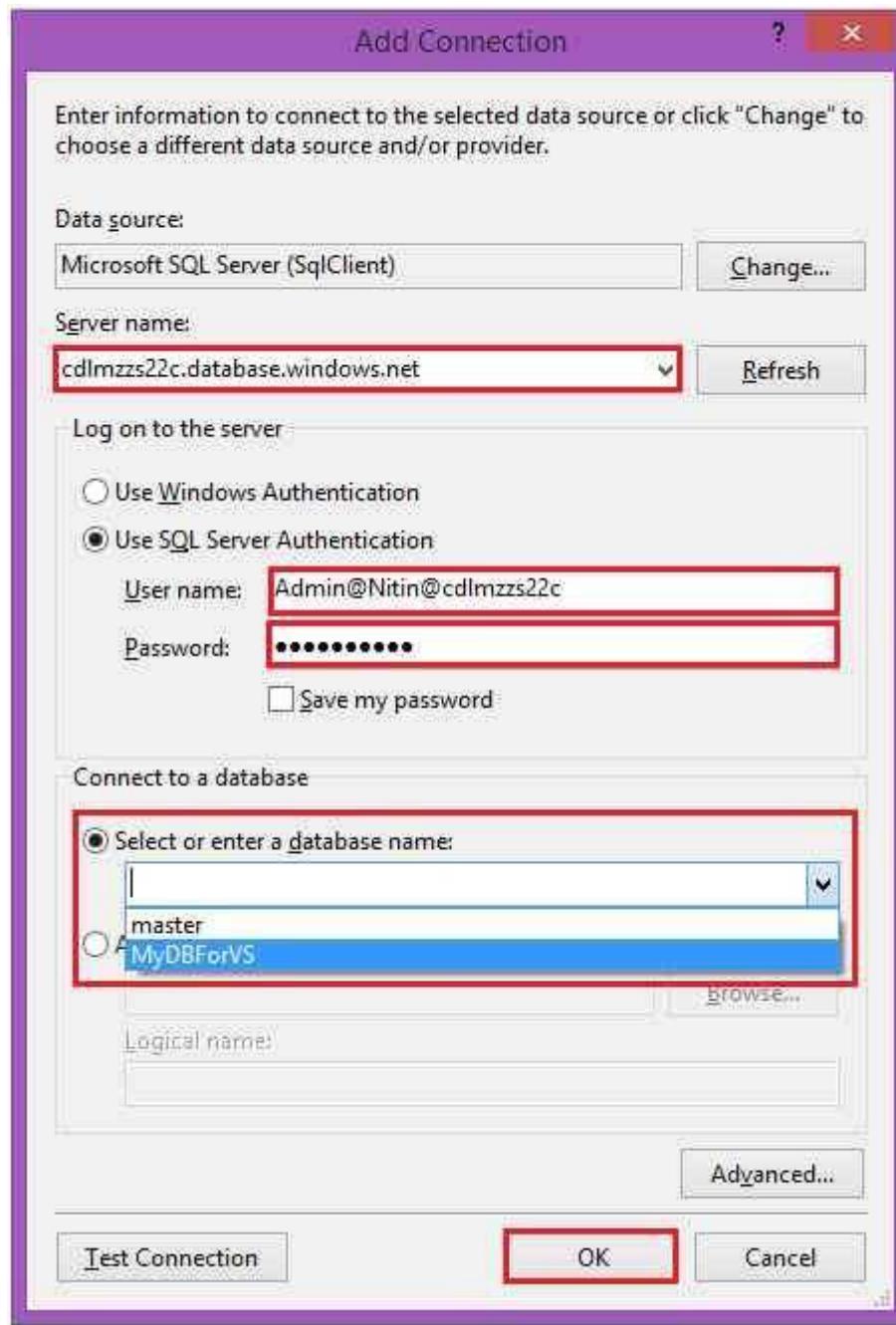
Now let's see how to use SQL DB to your Visual Studio Server Explorer or may be in SQL Server Management Studio. Open your visual studio first.



Open Server Explorer to connect SQL Azure Database.



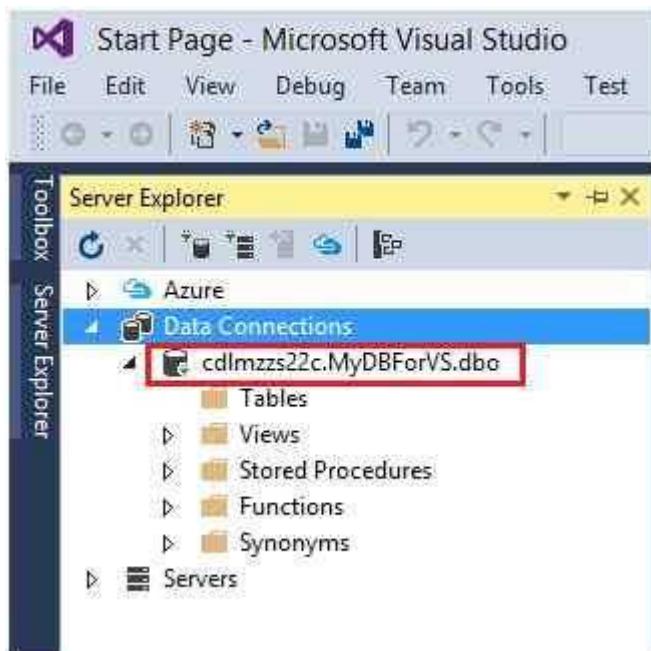
Right click on *Data Connections* and add Connections, then there is a window, fill all the details and copy the Server Name from Database Dashboard in Azure web portal and paste in this window. Now enter the user name which was entered by you at the time of server creation like "**ServerUserLoginName@ServerLogicalName**" and password. Select your database name from the list and click ok or Test Connection button if you want to check your connection is valid or not.



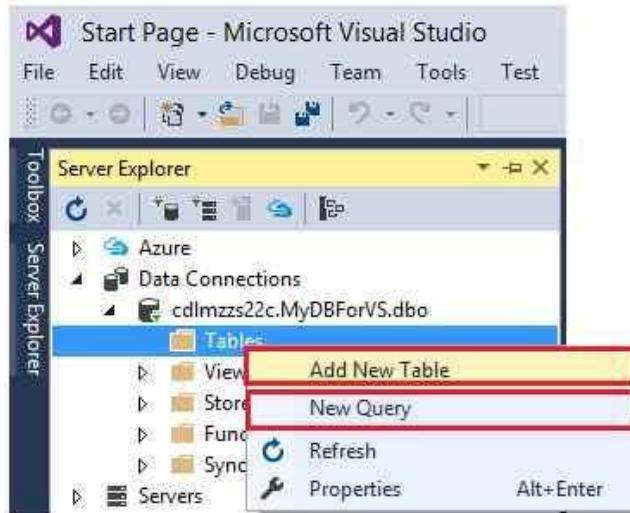
This is the same process for Microsoft SQL Server Management Studio, the people who don't have Visual Studio like DBA, they can use there .sql script file over there.



So let's see with Visual Studio I connect my Azure Database server in the Server Explorer.



Now if you want to add a new table right clicks on Tables and find two options Add new Table and New Query. Select any one.



If you want to create table by query, select New Query or if I want to add by wizard, click on Add new Table and then you get a window on your screen where you have to add columns and add constraint if required. After writing your query or adding columns, just click on Update Button. It will confirm and save the schema.

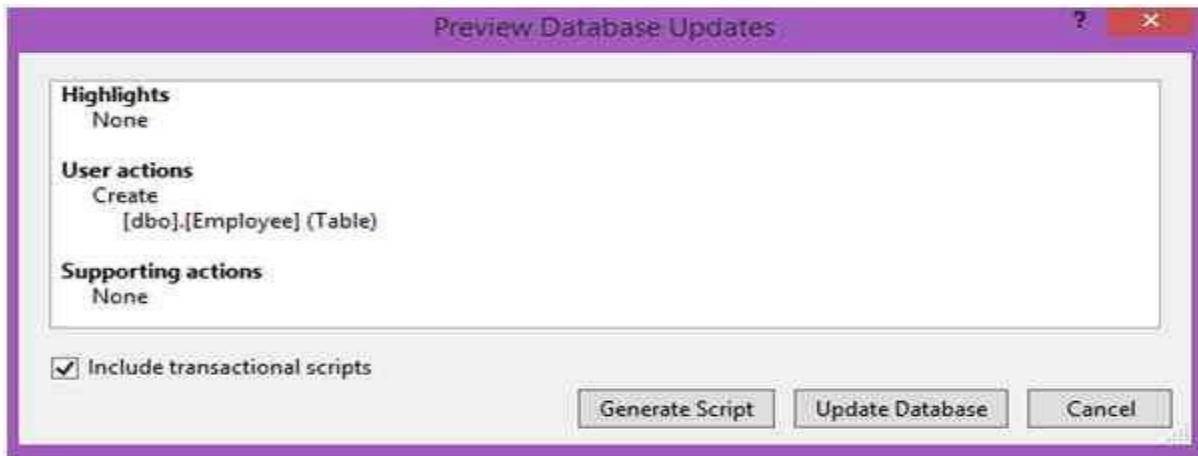
Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Name	varchar(50)	<input checked="" type="checkbox"/>
Salary	bigint	<input checked="" type="checkbox"/>
City	varchar(50)	<input checked="" type="checkbox"/>

```

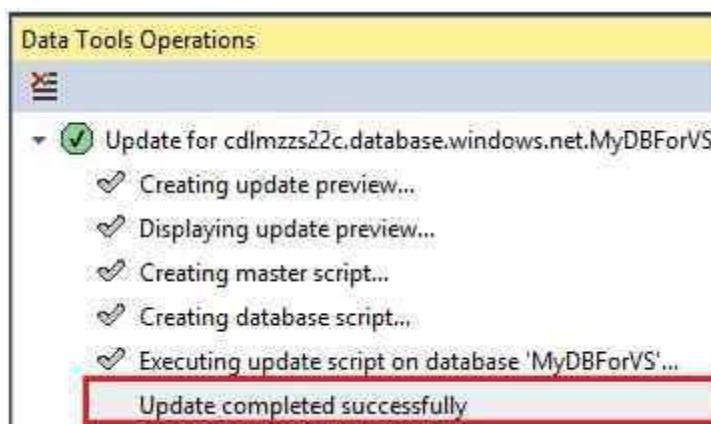
CREATE TABLE Employee
(
    [Id] INT NOT NULL PRIMARY KEY,
    Name varchar(50),
    Salary bigint,
    City Varchar(50)
)

```

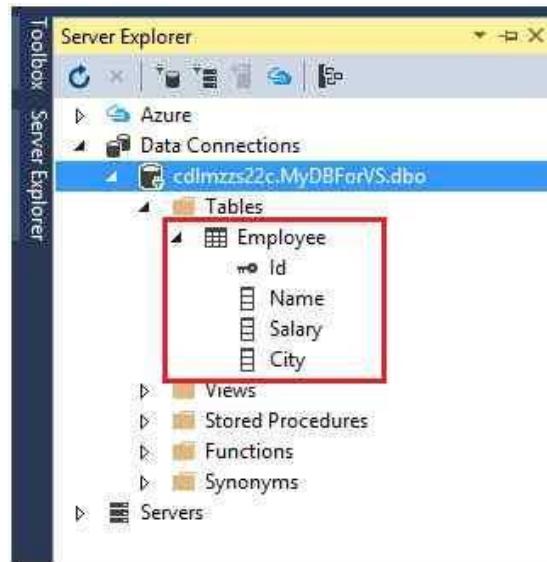
In this window we have two options: Click Generate Script if you want to create Script file and it will save automatically in the solution or any specified address. The second option is to Update Database if you want to update all schemas without saving query file, it will update SQL Azure Database with new Schema.



You can see the update of progress in the Data Tools Operations.



Now open server explorer and refresh connection, now you can see your table in Tables options.

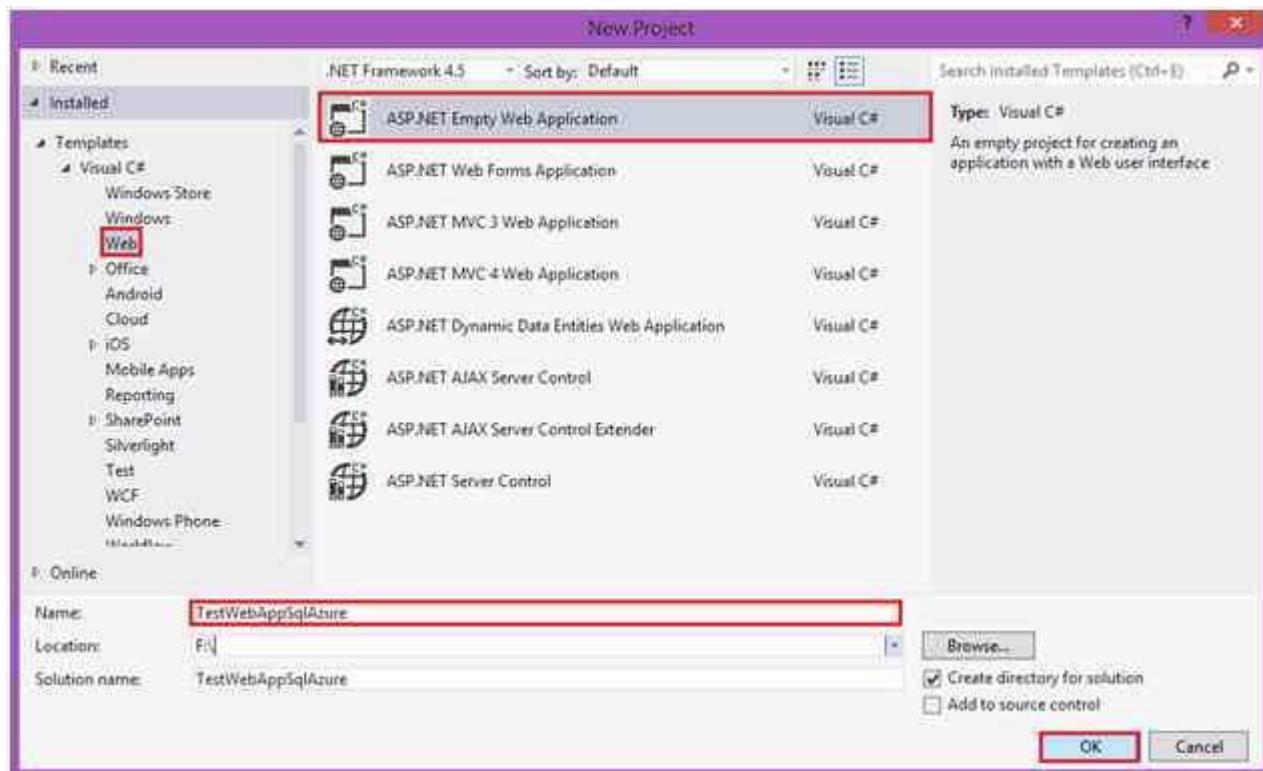


Now use this table anywhere with the connection string that we get by our Microsoft Azure portal.

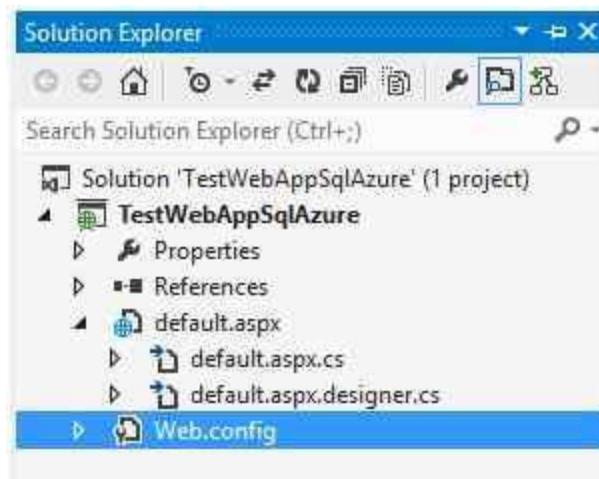
Now let's see How to Use SQL Azure Table for CRUD Operations in ADO.NET-

If you want to connect this use your SQL Azure table in any application such as Windows Form, WPF Application, Console Application or any web application. This is the same process so I am using a web application project in Visual Studio. Open Visual Studio, File, New Project, then select Web and click ASP.NET Empty Web Application.

Give a name and location and press OK.



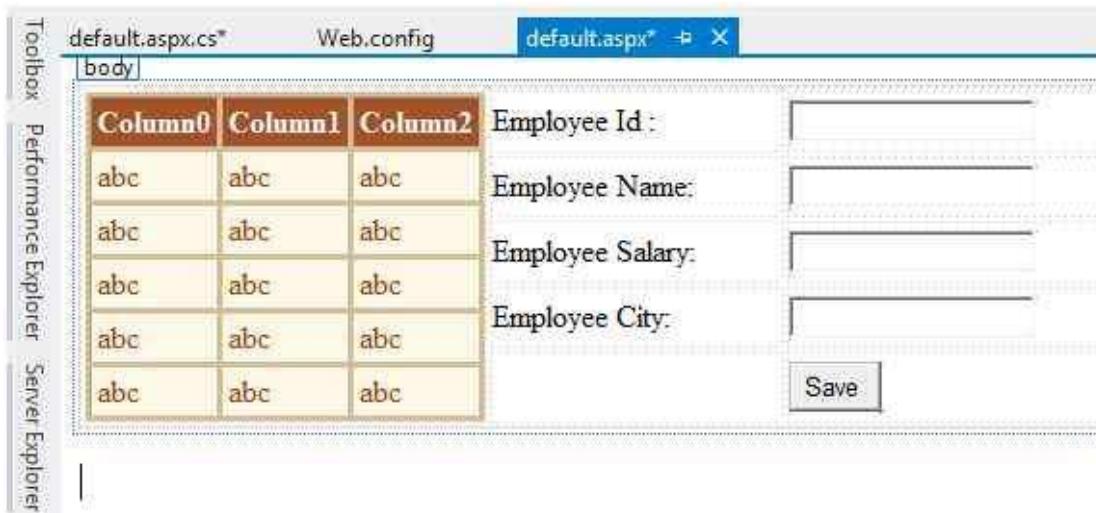
Now I have an empty Web Application Solution, so add any page to show the data or use CRUD operations in SQL Azure Table. I just added a page default.aspx to do that.



Now add a new Connection String into Connections String Sections in your web.config, copy connection string from Azure Web portal and paste with password in Web.config like the following image:

```
<connectionStrings>
<add name="MyDB"
      connectionString="Server=tcp:cdlmzzs22c.database.windows.net,1433;Database=MyDBForVS;
      User ID=Admin@Nitin@cdlmzzs22c;Password=***** ;
      Trusted_Connection=False;Encrypt=True;Connection Timeout=30;"/>
</connectionStrings>
```

Use some control like **GridView** and other control like **TextBox** and Button event.



The screenshot shows the Visual Studio IDE interface with three tabs: 'default.aspx.cs', 'Web.config', and 'default.aspx'. The 'default.aspx' tab is active, displaying an ASPX page. On the left, the 'Toolbox' is visible. The main content area contains a 'GridView' with three columns labeled 'Column0', 'Column1', and 'Column2', each containing the value 'abc'. To the right of the grid, there is a form with four text input fields labeled 'Employee Id:', 'Employee Name:', 'Employee Salary:', and 'Employee City:', each with a corresponding text box. Below these fields is a 'Save' button.

I just use a simple code of ADO.NET with SqlClient provider, on the button I run command and if it runs successfully, I called a function where I bind the grid from my Azure SQL Client.

Code:

```
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public void BindGrid()
{
    SqlDataAdapter da = new SqlDataAdapter("Select * from Employee", ConfigurationManager
        ..ConnectionStrings["MyDB"].ConnectionString);
```

```

DataSet ds = new DataSet();
da.Fill(ds);
GridView1.DataSource = ds.Tables[0];
GridView1.DataBind();
}
public int RunCommand(string Query)
{
    SqlConnection con = new SqlConnection(ConfigurationManager
        .ConnectionStrings["MyDB"].ConnectionString);
    SqlCommand com = new SqlCommand(Query,con);
    con.Open();
    int res=com.ExecuteNonQuery();
    con.Close();
    return res;
}
protected void Button1_Click(object sender, EventArgs e)
{
    string qry = "Insert into Employee Values ('" + TextBox1.Text + "','"'
        + TextBox2.Text + "','"' + TextBox3.Text + "','"' + TextBox4.Text + "')";
    if (RunCommand(qry) > 0)
    {
        BindGrid();
    }
}

```

Here's the output

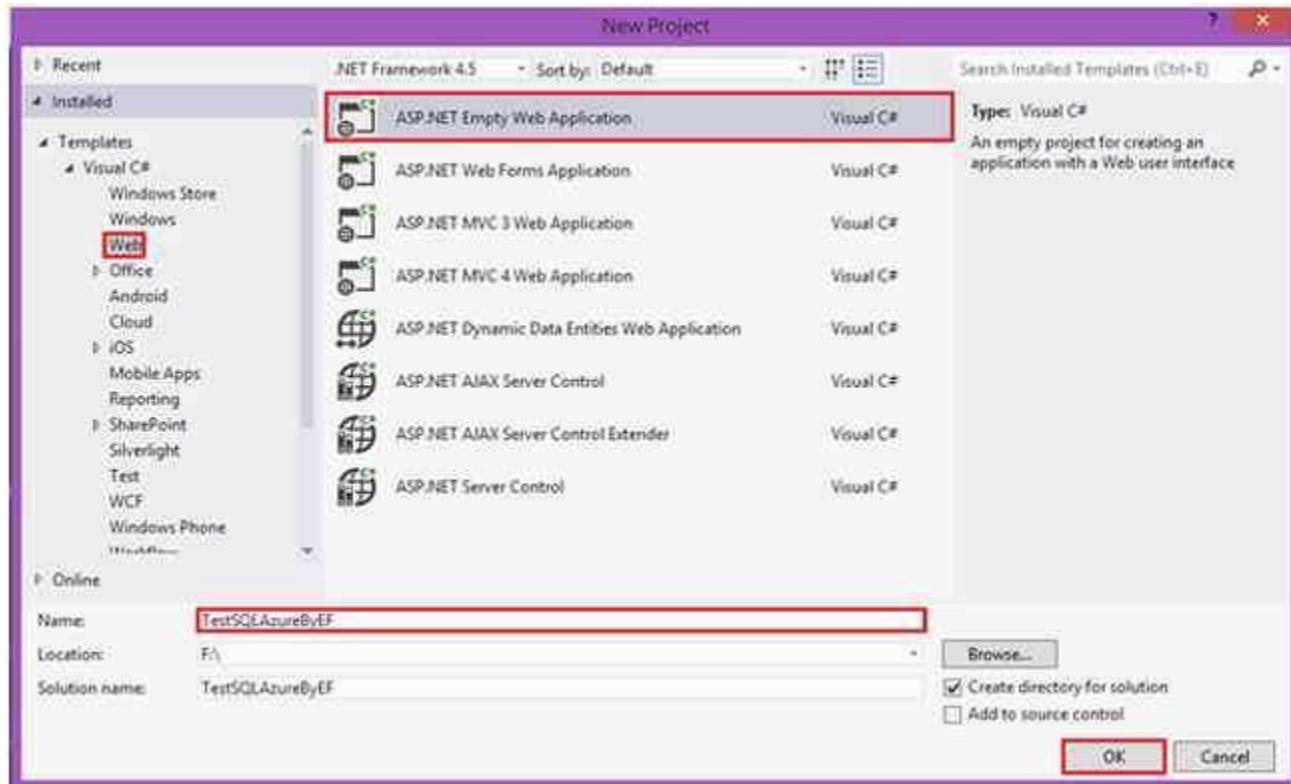


Id	Name	Salary	City
101	Nitin	1234	Noida

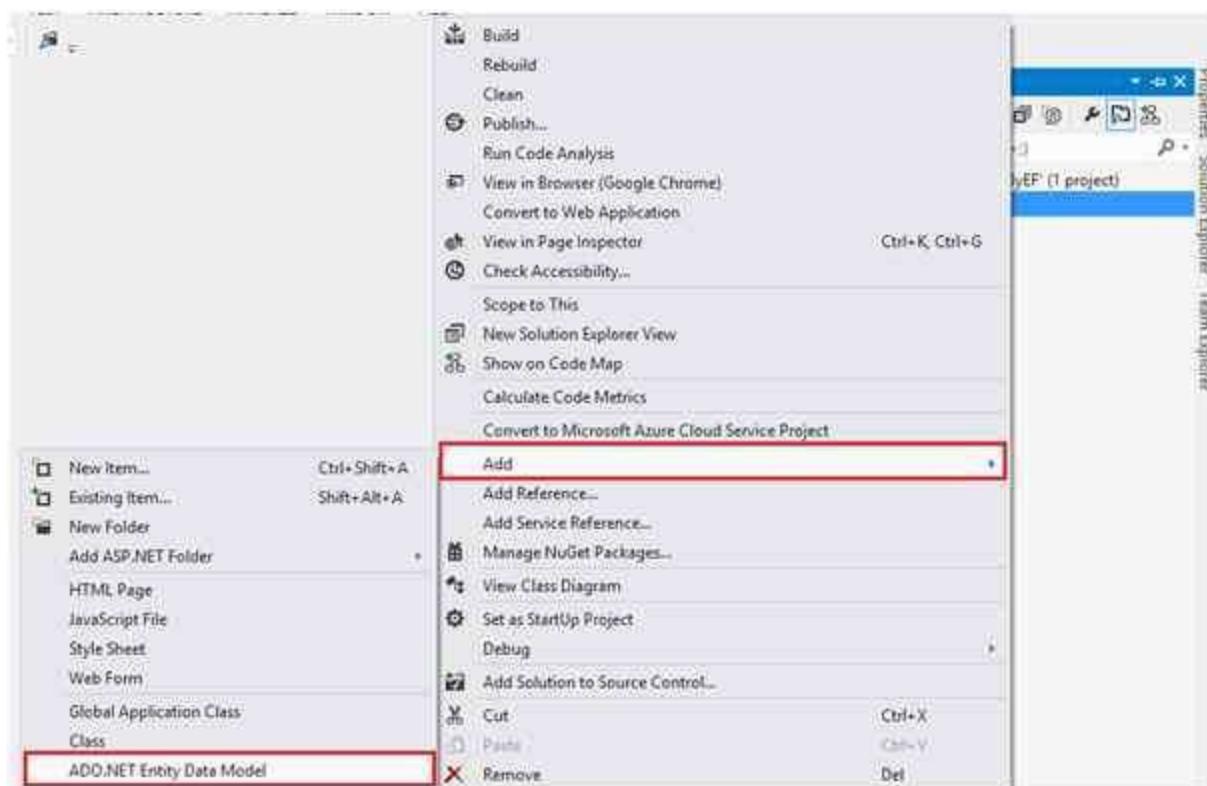
Employee Id :
Employee Name:
Employee Salary:
Employee City:

Now let's see How to Use SQL Azure Table for CRUD Operations by Entity Framework Database First approach

In the last point of this article we'll learn how to use SQL Azure Table in your Application with Entity Framework code first approach. So let's start, just create a basic empty application as the previous point.



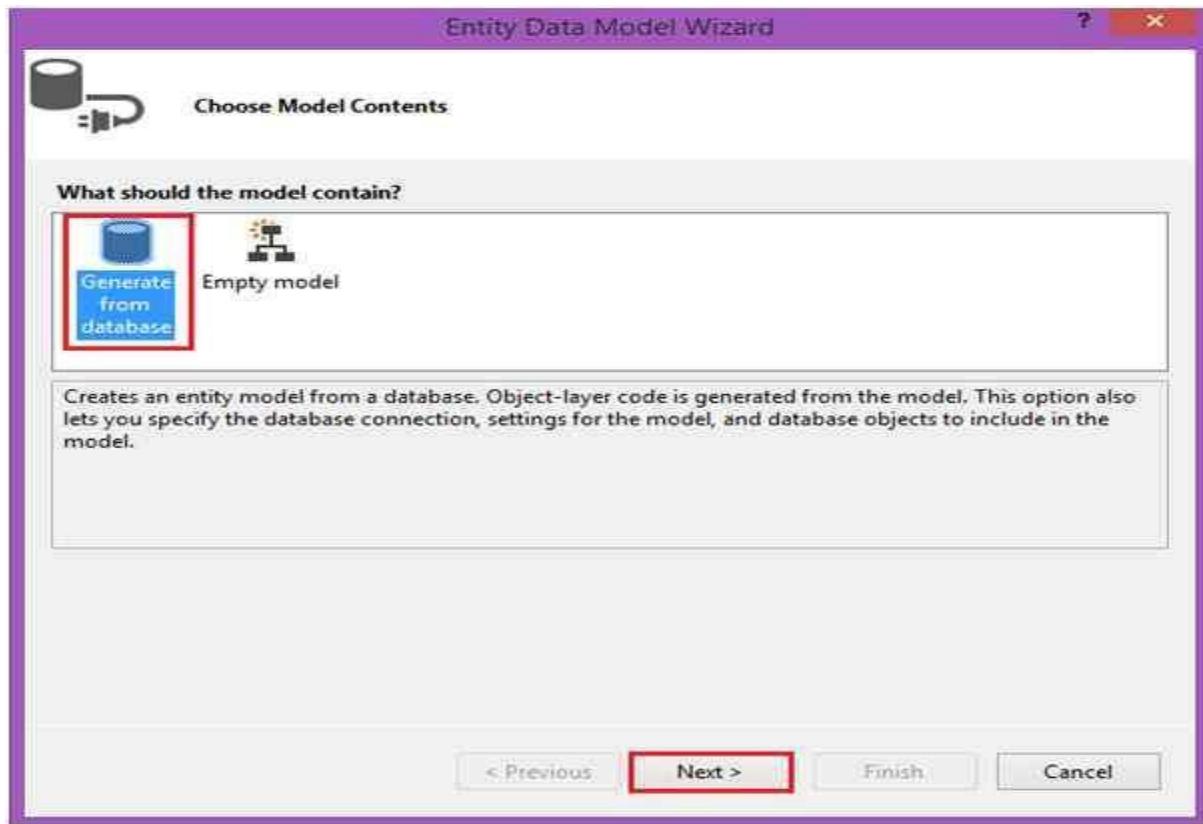
In Empty Solution, firstly, add an ADO.NET Entity Data Model.



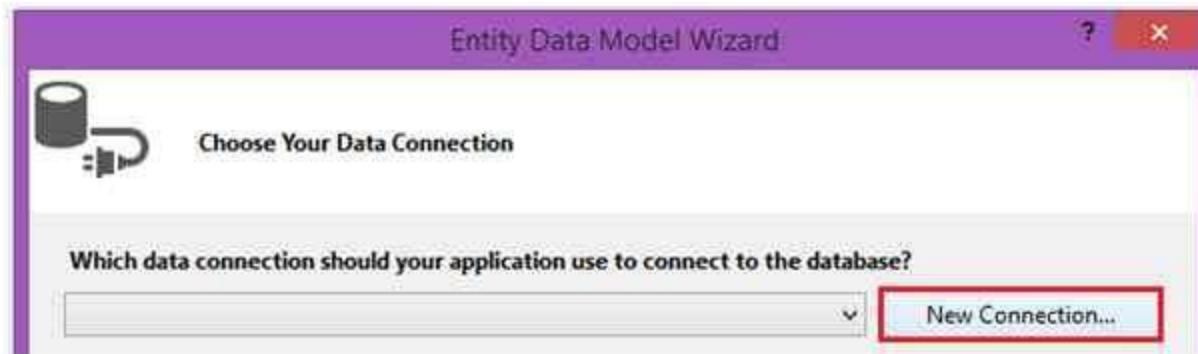
Give a name to your model and click OK.



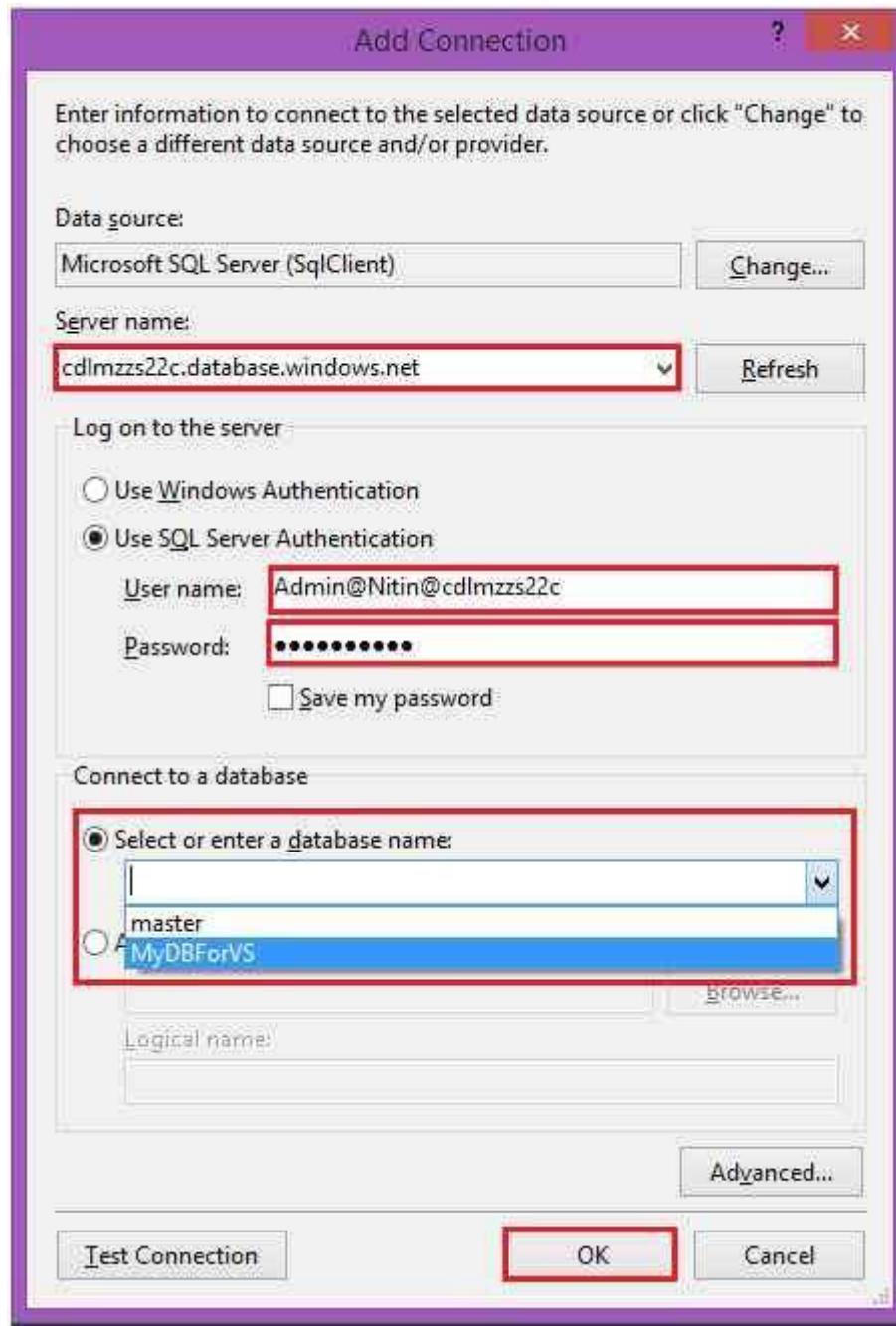
In this window, select Generate from database and click next.



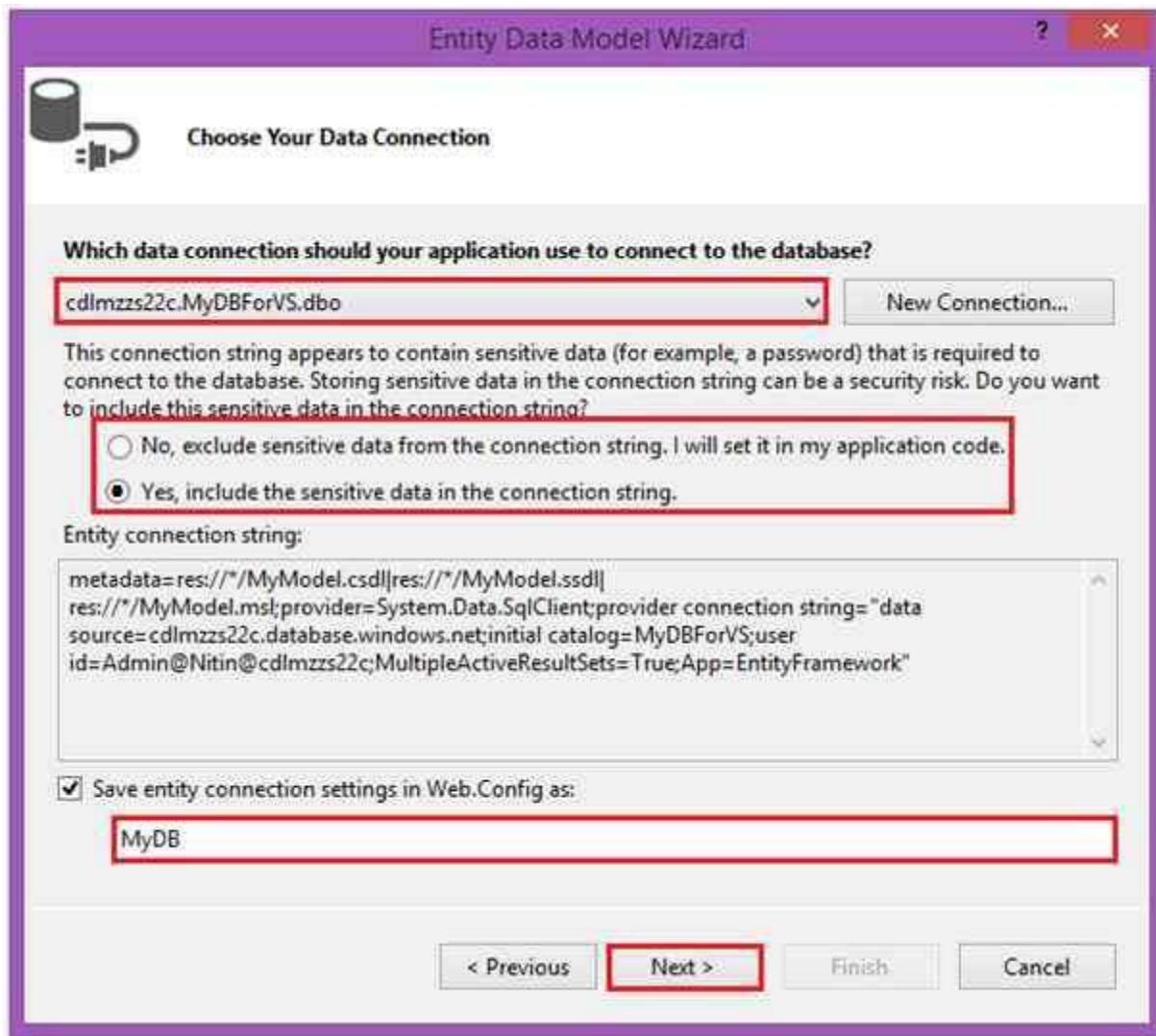
Now make a connection with SQL Azure Data server to ADO.NET Entity Model.



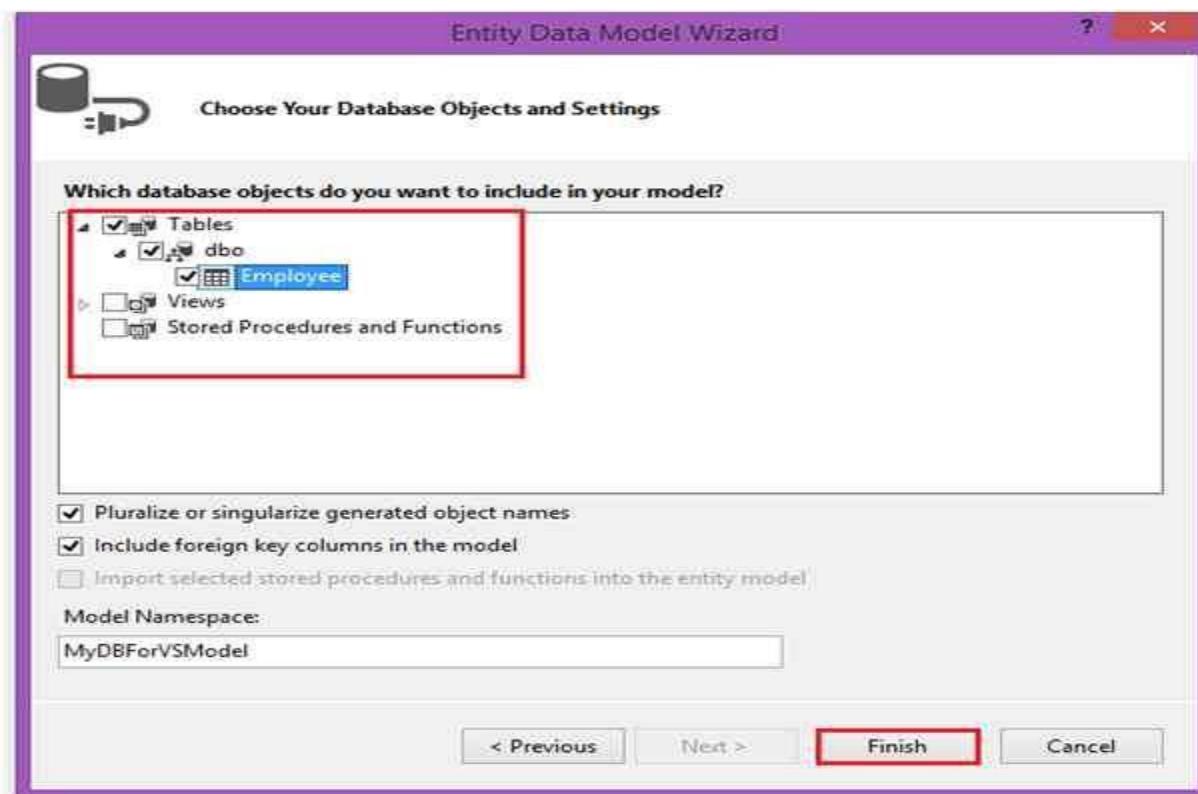
Enter all the details as per the previous point where we discussed how to connect in Server Explorer.



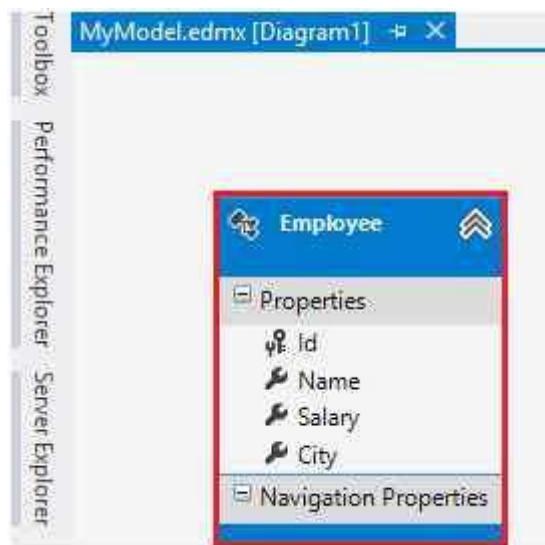
Click Next if the Next Button is enabled and if not, select a Radio Button, which is “*yes include the sensitive data to connection string*” and give a name to your Entity class name. Name it anything you want like I just gave a name **MyDB**. After that click on to next.



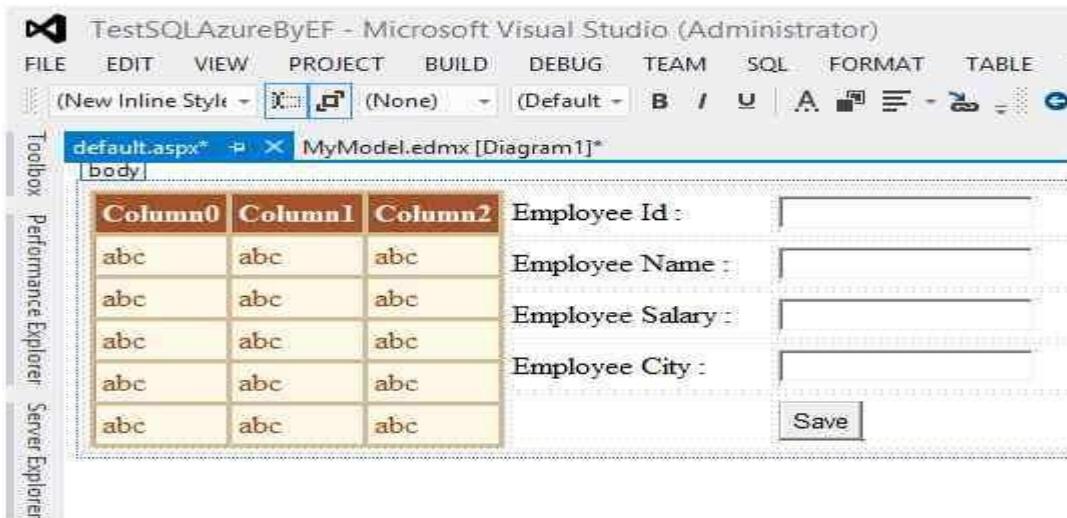
Now select how many tables you want, select them and press Finish Button.



Now you're Model and Entity File is ready to use.



Again create a view for the same as before for performing the CRUD operations on Entity model.



On the button click, I just created the object of my entity class and added this into entity collection and call the “SaveChange()” function of DbSet Class to save or update all the data to SQL Azure Table to live data.

Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

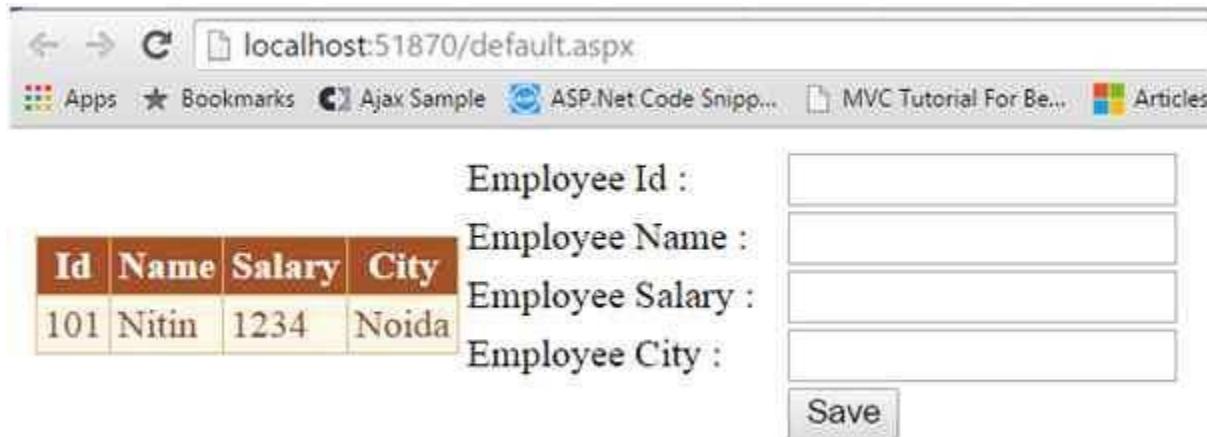
namespace TestSQLAzureByEF
{
    public partial class _default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
                Bind();
        }
        MyDB db = new MyDB();
        private void Bind()
        {
            GridView1.DataSource = db.Employees.Select(p=>p).ToList();
        }
    }
}
```

```

        GridView1.DataBind();
    }
protected void Button1_Click(object sender, EventArgs e)
{
    Employee emp = new Employee()
    {
        Id=Convert.ToInt32(TextBox1.Text),
        Name=TextBox2.Text,
        Salary=Convert.ToInt64(TextBox3.Text),
        City=TextBox4.Text
    };
    db.Employees.Add(emp);
    db.SaveChanges();
    Bind(); } } }
```

Output:

Initial output:



Id	Name	Salary	City
101	Nitin	1234	Noida

Employee Id :

Employee Name :

Employee Salary :

Employee City :

Save

After button click:



Id	Name	Salary	City
101	Nitin	1234	Noida
102	Rohan	12345	Delhi

Employee Id :

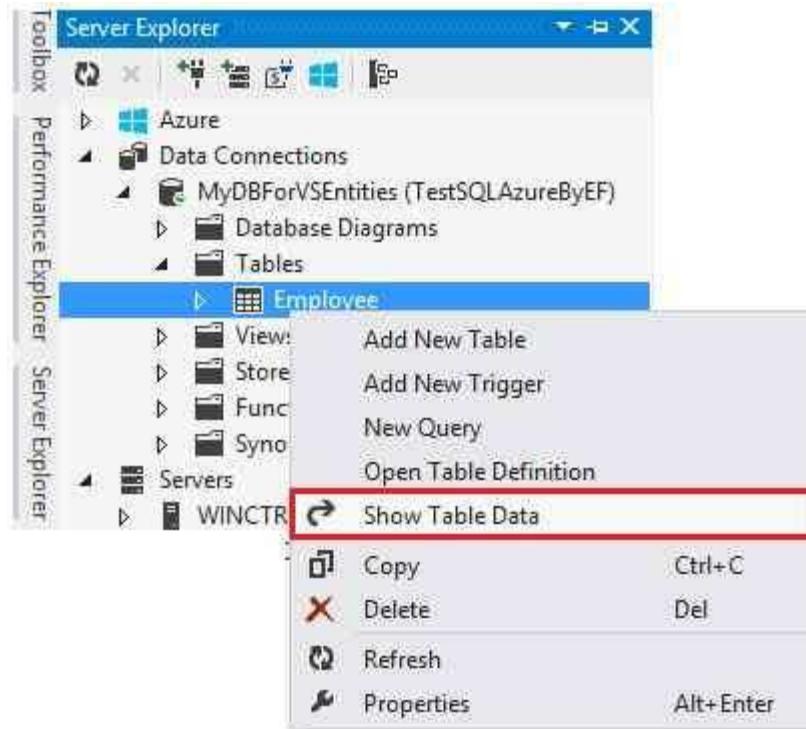
Employee Name :

Employee Salary :

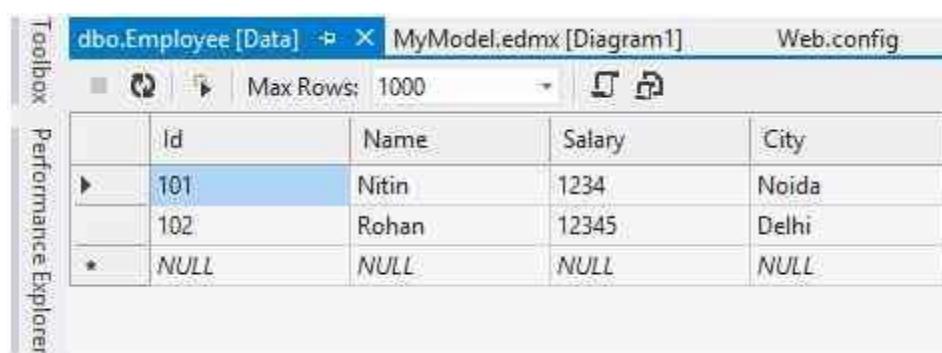
Employee City :

Save

After that go to Server Explorer, right click on table and show the table data.



Here's the data in table in Visual Studio.



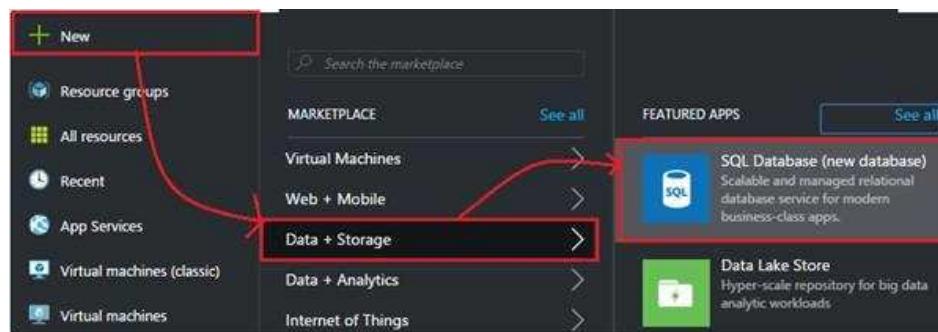
	ID	Name	Salary	City
▶	101	Nitin	1234	Noida
	102	Rohan	12345	Delhi
*	NULL	NULL	NULL	NULL

Entity Framework with Microsoft Azure SQL

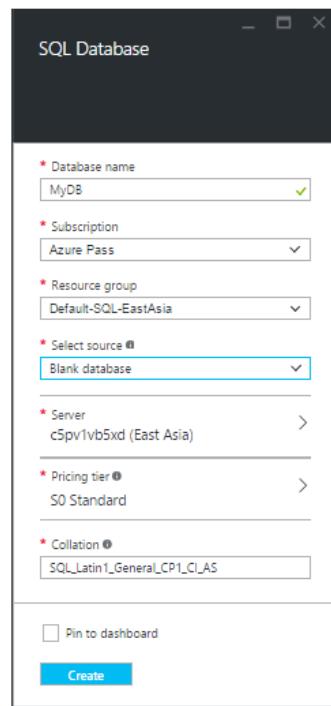
To use SQL Azure with your Entity Framework, first create an SQL database first in your Azure account.

So, let's make an SQL database. Here are the steps,

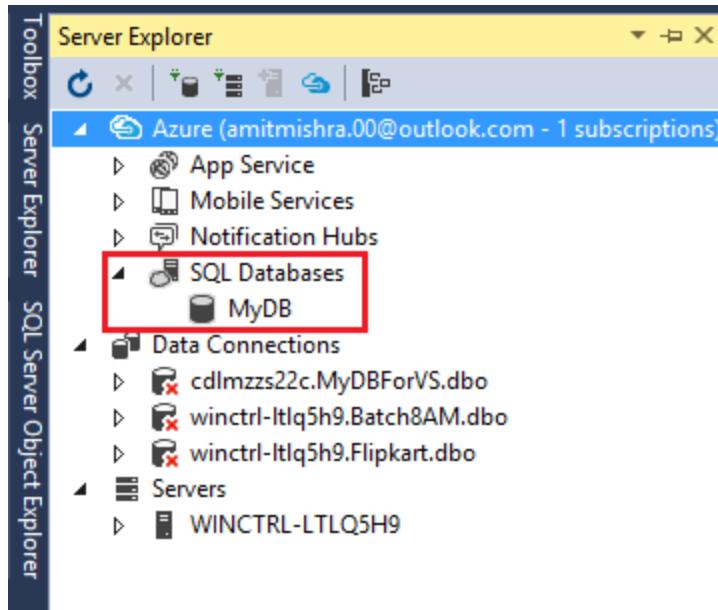
Go to your Azure account find *New, Data + Storage*, and then click *SQL Databases*.



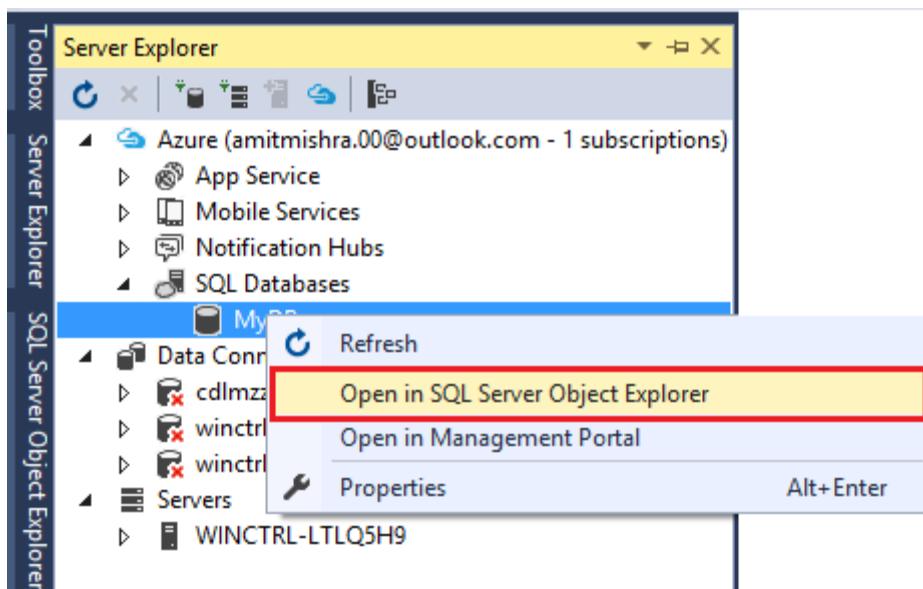
Fill the details and proceed to next.



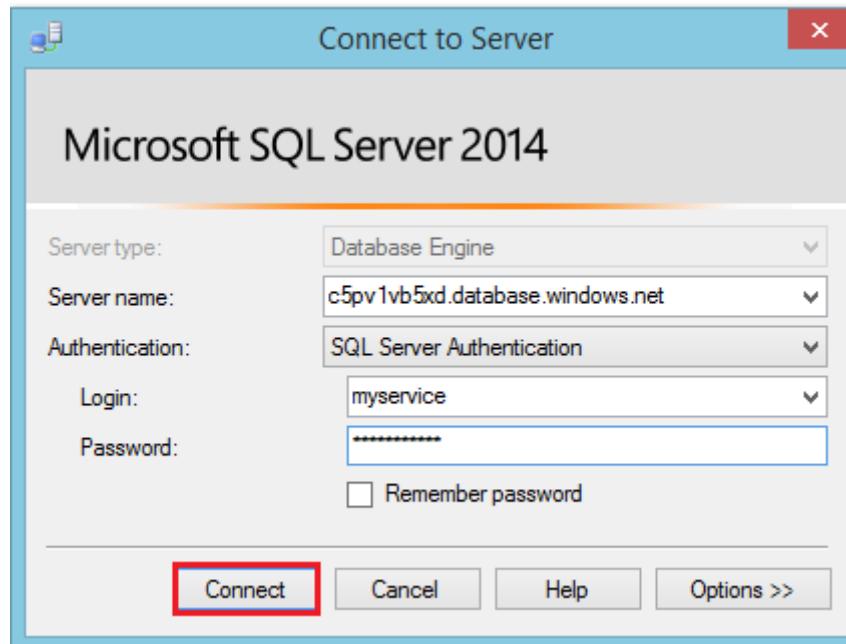
After creating SQL Azure database successfully, then just go to Visual Studio and open Server Explorer. Just login with Azure Account and find the database created in Azure.



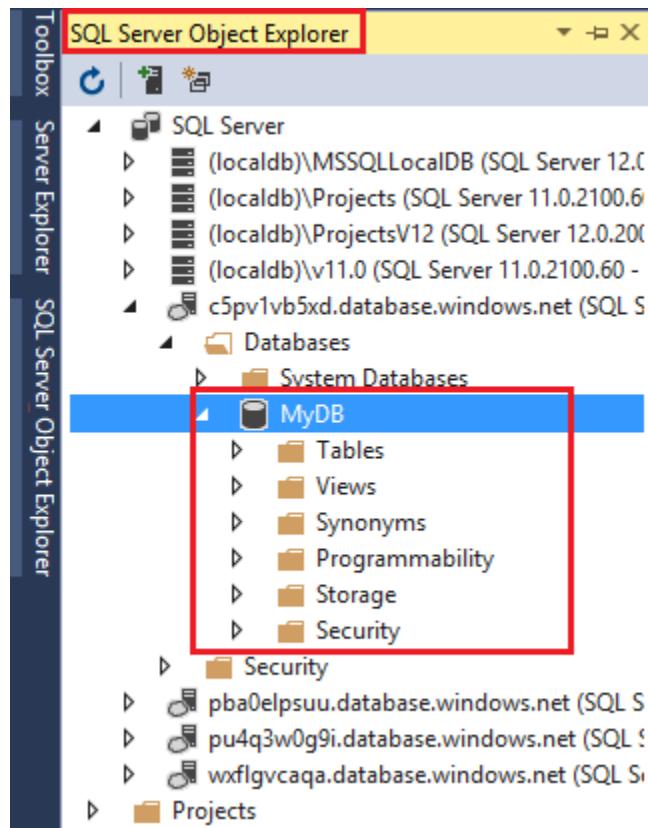
Right click on it and open it with SQL server Object Explorer so that we can create table and manage my database from wizard.

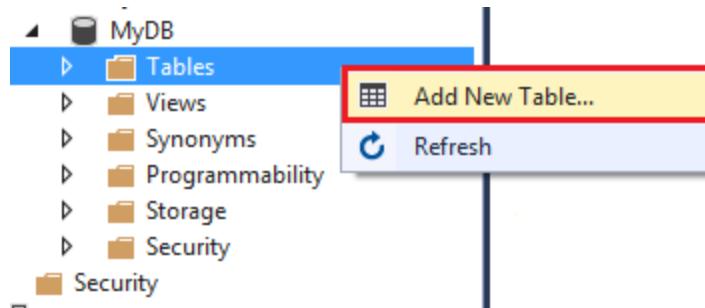


Just enter your database authentication details which were entered by you on the Azure portal to create a database and make sure that a client IP address has been added in to server configurations.

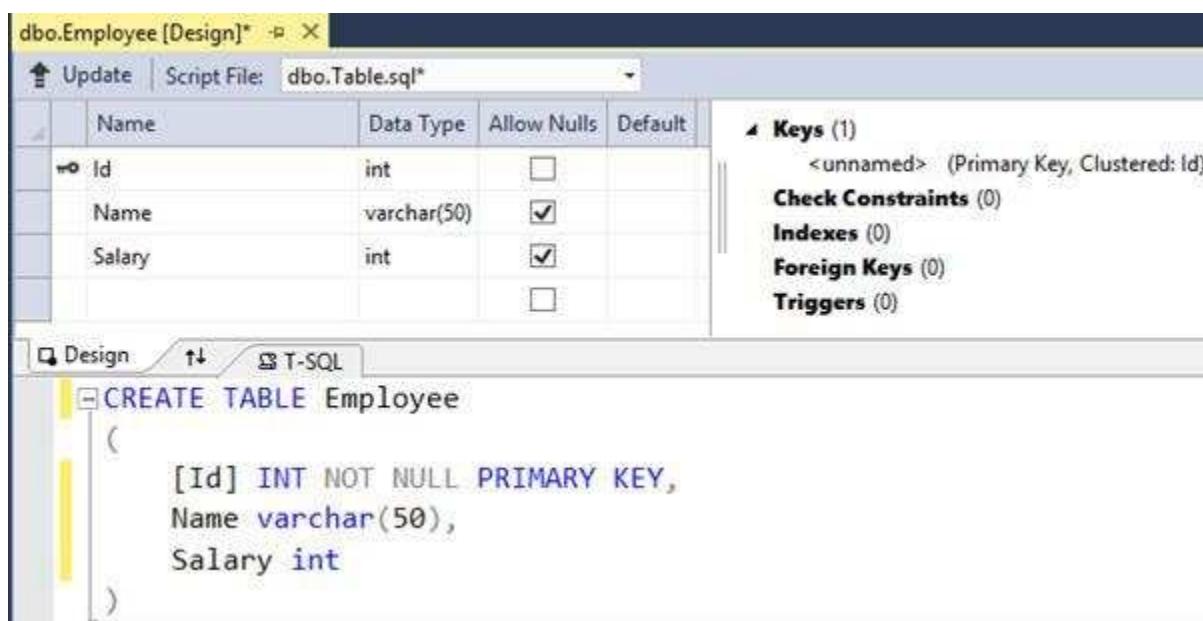


After connection with SQL Azure database into Server Explorer, you can see the database and all its tables and other things but I am not having any table yet, so make a table first. For that, right click on tables section and add a new table.





Just write your create table script on the script section and at last just click on update button.



dbo.Employee [Design]

Update | Script File: dbo.Table.sql*

	Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>		
Name	varchar(50)	<input checked="" type="checkbox"/>		
Salary	int	<input checked="" type="checkbox"/>		

Keys (1)
 <unnamed> (Primary Key, Clustered: Id)

Check Constraints (0)

Indexes (0)

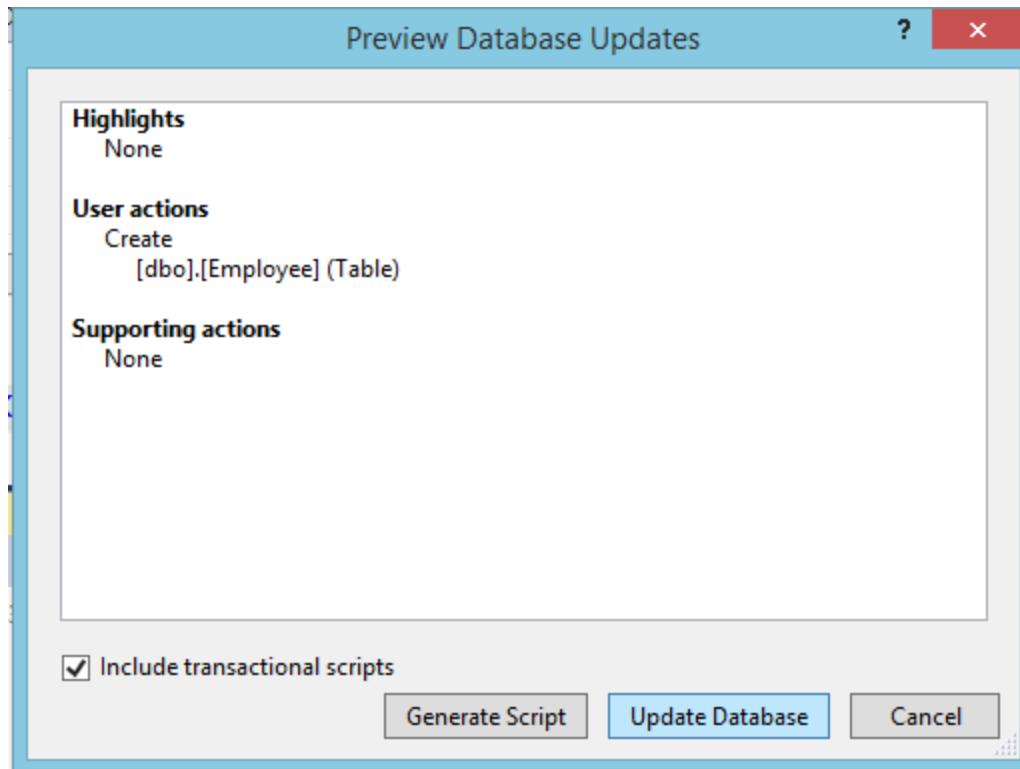
Foreign Keys (0)

Triggers (0)

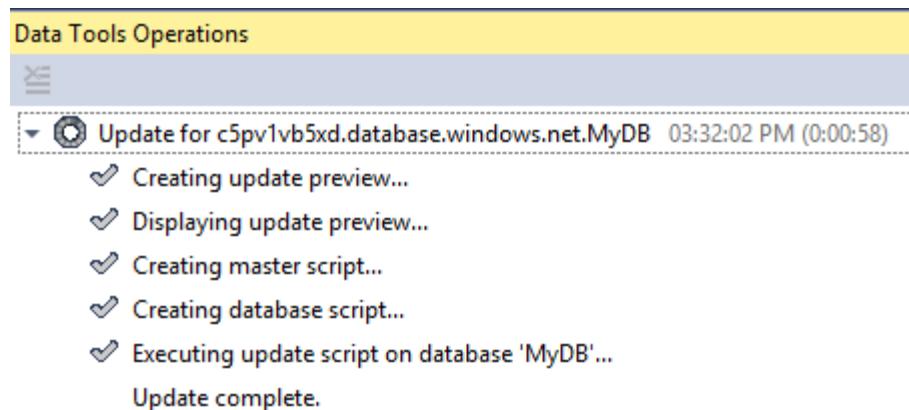
T-SQL

```
CREATE TABLE Employee
(
    [Id] INT NOT NULL PRIMARY KEY,
    Name varchar(50),
    Salary int
)
```

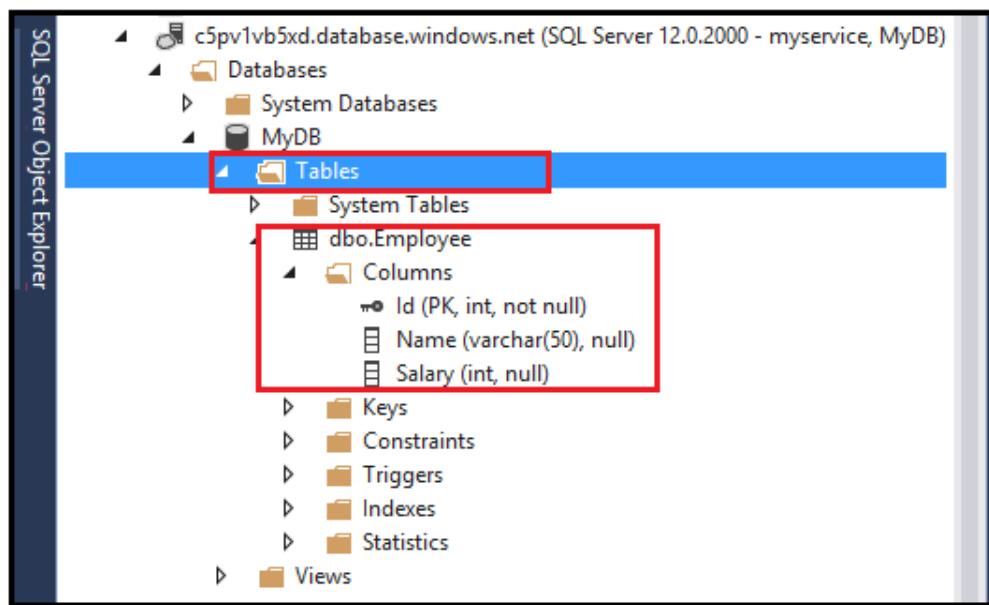
Now, you'll get dialog where you have to click on “*Update Database*” button.



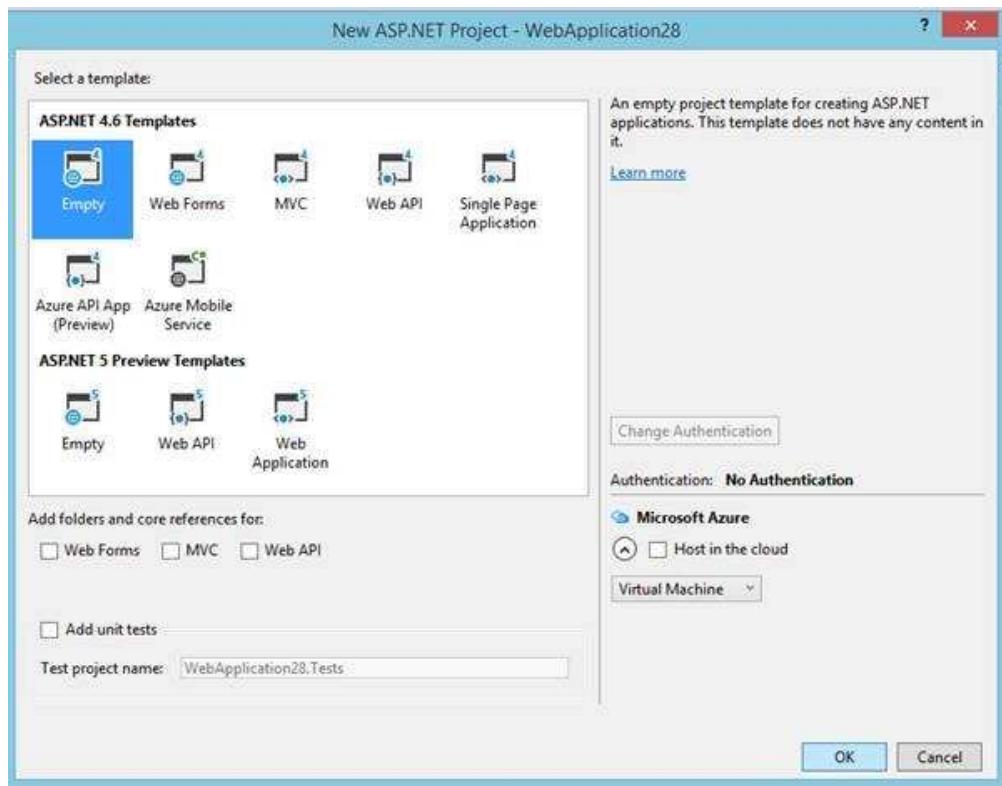
Now it's creating a table you see your query updates in Data tools options.



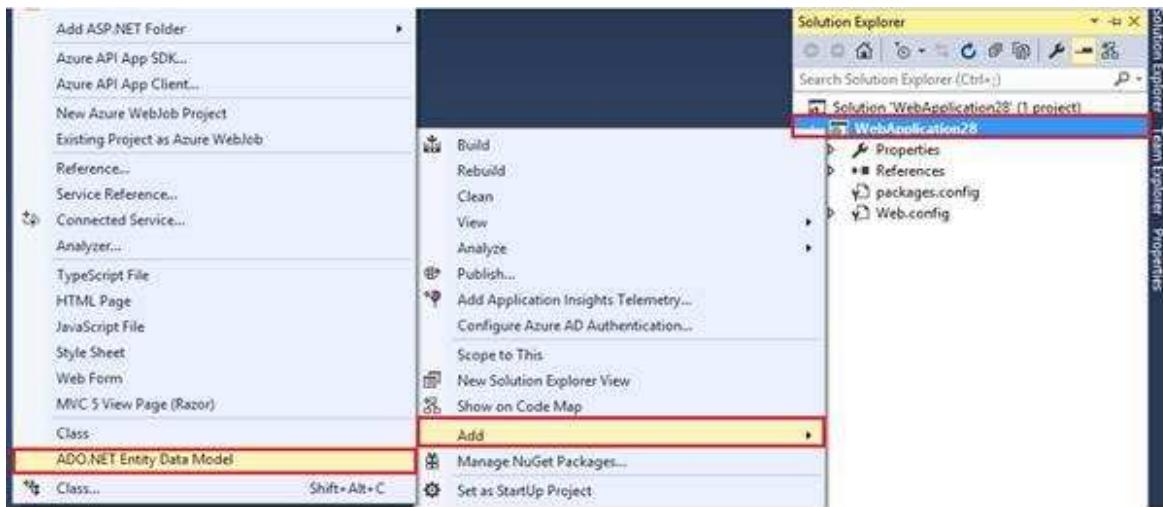
Just go to back on SQL Server object explorer and refresh tables. You'll get the new table created by your query.



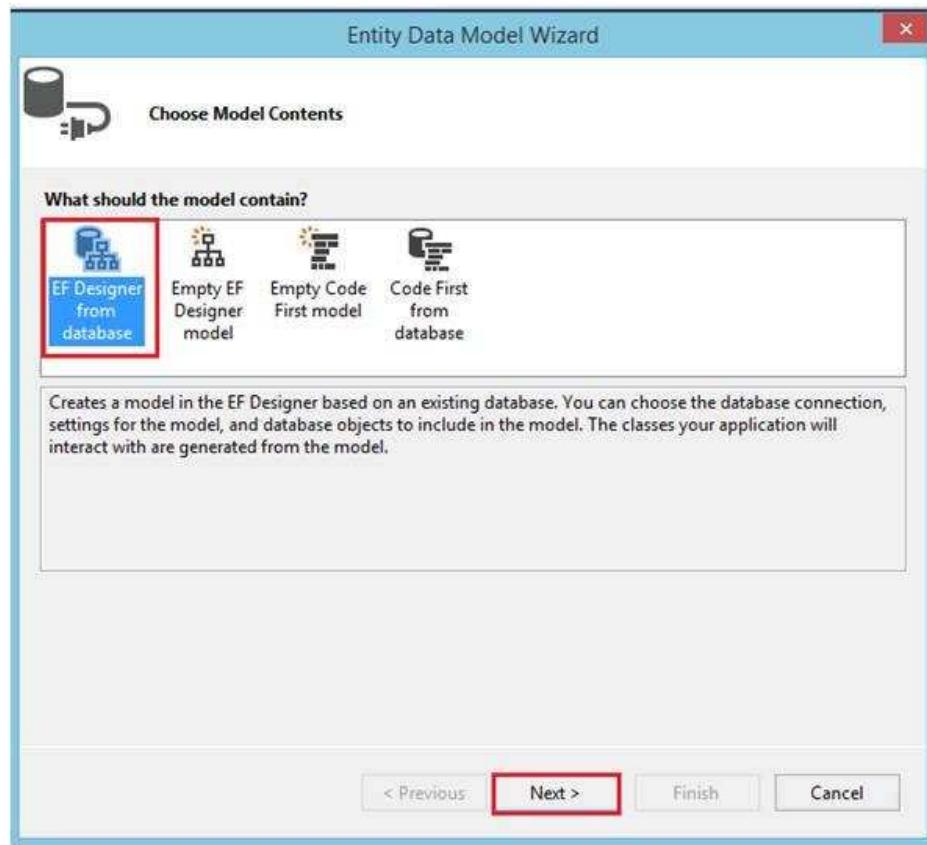
Now just add an ASP.NET or any other application where you want to use this database. I am going to use an ASP.NET project to use this database with entity framework.



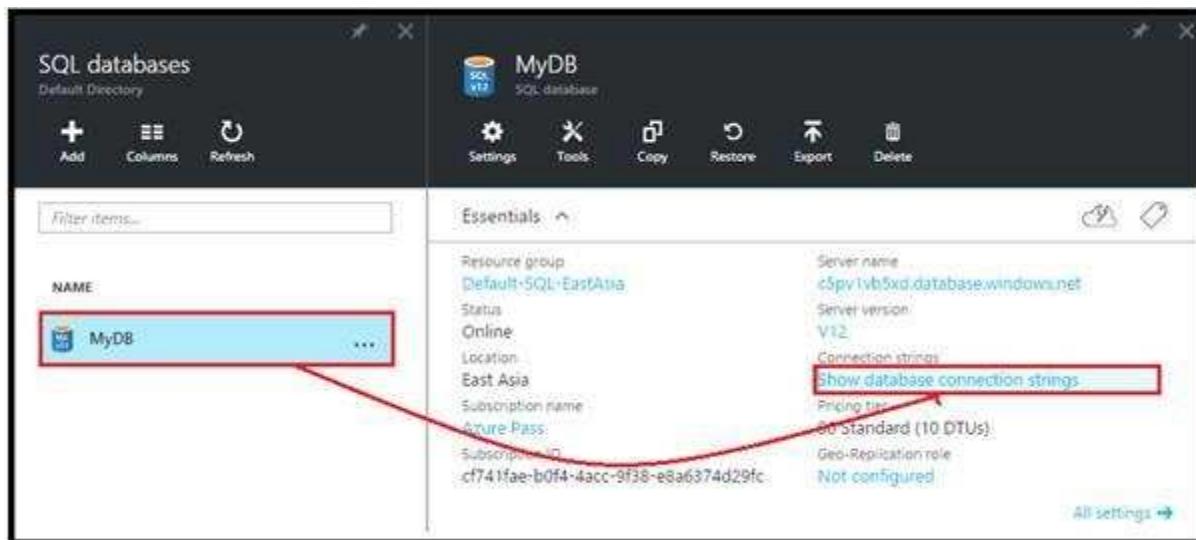
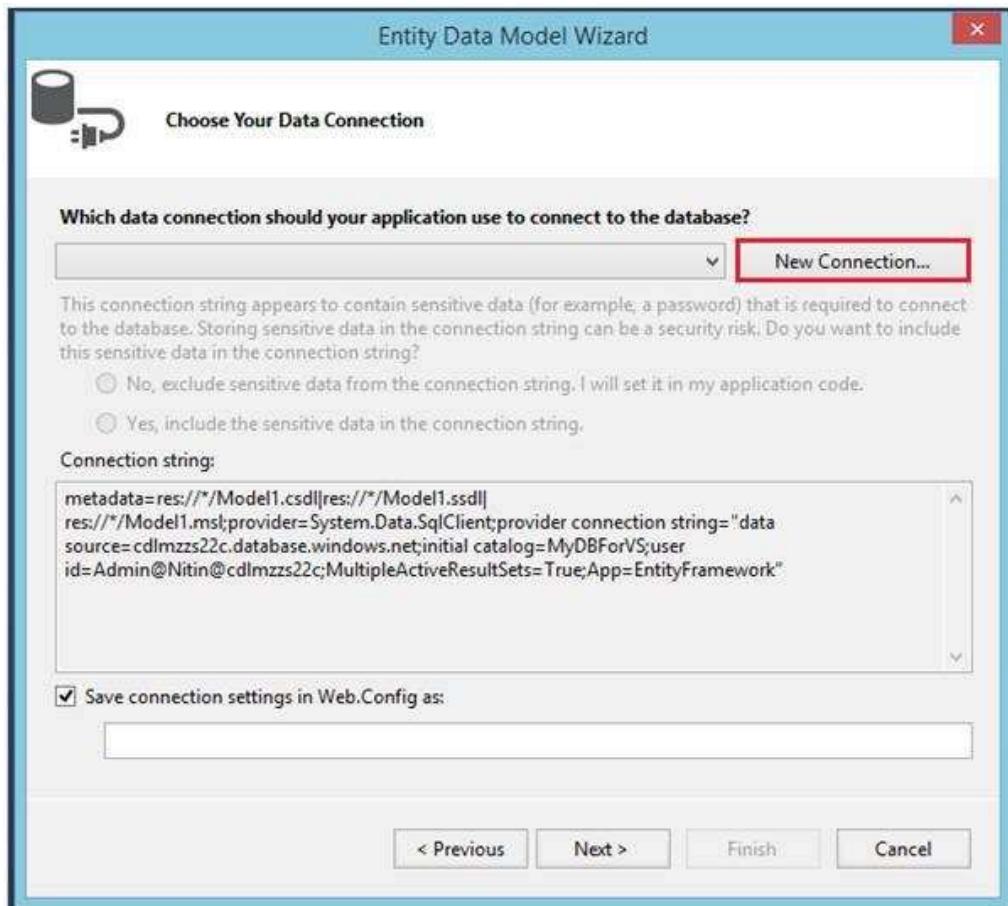
Now just add an Entity framework.



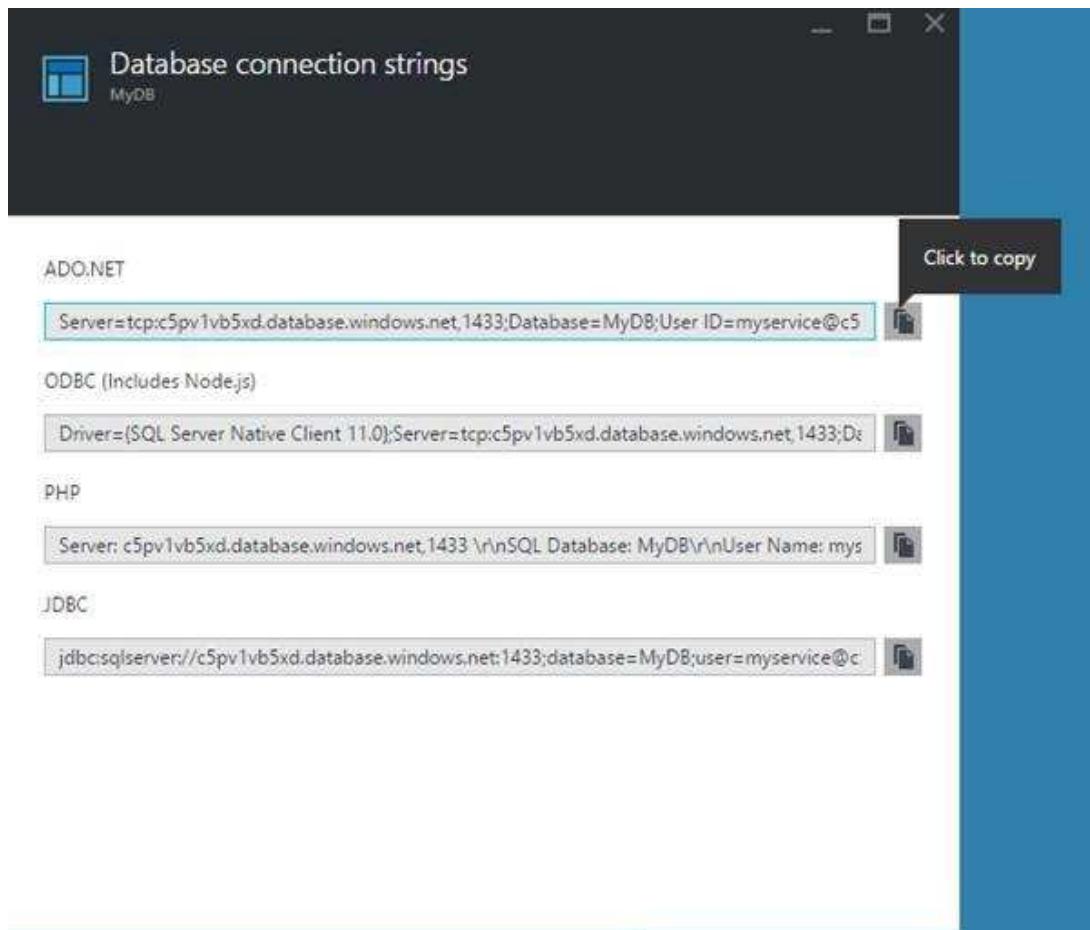
Select any one, but I am going to use Database first with entity.



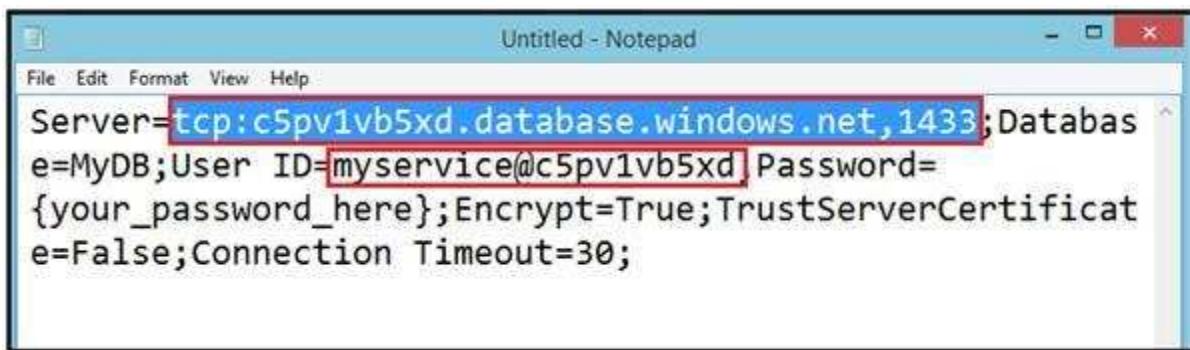
Now just make connection with Azure cloud service where your SQL Azure is hosted. For that you can go to your portal and get connection string.



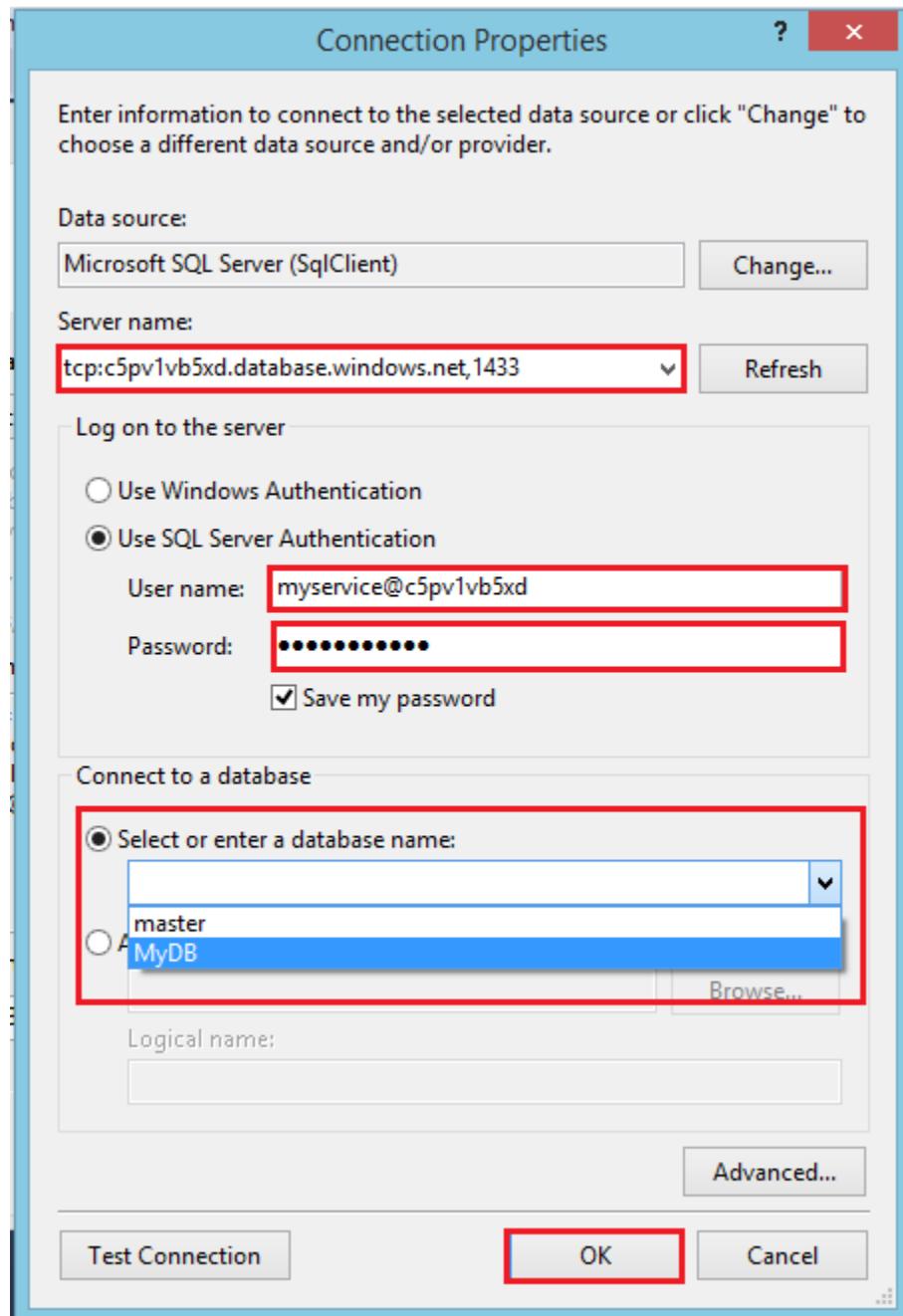
Copy that and open it in text file and find server name.

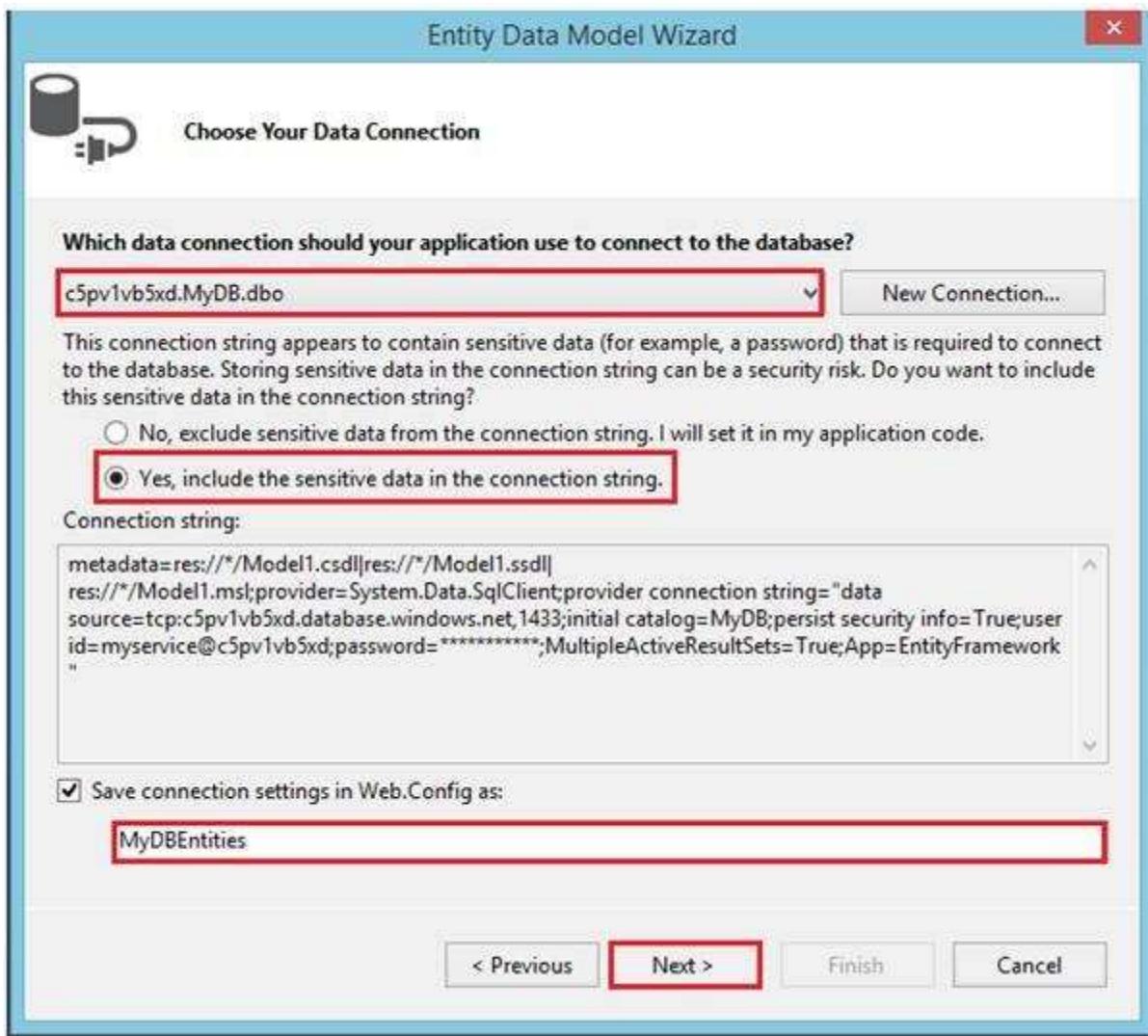


Copy Server name and user id,

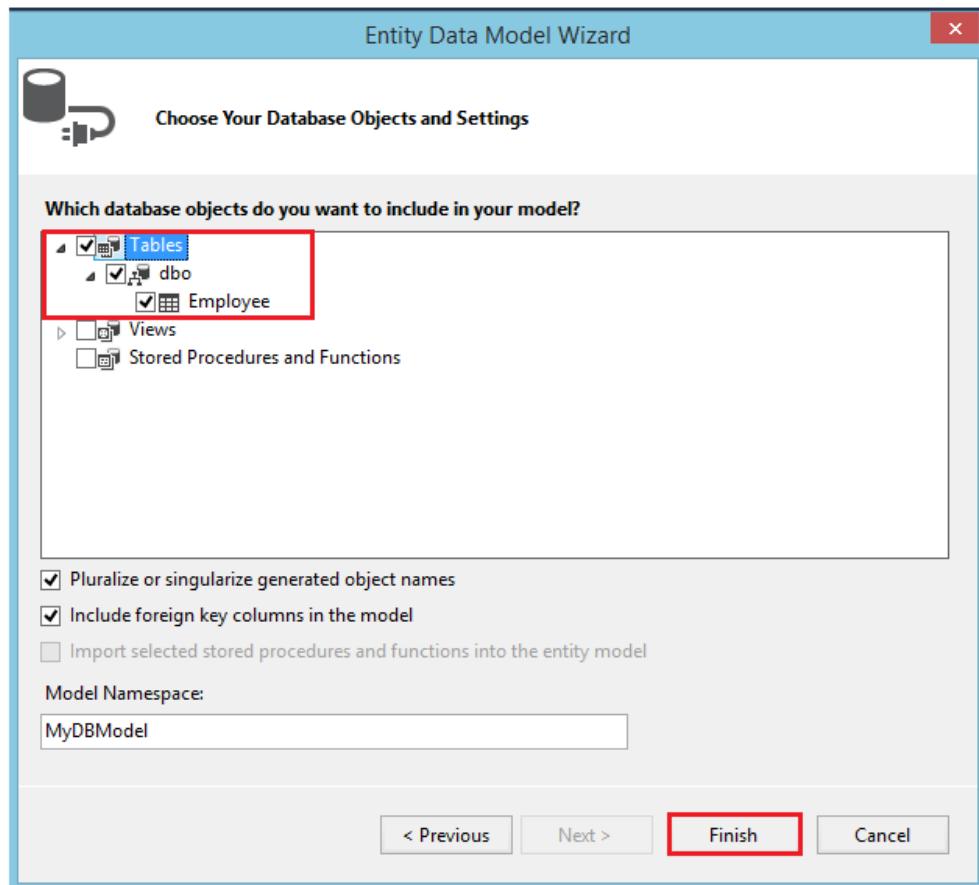


Use them and make a successful connection.

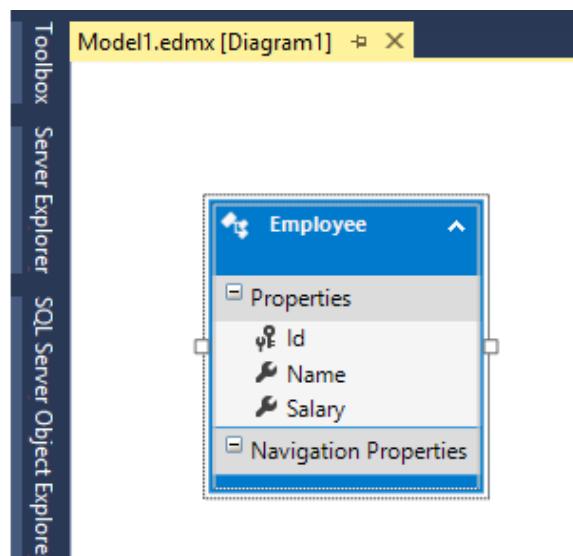




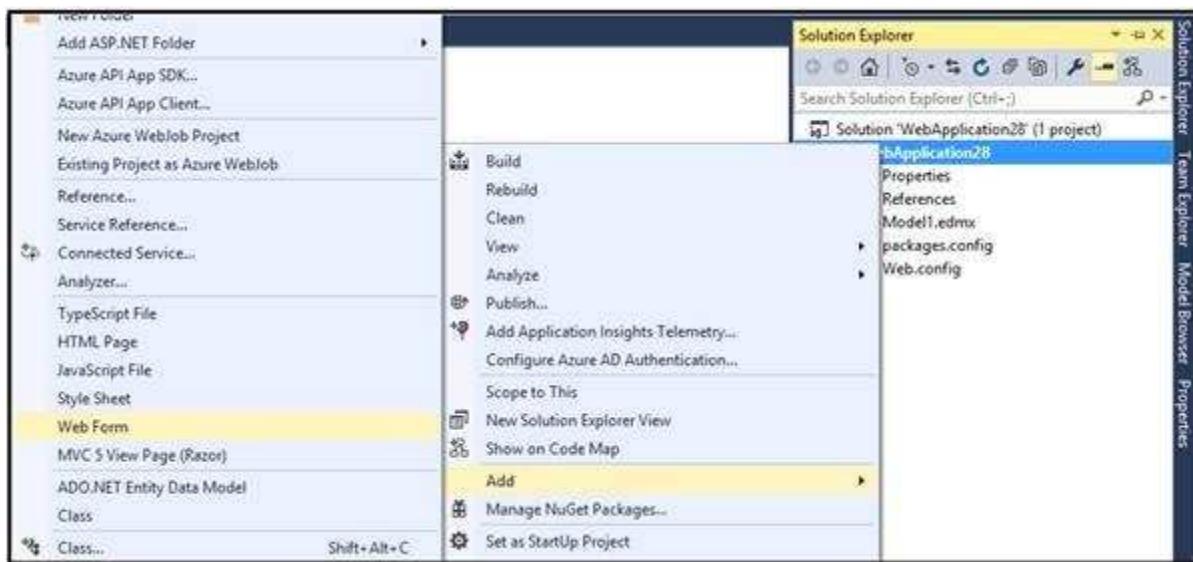
Give a name to your entity and select your tables which you want to use.



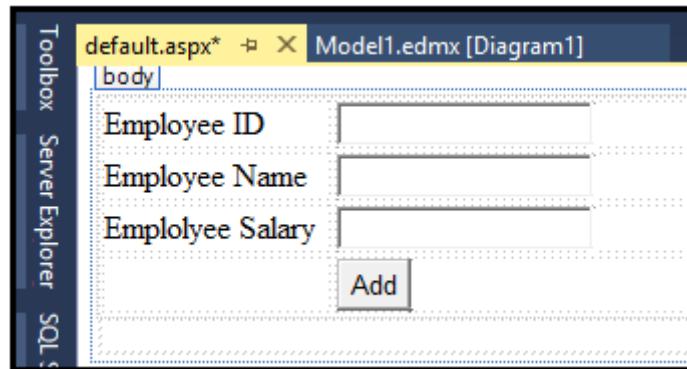
Now this is your model file.



Now just add an ASP Web Form to write operations.



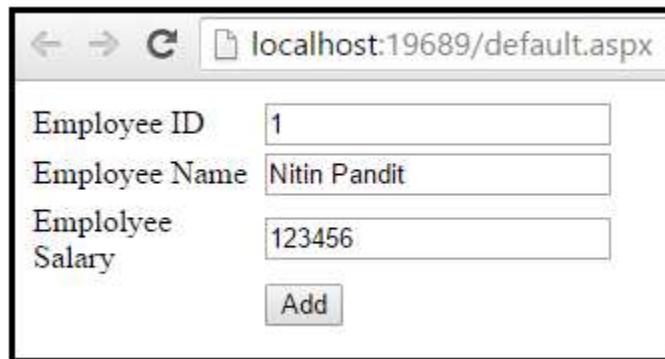
Just use some textbox to get the test information from page and a button. On button click event write code to add an object to an entity table as in the following code.



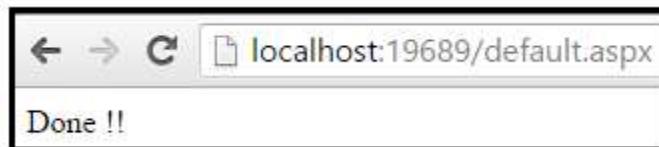
Test code to save object information to table.

```
0 references
protected void Button1_Click(object sender, EventArgs e)
{
    using (MyDBEntities db = new MyDBEntities())
    {
        Employee emp = new Employee()
        {
            Id=Convert.ToInt32(TextBox1.Text),
            Name=TextBox2.Text,
            Salary=Convert.ToInt32(TextBox3.Text)
        };
        db.Employees.Add(emp);
        db.SaveChanges();
        Response.Write("Done !!");
    }
}
```

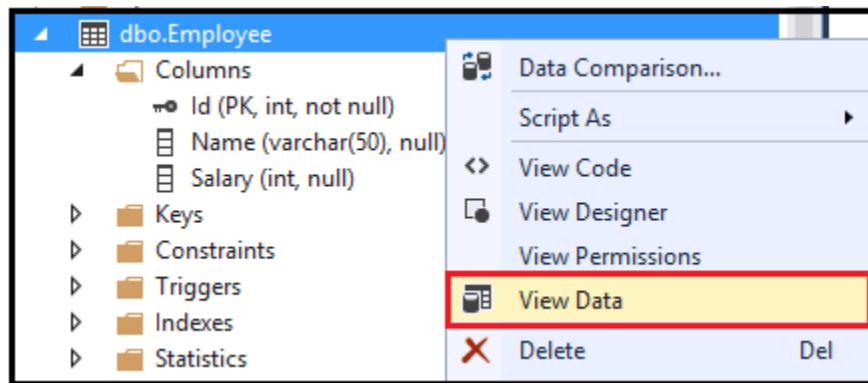
Now at last just run this application and enter some details and click on button.



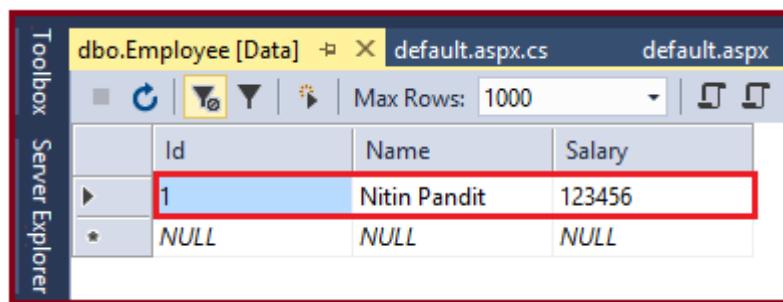
After that, you'll get the message “*Done !!*” as I use to notify all done successfully.



Go to table and right click on table in SQL server object explorer and find *View Data* option.



So this is the data entered by you from ASP form.



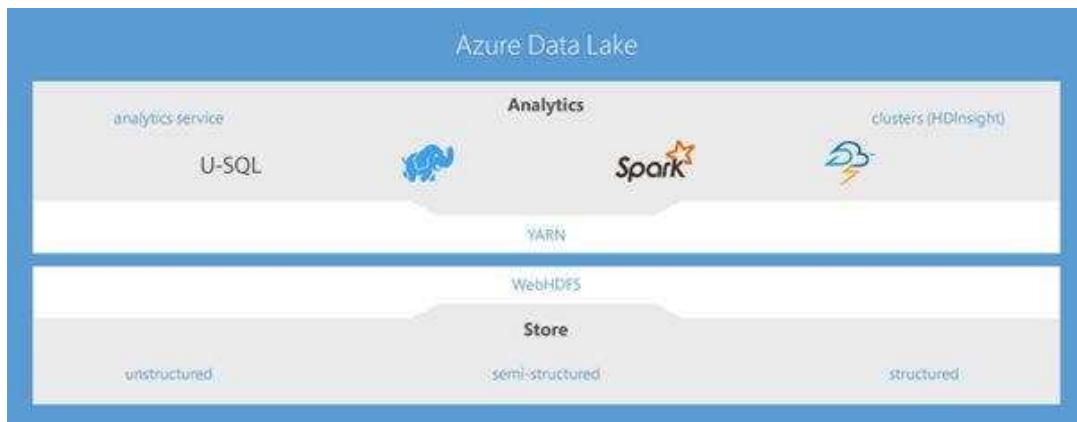
	Id	Name	Salary
▶	1	Nitin Pandit	123456
★	NULL	NULL	NULL

I hope you enjoyed this chapter.

Introducing U-SQL- Make Big Data Processing Easy

Microsoft recently announced a new feature for Azure Services, **Azure Data Lake** Services for analytics in the Microsoft Azure cloud services. It is having a large scale repository and also a default service **YARN** that provide a benefit for analyzing the large scale data to a developer or DBA and also for the data scientists. **YARN** is Apache Hadoop NextGen MapReduce known as MapReduce 2.0 (MRv2) or called **YARN**.

Azure Data Lake has a managed way to manage all **Hadoop**, **Spark** and **HBase** Services. Azure Data Lake use U-SQL. It is a language that explores the benefits of SQL and is scalable and provides a distributed query capabilities for a developer and DBA those are currently working with **Big Data**. Also, it provides you to efficiently analyze your data in the store and across relational stores such as Microsoft SQL Azure Database.



Benefits of Data Lake-

Batch, Real-time, and interactive analytics made it easy to:

- Store and analyze data of any kind and size.
- Develop faster, debug and optimize smarter.
- Interactively explore patterns in your data.
- No learning curves—use U-SQL, Spark, Hive, HBase, and Storm.
- Managed and supported with an enterprise-grade SLA.
- Dynamically scales to match your business priorities.

- Enterprise-grade security with Azure Active Directory.
- Built on YARN, designed for the cloud.

Why use U-SQL?

U-SQL is a large scalable Language and if we see the behaviour of Big Data analytics we can have several requirements. Here are the major requirements as per the MSDN blog.

- **Process any type of data** From analysing BotNet attack patterns from security logs to extracting features from images and videos for machine learning, the language needs to enable you to work on any data.
- **Use custom code easily** to express your complex, often proprietary business algorithms. The example scenarios above may all require custom processing that is often not easily expressed in standard query languages, ranging from user defined functions, to custom input and output formats.
- **Scale efficiently to any size of data** without you focusing on scale-out topologies, plumbing code, or limitations of a specific distributed infrastructure.

How to use U-SQL?

If you want to see how we can use U-SQL. Go through the MSDN blog example. Let's assume I have downloaded my Twitter history of all my tweets, retweets, and mentions as a CSV file and placed it into my Azure Data Lake Store. Here top 50 rows in the .csv file can be seen:

Stream Preview (MyTwitterHistory.csv) X Azure Data Lake Explorer: mryskona CodeBehindAggTwee...sMention:

File Information

Name: MyTwitterHistory
 Created: 9/10/2015 5:04:58 PM
 Modified: 9/10/2015 5:04:59 PM
 File Size: 70,824 bytes
 Path: swebhdfs://mryskona.azuredatalake.net/input/MyTwitterHistory.csv

Top 50 rows in file MyTwitterHistory.csv

	Column_0	Column_1	Column_2	Column_3
1	14/02/2015	19:23	iC	Mongo 3.0 allows to use different storage engines.
2	04/02/2015	08:16	MikeDoesBigData	For those of you who are into cars, check out my
3	04/02/2015	08:14	MikeDoesBigData	How time flies! Over 18 months. I guess I am too.
4	25/09/2014	03:50	iC	Animated @sqlservermike on first row drilling in
5	12/03/2014	20:35	jamiet	Listening to @robconery on #dotnetrocks eulogies.
6	22/01/2014	18:42	iC	Koverse + Smaza tonight at Seattle Scalability hackathon.
7	30/10/2013	10:55	syediddi	@SQLServerMike Thanks for a great video on SC
8	16/09/2013	07:55	GiessweinWeb	@SQLServerMike Hi do you know http://t.co/yIJv
9	04/09/2013	20:11	MikeDoesBigData	@dracopaladine You can use STEquals: SELECT *
10	04/09/2013	21:47	dracopaladine	@SQLServerMike @mypt any good URLs to reso
11	04/09/2013	21:49	dracopaladine	@SQLServerMike ah left off =1 in where clause.

In this just count the number of tweets for each of the authors in the tweet “**network**”:

```
@t = EXTRACT date string
, time string
, author string
, tweet string
FROM "/input/MyTwitterHistory.csv"
USING Extractors.Csv();
```

```
@res = SELECT author
, COUNT(*) AS tweetcount
FROM @t
GROUP BY author;
```

```
OUTPUT @res TO "/output/MyTwitterAnalysis.csv"
ORDER BY tweetcount DESC
USING Outputters.Csv();
```

Here are the following three major steps of processing data with U-SQL from the above U-SQL script:

1. **Extract** data from your source. Datatypes are based on C# data types and the built-in extractor's library to read and schematize the CSV file is used.
2. **Transform** using SQL and/or custom user defined operators. A familiar SQL expression that does a GROUP BY aggregation is used in the preceding example.
3. **Output** the result into a file. You can also store it into U-SQL table for further processing.

This was just an introduction to U-SQL, for further understanding refers the msdn blog examples to:

- Add additional information about the people mentioned in the tweets.
- Extend my aggregation to return how often people in my tweet network are authoring tweets and how often they are being mentioned.

Installation of Azure SDK and Blob Storage in Azure

This is about Azure SDK and Storage Account in Azure. In this article you will also learn how to store images in Blob storage to Azure. So, let's start with the following three steps:

1. How to install Microsoft Azure SDK in Visual Studio 2013?
2. How to install Azure Explorer?
3. How to create Storage Account in the Azure Microsoft Account?

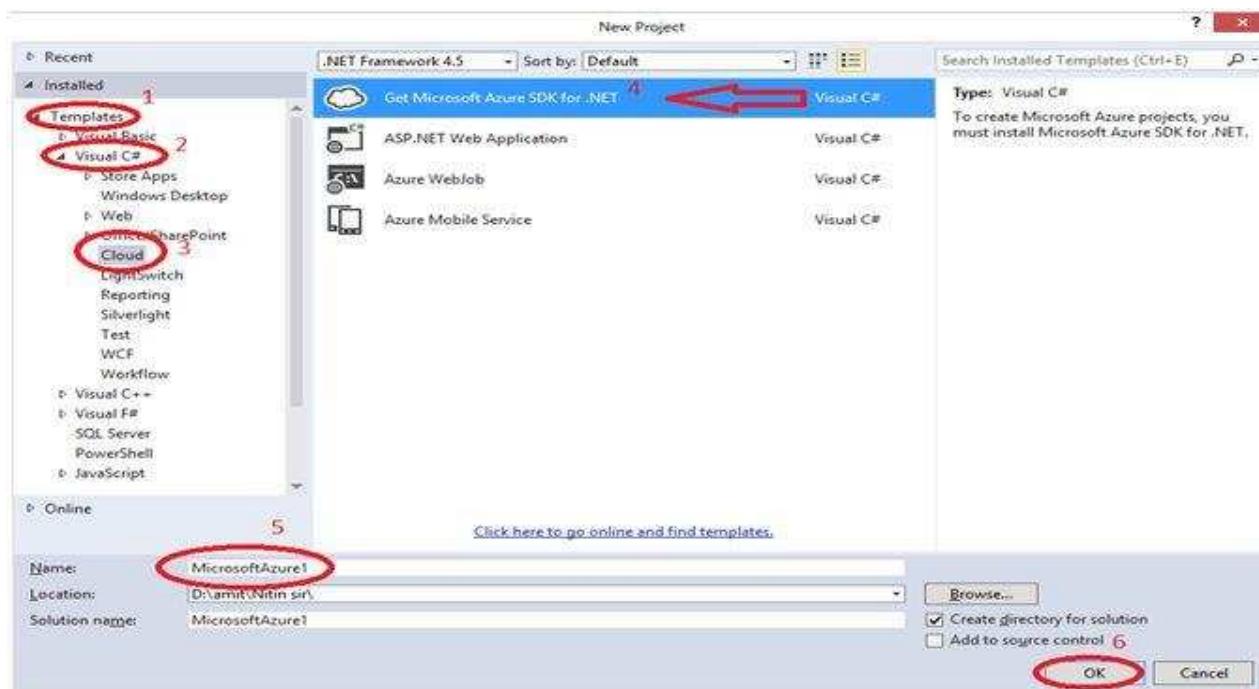
How to install Microsoft Azure SDK in Visual Studio 2013

Step 1: Open Visual Studio 2013, go to Templates, then Visual C# and click **Cloud**.

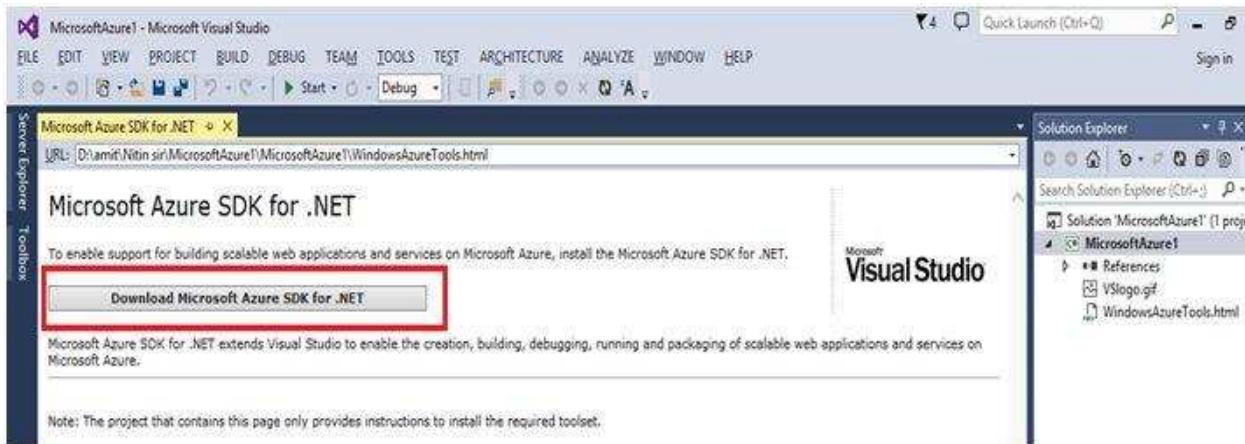
Step 2: Select “**Get Microsoft Azure SDK for .NET**”.

Step 3: In the name section, it will take a default name “**MicrosoftAzure1**”.

Step 4: Click OK button.



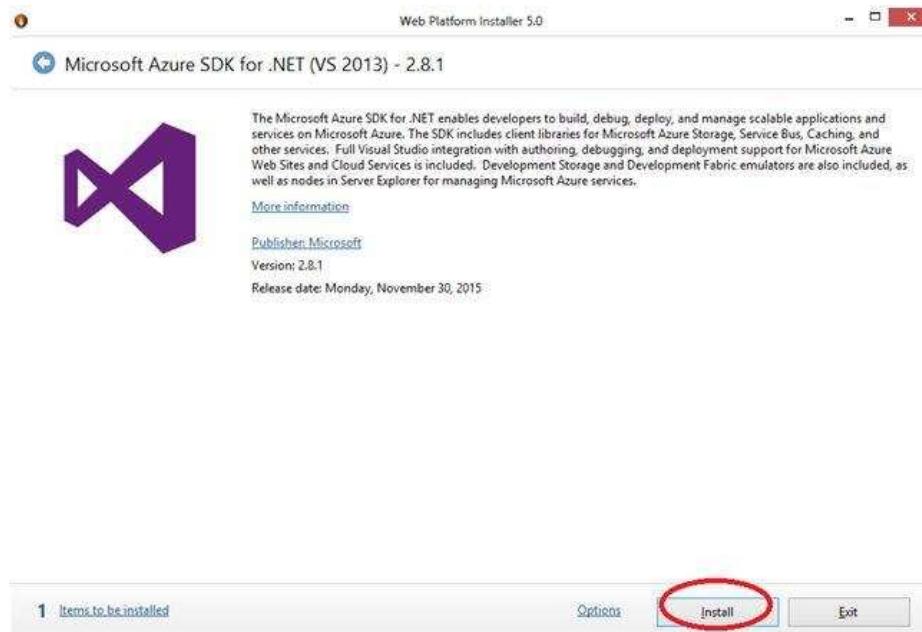
In the following figure, click on to “**Download Microsoft Azure SDK for .NET**”. (Shown in red rectangular)



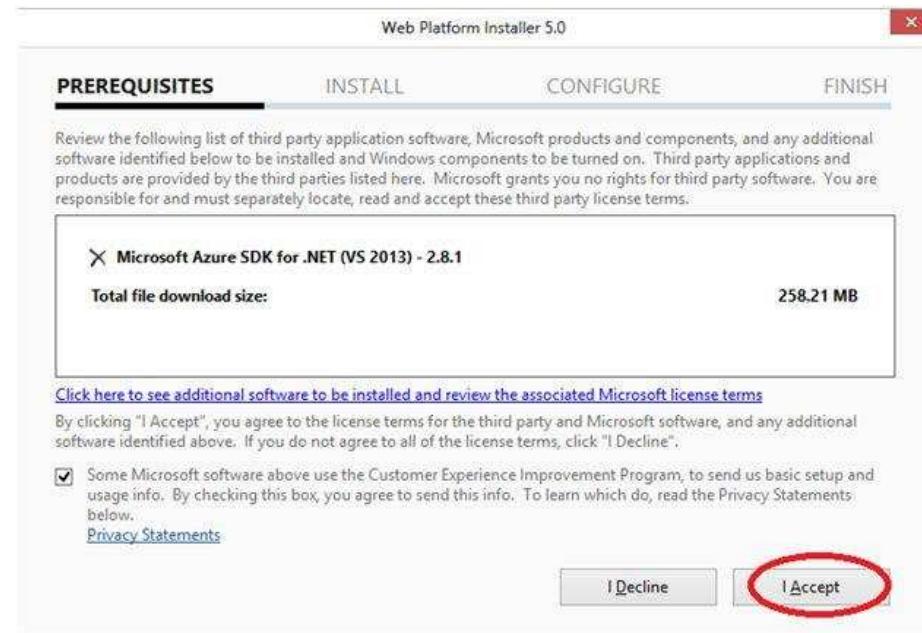
After that the installation will continue as in the following:



Now click “**Install**” button for further processing:



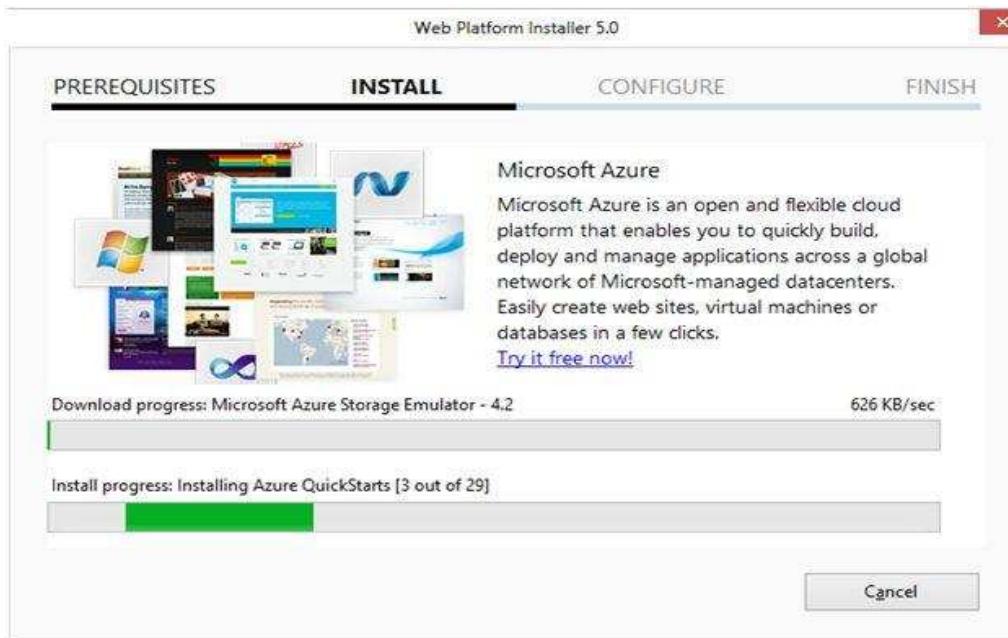
Now in the following figure, by clicking “**I Accept**” we agree to the license terms for the third party as in the following:



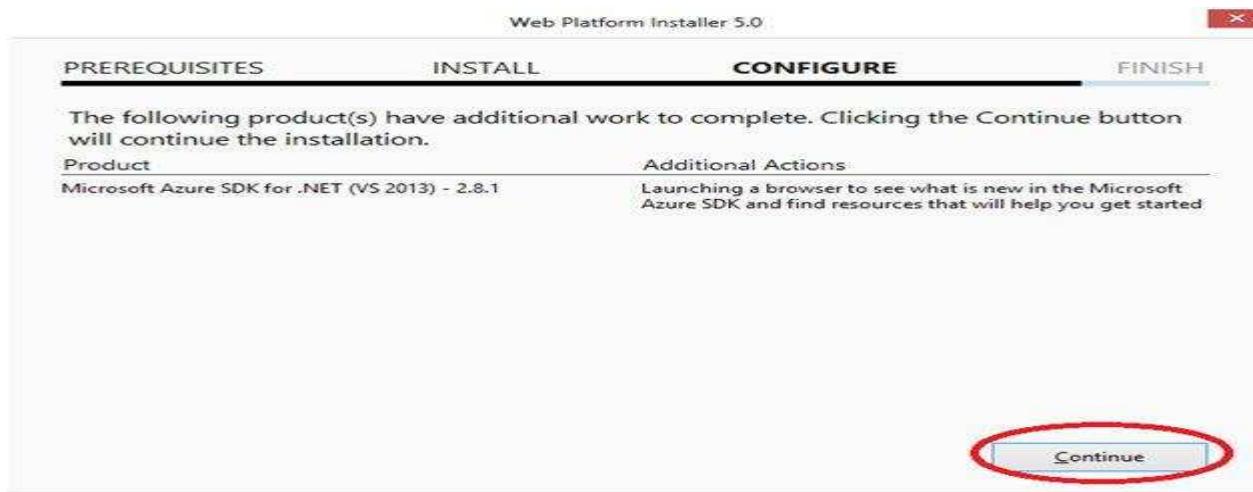
Click **OK** button to close your previous applications, which are running in the background:



Now Azure SDK is installing as in the following figure:

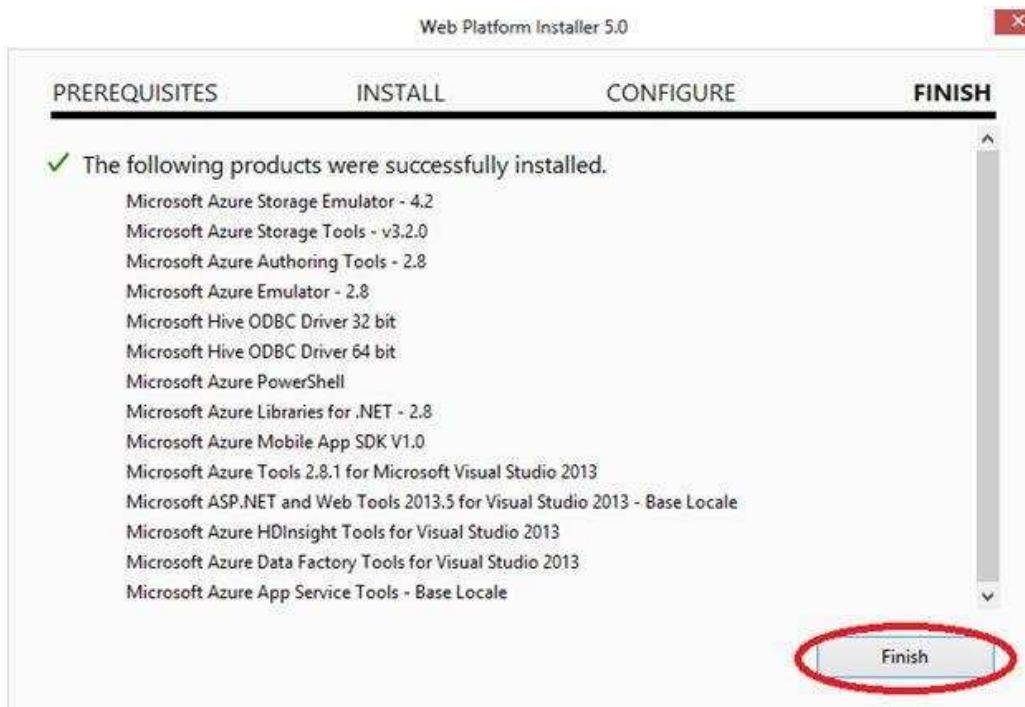


Click on the “**Continue**” button for further installation of SDK.



So, now our SDK is ready for use in Visual Studio 2013.

Click on to the “Finish” button to finish the installation of Azure SDK.

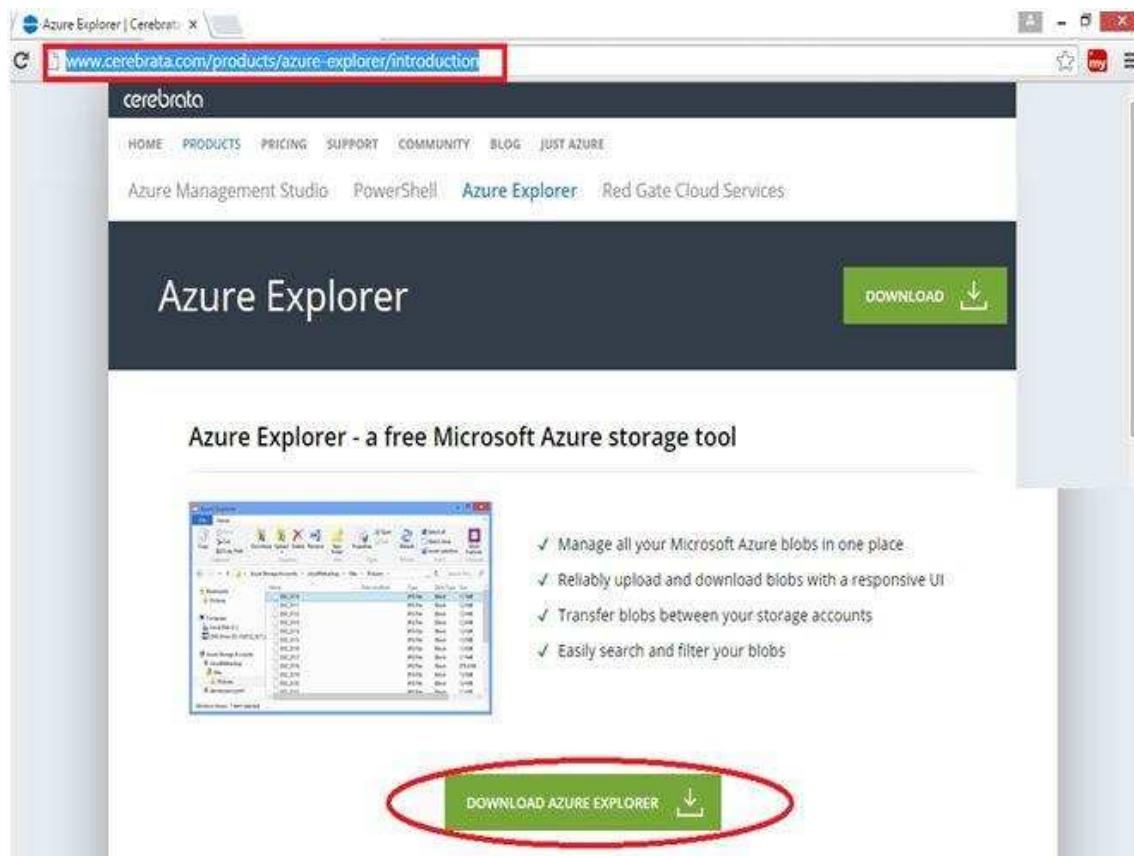


Now we are going to install Azure Explorer.

Azure Explorer: Azure Explorer is a free Microsoft Azure storage tool; in this we can manage all our Microsoft Azure blobs at a common place or one place. Azure Storage Explorer is a useful GUI tool for inspecting and altering the data.

Step 1: For the installation of Azure Explorer we need go to its download link,

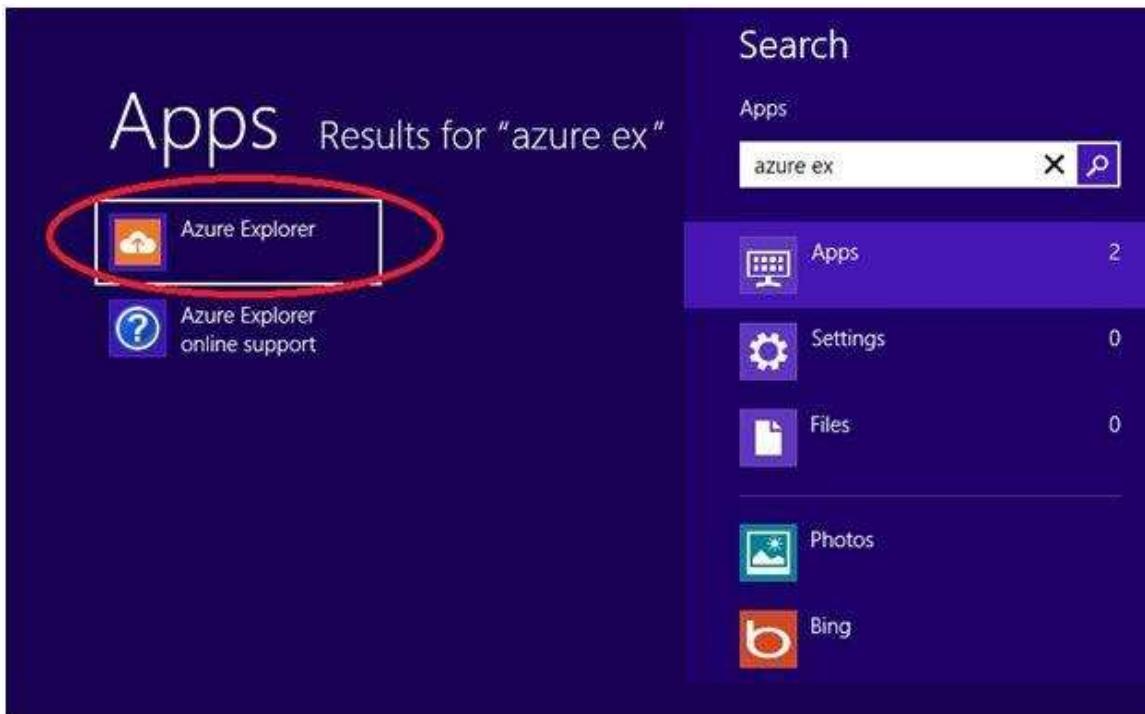
After going to above link, you will see the following window. Click “**DOWNLOAD AZURE EXPLORER**” (in red circle).



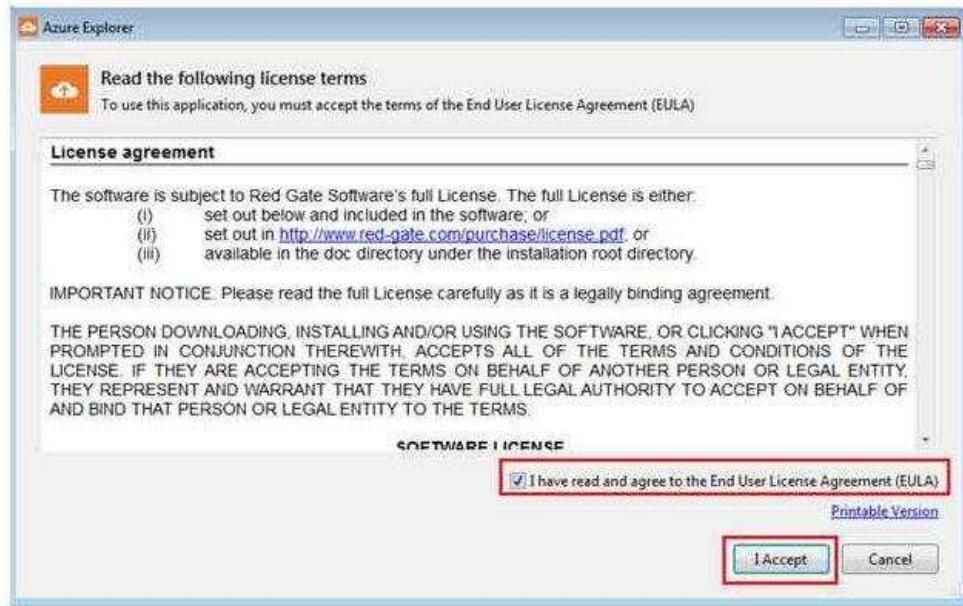
In the following figure the installation of Azure Explorer progress can be seen:



After completing the installation, just go in the search box and type Azure Explorer and open it.

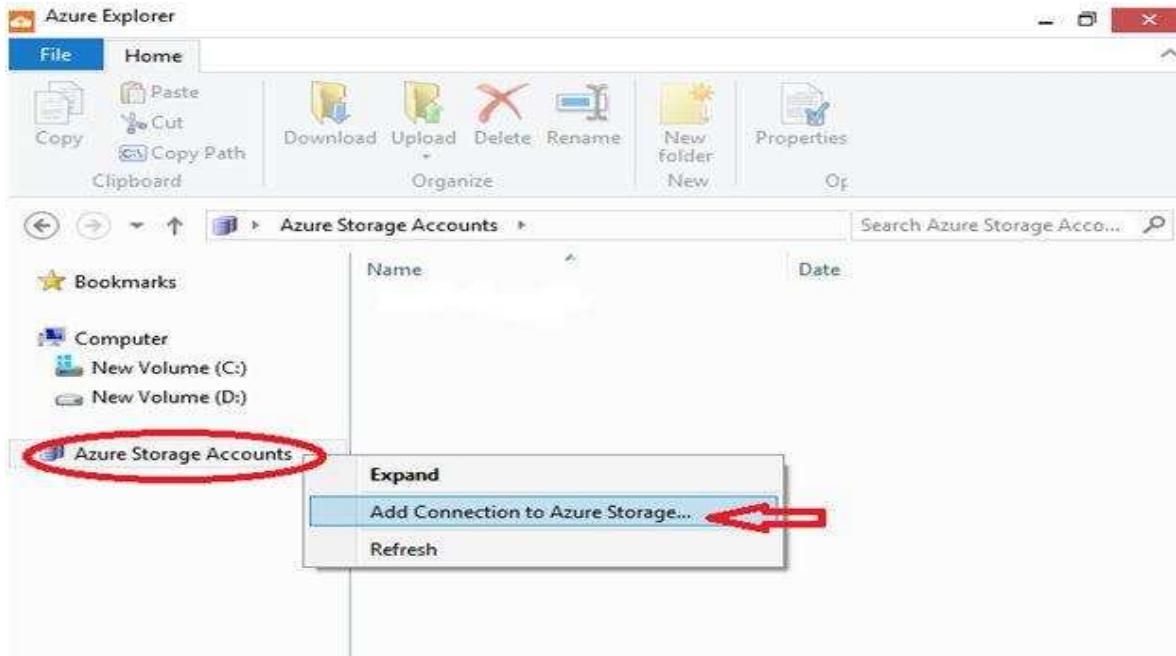


You need to check License Agreement, and click on “I Accept” button.



And finally we installed Azure Explorer successfully.

Here we need to add our account with the Azure Explorer, so right click on “Azure Storage Accounts” and choose “Add Connection to Azure Storage”.



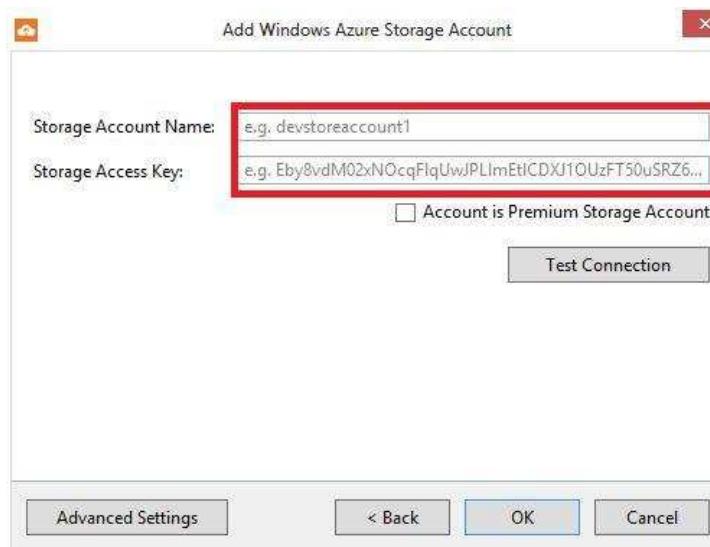
So in the following figure it will provide you different options to connect with the storage account, here we need to choose “**Connect to Azure with a Storage Account Name and an Access key**”.

Click on “**Next**” button.



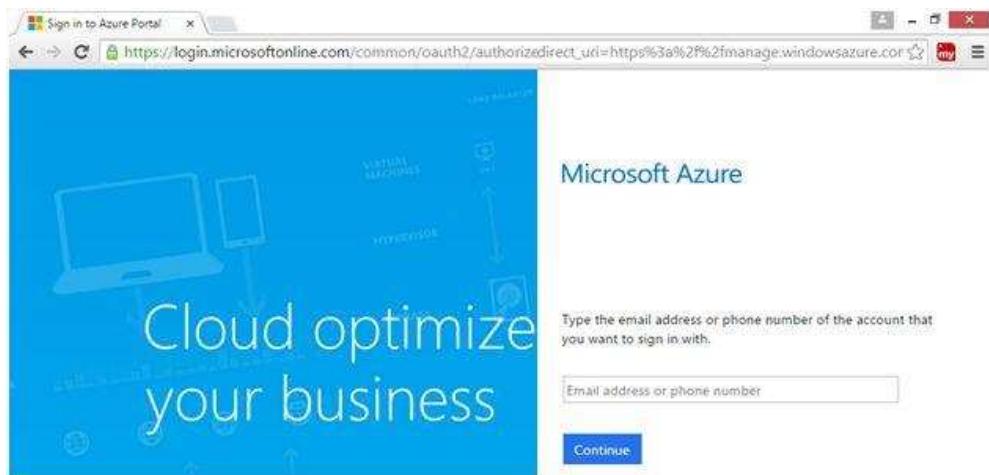
So, next window will ask you about “**Storage Account Name**” and “**Storage Access Key**”.

Provide the storage account name and storage access key. You will get this information from your Microsoft Azure Account.

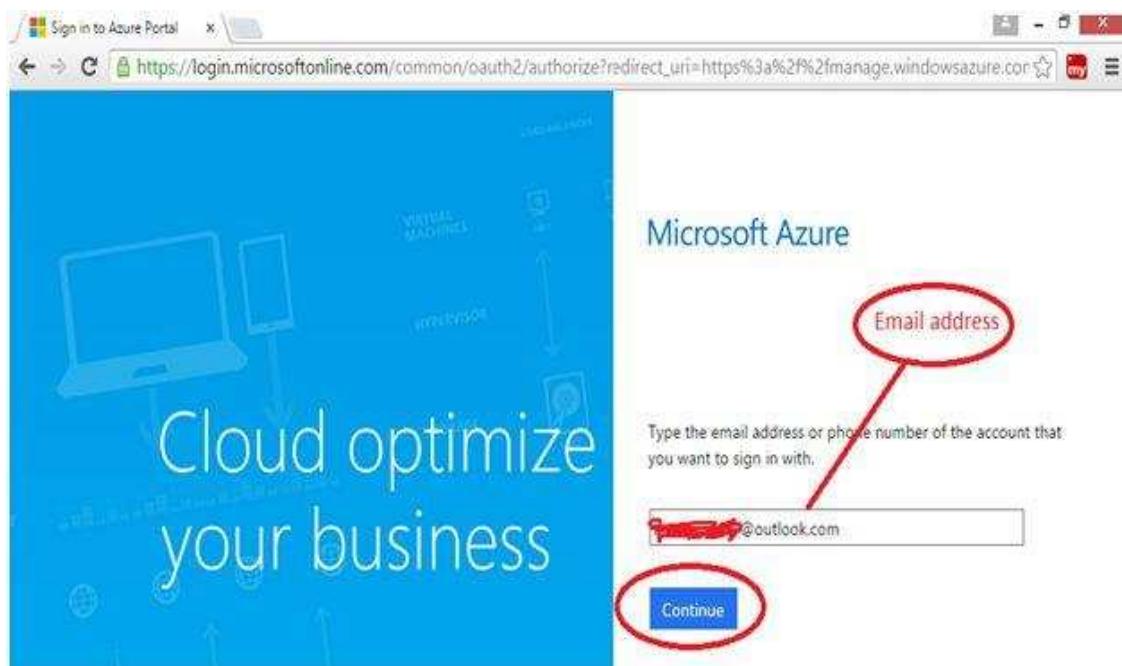


Just go to your Microsoft Account login where you can get the name and key from your storage account.

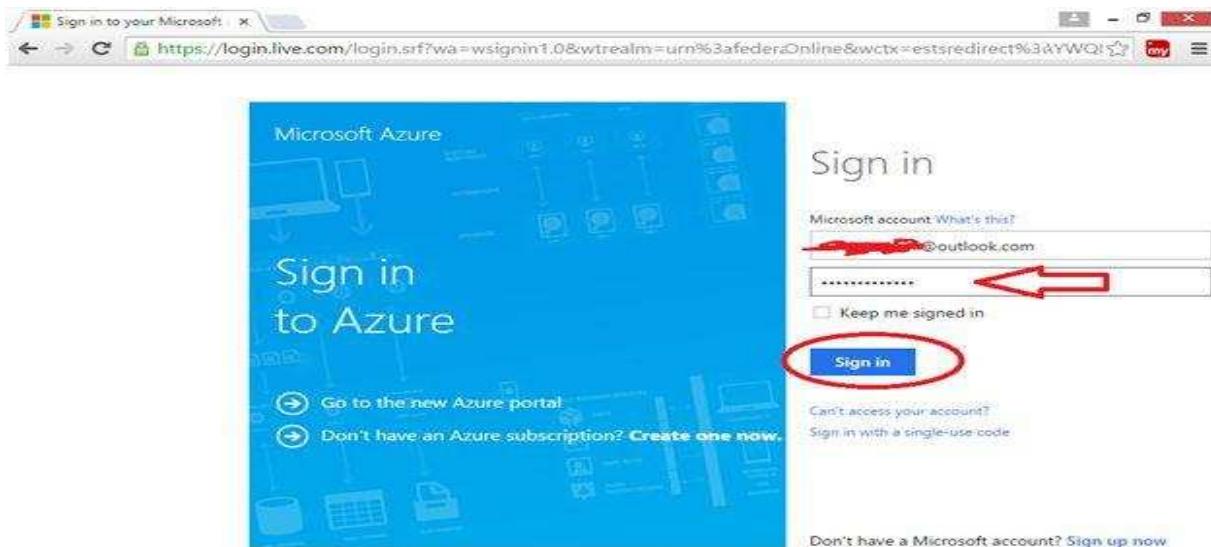
For this, simply go to your Microsoft login page. Here's the screenshot:



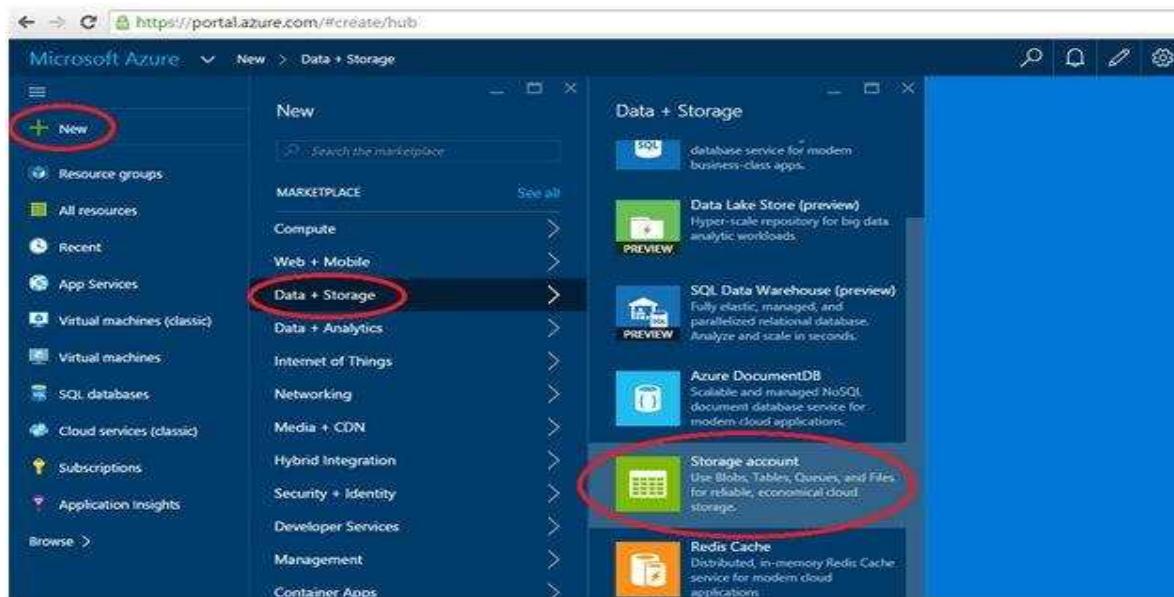
Fill the email address and click on "**Continue**:



Fill the password detail and click on “Sign in” as in the following figure:



So, for creating a storage account, you need to go to **New, Data + Storage**, then **Storage account**.



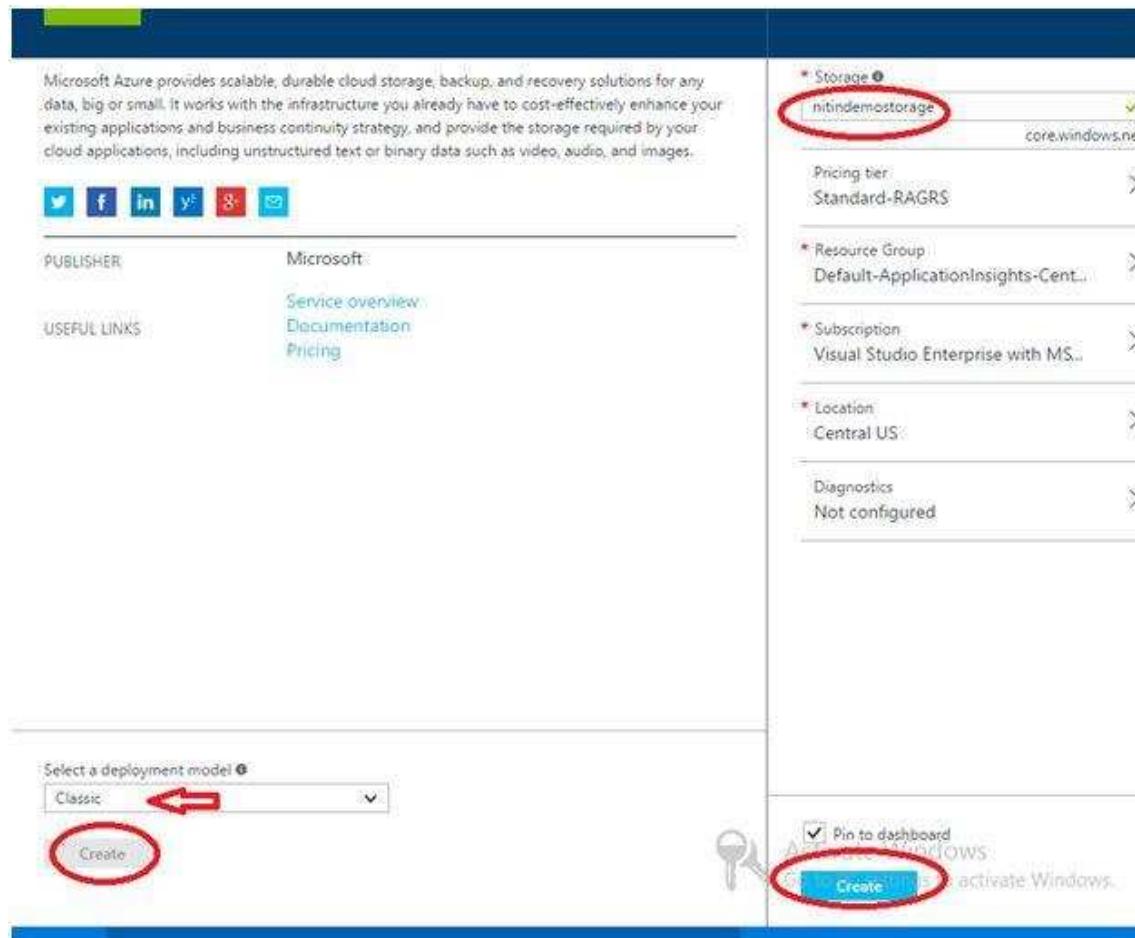
After clicking on the “**Storage account**”, you will see the following window.

Here, in the “**Select a deployment model**” (in the left side of the window), you need to select “**Classic**” from the dropdown items.

Click on the “**Create**” button. After clicking, you will see the right portion of the window.

In the **Storage** option, you need to give your storage account name; here I am giving the storage name “**nitindemostorage**”, you can give the storage name as your wish.

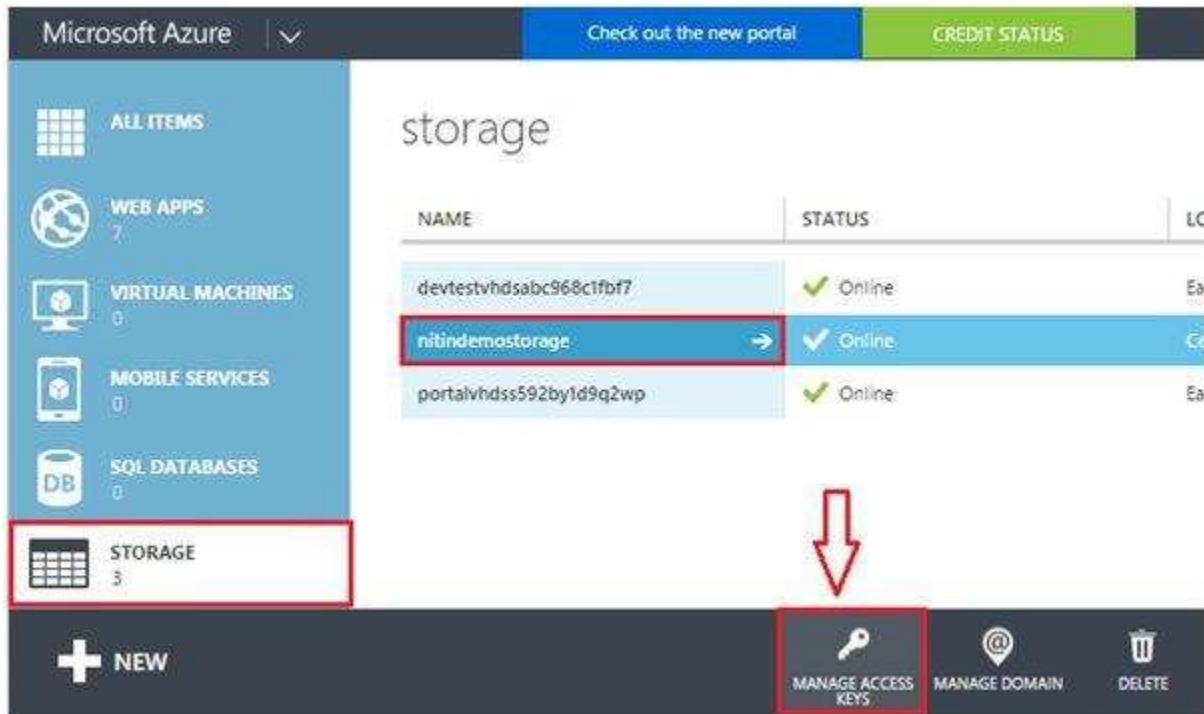
Finally, click on the “**Create**” button.



The screenshot shows the Azure Storage account creation interface. On the left, there's a summary of Azure storage benefits and links to service overview, documentation, and pricing. The main area is titled "Create a new storage account". It includes fields for "Storage account name" (nitindemostorage), "Pricing tier" (Standard-RAGRS), "Resource Group" (Default-ApplicationInsights-Cent...), "Subscription" (Visual Studio Enterprise with MS...), "Location" (Central US), and "Diagnostics" (Not configured). At the bottom, there's a "Select a deployment model" dropdown set to "Classic" (with a red arrow pointing to it) and a large blue "Create" button (also circled with a red circle).

After clicking the “**Create**” button, you will get the storage account of your name.

And now, simply go to the “**Storage**” and then “**nitindemostorage**”, which is my storage account name and then go to the “**MANAGE ACCESS KEYS**”, for the Storage Account Name and Primary Access Key.



NAME	STATUS	LO
devtestvhdsabc968c1fbf7	✓ Online	Ea;
nitindemostorage	✓ Online	Ce
portalvhdss592by1d9q2wp	✓ Online	Ea;

Now you will need to copy the Storage Account Name and Primary Access Key as in the following screenshot:



STORAGE ACCOUNT NAME

nitindemostorage

PRIMARY ACCESS KEY

8TeQ8Li/b5U75aSPeTUvmg6w6s6emWSC

SECONDARY ACCESS KEY

PjYyg5lvGMCe9Vk6BnZu99FeTJVob7ckWi

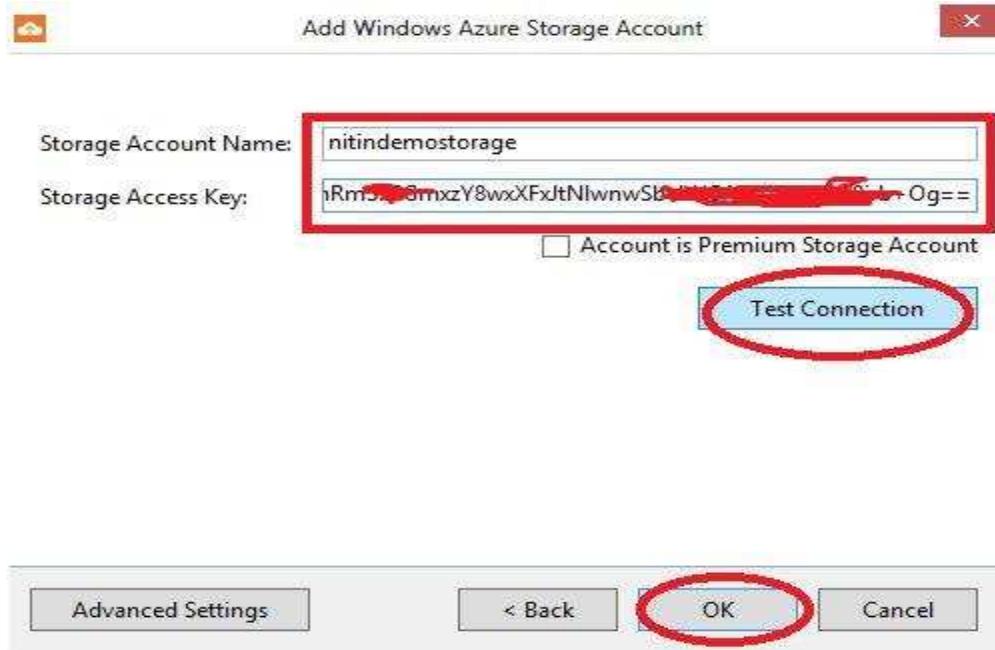
PRIMARY CONNECTION STRING

DefaultEndpointsProtocol=https;AccountName=

SECONDARY CONNECTION STRING

DefaultEndpointsProtocol=https;AccountName=

Now, copy the storage account name and primary access key, and paste over here as in the following screenshot. After that, click on the “**Test Connection**” button.



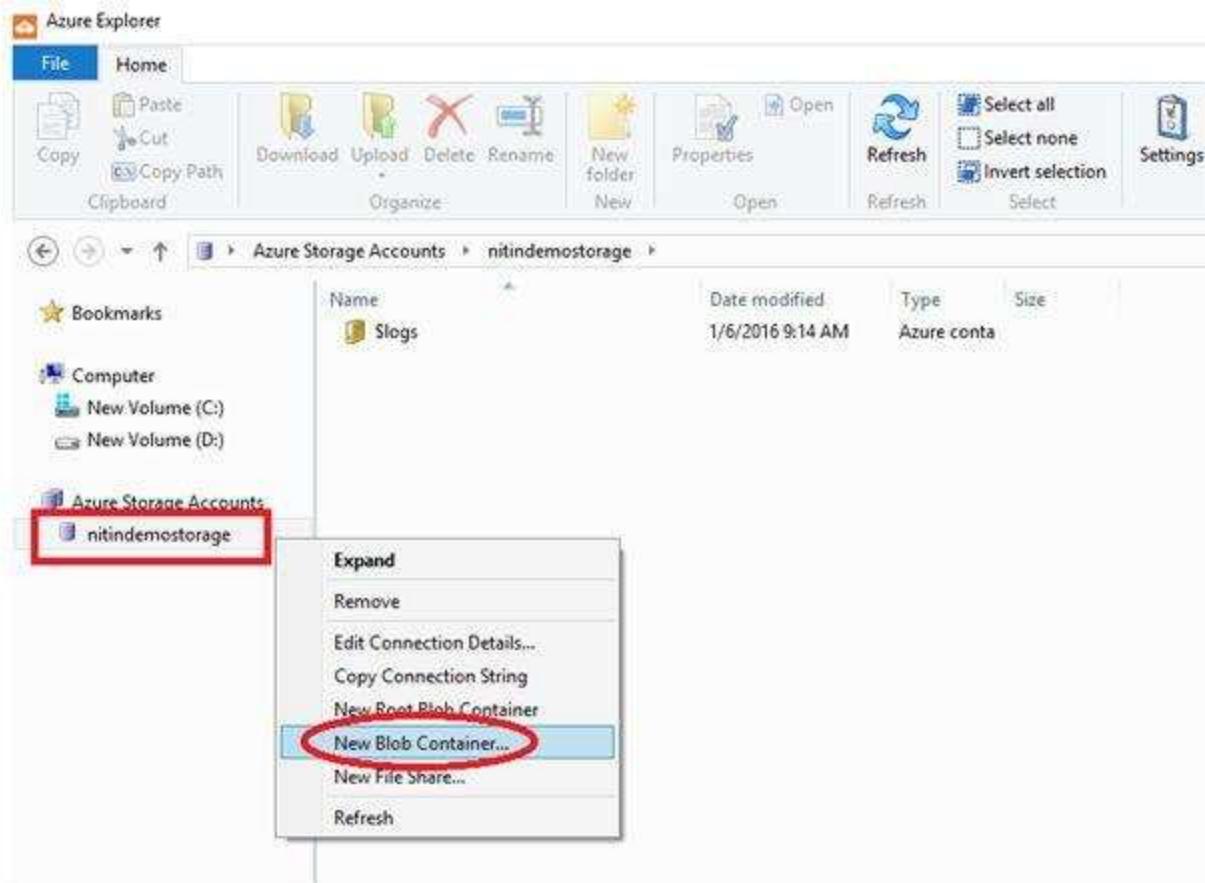
Connection is successful, so click on **OK** button.



Once again click on OK button to proceed.

Finally, we have created our storage account with Azure Explorer. You can see in the red rectangle that is “**nitindemostorage**”.

Now we have to create blob storage for accessing services in Azure Storage, so right click on storage account and click on “**New Blob Container**”.



Now in the name section, give the name to blob container as you wish, but you can't use uppercase letters while creating blob storage.

Here I have given “**mysample**” name to my blob container.

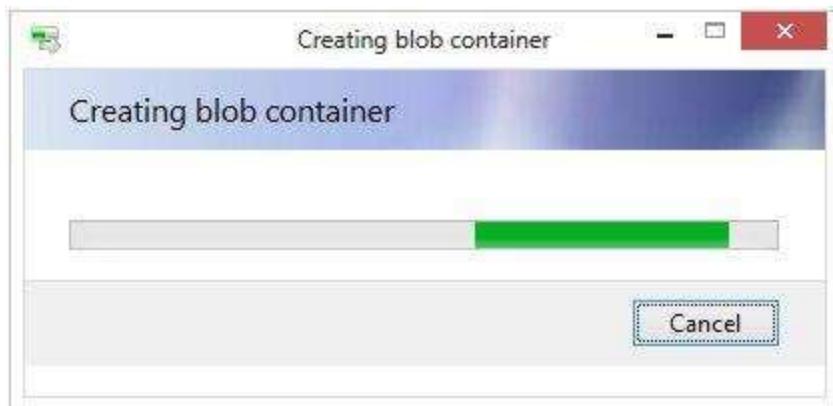
Now, if you are an owner of your account, choose the “**Private- Access to account owner only**” option from the dropdown list.



Click on **OK** button.

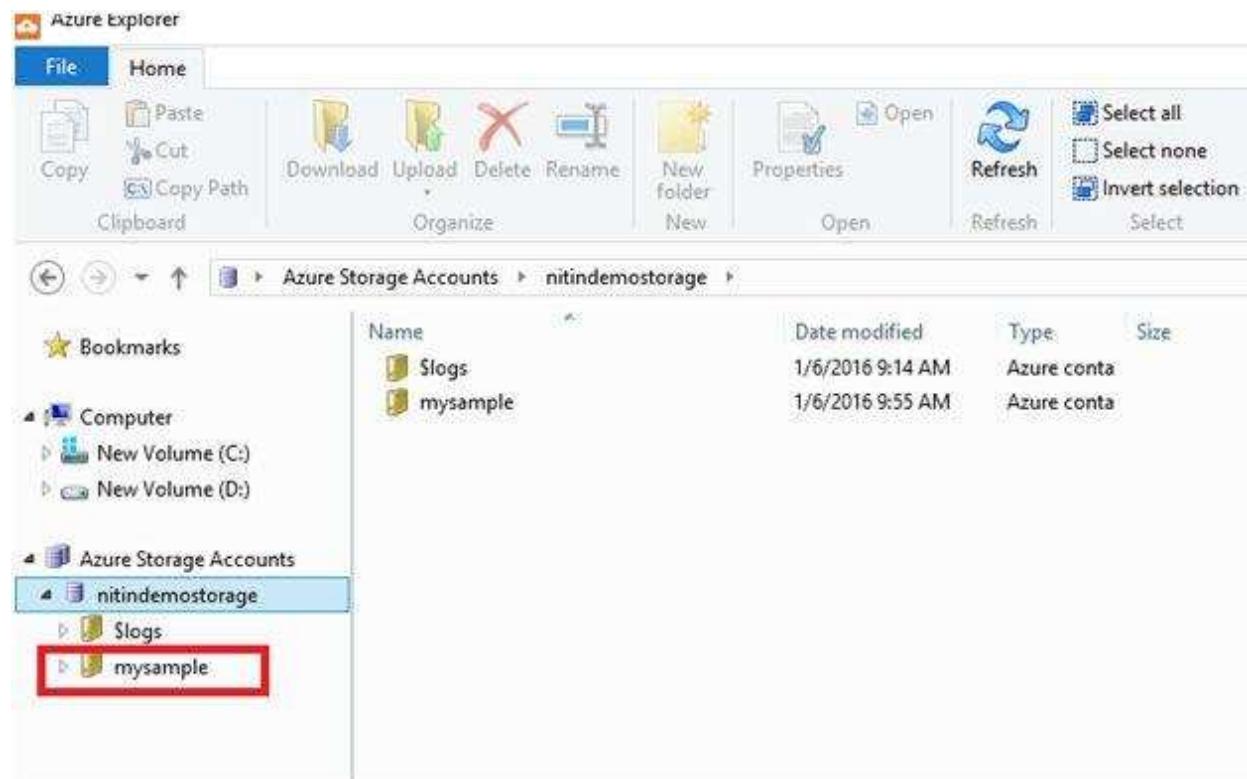


This will create a blob container.



In the following figure, “**mysample**” blob container successfully created. (In the red area).

Here is another blob with name “**\$logs**”, so don’t worry, since it is created by Azure Storage by default. We will focus on “**mysample**” named blob storage.

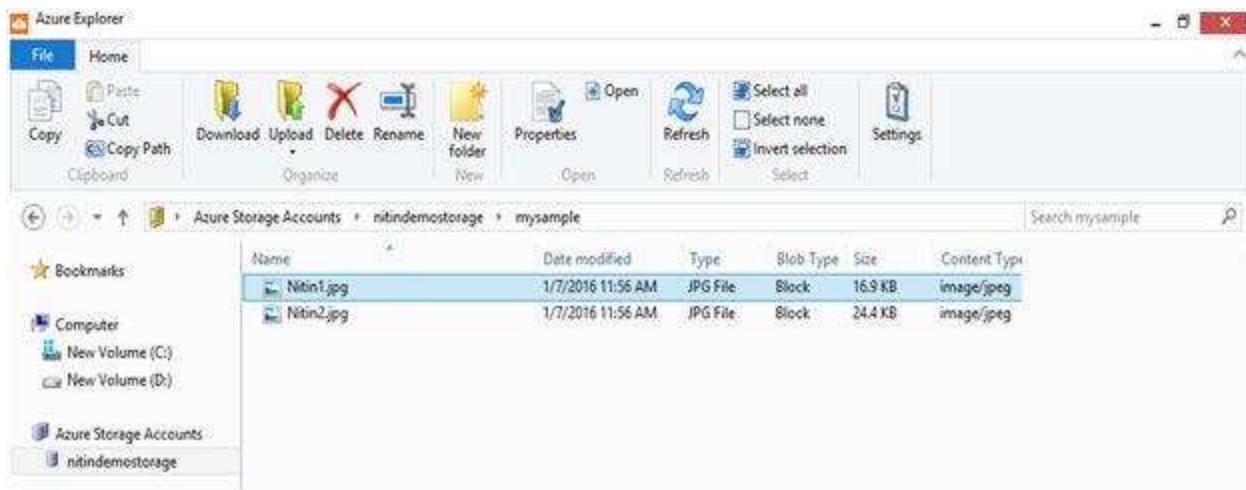


The screenshot shows the Azure Explorer interface. The ribbon bar at the top has 'File' and 'Home' tabs selected. Below the ribbon are various file operations: Copy, Paste, Cut, Copy Path, Clipboard; Download, Upload, Delete, Rename, Organize; New folder, New; Open Properties, Open; Refresh, Select all, Select none, Invert selection. The navigation bar below the ribbon shows 'Azure Storage Accounts > nitindemostorage'. The left sidebar lists 'Bookmarks', 'Computer' (with 'New Volume (C:)' and 'New Volume (D:)'), and 'Azure Storage Accounts' (with 'nitindemostorage' selected). Under 'nitindemostorage', there are two items: 'Slogs' and 'mysample', both listed under 'Name', 'Date modified' (1/6/2016 9:14 AM and 1/6/2016 9:55 AM), 'Type' (Azure conta), and 'Size'. The 'mysample' item is highlighted with a red box. The main pane displays the same list of items.

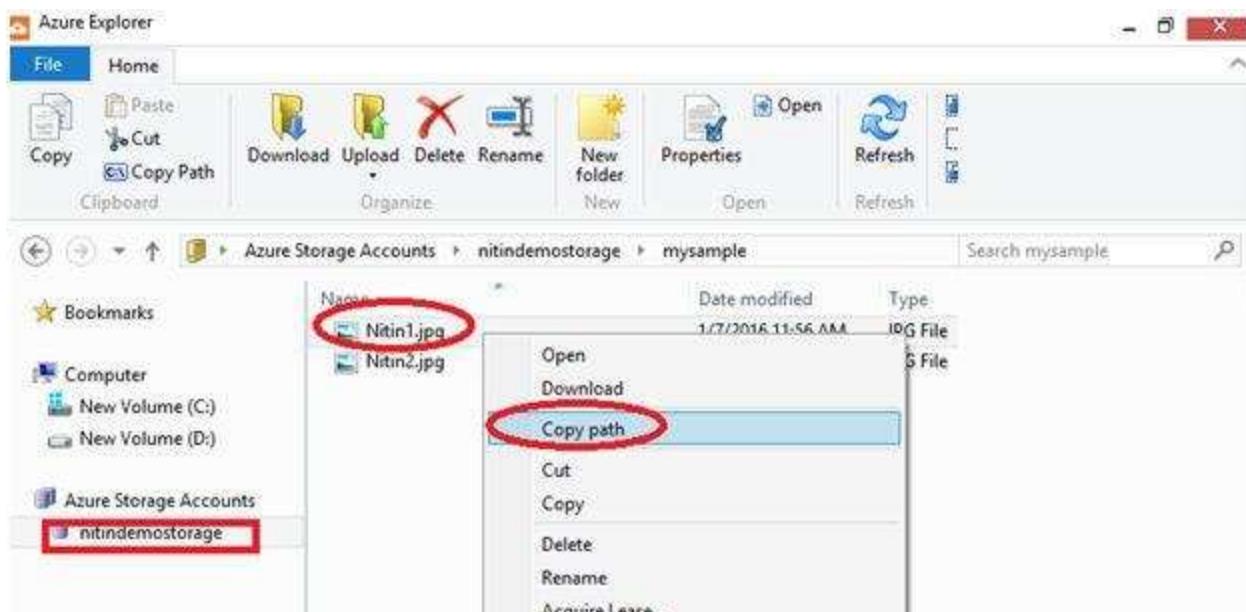
So, we have successfully created our “**mysample**” named blob storage.

Now, I am going to store images in my Azure explorer. You can directly paste the image or any file inside the “**mysample**” blob.

Here, I am going to paste two images: “**Nitin1**” and “**Nitin2**” as in the following figure:



Right click on your favorite image and click "**Copy path**".

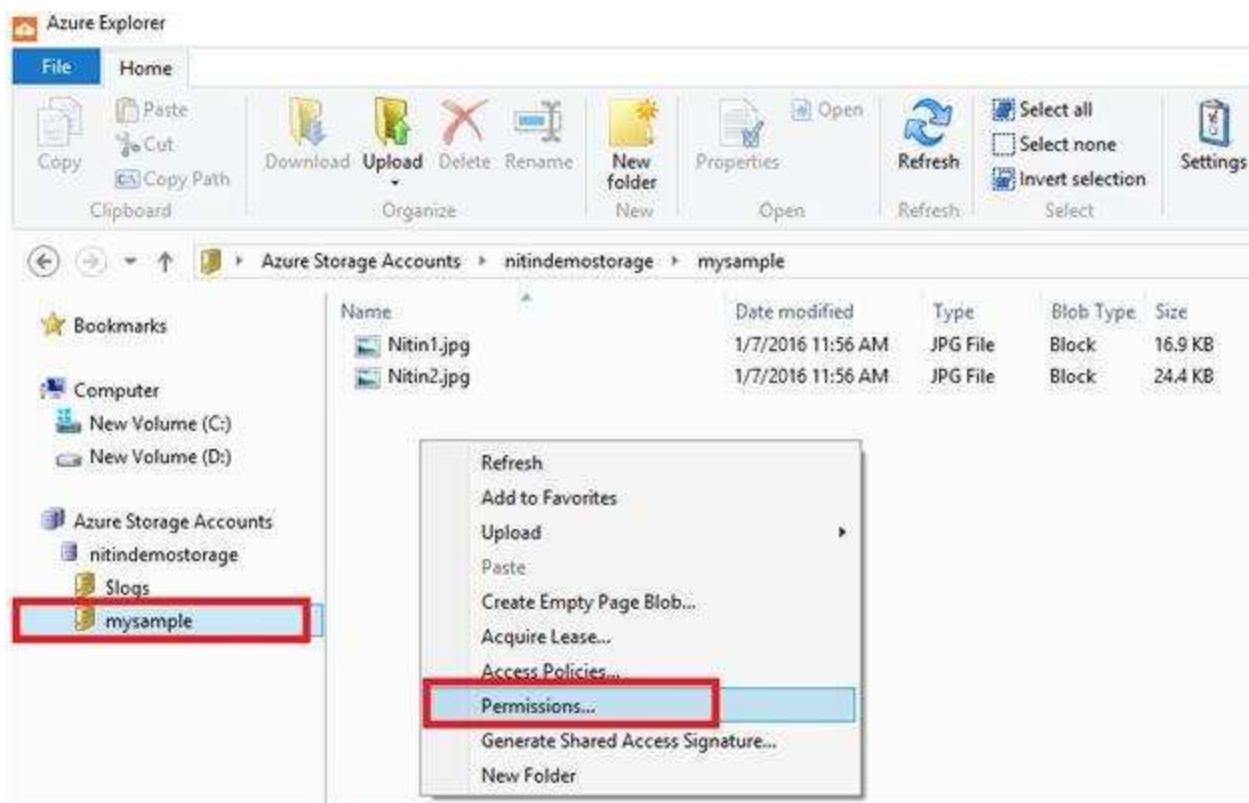


Now you can paste the path in our browser directly.

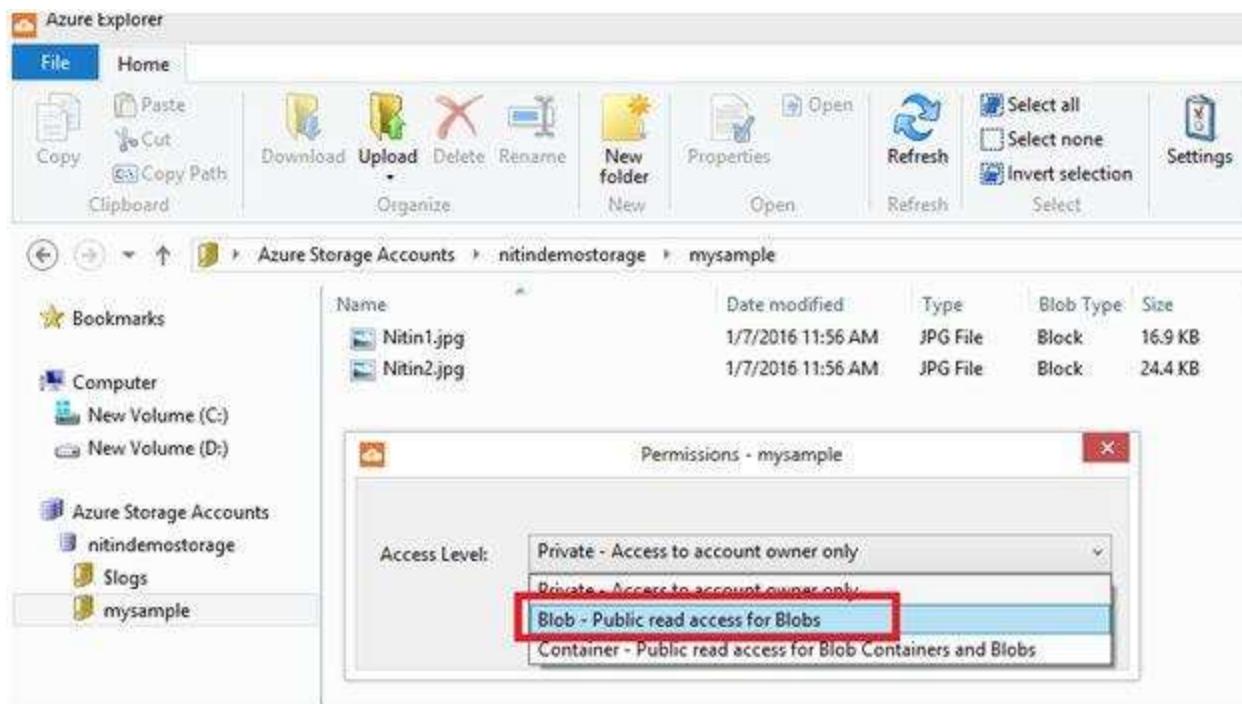


It gives an error because we need to change the access level of our blob container.

So, right click and choose “**Permissions**”.



Here you can choose the “**Blob - Public read access for Blobs**” option.

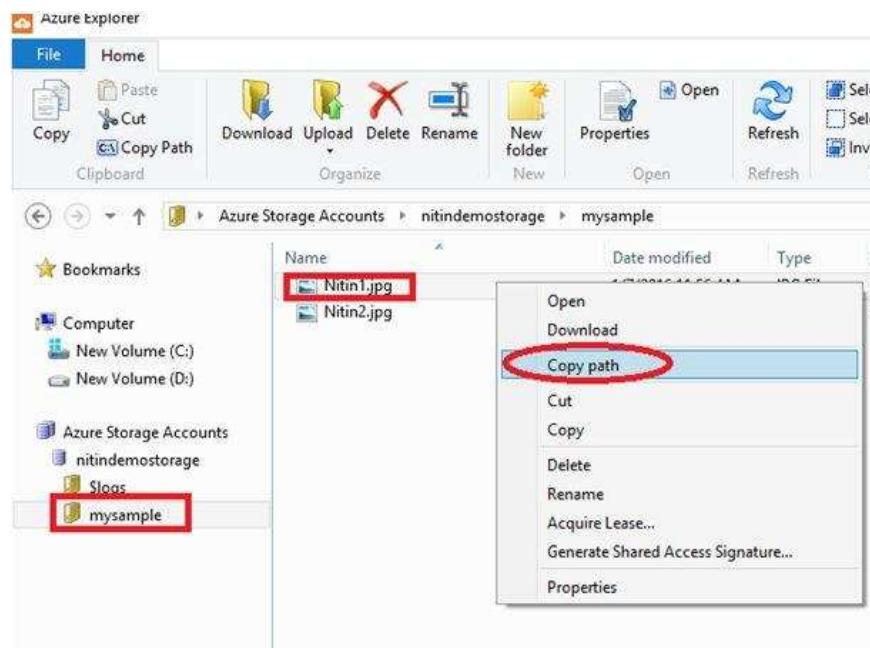


The screenshot shows the Azure Explorer interface. On the left, there's a sidebar with Bookmarks, Computer (New Volume C:, New Volume D:), and Azure Storage Accounts (nitudemostorage, Slogs, mysample). The main area shows a list of files in the 'mysample' container: Nitin1.jpg and Nitin2.jpg. Below the file list is a 'Permissions - mysample' dialog. The 'Access Level:' dropdown is set to 'Private - Access to account owner only'. A red box highlights the 'Blob - Public read access for Blobs' option, which is the second item in the dropdown list.

Click on **OK** button.



Now, again right click on the image and copy full path of the image.



Paste the copied path in the web browser.

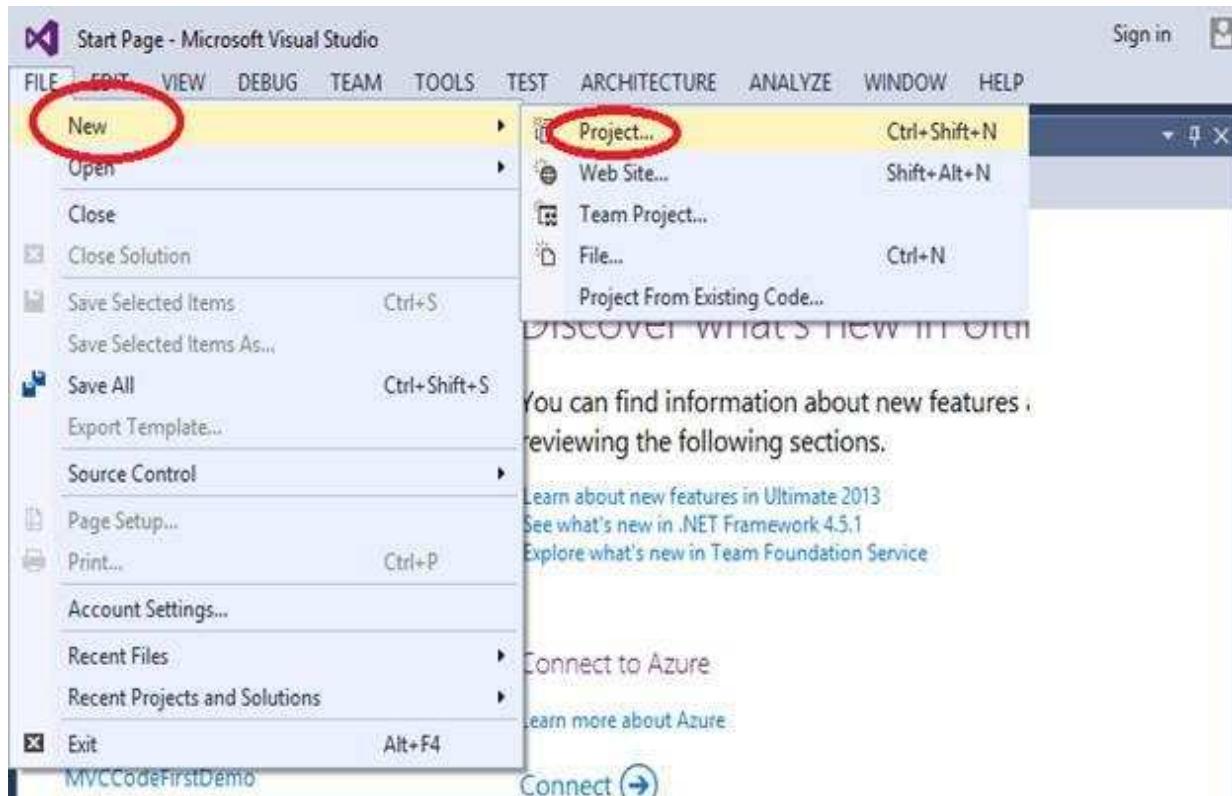
So, finally we will get our image.



Upload Image to Azure Blob Storage in ASP.NET

In this chapter, I'll tell you the uses of Azure Blob Storage and how to use blob storage in ASP.NET Project. Firstly, create an ASP.NET project in Visual Studio, so open Visual Studio and follow these steps:

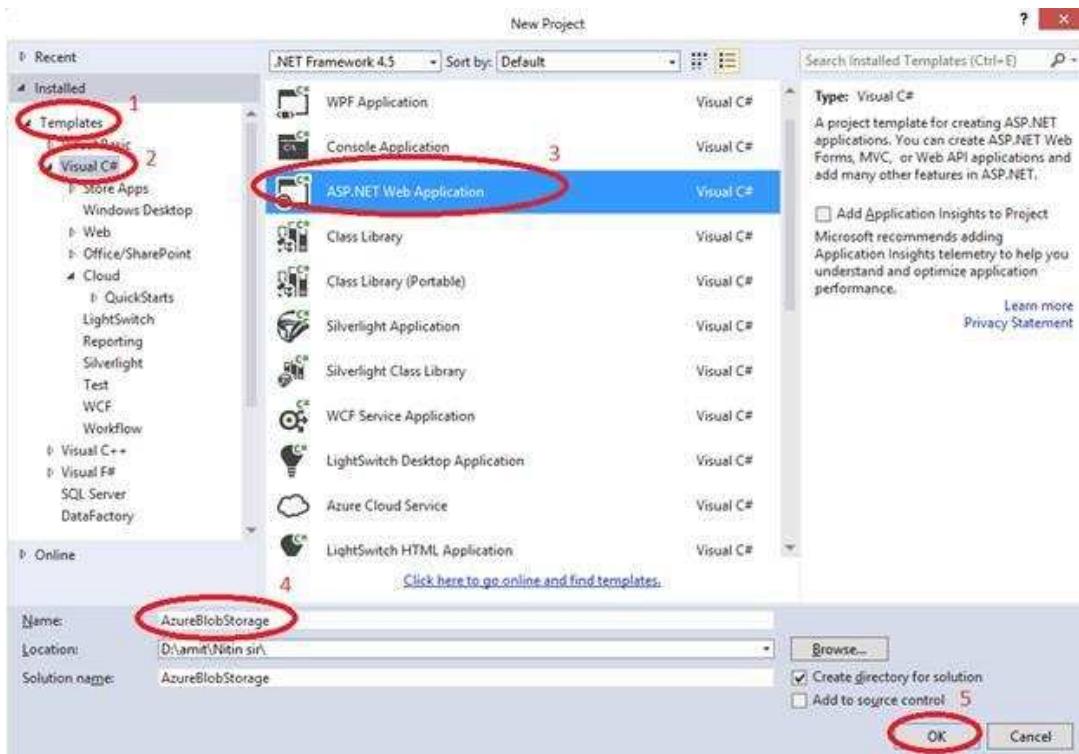
Step 1: Open Visual Studio and select the **File** menu, **New**, then **Project**.



Step 2: Go to **Templates**, **Visual C#**, and then **ASP.NET Web Application**.

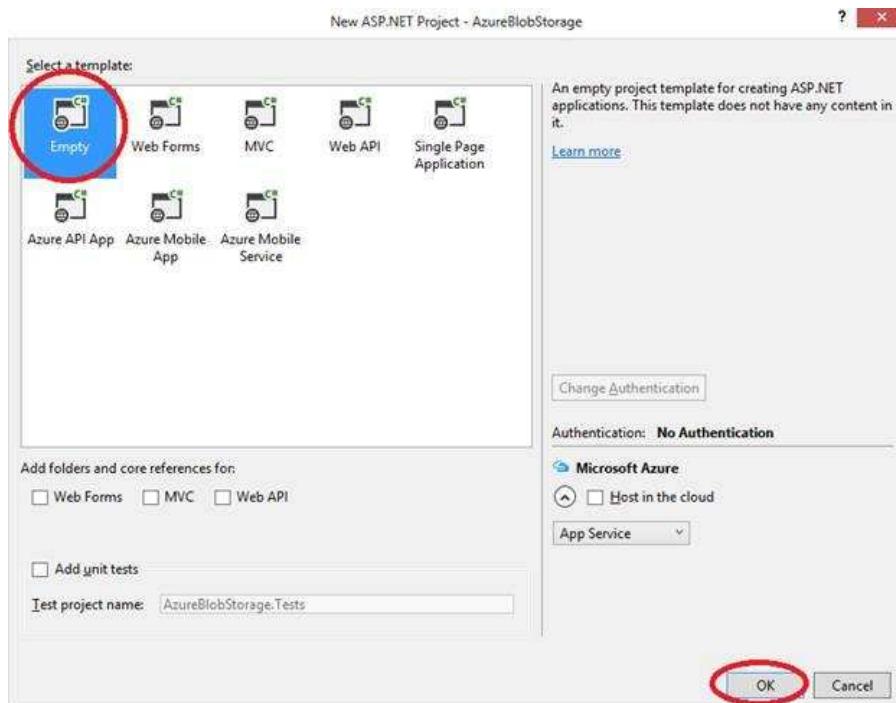
Step 3: In the name section, give a name to project. Here I have given the name “**Azure Blob Storage**”.

Step 4: Click on “**OK**” button to create a project.

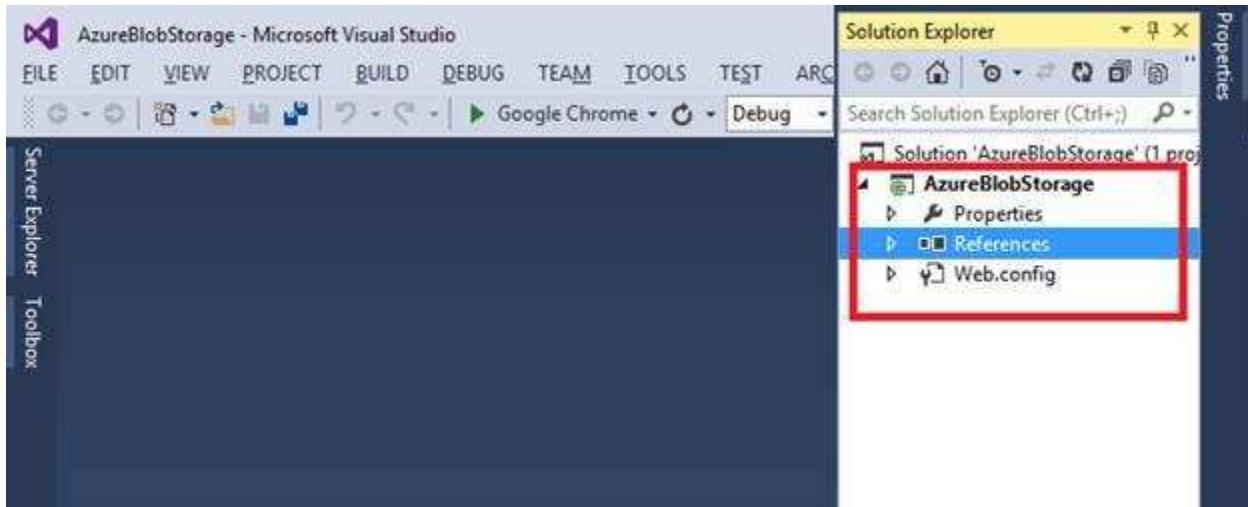


In the following figure, choose "**Empty**" project.

Click "**OK**" button.

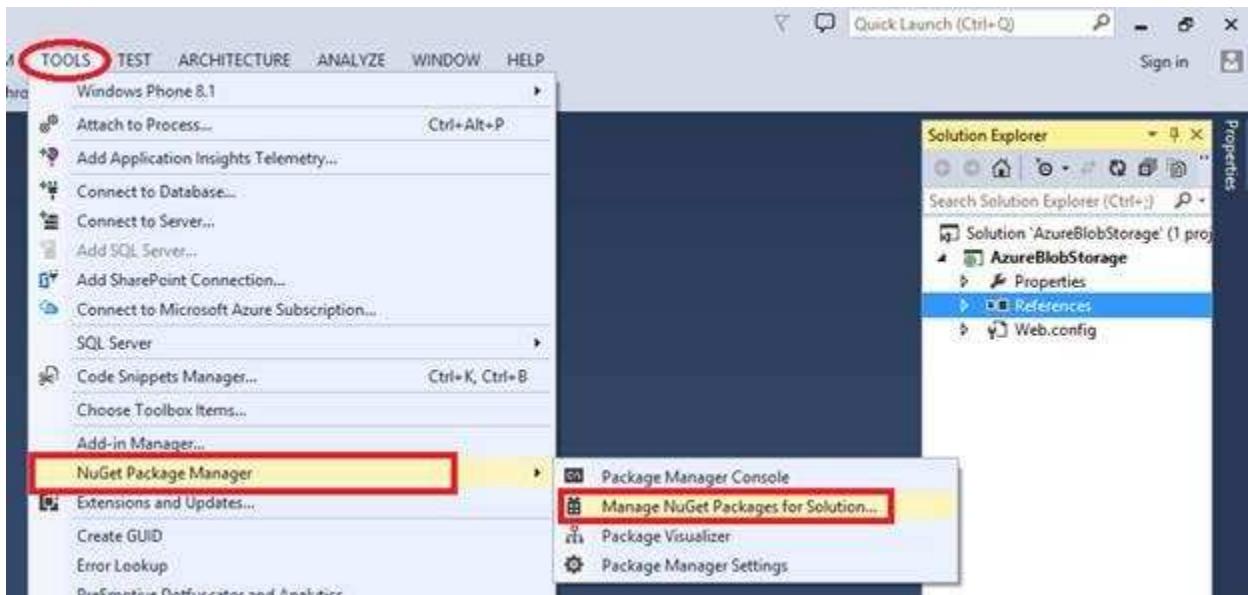


We can see, in our project, there are three types of categories items, properties, References, and Web.config.

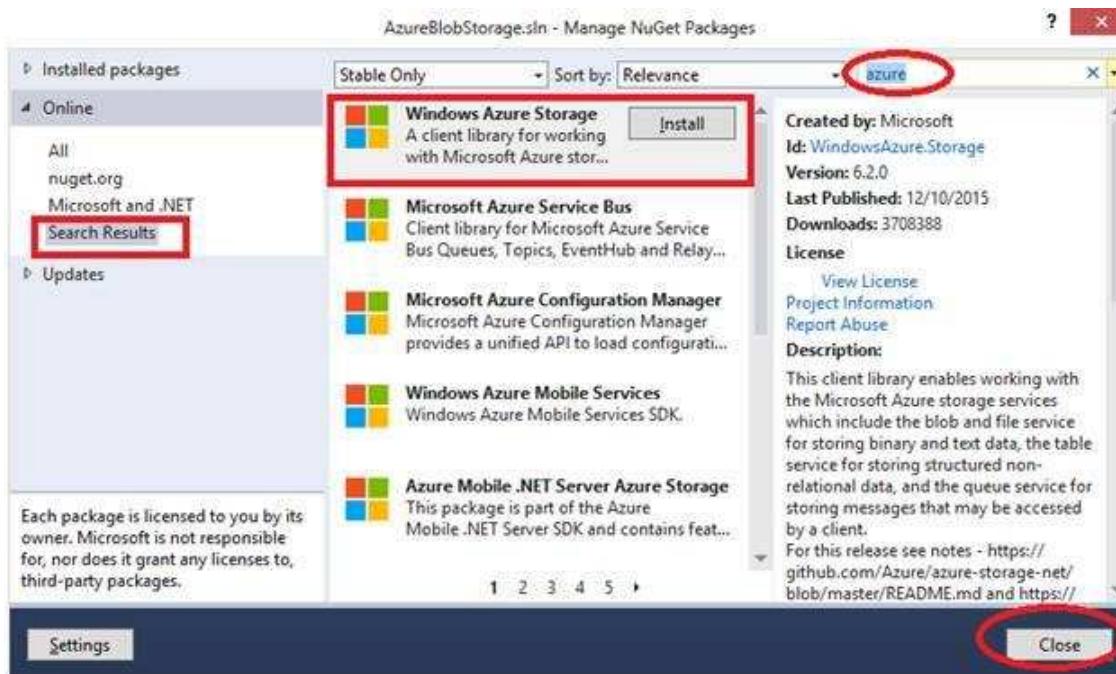


Now we need to install the Microsoft Azure Storage package for using Azure Storage Access and all supported classes.

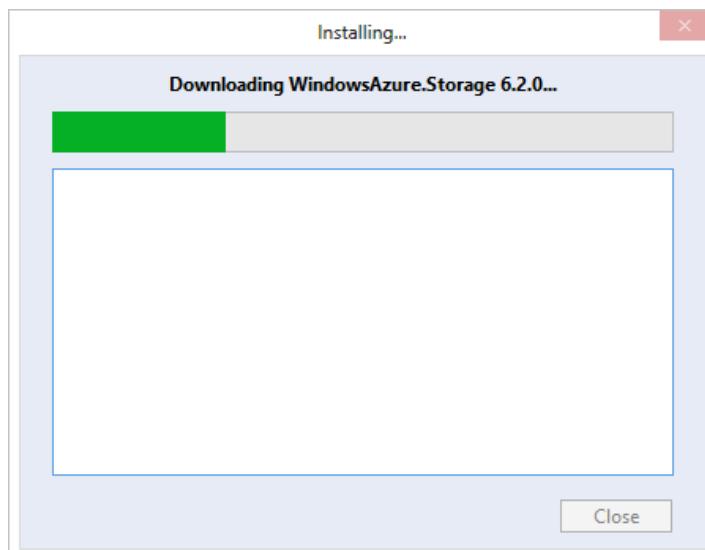
Now go to the “**TOOLS**” menu, select “**NuGet Package Manager**”, and select “**Manage NuGet Packages for Solution**”.



In the following figure, go to the search box and type “**Azure**”. After this you will find the “**Microsoft Azure Storage**”, click on “**Install**” button.

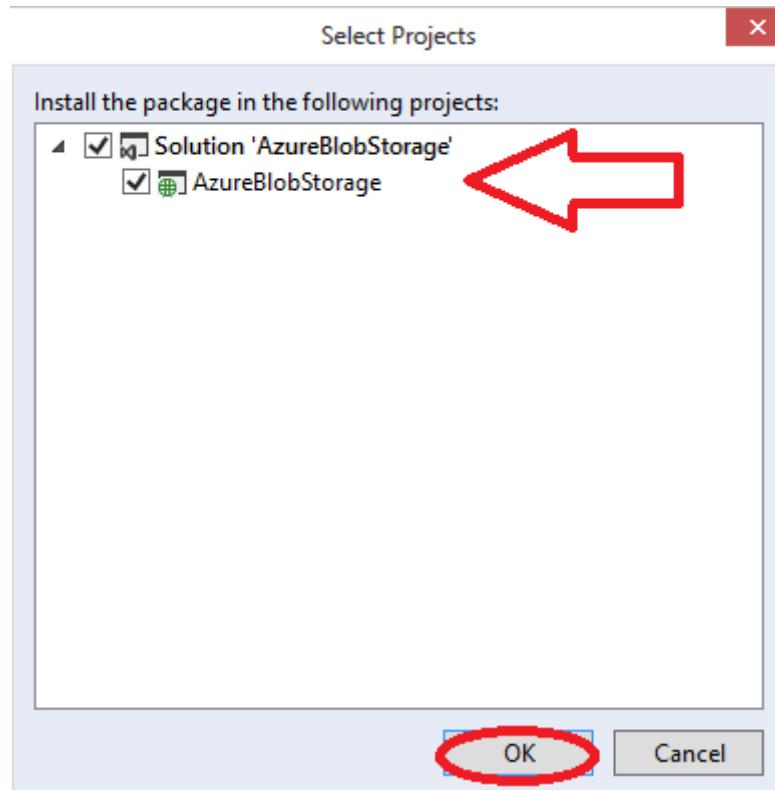


Now in the following figure, you can see the installation of the “Microsoft Azure Storage”.

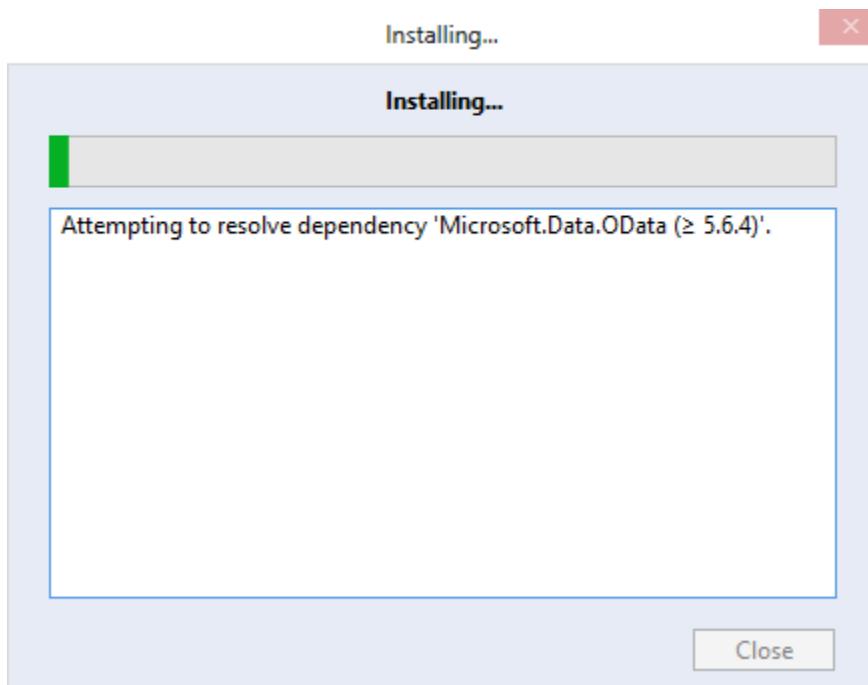


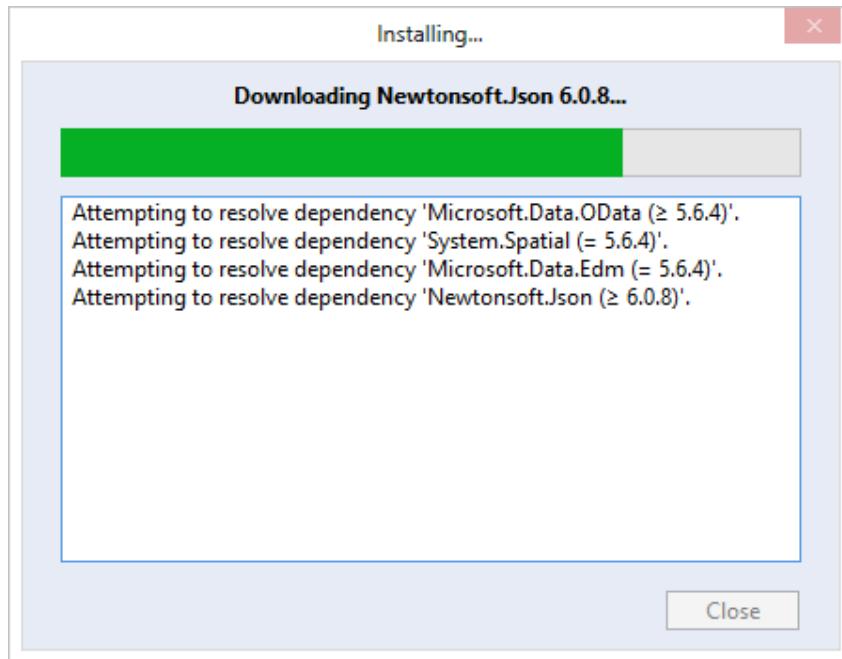
Here in the following window, you will be asked to install “Microsoft Azure Storage” for the current project, so here we want to install the packages for current project.

Simply click on “OK” button.

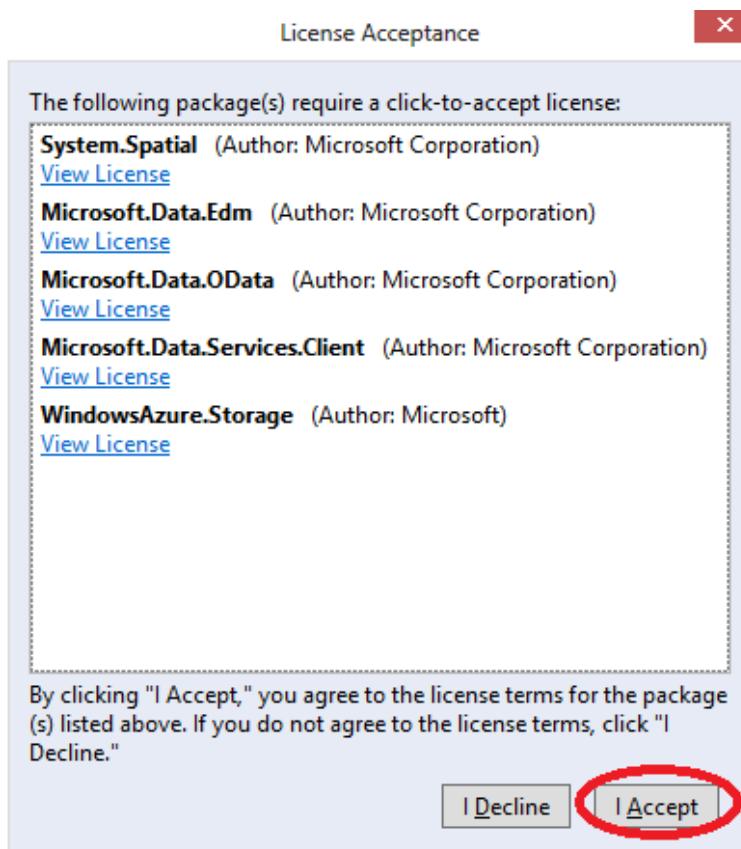


The installing process is going on.



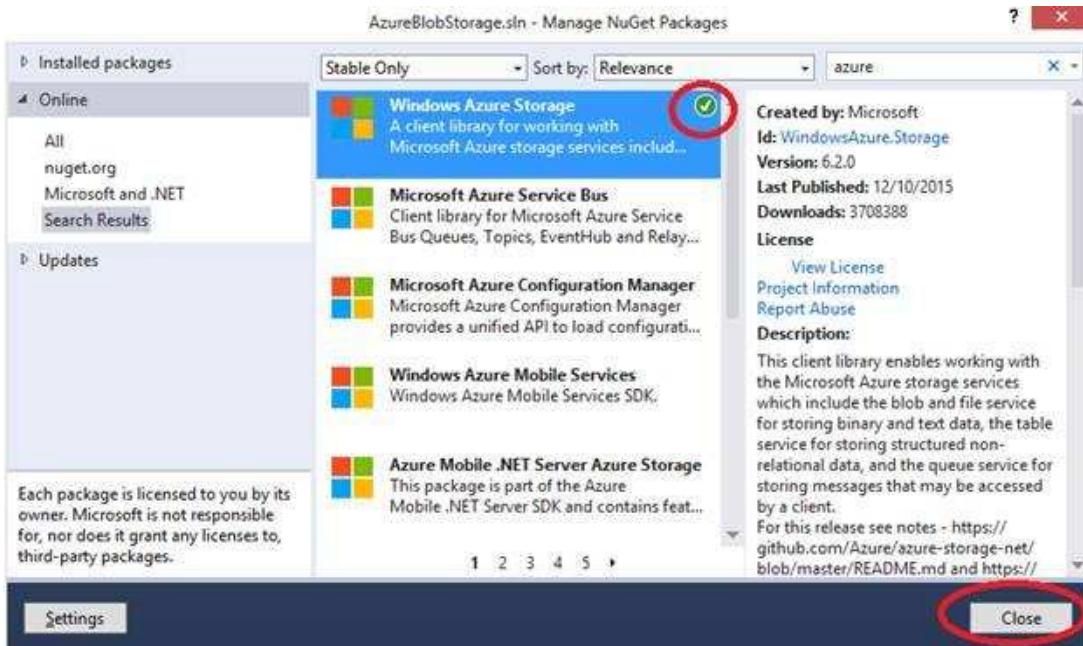


Here in the following figure, click on “I Accept” button.

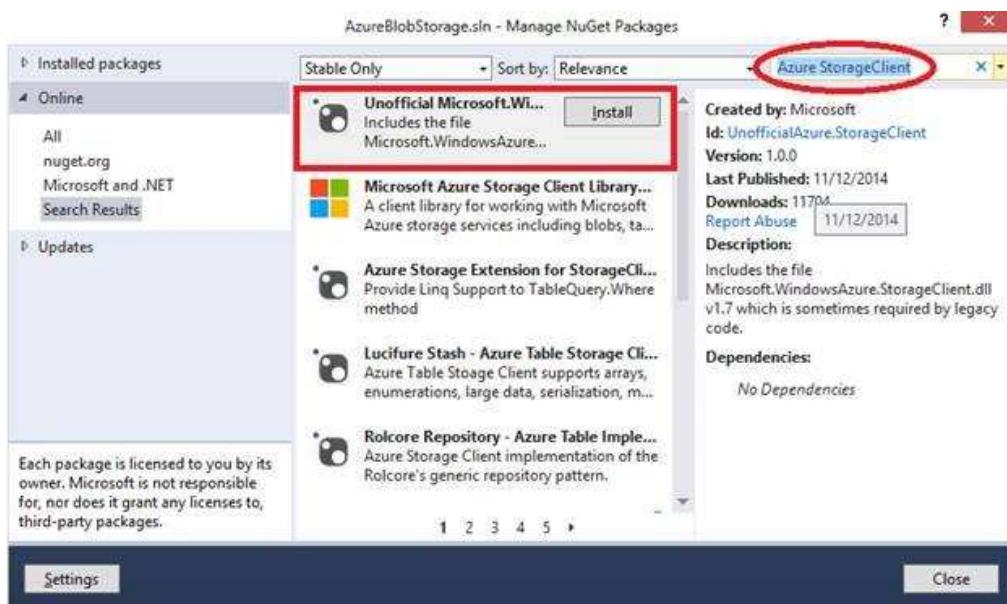


Now we successfully installed the “Microsoft Azure Storage” (in the small red circle).

Click on “Close” button.

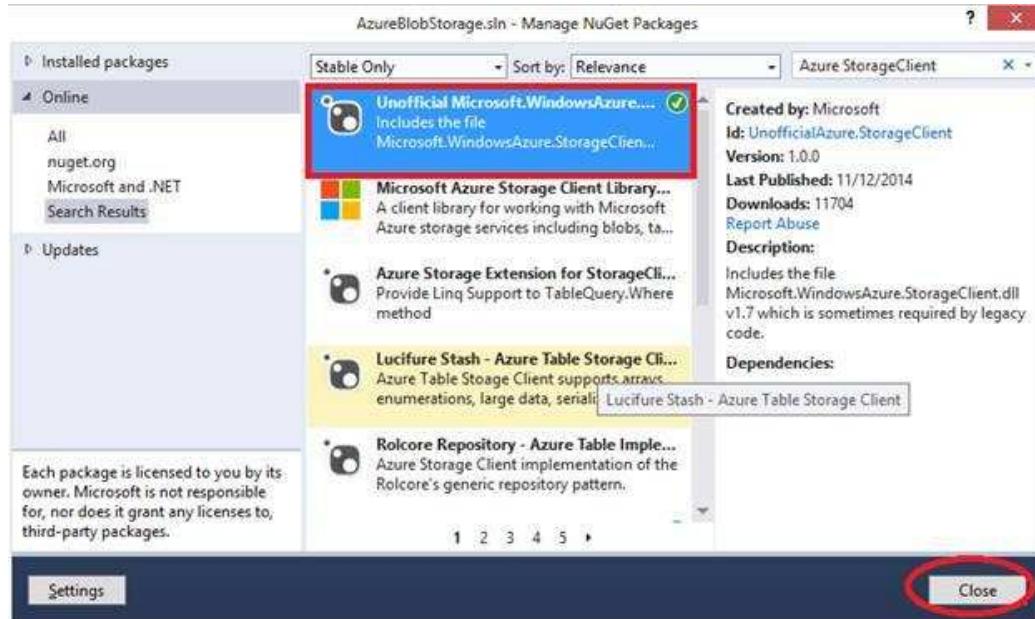


Here we also need to install the Azure storage client packages. So go to the search box, type the “Azure Storage Client”. After typing, click on “Install” button.



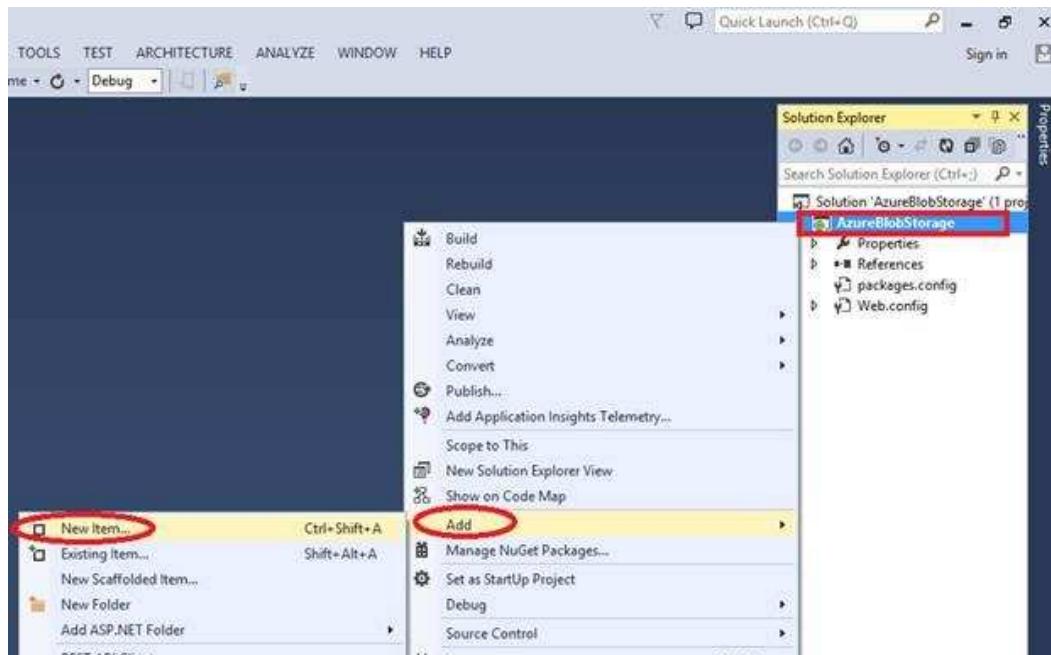
We successfully installed the “Azure Storage Client” package also.

Click on “Close” button.



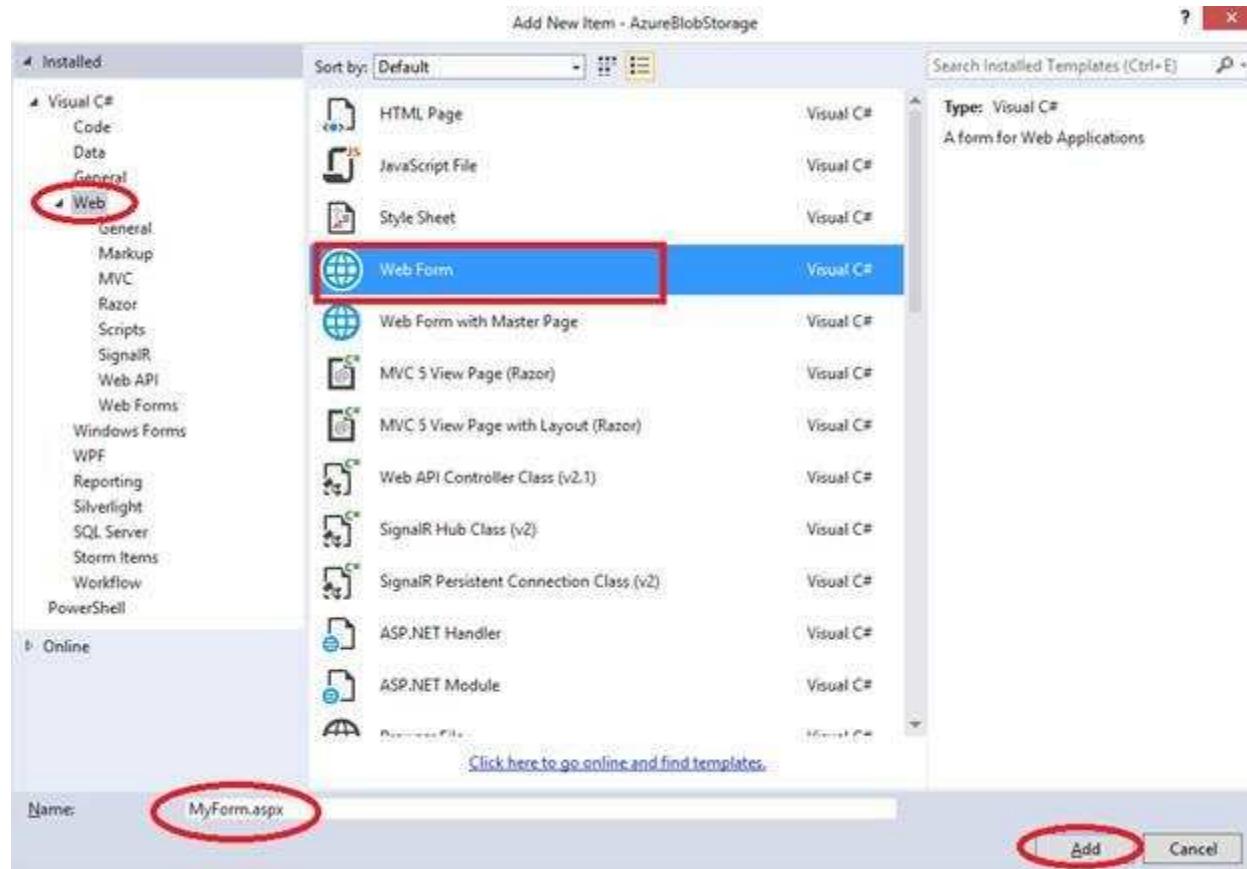
Go to your project, here my project name is “Azure Blob Storage”.

Right click on to project, select “Add” and select “New Item”.

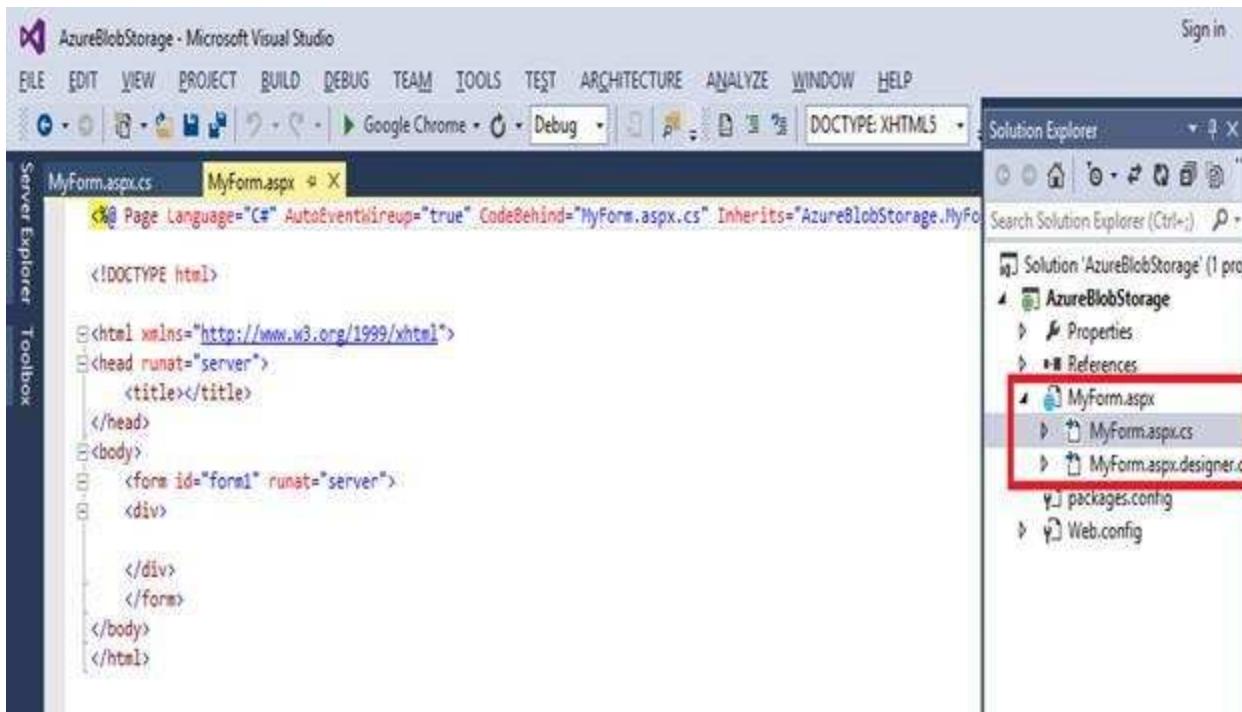


Here, I am going to create a Web Form, so select a Web Form in the list and give the name to your Web Form; here I gave the name “**MyForm**” to my Web Form, you can give any name to your Web Form.

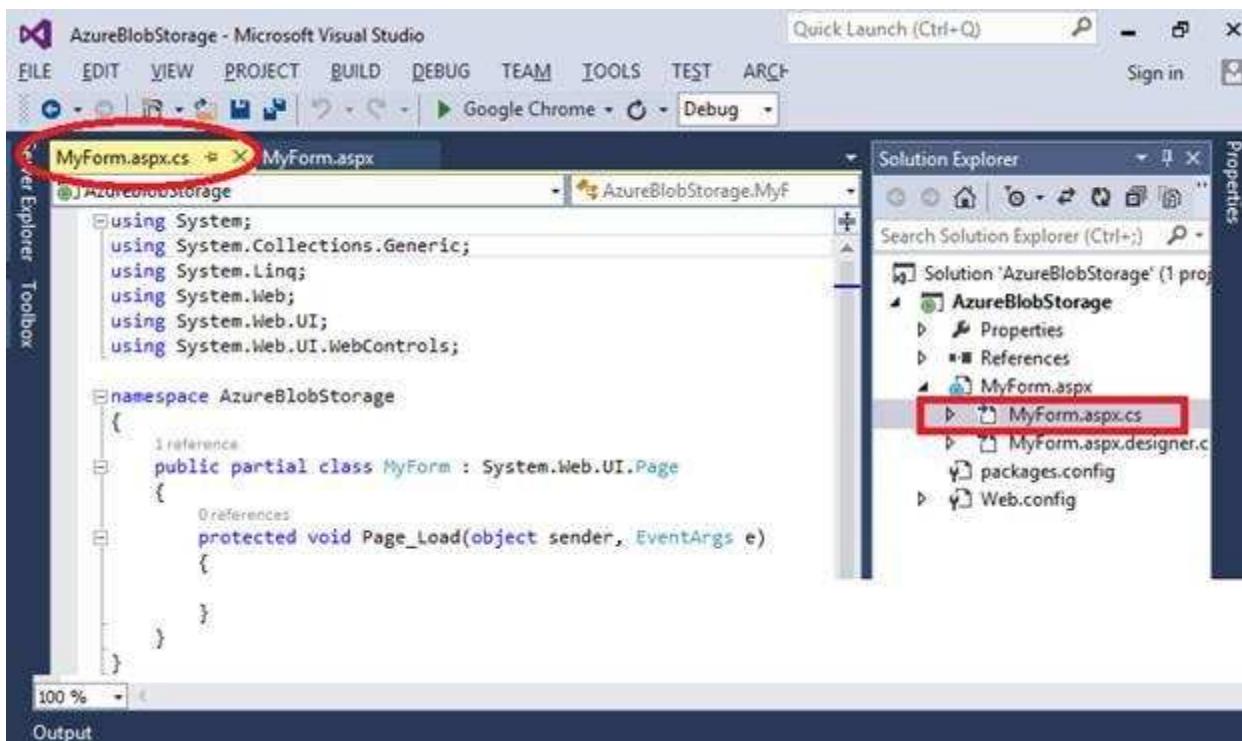
Click on “**Add**” button.



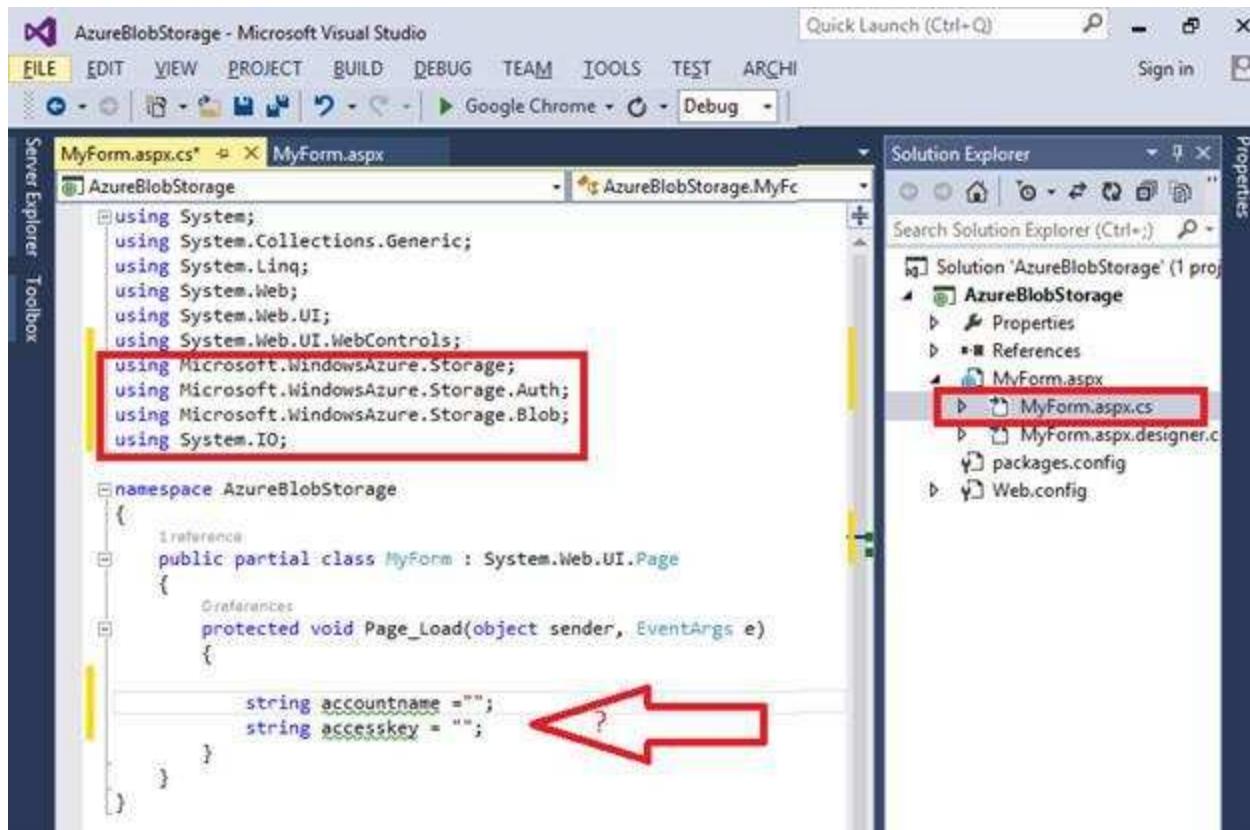
After creating a form, you can see your project, where form will be created successfully with the name “**MyForm.aspx**”



In the “**MyForm.aspx.cs**” page, let me do some changes.



In the “**MyForm.aspx.cs**” page, I have added all namespaces which is minimum required.



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Auth;
using Microsoft.WindowsAzure.Storage.Blob;
using System.IO;

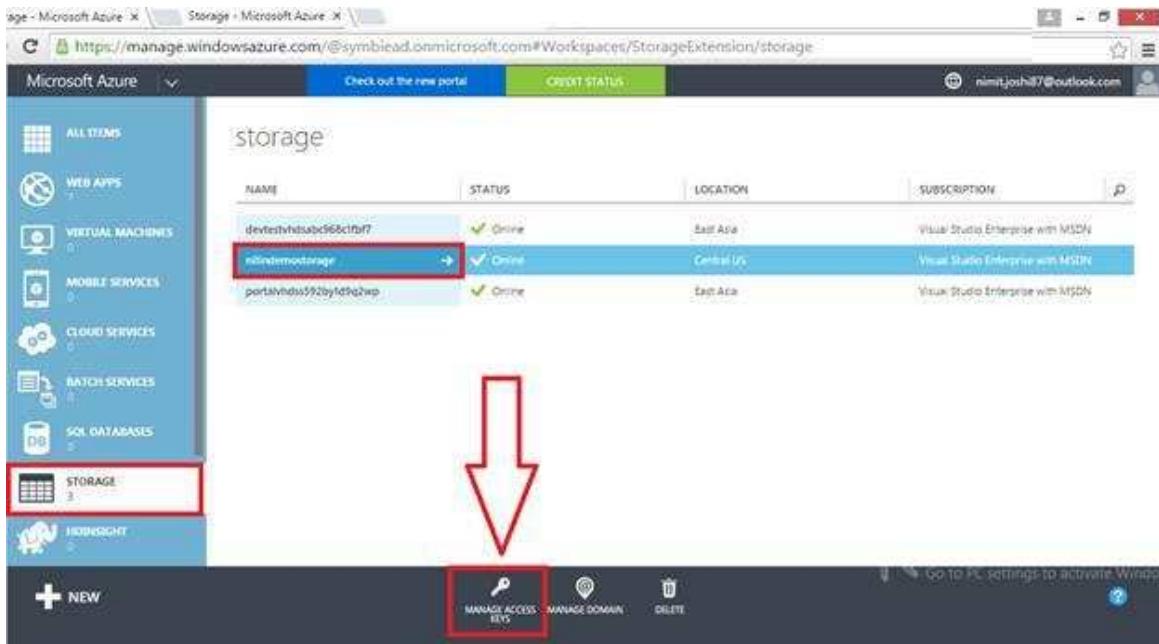
namespace AzureBlobStorage
{
    public partial class MyForm : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string accountname = "";
            string accesskey = "";
        }
    }
}

```

Now we need “**accountname**” and “**accesskey**” to get connected with Azure Account, so go to Azure portal.

Go to “**Storage**” option where I have created a Storage Account with the name “**nitindemostorage**” in my previous chapter.

Here, after clicking “**nitindemostorage**” named Storage Account, go to bottom of the window and select “**Manage Access Keys**”.

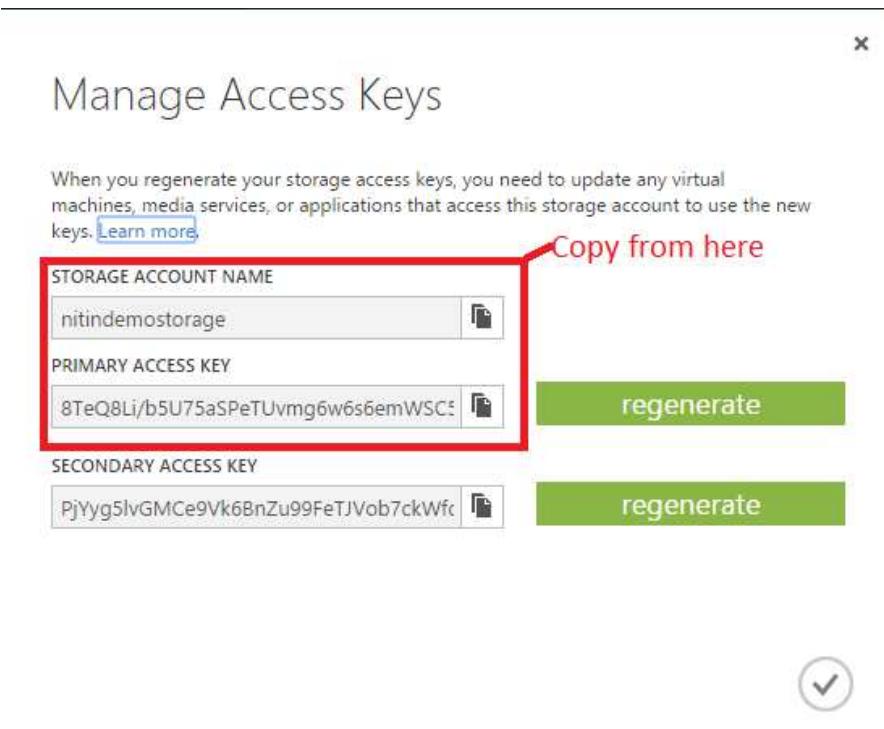


The screenshot shows the Microsoft Azure Storage blade. On the left sidebar, under the 'STORAGE' category, there is a red box around the 'STORAGE' icon. In the main area, there is a table listing three storage accounts:

NAME	STATUS	LOCATION	SUBSCRIPTION
devtestvhlsabc968ctbf7	✓ Online	East Asia	Visual Studio Enterprise with MSDN
nitin demostorage	✓ Online	Central US	Visual Studio Enterprise with MSDN
portalguidss592byld9q2wp	✓ Online	East Asia	Visual Studio Enterprise with MSDN

A large red arrow points downwards from the top of the screenshot towards the 'MANAGE ACCESS KEYS' button at the bottom of the page.

You will find the following window, copy the “Storage Account Name” and “Primary Access Key” from here,



Manage Access Keys

When you regenerate your storage access keys, you need to update any virtual machines, media services, or applications that access this storage account to use the new keys. [Learn more](#)

STORAGE ACCOUNT NAME
nitin demostorage

PRIMARY ACCESS KEY
8TeQ8Li/b5U75aSPeTUvmg6w6s6emWSC

SECONDARY ACCESS KEY
PjYyg5lvGMCe9Vk6BnZu99FeTJVob7ckWfc

regenerate

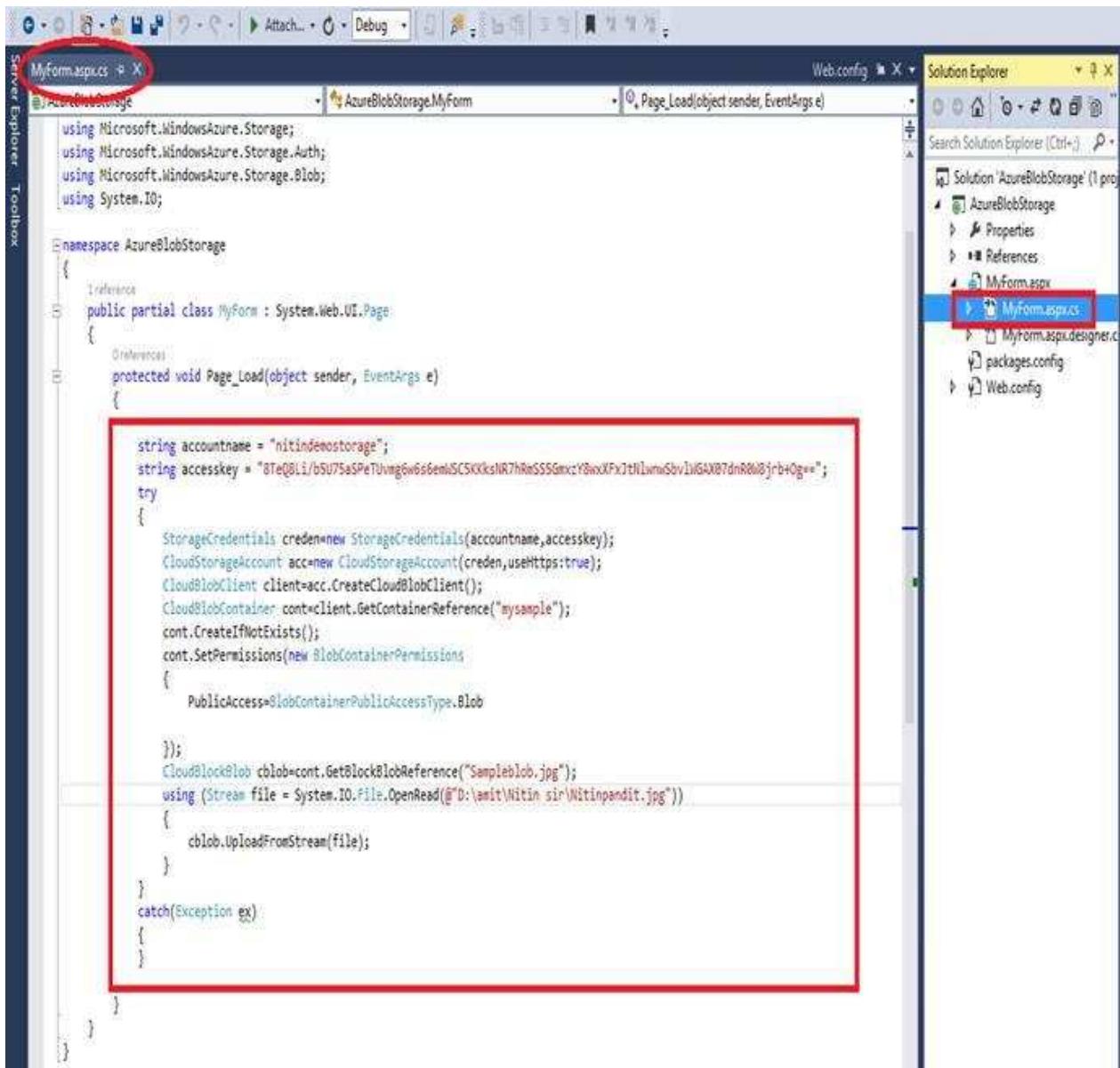
regenerate

(A red box highlights the 'STORAGE ACCOUNT NAME' and 'PRIMARY ACCESS KEY' fields. A red arrow labeled 'Copy from here' points to the 'PRIMARY ACCESS KEY' field.)

Go back in your project and make some changes over there.

In the “accountname” and “accesskey” paste the copied item over there.

And make some other code over there for getting connections with Azure storage account with “**CloudStorageAccount**” class object and at last convert your image into stream upload as in the following image:



```

using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Auth;
using Microsoft.WindowsAzure.Storage.Blob;
using System.IO;

namespace AzureBlobStorage
{
    public partial class MyForm : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string accountname = "nitindemostorage";
            string accesskey = "8TeQ8Li/bSU75aSPeTUvng6w6enlSC5CKksNR7hRmSSSGmxzY8nxXFxJtllwmwSbv1lGAX87dnR8%8jrb+0g==";
            try
            {
                StorageCredentials creden=new StorageCredentials(accountname,accesskey);
                CloudStorageAccount acc=new CloudStorageAccount(creden,useHttps:true);
                CloudBlobClient client=acc.CreateCloudBlobClient();
                CloudBlobContainer cont=client.GetContainerReference("mysample");
                cont.CreateIfNotExists();
                cont.SetPermissions(new BlobContainerPermissions
                {
                    PublicAccess=BlobContainerPublicAccessType.Blob
                });
                CloudBlockBlob cblob=cont.GetBlockBlobReference("Sampleblob.jpg");
                using (Stream file = System.IO.File.OpenRead(@"D:\amit\Nitin sir\Nitinpandit.jpg"))
                {
                    cblob.UploadFromStream(file);
                }
            }
            catch(Exception ex)
            {
            }
        }
    }
}

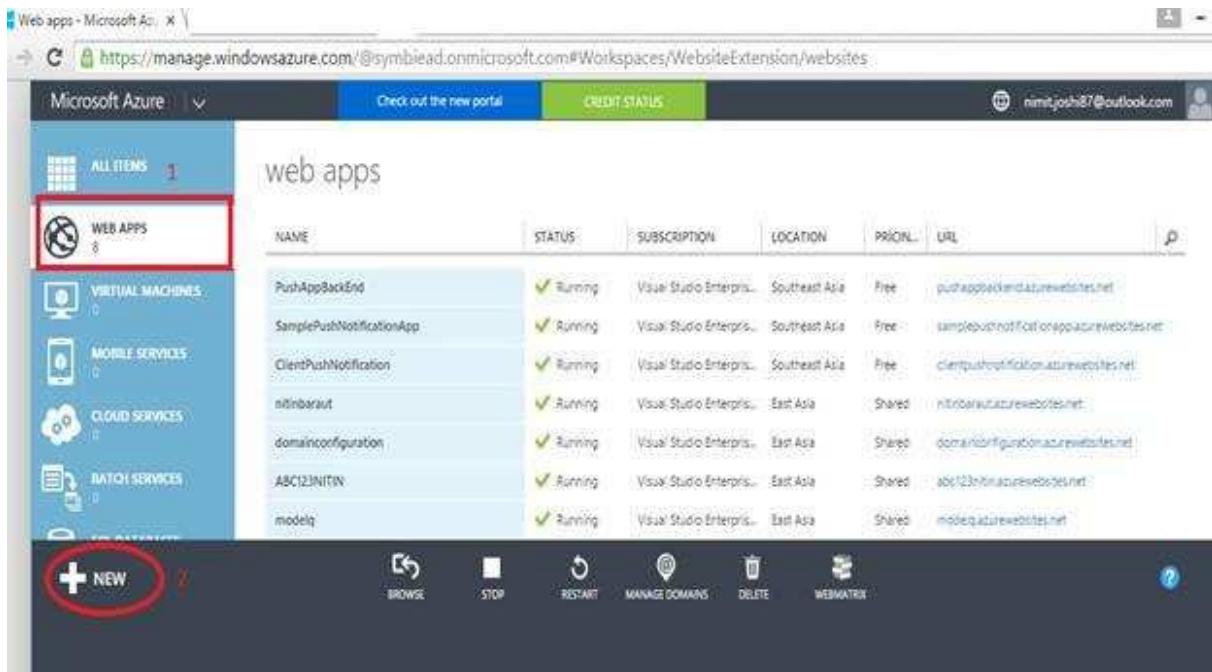
```

Code

```
1. string accountname = "nitindemostorage";
2.
3. string
accesskey = "8TeQ8Li/b5U75aSPeTUvng6w6s6emlWSC5KKksNR7hRmSS5GmxzY8wxXFxJtNlwnwSbv1WGAX07
dnR0W8jrb+0g==";
4.
5. try
6.
7. {
8.
9.     StorageCredentialscreden = newStorageCredentials(accountname, accesskey);
10.
11.    CloudStorageAccountacc = newCloudStorageAccount(creden, useHttps: true);
12.
13.    CloudBlobClient client = acc.CreateCloudBlobClient();
14.
15.    CloudBlobContainercont = client.GetContainerReference("mysample");
16.
17.    cont.CreateIfNotExists();
18.
19.    cont.SetPermissions(newBlobContainerPermissions
20.
21.    {
22.
23.        PublicAccess = BlobContainerPublicAccessType.Blob
24.
25.    });
26.
27.    CloudBlockBlobcblob = cont.GetBlockBlobReference("Sampleblob.jpg");
28.
29.    using(Stream file = System.IO.File.OpenRead(@"D:\amit\Nitin sir\Nitinpandit.jpg"))
30.
31.    {
32.
33.        cblob.UploadFromStream(file);
34.
35.    }
36.
37. } catch (Exception ex)
38.
39. {
40.
41. }
```

One more thing, here we need to create a Web App Profile for publishing our app into the cloud.

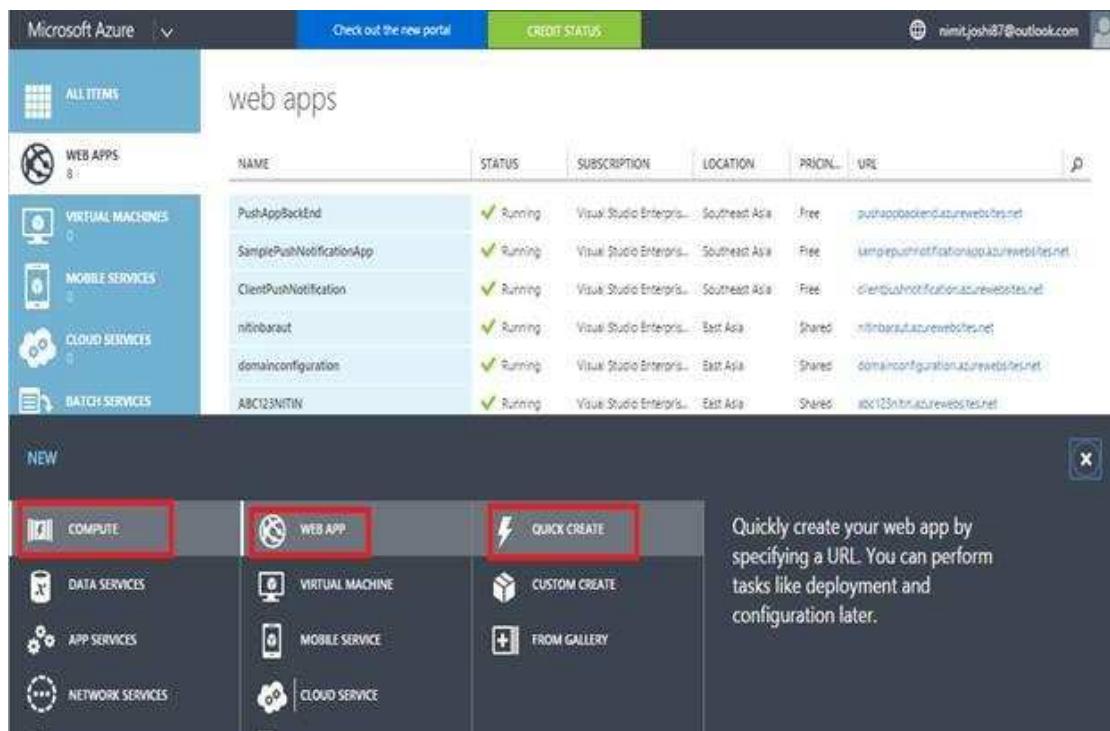
Simply go to “WEB APPS” in your Azure Portal and click on “New” to add a new Web App.



The screenshot shows the Microsoft Azure Web Apps dashboard. On the left, there's a sidebar with icons for ALL ITEMS, WEB APPS (highlighted with a red box), VIRTUAL MACHINES, MOBILE SERVICES, CLOUD SERVICES, and BATCH SERVICES. Below the sidebar is a large 'NEW' button with a plus sign, also highlighted with a red circle. The main area is titled 'web apps' and contains a table with columns: NAME, STATUS, SUBSCRIPTION, LOCATION, PRICING, and URL. The table lists several web apps: PushAppBackEnd, SamplePushNotificationApp, ClientPushNotification, nitinbaraut, domainconfiguration, ABC123INITIN, and modelq. Each row shows a green checkmark in the STATUS column, indicating they are all running.

After clicking on “**New**”, you will see the following window.

Go to “**COMPUTE**” option, select “**WEB APP**”, and select “**QUICK CREATE**”.

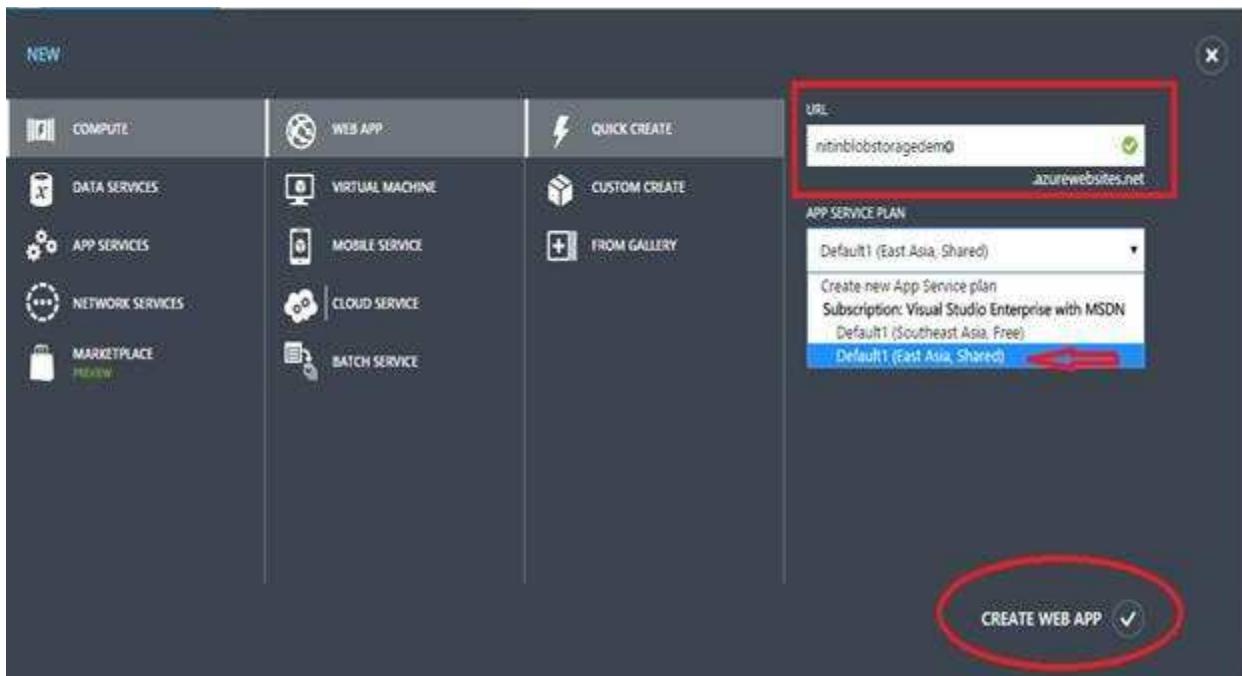


This screenshot shows the 'New' dialog box in Microsoft Azure. At the top, it says 'Microsoft Azure' and 'Check out the new portal'. On the left, there's a sidebar with 'ALL ITEMS' and a 'NEW' section containing options: COMPUTE (highlighted with a red box), WEB APP (highlighted with a red box), and QUICK CREATE (highlighted with a red box). Below this are other service categories: DATA SERVICES, VIRTUAL MACHINE, CUSTOM CREATE; APP SERVICES, MOBILE SERVICE, FROM GALLERY; and NETWORK SERVICES, CLOUD SERVICE. To the right of the sidebar, there's a large text area that reads: "Quickly create your web app by specifying a URL. You can perform tasks like deployment and configuration later." There's also a close button in the top right corner of the dialog box.

In the URL section, you can write your Web App name. Here I have given “**nitinblobstoragedemo**”, if the entered name is correct it will create a check sign (in blue color).

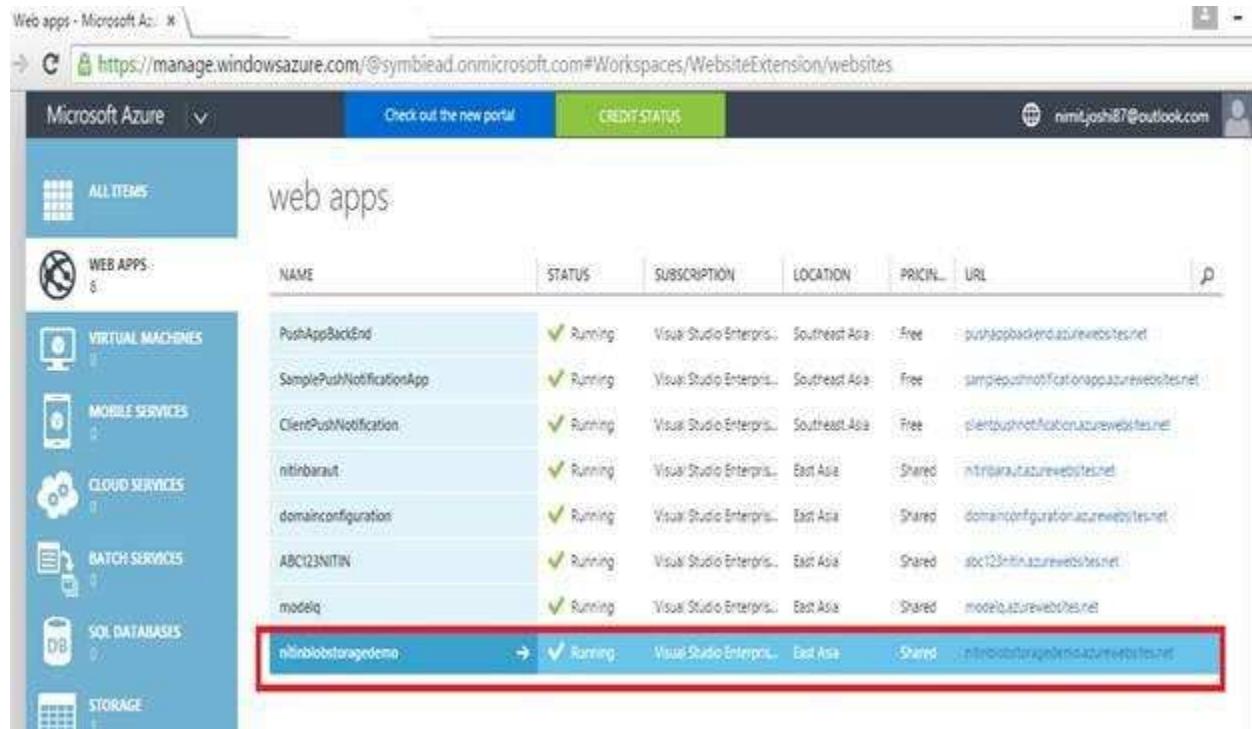
In the “APP SERVICE PLAN” click on “**Default1 (East Asia, Shared)**”.

Click on “**CREATE WEB APP**”.



We have created our Web App successfully. After doing this, we can see our created Web APP (“nitinblobstoragedemo”).

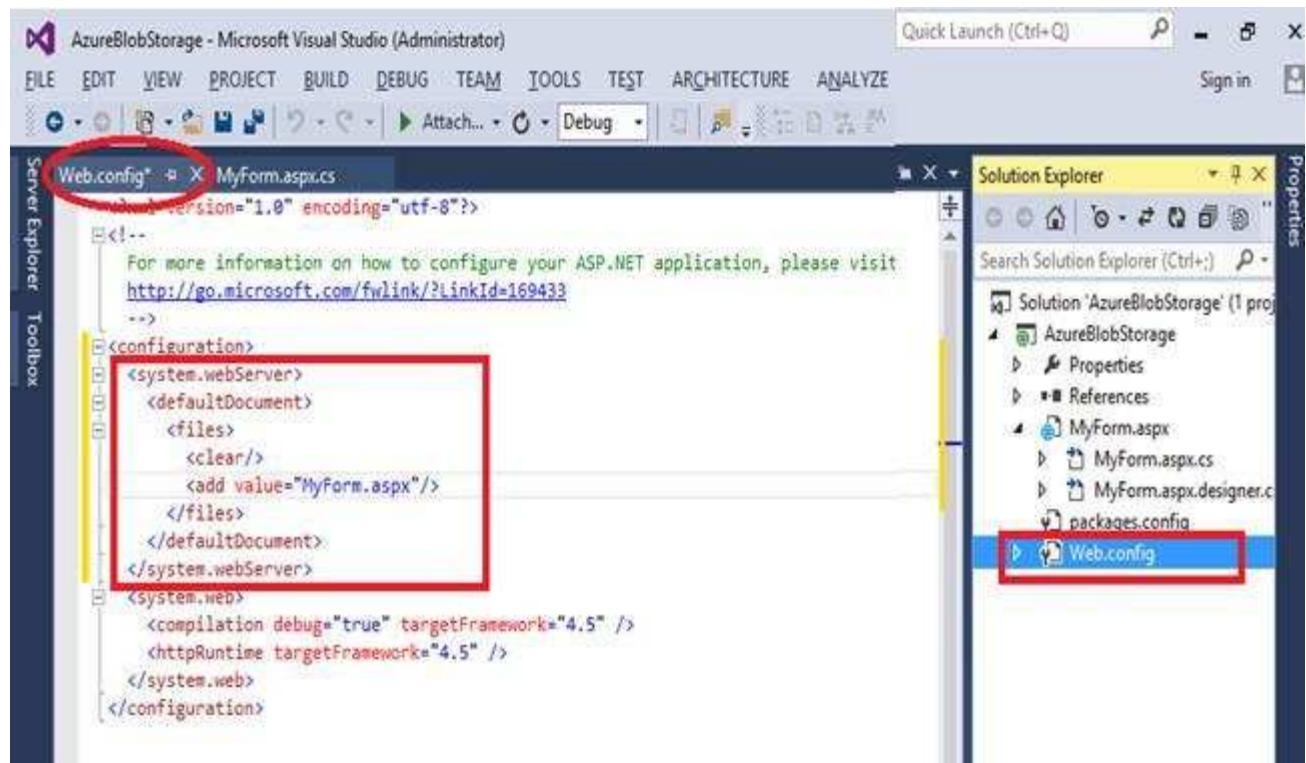
And it is in running mode.



NAME	STATUS	SUBSCRIPTION	LOCATION	PRICING	URL
PushAppBackend	Running	Visual Studio Enterprise	Southeast Asia	Free	pushappbackend.azurewebsites.net
SamplePushNotificationApp	Running	Visual Studio Enterprise	Southeast Asia	Free	samplepushnotificationapp.azurewebsites.net
ClientPushNotification	Running	Visual Studio Enterprise	Southeast Asia	Free	clientpushnotification.azurewebsites.net
nitinbaraut	Running	Visual Studio Enterprise	East Asia	Shared	nitinbaraut.azurewebsites.net
domainconfiguration	Running	Visual Studio Enterprise	East Asia	Shared	domainconfiguration.azurewebsites.net
ABC123NITIN	Running	Visual Studio Enterprise	East Asia	Shared	abc123nitin.azurewebsites.net
modelq	Running	Visual Studio Enterprise	East Asia	Shared	modelq.azurewebsites.net
nitinblobstorage	Running	Visual Studio Enterprise	East Asia	Shared	nitinblobstorage.azurewebsites.net

We make some code in the Web Config file also.

Here in the <add value="MyForm.aspx"/>, "My Form" is my Web form.



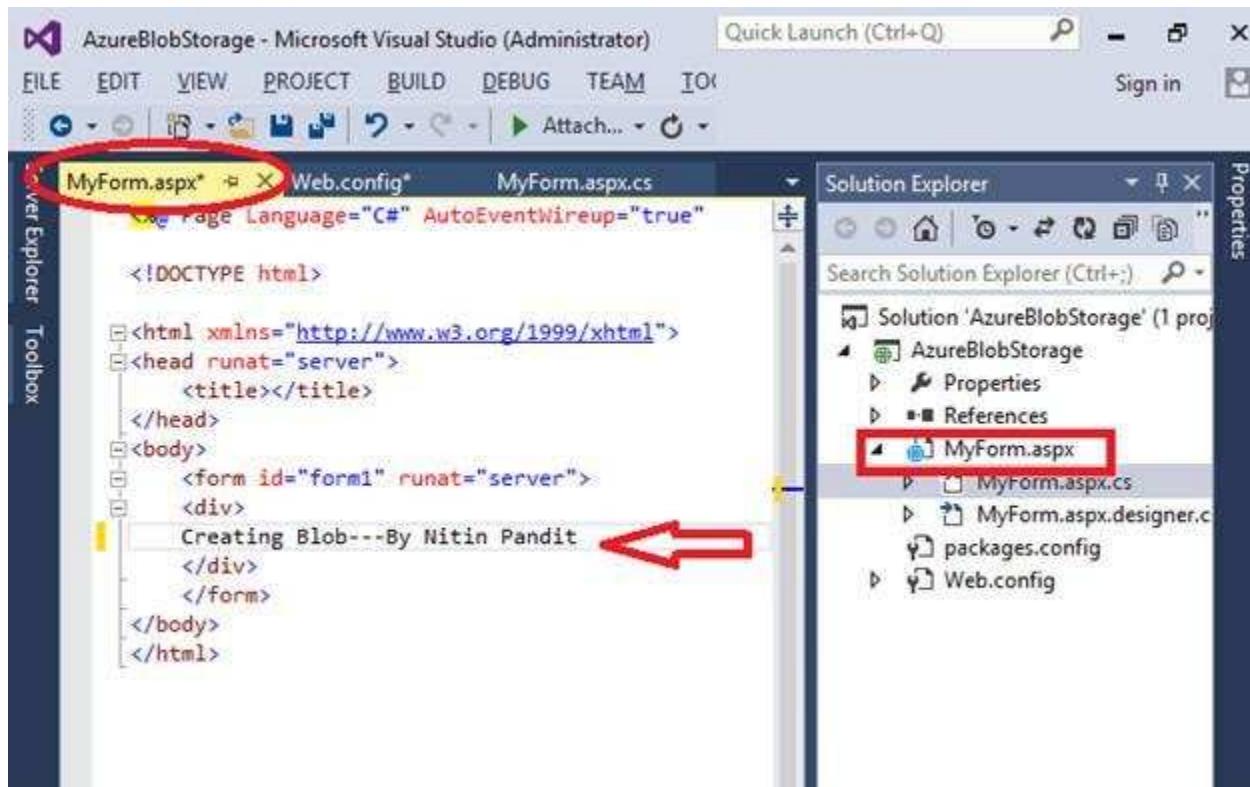
```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE configuration SYSTEM "http://www.microsoft.com/aspnet/xmlconfigvalidationschema/1.0/configuration.xsd">
<configuration>
    <system.webServer>
        <defaultDocument>
            <files>
                <clear/>
                <add value="MyForm.aspx"/>
            </files>
        </defaultDocument>
    </system.webServer>
    <system.web>
        <compilation debug="true" targetFramework="4.5" />
        <httpRuntime targetFramework="4.5" />
    </system.web>
</configuration>

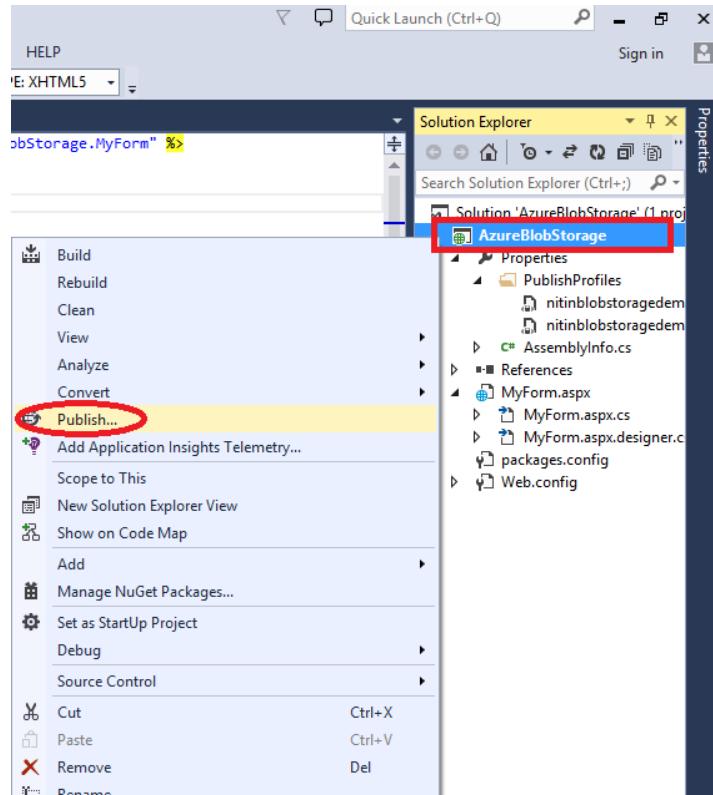
```

In the “MyForm.aspx” you can give any message for output. Whenever you build your project, the typed message will be your output.

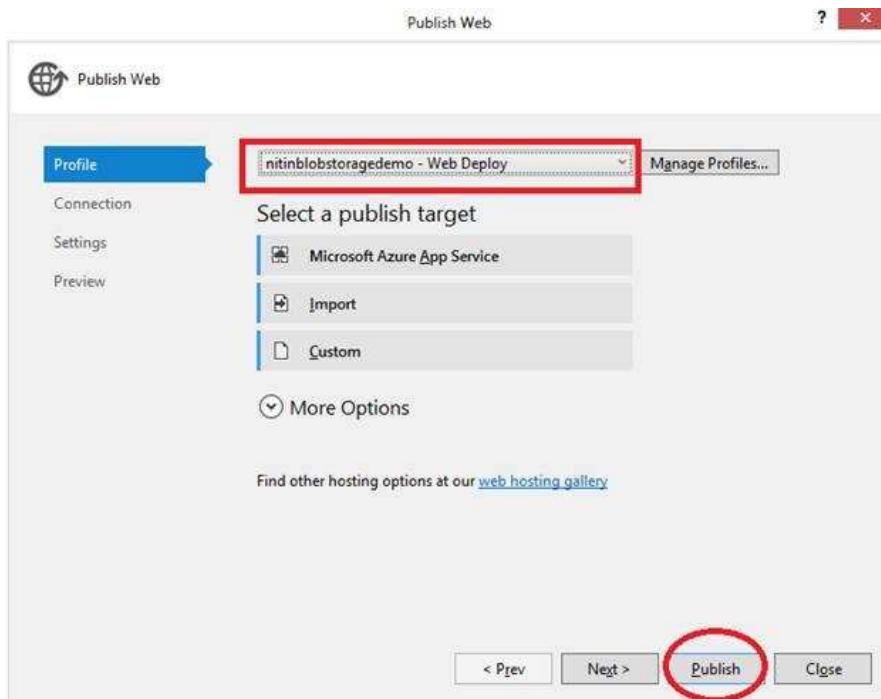
So, here in the div section I have given a message, “**Creating Blob---By Nitin Pandit**”.



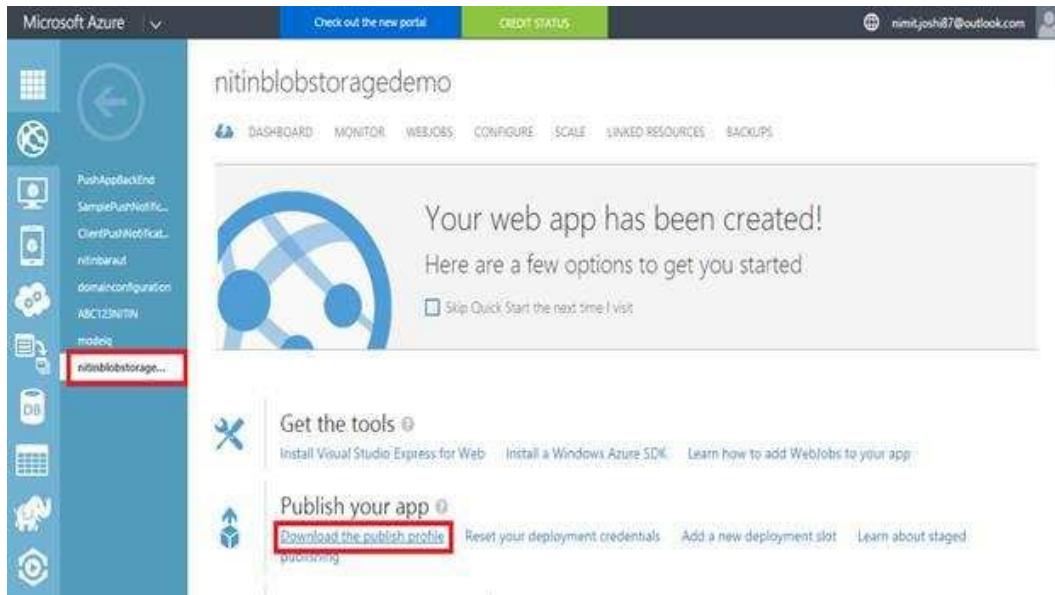
Now we need to publish our created project in the cloud. It will also save in the Azure Explorer. You can publish your project by two ways. Here, I am going to discuss both ways. Firstly, right click on your project. Here my project is “**AzureBlobStorage**”, so select the “**Publish**” option.



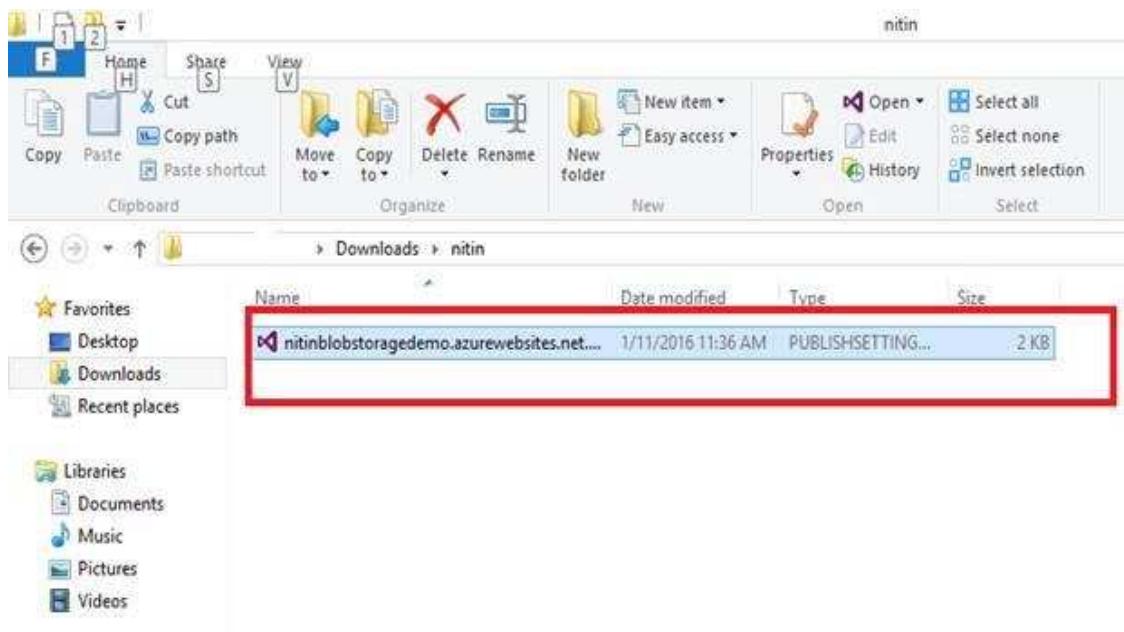
In the following figure, you can see “**nitinblobstoragedemo**” is my Web App name, which we have created in the starting. Select your Web APP profile and click “**Publish**” button.



After clicking on “publish” button the project would publish on cloud as well as in your Azure Explorer. The second way to publish your project is to go for cloud. You can download your profile from the Azure account in your Web Apps. For this click on your Web App profile, in the **Publish** your app section, click on “**Download the publish profile**”.

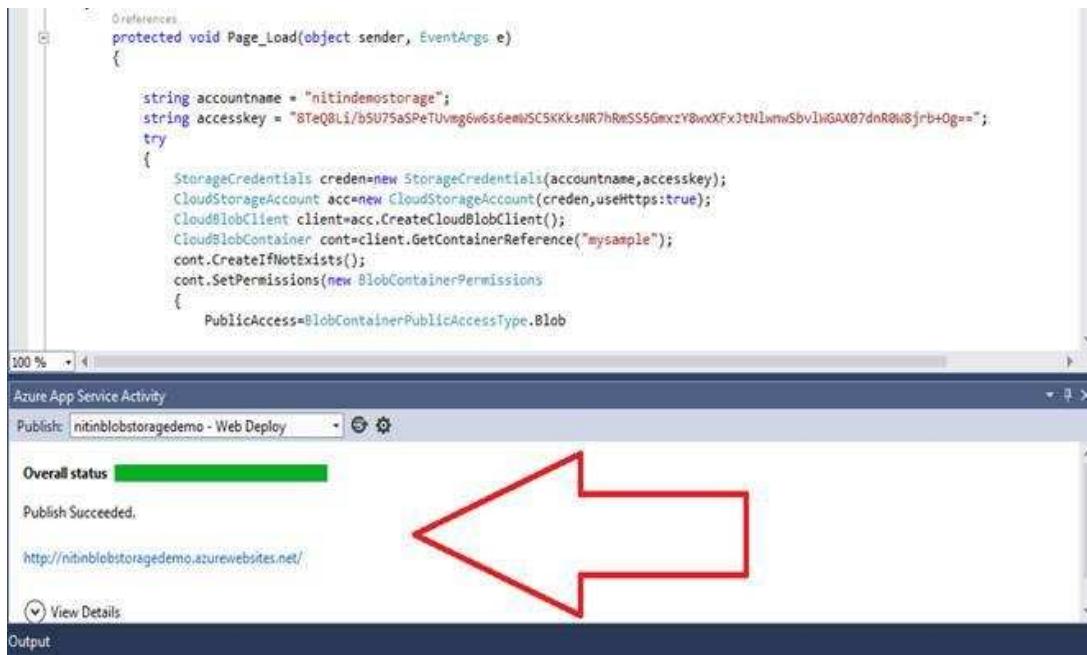


After clicking, your profile will be downloaded in your system.



After this, now we can see the status of our publishing. So my “**nitinblobstoragedemo**” app service published successfully.

Now build your project and get your output.



```

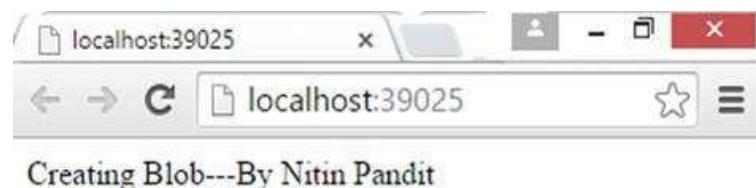
    References.
protected void Page_Load(object sender, EventArgs e)
{
    string accountname = "nitinblobstorage";
    string accesskey = "8TeQ8Li/bSU7saSPeTUvng6w6s6emjSCKKksNR7hRmSS5GmxzYBwxXFx3tNlwmwSbv1kGAX07dnR0n8jrb+0g==";
    try
    {
        StorageCredentials creden=new StorageCredentials(accountname,accesskey);
        CloudStorageAccount acc=new CloudStorageAccount(creden,useHttps:true);
        CloudBlobClient client=acc.CreateCloudBlobClient();
        CloudBlobContainer cont=client.GetContainerReference("mysample");
        cont.CreateIfNotExists();
        cont.SetPermissions(new BlobContainerPermissions
        {
            PublicAccess=BlobContainerPublicAccessType.Blob
        });
    }
}

100 % ▶
Azure App Service Activity
Publish: nitinblobstoragedemo - Web Deploy
Overall status [green bar]
Publish Succeeded.
http://nitinblobstoragedemo.azurewebsites.net/
View Details
Output:

```

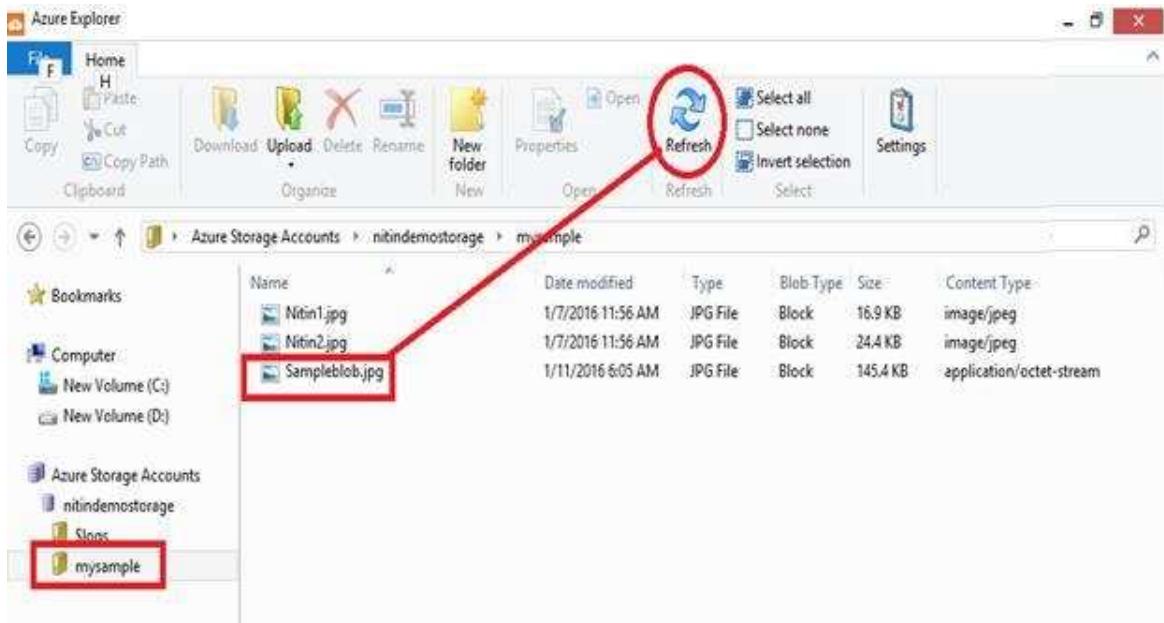
Now run the application, press F5.

So, finally we got our output.



We can see the changes of this in our Azure Explorer.

Now refresh the Azure Explorer to check your image.



“Sampleblob.jpg” is my image which is generated by programming,

```

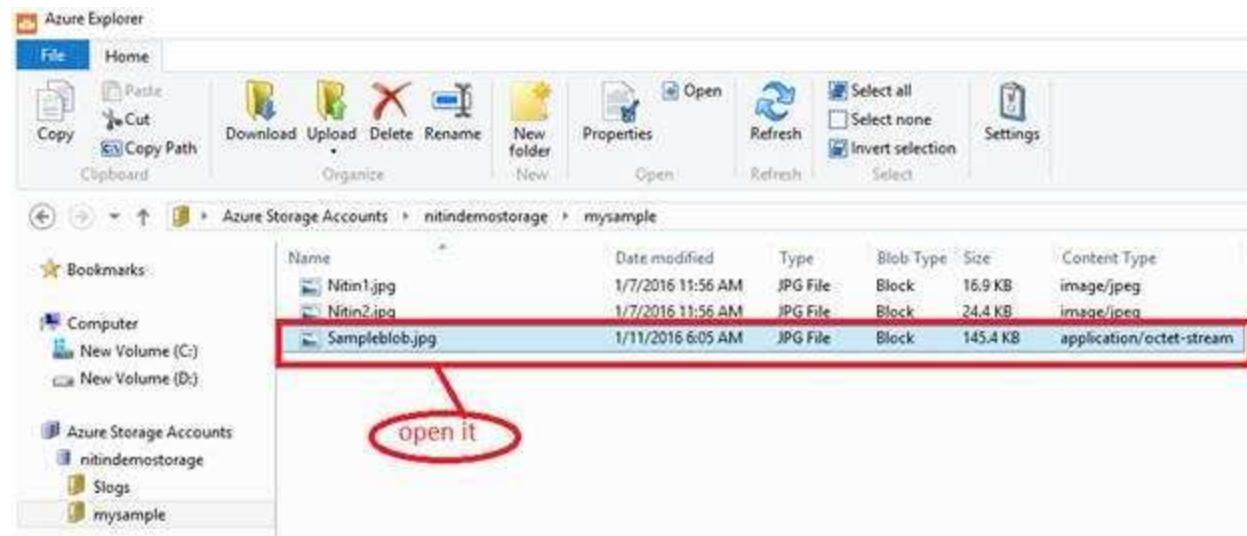

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Blob;

public partial class MyForm : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string accountname = "nitindemostorage";
        string accesskey = "8TeQ8Li/b5U75aSPeTUvmg6w6s6emWS5KKksNR7hRmSS5GmxzY8wxXFxJtNlwnwSbv1WGA";

        StorageCredentials creden=new StorageCredentials(accountname,accesskey);
        CloudStorageAccount acc=new CloudStorageAccount(creden,useHttps:true);
        CloudBlobClient client=acc.CreateCloudBlobClient();
        CloudBlobContainer cont=client.GetContainerReference("mysample");
        cont.CreateIfNotExists();
        cont.SetPermissions(new BlobContainerPermissions
        {
            PublicAccess=BlobContainerPublicAccessType.Blob
        });
        CloudBlockBlob cblob=cont.GetBlockBlobReference("Sampleblob.jpg");
        using (Stream file = System.IO.File.OpenRead(@"D:\amit\Nitin sir\Nitinpandit.jpg"))
        {
            cblob.UploadFromStream(file);
        }
    }
}


```

Right clicks on image and open it and if you want to get that images by code on your view, you can see in my next article. So, wait for that and stay tuned.

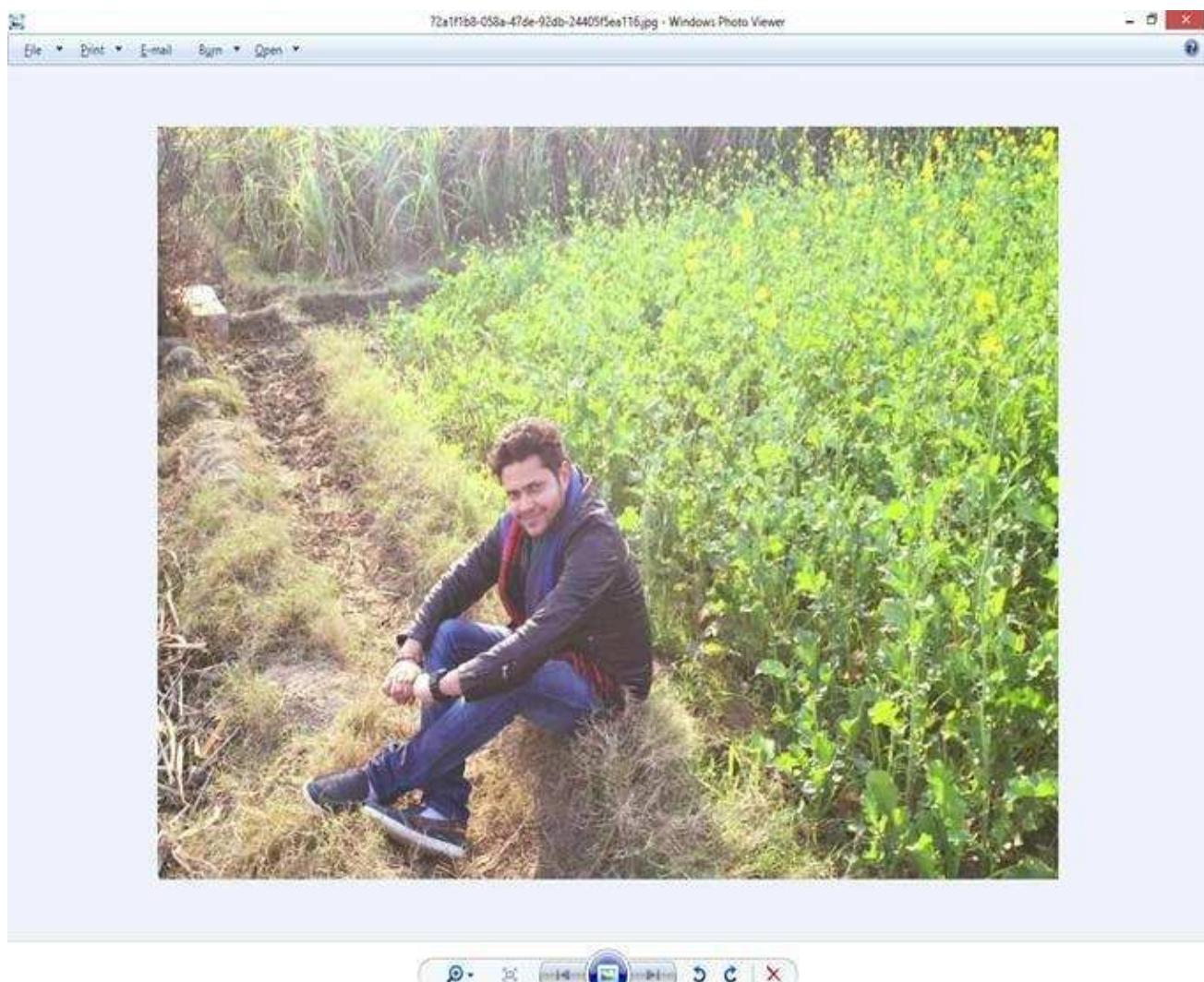


Name	Date modified	Type	Block Type	Size	Content Type
Nitin1.jpg	1/7/2016 11:56 AM	JPG File	Block	16.9 KB	image/jpeg
Nitin2.jpg	1/7/2016 11:56 AM	JPG File	Block	24.4 KB	image/jpeg
Sampleblob.jpg	1/11/2016 6:05 AM	JPG File	Block	145.4 KB	application/octet-stream

It can take few moments to download the image is in progress.



Here's the final output and you can see my image:



Demo Project Azure Blob Storage with ASP.NET MVC 5.0

This is my third one chapter on Azure Blob Storage in this I just tried to use blob storage in ASP.NET MVC application with helping of ASP.NET Web role it's a kind of demo project Application to store Images file to blob storage and edit update them and also get the list.

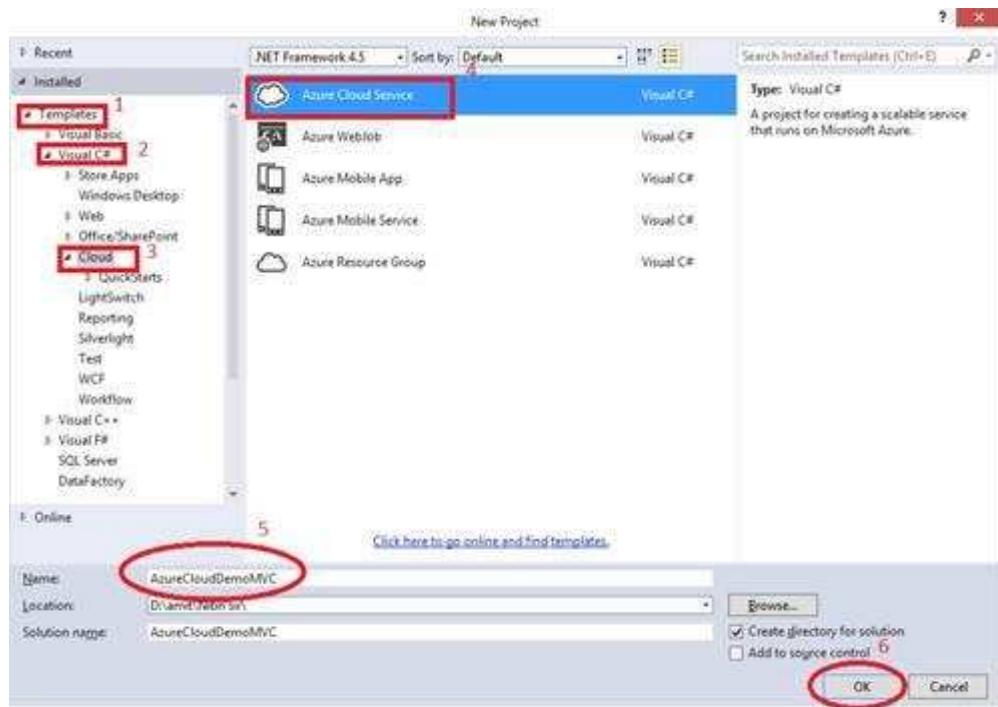
So, let's start now firstly you need Azure blob storage which I already explained in the previous chapter. After that open your Visual Studio and create Azure Cloud Service with ASP.NET Web roles as in the following steps:

Step 1: Open Visual Studio, Templates, Visual C#, and then Cloud.

Step 2: Select “Azure Cloud Service”.

Step 3: In the name section, I have given name to my project “AzureCloudDemoMVC”.

Step 4: Click on OK button.

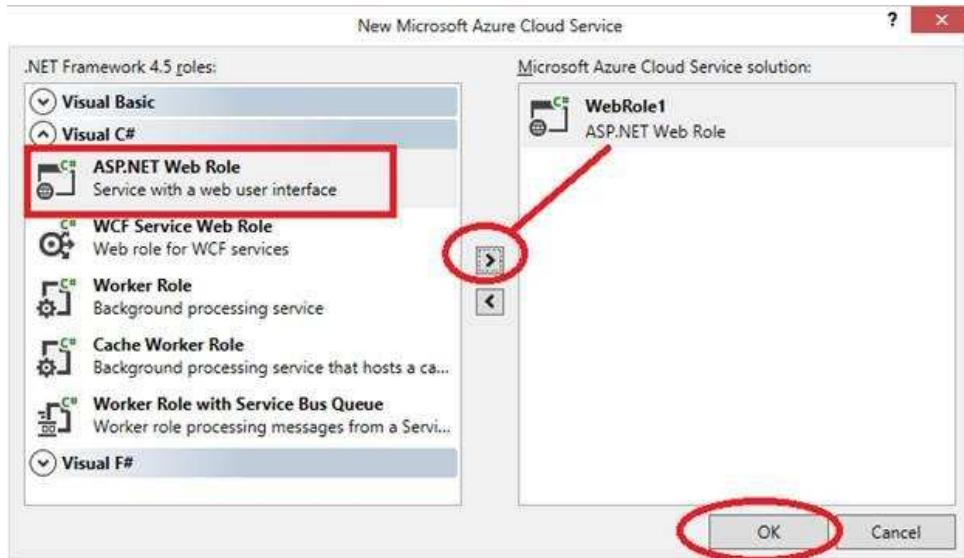


After clicking on “OK” button choose “ASP.NET Web Role”.

Web Role: Web Role is a font and application or we can say a user interface.

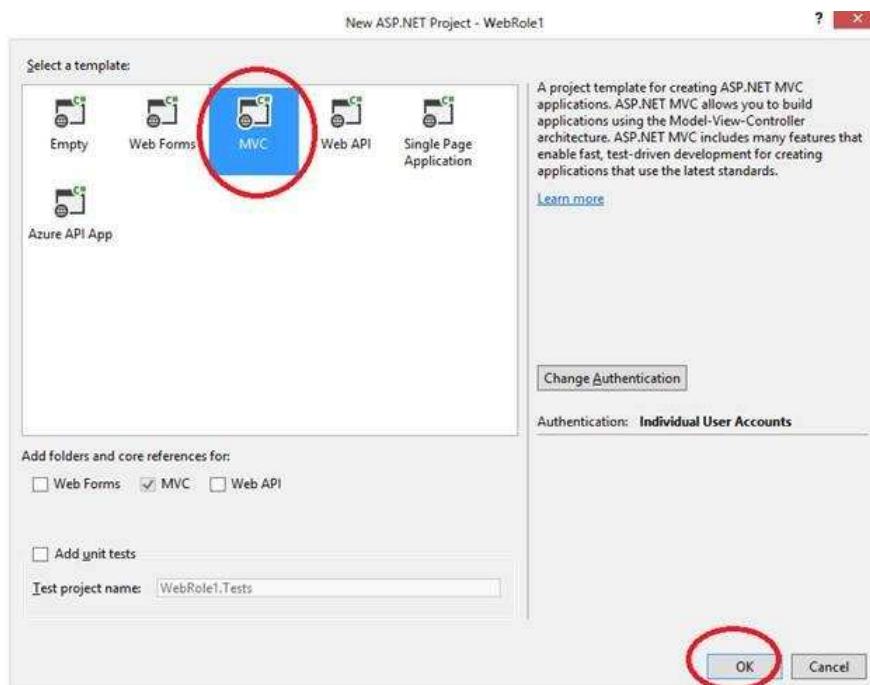
Click on (>) button (In the red circle).

Hit on “OK” button.



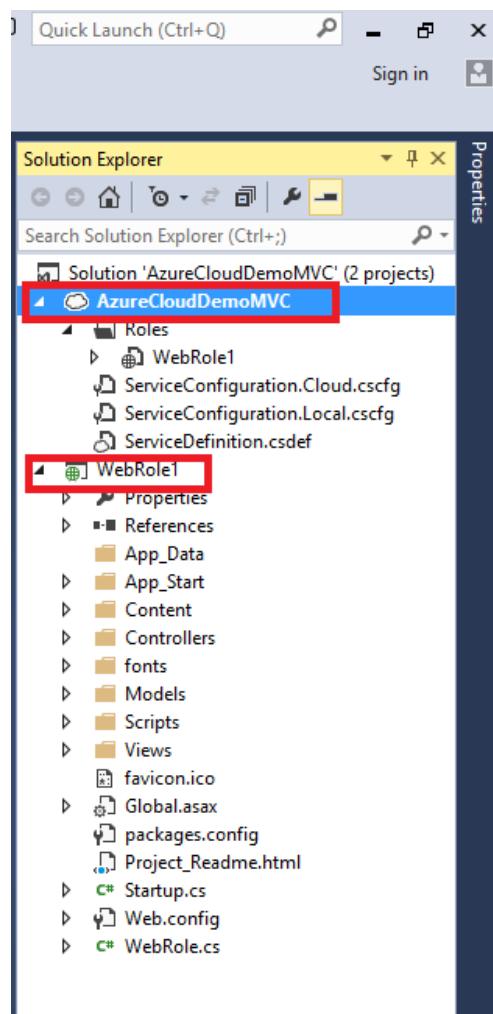
Click on “MVC” because I am going to create an MVC Application.

Click on “OK” button.

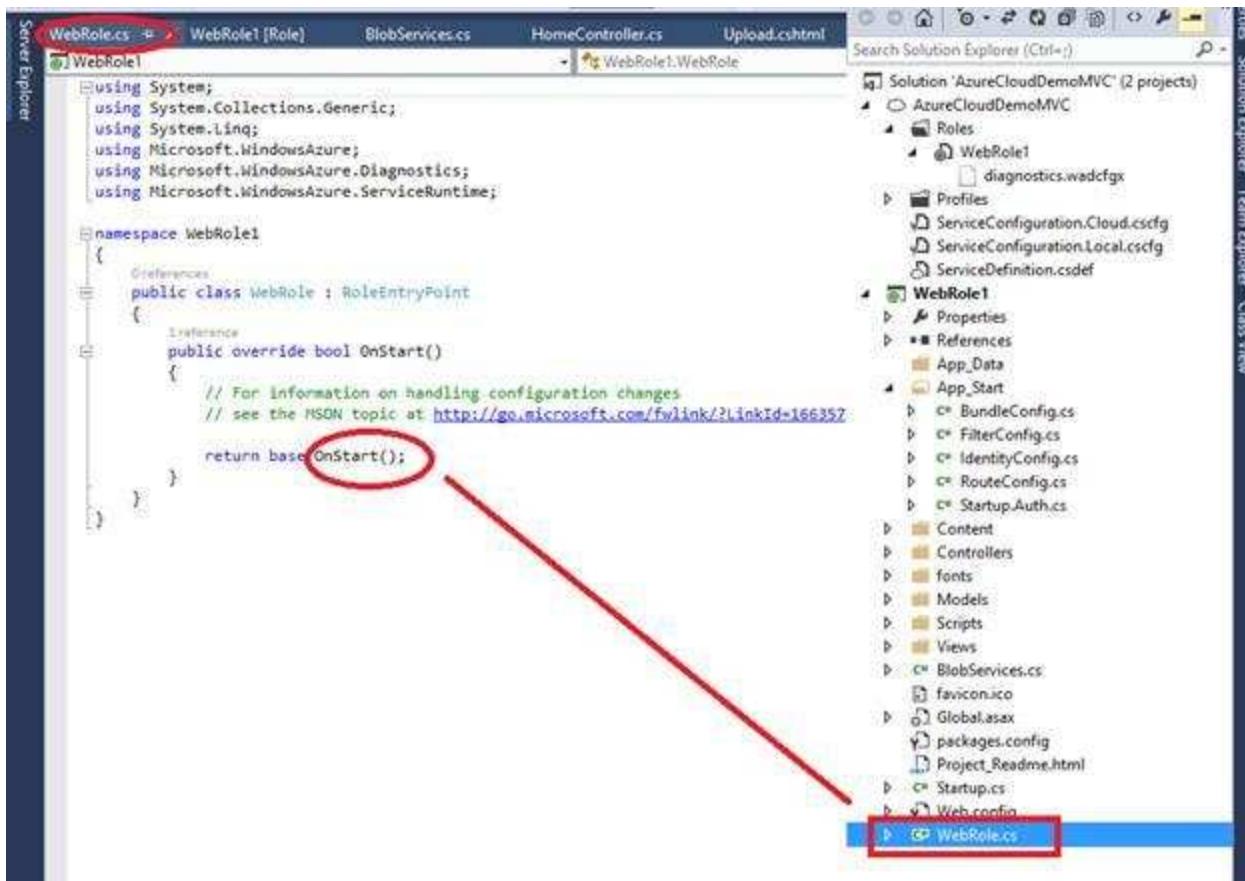


In the following figure, I have a Window Azure Cloud Project with name “AzureCloudDemoMVC”; you can see In Solution Explorer.

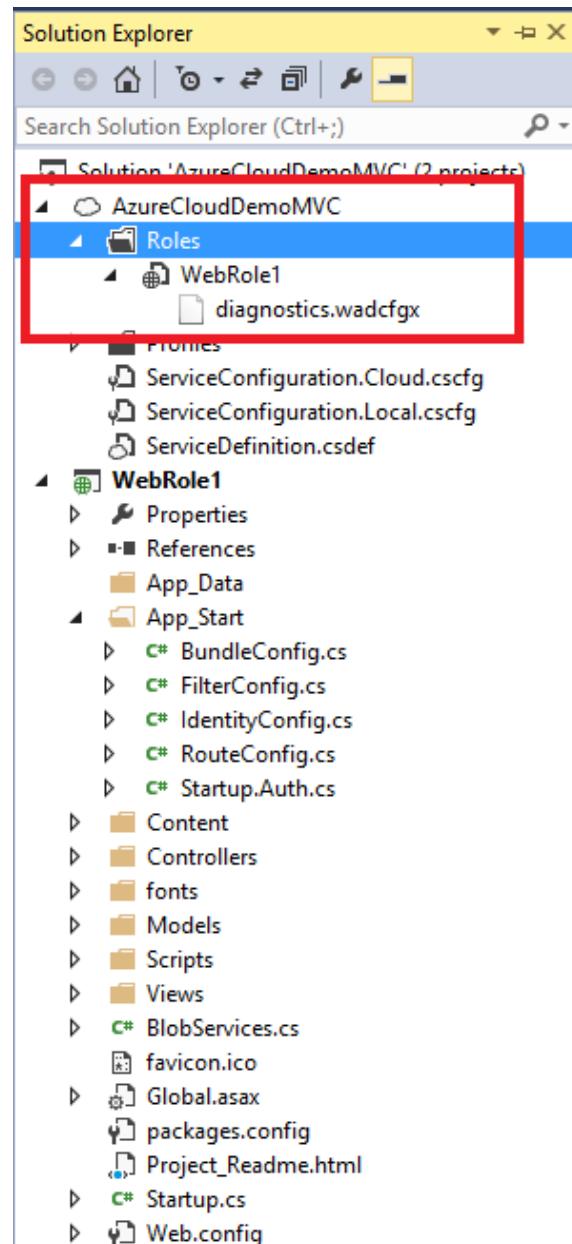
And “WebRole1” is a regular MVC project in my Solution Explorer.



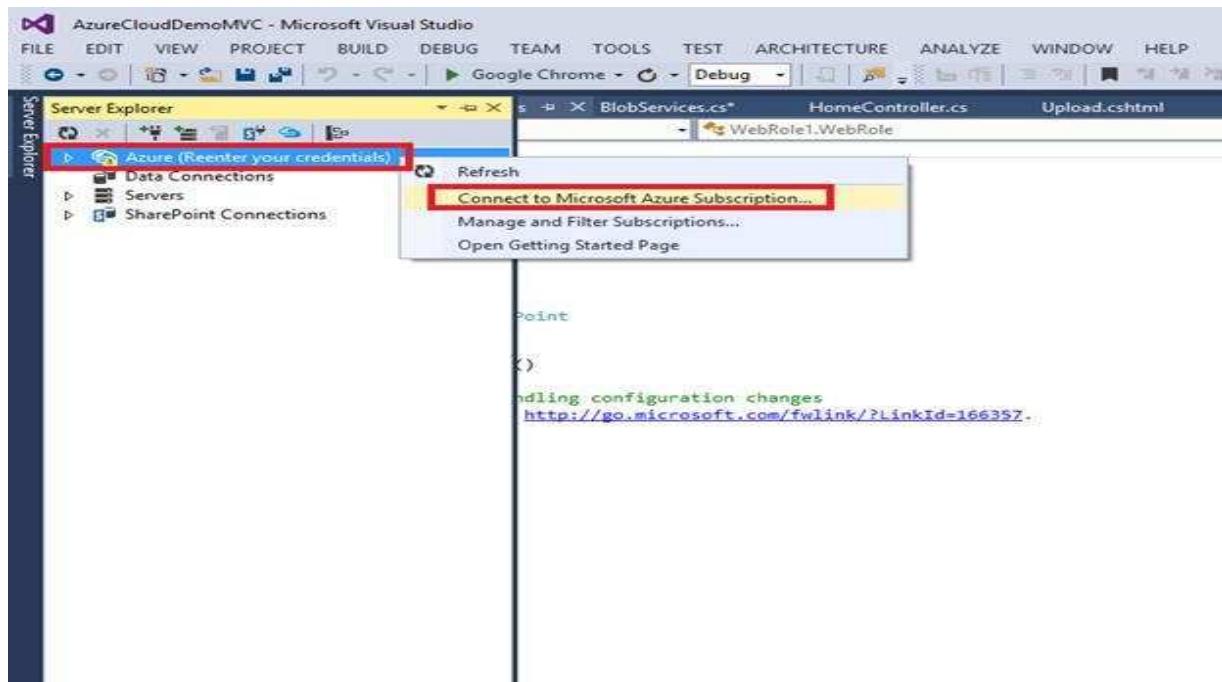
You can notice here, a new file with the name “**WebRole.cs**”. This file contains entry point settings of MVC application. You need this file while MVC running in Cloud Services. The information written in this file executes before the execution of “**Global.asax**” file.



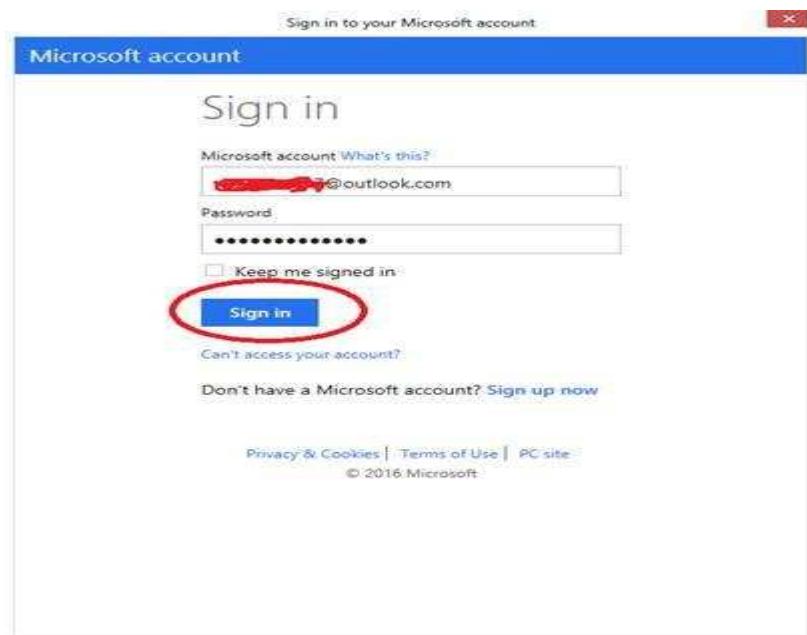
The other project is “AzureCloudDemoMVC”, this is actually a cloud project. Inside this project, you will find the “**Roles**” folder and in the “**Roles**” folder you will find “**WebRole1**”.



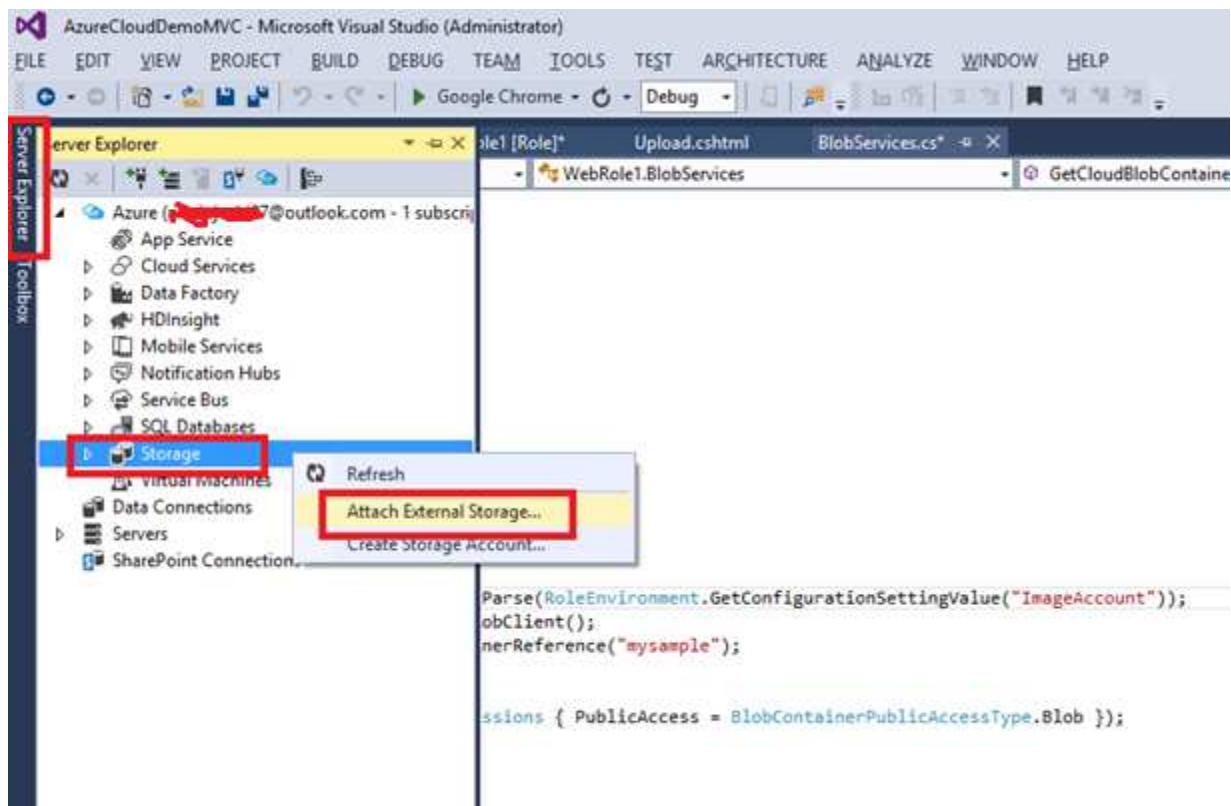
Now we need to connect with Azure, so go in the “Server Explorer” and right click on “**Azure (Reenter your credentials)**” and go to “**Connect to Microsoft Azure Subscription**”.



Fill the Login detail and click on “Sign in” button.

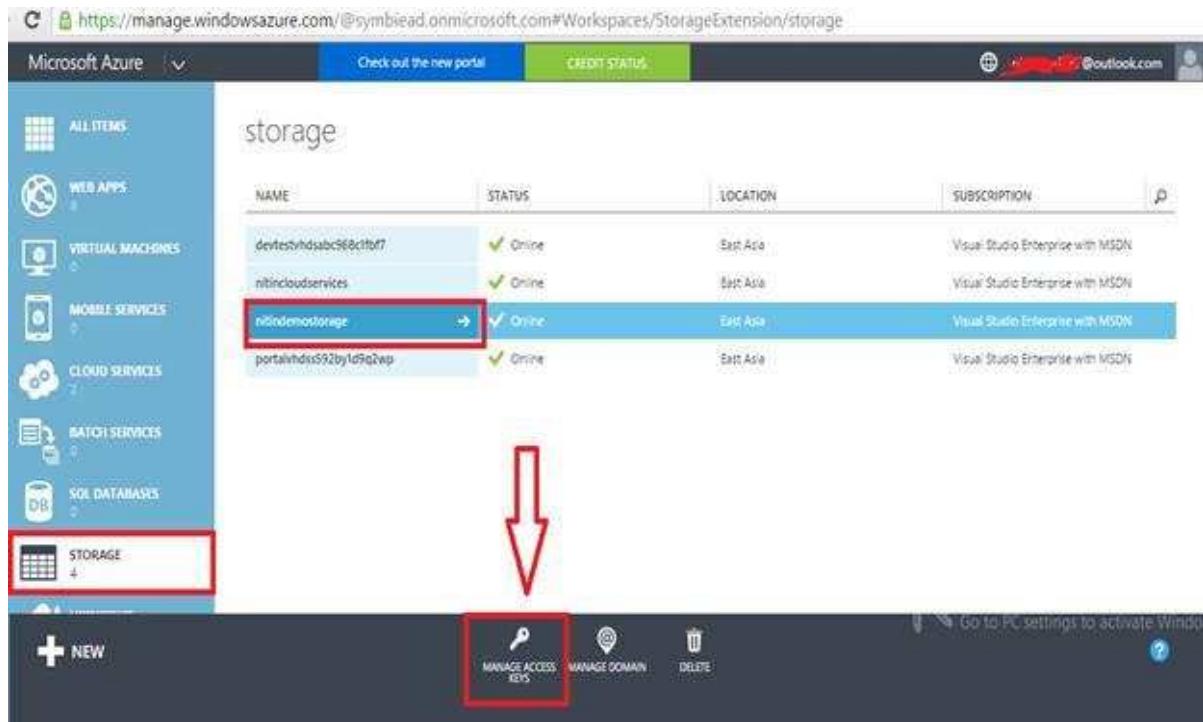


So, we have successfully connected with the Azure Account, and right click on the “**Storage**” and select “**Attach External Storage**”.



In the “Account Name” section and in “Account key” section provide name and key from the Azure Storage, so go to your Azure Storage and copy name and key from account.

In the Cloud, go to “**STORAGE**” and click on your storage account, here my storage account is “**nitindemostorage**”, and click on “**MANAGE ACCESS KEYS**”.



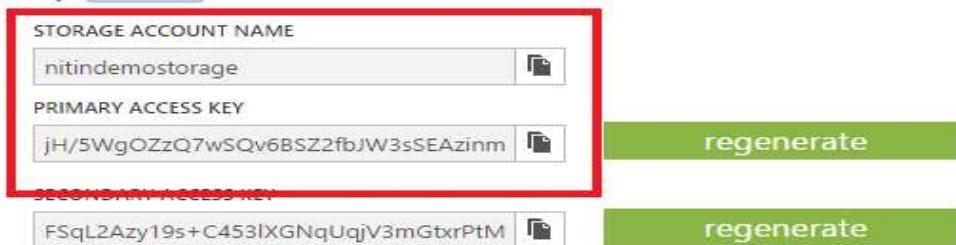
The screenshot shows the Microsoft Azure Storage blade. On the left sidebar, under the 'STORAGE' section, the 'nitindemostorage' account is selected and highlighted with a red box. A large red arrow points down from the account name to the 'MANAGE ACCESS KEYS' button, which is also highlighted with a red box. The main table lists four storage accounts:

NAME	STATUS	LOCATION	SUBSCRIPTION
devtestvhdsabc968c1f9f7	✓ Online	East Asia	Visual Studio Enterprise with MSDN
nitincloudservices	✓ Online	East Asia	Visual Studio Enterprise with MSDN
nitindemostorage	✓ Online	East Asia	Visual Studio Enterprise with MSDN
portaldvhds92b1d9e2wp	✓ Online	East Asia	Visual Studio Enterprise with MSDN

Copy the “**Storage Account Name**” and “**Primary Access Key**” from here and paste it over there.

Manage Access Keys

When you regenerate your storage access keys, you need to update any virtual machines, media services, or applications that access this storage account to use the new keys. [Learn more](#)



STORAGE ACCOUNT NAME
nitindemostorage

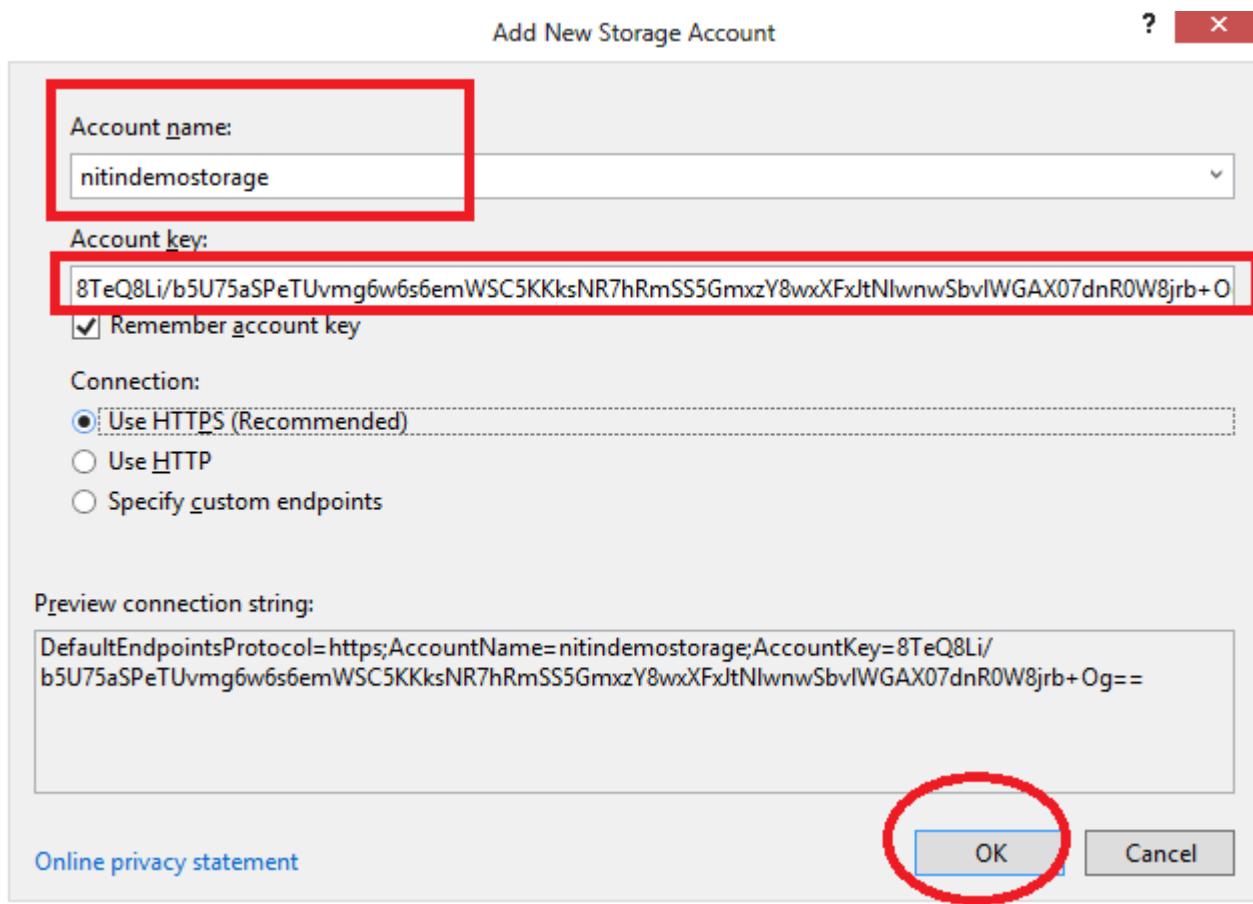
PRIMARY ACCESS KEY
jH/5WgOZzQ7wSQv6BSZ2fbJW3sSEAzimn

SECONDARY ACCESS KEY
FSqlL2Azy19s+C453IXGNqUqjV3mGtxrPtM

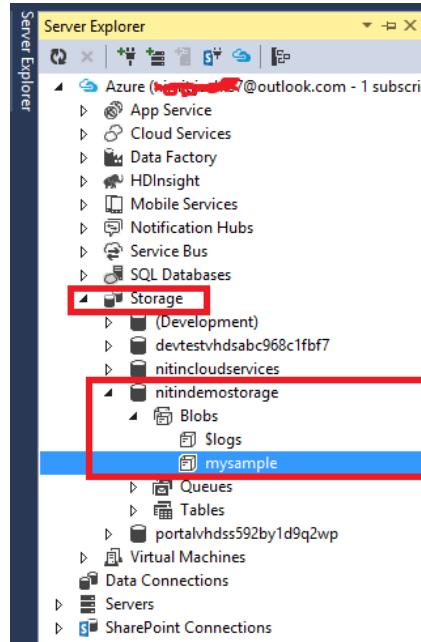
regenerate regenerate



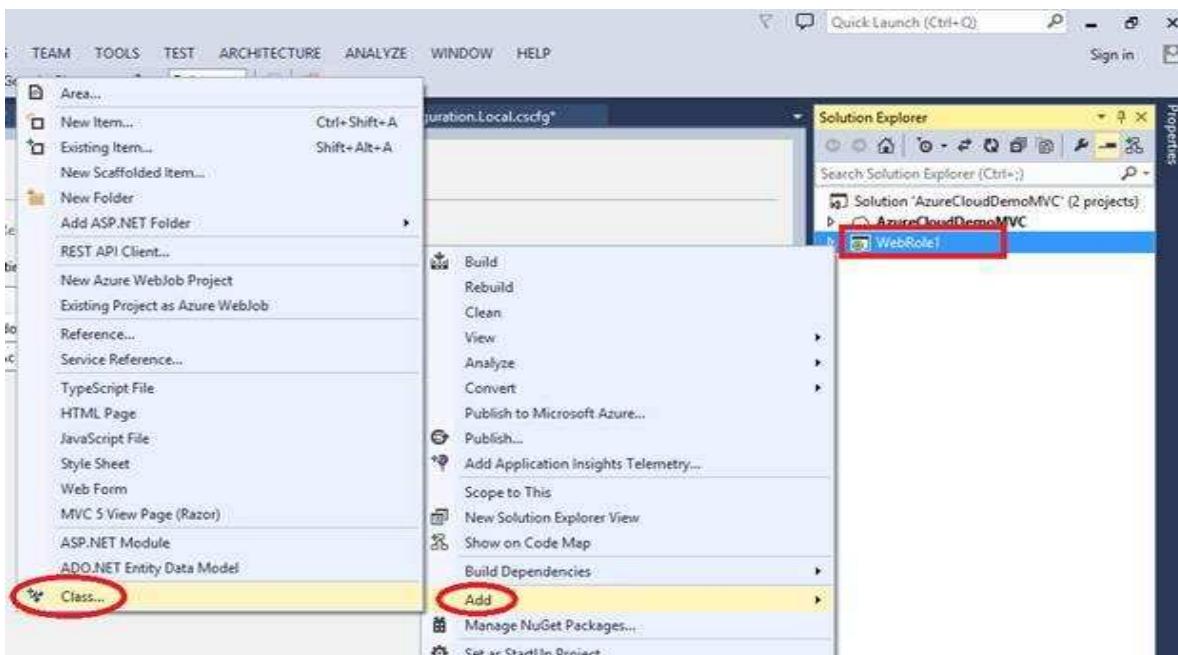
After pasting name and key click on “OK” button.



We got my storage account ”**nitindemostorage**” and in this we have a “Blob” and “mysample” named Blob Container.

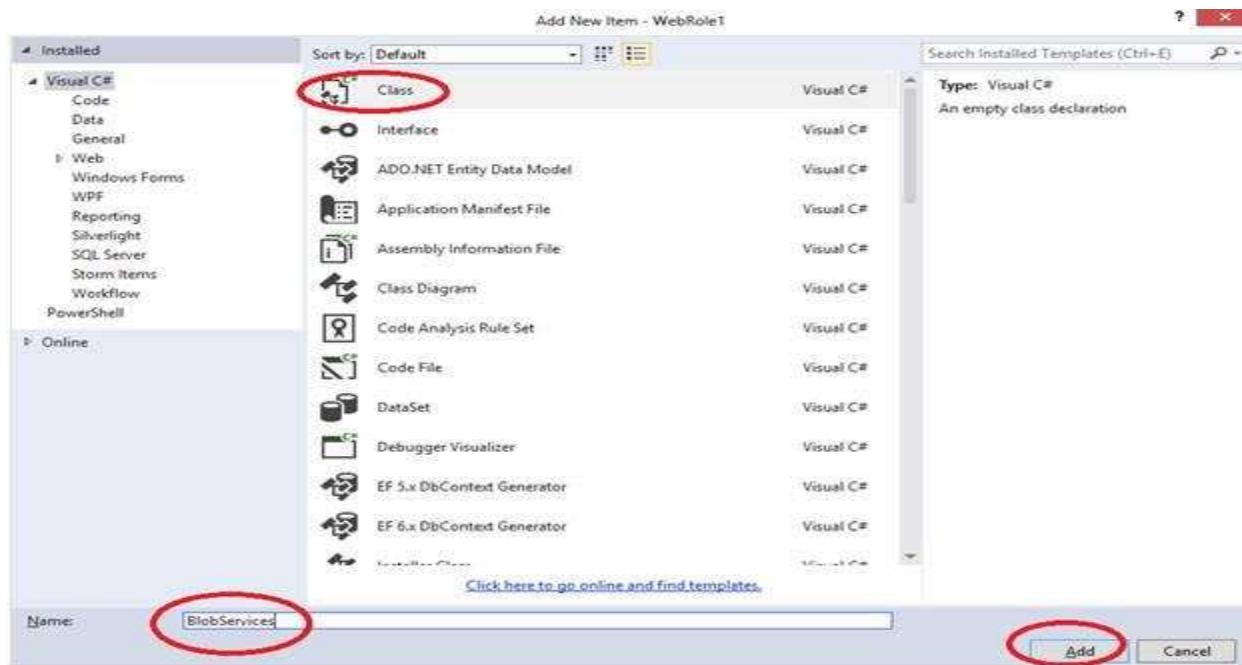


Now go in to project which is “**WebRole1**”, right click on project, select “**Add**” and then select a “**Class**”.

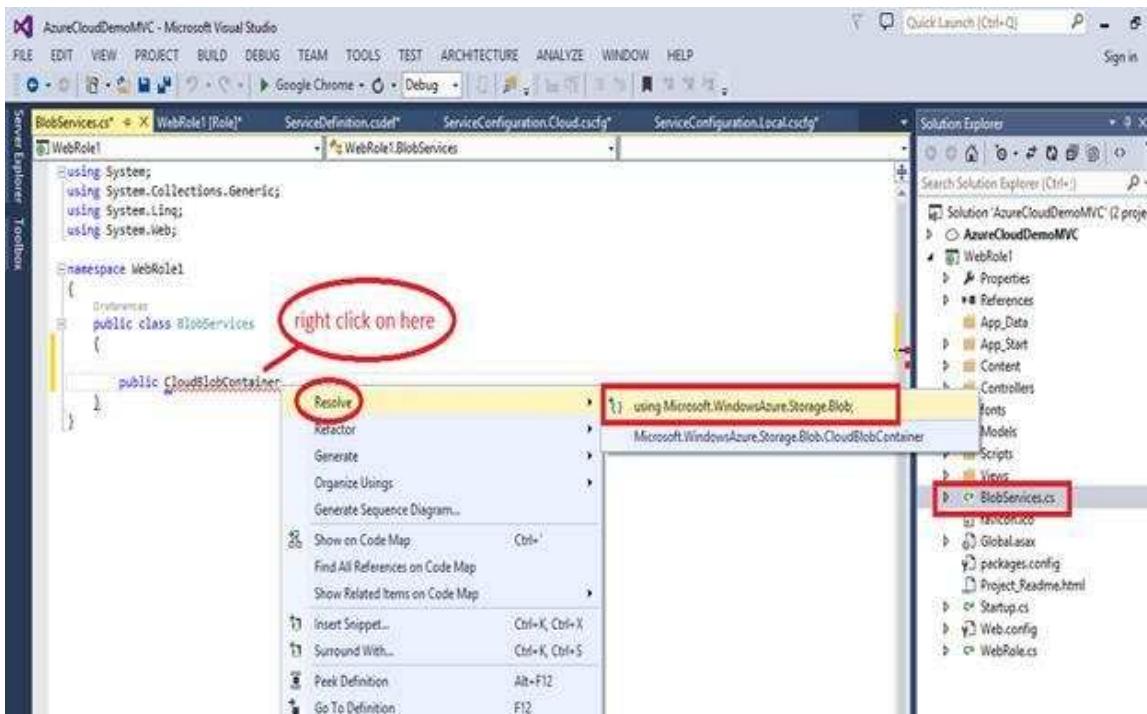


Give a name to your class, whatever you want; here I have given it “**BlobServices**”.

Then click on “**Add**” button.



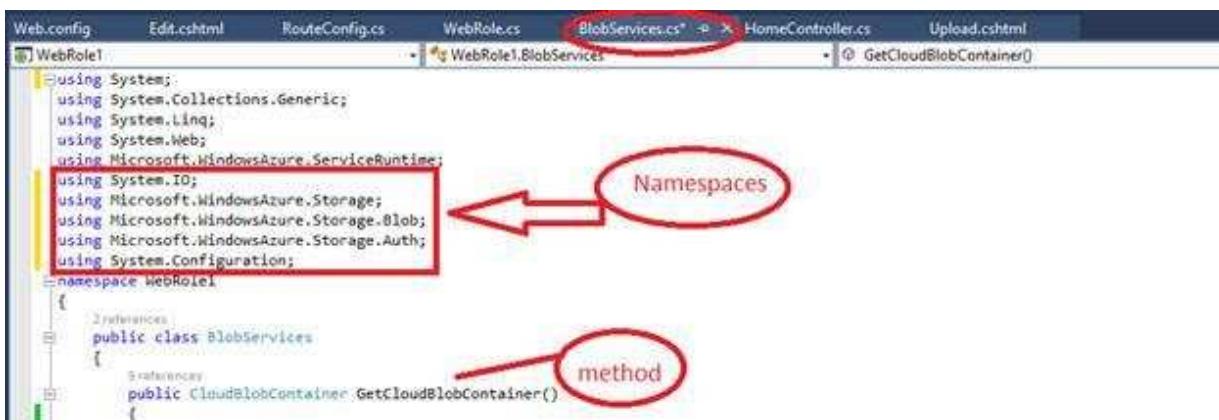
Now, let's go ahead and add a method that will get cloud blob container information. Right click near “CloudBlobContainer” and import a namespace, select “Resolve” and click on “**using Microsoft.WindowsAzure.Storage.Blob**”.



We need to write some important namespaces, so go ahead and type the namespaces.

```
using System.IO;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Blob;
using Microsoft.WindowsAzure.Storage.Auth;
using System.Configuration;
```

The method will be “GetCloudBlobContainer” as in the following screenshot:



Here, I have creating a string to give the string of my storage account. You can copy the connection string from your Azure Explorer. So go in Azure Explorer and copy the string from Azure Storage.

```
public CloudBlobContainer GetCloudBlobContainer()
{
    string connString = "DefaultEndpointsProtocol=https;AccountName=nitindemostorage;AccountKey=jH/5WgOZzQ7wSQv6BSZ2fbJW3sSEAzimnOUhorUWFpfJSUKPUopAD5uDIP3lvJuJmnw1hBoD2RFSzHIPYt2nLw==";
    string destContainer = "mysample";

    // Get a reference to the storage account
    CloudStorageAccount storageAccount = CloudStorageAccount.Parse(connString);
    CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
    CloudBlobContainer blobContainer = blobClient.GetContainerReference(destContainer);
    if (blobContainer.CreateIfNotExists())
    {
        blobContainer.SetPermissions(new BlobContainerPermissions
        {
            PublicAccess = BlobContainerPublicAccessType.Blob
        });
    }
    return blobContainer;
}
```

And “**mysample**” is my Blob Container in the Cloud.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Microsoft.WindowsAzure.ServiceRuntime;
using System.IO;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Blob;
using Microsoft.WindowsAzure.Storage.Auth;
using System.Configuration;
namespace WebRole1
{
    public class BlobServices
    {
        public CloudBlobContainer GetCloudBlobContainer()
        {
            string connString = "DefaultEndpointsProtocol=https;AccountName=nitindemostorage;AccountKey=jH/Swg0IZzQ7wSQv6BSZ2fbJW3sSEAZ";
            string destContainer = "mysample";
            // Get a reference to the storage account
            CloudStorageAccount storageAccount = CloudStorageAccount.Parse(connString);
            CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
            CloudBlobContainer blobContainer = blobClient.GetContainerReference(destContainer);
            if (blobContainer.CreateIfNotExists())
            {
                blobContainer.SetPermissions(new BlobContainerPermissions { PublicAccess = BlobContainerPublicAccessType.Blob });
            }
            return blobContainer;
        }
    }
}

```

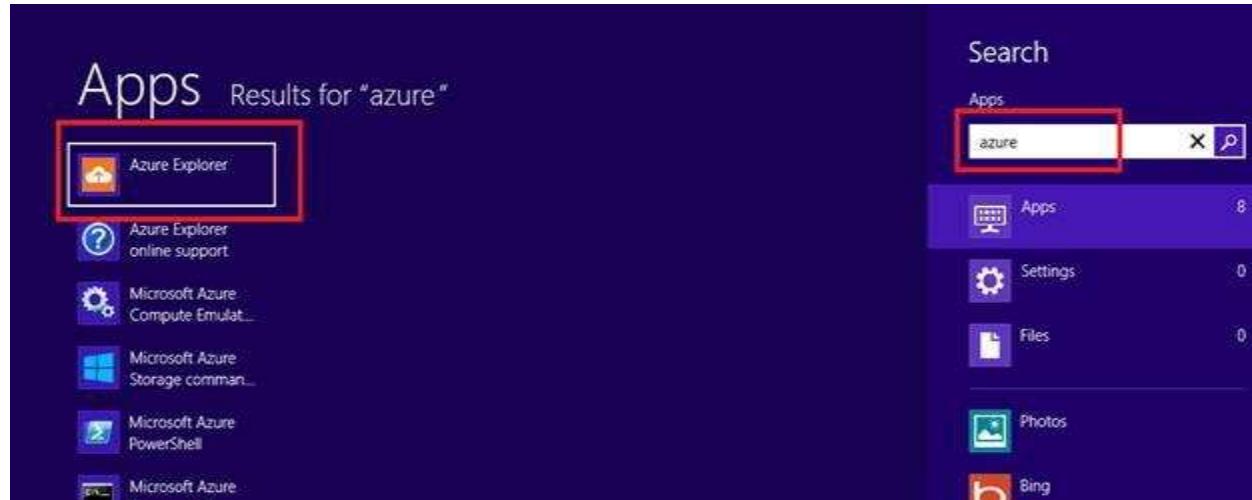
My storage account string

↓

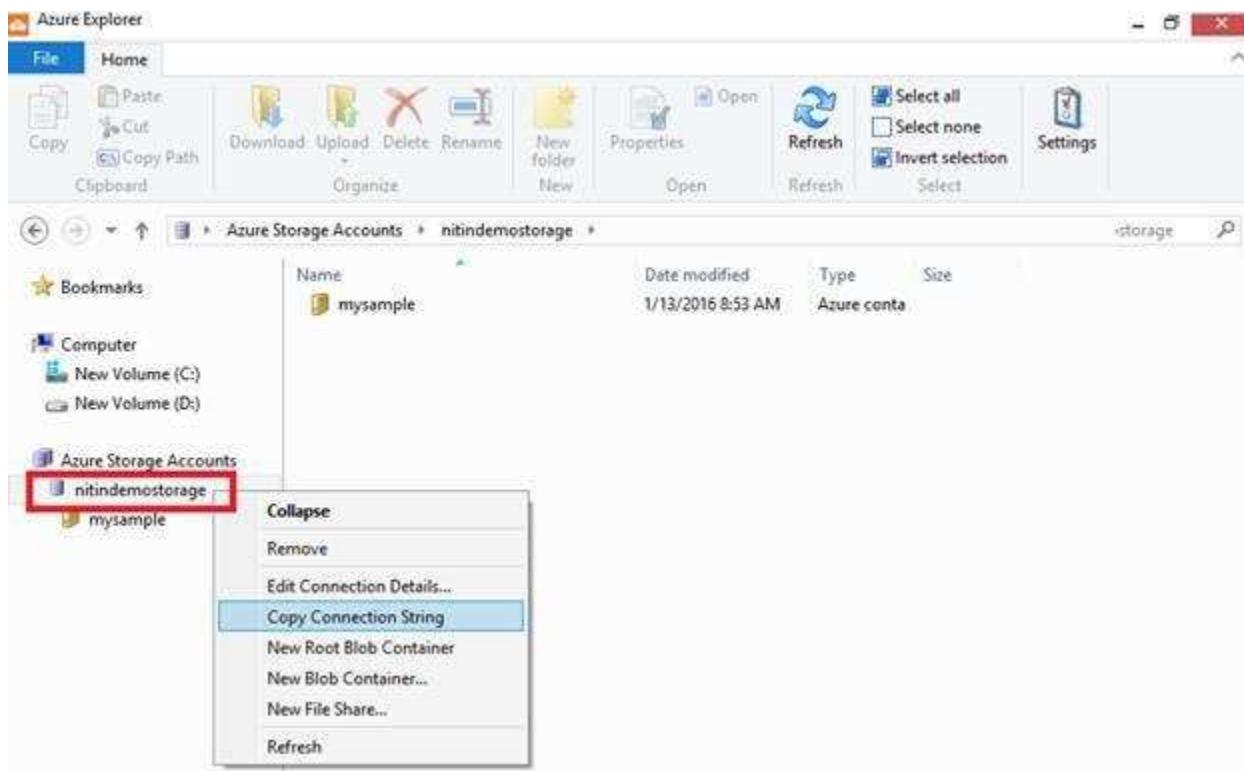
My Blob Container

Now go into Azure Explorer and copy the connection string from storage account.

Go into search box and type Azure Explorer.



Now right click on Azure Storage account that is “**nitindemostorage**” and select “**Copy Connection string**” to copy it. Paste the connection string in the “**string connString**”.



Now I am going to use here,

CloudBlobClient:- To create the cloud blob service client.

CloudBlobContainer: To check if there is any Blob Container of “mysample” named in the storage account.

And here I have used if condition to check if there is any Blob Container, so just go in to the storage account and create a Blob Storage of name “**mysample**”.

And finally it will return Blob Container. Container is nothing but just a folder in the cloud.

If there is a blob container then it will use existing.

```

namespace WebRole1
{
    public class BlobServices
    {
        public CloudBlobContainer GetCloudBlobContainer()
        {
            string connString = "DefaultEndpointsProtocol=https;AccountName=nitinstorage;AccountKey=jH/Swg0ZzQ7vSQv68S22fbJW3s5EAzi";
            string destContainer = "mysample";

            // Get a reference to the storage account
            CloudStorageAccount storageAccount = CloudStorageAccount.Parse(connString);
            CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
            CloudBlobContainer blobContainer = blobClient.GetContainerReference(destContainer);
            if (blobContainer.CreateIfNotExists())
            {
                blobContainer.SetPermissions(new BlobContainerPermissions { PublicAccess = BlobContainerPublicAccessType.Blob });
            }
            return blobContainer;
        }
    }
}

```

Now go ahead and use this setup in my Controller.

Go to “HomeController.cs”.

I’m going to add a new Action method inside this “HomeController”.

Blob Services class is nothing but a “BlobServices.cs” class which is used in the previous “BlobServices.cs” class.

We have created a “**Upload()**” method with ActionResult Return type.

Inside the “**Upload()**”, create “CloudBlobContainer” and list name all the blobs and “_blobServices” is object of my class.

Inside this Action Result, list available blobs, Uri in the container and return it to the view page to display all the blobs.

Means if you upload the blobs which is image it will list all the images and look through the images and return the Uri of the images to the view page.

```

BlobServices _blobServices = newBlobServices();
public ActionResult Upload()
{
    CloudBlobContainer blobContainer = _blobServices.GetCloudBlobContainer();
    List < string > blobs = newList < string > ();
    foreach(var blobItem in blobContainer.ListBlobs())
    {
        blobs.Add(blobItem.Uri.ToString());
    }
    return View(blobs);
}

```

Now, I'm going to implement its post version means user will be able to upload his Blob also, publicActionResult Upload(FormCollectionimage)- It will collect the posted file image information, and upload it to the Blob Container and at the end this will redirect to the same view to make recently image visible to user.

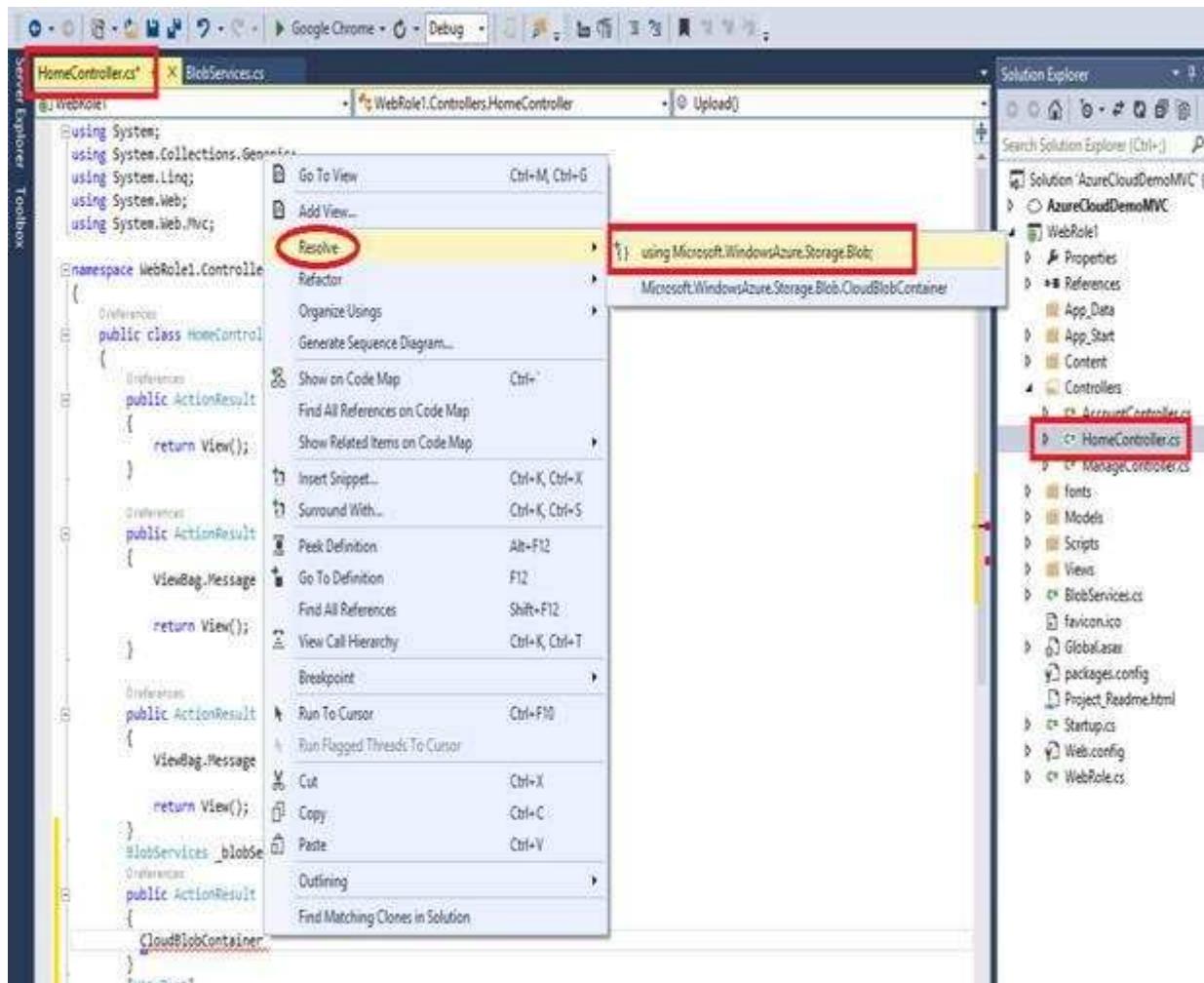
```

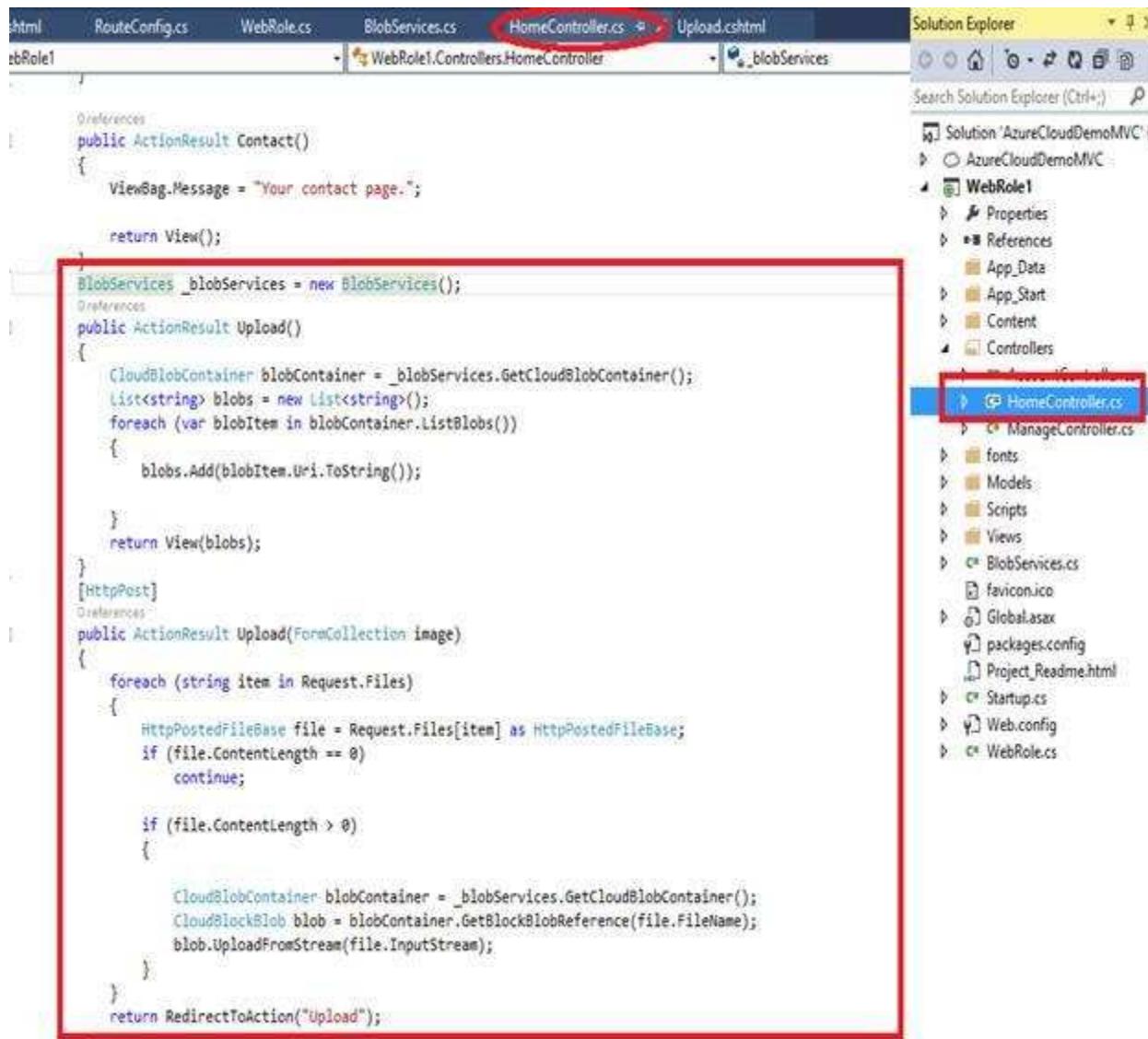
[HttpPost]
public ActionResult Upload(FormCollection image)
{
    foreach(string item in Request.Files)
    {
        HttpPostedFileBase file = Request.Files[item] asHttpPostedFileBase;
        if (file.ContentLength == 0)
            continue;

        if (file.ContentLength > 0)
        {
            CloudBlobContainer blobContainer = _blobServices.GetCloudBlobContainer();
            CloudBlockBlob blob = blobContainer.GetBlockBlobReference(file.FileName);
            blob.UploadFromStream(file.InputStream);
        }
    }
    return RedirectToAction("Upload");
}

```

When we write “CloudBlobContainer” we need to resolve it by right clicking near “CloudBlobContainer”. So simply go to **Resolve** and select “usingMicrosoft.WindowsAzure.Storage.Blob;”





```

    public ActionResult Contact()
    {
        ViewBag.Message = "Your contact page.";

        return View();
    }

    BlobServices _blobServices = new BlobServices();

    public ActionResult Upload()
    {
        CloudBlobContainer blobContainer = _blobServices.GetCloudBlobContainer();
        List<string> blobs = new List<string>();
        foreach (var blobItem in blobContainer.ListBlobs())
        {
            blobs.Add(blobItem.Uri.ToString());
        }
        return View(blobs);
    }

    [HttpPost]
    public ActionResult Upload/FormCollection image)
    {
        foreach (string item in Request.Files)
        {
            HttpPostedFileBase file = Request.Files[item] as HttpPostedFileBase;
            if (file.ContentLength == 0)
                continue;

            if (file.ContentLength > 0)
            {
                CloudBlobContainer blobContainer = _blobServices.GetCloudBlobContainer();
                CloudBlockBlob blob = blobContainer.GetBlockBlobReference(file.FileName);
                blob.UploadFromStream(file.InputStream);
            }
        }
        return RedirectToAction("Upload");
    }
}

```

Now we need to create one more method to delete the blobs, so let's go ahead and create a method for deleting the blob.

You can see, this method will accept name as a parameter and return stream message which is String and will return “File Successfully Deleted”.

“Name” is nothing but “Uri” of the images, Uri means the complete address of the image.

“uri.LocalPath” is a path that contains the actual description of the images.

```
[HttpPost]
public string DeleteImg(string Name)
{
    Uri uri = newUri(Name);
    string filename = System.IO.Path.GetFileName(uri.LocalPath);
    CloudBlobContainer blobContainer = _blobServices.GetCloudBlobContainer();
    CloudBlockBlob blob = blobContainer.GetBlockBlobReference(filename);
    blob.Delete();
    return "File Successfully Deleted";
}
```

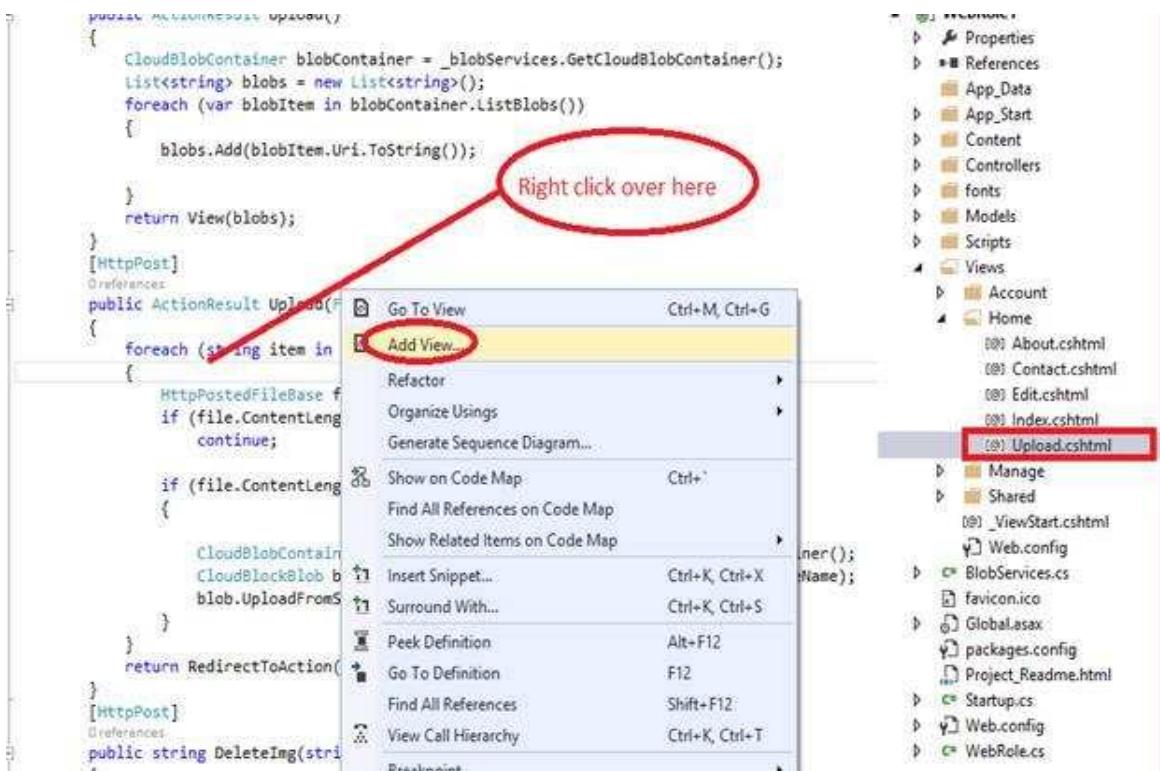


The screenshot shows the Visual Studio IDE. On the left is the code editor with the `HomeController.cs` file open. A red box highlights the `DeleteImg` method. On the right is the Solution Explorer showing the project structure:

- Content
 - Controllers
 - AccountController.cs
 - HomeController.cs
 - ManageController.cs
 - fonts
 - Models
 - Scripts
 - Views
- BlobServices.cs
- favicon.ico
- Global.asax
- packages.config
- Project_Readme.html
- Startup.cs
- Web.config
- WebRole.cs

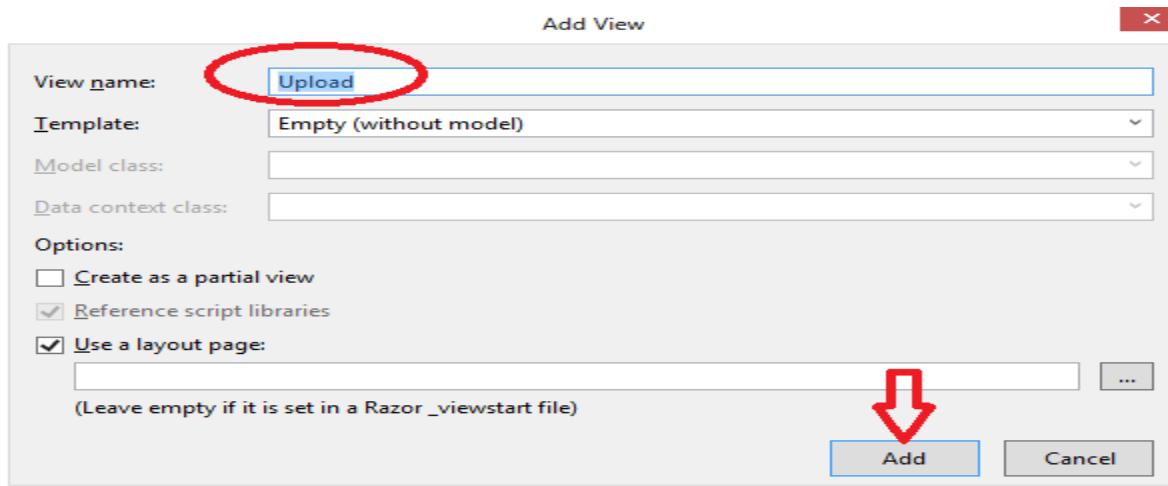
Now we need to add a view for the design that design will be shown on our browser, means that we want to show in the browser.

Simply go near to for each statement and click right over there and go to “**Add View**”, for adding a view.



In the following image, in the view name section, give any name to view whatever you want, here I have given it "Upload".

Click on "Add" button to add a view.



After adding a view, in the paragraph tag <P> </P>, I am creating a form section and using Razor syntax here.

Upload: My ActionResult

Home: My HomeController, and form method will be Post,

```
@using(Html.BeginForm("Upload", "Home", FormMethod.Post, new
{
    enctype = "multipart/form-data"
}))
{
    enctype = "multipart/form-data"
}
```

} - This code defines;

//we can select the multi part of the image.

Now **in** the < div > section, I'm going to print a message that **is** " Upload Image".

Now I'm going to create some Html tags. < inputtype = "file"

name = "avatar" / > -This will allows us to upload an image. < inputtype = "submit"

value = "upload" / > -This **is** upload a button.

I have created a table also for image and delete, image for Images and Delete for the Delete. So create a table class and in the "style" you can give the widths according to your choice. I have given 50% to image and 25% to delete.

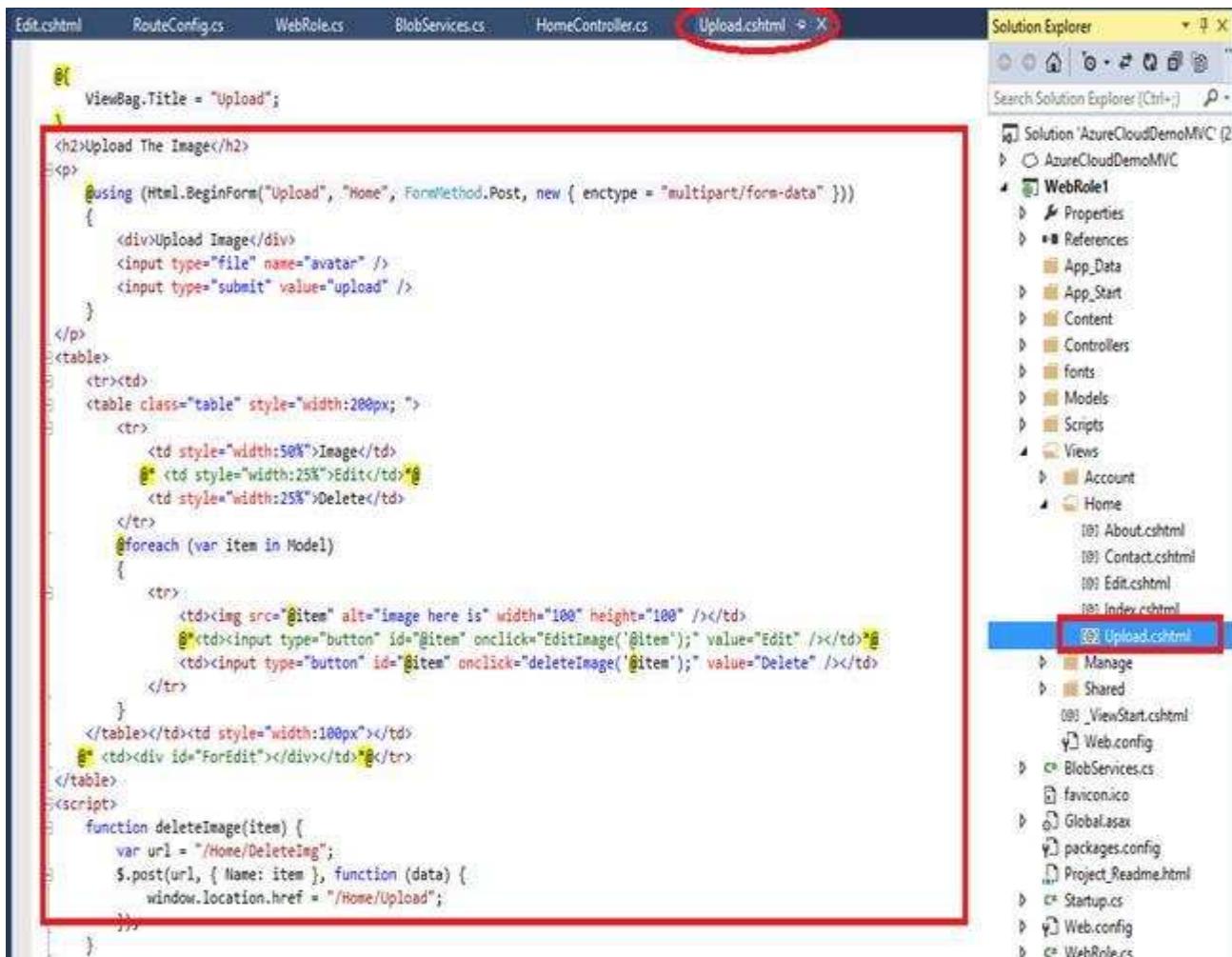
Here, I'm going to create also a script to delete button. So create a function "deleteimage(item)" and in the URL section give the Url of DeleteImage.

```
@ {
    ViewBag.Title = "Upload";
} < h2 > Upload The Image < /h2 > < p >
@using(Html.BeginForm("Upload", "Home", FormMethod.Post, new
{
    enctype = "multipart/form-data"
})) { < div > Upload Image < /div > < inputtype = "file"
    name = "avatar" / >
        < inputtype = "submit"
    value = "upload" / >
} < /p > < table >
< tr > < td >
< tableclass = "table"
style = "width:200px; " >
< tr >
    < tdstyle = "width:50%" > Image < /td > < tdstyle = "width:25%" > Delete < /td > < /tr >
@foreach(var item in Model) { < tr >
    < td > < imgsrc = "@item"
    alt = "image here is"
```

```

width = "100"
height = "100" / > </td> <td> <input type = "button"
id = "@item"
onclick = "deleteImage('@item');"
value = "Delete" / > </td> </tr>
} </table></td> <td style = "width:100px" > </td> </table> <script>
function deleteImage(item) {
var url = "/Home/DeleteImg";
$.post(url,
{
    Name: item
}, function(data)
{
    window.location.href = "/Home/Upload";
});
} </script>

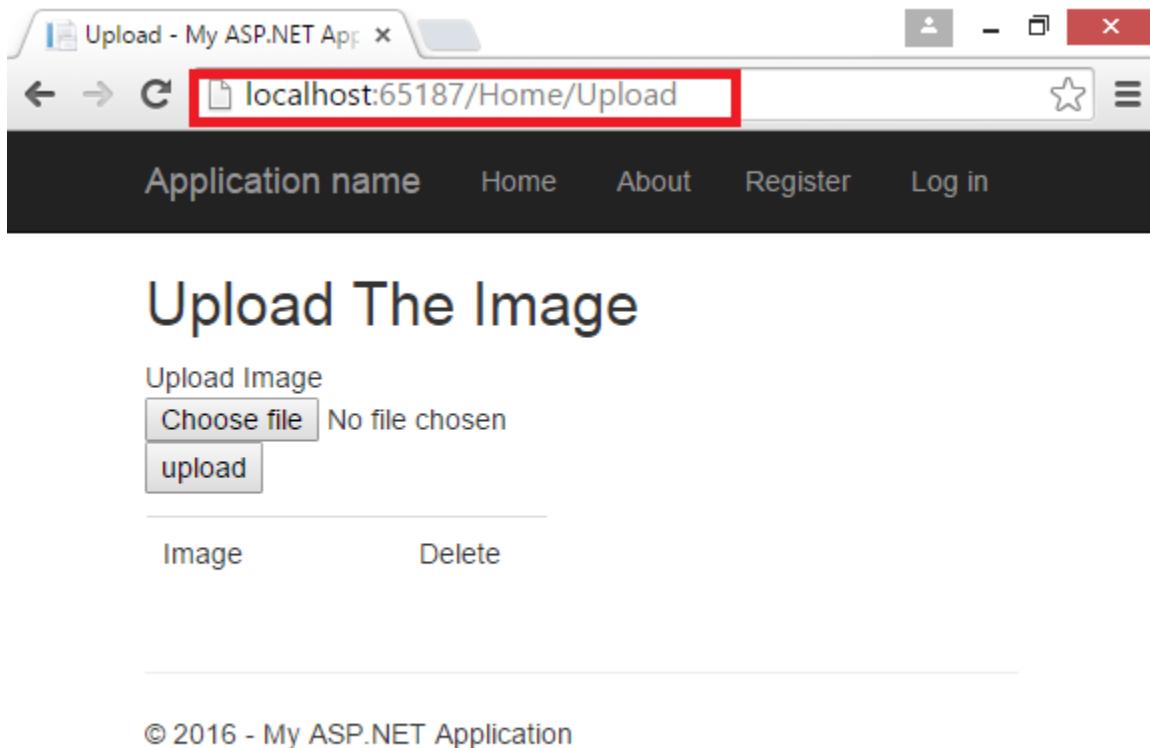
```



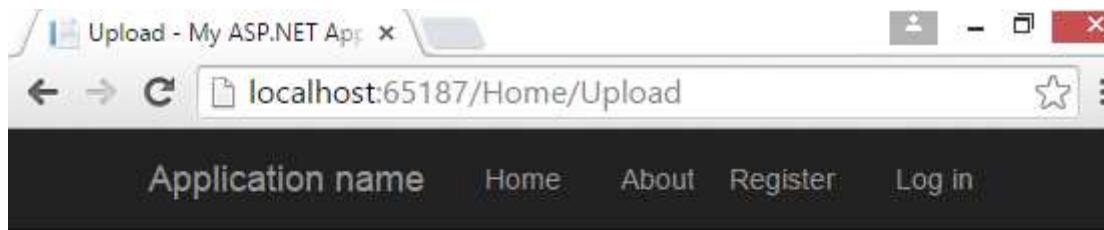
The screenshot shows the Visual Studio IDE interface. The top navigation bar includes 'Edit.cshtml', 'RouteConfig.cs', 'WebRole.cs', 'BlobServices.cs', 'HomeController.cs', and 'Upload.cshtml' (which is highlighted with a red oval). The main area displays the 'Upload.cshtml' code, which contains HTML for an upload form and a table for displaying items with edit and delete buttons. A red rectangle highlights the section of the code where the 'Delete' button's onclick event is defined. The 'Solution Explorer' pane on the right lists the project structure, including 'WebRole1' with its files like 'Properties', 'References', 'App_Start', 'Content', 'Controllers', 'Fonts', 'Models', 'Scripts', and 'Views' (which contains 'Account', 'Home' with 'About.cshtml', 'Contact.cshtml', 'Edit.cshtml', and 'Index.cshtml'). The 'Upload.cshtml' file is also listed under the 'Views' folder.

Finally, we successfully created a “**Delete**” button to delete the image and “upload” button to upload an image. Now let’s check our output, Build your project by Ctrl+Shift+B and after successful build you can run it by pressing F5 key.

Finally, I got my Output. Click on “**Choose file**” button and click on “**upload**” button. And choose your favorite image which you want to upload in the cloud as well as in the Azure Explorer and to show it on the browser.



I have successfully uploaded my image, now go ahead and check on its changes on cloud and as well as Azure Explorer.



Upload The Image

Upload Image

Choose file No file chosen

Image

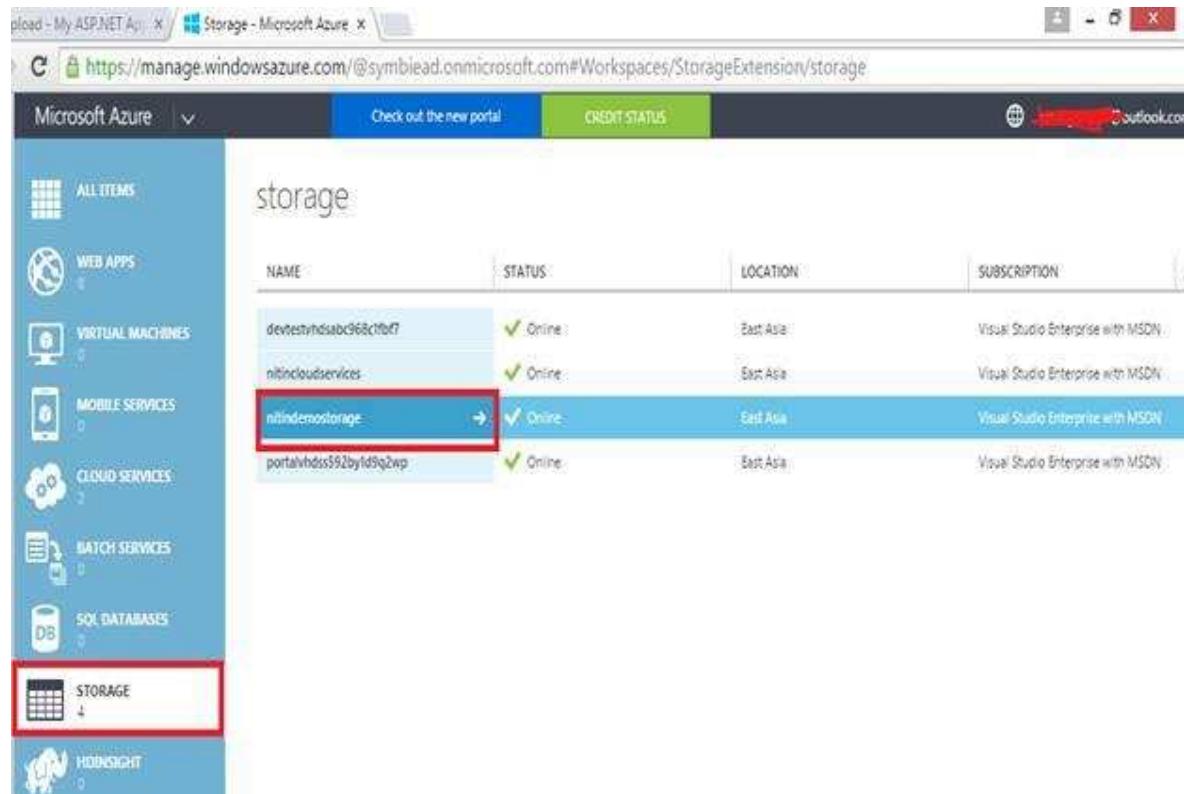
Delete



© 2016 - My ASP.NET Application

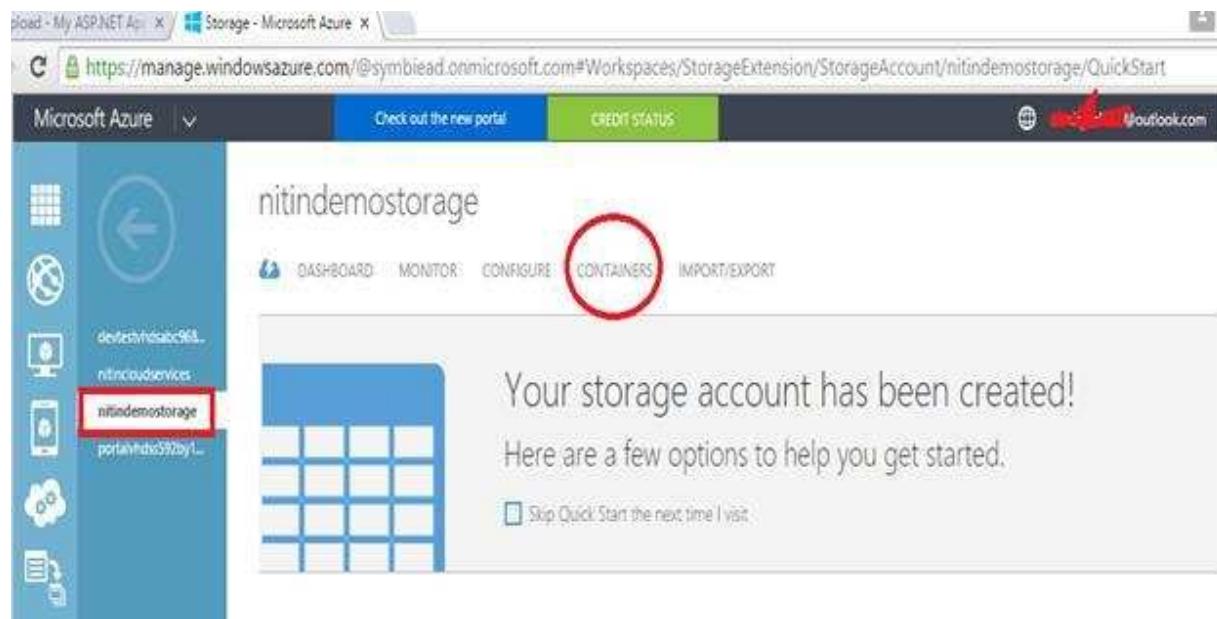
Sign In Azure Portal and go into Storage Account and select the Blob Container.

Go to “**STORAGE**”, and select “**nitindemostroage**”.



NAME	STATUS	LOCATION	SUBSCRIPTION
devtestvhdsabc968cfbf7	✓ Online	East Asia	Visual Studio Enterprise with MSDN
nitindcloudservices	✓ Online	East Asia	Visual Studio Enterprise with MSDN
nitindemostorage	✓ Online	East Asia	Visual Studio Enterprise with MSDN
portalvhds592byld9q2wp	✓ Online	East Asia	Visual Studio Enterprise with MSDN

In the “**nitindemostorage**”, select the “**CONTAINERS**”.



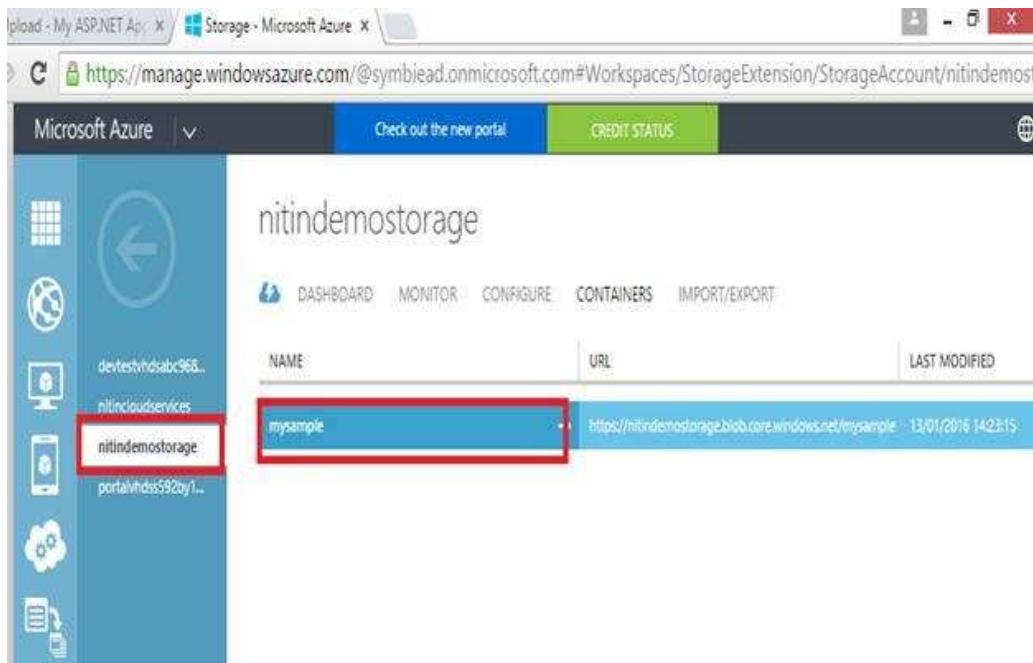
DASHBOARD MONITOR CONFIGURE CONTAINERS IMPORT/EXPORT

Your storage account has been created!
Here are a few options to help you get started.

Skip Quick Start the next time I visit

After clicking on the “CONTAINERS”, you will see your Blob Container; my Blob Container is “mysample”.

Click on “mysample”.



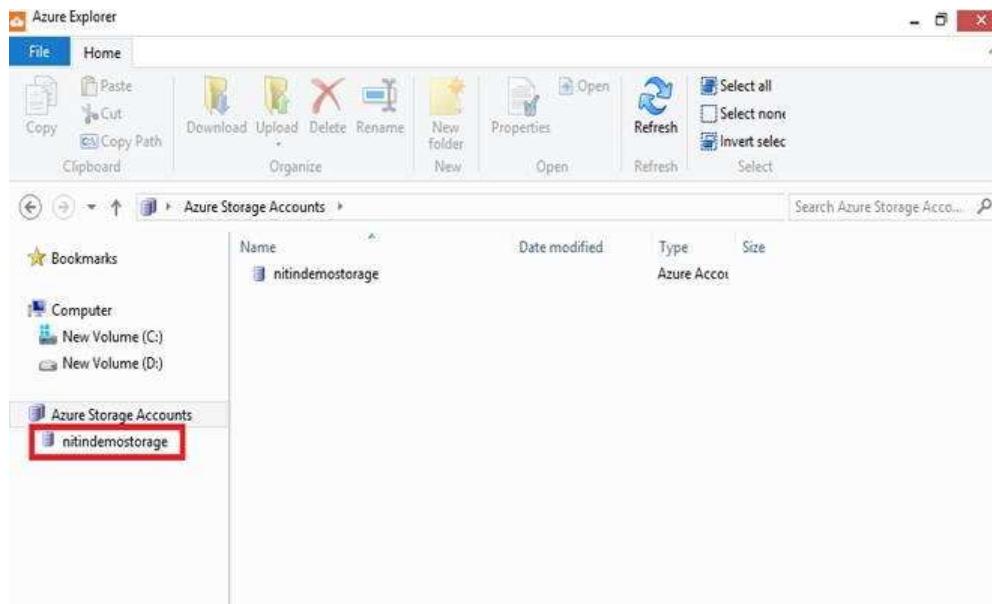
NAME	URL	LAST MODIFIED
mysample	https://nitindemostorage.blob.core.windows.net/mysample	13/01/2016 14:23:15

Finally, we can see the changes in my Cloud account, I got my output simply. We can see there ARE three images which can be seen in the browser as well.

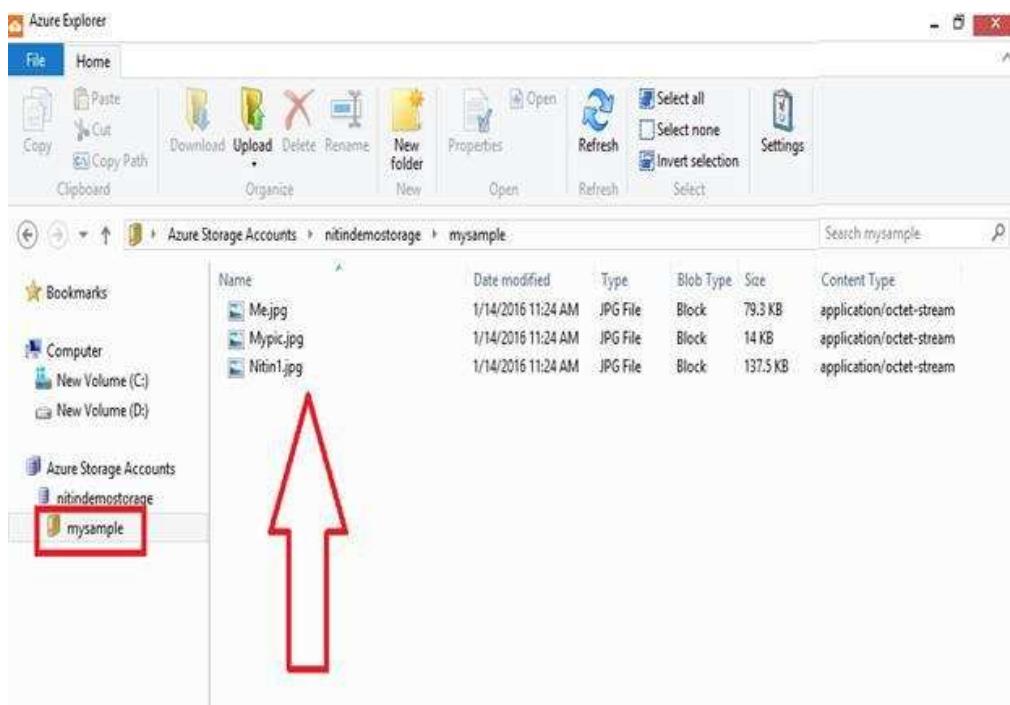


NAME	URL
Me.jpg	https://nitindemostorage.blob.core.windows.net/mysample/Me.jpg
Mypic.jpg	https://nitindemostorage.blob.core.windows.net/mysample/Mypic.jpg
Nitin1.jpg	https://nitindemostorage.blob.core.windows.net/mysample/Nitin1.jpg

Now go in to the Azure Explorer to see the above changes.



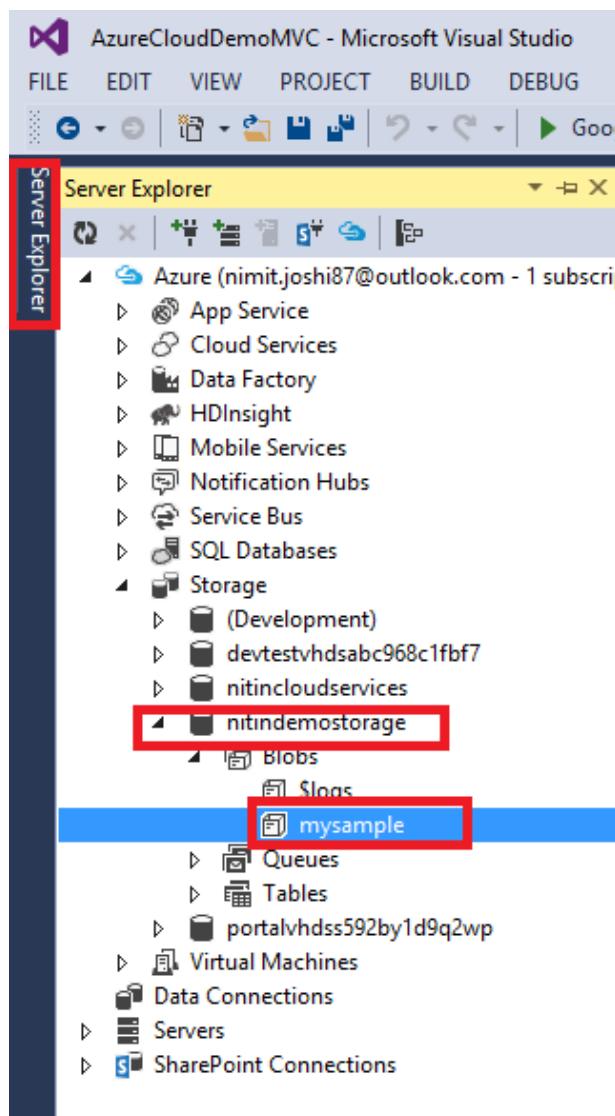
In the following figure, we can check our output as well. Here also three images with the same name means all changes on the browser will be made on cloud as well as Azure Explorer.



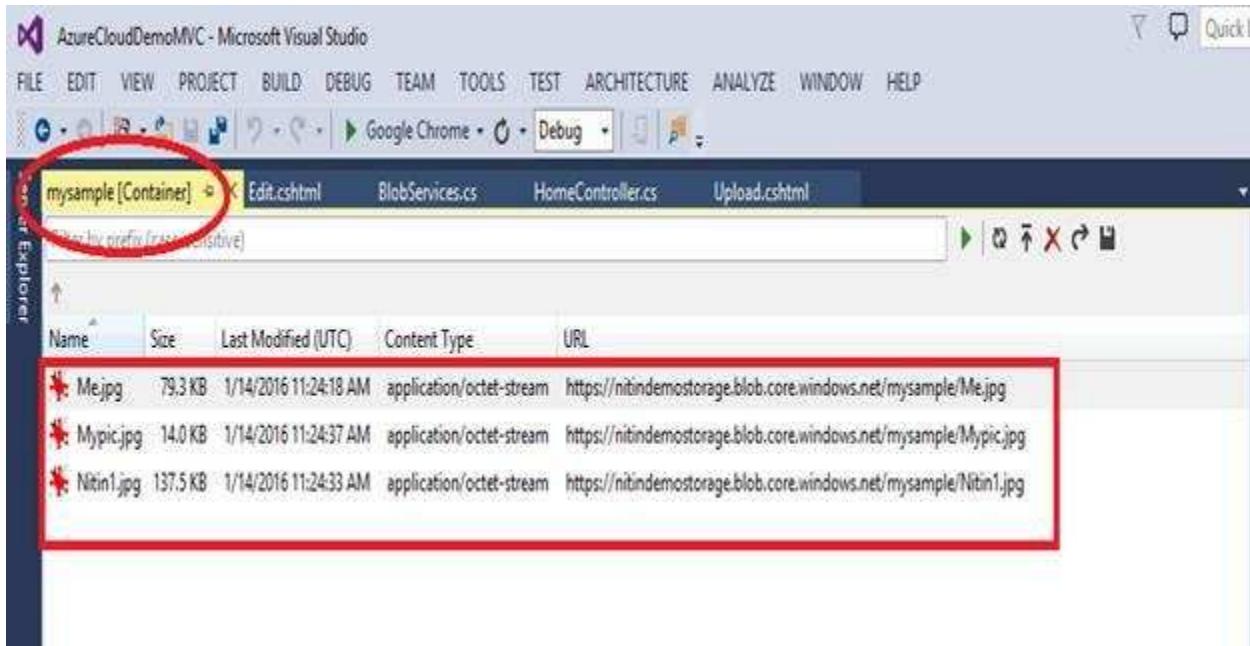
Now we can check the same output in Server Explorer as well in Visual Studio.

In the Server Explorer, select your storage accounts, my account is “**nitindemostorage**”, go in Blobs and then “**mysample**” Container.

Double click on “mysample” container to see the changes.



Now here also you can see the same output as well.

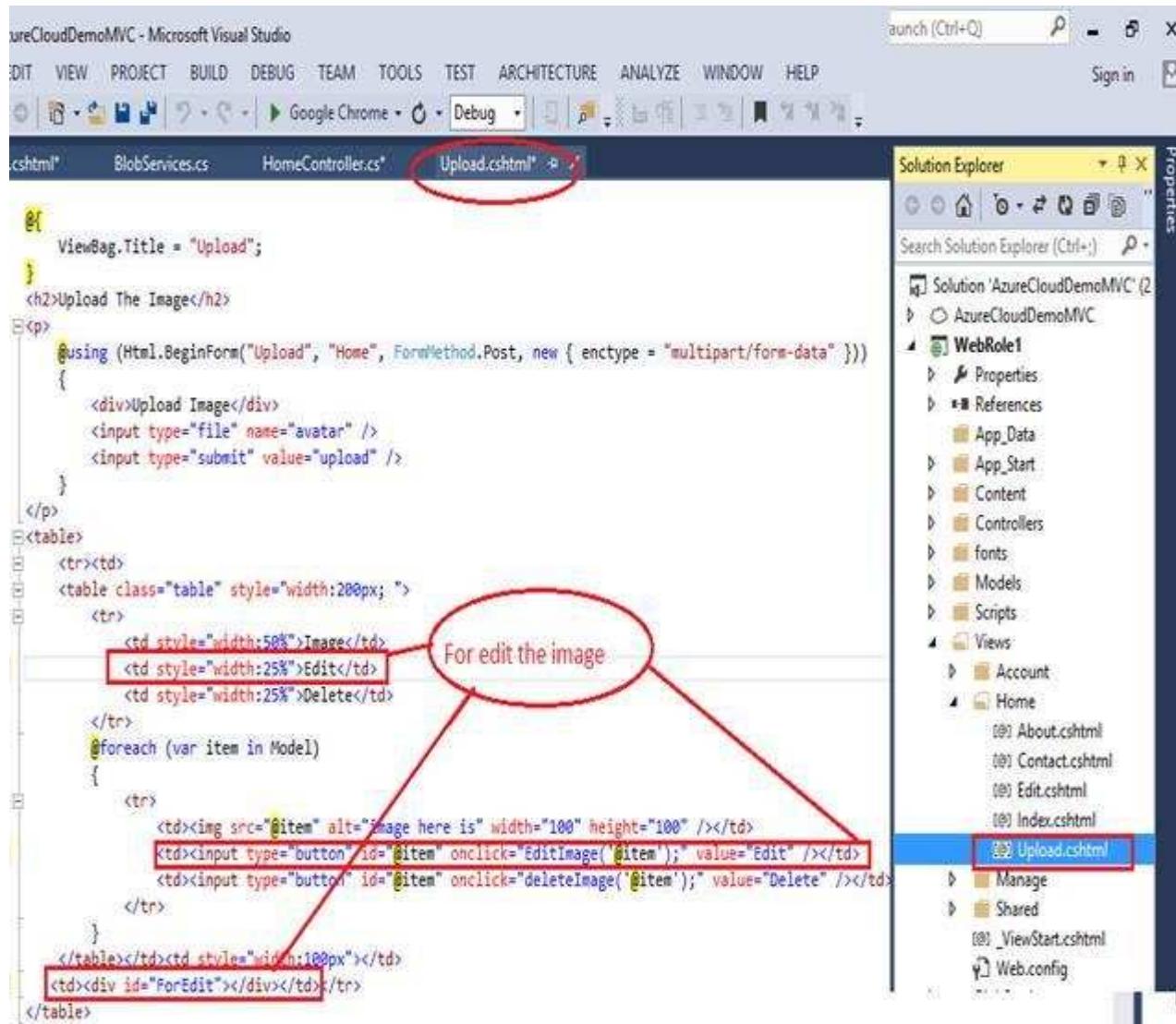


In the “**Upload.cshtml**”, now we have to create an edit button to edit an image.

```
<td style="width:25%>
    Edit</td>
    <td>
        <input type="button" id="@item" onclick="EditImage('@item');" value="Edit" />
    </td>
```

This Html Code will create an “**Edit**” button to the browser and just a div to load a partial view for Editing an image by jQuery.

```
<td>
    <div id="ForEdit">
    </div>
</td>
```



The screenshot shows the Microsoft Visual Studio interface with the 'Upload.cshtml' file open in the editor. The 'Solution Explorer' window on the right shows the project structure for 'AzureCloudDemoMVC' with 'WebRole1' selected. The 'Upload.cshtml' file is highlighted in the Solution Explorer. A red circle highlights the 'Upload.cshtml' tab in the top navigation bar. A red callout points to a button in the code with the text 'For edit the image'.

```

@{
    ViewBag.Title = "Upload";
}


## Upload The Image



using (Html.BeginForm("Upload", "Home", FormMethod.Post, new { enctype = "multipart/form-data" }))
    {
        <div>Upload Image</div>
        <input type="file" name="avatar" />
        <input type="submit" value="upload" />
    }



|                                                                                                                                                                                                                                                                                                                                                |                                                                                           |                                                                                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                                                                                                                                                                | <input id="@item" onclick="EditImage('@item'); value='Edit'" type="button" value="Edit"/> | <input id="@item" onclick="deleteImage('@item'); value='Delete'" type="button" value="Delete"/> |
| <tr>             <td></td>             <td><input type="button" id="@item" onclick="EditImage('@item'); value='Edit'" value="Edit" /></td>             <td><input type="button" id="@item" onclick="deleteImage('@item'); value='Delete'" value="Delete" /></td>         </tr> |                                                                                           |                                                                                                 |


```

And write a jQuery function to call on that edit button click for requesting that partial view to edit a record.

```

function EditImage(item)
{
    $("#ForEdit").load("/Home/Edit?Name=" + item, function()
    {
        }).css
        ({
            "border-color": "red",
            "border-width": "5px",
            "border-style": "solid"
        });
}

```

You can edit your image in the browser as well, now go for code,

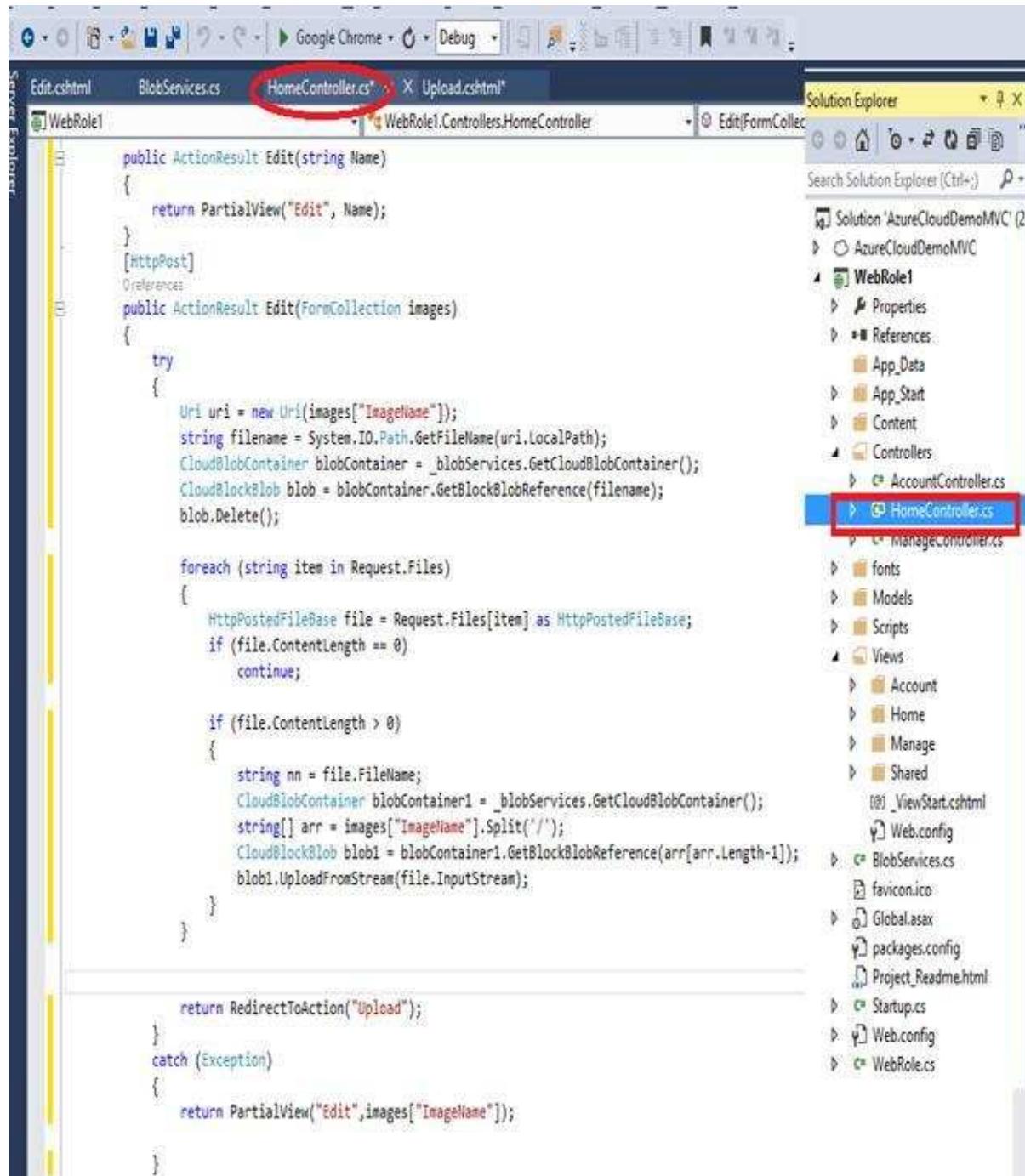
Here we created a new Action “**Edit**” with ActionResult Return type for both type of request [httpget] and also for [httppost]. First one only return a partial view with a string type of Model binder object which contain the image uri which I need to edit and second Action with [httppost] will use when a user will submit the request to server for image editing.

Code:

```
//First Action for [httpGet] request:  
[HttpGet]  
public ActionResult Edit(string Name)  
{  
    return PartialView("Edit", Name);  
}  
//Second Action for [HttpPost] request:  
  
[HttpPost]  
public ActionResult Edit(FormCollection images)  
{  
    try {  
        Uri uri = new Uri(images["ImageName"]);  
        string filename = System.IO.Path.GetFileName(uri.LocalPath);  
        CloudBlobContainer blobContainer = _blobServices.GetCloudBlobContainer();  
        CloudBlockBlob blob = blobContainer.GetBlockBlobReference(filename);  
        blob.Delete();  
  
        foreach(string item in Request.Files)  
        {  
            HttpPostedFileBase file = Request.Files[item] as HttpPostedFileBase;  
            if (file.ContentLength == 0)  
                continue;  
  
            if (file.ContentLength > 0)  
            {  
                string nn = file.FileName;  
                CloudBlobContainer blobContainer1 = _blobServices.GetCloudBlobContainer();  
                string[] arr = images["ImageName"].Split('/');  
                CloudBlockBlob blob1 = blobContainer1.GetBlockBlobReference(arr[arr.Length - 1]);  
                blob1.UploadFromStream(file.InputStream);  
            }  
        }  
        return RedirectToAction("Upload");  
    } catch (Exception)  
}
```

```
{  
    return PartialView("Edit", images["ImageName"]);  
}  
}  
}
```

Also changes can be seen in the following image:

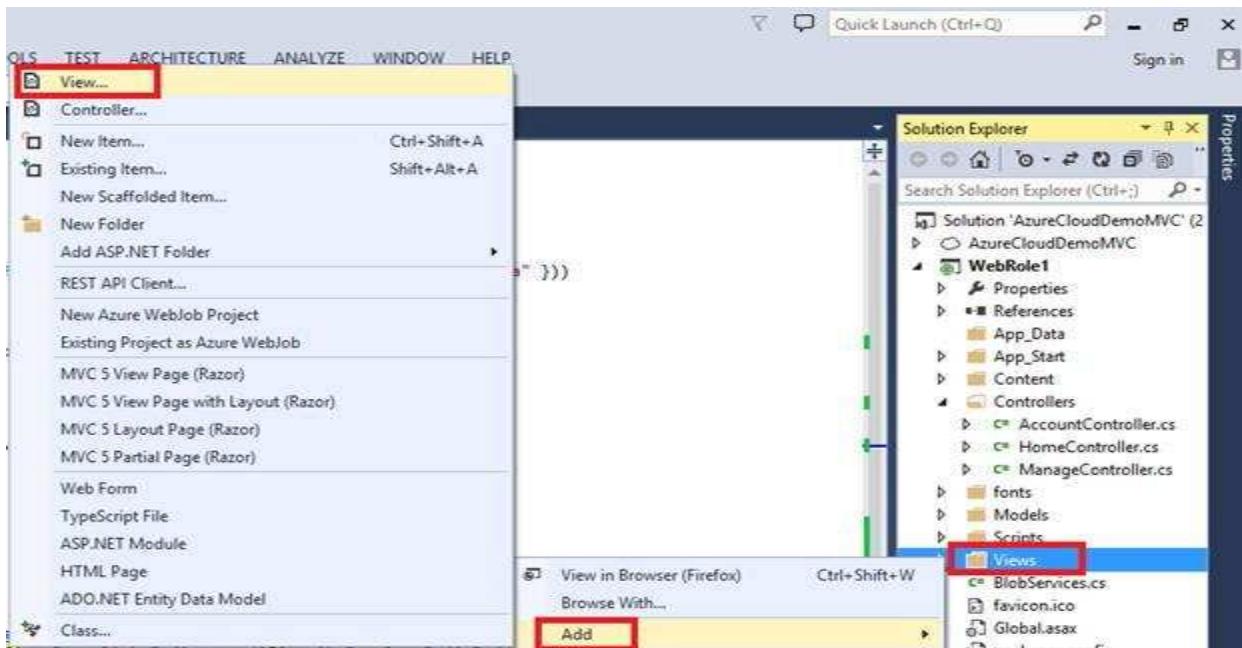


The screenshot shows the Microsoft Visual Studio IDE interface. The title bar indicates the project is 'WebRole1' and the current file is 'HomeController.cs'. The code editor displays the following C# code for the HomeController:

```
public ActionResult Edit(string Name)  
{  
    return PartialView("Edit", Name);  
}  
[HttpPost]  
[ValidateAntiForgeryToken]  
public ActionResult Edit/FormCollection images)  
{  
    try  
    {  
        Uri uri = new Uri(images["ImageName"]);  
        string filename = System.IO.Path.GetFileName(uri.LocalPath);  
        CloudBlobContainer blobContainer = _blobServices.GetCloudBlobContainer();  
        CloudBlockBlob blob = blobContainer.GetBlockBlobReference(filename);  
        blob.Delete();  
  
        foreach (string item in Request.Files)  
        {  
            HttpPostedFileBase file = Request.Files[item] as HttpPostedFileBase;  
            if (file.ContentLength == 0)  
                continue;  
  
            if (file.ContentLength > 0)  
            {  
                string nn = file.FileName;  
                CloudBlobContainer blobContainer1 = _blobServices.GetCloudBlobContainer();  
                string[] arr = images["ImageName"].Split('/');  
                CloudBlockBlob blob1 = blobContainer1.GetBlockBlobReference(arr[arr.Length-1]);  
                blob1.UploadFromStream(file.InputStream);  
            }  
        }  
  
        return RedirectToAction("Upload");  
    }  
    catch (Exception)  
    {  
        return PartialView("Edit", images["ImageName"]);  
    }  
}
```

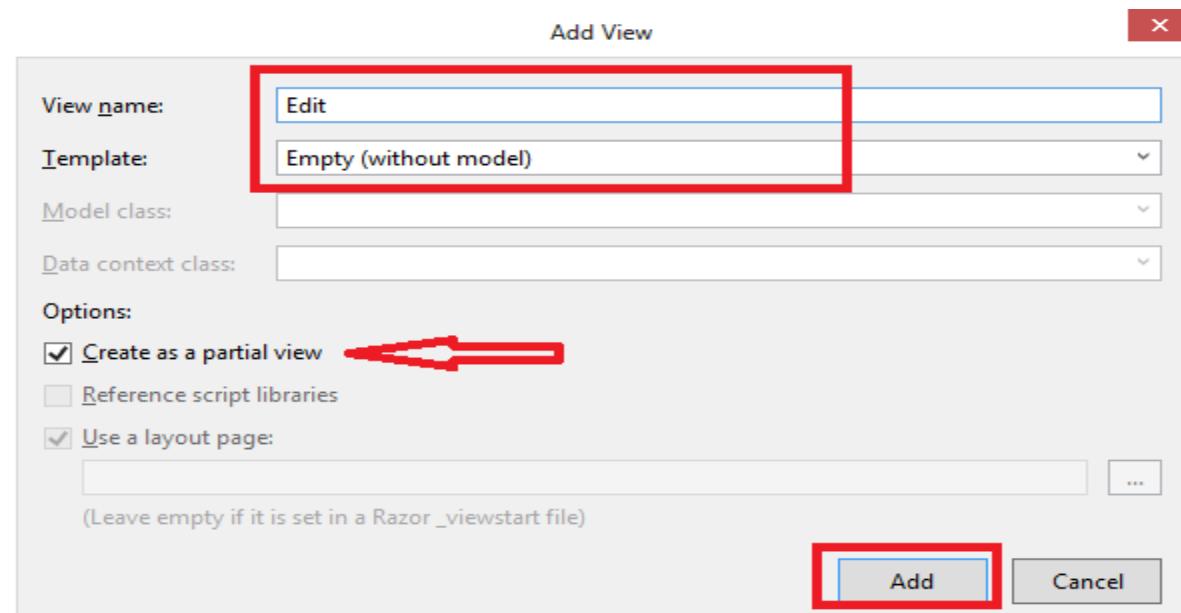
The Solution Explorer on the right side of the interface shows the project structure for 'AzureCloudDemoMVC' (2 files). The 'WebRole1' node is expanded, showing its properties, references, App_Data, App_Start, Content, Controllers (with AccountController.cs and HomeController.cs selected), Views (with Account, Home, Manage, Shared, and their respective views), Scripts, Models, and fonts.

Now go in “Views” folder and add another view.



Here, I gave “Edit” name to my view. I’m adding a “Partial View”.

Click on “Add” button to add a partial view.

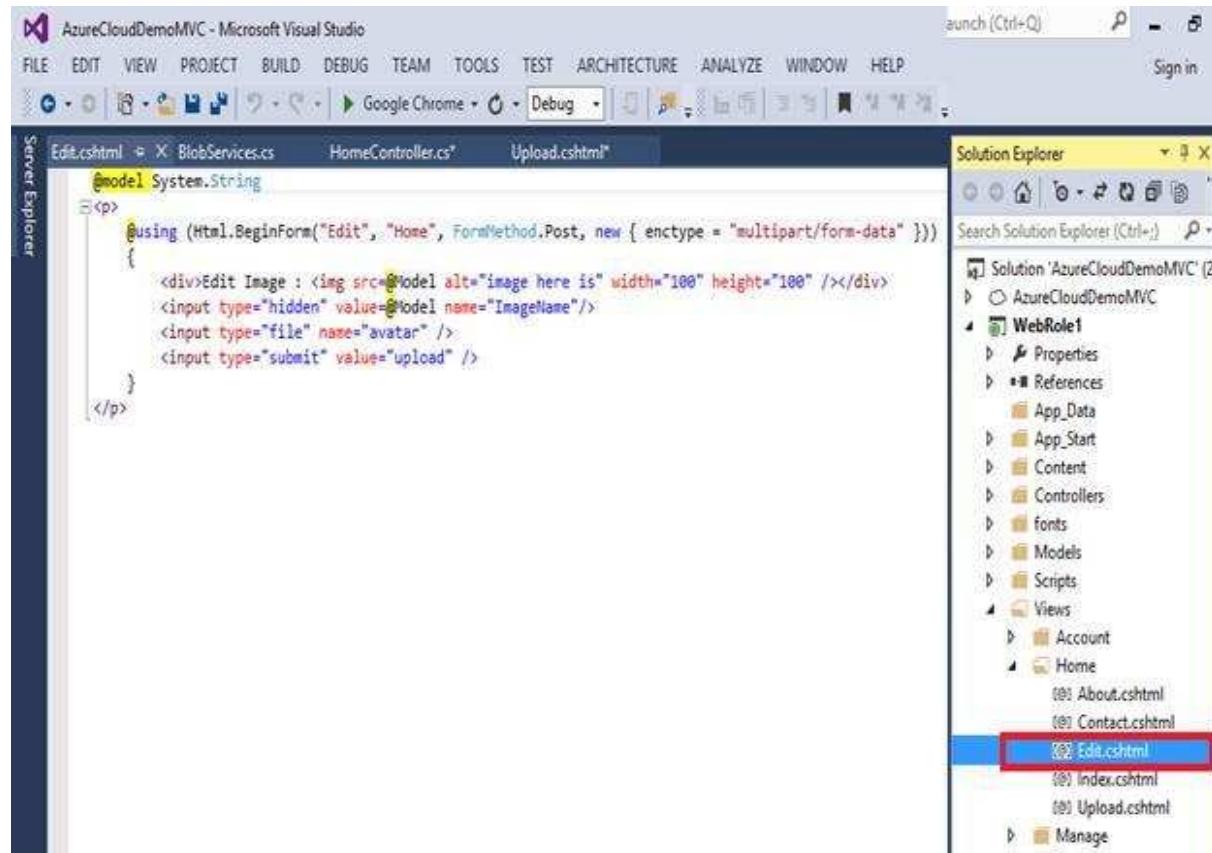


And now just write that minimum code in which I have to use the previous image and 1 fileupload control to upload new image and a submit button and a hidden control to store the previous image id/URI to change that with new image.

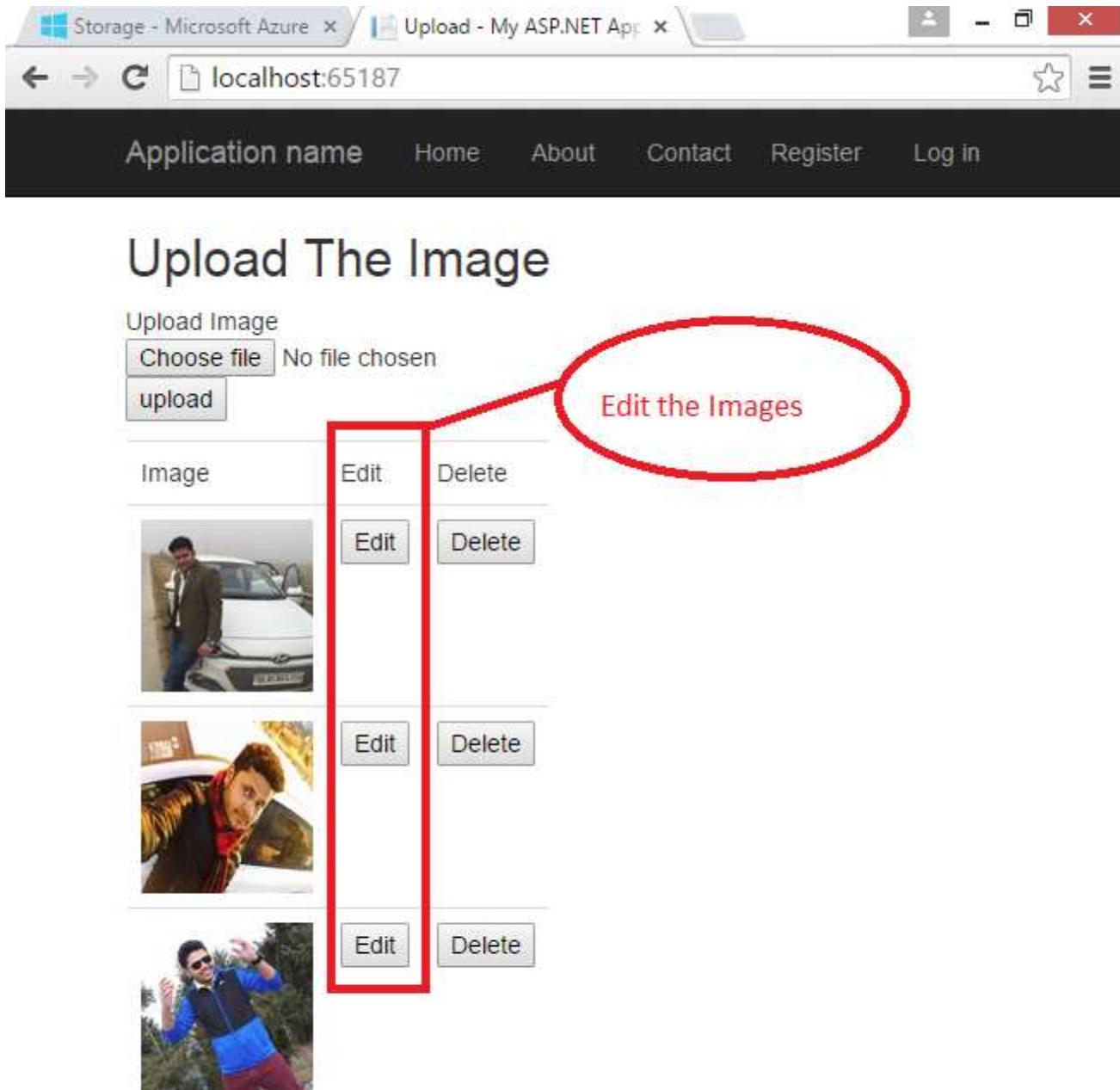
Code

```
@model System.String < p >
@using(Html.BeginForm("Edit", "Home", FormMethod.Post, new
{
    enctype = "multipart/form-data"
}) { < div > Edit Image: < img src = @Model alt = "image here is"
    width = "100"
    height = "100" / > < /div> < input type = "hidden"
    value = @Model name = "ImageName" / >
    < input type = "file"
    name = "avatar" / >
    < input type = "submit"
    value = "upload" / >
} < /p>
```

Can also show as in the following image:



Now, finally we can edit our image; click on your favorite image which you want to change.



Storage - Microsoft Azure Upload - My ASP.NET App

localhost:65187

Application name Home About Contact Register Log in

Upload The Image

Upload Image

Choose file No file chosen

upload

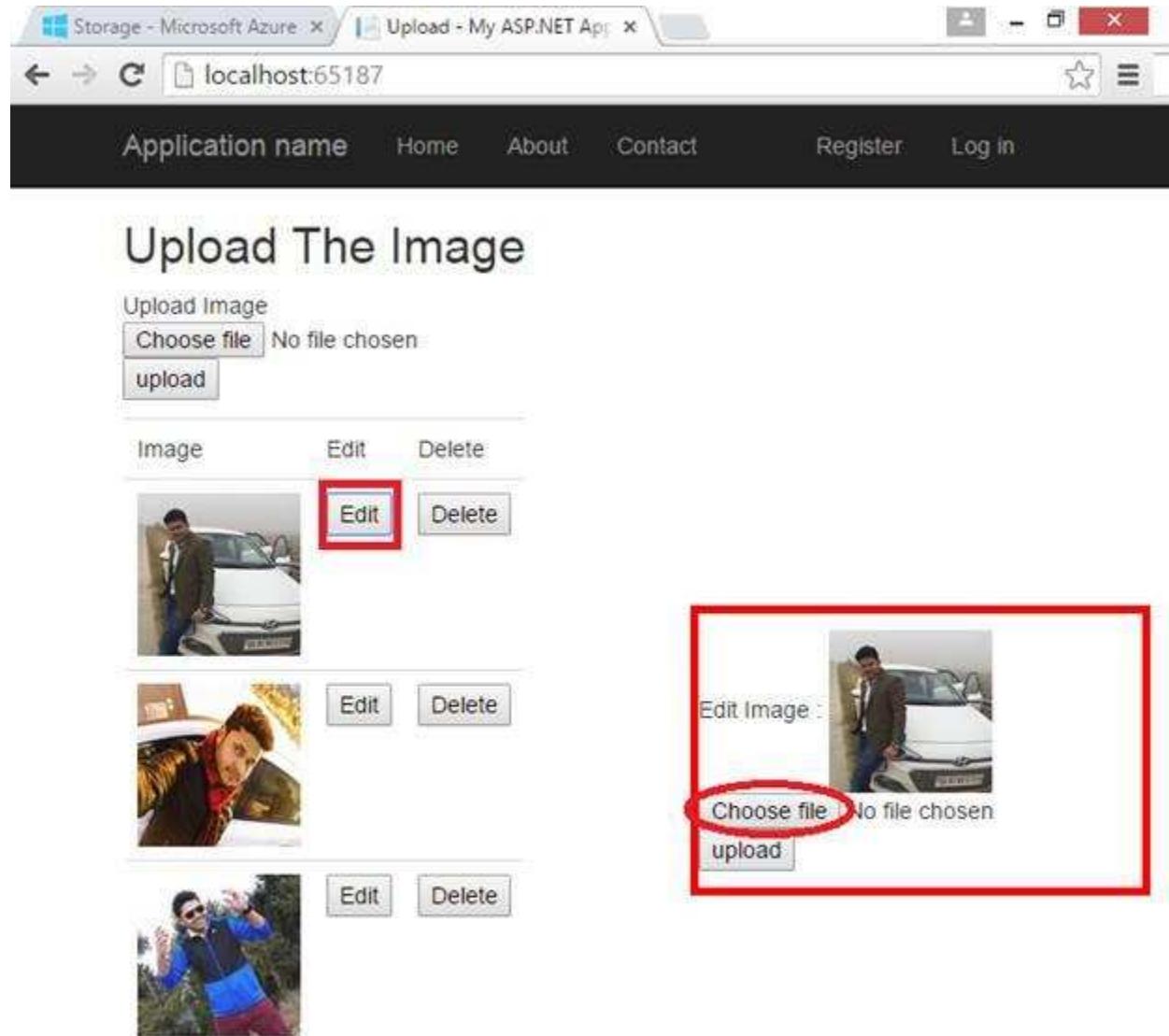
Image	Edit	Delete
	Edit	Delete
	Edit	Delete
	Edit	Delete

Here, I'm going to edit first image.

Click on “Edit” button.

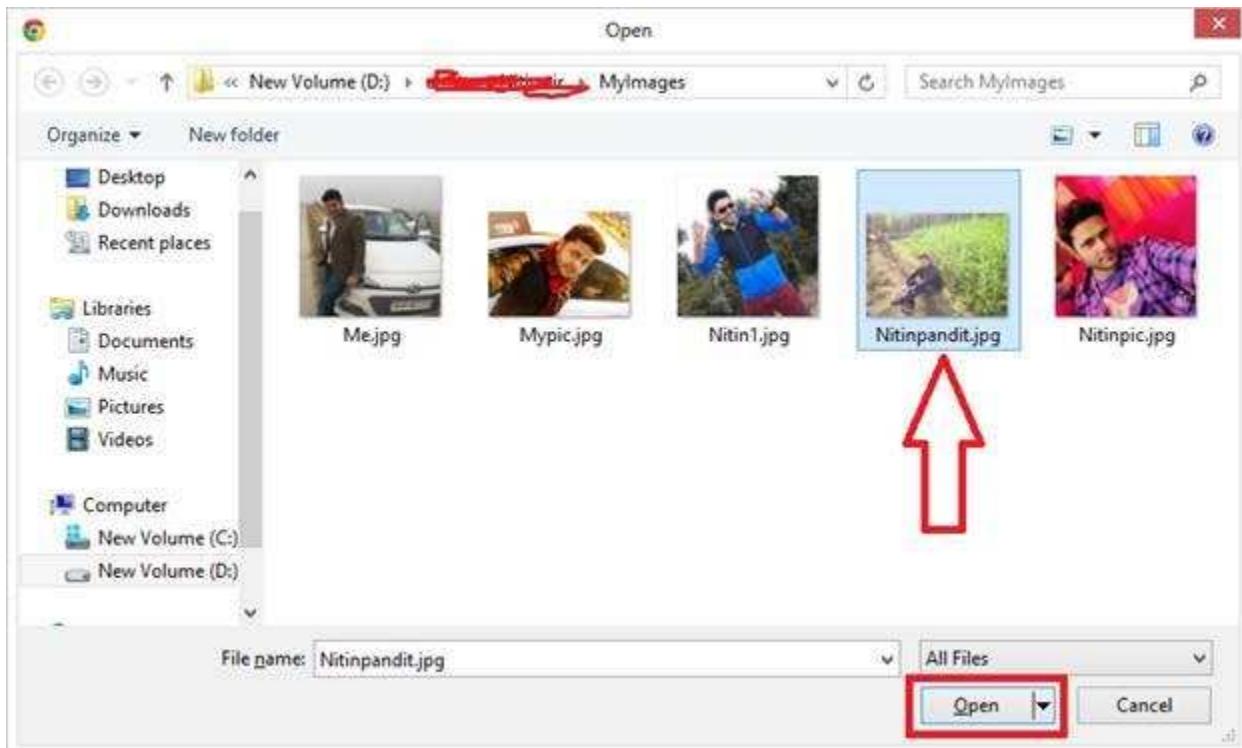
After Clicking on “Edit” button, you will see that image will open near “Delete” button.

Click on “Choose file” button.



Select your favorite image what you want to replace from old one.

Click on “**Open**” button to open that image.



After clicking on “**Open**” button, you will see the selected image name (**Nitinpandit.jpg**).

Click “**Upload**” button.

Storage - Microsoft Azure · Upload - My ASP.NET App · localhost:65187

Application name Home About Contact Register Log in

Upload The Image

Upload Image

Choose file No file chosen

upload

Image	Edit	Delete
	Edit	Delete
	Edit	Delete
	Edit	Delete

Edit Image :

Choose file Nitinpandit.jpg

upload

New image name selected

© 2016 - My ASP.NET Application

Finally, I got my output. That image successfully edits and updated, as well as in my cloud, Azure Explorer, and in Server Explorer (in Visual Studio).

Storage - Microsoft Azure Upload - My ASP.NET App

localhost:65187

Application name Home About Contact Register Log in

Upload The Image

Upload Image

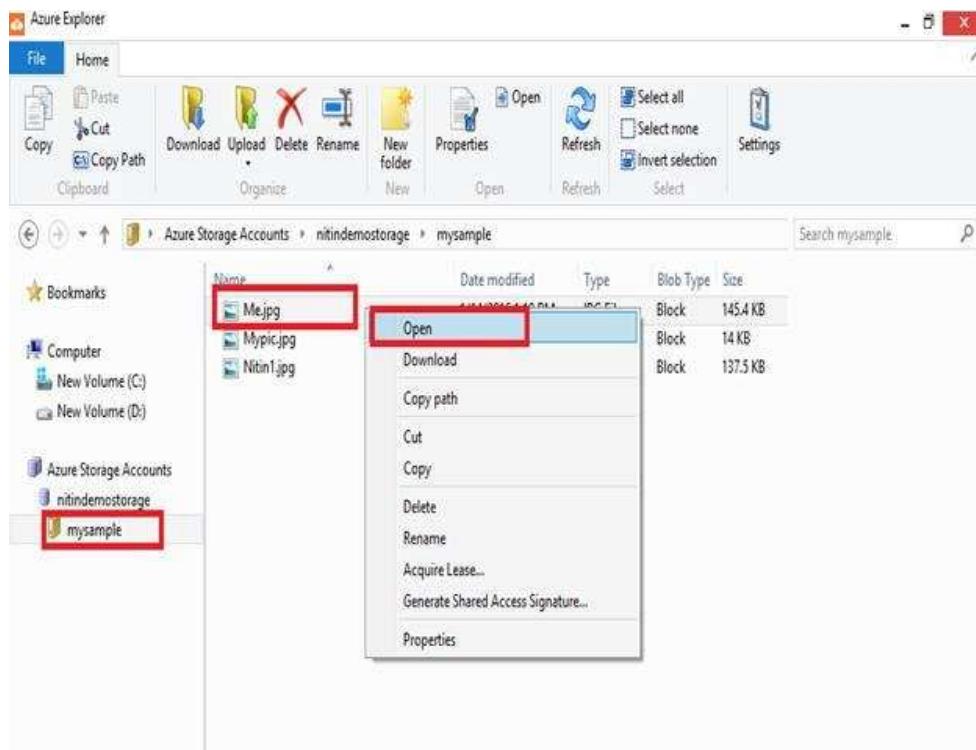
Choose file No file chosen

upload

Image	Edit	Delete
	Edit	Delete
	Edit	Delete
	Edit	Delete

© 2016 - My ASP.NET Application

To check our changes, just go in to Azure Explorer and right click on the top image and select “Open”.



Below is my editable image and final output.



Getting Started With Azure DocumentDB

What is DocumentDB?



DocumentDB is a fully managed, highly scalable, query-able, truly schema-free JSON document database, delivered as a service for modern applications.

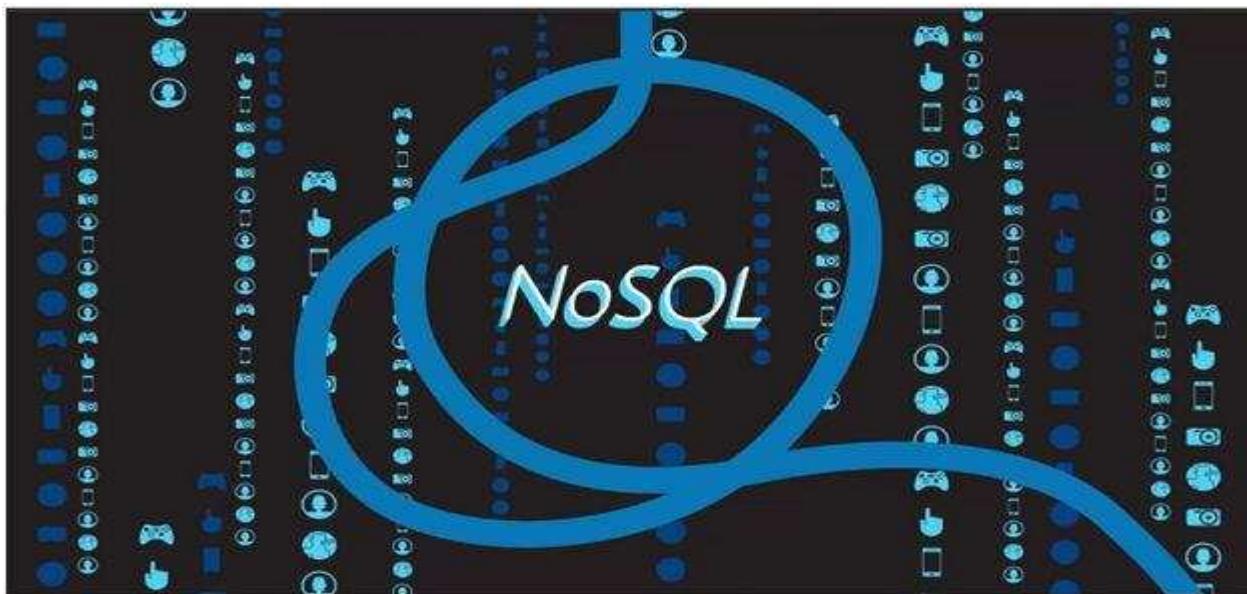
Azure DocumentDB is a NoSQL document database service designed from the ground up to natively support JSON and JavaScript directly inside the database engine.

Why

- Produce and consume data at a staggering rate while requiring near instantaneous response times to match user expectations.
- Increasingly complex, compensating for the lack of query and transactional processing in today's NOSQL database system.
- Experience unpredictable and explosive growth.

What is a database account?

A database account holds a set of databases, each database can contain multiple collections so every collection can contain stored procedure, triggers and related documents, JSON files.

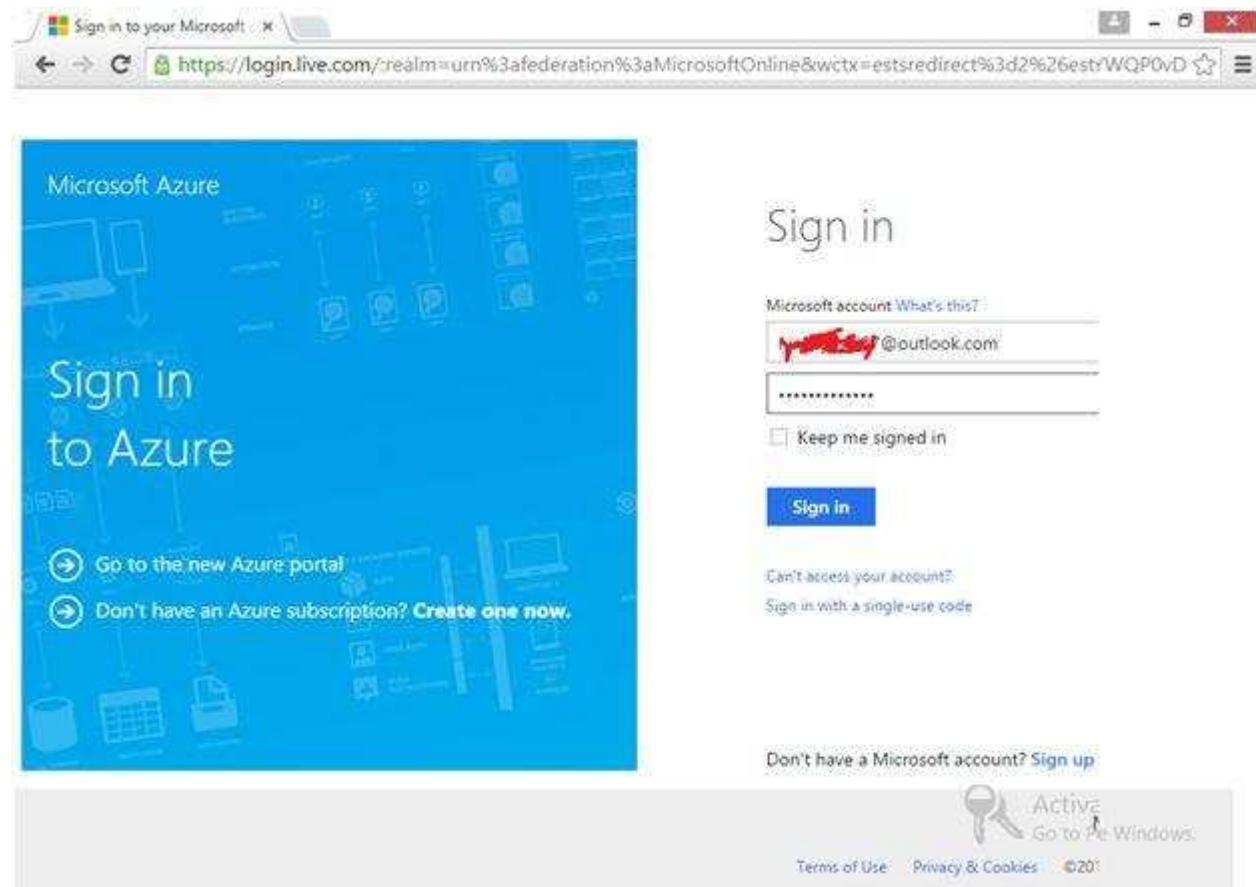


Azure DocumentDB is a NoSQL document database service, support JSON and JavaScript directly inside the database engine.

DocumentDB is designed for modern mobile and web applications. Azure DocumentDB supports querying using a SQL language.



To use Azure DocumentDB, you must have a DocumentDB account, a database, a collection, and documents. So for this, simply go to Microsoft Azure login page and login with your account.



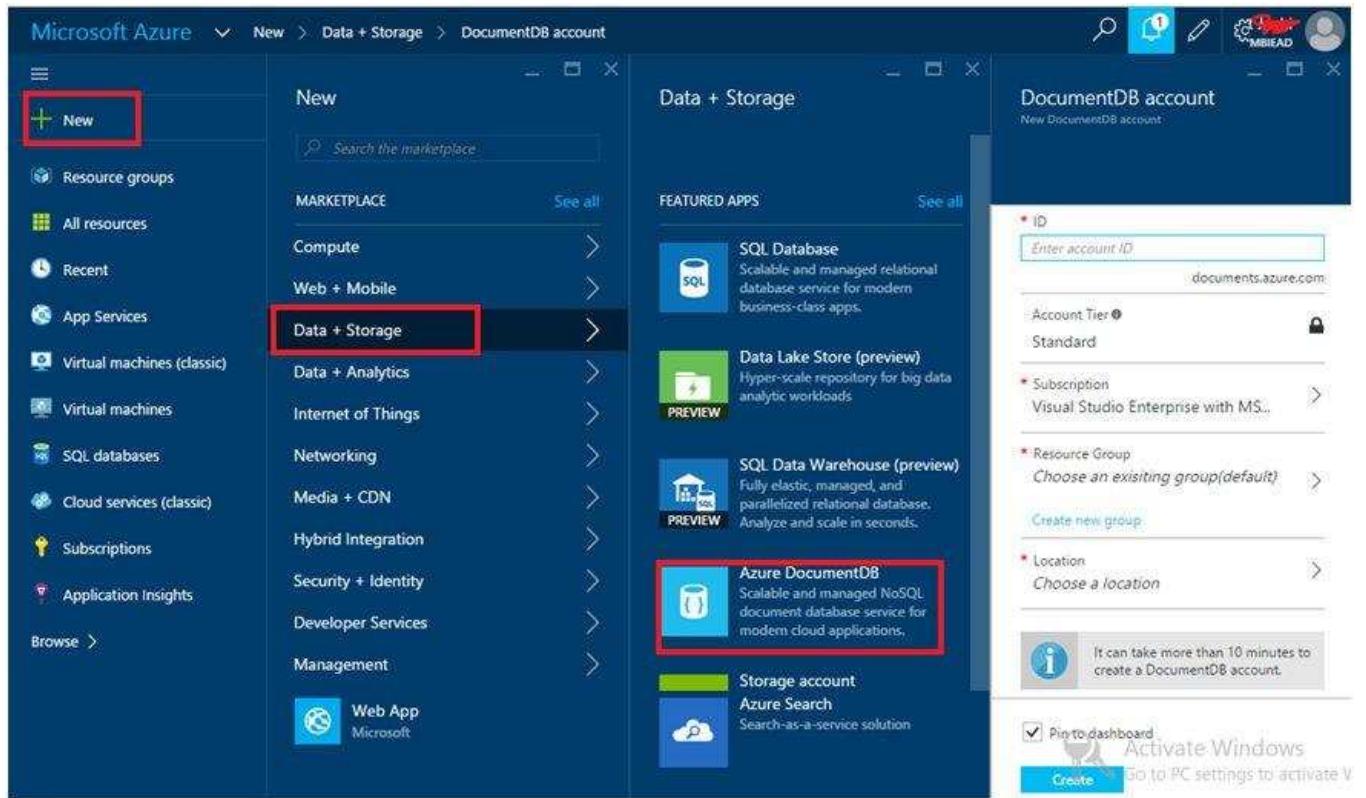
After the “**Sign in**”, go to “**Home**” and click on to “**Portal**”. After clicking on the portal you will see the following window. For DocumentDB you have to do:

Step 1: Click on “**New**” button.

Step 2: Scroll down and select “**Data+Storage**”.

Step 3: Again scroll down and you will see “**Azure DocumentDB**”, Click on it.

After clicking on the “**Azure DocumentDB**” you will automatic jump on the DocumentDB account. This will ask to you for the “**ID**”.



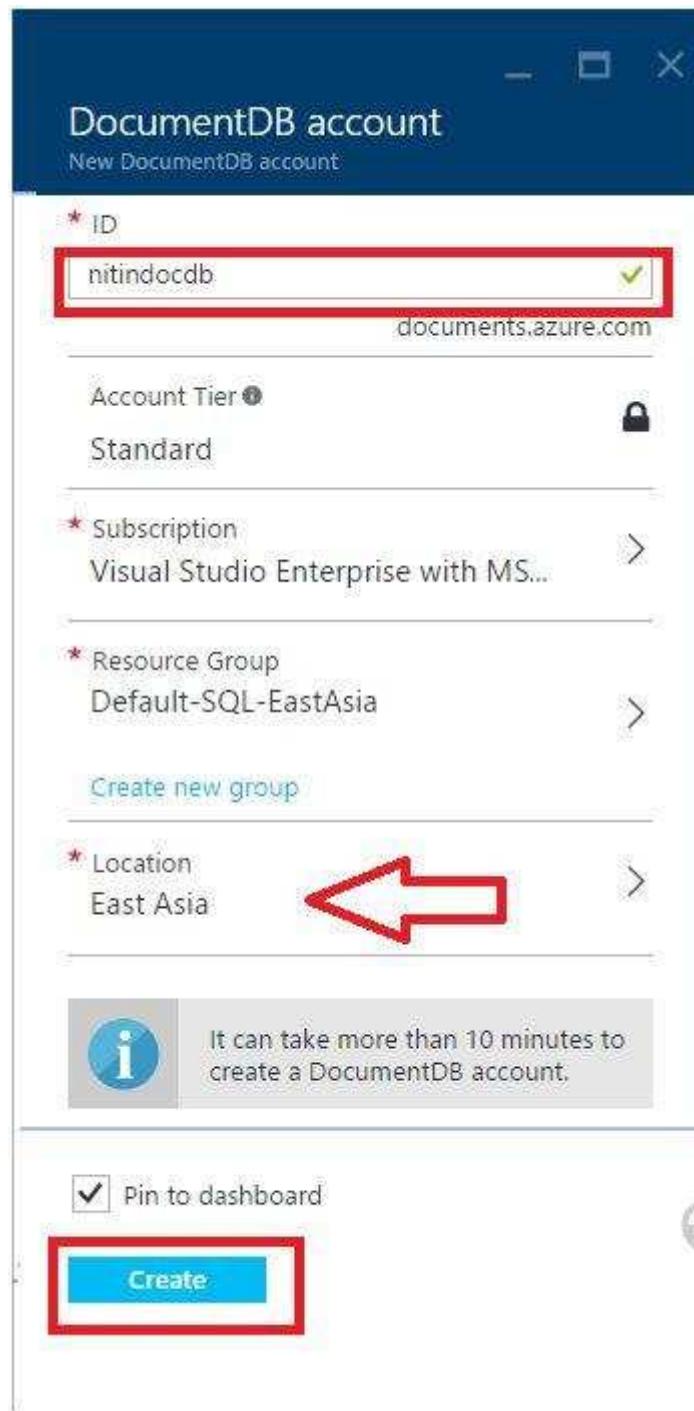
Give the “**ID**” name here, Id name is simply your “DocumentDB account” name what you want to use for further operations. Here I gave it “**nitindocdb**”, you can give any name here what you want. Make sure ID name cannot contain / \ #? or space between the words.

When your name is validated, this will mark a green check.

In the “**Resource Group**” you can choose the “**Default- SQL-EastAsia**”, this is nothing but your location where you are going to use this account.

In the “**Location**” section you can choose your location.

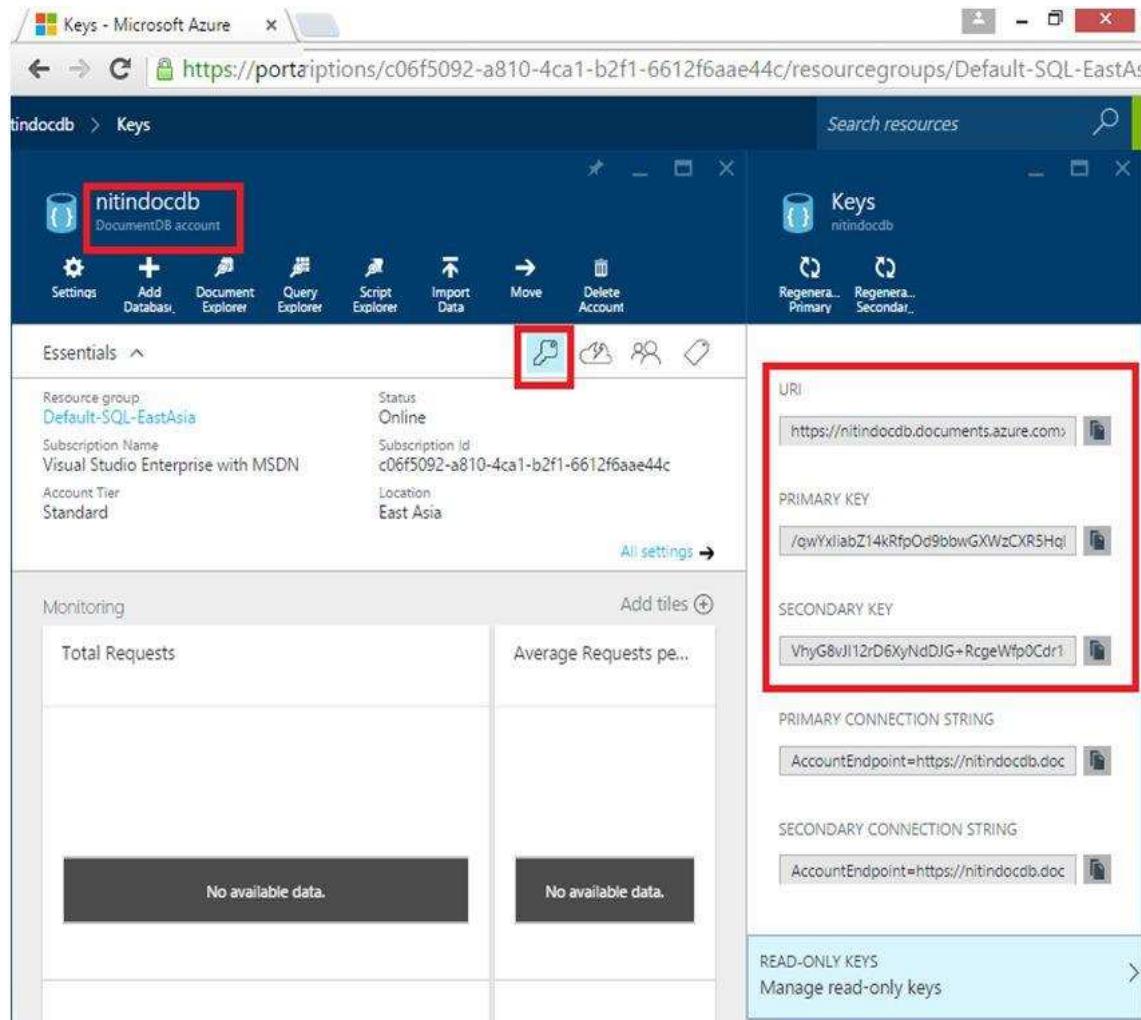
After this, hit on the “**Create**” button.



Here in the following figure, I have successfully created my DocumentDB account with name "nitindocdb".

Here you can see different types of operations like settings, Add Database, Document Explorer, Query Explorer, Script Explorer, Import Data, Move and Delete Account.

Click on key button (shown in middle red rectangle), you can see in the right side of the window it holds “URI” and “Keys”. URI and Keys are the particular information of the DocumentDB account.

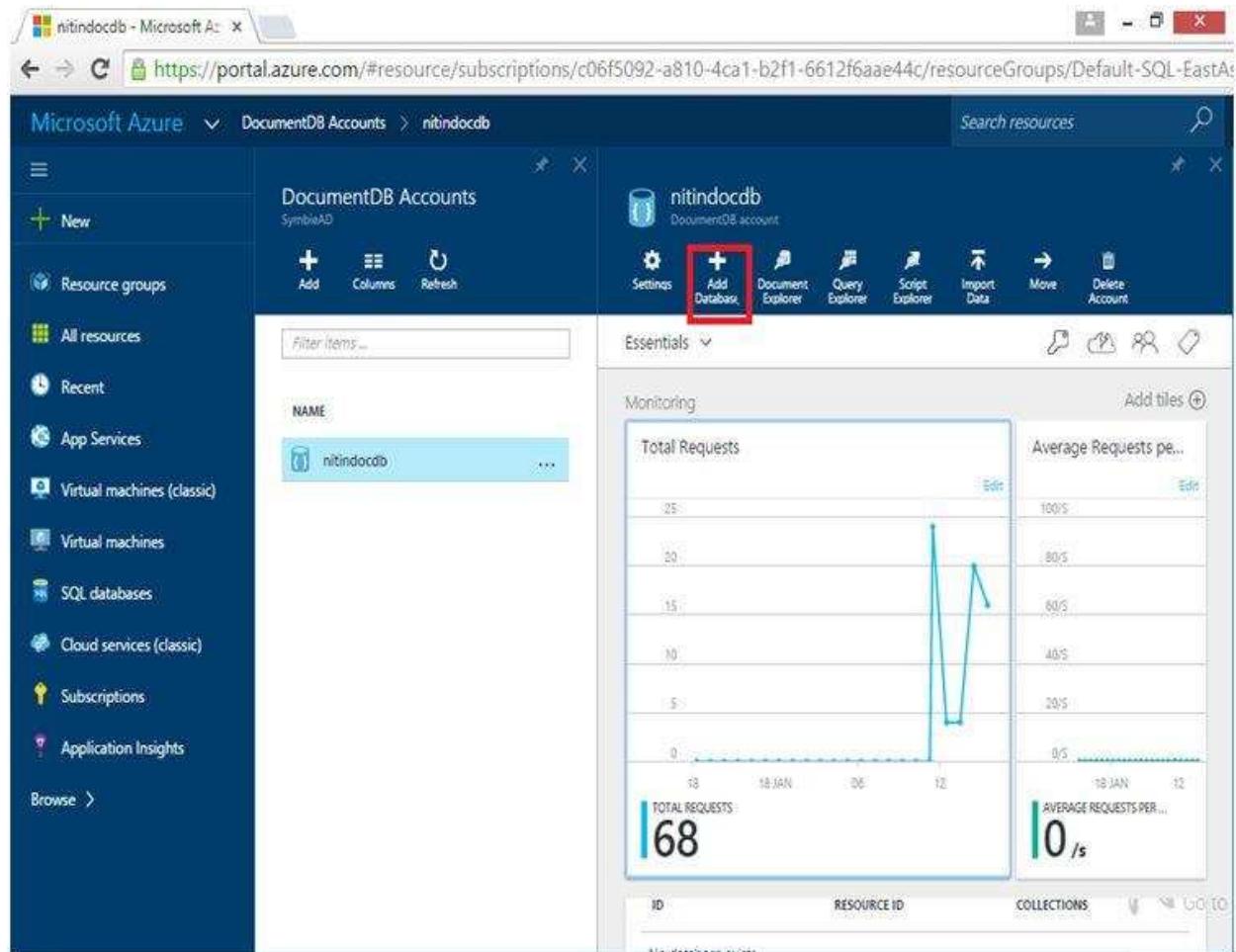


The screenshot shows the Microsoft Azure portal interface for managing a DocumentDB account named "nitindocdb". The top navigation bar includes "Keys - Microsoft Azure" and a URL "https://portraits/c06f5092-a810-4ca1-b2f1-6612f6aae44c/resourcegroups/Default-SQL-EastAsia/providers/Microsoft.DocumentDB/accounts/nitindocdb/keys". The main content area displays the account details under "Essentials" and monitoring metrics under "Monitoring". On the right side, the "Keys" section is expanded, showing the "URI" (highlighted with a red box) and "PRIMARY KEY" fields. Below these, there are sections for "SECONDARY KEY", "PRIMARY CONNECTION STRING", and "SECONDARY CONNECTION STRING". At the bottom, there is a link for "READ-ONLY KEYS" with the sub-instruction "Manage read-only keys". A red box also highlights the "URI" field in the "Keys" section.

Now, I'm going to create a database.

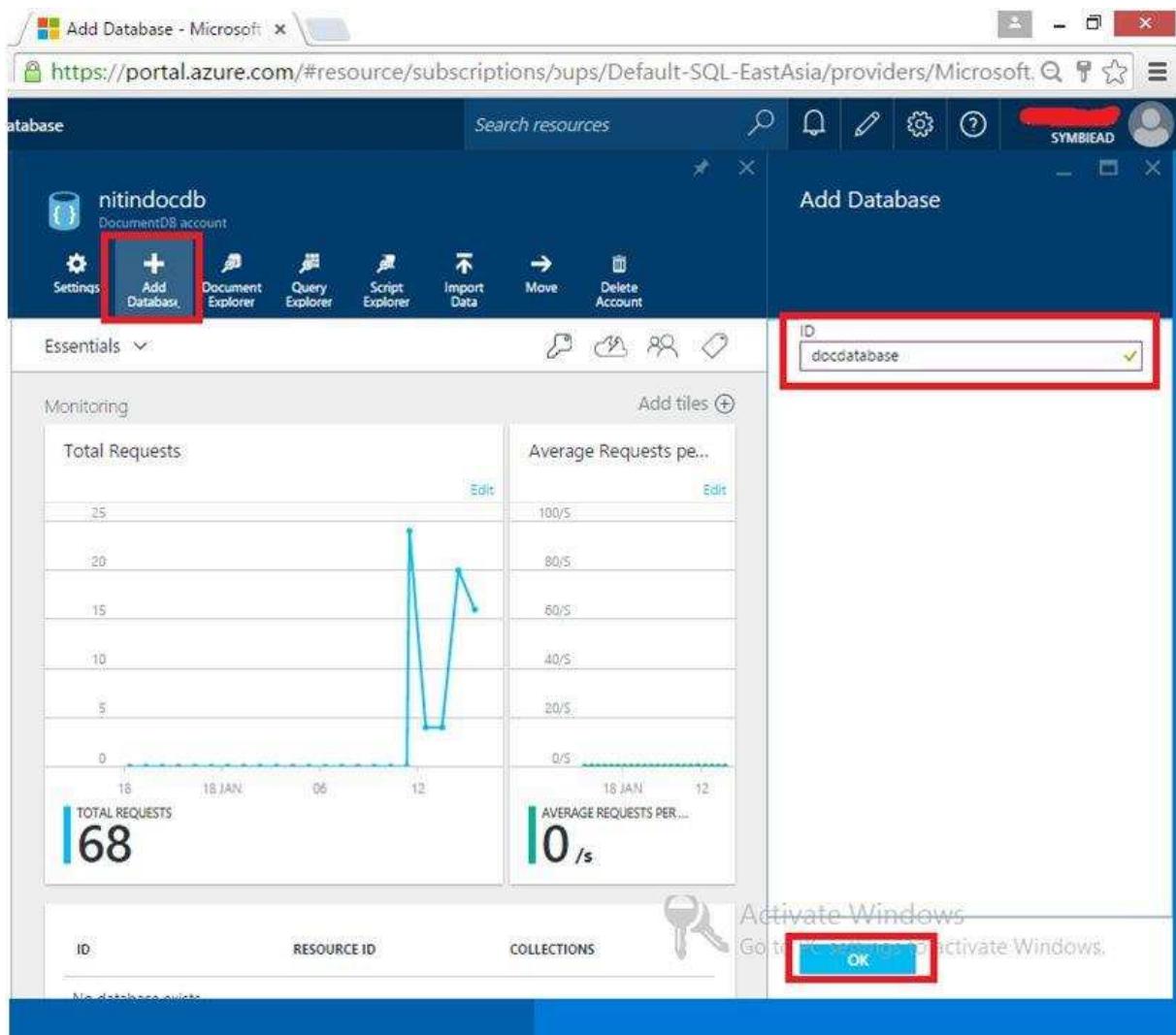
After clicking on my account name, which I created “nitindocdb”, we will redirect on the next database window.

Click on “Add Database”.

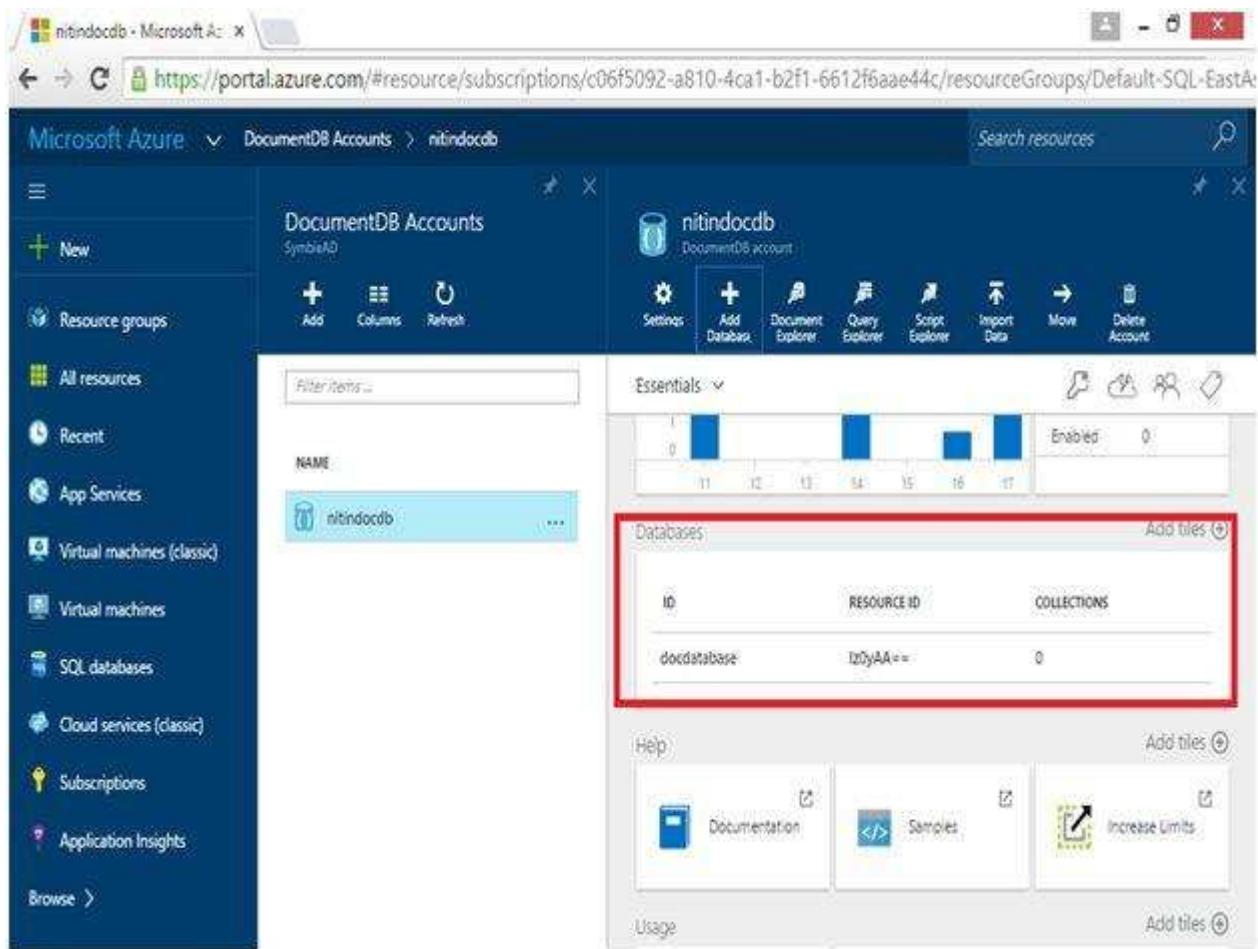


After clicking on the “**Add Database**” this will redirect in to new window, and again ask for “**ID**”, here “**ID**” is your database name, so in the “**ID**” section give the database name which is you want to use, here I gave “**docdatabase**” name to my database.

Hit on “**OK**” button.



Now you can see I have created a database with name “**docdatabase**”.



The screenshot shows the Microsoft Azure portal interface for a DocumentDB account named 'nitindocdb'. On the left, the navigation menu includes options like 'New', 'Resource groups', 'All resources', 'Recent', 'App Services', 'Virtual machines (classic)', 'Virtual machines', 'SQL databases', 'Cloud services (classic)', 'Subscriptions', and 'Application Insights'. The main content area displays the 'DocumentDB Accounts' blade for 'nitindocdb'. It features a summary card with metrics such as '0' for documents, '0' for collections, and '0' for enabled databases. Below this is a table titled 'Databases' with one entry:

ID	RESOURCE ID	COLLECTIONS
docdatabase	DDyAA==	0

A red box highlights the 'Databases' table. At the bottom of the blade, there are links for 'Documentation', 'Samples', and 'Increase Limits'.

Now we need to add Collection.

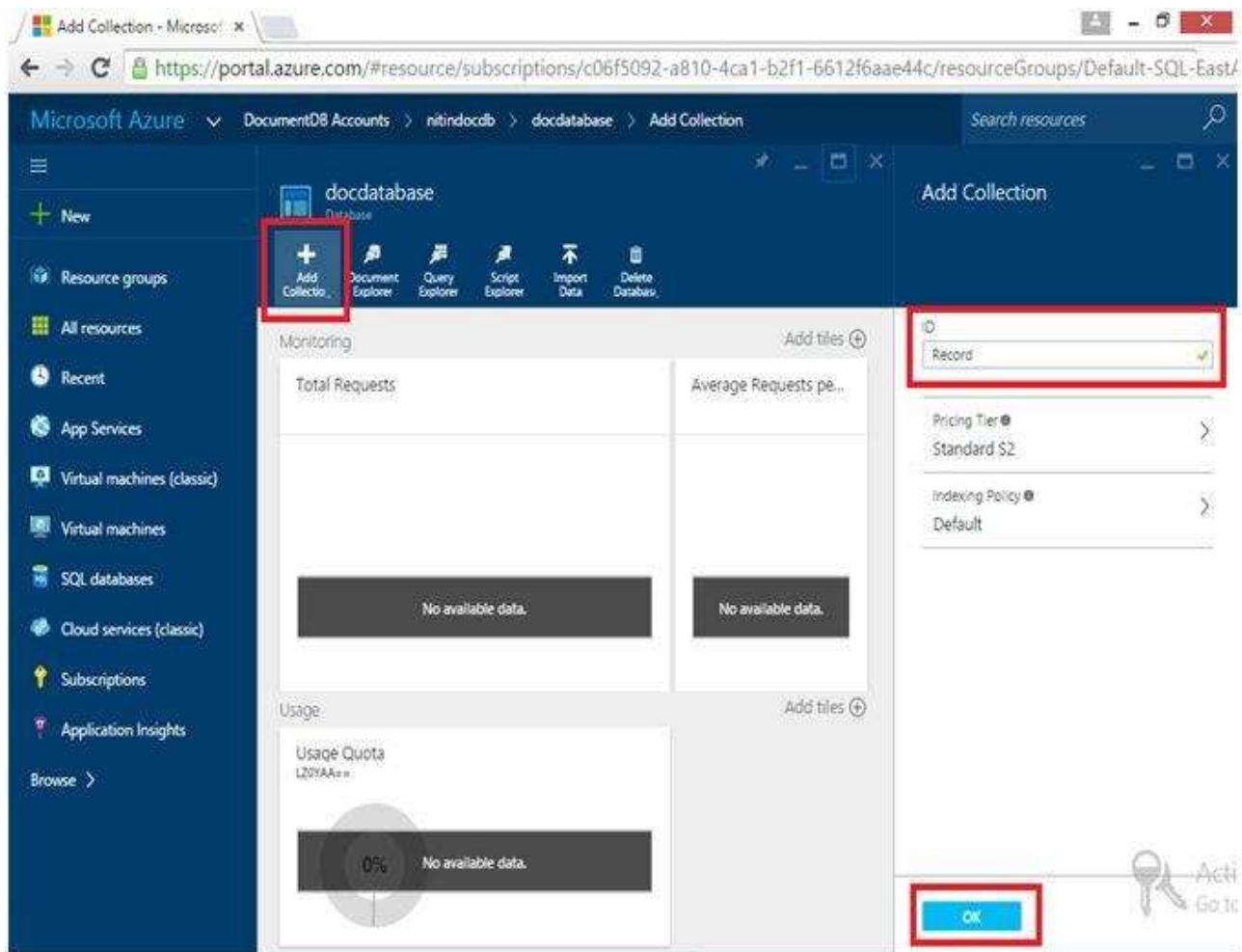
Collections are the transaction boundary for stored procedures and triggers, and the entry point to queries and CRUD operations.

Click on the “**docdatabase**” which I have created just and it will redirect you to new window, here you can see the “**Add Collection**” option.

Hit on “**Add Collection**”.

In the next window it will again ask for the “**ID**”, here “**ID**” is your collection ID, give the name in “**ID**” section, here I gave “Record” name to my collection.

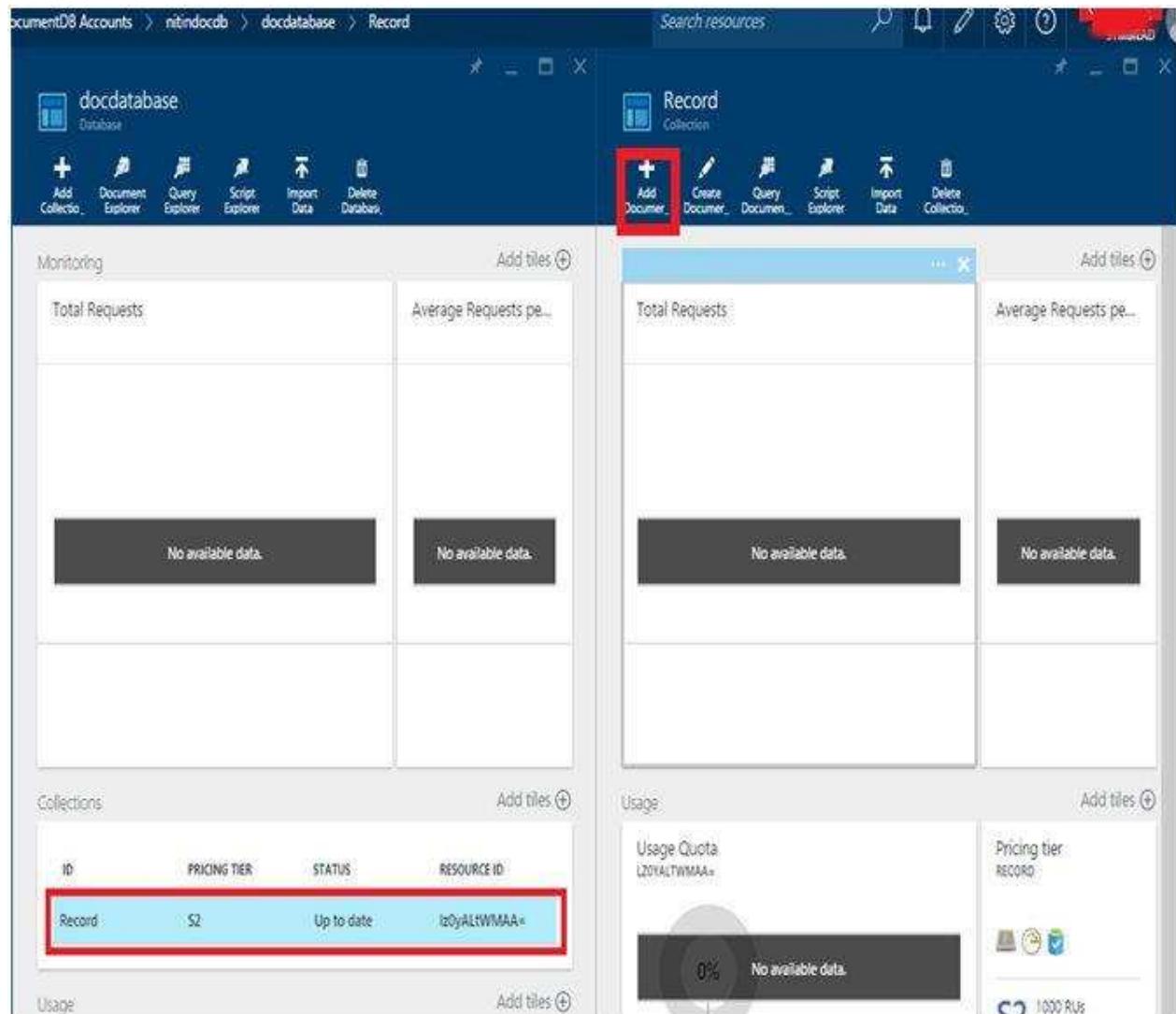
Click on “**OK**” button.



I have created a collection, click on it.

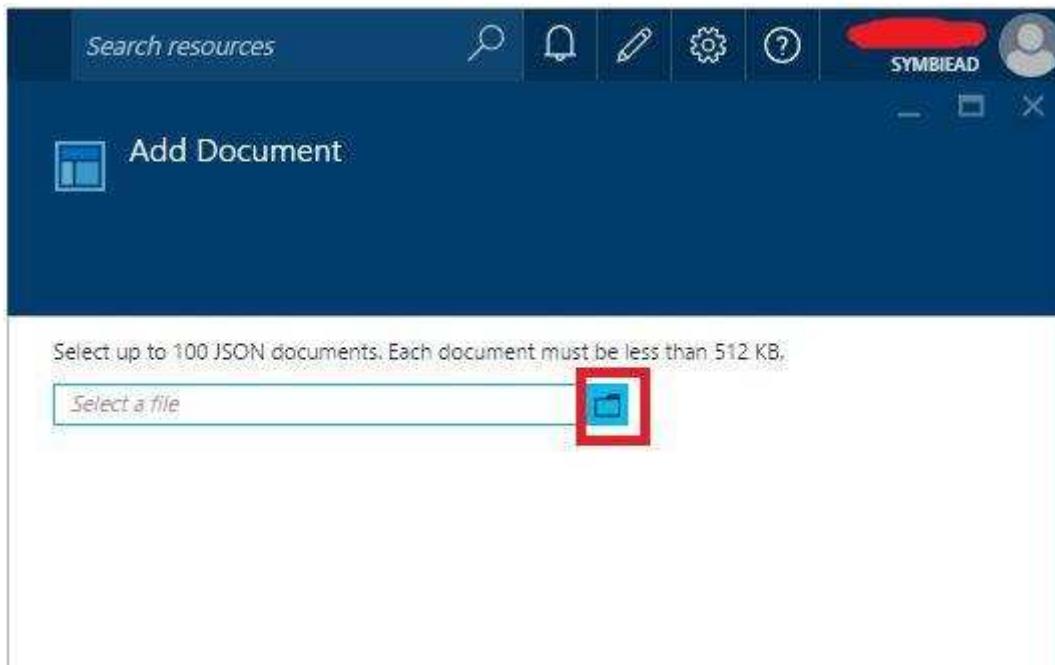
Click on the “**Record**” collection and you can find “**Add Documents**” in the next window.

In the next window you can add the records and documents in the collection section. We can add any record or documents in the collection section.

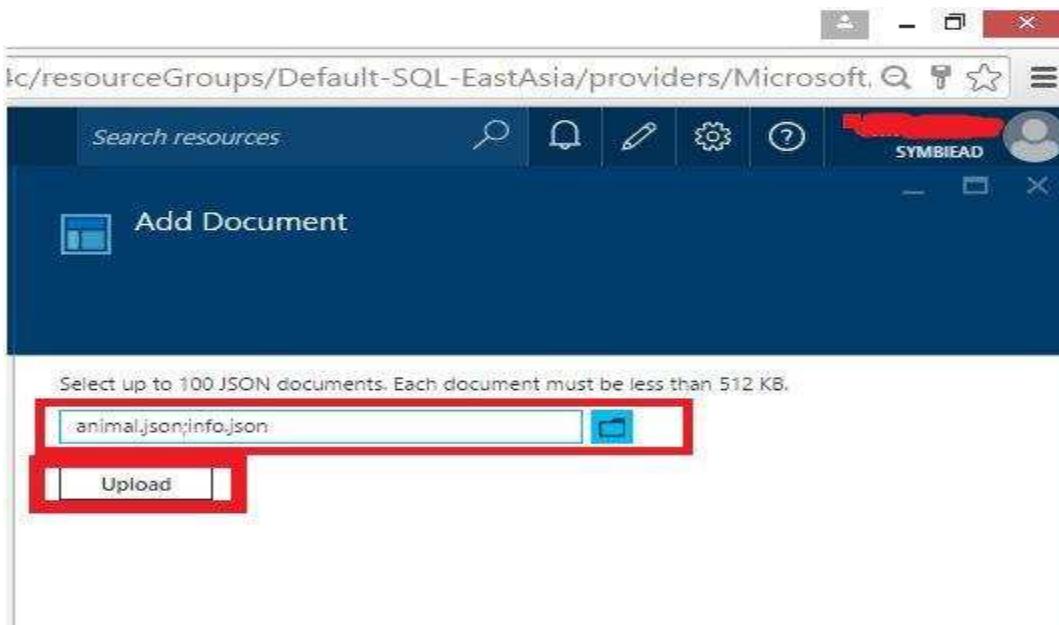


The screenshot shows the Azure DocumentDB portal interface. At the top, there's a navigation bar with 'documentDB Accounts > ntinodocdb > docdatabase > Record'. Below the navigation is a toolbar with icons for 'Add Collection', 'Document Explorer', 'Query Explorer', 'Script Explorer', 'Import Data', and 'Delete Database'. The main area has two sections: 'Monitoring' and 'Collections'. The 'Monitoring' section contains tiles for 'Total Requests' and 'Average Requests per second', both of which show 'No available data.'. The 'Collections' section shows a table with columns 'ID', 'PRICING TIER', 'STATUS', and 'RESOURCE ID'. A single row is highlighted with a red rectangle, showing 'Record' in the ID column, 'S2' in the Pricing Tier column, 'Up to date' in the Status column, and 'lz0yALTWMAA=' in the Resource ID column. To the right of the table is a 'Usage' section with a '0%' progress bar and a note 'No available data.'.

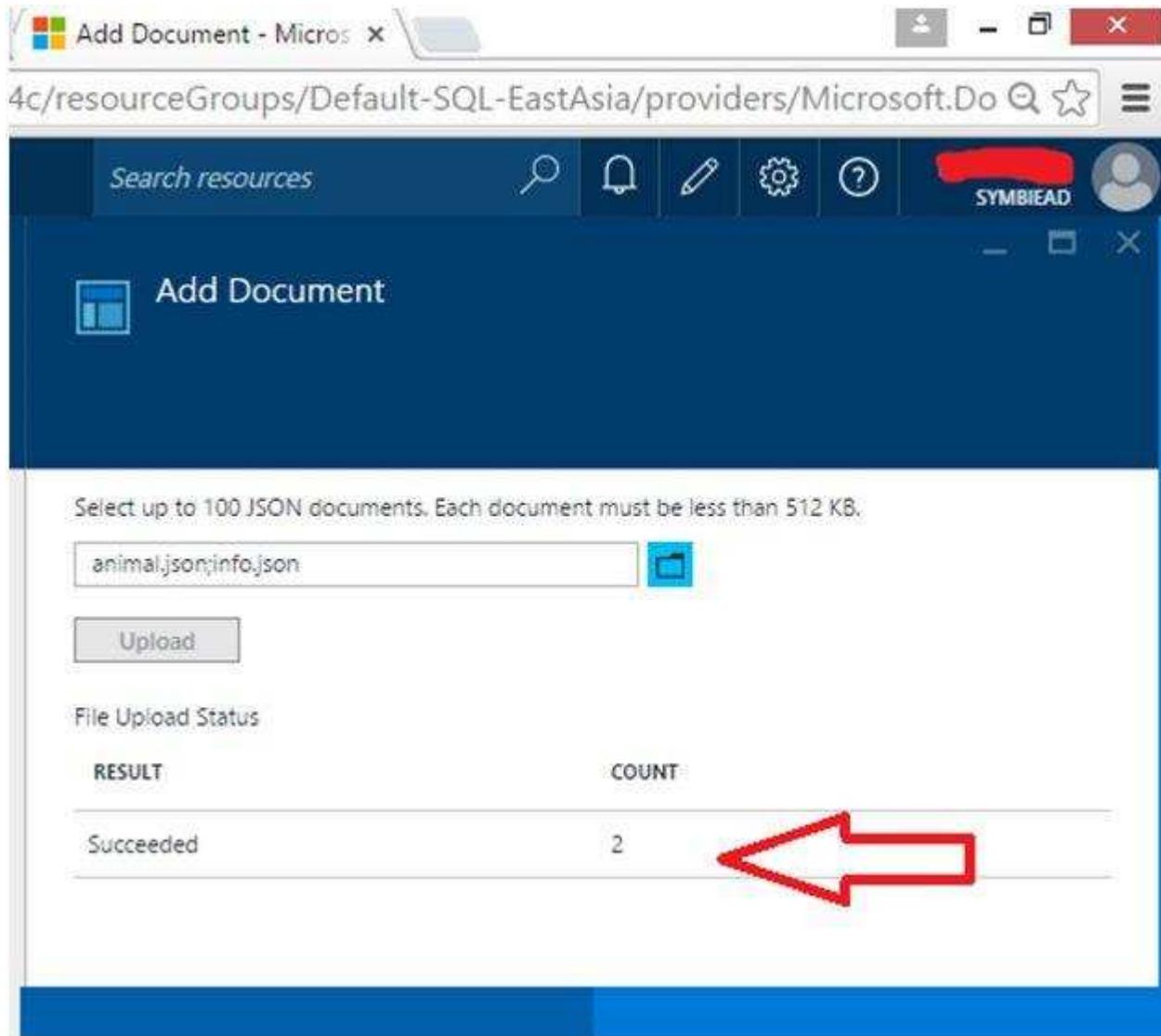
It will redirect to next page, here you can upload 100 JSON documents, but make sure document must be less than 512kb. To upload JSON documents click on the red rectangle.



Here I have selected two JSON files, animal.json and info.json with extension of json. Click on “**Upload**” button.



So, here I had successfully uploaded two files, you can see the result is succeeded.



The screenshot shows the Microsoft Azure portal interface for uploading documents. The title bar reads "Add Document - Micros". The URL in the address bar is "4c/resourceGroups/Default-SQL-EastAsia/providers/Microsoft.Do". The main content area is titled "Add Document" and contains a text input field with "animal.json;info.json" and a blue "Upload" button. Below this, a "File Upload Status" table shows one row: "RESULT" is "Succeeded" and "COUNT" is "2". A large red arrow points to the "COUNT" value of 2.

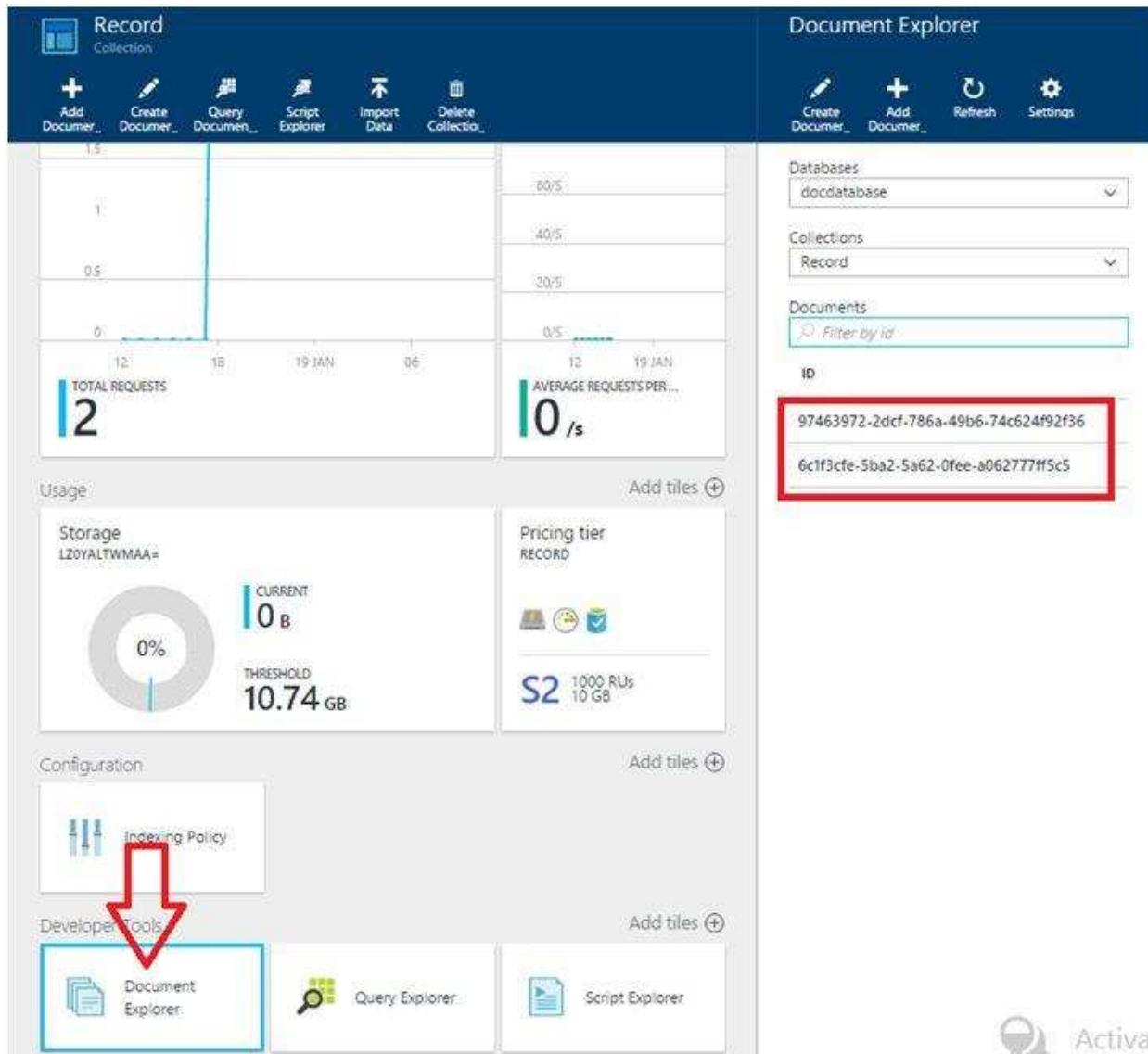
RESULT	COUNT
Succeeded	2

Now, go to the “**Document Explorer**” to check your uploaded files. Click on “**Document Explorer**” it will redirect you to the document explorer window.

You can see, there are two json files ID, it creates the file in the format of “ID” by default.

View, create, and edit documents with Document Explorer-

Document Explorer allows you to easily create, edit and delete documents. I'm going to edit my document, click on any one to edit.



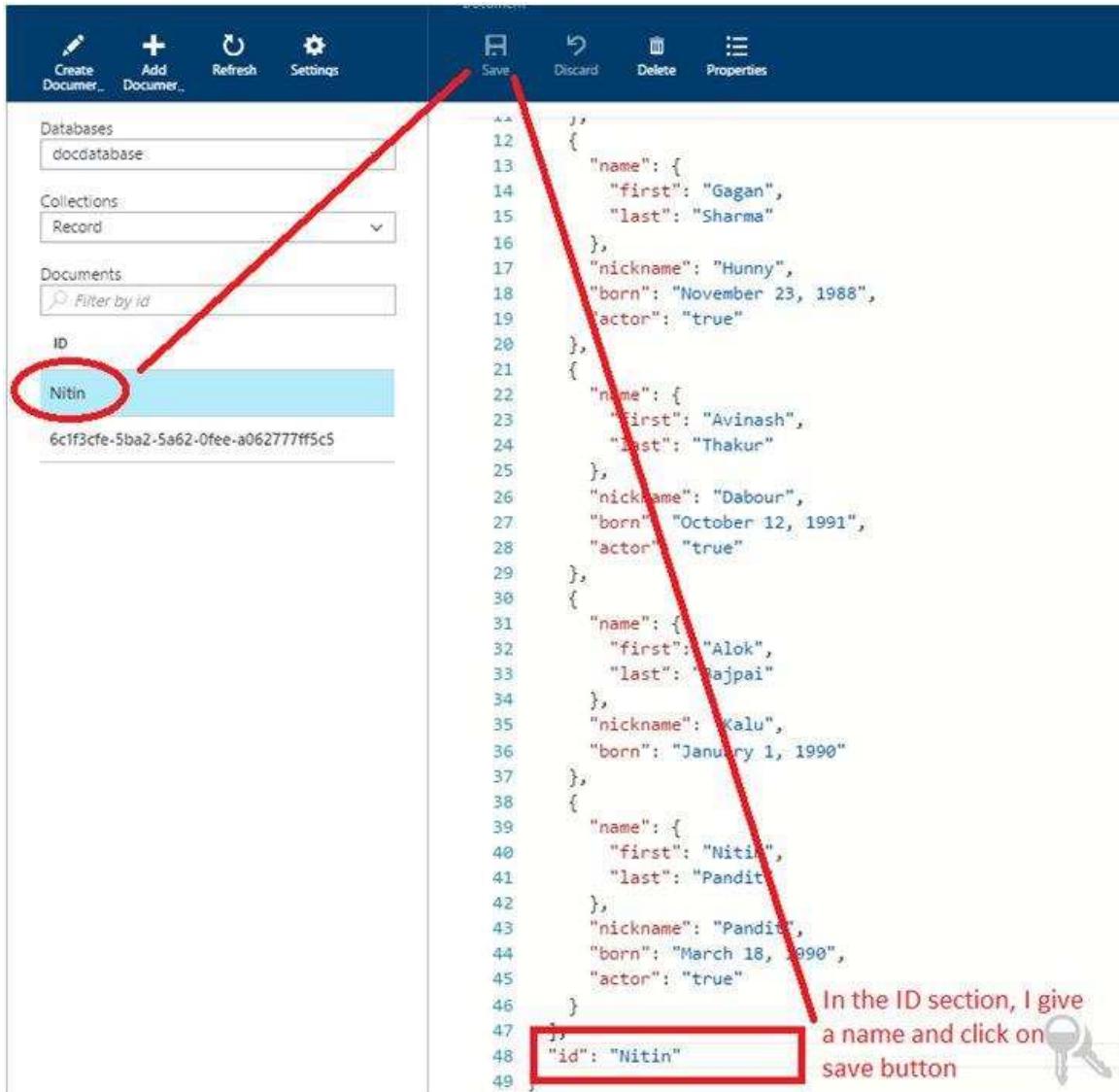
The screenshot shows the Azure portal interface with the following details:

- Record Collection:** A dashboard with various metrics and tiles. One tile shows "TOTAL REQUESTS" with a value of 2, and another shows "AVERAGE REQUESTS PER..." with a value of 0/s.
- Storage:** Shows current storage usage at 0B and a threshold of 10.74 GB.
- Pricing tier:** Set to RECORD, which includes 1000 RUs and 10 GB.
- Configuration:** Includes an "Indexing Policy" tile and a "Developer Tools" section containing "Document Explorer", "Query Explorer", and "Script Explorer". A red arrow points from the "Developer Tools" section to the "Document Explorer" tile.
- Document Explorer:** A separate pane on the right showing a list of documents. Two specific document IDs are highlighted with a red box: 97463972-2dcf-786a-49b6-74c624f92f36 and 6c1f3cfe-5ba2-5a62-0fee-a062777ff5c5.

I'm going to edit top Id, click on it, you will see the code of the JSON, scroll down and in the “ID” section give the name to “ID” whatever you want. Here I gave “Nitin”, and then click on to “Save” button.

After clicking on “Save” button, you can see in the left side of the window, the Id name will be changed.

You can edit everything in the document section whatever you want, after edit the document, click on save button it will be changed.



The screenshot shows a MongoDB interface with a sidebar and a main document view. In the sidebar, under 'Documents', the ID 'Nitin' is highlighted with a red oval. In the main area, a JSON document is displayed with the 'id' field set to 'Nitin'. A red arrow points from the highlighted ID in the sidebar to the 'id' field in the JSON code. A callout bubble with a key icon contains the text: "In the ID section, I give a name and click on save button".

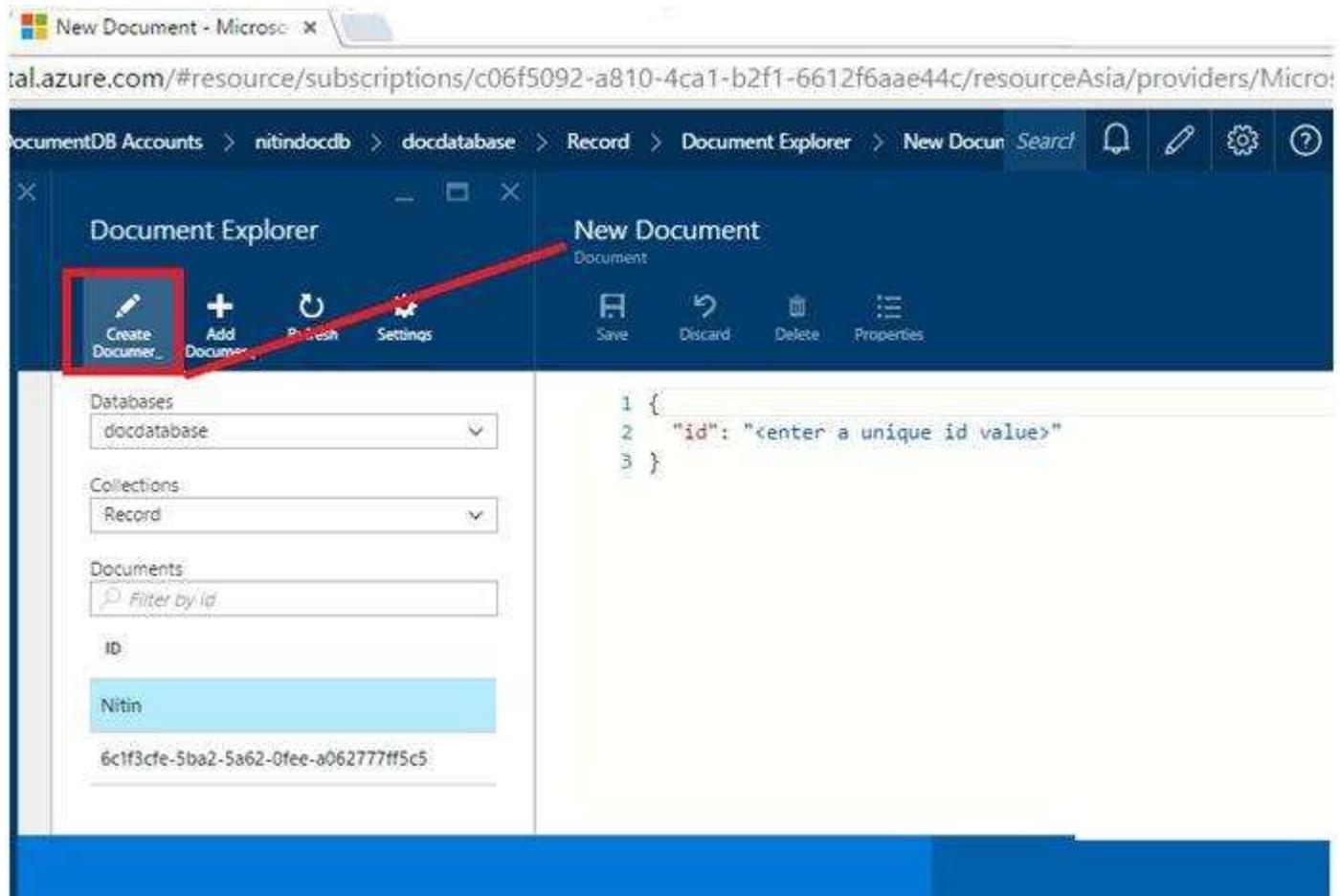
```

11
12
13   "name": {
14     "first": "Gagan",
15     "last": "Sharma"
16   },
17   "nickname": "Hunny",
18   "born": "November 23, 1988",
19   "actor": "true"
20 },
21 {
22   "name": {
23     "first": "Avinash",
24     "last": "Thakur"
25   },
26   "nickname": "Dabour",
27   "born": "October 12, 1991",
28   "actor": "true"
29 },
30 {
31   "name": {
32     "first": "Alok",
33     "last": "Vajpai"
34   },
35   "nickname": "Kalu",
36   "born": "January 1, 1990"
37 },
38 {
39   "name": {
40     "first": "Nitin",
41     "last": "Pandit"
42   },
43   "nickname": "Pandit",
44   "born": "March 18, 1990",
45   "actor": "true"
46 }
47 },
48 "id": "Nitin"
49

```

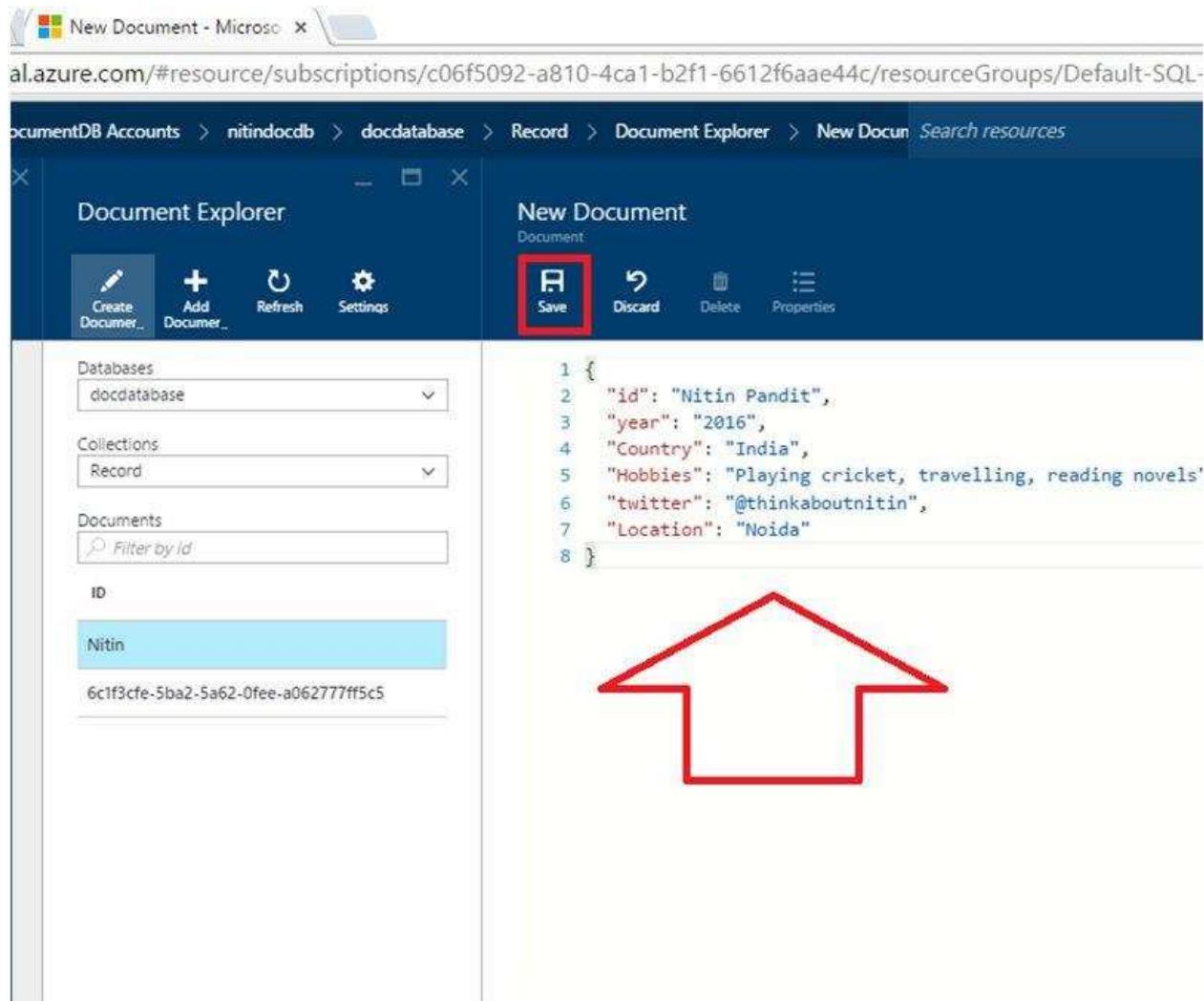
Now, I'm going to create a document.

To create a document, simply click the “**CreateDocument**” command and a minimal JSON snippet is provided



If you do not provide an "**id**" property, then Document Explorer automatically adds an id property and generates a GUID as the id value.

Simply type the JSON content of the document you wish to create and click on “**Save**” command, to commit your document.

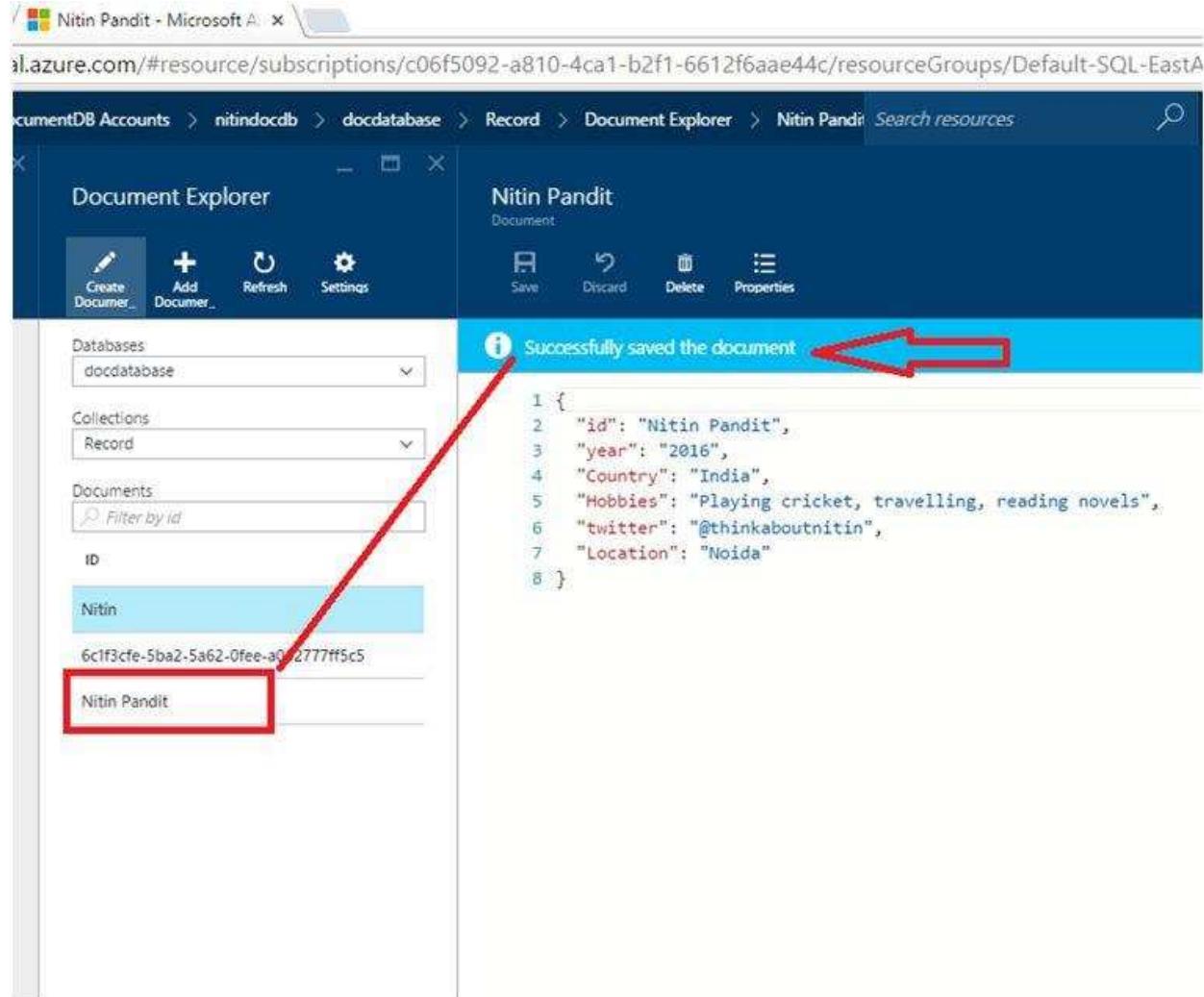


The screenshot shows the Microsoft Document Explorer interface. On the left, there's a sidebar with 'Document Explorer' buttons: 'Create Document...', 'Add Document...', 'Refresh', and 'Settings'. Below these are dropdown menus for 'Databases' (set to 'docdatabase'), 'Collections' (set to 'Record'), and 'Documents' (with a search bar 'Filter by id'). In the main area, a 'New Document' tab is active, showing a JSON document with the following code:

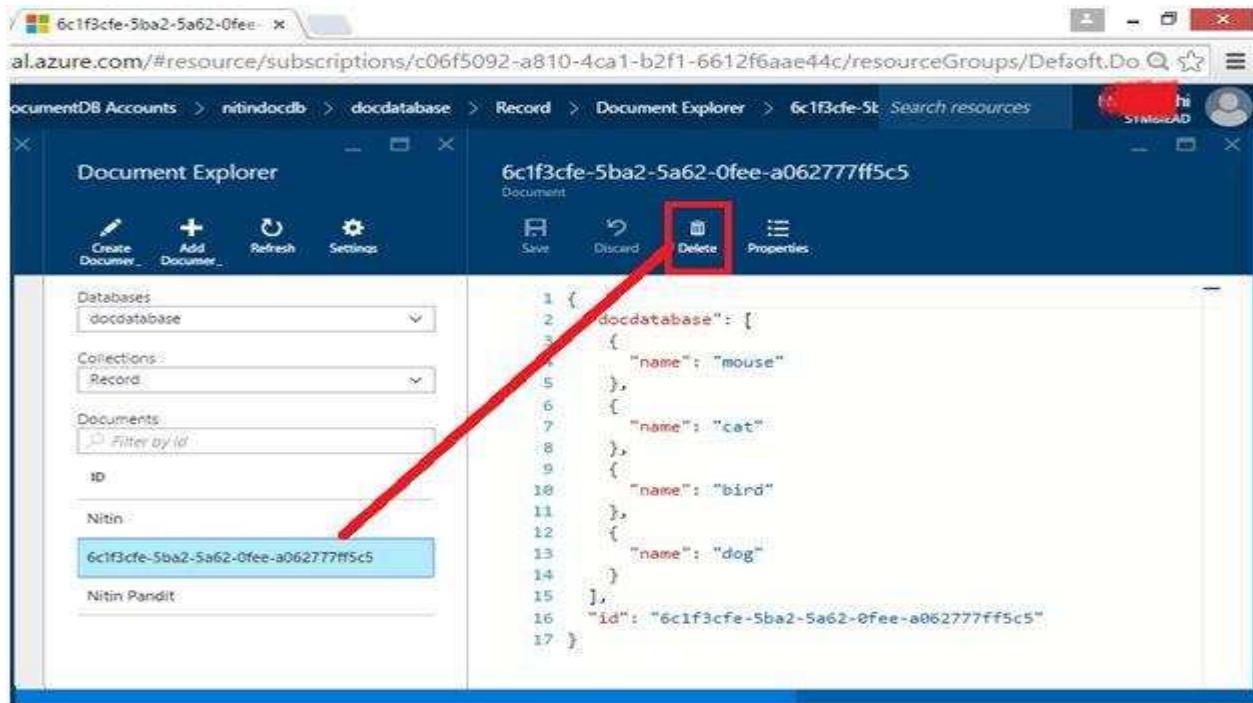
```
1 {  
2   "id": "Nitin Pandit",  
3   "year": "2016",  
4   "Country": "India",  
5   "Hobbies": "Playing cricket, travelling, reading novels'",  
6   "twitter": "@thinkaboutnitin",  
7   "Location": "Noida"  
8 }
```

The 'Save' button in the top right is highlighted with a red box. Below the JSON code, there is a large red outline drawing of a house.

The changes can be seen in the left side of the window, with “*Nitin Pandit*” ID.

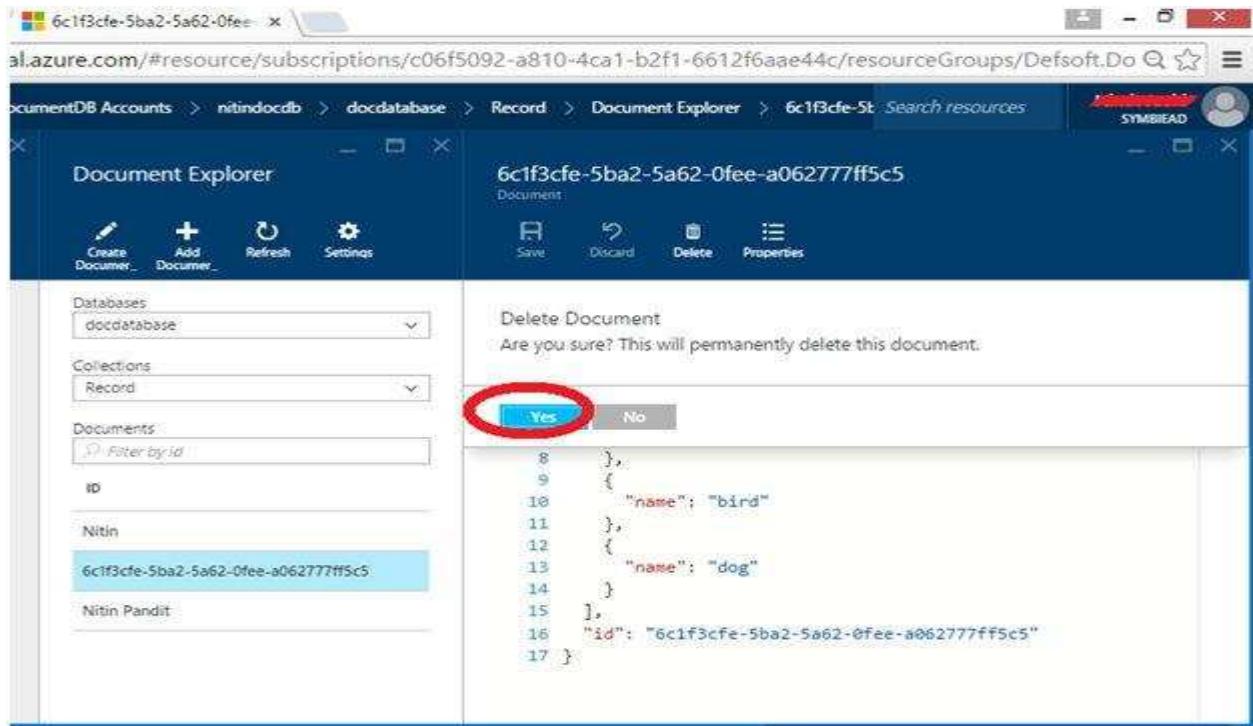


By the Delete command, you can delete the existing record, let's do it, I have record with default ID, click on **Delete** command to delete this document.

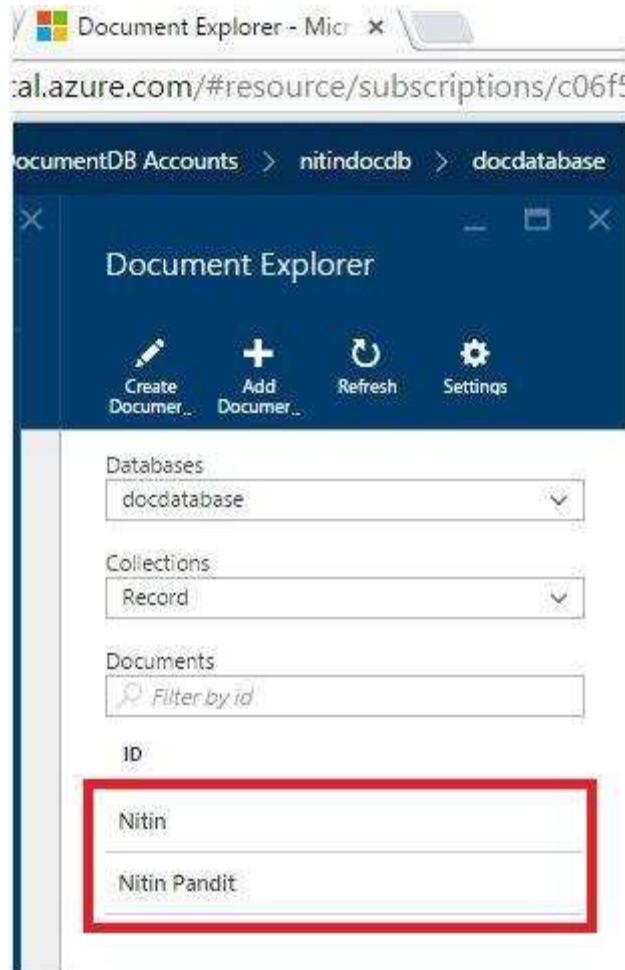


Click on “Yes” button.

The document will be deleted.



So, I had successfully deleted previous record and now only two collections are remaining.



The screenshot shows the Microsoft Document Explorer interface. The title bar indicates it is running in Microsoft Edge. The URL bar shows a connection to a Microsoft Azure account. The navigation path is: documentDB Accounts > nitindocdb > docdatabase. The main area is titled "Document Explorer" and contains the following controls: Create Document, Add Document, Refresh, and Settings. Below these are three dropdown menus: Databases (set to "docdatabase"), Collections (set to "Record"), and Documents (with a "Filter by id" input field). The "ID" section lists two items: "Nitin" and "Nitin Pandit". The entry "Nitin Pandit" is highlighted with a red rectangular border.

Querying In the Azure DocumentDB

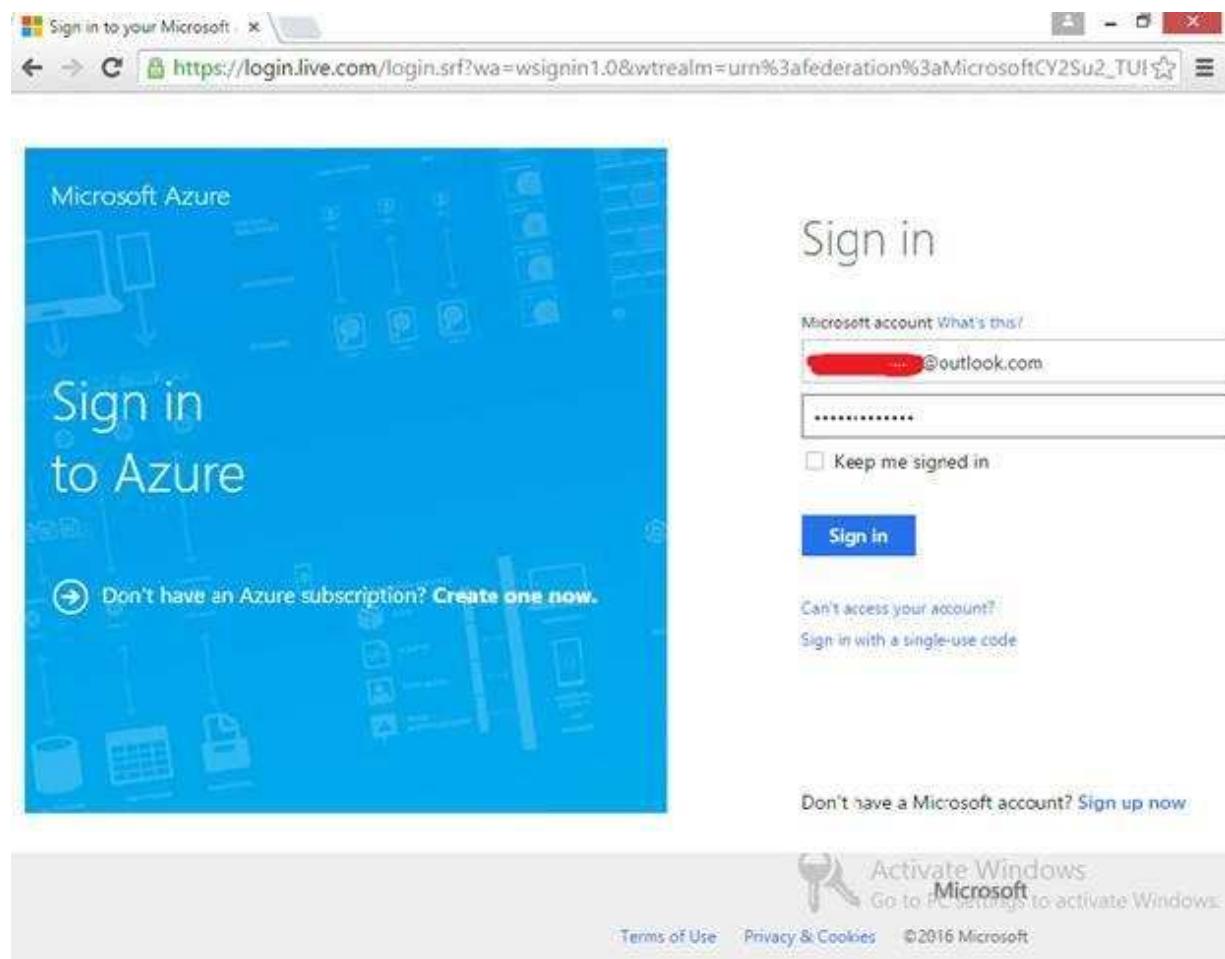
As you easily understood DocumentDB through the previous chapter, now the question is:

Question: Can we perform the querying in DocumentDB also?

So, the answer is yes, we can perform edit, delete, and create querying operations in the DocumentDB.

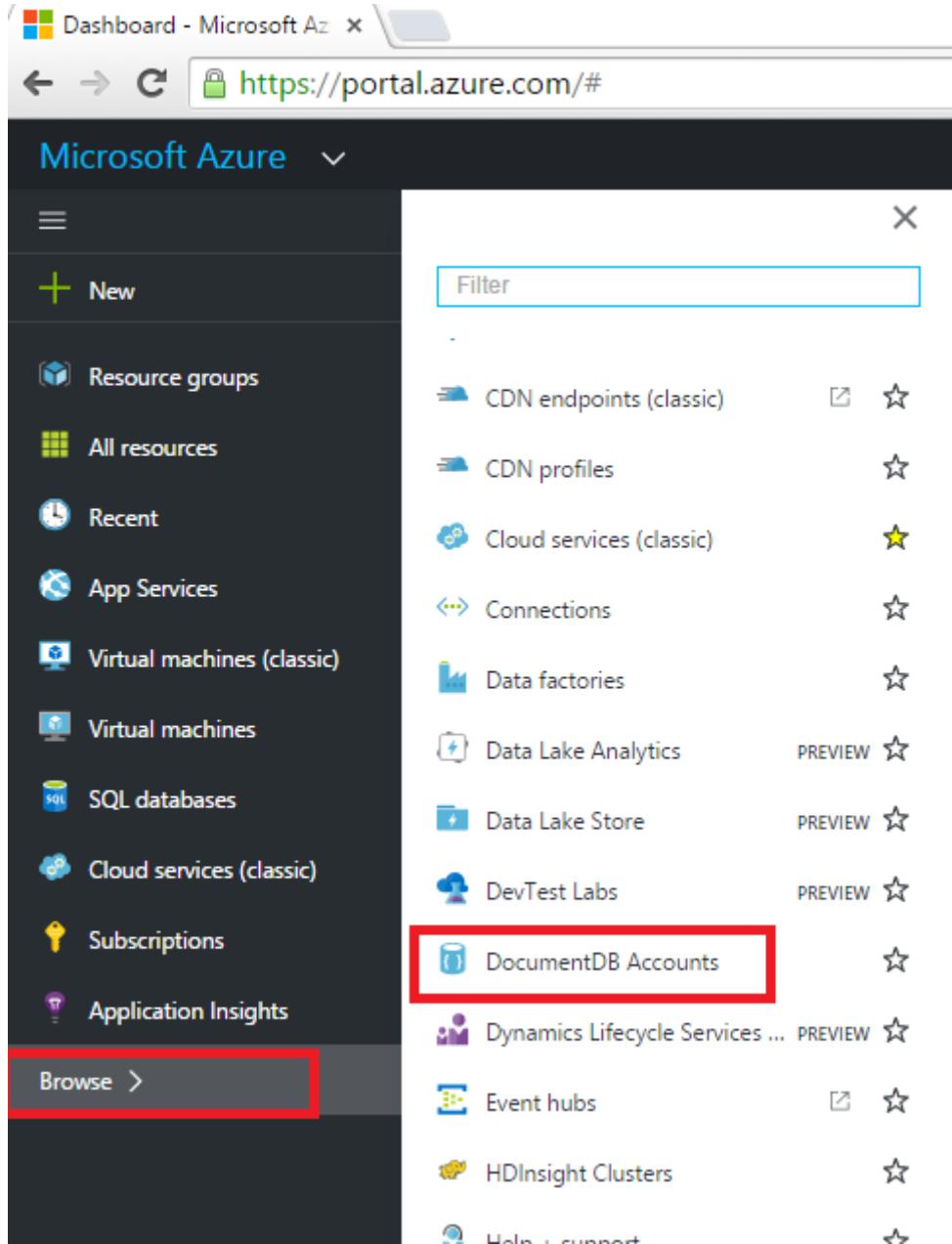
So let's do it step by step as in the following:

Go to Microsoft Azure Sign In window and fill the username and password to log in your account.



In the Azure portal, scroll down and go to the “**Browse**” option and select “**DocumentDB Accounts**”.

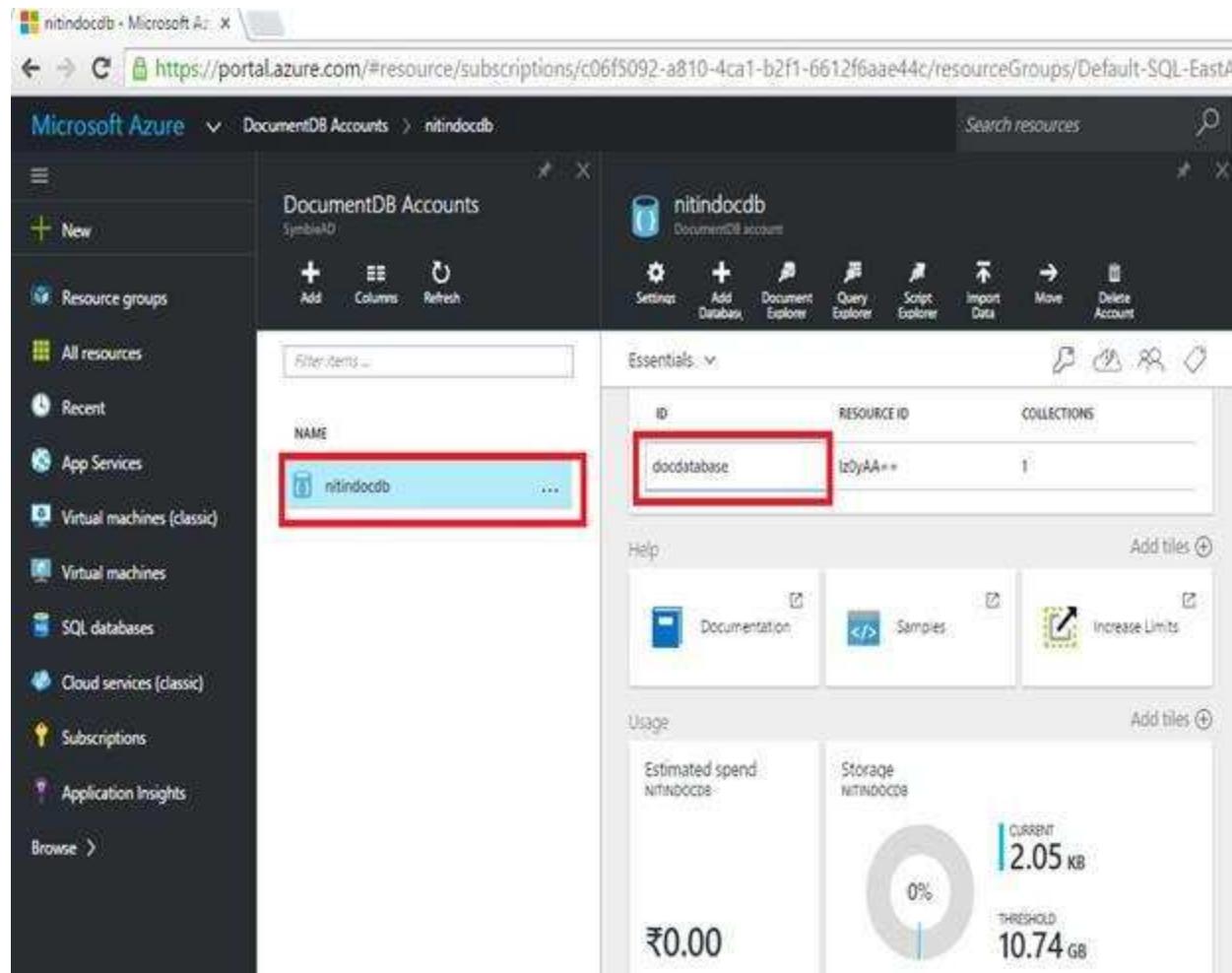
Note: I already created DocumentDB Account in the previous article.



The screenshot shows the Microsoft Azure portal dashboard. On the left, there is a sidebar with various navigation options: Resource groups, All resources, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Subscriptions, Application Insights, and a red-highlighted 'Browse >' button. The main content area displays a list of services with a filter bar at the top. The services listed are: CDN endpoints (classic), CDN profiles, Cloud services (classic), Connections, Data factories, Data Lake Analytics (PREVIEW), Data Lake Store (PREVIEW), DevTest Labs (PREVIEW), DocumentDB Accounts (highlighted with a red box), Dynamics Lifecycle Services ... (PREVIEW), Event hubs, HDInsight Clusters, and Help + support. Each service entry includes a star icon for favoriting.

You will see your DocumentDB Account in the next window, if there is no account simply go to the “Add” option and create your account.

Here I have my account “nitindocdb,” I’m going to click on it, clicking it will redirect to next window. In this window we can find our database, if it is there, scroll down and click on your database, I have created “docdatabase” named database. If there you don’t have any database, go to the “Add Database” option in the menu.

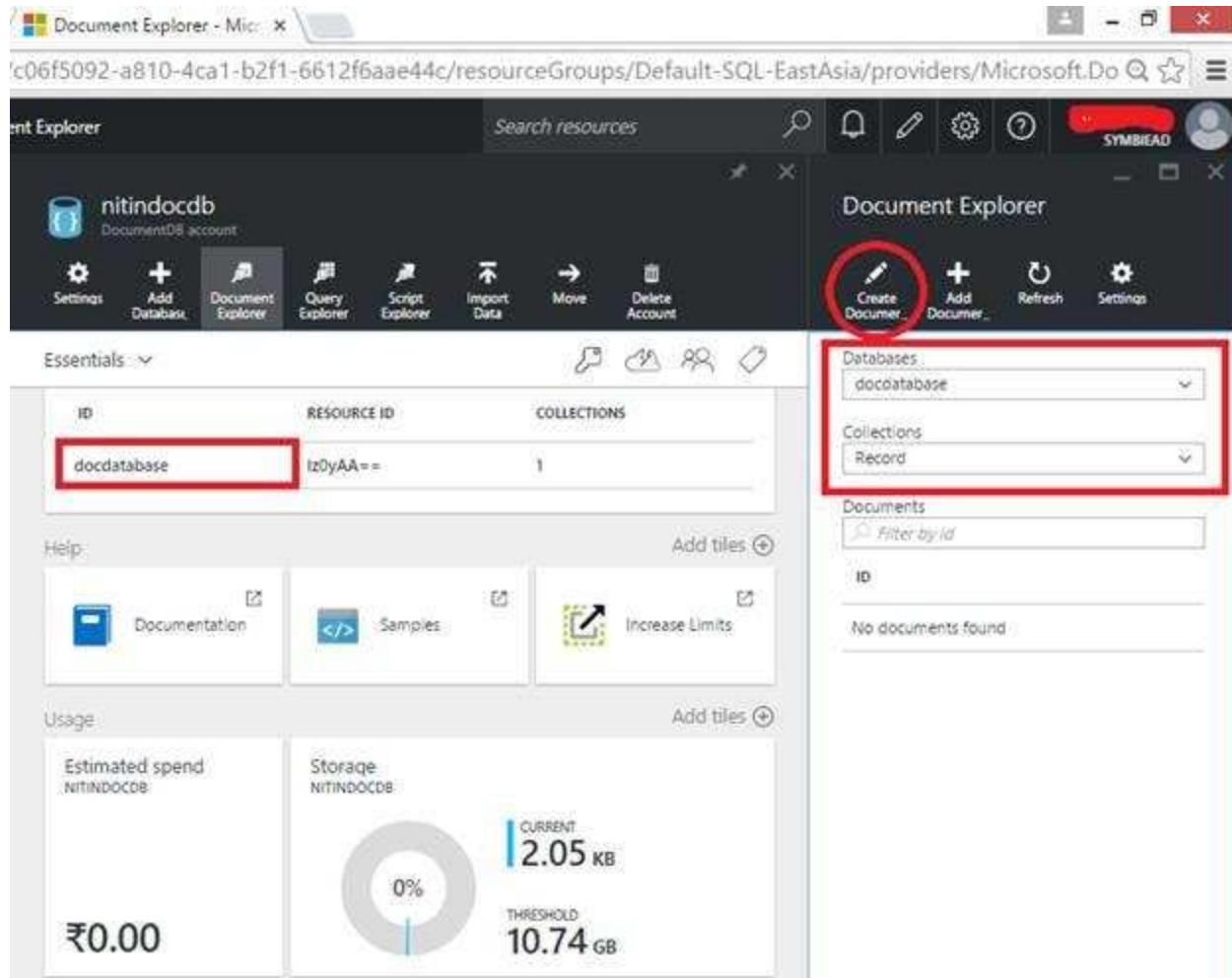


The screenshot shows the Microsoft Azure portal interface for DocumentDB Accounts. On the left sidebar, under the 'DocumentDB Accounts' section, the 'nitindocdb' account is selected. The main pane displays the 'nitindocdb' account details. A table lists databases, with the first entry 'docdatabase' highlighted by a red box. The table columns are ID, RESOURCE ID, and COLLECTIONS. The 'docdatabase' row has an ID of 'Iz0yAA==', a RESOURCE ID of 'Iz0yAA==', and 1 collection. Below the table, there are sections for Help (Documentation, Samples, Increase Limits) and Usage (Estimated spend: ₹0.00, Storage: Current 2.05 KB, Threshold 10.74 GB).

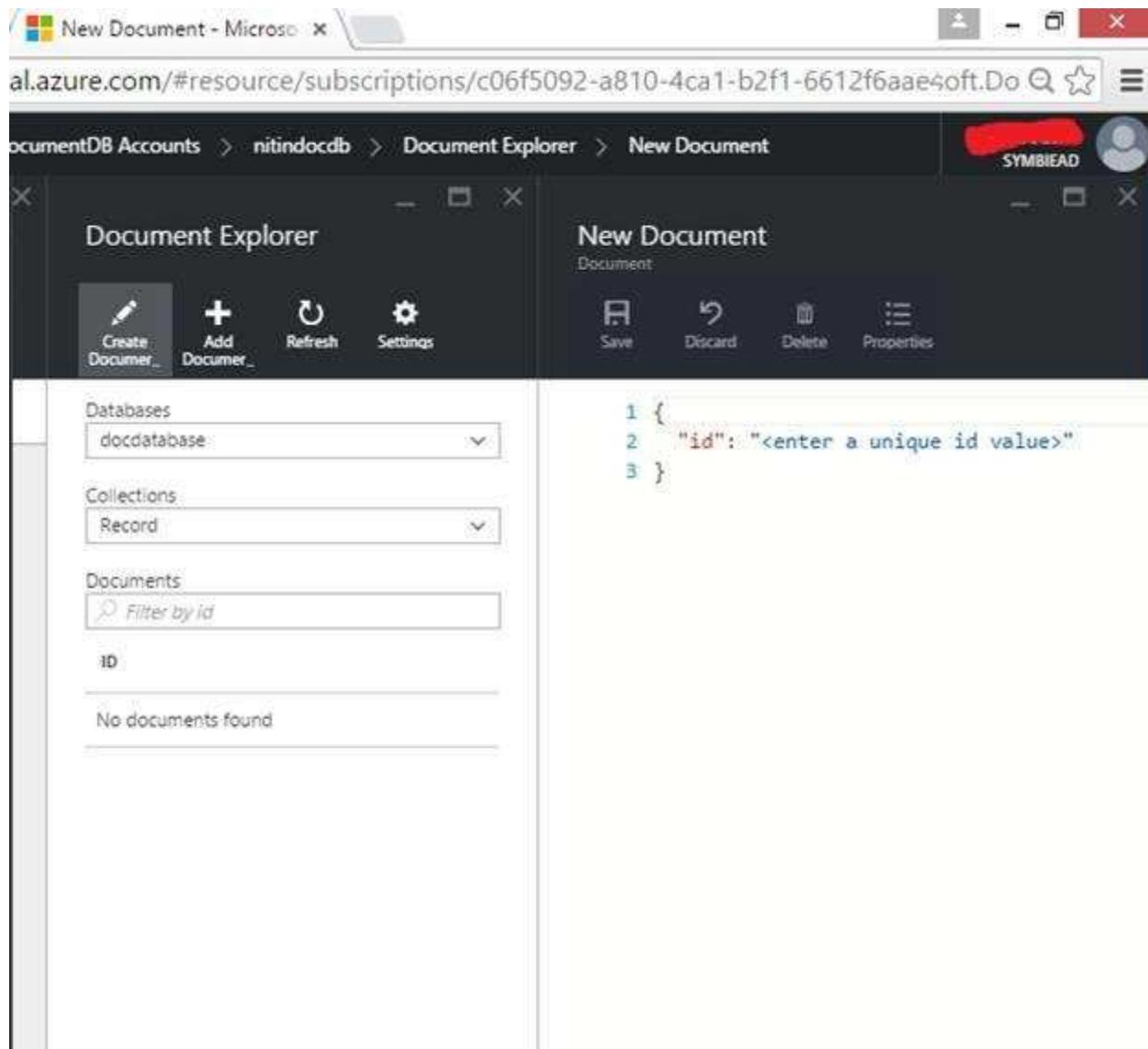
ID	RESOURCE ID	COLLECTIONS
docdatabase	Iz0yAA==	1

After clicking on the database you will be redirected to the “Document Explorer” window, where you will find your created database and collations.

I need to create documents, so click “**Create Document**”.



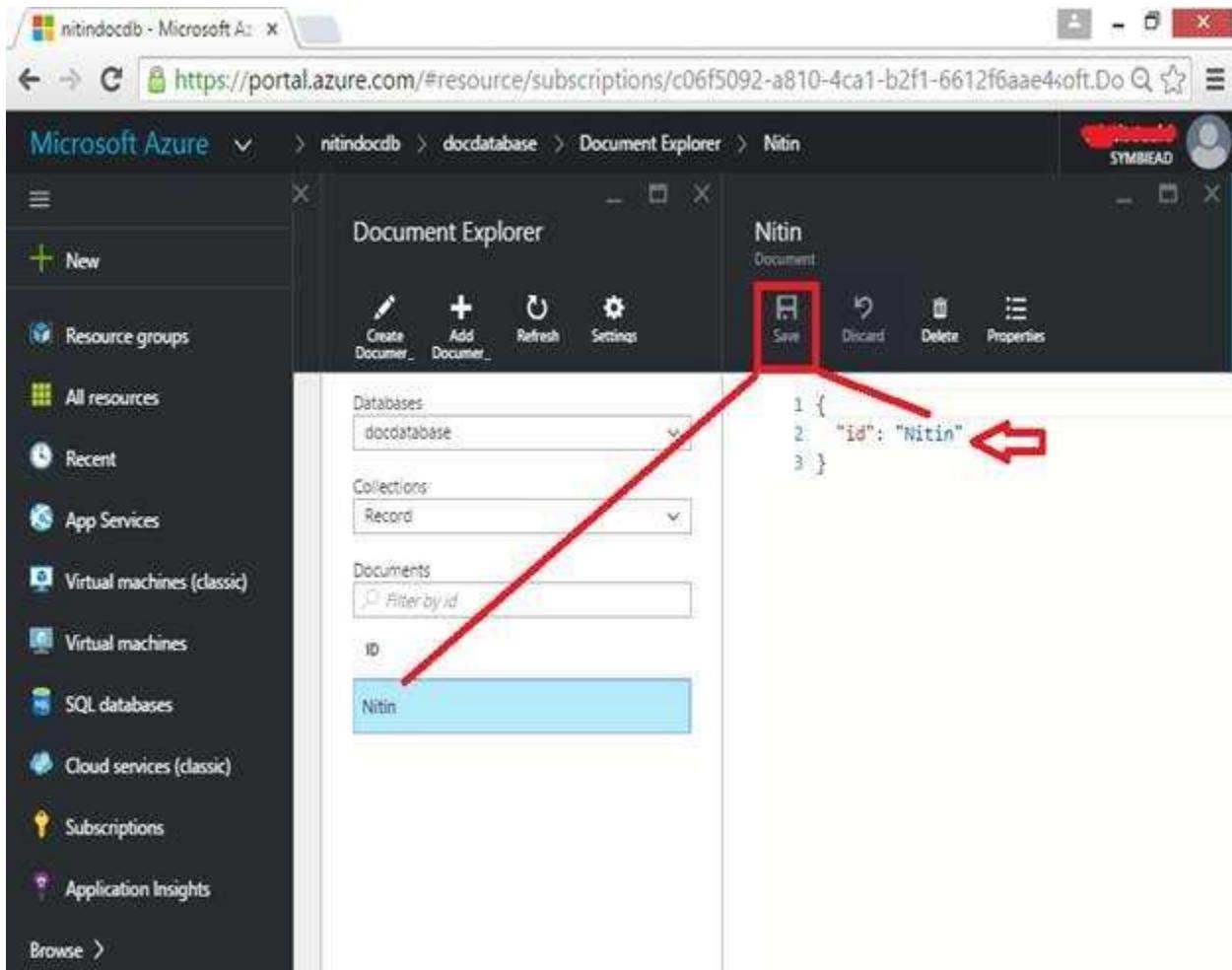
In the next figure, you can see there is a JSON type script where we need to enter a unique id value; it could be your name or your friend's name.



Here, I gave the name “Nitin” in the ID section.

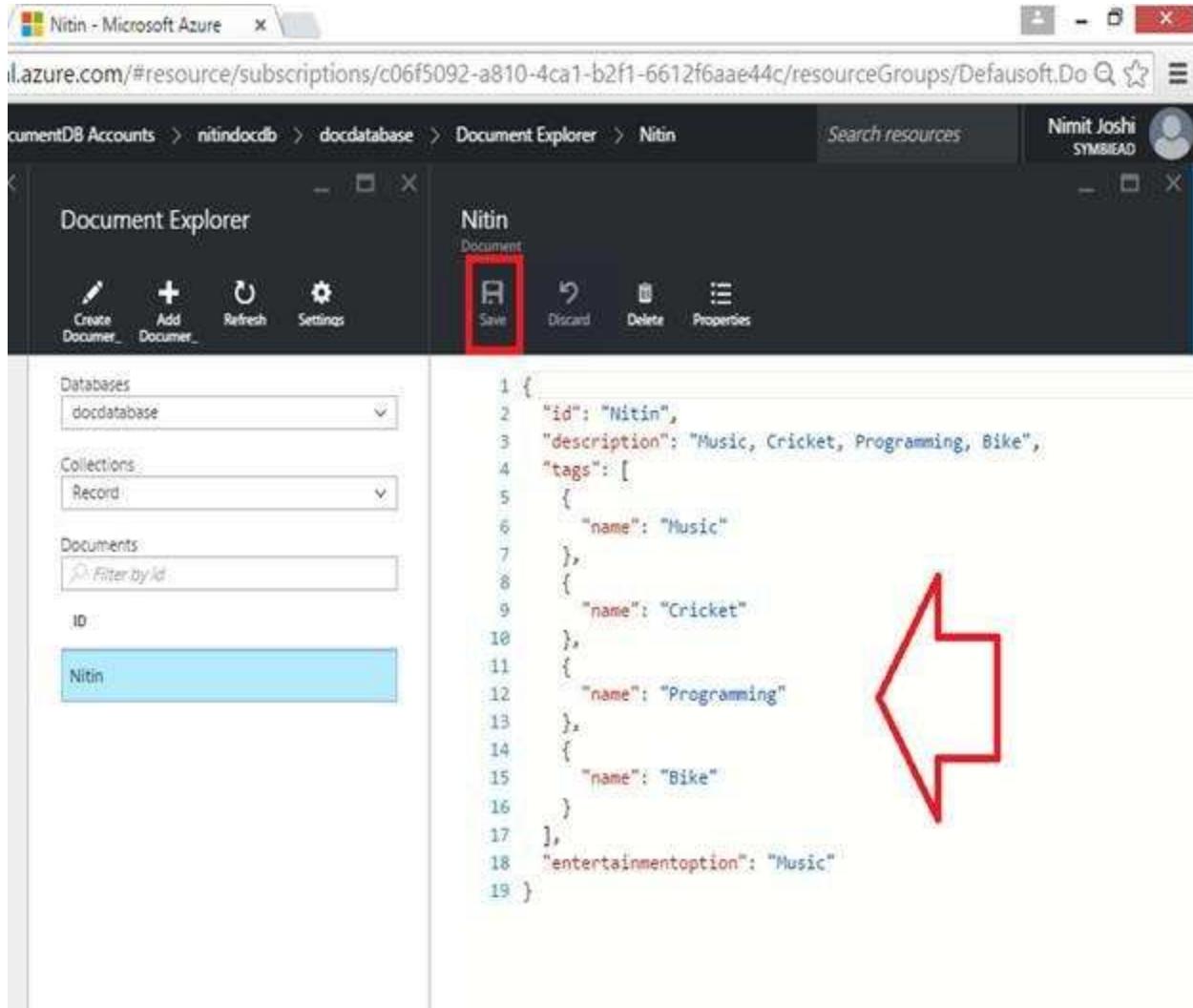
Click “Save” button to save the script.

After clicking on the “Save” button, it gives you a particular id. (See in the Document Explorer).



I'm going to write some JSON in the same document, you can write any type of JSON script in that document.

After completing your script, click on the “Save” button to save your script.

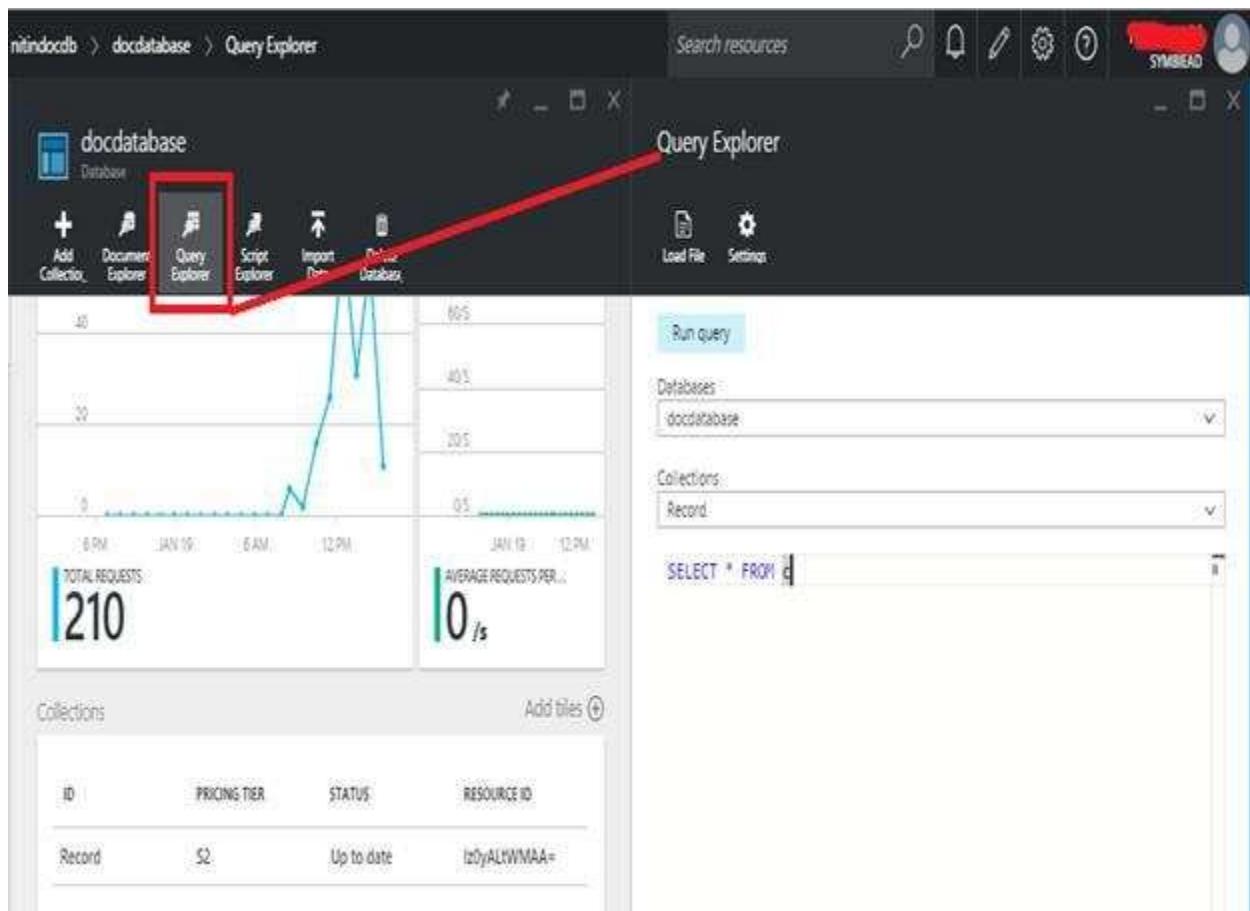


Script will be saved; you can call your script through the query, so let's go for querying.

Go to backward in the menus and select “**Query Explorer**”.

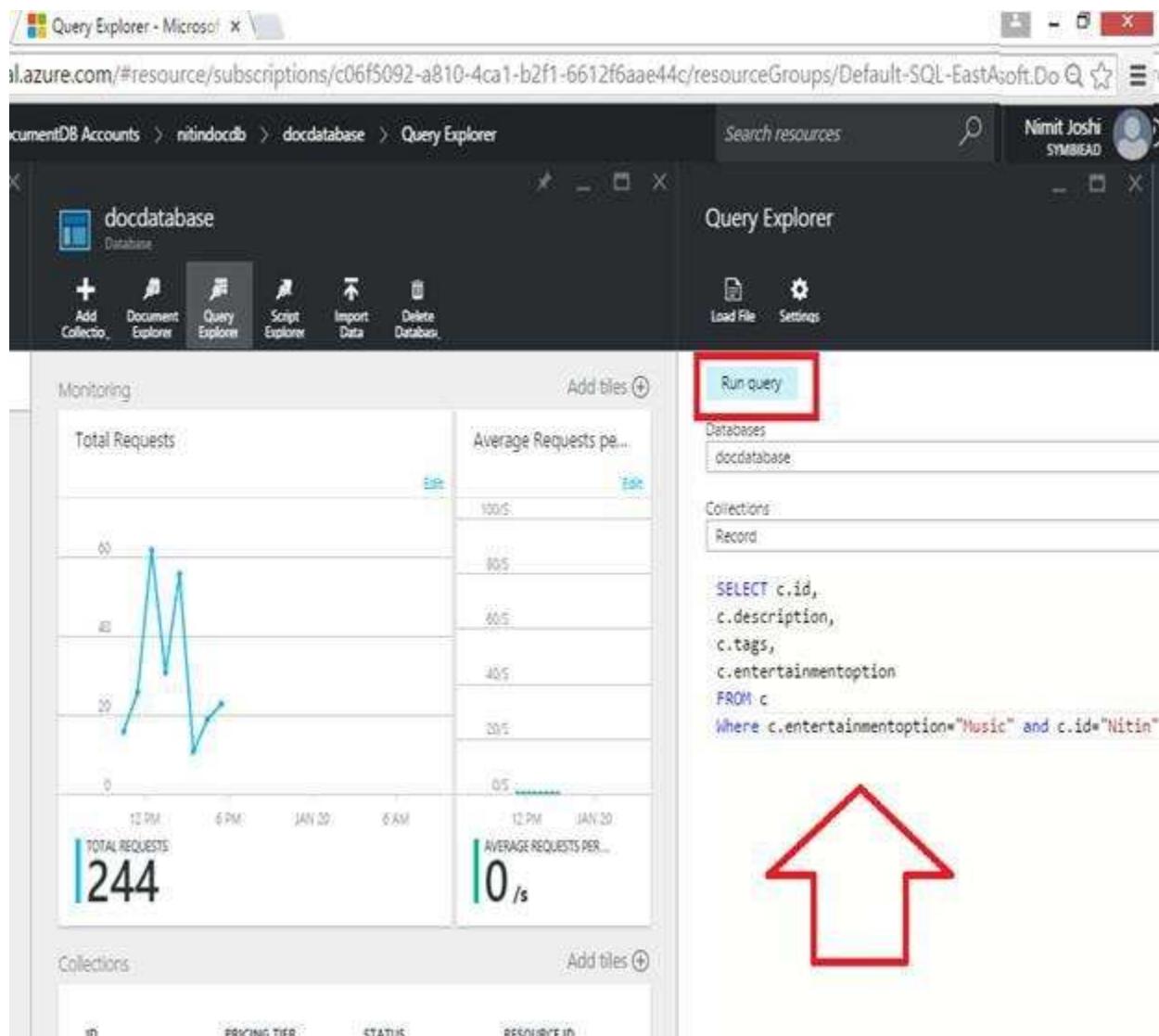
In the Query Explorer window you will see a short command with your **Database** and **Collections**.

Here c is an object. We can call our JSON property through the c object, here c object is not mandatory for you, you can take any type of object like (a,b,f.....etc).



In the next figure, I am adding some more commands and calling my JSON property through the `c` object.

After completing the command process, hit on the “Run query” button.

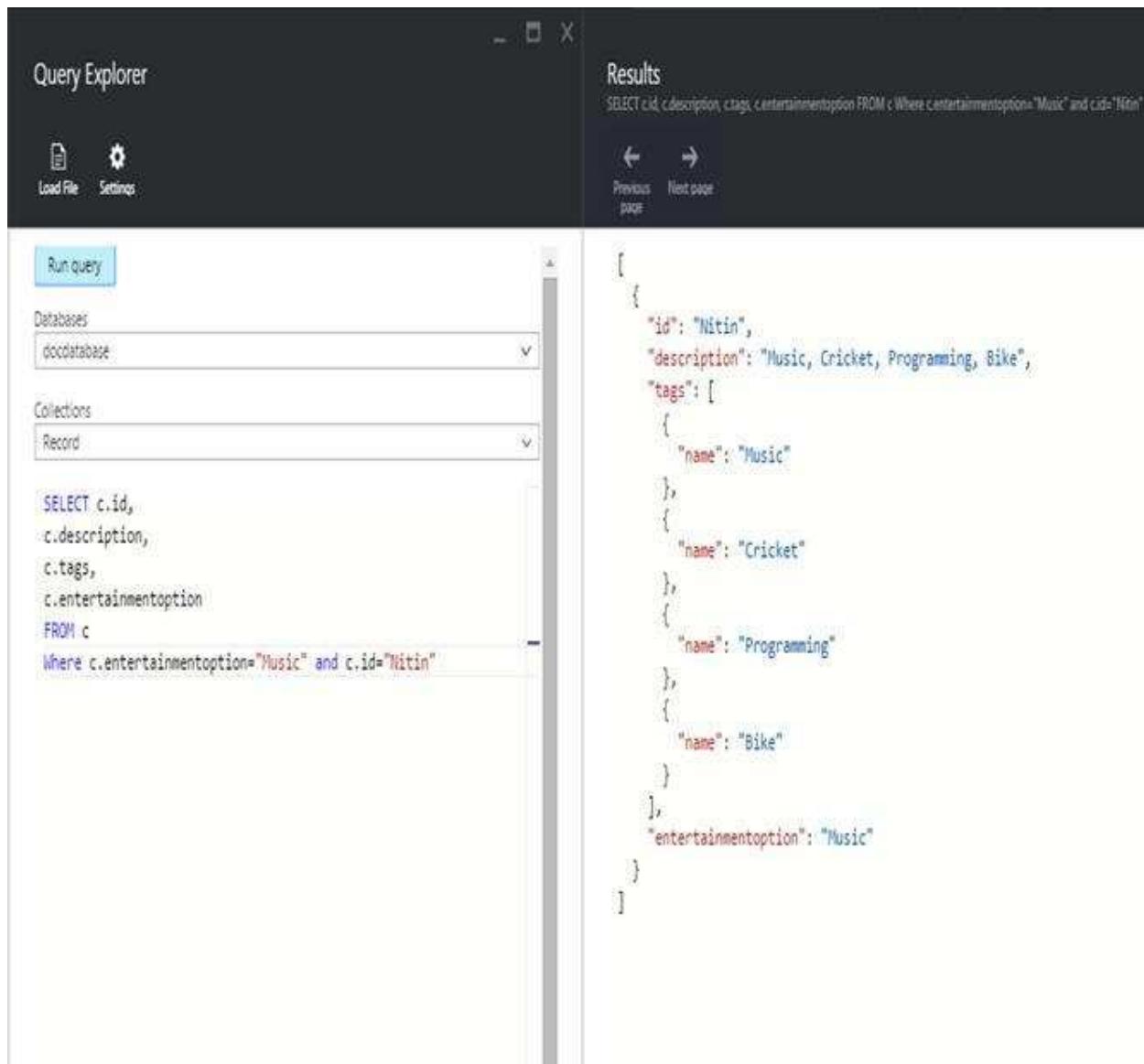


The screenshot shows the Microsoft Azure DocumentDB Query Explorer interface. At the top, there's a navigation bar with 'Query Explorer - Microsoft' and a URL 'al.azure.com/#resource/subscriptions/c06f5092-a810-4ca1-b2f1-6612f6aae44c/resourceGroups/Default-SQL-EastAsoft.Do'. Below the navigation bar, the path 'DocumentDB Accounts > nitindocdb > docdatabase > Query Explorer' is visible. On the left, there's a sidebar with icons for 'Add Collection', 'Document Explorer', 'Query Explorer' (which is selected), 'Script Explorer', 'Import Data', and 'Delete Database'. The main area has a title 'Query Explorer' and two tabs: 'Load File' and 'Settings'. A red box highlights the 'Run query' button. To the right of the button, there are sections for 'Databases' (showing 'docdatabase') and 'Collections' (showing 'Record'). Below these sections is a SQL query:

```
SELECT c.id,  
       c.description,  
       c.tags,  
       c.entertainmentoption  
FROM c  
Where c.entertainmentoption="Music" and c.id="Nitin"
```

On the far right, there is a large red arrow pointing upwards.

I have successfully obtained my result that is JSON script, in which I have defined in the document explorer. So we can edit, delete, and create our script according to use, and can call the script through the query.



The screenshot shows the Microsoft DocumentDB Query Explorer interface. On the left, there's a sidebar with 'Query Explorer' at the top, followed by 'Load File' and 'Settings' buttons. Below that is a 'Run query' button, which is highlighted with a blue border. Under 'Databases', 'docdatabase' is selected. Under 'Collections', 'Record' is selected. The main area contains a query editor with the following T-SQL code:

```
SELECT c.id,
       c.description,
       c.tags,
       c.entertainmentoption
  FROM c
 WHERE c.entertainmentoption = "Music" AND c.id = "Nitin"
```

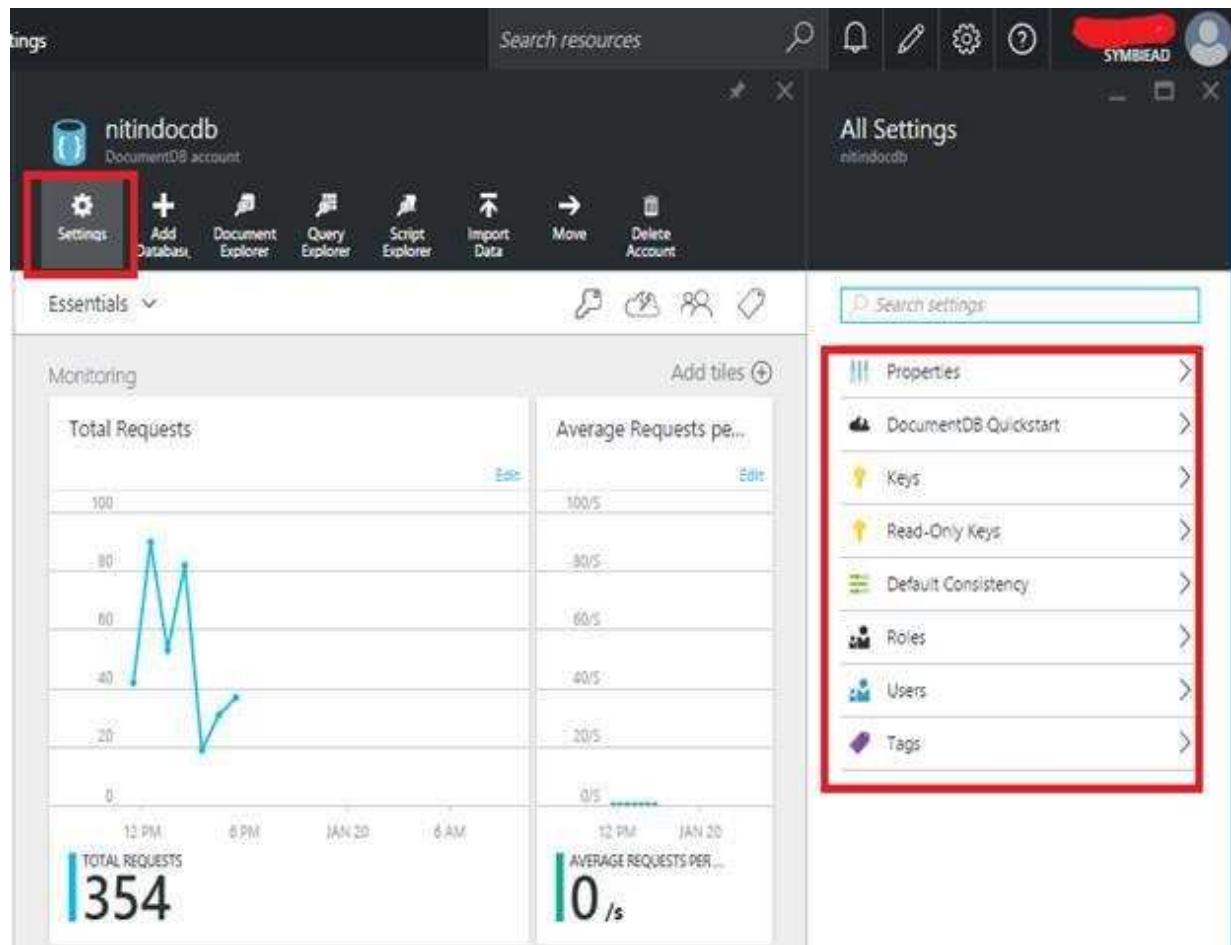
To the right, the results pane shows the output of the query. The output is a JSON array containing one document:

```
[{"id": "Nitin", "description": "Music, Cricket, Programming, Bike", "tags": [{"name": "Music"}, {"name": "Cricket"}, {"name": "Programming"}, {"name": "Bike"}], "entertainmentoption": "Music"}]
```

Let's understand the menus in the DocumentDB Account.

Settings:

In the DocumentDB the first level would be “**Settings**,” it contains the whole DocumentDB setting information like keys of the DocumentDB account, where you can find the”URI” and “Keys” of the account. In the properties, you can find the status and location of your account.



The screenshot shows the Azure DocumentDB settings interface for the database 'nitindocdb'. The left pane displays monitoring data, including a line chart of total requests over time and summary statistics: Total Requests (354) and Average Requests per second (0/s). The right pane lists various settings options, with the 'Properties' section highlighted by a red box.

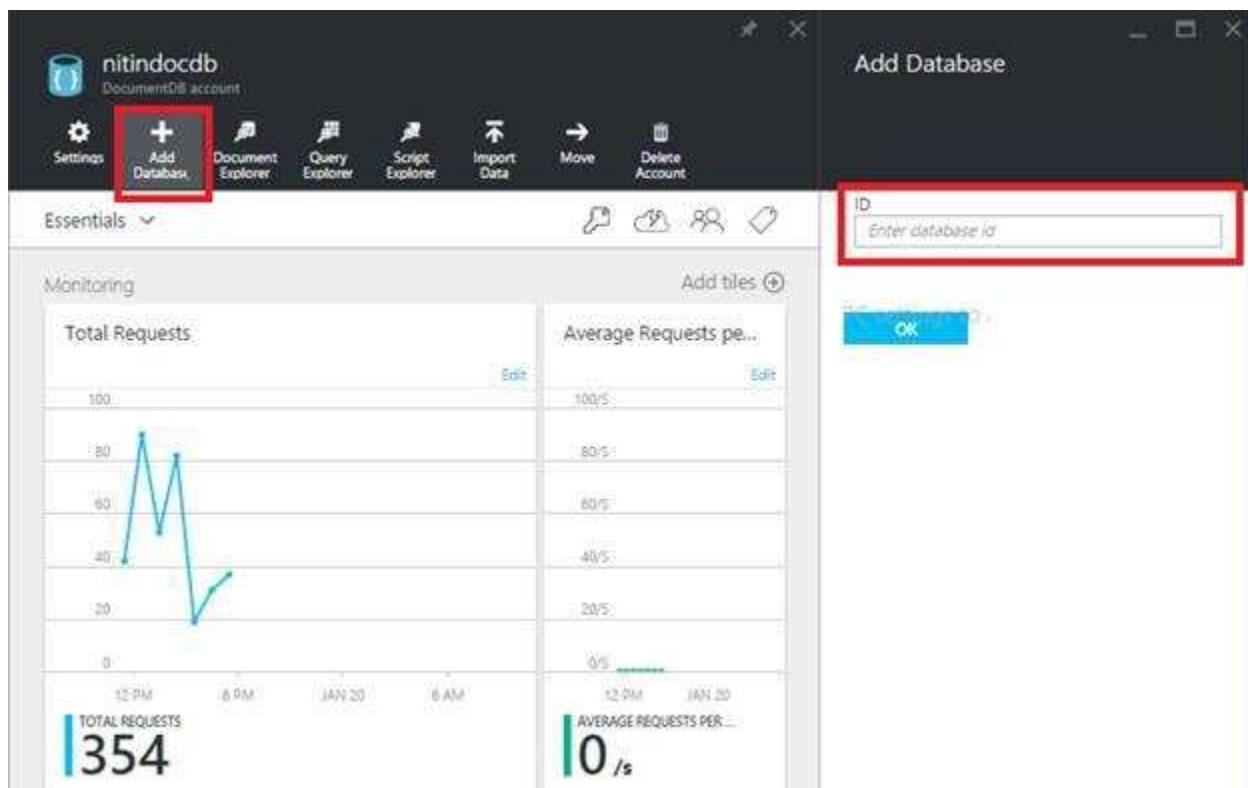
Total Requests	Average Requests per second
100	100/s
80	80/s
60	60/s
40	40/s
20	20/s
0	0/s

Properties

- DocumentDB Quickstart
- Keys
- Read-Only Keys
- Default Consistency
- Roles
- Users
- Tags

Add Database:

In this you can create a particular database, with the unique id, where you can add collections.



Document Explorer:

Document Explorer is a place where you can see your databases and collections in the dropdown menu. In this, you can also see your list of documents, and also it means means your created and uploaded documents in the JSON extension.

Let's focus on Document Explorer menu-

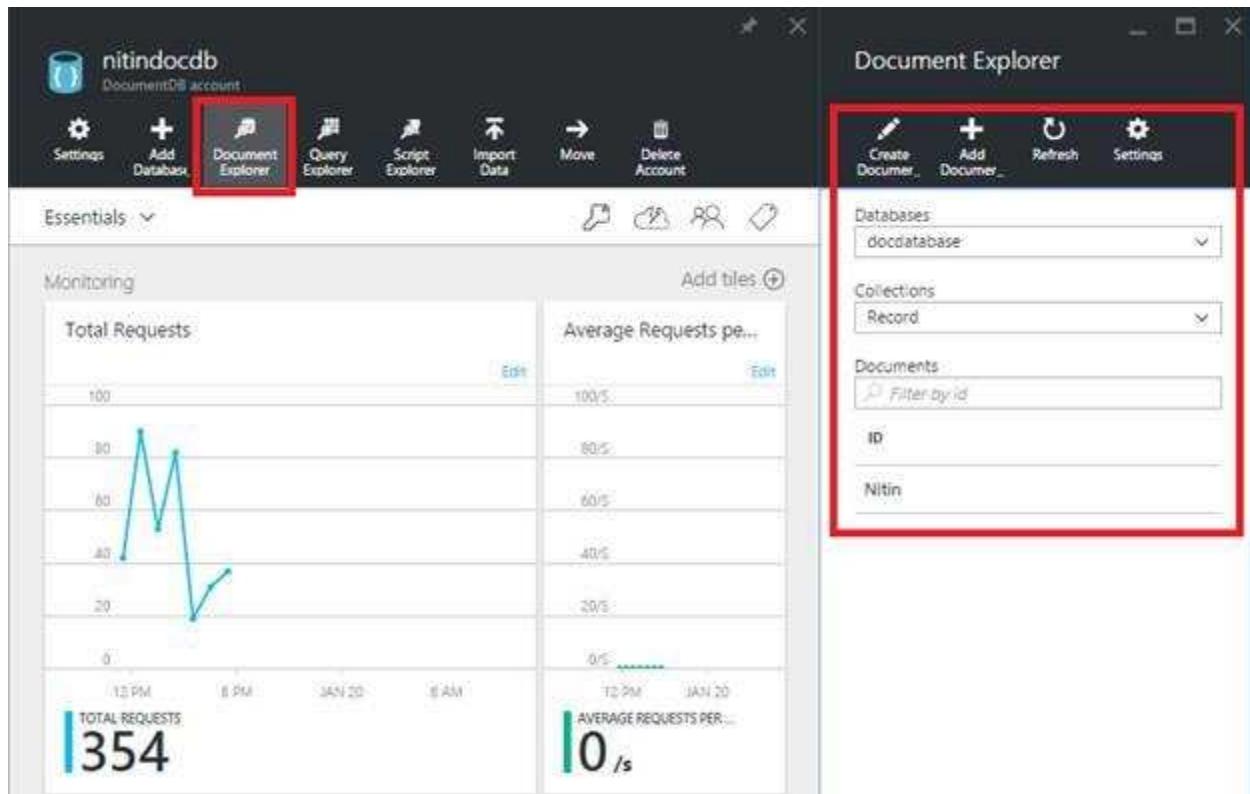
Create Document:

In this, you can create JSON scripts; after saving the JSON scripts, it will save as a ID and known as document.

Add Document: You can upload 100 JSON documents.

Refresh: You can refresh the Document Explorer.

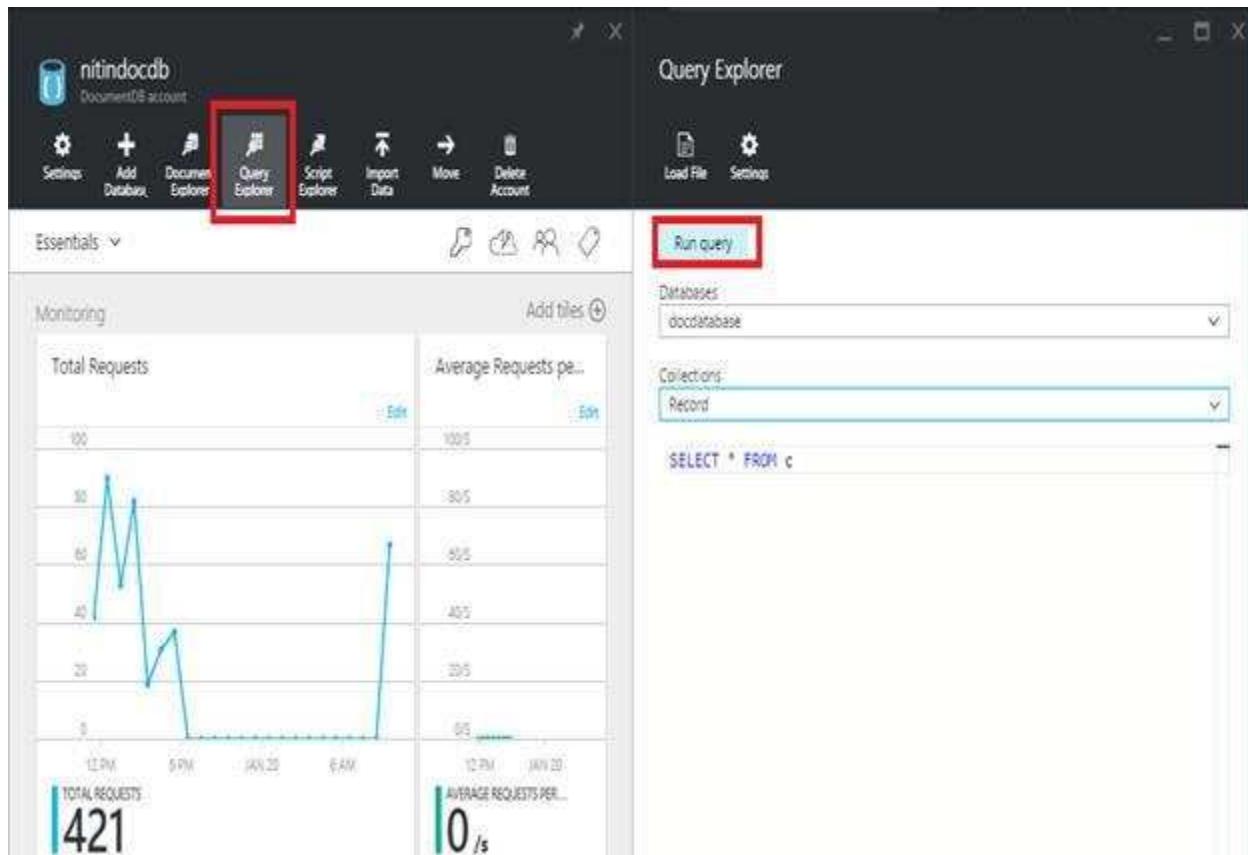
Settings: In the settings, you can select the items per page like custom or unlimited.



Query Explorer:

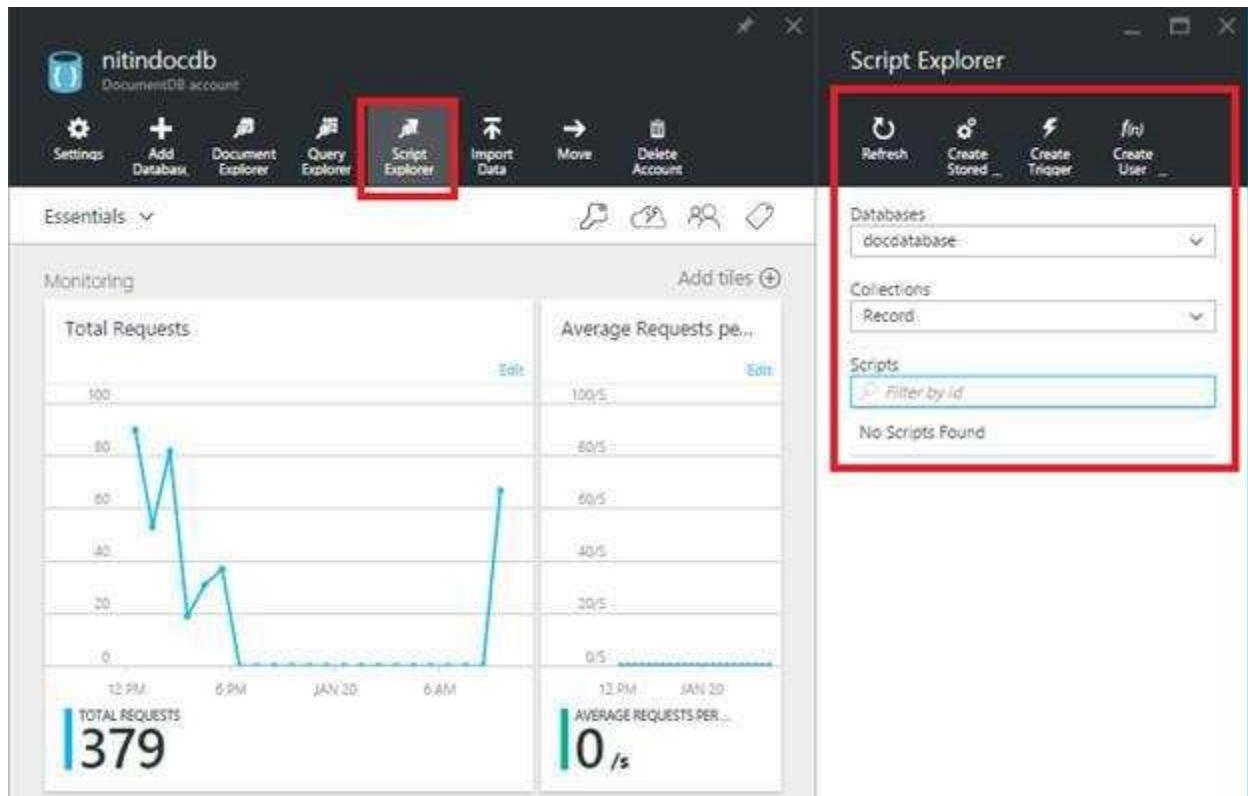
In the Query Explorer, you can choose the database and collections according to use, which means in which database and collection you want to perform Querying.

You can also Load the Query from the outsource collection, and perform the operations on that selected script.



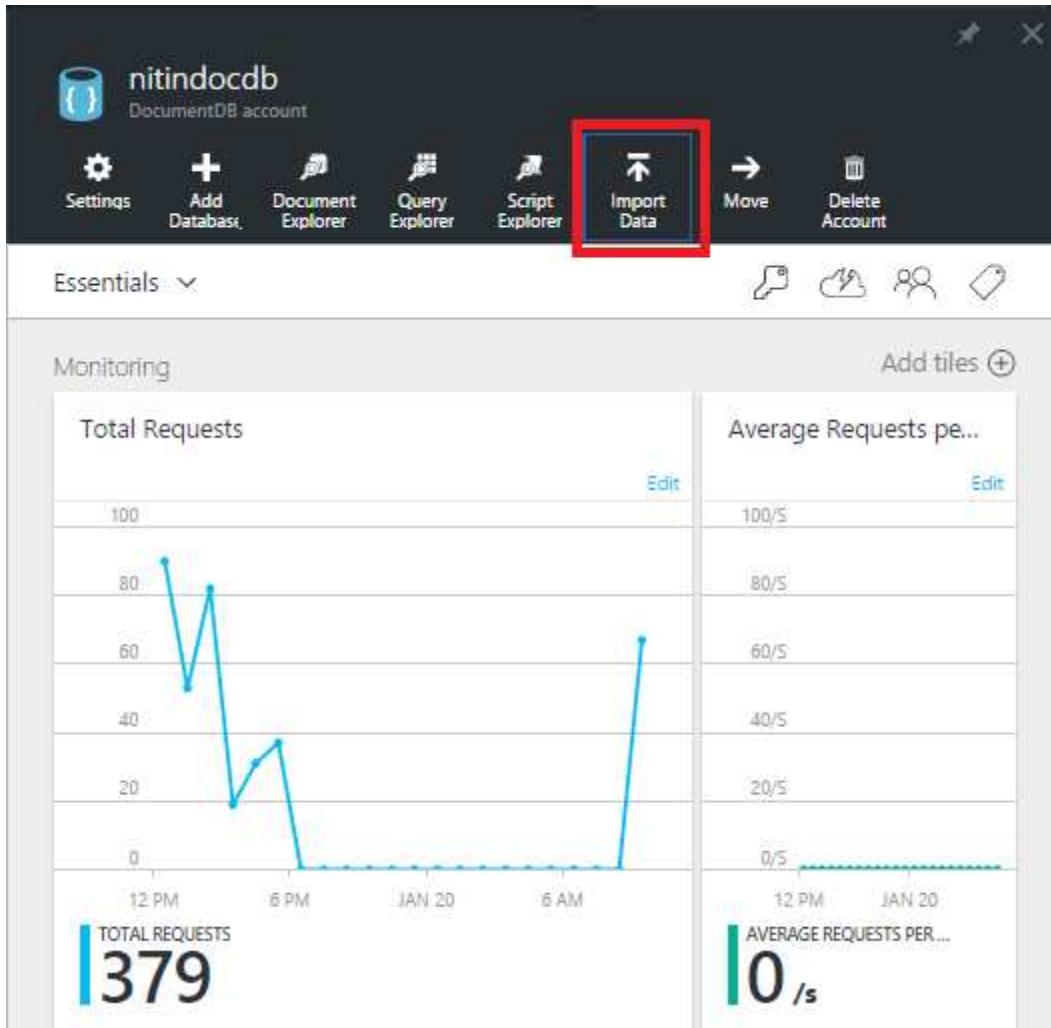
Script Explorer:

It is one of the important menu items of the Document Explorer. In Script Explorer you can create stored procedures and triggers as well as user defined functions. And if you save any script then it gets saved as a script.



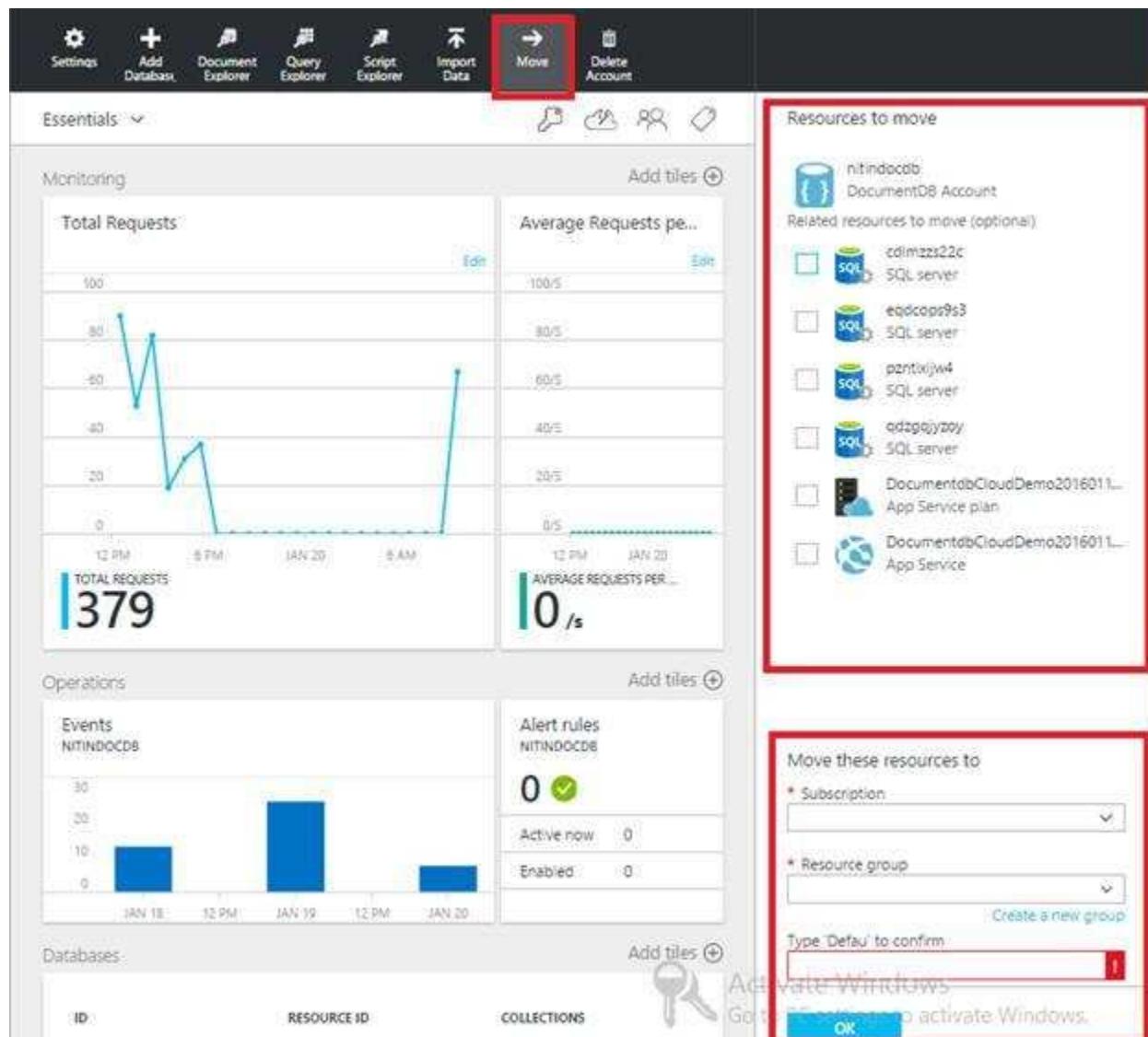
Import Data: You can import data in to DocumentDB also,

- You can import JSON file data to DocumentDB.
- You can import CSV file data to DocumentDB.
- You can import SQL Server data to DocumentDB.
- You can import MongoDB data to DocumentDB.
- You can import data from Azure Table storage to DocumentDB.
- You can import data from Amazon DynamoDB to DocumentDB.
- You can import data from HBase to DocumentDB.
- You can migrate data between DocumentDB collections.



The screenshot shows the Azure DocumentDB account management interface. At the top, there's a toolbar with icons for Settings, Add Database, Document Explorer, Query Explorer, Script Explorer, Import Data (which is highlighted with a red box), Move, and Delete Account. Below the toolbar, there are sections for Essentials and Monitoring. The Monitoring section displays two charts: 'Total Requests' and 'Average Requests per second'. The 'Total Requests' chart shows a fluctuating line graph with a total count of 379. The 'Average Requests per second' chart shows a constant value of 0/s.

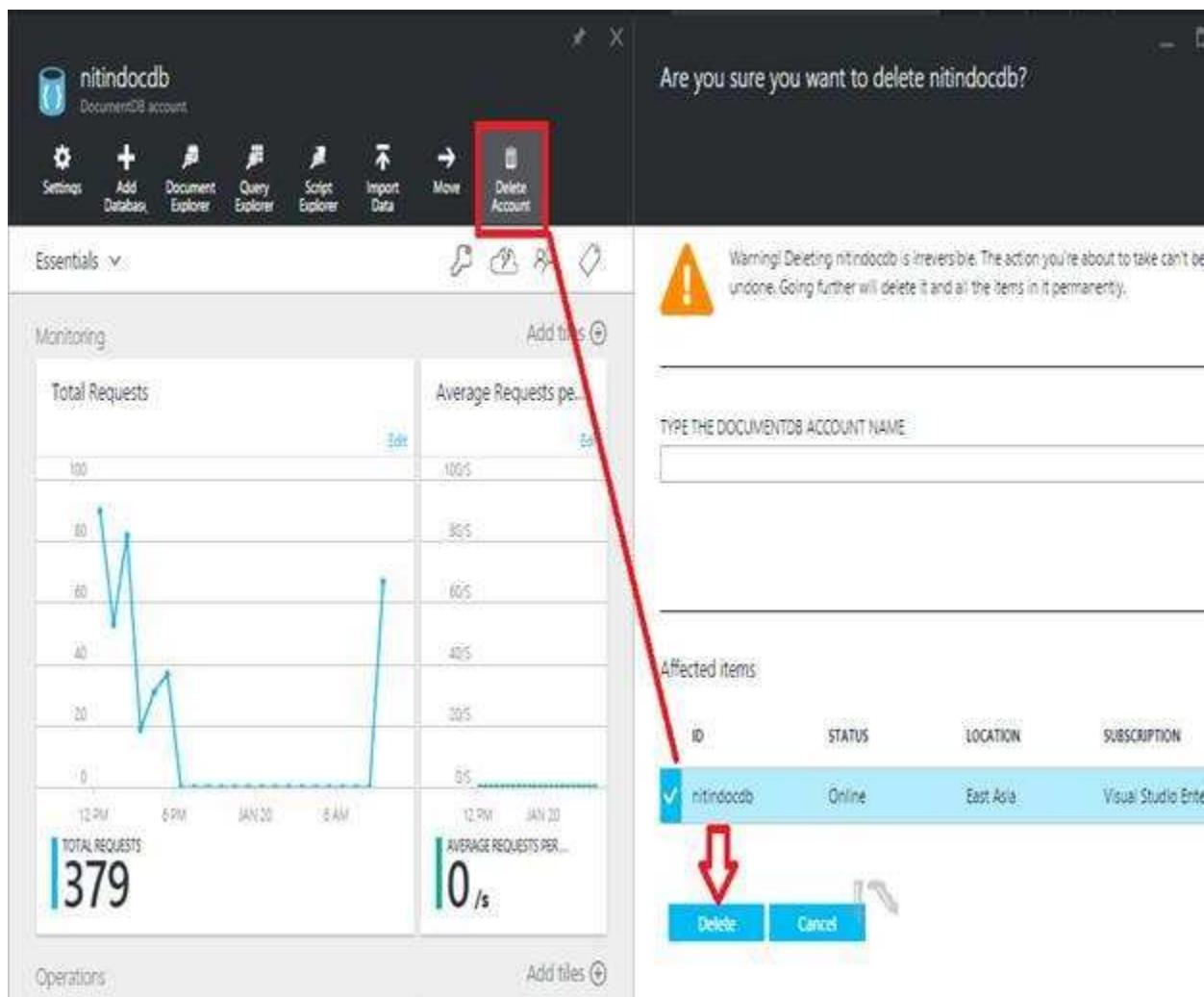
Move: You can move to resources.



The screenshot shows the Azure portal interface with the 'Move' button highlighted in a red box at the top right of the header bar. To the right of the main dashboard, there are two large red-bordered boxes. The top box is titled 'Resources to move' and lists several resources: 'nitindocdb: DocumentDB Account' (selected), 'cdimz2s22c: SQL server', 'eqdcop9s3: SQL server', 'pzntxjw4: SQL server', 'qdzgajyzoy: SQL server', 'DocumentdbCloudDemo2016011...: App Service plan', and 'DocumentdbCloudDemo2016011...: App Service'. The bottom box is titled 'Move these resources to' and contains fields for 'Subscription' (set to 'Default') and 'Resource group' (set to 'Create a new group'). A message at the bottom says 'Type "Default" to confirm' and 'Activate Windows' with an 'OK' button.

Delete Account:

You can delete your accounts, by simply selecting your account which you want to delete and click on “Delete Account” button.



Manage Azure Mobile Services

What is Azure Mobile Services?

Azure Mobile Services is one of the Microsoft Azure features; Microsoft Azure provides many features like Virtual Machine, Cloud Services, App Services, SQL Databases, etc. Window Azure mobile services client libraries are available in many platforms, and can be used on IOS, Android, Windows Phone, and Windows Store Apps.

Azure Mobile Services has many **advantages**:

- You can manage tables for storing data of the apps easily.
- Easy to connect scalable cloud backend to your mobile app.
- Push Notifications
- You can easily store structured data in the cloud.

Storing Data in the Cloud

Storing Data in the Cloud with Microsoft Azure Mobile Services is very easy for clients. Whenever we are going to create Mobile Service, it automatically takes the SQL Database inside Microsoft Azure.

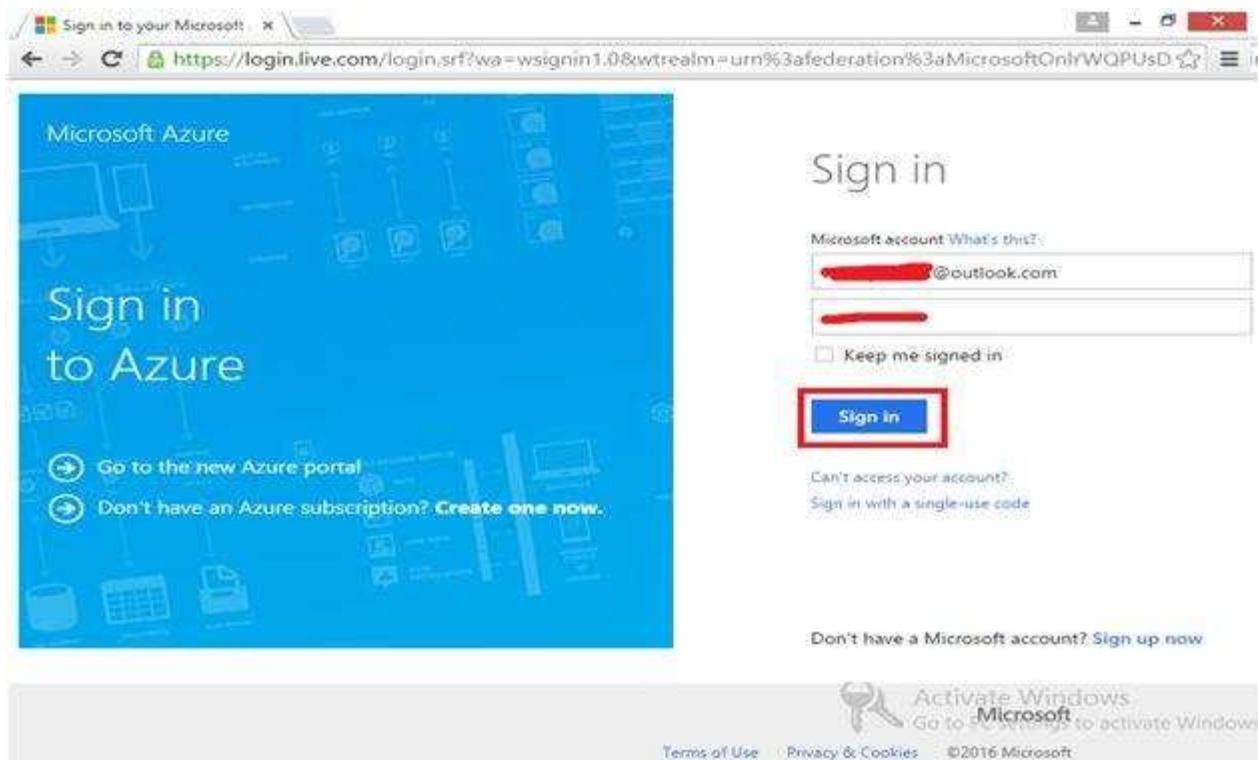


Firstly, I need to create server and database for our Mobile Service, where I can save my information of the application.

For creating SQL Database in Azure, first of all I need to create a server for the database, where I can manage multiple databases.

For this, follow the simple steps:

Step 1: Login to Microsoft Azure.

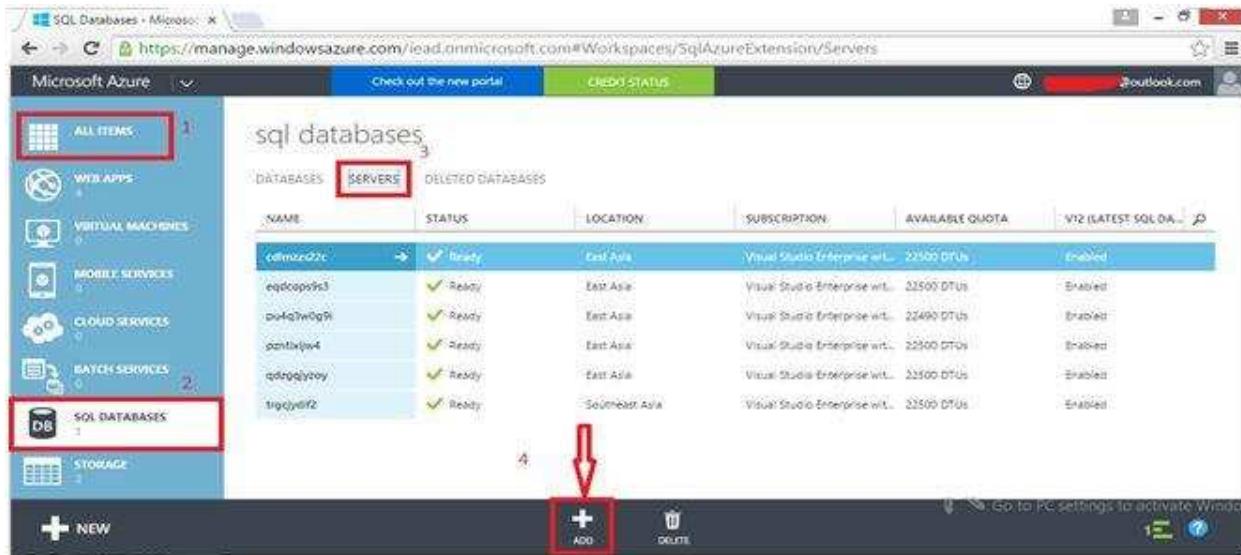


Step 2: In “All Items” panel, go to the “SQL Databases” and click on it.

Step 3: Create a server, select “**SERVERS**”.

Step 4: Click on “**Add**” button at the bottom of the page.

You can see there are many servers in the following figure, but I’m going to create a new one.



The screenshot shows the Microsoft Azure portal interface for managing SQL databases. The left sidebar has 'ALL ITEMS' selected under '1'. Under '2', 'BATCH SERVICES' is expanded, showing '3'. Under '3', 'SQL DATABASES' is selected. The main area shows a table of databases with columns: NAME, STATUS, LOCATION, SUBSCRIPTION, AVAILABLE QUOTA, and V12 (LATEST: SQL DA...). A red box highlights the 'SERVER' tab in the top navigation bar. A red arrow points to the 'ADD' button at the bottom center of the page.

Step 5: Give the login name, what you want to like, I have given “Nitin” for the SQL login used to authenticate to the server.

Step 6: Create password for the server, and confirm password.

Step 7: Select appropriate “Region”.

Step 8: Click on check button.

CREATE SERVER

SQL database server settings

LOGIN NAME
 

LOGIN PASSWORD
 

CONFIRM PASSWORD
 

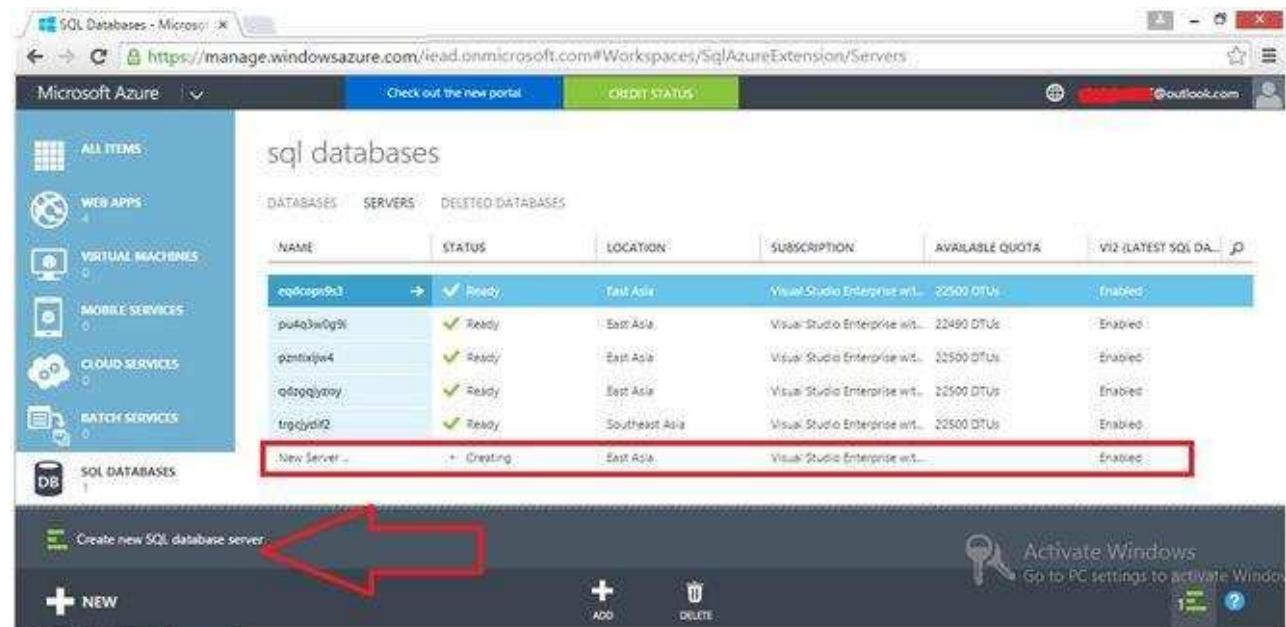
REGION
 

ALLOW WINDOWS AZURE SERVICES TO ACCESS THE SERVER. 

ENABLE LATEST SQL DATABASE UPDATE (V12) 



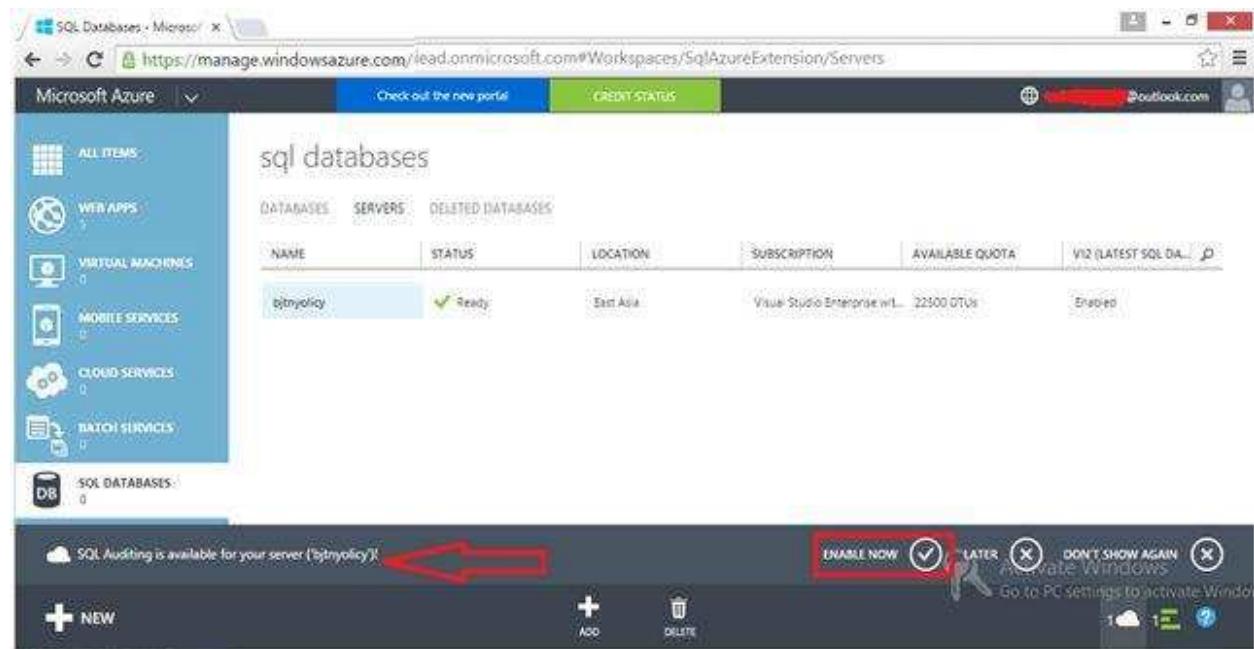
You can see it is going to create a new server.



The screenshot shows the Microsoft Azure portal interface for managing SQL databases. On the left sidebar, under the 'DB' category, the 'SQL DATABASES' option is selected. In the main content area, there is a table titled 'sql databases' with columns: NAME, STATUS, LOCATION, SUBSCRIPTION, AVAILABLE QUOTA, and V12 (LATEST SQL DA...). Several existing databases are listed, and a new row at the bottom is highlighted with a red border, showing the status as 'Creating'. A large red arrow points from the text above to this 'Creating' row. At the bottom of the page, a message bar says 'Create new SQL database server' and contains a 'NEW' button, a plus sign icon, and a delete icon.

When it is going to create it will ask to enable SQL auditing firewall settings for the current server.

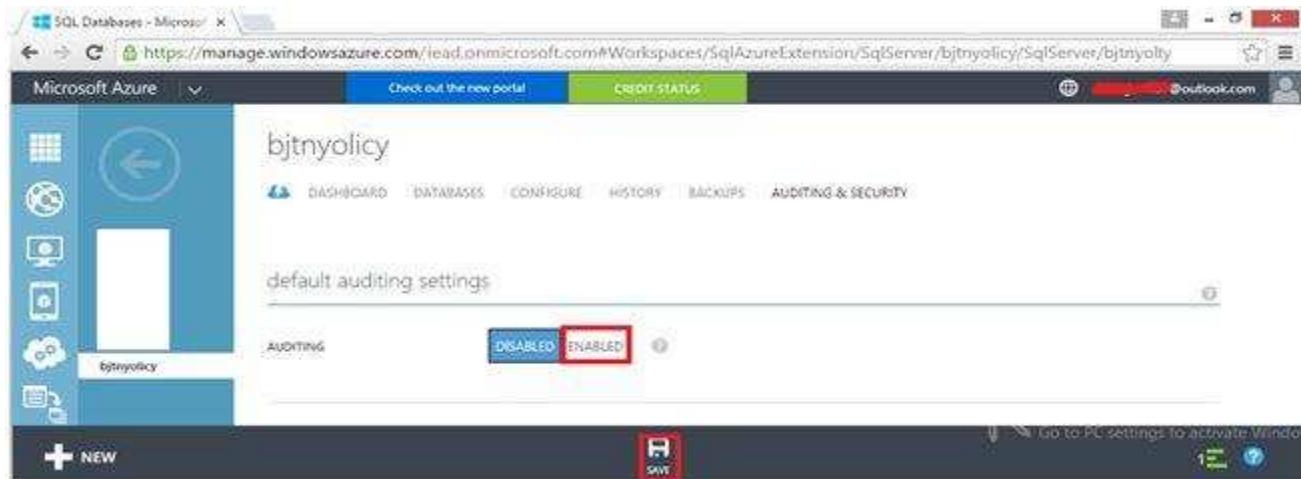
Click on “**Enable Now**” to enable the settings. (In rectangle, check mark)



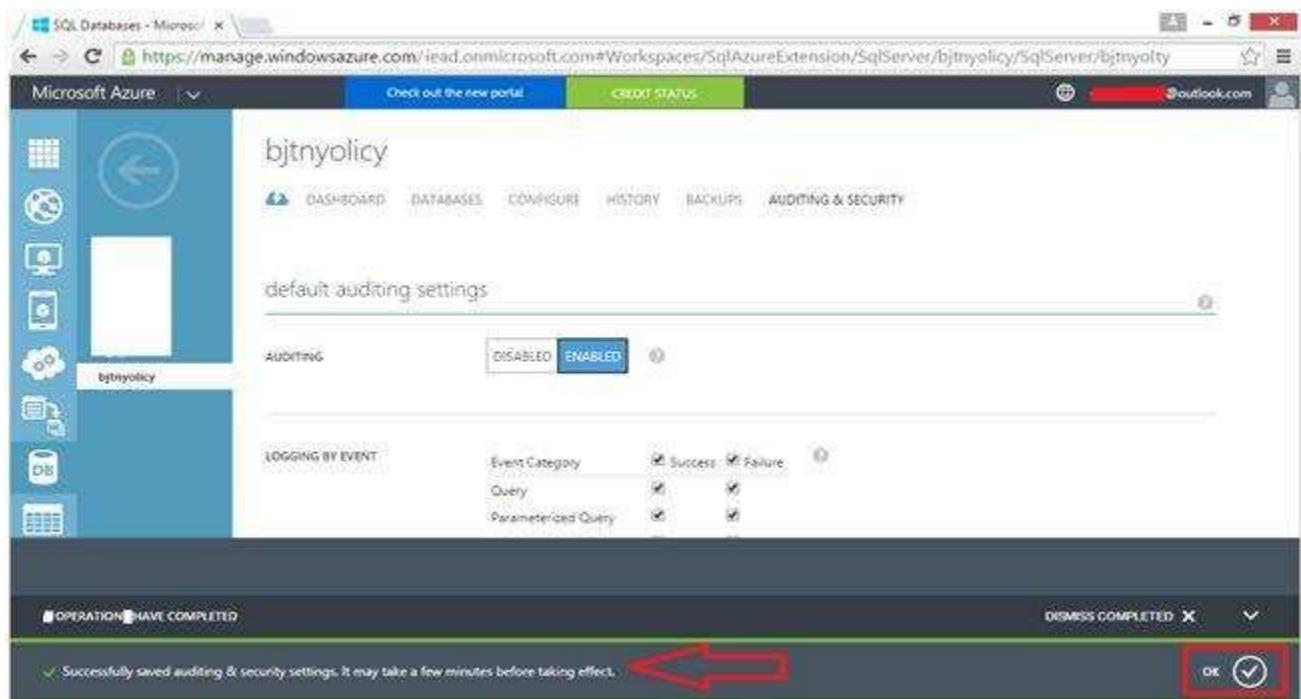
This screenshot shows the same Microsoft Azure portal interface as the previous one, but now it displays a message at the bottom: 'SQL Auditing is available for your server ('bjmyolicy').' A red arrow points from the text above to this message. To the right of the message are three buttons: 'ENABLE NOW' (with a checked checkbox), 'LATER' (with an unchecked checkbox), and 'DON'T SHOW AGAIN' (with an unchecked checkbox). Below these buttons is another message: 'Go to PC settings to activate Windows'.

In the next page, I need to enable default auditing settings, like storage account, table name, and storage access key.

Click on “Save” button to save the current settings.



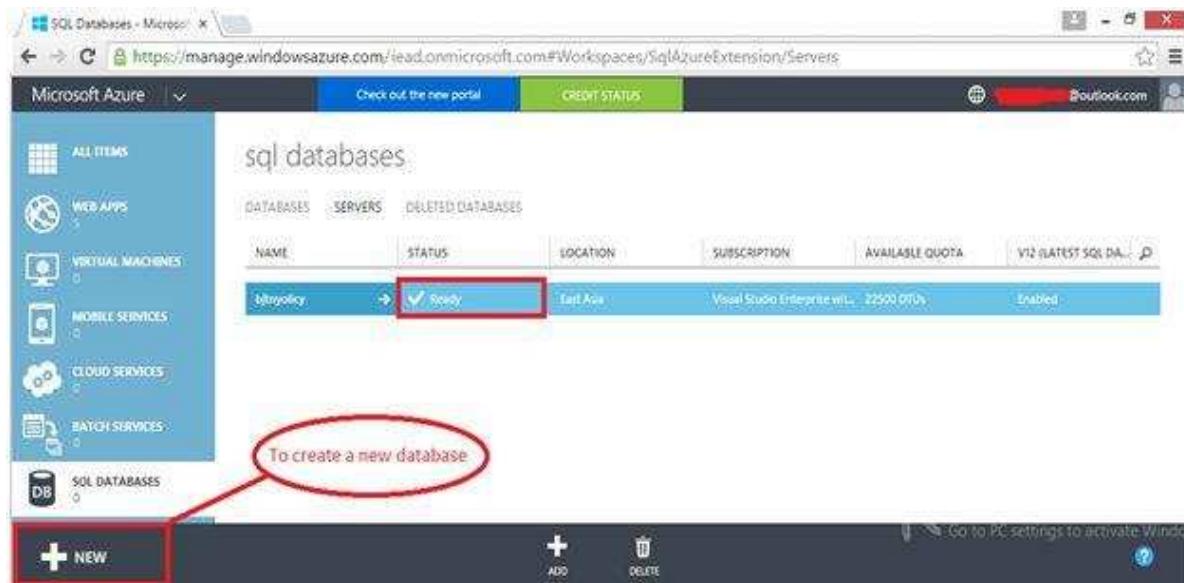
After clicking on save button, it will give message for saving stream.



So, our server has been created successfully, and is ready for use.

Now, I need to create a database, go back to the servers and select current server.

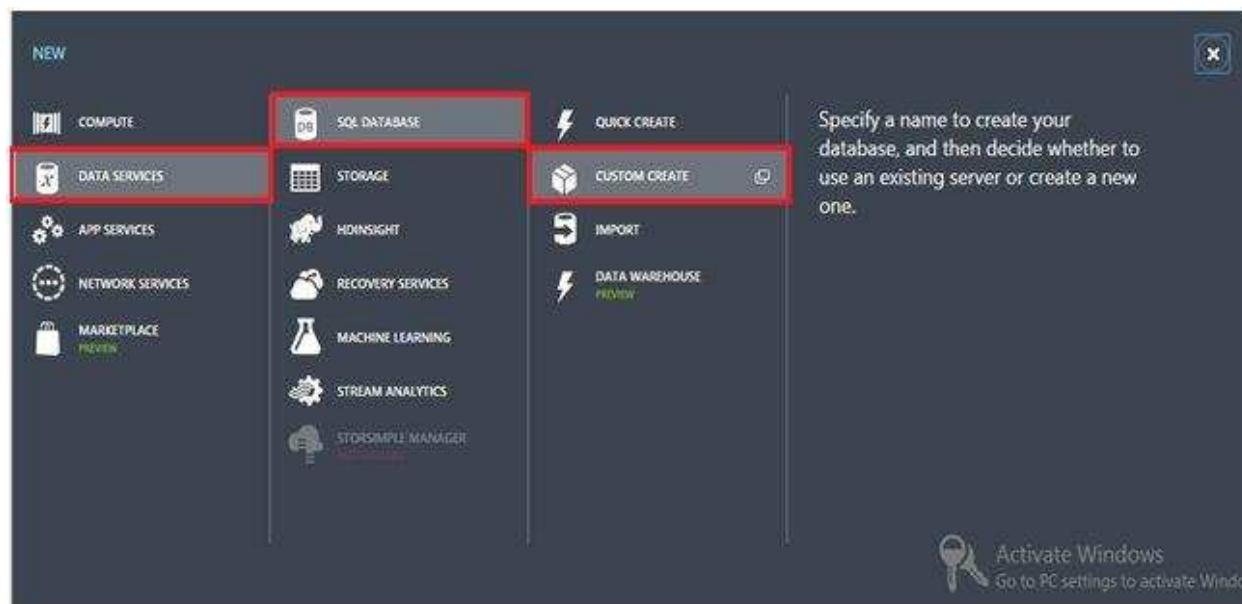
Click on “New” button to create a new database.



The screenshot shows the Microsoft Azure portal's SQL Databases page. On the left, there's a sidebar with icons for All Items, Web Apps, Virtual Machines, Mobile Services, Cloud Services, Batch Services, and SQL Databases. The SQL Databases icon is highlighted with a red box. Below the sidebar, a large red box surrounds the 'NEW' button. A red callout bubble with the text 'To create a new database' points to the 'SQL DATABASES' section. The main table lists databases with columns: NAME, STATUS, LOCATION, SUBSCRIPTION, AVAILABLE QUOTA, and V12 LATEST SQL DA... . The 'master' database is listed with a status of 'Ready'. The 'Status' column for the 'master' database is also highlighted with a red box.

Go to Data Services, SQL Database, then Custom Create.

I going to choose “**Custom Create**”, where we can decide whether to use an existing server or create a new one.



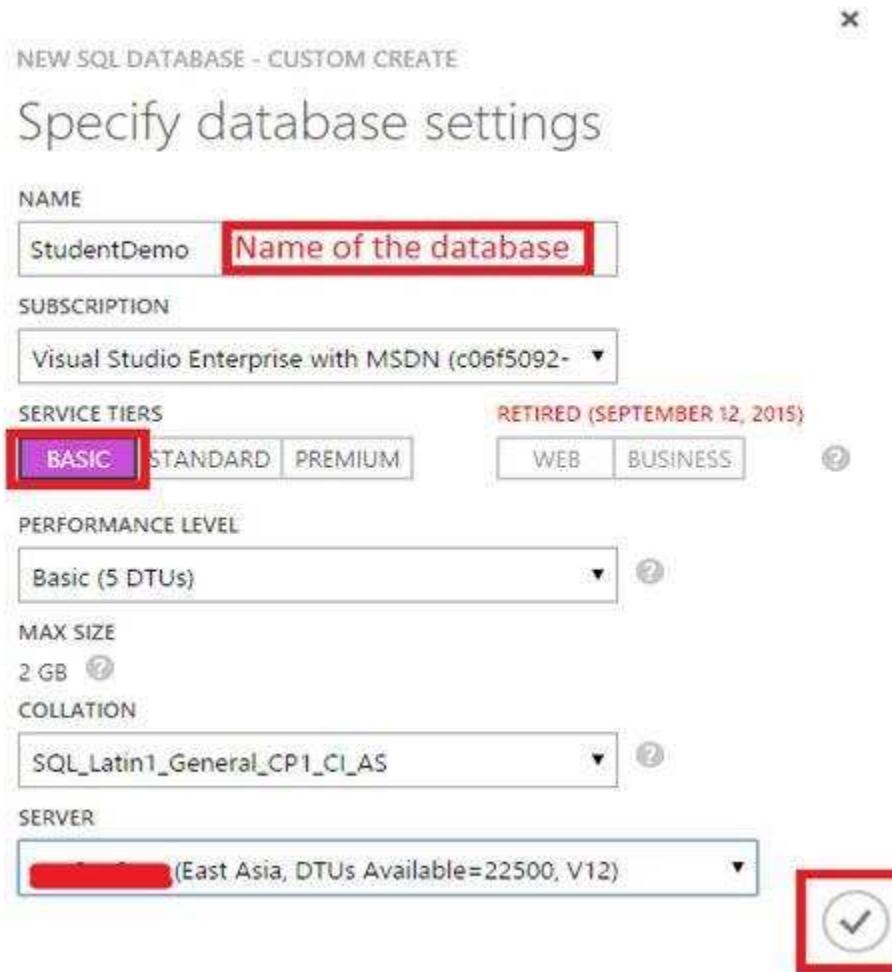
The screenshot shows the Microsoft Azure 'NEW' blade. On the left, there's a sidebar with icons for Compute, Data Services, App Services, Network Services, Marketplace, Storage, HDInsight, Recovery Services, Machine Learning, Stream Analytics, and StorSimple Manager. The 'DATA SERVICES' icon is highlighted with a red box. The 'SQL DATABASE' icon is also highlighted with a red box. To the right, there are two main sections: 'QUICK CREATE' and 'CUSTOM CREATE'. The 'CUSTOM CREATE' section is highlighted with a red box and contains the text: 'Specify a name to create your database, and then decide whether to use an existing server or create a new one.' At the bottom right, there's an 'Activate Windows' link.

Now, fill all the details for creation of the database.

Give the name to database, which you want to create; I have given “**StudentDemo**”, select “**Service Tier**” and then the server name.

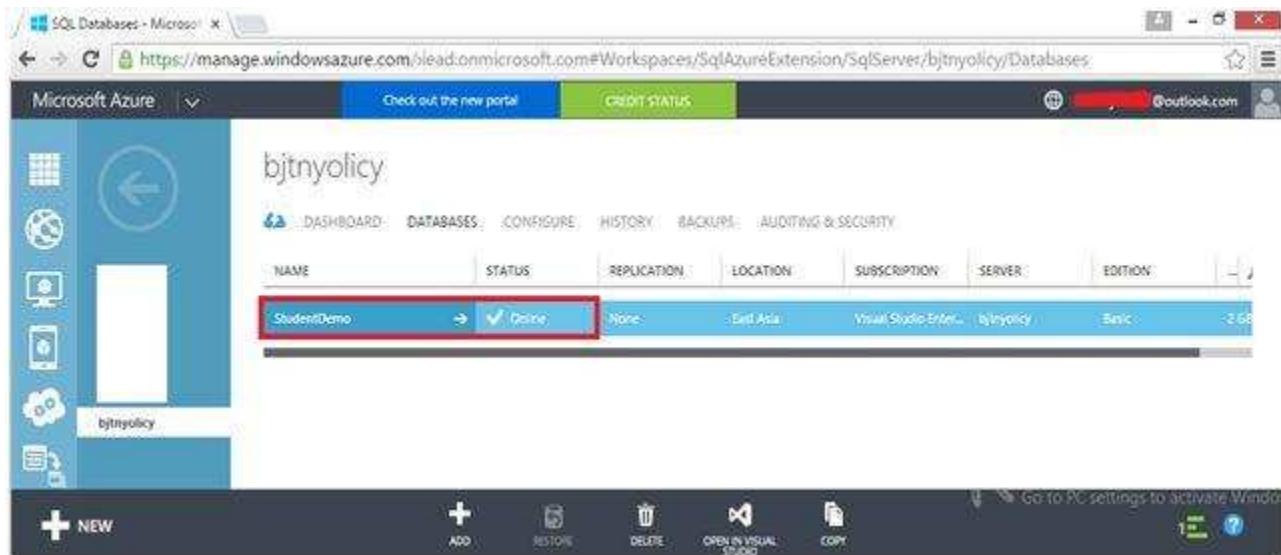
Click on check button to create database.

Note: If you are going to create a server and want to give the name to server like “*Nitin*”, so it will create a default name to server, like “*Nitin*” to “*plq4stu4ut*”.



You can see, the creation of database is on processing.

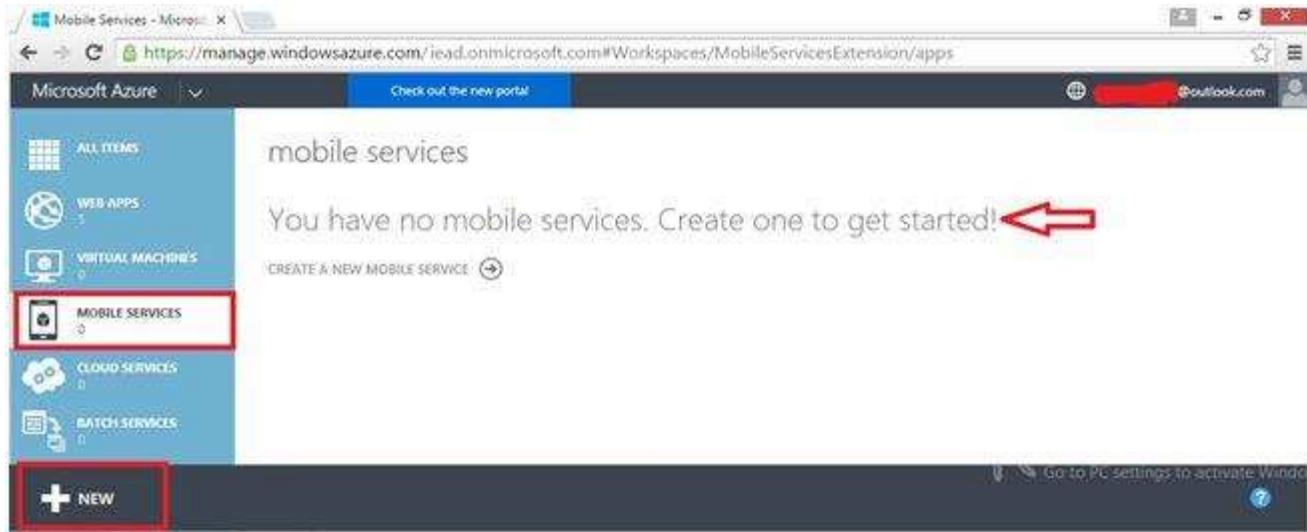
Now, I have successfully created database “**StudentDemo**” and named with server “*Nitin*”.



The screenshot shows the Microsoft Azure SQL Databases management interface. The left sidebar lists databases under the workspace 'bjtnyolc'. The main area displays the 'bjtnyolc' database with a table row for 'StudentDemo'. The table columns include NAME, STATUS, REPLICATION, LOCATION, SUBSCRIPTION, SERVER, and EDITION. The 'NAME' column shows 'StudentDemo' with a red border, and the 'STATUS' column shows 'Define' with a checkmark. The 'LOCATION' is listed as 'East Asia', 'SUBSCRIPTION' as 'Visual Studio Enterprise', 'SERVER' as 'bjtnyolc', and 'EDITION' as 'Basic'. Below the table are buttons for ADD, RESTORE, DELETE, OPEN IN VISUAL STUDIO, and COPY.

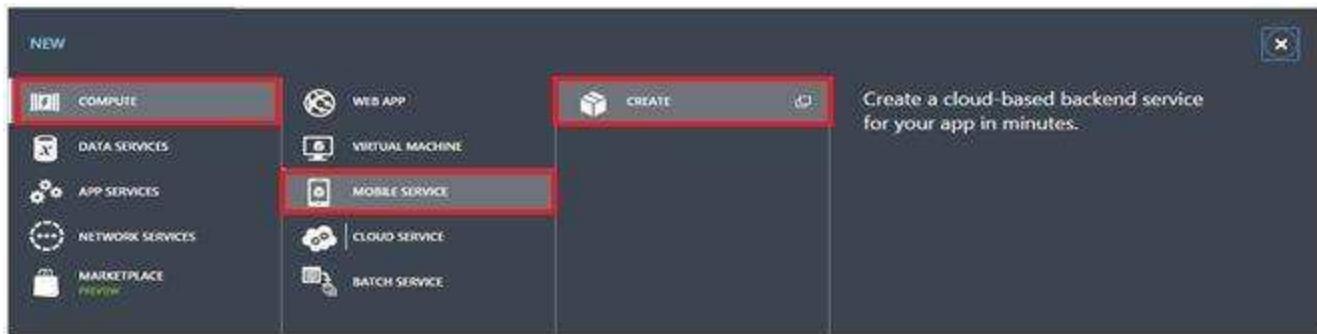
Now our, Server and Database is ready for use. Now, I'm going to create a Mobile Service.

Click on “**Mobile Services**”, we still don’t have any Mobile Services. Click on to “**New**” for creating a Mobile Service.



The screenshot shows the Microsoft Azure Mobile Services management interface. The left sidebar lists services: ALL ITEMS, WEB APPS, VIRTUAL MACHINES, MOBILE SERVICES (which is highlighted with a red box), CLOUD SERVICES, and BATCH SERVICES. At the bottom of the sidebar is a '+ NEW' button. The main area displays the message "mobile services" and "You have no mobile services. Create one to get started!" with a red arrow pointing to the text. Below this message is a 'CREATE A NEW MOBILE SERVICE' button.

Click on to **COMPUTE, MOBILE SERVICE**, and then **CREATE**.



Give the Name for your Mobile Service and for the database. Here, I'm going to select existing database, which I just created.

NEW MOBILE SERVICE

Create a Mobile Service

URL

mobileserviceofnitin  Name for Mobile Service

.azure-mobile.net

DATABASE

Use an existing SQL database  Existing DB

Use an existing SQL database 

Create a new SQL database instance

West US

BACKEND

JavaScript

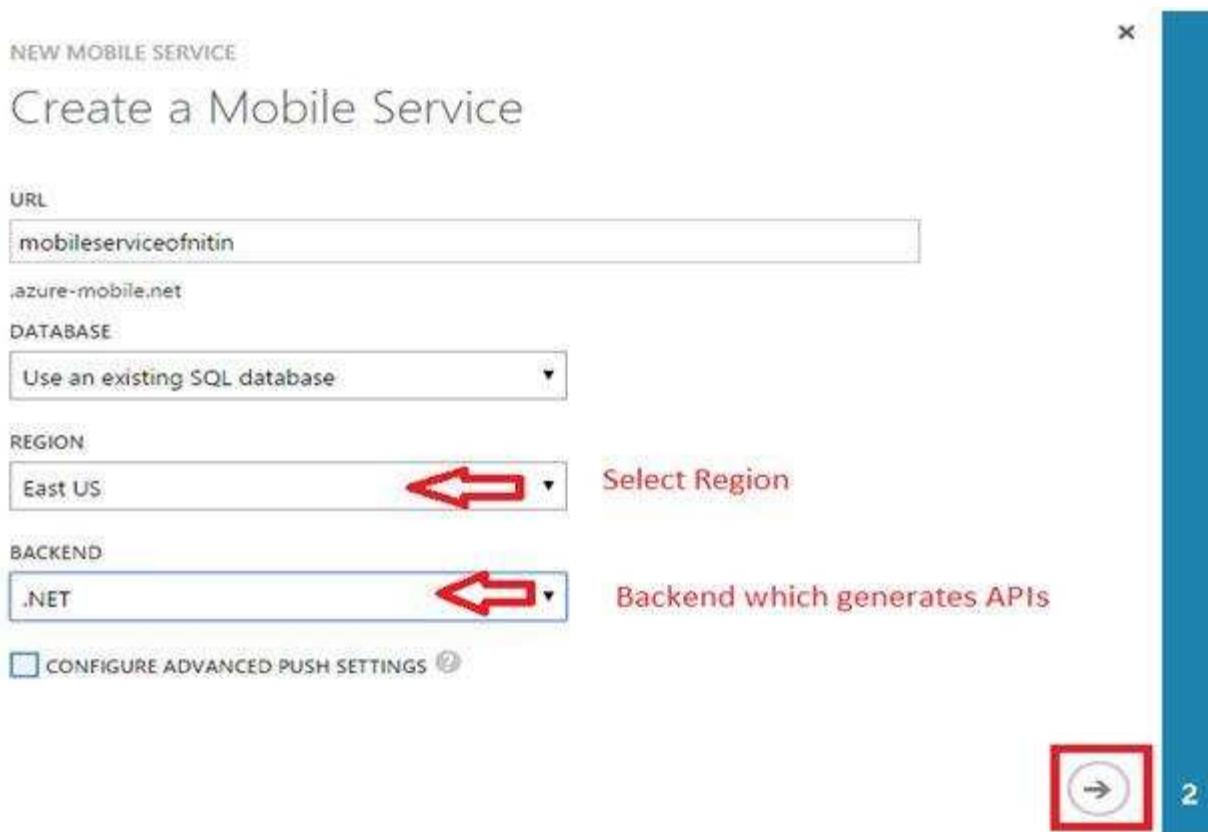
CONFIGURE ADVANCED PUSH SETTINGS 

2

Select your “Region”, where you want to create your Service.

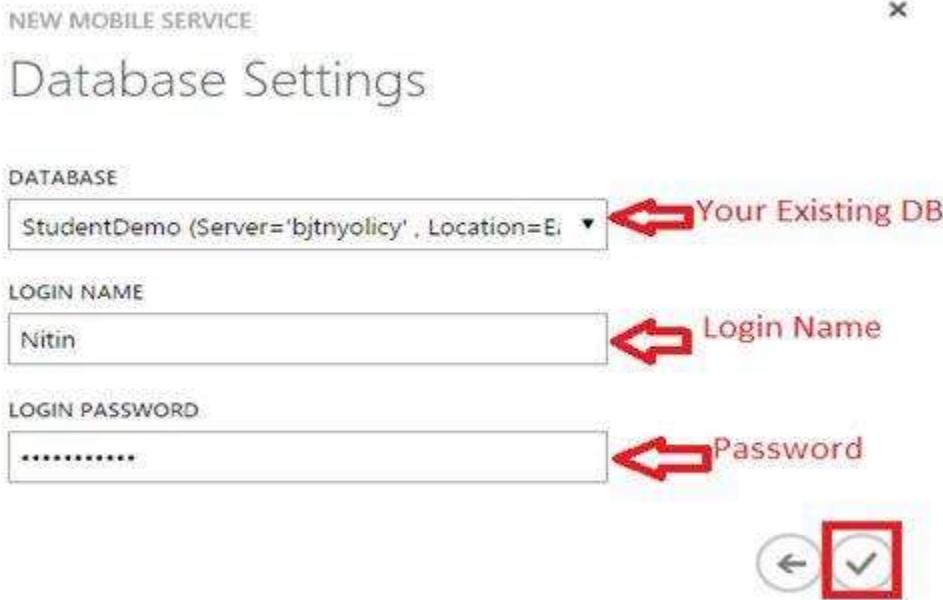
Backend provides two options for the use, .NET and JavaScript; these options generate APIs for performing operations on the backend table data.

Click on next arrow.



Specify the **Database Settings**; select your database, Login Name, and Password.

Click on to check button.



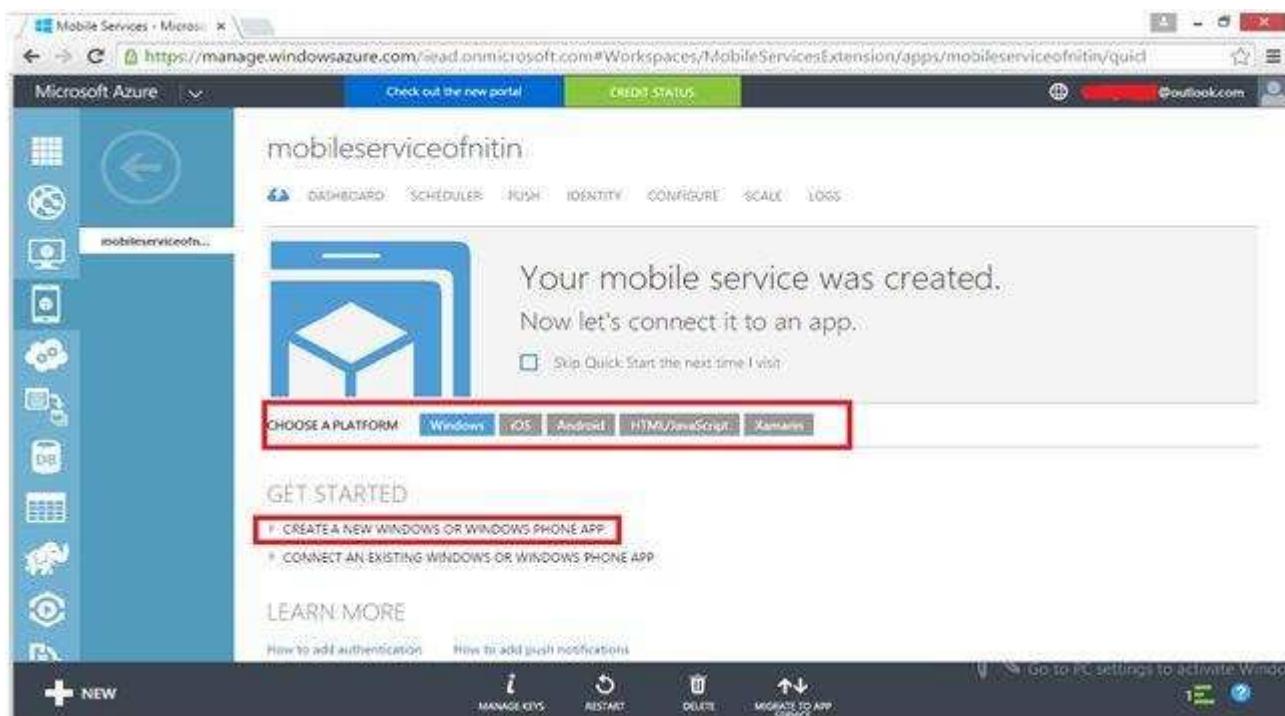
After clicking on check, you will find Mobile Service will be created as in the following screenshot and is ready for use.

NAME	STATUS	SUBSCRIPTION	BACKEND	LOCATION	URL
mobileserviceonlin...	Ready	Visual Studio Enterprise with...	.NET	East Asia	http://mobileserviceonlin...

Click on your Mobile Service, you will find more things for the service.

This page provides information about the platform for consuming the Mobile service. We have options like iOS, Windows, Android, HTML/JavaScript, etc. The portal provides a ready App for guiding us through the use of Mobile Services. On the top of the page, we have options to create Logs, Identity, Scheduler, PUSH configuration, the Host configuration, etc.

So, under the “Choose A Platform”, expand the “Create A New Windows or Windows Phone App”.



This displays the three easy steps to create a Windows Store app connected to your mobile service.

1. Install Visual Studio Professional 2013, if you don't have followed the first point and install on your local computer.
2. Select the language for your App (I have selected C#).

3. Click on “Download”.

CREATE A NEW WINDOWS OR WINDOWS PHONE APP

Follow these steps to create a simple Todo app that connects to your mobile service.

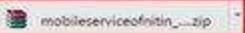
- 1 Get the tools**
Install Visual Studio Professional 2013
(If you are using Windows 8 or Windows Phone 8, follow this tutorial this tutorial)
- 2 Download and run your app and service locally**
Download your personalized starter project, extract it, and then open it in Visual Studio.
LANGUAGE: C# [Download](#)
- 3 Publish your service to the cloud**
Right-click your mobile service project and select Publish.
In App.xaml.cs, locate the MobileServiceClient definition, uncomment the constructor containing the URL to the published mobile service, and comment-out the one containing the local URL. Press F5 to run your client project against your service in the cloud.

After clicking on to “Download”, it will download a project, click on to open it.

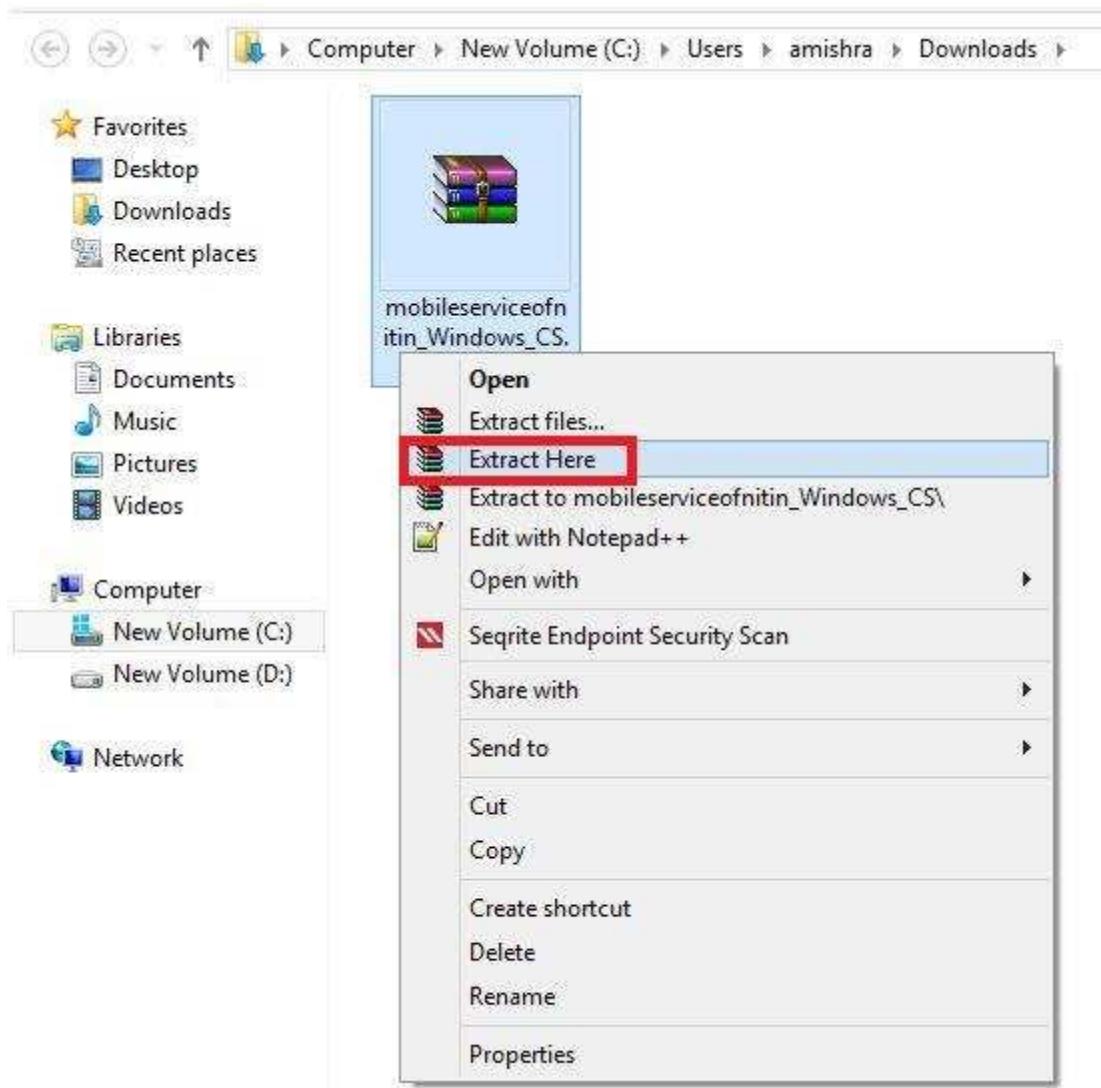
CREATE A NEW WINDOWS OR WINDOWS PHONE APP

Follow these steps to create a simple Todo app that connects to your mobile service.

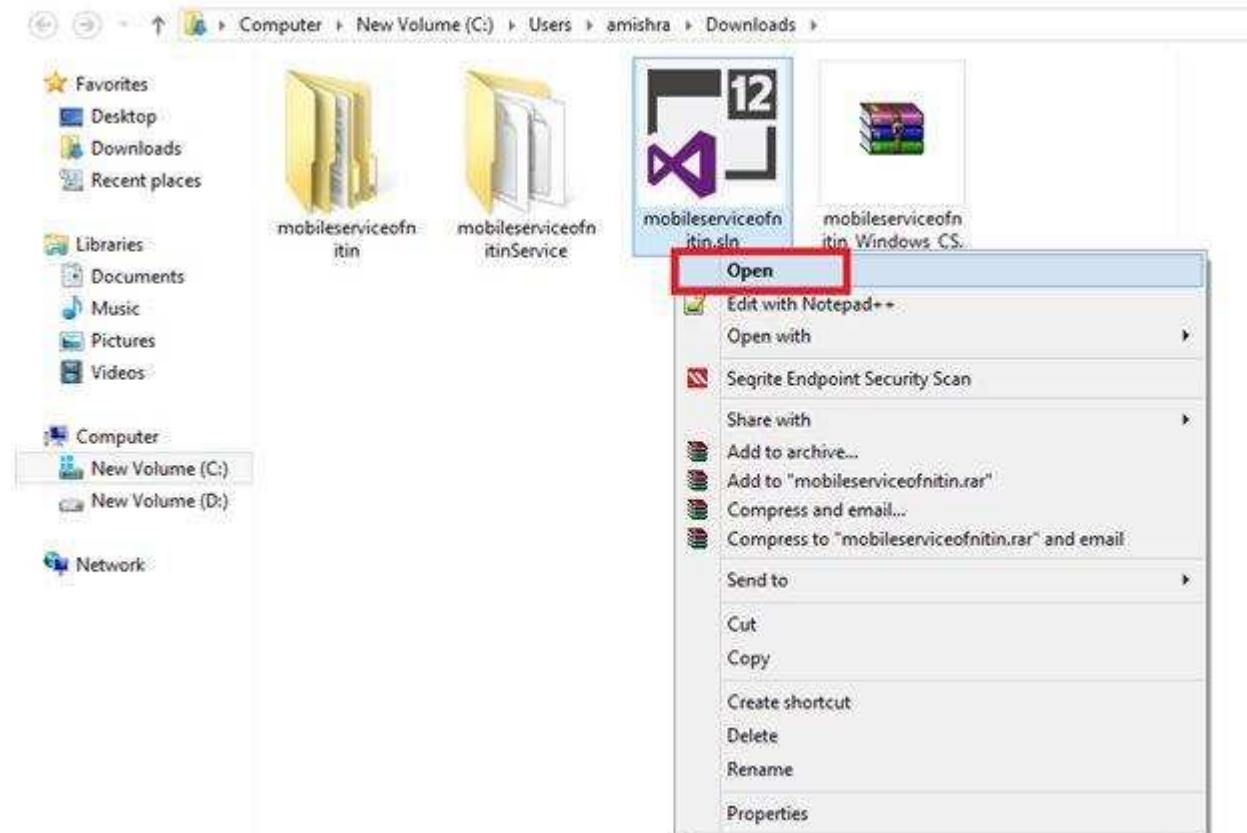
- 1 Get the tools**
Install Visual Studio Professional 2013
(If you are using Windows 8 or Windows Phone 8, follow this tutorial this tutorial)
- 2 Download and run your app and service locally**
Download your personalized starter project, extract it, and then open it in Visual Studio.
LANGUAGE: C# [Download](#)
- 3 Publish your service to the cloud**
Right-click your mobile service project and select Publish.

 [Show all downloads...](#)

Browse to the location where you saved the compressed project files, expand the files on your computer, and open the solution file in Visual Studio.

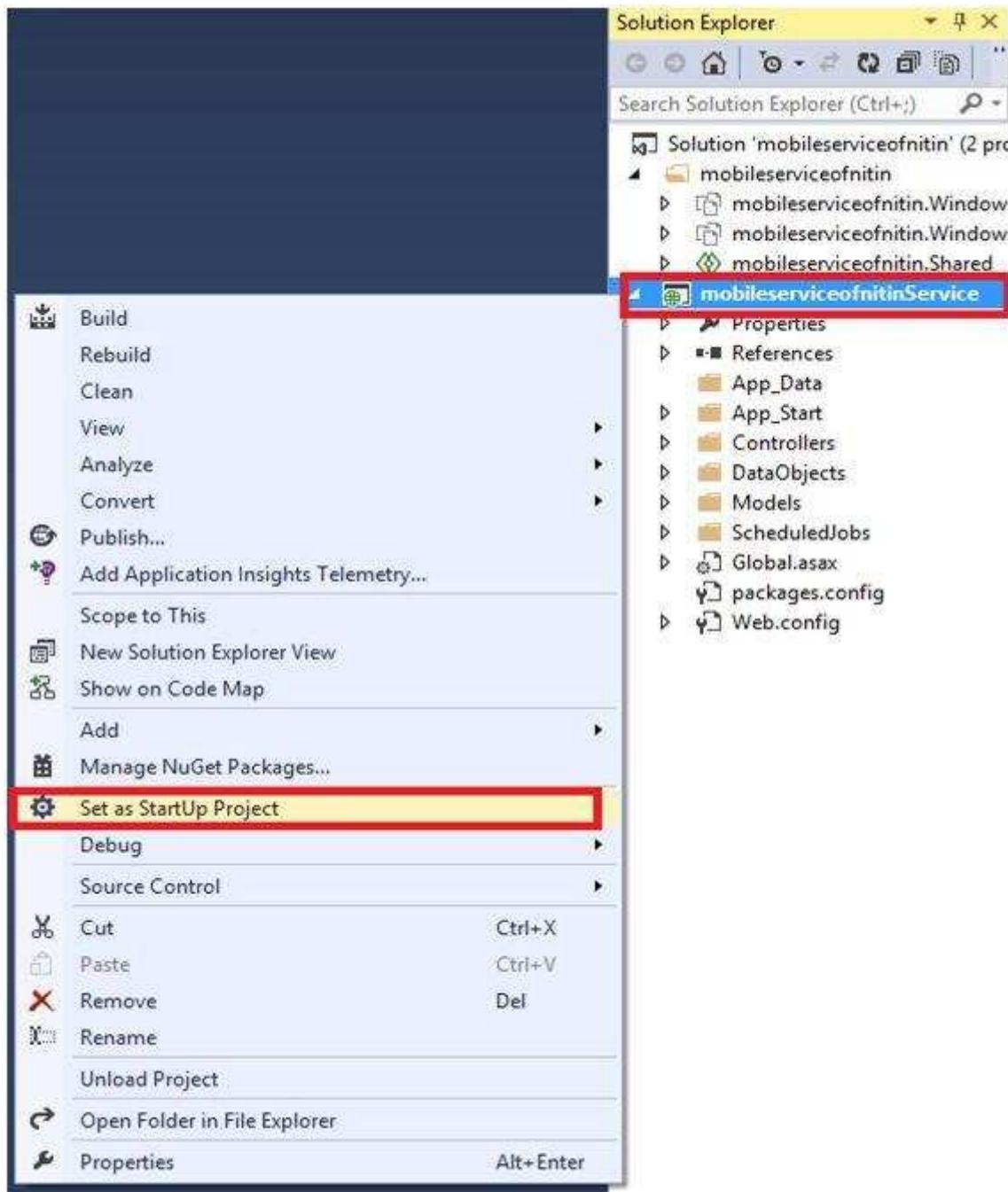


Open the Visual Studio file.

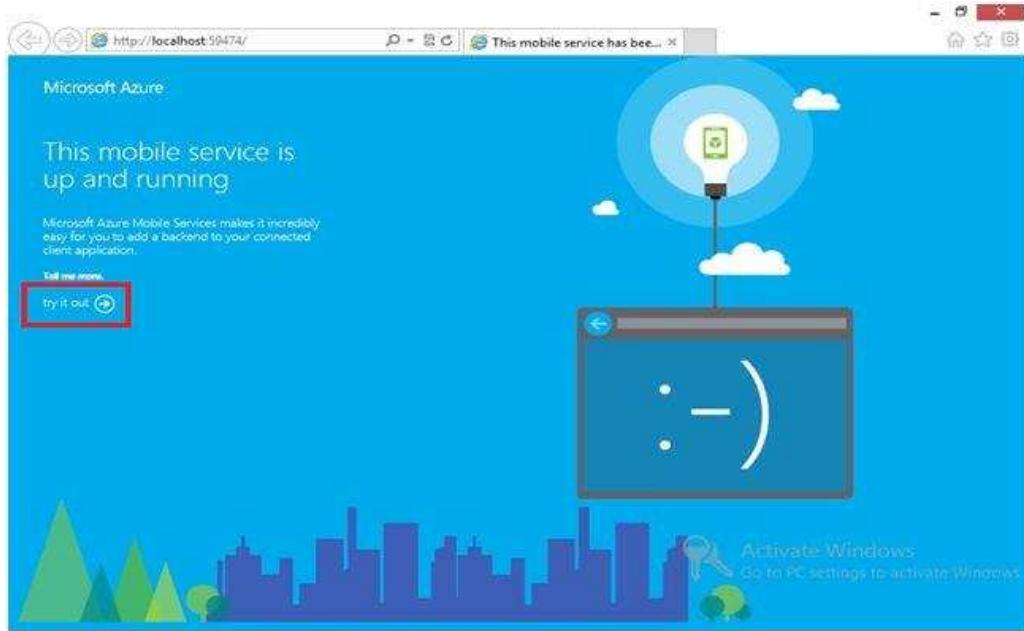


Right click on to your service project and set it “**Set as Startup Project**”.

Save your project and press F5 to run your project.

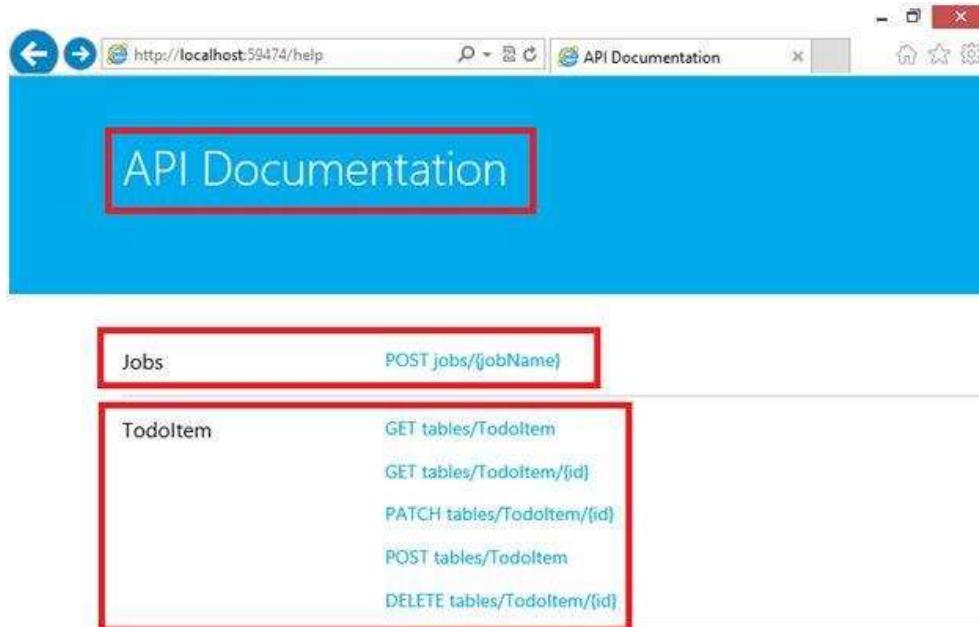


I've successfully run my application; follow the rectangle and click over it.



You can see, there is complete API Documentation of the Mobile Service.

You can perform any operation on the browser and after you can check Cloud where Mobile Service reside.



The screenshot shows the API Documentation page for a mobile service. The main title 'API Documentation' is highlighted with a red box. Below it, two sections are listed:

Jobs	POST jobs/{jobName}
TodoItem	GET tables/TodoItem GET tables/TodoItem/{id} PATCH tables/TodoItem/{id} POST tables/TodoItem DELETE tables/TodoItem/{id}

How to Create a Mobile App in Microsoft Azure

Mobile Apps in Microsoft Azure offer many available mobile application development platforms.

Build native and cross platform apps:

We can create native IOS, Android, and Windows apps or cross-platform Xamarin or Cordova (PhoneGap) apps, you can also consider many advantages of App Service using native SDKs.

Connect to enterprise systems: Mobile Apps you can connect to your enterprise cloud resources.

Push Notifications to billions in seconds

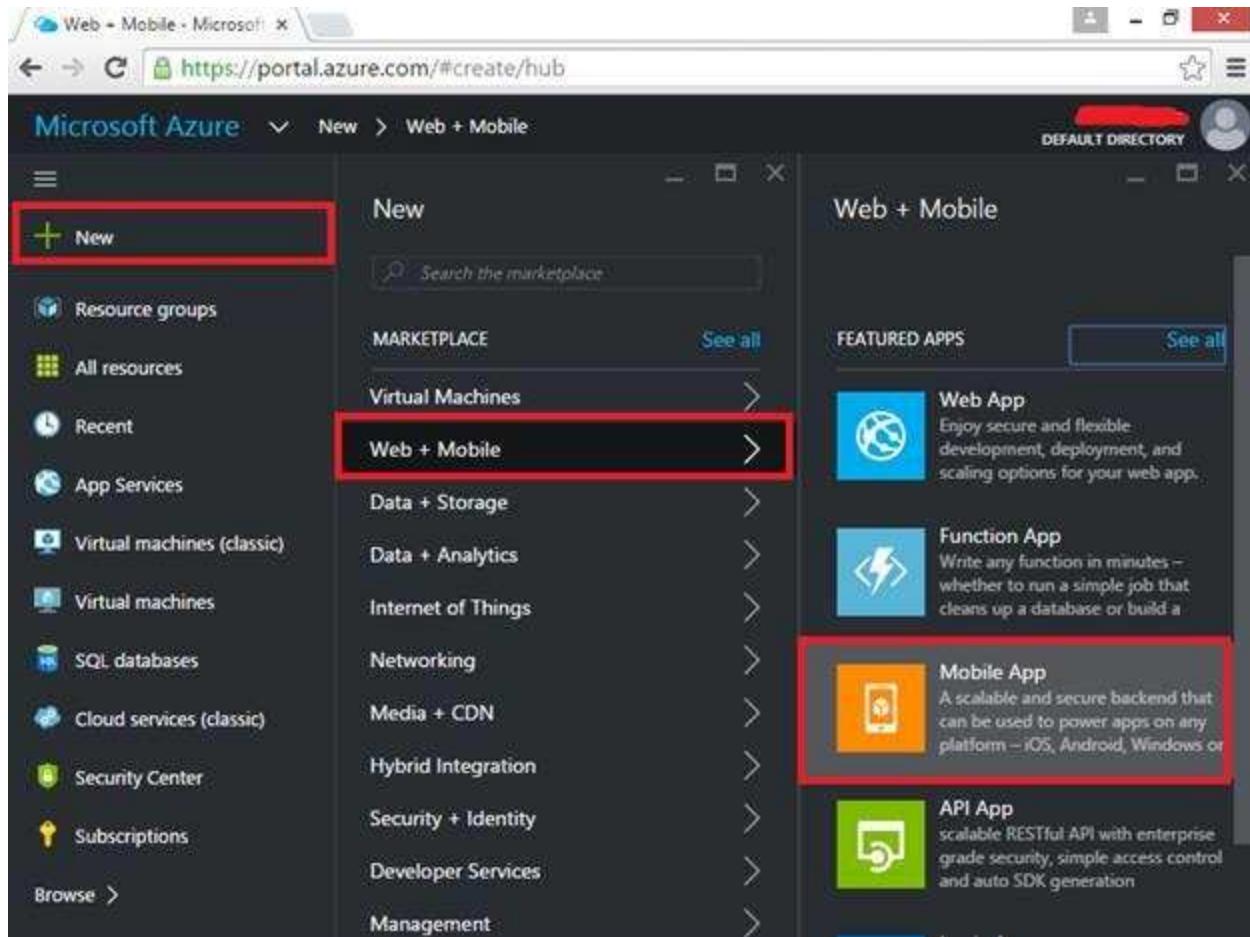
Mobile App Features:

The following features are important to cloud-enabled mobile development:

- Authentication and Authorization
- Integrate the SDK
- Data Access
- Offline synchronization
- Push Notifications

For creating an Android App you must have to login into the Azure Portal

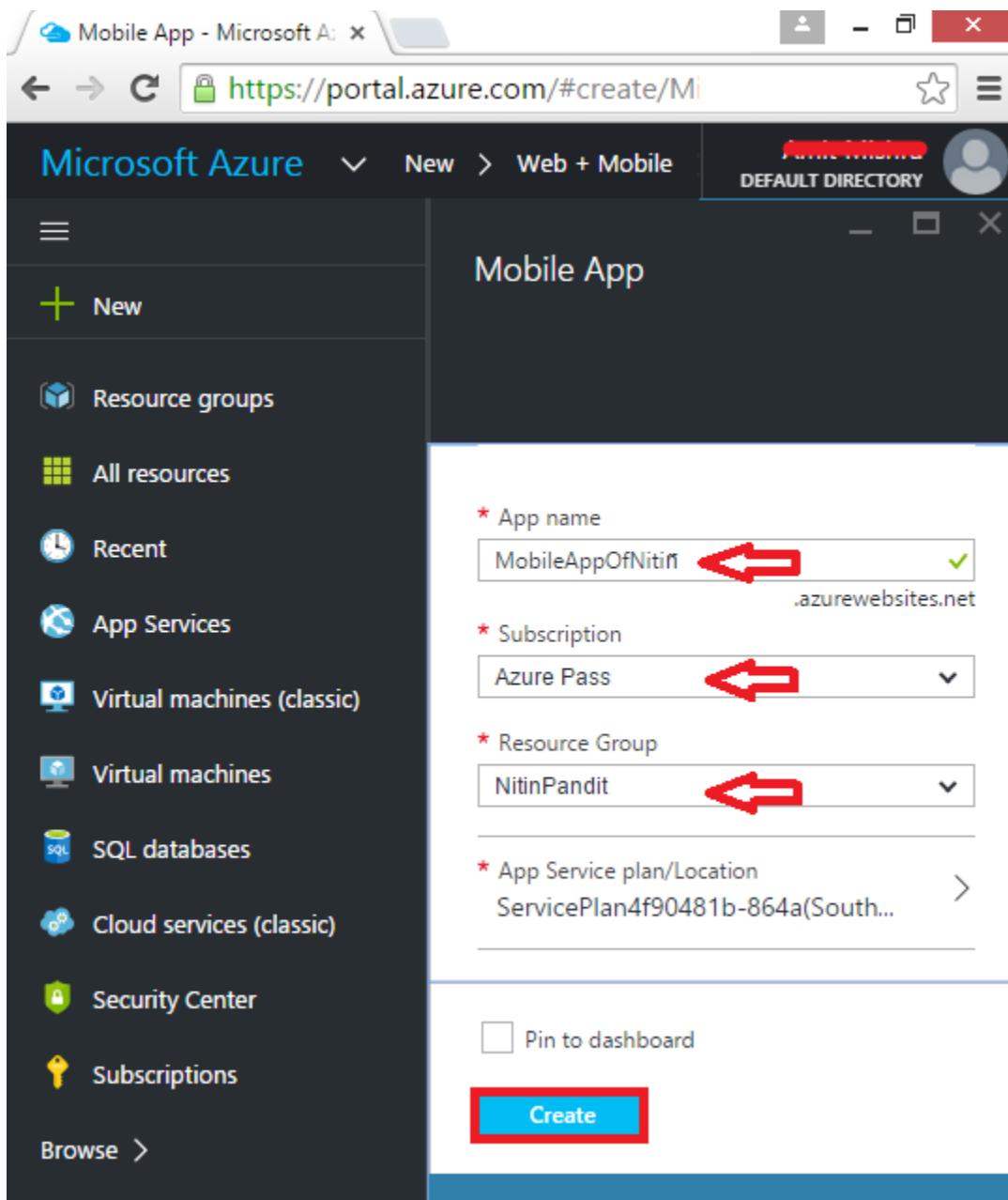
Click on to *New, Web + Mobile*, then *Mobile App*,



In the name box, give the app name whatever you want to create, and subscription would be default.

In the *Resource Group*, I have a resource group named *NitinPandit*, you can also create your by clicking on new resource group.

Now click on “Create” button.



Mobile App - Microsoft Azure

https://portal.azure.com/#create/Microsoft.MobileApp

Microsoft Azure

New Web + Mobile

DEFAULT DIRECTORY

Mobile App

+

New

Resource groups

All resources

Recent

App Services

Virtual machines (classic)

Virtual machines

SQL databases

Cloud services (classic)

Security Center

Subscriptions

Browse >

* App name
MobileAppOfNitin

* Subscription
Azure Pass

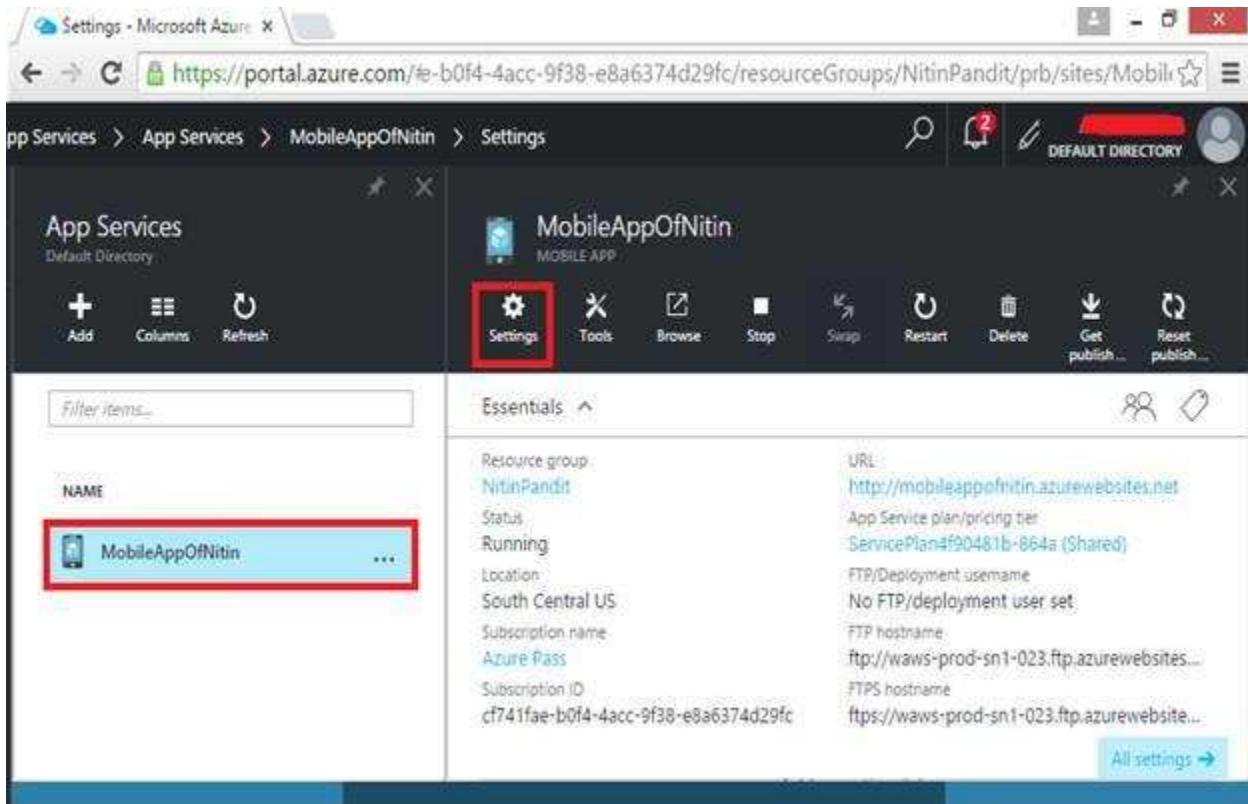
* Resource Group
NitinPandit

* App Service plan/Location
ServicePlan4f90481b-864a(South...)

Pin to dashboard

Create

After clicking on the button, your app will be created, click on app and go into the settings.

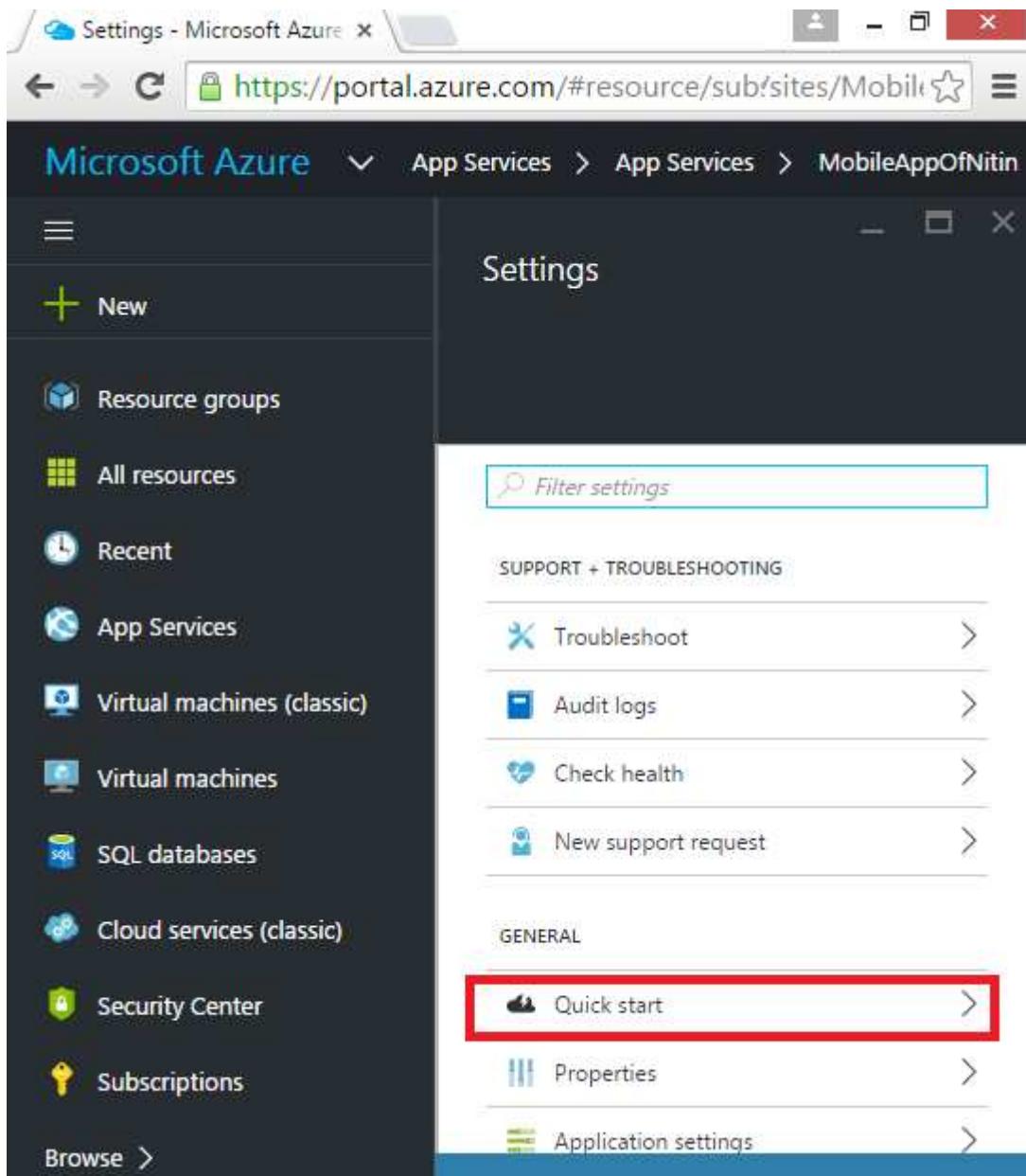


The screenshot shows the Microsoft Azure portal interface for App Services. The URL in the address bar is <https://portal.azure.com/#/resourceGroups/NitinPandit/prb/sites/MobileAppOfNitin/settings>. The left sidebar shows 'App Services' and the selected 'MobileAppOfNitin'. The main area has a title 'MobileAppOfNitin MOBILE APP'. Below it are several action buttons: 'Settings' (highlighted with a red box), 'Tools', 'Browse', 'Stop', 'Swap', 'Restart', 'Delete', 'Get publish...', and 'Reset publish...'. A 'Default Directory' dropdown is also present. On the left, there's a list of apps under 'NAME' with 'MobileAppOfNitin' selected and highlighted with a red box. The 'Essentials' tab is active, displaying various configuration details:

Setting	Value
Resource group	NitinPandit
Status	Running
Location	South Central US
Subscription name	Azure Pass
Subscription ID	cf741fae-b0f4-4acc-9f38-e8a6374d29fc
URL	http://mobileappofnitin.azurewebsites.net
App Service plan/pricing tier	ServicePlan4f90481b-864a (Shared)
FTP/Deployment username	No FTP/deployment user set
FTP hostname	ftp://waws-prod-sn1-023.ftp.azurewebsites.net
FTPS hostname	https://waws-prod-sn1-023.ftp.azurewebsite...

An 'All settings' button is located at the bottom right of the essentials section.

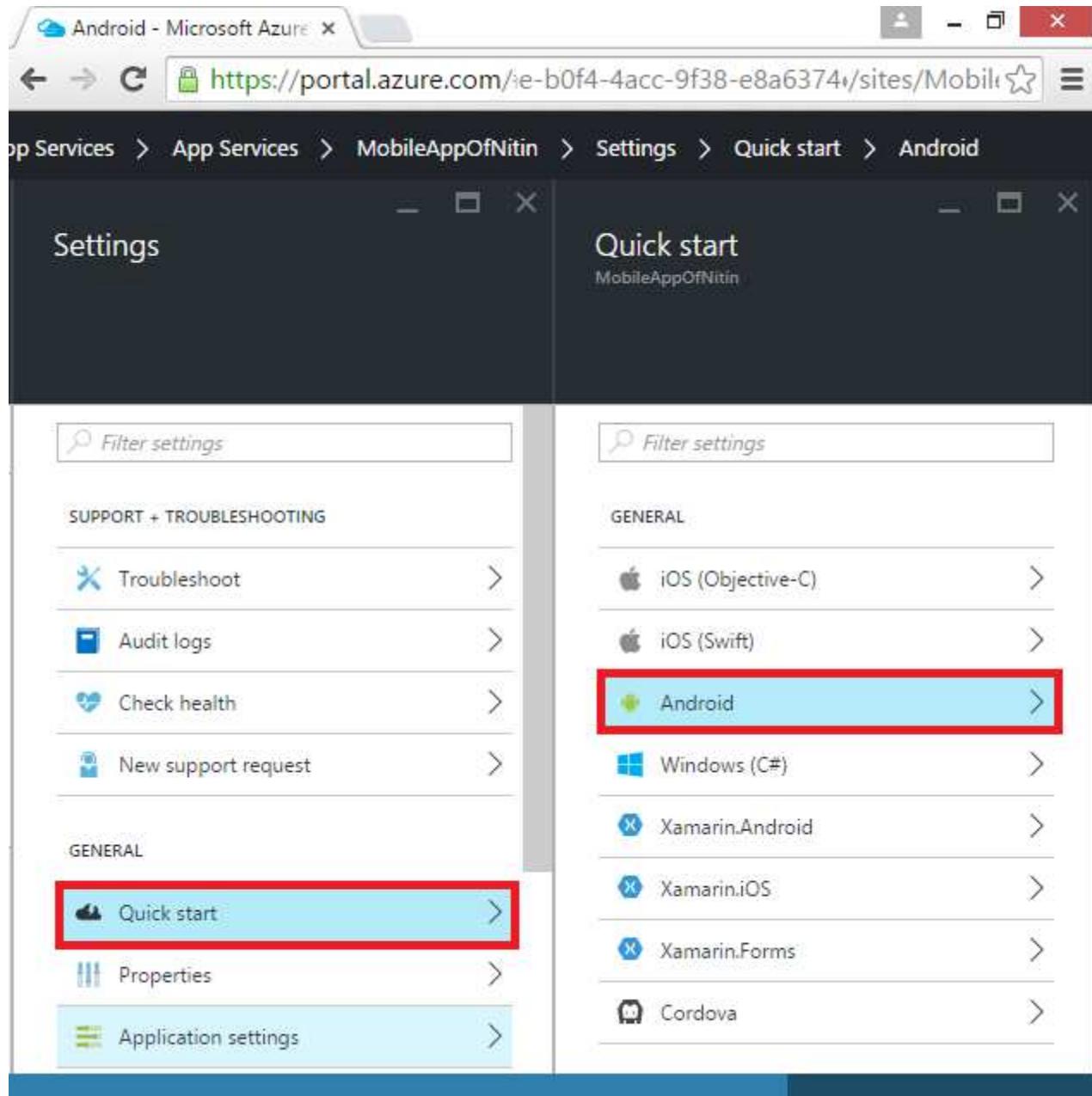
In the settings, click on to “*Quick start*” which is our client app platform,



The screenshot shows the Microsoft Azure portal interface. The URL in the browser is <https://portal.azure.com/#resource/sub/sites/MobileAppOfNitin>. The left sidebar lists various Azure services: Resource groups, All resources (selected), Recent, App Services (selected), Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Security Center (highlighted with a red box), Subscriptions, and Browse. The main content area is titled "Settings". It includes a "Filter settings" search bar and sections for "SUPPORT + TROUBLESHOOTING" (Troubleshoot, Audit logs, Check health, New support request) and "GENERAL" (Quick start, Properties, Application settings). The "Quick start" link under GENERAL is also highlighted with a red box.

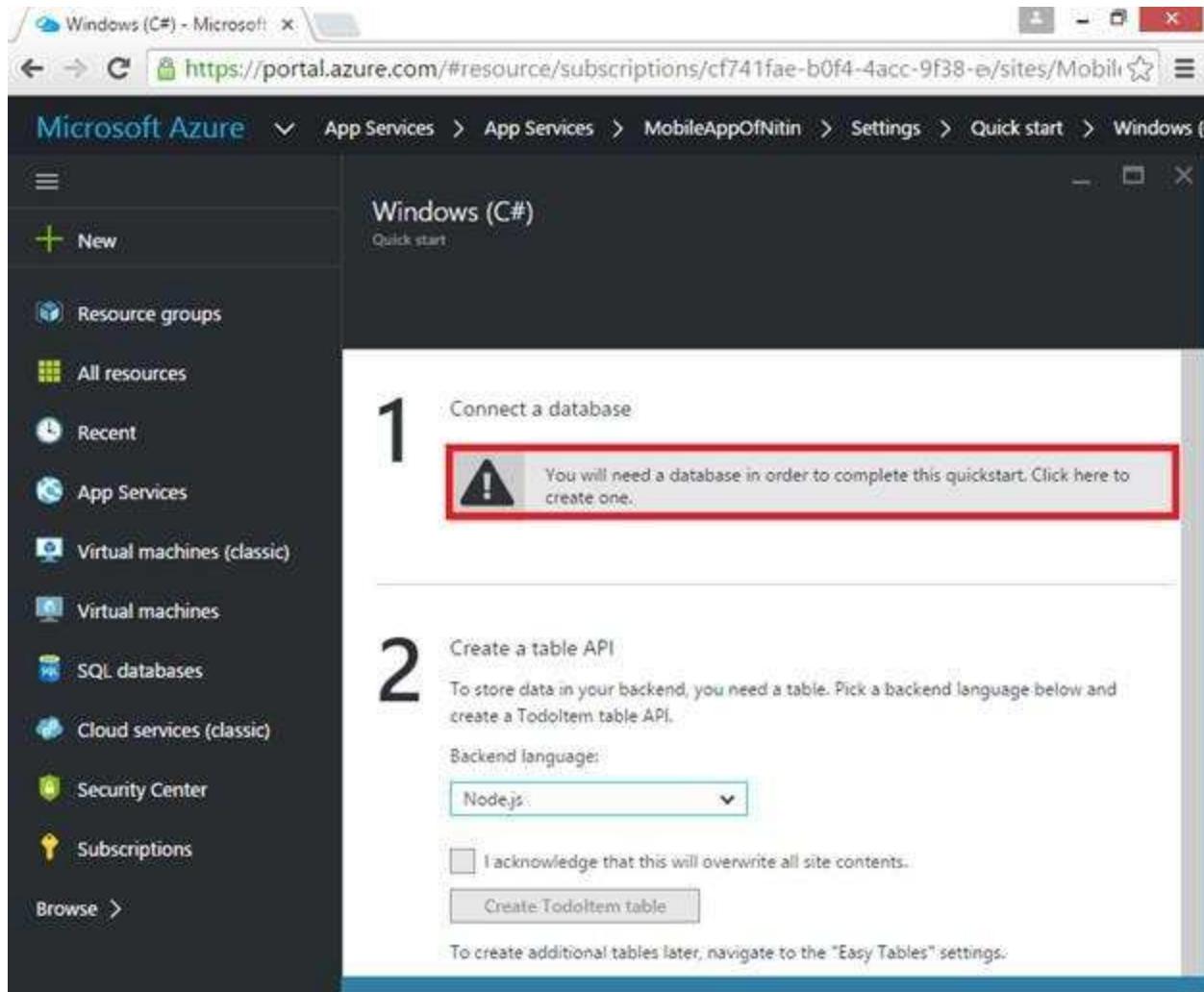
Now in the following screenshot, click on to any platform which one you want to create an application.

Here, I'm going to select “*Android*” because I’m creating an Android app.



The screenshot shows the Azure portal interface for managing an Android application. The left pane is titled 'Settings' and contains sections for 'SUPPORT + TROUBLESHOOTING' (Troubleshoot, Audit logs, Check health, New support request) and 'GENERAL' (Quick start, Properties, Application settings). The 'Quick start' item under 'GENERAL' is highlighted with a red box. The right pane is titled 'Quick start' and lists various platform options: iOS (Objective-C), iOS (Swift), Android (highlighted with a red box), Windows (C#), Xamarin.Android, Xamarin.iOS, Xamarin.Forms, and Cordova.

Here is the first step that connects to database, so click on to create or select the existing database which you have.



Windows (C#) - Microsoft X

https://portal.azure.com/#resource/subscriptions/cf741fae-b0f4-4acc-9f38-e/sites/MobileAppOfNitin

Microsoft Azure App Services > App Services > MobileAppOfNitin > Settings > Quick start > Windows (C#)

Windows (C#)
Quick start

1 Connect a database

2 Create a table API

You will need a database in order to complete this quickstart. Click here to create one.

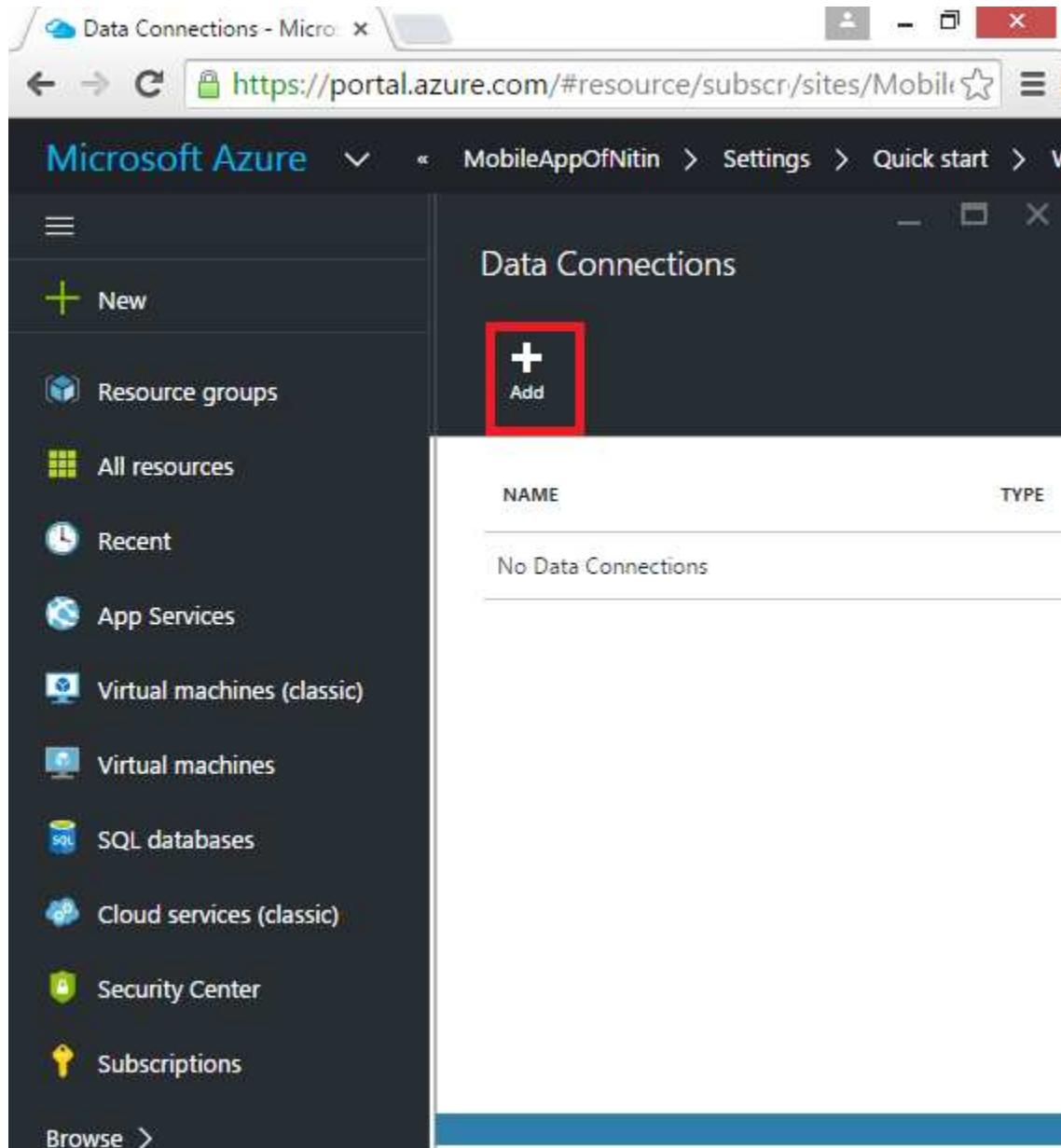
Backend language: Node.js

I acknowledge that this will overwrite all site contents.

Create TodoItem table

To create additional tables later, navigate to the "Easy Tables" settings.

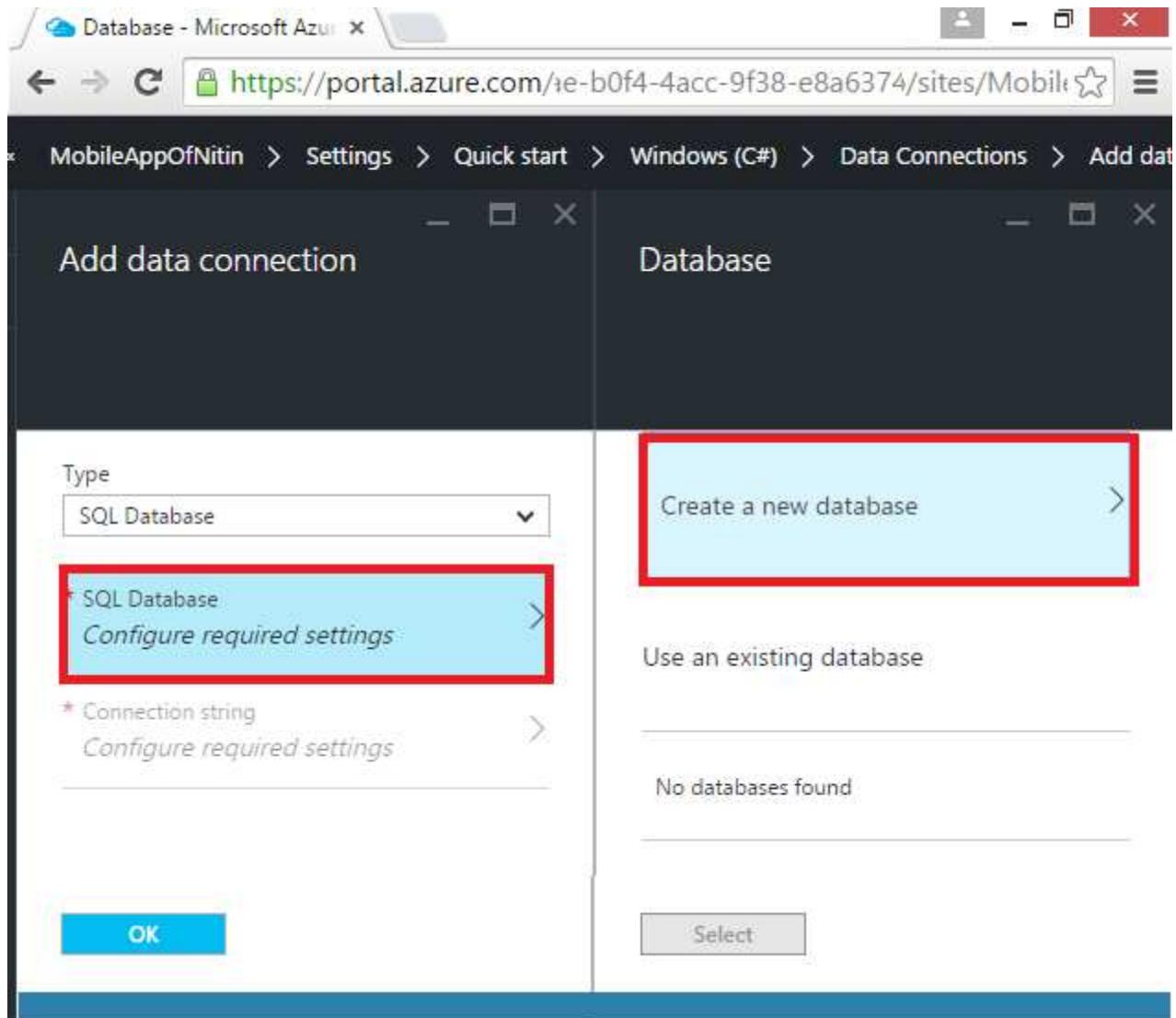
There are no data connections, Click on to “Add” button to add a database.



The screenshot shows the Microsoft Azure portal interface. The URL in the browser is <https://portal.azure.com/#resource/subscr/sites/MobileAppOfNitin>. The page title is "Data Connections - Micro". On the left sidebar, there is a list of services: Resource groups, All resources, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Security Center, and Subscriptions. Below the sidebar is a "Browse >" link. The main content area is titled "Data Connections" and contains a table with one row: "No Data Connections". A large red box highlights the "Add" button, which has a plus sign icon.

Choose SQL Database, and then click on to required settings.

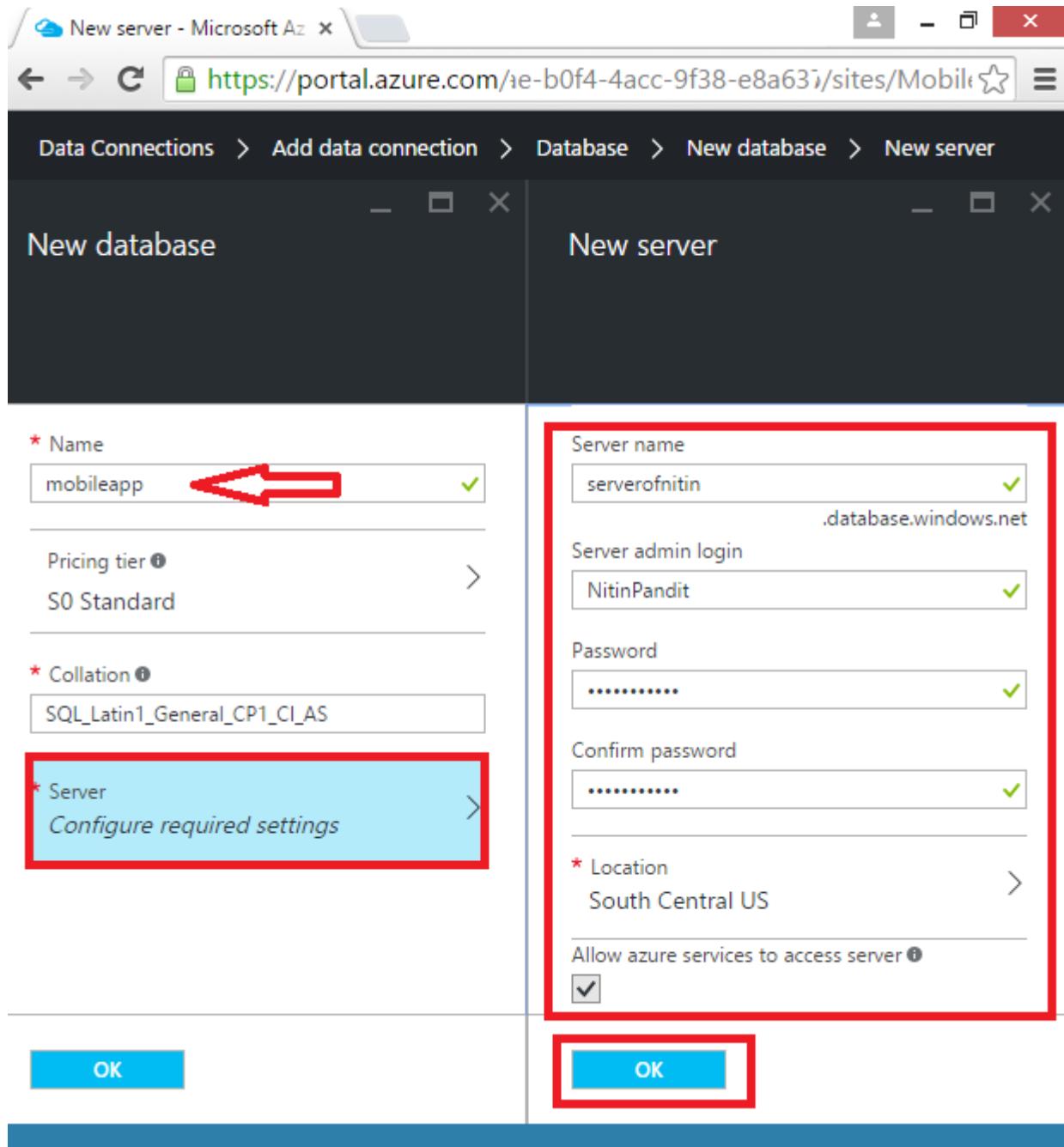
You will find a new database window, if you don't have any existing database click on to create a new database.



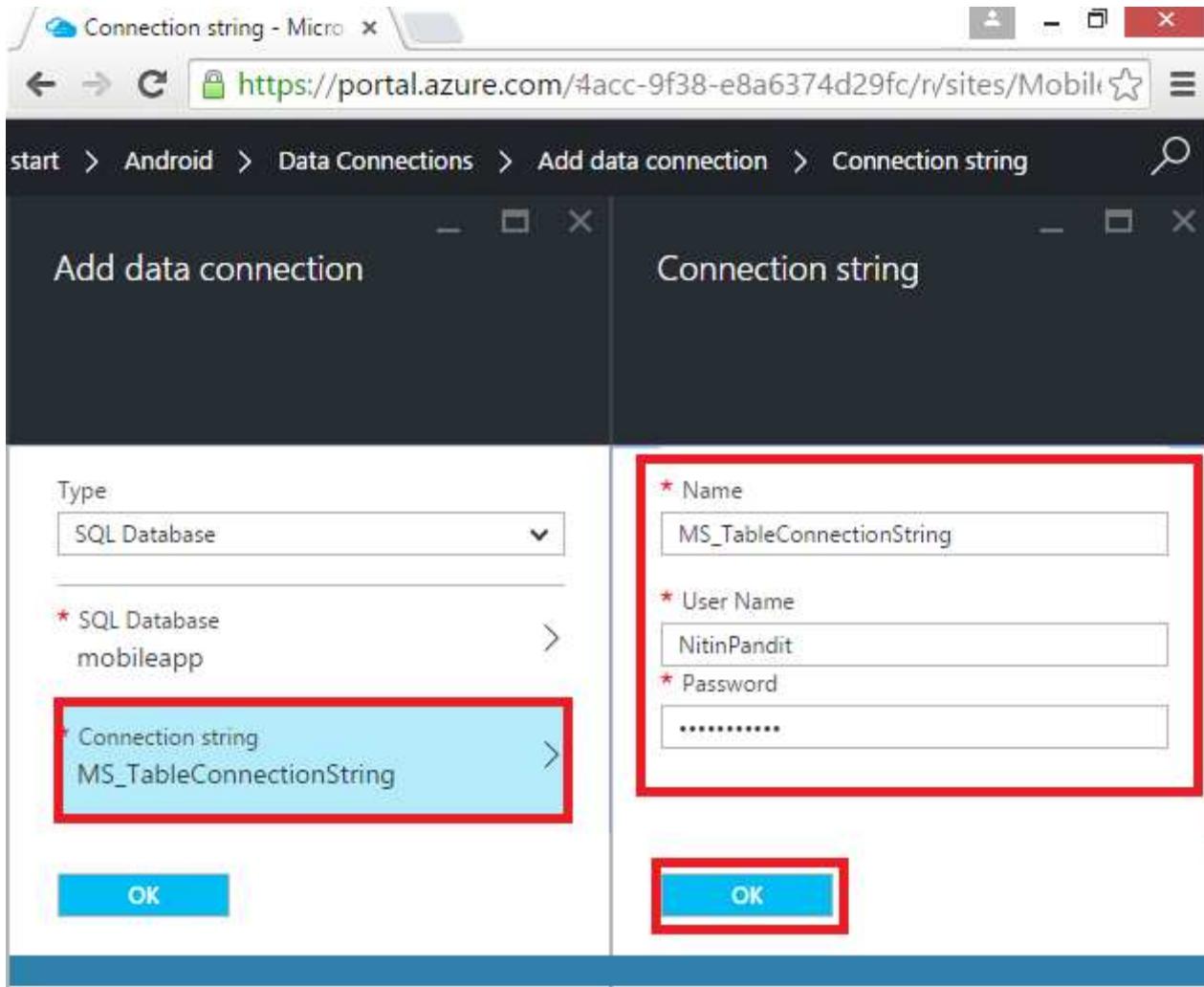
For creating database I also need of a server, so in the left side of the window give the name to your database and then click on to server for creating a new server. Whenever you click on it, the new server window will be open.

For creating a server give a unique name to your server, login name, and password with confirm password.

After filling all the necessary details, click on to “OK” button. So now our database and server is ready for use.



Click on to Connection string, type the login and password values for your database, then click OK twice.



After that our first step of creating the database would be ready, now go for the second step.

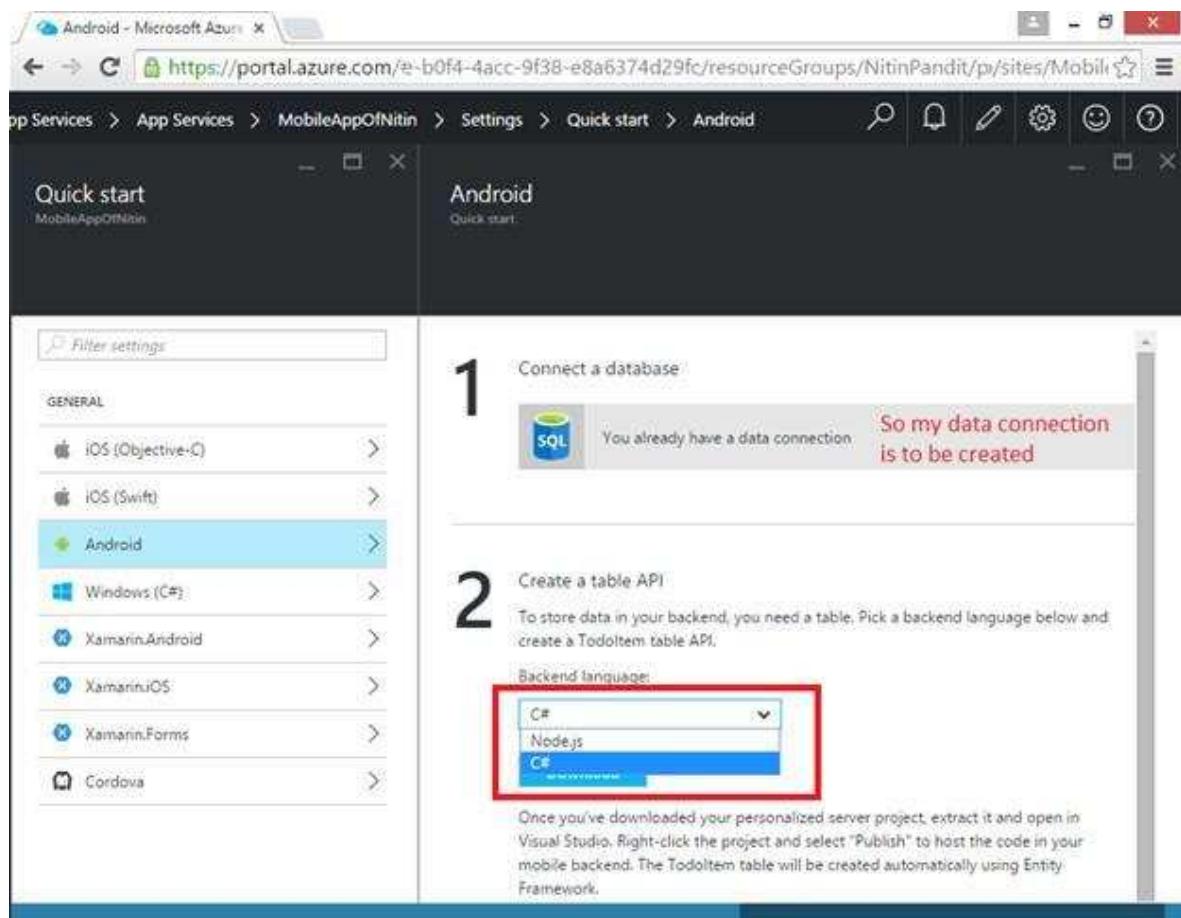
Under create a table API, select your Backend language, either C# or Node.js:

- **Node.js backend (via portal):**

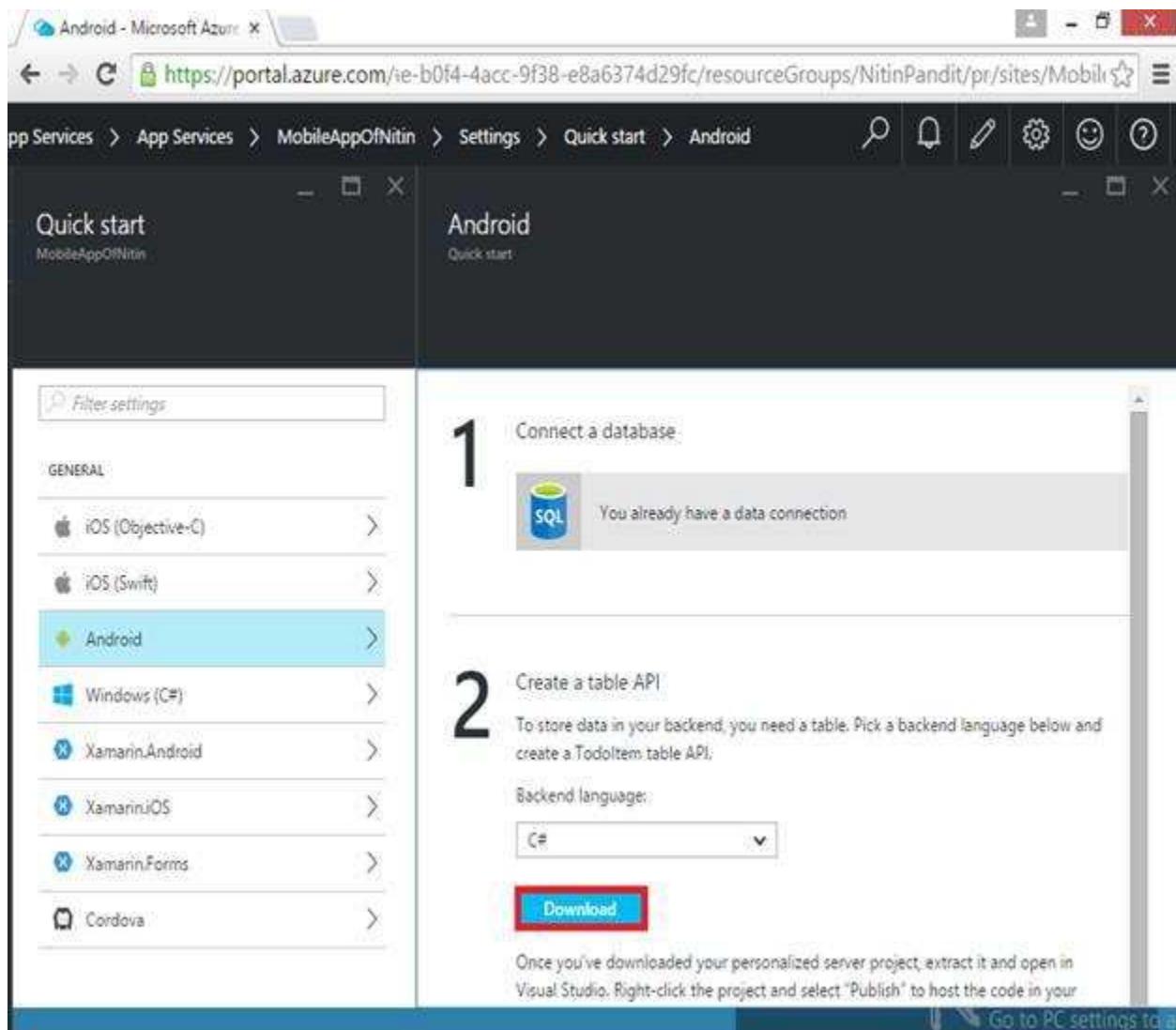
Click on to create TodoItem table and this creates a new *TodoItem* table in your database.

- **NET backend (C#):**

Click Download, extract the compressed project files to your local computer, open the solution in Visual Studio, build the project to restore the NuGet packages, and then deploy the project to Azure. Your Mobile App backend is now ready to use with your client app.

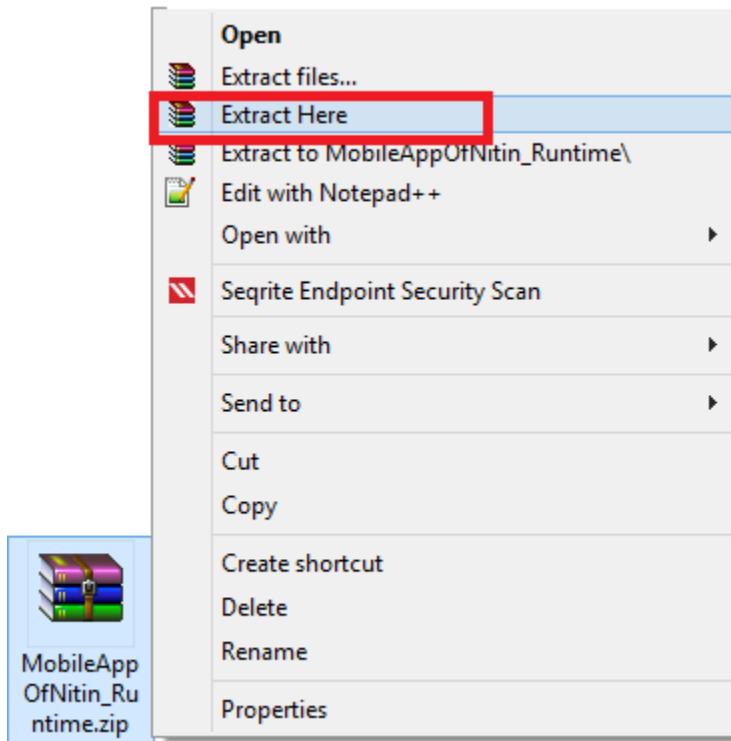


I'm going to select C#, and click on to “*Download*” button.



The screenshot shows the Azure portal interface for an Android application. The URL in the browser is <https://portal.azure.com/#ie-b0f4-4acc-9f38-e8a6374d29fc/resourceGroups/NitinPandit/pr/sites/MobileAppOfNitin>. The left sidebar shows a navigation path: App Services > App Services > MobileAppOfNitin > Settings > Quick start > Android. The main content area is titled "Android" and "Quick start". On the left, there's a "General" settings panel with a "Filter settings" input field and a list of options: iOS (Objective-C), iOS (Swift), **Android** (which is selected and highlighted in blue), Windows (C#), Xamarin.Android, Xamarin.iOS, Xamarin.Forms, and Cordova. To the right, there are two numbered steps: Step 1, "Connect a database", which shows a "SQL" icon and a message stating "You already have a data connection"; Step 2, "Create a table API", which asks to "store data in your backend, you need a table. Pick a backend language below and create a TodoItem table API". It includes a dropdown menu set to "C#" and a "Download" button. Below the download button, there's a note: "Once you've downloaded your personalized server project, extract it and open in Visual Studio. Right-click the project and select 'Publish' to host the code in your Azure mobile app." A "Go to PC settings to a..." link is also visible.

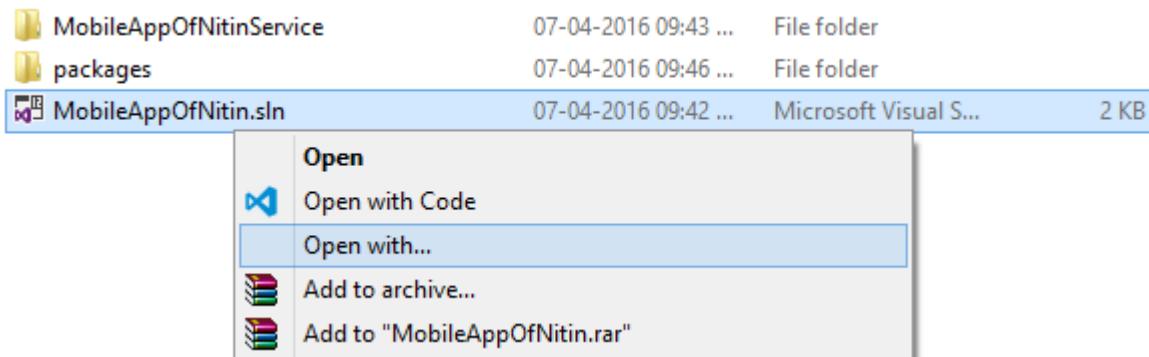
A file will be downloaded, extract the file and open it.



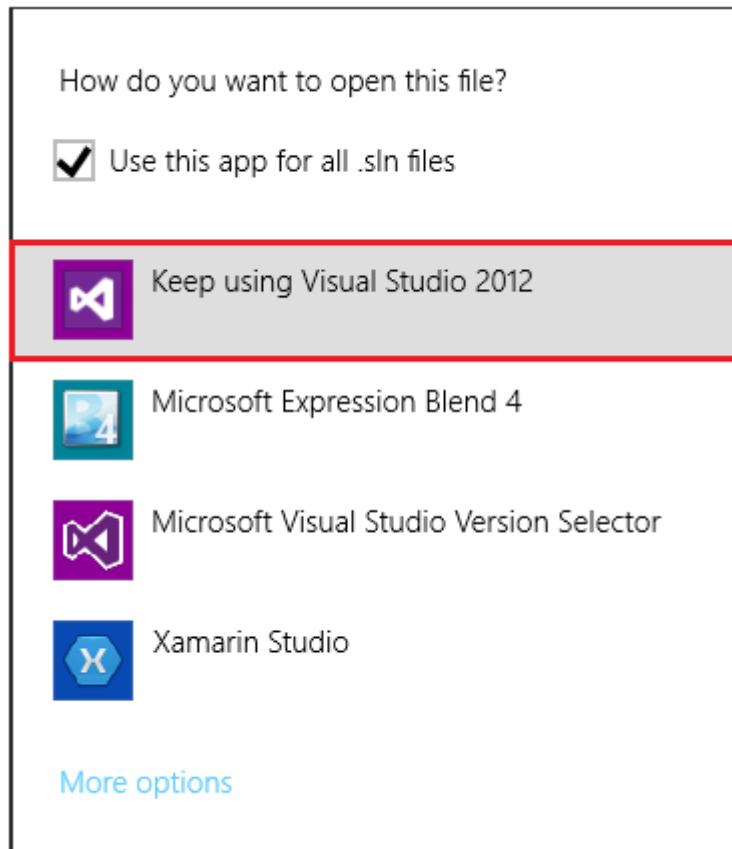
You'll get the solution of your Android mobile app with “.sln”, so you can directly use this into your Visual Studio to customize that.

Name	Date modified	Type	Size
MobileAppOfNitinService	07-04-2016 09:43 ...	File folder	
packages	07-04-2016 09:46 ...	File folder	
MobileAppOfNitin.sln	07-04-2016 09:42 ...	Microsoft Visual S...	2 KB

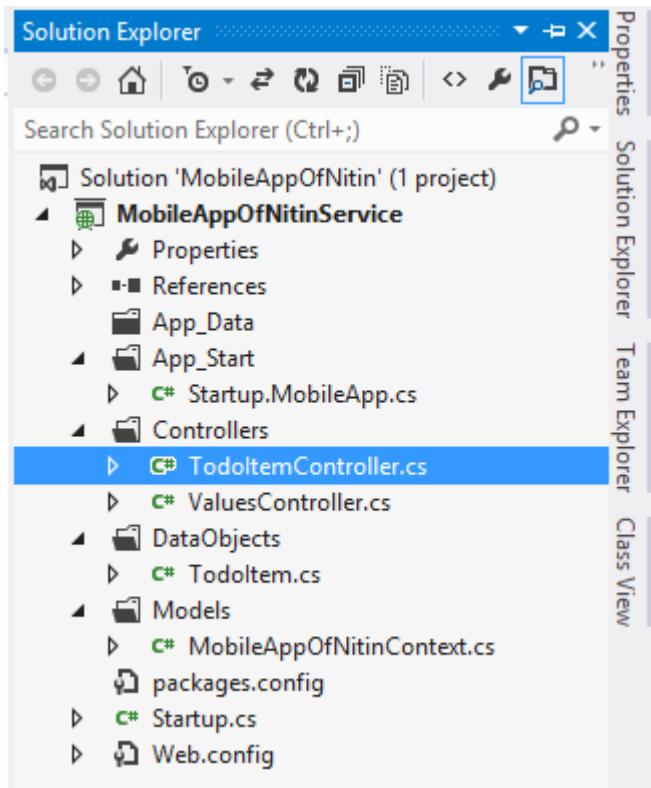
Now just right click on “.sln” file and click *Open with...*



From here just select any IDE by which you want to edit this.



This is your solution now.



We have a test API controller already created. Open it and change it as you want to get data.



```

namespace MobileAppOfNitinService.Controllers
{
    // Use the MobileAppController attribute for each ApiController you want to use
    // from your mobile clients.
    [MobileAppController]
    public class ValuesController : ApiController
    {
        // GET api/values
        public string Get()
        {
            MobileAppSettingsDictionary settings = this.Configuration.GetMobileAppSettingsProvider().GetMobileAppSettings();
            ITraceWriter traceWriter = this.Configuration.Services.GetTraceWriter();

            string host = settings.HostName ?? "localhost";
            string greeting = "Hello from " + host;

            traceWriter.Info(greeting);
            return greeting;
        }

        // POST api/values
        public string Post()
        {
            return "Hello World!";
        }
    }
}

```

Press the Run 'app' button to build the project and start the app in the Android simulator,

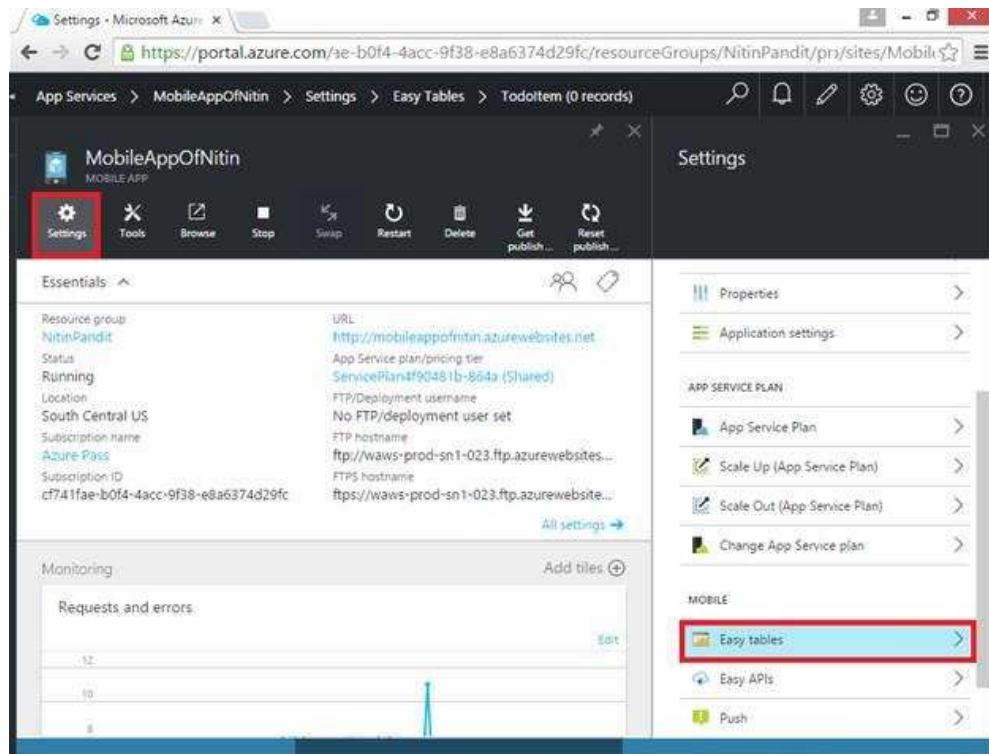


If you want to use Node.js, then select *Node.js* backend.

1 Connect a database
You already have a data connection

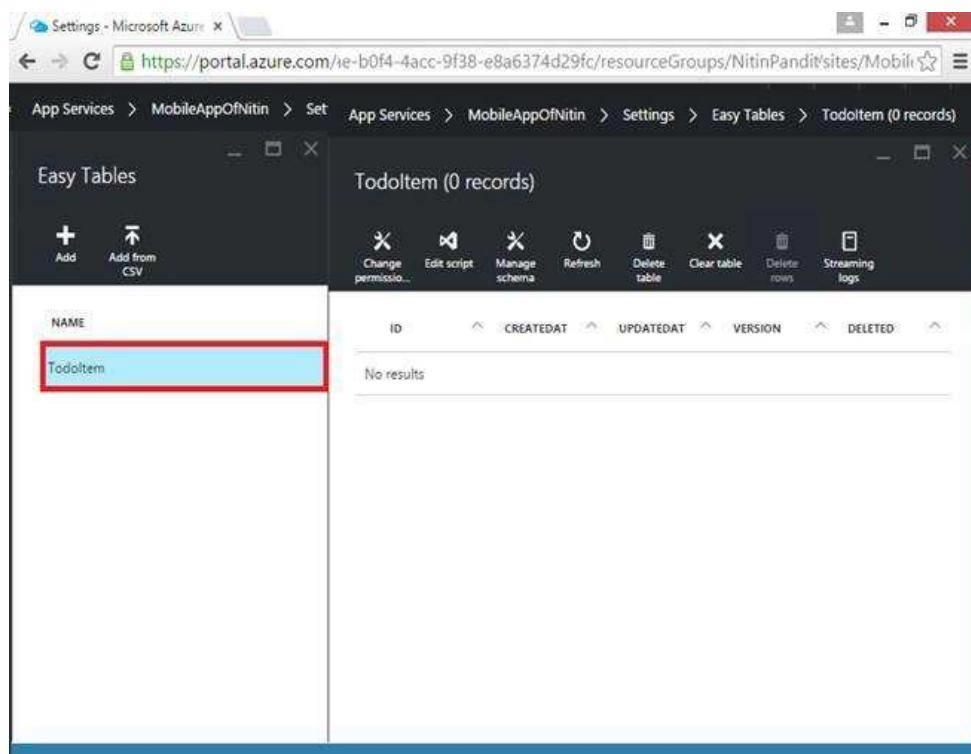
2 Create a table API
To store data in your backend, you need a table. Pick a backend language below and create a TodoItem table API.
Backend language:
 I acknowledge that this will overwrite all site contents.

Your TodoItem table is ready. To check it just go to “*Easy tables*”.



The screenshot shows the Azure portal interface for managing an app service named 'MobileAppOfNitin'. The 'Settings' tab is active. In the 'MOBILE' section, the 'Easy tables' option is highlighted with a red box.

Click on to “*TodoItem*” to check your table.



The screenshot shows the 'Easy Tables' section for the 'MobileAppOfNitin' app service. The 'TodoItem' table is listed in the table view, with its name highlighted with a red box.

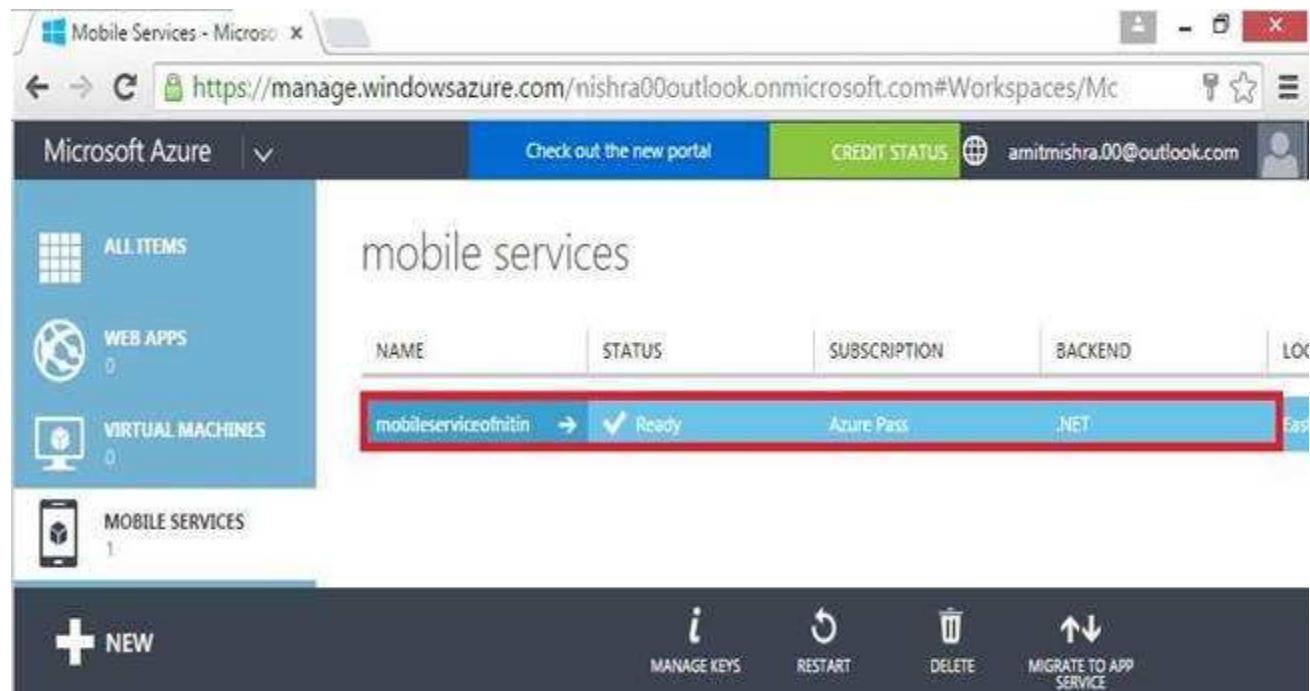
Now if you want to add new tables, then make them.

Manage Database Settings of Azure Mobile Services

In this chapter, I'm going to explain how we can change databases in existing Azure Mobile Services. The server of the new database will be the same. For creating databases you will need to create a server for the database. For creating databases as well as server you can refer to my below given chapter:

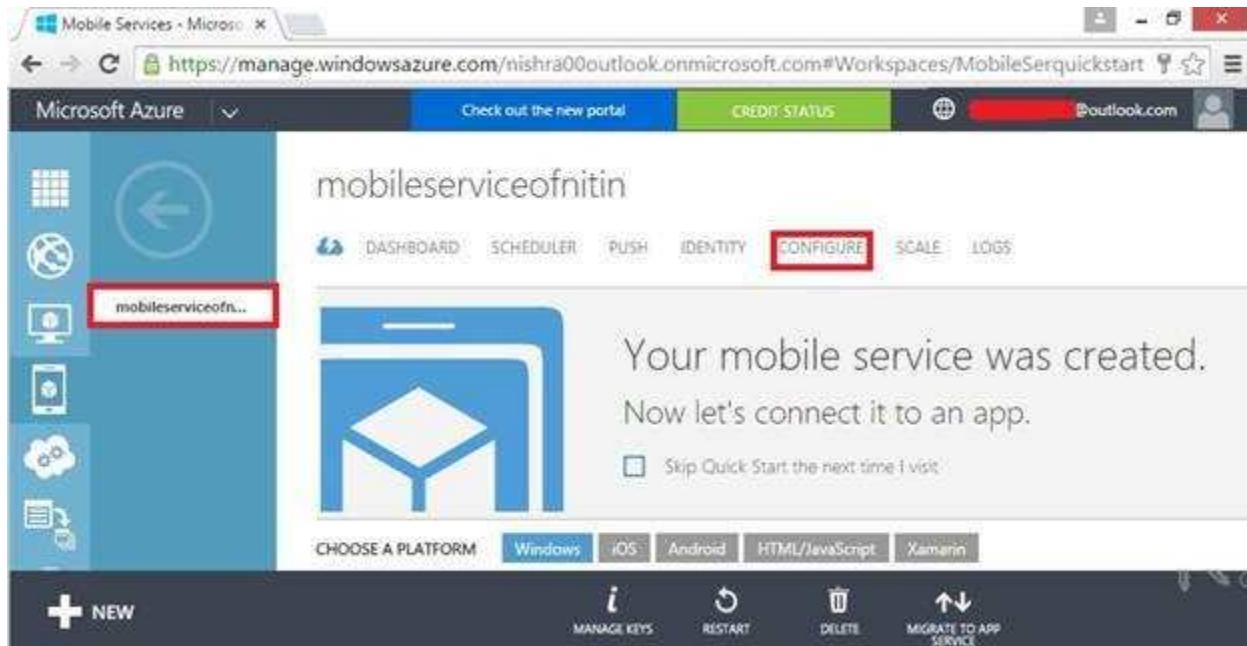
- Manage Azure Mobile Services

In the following figure, I have a mobile service, click on it.



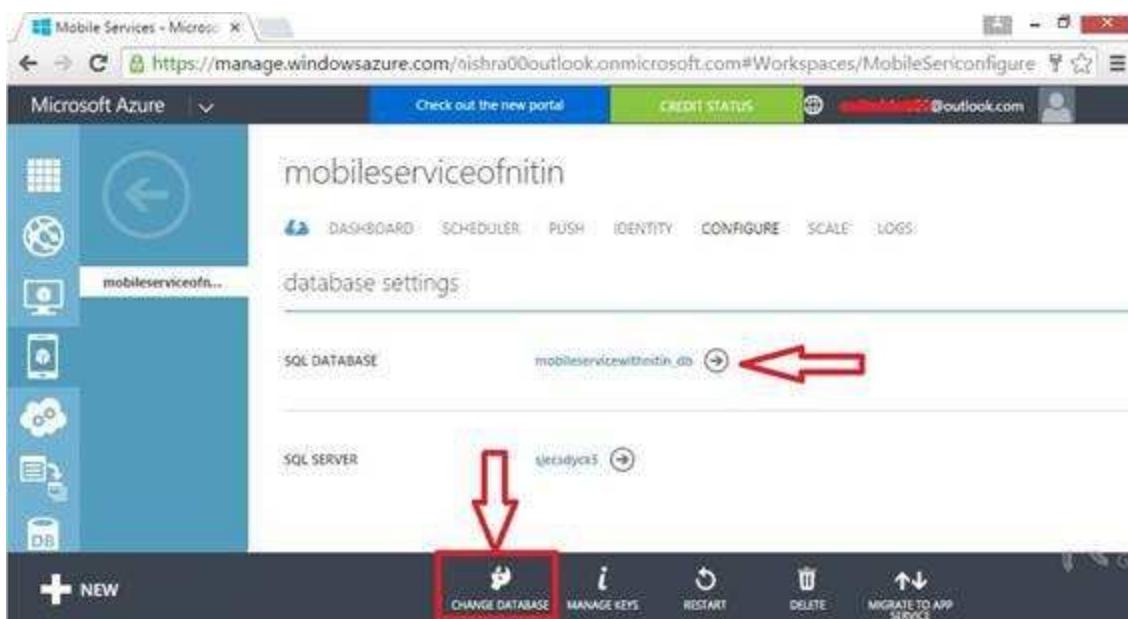
The screenshot shows the Microsoft Azure Management Portal. On the left, there's a sidebar with icons for ALL ITEMS, WEB APPS, VIRTUAL MACHINES, and MOBILE SERVICES. The MOBILE SERVICES icon has a red border around it, indicating it's selected. The main area is titled "mobile services". It lists a single service: "mobileserviceofnitin" with a status of "Ready". Below the table, there are four buttons: "MANAGE KEYS", "RESTART", "DELETE", and "MIGRATE TO APP SERVICE".

To change the database of this mobile service, click on “*CONFIGURE*”.



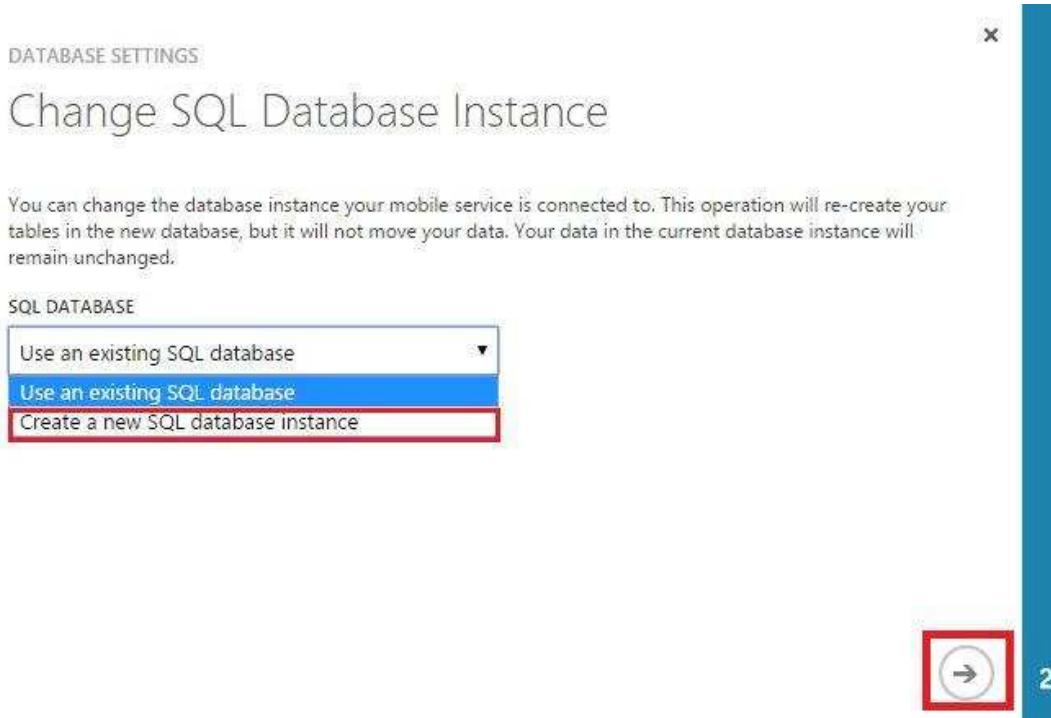
So, here in the following figure you can see SQL Database and SQL Server.

Click on to “ChangeDatabase” in the bottom of the window.



If you have more databases in your server then you can choose which you want to change, in this condition you must have to choose existing database. In my view I don't have any existing databases so I'm going to choose a new SQL Database instance option.

For further information, click on the arrow



In the following screenshot, give the database name and choose the server, my server is to be existing.

Give the server login name and password (Server login name and password would be that which you used at the time of creation your server)

Click on to check button.

DATABASE SETTINGS

Specify database settings

NAME
nitin **your db name** 

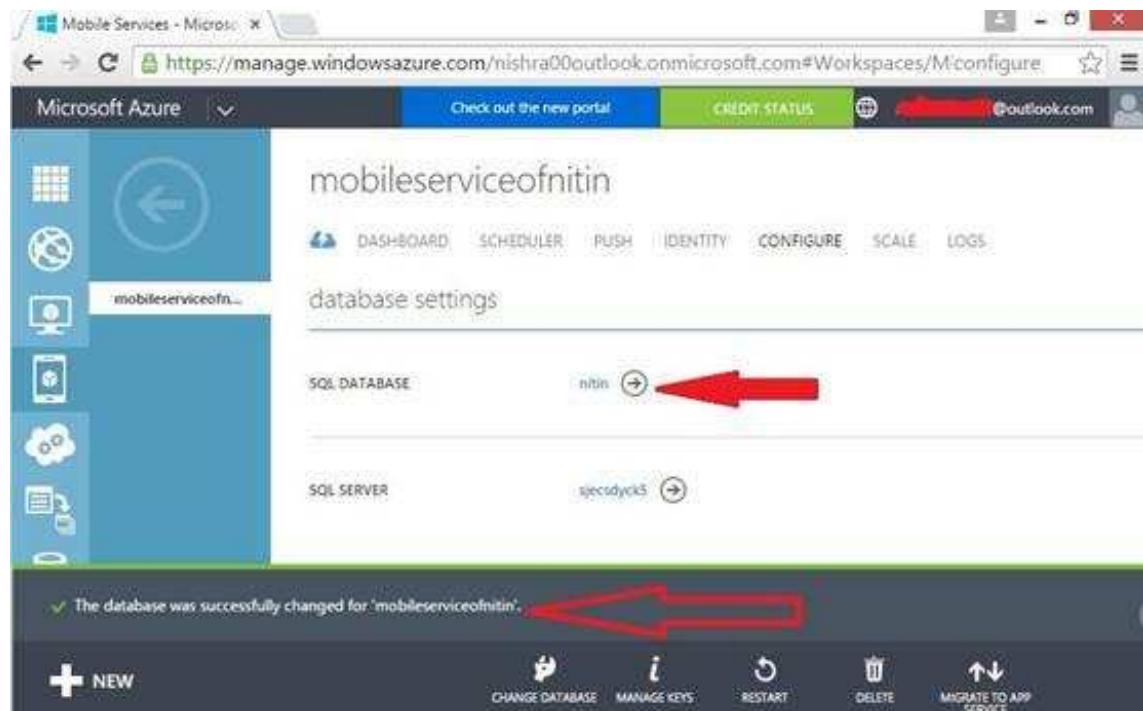
SERVER
sjecsdyck5 (East Asia, DTUs Available=44995, V) 

SERVER LOGIN NAME
serverofnitin **Login Name** 

SERVER LOGIN PASSWORD
***** **Password**

CONFIGURE ADVANCED DATABASE SETTINGS  

So here I have successfully changed my database.



The screenshot shows the Microsoft Azure mobile service configuration page for 'mobileserviceofnitin'. A red arrow points to the 'nitin' database entry under 'SQL DATABASE'. Another red arrow points to the success message at the bottom: 'The database was successfully changed for 'mobileserviceofnitin''. The interface includes a sidebar with various service icons and a navigation bar with links like DASHBOARD, SCHEDULER, PUSH, IDENTITY, and CONFIGURE.

So the benefit of changing the database is that we can use our best database in our existing mobile service.

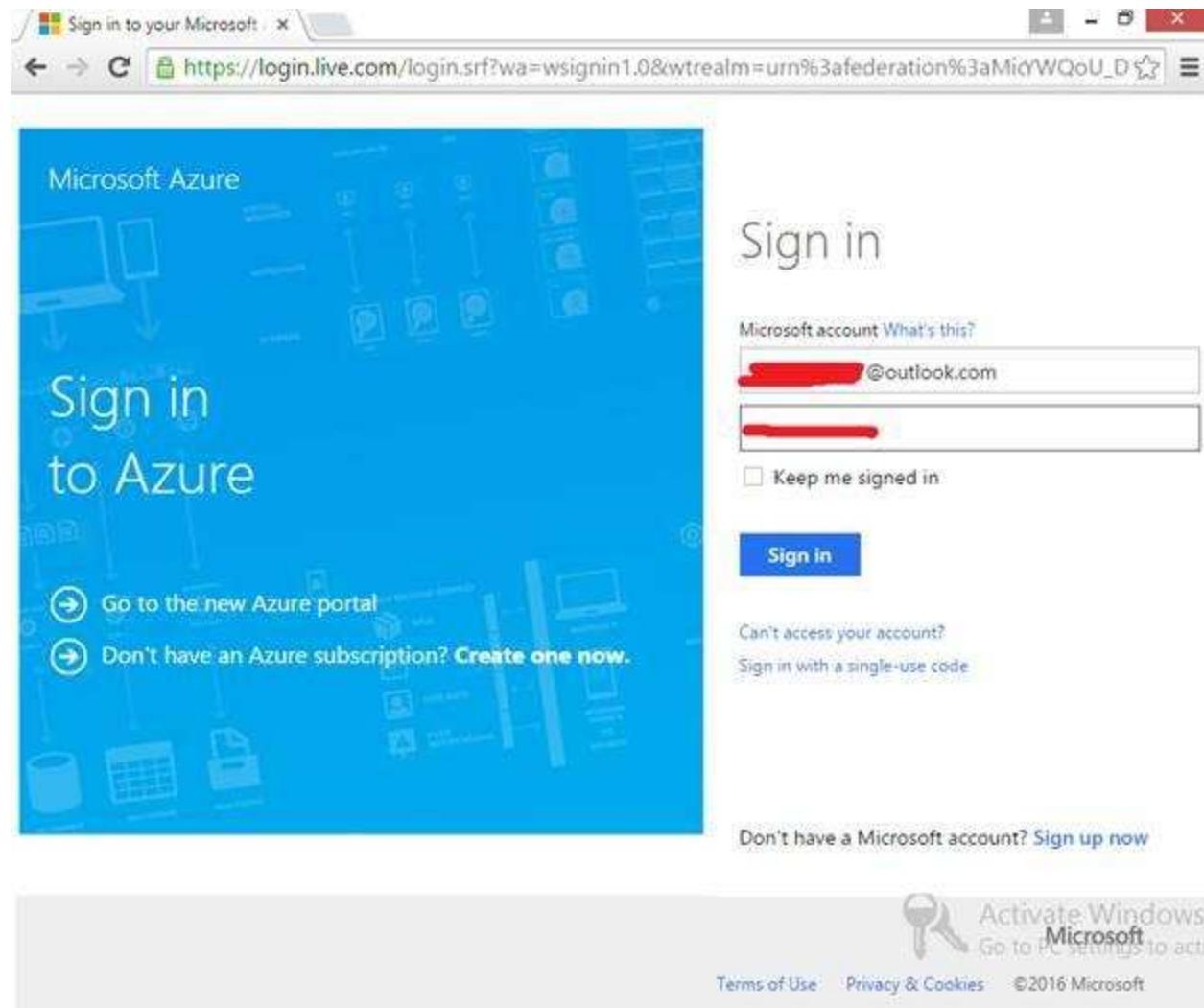
Get Twitter Account Connected Into Microsoft Azure Blob Storage Live

In this chapter we'll learn how to connect our Twitter account with Microsoft Azure Blob storage account and get the live tweets to Azure step by step.

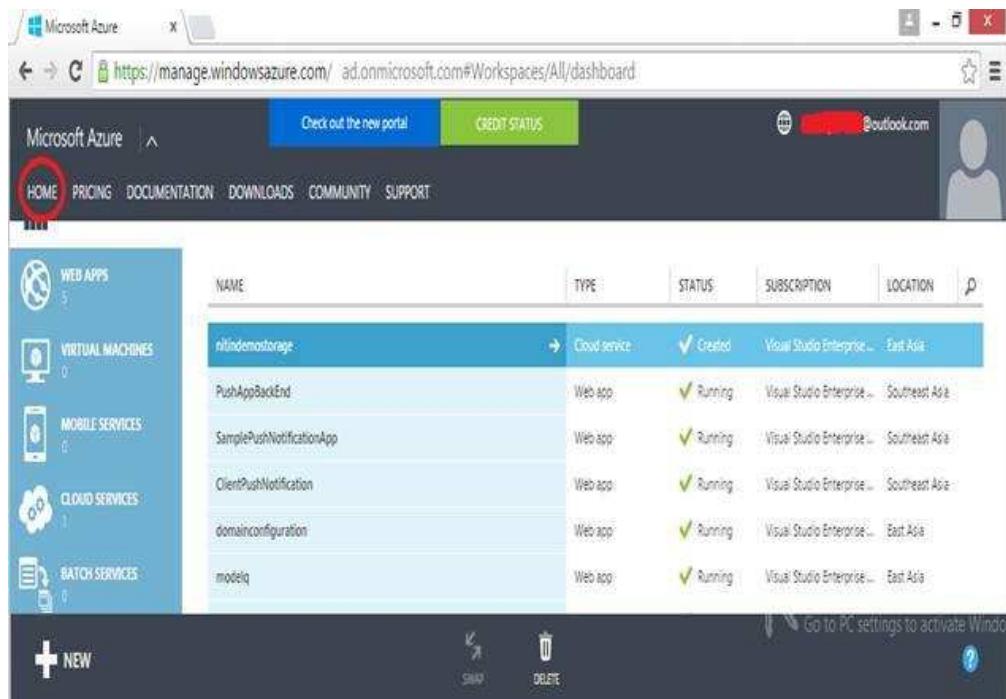
Here are the steps:

Step1: Go to Microsoft account login page and type your login details.

Step2: Click “Sign in” button.



For working on the Azure Portal, go to the “**HOME**” button on the top of the window and click on it.

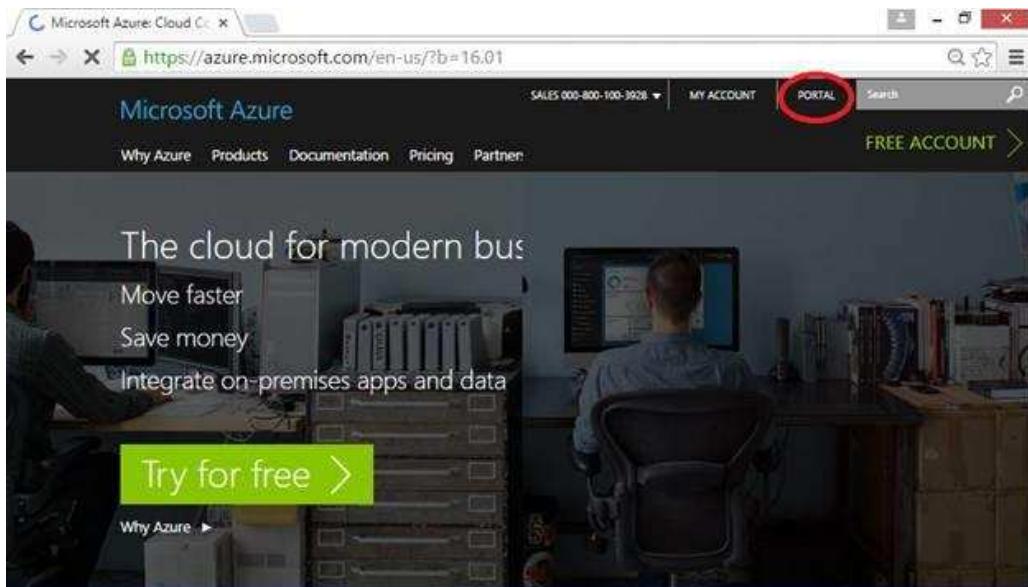


The screenshot shows the Microsoft Azure Management Portal. At the top, there's a navigation bar with links for "Check out the new portal", "CREDIT STATUS", and a user profile. Below the navigation bar is a menu bar with "HOME" (circled in red), "PRICING", "DOCUMENTATION", "DOWNLOADS", "COMMUNITY", and "SUPPORT". On the left, there's a sidebar with icons for "WEB APPS" (5), "VIRTUAL MACHINES" (0), "MOBILE SERVICES" (0), "CLOUD SERVICES" (1), and "BATCH SERVICES" (0). The main area displays a table of resources:

NAME	TYPE	STATUS	SUBSCRIPTION	LOCATION
nlindemstorage	Cloud service	✓ Created	Visual Studio Enterprise ...	East Asia
PushAppBackEnd	Web app	✓ Running	Visual Studio Enterprise ...	Southeast Asia
SamplePushNotificationApp	Web app	✓ Running	Visual Studio Enterprise ...	Southeast Asia
ClientPushNotification	Web app	✓ Running	Visual Studio Enterprise ...	Southeast Asia
domainconfiguration	Web app	✓ Running	Visual Studio Enterprise ...	East Asia
modelq	Web app	✓ Running	Visual Studio Enterprise ...	East Asia

At the bottom of the main area, there are "CREATE" and "DELETE" buttons. A message at the bottom right says "Go to PC settings to activate Windows 10".

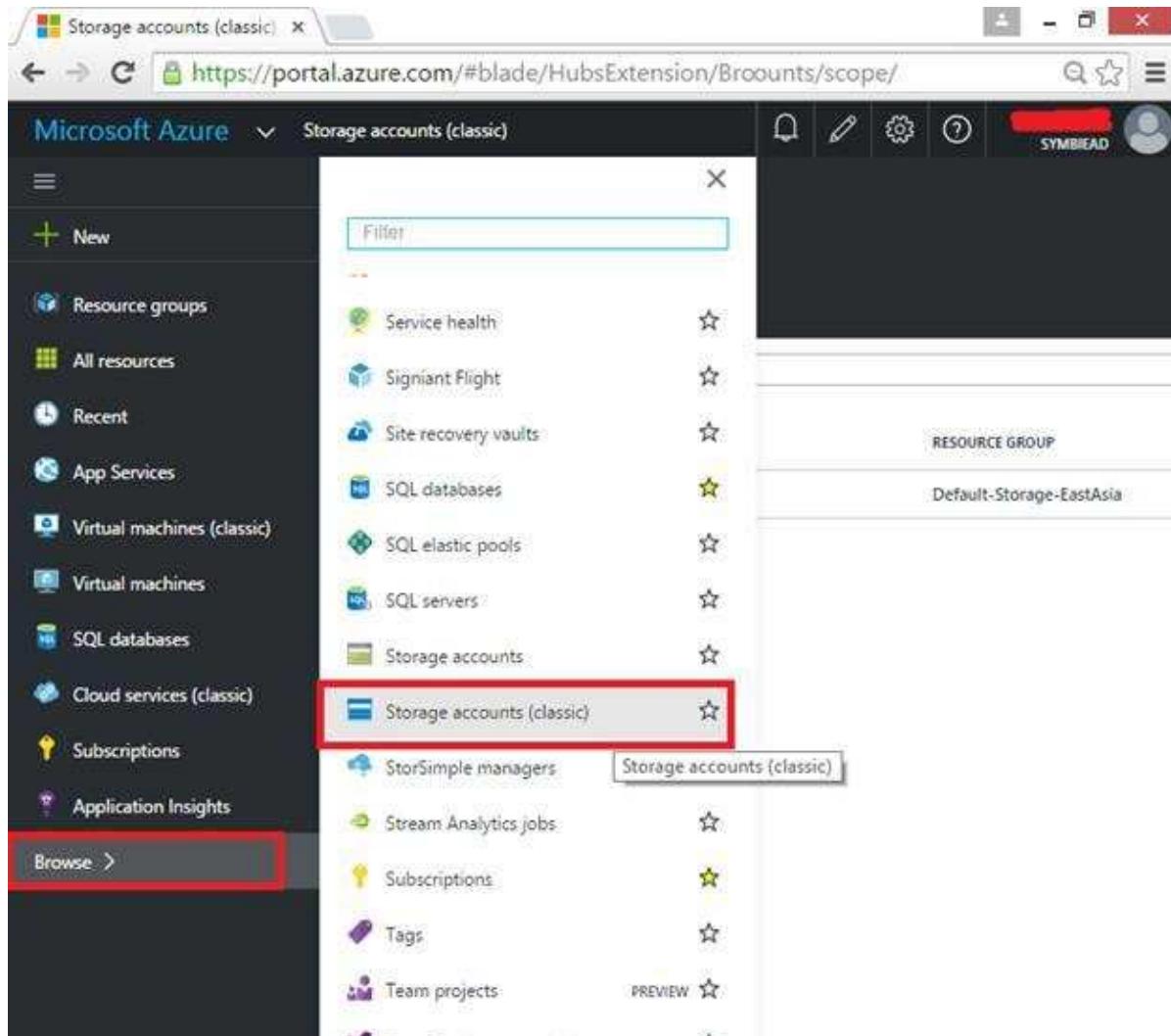
Click on to “**Portal**”.



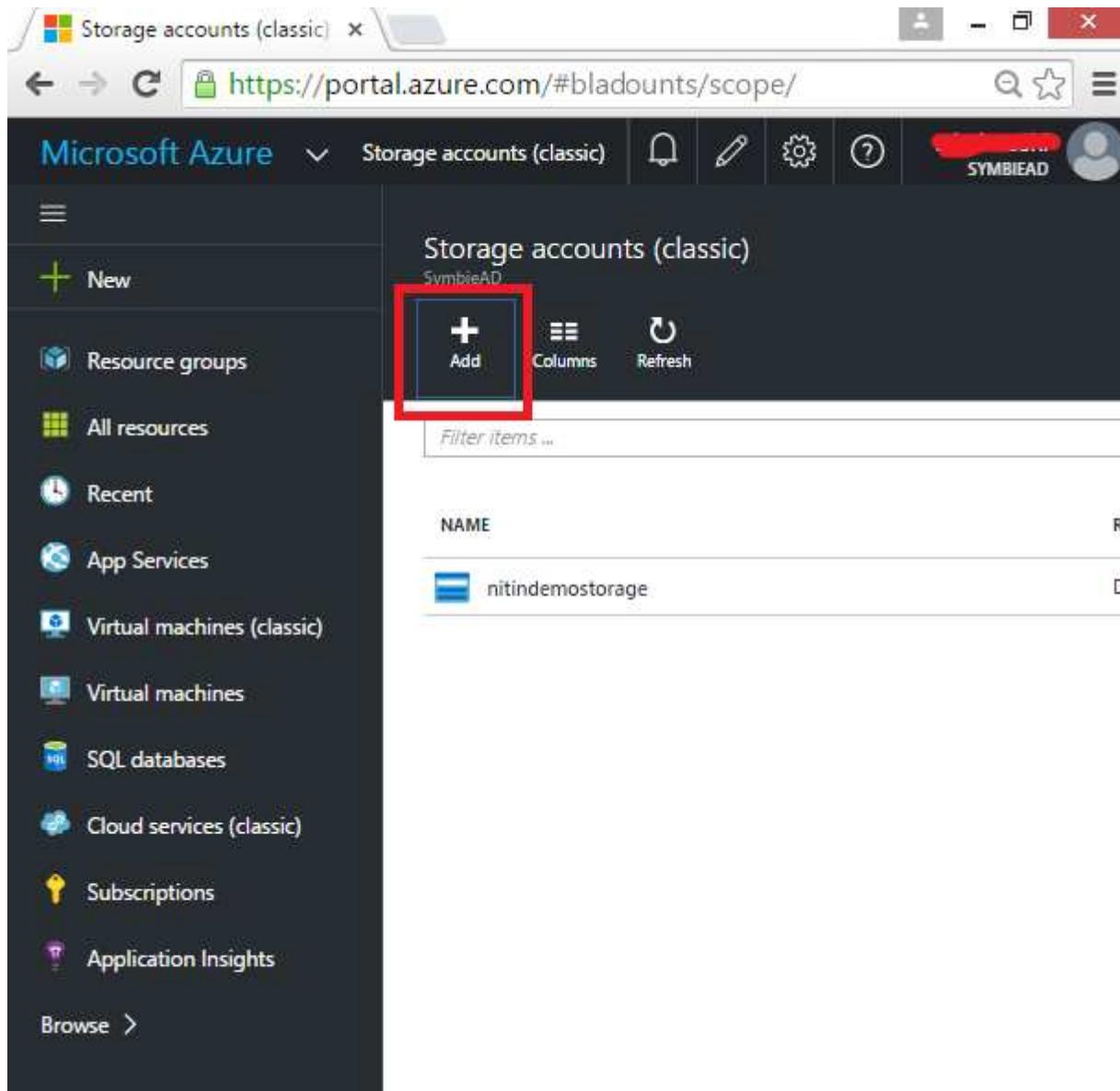
The screenshot shows the Microsoft Azure Cloud Portal. At the top, there's a navigation bar with links for "SALES 000-800-100-3928", "MY ACCOUNT", and a "PORTAL" button (circled in red). There's also a "FREE ACCOUNT" link. Below the navigation bar, there's a banner with the text "The cloud for modern business" and sub-points "Move faster", "Save money", and "Integrate on-premises apps and data". A large green "Try for free >" button is prominently displayed. At the bottom left, there's a "Why Azure" link.

First, I need to create a “Storage Account” and “Blob Container.”

So, go to “**Browse**” and scroll down, and select “**Storage accounts (classic)**.”



You will see the following figure, click on “**Add**” button to create a “Storage Account”.

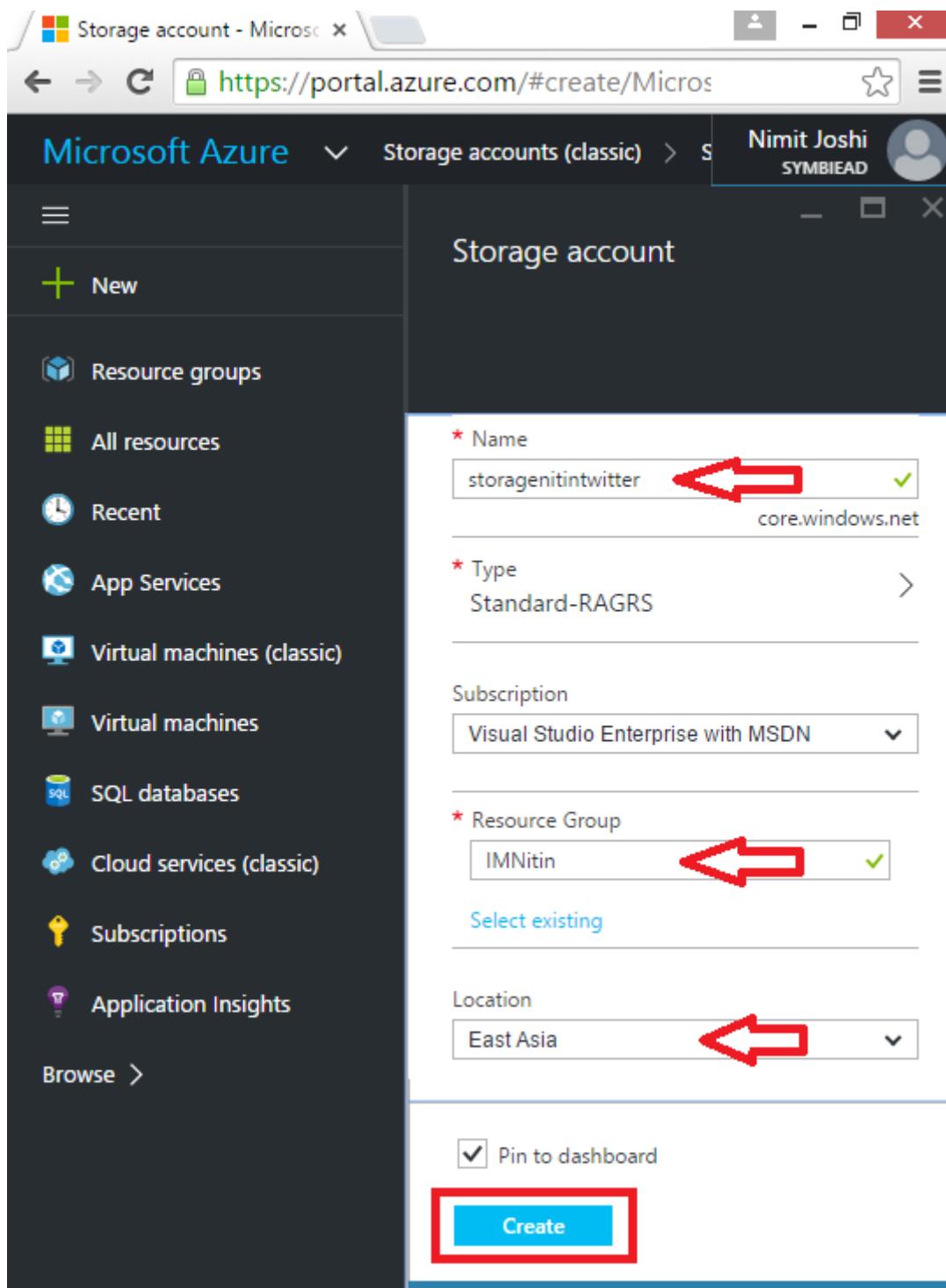


The screenshot shows the Microsoft Azure Storage accounts (classic) interface. On the left, there's a sidebar with various options like Resource groups, All resources, Recent, App Services, etc. The main area is titled 'Storage accounts (classic)' and shows a list of existing storage accounts, including 'nitindemostorage'. At the top right of this area, there are 'Add', 'Columns', and 'Refresh' buttons. The 'Add' button is highlighted with a red box.

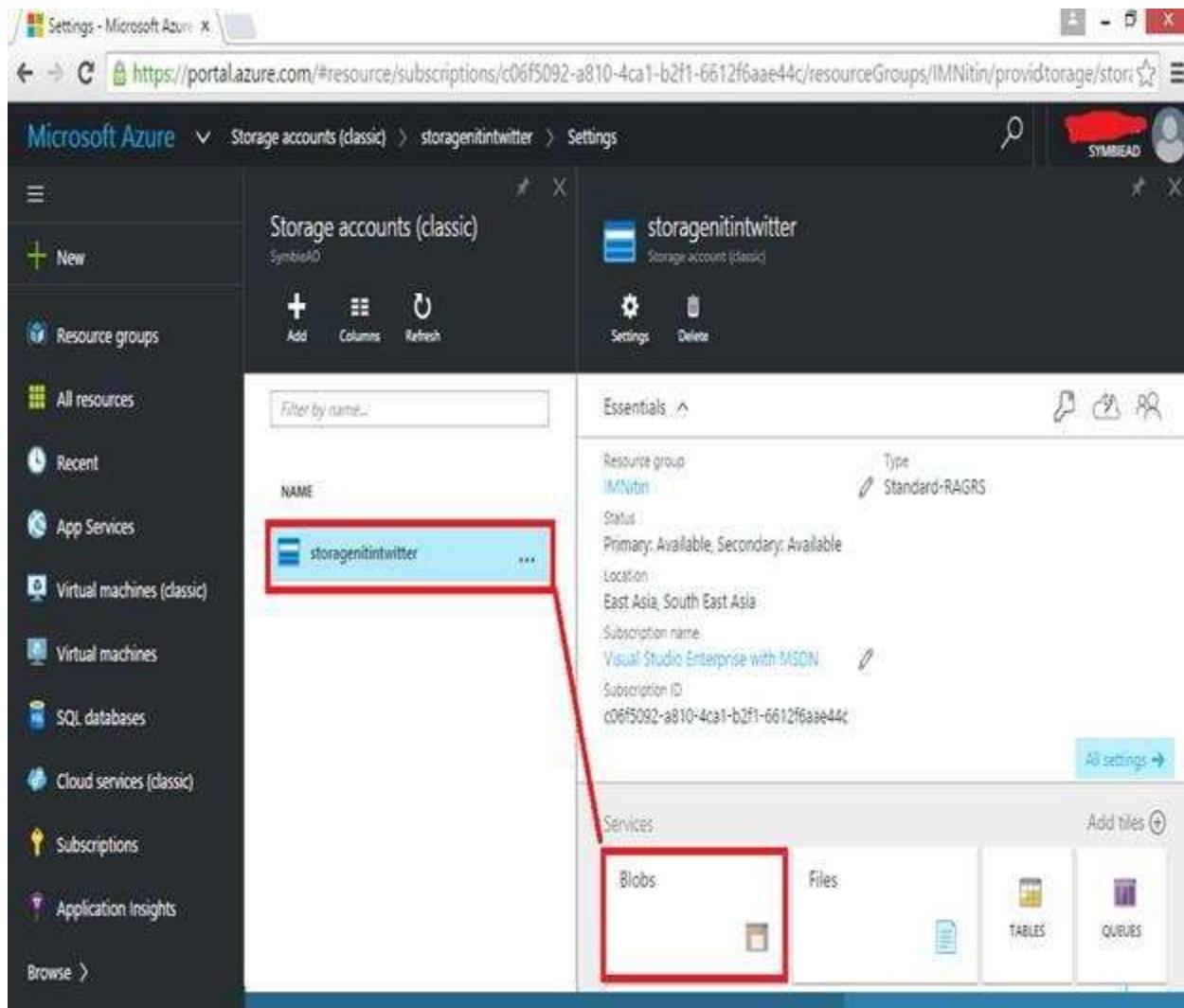
In the name section give whatever name you want, I gave it “**storagenitintwitter**.” I need to create a new “Resource Group” also. Resource Group is a place where I managed all the related account of particular storage account.

I gave “**IMNitin**” name to my resource group.

Select the location of your storage. My location of the storage is “**East Asia**”, so I selected East Asia. Click on “**Create**” button.

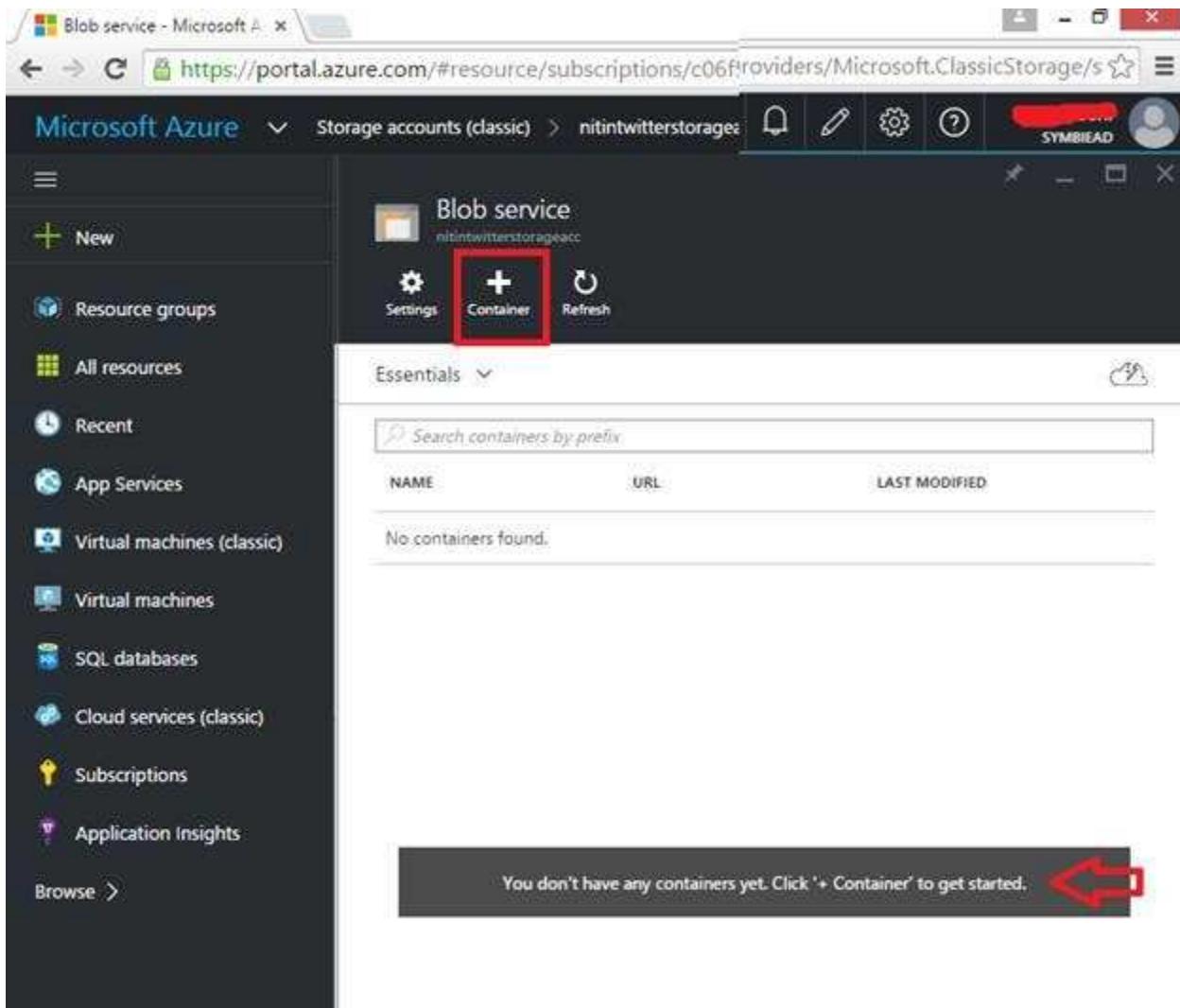


Now, I need to create a “Blob Container” for the account, so just click on Storage Account and then “Blobs” to create a blob.



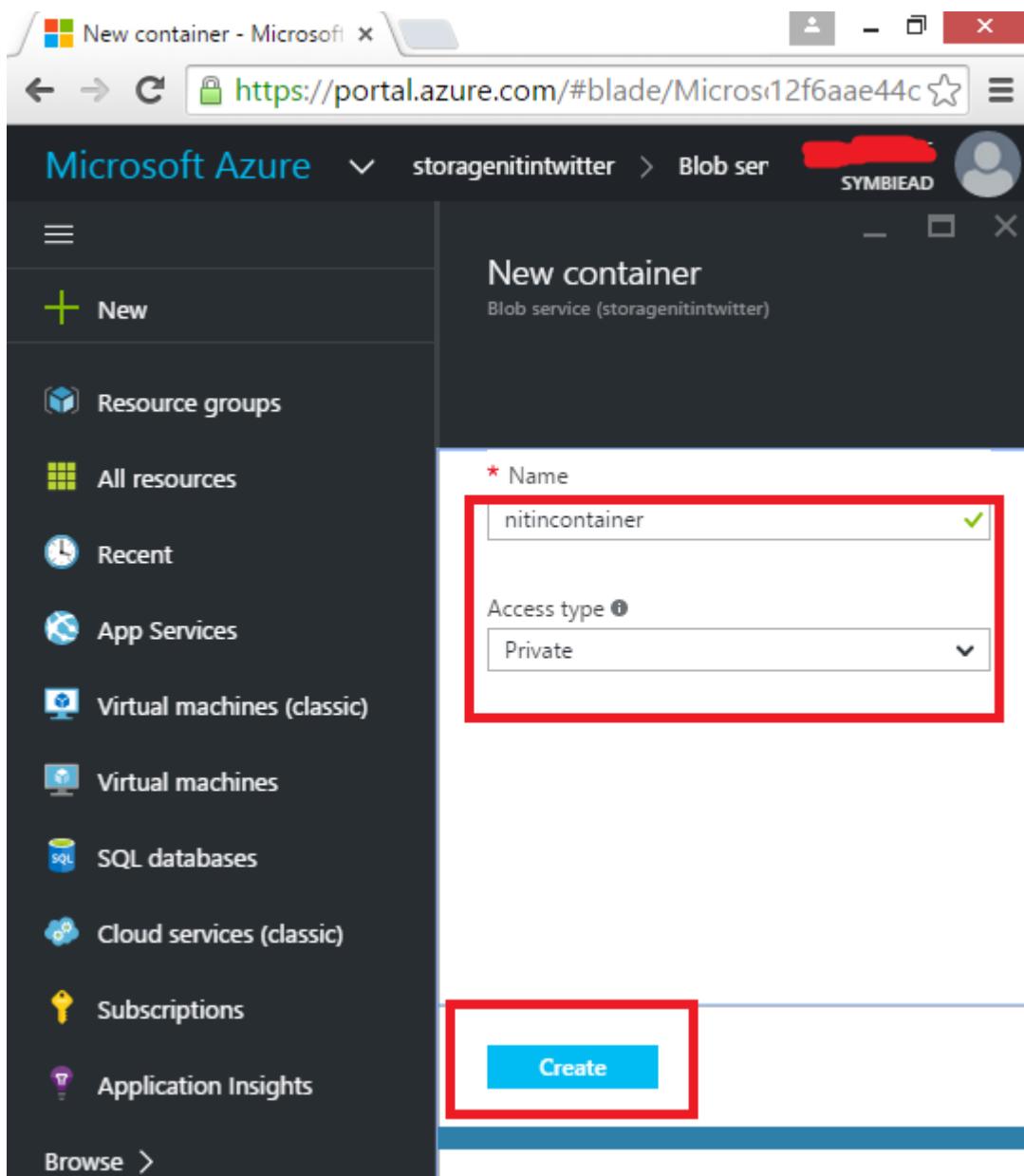
The screenshot shows the Microsoft Azure Storage accounts (classic) settings page. On the left, there's a navigation menu with options like New, Resource groups, All resources, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Subscriptions, Application Insights, and Browse. The main area displays a list of storage accounts under 'Storage accounts (classic)', with 'storagegenitintwitter' selected. This selection is highlighted with a red box. To the right of the list is a detailed view of the selected account, including its name ('storagegenitintwitter'), resource group ('IMNitin'), type ('Standard-RAGRS'), status ('Primary: Available, Secondary: Available'), location ('East Asia, South East Asia'), subscription name ('Visual Studio Enterprise with MSDN'), and subscription ID ('c06f5092-a810-4ca1-b2f1-6612f6aae44c'). Below this information is a section titled 'Services' with four tabs: Blobs (selected and highlighted with a red box), Files, TABLES, and QUEUES.

Now in the next window, click on to “Container”.



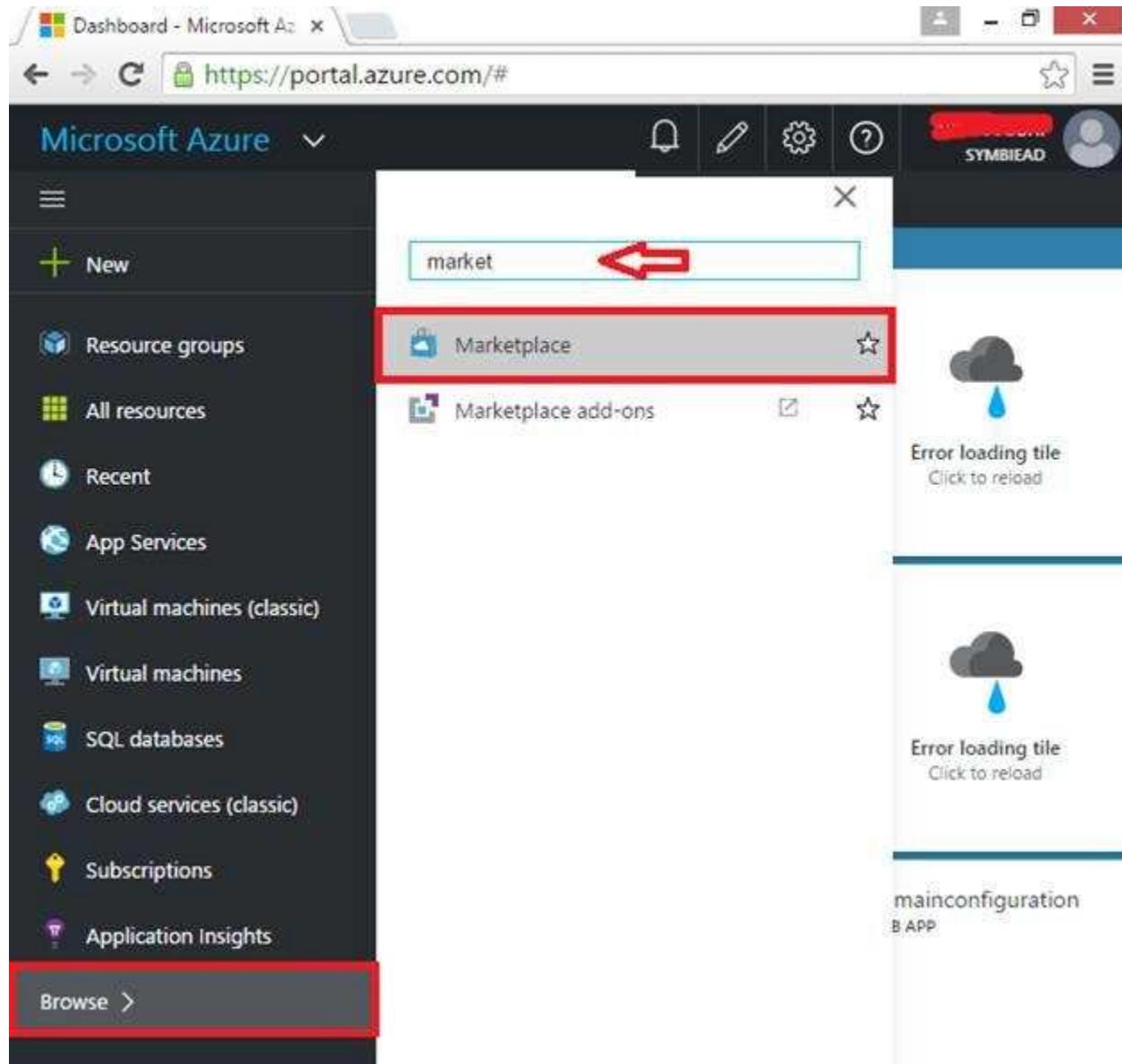
Give the name to your Container; here I have given it “**nitincontainer**.” Access type will be “Private” for your access only.

Click on “**Create**” button.



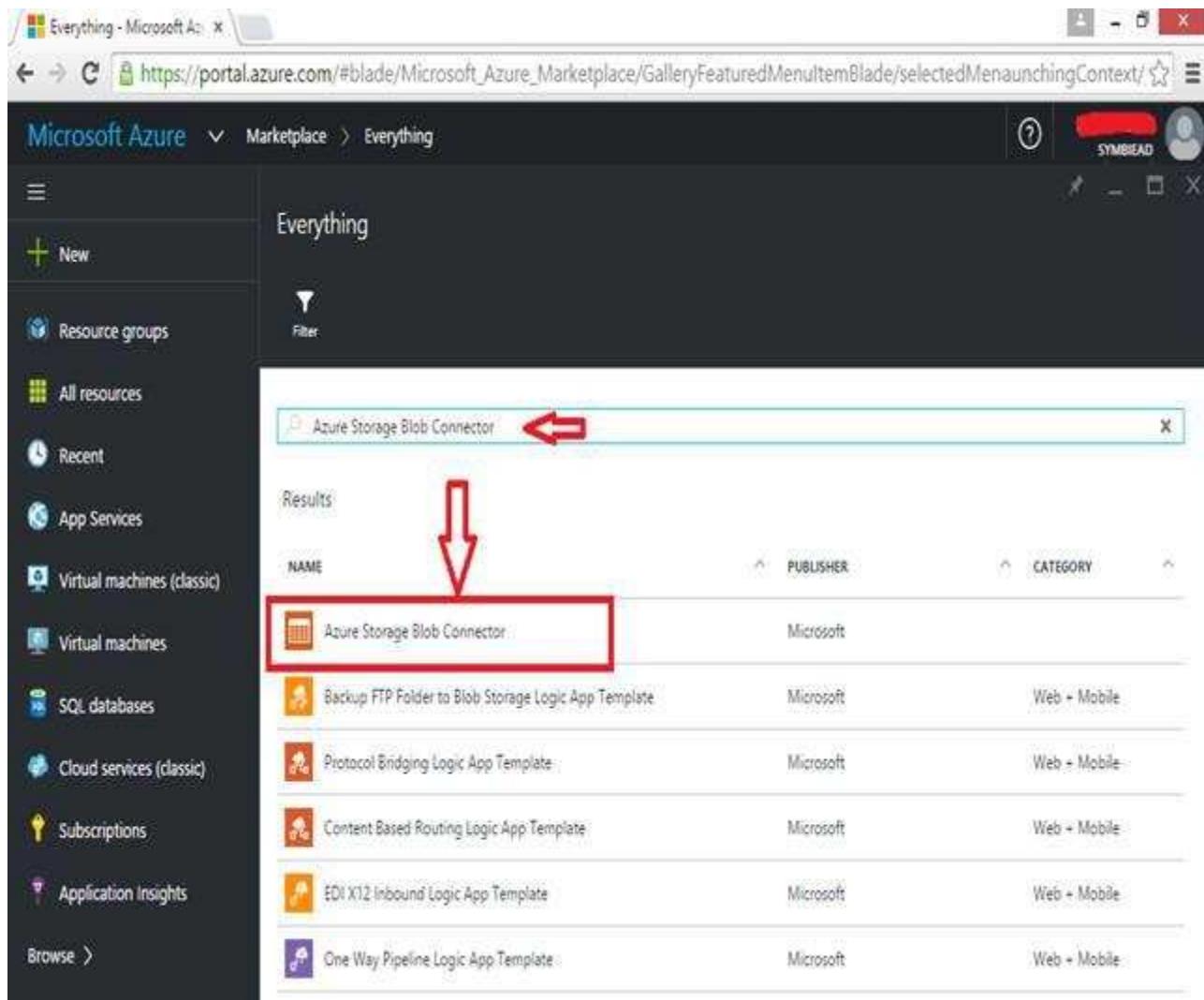
Now, I need to create an “**Azure Blob Storage Connector**.” Click on “Browse” and in the search panel type “**Marketplace**.”

Click on to “**Marketplace**.”



In the search box type “**Azure Storage Blob Connector.**”

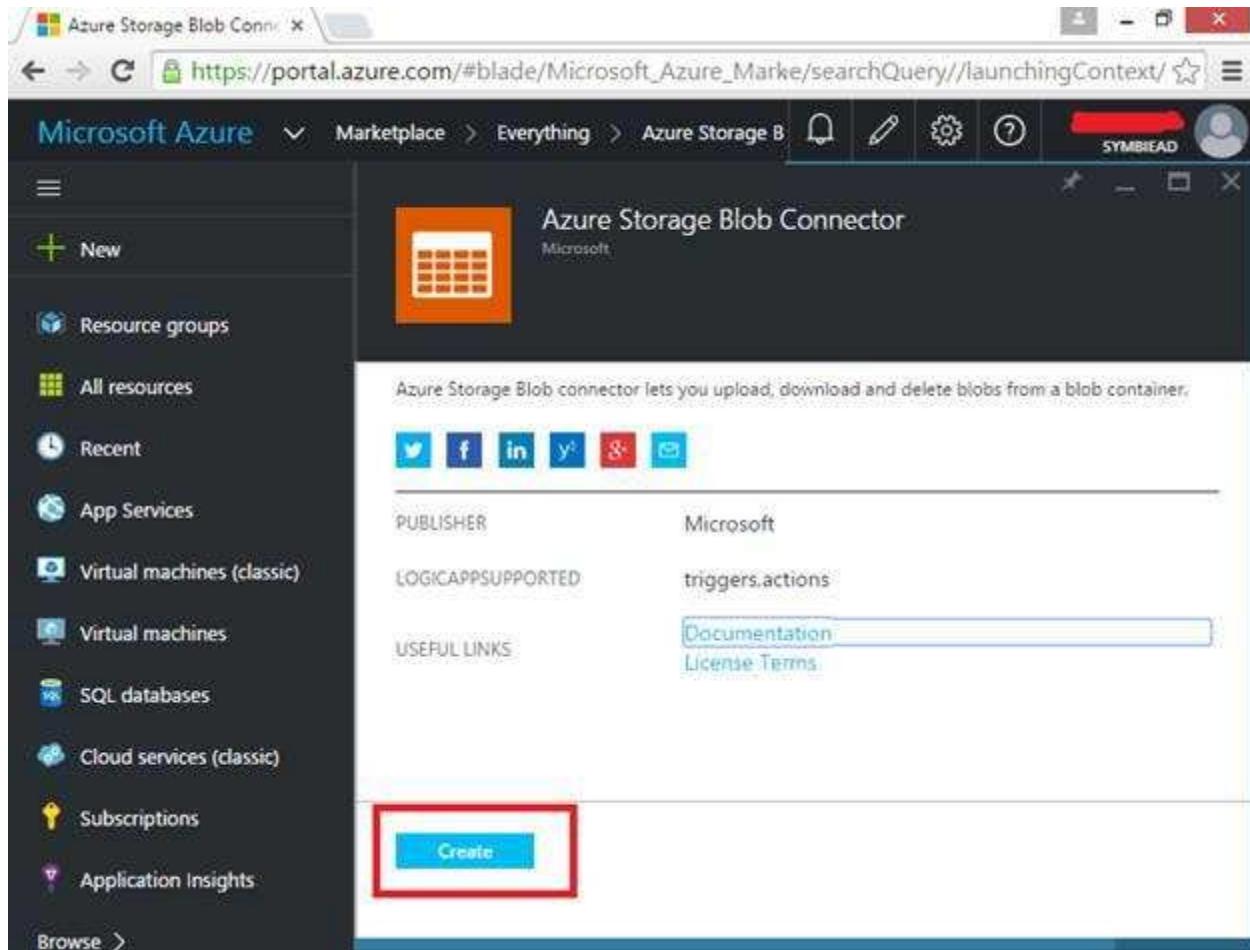
Click on to “**Azure Storage Blob Connector.**”



Azure Storage Blob Connector

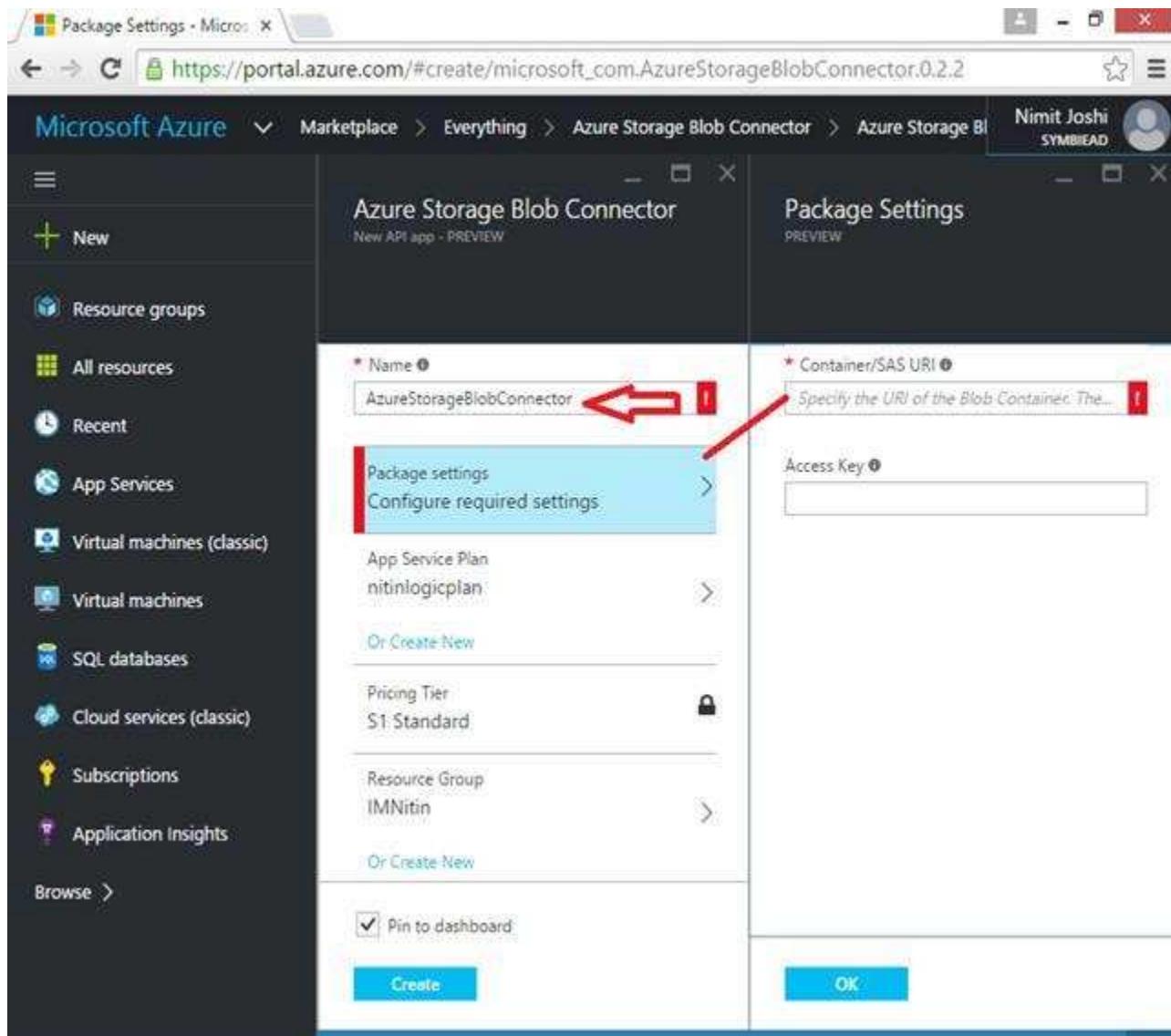
NAME	PUBLISHER	CATEGORY
Azure Storage Blob Connector	Microsoft	
Backup FTP Folder to Blob Storage Logic App Template	Microsoft	Web + Mobile
Protocol Bridging Logic App Template	Microsoft	Web + Mobile
Content Based Routing Logic App Template	Microsoft	Web + Mobile
EDI X12 Inbound Logic App Template	Microsoft	Web + Mobile
One Way Pipeline Logic App Template	Microsoft	Web + Mobile

Click on “Create” button.



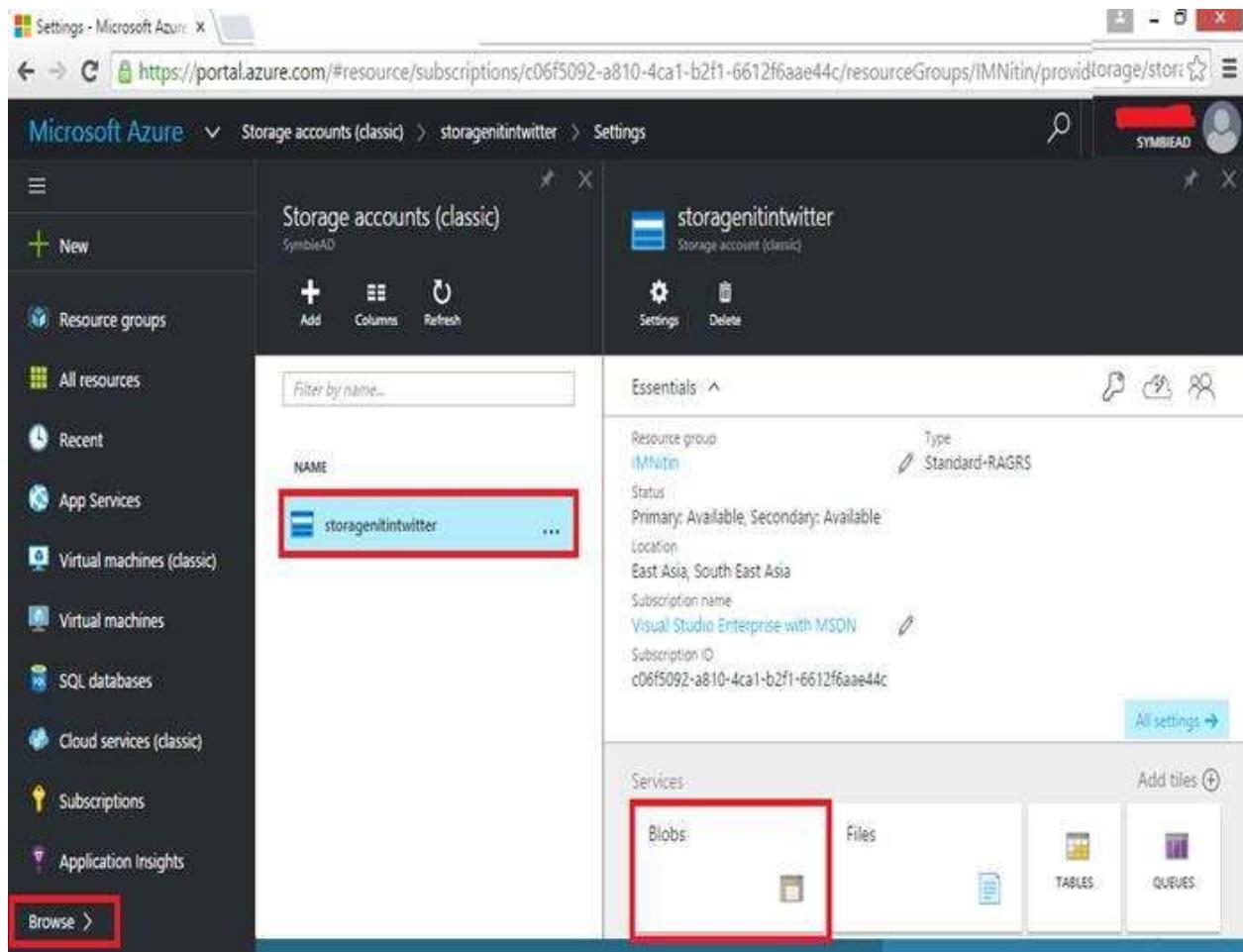
After clicking on “Create” button, give the name for Blob Connector, here I have given it “Azure Storage Blob Connector.”

Then click on “Package Settings,” it will ask for Blob Container Uri, so just go to your container and copy the path of Blob Container.



Click on “Browse” and go to your storage account.

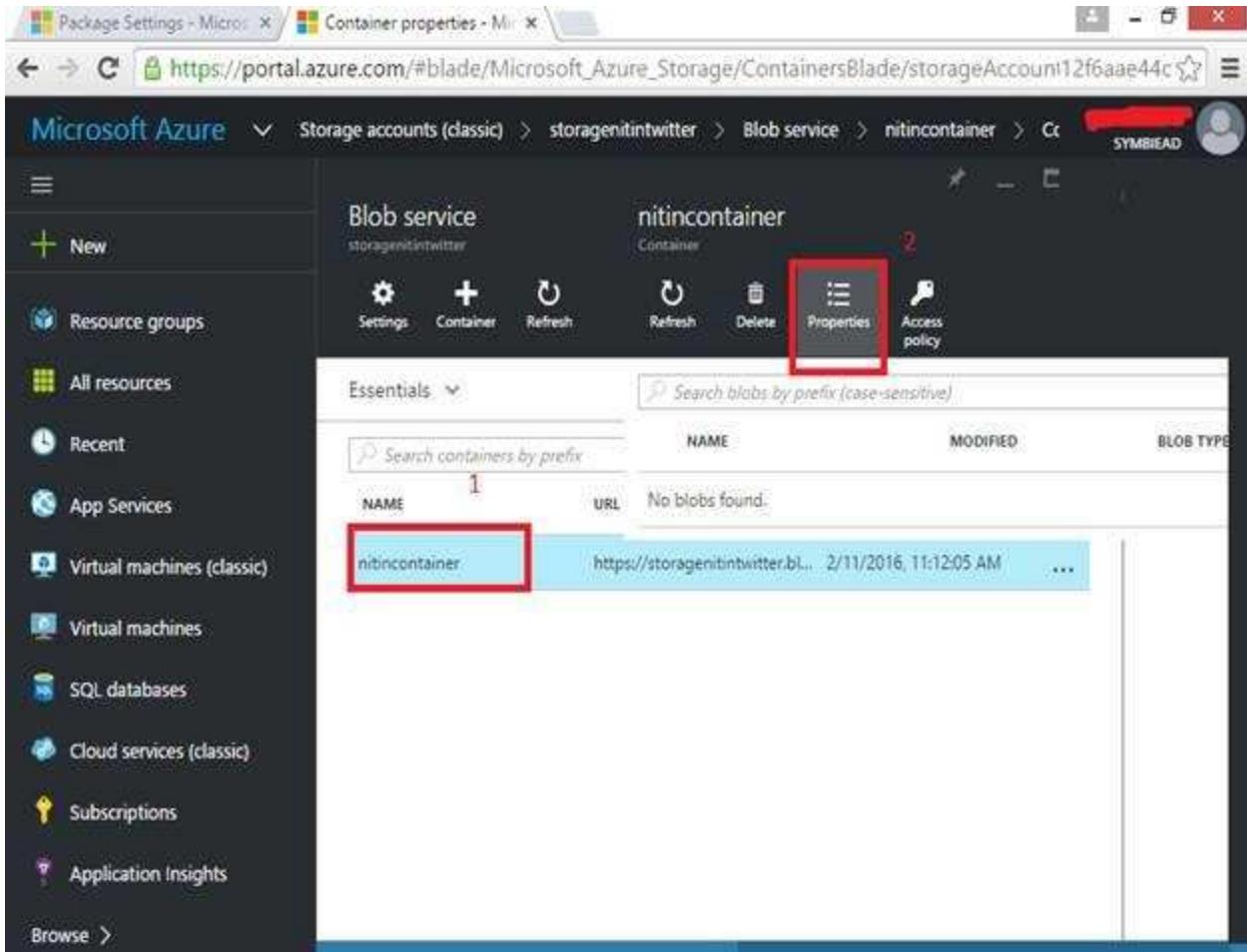
Click on to “**Blobs**.



The screenshot shows the Microsoft Azure portal interface. The left sidebar lists various services: New, Resource groups, All resources, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Subscriptions, Application Insights, and a 'Browse >' button which is also highlighted with a red box. The main content area is titled 'Storage accounts (classic)' and shows a list of storage accounts. One account, 'storagenitintwitter', is selected and highlighted with a red box. The right pane displays the 'Essentials' view for this account, showing details like Resource group (IMNitin), Type (Standard-RAGRS), Status (Primary: Available, Secondary: Available), Location (East Asia; South East Asia), Subscription name (Visual Studio Enterprise with MSDN), and Subscription ID (c06f5092-a810-4ca1-b2f1-6612f6aae44c). Below this, there are tiles for different services: Blobs (highlighted with a red box), Files, TABLES, and QUEUES.

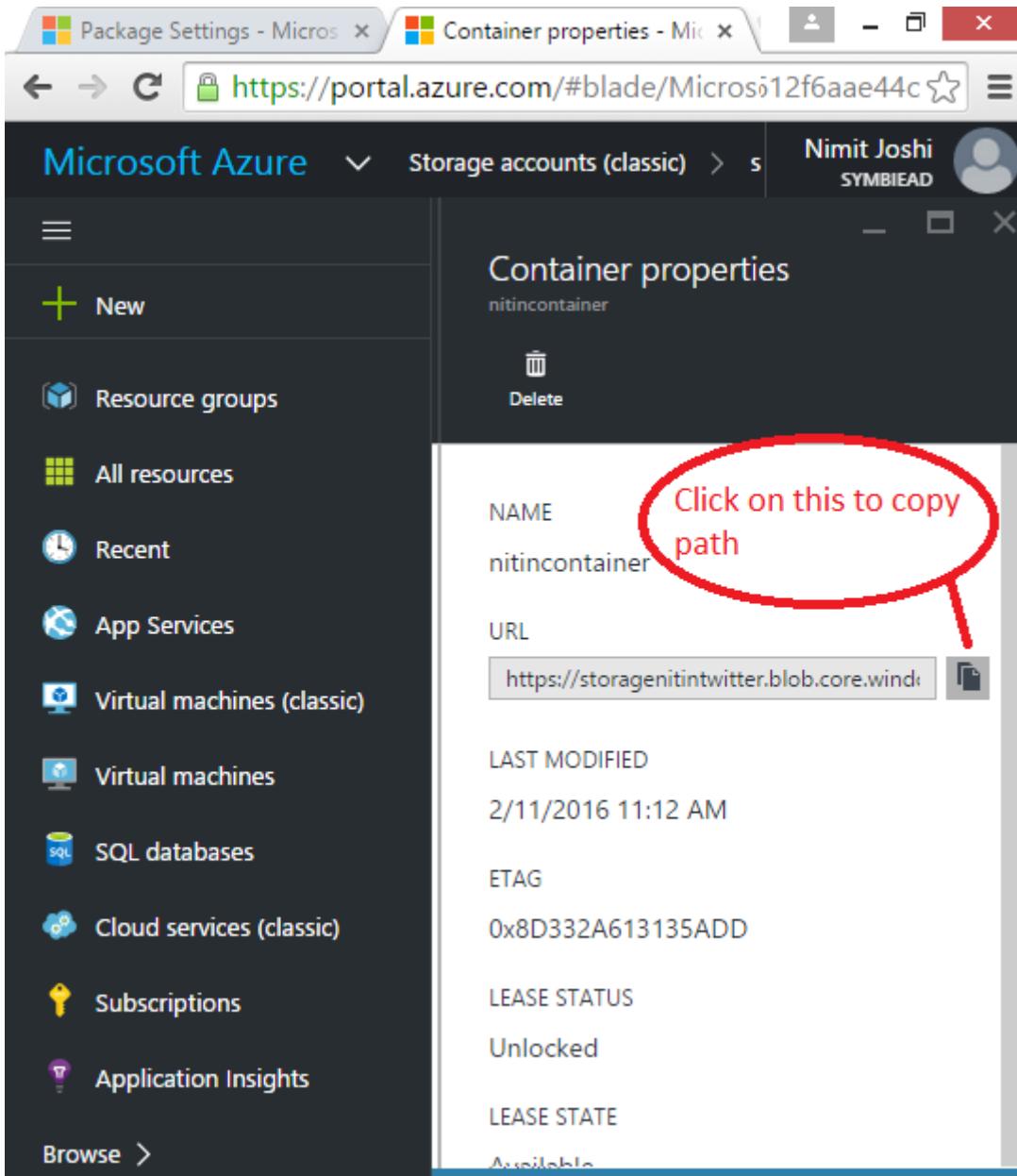
After clicking on “Blobs,” you will find your container; here my container is “**nitincontainer**.” Whenever I click on it, it will redirect to container properties.

Click on to “**Properties**.”



The screenshot shows the Microsoft Azure portal interface. The left sidebar lists various services: New, Resource groups, All resources, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Subscriptions, Application Insights, and Browse. The main content area is titled 'Blob service' under 'Storage accounts (classic) > storagenitintwitter > Blob service > nitincontainer'. A red box highlights the 'Properties' button in the top right of the container details. Another red box highlights the 'NAME' column in the blob list table, which contains one entry: '1 nitincontainer https://storagenitintwitter.bl... 2/11/2016, 11:12:05 AM ...'.

Copy the “URL” from container properties.

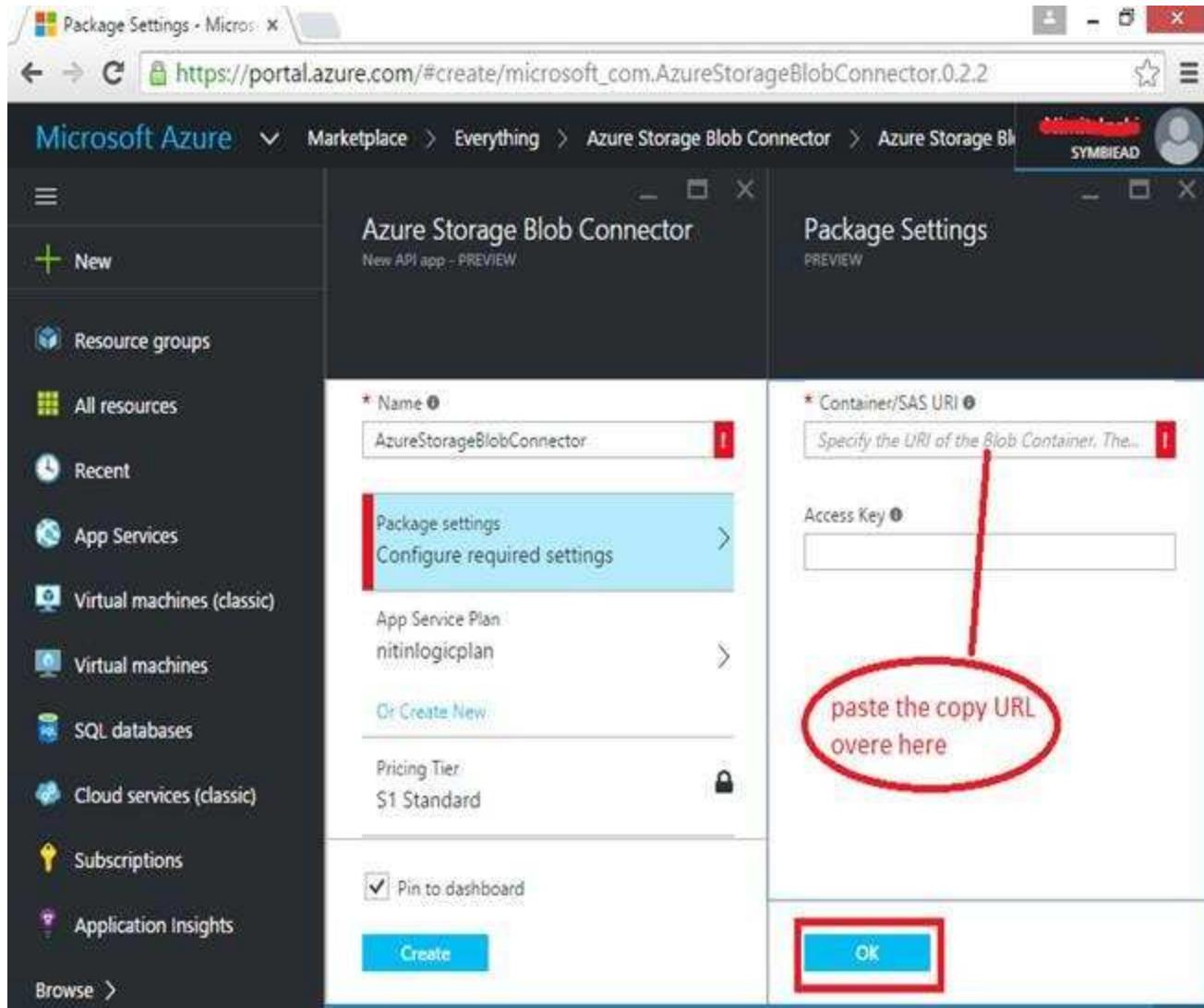


The screenshot shows the Microsoft Azure portal interface. The left sidebar lists various services: Resource groups, All resources, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Subscriptions, and Application Insights. The main content area is titled "Container properties" for a container named "nitincontainer". It displays the following details:

- NAME: nitincontainer
- URL: <https://storagenitintwitter.blob.core.windows.net> (this URL is circled in red with the annotation "Click on this to copy path")
- LAST MODIFIED: 2/11/2016 11:12 AM
- ETAG: 0x8D332A613135ADD
- LEASE STATUS: Unlocked
- LEASE STATE: Available

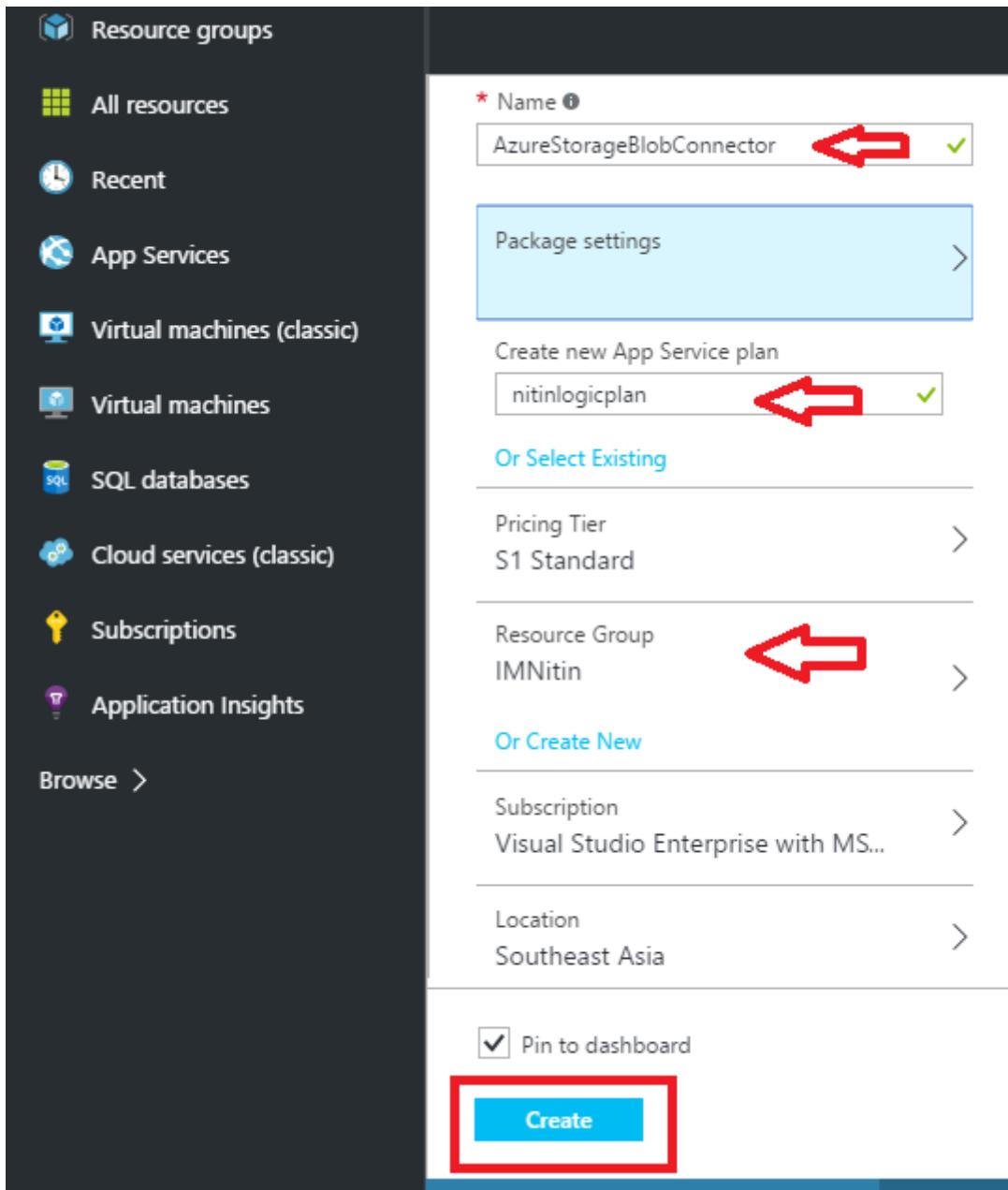
Paste the copy “URL” over here in the package settings.

Click on to “OK” button.



In the App service plan, give the service plan name; I have given “**nitinlogicplan**.” Select the “resource group” which is “IMNitin.”

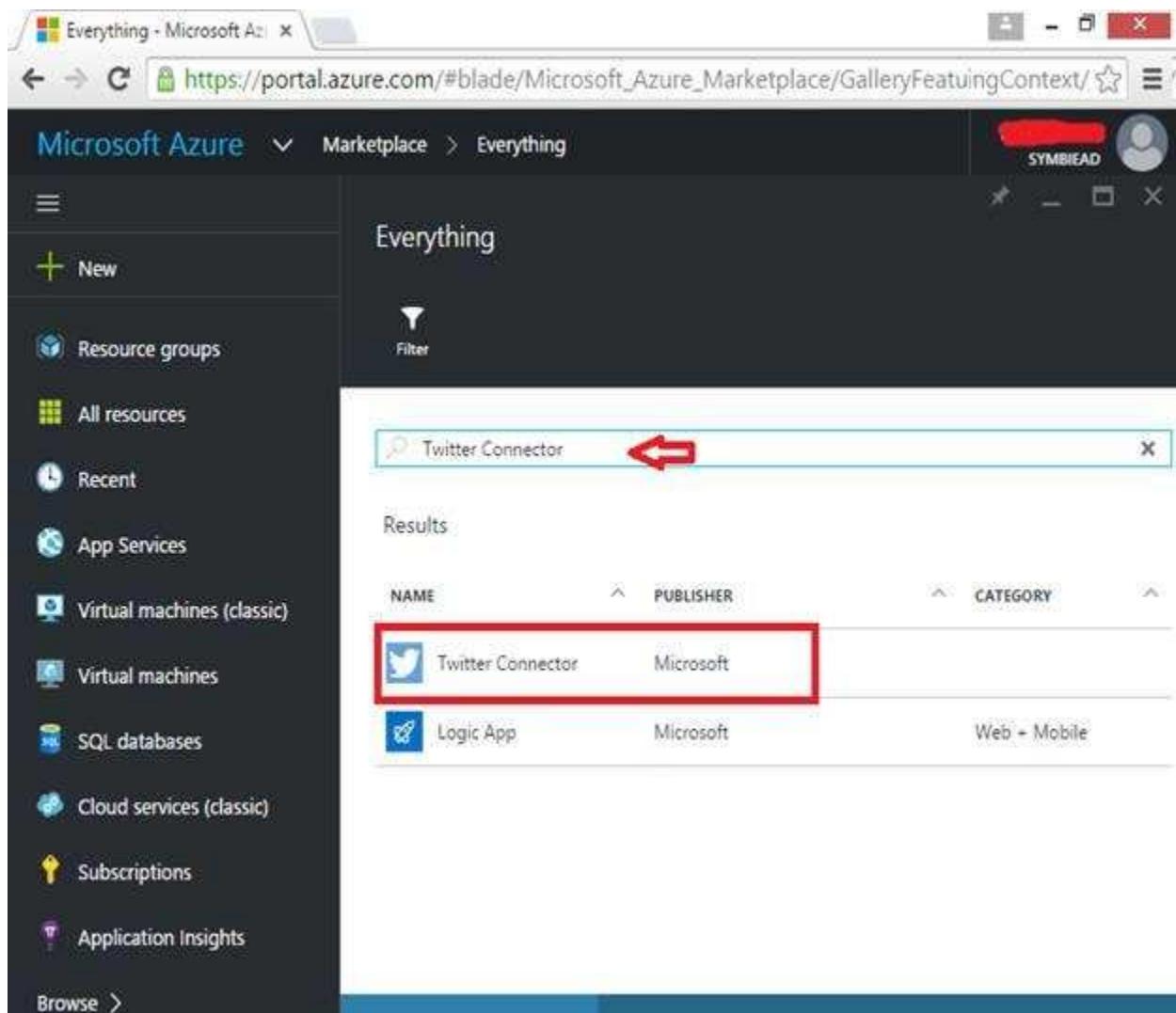
And click on “**Create**” button.



The screenshot shows the Azure portal's 'Resource groups' blade on the left and a configuration dialog on the right. The configuration dialog contains the following fields:

- Name: AzureStorageBlobConnector (highlighted with a red arrow)
- Package settings: nitinlogicplan (highlighted with a red arrow)
- Create new App Service plan: nitinlogicplan
- Or Select Existing
- Pricing Tier: S1 Standard
- Resource Group: IMNitIn (highlighted with a red arrow)
- Or Create New
- Subscription: Visual Studio Enterprise with MS...
- Location: Southeast Asia
- Pin to dashboard: checked
- Create button (highlighted with a red box)

Now, I need to create our “Twitter Connector,” for it go to “Browse” and type “Marketplace.” Click “**Marketplace**”.

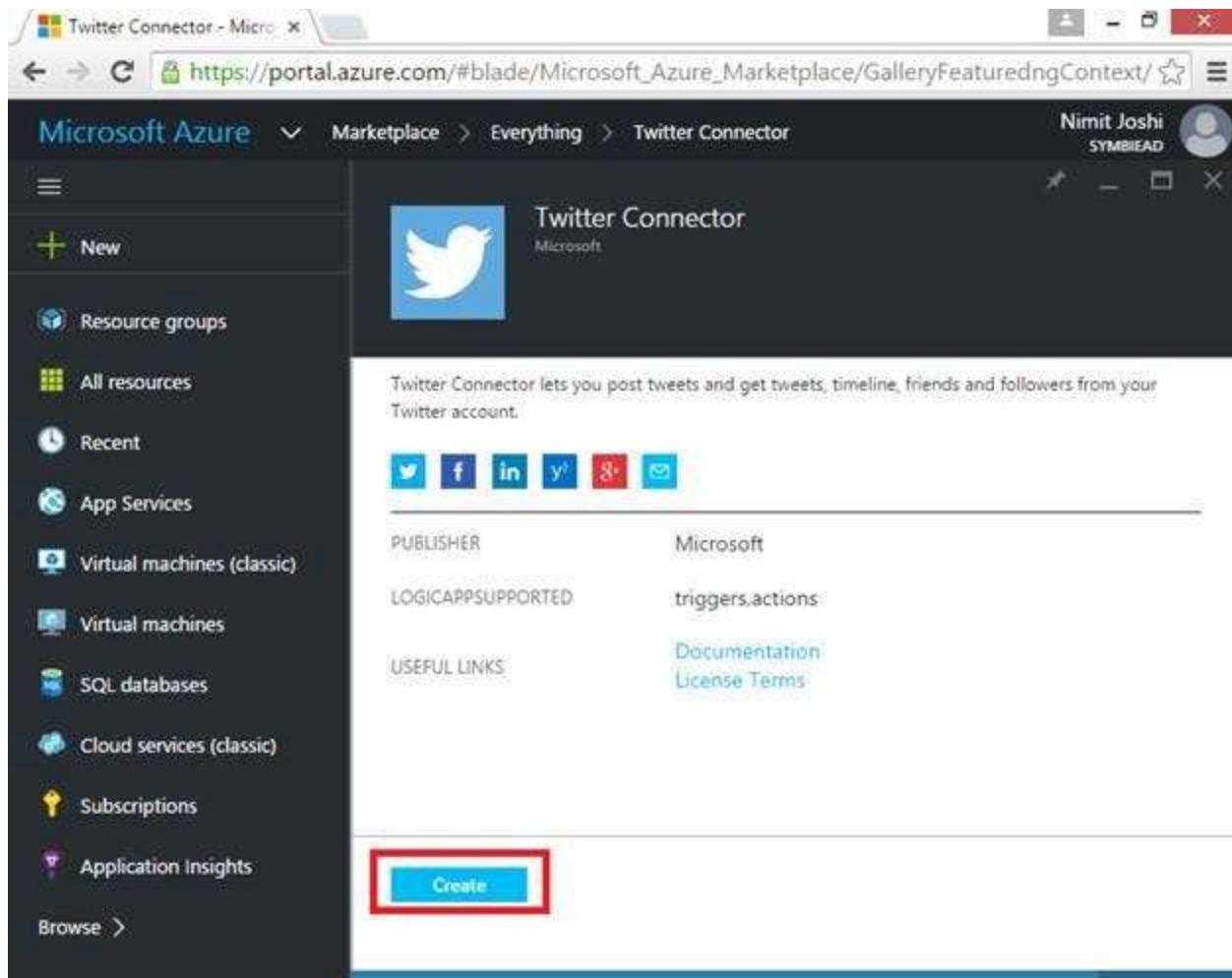


The screenshot shows the Microsoft Azure Marketplace interface. On the left, there's a sidebar with various navigation options like 'New', 'Resource groups', 'All resources', 'Recent', 'App Services', etc. The main area is titled 'Everything' and shows search results for 'Twitter Connector'. The results table has columns for NAME, PUBLISHER, and CATEGORY. The first result, 'Twitter Connector' by Microsoft, is highlighted with a red box and a red arrow points to the search bar above the results.

NAME	PUBLISHER	CATEGORY
Twitter Connector	Microsoft	
Logic App	Microsoft	Web + Mobile

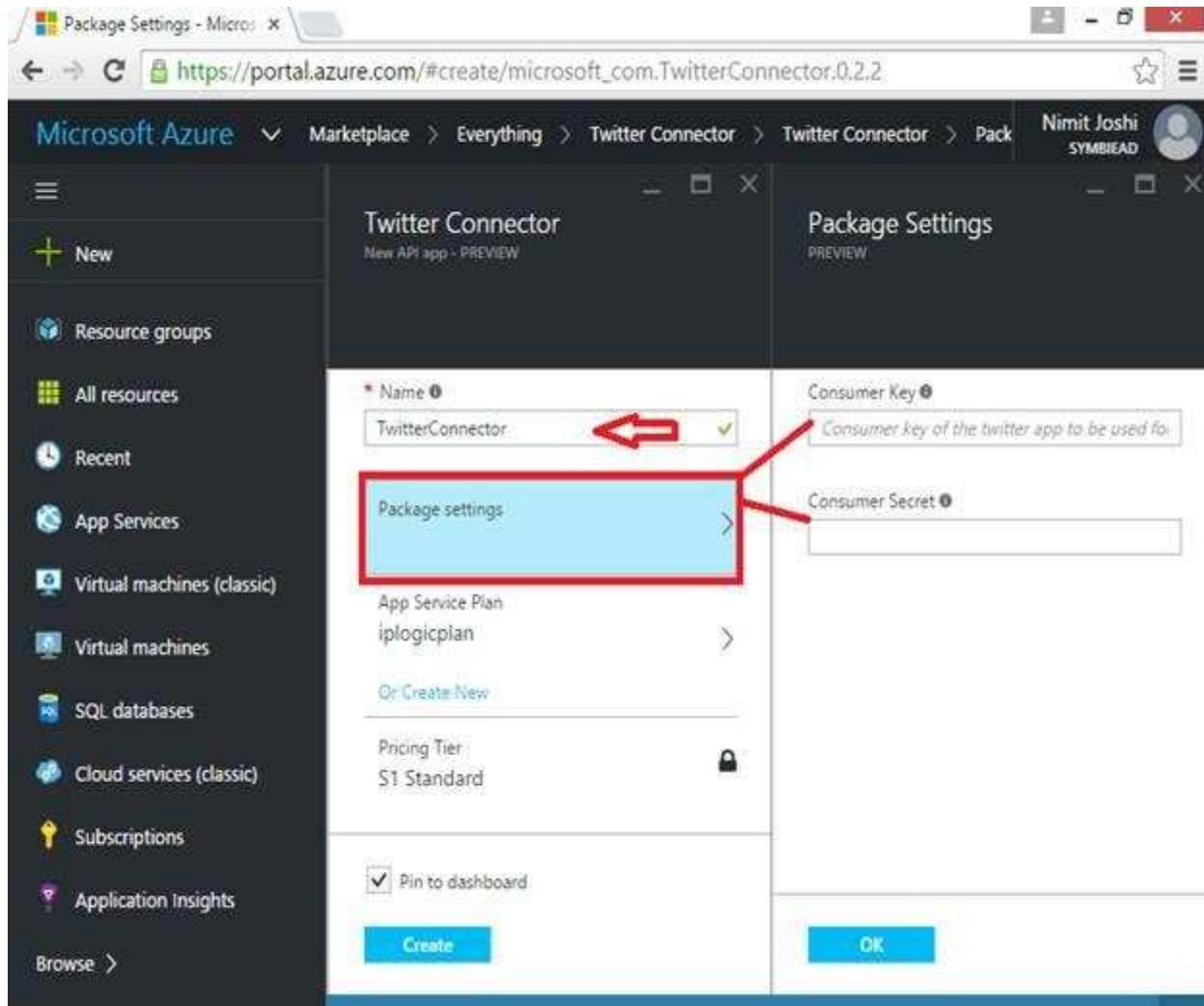
In search box, type "Twitter Connector."

You will find "Twitter Connector." Click on it as in the following:



The screenshot shows the Microsoft Azure Marketplace interface. On the left, there's a sidebar with various service icons like New, Resource groups, All resources, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Subscriptions, Application Insights, and a Browse link. The main content area is titled "Twitter Connector" by Microsoft. It features a large Twitter logo icon. Below the title, a brief description states: "Twitter Connector lets you post tweets and get tweets, timeline, friends and followers from your Twitter account." Underneath the description are several social media sharing icons. To the right of the description, there are two columns of metadata: PUBLISHER (Microsoft) and LOGICAPP SUPPORTED (triggers.actions). Below these, there are links for USEFUL LINKS: Documentation and License Terms. At the bottom of the page, there is a prominent blue "Create" button, which is highlighted with a red rectangular box.

Click on “Create” button.

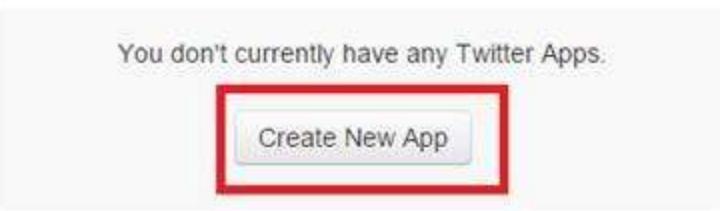


Here in the below figure, give the name for Connector; here I have given it “TwitterConnector.”

Next is “Package Settings,” after clicking on it you will be redirected to package settings window. This window will ask for “Consumer key” and “Consumer Secret.”



Twitter Apps



You don't currently have any Twitter Apps.

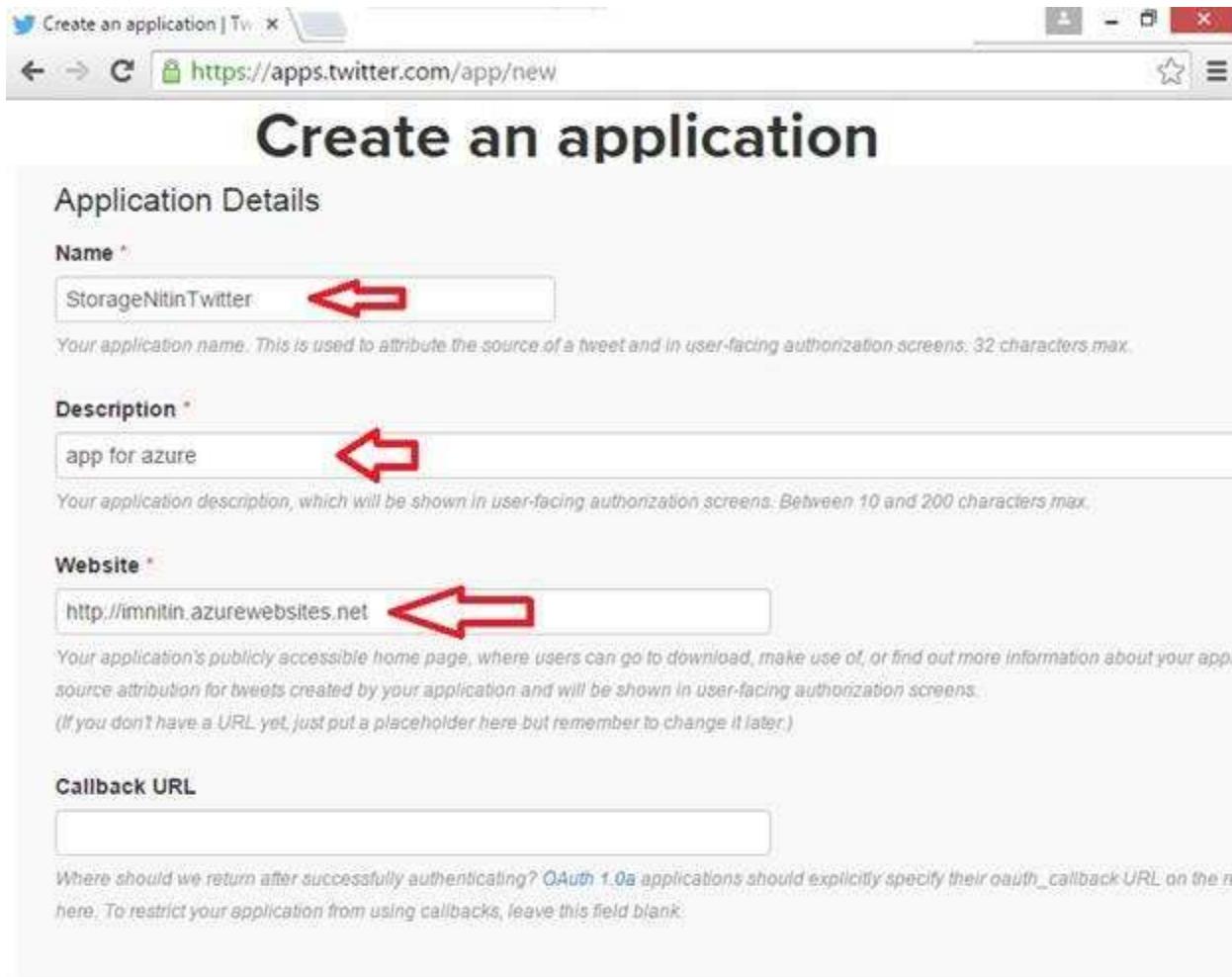
Create New App



For the “Consumer key” and “Consumer secret,” go to *the https://apps.twitter.com/*.

Note: Make sure for this process you should *log in* to your twitter account.

Click on to “**Create New App.**”



The screenshot shows the 'Create an application' form on the Twitter developer website. The 'Name' field contains 'StorageNitinTwitter'. The 'Description' field contains 'app for azure'. The 'Website' field contains 'http://imnitin.azurewebsites.net'. Red arrows point from the text descriptions to each of these three input fields.

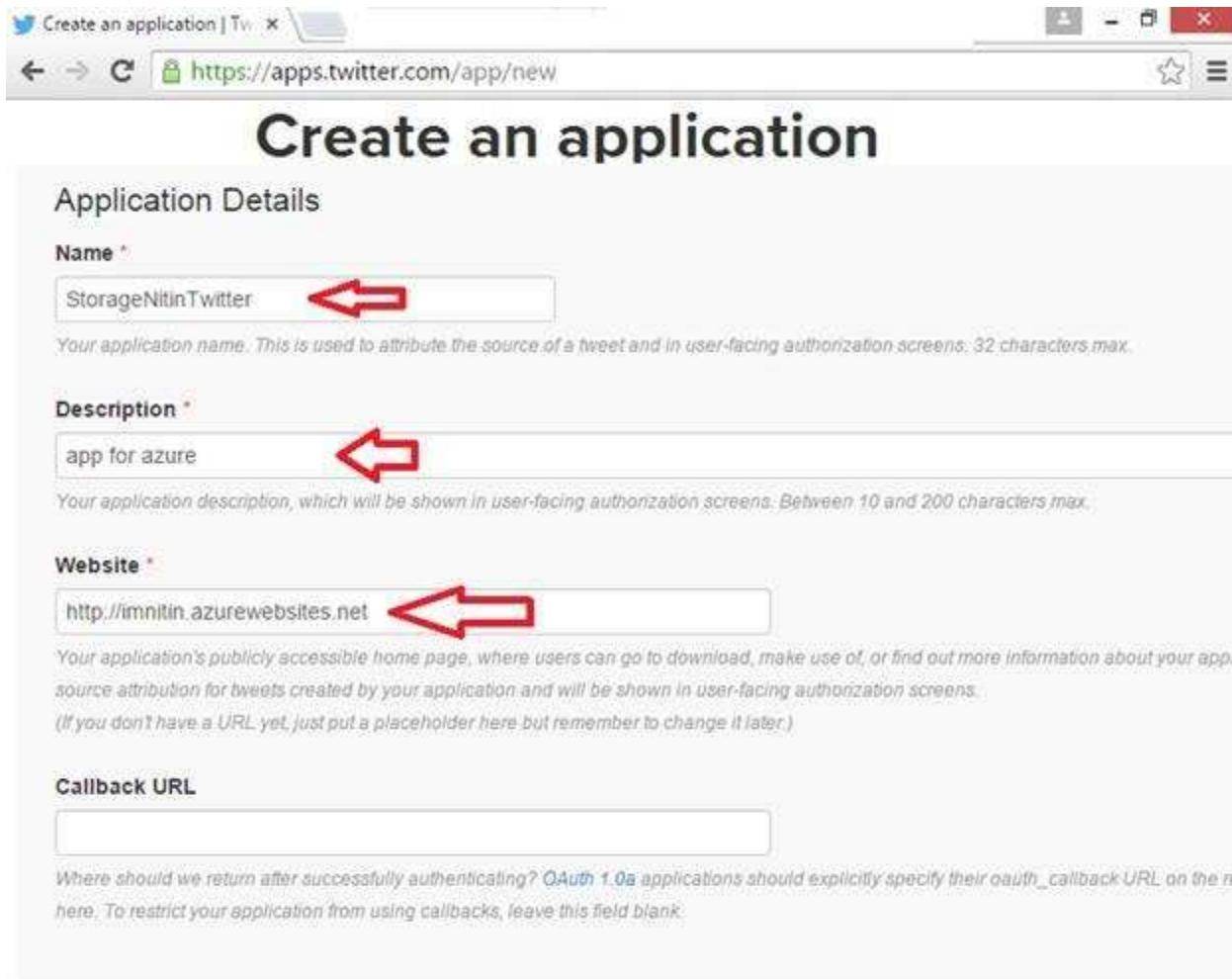
After clicking on button, you will see the below page, here I need to create an application, where we can find our “Consumer key” and “Consumer Secret.”

In the name section, I have given “StorageNitinTwitter.”

In the description section, you can give any script; I have given “app for Azure.”

In the “Website” section, give the temporary “URL” over here, your applications publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.

(If you don't have a URL yet, just put a placeholder here but remember to change it later.)



The screenshot shows the 'Create an application' page on the Twitter developer portal. The URL in the address bar is <https://apps.twitter.com/app/new>. The page has a header 'Create an application' and a section titled 'Application Details'. It includes fields for 'Name', 'Description', 'Website', and 'Callback URL'. Three red arrows point to the 'Name', 'Description', and 'Website' input fields, which contain the values 'StorageNitinTwitter', 'app for azure', and 'http://imnitin.azurewebsites.net' respectively. Below each field is a descriptive placeholder text.

Name *
StorageNitinTwitter

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *
app for azure

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *
http://imnitin.azurewebsites.net

*Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This URL is also used for source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)*

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the registration screen here. To restrict your application from using callbacks, leave this field blank.

Check on "Yes," and then click "**Create your Twitter application.**"

Developer Agreement

1. the Twitter Developer Policy located at <https://dev.twitter.com/overview/terms/policy> ("Developer Policy");
2. as it relates to your display of any of the Content, the Display Requirements located at <https://dev.twitter.com/terms/display-requirements> ("Display Requirements"); and
3. as it relates to your use and display of the Twitter Marks, the Twitter Brand Assets and Guidelines located at <https://twitter.com/logo> ("Brand Guidelines").

The Developer Policy, Display Requirements, and Brand Guidelines are collectively referred to herein as the "Developer Terms". You agree to the Developer Terms, which are hereby incorporated by reference and are available in hardcopy upon request to Twitter. In the event of a conflict between the Developer Terms and this Agreement, this Agreement shall control. None of the Developer Terms expand or extend the license to the Twitter API, Content or Twitter Marks granted in this Agreement.

II. Restrictions on Use of Licensed Materials

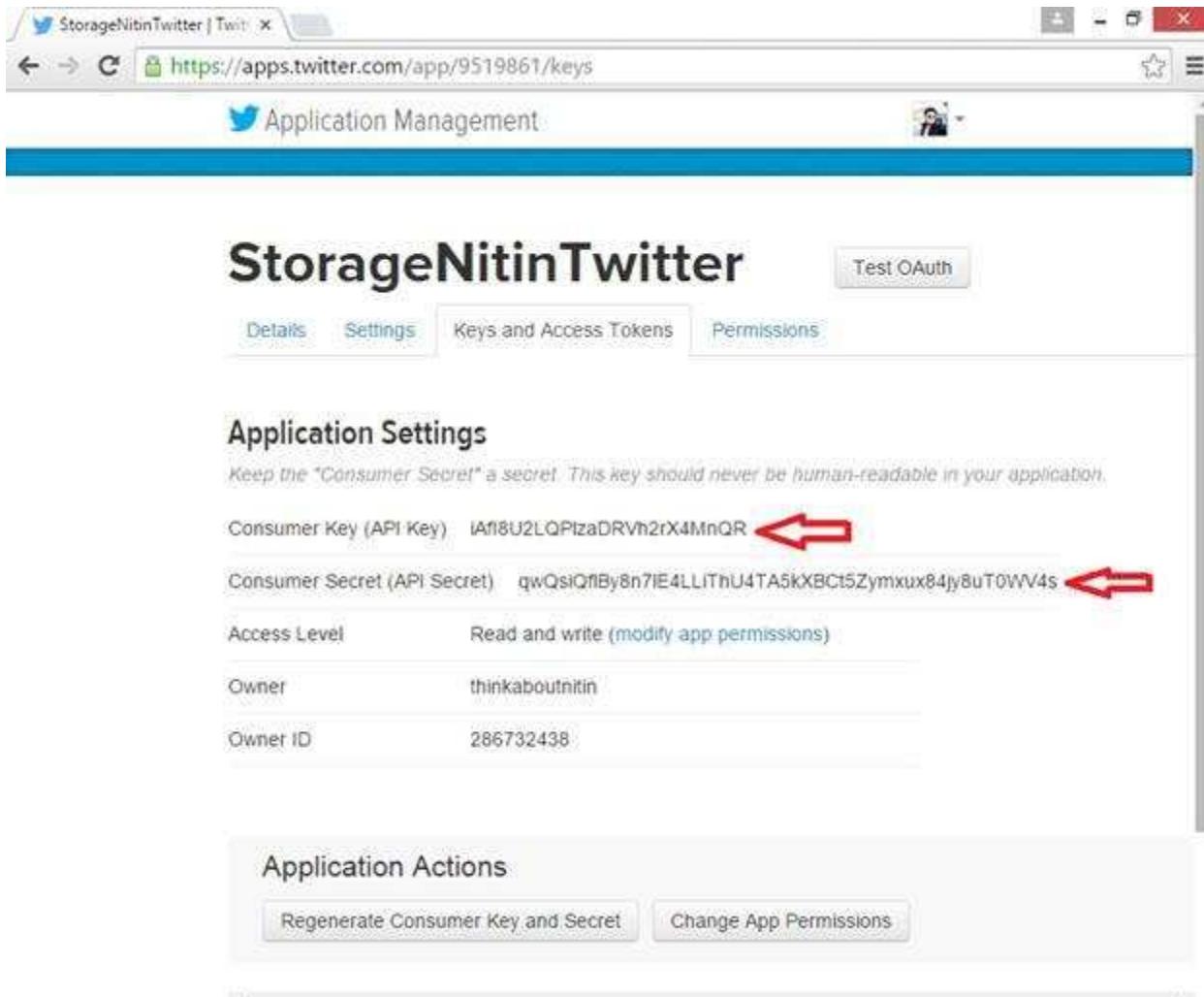
A. Reverse Engineering and other Limitations. You will not or attempt to (and will not allow others to) 1) reverse engineer.

Yes, I agree

[Create your Twitter application](#)



I have successfully created an application of twitter, click on "**Keys and Access Tokens**" on the top of the page, and copy the "Consumer Key" and "Consumer Secret" from here.



The screenshot shows the Twitter Application Management interface for the application "StorageNitinTwitter". The "Keys and Access Tokens" tab is selected. Two fields, "Consumer Key (API Key)" and "Consumer Secret (API Secret)", are highlighted with red arrows pointing to them.

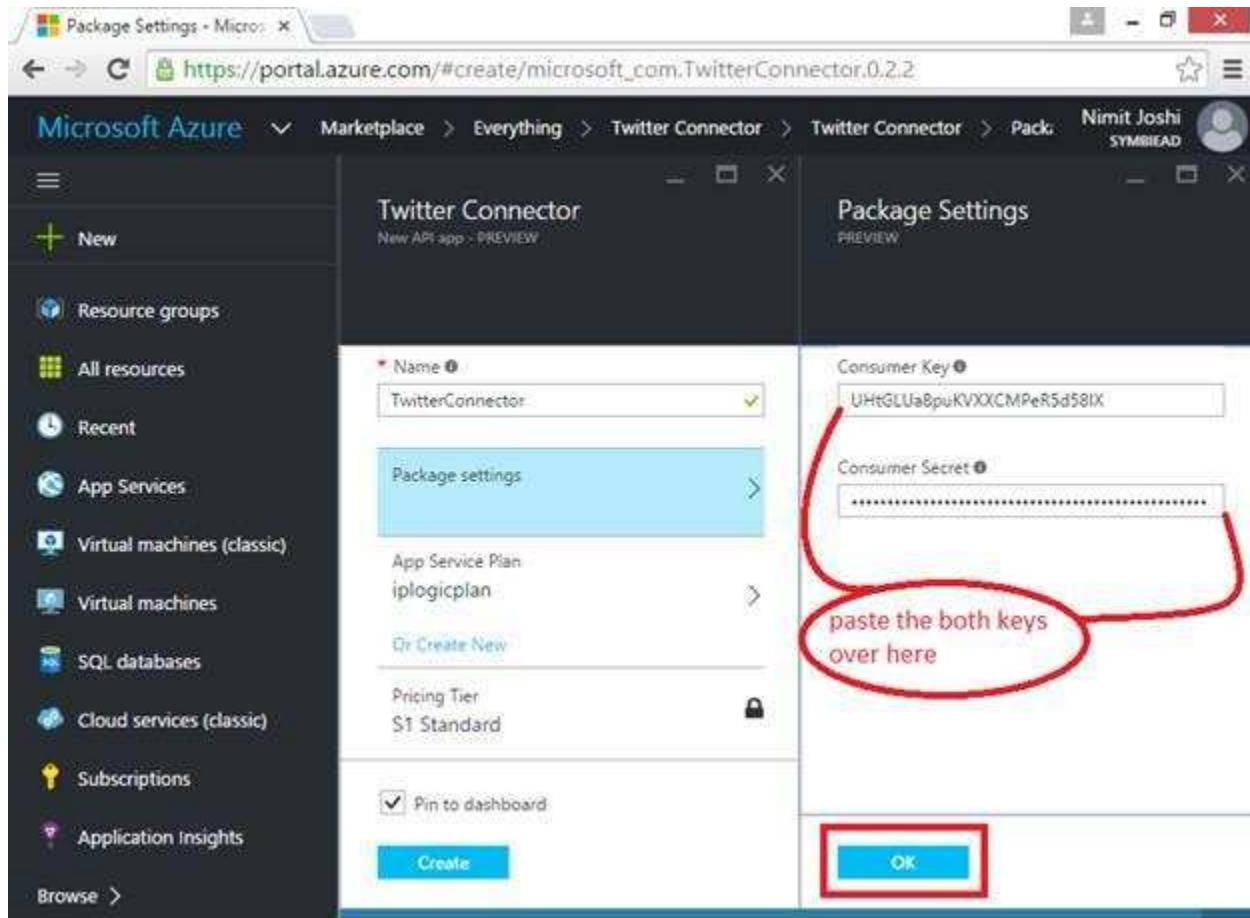
Setting	Value
Consumer Key (API Key)	IAfI8U2LQPlzaDRVh2rX4MnQR
Consumer Secret (API Secret)	qwQsiQflBy8n7IE4LLiThU4TA5kXBCt5Zymxux84jy8uT0WV4s
Access Level	Read and write (modify app permissions)
Owner	thinkaboutnitin
Owner ID	286732438

Application Actions

[Regenerate Consumer Key and Secret](#) [Change App Permissions](#)

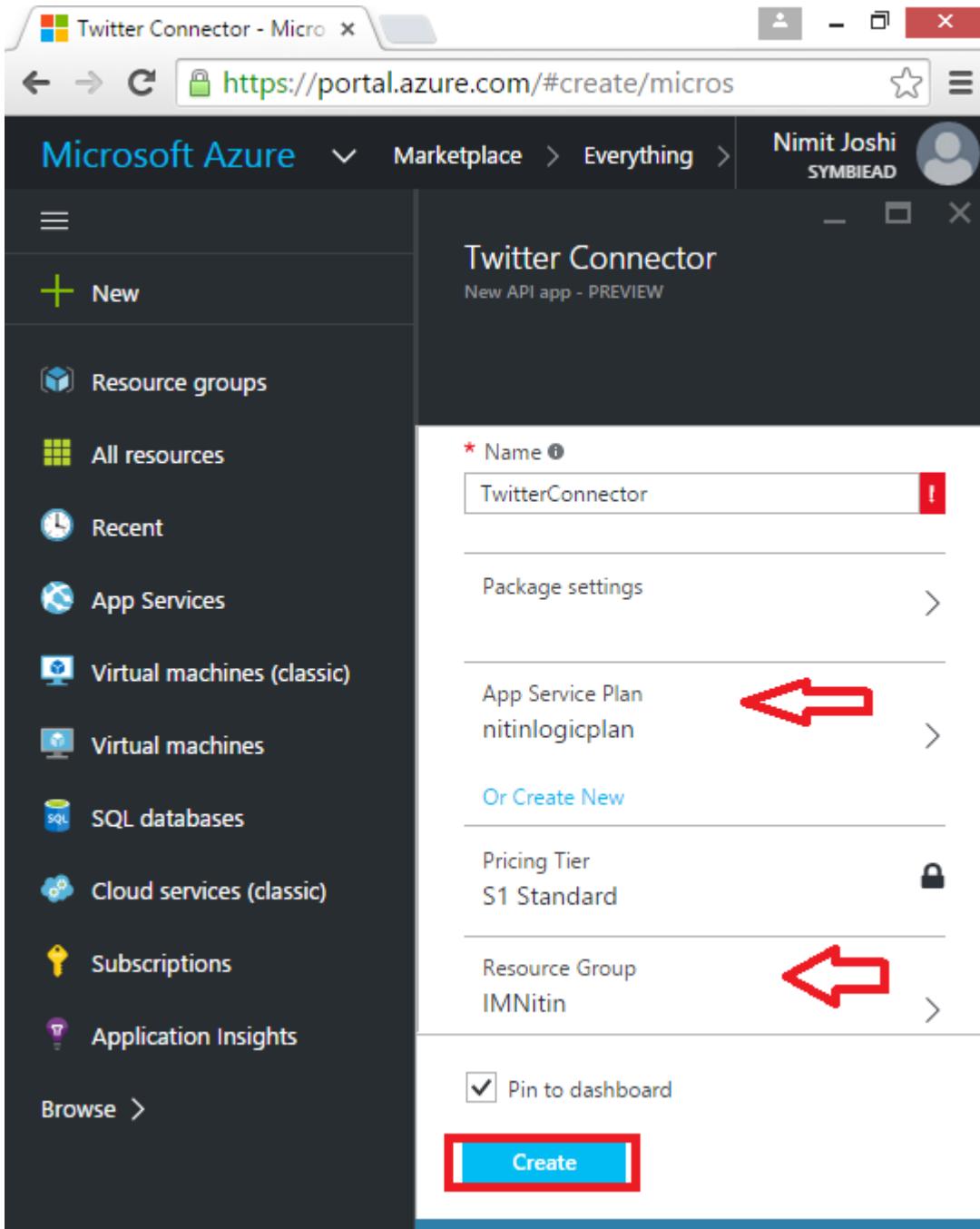
Paste both the keys over here in the “**Package Settings**.”

Click on “**OK**” button.



Select the app logic plan which is “**nitinlogicplan**,” and select the resource group which is “**IMNitin**.”

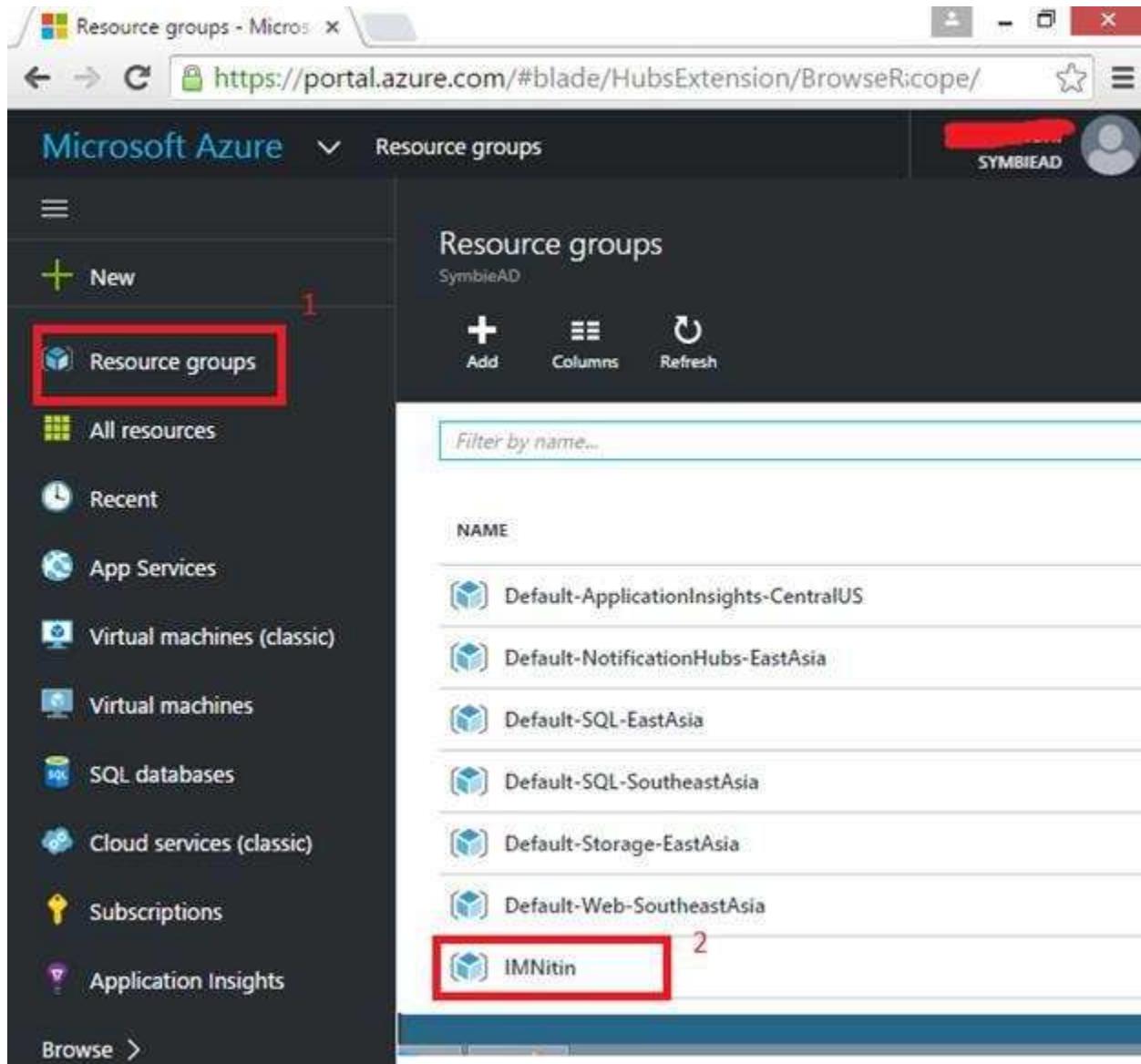
Click on “**Create**” button.



The screenshot shows the Microsoft Azure portal interface for creating a new API app named "TwitterConnector". The left sidebar lists various service categories like Resource groups, All resources, Recent, App Services, etc. The main panel displays the "Twitter Connector" creation form. Two red arrows point to specific fields: one to the "App Service Plan" dropdown set to "nitinlogicplan" and another to the "Resource Group" dropdown set to "IMNitin". Both fields have a right-pointing arrow indicating they can be modified.

So, I have successfully created my “Twitter Connector,” so now I need to copy the “URI” of the Twitter Connector and provide in the Twitter Application, in “Website” and “Callback URL” section.

Go to the “Resource groups” and select the “IMNitin” resource.

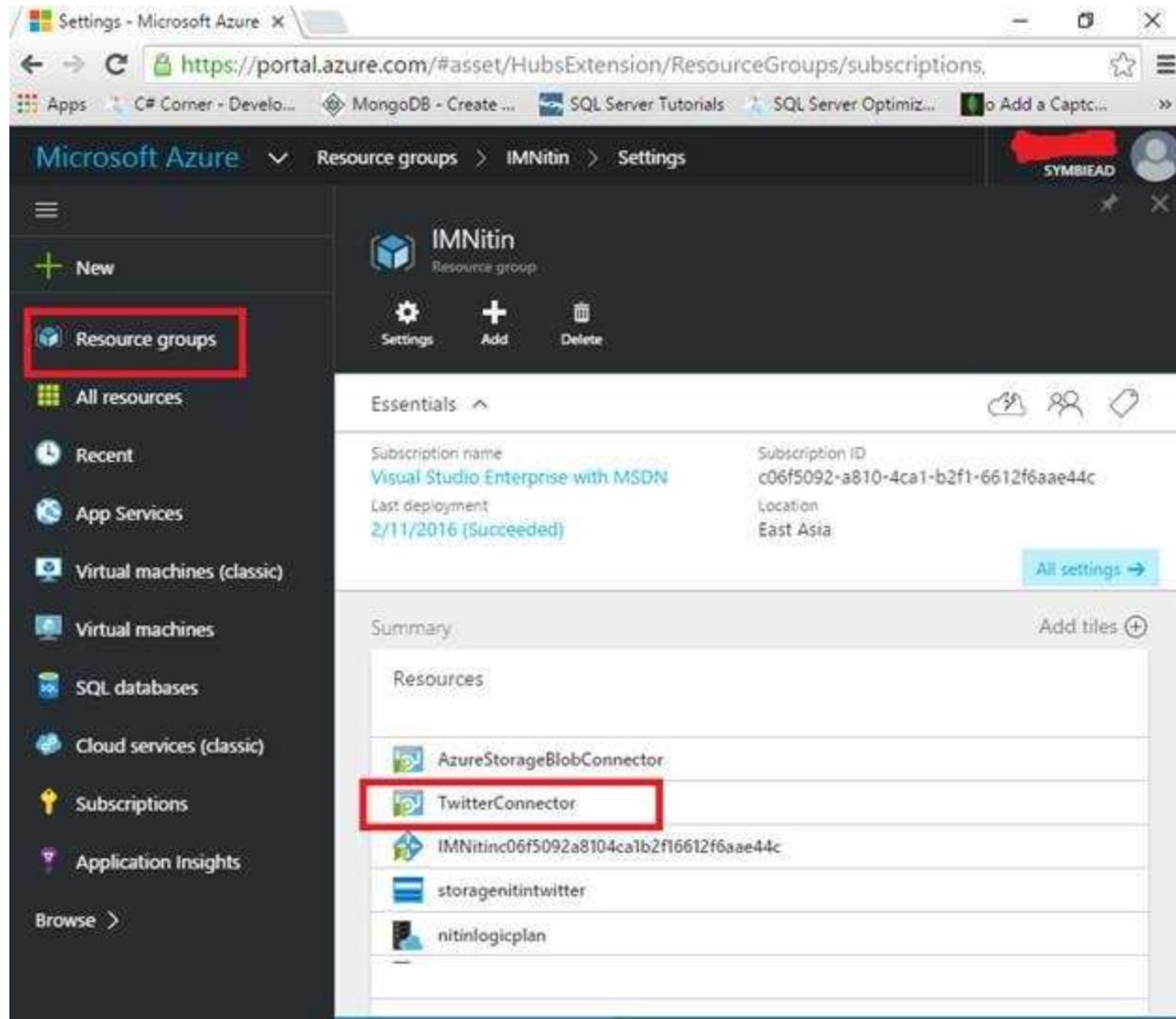


The screenshot shows the Microsoft Azure portal interface. The left sidebar has a red box around the 'Resource groups' item under the 'New' section. The main content area is titled 'Resource groups' and shows a list of resource groups with one named 'IMNitin' highlighted with a red box. The list includes:

NAME
Default-ApplicationInsights-CentralUS
Default-NotificationHubs-EastAsia
Default-SQL-EastAsia
Default-SQL-SoutheastAsia
Default-Storage-EastAsia
Default-Web-SoutheastAsia
IMNitin

Here are all my apps in one resource, “IMNitin” named resource.

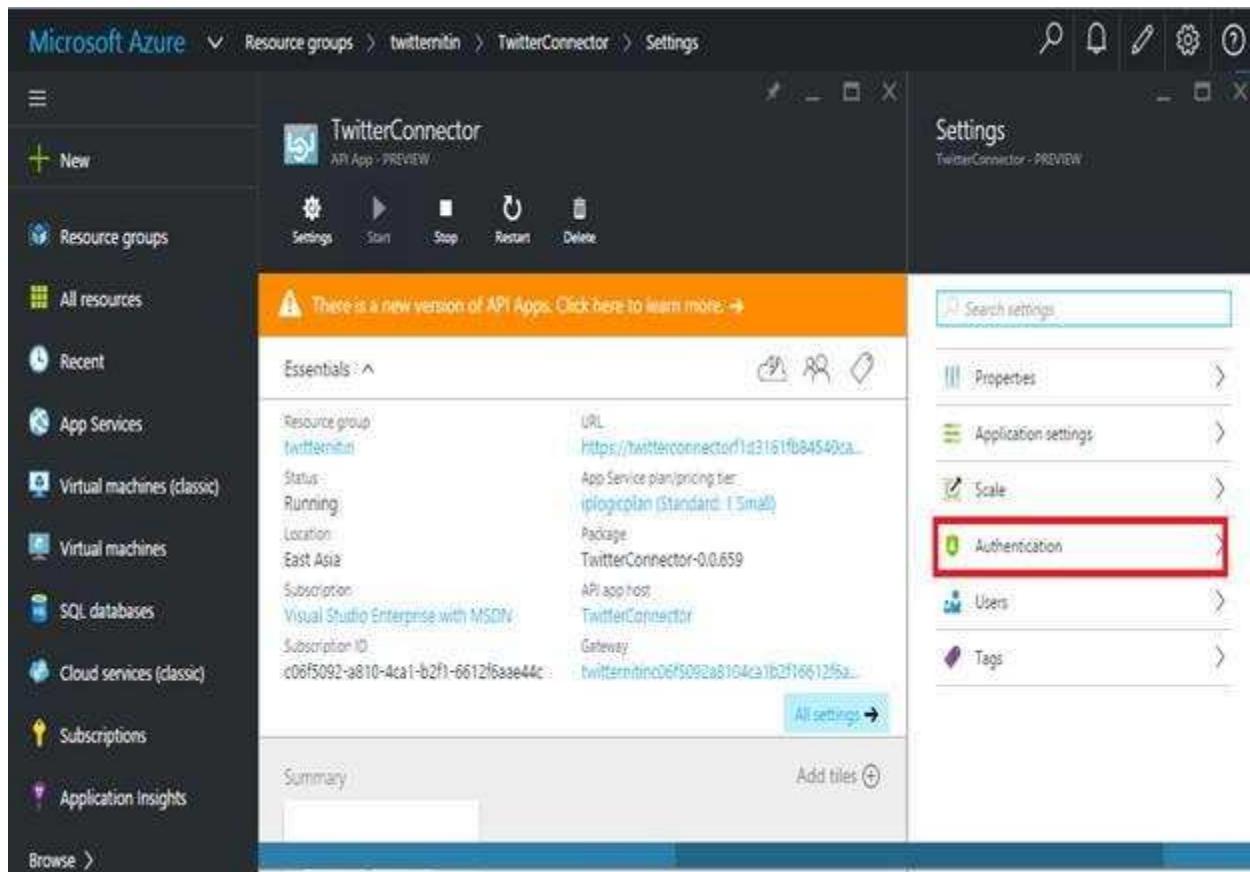
For the “URL”, click on to “Twitter Connector”.



The screenshot shows the Microsoft Azure portal interface. The URL in the browser is <https://portal.azure.com/#asset/HubsExtension/ResourceGroups/subscriptions>. The page title is "Settings - Microsoft Azure". The breadcrumb navigation shows "Microsoft Azure > Resource groups > IMNitin > Settings". On the left sidebar, under "Resource groups", there is a red box around the "Resource groups" link. In the main content area, under "Essentials", it shows the "Subscription name" as "Visual Studio Enterprise with MSDN", "Last deployment" as "2/11/2016 (Succeeded)", "Subscription ID" as "c06f5092-a810-4ca1-b2f1-6612f6aae44c", and "Location" as "East Asia". There is a blue button "All settings" with a right arrow. Below this, under "Summary", there is a table titled "Resources" with the following items:

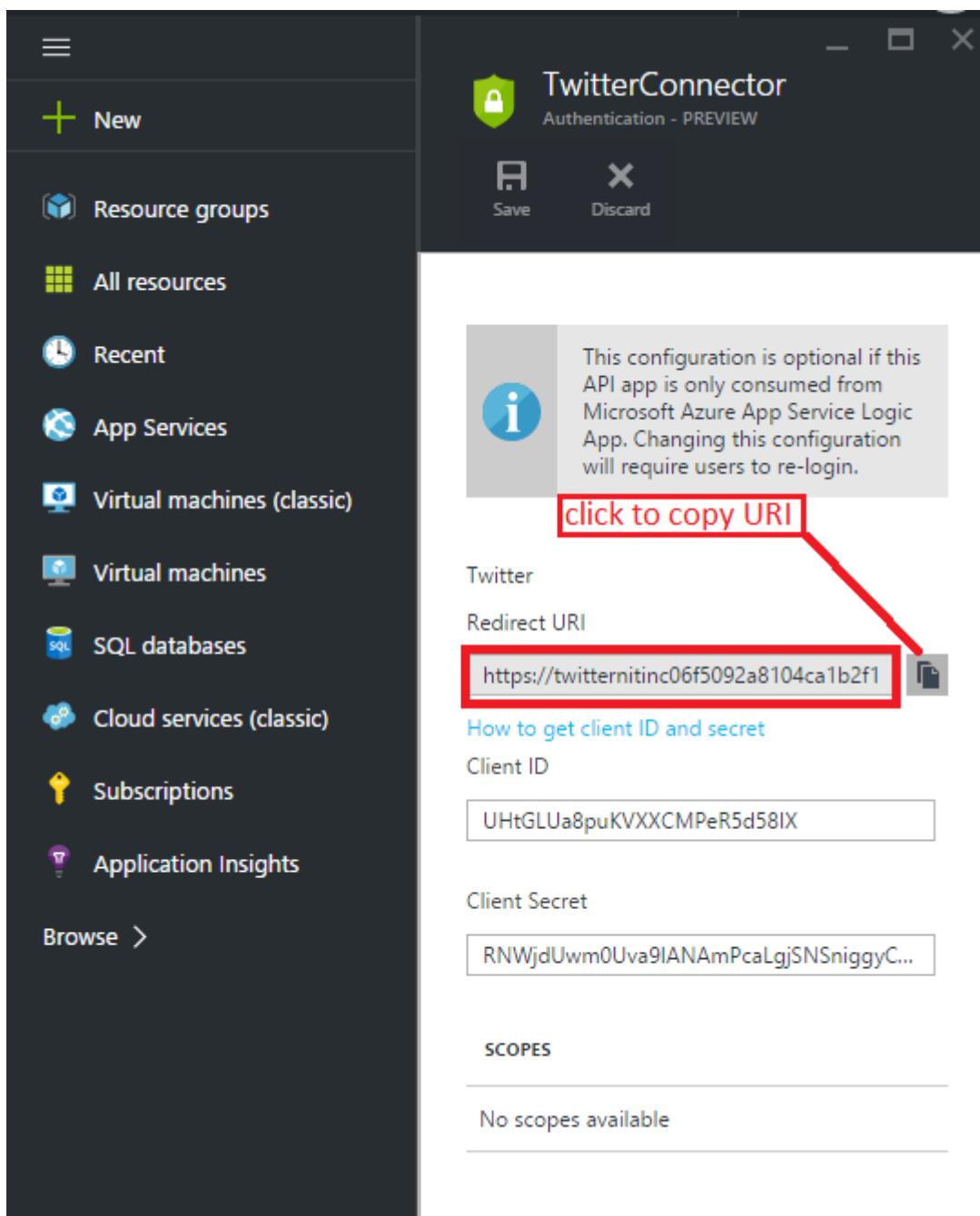
Resource
AzureStorageBlobConnector
TwitterConnector
IMNitinc06f5092a8104ca1b2f16612f6aae44c
storagenitintwitter
nitinlogicplan

Click on to “Authentication.”

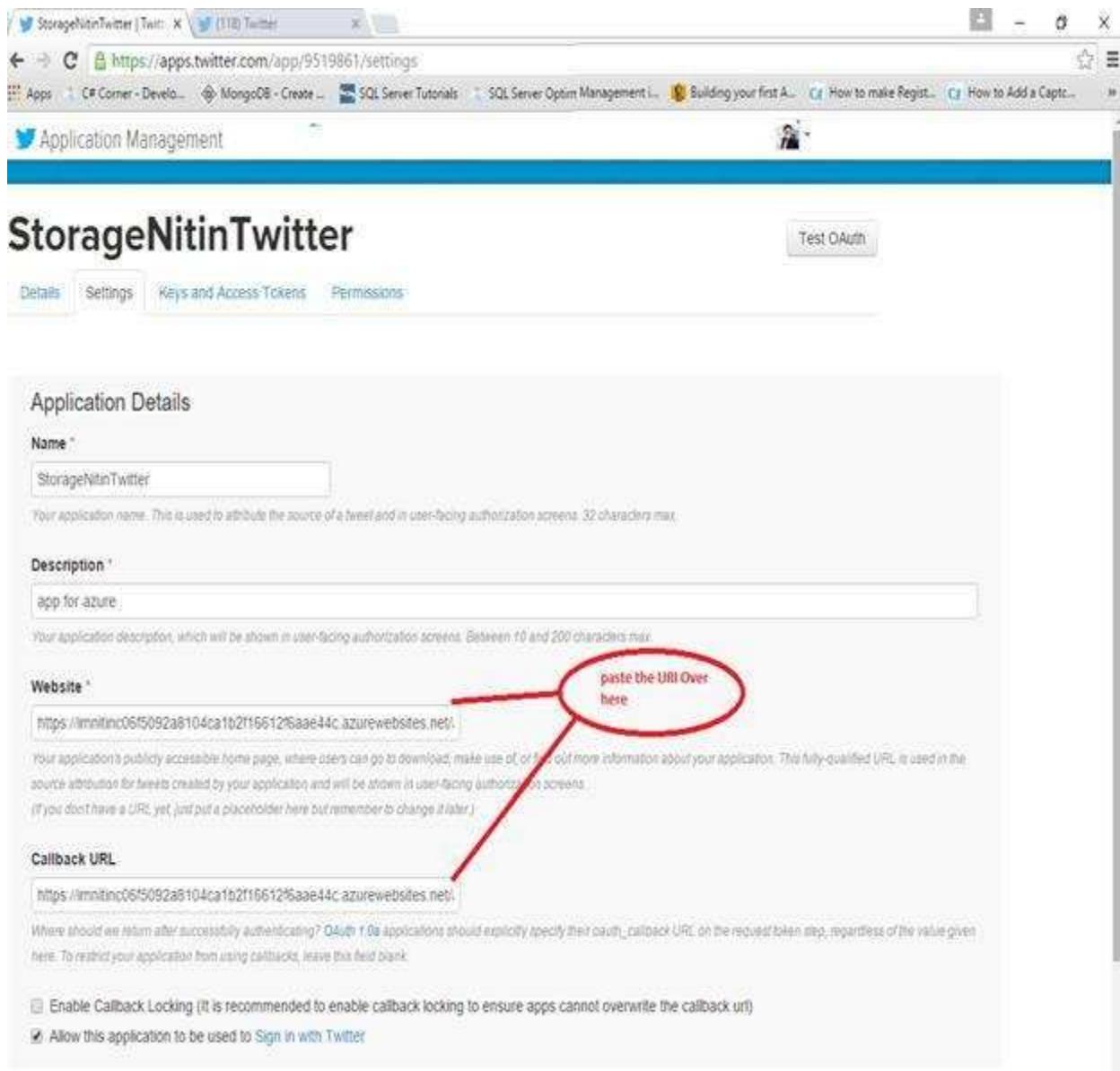


The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various service icons like App Services, Virtual machines, SQL databases, etc. The main area shows a resource group named 'twitterinit' containing an 'API App - PREVIEW' named 'TwitterConnector'. A prominent orange banner at the top says 'There is a new version of API Apps. Click here to learn more.' Below it, the 'Essentials' section provides basic details: Resource group (twitterinit), Status (Running), Location (East Asia), Subscription (Visual Studio Enterprise with MSDN), and Subscription ID (c06f5092-a810-4ca1-b2f1-66126aae44c). To the right, the 'Settings' pane for 'TwitterConnector - PREVIEW' is open, showing sections for Properties, Application settings, Scale, Authentication (which is highlighted with a red box), Users, and Tags.

Copy the “**Redirect URI**” from here.



Paste the “URI” in “WebSite” and “Callback URL” in the twitter application page.



StorageNitinTwitter

Application Details

Name * StorageNitinTwitter

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description * app for azure

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website * https://lmnitin06f5092a8104ca1b2f1f6612%6aae44c.azurewebsites.net/

Your application's publicly accessible home page, where users can go to download, make use of, or learn more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

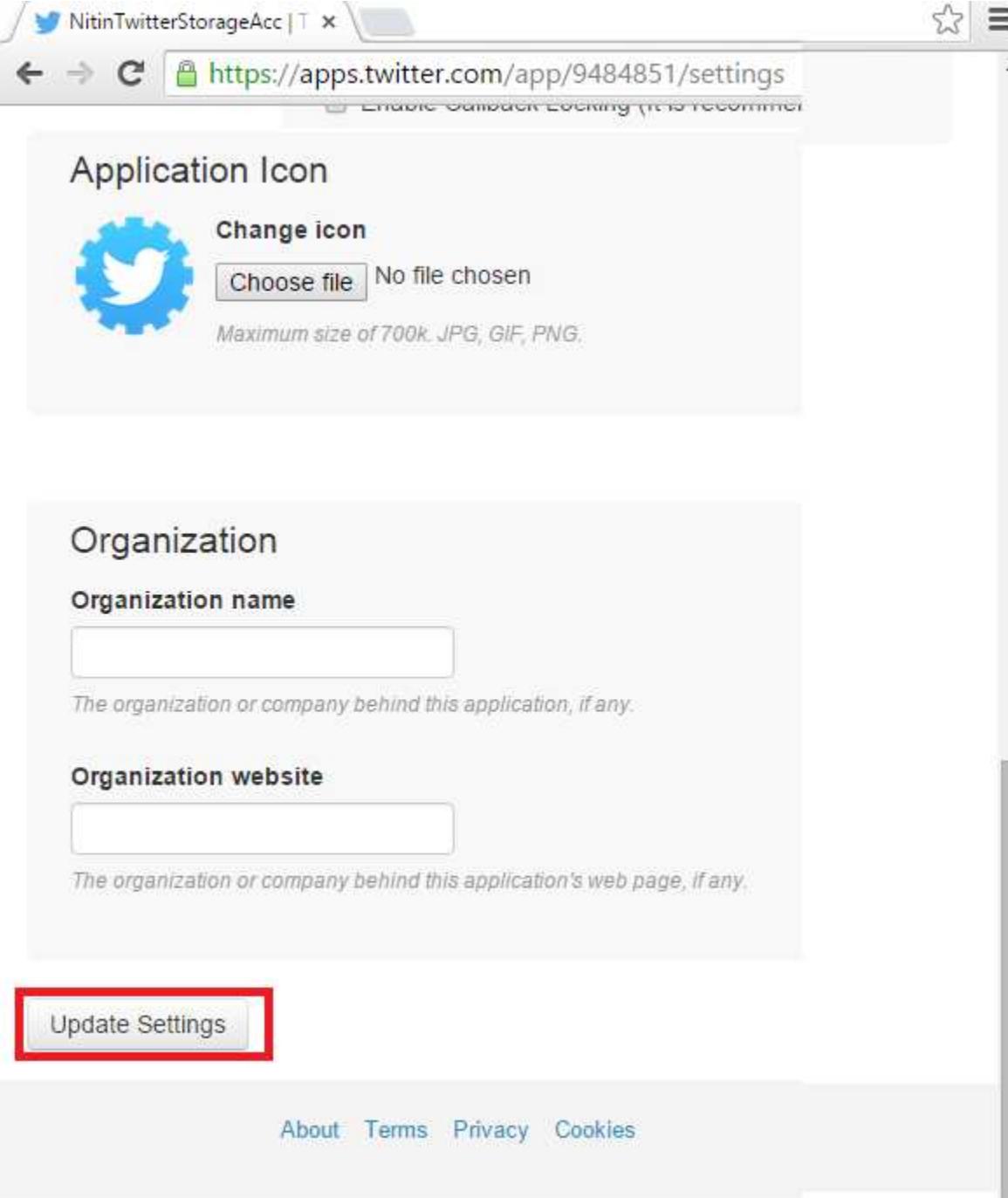
Callback URL https://lmnitin06f5092a8104ca1b2f1f6612%6aae44c.azurewebsites.net/

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Enable Callback Locking (It is recommended to enable callback locking to ensure apps cannot overwrite the callback url)

Allow this application to be used to Sign in with Twitter

Click on “**Update Settings**” button to save the details.

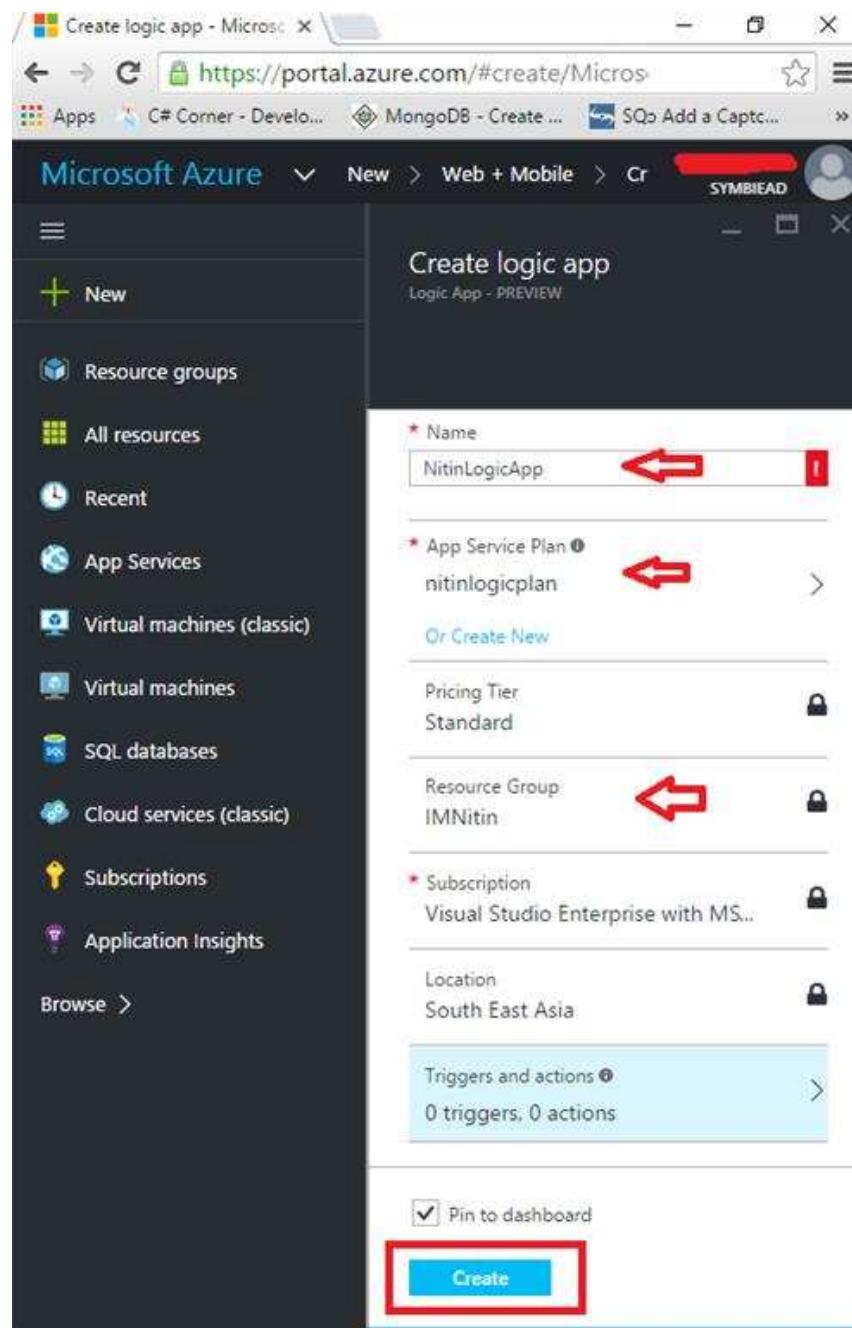


The screenshot shows the Twitter Application Settings page. At the top, the URL is https://apps.twitter.com/app/9484851/settings. The page has two main sections: "Application Icon" and "Organization". The "Application Icon" section contains a "Change icon" button with a Twitter logo, a "Choose file" button with "No file chosen", and a note about the maximum file size (700k, JPG, GIF, PNG). The "Organization" section contains fields for "Organization name" and "Organization website", both with placeholder text indicating they are for the organization behind the application. At the bottom, there is a "Update Settings" button, which is highlighted with a red box. A navigation bar at the bottom includes links for About, Terms, Privacy, and Cookies.

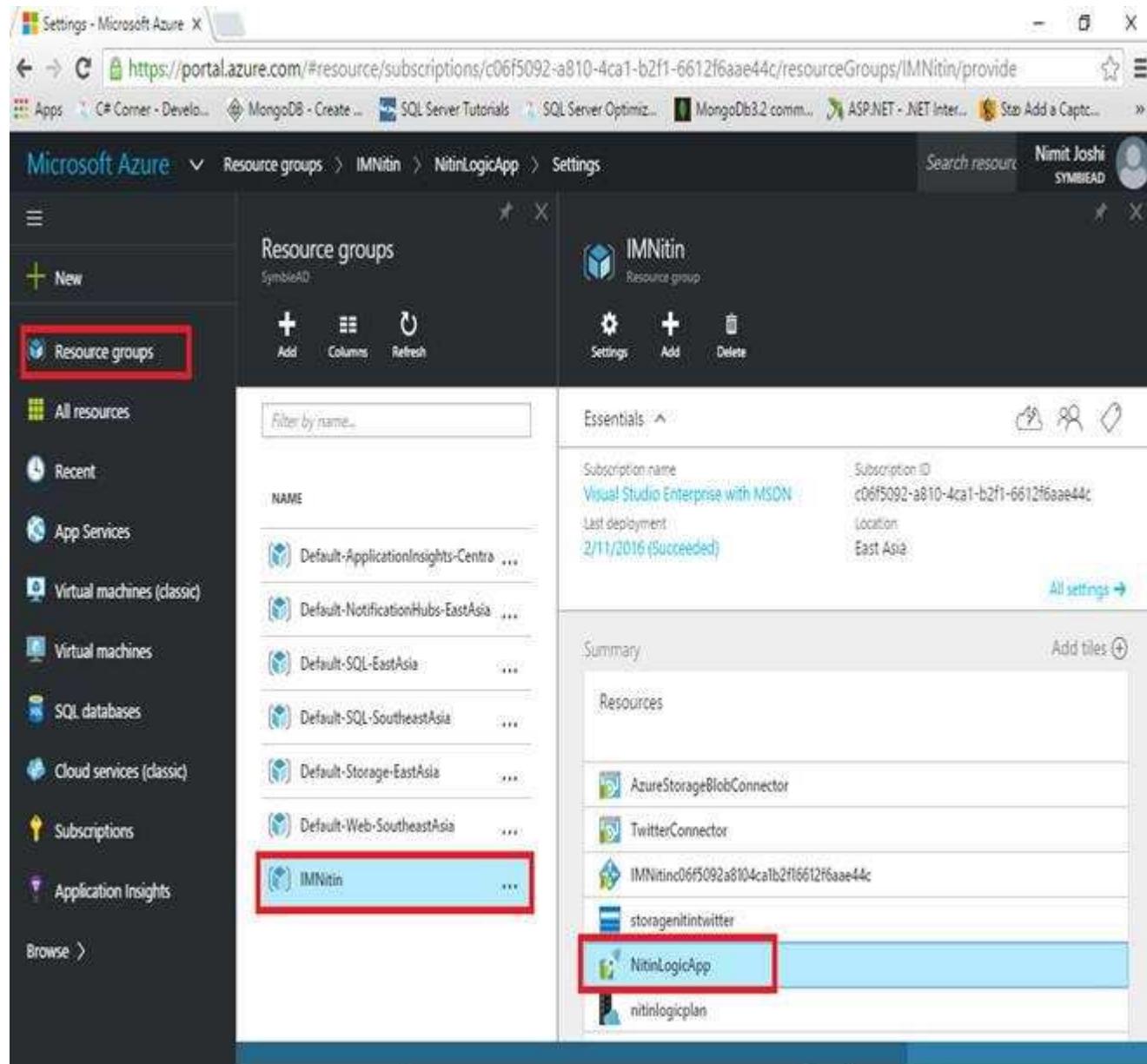
Now, I need to create logic app, so go to “New,” and select “Web+Mobile,” and select “Logic App.”

Give the name to your logic app; here I have given “**NitinLogicApp**” and select app service plan which is “**nitinlogicplan**,” select resource group which is “**IMNitin**.”

Click “**Create**” button.

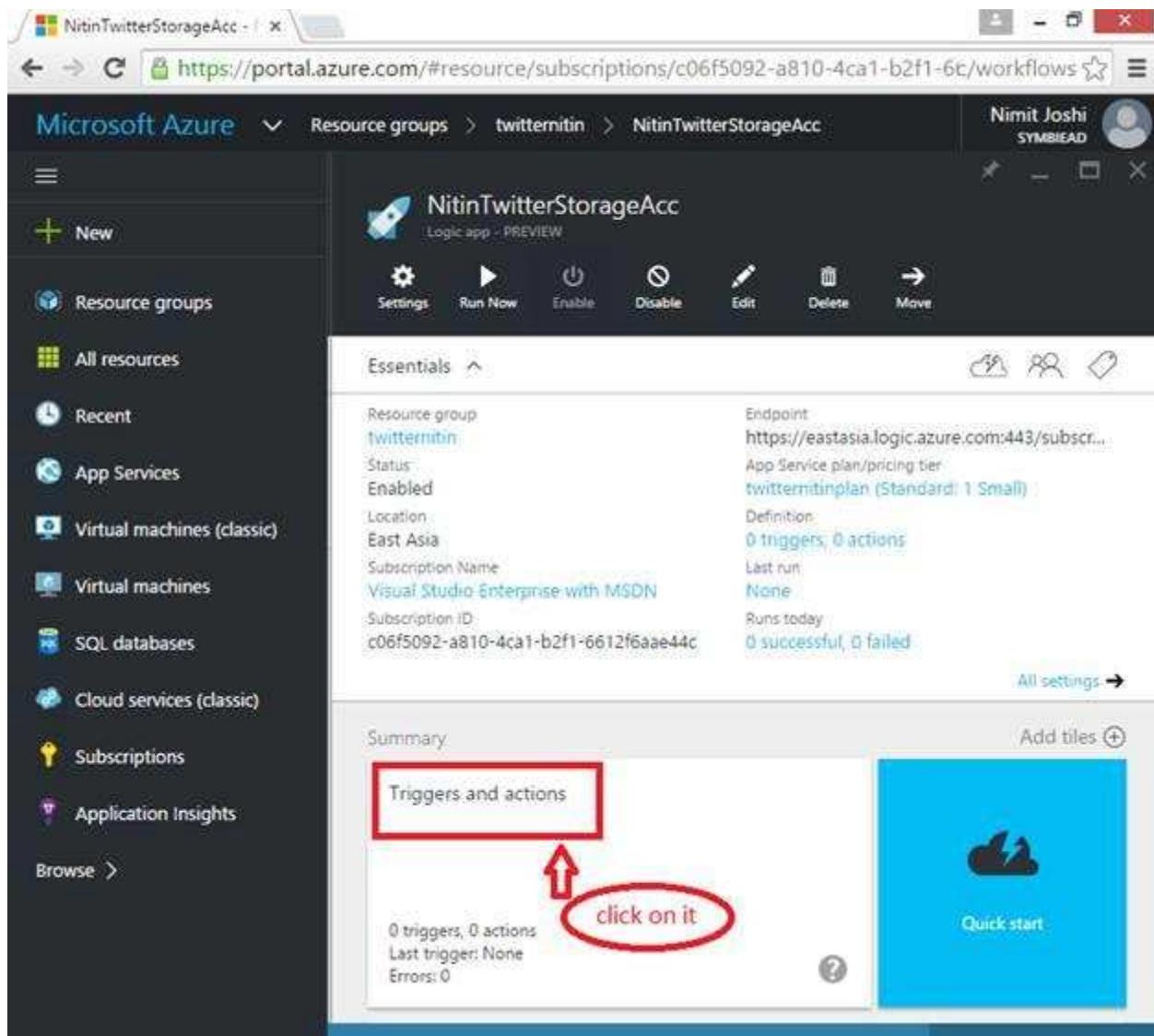


For creating Triggers and actions, go to the “resources groups,” click on resource group, which is “IMNitin,” and select “NitinLogicApp,” which is just created.



The screenshot shows the Microsoft Azure portal interface. The left sidebar has a 'Resource groups' section highlighted with a red box. The main pane displays the 'Resource groups' page for the 'IMNitin' resource group. The 'Essentials' section shows the subscription name 'Visual Studio Enterprise with MSDN' and the location 'East Asia'. The 'Resources' section lists several resources, including 'AzureStorageBlobConnector', 'TwitterConnector', 'IMNitinc06f5092a8104ca1b2f16612f6aae44c', 'storagenitwitter', and 'NitinLogicApp', which is also highlighted with a red box.

Click on “Triggers and actions.”

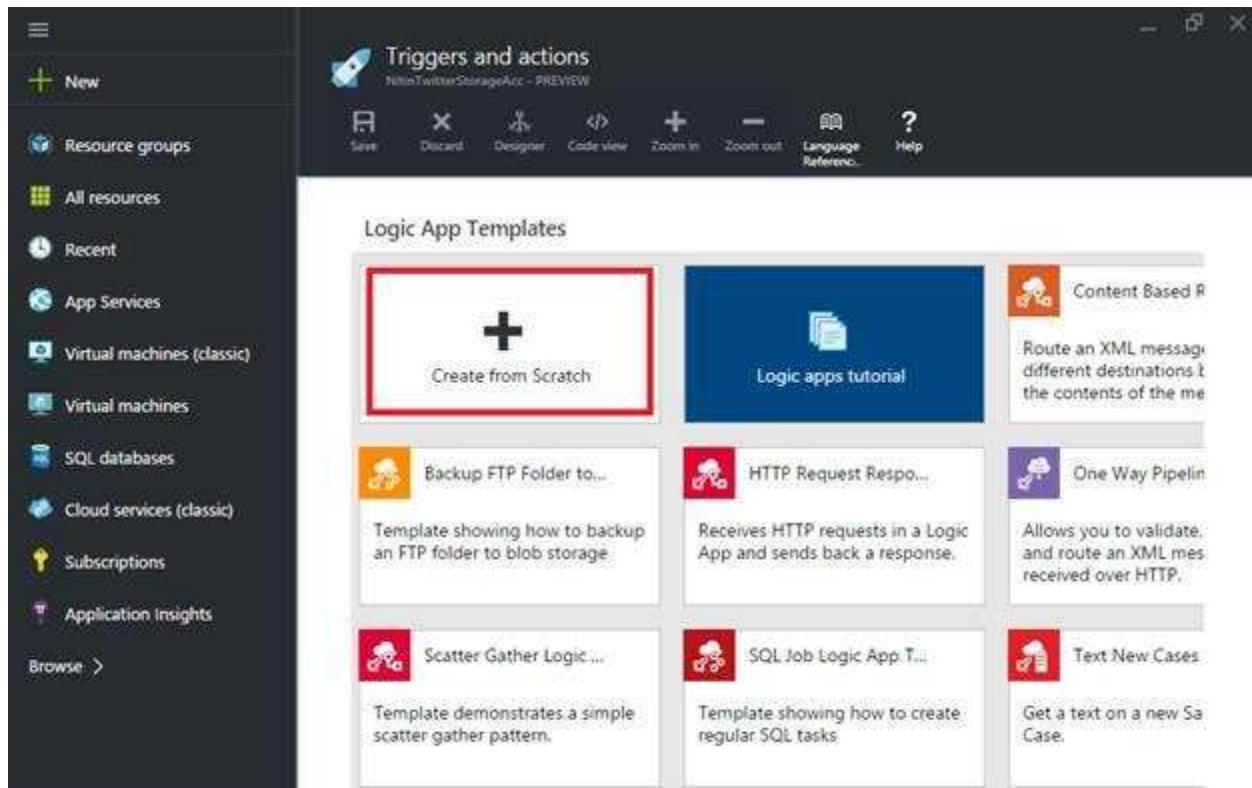


The screenshot shows the Microsoft Azure portal interface. The left sidebar lists various services: Resource groups, All resources, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Subscriptions, Application Insights, and a Browse link. The main content area displays the details for a Logic App named "NitinTwitterStorageAcc". The top navigation bar shows the URL <https://portal.azure.com/#resource/subscriptions/c06f5092-a810-4ca1-b2f1-6c/workflows>. The Logic App summary includes the following information:

Resource group	Endpoint
twitternitin	https://eastasia.logic.azure.com:443/subscr...
Status	Enabled
Location	East Asia
Subscription Name	Visual Studio Enterprise with MSDN
Subscription ID	c06f5092-a810-4ca1-b2f1-6612f6aae44c
Definition	0 triggers, 0 actions
Last run	None
Buns today	0 successful, 0 failed

Below the summary, there's a "Triggers and actions" section with a red box and a red arrow pointing to it, and a red oval with the text "click on it" overlaid. To the right of this section is a "Quick start" button with a cloud icon.

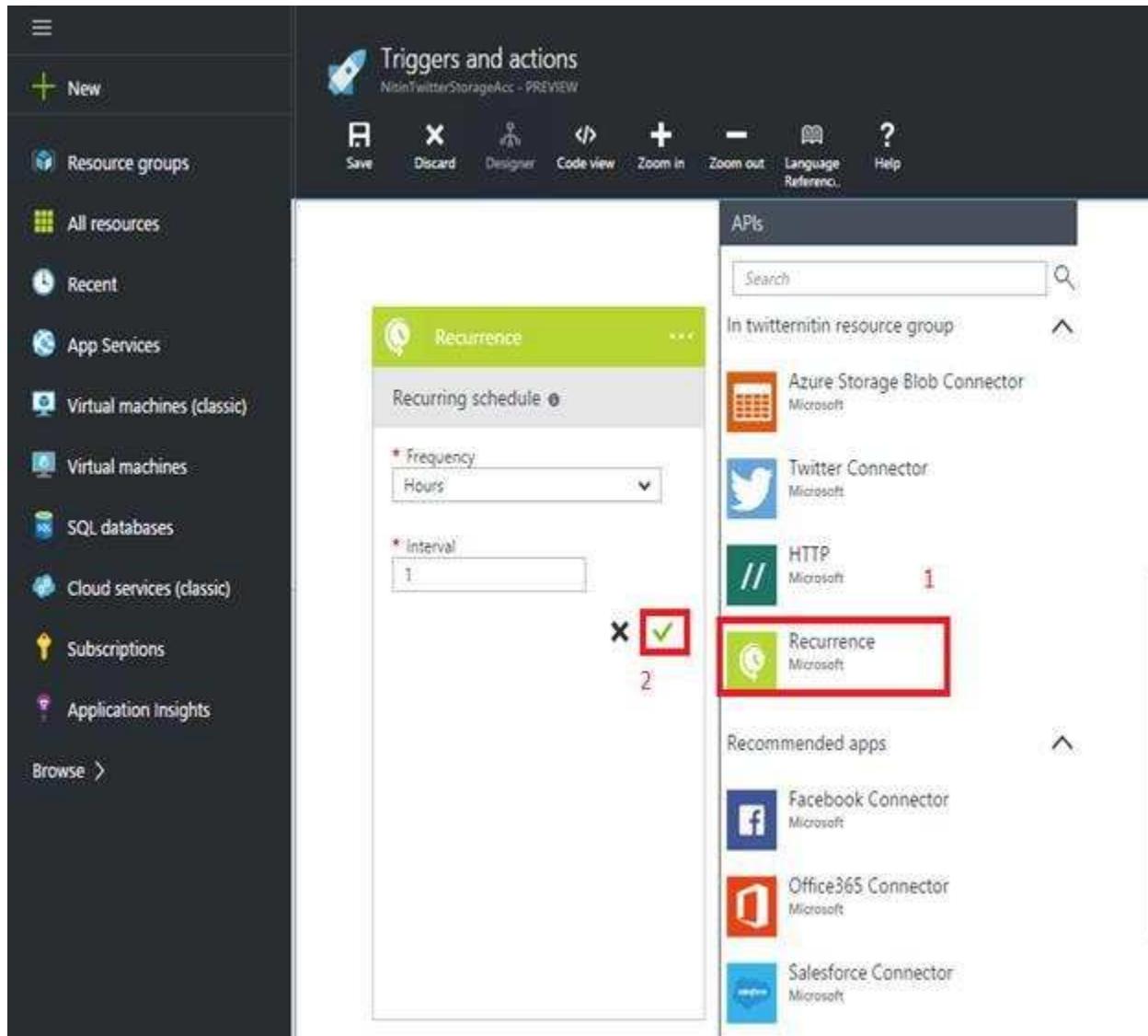
Select the “Create from Scratch” panel.



Now, you will see following figure. I need to create recurrence where I get the schedule of the recurrence.

Click on "**Recurrence**," right side of the window in resource group.

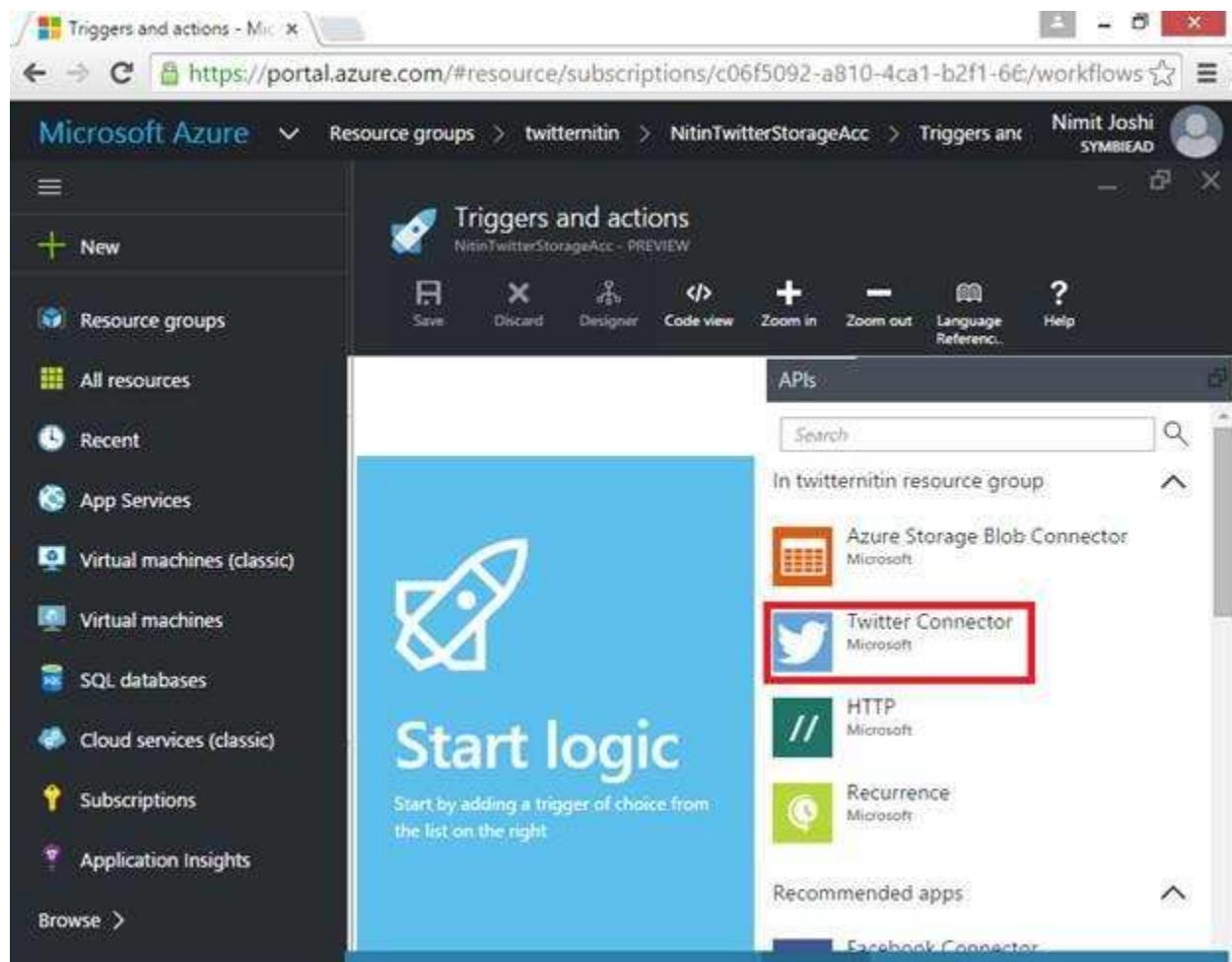
Choose the frequency and interval of the recurrence schedule, and click on the green check button.



The screenshot shows the Azure Functions 'Triggers and actions' blade for a function named 'NitinTwitterStorageAcc - PREVIEW'. On the left, there's a navigation bar with options like 'New', 'Resource groups', 'All resources', 'Recent', 'App Services', 'Virtual machines (classic)', 'Virtual machines', 'SQL databases', 'Cloud services (classic)', 'Subscriptions', and 'Application Insights'. The main area displays the 'Recurrence' trigger configuration. It includes fields for 'Frequency' (set to 'Hours') and 'Interval' (set to '1'). To the right, there's a sidebar titled 'APIs' with a search bar and a list of connectors. The 'Recurrence' connector is highlighted with a red box. Other connectors listed include 'Azure Storage Blob Connector', 'Twitter Connector', 'HTTP', and 'Facebook Connector', 'Office365 Connector', and 'Salesforce Connector' under 'Recommended apps'.

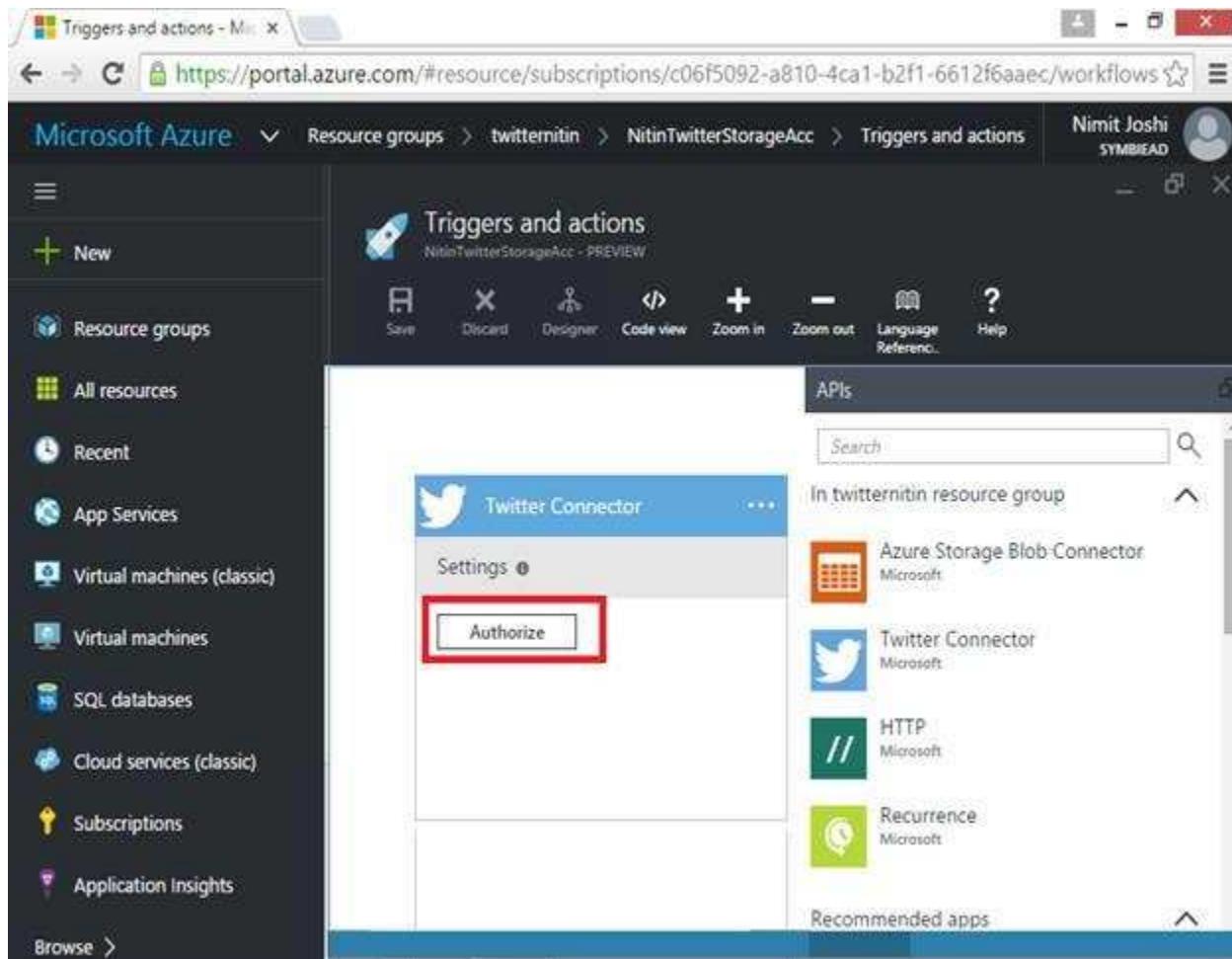
Now, we need to create twitter connector, where I fetch my twitter application.

So just select the "**Twitter Connector**."



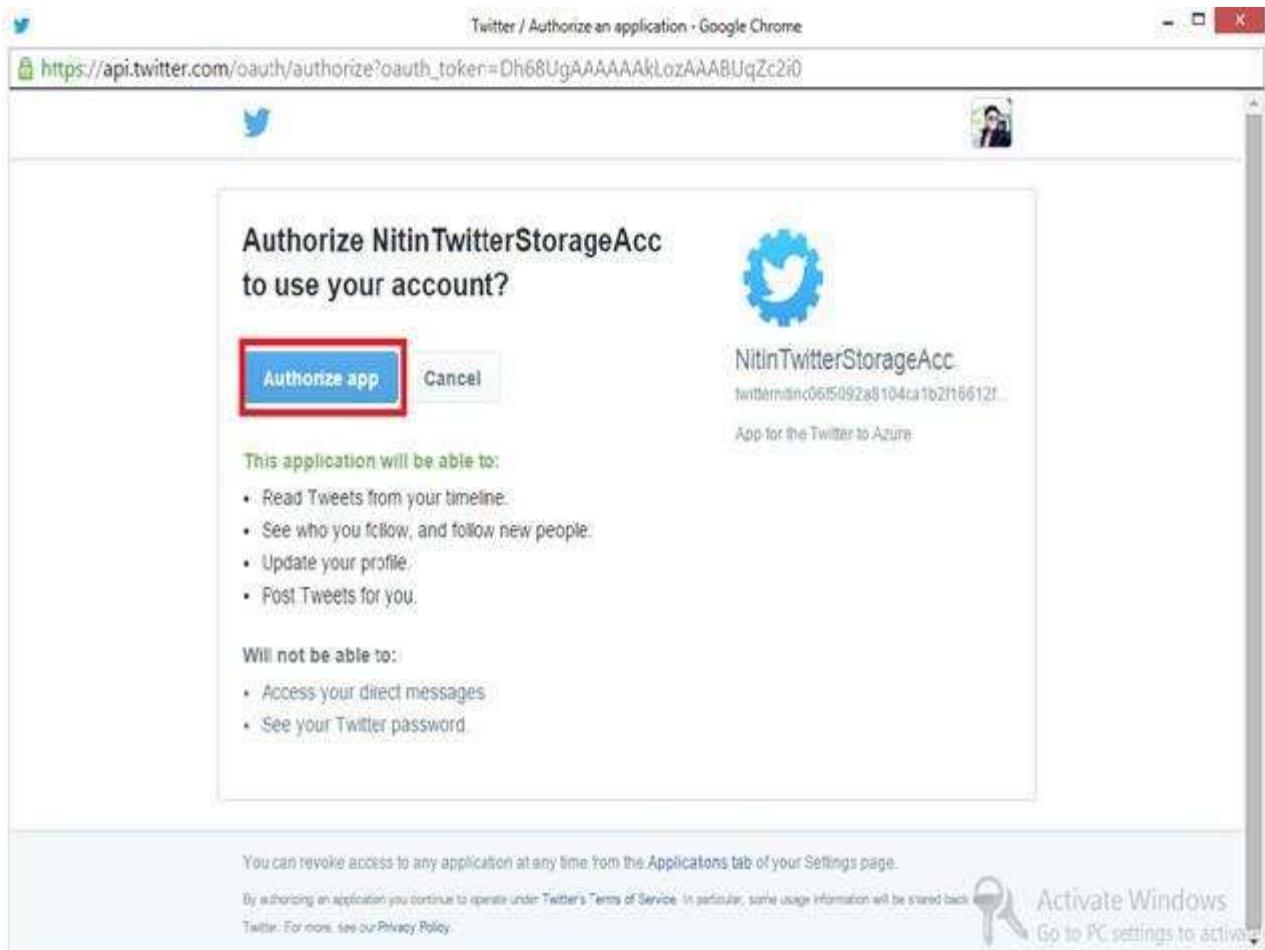
The screenshot shows the Microsoft Azure portal interface for managing triggers and actions. The left sidebar lists various service categories like Resource groups, All resources, and App Services. The main content area is titled "Triggers and actions" under "NitinTwitterStorageAcc - PREVIEW". It features a central "Start logic" button with a pencil icon and a placeholder text: "Start by adding a trigger of choice from the list on the right". To the right, there's a list of available connectors under the heading "APIs". The "Twitter Connector" (Microsoft) is highlighted with a red box. Other connectors listed include Azure Storage Blob Connector, HTTP, and Recurrence.

Click on “Authorize” button.



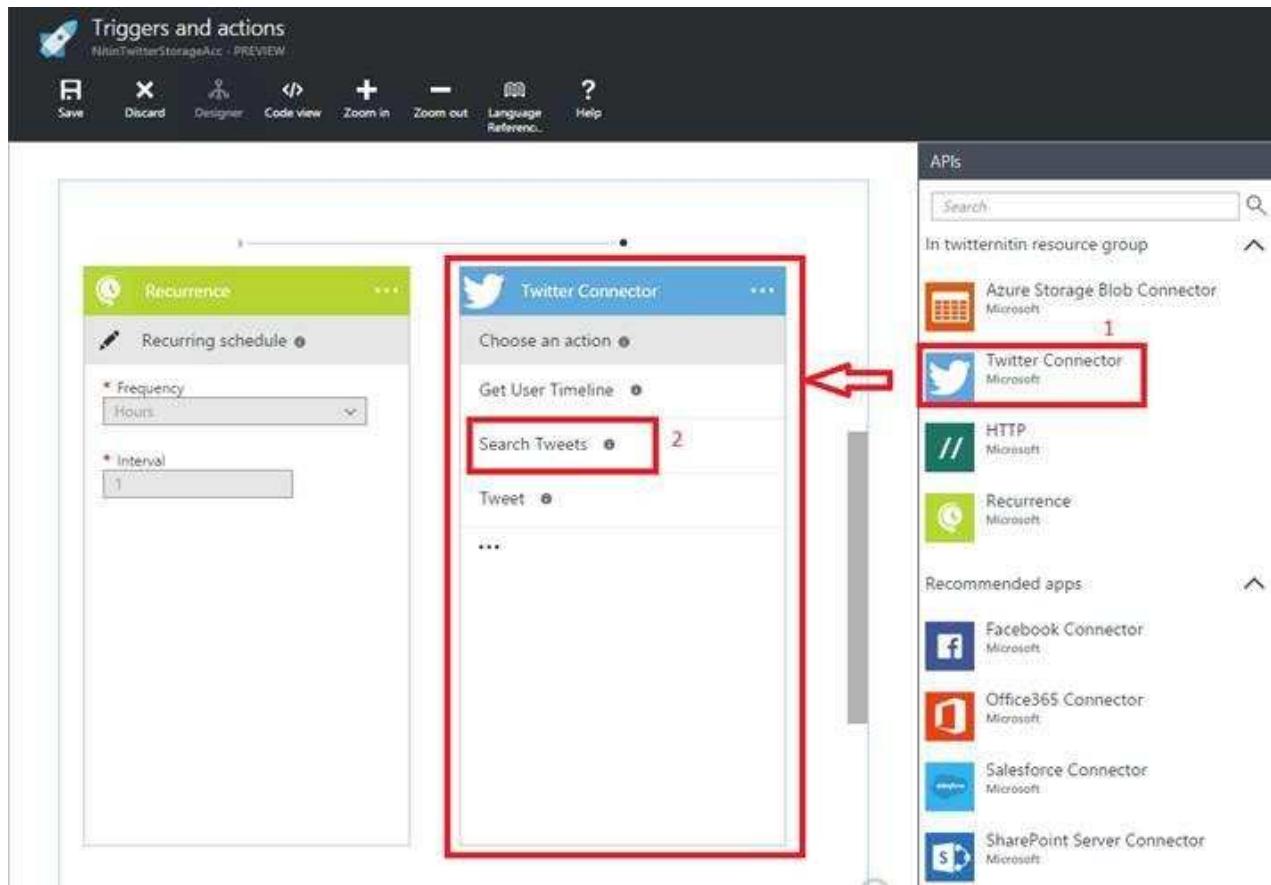
After clicking on the “Authorize” button, you will be redirected on twitter page for authorization.

Click on to “Authorize app.”



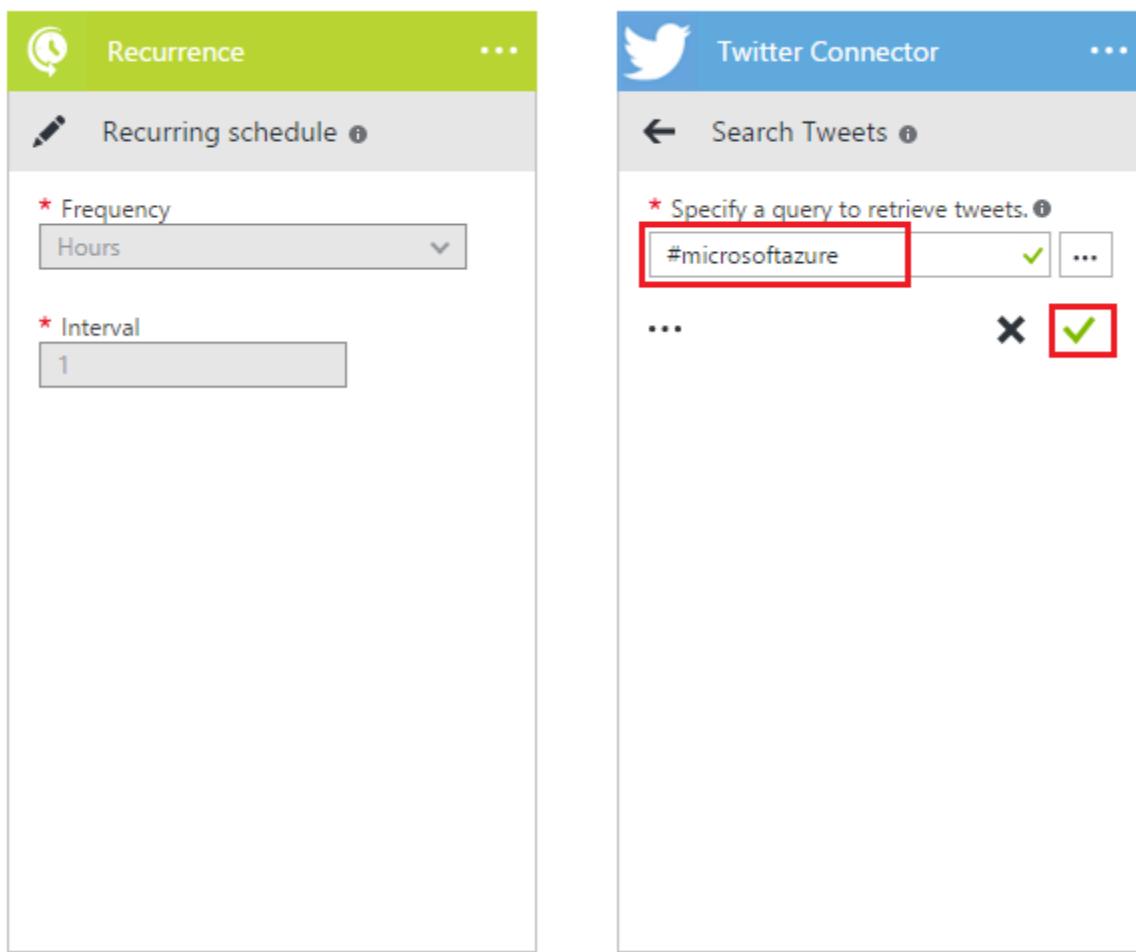
Now I'm successfully fetching with twitter.

Click on “**Search Tweets**” button, for the timeline of twitter account.



Give the query name to retrieve tweets from twitter. I have given the name “#microsoftAzure” to it.

Click on the green check.



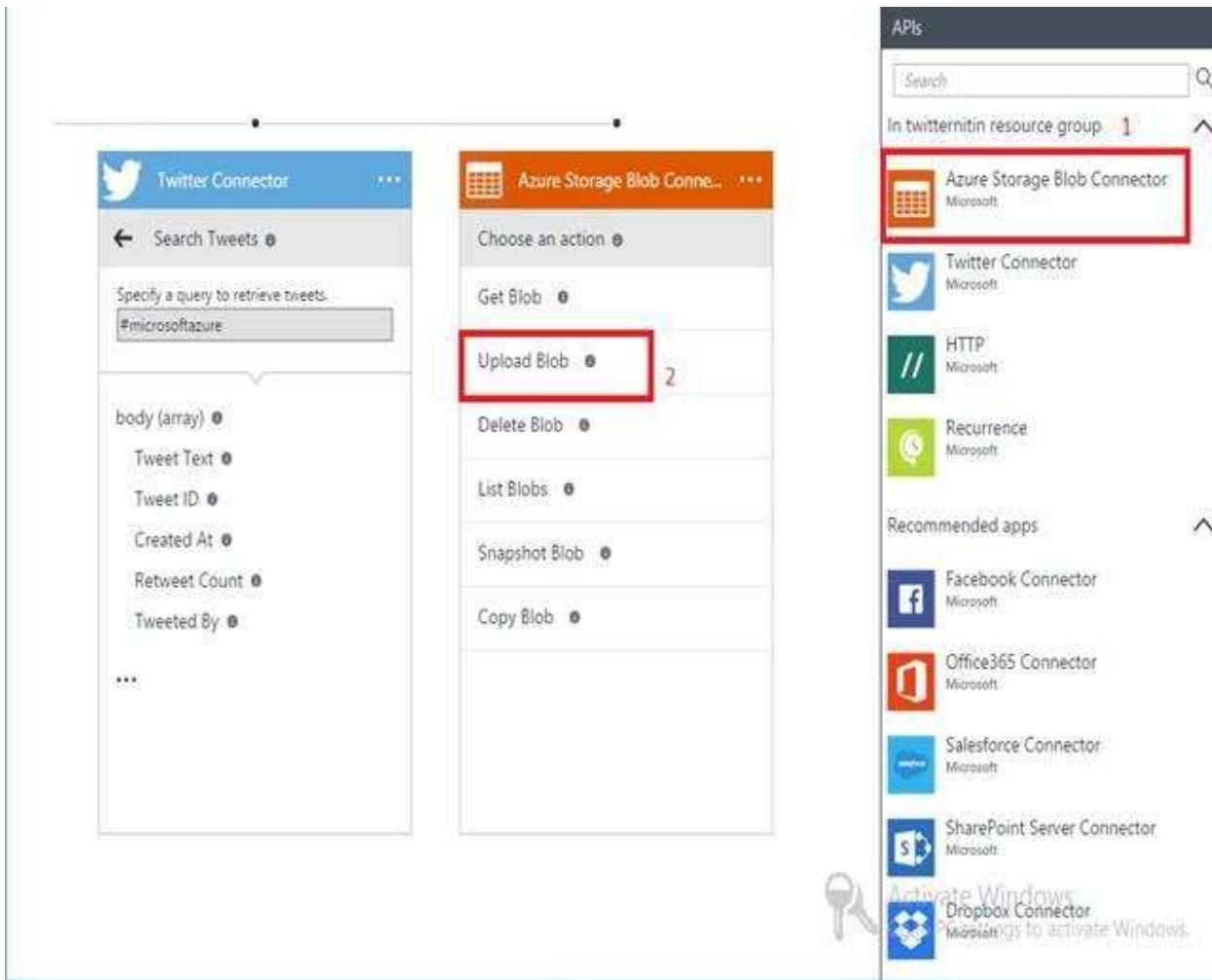
The screenshot shows the Microsoft Power Automate interface with two steps visible:

- Recurrence:** A step for setting a recurring schedule. It has a "Recurring schedule" section with "Frequency" set to "Hours" and "Interval" set to "1".
- Twitter Connector:** A step for searching tweets. It has a "Search Tweets" section with a query field containing "#microsoftazure". The query field is highlighted with a red box.

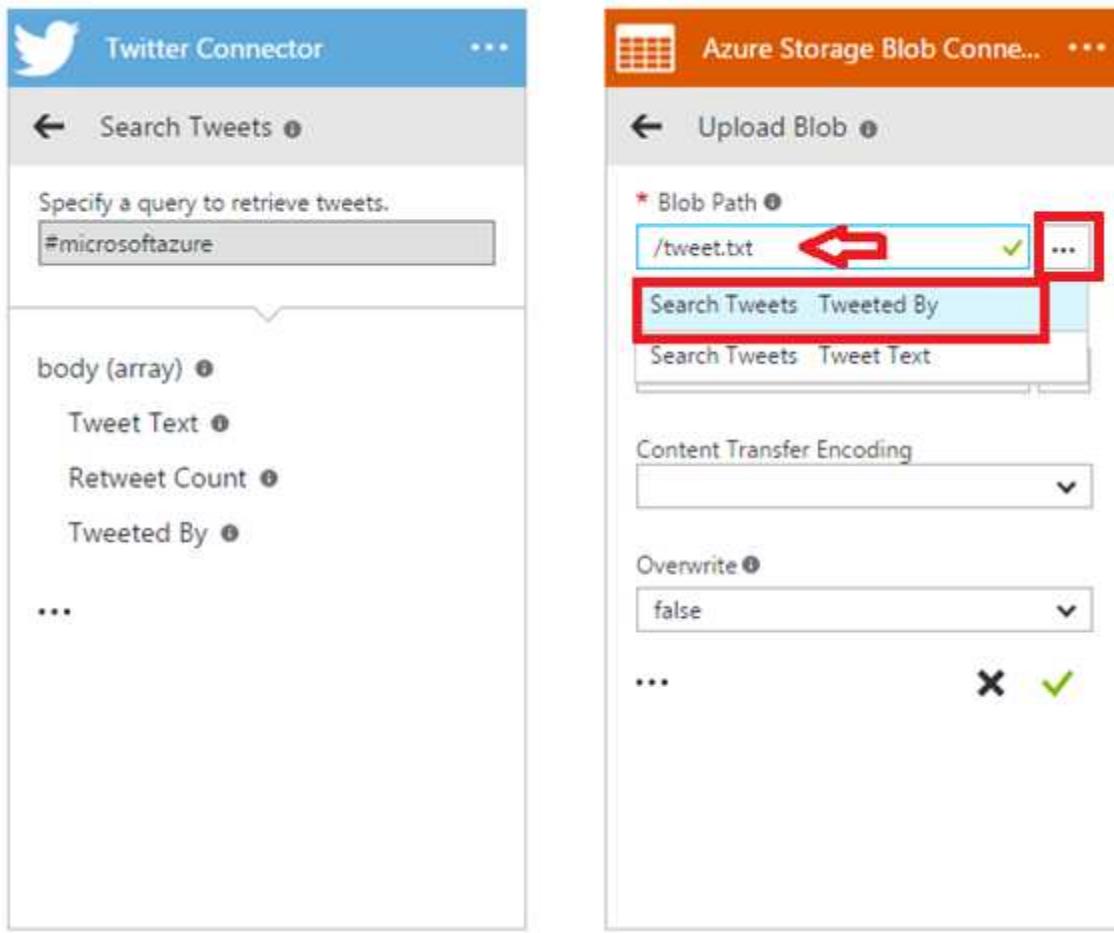
Now I need to fetch with “**Azure Storage Blob Connector**.” Select the “**Azure Storage Blob Connector**.”

After this, you will see the Azure connector window (Number 2).

Click on to “**Upload Blob**.”



In the “**Blob Path**” section, type the path “**/tweet.txt**” and just click on rectangle button, and select “**Tweeted By**” option from the dropdown.

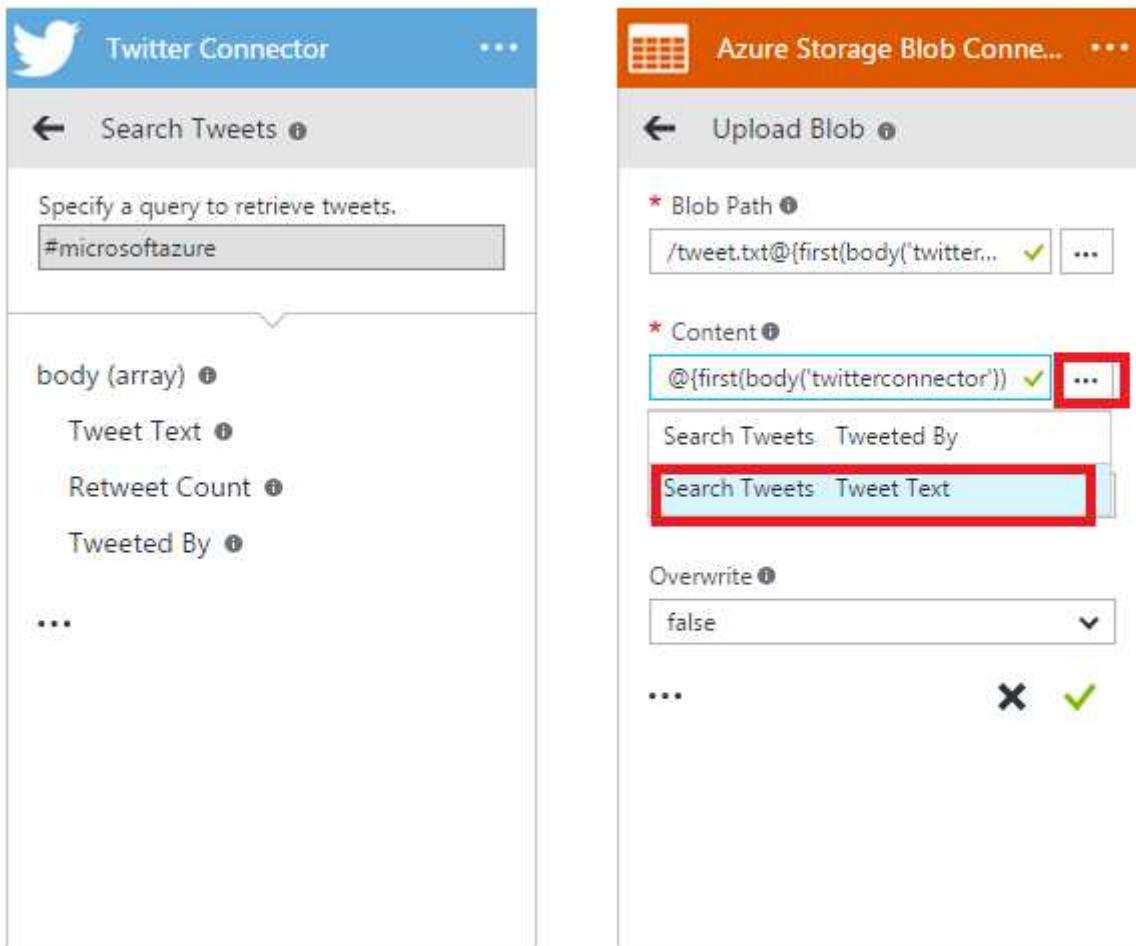


The image shows two side-by-side interface panels from the Microsoft Azure portal.

Twitter Connector Panel: This panel has a blue header with the Twitter logo and "Twitter Connector". Below it is a search bar labeled "Search Tweets" with the query "#microsoftazure". A list of properties is shown below the search bar: "body (array)", "Tweet Text", "Retweet Count", "Tweeted By", and three ellipsis (...).

Azure Storage Blob Connector Panel: This panel has an orange header with the Azure Storage logo and "Azure Storage Blob Conne...". Below it is a search bar labeled "Upload Blob" with the path "/tweet.txt". A dropdown menu is open, showing options: "Search Tweets", "Tweeted By" (which is highlighted with a red arrow and a green checkmark), and "Search Tweets", "Tweet Text". Other fields include "Content Transfer Encoding" and "Overwrite" set to "false". Ellipsis (...) buttons are at the bottom right.

Now click on “**Content**” option and click on rectangle button, and select “**Tweet Text**” option from the dropdown.

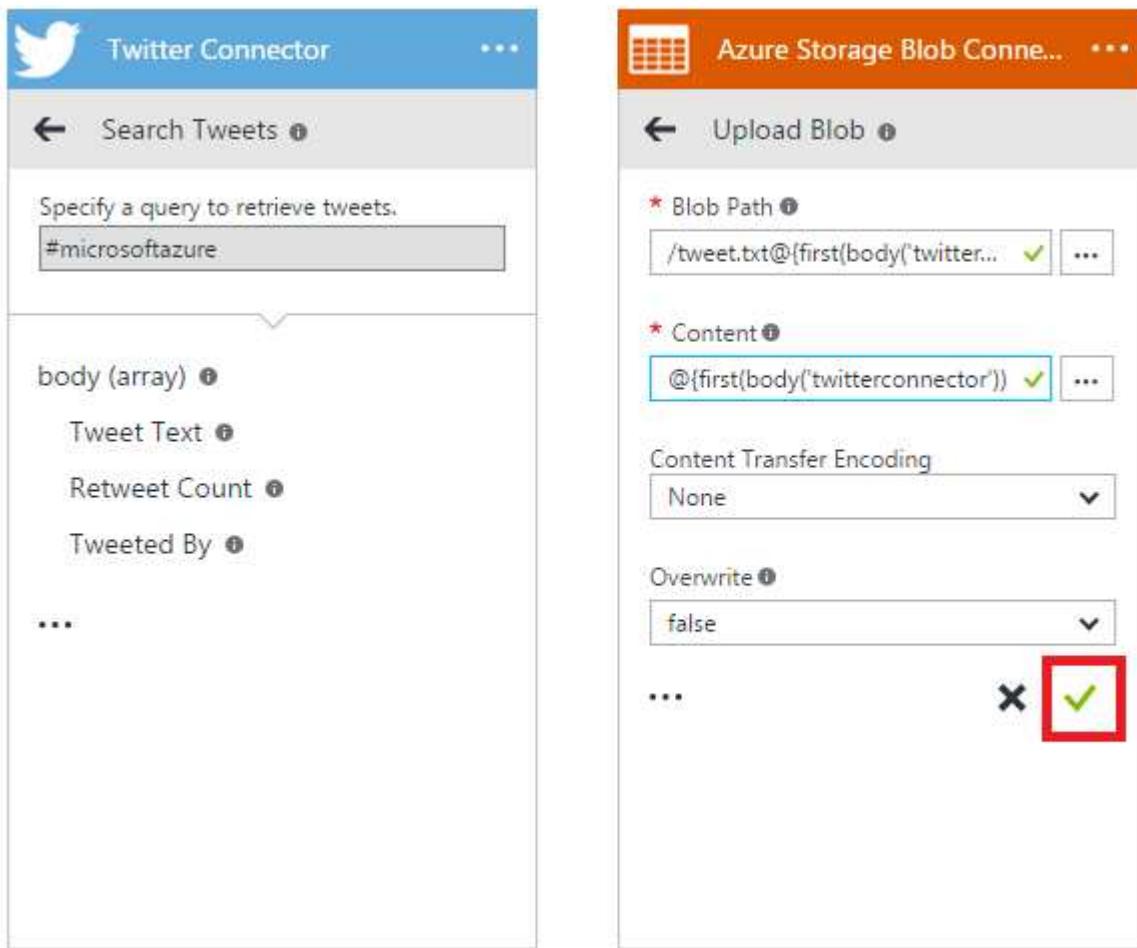


The image shows two side-by-side configuration screens from the Microsoft Azure portal.

Twitter Connector: This screen allows you to search tweets. A search query is entered: `#microsoftazure`. Below the search bar, the results are displayed as an array named `body`, which contains fields like `Tweet Text`, `Retweet Count`, and `Tweeted By`.

Azure Storage Blob Connector: This screen shows how to upload tweets to Azure Storage. The `Blob Path` is set to `/tweet.txt@{first(body('twitterconnector'))}`. The `Content` field is set to `@{first(body('twitterconnector'))}`. The `Search Tweets` field is highlighted with a red box and contains the value `Tweeted By`. The `Overwrite` dropdown is set to `false`.

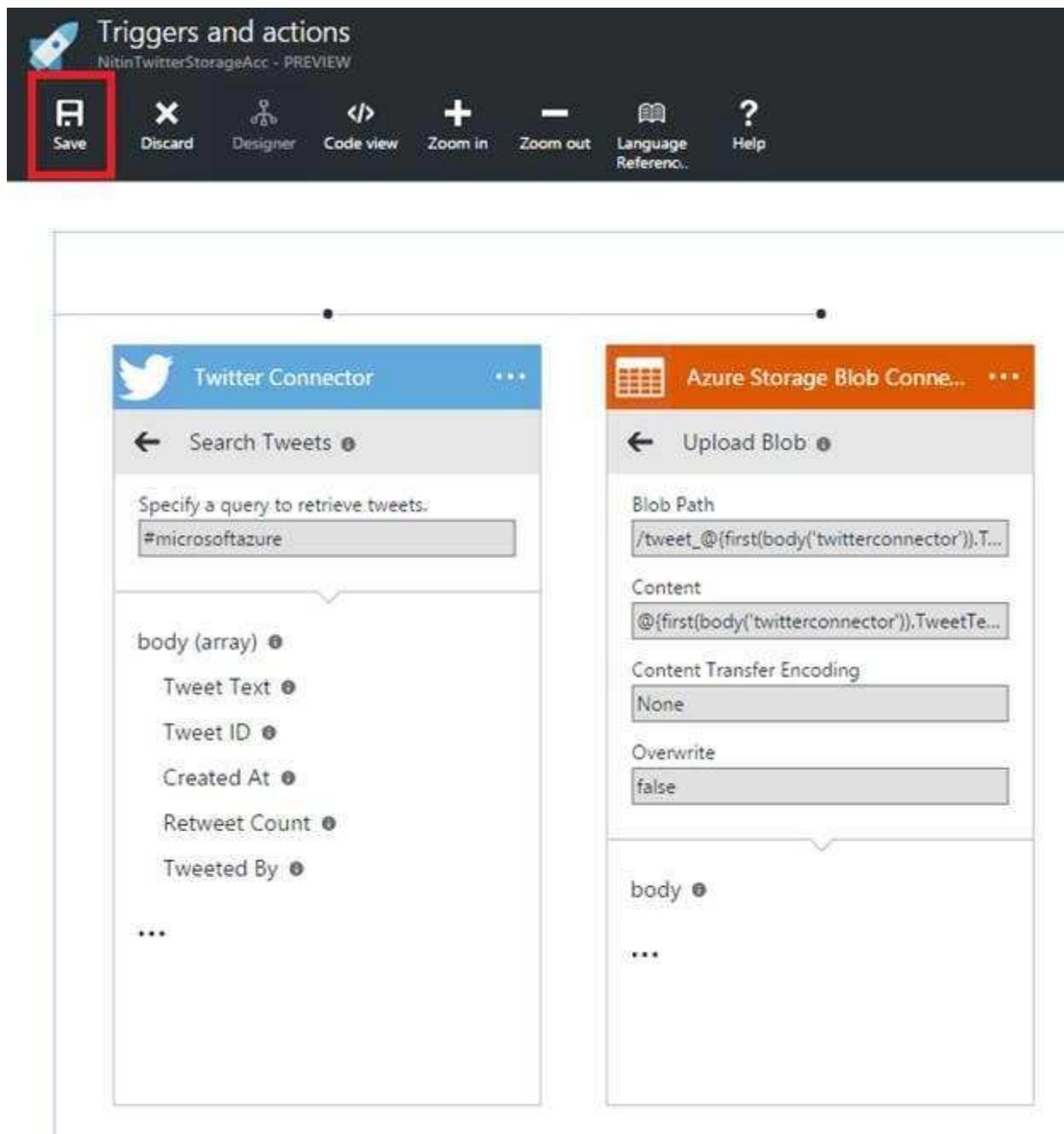
Click on the green check.



The screenshot displays two parallel workflows in the Microsoft Azure Logic Apps designer:

- Twitter Connector:** A blue header bar with the title "Twitter Connector". Below it, a search bar shows the query "#microsoftazure". A dropdown menu under "body (array)" lists three properties: "Tweet Text", "Retweet Count", and "Tweeted By".
- Azure Storage Blob Connector:** An orange header bar with the title "Azure Storage Blob Conne...". Below it, the "Upload Blob" step is configured with:
 - Blob Path:** /tweet.txt@{first(body('twitterconnector'))}
 - Content:** @{first(body('twitterconnector'))}
 - Content Transfer Encoding:** None
 - Overwrite:** false

Click on to “Save” button, to save the current actions.

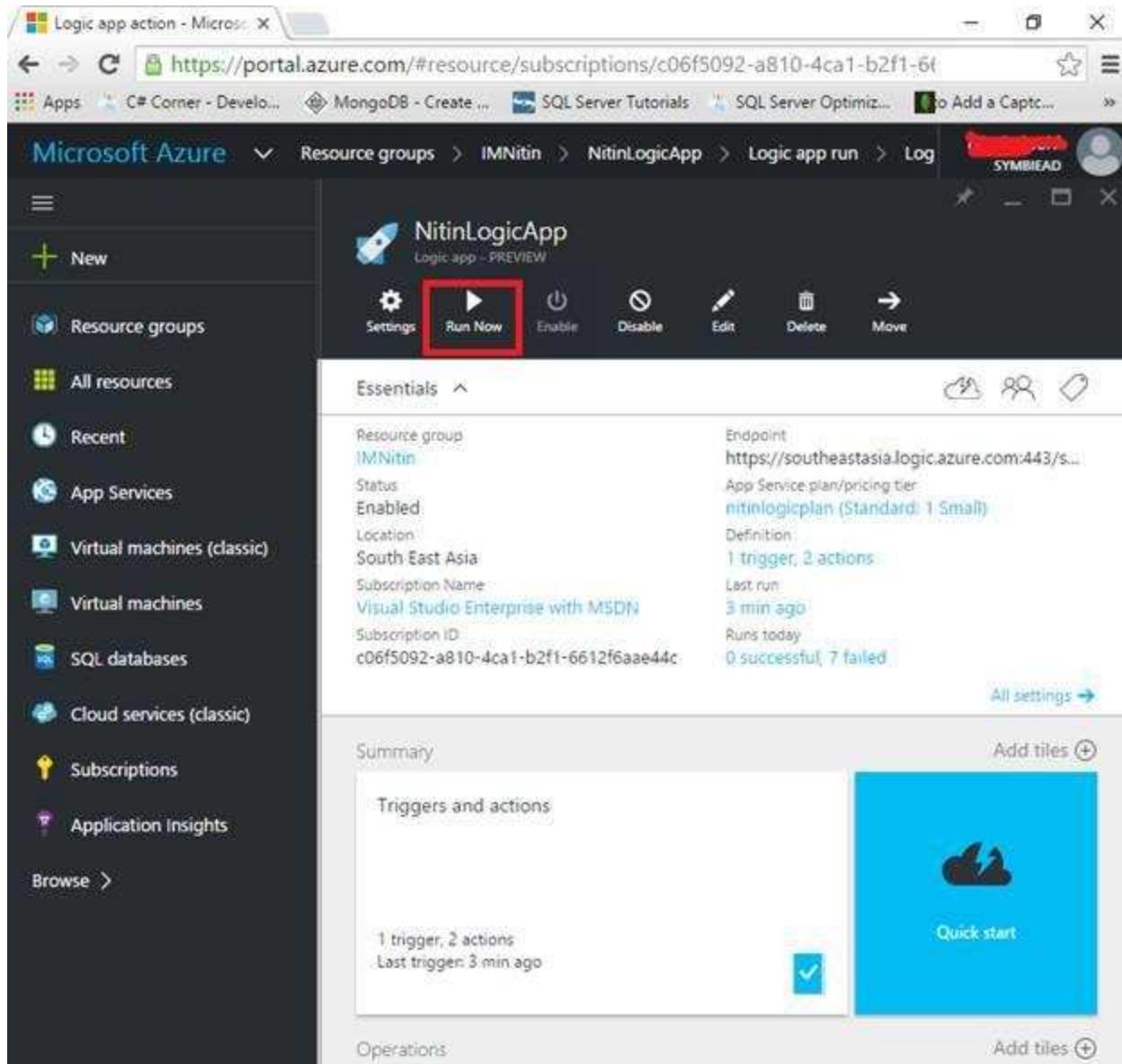


The screenshot shows the Azure Functions Triggers and actions interface. At the top, there's a toolbar with icons for Save (highlighted with a red box), Discard, Designer, Code view, Zoom in, Zoom out, Language Reference, and Help.

The main area displays two triggers/actions:

- Twitter Connector:** Triggered by "Search Tweets". The query specified is "#microsoftazure". The body (array) contains fields: Tweet Text, Tweet ID, Created At, Retweet Count, and Tweeted By.
- Azure Storage Blob Connector:** Triggered by "Upload Blob". The blob path is set to "/tweet_@(first(body('twitterconnector')).T...)" and the content is "@(first(body('twitterconnector')).TweetTe...)". The Content Transfer Encoding is set to "None" and Overwrite is set to "false".

Click on “Run Now” button to run the triggers and actions.



NitinLogicApp
Logic app - PREVIEW

Run Now

Essentials	Value
Resource group	IMNitin
Status	Enabled
Location	South East Asia
Subscription Name	Visual Studio Enterprise with MSDN
Subscription ID	c06f5092-a810-4ca1-b2f1-6612f6aae44c
Definition	1 trigger, 2 actions
Last run	3 min ago
Runs today	0 successful, 7 failed

Summary

Triggers and actions

1 trigger, 2 actions
Last trigger: 3 min ago

Quick start

We succeeded AND got my live twitter data in the Azure Blob.

Click on to “**twitterconnector**,” after clicking you will find the “Inputs Link” and “Outputs Link.”

Click on to “**INPUTS LINK**.”

The screenshot shows the Microsoft Azure portal interface for a Logic App run. On the left, the navigation bar includes 'Microsoft Azure', 'Resource groups', 'Logic app run' (selected), and 'Logic app action'. The main content area displays a 'Logic app run' card with a summary table and detailed logs.

Logic app run
05e7a6d75270704891 - PREVIEW

Action	Status	Duration
twitterconnector	Succeeded	1.61 Seconds
	FAILED	0:00:00.0000000
	RUNNING	-
	SKIPPED	-

DETAILS

START DATE	Thursday, February 11, 2016, 4:44:18 PM
END DATE	Thursday, February 11, 2016, 4:44:20 PM
CORRELATION ID	ffa59759-018d-4b85-9cc4-cc4a73be92a3
OUTPUTS	No outputs

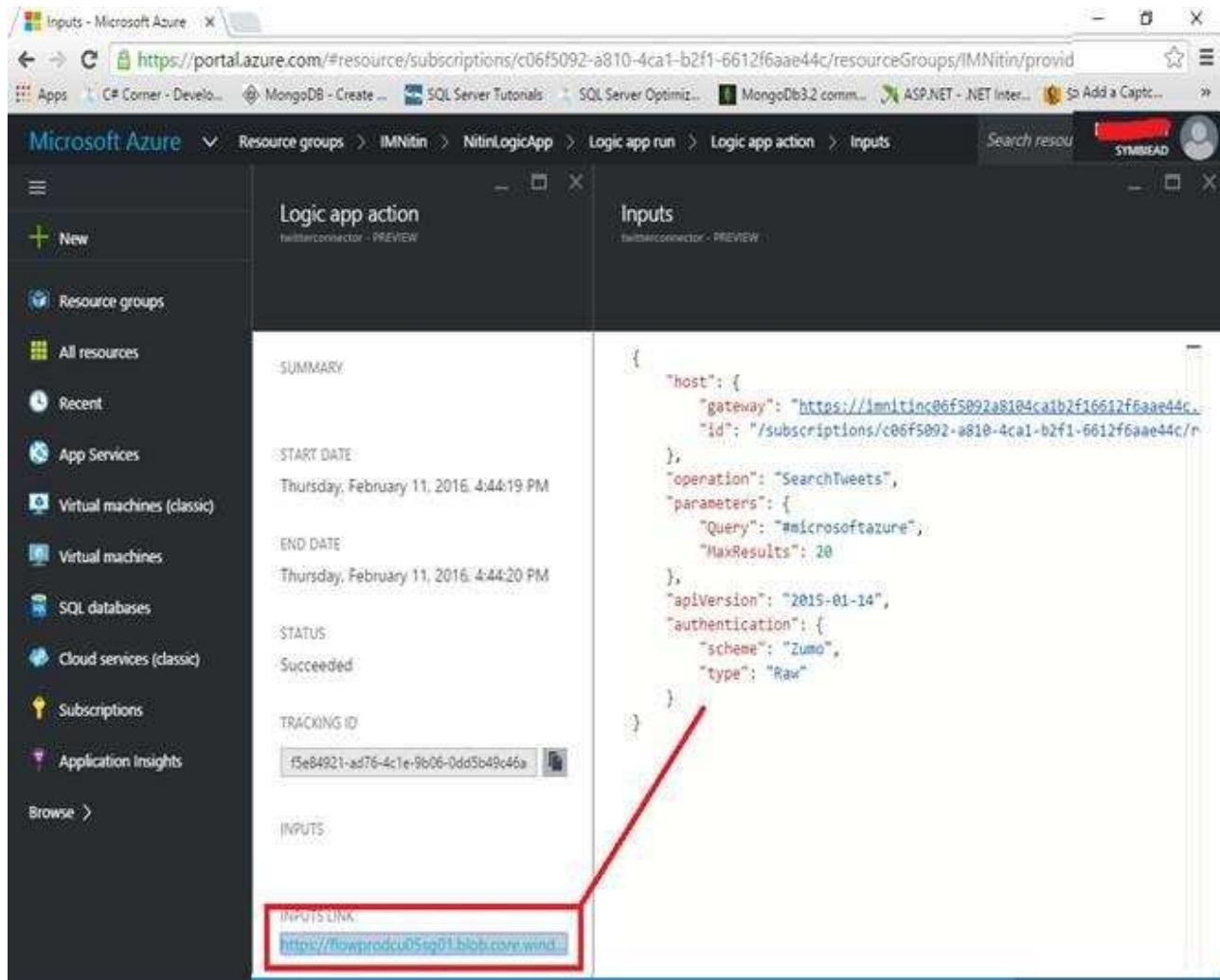
INPUTS

INPUTS LINK
<https://flowprodcu05sg01.blob.core.windows.net/>

OUTPUTS

OUTPUTS LINK
<https://flowprodcu05sg01.blob.core.windows.net/>

So, we can see the query, operations, authentication, and result.



The screenshot shows the Microsoft Azure Logic App interface. On the left, the navigation pane lists various resources like Resource groups, All resources, Recent, App Services, etc. The main area displays a "Logic app action" step with the "twittterconnector - PREVIEW" provider. The "Inputs" tab is selected, showing a JSON payload for a Twitter search:

```

{
  "host": {
    "gateway": "https://imnitinc06f5092-a810-4ca1-b2f1-6612f6aae44c",
    "id": "/subscriptions/c06f5092-a810-4ca1-b2f1-6612f6aae44c/resourceGroups/IMNitin/providers/Microsoft.Web/sites/imnitinc06f5092-a810-4ca1-b2f1-6612f6aae44c"
  },
  "operation": "SearchTweets",
  "parameters": {
    "Query": "#microsoftazure",
    "MaxResults": 20
  }
}
  
```

Below the JSON, there's a "TRACKING ID" field containing "f5e84921-ad76-4c1e-9b06-0dd5b49c46a". At the bottom, there's a link labeled "INPUTS LINK" with the URL "https://bowprodcl05sg01.blob.core.windows.net/". A red box highlights this link.

Now click on “Outputs Link.”

I have successfully fetched with Twitter data.

SUMMARY	
START DATE	Thursday, February 11, 2016, 4:44:19 PM
END DATE	Thursday, February 11, 2016, 4:44:20 PM
STATUS	Succeeded
TRACKING ID	f5e84921-ad76-4c1e-9b06-0dd5b49c46a
INPUTS	
INPUTS LINK	https://flowprodceu05sg01.blob.core.windows.net/
SIZE	385 bytes
OUTPUTS	
OUTPUTS LINK	https://flowprodceu05sg01.blob.core.windows.net/
SIZE	6441 bytes

```

    "x-access-level": "read-write",
    "x-connection-hash": "f15230fi3e7fc10552ib043fd7e31b3b",
    "x-content-type-options": "nosniff,nosniff",
    "x-frame-options": "SAMEORIGIN",
    "x-rate-limit-limit": "188",
    "x-rate-limit-remaining": "178",
    "x-rate-limit-reset": "1455189579",
    "x-response-time": "124",
    "x-transaction": "96f9802b4a6d6b09",
    "x-twitter-response-tags": "BouncerCompliant",
    "x-xss-protection": "1; mode=block",
    "x-AspNet-Version": "4.0.30319",
    "x-Powered-By": "ASP.NET",
    "x-ms-proxy-outgoing-newuri": "https://twitterconnectordfd19a",
    "date": "Thu, 11 Feb 2016 11:14:19 GMT"
},
"body": [
{
    "TweetText": "Can cloud-based #ALIP deployed on #Microso
    "TweetID": "697738369242112000",
    "Created_at": "Thu Feb 11 11:06:05 +0000 2016",
    "Retweet_Count": 0,
    "Tweeted_By": "AccentureIns"
},
{
    "TweetText": "RT ModernBizIN: Here's how you store &
    "TweetID": "697728266329792512",
    "Created_at": "Thu Feb 11 10:25:56 +0000 2016",
    "Retweet_Count": 0,
    "Tweeted_By": "Ilyas_tweets"
},
{
    "TweetText": "RT @ModernBizIN: Here's how you store &
    "TweetID": "697725363980472321",
    "Created_at": "Thu Feb 11 10:14:24 +0000 2016",
    "Retweet_Count": 4,
    "Tweeted_By": "Deananth"
},
{
    "TweetText": "Join in our 12th anniversary by taking part
    "TweetID": "697708042423554048",
    "Created_at": "Thu Feb 11 09:05:34 +0000 2016",
    "Retweet_Count": 0,
    "Tweeted_By": "think_S3"
}
]

```

Cloud Services Scheduler in Microsoft Azure

Cloud Services Scheduler is a cloud based task scheduling service provided by Microsoft Azure. You can perform many actions regarding your job details in the scheduler. We can also check Scheduler's monitoring and management capabilities. Scheduler can maintain multiple job collections.



To create a job, sign into Azure Portal.

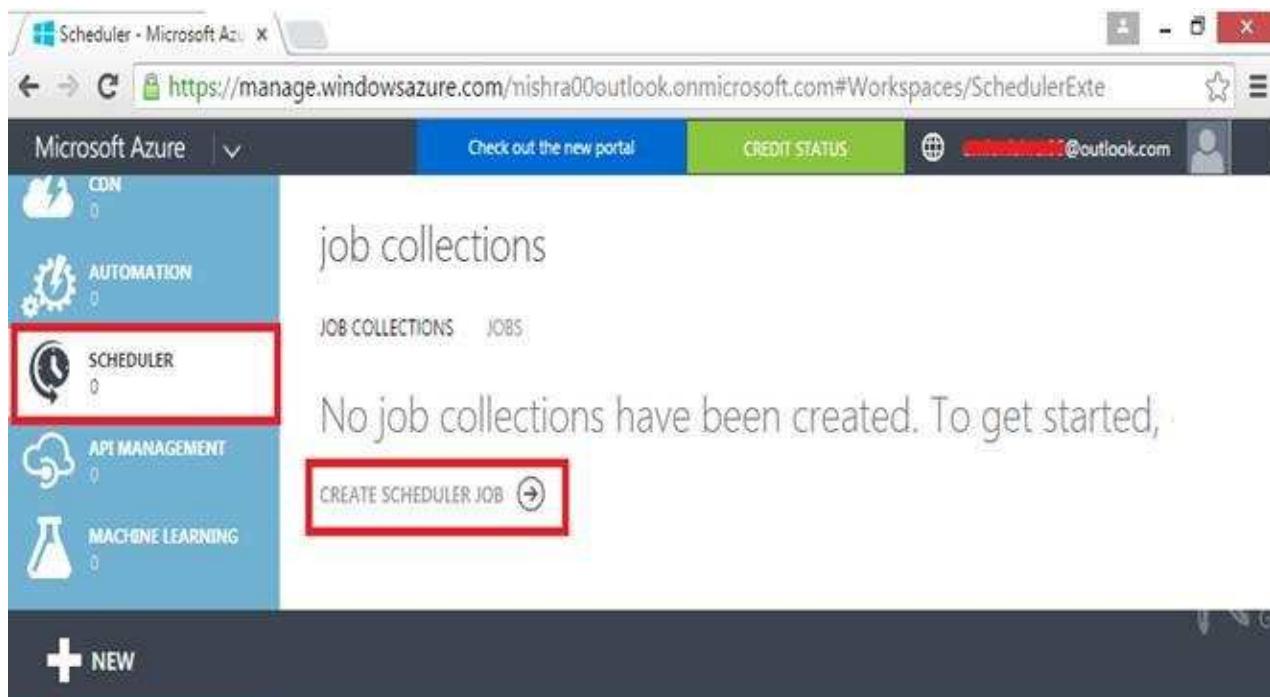
In the All Items panel, scroll down and click on to “Scheduler” to create a scheduler.

Click on to “Create Scheduler Job” or “New” button in the bottom of the window.

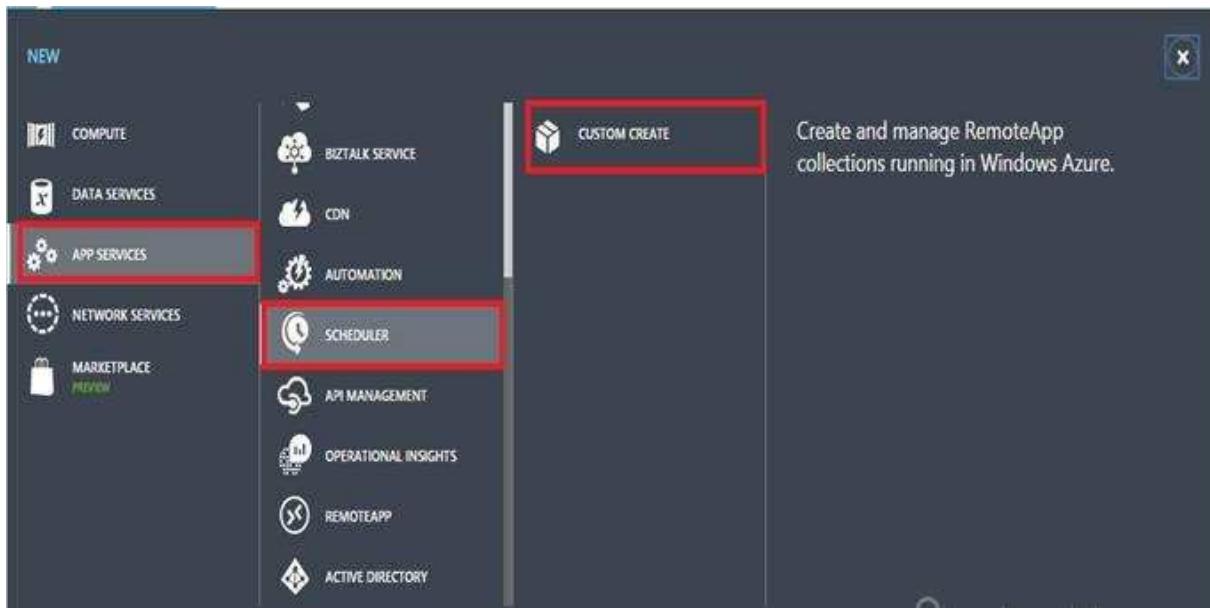


We Use Scheduler to:

- Invoke a web service over HTTP/s.
- Post a message to an Azure Storage Queue.
- Post a message to an Azure service bus queue.
- Post a message to an Azure Service Bus Topic.

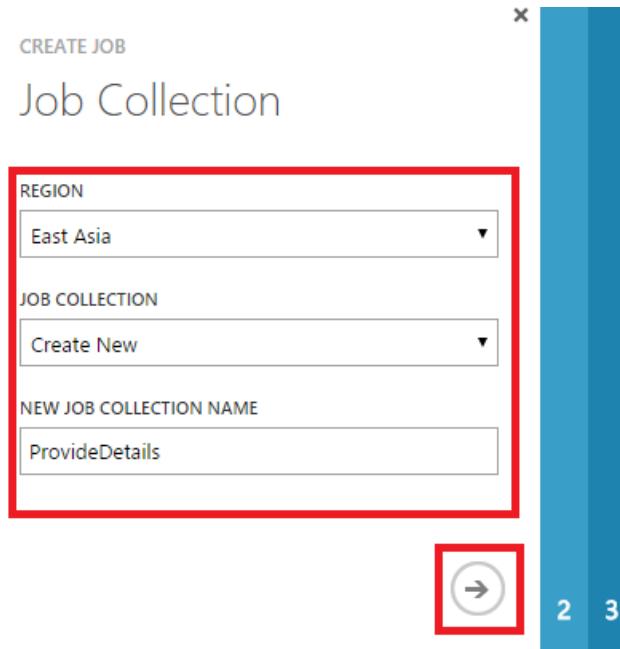


In the *APP SERVICES* select *SCHEDULER*, and then click *CUSTOM CREATE*.



The following is the job collection screenshot, to create a job collection, select your region and job collection name; I gave it “*ProvideDetails*”.

Click on to arrow for further information.



CREATE JOB

Job Collection

REGION
East Asia

JOB COLLECTION
Create New

NEW JOB COLLECTION NAME
ProvideDetails

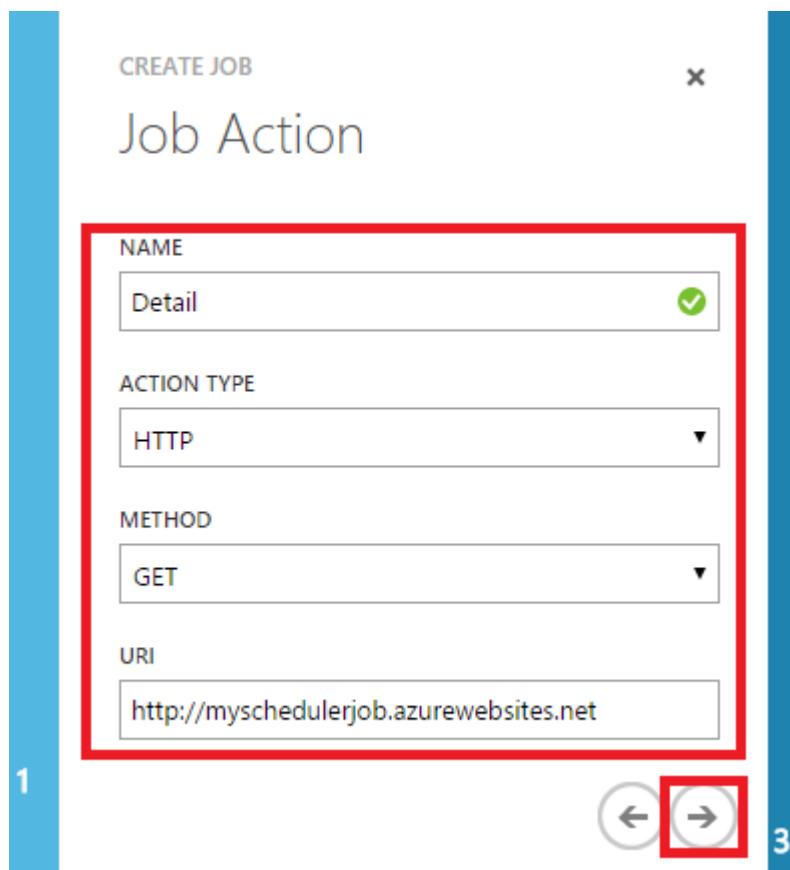
2 3

Now, here I need to create a job, give a name to your job, whatever you want to give.

In the Action Type, you will see three actions: HTTP, Storage Queue and HTTPS. If you are going to select HTTP then method name will be GET.

In the URI tab, you must have to enter your own URI; here, I've created my own URI.

Click on next,



Now define the scheduled details to your job.

Click on next button.

CREATE JOB

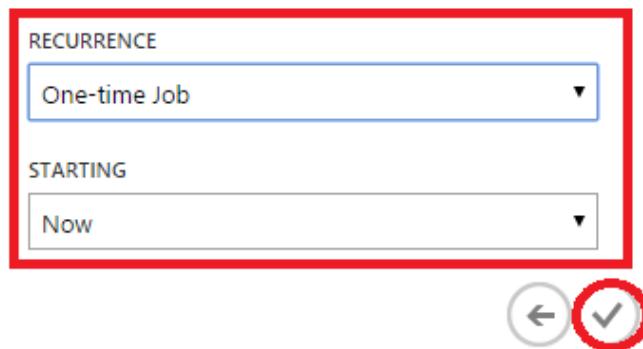
Define a Schedule

RECURRANCE

One-time Job

STARTING

Now



My job scheduler is successfully created, click on to dashboard to check your details of job, how many jobs you have completed, and how many jobs enabled, faulted and more information.



DASHBOARD SCALE JOBS HISTORY

job usage overview

1 JOBS

JOBS 2% of 50 JOBS STATUS Started REGION East Asia

0 JOBS

COMPLETED 0% of 1 COMPLETED HAS ERRORS 1

0 JOBS

FAULTED 0% of 1 FAULTED OCCURRENCE FAULTS 0

1 JOBS

ENABLED 100% of 1 ENABLED URL https://management.core.windows.net/cf741fae-b044-4acc-9f36-e8a6374d29fc/cloudservices/C5-EastAsia-scheduler/resources/scheduler/-/JobCollections/ProvideDetails

0 JOBS

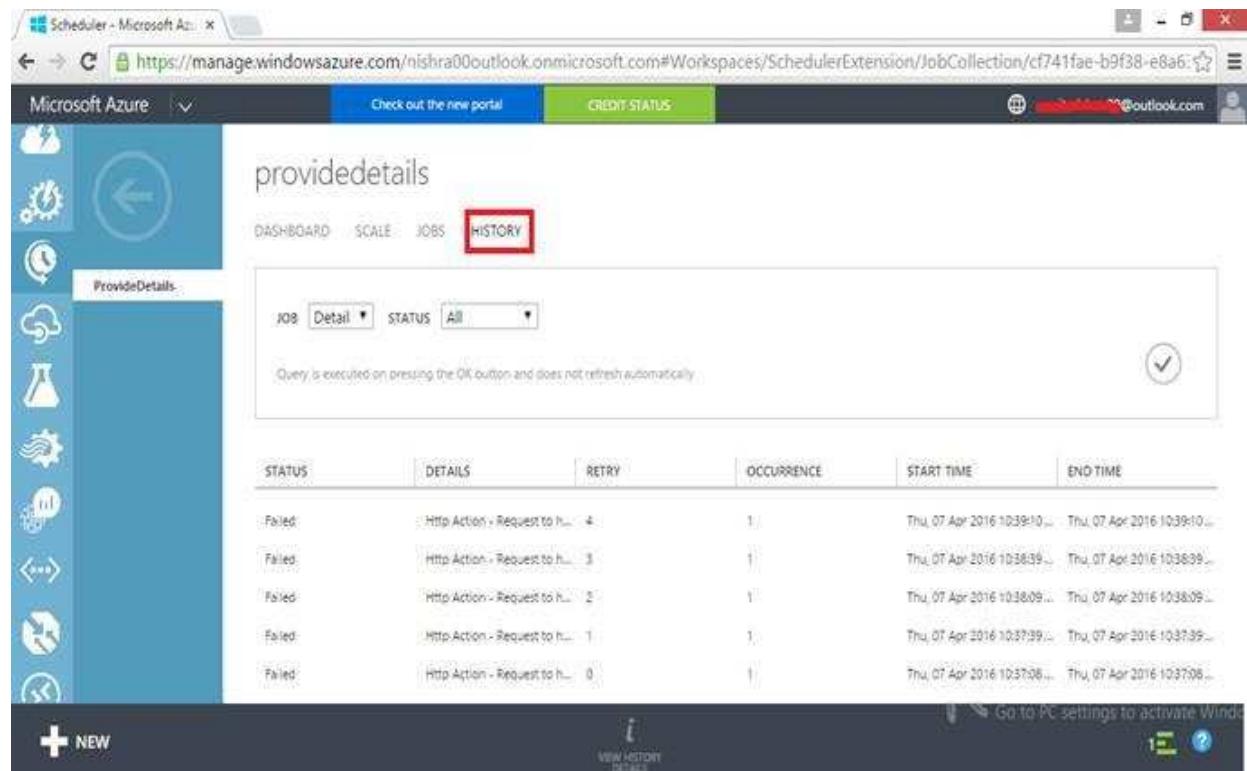
DISABLED 0% of 1 DISABLED

execution usage overview

3 ERRORS

EXECUTIONS 100% of 3 EXECUTIONS

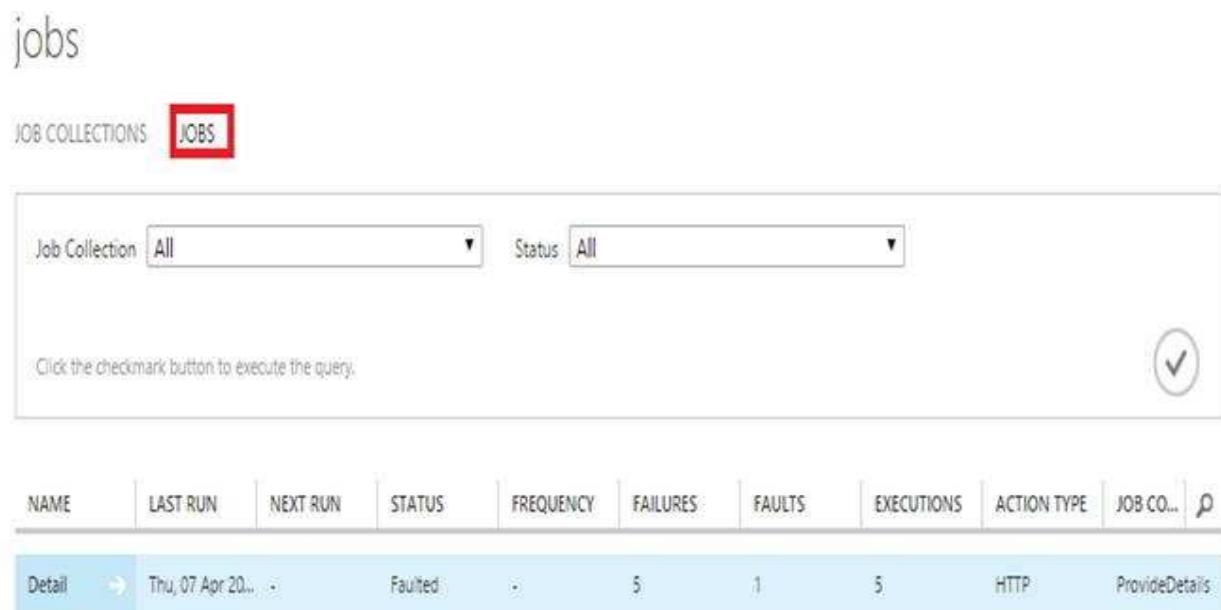
Click on “*HISTORY*” to check all the history of your jobs which are faulted or succeeded.



The screenshot shows the Microsoft Azure Scheduler interface. The top navigation bar includes links for "Check out the new portal" and "CREDIT STATUS". The left sidebar has various icons for different services, with "ProvideDetails" highlighted. The main content area is titled "providedetails". The "JOBS" tab is selected, and the "HISTORY" tab is highlighted with a red box. Below this, there are dropdown menus for "Detail" and "STATUS" set to "All", and a note stating "Query is executed on pressing the OK button and does not refresh automatically." A large table lists job execution details:

STATUS	DETAILS	RETRY	OCCURRENCE	START TIME	END TIME
Failed	Http Action - Request to h...	4	1	Thu, 07 Apr 2016 10:39:10...	Thu, 07 Apr 2016 10:39:10...
Failed	Http Action - Request to h...	3	1	Thu, 07 Apr 2016 10:38:39...	Thu, 07 Apr 2016 10:38:39...
Failed	Http Action - Request to h...	2	1	Thu, 07 Apr 2016 10:38:09...	Thu, 07 Apr 2016 10:38:09...
Failed	Http Action - Request to h...	1	1	Thu, 07 Apr 2016 10:37:39...	Thu, 07 Apr 2016 10:37:39...
Failed	Http Action - Request to h...	0	1	Thu, 07 Apr 2016 10:37:08...	Thu, 07 Apr 2016 10:37:08...

In the jobs panel, there will be all the details of your job collections as well as status of your job.



The screenshot shows the Microsoft Azure portal with the "JOBS" tab selected in the "JOB COLLECTIONS" section. The interface includes dropdown menus for "Job Collection" (set to "All") and "Status" (set to "All"), and a note "Click the checkmark button to execute the query." Below this is a table with columns: NAME, LAST RUN, NEXT RUN, STATUS, FREQUENCY, FAILURES, FAULTS, EXECUTIONS, ACTION TYPE, and JOB CO... (with a search icon). One row is visible, showing details for a job named "Detail" which last ran on "Thu, 07 Apr 2016" at "Faulted" status, with 5 failures, 1 fault, 5 executions, and an "HTTP" action type. The "ProvideDetails" link is also visible.

Azure Media Services Overview and Playing Media

Azure media services help you to build custom media workflows in the cloud. In Azure Media Services it is an end-to-end solution that lets you deliver audio and video virtually in any device anywhere. The Service offers end-to-end workflow which is very highly scalable in coding, packaging, storing and distribution of media content to a wide range of devices including TV, PC, and mobile devices.

Now let's start with Azure Media Services:

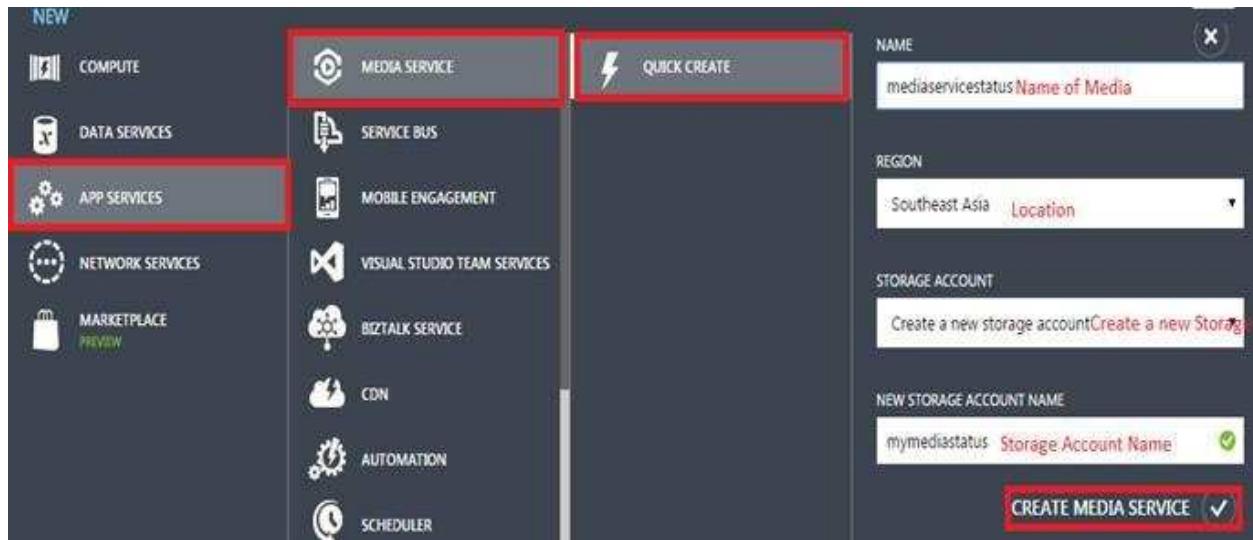
Sign in to Azure.



Click on AppServices, Media Service, and then Quick Create.

Give the name to your Azure Media Service, select the region/location and we need a new storage account where our media will be saved.

Give Storage Account Name, and click on to *CREATE MEDIA SERVICE*.



A new Azure Media Service will be created. Click on to it.



NAME	STATUS	SUBSCRIPTION	LOCATION
mediaservicestatus	Active	Azure Pass	Southeast Asia

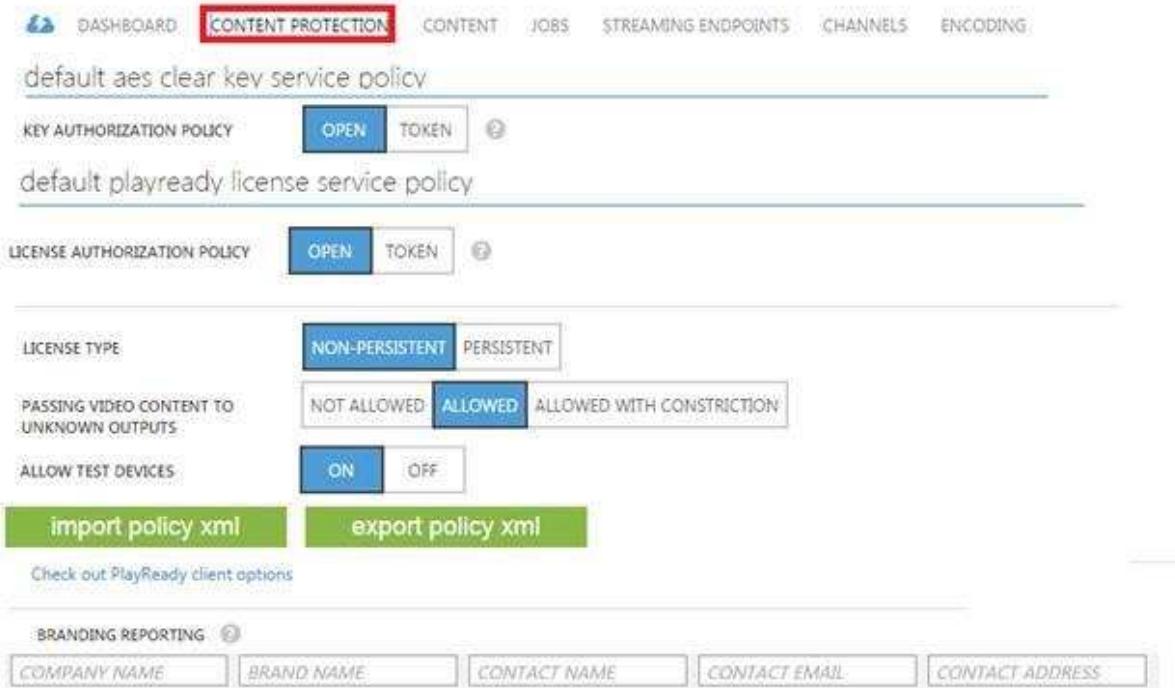
In this interface as you can see there are number of tabs.



Let's start with content protection tab, in content protection you can:

- To protect your premium video content, Azure media is introducing a content protection offering that features both static and dynamic encryption for a range of multi-DRM (PlayReady and Widevine).
- Secure your content from cloud upload to playback.
- You can secure playback with client SDKs for all your devices.
- Simplifies key Management and helps to maintain future proof content with dynamic Encryption.

mediaservicestatus



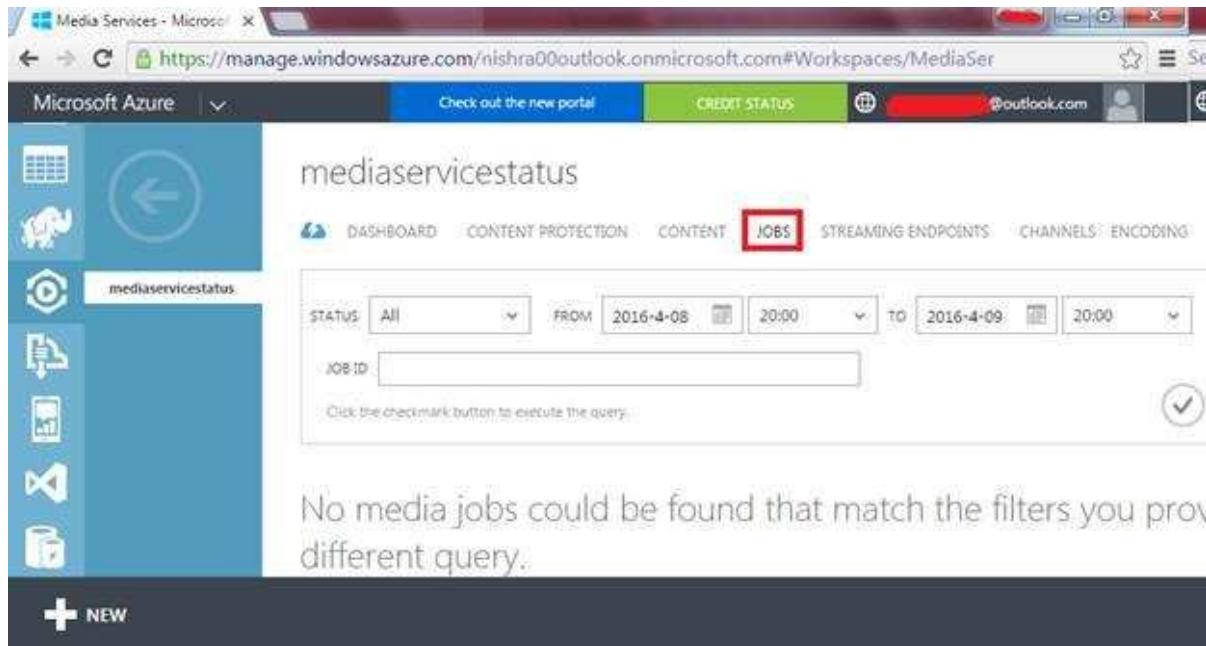
The screenshot shows the 'CONTENT PROTECTION' tab selected in the top navigation bar. It displays two policy sections: 'default aes clear key service policy' and 'default playready license service policy'. Each section has an 'OPEN' button and a 'TOKEN' button. Below these are tabs for 'LICENSE AUTHORIZATION POLICY' (with 'OPEN' and 'TOKEN' buttons) and 'LICENSE TYPE' (with 'NON-PERSISTENT' and 'PERSISTENT' buttons). Under 'PASSING VIDEO CONTENT TO UNKNOWN OUTPUTS', the 'ALLOWED' option is selected. The 'ALLOW TEST DEVICES' switch is set to 'ON'. At the bottom are green buttons for 'import policy xml' and 'export policy xml', and a link to 'Check out PlayReady client options'.

Now the content tab is linked to the Azure Storage that would store the media files that we upload. It is actually an Azure Blob Storage that would store the input files that I need to encode in packaged. It will also store the output files that would be result of encoding and packaging.



The screenshot shows the 'CONTENT' tab selected in the top navigation bar. On the left, there's a sidebar with icons for Dashboard, Content Protection, Content (selected), Jobs, Streaming Endpoints, Channels, and Encoding. A 'NEW' button is at the bottom of the sidebar. The main area displays the message 'You have no media content. Get started here.' and an 'UPLOAD CONTENT' button. There's also an 'UPLOAD' button at the bottom right.

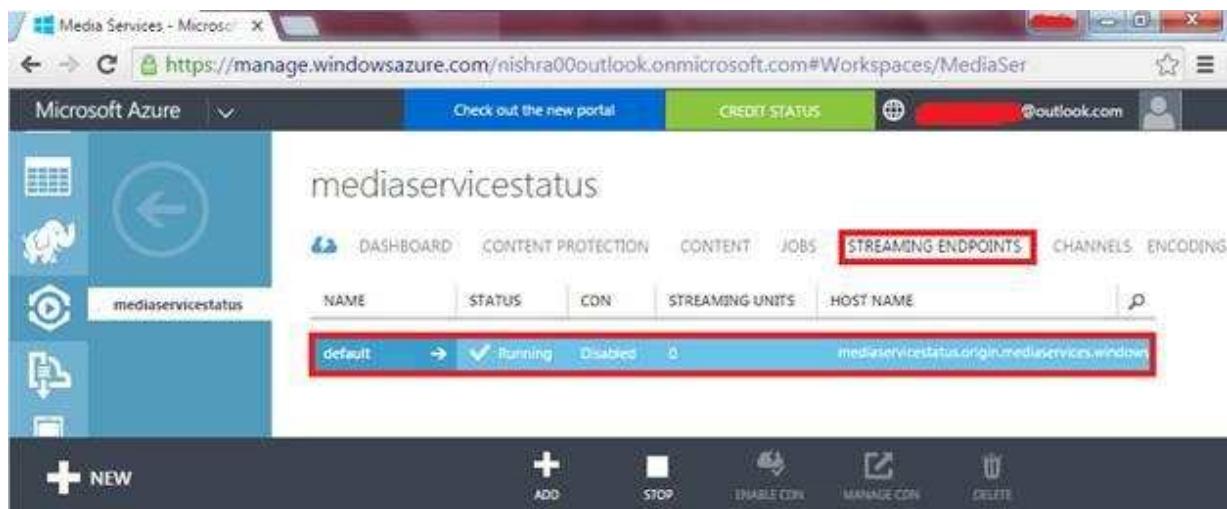
In the *JOB*S tab we can see all the jobs that are currently in progress in Azure Media Services. So if I have encoding jobs that's currently in progress so we can monitor that job over here.



Streaming Endpoint represents a streaming service that can deliver content directly to a client player application, or to a Content Delivery Network (CDN) for further distribution. Streaming Endpoints behave like a server and these are responsible to stream your contents.

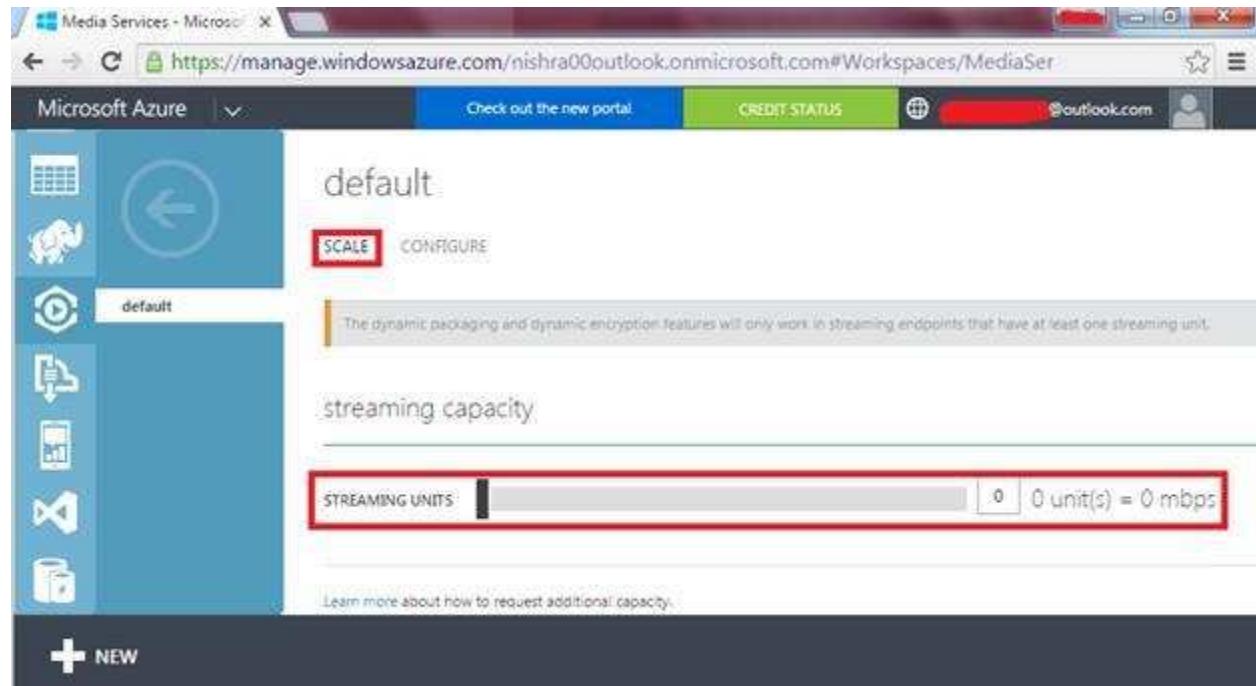
Looks like in figure there is a default streaming endpoint which cannot be deleted.

Click on it to open its other settings.

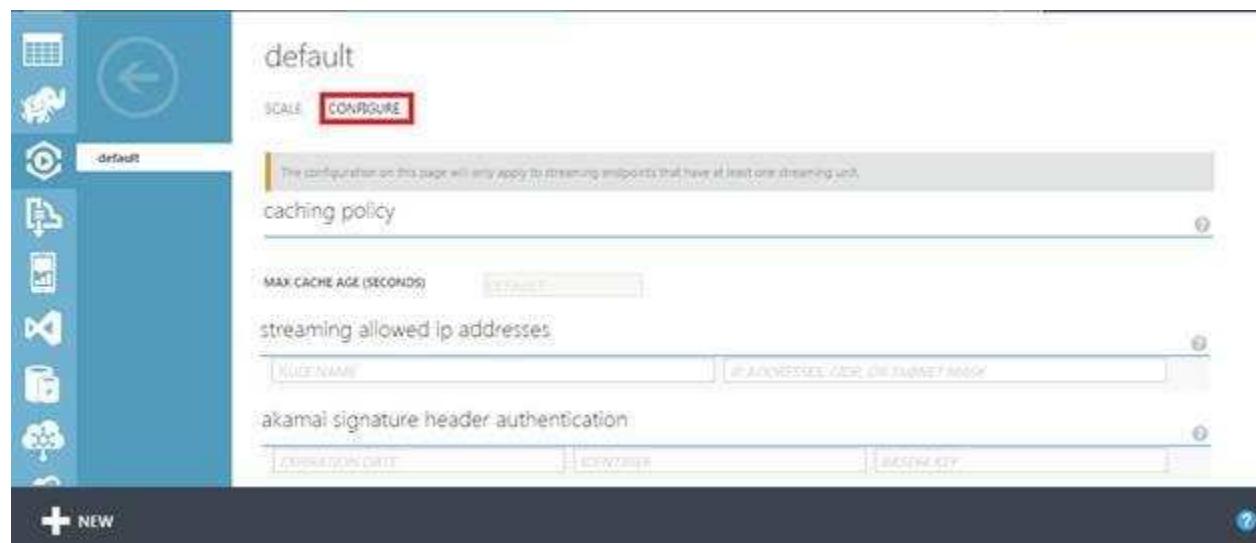


NAME	STATUS	CON	STREAMING UNITS	HOST NAME
default	running	Disabled	0	mediaservicestatusorigin.mediaservices.windows.net

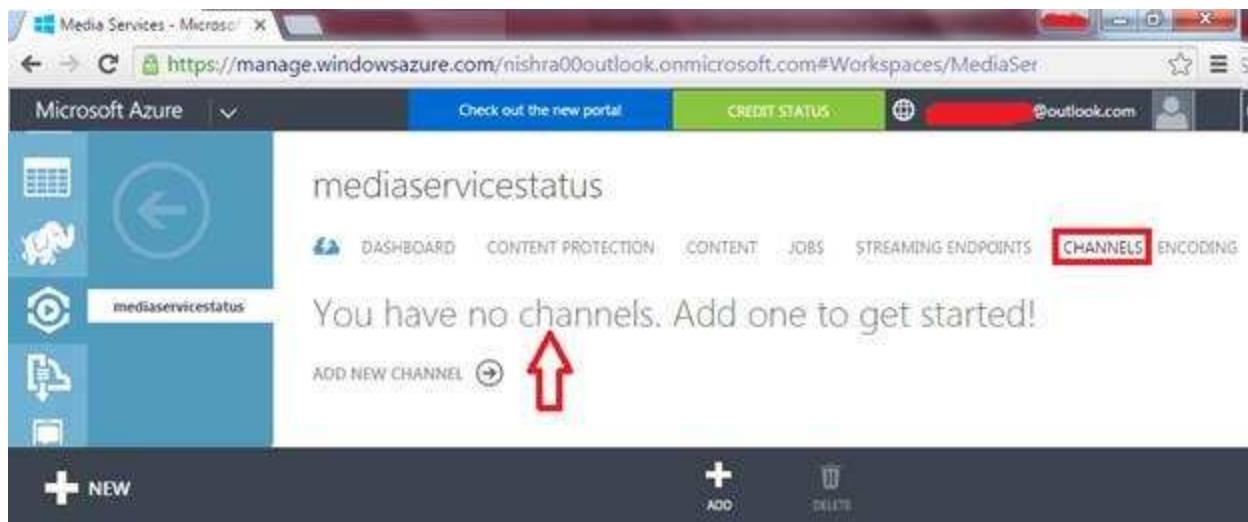
We can manage our scale of point of the receiver end in the media. We can manage the streaming capacity with the streaming units. For increasing streaming units move the reserved capacity slider.



We can configure some settings for the actual streaming like including the CDN settings and IP Addresses that are allowed.

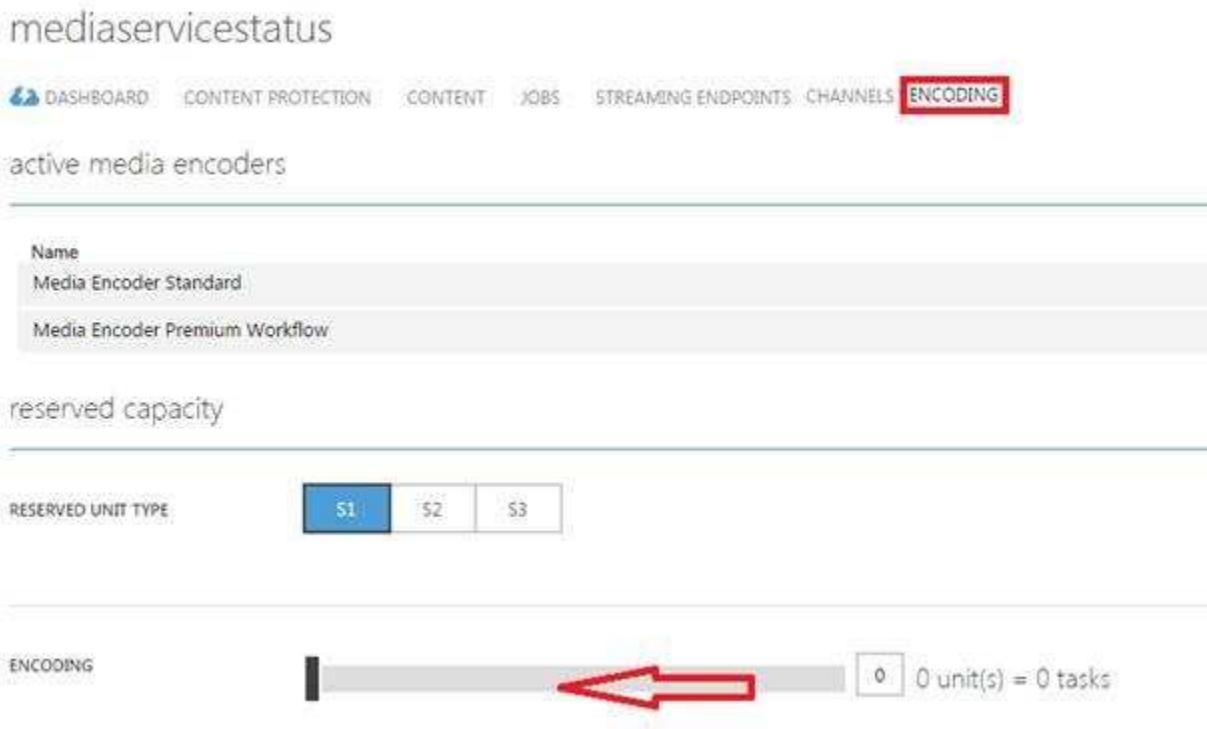


Channel entity represents the pipeline for processing live streaming content. A channel receives live input streams.



The screenshot shows the Microsoft Azure Media Services interface. The URL in the address bar is <https://manage.windowsazure.com/nishra00outlook.onmicrosoft.com#Workspaces/MediaServiceStatus>. The top navigation bar includes 'Microsoft Azure', 'Check out the new portal', 'CREDIT STATUS', and a link to 'outlook.com'. Below the navigation is a sidebar with icons for Calendar, Elephant, and Media Service Status, with 'mediaservicestatus' selected. The main content area has tabs for DASHBOARD, CONTENT PROTECTION, CONTENT, JOBS, STREAMING ENDPOINTS, CHANNELS (which is highlighted with a red box), and ENCODING. A message says 'You have no channels. Add one to get started!' with a red arrow pointing to the 'ADD NEW CHANNEL' button. At the bottom are 'NEW', 'ADD', and 'DELETE' buttons.

In the *ENCODING* tab we can see some settings for Azure media encoder and we can treat that setting in here. In this you also have a reserved category available for encoding in units of 200 megabits per second and this is recommended for production use.



The screenshot shows the 'mediaservicestatus' page with the 'ENCODING' tab selected. The main content area displays 'active media encoders' with two entries: 'Media Encoder Standard' and 'Media Encoder Premium Workflow'. Below this is a section for 'reserved capacity' showing 'RESERVED UNIT TYPE' with three options: S1 (selected), S2, and S3. At the bottom, there is an 'ENCODING' section with a progress bar and the text '0 unit(s) = 0 tasks'.

Let's go back into content tab to upload a media.

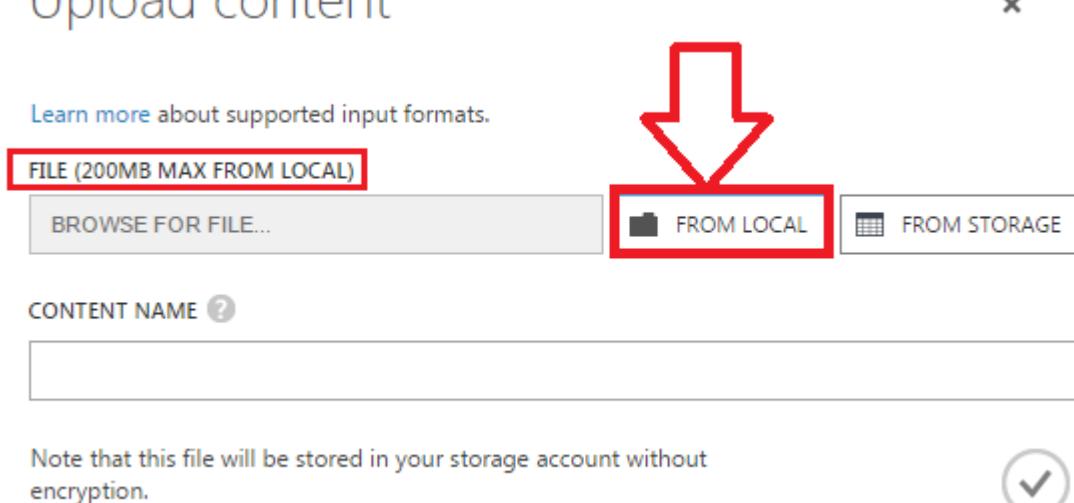
Click on to *Upload* button to upload a media.



The screenshot shows the Microsoft Azure Media Services portal. The URL in the browser is <https://manage.windowsazure.com/nishra00outlook.onmicrosoft.com#Workspaces/MediaServiceStatus>. The page title is "mediaservicestatus". The top navigation bar includes "Microsoft Azure", "Check out the new portal", "CREDIT STATUS", and "outlook.com". Below the navigation is a sidebar with icons for Dashboard, Content Protection, Content (which is selected), Jobs, Streaming Endpoints, Channels, and Encoding. The main content area says "You have no media content. Get started here." with a "UPLOAD CONTENT" button. A large red box highlights the "UPLOAD" button on the right side of the screen.

In the Upload Content choose the file from local computer; make sure file should not be more than 200 MB.

Upload content



The screenshot shows the "Upload content" dialog box. It has a "FILE (200MB MAX FROM LOCAL)" input field with a "BROWSE FOR FILE..." button. Below it are two options: "FROM LOCAL" (highlighted with a red box and a red arrow pointing to it) and "FROM STORAGE". There is also a "CONTENT NAME" input field with a placeholder. At the bottom, a note says "Note that this file will be stored in your storage account without encryption." with a checked checkbox.

It will generate a content name for your media which show that the file is in MP4 format.

Click on to next for further.

Upload content

[Learn more](#) about supported input formats.

FILE (200MB MAX FROM LOCAL)

284965_7d3a7593a473ff0fce4f31d574c314f0dbc3...

FROM LOCAL

FROM STORAGE

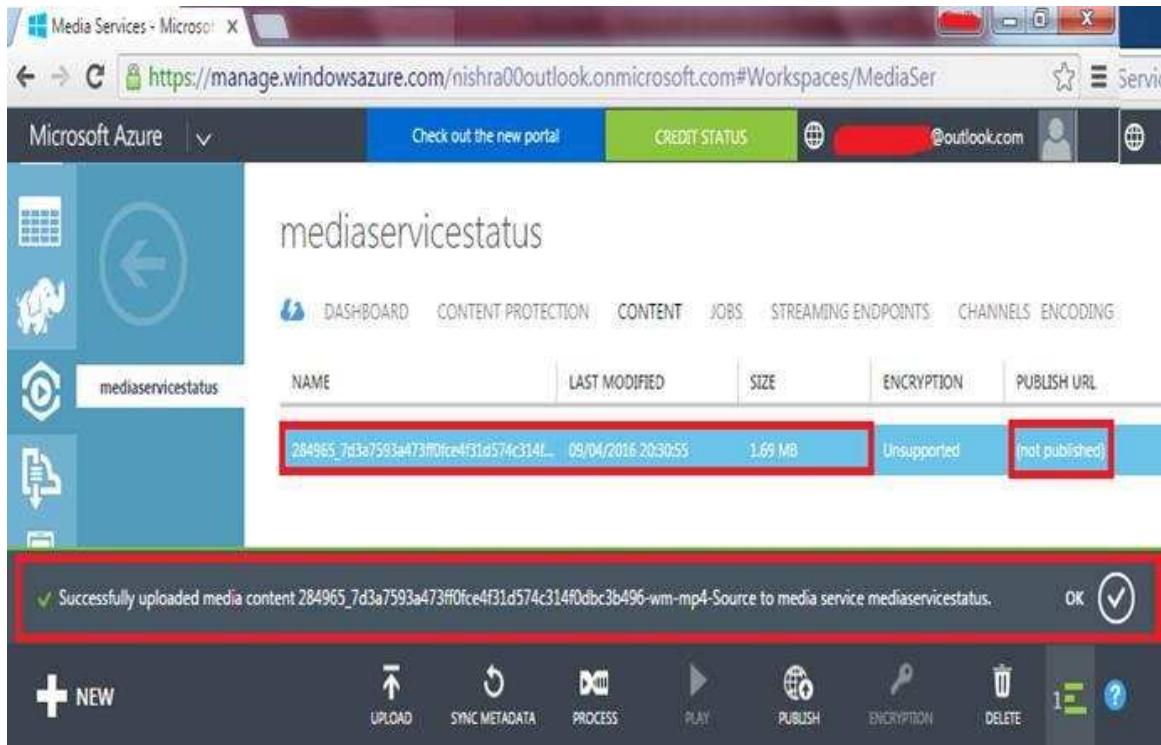
CONTENT NAME 

284965_7d3a7593a473ff0fce4f31d574c314f0dbc3b496-wm-mp4-Source

Note that this file will be stored in your storage account without encryption.



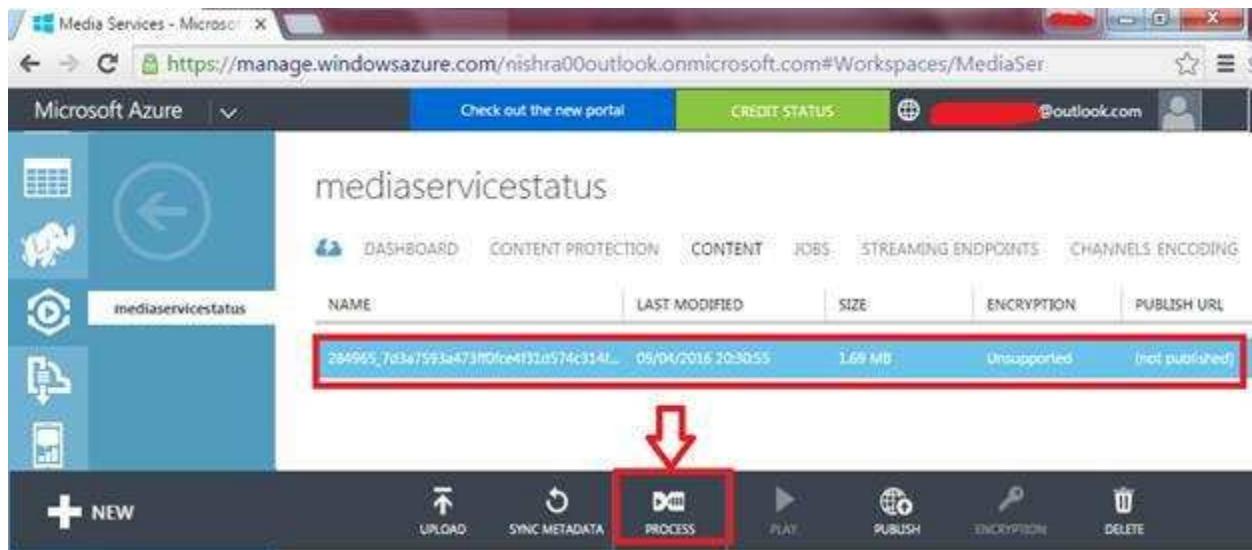
You can see the file is uploaded. It would upload fairly quick so that file is not published yet.



NAME	LAST MODIFIED	SIZE	ENCRYPTION	PUBLISH URL
284965_7d3a7593a473ff0fce4f31d574c314f0dbc3b496-wm-mp4-Source	09/04/2016 20:30:55	1.69 MB	Unsupported	(not published)

Successfully uploaded media content 284965_7d3a7593a473ff0fce4f31d574c314f0dbc3b496-wm-mp4-Source to media service mediaservicestatus.

Now the content is uploaded, now it's time to encode, so select the file and click on to "PROCESS" in the bottom of the window.

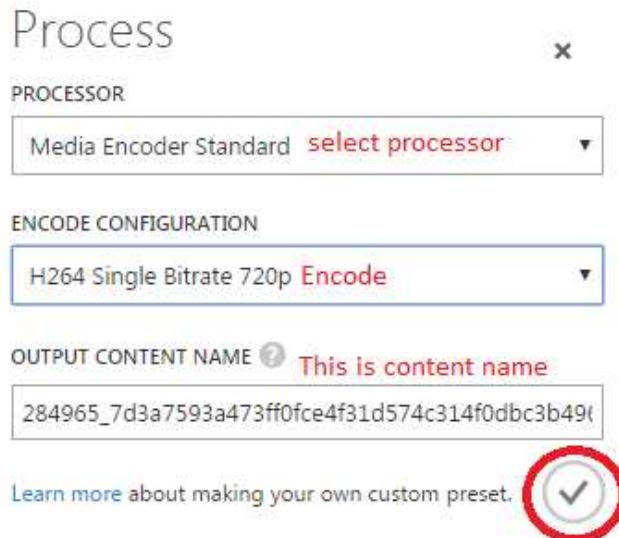


Now select a processor according to your file format.

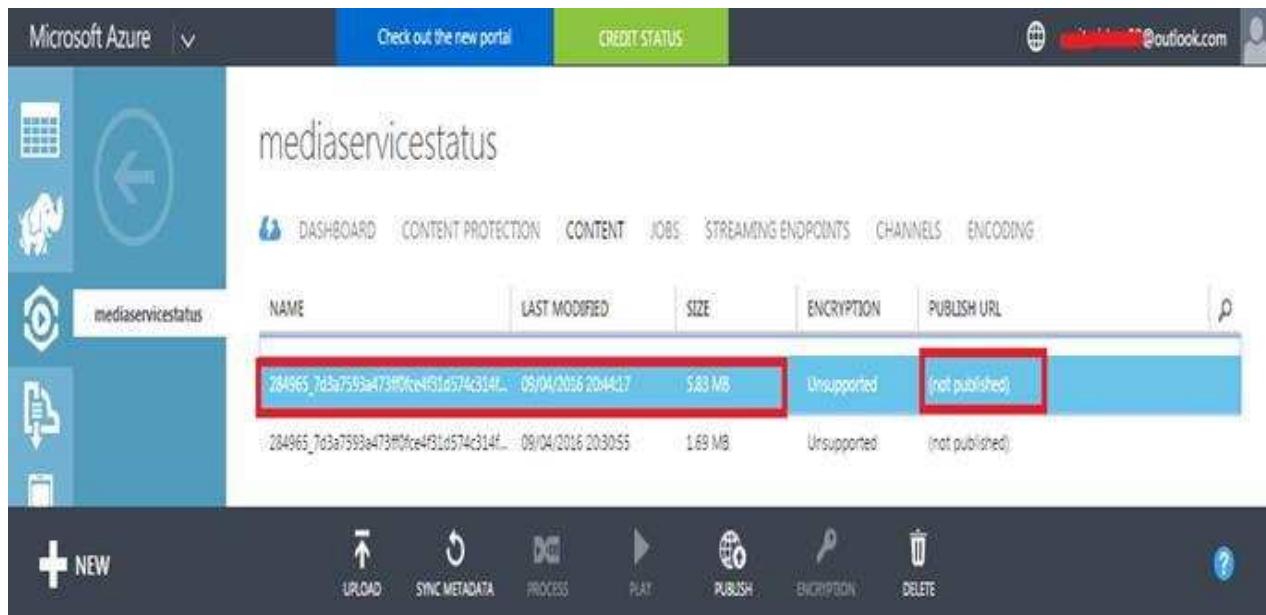
In Encode Configuration, there will a wide range of encoding and encode this file into H264 Single Bitrate 720P.

It takes *Output Content name* by default.

Click on to check to process it.

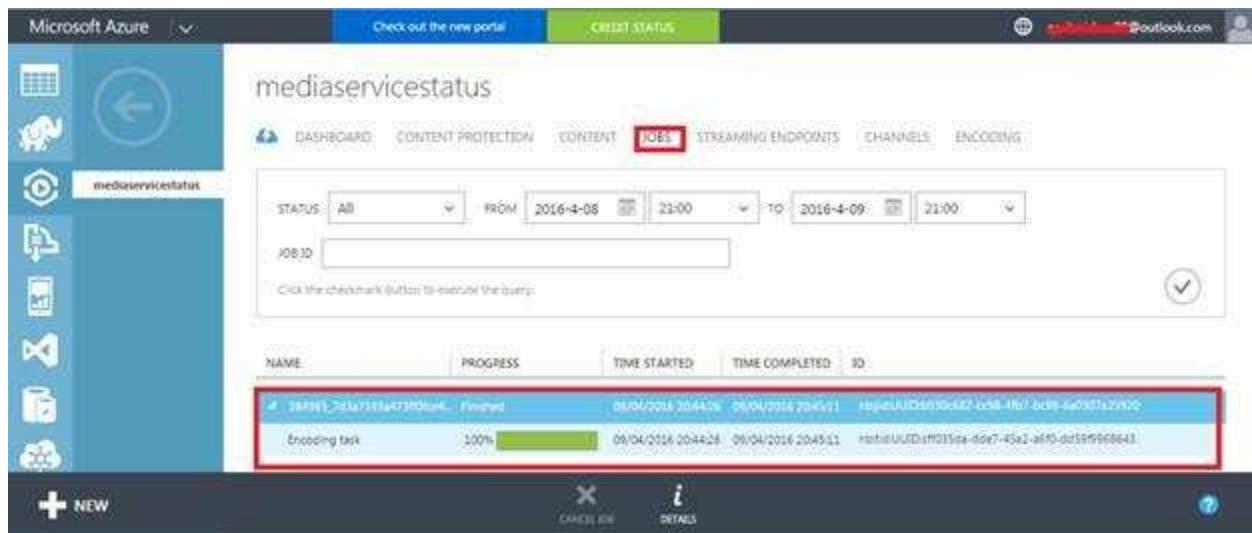


So the encoding job is being created for this particular file in the following screenshot, which is not published yet.



The screenshot shows the Microsoft Azure Media Services Content page. On the left sidebar, the 'mediaservicestatus' option is selected. The main area displays a table with two rows of media files. Both files have the same name, size (5.83 MB), and encryption status ('Unsupported'). The second file's publish URL status is also '(not published)'. The entire row for the second file is highlighted with a red box.

You can monitor its progress in the *JOBS* tab, so click on to Jobs to see its monitoring. You can monitor the status of your encoding task as green color.



The screenshot shows the Microsoft Azure Media Services Jobs page. The 'JOBS' tab is selected. A search bar allows filtering by status (All), date range (From 2016-04-08 To 2016-04-09), and time range (23:00 to 21:00). Below the search bar is a note: 'Click the checkmark button to execute the query.' The table lists one job entry: 'Encoding task' with a status of 'Finished', progress at 100%, and a timestamp from 09/04/2016 20:44:26 to 09/04/2016 20:45:11. The job ID is 284965_7e3a7593a473f0ce4f31d574c314f.... The entire row for the job is highlighted with a red box.

Go back into the Content, you would see the new file will be created and is not published.

Select the file and just click on to *PUBLISH* button to publish it.

It will look up for confirmation, click on to yes.



The screenshot shows the Microsoft Azure Media Services status page. A video file named "264963_7d3a7593a473f09e4f31d574c314f..." is listed. The "PUBLISH URL" column for this file is highlighted with a red box. Below the table, there is a row of buttons: UPLOAD, SYNC METADATA, PROCESS, PLAY, UNPUBLISH, ENCRYPTION, and DELETE. The "PUBLISH" button is also highlighted with a red box and has a red arrow pointing to it from below.

After clicking on to Publish, we will get the publish URL generated by the Azure Media Services. Click to copy the URL and paste into a browser tab. This publish URL is accessible on Internet.



The screenshot shows the Microsoft Azure Media Services status page after publishing. The "PUBLISH URL" column for the video file now contains a URL: "http://mymediaservicestatus.northcentralus.cloudapp.net/assets-...". A red box highlights this URL, and a small clipboard icon is shown next to it, indicating it has been copied.

You can see the video media player is running on to browser, which we just encoded.



I hope you enjoyed this chapter.

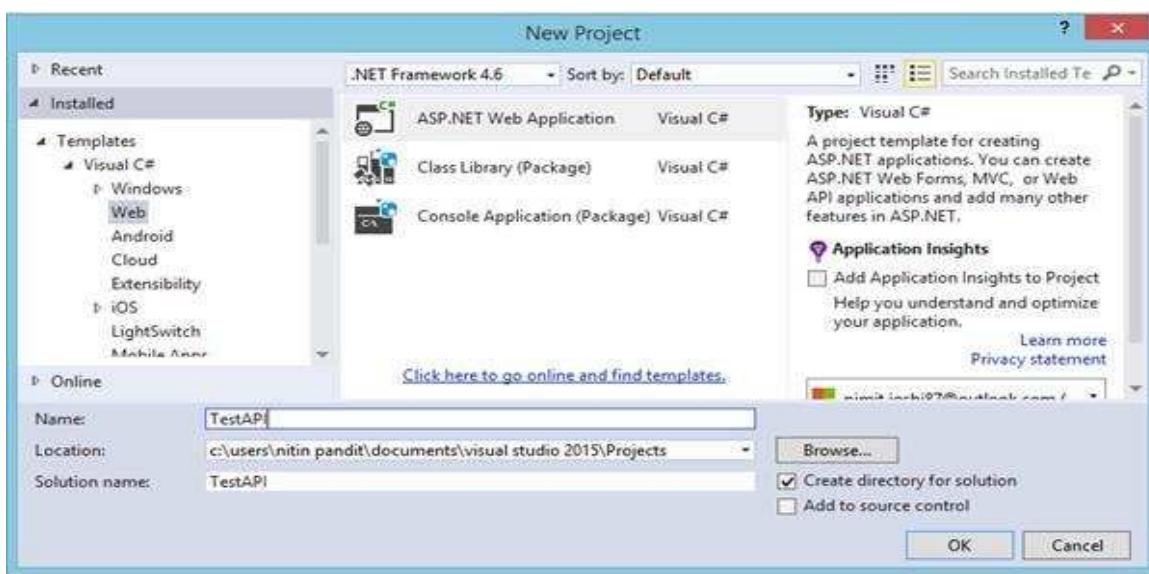
How to Use API Apps in Azure

Introduction

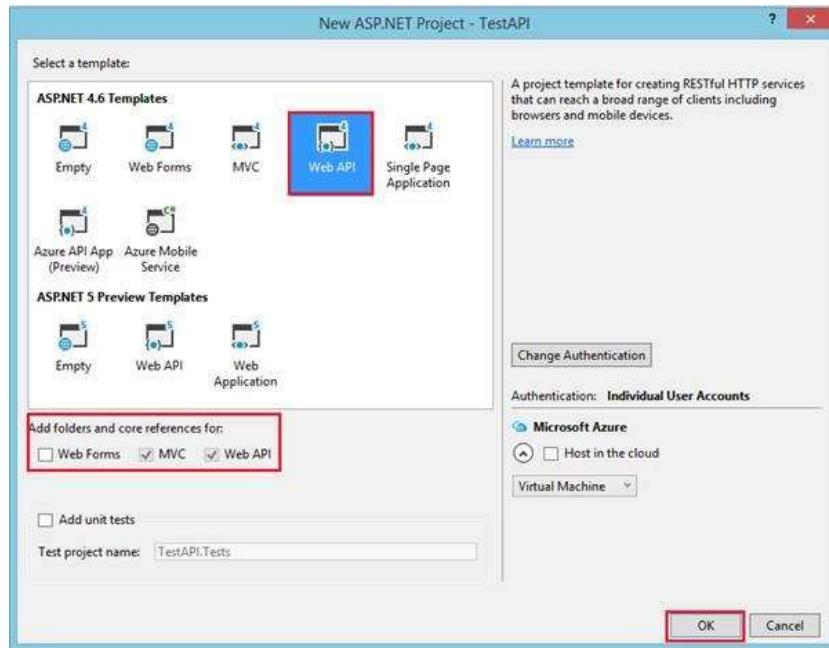
In this chapter you'll learn how to use API Apps from the Azure portal to host your RESTful APIs and the data live. Before understanding or using API Apps in Azure, just go through the actual benefits of App Services.



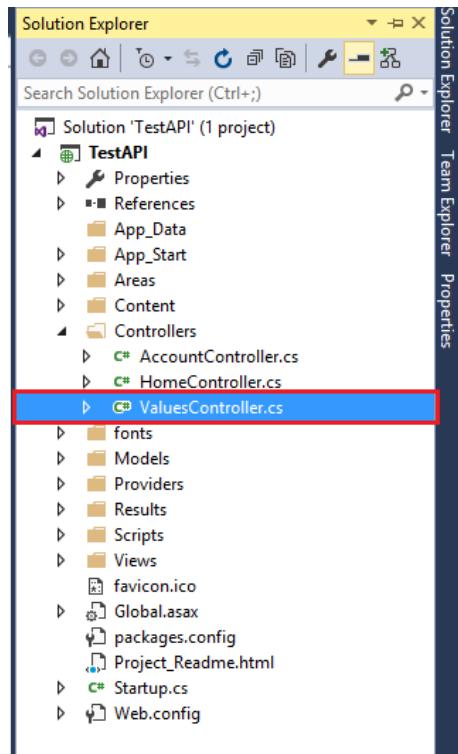
To use API Apps services from Azure, first create your rest service. Open your Visual Studio and create a web application with Web API references.



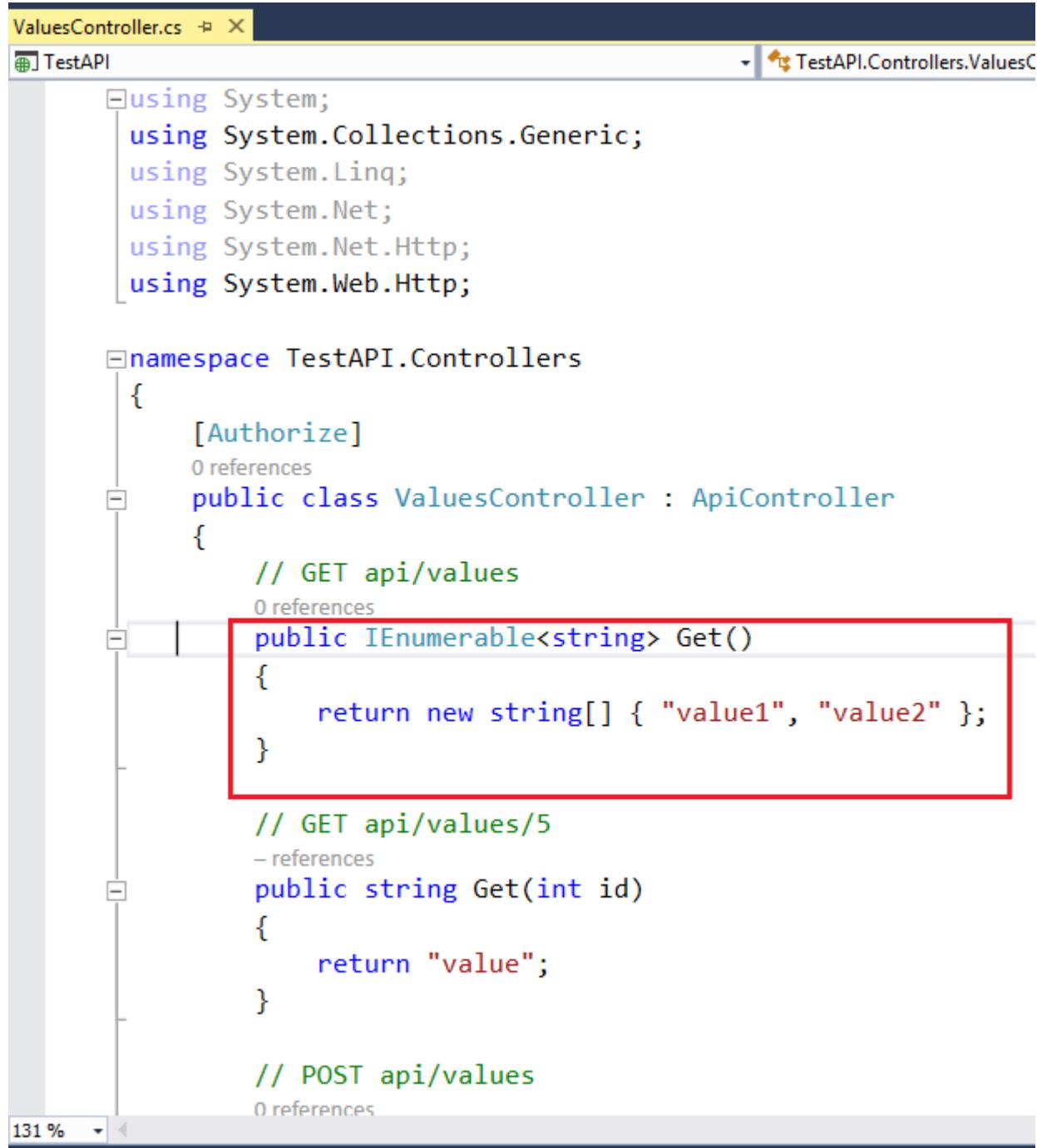
Select a template with Web API to get a default readymade API controller by Visual Studio with the name “ValuesController”.



This is your readymade solution for API in Visual Studio with a Values name Controller.



There is an httpGet method for testing. So, I am not going to make a new one because I can test this only.



The screenshot shows the code editor for `ValuesController.cs` in a .NET Core project named `TestAPI`. The code defines a `ValuesController` class that inherits from `ApiController`. It contains two methods: `Get()` and `Get(int id)`. The `Get()` method is highlighted with a red rectangle. The `Get(int id)` method returns the string "value".

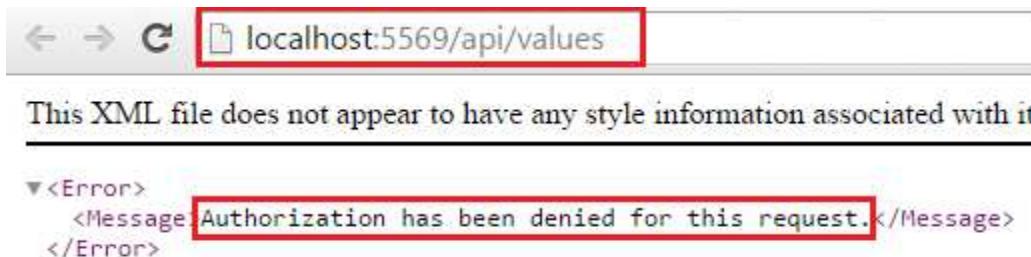
```
ValuesController.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace TestAPI.Controllers
{
    [Authorize]
    public class ValuesController : ApiController
    {
        // GET api/values
        public IEnumerable<string> Get()
        {
            return new string[] { "value1", "value2" };
        }

        // GET api/values/5
        public string Get(int id)
        {
            return "value";
        }

        // POST api/values
    }
}
```

If I run this on localhost the first time you'll get exception because your default API controller is having an `Authorize` attribute. So if you want just testing, go to controller file.



This XML file does not appear to have any style information associated with it

```
<Error>
  <Message>Authorization has been denied for this request.</Message>
</Error>
```

Now remove this, without removing it you have to authorize for this application.



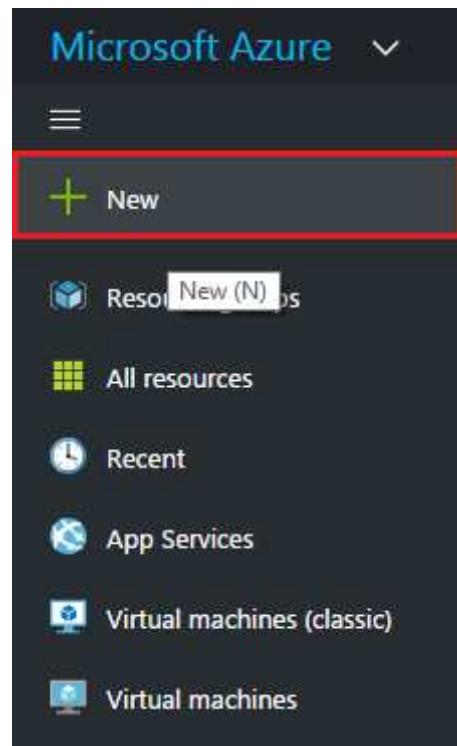
Now if you run your application, you'll get your data from values controller as XML document.



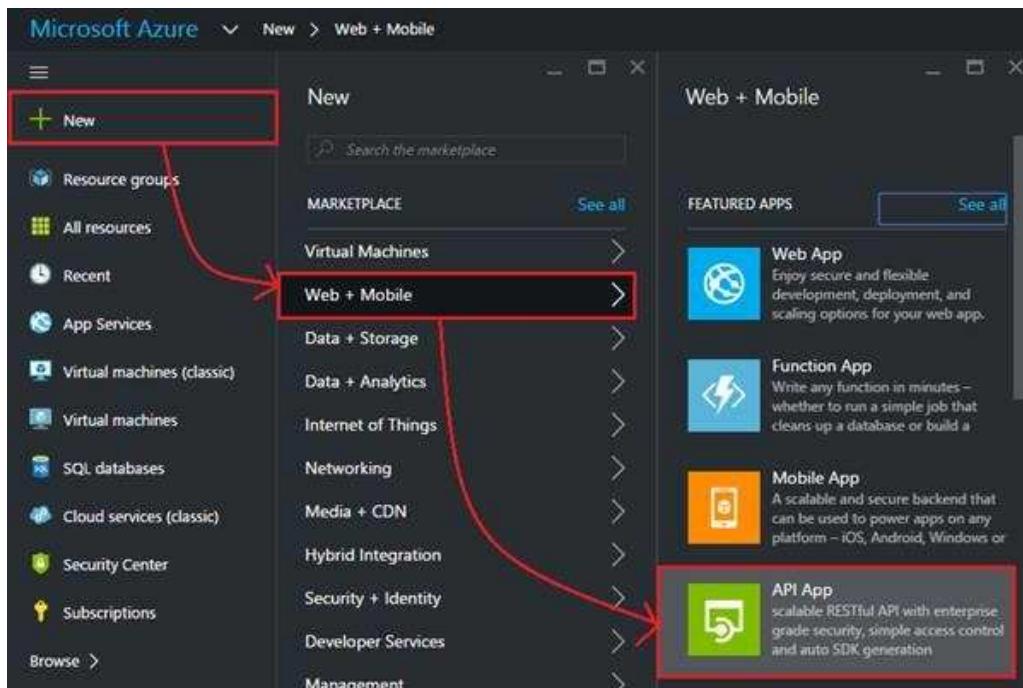
This XML file does not appear to have any style information associated with it

```
<ArrayOfstring xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <string>value1</string>
  <string>value2</string>
</ArrayOfstring>
```

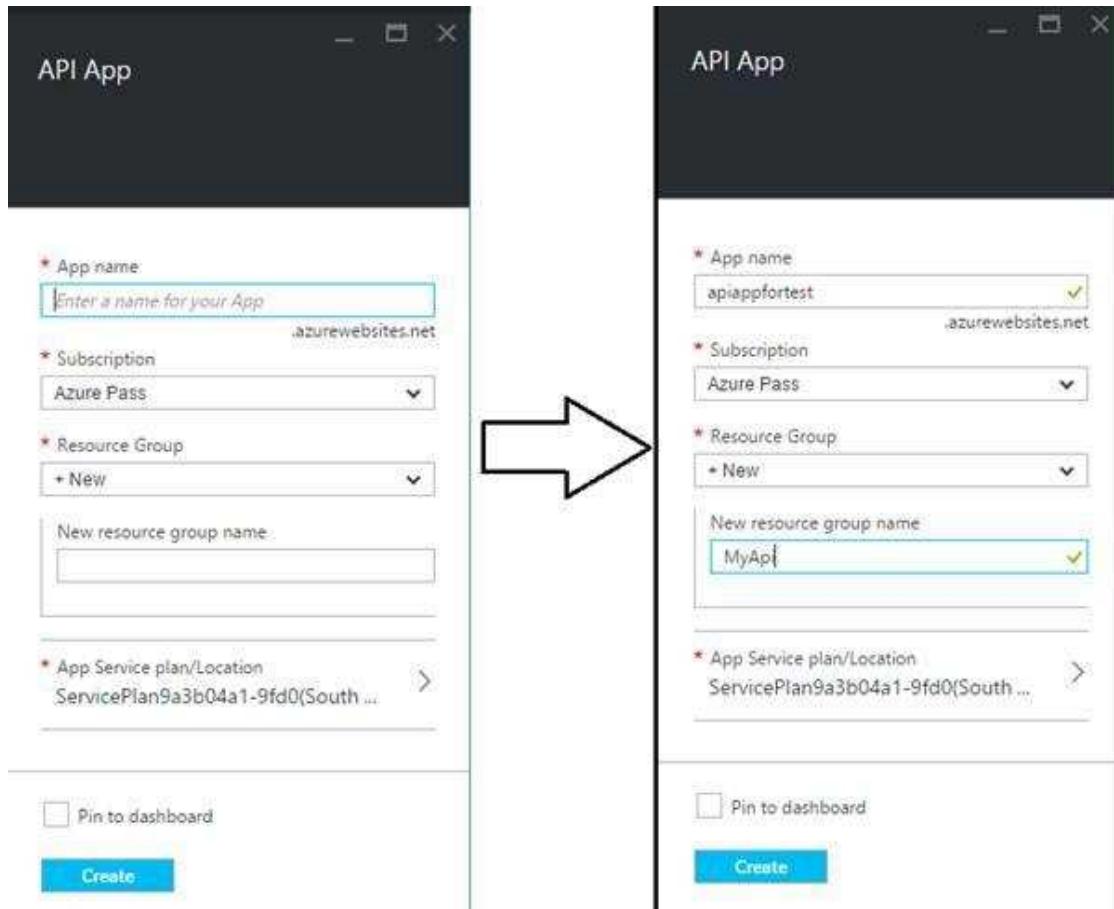
Now login with your account and follow the below given steps.



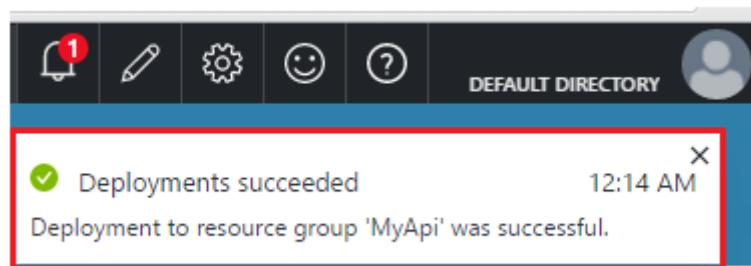
Click on *New, Web + Mobile*, then *API App*.



Just fill the details for your API to make a space at Azure with the application name.



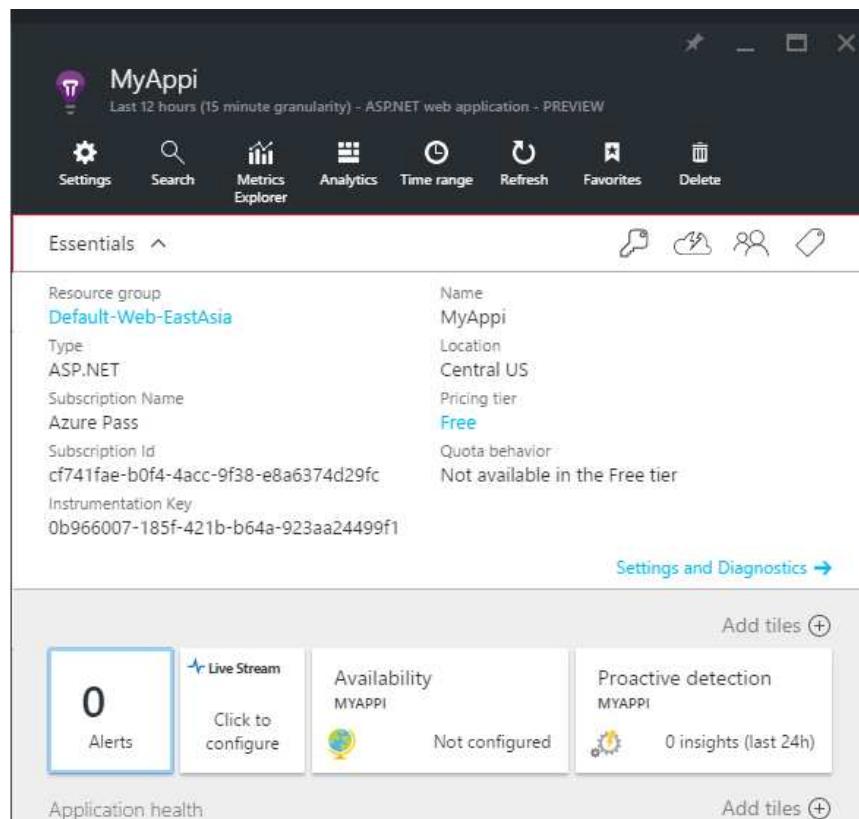
After clicking on *Create* button you will get notification for success.



Now you can see your API App service in Dashboard.



If you want to see more details, just click on the name of API app Service.

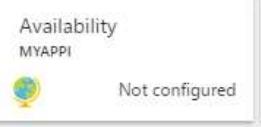


Resource group	Name
Default-Web-EastAsia	MyAppi
Type	Location
ASP.NET	Central US
Subscription Name	Pricing tier
Azure Pass	Free
Subscription Id	Quota behavior
cf741fae-b0f4-4acc-9f38-e8a6374d29fc	Not available in the Free tier
Instrumentation Key	
0b966007-185f-421b-b64a-923aa24499f1	

[Settings and Diagnostics →](#)

Add tiles 
 Click to configure

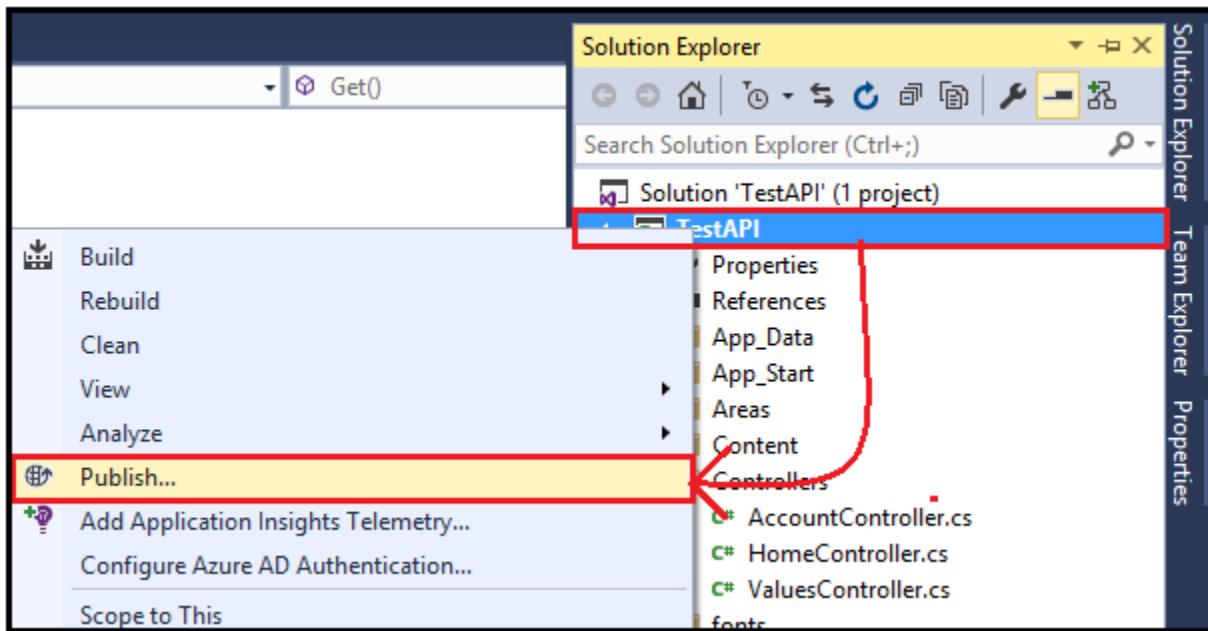

0
 Alerts


MYAPPI
 Availability

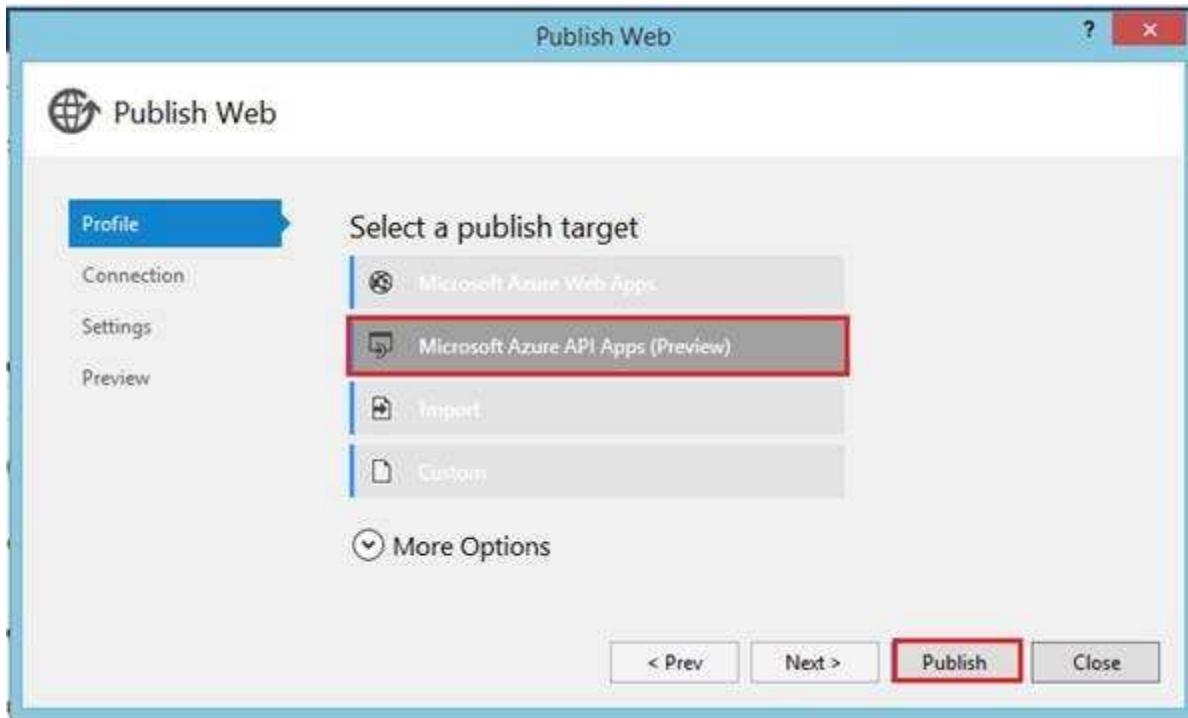

MYAPPI
 Proactive detection

Application health
Add tiles 

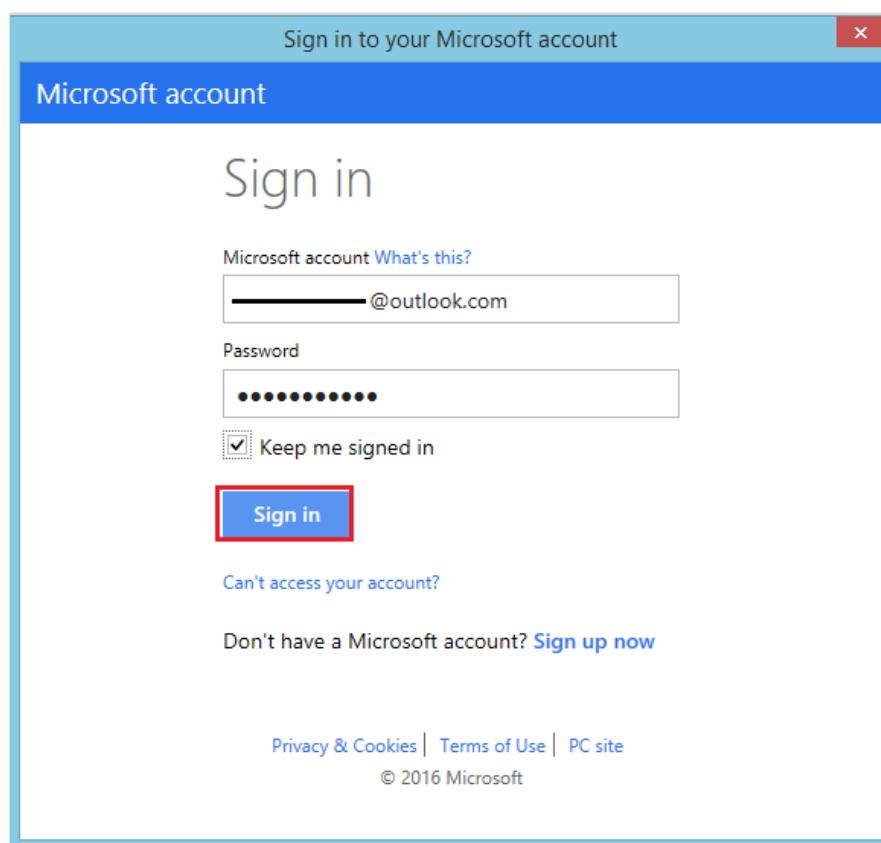
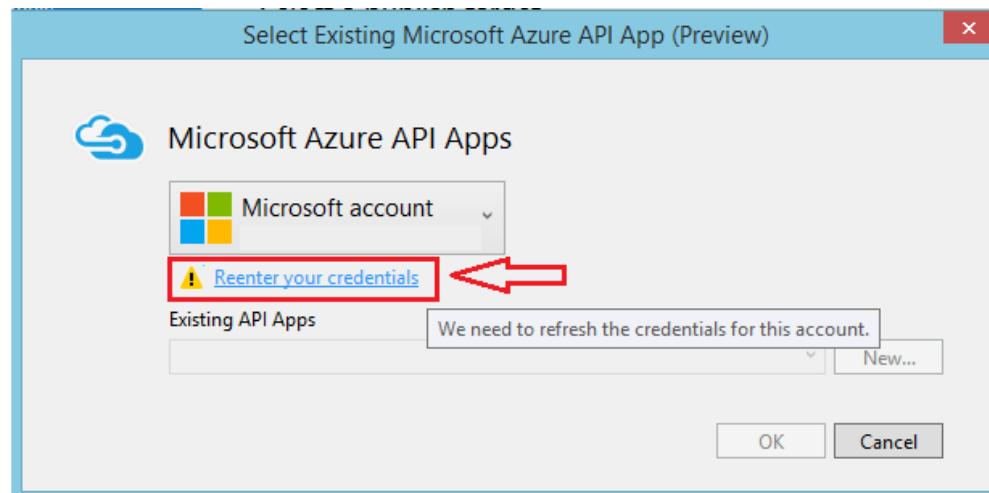
Now just go to your solution in Visual Studio and right click on application name, then find *Publish*.



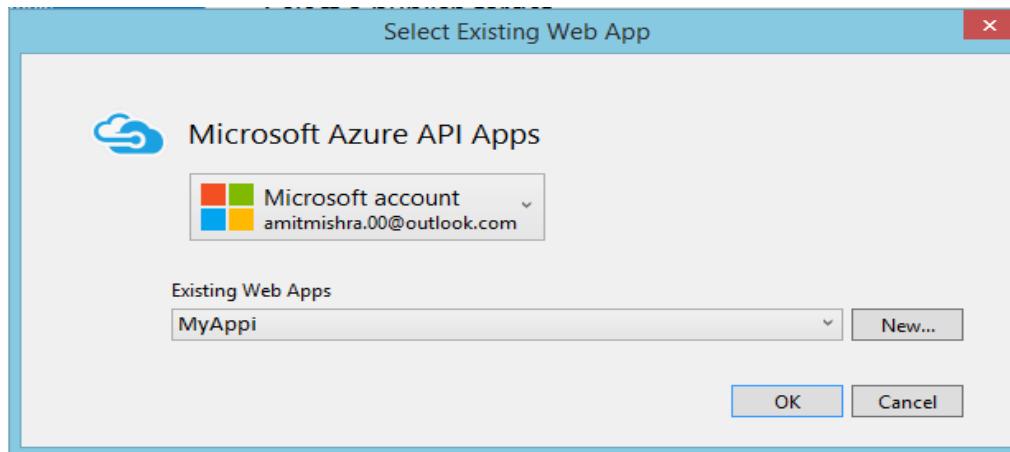
Select *Microsoft Azure API Apps*.



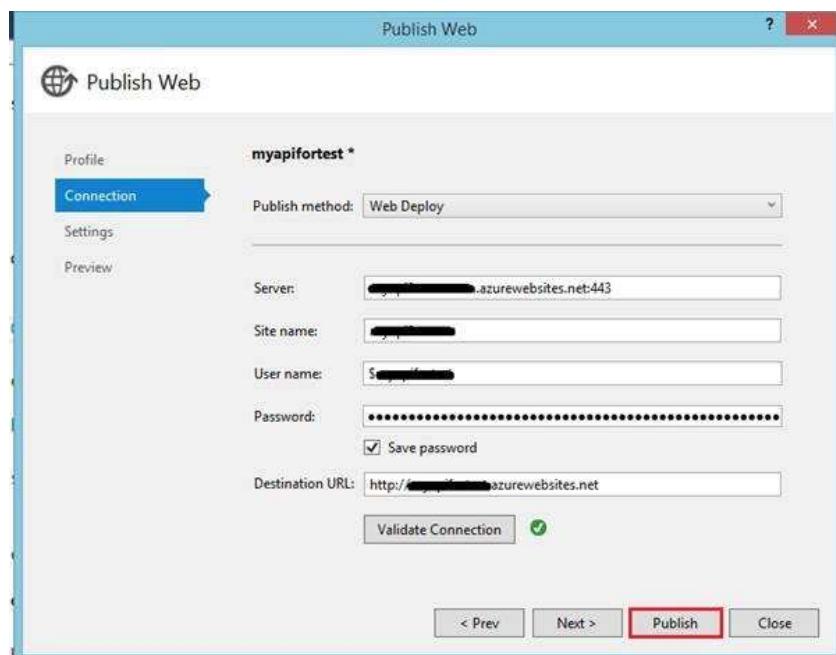
Click on next or publish and login with your Azure Account in Visual Studio.



Now you can find your already created APIs or you can create new directly from Visual Studio without using Azure Portal.



After that, I'll get all DNS or FTP details and allow you to publish.



Now just click on publish and browse this into browser, you'll get your API data.



Manage Data in Microsoft Azure Machine Learning

In this chapter, we'll learn about how we can generate titanic data in Azure Machine Learning. Microsoft Azure Machine Learning is a data modeling environment from which we can get an end-to-end approach to a problem to an answer. Data is summarized according to the status like class, age, sex, survival and this is a format of an ocean liner (Titanic).

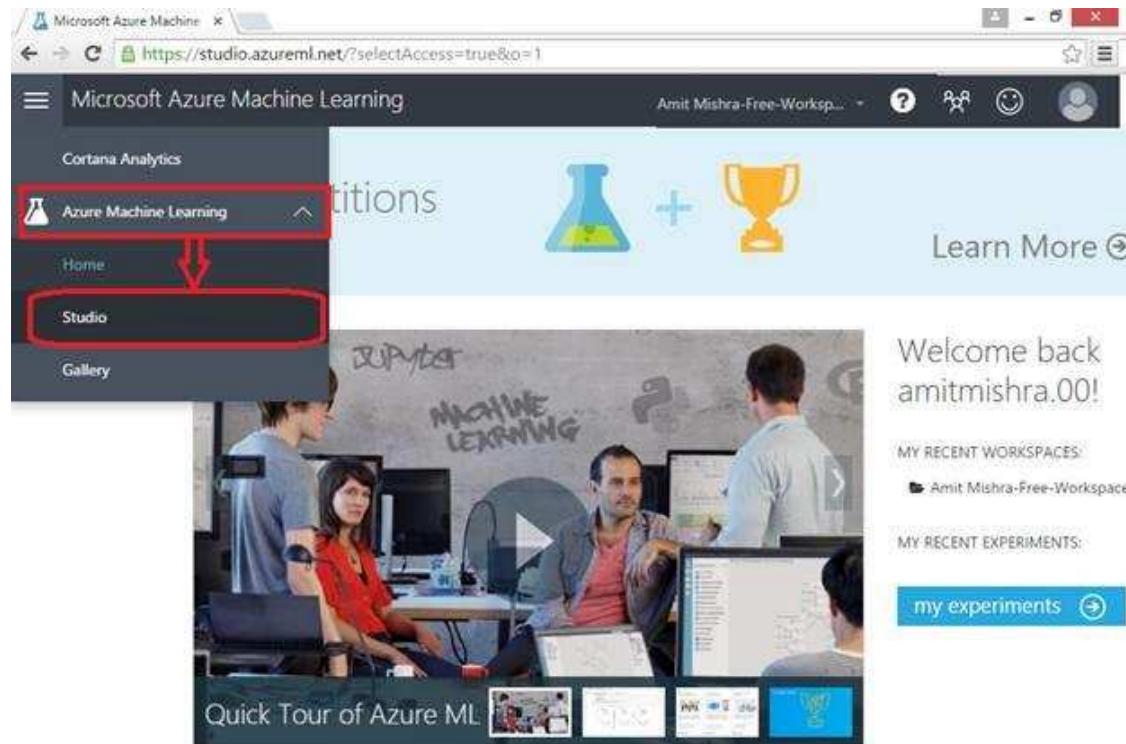
Firstly, you sign in to Microsoft Azure, after sign in go to the URL.

Azure Machine Learning page will open, make sure you must have to login into Azure.

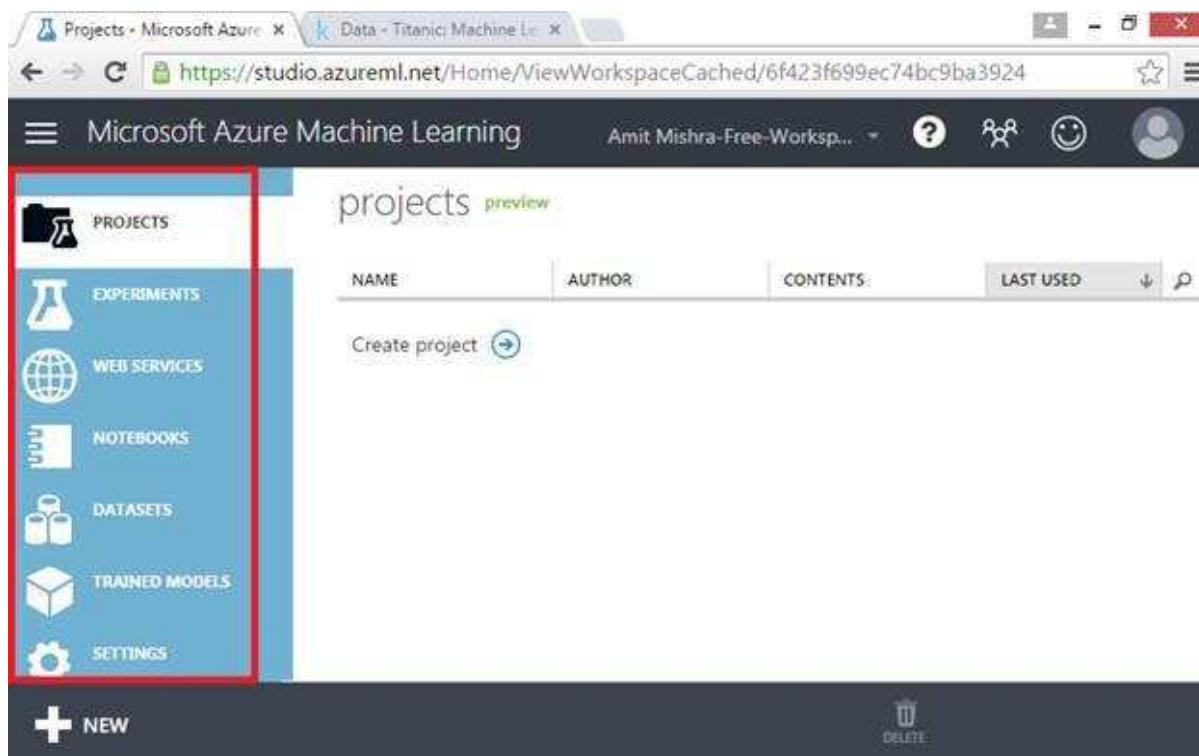
In the top left of the window, expand the details. (Shown in rectangle)



The menu for Azure Machine will open, inspect it and click on it to open its sub menu. There would be *Studio*, click on it. Azure machine Learning page will open.



The following screenshot shows there is a number items related to machine learning.



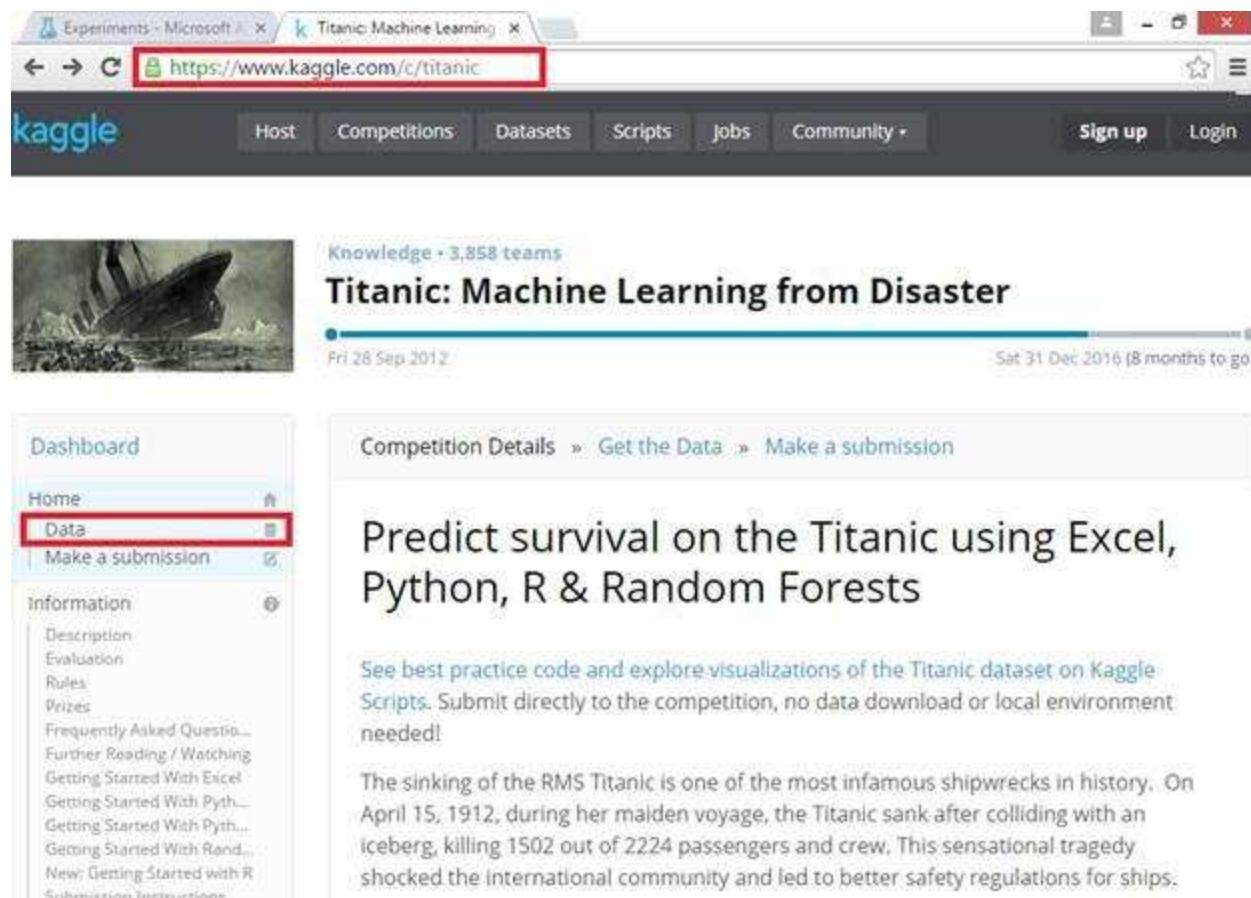
This screenshot shows a workspace within the Microsoft Azure Machine Learning Studio. The URL in the browser is 'https://studio.azureml.net/Home/ViewWorkspaceCached/6f423f699ec74bc9ba3924'. The interface includes a top navigation bar with 'Microsoft Azure Machine Learning' and 'Amit Mishra-Free-Worksp...', and standard UI elements like a question mark icon, a star, a smiley face, and a user profile. On the left, a vertical sidebar is highlighted with a red box, containing links for 'PROJECTS', 'EXPERIMENTS', 'WEB SERVICES', 'NOTEBOOKS', 'DATASETS', 'TRAINED MODELS', and 'SETTINGS'. Below this sidebar is a 'NEW' button with a plus sign. The main content area is titled 'projects preview' and shows a table with columns 'NAME', 'AUTHOR', 'CONTENTS', and 'LAST USED'. A single row is visible with a 'Create project' button. At the bottom right of the content area is a 'DELETE' button.

Next step is, go to the <https://www.kaggle.com/c/titanic>.

Firstly, you will have to login on the kaggle website.

This website is helpful for titanic data and contains very large datasets. It provides direct path of the file to download it in to your local system.

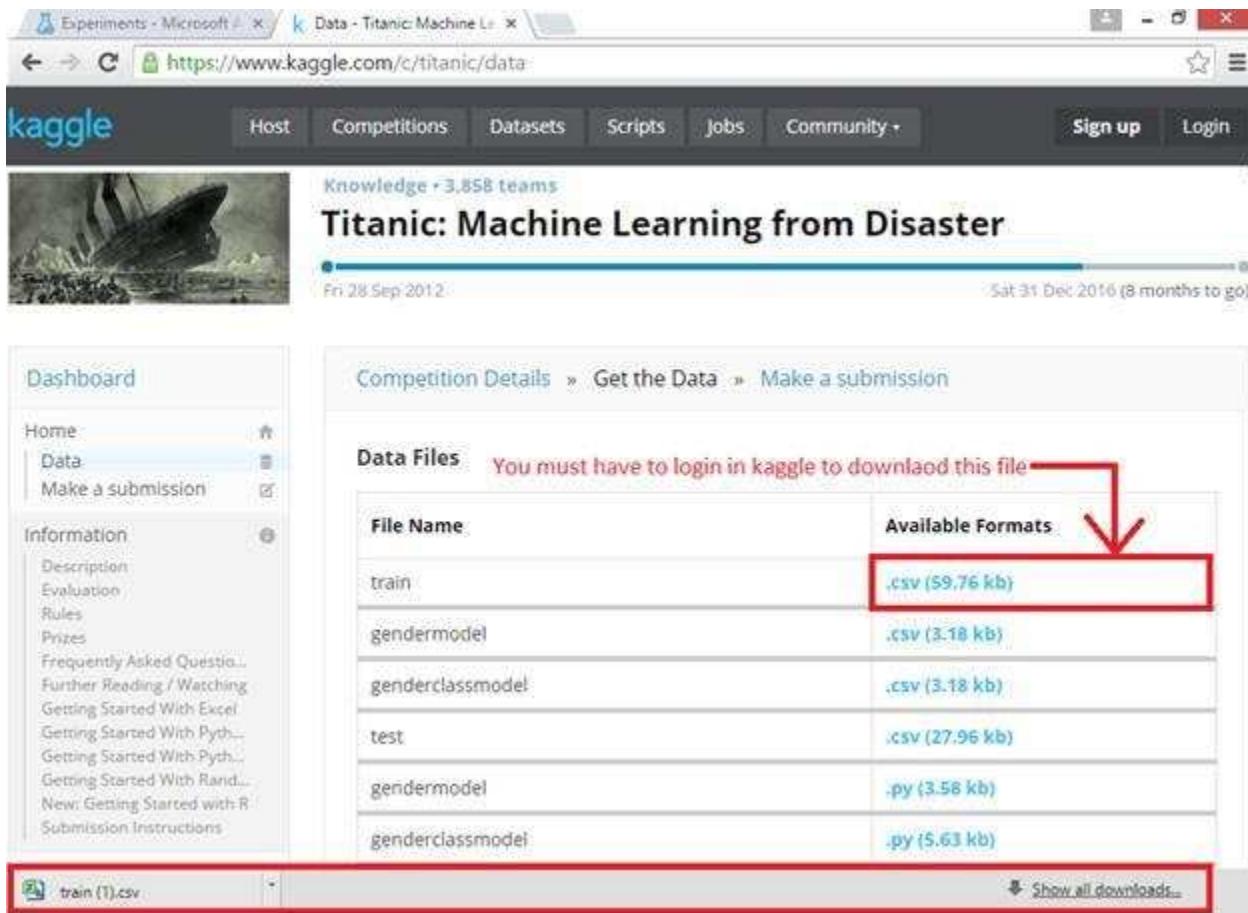
For the dataset, go to “*Data*” tab.



The screenshot shows a web browser window with the URL <https://www.kaggle.com/c/titanic> in the address bar. The page title is "Titanic: Machine Learning from Disaster". On the left, there's a sidebar with links like "Dashboard", "Home", "Data" (which is highlighted with a red box), and "Make a submission". The main content area features a large image of the Titanic shipwreck, the competition title, and dates (Fri 28 Sep 2012 to Sat 31 Dec 2016). Below this, a section titled "Predict survival on the Titanic using Excel, Python, R & Random Forests" contains descriptive text and links to best practice code and visualizations.

Whenever you click on data, there would be many files with their extension like .csv, .py.

We will use data in “*train.csv*” file to train a neural network to recognize digits. Click on to “*train.csv*” file to download it.

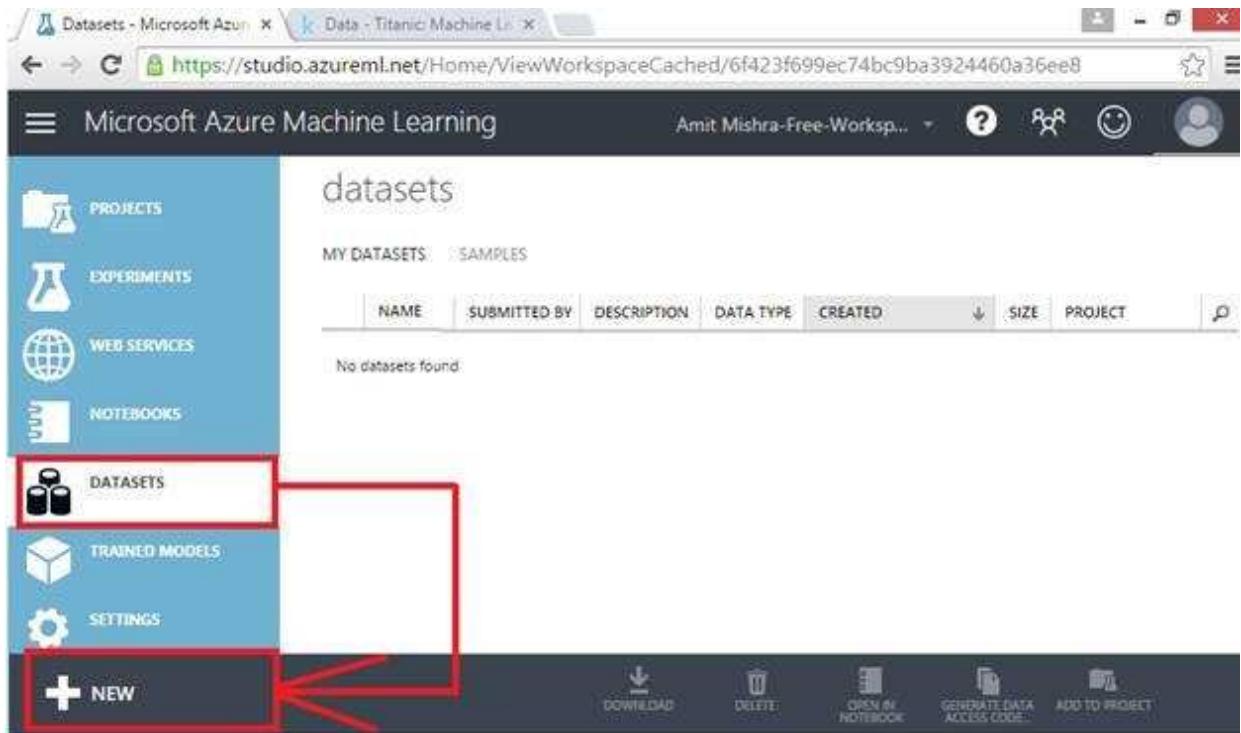


The screenshot shows the Kaggle interface for the "Titanic: Machine Learning from Disaster" competition. On the left, there's a sidebar with links like "Home", "Data", "Make a submission", "Information", and "Description". The main content area shows the competition details, including the title "Titanic: Machine Learning from Disaster", the start date "Fri 28 Sep 2012", and the end date "Sat 31 Dec 2016 (8 months to go)". Below this, there's a "Data Files" section with a table. The first row, "train", has its ".csv (59.76 kb)" link highlighted with a red box and a red arrow pointing to it. At the bottom of the table, there's a link "Show all downloads...".

File Name	Available Formats
train	.csv (59.76 kb)
gendermodel	.csv (3.18 kb)
genderclassmodel	.csv (3.18 kb)
test	.csv (27.96 kb)
gendermodel	.py (3.58 kb)
genderclassmodel	.py (5.63 kb)

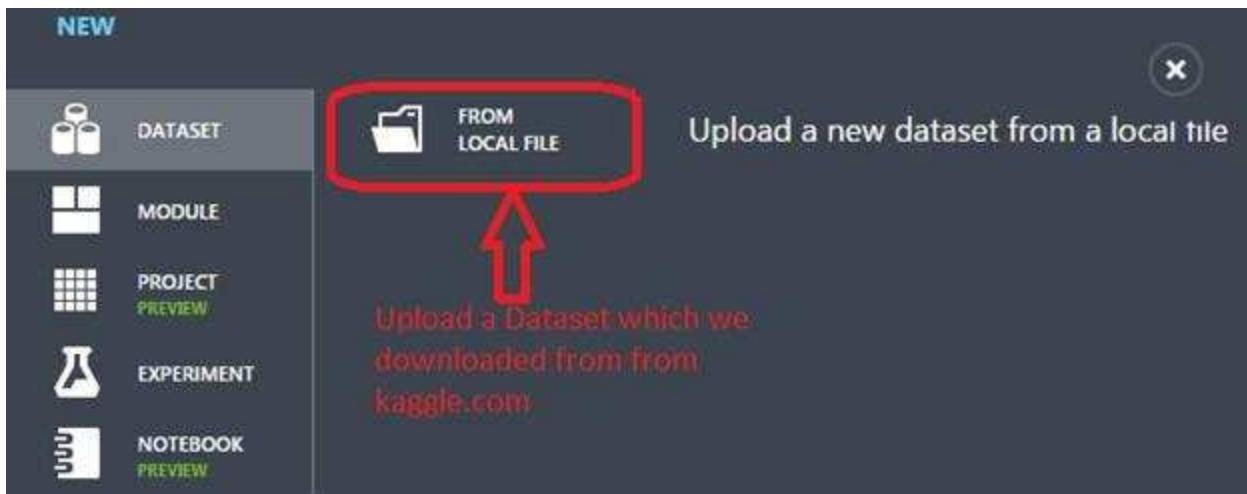
Go back in to the Azureml page; we need to upload that .csv file as a data set as in the following figure.

Create a dataset, click on it and click on to “New” button.



The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left, there's a sidebar with icons for Projects, Experiments, Web Services, Notebooks, Datasets (which is highlighted with a red box), Trained Models, and Settings. At the bottom of the sidebar is a 'NEW' button. The main area is titled 'datasets' and shows a table header for 'NAME', 'SUBMITTED BY', 'DESCRIPTION', 'DATA TYPE', 'CREATED', 'SIZE', and 'PROJECT'. Below the header, it says 'No datasets found.' To the right of the table are buttons for 'DOWNLOAD', 'DELETE', 'OPEN IN NOTEBOOK', 'GENERATE DATA ACCESS CODE', and 'ADD TO PROJECT'. A red box highlights the 'FROM LOCAL FILE' option in a dropdown menu labeled 'Upload a new dataset from a local file'.

It will ask for uploading from local file, so just click and upload that .csv file.



Choose the .csv data file from your location where you downloaded it. Enter a name for the Dataset and click on to check button to create a dataset.

Upload a new dataset

SELECT THE DATA TO UPLOAD:

train.csv **File name would be train.csv**

This is the new version of an existing dataset

ENTER A NAME FOR THE NEW DATASET:

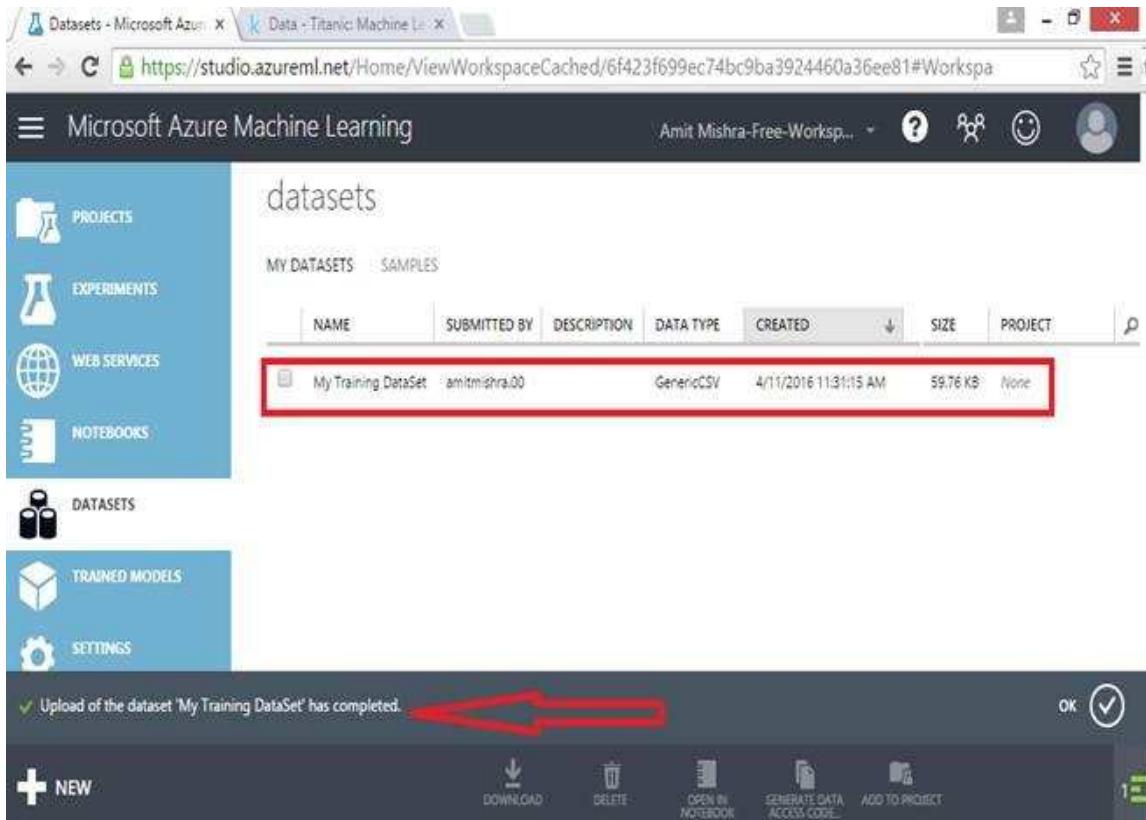
Enter a name for DataSet

SELECT A TYPE FOR THE NEW DATASET:

PROVIDE AN OPTIONAL DESCRIPTION:

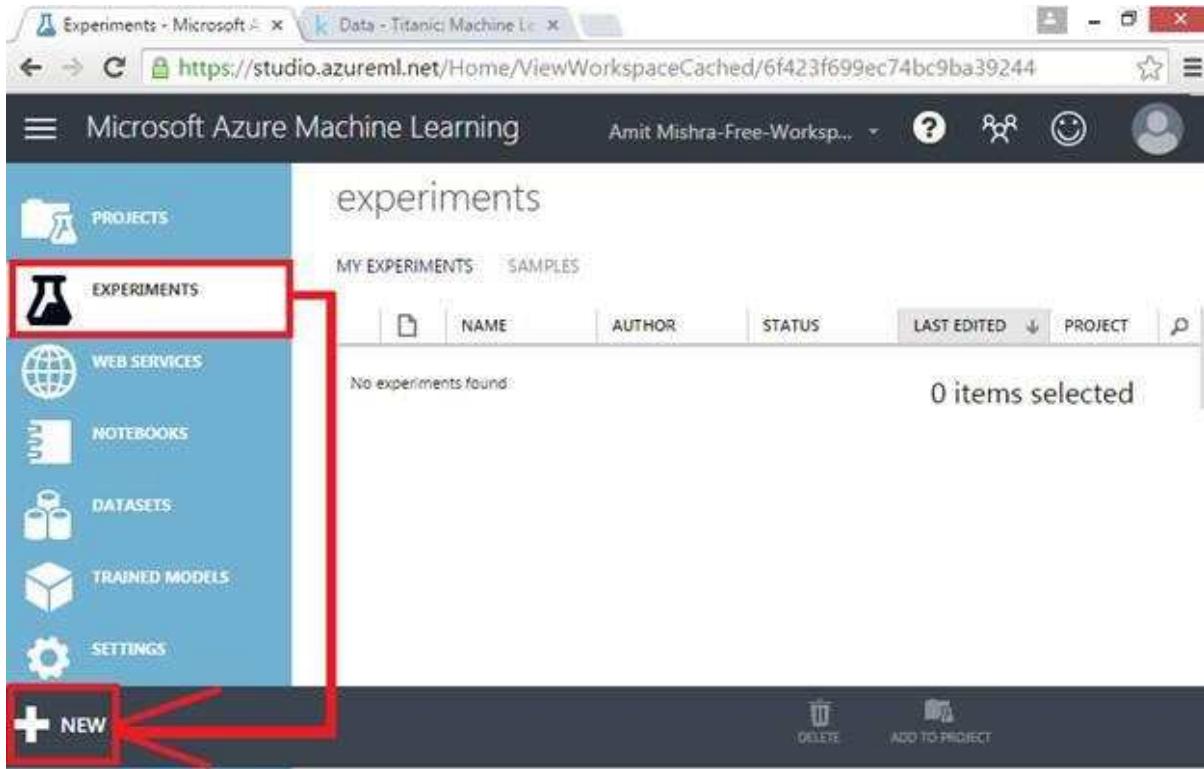


After clicking on check button, a new dataset created; in my norms my dataset name is “*My Training DataSet*”. Now we need to create an experiment.

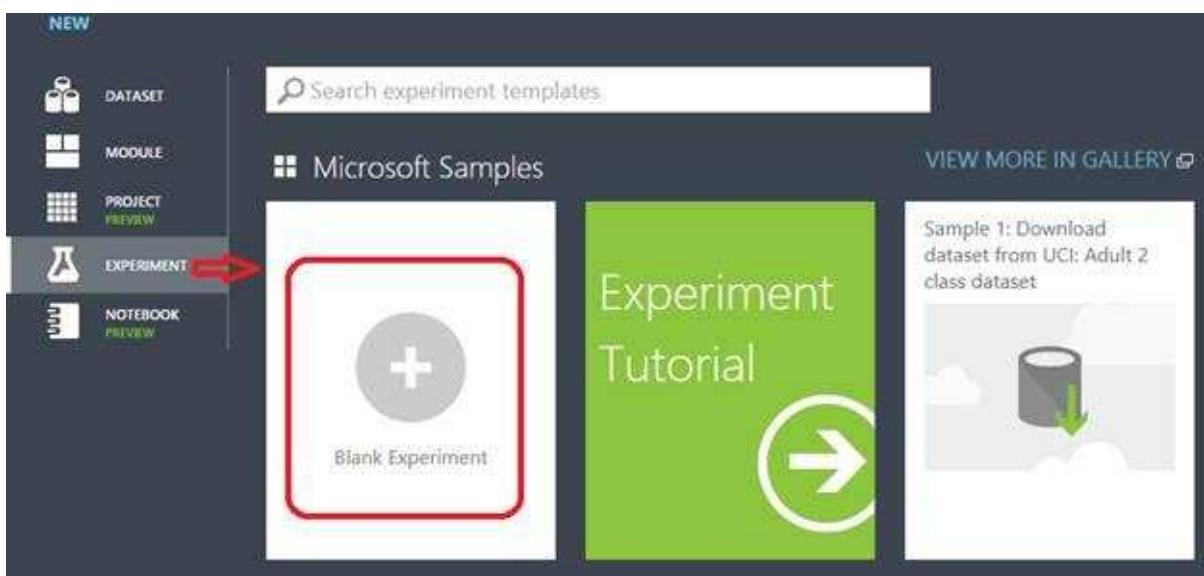


The screenshot shows the Microsoft Azure Machine Learning studio interface. On the left, there's a sidebar with icons for Projects, Experiments, Web Services, Notebooks, Datasets (which is selected), Trained Models, and Settings. The main area is titled "datasets" and shows a table of datasets. The table has columns: NAME, SUBMITTED BY, DESCRIPTION, DATA TYPE, CREATED, SIZE, and PROJECT. One row is highlighted with a red box: "My Training DataSet" by amitmishra.00, Data Type: GenericCSV, Created: 4/11/2016 11:31:15 AM, Size: 59.76 KB, Project: None. At the bottom of the screen, a message says "Upload of the dataset 'My Training DataSet' has completed." with a red arrow pointing to it. There are buttons for DOWNLOAD, DELETE, OPEN IN NOTEBOOK, GENERATE DATA ACCESS CODE, and ADD TO PROJECT. An "OK" button with a checkmark is also visible.

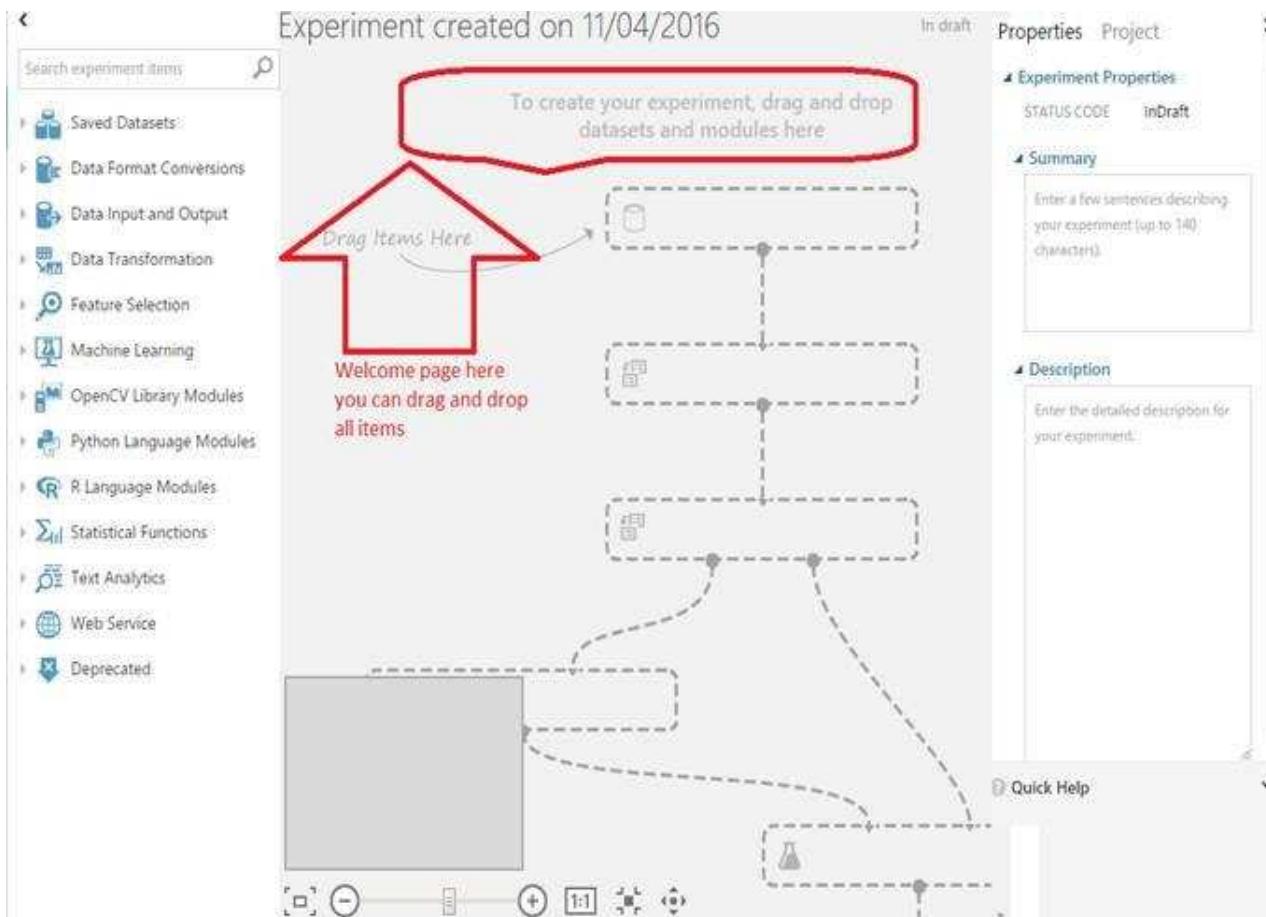
Click on “*EXPERIMENTS*” and create a new experiment.



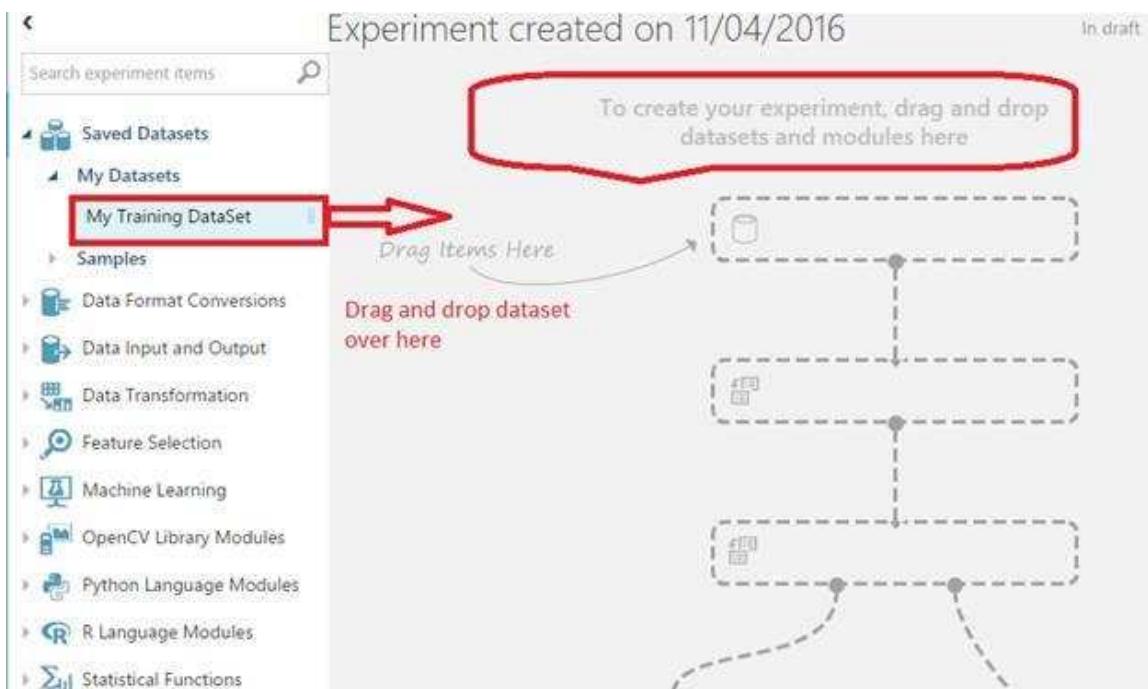
From the samples click on to “*Blank Experiment*”.



It will show a welcome page of the experiment like the following screenshot. Here, you can drag and drop items from the left side of the menu.

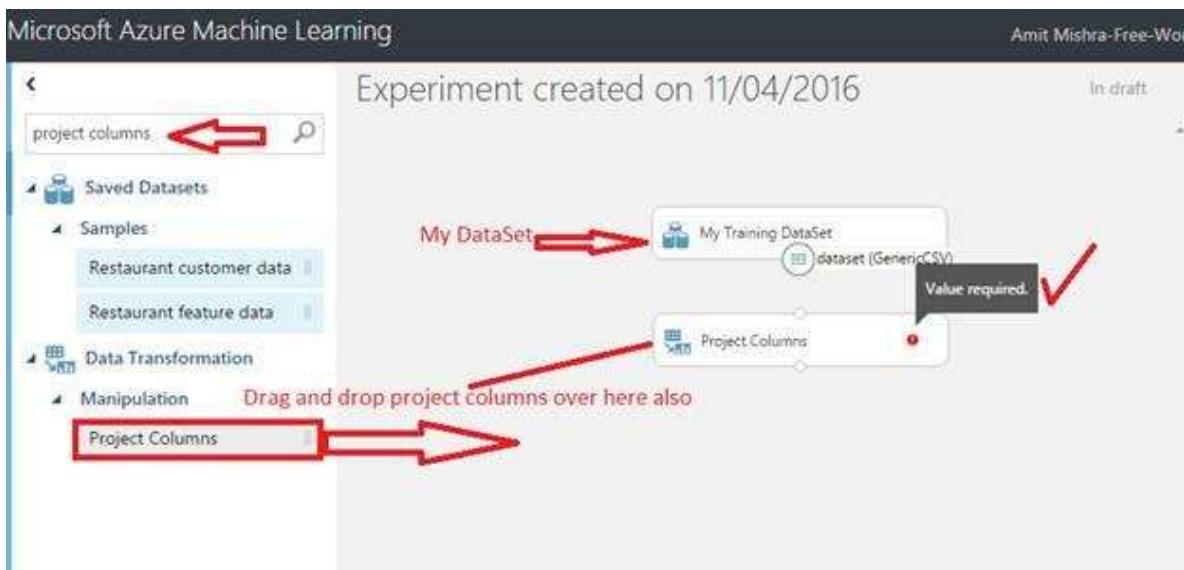


Under the saved datasets you can see your created dataset “*My training DataSet*”, drag and drop this dataset in to the right side of the page area.

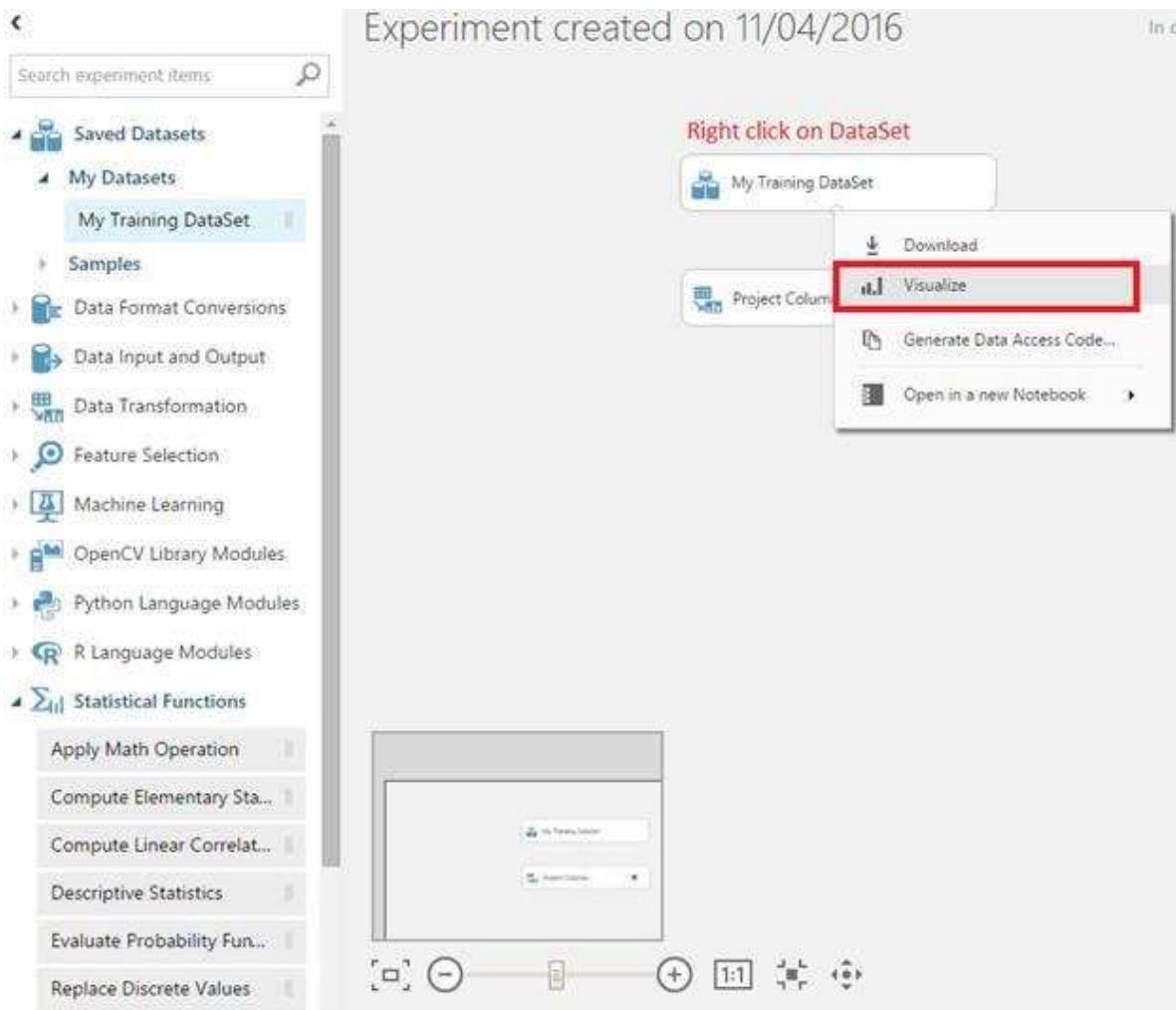


Under the Data Transformation, expand the *Manipulation* and click on to “*Project Columns*”. You can search it directly from Search box.

Drag and drop select columns in to the working area. You can see the “*Values required*” as red mark.



You can check all the information of your dataset, right click on to it and select “Visualize”.



There are 891 rows and 12 columns in “My training dataset”, which I have created in a titanic way.

Experiment created on 11/04/2016 > My Training DataSet > dataset

rows 891 columns 12

PassengerId Survived Pclass Name Sex Age

view as 

1	0	3	Braund, Mr. Owen Harris	male	22
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38
3	1	3	Heikkinen, Miss. Laina	female	26
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35
5	0	3	Allen, Mr. William Henry	male	35
6	0	3	Moran, Mr. James	male	
7	0	1	McCarthy, Mr. Timothy J	male	54
8	0	3	Palsson, Master. Gosta Leonard	male	2
			Johnson, Mrs. Oscar W	female	77

To create a graph, select a column in the table

Connect your dataset to project columns as a relationship,

Microsoft Azure Machine Learning

Experiment created on 11/04/2016

In draft

Search experiment items

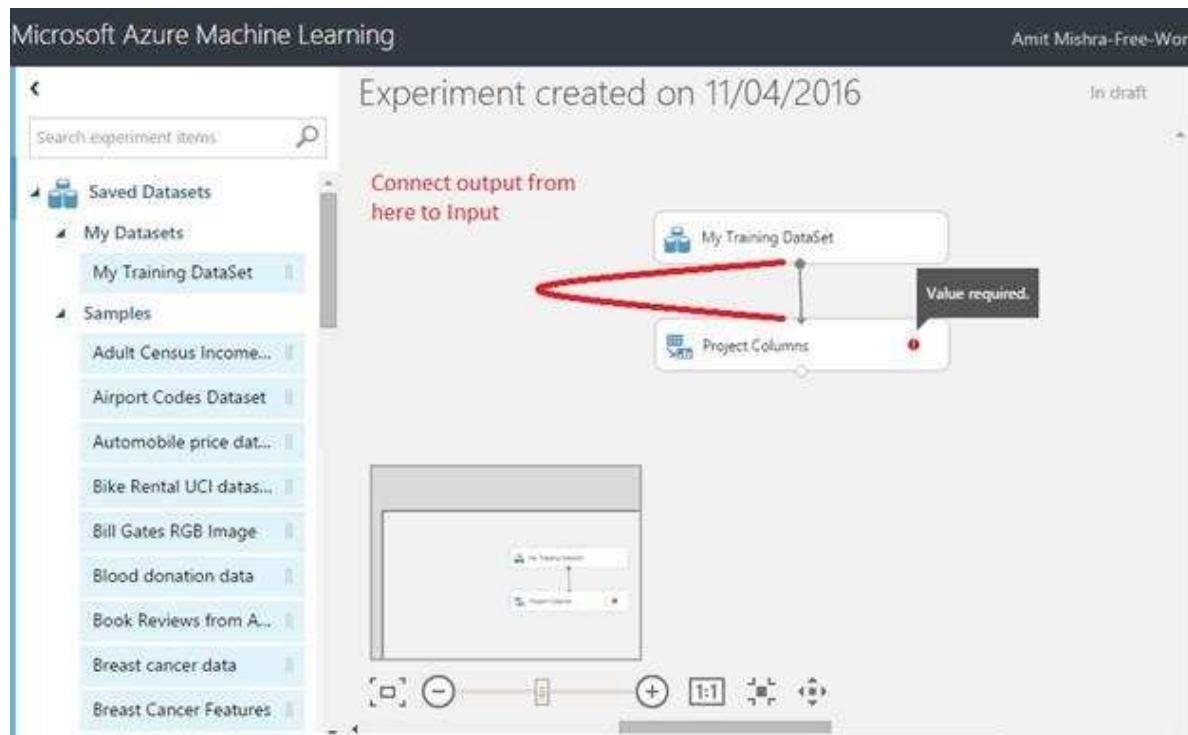
Saved Datasets

- My Datasets
 - My Training DataSet
- Samples
 - Adult Census Income...
 - Airport Codes Dataset
 - Automobile price dat...
 - Bike Rental UCI datas...
 - Bill Gates RGB Image
 - Blood donation data
 - Book Reviews from A...
 - Breast cancer data
 - Breast Cancer Features

Connect output from here to Input:

My Training DataSet → Project Columns

Value required.



Click on to project columns to launch the column selector where you can select the desired values.

Experiments - Microsoft Azure Machine Learning

https://studio.azureml.net/Home/ViewWorkspaceCached/6f423f699ec74bc9ba3924460a36ee81#Works&experiments

Experiment created on 11/04/2016

Amit Mishra-Free-Worksp...

Properties Project

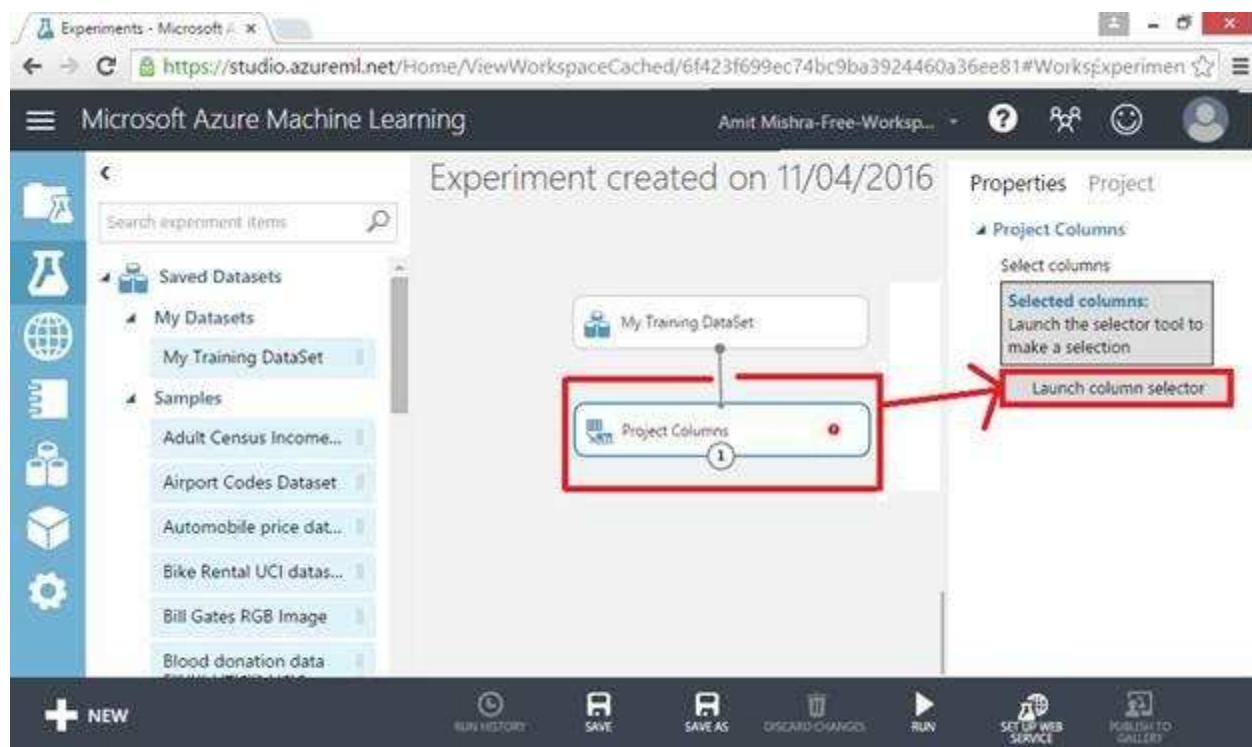
Project Columns

Select columns

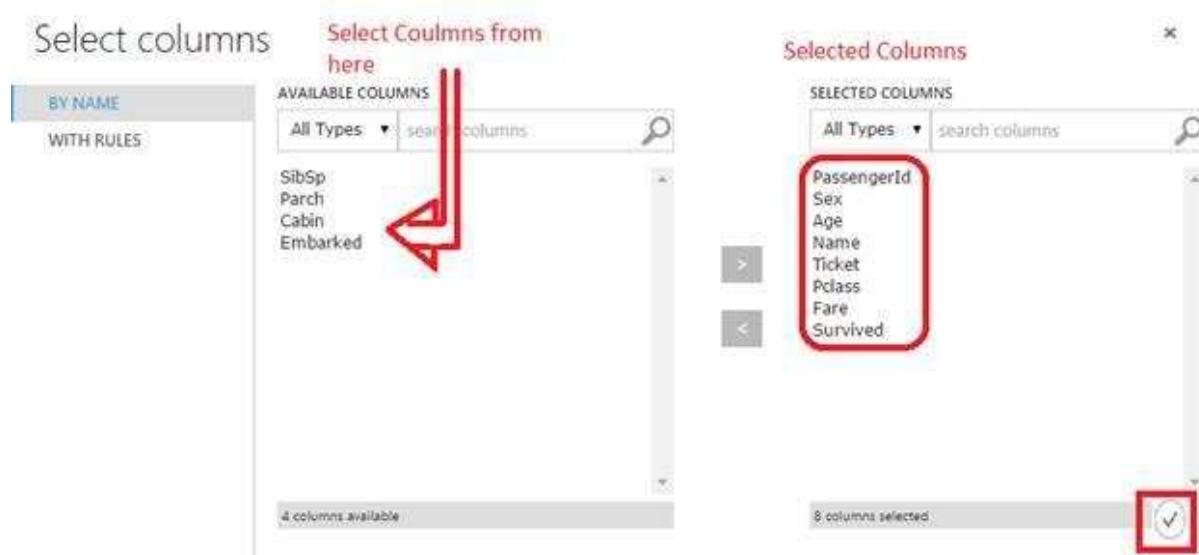
Selected column:
Launch the selector tool to make a selection

Launch column selector

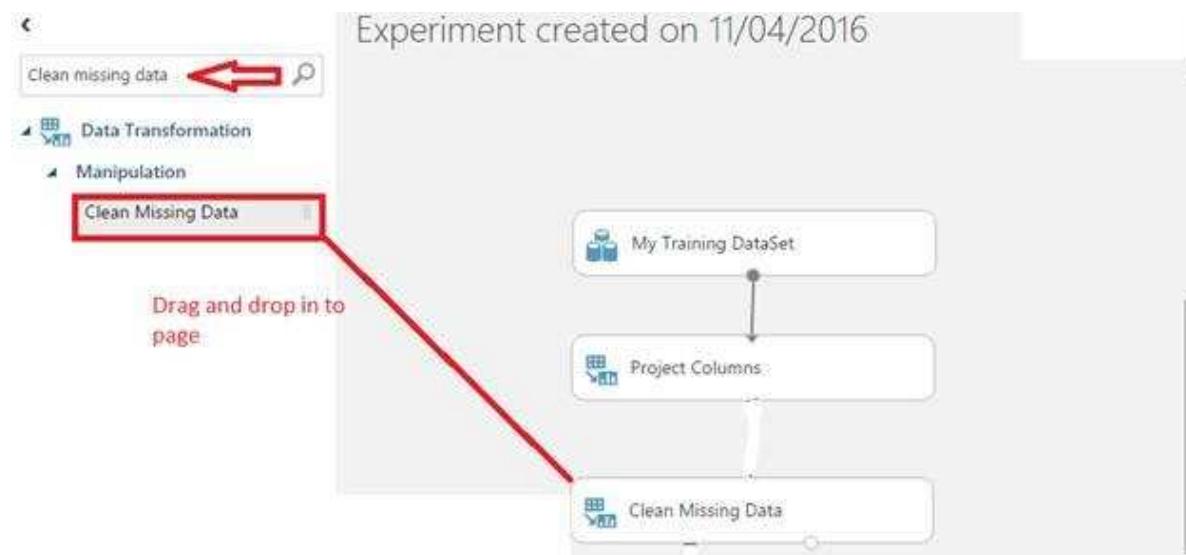
My Training DataSet → Project Columns



After clicking on launch column selector, the following pop up will open. Select columns from the left side of the page which you want to see in project columns, after selecting the desired columns click on the check button.

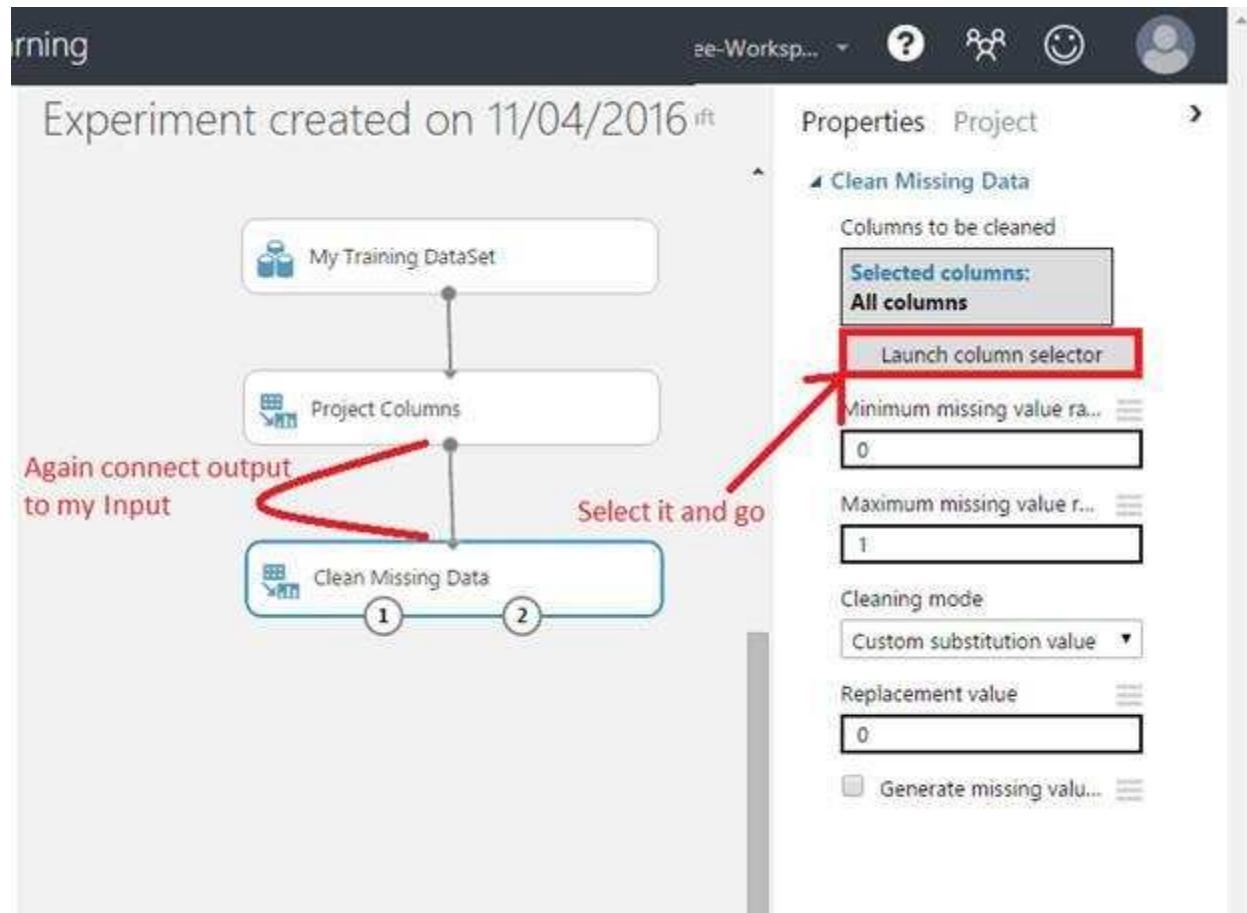


In some cases we are missing some data like age of some people, so in this situation we can use the clean missing data facility. Search the clean missing data facility from search box, drag and drop in the page.



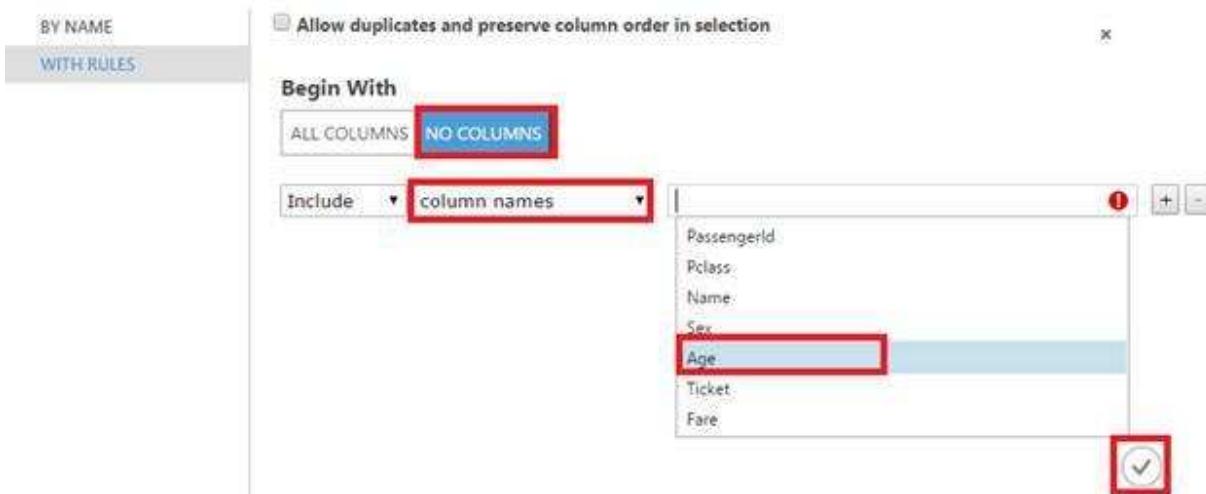
Again connect your output to the input.

Then I need columns to be cleaned, select clean missing data and in the right side of the page click on "*Launch column selector*".



We want to select age from all the columns, click column names and in dropdown menu select age. Click on to Next button to complete the action.

Select columns



Allow duplicates and preserve column order in selection

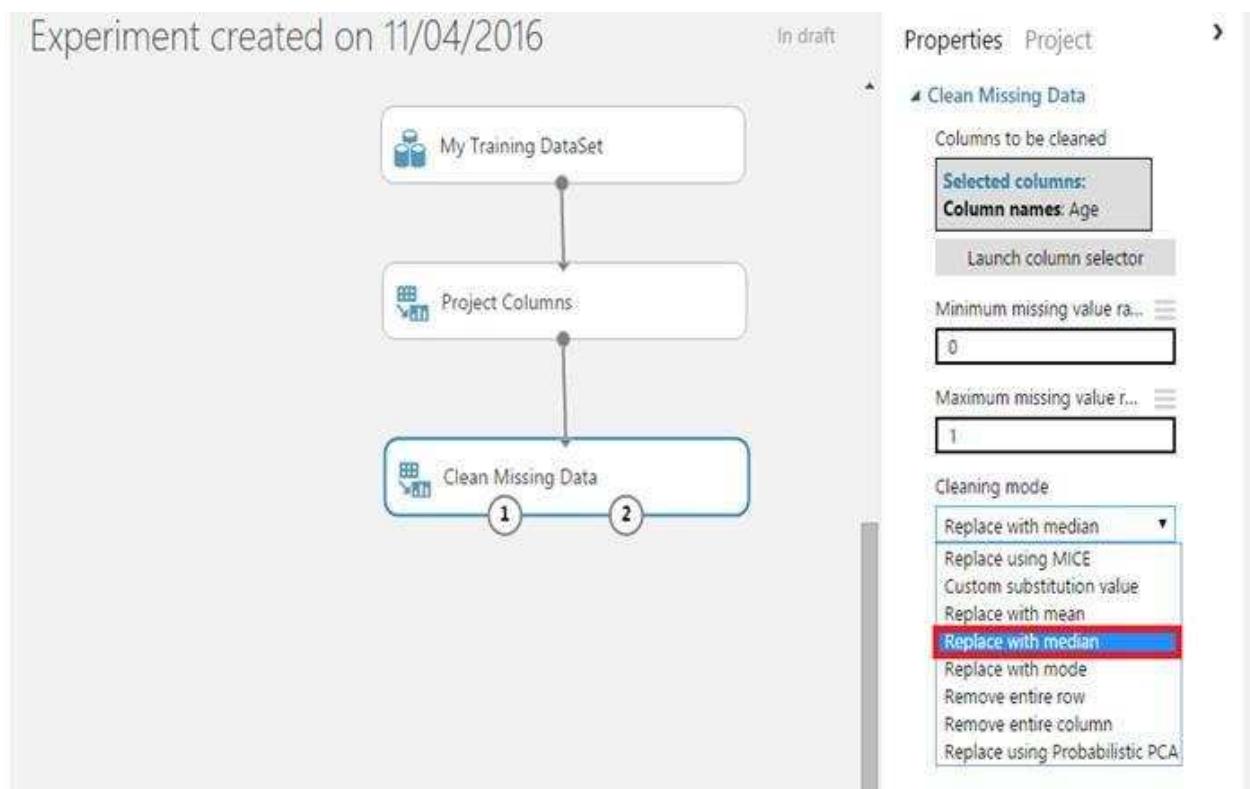
Begin With

ALL COLUMNS NO COLUMNS

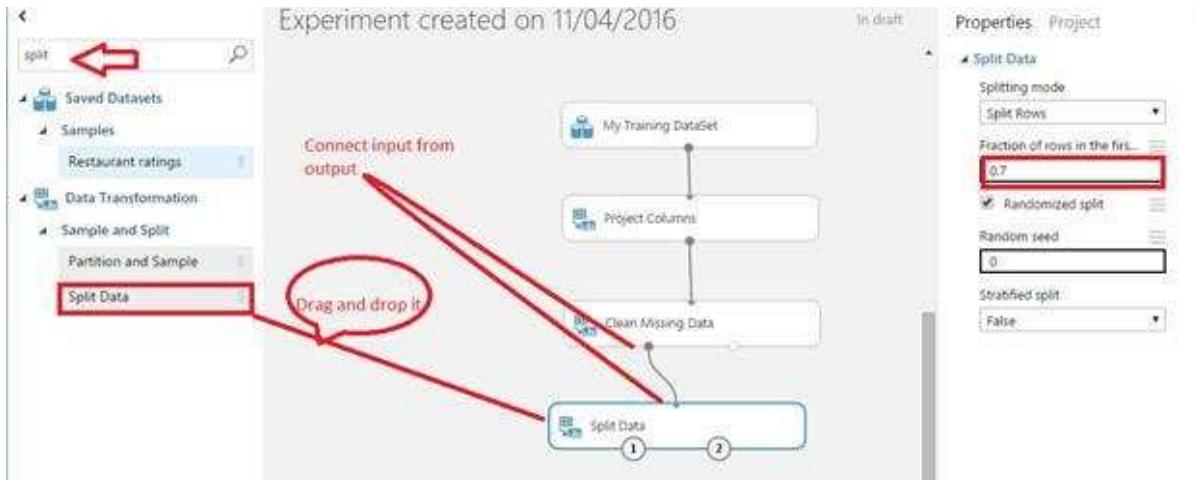
Include **column names**

PassengerId	
Pclass	
Name	
Sex	
Age	
Ticket	
Fare	

The cleaning mode will replace with median as in the following figure,

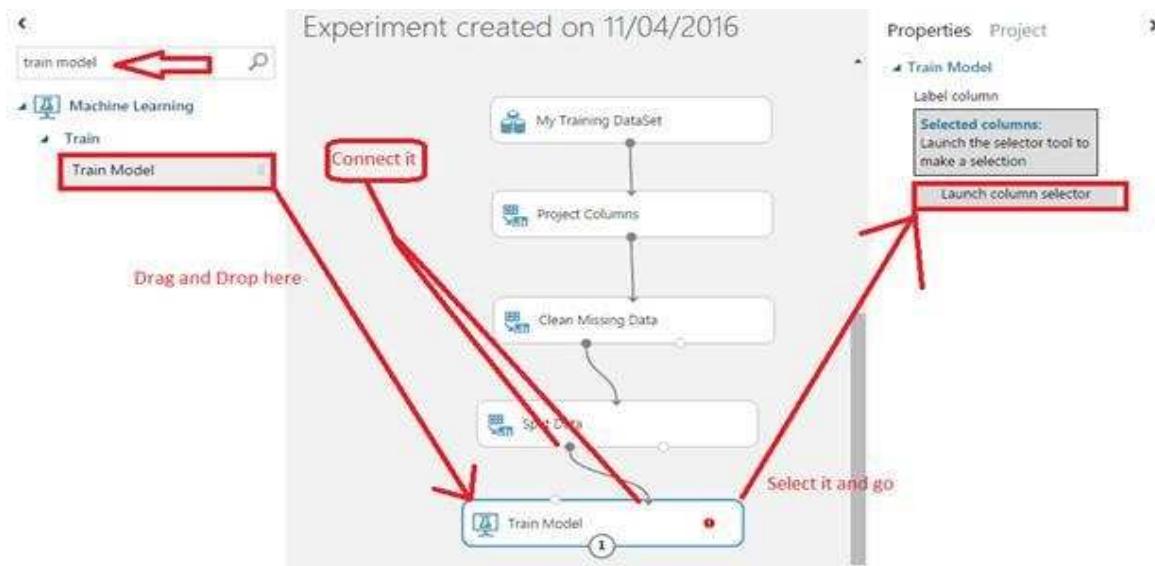


The next thing to do, we want to split data. Just go into search box and search it. Then, connect the clean missing data box to the Split Data box. Click on the Split Data box and you will see that you can set a few parameters for it on the right of the screen. Enter “0.7” for “fraction of rows”.



Now, Find and drop “*Train Model*” from the menu. Connect the first data set from the Split Data to the second dataset of Train Model.

In train model there is a red mark, because of train model does not know how many columns we are going to use, so we need to tell it that. Click on Train Model and then “*Launch column selector*” from the right pane.

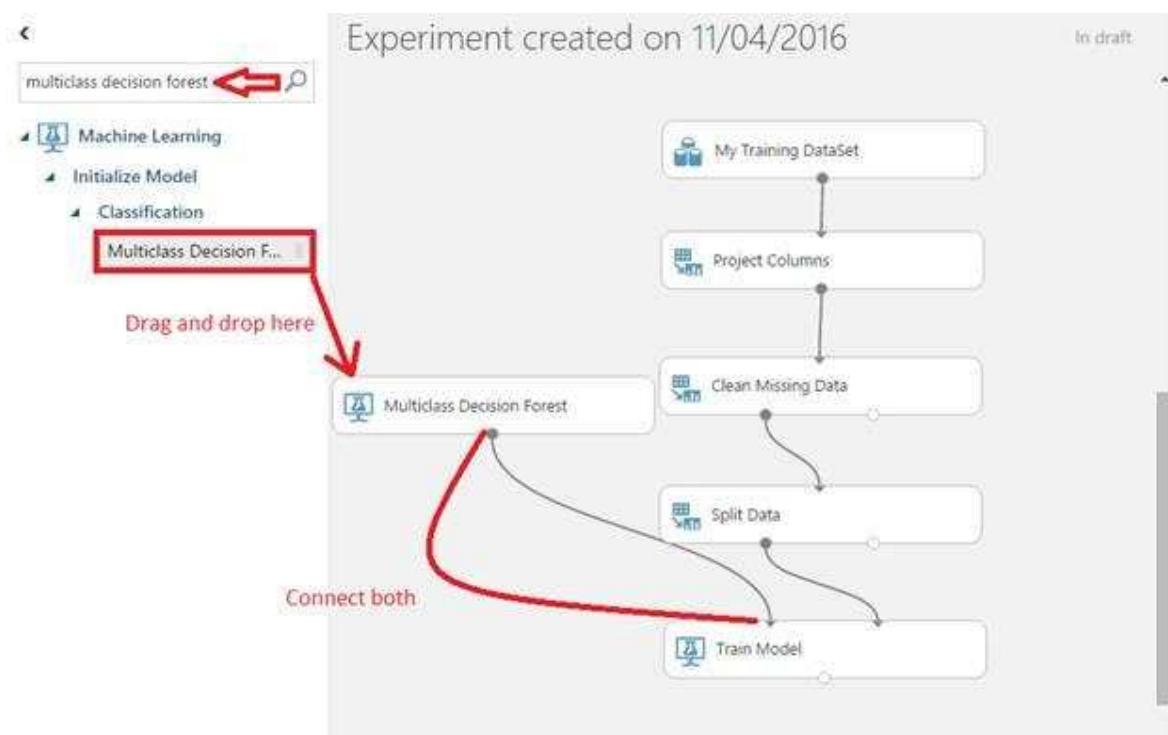


Start typing “*Survived*” into the column name box. A suggestion box will open. Select the first item named “*Survived*” in the list and click on check mark.

Select a single column

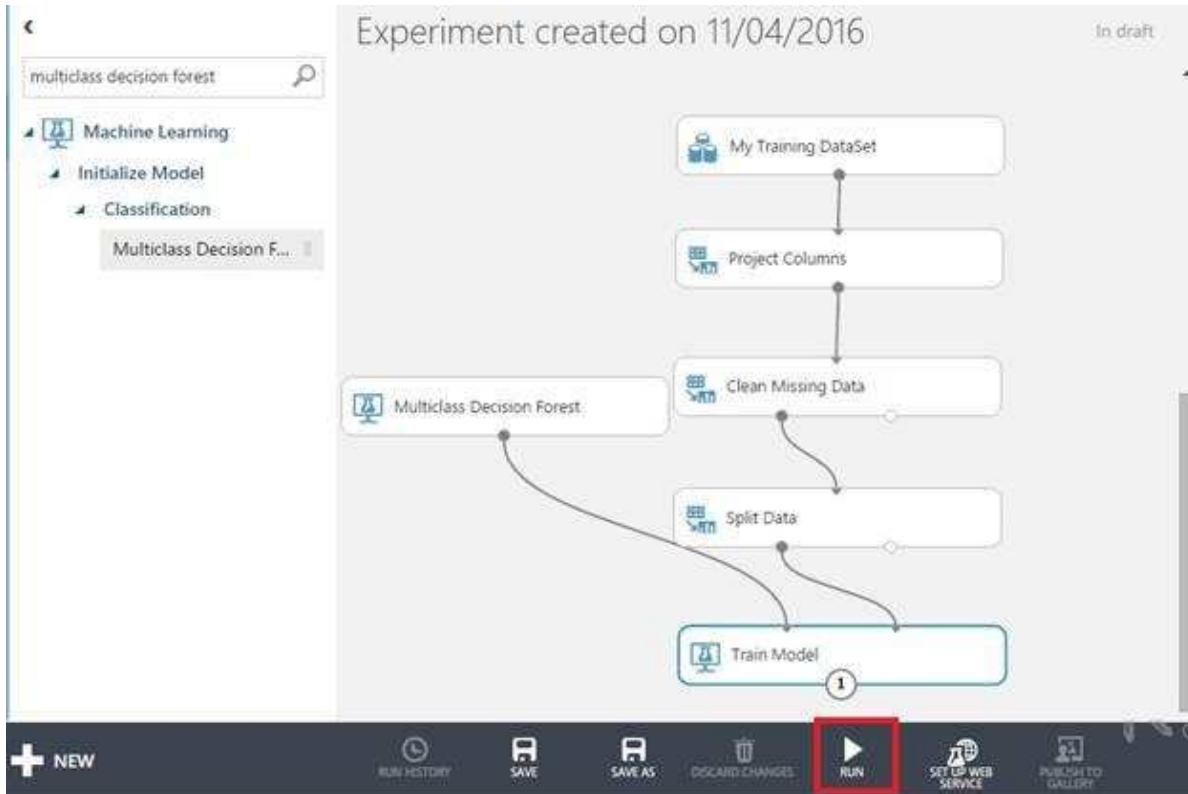


Now, I’m going to drop *multiclass decision forest* in to the working page, connect it with train model.



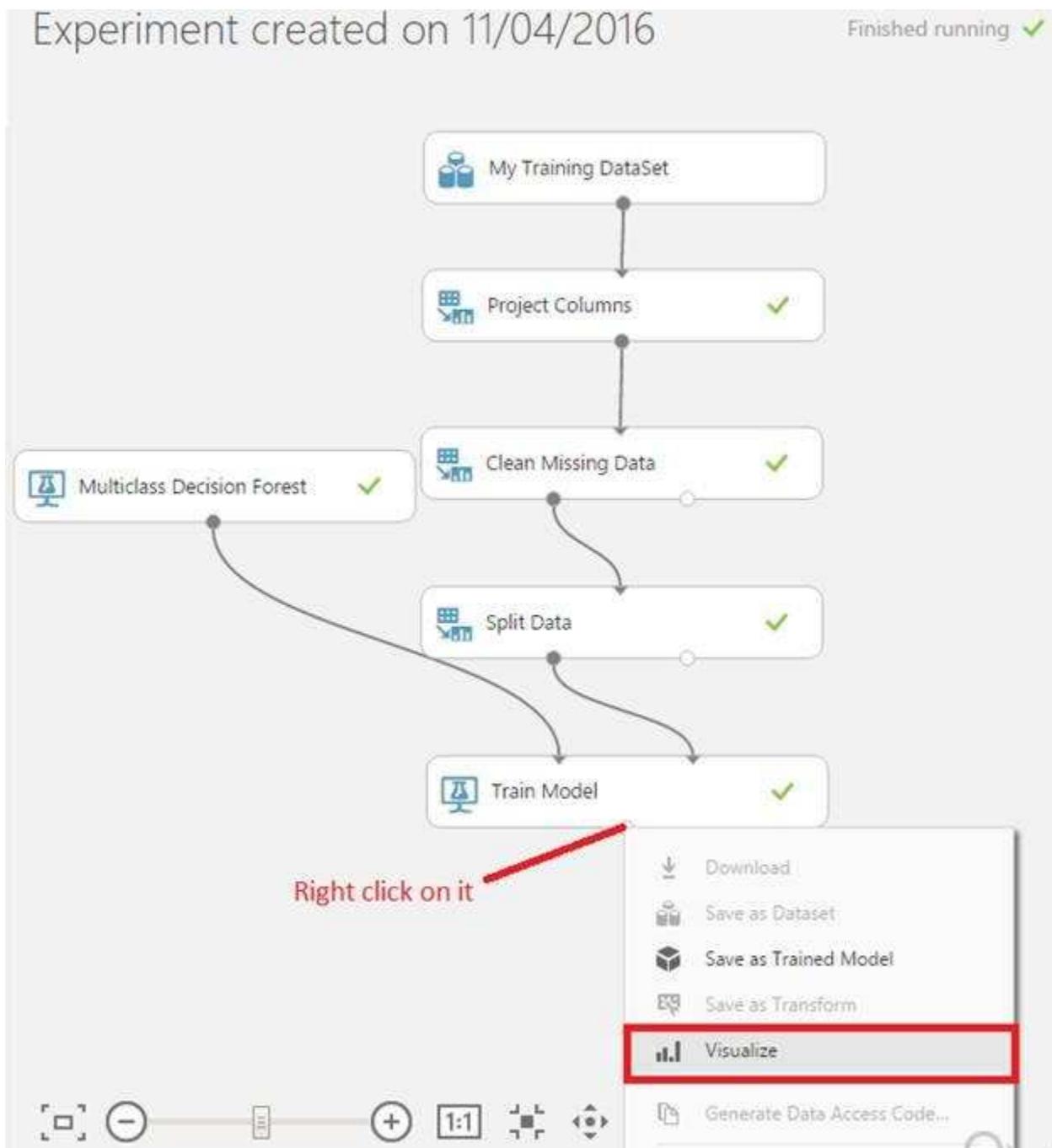
We have taken the necessary steps to train a model from our training data. Before going on with performance evaluation, let’s run our experiment to verify if it works or not.

Click on “Run” button at bottom of the page.



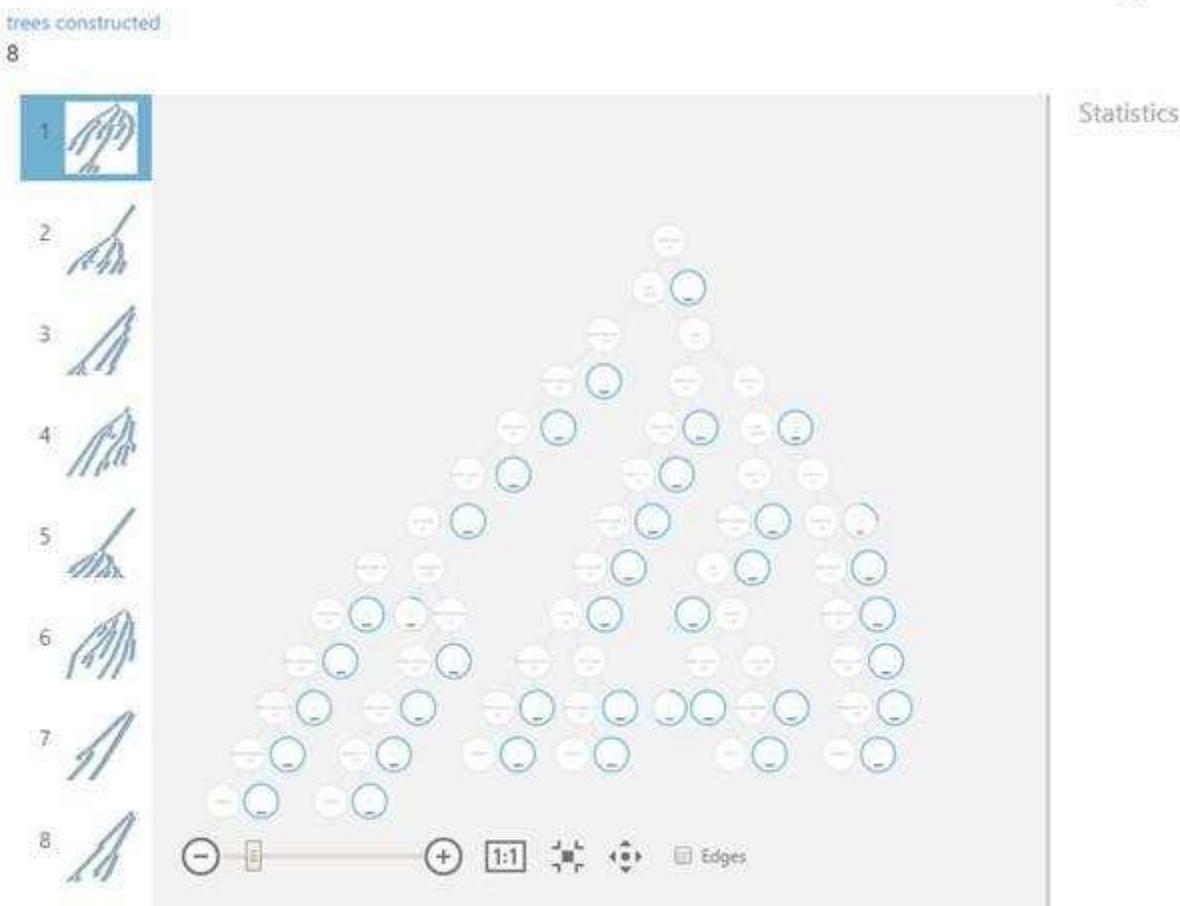
You will see that, one by one, green check boxes appear on the boxes on the working area. This means a calculation is performed inside the box and its results are ready for the next box to use.

For producing a model structure, right click on to Train Model and select “Visualize”.



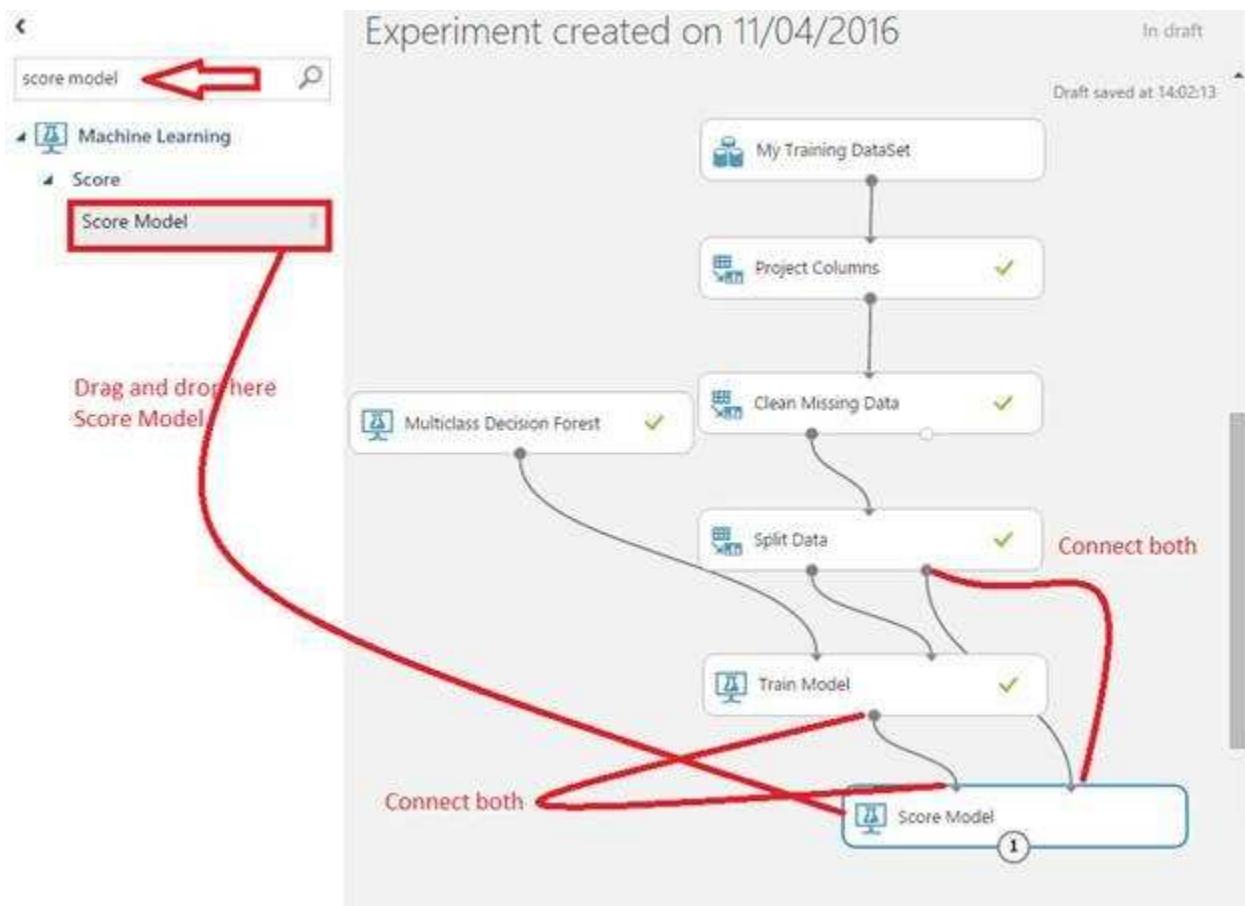
Now, the “Train Model” box contains a trained model.

Experiment created on 11/04/2016 ➤ Train Model ➤ Trained model X

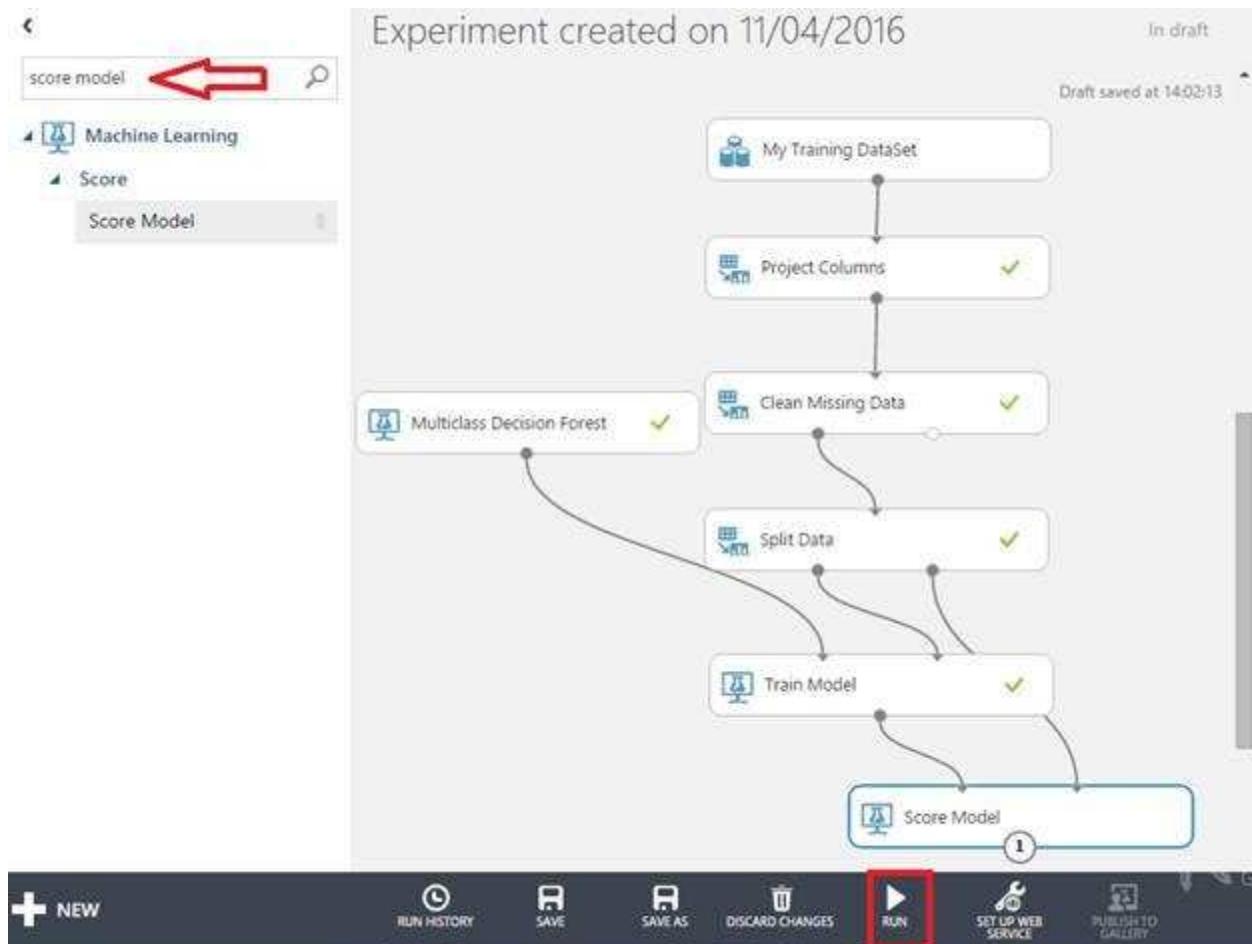


Now we need to drop a score model in the page to test our model to see how it performs. So go in to search box and type score model and drop it in to the working page.

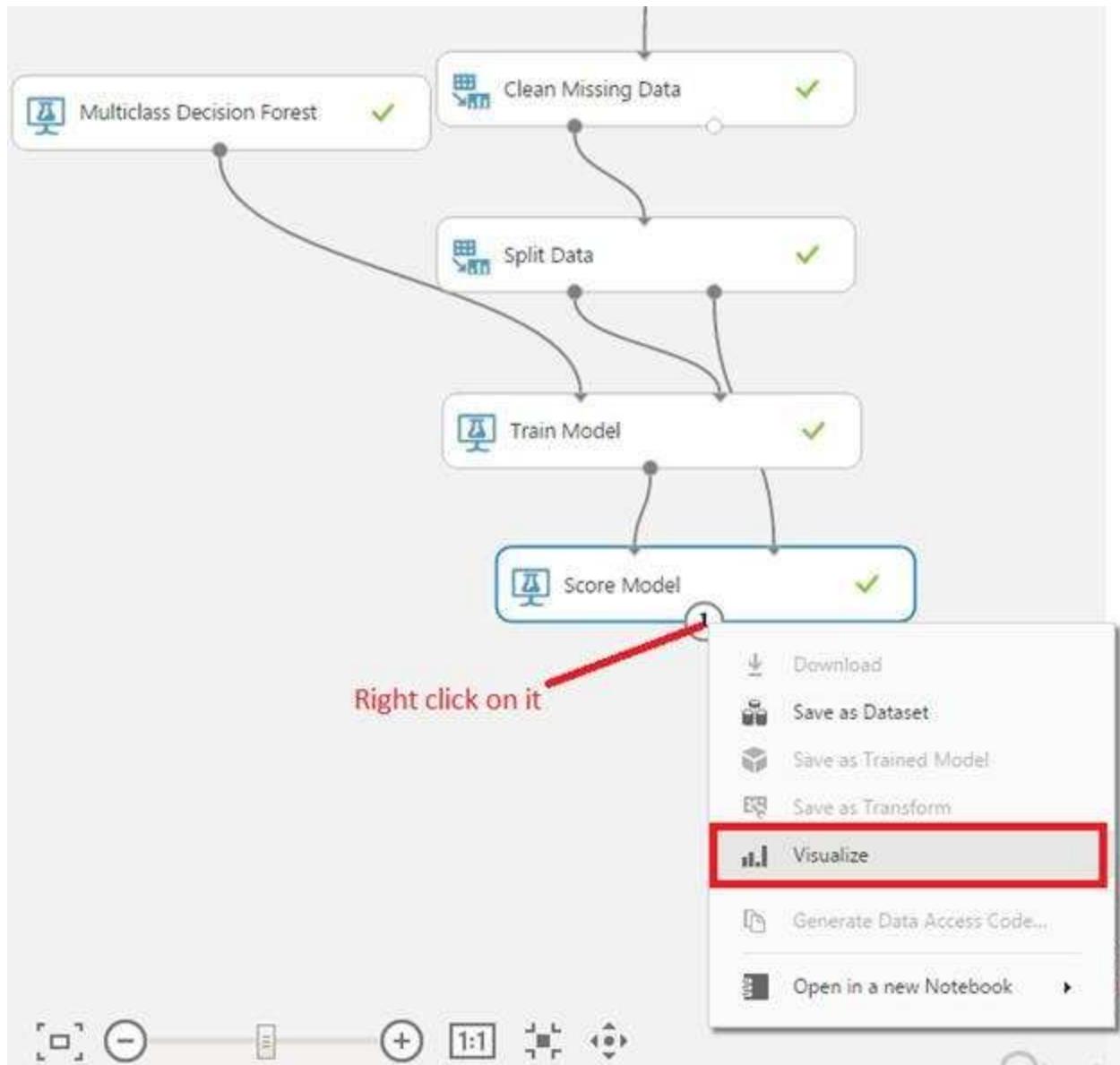
Connect Train Model with first input of Score Model and the second output from Split Data to Score Model.



Now run your score model.



Right click on Score Model and visualize it.



The result is shown in the following screenshot, where lowest fare is of Pclass.

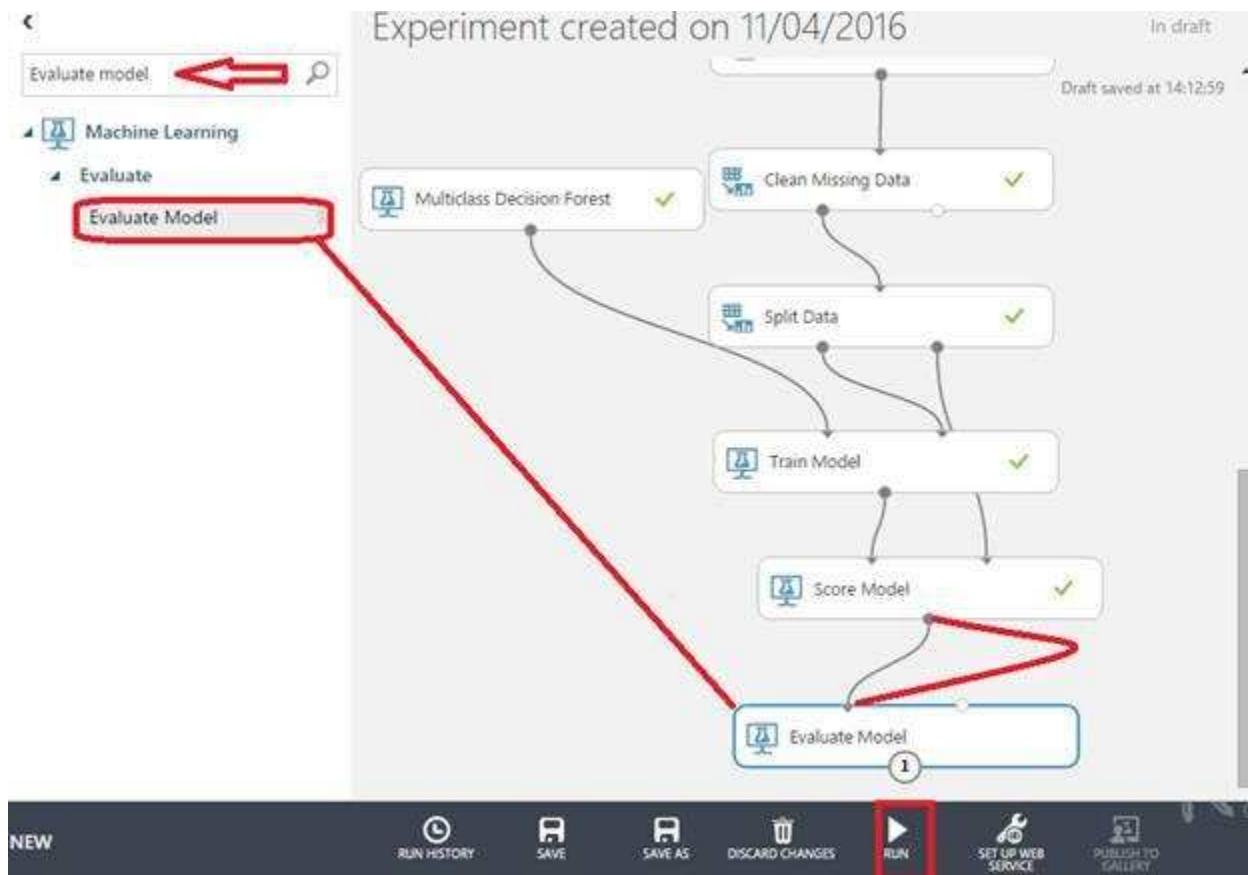
Experiment created on 11/04/2016 › Score Model › Scored dataset

rows 267 columns 11

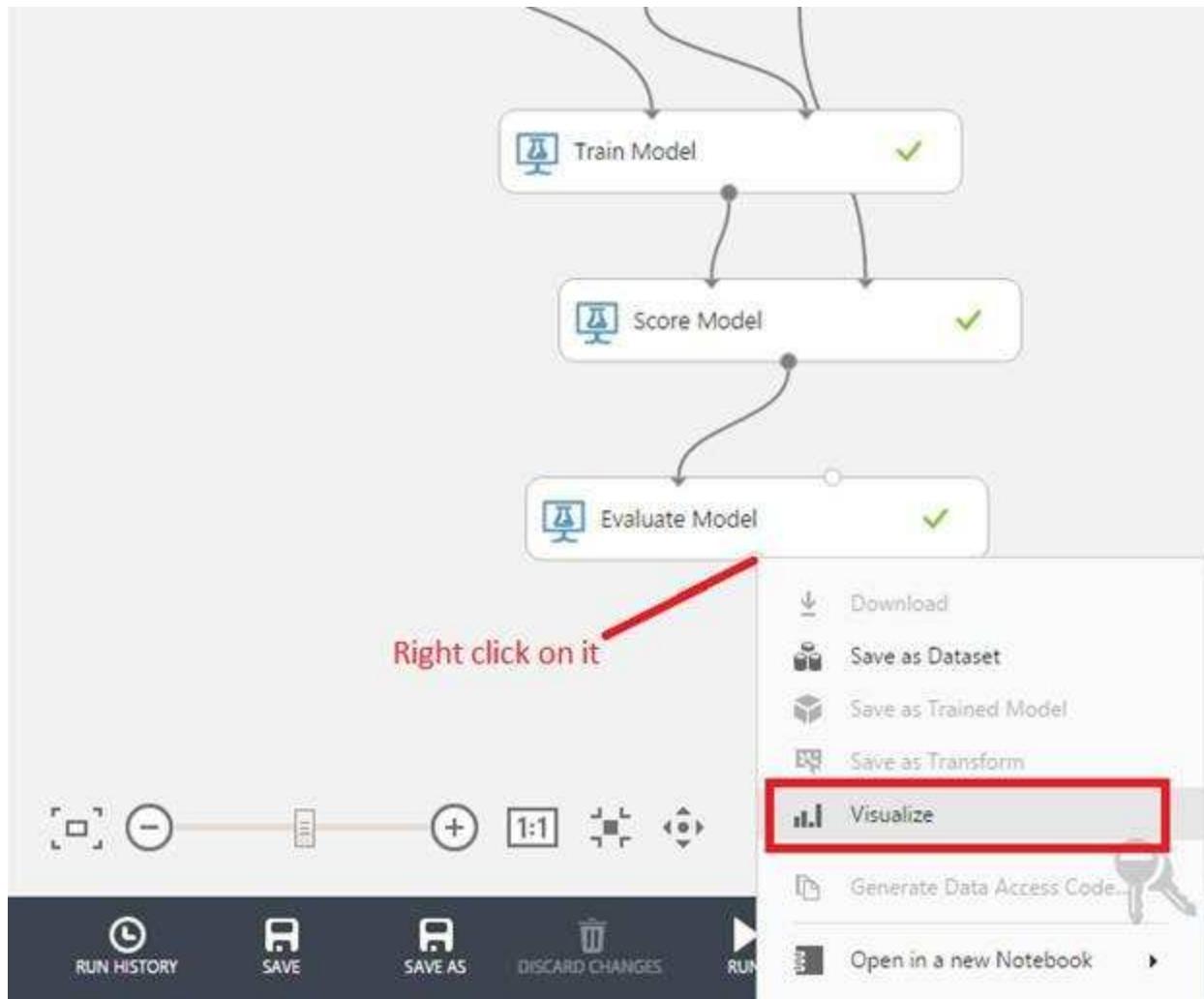
	PassengerId	Survived	Pclass	Name	Sex	Age
	view as					
720	0	3	Johnson, Mr. Malkcolm Joackim	male	33	
717	1	1	Endres, Miss. Caroline Louise	female	38	
450	1	1	Peuchen, Major, Arthur Godfrey	male	52	
580	1	3	Jussila, Mr. Eirik	male	32	
153	0	3	Meo, Mr. Alfonzo	male	55.5	
523	0	3	Lahoud, Mr. Sarkis	male	28	
206	0	3	Strom, Miss. Telma Matilda	female	2	
... 257 more rows						

To create a graph, select a column in the table

Now add an Evaluate Model, and connect it from Score Model, Click on to Run button.



Right click on Evaluate Model and click on “Visualize”.



After this, in the following screenshot there are many types of accuracy completed tasks.

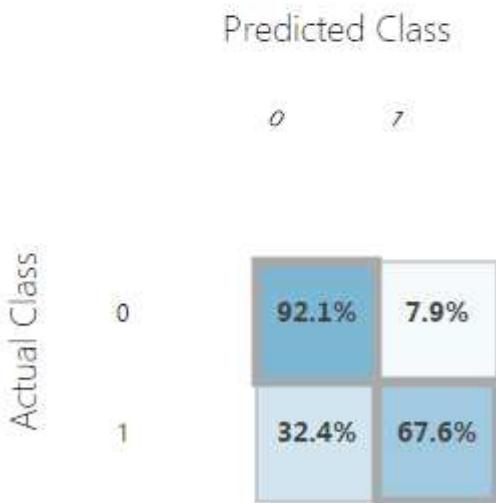
Under Metrics, you can see how well the algorithm performed by checking out “Overall Accuracy”. The value here shows what percentage of the digits in the cross validation set we have correctly predicted. In our case, it is somewhere around 67 percent. This means that we have learned a model that can be used to recognize handwritten digits with 67 percent accuracy.

Experiment created on 11/04/2016 ➤ Evaluate Model ➤ Evaluation results ✕

◀ Metrics

Overall accuracy	0.827715
Average accuracy	0.827715
Micro-averaged precision	0.827715
Macro-averaged precision	0.831543
Micro-averaged recall	0.827715
Macro-averaged recall	0.798841

◀ Confusion Matrix



I have tried to show how you can easily use Azure Machine Learning Studio for basic easy tasks.

I hope you enjoyed this chapter.

Working with Azure RemoteApp

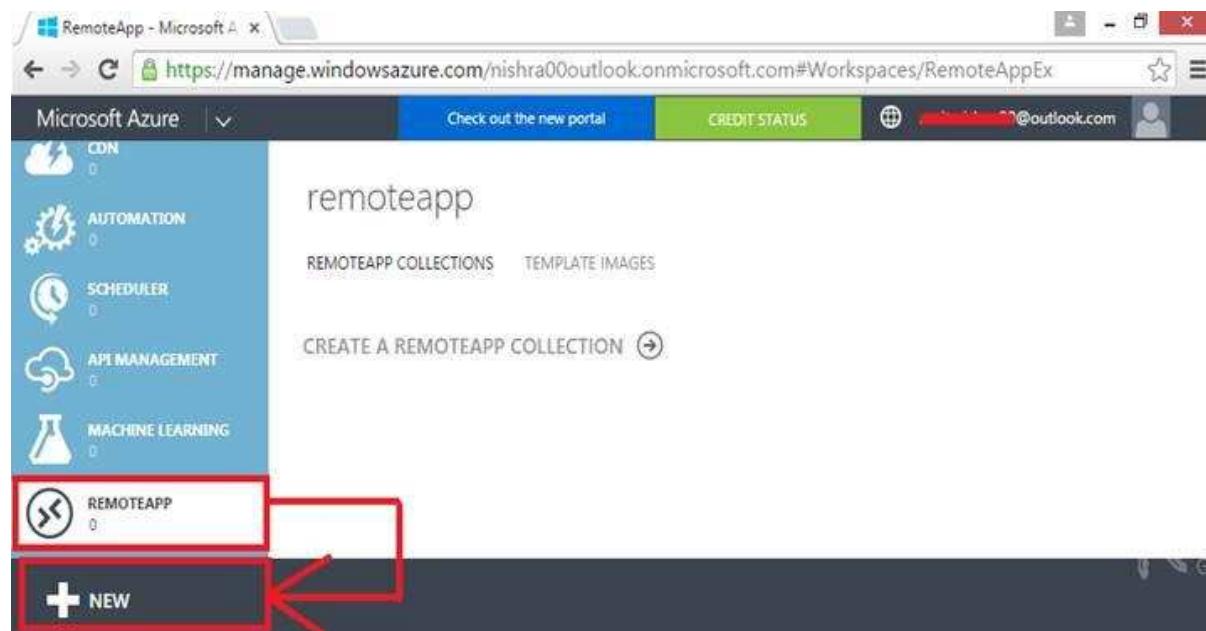
Azure RemoteApp provides you a cost effective platform to deliver your Windows Apps from the cloud. Azure RemoteApp combines the Window application experience and powerful RDS capabilities on Azure's reliable platform and helps IT sectors to bring, scale, agility to corporate applications.

Points of RemoteApp

- Remote applications delivered from the reliable Azure platform.
- Access from IOS, Mac OS X, and Android devices.
- Delivered via Microsoft Remote Desktop Protocol and RemoteFX.
- Scale without large capital expense.
- Flexible hybrid or cloud deployment options.

To create a RemoteApp you must have to login in Azure.

Scroll down, and hit on “New” button to create a RemoteApp.



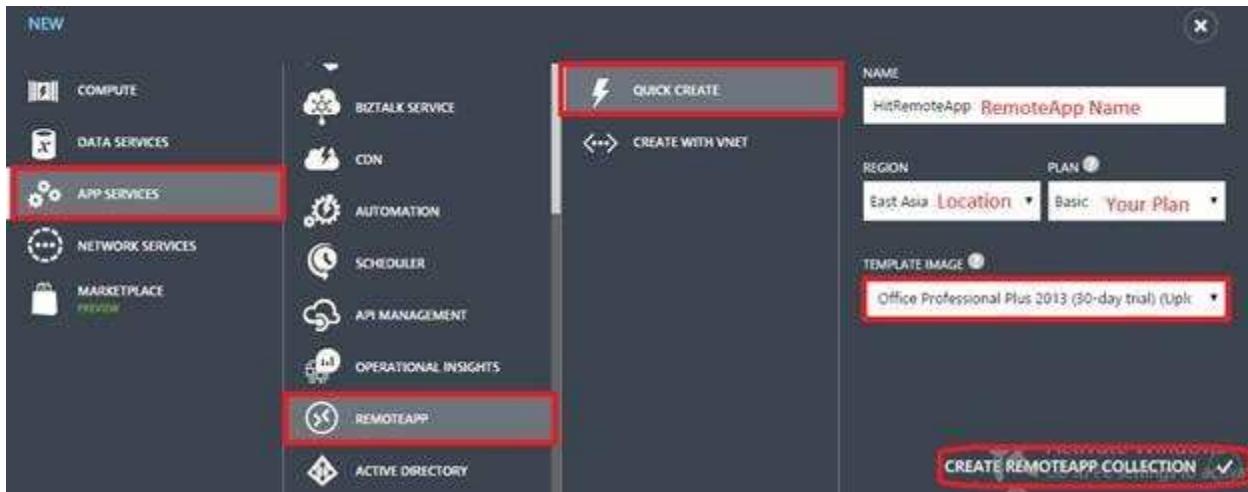
In App Services select RemoteApp, click on to quick create.

In the name section, give a name to your RemoteApp.

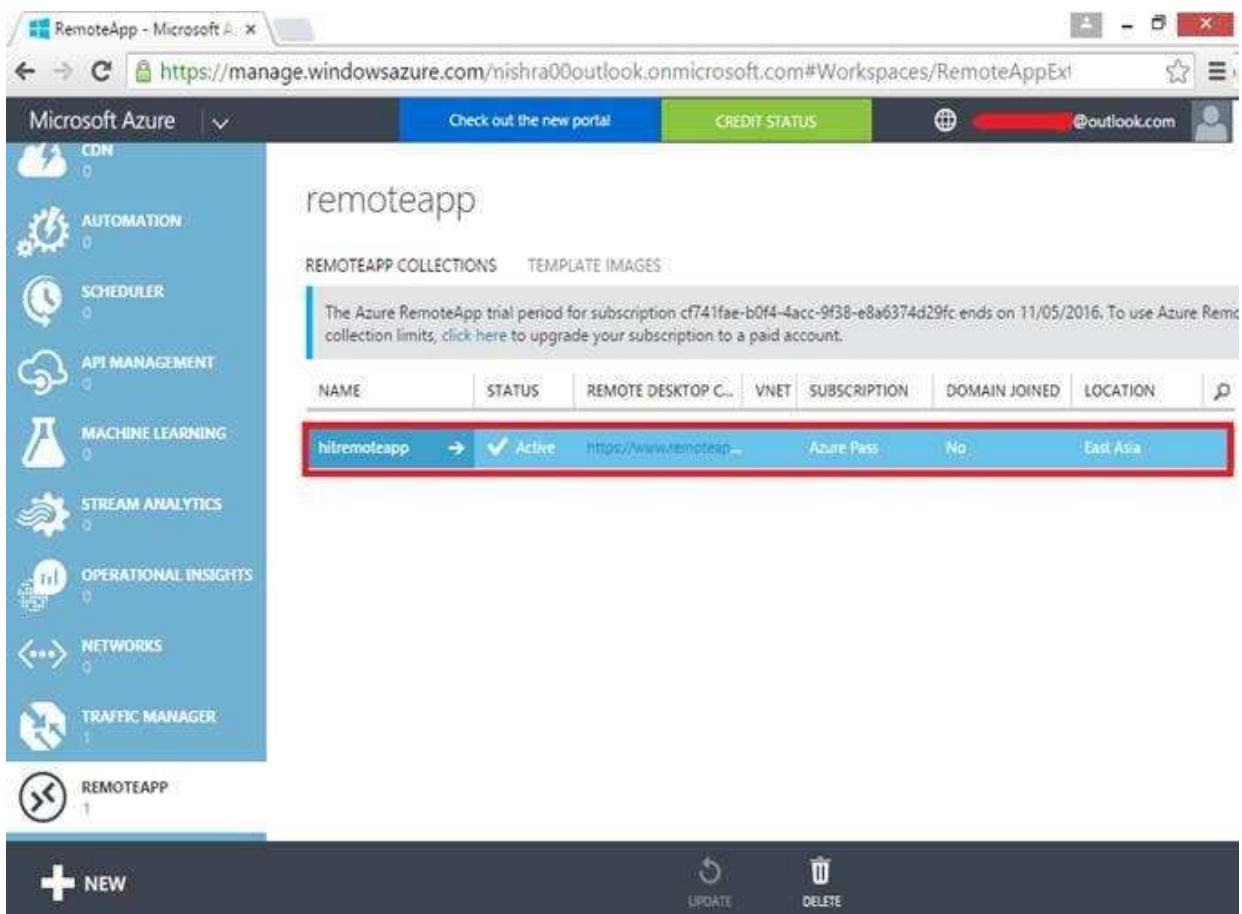
Select your location of App.

I'm selecting Basic plan.

Click on to “CreateRemoteAppCollection”.



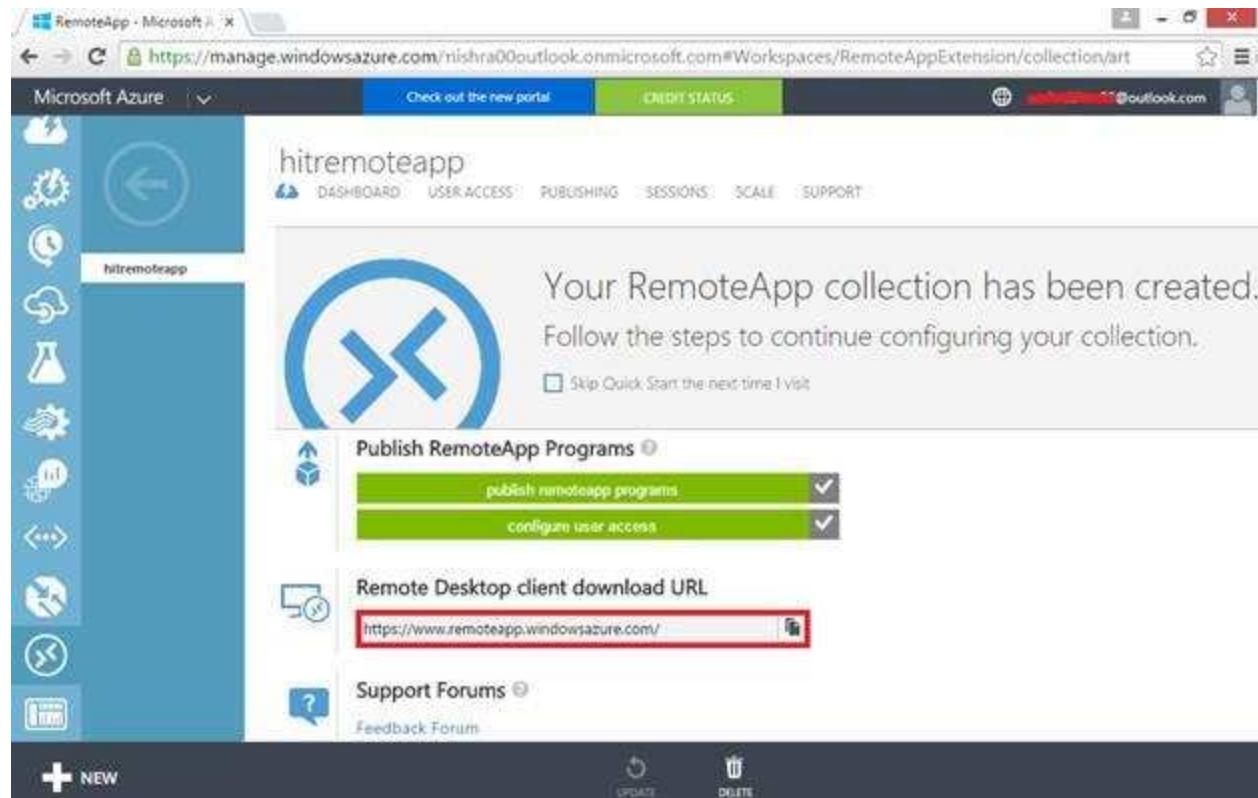
Here, my RemoteApp has been created (hitremoteapp) and that's status is active, click on to it for further.



NAME	STATUS	REMOTE DESKTOP C.	VNET	SUBSCRIPTION	DOMAIN JOINED	LOCATION
hitremoteapp	Active	https://www.remoteap...	Azure Pass	No	No	East Asia

Here I need to download and log in to Azure remote desktop client, before using it I need to enter my credentials. So the link of Azure remote desktop client can be finding as shown in following screenshot after clicking on to your existing RemoteApp.

Copy this URL from here, and paste it on a browser tab.



Your RemoteApp collection has been created. Follow the steps to continue configuring your collection.

Skip Quick Start the next time I visit

Publish RemoteApp Programs

- publish remoteapp programs
- configure user access

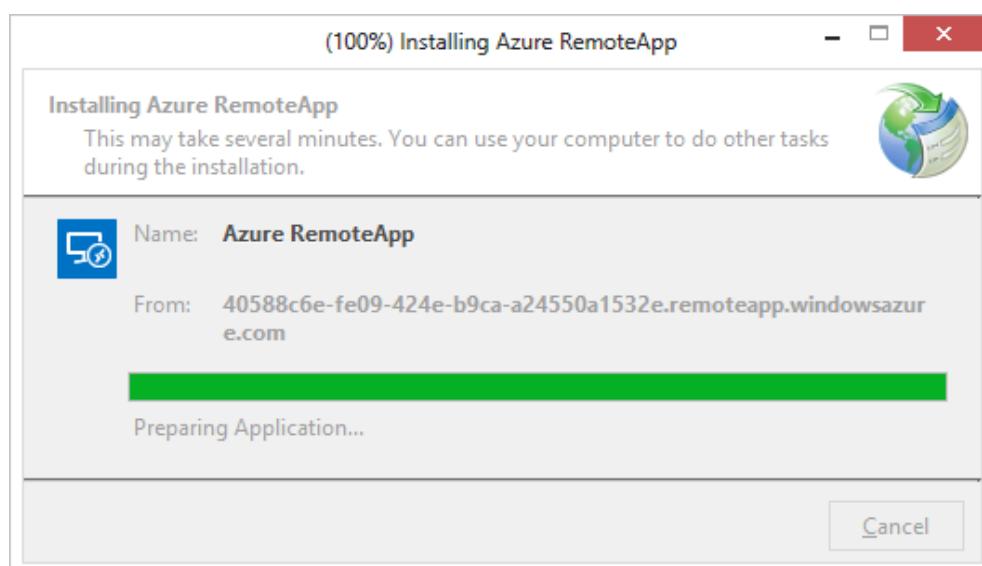
Remote Desktop client download URL

https://www.remoteapp.windowsazure.com/ 

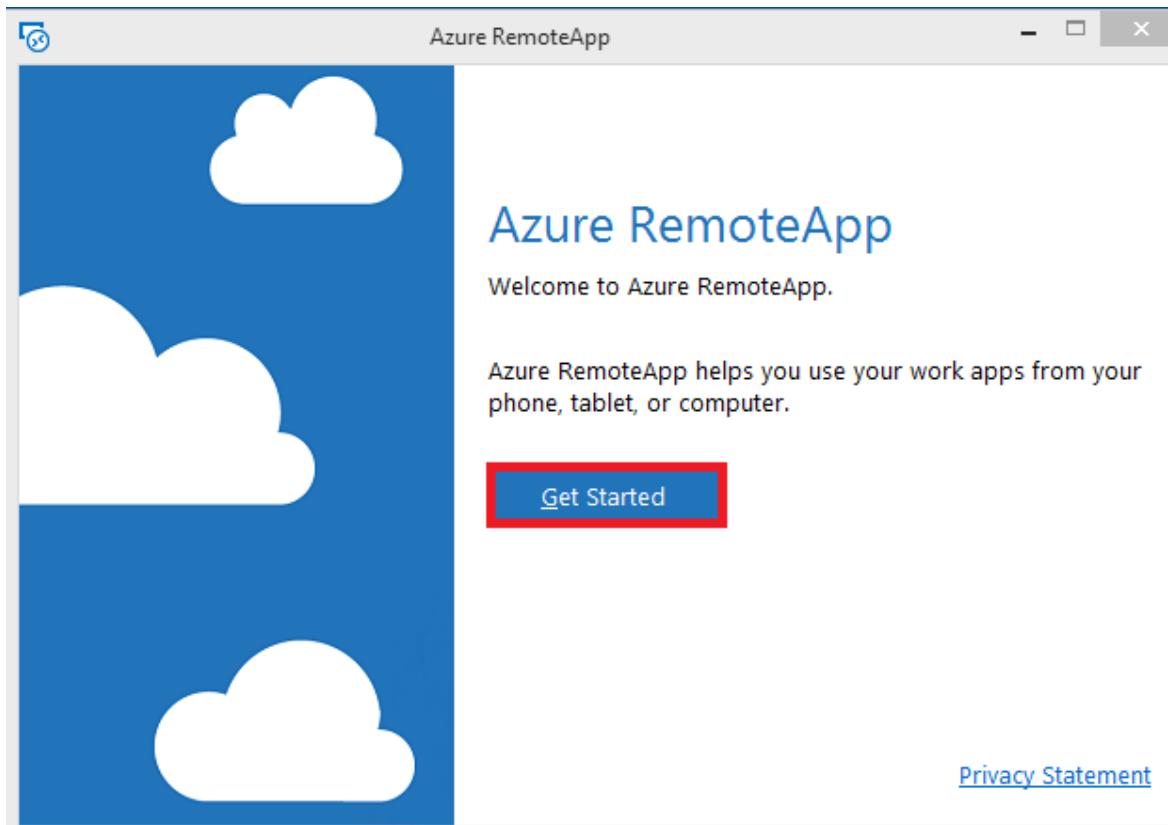
Support Forums

Feedback Forum

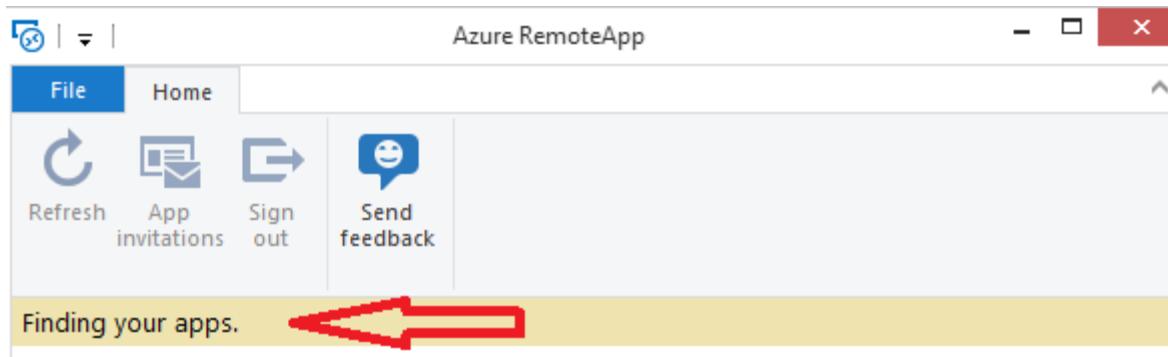
You will find there, client remote download (Azure RemoteApp), download it.



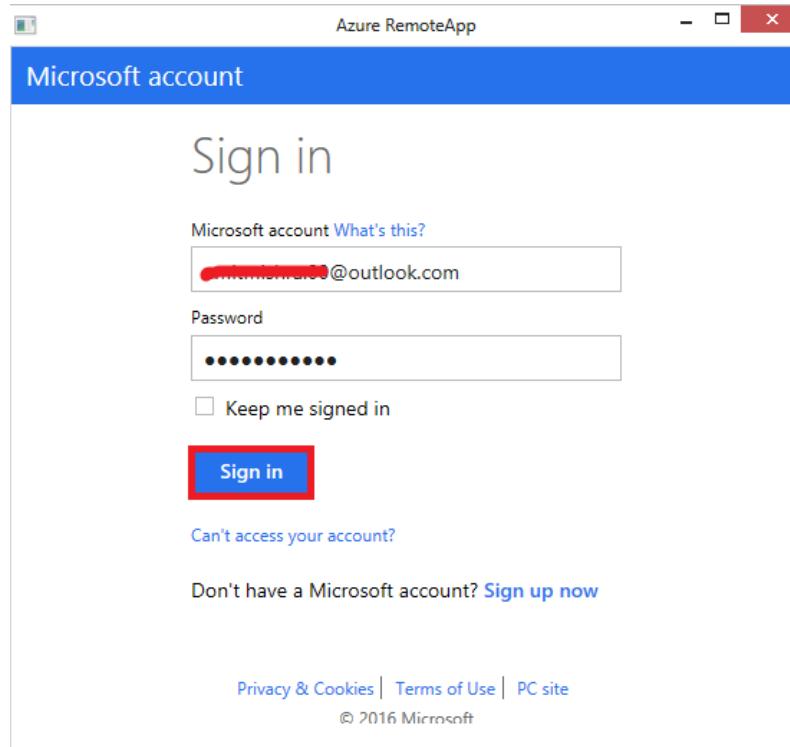
So the Azure Remote downloaded and click on to “Get Started”.



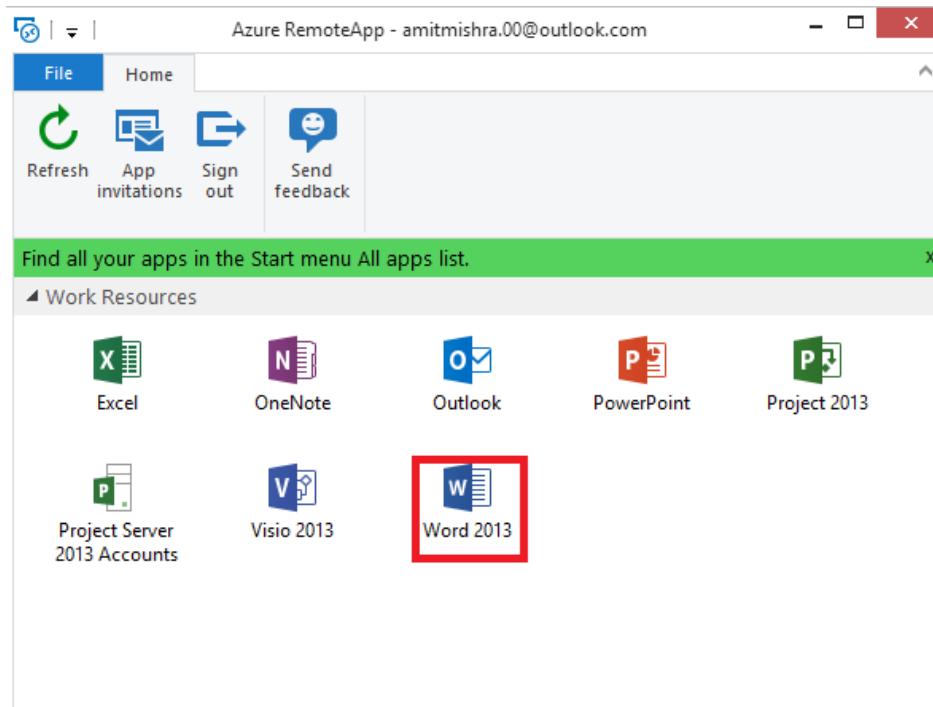
This will try to find your published app, but in few moments this will pop up to you Microsoft Account.



Enter your account details and click on to Sign In.



Here you will find your entire published app which is also published in Azure. Let's try some work on it. Click on to Word to launch it.



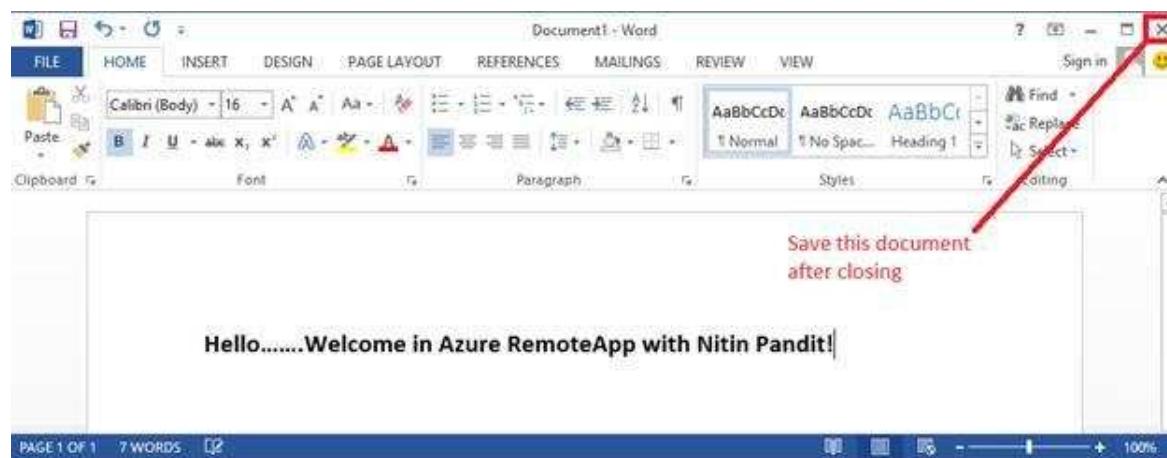
After clicking on any application which you want to working with that, Remote App will start connect with the service.

In my case, I'm going to launching word2013, and you can see remote desktop session has been launched.

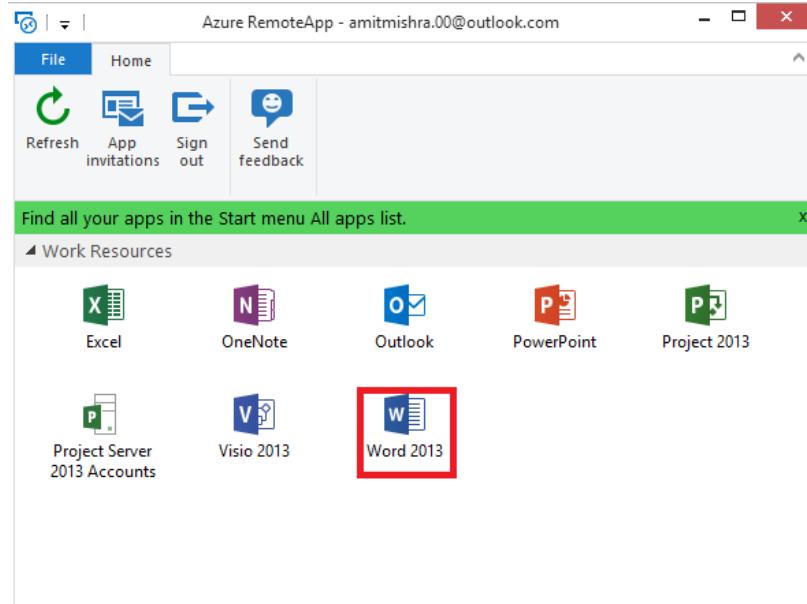


So we have successfully launched word, select blank document. Make some script in selected document, and I saved it with "Hello" named.

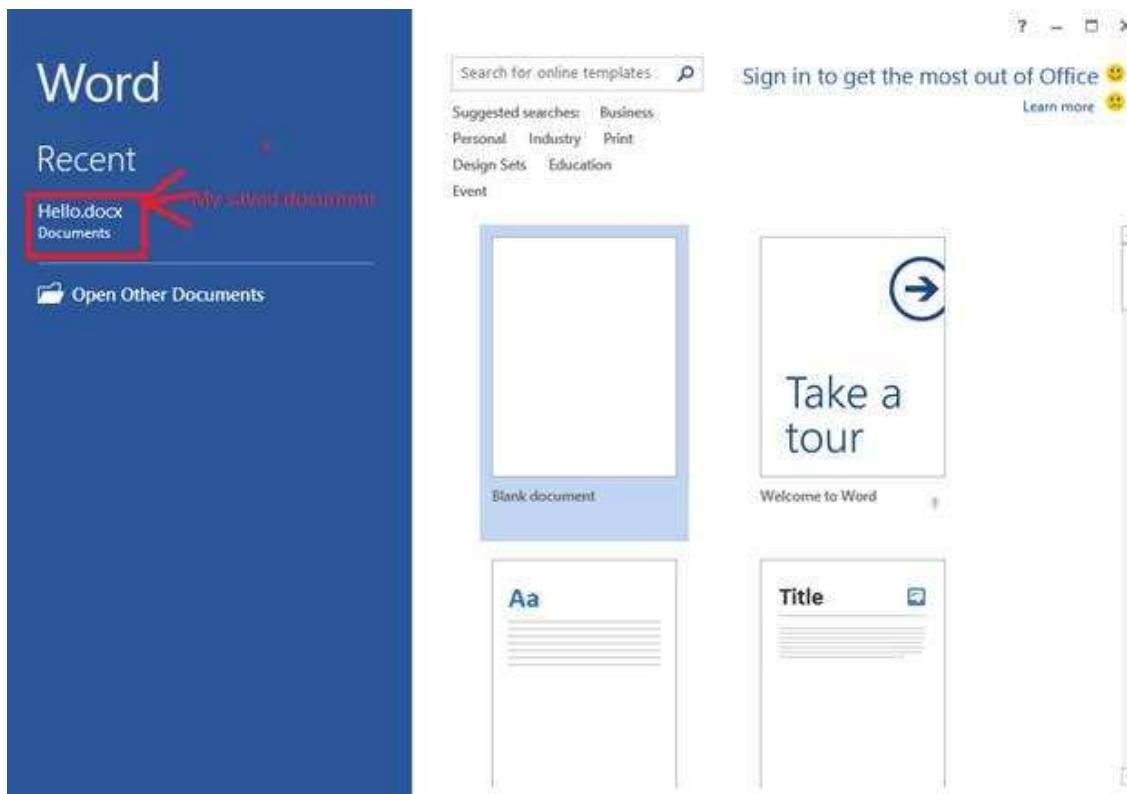
Keep in my mind each launched application is predicting to my email address.



Re-launch word, to see if the word is in there or not.

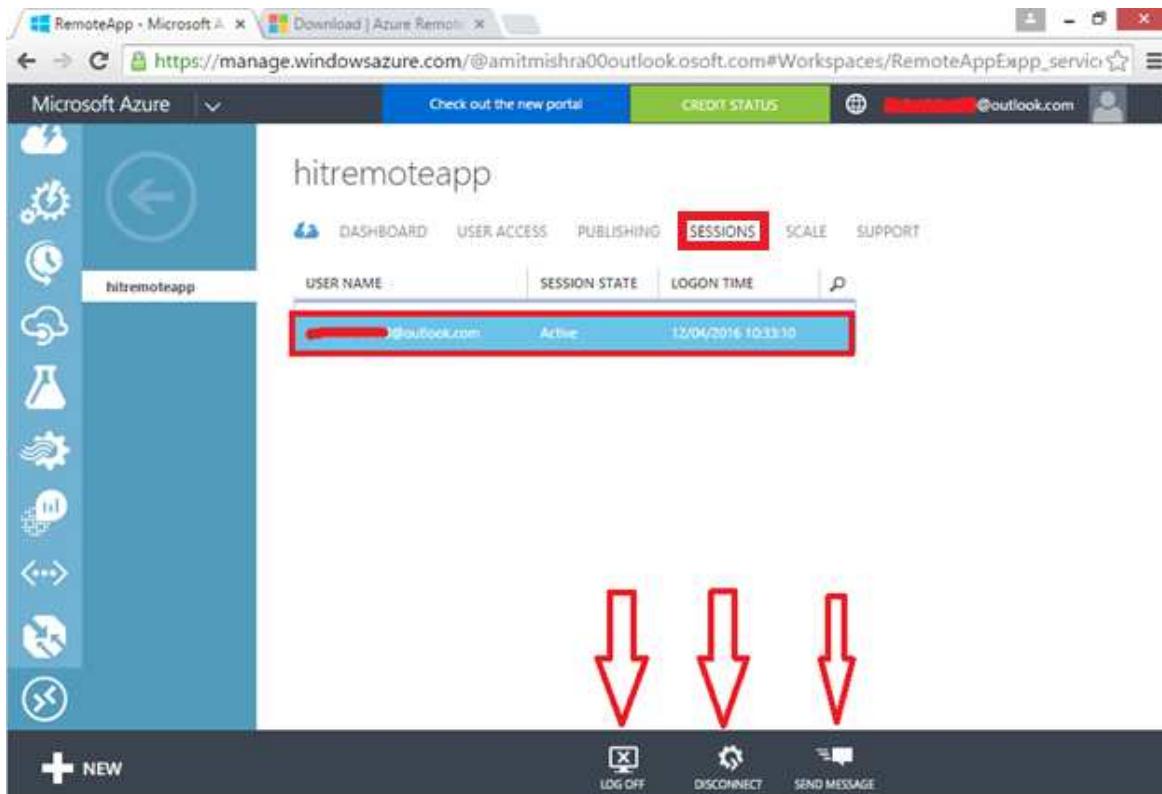


So we can see my Hello document is in Recent.



Now, back in to your Azure RemoteApp in cloud, click on to Sessions.

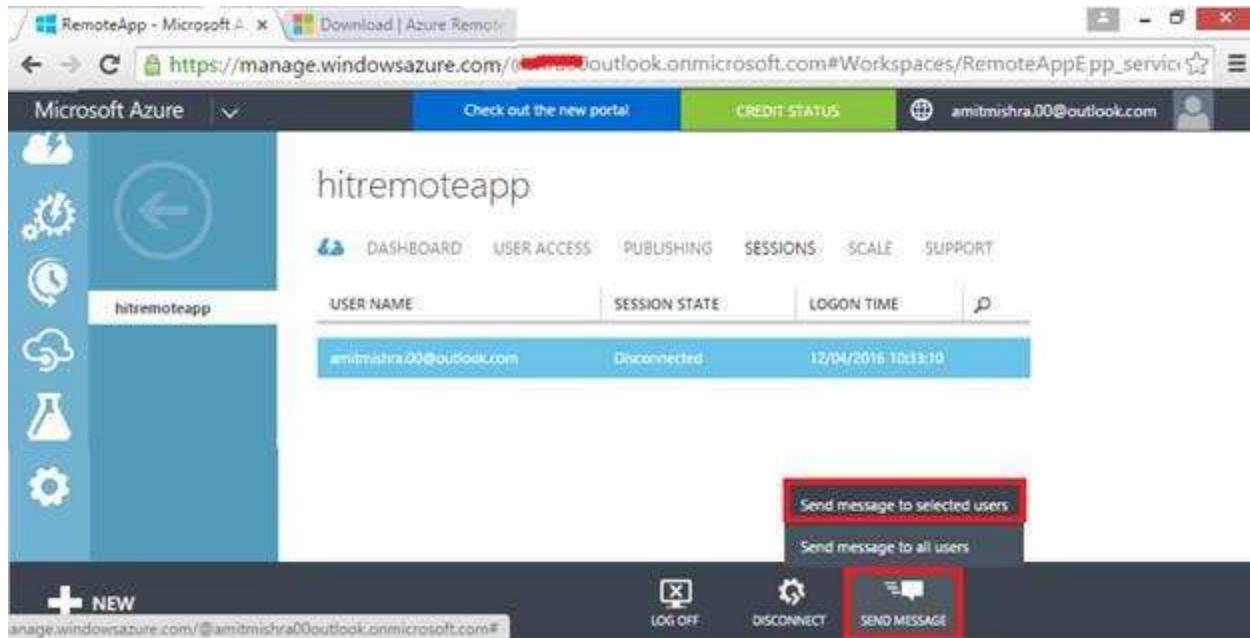
In Sessions, we can see my current session for the word is active, where with this session we can performed many options like, we can LogOff this session forcefully, we can disconnect it from administrative point of view and also we can send messages to the selected users or all users.



The screenshot shows the Microsoft Azure RemoteApp management interface. On the left, there's a sidebar with various icons: Cloud, Gear, Clock, Cloud, Test Tube, Gears, Network, and a plus sign labeled 'NEW'. The main area has tabs: DASHBOARD, USER ACCESS, PUBLISHING, SESSIONS (which is highlighted with a red border), SCALE, and SUPPORT. Below the tabs is a table with columns: USER NAME, SESSION STATE, and LOGON TIME. A single row is visible: [REDACTED]@outlook.com, Active, 12/04/2016 10:33:10. Three large red arrows point downwards from the text 'We want to send messages, click on to Send Messages where you will find two options for it.' towards the 'SEND MESSAGE' button at the bottom of the table row. At the bottom of the interface are buttons for LOG OFF, DISCONNECT, and SEND MESSAGE.

We want to send messages, click on to Send Messages where you will find two options for it. Send message to all users, which means you want to send your message to all users and Send message to selected users means you want to send message to the selected users.

I have active with only one user, so I have selected send message to selected users.

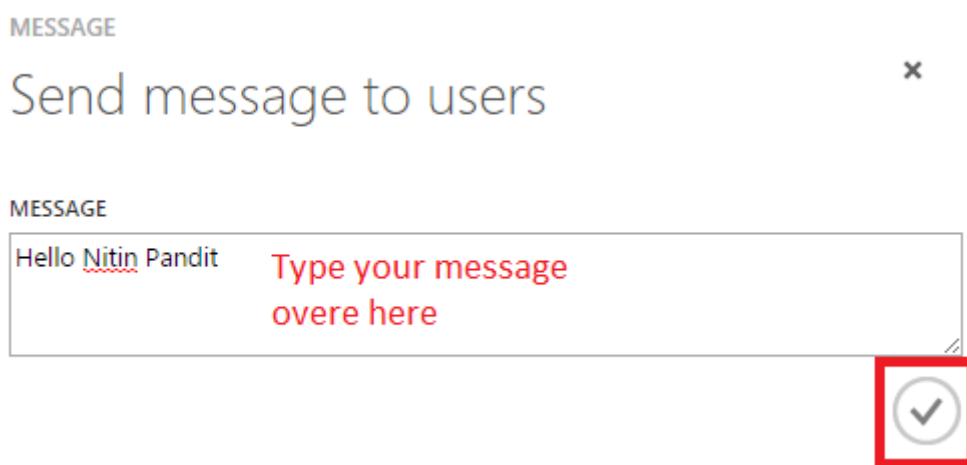


A screenshot of the Microsoft Azure RemoteApp Management Portal. The main title bar shows 'RemoteApp - Microsoft' and 'Download | Azure RemoteApp'. The URL is 'https://manage.windowsazure.com/...@outlook.onmicrosoft.com#Workspaces/RemoteAppEpp_service'. The top navigation bar includes 'Microsoft Azure', 'Check out the new portal', 'CREDIT STATUS', and a user account 'amitmishra.00@outlook.com'. On the left, there's a sidebar with icons for Cloud, Gears, Clock, Cloud, Test Tube, and Settings, with 'hitremoteapp' selected. The main content area shows a session titled 'hitremoteapp' with a table of user information:

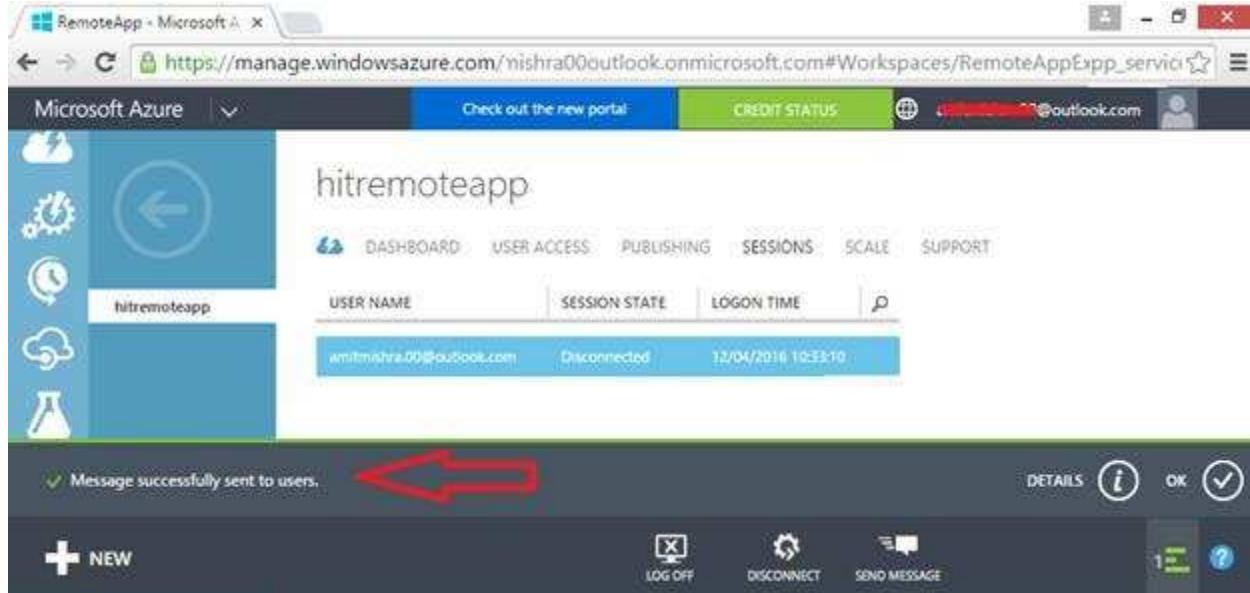
USER NAME	SESSION STATE	LOGON TIME
amitmishra.00@outlook.com	Disconnected	12/04/2016 10:13:10

Below the table, there's a message box with two options: 'Send message to selected users' and 'Send message to all users'. At the bottom right of the message box is a red-bordered button labeled 'SEND MESSAGE'.

A pop up will open, type a message which you want to send to the usres. After typing your message, click on to check button.



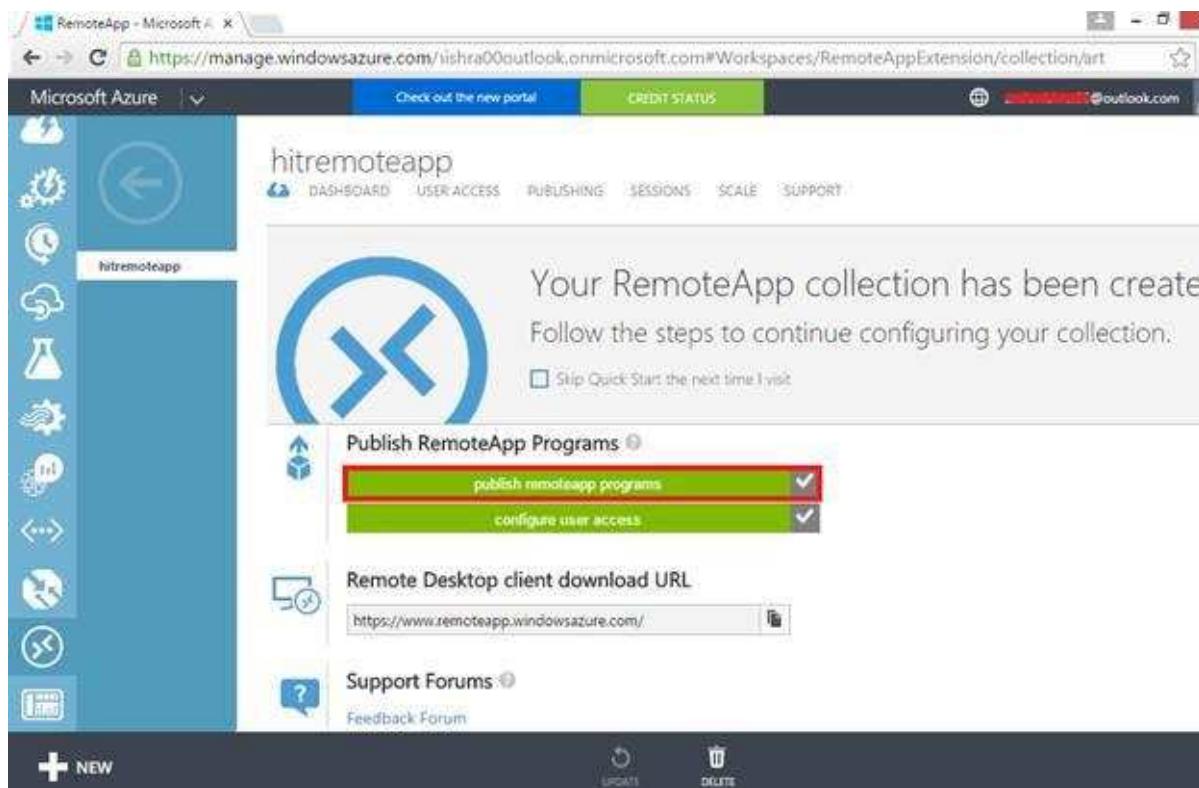
In the following screenshot you can see my message has been successfully sent to the users.



A screenshot of the Microsoft Azure RemoteApp management interface. The left sidebar shows a list of remote applications, with 'hitremoteapp' selected. The main panel displays the 'hitremoteapp' details, including a user session for 'amitmishra.00@outlook.com' which is 'Disconnected' at '12/04/2016 10:33:10'. A red arrow points to a success message at the bottom: 'Message successfully sent to users.' Below the message are buttons for 'DETAILS', 'OK', and a checked '✓' button.

Now we want to publish some Remote app programs, click on to your remote, which you have created.

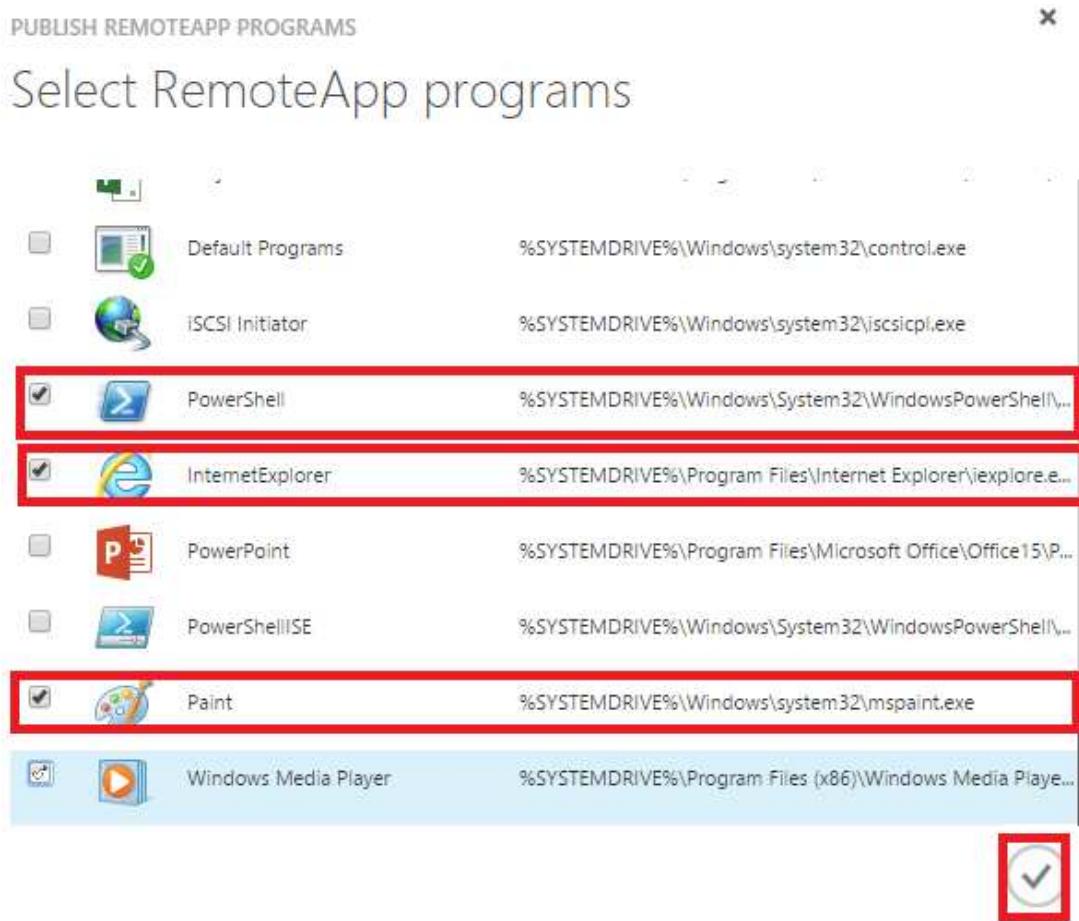
Click on to “publish RemoteApp Programs” to publish apps, which you want to publish.



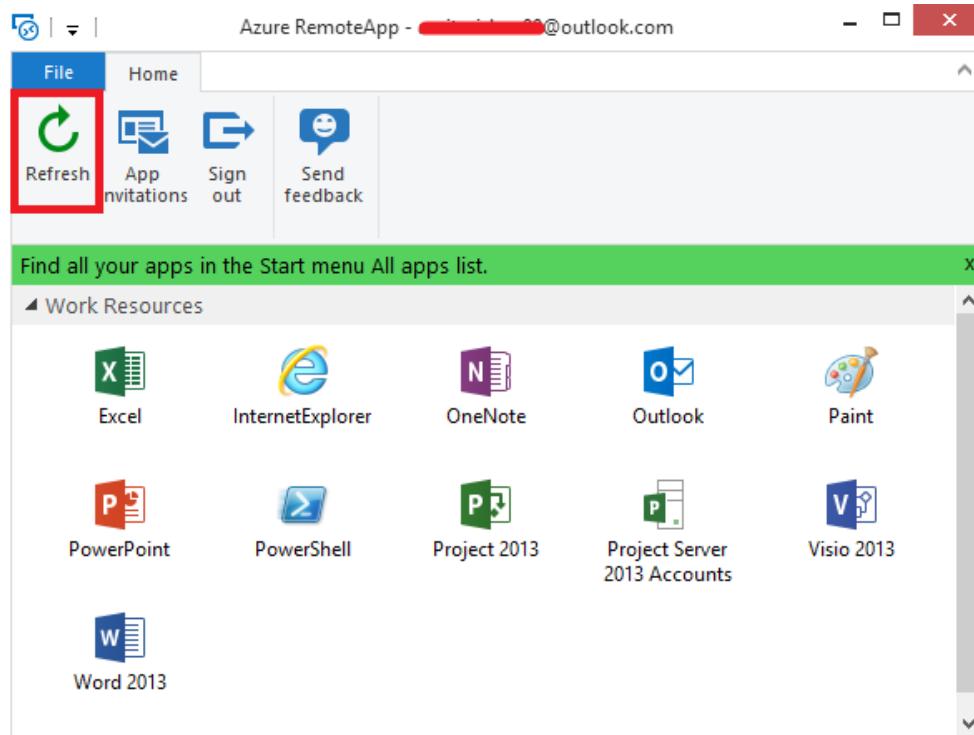
A screenshot of the Microsoft Azure RemoteApp collection creation page. It shows a success message: 'Your RemoteApp collection has been created. Follow the steps to continue configuring your collection.' There is a checkbox for 'Skip Quick Start the next time I visit'. Below, under 'Publish RemoteApp Programs', two options are listed: 'publish remoteapp programs' (selected) and 'configure user access' (selected). Other sections include 'Remote Desktop client download URL' (set to https://www.remoteapp.windowsazure.com/) and 'Support Forums' (Feedback Forum). A red box highlights the 'publish remoteapp programs' option.

There will be a number of programs, select app programs which you want to publish.

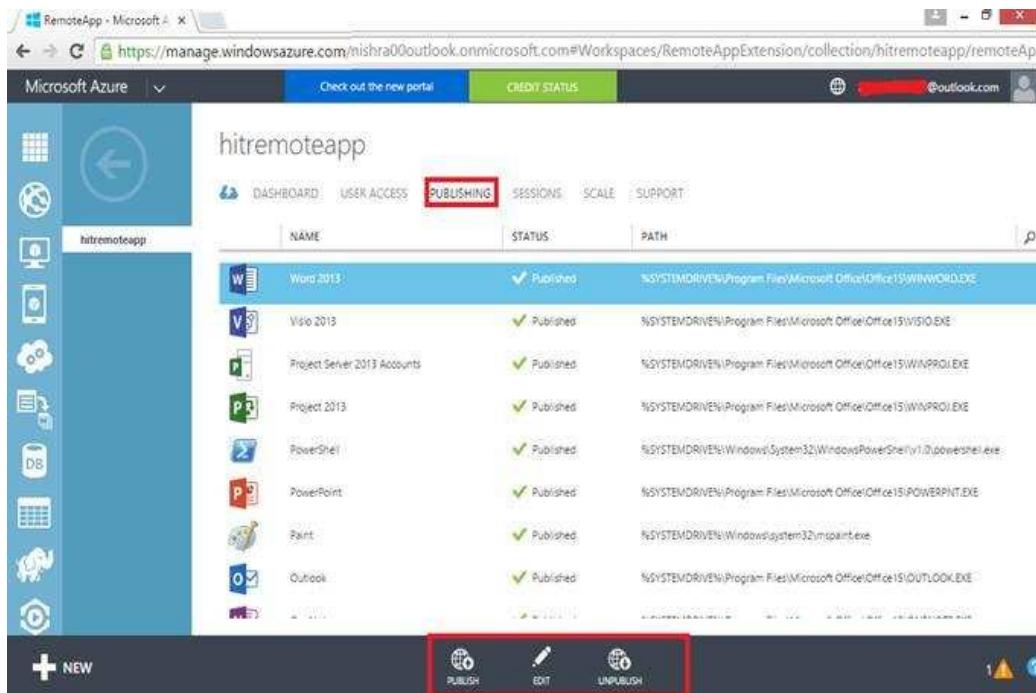
After selected programs, click on to check button.



The page will show a message of publishing, just go to Azure Remote App and refresh it, where you can find your selected publish app program.

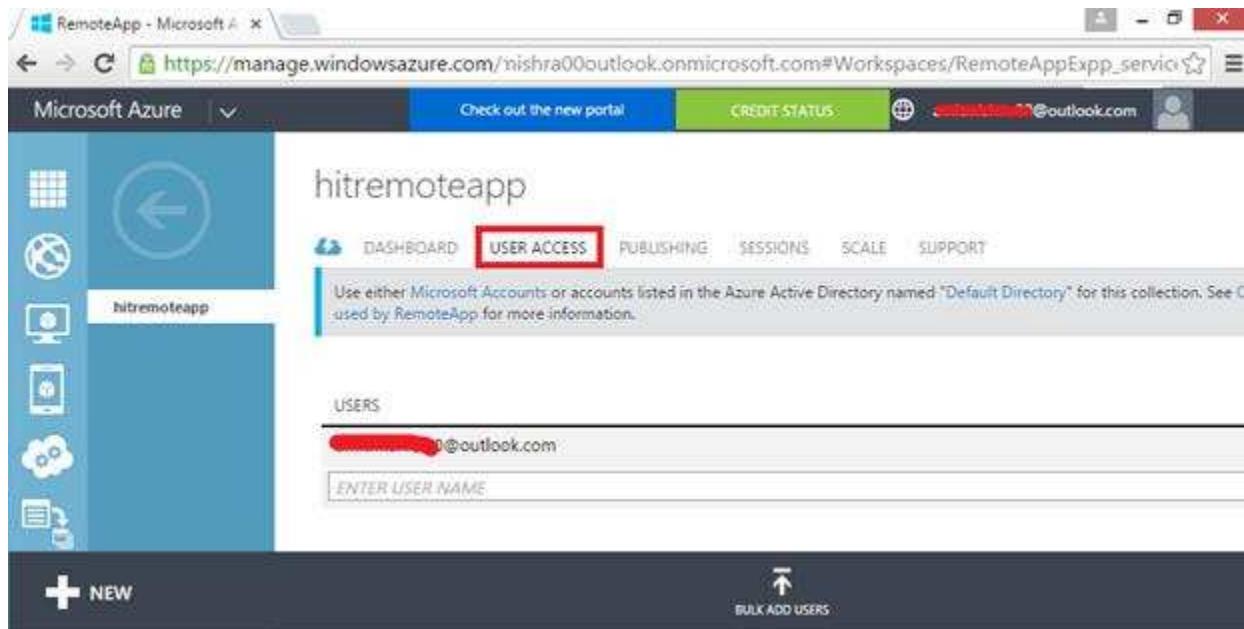


The changes can be seen in publishing option, so you can simply perform many options of one program like publish, edit and unpublishing of the particular app program.



The screenshot shows the Microsoft Azure portal for managing a remote application named 'hitremoteapp'. The 'PUBLISHING' tab is selected. On the left, there's a sidebar with various icons. The main area lists published applications: Word 2013, Visio 2013, Project Server 2013 Accounts, Project 2013, PowerShell, PowerPoint, Paint, and Outlook. Each item has a status of 'Published' and a path listed. At the bottom, there are buttons for '+ NEW', 'PUBLISH' (highlighted with a red box), 'EDIT', and 'UNPUBLISH'.

Go to the User Access tab, how many users are there, you can add more users, and can perform many operations.



The screenshot shows the Microsoft Azure portal interface for managing a RemoteApp collection named 'hitremoteapp'. The 'USER ACCESS' tab is selected. A message box states: 'Use either Microsoft Accounts or accounts listed in the Azure Active Directory named "Default Directory" for this collection. See Collection settings used by RemoteApp for more information.' Below this, there is a 'USERS' section with an input field containing a placeholder email address ('[REDACTED]@outlook.com') and a 'ENTER USER NAME' button. At the bottom right of the users section is a 'BULK ADD USERS' button with an upward arrow icon.

Add there user name or email id and share with them.

I hope you enjoyed this chapter.

Microsoft Azure Service Bus

The Service Bus is a communication engine which provides secure and reliable communication between components. Service Bus allows you to build applications and connect with different roles between Web and worker TL, you can use this for load labeling and or for decoupling where your web TS sent messages into a queue. Service Bus is a unified set of messaging capabilities.

With Azure Service Bus you can build cloud scale applications for the enterprise. Service Bus provides you Messaging and Relay Services and really helps to unblock enterprise data as well as business logic.

With Service Bus you can enable loosely coupled solutions.

Window Azure Service Bus provides multi-tenant service to the connecting applications.

You can use four different communications mechanism of one or more instances within a namespace.

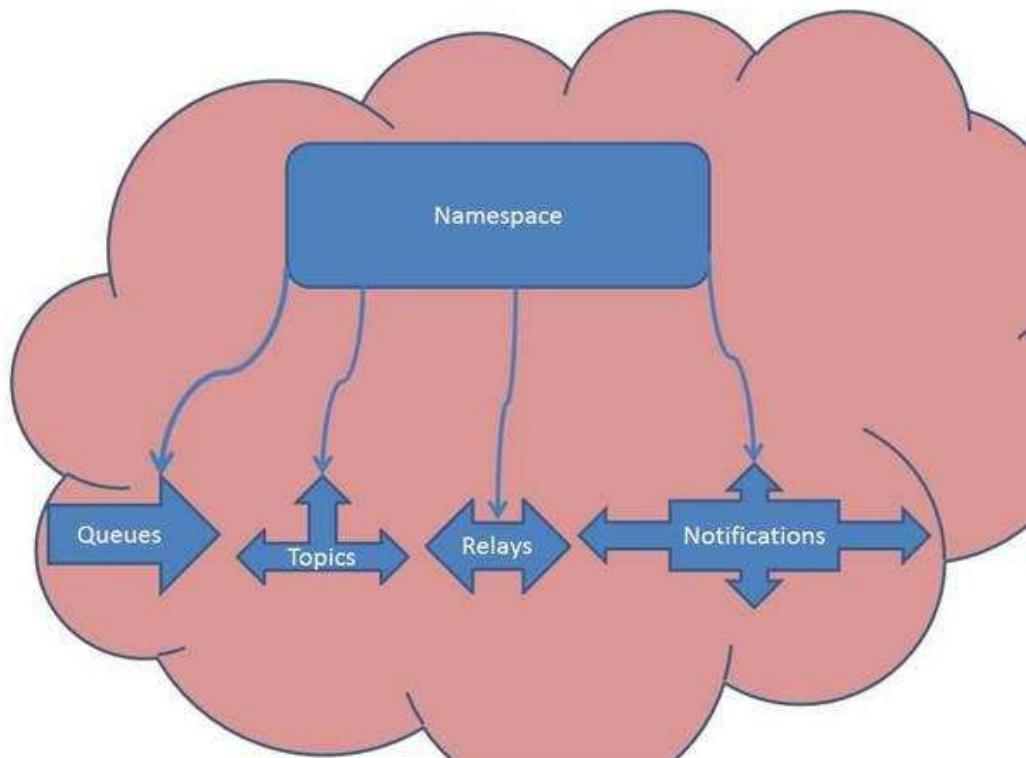
Queues: This allows one directional communications (also called Broker).

Topics: This allows one directional communications using subscriptions.

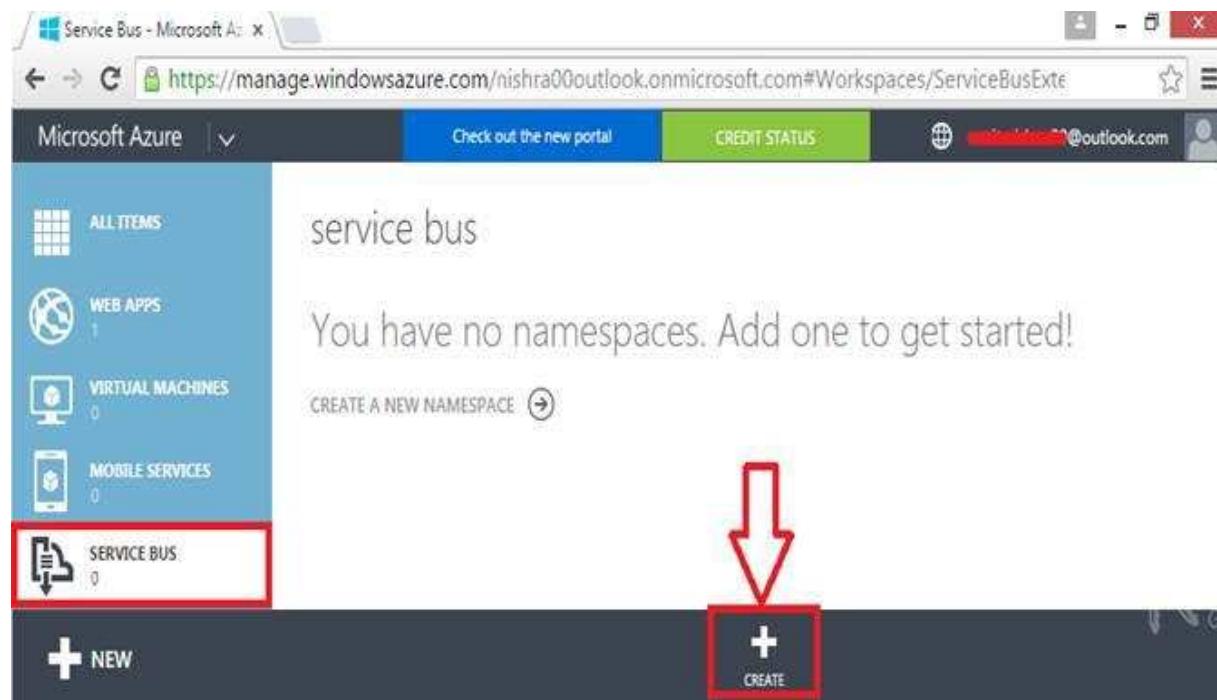
Relays: This is responsible to providing bi-directional communications. It passes the messages on to destination application.

Event Hubs: This is responsible to provide event and telemetry entry to the cloud.

The following figure shows, in a one namespace there may be many entities can be reside or created.



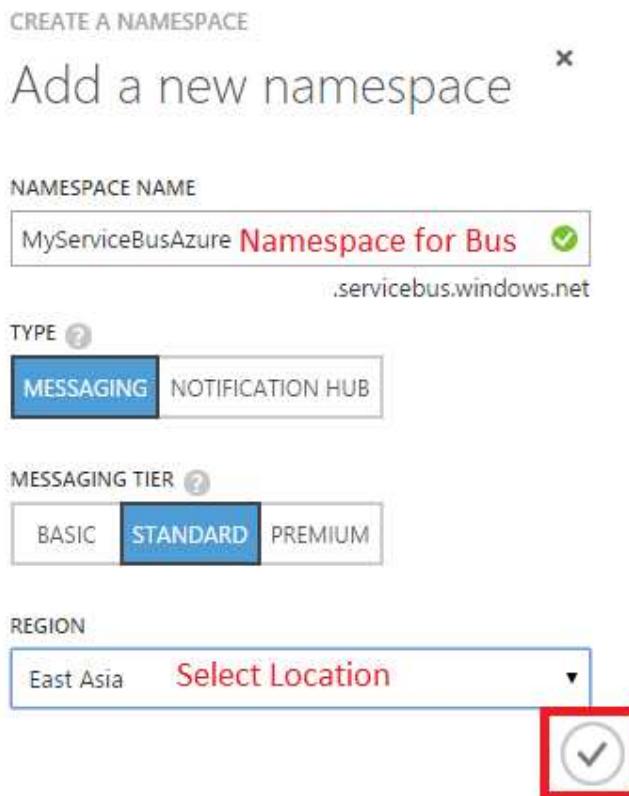
Now start with creating a new namespace, go to the Service Bus and click on to create a new namespace.



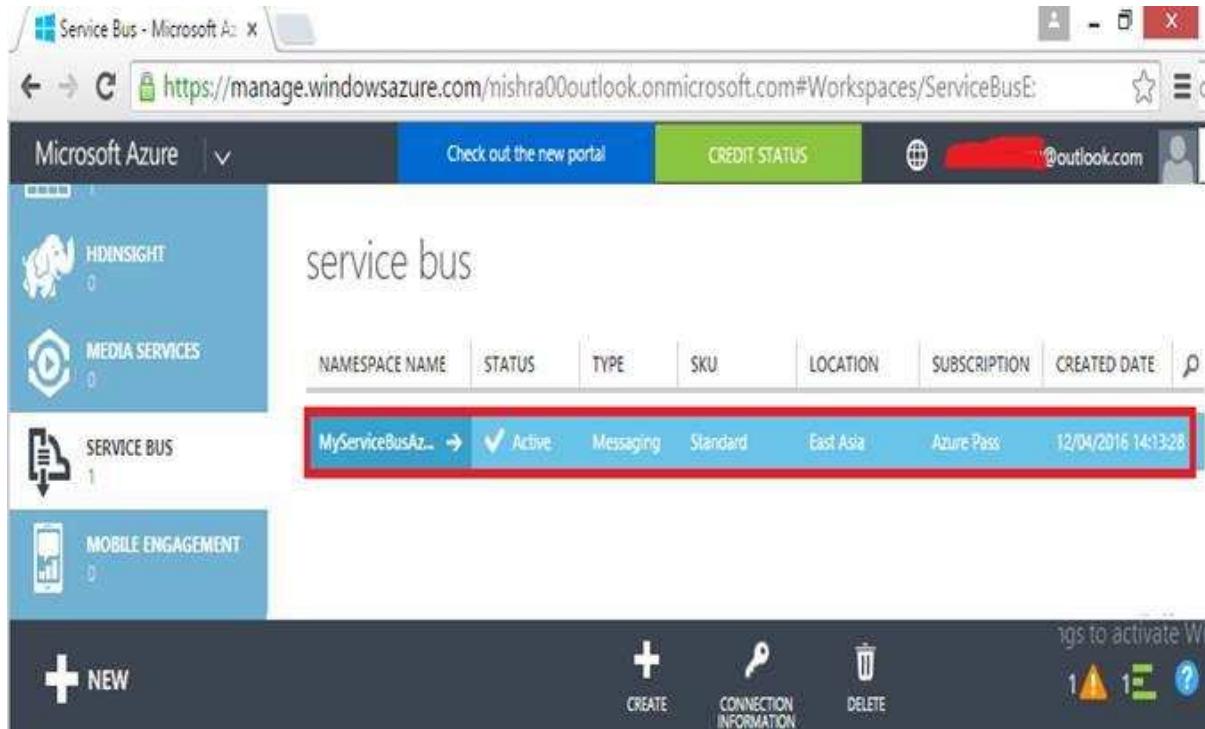
A screenshot of the Microsoft Azure Service Bus management portal. The URL in the browser is <https://manage.windowsazure.com/nishra00outlook.onmicrosoft.com#Workspaces/ServiceBusExt>. The page title is "Service Bus - Microsoft Azure". The left sidebar shows "ALL ITEMS" (0), "WEB APPS" (1), "VIRTUAL MACHINES" (0), "MOBILE SERVICES" (0), and "SERVICE BUS" (0). The "SERVICE BUS" item is highlighted with a red border. The main content area has a heading "service bus" and a message "You have no namespaces. Add one to get started!". Below this is a "CREATE A NEW NAMESPACE" button with a right-pointing arrow. At the bottom right is a large black "CREATE" button with a white plus sign. A red arrow points down to this "CREATE" button.

Namespace is a Global unique container. We can put all entities in namespace like Queues, Topics, Relays and Event Hubs. So now we need to create a namespace for Azure Service Bus, give a name to your service, type and messaging tier would by default add select the region where you want to create your Service Bus.

In the type, I'm selecting Messaging; Notification Hub is for how to send notifications to mobile devices.

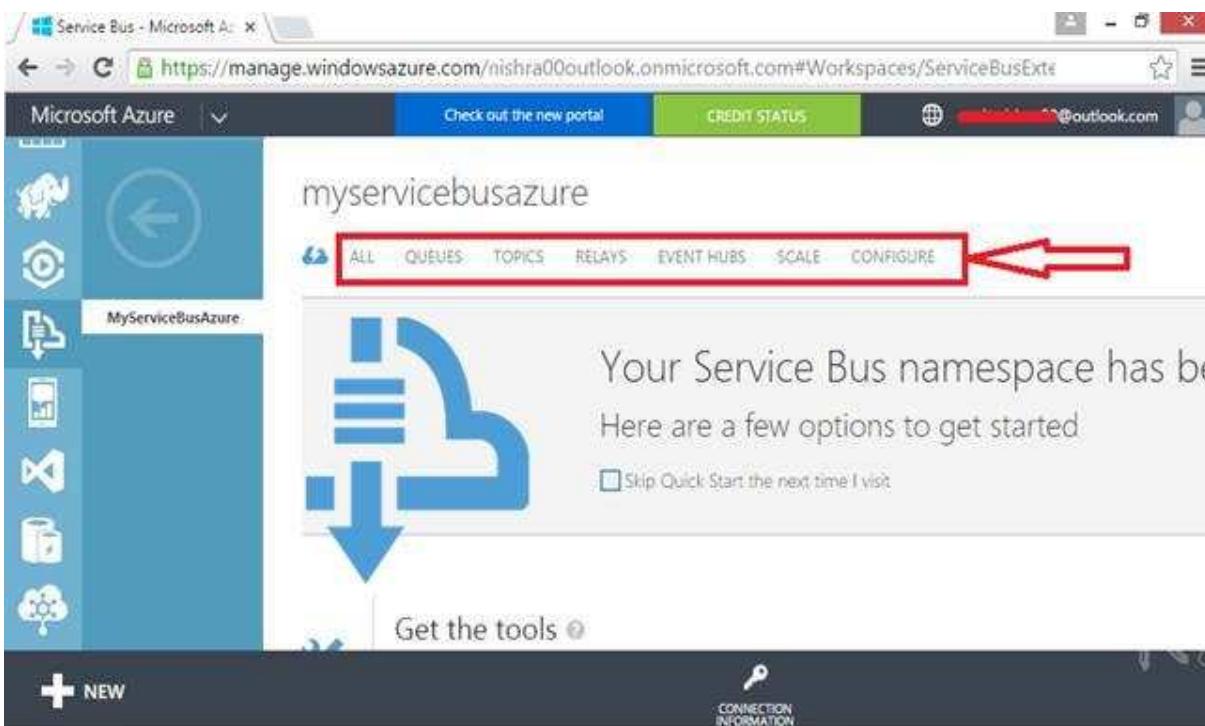


So here, my Service Bus namespace has been created “MyServiceBusAzure”. Click on to namespace as shown in the following screenshot:

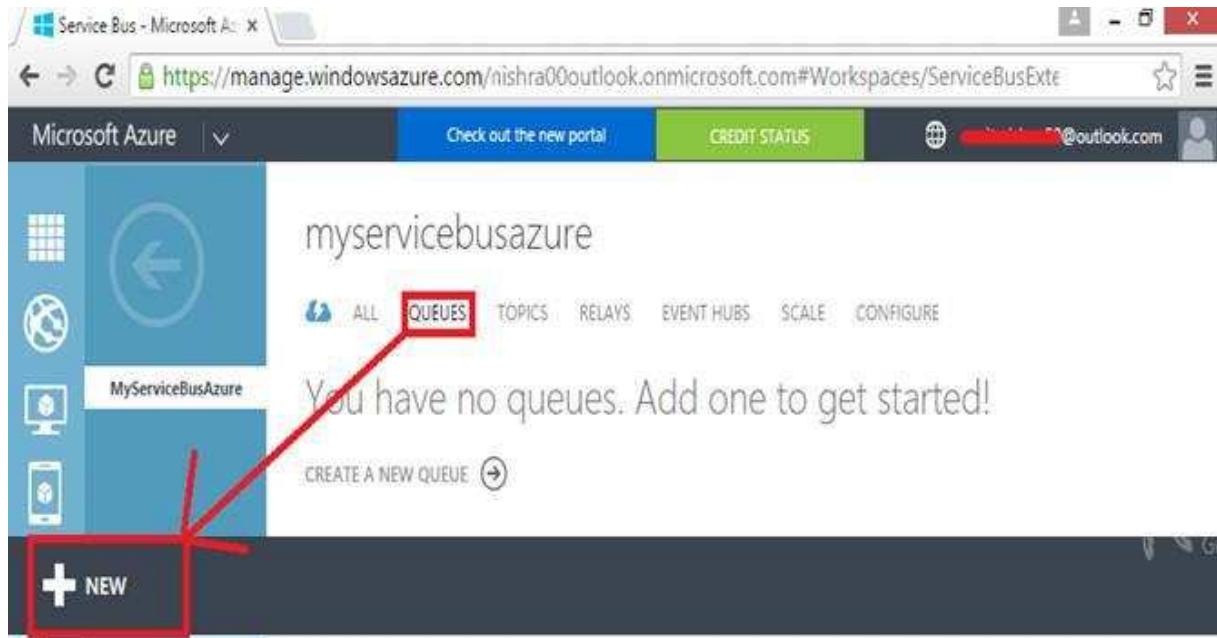


NAMESPACE NAME	STATUS	TYPE	SKU	LOCATION	SUBSCRIPTION	CREATED DATE
MyServiceBusAzure	Active	Messaging	Standard	East Asia	Azure Pass	12/04/2016 14:13:28

There are many entities inside the namespace like Queues, Topics, Relays, and Event Hubs.



Now first I'm going to create a Queue, click on *QUEUE* and click on “*NEW*” button in the bottom to create a Queue.

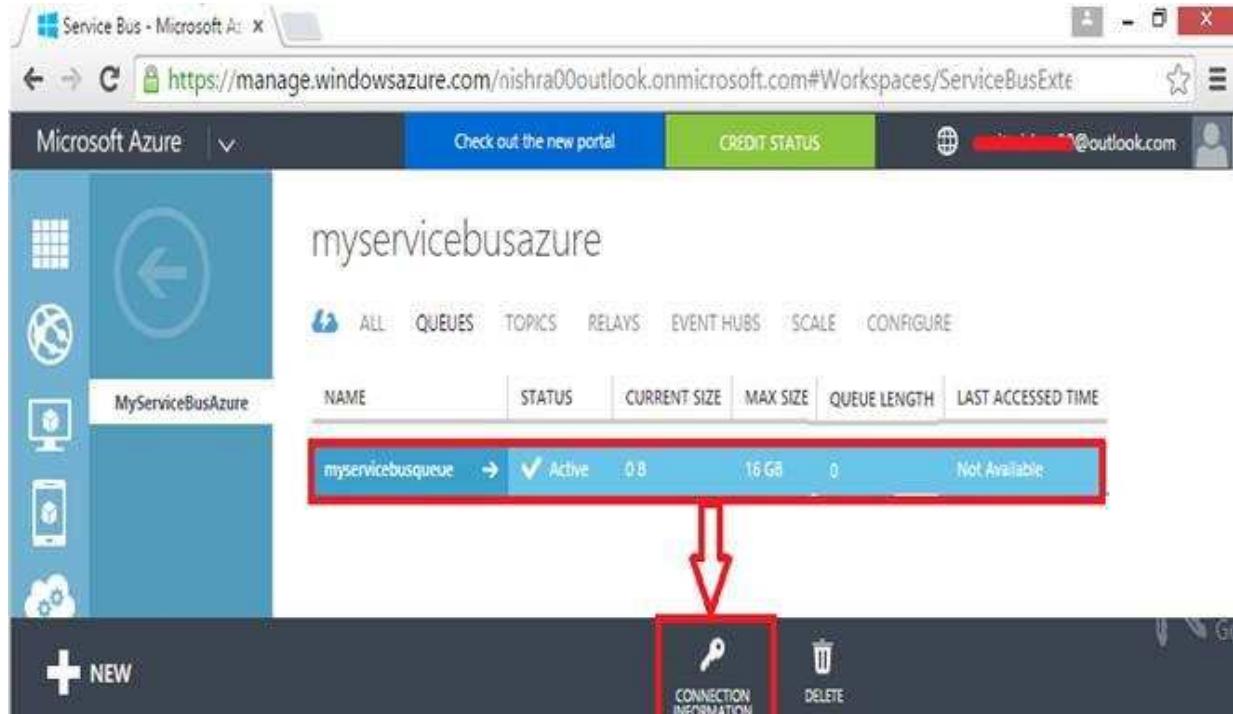


Inside *Service Bus*, click *Queue*, then *Quick Create*, a box would open for creating a new Queue, give a name to your queue, select the region for queue and namespace would be “*MyServiceBusAzure*”.

Click on to “*Create a New Queue*”.



So my Queue gets created, now what we can do with queue, queue is inside in my container (namespace). Here, I need a connection string of my queue, let's check connection string is available at this time, for this go to “*Connection Information*”.



The screenshot shows the Microsoft Azure Service Bus Management Portal. In the center, there is a list of queues under the namespace "myservicebusazure". One queue, "myservicebusqueue", is highlighted with a red box. Below the list, there are two buttons: "CONNECTION INFORMATION" and "DELETE". A red arrow points from the "CONNECTION INFORMATION" button to the text below it.

NAME	STATUS	CURRENT SIZE	MAX SIZE	QUEUE LENGTH	LAST ACCESSED TIME
myservicebusqueue	Active	0.8	16 GB	0	Not Available

So there is no Shared Access Service (SAS) available at this time, make sure connection string is very important to communicate with this queue to my application. For access connection string you need to define what are the authorizations, what are the capabilities you can do, may be you can also create items in a queue, may be you can read items from a queue as well as you can manage your queue. Let's see how we can configure that,

Access connection information

Use this connection information to connect to queue 'myservicebusqueue'.

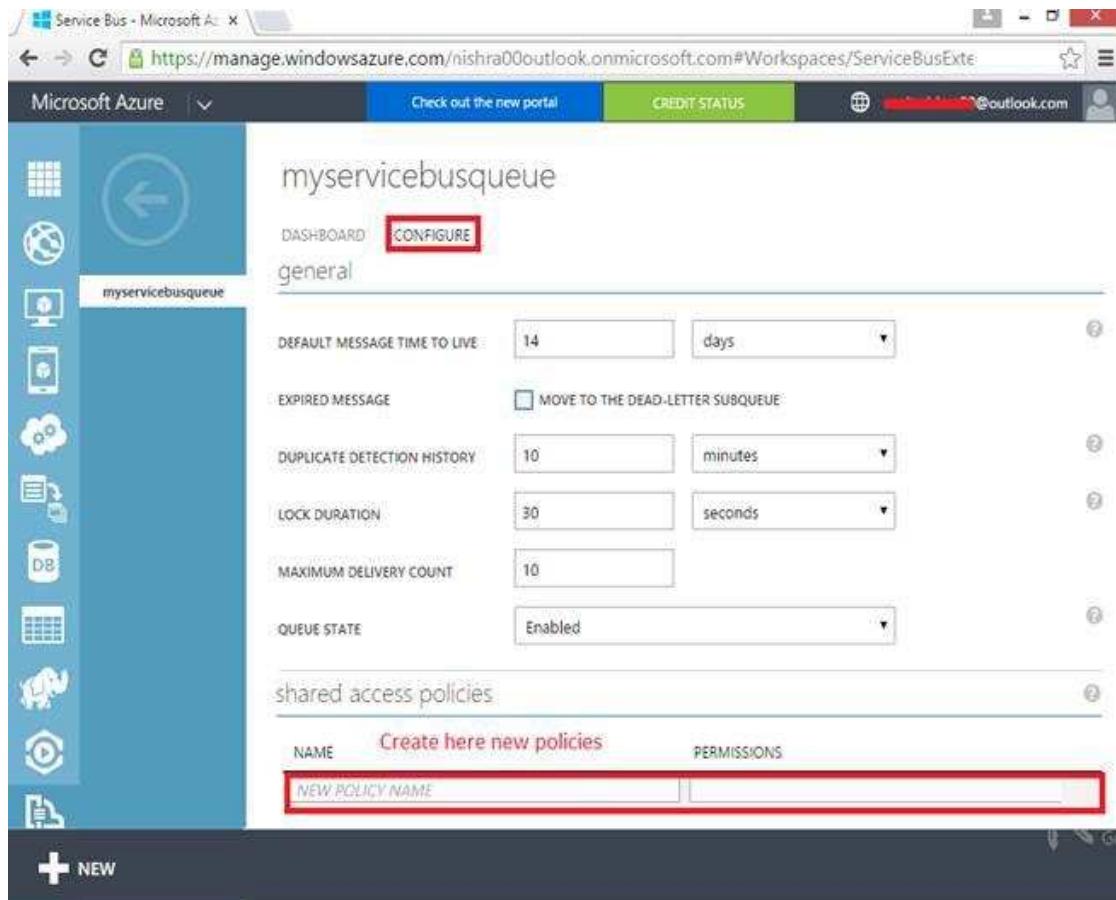
SAS 

You do not have any SAS Policies configured for this entity.

CLICK HERE TO CONFIGURE 

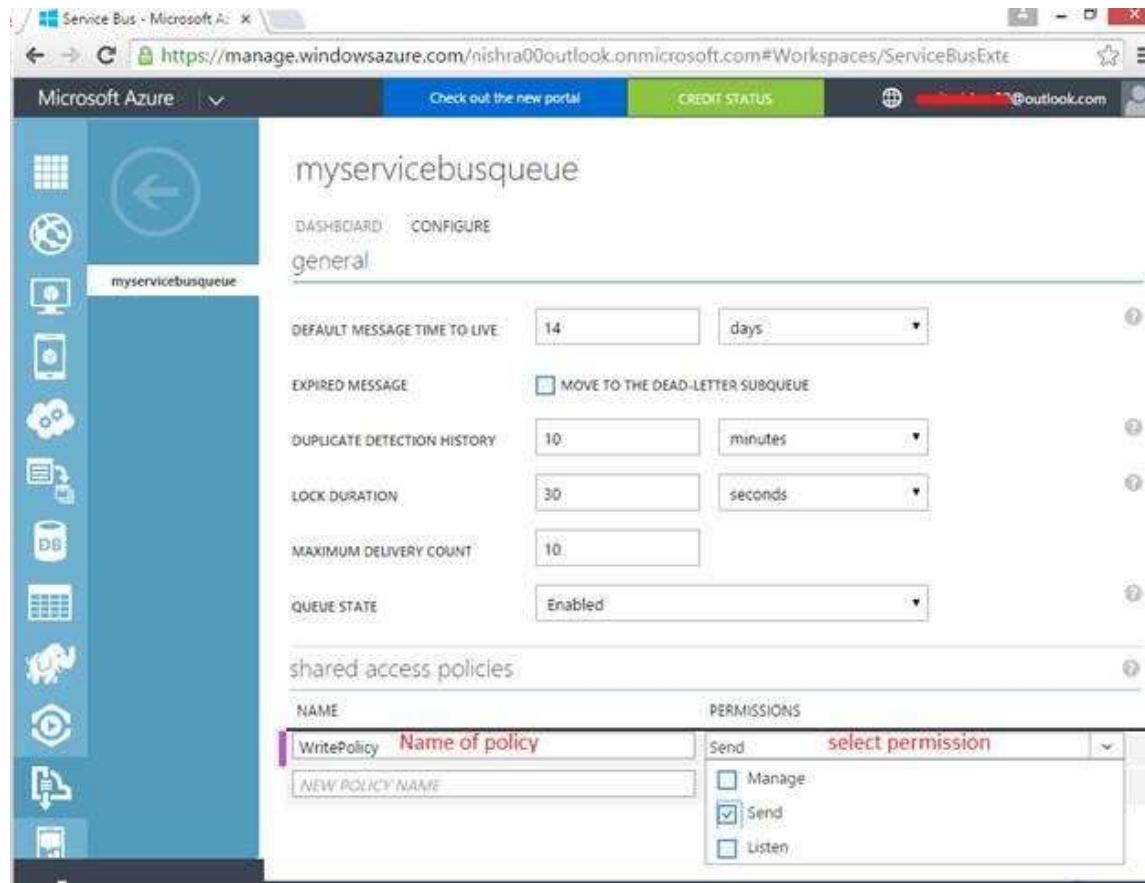


In *CONFIGURE* tab, scroll down and you will see Shared Access policies; access policy is responsible to provide the access rights.

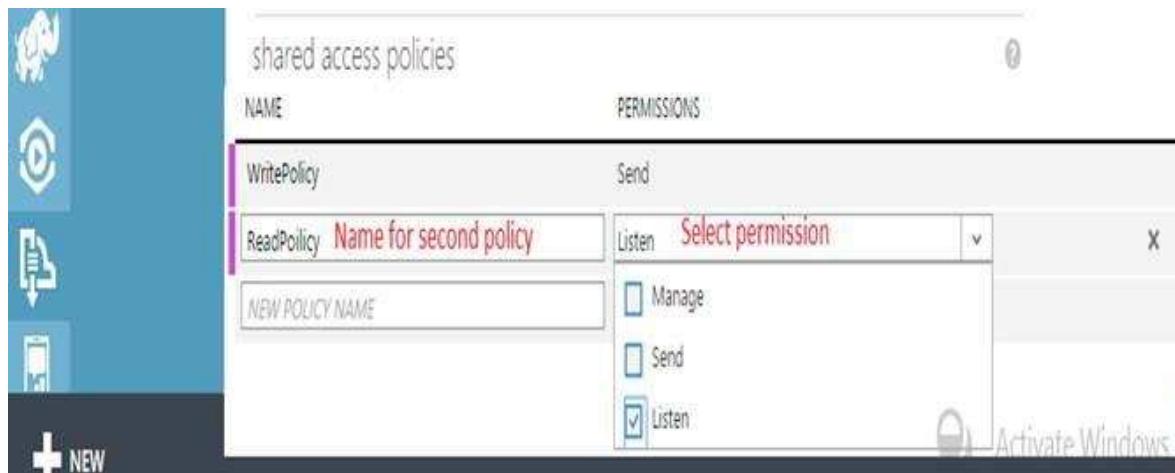


NAME	Create here new policies	PERMISSIONS
NEW POLICY NAME		<input type="button" value="..."/>

Let's create a new permission; we have given it "WritePolicy" name to my policy, what is the permissions available, click on to dropdown. You can receive three things: you can manage, send and listen. I'm interested in sending, so I have selected *Send*.



I created a second policy that is “ReadPolicy”, so I select “Listen” permission for it.



NAME	PERMISSIONS
WritePolicy	Send
ReadPolicy	Send Listen
NEW POLICY NAME	

Click on to “Save” button to save your policies.

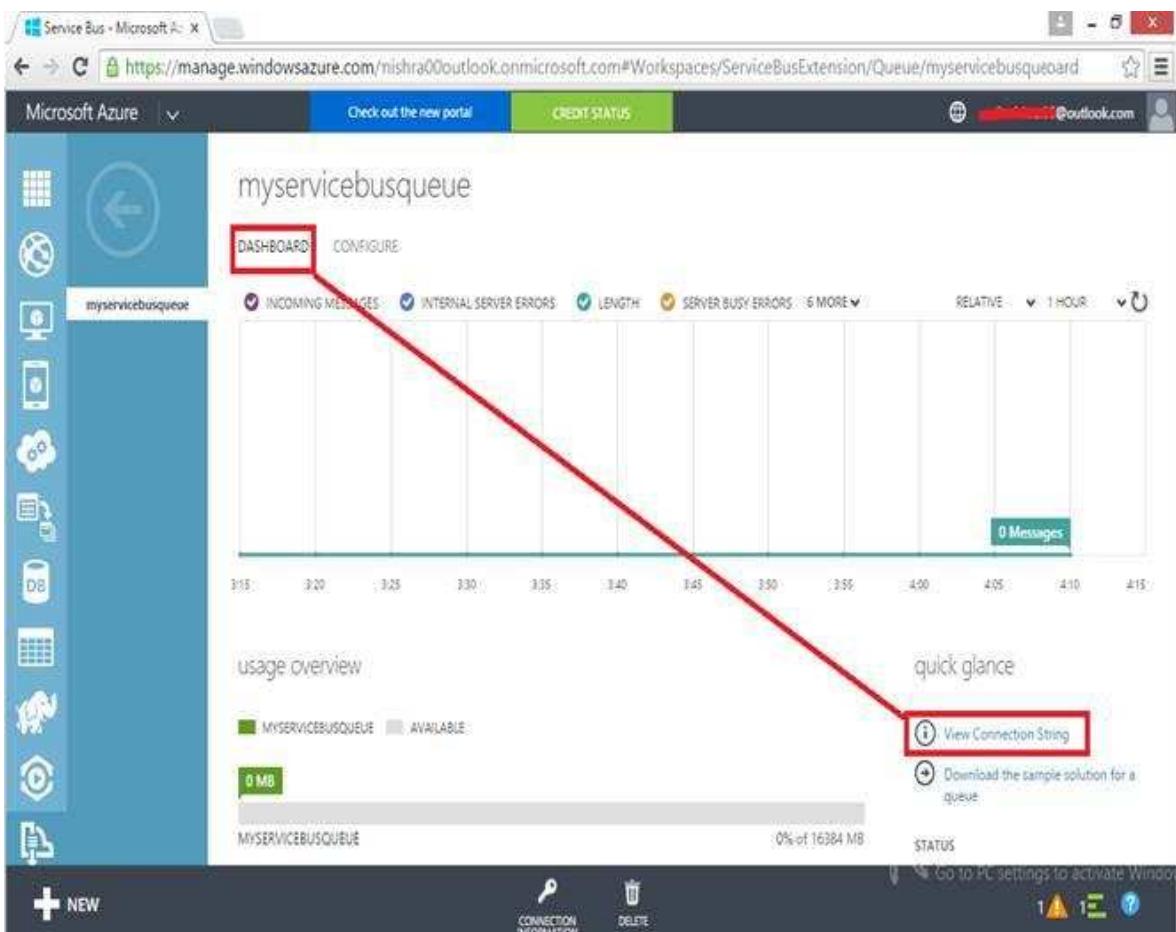
shared access policies

NAME	PERMISSIONS
WritePolicy	Send
ReadPolicy	Listen
<input type="text" value="NEW POLICY NAME"/>	<input type="button" value=""/>

Go to PC settings to activate Windows Firewall

1

Now I have two policies, now let's see my Queue Dashboard, scroll down and select "View Connection String".



The screenshot shows the Microsoft Azure Queue Dashboard for a queue named 'myservicebusqueue'. The 'DASHBOARD' tab is active. In the 'quick glance' section, there is a link labeled 'View Connection String' which is highlighted with a red box. The dashboard also displays a timeline showing '0 Messages' and usage statistics for 'MYSERVICEBUSQUEUE'.

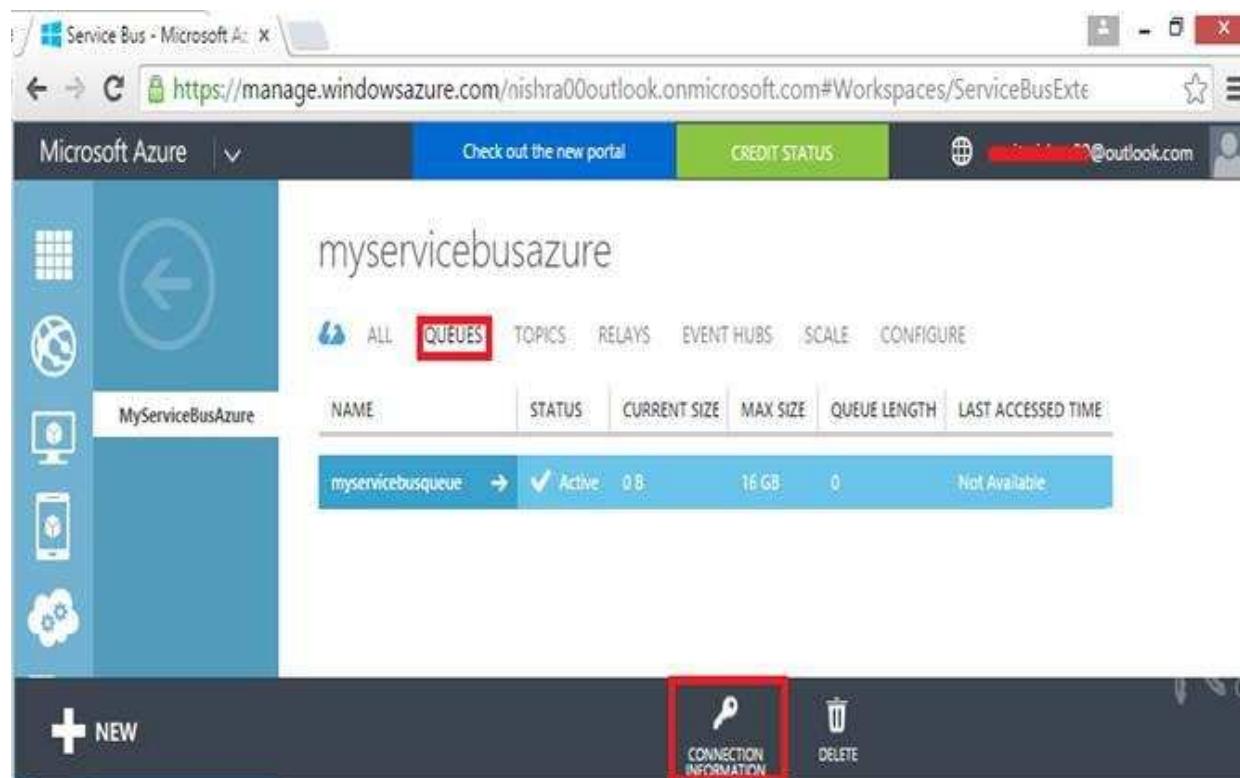
Here you can see the End Points of both policies, If I want someone to write things in my queue I can give this connection string to that person as well as to read things I can give read connection string to that person.

Access connection information

Use this connection information to connect to queue 'myservicebusqueue'.

SAS	NAME	CONNECTION STRING
WritePolicy		Endpoint=sb://myservicebusazure.servicebus.windows.net/;SharedAccessKeyName=W
ReadPolicy		Endpoint=sb://myservicebusazure.servicebus.windows.net/;SharedAccessKeyName=R

Now you can check the same connection string of both policies in the Queue tab. Click on *Connection Information* at the bottom of the page.



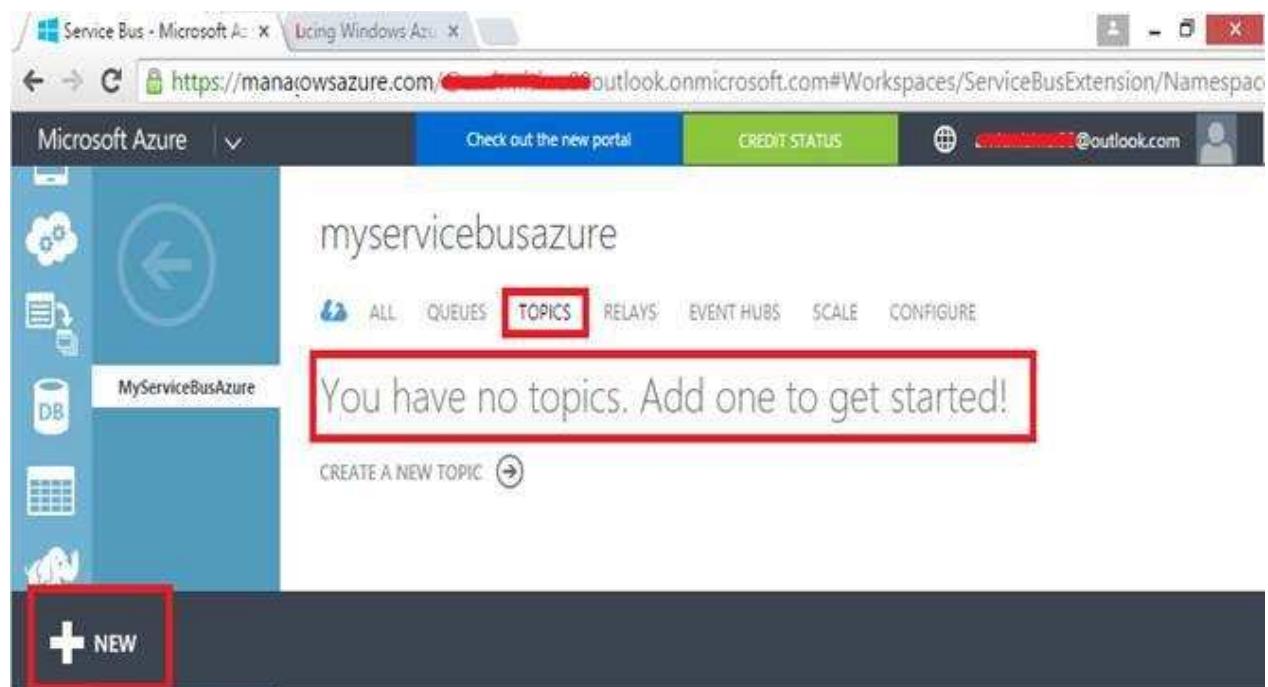
Here in the following screenshot, you can see same result.

Access connection information

Use this connection information to connect to queue 'myservicebusqueue'.

NAME	CONNECTION STRING
WritePolicy	Endpoint=sb://myservicebusazure.servicebus.windows.net/;SharedAccessKeyName=W
ReadPolicy	Endpoint=sb://myservicebusazure.servicebus.windows.net/;SharedAccessKeyName=R

In *TOPICS* panel, click on to New button to create a new topic.



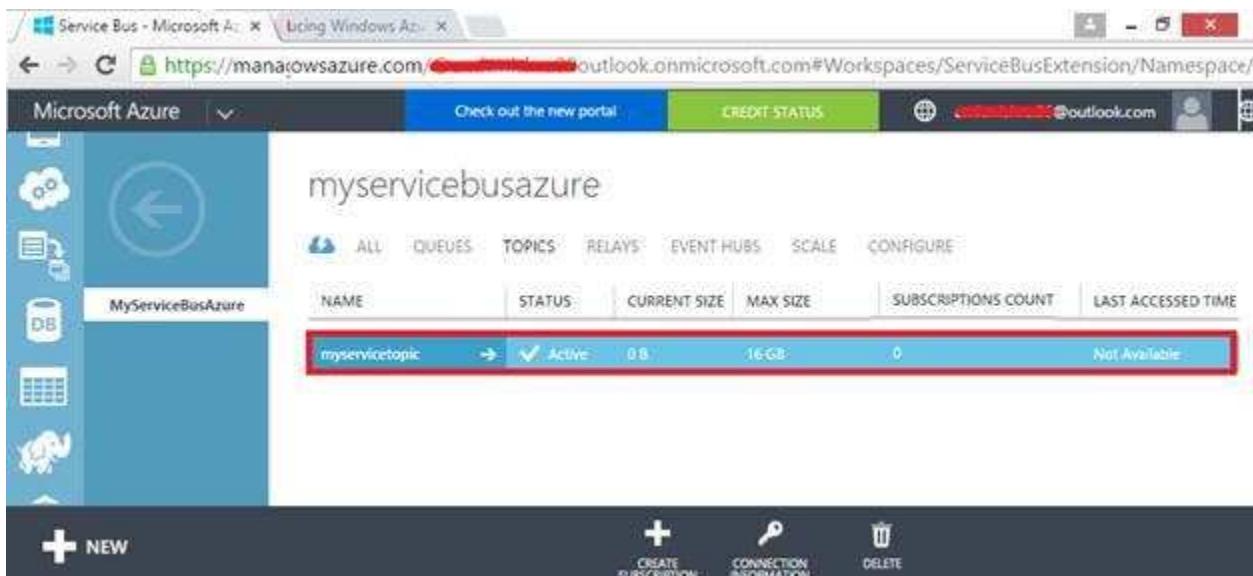
The screenshot shows the Microsoft Azure Service Bus Management Portal. The URL in the browser is https://management.azure.com/. The main navigation bar includes 'Check out the new portal', 'CREDIT STATUS', and a user profile. On the left, there's a sidebar with icons for Cloud, Queue, Topic, Relay, Event Hub, and Scale, with 'MyServiceBusAzure' selected. The main content area is titled 'myservicebusazure'. Below the title, there are tabs for 'ALL', 'QUEUES', 'TOPICS' (which is highlighted), 'RELAYS', 'EVENT HUBS', 'SCALE', and 'CONFIGURE'. A message 'You have no topics. Add one to get started!' is displayed. At the bottom left, there's a red-bordered 'CREATE A NEW TOPIC' button with a plus sign icon. A red box highlights this button.

Give the name of topic and select region/location of the topic, select your namespace for it.

Click on to “Create a New Topic”.



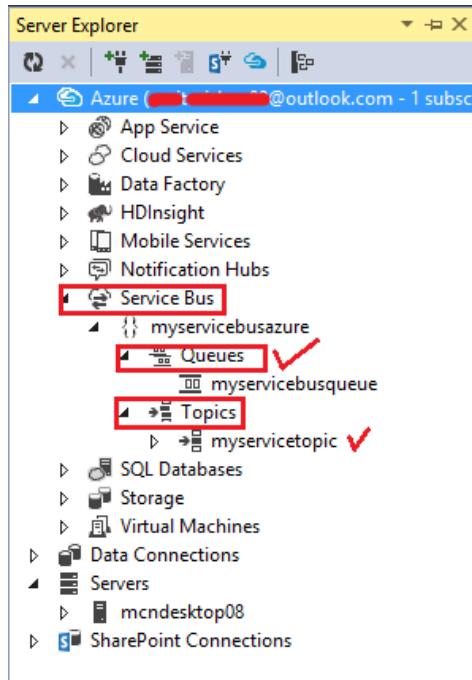
Here my topic is ready.



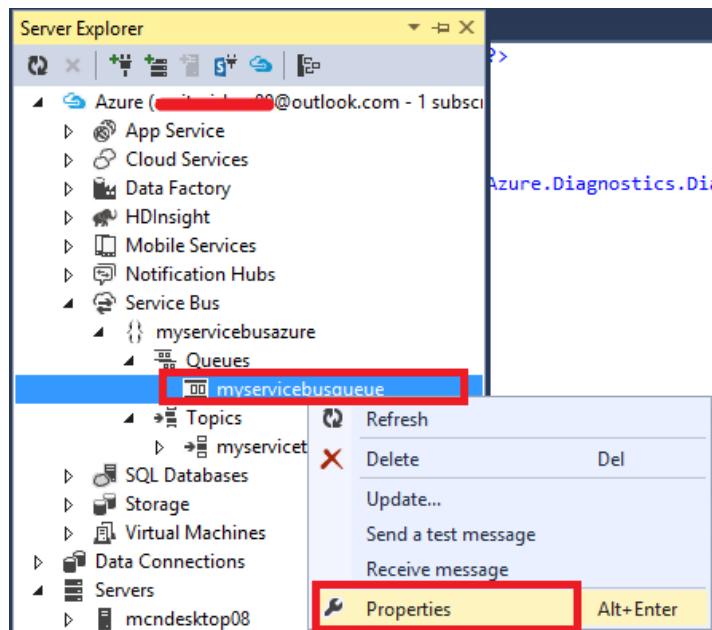
The screenshot shows the Microsoft Azure portal for the 'myservicebusazure' namespace. Under the 'TOPICS' tab, a table lists one topic: 'myservicetopic'. The table columns are: NAME, STATUS, CURRENT SIZE, MAX SIZE, SUBSCRIPTIONS COUNT, and LAST ACCESSED TIME. The 'myservicetopic' row is highlighted with a red box. At the bottom of the table, there are three buttons: 'CREATE SUBSCRIPTION', 'CONNECTION INFORMATION', and 'DELETE'.

NAME	STATUS	CURRENT SIZE	MAX SIZE	SUBSCRIPTIONS COUNT	LAST ACCESSED TIME
myservicetopic	Active	0 B	16 GB	0	Not Available

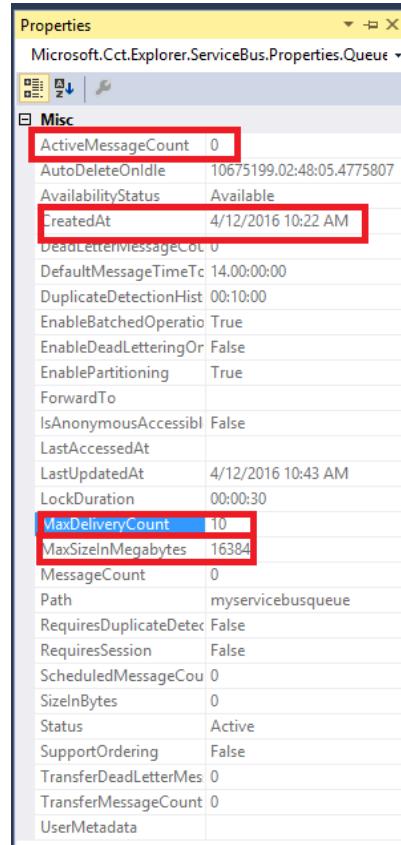
Open Visual Studio and connect with your Azure, in the left side of the window in Server Explorer. In the left hand side of Server Explorer you can see that I have Service Bus namespaces list. Under Service Bus I have one namespace that is "MyServiceBusAzure", in this I have one queue and one Topic that I have created in the Azure portal.



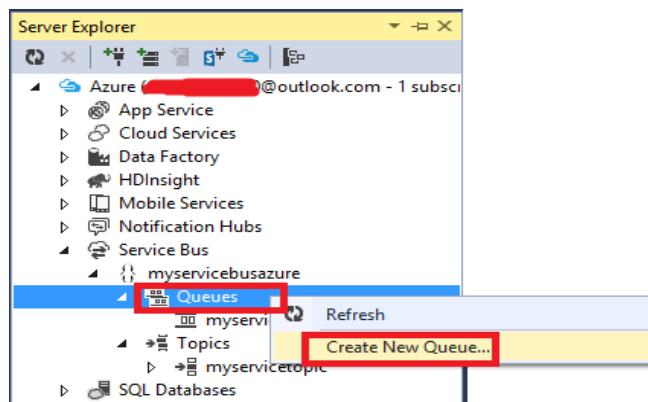
There is some additional debugging information, so right click on your queue and select properties.



In the following screenshot I have lots of information about my particular queue, like when the queue was created as well as important values like maximum delivery information and the max size of the queue.



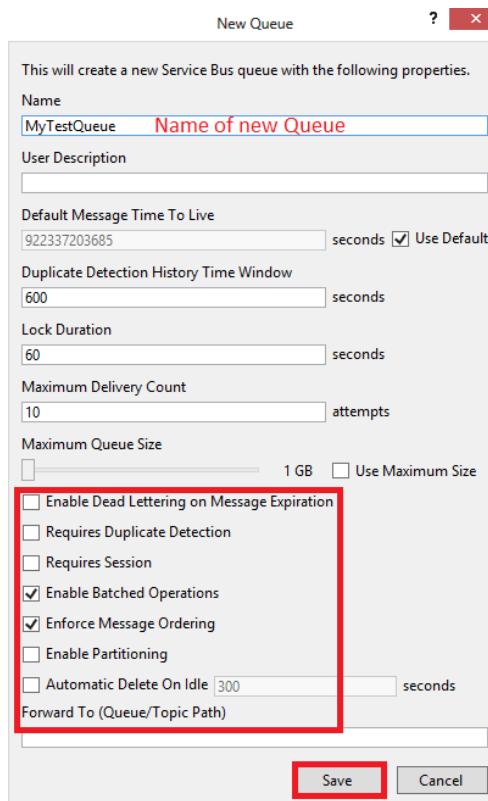
In Visual Studio I also have capability to create a new queue object, right click on to *Queues* and select “*Create New Queue*”. We also create Topics through Visual Studio like Queue.



Give the name to your Queue, as well as you can modify all the key properties of that.

Click on to “Save” button to create your queue.

How it works



So, Azure Service Bus provides messaging capabilities where we can connect about anything like application, services and devices.

I hope you enjoyed this chapter.

Top Interview Questions and Answers of Cloud Computing and Microsoft Azure

Question 1: What is Cloud Computing?

Answer: Cloud computing is internet-based computing whereby shared resources, software, and information are provided to computers and other devices on-demand, like the electricity grid.

Cloud computing is a culmination of numerous attempts at large-scale computing with seamless access to virtually limitless resources.

Cloud Computing = Software as a Service + Platform as a Service + Infrastructure as a Service.



A number of characteristics define cloud data, applications services, and infrastructure:

- **Remotely hosted:** Services or data are hosted on a remote infrastructure.
- **Ubiquitous:** Services or data are available from anywhere.
- **Commodified:** The result is a utility computing model similar to traditional utilities, like gas and electricity; you pay for what you need!

Computing can be categorized into three parts:

1. Platform as a Service (PaaS)
2. Infrastructure as a Service (IaaS)
3. Software as a Service (SaaS)

Question 2: What are the components of Cloud Computing?

Answer: Components in a cloud refer to the platforms, like front end, back end, and cloud-based delivery and the network used. All together it forms an architecture for cloud computing. With the main components like SAAS, PAAS and IAAS there are 11 more major categories in cloud computing that are:

- **Storage-as-a-Service:** This is the component where we can use or request storage. It is also called disk space on demand.
- **Database-as-a-Service:** This component acts as a live database from remote.
- **Information-as-a-Service:** Information that can be accessed remotely from anywhere is called Information-as-a-Service.
- **Process-as-a-Service:** This component combines various resources such as data and services. This happens either hosted within the same cloud computing resource or remote.
- **Application-as-a-Service:** Application-as-a-Service (also known as SAAS) is the complete application built ready for use by the client.
- **Platform-as-a-Service:** This is the component where the app is being developed and the database is being created, implemented, stored and tested.
- **Integration-as-a-Service:** Integration-as-a-Service deals with the components of an application that has been built but must be integrated with other applications.
- **Security-as-a-Service:** This is the main component many customers require. There are three-dimensional securities found in cloud platforms.
- **Management-as-a-service:** This is a component that is mainly useful for management of the clouds, like resource utilization, virtualization and server up and down time management.
- **Testing-as-a-Service:** Testing-as-a-Service refers to the testing of the applications that are hosted remotely.
- **Infrastructure-as-a-Service:** This is called as nearly as possible the taking of all the hardware, software, servers and networking that is completely virtual.

Question 3: What is the Service Model in Cloud Computing?

Answer: This is the **service model** on which you will host your application(s), for example **IaaS, PaaS or SaaS**. Each of these service models requires various levels of support and responsibilities once a system is deployed to the platform. To resolve the future challenges that you might encounter you need to understand the models carefully. Other than the service model, the second, you need to decide whether you want a **Private or On-Premise cloud**, or you want your application to be deployed on a Public Cloud.

Every service model offers you some kind of resource required to operate the most web-enabled systems imaginable.

Service Model	Short Description
1. IaaS (Infrastructure as a Service)	Allows to Set up and configure Operating Systems, Applications and it's settings.
2. PaaS (Platform as a Service)	Allows to configure Applications and it's settings.
3. SaaS (Software as a Service)	Allows to configure Application Settings only.

Question 4: What are the kind of cloud storage and why is cloud computing is so popular?

Answer: Data is centrally stored in the cloud and available across the internet. There are various storage options available or we can classify them broadly in the following three categories:

Public Cloud

In this model a service provider makes resources, such as application and storage, available to the general public over the internet. Public cloud services may be free or offered on a pay-per-use model.

Private Cloud

A private cloud provides more control over the company's data and under the control of the company's IT department.

Hybrid Cloud

A hybrid cloud is a combination of public cloud storage and private cloud storage, where some critical data resides in the enterprise's private cloud and other data is stored and accessible from a public cloud storage provider.



Benefits of a cloud:

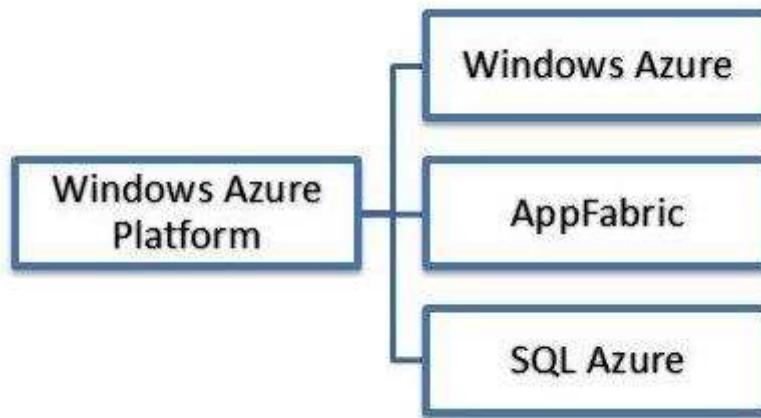
- Lower costs, “Pay as you use”
- Software updates
- Backups
- Data hosted centrally
- Scalability
- Fail over
- Monitoring services
- Data storage

There are many reasons why cloud computing is so widely popular:

- Reduction of costs
- Universal access
- Software updates
- Scalability
- Flexibility

Question 5: What is the Microsoft Azure Platform?

Answer: The Microsoft Azure Platform lays the foundation for running applications and keeping data on the cloud. It contains computer services, storage services and the fabric. Microsoft Azure affords a wide range of capabilities in the form of computing services to run applications, storage services and creating a framework that supports several applications, as well as host services and manage them centrally. This platform readily stipulates an internet infrastructure for deploying distributed applications and services since we can develop a cloud service in Visual Studio .NET and deploy it into the Azure cloud right from on-premise tools. The Azure platform is a group of three cloud technologies as in the following:



Question 6: What is Microsoft Azure and explain its services?

Answer: Microsoft Azure is created by Microsoft; it is a cloud computing infrastructure and platform. It is designed for working on the global network of Microsoft. It is for building, managing and deploying services and applications. Microsoft Azure supports various programming languages, tools, and frameworks. It can be included in both Microsoft specific and third party systems and software. It provides an infrastructure of services and platforms as services. It is also an open and flexible cloud platform that makes it easy to work on the global network. It helps us to quickly create, manage and deploy applications on the global network.

Features of Microsoft Azure:

Microsoft Azure runs and stores the data on Microsoft datacenters. There are many features that are specified here:

1. Websites allows the developers to build the sites using ASP.NET, PHP, etc and deploy these websites using FTP, Git etc.
2. SQL Database, formally known as Azure database creates, extends and scales the application into the cloud using Microsoft SQL Server.
3. This is Microsoft's platform as a service that supports the Multi-tier applications and automated deployment.

Microsoft Azure Services:

There are various types of services that are provided by Microsoft Azure:

- Web Sites
- Cloud services
- Virtual Machine
- Data Management

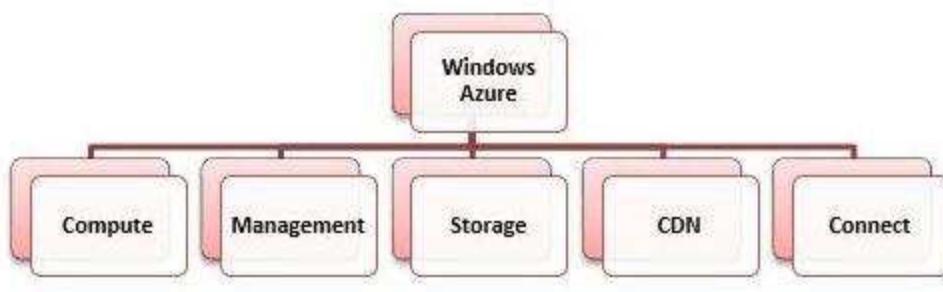
SQL Database.

Tables.

- Business Analytics

SQL Reporting.

Data Marketplace.



Question 7: What are the roles available in Microsoft Azure?

Answer: Roles are an important concept in Microsoft Azure and learning them is the base for further programming. There are mainly three roles in Microsoft Azure.

- Web Role
- Worker Role
- VM Role

Web Role:

It provides a web front-end solution. This is similar to an ASP.NET application. While under hosting Azure provides IIS and required services.

Worker Role:

It provides a background service solution. This can be thought as a windows service application. We can use this role to run background operations like database management tasks, report generation, etc. It can run lengthy operations.

According to MSDN, the Web Role can be considered as a Worker role loaded with IIS. Similarly the worker role can be used to host other application platforms.

VM Role (Virtual Machine Role):

The Web Role and Worker Role are executed on virtual machines. The Virtual Machine Roles provides the user the ability to customize the virtual machine on which the web and worker roles are running. The VM role runs a virtual hard disk (VHD) image which can be created and uploaded by the user. Through the VM role the customers can run scheduled tasks and other windows services.



Question 8: What is Microsoft Azure Portal and also explain the Azure Fabric?

Answer: Microsoft Azure Portal: To run an application, a developer accesses the Microsoft Azure portal through her Web browser, signing in with a Windows Live ID. She then chooses whether to create a hosting account for running applications, a storage account for storing data, or both.

Once the developer has a hosting account, the developer can use a Microsoft Azure portal to submit applications to Microsoft Azure.

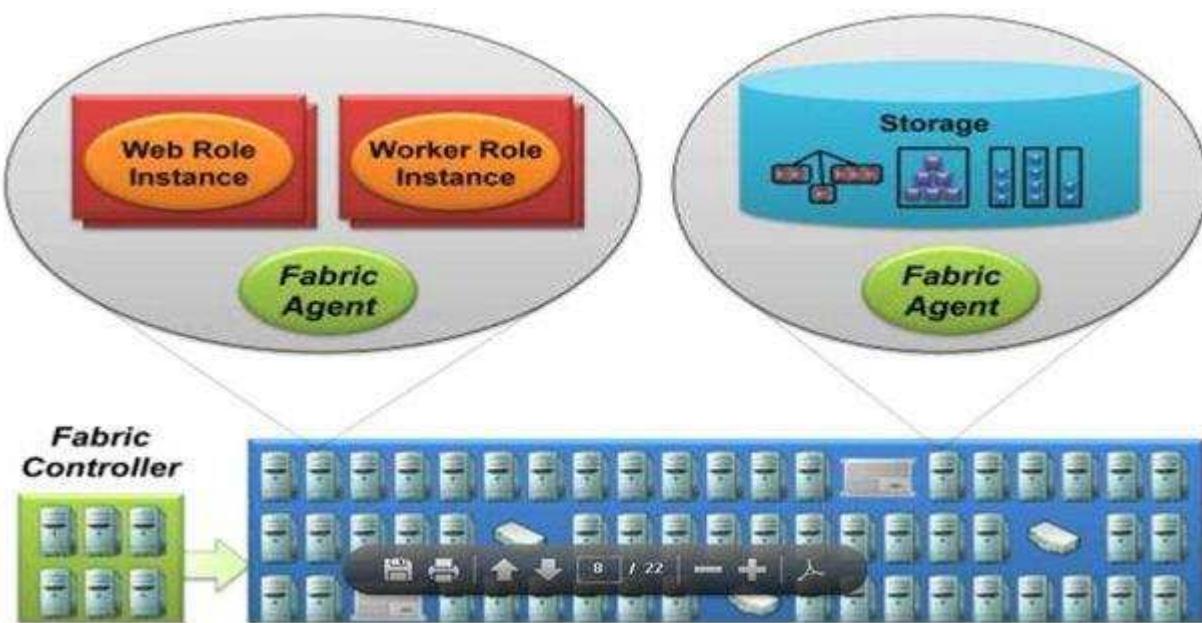
When the user sends a request to an application hosted on Azure (that can be passed by protocols like HTTP, HTTPS or TCP), the request will be received from the load balancer. The Load balancer balances the load across all the instances of the role (Web, Worker, and VM).

Azure Fabric:

The Azure fabric is the main core concept over here. It provides a service called the Azure Fabric Controller. It is called as OS for the Azure. Because it handles/manages:

1. All roles (computing) and resources.
2. Deployment and activating services.
3. Health monitoring for all services.
4. Allocating, releasing of resources.
5. Provisioning VM, terminating etc.
6. Updating patches for installed OS on VM automatically.

So there would be better to have two instances of roles and also no need to worry about software updates for user.



Question 9: What are the three main components of Microsoft Azure Platform?

Answer: Microsoft Azure provides platform and infrastructure by providing a scalable and cost-effective computing, storage, and networking resources on demand.

Microsoft Azure has three main components in Azure: **Compute, Storage and Fabric.**

1. Microsoft Azure Compute

Microsoft Azure provides a hosting environment for managed code. It provides computation service through roles. Microsoft Azure supports 3 types of roles:

- Web roles used for web application programming and supported by IIS7.
- Worker roles used for background processing of web roles.
- Virtual Machine (VM) roles used for migrating windows server applications to Microsoft Azure in an easy way.

2. Microsoft Azure Storage

Microsoft Azure provides storage in cloud. It provides 4 types of storage services:

- **Queues** for messaging between web roles and worker roles.
- **Tables** for storing structural data.
- **BLOBs** (Binary Large Objects) to store text, files or large data.
- **Microsoft Azure Drives (VHD)** to mount a page blob. These can be uploaded and downloaded via blobs.

3. Microsoft Azure AppFabric

AppFabric provides infrastructure services for developing, deploying and managing Microsoft Azure application. It provides 5 services:

- Service bus
- Access
- Caching
- Integration
- Composite

Question 10: What are the differences between a public cloud and a private cloud?

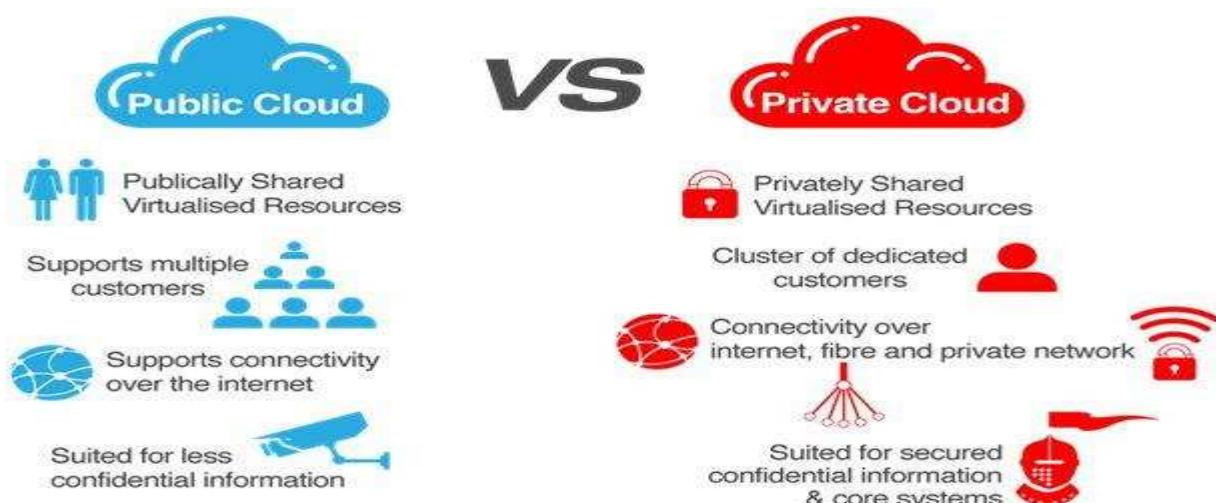
Answer: Private clouds are those that are built exclusively for an individual enterprise. They allow the firm to host applications in the cloud, while addressing concerns regarding data security and control that is often lacking in a public cloud environment. It is also known as an internal or enterprise cloud and resides on the company's intranet or hosted data center where all of your data is protected behind a firewall.

Public Cloud

- Pay for whatever resource you need at whatever time period.
- These are provided commercially.
- Supports heavy workloads without disturbing any functionality.
- It is very cheap for the consumers, since the hardware, application and other costs are handled by the providers.
- There is no wasted resource because consumers are charged for what they use.
- Scalability is always met here.

Private Cloud

- It is owned by a specific private group for their own use of employed, partners and their own customers.
- Highly controlled and not accessible by anyone other than allowed.
- Security, governance and compliance is highly automated.
- Similarly, the features are like a Public Cloud irrespective of security and maintenance.
- The cost is very high.



Question 11: What do you understand about Hybrid Cloud? Explain in detail.

Answer

A hybrid cloud is a mixture of internal and external cloud services, a combination of a private cloud combined with the use of public cloud services. This type of cloud is most suitable when you want to keep the confidential data at your premise (private cloud) and consume the other services from a public cloud.

Pros of Hybrid Cloud

- **Scalability:**

Usually the Private Cloud services will have a lesser scalability due to its security, cost and compliance whereas the Public Cloud has a high scalability and moving non-sensitive data from the private to the public will free up resources in the data centers in the Private Cloud and that increases a very high scalability for a Hybrid Cloud.

- **Cost effective:**

Similarly the Public Cloud is very cost effective rather than Private Cloud and here the Hybrid Cloud provides cost effectiveness with the data and other sensitive operations secured.

- **Security:**

Since there is a Private Cloud used; the data and sensitive operations are secured highly in the Hybrid Cloud.

- **Flexibility:**

We can easily move out the non-sensitive data and manage large scalability using a Public Cloud service along with the Private Cloud. So with the availability of a large scalability using Public Cloud and security using Private Cloud an enterprise has a vast opportunity in developing for new needs.

Cons of Hybrid Cloud

- Infrastructure dependency
- Networking
- Security compliance

Question 12: What is Diagnostics in Microsoft Azure?

Answer

Microsoft Azure diagnostics provides facility to store diagnostics data. Some diagnostics data is stored in a table, while some is stored in a blob. For collecting diagnostics data, we must initialize the Microsoft Azure diagnostic monitor. The Microsoft Azure diagnostic monitor runs in Microsoft Azure and in the computer's emulator and collects diagnostic data for a role instance.

Following diagnostics data is stored in table storage:

Data Source	Table name in Azure Storage	Detail
Windows Azure Logs	WADLogsTable	These are the application logs dumped from the application.
Windows Azure Diagnostics Infrastructure Logs	WADDiagnosticInfrastructureLogsTable	These are the logs about running of diagnostics service.
Windows Event logs	WADWindowsEventLogsTable	These are the logs generated on the instance where logs are running.
Performance counters	WADPerformanceCountersTable	These are the performance matrices like memory utilization, processor utilization, response time etc.

Following diagnostics data is stored in blob storage:

Data Source	Container name in Azure storage	Detail
IIS Logs	wad-iis-logfiles	These are the IIS logs generated by role instances.
Failed Request Logs	wad-iis-failedreqlogfiles	These are the IIS failed requests logs generated by role instances.
Crash Dumps	wad-crash-dumps	These are the logs generated on the application crash.

Question 13: What is Azure Queues?

Answer

The main reason for using queues is to provide loose connectivity among various components. For example, we have two components of an application to exchange data. Here one of them is on-premise and one exists in the cloud. Here if we use a web service to exchange data we have the following issues:

- Both components should be online simultaneously; if one partner is down, then the communication will not work.
- It's difficult to scale up if more work is present.

In Azure Queues you have a queue as mediator that connects the two components, so when the receiver is down, the sender can still insert messages into the queue, once a receiver comes online it can receive message from that queue. And for scale up we just need to add more receivers and your queue is processed in parallel.

The following are the drawbacks of using a queue:

- Queues has some charges, however these are minimal.
- Your queue is doing mediator work here. If your queue is down, devices will not able to communicate with each other.

Question 14: What is the difference between Microsoft Azure Queues and Microsoft Azure Service Bus Queues?

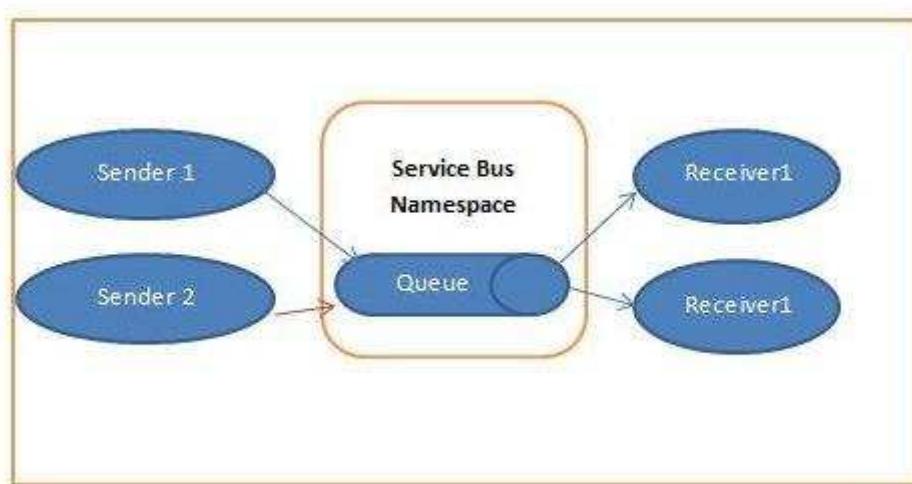
Answer

Microsoft Azure Queues:

Microsoft Azure Queue Storage is a service for storing a large number of messages that can be accessed from anywhere using HTTP or HTTPS. A single message can be up to 64KB in size. So, a queue may contain millions of messages. It is basically a part of Microsoft Azure Storage and enabled with a REST based architecture.

Service Bus Queues:

- Service Bus queues support brokered messaging communication.
- Queues provide First In, First Out (FIFO) message delivery, in other words messages are received and processed by the receivers in the order they were added to the queue.
- Each message is received and processed by only one message receiver.



Question 15: How can you create a Queue in storage account?

Answer

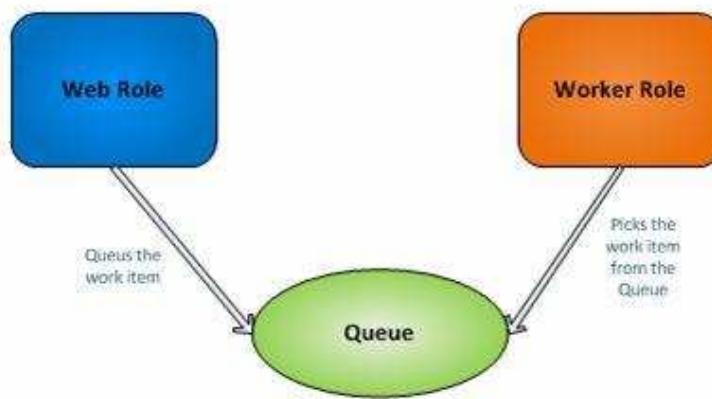
Queue is a one type of Azure Storage, where you can store your data as storage. *Blobs are stored in container, Entity in table and Message in Queue.*

Following are the key concepts in queue.

- FIFO implementation
- Messages are added to end of the Queue and processed from the front
- Queues provides a good way of Front end and Back end decoupling

In the real world example the user can queue a job through the web role (front end) and the job can be processed by a worker role (back end). This gives an opportunity to decouple the web role and worker role.

Decoupling using Queue



1. Place a `new` label control on the aspx page and add the following code **in** the page load event.
2. `protected void Page_Load(object sender, EventArgs e)`
3. {
4. `StorageCredentialsAccountAndKey accountAndKey = newStorageCredentialsAccountAndKey("account", "key");`
5. `CloudStorageAccount account = new CloudStorageAccount(accountAndKey, true);`
6. `CloudQueueClient client = account.CreateCloudQueueClient();`
7. `CloudQueue queue = client.GetQueueReference("workitems");`
8. `queue.CreateIfNotExist();`
9. `CloudQueueMessage message = new CloudQueueMessage("Test Work Item");`
10. `queue.AddMessage(message);`
11. `// Populate the messages`
12. `message = queue.GetMessage();`
13. `if (message != null) Label1.Text = "Message in Queue: " + messageAsString;`
14. }

Question 16: What is a Storage keys?

Answer

Storage keys or Access Keys are used as an authentication mode for accessing the storage services account to manipulate information based on our requirements. In Microsoft Azure we have an option to provide a Primary Access Key and a Secondary Access Key, even though we will use a single access key to authenticate our application to the storage. The main reason to provide the secondary access key is to avoid downtime to the application. If we need to change the application access key by regenerating the access key it takes quite some time to take effect; this provides a downtime.

To avoid this type of situation, a secondary access key is provided so that if the primary needs to be changed or regenerated we can map the secondary temporarily to the storage and regenerate the primary. Let us see step by step of how to get the Access Keys using Microsoft Azure Management Portal.

Question 17: What is the concept of the table in Microsoft Azure?

Answer

Table is a one type of Azure Storage, where you can store your data as storage. Blobs are stored in container and Entity in table.

Following are the key concepts in table.

- Tables allow structure data storage
- There can be 0..n tables in a storage account
- Table store data as a collection of entities
- Entity have a primary key and properties as key value pair

Question 18: How to send messages to a Queue and how to receive messages from a Queue?

Answer

Send Messages to a Queue

The code below demonstrates how to create a *QueueClient* object for the "TestQueue" queue created above using the *CreateFromConnectionString* API call:

```
1. string connectionString =
2. CloudConfigurationManager.GetSetting("Microsoft.ServiceBus.ConnectionString");
3. QueueClient Client =
4. QueueClient.CreateFromConnectionString(connectionString, "TestQueue");
5. Client.Send(new BrokeredMessage());
```

Messages sent to (and received from) Service Bus queues are instances of the *BrokeredMessage* class. *BrokeredMessage* objects have a set of standard properties (such as Label and TimeToLive), a dictionary that is used to hold custom application specific properties, and a body of arbitrary application data. An application can set the body of the message by passing any serializable object into the constructor of the *BrokeredMessage*, and the appropriate *DataContractSerializer* will then be used to serialize the object. Alternatively, a *System.IO.Stream* can be provided.

Receive Messages from a Queue

The easiest way to receive messages from a queue is to use a *QueueClient* object. These objects can work in two different modes: *ReceiveAndDelete* and *PeekLock*.

When using the *ReceiveAndDelete* mode, reception is a single-shot operation; that is, when the Service Bus receives a read request for a message in a queue, it marks the message as consumed, and returns it to the application.

The *ReceiveAndDelete* mode is the simplest model and works best for scenarios in which an application can tolerate not processing a message in the event of a failure. To understand this, consider a scenario in which the consumer issues the receive request and then crashes before processing it. Because the Service Bus will have marked the message as being consumed, when the application restarts and begins consuming messages again, it will have missed the message that was consumed prior to the crash.

This example creates an infinite loop and processes messages as they arrive in the "TestQueue":

```
1. Client.Receive();
2. // Continuously process messages sent to the "TestQueue"
3. while (true)
4. {
5.     BrokeredMessage message = Client.Receive();
6.     if (message != null)
7.     {
8.         try
9.         {
10.             Console.WriteLine("Body: " + message.GetBody < string > ());
11.             Console.WriteLine("MessageID: " + message.MessageId);
12.             Console.WriteLine("Test Property: " + message.Properties["TestProperty"]);
13.
14.             // Remove message from queue
15.             message.Complete();
16.         }
17.         catch (Exception)
18.         {
19.             // Indicate a problem, unlock message in queue
20.             message.Abandon();
21.         }
22.     }
23. }
```

Question 19: What are the Storages in Microsoft Azure?

Answer

Microsoft Azure data storage enables users to store, access, analyze and protect their data while making it available from anywhere and at any time. Microsoft Azure provides various services from storing data in SQL databases in the cloud for analysis and reporting to meet the needs of your business. Microsoft Azure ensures data security with high throughput of application data in the cloud. Microsoft Azure typically offers the following three types of storage in the cloud atmosphere.

- **BLOB:** BLOBs offer a mechanism for storing large amounts of text or binary data, such as images, audio and visual files. It can scale up to 200 terabytes and can be accessed using REST APIs. We can move BLOB data as a single volume between private and public clouds using Windows Azure Drive.
- **Table:** Tables represent storage locations across machines for data that reside in the form of entities and properties on the cloud. Tables store large amounts of unstructured data that can be accessed either using REST APIs from within a service running in Microsoft Azure or directly over the Internet using HTTP/HTTPS.
- **Queue:** The sole objective of a Queue is to enable communication between Web and Worker Role instances. They aid in storing messages that may be accessed by a client. Web Role instances can initiate user requests that need to be processed in the background. On the other side a Worker Role observes the queue to process the request and respond back via queue to the Web Role instance.

Question 20: What is Federation in SQL Azure?

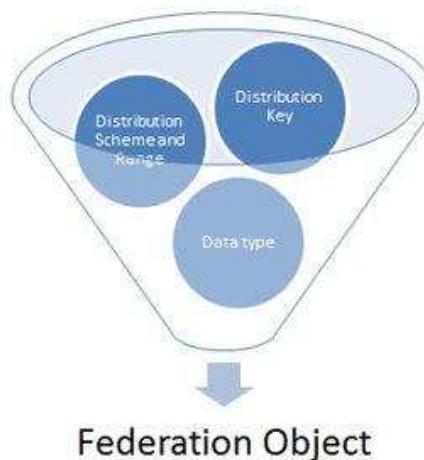
Answer

Federation is introduced in SQL Azure for scalability. Federation helps both administrators and developers to scale data. It helps administrators by making repartitioning and redistributing of data easier. It helps developers in the routing layer and sharding of data. It helps in routing without application downtime.

Federation does basic scaling of objects in a SQL Azure Database. Federations are the partitioned data. There can be multiple Federations within a database. And each Federation represents a different distribution scheme.

We create a Federation with a different distribution scheme and requirement. Student and Grades tables of a School Database may have a different distribution requirement so they are put into different Federations.

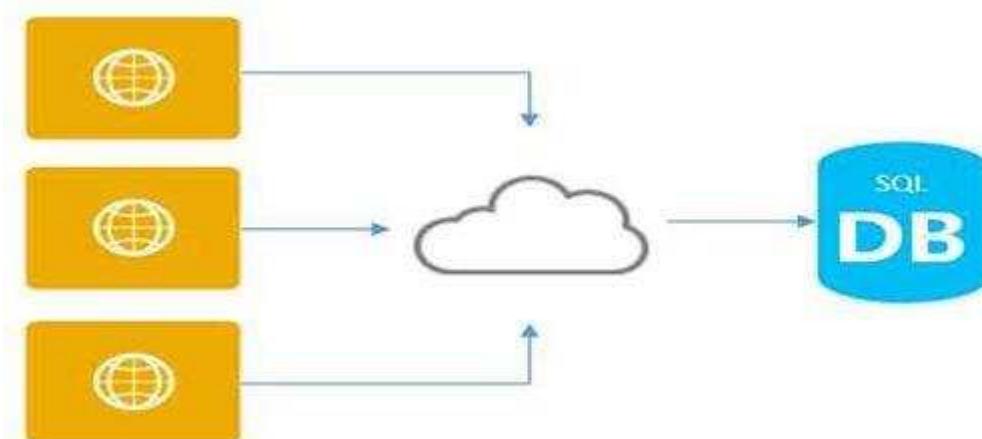
Each Federation object scales out data to many system managed nodes. A Federation object contains:



Question 21: What is SQL Azure Database?

Answer

SQL Azure database is simply a way to get connected in Cloud Services where we can store our database into Cloud. Microsoft SQL Services and Microsoft SQL Data Services are now known as Microsoft SQL Azure and SQL Azure Database. Microsoft Azure is the best way to use **PAAS (Platform as a Service)** where we can host multiple databases on the same Account.



Microsoft SQL Azure has the same feature of SQL Server, i.e. high availability, scalability and security in the core.

Microsoft Azure SQL Database have a feature, it automatically creates backups of every active database. Every hour a backup is taken and geo-replicated to enable the 1 hour recovery point objective (RPO) for Geo-Restore. Additionally, transaction log backups are taken every 5 minutes to enable Point in Time Restore.

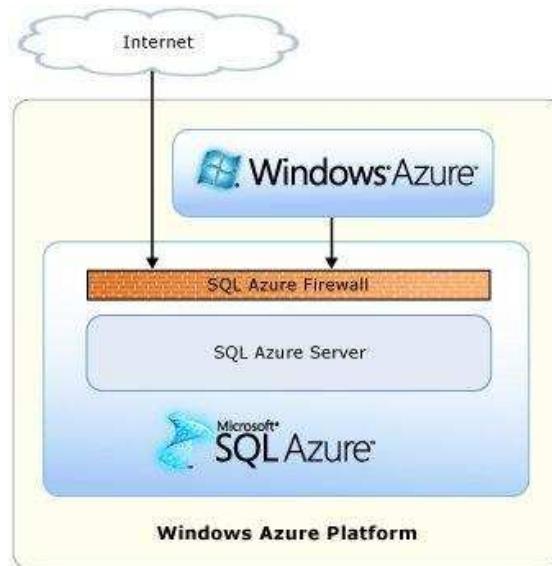
Question 22: What are SQL Azure firewall rules?

Answer

SQL Azure firewall rules are provided to protect the data and to prevent access restrictions to the SQL Azure database. Firewall grants access to the originating IP's from which we are trying to access the database. In order to configure the firewall we need to configure a range of acceptable IP addresses upon which we try to connect to the SQL Azure server using the Management Portal or with the SQL Server Management Studio. Let us see the step by step process of adding and deleting rules as per our IP configurations.

All access to SQL Azure is blocked by firewall.

By default Database created in SQL Azure is blocked by firewall for security reasons. Any attempt from external access or access from any Azure application is blocked by firewall.



Question 23: What is Microsoft Azure Traffic Manager? What are its benefits?

Answer

Traffic Manager Name is itself self-explanatory. It allows users to control the distribution of user traffic of deployed Azure cloud services, Azure websites or any other endpoint. In this the distribution of traffic includes Azure cloud services, Azure web sites and other endpoints. There are 3 different load balancing methods provided by Azure. The Traffic Manager works by applying an intelligent routing policy engine to the Domain Name Service (DNS) queries on your domain names and maps the DNS routes to the appropriate instances of your applications.

Benefits of Azure Traffic Manager

- **Increase Performance:** Can increase performance of your application that includes faster page loading and better user experience. This applies to the serving of users with the hosted service closest to them.
- **High Availability:** You can use the Traffic Manager to improve application availability by enabling automatic customer traffic fail-over scenarios in the event of issues with one of your application instances.
- **No Downtime Required for Upgrade / Maintenance:** Once you have configured the Traffic Manager you don't need downtime for application maintenance, patch purgation or complete new package deployment.
- **Easy to configure (Quick Setup):** It's very easy to configure Azure Traffic Manager on Widows Azure portal. If you have already hosted your application on Microsoft Azure (a cloud service, Azure website) you can easily configure this Traffic Manager with a simple procedure (setting routing policy).

Question 24: What are the data synchronize services in Azure?

Answer

Sync Services is provided by Microsoft Azure where you can keep a SQL Azure database synchronized with another database. You can take a backup of a database from one region to another region. There is a simple procedure you need to follow to configure this service. If not a complete database then you can at least keep selected tables or selected rows of tables synchronized.

This service provides the following two ways to synchronize:

1. Synchronize a SQL Azure database from one SQL Azure server to another SQL Azure server.
2. Synchronize a SQL Azure Database from an Azure Server to a Local Server database.

Question 25: How can you manage SQL Azure security?

Answer

SQL Azure also has a security management system very similar to the SQL on-premises versions. It basically consists of the following:

1. **Logins:** Server level.
2. **Users:** Database level, mapped to server logins.
3. **Schemas:** Database level, authorized/owned by a user or another schema.
4. **Roles:** Database level, authorized/owned by a user or another role.
5. **Permissions:** Database level, permission like SELECT, DELETE, ALTER and so on for objects/schemas granted to users/roles.

When you create a server in SQL Azure, it asks you to create a login at the same time. That login acts as the administrative login that has access to all the databases in that server. However, you might want to create other logins with less privilege. As of now, the SQL Azure portal doesn't have any UI to create these extra logins. So you'll need to resort to running T-SQL statements.

Note: All of the following procedures are done using the administrative login mentioned above.

Creating Logins

Login to the master database and run the following T-SQL statement.

1. **CREATE LOGIN MyServerLogin WITH password='My#Password123'**

This statement creates a login ID in the server. This is a normal login ID that doesn't have access to any of the databases in that server. So if you try to login to the Azure server with this account (either from SQL Server Management Studio 2008 R2 - SSMS or from Azure Portal) you would get an error saying this user doesn't have access to the master database. So the next step is to map this login to the required databases, not necessarily to the master db.

Question 26: What are the different types of databases in SQL Azure?

Answer

In the SQL Azure there are two types of databases:

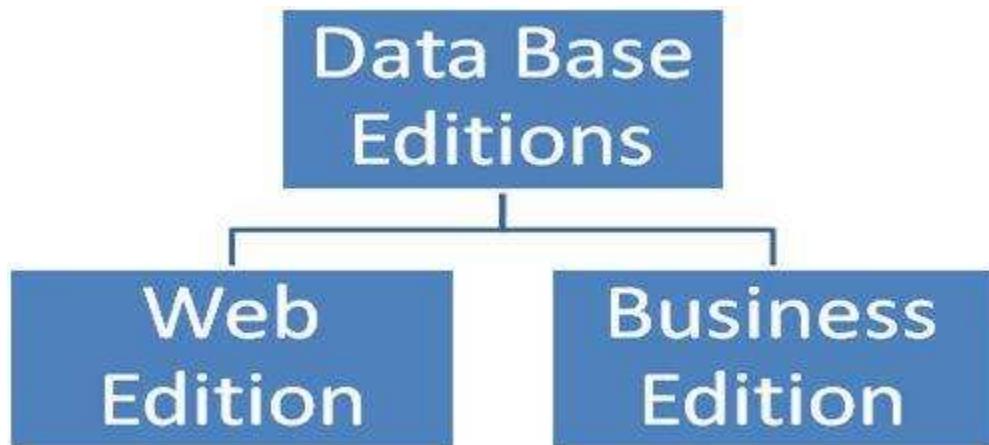
- Web Edition
- Business Edition

The Web Edition Relational Database includes:

- Up to 5 GB of a T-SQL based relational database*
- Self-managed DB, auto high availability and fault tolerance
- Supported by existing tools like Visual Studio, SSMS, SSIS, BCP
- Best suited for Web application, Departmental custom applications.

Business Edition DB includes:

- Up to 50 GB of T-SQL based relational database*
- Self-managed DB, auto high availability and fault tolerance
- Additional features in the future like auto-partition, CLR, fanouts etc
- Supported by existing tools like Visual Studio, SSMS, SSIS, BCP
- Best suited for SaaS ISV applications, custom Web application, Departmental applications.



Question 27: What is TFS build system in Azure?

Answer

A Build is nothing but the output of a solution. In the case of Azure projects, you generally get the file with a **.cspkg** extension that means a Cloud Service Package is used for the deployment of your cloud services.

Build Servers

In layman's terms a build server acts as the machine where you put your deployment packages.

To use Team Foundation Build, you must have at least one build machine. This machine can be a physical machine or a virtual machine.

Build Controllers

Build Controllers are the element in the build system that accepts the build requests from any project inside the team project collection. Each build controller is dedicated to a single-team project collection. So there is a one-to-one relationship between a team project collection and a build controller.

Build Agents

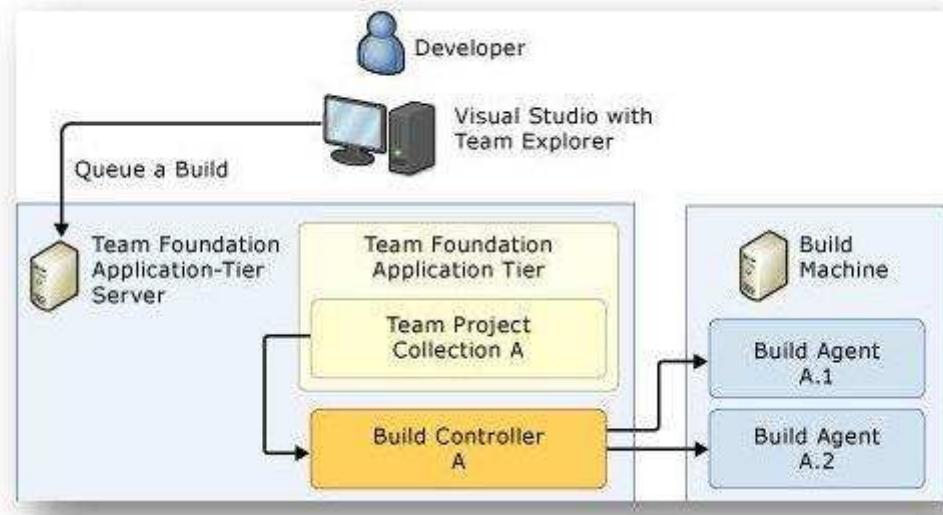
Build agents are elements in the build system that does more processor-intensive work.

Build Definitions

A Build definition is nothing but the process that the build controller uses to deploy your packages to a target site. Using Team Explorer you should be able to create a new build definition where you will need to associate the build controller and define process as well as trigger a point for the definition.

Once the build definition is in place, you will be able to queue new builds using it.

Typical Build System Topology



Question 28: What is the difference between IaaS and PaaS in Cloud?

Answer

Infrastructure as a Service (IaaS): is in effect much like building a set of services and virtual machines and networks in a Cloud as the user would On-Premises.

Platform as a Service (PaaS): A Platform is a cloud that gives the user various frameworks, tools and services on the cloud and the user are building the application in the cloud leveraging these various tools, services and components within the application.

	IaaS	PaaS
Granularity of Control of Virtual Machines	The user can remote into virtual machine running in the cloud using IDP. The user has the same level of control as the user would with a virtual machine on-premises. This means the user can get into the registry, install custom software the user want to install on that virtual machine.	The PaaS provider takes care of running the application. It means that the user doesn't get access to registry and neither can the user install custom software necessarily in that application or in the location where it runs. PaaS providers provide the infrastructure and the platforms.
Control over the environment configuration	The user can choose a certain number of virtual cores, virtual CPUs, RAM, networking set up, how many different networks, virtual VPNs, separate subnets, extensions to on-premises networks and load-balancing across multiple VMs.	PaaS can be thought of as the next step of IaaS where the configurations is also done for the user by the provider.
Maintenance	The user is responsible for the O/S, patching, firewall management, security, data, runtime, applications, middleware and everything that's running inside of the VM because the user is responsible for that virtual machine. The only difference is the location of the VM; instead of on-premise it is in the cloud. The user still gets to do all the configurations but the user is relieved of doing the physical hardware things.	The provider is responsible for application/services on PaaS. The user is just managing the various services.
Scalability	The user is responsible for scalability.	Scalability is the responsibility of the PaaS provider.
Pricing	Subscription Model.	Subscription Model, but it includes the cost of hardware for the installation of PaaS.

Both IaaS and PaaS help organizations to minimize operational costs and increase their productivity as they get faster time to market and require no up-front investments.

Question 29: What is the Azure App Service? What are the advantages of App Service over Mobile Service?

Answer

Azure App Service is a fully managed Platform as a Service (PaaS) offering for professional developers that brings a rich set of capabilities to web, mobile and integration scenarios. Mobile Apps in Azure App Service offer a highly scalable, globally available mobile application development platform for Enterprise Developers and System Integrators that brings a rich set of capabilities to mobile developers.

Advantages of App Service:

- Simpler, easier and more cost effective offering for apps that include both web and mobile clients.
- New host features including Web Jobs, custom CNames, better monitoring.
- Turnkey integration with Office 365, Dynamics CRM, Salesforce, and other vital SaaS APIs.
- Support for Java and PHP backend code, in addition to Node.js and .NET.
- Turnkey integration with Traffic Manager.
- Connectivity to your on-premise resources and VPNs using VNet in addition to Hybrid Connections.
- Monitoring and troubleshooting for your app using NewRelic or AppInsights, as well as alerts.
- Richer spectrum of the underlying compute resources, e.g. VM sizes.
- Built-in auto scale, load balancing, and performance monitoring.
- Built-in staging, backup, roll-back, and testing-in-production capabilities.

Question 30: How can you demonstrate between Azure mobile service and Web API?

Answer:

Web API: Web API is used only for that client who uses HTTP enabled services. There is a problem with Web API, if you want to create a Web API service, then you need knowledge of ASP.NET solution and you need to be familiar with .Net.

There are more benefits of Web API over Node.js. Web API uses HTTP request and process

them using the HTTP verbs. Actually, Web API is fully REST service. As per my thinking Web API is better approach to create APIs.

Azure Mobile Services: Microsoft Azure Mobile Service is a tool that can integrate with the Microsoft Azure cloud. This tool will help you to connect a cloud backend to your Windows 8 application. The goal of the Microsoft Azure Mobile Service is to make Microsoft Azure the obvious choice for the Windows 8 Metro style application developer. Currently we can only use the Microsoft Azure Mobile Service in Metro style applications but in the near future the mobile service can also work with iOS, Android and Windows Phone.

Mobile services can only be run on demand or you can schedule the service when it will start and when it will stop.

To use Microsoft Azure Mobile Services, you need to use any Source Control where you can take the backup.

Question 31: What is Google Cloud Platform?

Answer

Google Cloud Platform is a new way to use cloud computing platform in real time. It is developed by Google and offers every developer the chance to host their Web API's and Cloud Storage and many more things on the same supporting infrastructure that is used by Google itself internally in their products, such as Google Search, Google Play Store, Gmail and YouTube. Also for every product by Google, Google Cloud Platform provides developer products to build application and a range of programs from simple websites to complex applications. Google use Load Balancer technique that allows distributing cloud data into multiple data centers all over the world. With the help of Google Cloud **BigQuery** we can execute billions of queries per second.

Google Cloud Platform is specially designed for enterprise solutions from Google for Work and provides a set of modular cloud-based services with a host of development tools, test, and deploy applications on Google's highly-scalable and reliable infrastructure for your web, mobile and backend solutions. For example:

- Hosting and computing
 - App Engine.
 - Compute Engine.
- Cloud storage

- o Cloud Storage.
- o Cloud Store.
- o Cloud SQL
- BigData
 - o BigQuery
- Services
 - o Cloud Endpoints.
 - o Translate API.
 - o Prediction API.

Question 32: What is the VM (Role) and VM (Pass)?

Answer

VM Role: VM role is a kind of role in the Azure platform which helps to maintain service packs, patches, updates and applications already installed, to Microsoft Azure automatically.

We can use a VM Role in the case of:

1. Long-Running Setup: If the application requires a long-running setup.
2. Error-Prone Application: Application in which you expect more errors/risks while installing it.

VM (IaaS): These are literally VMs as we understood them from the On-Premise running in the Azure Microsoft Data Center.

Advantages of VM over VM role:

1. VM is durable.
2. IaaS VMs is that you can take just about any on-premises application, whether it makes changes to the local file system or not (even if the OS is one of the supported Linux distributions) and deploy it to the cloud for a few cents per hour.

Disadvantages:

1. Provisioning time will be increased.
2. Automatic OS/software updates will be there in the (IaaS) VM.

Question 33: What is Virtual Machine in Azure?

Answer

Google Cloud Platform Compute Engine is Google's Infrastructure-as-a-Service (IaaS), which is used to run large-scale workloads on virtual machines hosted on Google's infrastructure. It is used by Google itself for their end user products, such as Google search, YouTube and for all products of Google. Allows you to choose a VM with a specific requirement of hardware configurations with a list of server operating systems and now it also allow Windows Server.

Azure Virtual Machines could be your choice if you want frequent modifications and changes in your web server environment. It provides you rich set of features; however, correctly configuring, securing and maintaining VMs require much more time and more IT expertise compared to Azure Cloud Services and Azure Websites. You will need to make more effort on frequent maintenance and update patches to manage the VM environment.

Question 34: How to create a Virtual Machine in Azure?

Answer

Here are following steps to create a Virtual Machine in Cloud:

Step 1: Log in to your Azure management portal.

Step 2: Click New.

Step 3: Select "Compute" -> "Virtual Machine" -> "From Gallery".

Step 4: Select the Operating System that you would like to install on the VM. In this scenario we will install a Ubuntu server 13.04 because then it will be easy for me to continue with the later posts on creating a PHP app on our new VM.

Step 5: The next window will ask you about user details, VM RAM and number of cores and a name for the VM. Fill them in as you wish. I will use a password instead of a SSH key.

Step 6: Next window ask you about cloud configuration (DNS Setting) and Storage account and Region. Except for region leave the rest as it is unless you know what you are doing.

Step 7: Now we will need to create end points for us to access the VM. For now let's keep SSH access only.

Step 8: Then the VM will be created and will be running after a few minutes. You can see it in your Azure portal.

Question 35: What is Autoscaling in Azure?

Answer

Autoscaling Application Blocks can automatically scale the Microsoft Azure application based on the rules defined specifically for the application.

The Autoscaling Application Block supports two autoscaling mechanisms:

1. Instance Autoscaling, where the block changes the number of role instances based on constraint and reactive rules.
2. Throttling, where the application modifies its own behavior to change its resource utilization based on a set of reactive rules. For example switching off non-essential features, or gracefully degrading its UI.

So, there are two types of rules:

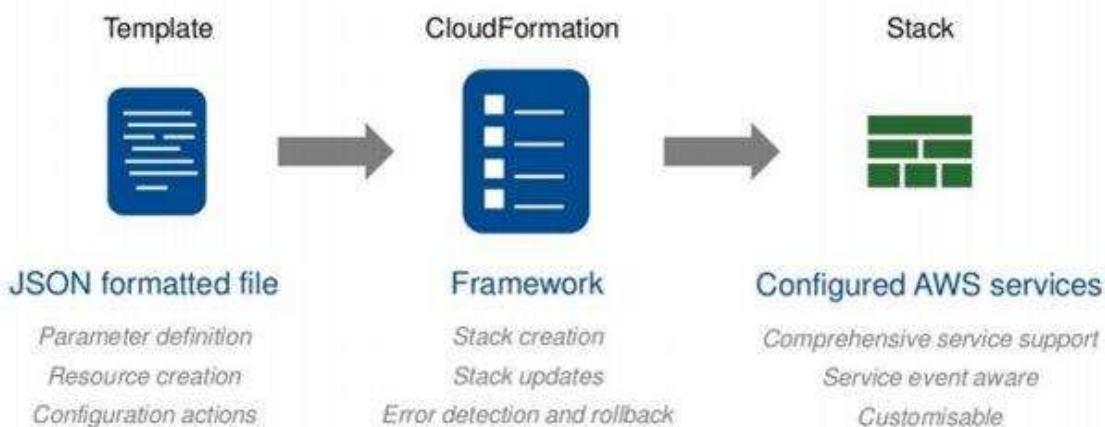
1. **Constraint rules:** Constraint rules set the upper and lower bounds on the number of instances. For example, in the evening between 6:00 and 8:00, you need a minimum of 3 instances and a maximum of 7 instances, and then use the constraint rule.
2. **Reactive rules:** Reactive rules enable the number of role instances to change in response to unpredictable changes in demand. For example, if the workload increases then increase the number of role instances by 1. The reactive rules can use a variety of techniques like performance counters, or the Microsoft Azure queue length to monitor and control the application's workload. A reactive rule makes changes to the number of role instances only if a constraint rule applies at the same time. It is easy to create a default constraint rule that always applies.

Question 36: What is AWS Cloud Formation?

Answer

AWS Cloud Formation is a cloud builder service that helps you model and set up your Amazon Web Services resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS. For this you need to create a template that describes all the AWS resources that you want, like Amazon EC2 instances that you want, Amazon VPC and subnet details and so on. AWS CloudFormation takes care of provisioning and configuring those resources for you. You don't need to individually create and configure AWS resources and figure out what's dependent on what, AWS CloudFormation handles all of that.

AWS CloudFormation is available at no additional charge. You will be billed only the normal rates for the AWS resources like EC2, RDS, LB and so on that AWS CloudFormation creates and your application and resources use.



Question 37: What is profiling in Azure?

Answer

Profiling is nothing but a process of measuring the performance analysis of an application. It is usually done to ensure that the application is stable enough and can sustain heavy traffic.

Visual Studio provides us various tools to do it by gathering the performance data from the application that also helps in the troubleshooting issues.

Once the profiling wizard is run, it establishes the performance session and collects the sampling data, then generates report files that can be opened and analyzed in Visual Studio.

The profiling reports can help us to:

- Determine the longest running methods within the application.
- Measure the execution time of each method in the call stack.
- Evaluate memory allocation.
- Analyze concurrency issues (usually for multi-threaded code).

Question 38: How can you connect an Azure Hosted website to FTP?

Answer

Here are some steps:

- **Stop:** Stopping the virtual server.
- **Restart:** It will restart the server if something gone wrong.
- **Mange Domains:** If upgrade from free account to basic or standard account you can manage your domains here.
- **Delete:** It will delete entire website.
- **WebMatrix:** To link website to WebMatrix.

If you click on website you will be taken in dashboard where different operations are available,

- **Monitor:** Graphical representation of request, cuptime, data in, data out.
- **Configure:** Change the language setting version used.
- **WebJobs:** Automatic task that are scheduled will be discussed in upcoming articles.
- **Scaling:** More than one instance of website can be created to balance load.

Linked Resources:

This will show how the resources are attached. You will see the cleardb database if you click on it will redirect you to cleardb website and you will be automatically logged into account created by Azure for you. You can view performance and other configurations of your database.

Connecting FTP

If we want to view files of our WordPress in order to change configuration setting, upload content or to make backup.

For this we have to set some deployment credentials in Microsoft Azure so that Azure verifies that we are authorized to make changes.

For this go to Management portal select website. Dashboard of app will be opened. If not, click on setup deployment credentials.

Question 39: What is Cmdlet in Azure?

Answer

A **cmdlet** is a lightweight command that is used in the Microsoft PowerShell environment. The Windows PowerShell runtime invokes these cmdlets within the context of automation scripts that are provided at the command line. The Windows PowerShell runtime also invokes them programmatically through Windows PowerShell APIs.

A cmdlet is a compiled piece of .NET code. Cmdlets handle object input and output as well as usually playing nice and well with the (object-based) pipeline. Cmdlets have no direct representation in the file system, as they are not programs or similar. They exist solely within PowerShell. You can use the **Get-Command** Cmdlet to query all available Cmdlets, functions, etc.

How to Start with Microsoft Azure Automation using Cmdlets:

- Go to Microsoft Azure home page.
- Click the Downloads link on the page (See at the very bottom of the page)
- Click the link for Command Line Tools
- The installation can take several minutes because there is a dependency on the Microsoft Azure SDK, which has its own set of dependencies.

Question 40: What is Azure Explorer?

Answer

Azure Explorer is a free Microsoft Azure storage tool; in this we can manage all our Microsoft Azure blobs at a common place or one place. Azure Storage Explorer is a useful GUI tool for inspecting and altering the data.

In Azure Explorer, we have “**Azure Storage Account**”, and in the Azure Storage Account, we can manage more than one Blob Container, in Blob Container we can create page blob as well as new folders to store our files.

Step 1: For the installation of Azure Explorer we need go to its download link, after going to above link, you will see the following window. Click “**DOWNLOAD AZURE EXPLORER**”.

Question 41: What is Service Fabric in Azure?

Answer

Azure Service Fabric is a next-generation middleware cloud platform to build scalable, reliable, and managed enterprise applications. Microsoft is already using Service Fabric powers in many of its products, including Azure SQL Database, Azure DocumentDB, Cortana, and Power BI.

- Develop massively scalable applications that are self-healing.
- Develop with a "**datacenter on your machine**" approach. The local development environment is the same code that runs in the Azure datacenters.
- Develop applications composed of microservices, executables, and other application frameworks of your choice, such as ASP.NET, Node.js, etc.
- Develop stateless and stateful (micro)services and make these highly reliable.
- Simplify the design of your application by using stateful (micro)services in place of caches and queues.
- Deploy applications in seconds.
- Deploy to Azure or to on-premises clouds running Windows Server with zero code changes. Write once and then deploy to any Service Fabric cluster.

- Deploy applications at higher density than virtual machines, deploying hundreds or thousands of applications per machine.
- Deploy different versions of the same application side by side, each independently upgradable.
- Manage the lifecycle of your stateful applications without any downtime, including breaking and nonbreaking upgrades.

Question 42: What is Microsoft Azure Scheduler?

Answer

Microsoft Azure Scheduler allow you to invoke actions – such as calling HTTP/S endpoints or posting a message to a storage queue on any schedule. With Scheduler, you create jobs in the cloud that reliably call services both inside and outside of Microsoft Azure and run those jobs on demand, on a regularly recurring schedule, or designate them for a future date.

Scheduling is the history details of your apps in the cloud; it gives information of the particular application, means its status, its start time and its end time.

Question 43: What is Blob Storage in Azure?

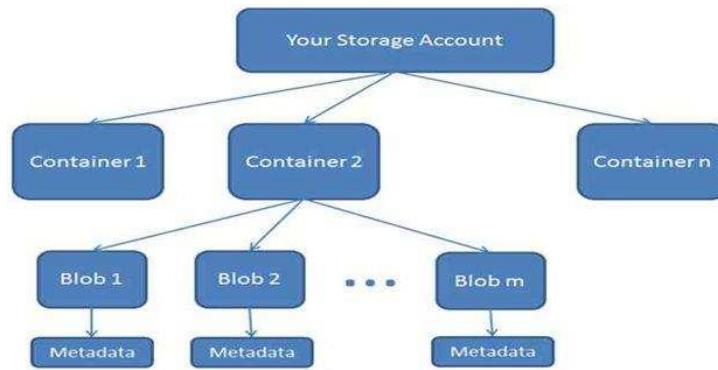
Answer

Microsoft Azure Blob Storage can store hundreds of terabytes data in one account. As you know you can create multiple blob storage accounts, so definitely you can store multiple hundreds of terabytes data with Microsoft Azure Blob Storage.

We don't require taking tension about the backup of the data which is stored on blob storage. It is because it automatically takes the backed up data. You can increase your storage as per requirement and you need to pay only which you have used and what you have used.

One more thing that you don't need to take headache about how to manage your data, your VM or any other things; actually Microsoft Azure manages everything automatically.

You can use Blob Storage as REST API and can use with any language like Java, .NET, etc. Actually technology doesn't matter with blob storage. It means all kinds of client can be accessible to this storage.



Question 44: Why do you go with Microsoft Azure?

Answer

There are many regions that people want to go with Microsoft Azure:

- **Flexibility:** Microsoft Azure supports wide range of Operating System that makes it reachable to wide range of customers. It also supports different kinds of programming language like PHP, .NET, JAVA, etc. You can use different kinds of frameworks and tools for building your Apps. So, basically it provides us flexibility to choose existing one which will help us to create variety of applications with Microsoft Azure.
- **Extendable:** Microsoft Azure easily integrates with your existing environment using largest network with secure private connectivity. It also provides database, storage which is extendable on demand. Microsoft Azure can run with your data center.
- **Scalable:** You can scale up and down your services as per your demand. Services, storage and performance also will be same if you scale up or down. So, small client can take benefit of Microsoft Azure and their services.
- **Protection:** Everyone is worried about their data. They need protection for their data; nobody can rely on some vendor for their data. Microsoft Azure has launched Azure Government for believing you that your data is secure with Microsoft Azure. It commits for securing, protecting and privacy of your data.
- **Trusted:** Today Microsoft Azure is used by top clients worldwide. They are using Microsoft Azure Services for a long time and they believe in it.

Question 45: What is Azure Mobile Service?

Answer

Microsoft Azure Mobile Service gives you the power to create a cloud service mobile application. This will make your work more flexible and gives your application more portability in an efficient way. This is a short introduction of Azure Mobile service. You will get to know, how to create and integrate new mobile service in your new or existing application. So let's get crack in Azure Mobile Service with Universal Windows Platform.

Question 46: What is the difference between Table, Queue, and Blob storage?

Answer

To use Azure Storage; you just need to have an Azure Storage Account. Using this account you can access the service of the Azure Storage. Actually Azure Storage provides two types of the storage account.

1. Standard Storage Account: It is for Blob, Table, and Queue Storage.
2. Premium Storage Account: It is for Azure Virtual Machine Disks Only.

BLOB STORAGE: It can store any type of data; data can be a document file, media file, exe or dll, images, text file or any other data file. So, basically it is used for storing the objects. So, as the definition says, we can store large amount of the unstructured data into the blob storage.

TABLE STORAGE: It is like a table inside the SQL Server. So, it is used to store the structured data. Table storage is a NoSQL key-attribute data store. It means every data stored into the table is stored with typed property name. It is very fast when you are working with large number of data. It doesn't use schema for developing the table.

QUEUE STORAGE: Sometimes, it is required to transfer the data in the format of the message over the internet between the two cloud services. In that scenario we prefer to use the queue storage. You can create large number of the queues to process your data. Every queue contains large number of the message and every message size may be up to 64 KB.

Question 47: What is the Migration Assistant tool in Azure Websites?

Answer

The migration Assistant tool will help to analyze your IIS installation and identify which sites can be migrated to the cloud, highlighting any elements which cannot be migrated or are unsupported on the platform.

Once analyzed this tool will also create websites and databases provided under given Azure subscription.

Automated Assessment and Migration

- This tool will provide a high level readiness assessment.
- The report outlines sites which are ready to move, elements which may need changes and highlights unsupported features.
- The detailed report offers expert guidance and advice tailored to your environment.
- The tool creates any Website(s) and associated database(s), if applicable, automatically and synchronizes your content.

Question 48: What is Azure Mobile Engagement?

Answer

Azure Mobile Engagement is a SaaS-delivered, data-driven user engagement platform that enables real-time fine-grain user segmentation, app user analytics, and contextually-aware smart push notifications and in-app messaging across all connected devices.

With Azure Mobile Engagement, Application publishers and marketing professionals can create cutting edge interactions like the following,

- Real Time actionable analytics to increase app usage.
- Push Notification and Communication Platform.
- Open API's and Ease of integration.
- Data Protection & Privacy across globe.

Azure Mobile Engagement is available across all major mobile platforms including Android, IOS and Windows.

Question 49: What is WordPress in Microsoft Azure?

Answer

WordPress is Web software you can use to create a Web sites, blog, or apps. The core software is built by hundreds of community volunteers, and when you're ready for more, there are thousands of plugins and themes available to transform your site into almost anything you can imagine.

We can install WordPress on our local computers to develop Web sites, blogs, or theme plugins; or we can install it on the cloud where everyone can access our Web sites.

Installing and Running WordPress.

- Login to manage.windowsAzure.com to access you Azure portal.
- Click on **CREATE A WEB APP** or **+NEW** button in the portal.
- You can select **QUICK CREATE** and then upload it from your computer.

On clicking Gallery the model window will appear where you can find WordPress by scrolling (or navigating to **BLOGS**, then **WordPress**).

- Select the WordPress and click next. Fill the required Information.
- Fill the URL, database, region and deployment setting details.
- **Click on Next:** Fill in information about MySQL database. Azure will choose the unique name of your database and region to place the database physically.

Question 50: What is Azure HDInsight?

Answer

Azure HDInsight deploys and provisions Apache Hadoop clusters in the cloud, providing a software framework designed to manage, analyze, and report on big data.

With the September 2015 release of HDInsight, now customers configure these clusters to run using both a Windows Server Operating System as well as an Ubuntu based Linux Operating System.

HDInsight on Linux enables even broader support for Hadoop ecosystem users to run in HDInsight providing you even greater choice of preferred tools and applications for running Hadoop workloads. Both Linux and Windows clusters in HDInsight are built on the same standard Hadoop distribution and offer the same set of rich capabilities.

HDInsight

Our 100% Apache Hadoop-based service in the cloud

- ✓ Scale to petabytes on demand
- ✓ Process unstructured and semi-structured data
- ✓ Develop in Java, .NET, and more
- ✓ No hardware to buy or maintain
- ✓ Deploy in Windows or Linux
- ✓ Spin up a Hadoop cluster in minutes
- ✓ Visualize your Hadoop data in Excel
- ✓ Easily integrate on-premises Hadoop clusters

Question 51: How can you create a HDInsight Cluster in Azure?

Answer

To create an Azure HDInsight Cluster, open the Azure portal then click on New, Data Services, and then HDInsight.

The following options are available:

- a. Hadoop is the default and native implementation of Apache Hadoop.
- b. HBase is an Apache open-source NoSQL database built on Hadoop that provides random access and strong consistency for large amounts of unstructured data.
- c. Storm is a distributed, fault-tolerant, open-source computation system that allows you to process data in real time.

The next step is to add a cluster name, select the cluster size, add a password, select a storage and click on create HDInsight cluster.

Enable Remote Desktop on the Cluster:

Once the cluster has been created, its jobs and contents can be viewed by remote connection. To enable remote connection to the cluster, use the following procedure:

1. Click HDINSIGHT on the left pane. You will see a list of deployed HDInsight clusters.
2. Click the HDInsight cluster that you want to connect to.
3. From the top of the page, click CONFIGURATION.
4. From the bottom of the page, click ENABLE REMOTE.

In the Configure Remote Desktop wizard, enter a user name and password for the remote desktop. Note that the user name must be different from the one used to create the cluster (admin by default with the Quick Create option). Enter an expiration date in the EXPIRES ON box.

Question 52: What is Text Analytics API in Azure Machine?

Answer

Text Analytics API is a suite of text analytics web services built with Azure Machine Learning. The API can be used to analyze unstructured text for tasks such as sentiment analysis and key phrase extraction.

The API returns a numeric score between 0 & 1. Scores close to 1 indicate positive sentiment, while scores close to 0 indicate negative sentiment.

The advantage of this API is that a new model need not be designed and trained, the user only needs to bring the data and call the service to get the sentiment results.

However, because this is only the initial release of this service only English is supported right now.

How it works:

Text Analytics API does not simply use a lexicon approach to map words such as “**good or bad**” to return the results.

Instead, it uses advanced natural language processing techniques under the hood.

Thanks for reading this chapter.