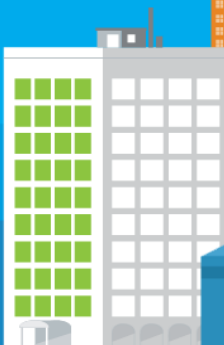# Microsoft AZURE
Step by Step Guide

By: Nitin pandit

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

Microsoft AZURE Makes cloud computing easy..

Total Pages : 463

Microsoft Azure Step by Step Guide

Nitin Pandit

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

Amit Mishra

(Co-Author, Format Editor)

About Author

He is a Microsoft Professional having Master's degree in Computer Science. With over 5 years of experience his rick 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 2.0/3x/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.

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.

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 you 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.

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

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:

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.

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,

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*".

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.

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.

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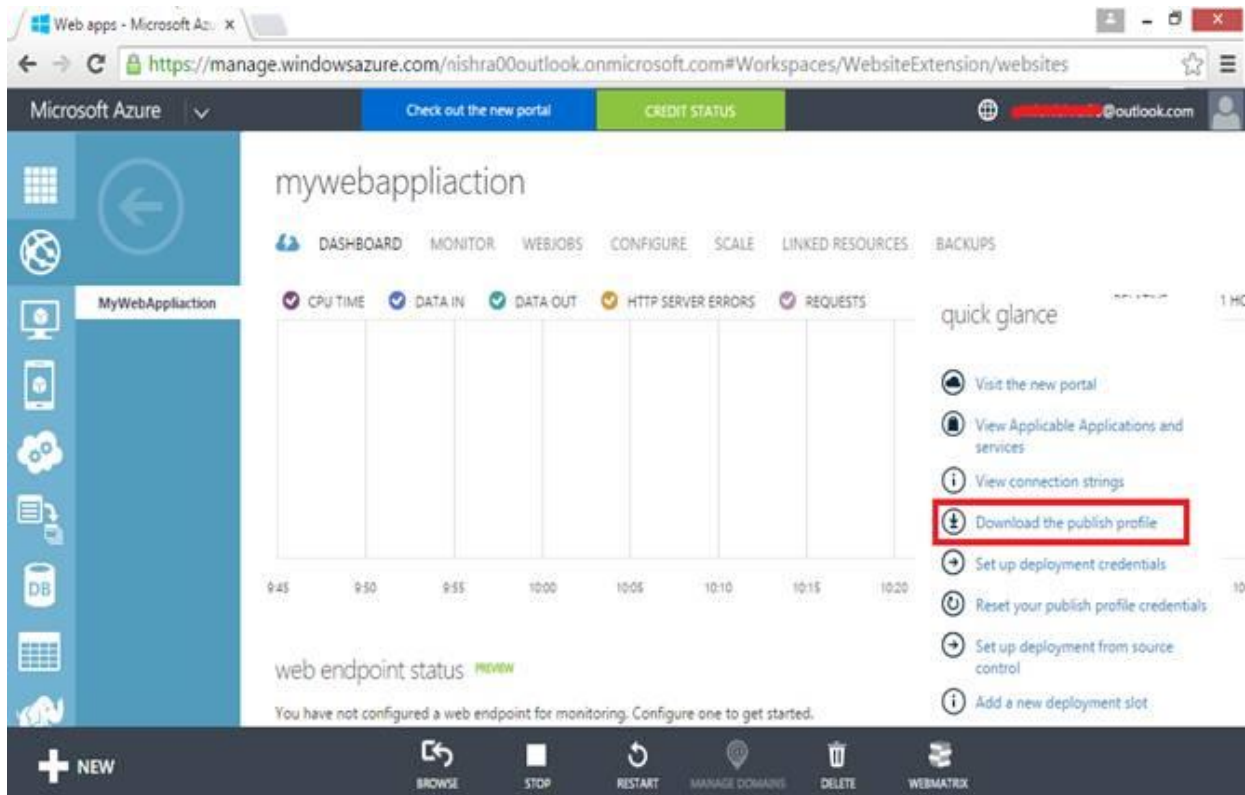

Hello Azure..I'm Nitin Pandit with Web App in Azure with FileZilla

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 particluar 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*".

SETTINGS | **DNS ZONE FILE** | **CONTACTS**

ZONE FILE ⓘ

11 records in this zone

Last updated 09-04-2016 00:55:52 MST

□ Add Record   ⊗ Delete   🖉 Bulk Actions ⌄   ⊙ Templates   ⊕ More ⌄   Filter List ⌄

A (Host) ⓘ

1 Records (0 Selected)

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.

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

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".

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 another CNAME points to map.

Click on to save changes to save all the changes.

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

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.

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.

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.

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

Here you can see, our virtual machine is creating.

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 in 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=MyD
BForVS;User ID=Admin@Nitin@cdlmzzs22c;Password=
{your_password_here};Trusted_Connection=False;Encrypt=True;C
onnection 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\n\r\ntry {\r\n   $conn = new
PDO ( \"sqlsrv:server =
tcp:cdlmzzs22c.database.windows.net,1433; Database =
MyDBForVS\", \"Admin@Nitin\", \"{your_password_here}\");\r\n
```

JDBC:

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

©2016 C# CORNER.

SHARE THIS DOCUMENT AS IT IS. PLEASE DO NOT REPRODUCE, REPUBLISH, CHANGE OR COPY.

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:

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

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

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

```csharp
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:

After button click:

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

Here's the data in table in Visual Studio.

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.

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.

← → C 🗋 localhost:19689/default.aspx
Employee ID   1
Employee Name   Nitin Pandit
Emplolyee Salary   123456
Add

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

← → C 🗋 localhost:19689/default.aspx
Done !!

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.

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:

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 foe .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.

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.

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

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.

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.

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.

©2016 C# CORNER.

SHARE THIS DOCUMENT AS IT IS. PLEASE DO NOT REPRODUCE, REPUBLISH, CHANGE OR COPY.

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.

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 "**nitindemostroage**" named Storage Account, go to bottom of the window and select "**Manage Access Keys**".

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

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:

Code

```csharp
1.  string accountname = "nitindemostorage";
2.
3.  string
    accesskey = "8TeQ8Li/b5U75aSPeTUvmg6w6s6emWSC5KKksNR7hRmSS5GmxzY8wxXFxJtNlwnwSbvlWGAX07
    dnR0W8jrb+Og==";
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.

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

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

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.

We make some code in the Web Config file also.

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

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.

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,

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.

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**".

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

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/5WgOZzQ7wSQv6BSZ2fbJW3sSEAzinmOUhorUWFpfJSUKPUopAD5uDIP3lvJujmnw1hBoD2RFSzHlPYt2nLw==;";
    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(newBlobContainerPermissions
        {
            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/5WgOZzQ7wSQv6BSZ2fbJW3sSEAz
            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 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
{
    2
    2 references
    public class BlobServices
    {
        5 references
        public CloudBlobContainer GetCloudBlobContainer()
        {
            string connString = "DefaultEndpointsProtocol=https;AccountName=nitindemostorage;AccountKey=jH/5WgOZzQ7wSQv6BSZ2fbJW3sSEAzi
            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();
publicActionResult 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 "using**Microsft.WindowsAzure.Storage.Blob**;"

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";

}
```


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

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.

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

Go to "**STORAGE**", and select "**nitindemostroage**".

In the "**nitindemostorage**", select the "**CONTAINERS**".

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

Click on "**mysample**".

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.

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 Blo**b**s 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>
```

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]
publicActionResult Edit(string Name)
{
    return PartialView("Edit", Name);
}
//Second Action for [HttpPost] request:

[HttpPost]
publicActionResult Edit(FormCollection images)
{
    try {
        Uri uri = newUri(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] asHttpPostedFileBase;
            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:

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: < imgsrc = @Model alt = "image here is"
    width = "100"
    height = "100" / > < /div> < inputtype = "hidden"
    value = @Model name = "ImageName" / >
      < inputtype = "file"
    name = "avatar" / >
      < inputtype = "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.

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.

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).

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 much 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 containing multiple collections so every collection can contain stored procedure, triggers and related documents, JSONfiles.

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.

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**".

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.

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.

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.

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.

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 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.

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.

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.

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.

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.

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.

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.

Move: You can move to resources.

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.

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.

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

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)

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.

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.

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*".

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.

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.

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.

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**".

After clicking on to "**Download**", it will download a project, click on to open it.

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.

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.

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

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

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.

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

There are no data connections, Click on to "*Add*" button to add a database.

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. You Mobile App backend is now ready to use with your client app.

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

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.

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.

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.

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

Click on to "*TodoItem*" to check your table.

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.

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.

So here I have successfully changed my database.

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.

Click on to "**Portal**".

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".

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.

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 Connecto**r."

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**.

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.**"

Copy the "URL" from container properties.

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.

Now, I need to create our "Twitter Connector," for it go to "Browse" and type "Marketplace." Click "**Marketplace**".

In search box, type "Twitter Connector."

You will find "Twitter Connector." Click on it as in the following:

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."

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."**

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.)

Check on "Yes," and then click "**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.

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.

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.

Here are all my apps in one resource, "**IMNitin**" named resource.

For the "URL", click on to "Twitter Connector".

Click on to "**Authentication."**

Copy the "**Redirect UR**I" from here.

Paste the "URI" in "WebSite" and "Callback URL" in the twitter application page.

Click on "**Update Settings**" button to save the details.

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.

Click on "**Triggers and actions.**"

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.

Now, we need to create twitter connector, where I fetch my twitter applciation.

So just select the "**Twitter Connector.''**

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.

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.

Now click on "**Content**" option and click on rectangle button, and select "**Tweet Text**" option from the dropdown.

Click on the green check.

Click on to "**Save**" button, to save the current actions.

Click on "**Run Now**" button to run the triggers and actions.

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."**

So, we can see the query, operations, authentication, and result.

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://flowprodcu05sg01.blob.core.wind...

SIZE
385 bytes

OUTPUTS

OUTPUTS LINK
https://flowprodcu05sg01.blob.core.wind...

SIZE
6441 bytes

"x-access-level": "read-write",
"x-connection-hash": "f15230f13e7fc105521b043fd7e31b3b",
"x-content-type-options": "nosniff,nosniff",
"x-frame-options": "SAMEORIGIN",
"x-rate-limit-limit": "180",
"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-newurl": "https://twitterconnectorfd19a
"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 &amp
        "TweetID": "697725363980472321",
        "Created_at": "Thu Feb 11 10:14:24 +0000 2016",
        "Retweet_Count": 4,
        "Tweeted_By": "Deananth"
    },
    {
        "TweetText": "Join in our 12th annivesary 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.

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

RECURRENCE

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.

Click on "*HISTORY*" to check all the history of your jobs which are faulted or succeeded.

In the jobs panel, there will be all the details of your job collections as well as status of your job.

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.

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 Wide vine).

- 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.

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.

In the *JOBS* 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.

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.

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.

Let's go back into content tab to upload a media.

Click on to *Upload* button to upload a media.

In the Upload Content choose the file from local computer; make sure file should not be more than 200 MB.

It will generate a content name for your media which show that the file is in MP4 format.

Click on to next for further.

You can see the file is uploaded. It would upload fairly quick so that file is not published yet.

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.

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.

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.

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.

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.

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace TestAPI.Controllers
{
    [Authorize]
    0 references
    public class ValuesController : ApiController
    {
        // GET api/values
        0 references
        public IEnumerable<string> Get()
        {
            return new string[] { "value1", "value2" };
        }

        // GET api/values/5
        — references
        public string Get(int id)
        {
            return "value";
        }

        // POST api/values
        0 references
```

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

Connect your dataset to project columns as a relationship,

Click on to project columns to launch the column selector where you can select the desired values.

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. 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.

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.

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*".

Experiment created on 11/04/2016   Finished running ✓

Now, the "Train Model" box contains a trained model.

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

> ◢ Statistics

	PassengerId	Survived	Pclass	Name	Sex	Age
view as						
	720	0	3	Johnson, Mr. Malkolm 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. Eiriik	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

◢ Visualizations

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

Predicted Class

	0	1
0	92.1%	7.9%
1	32.4%	67.6%

Actual Class

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.

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.

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 Micrsoft 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.

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 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.

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.

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.

Go to the User Access tab, how many users are there, you can add more users, and can perform many operations.

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.

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:

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*".

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,

In *CONFIGURE* tab, scroll down and you will see Shared Access policies; access policy is responsible to provide the access rights.

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.

Click on to "Save" button to save your policies.

Now I have two policies, now let's see my Queue Dashboard, scroll down and select "*View Connection String*".

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	
ReadPoilicy	Endpoint=sb://myservicebusazure.servicebus.windows.net/;SharedAccessKeyName=Re	

Now you can check the same connection string of both polices 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'.

SAS

NAME	CONNECTION STRING	
WritePolicy	Endpoint=sb://myservicebusazure.servicebus.windows.net/;SharedAccessKeyName=W	
ReadPoilicy	Endpoint=sb://myservicebusazure.servicebus.windows.net/;SharedAccessKeyName=Re	

In *TOPICS* panel, click on to New button to create a new topic.

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.

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-usage 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 easilyove out m 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: " + message.AsString;
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.             // Remove message from queue
14.             message.Complete();
15.         }
16.         catch (Exception)
17.         {
18.             // Indicate a problem, unlock message in queue
19.             message.Abandon();
20.         }
21.     }
22. }
```

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 observers 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 partioned 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:

Federation Object

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 a 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 using 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

    o App Engine.
    o 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 then 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.

Template	CloudFormation	Stack
JSON formatted file	Framework	Configured AWS services
Parameter definition	Stack creation	Comprehensive service support
Resource creation	Stack updates	Service event aware
Configuration actions	Error detection and rollback	Customisable

_

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.

C#Corner

- 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.

©2016 C# CORNER.

SHARE THIS DOCUMENT AS IT IS. PLEASE DO NOT REPRODUCE, REPUBLISH, CHANGE OR COPY.

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.

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.