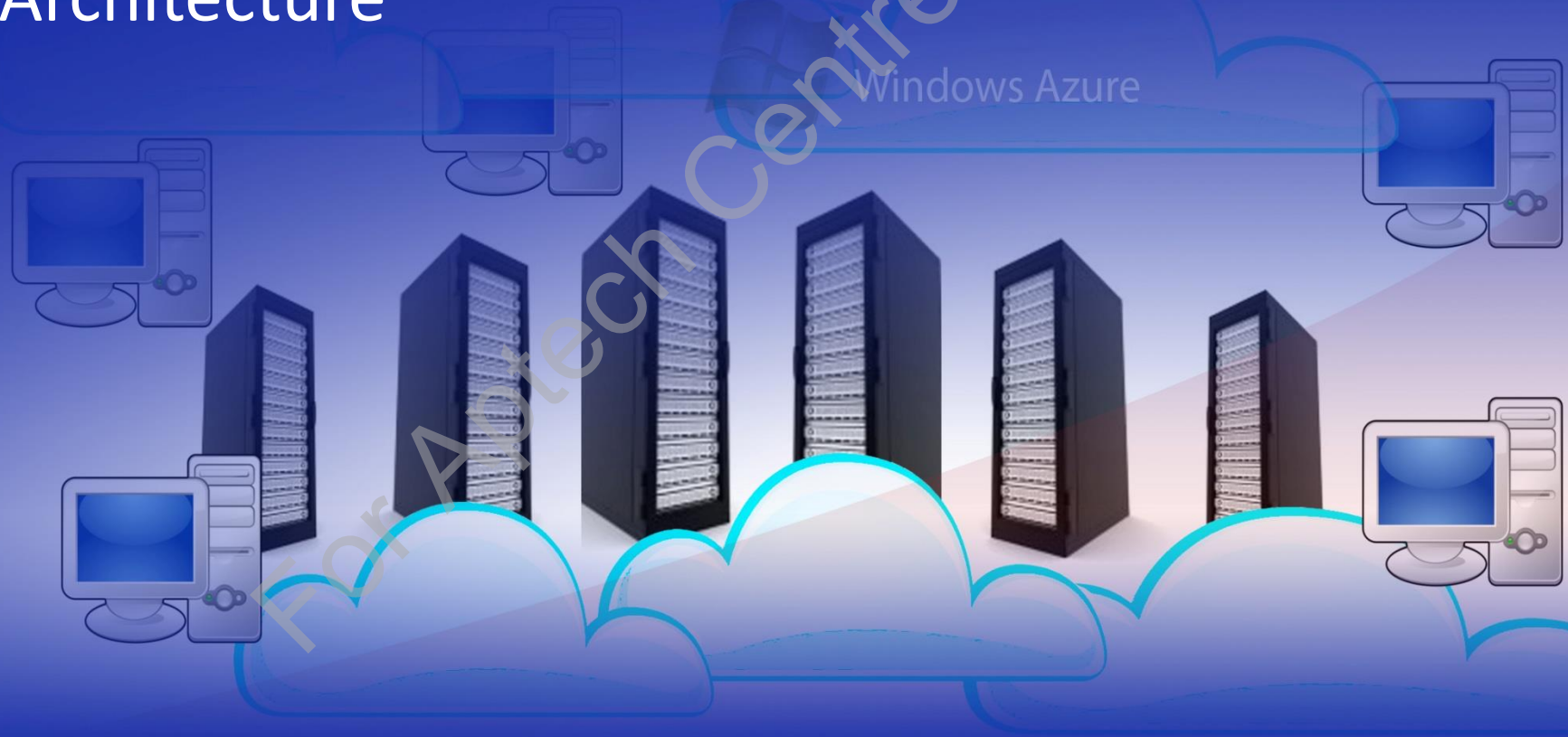


Enterprise Application Development Using Windows Azure and Web Services

Session 2

Understanding Windows Azure Architecture



Learning Objectives



- Describe the Windows Azure platform architecture
- Describe the components of the Windows Azure architecture
- Explain the process to create, test, and deploy an Azure cloud service using Visual Studio
- Explain the process to create, test, and deploy a Web application on the cloud

Window Azure Architecture 1-5

❑ The Windows Azure Services Platform has Windows Azure as its cloud OS.

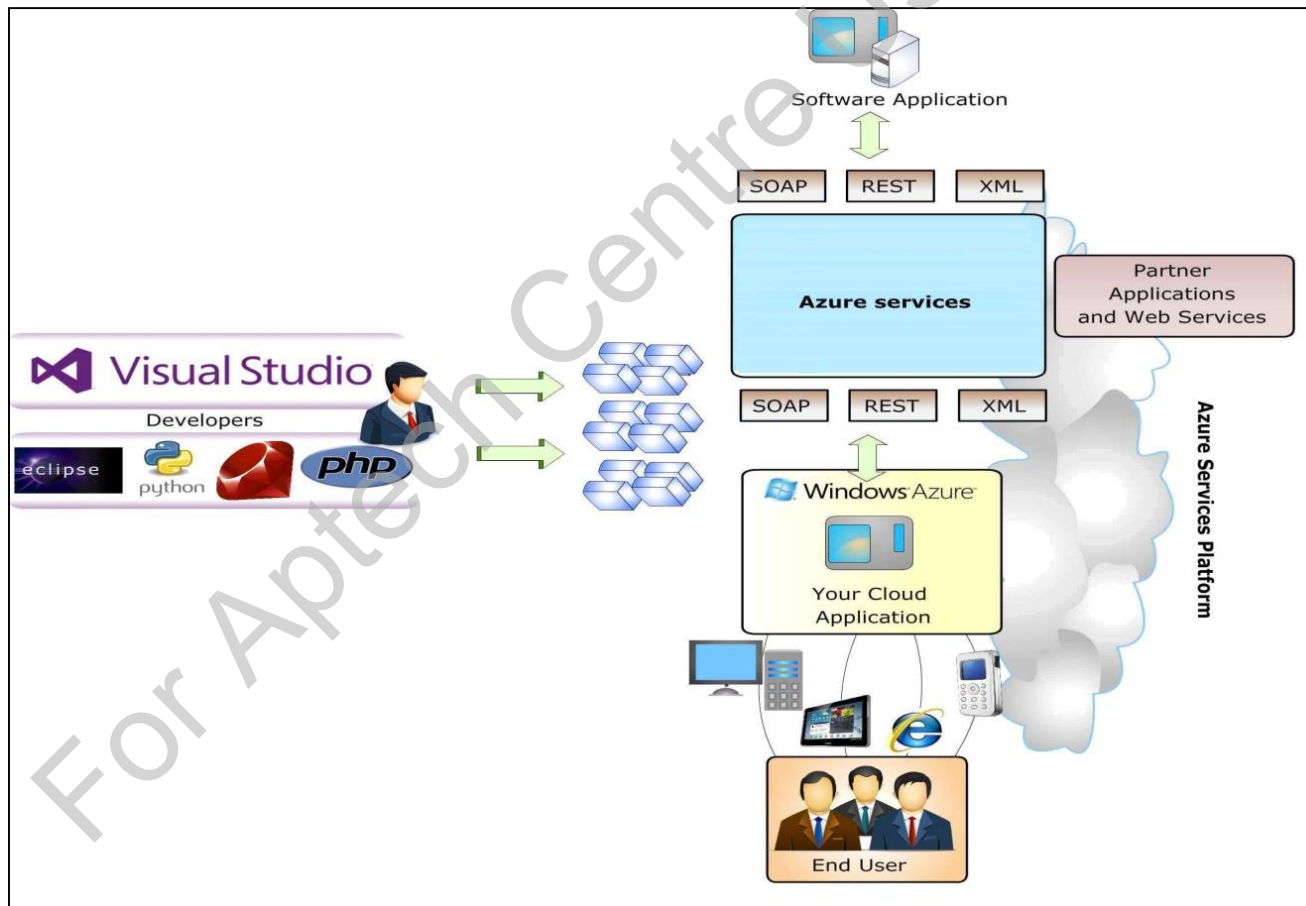
❑ The OS:

- Facilitates business and consumer applications to be built, deployed, and hosted in the cloud.
- Provides a platform to developers to address both business problems as well as customer needs.
- Allows deployment of Web based applications both in-premise and/or off-premise applications.
- Utilizes the organizations in-house services and serves as a runtime environment for applications.
- Includes a mix of application, virtualization service layers, storage, security, and a desktop development environment.
- Handles the load balancing, caching, redundancy, resource management, and life cycles of hosted services.



Windows Azure Architecture 2-5

- ❑ The following figure shows the Windows Azure PaaS cloud architecture:



Windows Azure Architecture 3-5

- ❑ The hosted services of Windows Azure OS can be categorized into one of the following roles:

Web role

- This role instance accepts all incoming HTTP or HTTPS requests. It also supports web applications that are hosted in Internet Information Services (IIS).



Worker role

- This role instance is similar to Web role, but it cannot support Web applications hosted in IIS.



Virtual Machine (VM) role

- This role instance allows creation and running of code in the Windows Azure OS.



Window Azure Architecture 4-5

❑ Applications hosted in the Windows Azure OS:

- Can be of one of the mentioned roles or a combination of these roles.
- Can be used to activate multiple virtual machines with multiple role instances.



❑ Windows Azure OS:

- ❑ Simplifies IT management and optimizes the up-front and ongoing expenses.
 - Facilitates management of scalable Web applications over the Internet.
 - Reduces the need for maintaining onsite resources that are expensive and underutilized.
 - Supports Microsoft based standards and protocols and also third-party standards, programming languages, and several other platforms.

For example, HTTP, HTTPS, XML, SOAP, Representational State Transfer (REST), Ruby, PHP, Python, and Eclipse.

Window Azure Architecture 5-5

❑ Windows Azure Services Platform:

- Has Windows Azure as its cloud OS.
- Facilitates business and consumer applications to be built, deployed, and hosted in the cloud.
- Provides a platform to developers to address both business problems as well as customer needs.
- Allows deployment of Web based applications both in-premise and/or off-premise applications.
- Utilizes the organizations in-house services and serves as a runtime environment for applications.
- Includes a mix of application, virtualization service layers, storage, security, and a desktop development environment.
- Handles the load balancing, caching, redundancy, resource management, and life cycles of hosted services.



Compute

- ❑ **Compute** is one of the core components of Windows Azure.
- ❑ Windows Azure provides four services under Compute, which are Virtual Machines, Cloud Services, Websites, and Mobile Services:

Virtual Machines

- Helps to create virtual machines based on the requirements.
- Performs a number of management tasks.

Cloud Services

- Helps to configure, manage, and scale the cloud services.
- Performs tasks, such as creating a cloud services, linking resources to the cloud services, and so on.

Web Sites

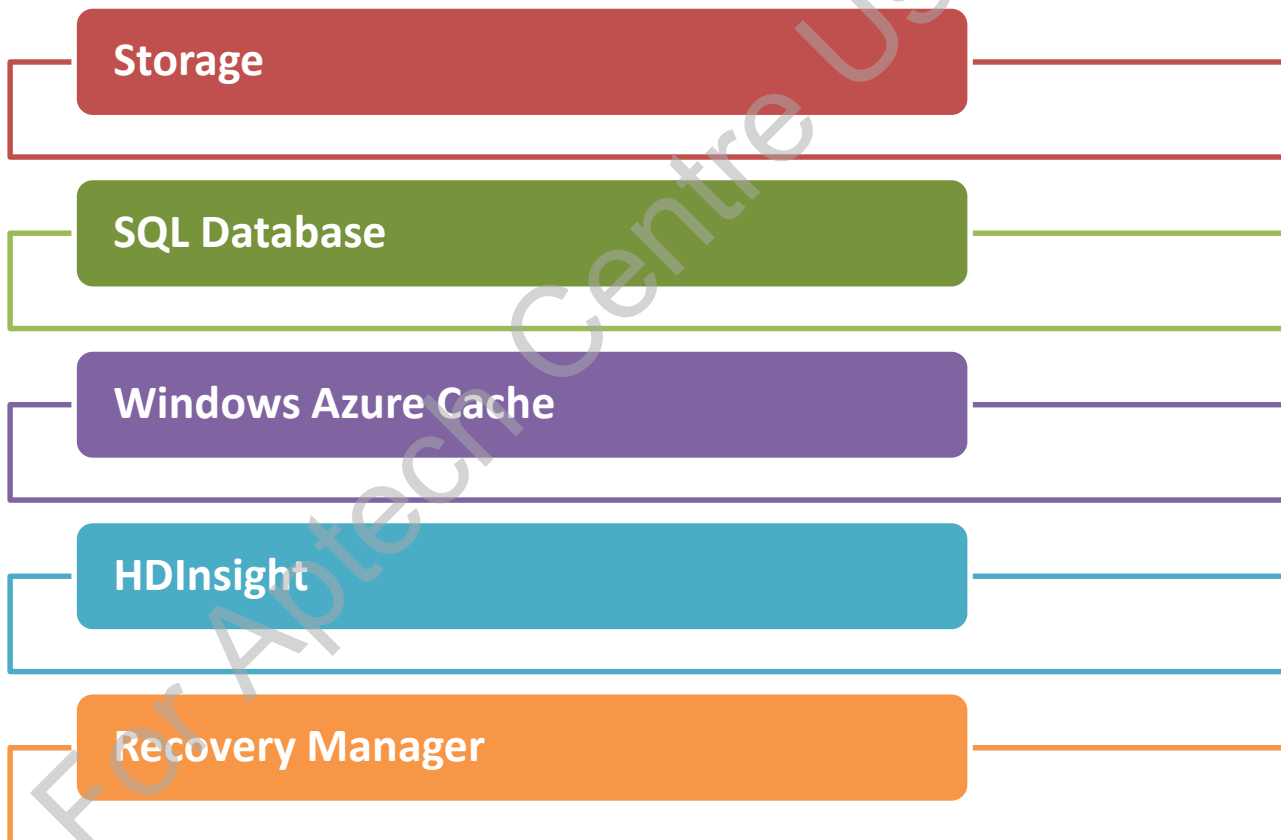
- Allows hosting a large number of Websites.
- Allows to version control the source code of these Websites through various version-controlling repositories such as Github.
- Performs tasks such as configuring session states, and so on.

Mobile Services

- Allows to create a backend of the mobile apps. The data can be stored on the cloud and fetched on demand.

Data Services 1-5

- Windows Azure provides five main data services:



Data Services 2-5

Storage

❑ Windows Azure Backup Storage services:

- Allows the developers to store different kinds of data, such as unstructured, non-relational, and binary data.
- Uses blobs for storing such data and queues for storing message information for the clients.



SQL Database

❑ Windows Azure Backup Storage services:

- Allows developers to either use the SQL Database on the cloud or share data between the SQL Database on the cloud and locally installed instances. These databases can then be used by the applications that are developed by the developers.



Data Services 3-5

Windows Azure Cache



- ❑ The Windows Azure Cache uses Azure Redis Cache for providing fast data access in the following ways:

Cache Service (Preview)

- This service creates and manages a cache only for your business critical data.
- This dedicated cache can be accessed by a secured and publicly addressable endpoint. You can also use the Windows Azure Management Portal to configure and monitor a cache.

In-Role Cache

- This service is used in sync with the role instances of Windows Azure cloud OS.
- This uses the one role – one service model for hosting a cache. This implies that a running instance of a role, be it worker role or a combination of roles, and the memory or storage resource combines to form a cache cluster.

Shared Cache

- This is a separate service provided by the Windows Azure Management Portal. It is utilized as a shared service and it resides on a group of servers.
- It is a paid service depending on the memory and bandwidth of client applications.

Data Services 4-5

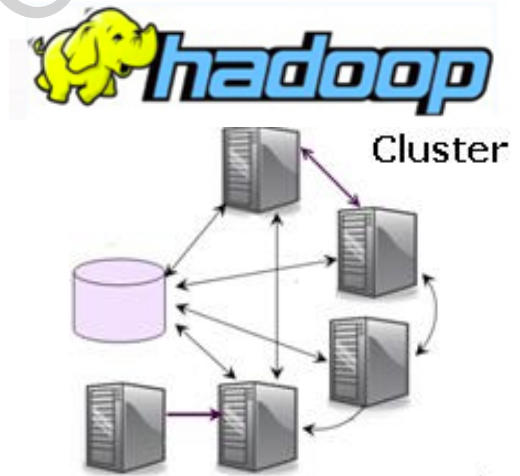
HDInsight

❑ The Windows Azure HDInsight Services:

- Uses Apache Hadoop for drawing analytics from the unstructured data.
- Allows the developers to build an appropriately sized Hadoop clusters as and when required for analysis of large amounts of unstructured data.

❑ The Windows Azure PowerShell Service:

- Helps HDInsight Service to configure, run, and post-process Hadoop jobs. To enable these jobs, HDInsight Service uses .NET library classes.



Data Services 5-5

Recovery Manager



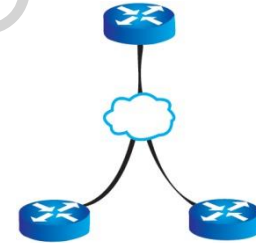
❑ The Windows Azure Services:

- Allows you to protect data.
- Ensures that your clouds are protected.
- Uses services such as Hyper-V Recover Manager to automate the recovery.
- Provides backup services to automate the backups.

Networking 1-4

- ❑ Datacenters are geographically spread out and they run Windows Azure cloud OS to manage and store applications and data.

- You can use any datacenter to run your applications or to store your data. These datacenters can be connected in two ways:



Windows Azure Virtual Private Network (VPN)

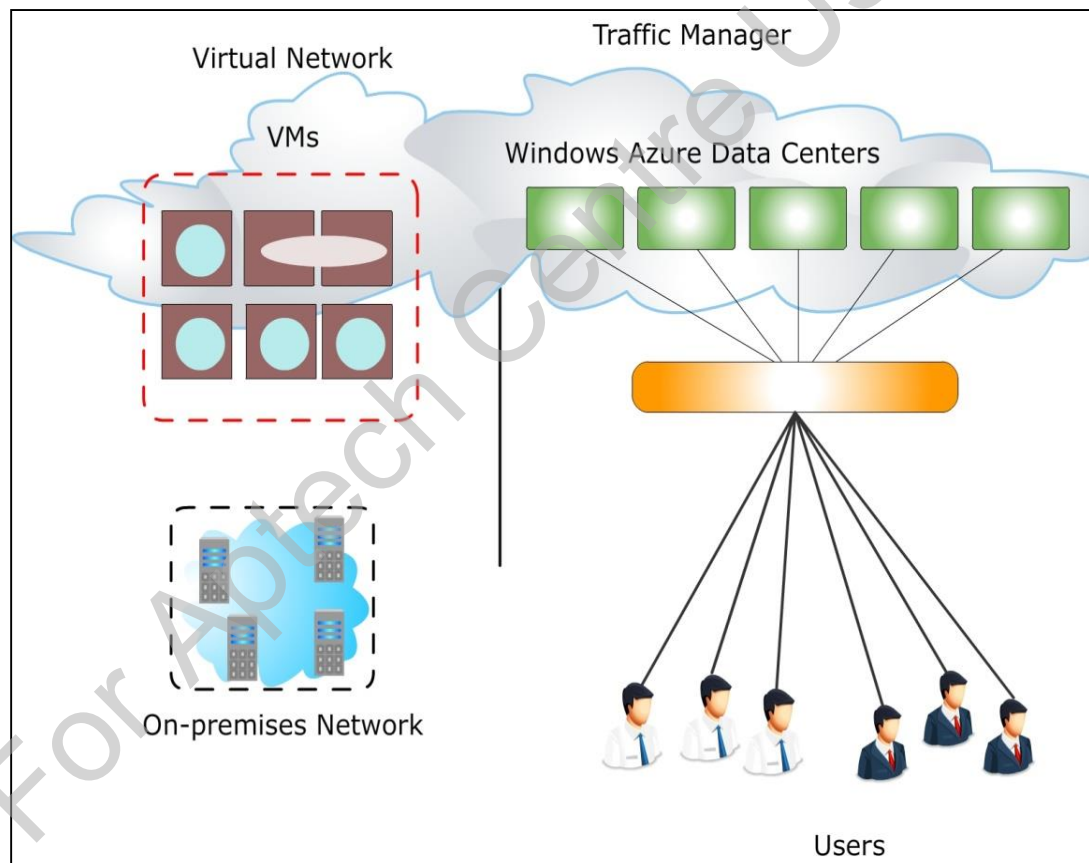
- You can use this network connection to connect an on-premises local network to a pre-defined set of Windows Azure VMs.

Windows Azure Traffic Manager

- When applications are running at more than one datacenters in a Windows Azure environment. This will help you to route requests from one instance to another.

Networking 2-4

- The following figure shows Cloud VPN and multiple datacenters for data routing:



Networking 3-4

Windows Azure Virtual Private Network (VPN)

❑ Windows Azure Virtual Network helps in:



- Extending your network to the Windows Azure environment as part of your on-premises network.
- Interacting with the datacenter as though it is present in your own premises.
- Setting up the VPN between the local machine and the groups of VMs in Windows Azure environment.

Networking 4-4

Windows Azure Traffic Manager

❑ Windows Azure Traffic Manager:

- Directs the user request automatically to the nearest or another datacenter.
- Instructs the Traffic Manager to perform this task, an application owner needs to define rules of how the user request would be directed to the datacenters.



App Services 1-6

❑ Service Bus Queues:

- Support messaging services.
- Use the queue to exchange the messages.

❑ Working of the queue:

- In an asynchronous manner, the message producer, which is a sender, sends the message to a queue and continues its processing.
- On the other side, the receiver or the message consumer pulls the message from the queue and gets back to its processing.
- Fundamentally, a queue has First In First Out (FIFO) mechanism of delivering the messages.
- To use the Service Bus Queue of Windows Azure, you need to create a service namespace.



App Services 2-6

❑ Steps to create a service namespace are:

Step 1

- Using your credentials, login to <http://manage.windowsazure.com>

Step 2

- Click **Service Bus**, on the left pane of the Management Portal.

Step 3

- Click **Create** that is at the lower pane of the Management Portal.

Step 4

- Enter the name of the namespace, in the **Add a new namespace** section. The system then checks if the name is available.

Step 5

- On confirmation, select the country where the namespace needs to be hosted (important to remember it should not be different from the country/region in which your computing resources resides).

Step 6

- Click the check mark. The system will create your service namespace and will enable it after you click the check mark.

App Services 3-6

- ❑ Steps to retrieve the credentials for the namespace in order to perform the management operations are:

Step 1

- Click the **Service Bus** node, which is on the left navigation pane. This will display the list of available namespaces.

Step 2

- From the populated list, select the namespace that was just created.

Step 3

- Click **Connection Information**.

Step 4

- In the **Access connection information** dialog box, look for the **Default Issuer** and **Default Key** entries and store them for future use.

App Services 4-6

❑ Creating a queue:

- You can use the `NamespaceManager` class to create and manage queues. The `NamespaceManager` is used for managing the namespace. You can use it to create a queue.
- For example, the following code can be used to create a queue:

```
namespaceManager.CreateQueue("DataCollectionQueue");  
NamespaceManager namespaceManager = new  
    NamespaceManager(uri, tokenProvider);
```

App Services 5-6

❑ Sending messages to a queue:

- For an application to send messages, it has to first create a `MessagingFactory` object. For example, the following code can be used to create the `MessagingFactory` object:

```
MessagingFactory factory = MessagingFactory.Create(uri,  
tokenProvider);
```

App Services 6-6

❑ Receiving messages from a queue:

- To be able to receive message, an application must use the `MessageReceiver` object, which is created from `MessagingFactory` using `CreateMessageReceiver`.
- For example, the following code can be used to create the `MessageReceiver` object:

```
MessageReceiver receiver = factory.CreateMessageReceiver  
    ("DataCollectionQueue");  
BrokeredMessage receivedMessage = receiver.Receive();
```

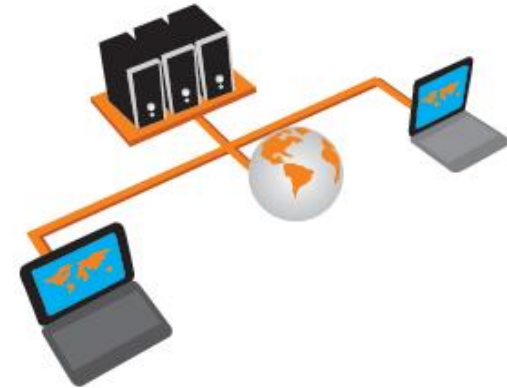
Commerce 1-3

❑ With the onset of Software as a Service (SaaS):

- Creation and usage of cloud applications have become commercial.
- Usage of cloud applications and the associated payments are made online.

❑ Windows Azure:

- Allows potential customers to search and buy Windows Azure applications and commercial datasets deployed in the cloud.
 - The service providers for such commercial exchange include Windows Azure Marketplace and Windows Azure Store.



Commerce 2-3

- ❑ The following figure shows the transaction Windows Azure Marketplace and other components:



Commerce 3-3

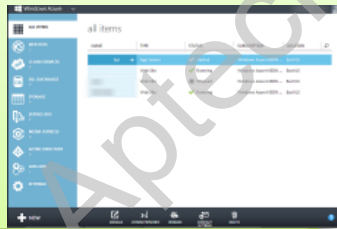
- ❑ There exists differences between Windows Azure Marketplace and Windows Azure Store:
 - The former is located out of the Windows Azure Management Portal and the latter is accessible within the Windows Azure Management Portal.
 - Customers can search for their required applications or datasets and then sign up to use them through the application's creator or directly either through the Marketplace or Store.
 - The Bing Search Application Programming Interface (API) can also be used to search for applications through the Marketplace.



Creating a Azure Cloud Service

- ❑ A cloud service must be created before it can be published in Windows Azure.
- ❑ There are two methods to create a service:

**Using the Windows
Azure Management
Portal**



OR

Using Visual Studio



Creating a Azure Cloud Service Using VS 2013 1-4

□ Steps to create an Azure cloud service using Visual Studio are:

1. Create an Account

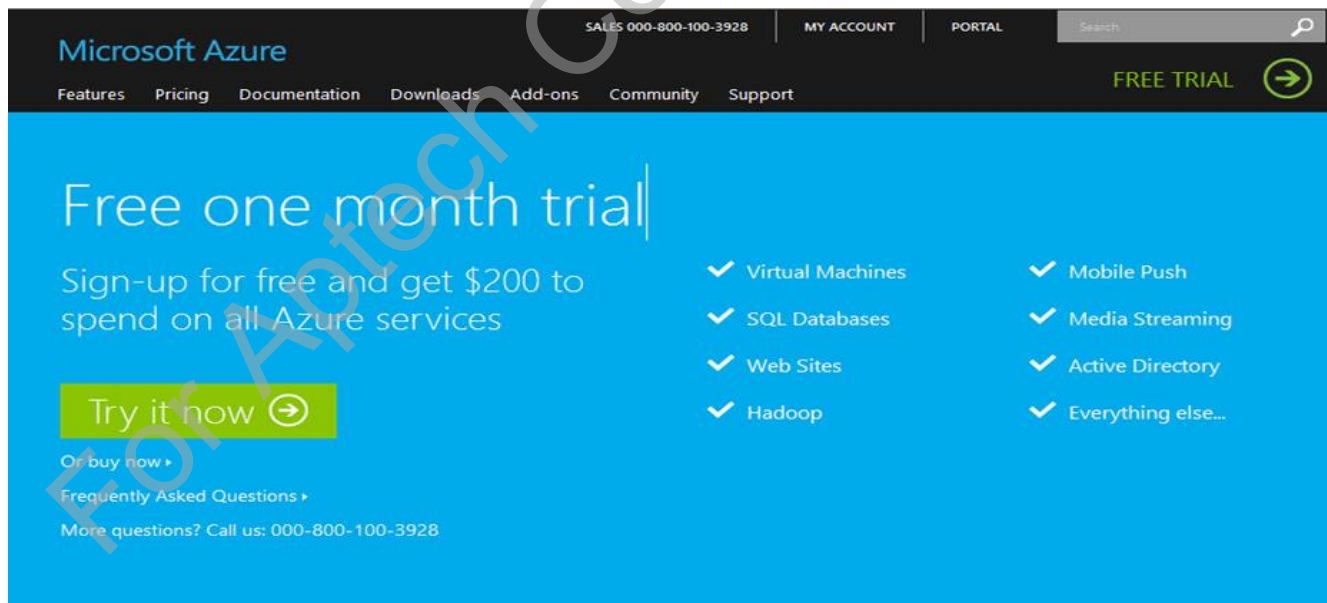
2. Create a Cloud Service



Creating a Azure Cloud Service Using VS 2013 2-4

1. Create an Account

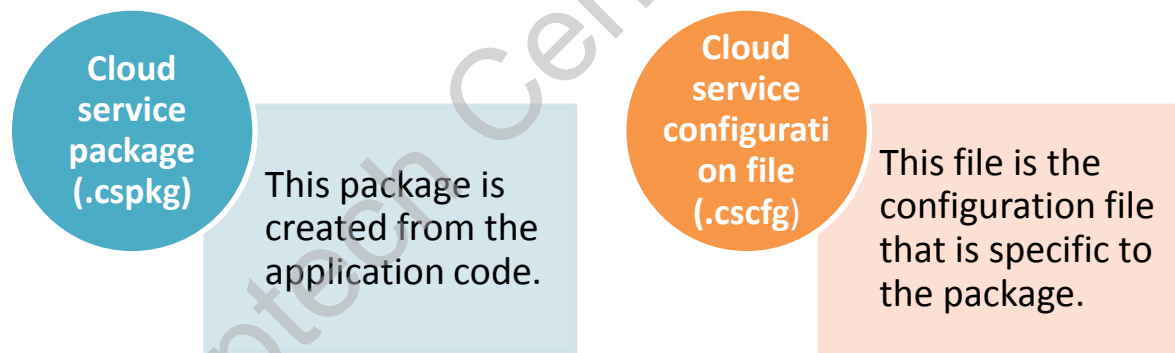
- ❑ Connect to <http://www.windowsazure.com/> and create an account, if you do not have one.
 - If you already have an account, access it by clicking the **Portal** button.
- Windows Azure also provides a free trial as shown in the following figure:



Creating a Azure Cloud Service Using VS 2013 3-4

2. Create a Cloud Service

- ❑ Before deploying a cloud application for Windows Azure, you must create two files:



- ❑ Both these files are specific to the application that you create. You can use the Azure Tools to create these deployment packages and configuration files.

Creating a Azure Cloud Service Using VS 2013 4-4

■ Following are the steps to create a cloud service:

Step 1

- Select the **Windows Azure** shortcut in Visual Studio and click **Publish**.

Step 2

- On the **Windows Azure Publish Settings** dialog box, sign in to your Windows Azure account when prompted. You need to use the account that is linked with the Windows Azure subscription.

Step 3

- Click **Next**.

Step 4

- In the **Windows Azure Publish Settings** dialog box, on the Settings page, from the Cloud Service drop-down, select **Create New**.

Step 5

- Create the service with a new unique name.

Step 6

- Click **Publish** to deploy the service.

Deploying the Cloud Service on Compute Emulator 1-8

❑ The Windows Azure Software Development Kit (SDK) tool:

- Includes a Windows Azure Compute Emulator.
- Helps you to build and test your applications before deploying it.
- Help you to test the basic functionalities of your services, such as configuration, roles, its instances, and its user interface status.



Deploying the Cloud Service on Compute Emulator 2-8

Step 1: Install Prerequisites for Windows Azure Compute Emulator and the Windows Azure Storage Emulator.

- ❑ The table displays the minimum requirements for installing Windows Azure Compute Emulator:

Component	Minimum Requirement
Operating System	64x versions of Windows 8, Windows 7, Windows Vista SP1, or Windows Server 2008. If you intend to use 32-bit version of the SDK, then you will need 32-bit version of the mentioned operating systems.
Framework	Microsoft .NET Framework 3.5 SP1 or higher.
Database	Microsoft SQL Server Express 2005, SQL Server 2008 R2 Express or higher.
Web Server	IIS 7.0 with ASP.NET.

Deploying the Cloud Service on Compute Emulator 3-8

Step 2: Configure operating system features to emulate Windows Azure

- ☐ By default, Windows Server 2008 and IIS 7.0 do not install all services.
- ☐ For example, Windows Azure requires IIS 7.0 with ASP.NET enabled. By default, this service is not installed in Windows Server 2008.
- ☐ You need to ensure all services that you require in cloud are enabled and configured in the same way in Compute Emulator.

Deploying the Cloud Service on Compute Emulator 4-8

Step 3: Trace the execution flow of application

- ❑ After you execute the application, you can monitor the application using the following three classes:
 - `System.Diagnostics.Trace`
 - `System.Diagnostics.Debug`
 - `System.Diagnostics.TraceSource`
- ❑ You can use these classes to record the execution data, which can be stored in the logs and text file.

Deploying the Cloud Service on Compute Emulator 5-8

Step 4: Package and run the application.

- ☐ After you create the application and test its execution, you need to package the application to run in Compute Emulator.
- ☐ You can use **CSPack**, which is a command-line tool for packaging the applications.
- ☐ After the application is successfully packaged, you can run the **CSRun** command-line tool to run the application.

Deploying the Cloud Service on Compute Emulator 6-8

Step 5: Change the configuration of a running application.

- ☐ You can add more instances of a role to the application without shutting down the running application.
- ☐ To achieve this goal, you will need to use an editor, such as Visual Studio, to modify the configuration file and then, update the application using the **CSRun** command-line tool.

Deploying the Cloud Service on Compute Emulator 7-8

Step 6: Determine Compute Emulator limits

❑ The following table lists the limitations of Compute Emulator:

Component	Minimum	Maximum
Roles	1	25
Core	1	20
Input Endpoints	1	25
Internal Endpoints	1	25

Deploying the Cloud Service on Compute Emulator 8-8

Step 7: Configure Internet Protocol (IP) addresses and port allocation.

- ☐ Each Web role is assigned a different IP address.
- ☐ The range of IP address starts from 127.0.0.1 and then incremented by one.
- ☐ For example, the first Web role will be assigned 127.0.0.1 and then second Web role will be 127.0.0.2.
- ☐ If both the Web roles are assigned the same port, then the same port is suffixed after the IP address.
- ☐ For example, the first Web role will be used as 127.0.0.1: 8181, here, 8181 being the port.
- ☐ You can also configure a range for the port numbers.

Deploying the Cloud Service on Windows Azure 1-7

- ❑ Deploying a cloud service on Windows Azure comprises the following processes:

1. Use the development and deployment tools to create an application service provided by the Windows Azure SDK.

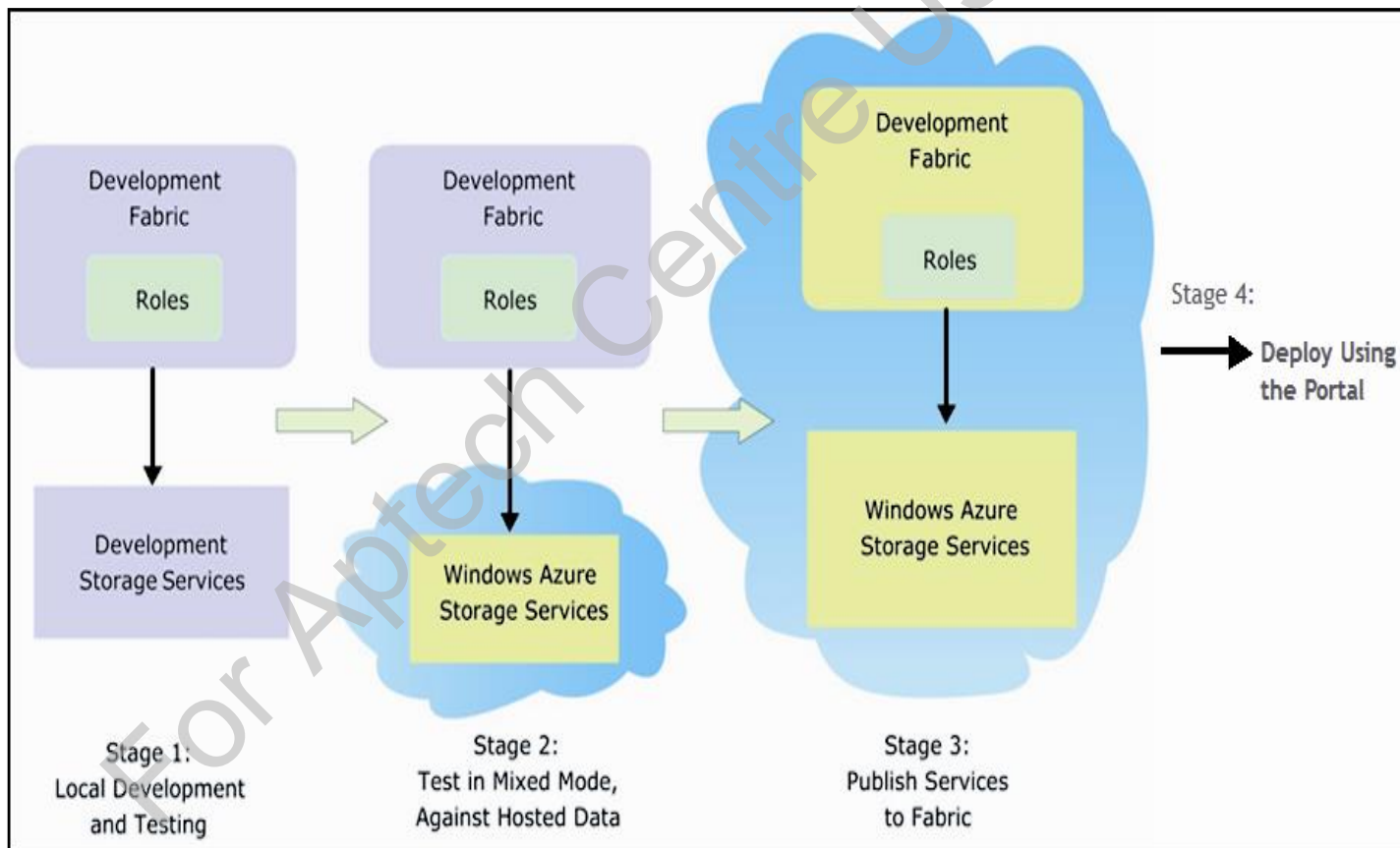
2. Perform the testing and debugging of the application using Windows Azure compute and storage emulators.



- ❑ To package the application for deployment, use the **CSPack** Command-Line Tool.

Deploying the Cloud Service on Windows Azure 2-7

- The following figure gives an overall picture of development and deployment stages:



Deploying the Cloud Service on Windows Azure 3-7

❑ Following are the development and deployment stages:

Step 1

Test your service locally with the Windows Azure compute and storage emulators

Step 2

Debug your service in mixed mode

Step 3

Package your service with CSPack

Step 4

Deploy your service using the Management Portal

Deploying the Cloud Service on Windows Azure 4-7

Test your service locally with the Windows Azure compute and storage emulators



- ❑ Using the Windows Azure compute and storage emulators, you can:
 - Test your application locally because the Windows Azure Compute Emulator simulates the Windows Azure fabric.
 - Check the accuracy of the code locally.
 - Simulates the Windows Azure storage services.
 - Helps you to run and debug code from the storage services.

Deploying the Cloud Service on Windows Azure 5-7

Debug your service in mixed mode



- ❑ In the production environment of Windows Azure services:
 - Run in mixed mode.
 - Execute in the compute emulator while the data is hosted in Windows Azure.
 - Test in the staging environment.
 - Debug and deploy in Windows Azure.

Deploying the Cloud Service on Windows Azure 6-7

Package your service with CSPack

□ During this stage:

- Use the CSPack Command-Line Tool to package your service.
- Use the **cspack.exe** utility file to upload the packaged service Windows Azure.
- Upload the packaged service to the Windows Management Portal and name as <service-name>.cspkg by default.



Deploying the Cloud Service on Windows Azure 7-7

Deploy your service using the Management Portal



□ In this last stage:

- Use the Windows Azure Management Portal to create a cloud service deployable in Windows Azure.
- Upload the following two files, namely, the service package file and the service configuration file. On uploading these files, an internal staging URL will be provided. This URL helps you to test your service in private in the Windows Azure staging environment.
- Finally, when your services are ready, change the service from the staging URL to the production URL.

Testing the Cloud Service 1-3

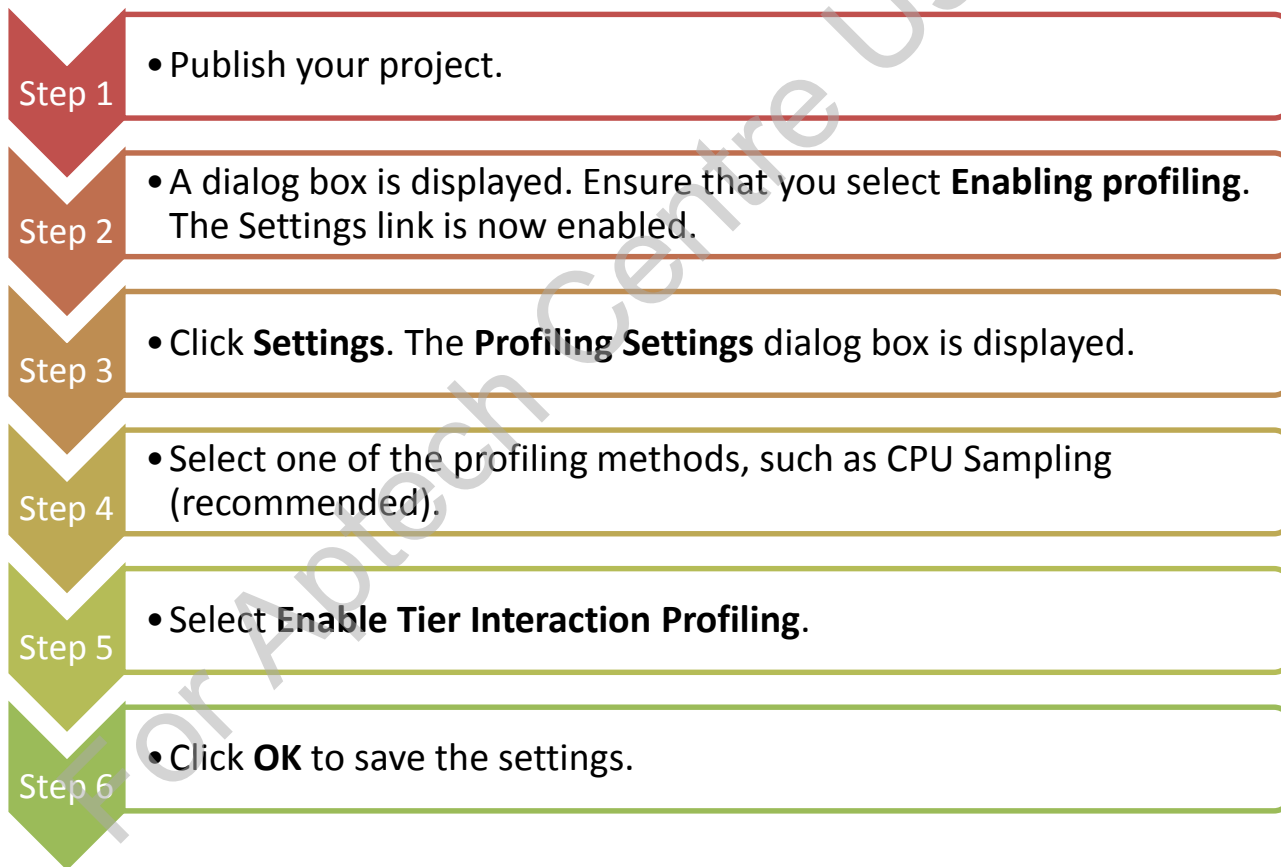
■ Using Visual Studio, you can:

- Perform a number of tests on a cloud service.
- Gather performance reports on Web pages, services, roles, and memory usage.
- Perform the performance testing with the help of Visual Studio Profiler, which is installed by default with Visual Studio Ultimate or Visual Studio Premium.



Testing the Cloud Service 2-3

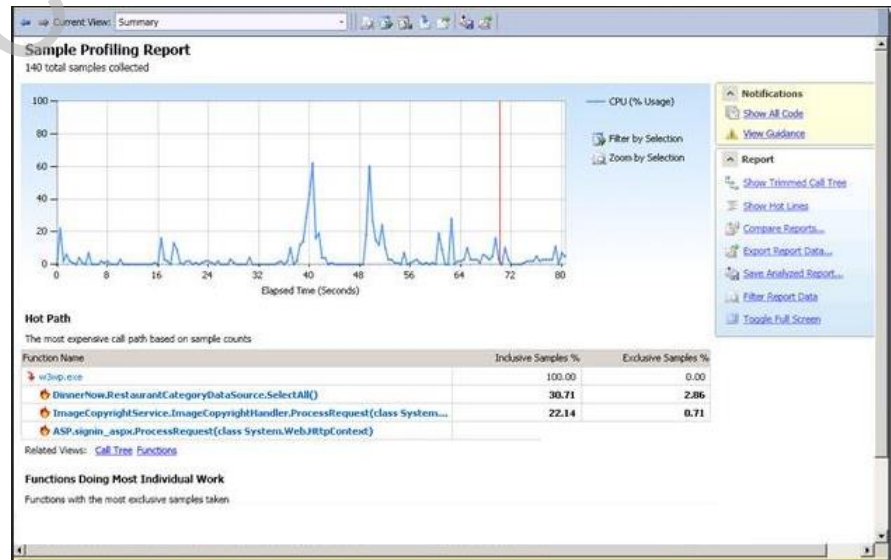
- ❑ To configure Visual Studio Profiler, use the following steps to execute performance testing:



Testing the Cloud Service S-S

- ❑ To view the profiling reports, perform the following steps:
 1. In Visual Studio, open Server Explorer.
 2. Under Windows Azure Compute, right-click the server node and select **ViewProfilingReport**.

The profiling report appears in a new tab in the editor window. The tab is named as **<Role name>_<Instance Number>_<identifier>.vsp.**



Creating and Deploying a Web Application to Windows Azure

- ❑ With the Windows Azure SDK, it is easy for the Web developers to:
 - Create and deploy Web applications and deploy them to Windows Azure.
- ❑ The developer must download and install Windows Azure SDK for .NET.



Creating an ASP.NET MVC Application

Step 1: Open Visual Studio 2013

Step 2: Select **File** → **New Project**.

Step 3: From the **New Project** dialog box, select C# and select **Web** under **Installed Templates**, and click **ASP.NET Web Application**.

Step 4: Select .NET Framework 4.5 as the target framework.

Step 5: Enter the name of application and click **OK**.

Step 6: Select the MVC template and click **Change Authentication**. Then in the **Change Authentication** dialog box, click **No Authentication**, and select **OK**.

Step 7: In the **New ASP.NET Project** dialog box, click **OK**. The application is created.

Deploying the Web Application on Compute Emulator 1-2

❑ Steps to deploy an application in Windows Azure Compute Emulator are:

Step 1

- Once the application is set to run by pressing Ctrl + F5, VS 2013 packages and deploys the application to the Windows Azure Emulator and later, a csx directory is created. Right-click the emulator icon to view and select the option **show compute emulator UI**.

Step 2

- In the Compute Emulator, the application is installed and runs at 127.0.0.1:81. Once the Visual Studio debugger is closed, the application stops running in Windows Azure Compute Emulator.

Step 3

- Now, a cts directory is created and the user can start the application (Website) without using Visual Studio. This leads to the Windows Azure Command Prompt.

Step 4

- Get the directory listings and from the list, select csx and also, a file with local.cscfg extension. This contains information needed to start the application in the Emulator.

Deploying the Web Application on Compute Emulator 2-2

Step 5

- In the console, execute csrun with the two parameters. The command to be given is: `csrun csx\debug ServiceConfiguration.local.cscfg`

Step 6

- This will install the application into Window Azure Compute Emulator. The result of the command identifies that the Web site will be running at 127.0.0.1:81.

Step 7

- Now in the browser, navigate to 127.0.0.1:81 and the browser will display the page.

Deploying the Web Application on Windows Azure 1-3

❑ Steps to deploy an application on Windows Azure are:

Step 1

- Close the browser. In Visual Studio Solution Explorer, right-click the project, and then, select **Publish**.

Step 2

- The **Publish Web** wizard opens, select the profile, and click **Import**.

Step 3

- Use any one of these methods to enable Visual Studio to connect to Windows Azure account:
- Click **Sign In** and enter the credentials for Windows Azure account.
- Select **Manage subscriptions** to install a management certificate that enables access to your account.
- The **Import Publish Settings** dialog box appears.

Deploying the Web Application on Windows Azure 2-3

Step 4

- Select **Import from a Windows Azure Web Site** radio button and select **New**.

Step 5

- In the **Create a site on Windows Azure** dialog box, enter text in the **Site** name box that is used as the unique URL for the Web application.

Step 6

- In the **Location** drop-down list, choose the location that is closest to the user.

Step 7

- Leave the database fields unchanged and then, click **Create**.
- The Website is created. In the **Import Publish Settings** dialog box, the new site appears and is selected in the drop-down list.

Deploying the Web Application on Windows Azure 3-3

Step 8

- Click **OK**.

Step 9

- In the **Connection** tab of the **Publish Web** wizard, click **Validate Connection** to make sure that settings are correct and then, click **Next**.

Step 10

- In the **Settings** tab, click **Next**.

Step 11

- In the **Preview** tab, select **Start Preview**. The tab will display the list of files that would be copied on to the server. The Output window shows the deployment actions taken and reports on successful completion of the deployment.

Summary 1-2

- ❑ The Windows Azure OS is the central component of the Windows Azure Services Platform.
- ❑ Windows Azure OS functions handle load balancing, caching, redundancy, resource management, and life cycles of hosted services.
- ❑ Windows Azure provides four core services, namely Virtual Machines, Cloud Services, Websites, and Mobile Services.
- ❑ Windows Azure provides three main data services, namely, Cache, HDInsight, and Recovery Services.
- ❑ Datacenters are geographically spread out and they run Windows Azure cloud OS to manage and store applications and data.

Summary 2-2

- ❑ Windows Azure Marketplace allows potential customers to search and buy the Windows Azure applications and commercial data sets deployed in the cloud.
- ❑ A cloud service must be created before it can be published in Windows Azure.
- ❑ Windows Azure Compute Emulator and Windows Storage Emulator, together help you to test basic functionalities of your services, such as configuration, roles, its instances, and its user interface status.
- ❑ Using Visual Studio, you can perform a number of tests on a cloud service.