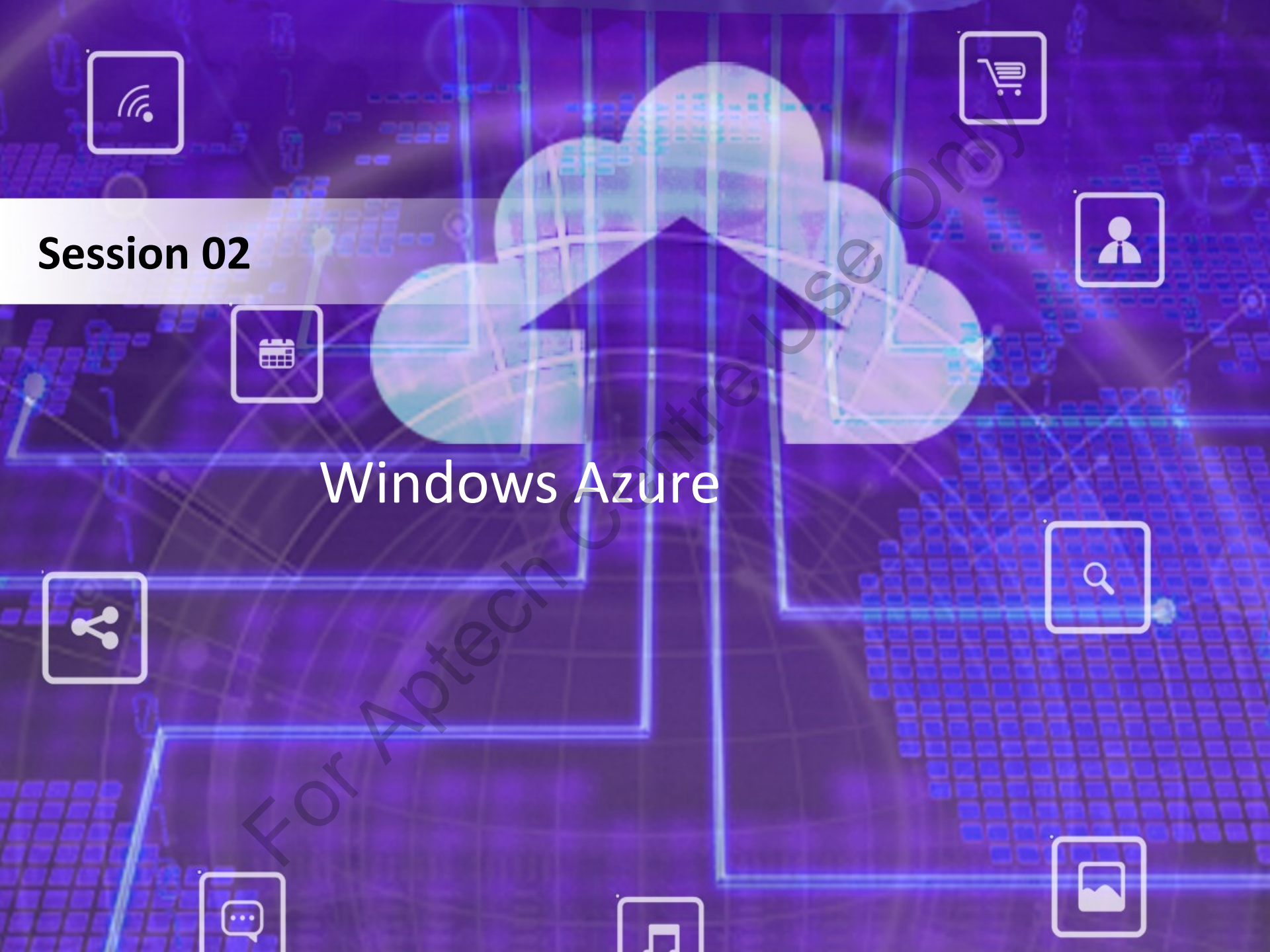


Session 02

Windows Azure

Objectives

- Explain the Windows Azure platform
- Describe the components of Windows Azure
- Describe the process of creation and deployment of a cloud service
- Describe SQL Database
- Describe the process of creation and use of a cloud database
- Explain the services under Windows Azure Client Libraries for .NET

Introduction

Microsoft Windows Azure is a cloud computing platform that enables you to build and run Windows applications and facilitates data storage on the cloud.

Windows Azure also supports non-Microsoft languages, such as Ruby and Python in addition to the various Microsoft languages such as C#, VB.NET, and so on.

Development environments such as Eclipse are also supported.

Necessity for the Windows Azure Platform 1-2

Web hosting is a hosting service provided by companies using which users can make their Web site accessible through the Internet. These companies have their own Web servers that are used to host the applications.

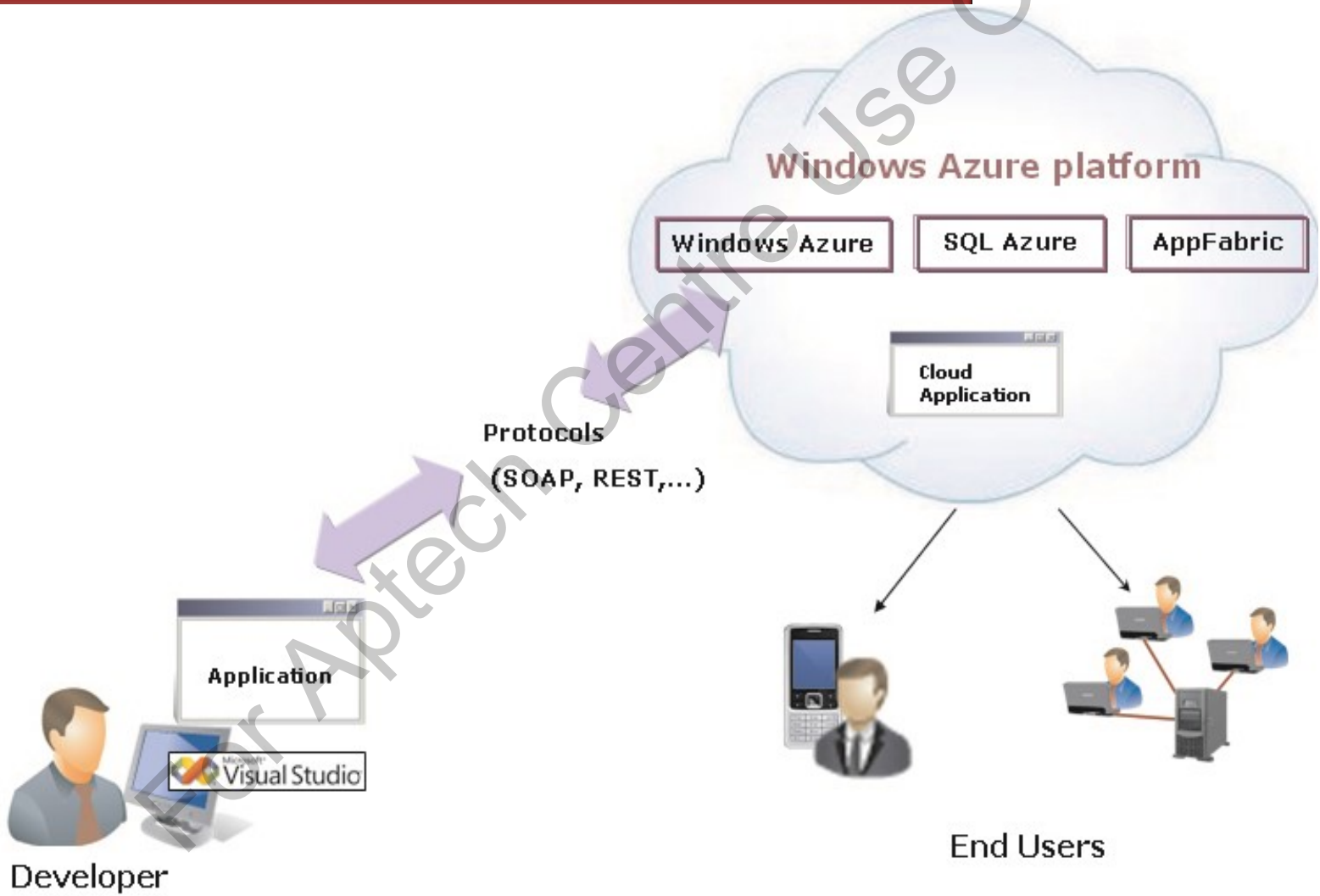
With the advent of cloud computing and globalized enterprises and organizations, there was a need for a cloud based solution for application development, deployment, and hosting.

Microsoft fulfilled this need by creating Windows Azure.

Windows Azure is a cloud computing platform and infrastructure that not only hosts applications, but also enables you to build, deploy, and manage them.

Necessity for the Windows Azure Platform 2-2

Following figure shows an overview of Windows Azure.



Key Features of Windows Azure 1-3

- Windows Azure was designed to simplify IT management and minimize costs. Azure is used to manage Web based applications. Microsoft uses its data centers across various countries to host and manage applications on the Windows Azure platform.
- Many services like virtual machines, network bandwidth, and other infrastructure resources are provided by Azure.
- A feature called dynamic scaling is provided by Windows Azure which can increase or decrease the resources, based on the application's requirements.
- Windows Azure also helps the applications to use and access back end data through storage services and SQL database.

Key Features of Windows Azure 2-3

Key Features of Windows Azure

Virtual Machines

- Enable users to use any OS image in the cloud. The images can be selected from the gallery or customized image can be used. Figure shows some of the OS that are supported in Windows Azure.

Key Features of Windows Azure 3-3

Cloud Services

- Azure cloud services eliminate the need of server infrastructure. Developers can use the Windows Azure SDK and Tools to build, deploy, and manage modern applications.

Web Sites

- Windows Azure Web Sites is a feature that empowers users to organize Web applications on a reliable cloud infrastructure.

Mobile Services

- The Windows Azure platform also includes support in the form of Mobile Services that supports development of applications such as Windows, Apple IOS, Android, and HTML.

Other Features of Windows Azure

- Data Services such as Storage, SQL database, reliable backup, cache, and Hyper-V Recovery manager are also part of the platform.

Benefits of Windows Azure Platform

Windows Azure offers various pricing schemes that can benefit different kinds of users.

There is a free trial for a period of one-month, a pay-as-you-go pricing plan, and a 6-month, and 12-month plan.

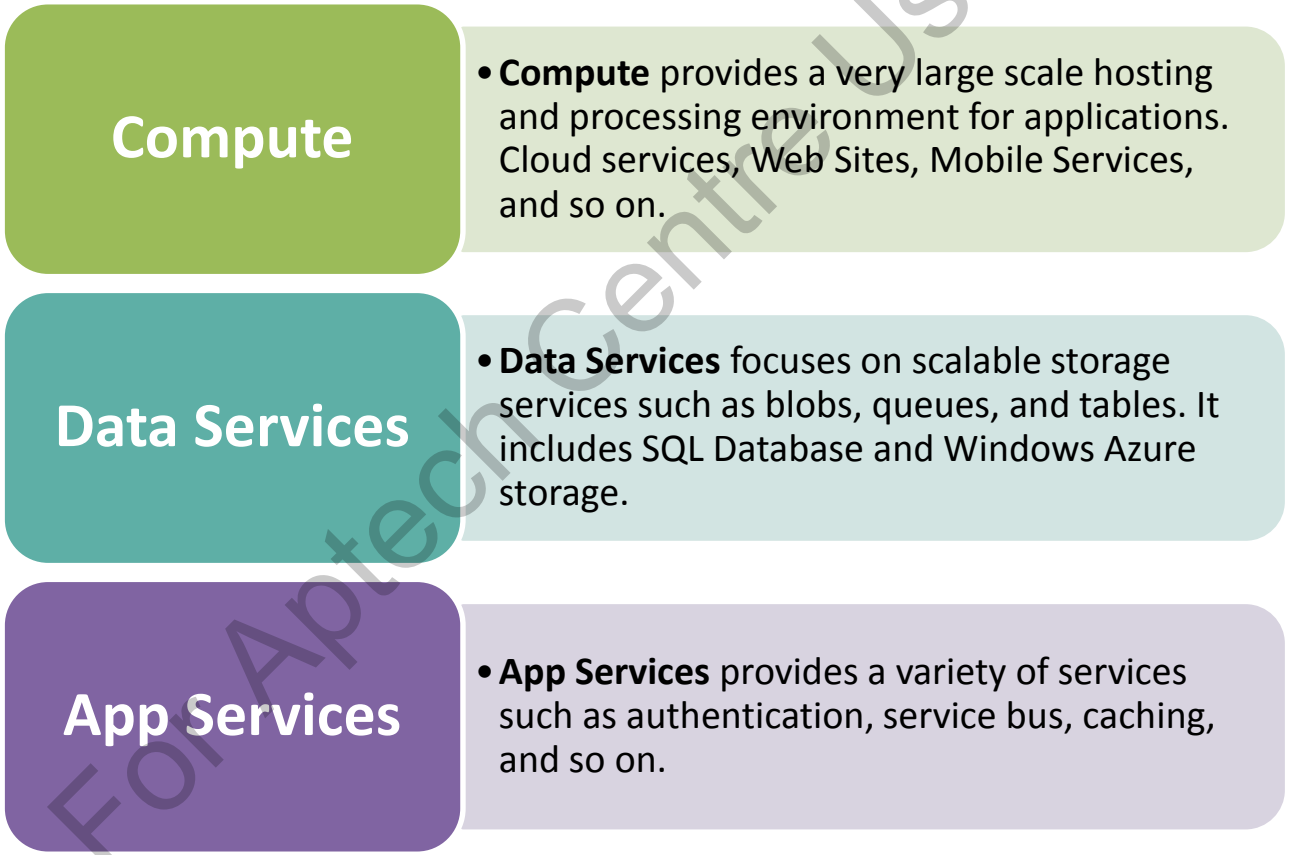
Based on individual or organizational needs, the suitable pricing scheme can be chosen.

Other advantages of using Azure platform include:

- High availability (nearly 99.9%) of the application and the data.
- Hardly any upfront costs of purchasing hardware and software licenses.
- Flexibility to support small to the very largest of customers.

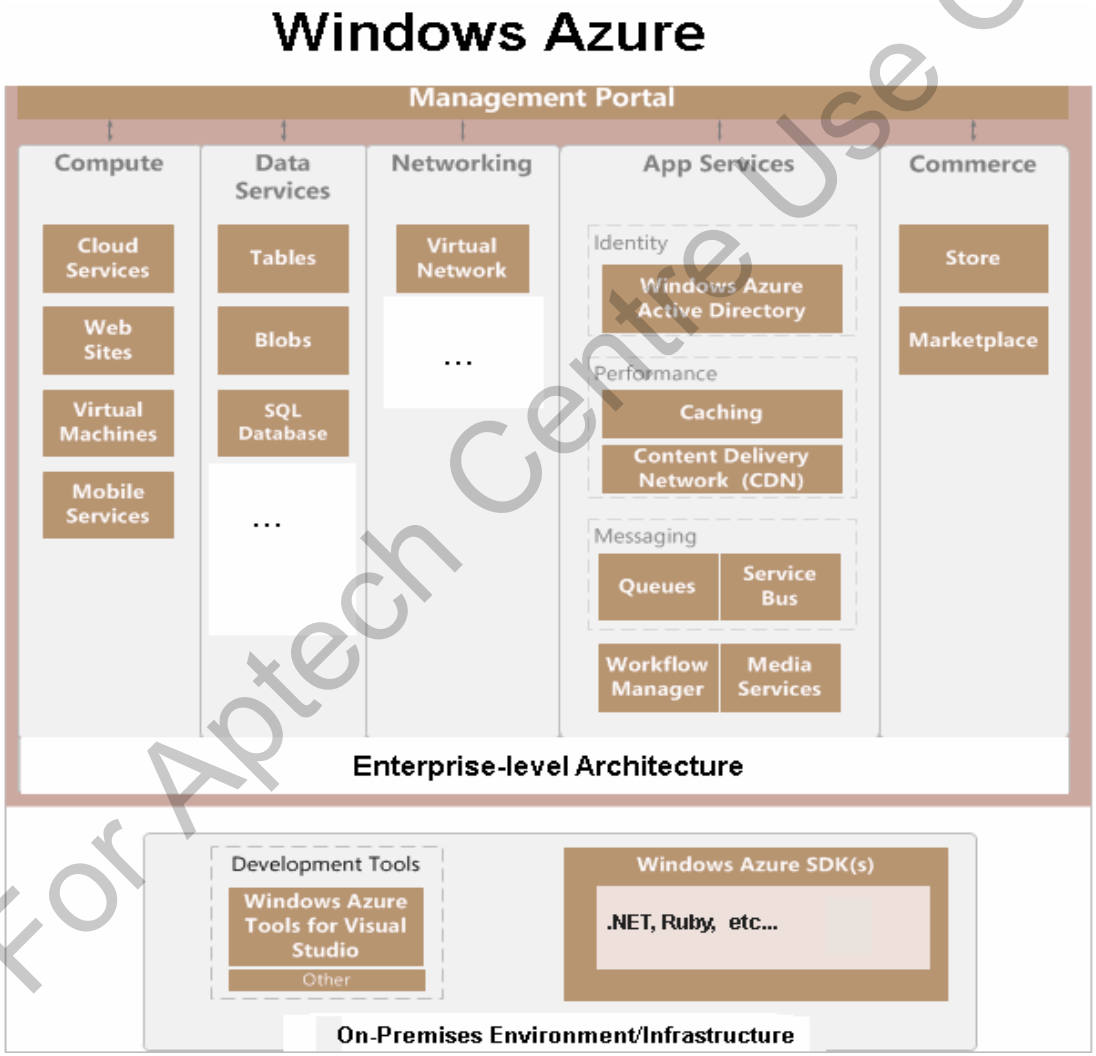
Components of Windows Azure 1-2

The Windows Azure platform architecture includes several components, of which the following are most important:

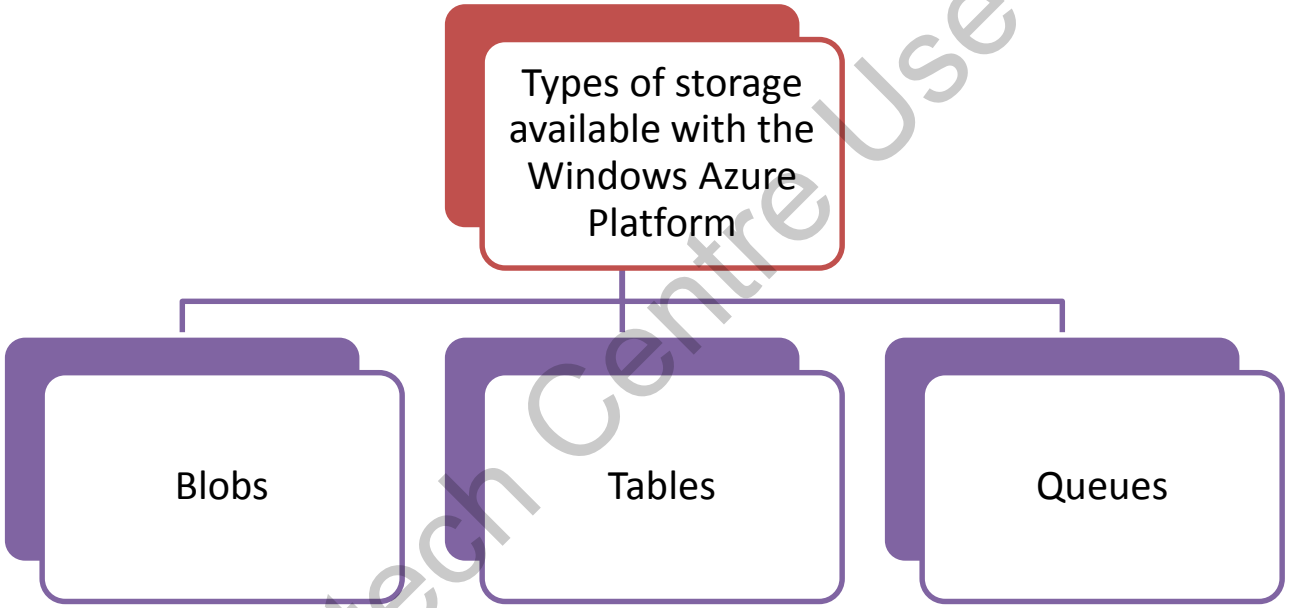


Components of Windows Azure 2-2

Following figure shows the detailed architecture of Windows Azure.



Windows Azure Storage



Windows Azure Storage-Windows Azure Blob Storage

- Windows Azure Blob Storage is a data storing mechanism that stores large amount of raw (binary or text) data.
- These data can be accessed by using Internet from anywhere in the world.
- Each blob can store up to hundreds of gigabytes of data.
- Each account can hold many blobs which account to data storage up to 200 TB.

Some of the scenarios where you can make use of Windows Azure Blob storage are as follows:

When you want to deliver images or documents directly to a browser

When you want to store files that may be accessed across distributed networks

For live-streaming video and audio

Windows Azure Storage-Windows Azure Table Storage

- Windows Azure Table storage is a means to store structured data. Table storage is ideal for the applications that use large amount of structured data.
- Windows Azure tables do not work similar to relational database tables as they do not support relationships or schemas.
- The Windows Azure Table storage service is a NoSQL data store that accepts authenticated calls from inside and outside the Windows Azure cloud.

The features of tables are as follows:

Tables store large amount of data that can be increased or decreased depending on the requirements of the applications.

Tables are ideal to store large data sets that do not require complex joins or relationships.

Data query is very fast as it uses a clustered index and keys.

Windows Azure Storage-Windows Azure Queue Storage

- Windows Azure Queue storage is a service for storing large numbers of messages that can be accessed from anywhere in the world via authenticated calls using HTTP or HTTPS.
- In the Windows Azure Queue storage, a single queue message can be up to 64 KB in size, and a queue can contain millions of messages, up to the total capacity limit of a storage account.

You can use queues in the following scenarios:

You need to store over 5 GB worth of messages in a queue, where the messages may have a lifespan shorter than 7 days.

Windows Azure Queues allows messages to have a very short lease time, so that if a Web worker crashes, the message can be processed again quickly.

Your application wants to track progress for processing inside of a message when the message arrives. This is useful if the Web worker working on a message crashes.

Windows Azure Storage-Restricting Access to Windows Azure Storage

- Windows Azure has been designed in such a manner that only the owner can access the blobs, tables, and queue inside the owned account.
- If a service or application needs to make these resources available to other clients without sharing the access key, one can have the following options for permitting access:

One can set a container's permissions to permit anonymous read access to the container and its blobs. However, this is not allowed for tables or queues.

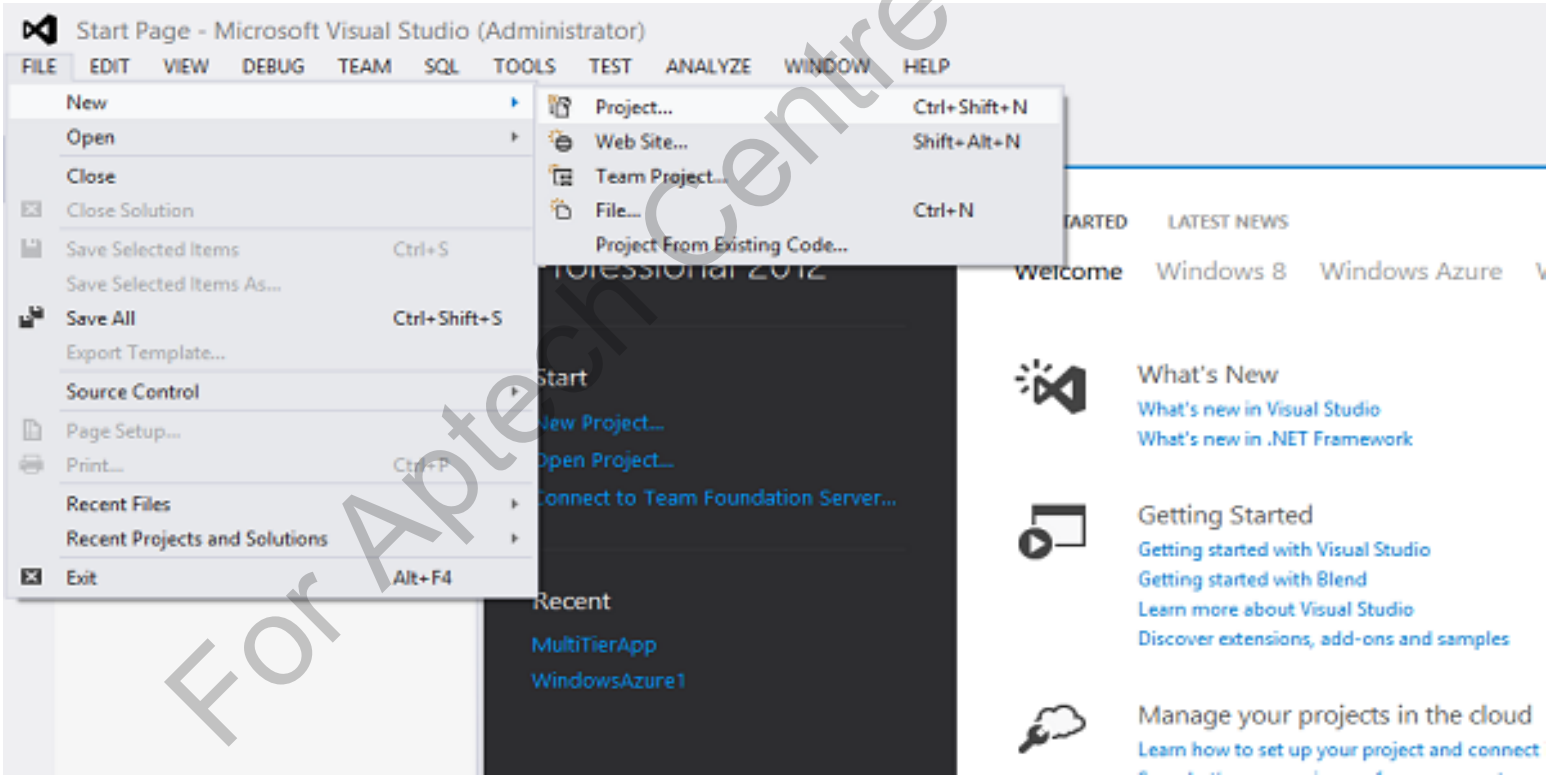
One can open a resource via a shared access signature, which enables to delegate restricted access to a container, blob, and table or queue resource by specifying the interval for which the resources are available and the permissions that a client will have to it.

One can use a stored access policy to manage shared access signatures for a container or its blobs, for a queue, or for a table.

Creating and Deploying a Cloud Services 1-15

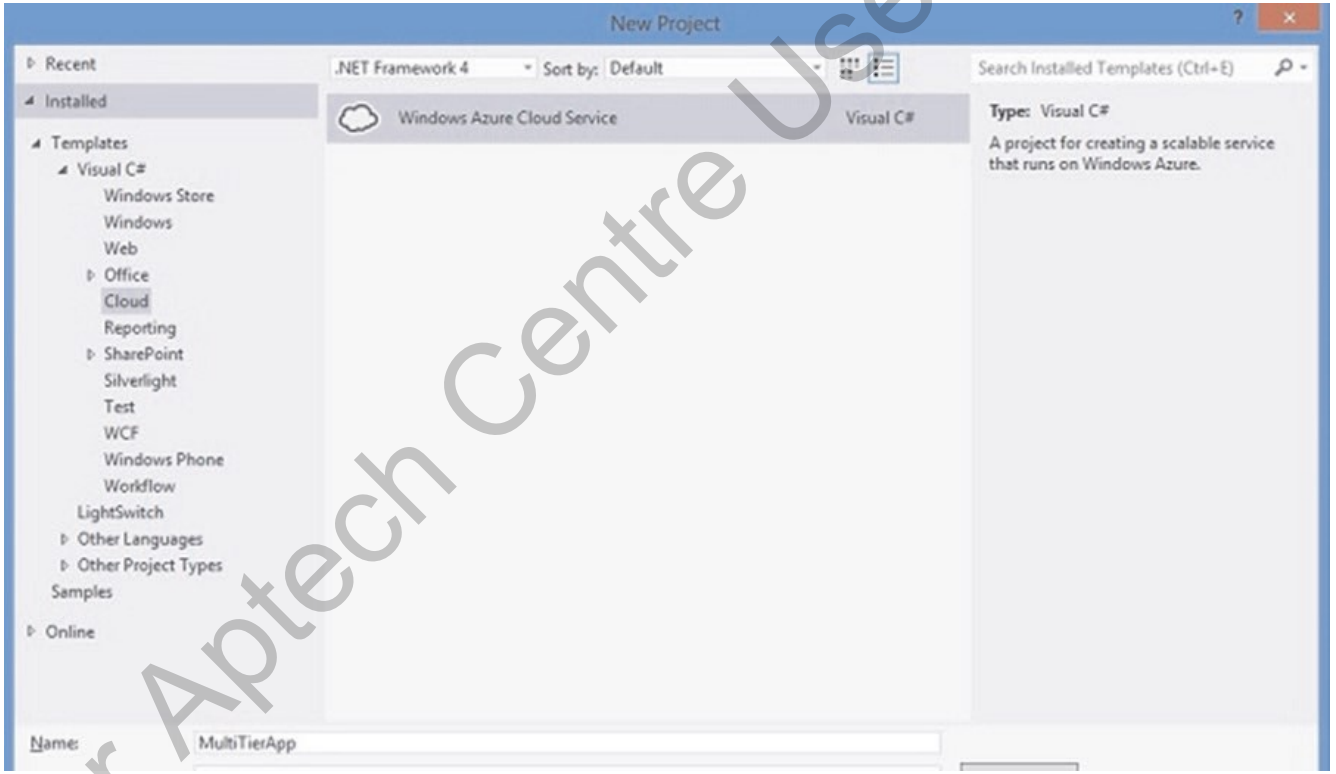
The following example shows the step-by-step procedure to build a small .NET Web application and deploy it to the cloud.

1. Launch **Visual Studio 2012**.
2. Click **File → New → Project** as shown in figure.



Creating and Deploying a Cloud Services 2-15

3. In the **Installed Templates** pane under Visual C#, click **Cloud** and then, click **`Windows Azure Cloud Service`** as shown in figure.

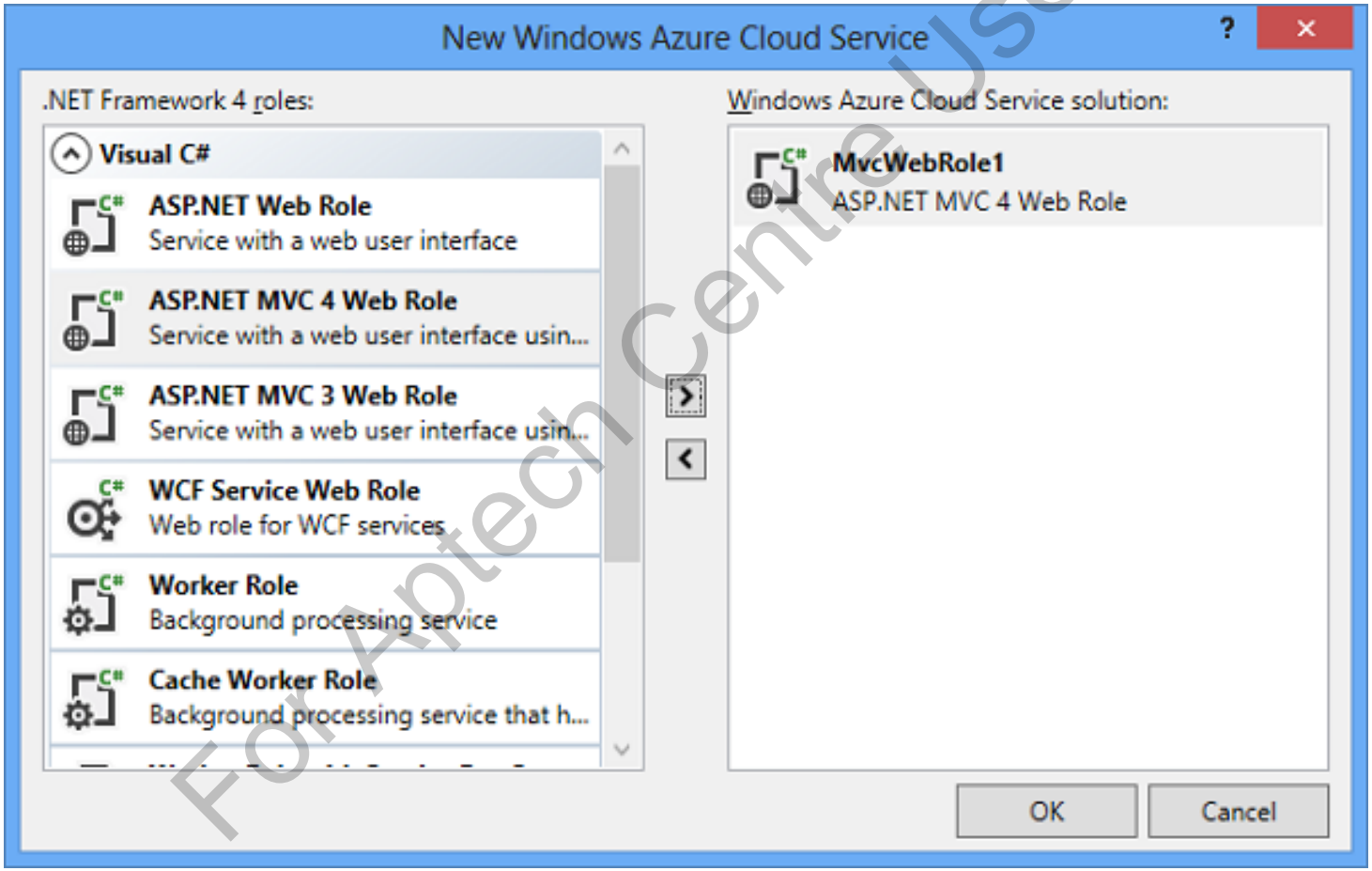


4. Name the project as **MultiTierApp**.

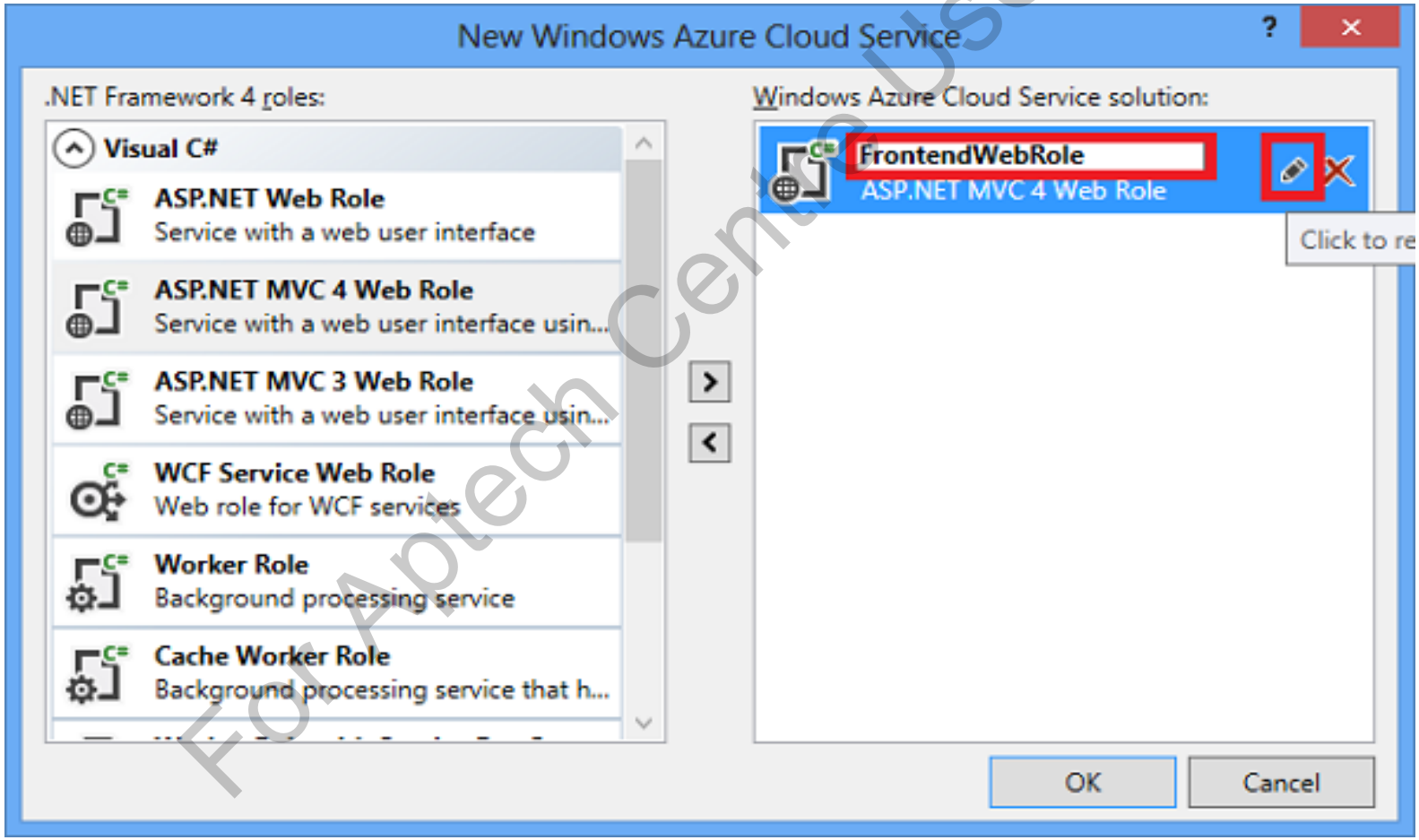
5. Click **OK.**

Creating and Deploying a Cloud Services 3-15

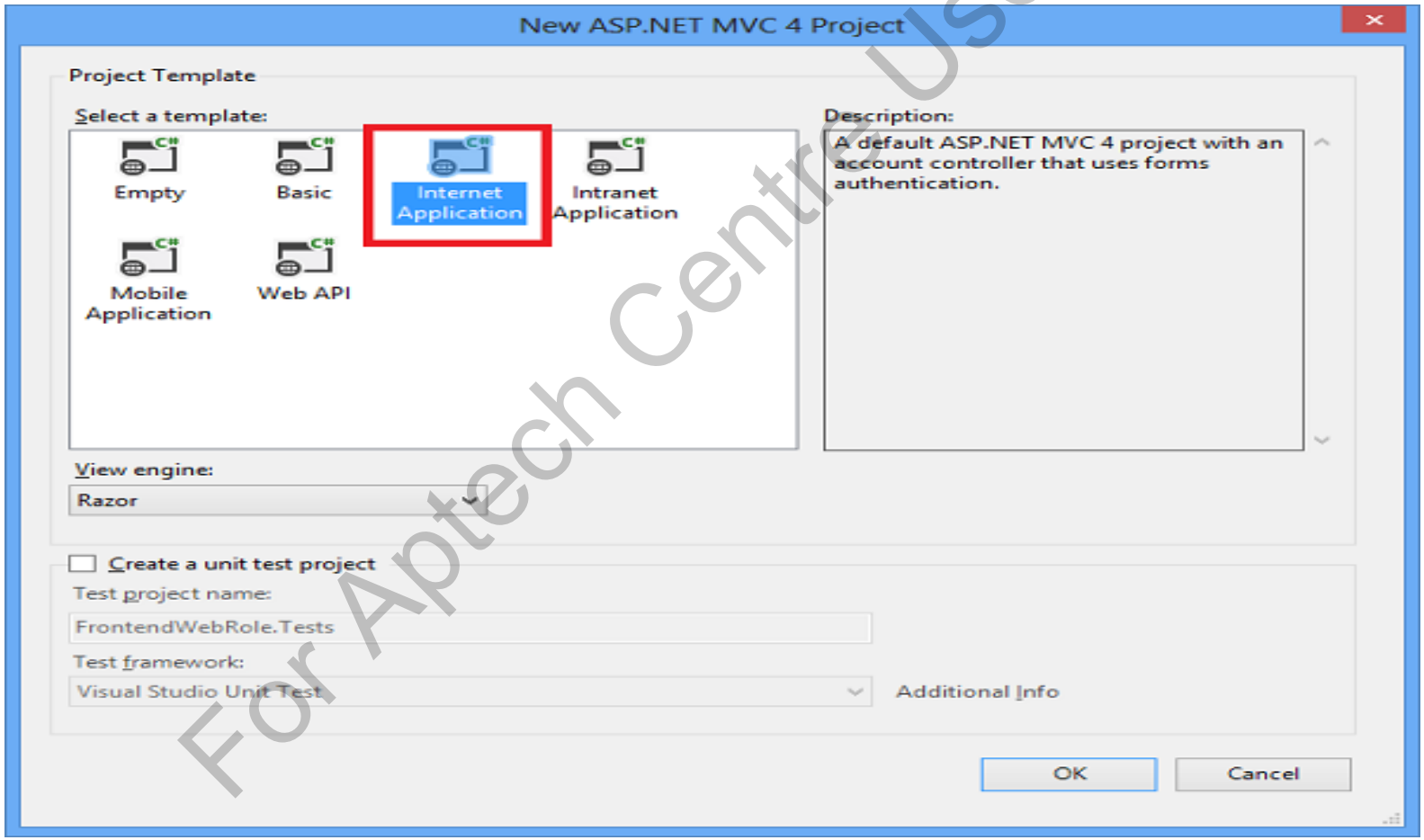
Now, the **New Windows Azure Cloud Service** window is displayed as shown in figure. This window shows .NET Framework 4 roles. Here, double-click **ASP.NET MVC 4 Web Role**.



6. To rename the role, hover over **MvcWebRole1** under Windows Azure Cloud Service solution. Then, click the **Pencil Icon**, and rename to **FrontendWebRole** as shown in figure.



7. Click **OK**. The **New ASP.NET MVC 4 Project** window appears as shown in figure. From the **Select a template** list, click **Internet Application.**



Creating and Deploying a Cloud Services 6-15

8. Click **OK**. The application will be successfully created. You can test in a browser. Right now, this application will run on a local server. To publish and run it on a cloud server, such as Windows Azure, you need to perform certain steps.

The following steps assume that you have already created a Windows Azure account/subscription.

9. Launch any Web browser.

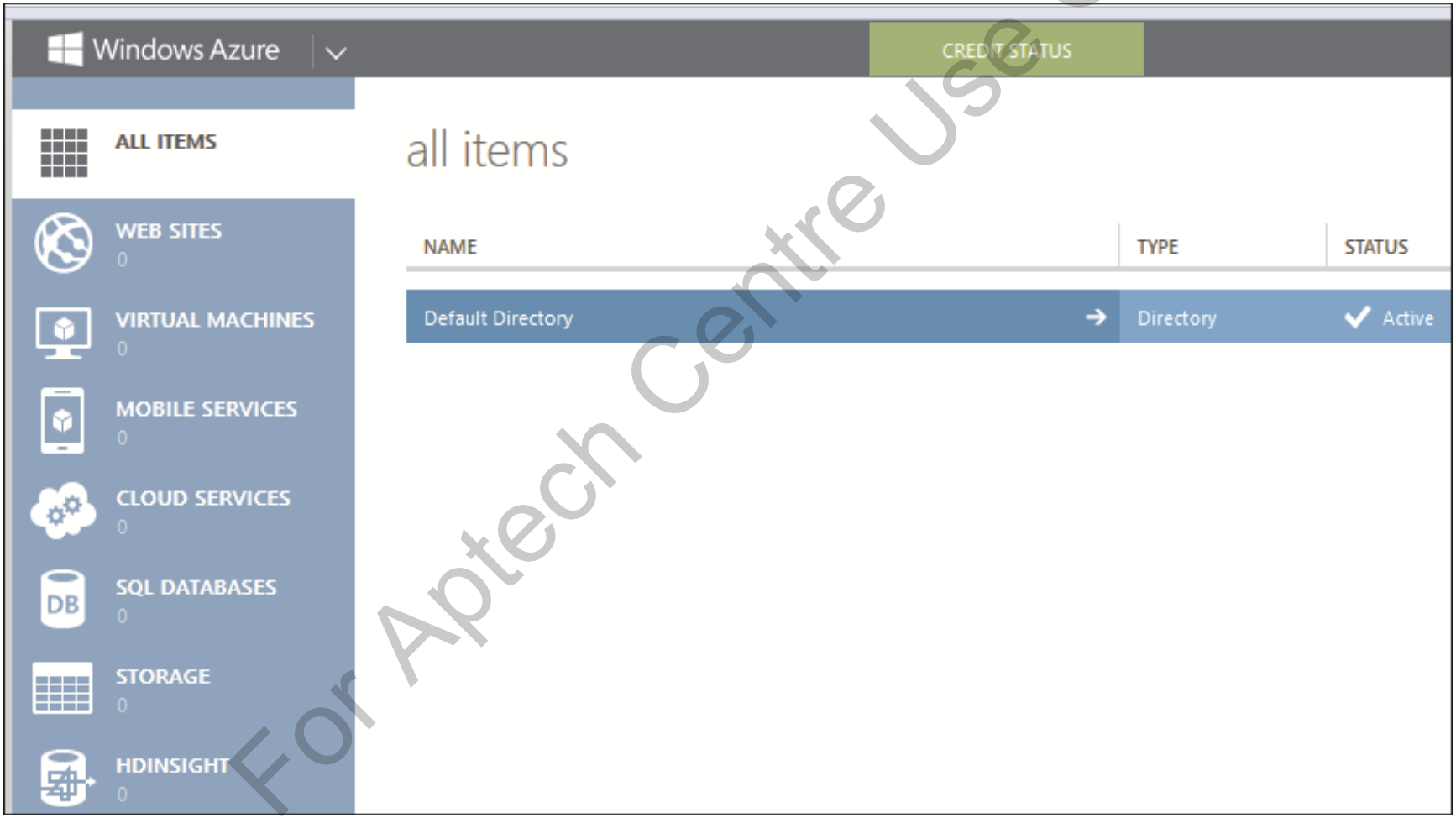
10. Type **`https://manage.windowsazure.com`** in the browser's address bar. You write **https** because everything under the cloud has security built-in with it.

Creating and Deploying a Cloud Services 7-15

11. Login with your Microsoft ID which you had used to sign up for Windows Azure.

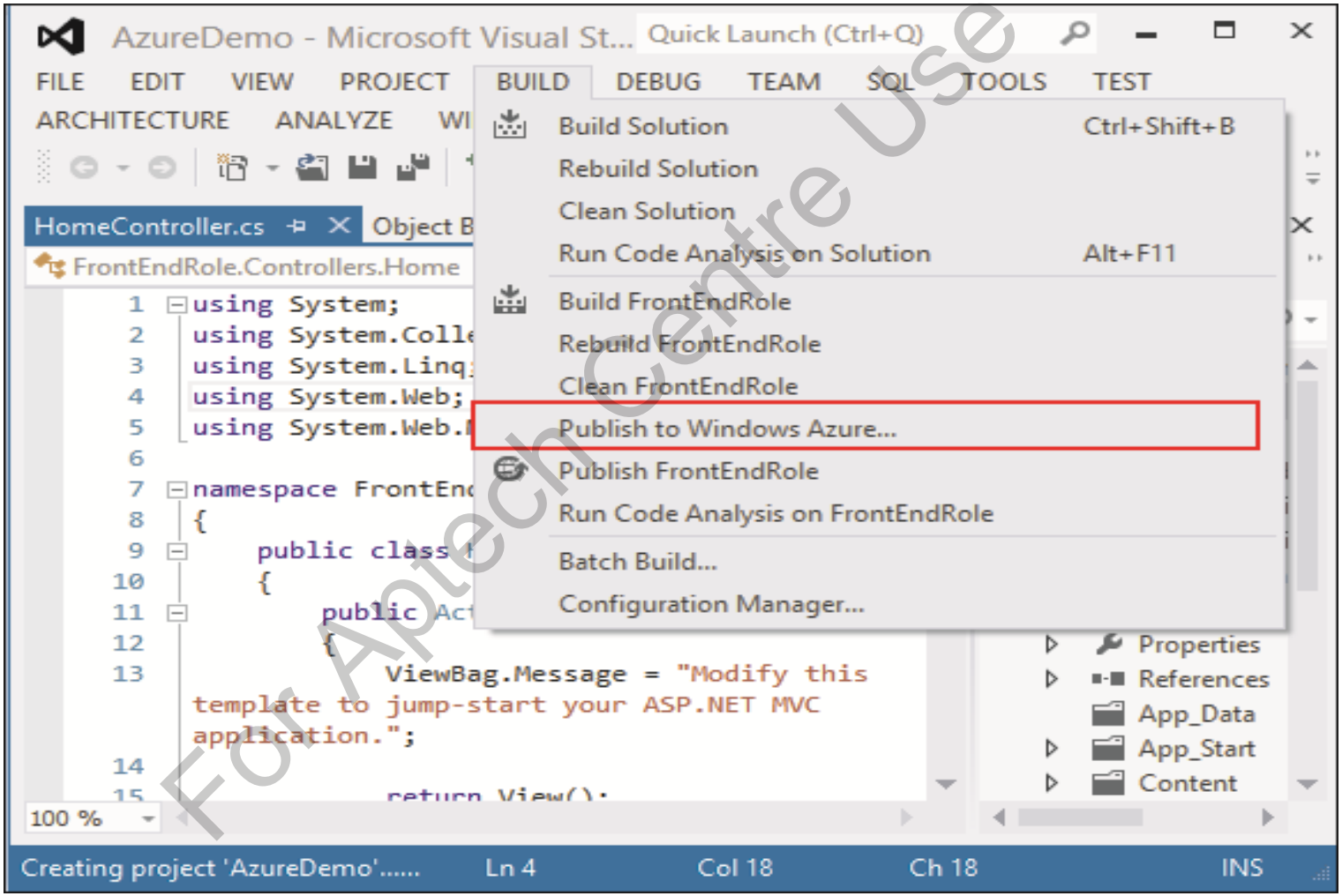
Creating and Deploying a Cloud Services 8-15

Upon successful login, you will see a dashboard resembling figure.



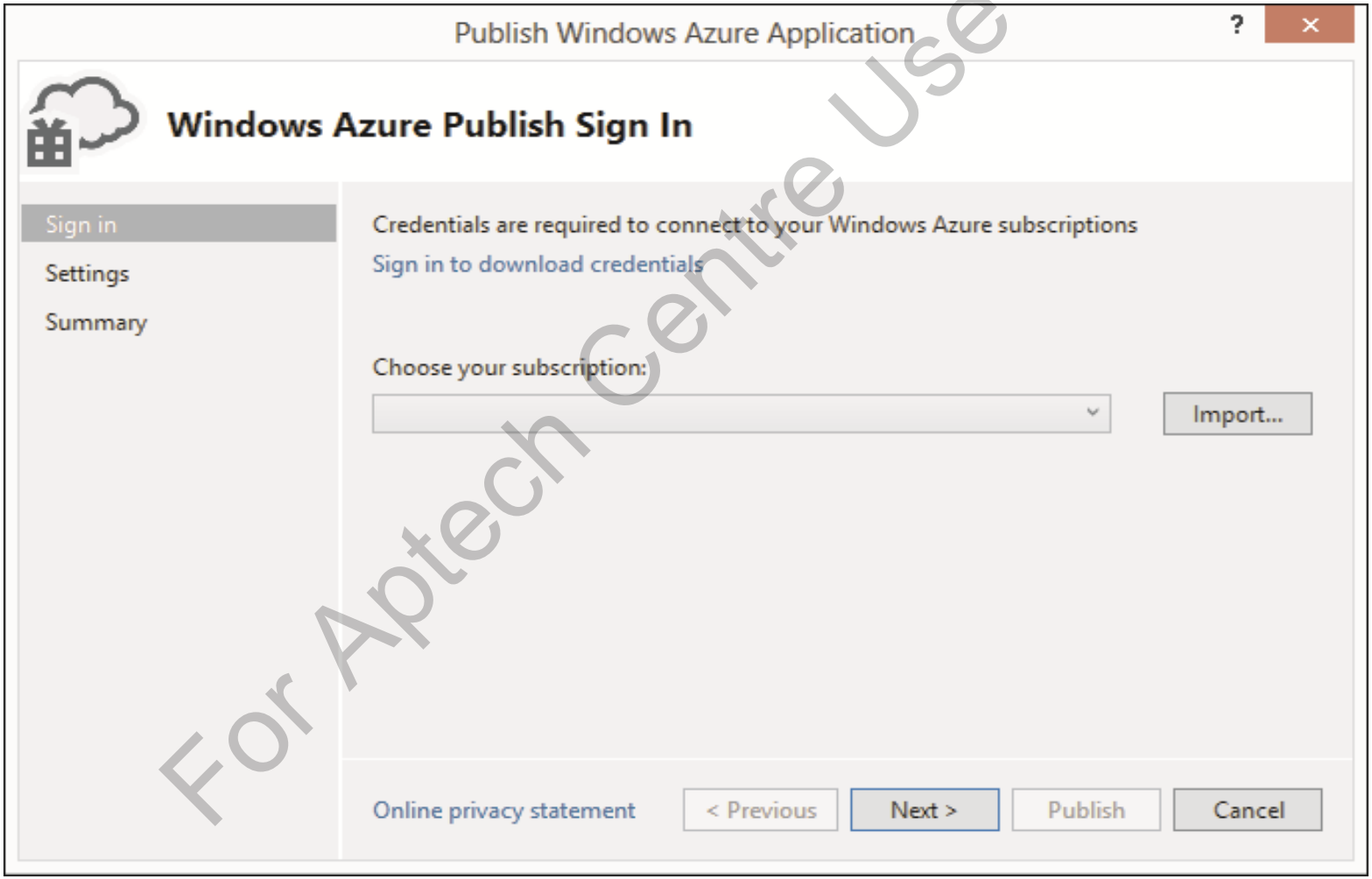
Creating and Deploying a Cloud Services 9-15

12. Go back to the Visual Studio IDE and click **Build→ Publish to Windows Azure** as shown in figure.



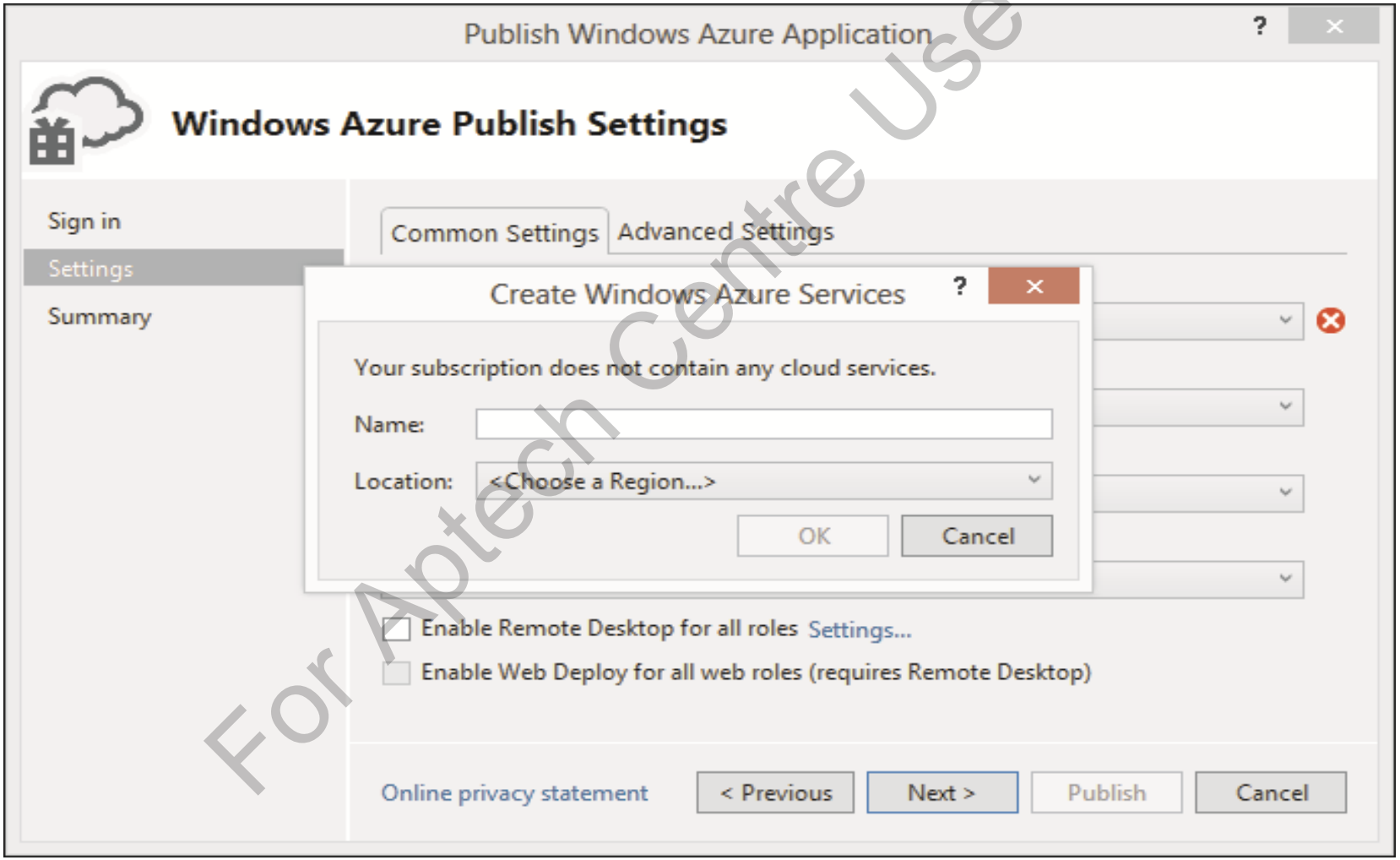
Creating and Deploying a Cloud Services 10-15

The **Publish Windows Azure Application** dialog box is displayed as shown in figure. Here, you can import your Windows Azure sign-in credentials.



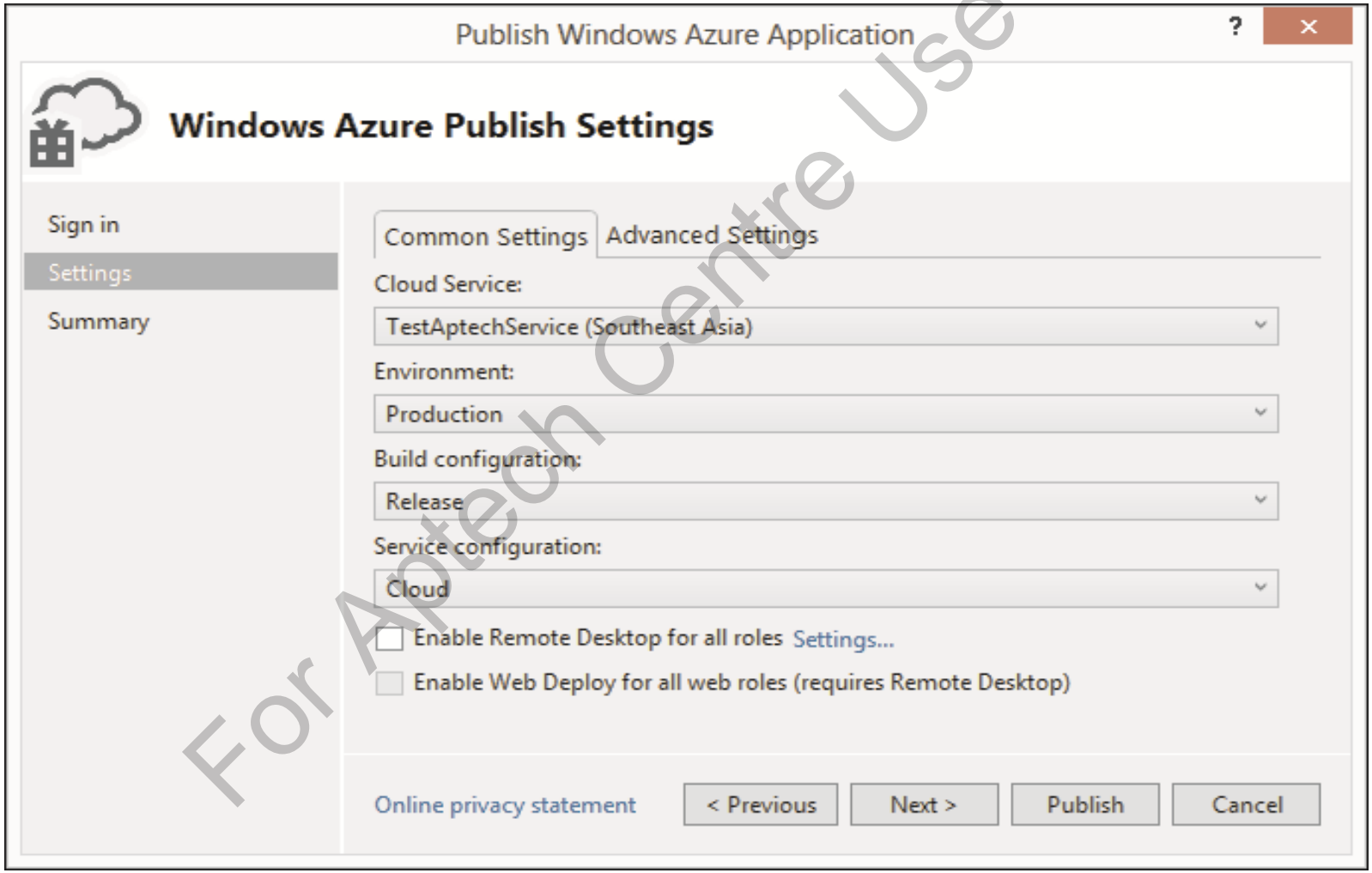
Creating and Deploying a Cloud Services 11-15

13. Click **Next**. The **Create Windows Azure Services** dialog is displayed, prompting you to enter a name for the cloud service and a region.

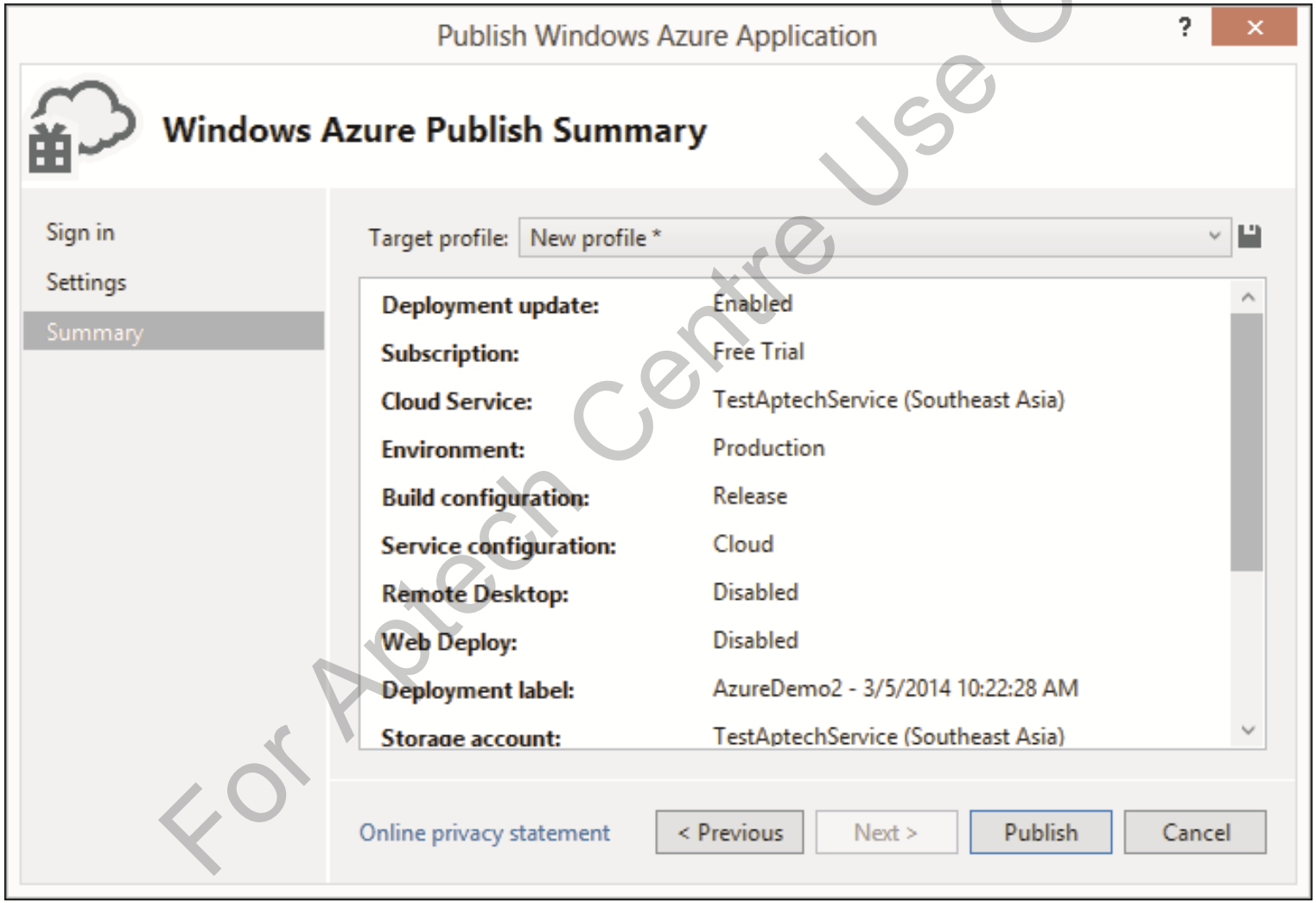


Creating and Deploying a Cloud Services 12-15

14. Type a suitable name and select a region. For example, in figure, **TestAptechService** is the service name and **Southeast Asia** is the region.

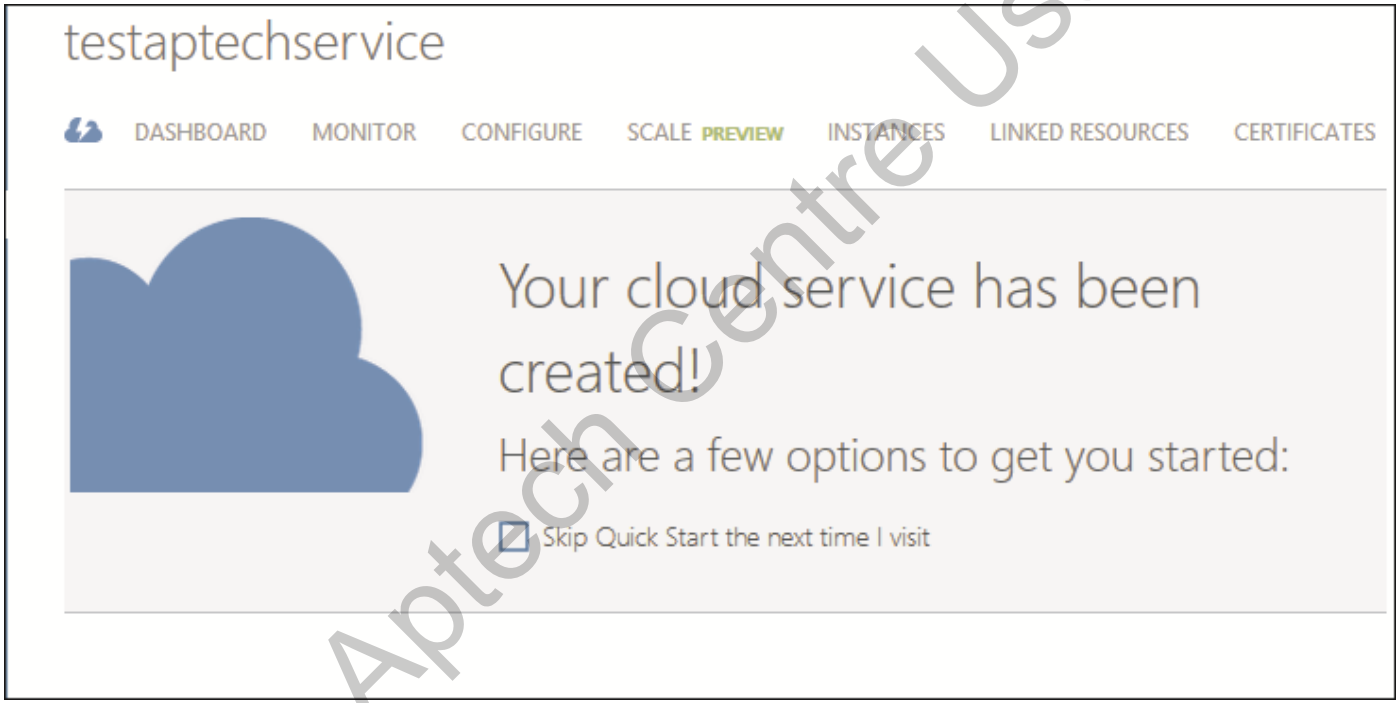


15. Click **Next**. The summary of all the publish settings is displayed as shown in figure.

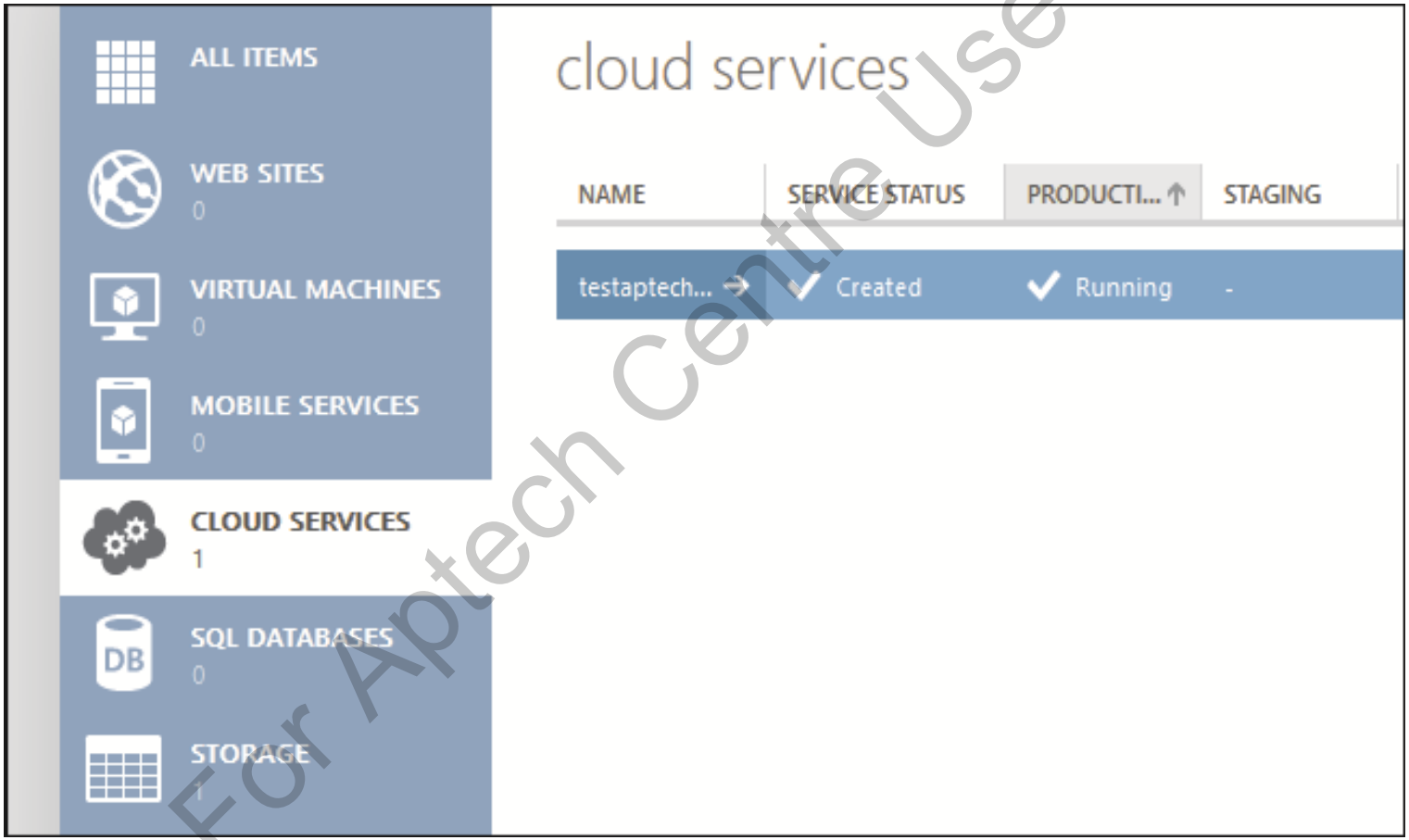


Creating and Deploying a Cloud Services 14-15

16. Click **Publish**. The Visual Studio 2012 IDE begins the publish and deploy action. In the Windows Azure portal, you will see a success message.



17. Click **CLOUD SERVICES** on the left pane. You will see the cloud service details in the portal.



SQL Database

Microsoft SQL Azure, now renamed as Windows Azure SQL Database, is a cloud based relational database service that leverages existing SQL Server technologies.

SQL Database enables users to perform relational queries, search operations, and synchronize data with mobile users and remote back offices.

SQL Azure can store and retrieve both structured and unstructured data. Both cloud based as well as on-premises applications can use the SQL Database.

Applications retrieve data from SQL Database through a protocol known as Tabular Data Stream (TDS). This protocol is not new to SQL Database.

Whenever on-premises applications involve interaction with SQL Server Database Engine, this protocol is used by the client and the server.

The SQL Data Sync technology is built on Microsoft Sync Framework and SQL Database.

SQL Database-Architecture of SQL Database

Microsoft Windows Azure SQL Database is hosted on servers running SQL server technologies. There are four layers that provide a relational database. They are as follows:

The Client Layer

- The Client layer communicates directly with SQL database through your application. It may either reside locally in your data center or be hosted on the cloud in Windows Azure.

The Services Layer

- It acts as a gateway between the client layer and the platform layer. The Services layer is responsible for managing all the connections routing between your application and the various physical servers where data resides.

The Platform Layer

- The Platform layer includes the physical servers and various services that support the services layer. This layer comprises many instances of SQL Server and each of these is managed by the SQL Database fabric, which is a distributed computing system of tightly integrated networks, servers, and storage.

Infrastructure Layer

- The Infrastructure layer indicates the IT administration of the infrastructure such as physical hardware and operating systems supporting the services layer.

SQL Database-Differences between SQL Server and Windows Azure SQL Database

In a local SQL server environment, when preparing an on-premises deployment, it may be the role of the DBA or IT department to prepare and configure the required hardware and software.

In Windows Azure SQL Database, the logical administration is separated from the physical administration.

In SQL Server, many Transact-SQL Statements have parameters that allow you to specify file groups or physical file paths.

SQL Database does not support any type of parameters because they have dependencies on the physical configuration.

SQL Database does not support all of the features and data types found in SQL Server.

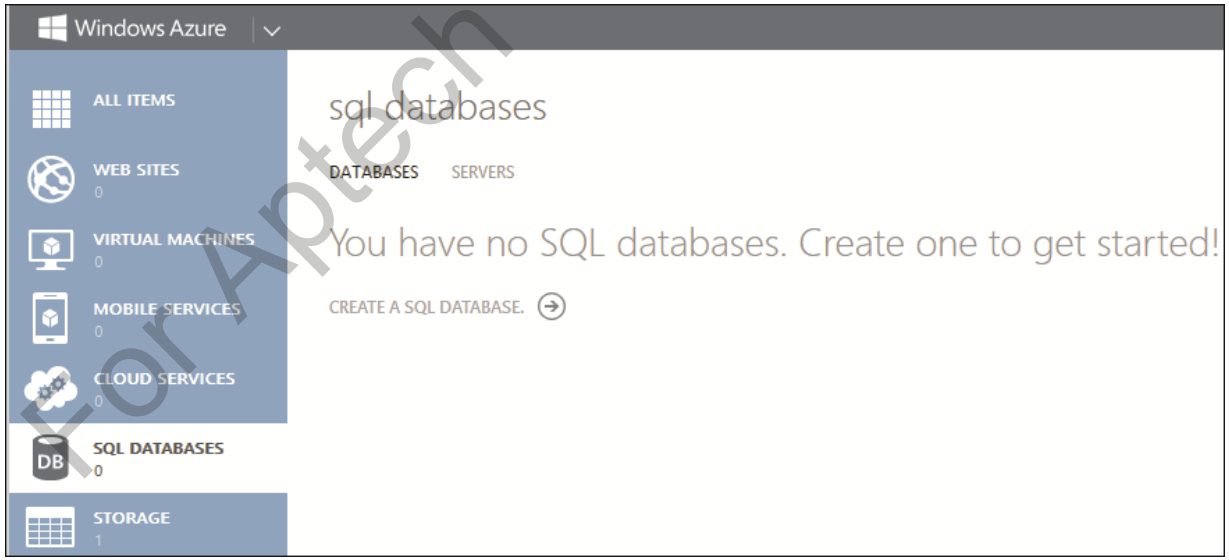
SQL Database offers the scale and functionality required in an enterprise data center, without the administrative overheads that would be normally incurred.

SQL Database-Creating Databases and Tables 1-11

- In order to begin creating a SQL database, the user must first create and activate a Windows Azure subscription.
- The user must be connected to the database in order to use the `CREATE DATABASE`, `ALTER DATABASE`, or `DROP DATABASE` statements.

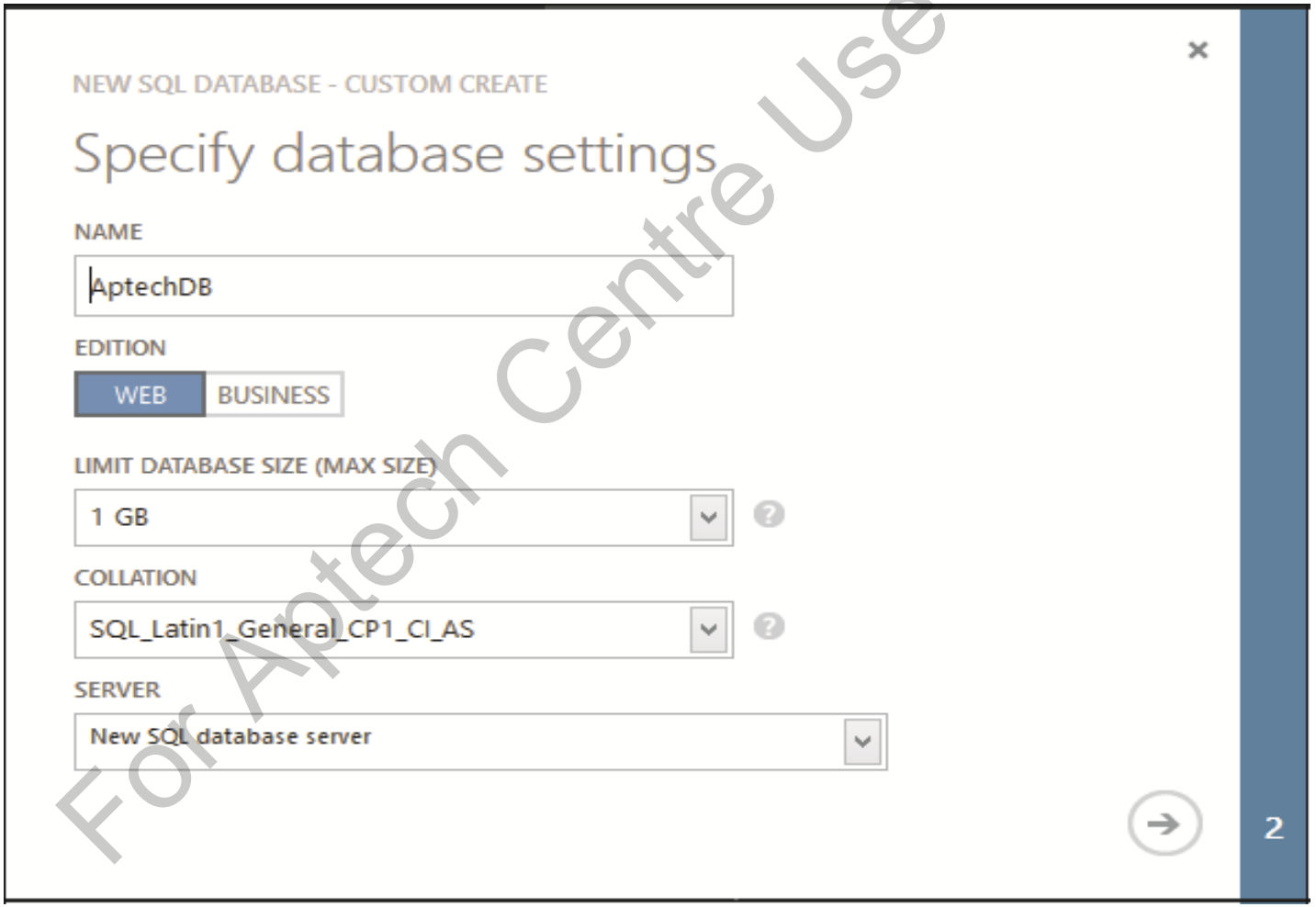
To create a SQL Database:

1. Visit the Windows Azure Platform Management Portal and login with a Windows Live ID account that has been associated with the Windows Azure Platform subscription.
2. Click **SQL DATABASES** on the left.

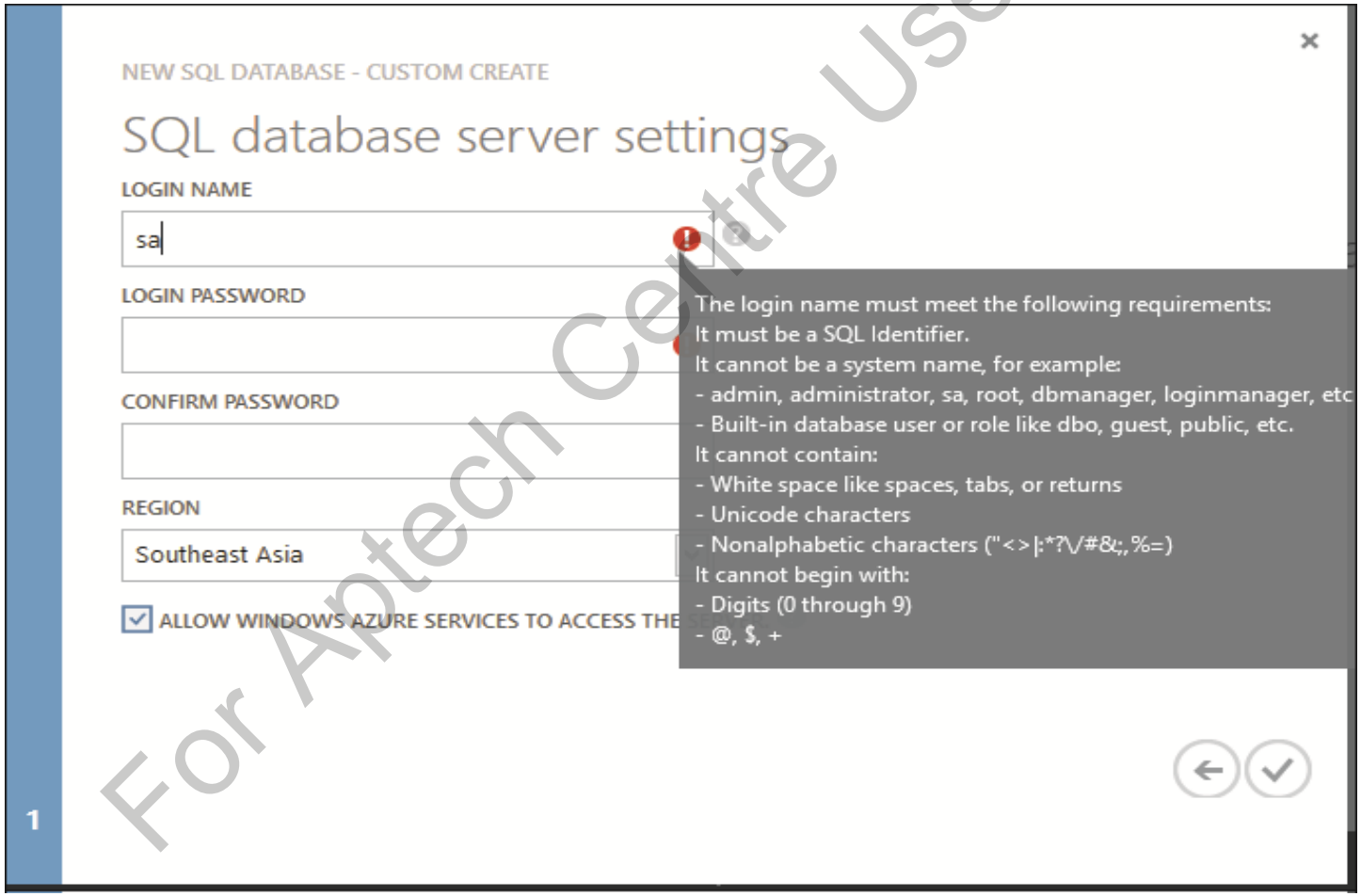


SQL Database-Creating Databases and Tables 2-11

3. Click **CREATE A SQL DATABASE**. You will be asked to specify the database details as shown in figure. Specify a suitable name.

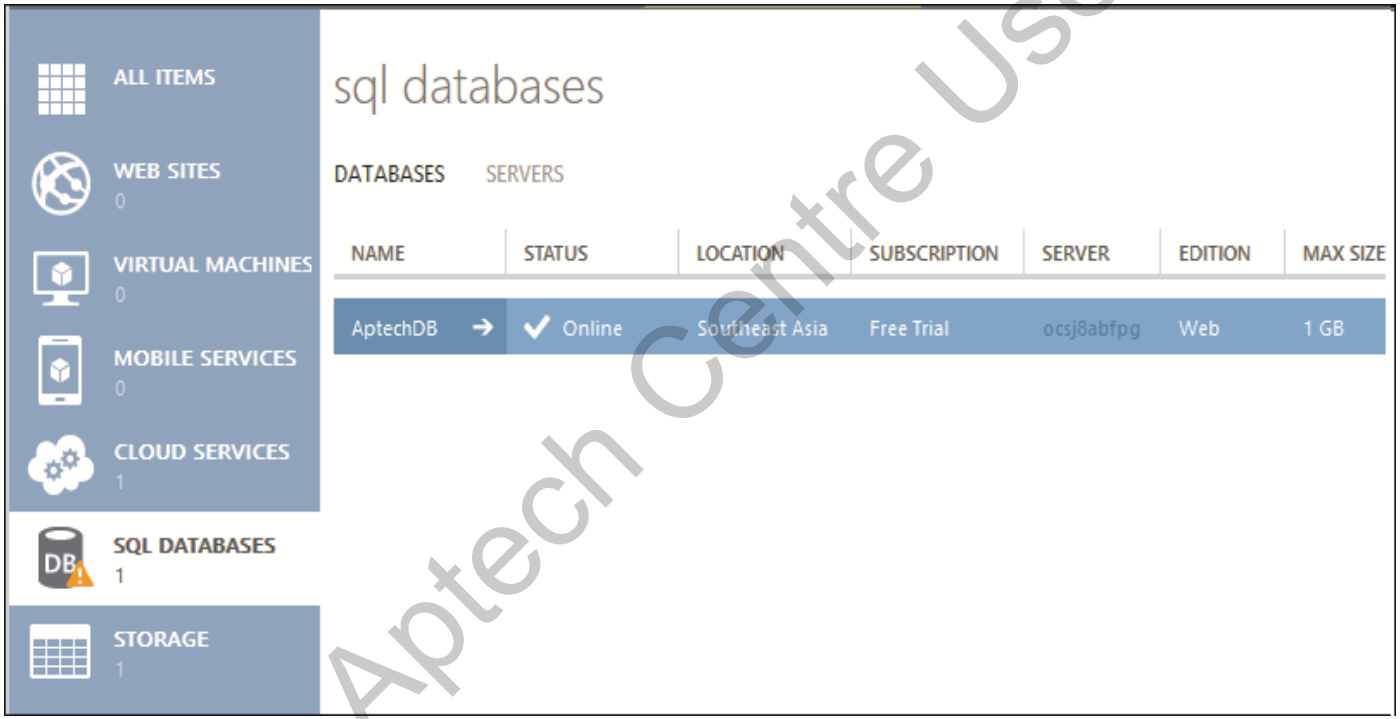


4. Click the arrow at the bottom of the pane to proceed further. Next, you will be asked to specify login credentials that you wish for the cloud database account. Note that as shown in figure, you have to follow certain naming conventions.



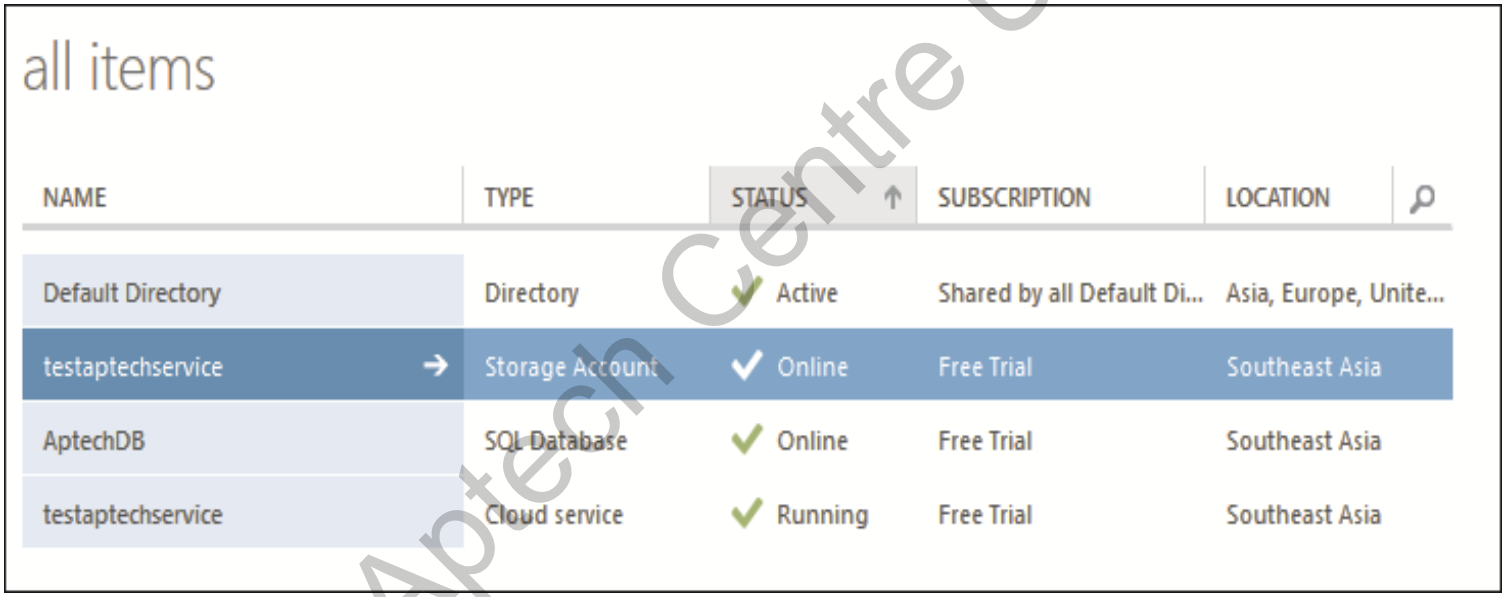
SQL Database-Creating Databases and Tables 4-11

5. Click the check mark arrow to complete the creation of the database. Once this is done, the database is successfully created. Figure shows the database created.



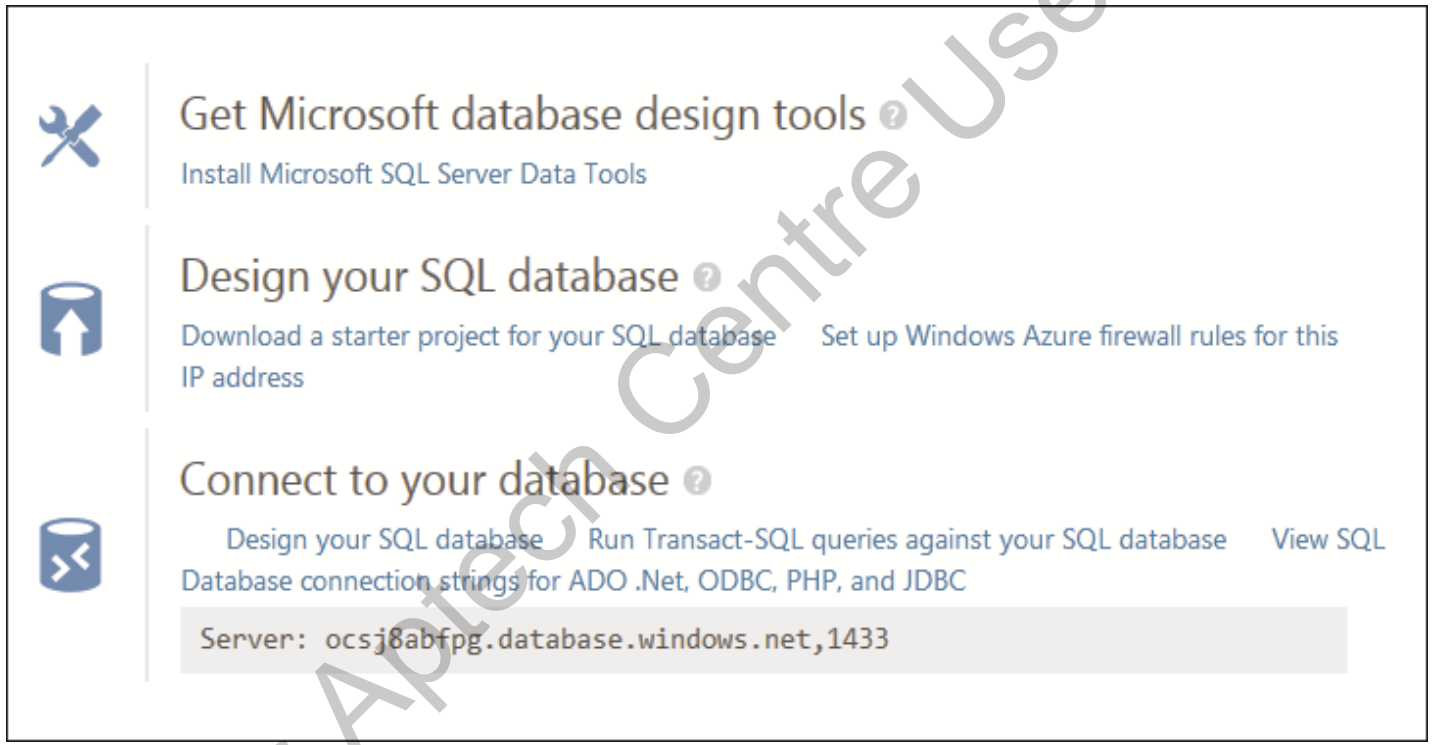
SQL Database-Creating Databases and Tables 5-11

If you check the main dashboard on the portal, you will see three items: the cloud service, the storage (which was automatically created when you created a cloud service), and the cloud database. Figure shows this.



SQL Database-Creating Databases and Tables 6-11

6. Click the database name, AptechDB. You will see the various tasks that you can perform with your database. Figure shows these.

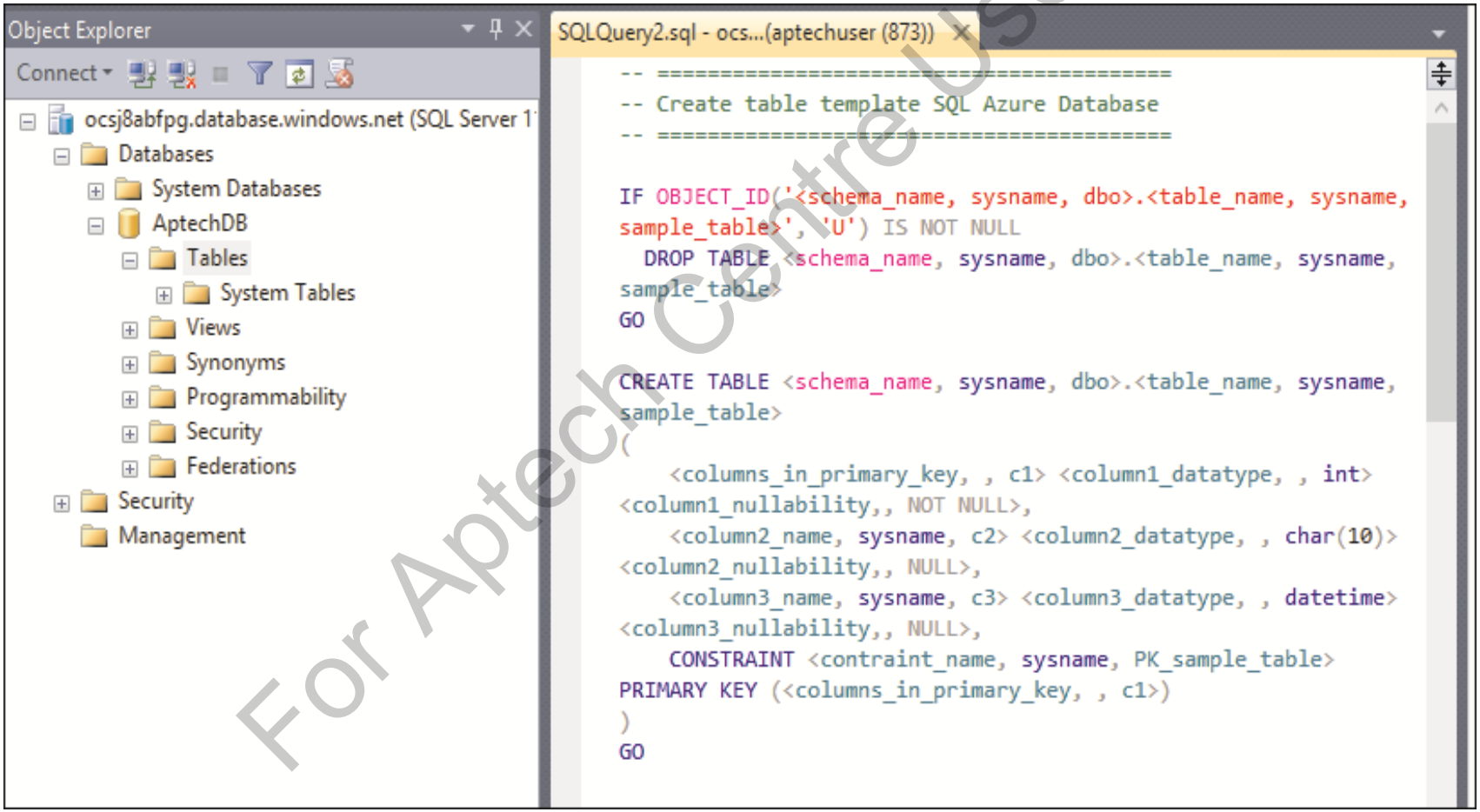


SQL Database-Creating Databases and Tables 7-11

7. Launch SSMS.
8. In the login credentials box, specify the cloud account credentials as shown in figure.

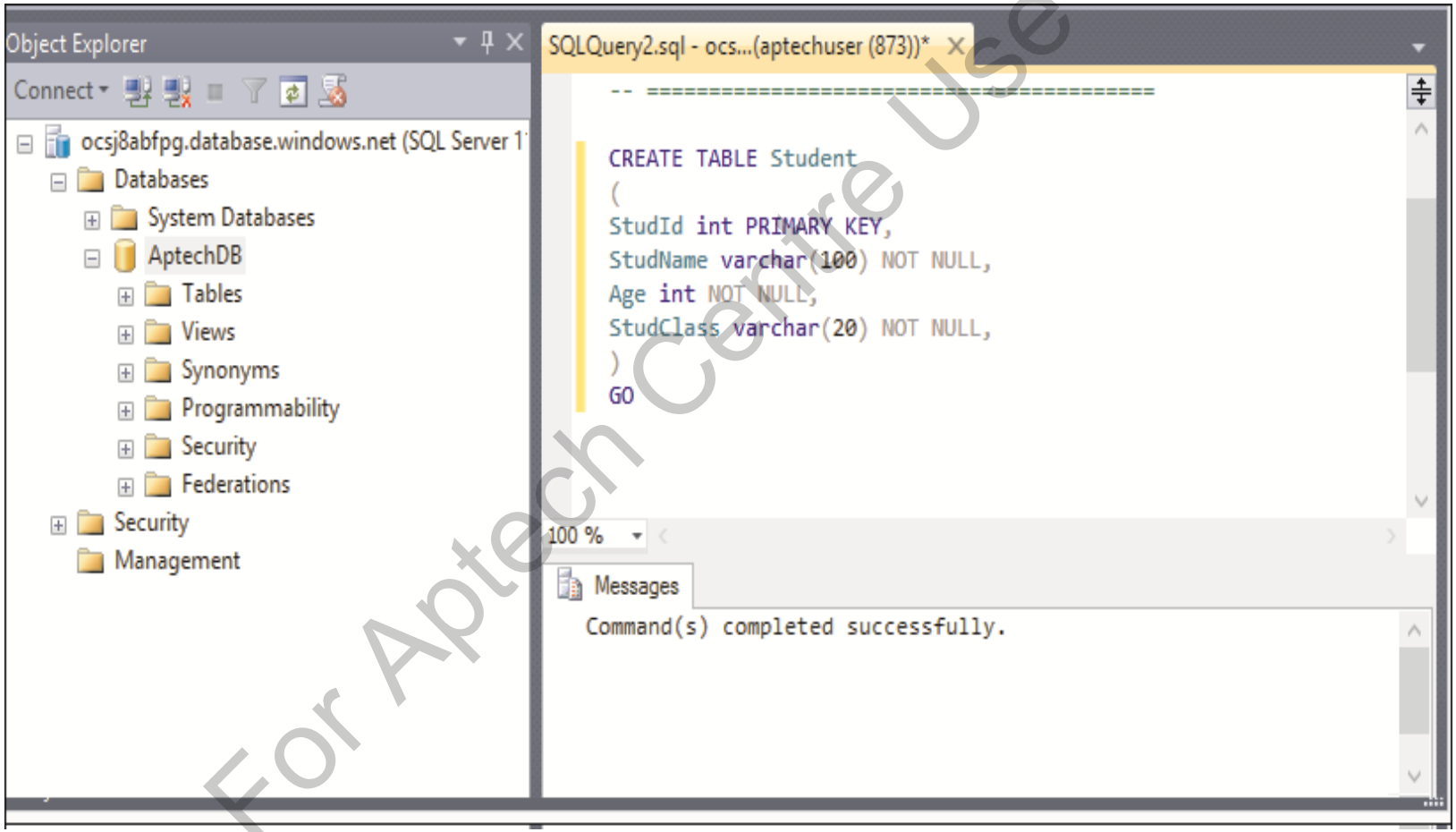
SQL Database-Creating Databases and Tables 8-11

9. Right-click **Tables** and select **New Table**. The default script for creating a new cloud based table is displayed on the right as shown in figure.



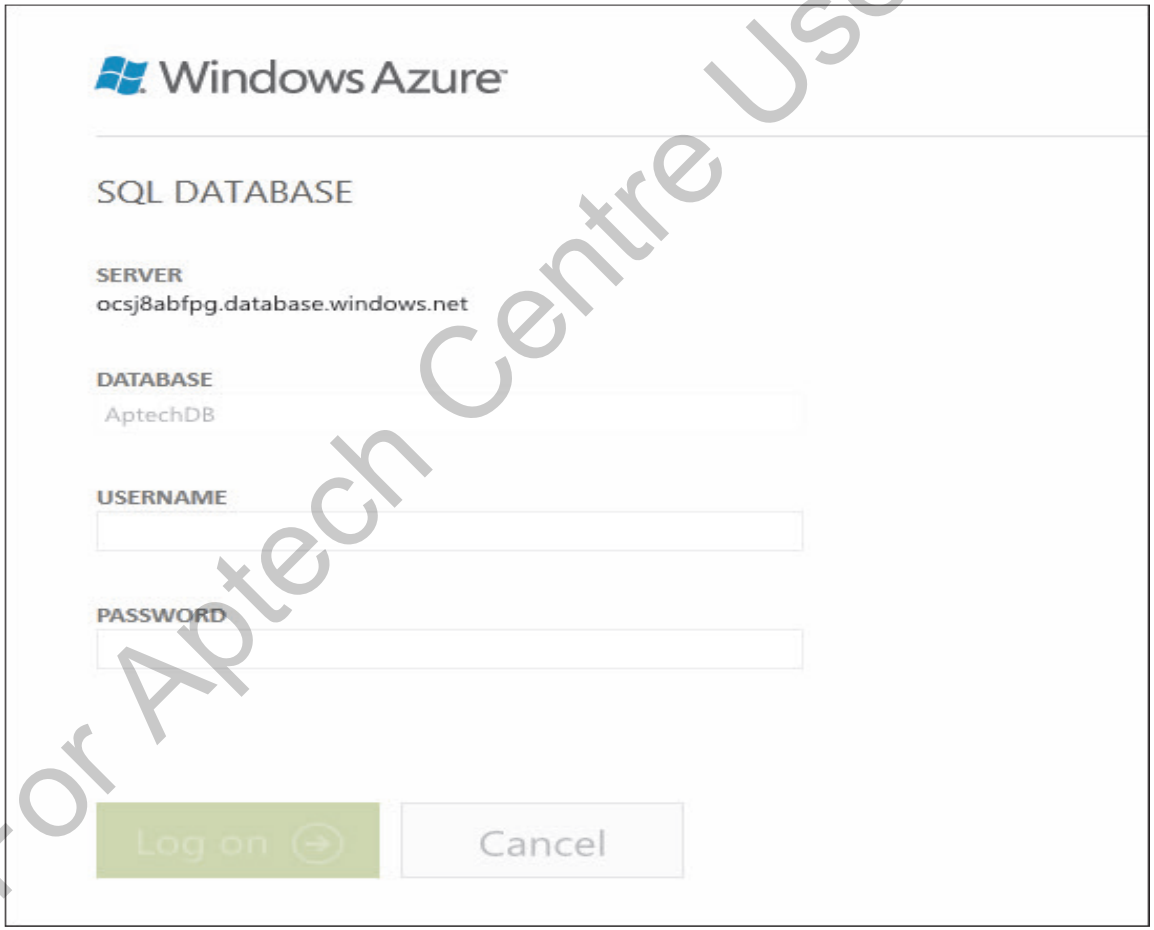
SQL Database-Creating Databases and Tables 9-11

10. Edit the code as shown in figure and then click **Execute** to create a table named **Student**.



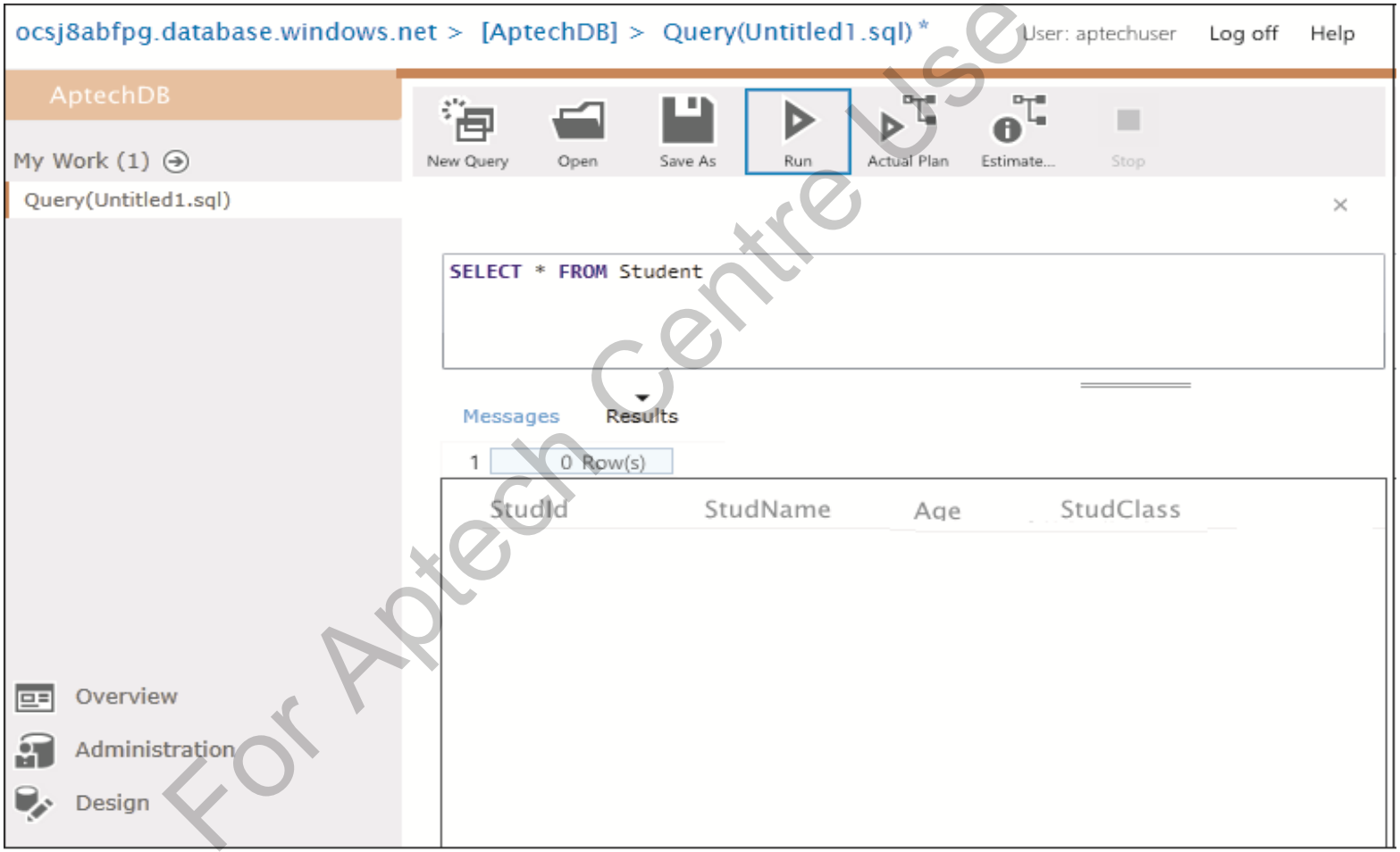
SQL Database-Creating Databases and Tables 10-11

11. To run queries against this table, go to the Windows Azure portal and click **Run Transact-SQL queries against your SQL database**. You will be asked to provide cloud database server login credentials as shown in figure.



SQL Database-Creating Databases and Tables 11-11

On successful login, you will see a new screen where you can type and execute Transact-SQL queries on cloud-based tables. Figure shows an example of a query.



Windows Azure AppFabric

In earlier versions of Windows Azure, one of the key components of its architecture was Windows Azure AppFabric.

However, Windows Azure AppFabric is no longer called by that name and instead, has become a part of Windows Azure 'Client Libraries for .NET'.

It represents a set of middleware services that aim to increase interoperability between the components of your software solution. It contains technologies such as Service Bus, Caching, Multi-Factor Authentication, and so on.

Windows Azure AppFabric-Service Bus 1-2

Regardless of whether an application is running in the cloud or on-premises, it may need to interact with other applications or software.

To provide a way to do this, Windows Azure offers Service Bus as a messaging solution for applications.

The Service Bus sits between components of a cloud app or between a cloud and on-premises applications and enables exchange of messages.

Service Bus is a multi-tenant cloud service, indicating that the service is shared by multiple users. Each user will create a *namespace and* then, define the communication mechanisms he/she needs within that namespace.

Windows Azure AppFabric-Service Bus 2-2

Within a namespace, users can have one or more instances of the following communication mechanisms:

Queues	• They allow one-directional communication. Each queue acts as a middleman or broker to store sent messages until they are received.
Topics	• They provide one-directional communication using subscriptions. Like a queue, a topic acts as a middleman or broker, but allows each subscription to see only those messages that match specific criteria.
Relays	• They provide bi-directional communication. However, unlike queues and topics, a relay does not store in-flight messages – it does not behave like a broker. A relay just passes the messages to the destination application.

Windows Azure AppFabric-Multi-Factor Authentication

Multi-Factor authentication enables additional authentication for cloud and on-premises applications.

Users must additionally authenticate through a mobile app or by responding to an automated text message or phone call before access is given.

It is implemented by enforcing any two or more of the following verification methods:

- Something the users know
- Something the users possess
- Something you are (such as individual biometrics)

Windows Azure Cache

Windows Azure Cache is another key part of Windows Azure 'Client Libraries for .NET'.

It is a distributed, in-memory, scalable solution that gives you super-fast access to data.

Cache temporarily stores information from other backend sources and thus increases performance.

Cache can reduce the costs of a Windows Azure project and raise the scalability of other storage services such as SQL Database or Azure storage.

Summary

- Windows Azure is a cloud computing platform and infrastructure that not only hosts applications but also enables you to build, deploy, and manage them.
- Windows Azure platform offers several powerful features and benefits.
- Windows Azure platform architecture includes several components, of which the following are most important: Compute, Data Services, and App Services.
- Windows Azure platform supports three types of storage: blobs, tables, and queues.
- Windows Azure has been designed in such a manner that only the owner can access the blobs, tables, and queues inside the owned account.
- Windows Azure SQL Database is a cloud based relational database service that leverages existing SQL Server technologies.
- SQL Database supports connection through SSMS to the cloud database.