DavidChappell
& Associates

INTRODUCING WINDOWS AZURE

DAVID CHAPPELL

OCTOBER 2010

SPONSORED BY MICROSOFT CORPORATION

CONTENTS

An Overview of Windows Azure 2

Compute 4

Storage 5

Fabric Controller 7

Content Delivery Network 8

Connect 9

Using Windows Azure: Scenarios 11

Creating a Scalable Web Application 11

Creating a Parallel Processing Application 12

Creating a Scalable Web Application with Background Processing 13

Creating a Web Application with Relational Data 14

Migrating an On-Premises Web Application with Relational Data 15

Using Cloud Storage from an On-Premises or Hosted Application 16

Understanding Windows Azure: A Closer Look 17

Creating Windows Azure Applications 17

Examining the Compute Service 18

Examining the Storage Service 19

Blobs 19

Tables 20

Queues 21

Examining the Fabric Controller 23

Looking Ahead 24

Conclusions 24

For Further Reading 25

About the Author 25

AN OVERVIEW OF WINDOWS AZURE

Cloud computing is here. Running applications and storing data on machines in an Internet-accessible data center can offer plenty of advantages. Yet wherever they run, applications are built on some kind of platform. For on-premises applications, such as those running inside an organization's data center, this platform usually includes an operating system, some way to store data, and perhaps more. Applications running in the cloud need a similar foundation.

The goal of Windows Azure is to provide this. Part of the larger Windows Azure platform, Windows Azure is a foundation for running applications and storing data in the cloud. Figure 1 illustrates this idea.

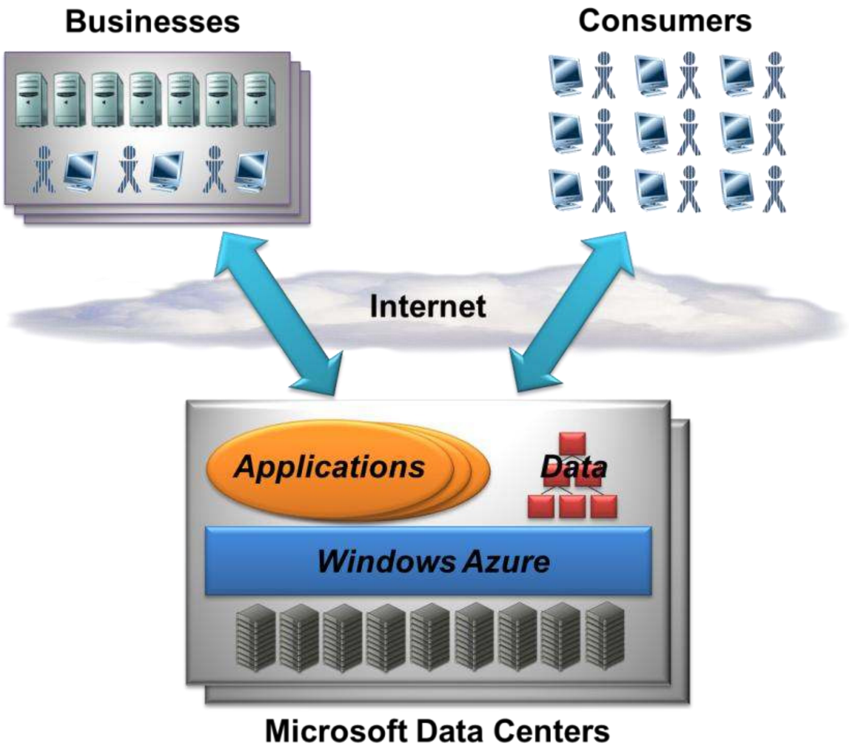

Figure 1: Windows Azure applications run in Microsoft data centers and are accessed via the Internet.

Rather than providing software that Microsoft customers can install and run themselves on their own computers, Windows Azure today is a service: Customers use it to run applications and store data on Internet-accessible machines owned by Microsoft. Those applications might provide services to businesses, to consumers, or both. Here are some examples of the kinds of applications that can be built on Windows Azure:

- An independent software vendor (ISV) could create an application that targets business users, an approach that's often referred to as *Software as a Service (SaaS).* Windows Azure was designed in part to support Microsoft's own SaaS applications, so ISVs can also use it as a foundation for a variety of business-oriented cloud software.

- An ISV might create a SaaS application that targets consumers rather than businesses. Because Windows Azure is intended to support very scalable software, a firm that plans to target a large consumer market might well choose it as a platform for a new application.

- Enterprises might use Windows Azure to build and run applications that are used by their own employees. While this situation probably won't require the enormous scale of a consumer-facing application, the reliability and manageability that Windows Azure offers could still make it an attractive choice.

To support cloud applications and data, Windows Azure has five components, as Figure 2 shows.

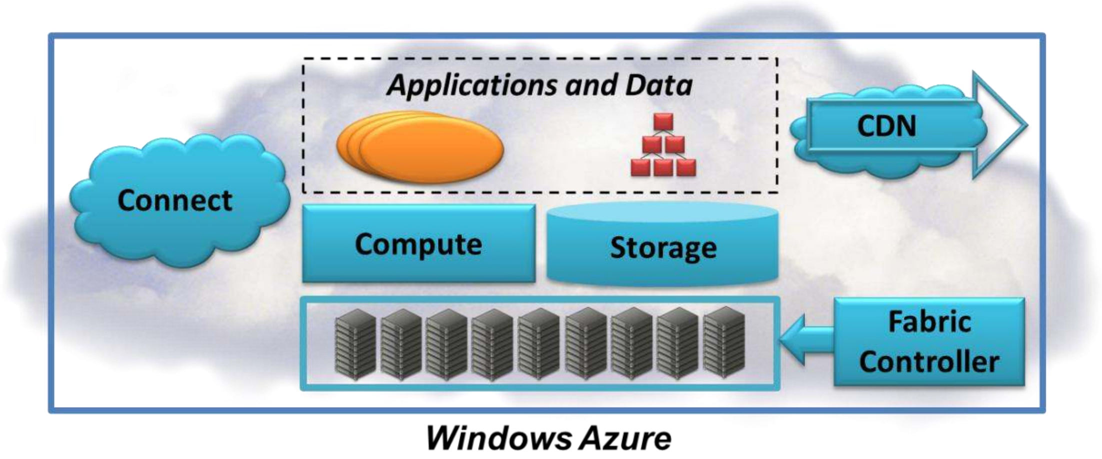

Figure 2: Windows Azure has five main parts: Compute, Storage, the Fabric Controller, the CDN, and Connect.

Those components are:

- Compute: runs applications in the cloud. Those applications largely see a Windows Server environment, although the Windows Azure programming model isn't exactly the same as the on-premises Windows Server model.

- Storage: stores binary and structured data in the cloud.

- Fabric Controller: deploys, manages, and monitors applications. The fabric controller also handles updates to system software throughout the platform.

- Content Delivery Network (CDN): speeds up global access to binary data in Windows Azure storage by maintaining cached copies of that data around the world.

- Connect: allows creating IP-level connections between on-premises computers and Windows Azure applications.

The rest of this section introduces each of these technologies.

COMPUTE

Windows Azure compute can run many different kinds of applications. Whatever an application does, however, it must be implemented as one or more *roles*. Windows Azure then typically runs multiple *instances* of each role, using built-in load balancing to spread requests across them. Figure 3 shows how this looks.

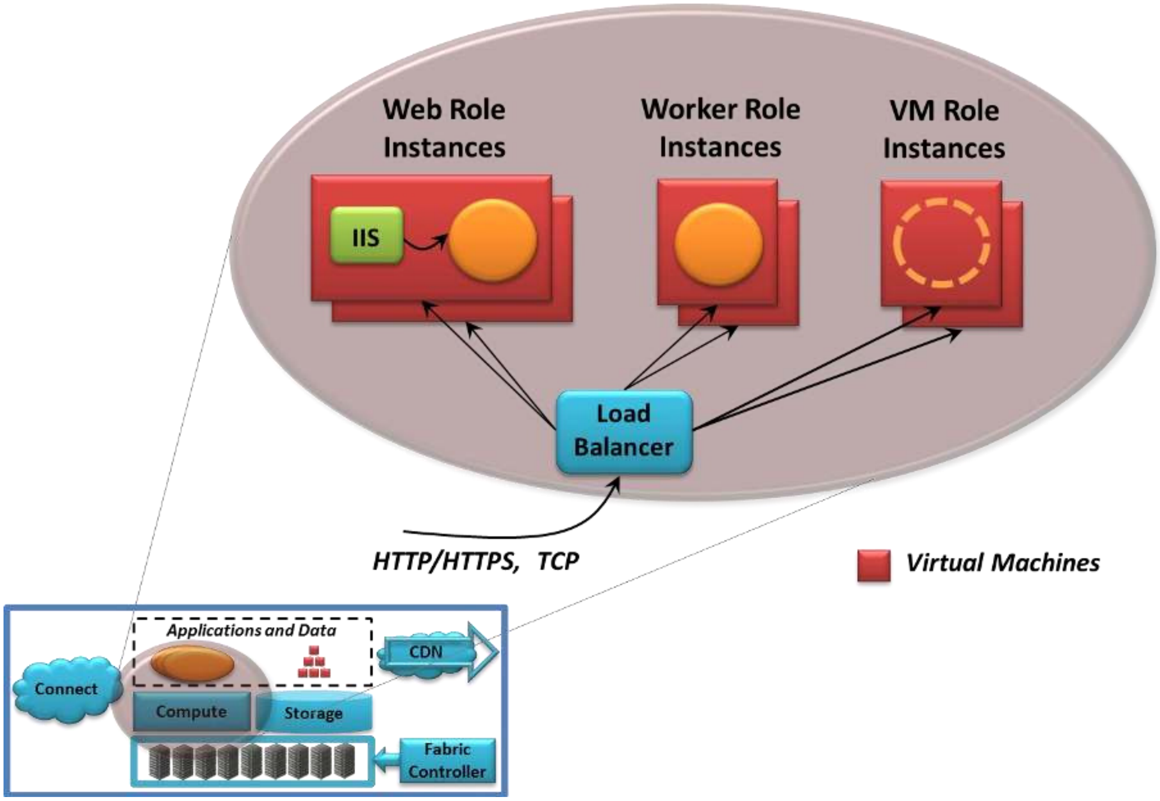

Figure 3: A running Windows Azure application consists of any combination of Web role instances, Worker role instances, and VM role instances.

In the current version of Windows Azure, developers can choose from three kinds of roles:

- Web roles, intended primarily to make it easier to create Web-based applications. Each Web role instance has Internet Information Services (IIS) 7 pre-configured inside it, so creating applications using ASP.NET, Windows Communication Foundation (WCF), or other Web technologies is straightforward. Developers can also create applications in native code—using the .NET Framework isn't required. This means that they can install and run non-Microsoft technologies as well, including PHP and Java.

- Worker roles, designed to run a variety of Windows-based code. The biggest difference between a Web role and a Worker role is that Worker roles don't have IIS configured inside them, and so the code they run isn't hosted by IIS. A Worker role might run a simulation, for example, or handle video processing or do nearly anything else. It's common for an application to interact with users through a Web role, then hand tasks off to a Worker role for processing. Once again, a developer is free to use the .NET Framework or other software that runs on Windows, including non-Microsoft technologies.