Microsoft

developer

// Step by step

Microsoft Azure SQL Database

Intermediate

TALLAN

Leonard G. Lobel
Eric D. Boyd

Microsoft

Microsoft® Azure™ SQL Database Step by Step

Leonard G. Lobel
Eric D. Boyd

Note The Target Server Response page also uses color coding to indicate success (green and blue) and failure (red).

22. Click Exit to close the wizard.

You have now deployed both the schema and data to the *WineCloudDb* SQL Database instance using an intuitive step-by-step tool, thanks to the Microsoft Azure SQL Database Migration Wizard. Beyond deploying both your database schema and data, it also analyzed your schema for compatibility issues when migrating from SQL Server to SQL Database.

To summarize, the tool performed the following actions:

1. Generated T-SQL scripts for all the database objects (schema) in the local SQL Server database
2. Exported data into data files using bcp
3. Analyzed the generated T-SQL script with a pattern matching rules engine that uncovers known incompatibilities and limitations
4. Deployed the database schema to SQL Database by executing the generated (and potentially autocorrected) T-SQL scripts
5. Imported data into SQL Database from the exported data files using bcp

All these steps (with the exception of the analysis step) could have been performed independently, as you did in the previous sections of this chapter. The SQL Database Migration Wizard just packages everything up in an easy-to-use tool that visually and interactively walks through the process, without you needing to use multiple tools and command prompts. But the rules engine analysis that the SQL Database Migration Wizard conducts on your local database schema is not something you can do with the other tools. This analysis is a unique and extremely compelling capability of the wizard.

The Microsoft Azure SQL Database Migration Wizard is open source, and you can look at the internals of this tool if you want. If you discover an incompatibility between SQL Server and SQL Database that the tool doesn't catch, or you're just curious about the predefined syntax rules, you can easily view the rules. They are defined in an XML file named *NotSupportedByAzureFile.Config*, which can be found in the same directory as the *SQLAzureMW.exe*. If you are comfortable with regular expressions, you can even add your own rules to the SQL Server Migration Wizard by modifying this XML file with a text editor.

Summary

It is rare to work on a project that is entirely greenfield and all new development. You're much more likely to work on a project involving Microsoft Azure SQL Database that will require the migration of existing databases. There are many ways to migrate your SQL Server databases to SQL Database, and they each have their own pros and cons to fit different scenarios. In this chapter, we walked through a number of tools and techniques for migrating existing SQL Server databases to SQL Database, including T-SQL scripts, .bacpac files, bcp, and the SQL Database Migration Wizard.

For lightweight scenarios, you saw how T-SQL scripts can be generated from a SQL Server database and executed against a SQL Database instance. SQL Data-Tier Application .bacpac files make it easy to package an entire database, including both schema and data, and import that into a SQL Database instance, but it operates at the database level and doesn't allow you to migrate individual database objects. Furthermore, for larger databases, the size of the .bacpac file can make it difficult to migrate to SQL Database. Bulk Copy with bcp is an efficient and high-performance way to migrate large amounts of data to SQL Database, but it doesn't do anything to migrate your database objects (schema). Finally, the Microsoft Azure SQL Database Migration Wizard is a free, open source project on Codeplex that is not commercially supported by a software vendor that brings together the process of migrating the schema and data to SQL Database while generating T-SQL scripts and automating bcp.

This page intentionally left blank

CHAPTER 5

Security and backup

—Eric Boyd

The topics of security, availability, and disaster recovery top the list of concerns that customers raise when considering the public cloud. These are certainly not new concerns introduced with the cloud; customers have been architecting solutions to deal with these same concerns since long before the cloud. The cloud is simply unfamiliar territory that causes these foundational concerns to be revisited. Thus, customers need these top concerns addressed with reasonable solutions before the public cloud is a viable option. Microsoft does a great job of putting customers' concerns at ease on these topics with the security processes and certifications that are in place in Microsoft Azure, along with the features of the platform that provide customers with the control and visibility they need.

In this chapter, we discuss security and backup concerns in the cloud. We start by explaining the general security responsibilities of any public cloud vendor, and then talk more specifically about security in Microsoft Azure and Microsoft Azure SQL Database. You will learn how to secure SQL Database by configuring the firewall as you create custom firewall rules and define users and permissions.

Security and backup often go hand in hand. Notwithstanding all other security-related concerns, how "secure" is your business if you have no backup in the event of an unforeseen disaster? So toward the end of this chapter, you will also learn how to copy and back up SQL Database, and how to schedule automated backups.

Addressing major cloud concerns

Two of the most common concerns users raise when considering public cloud platforms are security and business continuity, sometimes referred to as *disaster recovery*. Security is an overloaded term, and it can mean a lot of different things depending on the individual and context. So it is easier to think about security concerns by dividing them into two major categories: security concerns that are the sole responsibility of the public cloud vendor (Microsoft, in the case of Azure), and security concerns that are either the customer's responsibility or the combined responsibility of the customer and public cloud vendor.

Security responsibilities of the public cloud vendor

Some security concerns can be managed and addressed only by public cloud vendors, because customer access is limited to higher-level abstractions over the raw computing infrastructure, resources, and services. The customer typically cannot gain direct access to things like network routers, switches, and firewalls, as well as physical servers and the hypervisor, which is the software layer that virtualizes the hardware for multiple operating systems to run on a single physical server. As a result, it is very important to have a reputable cloud vendor with a successful history that you can count on and trust. But you cannot rely only on faith in a vendor, you also need transparency and insight into the resources and practices of your cloud vendor, and this includes their security practices and procedures, as described in the following sections.

Physical data center

Access to the physical data center—including entry inside the outermost security fence, entry into the building, and access to the physical infrastructure and hardware—must be managed with secure policies that are consistently enforced. You want it to be extremely difficult, and ideally impossible, for an unauthorized person to gain physical access to your servers.

Privacy from vendor personnel

The personnel who are authorized to gain access to the computing infrastructure and resources should still not be able to access your data, unless you explicitly grant them permission to do so. Because you don’t manage the foundational infrastructure, the vendor must ensure that it’s secure with the appropriate safeguards to prevent their personnel from accessing your data without your permission.

Isolating tenants

As is the case with vendor personnel, you don’t want other tenants of the cloud vendor to be able to gain access to your data and applications. (*Multitenancy* is an architecture in which a single infrastructure component serves multiple customers, where each customer is called a *tenant*.) When you are using multitenant services, this is a concern that must be managed by the vendor.

Preventing cyber attacks

A malicious attack, such as a denial-of-service attack, could occur against your applications and services, or at a broader level against the cloud vendor’s services. When these kinds of attacks occur, you want the cloud vendor to detect them and prevent them from causing a service outage.

Shared security responsibilities

Other security concerns are either the customer’s responsibility or are shared between the customer and the cloud vendor. Whenever a security concern can be affected by the customer’s configuration, implementation, or software, it cannot be the responsibility of the cloud vendor alone. The customer must secure aspects of their application to resolve these security concerns.

Meeting compliance requirements

A number of industries and organizations must meet regulatory requirements because of the nature of their businesses and the data they handle. These requirements often span the physical data center, applications running in the data center, and management processes across both of these areas. As a result, you need to understand what compliance certifications your cloud vendor has achieved. But you also need to understand that you also have a responsibility to meet the requirements that are outside of the cloud vendor's control that are application-centric and specific to your implementation.

Auditing activities

Much like compliance, knowing who did what and when they did it is a responsibility that is shared between the cloud vendor and the customer. Only the cloud vendor can track and provide an audit log of the activities that occur in the platform services. But it's the customer's responsibility to track the application-level activities. Because accurate and detailed auditing is a common requirement for most compliance certifications, it's an important capability both for your cloud vendor and your applications to provide. A core requirement for effective auditing requires you to provide unique credentials for every user and ensure that users do not share their credentials. If multiple users share a single account, you cannot possibly know exactly who performed an activity logged for that account.

Keeping electronic intruders out

Let's not forget the hackers who try to profit from stealing data and other hackers who just want to be malicious and aim to create chaos. You need to keep both of those types of hackers out. The cloud vendor must protect the infrastructure and core services and keep the electronic intruders out at that level. It's your responsibility to protect your applications from exploitations, and if you manage the virtual machines (VMs) in an Infrastructure-as-a-Service (IaaS) model, you must patch and secure your operating system whenever it is exploited.

Security in Microsoft Azure

Microsoft invests a lot of effort and talent into making sure Azure is a secure and reliable public cloud. Microsoft also does a great job being transparent and providing insight into the security and privacy practices of Microsoft Azure. One of the ways they do this is via a website called the Microsoft Azure Trust Center, which can be found at *http://azure.microsoft.com/en-us/support/trust-center/*. The Microsoft Azure Trust Center provides detailed information on Microsoft's practices that enable security, privacy, and compliance in Microsoft Azure. It is a great resource to gain a deeper understanding of security in Microsoft Azure and to find answers to your Microsoft Azure security questions.

Although Microsoft invests a lot of effort into ensuring SQL Database is a secure and reliable service, you still need to do a number of things to create a secure experience when using it. In the following sections, you will walk through step-by-step procedures that help to secure SQL Database. You will begin by securing access to and communication with SQL Database. Then you will walk through application-level security concerns such as SQL injection attacks and data encryption.