At a high level, the ServiceDefinition.csdef file contains metadata used by Windows Azure when hosting your application, including, for example, which roles are in your application. The ServiceConfiguration.Cloud.cscfg file and the ServiceConfiguration.Local.cscfg file provide configuration settings for your application as well as the number of instances to run for each role. By using multiple versions of the service configuration file, you can specify different settings to use when running in your local environment versus running on Windows Azure. For example, you can specify different connection strings based on the environment. Through Visual Studio, you can also create additional service configuration files, for example to maintain separate configuration settings for test and production deployments. For more information about how to leverage multiple configuration files, see [Configuring a Windows Azure Application](http://msdn.microsoft.com/en-us/library/ee405486.aspx).

The service definition also establishes configuration settings specific to the application. The service configuration files specify the number of instances to run for each role and sets the value of configuration settings defined in the service definition file. This separation between service definition and configuration allows you to update the settings of a running application by uploading a new service configuration file

Data Management

Windows Azure, SQL Azure, and the associated services provide

opportunities for storing and managing data in a range of ways. The

following data management services and features are available:

• **Azure Storage**. This provides four core services for persistent

and durable data storage in the cloud. The services support a

REST interface that can be accessed from within Azure-hosted

or on-premises (remote) applications. For information about the

REST API, see “*Windows Azure Storage Services REST API*

*Reference*” at http://msdn.microsoft.com/en-us/library/

dd179355.aspx. The four storage services are:

• **The Azure Table Service** provides a table-structured

storage mechanism based on the familiar rows and

columns format, and supports queries for managing the

data. It is primarily aimed at scenarios where large volumes

of data must be stored, while being easy to access and

update. For more detailed information see “*Table Service*

*Concepts*” at http://msdn.microsoft.com/en-us/library/

dd179463.aspx and “*Table Service REST API*” at http://

msdn.microsoft.com/en-us/library/dd179423.aspx.

• **The Binary Large Object (BLOB) Service** provides a

series of containers aimed at storing text or binary data. It

provides both Block BLOB containers for streaming data,

and Page BLOB containers for random read/write operations.

For more detailed information see “U*nderstanding*

*Block Blobs and Page Blobs*” at http://msdn.microsoft.com/

en-us/library/ee691964.aspx and “*Blob Service REST API*”

at http://msdn.microsoft.com/en-us/library/dd135733.

aspx.

• **The Queue Service** provides a mechanism for reliable,

persistent messaging between role instances, such as

between a Web role and a Worker role. For more detailed

information see “*Queue Service Concepts*” at http://msdn.

microsoft.com/en-us/library/dd179353.aspx and “*Queue*

*Service REST API*” at http://msdn.microsoft.com/en-us/

library/dd179363.aspx.

• **Windows Azure Drives** provide a mechanism for applications

to mount a single volume NTFS VHD as a Page

BLOB, and upload and download VHDs via the BLOB. For

more detailed information see “*Windows Azure Drive*”

(PDF) at <http://go.microsoft.com/?linkid=9710117>.

**SQL Azure Database**. This is a highly available and scalable

cloud database service built on SQL Server technologies, and

supports the familiar T-SQL based relational database model. It

can be used with applications hosted in Windows Azure, and

with other applications running on-premises or hosted elsewhere.

For more detailed information see “*SQL Azure Database*”

at http://msdn.microsoft.com/en-us/library/ee336279.aspx.

• **Data Synchronization**. SQL Azure Data Sync is a cloud-based

data synchronization service built on Microsoft Sync Framework

technologies. It provides bi-directional data synchronization

and data management capabilities allowing data to be easily

shared between multiple SQL Azure databases and between

on-premises and SQL Azure databases. For more detailed

information see “*Microsoft Sync Framework Developer Center*” at

http://msdn.microsoft.com/en-us/sync.

• **Caching**. This service provides a distributed, in-memory, low

latency and high throughput application cache service that

requires no installation or management, and dynamically increases

and decreases the cache size automatically as required.

It can be used to cache application data, ASP.NET session state

information, and for ASP.NET page output caching. For more

detailed information see “*Windows Azure Caching Service*” at

<http://msdn.microsoft.com/en-us/library/gg278356.aspx>.

**2 Windows Azure Drive Overview**

Customers have told us that one of the challenges is taking their already running Windows applications and running them in the cloud while making sure their data is durable while using the standard Windows NTFS APIs. With Windows Azure Drive, your Windows Azure applications running in the cloud can use existing NTFS APIs to access a durable drive. This can significantly ease the migration of existing Windows applications to the cloud. The Windows Azure application can read from or write to a drive letter (e.g., X:\) that represents a durable NTFS volume for storing and accessing data. The durable drive is implemented as a Windows Azure Page Blob containing an NTFS-formatted Virtual Hard Drive (VHD).

**Table Storage** allows you to store serialized entities in a table, but the term table here is not a relational database table. To provide people with some analogy they can use to get their arms around Table Storage, I like to tell people to think of Table Storage as a fancy spreadsheet. You can store the state of your entities in the columns of the spreadsheet. However, there is no linkage or relationship (therefore joins) between entities - at least none that is automatically managed and maintained by Azure. There are no custom indexes - at least not today.

The Table service offers semi-structured storage in the form of tables that contain collections of entities. Entities have a primary key and a set of properties, where a property is a name, typed-value pair.

In addition to the properties required by your model, every entity in Table Storage has two key properties: the PartitionKey and the RowKey. These properties together form the table's primary key and uniquely identify each entity in the table. Entities also have a Timestamp system

In addition to the properties required by your model, every entity in Table Storage has two key properties: the PartitionKey and the RowKey. These properties together form the table's primary key and uniquely identify each entity in the table.  
  
Read more: <http://www.intertech.com/Blog/Post/Windows-Azure-Table-Storage-vs-Windows-SQL-Azure.aspx#ixzz2HY1YtNsc>

Access to Azure Table Storage is accomplished either via REST API or Storage Client Library provided with the Windows Azure SDK. Using the REST API allows client applications to communicate and use data from Table Storage without having detailed and specific knowledge of an Azure API, but it is more complex and difficult to work with. The Storage Client Library (which leverages LINQ to Objects) provides a layer of convenience but requires the application reference the Storage Client Library APIs. REST and the Storage Client Library incur a learning curve that is typically not there when using SQL Azure.

Read more: <http://www.intertech.com/Blog/Post/Windows-Azure-Table-Storage-vs-Windows-SQL-Azure.aspx#ixzz2HY7kDJdj>

## Transactions and Concurrency

SQL Azure supports typical ACID transactions for work within the same database. Transactions across databases are not supported. SQL Azure allows for typical optimistic and pessimistic concurrency strategies.

Table Storage supports transactions for entities in the same table and table partition, but not across tables or partitions.

## Queries

Using Table Storage, queries are limited to 1000 entities by default. If more than 1000 entities are found, a continuation token is returned and must be used by the application to retrieve the next set of entities.

SQL Azure has no limitations, issues or special programming requirements to work with large queries. Good database and index design can help improve performance of queries; especially large ones.

Azure Table Storage **costs** 15? per GB of storage per month. Additionally, you pay 1 cent per 10,000 transactions with Table Storage. SQL Azure costs are $9.99 for 1 GB of storage per month ($99.99 for 10GB).

* "If size is the issue, that would be the first flag that you might want to consider Azure Tables.
* As long as the support Tables has for transactions and queries meets your needs. The size limit surely will, at 100TB."
* Suggest sophisticated transactions, or a complex authorization model might require the services of SQL Azure.
* "The final consideration is cost. I can store a lot of data in Azure Tables for a lot less money than I can in SQL Azure. SQL Azure is giving me a lot more features to use (joins, relationships, etc.), but it does cost more."

<http://www.intertech.com/Blog/Post/Windows-Azure-Table-Storage-vs-Windows-SQL-Azure.aspx>

# Networking Services

Windows Azure provides several networking services that you

can take advantage of to maximize performance, implement authentication,

and improve manageability of your hosted applications.

These services include the following:

**Content Delivery Network** (CDN). The CDN allows you to

cache publicly available static data for applications at strategic

locations that are closer (in network delivery terms) to end

users. The CDN uses a number of data centers at many locations

around the world, which store the data in BLOB storage that

has anonymous access. These do not need to be locations where

the application is actually running. For more detailed information

see “*Delivering High-Bandwidth Content with the Windows*

*Azure CDN*” at http://msdn.microsoft.com/en-us/library/

ee795176.aspx.

• **Virtual Network Connect**. This service allows you to configure

roles of an application running in Windows Azure and computers

on your on-premises network so that they appear to be on

the same network. It uses a software agent running on the

on-premises computer to establish an IPsec-protected connection

to the Windows Azure roles in the cloud, and provides the

capability to administer, manage, monitor, and debug the roles

directly. For more detailed information see “*Connecting Local*

*Computers to Windows Azure Roles*” at http://msdn.microsoft.

com/en-us/library/gg433122.aspx.

• **Virtual Network Traffic Manager**. This is a service that allows

you to set up request redirection and load balancing based on

three different methods. Typically you will use Traffic Manager

to maximize performance by redirecting requests from users to

the instance in the closest data center using the Performance

method. Alternative load balancing methods available are

Failover and Round Robin. For more detailed information see

“*Windows Azure Traffic Manager*” at http://msdn.microsoft.com/

en-us/WAZPlatformTr

**Access Control**. This is a standards-based service for identity

and access control that makes use of a range of identity providers

(IdPs) that can authenticate users. ACS acts as a Security

Token Service (STS), or token issuer, and makes it easier to take

advantage of federation authentication techniques where user

identity is validated in a realm or domain other than that in

which the application resides. An example is controlling user

access based on an identity verified by an identity provider such

as Windows Live ID or Google. For more detailed information

see “*Access Control Service 2.0*” at http://msdn.microsoft.com/

en-us/library/gg429786.aspx and “*Claims Based Identity & Access*

*Control Guide*” at http://claimsid.codeplex.com/.

• **Service Bus**. This provides a secure messaging and data flow

capability for distributed and hybrid applications, such as

communication between Windows Azure hosted applications

and on-premises applications and services, without requiring

complex firewall and security infrastructures. It can use a range

of communication and messaging protocols and patterns to

provide delivery assurance, reliable messaging; can scale to

accommodate varying loads; and can be integrated with onpremises

BizTalk Server artifacts. For more detailed information

see “*Service Bus*” at http://msdn.microsoft.com/en-us/library/

ee732537.aspx.

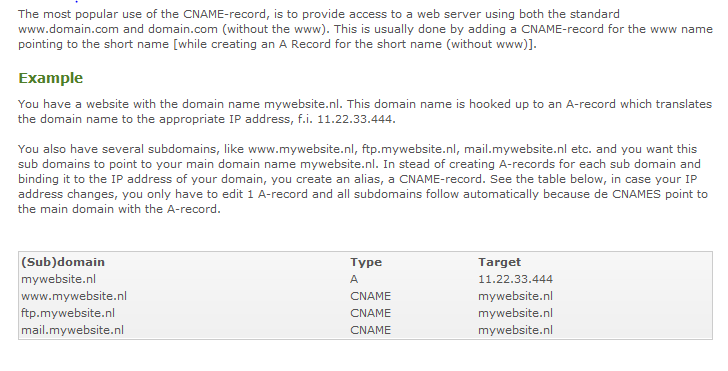
The traffic manager enables you to

control the distribution of traffic to multiple Windows Azure instances,

even if those instances are running in different data centers.

A **CNAME record** is an abbreviation for **Canonical Name record** and is a type of [resource record](http://en.wikipedia.org/wiki/Resource_record) in the [Domain Name System](http://en.wikipedia.org/wiki/Domain_Name_System) (DNS) that specifies that the [domain name](http://en.wikipedia.org/wiki/Domain_name) is an alias of another, canonical domain name. Here "canonical" usually means: a more generally accepted or standard name.

This helps when running multiple services (like an FTP server *and* a webserver; each running on different ports) from a single IP address. Each service can then have its own entry in the DNS (like ftp.example.com and www.example.com).



Autoscaling Application Block can automatically scale your Windows Azure application based on rules that you define specifically for your application. You can use these rules to help your Windows Azure application maintain its throughput in response to changes in its workload, while at the same time control the costs associated with hosting your application in Windows Azure. Scaling operations typically alter the number of role instances in your application, but the block also enables you to use other scaling actions such as throttling certain functionality within your application.

Typically, you will host the Autoscaling Application Block in its own worker role in the cloud, or in an on-premises application, from where it can monitor and scale your Windows Azure application.