Corso di Sistemi Distribuiti e Cloud Computing

Corso di Laurea Magistrale in Ingegneria Informatica A.A. 2019/2020

DIMES - Università degli Studi della Calabria

CLOUD COMPUTING

INTRODUCTION TO MICROSOFT AZURE

Cloud Computing

- It is a computing model where HW/SW resources are scalable, virtualized and provided by a service-oriented interface
- Location independent computing
- Shared servers (data center) offering resources, software, and data
- Elasticity
- Natural evolution of:
  - Virtualization
  - Service-Oriented Architecture
  - Utility computing*
- Details are abstracted from consumers

*a business model in which one company outsources part or all of its computer support to another company

Cloud Computing advantages:

- **High Scalability, Elasticity and Cost Efficiency**: cloud instances are deployed automatically only when needed and as a result, you pay only for the resources you need. Hand in hand, also comes elasticity, since clouds can be scaled to meet changing IT system demands.

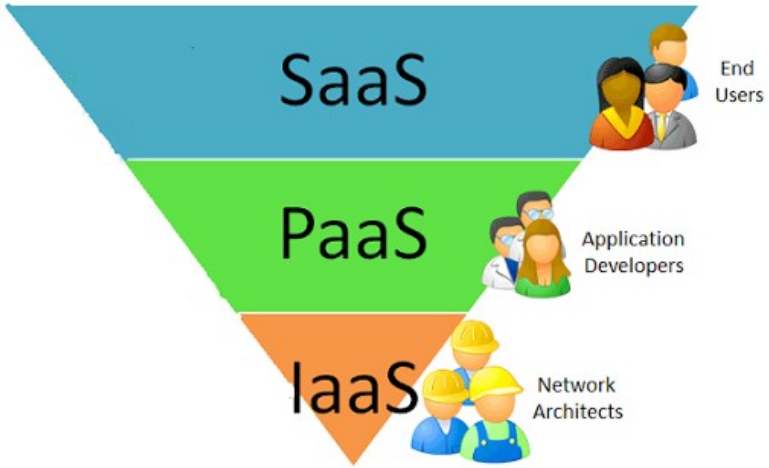
- **Broad network access**: cloud services are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

- **Energy saving up to 90%** using virtualization and workload consolidation.

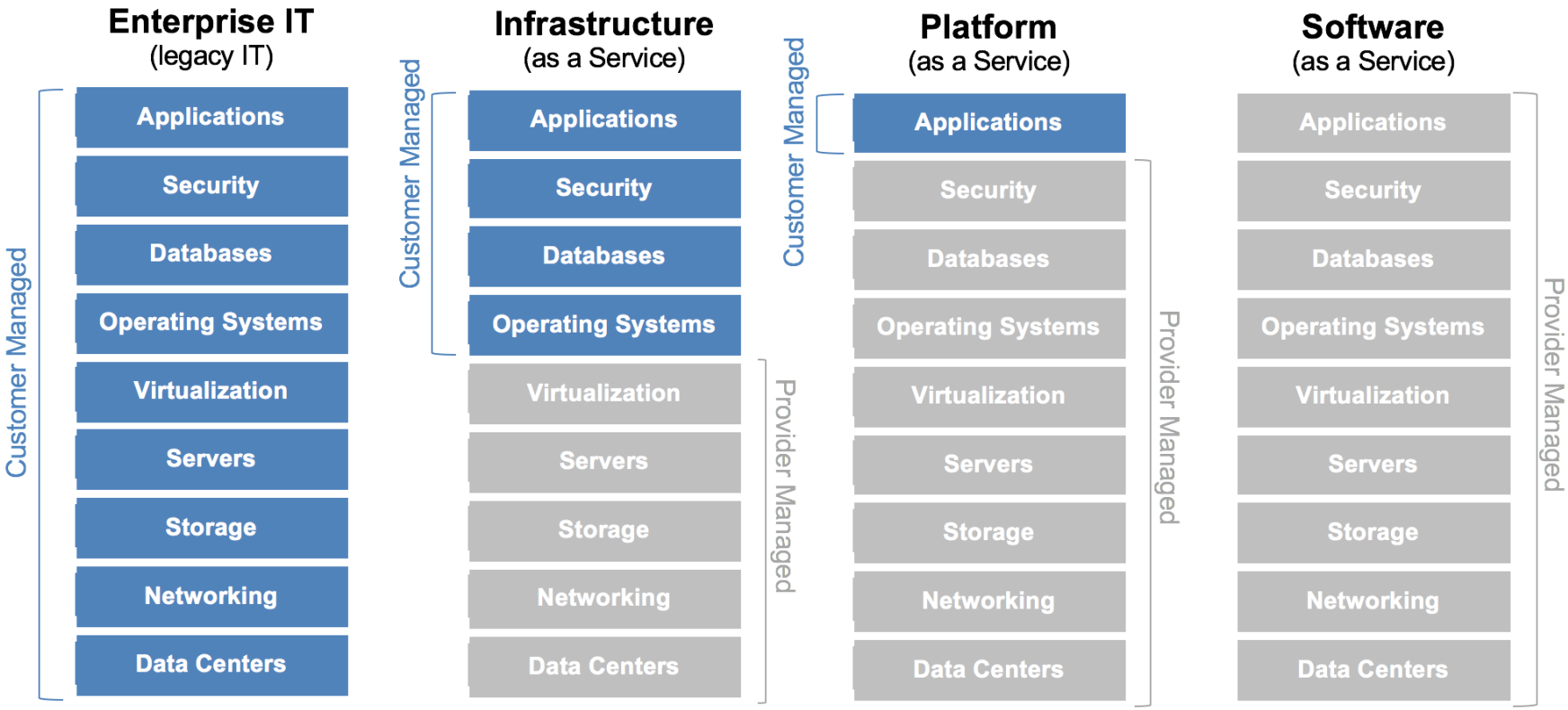
- **Resiliency and disaster recovery:** use of multiple redundant sites makes well-designed cloud computing suitable for business continuity and disaster recovery.

Cloud Computing

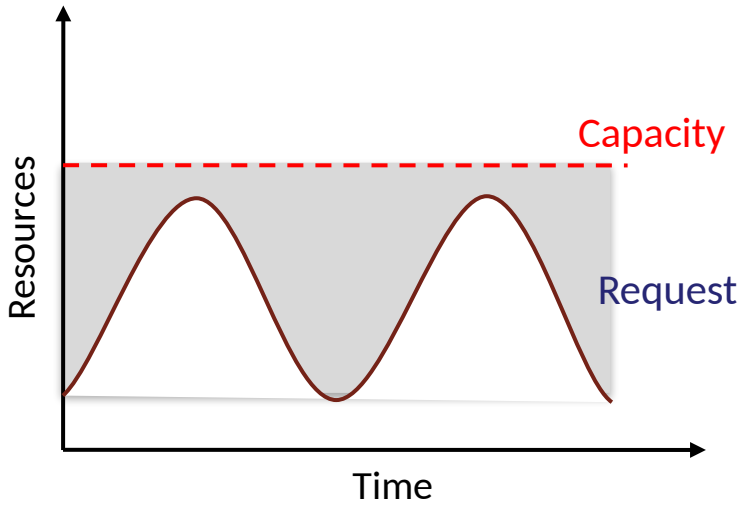
- Infrastructure as a Service (IaaS): basic compute and storage resources
  - On-demand servers
  - Amazon EC2, VMWare vCloud
- Platform as a Service (PaaS): cloud application infrastructure
  - On-demand application-hosting environment
  - E.g. Google AppEngine, Salesforce.com, Microsoft Azure, Amazon
- Software as a Service (SaaS): cloud applications
  - On-demand applications
  - E.g. GMail, Microsoft Office Web Companions



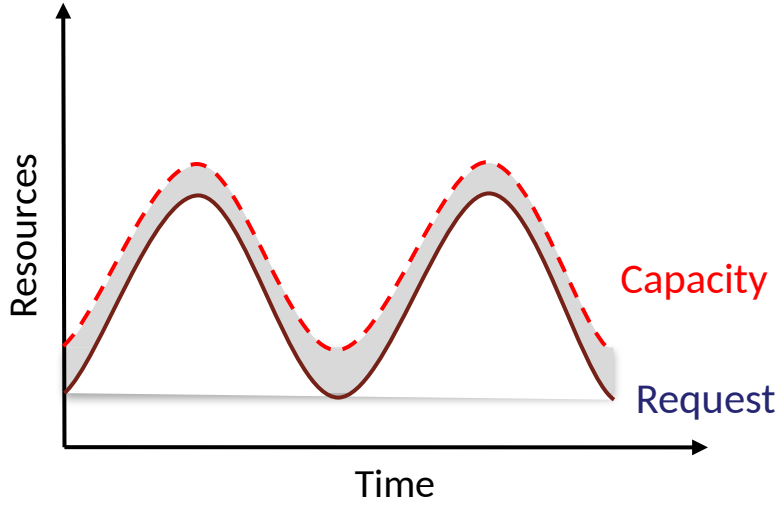
Management responsibilities on Cloud Computing



Economical aspect



Static Data center



Cloud Data center

Not Used Resources

From "Above the Clouds", by UC Berkeley RAD Lab.

Microsoft Azure

- Microsoft Azure is an Operating System for the data center
  - Model: Treat the data center as a machine
  - Handles resource management, provisioning, and monitoring
  - Manages application lifecycle
  - Allows developers to concentrate on business logic

- Provides shared pool of compute, disk and network
  - Virtualized storage, compute and network
  - Illusion of boundless resources

- Provides common building blocks for distributed applications
  - Reliable queuing, simple structured storage, SQL storage
  - Application services like access control and connectivity

Microsoft Azure

- Microsoft Azure provides a very large catalog of Services to cope with the needs of the different users.
  - Machine learning, Mobile Apps, Storage, Automation, Data Analytics.

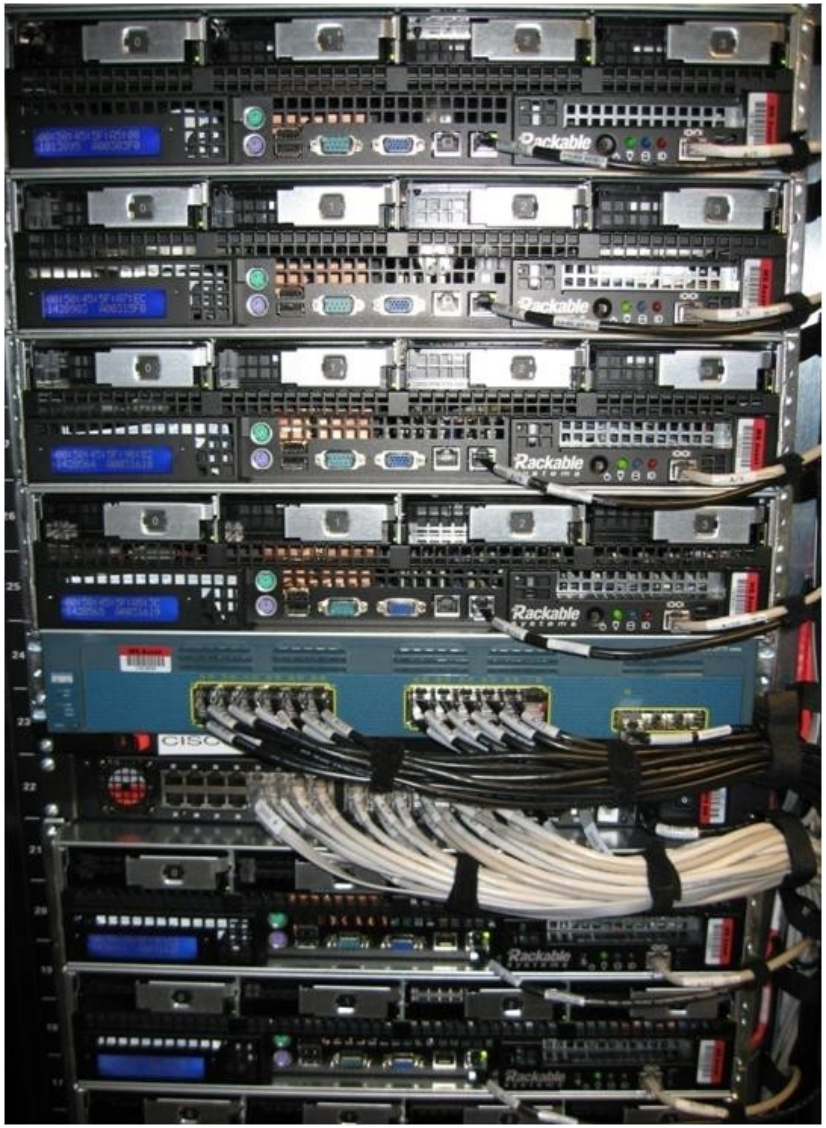
Microsoft Azure for Students

Microsoft Azure

- Platform as a Service
  - Application Platform in the Cloud

- Provides:
  - Compute
    - Web, Worker & VM Role
  - Storage
    - Blob, Table, Queue & Azure SQL Server
  - Application Fabric
    - Service Bus, Access Control, (Future: Cache, Integration & Composite)

Microsoft Azure Datacenters

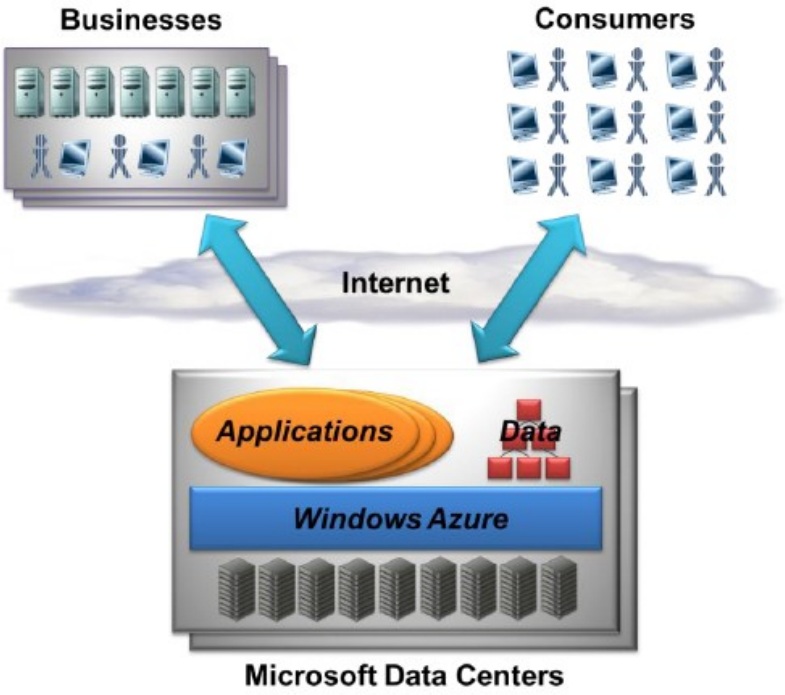


Microsoft Azure

- Part of the larger Microsoft Azure platform, Microsoft Azure is a foundation for running applications and storing data in the cloud

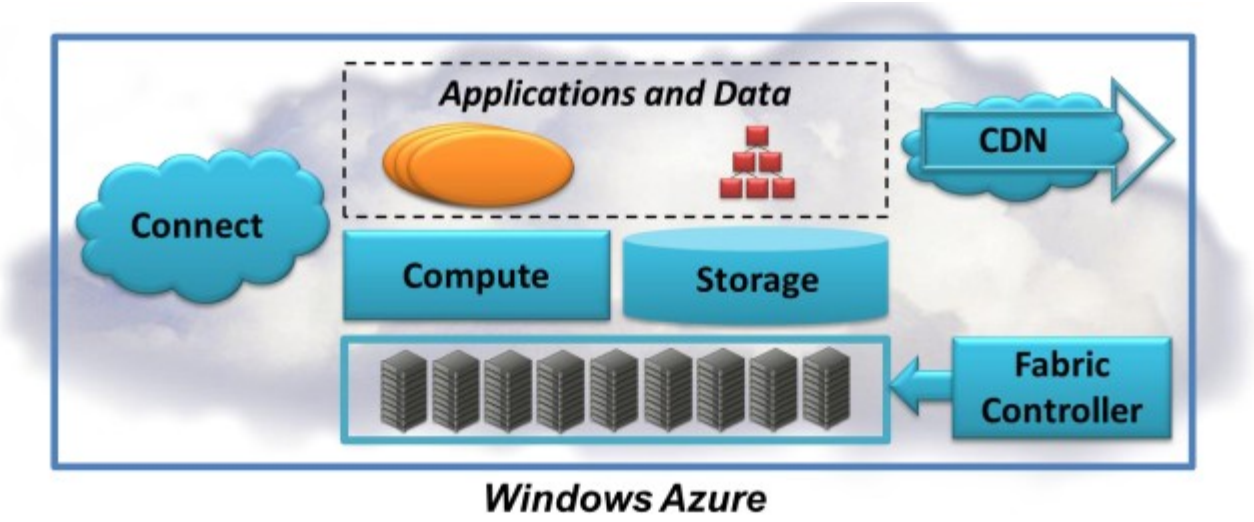
- Rather than providing software that Microsoft customers can install and run themselves on their own computers, Microsoft Azure provides this as a service

- Customers use it to run applications and store data on Internet-accessible machines owned by Microsoft



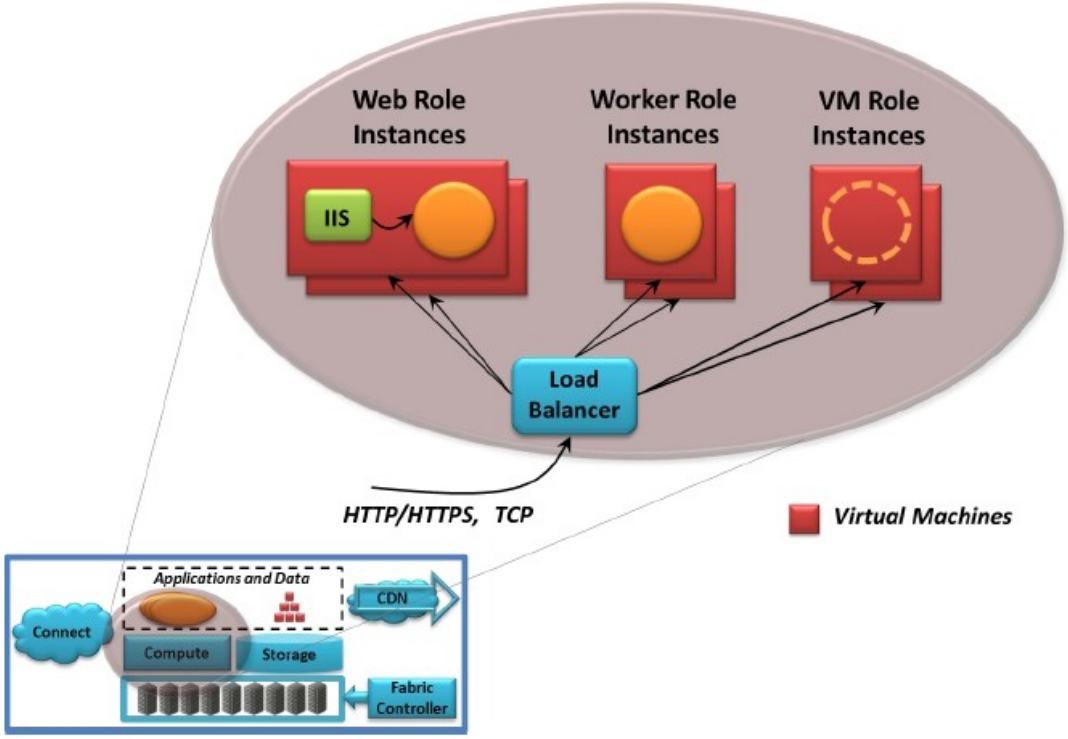
Microsoft Azure: Components

- To support cloud applications and data, Microsoft Azure is a composition of five components.
  - Compute
  - Storage
  - Content Delivery Network (CDN)
  - Fabric Controller
  - Connect



Compute

- Microsoft Azure compute can run many different kinds of applications
- Whatever an application does, it must be implemented as one or more roles
  - Web Role
  - Worker Role
  - VM Role (now Windows Azure Virtual Machines)
- Microsoft Azure then typically runs multiple instances of each role, using built-in load balancing to spread requests across them



Compute

- Web roles
  - make easier to create Web-based applications
  - each Web role instance has Internet Information Services (IIS) 7 pre-configured inside it, so it is simple to create applications using **ASP.NET**, Windows Communication Foundation (WCF), or other Web technologies
  - developers can also create applications in native code (using the .NET Framework isn't required)
  - 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
  - they might run a simulation, handle video processing, etc.
  - it's common for an application to interact with users through a Web role, then hand tasks off to a Worker role for processing
  - a developer is free to use the .NET Framework or other software that runs on Windows, including non-Microsoft technologies

Compute

- Windows Azure Virtual Machines Roles
  - by the VM roles it is possible to deploy custom OS images on Microsoft Azure
  - each instance runs a user-provided Windows Server 2012 image
  - among other things, a VM role can sometimes be useful in moving an on-premises Windows Server application to Microsoft Azure

Compute

- To submit an application to Azure, a developer specifies to the portal the number of instances (for each role) to run on the platform
- Requests from the application's users can be made using protocols such as HTTP, HTTPS, and TCP
- A developer can use any combination of different role instances
  - If the application's load increases\decreases, he\she can use the Microsoft Azure portal to request more\less instances for any of the roles in his application
  - Azure also exposes an API that lets all of these things be done programmatically
- To create Microsoft Azure applications, a developer uses the same languages and tools as for any Windows application.
  - programming Web role in ASP.NET, Visual Basic, C#
  - programming Worker role in .NET languages, C++, or Java
- Each instance can call a logging API for logging and debugging

Microsoft Azure

A full view of Azure Components and Services (updated to March 2015)

- Microsoft Azure recently changed the name of its components and reorganized them in different areas.

- More info at:

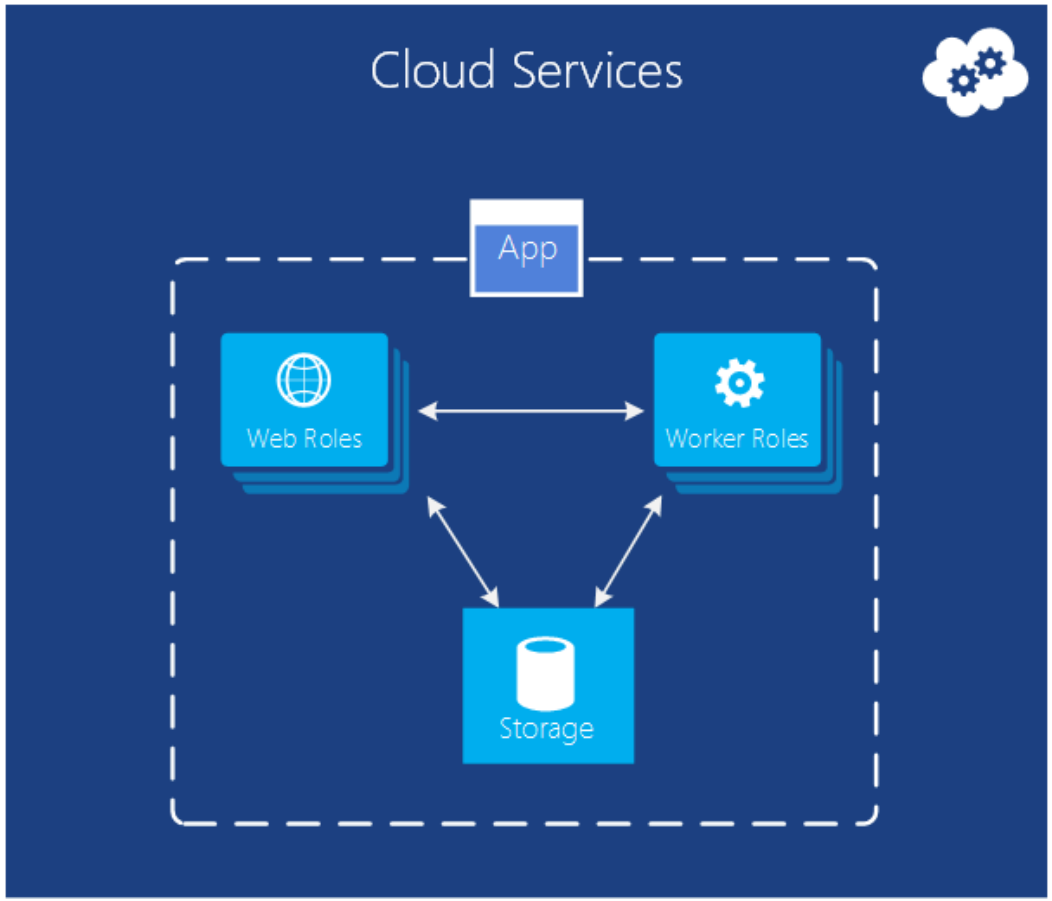
http://azure.microsoft.com/en-us/documentation/articles/fundamentals-introduction-to-azure/

http://azure.microsoft.com/en-us/documentation/infographics/azure/

Azure Cloud Services

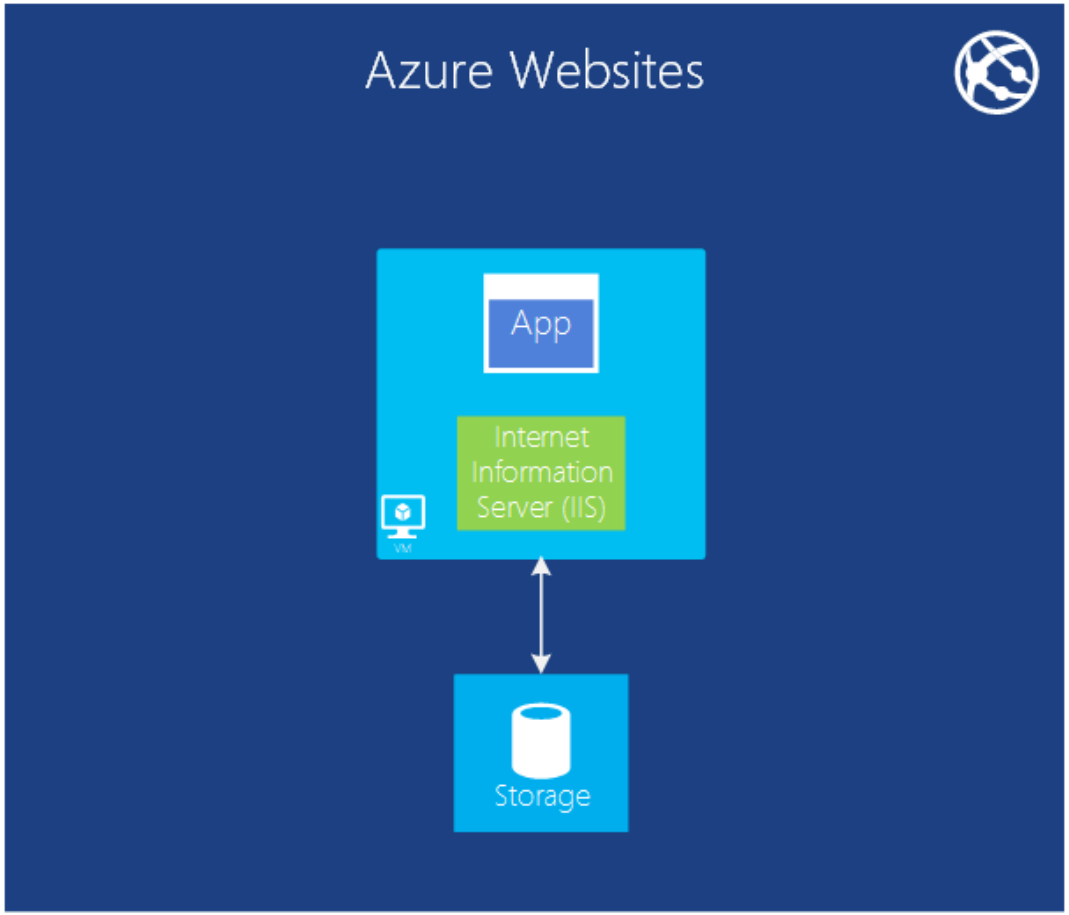
- Cloud Services is a platform-as-a-service (PaaS) option tuned to create highly scalable and fault resistant applications, but with more flexibility than Websites.

- It allows creation of Worker Roles and Web Roles.



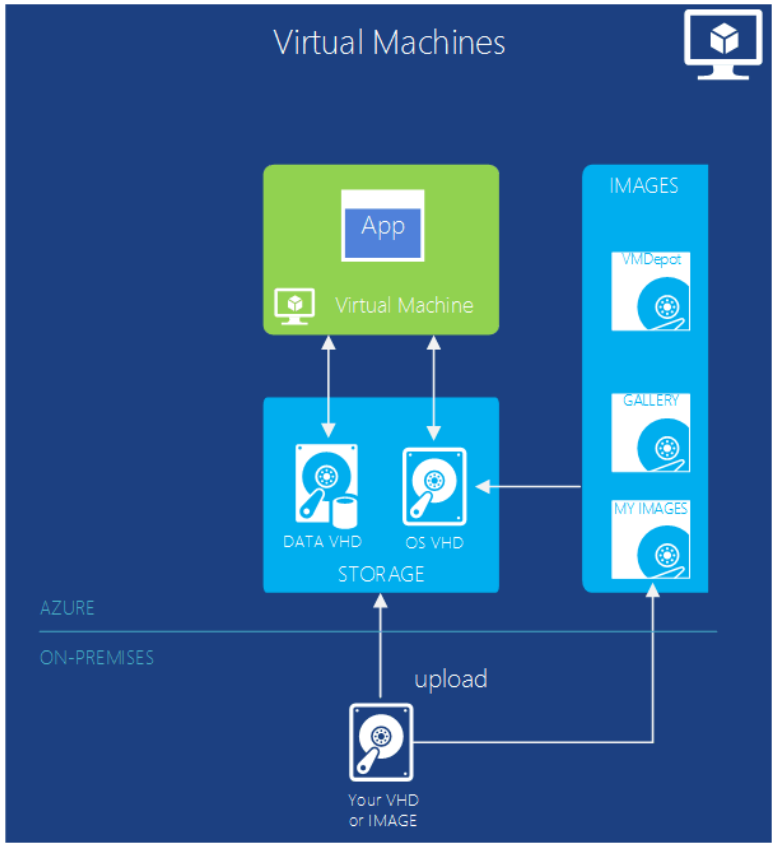
Azure Websites

- Websites offers a range of applications, frameworks, and templates for you to build large, scalable web applications and presence websites quickly, and then efficiently manage development, test, and operations.



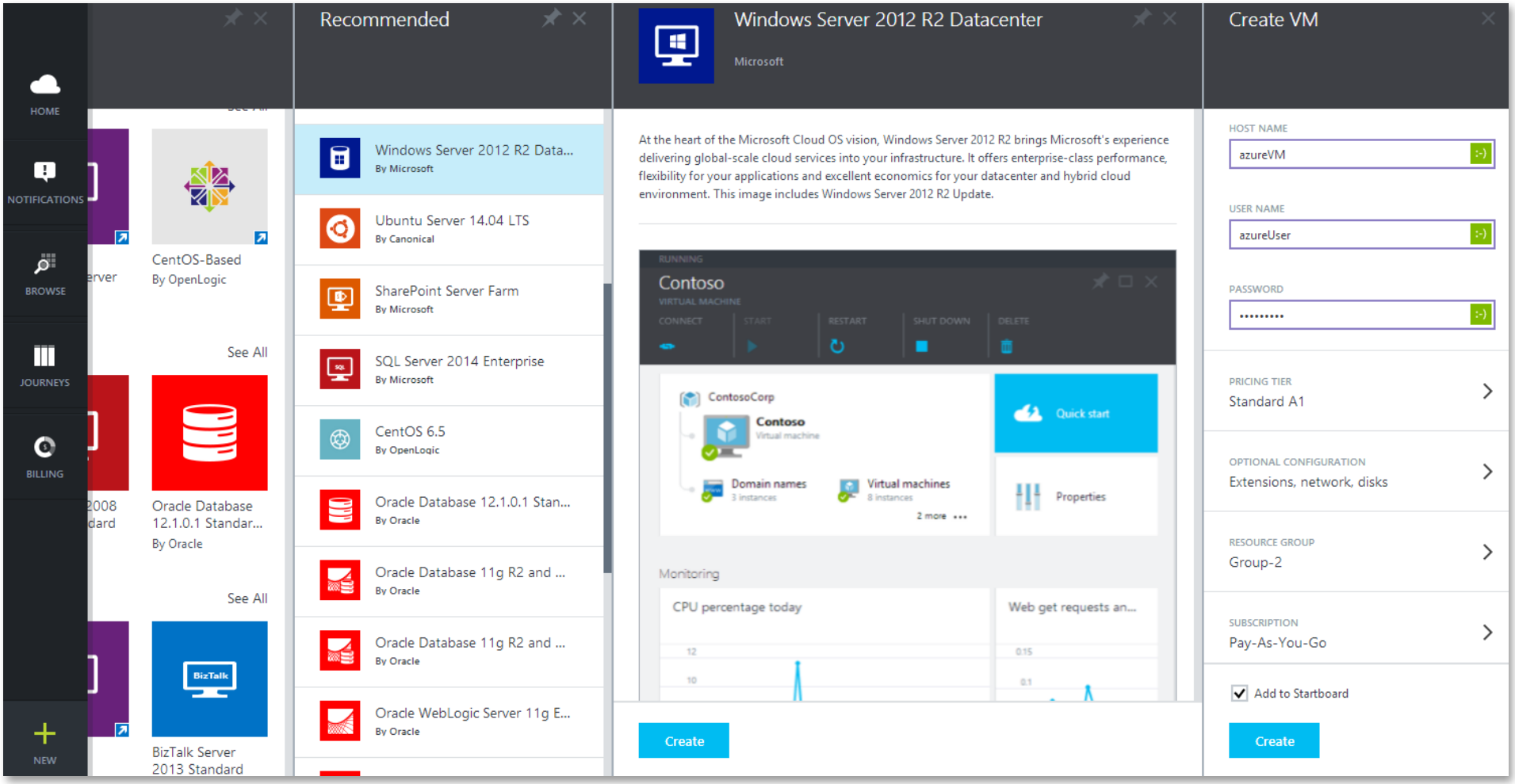
Azure Virtual Machines

- To create a VM, you specify which VHD to use and the VM's size.
- You then pay for the time that the VM is running
- Azure offers a gallery of stock VHDs (called "images") that contain a bootable operating system to start from (including Microsoft and partner options, such as Windows Server and Linux, SQL Server, Oracle and many more).
- You're free to create VHDs and images, and then upload them yourself.



Azure Virtual Machines

- Virtual Machines gives you control over your own virtual machine, including the operating system.



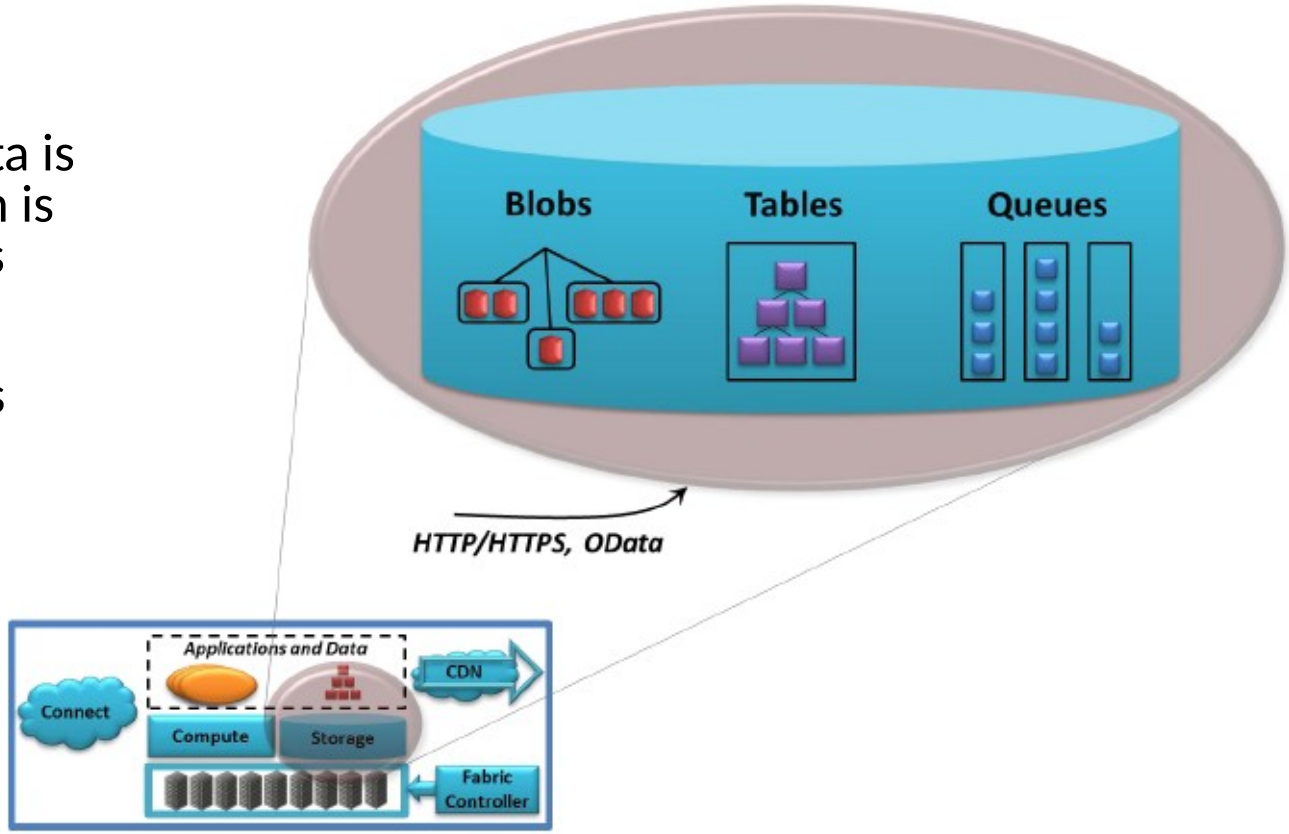
Storage

- Since the applications work with data in many different ways, the storage service provides several options
  - Blobs
  - Tables
  - Queues

- Regardless of how data is stored, all information is replicated three times

- This replication allows fault tolerance

- It exploits a strong consistency policy

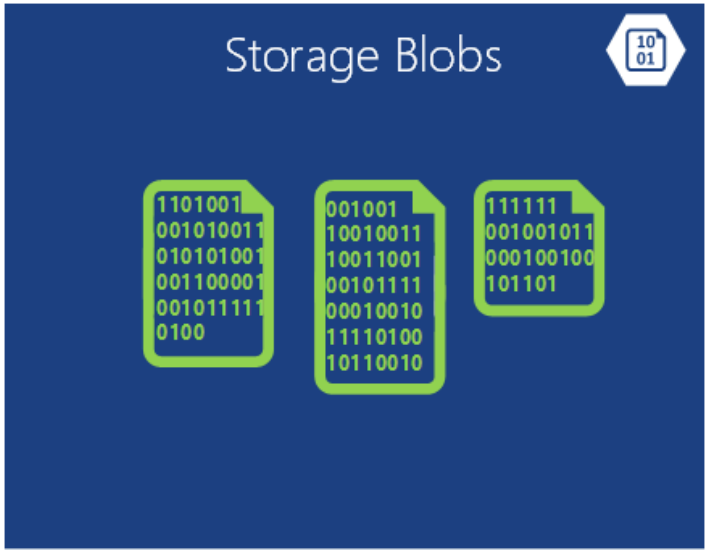


Storage

- Blobs
  - They are the simplest way to store data in Microsoft Azure storage
  - They contain binary data and can be very large (up to a Terabyte)
  - They can also have associated metadata, i.e., the singer for an MP3 file
  - They provide the underlying storage for Microsoft Azure drives for a persistent storage
  - On the other side, they are just right for some situations, but they are too unstructured for others
  - Blobs are organized inside containers which can be accessed using HTTP or HTTPS.
  - A storage account can contain any number of containers, and a container can contain any number of blobs, up to the 500 TB capacity limit of the storage account.

Blob storage offers two types of blobs:

- block blobs: optimized for streaming and storing cloud objects, and are a good choice for storing documents, media files, backups etc. A block blob can be up to 200 GB in size.
- page blobs: optimized for representing IaaS disks, and may be up to 1 TB in size.



The Blob Service - Concepts

The Blob service contains the following components:

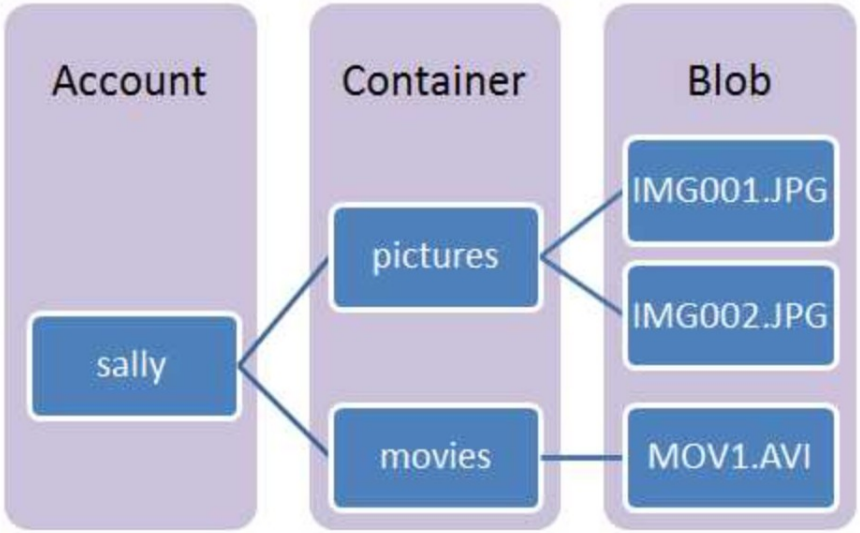
Storage Account: it is needed to access to any Azure Storage service

Container: it provides a grouping of a set of blobs

- All blobs must be in a container
- An account can contain an unlimited number of containers
- A container can store an unlimited number of blobs.

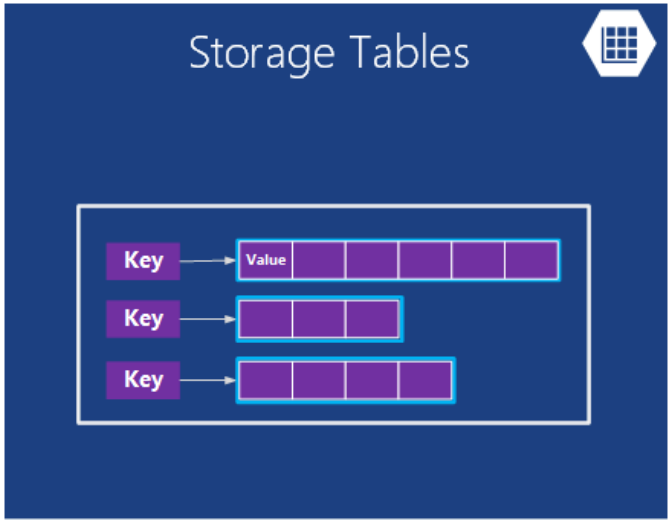
Blob: it is a file of any type and size

- two types of blobs: block and page blobs.
- A Block blob can be up to 200 GB in size.
- A Page blob, can be up to 1 TB in size
  - (more efficient when ranges of bytes in a file are modified frequently)
- A Directory blob is a directory



Storage

- Tables
  - They are not relational tables (don't be misled by the name!!!)
  - They let applications work with data in a more fine-grained way than blobs
  - The data each one holds is actually stored in a group of entities containing properties
  - They allows scale-out storage (scaling by spreading data across many machines), in fact a single table can contain billions of entities holding Terabytes of data.
  - Applications that rely on relational storage can use SQL Azure (that gives SQL-based access to relational storage in the cloud)
  - Like Blobs and Queues, developers can manage and access Table Storage using standard REST protocols, however Table Storage also supports a subset of the OData protocol, simplifying advanced querying capabilities and enabling both JSON and AtomPub (XML based) formats.



The Table Service

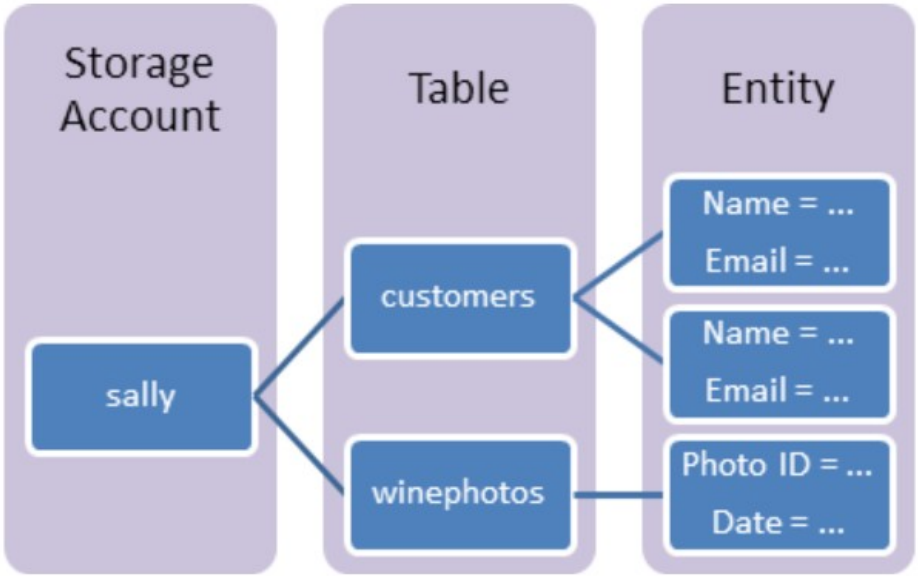
- The Azure Table storage service stores large amounts of structured data.
  - tables are ideal for storing structured, non-relational data
  - the service is a NoSQL datastore
  - accepts authenticated calls from inside and outside the Azure cloud

- Table service can be exploited to:
  - store TBs of structured data capable of serving web scale applications
  - store datasets that don't require complex joins, foreign keys, or stored procedures and can be denormalized for fast access
  - quickly query data using a clustered index
  - access data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries

- Following the Cloud philosophy, tables will scale as demand increases

The Table Service - Concepts

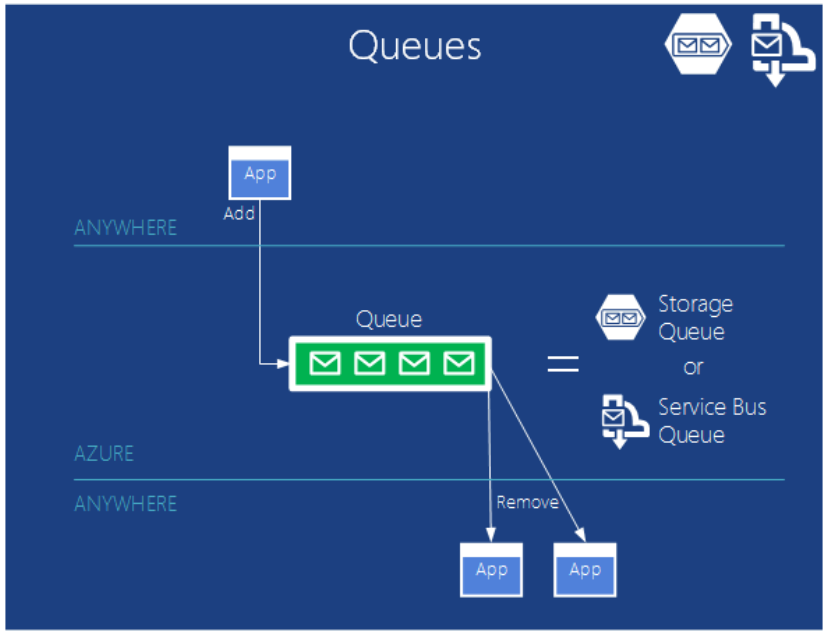
- **Storage Account:** it needs to access to any Azure Storage service.
- **Table**: a collection of entities, no table schema enforced (i.e., a single table can contain entities that have different sets of properties).
- **Entity:** a set of properties, like a database row (size up to 1MB).
- **Property:** a name-value pair. Each entity can include up to 252 properties to store data. Each entity also has 3 system properties that specify a partition key, a row key, and a timestamp. Entities with the same partition key can be queried more quickly, and inserted/updated in atomic operations. An entity's row key is its unique identifier within a partition



The Table Service - Entity class

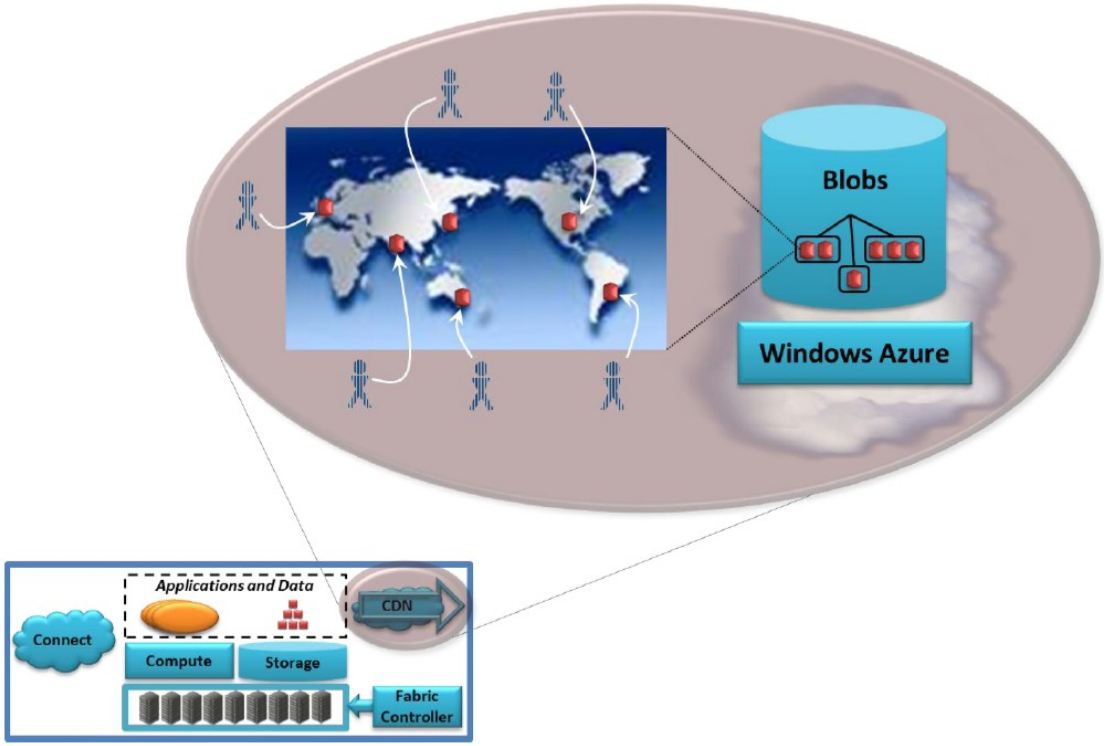
- Entities are custom classes derived from TableEntity
  - The entity class defines the properties of the entity
  - Each entity class (or type) must expose a parameter-less constructor.
  - About properties:
    - Row key
    - Partition key
      - Entities with the same partition key can be queried faster
      - Nevertheless, using diverse partition keys allows for greater parallel operation scalability.
    - Other properties…, but each one must be public and of a supported type
    - Each property must expose both get and set

- Queues
  - They have a different task than blobs and tables (both focused on storing and accessing data)
  - They provide a way for Web role instances to communicate asynchronously with Worker role instances
  - Example:
    1. a user submits a request to perform some compute-intensive task via a Web interface implemented by a Web role.
    2. the Web role instance that receives this request can write a message into a queue describing the work to be done
    3. a Worker role instance that's waiting on this queue can then read the message and carry out the task it specifies
    4. any results can be returned via another queue or handled in some other way



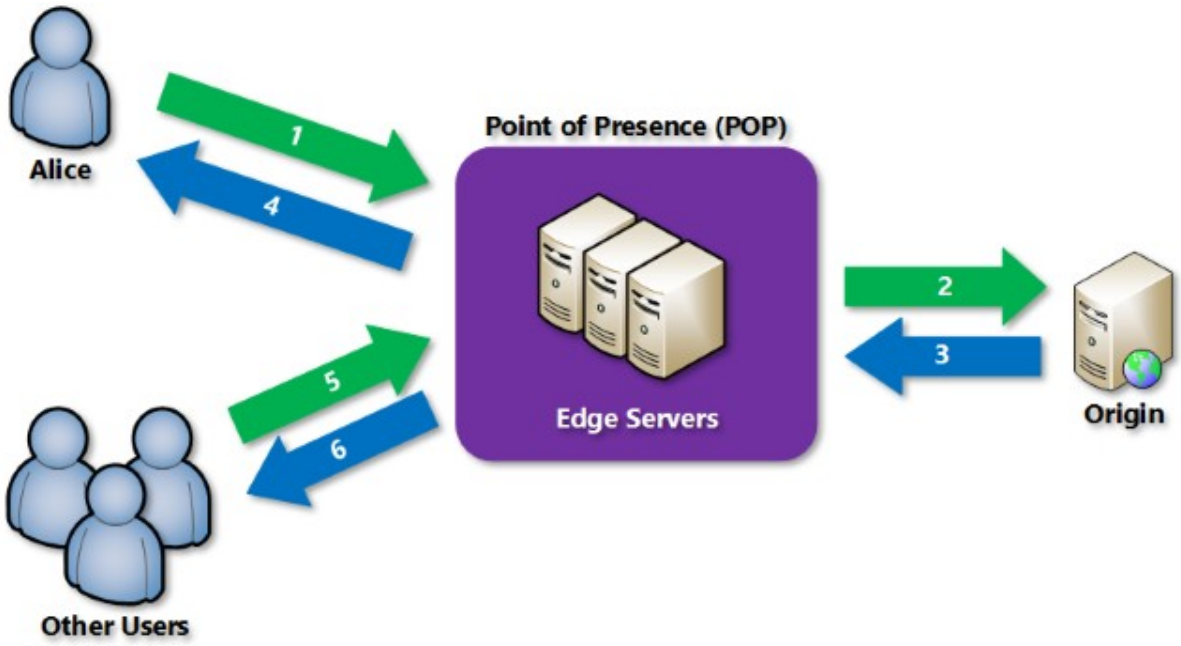
Content Delivery Network (CDN)

- The CDN is aimed at improving efficiency in access/storage data
- It stores copies of a blob at sites closer to the clients that use it
- After the first time a particular blob is accessed by a user, the CDN stores a copy of that blob at a location that's geographically close to that user.
- The next time this blob is accessed, its contents will be delivered from the cache



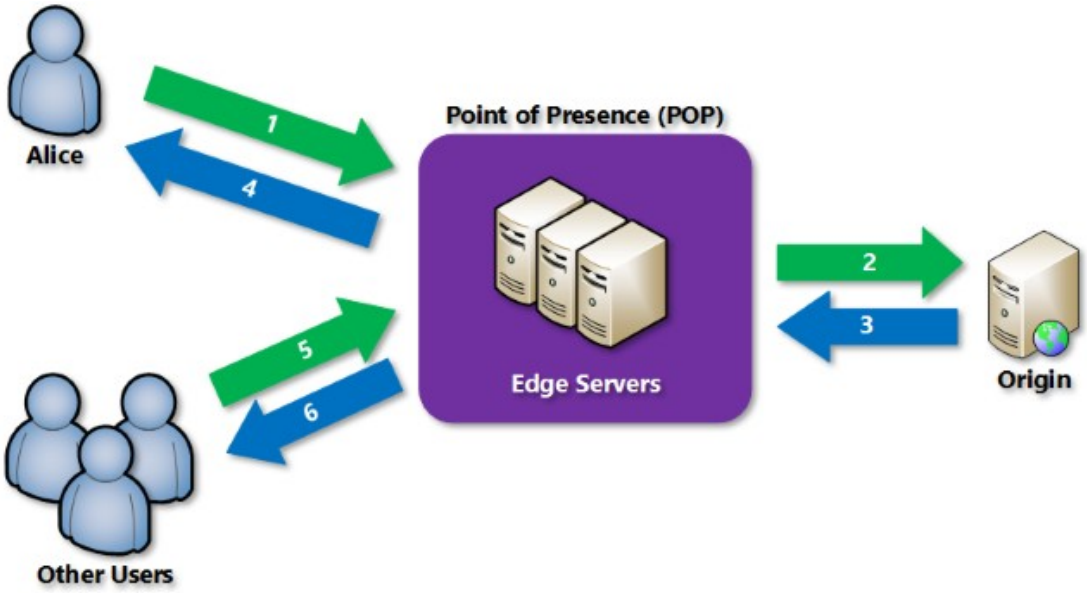
Content Delivery Network (CDN)

- A user (Alice) requests a file (also called an asset) using a URL with a special domain name, such as *<endpointname>.azureedge.net*.
- DNS routes the request to the best performing Point-of-Presence (POP) location. Usually this is the POP that is geographically closest to the user.
- If the edge servers in the POP do not have the file in their cache, the edge server requests the file from the origin.



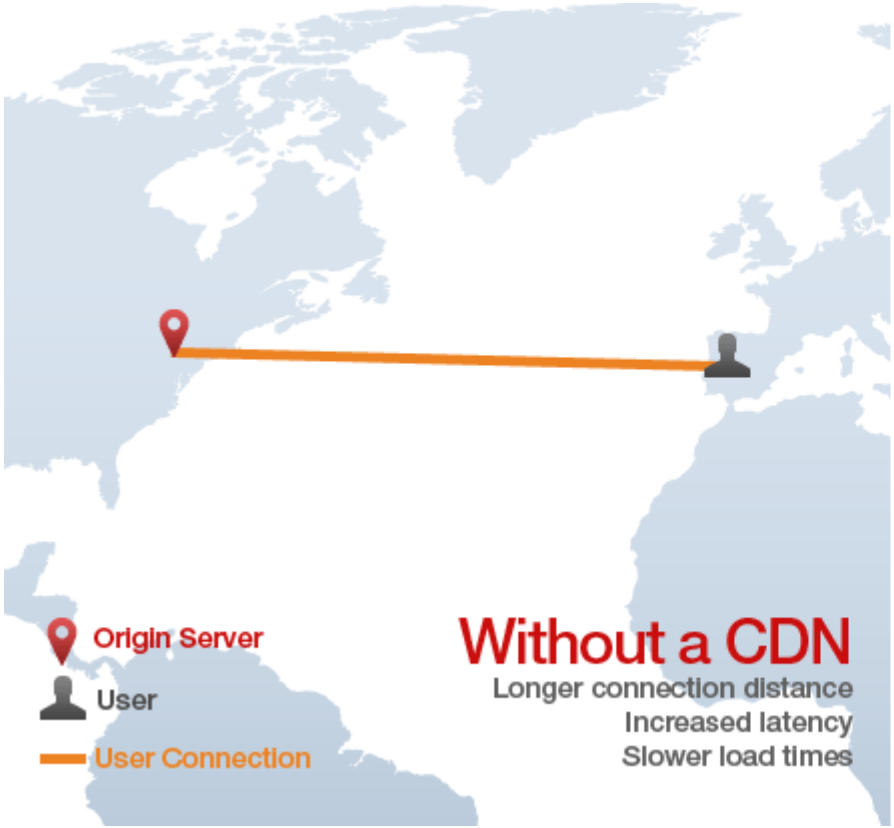
Content Delivery Network (CDN)

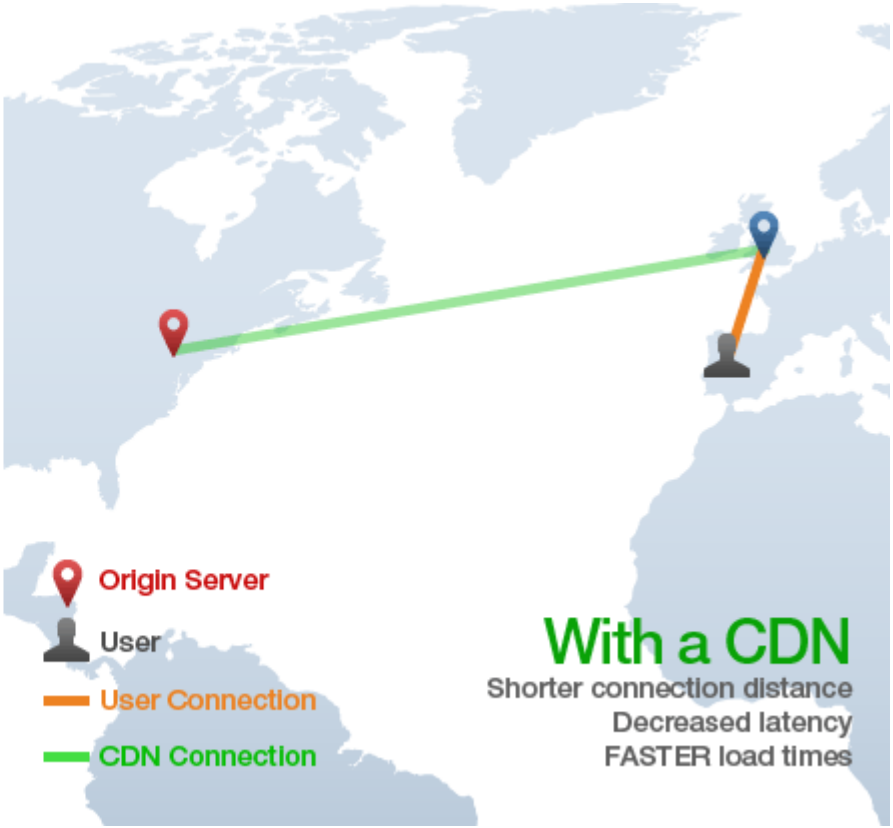
- The origin returns the file to the edge server, including optional HTTP headers describing the file's Time-to-Live (TTL).
- The edge server caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server until the TTL expires. If the origin didn't specify a TTL, the default TTL is seven days.
- Additional users may then request the same file using that same URL, and may also be directed to that same POP. If the TTL for the file hasn't expired, the edge server returns the file from the cache. This results in a faster, more responsive user experience.



Content Delivery Network (CDN)

- With a CDN, users from a European point of origin will be able to download your site's static content faster from a closer server node.





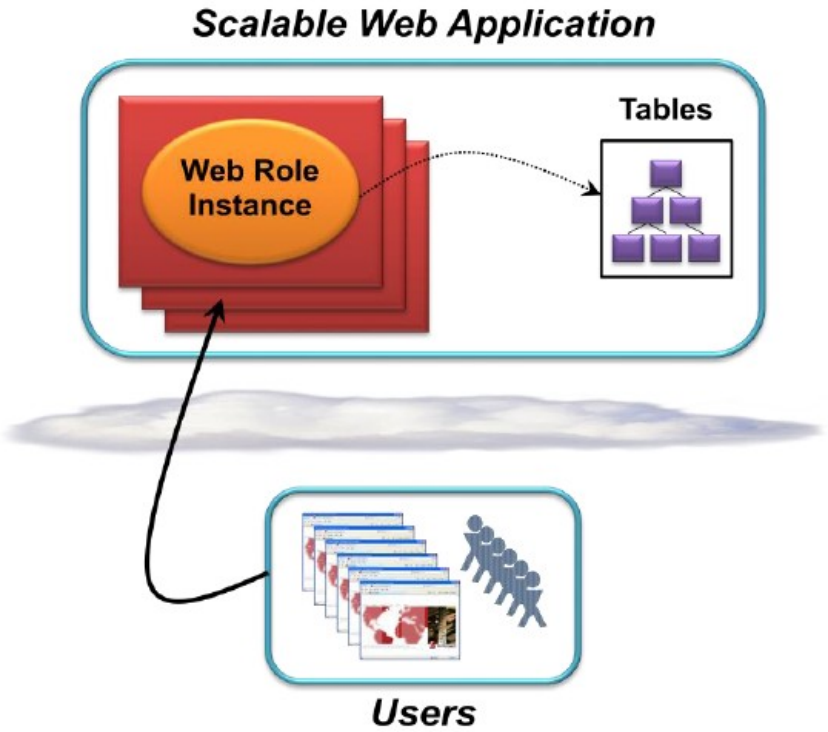
Applicative Examples

- A Scalable Web Application
- A Parallel Processing Application
- A Scalable Web Application with Background Processing

…

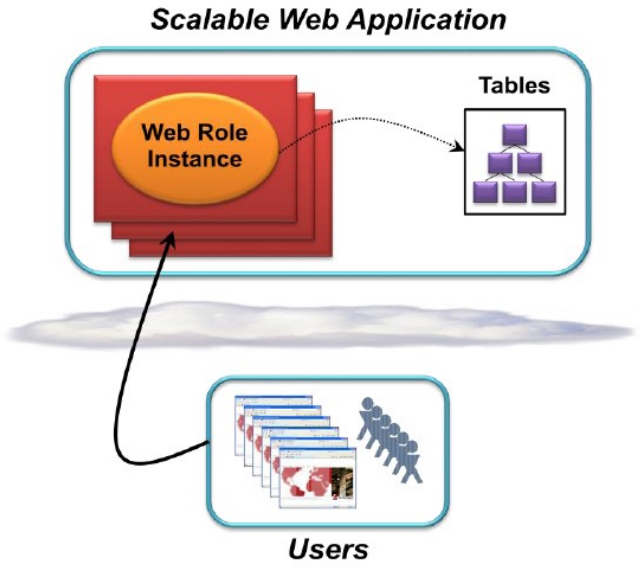
A Scalable Web Application

- Scenario: an organization wishes to create an Internet-accessible Web application
  - needs to handle a large number of simultaneous users
  - load will vary significantly: occasional spikes in the midst of long periods of lower usage
  - requires always having enough machines on hand
  - to handle the peaks, even though most of those systems go unused most of the time



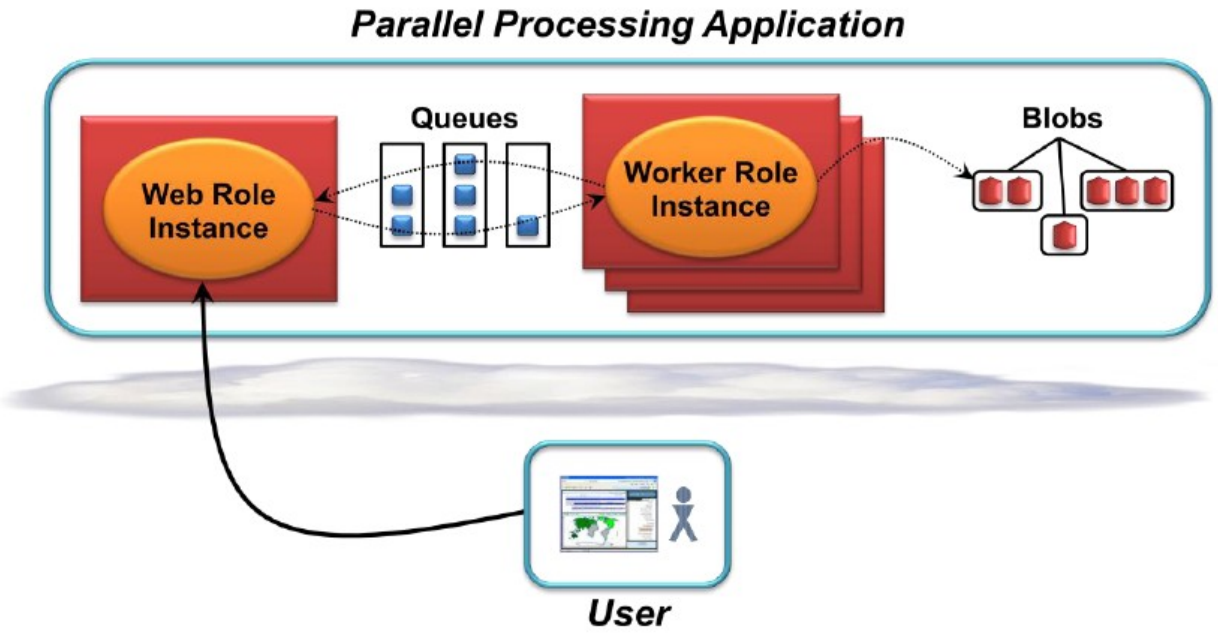
A Scalable Web Application

- Cloud solution
  - the organization running it can expand/reduce the number of instances when needed
  - the Cloud charging is usage-based (you pay per hour for each instance), so this is likely to be cheaper than maintaining lots of mostly unused machines
- Microsoft Azure
  - the developer specifies how many instances of the Web role should run
  - the Microsoft Azure fabric controller creates this number of VMs and also monitors these instances, making sure that the requested number is always available
  - for data storage, the application uses Microsoft Azure storage tables, which provide scale-out storage capable of handling very large amounts of data



A Parallel Processing Application

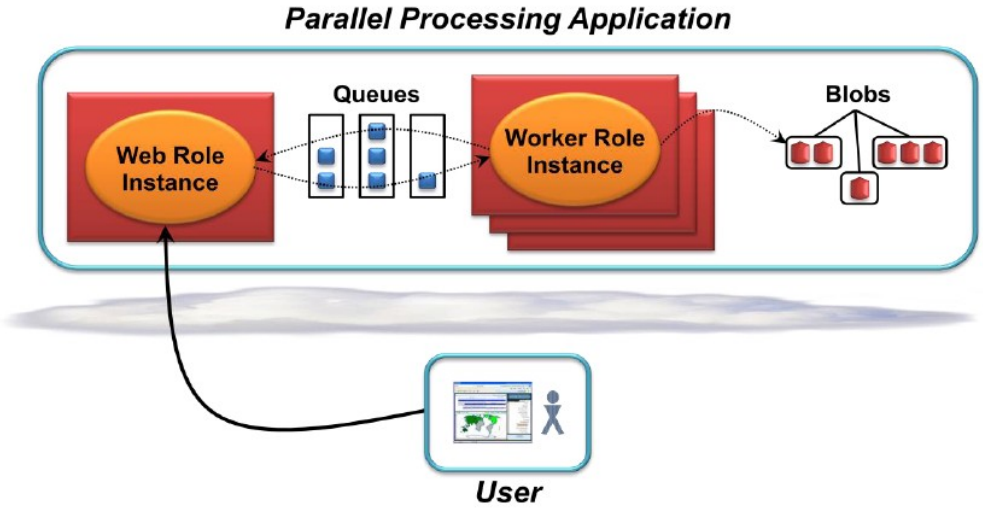
- Scenario: an organization that occasionally needs lots of computing power for a parallel processing application
  - Applications: financial modeling at a bank, rendering at a film special effects house, new drug development in a pharmaceutical company
  - Requires a large cluster of machines to meet this occasional need, but this is expensive (in terms of hardware and management)



A Parallel Processing Application

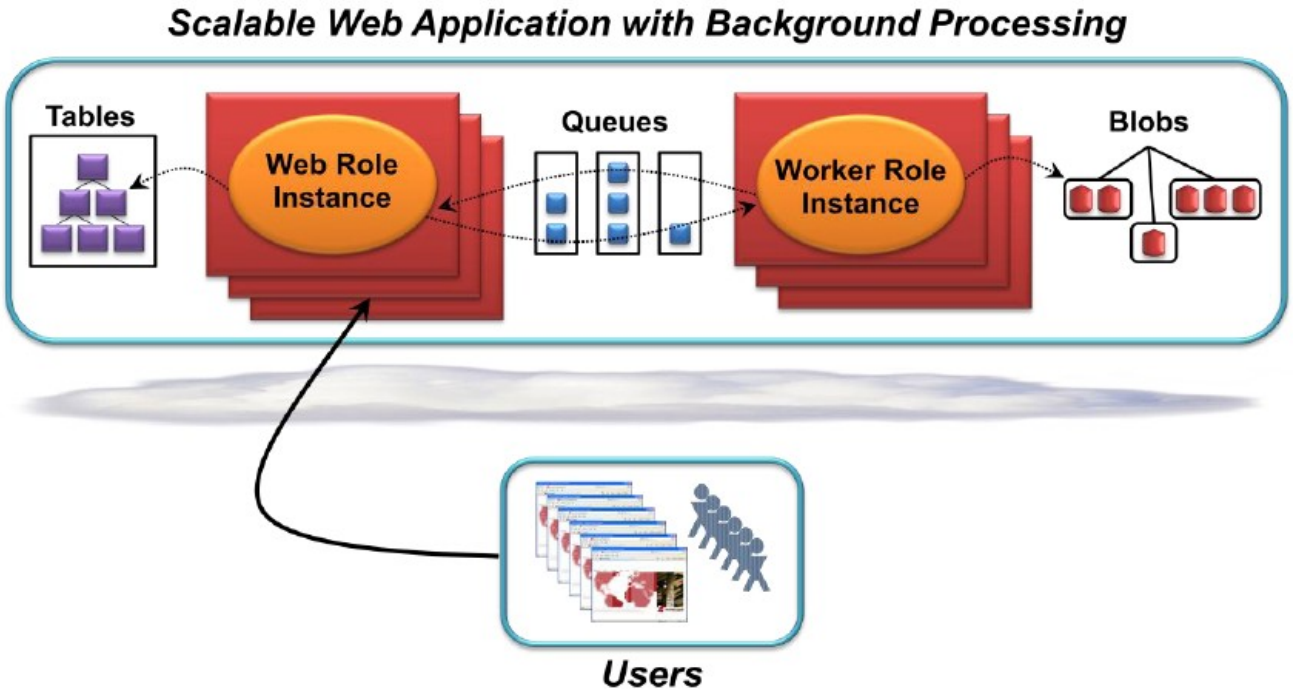
- Cloud solution
  - the Cloud can provide these resources as needed, offering an on-demand compute cluster

- Microsoft Azure
  - The user relies on a single Web role instance to handle the application
  - More Worker roles can in parallel create and execute sub-tasks of the whole application, whose data can be stored in Microsoft Azure blobs (parallel applications commonly use large datasets)
  - Through the Web role interface, the user might determine how many Worker instances should run, start and stop those instances, access results, and more
  - Communication between the Web role instance and the Worker role instances relies on Microsoft Azure storage queues.



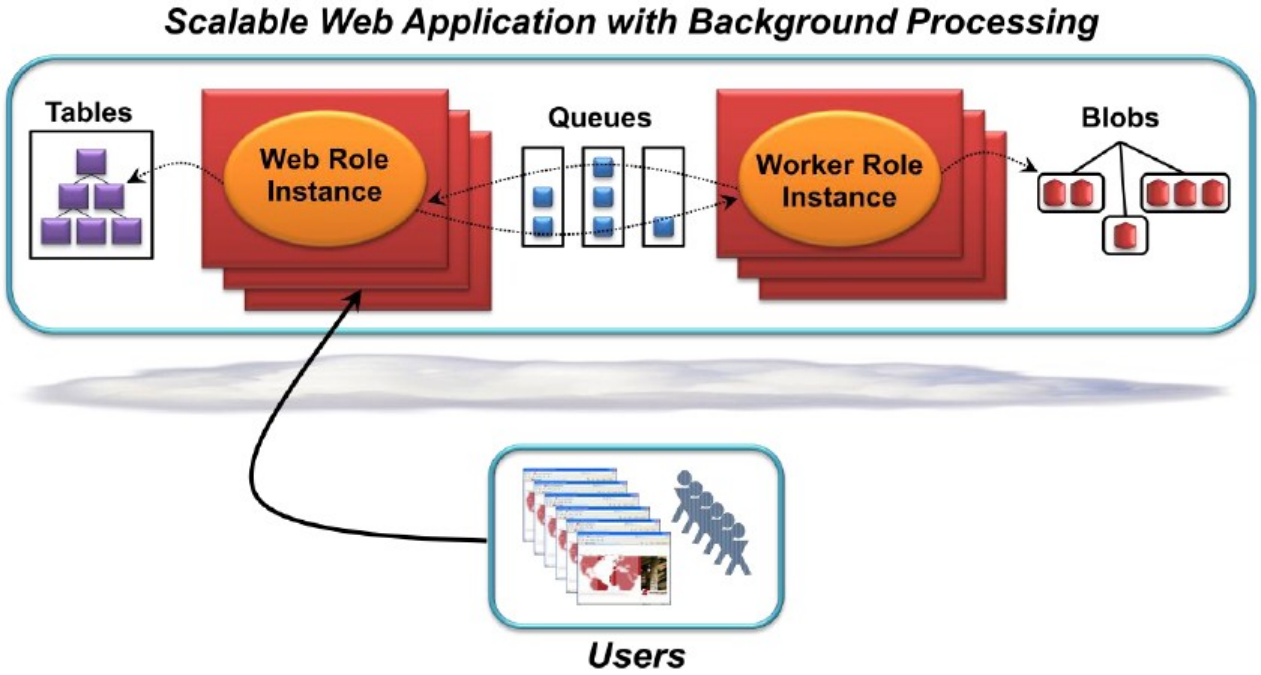
A Scalable Web Application with Background Processing

- Scenario: Web-accessible software that runs in the background
  - Example: a Web application for video sharing, that accepts browser requests (i.e., from a large number of simultaneous users)
  - In order to avoid user waiting, the part of the application that accepts browser requests (i.e., video upload) should be able to initiate a background task that carries out this work.



A Scalable Web Application with Background Processing

- Microsoft Azure
  - uses some number of Web role instances to handle user requests (likes previous application)
  - to support a large number of simultaneous users, it uses tables to store their profile information
  - for background processing, it relies on Worker role instances, passing them tasks via queues (in this example, those Worker role instances work on blob data, but other approaches are also possible)
  - this example shows how an application is built by a combination of many basic MS Azure capabilities (Web role instances, Worker role instances, blobs, tables, and queues)
  - while not every application needs all of these features, having them all available is essential for supporting more complex scenarios like this one



Documentation on MS Azure

- Azure documentation
  - http://azure.microsoft.com/en-us/documentation/
- Microsoft Azure Tutorial for Beginners
  - http://windowsazure4beginner.com/
- .NET Developer Center (download SDK Azure)
  - http://azure.microsoft.com/en-us/develop/net/