AZURE FUNDAMENTALS

Azure architecture and service guarantees

-Microsoft Learning

Disclaimer: The information contained in this document is completely owned and published by Microsoft Learning. I claim no credits to any of the content as I have just accumulated the available information for ease of read and learning. This document is/will/should not be used for any revenue systems.

Contents

Introduction

You're a small business owner with a great set of web-based services that your clients love. The one difficulty you face is some clients are experiencing a network lag accessing your services from distant locations. This problem used to be expensive to solve - requiring new data centers and costly networks to connect them. The emergence of cloud computing has made the solution easy.

Microsoft Azure provides a reliable, redundant, energy-efficient infrastructure that spans more than 100 highly secure facilities worldwide, linked by one of the largest networks on earth. Azure allows you to gain **global reach** with **local presence**, keep your data secure and compliant with local laws, and have a reduced carbon footprint with Microsoft's environment-friendly datacenters.

Datacenters and Regions in Azure

Microsoft Azure is made up of datacenters located around the globe. When you leverage a service or create a resource such as a SQL database or virtual machine, you are using physical equipment in one or more of these locations.

The specific datacenters aren't exposed to end users directly; instead, Azure organizes them into *regions*.

What is a region?

A **region** is a geographical area on the planet containing at least one, but potentially multiple datacenters that are nearby and networked together with a low-latency network. Azure intelligently assigns and controls the resources within each region to ensure workloads are appropriately balanced.

When you deploy a resource in Azure, you will often need to choose the region where you want your resource deployed.

Important

Some services or virtual machine features are only available in certain regions, such as specific virtual machine sizes or storage types. There are also some global Azure services that do not require you to select a particular region, such as Microsoft Azure Active Directory, Microsoft Azure Traffic Manager, and Azure DNS.

A few examples of regions are *West US*, *Canada Central*, *West Europe*, *Australia East*, and *Japan West*. Here's a view of all the available regions as of December 2018:

Why is this important?

Azure has more global regions than any other cloud provider. This gives you the flexibility to bring applications closer to your users no matter where they are. It also provides better scalability, redundancy, and preserves data residency for your services.

Special Azure regions

Azure has specialized regions that you might want to use when building out your applications for compliance or legal purposes. These include:

- *US DoD Central*, *US Gov Virginia*, *US Gov Iowa* and more: These are physical and logical network-isolated instances of Azure for US government agencies and partners. These datacenters are operated by screened US persons and include additional compliance certifications.
- *China East*, *China North* and more: These regions are available through a unique partnership between Microsoft and 21Vianet, whereby Microsoft does not directly maintain the datacenters.

Regions are what you use to identify the location for your resources, but there are two other terms you should also be aware of: *geographies* and *availability zones*.

Geographies

Azure divides the world into *geographies* that are defined by geopolitical boundaries or country borders. An Azure geography is a discrete market typically containing two or more regions that preserve data residency and compliance boundaries. This division has several benefits.

- Geographies allow customers with specific data residency and compliance needs to keep their data and applications close.
- Geographies ensure that data residency, sovereignty, compliance, and resiliency requirements are honored within geographical boundaries.
- Geographies are fault-tolerant to withstand complete region failure through their connection to dedicated high-capacity networking infrastructure.

Tip

Data residency refers to the physical or geographic location of an organization's data or information. It defines the legal or regulatory requirements imposed on data based on the country or region in which it resides and is an important consideration when planning out your application data storage.

Geographies are broken up into the following areas:

- Americas
- Europe
- Asia Pacific
- Middle East and Africa

Each region belongs to a single geography and has specific service availability, compliance, and data residency/sovereignty rules applied to it. Check the documentation for more information (a link is available in the summary unit).

Availability Zones

You want to ensure your services and data are redundant so you can protect your information in case of failure. When you are hosting your infrastructure, this requires creating duplicate hardware environments. Azure can help make your app highly available through *Availability Zones*.

What is an Availability Zone?

Availability Zones are physically separate datacenters within an Azure region.

Each Availability Zone is made up of one or more datacenters equipped with independent power, cooling, and networking. It is set up to be an *isolation boundary*. If one zone goes down, the other continues working. Availability Zones are connected through high-speed, private fiber-optic networks.

Supported regions

Not every region has support for Availability Zones. The following regions have a minimum of three separate zones to ensure resiliency.

- Central US
- East US 2
- West US 2
- West Europe
- France Central
- North Europe
- Southeast Asia

Tip

The list of supported regions is expanding - check the documentation for the latest information.

Using Availability Zones in your apps

You can use Availability Zones to run mission-critical applications and build high-availability into your application architecture by co-locating your compute, storage, networking, and data resources within a zone and replicating in other zones. Keep in mind that there could be a cost to duplicating your services and transferring data between zones.

Availability Zones are primarily for VMs, managed disks, load balancers, and SQL databases. Azure services that support Availability Zones fall into two categories:

- **Zonal services** – you pin the resource to a specific zone (for example, virtual machines, managed disks, IP addresses)
- **Zone-redundant services** – platform replicates automatically across zones (for example, zone-redundant storage, SQL Database).

Check the documentation to determine which elements of your architecture you can associate with an Availability Zone.

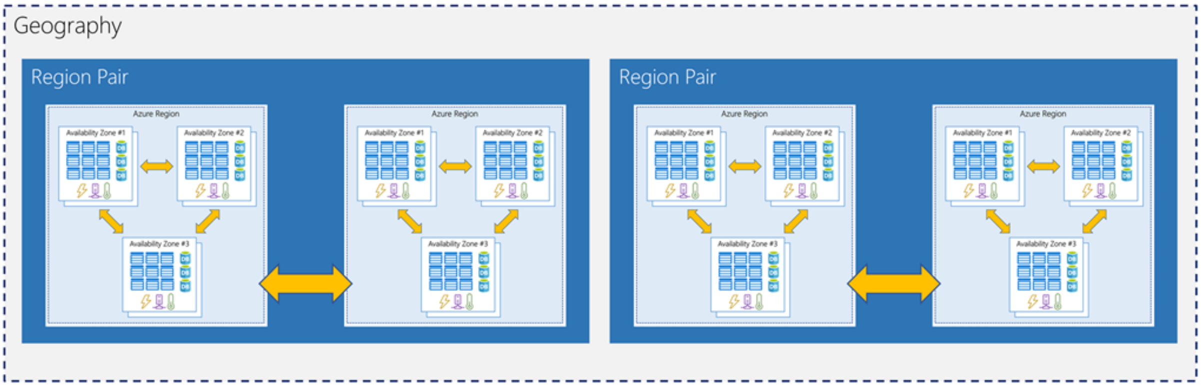
Region Pairs

Availability zones are created using one or more datacenters, and there are a minimum of three zones within a single region. However, it's possible that a large enough disaster could cause an outage big enough to affect even two datacenters. That's why Azure also creates *region pairs*.

What is a region pair?

Each Azure region is always paired with another region within the same geography (such as US, Europe, or Asia) at least **300 miles away**. This approach allows for the replication of resources (such as virtual machine storage) across a geography that helps reduce the likelihood of interruptions due to events such as natural disasters, civil unrest, power outages, or physical network outages affecting both regions at once.

Examples of region pairs in Azure are West US paired with East US, and SouthEast Asia paired with East Asia.



Since the pair of regions is directly connected and far enough apart to be isolated from regional disasters, you can use them to provide reliable services and data redundancy. Some services offer automatic geo-redundant storage using region pairs.

Additional advantages of region pairs include:

- If there's an extensive Azure outage, one region out of every pair is prioritized to help reduce the time it takes to restore them for applications.
- Planned Azure updates are rolled out to paired regions one region at a time to minimize downtime and risk of application outage.
- Data continues to reside within the same geography as its pair (except for Brazil South) for tax and law enforcement jurisdiction purposes.

Having a broadly distributed set of datacenters allows Azure to provide a high guarantee of availability. Let's explore what that means.

Service Level Agreements for Azure

Microsoft maintains its commitment to providing customers with high-quality products and services by adhering to comprehensive operational policies, standards, and practices. Formal documents called *Service-Level Agreements* (SLAs) capture the specific terms that define the performance standards that apply to Azure.

- SLAs describe Microsoft's commitment to providing Azure customers with specific performance standards.
- There are SLAs for individual Azure products and services.
- SLAs also specify what happens if a service or product fails to perform to a governing SLA's specification.

Important

Azure does not provide SLAs for most services under the *Free* or *Shared* tiers. Also, free products such as Azure Advisor do not typically have an SLA.

SLAs for Azure products and services

There are three key characteristics of SLAs for Azure products and services:

1. Performance Targets
2. Uptime and Connectivity Guarantees
3. Service credits

Performance Targets

An SLA defines performance targets for an Azure product or service. The performance targets that an SLA defines are specific to each Azure product and service. For example, performance targets for some Azure services are expressed as uptime guarantees or connectivity rates.

Uptime and Connectivity Guarantees

A typical SLA specifies performance-target commitments that range from 99.9 percent ("three nines") to 99.999 percent ("five nines"), for each corresponding Azure product or service. These targets can apply to such performance criteria as uptime or response times for services.

The following table lists the potential cumulative downtime for various SLA levels over different durations:

SLA %	Downtime per week	Downtime per month	Downtime per year
99	1.68 hours	7.2 hours	3.65 days
99.9	10.1 minutes	43.2 minutes	8.76 hours
99.95	5 minutes	21.6 minutes	4.38 hours
99.99	1.01 minutes	4.32 minutes	52.56 minutes
99.999	6 seconds	25.9 seconds	5.26 minutes

For example, the SLA for the Azure Cosmos DB (Database) service SLA offers 99.999 percent uptime, which includes low-latency commitments of less than 10 milliseconds on DB read operations as well as on DB write operations.

Service Credits

SLAs also describe how Microsoft will respond if an Azure product or service fails to perform to its governing SLA's specification.

For example, customers may have a discount applied to their Azure bill, as compensation for an under-performing Azure product or service. The table below explains this example in more detail.

The first column in the table below shows monthly uptime percentage SLA targets for a single instance Azure Virtual Machine. The second column shows the corresponding service credit amount you receive if the *actual* uptime is less than the specified SLA target for that month.

MONTHLY UPTIME PERCENTAGE	**SERVICE CREDIT PERCENTAGE**
< 99.9	10
< 99	25
< 95	100

Composing SLAs across services

When combining SLAs across different service offerings, the resultant SLA is called a *Composite SLA*. The resulting composite SLA can provide higher or lower uptime values, depending on your application architecture.

Calculating downtime

Consider an App Service web app that writes to Azure SQL Database. These Azure services currently have the following SLAs:

In this example, if either service fails the whole application will fail. In general, the individual probability values for each service are independent. However, the composite SLA value for this application is:

99.95 percent × 99.99 percent = 99.94 percent

This means the **combined probability of failure** is higher than the individual SLA values. This isn't surprising, because an application that relies on multiple services has more potential failure points.

Conversely, you can improve the composite SLA by creating independent fallback paths. For example, if SQL Database is unavailable, you can put transactions into a queue for processing at a later time.

With this design, the application is still available even if it can't connect to the database. However, it fails if both the database *and* the queue fail simultaneously.

If the expected percentage of time for a simultaneous failure is **0.0001 × 0.001**, the composite SLA for this combined path of a database *or* queue would be:

1.0 − (0.0001 × 0.001) = 99.99999 percent

Therefore, if we add the queue to our web app, the total composite SLA is:

99.95 percent × 99.99999 percent = ~99.95 percent

Notice we've improved our SLA behavior. However, there are trade-offs to using this approach: the application logic is more complicated, you are paying more to add the queue support, and there may be data-consistency issues you'll have to deal with due to retry behavior.

Improve your app reliability

You can use SLAs to evaluate how your Azure solutions meet business requirements and the needs of your clients and users. By creating your own SLAs, you can set performance targets to suit your specific Azure application. This approach is known as an *Application SLA*.

Understand your app requirements

Building an efficient and reliable Azure solution requires knowing your workload requirements. You can then select Azure products and services, and provision resources according to those requirements. It's important to understand the Azure SLAs that define performance targets for the Azure products and services within your solution. This understanding will help you create achievable Application SLAs.

In a distributed system, failures will happen. Hardware can fail. The network can have transient failures. It's rare for an entire service or region to experience a disruption, but even this must be planned for.

Resiliency

Resiliency is the ability of a system to recover from failures and continue to function. It's not about avoiding failures, but responding to failures in a way that avoids downtime or data loss. The goal of resiliency is to return the application to a fully functioning state following a failure. High availability and disaster recovery are two crucial components of resiliency.

When designing your architecture you need to design for resiliency, and you should perform a *Failure Mode Analysis* (FMA). The goal of an FMA is to identify possible points of failure and to define how the application will respond to those failures.

Cost and complexity vs. high availability

Availability refers to the time that a system is functional and working. Maximizing availability requires implementing measures to prevent possible service failures. However, devising preventative measures can be difficult and expensive, and often results in complex solutions.

As your solution grows in complexity, you will have more services depending on each other. Therefore, you might overlook possible failure points in your solution if you have several interdependent services.

Tip

For example: A workload that requires 99.99 percent uptime shouldn't depend upon a service with a 99.9 percent SLA.

Most providers prefer to maximize the availability of their Azure solutions by minimizing downtime. However, as you increase availability, you also increase the cost and complexity of your solution.

Tip

For example: An SLA that defines an *uptime of 99.99% only allows for about 5 minutes of total downtime per month.*

The risk of potential downtime is cumulative across various SLA levels, which means that complex solutions can face greater availability challenges. Therefore, how critical high-availability is to your requirements will determine how you handle the addition of complexity and cost to your application SLAs.

Considerations for defining application SLAs

- If your application SLA defines four 9's (99.99%) performance targets, recovering from failures by manual intervention may not be enough to fulfill your SLA. Your Azure solution must be self-diagnosing and self-healing instead.
- It is difficult to respond to failures quickly enough to meet SLA performance targets above four 9's.

- Carefully consider the time window against which your application SLA performance targets are measured. The smaller the time window, the tighter the tolerances. If you define your application SLA as hourly or daily uptime, you need to understand these tighter tolerances might not allow for achievable performance targets.

Summary

Microsoft provides more global presence than any other cloud provider with over 54 regions distributed worldwide. This infrastructure gives you the scale needed to bring your applications closer to users around the world. Azure also has dedicated regions to support government use and applications that need to be deployed in China so you can ensure data security and residency and meet compliance and resilience requirements for your customers no matter what type of business requirements you have.

Learn more

Visit the following links to learn more about some of the topics we explored in this module.

- Azure regions
- Azure geographies
- Azure Service Level Agreements
- Designing resilient applications for Azure