Core Azure Services

Scott Tremaine
Software Developer and Educator

Breakpoint Coding Tutorials

Contents

Azure Subscriptions

Azure subscriptions act as a foundational structure within Microsoft Azure, facilitating the organization, management, and billing of various resources and services. They can be visualized as containers or units of management. They are the gatekeepers that control access to Azure resources, providing a framework for assigning permissions and tracking usage. Each subscription is linked to a particular Azure account, and within this structure, you can create and manage resources, track costs, and enforce governance policies. By organizing resources into subscriptions, you gain the flexibility to isolate workloads, assign budgets, and implement access controls tailored to the needs of specific projects or departments.

Different types of Azure subscriptions cater to diverse requirements and usage scenarios, each offering unique features and benefits. These subscription models are designed to provide flexibility and scalability, accommodating the needs of individual developers, small businesses, and large enterprises alike.

Free Subscription

The Free subscription is an entry point for users new to Azure. It provides a limited amount of free resources and services for a specified period, usually 12 months. This subscription is an excellent way for individuals and organizations to explore Azure's capabilities without incurring initial costs. During this trial period, users can experiment with a range of services, including virtual machines, databases, and AI tools, all while gaining a foundational understanding of Azure's functionalities. The Free subscription also includes a set of services that remain free perpetually, allowing users to continue using essential services at no cost even after the trial period ends.

Pay-As-You-Go Subscription

The Pay-As-You-Go subscription model offers unparalleled flexibility, charging users based on their actual usage of Azure services. This model is ideal for small to medium-sized businesses and individual developers who need scalable solutions without long-term commitments. There are no upfront costs or termination fees, making it a cost-effective option for projects with variable workloads. Users only pay for what they use, whether it's storage, compute power, or data transfer, and they can easily scale their resources up or down based on demand. This flexibility ensures that businesses can efficiently manage their cloud expenditure while leveraging Azure's robust infrastructure.

Enterprise Subscription

For large organizations with extensive cloud needs, the Enterprise subscription provides a comprehensive and customizable solution. This model is designed for enterprises that require a higher level of control, support, and governance. Enterprise subscriptions typically involve a contractual agreement with Microsoft, offering benefits such as volume discounts, dedicated account management, and enhanced support options. These subscriptions also include advanced features for managing multiple accounts and subscriptions within a single organization, enabling centralized billing, consolidated reporting,

and streamlined compliance management. Additionally, enterprise customers can leverage Azure's hybrid capabilities, integrating on-premises infrastructure with Azure's cloud services to create a cohesive and flexible IT environment.

Each subscription type serves a specific purpose, allowing users to choose the model that best fits their needs. By understanding the distinct characteristics and benefits of these subscription models, users can make informed decisions that align with their organizational goals and usage patterns.

Management Groups

At their core, management groups are designed to facilitate streamlined governance, security, and policy management across an organization's Azure resources. Understanding the structure and hierarchy of management groups is fundamental to leveraging their full potential.

Hierarchy of Management Groups

The structure of management groups in Azure is hierarchical, allowing for a tiered approach to resource management. At the top of this hierarchy is the Root Management Group. This is the default parent group for all management groups and subscriptions within an Azure Active Directory (Azure AD) tenant. The root management group serves as the apex point of management, ensuring that all resources under it inherit its policies and settings unless explicitly overridden.

Beneath the root management group, administrators can create multiple management groups. These management groups can further contain other management groups or subscriptions, forming a tree-like structure. This hierarchical model enables organizations to mirror their corporate structure or business units within Azure.

For example, a company might have a root management group named "Corporate," under which there are separate management groups for "Finance," "IT," and "Marketing." Each of these departments can have their own management groups, such as "IT-Infrastructure" and "IT-Development," each containing various subscriptions.

Policy and Access Control Inheritance

Policies, role-based access control (RBAC) assignments, and governance rules applied at a higher level are automatically inherited by all lower levels. This inheritance ensures consistency and compliance across the entire organization, significantly reducing the administrative overhead involved in managing policies and access controls individually for each subscription.

For instance, a security policy applied at the "Corporate" level will trickle down to all management groups and subscriptions under it.

Azure Resource Manager

Azure Resource Manager (ARM) plays a pivotal role in managing resources within Microsoft Azure. It acts as the deployment and management service for Azure, providing a consistent management layer that enables you to create, update, and delete resources in your Azure account. ARM simplifies and streamlines the process, allowing for a more organized and efficient approach to resource management.

Resource Groups

The first key concept to understand is resource groups. These are logical containers into which Azure resources such as virtual machines, databases, and networking interfaces are deployed and managed. The beauty of resource groups lies in their ability to consolidate resources that share a common lifecycle, making it easier to manage and monitor them as a single entity. For example, all resources for a web application – the server, database, and networking components – can be placed in a single resource group. This organization not only simplifies management but also enhances automation and monitoring.

ARM Templates

Central to ARM's power and flexibility is the use of ARM templates. ARM templates are JSON (JavaScript Object Notation) files that define the infrastructure and configuration for your project. They enable you to declaratively define the resources you need and their configurations. The benefits of using ARM templates are manifold. They allow for consistent deployments, as the template can be reused to create the same resources repeatedly across different environments. Furthermore, ARM templates facilitate infrastructure as code (IaC), a practice that brings the benefits of code-based management to infrastructure provisioning, including version control and easier collaboration.

Creating ARM Templates

Creating an ARM template involves specifying the resources and their properties in a structured format. The template is then used by ARM to deploy the defined resources in a desired state. This method of deployment ensures that resources are created in a consistent and predictable manner. ARM templates also support parameters, allowing for the creation of dynamic and reusable templates. Parameters can be used to pass values such as resource names, sizes, and configurations, making the templates flexible and adaptable to various scenarios.

Deploying ARM Templates

In practice, deploying an ARM template is a straightforward process. Once the template is defined, it can be deployed through multiple methods, including the Azure portal, Azure PowerShell, Azure CLI, or programmatically using Azure SDKs. This versatility ensures that you can integrate ARM template deployment into your existing workflows and automation pipelines with ease.

Dependencies in ARM Templates

Furthermore, ARM templates support the concept of dependencies. When you define a resource that depends on another resource, ARM ensures that the dependent resource is created first. This feature is particularly useful in complex deployments where certain resources need to be in place before others can be provisioned. By defining dependencies in the template, you can manage these relationships effectively and avoid deployment errors.

Monitoring and Logging

ARM also provides extensive monitoring and logging capabilities, allowing you to track the status of your deployments and operations. Azure Resource Manager logs detailed information about the requests made to it, which can be invaluable for troubleshooting and auditing purposes. This transparency ensures that you can maintain control over your environment and quickly identify and resolve any issues that arise during resource management operations.

Azure Regions

The vast, intricate network of data centers known as Azure regions forms the backbone of Microsoft's cloud infrastructure. These regions are strategically distributed across the globe, ensuring that Azure services are readily available and perform optimally for users no matter where they are located.

Regional Distribution

Each Azure region consists of multiple data centers situated within a specific geographic area. These regions are distributed across various continents to serve local and international customers efficiently. For instance, in North America, you will find regions such as East US, West US, and Central US. Similarly, Europe hosts regions like North Europe and West Europe, while Asia boasts East Asia and Southeast Asia regions. This geographical spread ensures that no matter where a business operates, there is a nearby Azure region capable of providing robust cloud services.

Strategic Placement

The strategic placement of these regions is designed to meet local regulatory requirements, enhance data sovereignty, and reduce latency. By having data centers close to the end-users, Azure minimizes the time it takes for data to travel between the user and the server, thereby improving the performance and responsiveness of applications. Moreover, local regions help comply with various data protection laws that mandate data to reside within specific geographical boundaries, offering peace of mind to businesses concerned about regulatory compliance.

Paired Regions

One of the features of Azure's regional architecture is the concept of paired regions. Microsoft has established a system where each Azure region is paired with another region within the same geographic vicinity but sufficiently distanced to mitigate regional disasters. For example, the East US region is paired with the West US region, and North Europe is paired with West Europe.

Paired regions offer several significant advantages, particularly in the realm of disaster recovery. In the event of a large-scale disaster that affects one region, its paired region can take over the services, ensuring that applications remain available and data is protected. This automatic failover capability is a critical component of Azure's high availability and disaster recovery strategy.

Furthermore, updates and maintenance operations are carefully managed across paired regions to minimize the risk of simultaneous downtime. Microsoft ensures that planned updates are rolled out sequentially rather than simultaneously to both regions in a pair. This practice ensures that even during maintenance periods, one region remains fully operational, providing continuous service availability.

Compute Services

Overview of Virtual Machines (VMs)

In essence, virtual machines are software emulations of physical computers. They operate in a self-contained environment, mimicking the functionalities of an actual computer. This means you can run an operating system and applications on a VM just as you would on a physical machine, but with added flexibility and efficiency.

Virtual machines are instrumental in modern computing because they allow for better resource utilization and scalability. By leveraging VMs, businesses can run multiple operating systems on a single physical server, isolating applications to prevent conflicts and ensuring better use of hardware resources. This isolation also enhances security, as each VM operates independently from others.

On the Microsoft Azure platform, VMs offer a plethora of configurations, catering to various needs—from general-purpose VMs for everyday workloads to specialized VMs optimized for machine learning or large-scale data processing. When you deploy a VM in Azure, you can choose the size and capacity that match your specific requirements, allowing for cost-effective scaling. This flexibility is crucial for businesses that need to adjust their computing power dynamically based on demand.

Azure provides comprehensive management tools for VMs, making it easier to handle tasks such as deploying, configuring, and monitoring your virtual environments. These tools include Azure Resource Manager, which allows for the deployment and management of VMs through templates, ensuring consistency and efficiency in your cloud operations.

Azure App Services and Azure Functions

Transitioning from VMs to more advanced and specialized compute services, Azure App Services and Azure Functions represent the evolution of cloud computing towards platform-as-a-service (PaaS) and serverless computing.

Azure App Services

Azure App Services is a fully managed platform designed to build, deploy, and scale web apps, mobile backends, and RESTful APIs. It abstracts much of the underlying infrastructure management, allowing developers to focus on the application logic rather than the server maintenance. With App Services, you can deploy applications written in various programming languages, including .NET, Java, Ruby, Node.js, PHP, and Python. This flexibility ensures that you can use the best tools and frameworks for your specific project needs.

One of the standout features of Azure App Services is its seamless integration with other Azure services and DevOps tools. Continuous deployment options enable automatic updates from your source code repository, ensuring that your applications are always up-to-date with the latest changes. Furthermore, built-in monitoring and diagnostics tools provide insights into application performance, helping you to identify and resolve issues promptly.

Azure Functions

Azure Functions takes this a step further by introducing the concept of serverless computing. Serverless doesn't mean there are no servers involved, but rather that the server management and capacity planning are handled entirely by Azure. Azure Functions allows you to execute code in response to various triggers, such as HTTP requests, timers, or messages from other Azure services. This model is particularly useful for scenarios that require scalable and event-driven processing.

With Azure Functions, you only pay for the compute resources used during the execution of your functions, which can result in significant cost savings for intermittent workloads. The functions can be written in a variety of programming languages, and Azure provides extensive templates and bindings to simplify the development process. Additionally, the serverless model facilitates rapid development and deployment, enabling you to iterate quickly and bring solutions to market faster.

Both Azure App Services and Azure Functions offer robust security features, including authentication and authorization mechanisms, to protect your applications and data. They also support scalability options, ensuring that your applications can handle varying loads without manual intervention.

Networking Services

Networking services provided by Azure offers an array of networking services, including Virtual Networks (VNets), subnets, Network Security Groups (NSGs), Azure Load Balancer, and Azure Application Gateway. These components form the backbone of your

cloud infrastructure, enabling seamless communication, robust security, and optimal performance.

Virtual Networks (VNet), Subnets, and Network Security Groups (NSGs)

A Virtual Network (VNet) in Azure simulates a traditional network in your on-premises environment but with the added flexibility and scalability of the cloud. VNets enable Azure resources, such as Virtual Machines (VMs), to securely communicate with each other, the internet, and on-premises networks. Setting up a VNet is the first step in building your network infrastructure in Azure.

When you create a VNet, you define an IP address space, which determines the range of IP addresses that resources within the VNet can use. This IP address space is further segmented into subnets, each representing a smaller range of IP addresses within the VNet. Subnets allow you to isolate different parts of your application or workload, providing an additional layer of security and organizational structure.

For instance, you might have a subnet dedicated to your web servers, another for your application servers, and a separate one for your database servers. This segregation helps manage traffic flow and apply distinct security policies to each subnet.

Network Security Groups (NSGs) play a pivotal role in securing your VNet. NSGs contain a list of security rules that allow or deny inbound and outbound traffic to resources connected to Azure VNets. By associating NSGs with subnets or individual network interfaces, you can control the traffic flowing into and out of your network, effectively creating a robust security perimeter around your resources. For example, you might configure an NSG to allow only HTTP and HTTPS traffic to your web servers while blocking all other inbound traffic.

Azure Load Balancer and Azure Application Gateway

As your application scales, managing the distribution of incoming traffic becomes critical. This is where Azure Load Balancer and Azure Application Gateway come into play, ensuring high availability and reliability of your applications by distributing traffic efficiently.

Azure Load Balancer operates at the transport layer (Layer 4) of the OSI model, handling all incoming and outgoing traffic based on the IP address and port number. It provides both public and internal load balancing. Public Load Balancers distribute incoming internet traffic across multiple VMs, ensuring no single VM becomes a bottleneck. Internal Load Balancers, on the other hand, manage traffic within a VNet, directing it to healthy instances of your service running across multiple VMs or availability zones.

For example, consider an e-commerce application where customer requests need to be handled by multiple web servers. An Azure Load Balancer can distribute incoming requests evenly across all web servers, ensuring optimal resource utilization and preventing

any single server from being overwhelmed. This not only improves the application's performance but also enhances its reliability by routing traffic only to healthy VMs.

Azure Application Gateway, in contrast, operates at the application layer (Layer 7) of the OSI model. It provides advanced routing capabilities, including URL-based routing and SSL termination, making it ideal for modern web applications. With URL-based routing, Application Gateway can direct traffic to different backend pools based on the URL path of the request. This is particularly useful for microservices architectures, where different services handle different parts of an application.

Imagine a scenario where your application comprises various microservices such as user authentication, product catalog, and order processing. By configuring an Application Gateway, you can route traffic to the appropriate service based on the URL path, ensuring each request is handled by the correct backend pool. Additionally, Application Gateway supports SSL termination, offloading the SSL decryption work from your application servers. This not only simplifies your SSL management but also enhances the performance of your backend servers.

Both Azure Load Balancer and Azure Application Gateway include health probes to monitor the status of your application instances. These probes periodically check the health of your services, ensuring that traffic is only directed to healthy instances.

Azure Storage Services

Azure Storage provides a scalable, durable, and highly available solution to store and manage large amounts of data in the cloud. Among the various types of storage services offered by Azure, each is designed to meet specific data storage needs, whether you're storing large amounts of unstructured data, sharing files between virtual machines, or queuing messages for asynchronous processing.

Blob Storage

Azure Blob Storage is designed to store unstructured data, which can include any type of text or binary data, such as documents, media files, or log files. This service is highly scalable, capable of handling massive amounts of data efficiently. Blob Storage is particularly useful for applications that need to store large quantities of data that don't fit neatly into traditional databases. It's ideal for serving images or documents directly to a browser, storing files for distributed access, or streaming audio and video.

File Storage

For those familiar with traditional file systems, Azure File Storage offers a fully managed file share in the cloud that can be accessed via the Server Message Block (SMB) protocol. This means you can mount Azure file shares on Windows, Linux, and macOS, allowing applications to share files across various environments seamlessly. Azure File Storage is particularly beneficial for applications that need to share data between multiple virtual machines or for organizations looking to replace or supplement on-premises file servers.

Queue Storage

Queue Storage provides a reliable messaging solution for asynchronous communication between different components of an application. This service can store large numbers of messages that can be accessed from anywhere in the world via authenticated calls using HTTP or HTTPS. Queue Storage is an excellent choice for building scalable and decoupled applications, where different parts of the system can work independently and interact through message queues. This approach enhances the resilience and scalability of complex systems by ensuring components can handle loads independently.

Table Storage

Azure Table Storage offers a NoSQL key-value store for rapid development and storage of massive amounts of structured data. It's an ideal solution for applications requiring a flexible schema, such as web applications, IoT solutions, and more. Table Storage provides fast and cost-effective access to data with a schema-less design, which can be advantageous when dealing with diverse data types or when you need to perform quick lookups and complex queries.

Azure Storage Accounts and Redundancy Options

When setting up your storage in Azure, it all begins with creating an Azure Storage account. This account acts as a container for multiple types of storage services, including Blob, File, Queue, and Table storage. By centralizing these services within a single account, Azure offers a unified management interface and billing structure, simplifying the administration of your storage resources.

Azure provides several redundancy options to ensure the durability and availability of your data. These options are crucial for disaster recovery and business continuity planning.

Locally Redundant Storage (LRS)

LRS replicates your data three times within a single data center. This option offers the least expensive redundancy but only protects against hardware failures, not data center outages.

Zone-Redundant Storage (ZRS)

ZRS replicates your data synchronously across three different zones within a single region. Each zone is a separate physical location with independent power, cooling, and networking. This redundancy model provides higher availability and protection against data center-level failures.

Geo-Redundant Storage (GRS)

GRS replicates your data to a secondary region hundreds of miles away from the primary location. This model provides a higher level of durability by protecting against regional outages. Your data is first replicated three times within the primary region using LRS, then asynchronously replicated to the secondary region, where it is stored using LRS as well.

Read-Access Geo-Redundant Storage (RA-GRS)

RA-GRS offers the same replication as GRS but with the added benefit of read access to the data in the secondary region. This means that in the event of an outage in the primary region, you can still read your data from the secondary region, ensuring greater availability and disaster recovery capabilities.

Database Services

Azure offers a variety of database services designed to meet diverse needs, from structured SQL databases to flexible NoSQL solutions.

Azure SQL Database

Azure SQL Database is a fully managed relational database service that offers robust features and high performance, making it an ideal choice for a wide range of applications. Built on the familiar SQL Server engine, Azure SQL Database ensures that developers and database administrators can leverage their existing SQL Server knowledge while benefiting from the cloud's flexibility and scalability.

The service provides automatic updates, patching, and backups, ensuring that the database remains secure and up-to-date without manual intervention. Moreover, Azure SQL Database offers dynamic scalability, allowing you to adjust the compute and storage resources based on the workload demands. This feature is particularly useful for applications with variable or unpredictable usage patterns.

High availability and disaster recovery are integral components of Azure SQL Database. The service provides built-in high availability through features like geo-replication and automated backups. Geo-replication allows you to create readable replicas of your database in different geographical locations, ensuring data redundancy and quick recovery in the event of a regional outage.

Security is another strong suit of Azure SQL Database. The service includes advanced threat protection, encryption at rest and in transit, and compliance with various industry standards. These security features help protect sensitive data and meet regulatory requirements.

Azure Cosmos DB

Azure Cosmos DB is a globally distributed, multi-model database service designed to handle massive amounts of data with low latency. It supports multiple data models, including document, key-value, graph, and column-family, making it a versatile choice for diverse applications.

One of the standout features of Azure Cosmos DB is its global distribution capabilities. You can distribute your data across multiple Azure regions, providing low-latency access to users around the world. This global distribution is complemented by multi-master

replication, which allows for write operations to be performed on any replica. This ensures high availability and fast data access regardless of the user's location.

Consistency is another aspect of distributed databases, and Azure Cosmos DB offers five consistency levels to balance between performance and data consistency: strong, bounded staleness, session, consistent prefix, and eventual. This flexibility allows you to choose the appropriate consistency level based on your application's requirements.

Azure Cosmos DB is also designed for high throughput. It provides automatic scaling of throughput and storage, allowing you to handle varying workloads efficiently. The service's performance is further enhanced by features like SSD-backed storage and a fully managed indexing system, which optimizes query performance without manual intervention.

Overview of Azure Database for MySQL, PostgreSQL, and MariaDB

Azure Database for MySQL, PostgreSQL, and MariaDB are fully managed relational database services that provide enterprise-grade capabilities with the flexibility and scalability of the cloud. These services are built on the community editions of the respective database engines, ensuring compatibility with existing applications and tools.

Azure Database for MySQL

Azure Database for MySQL offers a reliable and scalable solution for MySQL users. It provides features such as automated backups, patching, and monitoring, which simplify database management. The service also supports dynamic scaling, allowing you to adjust resources based on your application's needs.

Azure Database for PostgreSQL

Azure Database for PostgreSQL is designed for developers who need a robust and versatile database solution. PostgreSQL is known for its extensibility and support for advanced data types and performance optimization. Azure Database for PostgreSQL leverages these capabilities while adding the benefits of automated management and scaling.

Azure Database for MariaDB

Azure Database for MariaDB provides a fully managed MariaDB service that ensures high availability and performance. MariaDB, known for its robust performance and security features, is a popular choice for web applications and enterprise solutions. Azure Database for MariaDB enhances these features with cloud benefits like automatic backups and scaling.

All three services—MySQL, PostgreSQL, and MariaDB—offer high availability, automated backups, and enterprise-grade security. They integrate seamlessly with other Azure services, enabling you to build comprehensive and scalable solutions.