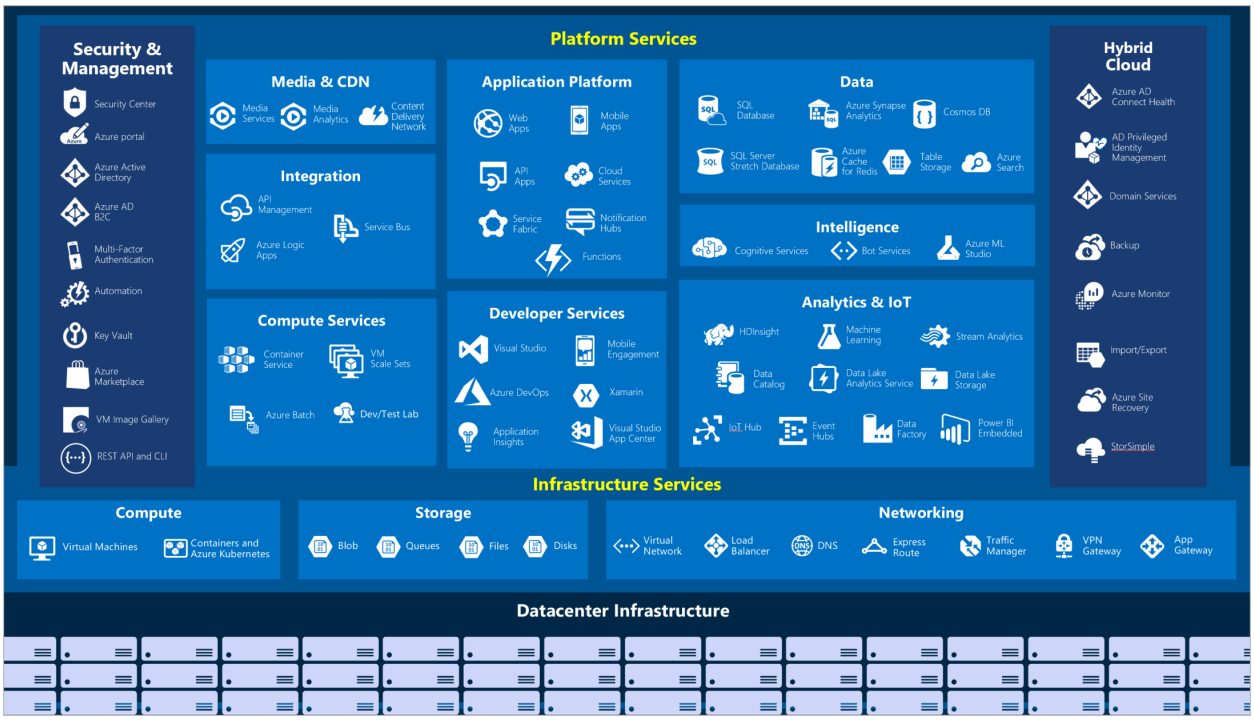
Azure Fundamentals

Notes for Self Study by Neil Bagchi



Describe cloud concepts (25–30%)

Describe cloud computing

• define cloud computing
• describe the shared responsibility model
• define cloud models, including public, private, and hybrid
• identify appropriate use cases for each cloud model
• describe the consumption-based model
• compare cloud pricing models

Describe the benefits of using cloud services

• describe the benefits of high availability and scalability in the cloud
• describe the benefits of reliability and predictability in the cloud
• describe the benefits of security and governance in the cloud
• describe the benefits of manageability in the cloud

Describe cloud service types

• describe infrastructure as a service (IaaS)
• describe platform as a service (PaaS)
• describe software as a service (SaaS)
• identify appropriate use cases for each cloud service (IaaS, PaaS, SaaS)

PART 1: Describe core Azure concepts

What is Cloud Computing?

Cloud computing is the delivery of computing services over the internet by using a pay-as-you-go pricing model. You typically **pay only for the cloud services you use**, i.e a consumption-based model which helps end-users pay for the resources that they use. This has many benefits:

- **Lower operating costs** since there are no upfront costs. Plus, there is no need to purchase and manage the costly infrastructure that users might not use to its fullest, thus, **running infrastructure more efficiently**.
- **Scale as business needs change** since you have the ability to pay for additional resources when they are needed. Also, you have the ability to stop paying for resources that are no longer needed.

Thus, cloud computing is a way to **rent compute power** (how much processing a computer can do) **and storage** from someone else's data center where the Cloud provider maintains the underlying infrastructure.

How does Cloud Computing work?

Azure uses a technology known as **Virtualization** that separates the tight coupling between a computer's hardware and its operating system, using an abstraction layer called a **hypervisor**. The hypervisor emulates all the functions of a real computer and its CPU in a **Virtual Machine**. Multiple Virtual Machines can run at the same time in an optimized capacity and each Virtual Machine can run any compatible operating system, such as Windows or Linux.

So Azure repeats this on a massive scale in Microsoft data centers throughout the world. **Each data center has mini racks filled with servers, and each server includes a hypervisor to run multiple Virtual Machines. A network switch provides connectivity to all of those servers. And one server in each rack runs a special piece of software called a Fabric Controller** that connects to another special piece of software known as the **Orchestrator** which is responsible for managing everything that happens in Azure, including responding to user requests. And users make requests using the Orchestrators Web API. **The Web API can be called by many tools, including the user interface of the Azure portal.**

So, when a user makes a request to create a Virtual Machine, the Orchestrator packages everything that's needed, picks the best server rack, and then sends the package and request to the Fabric Controller. Once the Fabric Controller has created the Virtual Machine, the user can connect to it.

Cloud Service Model

Public Cloud
Services are offered over the public internet and available to anyone who wants to purchase them. Cloud resources, such as servers and storage, are owned and operated by a third-party cloud service provider, and delivered over the internet.

Private Cloud
A private cloud consists of computing resources used exclusively by users from one business or organization. A private cloud can be physically located at your organization's on-site (on-premises) datacenter, or it can be hosted by a third-party service

Hybrid Cloud
A hybrid cloud is a computing environment that combines a public cloud and a private cloud by allowing data and applications to be shared between them.

	provider.	
- No capital expenditures to scale up. - Applications can be quickly provisioned and deprovisioned. - Organizations pay only for what they use.	- Hardware must be purchased for start-up and maintenance. - Organizations have complete control over resources and security. - Organizations are responsible for hardware maintenance and updates.	- Provides the most flexibility. - Organizations determine where to run their applications. - Organizations control security, compliance, or legal requirements.

Benefits of Cloud Computing

- **High availability**: Depending on the service-level agreement (SLA), your cloud-based apps can provide a continuous user experience with no apparent downtime, even when things go wrong.
- **Scalability**: Apps in the cloud can scale ***vertically*** and ***horizontally***:
  - Scale vertically to increase compute capacity by adding RAM or CPUs to a virtual machine.
  - Scaling horizontally increases compute capacity by adding instances of resources, such as adding VMs to the configuration.
- **Elasticity**: Configure cloud-based apps to take advantage of **autoscaling**, i.e. scale up or down as per need.
- **Agility**: Deploy and configure cloud-based resources quickly.
- **Geo-distribution**: Deploy apps and data to regional data centers around the globe, thereby ensuring that customers always have the best performance in their region.
- **Disaster recovery**: By taking advantage of cloud-based backup services, data replication, and geo-distribution, you can deploy your apps with the confidence that comes from knowing that your data is safe in the event of a disaster.

Cloud Service Types

IaaS
A cloud provider keeps the hardware up to date, but operating system maintenance and network configuration are left to the cloud tenant. For example, Azure virtual machines are fully operational virtual compute devices running in Microsoft's datacenters

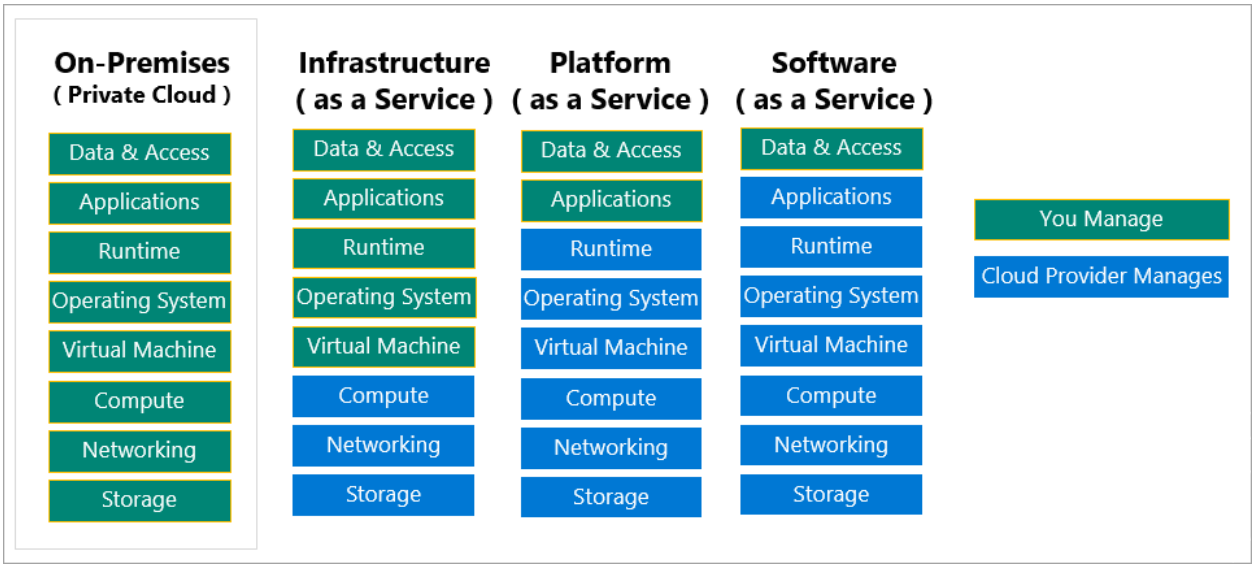
EX: Azure VMs, SQL Server on Azure VM

PaaS
The cloud provider manages the virtual machines and networking resources, and the cloud tenant deploys their applications into the managed hosting environment.

EX: Azure SQL Database, Azure Cosmos Db, Azure Kubernetes Service

SaaS
The cloud provider manages all aspects of the application environment, such as virtual machines, networking resources, data storage, and applications. The cloud tenant only needs to provide their data to the application managed by the cloud provider

EX: Office 365, Azure DevOps Service, Slack



Shared Responsibility Model

Serverless Computing

Like PaaS, with serverless computing, the cloud service provider automatically provisions, scales, and manages the infrastructure required to run the code. Serverless architectures

are highly scalable and event-driven. **They use resources only when a specific function or trigger occurs.**

It's important to note that servers are still running the code. The serverless name comes from the fact that the tasks associated with **infrastructure provisioning and management are invisible to the developer**.

Describe Azure architecture and services (35–40%)

Describe the core architectural components of Azure

• describe Azure regional, regional pairs, and sovereign regions
• describe availability zones
• describe Azure datacenters
• describe Azure resources and resource groups
• describe subscriptions
• describe management groups
• describe the hierarchy of resource groups, subscriptions, and management groups

Describe Azure compute and networking services

• compare compute types, including container instances, virtual machines (VMs), and functions
• describe VM options, including Azure Virtual Machines, Azure Virtual Machine Scale Sets,
availability sets, and Azure Virtual Desktop
• describe resources required for virtual machines
• describe application hosting options, including the Web Apps feature of Azure App Service, containers, and virtual machines
• describe virtual networking, including the purpose of Azure Virtual Networks, Azure virtual subnets, peering, Azure DNS, Azure VPN Gateway, and Azure ExpressRoute
• define public and private endpoints

Describe Azure storage services

• compare Azure storage services
• describe storage tiers
• describe redundancy options
• describe storage account options and storage types
• identify options for moving files, including AzCopy, Azure Storage Explorer, and Azure

File Sync
• describe migration options, including Azure Migrate and Azure Data Box

Describe Azure identity, access, and security
• describe directory services in Azure, including Azure Active Directory (Azure AD) and Azure Active Directory Domain Services (Azure AD DS)
• describe authentication methods in Azure, including single sign-on (SSO), multifactor authentication, and passwordless
• describe external identities and guest access in Azure
• describe Azure AD Conditional Access
• describe Azure role-based access control (RBAC)
• describe the concept of Zero Trust
• describe the purpose of the defense in depth model
• Describe the purpose of Microsoft Defender for Cloud

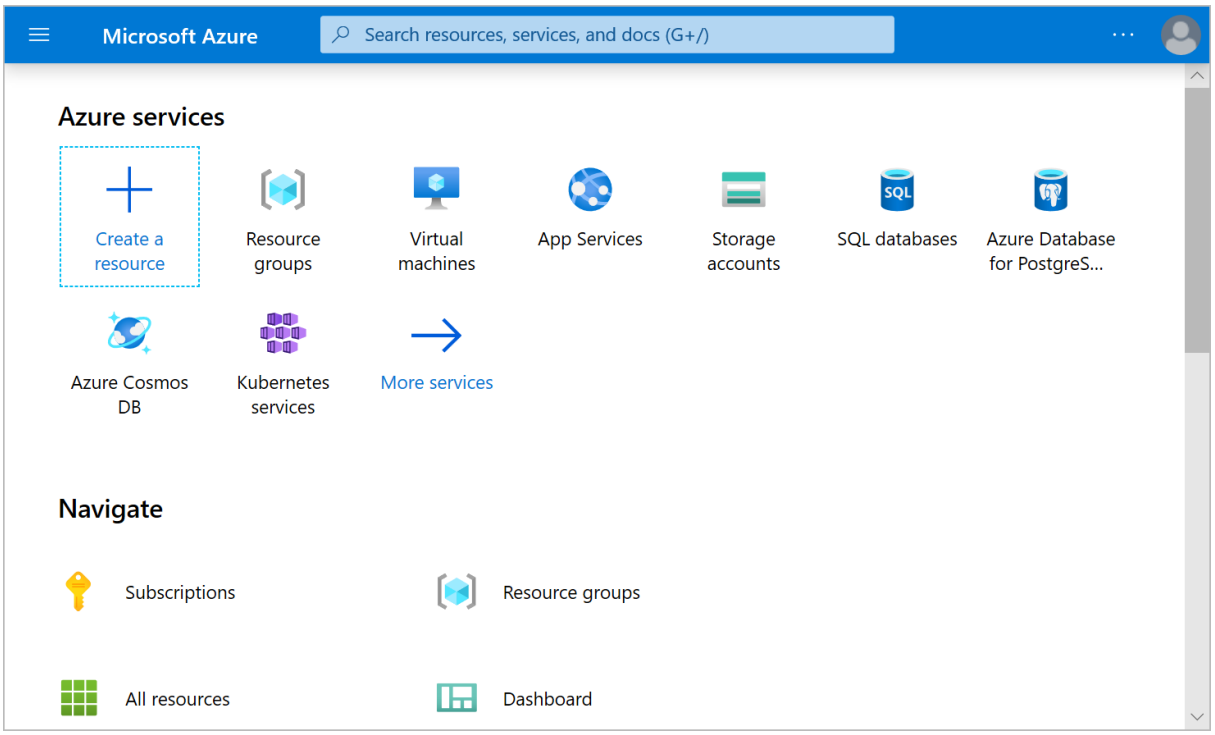
Azure Portal

> ***The Azure portal is a web-based, unified console that provides an alternative to command-line tools. With the Azure portal, you can manage your Azure subscription by using a graphical user interface.***

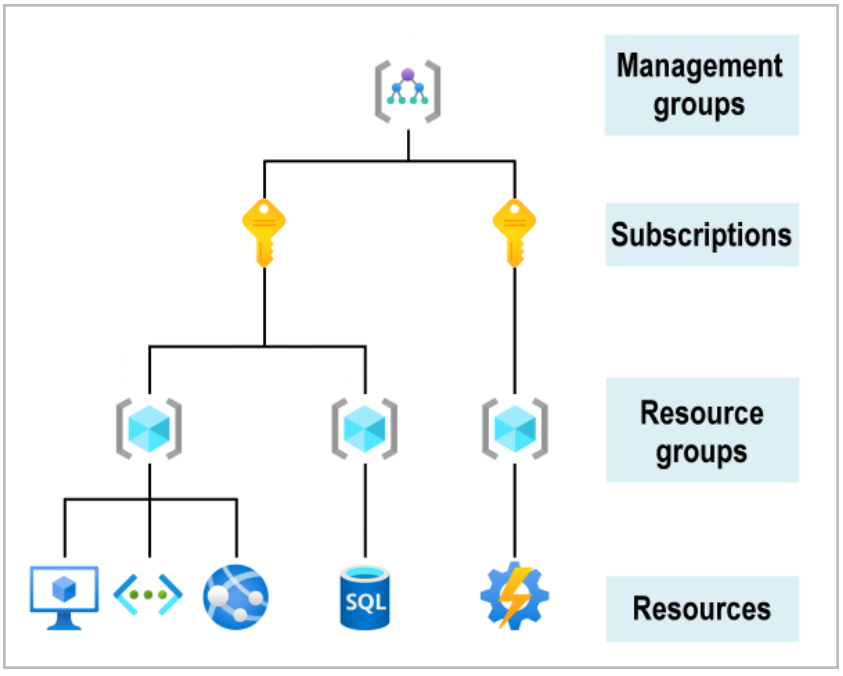
You can:

- Build, manage, and monitor everything from simple web apps to complex cloud deployments.
- Create custom dashboards for an organized view of resources.
- Configure accessibility options for an optimal experience.

The Azure portal is designed for resiliency and continuous availability. It maintains a presence in every Azure data center. This configuration makes the Azure portal resilient to individual data center failures and avoids network slowdowns by being close to users. The Azure portal updates continuously and requires no downtime for maintenance activities.



Organizing structure for Resources



Resources:

> ***A manageable item that's available through Azure. Virtual machines (VMs), storage accounts, web apps, databases, and virtual networks are examples of resources.***

Resource groups:

> ***A resource group is a logical container holding related resources deployed on Azure that you want to manage as a group. These resources are anything you create in an Azure subscription-like VMs, Azure Application Gateway instances, and Azure Cosmos DB instances***

Resource groups can't be nested. If you delete a resource group, all resources contained within it are also deleted. Organizing resources by **life cycle** can be useful in nonproduction environments, where you might try an experiment and then dispose of it. Resource groups also provide a scope for applying role-based access control (RBAC) **permissions**.

By placing resources of similar usage, type, or location in a resource group, you can provide order and organization to resources you create in Azure.

Subscriptions:

> ***An Azure subscription is a logical unit of Azure services that links to an Azure account, which is an identity in Azure Active Directory (Azure AD) or in a directory that Azure AD trusts. An Azure subscription is an object that represents a container that you can put resources in. Subscriptions are tied to tenants, so one tenant can have many subscriptions, but not vice versa.***

A subscription groups together user accounts and the resources that have been created by those user accounts. For each subscription, there are limits or quotas on the amount of resources that you can create and use. Organizations can use subscriptions to manage costs and the resources that are created by users, teams, or projects.

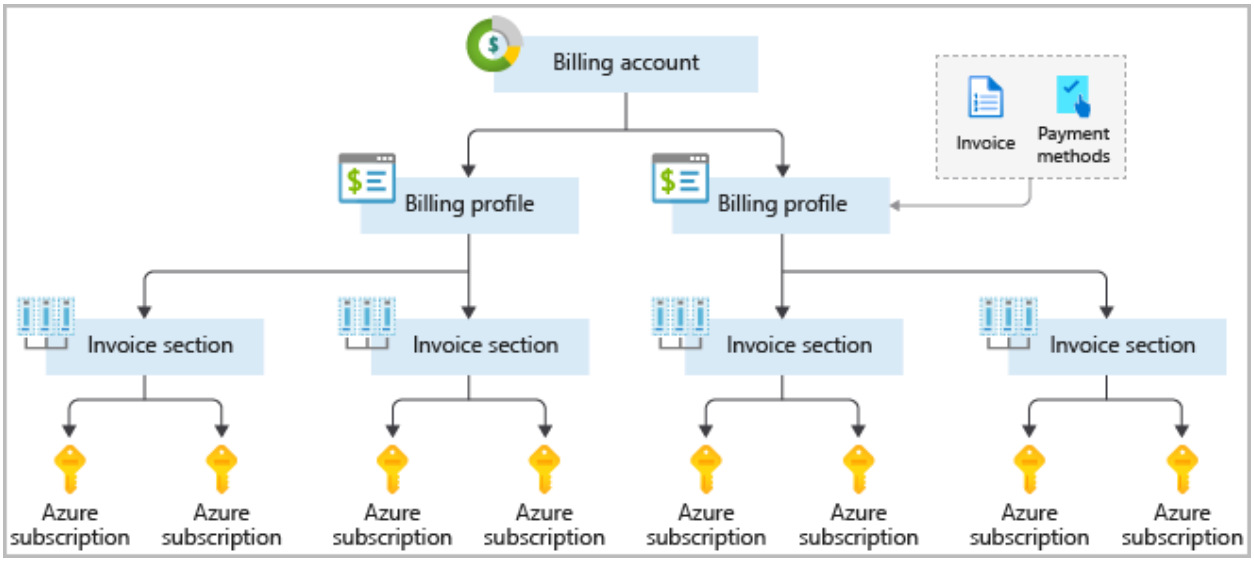
There are two types of subscription boundaries that you can use:

1. **Billing boundary**

   This subscription type determines how an Azure account is billed for using Azure. You can create multiple subscriptions for different types of billing requirements. Azure generates separate billing reports and invoices for each subscription so that you can organize and manage costs.

2. **Access control boundary**

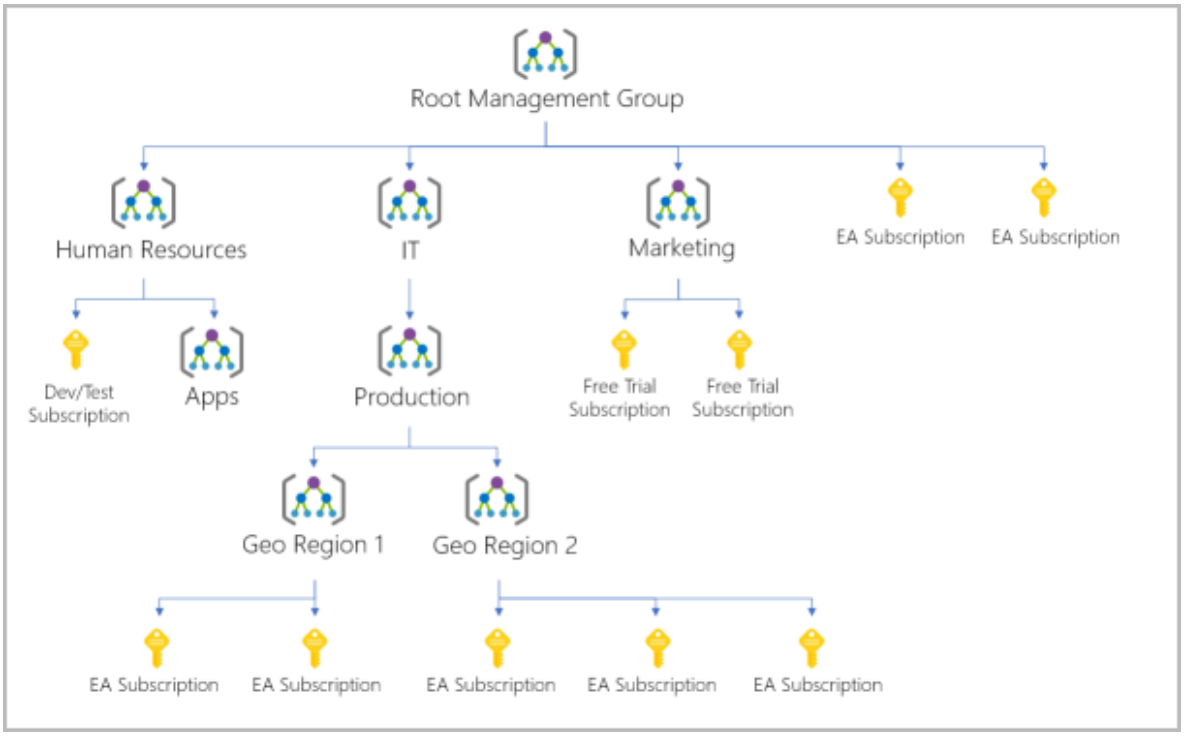
   Azure applies access-management policies at the subscription level, and you can create separate subscriptions to reflect different organizational structures. An example is that within a business, you have different departments to which you apply distinct Azure subscription policies. This billing model allows you to manage and control access to the resources that users provision with specific subscriptions.



Management groups:

> ***You organize subscriptions into containers called management groups and apply your governance conditions to the management groups thus helping you manage access, policy, and compliance for multiple subscriptions. All subscriptions within a management group automatically inherit the conditions applied to the management group.***

We can also create a hierarchy that applies policy i.e. production subscriptions can have a different policy than other subscriptions. Another scenario can be to provide users access to multiple subscriptions. Role-based access control (RBAC) can be assigned over different subscriptions.



• 10,000 management groups can be supported in a single directory.
• A management group tree can support up to six levels of depth. This limit doesn't include the root level or the subscription level.
• Each management group and subscription can support only one parent.
• Each management group can have many children.
• All subscriptions and management groups are within a single hierarchy in each directory.

Azure Regions, Availability Zones & Region Pairs

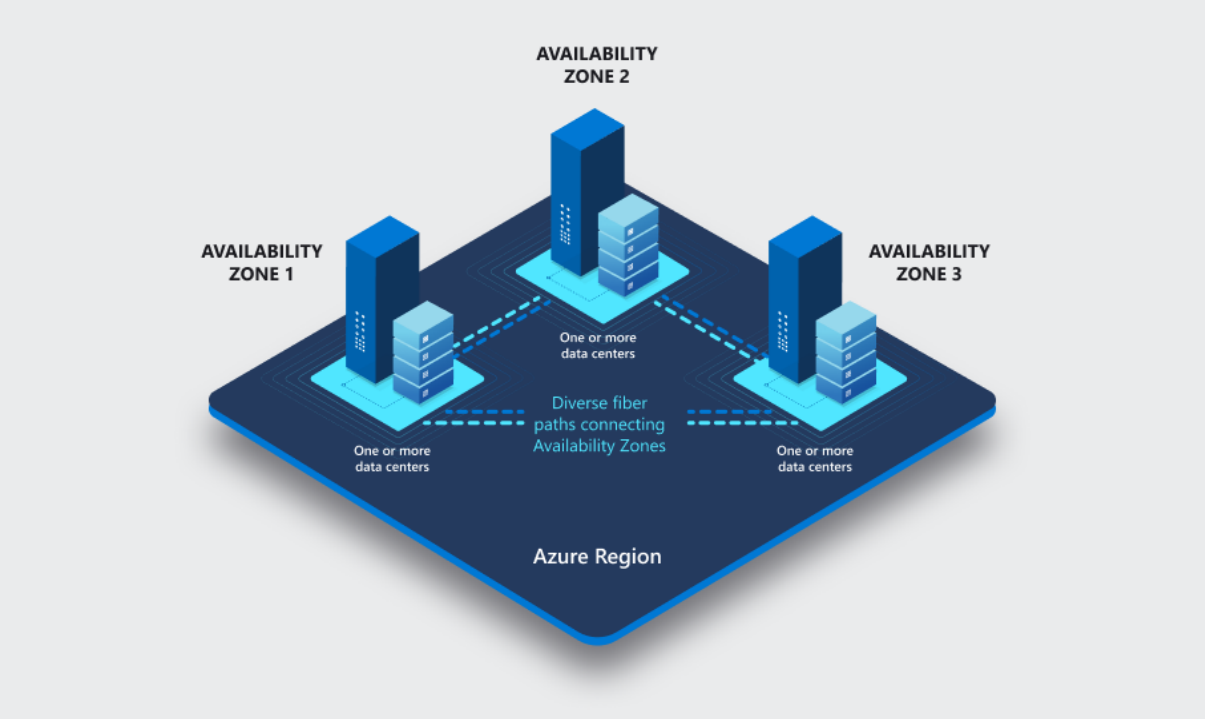
Region

> ***A region is a geographical area on the planet that contains at least one but potentially multiple data centers that are nearby and networked together with a low-latency network.***

Some services or VM features are only available in certain regions, such as specific VM sizes or storage types. There are also some global Azure services that don't require you to select a particular region, such as **Azure Active Directory**, **Azure Traffic Manager**, and **Azure DNS**.

Availability Zones

> ***Availability zones are physically separate datacenters within an Azure region. Each availability zone is made up of one or more data centers equipped with independent power, cooling, and networking. An availability zone is set up to be an isolation boundary. If one zone goes down, the other continues working.***

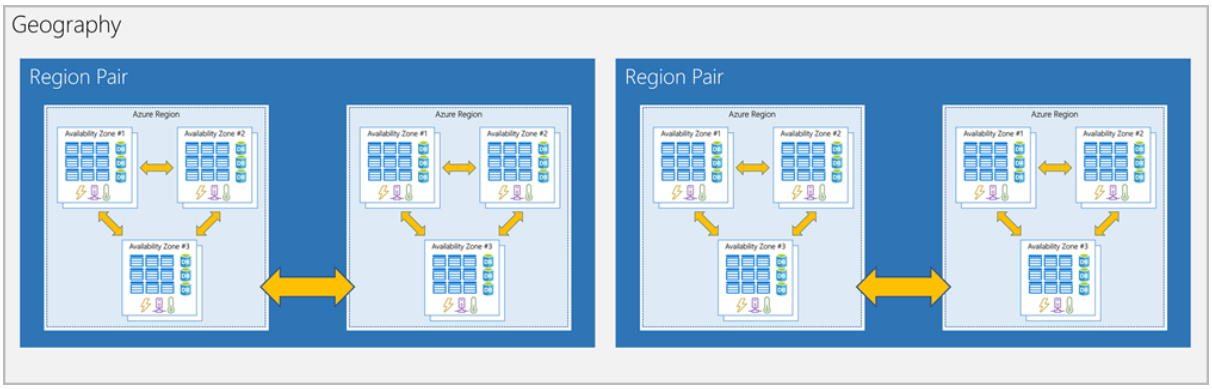


There's a minimum of three availability zones within a single region if applicable. However, not all regions have availability zones.

Region Pair

> ***Each Azure region is always paired with another region within the same geography (such as the US, Europe, or Asia) at least***

> ***300 miles away. This approach allows for the replication of resources across geography which helps reduce the likelihood of interruptions.***



Because 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 by using region pairs.

Special Azure regions

Azure has specialized regions that you might want to use when you build out your applications for compliance or legal purposes. A few examples include:

- **US DoD Central, US Gov Virginia, US Gov Iowa and more:** These regions are physical and logical network-isolated instances of Azure for U.S. government agencies and partners. These datacenters are operated by screened U.S. personnel 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 doesn't directly maintain the datacenters.

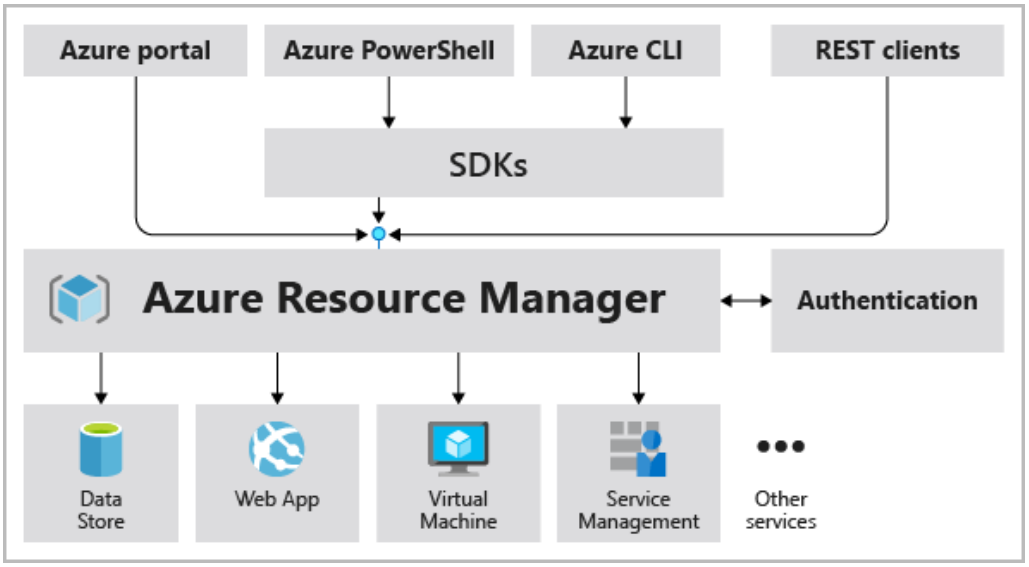
Azure Resource Manager (ARM)

> ***Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure***

> ***account. You use management features like access control, locks, and tags to secure and organize your resources after deployment.***

When a user sends a request from any of the Azure tools, APIs, or SDKs, the Resource Manager receives the request. It authenticates and authorizes the request. Resource Manager sends the request to the Azure service, which takes the requested action.

All capabilities that are available in the Azure portal are also available through PowerShell, the Azure CLI, REST APIs, and client SDKs.



With Resource Manager, you can:

- Manage your infrastructure through declarative templates (JSON) rather than scripts.
- Deploy, manage, and monitor all the resources as a group,
- Redeploy your solution throughout the development life cycle
- Define the dependencies between resources so they're deployed in the correct order.
- Apply access control to all services because RBAC is natively integrated into the management platform.

PART 2: Describe core Azure services

⭐DATABASE SERVICES

▼ Database Fundamentals (Optional)

> ***▶️ A database is used to define a central system in which data can be stored and queried.***

Availability: Percentage of time application provides the expected operation
Durability: Amount of time the data will be available for e.g. 10 years, 100 years.
Both of these are technical measures

How do we measure how quickly we can recover from database failure?
RPO (Recovery Point Objective): Maximum acceptable period of data loss
RTO (Recovery Time Objective): Maximum acceptable downtime
Achieving minimum RTO and RPO is expensive thus we go for trade-off based on the criticality of the data.

Failover Examples

Scenario	Solution
Very small data loss (RPO - 1 minute) **Very small downtime (RTO - 5 minutes)**	**Hot standby** - Automatically synchronize data Have a standby ready to pick up load Use automatic failover from master to standby
Very small data loss (RPO - 1 minute) **BUT I can tolerate some downtimes (RTO - 15 minutes)**	**Warm standby** - Automatically synchronize data Have a standby with minimum infrastructure Scale it up when a failure happens
Data is critical (RPO - 1 minute) but I can tolerate downtime of a few hours (RTO - few hours)	Create regular data **snapshots and transaction logs** Create database from snapshots and transactions logs when a failure happens
Data can be lost without a problem (for example: cached data)	Failover to a completely new server

Relational Database

Relational databases are commonly used to store and query structured data. The data is stored in tables that represent entities. Each instance of an entity is assigned a *primary key* that uniquely identifies it, and these keys are used to reference the entity instance in other tables. The use of keys to reference data entities enables a relational database to be *normalized*; which in part means the elimination of duplicate data values. The tables are managed and queried using Structured Query Language (SQL)

Non-Relational Database

Non-relational databases are data management systems that don't apply a relational schema to the data. Non-relational databases are often referred to as NoSQL (not only SQL) databases, even though some support a variant of the SQL language.

There are four common types of Non-relational databases commonly used.

- **Key-value databases** in which each record consists of a unique key and an associated value, which can be in any format.

Products	
Key	**Value**
123	"Hammer ($2.99)"
162	"Screwdriver ($3.49)"
201	"Wrench ($4.25)"

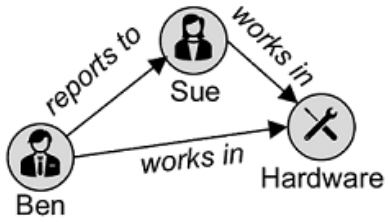
- **Document databases** are a specific form of key-value database in which the value is a JSON document (which the system is optimized to parse and query)

Customers	
Key	**Document**
1	{ "name": "Joe Jones", "email": "joe@litware.com" }
2	{ "name": "Samir Nadoy", "email": "Samir@northwind.com" }

- **Column family databases**, store tabular data comprising rows and columns, but you can divide the columns into groups known as column families. Each column family holds a set of columns that are logically related together.

Orders				
Key	**Customer**		**Product**	
	Name	**Address**	**Name**	**Price**
1000	Joe Jones	1 Main St.	Hammer	2.99
1001	Samir Nadoy	123 Elm Pl.	Wrench	4.25

- **Graph databases** store entities as nodes with links to define relationships between them.



Transactional Data Processing

> ***A transactional system records transactions that encapsulate specific events that the organization wants to track. Transactional systems are often high-volume, sometimes handling many millions of transactions in a single day. The data being processed has to be accessible very quickly. The work performed by transactional systems is often referred to as Online Transactional Processing (OLTP).***

OLTP solutions rely on a database system in which data storage is optimized for both reading and write operations in order to support transactional workloads in which data records are **created, retrieved, updated, and deleted** (often referred to as *CRUD* operations). These operations are applied transactionally, in a way that ensures the integrity of the data stored in the database. To accomplish this, OLTP systems enforce transactions that support so-called ACID semantics:

- **Atomicity** – each transaction is treated as a single unit, which succeeds completely or fails completely. For example, a transaction that involved debiting funds from one account and crediting the same amount to another account must complete both actions. If either action can't be completed, then the other action must fail.
- **Consistency** – transactions can only take the data in the database from one valid state to another. To continue the debit and credit example above, the completed state of the transaction must reflect the transfer of funds from one account to the other.

- **Isolation** – concurrent transactions cannot interfere with one another and must result in a consistent database state. For example, while the transaction to transfer funds from one account to another is in process, another transaction that checks the balance of these accounts must return consistent results - the balance-checking transaction can't retrieve a value for one account that reflects the balance *before* the transfer, and a value for the other account that reflects the balance *after* the transfer.
- **Durability** – when a transaction has been committed, it will remain committed. After the account transfer transaction has been completed, the revised account balances are persisted so that even if the database system were to be switched off, the committed transaction would be reflected when it is switched on again.

Analytical Data Processing

Analytical data processing typically uses read-only (or read-*mostly*) systems that store vast volumes of historical data or business metrics. Analytics can be based on a snapshot of the data at a given point in time or a series of snapshots.

1. Data files may be stored in a central data lake for analysis.
2. An extract, transform, and load (ETL) process copies data from files and OLTP databases into a data warehouse that is optimized for reading activity. Commonly, a data warehouse schema is based on *fact* tables that contain numeric values you want to analyze (for example, sales amounts), with related *dimension* tables that represent the entities by which you want to measure them (for example, customer or product),
3. Data in the data warehouse may be aggregated and loaded into an online analytical processing (OLAP) model, or *cube*. Aggregated numeric values (*measures*) from fact tables are calculated for intersections of *dimensions* from dimension tables. For example, sales revenue might be totaled by date, customer, and product.
4. The data in the data lake, data warehouse, and analytical model can be queried to produce reports, visualizations, and dashboards.

Data lakes are common in modern data analytical processing scenarios, where a large volume of file-based data must be collected and analyzed.

Data warehouses are an established way to store data in a relational schema that is optimized for reading operations – primarily queries to support reporting and data visualization. The data warehouse schema may require some denormalization of data in an OLTP data source (introducing some duplication to make queries perform faster).

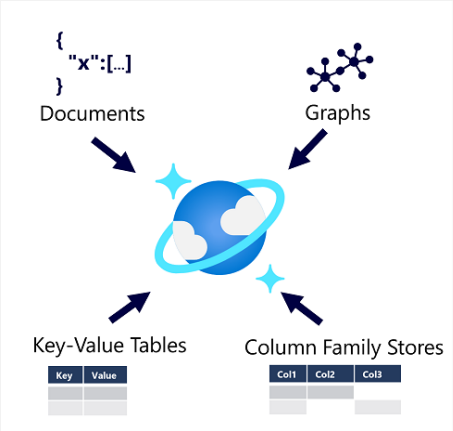
An OLAP model is an aggregated type of data storage that is optimized for analytical workloads. Data aggregations are across dimensions at different levels, enabling you to *drill up/down* to view aggregations at multiple hierarchical levels

Azure Cosmos DB (PaaS - NoSQL)

> ***Globally distributed multi-model database service that supports NoSQL options. At the lowest level, Azure Cosmos DB stores data in atom-record-sequence (ARS) format. The data is then abstracted and projected as an API, which you specify when you're creating your database (choices include SQL, MongoDB, Cassandra, Tables, and Gremlin). Cosmos DB uses indexes and partitioning to provide fast read and write performance and can scale to massive volumes of data.***

▼ Details (Optional)

Azure Cosmos DB supports multiple application programming interfaces (APIs) that enable developers to use the programming semantics of many common kinds of data store to work with data in a Cosmos DB database. The internal data structure is abstracted, enabling developers to use Cosmos DB to store and query data using APIs with which they're already familiar. Cosmos DB uses indexes and partitioning to provide fast read and write performance and can scale to massive volumes of data.



Cosmos DB is a highly scalable database management system. Cosmos DB automatically allocates space in a container for your partitions, and each partition can grow up to 10 GB in size. Indexes are created and maintained automatically. There's virtually no administrative overhead.

When you provision a new Cosmos DB instance, you select the API that you want to use. The choice of API depends on many factors including, the type of data to be stored, the need to support existing applications, and the API skills of the developers who will work with the data store.

Core (SQL) API

The native API in Cosmos DB manages data in JSON document format, and despite being a NoSQL data storage solution, uses SQL syntax to work with the data.

```
Input (SQL)
SELECT *
FROM customers c
WHERE c.id = "joe@litware.com"

Output (JSON)
{
   "id": "joe@litware.com",
   "name": "Joe Jones",
   "address": {
        "street": "1 Main St.",
        "city": "Seattle"
    }
}
```

MongoDB API

MongoDB is a popular open-source database in which data is stored in Binary JSON (BSON) format. MongoDB Query Language (MQL) uses a compact, object-oriented syntax in which developers use ***objects* to call *methods***. For example, the following query uses the **find** method to query the **products** collection in the **db** object:

```
Input (Javascript)
db.products.find({id: 123})

Output (BSON)
{
   "id": 123,
   "name": "Hammer",
   "price": 2.99}
}
```

Table API

The Table API is used to work with data in key-value tables, similar to Azure Table Storage. The Azure Cosmos DB Table API offers greater scalability and performance than Azure Table Storage.

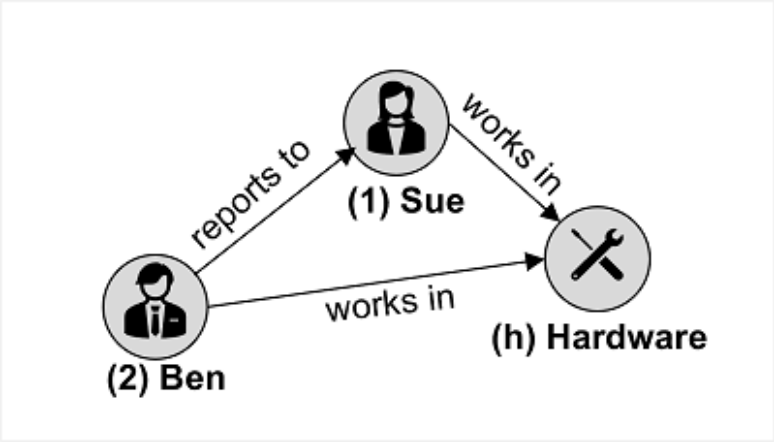
```
https://endpoint/Customers(PartitionKey='1',RowKey='124')
```

Cassandra API

The Cassandra API is compatible with Apache Cassandra, which is a popular open-source database that uses a column-family storage structure. Column families are tables, similar to those in a relational database, with the exception that it's not mandatory for every row to have the same columns. Cassandra supports a syntax based on SQL.

Gremlin API

The Gremlin API is used with data in a graph structure; in which entities are defined as *vertices* that form nodes in a connected graph. Nodes are connected by *edges* that represent relationships, like this:



The example in the image shows two kinds of vertex (employee and department) and edges that connect them (employee "Ben" reports to employee "Sue", and both employees work in the "Hardware" department).

Gremlin syntax includes functions to operate on vertices and edges, enabling you to insert, update, delete, and query data in the graph. For example, you could use the following code to add a new employee named *Alice* who reports to the employee with ID **1** (*Sue*)

```
g.addV('employee').property('id', '3').property('firstName', 'Alice')
g.V('3').addE('reports to').to(g.V('1'))
```

Azure SQL Database (PaaS)

> ***Fully managed relational database with auto-scale, integral intelligence, and robust security. It enables to process both relational data and non-relational structures, such as graphs, JSON, spatial, and XML.***

You can use advanced query processing features, such as high-performance, in-memory technologies and intelligent query processing. The newest capabilities of SQL Server are released first to SQL Database.

Azure SQL Database is available as a *Single Database* or an *Elastic Pool*. Azure SQL Database gives you the best option for low cost with minimal administration. It **isn't** fully compatible with on-premises SQL Server installations.

SQL Managed Instance (PaaS - Preferred to SQL Database)

Azure SQL Database and Azure SQL Managed Instance offer many of the same features; however, Azure SQL Managed Instance provides several options that might not be available to Azure SQL Database such as linked servers, Service Broker (a message processing system that can be used to distribute work across servers), or Database Mail (which enables your database to send email messages to users), Cyrillic characters for collation.

SQL Server on Azure Virtual Machines (IaaS)

Service that hosts enterprise SQL Server apps in the cloud which is fully compatible with on-premises installations.

	SQL Server on Azure VMs	Azure SQL Managed Instance	Azure SQL Database
			
Type of cloud service	IaaS	PaaS	PaaS
SQL Server compatibility	Fully compatible with on-premises physical and virtualized installations. Applications and databases can easily be "lift and shift" migrated without change.	Near-100% compatibility with SQL Server. Most on-premises databases can be migrated with minimal code changes by using the Azure Database Migration service	Supports most core database-level capabilities of SQL Server. Some features depended on by an on-premises application may not be available.
Architecture	SQL Server instances are installed in a virtual machine. Each instance can support multiple databases.	Each managed instance can support multiple databases. Additionally, *instance pools* can be used to share resources efficiently across smaller instances.	You can provision a *single database* in a dedicated, managed (logical) server; or you can use an *elastic pool* to share resources across multiple databases and take advantage of on-demand scalability.
Availability	99.99%	99.99%	99.995%
Management	You must manage all aspects of the server, including operating system and SQL Server updates, configuration, backups, and other maintenance tasks.	Fully automated updates, backups, and recovery.	Fully automated updates, backups, and recovery.
Use cases	Use this option when you need to migrate or extend an on-premises SQL Server solution and retain full control over all aspects of server and database configuration.	Use this option for most cloud migration scenarios, particularly when you need minimal changes to existing applications.	Use this option for new cloud solutions, or to migrate applications that have minimal instance-level dependencies.

Azure Database Migration Assistant Service (DMA)

Service that migrates databases to the cloud with no application code changes. This tool analyzes your existing on-premise databases on SQL Server and reports any issues that could block migration to a **managed instance**.

Azure Database for MySQL

> ***Fully managed and scalable MySQL relational database with high availability and security.***

Azure Database for MySQL is a relational database service in the cloud, and it's based on the MySQL Community Edition database engine. Azure Database for MySQL offers

several service tiers, and each tier provides different performance and capabilities to support lightweight to heavyweight database workloads.

Azure Database for PostgreSQL

> ***Fully managed and scalable PostgreSQL relational database with high availability and security.***

Azure Database for PostgreSQL is a relational database service in the cloud. The server software is based on the community version of the open-source PostgreSQL database engine

Single Server (Vertical Scaling)
The Single Server deployment option offers three pricing tiers:
Basic, General Purpose, and Memory Optimized. Each tier offers different resource capabilities to support database workloads.

Hyperscale (Horizontal Scaling)
The Hyperscale (Citus) option horizontally scales queries across multiple machines by using sharding. Its query engine parallelizes incoming SQL queries across the servers for faster responses on large datasets. It serves applications that require greater scale and performance, generally, workloads that are approaching, or already exceed, 100 GB of data.
The Hyperscale (Citus) deployment option supports multi-tenant applications, real-time operational analytics, and high throughput transactional workloads.

Azure Database for MariaDB

> ***Fully managed and scalable MariaDB relational database with high availability and security***

Azure Cache for Redis (PaaS)

Retrieving data from memory is much faster than retrieving data from disk thus **In-memory databases** like Redis deliver microsecond latency by storing persistent data in memory.

Use cases : Caching, session management, gaming leader boards, geospatial applications

⭐BIG DATA SERVICES (OLAP - Columnar Storage)

Azure Synapse Analytics

> ***Run analytics at a massive scale by using a cloud-based enterprise data warehouse that takes advantage of massively parallel processing (MPP) to run complex queries quickly across petabytes of data.***

Azure Synapse Analytics (formerly Azure SQL Data Warehouse) is a limitless analytics service that brings together enterprise data warehousing and big data analytics. You can query data on your terms by using either serverless or provisioned resources at scale. You have a unified experience to ingest, prepare, manage, and serve data for immediate BI and machine learning needs.

Azure HDInsight

> ***Process massive amounts of data with managed clusters of Hadoop clusters in the cloud.***

Azure HDInsight is a fully managed, open-source analytics service for enterprises that makes it easier, faster, and more cost-effective to process massive amounts of data. You can run popular open-source frameworks and create cluster types such as Apache Spark , Apache Hadoop , Apache Kafka , Apache HBase , Apache Storm , and Machine Learning Services.

Apache Hadoop: a distributed system that uses *MapReduce* jobs to process large volumes of data efficiently across multiple cluster nodes. MapReduce jobs can be written in Java or abstracted by interfaces such as Apache Hive - a SQL-based API that runs on Hadoop

Azure Databricks

> ***Azure-integrated version of the popular Databricks platform, which combines the Apache Spark data processing platform with SQL database semantics and an integrated management interface to enable large-scale data analytics. Data engineers can use existing Databricks and Spark skills to create analytical data stores in Azure Databricks.***

Azure Databricks supports Python, Scala, R, Java, and SQL, as well as data science frameworks and libraries including TensorFlow, PyTorch, and scikit-learn.

Azure Data Lake Analytics

> ***Azure Data Lake Analytics is an on-demand analytics job service that simplifies big data. Instead of deploying, configuring, and tuning hardware, you write queries to transform your data and extract valuable insights.***

⭐COMPUTE SERVICES

Azure Virtual Machines (IaaS)

> ***Virtual machines are virtual servers (software emulations of physical computers) that include a virtual processor, memory, storage, and networking resources.***

Features

- **Load Balancing** and **Auto Scaling** for multiple VM instances
- **Attach Storage** to VM instances
- Manage **network connectivity** and **configuration** for VM instances

You can create and provision a VM in minutes when you **select a preconfigured VM image.** An image is a template used to create a VM. These templates already include an OS and often other software, like development tools or web hosting environments e.g. Ubuntu Server, Windows Server, Oracle Linux, Red Hat Linux, etc.

VMs host an operating system where you can install and run software just like a physical computer. **You still need to configure, update, and maintain the software that runs on the VM.** You can run single VMs for testing, development, or minor tasks. Or you can group VMs together to provide high availability, scalability, and redundancy

▼ (Optional) Creating a VM

We have selected the following configuration:
Image: Ubuntu
Region: East US
Size: Standard B1s
Authentication Type: SSH Public Key
Select inbound ports: HTML(80), SSH(22)

Download the SSH key pair which we will use to connect to the VM

Once the VM is deployed, connect via SSH with clients such as PuTTY, PowerShell, Bash etc

```
chmod 400 my-first-vm_key.pem #Name of the key-pair file-path
ssh -i YOUR_KEY_PATH azureuser@PUBLIC_IP_ADDRESS #my-first-vm_key.pem in key_path
ssh -i my-first-vm_key.pem azureuser@13.90.35.80

sudo su
apt-get -y update
apt-get -y install nginx

echo "Hello World"
echo "Hello World" > /var/www/html/index.html
echo "Hello World from in28minutes" > /var/www/html/index.html

hostname
echo "$(hostname)"
echo "Hello World from $(hostname)"
echo "Hello World from $(hostname)" > /var/www/html/index.html
```

If we want to automate certain pre-defined steps when a new VM is being deployed, we can add those commands in the custom data box in the Advanced tab while creating

the VM.

Availability

Availability refers to the percentage of time an application performs the expected operations
99.95% = 22 mins downtime in a month and 99.99% =4.5 mins downtime (most app aim for).

‼**Options to increase Availability**

1. For Single Instance VM, **upgrading the Disk** used:
    a. Premium OR Ultra SSD Disk: 99.9%
    b. Standard SSD: 99.5%
    c. Standard HDD: 95%
2. Create two or more instances in the same **Availability Set**: 99.95%

> ***Availability Set is a logical grouping of VMs which are distributed across multiple faults and update domains (user choice)***

    - **Fault Domains**: Group of VMs sharing a common power source and network switch
    - **Update Domains**: Group of VMs that are rebooted (updated) at the same time
3. Two or more instances in two or more **Availability Zones** in the same Azure region: 99.99%

Azure Virtual Machine Scale Sets

> ***Scale Sets are used to deploy and manage a set of identical, load-balanced VMs. With all VMs configured the same, virtual machine scale sets are designed to support true autoscale. No pre-provisioning of VMs is required. For this reason, it's easier to build large-scale services targeting big compute, big data, and***

> ***containerized workloads. The number of VM instances automatically increases or decreases in response to demand or a defined schedule.***

‼When creating a scale set, remember to change the network interface option to allow public IP address and enable the use of a load balancer. We can mention how we want to configure our auto-scaling options here e.g. we can scale up to create 1 new instance if our CPU load threshold crosses 75%.

Features and Explanation
Static IP Address: Assign a fixed IP address to VM. Public IP addresses are charged per IP/hour
Azure Monitoring: Monitoring for your Azure VMs
Dedicated Hosts: Physical servers dedicated to one customer
Create cheaper, temporary instances for non-critical workloads: Azure Spot instances
Reserve compute instances ahead of time: Reserved VM Instances (1 or 3 years)

Azure App Service (PaaS)

> ***Fully managed platform for building, deploying, and scaling enterprise-grade web, mobile, and REST API apps in the programming language of our choice running on any platform (Windows/Linux) using GitHub, Azure DevOps or any Git repo. It offers automatic scaling, built in load balancing and high availability.***

You pay for the Azure compute resources your app uses **while it processes requests** based on the App Service plan you choose. The App Service plan determines how much hardware is devoted to your host. We can also manually scale up or scale out based on requirements.

With App Service, you can host most common app service styles like:

- **Web apps** (support for hosting web apps by using ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python)

- **API apps** (build REST-based web APIs by using your choice of language and framework)
- **WebJobs** (run a program (.exe, Java, PHP, Python, or Node.js) or script (.cmd, .bat, PowerShell, or Bash) in the same context as a web app, API app, or mobile app. They can be scheduled or run by a trigger.)
- **Mobile apps** (build a back end for iOS and Android apps)

Azure Container Instance & Azure Kubernetes Service (PaaS)

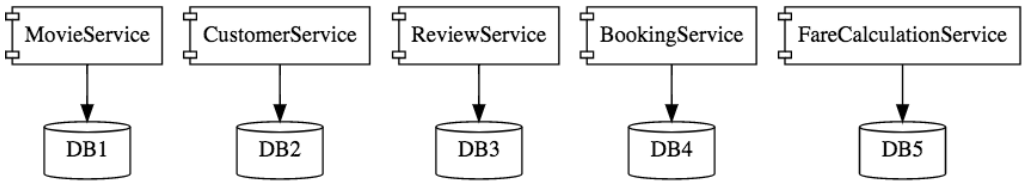
Microservices

> ***▶️ A microservice is just a web service that is a small, well-defined scope and is loosely coupled from any other web service.***

Instead of maintaining a single huge application or software with interdependencies and restrictions, small focused microservices is gaining more popularity. This architecture is where you break solutions into smaller, independent pieces. For example, you might split a website into a container hosting your front end, another hosting your back end, and a third for storage. This split allows you to separate portions of your app into logical sections that can be maintained, scaled, or updated independently.

A team can update an existing service without rebuilding and redeploying the entire application. Plus, they can also easily roll back or roll forward and update if something goes wrong. The best part, this makes bug fixes and feature releases more manageable and less risky. The deployment strategy also means that each microservice can be scaled independently. **Microservices can communicate with each other by using well-defined APIs.** The internal implementation details of each service are encapsulated behind their interface.

‼️However, typically you'd want to reduce those interdependencies and try to introduce an orchestration or **management layer** in the higher level consuming application that coordinates calls to various lower level microservices and combines results.



Containers

> ***Containers are lightweight, virtualized application environments. They're designed to be quickly created, scaled out, and stopped dynamically. VMs virtualize hardware wheras containers virtualize OS.***

Containerized apps run on Azure without provisioning servers or VMs. Much like running multiple virtual machines on a single physical host, you can run **multiple containers on a single physical or virtual host**.

Containers are managed through a **container orchestrator**, which can start, stop, and scale out application instances as needed. The task of automating, managing, and interacting with a large number of containers is known as **orchestration.**

There are two ways to manage both Docker and Microsoft-based containers in Azure: **Azure Container Instances** and **Azure Kubernetes Service** (AKS)

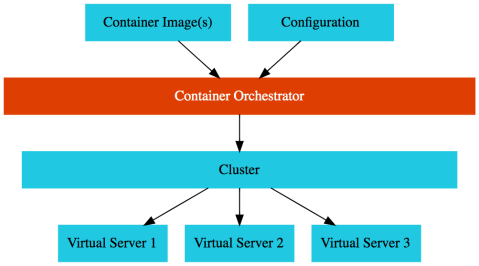
Azure Container Instances (ACI)

This is used to manage and run simple container based application that don’t require provisioning or managing VMs. However, ACI doesn’t provided advanced orchestration features and this is where AKS comes in.

Azure Kubernetes Service (AKS)

Container Orchestration: The task of automating, managing, and interacting with a large number of containers is known as **orchestration.**

Typical Features:
Auto Scaling - Scale containers based on demand
Service Discovery - Help microservices find one another
Load Balancer - Distribute load among multiple instances of a microservice
Self Healing - Do health checks and replace failing instances
Zero Downtime Deployments - Release new versions without downtime



> !!Microsoft provides a container orchestrator solution too which is called **Azure Service Fabric**.

> ***Kubernetes is the most used Cluster management service for VMs that run containerized services. AKS combines container management automation with an extensible API to create a cloud-native application management powerhouse.***

Kubernetes manages the placement of pods, which can consist of one or more containers, on a Kubernetes cluster node. Additionally, if one of these pods crashes, Kubernetes can create a new instance of it. If a cluster node is removed, Kubernetes can move any affected workload to a different node in the cluster. On top of that, Kubernetes pods can be scaled to provide more or less throughput to meet scale demands. And these scale operations can be triggered manually or automatically using Kubernetes horizontal pod auto-scaling. Finally, if an application needs to be updated, Kubernetes can stagger the update deployment to minimize downtime. Plus, if the update is problematic, Kubernetes can roll back to a previous version.

Along with pod management, Kubernetes can also manage container storage and networking. It is also common for an application running in Kubernetes to use cloud-based storage and data systems such as Azure Storage or Azure Cosmos DB for data storage and retrieval. In regard to networking, Kubernetes network plugins provide capabilities such as exposing pods to the internet, load balancing traffic across multiple replicas of a pod, network isolation, and policy-driven network security.

Azure Functions and Logic Apps (Serverless Computing)

> ***An event-driven, serverless compute service. Both of these are commonly used when you need to perform work in response to an event (often via a REST request), timer, or message from another Azure service, and when that work can be completed quickly, within seconds or less.***

Instead of writing an entire application, **the developer authors a function, which contains both code and metadata about its triggers and bindings.** The platform automatically schedules the function to run and scales the number of compute instances based on the rate of incoming events. ***Triggers define how a function is invoked. Bindings provide a declarative way to connect to services from within the code.***

Functions can be either stateless or stateful.

When they're **stateless** (the default), they behave as if they're **restarted every time** they respond to an event.

When they're **stateful** (called Durable Functions), Azure Functions can perform orchestration tasks which allow developers to chain functions together while maintaining state. **Azure Logic Apps was designed with orchestration in mind.**

Azure Functions:

Functions can execute code in almost any modern language (C#, Python, JavaScript, Typescript, Java, and PowerShell). This lets you concisely build complex algorithms, or data lookup and parsing operations. You would be responsible for maintaining the code, handling exceptions resiliently, and so on.

Azure Logic Apps:

Logic apps are designed in a web-based designer and can execute logic triggered by Azure services **without writing any code**. (It's possible to create the same workflow by using Azure Functions, but it might take a considerable amount of time to research which

APIs to call and how to call them. Azure Logic Apps has already componentized these API calls so that you supply only a few details and the details of calling the necessary APIs is abstracted away.)

Logic apps are similar to functions. Both enable you to trigger logic based on an event. **Where functions execute code, logic apps execute workflows that are designed to automate business scenarios and are built from predefined logic blocks**.

Azure Functions is a serverless compute service, and Azure Logic Apps is intended to be a serverless orchestration service. Although you can use Azure Functions to orchestrate a long-running business process that involves various connections, this was not its primary use case when it was designed.

Azure Functions pricing is based on the number of executions and the running time of each execution. Logic Apps pricing is based on the number of executions and the type of connectors that it utilizes.

	Functions	Logic Apps
State	Normally stateless, but Durable Functions provide state.	Stateful.
Development	Code-first (imperative).	Designer-first (declarative).
Connectivity	About a dozen built-in binding types. Write code for custom bindings.	Large collection of connectors. Enterprise Integration Pack for B2B scenarios. Build custom connectors.
Actions	Each activity is an Azure function. Write code for activity functions.	Large collection of ready-made actions.
Monitoring	Azure Application Insights.	Azure portal, Log Analytics.
Management	REST API, Visual Studio.	Azure portal, REST API, PowerShell, Visual Studio.
Execution context	Can run locally or in the cloud.	Runs only in the cloud.

⭐STORAGE SERVICES

Data Redundancy

Option	Redundancy	Discussion
Locally redundant storage (LRS)	Three synchronous copies in same data center	Least expensive and least availability
Zone-redundant storage (ZRS)	Three synchronous copies in three AZs in the primary region	
Geo-redundant storage (GRS)	LRS + Asynchronous copy to secondary region (three more copies using LRS)	
Geo-zone-redundant storage (GZRS)	ZRS + Asynchronous copy to secondary region (three more copies using LRS)	Most expensive and highest availability

Ex: If you use Geo-redundant storage (GRS) and choose region as East US; 3 copies stored in East US and 3 copies in the corresponding paired region - West US

Azure Blob Storage

> ***Blob (binary large object) Storage is an object storage solution for the cloud. It can store massive amounts of data, such as text or binary data***.

Azure Blob Storage is unstructured, meaning that there are no restrictions on the kinds of data it can hold. Blob Storage can manage thousands of simultaneous uploads, massive amounts of video data, constantly growing log files, and can be reached from anywhere with an internet connection.

Azure Blob Storage supports three different types of blob:

- **Block blobs:** A block blob is handled as a set of blocks. Each block can vary in size, **up to 100 MB**. A block blob can contain **up to 50,000 blocks**, giving a
- **Page blobs:** A page blob is organized as a collection of **fixed-size 512-byte page**s. A page blob is **optimized to support random**
- **Append blobs:** An append blob is a **block blob optimized to support append operations**. You can only add blocks to the end of an append blob;

maximum size of over 4.7 TB. **The block is the smallest amount of data that can be read or written as an individual unit**. Block blobs are best used to store discrete, large, binary objects that change **infrequently**.

read and write operations; you can fetch and store data for a single page if necessary. A page blob can hold **up to 8 TB** of data. Azure uses page blobs to implement virtual disk storage for virtual machines.

updating or deleting existing blocks isn't supported. Each block can vary in size, **up to 4 MB**. The maximum size of an append blob is just over **195 GB**.

💡 **Azure Data Lake Storage Gen2**: Enhanced Azure Blob Storage for enterprise big data analytics (exabytes, hierarchical). It provides low-cost, tiered storage, with high availability/disaster recovery.

Blob Access Tiers

The available access tiers include:
• **Hot access tier**: Optimized for storing data that is accessed frequently (for example, images for your website).
• **Cool access tier**: Optimized for data that is infrequently accessed and stored for at least 30 days (for example, invoices for your customers).
• **Archive access tier**: Appropriate for data that is rarely accessed and stored for at least 180 days, with flexible latency requirements (for example, long-term backups

You can create **lifecycle management policies** for blobs in a storage account. A lifecycle management policy can automatically move a blob from **Hot** to **Cool**, and then to the **Archive** tier, as it ages and is used less frequently (***policy is based on the number of days since modification***). A lifecycle management policy can also arrange to delete outdated blobs.

Azure File Storage

> ***File shares that can be accessed and managed like a file server. Azure Files offers fully managed file shares in the cloud that are***

> ***accessible via the industry standard Server Message Block (SMB) and Network File System (NFS) protocols.***

Applications running in Azure virtual machines or cloud services can mount a file storage share to access file data. Any number of Azure virtual machines or roles can mount and access the file storage share simultaneously.

Azure Files enables you to share **up to 100 TB of data in a single storage account**. This data can be distributed across any number of file shares in the account. Azure File Storage supports **up to 2000 concurrent connections per shared file**. Azure File Storage offers two performance tiers. *The Standard tier* uses hard disk-based hardware in a data center, and the *Premium tier* uses solid-state disks

Azure File Sync

> ***Azure File Sync enables centralizing your organization's file shares in Azure Files, while keeping the flexibility, performance, and compatibility of a Windows file server.***

While some users may opt to keep a full copy of their data locally, Azure File Sync additionally has the ability to transform Windows Server into a quick cache of your Azure file share. You can use any protocol that's available on Windows Server to access your data locally, including SMB, NFS, and FTPS.

Azure Table Storage

> ***A NoSQL store that hosts unstructured data independent of any schema. It makes use of tables containing key/value data items. Each item is represented by a row that contains columns for the data fields that need to be stored.***

However, don't be misled into thinking that an Azure Table Storage table is like a table in a relational database. An Azure Table enables you to store semi-structured data. **All rows in a table must have a unique key (composed of a partition key and a row key**), and when you modify data in a table, a *timestamp* column records the date and time the modification was made; but other than that, the columns in each row can vary.

> 💡 **Azure Table Storage tables have no concept of foreign keys, relationships, stored procedures, views, or other objects you might find in a relational database.** Data in Azure Table storage is usually **denormalized**, with each row holding the entire data for a logical entity.

Azure Queue Storage

A data store for queuing and reliably delivering messages between applications

Azure Disk Storage

> ***Disk Storage provides disks for Azure virtual machines. Applications and other services can access and use these disks as needed, similar to how they would in on-premises scenarios.***

Typically, ONE storage device is connected with ONE virtual server but multiple storage can be connected to ONE VM as well. You can use standard SSD and HDD disks for less critical workloads, premium SSD disks for mission-critical production applications, and ultra disks for data-intensive workloads such as SAP HANA, top tier databases, and transaction-heavy workloads

⭐NETWORKING SERVICES (Revise from Learn)

Azure Virtual Networks

> ***Azure virtual networks enable Azure resources, such as VMs, web apps, and databases, to communicate with each other, with users on the internet, and with your on-premises client computers. You control all the traffic coming in and going outside a Virtual Network.***

💡 **(Best Practice)** Create all your Azure resources (compute, storage, databases etc) within a Virtual Network

Subnets:

Different resources are created on cloud and each type of resource has its **own access needs.**
Although **Load Balancers** are used to communicate to private resources on the cloud, but they are accessible from internet (**public** resources). However, we don't want databases or VM instances to be accessible from internet. ONLY applications within your virtual network should be able to access
them (private resources). To solve this, we create different subnets for public and private resources.

💡 Resources in a public subnet CAN be accessed from internet
Resources in a private subnet CANNOT be accessed from internet
BUT resources in public subnet can talk to resources in private subnet

Azure virtual networks provide the following key networking capabilities:

- **Isolation and segmentation**

  Azure virtual network allows you to create multiple isolated virtual networks. When you set up a virtual network, you define a private IP address space by using either public or private IP address ranges. The public IP range only exists within the virtual network and isn't internet routable. You can divide that IP address space into subnets and allocate part of the defined address space to each named subnet.

  💡 You can use the **name resolution service** that's built in to Azure to configure the virtual network to use an internal or an external DNS server.

- **Internet communications**

  A VM in Azure can connect to the internet by default. You can enable incoming connections from the internet by assigning a public IP address to the VM or by putting the VM behind a public load balancer.

- **Communicate between Azure resources**
  - **Virtual networks** Virtual networks can connect not only VMs but other Azure resources, such as the App Service Environment for Power Apps, Azure Kubernetes Service, and Azure virtual machine scale sets.
  - **Service endpoints** You can use service endpoints to connect to other Azure resource types, such as Azure SQL databases and storage accounts. This approach enables you to link multiple Azure resources to virtual networks to improve security and provide optimal routing between resources.
- **Communicate with on-premises resources**

  Azure virtual networks enable you to link resources together in your on-premises environment and within your Azure subscription. In effect, you can create a network that spans both your local and cloud environments. There are three mechanisms for you to achieve this connectivity:

  - **Point-to-site virtual private networks**
    The typical approach to a virtual private network (VPN) connection is **from a computer outside your organization, back into your corporate network**. In this case, the client computer initiates an encrypted VPN connection to connect that computer to the Azure virtual network.
  - **Site-to-site virtual private networks**
    A site-to-site VPN links your **on-premises VPN device or gateway to the Azure VPN gateway in a virtual network**. In effect, the devices in Azure can appear as being on the local network. The connection is encrypted and works over the internet.
  - **Azure ExpressRoute**
    For environments where you need greater bandwidth and even higher levels of security, Azure ExpressRoute is the best approach. ExpressRoute provides a dedicated private connectivity to Azure that **doesn't travel over the internet.**
- **Route network traffic**

  By default, Azure routes traffic between subnets on any connected virtual networks, on-premises networks, and the internet. You can also control routing and override those settings, as follows:

  Route tables A route table allows you to define rules about how traffic should be directed. You can create custom route tables that control how packets are routed

between subnets.
Border Gateway Protocol Border Gateway Protocol (BGP) works with Azure VPN gateways, Azure Route Server, or ExpressRoute to propagate on-premises BGP routes to Azure virtual networks.

- **Filter network traffic**

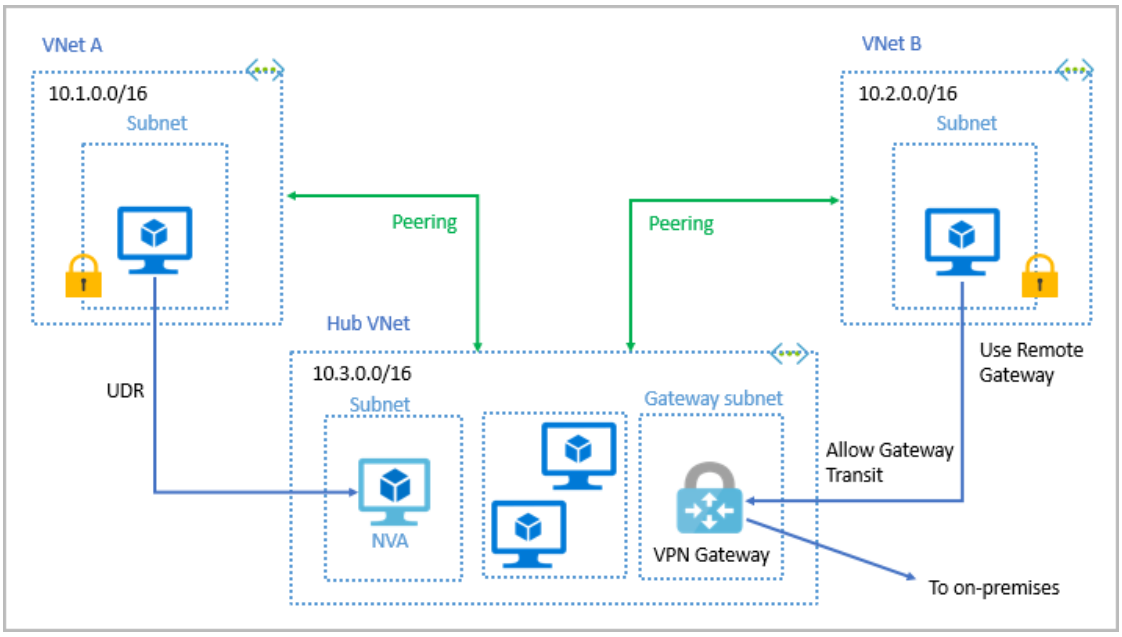
  Azure virtual networks enable you to filter traffic between subnets by using the following approaches:

  - A **network security group** is an Azure resource that can contain multiple inbound and outbound security rules. You can define these rules to allow or block traffic, based on factors such as source and destination IP address, port, and protocol.
  - A **network virtual appliance** is a specialized VM that can be compared to a hardened network appliance. A network virtual appliance carries out a particular network function, such as running a firewall or performing wide area network (WAN) optimization.

- **Connect virtual networks**

  You can link virtual networks together by using virtual network peering. Peering enables resources in each virtual network to communicate with each other. These virtual networks can be in separate regions, which allows you to create a global interconnected network through Azure.

  - **User-defined routes (UDR)** are a significant update to Azure's Virtual Networks that allows for greater control over network traffic flow. This method allows network administrators to control the routing tables between subnets within a VNet, as well as between VNets.



Azure Virtual Network	Connects VMs to incoming virtual private network (VPN) connections
Azure Load Balancer	Balances inbound and outbound connections to applications or service endpoints
Azure Application Gateway	Optimizes app server farm delivery while increasing application security
Azure VPN Gateway	Accesses Azure Virtual Networks through high-performance VPN gateways VPNs use an encrypted tunnel within another network. They're typically deployed to connect two or more trusted private networks to one another over an untrusted network (typically the public internet). Traffic is encrypted while traveling over the untrusted network to prevent eavesdropping or other attacks
Azure DNS	Provides ultra-fast DNS responses and ultra-high domain availability
Azure Content Delivery Network	Delivers high-bandwidth content to customers globally
Azure DDoS Protection	Protects Azure-hosted applications from distributed denial of service (DDOS) attacks
Azure Traffic Manager	Distributes network traffic across Azure regions worldwide
Azure ExpressRoute	Connects to Azure over high-bandwidth dedicated secure connections
Azure Network Watcher	Monitors and diagnoses network issues by using scenario-based analysis

Azure Firewall	Implements high-security, high-availability firewall with unlimited scalability
Azure Virtual WAN	Creates a unified wide area network (WAN) that connects local and remote sites

PART 3: Describe core solutions and management tools on Azure

▼ Core Solutions (Deleted)

⭐IoT Services

Smart devices are equipped with sensors that collect data. A few common sensors that measure attributes of the physical world include:

- Environmental sensors that capture temperature and humidity levels.
- Barcode, QR code, or optical character recognition (OCR) scanners.
- Motion and touch sensors.
- Smoke, gas, and alcohol sensors.
- Flow, level, and pressure sensors for measuring gasses and liquids.

Devices that are equipped with these kinds of sensors and that can connect to the internet could send their sensor readings to a specific endpoint in Azure via a **message**.

Azure IoT Hub

> ***▶️ Messaging hub that provides secure bi-directional communications between and monitoring of millions of IoT devices.***

It supports multiple messaging patterns, such as device-to-cloud telemetry, file upload from devices, and request-reply methods to control your devices from the cloud. After

an IoT hub receives messages from a device, it can route that message to other Azure services.

Azure IoT Central

> ***▶️ Fully managed global IoT software as a service (SaaS) solution that makes it easy to connect, monitor, and manage IoT assets at scale. Its built on top of IoT hub by adding a dashboard that allows you to connect, monitor, and manage your IoT devices. The visual user interface (UI) makes it easy to quickly connect new devices and watch as they begin sending telemetry or error messages.***

If you want a pre-built customizable user interface with which you can view and control your devices remotely, you might prefer to start with IoT Central. Otherwise you might prefer to implement Azure IoT Hub by itself. Your programmers can still create a customized set of management tools and reports by using the IoT Hub RESTful API.

Azure Sphere

> ***▶️ Azure Sphere creates an end-to-end, HIGHLY SECURE IoT solution for customers that encompasses everything from the hardware and operating system on the device to the secure method of sending messages from the device to the message hub. Azure Sphere has built-in communication and security features for internet-connected devices.***

The first part is the Azure Sphere micro-controller unit (MCU), which is responsible for processing the operating system and signals from attached sensors.

The second part is a customized Linux operating system (OS) that handles communication with the security service and can run the vendor's software.

The third part is **Azure Sphere Security Service**, also known as AS3. Its job is to make sure that the device has not been maliciously compromised. When the device attempts to connect to Azure, it first must authenticate itself, per device, which it does by using certificate-based authentication.

When security is a critical consideration in your product's design, the best product option is Azure Sphere, which provides a comprehensive end-to-end solution for IoT devices.

⭐AI Services And Solutions

> 💡 **Web API**: An API that's accessible from servers that accept requests via HTTP.
> **Web API endpoint:** The location of the code library.
> **REST API:** The design of the URL style that's used to expose the API's functionality.

At a high level, there are three primary product offerings from Microsoft, each of which is designed for a specific audience and use case.

Azure Machine Learning Service

> ***▶️Cloud-based environment you can use to develop, train, test, deploy, manage, and track machine learning models. It can auto-generate a model and auto-tune it for you. It will let you start training on your local machine, and then scale out to the cloud.***

Automated machine learning (**AutoML**)	This feature enables non-experts to quickly create an effective machine learning model from data. We identify the problem - **classification, regression, or time-series forecasting** followed by choosing the environment-Python SDK or ML Studio. We configure the Auto ML parameters and submit a training run that will go through all the possible algorithms.

Azure Machine Learning Designer	A graphical interface enabling no-code development of machine learning solutions. No code is required as we drag and drop the dataset and choose the algorithm. We can remove columns/rows as well as clean and normalize the dataset.

Azure Cognitive Services

> ***▶️ Provides prebuilt machine learning models that enable applications to see, hear, speak, understand, and even begin to reason. Use Azure Cognitive Services to solve general problems, such as analyzing text for emotional sentiment or analyzing images to recognize objects or faces. You don't need special machine learning or data science knowledge to use these services. Developers access Azure Cognitive Services via APIs and can easily include these features in just a few lines of code.***

Azure Cognitive Services can be divided into the following categories:
Language services: Allow your apps to process natural language with prebuilt scripts, evaluate sentiment, and learn how to recognize what users want.
Speech services: Convert speech into text and text into natural-sounding speech. Translate from one language to another and enable speaker verification and recognition.
Vision services: Add recognition and identification capabilities when you're analyzing pictures, videos, and other visual content.
Decision services: Add personalized recommendations for each user that automatically improve each time they're used, moderate content to monitor and remove offensive or risky content, and detect abnormalities in your time series data.

Azure Bot Service

> ***▶️ Bot Framework are platforms for creating virtual agents that understand and reply to questions just like a human.***

> ***Namely, it creates a virtual agent that can intelligently communicate with humans.***

This service provides a platform for **conversational AI**, the capability of a software "agent" to participate in a conversation. Developers can use the *Bot Framework* to create a bot and manage it with Azure Bot Service - integrating back-end services like Language (Azure Cogntive Services), and connecting to channels for web chat, email, Microsoft Teams, and others.

⭐Software Development Process Services

Microsoft tools enable source-code management, continuous integration and continuous delivery (CI/CD), and automate the creation of testing environments (DevOps)

Azure DevOps Services (SaaS)

> ***▶️ Azure DevOps Services is a suite of services that address every stage of the software development lifecycle.***

Azure Repos is a centralized source-code repository where software development, DevOps engineering, and documentation professionals can publish their code for review and collaboration.
Azure Boards is an agile project management suite that includes Kanban boards, reporting, and tracking ideas and work from high-level epics to work items and issues.
Azure Pipelines is a CI/CD pipeline automation tool.
Azure Artifacts is a repository for hosting artifacts, such as compiled source code, which can be fed into testing or deployment pipeline steps.
Azure Test Plans is an automated test tool that can be used in a CI/CD pipeline to ensure quality before a software release.

Github and Github Actions

> ***▶️ Git is a decentralized source-code management tool, and GitHub is a hosted version of Git that serves as the primary***

> ***remote. GitHub builds on top of Git to provide related services for coordinating work, reporting and discussing issues, providing documentation, and more.GitHub Actions enables workflow automation with triggers for many lifecycle events.***

One such example would be automating a CI/CD toolchain.
(A toolchain is a combination of software tools that aid in the delivery, development, and management of software applications throughout a system's development lifecycle. The output of one tool in the toolchain is the input of the next tool in the toolchain. Typical tool functions range from performing automated dependency updates to building and configuring the software, delivering the build artifacts to various locations, testing, and so on.)

Although both Azure DevOps and GitHub allow public and private code repositories, GitHub has a long history with public repositories and is trusted by tens of thousands of open-source project owners. **GitHub is a lighter-weight tool than Azure DevOps**, with a focus on individual developers contributing to the open-source code. **Azure DevOps, on the other hand, is more focused on enterprise development, with heavier project-management and planning tools, and finer-grained access control.**

Azure DevTest Labs

> ***▶️ Azure DevTest Labs provides an automated means of managing the process of building, setting up, and tearing down virtual machines (VMs) that contain builds of your software projects.***

This helps developers and testers perform tests across a variety of environments and builds. And this capability isn't limited to VMs. Anything you can deploy in Azure via an ARM template can be provisioned through DevTest Labs. Provisioning pre-created lab environments with their required configurations and tools already installed is a huge time saver for quality assurance professionals and developers.

Suppose you need to test a new feature on an old version of an operating system. Azure DevTest Labs can set up everything automatically upon request. After the testing

is complete, DevTest Labs can shut down and deprovision the VM, which saves money when it's not in use.

Monitoring and Managing

At a high level, there are two broad categories of management tools: **visual tools** and **code-based tools.**

When you're attempting to quickly set up and configure Azure resources, a code-based tool is usually the better choice. Although it might take time to understand the right commands and parameters at first, after they've been entered, they can be saved into files and used repeatedly as needed. Also, the code that performs setup and configuration can be stored, versioned, and maintained along with application source code in a source code-management tool such as Git. This approach to managing hardware and cloud resources, which developers use when they write application code, is referred to as **infrastructure as code**.

There are two approaches to infrastructure as code: **imperative code** and **declarative code**.

Imperative code details each individual step that should be performed to achieve a desired outcome.
Declarative code details only a desired outcome, and it allows an interpreter to decide how to best achieve that outcome. This distinction is important because tools that are based on declarative code can provide a more robust approach to deploying dozens or hundreds of resources simultaneously and reliably.

Azure Portal	**The Azure portal provides a friendly, graphical UI to view all the services** you're using, create new services, configure your services, and view reports. The Azure portal is how most users first experience Azure. But, as your Azure usage grows, you'll likely choose a more repeatable code-centric approach to managing your Azure resources.
Azure Mobile App	Access to your Azure resources when you're away from your computer. With it, you can: -Monitor the health and status of your Azure resources. -Check for alerts, quickly diagnose and fix issues, and restart a web app or virtual machine (VM). -Run the Azure CLI or Azure PowerShell commands to manage your Azure resources.
Azure	**Azure PowerShell is a shell with which developers and DevOps and IT**

powerShell (Windows Background)	**professionals can execute commands called cmdlets (pronounced command-lets). These commands call the Azure Rest API to perform every possible management task in Azure.** Cmdlets can be executed independently or combined into a script file and executed together to orchestrate: -The routine setup, teardown, and maintenance of a single resource or multiple connected resources. -The deployment of an entire infrastructure, which might contain dozens or hundreds of resources, from imperative code. Capturing the commands in a script makes the process repeatable and automatable.
Azure CLI (Linux Background)	In many respects, the Azure CLI is almost identical to Azure PowerShell in what you can do with it. Both run on Windows, Linux, and Mac, and can be accessed in a web browser via Cloud Shell. The primary difference is the syntax you use. If you're already proficient in PowerShell or Bash, you can use the tool you prefer.
ARM templates	**By using Azure Resource Manager templates (ARM templates), you can describe the resources you want to use in a declarative JSON format. The benefit is that the entire ARM template is verified before any code is executed to ensure that the resources will be created and connected correctly.** The template then orchestrates the creation of those resources in parallel. That is, if you need 50 instances of the same resource, all 50 instances are created at the same time. Ultimately, the developer, DevOps professional, or IT professional needs only to define the desired state and configuration of each resource in the ARM template, and the template does the rest. Templates can even execute PowerShell and Bash scripts before or after the resource has been set up.

Azure PowerShell and the Azure CLI are Azure management tools that allow you to quickly obtain the IP address of a virtual machine (VM) you've deployed, reboot a VM, or scale an app. You might want to keep custom scripts for both tools handy on your local hard drive for certain operations that you need to perform multiple times.

In contrast to the Azure CLI and PowerShell, Azure Resource Manager templates (ARM templates) define the infrastructure requirements in your application for repeatable deployments. **A validation step ensures that all resources can be created in the proper order based on dependencies, in parallel, and idempotent.** Although ARM templates aren't intended for one-off scenarios, it's possible to use them for this purpose. However, for one-off scenarios, you may prefer more agile tools like PowerShell, Azure CLI scripts, or the Azure portal.

Keep in mind that **ARM templates can include both PowerShell and/or Azure CLI scripts, which will give you the ability to utilize scripts for tasks that may not be possible with the ARM template itself.** The ability to combine Azure management tools gives flexibility in choosing the right tool(s) for your particular need.

⭐Visibility, Insight, and Outage Mitigation

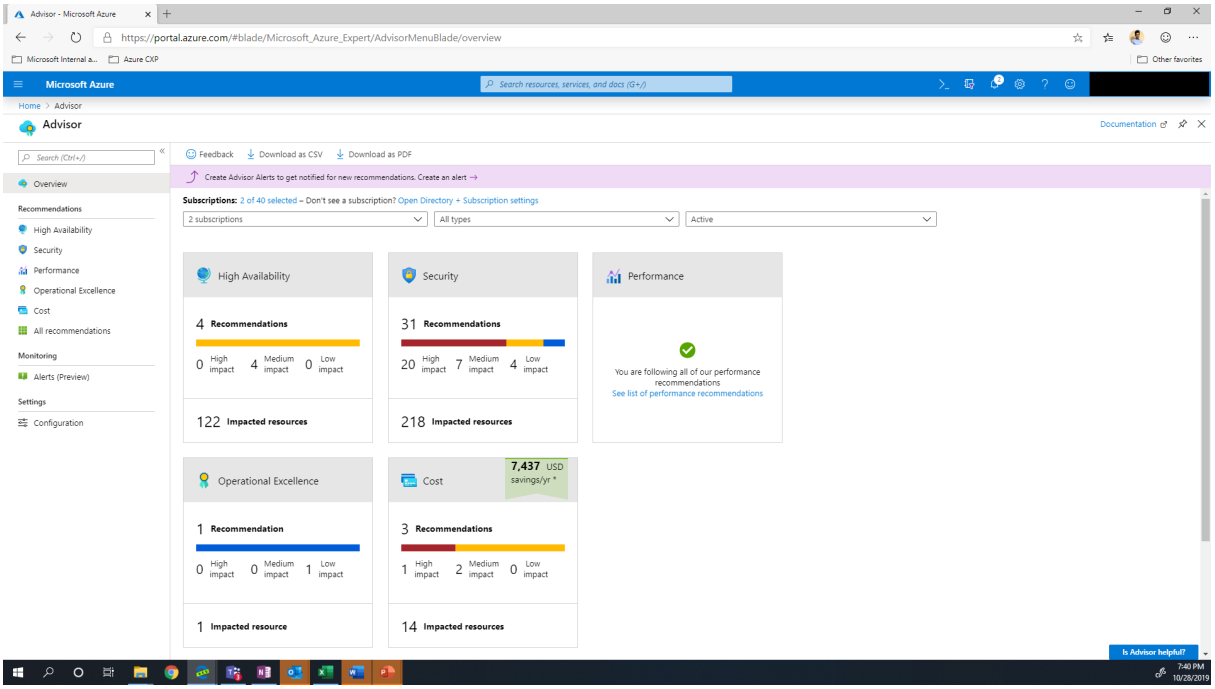
Several basic questions or concerns face all companies that use the cloud.

Are we using the cloud correctly? Can we squeeze more performance out of our cloud spend? Are we spending more than we need to?
Do we have our systems properly secured? How resilient are our resources?
How can we diagnose and fix issues that occur intermittently?
How can we quickly determine the cause of an outage?
How can we learn about planned downtime?

Azure Advisor

> ***Evaluates your Azure resources and makes recommendations to help improve reliability, security, and performance, achieve operational excellence, and reduce costs. Advisor is designed to help you save time on cloud optimization.***

The recommendations are divided into five categories:
Reliability: Used to ensure and improve the continuity of your business-critical applications.
Security: Used to detect threats and vulnerabilities that might lead to security breaches.
Performance: Used to improve the speed of your applications.
Cost: Used to optimize and reduce your overall Azure spending.
Operational Excellence: Used to help you achieve process and workflow efficiency, resource manageability, and deployment best practices.

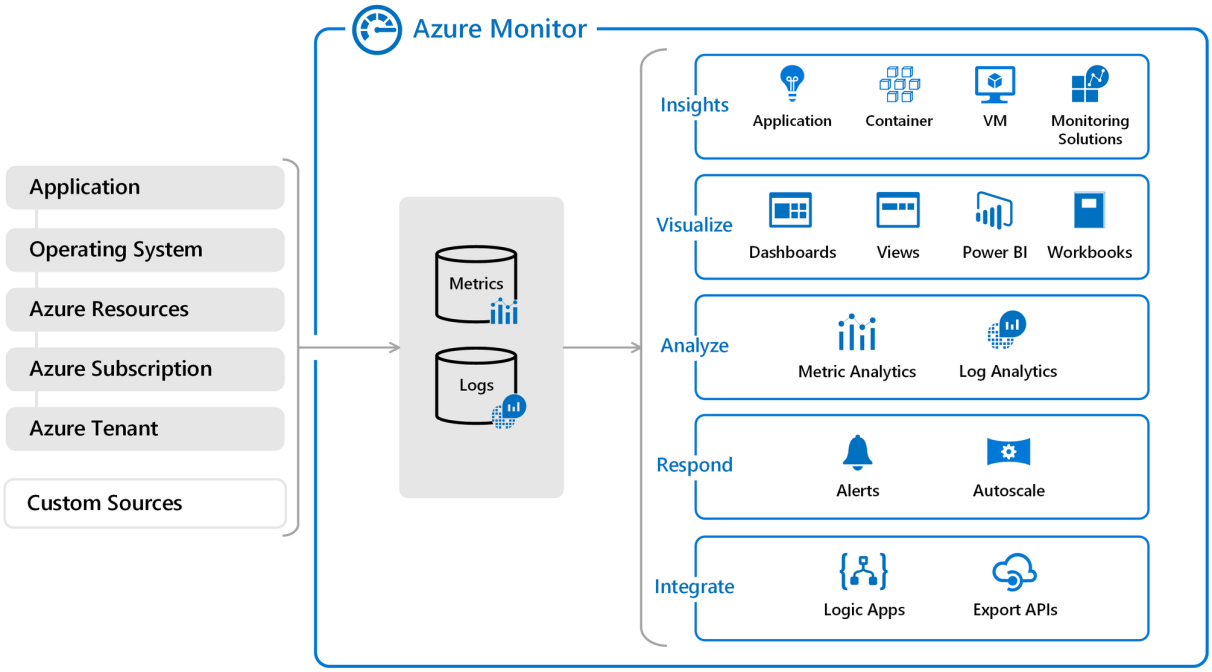


Azure Monitor

> ***Platform for collecting, analyzing, visualizing, and potentially taking action based on the metric and logging data from your entire Azure and on-premises environment.***

You can view real-time and historical performance across each layer of your architecture, or aggregated and detailed information. The data is displayed at different levels for different audiences. You can view high-level reports on the Azure Monitor Dashboard or create custom views by using Power BI and Kusto queries.

Additionally, you can use the data to help you react to critical events in real time, through alerts delivered to teams via SMS, email, and so on. Or you can use thresholds to trigger autoscaling functionality to scale up or down to meet the demand.



Azure Service Health

> ***Provides a personalized view of the health of the Azure services, regions, and resources you rely on. The status.azure.com website, which displays only major issues that broadly affect Azure customers, doesn't provide the full picture. But Azure Service Health displays both major and smaller, localized issues that affect you.***

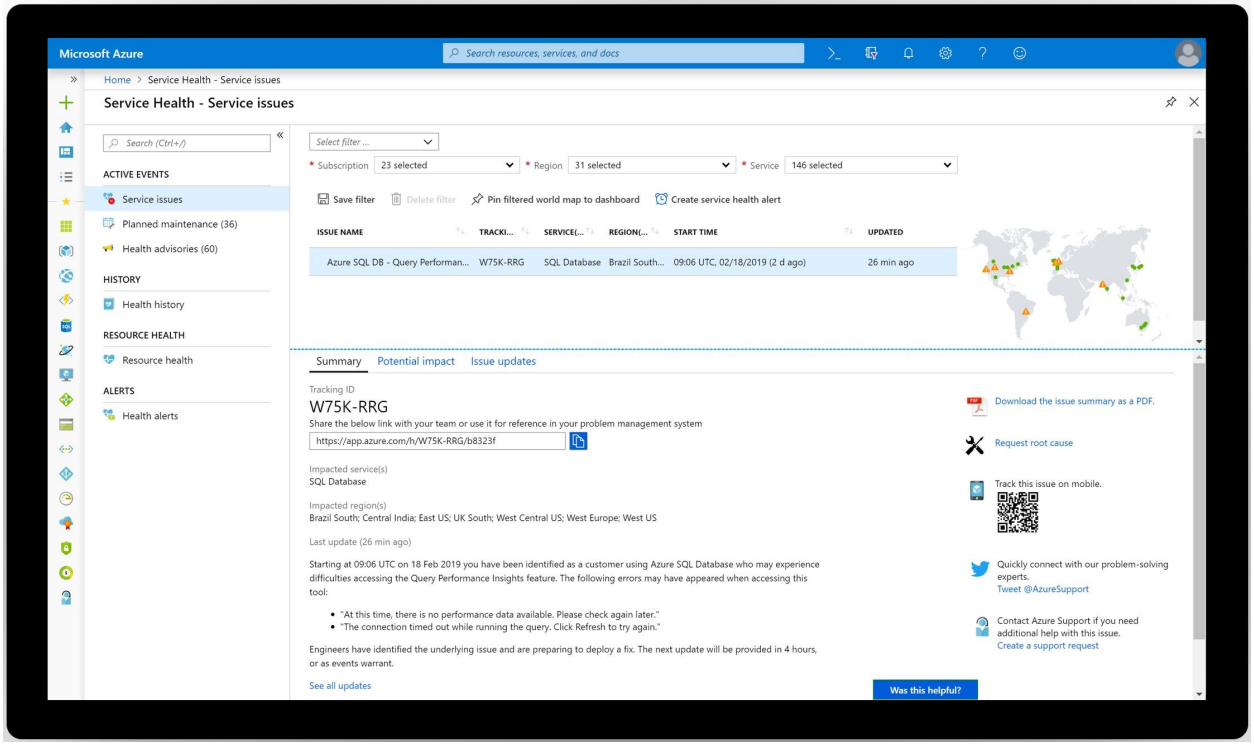
Service issues are rare, but it's important to be prepared for the unexpected. You can set up alerts that help you triage outages and planned maintenance. After an outage, Service Health provides official incident reports, called root cause analyses (RCAs), which you can share with stakeholders.

Service Health helps you keep an eye on several event types:

Service issues are problems in Azure, such as outages, that affect you right now. You can drill down to the affected services, regions, updates from your engineering teams, and find ways to share and track the latest information.

Planned maintenance events can affect your availability. You can drill down to the affected services, regions, and details to show how an event will affect you and what you need to do. Most of these events occur without any impact to you and aren't shown here. In the rare case that a reboot is required, Service Health allows you to choose when to perform the maintenance to minimize the downtime.

Health advisories are issues that require you to act to avoid service interruption, including service retirements and breaking changes. Health advisories are announced far in advance to allow you to plan.



PART 4: Describe general security and network security features

Network Security

Azure Security Center provides visibility of your security posture across all of your services, both on Azure and on-premises.

Azure Sentinel aggregates security data from many different sources, and provides additional capabilities for threat detection and response.

Azure Key Vault stores your applications' secrets, such as passwords, encryption keys, and certificates, in a single, central location.

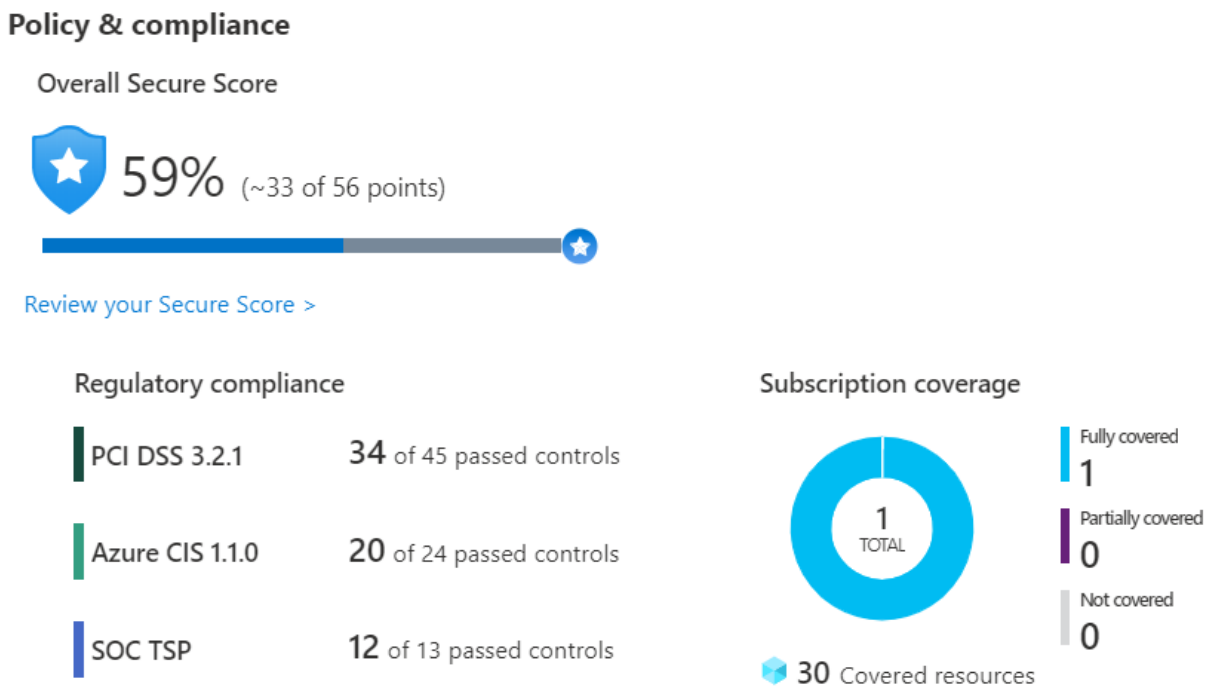
Azure Dedicated Host provides dedicated physical servers to host your Azure VMs for Windows and Linux.

Microsoft Defender for Cloud (Previously Azure Security Center)

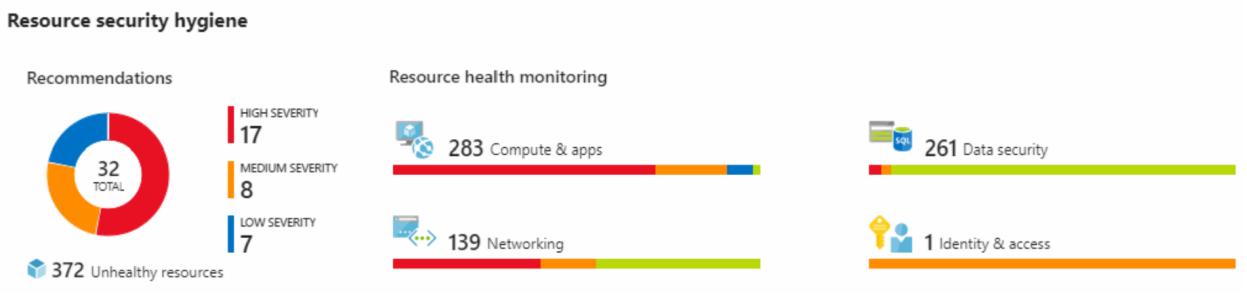
> ***Cloud Workload Protection Platform (CWPP) that also delivers Cloud Security Posture Management (CSPM) for all of your Azure, on-premises, and multi-cloud (Amazon AWS and Google GCP) resources. Thus it is a monitoring service that provides visibility of your security posture across all of your services. The term security posture refers to cybersecurity policies and controls, as well as how well you can predict, prevent, and respond to security threats.***

Security Center can:

-**Monitor security settings** across on-premises and cloud workloads.
-**Automatically apply required security settings** to new resources as they come online.
-**Provide security recommendations** that are based on your current configurations, resources, and networks.
-**Continuously monitor your resources** and perform automatic security assessments to identify potential vulnerabilities before those vulnerabilities can be exploited.
-**Use machine learning to detect and block malware** from being installed on your virtual machines (VMs) and other resources. You can also use **adaptive application controls** to define rules that list allowed applications to ensure that only applications you allow can run.
-**Detect and analyze potential inbound attacks** and investigate threats and any post-breach activity that might have occurred.
-**Provide just-in-time access control for network ports**. Doing so reduces your attack surface by ensuring that the network only allows traffic that you require at the time that you need it to.



Secure score is based on security controls, or groups of related security recommendations. Your score is based on the percentage of security controls that you satisfy. The more security controls you satisfy, the higher the score you receive. Your score improves when you remediate all of the recommendations for a single resource within a control.



In the **Resource security hygiene** section, you can see the health of resources from a security perspective. To help prioritize remediation actions, recommendations are categorized as low, medium, and high

Security Center includes advanced cloud defense capabilities for VMs, network security, and file integrity

- **Just-in-time VM access**: This access blocks traffic by default to specific network ports of VMs, but **allows traffic for a specified time** when an admin requests and approves it.
- **Adaptive application controls:** Control which applications are allowed to run on VMs. In the background, Security Center uses machine learning to look at the processes running on a VM. It creates exception rules for each resource group that holds the VMs and provides recommendations. This process provides alerts that inform the company about unauthorized applications that are running on its VMs.
- **Adaptive network hardening:** Security Center can **monitor the internet traffic patterns** of the VMs, and compare those patterns with the **company's current network security group (NSG)** settings. From there, Security Center can make recommendations about whether the NSGs should be locked down further and provide remediation steps.

- **File integrity monitoring:** Configure the monitoring of changes to important files on both Windows and Linux, registry settings, applications, and other aspects that might indicate a security attack.

Azure Sentinel

Security management on a large scale can benefit from a dedicated security information and event management (SIEM) system. A SIEM system aggregates security data from many different sources (as long as those sources support an open-standard logging format). It also provides capabilities for threat detection and response.

> ***Azure Sentinel is Microsoft's cloud-based SIEM system. It uses intelligent security analytics and threat analysis.***

Azure Sentinel enables you to:

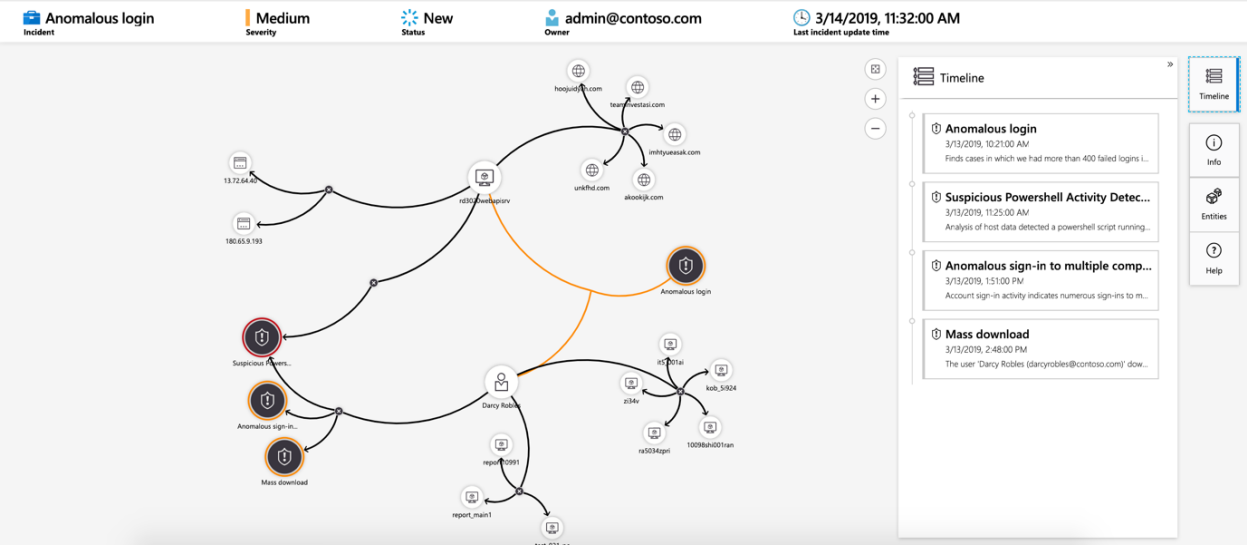
- **Collect cloud data at scale** Collect data across all users, devices, applications, and infrastructure, both on-premises and from multiple clouds.
- **Detect previously undetected threats** Minimize false positives by using Microsoft's comprehensive analytics and threat intelligence.
- **Investigate threats with artificial intelligence** Examine suspicious activities at scale, tapping into years of cybersecurity experience from Microsoft.
- **Respond to incidents rapidly** Use built-in orchestration and automation of common tasks.

Azure Sentinel supports a number of data sources, which it can analyze for security events. These connections are handled by built-in connectors or industry-standard log formats and APIs.

- **Connect Microsoft solutions** Connectors provide real-time integration for services like Microsoft Threat Protection solutions, Microsoft 365 sources (including Office 365), Azure Active Directory, and Windows Defender Firewall.
- **Connect other services and solutions** Connectors are available for common non-Microsoft services and solutions, including AWS CloudTrail, Citrix Analytics (Security), Sophos XG Firewall, VMware Carbon Black Cloud, and Okta SSO.
- **Connect industry-standard data sources** Azure Sentinel supports data from other sources that use the Common Event Format (CEF) messaging standard, Syslog, or

REST API.

With the investigation graph, the company can review information from entities directly connected to the alert, and see common exploration queries to help guide the investigation.



Use **Azure Monitor Workbooks** to automate responses to threats. For example, it can set an alert that looks for malicious IP addresses that access the network and create a workbook that does the following steps:

1. When the alert is triggered, open a ticket in the IT ticketing system.
2. Send a message to the security operations channel in Microsoft Teams or Slack to make sure the security analysts are aware of the incident.
3. Send all of the information in the alert to the senior network admin and to the security admin. The email message includes two user option buttons: **Block** or **Ignore**.

When an admin chooses **Block**, the IP address is blocked in the firewall, and the user is disabled in Azure Active Directory. When an admin chooses **Ignore**, the alert is closed in Azure Sentinel, and the incident is closed in the IT ticketing system.

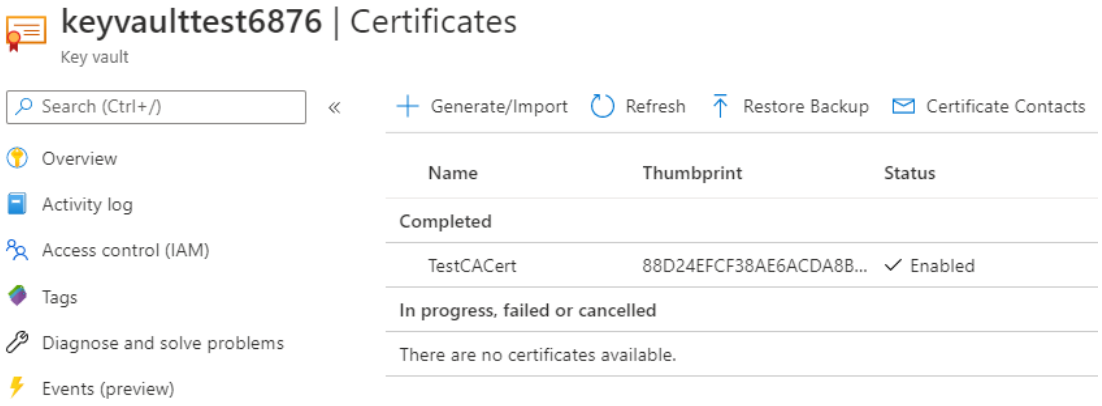
Azure Key Vault

> ***Azure Key Vault is a centralized cloud service for storing an application's secrets in a single, central location. It provides***

> ***secure access to sensitive information by providing access control and logging capabilities.***

Azure Key Vault can help you:

- **Manage secrets** You can use Key Vault to securely store and tightly control access to tokens, passwords, certificates, API keys, and other secrets.
- **Manage encryption keys** You can use Key Vault as a key management solution. Key Vault makes it easier to create and control the encryption keys that are used to encrypt your data.
- **Manage SSL/TLS certificates** Key Vault enables you to provision, manage, and deploy your public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for both your Azure resources and your internal resources.
- **Store secrets backed by hardware security modules (HSMs)** These secrets and keys can be protected either by software or by FIPS 140-2 Level 2 validated HSMs.



The benefits of using Key Vault include:

- **Centralized application secrets** Centralizing the storage for your application secrets enables you to control their distribution, and reduces the chances that secrets are accidentally leaked.
- **Securely stored secrets and keys** Azure uses industry-standard algorithms, key lengths, and HSMs. Access to Key Vault requires proper authentication and authorization.

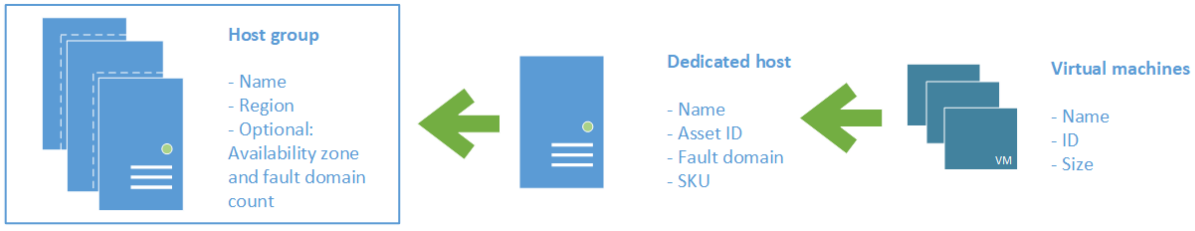
- **Access monitoring and access control** By using Key Vault, you can monitor and control access to your application secrets.
- **Simplified administration of application secrets** Key Vault makes it easier to enroll and renew certificates from public certificate authorities (CAs). You can also scale up and replicate content within regions and use standard certificate management tools.
- **Integration with other Azure services** You can integrate Key Vault with storage accounts, container registries, event hubs, and many more Azure services. These services can then securely reference the secrets stored in Key Vault.

Azure Dedicated Host

On Azure, virtual machines (VMs) run on shared hardware that Microsoft manages. Although the underlying hardware is shared, your VM workloads are isolated from workloads that other Azure customers run.

> ***Some organizations must follow regulatory compliance that requires them to be the only customer using the physical machine that hosts their virtual machines. Azure Dedicated Host provides dedicated physical servers to host your Azure VMs for Windows and Linux.***

Here's a diagram that shows how VMs relate to dedicated hosts and host groups. A dedicated host is mapped to a physical server in an Azure datacenter. A host group is a collection of dedicated hosts.



Azure Dedicated Host:

- Gives you visibility into, and control over, the server infrastructure that's running your Azure VMs.

- Helps address compliance requirements by deploying your workloads on an isolated server.
- Lets you choose the number of processors, server capabilities, VM series, and VM sizes within the same host.

You're charged per dedicated host, independent of how many VMs you deploy to it. The host price is based on the VM family, type (hardware size), and region.

Software licensing, storage, and network usage are billed separately from the host and VMs.

Secure Network Connectivity

Azure Firewall is a managed, cloud-based network security service that helps protect resources in Azure virtual networks.

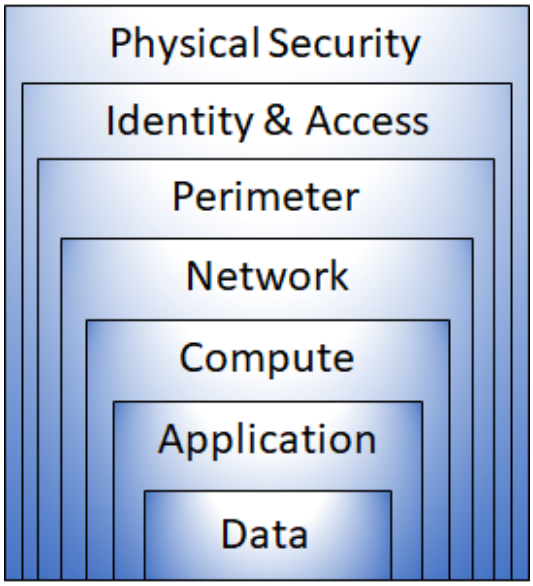
An **Azure virtual network** is similar to a traditional network that you'd operate in your own datacenter. It enables virtual machines and other compute resources to securely communicate with each other, the internet, and on-premises networks.

A **network security group (NSG)** enables you to filter network traffic to and from Azure resources within a virtual network.

Azure DDoS Protection helps protect Azure resources from DDoS attacks.

Defense in Depth

> ***The objective of defense in depth is to protect information and prevent it from being stolen by those who aren't authorized to access it. A defense-in-depth strategy uses a series of mechanisms to slow the advance of an attack that aims at acquiring unauthorized access to data.***



Each layer provides protection so that if one layer is breached, a subsequent layer is already in place to prevent further exposure.

- The *physical security* layer is the first line of defense to protect computing hardware in the datacenter.
- The *identity and access* layer controls access to infrastructure and change control. The identity and access layer is all about ensuring that identities are secure, access is granted only to what's needed, and sign-in events and changes are logged.
- The *perimeter* layer uses distributed denial of service (DDoS) protection to filter large-scale attacks before they can cause a denial of service for users.
- The *network* layer limits communication between resources through segmentation and access controls.
- The *compute* layer secures access to virtual machines. Malware, unpatched systems, and improperly secured systems open your environment to attacks. The focus in this layer is on making sure that your compute resources are secure and that you have the proper controls in place to minimize security issues.
- The *application* layer helps ensure that applications are secure and free of security vulnerabilities. Integrating security into the application development lifecycle helps reduce the number of vulnerabilities introduced in code. Every development team should ensure that its applications are secure by default.

- The *data* layer controls access to business and customer data that you need to protect.

Security posture

> **Your *security posture* is your organization's ability to protect from and respond to security threats. The common principles used to define a security posture are *confidentiality*, *integrity*, and *availability*, known collectively as CIA.**

- **Confidentiality**

  The ***principle of least privilege*** means restricting access to information only to individuals explicitly granted access, at only the level that they need to perform their work.

- **Integrity**

  Prevent unauthorized changes to information:

  - At rest: when it's stored.
  - In transit: when it's being transferred from one place to another, including from a local computer to the cloud.

  A common approach used in data transmission is for the sender to create a unique fingerprint of the data by using a one-way hashing algorithm. The hash is sent to the receiver along with the data. The receiver recalculates the data's hash and compares it to the original to ensure that the data wasn't lost or modified in transit.

- **Availability**

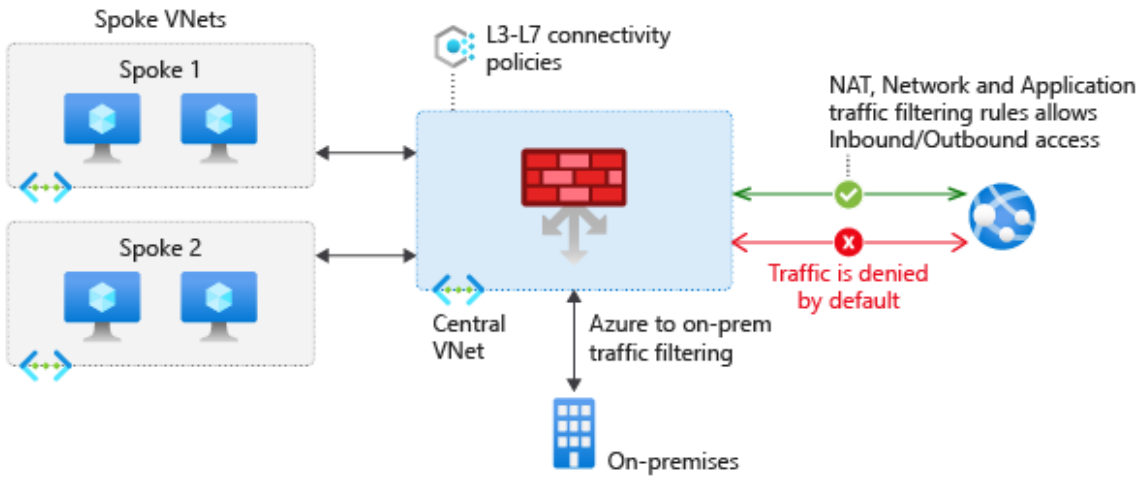
  Ensure that services are functioning and can be accessed only by authorized users. ***Denial-of-service attacks*** are designed to degrade the availability of a system, affecting its users.

Azure Firewall

A *firewall* is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules. You can create firewall rules that specify ranges of IP addresses. Only clients

granted IP addresses from within those ranges are allowed to access the destination server. Firewall rules can also include specific network protocol and port information.

> ***Azure Firewall is a managed, cloud-based network security service that helps protect resources in your Azure virtual networks.***



Azure Firewall is a ***stateful*** firewall. A stateful firewall analyzes the complete context of a network connection, not just an individual packet of network traffic.

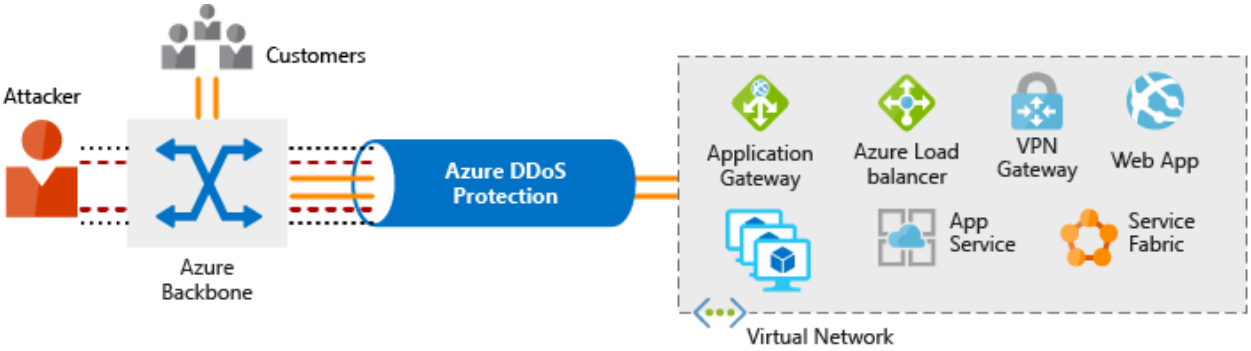
Azure Firewall provides a central location to create, enforce, and log application and network connectivity policies across subscriptions and virtual networks. Azure Firewall **uses a static (unchanging) public IP address** for your virtual network resources, which enables outside firewalls to identify traffic coming from your virtual network. The service is integrated with Azure Monitor to enable logging and analytics.

Azure Application Gateway *also provides a firewall that's called the* ***web application firewall (WAF)****. WAF provides centralized, inbound protection for your web applications against common exploits and vulnerabilities.* ***Azure Front Door*** *and* ***Azure Content Delivery Network*** *also provide WAF services.*

Azure DDos Protection

A distributed denial of service attack attempts to overwhelm and exhaust an application's resources, making the application slow or unresponsive to legitimate users. DDoS attacks can target any resource that's publicly reachable through the internet, including websites.

> ***Azure DDoS Protection (Standard) helps protect your Azure resources from DDoS attacks by analyzing and discarding DDoS traffic at the Azure network edge, before it can affect your service's availability.***



In the cloud, elastic computing means that you can automatically scale out your deployment to meet demand. A cleverly designed DDoS attack can cause you to increase your resource allocation, which incurs unneeded expense. DDoS Protection Standard helps ensure that the network load you process reflects customer usage.

DDoS Protection provides two service tiers:

- **Basic**

  The Basic service tier is automatically enabled for free as part of your Azure subscription. The Basic service tier ensures that Azure infrastructure itself is not affected during a large-scale DDoS attack.

- **Standard**

  The Standard service tier provides additional mitigation capabilities that are tuned specifically to Azure Virtual Network resources. Protection policies are tuned through dedicated traffic monitoring and machine learning algorithms. Policies are applied to public IP addresses, which are associated with resources deployed in virtual networks such as Azure Load Balancer and Application Gateway.

  The Standard service tier can help prevent:

  - **Volumetric attacks**

The goal of this attack is to flood the network layer with a substantial amount of seemingly legitimate traffic.

- **Protocol attacks**

  These attacks render a target inaccessible by exploiting a weakness in the layer 3 and layer 4 protocol stack.

- **Resource-layer (application-layer) attacks (only with web application firewall)**

  These attacks target web application packets to disrupt the transmission of data between hosts. You need a web application firewall (WAF) to protect against L7 attacks. DDoS Protection Standard protects the WAF from volumetric and protocol attacks.

Filter network traffic by using network security groups (NSG)

Although Azure Firewall and Azure DDoS Protection can help control what traffic can come from outside sources, its important to understand how to protect internal networks on Azure too.

> ***A network security group enables you to filter network traffic to and from Azure resources within an Azure virtual network. You can think of NSGs like an internal firewall. An NSG can contain multiple inbound and outbound security rules that enable you to filter traffic to and from resources by source and destination IP address, port, and protocol.***

Describe Azure management and governance (30–35%)

Describe cost management in Azure

• describe factors that can affect costs in Azure
• compare the Pricing calculator and the Total Cost of Ownership (TCO) calculator
• describe the Azure Cost Management and Billing tool
• describe the purpose of tags

Describe features and tools in Azure for governance and compliance

• describe the purpose of Azure Blueprints

• describe the purpose of Azure Policy
• describe the purpose of resource locks
• describe the purpose of the Service Trust Portal

Describe features and tools for managing and deploying Azure resources
• describe the Azure portal
• describe Azure Cloud Shell, including Azure CLI and Azure PowerShell
• describe the purpose of **Azure Arc**
• describe Azure Resource Manager and Azure Resource Manager templates (ARM templates)

Describe monitoring tools in Azure
• describe the purpose of Azure Advisor
• describe Azure Service Health
• describe Azure Monitor, including Log Analytics, Azure Monitor alerts, and Application Insights

PART 5: Describe identity, governance, privacy, and compliance features

Azure Identity Services

Authentication (AuthN) establishes the user's identity.

Authorization (AuthZ) establishes the level of access that an authenticated user has.

Single sign-on (SSO) enables a user to sign in one time and use that credential to access multiple resources and applications.

Azure Active Directory (Azure AD) is a cloud-based identity and access management service. Azure AD enables an organization to control access to apps and resources based on its business requirements.

Azure AD Multi-Factor Authentication provides additional security for identities by requiring two or more elements to fully authenticate. In general, multifactor authentication can include something the user knows, something the user has, and something the user is.

Conditional Access is a tool that Azure AD uses to allow or deny access to resources based on identity signals such as the user's location.

Identity has become the new primary security boundary. Accurately proving that someone is a valid user of your system, with an appropriate level of access, is critical to maintaining control of your data. This identity layer is now more often the target of attack than the network is.

> ***Authentication is the process of establishing the identity of a person or service that wants to access a resource. It involves the act of challenging a party for legitimate credentials and provides the basis for creating a security principal for identity and access control. It establishes whether the user is who they say they are.***

> ***Authorization is the process of establishing what level of access an authenticated person or service has. It specifies what data they're allowed to access and what they can do with it.***

Active Directory (on-premise) & Azure AD (cloud)

For on-premises environments, Active Directory running on Windows Server provides an **identity and access management service** that's managed by your own organization. Azure AD is Microsoft's cloud-based identity and access management service. With Azure AD, you control the identity accounts, but Microsoft ensures that the service is available globally.

Azure AD is for:

- **IT administrators**

  Administrators can use Azure AD to control access to applications and resources based on their business requirements.

- **App developers**

  Developers can use Azure AD to provide a standards-based approach for adding functionality to applications that they build, such as adding SSO functionality to an app or enabling an app to work with a user's existing credentials.

- **Users**

  Users can manage their identities. For example, self-service password reset enables users to change or reset their password with no involvement from an IT administrator or help desk.

- **Online service subscribers**

  Microsoft 365, Microsoft Office 365, Azure, and Microsoft Dynamics CRM Online subscribers are already using Azure AD.

  > 💡 A ***tenant*** is a representation of an organization. A tenant is typically separated from other tenants and has its own identity.

  Each Microsoft 365, Office 365, Azure, and Dynamics CRM Online tenant is automatically an Azure AD tenant.

Azure AD provides services such as:

- **Authentication**

  This includes verifying identity to access applications and resources. It also includes providing functionality such as self-service password reset, multifactor authentication,

a custom list of banned passwords, and smart lockout services.

- **Single sign-on**

  SSO enables you to remember only one username and one password to access multiple applications. A single identity is tied to a user, which simplifies the security model. As users change roles or leave an organization, access modifications are tied to that identity, which greatly reduces the effort needed to change or disable accounts.

- **Application management**

  You can manage your cloud and on-premises apps by using Azure AD. Features like Application Proxy, SaaS apps, the My Apps portal (also called the *access panel*), and single sign-on provide a better user experience.

- **Device management**

  Along with accounts for individual people, Azure AD supports the registration of devices. Registration enables devices to be managed through tools like Microsoft Intune. It also allows for device-based Conditional Access policies to restrict access attempts to only those coming from known devices, regardless of the requesting user account.

There are a few ways to connect your existing Active Directory installation with Azure AD. Perhaps the most popular method is to use Azure AD Connect.

> ***Azure AD Connect synchronizes user identities between on-premises Active Directory and Azure AD. Azure AD Connect synchronizes changes between both identity systems, so you can use features like SSO, multifactor authentication, and self-service password reset under both systems.***

> ***Azure AD Multi-Factor Authentication is a Microsoft service that provides multifactor authentication capabilities. Azure AD Multi-Factor Authentication enables users to choose an additional***

> ***form of authentication during sign-in, such as a phone call or mobile app notification.***

Azure Active Directory Premium (P1 or P2 licenses) allows for comprehensive and granular configuration of Azure AD Multi-Factor Authentication through Conditional Access policies (explained shortly).

> ***Conditional Access is a tool that Azure Active Directory uses to allow (or deny) access to resources based on identity signals. These signals include who the user is, where the user is, and what device the user is requesting access from.***

Conditional Access also provides a more granular multifactor authentication experience for users. For example, a user might not be challenged for second authentication factor if they're at a known location. However, they might be challenged for a second authentication factor if their sign-in signals are unusual or they're at an unexpected location. To use Conditional Access, you need an Azure AD Premium P1 or P2 license.

Cloud Governance in Azure

> ***The term governance describes the general process of establishing rules and policies and ensuring that those rules and policies are enforced.***

When running in the cloud, a good governance strategy helps you maintain control over the applications and resources that you manage in the cloud. Maintaining control over your environment ensures that you stay compliant with:

- Industry standards, like PCI DSS.
- Corporate or organizational standards, such as ensuring that network data is encrypted.

Cloud governance requires good analysis and requirement gathering. Luckily, the Cloud Adoption Framework for Azure can help you define and implement your governance strategy. There are several services and features in Azure to support these efforts:

Azure role-based access control (Azure RBAC) enables you to create roles that define access permissions.

Resource locks prevent resources from being accidentally deleted or changed.

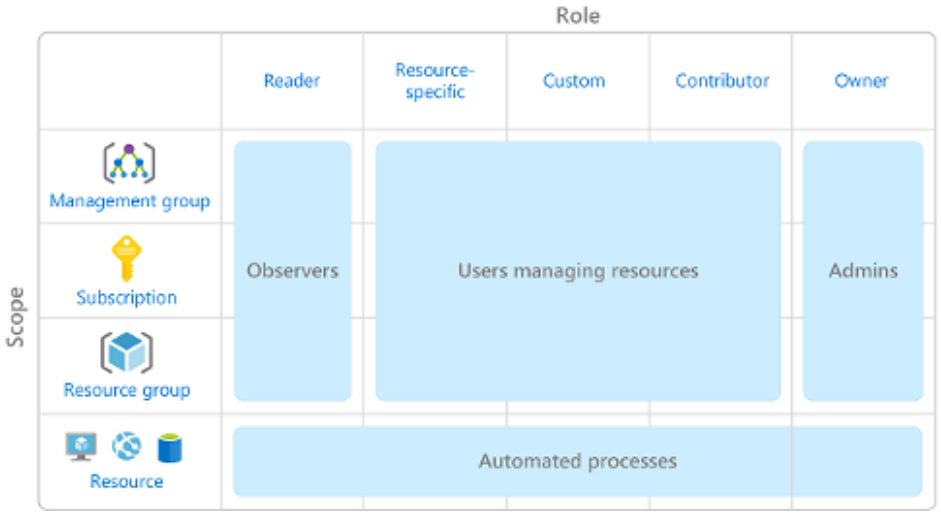
Resource tags provide extra information, or metadata, about your resources.

Azure Policy is a service in Azure that enables you to create, assign, and manage policies that control or audit your resources.

Azure Blueprints enables you to define a repeatable set of governance tools and standard Azure resources that your organization requires.

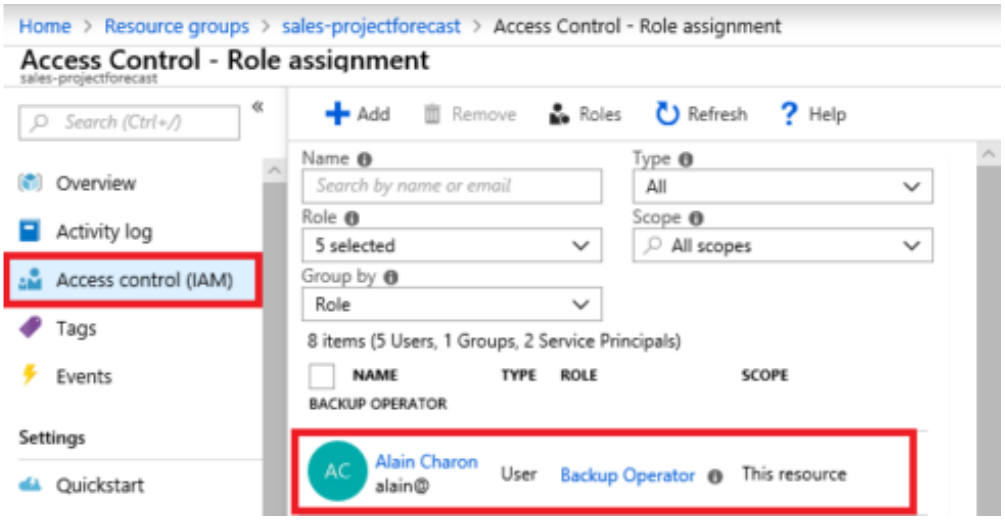
RBAC (Role based Access Control)

Instead of defining the detailed access requirements for each individual, and then updating access requirements when new resources are created, Azure enables you to control access through Azure role-based access control (Azure RBAC). Azure provides built-in roles that describe common access rules for cloud resources. You can also define your own roles. Each role has an associated set of access permissions that relate to that role. When you assign individuals or groups to one or more roles, they receive all of the associated access permissions.



You manage access permissions on the **Access control (IAM)** pane in the Azure portal. This pane shows who has access to what scope and what roles apply. You can also grant or remove access from this pane.

The following screenshot shows an example of the **Access control (IAM)** pane for a resource group. In this example, Alain Charon has been assigned the **Backup Operator** role for this resource group.

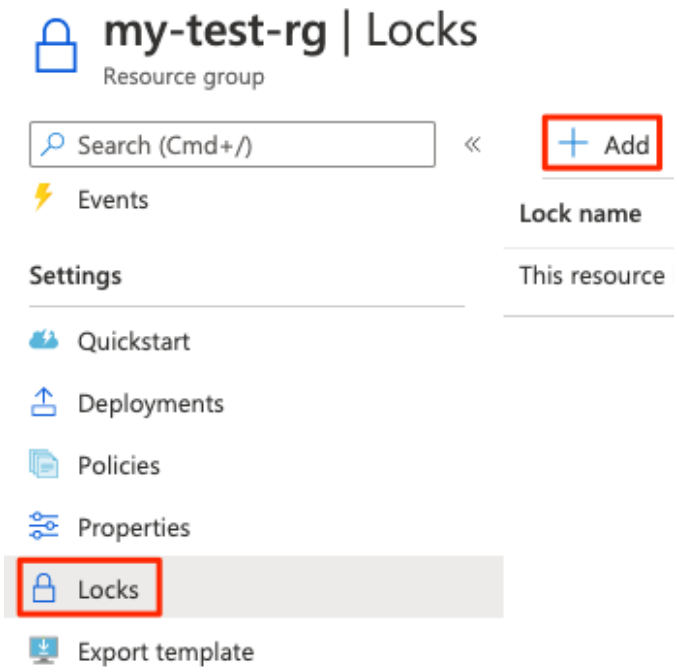


Resource Locks

> ***A resource lock prevents resources from being accidentally deleted or changed.***

Even with Azure role-based access control (Azure RBAC) policies in place, there's still a risk that people with the right level of access could delete critical cloud resources. Think of a resource lock as a warning system that reminds you that a resource should not be deleted or changed.

You can manage resource locks from the Azure portal, PowerShell, the Azure CLI, or from an Azure Resource Manager template. To view, add, or delete locks in the Azure portal, go to the **Settings** section of any resource's **Settings** pane in the Azure portal.



You can apply locks to a subscription, a resource group, or an individual resource. You can set the lock level to **CanNotDelete** or **ReadOnly**.

> **CanNotDelete** means authorized people can still read and modify a resource, but they can't delete the resource without first removing the lock.
>
> **ReadOnly** means authorized people can read a resource, but they can't delete or change the resource. Applying this lock is like restricting all authorized users to the permissions granted by the **Reader** role in Azure RBAC.

Azure Blueprints

What if a cloud administrator accidentally deletes a resource lock? If the resource lock is removed, its associated resources can be changed or deleted. To make the protection process more robust, you can combine resource locks with Azure Blueprints.

> ***Azure Blueprints enables you to define the set of standard Azure resources that your organization requires. For example, you can define a blueprint that specifies that a certain resource lock must***

> ***exist. Azure Blueprints can automatically replace the resource lock if that lock is removed.***

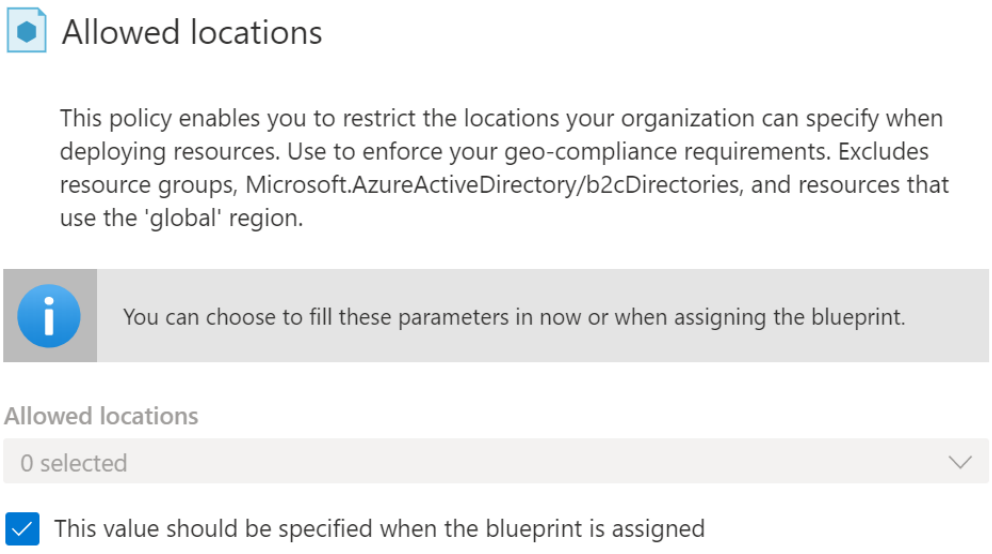
Azure Blueprints orchestrates the deployment of various resource templates and other artifacts, such as:

- Role assignments
- Policy assignments
- Azure Resource Manager templates
- Resource groups

> Each component in the blueprint definition is known as an ***artifact***.

It is possible for artifacts to have no additional parameters (configurations). An example is the **Deploy threat detection on SQL servers** policy, which requires no additional configuration.

Artifacts can also contain one or more parameters that you can configure. The following screenshot shows the **Allowed locations** policy. This policy includes a parameter that specifies the allowed locations.



Resource Tags

As your cloud usage grows, it's increasingly important to stay organized. A good organization strategy helps you understand your cloud usage and can help you manage costs. One way to organize related resources is to place them in their own subscriptions. You can also use resource groups to manage related resources.

> ***Resource tags are a way to organize resources. Tags provide extra information, or metadata, about your resources.***

This metadata is useful for:

- **Resource management** Tags enable you to locate and act on resources that are associated with specific workloads, environments, business units, and owners.
- **Cost management and optimization** Tags enable you to group resources so that you can report on costs, allocate internal cost centers, track budgets, and forecast estimated cost.
- **Operations management** Tags enable you to group resources according to how critical their availability is to your business. This grouping helps you formulate service-level agreements (SLAs). An SLA is an uptime or performance guarantee between you and your users.
- **Security** Tags enable you to classify data by its security level, such as *public* or *confidential*.
- **Governance and regulatory compliance** Tags enable you to identify resources that align with governance or regulatory compliance requirements, such as ISO 27001. Tags can also be part of your standards enforcement efforts. For example, you might require that all resources be tagged with an owner or department name.
- **Workload optimization and automation** Tags can help you visualize all of the resources that participate in complex deployments. For example, you might tag a resource with its associated workload or application name and use software such as Azure DevOps to perform automated tasks on those resources.

You can add, modify, or delete resource tags through PowerShell, the Azure CLI, Azure Resource Manager templates, the REST API, or the Azure portal. You can also manage tags by using **Azure Policy**. For example, you can apply tags to a resource group, but those tags aren't automatically applied to the resources within that resource group. You can use Azure Policy to ensure that a resource inherits the same tags as its parent resource group.

Azure Policy

> ***Azure Policy is a service in Azure that enables you to create, assign, and manage policies that control or audit your resources. These policies enforce different rules across all of your resource configurations so that those configurations stay compliant with corporate standards.***

Azure Policy enables you to define both individual policies and ***groups* of related policies**, known as ***initiatives***. Azure Policy evaluates your resources and highlights resources that aren't compliant with the policies you've created. **Azure Policy can also prevent noncompliant resources from being created**.

Azure Policy comes with built-in policy and initiative definitions for Storage, Networking, Compute, Security Center, and Monitoring. For example, if you define a policy that allows only a certain SKU (stock-keeping unit) size for the virtual machines (VMs) to be used in your environment, that policy is invoked when you create a new VM and whenever you resize existing VMs. Azure Policy also evaluates and monitors all current VMs in your environment.

Implementing a policy in Azure Policy involves three tasks:

1. **Create a policy definition.**

   A policy definition expresses what to evaluate and what action to take. For example, you could prevent VMs from being deployed in certain Azure regions. You also could audit your storage accounts to verify that they only accept connections from allowed networks.

   Every policy definition has conditions under which it's enforced. A policy definition also has an accompanying effect that takes place when the conditions are met.

2. **Assign the definition to resources.**

   To implement your policy definitions, you assign definitions to resources. A *policy assignment*
   is a policy definition that takes place within a specific scope. This scope could be a management group (a collection of multiple subscriptions), a single subscription, or a resource group.

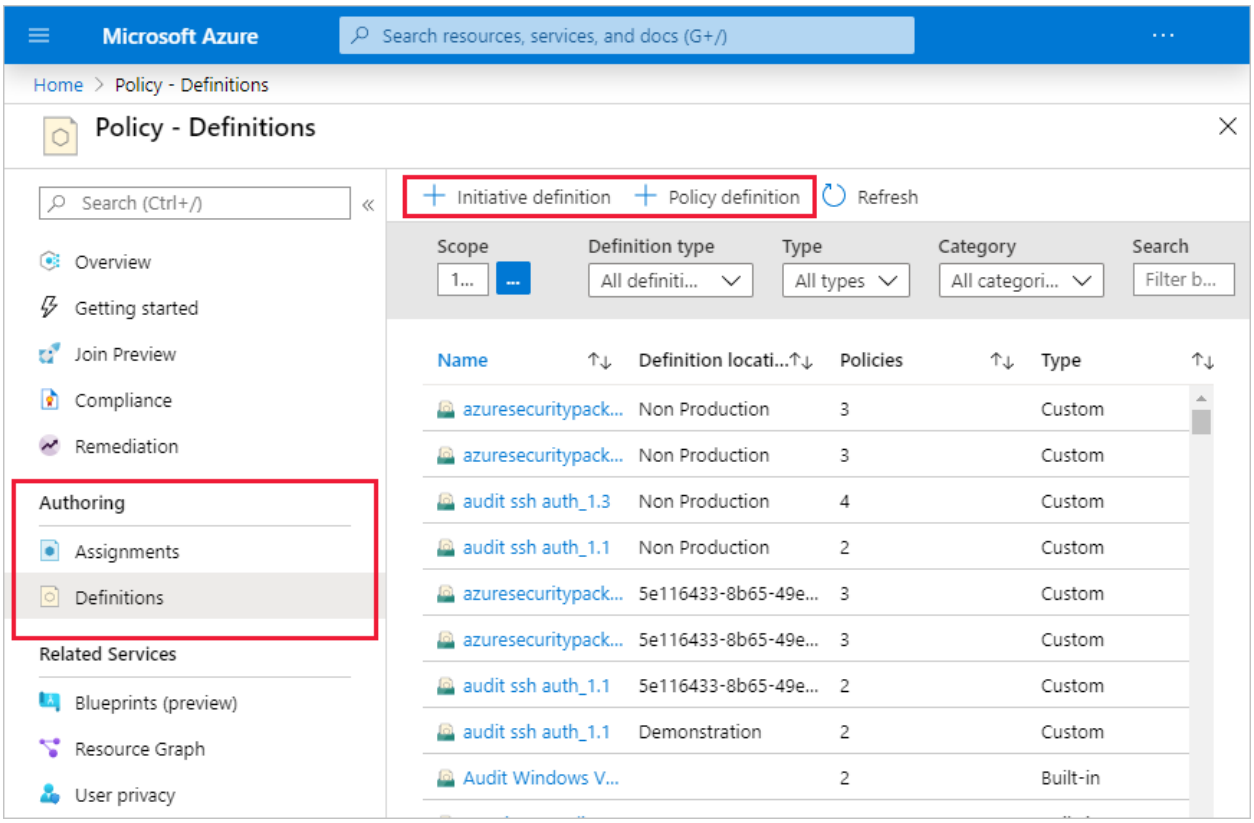
3. **Review the evaluation results.**

When a condition is evaluated against your existing resources, each resource is marked as compliant or noncompliant. You can review the noncompliant policy results and take any action that's needed.

Policy evaluation happens about once per hour. If you make changes to your policy definition and create a policy assignment, that policy is evaluated over your resources within the hour.

> ***An Azure Policy initiative is a way of grouping related policies together. The initiative definition contains all of the policy definitions to help track your compliance state for a larger goal.***

For example, Azure Policy includes an initiative named **Enable Monitoring in Azure Security Center**. Its goal is to monitor all of the available security recommendations for all Azure resource types in Azure Security Center.

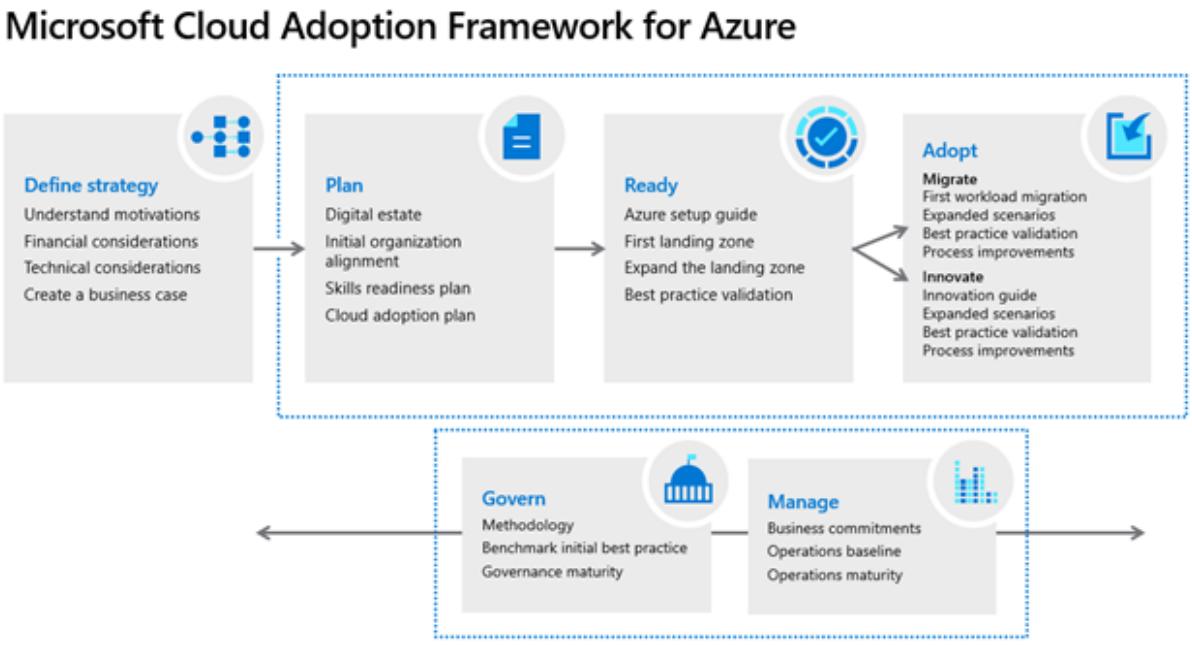


Cloud Adoption Framework

> ***The Cloud Adoption Framework helps you create and implement the business and technology strategies needed to succeed in the cloud. Cloud Adoption Framework consists of tools, documentation, and proven practices.***

The Cloud Adoption Framework includes these stages:

1. Define your strategy.
2. Make a plan.
3. Ready your organization.
4. Adopt the cloud.
5. Govern and manage your cloud environments.



▼ Privacy, Compliance and data protection standards (Deleted)

In general, *compliance* means to adhere to a law, standard, or set of guidelines.

> ***Regulatory compliance refers to the discipline and process of ensuring that a company follows the laws that governing bodies enforce.***

Although there are many more, the following image shows some of the more popular compliance offerings that are available on Azure. These offerings are grouped under four categories: **Global, US Government, Industry, and Regional**.

Global	ISO 27001:2013	ISO 22301:2012	SOC 1 Type 2	CSA STAR Certification
	ISO 27017:2015	ISO 9001:2015	SOC 2 Type 2	CSA STAR Attestation
	ISO 27018:2014	ISO 20000-1:2011	SOC 3	CSA STAR Self-Assessment
				WCAG 2.0 (ISO 40500:2012)
US Gov	FedRAMP High	DFARS	DoE 10 CFR Part 810	FIPS 140-2
	FedRAMP Moderate	DoD DISA SRG Level 5	NIST SP 800-171	ITAR
	EAR	DoD DISA SRG Level 4	NIST CSF	CJIS
		DoD DISA SRG Level 2	Section 508 VPATs	IRS 1075
Industry	PCI DSS Level 1	FCA (UK)	21 CFR Part 11 (GxP)	CDSA
	GLBA	MAS + ABS (Singapore)	MARS-E	MPAA
	FFIEC	23 NYCRR 500	NHS IG Toolkit (UK)	DPP (UK)
	Shared Assessments	HIPAA BAA	NEN 7510:2011 (Netherlands)	FACT (UK)
	FISC (Japan)	HITRUST	FERPA	SOX
	APRA (Australia)			
Regional	Argentina PDPA	China TRUCS / CCCPPF	Germany IT-Grundschutz	Singapore MTCS Level 3
	Australia IRAP Unclassified	EN 301 549	India MeitY	Spain ENS
	Australia IRAP PROTECTED	EU ENISA IAF	Japan CS Mark Gold	Spain DPA
	Canada Privacy Laws	EU Model Clauses	Japan My Number Act	UK Cyber Essentials Plus
	China GB 18030:2005	EU – US Privacy Shield	Netherlands BIR 2012	UK G-Cloud
	China DJCP (MLPS) Level 3	Germany C5	New Zealand Gov CC	UK PASF

The **Microsoft Privacy Statement** explains what personal data Microsoft collects, how Microsoft uses it, and for what purposes.

The privacy statement covers all of Microsoft's services, websites, apps, software, servers, and devices. This list ranges from enterprise and server products to devices that you use in your home to software that students use at school. **The Microsoft Privacy Statement provides information that's relevant to specific services, including Cortana.**

The **Online Services Terms** (OST) is a legal agreement between Microsoft and the customer. The OST details the obligations by both parties with respect to the processing and security of customer data and personal data.

The **Data Protection Addendum** (DPA) further defines the data processing and security terms for online services. These terms include:

- Compliance with laws.
- Disclosure of processed data.
- Data Security, which includes security practices and policies, data encryption, data access, customer responsibilities, and compliance with auditing.
- Data transfer, retention, and deletion.

The **Trust Center** showcases Microsoft's principles for maintaining data integrity in the cloud and how Microsoft implements and supports security, privacy, compliance, and transparency in all Microsoft cloud products and services.

The Trust Center is an important part of the Microsoft Trusted Cloud Initiative and provides support and resources for the legal and compliance community. **The Trust Center is a great resource for people in your organization who might play a role in security, privacy, and compliance.**

> 💡 The **Azure compliance documentation** provides you with detailed documentation about legal and regulatory standards and compliance on Azure. **The compliance documentation provides reference blueprints, or policy definitions, for common standards that you can apply to your Azure subscription.**

Here you find compliance offerings across these categories:

- Global
- US government
- Financial services
- Health
- Media and manufacturing
- Regional

There are also additional compliance resources, such as audit reports, privacy information, compliance implementations and mappings, and white papers and analyst reports. Country and region privacy and compliance guidelines are also included. Some resources might require you to be signed in to your cloud service to access them.

> 💡 **Azure Government** is a separate instance of the Microsoft Azure service. It addresses the security and compliance needs of **US federal agencies, state and local governments**, and their solution providers. Azure Government offers physical isolation from non-US government deployments and provides screened US personnel.

PART 6: Describe Azure cost management and service level agreements

▼ Service Level Agreement (Deleted)

> ***Service-level agreement (SLA) is a formal agreement between a service company and the customer. For Azure, this agreement defines the performance standards that Microsoft commits to for you, the customer.***

A typical SLA breaks down into these sections:

- **Introduction**

  This section explains what to expect in the SLA, including its scope and how subscription renewals can affect the terms.

- **General terms**

  This section contains terms that are used throughout the SLA so that both parties (you and Microsoft) have a consistent vocabulary. For example, this section might define what's meant by downtime, incidents, and error codes.

  This section also defines the general terms of the agreement, including how to submit a claim, receive credit for any performance or availability issues, and limitations of the agreement.

- **SLA details**

  This section defines the specific guarantees for the service. Performance commitments are commonly measured as a percentage. That percentage typically ranges from 99.9 percent ("three nines") to 99.99 percent ("four nines").

  The primary performance commitment typically focuses on *uptime*, or the percentage of time that a product or service is successfully operational. Some SLAs focus on other factors as well, including *latency*, or how fast the service must respond to a request.

> ***Downtime refers to the time duration that the service is unavailable.***

SLA percentage	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

SLA percentage	Downtime per week	Downtime per month	Downtime per year
99.99	1.01 minutes	4.32 minutes	52.56 minutes
99.999	6 seconds	25.9 seconds	5.26 minutes

> ***A service credit is the percentage of the fees you paid that are credited back to you according to the claim approval process.***

An SLA describes how Microsoft responds when an Azure service fails to perform to its specification. For example, you might receive a discount on your Azure bill as compensation when a service fails to perform according to its SLA.

Monthly uptime percentage	Service credit percentage
< 99.99	10
< 99	25
< 95	100

Azure status provides a global view of the health of Azure services and regions. If you suspect there's an outage, this is often a good place to start your investigation. Azure status provides an RSS feed of changes to the health of Azure services that you can subscribe to. You can connect this feed to communication software such as Microsoft Teams or Slack. From the Azure status page, you can also access Azure Service Health. This provides a personalized view of the health of the Azure services and regions that you're using, directly from the Azure portal.

> ***A workload is a distinct capability or task that's logically separated from other tasks, in terms of business logic and data storage requirements. Each workload defines a set of requirements for availability, scalability, data consistency, and disaster recovery.***

On Azure, suppose an application requires:

- Two virtual machines.
- One instance of Azure SQL Database.
- One instance of Azure Load Balancer.

Service	SLA
Azure Virtual Machines	99.9 percent
Azure SQL Database	99.99 percent
Azure Load Balancer	99.99 percent

> 💡 **Therefore, for the Special Orders application, the composite SLA would be:**
> **99.9%×99.9%×99.99%×99.99%=0.999×0.999×0.9999×0.9999=0.9978=99.78%**

Recall that you need two virtual machines. Therefore, you include the Virtual Machines SLA of 99.9 percent two times in the formula. You see here that the composite SLA of 99.78 percent doesn't meet the required SLA of 99.9 percent. You might go back to your team and ask whether this is acceptable. Or you might implement some other strategies into your design to improve this SLA.

Access Preview and Preview Features

> ***The service lifecycle defines how every Azure service is released for public use.***

Every Azure service starts in the development phase. In this phase, the Azure team collects and defines its requirements, and begins to build the service.

Next, the service is released to the public preview phase. During this phase, the public can access and experiment with it and provide real-world feedback. Your feedback helps Microsoft improve services. More importantly, providing feedback gives you the opportunity to request new or different capabilities so that services better meet your needs.

After a new Azure service has been validated and tested, it's released to all customers as a production-ready service. This is known as *general availability* (GA).

Manage Azure Costs

> ***The TCO Calculator helps you estimate the cost savings of operating your solution on Azure over time compared to operating in your on-premises datacenter.***

The term ***total cost of ownership*** is used commonly in finance. It can be hard to see all the hidden costs related to operating a technology capability on-premises. Software licenses and hardware are additional costs.

Working with the TCO Calculator involves three steps:

1. Define your workloads
2. Adjust assumptions
3. View the report

Step 1: Define your workloads

First, you'll enter the specifications of your on-premises infrastructure into the TCO Calculator, based on these four categories:

- **Servers**

  This category includes operating systems, virtualization methods, CPU cores, and memory (RAM).

- **Databases**

  This category includes database types, server hardware, and the Azure service you want to use, which includes the expected maximum concurrent user sign-ins.

- **Storage**

  This category includes storage type and capacity, which includes any backup or archive storage.

- **Networking**

  This category includes the amount of network bandwidth you currently consume in your on-premises environment.

Step 2: Adjust assumptions

Next, you'll specify whether your current on-premises licenses are enrolled for Software Assurance, which can save you money by reusing those licenses on Azure. You'll also specify whether you need to replicate your storage to another Azure region for greater redundancy.

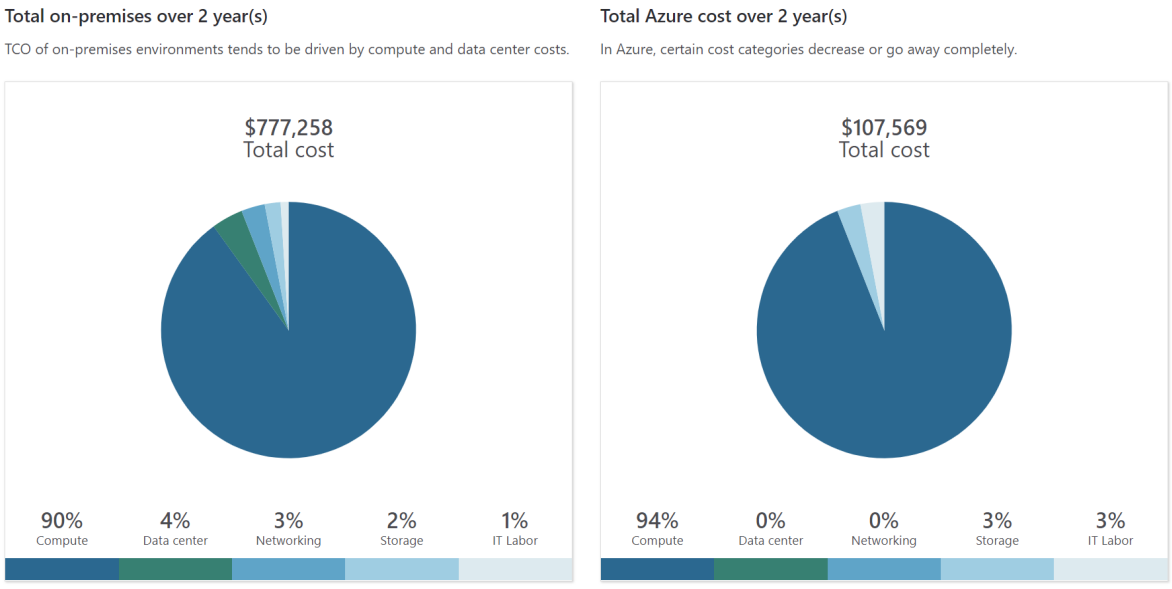
Then, you can see the key operating cost assumptions across several different areas, which will vary among teams and organizations. These costs have been certified by Nucleus Research, an independent research company. For example, these costs include:

- Electricity price per kilowatt hour (KWh)
- Hourly pay rate for IT administration
- Network maintenance cost as a percentage of network hardware and software costs

To improve the accuracy of the TCO Calculator results, you can adjust the values so that they match the costs of your current on-premises infrastructure.

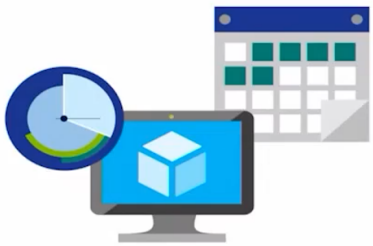
Step 3: View the report

Choose a timeframe between one and five years. the TCO Calculator generates a report that's based on the information you've entered. Here's an example:



You probably know that an Azure *subscription* provides you with access to Azure resources such as virtual machines (VMs), storage, and databases. The types of resources you use affect your monthly bill.

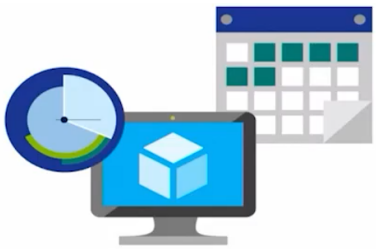
Factors affecting costs (part 1)



There are six primary factors affecting costs:

1) Resource Type	2) Services	3) Location
Costs are resource-specific, so the usage that a meter tracks and the number of meters associated with a resource, depend on the resource type.	Azure usage rates and billing periods can differ between Enterprise, Web Direct, and CSP customers.	The Azure infrastructure is globally distributed, and usage costs might vary between locations that offer Azure products, services, and resources.

Factors affecting costs (part 2)



There are six primary factors affecting costs:

4) Bandwidth	5) Reserved Instances	6) Azure Hybrid Use Benefit
Some inbound data transfers are free, such as data going into Azure datacenters. For outbound data transfers, such as data going out of Azure datacenters, pricing is based on Zones.	With Azure Reservations, you commit to buying one-year or three-year plans for multiple products. Reservations can significantly reduce your resource costs up to 72% on pay-as-you-go prices.	For customers with Software Assurance, Azure Hybrid Benefit allows you to use your on-premises licenses on Azure at a reduced cost.

Purchase Azure Services

Azure offers both free and paid subscription options to fit your needs and requirements. They are:

- **Free trial**

  A free trial subscription provides you with 12 months of popular free services, a credit to explore any Azure service for 30 days, and more than 25 services that are always free. Your Azure services are disabled when the trial ends or when your credit expires for paid products unless you upgrade to a paid subscription.

- **Pay-as-you-go**

A pay-as-you-go subscription lets you pay for what you use by attaching a credit or debit card to your account. Organizations can apply for volume discounts and prepaid invoicing.

- **Member offers**

  Your existing membership to certain Microsoft products and services might provide you with credits for your Azure account, and reduced rates on Azure services. For example, member offers are available to Visual Studio subscribers, Microsoft Partner Network members, Microsoft for Startups members, and Microsoft Imagine members.

There are three main ways to purchase services on Azure. They are:

- **Through an Enterprise Agreement**

  Larger customers, known as enterprise customers, can sign an Enterprise Agreement with Microsoft. This agreement commits them to spend a predetermined amount on Azure services over a period of three years. The service fee is typically paid annually. As an Enterprise Agreement customer, you'll receive the best-customized pricing based on the kinds and amounts of services you plan on using.

- **Directly from the web**

  Here, you can purchase Azure services directly from the Azure portal website and pay standard prices. You're billed monthly, either as a credit card payment or through an invoice. This purchasing method is known as Web Direct.

- **Through a Cloud Solution Provider**

  A Cloud Solution Provider (CSP) is a Microsoft Partner that helps you build solutions on top of Azure. Your CSP bills you for your Azure usage at a price they determine. They also answer your support questions and escalate them to Microsoft, as needed.

Minimise and Manage Total Cost

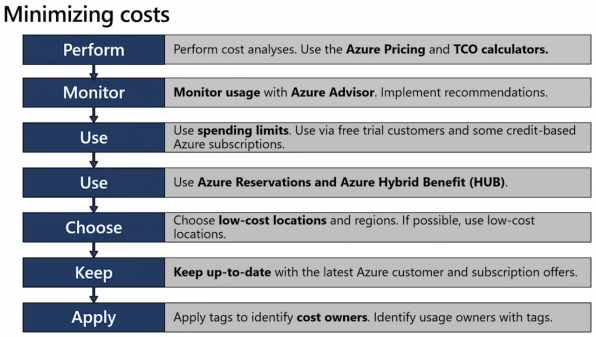
To start, use the Total Cost of Ownership Calculator to estimate the cost savings of operating its solution on Azure instead of in its on-premises data center.

From there, use the Pricing calculator to get a more detailed estimate for running a typical workload on Azure each month.

A checklist of cost-saving measures that can help keep down costs. This list includes:

- Perform cost analysis before you
  deploy

- Use Azure Advisor to monitor your usage
- Use spending limits to prevent accidental spending
- Use Azure Reservations to prepay
- Choose low-cost locations and regions
- Research available cost-saving offers
- Apply tags to identify cost owners



https://miro.com/app/board/uXjVO0R490U=/?share_link_id=423210652868