**fChapter #1 :**

**Understand cloud concepts (15-20%)**

**Describe the benefits and considerations of using cloud services**

• understand terms such as high availability, scalability, elasticity, agility, fault tolerance, and disaster recovery

• understand the principles of economies of scale

• understand the differences between Capital Expenditure (CapEx) and Operational Expenditure (OpEx) • understand the consumption-based model

# What is cloud computing?

Cloud computing is renting resources, like storage space or CPU cycles, on another company's computers. You only pay for what you use. The company providing these services is referred to as a cloud provider. Some example providers are Microsoft, Amazon, and Google.

The cloud provider is responsible for the physical hardware required to execute your work, and for keeping it up-to-date. The computing services offered tend to vary by cloud provider. However, typically they include:

* **Compute power** - such as Linux servers or web applications
* **Storage** - such as files and databases
* **Networking** - such as secure connections between the cloud provider and your company
* **Analytics** - such as visualizing telemetry and performance data

# Cloud computing services

The goal of cloud computing is to make running a business easier and more efficient, whether it's a small start-up or a large enterprise. Every business is unique and has different needs. To meet those needs, cloud computing providers offer a wide range of services.

You need to have a basic understanding of some of the services it provides. Let's briefly discuss the two most common services that all cloud providers offer – compute power and storage.

# Compute power

When you send an email, book a reservation on the Internet, pay a bill online, or even take this Microsoft Learn module you're interacting with cloud-based servers that are processing each request and returning a response. As a consumer, we're all dependent on the computing services provided by the various cloud providers that make up the Internet.

When you build solutions using cloud computing, you can choose how you want work to be done based on your resources and needs. For example, if you want to have more control and responsibility over maintenance, you could create a *virtual machine* (VM). A VM is an emulation of a computer - just like your desktop or laptop you're using now. Each VM includes an operating system and hardware that appears to the user like a physical computer running Windows or Linux. You can then install whatever software you need to do the tasks you want to run in the cloud.

The difference is that you don't have to buy any of the hardware or install the OS. The cloud provider runs your virtual machine on a physical server in one of their datacenters - often sharing that server with other VMs (isolated and secure). With the cloud, you can have a VM ready to go in minutes at less cost than a physical computer.

**VMs aren't the only computing choice - there are two other popular options: containers and serverless computing.**

# What are containers?

**Containers** provide a consistent, isolated execution environment for applications. They're similar to VMs except they don't require a guest operating system. Instead, the application and all its dependencies is packaged into a "container" and then a standard runtime environment is used to execute the app. This allows the container to start up in just a few seconds, because there's no OS to boot and initialize. You only need the app to launch.

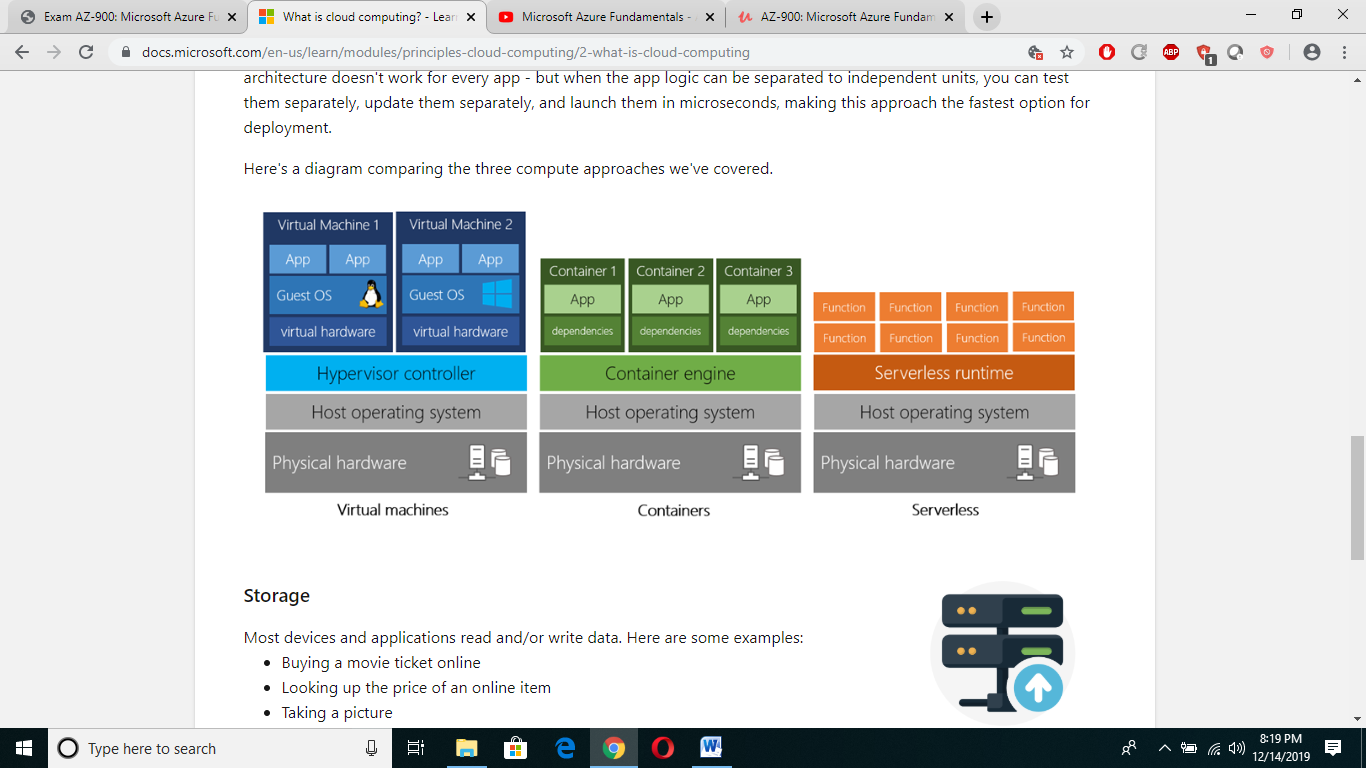
**The open-source project, Docker,** is one of the leading platforms for managing containers. Docker containers provide an efficient, lightweight approach to application deployment because they allow different components of the application to be deployed independently into different containers. Multiple containers can be run on a single machine, and containers can be moved between machines. The portability of the container makes it easy for applications to be deployed in multiple environments, either on-premises or in the cloud, often with no changes to the application.

# What is serverless computing?

**Serverless computing** lets you run application code without creating, configuring, or maintaining a server. The core idea is that your application is broken into separate functions that run when triggered by some action. This is ideal for automated tasks - for example, you can build a serverless process that automatically sends an email confirmation after a customer makes an online purchase.

The serverless model differs from VMs and containers in that you only pay for the processing time used by each function as it executes. VMs and containers are charged while they're running - even if the applications on them are idle. This architecture doesn't work for every app - but when the app logic can be separated to independent units, you can test them separately, update them separately, and launch them in microseconds, making this approach the fastest option for deployment.

Here's a diagram comparing the three compute approaches we've covered.



# Storage

Most devices and applications read and/or write data. Here are some examples:

* Buying a movie ticket online
* Looking up the price of an online item
* Taking a picture
* Sending an email
* Leaving a voicemail

In all of these cases, data is either *read* (looking up a price) or *written* (taking a picture). The type of data and how it's stored can be different in each of these cases.

Cloud providers typically offer services that can handle all of these types of data. For example, if you wanted to store text or a movie clip, you could use a file on disk. If you had a set of relationships such as an address book, you could take a more structured approach like using a database.

The advantage to using cloud-based data storage is you can scale to meet your needs. If you find that you need more space to store your movie clips, you can pay a little more and add to your available space. In some cases, the storage can even expand and contract automatically - so you pay for exactly what you need at any given point in time.

# Benefits of cloud computing

# It's cost-effective

Cloud computing provides a **pay-as-you-go** or **consumption-based** pricing model.

This consumption-based model brings with it many benefits, including:

* No upfront infrastructure costs
* No need to purchase and manage costly infrastructure that you may not use to its fullest
* The ability to pay for additional resources only when they are needed
* The ability to stop paying for resources that are no longer needed

# It's scalable

You can increase or decrease the resources and services used based on the demand or workload at any given time. Cloud computing supports both *vertical* and *horizontal* scaling depending on your needs.

**Vertical scaling**, also known as **"scaling up",** is the process of adding resources to increase the power of an existing server. Some examples of vertical scaling are: adding more CPUs, or adding more memory. **Ex : Adding more CPU, RAMs ,  increasing or decreasing virtual machine (VM) sizes**

**Horizontal scaling**, also known as **"scaling out",** is the process of adding more servers that function together as one unit. For example, you have more than one server processing incoming requests.

**Ex : Adding More VMs.**

Scaling can be done manually or automatically based on specific triggers such as CPU utilization or the number of requests and resources that can be allocated or de-allocated in minutes.

# It's elastic

As your workload changes due to a spike or drop in demand, a cloud computing system can compensate by automatically adding or removing resources.

## It's current

When you use the cloud, you're able to focus on what matters: building and deploying applications. Cloud usage eliminates the burdens of maintaining software patches, hardware setup, upgrades, and other IT management tasks. All of this is automatically done for you to ensure you're using the latest and greatest tools to run your business.

## It's reliable

When you're running a business, you want to be confident your data is always going to be there. Cloud computing providers offer data backup, disaster recovery, and data replication services to make sure your data is always safe. In addition, redundancy is often built into cloud services architecture so if one component fails, a backup component takes its place. This is referred to as *fault tolerance* and it ensures that your customers aren't impacted when a disaster occurs.

## It's global

Cloud providers have fully redundant datacenters located in various regions all over the globe. This gives you a local presence close to your customers to give them the best response time possible no matter where in the world they are.

## It's secure

Think about how you secure your datacenter. You have *physical security* – who can access the building, who can operate the server racks, and so on. You also have *digital security* – who can connect to your systems and data over the network.

## Economies of scale

**Economies of scale** is the ability to do things more efficiently or at a lower-cost per unit when operating at a larger scale. This cost advantage is an important benefit in cloud computing.

Cloud providers such as Microsoft, Google, and Amazon are large businesses leveraging the benefits of economies of scale. These providers can then pass the savings on to their customers.

## Capital expenditure (CapEx) versus operational expenditure (OpEx)

In the past, companies needed to acquire physical premises and infrastructure to start their business. There was a substantial up-front cost in hardware and infrastructure to start or grow a business. Cloud computing provides services to customers without significant upfront costs or equipment setup time.

These two approaches to investment are referred to as:

* **Capital Expenditure (CapEx):** CapEx is the spending of money on physical infrastructure up front, and then deducting that expense from your tax bill over time. CapEx is an upfront cost, which has a value that reduces over time.
* **Operational Expenditure (OpEx)**: OpEx is spending money on services or products now and being billed for them now. You can deduct this expense from your tax bill in the same year. There's no upfront cost. You pay for a service or product as you use it.

**CapEx computing costs**

A typical on-premises datacenter includes costs such as:

**Server costs**

This area includes all hardware components and the cost of supporting them. When purchasing servers, make sure to design fault tolerance and redundancy, such as server clustering, redundant power supplies, and uninterruptible power supplies. When a server needs to be replaced or added to a datacenter, you need to pay for the computer. This can affect your immediate cash flow because you must pay for the server up front.

**Storage costs**

This area includes all storage hardware components and the cost of supporting it. Based on the application and level of fault tolerance, centralized storage can be expensive. For larger organizations, you can create tiers of storage where more expensive fault‐tolerant storage is used for critical applications and lower expense storage is used for lower priority data.

**Network costs**

Networking costs include all on-premises hardware components, including cabling, switches, access points, and routers. This also includes wide area network (WAN) and Internet connections.

**Backup and archive costs**

This is the cost to back up, copy, or archive data. Options might include setting up a backup to or from the cloud. There's an upfront cost for the hardware and additional costs for backup maintenance and consumables like tapes.

**Organization continuity and disaster recovery costs**

Along with server fault tolerance and redundancy, you need to plan for how to recover from a disaster and continue operating. Your plan should consist of creating a data recovery site. It could also include backup generators. Most of these are upfront costs, especially if you build a data recovery site, but there's an additional ongoing cost for the infrastructure and its maintenance.

**Datacenter infrastructure costs**

These are costs for electricity, floor space, cooling, and building maintenance.

**Technical personnel**

While not a capital expenditure, the personnel required to work on your infrastructure are specific to on-premises datacenters. You will need the technical expertise and workforce to install, deploy, and manage the systems in the datacenter and at the data recovery site.

**OpEx cloud computing costs**

With cloud computing, many of the costs associated with an on-premises datacenter are shifted to the service provider. Instead of thinking about physical hardware and datacenter costs, cloud computing has a different set of costs. For accounting purposes, all these costs are operational expenses:

**Leasing software and customized features**

Using a pay-per-use model requires actively managing your subscriptions to ensure users do not misuse the services, and that provisioned accounts are being utilized and not wasted.

**Scaling charges based on usage/demand instead of fixed hardware or capacity.**

Cloud computing can bill in various ways, such as the number of users or CPU usage time. However, billing categories can also include allocated RAM, I/O operations per second (IOPS), and storage space. Plan for backup traffic and data recovery traffic to determine the bandwidth needed.

**Billing at the user or organization level.**

**The subscription (pay-per-use) model is a computing billing method** that is designed for both organizations and users. The organization or user is billed for the services used, typically on a recurring basis. You can scale, customize, and provision computing resources, including software, storage, and development platforms. For example, when using a dedicated cloud service, you could pay based on server hardware and usage.

**Benefits of CapEx**With capital expenditures, you plan your expenses at the start of a project or budget period. Your costs are fixed, meaning you know exactly how much is being spent. This is appealing when you need to predict the expenses before a project starts due to a limited budget.

**Benefits of OpEx**

Demand and growth can be unpredictable and can outpace expectation, which is a challenge for the CapEx model as shown in the following graph.

Microsoft Azure, like many other cloud service providers, have built their billing and subscription models on a pay-per-use basis, i.e., if you use one hour of CPU time you pay for the hour you have consumed. No more, no less. This basic economic principle forms the foundation of most public cloud services.

**Cloud deployment models**

There are three different cloud deployment models. A cloud deployment model defines where your data is stored and how your customers interact with it – how do they get to it, and where do the applications run? It also depends on how much of your own infrastructure you want or need to manage.

**The three deployment methods of cloud computing**

## Public Cloud

## Private Cloud

## Hybrid Cloud

**Public cloud**

This is the most common deployment model. In this case, you have no local hardware to manage or keep up-to-date – everything runs on your cloud provider's hardware. In some cases, you can save additional costs by sharing computing resources with other cloud users.

#### Advantages

* High scalability/agility – you don't have to buy a new server in order to scale
* **Pay-as-you-go pricing** – you pay only for what you use, no CapEx costs
* You're not responsible for maintenance or updates of the hardware
* Minimal technical knowledge to set up and use - you can leverage the skills and expertise of the cloud provider to ensure workloads are secure, safe, and highly available

Ex : Deploying Web App, Blog and then focus on maintaining the site without having to worry about purchasing, managing or maintaining the hardware on which it runs.

#### Disadvantages

* There may be specific security requirements that cannot be met by using public cloud
* There may be government policies, industry standards, or legal requirements which public clouds cannot meet
* You don't own the hardware or services and cannot manage them as you may want to
* Unique business requirements, such as having to maintain a legacy application might be hard to meet

**Private cloud**

In a private cloud, you create a cloud environment in your own datacenter and provide self-service access to compute resources to users in your organization. This offers a simulation of a public cloud to your users, but you remain completely responsible for the purchase and maintenance of the hardware and software services you provide.  
  
**Advantages**

This approach has several advantages:

* You can ensure the configuration can support any scenario or legacy application
* You have control (and responsibility) over security
* Private clouds can meet strict security, compliance, or legal requirements

#### Disadvantages

Some reasons teams move away from the private cloud are:

* You have some initial CapEx costs and must purchase the hardware for startup and maintenance
* Owning the equipment limits the agility - to scale you must buy, install, and setup new hardware
* Private clouds require IT skills and expertise that's hard to come by

A use case scenario for a private cloud would be when an organization has data that cannot be put in the public cloud, perhaps for legal reasons. An example scenario may be where government policy requires specific data to be kept in-country or privately.

A private cloud can provide cloud functionality to external customers as well, or to specific internal departments such as Accounting or Human Resources.

### Hybrid cloud

A hybrid cloud combines public and private clouds, allowing you to run your applications in the most appropriate location. For example, you could host a website in the public cloud and link it to a highly secure database hosted in your private cloud (or on-premises datacenter).

This is helpful when you have some things that cannot be put in the cloud, maybe for legal reasons. For example, you may have some specific pieces of data that cannot be exposed publicly (such as medical data) which needs to be held in your private datacenter. Another example is one or more applications that run on old hardware that can't be updated. In this case, you can keep the old system running locally, and connect it to the public cloud for authorization or storage.

#### Advantages

Some advantages of a hybrid cloud are:

* You can keep any systems running and accessible that use out-of-date hardware or an out-of-date operating system
* You have flexibility with what you run locally versus in the cloud
* You can take advantage of economies of scale from public cloud providers for services and resources where it's cheaper, and then supplement with your own equipment when it's not
* You can use your own equipment to meet security, compliance, or legacy scenarios where you need to completely control the environment

#### Disadvantages

Some concerns you'll need to watch out for are:

* It can be more expensive than selecting one deployment model since it involves some CapEx cost up front
* It can be more complicated to set up and manage

## Describe the differences between Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS)

### Infrastructure as a service (IaaS)

Infrastructure as a Service is the most flexible category of cloud services. It aims to give you complete control over the hardware that runs your application (IT infrastructure servers and virtual machines (VMs), storage, networks, and operating systems). Instead of buying hardware, with IaaS, you rent it. It's an instant computing infrastructure, provisioned and managed over the internet.

**Vendor : VM, Hardware, Storage, Virulization**

**You : Application, Data, Middleware, Runtime.**

IaaS is commonly used in the following scenarios:

* **Migrating workloads.** Typically, IaaS facilities are managed in a similar way as on-premises infrastructure and provide an easy migration path for moving existing applications to the cloud.
* **Test and development.** Teams can quickly set up and dismantle test and development environments, bringing new applications to market faster. IaaS makes scaling development and testing environments, fast and economical.
* **Storage, backup, and recovery.** Organizations avoid the capital outlay and complexity of storage management, which typically requires skilled staff to manage data and meet legal and compliance requirements. IaaS is useful for managing unpredictable demand and steadily growing storage needs. IaaS can also simplify the planning and management of backup and recovery systems.

## Platform as a service (PaaS)

PaaS provides an environment for building, testing, and deploying software applications. The goal of PaaS is to help you create an application quickly without managing the underlying infrastructure. For example, when deploying a web application using PaaS, you don't have to install an operating system, web server, or even system updates.

**Vendor : VM, Hardware, Storage, Virulization, Middleware, Runtime**

**You : Application, Data,.**

PaaS is commonly used in the following scenarios

* **Development framework.** PaaS provides a framework that developers can build upon to develop or customize cloud-based applications. Just like Microsoft Excel macro, PaaS lets developers create applications using built-in software components.
* **Analytics or business intelligence.** Tools provided as a service with PaaS allow organizations to analyze and mine their data. They can find insights and patterns, and predict outcomes to improve business decisions such as forecasting, product design, and investment returns.

## Software as a service (SaaS)

SaaS is software that is centrally hosted and managed for the end customer. It is usually based on an architecture where one version of the application is used for all customers, and licensed through a monthly or annual subscription. Office 365, Skype, and Dynamics CRM Online are perfect examples of SaaS software.

**Vendor : VM, Hardware, Storage, Virulization, Middleware, Runtime, Application, Data**

**You : Deploy the application,.**

## Cost and Ownership

|  | **IaaS** | **PaaS** | **SaaS** |
| --- | --- | --- | --- |
| Upfront costs | There are no upfront costs. Users pay only for what they consume. | There are no upfront costs. Users pay only for what they consume. | Users have no upfront costs; they pay a subscription, typically on a monthly or annual basis. |
| User ownership | The user is responsible for the purchase, installation, configuration, and management of their own software, operating systems, middleware, and applications. | The user is responsible for the development of their own applications. However, they are not responsible for managing the server or infrastructure. This allows the user to focus on the application or workload they want to run. | Users just use the application software; they are not responsible for any maintenance or management of that software. |
| Cloud provider ownership | The cloud provider is responsible for ensuring that the underlying cloud infrastructure (such as virtual machines, storage, and networking) is available for the user. | The cloud provider is responsible for operating system management, network, and service configuration. Cloud providers are typically responsible for everything apart from the application that a user wants to run. They provide a complete managed platform on which to run the application. | The cloud provider is responsible for the provision, management, and maintenance of the application software. |

## Management responsibilities



## Combine cloud services to fit your needs

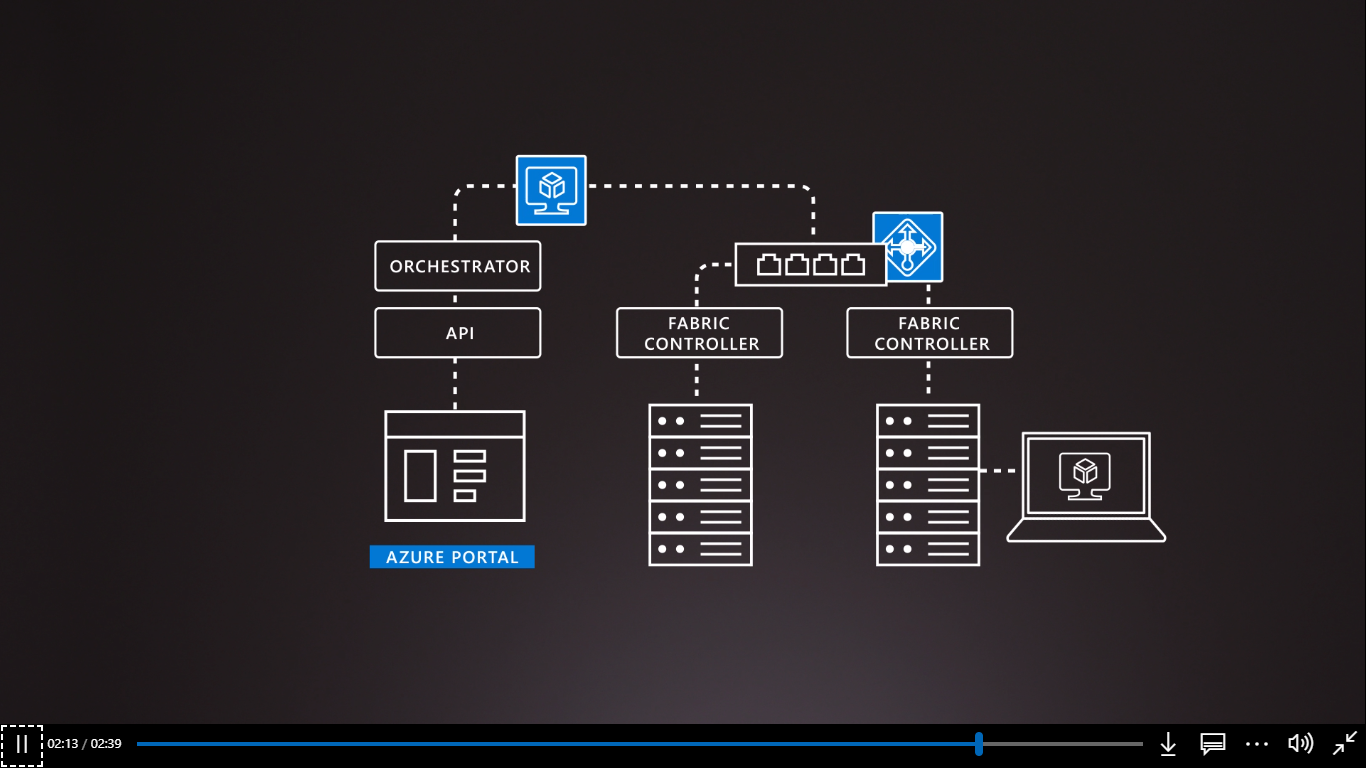
IaaS, PaaS, and SaaS each contain different levels of managed services. You may easily use a combination of these types of infrastructure. You could use Office 365 on your company's computers (SaaS), and in Azure, you could host your VMs (IaaS) and use Azure SQL Database (PaaS) to store your data. With the cloud's flexibility, you can use any combination that provides you with the maximum result.

## Chapter #2 :

## Understand core Azure services (30-35%)

How Azure works :

1. It uses Virtulization to decouple CPU, RAM & OS from a computer using with the abstraction technology called HYPERVISOR.
2. Hypervisor further uses to add multiple VMs, add more resources etc. HYPERVISOR allow VMS to run on any OS to make it platform independent.



**What is cloud computing?**

Cloud computing is the delivery of computing services over the Internet using a **pay-as-you-go** pricing model. Put another way; it's a way to rent compute power and storage from someone else's data center.

Instead of maintaining CPUs and storage in your data center, you rent them for the time that you need them. The cloud provider takes care of maintaining the underlying infrastructure for you.

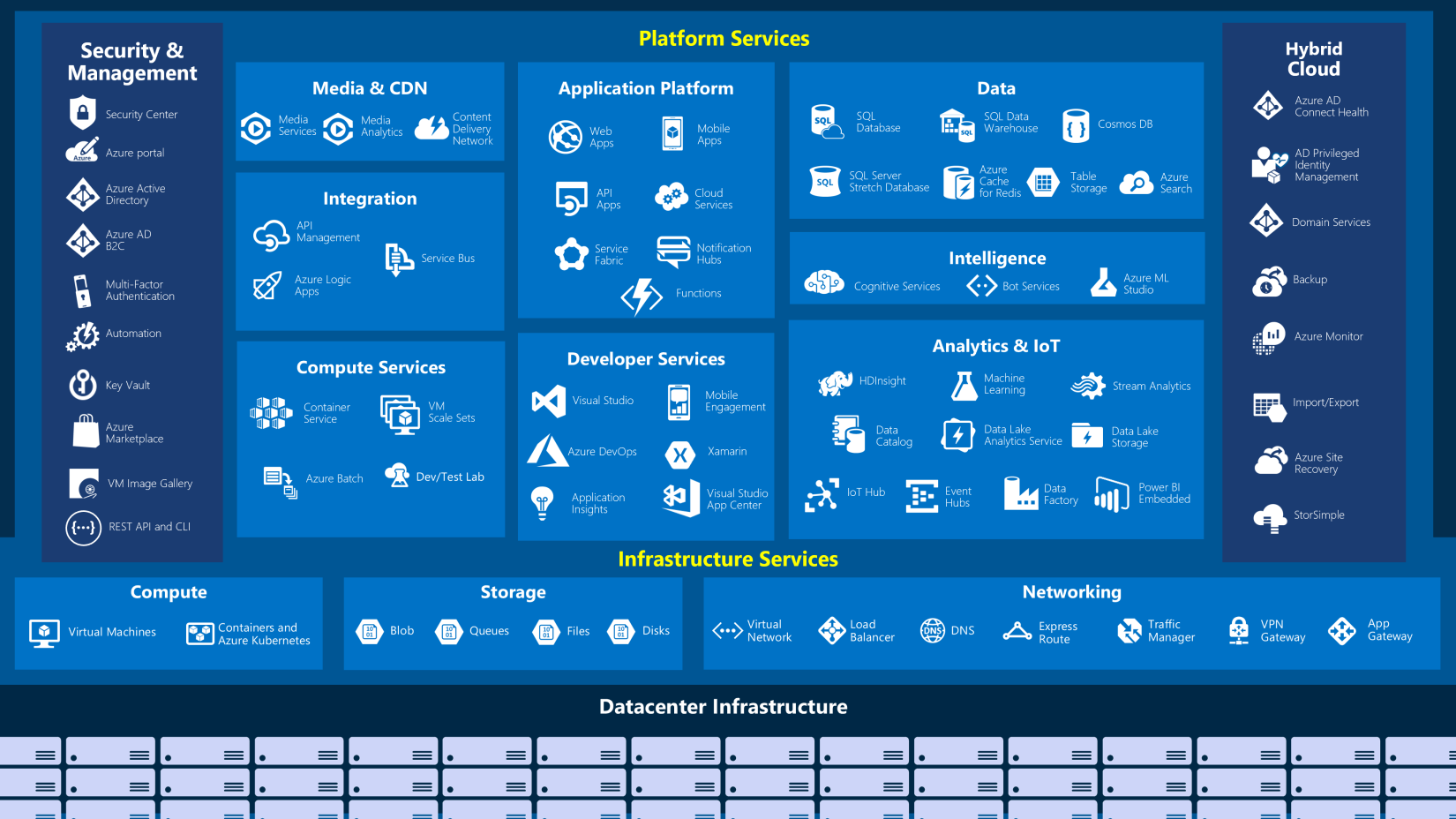
**What can I do on Azure?**

Azure provides over 100 services that enable you to do everything from running your existing applications on virtual machines to exploring new software paradigms such as intelligent bots and mixed reality.

For example, Azure provides AI and machine-learning services that can naturally communicate with your users through vision, hearing, and speech. It also provides storage solutions that dynamically grow to accommodate massive amounts of data. Azure services enable solutions that are not feasible without the power of the cloud.

**Azure services**

Here's a big-picture view of the available services and features in Azure.

[](https://docs.microsoft.com/en-us/learn/modules/welcome-to-azure/media/3-azure-services.png#lightbox)

Let's take a closer look at the most commonly-used categories:

* Compute
* Networking
* Storage
* Mobile
* Databases
* Web
* Internet of Things
* Big Data
* Artificial Intelligence
* DevOps

**Compute**

Compute services are often one of the primary reasons why companies move to the Azure platform. Azure provides a range of options for hosting applications and services. Here are some examples of compute services in Azure:

| **Service name** | **Service function** |
| --- | --- |
| Azure Virtual Machines | Windows or Linux virtual machines (VMs) hosted in Azure |
| Azure Virtual Machine Scale Sets | Scaling for Windows or Linux VMs hosted in Azure |
| Azure Kubernetes Service | Enables management of a cluster of VMs that run containerized services |
| Azure Service Fabric | Distributed systems platform. Runs in Azure or on-premises |
| Azure Batch | Managed service for parallel and high-performance computing applications |
| Azure Container Instances | Run containerized apps on Azure without provisioning servers or VMs |
| Azure Functions | An event-driven, serverless compute service |

**Networking**

Linking compute resources and providing access to applications is the key function of Azure networking. Networking functionality in Azure includes a range of options to connect the outside world to services and features in the global Microsoft Azure datacenters.

Azure networking facilities have the following features:

| **Service name** | **Service function** |
| --- | --- |
| 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 |
| 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 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), connecting local and remote sites |

**Storage**

Azure provides four main types of storage services. These services are:

| **Service name** | **Service function** |
| --- | --- |
| Azure Blob storage | Storage service for very large objects, such as video files or bitmaps |
| Azure File storage | File shares that you can access and manage like a file server |
| Azure Queue storage | A data store for queuing and reliably delivering messages between applications |
| Azure Table storage | A NoSQL store that hosts unstructured data independent of any schema |

These services all share several common characteristics:

* **Durable** and highly available with redundancy and replication.
* **Secure** through automatic encryption and role-based access control.
* **Scalable** with virtually unlimited storage.
* **Managed**, handling maintenance and any critical problems for you.
* **Accessible** from anywhere in the world over HTTP or HTTPS.

**Mobile**

Azure enables developers to create mobile backend services for iOS, Android, and Windows apps quickly and easily. Features that used to take time and increase project risks, such as adding corporate sign-in and then connecting to on-premises resources such as SAP, Oracle, SQL Server, and SharePoint, are now simple to include.

Other features of this service include:

* Offline data synchronization.
* Connectivity to on-premises data.
* Broadcasting push notifications.
* Autoscaling to match business needs.

**Databases**

Azure provides multiple database services to store a wide variety of data types and volumes. And with global connectivity, this data is available to users instantly.

| **Service name** | **Service function** |
| --- | --- |
| Azure Cosmos DB | Globally distributed database that supports NoSQL options |
| Azure SQL Database | Fully managed relational database with auto-scale, integral intelligence, and robust security |
| Azure Database for MySQL | Fully managed and scalable MySQL relational database with high availability and security |
| Azure Database for PostgreSQL | Fully managed and scalable PostgreSQL relational database with high availability and security |
| SQL Server on VMs | Host enterprise SQL Server apps in the cloud |
| Azure SQL Data Warehouse | Fully managed data warehouse with integral security at every level of scale at no extra cost |
| Azure Database Migration Service | Migrates your databases to the cloud with no application code changes |
| Azure Cache for Redis | Caches frequently used and static data to reduce data and application latency |
| Azure Database for MariaDB | Fully managed and scalable MariaDB relational database with high availability and security |

**Web**

Having a great web experience is critical in today's business world. Azure includes first-class support to build and host web apps and HTTP-based web services. The Azure services focused on web hosting include:

| **Service Name** | **Description** |
| --- | --- |
| Azure App Service | Quickly create powerful cloud web-based apps |
| Azure Notification Hubs | Send push notifications to any platform from any back end. |
| Azure API Management | Publish APIs to developers, partners, and employees securely and at scale. |
| Azure Cognitive Search | Fully managed search as a service. |
| Web Apps feature of Azure App Service | Create and deploy mission-critical web apps at scale. |
| Azure SignalR Service | Add real-time web functionalities easily. |

**Internet of Things**

People are able to access more information than ever before. It began with personal digital assistants (PDAs), then morphed into smartphones. Now there are smart watches, smart thermostats, even smart refrigerators. Personal computers used to be the norm. Now the internet allows any item that's online-capable to access valuable information. This ability for devices to garner and then relay information for data analysis is referred to as the Internet of Things (IoT).

There are a number of services that can assist and drive end-to-end solutions for IoT on Azure.

| **Service Name** | **Description** |
| --- | --- |
| IoT Central | Fully-managed global IoT software as a service (SaaS) solution that makes it easy to connect, monitor, and manage your IoT assets at scale |
| Azure IoT Hub | Messaging hub that provides secure communications and monitoring between millions of IoT devices |
| IoT Edge | Push your data analysis models directly onto your IoT devices, allowing them to react quickly to state changes without needing to consult cloud-based AI models. |

**Big Data**

Data comes in all formats and sizes. When we talk about Big Data, we're referring to *large* volumes of data. Data from weather systems, communications systems, genomic research, imaging platforms, and many other scenarios generate hundreds of gigabytes of data. This amount of data makes it hard to analyze and make decisions around. It's often so large that traditional forms of processing and analysis are no longer appropriate.

Open source cluster technologies have been developed to deal with these large data sets. Microsoft Azure supports a broad range of technologies and services to provide big data and analytic solutions.

| **Service Name** | **Description** |
| --- | --- |
| Azure SQL Data Warehouse | Run analytics at a massive scale using a cloud-based Enterprise Data Warehouse (EDW) that leverages massive parallel processing (MPP) to run complex queries quickly across petabytes of data |
| Azure HDInsight | Process massive amounts of data with managed clusters of Hadoop clusters in the cloud |
| Azure Databricks (preview) | Collaborative Apache Spark–based analytics service that can be integrated with other Big Data services in Azure. |

**Artificial Intelligence**

Artificial Intelligence, in the context of cloud computing, is based around a broad range of services, the core of which is Machine Learning. Machine Learning is a data science technique that allows computers to use existing data to forecast future behaviors, outcomes, and trends. Using machine learning, computers learn without being explicitly programmed.

Forecasts or predictions from machine learning can make apps and devices smarter. For example, when you shop online, machine learning helps recommend other products you might like based on what you've purchased. Or when your credit card is swiped, machine learning compares the transaction to a database of transactions and helps detect fraud. And when your robot vacuum cleaner vacuums a room, machine learning helps it decide whether the job is done.

Some of the most common Artificial Intelligence and Machine Learning service types in Azure are:

| **Service Name** | **Description** |
| --- | --- |
| 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 |
| Azure Machine Learning Studio | Collaborative, drag-and-drop visual workspace where you can build, test, and deploy machine learning solutions using pre-built machine learning algorithms and data-handling modules |

A closely related set of products are the *cognitive services*. These are pre-built APIs you can leverage in your applications to solve complex problems.

| **Service Name** | **Description** |
| --- | --- |
| Vision | Image-processing algorithms to smartly identify, caption, index, and moderate your pictures and videos. |
| Speech | Convert spoken audio into text, use voice for verification, or add speaker recognition to your app. |
| Knowledge mapping | Map complex information and data in order to solve tasks such as intelligent recommendations and semantic search. |
| Bing Search | Add Bing Search APIs to your apps and harness the ability to comb billions of webpages, images, videos, and news with a single API call. |
| Natural Language processing | Allow your apps to process natural language with pre-built scripts, evaluate sentiment and learn how to recognize what users want. |

**DevOps**

DevOps (Development and Operations) brings together people, processes, and technology, automating software delivery to provide continuous value to your users. Azure DevOps Services allows you to create *build* and *release* pipelines that provide continuous integration, delivery, and deployment for your applications.

| **Service Name** | **Description** |
| --- | --- |
| Azure DevOps | Azure DevOps Services (formerly known as Visual Studio Team Services, or VSTS), provides development collaboration tools including high-performance pipelines, free private Git repositories, configurable Kanban boards, and extensive automated and cloud-based load testing |
| Azure DevTest Labs | Quickly create on-demand Windows and Linux environments you can use to test or demo your applications directly from your deployment pipelines |

## Next unit: Exercise - Create a website hosted in Azure

# Exercise - Access an App Service using Azure Cloud Shell

* 1. CLI command to get the list of Subscriptions.

az account list --output table

* 1. CLI command to get all the Resource Group created.

az group list --output table

* 1. List all the Resources under a Resource Group.

az resource list \

--resource-group [Resource Group Name] \

--resource-type Microsoft.Web/sites

* 1. How to Stop Resource

az webapp stop \

--resource-group [Resource Group Name] \

--name [Resource Name]

* 1. How to Start Resource   
     az webapp stop \  
      --resource-group [Resource Group Name] \  
      --name [Resource Name]
  2. dsfd

# Core Cloud Services - Azure architecture and service guarantees

# Understand Datacenters and Regions in Azure

## What is a region?

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

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

It also provides better scalability, redundancy, and preserves data residency for your services.

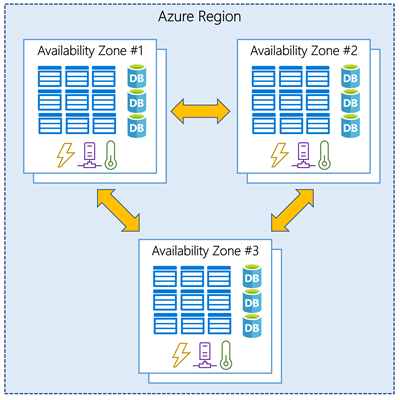
**Understand Availability Zones in Azure**

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

**What is an Availability Zone?**

Availability Zones are physically separate datacenters within an Azure region.

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



**Supported regions**

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

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

**Using Availability Zones in your apps**

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

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

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

# Understand Service-Level Agreements for Azure

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

**SLAs for Azure products and services**

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

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

**Performance Targets -** An SLA defines performance targets for an Azure product or service. The performance targets that an SLA defines are specific to each Azure product and service. For example, performance targets for some Azure services are expressed as uptime guarantees or connectivity rates.  
  
**Uptime and Connectivity Guarantees -** SLA specifies performance-target commitments that range from 99.9 percent ("three nines") to 99.999 percent ("five nines"), for each corresponding Azure product or service.  
"**Monthly Uptime Percentage**" for Virtual Machines in Availability Zones is calculated as Maximum Available Minutes less Downtime divided by Maximum Available Minutes in a billing month for a given Microsoft Azure subscription. Monthly Uptime Percentage is represented by the following formula:

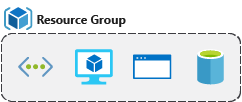
Monthly Uptime % = (Maximum Available Minutes – Downtime) / Maximum Available Minutes X 100  
  
**Service Credits –** Discount on Billing if Microsoft fails to perform to its governing SLA’s specification.

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

Composite SLA = Component 1 SLA \* Component 2 SLA

## describe Resource Groups & Azure Resource Manager

## Resource Group - Resource groups are a fundamental element of the Azure platform. A resource group is a logical container for resources deployed on Azure. These resources are anything you create in an Azure subscription like virtual machines, Application Gateways, and CosmosDB instances.



Resource Group = VM + DB + APP Service, Web App or any combination which is required to build app in azure.

### Life cycle - If you delete a resource group, all resources contained within are also deleted. Organizing resources by life cycle can be useful in non-production environments, where you might try an experiment, but then dispose of it when done

### Authorization - Resource groups are also a scope for applying role-based access control (RBAC) permissions. By applying RBAC permissions to a resource group, you can ease administration and limit access to allow only what is needed.

VNET

<https://github.com/MicrosoftDocs/azure-docs/blob/master/includes/virtual-machines-common-network-overview.md>

**What are tags?**

Tags are name/value pairs of text data that you can apply to resources and resource groups. Tags allow you to associate custom details about your resource, in addition to the standard Azure properties a resource has:

* department (like finance, marketing, and more)
* environment (prod, test, dev),
* cost center
* life cycle and automation (like shutdown and startup of virtual machines).

A resource can have up to 50 tags. The name is limited to 512 characters for all types of resources except storage accounts, which have a limit of 128 characters. The tag value is limited to 256 characters for all types of resources. Tags aren't inherited from parent resources. Not all resource types support tags, and tags can't be applied to classic resources.

**CLI to add tag**

|  |
| --- |
| az resource tag --tags Department=Finance \  --resource-group msftlearn-core-infrastructure-rg \  --name msftlearn-vnet1 \  --resource-type "Microsoft.Network/virtualNetworks" |

## What is Azure Policy?

Azure Policy is a service you can use to create, assign, and manage policies. These policies apply and enforce rules that your resources need to follow. These policies can enforce these rules when resources are created, and can be evaluated against existing resources to give visibility into compliance.

# Secure resources with role-based access control (IAM – Identity & Access Management)

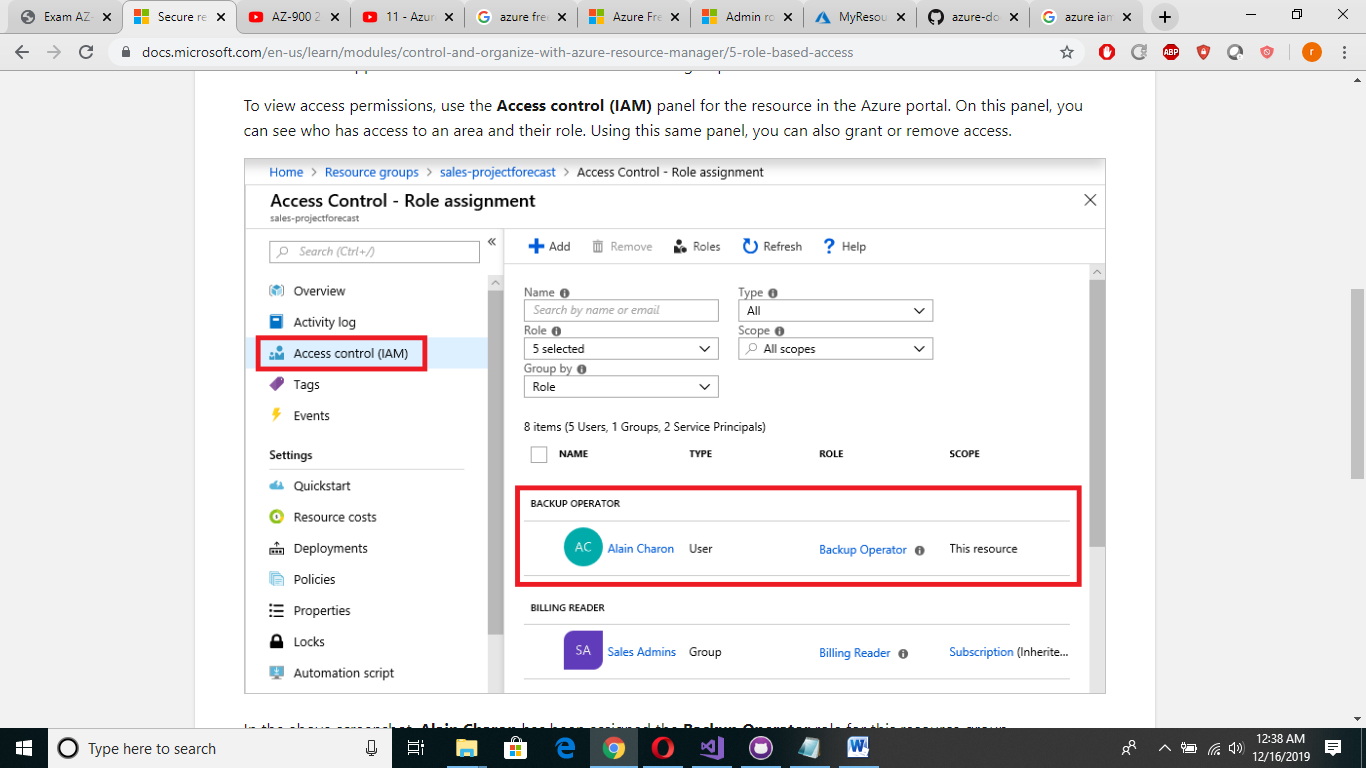
Implementing Azure Policy ensured that all our employees with Azure access are following our internal standards for creating resources, but we have a second issue we need to solve: how do we protect those resources once they are deployed? We have IT personnel that need to manage settings, developers that need to have read-only access, and administrators that need to be able to control them completely. Enter Role-Based Access Control (RBAC).

RBAC provides fine-grained access management for Azure resources, enabling you to grant users the specific rights they need to perform their jobs. RBAC is considered a core service and is included with all subscription levels at no cost.

Using RBAC, you can:

* Allow one user to manage VMs in a subscription, and another user to manage virtual networks.
* Allow a database administrator (DBA) group to manage SQL databases in a subscription.
* Allow a user to manage all resources in a resource group, such as VMs, websites, and virtual subnets.
* Allow an application to access all resources in a resource group.

To view access permissions, use the **Access control (IAM)** panel for the resource in the Azure portal. On this panel, you can see who has access to an area and their role. Using this same panel, you can also grant or remove access.



**Describe some of the core products available in Azure**

## What is Azure compute?

Azure compute is an on-demand computing service for running cloud-based applications. It provides computing resources like multi-core processors and supercomputers via virtual machines and containers. It also provides serverless computing to run apps without requiring infrastructure setup or configuration.

There are four common techniques for performing compute in Azure:

* Virtual machines
* Containers
* Azure App Service
* Serverless computing

## Explore What are virtual machines?

**Virtual machines**, or VMs, are software emulations of physical computers. They include a virtual processor, memory, storage, and networking resources. They host an operating system (OS), and you're able to install and run software just like a physical computer. And by using a remote desktop client, you can use and control the virtual machine as if you were sitting in front of it.

## What are containers?

Containers are a virtualization environment for running applications. Just like virtual machines, containers are run on top of a host operating system. But unlike VMs, containers don't include an operating system for the apps running inside the container. Instead, containers bundle the libraries and components needed to run the application and use the existing host OS running the container. For example, if five containers are running on a server with a specific Linux kernel, all five containers and the apps within them share that same Linux kernel.

## What is Azure App Service?

Azure App Service is a platform-as-a-service (PaaS) offering in Azure that is designed to host enterprise-grade web-oriented applications. You can meet rigorous performance, scalability, security, and compliance requirements while using a fully managed platform to perform infrastructure maintenance.

## What is Serverless Computing?

Serverless computing is a cloud-hosted execution environment that runs your code but completely abstracts the underlying hosting environment. You create an instance of the service, and you add your code; no infrastructure configuration or maintenance is required, or even allowed.

**Scaling VMs in Azure**

You can run single VMs for testing, development, or minor tasks; or you can group VMs together to provide high availability, scalability, and redundancy. Azure has several features such that, no matter what your uptime requirements are, Azure can meet them. These features include:

* Availability sets
* Virtual Machine Scale Sets - Azure Virtual Machine Scale Sets let you create and manage a group of identical, load balanced VMs.
* Azure Batch - Azure Batch enables large-scale job scheduling and compute management with the ability to scale to tens, hundreds, or thousands of VMs.

**What are availability sets?**

An **availability set** is a logical grouping of two or more VMs that help keep your application available during planned or unplanned maintenance.

**Fault Domain -** A fault domain is essentially a rack of servers. It provides the physical separation of your workload across different power, cooling, and network hardware that support the physical servers in the data center server racks. In the event the hardware that supports a server rack becomes unavailable, only that rack of servers is affected by the outage.

# Explore Containers in Azure

# If you wish to run multiple instances of an application on a single host machine, containers are an excellent choice.

**Containers** provide a consistent, isolated execution environment for applications. They're similar to VMs except they don't require a guest operating system. Instead, the application and all its dependencies is packaged into a "container" and then a standard runtime environment is used to execute the app. This allows the container to start up in just a few seconds, because there's no OS to boot and initialize. You only need the app to launch.

A container is a modified runtime environment built on top of a host OS that executes your application. A container doesn't use virtualization, so it doesn't waste resources simulating virtual hardware with a redundant OS. This environment typically makes containers more lightweight than VMs.

**Containers in Azure**

Azure supports Docker containers (a standardized container model), and there are several ways to manage containers in Azure.

* Azure Container Instances (ACI)
* Azure Kubernetes Service (AKS)

### Azure Container Instances

Azure Container Instances (ACI) offers the fastest and simplest way to run a container in Azure. You don't have to manage any virtual machines or configure any additional services. It is a PaaS offering that allows you to upload your containers and execute them directly with automatic elastic scale.

### Azure Kubernetes Service

The task of automating, managing, and interacting with a large number of containers is known as orchestration. Azure Kubernetes Service (AKS) is a complete orchestration service for containers with distributed architectures with multiple containers.

Kubernet – (Container Management Automation) technique to manage Multiple Containers work load

# Explore Azure App Service

Azure App Service enables you to build and host web apps, background jobs, mobile backends, and RESTful APIs in the programming language of your choice without managing infrastructure. It offers automatic scaling and high availability. App Service supports both Windows and Linux

This platform as a service (PaaS) allows you to focus on the website and API logic while Azure handles the infrastructure to run and scale your web applications.

## App Service costs

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 - for example, whether it's dedicated or shared hardware, and how much memory is reserved for it. There is even a free tier you can use to host small, low-traffic sites.

# Explore Serverless computing in Azure

Serverless computing is the abstraction of servers, infrastructure, and OSs. With serverless computing, Azure takes care of managing the server infrastructure and allocation/deallocation of resources based on demand. Infrastructure isn't your responsibility. Scaling and performance are handled automatically, and you are billed only for the exact resources you use. There's no need to even reserve capacity.

Serverless computing encompasses three ideas:

1. the abstraction of servers
2. an event-driven scale
3. and micro-billing:

**Abstraction of servers**: Serverless computing abstracts the servers you run on. You never explicitly reserve server instances; the platform manages that for you. Each function execution can run on a different compute instance, and this execution context is transparent to the code. With serverless architecture, you simply deploy your code, which then runs with high availability.

**Event-driven scale**: Serverless computing is an excellent fit for workloads that respond to incoming events. Events include triggers by timers (for example, if a function needs to run every day at 10:00 AM UTC), HTTP (API and webhook scenarios), queues (for example, with order processing), and much more. 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 and bindings provide a declarative way to connect to services from within the code.

**Micro-billing**: Traditional computing has the notion of per-second billing, but often, that's not as useful as it seems. Even if a customer's website gets only one hit a day, they still pay for a full day's worth of availability. With serverless computing, they pay only for the time their code runs. If no active function executions occur, they're not charged. For example, if the code runs once a day for two minutes, they're charged for one execution and two minutes of computing time.

<https://docs.microsoft.com/en-us/learn/modules/intro-to-security-in-azure/>