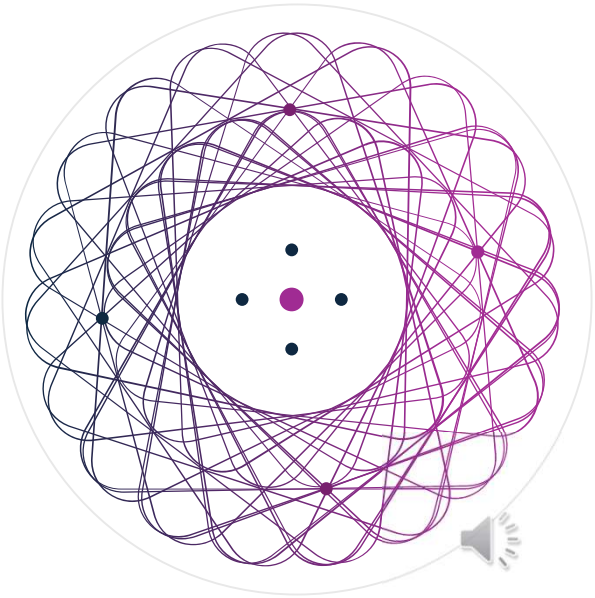
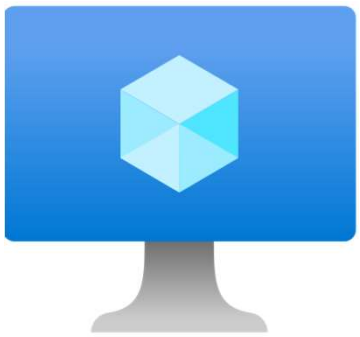
Microsoft Azure

Azure Compute



Azure virtual machines

- Azure **Virtual Machines (VM)** are software emulations of physical computers.
- Includes virtual processor, memory, storage, and networking.
- IaaS offering that provides total control and customization.



https://docs.microsoft.com/learn/modules/describe-azure-compute-networking-services/2-virtual-machines

Development and test – Azure VMs offer a quick and easy way to create a computer with specific configurations required to code and test an application.
Applications in the cloud – Because demand for your application can fluctuate, it might make economic sense to run it on a VM in Azure. You pay for extra VMs when you need them and shut them down when you don't.
Extended datacenter – Virtual machines in an Azure virtual network can easily be connected to your organization's network.

Azure virtual machines - https://azure.microsoft.com/en-us/services/virtual-machines/

Determine Virtual Machine Sizing

Type	Description
General purpose	Balanced CPU-to-memory ratio.
Compute optimized	High CPU-to-memory ratio.
Memory optimized	High memory-to-CPU ratio.
Storage optimized	High disk throughput and I/O.
GPU	Specialized virtual machines targeted for heavy graphic rendering and video editing..
High performance compute	Our fastest and most powerful CPU virtual machines

✔ Share VM images in a compute gallery

Create Virtual Machines in the Portal

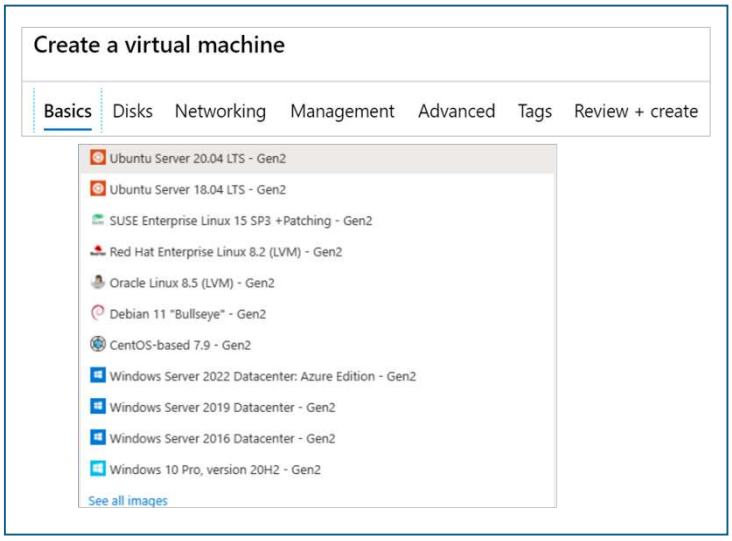
Basic (required) – Project details, Administrator account, Inbound port rules

Disks – OS disk type, data disks

Networking – Virtual networks, load balancing

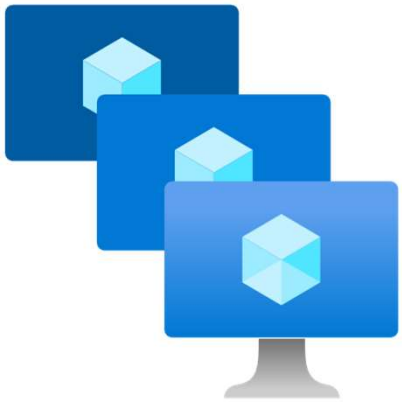
Management – Monitoring, Auto-shutdown, Backup

Advanced – Add additional configuration, agents, scripts or applications



VM scale sets

- Scale sets provide a load-balanced opportunity to automatically scale resources.
- Scale out when resource needs increase.
- Scale in when resource needs are lower.



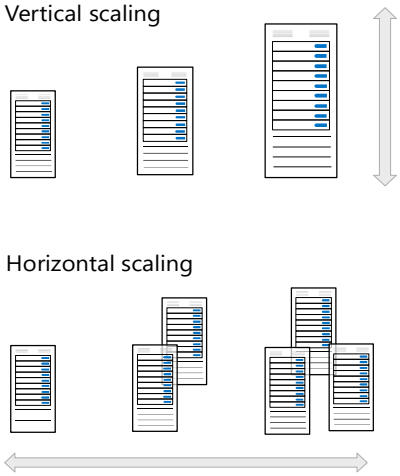
https://docs.microsoft.com/learn/modules/describe-azure-compute-networking-services/2-virtual-machines

Compare Vertical to Horizontal Scaling

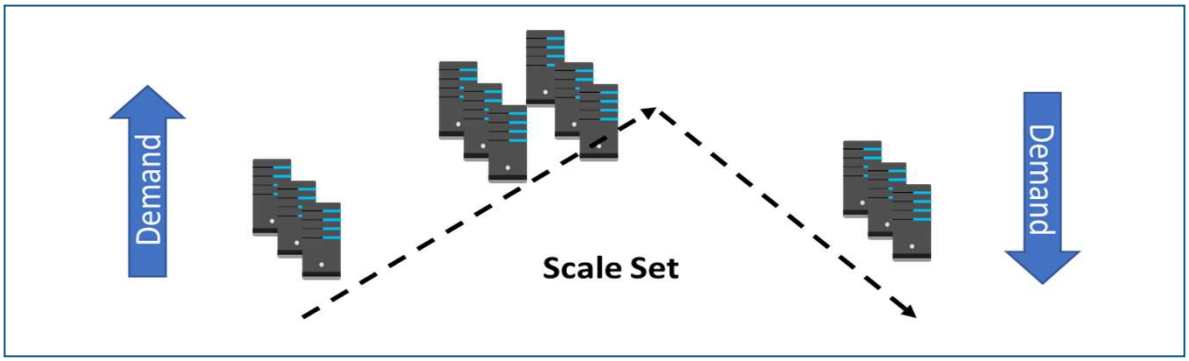
Vertical scaling (scale up and scale down) is the process of increasing or decreasing power to a single instance of a workload; **usually manual**

Horizontal scaling (scale out and scale in) is the process of increasing or decreasing the number of instances of a workload; **frequently automated**

Virtual Machine Scale Sets (VMSS) ONLY deal with horizontal scaling of the virtual machines.



Implement Scale Sets



Scale sets deploy a set of **identical** VMs	No pre-provisioning of VMs is required	As demand goes up VMs are added	As demand goes down VM are removed	The process can be manual, automated, or a combination of both

Create Scale Sets

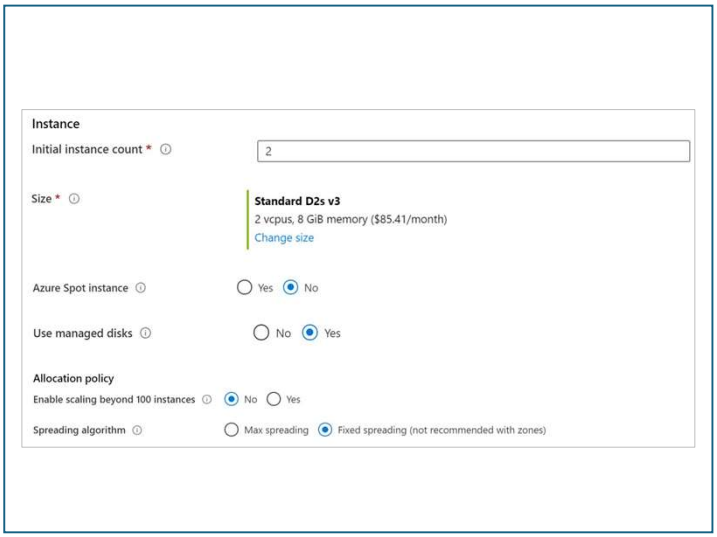
Instance count. Number of VMs in the scale set (0 to 1000)

Instance size. The size of each virtual machine in the scale set

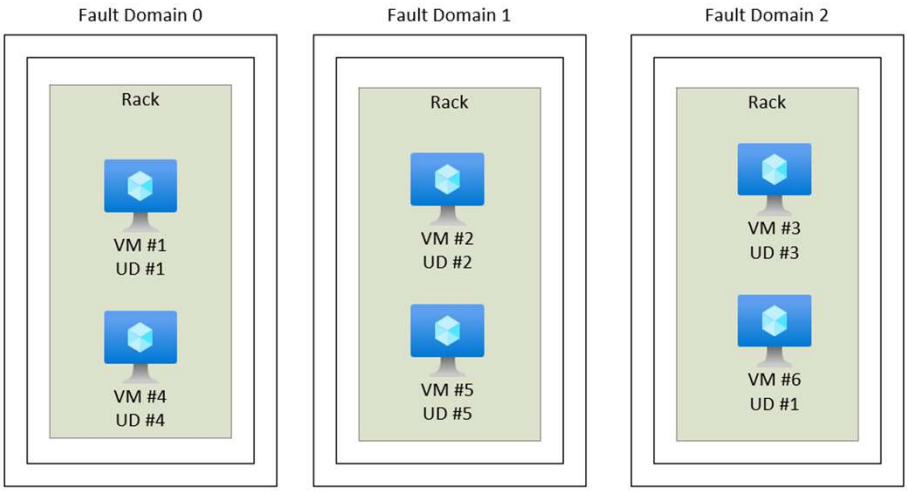
Azure Spot Instance. Unused capacity at a discounted rate

Use managed disks

Enable scaling beyond 100 instances



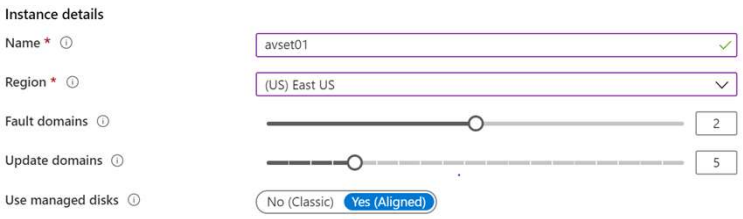
VM availability sets



https://docs.microsoft.com/learn/modules/describe-azure-compute-networking-services/2-virtual-machines

Update domain keeps VMs grouped if they can be rebooted at the same time without causing an outage.
Fault domains group VMs based on common power and networking.

Setup Availability Sets



Two or more instances in Availability Sets = 99.95% SLA

Configure multiple Virtual Machines in an Availability Set

Configure each application tier into separate Availability Sets

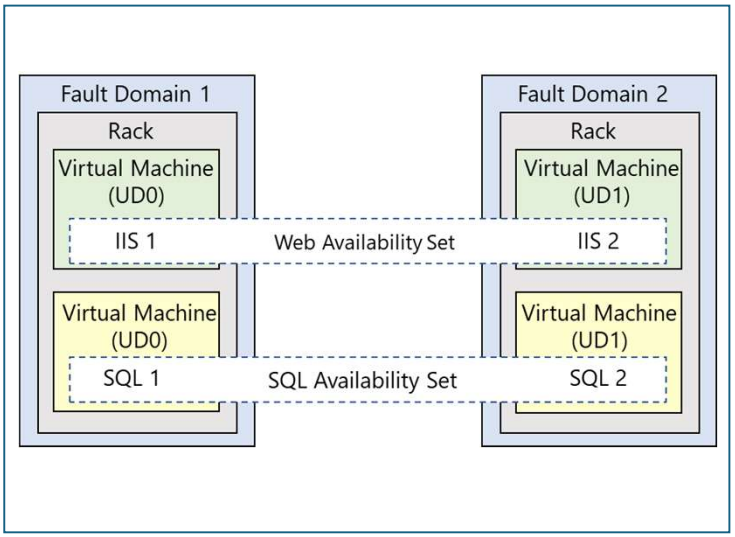
Combine a Load Balancer with Availability Sets

Use managed disks with the Virtual Machines

Review Update and Fault Domains

Update domains allows Azure to perform incremental or rolling upgrades across a deployment. During planned maintenance, only one update domain is rebooted at a time

Fault Domains are a group of Virtual Machines that share a common set of hardware, switches, that share a single point of failure. VMs in an availability set are placed in at least two fault domains



Fault domain
Prevent Hardware failures like limit the impact of potential physical hardware failures, network outages, or power interruptions
1 Rack that share common power source and network switch.
Max- 3 FD per availability set, Default value=2
Update domain
Max= 20 UD, Default=5
Update domains indicate groups of virtual machines and underlying physical hardware that can be rebooted at the same time
The order of update domains being rebooted may not proceed sequentially during planned maintenance, but only one update domain is rebooted at a time. A rebooted update domain is given 30 minutes to recover before maintenance is initiated on a different update domain.

VM Creation - Planning Checklist

Start with the network

Understand the pricing model

Name the VM

Consider storage for the VM

Decide the location for the VM

Select an operating system

Determine the size of the VM

Azure Virtual Desktop

- **Azure Virtual Desktop** is a desktop and app virtualization that runs in the cloud.
- Create a full desktop virtualization environment without having to run additional gateway servers.
- Reduce risk of resource being left behind.
- True multi-session deployments.

https://docs.microsoft.com/learn/modules/describe-azure-compute-networking-services/4-virtual-desktop

https://docs.microsoft.com/en-us/azure/virtual-desktop/overview

Azure Container Services

- Azure **Containers** are a light-weight, virtualized environment that does not require operating system management, and can respond to changes on demand.

Azure Container Instances: a PaaS offering that runs a **single container** in Azure without the need to manage a virtual machine or additional services.

Azure Kubernetes Service: an orchestration service for containers with distributed architectures and large volumes of containers.

https://docs.microsoft.com/learn/modules/describe-azure-compute-networking-services/5-containers

Containers are a virtualization environment. However, unlike virtual machines, you do not manage an operating system. Containers are meant to be lightweight, and are designed to be created, scaled out, and stopped dynamically.

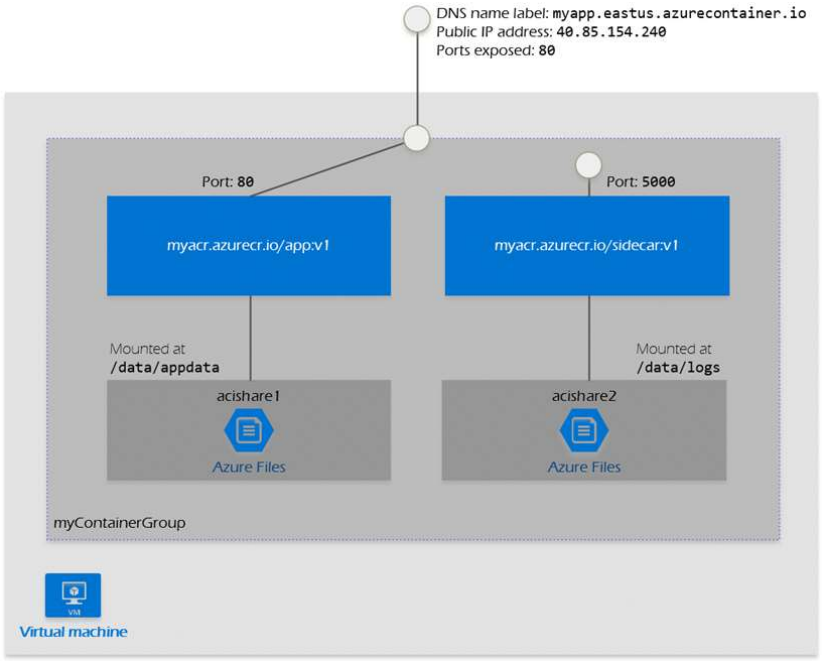
Azure Container Instances - https://azure.microsoft.com/en-us/services/container-instances/
Azure Kubernetes Service -What is Azure Kubernetes Service (AKS)? - Azure Kubernetes Service | Microsoft Learn

ACI

- Azure Container Instances (ACI) offers the fastest and simplest way to run a container in Azure, without having to manage any virtual machines and without having to adopt a higher-level service (e.g., AKS).
- The task of automating and managing many containers and how they interact is known as ***orchestration***. Popular container orchestrators include Kubernetes, and Docker Swarm.
  - Azure Container Instances provides some of the basic scheduling capabilities of orchestration platforms.
- A container group is a collection of containers that get scheduled on the same host machine.
  - The containers in a container group share a lifecycle, resources, local network, and storage volumes.
  - It's similar in concept to a *pod* in Kubernetes.

Serverless containers in Azure - Azure Container Instances | Microsoft Learn

Container Groups in ACI

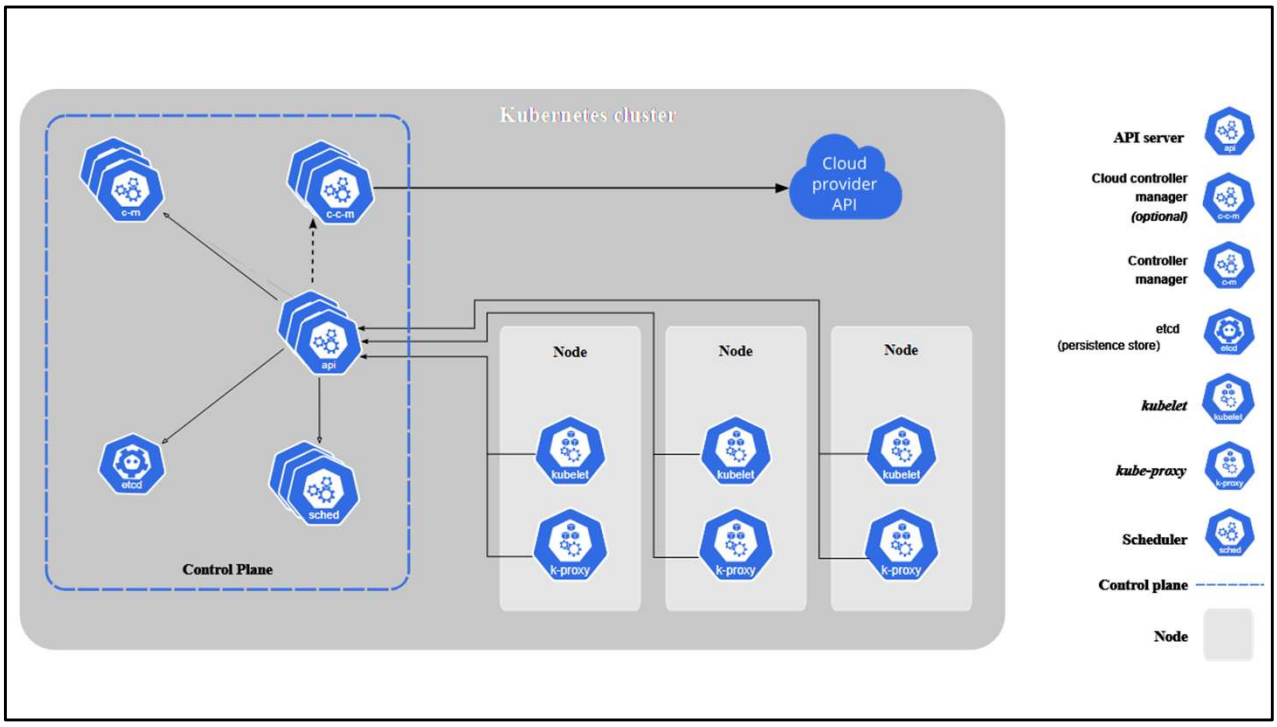


[Introduction to container groups - Azure Container Instances | Microsoft Learn]

Azure Kubernetes Service (AKS)

- Kubernetes is an open-source container orchestration platform for automating the deployment, scaling, and management of containerized applications.
- AKS is a managed Kubernetes service that simplifies deploying, managing, and scaling containerized applications using Kubernetes.
- An AKS cluster is divided into two main components:
  - **Control plane**: The control plane provides the core Kubernetes services and orchestration of application workloads.
  - **Nodes**: Nodes are the underlying virtual machines (VMs) that run your applications.
- Each AKS cluster has at least one node, which is an Azure virtual machine (VM) that runs Kubernetes node components.

What is Azure Kubernetes Service (AKS)? - Azure Kubernetes Service | Microsoft Learn



Azure Kubernetes Services (AKS) core concepts - Azure Kubernetes Service | Microsoft Learn

An AKS cluster is divided into two main components:

•**Control plane**: The control plane provides the core Kubernetes services and orchestration of application workloads.
•**Nodes**: Nodes are the underlying virtual machines (VMs) that run your applications.

AKS Control Plane

The Azure managed control plane is comprised of several components that help manage the cluster:

Component	Description
kube-apiserver	The API server (kube-apiserver) exposes the Kubernetes HTTP API to enable requests to the cluster from inside and outside of the cluster.
etcd	etcd is a highly available key-value store for API Server Data
kube-scheduler	The scheduler (kube-scheduler) helps make scheduling decisions, watching for new pods with no assigned node and selecting a node for them to run on.
kube-controller-manager	The controller manager (kube-controller-manager) runs controller processes, to implement Kubernetes API logic/behavior.
cloud-controller-manager	The cloud controller manager (cloud-controller-manager) embeds cloud-specific control logic to run controllers specific to the cloud provider. Integrates with underlying cloud provider

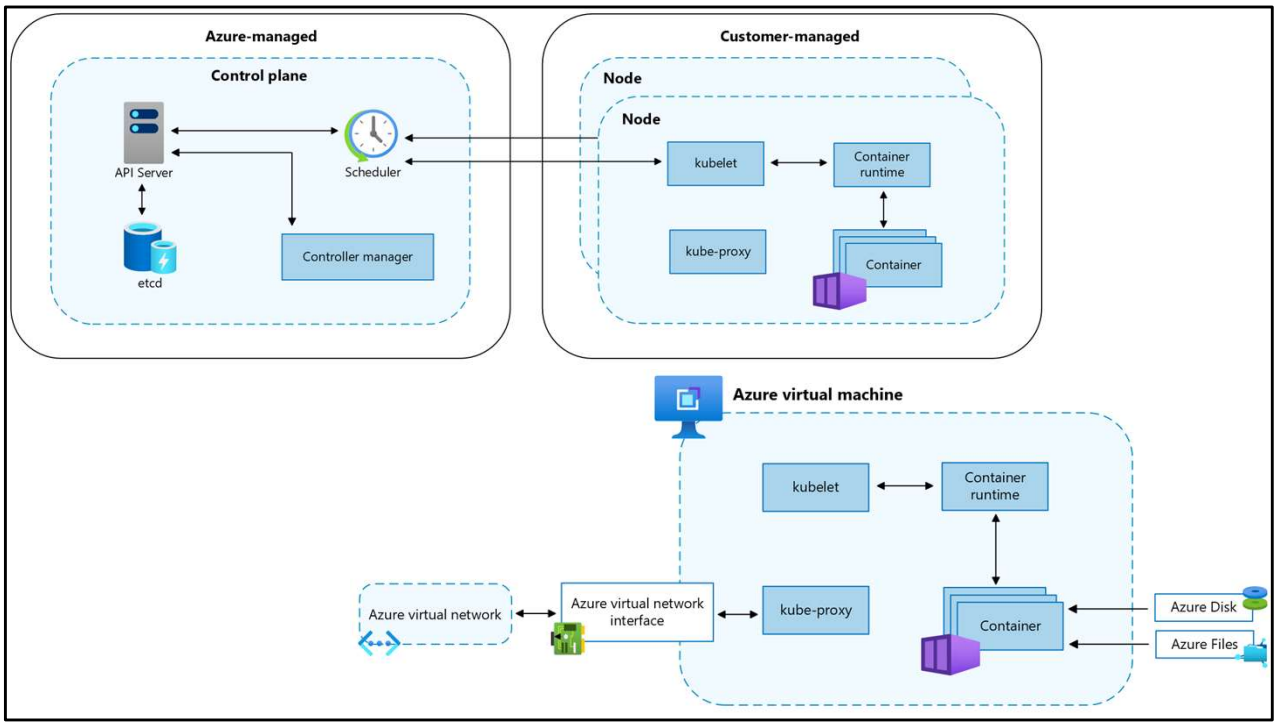
Azure Kubernetes Services (AKS) core concepts - Azure Kubernetes Service | Microsoft Learn

AKS Nodes

- Each AKS cluster has at least one node, which is an Azure virtual machine (VM) that runs Kubernetes node components. The following components run on each node:

Component	Description
kubelet	The kubelet ensures that containers are running in a pod.
kube-proxy	The kube-proxy is a network proxy that maintains network rules on nodes to implement services
container runtime	The container runtime manages the execution and lifecycle of containers. This is the software layer that is responsible for running containers.

Azure Kubernetes Services (AKS) core concepts - Azure Kubernetes Service | Microsoft Learn



Azure Kubernetes Services (AKS) core concepts - Azure Kubernetes Service | Microsoft Learn

Azure Functions

- Azure Functions is an event-driven, serverless compute option that doesn't require maintaining virtual machines or containers.
  - If you build an app using VMs or containers, those resources have to be "running" for your app to function.
  - With Azure Functions, an event wakes the function, alleviating the need to keep resources provisioned when there are no events.
- Functions are commonly used when you need to perform work in response to an event (often via an HTTP request), timer, or message from another Azure service, and when that work can be completed quickly, within seconds or less.

Azure Functions

Event based code running your service and not the underlying infrastructure.

https://docs.microsoft.com/learn/modules/describe-azure-compute-networking-services/6-functions

Serverless computing is the evolution of cloud platforms in the direction of pure cloud native code. Serverless brings developers closer to business logic while insulating them from infrastructure concerns. It's a pattern that doesn't imply "no server" but rather, "less server." Serverless code is event-driven. Code may be triggered by anything from a traditional HTTP web request to a timer or the result of uploading a file. The infrastructure behind serverless allows for instant scale to meet elastic demands and offers micro-billing to truly "pay for what you use." Serverless requires a new way of thinking and approach to building applications and isn't the right solution for every problem.

Azure Functions is code running your service and not the underlying platform or infrastructure. Creates infrastructure based on an event.

Azure Functions - https://docs.microsoft.com/en-us/azure/azure-functions/

Note: For more details about serverless services available with Azure, see https://azure.microsoft.com/en-us/solutions/serverless

Azure Functions – Common Scenarios

- The following are a common, *but by no means exhaustive*, set of scenarios for Azure Functions.

If you want to…	then…
Build a web API	Implement an endpoint for your web applications using the HTTP trigger
Process file uploads	Run code when a file is uploaded or changed in blob storage
Build a serverless workflow	Create an event-driven workflow from a series of functions using durable functions
Respond to database changes	Run custom logic when a document is created or updated in Azure Cosmos DB
Run scheduled tasks	Execute code on pre-defined timed intervals
Create reliable message queue systems	Process message queues using Queue Storage, Service Bus, or Event Hubs
Analyze IoT data streams	Collect and process data from IoT devices
Process data in real time	Use Functions and SignalR to respond to data in the moment

https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview

Comparing Azure compute options

• Virtual machines	• Virtual Desktop	• Containers
Cloud based server that supports either Windows or Linux environments.	• Provides a cloud based personal computer Windows desktop experience.	• Lightweight, miniature environment well suited for running microservices.
Useful for lift-and-shift migrations to the cloud.	• Dedicated applications to connect and use, or accessible from any modern browser.	• Designed for scalability and resiliency through orchestration.
Complete operating system package, including the host operating system.	• Multi-client login allows multiple users to log into the same machine at the same time.	• Applications and services are packaged in a container that sits on-top of the host operating system. Multiple containers can sit on one host OS.

Azure App Services

- Azure **App Services** is a fully managed platform to build, deploy, and scale web apps and APIs quickly.
- Works with .NET, .NET Core, Node.js, Java, Python, or php.
- PaaS offering with enterprise-grade performance, security, and compliance requirements.

https://docs.microsoft.com/learn/modules/describe-azure-compute-networking-services/7-describe-application-hosting-options

Overview - Azure App Service | Microsoft Learn

App Services – Uses & Benefits

- With App Service, you can host most common app service styles like:
  - Web apps
  - RESTful APIs
  - Background jobs
  - Mobile apps
- App Service handles most of the infrastructure decisions you deal with in hosting web-accessible apps:
  - Deployment and management are integrated into the platform.
  - Endpoints can be secured.
  - Sites can be scaled quickly to handle high traffic loads.
  - The built-in load balancing and traffic manager provide high availability.

Web apps
App Service includes full support for hosting web apps by using ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. You can choose either Windows or Linux as the host operating system.

API apps
Much like hosting a website, you can build REST-based web APIs by using your choice of language and framework. You get full Swagger support and the ability to package and publish your API in Azure Marketplace. The produced apps can be consumed from any HTTP- or HTTPS-based client.

WebJobs
You can use the WebJobs feature to 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. WebJobs are often used to run background tasks as part of your application logic.

Mobile apps
Use the Mobile Apps feature of App Service to quickly build a back end for iOS and Android apps. With just a few actions in the Azure portal, you can:
Store mobile app data in a cloud-based SQL database.
Authenticate customers against common social providers, such as MSA, Google, Twitter, and

Facebook.
Send push notifications.
Execute custom back-end logic in C# or Node.js.
On the mobile app side, there's SDK support for native iOS and Android, Xamarin, and React native apps.

Azure App Service (cont ...)

- Azure App Service is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends.
- It supports multiple languages, including .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python.
- It also supports both Windows and Linux environments.
- Azure App Service lets you focus on building and maintaining your app, and Azure focuses on keeping the environment up and running.

App Service Plans

- In App Service, an app always runs in an *App Service plan*.
  - An App Service plan defines a set of compute resources for a web app to run.
- One or more apps can be configured to run on the same computing resources (or in the same App Service plan). Each App Service plan defines:
  - Operating System (Windows, Linux)
  - Region (West US, East US, etc.)
  - Number of VM instances
  - Size of VM instances (Small, Medium, Large)
  - Pricing tier (Free, Shared, Basic, Standard, Premium, PremiumV2, PremiumV3, Isolated, IsolatedV2)

Examine Azure App Service plans - Training | Microsoft Learn

App Service Plans

- The *pricing tier* of an App Service plan determines what App Service features you get and how much you pay for the plan. There are a few categories of pricing tiers:
  - **Shared compute**: **Free** and **Shared**, the two base tiers, runs an app on the same Azure VM as other App Service apps, including apps of other customers. These tiers allocate CPU quotas to each app that runs on the shared resources, and the resources can't scale out. [Used for Dev/Testing purposes]
  - **Dedicated compute**: The **Basic**, **Standard**, **Premium**, **PremiumV2**, and **PremiumV3** tiers run apps on dedicated Azure VMs. Only apps in the same App Service plan share the same compute resources. The higher the tier, the more VM instances are available to you for scale-out.
  - **Isolated**: The **Isolated** and **IsolatedV2** tiers run dedicated Azure VMs on dedicated Azure Virtual Networks. It provides network isolation on top of compute isolation to your apps. It provides the maximum scale-out capabilities.