

2- Describe Azure containers

- VM's
 - con's
 - pro's
- Containers
 - What are containers?
 - Azure Container Instances
 - Azure Container Apps:
 - Azure Kubernetes Service
 - Use containers in your Solutions

VM's

con's

- limited to a single operating system per virtual machine.
- Though containers are portable, they're constrained to the operating system they're defined for. For example, a container for Linux can't run on Windows, and vice versa.

pro's

- virtual machines are an excellent way to reduce costs versus the investments that are necessary for physical hardware

Containers

- **!** If you want to run multiple instances of an application on a single host machine, containers are an excellent choice.

What are containers?

- Containers are a virtualization environment.
- Like running multiple virtual machines on a single physical host. you can run multiple containers on a single physical or virtual host.
- Unlike virtual machines, you don't manage the operating system for a container. Virtual machines appear to be an instance of an operating system that you can connect to and manage.
- Containers are
 - *Lightweight*
 - designed to be *created*, *scaled out*, and *stopped dynamically*
- containers are a lighter weight, more agile method.

- Containers are designed to allow you to respond to changes on demand.
- With containers, you can quickly restart if there's a crash or hardware interruption.
- ! one of the most popular container engines is **Docker**, and Azure supports Docker.

Certainly! Here's a comparison between virtual machines (VMs) and containers in table format with expanded definitions:

Aspect	Virtual Machines (VMs)	Containers
Description	VMs are like having multiple separate computers within one physical computer.	Containers are like separate, self-contained applications that share the same operating system.
Operating System	Each VM includes its own full operating system, like Windows or Linux.	Containers do not have their own full operating system; they use the host OS, which makes them lightweight.
Resource Usage	VMs can be resource-intensive because of the OS duplication, taking up more memory and storage.	Containers are efficient in terms of resources since they don't duplicate the entire OS, making them lightweight.
Startup Time	VMs can be slower to start because they need to load a full operating system.	Containers start up quickly because they don't need to load a full OS.
Isolation	VMs provide strong isolation, separating applications from each other securely.	Containers provide good isolation, usually sufficient for most applications, but not as strong as VMs.
Size	VMs are generally larger in size due to the inclusion of a full OS.	Containers are generally smaller in size because they don't include a full OS.

Azure Container Instances

- Azure Container Instances are a **platform as a service (PaaS)** offering.
- 🌐 ACI is like a super quick and easy way to run your computer programs inside boxes in the cloud (containers).
- You don't have to worry about setting up or managing any special computer servers; just upload your program boxes, and ACI takes care of running them for you.
- It's like ordering takeout food; you get the meal without needing to cook or clean up afterward.

Azure Container Apps:

- Container Apps are a lot like ACI, but they offer some extra features.
- They also run your program boxes in the cloud, but they can do more. They can balance the workload and make sure everything runs smoothly even when lots of people use your programs at the same time.
- It's like having a team of helpers at a restaurant who not only serve your food but also make sure everyone gets their food quickly and the restaurant stays nice and organized.

In simple terms, both ACI and Container Apps make it easy to run your programs in the cloud, but Container Apps give you some extra tools to handle more customers and keep everything running smoothly. 🌐

Azure Kubernetes Service

- ! It is a container orchestration service.
 - service that manages the lifecycle of containers.
 - makes fleet management simpler and more efficient.
-

Use containers in your Solutions

- used to create solutions by using microservice architecture.
 - architecture: 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.
 - Imagine your website back-end has reached capacity but the front end and storage aren't being stressed. With containers, you could scale the back end separately to improve performance. If something necessitated such a change, you could also choose to change the storage service or modify the front end without impacting any of the other components.
-