

Azure Compute Options

1 Essential Azure compute concepts

Definition: Virtual Machines

Software emulations of physical computers.

Definition: Containers

A virtualization environment for running applications. Don't include an operating system for the apps running inside the container. Bundle the libraries and components needed to run the application and use the existing host OS running the container.

Definition: Azure App Service

Platform as a service designed to host web oriented applications.

Definition: Serverless Computing

Cloud hosted execution environment that runs your code but abstracts the underlying hosting environments.

2 Azure Virtual Machines

VMs are an ideal choice when you need:

- Total control over the OS
- The ability to run custom software
- To use custom hosting configurations

When to use virtual machines:

- During testing and development
- When running applications in the cloud
- When extending your datacenter to the cloud
- During disaster recovery

2.1 Scaling VMs

2.1.1 Availability set

Definition: Availability set

A logical grouping of two or more VMs that help keep your application available during planned or unplanned maintenance.

Definition: Planned maintenance event

When the underlying Azure fabric that hosts VMs is updated by Microsoft

When the VM is part of an availability set, updates are sequenced so that not all the associated VMs are rebooted at the same time. They are put into different update domains

Definition: Update domains

Groups of VMs and underlying physical hardware that can be rebooted at the same time

Definition: Unplanned maintenance

Hardware failure in the data centre, such as power outage or disk failure

Definition: Fault domain

The group of virtual machines that all share common hardware that has failed.

With an availability set you get:

- Up to three fault domains that have a server rack with dedicated power and network resources
- Five logical update domains which then can be increased to 10

There is no cost for an availability set, just for the VMs in the set

2.1.2 Virtual Machine Scale Sets

Virtual machine scale sets let you create and manage a group of identical, load balanced VMs.

This allows you to centrally manage, configure and update many VMs to provide highly available applications.

2.1.3 Azure Batch

Azure batch enables large scale job scheduling and compute management

Batch:

- Starts a pool of compute VMs for you
- Installs applications and staging data
- Runs jobs with as many tasks as you have
- Identifies failures
- Requeues work
- Scales down the pool as work completes

3 Containers

3.1 Azure Container Instances

PaaS to upload containers and execute them directly with automatic elastic scale

3.2 Azure Kubernetes Service

A complete orchestration service for containers with multiple architectures with multiple containers

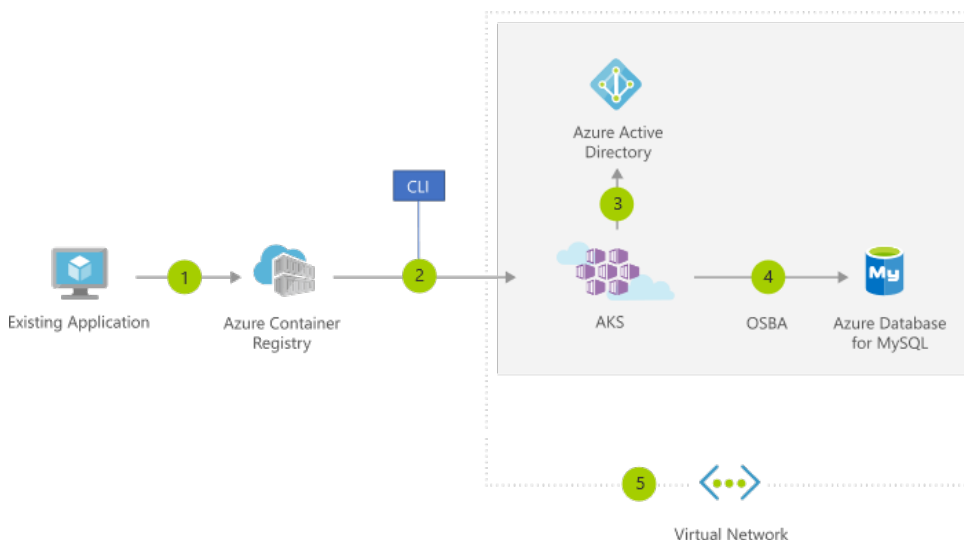
Definition: Orchestration

The task of automating, managing and interacting with many containers

3.3 Microservice architecture

Breaking solutions into smaller, independent pieces. This allows the separate components of the app to be maintained, scaled or updated independently.

3.4 Migrating to containers



The process is as

follows:

1. Convert the application to containers and publish to Azure Container Registry
2. Deploy the containers to an AKS cluster
3. Azure AD controls access to AKS resources
4. You access Azure services
5. Optionally, AKS is deployed with a virtual network

4 Azure App Service

4.1 Costs

You pay for the compute resources your app uses. The app service plan determines how much hardware is devoted to the host. There's even a free tier for low traffic sites

4.2 Types of app services

- Web apps - Host web apps
- API apps - Build REST-based Web APIs
- Web Jobs - Run a program or script in the same context as a web app. They can be scheduled or run by a trigger
- Mobile app back ends
 - Store mobile app data in a cloud based SQL database
 - Authenticate customers
 - Send push notifications
 - Execute custom back end logic

5 Serverless

Serverless encompasses three ideas:

- Abstraction of servers - Don't need to manage them
- Event-driven scale - Code runs when events come in
- Micro billing - Only pay for the time code runs

5.1 Azure functions