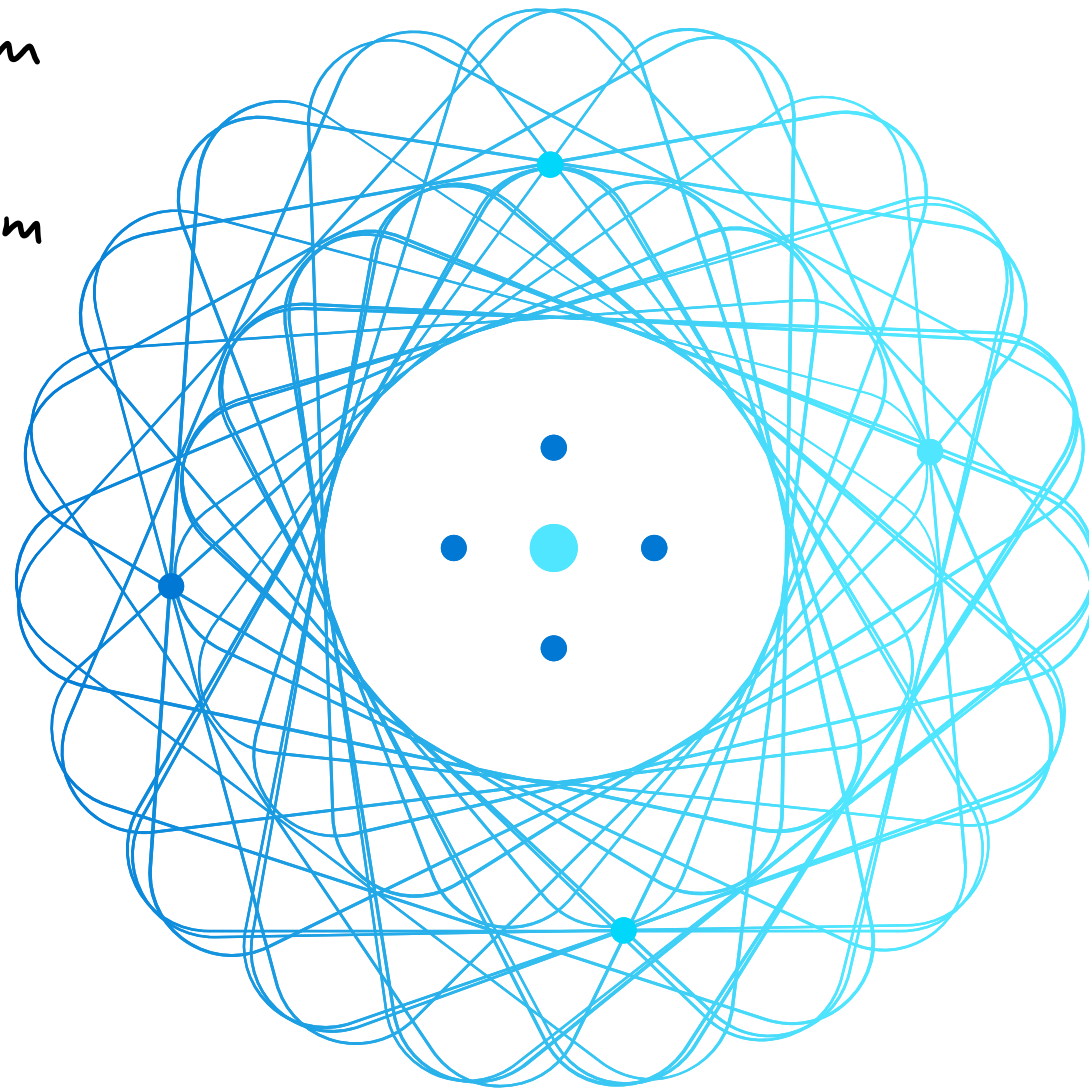


AZ-104 exam

AZ-305 exam

# AZ-305

# Designing Microsoft Azure Infrastructure Solutions



# AZ-305 Agenda

Module 01 Design a governance solution

Module 02 Design a compute solution ←

Module 03 Design a non-relational data storage solution

Module 04 Design a data storage solution for relational data

Module 05 Design a data integration solution

Module 06 Design an application architecture solution

Module 07 Design Authentication and Authorization Solutions

Module 08 Design a solution to log and monitor Azure resources

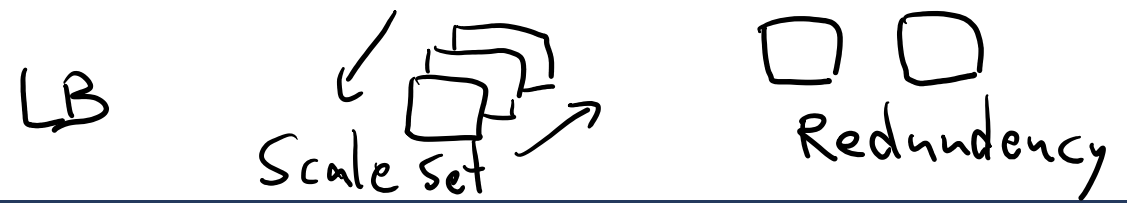
Module 09 Design a network infrastructure solution

Module 10 Design a business continuity solution

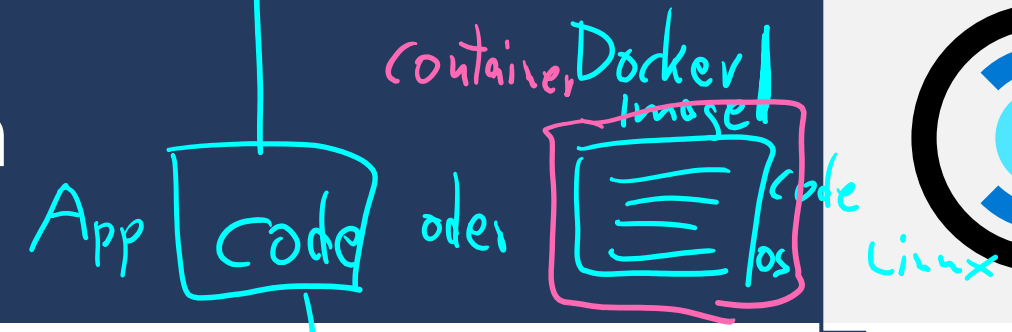
Module 11 Design a migration solution



VMWare on Azure



# Design a compute solution



Storage ← Security  
Read

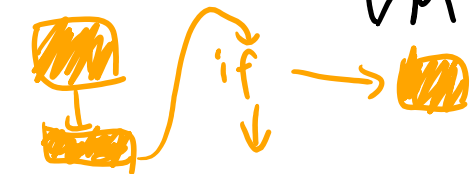
Serverless Computing

- Functions
- Logic App

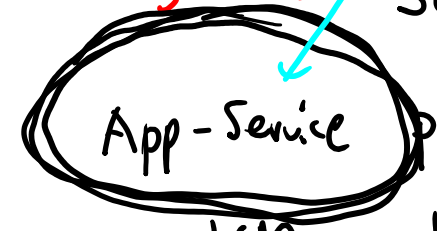
Event

Trigger

Flow



Managed ID



CI/CD  
SaaS

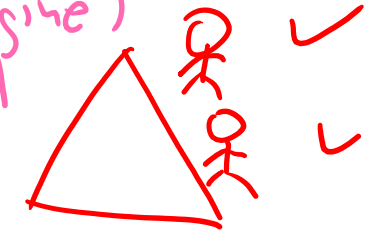
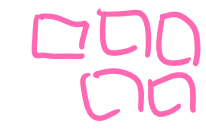
PaaS

VM

IaaS

(Docker Engine)  
containerd

ACI



Aks

SF  
Service Fabric

# Introduction

- Choose a compute service
- Design for Azure virtual machine solutions
- Design for Azure Batch solutions
- Design for Azure App Services solutions
- Design for Azure Container Instances solutions
- Design for Azure Kubernetes Service solutions
- Design for Azure Function solutions
- Design for Azure Logic App solutions
- Case study
- Summary and resources

## AZ-305: Design Infrastructure Solutions (25-30%)

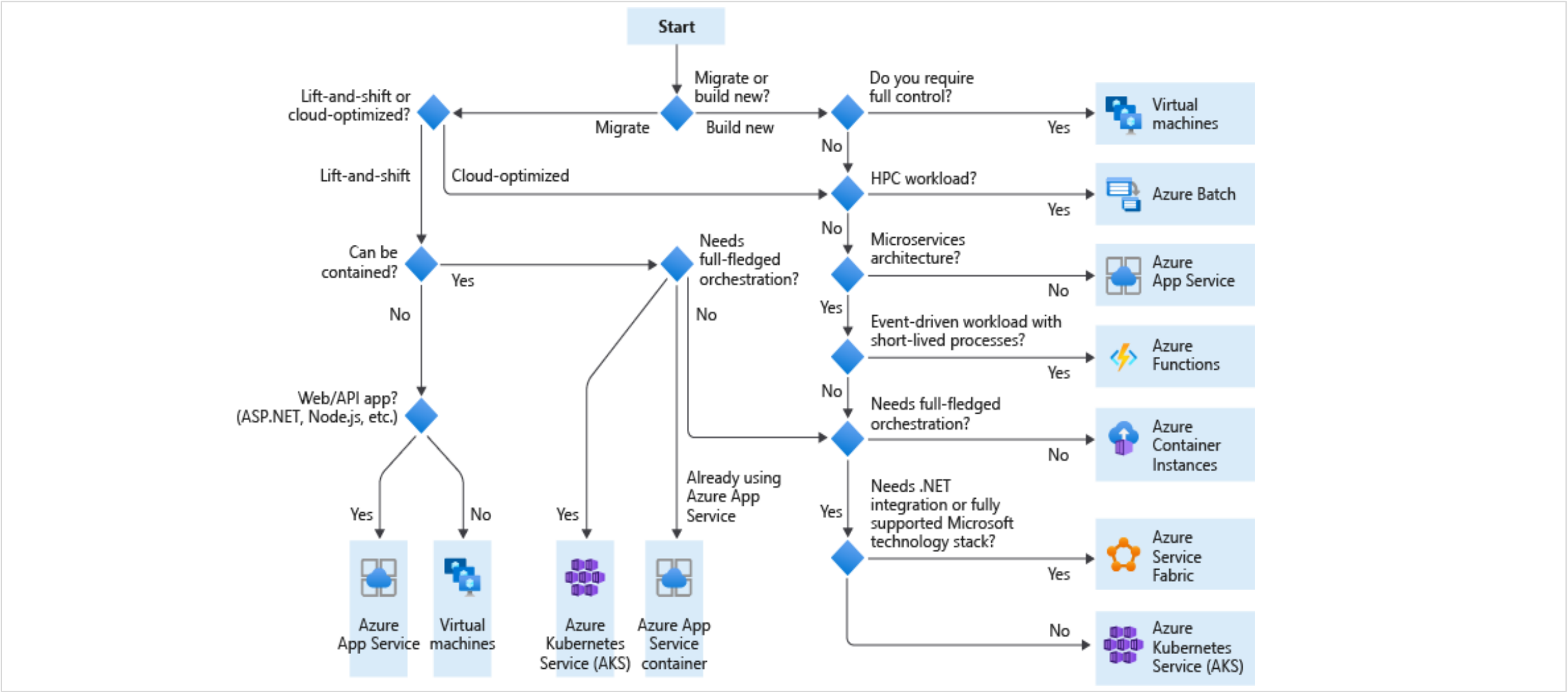
### Design for compute solutions

- Recommend an appropriately sized compute solution based on workload requirements.
- Recommend a Container-based compute solution.
- Recommend a Serverless-based compute solution
- Recommend a Virtual Machine-based compute solution

# Choose a compute solution



# Choose a compute service for your application



# Design for Azure virtual machine solutions



# When to select virtual machines

## Know when to use virtual machines

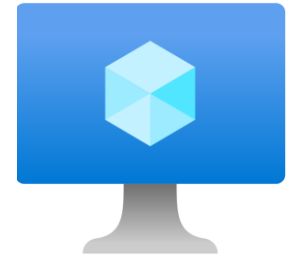
Lift Shift

- Quickly test and troubleshoot different configurations
- Use specialty hardware such as high-performance computing
- Extend your on-premises datacenter without purchasing additional hardware
- Run legacy apps on modern hardware
- Access third party software that requires local add-ons or plug-ins
- Quickly migrate apps to the cloud
- Fully control the computing environment

Rehosting

VM Wave  
Hyper-V Repl.

ASR Site Recovery

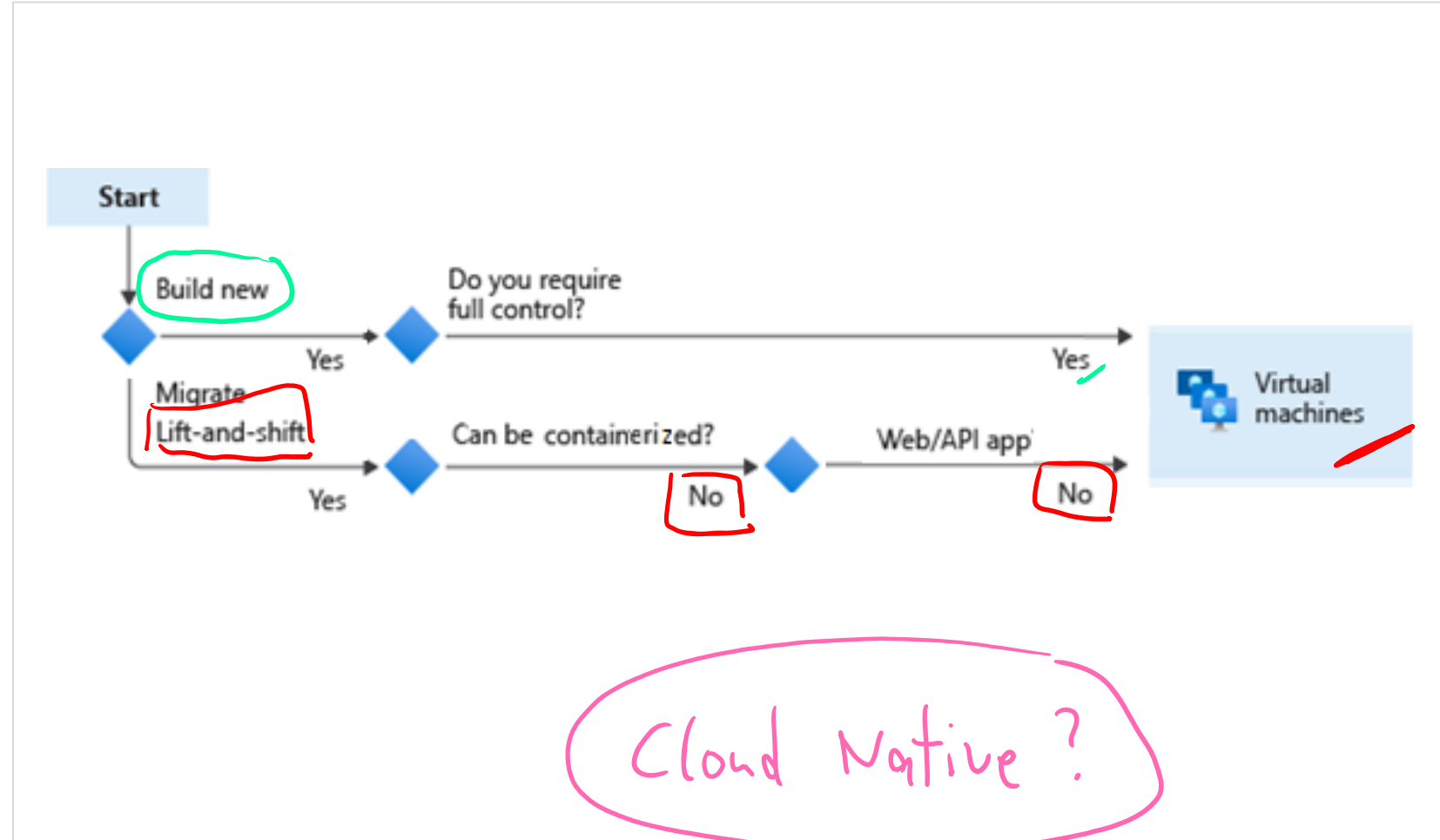




# Plan the Azure virtual machine deployment

Azure virtual machines scenarios include build new or migrate patterns

- Start with the network
- Name the VM
- Decide the VM location
- Select the VM storage
- Select an VM operating system
- Keep the VM up to date
- Monitor the VM



# Determine the virtual machine family

The virtual machine size determines pricing

B

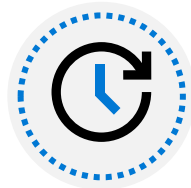
n



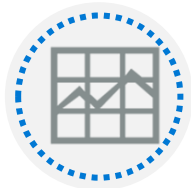
General purpose



Storage optimized



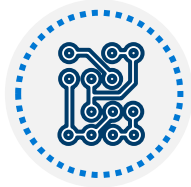
Compute optimized



GPU



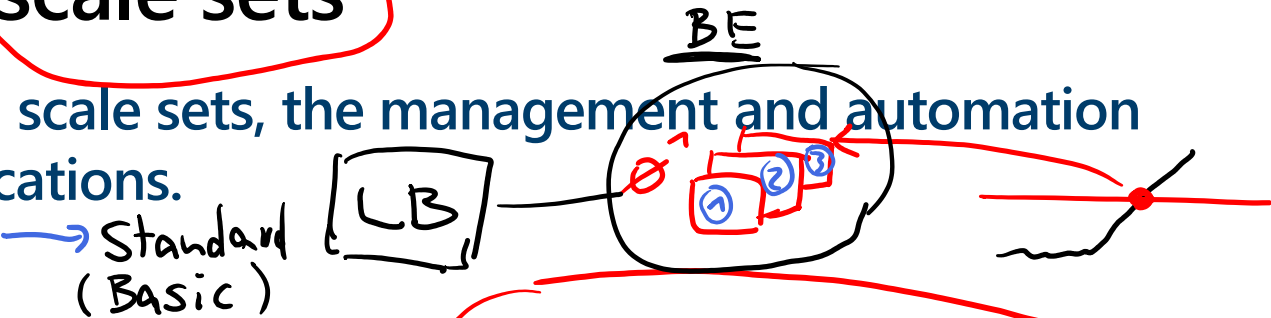
Memory optimized



High performance compute

# When to select virtual machine scale sets

Scale sets are built from virtual machines. With scale sets, the management and automation layers are provided to run and scale your applications.



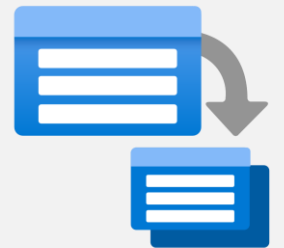
Scenario	Group of virtual machines	Virtual machine scale sets*
You need to add VM instances for changing workload	Manual process to create, configure, and ensure compliance	Automatically create from central configuration
You need to balance and distribute workloads	Manual process to create and configure Azure load balancer or Application Gateway	Can automatically create and integrate with Azure load balancer or Application Gateway
You need high availability and redundancy	Manually create <u>Availability Set</u> or distribute and track VMs across <u>Availability Zones</u>	Automatic distribution of VM instances across Availability Zone (when zones are selected) and Availability Set
You need to monitor and then scale virtual machines	Manual monitoring and Azure Automation	Autoscale based on host metrics, in-guest metrics, Application Insights, or schedule

\* Ensure application supports VMSS

Handwritten notes in blue and green ink:

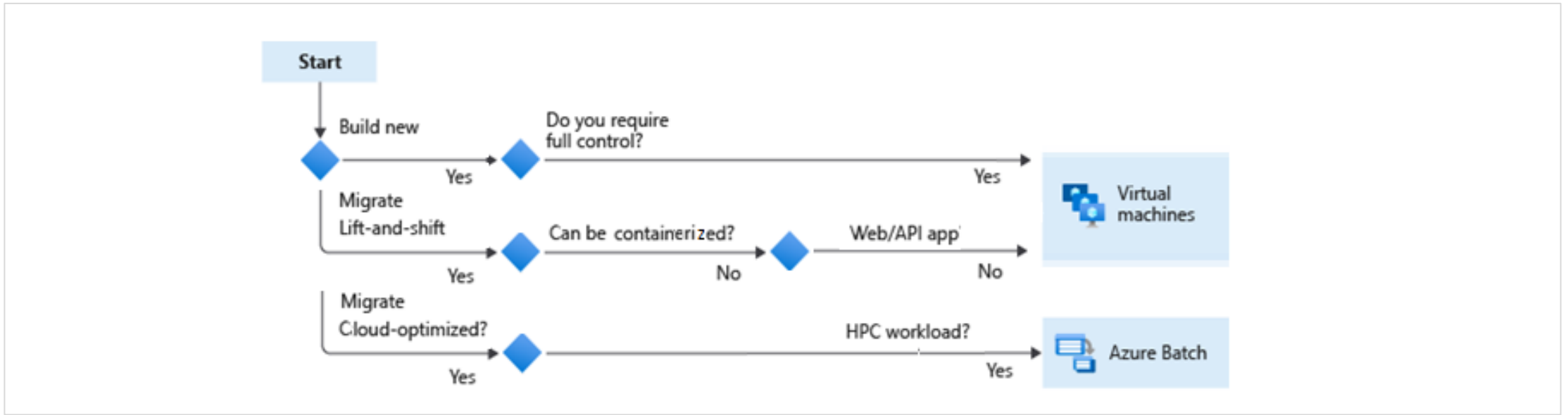
- DC 99.95
- AZ 1, 2, 3 (circled in green)
- 99.99
- Region
- vm 99.9% Premium Storage

# Design for Azure Batch solutions



# When to use Azure Batch

Azure Batch enables large-scale parallel and HPC batch jobs



- Compute-intensive tasks and dynamically adjust resources for your solution without managing infrastructure
- Create and manage jobs in a pool of compute nodes (virtual machines)
- Azure Batch can also install the application that you want to run, and schedule jobs to run on the compute nodes

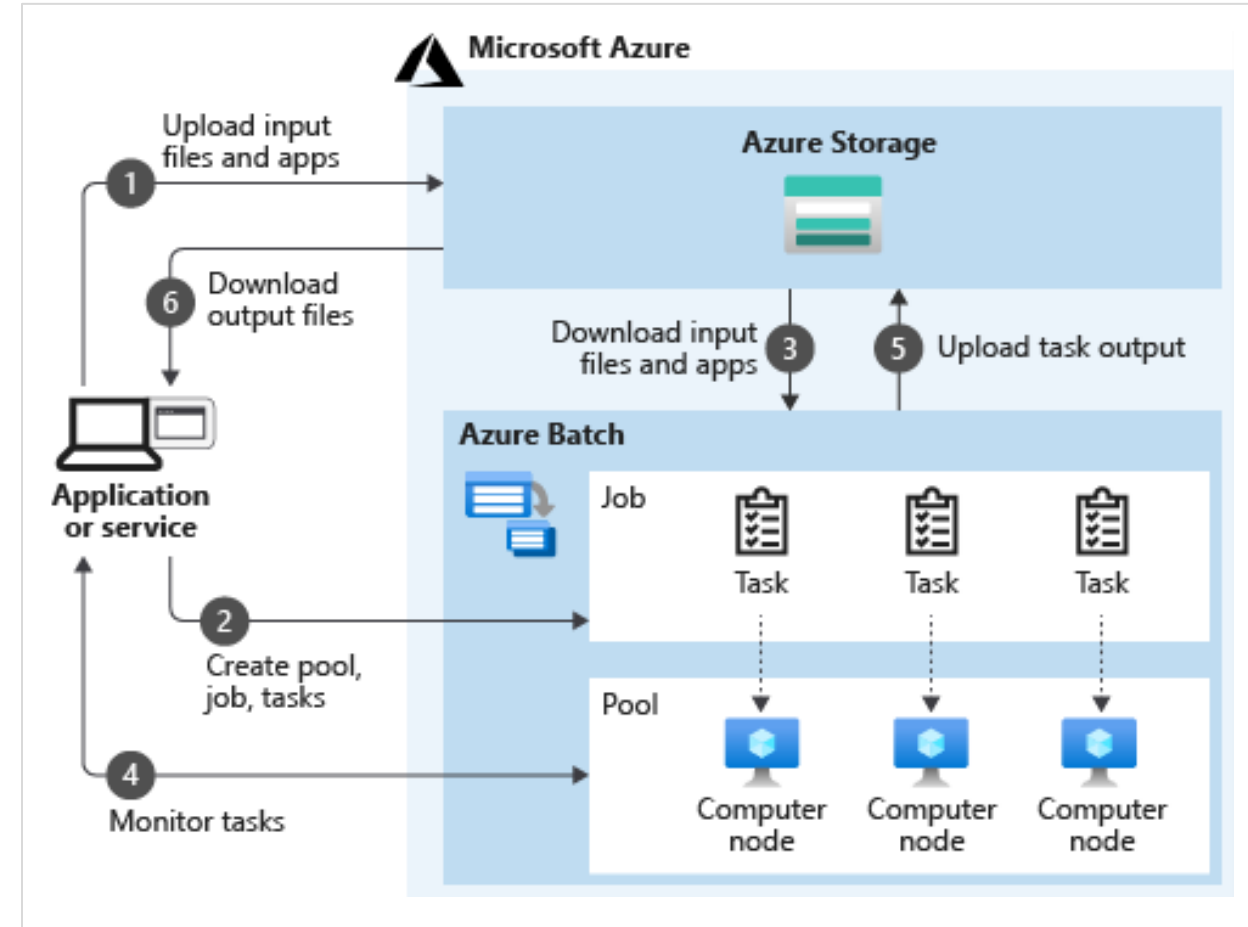
# How Azure Batch works

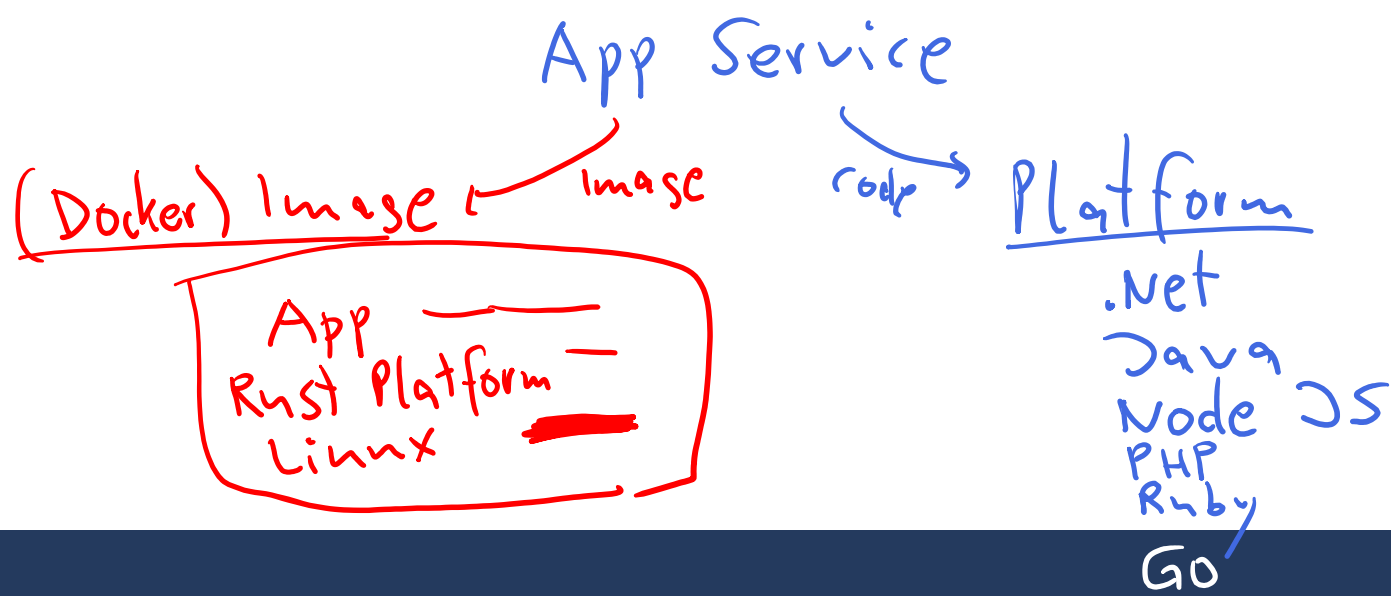
Think of the diagram in two parts:

- **Your service** that uses Azure as the platform
- **Batch as the compute platform behind your service.** Batch uses Azure Storage to fetch applications or data needed to complete a task.

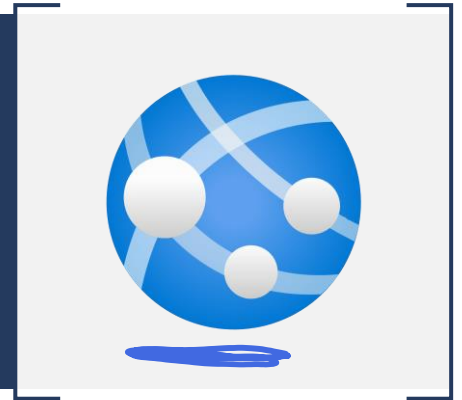
## Considerations

- Pools – dynamically allocate jobs, right size the pools
- Nodes – use multiple nodes in the pool, right size the machines
- Jobs – uniquely name the jobs, monitor and log activity





# Design for Azure App Services solutions



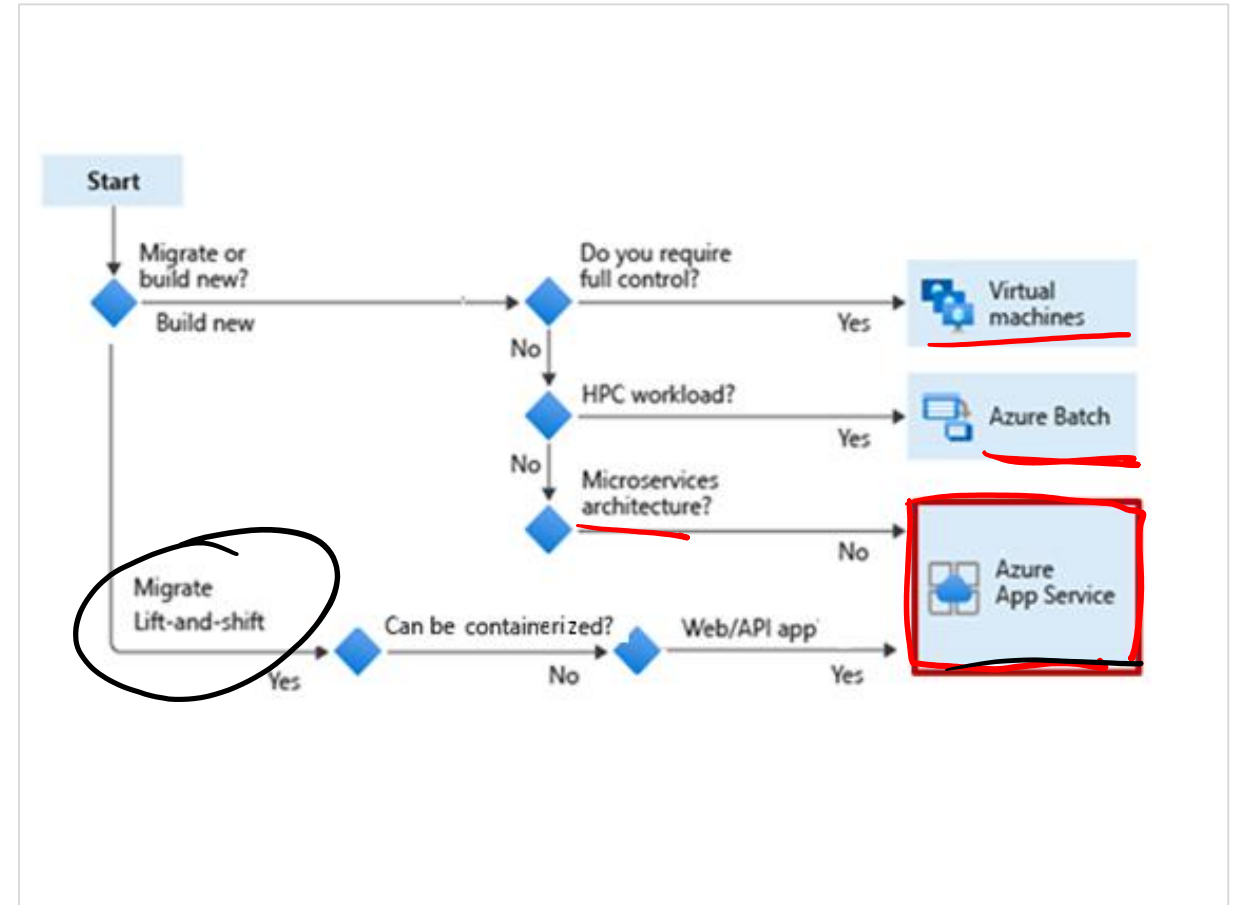
VM

Linux Windows Apache IIS

# When to use Azure App Services Web Apps

Azure App Service is an HTTP-based service that lets you build and host apps

- Web apps, background jobs, mobile backends, and RESTful APIs.
- Use the programming language of your choice
- Automatic scaling and high availability
- App Service enables automated deployments from GitHub, Azure DevOps, or other source control services



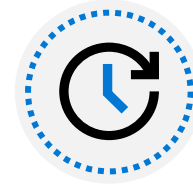


# Considerations for App Service Web Apps

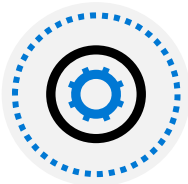
Fortran 77



Determine the appropriate app service plan – determines cost and scaling



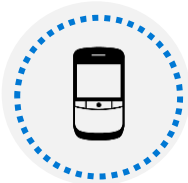
Use built-in authentication and authorization capabilities



Use App Services deployment slots for continuous deployment



Build REST-based web apps with the API



Use the Mobile Apps to build a back end for iOS and Android apps.



Use WebJobs to run a program or script

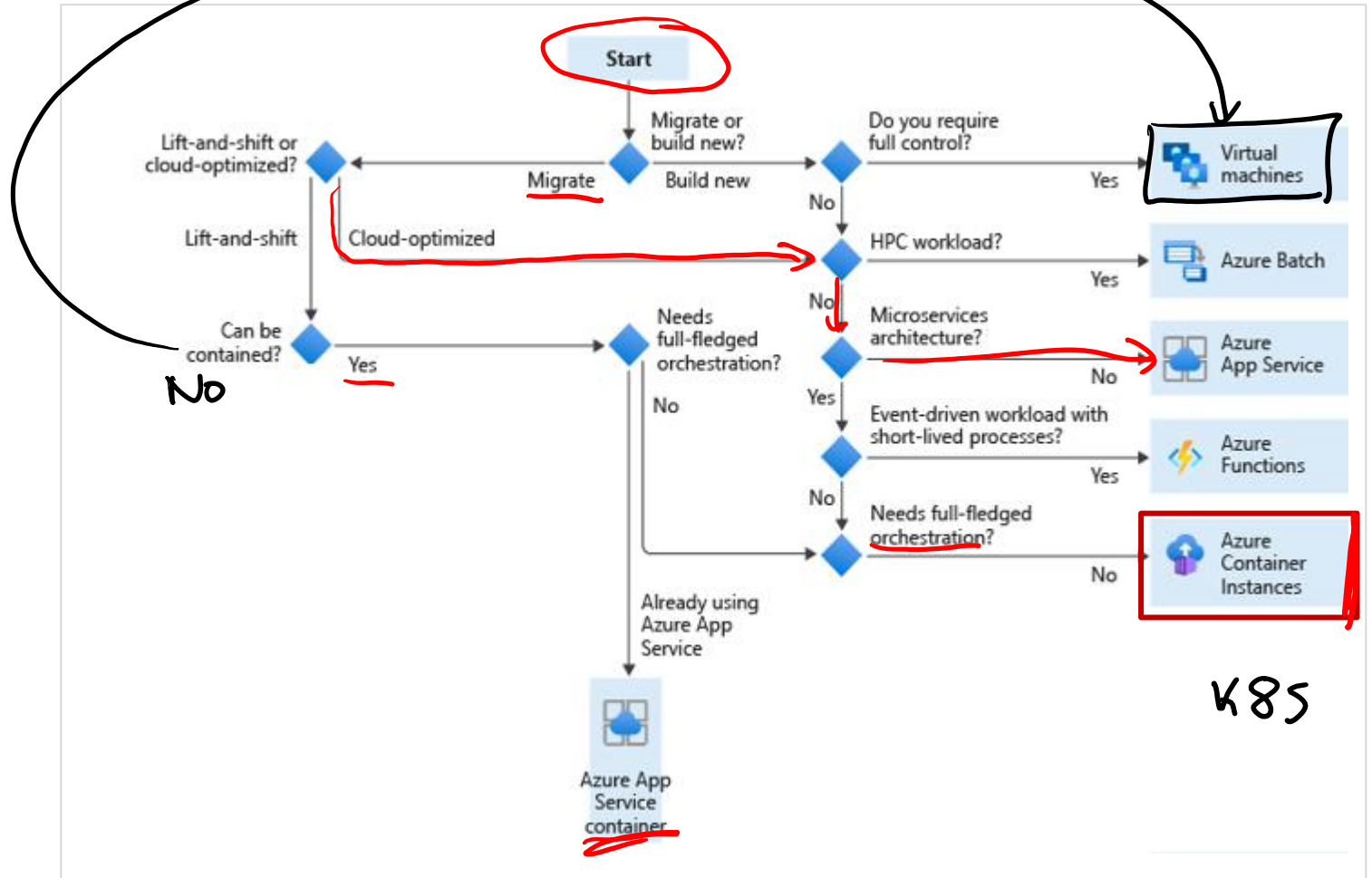
# Design for Azure Container Instances solutions



# What is Azure Container Instance

Azure Container Instances 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.

- Ensure the integrity of images throughout the lifecycle
- Monitor container resource activity
- Consider container groups



# When to use Azure Container Instances

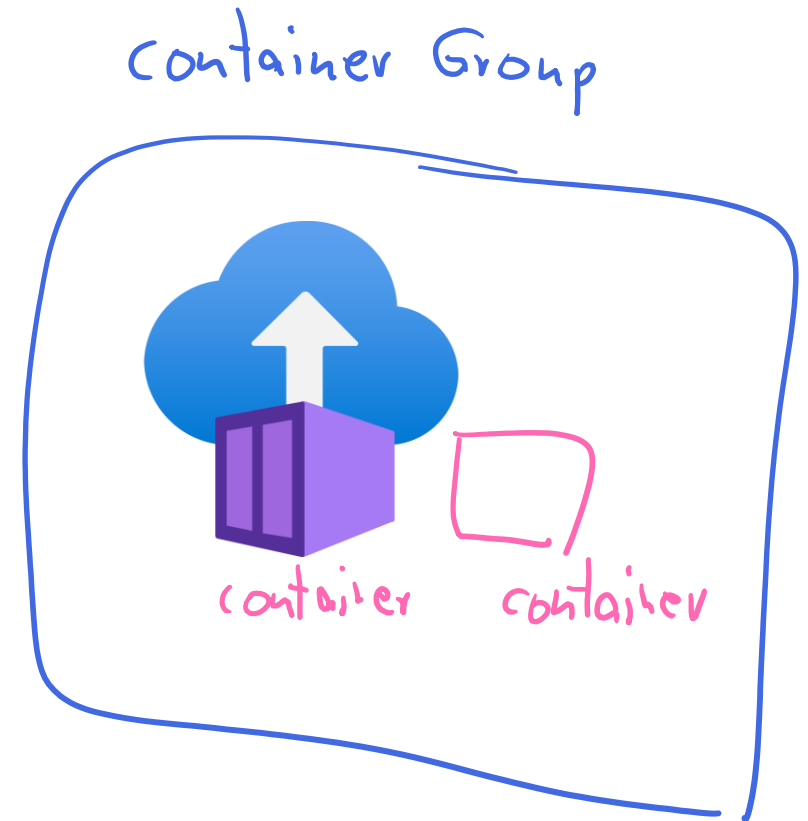
Azure Container Instances are a fast and simple way to run a container on Azure.

## Pros:

- Fast and Easy
- Used for testing and development.
- Used for short-lived processes.
- Can be used for AKS overflow

## Cons:

- Doesn't scale !
- Not designed for microservices !



Dockerfile

~~docker build .~~  
az acr build

ACR

ACI

AKS

API Registry

# Compare containers to virtual machines

App Services can also run containers

Feature	Containers	Virtual Machines
Isolation	Lightweight isolation from the host and other container. <i>Shared Kernel</i>	Complete isolation, strong security boundary, from the host and other VMs. <i>Own Kernel</i>
Operating system	Runs the user mode portion of an operating system	Runs a complete operating system
Deployment	Deploy using Docker <i>docker run</i>	Deploy in the cloud or on-premises – portal, PowerShell, CLI, templates, automation ..
Persistent storage	Azure disks or file share <i>✗</i>	Virtual hard disk (VHD) or a file share
Fault tolerance	If a cluster node fails, any containers running on it are rapidly recreated by the orchestrator on another cluster node.	VMs can fail over to another server in a cluster, with the VM's operating system restarting on the new server.

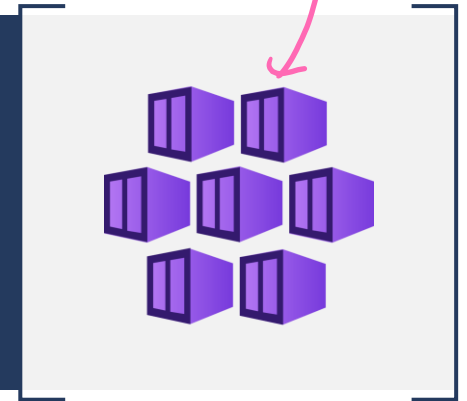
# Design for Azure Kubernetes Service solutions

K8S

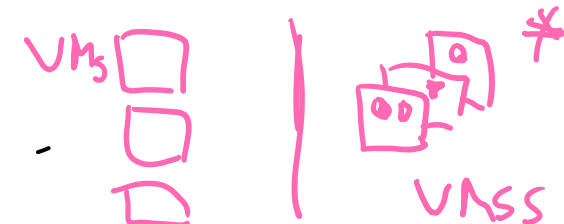
Microservices

Service Mesh

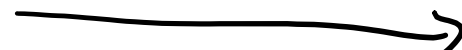
Pod



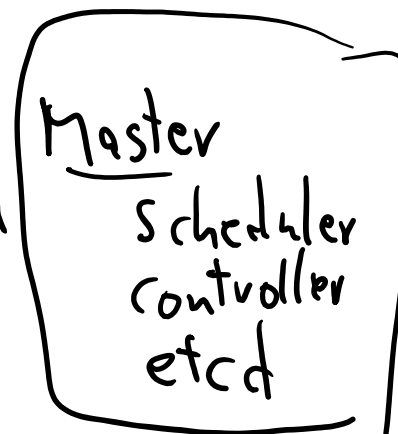
Cluster  
Compute Nodes



yaml



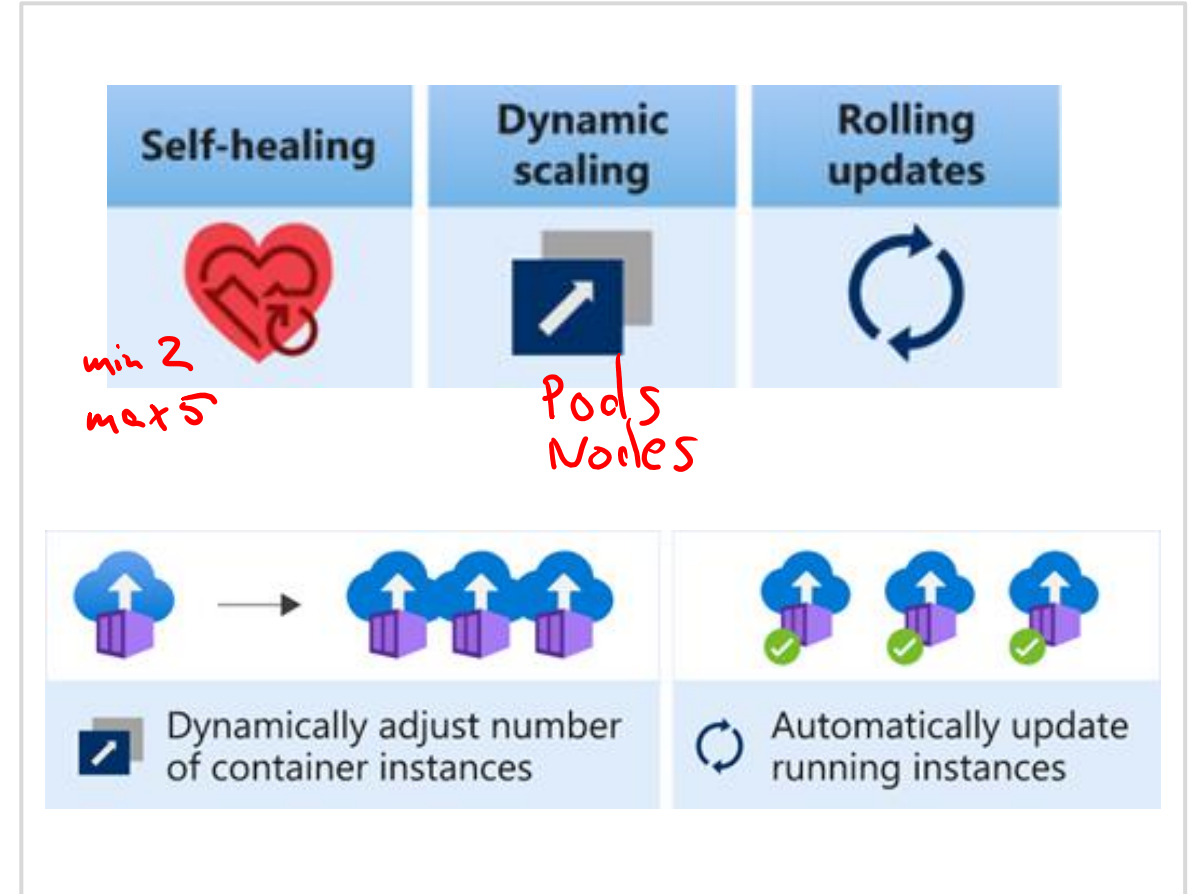
API



# Considerations for Azure Kubernetes Services

Azure Kubernetes Service (AKS) orchestrates your containerized applications

- Enterprise scalability
- Automatic cluster node and pod scaling
- Granular network control
- Cluster node upgrades
- Storage volume support
- Ingress with HTTP application routing support
- Private container registry support



# Design a highly available container solution

AKS provides high availability using multiple nodes in a virtual machine scale set.

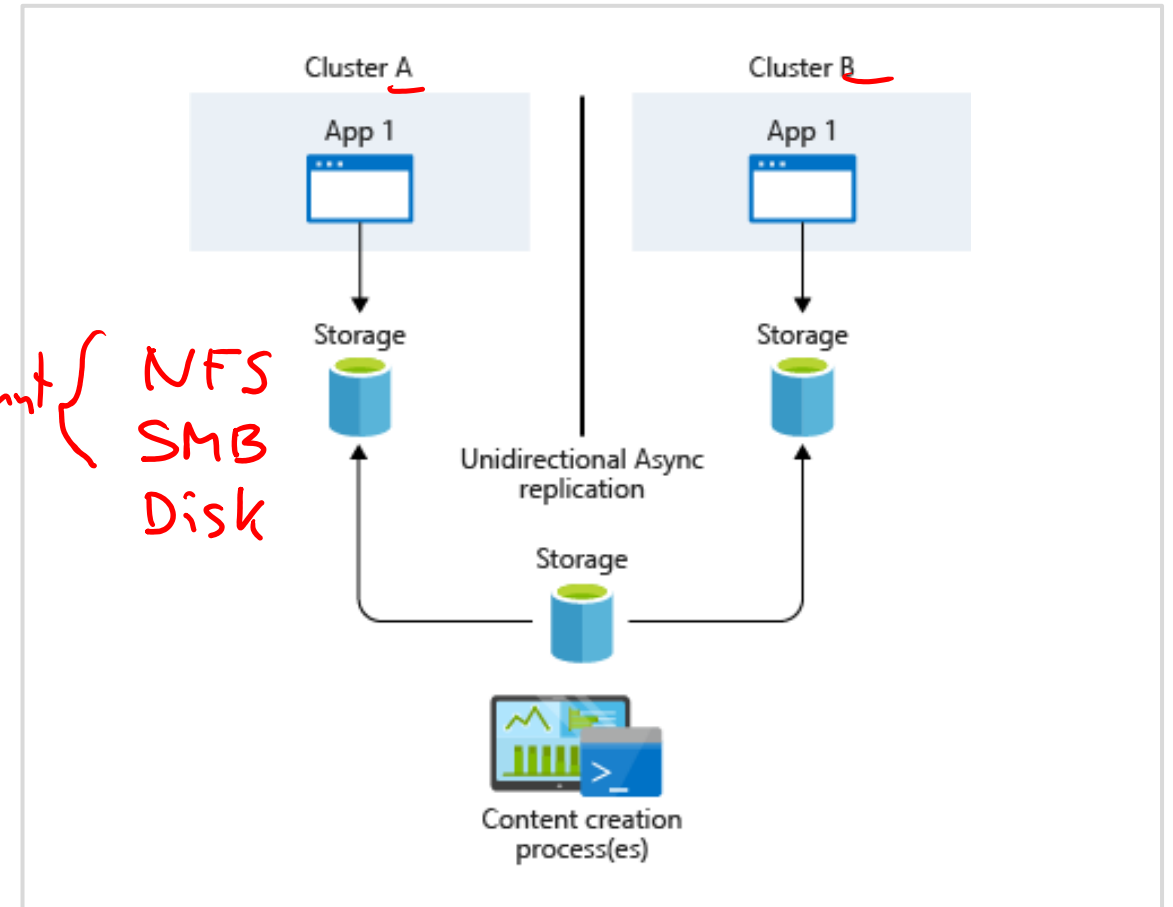
When planning to implement AKS clusters across multiple region deployments, consider the following:

- AKS region availability
- Azure paired regions
- Service availability

Two ways to synchronize storage.

- Infrastructure-based asynchronous replication
- Application-based asynchronous replication

Storage Account { NFS  
SMB  
Disk





"Serverless"

# Design for Azure Functions



# When to use Azure Functions

Azure Functions is a serverless application platform for compute-on-demand

- Implement your system's logic into readily available blocks of code
- Supports a microservice design
- Promotes code reuse
- Scales easily
- Event-driven

Trigger

Code

Logic

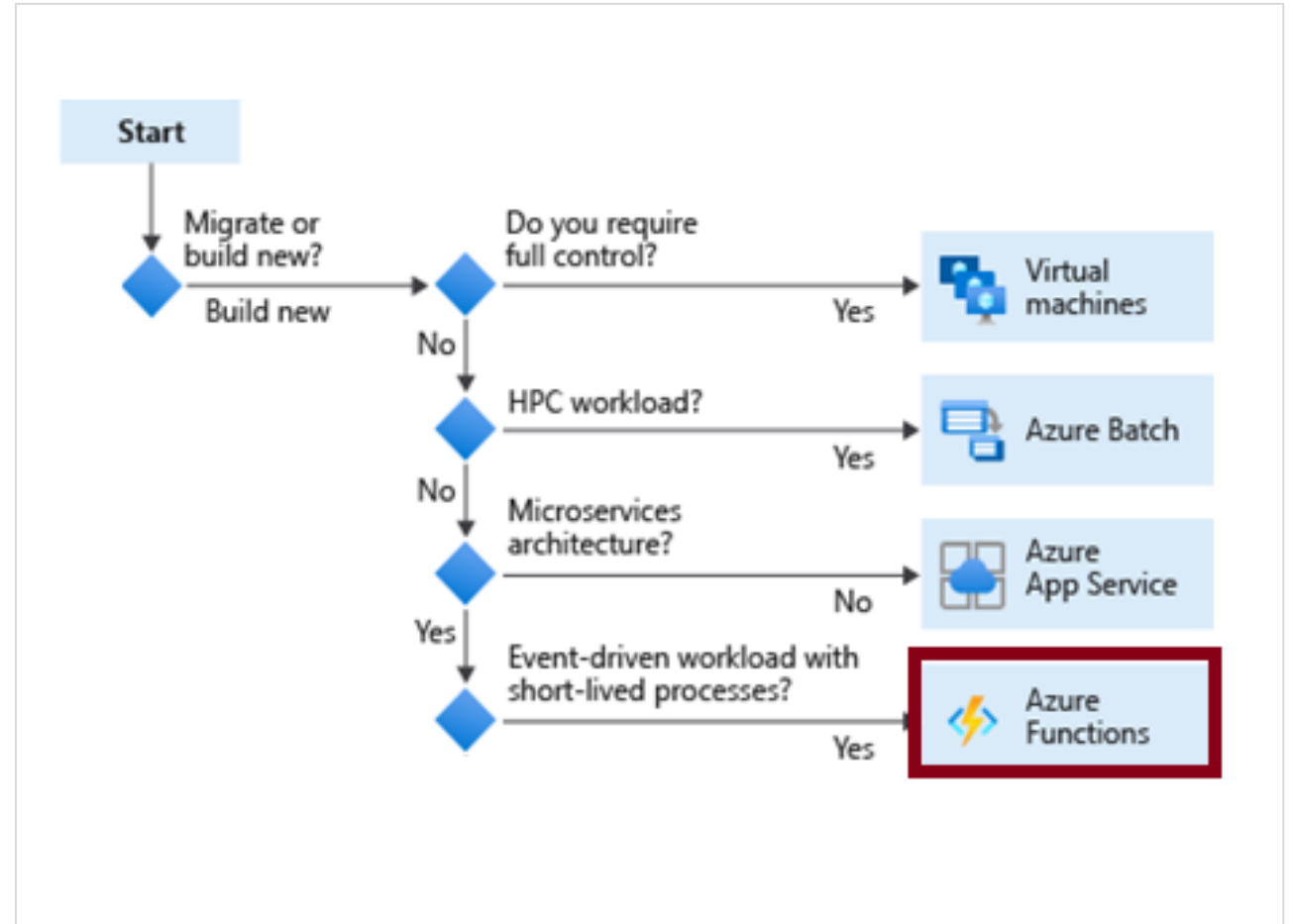
App

Connector

Trigger

Monitor  
Blob

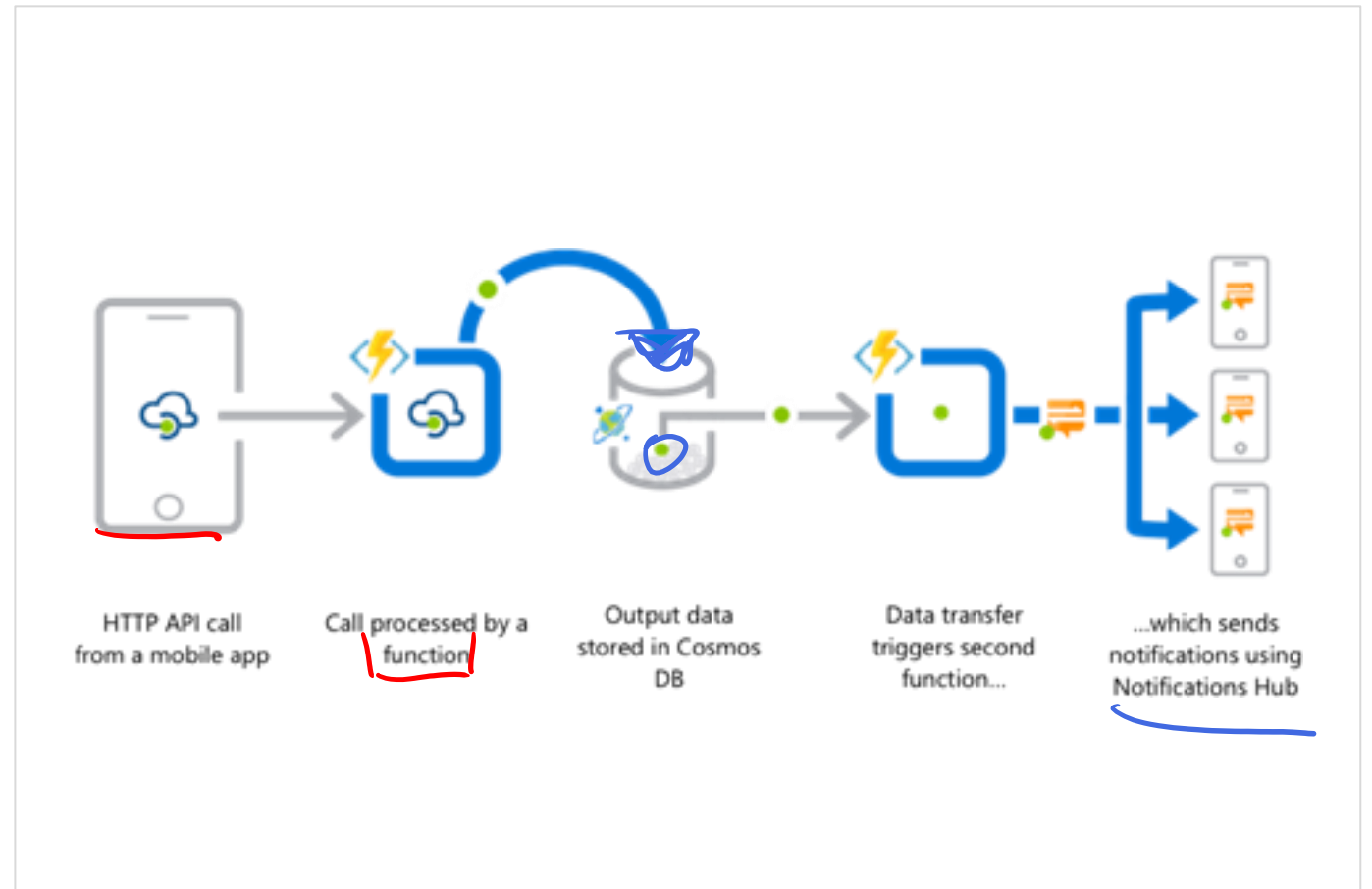
Event Grid



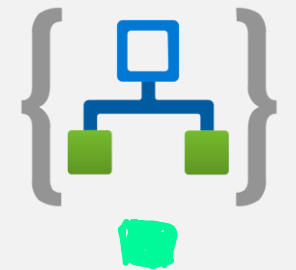
# Considerations for Azure Functions

Azure Functions are best when handling specific definable actions triggered by an event.

- Avoid long running functions
- Know when to use durable functions
- Organize functions for performance and scaling
- Write defensive functions



# Design for Azure Logic App solutions

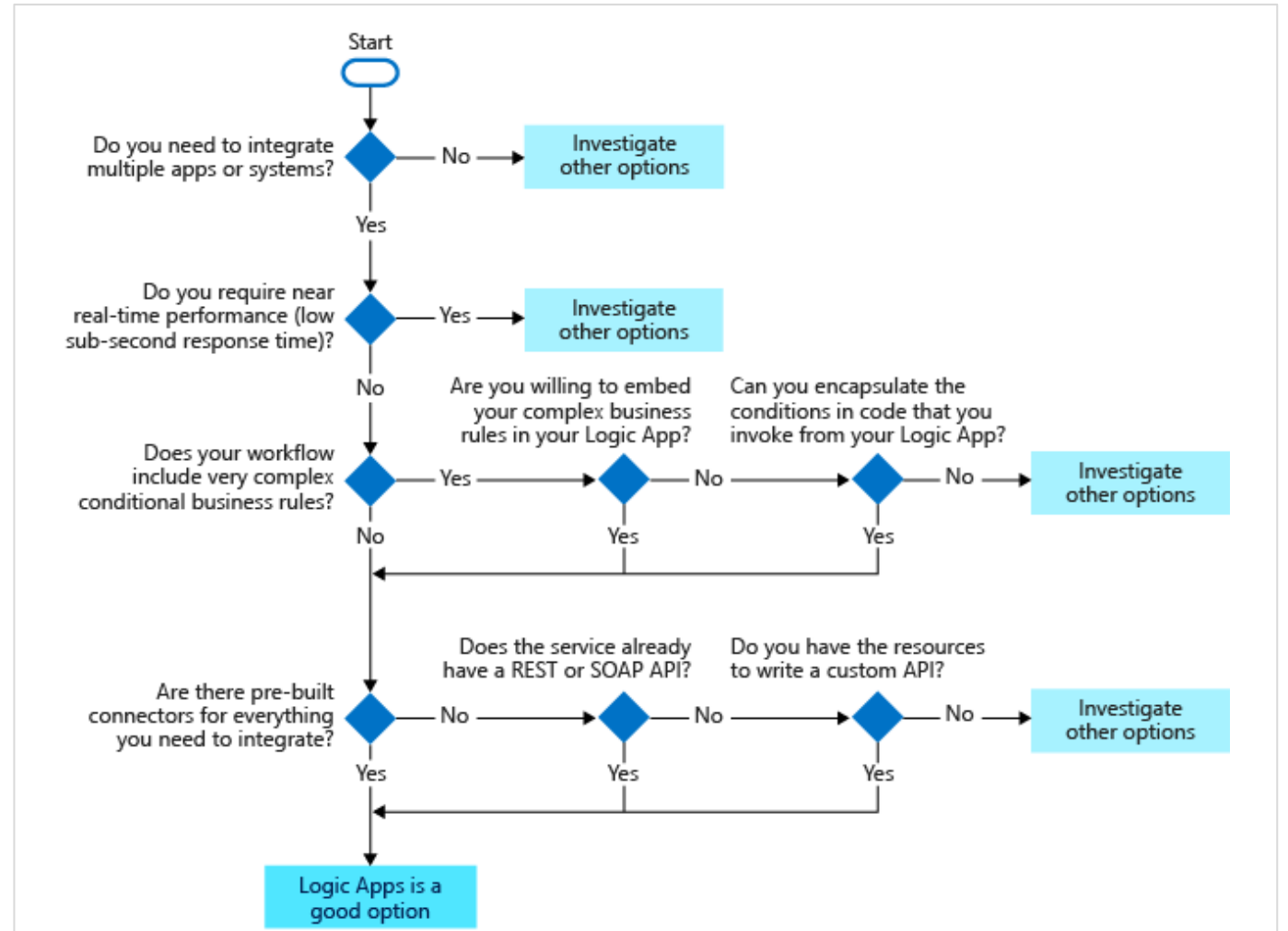


Custom  
connector

# When to use Azure Logic Apps

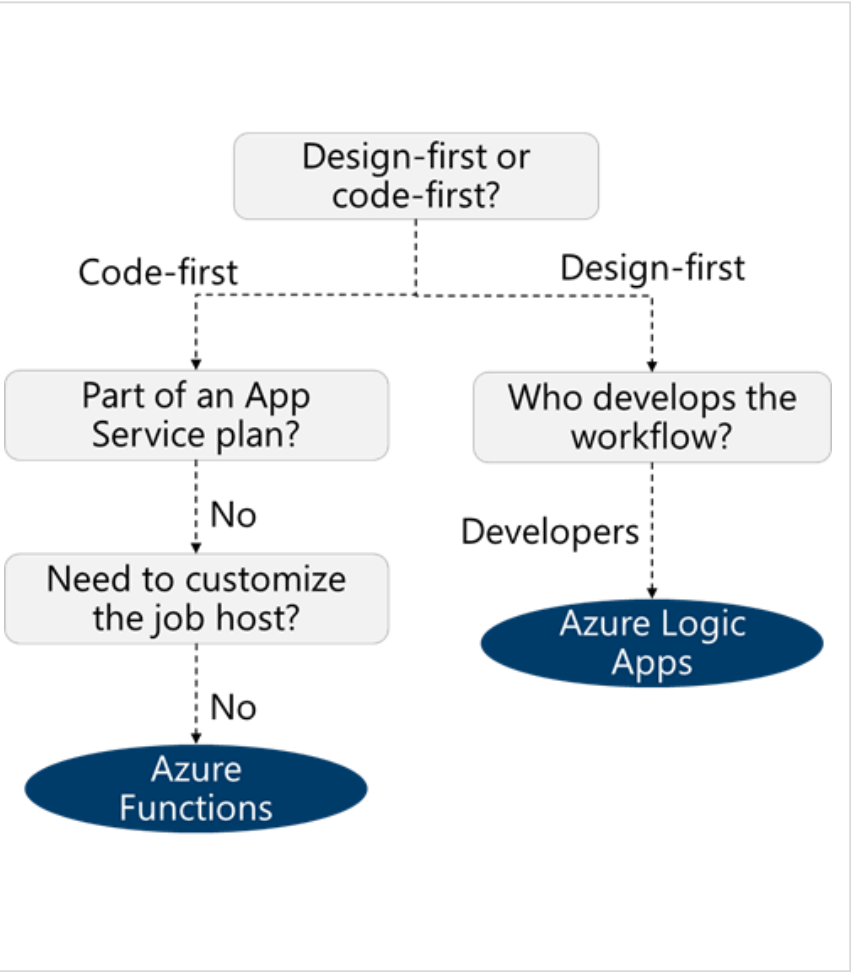
Azure Logic Apps is a cloud-based platform creating and running workflows.

- Send email notifications using Office 365 when a specific event happens
- Route and process customer orders across on-premises systems and cloud services.
- Move uploaded files from an FTP server to Azure Storage.
- Monitor tweets, analyze the sentiment, and create alerts or tasks for items that need review.



# How are Logic Apps different from Functions?

Code-first vs designer-first

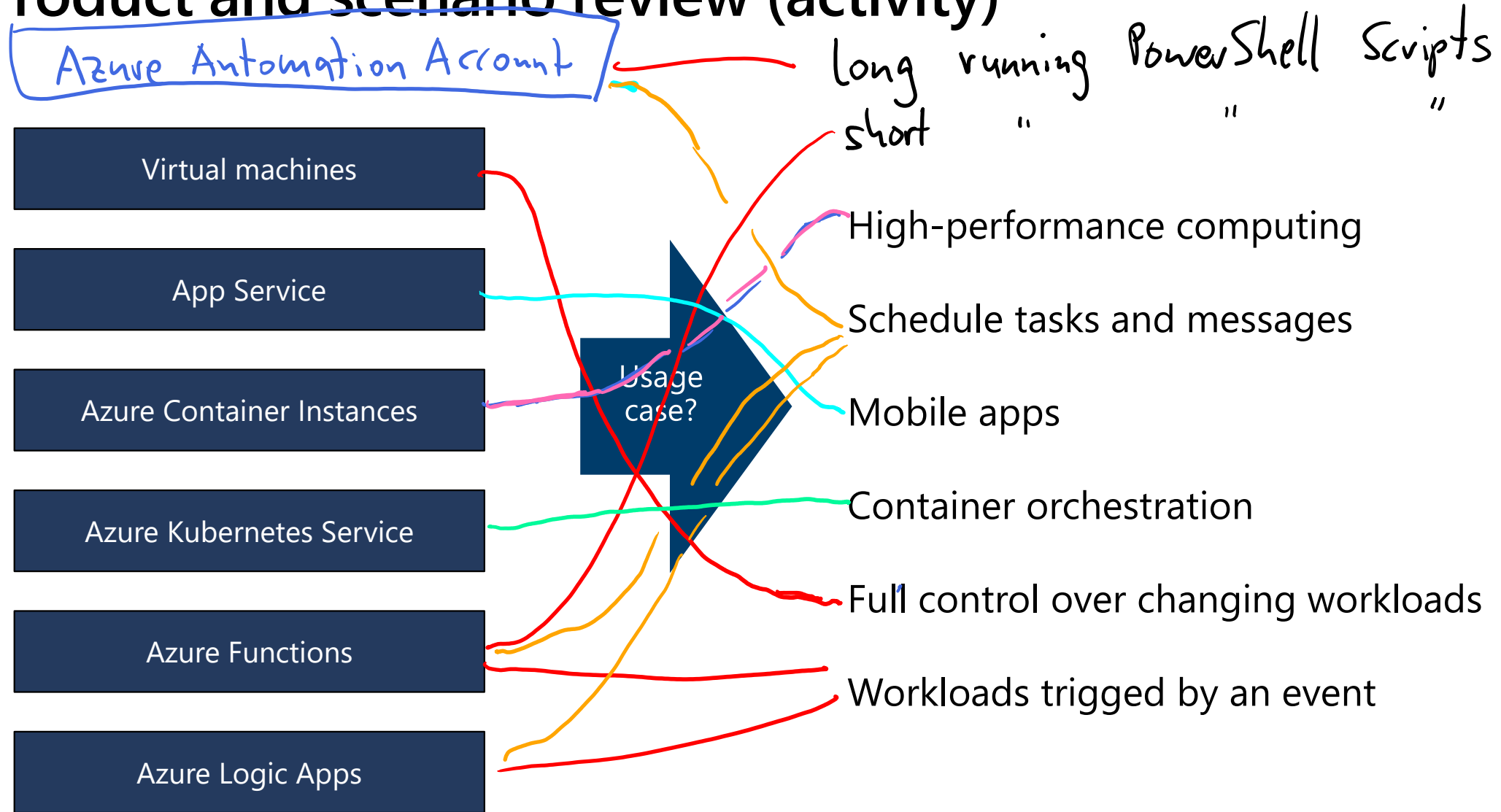


Comparison	Logic Apps	Durable Functions
Development	Designer-first	Code-first
Method	Create orchestrations by using a <u>GUI</u> or editing configuration files	Write code and use the durable functions extension
Connectivity	Large collection of connectors , B2B pack, <u>custom</u> connectors	Large selection of built-in binding types, write code for custom bindings
Monitoring	Azure portal, Azure Monitor logs	Azure Application Insights

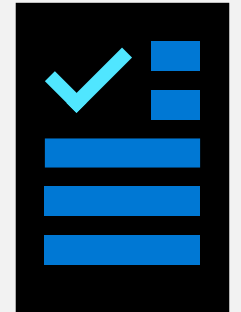
Monitor

App Service Plan  
Consumption Plan

# Product and scenario review (activity)



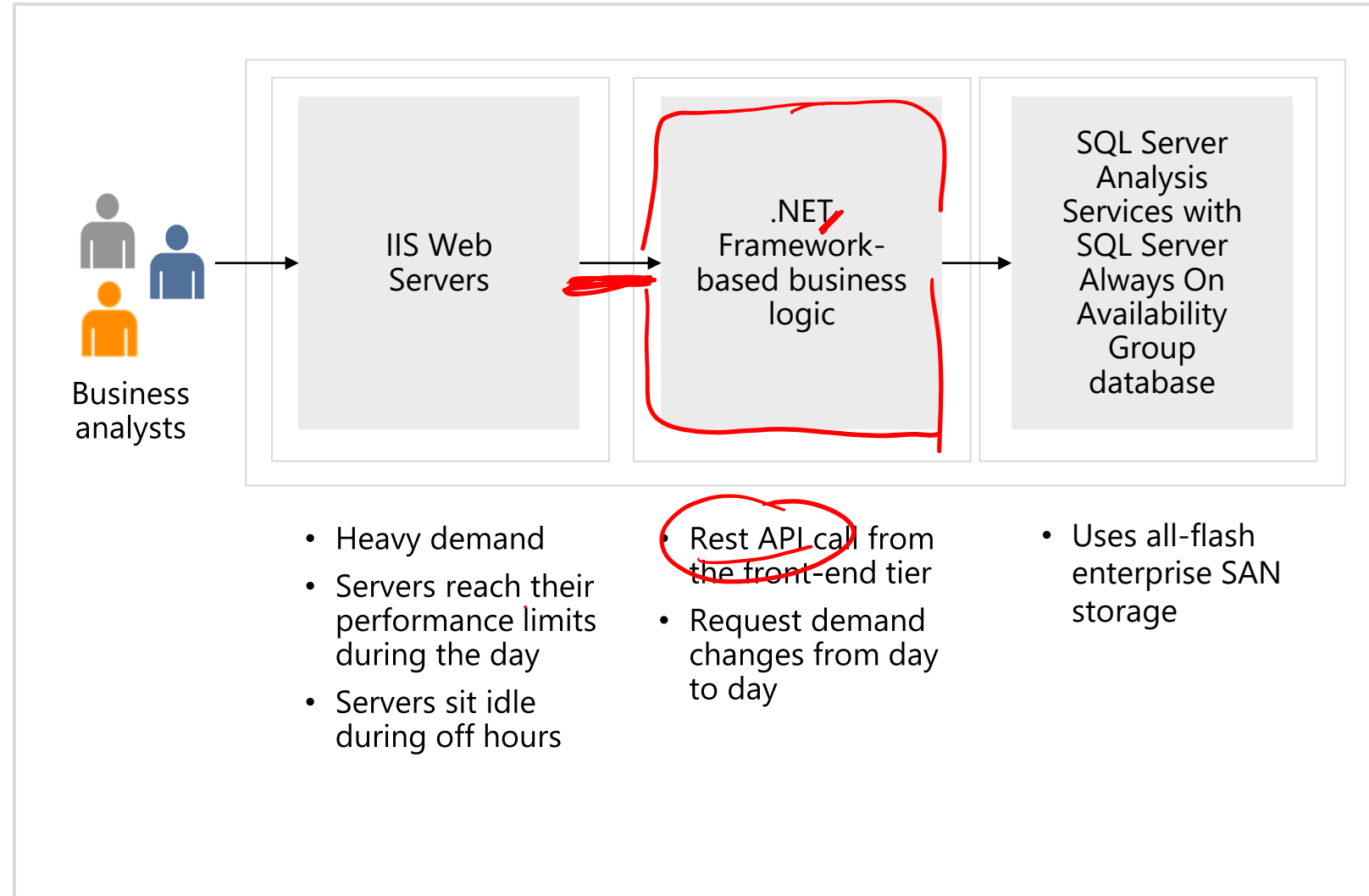
# Case studies and review



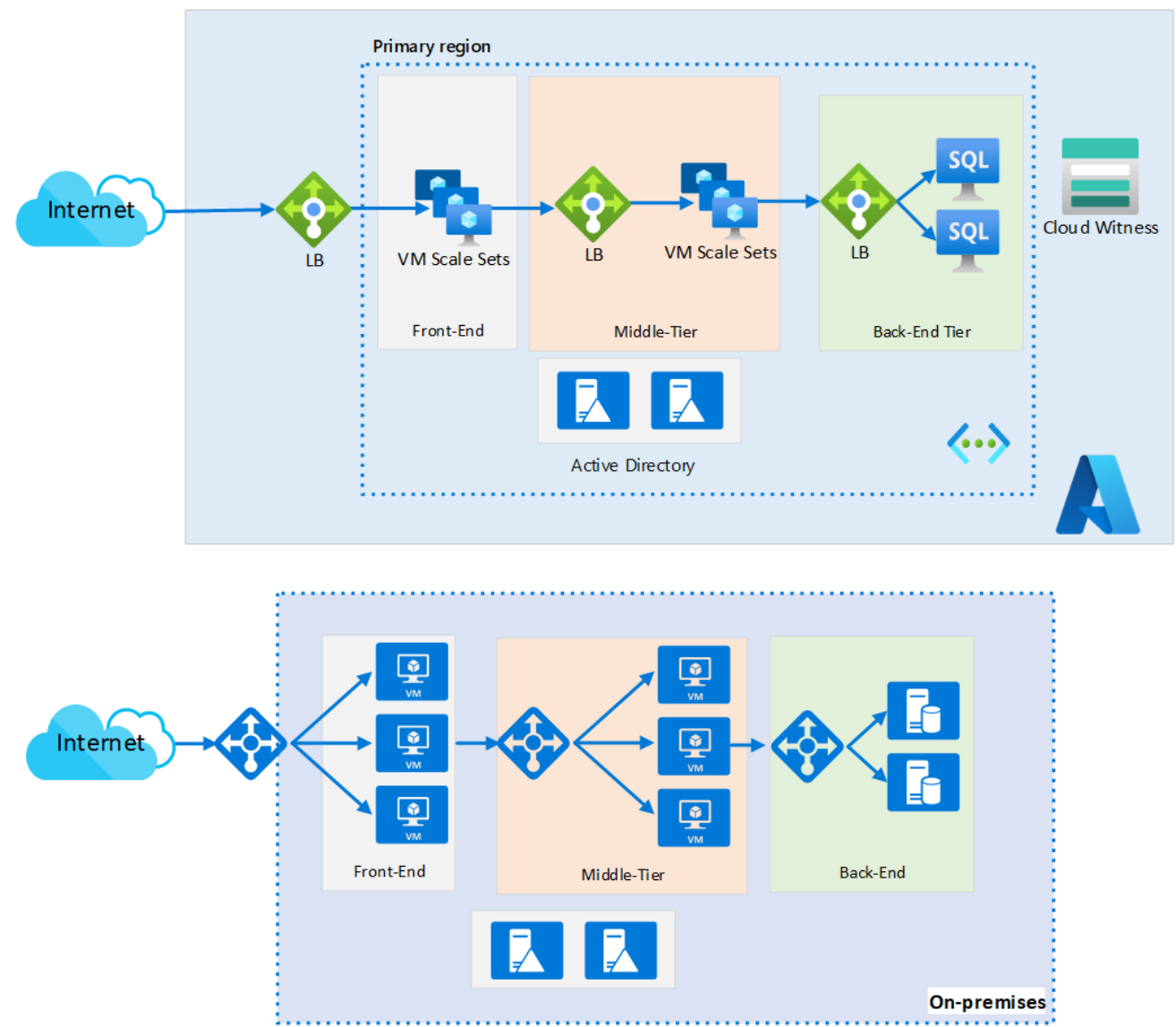


# Case study – Compute solutions

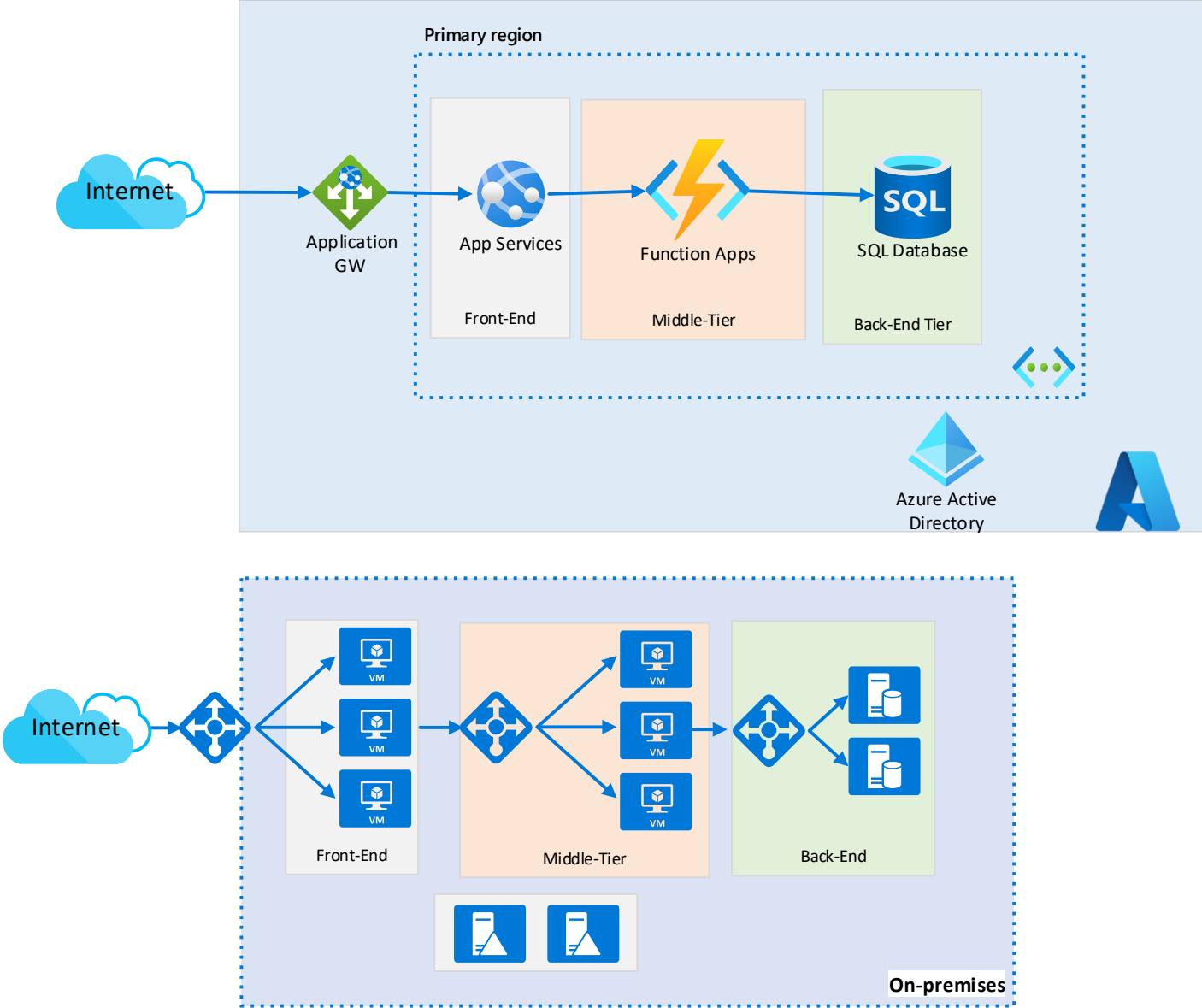
- Which Azure compute service would you recommend for the front-end tier? Address both the workload hosting and the web application.
- Which Azure compute service would you recommend for the middle tier application? Diagram what you decide and explain your solution.



# Instructor Solution – IaaS Version



# Instructor Solution – PaaS Solution



# Summary and resources

## Check your knowledge



## Microsoft Learn Modules ([docs.microsoft.com/Learn](https://docs.microsoft.com/Learn))

[Choose the best Azure service to automate your business processes](#)

[Introduction to Azure virtual machines](#)

[Create an Azure Batch account](#)

[Create serverless logic with Azure Functions](#)

[Introduction to Azure Logic Apps](#)

[Host a web application with Azure App Service](#)

[Introduction to Azure Kubernetes Service](#)

Optional hands-on exercises:

- [Create a web app in the Azure portal](#)
- [Create a Windows virtual machine](#)
- [Create the social media tracker Logic App](#)

# End of presentation

