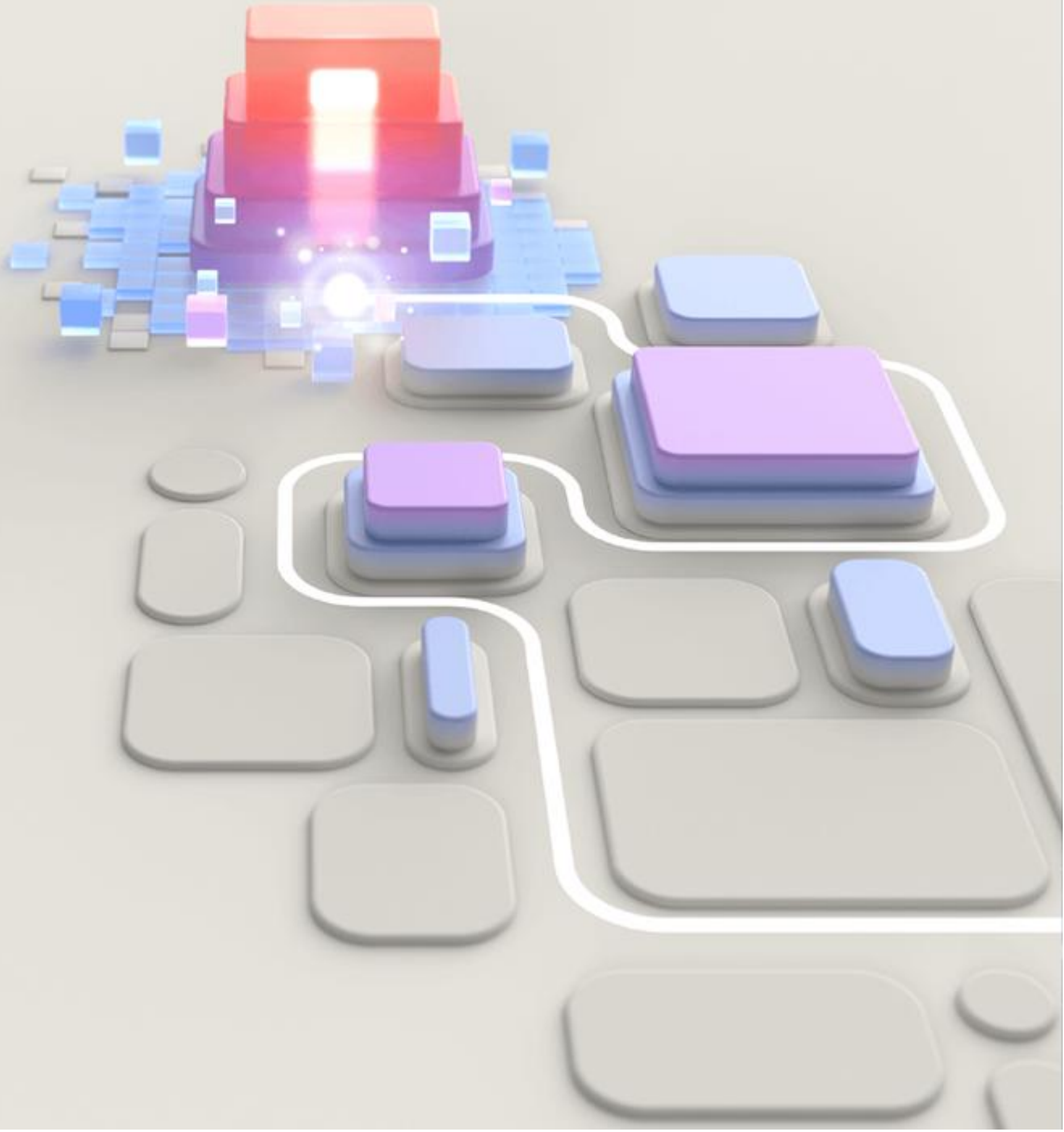
Microsoft

AZ-104

Administer PaaS Compute Options



AZ-104 Course Outline

01: Administer Identity

02: Administer Governance and Compliance

03: Administer Azure Resources

04: Administer Virtual Networking

05: Administer Intersite Connectivity

06: Administer Network Traffic Management

07: Administer Azure Storage

08: Administer Azure Virtual Machines

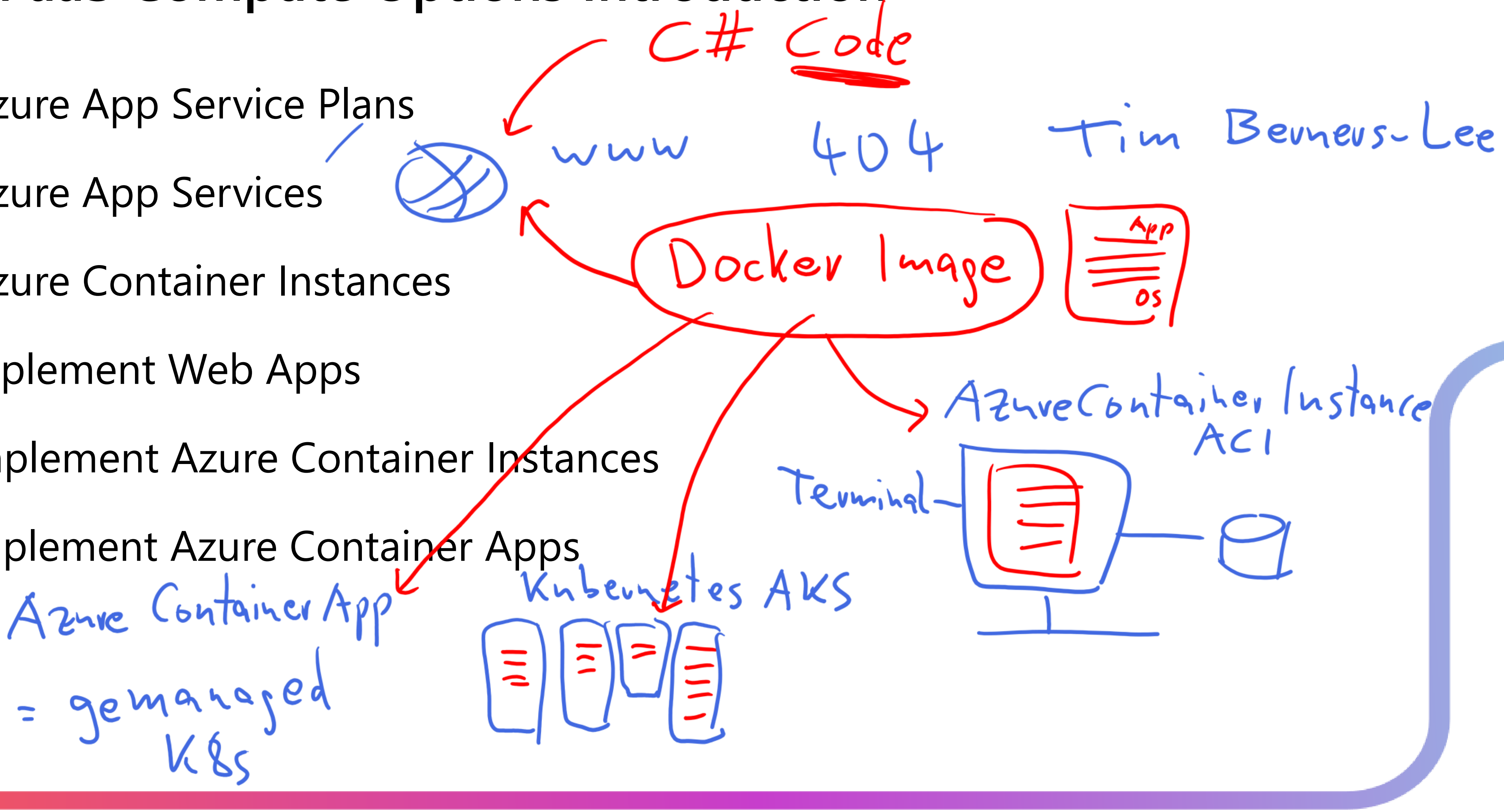
09: Administer PaaS Compute Options ←

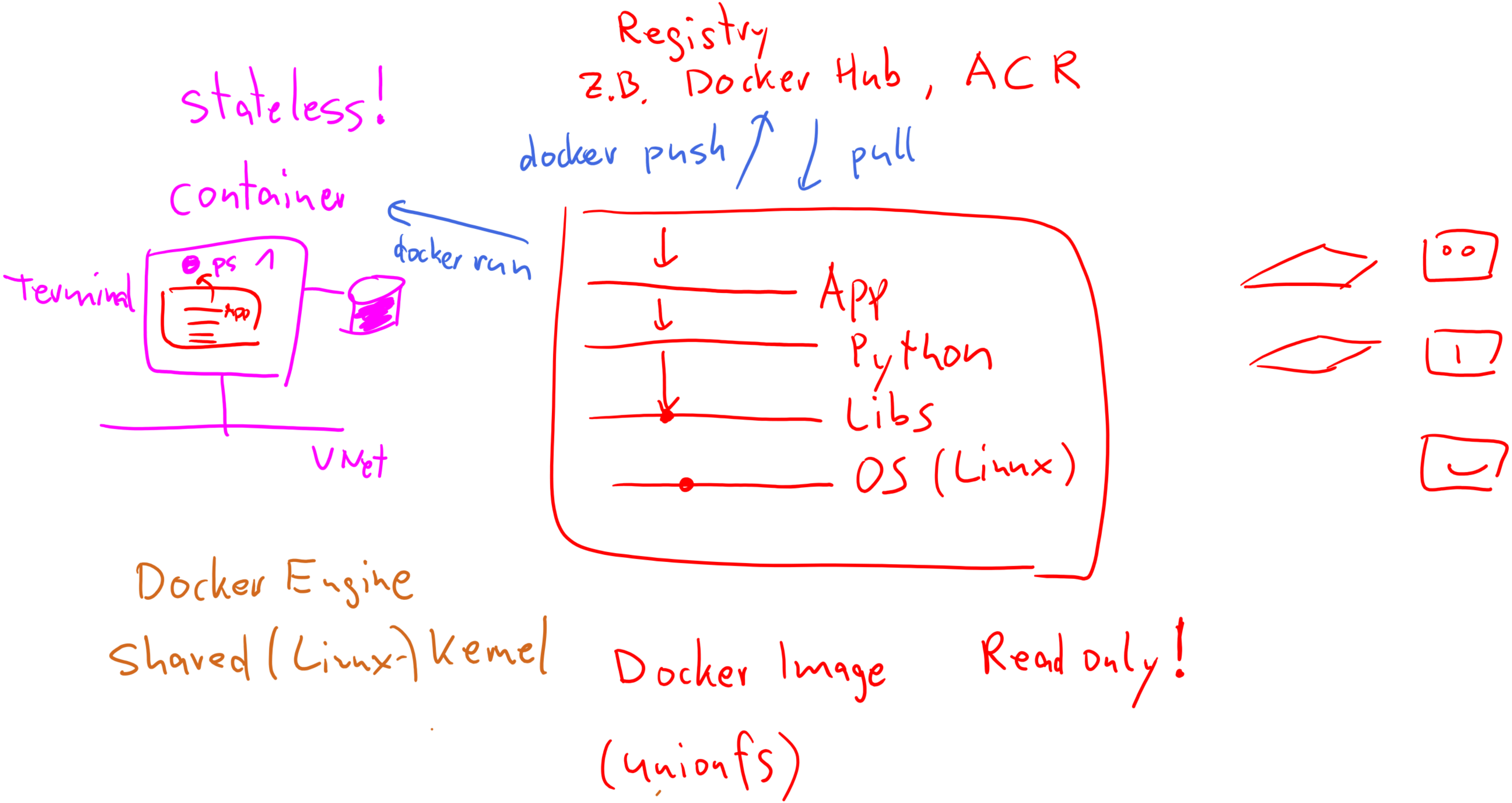
10: Administer Data Protection

11: Administer Monitoring

Administer PaaS Compute Options Introduction

- Configure Azure App Service Plans
- Configure Azure App Services
- Configure Azure Container Instances
- Lab 09a - Implement Web Apps
- Lab 09b - Implement Azure Container Instances
- Lab 09c – Implement Azure Container Apps




Registry
z.B. Docker Hub, ACR
docker push
pull
stateless!
Container
ps 1
App
Terminal
docker run
VNet
App
Python
Libs
OS (Linux)
Docker Engine
Shared (Linux) Kernel
Docker Image
(unionfs)
Read only!

Configure Azure App Service Plans

Implement Azure App Service Plans

- Determines performance, price, and features
- Defines a set of compute resources for a web app to run
  - Region where compute resources will be created
  - Number of virtual machine instances
  - Size of virtual machine instances
  - Pricing tier (next slide)
- One or more apps can be configured to run in the same App Service plan

Determine App Service Plan Pricing

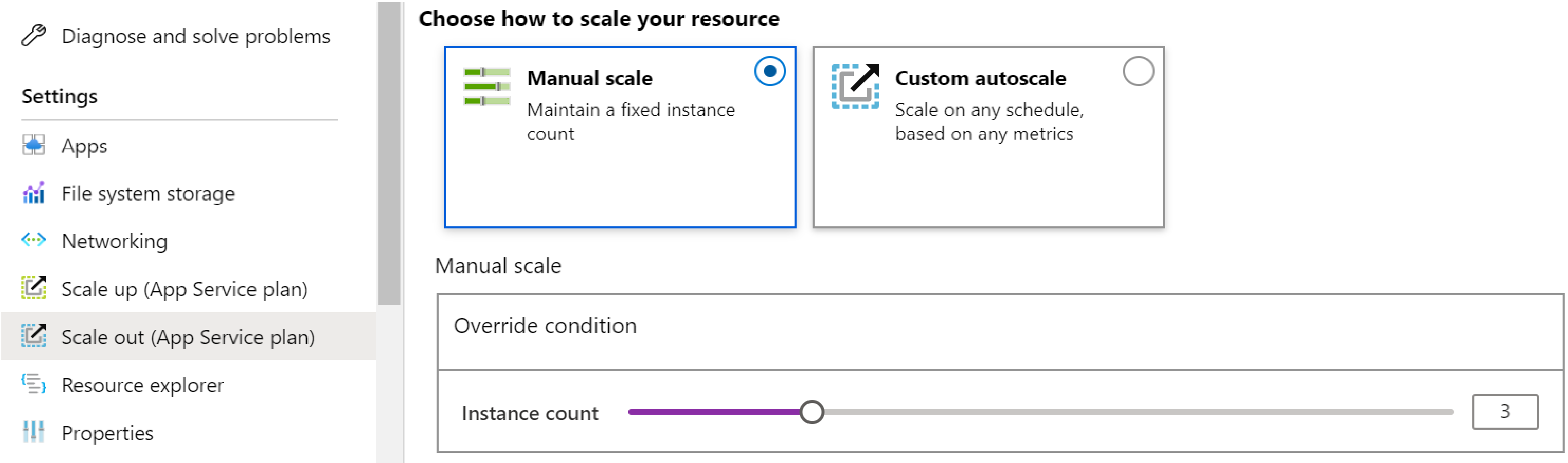
Selected Features	Free	Shared (dev/test)	Basic (dedicated dev/test)	Standard (production workloads)	Premium (enhanced scale and performance)	Isolated (high-performance, security and isolation)
Web, mobile, or API apps	10	100	Unlimited	Unlimited	Unlimited	Unlimited
Disk space	1 GB	1 GB	10 GB	50 GB	250 GB	1 TB
Auto Scale	–	–	–	Supported	Supported	Supported
Deployment Slots	0	0	0	5	20	20
Max Instances	–	–	Up to 3	Up to 10	Up to 30	Up to 100

Shared compute (Free and Shared). Run apps on the same Azure VM as other App Service apps, and the resources cannot scale out

Dedicated compute (Basic, Standard, Premium). Run apps in the same plan in dedicated Azure VMs

Isolated. Runs apps on dedicated Azure VMs in dedicated Azure virtual networks

Scale Up and Scale Out the App Service Plan


Diagnose and solve problems
Settings
Apps
File system storage
Networking
Scale up (App Service plan)
Scale out (App Service plan)
Resource explorer
Properties

Choose how to scale your resource

● **Manual scale**
Maintain a fixed instance count

○ **Custom autoscale**
Scale on any schedule, based on any metrics

Manual scale

Override condition

Instance count: 3

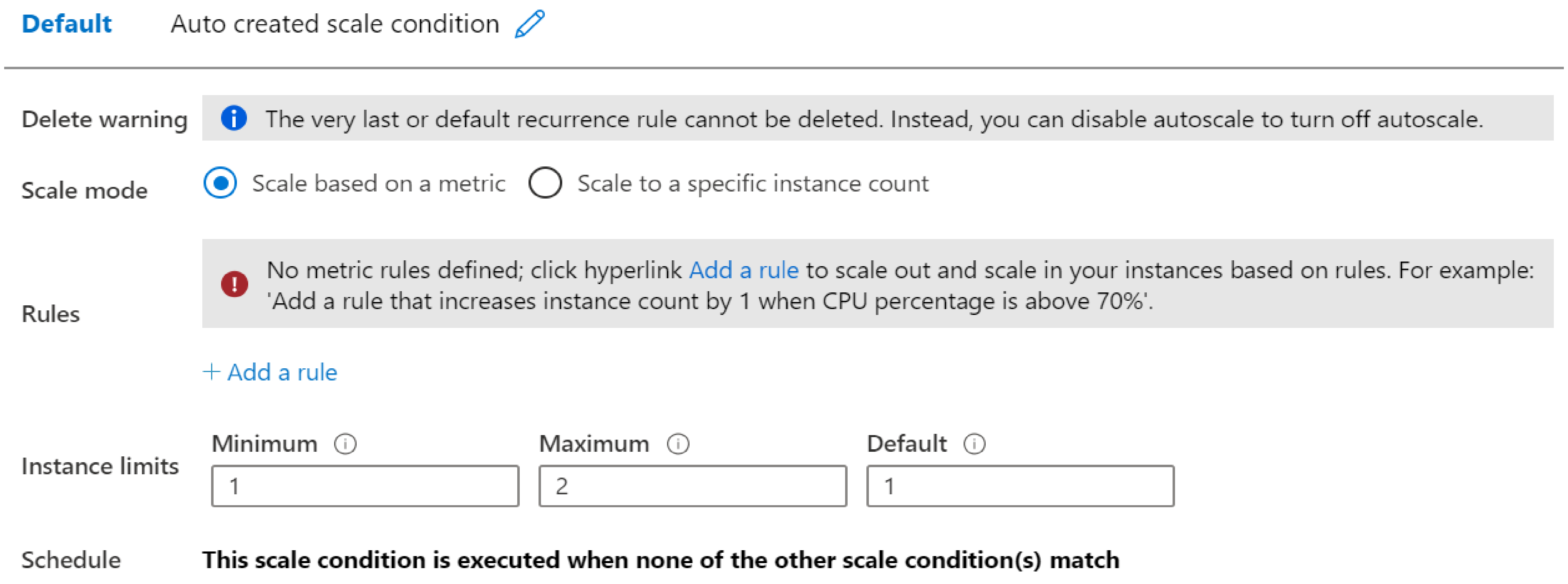
Scale up (change the App Service plan):

- More hardware (CPU, memory, disk)
- More features (dedicated virtual machines, staging slots, autoscaling)

Scale out (increase the number of VM instances):

- Manual (fixed number of instances)
- Auto scale (based on predefined rules and schedules)

Configure App Service Plan Scaling



Delete warning: The very last or default recurrence rule cannot be deleted. Instead, you can disable autoscale to turn off autoscale.

Scale mode: ● Scale based on a metric ○ Scale to a specific instance count

Rules: No metric rules defined; click hyperlink Add a rule to scale out and scale in your instances based on rules. For example: 'Add a rule that increases instance count by 1 when CPU percentage is above 70%'.

+ Add a rule

Instance limits: Minimum 1 Maximum 2 Default 1

Schedule: **This scale condition is executed when none of the other scale condition(s) match**

Adjust available resources based on the current demand

Improves availability and fault tolerance

Scale based on a metric (CPU percentage, memory percentage, HTTP requests)

Scale according to a schedule (weekdays, weekends, times, holidays)

Can implement multiple rules – combine metrics and schedules

Don’t forget to scale in

Learning Recap – Configure Azure App Service Plans

Check your knowledge questions and additional study

- Scale an App Service web app to efficiently meet demand with App Service scale up and scale out

Configure Azure App Services

Implement Azure App Service

.NET Node.js PHP Java Python (on Linux) HTML Custom Windows/Linux Container

- Includes Web Apps, API Apps, Mobile Apps, and Function Apps
- Fully managed environment enabling high productivity development
- Platform-as-a-service (PaaS) offering for building and deploying highly available cloud apps for web and mobile
- Platform handles infrastructure so developers focus on core web apps and services
- Developer productivity using .NET, .NET Core, Java, Python and a host of others
- Provides enterprise-grade security and compliance

Create an App Service

Name must be unique

Access using *azurewebsites.net* – can map to a custom domain

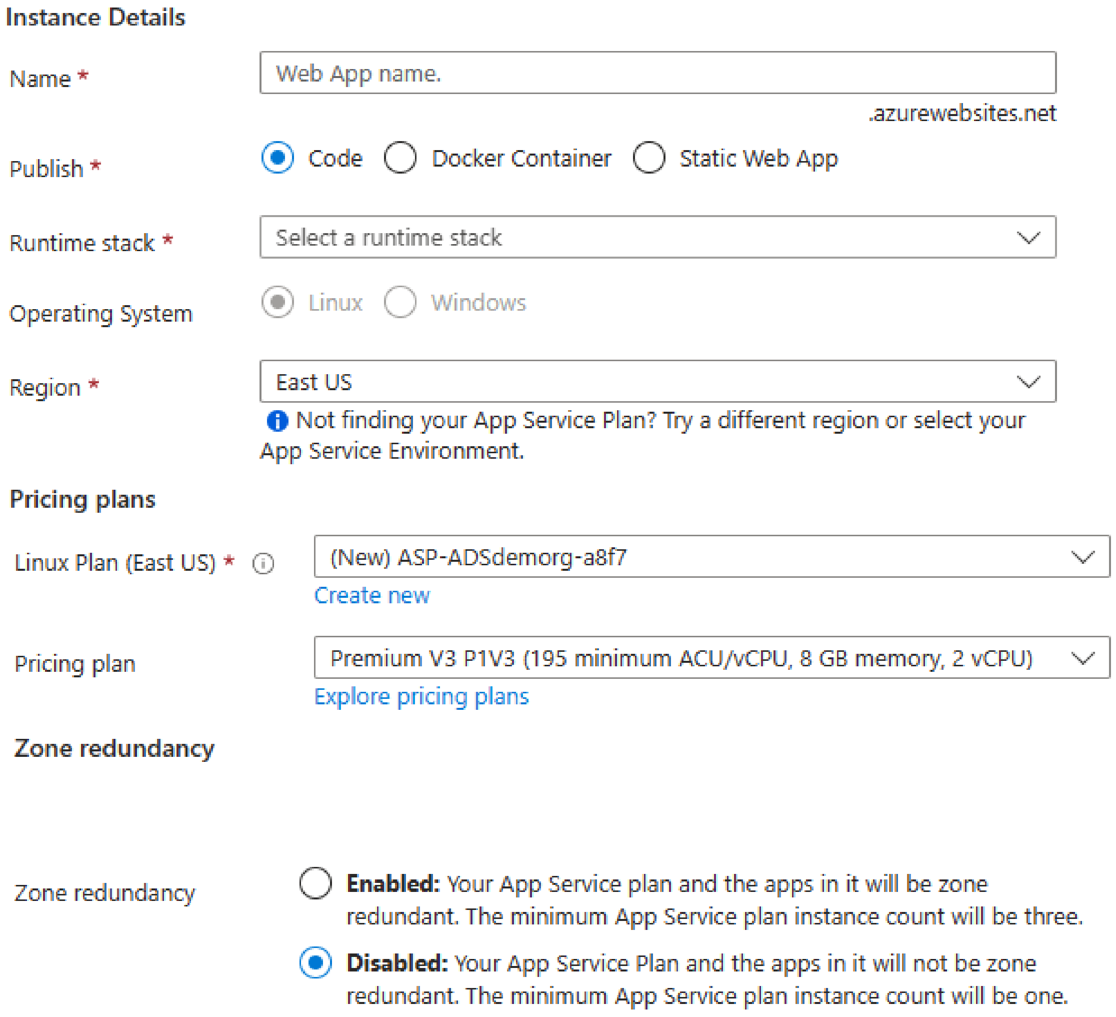
Publish Code (Runtime Stack)

Publish Docker Container

Linux or Windows

Region closest to your users

App Service Plan



Instance Details

Name * Web App name.
.azurewebsites.net

Publish * ● Code ○ Docker Container ○ Static Web App

Runtime stack * Select a runtime stack

Operating System ● Linux ○ Windows

Region * East US

Not finding your App Service Plan? Try a different region or select your App Service Environment.

Pricing plans

Linux Plan (East US) * (New) ASP-ADSdemorg-a8f7

Pricing plan Premium V3 P1V3 (195 minimum ACU/vCPU, 8 GB memory, 2 vCPU)

Zone redundancy

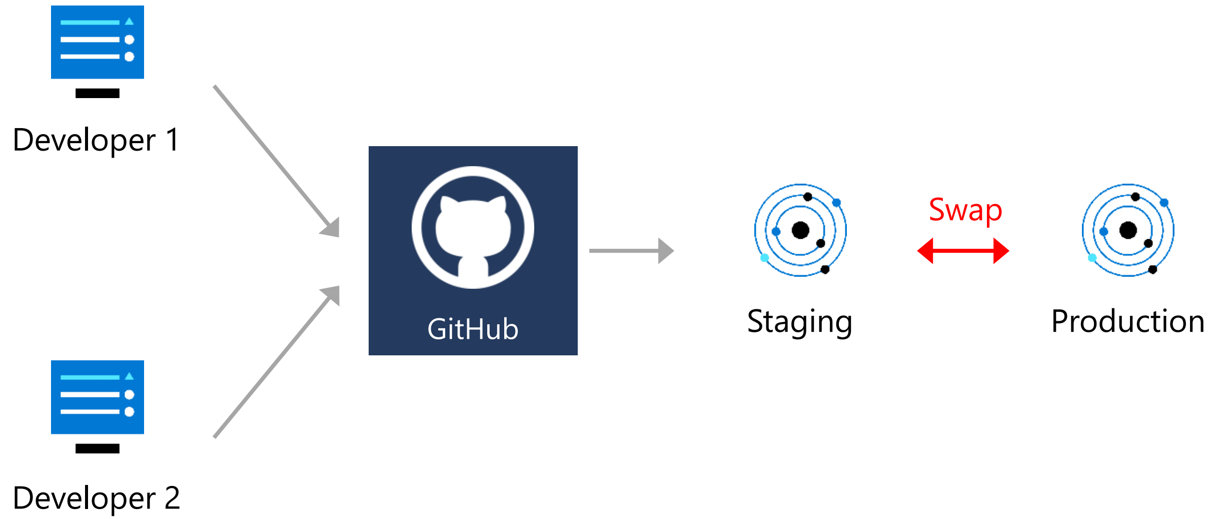
Zone redundancy

○ **Enabled:** Your App Service plan and the apps in it will be zone redundant. The minimum App Service plan instance count will be three.

● **Disabled:** Your App Service Plan and the apps in it will not be zone redundant. The minimum App Service plan instance count will be one.

Create Deployment Slots

Continuous Deployment with Stage Slot



Service Plan	Slots
Free, Shared, Basic	0
Standard	Up to 5
Premium	Up to 20
Isolated	Up to 20

- Deploy to a different deployment slots (depends on service plan)
- Validate changes before sending to production
- Deployment slots are live apps with their own hostnames
- Avoids a cold start – eliminates downtime
- Fallback to a last known good site
- Auto Swap when pre-swap validation is not needed

Add Deployment Slots

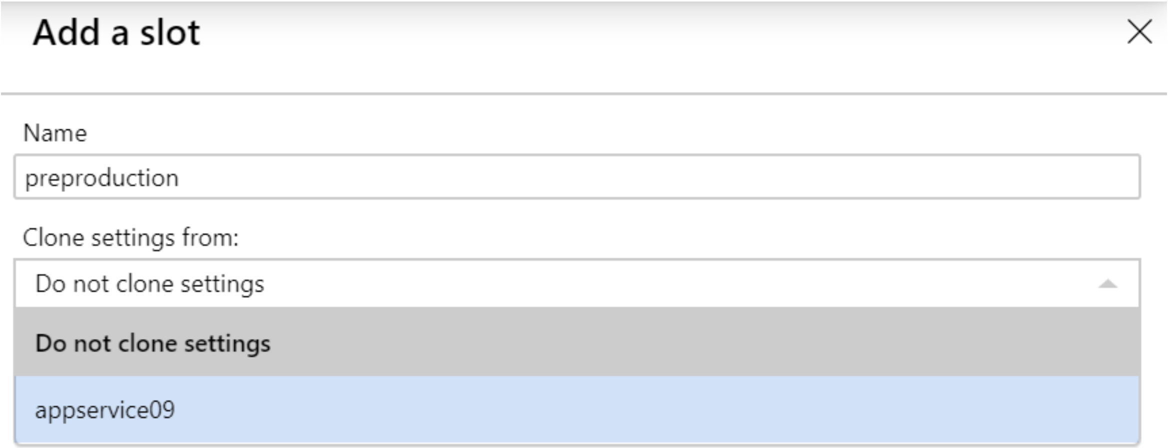
Select whether to clone an app configuration from another deployment slot

When you clone, pay attention to the settings:

- Slot-specific app settings and connection strings
- Continuous deployment settings
- App Service authentication settings

Not all settings are sticky (endpoints, custom domain names, SSL certificates, scaling)

Review and edit your settings before swapping



Add a slot

Name: preproduction

Clone settings from: Do not clone settings

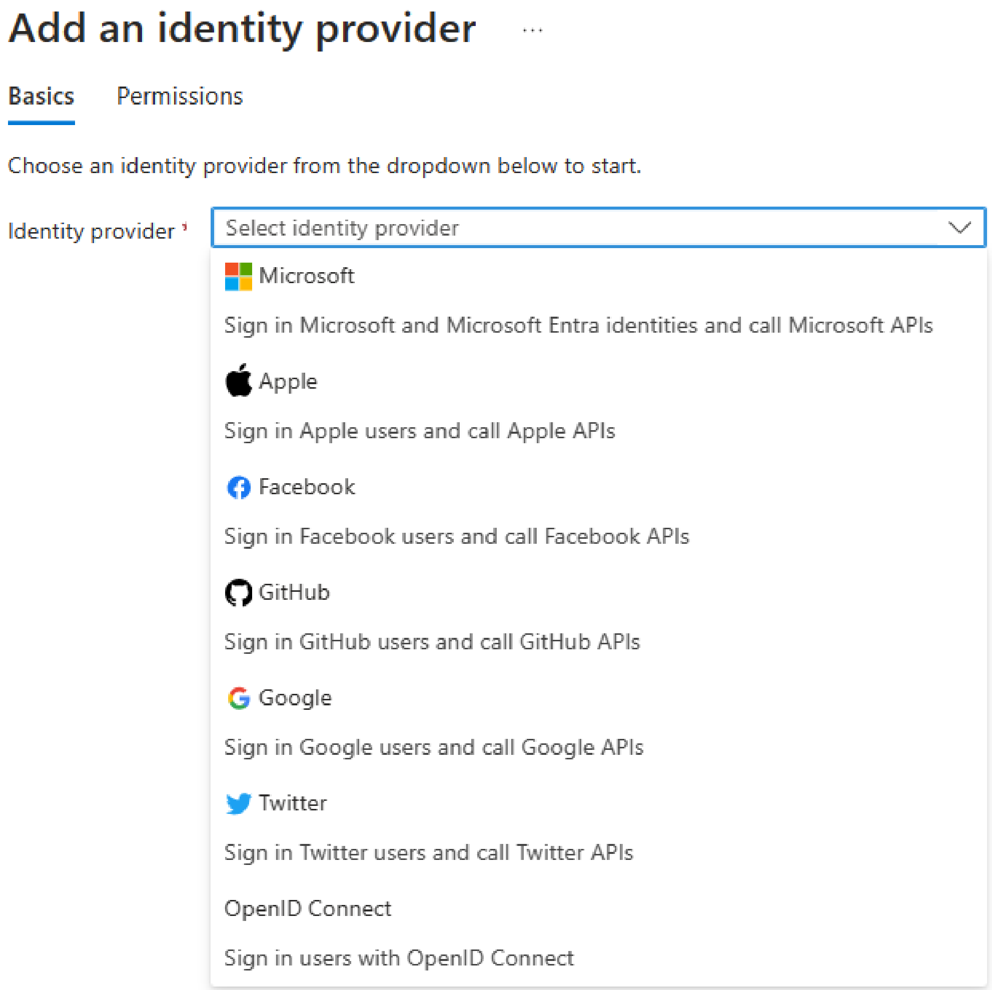
Secure an App Service

Authentication:

- Enable authentication – default anonymous
- Log in with a third-party identity provider

Security:

- Troubleshoot with Diagnostic Logs – failed requests, app logging
- Add an SSL certificate – HTTPS
- Define a priority ordered allow/deny list to control network access to the app
- Store secrets in the Azure Key Vault



Add an identity provider

Choose an identity provider from the dropdown below to start.

Identity provider * Select identity provider

Microsoft
Sign in Microsoft and Microsoft Entra identities and call Microsoft APIs

Apple
Sign in Apple users and call Apple APIs

Facebook
Sign in Facebook users and call Facebook APIs

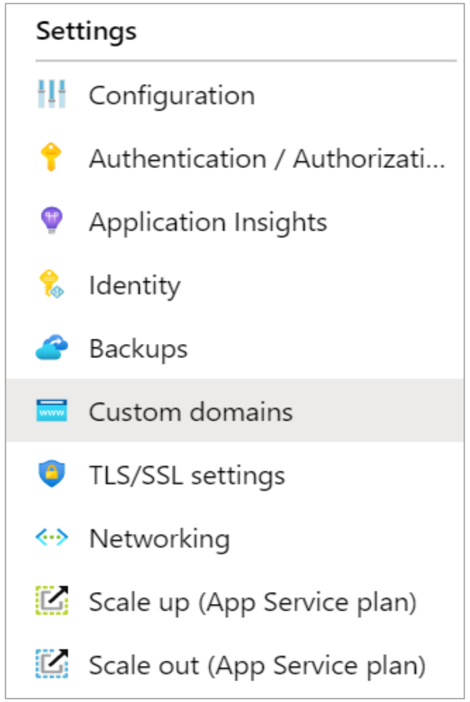
GitHub
Sign in GitHub users and call GitHub APIs

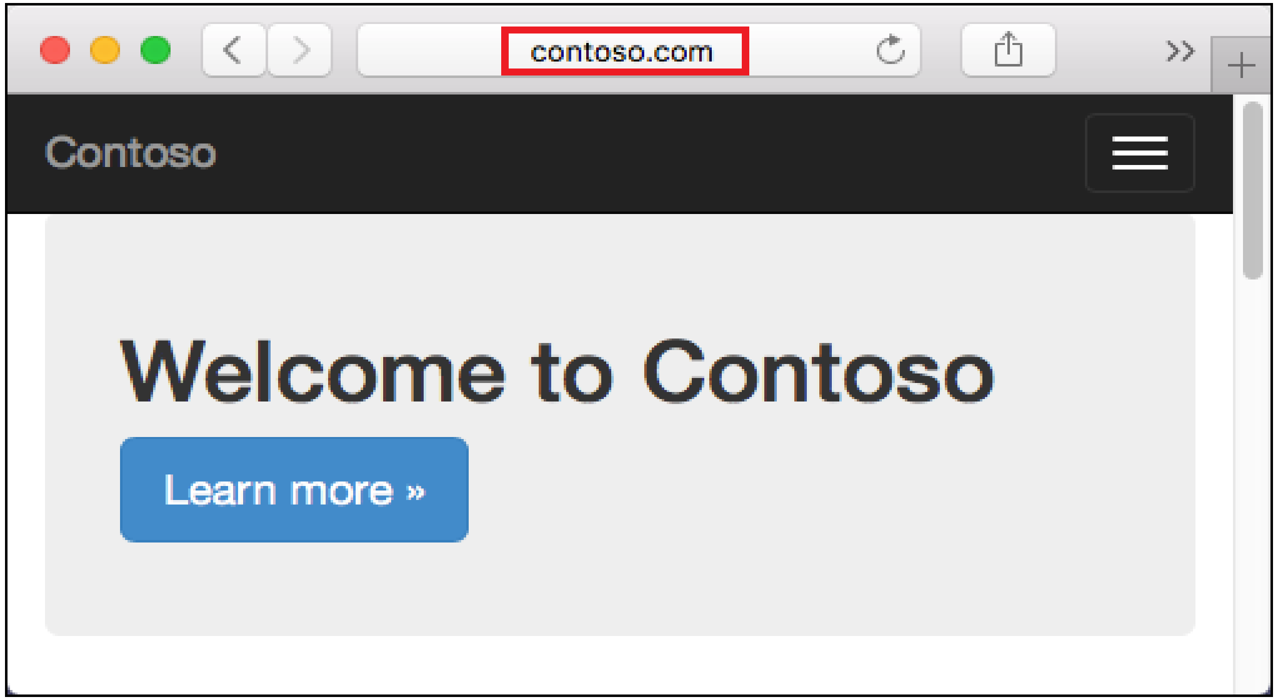
Google
Sign in Google users and call Google APIs

Twitter
Sign in Twitter users and call Twitter APIs

OpenID Connect
Sign in users with OpenID Connect

Create Custom Domain Names





Redirect the default web app URL

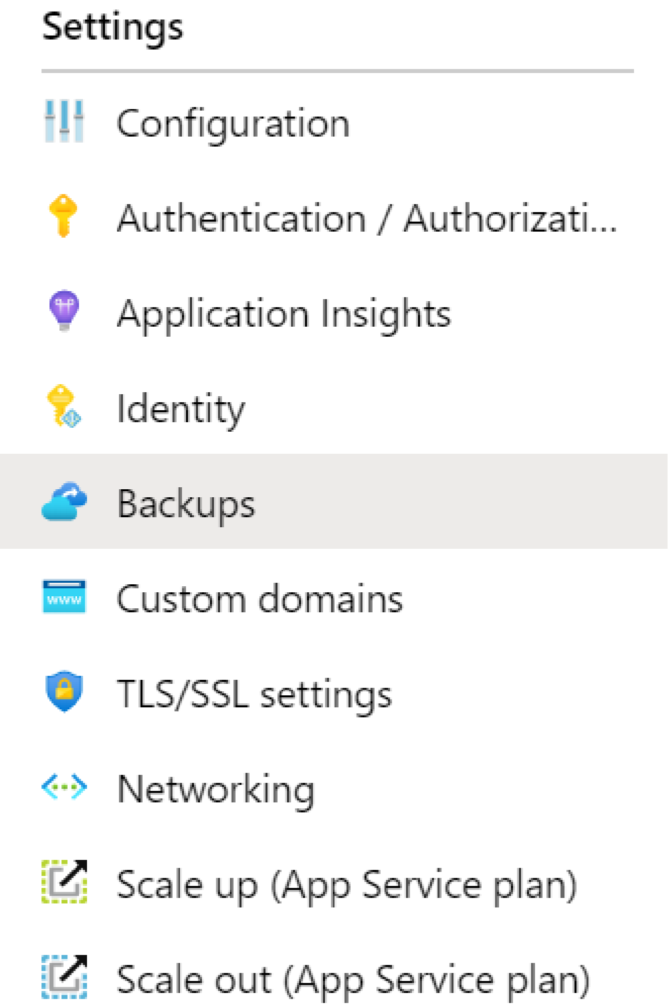
Validate the custom domain in Azure

Use the DNS registry for your domain provider – create a CNAME or A record with the mapping

Ensure App Service plan supports custom domains

Backup an App Service

- Create app backups manually or on a schedule
- Backup the configuration, file content, and database connected to the app
- Requires Standard or Premium plan
- Backups can be up to 10 GB of app and database content
- Configure partial backups and exclude items from the backup
- Restore your app on-demand to a previous state, or create a new app



Learning Recap – Configure Azure App Services

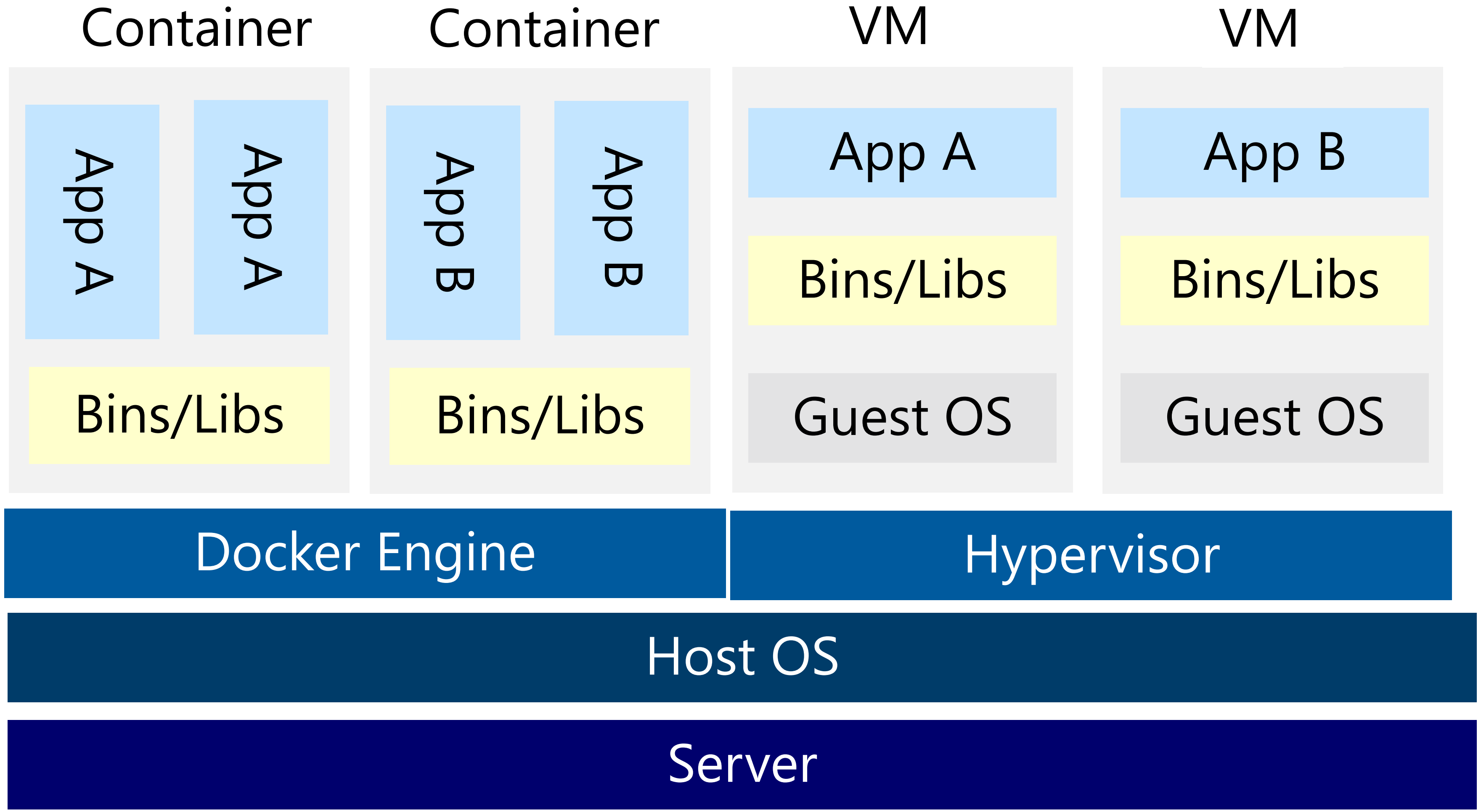
Check your knowledge questions and additional study

- Host a web application with Azure App Service
- Stage a web app deployment for testing and rollback by using App Service deployment slots
- Dynamically meet changing web app performance requirements with autoscale rules

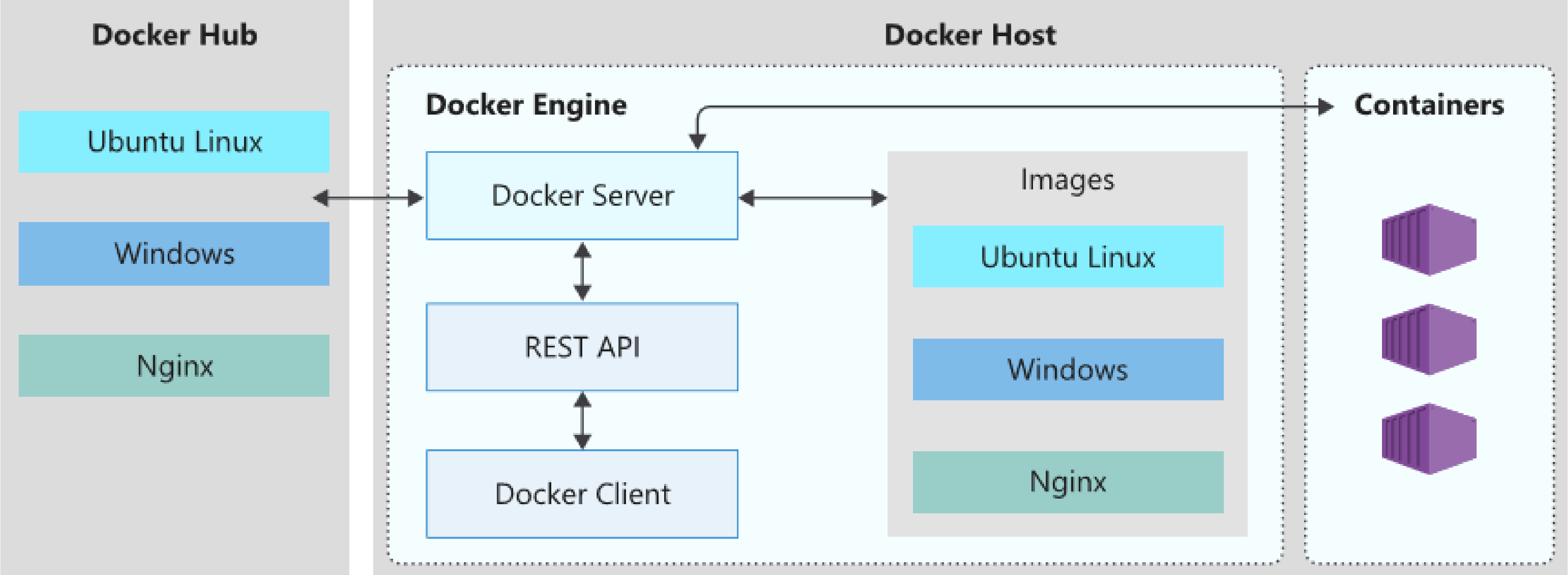
Configure Azure Container Instances

Compare Containers to Virtual Machines

- Isolation
- Operating System
- Deployment
- Persistent storage
- Fault tolerance



Understand the Docker Platform (optional)



Enables developers to host applications within a container

A container is a standardized “unit of software“ that contains everything required for an application to run

Available on both Linux and Windows and can be hosted on Azure

Understand Container Images

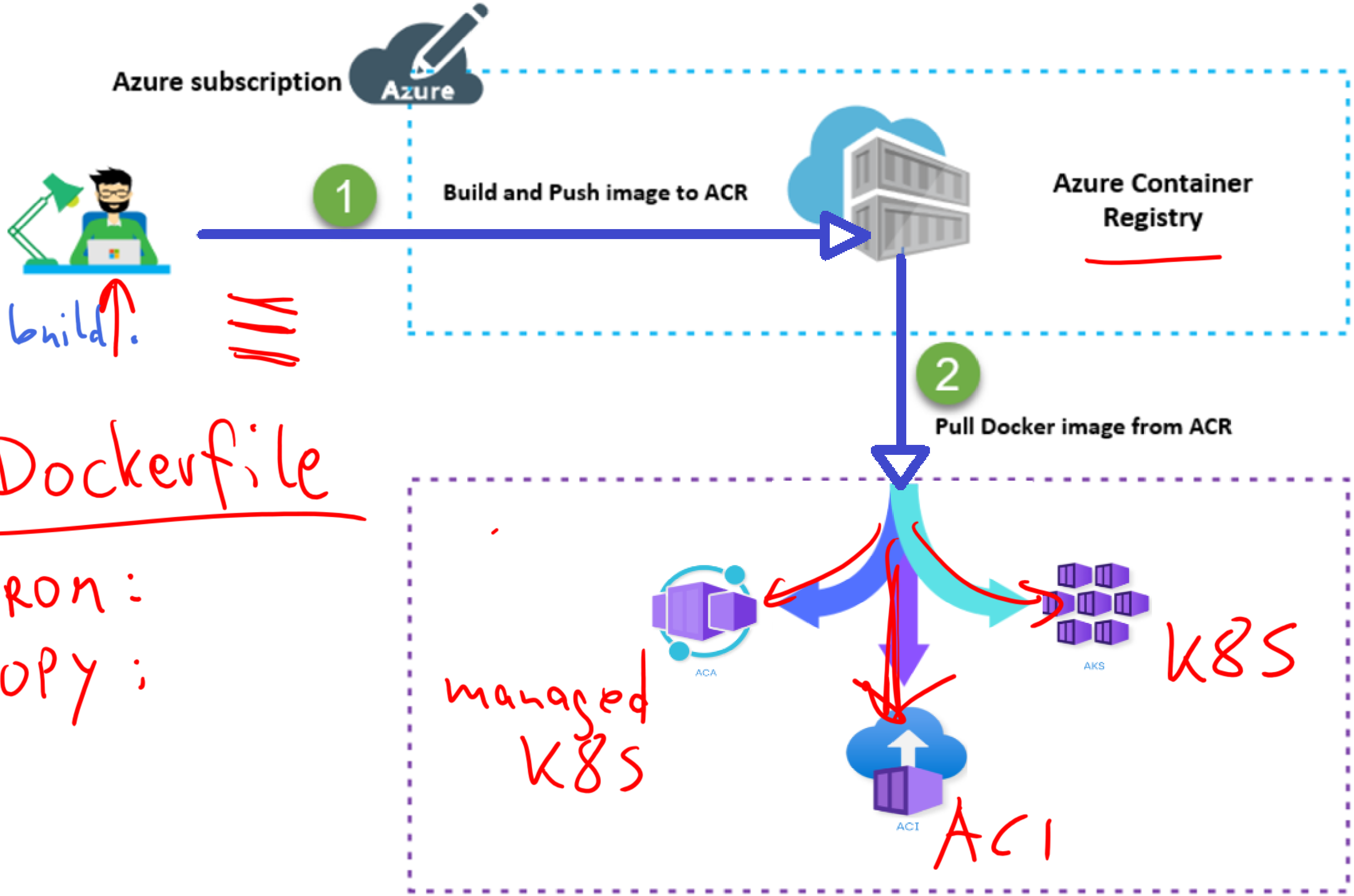
A container image is a lightweight, standalone, executable package of software that encapsulates everything needed to run an application.

docker build .

Dockerfile

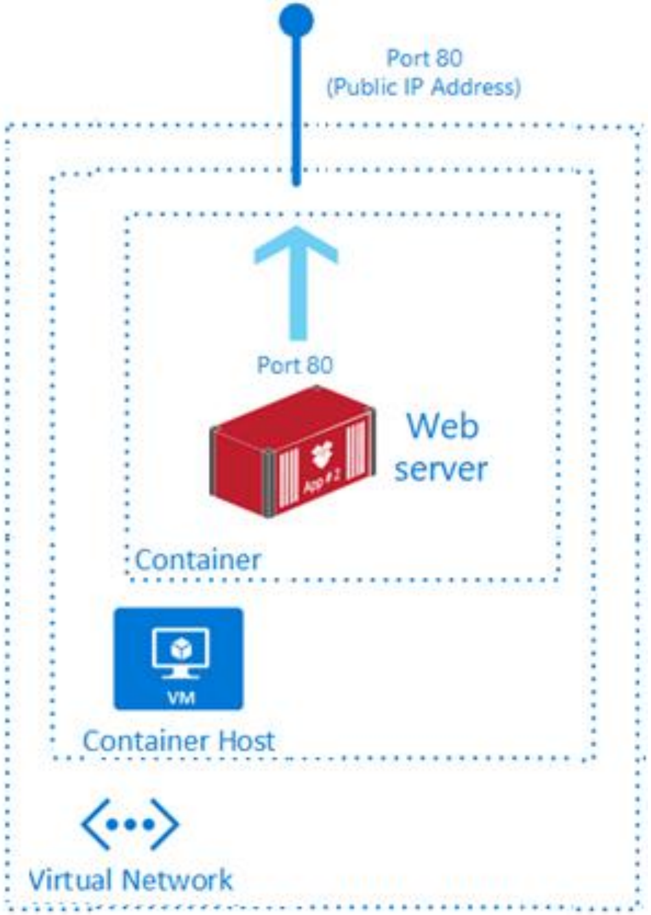
FROM :

COPY ;



Review Azure Container Instances

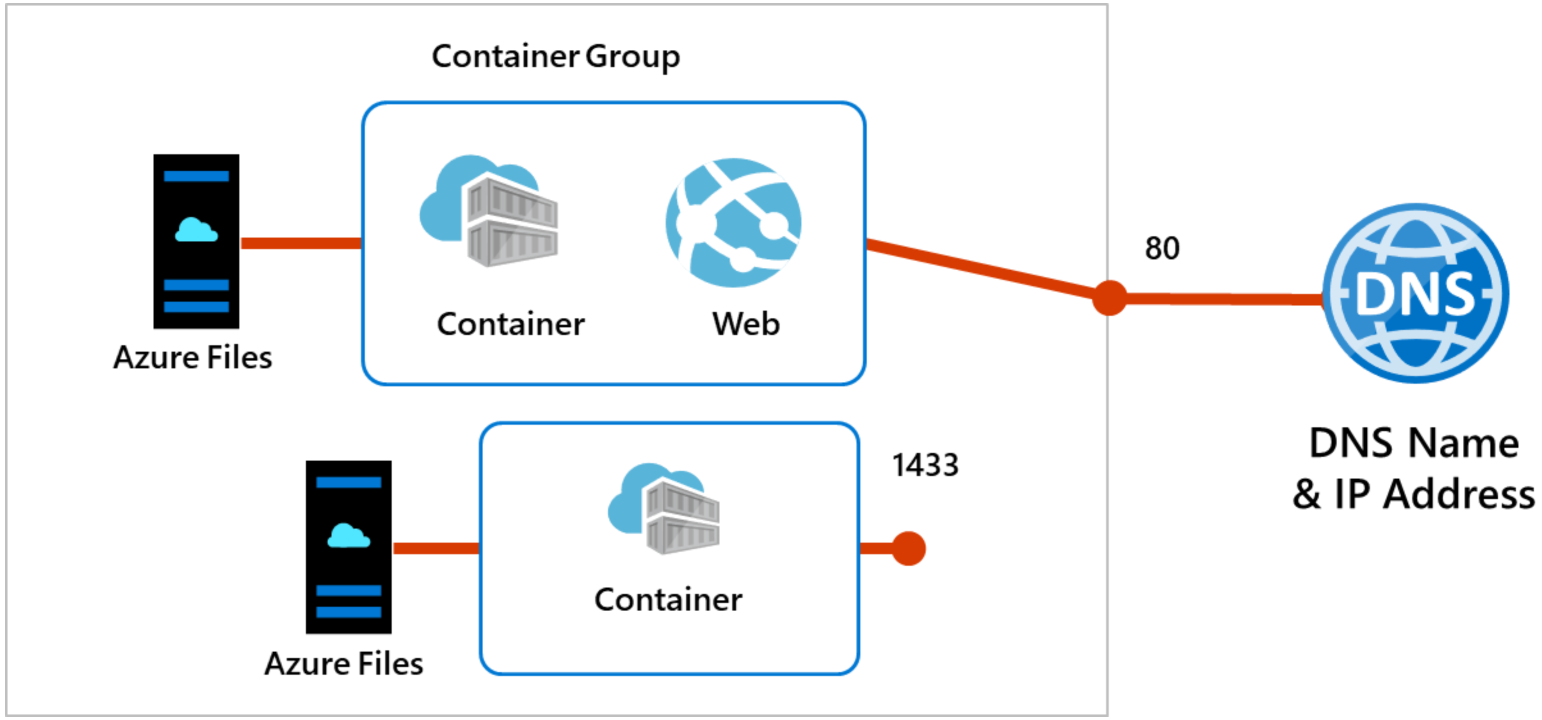
- PaaS Service
- Fast startup times
- Public IP connectivity and DNS name
- Isolation features
- Custom sizes
- Persistent storage
- Linux and Windows Containers
- Co-scheduled Groups
- Virtual network Deployment



Fastest way to run a container in Azure without provisioning a VM

Implement Container Groups

- Top-level resource in Azure Container Instances
- A collection of containers that get scheduled on the same host
- The containers in the group share a lifecycle, resources, local network, and storage volumes

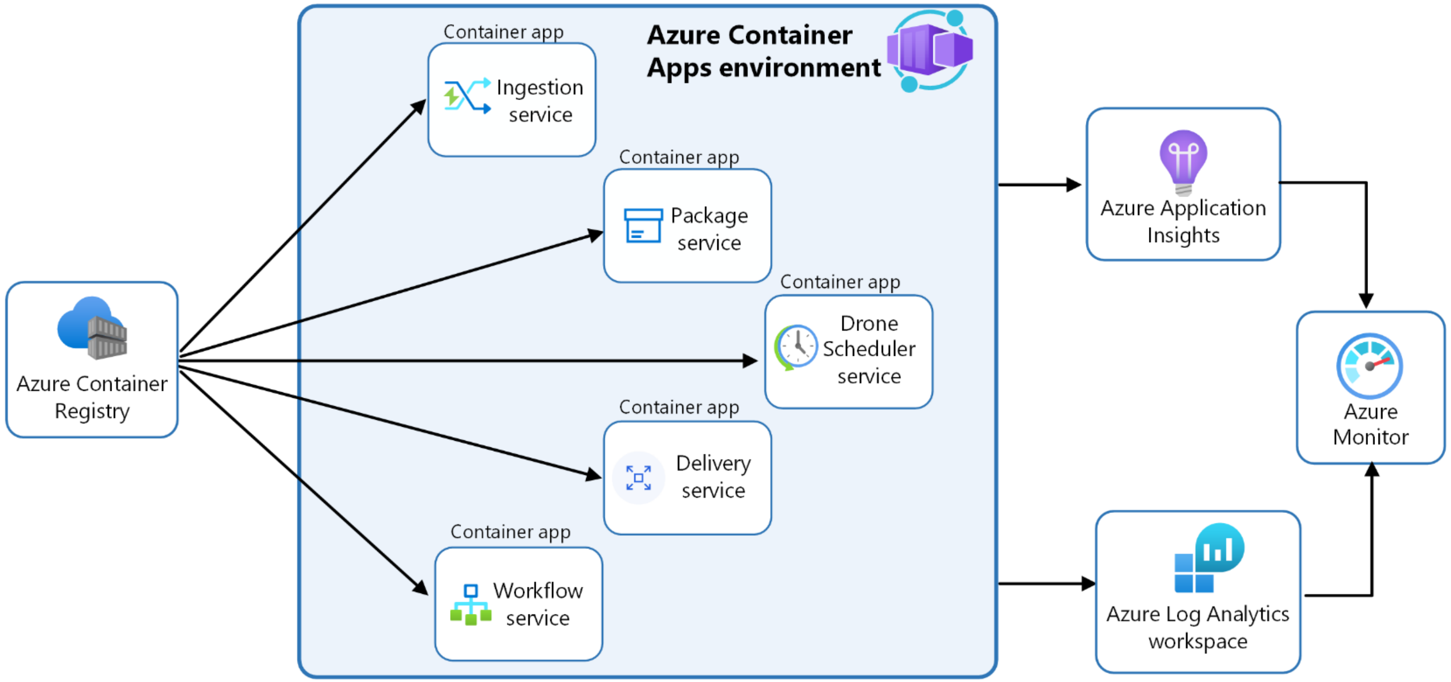


Compare container management solutions

	Azure Container Apps	Azure Kubernetes Service
Overview	Simplifies the deployment and management of microservices-based applications by abstracting away the underlying infrastructure.	Simplifies deploying a managed Kubernetes cluster in Azure by offloading the operational overhead to Azure.
Deployment	PaaS experience.	Offers more control and customization.
Management	Fully managed by Azure.	Partially managed by Azure (control plane).
Scalability	HTTP-based autoscaling and event-driven scaling.	Horizontal pod autoscaling and cluster autoscaling.
Use Cases	Rapid scaling and simplified management.	Complex, long-running applications that require full Kubernetes features.
Integration	Azure Logic Apps, Functions, and Event Grid for event-driven architecture.	Azure Policy for Kubernetes, Azure Monitor for containers, and Azure Defender for Kubernetes for comprehensive security and governance.

Manage Containers with Azure Container Apps

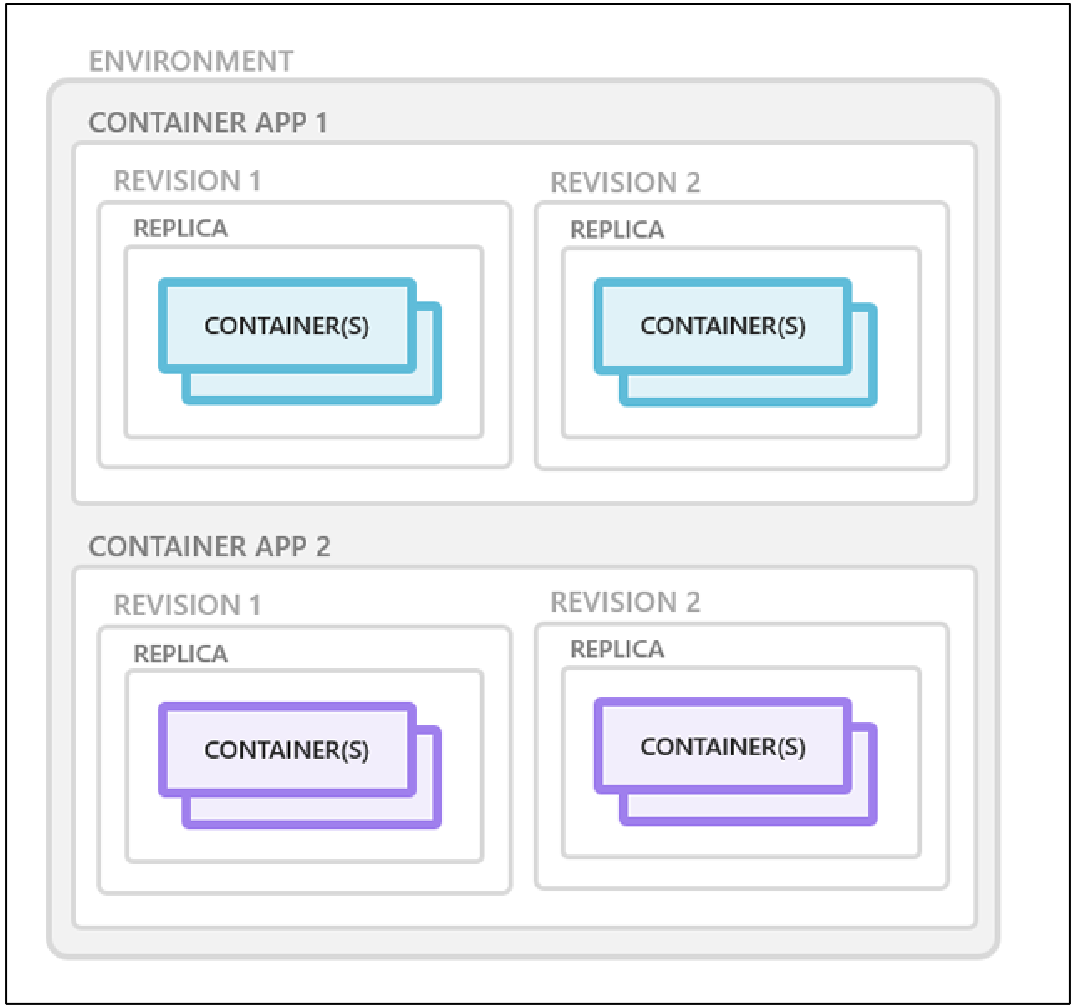
- Alternative to Azure Kubernetes Service – manages container orchestration
- The Container App environment creates a secure boundary around the apps and jobs
- The Container App runtime manages the environment (OS upgrades, scaling, versioning, and failover)



Microsoft Azure

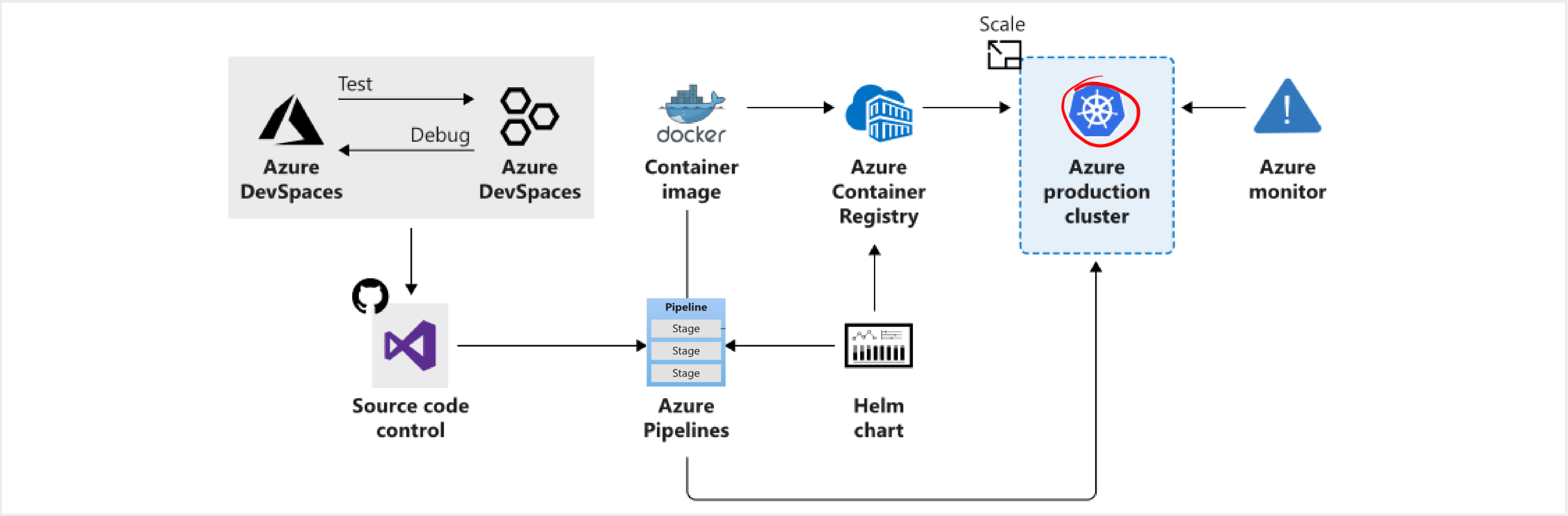
Explore containers in Azure Container Apps

- Containers for an Azure Container App are grouped together in pods inside revision snapshots.
- Can define multiple containers in a single container app to implement the sidecar pattern.
- Deploy images hosted on private registries by providing credentials in the Container Apps configuration.



© Copyright Microsoft Corporation. All rights reserved.

Azure Kubernetes Service



Manages health monitoring and maintenance

Performs simple cluster scaling

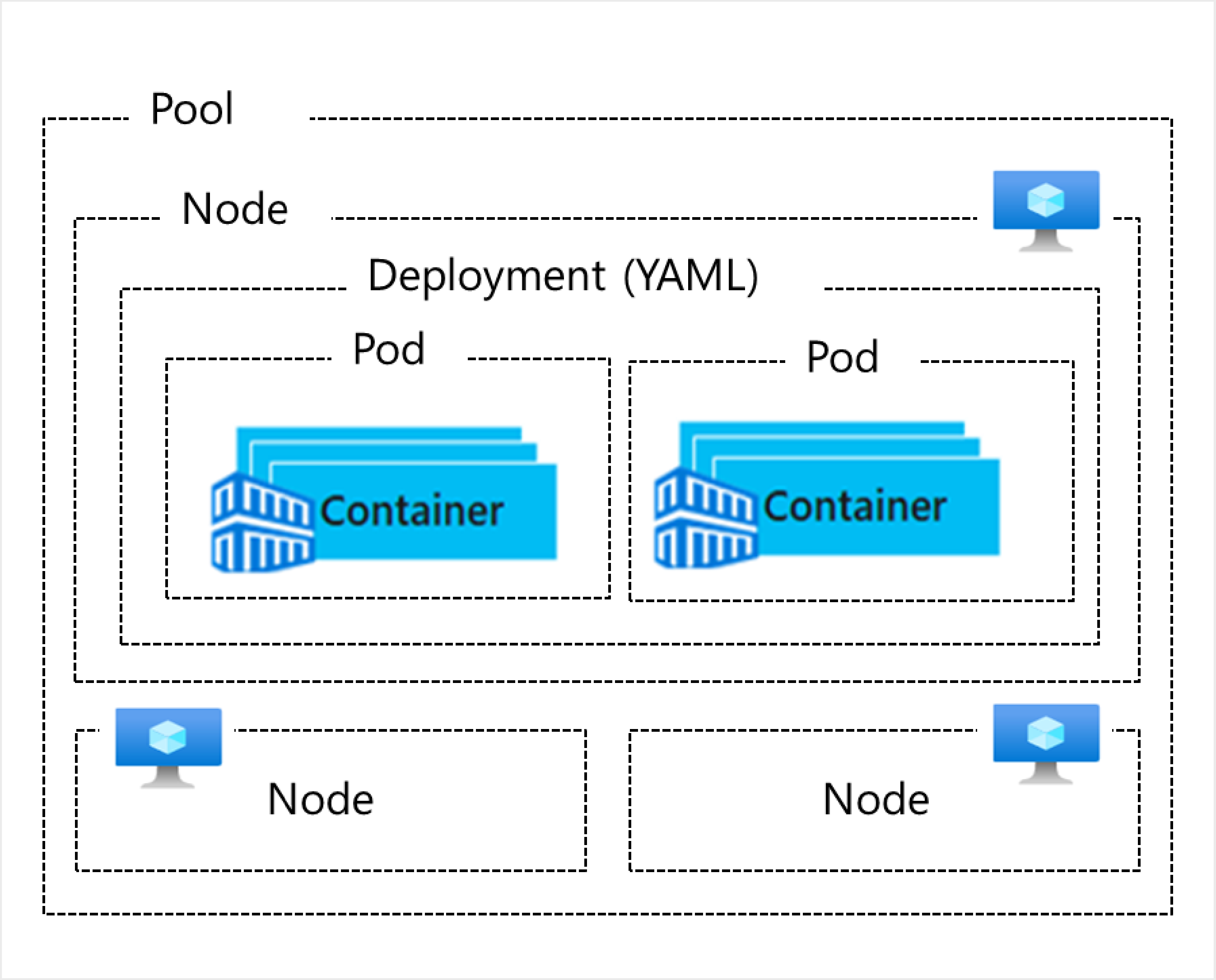
Enables nodes to be fully managed by Microsoft

You’re responsible only for managing the agent nodes

You pay only for the agent nodes

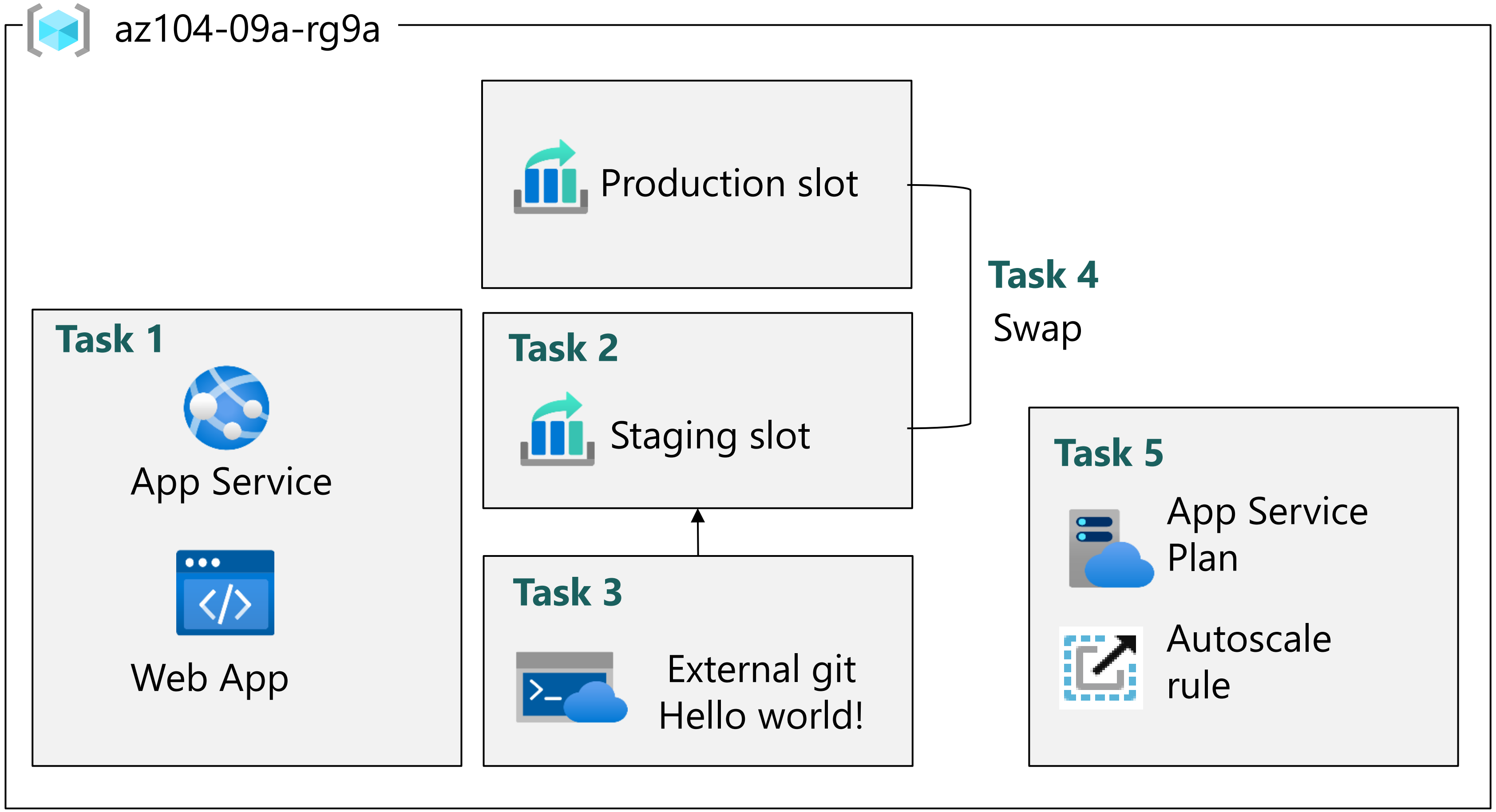
Understand AKS Terminology

Term	Description
Pools	Groups of nodes with identical configurations
Nodes	Individual VMs running containerized applications
Pods	Single instance of an application. A pod can contain multiple containers
Deployment	One or more identical pods managed by Kubernetes
Manifest	YAML file describing a deployment



Lab 09a – Implement Web Apps
Lab 09b – Implement Azure Container Instances
Lab 09c – Implement Azure Container Apps

Lab 09a – Web App Architecture Diagram



Lab 09b – Implement Azure Container Instances

In this lab, you learn how to implement Azure Container Instances.

You learn to deploy an Azure Container Instance to display a Hello World app.

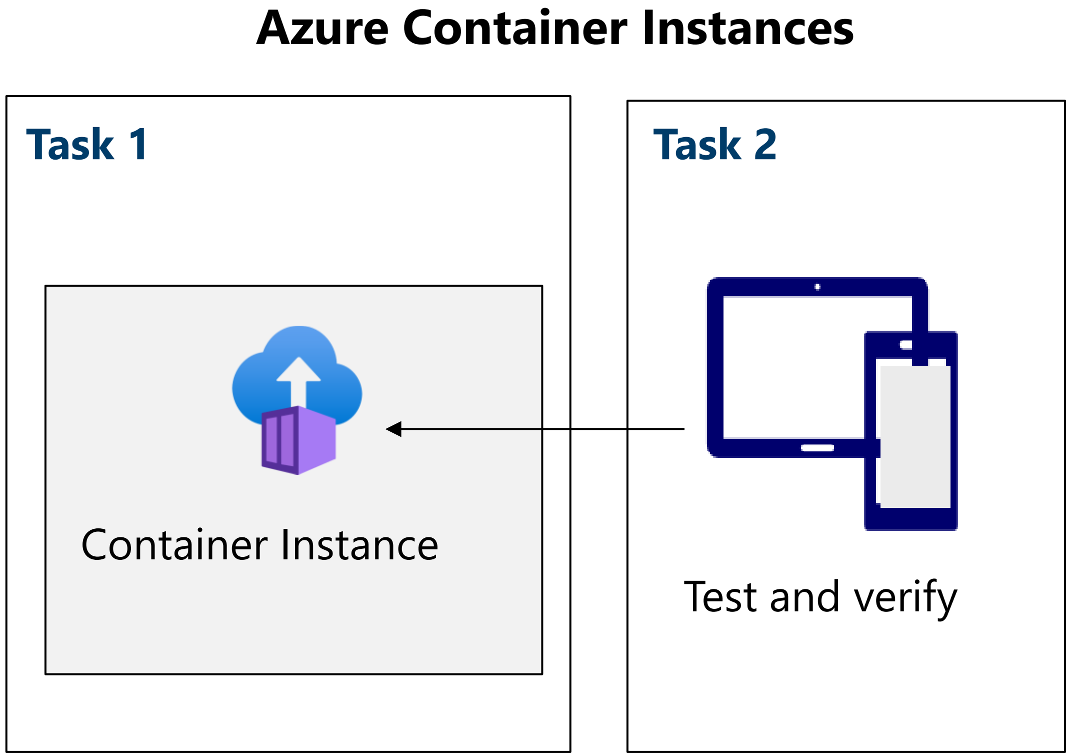
Job Skills

Task 1: Deploy an Azure Container Instance using a Docker image.

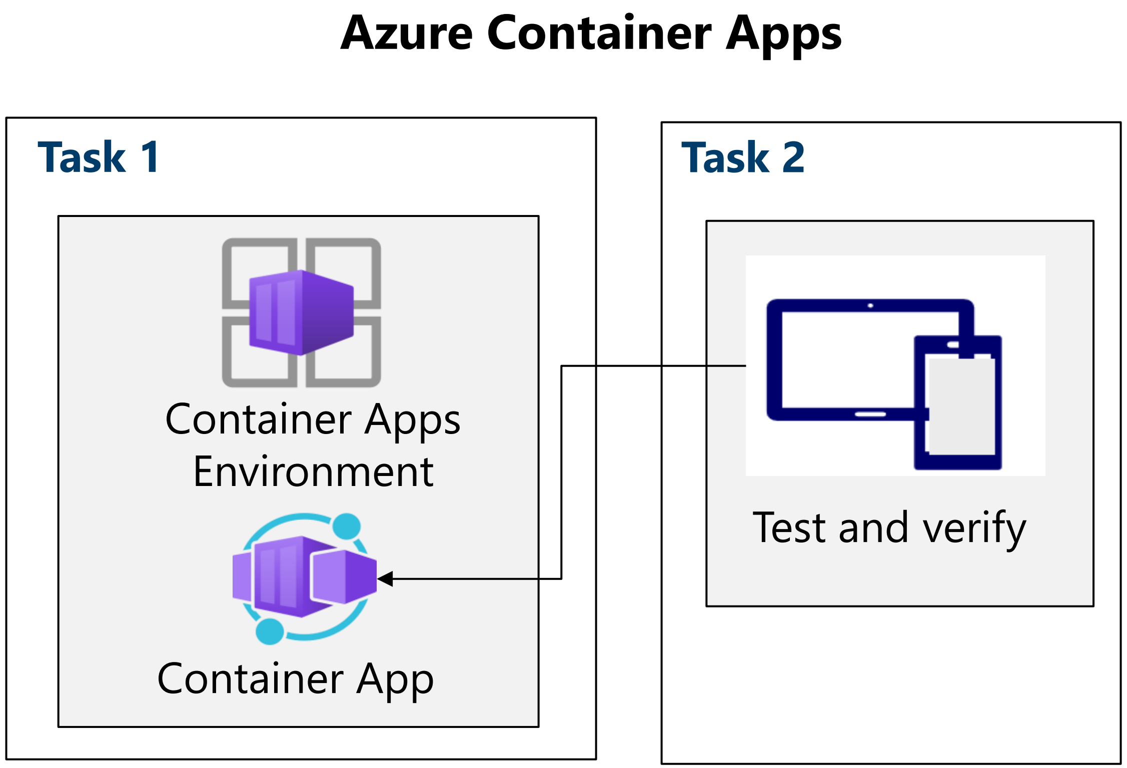
Task 2: Test and verify deployment of an Azure Container Instance.

Next slide for an architecture diagram →

Lab 09b – Azure Container Instances Diagram



Lab 09c – Azure Container Architecture Diagram



End of presentation