

# 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](#)

# Configure Azure App Service Plans



# Learning Objectives - Configure Azure App Service Plans

- Implement Azure App Service Plans
- Determine App Service Plan Pricing
- Scale Up and Scale Out the App Service Plan
- Configure App Service Plan Scaling
- Demonstration – Configure Azure App Service Plans
- Learning Recap

Implement and manage Azure compute resources (20-25%)

Create and configure Azure App Service

- Provision an App Service plan
- Configure scaling for an App Service plan

# 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

The screenshot shows the Azure App Service Scale settings. On the left is a sidebar with navigation options: 'Diagnose and solve problems', 'Settings', 'Apps', 'File system storage', 'Networking', 'Scale up (App Service plan)', 'Scale out (App Service plan)' (which is highlighted), 'Resource explorer', and 'Properties'. The main area is titled 'Choose how to scale your resource' and contains two options: 'Manual scale' (selected with a blue radio button) and 'Custom autoscale' (unselected). Below these, the 'Manual scale' section is expanded, showing an 'Override condition' field and an 'Instance count' slider. The slider is currently set to 3, with a numeric input box to its right.

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

**Default** Auto created scale condition 




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  | Maximum  | Default  |
|---|---|---|
| <input type="text" value="1"/>  | <input type="text" value="2"/>  | <input type="text" value="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

# 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

[i](#) 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

[Create new](#)

Pricing plan

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

[Explore pricing plans](#)

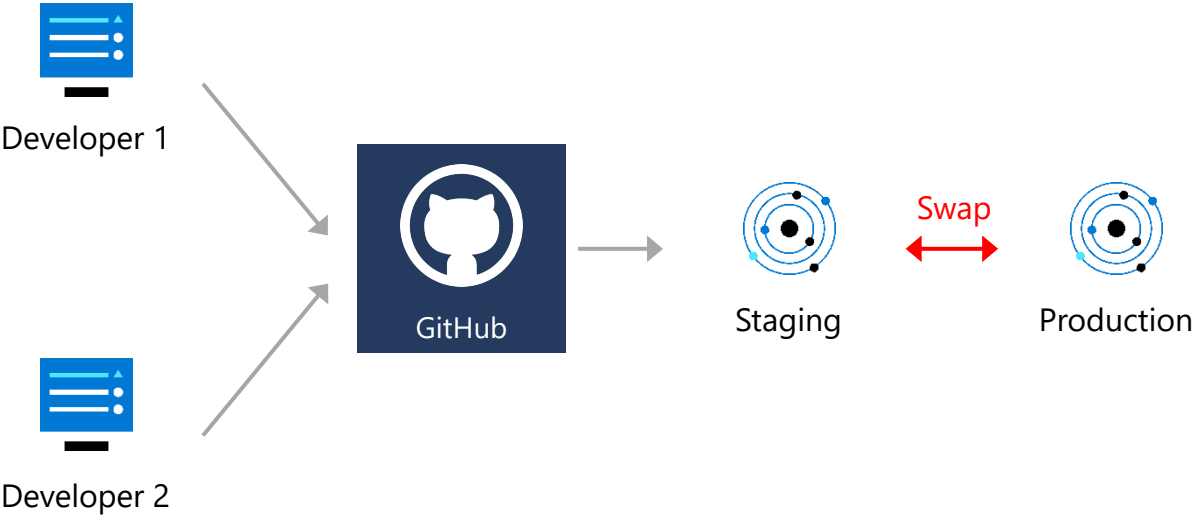
## Zone redundancy

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

Do not clone settings

appservice09

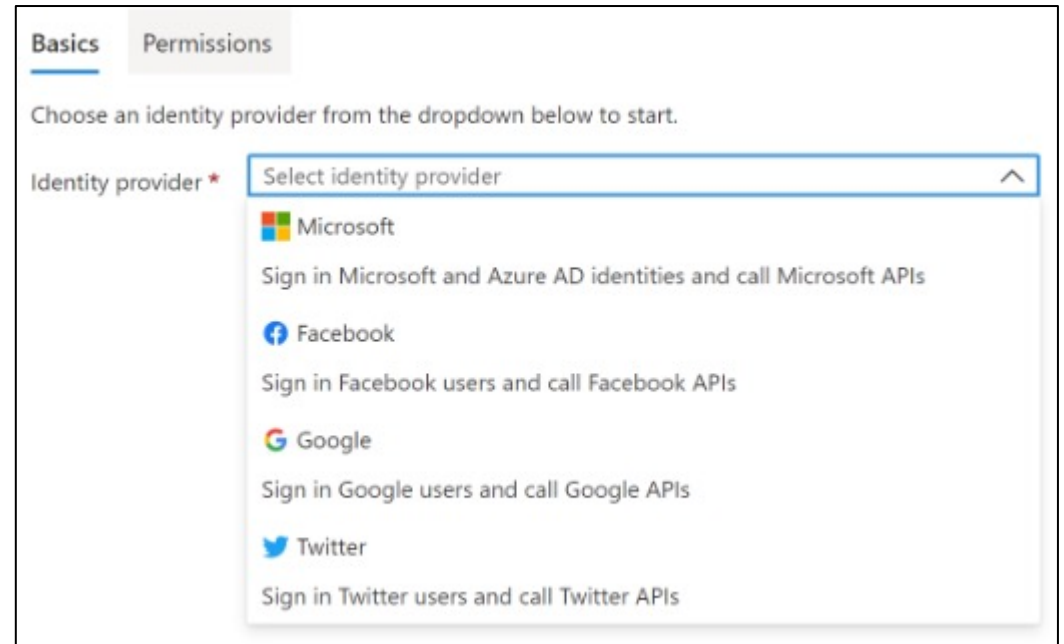
# Secure an App Service

## Authentication:

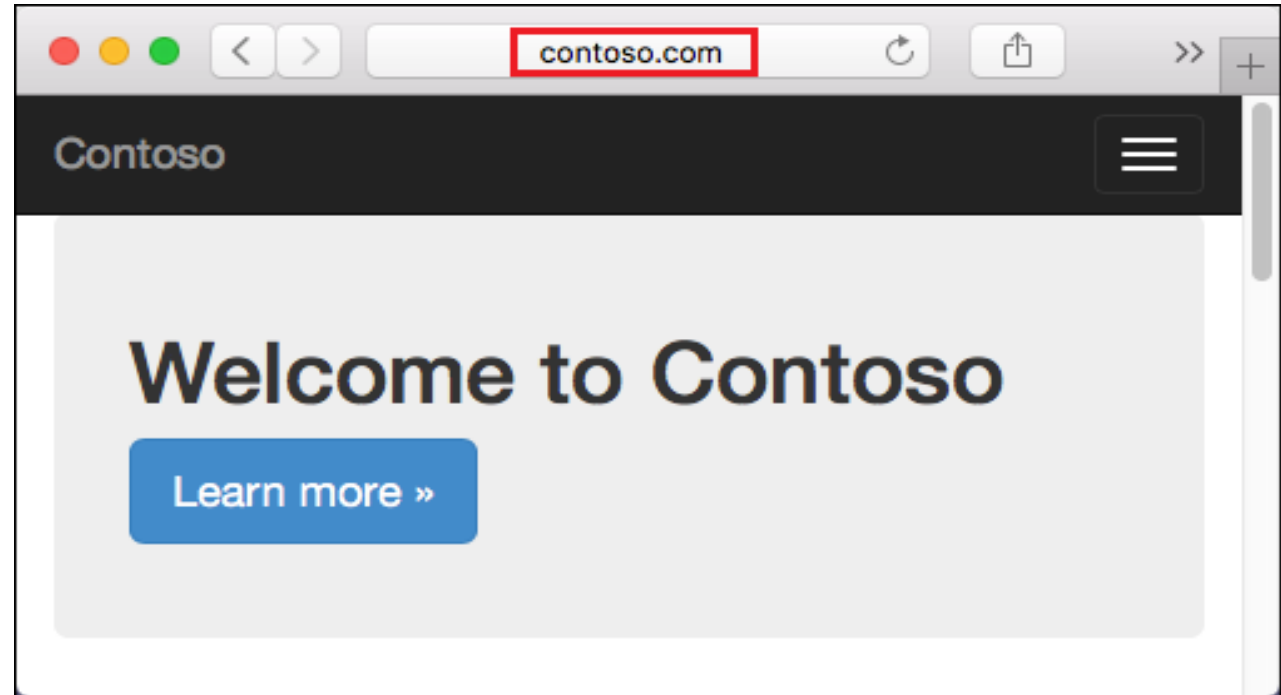
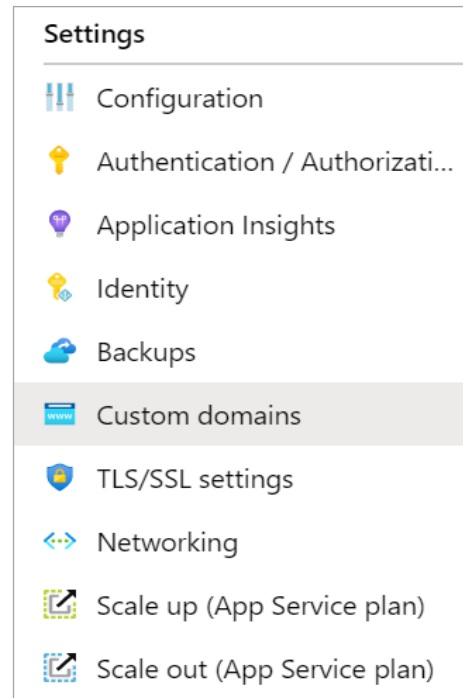
- Enable authentication – default anonymous
- Log in with a third-party identity provider

## 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



# 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

## Settings



Configuration



Authentication / Authorizati...



Application Insights



Identity



Backups



Custom domains



TLS/SSL settings



Networking



Scale up (App Service plan)



Scale out (App Service plan)

# Configure Azure Container Instances



# Learning Objectives - Configure Azure Container Instances

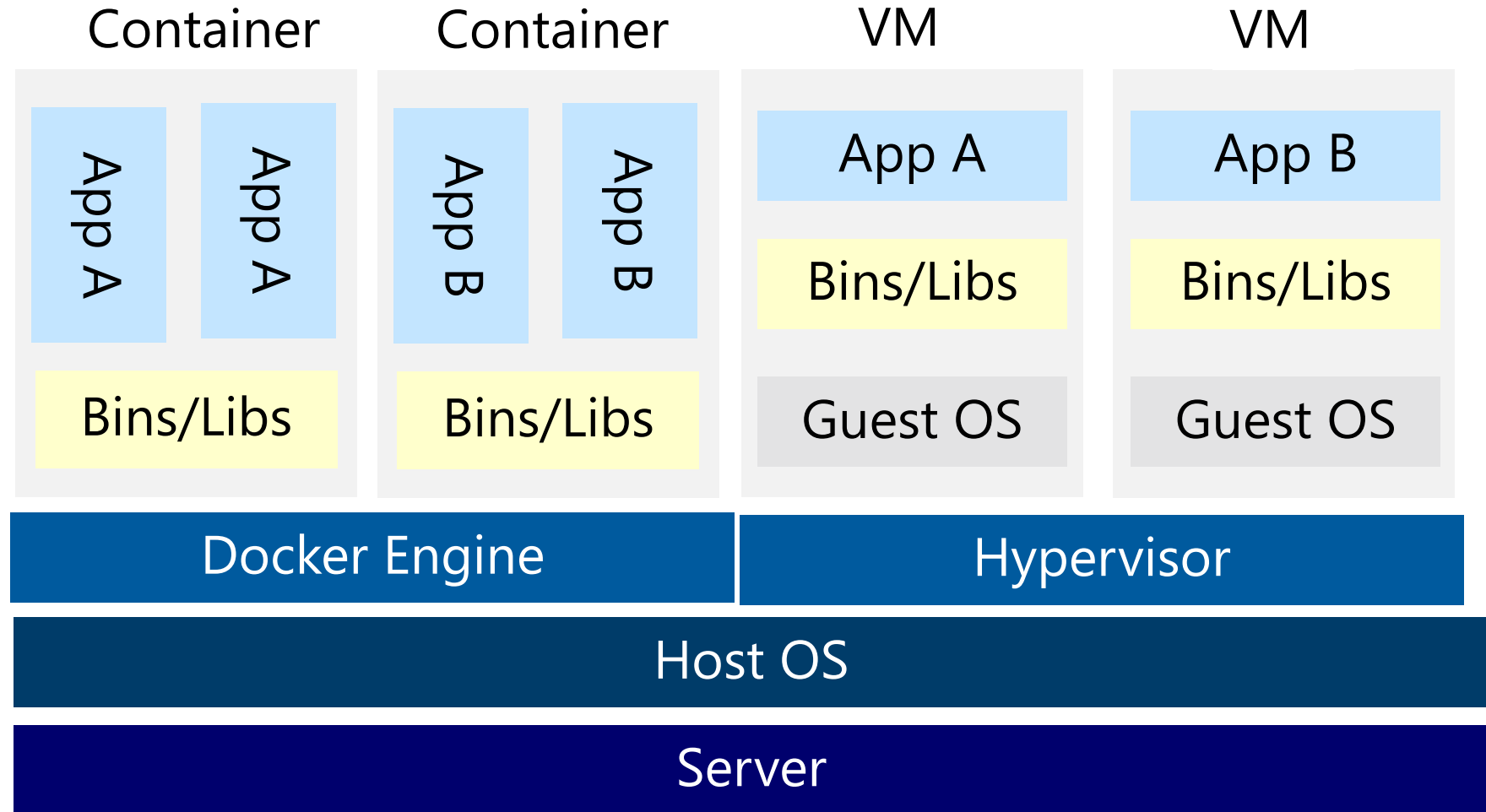
- Compare Containers to Virtual Machines
- Explore Azure Container Instances Benefits
- Implement Container Groups
- Understand the Docker Platform (optional)
- Demonstration – Configure Azure Container Instances
- Manage Containers with Azure Container Apps (new)
- Demonstration – Configure Azure Container Apps
- Learning Recap

Implement and manage Azure compute resources (20-25%): Create and configure containers in Azure portal

- Create and manage an Azure container registry
- Provision a container by using Azure Container Instances (ACI)
- Provision a container by using Azure Container Apps (ACA)
- Manage sizing and scaling for containers, including ACI and ACA

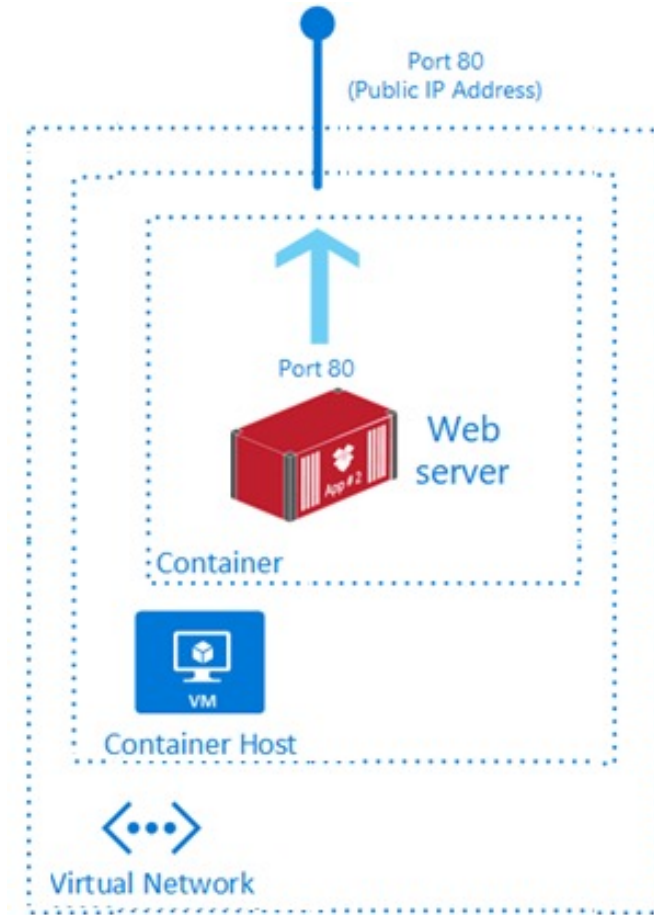
# Compare Containers to Virtual Machines

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



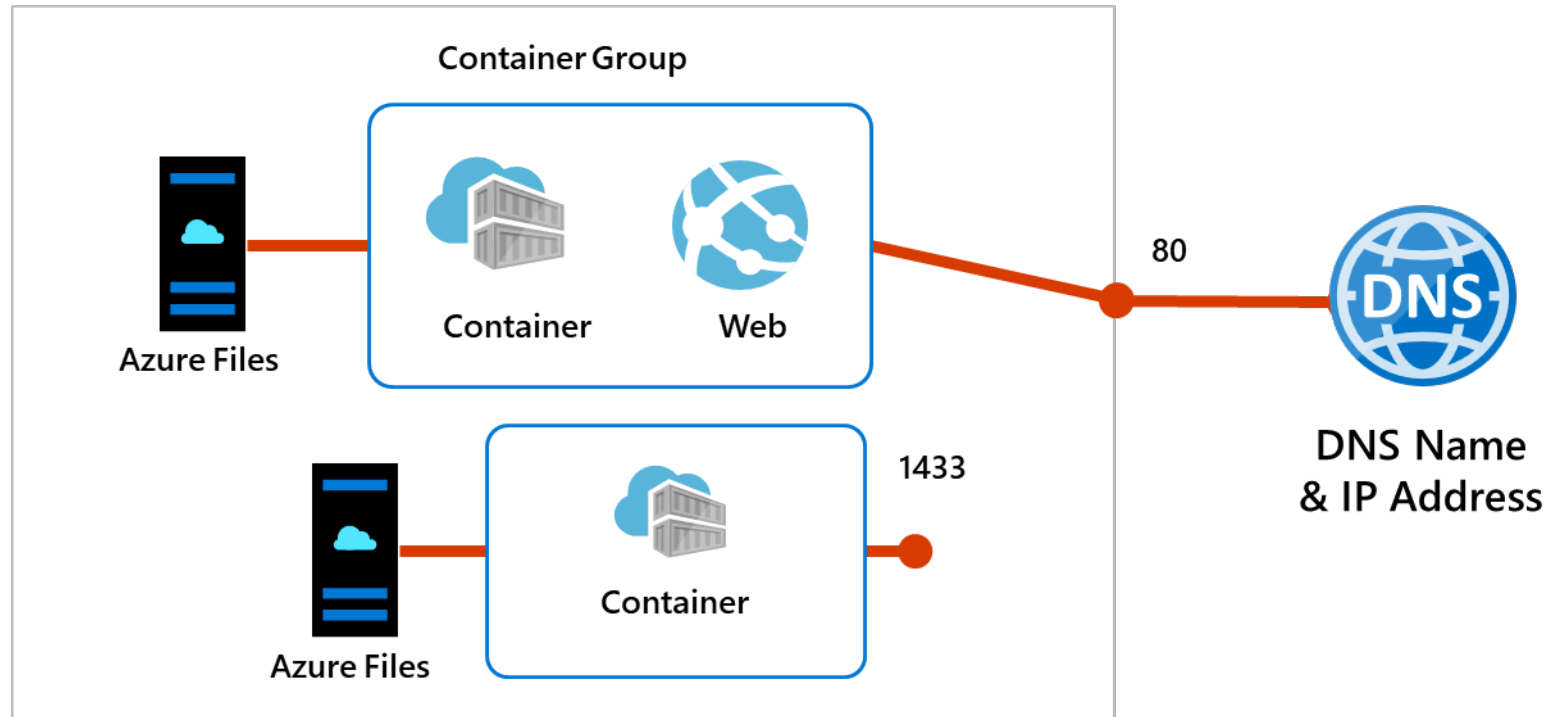
# Explore Azure Container Instances Benefits

- 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

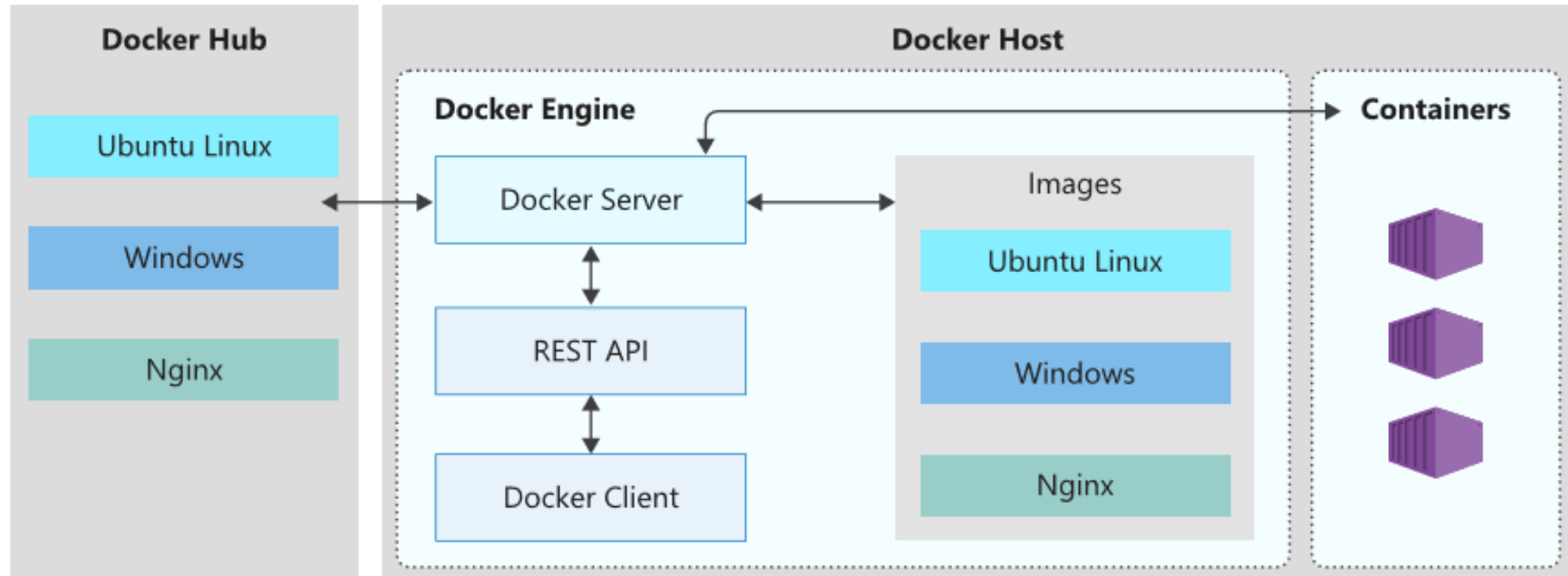


Top-level resource in Azure Container Instances

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

# Understand the Docker Platform



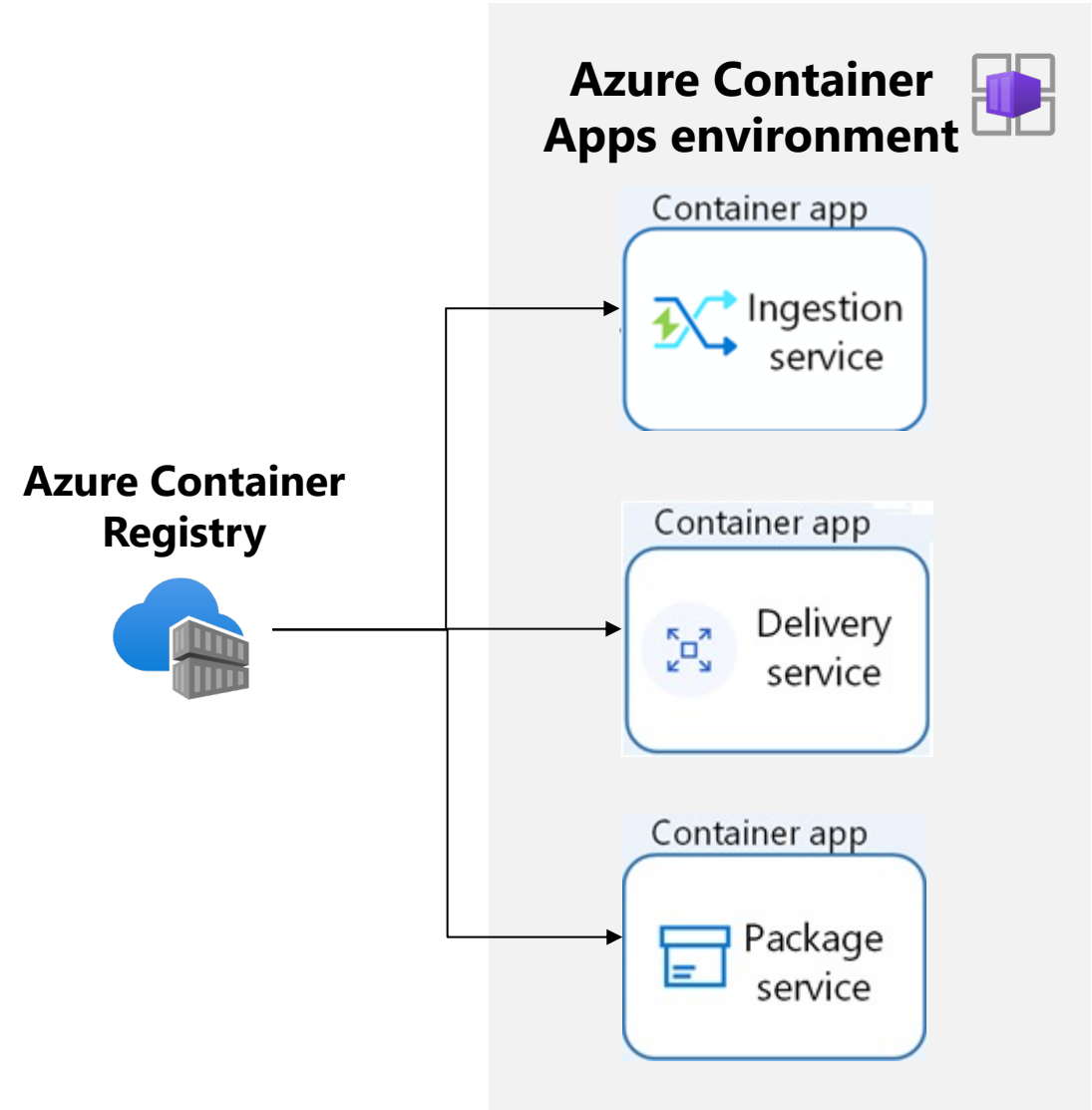
Enables developers to host applications within a container

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

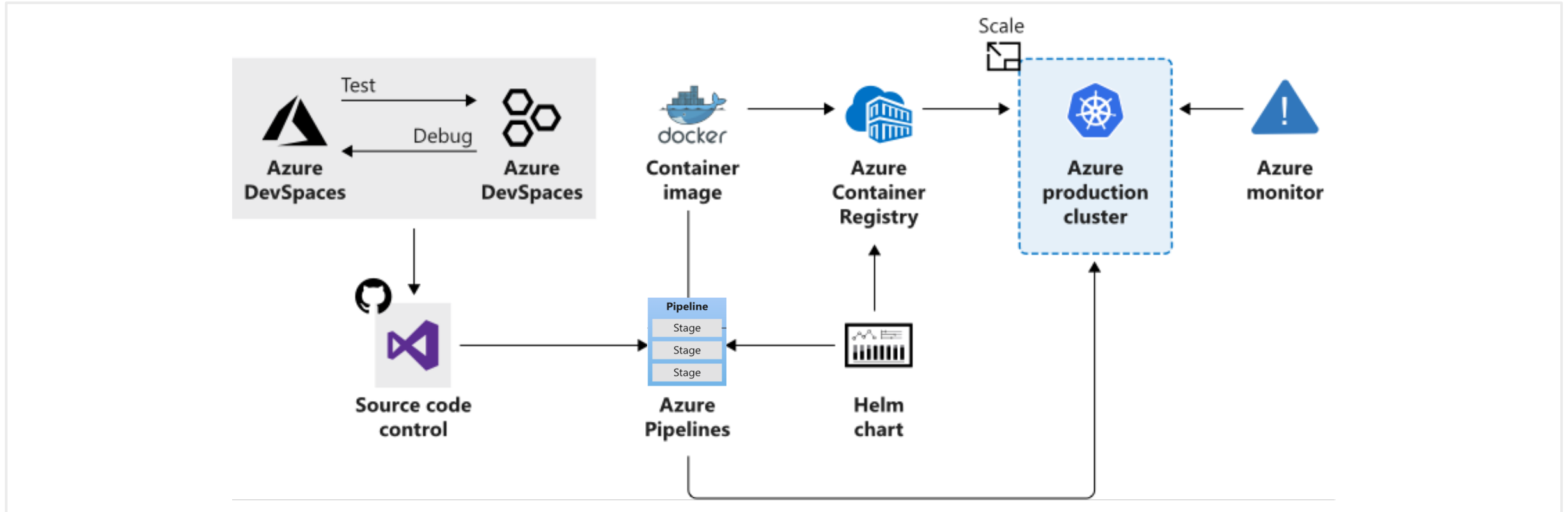
# Manage Containers with Azure Container Apps

- Alternative to Azure Kubernetes Service
- Integrates with Azure Container Registry
- Simplifies complex infrastructures
- Manages container orchestration





# Azure Kubernetes Service



Manages health monitoring and maintenance

Performs simple cluster scaling

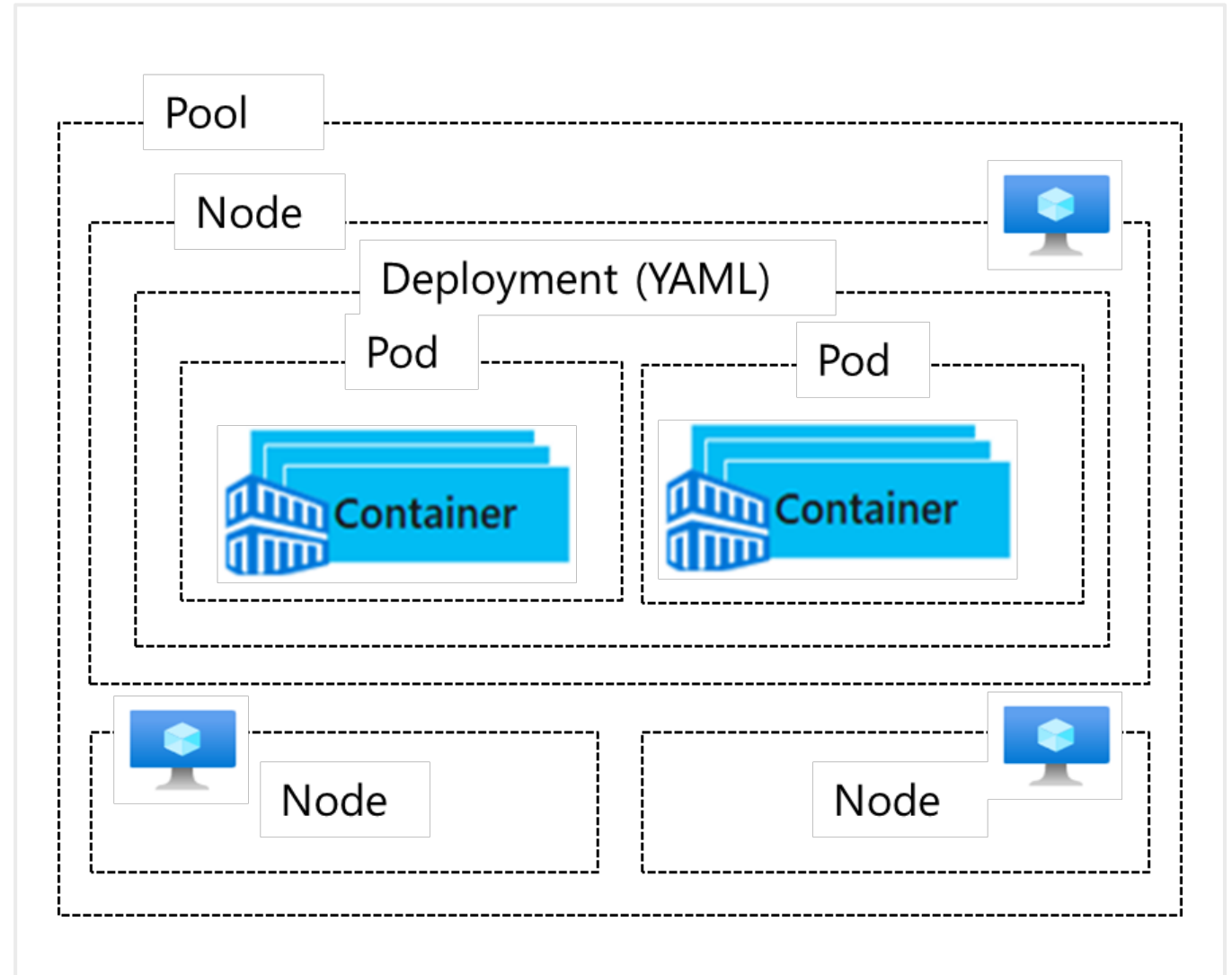
Enables nodes to be fully managed by Microsoft

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 – Implement web apps



You need to evaluate the use of Azure Web apps for hosting Contoso's web sites, hosted currently in the company's on-premises data centers. The web sites are running on Windows servers using PHP runtime stack. You also need to determine how you can implement DevOps practices by leveraging Azure web apps deployment slots.

## Objectives

**Task 1:** Create an Azure web app

**Task 2:** Create a staging deployment slot

**Task 3:** Configure web app deployment settings

**Task 4:** Deploy code to the staging deployment slot

**Task 5:** Swap the staging slots

**Task 6:** Configure and test autoscaling of a web app

# Lab 09a – Architecture diagram

## Task 1



az104-09a-rg1

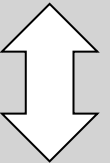


AppService



Production slot

## Task 5



Swap the  
staging slot

## Task 2



Staging slot

## Task 3



Local git



AppServiceplan

## Task 6



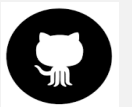
Autoscale rule

php-docs-hello-world  
code

## Task 4



php-docs-hello-world  
code



# Lab 09b – Implement Azure Container Instances



Contoso wants to find a new platform for its virtualized workloads. You identified several container images that can be leveraged to accomplish this objective. Since you want to minimize container management, you plan to evaluate the use of Azure Container Instances for deployment of Docker images

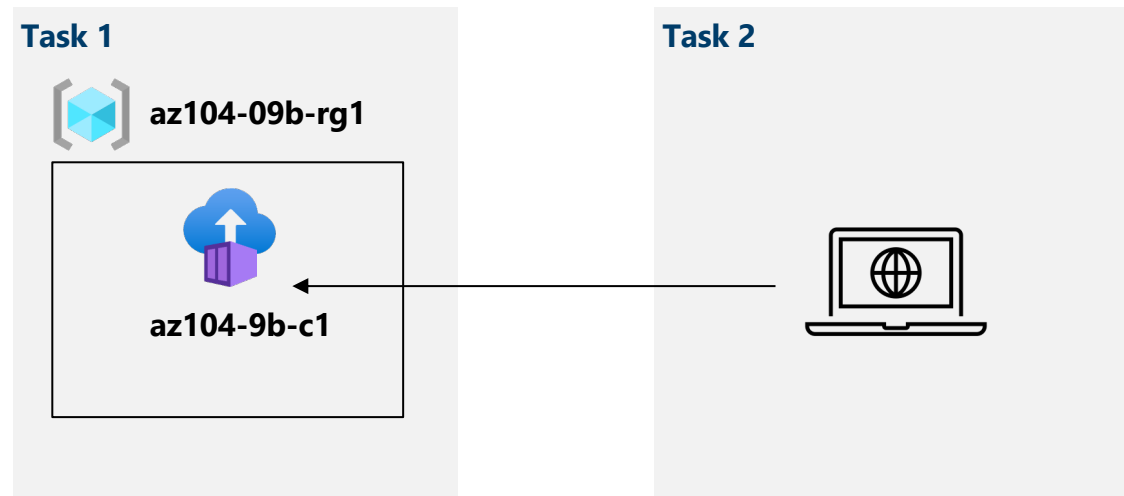
## Objectives

**Task 1:** Deploy a Docker image by using the Azure Container Instance

**Task 2:** Review the functionality of the Azure Container Instance

Next slide for an architecture diagram 

# Lab 09b – Architecture diagram



# Lab 09c – Implement Azure Container Apps



Azure Container Apps enables you to run microservices and containerized applications on a serverless platform. With Container Apps, you enjoy the benefits of running containers while leaving behind the concerns of manually configuring cloud infrastructure and complex container orchestrators.

## Objectives

**Task 1:** Create and configure the Azure Container App and environment

**Task 2:** Deploy the Azure Container App

**Task 3:** Test and verify the Azure Container App



# End of presentation

