



Host a web application with Azure App Service

Callon Campbell

Technical Architect | Developer | Microsoft MVP

Cloud Mavericks Inc.

 @flying_maverick

Instructor introduction



Callon Campbell

Consultant | Technical Architect | Developer
Microsoft MVP in Azure



Callon is the **Founder of Cloud Mavericks Inc** and a **Microsoft MVP** in Azure with over 20 years of experience designing and building applications with Microsoft technologies. I'm passionate about serverless and cloud-native development in Azure.



Cloud Mavericks Inc.

Passionate craftsmanship with a
commitment to excellence

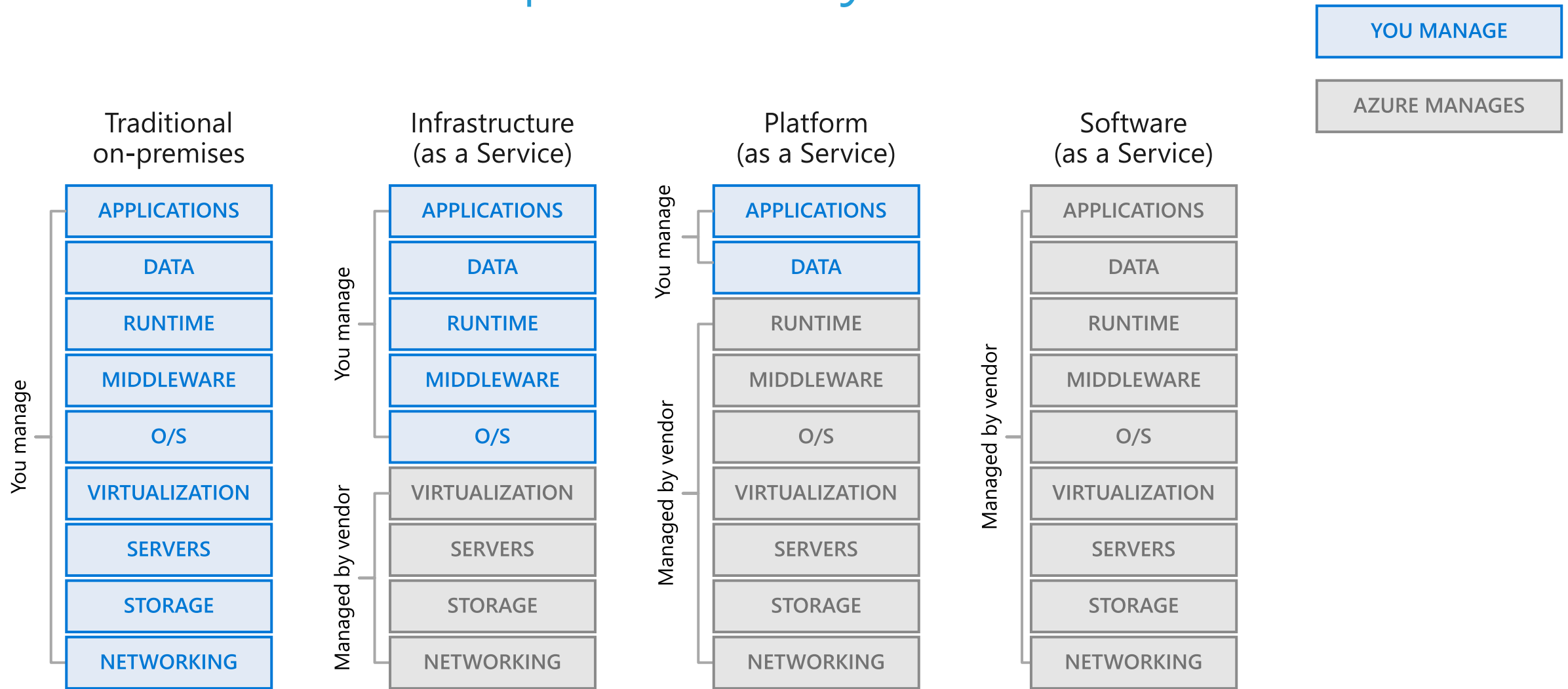
Agenda

- Azure App Service core concepts
- Creating an App Service Web App
- Configuring App Service apps
- Scaling App Service apps
- App Service deployment slots (staging environment)
- Demos
- Labs

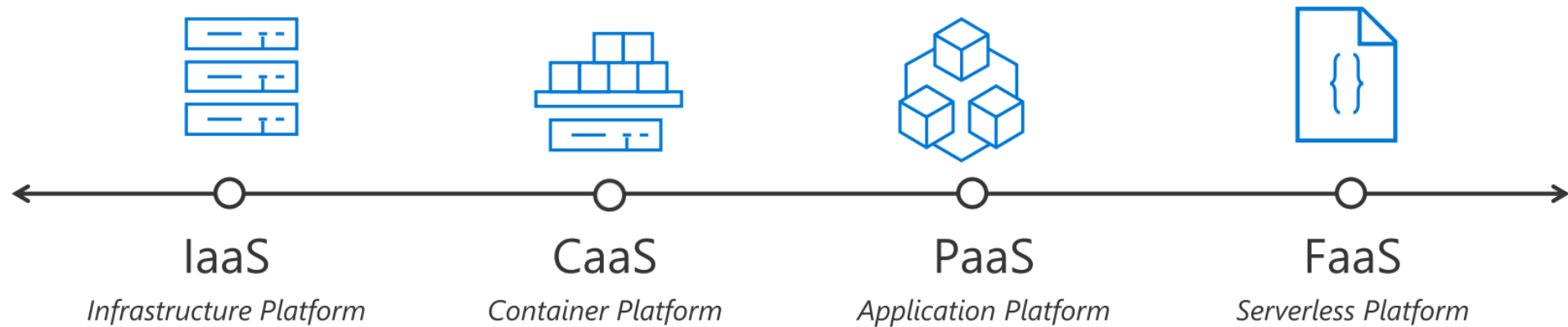
Intro



Balance of responsibility



Cloud application hosting



More Control

of execution environment

Less Control

of execution environment



Less Agile

development & deployment

More Agile

development & deployment

Azure App Service core concepts



What is Azure App Service?

- HTTP-based service for hosting web applications, REST APIs, and mobile backends developed in the following languages:

.NET

.NET Core

Java

Ruby

Node.JS

PHP

Python

- Applications run and scale with ease in a fully managed, sandbox environment with both Windows and Linux-based environments.

Keys features of App Service Web Apps

- Serverless code
 - Run code on-demand without having to explicitly provision or manage infrastructure
 - First-class support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python
- DevOps optimization
 - Continuous integration and deployment with GitHub, Azure DevOps, Bitbucket, Docker Hub, or Azure Container Registry
- Auto scale and high availability
 - Scale up or out manually or automatically
 - App Service SLA
- Connections to SaaS platforms and on-premises data:
 - Connectors to SAP, Salesforce, Facebook
 - Access on-premises data using Hybrid Connections and Azure Virtual Networks

Key features of App Service Web Apps (cont.)

- Managed production environment
 - App Service automatically patches and maintains the OS and language frameworks for you
- Security and compliance
 - App Service is ISO, SOC, and PCI compliant
- Application templates
 - Templates in the Azure Marketplace, such as WordPress, Joomla, and Drupal
- Visual Studio and Visual Studio Code integration
 - Streamline the work of creating, deploying and debugging
- API and mobile features
 - Turn-key Cross-Origin Resource Sharing (CORS) support for RESTful API scenarios, and enables authentication, push notifications, and more
- WebJobs
 - Run background processes at scale

App Service on Linux

- Host web apps natively on Linux for supported application stacks.
- Run custom Linux containers (also known as Web App for Containers).
- Supported languages include: Node.js, Java (JRE 8 & JRE 11), PHP, Python, .NET Core, and Ruby.

App Service Plan

- Provides your application with the necessary capacity
 - CPU, RAM, Storage
- A plan can be shared among multiple applications
- The App Service Plan is the unit of billing
- Different plans provide your application with different features

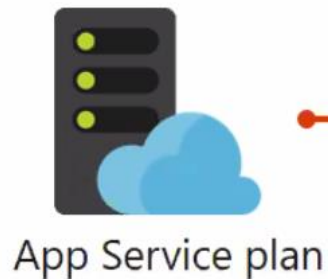
App Service Plans

- Free
- Shared
- Dedicated
 - Basic (Scale Out, Always On)
 - Standard (Auto Scale, Staging Slots, Vnet integration)
 - Premium (Faster CPU's, Private Endpoint)
- Isolated (App Service Environment)
 - Host within your Vnet

Features and pricing

<https://azure.microsoft.com/en-us/pricing/details/app-service/windows/>

App Service Plans (cont.)



App Service plans can logically group apps within a subscription

- Resources are shared among the website instances in the group
- All apps must use the same OS type
- All apps will be deployed in the same region

Web app



Web app



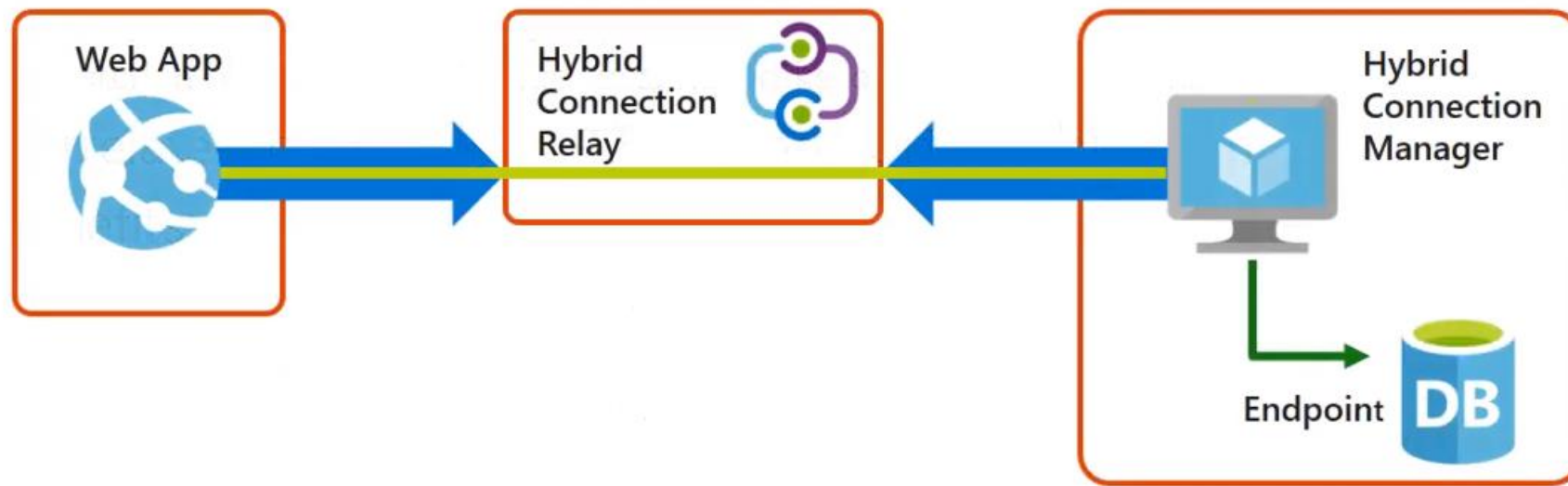
Web app



Azure App Service Hybrid Connections

- Enables access to resources in other networks:
 - Any operating system and any application
 - Hosted in other cloud networks, local networks, or even a specific machine
- Correlates to a single TCP host and port combination
- Benefits:
 - Doesn't require internet-facing endpoint (public IP)
 - Does not require firewall changes in most scenarios
 - Framework and operating system agnostic

Azure App Service Hybrid Connections



Creating an Azure App Service



Demo 1

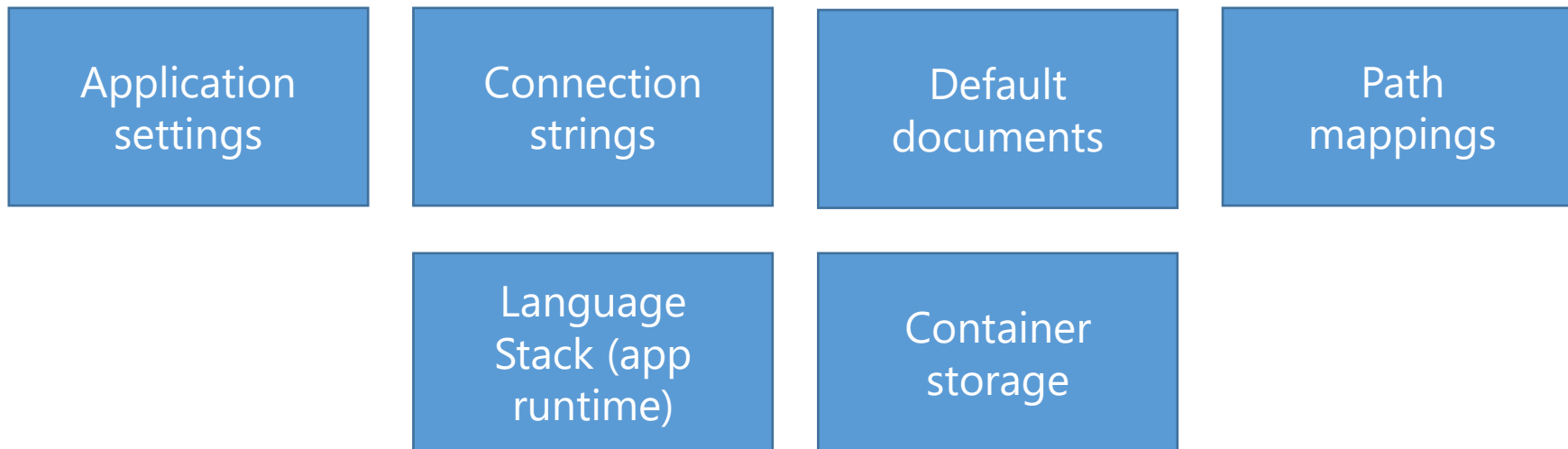
- Creating Azure App Service
- Deploy Web App to App Service

Configuring App Service Apps



App Service Configuration

- Overrides settings in web.config or appsettings.json
- Hidden by default in Azure portal
- You can configure:



Default Documents

- Only available for Windows plans
- Only needed for static websites
- List of documents to show when navigating to a directory on the web server:
 - First matching file is used

Path Mappings

- Windows
 - Custom IIS handler mappings
 - Virtual applications/directories
- Containerized:
 - Custom-mounted storage

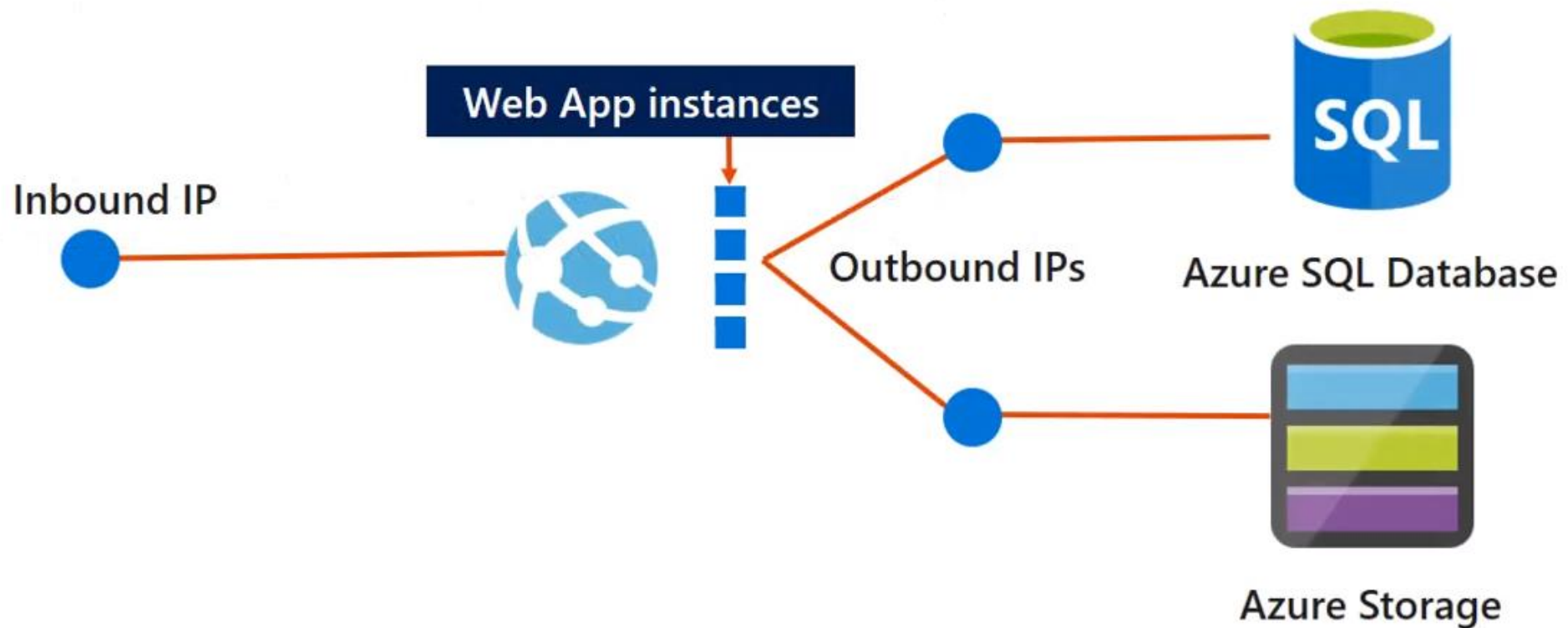
OS and Runtime Patching

- OS and application stack are managed by Azure on your behalf
- Monthly OS patching:
 - Physical servers
 - Hypervisor
 - Guest virtual machines
- Stable versions of application runtimes are periodically added to App Services:
 - Some are installed side by side, while others replace existing versions
 - You can manually migrate from one application runtime to another

Inbound and Outbound IP Addresses

- Each app has a single inbound IP address:
 - Regardless of scale-out quantity
 - You can opt to use a static inbound IP
- Each app has a set number of outbound IP addresses:
 - The set and quantity changes as you scale your app between tiers

Inbound and Outbound IP addresses



When IP addresses change

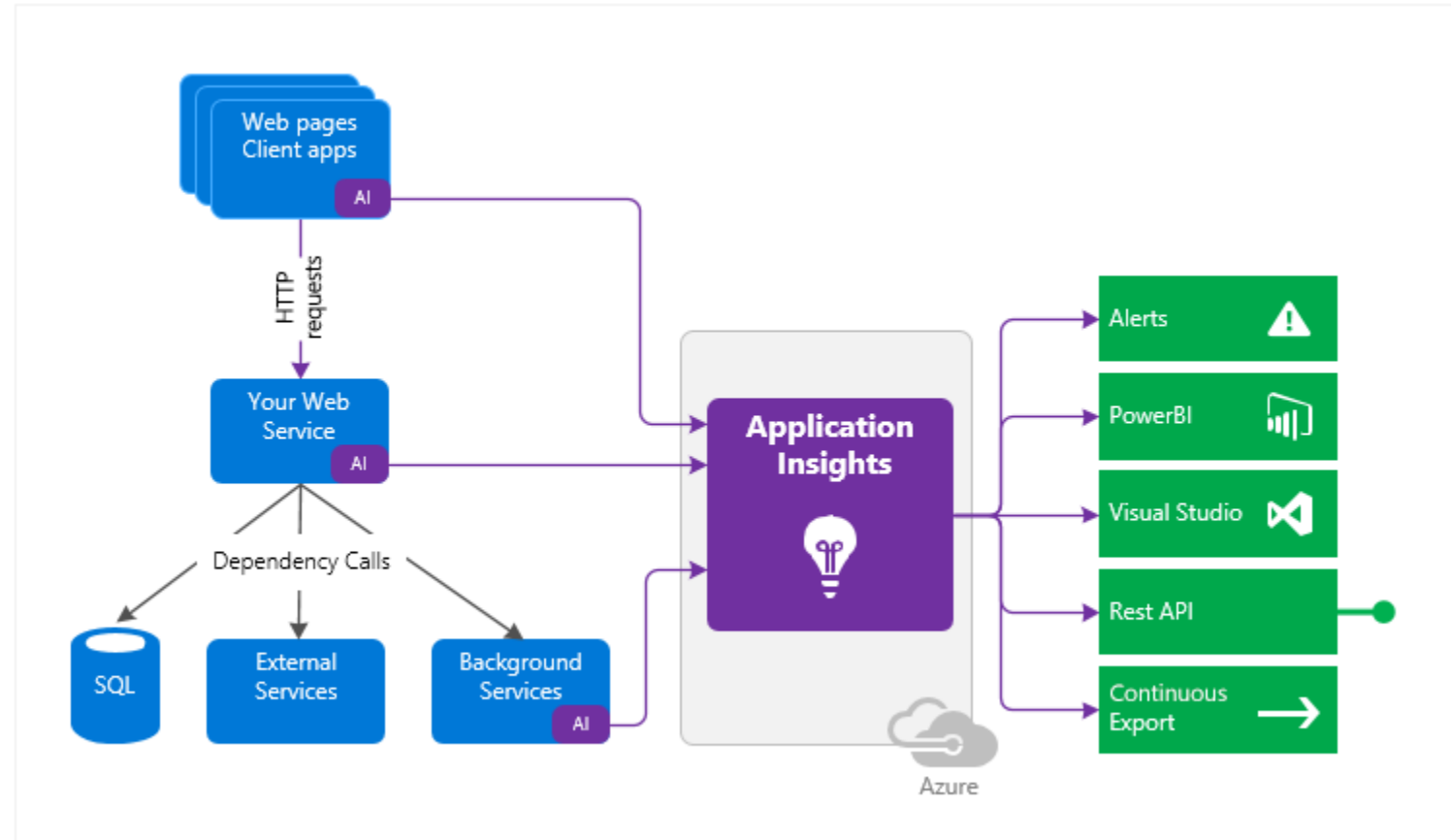
- Inbound:
 - When you delete an app and recreate it in a different resource group
 - When you delete the last app in a resource group and recreate it
 - When you delete an existing SSL binding
- Outbound:
 - When you scale from a lower tier (basic, standard, premium) to the Premium V2 tier

Logging

- Application Logging
 - Windows/Linux
 - Generated from web app
 - Configurable (Critical, Error, Warning, Info, Debug, and Trace)
- Web Server Logging
 - Windows
 - Raw HTTP request data
- Deployment Logging
- Other Logs (Windows Only)
 - Detailed Error Logging (.htm error pages)
 - Failed Request Tracing (IIS components trace)

Enable Application Insights

Application map
Live metrics stream
Search
Failures
Performance profiling
Track dependencies



Demo

- App Service Configuration
- Application Insights

Scaling App Service Apps



Scaling Options

- Scaling Up / Down
- Scaling Out / In
 - Manual
 - Automatic
- Scaling is always on the plan level

Autoscale

- A primary advantage of the cloud is elastic scaling (ability to use as much as you need)
 - Scaling out as load increases
 - Scaling in when the extra capacity is not needed
- Autoscale refers to the capability of monitoring the application instances to automatically scale out/in
- Standard Plan and above provide the capability to scale automatically

Autoscale Metrics

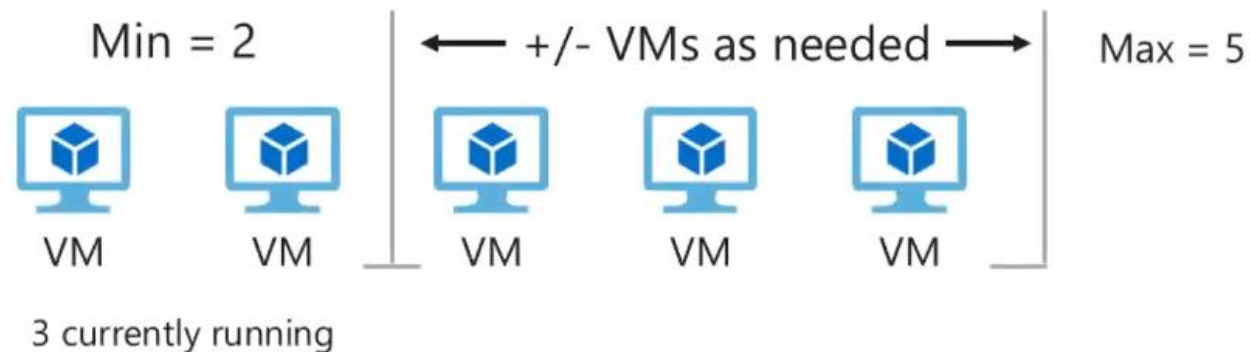
Metric	Metric identifier	Description
CPU	CpuPercentage	The average amount of CPU time used across all instances of the plan
Memory	MemoryPercentage	The average amount of memory used across all instances of the plan
Data in	BytesReceived	The average incoming bandwidth used across all instances of the plan
Data out	BytesSent	The average outgoing bandwidth used across all instances of the plan
HTTP queue	HttpQueueLength	The average number of HTTP requests that had to sit in the queue before being fulfilled. A high or increasing HTTP queue length is a symptom of a plan under a heavy load.
Disk queue	DiskQueueLength	The average number of both read and write requests that were queued on storage. A high disk queue length is an indication of an application that might be slowing down due to excessive disk I/O.

Autoscale Patterns

- Scale based on metrics (ex. CPU, Memory...)
- Scale differently on weekdays vs. weekends
- Scale differently during holidays
- Scale based on custom metric

Autoscale Thresholds

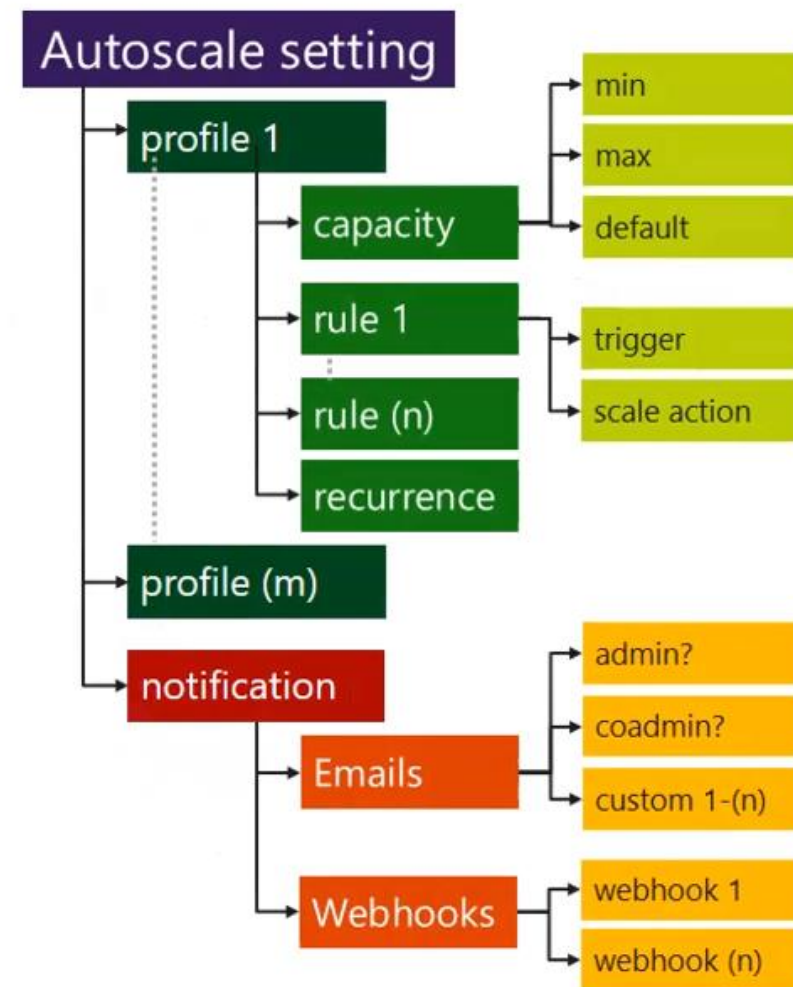
- Scale is constrained to a minimum and maximum:
 - Your current instance count must be between the minimum and maximum:
 - Minimum can help guarantee availability
 - Maximum can help control costs



Autoscale Hierarchy

- One autoscale setting
- Settings have one or more profiles
- Profiles* have one or more rules:
 - Profiles can also have recurrences and capacity settings
- Notifications can be directly associated with an autoscale setting

* On the portal a profile is called condition



Autoscale Concepts



- Each resource can have one autoscale setting:
 - Autoscale settings can have one-to-many profiles
 - Profiles can have one-to-many rules
- Autoscale increases instances horizontally within bounds:
 - Bounds are set by using the min, max and default values
- Thresholds are calculated at an average level across active instances
- Autoscale successful actions and failures are logged to the Activity Log


Scale based on CPU

Custom autoscale

Autoscale setting name *

Resource group

Default*  

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

Scale out



When ASP-lhlwebaprg-8e34 (Average) CpuPercentage > 70 Increase count by 2



Scale in



When ASP-lhlwebaprg-8e34 (Average) CpuPercentage < 30 Decrease count to 1

[+ Add a rule](#)

Instance limits

Minimum  

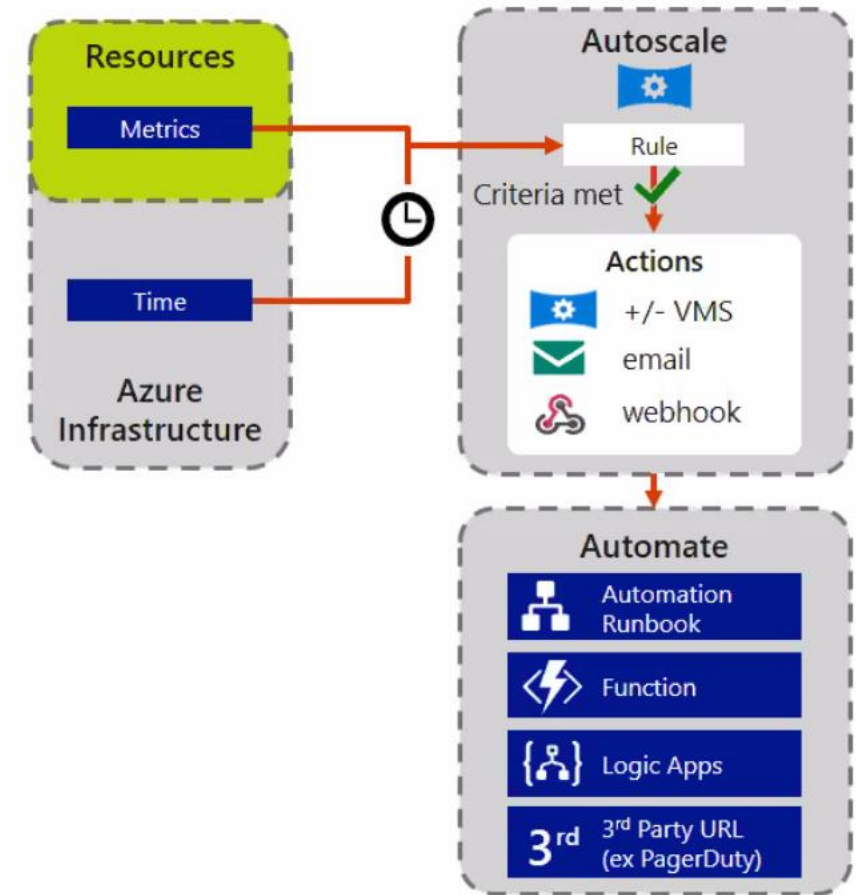
Maximum  

Default  

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

Autoscale Workflow

- Metrics are measured for a resource
- When conditions are met (threshold surpassed), autoscale triggers:
 - Perform scale actions
 - Send notifications (alerts)
 - Send messages to webhooks for external automation



Best practices

- Choose the thresholds carefully for all metric types
- Choose the appropriate statistic for your diagnostic metric (avg is mostly used)
- Ensure that the maximum and minimum values are different and have adequate margin between them
- Manual scaling is reset by autoscale min and max
- Always uses scale-out and scale-in rule combination that perform an increase and decrease

Demo

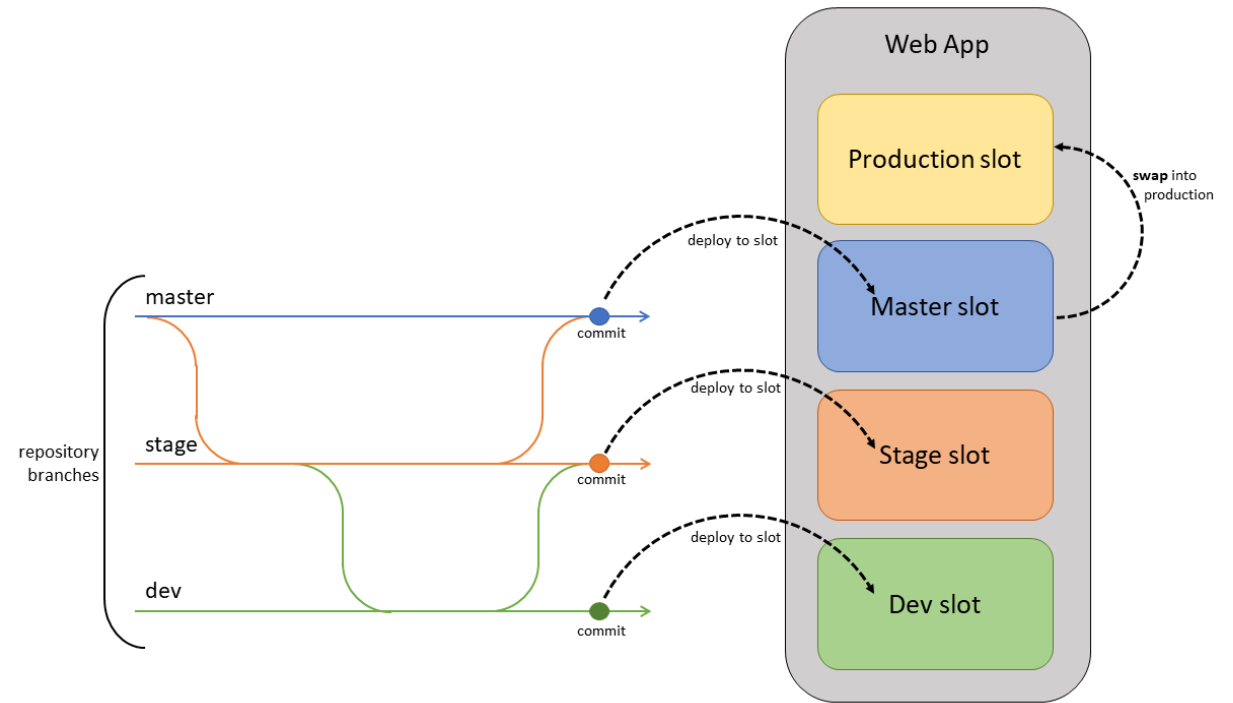
- Scaling Up
- Scaling Out

Azure App Service Deployment Slots

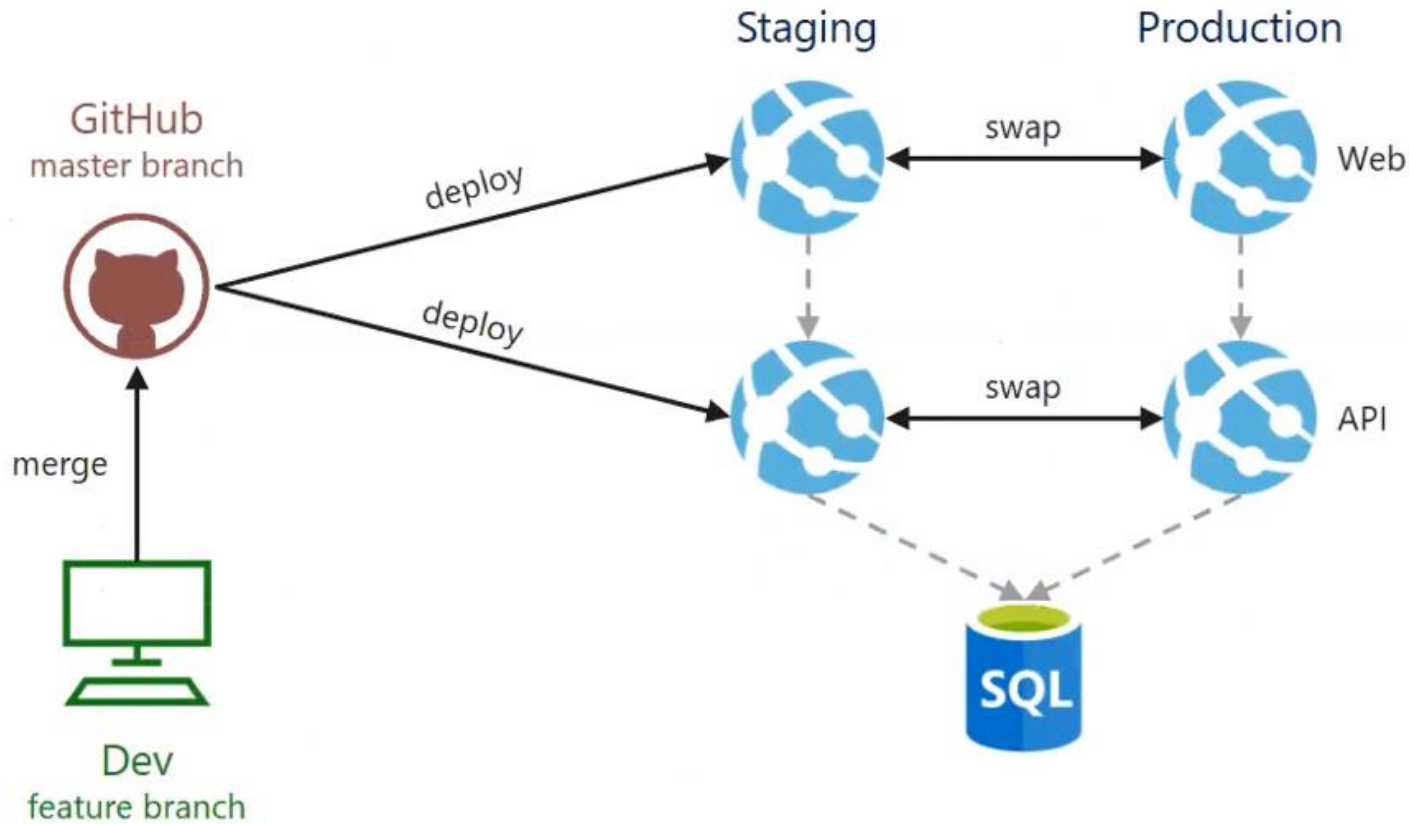


Leverage deployment slots

- Live apps with their own
 - Host names
 - Content
 - Configuration
- Can be swapped between each other
- Prewarming (no downtime)
- Easy fallbacks



Modern deployment workflow

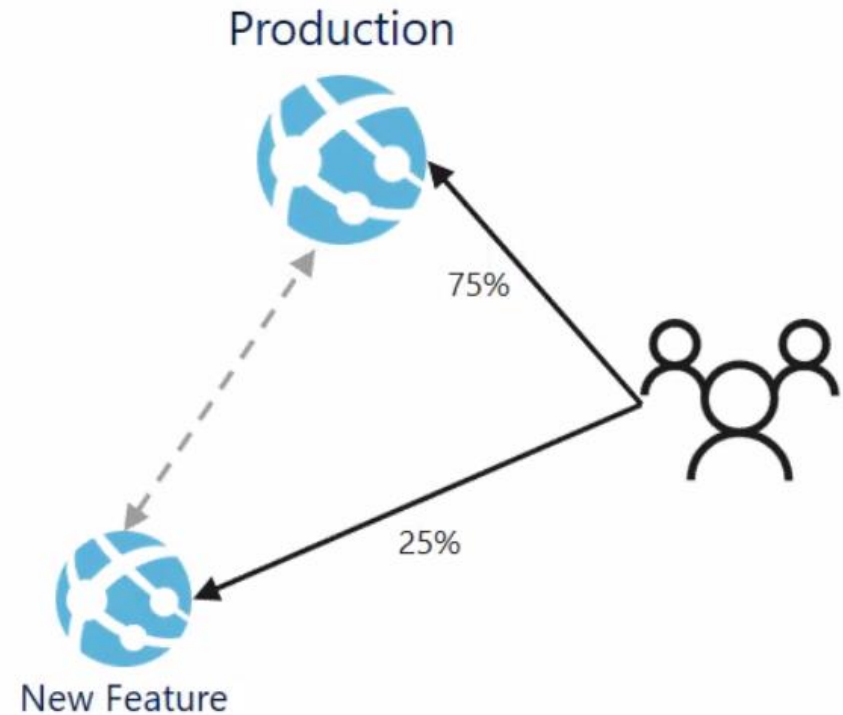


Auto swap

- Automatically swaps an application to the production slot:
 - Performed after the application is “warmed up”
 - Swaps the deployment target slot with the production slot
- Deploy apps continuously while minimizing cold starts and downtime

Route traffic between slots

- All traffic is normally routed to production
 - Production slot has a 100% weighting
- You can manually configure the weight of traffic between multiple slots



Demo

Deployment Slots

Summary

- App Service easily adds the power of Azure to your application
 - Ease of use
 - Scalability
 - Only pay for the resources you use

Labs



Labs

- Labs

- Lab 1 – Create an Azure App Service Web App
- Lab 2 – Create and deploy an ASP.NET app to App Service
- Lab 3 – Scale Up and Out with Autoscale
- Lab 4 – Use Deployment Slots for staging environment and testing in production

- Microsoft Learn

- Host a web application with Azure App Service

<https://docs.microsoft.com/en-us/learn/modules/host-a-web-app-with-azure-app-service/>

Getting help



Get started on Azure

Create your free account: <http://azure.com/free>

What do I get?

With your Azure free account, you get all of this – and you won't be charged until you choose to upgrade.

12 months

of popular free services

+

\$260 credit

to explore Azure for 30
days

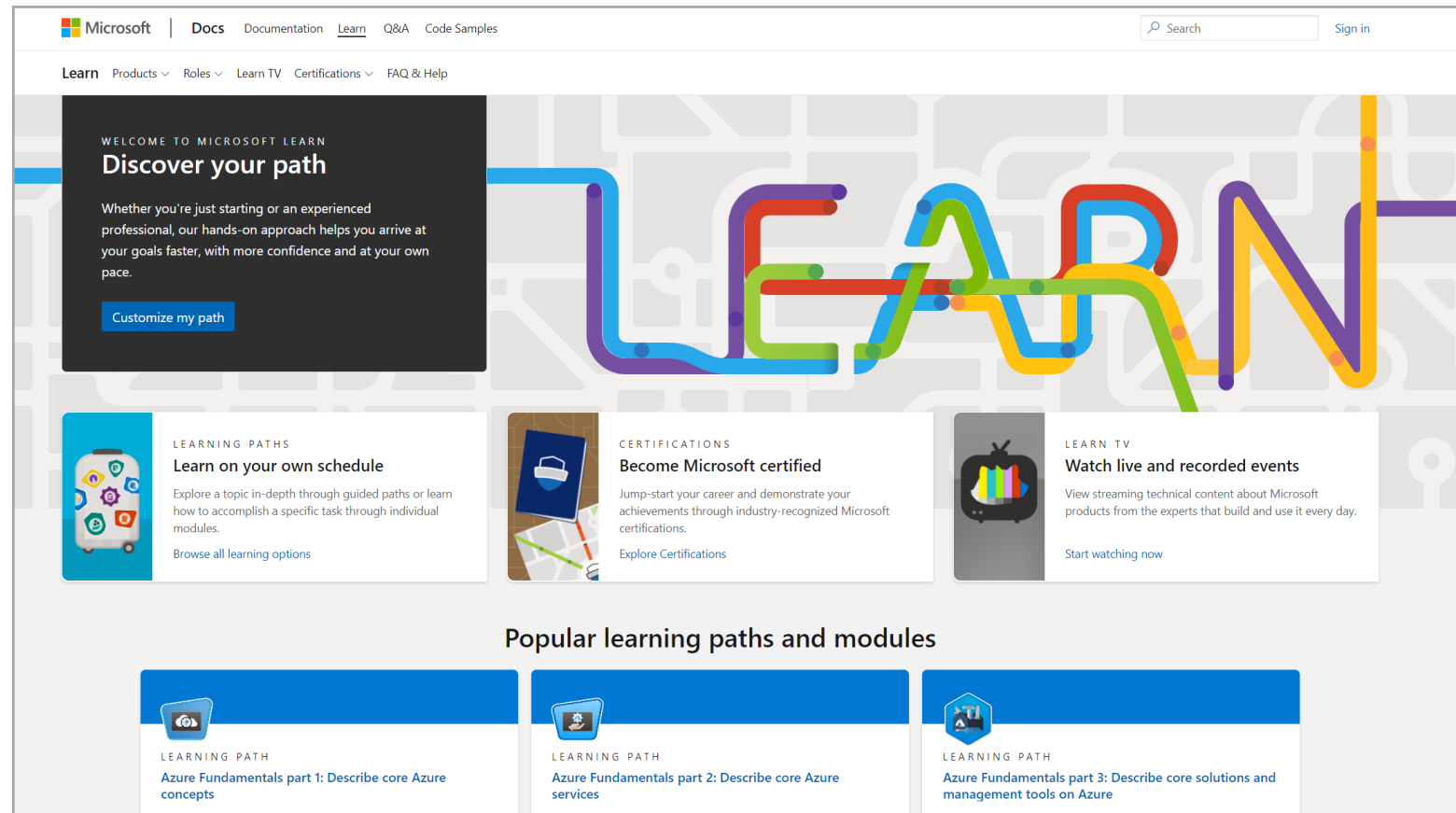
+

Always free

25+ services

Microsoft Learn

Complete interactive learning exercises, watch videos, and practice and apply your new skills.



Microsoft Azure

Resources to continue learning

Microsoft Azure



Learning path

[Collections - cloudskillschallenge | Microsoft Docs](#)

The screenshot shows the Microsoft Docs interface for the 'Azure Solutions Architect' learning path. The top navigation bar includes the Microsoft logo, 'Docs', 'Documentation', 'Learn', 'Q&A', and 'Code Samples'. A search icon and 'Sign in' link are on the right. The main header area features the title 'Azure Solutions Architect' in white, followed by '25 hr 13 min • 33 Modules • Created by Microsoft' and a 'Share' button. A 'See the challenge' button is prominently displayed. Below this, a section titled 'Items in this collection' lists the first module: 'Control and organize Azure resources with Azure Resource Manager', which is 46 minutes long and has a 4.7 rating from 44,059 reviews.

Microsoft | Docs Documentation Learn Q&A Code Samples Search Sign in

Azure Solutions Architect

25 hr 13 min • 33 Modules • Created by Microsoft Share

See the challenge

Items in this collection

MODULE

Control and organize Azure resources with Azure Resource Manager

46 min

★★★★★ 4.7 (44,059)

Resources



Session Materials and Labs on GitHub

<https://github.com/calloncampbell/lighthouse-labs-bootcamp-2021>



Azure App Service Documentation

<https://docs.microsoft.com/en-us/azure/app-service/>

Let's connect



Callon@CloudMavericks.ca



@flying_maverick



<https://LinkedIn.com/in/CallonCampbell>



<https://GitHub.com/CallonCampbell>

Thank You

ευχαριστώ Salamat Po متشكراً شكراً Grazie
благодаря ありがとうございます Kiitos Teşekkürler 谢谢
ឧបត្ថម្ភ Obrigado شكریه Terima Kasih Dziękuję
Hvala Köszönöm Tak Dank u wel дякую Tack
Mulțumesc спасибо Danke Cám ơn Gracias
多謝晒 Ďakujem תודה நன்றி Děkuji 감사합니다