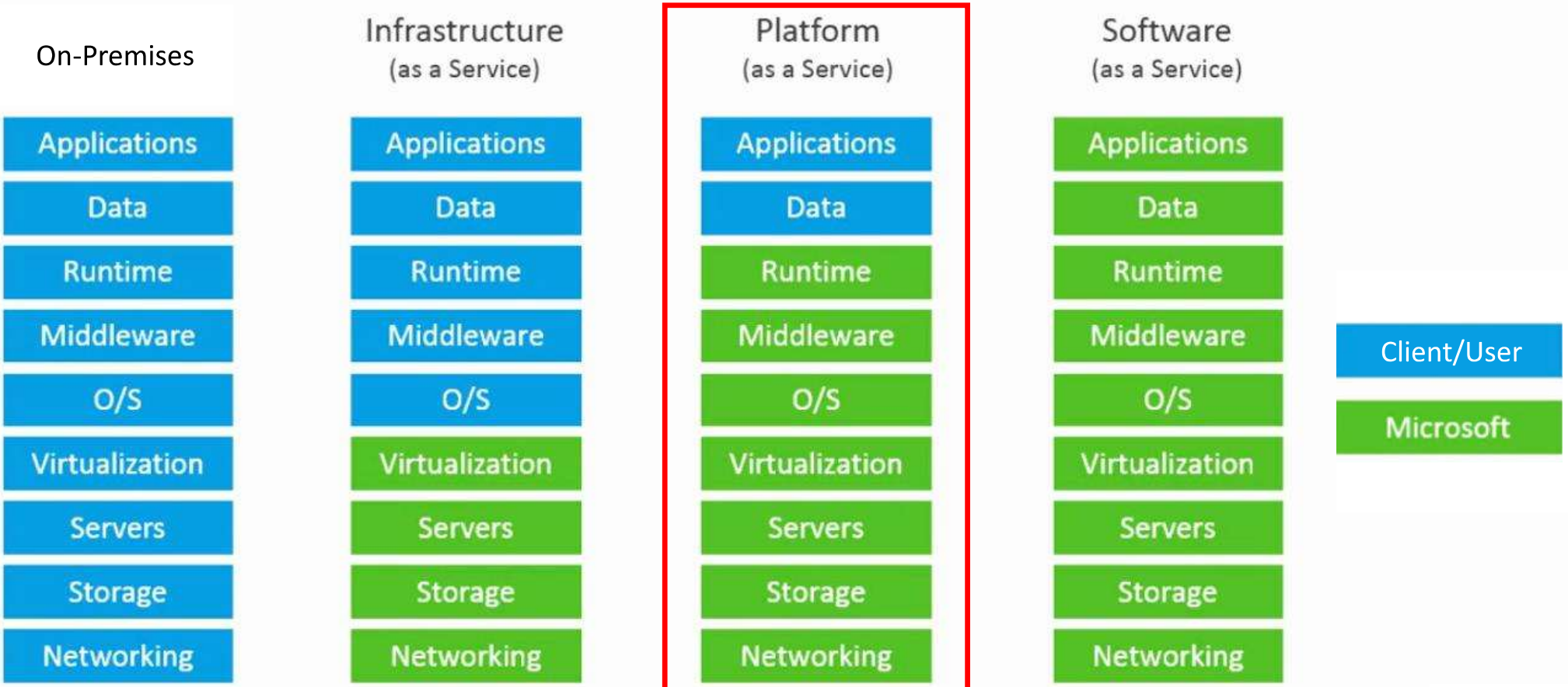
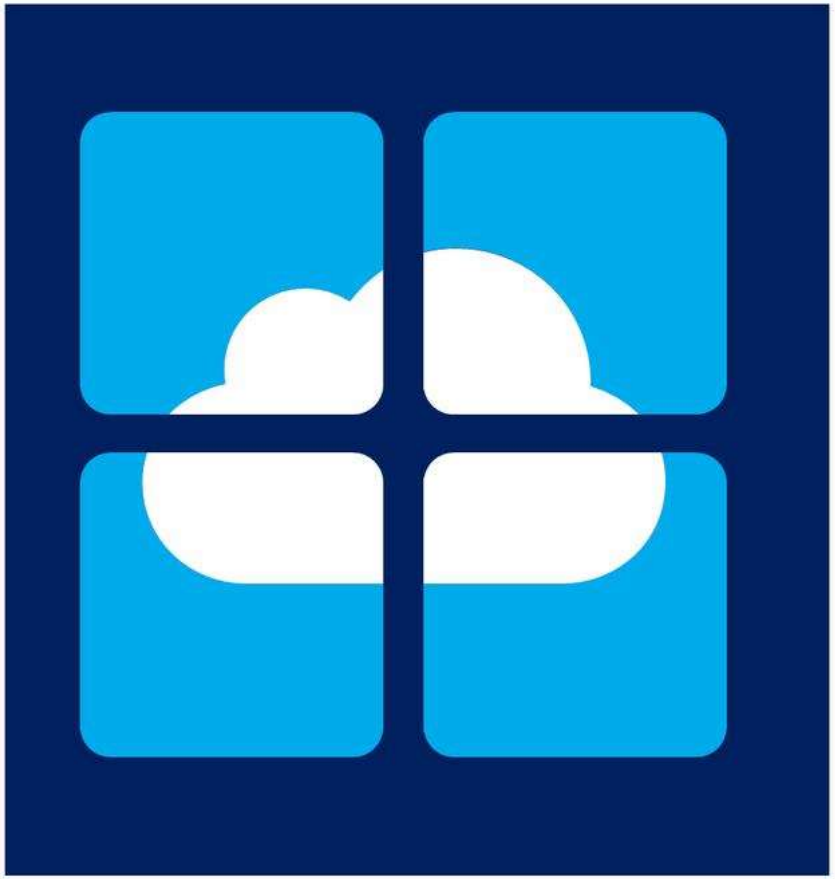
Module 04:

Develop Azure App Service – Web Apps

Cloud Service Models

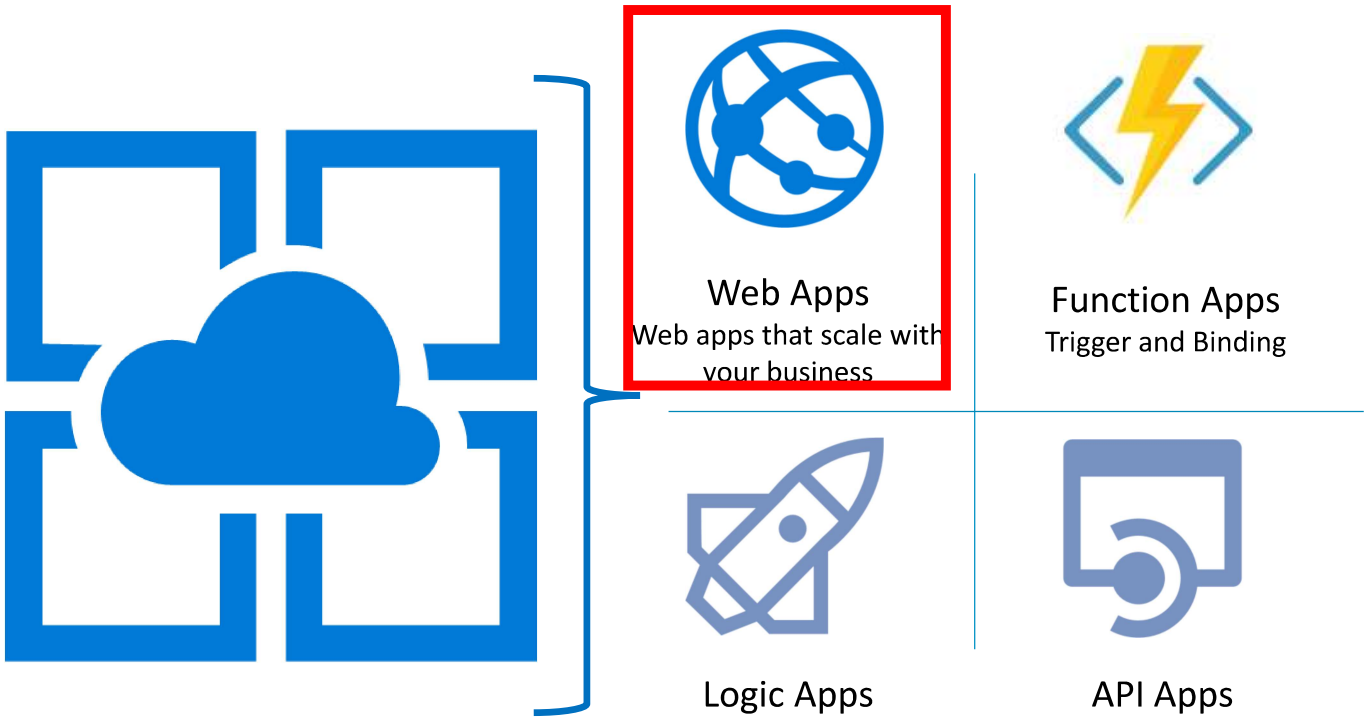




Azure App Service

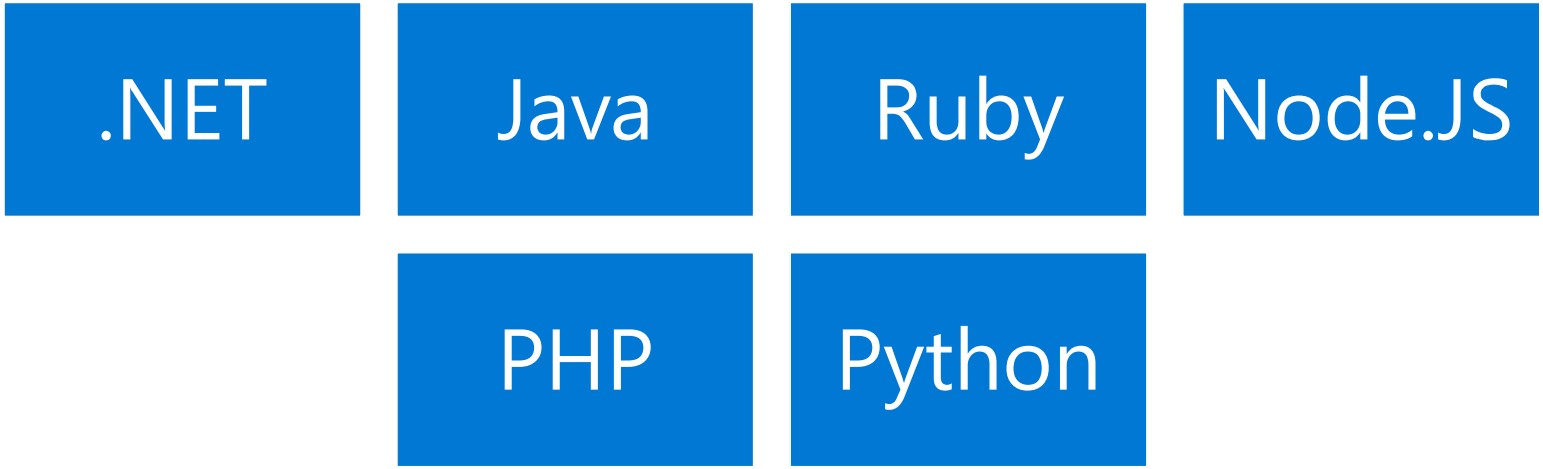
Azure App Service provides an integrated cloud app platform for delivering modern enterprise apps across cloud and mobile devices. Azure App Service is an integrated offering that delivers features and capabilities from a number of existing Azure services – Websites, Mobile Services, API Mgmt. and BizTalk Services

App Service Family



App Service

- Service for hosting web applications, REST APIs, and mobile backends can be developed in many of the following languages:



- Applications can execute and scale in a fully managed, sandbox environment

Key features of App Service Web Apps

- Multiple languages and frameworks:
  - First-class support for Microsoft ASP.NET, Java, Ruby, Node.js, PHP, or Python
- DevOps optimization:
  - Continuous integration and deployment with Visual Studio Team Services, GitHub, Bitbucket, Docker Hub, or Azure Container Registry
- Global scale with high availability:
  - Scale up or out manually or automatically. Host anywhere in the Microsoft global datacenter infrastructure
- Connections to SaaS platforms and on-premises data:
  - More than 50 connectors for enterprise systems (such as SAP), SaaS services (such as Salesforce), and internet services (such as Facebook)

Key features of App Service Web Apps (cont.)

- 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 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, offline data sync, push notifications, and more
- Serverless code:
  - Run code on-demand without having to explicitly provision or manage infrastructure

App Service Plans

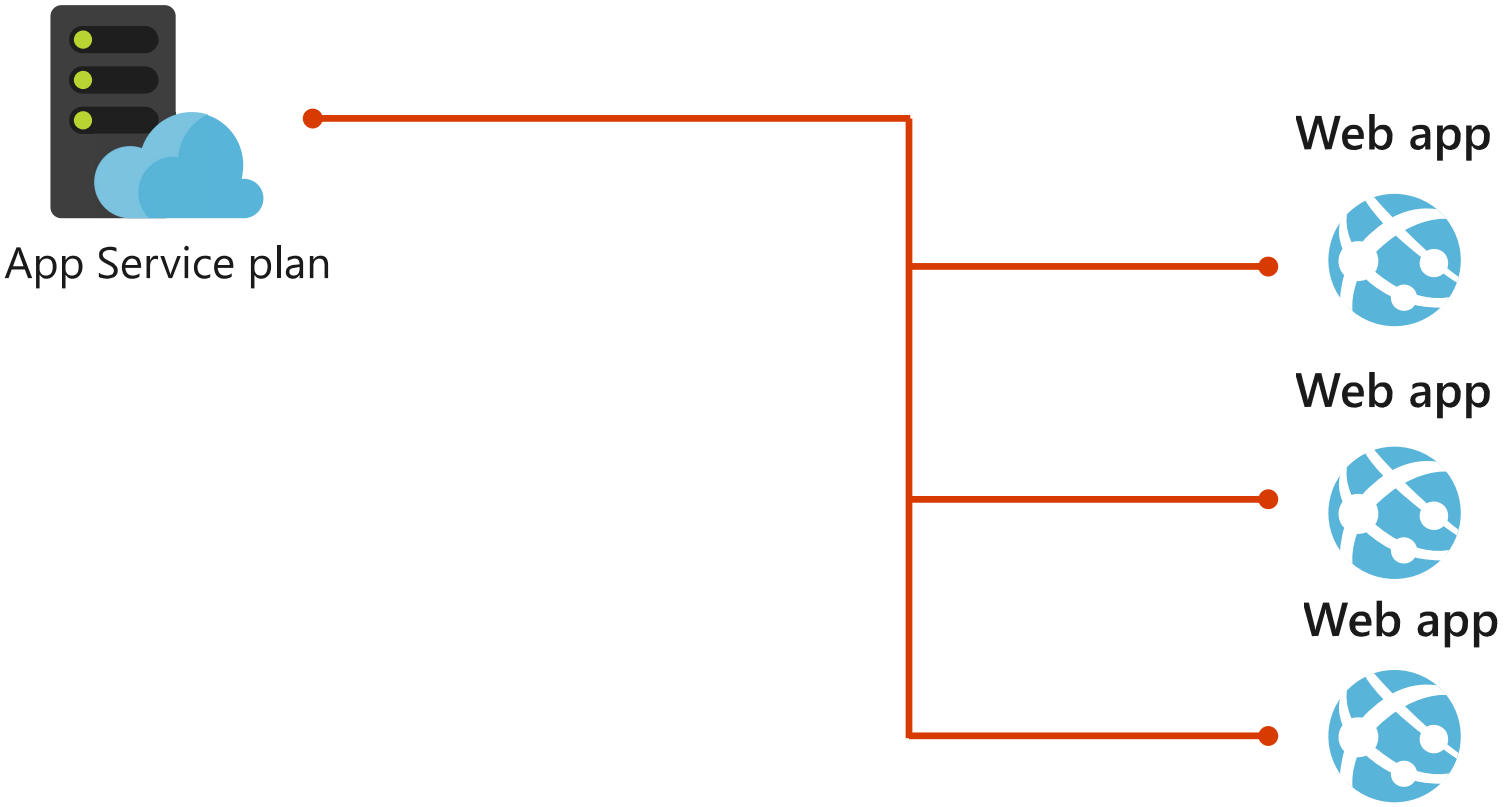
- Define a set of compute resources for a web app to run
- Determines performance, price, and features
- One or more apps can be configured to run in the same App Service plan
- App Service plans define:
  - Region where compute resources will be created
  - Number of virtual machine instances
  - Size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

App Service Plans – Pricing Tier

	Free	Shared	Basic	Standard	Premium	Isolated
# of Apps	10	100	Unlimited	Unlimited	Unlimited	Unlimited
Disk Space	1 GB	1 GB	10 GB	50 GB	250 GB	1 TB
Maximum Instances	1	1	3	10	20	100
Autoscale	No	No	No	Yes	Yes	Yes
Staging Environments	No	No	No	5	20	100
Custom Domains	No	Yes	Yes	Yes	Yes	Yes
SLA			99.95%			

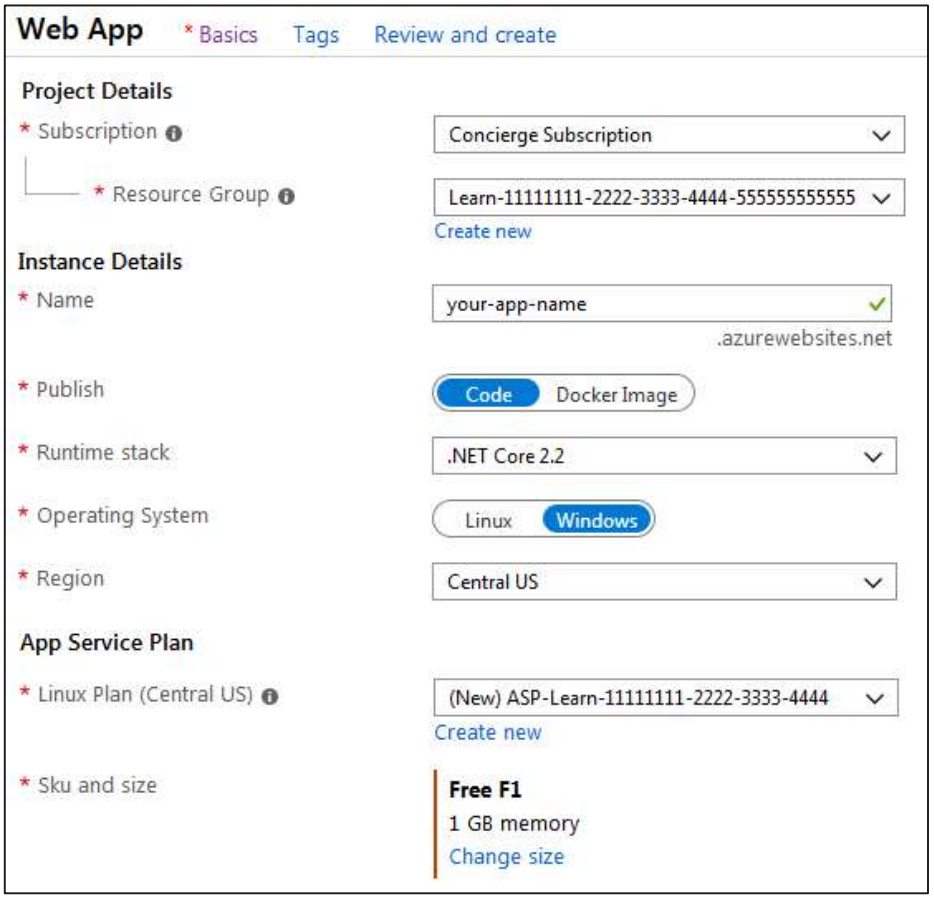
- **Shared compute** (Free and Shared). Run apps on the same Azure VM as other App Service apps, and the resources cannot scale out
- **Dedicated compute** (Basic, Standard, Premium). Run apps in the same plan in dedicated Azure VMs
- **Isolated**. Runs apps on dedicated Azure VMs in dedicated Azure virtual networks

App Service Plans (continued)

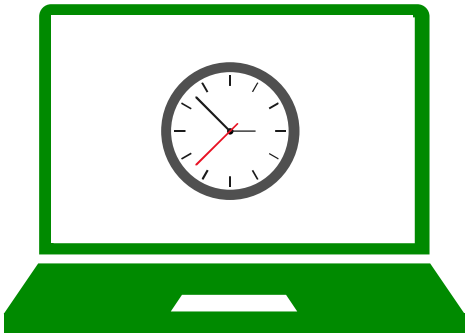


Creating an App Service

- Name must be unique
- Access using *azurewebsites.net* – can map to a custom domain
- Publish Code (Runtime Stack)
- Publish Docker Image (Image source)
- Linux or Windows
- Region closest to your users
- App Service Plan

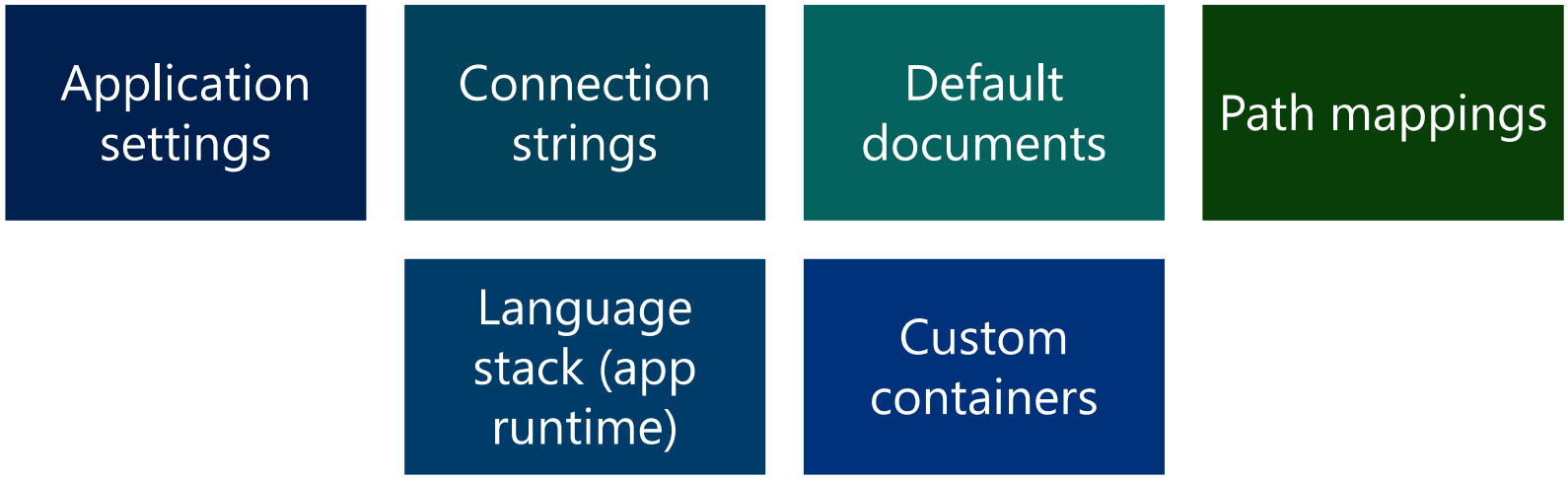


Demo & Lab:
Create & Deploy First Web App



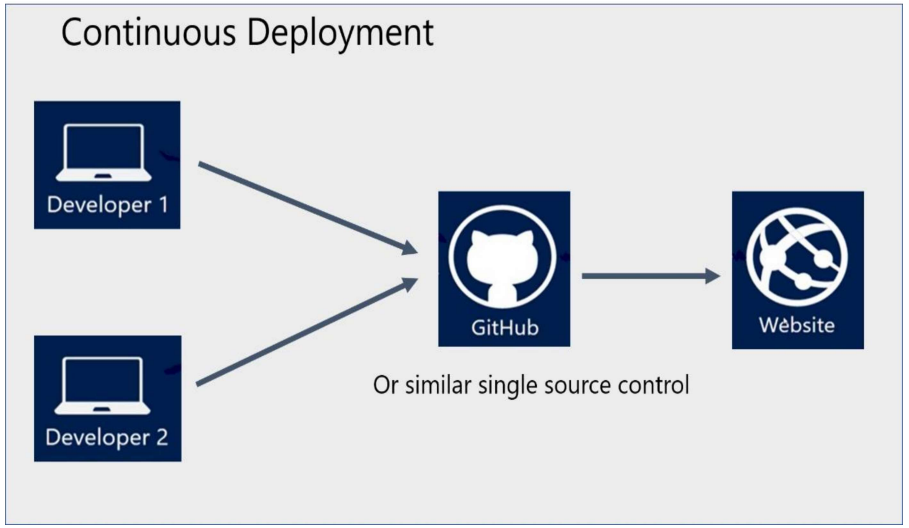
App Service settings

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



Continuous Deployment

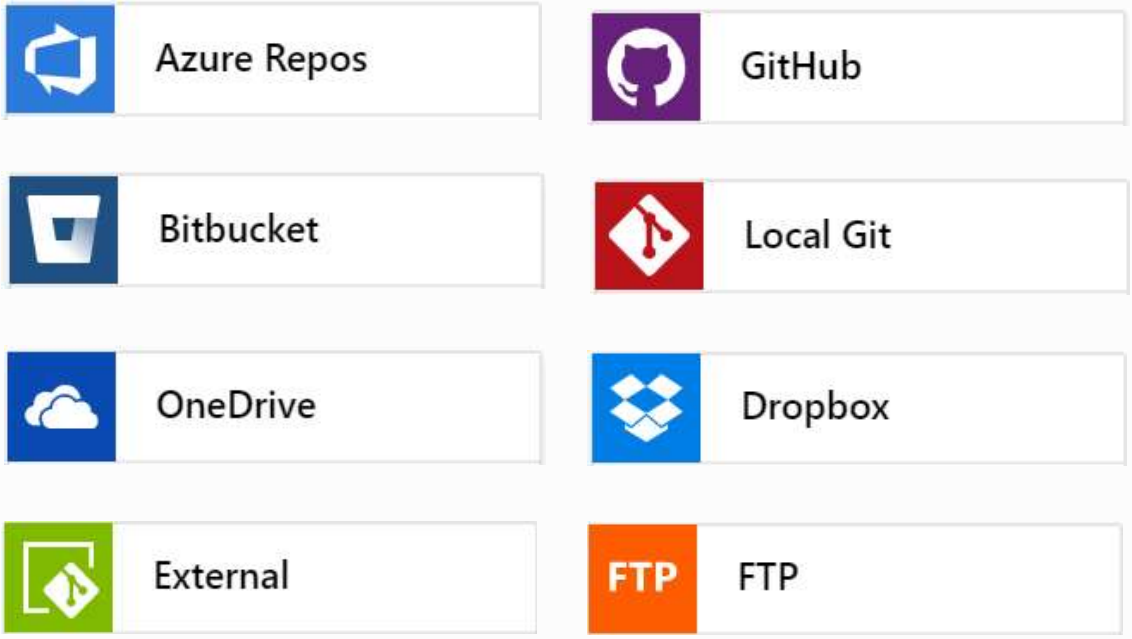
- Work in a single source control
- Whenever code updates are pushed to the source control, then the website or web app will automatically pick up the updates
- A continuous deployment workflow publishes the most recent updates from a project
- Use the portal for continuous deployments from GitHub, Bitbucket, or Visual Studio Team Services



Azure App Service Staging Environment

Publishing from Source Control

Supported Options



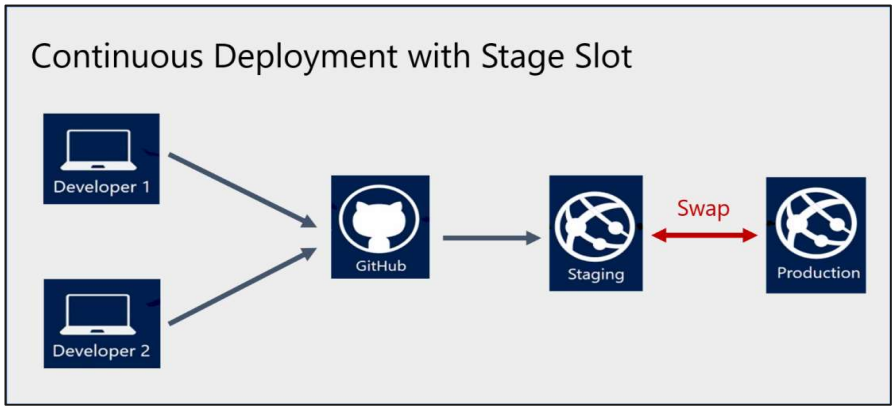
Azure App Service Staging Environment

- When you have a successful or business-critical web app, you need to update it to respond to business changes, user demands, or security issues. But you can't allow service interruptions.

- Suppose you work for a company that runs a popular social media web platform. The user interface for this platform is set up as an ASP.NET Core MVC web app that's hosted in Azure App Service. You regularly update the app's source code and roll out the updates to production. These updates occasionally cause problems when testers fail to catch bugs.

- You want a way to deploy a new version of the app without downtime or a service interruption. You also want to be able to rapidly roll back a new deployment to the previous version if it causes problems.

Deployment Slots

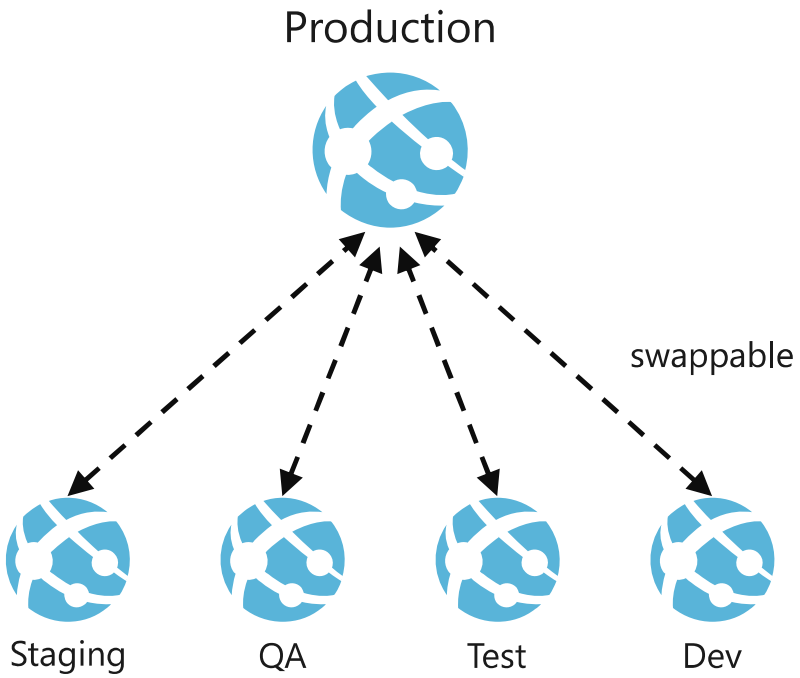


Service Plan	Slots
Free, Shared, Basic	0
Standard	Up to 5
Premium	Up to 20
Isolated	Up to 20

- 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

Deployment Slots

- Live apps with their own:
  - Host names
  - Content
  - Configuration
- Can be swapped between each other. For example:
  - Staging ↔ Production
  - Production ↔ Staging
  - Dev ↔ Test
  - Test ↔ QA
  - QA ↔ Staging



Deployment Slots - Example

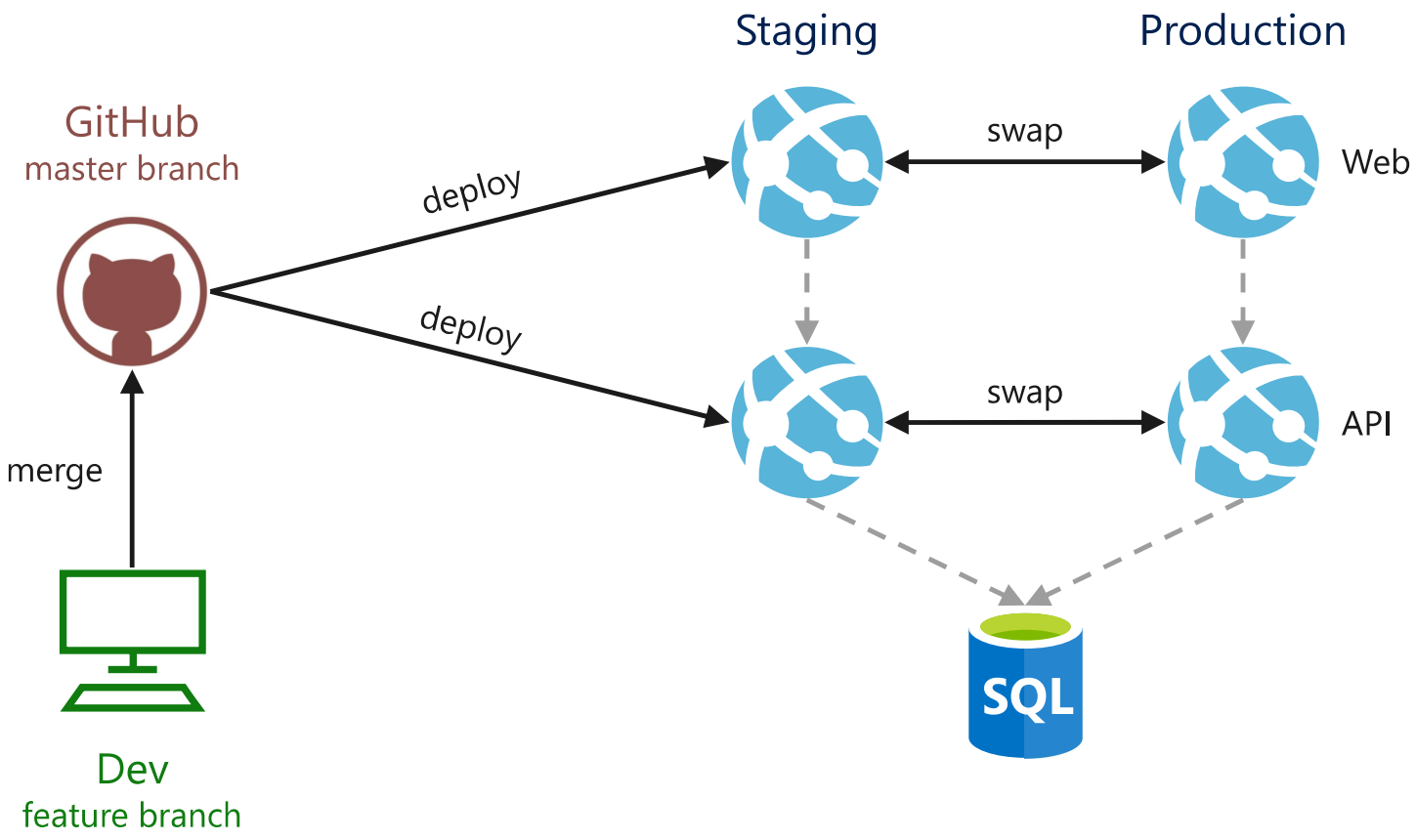
- **Production Web App URL**

contoso.azurewebsites.net

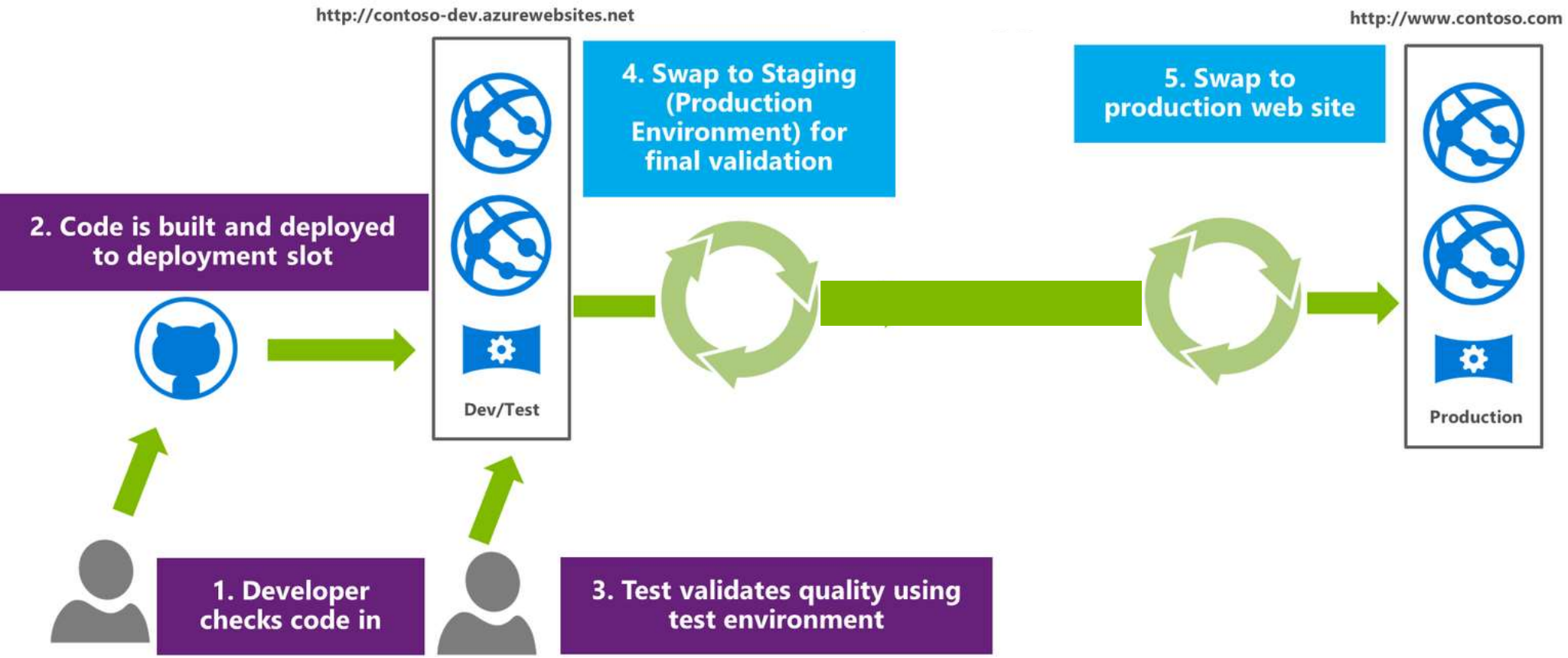
- **Deployment Slot Web App URL**

contoso-dev.azurewebsites.net

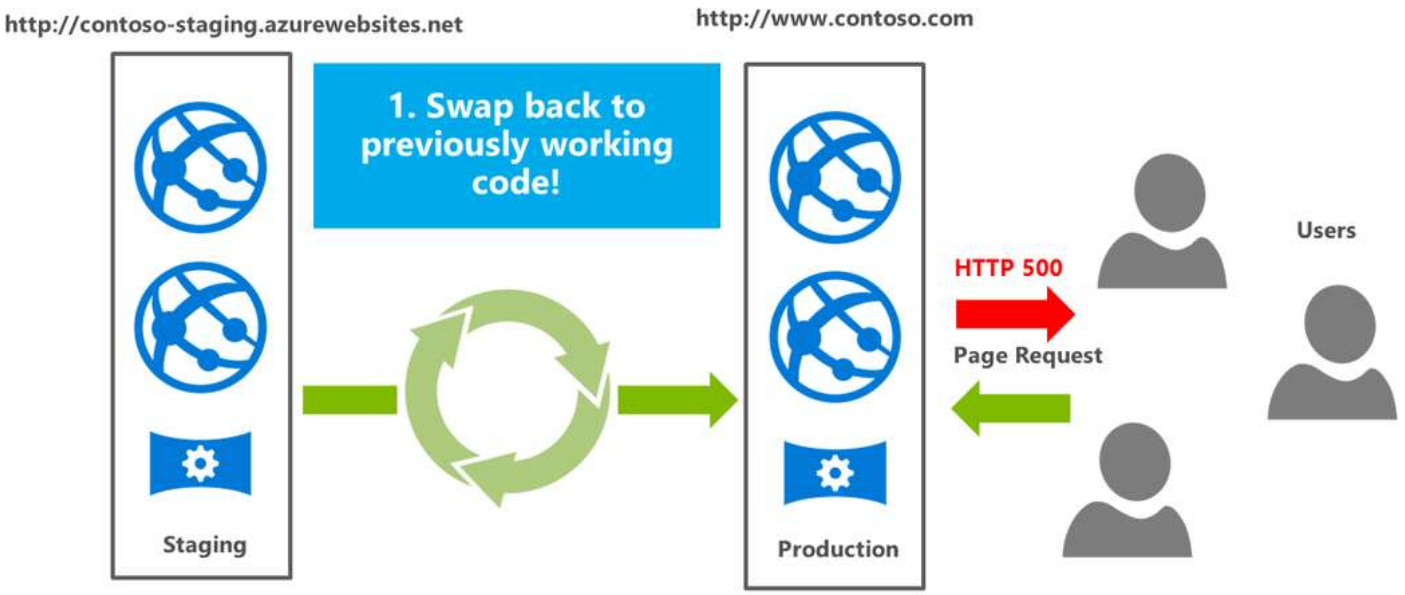
Modern deployment workflow



Staged Publishing

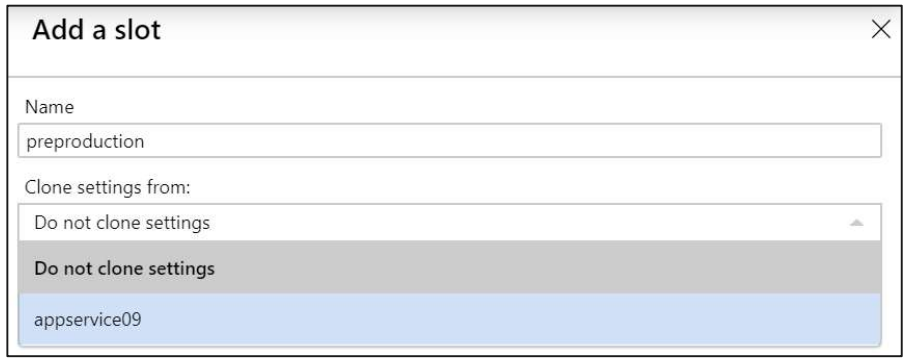


Rollback (Oops)

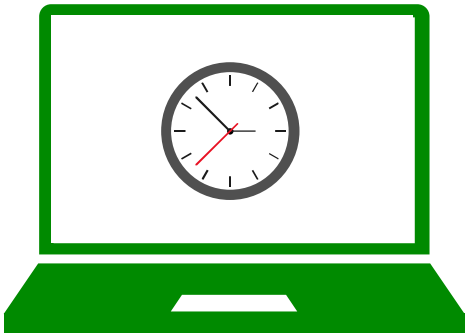


Creating Deployment Slots

- A new slot can be empty or cloned
- 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

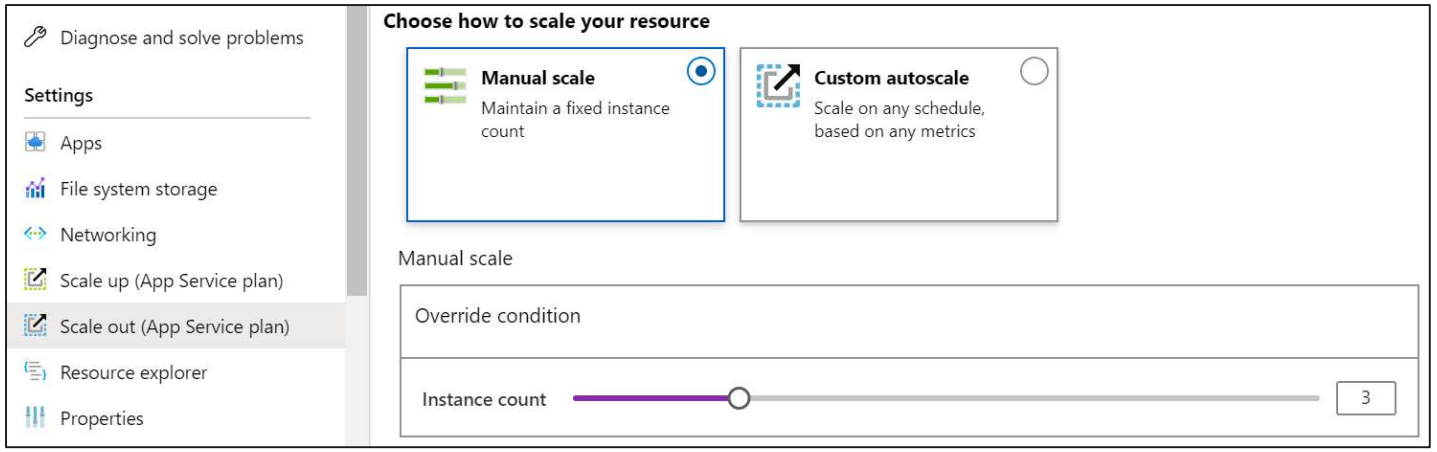


Lab:
Configure Deployment Slots



Scaling App Service apps

App Service Plan Scaling



- 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)
  - Autoscale (based on predefined rules and schedules)

Autoscale

- A primary advantage of the cloud is elastic scaling (the ability to use as much capacity as you need):
  - Scaling out as load increases
  - Scaling in when the extra capacity is not needed
- Many Microsoft Azure services provide the capability to scale both manually and automatically
- Autoscale refers to the capability of many of these services to monitor the application instances and automatically scale appropriately to handle the current usage of the application:
  - Using autoscale, your cloud service can scale out and in to exactly match the amount of instances needed for your specific computing pattern

Scaling Up vs. Scaling Out

Scale Up

Vary the size

1 Core w/ 1.75 GB RAM
2 Cores w/ 3.5 GB RAM
4 Cores w/ 7 GB RAM

Vertical Scaling

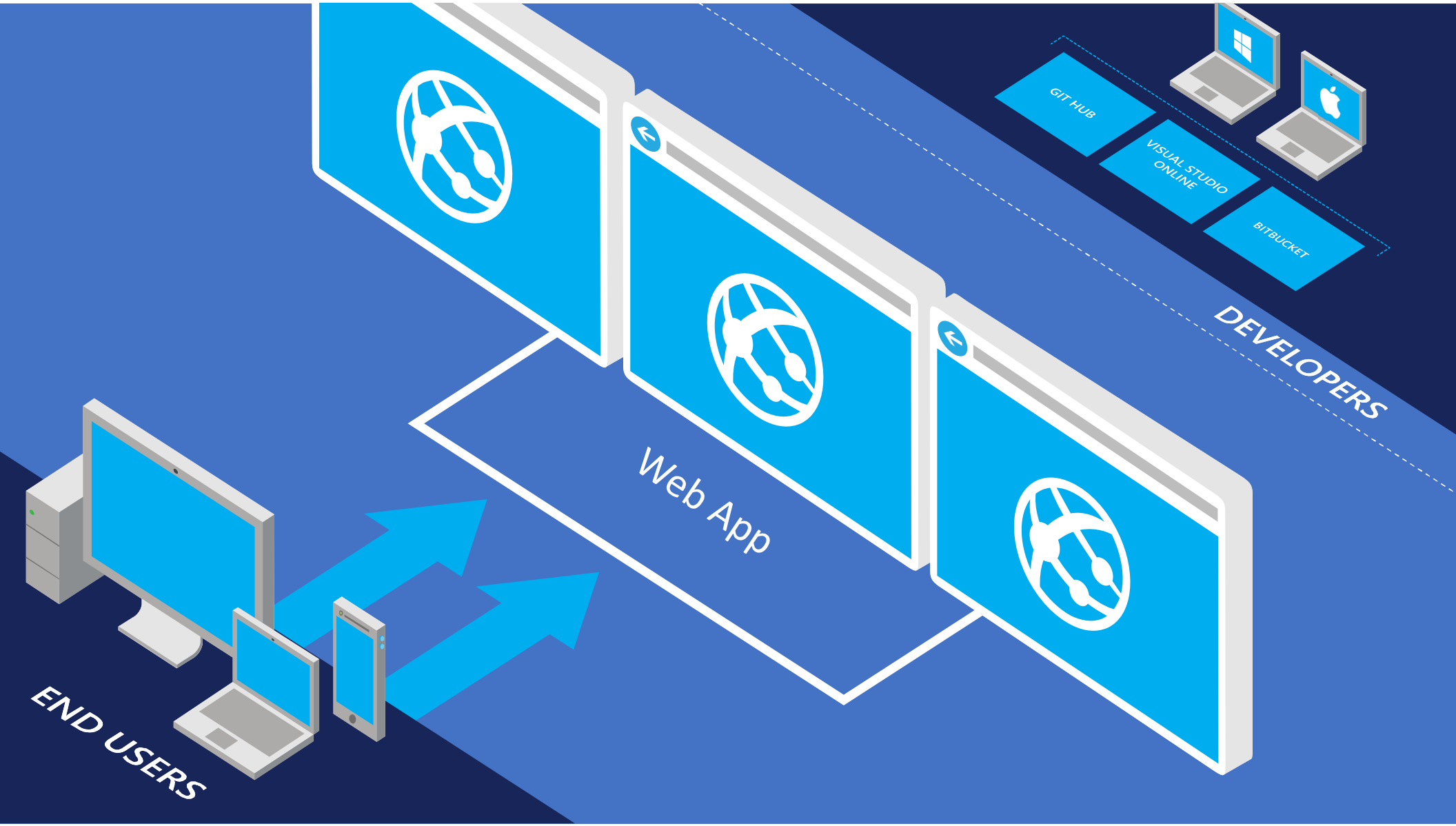
Scale Out

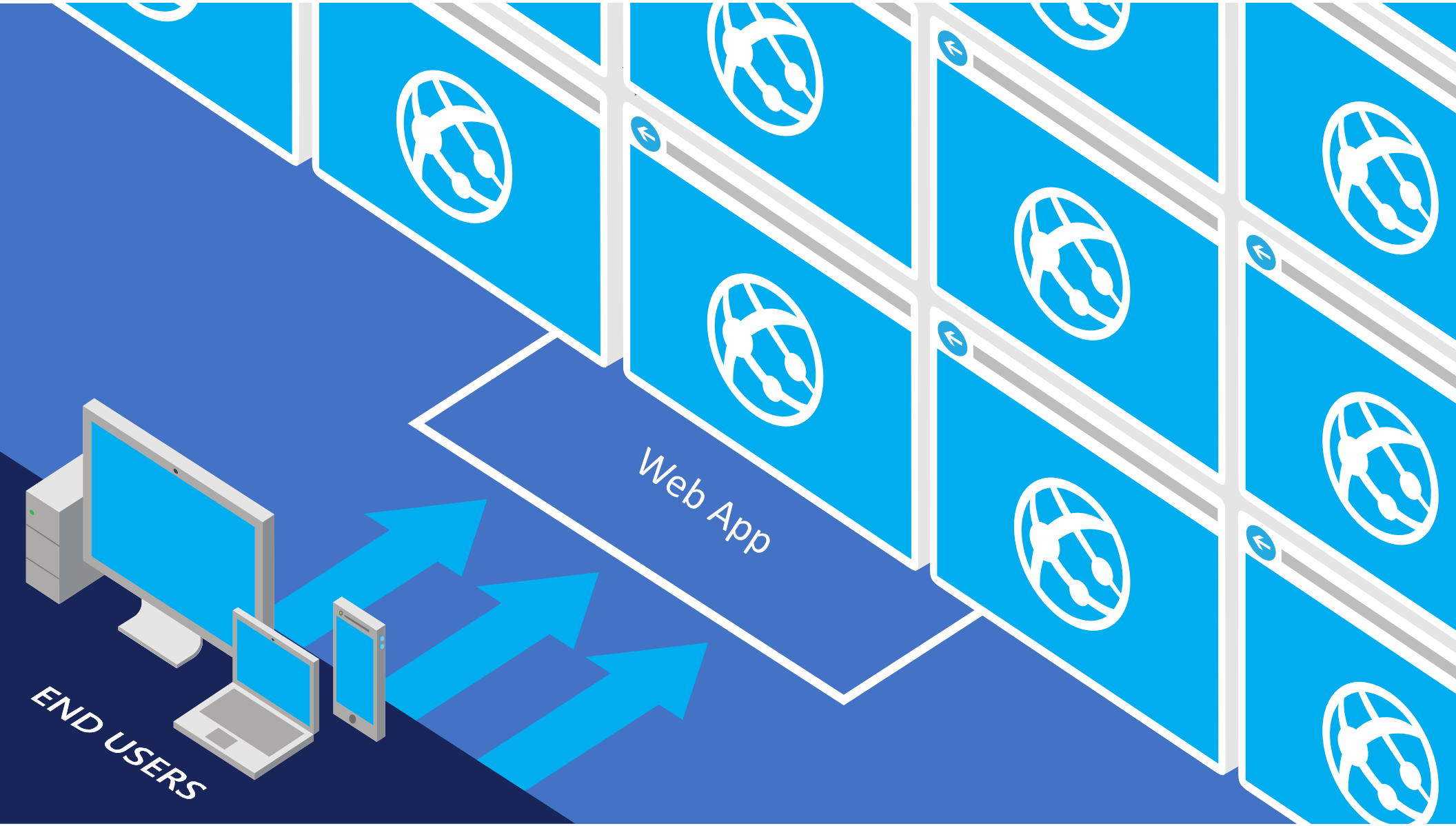
Vary the count

Max 3 instances*
Max 10 instances
*Max 20/50** instances*

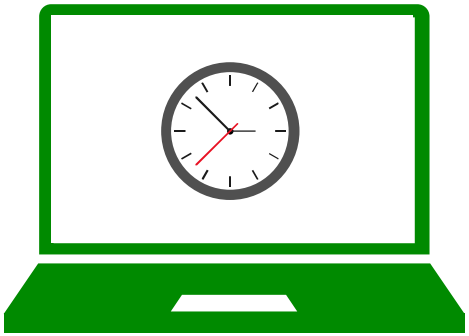
Horizontal Scaling


GIT HUB
VISUAL STUDIO ONLINE
BITBUCKET
DEVELOPERS
Web App
END USERS


GIT HUB
VISUAL STUDIO ONLINE
BITBUCKET
DEVELOPERS
Web App
END USERS

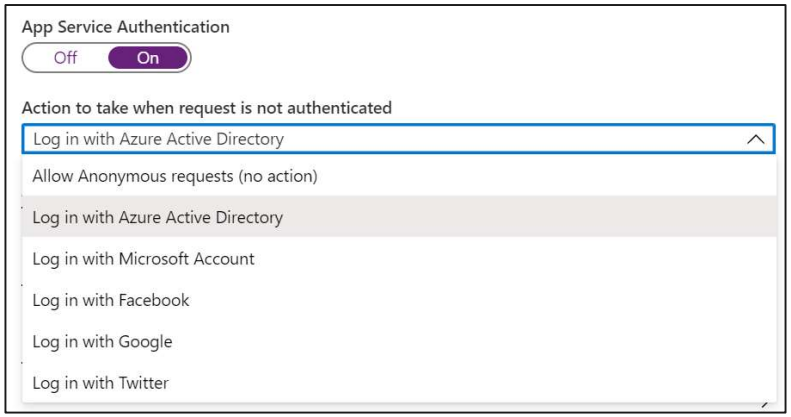

Web App
END USERS

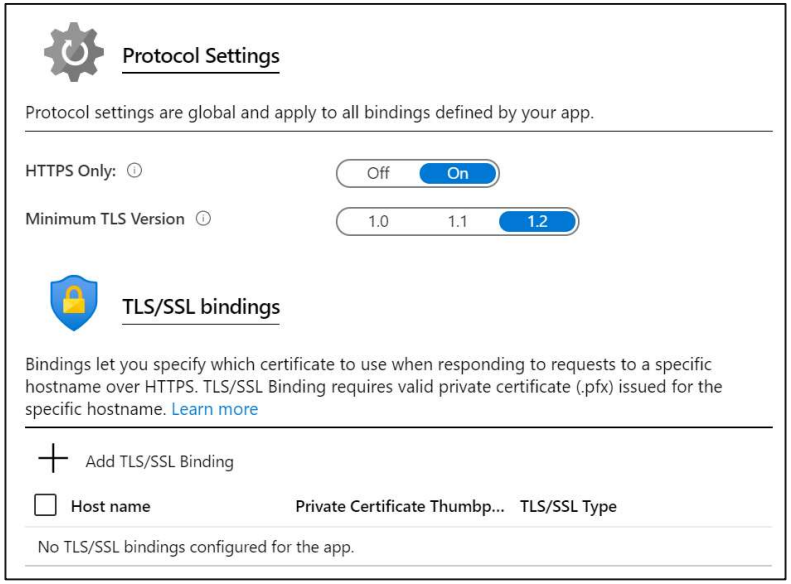
Demo:
Scaling Web App



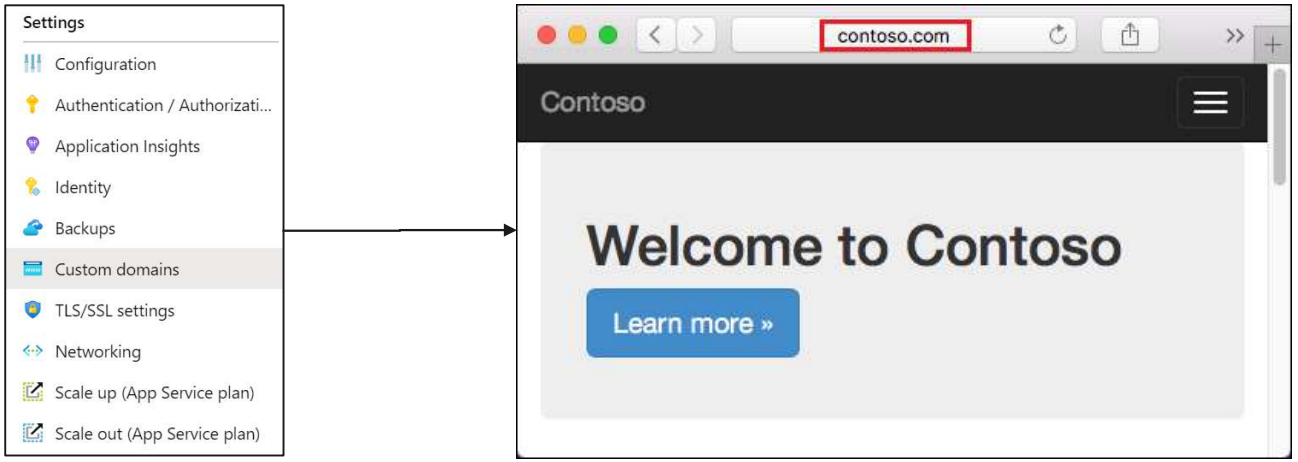
Securing an App Service

- 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





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