70-532 Exam Prep Session 4 of 5

Design and Implement Azure Web and Mobile Services

Agenda

- Azure Web Apps
- Hosting Web Applications in Azure
- Configuring an Azure Web App
- Azure API & Logic Apps
- Azure Mobile Apps
- Monitoring Web Apps and Web jobs
- Azure Container Services

Scenario

- You have an events administration application that is currently used by a static set of users. The application must be upgraded to handle all the users in your organization in the future. You need a hosting option that provides the least amount of friction so that you can immediately deploy the web application for immediate use. You also need the hosting option to be flexible enough so that it allows you to configure and scale the web application, thereby ensuring that it can handle an increase in the number of administrative users. For these reasons, you have chosen to deploy the application to Web Apps. Web Apps will also give you the flexibility to integrate your application with Azure Active Directory in the future so that all of your organization's users can access the application.

Azure Web Apps Overview

- Simple, scalable hosting for websites in Windows Azure with the following benefits:
  - Provides a quick way to host your web application in the cloud
  - Allows you to scale your web app without being required to redesign for scalability
  - Integrates with Visual Studio
  - Provides an open platform for many different programming languages

Web App Tiers

- Web Apps can be scaled to run in one of the following modes:

Free

- Shared compute resources
- Limited bandwidth and CPU time
- Limited customization options

Shared

- Shared compute resources
- No upper-limit to bandwidth and CPU time
- Additional customization options

Web App Tiers (continued)

- Web Apps can be scaled to run in one of the three following modes:

Basic

- Reserved instance for multiple web applications
- Websites are pooled under the same instance

Standard

- Reserved instance for multiple web applications
- Websites are pooled under the same instance
- Supports auto-scale

Web App Tiers

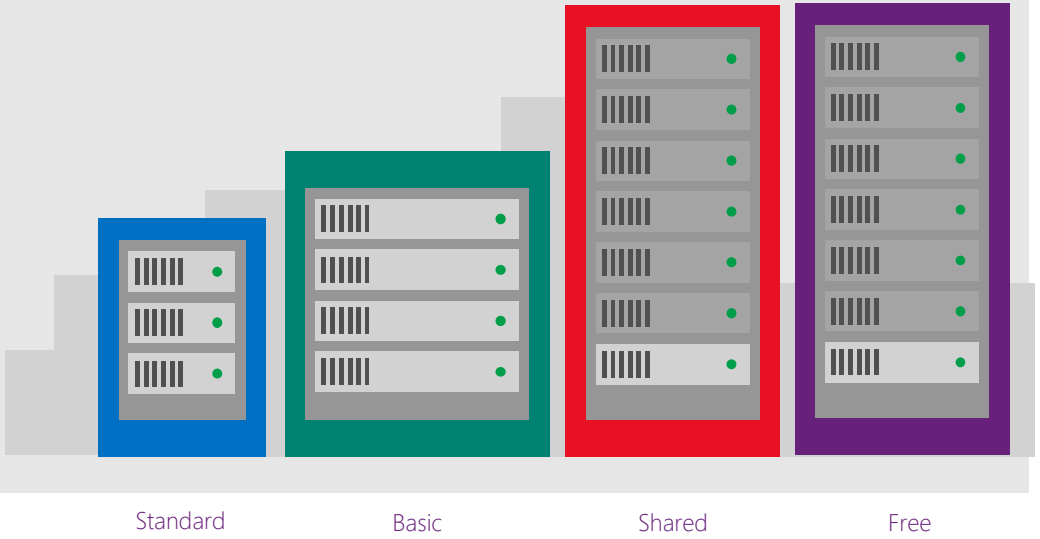
- Web Apps can be scaled to run in one of the following modes:

Premium

- Most advanced Plans, giving you the best performance, and the most features

Web App Tiers (continued)

- To host multiple Web Apps, use a Standard or Basic Web Hosting Plan
  - Reserves a virtual machine for your Websites
  - Billed based on number of virtual machine instances and not number of Websites



Demo: Creating a Web App

- In this demo, you will learn how to:
  - Create a Web App instance by using the Azure Portal
  - Create a SQL Database instance by using the Portal
  - Manage the Web App and SQL Database instances as a Resource Group

Web App Configuration

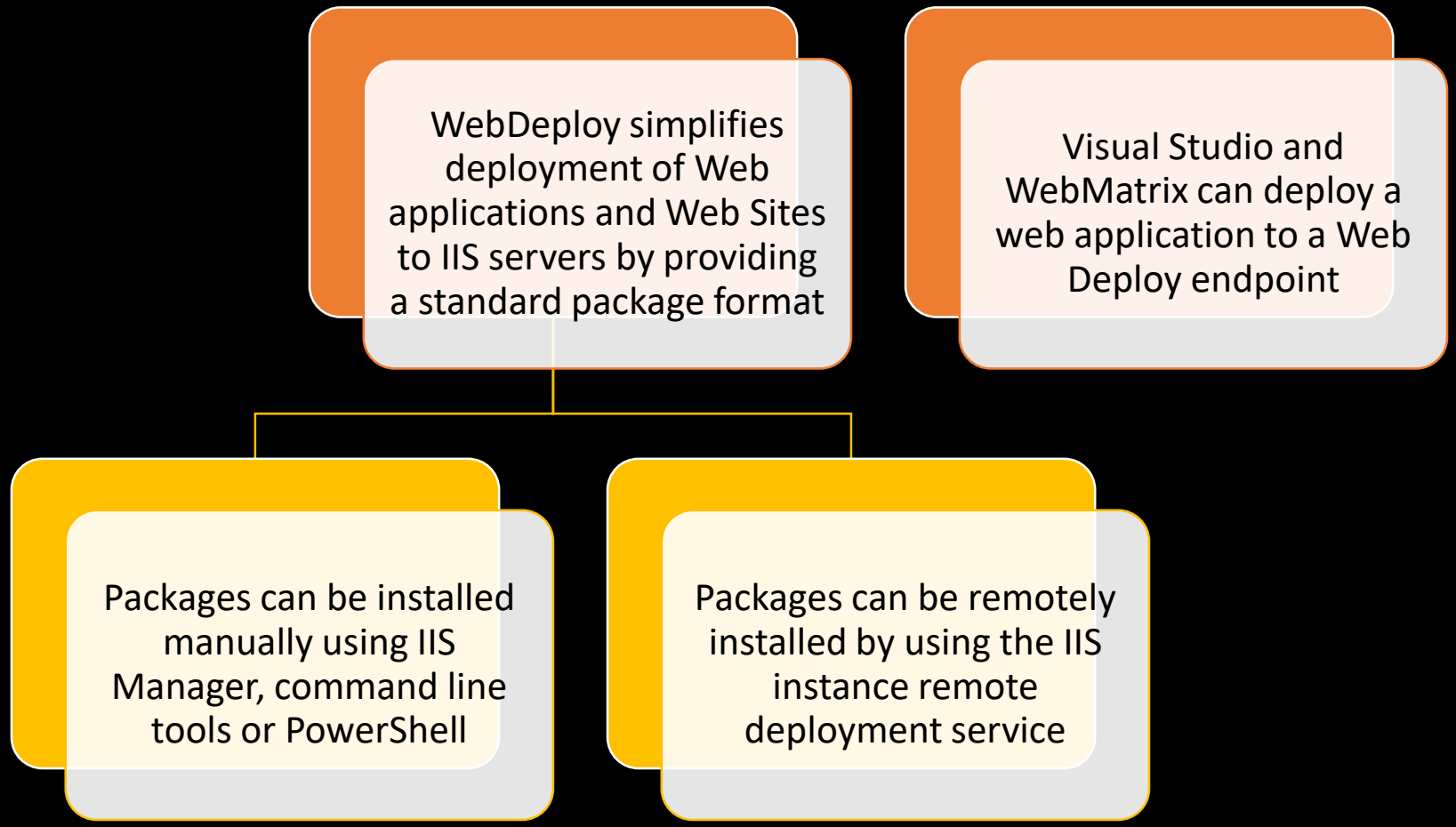
The Web App deployment package and it's configuration are both stored in an external store.

App Settings and Connection Strings are intercepted and changed in the application during startup

Applications can be scaled by:

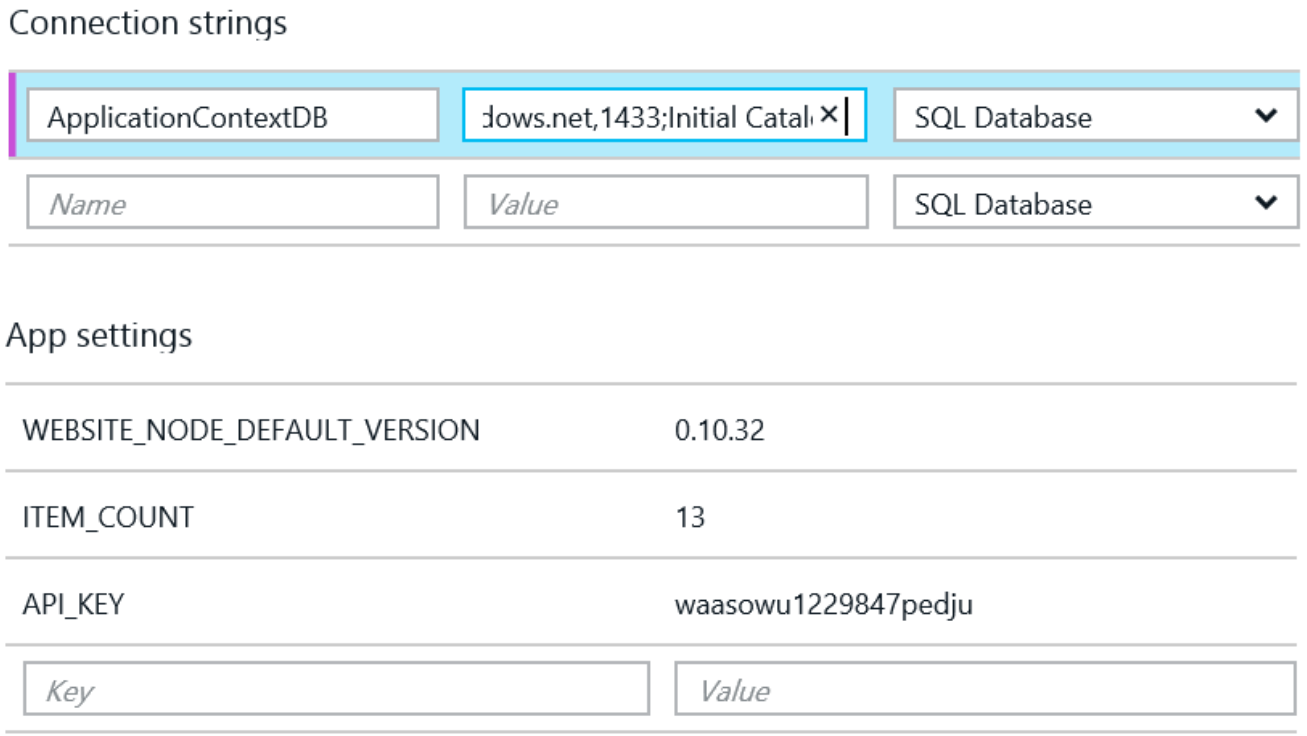
- Creating IIS web sites using the Web Deploy package
- Applying configuration options from the external store

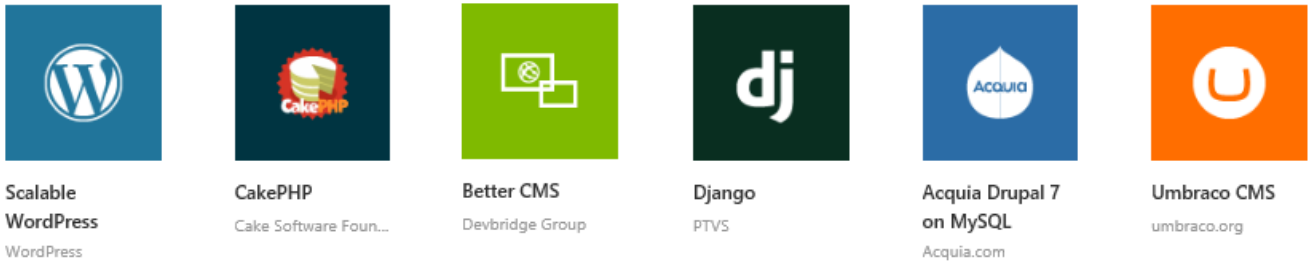
The Web Deploy Protocol



App Settings and Connection Strings

- Application settings and connection strings can be managed in the portal.
  - Connection strings are hidden by default.





- Create a Web App using a pre-built template from the Azure Marketplace.
- There are over 30 open source applications, frameworks and templates in the Marketplace:

Prebuilt Web App Templates

Demo: Deploying a MarketPlace Web App

- In this demo, you will learn how to:
  - Deploy a Web App from a MarketPlace template

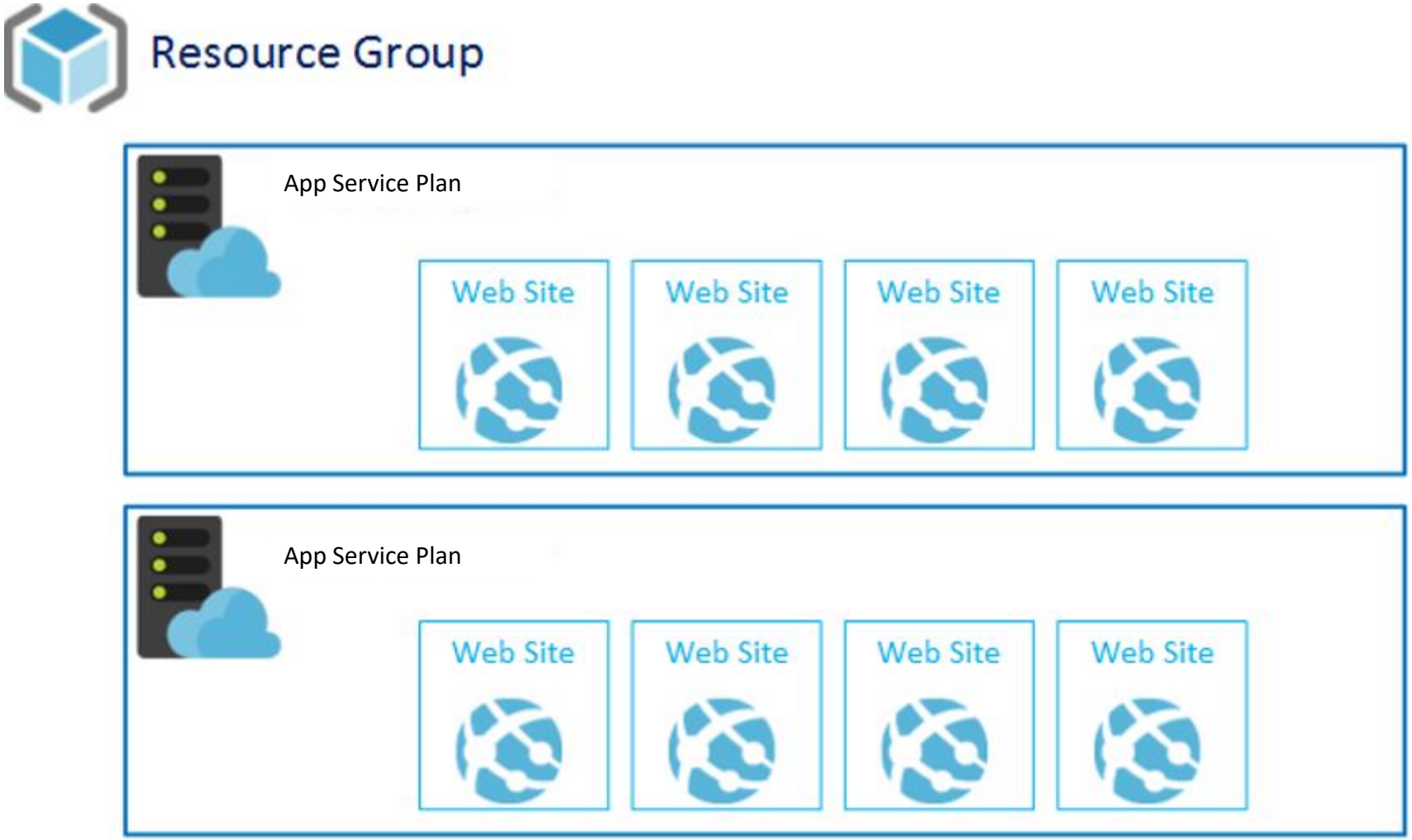
Web App Configuration (continued)

- Web Apps share functionality with IIS web sites
  - For Free and Shared instances, Web Sites are implemented similar to IIS web sites
  - For Standard instances, a reserved virtual machine is made available and each individual Website is similar to an IIS web site
  - Azure Websites can also be managed remotely using the IIS Manager

App Service Plans

- App Service Plans can logically group Web Apps within a subscription.
  - Characteristics such as features, capacity and tiers are shared amongst the Website instance in the group.
- Multiple App Service Plans can exist in a single Resource Group and multiple Web Appscan exist in a single App Service Plan.

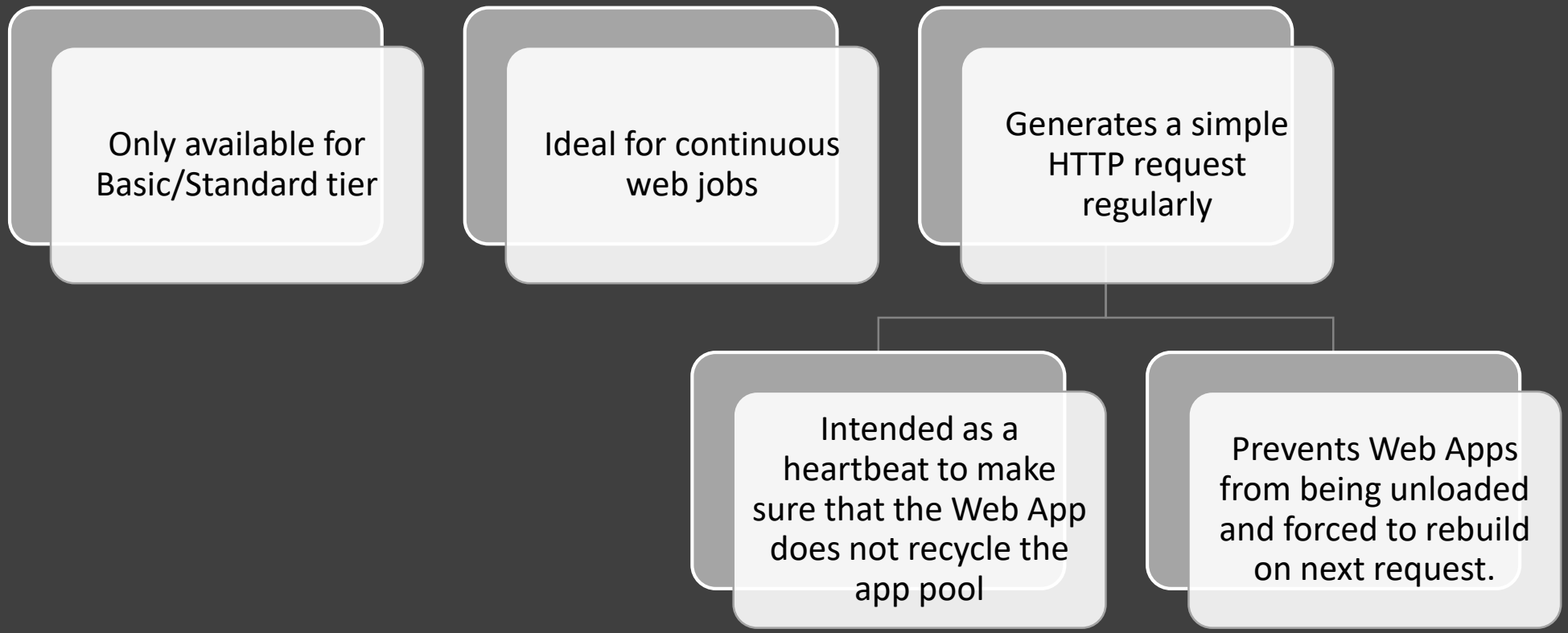
App Service Plans (continued)



App Service Plans (continued)

- App Service Plans are a required parameter when creating a new Web App instance.
  - You can chose to add a new Web App to an existing plan.
  - You can also define a new App Service Plan as a part of creating a new Web App.

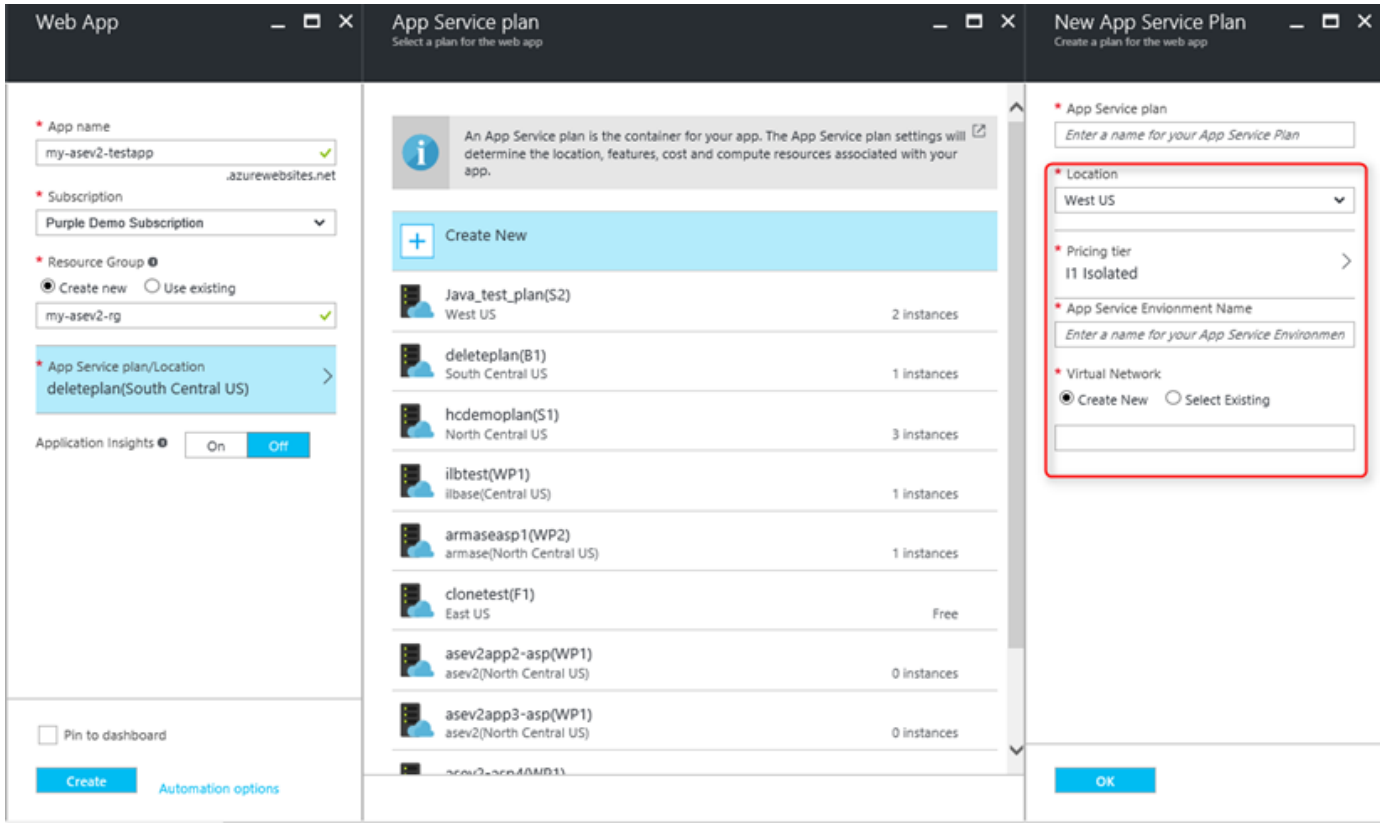
AlwaysOn


Only available for Basic/Standard tier
Ideal for continuous web jobs
Generates a simple HTTP request regularly
Intended as a heartbeat to make sure that the Web App does not recycle the app pool
Prevents Web Apps from being unloaded and forced to rebuild on next request.

App Service Isolated

- Allow you to run apps in your dedicated virtual networks with even better scale and performance through an intuitive user experience

New



Domain Names

Standard domain

- [http/https]://<sitename>.azurewebsites.net

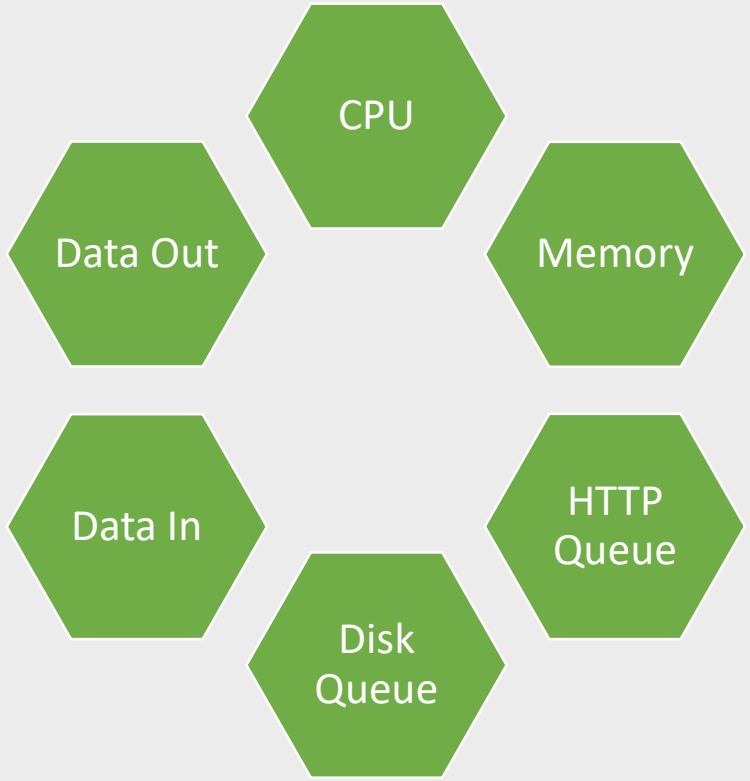
In Shared, Basic or Standard mode, you can configure the Web App to use a custom domain

- This involves managing the A and CNAME records with your registrar

Traffic Manager supports custom domain names

Autoscaling Web Apps

- Scaling rules are specific to a schedule
- Performance scaling can be configured using various metrics:



Performance Scaling

- The Instance Range setting defines a minimum and maximum quantity of instances
- You can use the CPU Percentage setting to set a CPU range
  - Minimum defines a threshold when instances should be removed
  - Maximum defines a threshold when instanced should be added

Demo: Web App AutoScale

- In this demo, you will learn how to:
  - Configure Web App Autoscale feature

Implementing WebJobs

WebJobs are scripts that run:

- Continuously
- Triggered
  - Scheduled
  - Manual

WebJobs can be:

- Batch files (.cmd, .bat)
- PowerShell scripts (.ps1)
- Bash shell scripts (.sh)
- PHP scripts (.php)
- Python scripts (.py)
- Node.js JavaScripts (.js)
- JavaScript (.jar)

Process events with Serverless code.

Make composing Cloud Apps insanely easy

Develop Functions in C#, Node.js, F#, Python, PHP, Batch and more

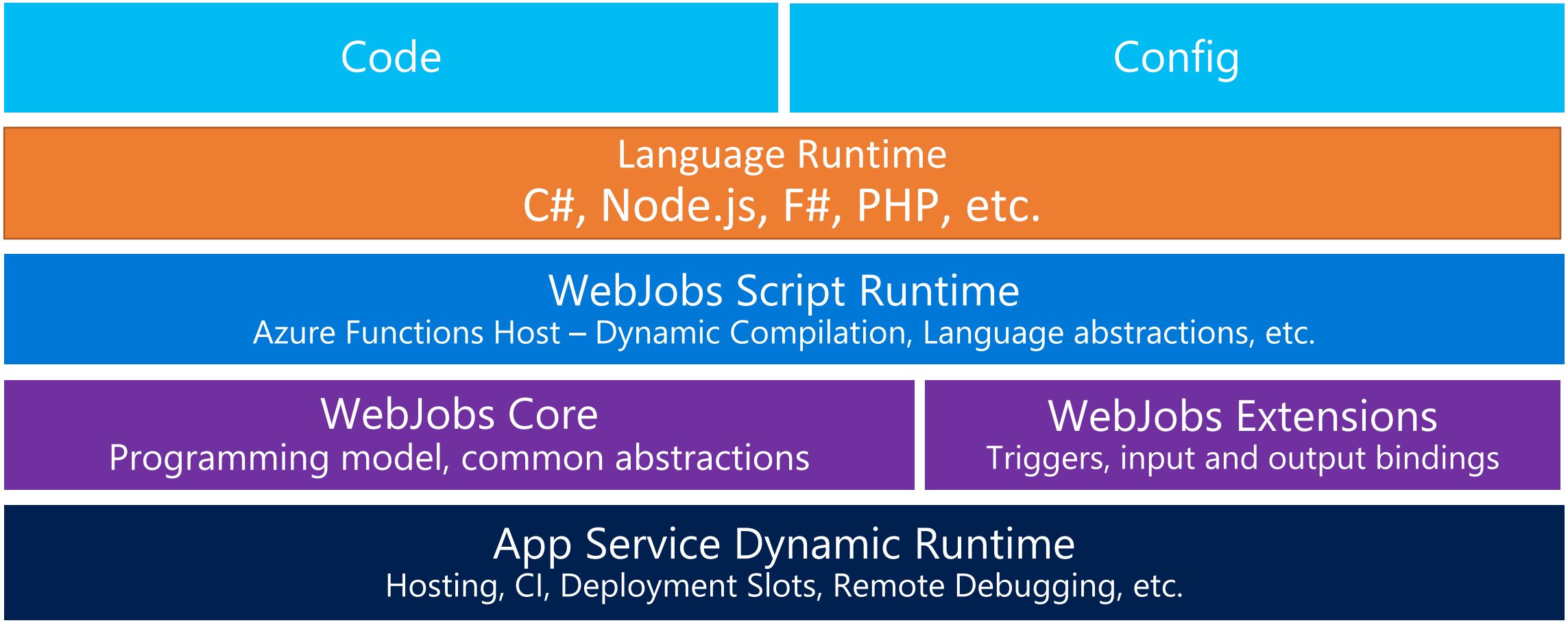
Easily schedule event-driven tasks across services

Expose Functions as HTTP API endpoints

Scale Functions based on customer demand

Easily integrate with Logic Apps

Built on top of App Service and WebJobs SDK

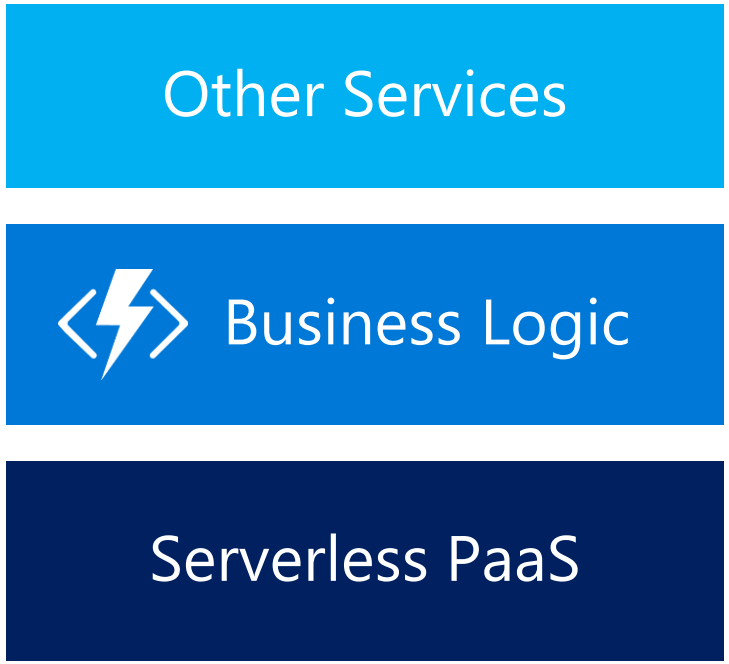


Demo: Azure Web Job

- In this demo, you will learn how to:
  - Run / Validate an Azure Web Job
  - Publish an Azure Web Job to Azure Web Apps

Dual abstraction

- Serverless compute abstracts away the compute
- Bindings abstract away the services you interact with



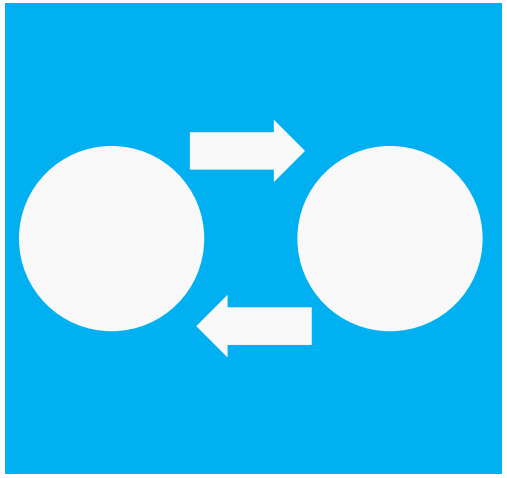
What is the “Functions” programming model?

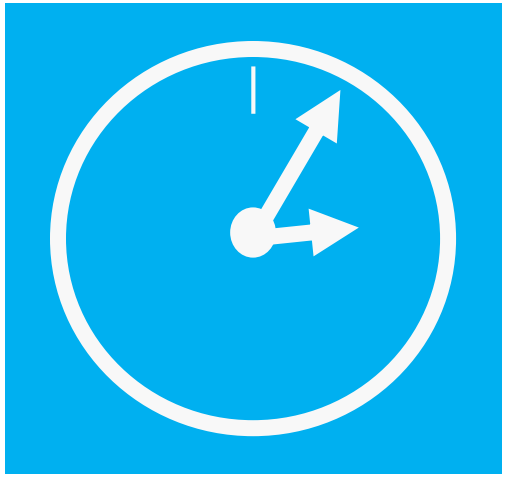
- Function as the unit of work
- Functions are executed; they start and finish
- Functions have inputs and outputs

```
public async static Task ProcessQueueMessageAsyncCancellationToken(
    [QueueTrigger("blobcopyqueue")] string blobName,
    [Blob("textblobs/{queueTrigger}",FileAccess.Read)] Stream blobInput,
    [Blob("textblobs/{queueTrigger}-new",FileAccess.Write)] Stream blobOutput,
    CancellationToken token)
{
    await blobInput.CopyToAsync(blobOutput, 4096, token);
}
```

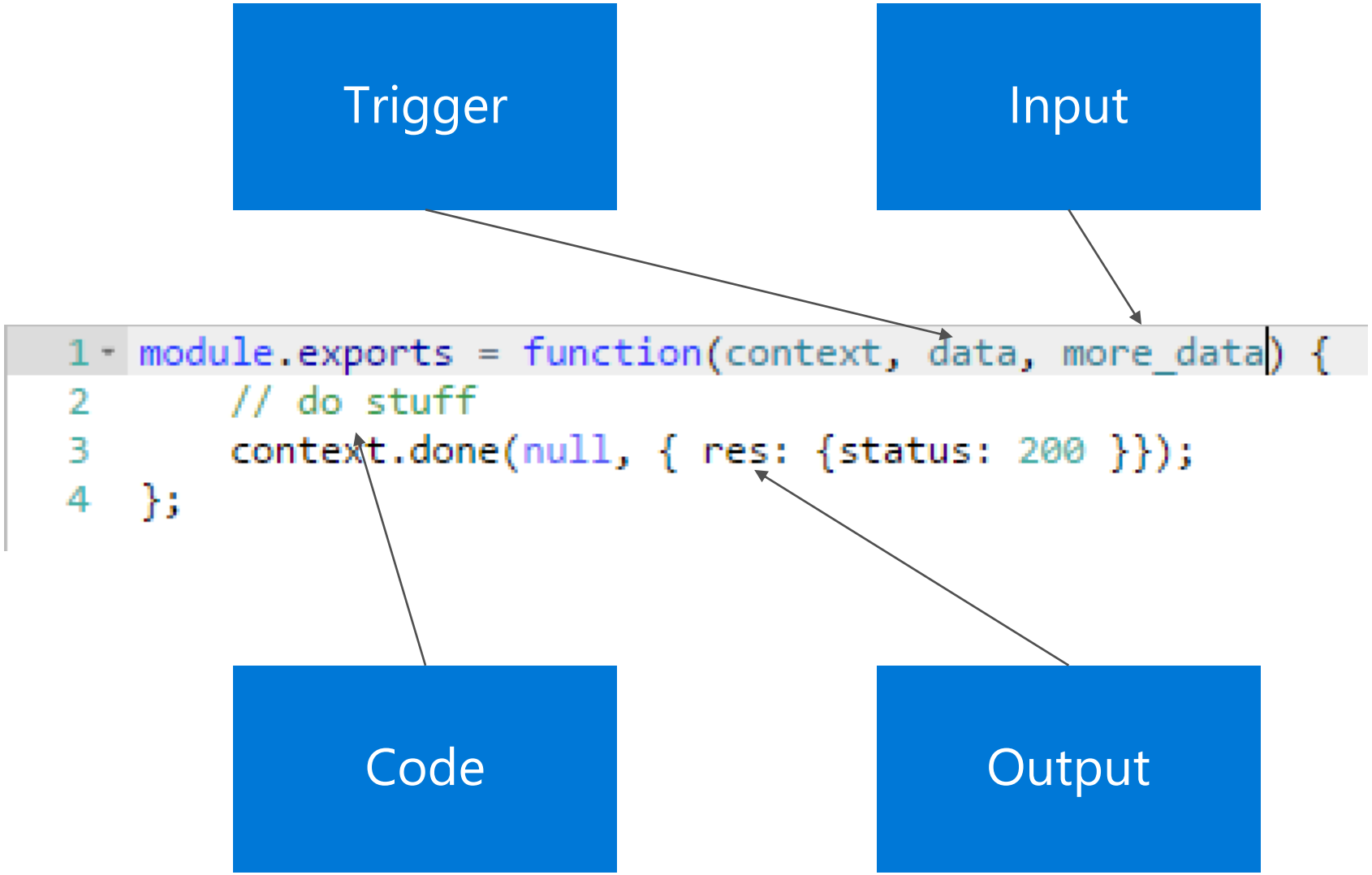
Best practices for the “Functions” model

- Functions *should* “do one thing”
- Functions *should* be idempotent
- Functions *should* finish as quickly as possible





Functions programming concepts



Demo: Azure Function

- In this demo, you will learn how to:
  - Run / Validate your first Azure Function

WebJobs SDK vs Azure Functions

- Generalizing

  WebJobs SDK == more freedom -> more work

  Azure Functions == less responsibilities -> less work

- Commonalities

  Programming model differences

  Hosting model differences

Commonalities

Both use the ”Function” oriented programming model

Both support ”bindings” for trigger/input/output

Both support WebJobs SDK extensions model

Both support external libraries being used

Both can run locally and be debugged

Both have runtime telemetry via the WebJobs Dashboard

Programming Model Differences

WebJobs SDK

- ✓ C# only
- ✓ Attributes for configuring bindings
- ✓ Traditional .NET developer experience (Visual Studio, NuGet, MSBuild)
- ✓ Many functions per class
- ✓ Can access and manipulate many core SDK features
- ✓ Can’t listen for HTTP requests*

Azure Functions

- ✓ C# & Node.js + more
- ✓ Config files for configuring bindings
- ✓ More diverse development experience (Web portal, VSCode, dynamically builds itself)
- ✓ Limited access to manipulate core SDK features, but (C# only) still some access
- ✓ Supports HTTP!

Hosting Model Differences

- **WebJobs SDK**
  - You configure host
  - Build a console app which is run
- **WebJobs and Dedicated App Service plans**
  - Runs the service in the background of Web/Mobile/API app
  - Runs any console app (not just SDK based ones)
  - You manage scale
- **Azure Functions**
  - Limited control over the host
  - Just give it your code/config
- **Function App + Dynamic App Service Plans**
  - Function app owns the whole host, including web frontend.
  - Only runs Azure Functions stuff – no other things
  - Scale is managed for you

Configuring application and site diagnostics

Configure the following application logging settings:

- Log storage location
- Logging level
- Retention period

Configure the following site diagnostics settings:

- Web server logging
- Detailed error messages
- Failed request tracing

Monitoring web apps

Access diagnostic logs by using:

- FTP
- Azure PowerShell
- Azure CLI

View logs in Visual Studio by using Application Insights

Monitor web apps in the Azure portal by:

- Adding metrics
- Configuring alerts:
  - Email notifications
  - Webhooks
  - Logic apps

Advanced Monitoring Options

You can also monitor Azure Web Apps using advanced monitoring solutions available with Azure Resource Manager:

- Azure Application Insights
- Azure Monitor
- Azure Resource Health
- Azure Security Center
- Log Analytics (OMS)

Demo: Monitoring Azure Web Apps

- In this demo, you will learn how to:
  - Use Kudu logging
  - Application Insights
  - Azure Monitor
  - Azure Security Center

Overview of API Apps

- Features of API Apps:
  - Visual Studio integration
  - Consumption model
  - Bring your existing API
  - Support for Swagger metadata
  - Support for cross-origin resource sharing
  - Triggering actions

Why Azure API Apps?

Benefits of App Services

- Automatic OS patching
- Enterprise grade security
- High availability
- Support for many platforms & languages
- Auto scaling and load balancing
- WebJobs for background processing
- Easy deployment, including continuous delivery
- Access on-premises data

Additional Benefits

Bring your API as-is

Simple access control

Connectivity to SaaS platforms

Swagger metadata

Logic App integration

Visual Studio tooling and support

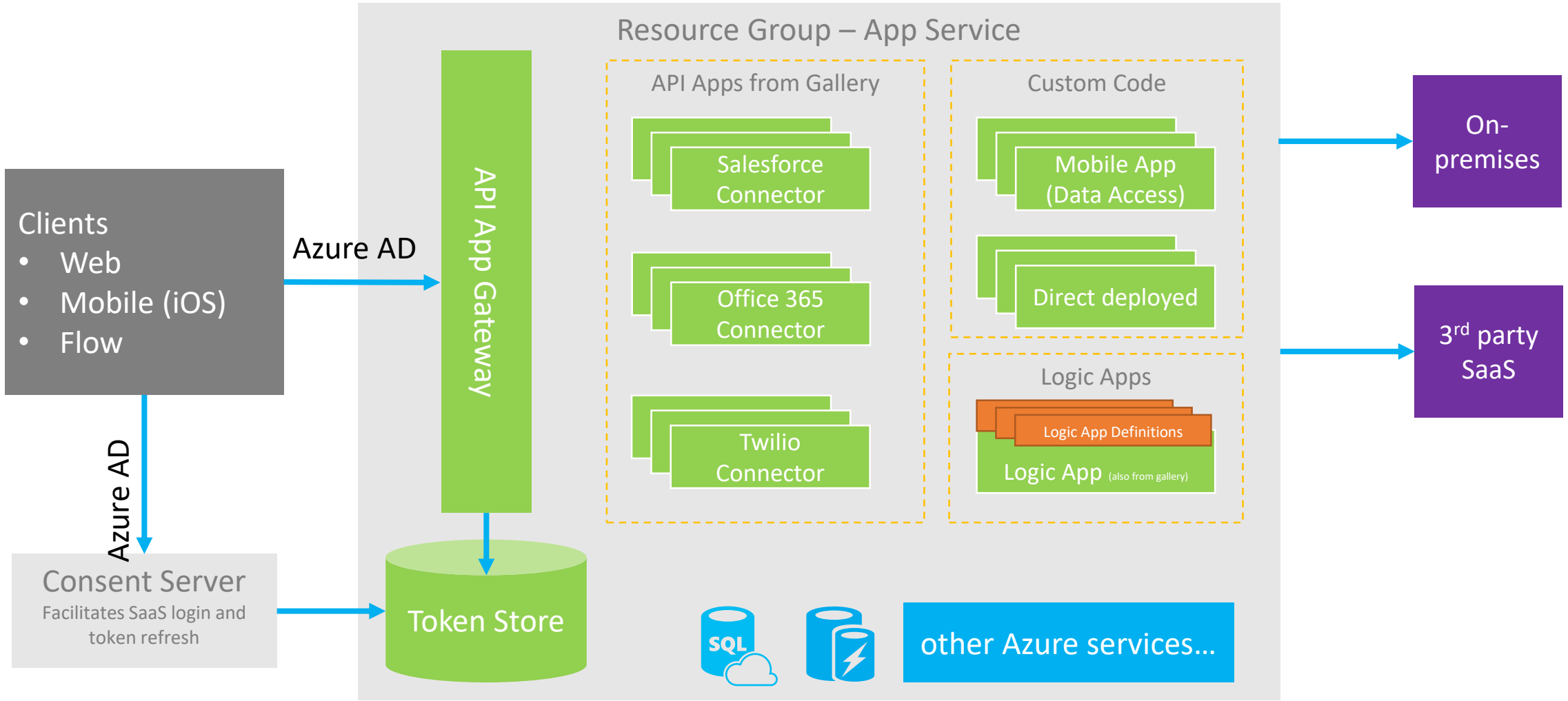
Public and private marketplaces

Automatic dependency deployment

Automatic updates

API Apps Architecture Example

Backend is an API App with APIs from the gallery, as well as custom code. It is registered with and protected by AAD. Logins to downstream SaaS are facilitated by a consent server and token store, using a server flow.



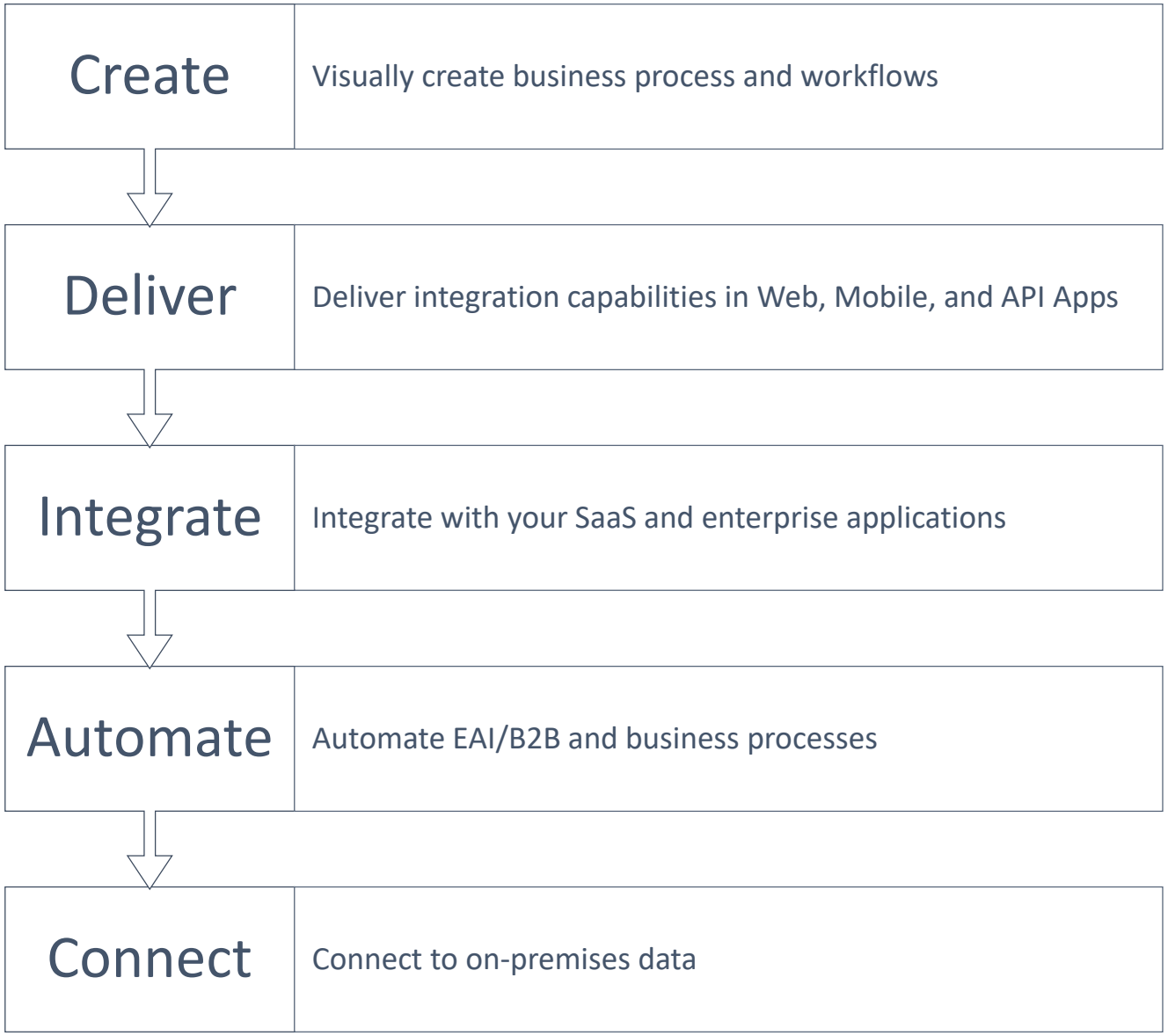
Demo: Azure API App

- In this demo, you will learn how to:
  - Manage and configure an Azure API App

Overview of Logic Apps

- Logic apps integrate apps by using connectors:
  - Core connectors:
    - Office 365 Connector
    - Microsoft OneDrive Connector
    - Yammer Connector
    - Facebook Connector
    - HTTP Connector
  - Enterprise integration connectors:
    - SAP
    - Oracle
    - DB2
    - Informix

Logic Apps Allows Developer to:



Logic Apps SaaS API Connectors

Connectors

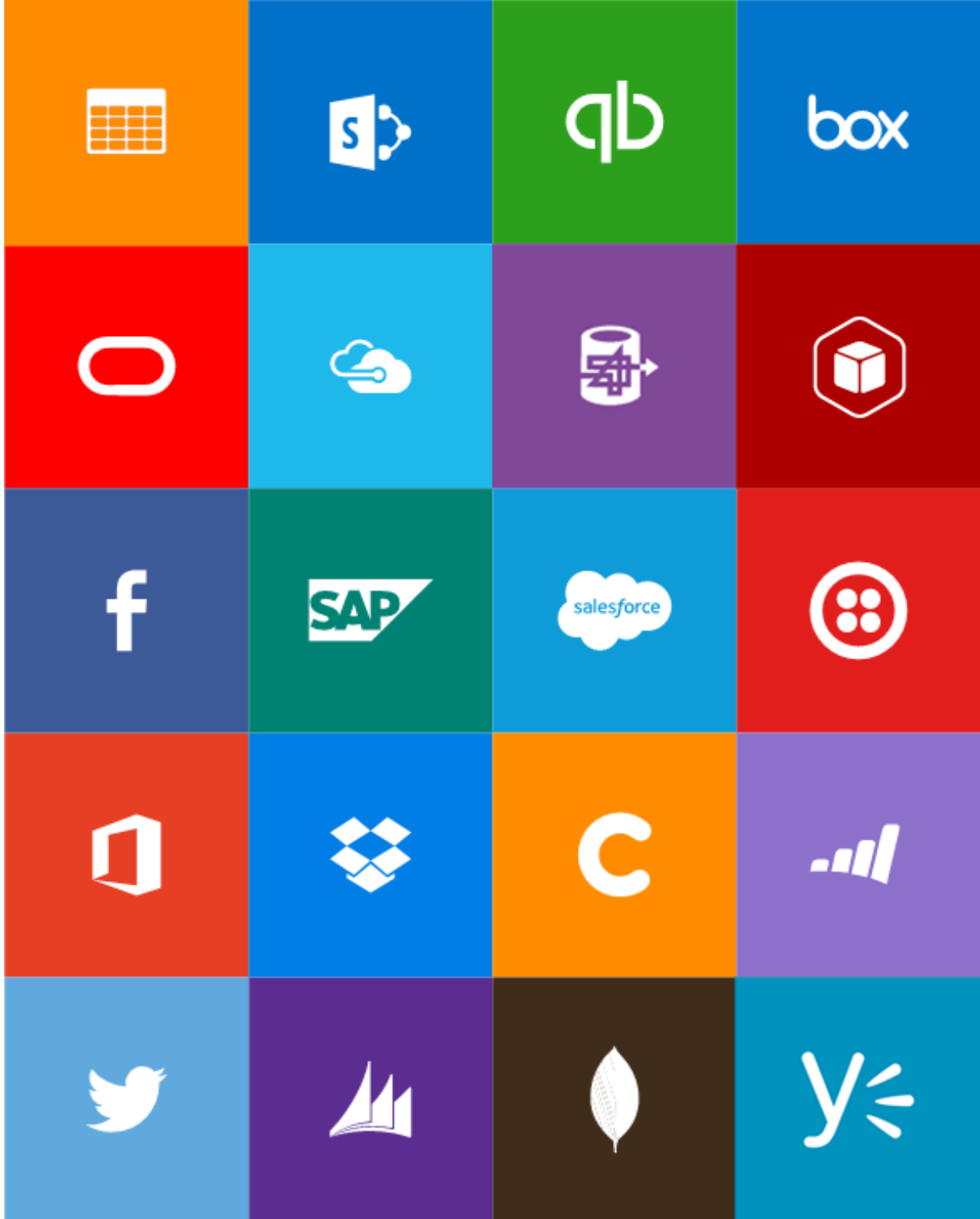
- Box
- Chatter
- Delay
- Dropbox
- Azure HD Insight
- Marketo
- Azure Media Services
- OneDrive
- SharePoint
- SQL Server
- Office 365
- Oracle
- QuickBooks
- SalesForce
- Sugar CRM
- SAP
- Azure Service Bus
- Azure Storage
- Timer / Recurrence
- Twilio
- Twitter
- IBM DB2
- Informix
- Websphere MQ
- Azure Web Jobs
- Yammer
- Dynamics CRM
- Dynamics AX
- Hybrid Connectivity

Protocols

- HTTP, HTTPS
- File
- Flat File
- FTP, SFTP
- POP3/IMAP
- SMTP
- SOAP + WCF

BizTalk Services

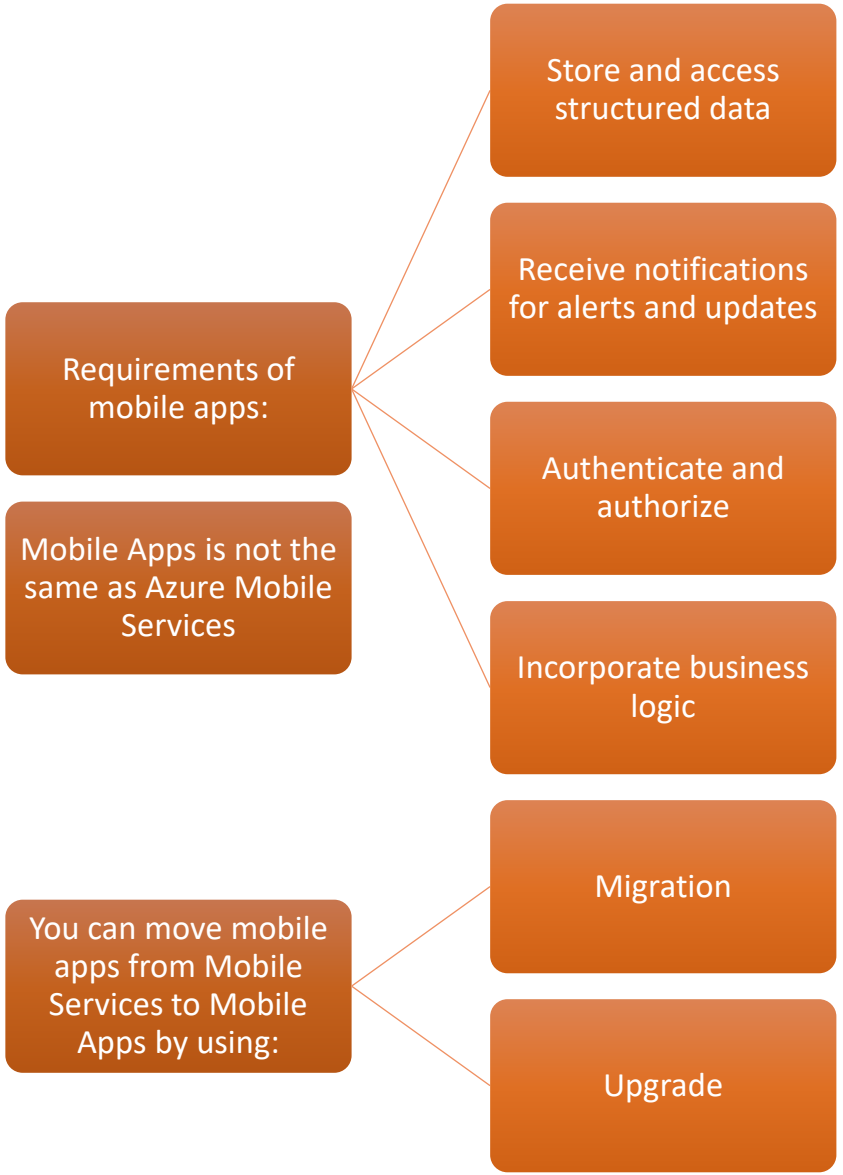
- Batching / Debatching
- Validate
- Extract (XPath)
- Transform (+Mapper)
- Convert (XML-JSON)
- Convert (XML-FF)
- X12
- EDIFACT
- AS2
- TPMOM
- Rules Engine



Demo: Azure Logic App

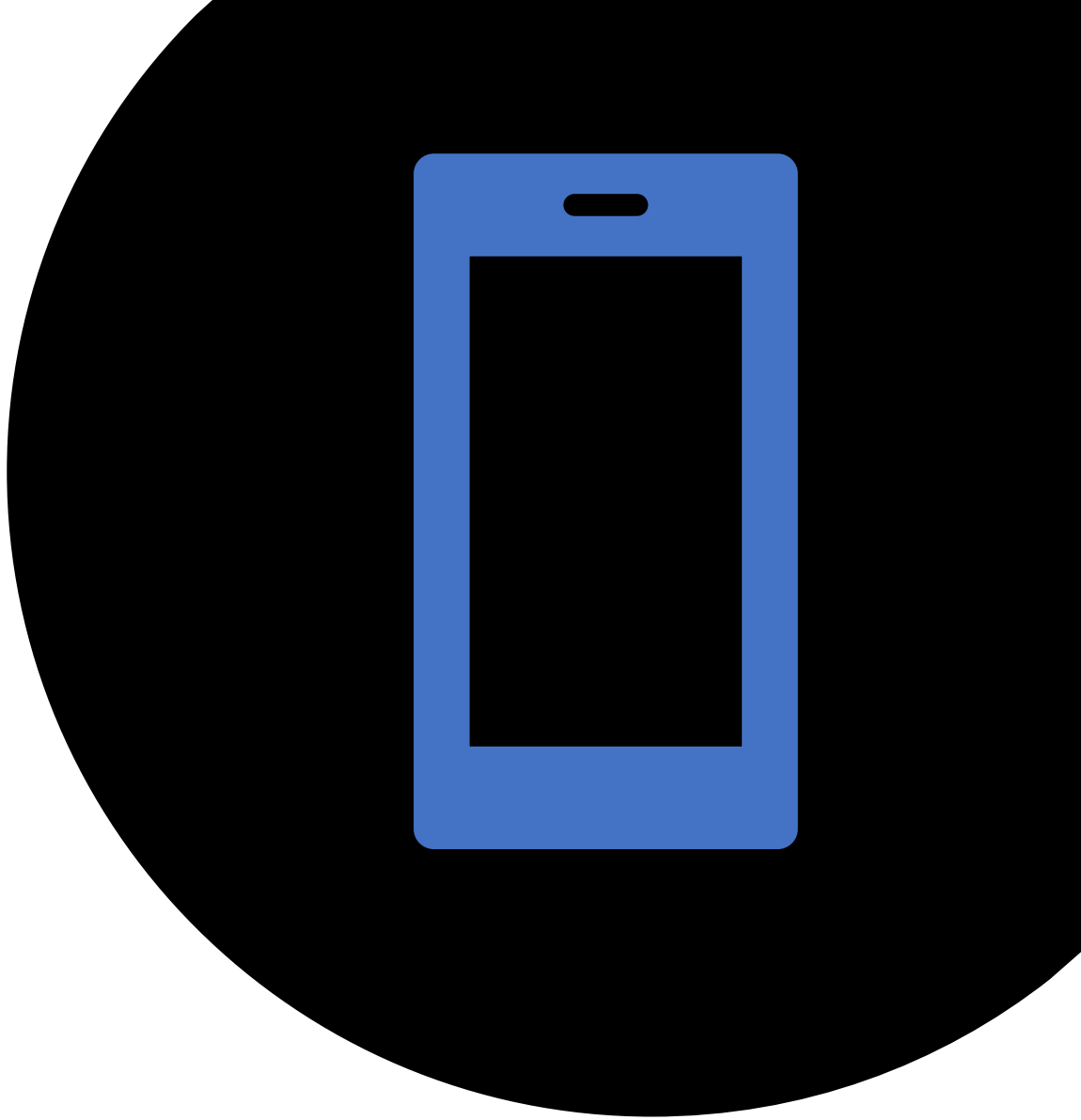
- In this demo, you will learn how to:
  - Manage and configure an Azure Logic App

Overview of Mobile Apps

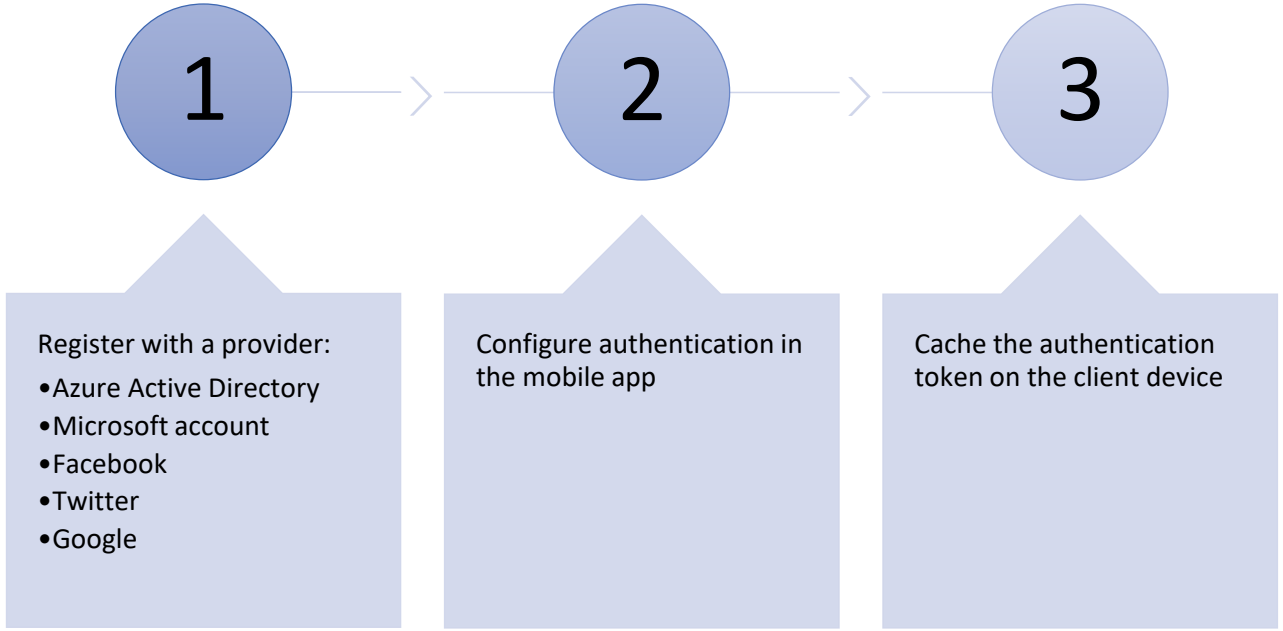


Creating and configuring mobile apps

- The features of Mobile Apps:
  - Single sign-on
  - Offline synchronization
  - Push notifications
  - Autoscaling
  - WebJobs
  - Connect to a SaaS API
  - Virtual network and hybrid integration
  - Staging environment

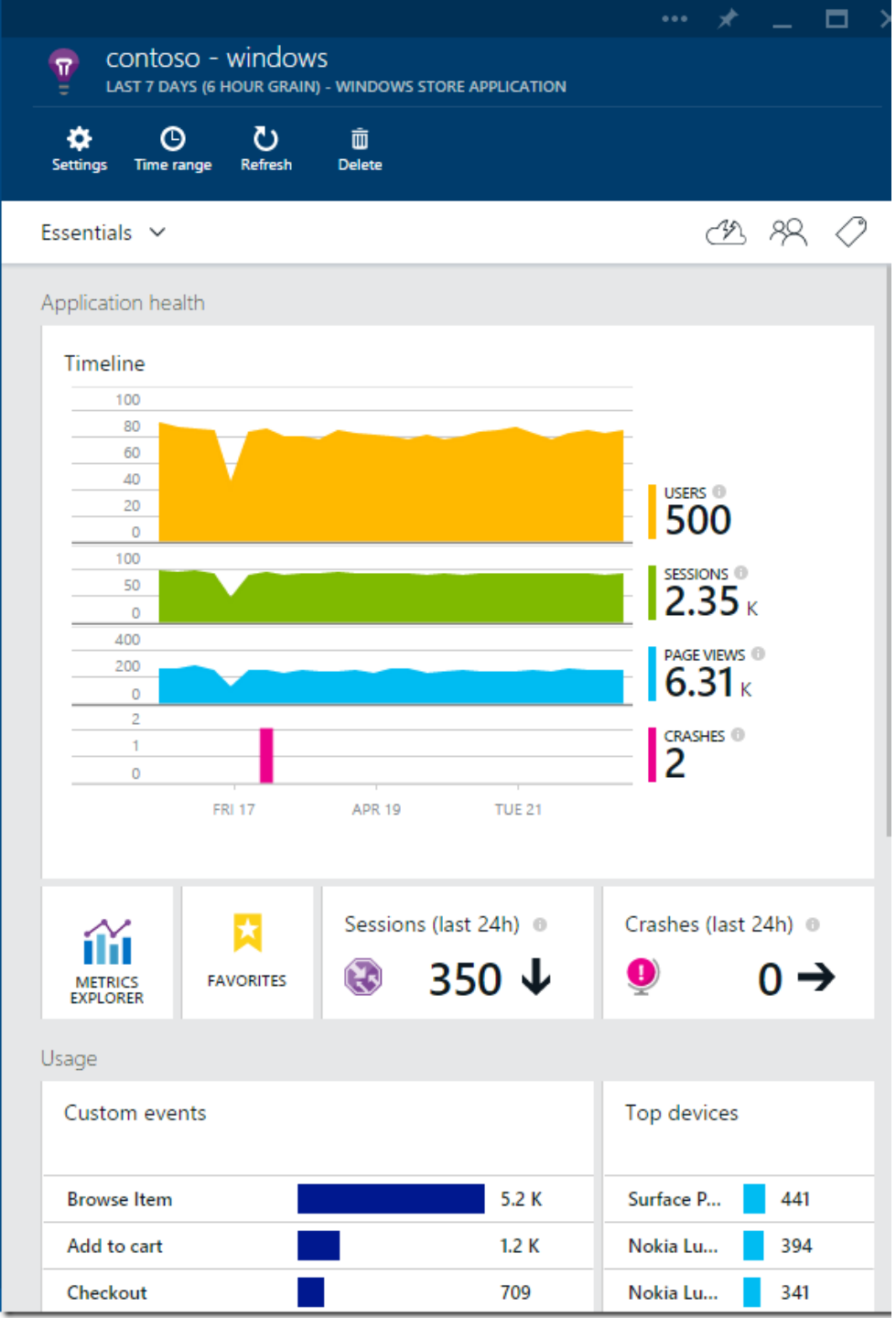


Configuring authentication



Deploying a mobile app

Using a publish profile:	Download the profile from the Azure portal
	Import the profile into Visual Studio
	Complete the publishing wizard
Using a Git repository:	Install Git and create a local repository
	Configure Local Git Depository deployment option of the Azure mobile app
	Push mobile app files to the Azure mobile app



New

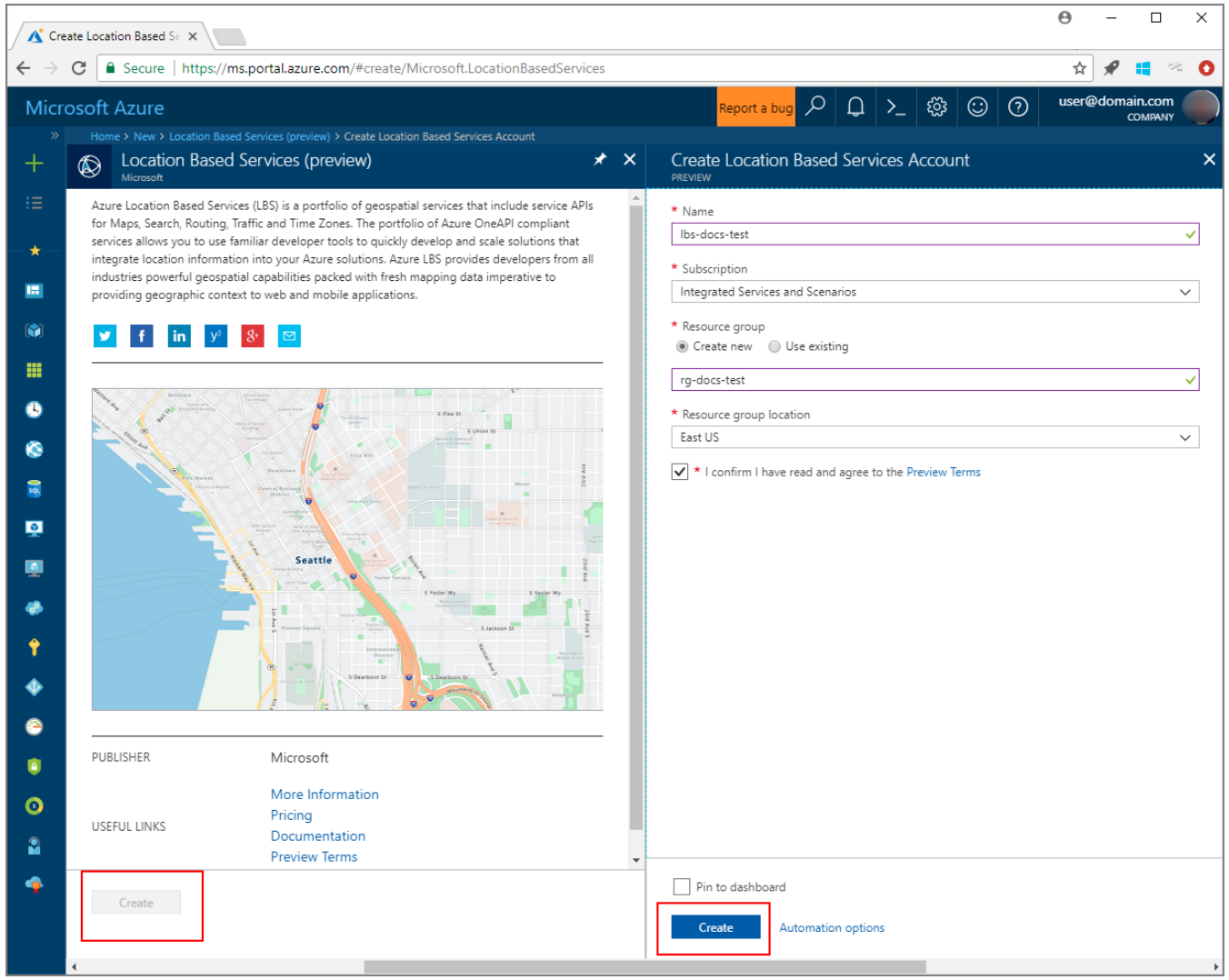
Analytics for Mobile Apps

- know in real time what your users are experiencing
- fix rapidly any problems they have
- You include in your app a small SDK, which monitors crashes and your users' actions
- you can read metrics and analyse failures. And you can set up alerts
- Application Insights provides great support for both aspects we've discussed:
  - Diagnose crashes so that you can fix the problems before they have a significant impact on your customers.
  - Analyze usage patterns and find out how your customers use your app, so you can prioritize working on the scenarios they find important.

Build location-aware applications in Azure

New

- Use Azure Location Based Services with familiar tools and services
- Easily integrate other tools into your location-aware applications, such as Power BI, SQL Database and Stream Analytics.
- JavaScript Map Control APIs make it simple to incorporate mapping capabilities into your apps
- The unique data structure of Azure Location Based Services means your maps update fast—route and traffic flow details automatically update based on what's happening in the moment—in countries all over the world.



Student Reference

https://tinyurl.com/532S04Web