



AZ-300T04 Module 01: Creating Web Applications using PaaS

Subtitle or speaker name



1

Module 01: Creating Web Applications using PaaS

Lesson 01: Introduction to Web Apps



2

Web Apps Overview

- A PaaS offering for hosting web applications:
 - **Flexibility:**
 - Support for variety of frameworks, including .NET, .NET Core, Java, PHP, Node.js, Python, and Ruby.
 - The ability to use different deployment methods, including Visual Studio, FTP, and Web Deploy.
 - Integration with a wide range of code repositories.
 - **Scalability:**
 - Vertical scaling (manual)
 - Horizontal scaling (manual and automatic)
 - Built-in load balancing
 - **Hosting model:**
 - Available on both Windows and Linux OS
 - Support for containerized workloads on Linux OS
 - Integration with CI/CD pipelines.

3

Key Features of App Service Web Apps

- Multiple languages and frameworks
- DevOps optimization
- Global scale with high availability
- Connections to SaaS platforms and on-premises data
- Security and compliance
- Application templates
- Visual Studio integration
- API and mobile features
- Serverless code

4

Azure App Service plans

In App Service, an app runs in an *App Service plan*.

- An App Service plan defines a set of compute resources for a web app to run.
- Compute resources are analogous to the server farm in conventional web hosting.
- One or more apps can be configured to run on the same computing resources (or in the same App Service plan).

When an App Service plan is created in a certain region a set of compute resources is created for that plan in that region. Whatever apps are put into the App Service plan run on those compute resources as defined by your App Service plan. Each App Service plan defines:

- Region (West US, East US, etc.)
- Number of VM instances
- Size of VM instances (Small, Medium, Large)
- Pricing tier (Free, Shared, Basic, Standard, Premium, PremiumV2, Isolated, Consumption)

5

Controlling App Service traffic by using Azure Traffic Manager

Azure Traffic Manager controls how requests from web clients are distributed to apps in Azure App Service.

When App Service endpoints are added to an Azure Traffic Manager profile, Azure Traffic Manager keeps track of the status of your App Service apps (running, stopped, or deleted)... it decides which of the endpoints should receive traffic.

Azure Traffic Manager uses four different routing methods. These methods are described in the following list:

- **Priority:** use a primary app for all traffic, and provide backups in case the primary or the backup apps are unavailable.
- **Weighted:** distribute traffic across a set of apps, either evenly or according to weights, which you define.
- **Performance:** when you have apps in different geographic locations, use the "closest" app in terms of the lowest network latency.
- **Geographic:** direct users to specific apps based on which geographic location their DNS query originates from.

6

About App Service Environments

Isolated and dedicated environments for securely running App Service at high scale

Offer hosting for:

- Windows web apps
- Linux web apps
- Docker containers
- Mobile apps
- Functions

Provide unique (to App Service) functionality, including:

- Isolated pricing tier
- Virtual network integration
- Support for layered security model and upstream security devices
- The ability to host up to 100 App Service plan instances
- Three worker tier sizes:
 - One vCPU/3.5 GB RAM
 - Two vCPU/7 GB RAM
 - Four vCPU/14 GB RAM

7

Module 01: Creating Web Applications using PaaS

Lesson 02: Using shell commands to create an App Service Web App



8

Using shell commands

- Azure Web Apps can be managed by using shell-based interface:
 - **Azure Cloud Shell:**
 - Integrates directly with the Azure portal
 - Offers the choice of Bash and PowerShell
 - Includes popular command line tools
 - **Azure CLI:**
 - Offers multi-platform support
 - Facilitates automation
 - Allows Linux/UNIX admins to leverage their scripting experience
 - **Azure PowerShell:**
 - Offers multi-platform support
 - Facilitates automation
 - Allows Windows admins to leverage their scripting experience

9

Module 01: Creating Web Applications using PaaS

Lesson 03: Creating Background Tasks



10

Overview of WebJobs

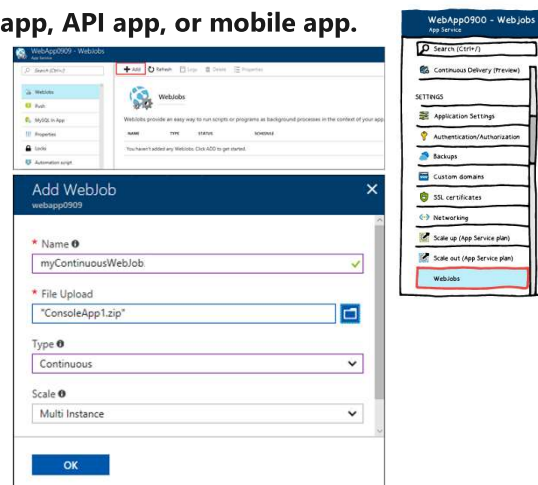
- A feature of Azure App Service for execution of background tasks:
 - Includes support for executables or scripts (.cmd, .bat, .exe, .ps1, .sh, .php, .py, .js, .jar)
 - Tasks execute in the same context as the App Service app
 - Azure WebJobs SDK provides programmability
 - Can run:
 - Continuously:
 - Start immediately when the WebJob is created and run in an endless loop
 - Run on all instances of the web app (you can restrict it to individual instances)
 - Support remote debugging
 - On trigger:
 - Start on-demand or on-schedule
 - Run on a single instance selected by the Azure platform
 - Do not support remote debugging
 - Times out after 20 minutes of inactivity, unless you enable the Always On feature;
 - Always On requires Basic, Standard, or Premium pricing tier

11

Creating a continuous Webjob

- Implementation steps (the Azure portal):
 1. Navigate to the App Service page of the web app, API app, or mobile app.
 2. On the WebJobs blade, click Add.
 3. Configure Add WebJob settings.

Setting	Sample Value	Description
Name	myContinuousWebJob	A name that is unique within an App Service app. Must start with a letter or a number and cannot contain special characters other than "." and "-".
File Upload	ConsoleApp.zip	A .zip file that contains your executable or script file as well as any supporting files needed to run the program or script. The supported executable or script file types are listed in <i>Overview of WebJobs</i> section of this course.
Type	Continuous	Choose the type of WebJob you want to create, this example uses <i>continuous</i>
Scale	Multi instance	Available only for Continuous WebJobs. Determines whether the program or script runs on all instances or just one instance. The option to run on multiple instances doesn't apply to the Free or Shared pricing tiers.

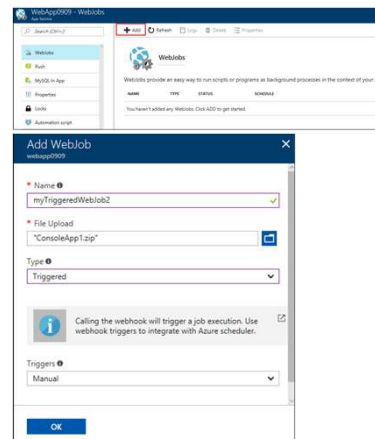


12

Creating a triggered Webjob

- Implementation steps (the Azure portal):
 - 1. Navigate to the App Service page of the web app, API app, or mobile app.
 - 2. On the WebJobs blade, click Add.
 - 3. Configure Add WebJob settings.

Setting	Sample Value	Description
Name	myContinuousWebJob	A name that is unique within an App Service app. Must start with a letter or a number and cannot contain special characters other than "-" and ".".
File Upload	ConsoleApp.zip	A .zip file that contains your executable or script file as well as any supporting files needed to run the program or script. The supported executable or script file types are listed in <i>Overview of Webjobs</i> section of this course.
Type	Triggered	Choose the type of Webjob you want to create, this example uses <i>Triggered</i> .
Triggers	Manual	



13

Module 01: Creating Web Applications using PaaS

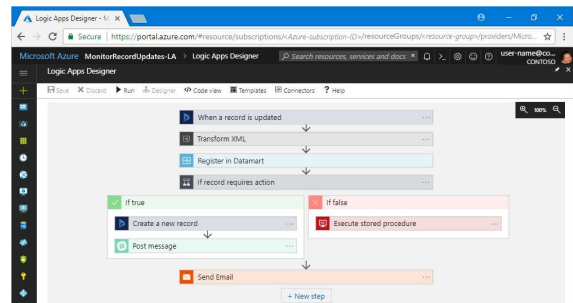
Lesson 04: Creating an App Service Logic App



14

Azure Logic Apps explained

- Trigger-based workflows integrating apps, data, systems, and services:
 - **For example:**
 - Processing and routing orders across on-premises systems and cloud services.
 - Moving uploaded files from an SFTP or FTP server to Azure Storage.
 - Sending email notifications with Office 365 in response to external events.
 - Monitor tweets for a specific subject, analyze the sentiment, and create alerts or tasks.
 - **Can be created using minimal or no-code:**
 - Include 200+ predefined connectors
 - Support integration with custom APIs (API apps)
 - **Consist of:**
 - A workflow – a sequence of steps in an acyclic graph
 - A trigger – an event that initiates workflow execution
 - Actions – workflow steps (a service, application or API)
 - Connectors – representations of APIs or data sources



15

Build the Logic App workflow

- Implementation steps:
 - 1. In Logic App Designer, select the When a feed item is published trigger
 - 2. Configure trigger properties:

Property	Value	Description
The RSS feed URL	http://feeds.reuters.com/reuters/topNews	The link for the RSS feed that you want to monitor
Interval	1	The number of intervals to wait between checks
Frequency	Minute	The unit of time for each interval between checks

- 3. Under the When a feed item is published trigger, add the Send an email action.
- 4. When prompted, sign in to your email account.
- 5. In the Send an email action, specify the data that you want the email to include.
- 6. Save the logic app.

16



17