

Get the best out of Live Sessions HOW? e!



Check your Internet Connection

Log in 10 mins before, and check your internet connection to avoid any network issues during the LIVE session.

Speak with the Instructor

By default, you will be on mute to avoid any background noise. However, if required you will be **unmuted by instructor**.



Clear Your Doubts

Feel free to clear your doubts. Use the “**Questions**” tab on your webinar tool to interact with the instructor at any point during the class.

Let us know if you liked our content

Please share feedback after each class. It will help us to enhance your learning experience.



edureka!



Microsoft Azure Developer Associate (AZ-204)

COURSE OUTLINE

MODULE 04

Introduction to Azure IaaS Compute Solutions

Implementing Azure Batch Service and
Disk Encryption

Designing and Developing Applications
That Use Containers

**Implementing Azure App Service Web Apps
and Mobile Apps**

Implementing Azure App Service API Apps and
Azure Functions

Developing Solutions That Use Azure Table
Storage and Cosmos DB

Developing Solutions That Use Relational Database
And Azure Blob Storage

Implementing Authentication and Access
Control In Azure

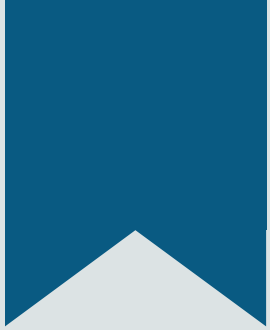
Implementing Secure Data Solutions and Integrate
Caching & CDN

Instrument Monitoring, Logging and Scalability
Of Apps & Services

Connecting to and Consuming Azure and
Third-party Services

Developing Event-based and Message-based
Solutions in Azure





Module 4 – Implementing Azure App Service Web Apps and Mobile Apps

Topics

- Azure App Services
- Azure App Service Plan
- Scaling An App Service Plan
- Azure App Service WebApps
- Azure WebApp Concepts
- Azure WebApp deployment through Portal, PowerShell and CLI
- Azure WebJobs
- Continuous And Triggered WebJobs
- Azure App Service Mobile Apps
- Registering Apps For Push Notifications
- Azure Notification Hubs

Objectives

After completing this module, you should be able to:

- Understand App Service core concepts and capabilities
- Create App Service web apps by using Azure CLI, Azure Portal, and PowerShell
- Create continuous and triggered WebJobs
- Push an app on to the Mobile App service
- Know how to register apps for push notifications





Azure App Services

Understanding Azure App Services

- Azure App Service is the only cloud service that integrates everything you need to quickly and easily build web and mobile apps for any platform and any device
- It is built for developers
- App Service is a fully managed platform with powerful capabilities like
 - Built-in DevOps
 - Continuous integration with Visual Studio Team Services and GitHub
 - Staging and production support
 - Automatic patching

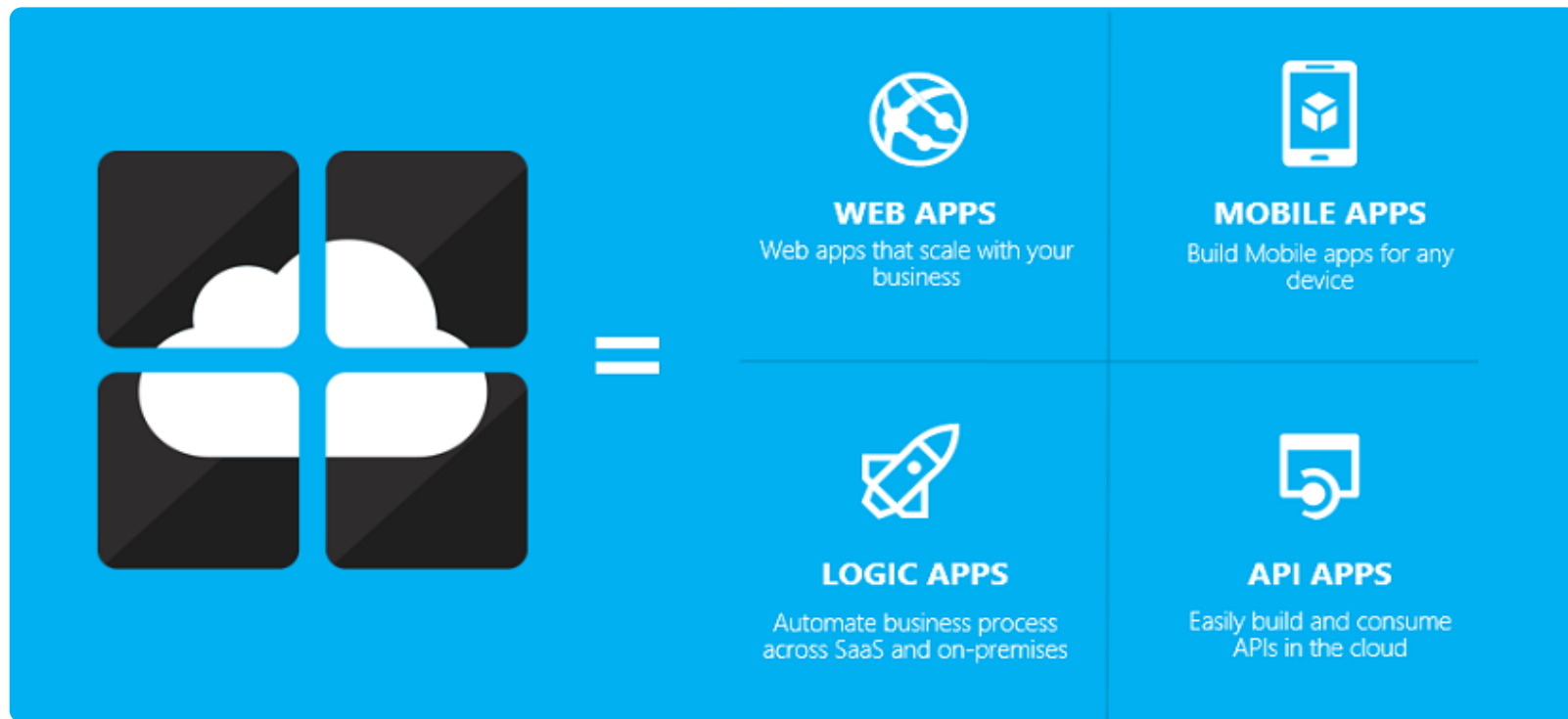


Benefits of Azure App Services

- Multiple language support - .NET, Java, Node JS, PHP, or Python
- Easily integrate your logic with any mobile or web app via standard REST APIs
- Optimized for DevOps, with continuous integration support for Visual Studio Team Services and GitHub
- Enterprise Ready, Enterprise grade security and management and PCI compliant



Azure App Service Types



Azure App Service Plan

- Before getting done in creating any kind of App, we need to Create an App Service Plan
- An App Service Plan is a set of configurations that we choose for running our application
- In creating an App Service Plan, we specify 3 things;
 1. Name of the App service plan
 2. Location of the App deployment
 3. A Pricing Tier

You can deploy more than a single app into the same App Service Plan at no additional cost



Pricing Tier

- A Pricing Tier gives a user a specific amount of resources at a specific cost
- User selects a Pricing Tier according to their App's initial requirements
- These resources can also be scaled up later






Choose your pricing tier

Browse the available plans and their features

* Linux App Service plans will be discounted at 50% for preview. [Learn more](#)

S1 Standard	S2 Standard	S3 Standard
1 Core	2 Core	4 Core
1.75 GB RAM	3.5 GB RAM	7 GB RAM
50 GB Storage	50 GB Storage	50 GB Storage
Custom domains / SSL SNI Incl & IP SSL Support	Custom domains / SSL SNI Incl & IP SSL Support	Custom domains / SSL SNI Incl & IP SSL Support
Up to 10 instances Auto scale	Up to 10 instances Auto scale	Up to 10 instances Auto scale
Daily Backup	Daily Backup	Daily Backup
5 slots Web app staging	5 slots Web app staging	5 slots Web app staging
Preview Pricing 50% off price below	Preview Pricing 50% off price below	Preview Pricing 50% off price below
44.64 USD/MONTH (ESTIMATED)	89.28 USD/MONTH (ESTIMATED)	178.56 USD/MONTH (ESTIMATED)
B1 Basic	B2 Basic	B3 Basic
1 Core	2 Core	4 Core
1.75 GB RAM	3.5 GB RAM	7 GB RAM
10 GB Storage	10 GB Storage	10 GB Storage
Custom domains	Custom domains	Custom domains
SSL Support SNI SSL Included	SSL Support SNI SSL Included	SSL Support SNI SSL Included
Up to 3 instances Manual scale	Up to 3 instances Manual scale	Up to 3 instances Manual scale
Preview Pricing 50% off price below	Preview Pricing 50% off price below	Preview Pricing 50% off price below
32.74 USD/MONTH (ESTIMATED)	65.47 USD/MONTH (ESTIMATED)	130.94 USD/MONTH (ESTIMATED)

Azure Pricing Tiers Overview

	<u>Free</u> To quickly evaluate platform and convert the app at any time to one of the paid plans without delays or downtime
	<u>Shared</u> It allows you to develop and test in an environment with features such as SSL, domain names, and more before production. It is suitable for low traffic sites
	<u>Basic</u> It is for apps that have lower traffic requirements and do not need more advanced feature like auto scale and traffic management features.
	<u>Standard</u> This plan is for production API, Mobile, and Web apps. There are no limits on the number of apps/domains you can host using the Standard service plan.
	<u>Premium</u> Premium plan supports larger numbers of scale instances , additional connectors, and BizTalk capabilities, and includes all the advanced capabilities found in the Standard plan.

Which Pricing Tier will we Choose for our Web App?



Because our Edureka Web App is simple and doesn't require much resources





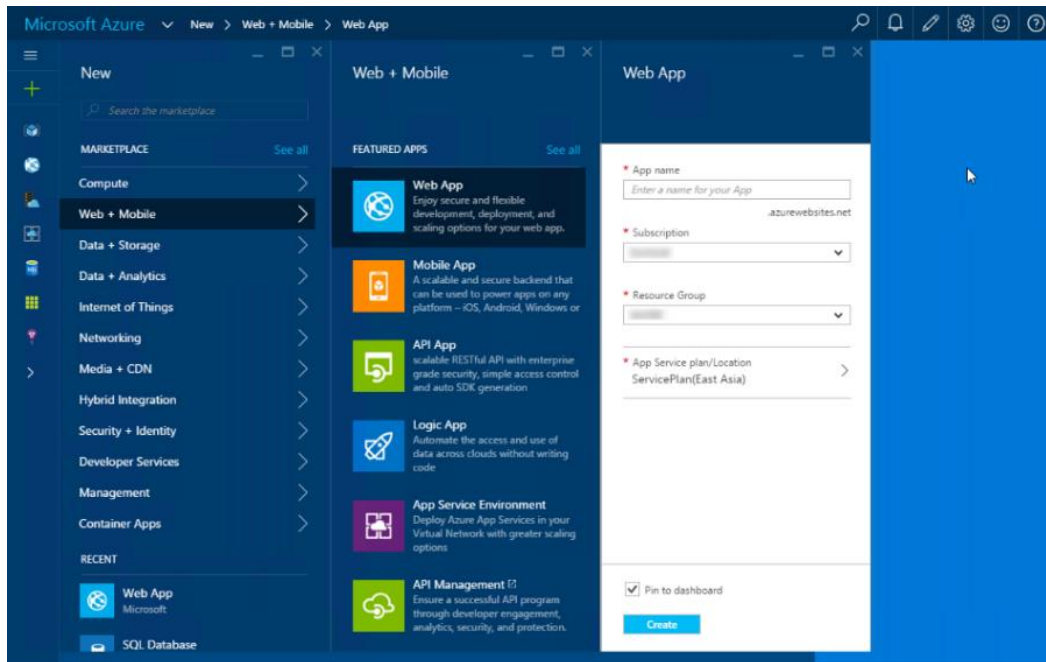
Creating an App Service Plan

Creating an App Service Plan

- You can create an empty **App Service plan**

from the **App Service plan** browse experience or as part of app creation

1. Login in to <http://portal.azure.com>
2. Click NEW, then select Web + mobile, then select Web Apps



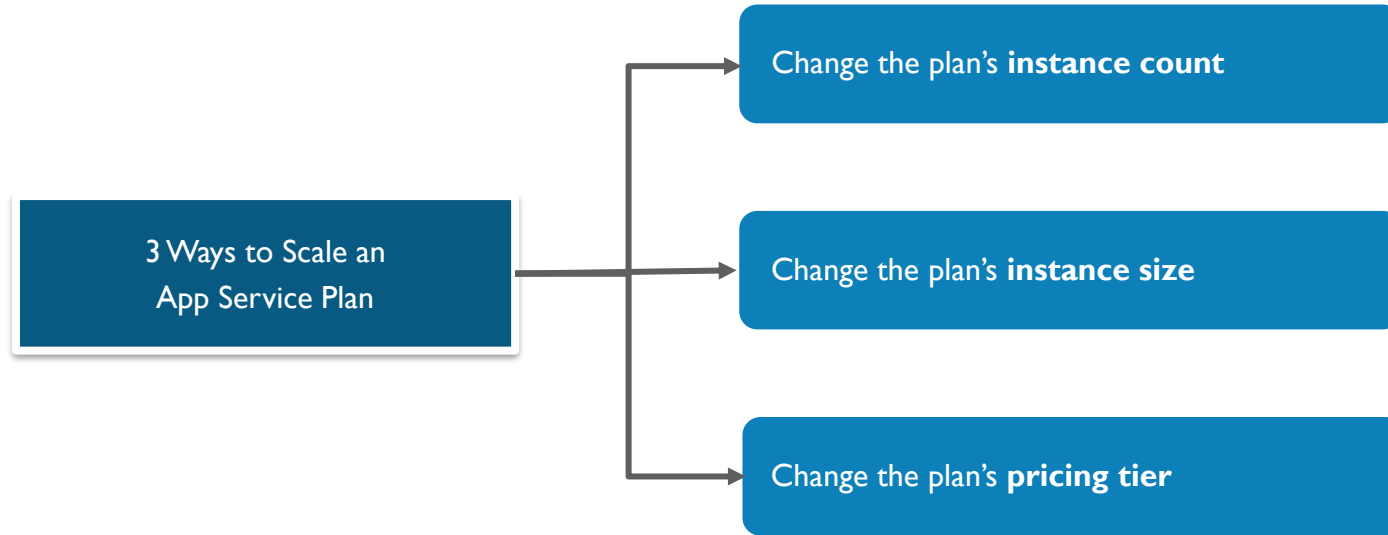
Creating an App Service Plan (Cont.)

3. Create a New App Service Plan

- Click **+ Create New**
- Type the **App Service plan** name and select an appropriate **Location**
- Click **Pricing tier** and select an appropriate pricing tier for the service.
We have used **FREE App Service Plan**
- Select **FREE App Service Plan**
- Click **Create**

The screenshot displays the Microsoft Azure portal interface for creating a new App Service plan. The breadcrumb navigation at the top indicates the path: Microsoft Azure > New > Web + Mobile > Web App > App Service plan > App Service plan. The left-hand navigation pane shows the 'Web App' section with a '+ Create New' button highlighted. The main content area is divided into three panes. The left pane shows the configuration steps: 'App name' (with a placeholder 'Enter a name for your App'), 'Subscription' (set to 'azurewebsites.net'), 'Resource Group' (set to 'sandbox'), and 'App Service plan/Location' (set to 'sandbox(West US)'). The middle pane shows a 'Create New' button and a list of existing plans: 'sandbox(S1) West US' (1 instances, 0 ap...) and 'sandbox2(S1) West US' (1 instances, 2 ap...). The right pane shows the 'App Service plan' configuration with fields for 'App Service plan' (placeholder 'Enter a name for your App Service Plan'), 'Location' (set to 'West US'), and 'Pricing tier' (set to 'S1 Standard'). A 'Create' button is visible at the bottom right.

Scaling an App Service Plan



App Service Plan Dashboard View

The screenshot displays the Azure App Service Plan dashboard. The left sidebar contains navigation icons. The main content area is divided into three sections:

- Settings:** A list of settings for the App Service Plan, including URL, App Service plan/pricing tier, FTP/Deployment username, FTP hostname, and FTPS hostname. The 'Scale Up (App Service Plan)' option is highlighted with a red box.
- Choose your pricing tier:** A section showing available pricing tiers for App Service Environments. It includes a table with columns for S1 Standard, S2 Standard, and S3 Standard, and rows for B1 Basic, B2 Basic, and B3 Basic. The 'Scale Up (App Service Plan)' option is highlighted with a red box.
- Scale Up (App Service Plan):** A section showing the details of the selected App Service Plan, including its name, location, and pricing tier.

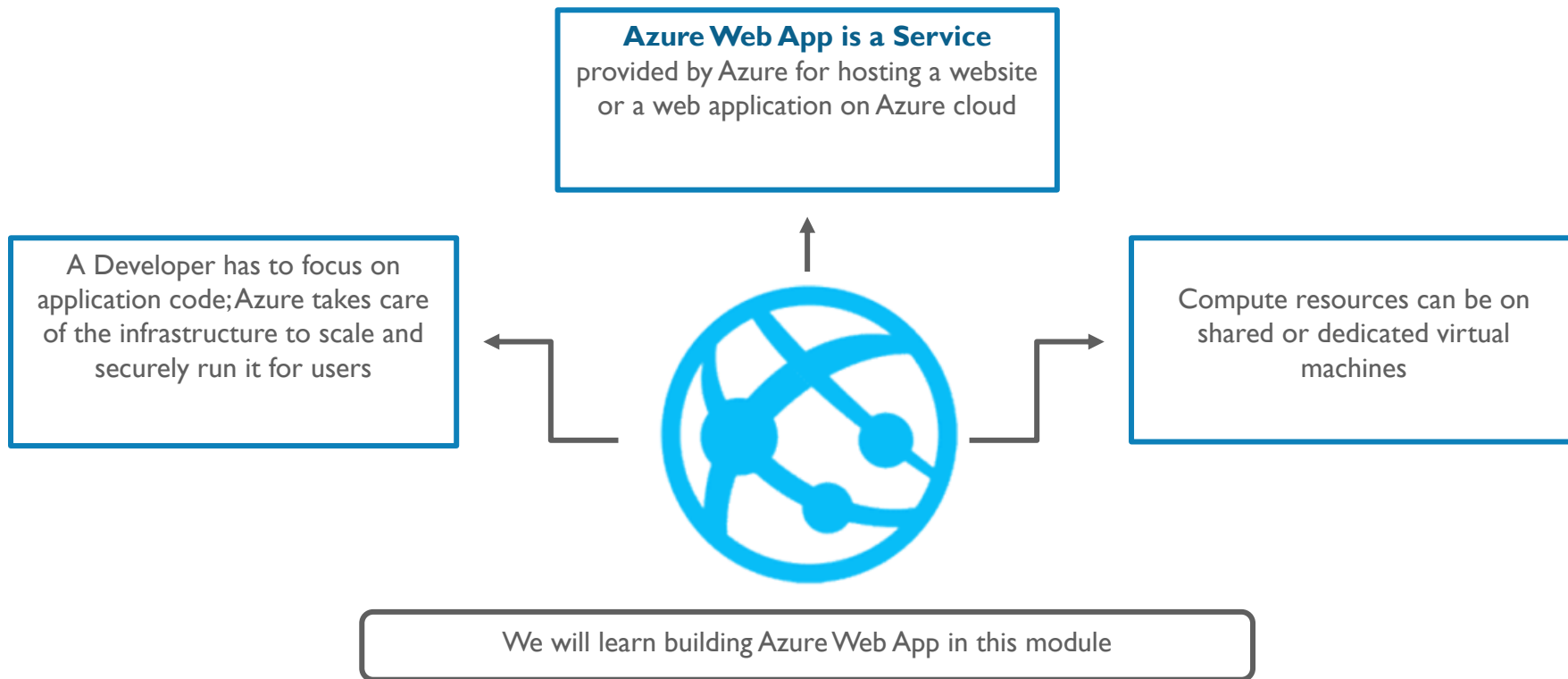
S1 Standard		S2 Standard		S3 Standard	
1	Core	2	Core	4	Core
1.75	GB RAM	3.5	GB RAM	7	GB RAM
50 GB Storage		50 GB Storage		50 GB Storage	
5 SNI, 1 IP Custom domains		5 SNI, 1 IP Custom domains / SSL		5 SNI, 1 IP Custom domains / SSL	
Up to 10 instances Auto scale		Up to 10 instances Auto scale		Up to 10 instances Auto scale	
Daily Backup		Daily Backup		Daily Backup	
5 slots Web app staging		5 slots Web app staging		5 slots Web app staging	
Traffic Manager Geo availability		Traffic Manager Geo availability		Traffic Manager Geo availability	
74.40 USD/MONTH (ESTIMATED)		148.80 USD/MONTH (ESTIMATED)		297.60 USD/MONTH (ESTIMATED)	

B1 Basic		B2 Basic		B3 Basic	
1	Core	2	Core	4	Core
1.75	GB RAM	3.5	GB RAM	7	GB RAM
10 GB Storage		10 GB Storage		10 GB Storage	
Custom domains		Custom domains		Custom domains	
Up to 3 instances Manual scale		Up to 3 instances Manual scale		Up to 3 instances Manual scale	



Azure App Service WebApp

What is an Azure WebApp?



Why Use Azure WebApp?

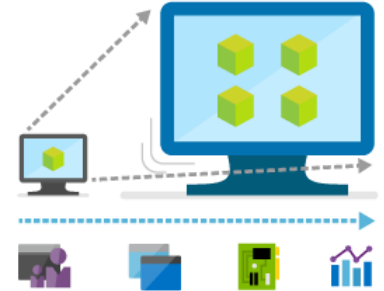


It is Flexible

- Supports many IDEs and frameworks, such as .NET, Java, PHP, Node.js
- You can use Git and Kudu to deploy Node.js or PHP web applications

It is Scalable

- Web application can be easily scaled up by configuring auto scale in the Azure portal
- Auto scale creates multiple instances a Web App that are load balanced automatically



Azure WebApp Concepts



WebApp Concepts

WebApp Gallery

- Existing WebApp templates with one-click installation of packages such as WordPress, Joomla, Drupal etc.

Auto-Scaling

- WebApps enables you to quickly scale-up or out to handle any incoming customer load

Continuous Integration

- Enables you to automatically build, test & deploy WebApp on each successful code check-in or integration tests

Deployment Slots

- Implement staged deployment to verify your code in a pre-production environment which is identical to your production WebApp in Azure App Service

Testing in Production

- Take Staged Deployments to the next level and perform A/B testing to verify your new code with a configurable fraction of your live traffic

WebJobs

- Run any program or script on Web Apps VMs. Run jobs continuously or on a schedule and scale to run on multiple VMs



Using Azure WebApp

Steps to Use Azure WebApp Service

1. Create a Web App instance by using the Portal

- It involves the type of App, creating resource, service plan etc.

2. Deploying an ASP.NET Web Application to an Azure Web App instance

- Create and Publish a web app through an IDE by configuring it with Azure Web App instance

3. Manage the Azure Web App

- Manage the Web App through the Azure portal

Creating a WebApp Instance





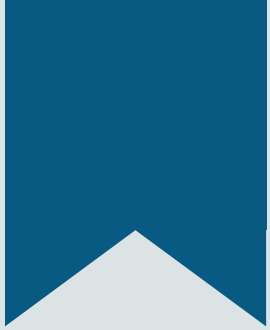
Demo 1.1 – Create App Service WebApps By Using Portal

(Refer [demo doc-1.1](#))



Demo 1.2 – Create App Service WebApps By Using Azure CLI

(Refer [demo doc-1.2](#))



Run Background Tasks With WebJobs in Azure App Service

Run Background Tasks With WebJobs

- WebJobs is a feature of Azure App Service that enables you to *run a program or script* in the same context as a web app, API app, or mobile app
- There is no additional cost to use WebJobs
- The Azure WebJobs SDK can be used with WebJobs to simplify many programming tasks



Types of WebJobs

- The following table describes the differences between *Continuous* and *Triggered* WebJobs:

Continuous	Triggered
<ul style="list-style-type: none">• Starts immediately when the WebJob is created• To keep the job from ending, the program or script typically does its work inside an endless loop• If the job does end, you can restart it	<ul style="list-style-type: none">• Starts only when triggered manually or on a schedule.
<ul style="list-style-type: none">• Runs on all instances that the web app runs on• You can optionally restrict the WebJob to a single instance.	<ul style="list-style-type: none">• Runs on a single instance that Azure selects for load balancing.
<ul style="list-style-type: none">• Supports remote debugging	<ul style="list-style-type: none">• Doesn't support remote debugging.

Types of WebJobs

NOTE: A web app can time out after 20 minutes of inactivity. Only requests to the actual web app reset the timer.

Viewing the app's configuration in the Azure portal or making requests to the advanced tools site (https://<app_name>.scm.azurewebsites.net) don't reset the timer. If your app runs continuous or scheduled WebJobs, enable **Always On** to ensure that the WebJobs run reliably. This feature is available only in the Basic, Standard, and Premium [pricing tiers](#).

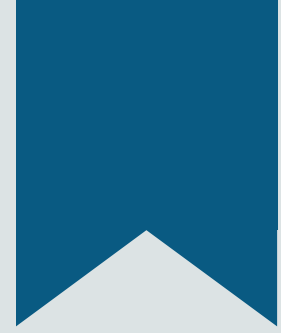
Supported File Types for Scripts or Programs

- The following file types are supported:
 - .cmd, .bat, .exe (using Windows cmd)
 - .ps1 (using PowerShell)
 - .sh (using Bash)
 - .php (using PHP)
 - .py (using Python)
 - .js (using Node.js)
 - .jar (using Java)



Demo 2 – Create Continuous and Triggered WebJobs

(Refer demo doc-2)



Azure App Service Mobile Apps

Mobile Apps in Azure App Service

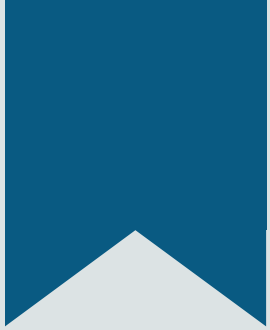
- Highly scalable
- Globally available, mobile application development platform for:
 - Enterprise Developers and
 - System Integrators,that brings a rich set of capabilities to mobile developers



With Mobile Apps You Can

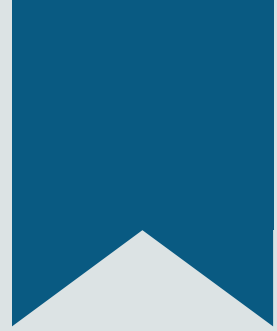
- Build native and cross platform apps
- Connect to your enterprise systems
- Build offline-ready apps with data sync
- Push Notifications to millions in seconds





Demo 3 – Push an App onto the Mobile App Service

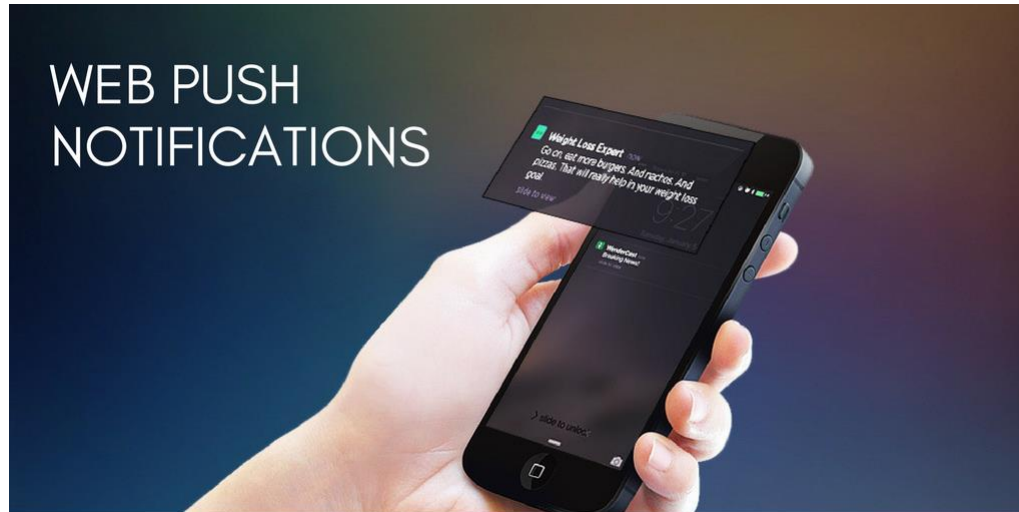
(Refer demo doc-3)



Registering Apps for Push Notifications

What are Push Notifications?

- Push notifications is a form of *app-to-user* communication where users of mobile apps are **notified** of certain **desired information**, usually in a **pop-up or dialog box** on a mobile device
- Users generally choose to **view or dismiss** the message; choosing the former **opens** the **mobile application** that communicated the notification



Working of Push Notifications

Push notifications are delivered through platform-specific infrastructures called **Platform Notification Systems (PNSes)**

They offer *barebone* push functionalities to deliver a message to a **device** with a provided **handle**, and have **no** common interface

To send a notification to all customers across the **Android, iOS, and Windows** versions of an app, the developer must work *Separately* with:

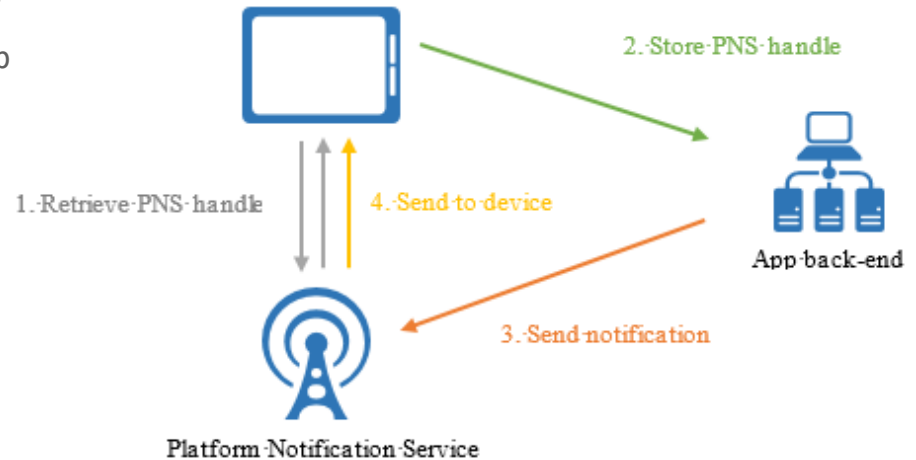
Apple Push Notification Service(APNS)

Windows Notification Service(WNS)

Firebase Cloud Messaging(FCM)

How Push Works at a High Level?

1. An app decides it wants to receive notification, so it **contacts** PNS for the **target platform** where the app is running and requests a **unique and temporary push handle**
 - The handle **type** depends on the system (WNS uses **URIs** while APNS uses **tokens**)
2. The client app **stores** this handle in the app backend or provider
3. To send a push notification, the **app backend** **contacts** the PNS using the handle to target a specific client app
4. The PNS **forwards** the notification to the **device** specified by the handle



Challenges of Push Notifications

Some of the infrastructural challenges are:

1

Backend has a hard time handling the **Platform dependency**

Scalability and **broadcasting** issue when there is a huge increase in devices

2

3

Maintenance of a registry for **Routing** purpose causes overhead of cost and time

What are Azure Notification Hubs?



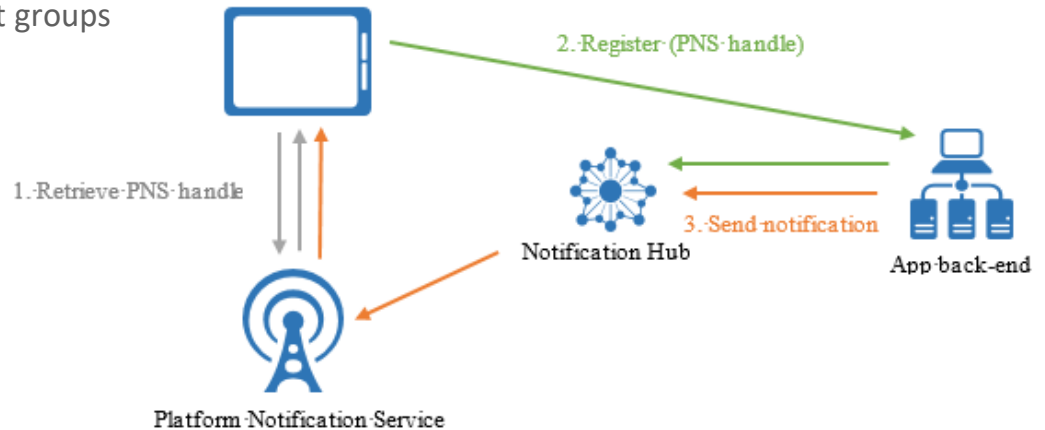
Azure Notification Hubs provide an *easy-to-use* and *scaled-out push engine* that allows you to send notifications to **any platform** (iOS, Android, Windows, Kindle, Baidu, etc.) from **any backend** (cloud or on-premises).



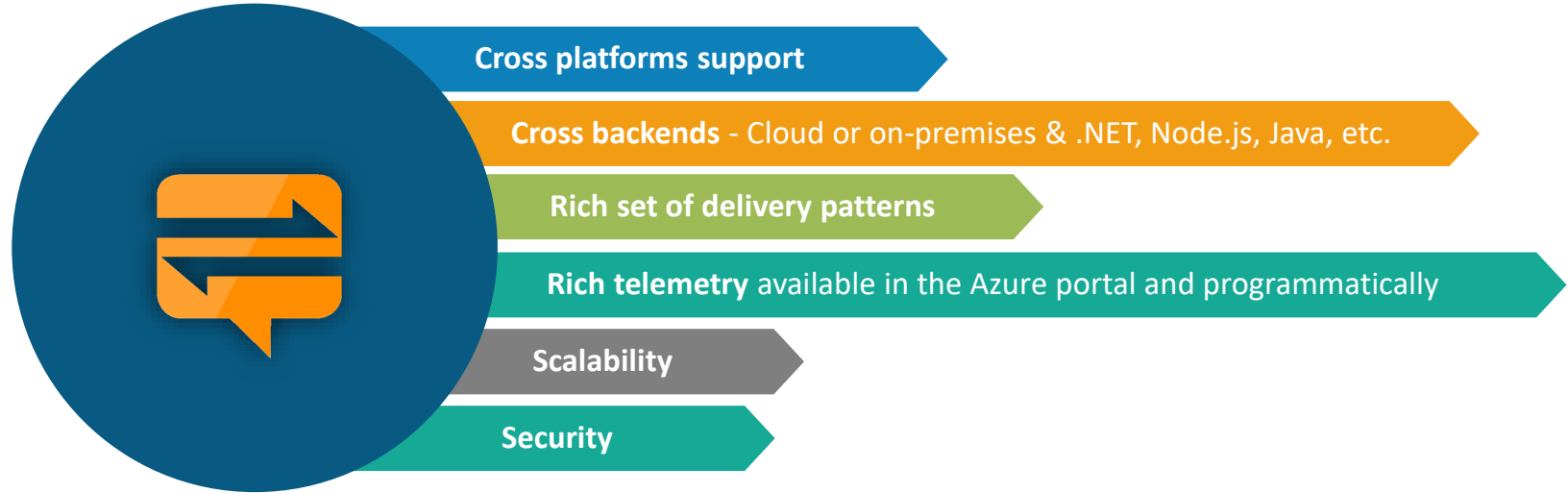
Notification Hubs works great for both *enterprise* and *consumer* scenarios.

Why Use Azure Notification Hubs?

- Notification Hubs **eliminates** all complexities associated with pushing notifications *on your own* from your **app backend**
- Its **multi-platform, scaled-out** push notification infrastructure **reduces** push-related coding and **simplifies** your backend
- With Notification Hubs, devices are merely responsible for registering their PNS handles with a hub, while the backend sends messages to users or interest groups



Notification Hubs – Advantages



Notification Hubs – Usage Scenarios

Send breaking news notifications to millions with low latency

Send location-based coupons to interested user segments

Send event-related notifications to users/groups for media/sports/finance/gaming apps

Push promotional contents to apps to engage and market to customers

Notify users of enterprise events like new messages and work items

Send codes for multi-factor authentication



Demo 4 – Registering Apps for Push Notifications

(Refer demo doc-4)

Summary

Application Environment Components



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

Container V/S VM

Parameters	VM	Container
Work Scope	If you want to run multiple applications, use VM	If you want to run multiple copies of a single app, use Containers
Portability	Working with VMs is platform independent	Working with containers is platform dependent
Resources	VMs take up a lot of system resources	When resources are a constraint, use Containers
Security	You need not have sudo privileges to import subsystems	You need to have sudo privileges to import subsystems

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

What Is Docker?

"BUILD, SHIP & RUN ANY SOFTWARE ANY WHERE"

- Docker is a tool designed to create, deploy, and run applications with ease by using containers
- It allows a developer packaging of an application with all of the requirements such as libraries and other dependencies, ship it all as one package
- It ensure that your application works seamlessly in any environment be it Development, Test or Production



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

Azure Container Registry (ACR)

Azure Container Registry is a managed docker registry to store and manage your private Docker container images

ACR is available in three SKUs, namely Basic, Standard and Premium

All the images in your registry are encrypted at rest

Geo-Redundant Storage feature in ACR automatically replicates your images to multiple data centre to prevent the loss of storage failure in a region

Geo-Replication is a feature in premium registry to prevent the loss of registry in the event of total region failure and just the storage



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

Azure Container Instances (ACI)

Azure Container Instance provides a platform where we can run isolated Docker containers, including simple applications, build jobs and task automation

Benefits of ACI

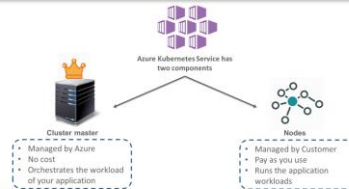


edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

Azure Kubernetes Service (AKS)

AKS is a service which gives a managed Kubernetes cluster in Azure to reduce the complexity of managing and operating the containers by provisioning, upgrading and scaling resources on demand without any downtime



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

Questions



FEEDBACK





Thank You

For more information please visit our website
www.edureka.co