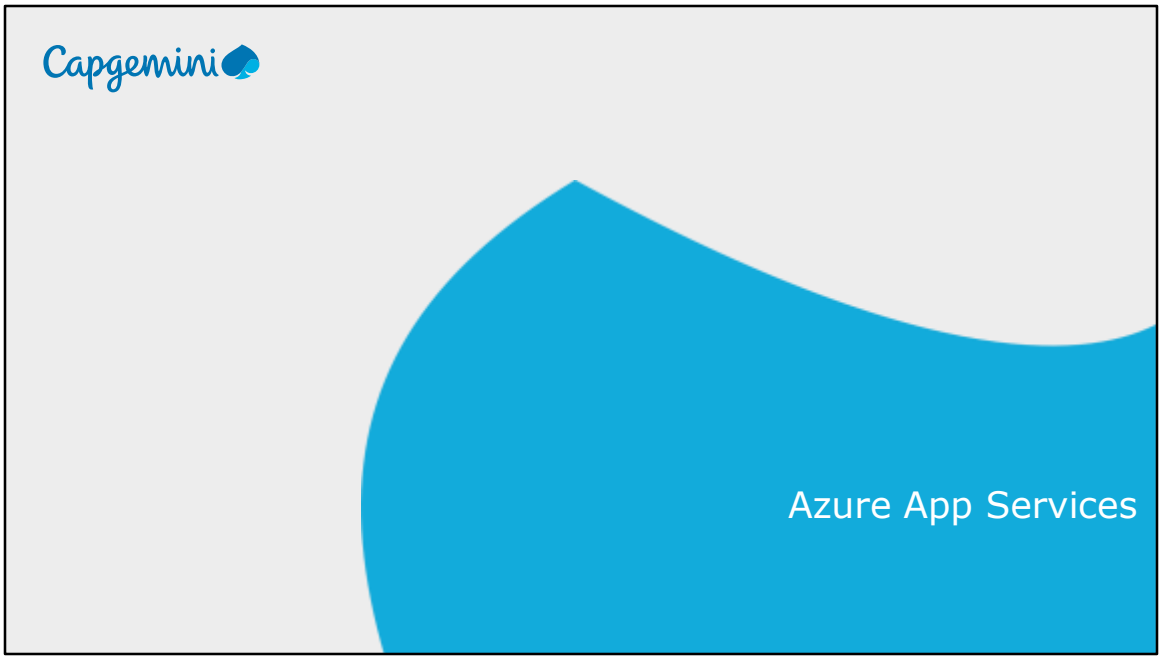

Capgemini
Azure App Services

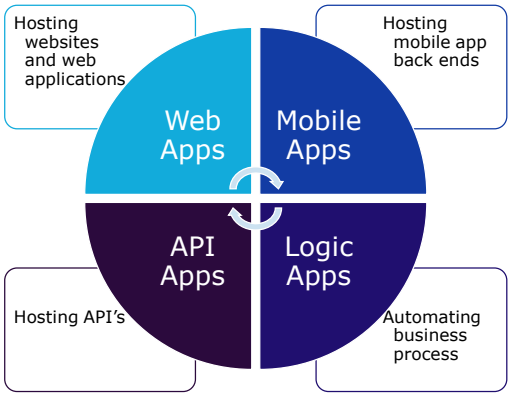
Lesson Objectives

At the end of this module you will be able to:

- ✓ Understand the importance of App Service plan
- ✓ Create and Publish Web application and API in Azure
- ✓ Understand how Azure CDN works
- ✓ Create Deployment Slots
- ✓ Understand the need for Enabling CORS

Azure App Services

- App Service is a Platform as a Service that lets you create web, mobile and many other apps and run these apps on fully managed virtual machines with your choice of shared VM resources or dedicated VMs



Azure App Service Web Apps (or just Web Apps) is a service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, be it .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. Applications run and scale with ease on Windows-based environments

Web Apps not only adds the power of Microsoft Azure to your application, such as security, load balancing, autoscaling, and automated management. You can also take advantage of its DevOps capabilities, such as continuous deployment from VSTS, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and SSL certificates.

Besides Web Apps in App Service, Azure offers other services that can be used for hosting websites and web applications. For most scenarios, Web Apps is the best choice.

For microservice architecture, consider Service Fabric.

If you need more control over the VMs that your code runs on, consider Azure Virtual Machines.

App Service Environments

- Azure App Service Environment is an Azure App Service feature that provides a fully isolated and dedicated environment for securely running App Service apps at high scale.
- This capability can host your:
  - Windows web apps
  - Linux web apps (in Preview)
  - Docker containers (in Preview)
  - Mobile apps
  - Functions

App Service environments (ASEs) are appropriate for application workloads that require:

- Very high scale.
- Isolation and secure network access.
- High memory utilization.

Customers can create multiple ASEs within a single Azure region or across multiple Azure regions.

ASEs are isolated to running only a single customer's applications and are always deployed into a virtual network. Customers have fine-grained control over inbound and outbound application network traffic. Applications can establish high-speed secure connections over VPNs to on-premises corporate resources.

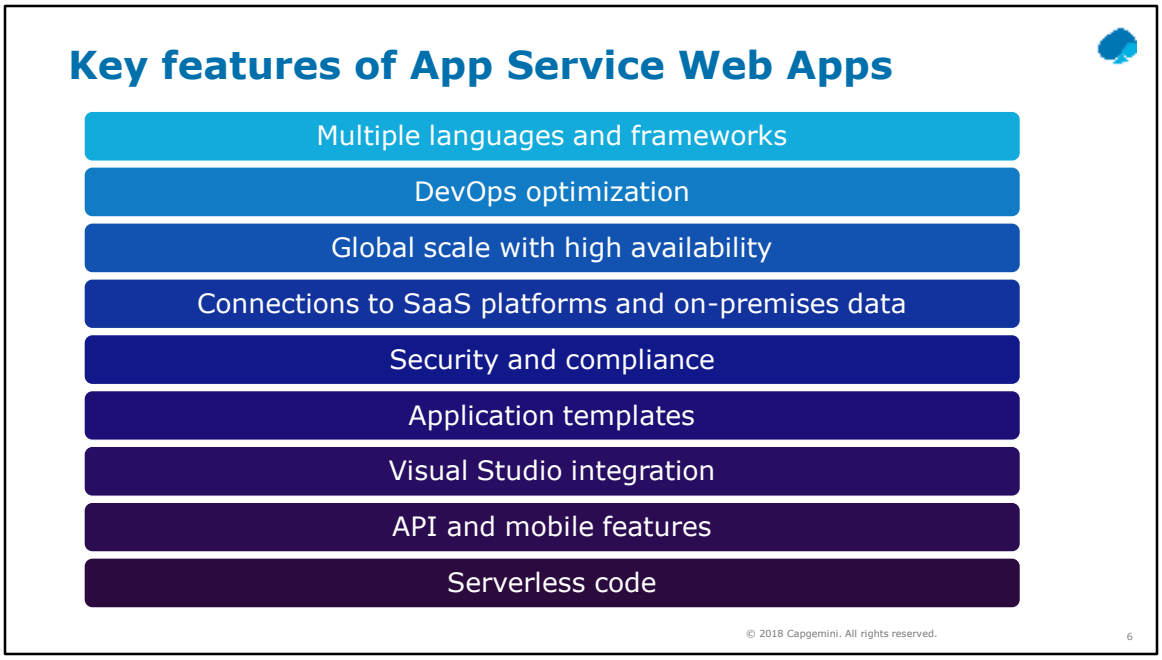
App Service plan

- An App Service plan defines a set of compute resources for a web app to run.
  - These 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)
- The pricing tier of an App Service plan determines what App Service features you get and how much you pay for the plan.
  - Shared compute(Free and Shared), Dedicated compute(Basic, Standard, Premium, and PremiumV2), Isolated and Consumption are few categories of pricing tiers.

When you create an App Service plan in a certain region (for example, West Europe), a set of compute resources is created for that plan in that region. Whatever apps you put into this App Service plan run on these 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)

Shared compute: Free and Shared, the two base tiers, runs an app on the same Azure VM as other App Service apps, including apps of other customers. These tiers allocate CPU quotas to each app that runs on the shared resources, and the resources cannot scale out.
Dedicated compute: The Basic, Standard, Premium, and PremiumV2 tiers run apps on dedicated Azure VMs. Only apps in the same App Service plan share the same compute resources. The higher the tier, the more VM instances are available to you for scale-out.
Isolated: This tier runs dedicated Azure VMs on dedicated Azure Virtual Networks, which provides network isolation on top of compute isolation to your apps. It provides the maximum scale-out capabilities.
Consumption: This tier is only available to function apps. It scales the functions dynamically depending on workload.



Multiple languages and frameworks: Web Apps has first-class support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python.

DevOps optimization: Set up 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 your apps anywhere in Microsoft's global datacenter infrastructure, and the App Service SLA promises high availability.

Connections to SaaS platforms and on-premises data: Choose from more than 50 connectors for enterprise systems (such as SAP), SaaS services (such as Salesforce), and internet services (such as Facebook). Access on-premises data using Hybrid Connections and Azure Virtual Networks.

Security and compliance: App Service is ISO, SOC, and PCI compliant. Authenticate users with Azure Active Directory or with social login. Create IP address restrictions and manage service identities.

Application templates: Choose from an extensive list of application templates in the Azure Marketplace, such as WordPress, Joomla, and Drupal.

Visual Studio integration: Dedicated tools in Visual Studio streamline the work of creating, deploying, and debugging.

API and mobile features: Web Apps provides turn-key CORS support for RESTful API scenarios, and simplifies mobile app scenarios by enabling authentication, offline data sync, push notifications, and more.

Serverless code: Run a code snippet or script on-demand without having to explicitly provision or manage infrastructure, and pay only for the compute time your code actually uses

Deployment Slots

- Deployment slots allows you to swap between staged and production web app with minimal downtime
- Requires Standard tier or higher to provide deployment slots (staging environments)

Azure API Apps

- API apps features makes it easy to develop, host, and consume APIs in the cloud and on-premises.
- With API Apps, you can have enterprise grade security, simple access control, automatic SDK generation and seamless integration with Logic Apps
- Following are the key features of the app
  - Bring your own existing API
  - Easy Consumption
  - Simple access control
  - Visual studio integration
  - Integration with Logic apps

Bring your own existing API
No need to change the code, Just deploy the same to an API app. API can be developed in any language framework supported by App Service such as C#, Java, PHP, Node.js etc.

Easy Consumption
Integrated support for Swagger API metadata makes your APIs easily consumable by a variety of clients. Automatically generate client code for your APIs in a variety of languages including C#, Java, and Javascript.

Simple access control
Protect an API app from unauthenticated access with no changes to your code. Built-in authentication services secure APIs for access by other services or by clients representing users

API Apps provides a platform in which you can host your app, an API in this case, with ease. It is designed so that you can bring your own API and reap the benefits of hosting it in a platform as a service environment

API Apps can provide authentication and authorization for your API out-of-the-box

API Definition

- An API Definition is basically metadata about your API that enables client applications to know which applications they can call, and how to do so.
- An API Definition enables applications like Visual Studio to generate client code that can call the API
- API Apps supports creating and exposing API definitions with Swagger 2.0
- Using the Swagger meta data, we can use Visual Studio or the command line to generate client code that can call our API easily.
  - API Apps enable you to use CORS to do cross-domain JavaScript calls to your API.

API Apps make it easy to expose an API definition in the form of metadata about your API. Consumption of your API easy by Visual Studio, which allow you to generate client code to call your API. You do need to have the Azure SDK installed to do this

Azure API Apps supports creating and exposing API Definition Data through the use of Swagger 2.0. Swagger is a popular open source metadata format that works with a variety of technologies, including .NET. In .NET API Apps we use Swashbuckle to generate the Swagger. This is also an open source library that helps to generate the Swagger of the API. It also provides a UI, which you can use to navigate the Swagger metadata, and even test the API. Client applications can use tools like Visual Studio to generate client-side SDK code that can interact with the API.

CORS

- By default, calls come from a domain other than the domain of the API are blocked for security reasons. CORS(Cross Origin Resource Sharing), can help manage exceptions to this behavior.
- CORS represents methods that can be used to share resources, like API calls across domains

When a web browser running in one domain makes an HTTP request for a resource in another domain, it's called a cross-origin HTTP request. If the request is made in a script language such as JavaScript, the browser will not allow the request.

For example, if a web application running on contoso.com makes a request for a jpeg on fabrikam.blob.core.windows.net, it will be blocked.

What if you actually want to share the images in your storage account with Contoso? Azure Storage allows you to enable CORS—Cross-Origin Resource Sharing. For this example, you would enable CORS on the fabrikam storage account and allow access from contoso.com.

Azure CDN

- A content delivery network (CDN) is a distributed network of servers that can efficiently deliver web content to users.
  - CDNs store cached content on edge servers in point-of-presence (POP) locations that are close to end users, to minimize latency
- Azure Content Delivery Network (CDN) offers developers a global solution for rapidly delivering high-bandwidth content to users by caching their content at strategically placed physical nodes across the world.

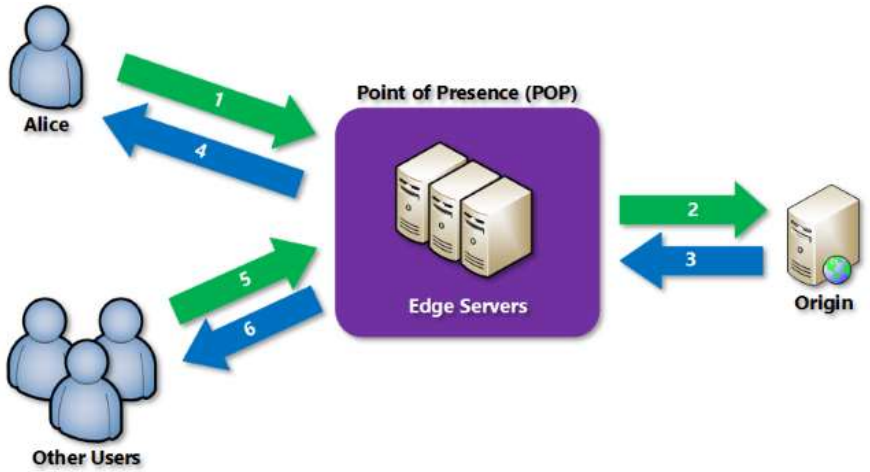
The Azure Content distribution Network (CDN) caches static web content at strategically placed locations to provide maximum throughput for delivering content to users.

The benefits of using Azure CDN to deliver web site assets include:

- Better performance and improved user experience for end users, especially when using applications in which multiple round-trips are required to load content.
- Large scaling to better handle instantaneous high loads, such as the start of a product launch event.
- Distribution of user requests and serving of content directly from edge servers so

that less traffic is sent to the origin server.

How Azure CDN Works



1. A user (Alice) requests a file (also called an asset) by using a URL with a special domain name, such as <endpoint name>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

2. If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

3. The origin server returns the file to an edge server in the POP.

4. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time-to-live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

5. Additional users can then request the same file by using the same URL that Alice used, and can also be directed to the same POP.

6. If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

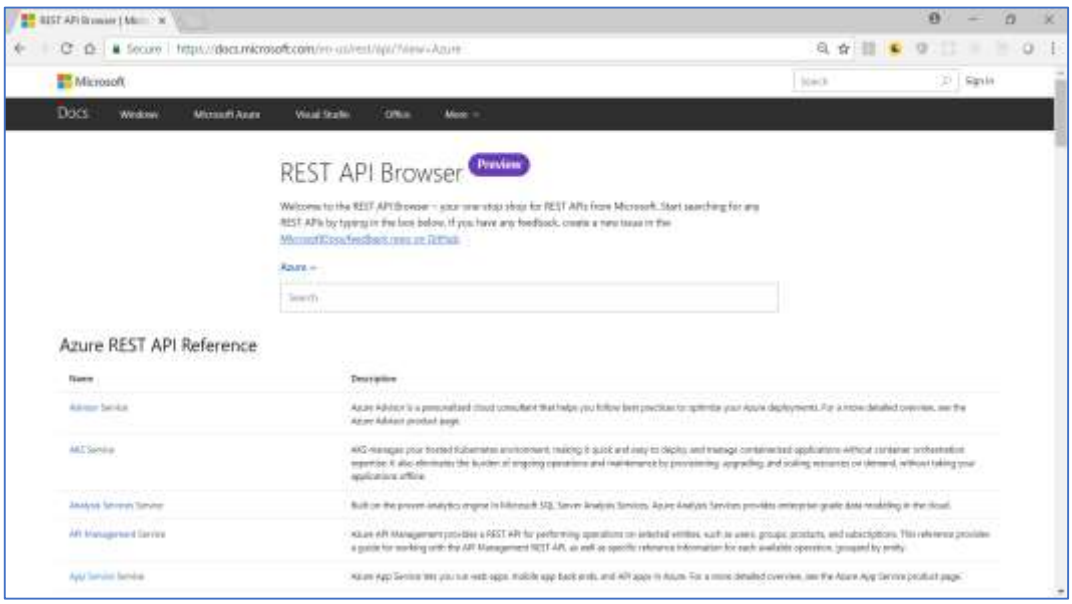
Azure REST API

- Representational State Transfer (REST) APIs are service endpoints that support sets of HTTP operations (methods), which provide create, retrieve, update, or delete access to the service's resources.
- Azure REST APIs allow us to interact with nearly every type of resource in Azure programmatically.
- The Azure REST API Reference includes a list of all possible operations categorized by resource

API Reference : https://docs.microsoft.com/en-us/rest/api/?view=Azure

Most Azure services require your client code to authenticate with valid credentials before you can call the service's API. Authentication is coordinated between the various actors by Azure AD, and provides your client with an access token as proof of the authentication. The token is then sent to the Azure service in the HTTP Authorization header of subsequent REST API requests. The token's claims also provide information to the service, allowing it to validate the client and perform any required authorization.

REST API Browser



Summary

- Azure App Service Web Apps (or just Web Apps) is a service for hosting web applications, REST APIs, and mobile back ends.
- App Service plan defines a set of compute resources for a web app to run. One or more apps can be configured to run on the same App Service plan
- Web Apps has first-class support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python

Summary

- API Apps can provide authentication and authorization for your API out-of-the-box
- Deployment slots allows you to swap between staged and production web app with minimal downtime
- Azure Content distribution Network (CDN) caches static web content at strategically placed locations to provide maximum throughput for delivering content to users.

People matter, results count.

About Capgemini

With more than 190,000 people, Capgemini is present in over 40 countries and celebrates its 50th Anniversary year in 2017. A global leader in consulting, technology and outsourcing services, the Group reported 2016 global revenues of EUR 12.5 billion. Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at

www.capgemini.com