Introduction to Web App Service overview

- *Azure App Service* is an HTTP-based 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 both Windows and Linux-based environments.
- App Service 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 Azure DevOps, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and TLS/SSL certificates.
- Link : https://docs.microsoft.com/en-us/azure/app-service/overview

Usage of Azure App Service

- Azure App Service is a fully managed platform as a service (PaaS) offering for developers. Here are some key features of App Service:

- **Multiple languages and frameworks** - App Service has first-class support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. You can also run PowerShell and other scripts or executables as background services.
- **Managed production environment** - App Service automatically patches and maintains the OS and language frameworks for you. Spend time writing great apps and let Azure worry about the platform.
- **Containerization and Docker** - Dockerize your app and host a custom Windows or Linux container in App Service. Run multi-container apps with Docker Compose. Migrate your Docker skills directly to App Service.
- **DevOps optimization** - Set up continuous integration and deployment with Azure DevOps, GitHub, BitBucket, Docker Hub, or Azure Container Registry. Promote updates through test and staging environments. Manage your apps in App Service by using Azure PowerShell or the cross-platform command-line interface (CLI).
- **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.

Usage of Azure App Service – Continued..

- **Security and compliance** - App Service is ISO, SOC, and PCI compliant. Authenticate users with Azure Active Directory, Google, Facebook, Twitter, or Microsoft account. 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 and Visual Studio Code integration** - Dedicated tools in Visual Studio and Visual Studio Code streamline the work of creating, deploying, and debugging.
- **API and mobile features** - App Service 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 (see Azure Functions).

Azure Web App Service on Linux

- App Service can also host web apps natively on Linux for supported application stacks. It can also run custom Linux containers (also known as Web App for Containers).

Built-in languages and frameworks:

- App Service on Linux supports a number of language specific built-in images. Just deploy your code. Supported languages include: Node.js, Java (JRE 8 & JRE 11), PHP, Python, .NET Core, and Ruby. Run az webapp list-runtimes --os linux to view the latest languages and supported versions. If the runtime your application requires is not supported in the built-in images, you can deploy it with a custom container.

- Outdated runtimes are periodically removed from the Web Apps Create and Configuration blades in the Portal. These runtimes are hidden from the Portal when they are deprecated by the maintaining organization or found to have significant vulnerabilities. These options are hidden to guide customers to the latest runtimes where they will be the most successful.

- When an outdated runtime is hidden from the Portal, any of

Plan and manage costs for Azure App Service

- Link : https://docs.microsoft.com/en-us/azure/app-service/overview-manage-costs

- This article describes how you plan for and manage costs for Azure App Service. First, you use the Azure pricing calculator to help plan for App Service costs before you add any resources for the service to estimate costs. Next, as you add Azure resources, review the estimated costs.
- After you've started using App Service resources, use Cost Management features to set budgets and monitor costs. You can also review forecasted costs and identify spending trends to identify areas where you might want to act. Costs for Azure App Service are only a portion of the monthly costs in your Azure bill. Although this article explains how to plan for and manage costs for App Service, you're billed for all Azure services and resources used in your Azure subscription, including the third-party services.

Demo/Create a web app in the Azure portal

- Link : https://docs.microsoft.com/en-us/learn/modules/host-a-web-app-with-azure-app-service/2-create-a-web-app-in-the-azure-portal

- **Why use the Azure portal?**
- The first step in hosting your web application is to create a web app (an App Service app) inside your Azure subscription.
- There are several ways you can create a web app. You can use the Azure portal, the Azure Command Line Interface (CLI), a script, or an IDE.
- The information presented in this unit will discuss how to use the Azure portal to create a web app, and in the next exercise you will use this information to create a web app. For this module, we will demonstrate using the Azure portal because it's a graphical experience, which makes it a great learning tool. The portal helps you discover available features, add additional resources, and customize existing resources.

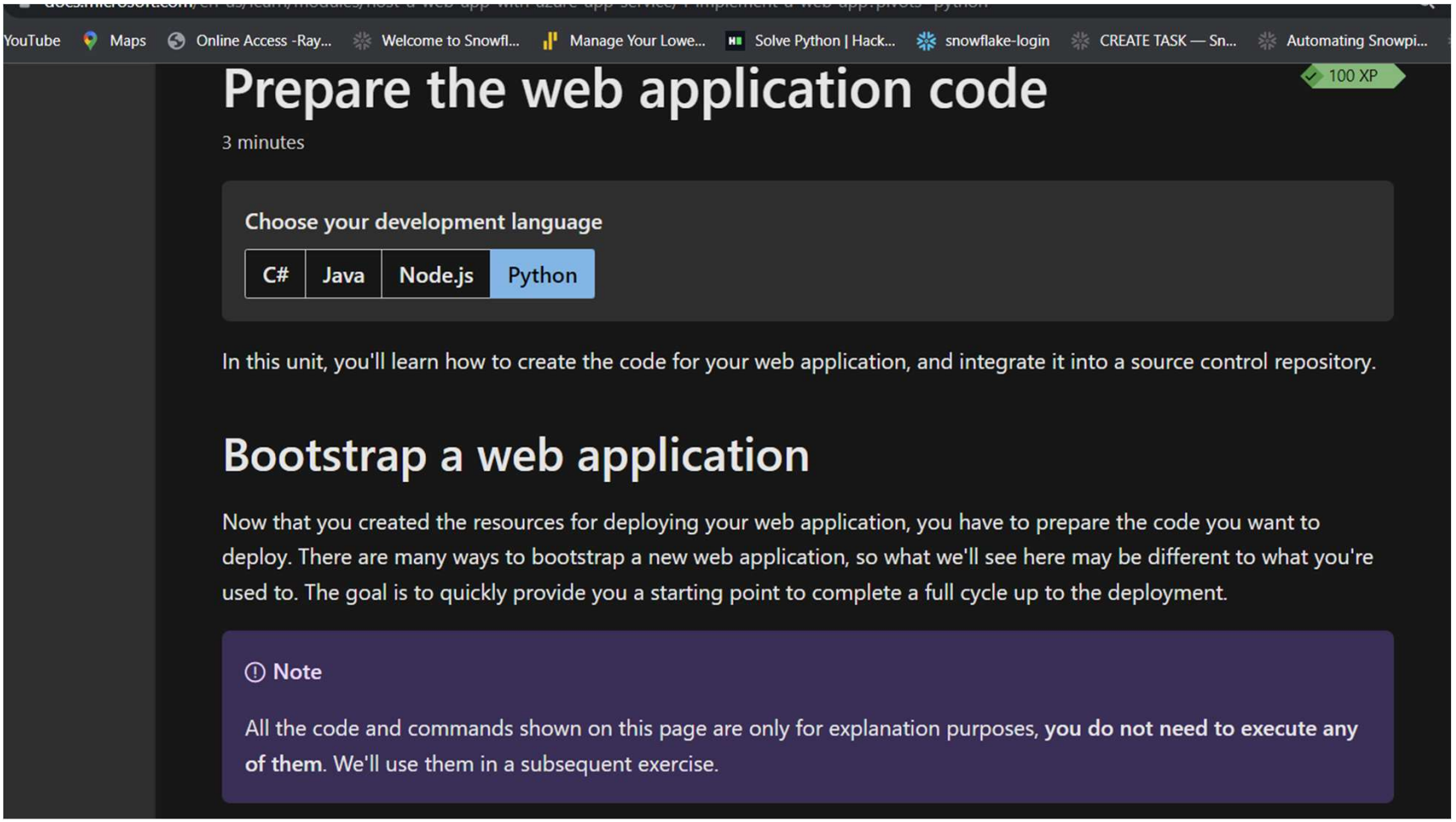
- **What is Azure App Service?**

➢ Azure App Service is a fully managed web application hosting platform. This platform as a service (PaaS) offered by Azure allows you to focus on designing and building your app while Azure takes care of the infrastructure to run and scale your applications.

- **Deployment slots**

➢ Using the Azure portal, you can easily add **deployment slots** to an App Service web app. For instance, you can create a **staging** deployment slot where you can push your code to test on Azure. Once you are happy with your code, you can easily **swap** the staging deployment slot with the production slot. You do all this with a few simple mouse clicks in the Azure portal.

- **Prepare the web application code**
- Link : https://docs.microsoft.com/en-us/learn/modules/host-a-web-app-with-azure-app-service/3-exercise-create-a-web-app-in-the-azure-portal?pivots=python
- **Quickstart: Deploy a Python (Django or Flask) web app to Azure App Service**
- Link : https://docs.microsoft.com/en-us/azure/app-service/quickstart-python?tabs=django%2Cmac-linux%2Cazure-portal%2Cvscode-deploy%2Cterminal-bash%2Cdeploy-instructions-azportal%2Cdeploy-instructions-zip-azcli


YouTube
Maps
Online Access -Ray...
Welcome to Snowfl...
Manage Your Lowe...
Solve Python | Hack...
snowflake-login
CREATE TASK — Sn...
Automating Snowpi...
Prepare the web application code
100 XP
3 minutes
Choose your development language
C#
Java
Node.js
Python
In this unit, you'll learn how to create the code for your web application, and integrate it into a source control repository.
Bootstrap a web application
Now that you created the resources for deploying your web application, you have to prepare the code you want to deploy. There are many ways to bootstrap a new web application, so what we'll see here may be different to what you're used to. The goal is to quickly provide you a starting point to complete a full cycle up to the deployment.
Note
All the code and commands shown on this page are only for explanation purposes, you do not need to execute any of them. We'll use them in a subsequent exercise.

App Service networking features

- Link : https://docs.microsoft.com/en-us/azure/app-service/networking-features
- You can deploy applications in Azure App Service in multiple ways. By default, apps hosted in App Service are accessible directly through the internet and can reach only internet-hosted endpoints. But for many applications, you need to control the inbound and outbound network traffic. There are several features in App Service to help you meet those needs. The challenge is knowing which feature to use to solve a given problem. This article will help you determine which feature to use, based on some example use cases.
- There are two main deployment types for Azure App Service:
- The multi-tenant public service hosts App Service plans in the Free, Shared, Basic, Standard, Premium, PremiumV2, and PremiumV3 pricing SKUs.
- The single-tenant App Service Environment (ASE) hosts Isolated SKU App Service plans directly in your Azure virtual network.

Web App Service pricing

- Link : https://azure.microsoft.com/en-us/pricing/details/app-service/linux/

Basic Dedicated environment for dev/test	Standard Run production workloads	Premium Enhanced performance and scale	Isolated High-Performance, Security and Isolation
Unlimited	Unlimited	Unlimited	Unlimited
10 GB	50 GB	250 GB	1 TB
Up to 3	Up to 10	Up to 30*	Up to 100
Supported	Supported	Supported	Supported
–	Supported	Supported	Supported
Supported	Supported	Supported	Supported
–	Supported	Supported	Supported
–	–	Supported	Supported
Dedicated	Dedicated	Dedicated	Isolated
$0.018/hour	$0.095/hour	$0.111/hour	$0.38/hour