

Azure Development for Python Professionals

Course Description

Azure Development for Python Professionals is an intensive 40-hour, 5-day course designed to equip Python developers with the skills and knowledge needed to effectively utilize Microsoft Azure for cloud-based development. The course covers a wide range of topics, including setting up development environments, deploying web apps, serverless computing, integrating with Azure storage solutions, working with databases, containerization, authentication, security, API management, event-driven solutions, message-based solutions, and monitoring and troubleshooting applications.

Participants will engage in hands-on labs and practical exercises to reinforce their learning and gain real-world experience in deploying and managing Python applications on Azure.

Duration: 40 hours (5-Day)

◆ Day 1:

◆ Basics + Setup

- Developer tools setup:
 - Python environment (venv, pip)
 - Azure CLI, Azure SDK for Python (`azure` packages)
 - VS Code + Azure extensions

◆ Azure App Service web apps

- Explore Azure App Service
- Configure web app settings
- Scale apps in Azure App Service
- Explore Azure App Service deployment slots
- Hands-on:
 - Create resource group, App Service Plan, App Service
 - Deploying a basic **Flask** app on **Azure App Service**

◆ Serverless with Azure Functions

- Azure Functions overview (consumption vs premium plan)
 - Creating Python-based Functions
 - Trigger types: HTTP, Timer, Blob
 - Integrating with Azure services
 - Hand-on:
 - Implement task processing logic by using Azure Functions
-

◆ Day 2:

◆ Azure Storage Integration

- Azure Storage overview: Blobs, Queues, Tables
- Explore Azure Blob storage
- Manage the Azure Blob storage lifecycle
- Work with Azure Blob storage
- Working with **Azure Storage SDK for Python**
- Hands-on:
 - Retrieve Azure Storage resources and metadata by using the Azure Storage SDK for Python

◆ **Azure Cosmos DB with Python**

- Cosmos DB overview
- Working with **Azure Cosmos DB** (NoSQL API + SDK for Python)
- Hands-on:
 - Construct a polyglot data solution

◆ **Day 3:**

◆ **Implement containerized solutions**

- Manage container images in Azure Container Registry
- Run container images in Azure Container Instances
- Implement Azure Container Apps
- Hand-on:
 - Deploy compute workloads by using images and containers
- ◆ **Implement user authentication and authorization**
 - Explore the Microsoft identity platform
 - Authenticating Python apps via MSAL (Microsoft Authentication Library)
 - Explore Microsoft Graph
 - Hand-on:
 - Authenticate by using OpenID Connect, MSAL, and python SDKs

◆ **Day 4**

◆ **Implement secure cloud solutions**

- Azure Key Vault: store/retrieve secrets via Python
- Implement managed identities
- Hands-on:
 - Access resource secrets more securely across services

◆ **Implement API Management**

- Explore API Management
- Hands-on:
 - Working with API Management Service

◆ Develop event based solution

- Explore Azure Event Grid
 - Explore Azure Event Hubs
 - Hand-on:
 - Publish and subscribe to Event Grid events
-

◆ Day 5:

◆ Develop message based solutions

- Discover Azure message queues
- Hand-on:
 - Asynchronously process messages by using Azure Service Bus Queues

◆ Troubleshoot solutions by using Application Insights

- Azure Monitor + App Insights for Python apps
- Hands-on:
 - Monitor services that are deployed to Azure

◆ Implement caching for solutions

- Develop for Azure Cache for Redis
- Develop for storage on CDNs
- Hands-on:
 - Enhance a web application by using the Azure Content Delivery Network