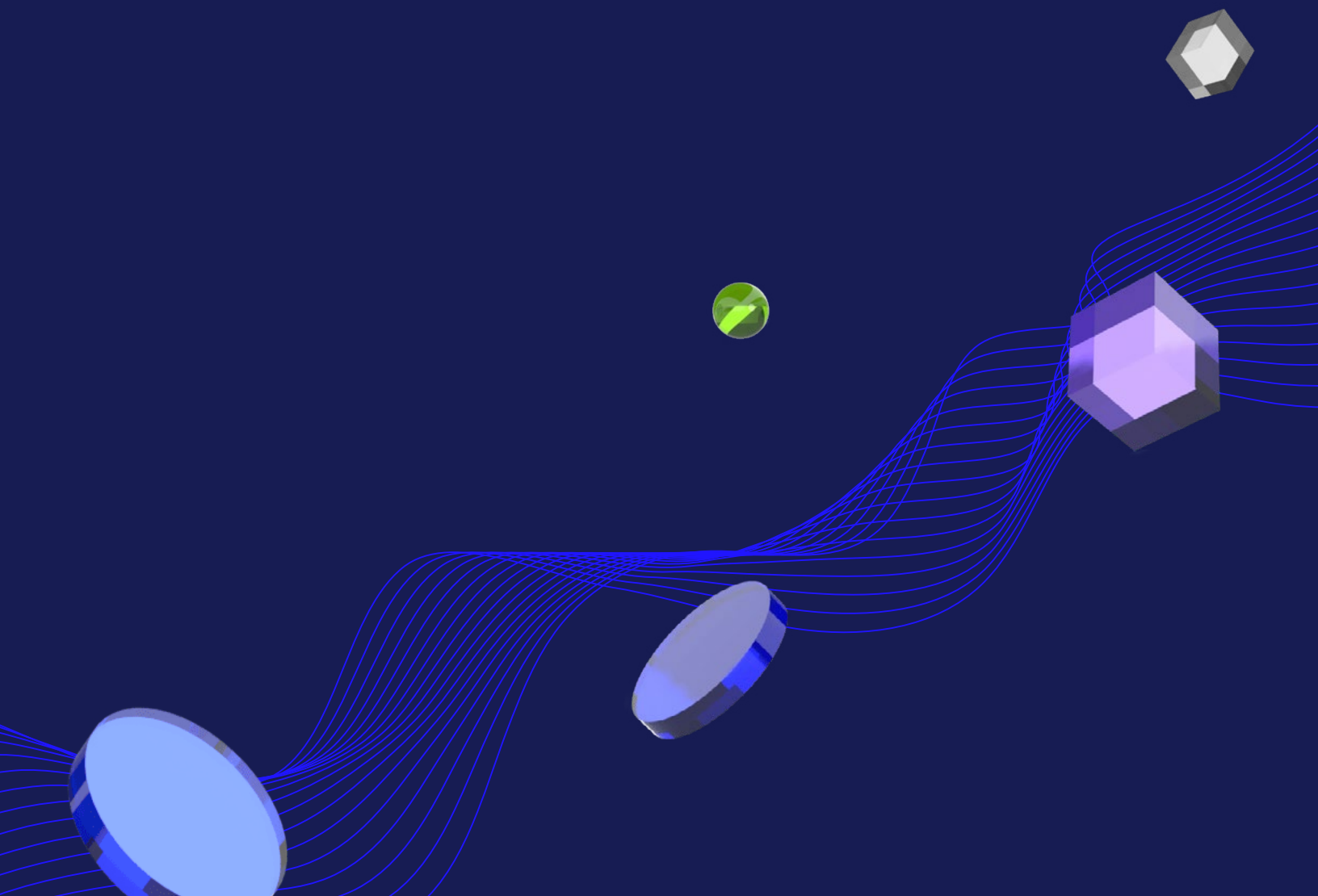




SCHOOL OF CLOUD COMPUTING

Cloud Developer using Microsoft Azure

Nanodegree Program Syllabus



Overview

Microsoft Azure is one of the most popular cloud services platforms used by enterprises, making it a crucial tool for cloud computing professionals to learn. The Cloud Developer using Microsoft Azure Nanodegree program teaches students how to deploy, build, migrate, and monitor applications on Azure, thereby preparing learners for success on Microsoft's AZ-204 Azure Developer Associate Expert certification.



Learning Objectives

A graduate of this program will be able to:

- Deploy storage and app solutions to Azure for a system that allow users to log in, view content, and publish content.
- Use Azure Functions, Cosmos DB, Event Hub to implement a serverless microservice back end architecture.
- Perform a “lift and shift” application migration to Azure through:
 - Migrate and deploy the preexisting web app to an Azure App Service.
 - Migrate a PostgreSQL database backup to an Azure Postgres database instance.
 - Use Azure Monitor, Azure App Insights, and Azure Metric Insights to enhance the performance of an application that has been deployed to Azure or post-migration.

Built in collaboration with:



Microsoft

Program information



Estimated Time

4 months at 10hrs/week*



Skill Level

Intermediate



Prerequisites

A well-prepared learner should have:

- At least 1-3 years of web development experience, preferably programming in Python.
- Experience creating and managing databases such as SQL Server or PostgreSQL.
- Comfortability using Git as a version control system to clone, pull, or push code.
- A free or existing Azure account, needed to create and provision Azure services.



Required Hardware/Software

Learners need access to a computer running recent versions of Windows, Mac OS X, or Linux and an unmetered broadband internet connection. They'll also need a free or existing Azure account in order to create and provision Azure services.

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Azure Applications

In this course, students will learn the basics of deploying an application to Azure. Understand the benefits and costs of cloud deployments, different types of service models, and how to navigate the Microsoft Azure platform. Distinguish two types of compute services in Azure, virtual machines and app services, when to use each, and how to deploy applications onto the related service. Connect two useful types of cloud storage to a cloud application, Azure SQL databases and blob containers. Differentiate between the different security options available in Azure and how to implement Microsoft's OAuth 2.0 capabilities with Azure Active Directory, along with useful monitoring and logging tools in the cloud.



Course Project

Deploy an Article CMS to Azure

Deploy an article content management system (CMS), built with a Python Flask application, to Microsoft Azure. The CMS system lets a user log in, view published articles, and publish new articles. First, deploy storage solutions for the application to interact with, such as a SQL database that contains a user table and an article table for the webapp to query, along with a blob storage container where images are stored. In addition to a simple username/password login, add an option to "Sign in with Microsoft" for authentication using OAuth 2.0 and Azure Active Directory. Lastly, add logging to the cloud application to be able to track successful or unsuccessful login attempts.

Lesson 1

Introduction to Microsoft Azure Development

- Assess the value, cost, and benefits of hosting their applications and systems in the cloud.
- Interpret the benefits/challenges of IaaS vs. PaaS vs. SaaS and where Microsoft Azure fits into each of these.
- Navigate the Azure Portal.

Lesson 2

Azure Compute Services

- Choose the correct Azure Compute Service for their applications and storage needs.
 - Create a subscription and resource group.
 - Manage and clean up resource groups.
 - Create a Linux VM.
 - Create an app service and app service plan.
 - Manage basic app service settings.
 - Justify which resource to use for app deployment based on costs, scalability, availability, and workflow.
-

Lesson 3

Azure Storage Options

- Create a SQL database in Azure.
 - Make updates to the schema of an Azure SQL database and add data to it.
 - Create an Azure blob storage container.
 - Upload blobs to the container.
 - Set up configurations for an app service to connect to both a SQL database and Azure blob storage container.
-

Lesson 4

Security & Monitoring Basics

- Differentiate security options between different Azure resources.
- Compare security responsibilities between the developer and cloud provider for different applications.
- Implement OAuth2 such as “Sign in with Microsoft” in their apps.
- Utilize Azure Active Directory for single sign-on and multi-factor authentication in their app.
- Monitor and analyze the log output of an app service running in Azure.

Azure Microservices

In this course, students will learn how to implement a serverless microservice back end architecture in Python using Azure cloud serverless offerings. Learners will compare and contrast common Azure microservices architecture and compute options, configure an instance of a MongoDB database with Azure CosmosDB, and allow the API to talk to this database. They will also apply enterprise logic apps and event grid to structure an application workflow. Finally, learners will deploy published API endpoints so they provide the necessary responses to complete the client-side requests of the front end web application.



Course Project

Deploying the Neighborly App with Azure Functions

Implement a serverless microservice back end architecture for a social networking web application called Neighborly, a service for neighbors to exchange helpful information, goods, and services. First, build the back end services that leverage an API to communicate with a MongoDB database. Then, integrate the client-side application and server-side API endpoints in Python. Finish by deploying and managing their service with AKS for future CI/CD integration.

Lesson 1

Introduction to Microservices

- Distinguish the benefits of a microservice architecture versus a monolithic architecture.
- Compare and contrast different services as a viable option for building microservices in Azure.
- Outline the different microservices patterns in Azure.
- Identify various compute options for microservice: Kubernetes Service, Docker, and Function (as a service).
- Identify the costs of implementing microservices.

Lesson 2

Serverless Functions

- Define Azure functions as well as common bindings and triggers.
 - Create Azure functions in the Azure CLI and VS IDE.
 - Create a MongoDB database in CosmosDB and connect it with Azure functions.
 - Secure an API endpoint with function-level authorization keys.
-

Lesson 3

Enterprise Logic Apps & Event Grid

- Create an Event Grid Topic in Azure Portal.
 - Apply Event Grid output binding for Azure Functions.
 - Create a workflow by using Azure Logic Apps.
 - Design a Logic App workflow for email.
-

Lesson 4

Deploying Your Services

- Deploy API endpoints in Azure CLI to Azure.
- Deploys a client-side Flask application to App Services.
- Dockerize an application to a container and upload it to Azure Container Registry.
- Deploy a client-side application to Kubernetes Service in Azure.

Course 3

Azure Migration

This course focuses on the techniques, processes, and nuances of migrating an existing application to Azure. It will cover the whole end-to-end process of an Azure migration from predicting costs of the migration to refactoring the code to ensure the application and corresponding databases are compatible with Azure. In addition, the course walks through best practices of the different application components migrated to Azure: web applications, background processes, and databases.



Course Project

Migrate App to Azure

Strategically migrate a preexisting conference registration system to Azure. Taking into consideration cost, architect a resilient and scalable system in Azure with the knowledge that the legacy application is very expensive, unable to scale at peak, has one single point of failure and performance issues, and is underutilized during off hours. First, migrate and deploy the preexisting web app to an Azure App Service. Then, migrate a PostgreSQL database backup to an Azure Postgres database instance. Finally, will refactor the notification logic to an Azure Function via a service bus queue message.

Lesson 1

Introduction to Cloud Migration

- Assess cloud migration needs.
- Identify different types of migration and their use cases.
- Implement a migration plan using the migration cycle.

Lesson 2

Migrating Web Applications

- Assess web application migration paths with cost prediction.
- Migrate existing web applications to Azure Web App.
- Migrate a web application using lift and shift method.

Lesson 3

Migrating Database Workload

- Assess Azure Database options with cost prediction.
- Migrate a database using Azure Database Migration Service.
- Manually migrate a database from backups.

Lesson 4

Migrating Background Job Processing Services

- Architect offline background job services with cost prediction.
- Migrate services to Azure Web Jobs, Azure Functions, and Azure Durable Functions.
- Migrate background job services to Azure Batch.

Azure Performance

This course enables students to acquire skills which allow them to collect data about the health and performance of an application, analyze and display the collected data to make informed decisions, and create automation to remedy application health or performance issues. Learners will set up and use Application Insights on a variety of Azure resources, and use the Application Insights SDK in a Python application to collect and transmit data about the application. They'll also query, transform, and display the collected application data so that the data can easily be analyzed, and use automation in Azure to manage cloud resources.



Course Project

Enhancing Applications

Collect and display performance and health data about an application post-migration to Azure. First, set up Application Insights monitoring on a virtual machine scale set (VMSS) and implement monitoring in an application to collect telemetry data. Then, create auto-scaling for a VMSS and an Azure Automation account to create a RunBook that automates the resolution of performance issues. Finally, create alerts to trigger auto-scaling on an Azure Kubernetes Service (AKS) cluster and trigger a Runbook to execute.

Lesson 1

Application Insights

- Set up Application Insights.
- Configure Application Insights on a VMSS.
- Configure Application Insights for an AKS cluster.
- Collect telemetry data.

Lesson 2

Application Analytics

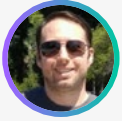
- Create an Azure Log Analytics Workspace.
 - Collect and ingest log data.
 - Write queries to return only relevant data of application performance.
 - Create reporting charts from data.
-

Lesson 3

Azure Management

- Create an Azure Automation Account.
- Autoscale rules for VMSS.
- Create a RunBook.

Meet your instructors.



Chris Vasquez

Full Stack Product Engineer at Udacity

After beginning in restaurant and retail management, Chris transitioned his career to tech, leveraging Udacity and other training platforms to master several different programming languages on his journey to becoming a full stack software engineer.



Ann K. Hoan

Engineer & Research Scientist

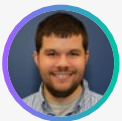
Ann is a senior software engineer and AI research scientist at Raytheon BBN Technologies, where she currently works on maritime predictive modelings for the Department of Defense. She also has experience building applications at startups and deploying AWS cloud infrastructure for the Department of Veterans Affairs.



Golois Mouelet

Premier Field Engineer at Microsoft

Golois is an experienced engineer specialized in building resilient and scalable cloud native applications as well as migrating software and systems to Azure. He has worked with enterprises on both cloud migration projects and application performance, and is Azure Certified both as a Developer and Solution Architect.



Nathan Anderson

DevOps Engineer at Goodyear Tire & Rubber Company

Nathan has worked on implementing DevOps solutions for the past 8 years across the financial, educational, logistics, and manufacturing industries.

A 3D illustration of various digital and tech icons. The scene includes a white document with a purple and green circular graphic, a green shield with a white checkmark, a yellow lightbulb, a blue folder with a white document inside, a blue code symbol (</>), a grey database cylinder, a green cube, a blue speech bubble with a white exclamation mark, a green speech bubble with a white checkmark, a blue speech bubble with a white question mark, and a small blue globe. The icons are arranged in a cluster, with some overlapping, creating a sense of depth and connectivity.



Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

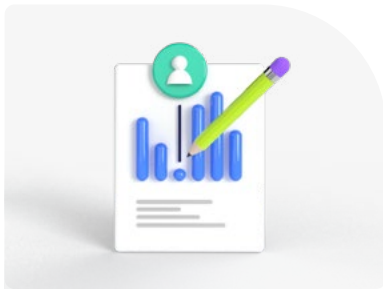


Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.



Take advantage of milestone reminders to stay on schedule and complete your program.

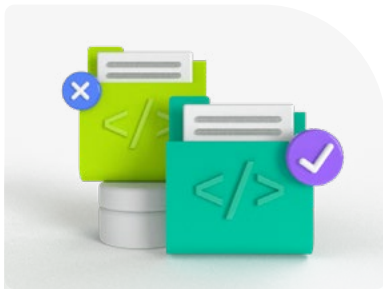
Our proven approach for building job-ready digital skills.



Pre-Assessments

Identify skills gaps.

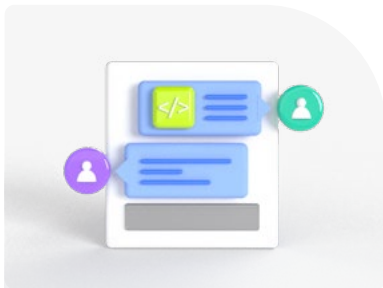
- In-depth assessments benchmark your team's current level of knowledge in key areas.
- Results are used to generate custom learning paths.



Experienced Project Reviewers

Verify skills mastery.

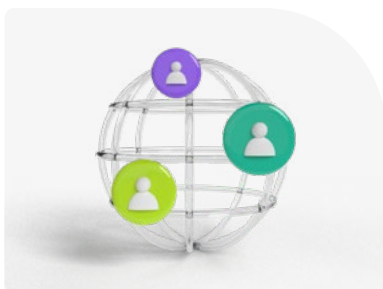
- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.



Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.



Dashboard & Reporting

Track course progress.

- Udacity's enterprise management console simplifies management of bulk enrollments and employee onboarding.
- Interactive views help achieve targeted results to increase retention and productivity.
- Maximize ROI while optimizing job readiness.



Learn more at

udacity.com/enterprise →