



NANODEGREE PROGRAM SYLLABUS

Cloud DevOps using Microsoft Azure



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 add to their skillset. The Developer for 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.

Prerequisites

A well-prepared learner has:

- 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

Educational Objectives

Students will learn to:

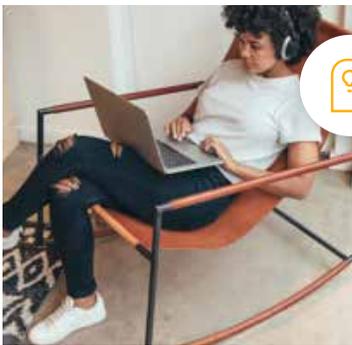
- Deploy storage and app solutions to Azure for a system that allow users to log in, view content, and publish content
- Use Azure Functions, Cosmos DB, Event Hub to implement a serverless microservice back-end architecture
- Perform a "lift and shift" application migration to Azure through:
 - Migrate and deploy the pre-existing web app to an Azure App Service
 - Migrate a PostgreSQL database backup to an Azure Postgres database instance
 - Refactor the notification logic to an Azure Function via a service bus queue message
 - 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



Estimated Time:
4 months at
10hrs/week



Instructional Tools Available:
Video lectures,
mentor-led student
community, forums,
project reviews



Flexible Learning:
Self-paced



Need Help?
udacity.com/advisor

*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.

Course 1: Azure Applications

In this course, you 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 1 Project

Deploy an Article CMS to Azure

You will 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, you will 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, you will then need to add an option to "Sign in with Microsoft" for authentication using OAuth 2.0 and Azure Active Directory. Lastly, you will add logging to the cloud application to be able to track successful or unsuccessful login attempts.

LEARNING OUTCOMES

LESSON ONE

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 TWO

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 THREE

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 FOUR

Security and 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



Course 2: Azure Microservices

In this course, you will learn how to implement a serverless microservice back-end architecture in Python using Azure cloud serverless offerings. You'll 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. You will also apply Enterprise Logic Apps and Event Grid to structure an application workflow. Finally, you'll deploy published API endpoints so they provide the necessary responses to complete the client-side requests of the front-end web application.

Course 2 Project

Deploying the Neighborly App with Azure Functions

In this project, you will 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, you'll build the back-end services that leverage an API to communicate with a MongoDB database. Then, you'll integrate the client-side application and server-side API endpoints in Python. You'll finish by deploying and managing their service with AKS for future CI/CD integration.

LEARNING OUTCOMES

LESSON ONE

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 TWO

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 THREE

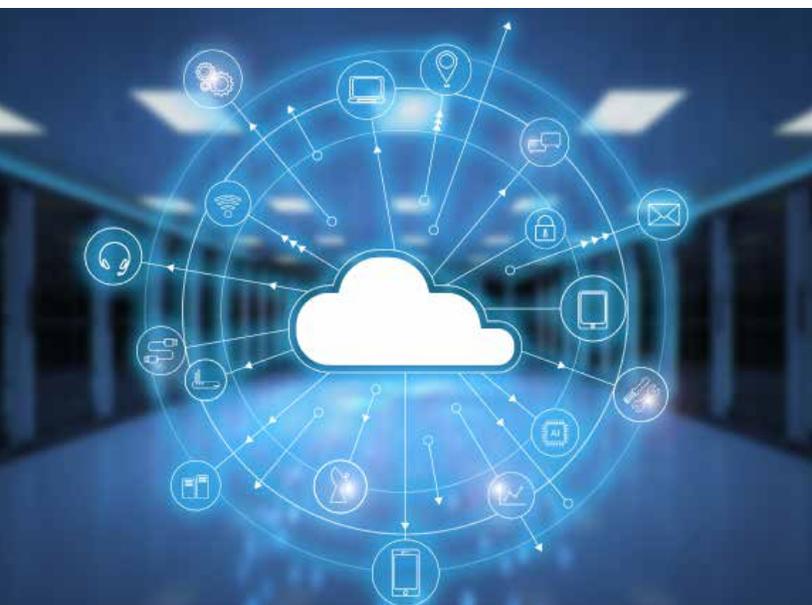
Enterprise Logic Apps and Event Grid

- 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 FOUR

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 3 Project

Migrate App to Azure

You will strategically migrate a pre-existing conference registration system to Azure. Taking in consideration cost, you will 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, you'll migrate and deploy the pre-existing web app to an Azure App Service. Then, you'll migrate a PostgreSQL database backup to an Azure Postgres database instance. Finally, you will refactor the notification logic to an Azure Function via a service bus queue message.

LEARNING OUTCOMES

LESSON ONE

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 TWO

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 THREE

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 FOUR

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

Course 4: Azure Performance

This course enables students to acquire skills which allow them to collect data about the health & performance of an application, analyze and display the collected data to make informed decisions, and create automation to remedy application health or performance issues. You 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. You'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 3 Project Enhancing Applications

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

LEARNING OUTCOMES

LESSON ONE

Application Insights

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

LESSON TWO

Application Analytics

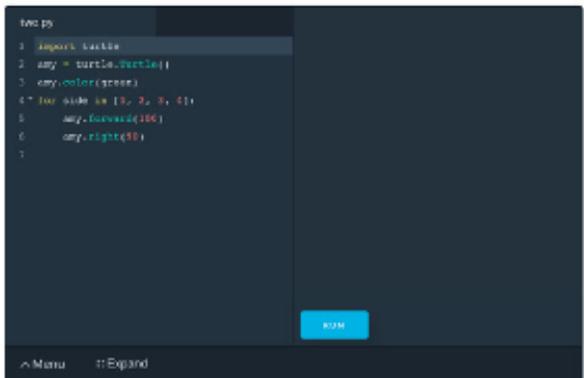
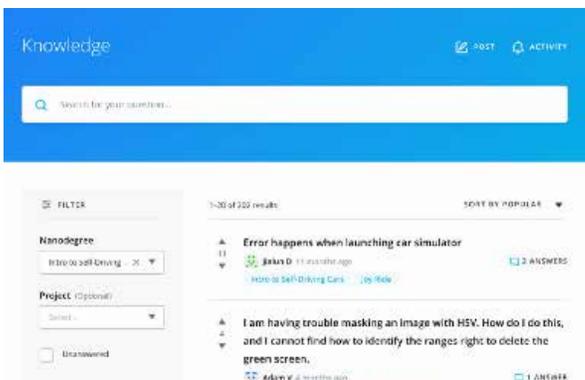
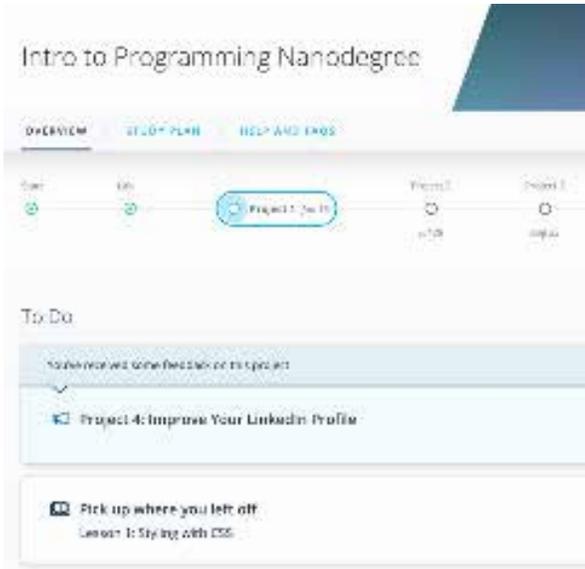
- Create an Azure Log Analytics Workspace
- Collect and ingest log data
- Write queries to return only relevant data of application performance
- Create reporting charts from data

LESSON THREE

Azure Management

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

Our Classroom Experience



REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

KNOWLEDGE

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students and discover in real-time how to solve the challenges that you encounter.

STUDENT HUB

Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with your technical mentor and fellow students in your Nanodegree program.

WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS

Work with a mentor to create a custom study plan to suit your personal needs. Use this plan to keep track of your progress toward your goal.

PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best



Chris Vasquez

FULL-STACK PRODUCT ENGINEER,
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 & 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 Dept. of Veterans Affairs.



Golois Mouelet

PREMIER FIELD ENGINEER, 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, GOODYEAR TIRE
& RUBBER COMPANY

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

All Our Nanodegree Programs Include:



EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve



TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions



PERSONAL CAREER SERVICES

CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW

WHY SHOULD I ENROLL?

Microsoft Azure is one of the fastest growing cloud service providers around, with 95% of Fortune 500 companies trusting the platform with their business. That's why companies across industries are looking to hire developers with Azure expertise, as Azure-related job growth is projected to hit almost 40% over the next decade. With Udacity's Developer for Microsoft Azure Nanodegree program, you will learn the skills you need to be in demand, and you'll be prepared to take the **Microsoft AZ-204 Certification exam**, proving your proficiency as an Azure Developer.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

Companies are hiring professionals with Azure skills for roles such as Cloud Developer, Full-Stack Software Engineer, Systems Architect, Security Engineer, Database Architect, Network Architect, SysAdmin, Cyber Analyst, Database Admin, and more.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

This Nanodegree program offers an ideal path for experienced software/web developers to advance their career. If you enjoy building web applications and want to learn how to build them on cloud, this is a great way to get hands-on practice with a variety of cloud computing principles and best practices. You should also review the prerequisites on this page to confirm you have the appropriate experience before beginning in the program.

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

There is no application. This Nanodegree program accepts everyone, regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

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



FAQs Continued

IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

Udacity's [Intro to Cloud Computing](#) course is a great way to brush up on foundational Cloud concepts. If you're looking to develop your programming skills, our [Intro to Programming](#) and [Full Stack Web Developer](#) Nanodegree programs would be the best pathway to take.

TUITION AND TERM OF PROGRAM

HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Developer for Microsoft Azure Nanodegree program is comprised of content and curriculum to support four projects. Once you subscribe to a Nanodegree program, you will have access to the content and services for the length of time specified by your subscription. We estimate that students can complete the program in four months, working 5-10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the [Terms of Use](#) and [FAQs](#) for other policies regarding the terms of access to our Nanodegree programs.

CAN I SWITCH MY START DATE? CAN I GET A REFUND?

Please see the Udacity Program [Terms of Use](#) and [FAQs](#) for policies on enrollment in our programs.

WHAT SOFTWARE AND VERSIONS WILL I NEED FOR THIS PROGRAM?

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

There are no specific hardware or software requirements for this program, other than those outlined on Udacity's general [Technology Requirements page](#).

