Azure DevOps Engineer Learning Pathway

Getting Started

Gain the knowledge and skills to design and implement DevOps processes and practices. Students will learn how to plan for DevOps, use source control, scale Git for an enterprise, consolidate artifacts, design a dependency management strategy, manage secrets, implement continuous integration, implement a container build strategy, design a release strategy, set up a release management workflow, implement a deployment pattern, and optimize feedback mechanisms

New to the Cloud or Azure? Start with Azure Fundamentals

- Build your Tech resilience
- Get started with Windows PowerShell
- Get started with Azure DevOps
- DevOps at Microsoft
- Want to get Started learning GitHub and DevOps?
- Threat Modelling Security Fundamentals
- Secure your infrastructure with threat modelling
- Explore DevOps Technology
- Intro to data protection and privacy regulations

Prerequisites:
Microsoft Certified: Azure Administrator or Microsoft Certified: Azure Developer

Check out the Administrator and Developer Pathways for more information on this training and certification

Additional Study

Continuous Integration

- Key concepts for new Azure Pipelines users
- Pipeline Tasks | Pipeline Agents
- Run quality tests in your build pipeline by using Azure Pipelines
- Quickstart: trigger a pipeline run from GitHub Actions
- Run functional tests in Azure Pipelines
- Overview of testing with Azure DevOps
- Create test plans and test suites
- Release approvals and gates overview
- Manage build dependencies with Azure Artifacts

Azure Artifacts: Feeds | Views

- Semantic Versioning (SemVer)
- Use a PowerShell script to customize your pipeline
- Desired State Configuration (DSC)

Azure Pipelines: Agents | Triggers

- Configure infrastructure in Azure Pipelines

Continuous Delivery

- Infrastructure as code
- Create a release pipeline in Azure Pipelines
- Deployment Jobs
- Migrate from Jenkins to Azure Pipelines
- Automate deployments with Release Management
- What is Azure Pipelines?
- CI/CD Deployment: Classic pipelines
- Deploy applications with Azure DevOps
- Third Party DevOps solutions
- Azure Automation State Configuration overview
- Exercise: Hotfix Changes using releases

Site Reliability Engineering

- SRE principles and practices: virtuous cycles

Azure Monitor: Action Groups | Schema

- Health monitoring
- Service Fabric health monitoring

Communication & Collaboration

- About dashboards/ charts/reports/widgets
- DevOps Dashboards | Adding a Chart
- Azure DevOps Reporting
- Azure Boards documentation
- Azure Boards & GitHub
- Link GitHub commits, pull requests, and issues to work items
- Tutorial: Follow a user story, bug, issue, or other work item or pull request
- AbRelease artifacts and artifact sources
- Integrate with service hooks
- Create a service hook for Azure DevOps with Microsoft Teams
- Create a service hook for Azure DevOps with Slack
- Integrate third-party services
- Webhooks

Instrumentation and

- Use monitoring and analytics to gain operational insights
- Designing your Azure Monitor Logs deployment
- Roles, permissions, and security
- What is Distributed Tracing?
- Unify monitoring solutions in Azure
- Identify performance bottlenecks
- Monitoring solutions in Azure Monitor
- Analyse alerts to establish a baseline
- Analyse and understand mobile application use

Security/Compliance

- Identity and access | Managed Identities
- Azure Active Directory groups
- Service Endpoints
- What is Azure Key Vault?
- Azure Key Vault – Overview
- Configure and manage secrets in Key Vault
- Key Vault certificates
- Using secrets from Key Vault in a pipeline
- Scanning Open-Source Libraries
- Protect your cloud workloads
- Understand Security Considerations for Application Lifecycle Management Solution
- Identity and access | Managed Identities
- Azure Active Directory groups
- Service Endpoints
- Protect your cloud workloads
- Understand Security Considerations for Application Lifecycle Management Solution

Source Control

- Monorepo vs. multi-repo
- Choosing the right version control for your project
- What is Azure Repos?
- Improve code quality with branch policies
- Check out multiple repositories in your pipeline
- Managing code review assignment for your team
- Create your first pipeline
- Create a build pipeline
- Tutorial: Azure Active Directory single sign-on (SSO) integration with a GitHub Enterprise Cloud Organization
- Is GitOps the next big thing in DevOps?
- ChatOps - Communication and collaboration
- DevOps Resource Center
- Enabling resilient DevOps practices with code to cloud automation
- Cloud Adoption Framework for Azure
- Introduction to Azure DevTest Labs

Role Based Certification

Microsoft Certified: DevOps Engineer Expert

Prerequisite: Azure Admin (AZ-104) or Azure Developer (AZ-204)

AZ-400: Designing and Implementing Microsoft DevOps Solutions

Skills measured:

- Configure processes and communications (10-15%)
- Design and implement source control (15-20%)
- Design and implement build and release pipelines (40-45%)
- Develop a security and compliance plan (10-15%)
- Implement an instrumentation strategy (10-15%)

- Getting started on a DevOps transformation journey
- Work with Git for enterprise DevOps
- Implement CI with Azure Pipelines and GitHub Actions
- Design and implement a release strategy
- Implement a secure continuous deployment using Azure Pipelines
- Manage Infrastructure as code using Azure and DSC
- Design and implement a dependency management strategy
- Implement continuous feedback
- Implement security and validate code for compliance

Exam Study Guide

Course Page

30 Days to Learn Challenge!

Exam Page

Azure DevOps Documentation

Practice Test

Check out the GitHub Learning Companion for more learning resources

GitHub Learning Resources

Microsoft Learn / Documentation

GitHub fundamentals - Administration basics and product features.

Learn the basics of GitHub administration and discover the products that GitHub offers.

- Introduction to GitHub
- Introduction to GitHub administration
- Introduction to GitHub's Products

Learning More:

- Introduction to GitHub in Visual Studio Code
- Get started with Git and GitHub in Visual Studio
- Want to get Started learning GitHub and DevOps

Automate workflow with GitHub Actions

Learn how GitHub Actions enables you to automate your software development cycle and deploy applications to Azure

START

Collaborate with others with Markdown and GitHub Pages

Learn how to use Markdown to effectively communicate with others in your GitHub issues, pull requests, comments, and documentation.

START

Introduction to GitHub Packages

Explore ways to control permissions and visibility, publish, install, delete and restore packages using GitHub

START

Deploy Azure resources by using Bicep and GitHub Actions

Gain all of the benefits of infrastructure as code by using an automated workflow to deploy your Bicep templates, and integrate other deployment activities with your workflows. You'll build workflows using GitHub Actions.

START

Manage Source Control

Learn how GitHub enables you to build a modern source control strategy that fosters collaboration and enables you to automate your build and deployment processes.

START

Build Community-Driven Software Projects

Whether you manage enterprise or open-source software projects, learn how GitHub enables you to build communities that foster communication and collaboration while reinforcing recommended guidelines, codes of conduct, and security best practices.

START

Manage the lifecycle of your products

Take full control of your GitHub projects. Through work planning and tracking, effective branching strategies, and extending GitHub through its API, manage releases all the way from idea to working software in the hands of your users.

- Introduction to GitHub
- Manage software delivery by using a release-based workflow on GitHub
- Manage repository changes by using pull requests on GitHub
- Settle competing commits by using merge conflict resolution on GitHub
- Automate DevOps processes by using GitHub Apps
- Search and organize repository history

Work with Azure Repos and GitHub

This module introduces you to Azure Repos and GitHub and explores ways to migrate from TFVC to Git, and work with GitHub Codespaces for development.

START

Plan Agile with GitHub Projects

This module introduces you to GitHub Projects, GitHub Project Boards and Azure Boards. It explores ways to link

Azure Boards and GitHub, configure GitHub Projects and Project views, and manage work with GitHub Projects.

START

Doing More

- GitHub administration for enterprise support and adoption
- Deploy a cloud-native ASP.NET Core microservice with GitHub Actions
- Azure Kubernetes Service deployment pipeline and GitHub Actions
- Create and publish custom GitHub actions
- Manage GitHub Actions in the enterprise
- Deploy and maintain cloud-native apps with GitHub actions and Azure Pipelines
- Learn how Microsoft supports secure software development as part of a cybersecurity solution
- Introduction to GitHub Advanced Security
- Configure and use secret scanning in your GitHub repository
- Authenticate and authorize user identities on GitHub
- Manage sensitive data and security policies within GitHub
- GitHub administration for GitHub Advanced Security
- Introduction to GitHub in Visual Studio Code
- Architect full-stack applications and automate deployments with GitHub
- Configure Dependabot security updates on your GitHub repo
- Configure code scanning on GitHub
- Code scanning with GitHub CodeQL

GitHub Guides | guides.github.com

Expand your knowledge with GitHub Guides
Explore feature demos, tips, and workflow essentials for effective daily GitHub use.

1. GitHub Flow
2. Hello World
3. GitHub Pages
4. Git Handbook
5. Forking Projects
6. Be Social
7. Making your code Citable
8. Mastering Issues
9. Mastering Markdown
10. Documenting your Projects

GitHub Learning Lab

https://lab.github.com/

GitHub Docs

https://docs.github.com/

YouTube GitHub Training & Guides