

Azure DevOps Engineer Learning Pathway

www.aka.ms/pathways

Get 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)

- 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 monitoring

- 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

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

[Azure DevOps Documentation](#)

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 Design and implement source control
- Design and implement build and release pipelines
- Develop a security and compliance plan
- Implement an instrumentation strategy

Microsoft Learn:

- Getting started on a DevOps transformation journey
- Development 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](#)

[Exam Page](#)

[Practice Assessment](#)

30 days to Learn it Challenge

30 Days to Learn It can help you build skills and start your preparation for Microsoft Certifications. Select your challenge below.

[Azure DevOps Engineer](#)

