Introduction

Completed

- 1 minute

Do you currently use tools to manage daily work and plan for the future? It is essential when implementing DevOps if you want to increase your maturity level.

This module explores Azure DevOps and GitHub tools and helps organizations define their work management tool and licensing strategy.

Learning objectives

After completing this module, students and professionals can:

- Design a tool integration strategy.
- Design a license management strategy (for example, Azure DevOps and GitHub users).
- Design a strategy for end-to-end traceability from work items to working software.
- Design an authentication and access strategy.
- Design a strategy for integrating on-premises and cloud resources.

Prerequisites

- Understanding of what DevOps is and its concepts.
- Familiarity with DevOps tools is helpful but is not necessary.
- Beneficial to have experience in an organization that delivers software.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

What is Azure DevOps?

Completed

- 2 minutes

Azure DevOps is a Software as a service (SaaS) platform from Microsoft that provides an end-to-end DevOps toolchain for developing and deploying software.

It also integrates with the most-leading tools on the market and is an excellent option for orchestrating a DevOps toolchain.

What does Azure DevOps provide?

Azure DevOps includes a range of services covering the complete development life cycle.

- Azure Boards: agile planning, work item tracking, visualization, and reporting tool.
- Azure Pipelines: a language, platform, and cloud-agnostic CI/CD platform-supporting containers or Kubernetes.
- Azure Repos: provides cloud-hosted private git repos.
- Azure Artifacts: provides integrated package management with support for Maven, npm, Python, and NuGet package feeds from public or private sources.
- Azure Test Plans: provides an integrated planned and exploratory testing solution.

Also, you can use Azure DevOps to orchestrate third-party tools.

What if we are not a Microsoft / Microsoft .NET organization?

Azure DevOps is not focused on organizations that are end-to-end Microsoft or Windows.

Azure DevOps provides a platform that is:

- Flexible: you do not have to go 'all in' on Azure DevOps. It is possible to adopt each of the services independently and integrate them with your existing toolchain; most popular tools are supported.
- Cross-Platform: designed to work with any platform (Linux, macOS, and Windows). Or language (including Node.js, Python, Java, PHP, Ruby, C/C++, .NET, Android, and iOS apps). Azure DevOps is not aimed at organizations building and shipping on the Microsoft technology stack.
- Cloud Agnostic: continuous delivery is supported to AWS, GCP, and Azure.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

What is GitHub?

Completed

- 1 minute

GitHub is a Software as a service (SaaS) platform from Microsoft that provides Git-based repositories and DevOps tooling for developing and deploying software.

It has a wide range of integrations with other leading tools.

What does GitHub provide?

GitHub provides a range of services for software development and deployment.

- **Codespaces:** Provides a cloud-hosted development environment (based on Visual Studio Code) that can be operated from within a browser or external tools. Eases cross-platform development.
- **Repos:** Public and private repositories based upon industry-standard Git commands.
- **Actions:** Allows for the creation of automation workflows. These workflows can include environment variables and customized scripts.
- **Packages:** The majority of the world's open-source projects are already contained in GitHub repositories. GitHub makes it easy to integrate with this code and with other third-party offerings.
- **Security:** Provides detailed code scanning and review features, including automated code review assignment.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Explore an authorization and access strategy

Completed

- 3 minutes

Azure DevOps Services uses enterprise-grade authentication. To protect and secure your data, you can use:

- Microsoft account.

- GitHub account.
- Microsoft Entra ID.

Tools like Visual Studio and Azure DevOps natively support the use of Microsoft Accounts and Microsoft Entra ID. Eclipse can also support this form of authentication if you install a Team Explorer Everywhere plug-in.

Personal access tokens

Use personal access tokens (PAT) for tools that don't directly support Microsoft accounts or Microsoft Entra ID for authentication. You can use it if you want them to integrate with Azure DevOps.

For example, tools like:

- Git-based repositories.
- NuGet.
- Xcode.

These tokens can be set up using Git Credential managers, or you can create them manually.

Personal access tokens are also helpful when establishing access to command-line tools, external tools, and tasks in build pipelines.

Also, when calling REST-based APIs, you don't have a UI popping out to do the authentication. When access is no longer required, you can revoke the personal access token.

Security groups

Azure DevOps is pre-configured with default security groups.

Default permissions are assigned to the default security groups. You can also configure access at the organization, collection, and project or object levels.

In the organization settings in Azure DevOps, you can configure app access policies. Based on your security policies, you might allow alternate authentication methods, enable third-party applications to access via OAuth, or even allow anonymous access to some projects.

For even tighter control, you can use Conditional Access policies. These offer simple ways to help secure resources such as Azure DevOps when using Microsoft Entra ID for authentication.

Multifactor authentication

Conditional Access policies such as multifactor authentication can help to minimize the risk of compromised credentials.

As part of a Conditional Access policy, you might require:

- Security group membership.
- A location or network identity.
- A specific operating system.
- A managed device or other criteria.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Migrate or integrate existing work management tools

Completed

- 2 minutes

Both Azure DevOps and GitHub can be integrated with different kinds of work management tools.

As an example, in the Visual Studio Marketplace, Microsoft offers Trello integration tooling.

Migrating from other work management tools to Azure DevOps takes considerable planning.

Most work management tools are highly configurable by the end user. There might not be a tool available that will do the migration without further configuration.

Jira

Jira is a commonly used work management tool.

In the Visual Studio Marketplace, Solidify offers a tool for Jira to Azure DevOps migration. It migrates in two phases. Jira issues are exported to files, and then the files are imported to Azure DevOps.

If you decide to write the migration code yourself, the following blog post provides a sample code that might help you get started: Migrate your project from Jira to Azure DevOps .

Other applications

Third-party organizations offer commercial tooling to assist with migrating other work management tools like:

- Aha.
- BugZilla.
- ClearQuest.
- And others to Azure DevOps.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Migrate or integrate existing test management tools

Completed

- 1 minute

Azure Test Plans track manual testing for sprints and milestones, allowing you to follow when that testing is complete.

Azure DevOps also has a Test Feedback extension available in the Visual Studio Marketplace. The extension is used to help teams do exploratory testing and provide feedback.

All team members and other stakeholders can use the extension to submit bugs or provide feedback. For example:

- Developers.
- Product owners.
- Managers.
- UX.
- UI engineers.
- Marketing teams.
- Early adopters.

For load tests, you can use Azure Load Testing. For more information, see What is Azure Load Testing? .

Other helpful testing tools:

Apache JMeter is open-source software written in Java and designed to load test, and measure performance.

Pester is a tool that can automate the testing of PowerShell code.

SoapUI is another testing framework for SOAP and REST testing.

If you are using Microsoft Test Manager, you should plan to migrate to Azure Test Plans instead.

For more information, search for Test Management at Visual Studio Marketplace.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Design a license management strategy

Completed

- 1 minute

When designing a license management strategy, you first need to understand your progress in the DevOps implementation phase.

If you have a draft of the architecture, you're planning for the DevOps implementation; you already know part of the resources to consume.

For example, you started with a version control-implementing Git and created some pipelines to build and release your code.

If you have multiple teams building their solutions, you don't want to wait in the queue to start building yours.

Probably, you want to pay for parallel jobs and make your builds run in parallel without depending on the queue availability.

To consider:

- What phase are you in?
- How many people are using the feature?
- How long are you willing to wait if in the queue for pipelines? Is this urgent? Is this a validation only?
- Should all users access all features? Are they Stakeholders? Basic users? Do they already have a Visual Studio license?
- Do you have an advanced Package Management strategy? Maybe you need more space for Artifacts.

For the latest, most up-to-date Azure DevOps pricing information, visit Azure DevOps Pricing .

For the latest, most up-to-date GitHub pricing information, visit GitHub Pricing .

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Knowledge check

Completed

- 4 minutes

Choose the best response for each question. Then select **Check your answers** .

Check your knowledge

1.

Which of the following choices isn't an Azure DevOps service/feature?

○

Azure Boards.

○

Azure Monitor.

○

Azure Repos.

2.

Which of the following is a correct statement about Azure DevOps?

○

Azure DevOps only works in on-premises environments. If you want to use a cloud service, you need to choose GitHub.

○

Azure DevOps and GitHub are only available in the cloud.

○

Azure DevOps and GitHub provide options to work on-premises and in the cloud.

3.

Which of the following choices does Azure DevOps Services enterprise-grade authentication support?

○

Microsoft Account, Microsoft Entra ID, GitHub Account.

○

Microsoft Account, Microsoft Entra ID.

○

GitHub Account.

Check your answers

You must answer all questions before checking your work.

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .

Summary

Completed

- 1 minute

This module explored the basics of Azure DevOps and GitHub and how to start using them to implement DevOps.

You learned how to describe the benefits and usage of:

- Design a tool integration strategy.
- Design a license management strategy (for example, Azure DevOps and GitHub users).
- Design a strategy for end-to-end traceability from work items to working software.
- Design an authentication and access strategy.
- Design a strategy for integrating on-premises and cloud resources.

Learn more

- Azure DevOps Services | Microsoft Azure .
- Features | GitHub .
- Pricing Calculator | Microsoft Azure .
- Azure DevOps Server to Services Migration overview - Azure DevOps | Microsoft Docs .
- Azure DevOps Services Pricing | Microsoft Azure .
- GitHub Pricing .

Continue

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue .