**Explore Azure Pipelines**

Advanced

Administrator

Developer

DevOps Engineer

Security Engineer

Security Operations Analyst

Service Adoption Specialist

Solution Architect

Technology Manager

Azure

Azure Artifacts

Azure Boards

Azure Cloud Services

Azure DevOps

Azure Pipelines

Azure Repos

Azure Test Plans

GitHub

This module introduces Azure Pipelines concepts and explains key terms and components of the tool, helping you decide your pipeline strategy and responsibilities.

**Learning objectives**

By the end of this module, you're able to:

* Describe Azure Pipelines
* Explain the role of Azure Pipelines and its components
* Decide Pipeline automation responsibility
* Understand Azure Pipeline key terms

[**Start**](https://learn.microsoft.com/en-us/training/modules/explore-azure-pipelines/1-introduction/)Add

**Prerequisites**

None

**This module is part of these learning paths**

* [AZ-400: Implement CI with Azure Pipelines and GitHub Actions](https://learn.microsoft.com/training/paths/az-400-implement-ci-azure-pipelines-github-actions/)
* [Explore Azure DevOps with GitHub to streamline your development process](https://learn.microsoft.com/training/paths/explore-azure-devops-with-github/)

**Module assessment**

Assess your understanding of this module. Sign in and answer all questions correctly to earn a pass designation on your profile.

[**Take the module assessment**](https://learn.microsoft.com/training/modules/explore-azure-pipelines/5-knowledge-check/)

* [Introduction](https://learn.microsoft.com/en-us/training/modules/explore-azure-pipelines/1-introduction)1 min
* [Explore the concept of pipelines in DevOps](https://learn.microsoft.com/en-us/training/modules/explore-azure-pipelines/2-explore-concept-of-pipelines-devops)3 min
* [Describe Azure Pipelines](https://learn.microsoft.com/en-us/training/modules/explore-azure-pipelines/3-describe-azure-pipelines)3 min
* [Understand Azure Pipelines key terms](https://learn.microsoft.com/en-us/training/modules/explore-azure-pipelines/4-understand-key-terms)3 min
* [Knowledge check](https://learn.microsoft.com/en-us/training/modules/explore-azure-pipelines/5-knowledge-check)6 min
* [Summary](https://learn.microsoft.com/en-us/training/modules/explore-azure-pipelines/6-summary)1 min

**Introduction**

Completed100 XP

* 1 minute

Azure Pipelines is a fully featured service used to create cross-platform CI (Continuous Integration) and CD (Continuous Deployment). It works with your preferred Git provider and can deploy to most major cloud services, including Azure. Azure DevOps offers a comprehensive Pipelines offering.

This module introduces Azure Pipelines concepts and explains key terms and components of the tool, helping you decide your pipeline strategy and responsibilities.

**Learning objectives**

After completing this module, students and professionals can:

* Describe Azure Pipelines.
* Explain the role of Azure Pipelines and its components.
* Decide Pipeline automation responsibility.
* Understand Azure Pipeline key terms.

**Prerequisites**

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

**Next unit: Explore the concept of pipelines in DevOps**

**Explore the concept of pipelines in DevOps**

Completed100 XP

* 3 minutes

Business demands continuous delivery of value. Value is created only when a product is delivered to a satisfied customer.

It's not created when one silo in the process is completed.

It demands that you reset focus from silos to an end-to-end flow of value.

The core idea is to create a repeatable, reliable, and incrementally-improving process for taking software from concept to customer.

The goal is to enable a constant flow of changes into production via an automated software production line.

Think of it as a pipeline. The pipeline breaks down the software delivery process into stages.

Each stage aims to verify the quality of new features from a different angle to validate the new functionality and prevent errors from affecting your users.

The pipeline should provide feedback to the team. Also, visibility into the changes flows to everyone involved in delivering the new feature(s).

A delivery pipeline enables the flow of more minor changes more frequently, with a focus on flow.

Your teams can concentrate on optimizing the delivery of changes that bring quantifiable value to the business.

This approach leads teams to continuously monitor and learn where they're finding obstacles, resolve those issues, and gradually improve the pipeline's flow.

As the process continues, the feedback loop provides new insights into new issues and barriers to be resolved.

The pipeline is the focus of your continuous improvement loop.

A typical pipeline will include the following stages: build automation and continuous integration, test automation, and deployment automation.

**Build automation and continuous integration**

The pipeline starts by building the binaries to create the deliverables passed to the following stages. New features implemented by the developers are integrated into the central code base, built, and unit tested. It's the most direct feedback cycle that informs the development team about the health of their application code.

**Test automation**

The new version of an application is rigorously tested throughout this stage to ensure that it meets all wished system qualities. It's crucial that all relevant aspects—whether functionality, security, performance, or compliance—are verified by the pipeline. The stage may involve different types of automated or (initially, at least) manual activities.

**Deployment automation**

A deployment is required every time the application is installed in an environment for testing, but the most critical moment for deployment automation is rollout time.

Since the preceding stages have verified the overall quality of the system, It's a low-risk step.

The deployment can be staged, with the new version being initially released to a subset of the production environment and monitored before being rolled out.

The deployment is automated, allowing for the reliable delivery of new functionality to users within minutes if needed.

**Your pipeline needs platform provisioning and configuration management**

The deployment pipeline is supported by platform provisioning and system configuration management. It allows teams to create, maintain, and tear down complete environments automatically or at the push of a button.

Automated platform provisioning ensures that your candidate applications are deployed to, and tests carried out against correctly configured and reproducible environments.

It also helps horizontal scalability and allows the business to try out new products in a sandbox environment at any time.

**Orchestrating it all: release and pipeline orchestration**

The multiple stages in a deployment pipeline involve different groups of people collaborating and supervising the release of the new version of your application.

Release and pipeline orchestration provide a top-level view of the entire pipeline, allowing you to define and control the stages and gain insight into the overall software delivery process.

By carrying out value stream mappings on your releases, you can highlight any remaining inefficiencies and hot spots and pinpoint opportunities to improve your pipeline.

These automated pipelines need infrastructure to run on. The efficiency of this infrastructure will have a direct impact on the effectiveness of the pipeline.

**Next unit: Describe Azure Pipelines**

**Describe Azure Pipelines**

Completed100 XP

* 3 minutes

Azure Pipelines is a cloud service that automatically builds and tests your code project and makes it available to other users. It works with just about any language or project type.

Azure Pipelines combines continuous integration (CI) and continuous delivery (CD) to test and build your code and ship it to any target constantly and consistently.

**Does Azure Pipelines work with my language and tools?**

Azure Pipelines is a fully featured cross-platform CI and CD service. It works with your preferred Git provider and can deploy to most major cloud services, including Azure services.

**Languages**

You can use many languages with Azure Pipelines, such as Python, Java, PHP, Ruby, C#, and Go.

**Version control systems**

Before you use continuous integration and continuous delivery practices for your applications, you must have your source code in a version control system. Azure Pipelines integrates with GitHub, GitLab, Azure Repos, Bitbucket, and Subversion.

**Application types**

You can use Azure Pipelines with most application types, such as Java, JavaScript, Python, .NET, PHP, Go, XCode, and C++.

**Deployment targets**

Use Azure Pipelines to deploy your code to multiple targets. Targets including:

* Container registries.
* Virtual machines.
* Azure services, or any on-premises or cloud target such:
  + Microsoft Azure.
  + Google Cloud.
  + Amazon Web Services (AWS).

**Package formats**

To produce packages that others can consume, you can publish NuGet, npm, or Maven packages to the built-in package management repository in Azure Pipelines.

You also can use any other package management repository of your choice.

**Why should I use CI and CD, and Azure Pipelines?**

Implementing CI and CD pipelines help to ensure consistent and quality code that's readily available to users.

Azure Pipelines is a quick, easy, and safe way to automate building your projects and making them available to users.

**Use CI and CD for your project**

Continuous integration is used to automate tests and builds for your project. CI helps to catch bugs or issues early in the development cycle when they're easier and faster to fix. Items known as artifacts are produced from CI systems. The continuous delivery release pipelines use them to drive automatic deployments.

Continuous delivery is used to automatically deploy and test code in multiple stages to help drive quality. Continuous integration systems produce deployable artifacts, which include infrastructure and apps. Automated release pipelines consume these artifacts to release new versions and fixes to the target of your choice.

Expand table

| **Continuous integration (CI)** | **Continuous delivery (CD)** |
| --- | --- |
| Increase code coverage. | Automatically deploy code to production. |
| Build faster by splitting test and build runs. | Ensure deployment targets have the latest code. |
| Automatically ensure you don't ship broken code. | Use tested code from the CI process. |
| Run tests continually. |  |

**Use Azure Pipelines for CI and CD**

There are several reasons to use Azure Pipelines for your CI and CD solution. You can use it to:

* Work with any language or platform.
* Deploy to different types of targets at the same time.
* Integrate with Azure deployments.
* Build on Windows, Linux, or macOS machines.
* Integrate with GitHub.
* Work with open-source projects.

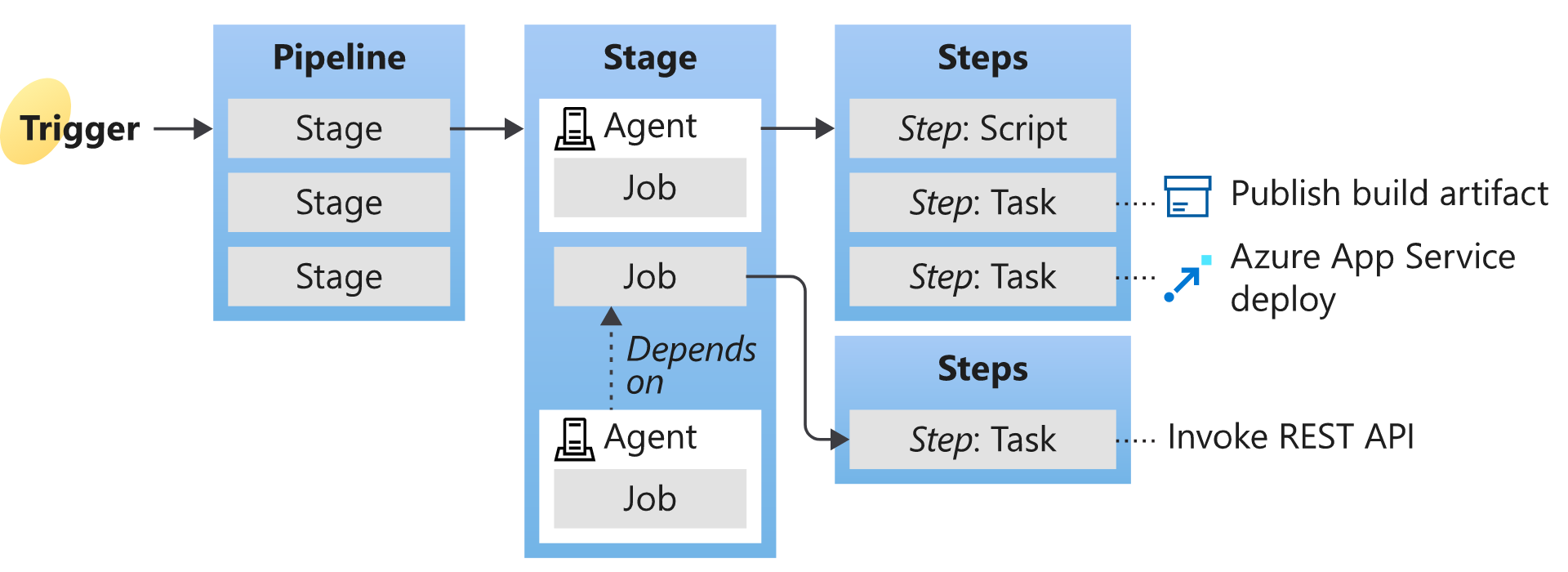
**Next unit: Understand Azure Pipelines key terms**

**Understand Azure Pipelines key terms**

Completed100 XP

* 3 minutes

Understanding the basic terms and parts of Azure Pipelines helps you further explore how it can help you deliver better code more efficiently and reliably.



**Agent**

When your build or deployment runs, the system begins one or more jobs. An agent is installable software that runs a build or deployment job.

**Artifact**

An artifact is a collection of files or packages published by a build. Artifacts are made available for the tasks, such as distribution or deployment.

**Build**

A build represents one execution of a pipeline. It collects the logs associated with running the steps and the test results.

**Continuous delivery**

Continuous delivery (CD) (also known as Continuous Deployment) is a process by which code is built, tested, and deployed to one or more test and production stages. Deploying and testing in multiple stages helps drive quality. Continuous integration systems produce deployable artifacts, which include infrastructure and apps. Automated release pipelines consume these artifacts to release new versions and fix existing systems. Monitoring and alerting systems constantly run to drive visibility into the entire CD process. This process ensures that errors are caught often and early.

**Continuous integration**

Continuous integration (CI) is the practice used by development teams to simplify the testing and building of code. CI helps to catch bugs or problems early in the development cycle, making them more accessible and faster to fix. Automated tests and builds are run as part of the CI process. The process can run on a schedule, whenever code is pushed, or both. Items known as artifacts are produced from CI systems. The continuous delivery release pipelines use them to drive automatic deployments.

**Deployment target**

A deployment target is a virtual machine, container, web app, or any service used to host the developed application. A pipeline might deploy the app to one or more deployment targets after the build is completed and tests are run.

**Job**

A build contains one or more jobs. Most jobs run on an agent. A job represents an execution boundary of a set of steps. All the steps run together on the same agent.

For example, you might build two configurations - x86 and x64. In this case, you have one build and two jobs.

**Pipeline**

A pipeline defines the continuous integration and deployment process for your app. It's made up of steps called tasks.

It can be thought of as a script that describes how your test, build, and deployment steps are run.

**Release**

When you use the visual designer, you can create a release or a build pipeline. A release is a term used to describe one execution of a release pipeline. It's made up of deployments to multiple stages.

**Stage**

Stages are the primary divisions in a pipeline: "build the app," "run integration tests," and "deploy to user acceptance testing" are good examples of stages.

**Task**

A task is the building block of a pipeline. For example, a build pipeline might consist of build and test tasks. A release pipeline consists of deployment tasks. Each task runs a specific job in the pipeline.

**Trigger**

A trigger is set up to tell the pipeline when to run. You can configure a pipeline to run upon a push to a repository at scheduled times or upon completing another build. All these actions are known as triggers.

**Next unit: Knowledge check**

**Knowledge check**

Completed200 XP

* **Module assessment**
* 6 minutes

 Answer 100% of questions correctly in order to pass. [**Retake**](https://learn.microsoft.com/en-us/training/modules/explore-azure-pipelines/5-knowledge-check)

Dismiss alert

Choose the best response for each question.

**Check your knowledge**

Top of Form

**1.**

**Which of the following choices aren't characteristic of a pipeline?**

Reliable.

**Incorrect. The core pipeline idea is to create a repeatable, reliable, and incrementally improving process.**

Repeatable.

Immutable.

**Correct. The core pipeline idea is to create a repeatable, reliable, and incrementally improving process.**

**2.**

**Which of the following is a correct statement about Azure Pipelines?**

Azure Pipelines works exclusively with Azure and Microsoft Technology. For open-source, GitHub is recommended.

Azure Pipelines work with any language or platform.

**Correct. Azure Pipelines work with any language or platform.**

Azure Pipelines only deploys in cloud environments.

**3.**

**Which of the following choices is a Continuous Integration (CI) benefit?**

Automatically ensure you don't ship broken code.

**Correct. Automatically ensure you don't ship broken code.**

Automatically deploy code to production.

Ensure deployment targets have the latest code.

**Incorrect. Automatically ensure you don't ship broken code.**

Bottom of Form

**Summary**

Completed100 XP

* 1 minute

This module introduced Azure Pipelines concepts and explained key terms and components of the tool. It helped you decide your pipeline strategy and responsibilities.

You learned how to describe the benefits and usage of:

* Describe Azure Pipelines.
* Explain the role of Azure Pipelines and its components.
* Decide Pipeline automation responsibility.
* Understand Azure Pipeline key terms.

**Learn more**

* [What is Azure Pipelines? - Azure Pipelines | Microsoft Learn](https://learn.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines).
* [Use Azure Pipelines - Azure Pipelines | Microsoft Learn](https://learn.microsoft.com/en-us/azure/devops/pipelines/get-started/pipelines-get-started).