**Development for enterprise DevOps**

**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**

**Introduction**

**Completed100 XP**

* **4 minutes**

**"DevOps is the union of people, process, and products to enable continuous delivery of value to our end users." - According to Donovan Brown in**[**What is DevOps?**](https://www.donovanbrown.com/post/what-is-devops)

**The DevOps learning paths will help you prepare for a DevOps journey. You'll learn the main characteristics of the DevOps process, tools, and people involved during the lifecycle. Also, it prepares you for the Microsoft DevOps Solution certification exam. You'll see other content to ensure you have a complete picture of DevOps. The module's content includes graphics, reference links, module review questions, and optional hands-on labs.**

**You'll learn the following:**

* **How to plan for DevOps.**
* **Use source control.**
* **Scale Git for an enterprise.**
* **Combine 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.**
* **Optimize feedback mechanisms.**

**Plan before you act. This module will help you understand what DevOps is and how to plan for a DevOps transformation journey.**

**What is the DevOps transformation journey?**

**The DevOps transformation journey is a series of 8 learning paths. It familiarizes you with Azure DevOps and GitHub. Also, learn its many services, features, and integration with tools to support your DevOps process.**

**Why should I take the DevOps learning path?**

**People in these modules are interested in designing and implementing DevOps processes. Also, they're preparing for the**[**AZ-400 - Designing and Implementing Microsoft DevOps Solutions**](https://learn.microsoft.com/en-us/learn/certifications/exams/az-400)**certification exam.**

**The certification exam is for DevOps professionals. Combine people, processes, and technologies to continuously deliver valuable products and services that meet end-user needs and business goals. DevOps professionals streamline delivery by optimizing practices, improving communications and collaboration, and creating automation.**

**They design and implement application code and infrastructure strategies that allow continuous integration, testing, delivery, monitoring, and feedback.**

**Exam candidates must be proficient with Agile practices. They must be familiar with Azure administration, development and experts in at least one of these areas.**

**DevOps professionals must design and implement DevOps practices for version control, compliance, infrastructure as code, configuration management, build, release, and testing-using Azure technologies.**

**There are five domain areas.**

**Expand table**

| **AZ-400 Domain Area** | **Weight** |
| --- | --- |
| **Design and implement processes and communications.** | **11%** |
| **Design and implement a source control strategy.** | **13%** |
| **Design and implement build and release pipelines.** | **51%** |
| **Develop a security and compliance plan.** | **16%** |
| **Implement an instrumentation strategy.** | **9%** |

**Learning objectives**

**After completing this series, students and professionals can:**

* **Design and implement traceability and flow of work.**
* **Design and implement appropriate metrics and queries for DevOps.**
* **Configure collaboration and communication.**
* **Design and implement branching strategies for the source code.**
* **Configure and manage repositories.**
* **Design and implement a package management strategy.**
* **Design and implement a testing strategy for pipelines.**
* **Design and implement pipelines.**
* **Design and implement deployments.**
* **Design and implement infrastructure as code (IaC).**
* **Maintain pipelines.**
* **Design and implement authentication and authorization methods.**
* **Design and implement a strategy for managing sensitive information in automation.**
* **Automate security and compliance scanning.**
* **Configure monitoring for a DevOps environment.**
* **Analyze metrics from instrumentation.**

**Prerequisites**

**Successful learners will have prior knowledge and understanding of the following:**

* **Cloud computing concepts include understanding PaaS, SaaS, and IaaS implementations.**
* **Azure administration and Azure development with proven expertise in at least one of these areas.**
* **Foundational DevOps concepts, including version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software.**

**If you're new to DevOps, consider taking the:**

* **Free online:**[**DevOps foundations: The core principles and practices**](https://go.microsoft.com/fwlink/?linkid=2268284)**- NEW.**
* **Instructor-led course: AZ-2008: DevOps foundations: The core principles and practices - NEW.**

**If you're new to Azure and cloud computing, consider one of the following resources:**

* **Free online:**[**Azure Fundamentals**](https://learn.microsoft.com/en-us/learn/paths/az-900-describe-cloud-concepts/)**.**
* **Instructor-led course:**[**AZ-900: Azure Fundamentals**](https://learn.microsoft.com/en-us/learn/certifications/courses/az-900t01)**.**

**If you're new to Azure Administration, consider taking the:**

* **Free online:**[**Prerequisites for Azure Administrators**](https://learn.microsoft.com/en-us/learn/paths/az-104-administrator-prerequisites/)**.**
* **Instructor-led courses:**[**AZ-104: Microsoft Azure Administrator**](https://learn.microsoft.com/en-us/learn/certifications/courses/az-104t00)**.**

**If you're new to Azure Developer, consider taking the:**

* **Free online:**[**Create serverless applications**](https://learn.microsoft.com/en-us/learn/paths/create-serverless-applications/)**.**
* **Instructor-led courses:**[**AZ-204: Developing Solutions for Microsoft Azure**](https://learn.microsoft.com/en-us/learn/certifications/courses/az-204t00)**.**

**You must create an Azure DevOps Organization for some exercises. If you don't have it yet, see the following:**

* [**Create an organization - Azure DevOps**](https://learn.microsoft.com/en-us/azure/devops/organizations/accounts/create-organization)**.**

**You must create a GitHub account at GitHub.com for some exercises. If you don't have it yet, see the following:**

* [**Join GitHub · GitHub**](https://github.com/signup)
* **If you already have your GitHub account, create a new repository**[**Creating a new repository - GitHub Docs**](https://docs.github.com/repositories/creating-and-managing-repositories/creating-a-new-repository)**.**

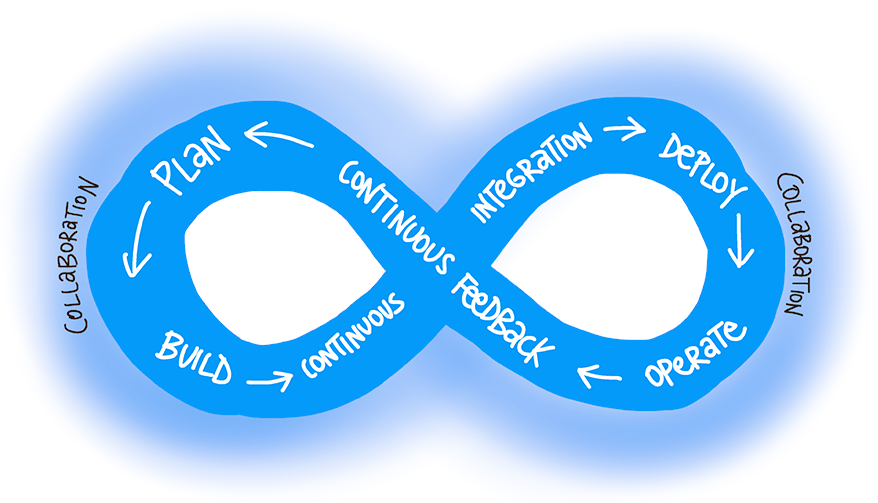
**Next unit: What is DevOps?**

**What is DevOps?**

**Completed100 XP**

* **3 minutes**

**The contraction of "Dev" and "Ops" refers to replacing siloed Development and Operations. The idea is to create multidisciplinary teams that now work together with shared and efficient practices and tools. Essential DevOps practices include agile planning, continuous integration, continuous delivery, and monitoring of applications. DevOps is a constant journey.**

****

**Understand your cycle time**

**Let us start with a basic assumption about software development. We will describe it with the OODA (Observe, Orient, Decide, Act) loop. Originally designed to keep fighter pilots from being shot out of the sky, the OODA loop is an excellent way to think about staying ahead of your competitors. You start with observing business, market, needs, current user behavior, and available telemetry data. Then you orient with the enumeration of options for what you can deliver, perhaps with experiments. Next, you decide what to pursue, and you act by delivering working software to real users. You can see all occurring in some cycle time.**

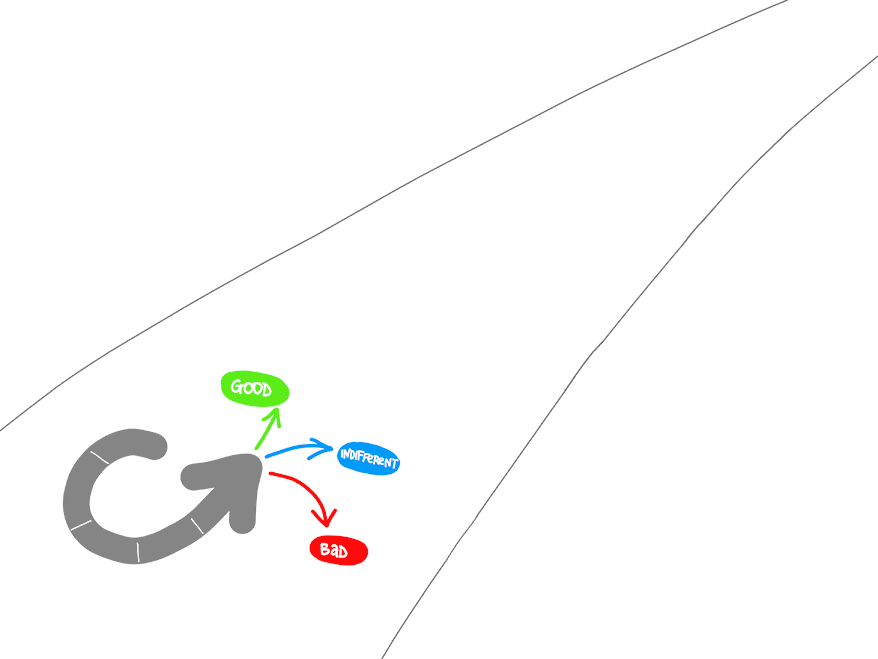
****

**Become data-informed**

**We recommend you use data to inform what to do in your next cycle. Many experience reports tell us that roughly one-third of the deployments will have negative business results. Approximately one-third will have positive results, and one-third will make no difference. Fail fast on effects that do not advance the business and double down on outcomes that support the business. Sometimes the approach is called pivot or persevere.**

**Strive for validated learning**

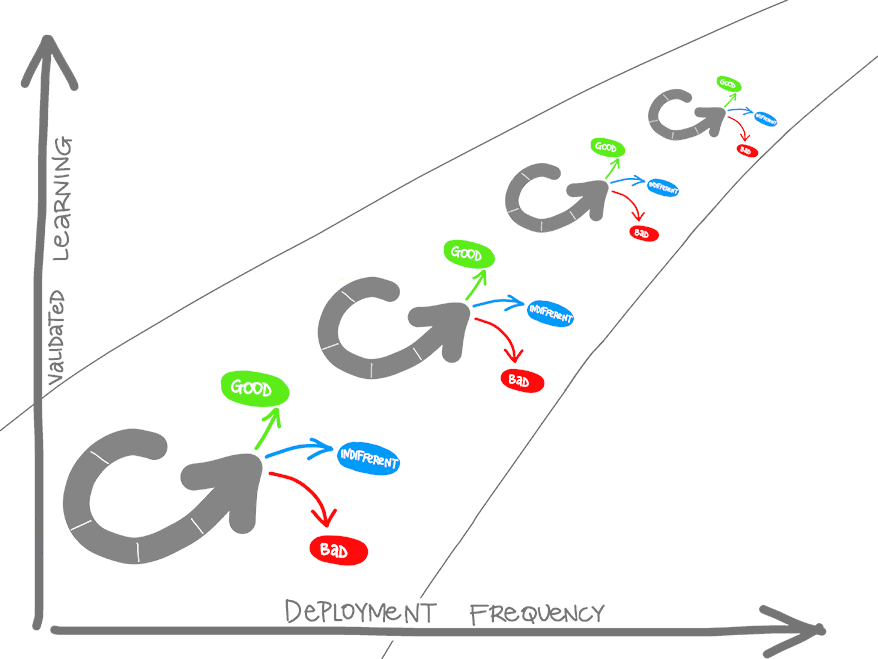
**How quickly you can fail fast or double down is determined by your cycle time. Also, in how long that loop takes, or in lean terms. Your cycle time determines how quickly you can gather feedback to determine what happens in the next loop. The feedback that you collect with each cycle should be factual, actionable data. We call it validated learning.**

****

**Shorten your cycle time**

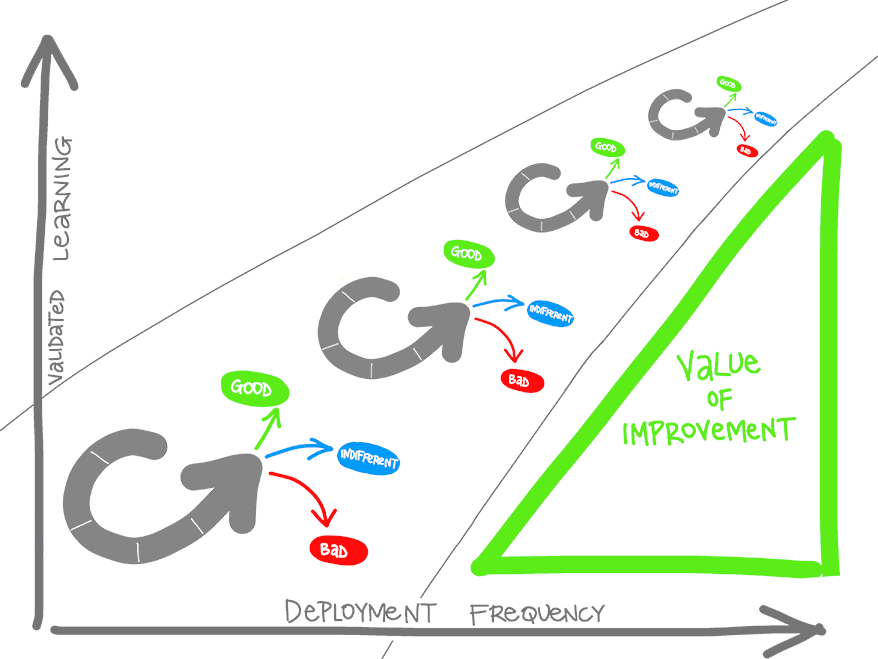
**When you adopt DevOps practices:**

* **You shorten your cycle time by working in smaller batches.**
* **Using more automation.**
* **Hardening your release pipeline.**
* **Improving your telemetry.**
* **Deploying more frequently.**

****

**Optimize validated learning**

**The more frequently you deploy, the more you can experiment. The more opportunity you have to pivot or persevere and gain validated learning each cycle. This acceleration in validated learning is the value of the improvement. Think of it as the sum of progress that you achieve and the failures that you avoid.**

****

**Next unit: Explore the DevOps journey**

[**Previous**](https://learn.microsoft.com/en-us/training/modules/introduction-to-devops/1-introduction/)

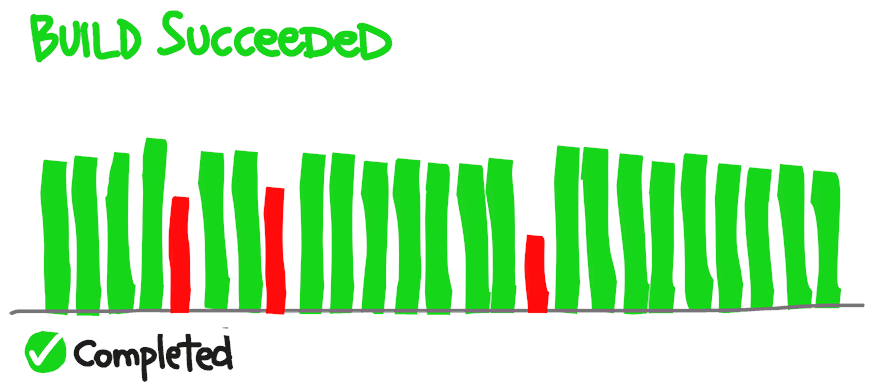
**Explore the DevOps journey**

**Completed100 XP**

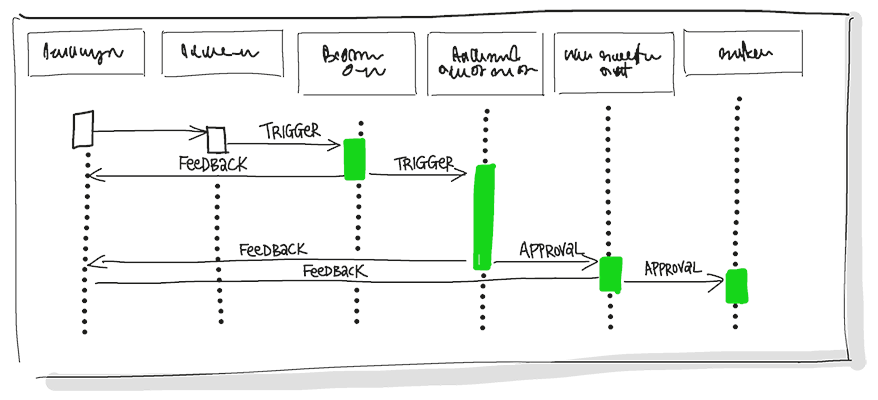
* **4 minutes**

**Remember, the goal is to shorten cycle time. Start with the release pipeline. How long does it take to deploy a change of one line of code or configuration? Ultimately, that is the brake on your velocity.**

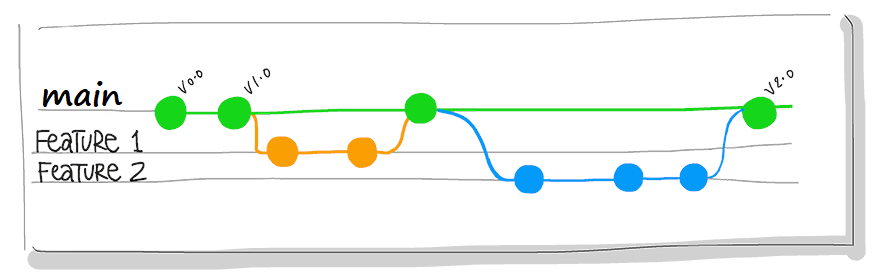
* **Continuous Integration drives the ongoing merging and testing of code, leading to an early finding of defects. Other benefits include less time wasted fighting merge issues and rapid feedback for development teams.**

****

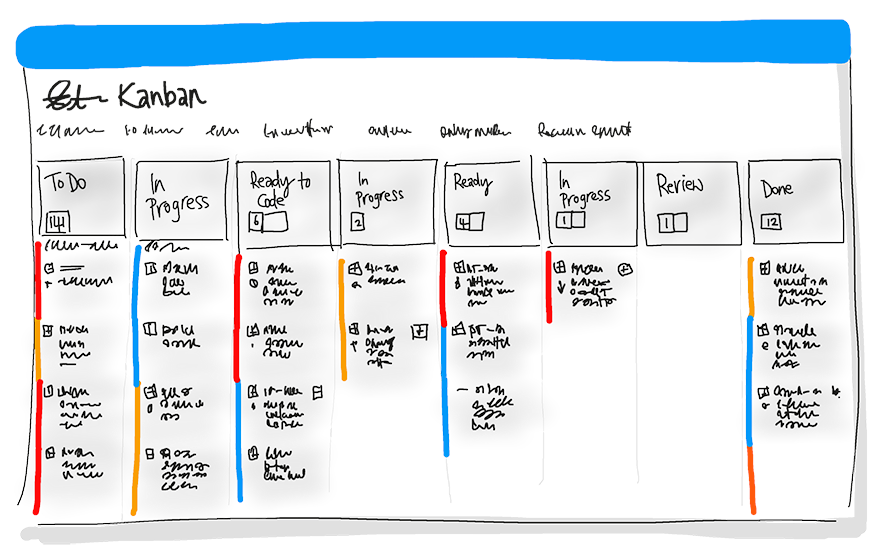
* **Continuous Delivery of software solutions to production and testing environments helps organizations quickly fix bugs and respond to ever-changing business requirements.**

****

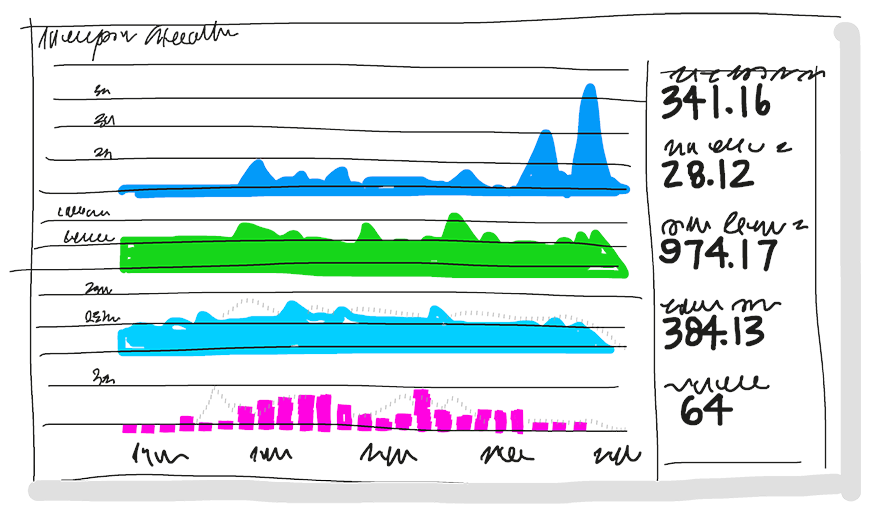
* **Version Control, usually with a Git-based Repository, enables teams worldwide to communicate effectively during daily development activities. Also, integrate with software development tools for monitoring activities such as deployments.**

****

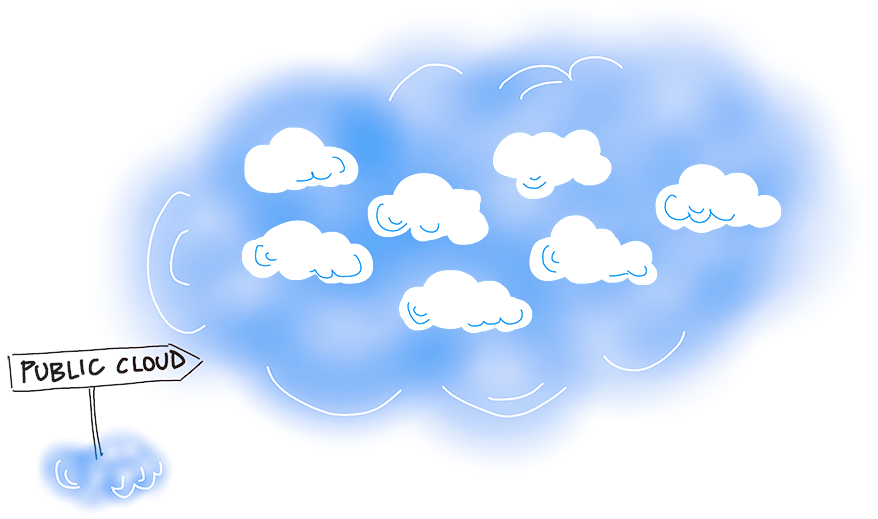
* **Use Agile planning and lean project management techniques to:**
  + **Plan and isolate work into sprints.**
  + **Manage team capacity and help teams quickly adapt to changing business needs.**
  + **A DevOps Definition of Done is working software collecting telemetry against the intended business goals.**

****

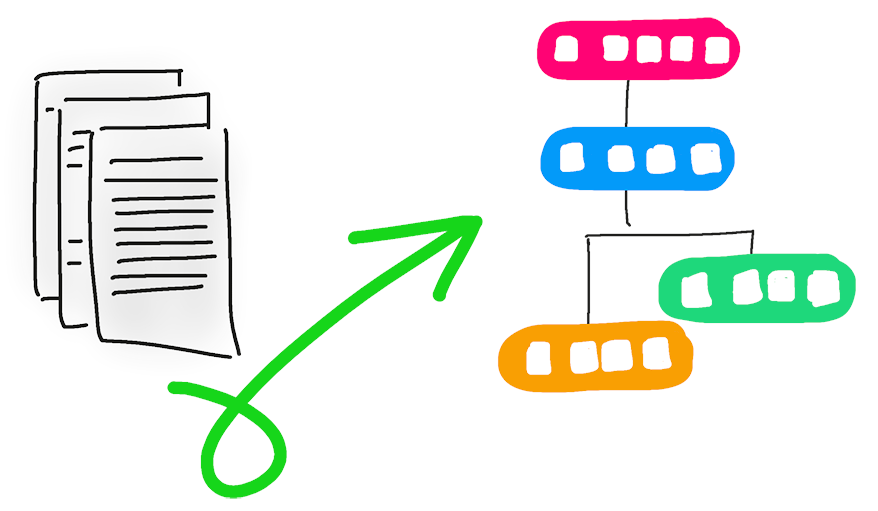
* **Monitoring and Logging of running applications. Including production environments for application health and customer usage. It helps organizations create a hypothesis and quickly validate or disprove strategies. Rich data is captured and stored in various logging formats.**

****

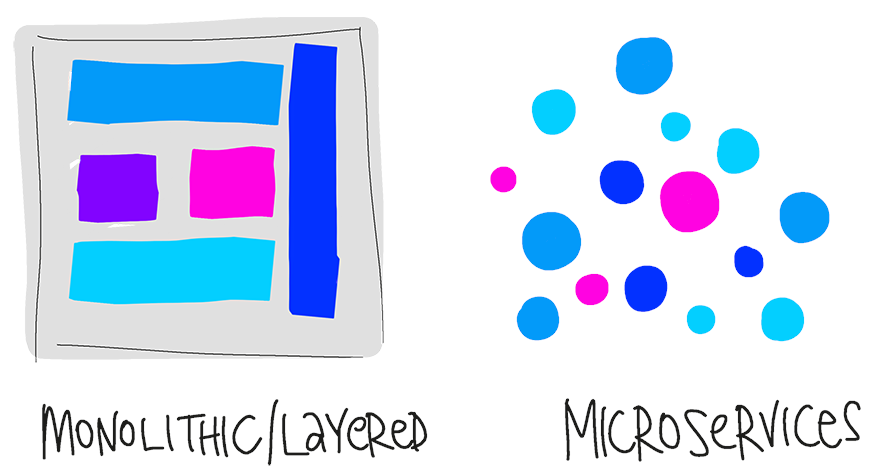
* **Public and Hybrid Clouds have made the impossible easy. The cloud has removed traditional bottlenecks and helped commoditize Infrastructure. You can use Infrastructure as a Service (IaaS) to lift and shift your existing apps or Platform as a Service (PaaS) to gain unprecedented productivity. The cloud gives you a data center without limits.**

****

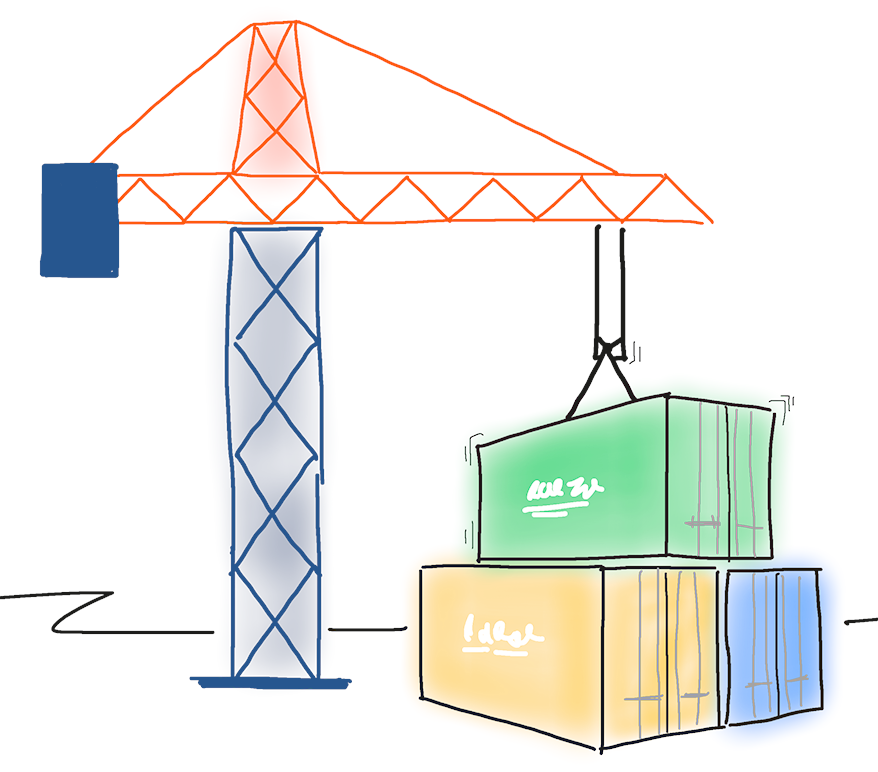
* **Infrastructure as Code (IaC): Enables the automation and validation of the creation and teardown of environments to help deliver secure and stable application hosting platforms.**

****

* **Use Microservices architecture to isolate business use cases into small reusable services that communicate via interface contracts. This architecture enables scalability and efficiency.**

****

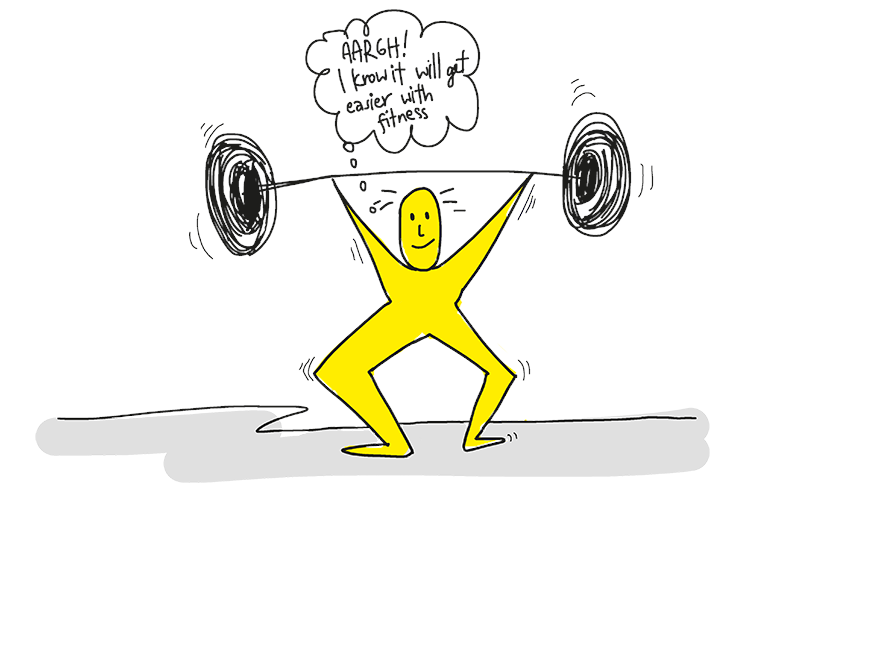
* **Containers are the next evolution in virtualization. They're much more lightweight than virtual machines, allow much faster hydration, and easily configure files.**

****

**DevOps may hurt at first.**

**If it hurts, do it more often. Adopting new practices like going to the gym is likely to hurt first. The more you exercise the new techniques, the easier they'll become.**

**Like training at the gym, where you first exercise large muscles before small muscles, adopt practices that have the most significant impact. Cross-train to develop synergy.**

****

**Note**

**The source article**[**defines DevOps**](https://learn.microsoft.com/en-us/azure/devops/learn/what-is-devops)**.**

**Next unit: Identify transformation teams**

**Identify transformation teams**

**Completed100 XP**

* **2 minutes**

**Unless you're building an entirely new organization, one of the significant challenges of any DevOps Transformation Project is dealing with actions that conflict in some way with ongoing business states.**

**The first challenge is the availability of staff. If the staff members leading the transformation project are also involved in existing day-to-day work within the organization, it will be challenging for them to focus on the transformation when their current role directly impacts customer outcomes. We all know that desperate situations involving customers will always win over a long-term project like DevOps transformations.**

**Another issue is how the organization operates. Implementing existing processes and procedures to support current business outcomes can make it difficult to disrupt the status quo required for a true DevOps Transformation.**

**In the book "Beyond the Idea: How to Execute Innovation," Dr. Vijay Govindarajan and Dr. Chris Trimble noted that successful innovation often occurs despite existing organizational processes. They concluded that it only works by creating a separate team to pursue the transformation.**

**For DevOps transformations, the separate team should be composed of staff members focused on and measured by the transformation outcomes, not involved in the day-to-day operational work. The team might also include external experts who can fill knowledge gaps and advise on processes that are new to existing staff members. Ideally, the staff members recruited should already be well-regarded throughout the organization and offer a broad knowledge base to think outside the box as a group.**

**Next unit: Define organization structure for agile practices**

**Define organization structure for agile practices**

**Completed100 XP**

* **3 minutes**

**For most organizations, reorganizing to be agile is difficult. It requires a mind-shift and a culture-shift that challenges many existing policies and processes within the organization.**

**Good governance in organizations, particularly in large organizations, often leads to many relatively rigid rules, operating structures, and methods. It also tends to avoid a broad delegation of authority.**

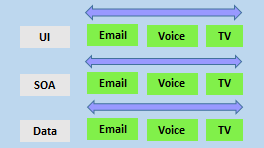
**While most large organizations haven't moved to an agile structure, most are now experimenting with doing so.**

**Their business environments are volatile and complex, and they have seen the limitations of their current systems, mainly an inability to cope with change quickly.**

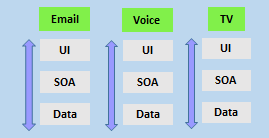
**They realize that it's common today for long-term established businesses and their industries to be disrupted by startups.**

**Horizontal vs. vertical teams**

**Traditionally, horizontal team structures divide teams according to the software architecture. In this example, the teams have been divided into the user interface, service-oriented architecture, and data teams:**

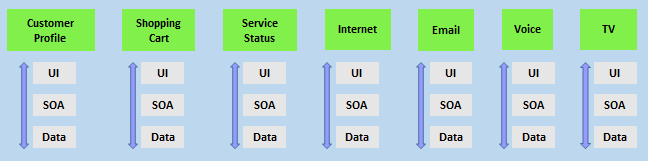
****

**By comparison, vertical team structures span the architecture and are aligned with skillsets or disciplines:**

****

**Vertical teams have been shown to provide more good outcomes in Agile projects. Each product must have an identified owner.**

**Another key benefit of the vertical team structure is that scaling can occur by adding teams. In this example, feature teams have been created rather than just project teams:**

****

**Next unit: Explore shared goals and define timelines**

**Explore shared goals and define timelines**

**Completed100 XP**

* **2 minutes**

**Explore shared goals**

**These outcomes should include specific, measurable targets like:**

* **Reduce the time spent on fixing bugs by 60%.**
* **Reduce the time spent on unplanned work by 70%.**
* **Reduce the out-of-hours work required by staff to no more than 10% of total working time.**
* **Remove all direct patching of production systems.**

**Note**

**One of the aims of DevOps is to provide more excellent customer value, so outcomes should have a customer value focus.**

**Define timelines for goals**

**Measurable goals also need to have timelines. While it is easy to set longer-term goals, it is also easy to put off work when you do not require it for a while.**

**It is essential to have an ongoing series of short-term goals. Overall, projects should have timelines that span anywhere from a few months to a year or two in any DevOps transformation project.**

**Every few weeks, the improvements should be clear and measurable. Ideally, evident to the organization or its customers.**

**The timeline should not be too short and should always be challenging yet achievable. A review should occur after each short-term goal to help plan the next one.**

**There are several advantages of the shorter timelines:**

* **It is easier to change plans or priorities when necessary.**
* **The reduced delay between doing work and getting feedback helps ensure that the learnings and feedback are incorporated quickly.**
* **It is easier to keep organizational support when positive outcomes are clear.**

**Next unit: What is Azure DevOps?**

**What is Azure DevOps?**

**Completed100 XP**

* **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.**

**Next unit: What is GitHub?**

[**Previous**](https://learn.microsoft.com/en-us/training/modules/introduction-to-devops/6-explore-shared-goals-define-timelines/)

**What is GitHub?**

**Completed100 XP**

* **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.**

**Next unit: Design a license management strategy**

**Design a license management strategy**

**Completed100 XP**

* **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**](https://azure.microsoft.com/pricing/details/devops/azure-devops-services/)**.**

**For the latest, most up-to-date GitHub pricing information, visit**[**GitHub Pricing**](https://github.com/pricing/)**.**

**Next unit: What is source control?**

**What is source control?**

**Completed100 XP**

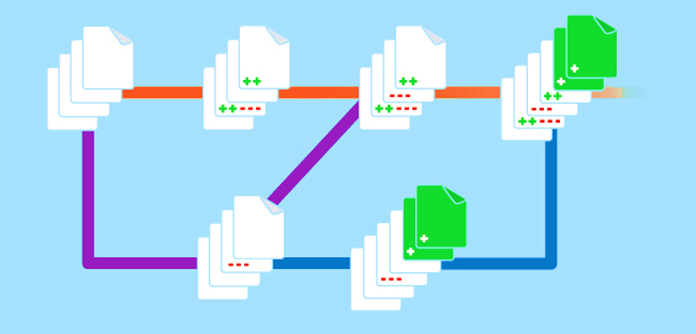
* **2 minutes**

**A Source control system (or version control system) allows developers to collaborate on code and track changes. Use version control to save your work and coordinate code changes across your team. Source control is an essential tool for multi-developer projects.**

**The version control system saves a snapshot of your files (history) so that you can review and even roll back to any version of your code with ease. Also, it helps to resolve conflicts when merging contributions from multiple sources.**

**For most software teams, the source code is a repository of invaluable knowledge and understanding about the problem domain that the developers have collected and refined through careful effort.**

**Source control protects source code from catastrophe and the casual degradation of human error and unintended consequences.**

****

**Without version control, you're tempted to keep multiple copies of code on your computer. It could be dangerous. Easy to change or delete a file in the wrong code copy, potentially losing work.**

**Version control systems solve this problem by managing all versions of your code but presenting you with a single version at a time.**

**Tools and processes alone aren't enough to accomplish the above, such as adopting Agile, Continuous Integration, and DevOps. Believe it or not, all rely on a solid version control practice.**

**Version control is about keeping track of every change to software assets—tracking and managing the who, what, and when. Version control is the first step needed to assure quality at the source, ensure flow and pull value, and focus on the process. All of these create value not just for the software teams but ultimately for the customer.**

**Version control is a solution for managing and saving changes made to any manually created assets. If changes are made to the source code, you can go back in time and easily roll back to previous-working versions.**

**Version control tools will enable you to see who made changes, when, and what exactly was changed.**

**Version control also makes experimenting easy and, most importantly, makes collaboration possible. Without version control, collaborating over source code would be a painful operation.**

**There are several perspectives on version control.**

* **For developers, it's a daily enabler for work and collaboration to happen. It's part of the daily job, one of the most-used tools.**
* **For management, the critical value of version control is in:**
  + **IP security.**
  + **Risk management.**
  + **Time-to-market speed through Continuous Delivery, where version control is a fundamental enabler.**

**Next unit: Describe working with Git locally**

**Describe working with Git locally**

**Completed100 XP**

* **7 minutes**

**Git and Continuous Delivery is one of those delicious chocolate and peanut butter combinations. We occasionally find two great tastes that taste great together in the software world!**

**Continuous Delivery of software demands a significant level of automation. It's hard to deliver continuously if you don't have a quality codebase.**

**Git provides you with the building blocks to take charge of quality in your codebase. It allows you to automate most of the checks in your codebase. Also, it works before committing the code into your repository.**

**To fully appreciate the effectiveness of Git, you must first understand how to carry out basic operations on Git. For example, clone, commit, push, and pull.**

**The natural question is, how do we get started with Git?**

**One option is to go native with the command line or look for a code editor that supports Git natively.**

**Visual Studio Code is a cross-platform, open-source code editor that provides powerful developer tooling for hundreds of languages.**

**To work in open-source, you need to embrace open-source tools.**

**This recipe starts by:**

* **Setting up the development environment with Visual Studio Code.**
* **Creating a new Git repository.**
* **Committing code changes locally.**
* **Pushing changes to a remote repository on Azure DevOps.**

**Getting ready**

**This tutorial teaches us how to initialize a Git repository locally.**

**Then we use the ASP.NET Core MVC project template to create a new project and version it in the local Git repository.**

**We'll then use Visual Studio Code to interact with the Git repository to do basic commit, pull, and push operations.**

**You need to set up your working environment with the following:**

* **.NET 8 SDK or later:**[**Download .NET**](https://dotnet.microsoft.com/download)**.**
* **Visual Studio Code:**[**Download Visual Studio Code**](https://code.visualstudio.com/Download)**.**
* **C# Visual Studio Code extension:**[**C# programming with Visual Studio Code**](https://code.visualstudio.com/docs/languages/csharp)**.**
* **Git:**[**Git - Downloads**](https://git-scm.com/downloads)
* **Git for Windows (if you're using Windows):**[**Git for Windows**](https://gitforwindows.org/)

**The Visual Studio Marketplace features several extensions for Visual Studio Code that you can install to enhance your experience of using Git:**

* [**Git Lens**](https://gitlens.amod.io/)**: This extension brings visualization for code history by using Git blame annotations and code lens. The extension enables you to seamlessly navigate and explore the history of a file or branch. Also, the extension allows you to gain valuable insights via powerful comparison commands and much more.**
* [**Git History**](https://github.com/DonJayamanne/gitHistoryVSCode/blob/master/README.md)**: Brings visualization and interaction capabilities to view the Git log, file history and compare branches or commits.**

**How to do it**

1. **Open the Command Prompt and create a new-working folder:**

**CmdCopy**

**mkdir myWebApp**

**cd myWebApp**

1. **In myWebApp, initialize a new Git repository:**

**CmdCopy**

**git init**

1. **Configure global settings for the name and email address to be used when committing in this Git repository:**

**CmdCopy**

**git config --global user.name "John Doe"**

**git config --global user.email "john.doe@contoso.com"**

**If you're working behind an enterprise proxy, you can make your Git repository proxy-aware by adding the proxy details in the Git global configuration file. Different variations of this command allow you to set up an HTTP/HTTPS proxy (with username/password) and optionally bypass SSL verification. Run the below command to configure a proxy in your global git config.**

**CmdCopy**

**git config --global http.proxy**

**http://proxyUsername:proxyPassword@proxy.server.com:port**

1. **Create a new ASP.NET core application. The new command offers a collection of switches that can be used for language, authentication, and framework selection. More details can be found on**[**Microsoft Learn**](https://learn.microsoft.com/en-us/dotnet/core/tools/dotnet-new)**.**

**CmdCopy**

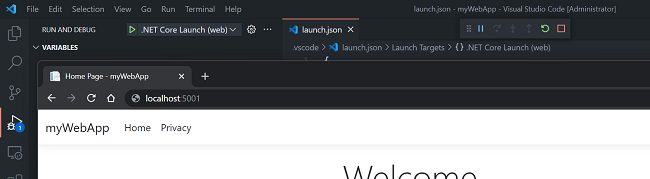
**dotnet new mvc**

**Launch Visual Studio Code in the context of the current-working folder:**

**CmdCopy**

**code .**

1. **When the project opens in Visual Studio Code, select Yes for the Required assets to build and debug are missing from 'myWebApp.' Add them? Warning message. Select Restore for the There are unresolved dependencies info message. Hit F5 to debug the application, then myWebApp will load in the browser, as shown in the following screenshot:**

****

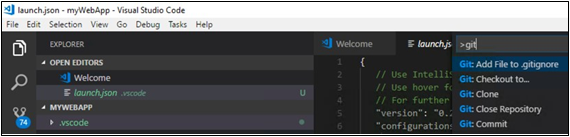
**If you prefer to use the command line, you can run the following commands in the context of the git repository to run the web application.**

**CmdCopy**

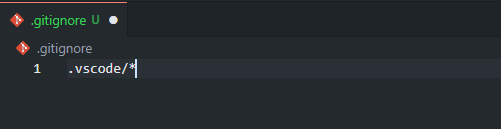
**dotnet build**

**dotnet run**

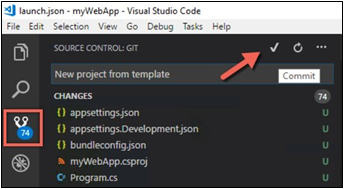
**You notice the ".VS Code" folder is added to your working folder. To avoid committing this folder to your Git repository, you can include it in the .gitignore file. Select a file from the ".VS Code" folder, hit F1 to launch the command window in Visual Studio Code, type gitIgnore, and accept the option to include the selected file in the new. gitIgnore file.**

****

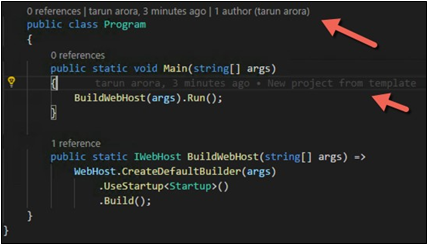
**Open your .gitignore, remove the file name from the path, and leave the folder with a slash, for example, *.VS Code/\**.**

****

1. **To stage and commit the newly created myWebApp project to your Git repository from Visual Studio Code, navigate the Git icon from the left panel. Add a commit comment and commit the changes by clicking the checkmark icon. It will stage and commit the changes in one operation:**

****

**Open Program.cs, you notice Git lens decorates the classes and functions with the commit history and brings this information in line to every line of code:**

****

1. **Now launch cmd in the context of the git repository and run git branch --list. It shows you that only the main branch currently exists in this repository. Now run the following command to create a new branch called feature-devops-home-page.**

**CmdCopy**

**git branch feature-devops-home-page**

**git checkout feature-devops-home-page**

**git branch --list**

**You have created a new branch with these commands and checked it out. The --list keyword shows you a list of all branches in your repository. The green color represents the branch that is currently checked out.**

1. **Now navigate to the file ~\Views\Home\Index.cshtml and replace the contents with the text below.**

**C#Copy**

**@{**

**ViewData["Title"] = "Home Page";**

**}**

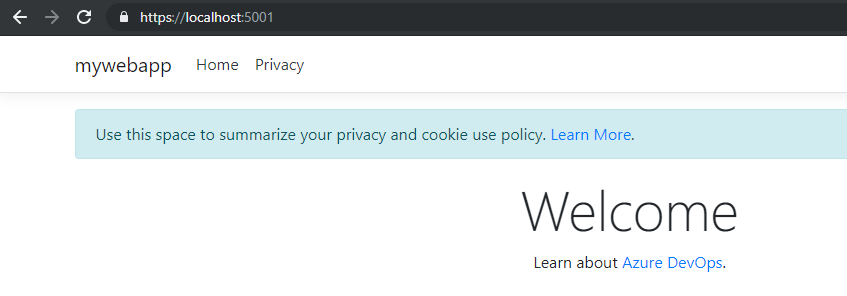
**<div class="text-center">**

**<h1 class="display-4">Welcome</h1>**

**<p>Learn about <a href="https://azure.microsoft.com/services/devops/">Azure DevOps</a>.</p>**

**</div>**

1. **Refresh the web app in the browser to see the changes.**

****

1. **In the context of the git repository, execute the following commands. These commands stage the changes in the branch and then commit them.**

**CmdCopy**

**git status**

**git add .**

**git commit -m "updated welcome page."**

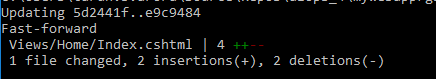
**git status**

1. **To merge the changes from the feature-devops-home-page into the main, run the following commands in the context of the git repository.**

**CmdCopy**

**git checkout main**

**git merge feature-devops-home-page**

****

1. **Run the below command to delete the feature branch.**

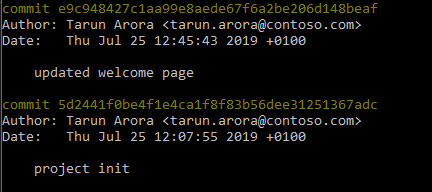
**CmdCopy**

**git branch --delete feature-devops-home-page**

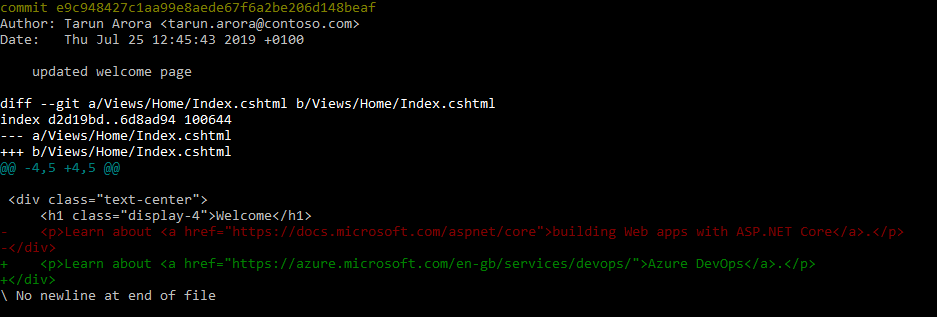
**How it works**

**The easiest way to understand the outcome of the steps done earlier is to check the history of the operation. Let us have a look at how to do it.**

1. **In Git, committing changes to a repository is a two-step process. Running: add . The changes are staged but not committed. Finally, running the commit promotes the staged changes in the repository.**
2. **To see the history of changes in the main branch, run the command git log -v**

****

1. **To investigate the actual changes in the commit, you can run the command git log -p**

****

**There is more**

**Git makes it easy to back out changes. Following our example, if you want to take out the changes made to the welcome page.**

**You can do It hard resetting the main branch to a previous version of the commit using the following command.**

**CmdCopy**

**git reset --hard 5d2441f0be4f1e4ca1f8f83b56dee31251367adc**

**Running the above command would reset the branch to the project init change.**

**If you run git log -v, you see that the changes done to the welcome page are removed from the repository.**

**Next unit: Introduction to Azure Repos**

**Introduction to Azure Repos**

**Completed100 XP**

* **2 minutes**

**Azure Repos is a set of version control tools that you can use to manage your code.**

**Using version control is a good idea whether your software project is large or small.**

**Azure Repos provides two types of version control:**

* **Git: distributed version control**
* **Team Foundation Version Control (TFVC): centralized version control**

**What do I get with Azure Repos?**

* **Use free private Git repositories, pull requests, and code search: Get unlimited private Git repository hosting and support for TFVC that scales from a hobby project to the world’s largest repository.**
* **Support for any Git client: Securely connect with and push code into your Git repository from any IDE, editor, or Git client.**
* **Web hooks and API integration: Add validations and extensions from the marketplace or build your own-using web hooks and REST APIs.**
* **Semantic code search: Quickly find what you are looking for with a code-aware search that understands classes and variables.**
* **Collab to build better code: Do more effective Git code reviews with threaded discussion and continuous integration for each change. Use forks to promote collaboration with inner source workflows.**
* **Automation with built-in CI/CD: Set up continuous integration/continuous delivery (CI/CD) to automatically trigger builds, tests, and deployments. Including every completed pull request using Azure Pipelines or your tools.**
* **Protection of your code quality with branch policies: Keep code quality high by requiring code reviewer sign-out, successful builds, and passing tests before merging pull requests. Customize your branch policies to maintain your team’s high standards.**
* **Usage of your favorite tools: Use Git and TFVC repositories on Azure Repos with your favorite editor and IDE.**

**For further reference on using git in Azure Repos, refer to**[**Microsoft Learn**](https://learn.microsoft.com/en-us/azure/devops/repos)**.**

**Next unit: Introduction to GitHub**

**Introduction to GitHub**

**Completed100 XP**

* **3 minutes**

**GitHub is the largest open-source community in the world. Microsoft owns GitHub. GitHub is a development platform inspired by the way you work.**

**You can host and review code, manage projects, and build software alongside 40 million developers from open source to business.**

**GitHub is a Git repository hosting service that adds many of its features.**

**While Git is a command-line tool, GitHub provides a Web-based graphical interface.**

**It also provides access control and several collaboration features, such as wikis and essential task management tools for every project.**

**So what are the main benefits of using GitHub? Nearly every open-source project uses GitHub to manage its project.**

**Using GitHub is free if your project is open source and includes a wiki and issue tracker, making it easy to have more in-depth documentation and get feedback about your project.**

**What are some of the features offered by GitHub?**

* **Automate from code to cloud: Cycle your production code faster and simplify your workflow with GitHub Packages and built-in CI/CD using GitHub Actions.**
  + **Automate your workflows: Build, test, deploy, and run CI/CD how you want in the same place you manage code. Trigger Actions from any GitHub event to any available API. Build your Actions in the language of your choice, or choose from thousands of workflows and Actions created by the community.**
  + **Packages at home with their code: Use Actions to automatically publish new package versions to GitHub Packages. Install packages and images hosted on GitHub Packages or your preferred packages registry in your CI/CD workflows. It is always free for open source, and data transfer within Actions is unlimited for everyone.**
* **Securing software together: GitHub plays a role in securing the world's code—developers, maintainers, researchers, and security teams. On GitHub, development teams everywhere can work together to secure the world's software supply chain, from fork to finish.**
  + **Get alerts about vulnerabilities in your code: GitHub continuously scans security advisories for popular languages. Also, it sends security alerts to maintainers of affected repositories with details so they can remediate risks.**
  + **Automatically update vulnerabilities: GitHub monitors your project dependencies and automatically opens pull requests to update dependencies to the minimum version that resolves known vulnerabilities.**
  + **Stay on top of CVEs: Stay updated with the latest Common Vulnerabilities and Exposures (CVEs), and learn how they affect you with the GitHub Advisory Database.**
  + **Find vulnerabilities that other tools miss: CodeQL is the industry's leading semantic code analysis engine. GitHub's revolutionary approach treats code as data to identify security vulnerabilities faster.**
  + **Eliminate variants: Never make the same mistake twice. Proactive vulnerability scanning prevents vulnerabilities from ever reaching production.**
  + **Keep your tokens safe: Accidentally commit a token to a public repository? GitHub got you. With support from 20 service providers, GitHub takes steps to keep you safe.**
* **Seamless code review: Code review is the surest path to better code and is fundamental to how GitHub works. Built-in review tools make code review an essential part of your team's process.**
  + **Propose changes: Better code starts with a Pull Request, a living conversation about changes where you can talk through ideas, assign tasks, discuss details, and conduct reviews.**
  + **Request reviews: If you are on the other side of a review, you can request reviews from your peers to get the detailed feedback you need.**
  + **See the difference: Reviews happen faster when you know exactly what changes. Diffs compare versions of your source code, highlighting the new, edited, or deleted parts.**
  + **Comment in context: Discussions happen in comment threads within your code—bundle comments into one review or reply to someone else who is in line to start a conversation.**
  + **Give clear feedback: Your teammates should not have to think too hard about what a thumbs-up emoji means. Specify whether your comments are required changes or just a few suggestions.**
  + **Protect branches: Only merge the highest-quality code. You can configure repositories to require status checks, reducing human error and administrative overhead.**
* **All your code and documentation in one place: Hundreds of millions of private, public, and open-source repositories are hosted on GitHub. Every repository has tools to help your host, version, and release code and documentation.**
  + **Code where you collaborate: Repositories keep code in one place and help your teams collaborate with the tools they love, even if you work with large files using Git LFS. You can create or import as many projects as possible with unlimited private repositories for individuals and groups.**
  + **Documentation alongside your code: Host your documentation directly from your repositories with GitHub Pages. Use Jekyll as a static site generator and publish your Pages from the /docs folder on your main branch.**
* **Manage your ideas: Coordinate early, stay aligned, and get more done with GitHub's project management tools.**
  + **See your project's large picture: See everything happening in your project and choose where to focus your team's efforts with Projects and task boards that live right where they belong: close to your code.**
  + **Track and assign tasks: Issues help you identify, assign, and keep track of tasks within your team. You can open an Issue to track a bug, discuss an idea with an @mention, or start distributing work.**
* **The human side of software: Building software is more about managing teams and communities than coding. Whether on a group of two or 2000, GitHub has the support your people need.**
  + **Manage and grow teams: Help people organize with GitHub teams, level up to access administrative roles, and fine-tune your permissions with nested teams.**
  + **Keep conversations: Moderation tools, like issue and pull request locking, help your team stay focused on code. And if you maintain an open-source project, user blocking reduces noise and ensures productive conversations.**
  + **Set community guidelines: Set roles and expectations without starting from scratch. Customize standard codes of conduct to create the perfect one for your project. Then choose a pre-written license right from your repository.**

**GitHub offers excellent learning resources for its platform. You can find everything from git introduction training to deep dive on publishing static pages to GitHub and how to do DevOps on GitHub right**[**here**](https://skills.github.com/)**.**

**Next unit: Knowledge check**

**Knowledge check**

**Completed200 XP**

* **Module assessment**
* **4 minutes**

**Answer 100% of questions correctly in order to pass.**[**Retake**](https://learn.microsoft.com/en-us/training/modules/introduction-to-devops/14-knowledge-check)

**Dismiss alert**

**Choose the best response for each question.**

**Check your knowledge**

**Top of Form**

**1.**

**Which of the following choices best describes DevOps?**

**DevOps is the role of who manages source control, pipelines, and monitor environments to continue delivering value to the software project.**

**DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.**

**Correct. According to Donovan Brown, "DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.**

**DevOps is the new process of creating continuous delivery and continuous integration for software projects.**

**Incorrect. According to Donovan Brown, "DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.**

**2.**

**Which of the following choices drives the ongoing merging and testing of code that leads to finding defects early?**

**Continuous Integration.**

**Correct. Continuous Integration drives the ongoing merging and testing of code, which leads to finding defects early.**

**Continuous Delivery.**

**Incorrect. Continuous Integration drives the ongoing merging and testing of code, which leads to finding defects early.**

**Continuous Feedback.**

**3.**

**Which of the following choices is a practice that enables the automated creation of environments?**

**Infrastructure as a Service (IaaS).**

**Infrastructure as Code (IaC).**

**Correct. Infrastructure as Code (IaC) is a practice that enables the automation and validation of the creation and teardown of environments to help with delivering secure and stable application hosting platforms.**

**Software as a Service (SaaS).**

**Bottom of Form**

**Next unit: Summary**

[**Previous**](https://learn.microsoft.com/en-us/training/modules/introduction-to-devops/13-introduction-to-github/)

**Summary**

**Completed100 XP**

* **2 minutes**

**This module explored the key areas that organizations must apply to start their DevOps transformation Journey, change the team's mindset, and define timelines and goals.**

**You learned how to describe the benefits and usage of:**

* **Understand what DevOps is and the steps to accomplish it.**
* **Identify teams to implement the process.**
* **Plan for the transformation with shared goals and timelines.**
* **Plan and define timelines for goals.**

**Learn more**

* [**Donovan Brown | What is DevOps?**](https://www.donovanbrown.com/post/what-is-devops)
* [**What is DevOps? - Azure DevOps | Microsoft Learn**](https://learn.microsoft.com/en-us/devops/what-is-devops)
* [**Getting started with GitHub - GitHub Docs**](https://docs.github.com/get-started)
* [**View of features and epics on the Feature Timeline - Azure DevOps | Microsoft Learn**](https://learn.microsoft.com/en-us/azure/devops/boards/extensions/feature-timeline)
* [**Plan and track work in Azure Boards with Basic or Agile processes - Azure Boards | Microsoft Learn**](https://learn.microsoft.com/en-us/azure/devops/boards/get-started/plan-track-work)
* [**Agile Manifesto for Software Development | Agile Alliance**](https://www.agilealliance.org/agile101/the-agile-manifesto)
* [**12 Principles Behind the Agile Manifesto | Agile Alliance**](https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto)

**Knowledge check**

Completed200 XP

* **Module assessment**
* 3 minutes

 Great job! You passed the module assessment.

Dismiss alert

Top of Form

**1.**

**You're considering whether to add Microsoft Purview to your Microsoft Fabric environment for its extra governance features. Which of the following features is only available in Purview?**

Data Loss Prevention (DLP)

**Correct. DLP policies aren't available in Microsoft Fabric until you add Purview to your subscriptions.**

Sensitivity tags

Capacities

**2.**

**You open the Microsoft Purview hub in Microsoft Fabric but don't see a link to the Microsoft Purview Audit tool. What should you do?**

Sign out and sign back in as a Fabric administrator.

**Correct. Only Fabric administrators see a link to the Microsoft Purview Audit tool.**

Add a Microsoft Purview subscription.

Select the Open full report button.

**3.**

**You're setting up a connection to Microsoft Fabric from Microsoft Purview in a different tenant. Which of these security entities can you use for authentication?**

An Outlook.com account

A managed identity

A service principal

**Correct. You can only use a service principal or delegated authentication when Purview and Fabric are in different tenants.**

Bottom of Form

**Next unit: Summary**