edureka!

Microsoft Azure DevOps Solution Certification (AZ-400)

COURSE OUTLINE

Azure AZ-400

MODULE 1: Introduction to Azure DevOps

MODULE 2: Implementing Continuous Integration

MODULE 3: Build Containers with Azure DevOps

MODULE 4: Designing a Dependency Management Strategy and Managing Artifact Versioning

MODULE 5: Setting up Release Management Workflow

MODULE 6: Implementing Deployment Models and Services

MODULE 7: Implement and Optimize Continuous Feedback Mechanism

MODULE 8: Azure Tools: Infrastructure and Configuration, and Third-Party Tools

MODULE 9: Implementing Compliance and Security

MODULE 10: Azure Case Studies

edureka!

Introduction to Azure DevOps - DevOps Planning, Source Control, and Git

Topics

Following are the topics covered in this module:

- Introduction to Azure
- Introduction to DevOps
- Introduction to Azure DevOps
- Introduction to Transformation Planning
- Introduction to Source Control
- Migrating to Azure DevOps
- Git Authentication in Azure Repos

Objectives

After completing this module, you should be able to:

- Describe Azure and its benefits
- Understand DevOps and its lifecycle
- Describe Azure DevOps and identify the services of Azure DevOps
- Understand the role of transformation planning, project selection, and team structure in DevOps
- Explain source control and determine its types
- Migrate to Azure DevOps
- Demonstrate Git Version Controlling
- Demonstrate code review with the pull request

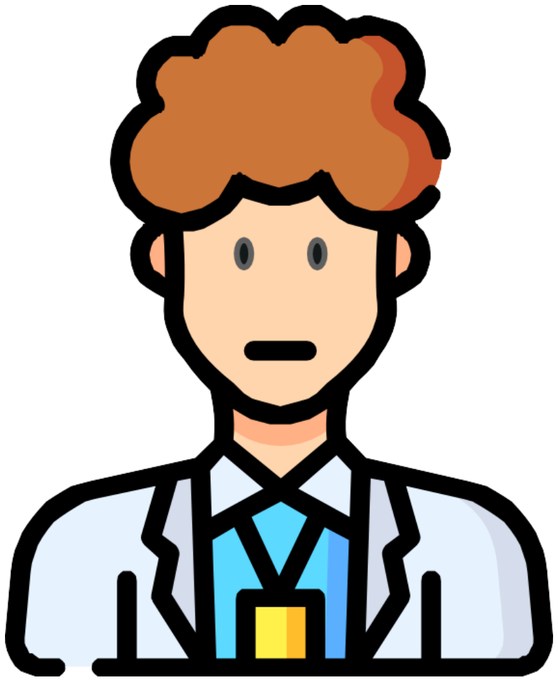
RedHat’s New Endeavour

edureka!

RedHat

RedHat is a multinational software corporation that provides open-source solutions to enterprises. It and has employees working from all over the world on its various software products

RedHat’s New Endeavor



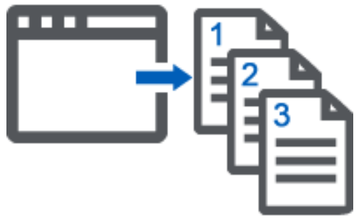
Tim-
The Lead

THE NEW FRAMEWORK

RedHat is planning to launch a new open-source framework to manage salesforce workloads. They have hired Tim as a team leader whose initial job is to **plan** the project and work with a distributed team of developers

The Issue

Tim is facing two issues with starting the project



PLANNING

- Managing timeline for the entire project
- Planning, managing, and monitoring tasks for multiple teams

CODE MANAGEMENT

- Tracking and managing changes made in code by 40+ developers
- Managing collaboration between developers working from various remote locations

The Research

Tim
The Lead

From my research, it is clear that I'll need a cloud-based planning tool that can track tasks for all the team members and a Distributed Version Control tool that can help me manage the code from developers

It seems Azure Boards and Git exactly fit my requirement criteria.

The Solution

AZURE BOARDS

- Azure Boards enables users to manage the entire software project from a single place
- It can be used to manage various teams associated with the project
- It can also trace user stories, backlog items, tasks, features, and bugs associated with the project

GIT

- Git is a distributed SCM tool that enables code tracking and easy collaboration amongst developers
- The branching and merging mechanism enables different users to work on the same feature concurrently

Introduction to Azure

edureka!

What is Cloud Computing?

Cloud computing is defined as the delivery of computing services over the internet using a pay-as-you-go pricing model.
It helps the user or organization build, manage, and deploy applications on a massive global network using various tools and frameworks.

edureka!

What is Azure?

Azure is Microsoft's cloud computing platform. It is a set of cloud services that help the organization meet the current and future business challenges.

Benefits of Azure

Azure provides:

- A limitless pool of raw compute, storage, and networking components
- AI and Machine Learning services like speech recognition and cognitive services to design and develop various applications
- Analytics and IoT services that enable you to make sense of telemetry data coming from your software and devices

Azure Services

Compute

Networking

Storage

Mobile

Database

Web

Internet-of-Things

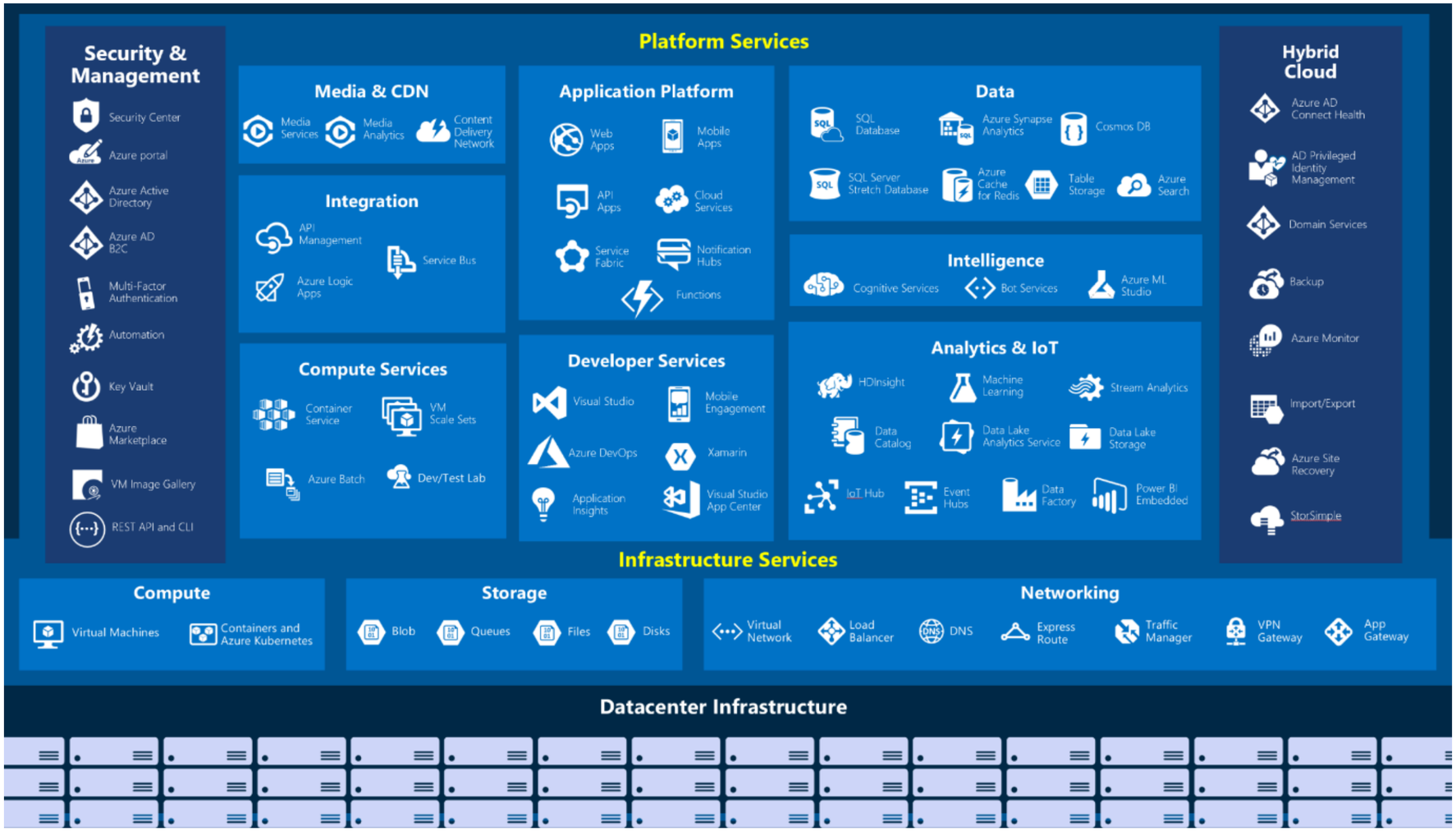
Big Data

Artificial Intelligence

DevOps

edureka!

Azure Offerings



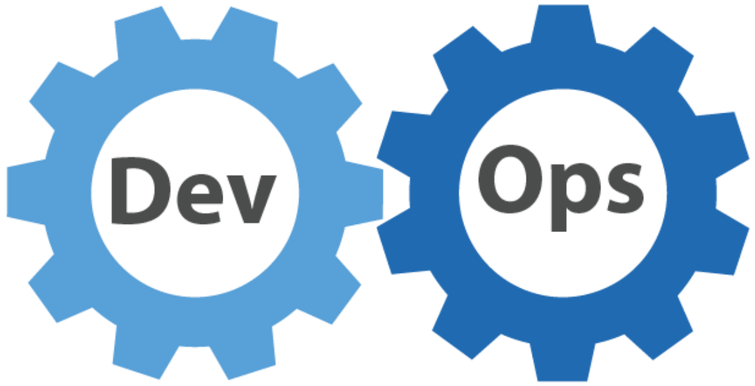
edureka!

Introduction to DevOps

edureka!

What is DevOps?

DevOps is composed of Dev (Development) and Ops (Operation)



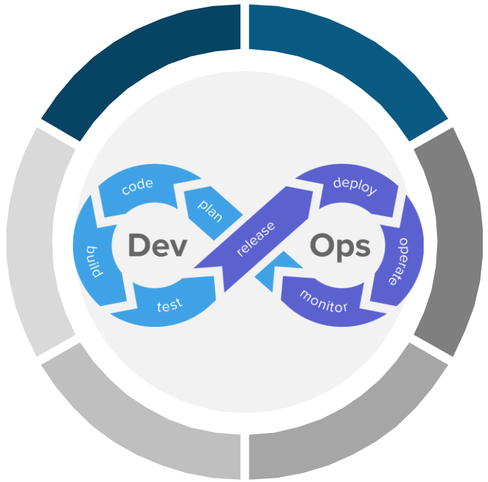
DevOps is the union of people, processes, and technology to provide value to customers in less time and more efficiently through rapid development, integration, and deployment.

edureka!

Benefits of DevOps

Accelerated Time to Market

Adapting to Market and Competition



Improving Mean Time to Recovery

Maintaining System Stability and Reliability

edureka!

DevOps Lifecycle

DevOps Lifecycle: Plan

Ideating, defining, and describing features and capabilities of applications and system

Tracking progress

Creating backlogs, tracking bugs, managing software development using Scrum, Kanban boards

Visualizing progress with dashboards

DevOps Lifecycle: Develop

Involves all aspects of an application or product development like writing, testing, reviewing, and integration of code

Building code to build artifacts for deployment in various environments

Rapid deployment through automated testing and continuous integration

DevOps Lifecycle: Deliver

1. Delivery is the process of deploying applications to production environments consistently and reliably

2. Defining the release management process with clear manual approval stages

3. Automating the delivery process helps the team frequently deliver with ease and confidence

DevOps Lifecycle: Operate

01 Involves maintaining, monitoring, and troubleshooting of applications in the production environment

02 The goal is to achieve zero-downtime in production while enforcing security and governance

03 Involves predictive identification of issues through application telemetry data and actionable alerting

edureka!

Introduction to Azure DevOps

edureka!

What is Azure DevOps?

Azure DevOps consists of a set of services that provides the tools for implementing and achieving the DevOps application lifecycle. Using Azure DevOps, we can:

01 Build and maintain a backlog

02 Host source code repositories

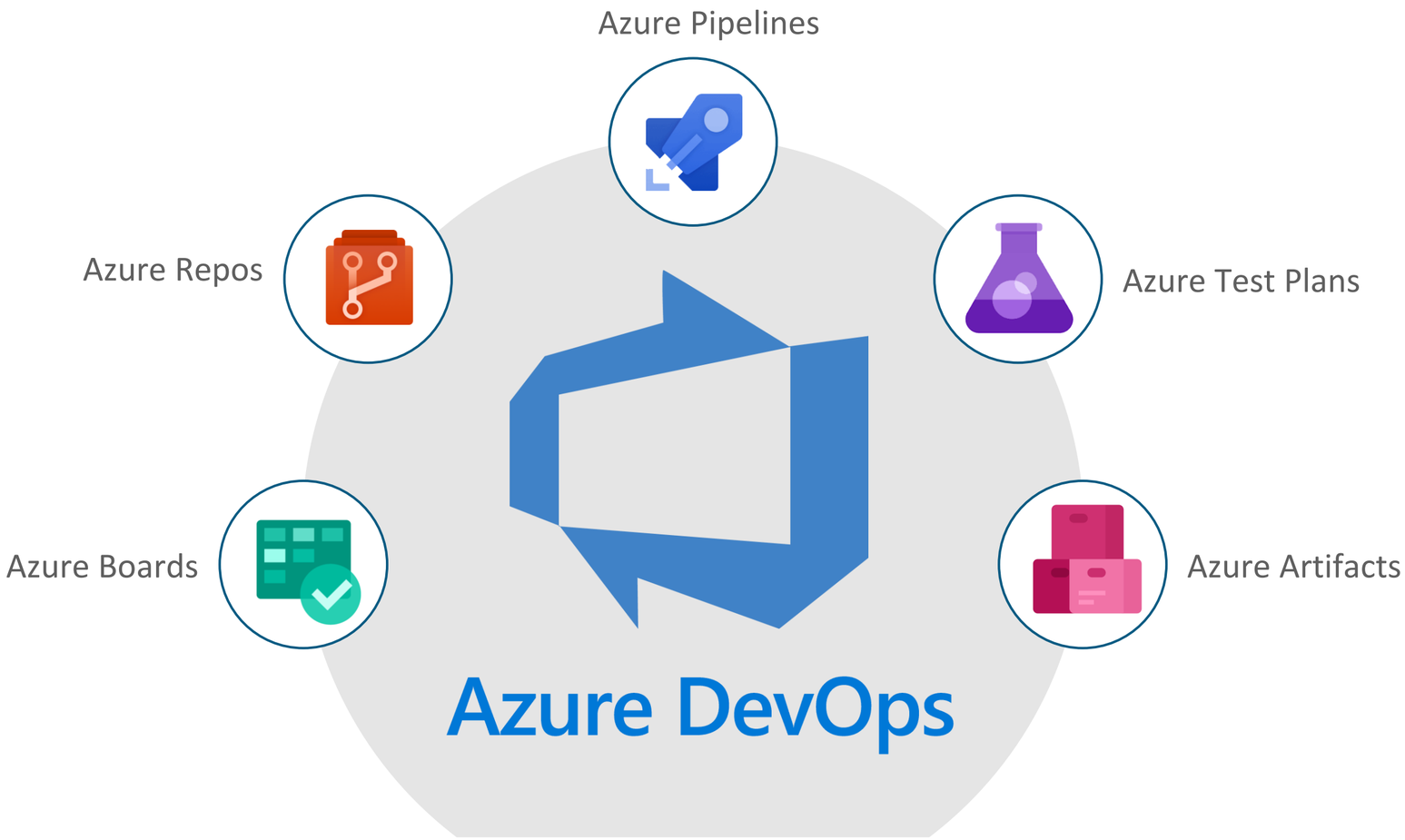
03 Achieve Continuous Integration

04 Achieve Continuous Deployment

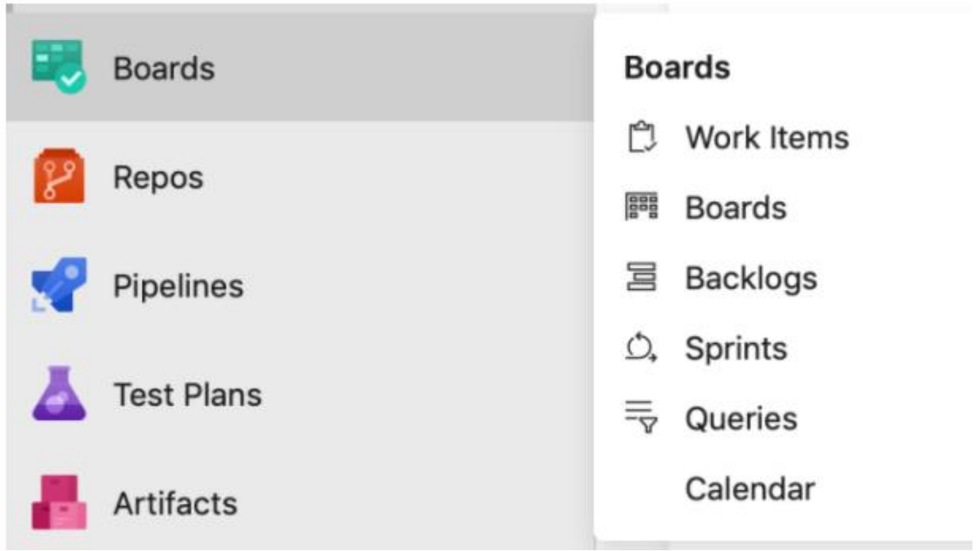
05 Automate testing before deployment

06 Implement CI/CD pipeline from code check in to code deployment

Azure DevOps Services



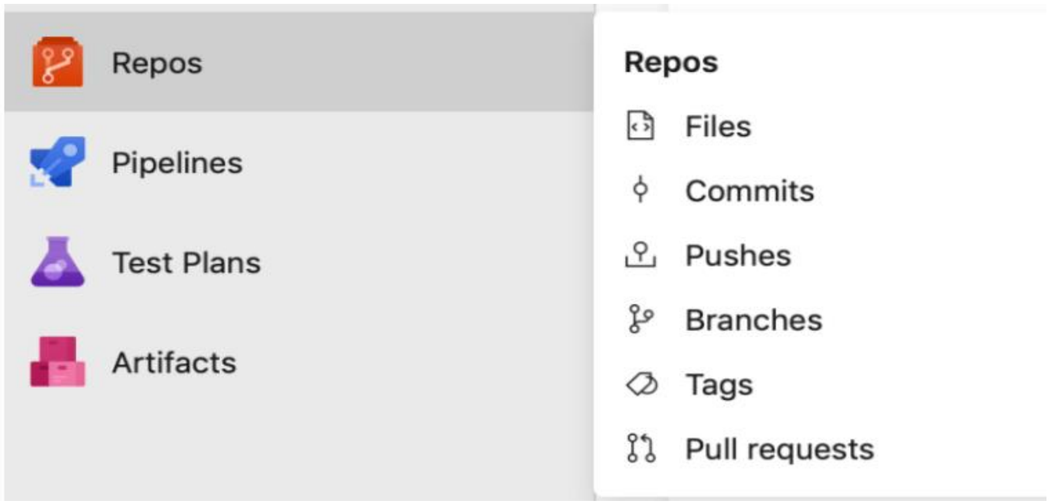
Azure Boards



Azure Boards is used to:

- Manage software projects
- Support Scrum and Kanban
- Customize the dashboard
- Create reports
- Trace user stories, backlog items, tasks, features, and bugs associated with the project

Azure Repos



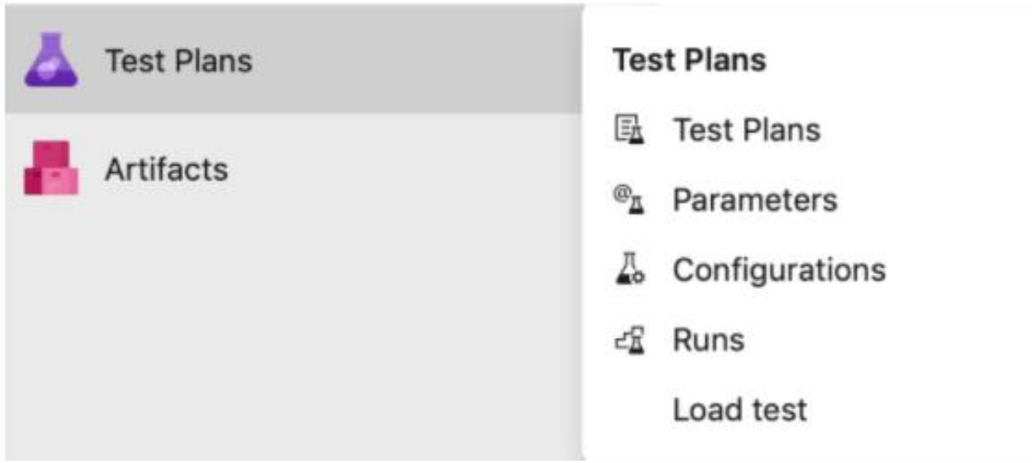
- Azure Repos is a set of version control tools used for managing the code
- Azure Repos supports Git and Team Foundation Version Control (TFVC)

Azure Pipelines

- Azure Pipelines is a cloud service that can automatically build and test code projects and make them available to end-users. It supports many languages

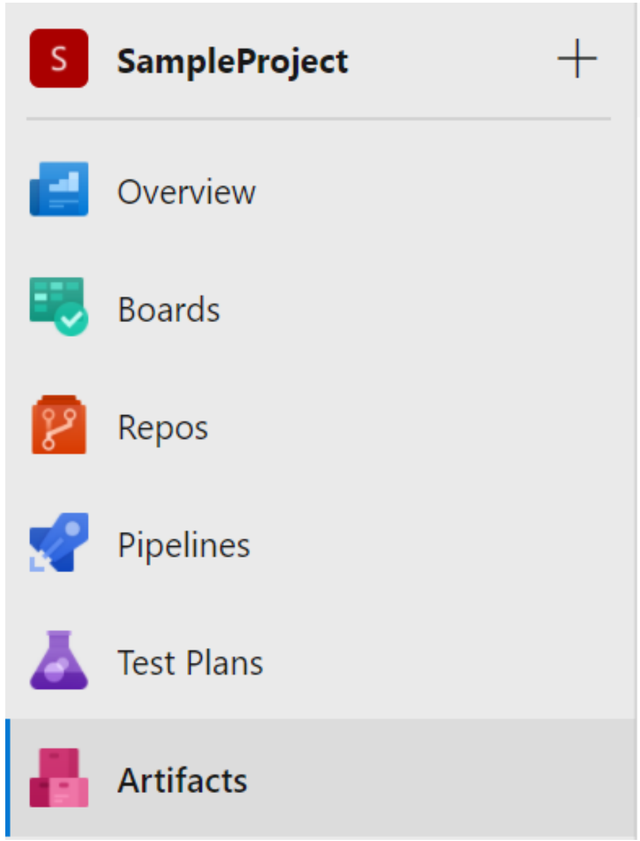
- Azure Pipelines combines Continuous Integration (CI) and Continuous Delivery (CD) to regularly and consistently test and build code and ship to any location

Azure Test Plans



It is a browser-based test management solution that provides all the capabilities required for planned manual testing, user acceptance testing, exploratory testing, and stakeholders' feedback.

Azure Artifacts



With Azure Artifacts, users can create and share NuGet, Maven, and NPM package feeds from public and private sources with any team.

Transformation Planning Using Azure Boards

edureka!

Transformation Planning

Azure DevOps follows an agile-based approach.

- Agile is defined as a term to describe the approaches for software development
- It emphasizes incremental delivery, team collaboration, continuous planning, and continuous learning
- It is based on iterative development and helps a team better plan and react to the inevitable changes in software development

Transformation Planning: Agile Manifesto

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Individuals and interactions over processes and tools

Responding to change over following a plan

Transformation Planning

- Planning is a key concept of DevOps
- Azure Board is used for Agile Planning
- Using Azure Boards Web Service, the teams can manage the software projects

Transformation Planning Using Azure Boards

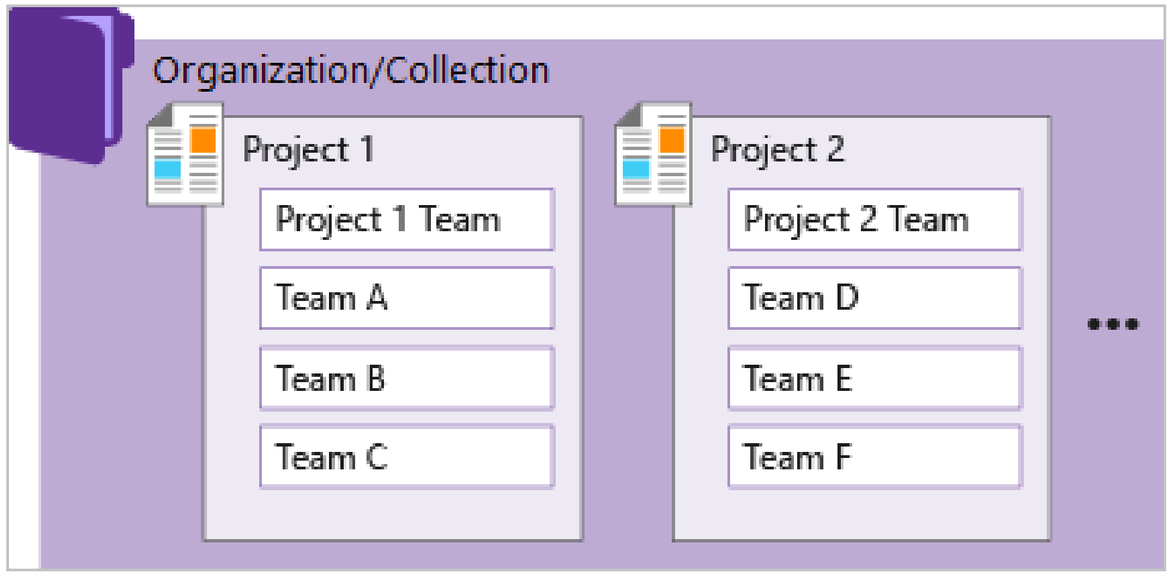
01 It provides a rich set of capabilities that includes native support for Scrum and Kanban, customizable dashboards, and integrated reporting

02 Developers/Teams can quickly and easily start tracking user stories, backlog items, tasks, features, and bugs associated with the project

03 Tracking can be done by adding work items based on the process and work item types available to your project

Project Selection and Team Structure

Project selection can be made from Azure Boards, and it can be set as an active project. All the further work can be done on that selected project. A team structure is defined to associate a user story or epic to a particular team.



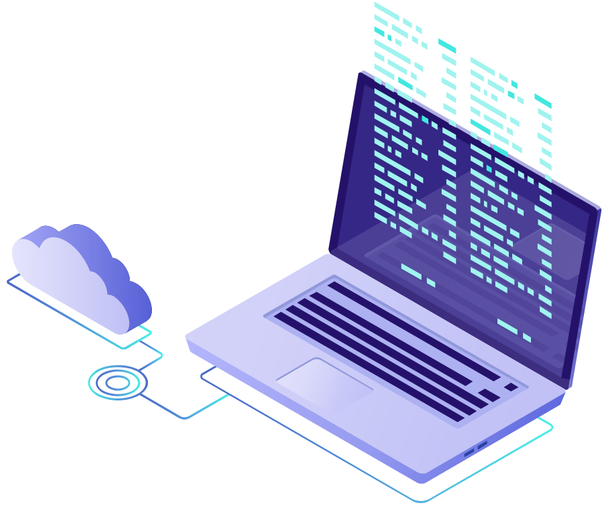
For example, a user story related to a feature implementation should be assigned to the relevant feature team. In this way, the team structures should be defined.

Introduction to Source Control

edureka!

What is Source Control?

Source control refers to code storage, code tracking, versioning, and managing code changes. Due to this, developers always work on the latest and right version of the code. Any older version of code can be accessed for any requirement.



It helps facilitate collaboration and accelerates release velocity. In the distributed teams, where work is happening at various geographical locations, it becomes quite challenging to manage, collaborate, and synchronize work.

edureka!

Benefits of Source Control

Types of Source Control Systems

Git: Distributed Version Control

TFVC: Centralized Version Control

Git: Distributed Version Control

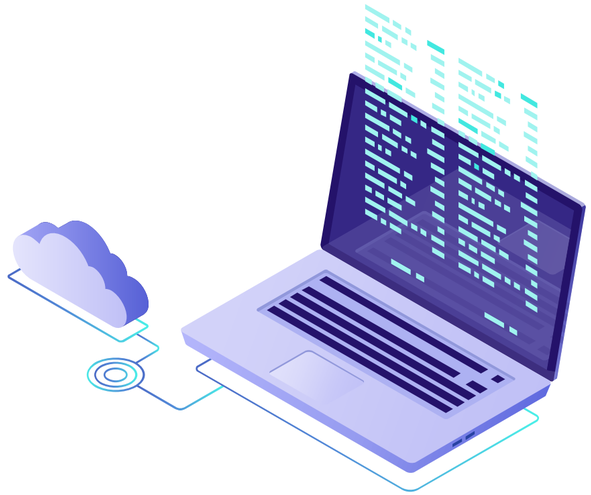
- Git is the most used version control system
- Users can create a branch of the main repository in their local system and work with the local repository
- Once the development is complete, the codes need to be merged into the main repository

TFVC: Centralized Version Control

Azure Repos also supports Team Foundation Version Control (TFVC), a centralized version control system. While team members have only one version of each file on their dev machines, the historical data is maintained only on the server.

Version Controlling

Version control is an integral part of any development process. Azure Repos is a set of version control tools that users can use to manage code.



01 Helps you track code changes

02 Takes the snapshot of the file that can be recalled later

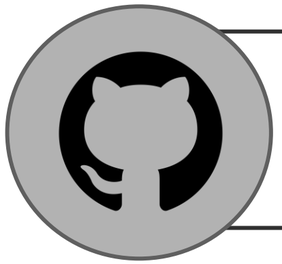
03 Helps in saving the work

04 Helps in coordinating the code changes across the team

Introduction to GitHub

GitHub

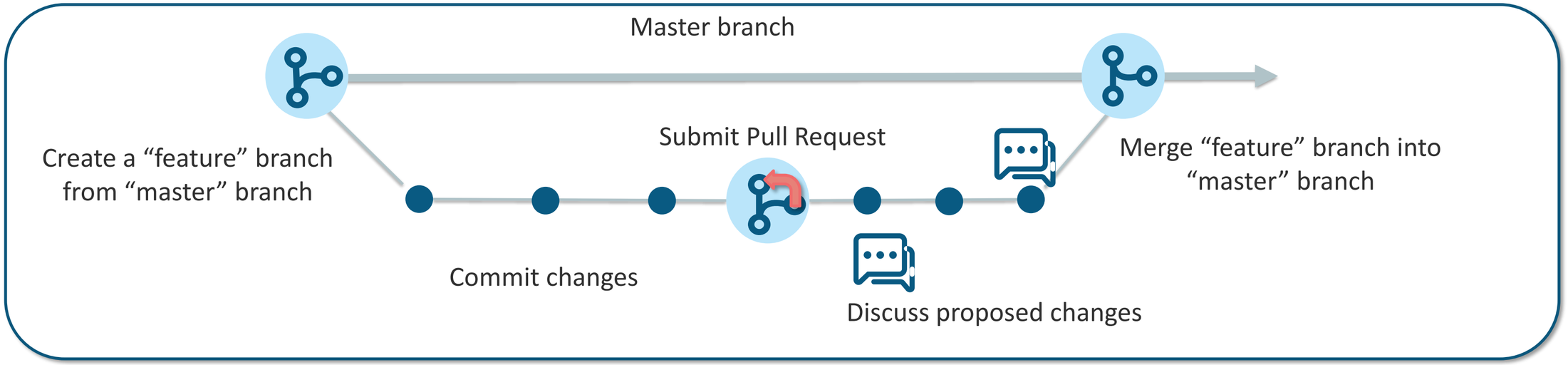
GitHub is a development platform that enables users/organizations to host and review code, manage projects, and build software. It provides essential DevOps features for the public and private projects of the user/organization.



GitHub supports features planning, bug fixing, or collaboration on changes. GitHub is the most preferred among software developers.

Introduction to GitHub

GitHub flow consists of various steps as described below:

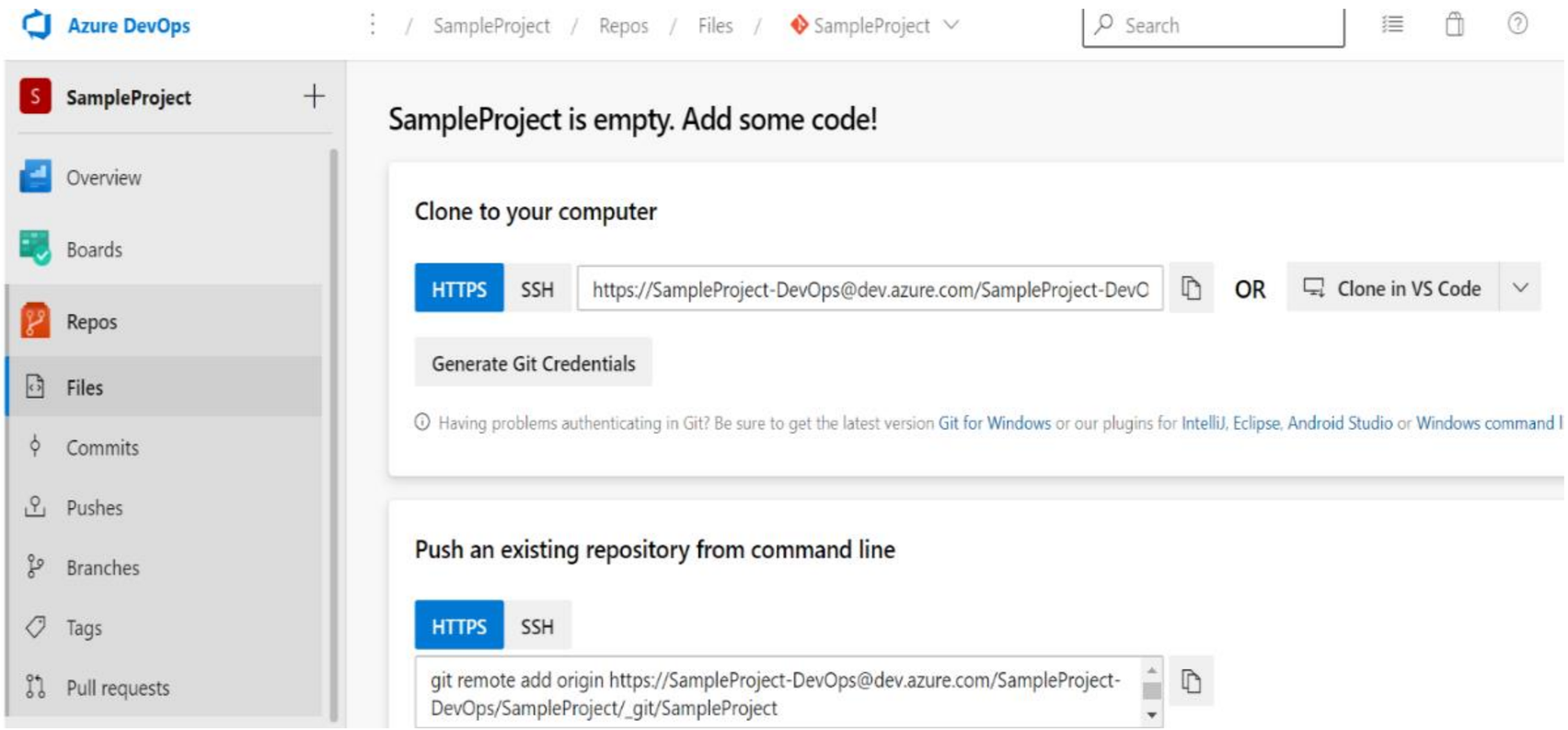


Demo: Migrating to Azure DevOps

edureka!

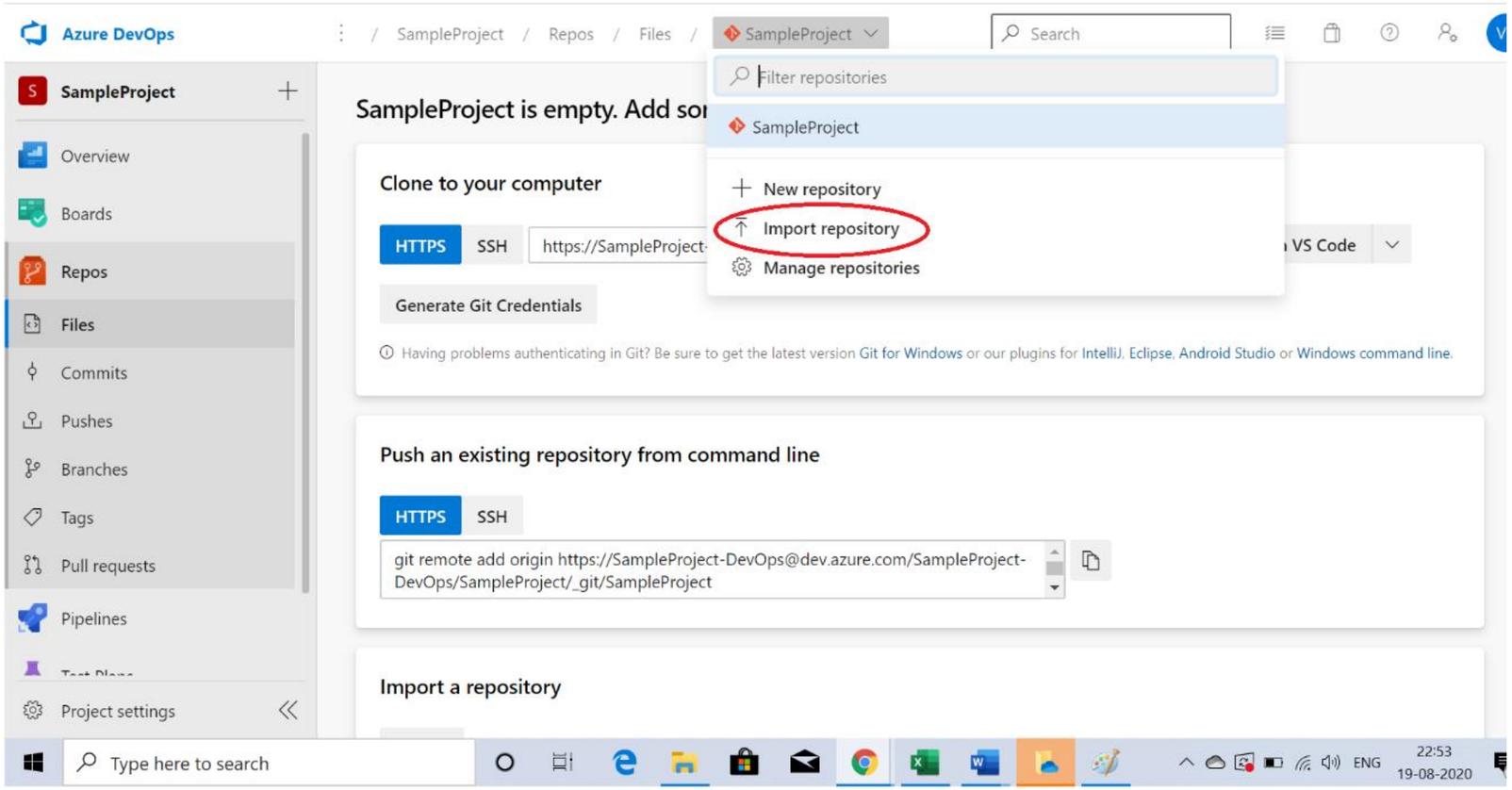
Migrating to Azure DevOps

- Create a project in Azure DevOps where the code from TFVC needs to be migrated
- Go to Repos→ Files. The snapshot should be as below



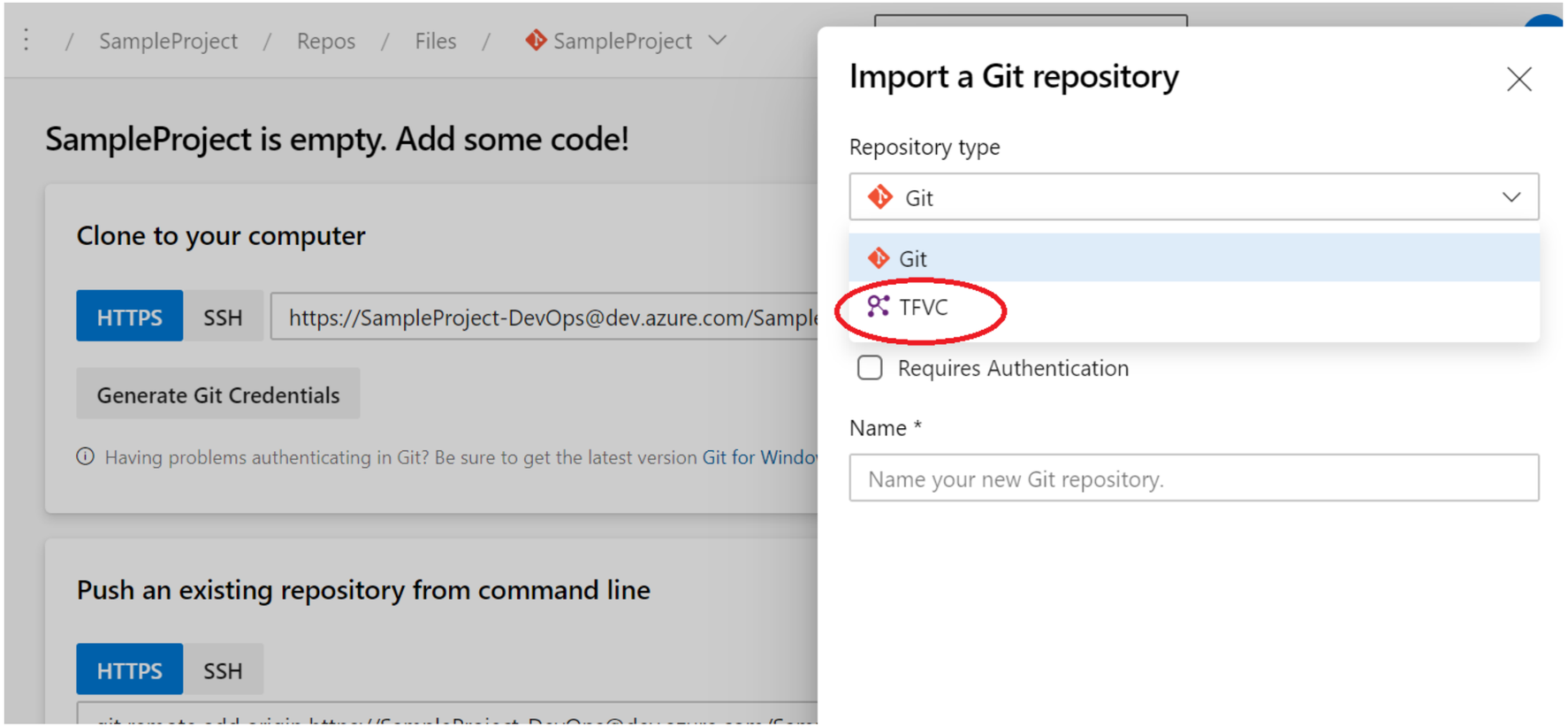
Migrating to Azure DevOps

- Click on the Project dropdown and select the Import Repository



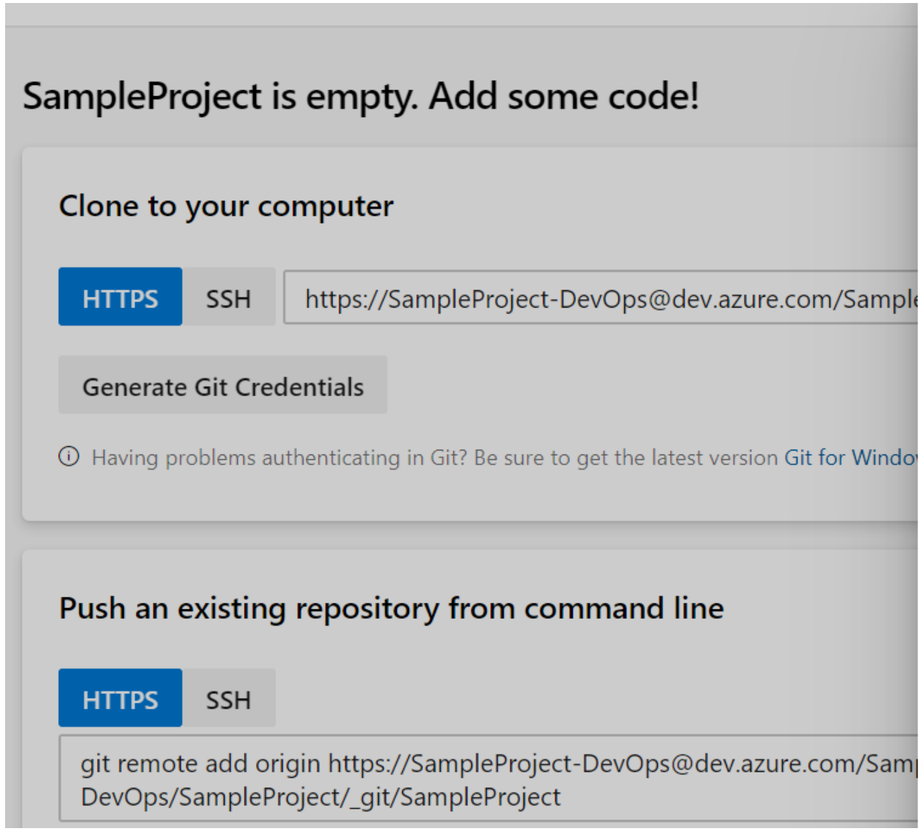
Migrating to Azure DevOps

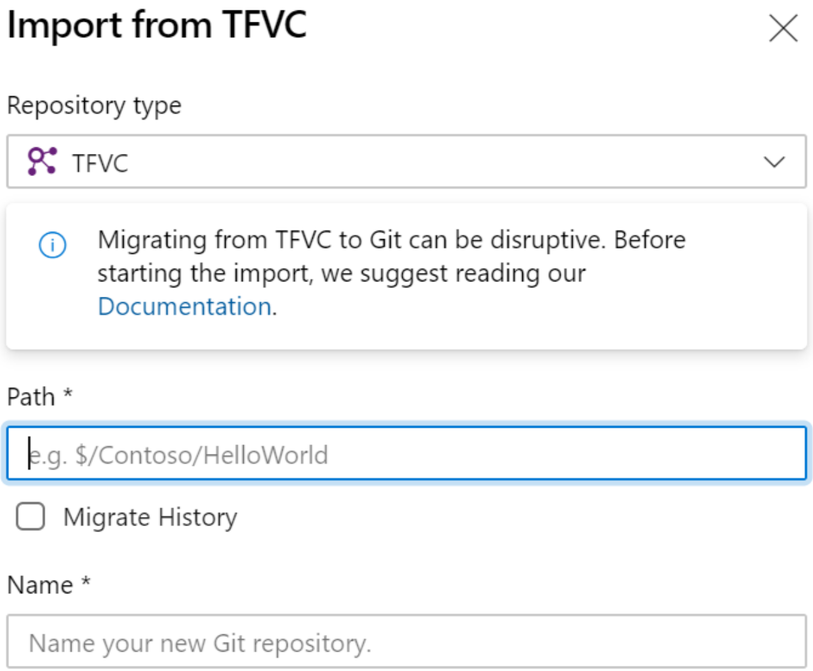
- Select TFVC, as shown below



Migrating to Azure DevOps

- Select the path of the TFVC source code path, as shown below. This will import the TFVC code repository to Git





Git Authentication in Azure Repos

edureka!

Git Authentication in Azure Repos

Two types of authentication can be used:

01.

Personal Access Tokens (PATs)

- Provides access to Azure DevOps and Team Foundation Server (TFS) without directly using username and password
- These tokens have an expiration date from when they are created
- Git Credential Manager can be used for generating tokens
- It is a very secure method of authentication

02.

Secure Shell (SSH)

- Works through public and private key pair that the user creates
- The public key is associated with username
- Azure DevOps will encrypt the data sent to the user with the public key
- Data is decrypted with the private key on the user computer, and the user gets the code

Structuring Git Repo

01

Distributed version control Git gives flexibility in how to use version control to share and manage code

02

Team members publish, share, review, and iterate the code changes through the Git branches shared with others

03

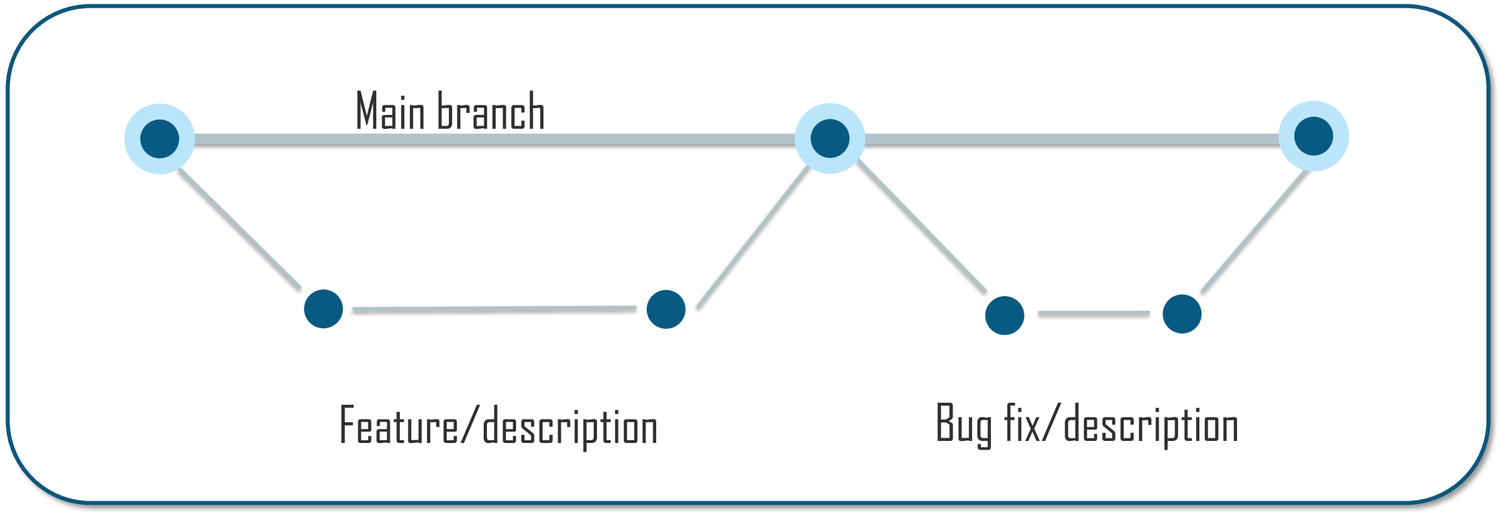
Any feature development should be done on the local branch and then merged into the Main branch

04

Similarly, any bug fix should be done on the local branch and merged into the Main branch. The development happens in an isolated local branch

Branch Strategy

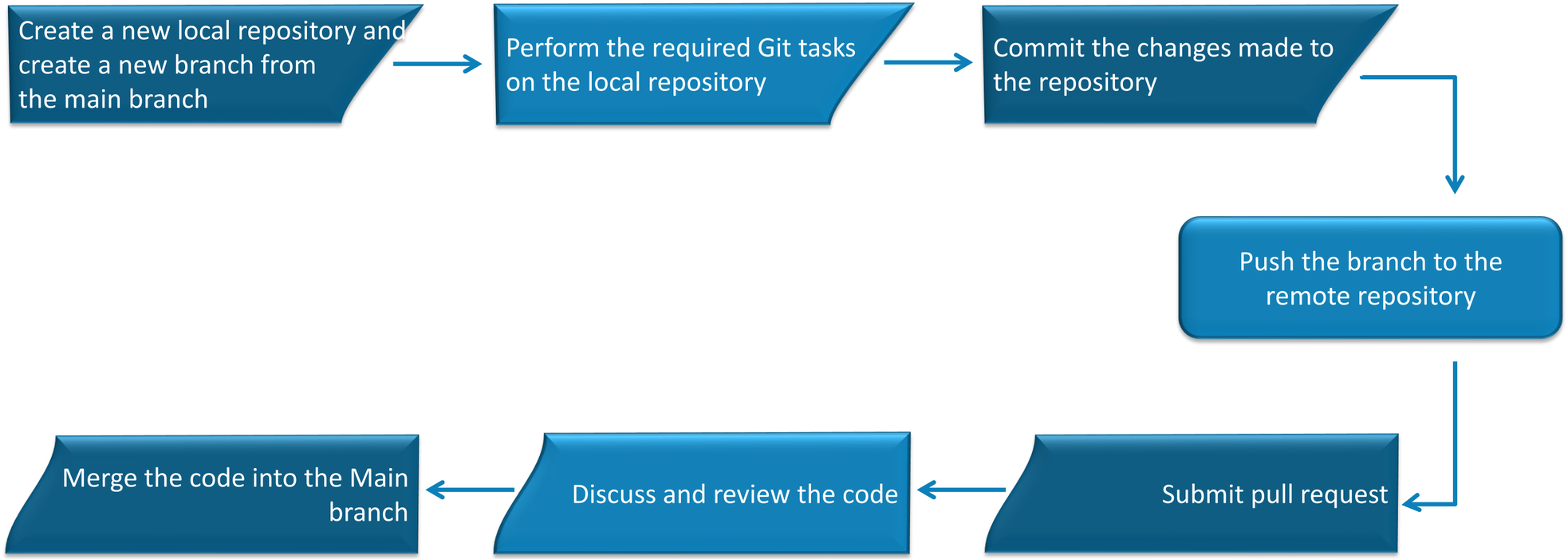
- Use the feature branches for all new features and bug fixes
- All the development work must be done on the local branch (**Feature branch**) taken from the **Main branch**
- These branches are also called **Topic branches**



Branch Strategy

- Merge feature branches into the Main branch using the pull requests
- Keep high quality, up-to-date Main branch
- Release Branch

Git Branching Workflow



Collaborating with Pull Requests in Azure Repos

The pull request is the collaborative process that lets the team discuss the changes in a branch and merge them once everyone approves.

Pull requests are used to get early feedback from others on work in progress, even if the team/user is not ready to merge the changes into another branch.

It is the best practice to download the main branch copy to the local branch and compile it to ensure the compilation.

Collaborating with Pull Requests in Azure Repos

Steps to follow for the Pull request:

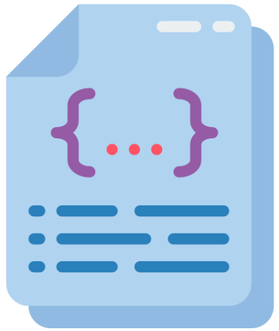
01	02	03	04	05
Do the required dev changes in the new branch	Commit the changes to the newly created branch	Push the branch to the main repository	Create a pull request along with specifying the reviewer	Get the review done successfully

edureka!

Git Hooks

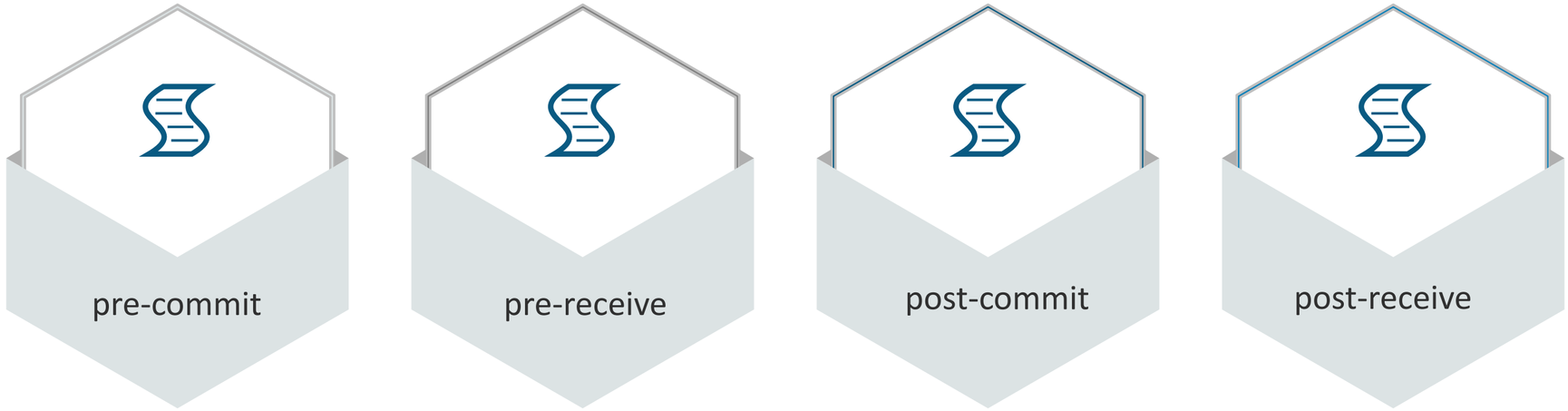
Git Hooks are scripts that Git executes after/before any Git event like Push, commit, and receive.

- This functionality helps the team enforce the compliance and get the event notification
- Action can be defined for execution before or after Git Hook script execution
- Git Hook resides in the local or server-side repository, and it is executed in response to any actions taken place in that repository
- The Git Hook resides in the .git/hooks directory of the Git repository



Git Hooks Scripts

Git Hooks scripts



- Checks the commit message for spelling errors
- Enforces project coding standards
- Sends Email/SMS to team members for any new commit
- Pushes the code to production

Fostering Inner Source

InnerSource is the practice of applying open-source patterns to projects with a limited audience.

Example

- A company may establish an InnerSource program that mirrors the structure of a typical open-source project, but it is accessible to that company's employees only
- It can be taken as an open-source program behind your company's firewall
- The Git repository has to be defined as Internal for InnerSource

Fostering Inner Source - Benefits

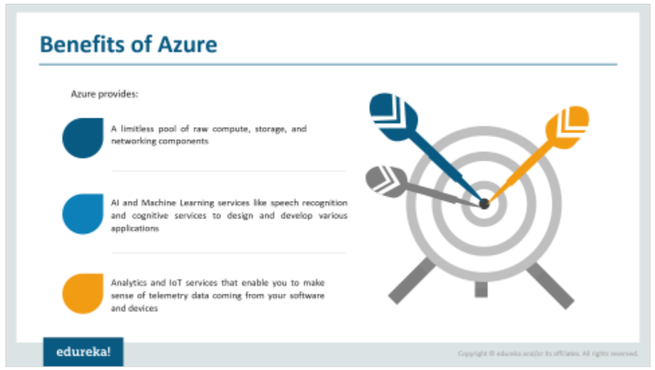
- Adapting to the best practices of other teams
- Transparent decision-making helps build a process, trust, and alignment
- Confidentiality of code while using the best practice

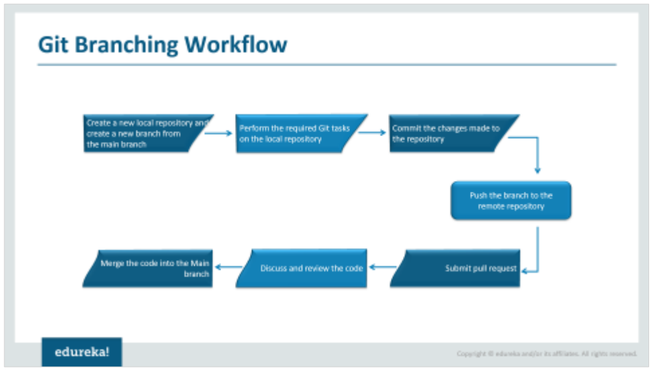
edureka!

Demonstration

- Demo 1: Create an Azure Account
- Demo 2: Azure Boards - Agile Planning and Portfolio Management
- Demo 3: Git Versioning Controlling
- Demo 4: Git Version Controlling through Visual Studio Code Editor
- Demo 5: Code Review with Pull Requests

Summary





edureka!

edureka!

Thank You

For more information please visit our website
www.edureka.co