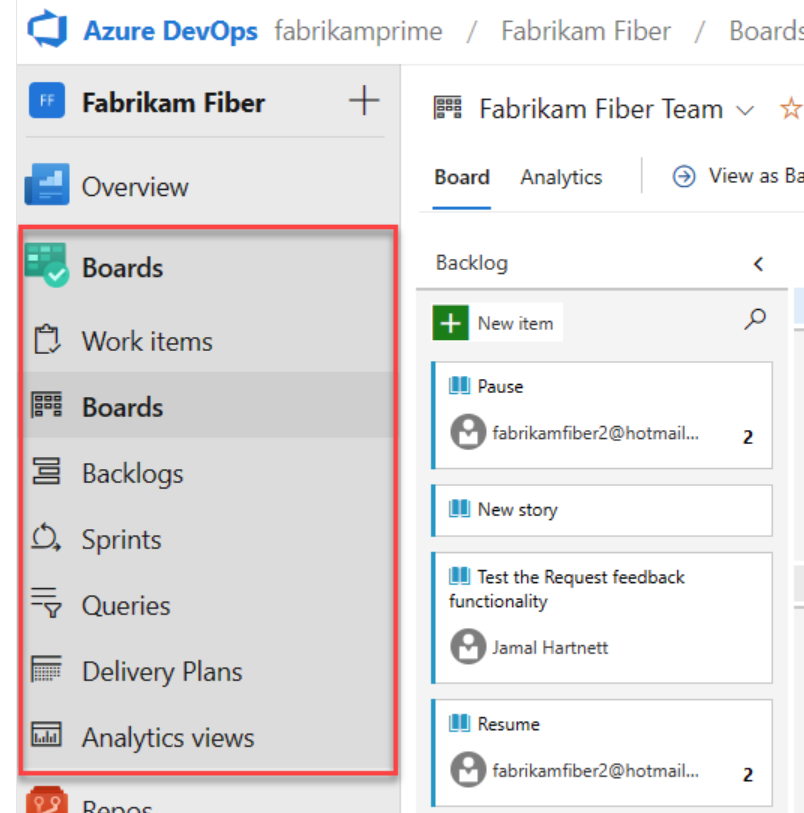
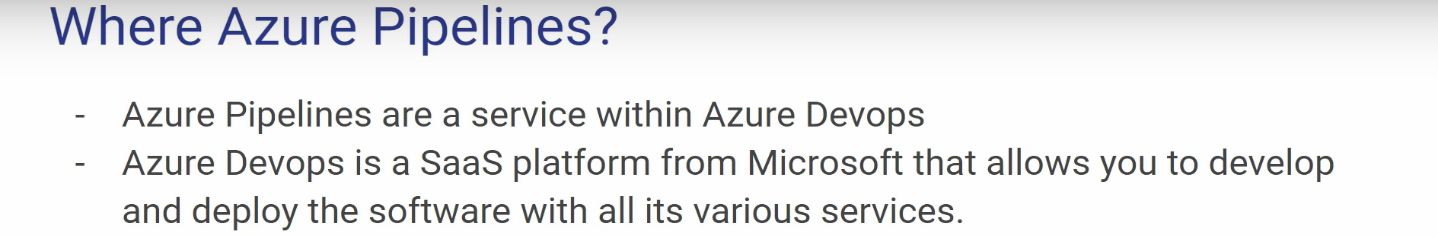
**Azure DevOps** supports a collaborative culture and set of processes that bring together developers, project managers, and contributors to develop software. It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches.

|  |  |
| --- | --- |
| [Azure Boards](https://azure.microsoft.com/services/devops/boards/) | Delivers a suite of Agile tools to support planning and tracking work, code defects, and issues using Kanban and Scrum methods. |
| [Azure Repos](https://azure.microsoft.com/services/devops/repos/) | Provides Git repositories or Team Foundation Version Control (TFVC) for source control of your code. |
| [Azure Pipelines](https://azure.microsoft.com/services/devops/pipelines/) | Provides build and release services to support continuous integration and delivery of your applications |
| [Azure Test Plans](https://azure.microsoft.com/products/devops/test-plans/) | Provides several tools to test your applications, including manual/exploratory testing and continuous testing. |
| [Azure Artifacts](https://azure.microsoft.com/products/devops/artifacts/) | Allows teams to share packages such as Maven, npm, NuGet, and more from public and private sources and integrate package sharing into your pipelines. |







**Types of Pipelines in Azure:**

1. Azure Pipelines (CI/CD Pipelines)
2. Data Pipelines (Azure Data Factory)
3. Synapse Pipelines
4. Machine Learning Pipelines

Types of Pipelines in Azure:

Azure Pipelines (CI/CD Pipelines): These pipelines are part of Azure DevOps and automate the process of building, testing, and deploying applications. It integrates with Git repositories, including GitHub and Azure Repos, to enable CI/CD workflows.

**Build Pipeline**: Focuses on automating the compilation and testing of the code. It ensures the codebase is correct before deployment by running unit tests, security checks, and static code analysis.

**Release Pipeline**: Manages the deployment of applications to various environments, such as development, testing, staging, and production. It can also handle rollback scenarios.

Data Pipelines (Azure Data Factory): In Azure Data Factory, a data pipeline is used to move and transform data from different sources to destinations. This is useful for orchestrating ETL (Extract, Transform, Load) operations.

**Control Flow Pipeline**: Manages the orchestration of different activities like copying data, running scripts, or processing data. It can contain triggers, conditions, and loops.

**Data Flow Pipeline**: Designed for transforming data during transit, such as data filtering, aggregation, or joining datasets before they reach their destination.

Synapse Pipelines: Azure Synapse Analytics uses pipelines for big data processing and analytics. These pipelines can run Spark jobs, SQL queries, and ETL processes in a massively parallel processing environment.

Machine Learning Pipelines: In Azure Machine Learning, pipelines automate the process of building, training, and deploying machine learning models. These pipelines help with data preprocessing, model training, validation, and deployment.

A **YAML pipeline** in Azure refers to a pipeline defined using YAML (Yet Another Markup Language) syntax.  
  
**Structure of a YAML Pipeline**

A YAML pipeline consists of stages, jobs, steps, tasks, and triggers, arranged in a hierarchical structure.

Here’s an example of a simple YAML pipeline:  
  
trigger:

branches:

include:

- main

pool:

vmImage: 'ubuntu-latest'

stages:

- stage: Build

jobs:

- job: BuildJob

steps:

- task: UseDotNet@2

inputs:

packageType: 'sdk'

version: '6.x'

installationPath: $(Agent.ToolsDirectory)/dotnet

- script: dotnet build

displayName: 'Build Solution'

- stage: Deploy

dependsOn: Build

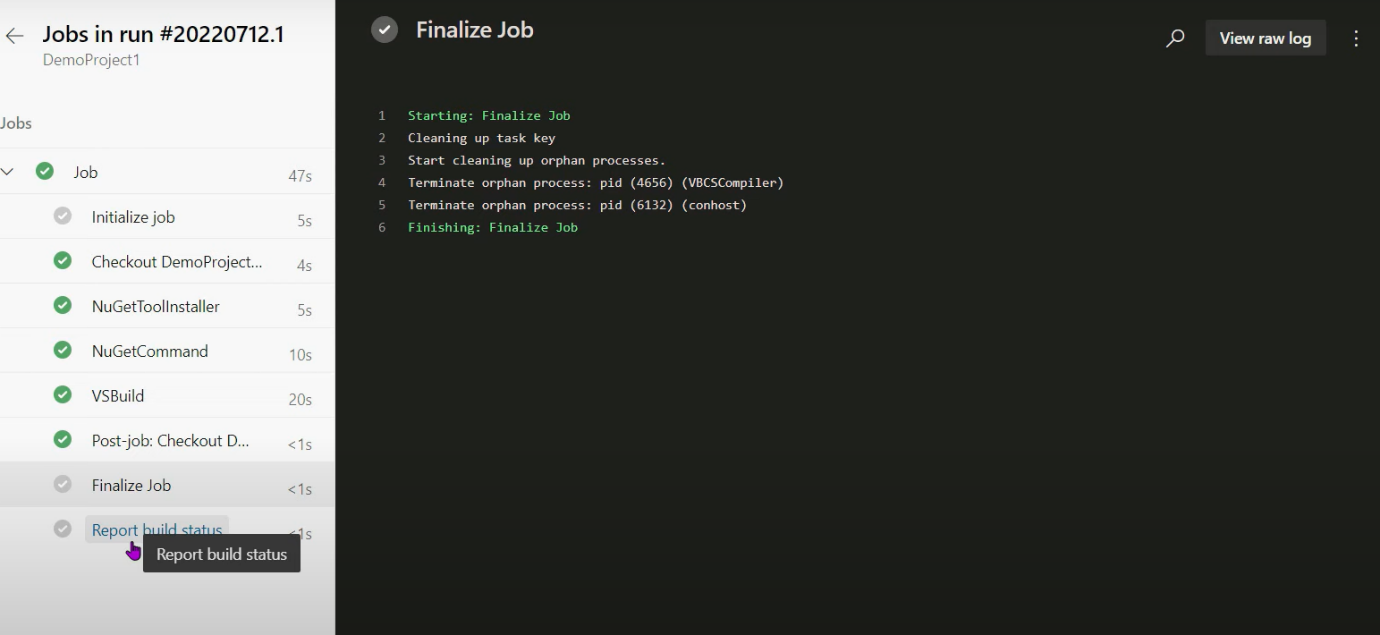
jobs:

- job: DeployJob

steps:

- script: echo "Deploying to Production"

displayName: 'Deploy'



Key Components:

Trigger: Specifies when the pipeline should run.   
This section defines when the pipeline will be triggered. In this case, it runs automatically when changes are made to the main branch of the repository.

Pool: Defines the virtual machine or agent that will run the tasks. Azure Pipelines provides different agent types like ubuntu-latest, windows-latest, etc.  
The agent used here is an Ubuntu-based system (ubuntu-latest).

Stages: These are high-level phases in the pipeline. For example, you might have stages like Build, Test, and Deploy.  
This pipeline is divided into two stages: **Stage 1**: Build Contains a single job, BuildJob.

It first sets up the .NET SDK using the task UseDotNet@2, specifying version 6.x.

Then, it runs the dotnet build command to build the project.   
  
**dotnet build** is a command that tells the build system to compile the .NET project.The script is not custom but part of the .NET CLI (Command Line Interface).  
  
**Stage2**: The Deploy stage only runs after the Build stage completes (dependsOn: Build).

It contains a single job, DeployJob, that runs a simple shell command to display the message "Deploying to Production." This simulates a deployment process.

Jobs: A job is a group of steps that run on the same agent. Each stage can contain multiple jobs that run in parallel or sequentially.

Steps: These are the individual actions within a job, like running scripts or tasks (e.g., dotnet build, npm install).

Tasks: Predefined actions such as running a script, installing dependencies, or deploying to a server. Tasks can be custom or predefined by Azure.  
  
**Summary**:

The pipeline automatically triggers on changes to the main branch.

It uses an Ubuntu machine to run the steps.

It first builds the application using .NET in the Build stage.

After the build succeeds, it moves to the Deploy stage, simulating a deployment step.