Chapter1: Introduction to DevOps

- Different team involved in SDLC [Business Analyst, Development team and Operation team]
- How software development worked before DevOps #Waterfall
- What is DevOps? #Improvement
- Who should learn DevOps? #Everyone
- What are the benefits of DevOps? #Cost optimization, Agility and time to market.

Chapter2: What is Azure DevOps

- What is Azure DevOps?
- Relationship between Azure and Azure DevOps
- Azure DevOps Services #Repo, Board, Pipeline, Test Plan(For manual and automation testing) and Artifact.
- Choose Azure DevOps Server# Cloud and on promises.
- Relation Between VSTS [formal name of Azure DevOps] to Azure DevOps

Chapter3: Introduction to Azure Portal

- Azure DevOps Portal [https://dev.azure.com/]
- Create an account on Azure DevOps

Chapter4: Organisation setting

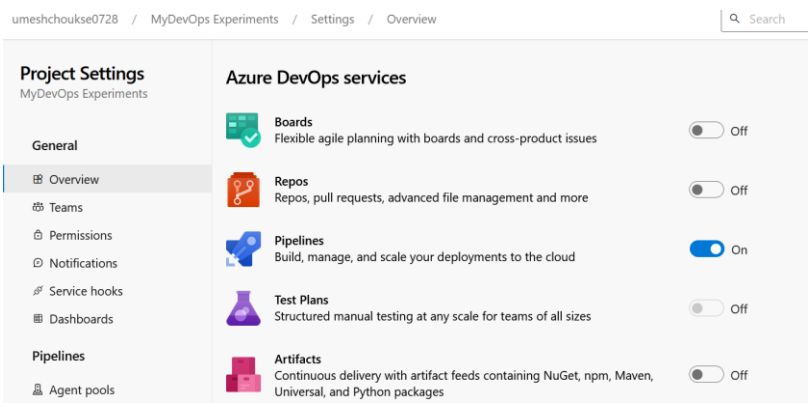
- What is an organization in Azure DevOps #Organisation#Project#Teams
- Multiple organization in Azure DevOps
- How to create an organization on Azure DevOps
- Organization Settings

Chapter5: Project Setting

- What is Project? #Project is a place where complete development of application is happening.
- What types of projects we can create in Azure DevOps? #Public project and Private project
- How to Create New Project in Azure DevOps? #Basic, Agile, CMMI and Scrum
- How to View already created projects?
- Project Settings

Issue1: Unable to see Repo, Board, Pipeline, TestPlan and artifact?

Goto Project Setting > Overview > Enable the checkbox



Chapter6: Setup a team in Azure DevOps project

- what is the team?
- invite a new team member in your team
- set member permission

What is a Team

- ✓ A team is a group of people who are responsible for the software development.
- ✓ A team includes Developers, QA, Scrum master, PO, Person responsible for deployment etc.
- ✓ A user must have Administrator level access to make changes in Team.

Chapter7: Azure board introduction

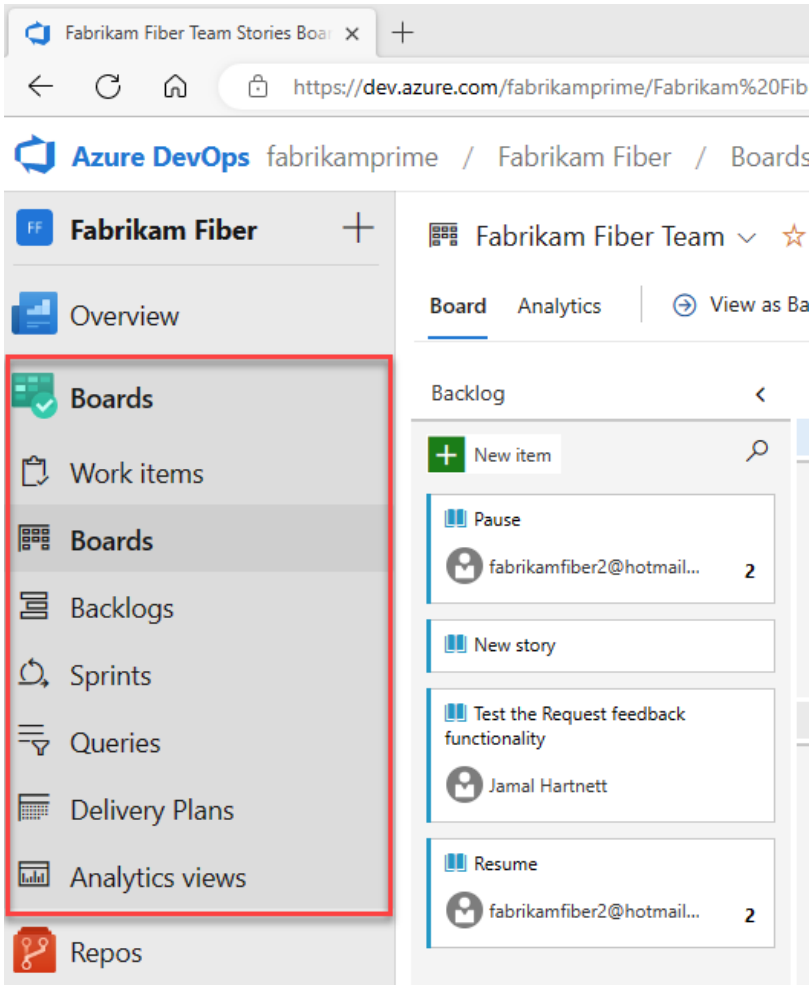
- what are Azure Boards?
- use of Azure Boards
- who will work on Azure Board
- Azure Board types

What is a Azure Boards

- ✓ Azure board is one of the main service of Azure DevOps which is used in a project to
  - ✓ Track the work with Kanban board
  - ✓ Work with backlog
  - ✓ Team dashboard
  - ✓ Reporting

Kanban Board: This board has all the information that you are going to deliver as a part of project.

Work items: Task, subtask, EPIC, USER STORY and Improvement

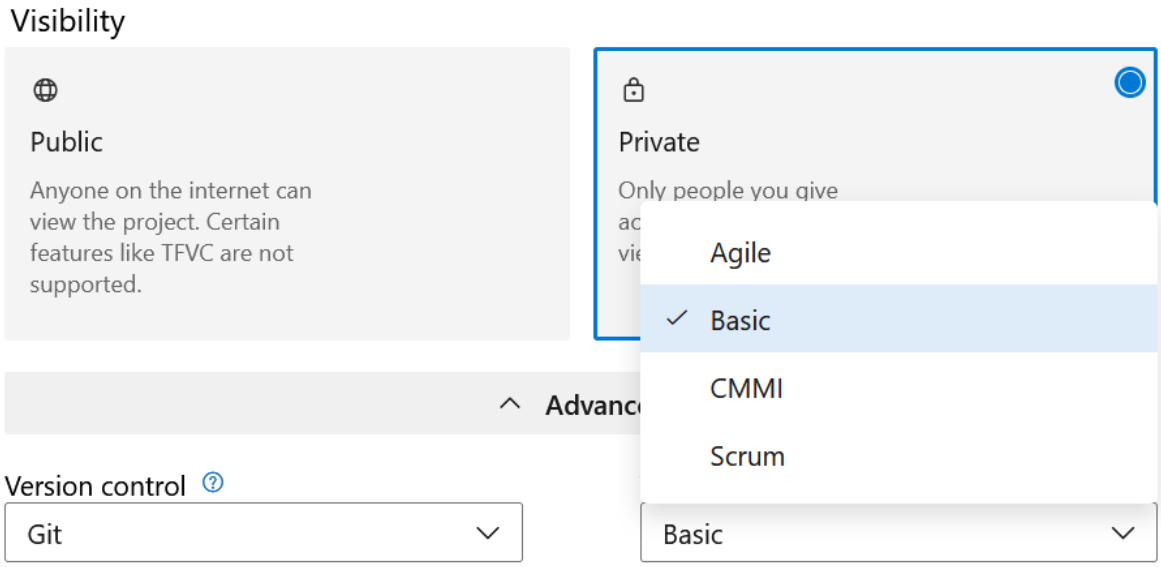


Use of Azure Boards

- ✓ Azure boards are useful in -
- ✓ Managing and tracking work.
- ✓ It provides a clear picture of work done/doing by team member
- ✓ You can give access of board to you stakeholder allow him to see the progress of your development.
- ✓ You can manage scrum, sprint with azure boards.
- ✓ Get different type of reports.
- ✓ And lots other that we will learn when we will work on boards.

Who will work on Azure Board

- ✓ The Team.
- ✓ Development team will work on managing their work (task, stories, bug etc..).
- ✓ PO will work to manage the backlog.
- ✓ Stakeholder will work to see the status of what team is doing.
- ✓ Scrum master will work on azure board to get different types of reports and to see the progress of team.



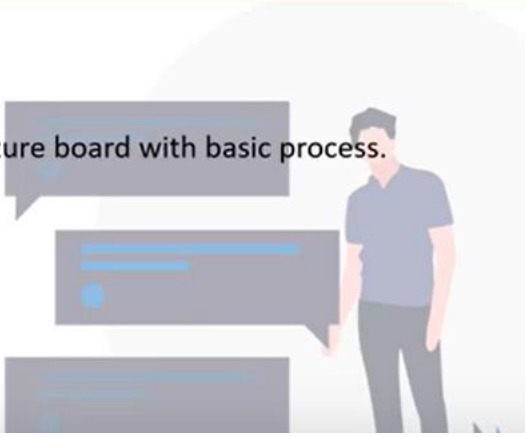
Azure board types

- ✓ There are four type of **work item process** while creating a new project.
- ✓ The type of board depends on these work item process.
- ✓ Each type provide different types of work item and work flow.

Chapter8: Azure Board with basic process type

We will learn

- ✓ What is work item?
- ✓ What type of work items are available in Azure board with basic process.
- ✓ Introduction to Kanban board.
- ✓ Create a work item.
- ✓ Update work item status.
- ✓ Add task to an issue.



What is a Work item

- ✓ A work item is a unit (small or large) of work which has several characteristics and is a part of your product development.
- ✓ Characteristics –
  - ✓ Title
  - ✓ Description
  - ✓ AssignedTo
  - ✓ CompletedBy
  - ✓ Etc...

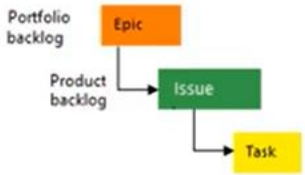
Example: EPIC, USERSTORY, BUG, IMPROVEMENT.

Work items in Azure boards (basic)

- ✓ There are three work items in Azure board with basic process.
- ✓ Epic
- ✓ Issue
- ✓ Task

Work items in Azure boards (basic)

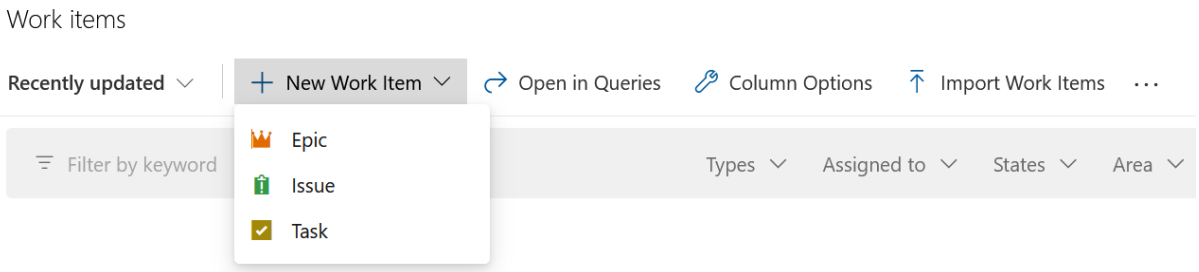
- ✓ There are three work items in Azure board with basic process.
- ✓ Epic
- ✓ Issue
- ✓ Task



WorkFlow

- ✓ Work flow is the process of updating work item progress.
- ✓ This is very basic work flow

  TODO -> InProgress -> Done



Chapter9: Azure Board

- ✓ Create a new work item on azure board.
- ✓ What fields are available in a work item.
- ✓ How to add attachment to an work item?
- ✓ How to see history of work item?

Work items fields

- ✓ Following fields are available in a work item-
- ✓ **Title –** Enter a description of 255 characters or less. You can always modify the title later.
- ✓ **AssignedTo –** Assign the work item to the team member responsible for performing the work.
- ✓ **State –** Represents the current state (Column of Kanban board) of work item.
- ✓ **Reason –** Represents the reason for state change.
- ✓ **Area-** Represents the area path associated with the product or team
- ✓ **Iteration –** Represents the sprint or the iteration in which the work is to be completed.
- ✓ **Description –** The details of the work item.
- ✓ **Effort –** A relative estimate of time required to complete a work flow.
- ✓ **Discussion –** This field is used to add a comment to give/ask information or to start a discussion regarding that work item.
- ✓ **Start Date –** A start date for the work item.
- ✓ **Target Date –** An end/target date indicating the successful workflow of work item.
- ✓ **Priority –** Represents a number which specify the value of this work item to business –
  - ✓ **1**: Product cannot ship without the successful resolution of the work item, and it should be addressed as soon as possible.
  - ✓ **2**: Product cannot ship without the successful resolution of the work item, but it does not need to be addressed immediately.
  - ✓ **3**: Resolution of the work item is optional based on resources, time, and risk.
  - ✓ **4**: Resolution of the work item is not required.

Chapter10: Working with backlog

- ✓ What is backlog?
- ✓ What is the benefit of creating a backlog?
- ✓ How to create the backlog?

What is a backlog

- ✓ A backlog is a collection of work items which will be used for future development.
- ✓ There are two types of backlog –
  - ✓ Product backlog
  - ✓ Sprint backlog
- ✓ Product backlog is an ordered list of everything that is known to be needed in the product.
- ✓ Sprint backlog is the collection of work items which are in TOTO state.

Backlog is used for-

- ✓ Quickly define the task (user story, bug, product backlog etc.)
- ✓ Manage priority
- ✓ Add details and estimate to the backlog items in advance
- ✓ Quickly assign work items to team members
- ✓ Group work items in a hierarchy
- ✓ Forecast work to estimate what can be delivered within a sprint

Open your backlog

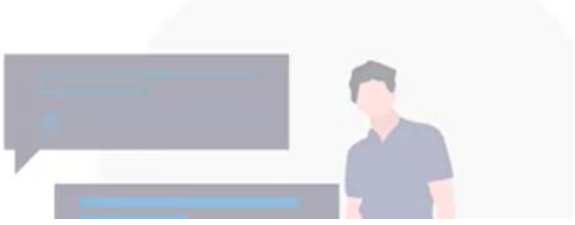
From your web browser, do the following steps to open your product backlog.

1. Sign in to your project (https://dev.azure.com/{yourorganization}/{yourproject}).
2. Select (1) **Boards** > (2) **Backlogs**.

Chapter11: Sprint

A sprint is a short, time-boxed period when a scrum team works to complete a set amount of work.

- ✓ Create a new sprint.
- ✓ Assign work items to a particular sprint.
- ✓ Add start date and end date to a sprint.



Chapter12: Azure DevOps Board with Agile work process

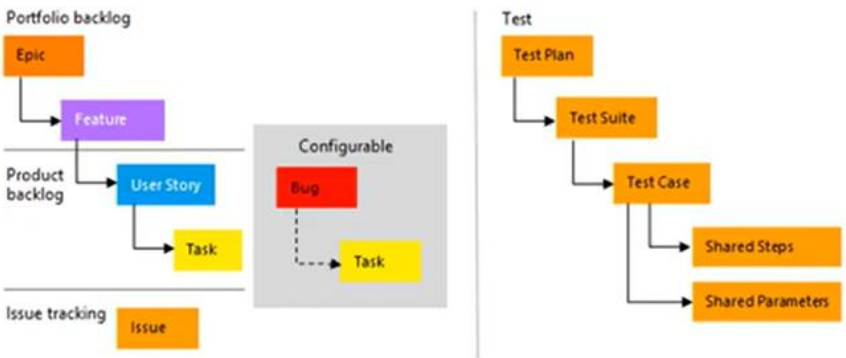
We will learn

- ✓ How to create project with Agile process?
- ✓ What types of work items are available agile process.
- ✓ What is the relationship of work items in agile process.
- ✓ Workflow in agile process.

Work items in Agile process

- ✓ Following work item types are available in Agile process -
  - ✓ Bug
  - ✓ Epic
  - ✓ Feature
  - ✓ Issue
  - ✓ Task
  - ✓ Test Case
  - ✓ User Story

Work items relationship in Agile process



- ✓ **Bug** - Something which is missed or implemented in wrong way.
- ✓ **Epic** - An epic represents a business initiative to be accomplished.
- ✓ **Feature** - A feature typically represents a shippable component of software.
- ✓ **Issue** - Any other custom type
- ✓ **Task** – Small unit of work.
- ✓ **Test Case** – Test case for a feature
- ✓ **User Story** – Implementation of new work.

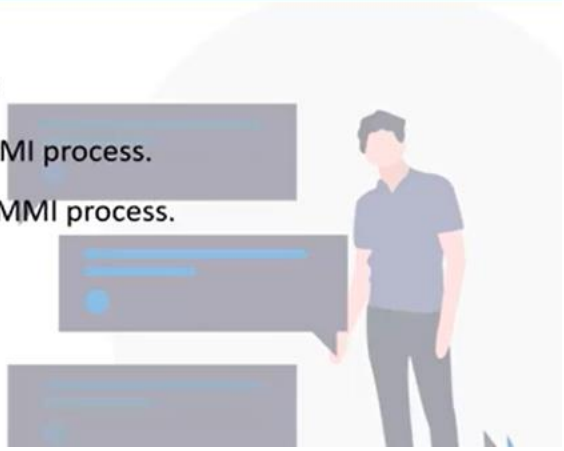
Chapter13: Create backlog and sprint in Agile process

Link: https://www.atlassian.com/agile/project-management/sprint-backlog

Chapter14: Azure board with CMMI process

We will learn

- ✓ How to create project with CMMI process?
- ✓ What types of work items are available CMMI process.
- ✓ What is the relationship of work items in CMMI process.
- ✓ Workflow in CMMI process.
- ✓ Creating backlog in CMMI process.
- ✓ Working with Sprints in CMMI process.



- ✓ Bug
- ✓ Change Request
- ✓ Epic
- ✓ Feature
- ✓ Issue
- ✓ Requirement
- ✓ Review
- ✓ Risk
- ✓ Task
- ✓ Test case

Chapter15: Azure board with Scrum process

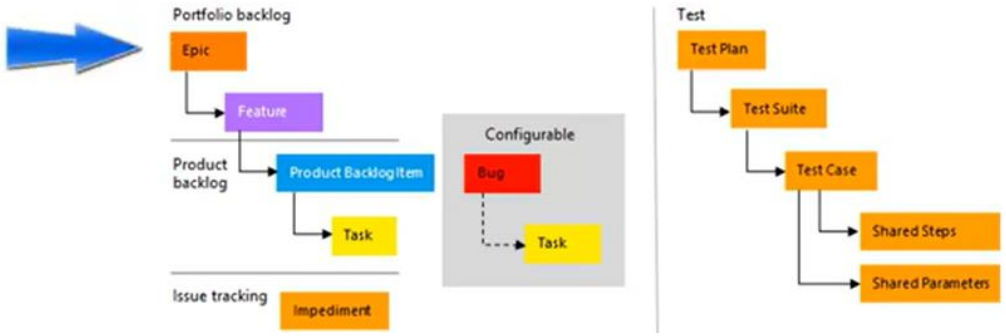
We will learn

- ✓ How to create project with Scrum process?
- ✓ What types of work items are available Scrum process.
- ✓ What is the relationship of work items in Scrum process.
- ✓ Workflow in Scrum process.
- ✓ Creating backlog in Scrum process.

Work items in Scrum process

- ✓ Following work item types are available in Agile process -
  - ✓ Bug
  - ✓ Epic
  - ✓ Feature
  - ✓ Impediment
  - ✓ Product backlog item
  - ✓ Task
  - ✓ Test case

Work items relationship in Scrum process



- ✓ **Bug** - Something which is missed or implemented in wrong way.
- ✓ **Epic** - An epic represents a business initiative to be accomplished.
- ✓ **Feature** - A feature typically represents a shippable component of software.
- ✓ **Impediment** - Any work which has an impediment

Chapter16: Change column options on azure board

We will learn

- ✓ What is this feature column options.
- ✓ Add column options
- ✓ Remove column options.
- ✓ Sort by column
- ✓ Order of column

Chapter17: Query in Azure DevOps

We will learn

- ✓ What is a query
- ✓ How to see query dashboard
- ✓ How to view a query
- ✓ How to run a query
- ✓ How to rename, delete a query
- ✓ How to send query on email

Query:

What is a query :

A query is a combination of few logic which is applied on work items.

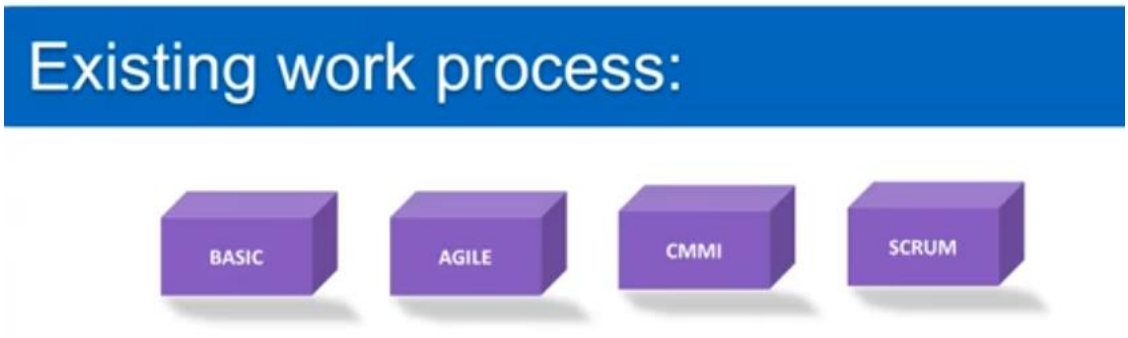
What do we need query :

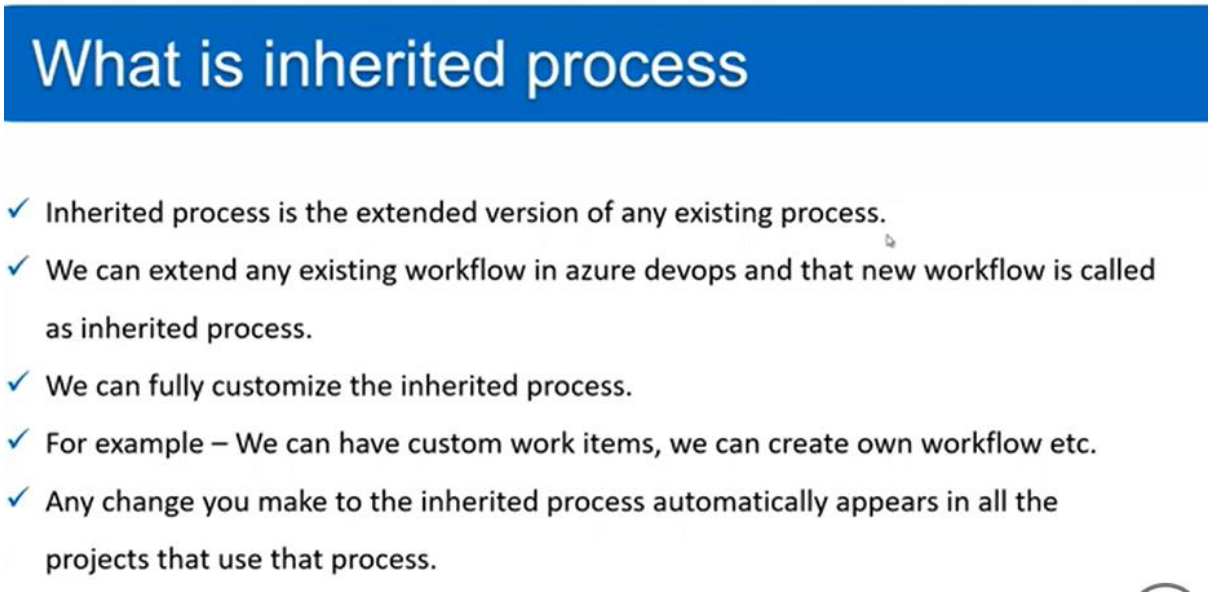
A query is a used to filter work items based on several logic.

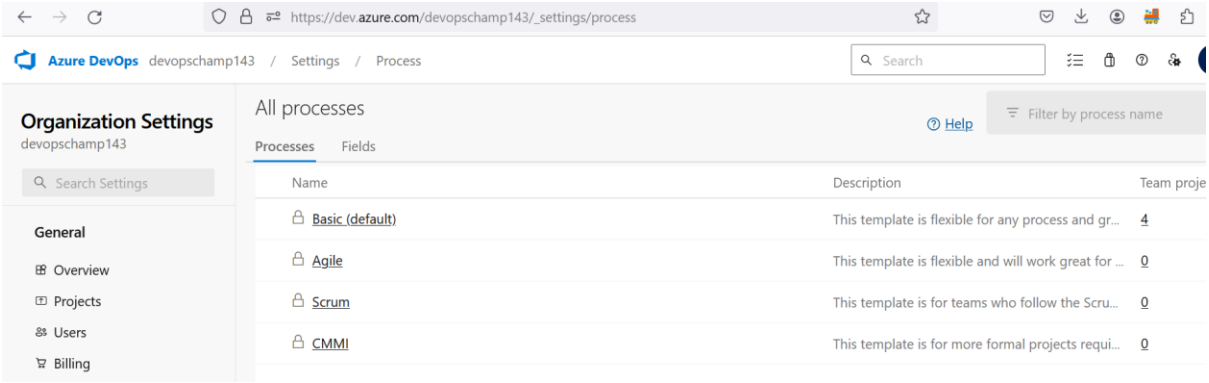
Chapter18: Add new Query in Azure DevOps

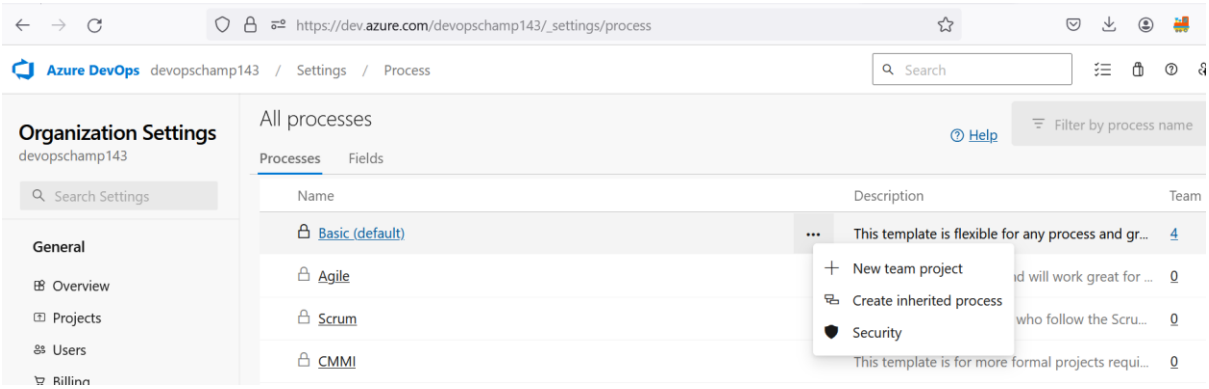
1. **Navigate to Queries**: Go to your Azure DevOps project and select the "Boards" or "Work" tab, depending on the version you're using.
2. **Access Queries**: Look for the "Queries" option in the navigation menu. Click on it to enter the queries section.
3. **Create a New Query**: Depending on whether you want to create a new query from scratch or based on an existing one, you can either click on "New Query" or "New Folder" and then "New Query".
4. **Define Query Criteria**: Name your query and define the criteria based on your requirements. You can specify things like work item type, state, assigned to, iteration path, and more. Azure DevOps provides a query builder interface to help you construct your query with various filter options.
5. **Save the Query**: Once you've defined your query criteria, save the query. You can also choose to run it immediately to see the results.
6. **Optional: Share the Query**: If you want to share the query with your team members, you can do so by setting permissions on the query folder.
7. **Run and Manage Queries**: After creating the query, you can run it to see the results. You can also modify or delete the query as needed.

Chapter19: Customize project using inherited process









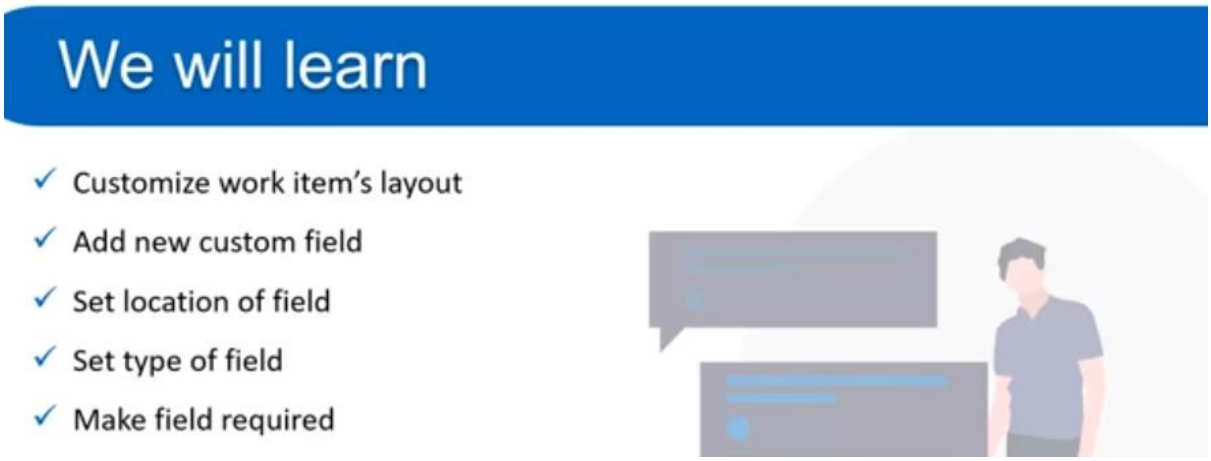
Chapter20: Create custom work item and fields in azure Devops

1. **Identify the Purpose**: Determine why you need this custom work item and what specific information you want to capture with it. This could be a new type of task, issue, feature, or any other work item relevant to your project.
2. **Access Settings**: Log in to your project management tool with appropriate permissions that allow you to create custom work items and fields.
3. **Navigate to Customization**: Find the customization settings within your project management tool. This is usually found in the admin or settings section.
4. **Create a New Work Item Type**: Depending on the tool you're using, there should be an option to create a new work item type. Give it a descriptive name, such as "Feature Request" or "Bug Report".
5. **Define Custom Fields**: Once you've created the work item type, you'll want to define the specific fields you want to include. These could be text fields, dropdowns, checkboxes, dates, etc. Common fields might include:
    - Title/Summary
    - Description
    - Priority
    - Status
    - Assignee
    - Due Date
    - Labels/Tags
    - Estimated Effort
    - etc.
6. **Set Field Properties**: For each field, set properties such as whether it's required, its default value, whether it's editable after creation, etc.

7. **Save and Apply Changes**: Once you've defined your custom work item type and fields, save your changes. Depending on the tool, you may need to apply these changes to specific projects or boards.
8. **Test**: Create a sample work item using your new custom type and ensure that all the fields are appearing as expected and that you can enter and save data correctly.
9. **Iterate**: Based on feedback and evolving project needs, you might need to iterate on your custom work item and field definitions. Make adjustments as necessary.
10. **Train Users**: If you're working with a team, make sure everyone understands the purpose of the new work item type and how to use it effectively.

Chapter21: Customize fields of a work item

Customizing fields of a work item typically depends on the specific project management or issue tracking system you're using.



Chapter22: Add column in kanban board

Chapter23: Update fields from the cards

Chapter24: Define style rules to highlight cards

Chapter25: Add color tag to work item card

Chapter26: Repo and version control system

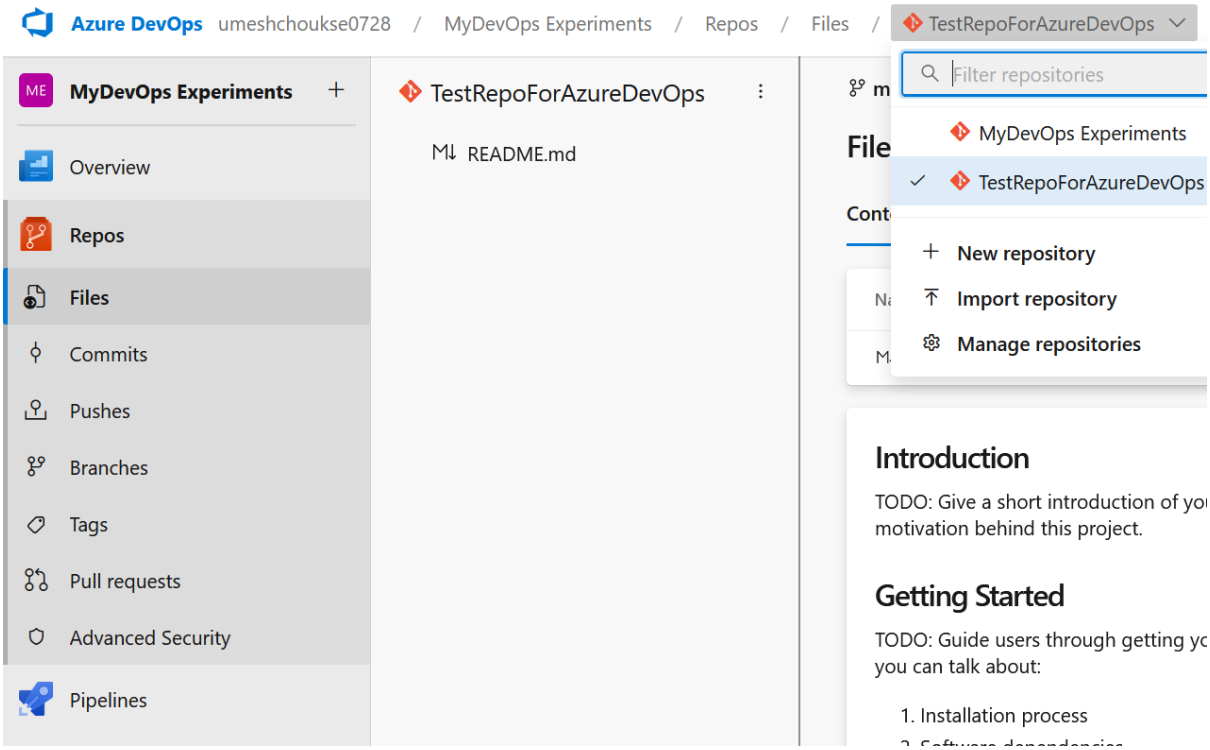
- What is repo- repo is a container that manage your code in systematic way.
- What is VCS- VCS is a system which track each and every change in a file.
- What is use of repo- store code of project
- Who should use repo- Everyone who has data
- What is the type of version control in repo- Central and distributed

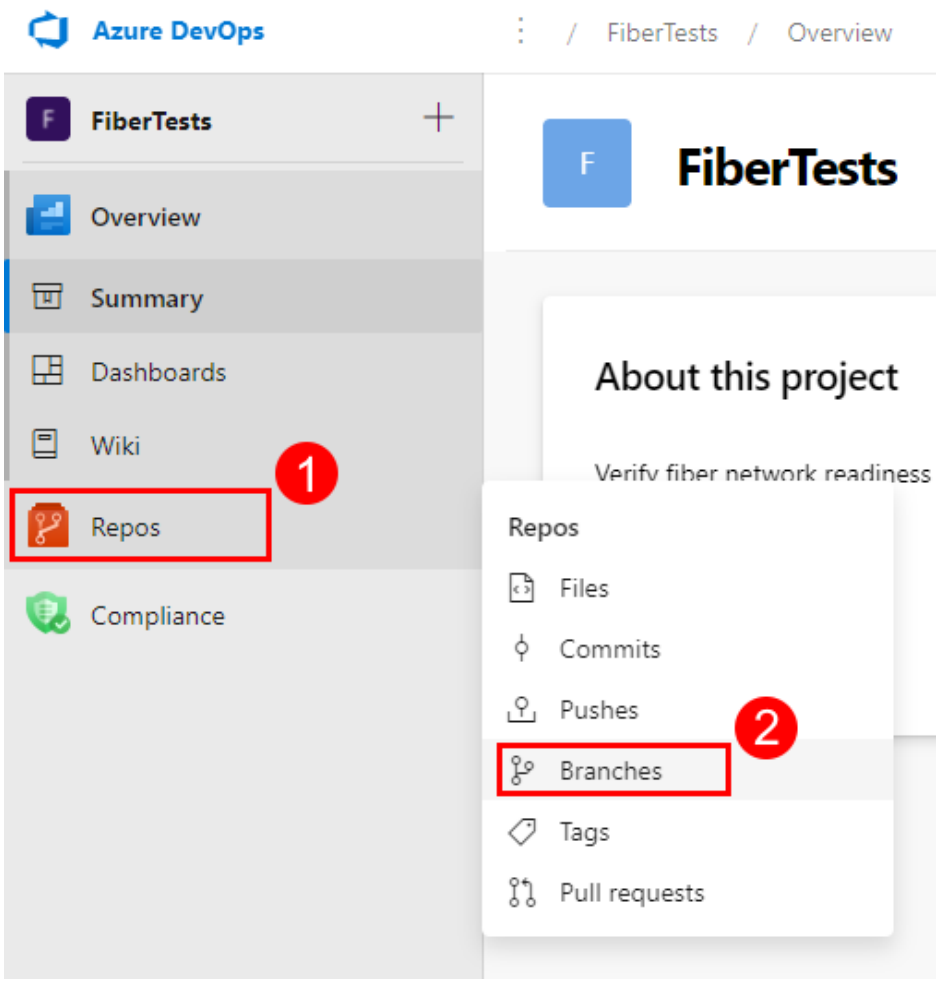
Chapter27: GIT vs TFVC

- GIT vs TFVC
- How GIT work #Working of distributed system.
- How TFVC work # Working of locking mechanism.

Chapter28: Create project with Git or TFVC

Create a project



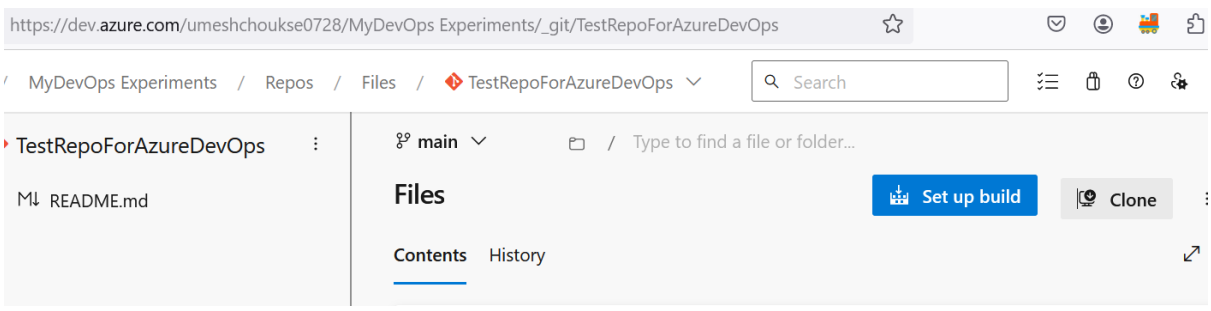


Support link: https://learn.microsoft.com/en-us/azure/devops/repos/git/create-branch?view=azure-devops&tabs=browser

Chapter29: Setup Git on local system

```
C:\Users\juhi3>git --version
git version 2.41.0.windows.2
```

Chapter30: Clone Azure DevOps Repo




```
juhi3@LAPTOP-KR8P2IGH MINGW64 ~/Desktop
$ git clone https://umeshchoukse0728@dev.azure.com/umeshchoukse0728/MyDevOps%20Experiments/_git/TestRepoForAzureDevOps
Cloning into 'TestRepoForAzureDevOps'...
remote: Azure Repos
remote: Found 3 objects to send. (15 ms)
Unpacking objects: 100% (3/3), 734 bytes | 38.00 KiB/s, done.
```

Chapter31: Git Commit, Push & Pull in Azure DevOps Repo

Chapter32: Working directly on azure Devops server repository

Chapter33: Branch in azure Devops repo

- What are branches in GIT
- Branching strategy
- Why we need branching
- Different way to create branch
- POC: proof of concept
- WOW: Way of working

Chapter34: Create new branch in azure DevOps using server repository

Chapter35: Create new branch in Azure DevOps using local repository

Chapter36: Creating pull request for Azure DevOps pipeline.

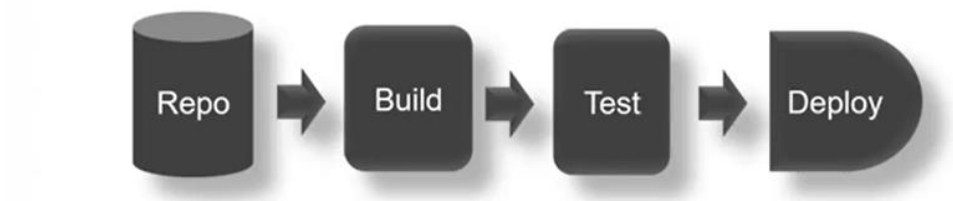
Chapter37: Pipeline as code

- What is pipeline in Azure DevOps
- What process are available in Azure Devops Pipeline
- What type of application and programming language can we used in pipeline #All
- where I can store my project code in pipeline #must be in VCS
- where can I deploy my project code
- what is CICD
- when CI and CI will be trigger

- ✓ Pipeline in DevOps is a set of process (automated or can be triggered manually) which is used to make available your project code to users.
- ✓ Azure Pipelines is a cloud service that you can use to automatically build and test your code project and make it available to other users.

Process in pipeline

- ✓ Pipeline in azure DevOps is designed as per project need.
- ✓ We can increase or decrease number of steps while designing the pipeline.



Application type & Language

- ✓ Azure devops works with any programming language and any platform.
- ✓ Ex – .Net, .Net Core, Python, C#, Java, JavaScript, Angular, PHP, Ruby, C++ etc..

Source Code

- ✓ Your project source code must be in a version control system.
- ✓ Azure Pipelines integrates with

  GitHub,

  GitHub Enterprise,

  Azure Repos Git & TFVC,

  Bitbucket Cloud,

  and Subversion.

Deployment target

- ✓ Azure devops can be used to deploy your code on multiple targets.
- ✓ Example –

  Container registries,

  VM (Virtual machine),

  Azure Service,

  Any cloud target,

  Any on-premises

Pipeline pricing

- ✓ For public repo – Always Free
- ✓ For private repo – 30 hours free every month

CI & CD in Pipelines

- ✓ **CI (Continuous Integration):** CI is used to automate tests and build for the projects. It is used to find bugs or other build issue in early phase.
- ✓ **CD (Continuous Delivery):** CD is used to automatically deploy and test code in multiple stages (environments like dev, qa, staging etc.) to maintain the quality.

When CI & CD are triggered

- ✓ **Manually:** Anytime you can run the CI, CD pipeline to build and test your code
- ✓ **Automatically:**
  - ✓ On each commit
  - ✓ On a fix interval
  - ✓ At a particular time

Chapter38: Create Build Pipeline in Azure DevOps

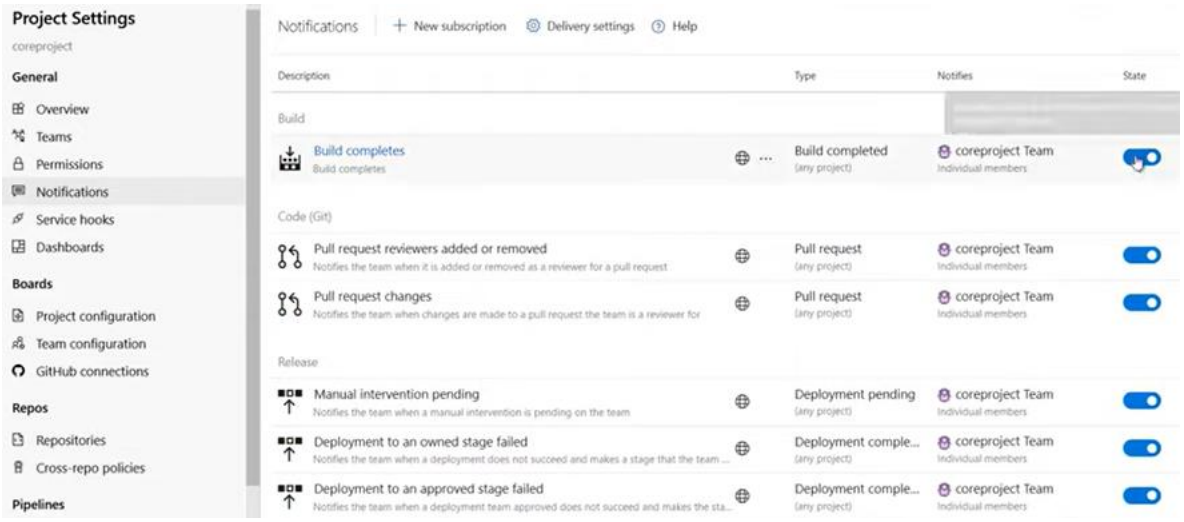
- ✓ How to create new build (CI) pipeline.
- ✓ Build project automatically.
- ✓ Get success/ error notification on email.
- ✓ View build logs to fix the error.

```
trigger:
- master

pool:
  vmImage: 'ubuntu-latest'

variables:
  buildConfiguration: 'Release'

steps:
- script: dotnet build --configuration $(buildConfiguration)
  displayName: 'dotnet build $(buildConfiguration)'
```



Chapter39: Important terms used during CI pipeline

Environment

- ✓ An environment is the place where we deploy our application.
- ✓ An environment is a collection of resources.
- ✓ Example – VMs, Container, Web app etc.
- ✓ A release pipeline can deploy the code on one or more VMs (Environments) after the build pipeline is completed.

Trigger

- ✓ A trigger is a setup that tells the pipeline when to run.
- ✓ We can configure a pipeline when-
  - ✓ New push in repo
  - ✓ At scheduled time
  - ✓ Upon completion of another build.

Artifacts

- ✓ An artifact is a collection of files or packages which are created by a build run.
- ✓ These artifacts are then made available for the next task i.e. deployment

Job

- ✓ A job represents an execution boundary of a set of steps.
- ✓ All of 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 stage and two jobs.

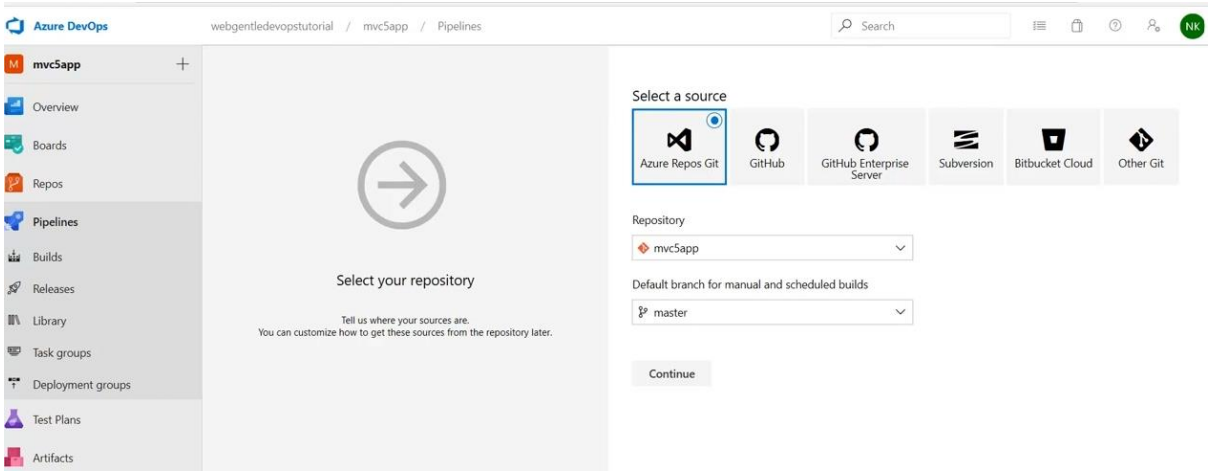
Agent

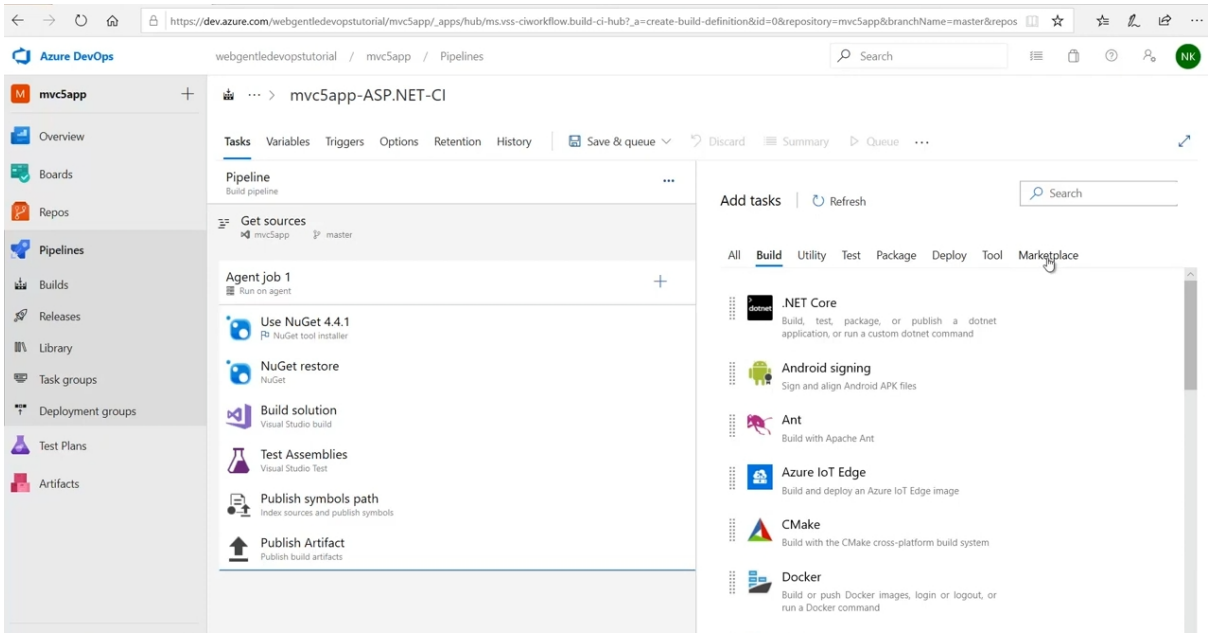
- ✓ When your build or deployment runs, the system begins one or more jobs. An agent is installable software that runs one job at a time.
- ✓ To build your code or deploy your software using Azure Pipelines, you need at least one agent.
- ✓ In Azure DevOps we can use two types of agent
  - ✓ Microsoft-hosted agent
  - ✓ Self-hosted agent

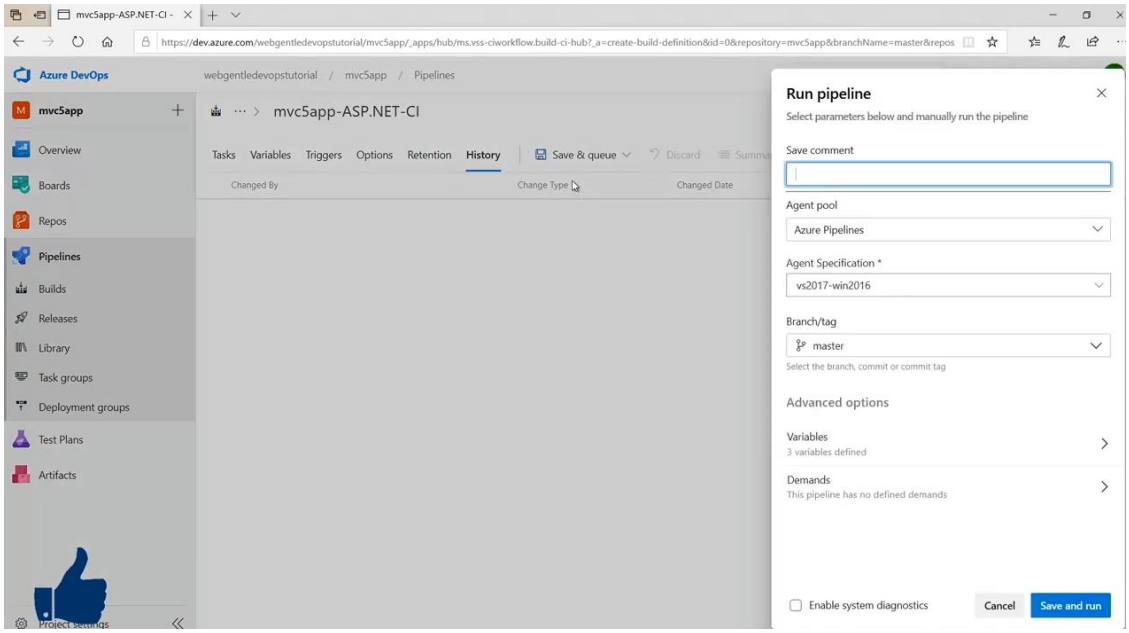
Approvals

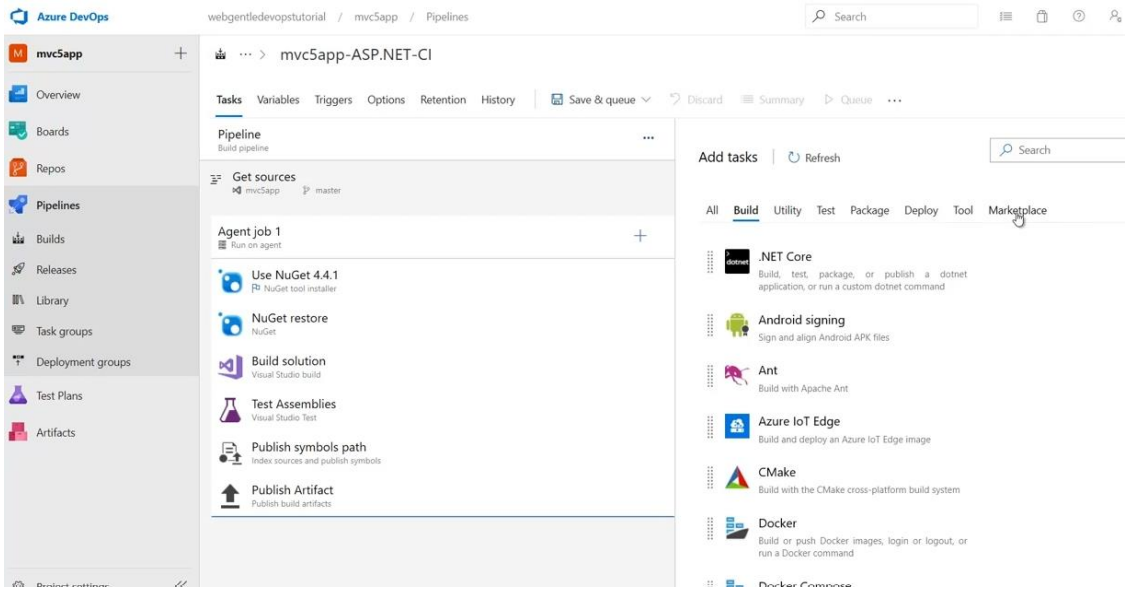
- ✓ Approvals are a set of validations which are required before a deployment can be performed.
- ✓ Example – *Getting permission from someone in a team before the deployment on production*

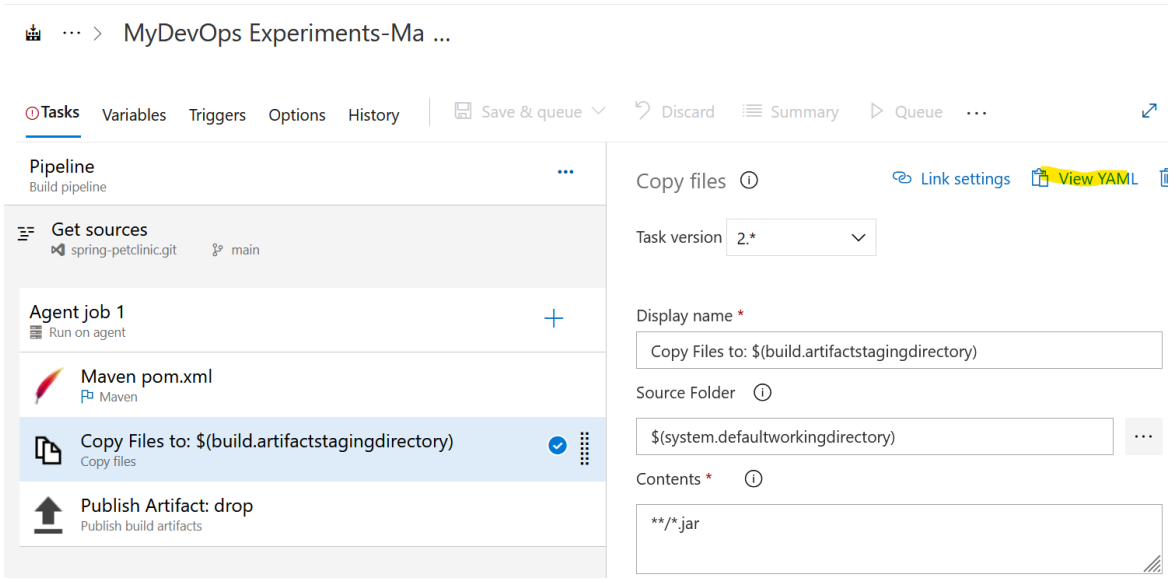
Chapter39: Creating build pipeline by classic editor



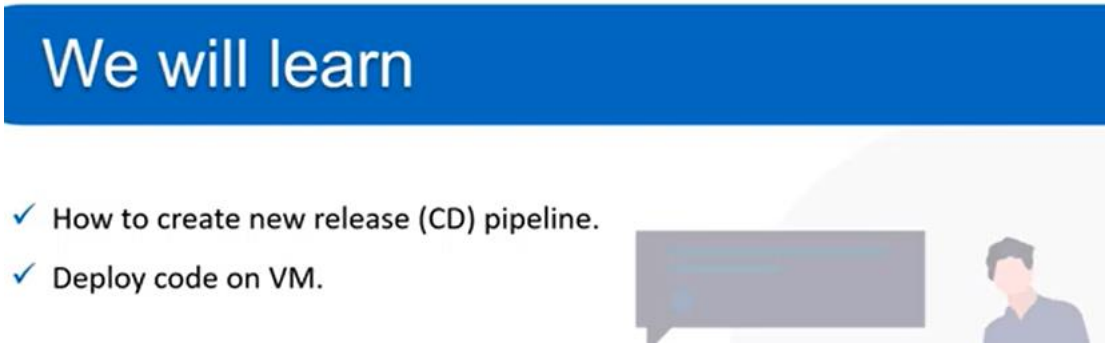



Azure DevOps
webgentledevopstutorial / mvc5app / Pipelines
mvc5app
Overview
Boards
Repos
Pipelines
Builds
Releases
Library
Task groups
Deployment groups
Test Plans
Artifacts
mvc5app-ASP.NET-CI
Tasks Variables Triggers Options Retention History
Save & queue
Discard
Changed By
Change Type
Changed Date
Run pipeline
Select parameters below and manually run the pipeline
Save comment
Agent pool
Azure Pipelines
Agent Specification *
vs2017-win2016
Branch/tag
master
Select the branch, commit or commit tag
Advanced options
Variables
3 variables defined
Demands
This pipeline has no defined demands
Enable system diagnostics
Cancel
Save and run

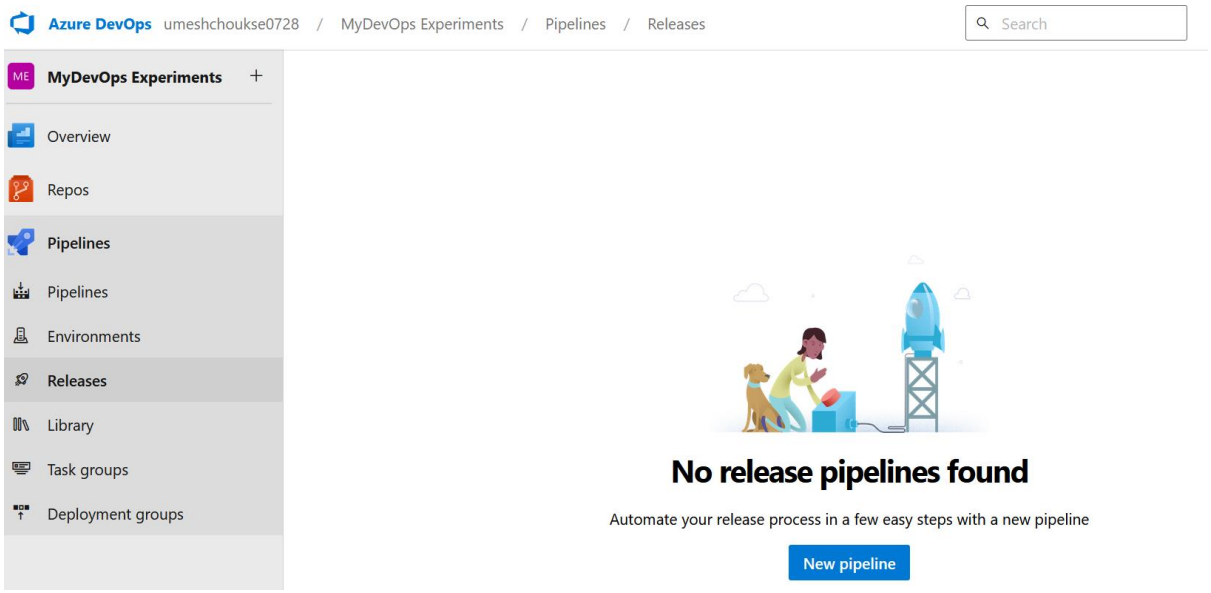

Azure DevOps
webgentledevopstutorial / mvc5app / Pipelines
Search
mvc5app
Overview
Boards
Repos
Pipelines
Builds
Releases
Library
Task groups
Deployment groups
Test Plans
Artifacts
mvc5app-ASP.NET-CI
Tasks Variables Triggers Options Retention History
Save & queue
Discard
Summary
Queue
Pipeline
Build pipeline
Get sources
mvc5app master
Agent job 1
Run on agent
Use NuGet 4.4.1
NuGet tool installer
NuGet restore
NuGet
Build solution
Visual Studio build
Test Assemblies
Visual Studio Test
Publish symbols path
Index sources and publish symbols
Publish Artifact
Publish build artifacts
Add tasks
Refresh
Search
All Build Utility Test Package Deploy Tool Marketplace
.NET Core
Build, test, package, or publish a dotnet application, or run a custom dotnet command
Android signing
Sign and align Android APK files
Ant
Build with Apache Ant
Azure IoT Edge
Build and deploy an Azure IoT Edge image
CMake
Build with the CMake cross-platform build system
Docker
Build or push Docker images, login or logout, or run a Docker command


MyDevOps Experiments-Ma ...
Tasks
Variables
Triggers
Options
History
Save & queue
Discard
Summary
Queue
Pipeline
Build pipeline
Get sources
spring-petclinic.git
main
Agent job 1
Run on agent
Maven pom.xml
Maven
Copy Files to: $(build.artifactstagingdirectory)
Copy files
Publish Artifact: drop
Publish build artifacts
Copy files
Link settings
View YAML
Task version
2.*
Display name *
Copy Files to: $(build.artifactstagingdirectory)
Source Folder
$(system.defaultworkingdirectory)
Contents *
**/*.jar

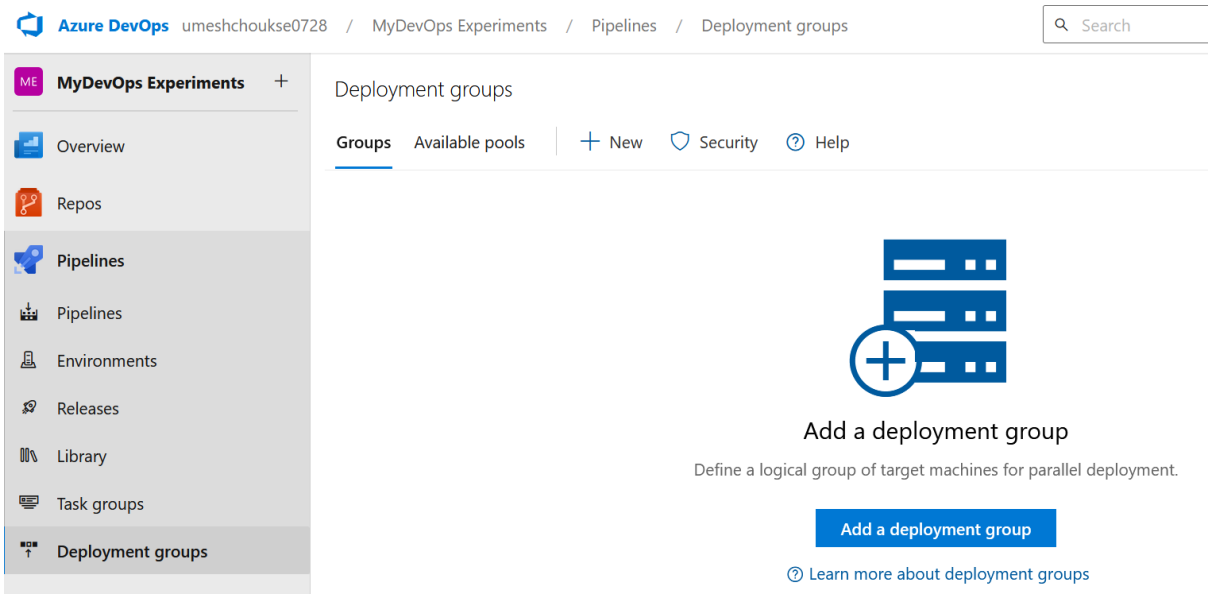
Chapter41: Release Pipeline



Deploying the code artificed by release pipeline.



Deployment group: Deployment group is connection between Azure Devops and virtual machine.



Deployment groups > test

Details | Targets | Save | Share | Security | Help

Deployment group name

test

Description

test

Deployment pool

MyDevOps Experiments-test | Manage

Type of target to register:

Linux | System prerequisites

Registration script

```
mkdir azagent;cd azagent;curl -fkSL -o vstsagent.tar.gz https://
vstsagentpackage.azureedge.net/agent/3.232.3/vsts-agent-linux-
x64-3.232.3.tar.gz;tar -zxvf vstsagent.tar.gz; if [ -x "$
(command -v systemctl)" ]; then ./config.sh --deploymentgroup --
deploymentgroupname "test" --acceptteeeula --agent $HOSTNAME --
url https://dev.azure.com/umeshchoukse0728/ --work _work --
projectname 'MyDevOps Experiments' --runasservice; sudo ./svc.sh
install; sudo ./svc.sh start; else ./config.sh --deploymentgroup
--deploymentgroupname "test" --acceptteeeula --agent $HOSTNAME
--url https://dev.azure.com/umeshchoukse0728/ --work _work --
projectname 'MyDevOps Experiments'; ./run.sh; fi
```

Use a personal access token in the script for authentication

Copy script to the clipboard

Run from an administrator command prompt

```
Scanning for tool capabilities.
Connecting to the server.
Enter deployment group tags for agent? (Y/N) (press enter for N) >
Successfully added the agent
Testing agent connection.
2020-01-07 17:09:37Z: Settings Saved.
Enter User account to use for the service (press enter for NT AUTHORITY\SYSTEM) >
```

Deployment groups

Groups | Available pools | New | Security | Help

Name	Target status	Deployment status
mvc5appgroup	1 Online	1 Never deployed

All pipelines > New release pipeline

Pipeline | Tasks | Variables | Retention | Options | History

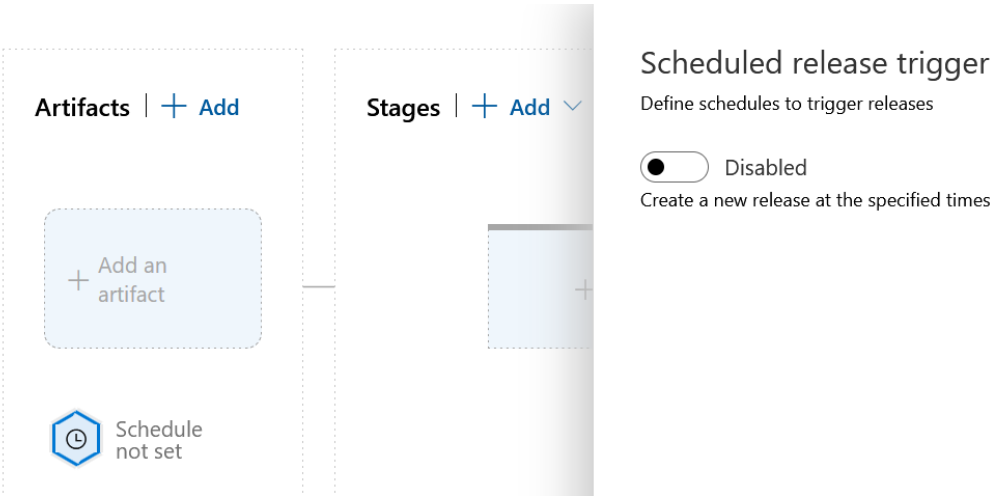
You cannot save a release pipeline that contains zero stages.

Artifacts | Add

Add an artifact

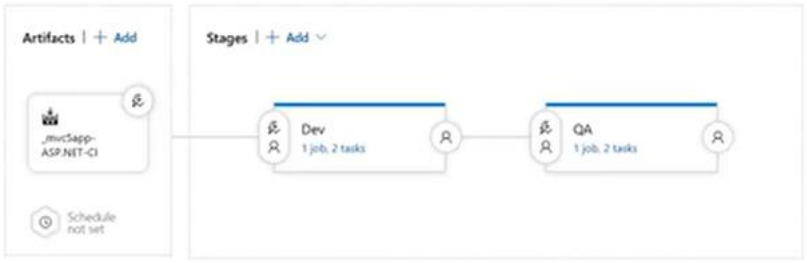
Stages | Add

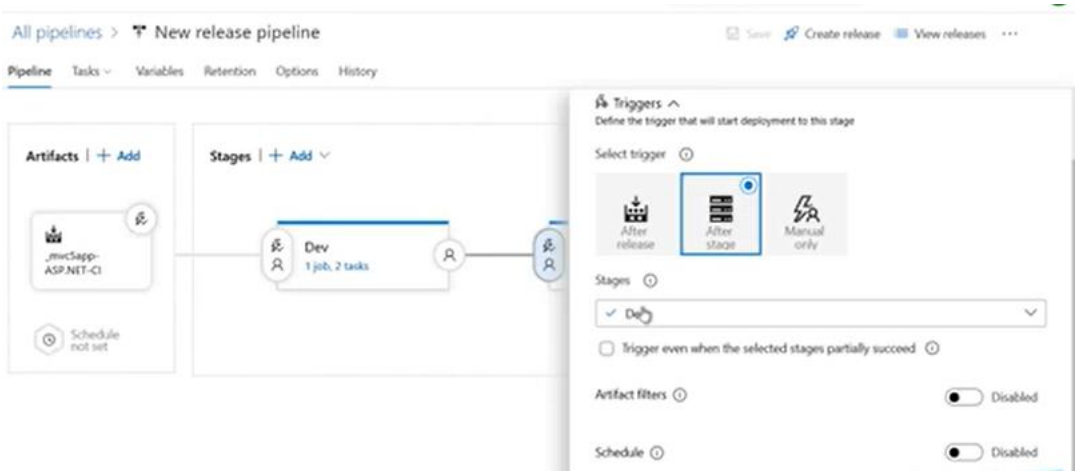
Add a stage

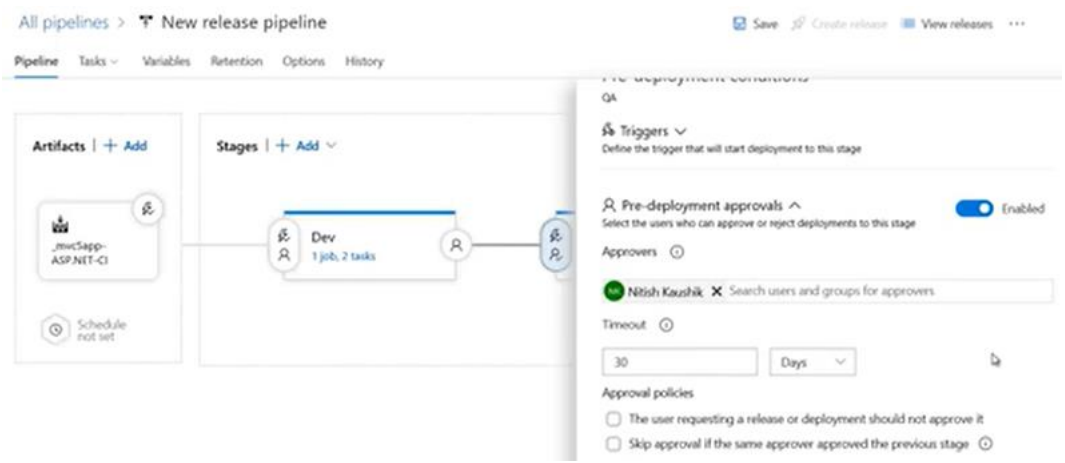


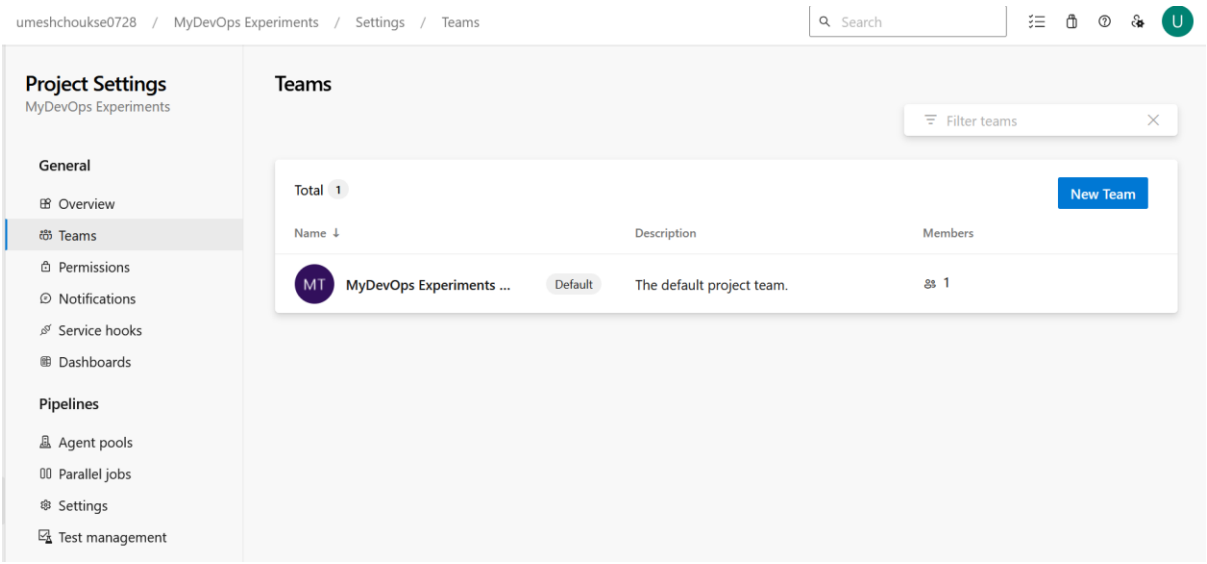
Chapter42: Multi stage release pipeline

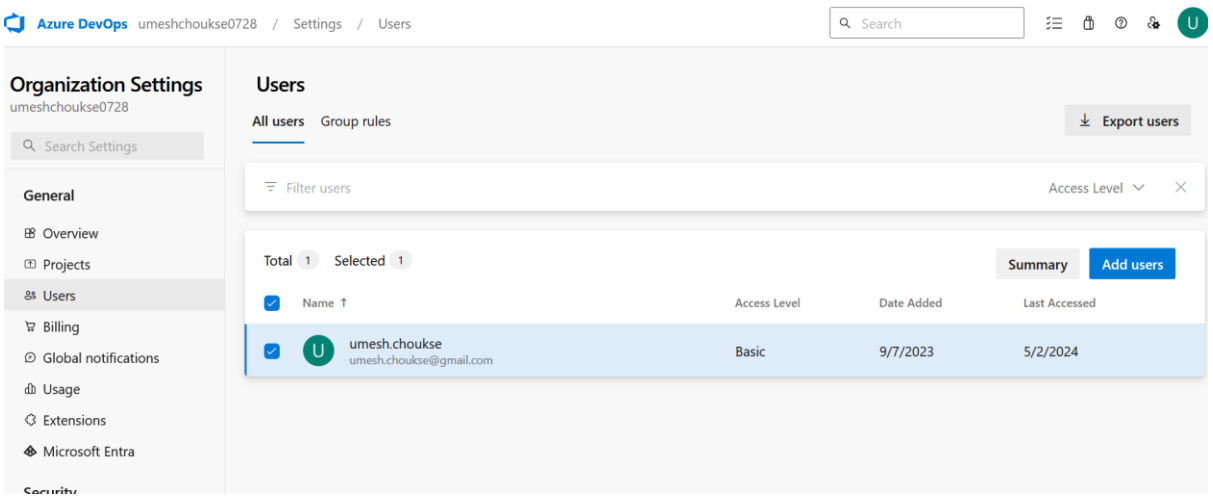
Multiple stages are required to deploy and test the application on multiple servers. A very basic example could be creating a release pipeline that deploys the code on Dev env, QA env, Staging env, Production env






All pipelines > New release pipeline
Save
Create release
View releases
Pipeline
Tasks
Variables
Retention
Options
History
Artifacts | + Add
_mvcSapp-ASP.NET-CI
Schedule not set
Stages | + Add
Dev
1 job, 2 tasks
QA
Triggers
Define the trigger that will start deployment to this stage
Pre-deployment approvals
Select the users who can approve or reject deployments to this stage
Enabled
Approvers
Nitish Kaushik
Search users and groups for approvers
Timeout
30
Days
Approval policies
The user requesting a release or deployment should not approve it
Skip approval if the same approver approved the previous stage


umeshchoukse0728 / MyDevOps Experiments / Settings / Teams
Search
Project Settings
MyDevOps Experiments
General
Overview
Teams
Permissions
Notifications
Service hooks
Dashboards
Pipelines
Agent pools
Parallel jobs
Settings
Test management
Teams
Filter teams
Total 1
New Team
Name
Description
Members
MT
MyDevOps Experiments ...
Default
The default project team.
1


Azure DevOps
umeshchoukse0728
/
Settings
/
Users
Search
Organization Settings
umeshchoukse0728
Search Settings
General
Overview
Projects
Users
Billing
Global notifications
Usage
Extensions
Microsoft Entra
Security
Users
All users
Group rules
Export users
Filter users
Access Level
Total 1
Selected 1
Summary
Add users
Name
Access Level
Date Added
Last Accessed
umesh.choukse
umesh.choukse@gmail.com
Basic
9/7/2023
5/2/2024