

# Devops – Final Assessment

## Section 1: Multiple-Choice Questions (MCQs)

1. What does WSL stand for in the context of Windows?

Ans: c. Windows Subsystem for Linux

2. What is the primary goal of continuous integration (CI) in DevOps?

Ans: b. Frequent integration of code changes

3. In the Linux command line, what does the cd command do?

Ans: b. Change the working directory

4. Which of the following is not a Linux distribution?

Ans: c. Docker

5. What is Docker primarily used for in DevOps and containerization?

Ans: c. Packaging and deploying applications in containers

6. What is the primary purpose of Azure DevOps?

Ans: b. Software development and delivery

7. Which components are part of Azure DevOps?

Ans: c. Azure Boards and Azure Pipelines

8. How does Azure DevOps support version control in software development?

Ans: b. It tracks changes in source code and manages versions.

9. In Linux, what is the primary role of the root user?

Ans: c. Administrative tasks with superuser privileges

10. In Azure DevOps, which component is used to define, build, test, and deploy applications? Ans: c. Azure Pipelines.

## Section 2: Labs

### Lab 1: File and Directory Management

Objective: Practice basic file and directory management commands.

Tasks:

1. Create a directory called "lab1" in your home directory.
2. Inside "lab1," create a text file named "sample.txt" with some content.
3. Make a copy of "sample.txt" and name it "sample\_copy.txt."
4. Rename "sample\_copy.txt" to "new\_sample.txt."
5. List the files in the "lab1" directory to confirm their names

Ans:

1. To create a directory, use mkdir command.

Eg: mkdir lab1

2. Now, list the directories using ls, move to lab1 using cd, create a file using touch.

Eg: ls

Eg: cd lab1

Eg: touch sample.txt

3. Make a copy of sample.txt and name it sample\_copy.txt using cp

Eg: cp sample.txt sample\_copy.txt

4. Rename it to new\_sample.txt using mv

Eg: mv sample\_copy.txt new\_sample.txt

5. List using ls.

```
mridhul@MSI: /mnt/c/Users/l | x + v
C:\Users\Mridhul>wsl
mridhul@MSI:/mnt/c/Users/Mridhul$ ls
-1.14-windows.xml
Application Data
Cookies
Local Settings
My Documents
NTUSER.DAT
NTUSER.DAT[a2332f18-cdbf-11ec-8680-002248483d79].TM.blf
NTUSER.DAT[a2332f18-cdbf-11ec-8680-002248483d79].TMContainer000000000000000001.regtrans-ms
NTUSER.DAT[a2332f18-cdbf-11ec-8680-002248483d79].TMContainer000000000000000002.regtrans-ms
mridhul@MSI:/mnt/c/Users/Mridhul$ mkdir lab1
mridhul@MSI:/mnt/c/Users/Mridhul$ cd lab1
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ touch sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ cp sample.txt sample_copy.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ mv sample_copy.txt new_sample.txt
mv: command not found
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ mv sample_copy.txt new_sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ ls
new_sample.txt sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ |
```

## Lab 2: Permissions and Ownership

Objective: Understand and manage file permissions and ownership.

Tasks:

- 1.Create a new file named "secret.txt" in the "lab2" directory.
- 2.Set the file permissions to allow read and write access only to the owner.
- 3.Change the owner of "secret.txt" to another user.
- 4.Verify the new permissions and owner using the ls -l and ls -n commands.

Ans:

1. Create new file using touch

Eg: touch secret.txt

2. Set file permissions to allow read, write access using chmod 600

Eg: chmod 600 secret.txt

3. Change owner using chown

Eg: chown mahasri:mahasri secret.txt

4. Verify new permission using ls -l , ls -n

Eg: ls-l secret.txt and ls-n secret.txt

```
mridhul@MSI: /mnt/c/Users/l | + - x
'My Documents'
NTUSER.DAT
NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TH.blf
NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TH.Container000000000000000001.regtrans-ms
NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TH.Container000000000000000002.regtrans-ms
mridhul@MSI: /mnt/c/Users/Mridhul$ mkdir lab1
mridhul@MSI: /mnt/c/Users/Mridhul$ cd lab1
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ touch sample.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ cp sample.txt sample_copy.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ mv sample_copy.txt new_sample.txt
my: command not found
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ mv sample_copy.txt new_sample.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ ls
new_sample.txt sample.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ touch secret.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ : touch secret.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ ls
new_sample.txt sample.txt secret.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ chmod 600 secret.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ chmod mridhul:mridhul secret.txt
chmod: invalid mode: 'mridhul:mridhul'
Try 'chmod --help' for more information.
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ chown mridhul:mridhul secret.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ ls-l secret.txt
ls-l: command not found
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ ls -l secret.txt
-rwxrwxrwx 1 mridhul mridhul 0 Oct 24 11:27 secret.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ ls -n secret.txt
-rwxrwxrwx 1 1000 1000 0 Oct 24 11:27 secret.txt
mridhul@MSI: /mnt/c/Users/Mridhul/lab1$ |
```

### Lab 3: Text Processing with Command Line Tools

Objective: Practice text processing using command-line tools.

Tasks:

1. Create a text file with some random text in the "lab3" directory.
2. Use the grep command to search for a specific word or pattern in the file.
3. Use the sed command to replace a word or phrase with another in the file.
4. Use the wc command to count the number of lines, words, and characters in the file.

Ans:

1. Create text file using touch

Eg: touch sample.txt

2. Use grep to search for a word

Eg: grep "kanini" sample.txt

3. Use sed to replace a word

Eg: sed -i 's/Mahasri/Maha/g' sample.txt

4. Use wc to count no.of lines

Eg: wc sample.txt

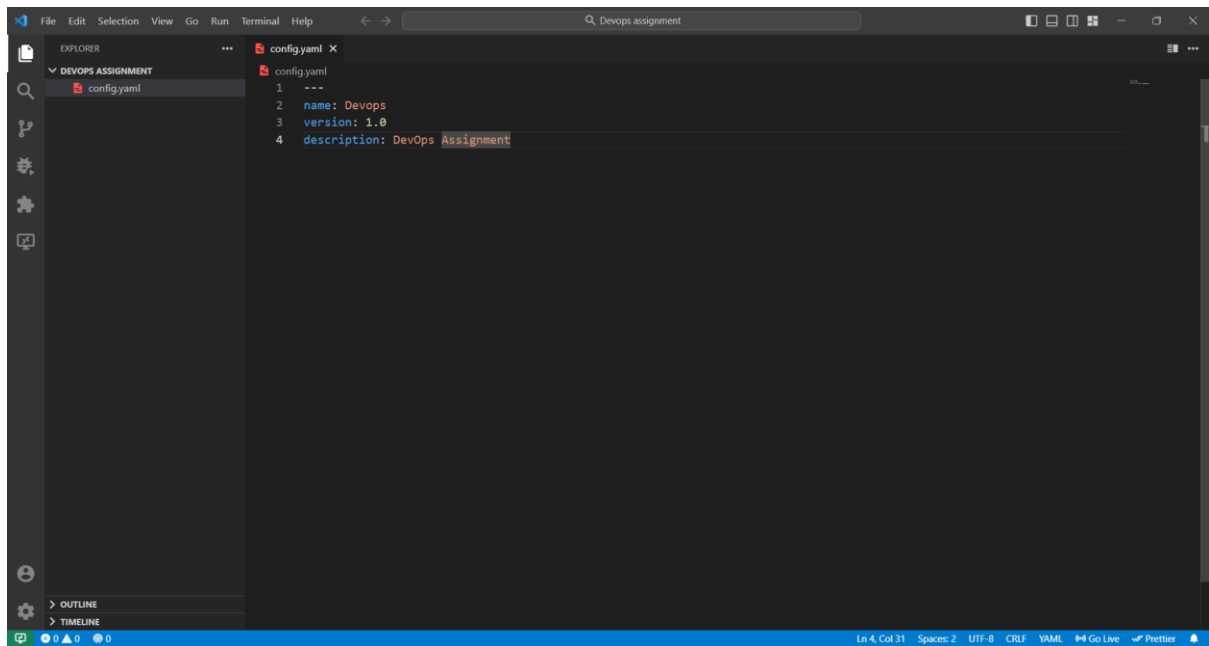
```
mridhul@MSI: /mnt/c/Users/l x + v
mridhul@MSI:/mnt/c/Users/Mridhul$ mkdir lab1
mridhul@MSI:/mnt/c/Users/Mridhul$ cd lab1
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ touch sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ cp sample.txt sample_copy.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ mv sample_copy.txt new_sample.txt
mv: command not found
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ mv sample_copy.txt new_sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ ls
new_sample.txt sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ touch secret.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ : touch secret.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ ls
new_sample.txt sample.txt secret.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ chmod 600 secret.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ chmod mridhul:mridhul secret.txt
chmod: invalid mode: 'mridhul:mridhul'
Try 'chmod --help' for more information.
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ chown mridhul:mridhul secret.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ ls-l secret.txt
ls-l: command not found
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ ls -l secret.txt
-rwxrwxrwx 1 mridhul mridhul 0 Oct 24 11:27 secret.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ ls -n secret.txt
-rwxrwxrwx 1 1000 1000 0 Oct 24 11:27 secret.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ touch sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ grep "kanini" sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ sed -i "s/mridhul/Mridhul/g" sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ wc sample.txt
0 0 0 sample.txt
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ |
```

## Lab 4: Creating a Simple YAML File

Objective: Create a basic YAML configuration file.

Task:

- 1.Create a YAML file named "config.yaml."
- 2.Define key-value pairs in YAML for a fictitious application, including name, version, and description.
- 3.Save the file.
- 4.Validate that the YAML file is correctly formatted.



```
mridhul@MSI:/mnt/c/Users/Mridhul/lab1$ cat config.yaml
---
name: Devops
version: 1.0
description: DevOps Assignmentmridhul@MSI:/mnt/c/Users/Mridhul/lab1$ |
```

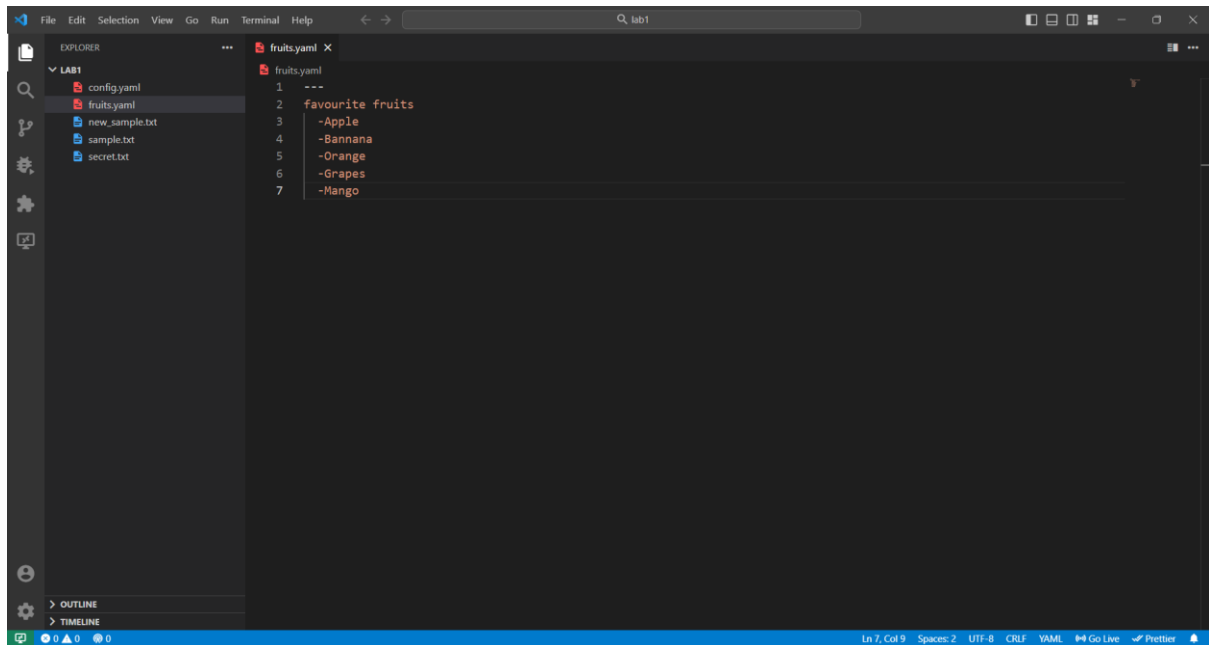
### Lab 5: Working with Lists in YAML:

Objective: Practice working with lists (arrays) in YAML.

Task:

- 1.Create a YAML file named "fruits.yaml."
- 2.Define a list of your favorite fruits using YAML syntax.
- 3.Add items from the list.
- 4.Save and validate the YAML file.

ANS



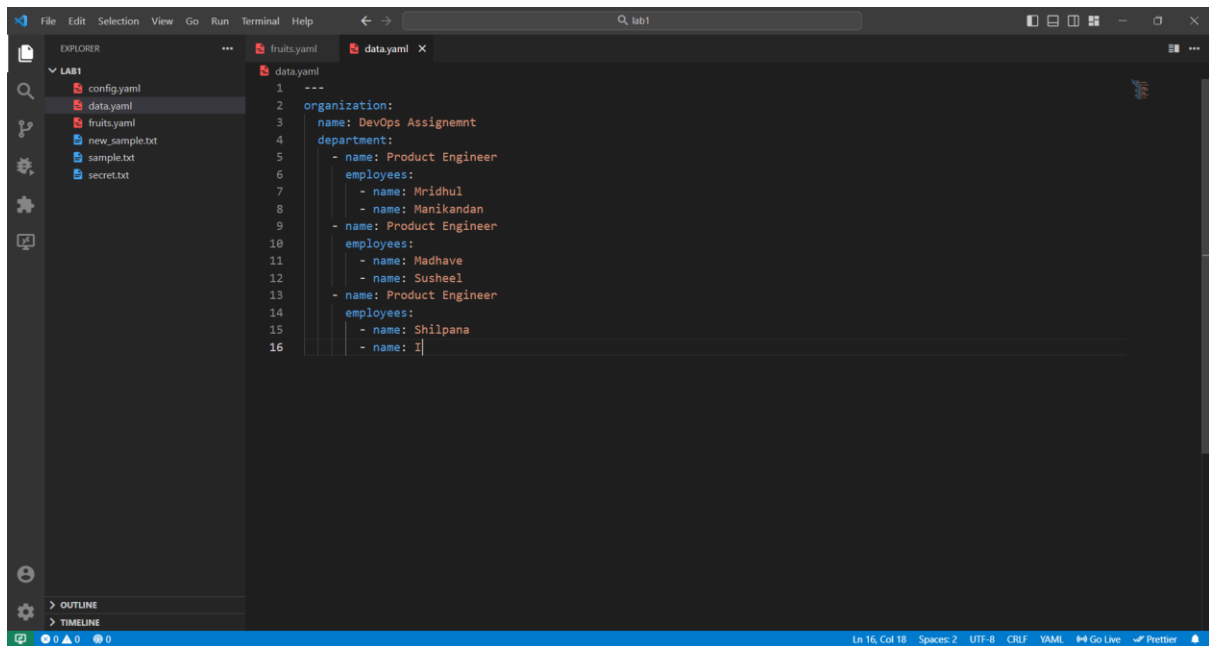
```
description: DevOps Assignmentmridhul@MSI:/mnt/c/Users/Mridhul/lab1$ cat fruits.yaml
---
favourite fruits
-Apple
-Bannana
-Orange
-Grapes
-Mangomridhul@MSI:/mnt/c/Users/Mridhul/lab1$ |
```

## Lab 6: Nested Structures in YAML

Objective: Explore nested structures within YAML.

Task:

- 1.Create a YAML file named "data.yaml."
- 2.Define a nested structure representing a fictitious organization with departments and employees.
- 3.Use YAML syntax to add, update, or remove data within the nested structure.
- 4.Save and validate the YAML file.



## Lab 7: Create Classic Azure CI Pipeline for Angular Application

Objective: Set up a classic Azure CI pipeline to build a simple Angular application with unit testing using Jasmine and Karma.

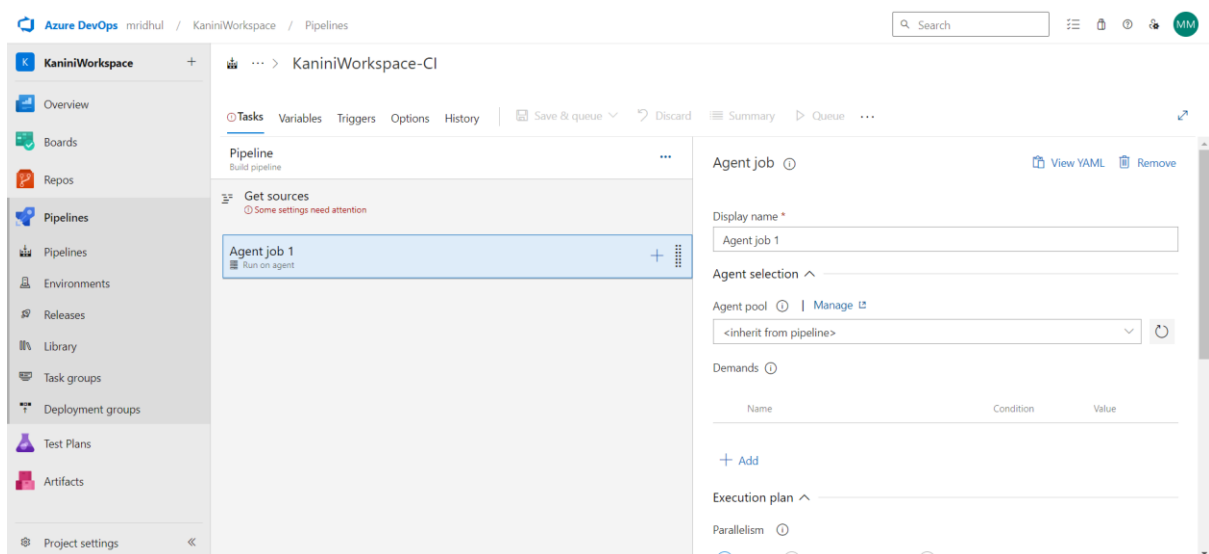
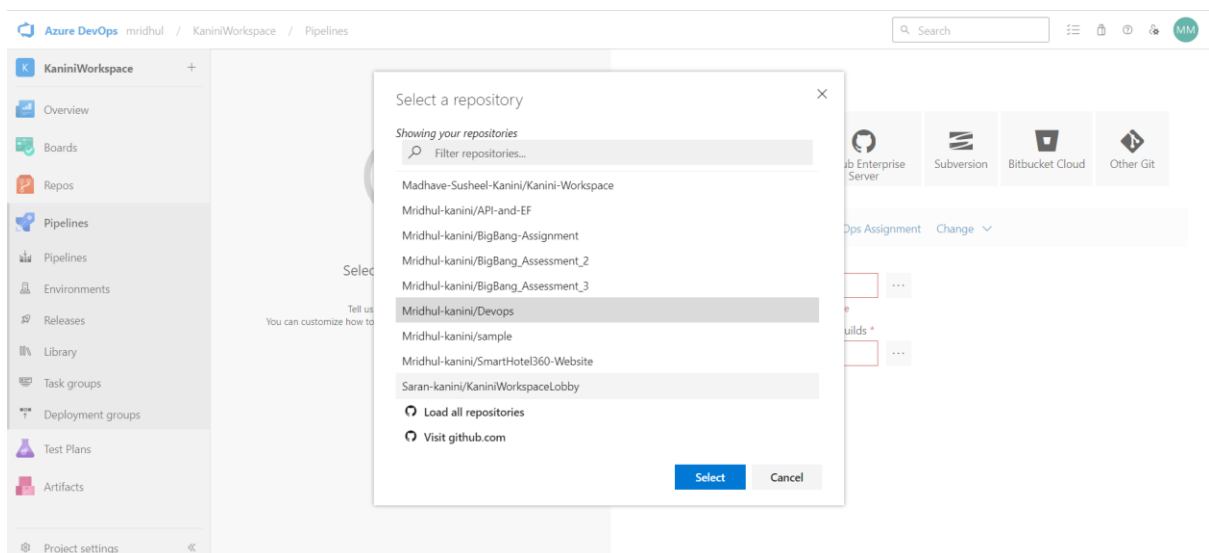
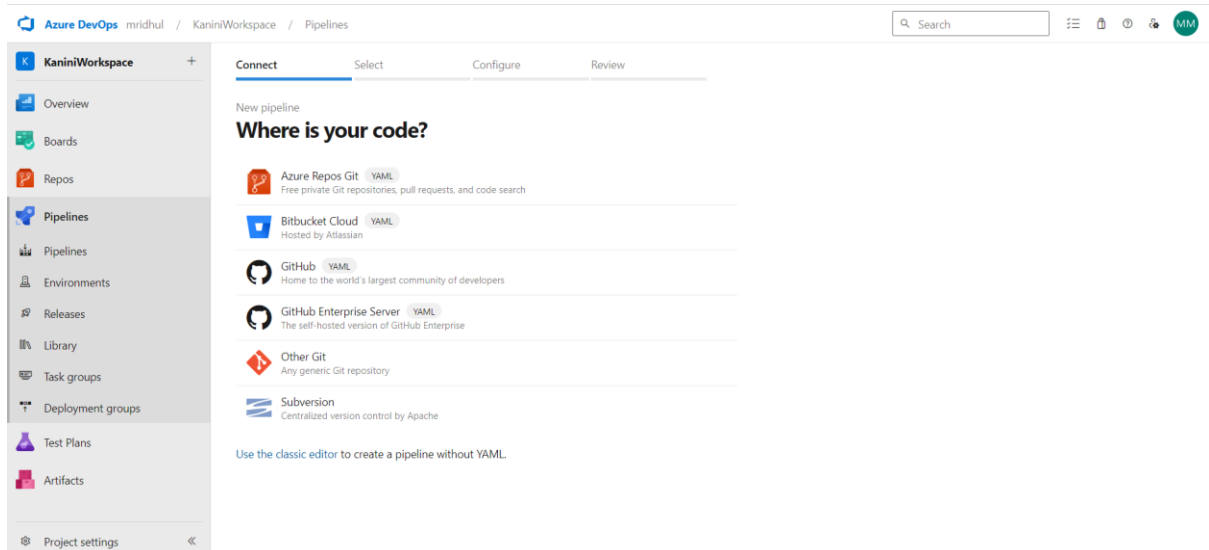
Tasks:

1. Create an Azure DevOps project.
2. Set up a classic CI pipeline to build an Angular application.
3. Configure the pipeline to use Jasmine and Karma for unit testing.
4. Run the pipeline and validate the test results.

Ans:

Create pipeline using classic editor:





KaniniWorkspace

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

KaniniWorkspace-CI

TasksVariablesTriggersOptionsHistory

Save & queueDiscardSummaryQueue

PipelineBuild pipeline

Get sourcesSome settings need attention

Agent job 1Run on agent

npm installnpm

npm customnpm

npm testnpm

npm

Link settingsView YAMLRemove

Task version1.\*

Display name \*npm install

Command \*install

Working folder that contains package.jsonAngular Testing

Custom registries and authentication

Advanced

Control Options

Output Variables

KaniniWorkspace

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

KaniniWorkspace-CI

TasksVariablesTriggersOptionsHistory

Save & queueDiscardSummaryQueue

PipelineBuild pipeline

Get sourcesSome settings need attention

Agent job 1Run on agent

npm installnpm

npm customnpm

npm testnpm

npm

Link settingsView YAMLRemove

Task version1.\*

Display name \*npm custom

Command \*custom

Working folder that contains package.jsonAngular Testing

Command and arguments \*run build

Custom registries and authentication

Control Options

Output Variables

KaniniWorkspace

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

KaniniWorkspace-CI

TasksVariablesTriggersOptionsHistory

Save & queueDiscardSummaryQueue

PipelineBuild pipeline

Get sourcesSome settings need attention

Agent job 1Run on agent

npm installnpm

npm customnpm

npm testnpm

npm

Link settingsView YAMLRemove

Task version1.\*

Display name \*npm test

Command \*custom

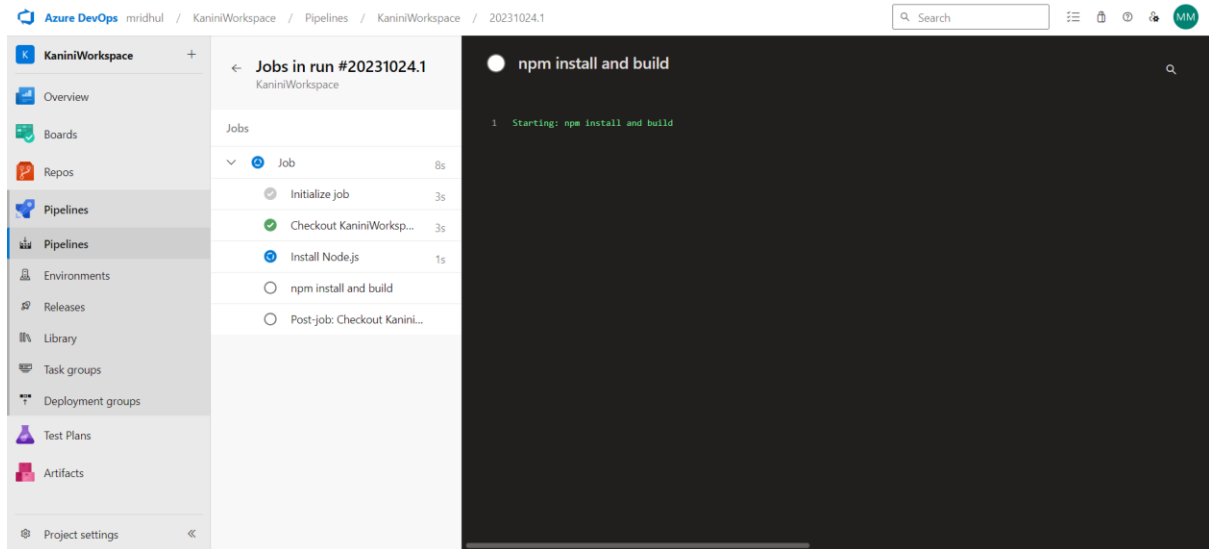
Working folder that contains package.jsonAngular Testing

Command and arguments \*test

Custom registries and authentication

Control Options

Output Variables



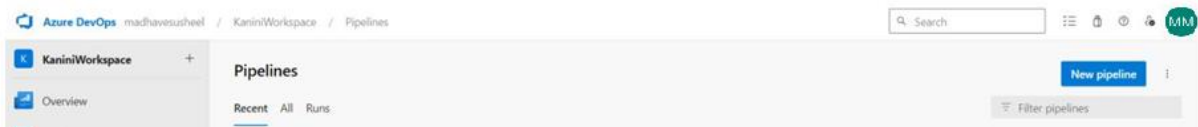
## Lab 8: Create YAML Azure CI Pipeline for React Application

Objective: Create a YAML-based Azure CI pipeline to build a simple React application with unit testing using Enzyme and Jest.

Tasks:

1. Create an Azure DevOps project.
2. Create a YAML-based CI pipeline to build a React application.
3. Configure the pipeline to use Enzyme and Jest for unit testing.
4. Trigger the pipeline and verify the test results.

Create New Pipeline



## Choose Github

The screenshot shows the 'New pipeline' interface in Azure DevOps. The left sidebar contains the 'KaniniWorkspace' menu with options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The main area has tabs for 'Connect', 'Select', 'Configure', and 'Review'. Under the 'Connect' tab, the heading 'Where is your code?' is followed by a list of repository providers: Azure Repos Git, Bitbucket Cloud, GitHub, GitHub Enterprise Server, Other Git, and Subversion. Each provider has a brief description and a 'YAML' icon. At the bottom, there is a link to 'Use the classic editor to create a pipeline without YAML'.

## Choose a repo

The screenshot shows the 'Choose a repo' section of the 'New pipeline' interface. The left sidebar is the same as in the previous screenshot. The main area displays a list of repositories under the heading 'Showing the most recently used repositories where you are a collaborator.' The list includes repositories like 'Mridhul-kanini/Devops', 'Mridhul-kanini/SmartHotel360-Website', 'Saran-kanini/KaniniWorkspaceLobby', 'Mridhul-kanini/BigBang\_Assessment\_3', 'Mridhul-kanini/BigBang\_Assessment\_2', 'Mridhul-kanini/BigBang\_Assignment', 'Mridhul-kanini/API-and-EF', 'Mridhul-kanini/sample', and 'Madhave-Susheel-Kanini/Kanini-Workspace'. Each entry shows the repository name, the user's profile picture, and the date of last activity. A note at the bottom states: 'Showing the most recently used repositories where you are a collaborator. If you can't find a repository, make sure you provide access. You may also select a specific connection.'

## Edit Yaml

The screenshot shows the Azure DevOps web interface for reviewing a pipeline YAML file. The left sidebar contains navigation links for KaniniWorkspace, Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The main area is titled 'Review your pipeline YAML' and shows a preview of the 'azure-pipelines.yml' file. The file content is as follows:

```
3 # Add steps that analyze code, save build artifacts, deploy, and more:
4 # https://docs.microsoft.com/azure/devops/pipelines/languages/javascript
5
6 trigger:
7   - main
8
9 pool:
10  - vmImage: ubuntu-latest
11    name: default
12
13 steps:
14   - task: NodeTool@0
15     inputs:
16       - versionSpec: '10.x'
17       - displayName: 'Install Node.js'
18
19   - script: |
20     npm install -g @angular/cli
21     npm install
```

At the top right of the main area, there are buttons for 'Variables' and 'Save and run'. A warning message at the top states: 'You selected a public repository, but this is not a public project. Go to project settings to change the visibility of the project. Learn more'.

## Save and Run

This screenshot shows the 'Save and run' dialog box overlaid on the same Azure DevOps interface. The dialog box contains the following fields and options:

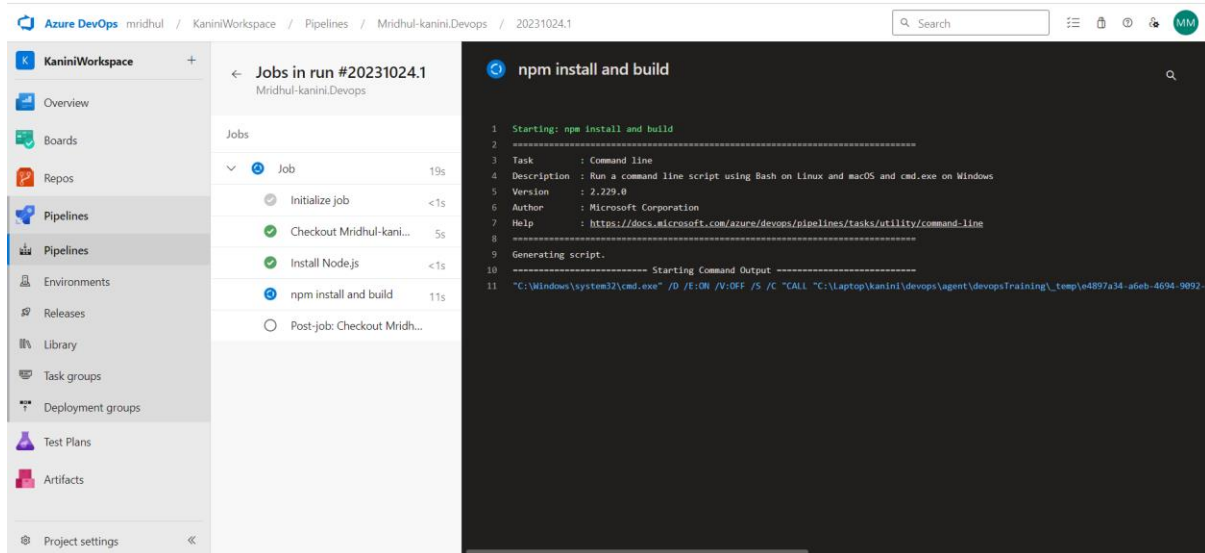
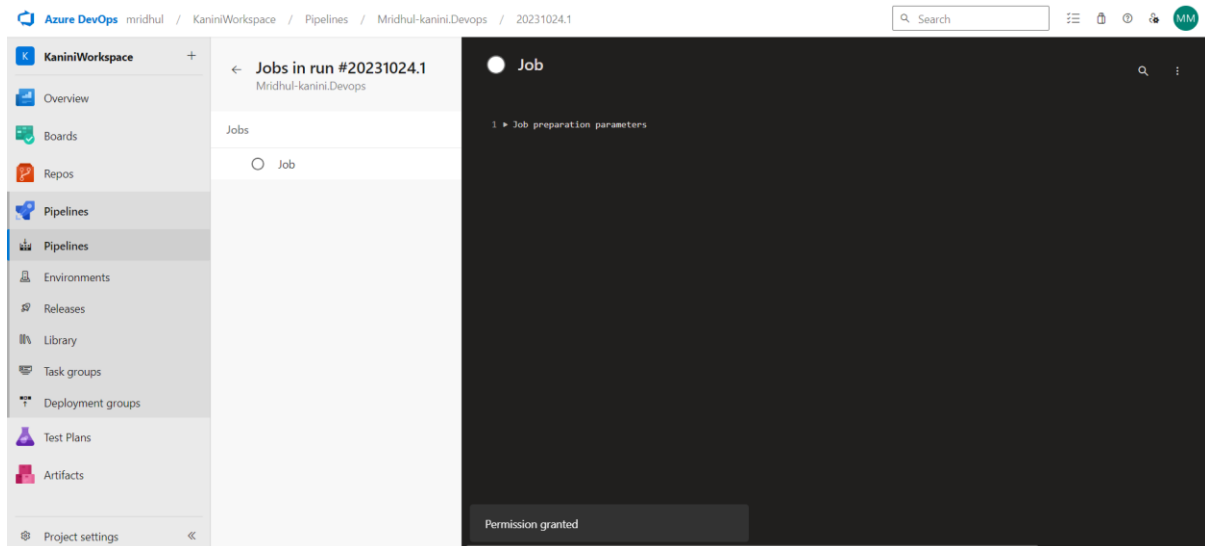
- Commit message:** A text input field containing 'Set up CI with Azure Pipelines'.
- Optional extended description:** A larger text area for additional details.
- Commit options:** Two radio buttons: 'Commit directly to the main branch' (which is selected) and 'Create a new branch for this commit'.
- Save and run button:** A blue button at the bottom right of the dialog.

The background interface remains the same, showing the 'Review your pipeline YAML' screen.

## Give Permission

The screenshot shows the Azure DevOps interface for a workspace named 'KaniniWorkspace'. The left sidebar contains a navigation menu with options: Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Artifacts, and Project settings. The main area is titled 'Jobs in run #20231024.1' and shows a single job in a pending state. A warning message at the top of the job details states: 'This pipeline needs permission to access a resource before this run can continue'. The job details also show a single step: 'Job is pending...'. A 'View' button is visible next to the warning message.

This screenshot is similar to the one above, but it includes a modal dialog box titled 'Checks and manual validations for Stage 0'. The dialog box has a close button (X) in the top right corner. It displays a 'Permission' section with a lock icon, indicating that 'Agent pool Default' requires 'Permission needed'. A blue 'Permit' button is located at the bottom right of the dialog box. The background interface remains the same, showing the pipeline job in a pending state with the same warning message.

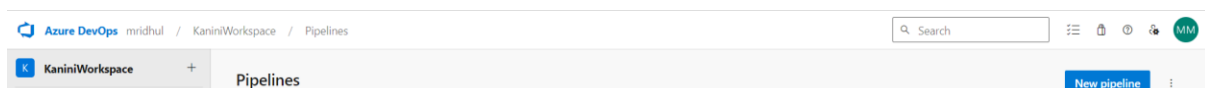


## Lab 9: Create CI Pipeline for .NET Core Application with MS Unit Test

Objective: Create a CI pipeline, either classic or YAML, to build a .NET Core application and run MS Unit tests.

Tasks:

1. Set up a new Azure DevOps project.
2. Create a CI/CD pipeline for a .NET Core application.
3. Configure the pipeline to use MS Unit tests.
4. Trigger the pipeline and validate the test results.



## Choose Github

This screenshot shows the 'Where is your code?' page in the Azure DevOps interface. The left sidebar contains navigation links for KaniniWorkspace, Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Artifacts, and Project settings. The main content area has tabs for Connect, Select, Configure, and Review. Under the 'Connect' tab, there is a section titled 'New pipeline' and 'Where is your code?'. It lists several options: Azure Repos Git (YAML), Bitbucket Cloud (YAML), GitHub (YAML), GitHub Enterprise Server (YAML), Other Git, and Subversion. A link at the bottom says 'Use the classic editor to create a pipeline without YAML'.

Azure DevOps mridhul / KaniniWorkspace / Pipelines

Search

KaniniWorkspace +

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

Connect Select Configure Review

New pipeline

**Where is your code?**

Azure Repos Git (YAML)  
Free private Git repositories, pull requests, and code search

Bitbucket Cloud (YAML)  
Hosted by Atlassian

GitHub (YAML)  
Home to the world's largest community of developers

GitHub Enterprise Server (YAML)  
The self-hosted version of GitHub Enterprise

Other Git  
Any generic Git repository

Subversion  
Centralized version control by Apache

Use the classic editor to create a pipeline without YAML

## Choose Repo

This screenshot shows the 'Select a repository' page in the Azure DevOps interface. The left sidebar is the same as the previous screenshot. The main content area has tabs for Connect, Select, Configure, and Review. Under the 'Select' tab, there is a section titled 'New pipeline' and 'Select a repository'. It includes a filter bar with 'Filter by keywords' and 'My repositories'. Below this, a list of repositories is shown, each with a user profile picture, repository name, and date. The repositories listed are: Mridhul-kanini/Devops (Oct 16), Mridhul-kanini/SmartHotel360-Website (fork, Oct 6), Saran-kanini/KaniniWorkspaceLobby (private, Aug 25), Mridhul-kanini/BigBang\_Assessment\_3 (Aug 9), Mridhul-kanini/BigBang\_Assessment\_2 (Jul 9), Mridhul-kanini/BigBang-Assignment (May 26), and Mridhul-kanini/API-and-EF (private, May 25). A status bar at the bottom indicates 'Waiting for dev.azure.com...'.

Azure DevOps mridhul / KaniniWorkspace / Pipelines

Search

KaniniWorkspace +

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

Connect Select Configure Review

New pipeline

**Select a repository**

Filter by keywords My repositories

Mridhul-kanini/Devops  
Oct 16

Mridhul-kanini/SmartHotel360-Website (fork)  
Oct 6

Saran-kanini/KaniniWorkspaceLobby (private)  
Aug 25

Mridhul-kanini/BigBang\_Assessment\_3  
Aug 9

Mridhul-kanini/BigBang\_Assessment\_2  
Jul 9

Mridhul-kanini/BigBang-Assignment  
May 26

Mridhul-kanini/API-and-EF (private)  
May 25

Waiting for dev.azure.com...



## Choose ASP .net Framework

Azure DevOps mridhul / KaniniWorkspace / Pipelines

You selected a public repository, but this is not a public project. Go to [project settings](#) to change the visibility of the project. [Learn more](#)

Existing Azure Pipelines YAML file  
Select an Azure Pipelines YAML file in any branch of the repository.

- .NET Core Function App to Windows on Azure  
Build a .NET Core function app and deploy it to Azure as a Windows function App.
- .NET Desktop  
Build and run tests for .NET Desktop or Windows classic desktop solutions.
- Android  
Build your Android project with Gradle.
- Ant  
Build your Java projects and run tests with Apache Ant.
- ASP.NET  
Build and test ASP.NET projects.
- ASP.NET Core  
Build and test ASP.NET Core projects targeting .NET Core.
- ASP.NET Core (.NET Framework)  
Build and test ASP.NET Core projects targeting the full .NET Framework.
- C/C++ with GCC  
Build your C/C++ project with GCC using make.
- Deploy to Azure Kubernetes Service  
Build and push image to Azure Container Registry; Deploy to Azure Kubernetes Service

## Edit yaml

Azure DevOps mridhul / KaniniWorkspace / Pipelines

You selected a public repository, but this is not a public project. Go to [project settings](#) to change the visibility of the project. [Learn more](#)

✓ Connect ✓ Select ✓ Configure **Review**

New pipeline

**Review your pipeline YAML** Variables Save and run

Mridhul-kanini/DevOps / azure-pipelines-1.yml \* Show assistant

```
1 # ASP.NET Core (.NET Framework)
2 # Build and test ASP.NET Core projects targeting the full .NET Framework.
3 # Add steps that publish symbols, save build artifacts, and more:
4 # https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core
5
6 trigger:
7   - main
8
9 pool:
10  - vmImage: 'windows-latest'
11
12 variables:
13   - solution: '**/*.sln'
14   - buildPlatform: 'Any CPU'
15   - buildConfiguration: 'Release'
16
17 steps:
18   - task: NuGetToolInstaller@1
19
```

## Save and run

The screenshot shows the 'Save and run' dialog box in Azure DevOps. The dialog has a title bar with a close button. Below the title bar, it says 'Saving will commit azure-pipelines-1.yml to the repository.' There is a 'Commit message' field with the text 'Set up CI with Azure Pipelines'. Below that is an 'Optional extended description' field with the placeholder text 'Add an optional description...'. At the bottom, there are two radio buttons: 'Commit directly to the main branch' (which is selected) and 'Create a new branch for this commit'. A 'Save and run' button is at the bottom right.

**Save and run**

Saving will commit azure-pipelines-1.yml to the repository.

Commit message  
Set up CI with Azure Pipelines

Optional extended description  
Add an optional description...

☒ Commit directly to the main branch  
☐ Create a new branch for this commit

Save and run

## Permit

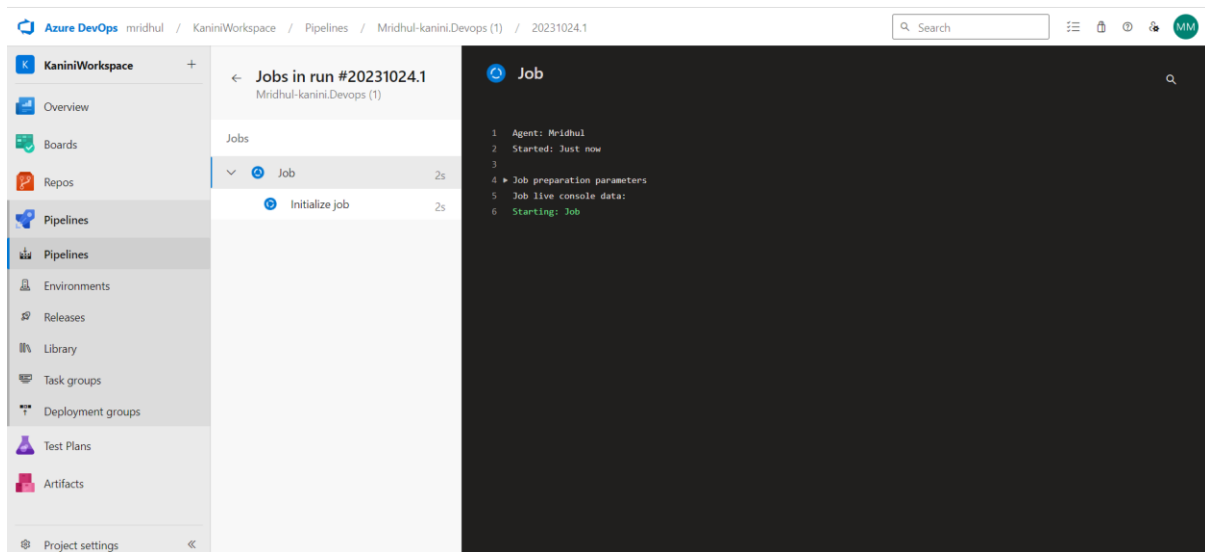
The screenshot shows the 'Permit' dialog box in Azure DevOps. The dialog has a title bar with a close button. Below the title bar, it says 'Checks and manual validations for Stage 0'. There is a 'Permission' field with the text 'Agent pool Default' and a 'Permit' button. Below that, it says 'Permission needed'. The background shows the 'Jobs in run #20231024.1' page with a 'Job' tab selected. The job is in a 'pending' state.

**Checks and manual validations for Stage 0**

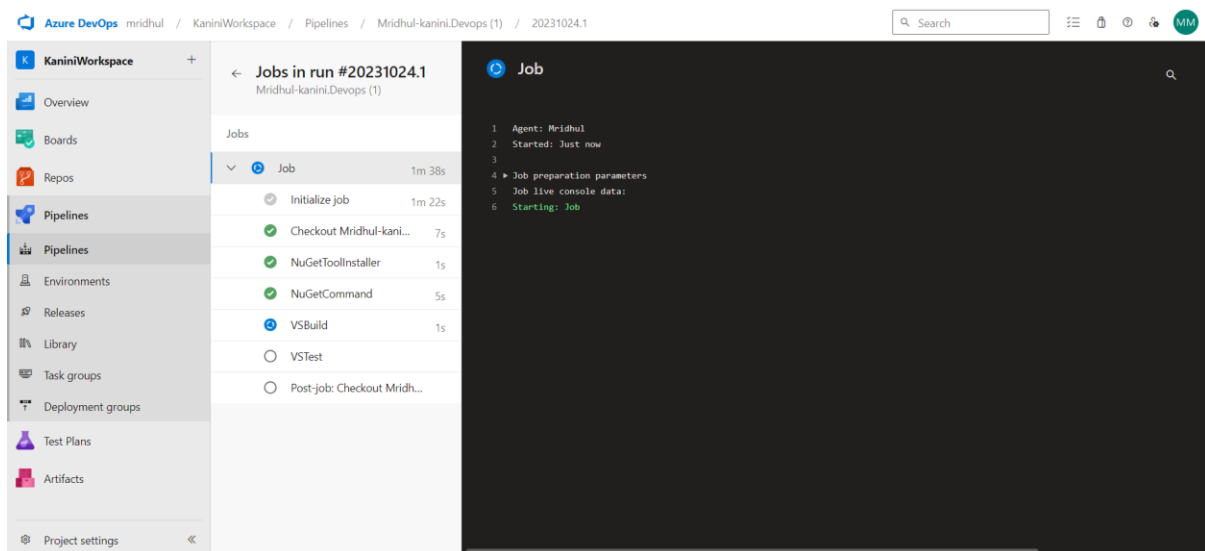
Permission Agent pool Default  
Permission needed

Permit

## Starting Pipeline



## Pipeline ran successfully



## Lab 10: Creating a Docker Image for a .NET Core Web API and Running it in

### Rancher Desktop

Objective: In this lab, you will create a Docker image for a sample .NET Core Web

API application and then run the Web API container in Rancher Desktop.

Prerequisites:

Rancher Desktop installed and running.

.NET Core SDK installed on your machine.

## Tasks

Step 1: Create a .NET Core Web API Project

Step 2: Build the .NET Core Web API Project

Step 3: Dockerize the .NET Core Web API

Step 4: Build the Docker Image

Step 5: Run the Docker Container in Rancher Desktop

Step 6: Test the .NET Core Web API via swagger

Ans:

Step 1: Create a .NET Core Web API Project

### Additional information

ASP.NET Core Web API

C# Linux macOS Windows Cloud Service Web WebAPI

Framework ⓘ

.NET 6.0 (Long Term Support)

Authentication type ⓘ

None

☒ Configure for HTTPS ⓘ

☒ Enable Docker ⓘ

Docker OS ⓘ

Linux

☒ Use controllers (uncheck to use minimal APIs) ⓘ

☒ Enable OpenAPI support ⓘ

☐ Do not use top-level statements ⓘ

Back Create

## Creating Docker File

This PC > Local Disk (C:) > Kanini Software Solutions > KaniniWorkspacePortal > DevOpsAssessment >

Name	Date modified	Type	Size
.vs	10/20/2023 12:59 PM	File folder	
DevOpsAssessment	10/20/2023 12:40 PM	File folder	
DevOpsAssessment.sln	10/20/2023 12:40 PM	Visual Studio Solut...	2 KB
dockerfile	10/20/2023 12:49 PM	File	1 KB

```
1 # Use the official .NET Core SDK image as a build stage
2 FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build
3 WORKDIR /source
4 COPY . .
5 RUN dotnet restore "./DevOpsAssessment/DevOpsAssessment.csproj" --disable-parallel
6 RUN dotnet publish "./DevOpsAssessment/DevOpsAssessment.csproj" -c release -o /app --no-restore
7
8 # Build the runtime image
9 FROM mcr.microsoft.com/dotnet/aspnet:6.0
10 WORKDIR /app
11 COPY --from=build /app ./
12 EXPOSE 5000
13
14 ENTRYPOINT ["dotnet", "DevOpsAssessment.dll", "--environment=Development"]
15
```

## Listing All Images

```
C:\Kanini Software Solutions\KaniniWorkspacePortal\DevOpsAssessment>docker images
```

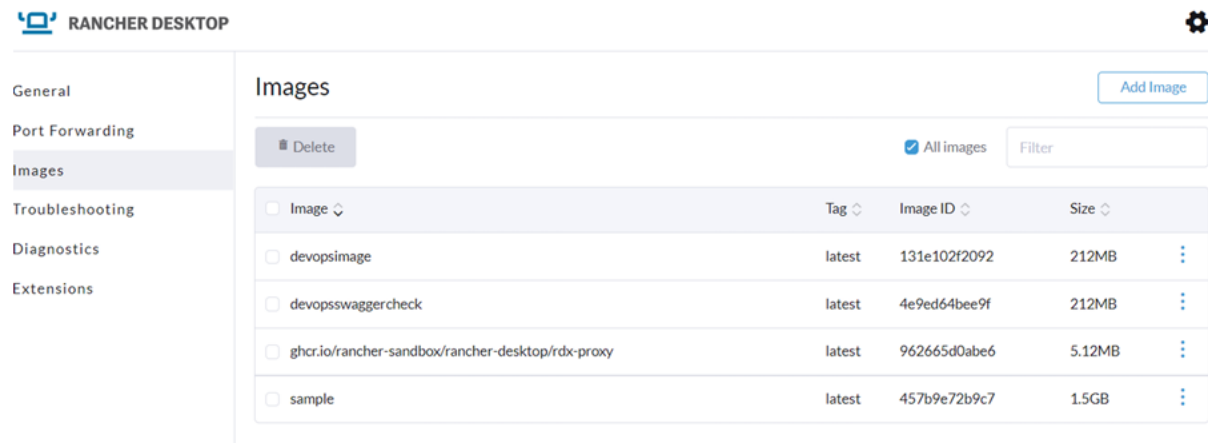
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
devopsimage	latest	131e102f2092	2 days ago	212MB
devopsswaggercheck	latest	4e9ed64bee9f	2 days ago	212MB
sample	latest	457b9e72b9c7	3 weeks ago	1.5GB
ghcr.io/rancher-sandbox/rancher-desktop/rdx-proxy	latest	962665d0abe6	N/A	5.12MB

## Building the docker file

```
C:\Kanini Software Solutions\KaniniWorkspacePortal\DevOpsAssessment>docker build -t devopsswagger .
[+] Building 17.9s (14/14) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 32B
=> [internal] load .dockerignore
=> => transferring context: 28
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:6.0
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:6.0
=> [internal] load build context
=> => transferring context: 5.03MB
=> [build 1/5] FROM mcr.microsoft.com/dotnet/sdk:6.0@sha256:1167d21bf2d6b0aeb9fce863a73e90defa9233574b74d9deabee
=> [stage-1 1/3] FROM mcr.microsoft.com/dotnet/aspnet:6.0@sha256:d4ff6177eb040028c0e2d9d6a4ba813841826e412e5b36c
=> => resolve mcr.microsoft.com/dotnet/aspnet:6.0@sha256:d4ff6177eb040028c0e2d9d6a4ba813841826e412e5b36c1a82fc93
=> CACHED [build 2/5] WORKDIR /source
=> [build 3/5] COPY . .
=> [build 4/5] RUN dotnet restore "./DevOpsAssessment/DevOpsAssessment.csproj" --disable-parallel
=> [build 5/5] RUN dotnet publish "./DevOpsAssessment/DevOpsAssessment.csproj" -c release -o /app --no-restore
=> CACHED [stage-1 2/3] WORKDIR /app
=> CACHED [stage-1 3/3] COPY --from=build /app ./
=> exporting to image
=> => exporting layers
=> => writing image sha256:4e9ed64bee9fc1b1e862305148d911a3aaf9e0e4f971415989d1d7b455e03b97
=> => naming to docker.io/library/devopsswagger
C:\Kanini Software Solutions\KaniniWorkspacePortal\DevOpsAssessment>
```

Run the Docker Container in Rancher Desktop:

You can find the docker image “devopsswaggercheck” in Rancher

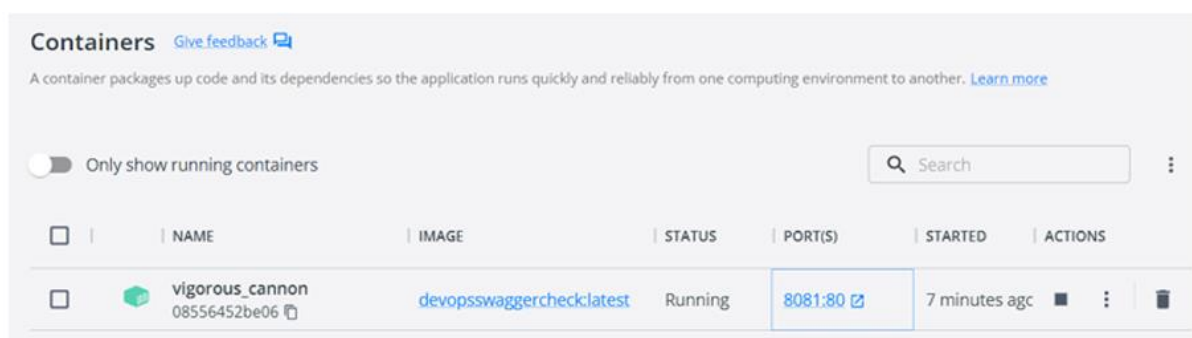


Test the .NET Core Web API via swagger

Now I am running the docker image

```
C:\Kanini Software Solutions\KaniniWorkspacePortal\DevOpsAssessment>docker run -p 8081:80 devopsswaggercheck:latest
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:80
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /app/
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```

Its runs in swagger via docker



Finally the .net application runs on swagger via docker and rancher

