

Devops Final Assesment

Sandarsh A

Section 1: Multiple-Choice Questions (MCQs)

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

- a. Windows Software Locator
- b. Windows System Locator
- c. Windows Subsystem for Linux
- d. Windows Shell Language

ANS c. Windows Subsystem for Linux

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

- a. Automating manual testing
- b. Frequent integration of code changes
- c. Managing cloud infrastructure
- d. Monitoring server performance

ANS b. Frequent integration of code changes

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

- a. Copy files and directories
- b. Change the working directory
- c. Create a new directory
- d. Calculate directory size

ANS b. Change the working directory

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

- a. Ubuntu
- b. CentOS
- c. Docker
- d. Debian

ANS a. Ubuntu

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

- a. Managing cloud infrastructure
- b. Running virtual machines
- c. Packaging and deploying applications in containers
- d. Managing network security

ANS c. Packaging and deploying applications in containers

6. What is the primary purpose of Azure DevOps?

- a. Infrastructure management
- b. Software development and delivery
- c. Network security
- d. Virtualization

ANS b. Software development and delivery

7. Which components are part of Azure DevOps?

- a. Azure App Service and Azure Functions
- b. Azure Monitor and Azure Security Center
- c. Azure Boards and Azure Pipelines
- d. Azure Virtual Machines and Azure SQL Database

ANS c. Azure Boards and Azure Pipelines

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

- a. It provides automated database backups.
- b. It tracks changes in source code and manages versions.
- c. It monitors server performance.
- d. It optimizes network configurations.

ANS b. It tracks changes in source code and manages versions.

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

- a. Managing user accounts
- b. Running GUI applications
- c. Administrative tasks with superuser privileges
- d. Monitoring network traffic

ANS c. Administrative tasks with superuser privileges

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

- a. Azure Boards
- b. Azure Repos
- c. Azure Pipelines
- d. Azure Artifacts

ANS c. Azure Pipelines

Section 2: Labs

Lab 1: File and Directory Management

1. open the ubuntu command prompt

2. Open the home folder by using the below command

```
root@Sandarsh:/# cd home
```

3. creating the dictionary with name lab1

```
root@Sandarsh:/home# mkdir lab1
```

4. creating the text file in lab1

```
root@Sandarsh:/home/lab1# nano sample.txt
```

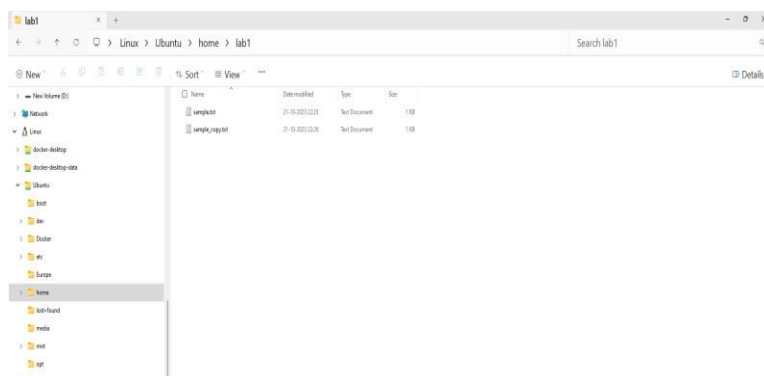
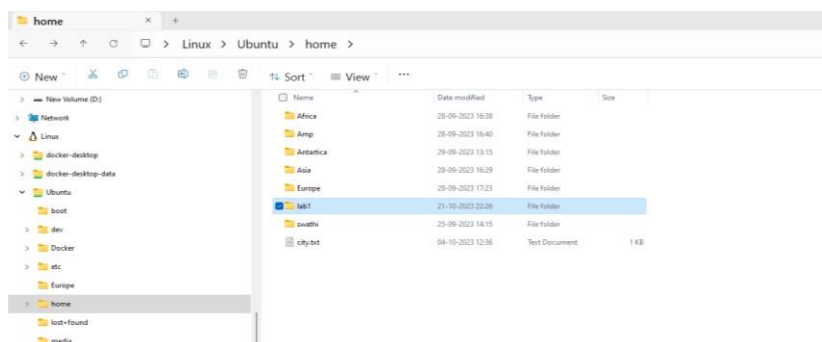
5. root@Sandarsh:/home/lab1# cp sample.txt sample_copy.txt

```
root@Sandarsh:/home/lab1# ls
```

```
sample.txt sample_copy.txt
```

6. List the files in the lab1 directory to confirm their names

```
sample.txt sample_copy.txt
```



Lab2 Permissions and Ownership

1.Create the folder with name lab2

```
root@Sandarsh:/home# mkdir lab2
```

```
root@Sandarsh:/home# ls
```

folders present in home folder

```
Africa Amp Antartica Asia Europe city.txt lab1 lab2 sandarsh
```

```
root@Sandarsh:/home/lab2# ls
```

2.create the text file

```
root@Sandarsh:/home/lab2 touch secret.txt
```

3.To give permission

```
chmod 600 secret.txt
```

4.chownsandarsh secret.txt

```
root@Sandarsh:/home# ls -l lab2/secret.txt
```

```
-rw ----- 1sandarsh root 0 Oct 23 05:39 lab2/secret.txt
```

5.root@Sandarsh:/home# ls -n lab2/secret.txt

```
-rw ----- 1 1000 0 0 Oct 23 05:39 lab2/secret.txt
```

Lab 3: Text Processing with Command Line Tools

1.Create the folder with name lab3

```
root@Sandarsh:/home/lab3# ls
```

```
random.txt
```

2. use echo command to write text in the file

```
hello this issandarsh from batch 4
```

3.to search for specific word

```
root@Sandarsh:/home# grep "sandarsh" lab3/random.txt
```

```
hello this issandarsh from batch 4
```

4.replacing the wordsandarsh with the word replcaemnt

```
root@Sandarsh:/home# sed -i 's/sandarsh/replacement/g' lab3/random.txt
```

```
hello this is replacement from batch 4
```

5. to count lines and numbers

```
root@Sandarsh:/home# wc lab3/random.txt
```

```
0  7 38 lab3/random.txt
   7 words
  38 characters
```

Lab 4 Working with Lists in YAML

1. Creating the config.yaml in vs code

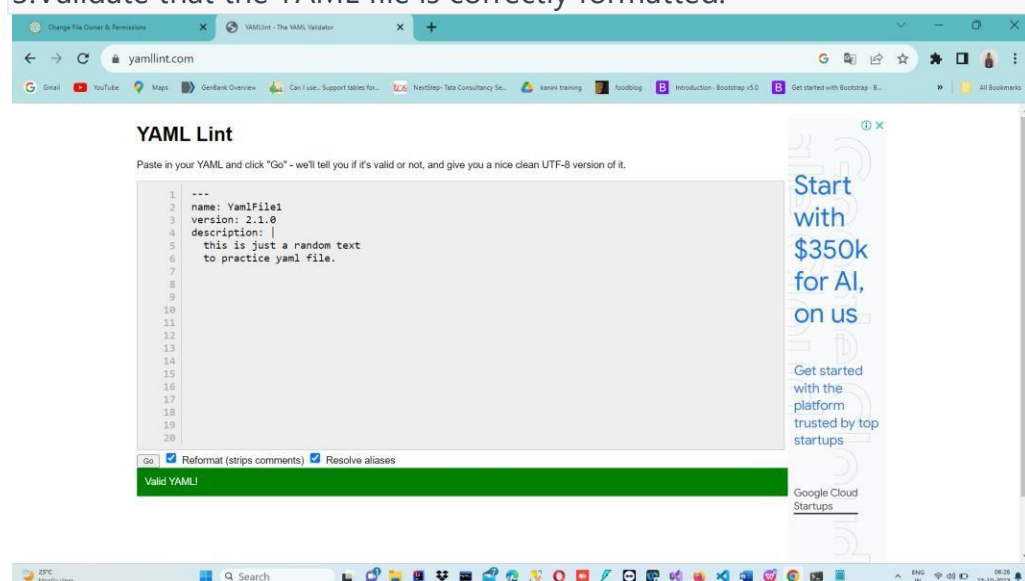
C:\Users\SANDARSH\Documents\config.yaml

1. Define key-value pairs in YAML for a fictitious application, including name, version, and description.

```
name: YamlFile1
version: 2.1.0
description: |
  this is just a random text
  to practice yaml file.
```

2. Save the file

3. Validate that the YAML file is correctly formatted.



Lab 5: Working with Lists in YAML

1. Create the fruits.yaml file

C:\Users\SANDARSH\Documents\fruits.yaml

```
favorite_fruits:
```

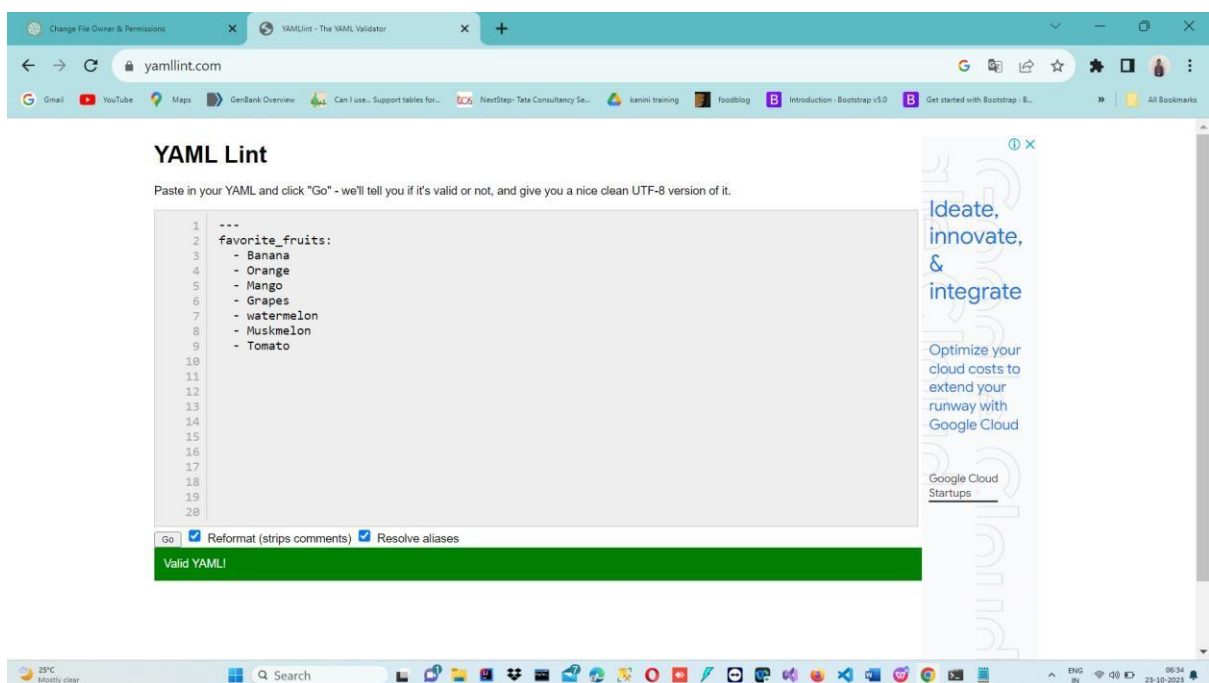
- Banana
- Orange
- Mango
- Grapes

2. Adding items to the file

```
favorite_fruits:
```

- Banana
- Orange
- Mango
- Grapes
- watermelon
- Muskmelon
- Tomato

3. Validating the file



Lab 6: Nested Structures in YAML

1. Create the data.yaml file

C:\Users\SANDARSH\Documents\data.yaml

Define a nested structure representing a fictitious organization with departments and employees.

```
organization:
  name: XYZ
  departments:
    - name: Marketing
      employees:
        - name: Suresh
          position: Marketing Director
        - name: Prabhu
          position: Marketing Specialist
    - name: Development
      employees:
        - name: Tom
          position: Development Manager
        - name: sandarsh
          position: Software Engineer
    - name: Data
      employees:
        - name: Jerry
          position: Team lead
        - name: Thrupthi
          position: Data Analyst
```

2. Adding new member to the data

```
- name: Data
  employees:
    - name: Jerry
      position: Team lead
    - name: Thrupthi
      position: Data Analyst
```

```
- name: Bob Johnson
  position: Sales Representative
```

3. removing tom from the development list

```
organization:
  name: XYZ
  departments:
    - name: Marketing
      employees:
        - name: Suresh
          position: Marketing Director
        - name: Prabhu
          position: Marketing Specialist
    - name: Development
      employees:
        - name: sandarsh
          position: Software Engineer
    - name: Data
      employees:
        - name: Jerry
          position: Team lead
        - name: Thrupthi
          position: Data Analyst
        - name: Bob Johnson
          position: Sales Representative
```

4. updated data.yaml file

```
organization:
  name: XYZ
  departments:
    - name: Marketing
      employees:
```

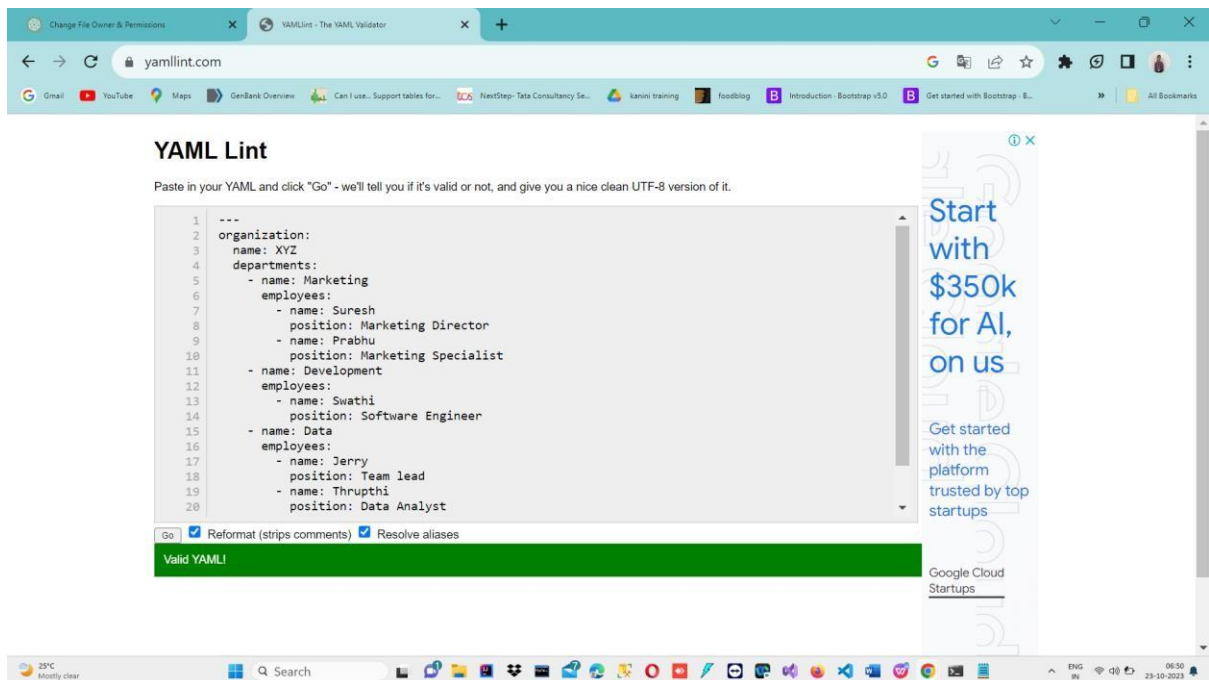


```

- name: Suresh
  position: Marketing Director
- name: Prabhu
  position: Marketing Specialist
- name: Development
  employees:
    - name: sandarsh
      position: Software Engineer
- name: Data
  employees:
    - name: Jerry
      position: Team lead
    - name: Thrupthi
      position: Data Analyst
    - name: Bob Johnson
      position: Sales Representative

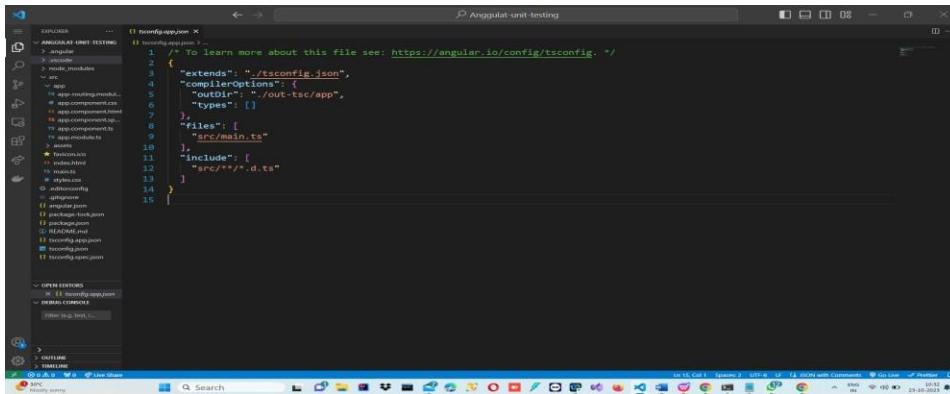
```

5. validating the data.yaml file

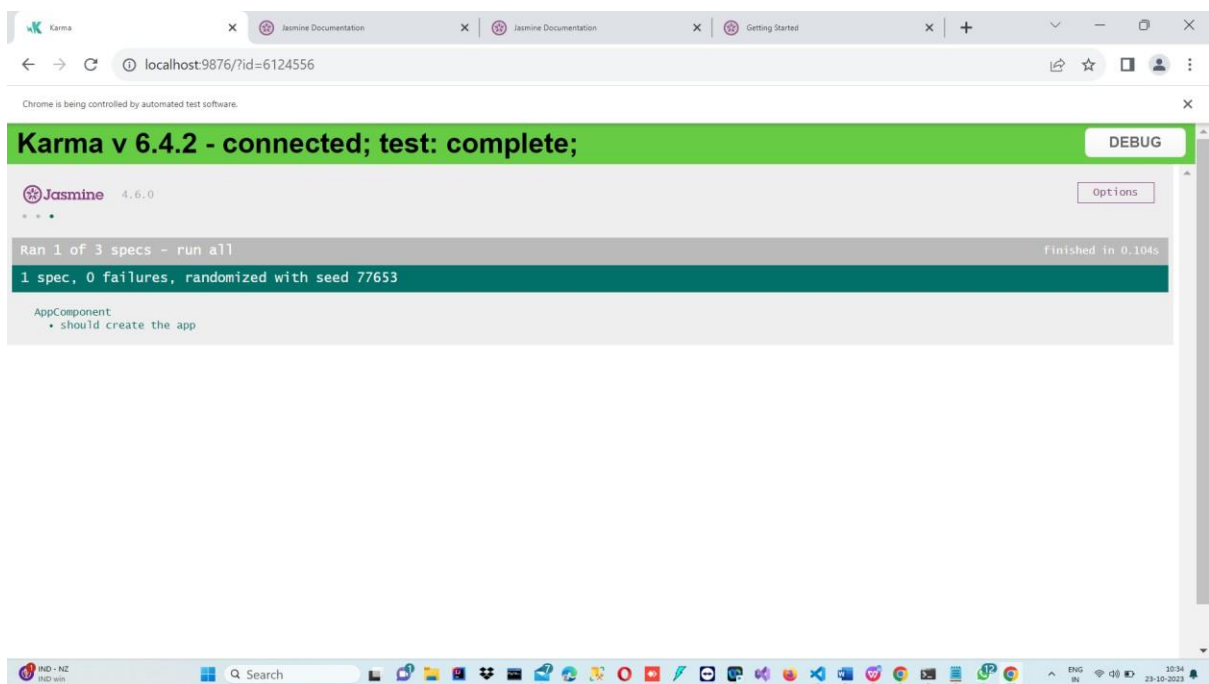


Lab 7: Create Classic Azure CI Pipeline for Angular Application

1. Create the new angular project



2. run the sample test cases



Upload to the Azure Devops Repository by using the below commands

```
PS C:\Users\SANDARSH\Angular-test-project\Anggulat-unit-testing> git init
```

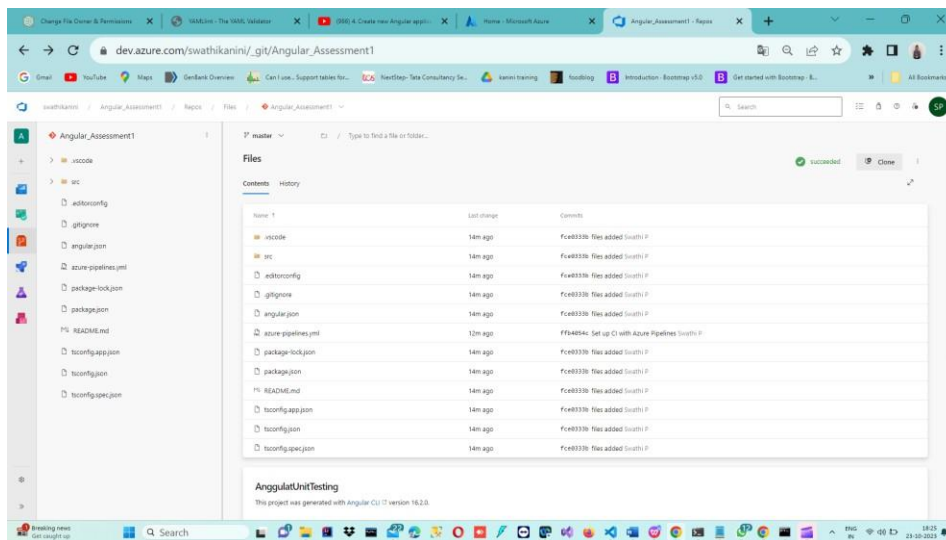
```
PS C:\Users\SANDARSH\Angular-test-project\Anggulat-unit-testing> git remote add origin  
https://sandarshkanini@dev.azure.com/sandarshkanini/Angular_Assessment1/_git/Angular_  
Assessment1
```

```
PS C:\Users\SANDARSH\Angular-test-project\Anggulat-unit-testing> git add .
```

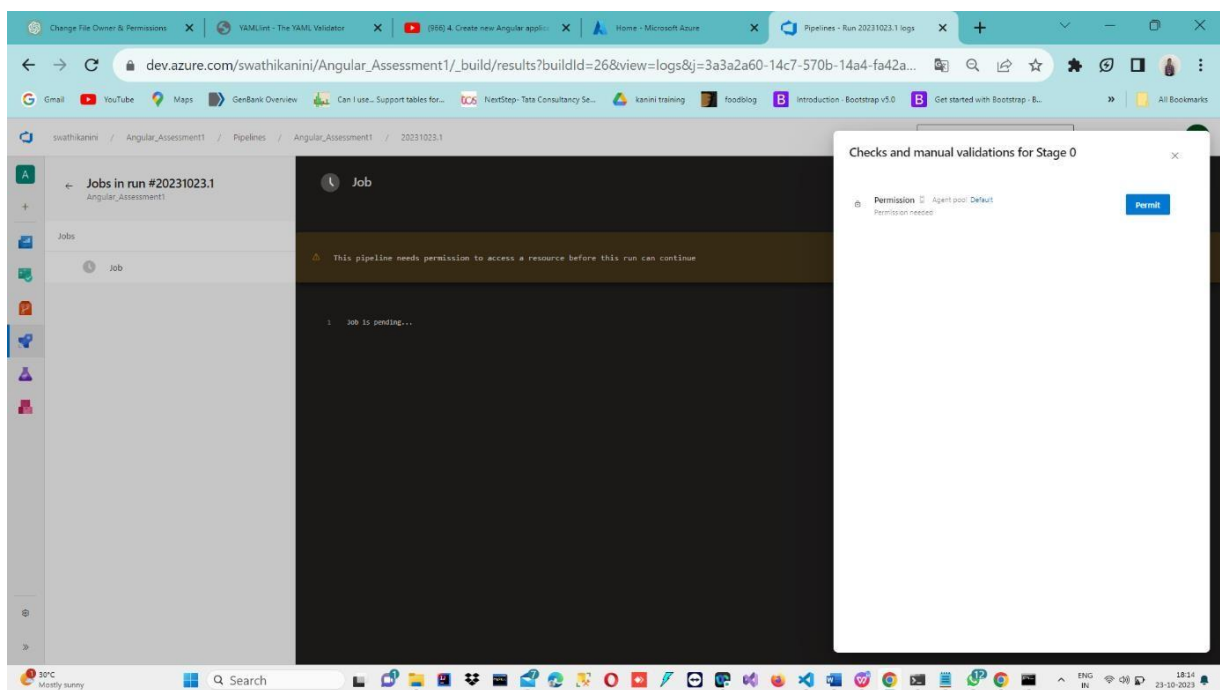
```
PS C:\Users\SANDARSH\Angular-test-project\Anggulat-unit-testing> git commit -m "files  
added"
```

```
PS C:\Users\SANDARSH\Angular-test-project\Anggulat-unit-testing> git push -u origin --all
```

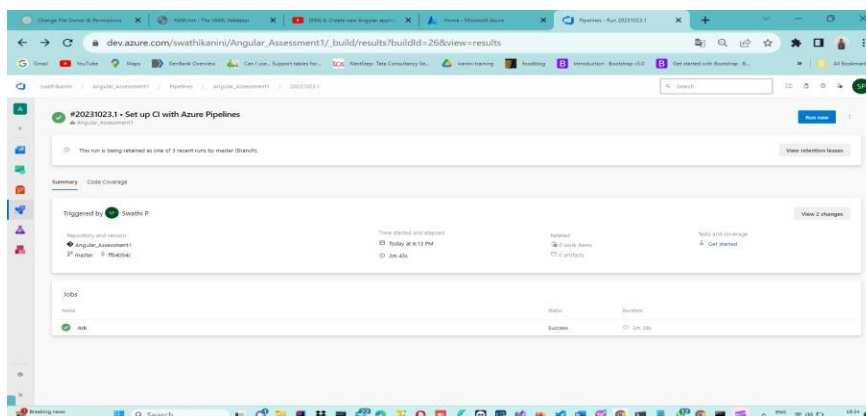
3. Files are uploaded in the Azure Devops Repository

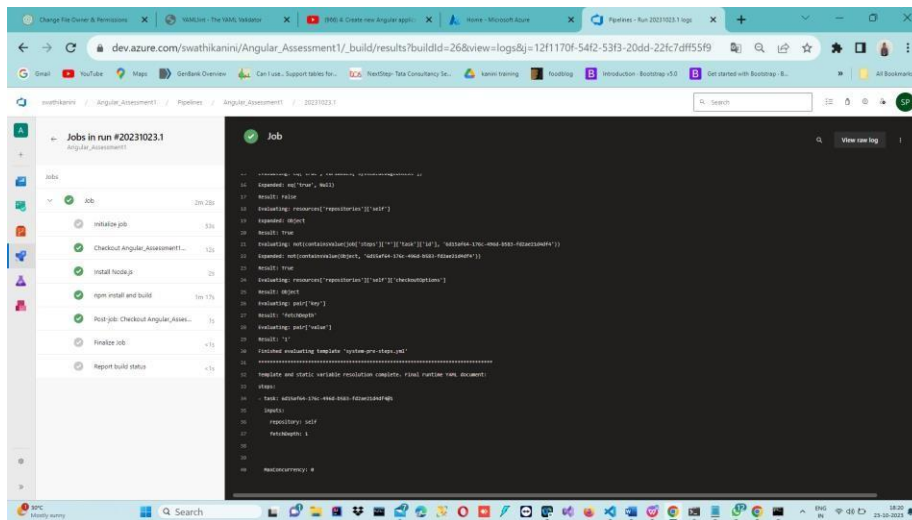


4. Create the pipeline



5. After creating the successful pipeline



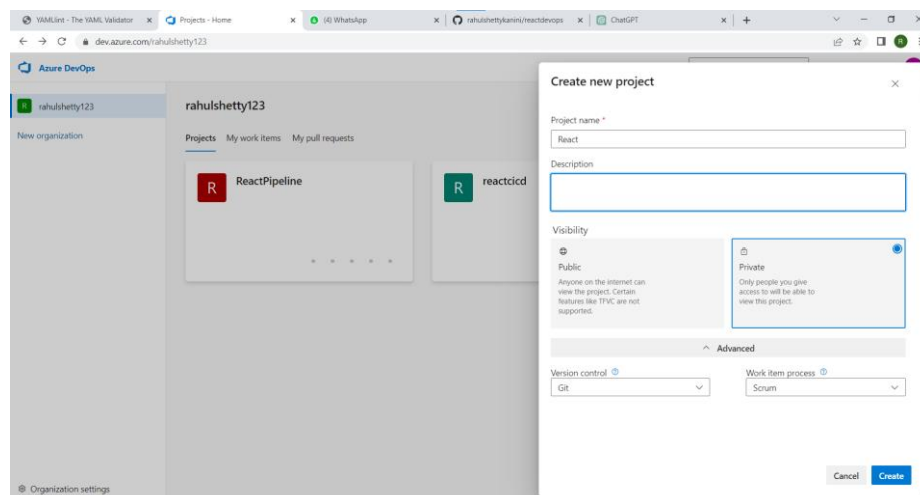


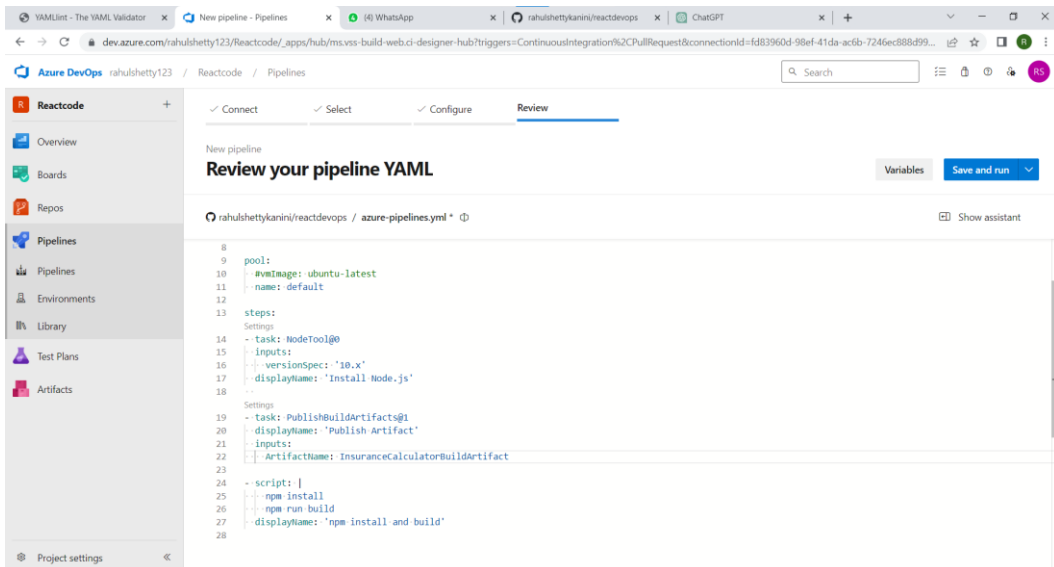
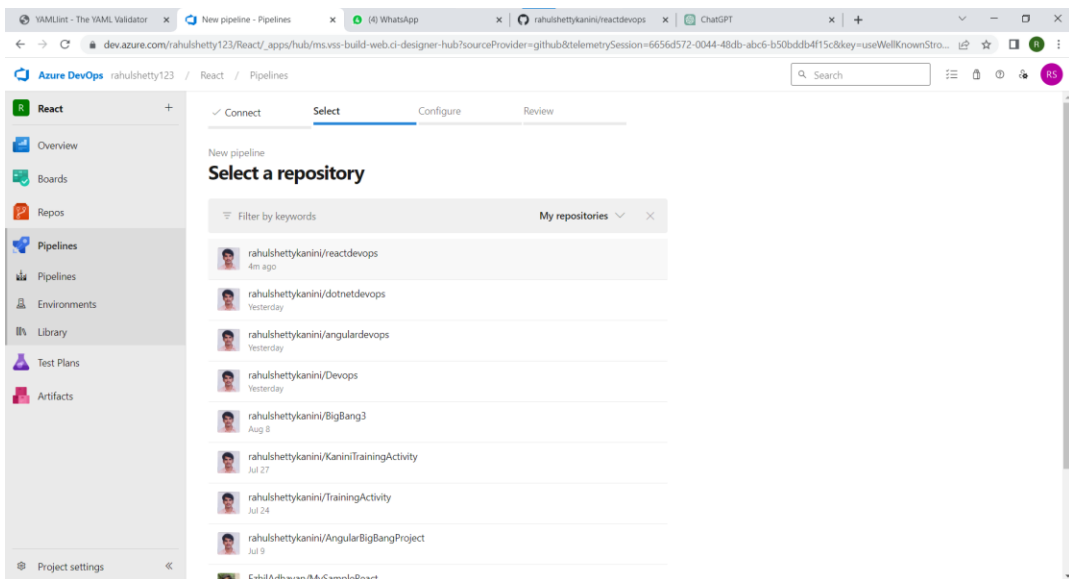
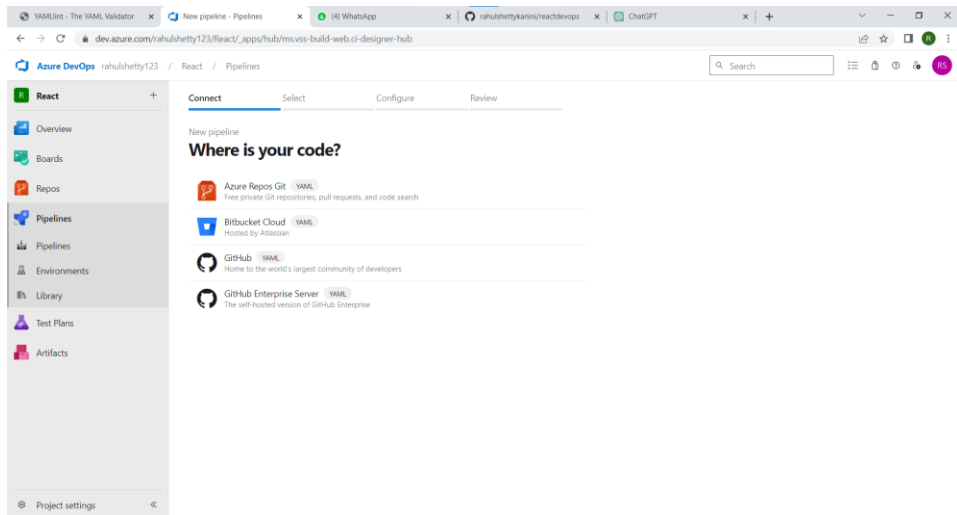
Lab 8: Create YAML Azure CI Pipeline for React Application

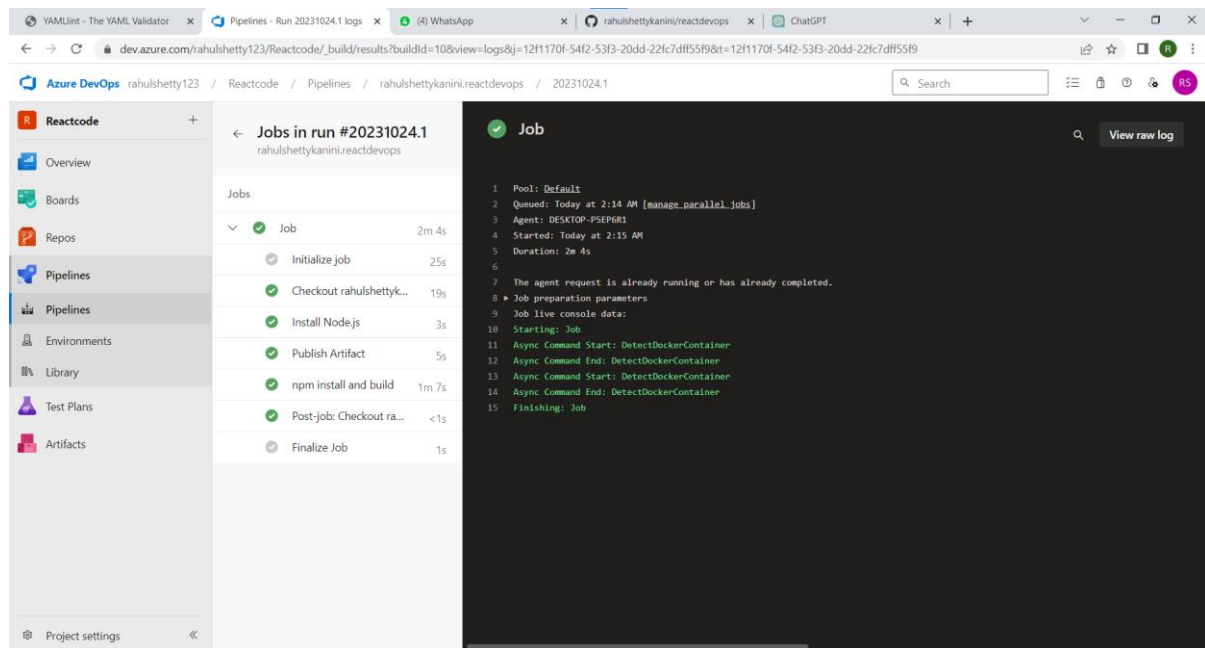
Objective: Create a YAML-based Azure CI pipeline to build a simple React application with unit testing using Enzyme and Jest.

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.





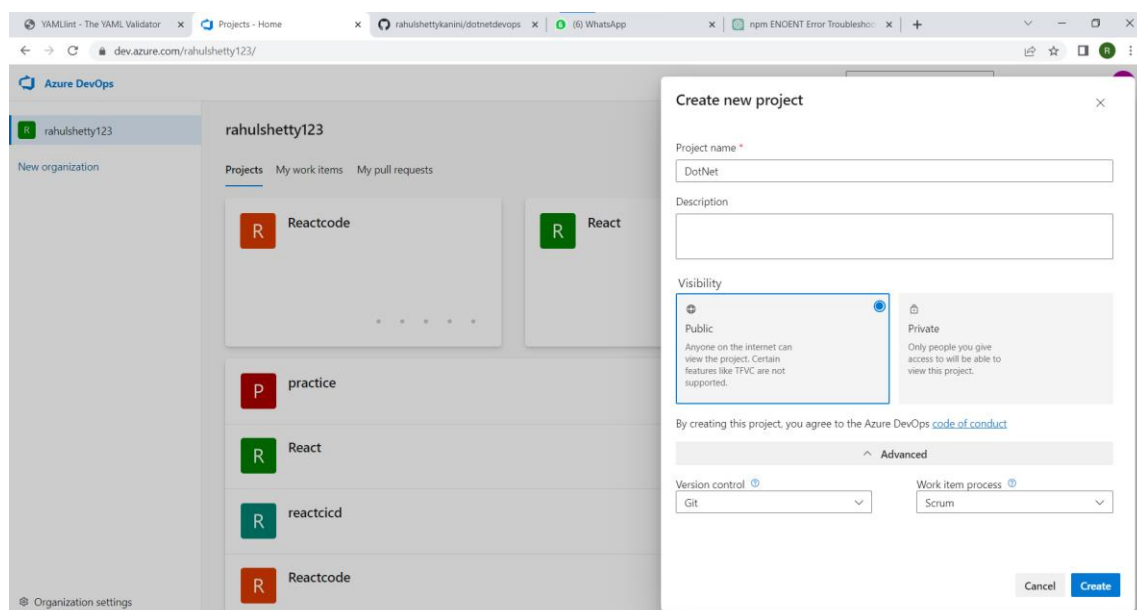


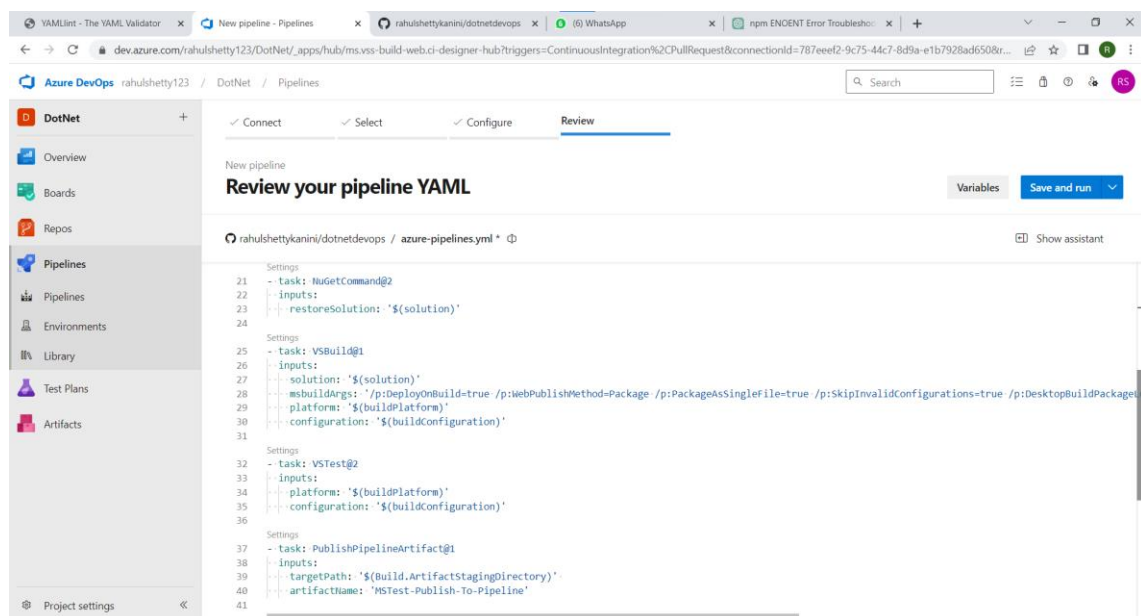
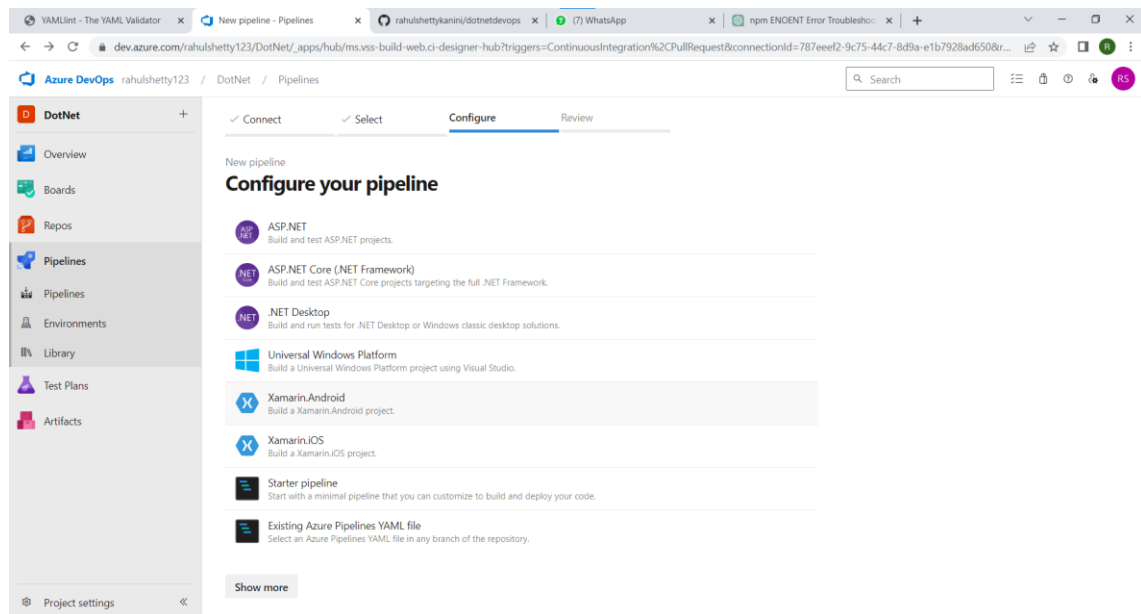
Lab 9: Create CI Pipeline for .NET Core Application with MS Unit Test

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.





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

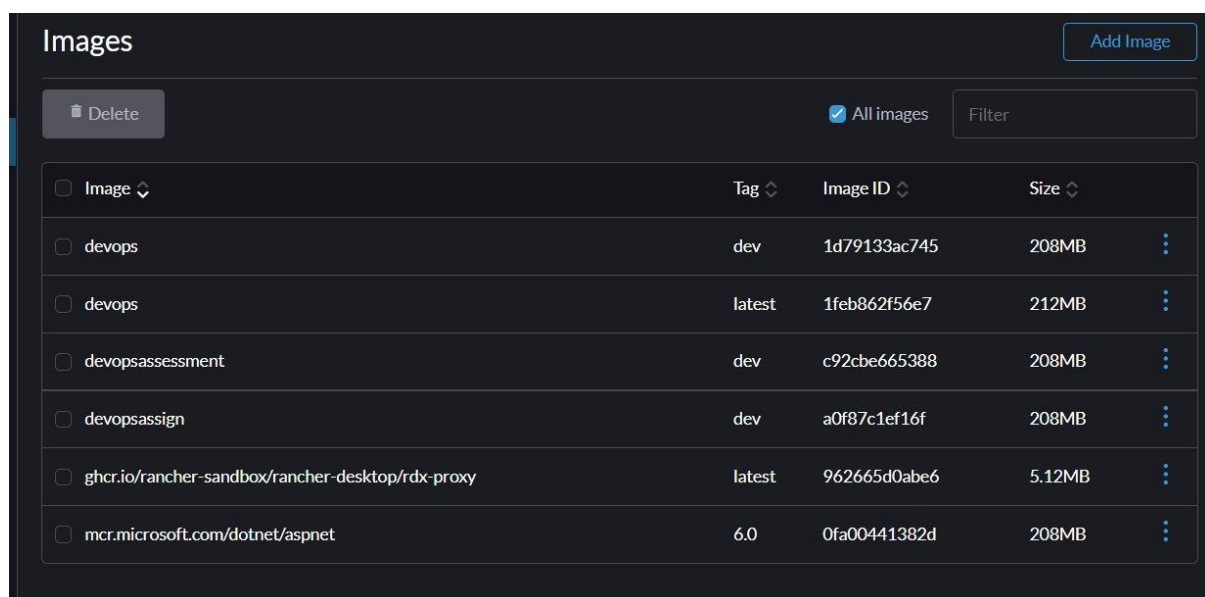
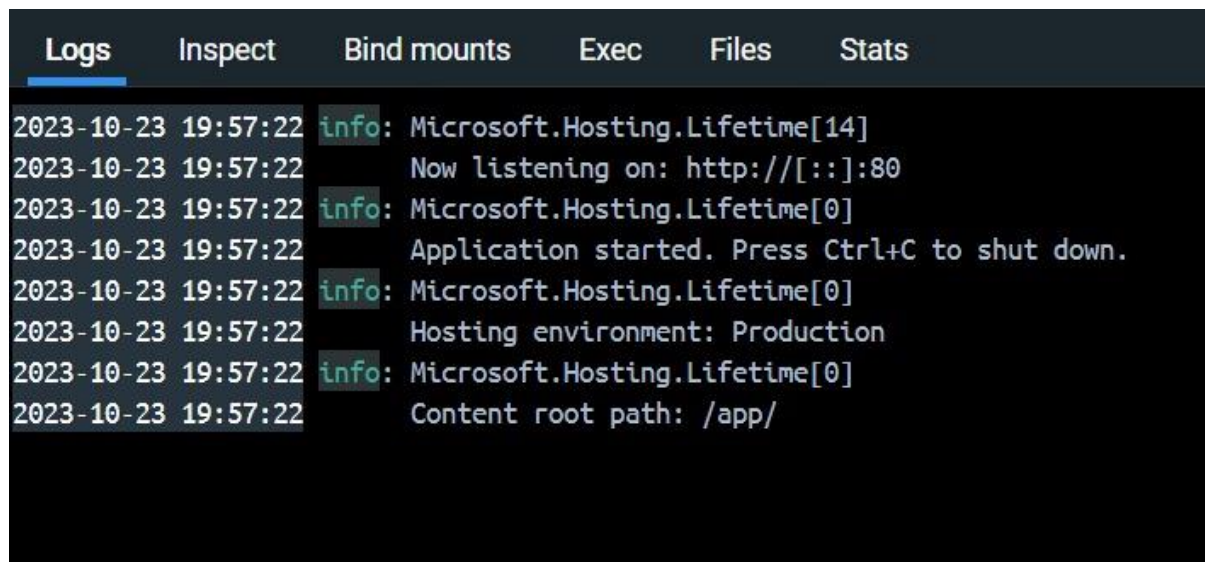
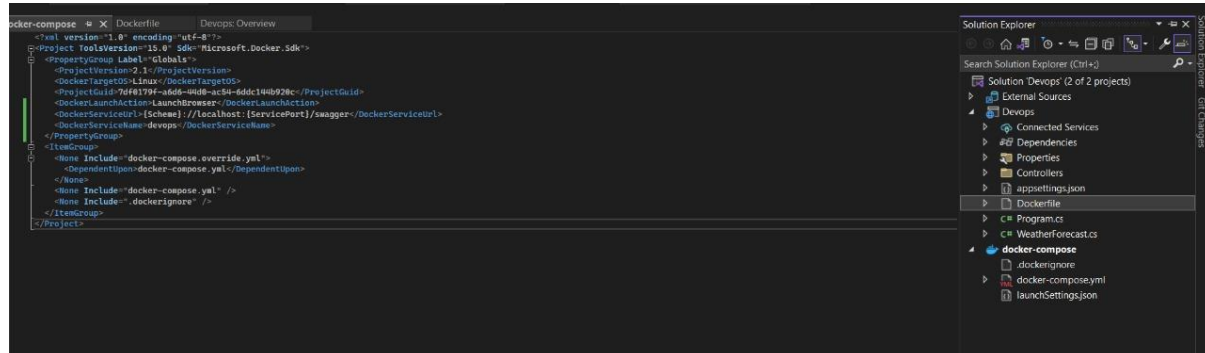
Step 2: Build the .NET Core Web API Project

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.



Devops 1.0 CLASS

<https://localhost:60357/swagger/v1/swagger.json>

WeatherForecast

GET /WeatherForecast

Schemas

WeatherForecast >