Devops – Final Assessment

*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

**Answer : (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

**Answer : (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

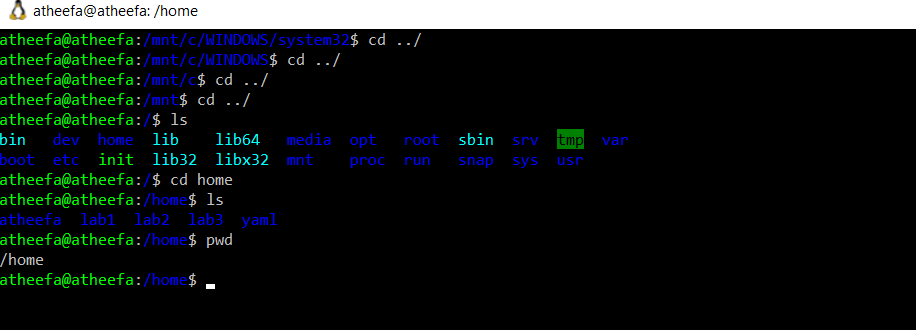
c. Create a new directory

d. Calculate directory size

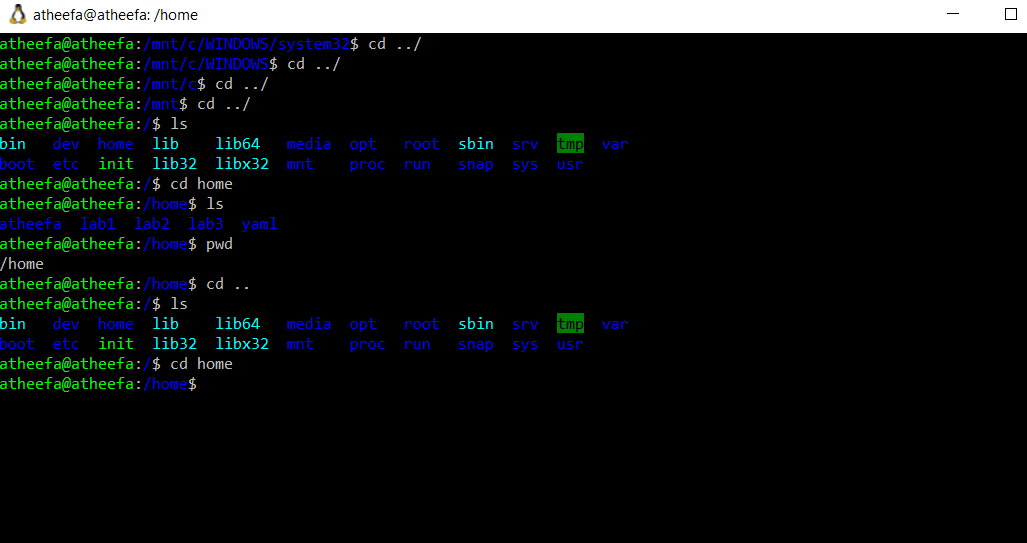
**Answer : (b) Change the working directory**

Example :

Current Working Directory :



If I wanna navigate one step up the directory level,



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

a. Ubuntu

b. CentOS

c. Docker

d. Debian

**Answer : (c) Docker**

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

**Answer : (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

**Answer : (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

**Answer : (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.

**Answer : (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

**Answer : (c) Administrative tasks with superuser privileges**

10) In Azure DevOps, which component is used to define, build, test, and deployapplications?

a. Azure Boards

b. Azure Repos

c. Azure Pipelines

d. Azure Artifacts

**Answer : (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”

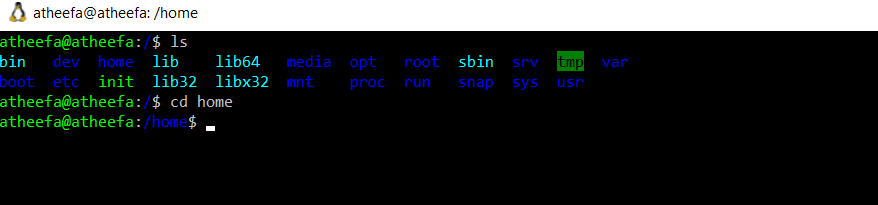
4. Rename “sample\_copy.txt” to “new\_sample.txt”

5. List the files in the “lab1” directory to confirm their names.

Solution :

Step 1 :

Navigate to your home directory by typing cd /home and pressing Enter.



Step 2 :

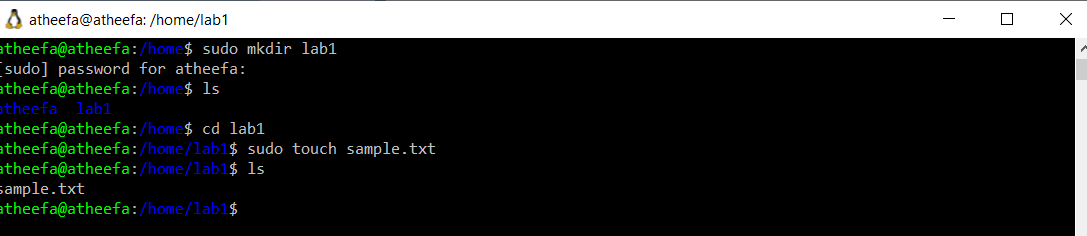
Create a directory called lab1 by typing mkdir lab1 and pressing Enter.



Step 3 , Step 4 :

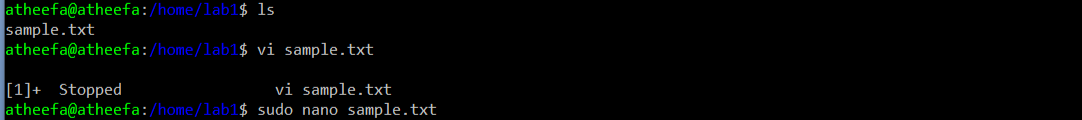
Change to the lab1 directory by typing cd lab1 and pressing Enter.

Create a text file named sample.txt by typing touch sample.txt and pressing Enter.



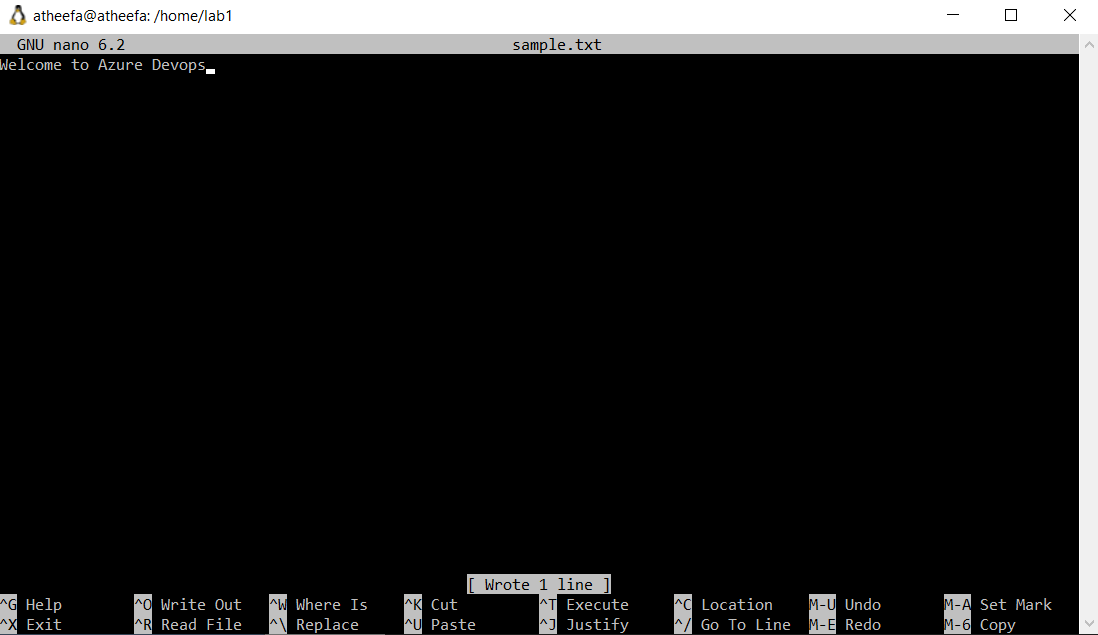
Step 5 :

Open sample.txt in a text editor and add some content.



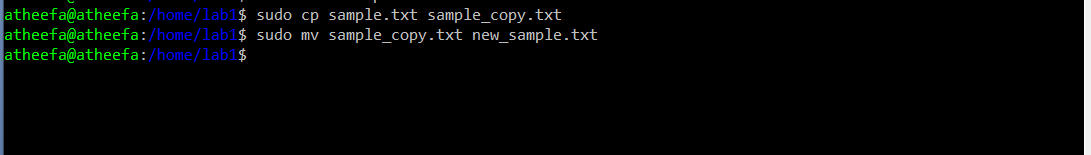
Step 6 :

Save and close the text editor.



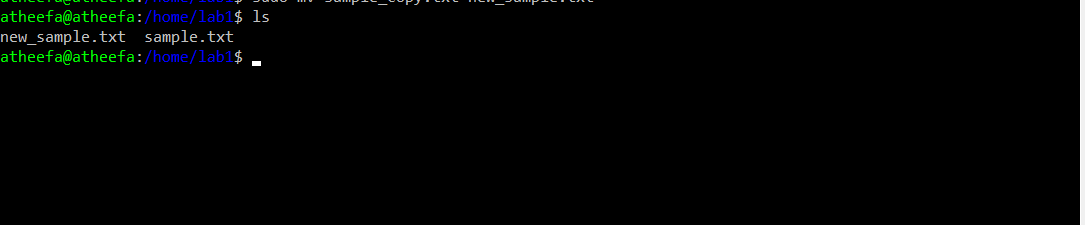
Step 7 , Step 8 :

Make a copy of sample.txt and name it sample\_copy.txt by typing cp sample.txt sample\_copy.txt and pressing Enter.Rename sample\_copy.txt to new\_sample.txt by typing mv new\_sample.txtsample\_copy.txt and pressing Enter.

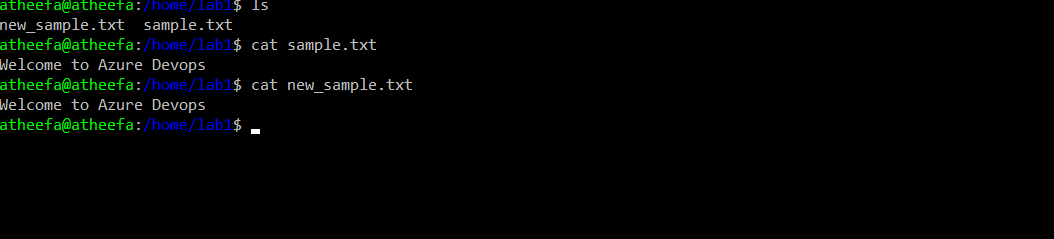


Step 9 :

List the files in the lab1 directory to confirm their names by typing ls and pressing Enter.



Step 10 : Viewing the content of the files using cat command.



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.

Solution :

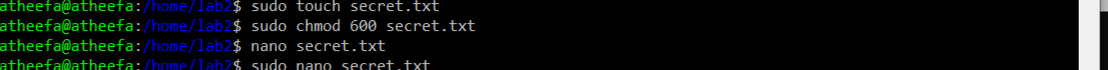
Step 1 :

Create a new file named "secret.txt" in the "lab2" directory.

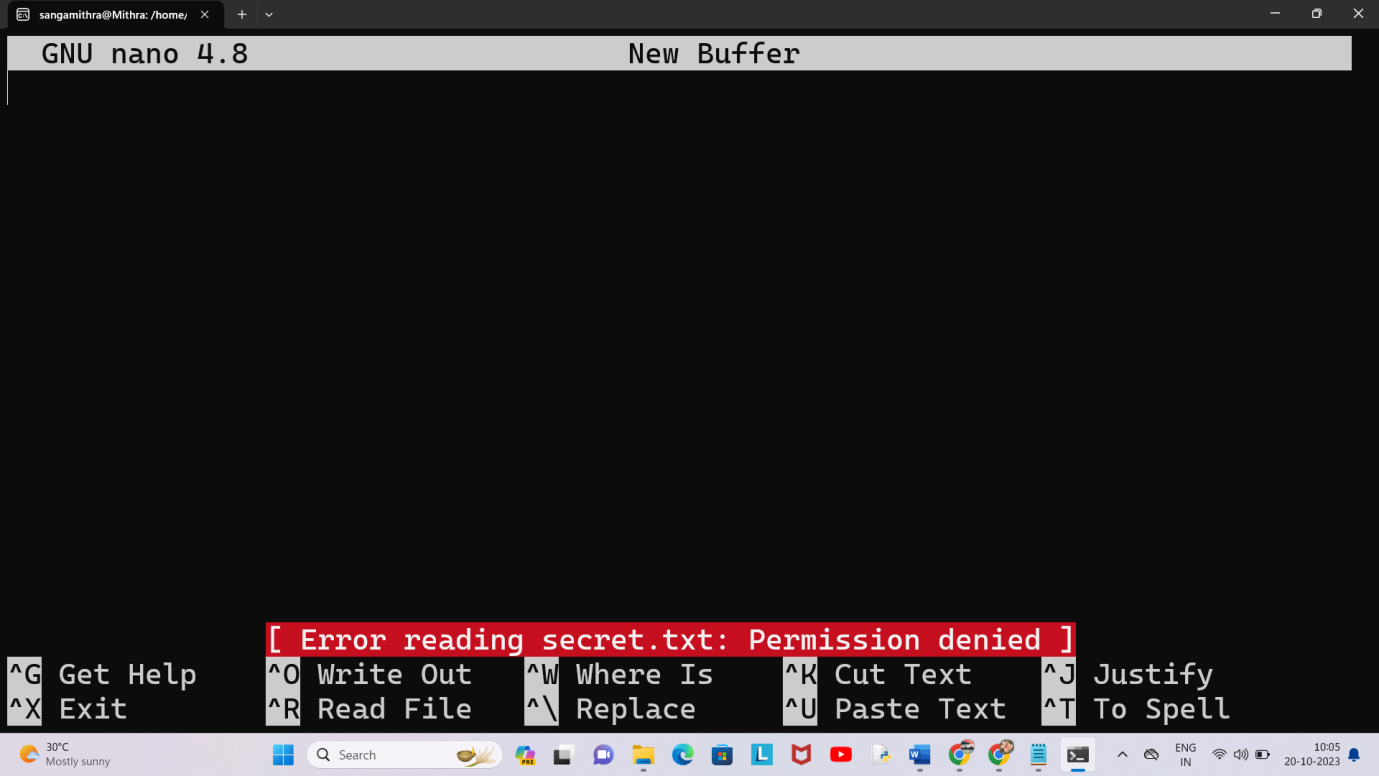


Step 2 :

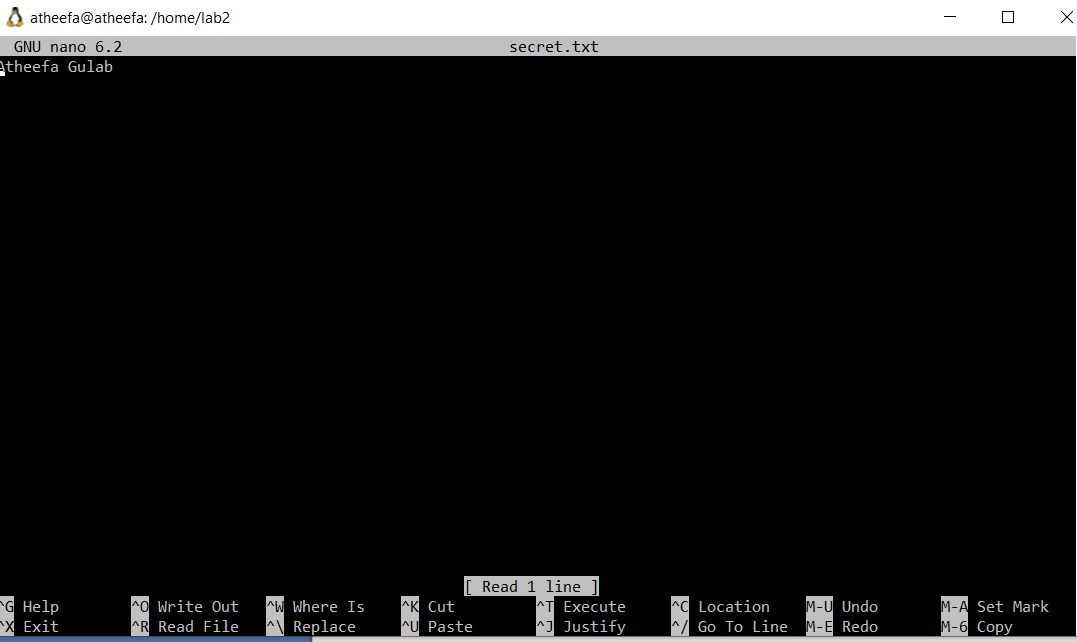
Set the file permissions to allow read and write access only to the owner.



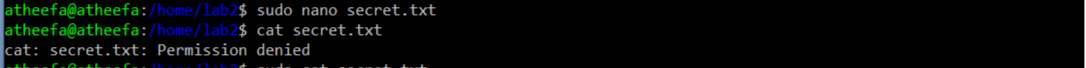
Result for nano secret.txt :



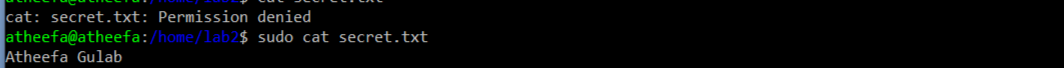
Result for sudo nano secret.txt :



Result for cat secret.txt :



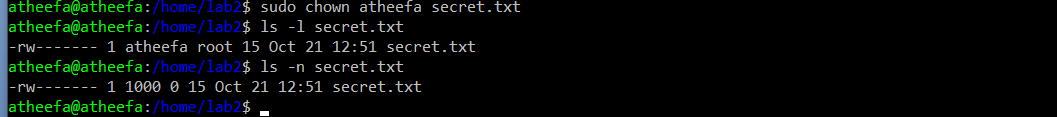
Result for sudo cat secret.txt :



Step 3 , Step 4 :

Change the owner of "secret.txt" to another user.

Verify the new permissions and owner using the ls -l and ls -n commands.



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.

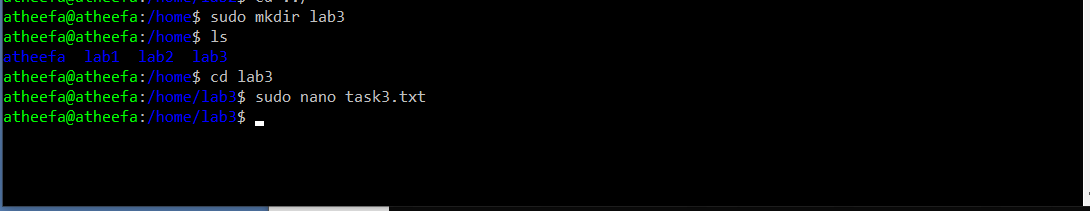
4. Use the wc command to count the number of lines, words, and

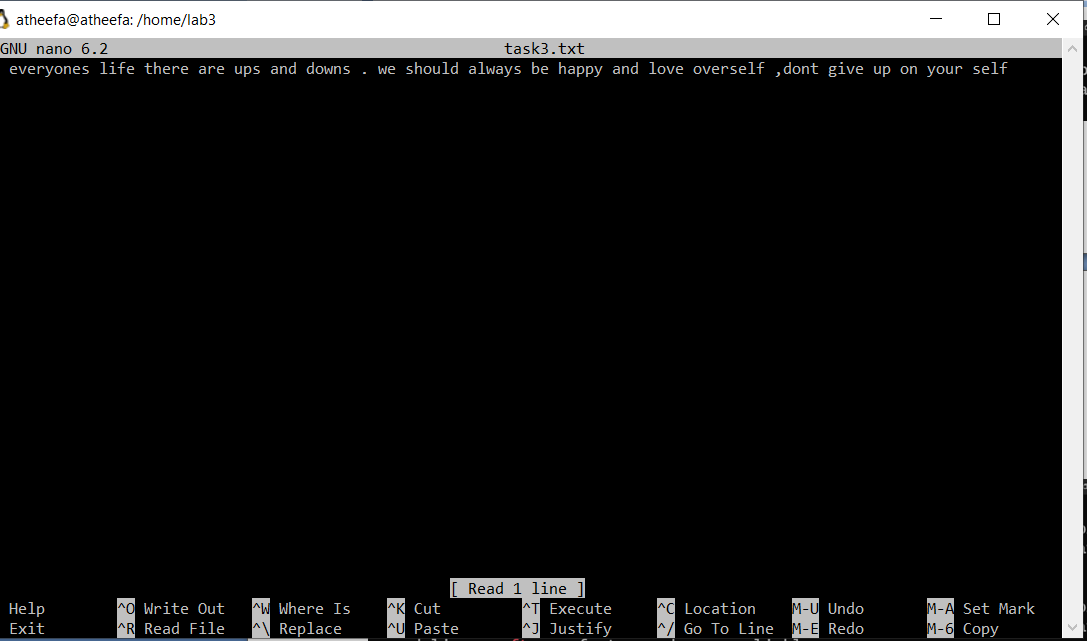
characters in the file.

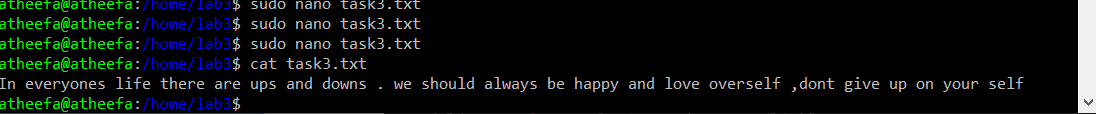
Solution :

Step 1 :

Create a text file with some random text in the "lab3" directory.

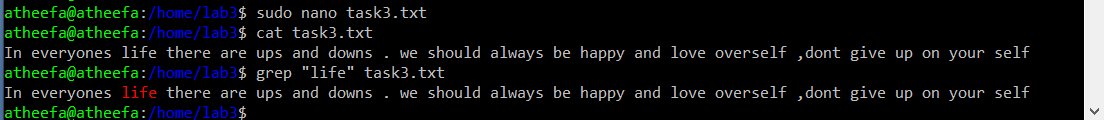






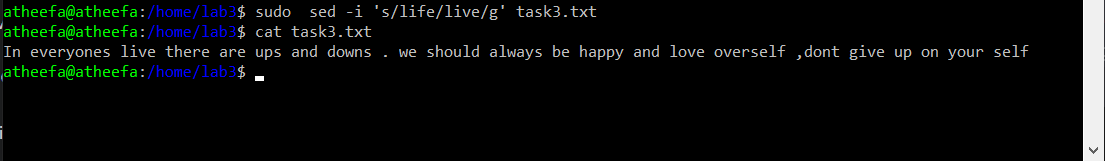
Step 2 :

Use the grep command to search for a specific word or pattern in the file.



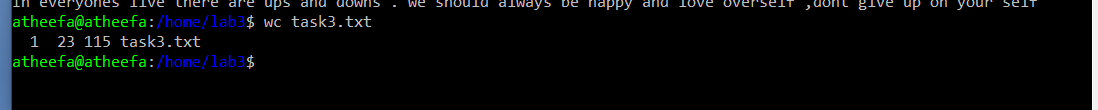
Step 3 :

Use the sed command to replace a word or phrase with another in the file.



Step 4 :

Use the wc command to count the number of lines, words, and characters in the file.



Here's what each number represents:

Line count (1): This indicates the number of newline characters, which corresponds to the number of lines in the file.

Word count (23): This represents the number of words in the file. Words are separated by whitespace (spaces, tabs, or newline characters).

Character count (115): This is the total number of characters in the file, including letters, numbers, punctuation, and whitespace.

So, the file "task3.txt" contains 1 line, 23 words, and 115 characters.

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.

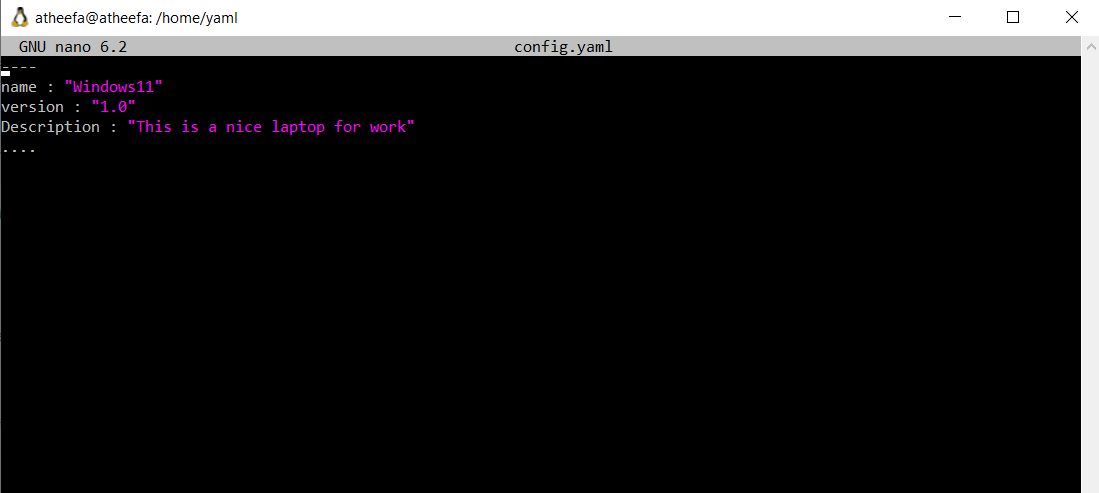
4. Validate that the YAML file is correctly formatted.

Step 1 ,Step 2:

Create a YAML file named "config.yaml"

Define key-value pairs in YAML for a fictitious application

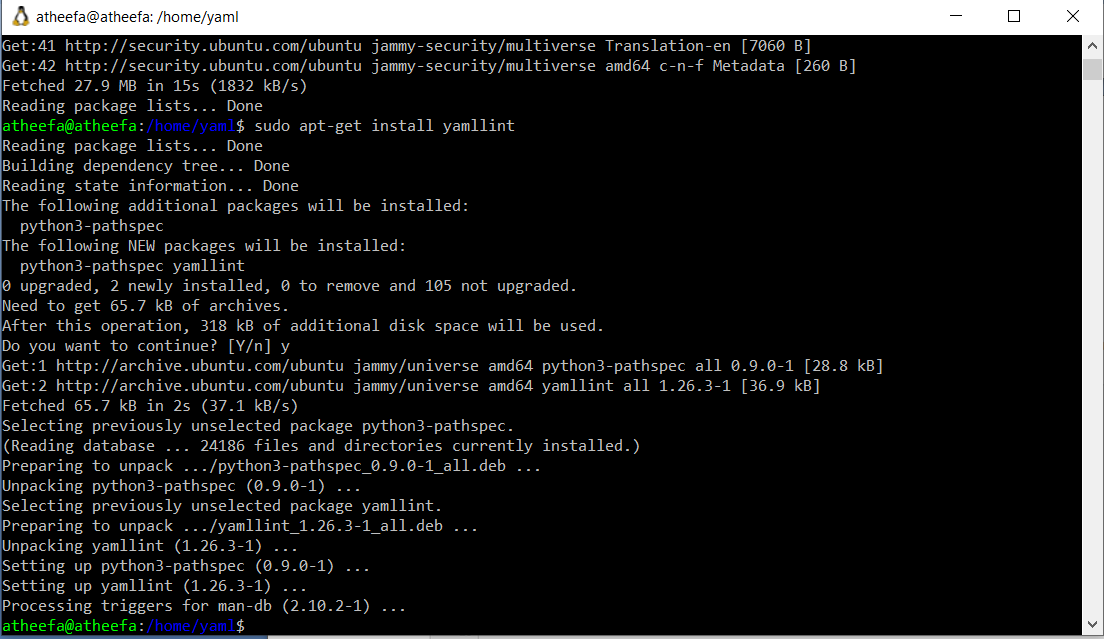




Step 3 :

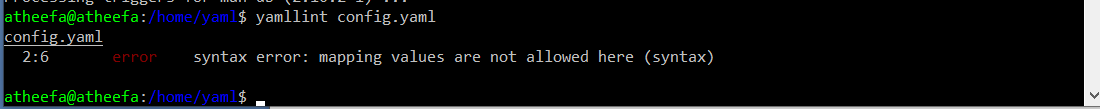
Validate that the YAML file is correctly formatted

To ensure that your YAML file is correctly formatted, we can use a YAML validator. There are online tools available for this purpose, or we can use a command-line tool. One such command-line tool is yamllint.

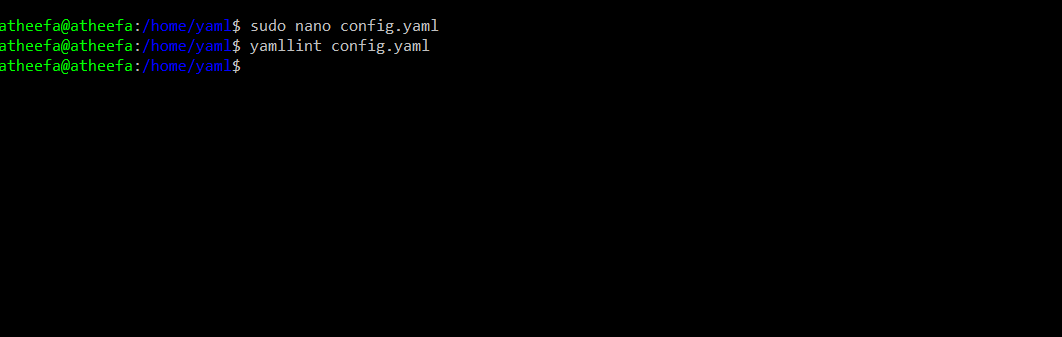


Case 1 :

With error :



Without error :



Online Editor :



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.

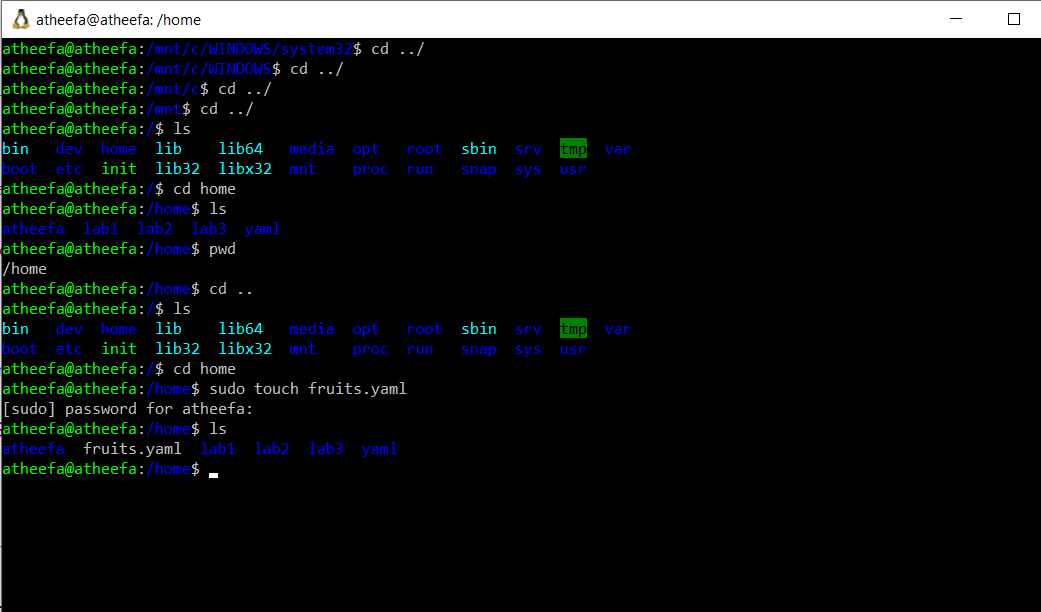
3. Add items from the list.

4. Save and validate the YAML file.

Solution :

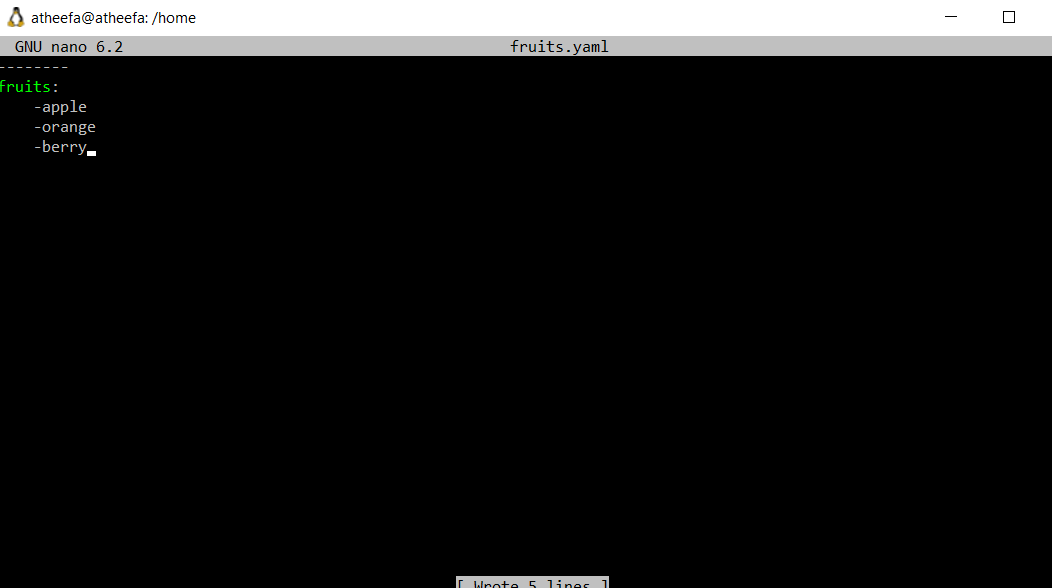
Step 1 :

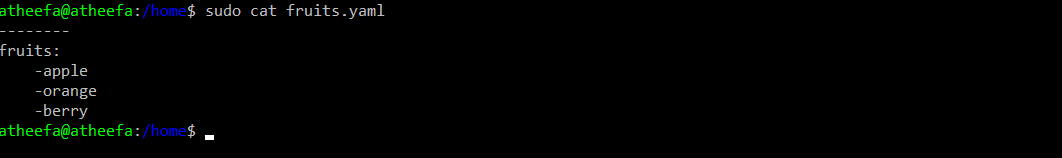
Create a YAML file named "fruits.yaml."



Step 2 :

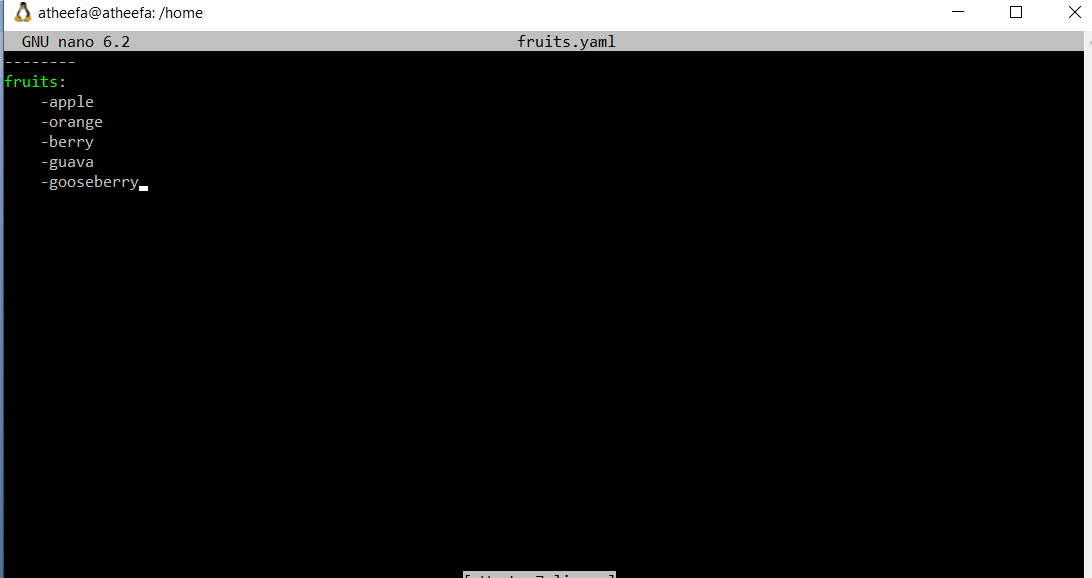
Define a list of your favorite fruits using YAML syntax.





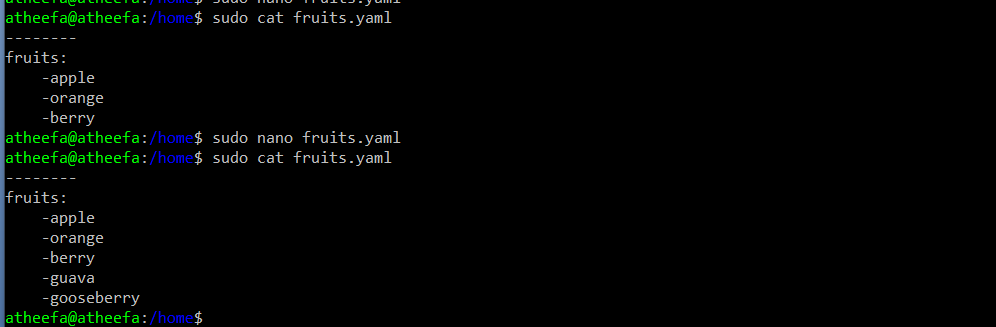
Step 3 :

Add items to the list.



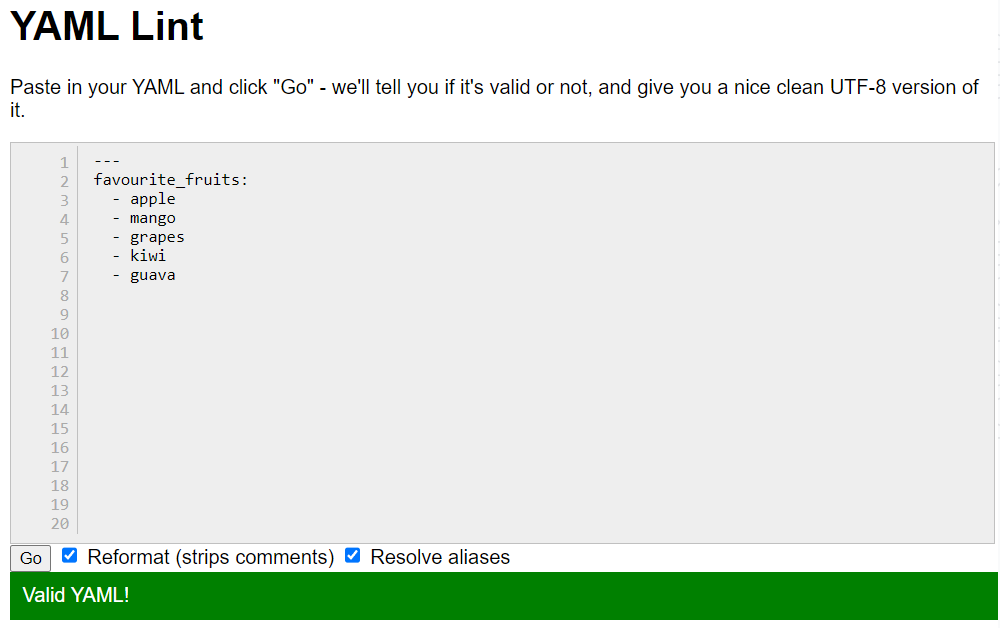
Step 4 :

Save and validate the YAML file.





Online Editor :



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

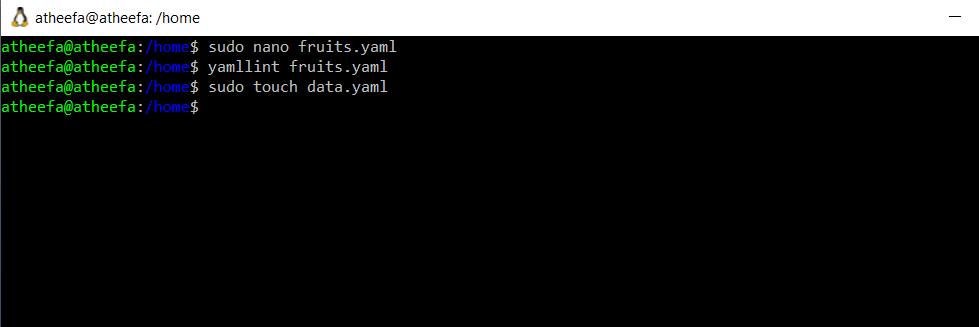
structure.

4. Save and validate the YAML file.

Solution :

Step 1 :

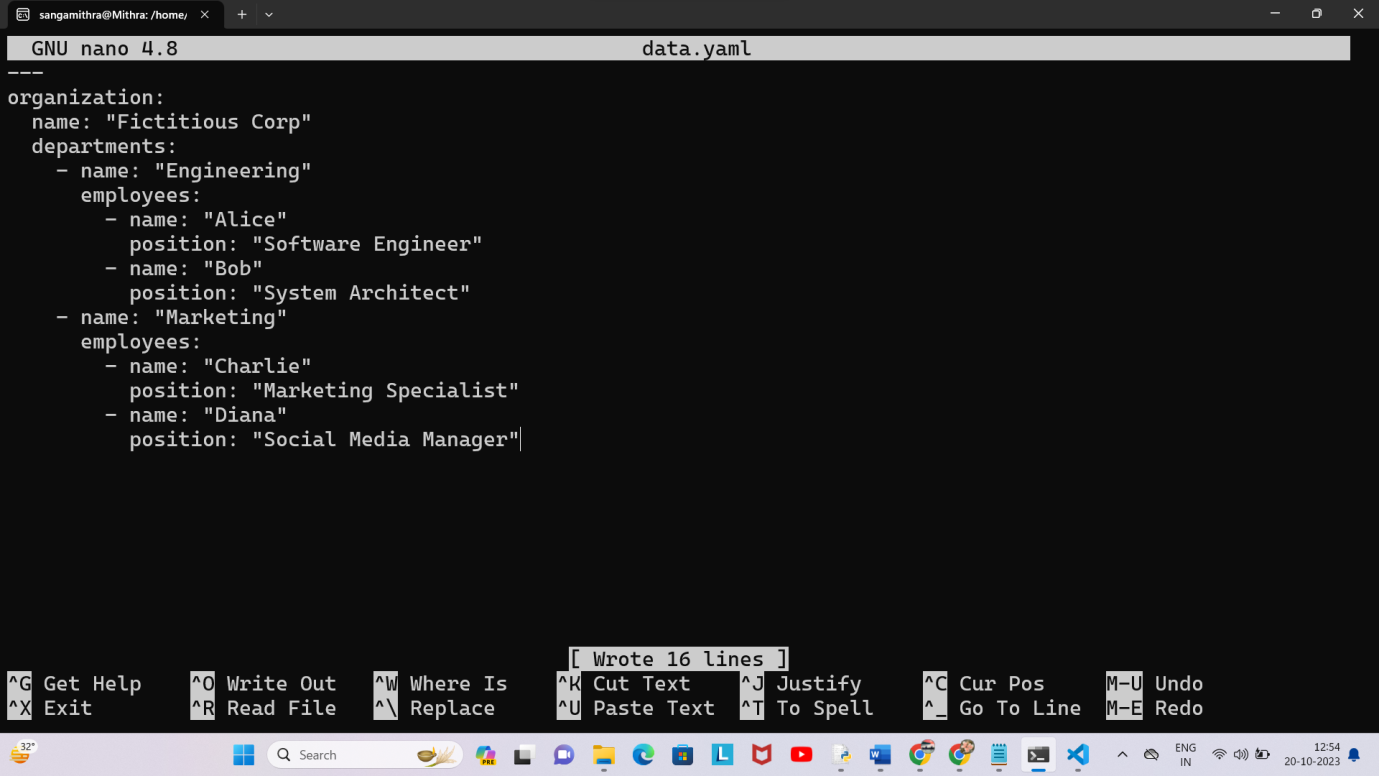
Create a YAML file named "data.yaml"



Step 2 :

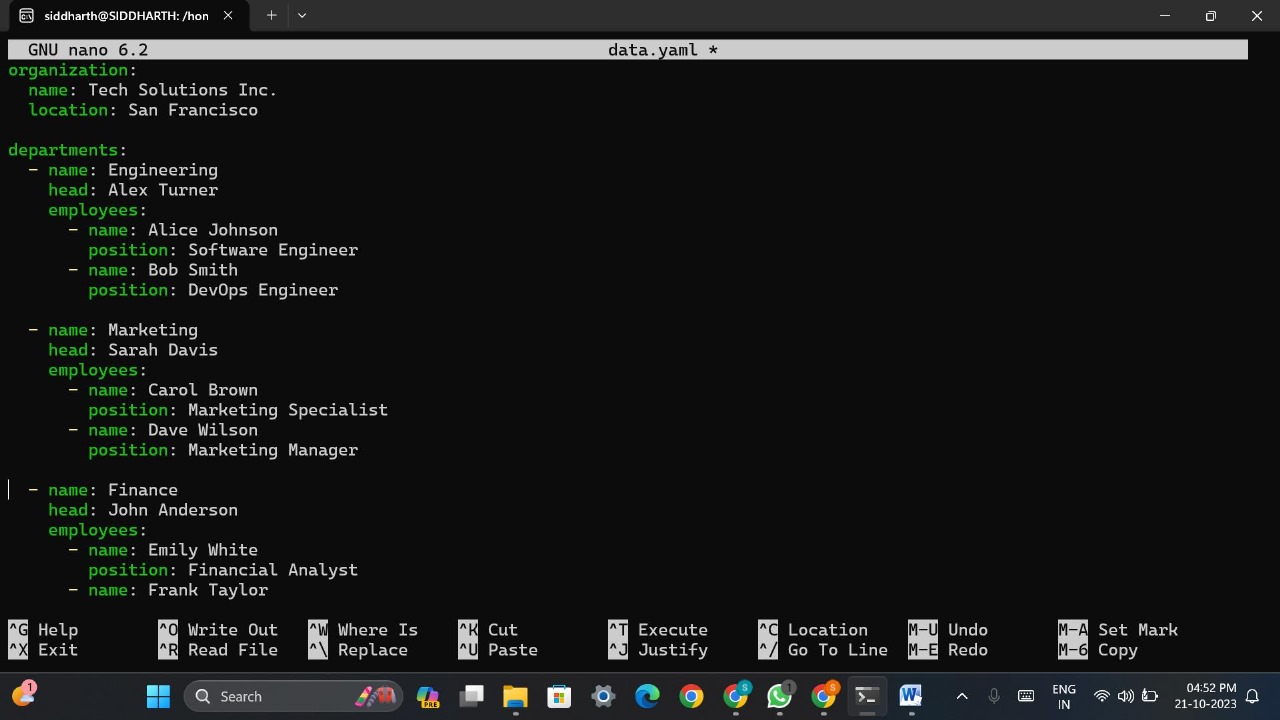
Define a nested structure representing a fictitious organization

1b.png



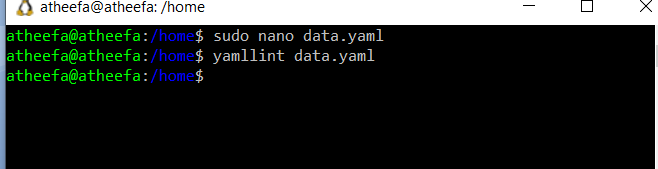
Step 3 :

Use YAML syntax to add, update, or remove data within the nested structure

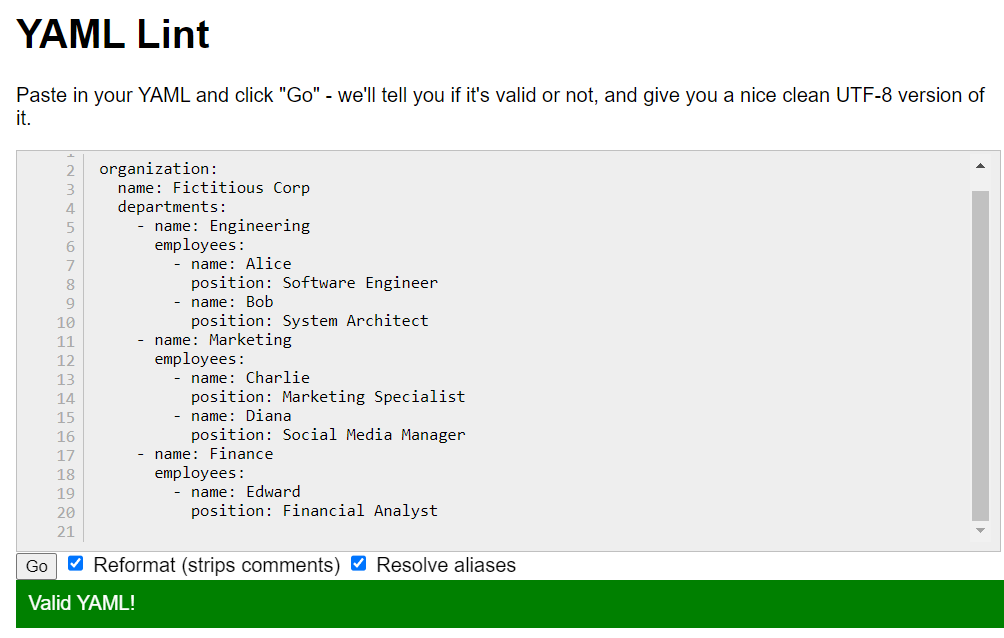


Step 4 :

Save and validate the YAML file



Online Editor :



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.

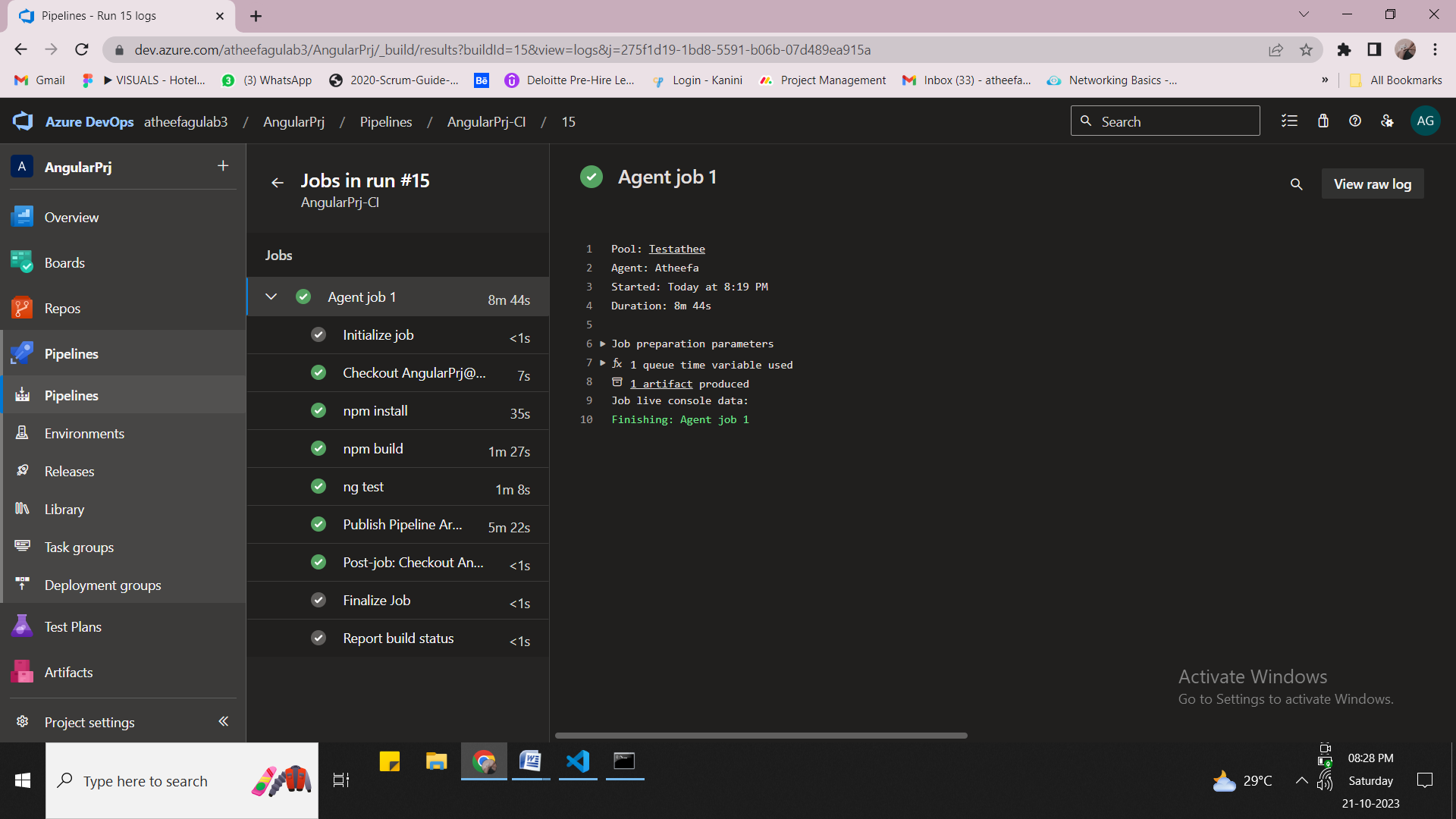
2. Set up a classic CI pipeline to build an Angular application.

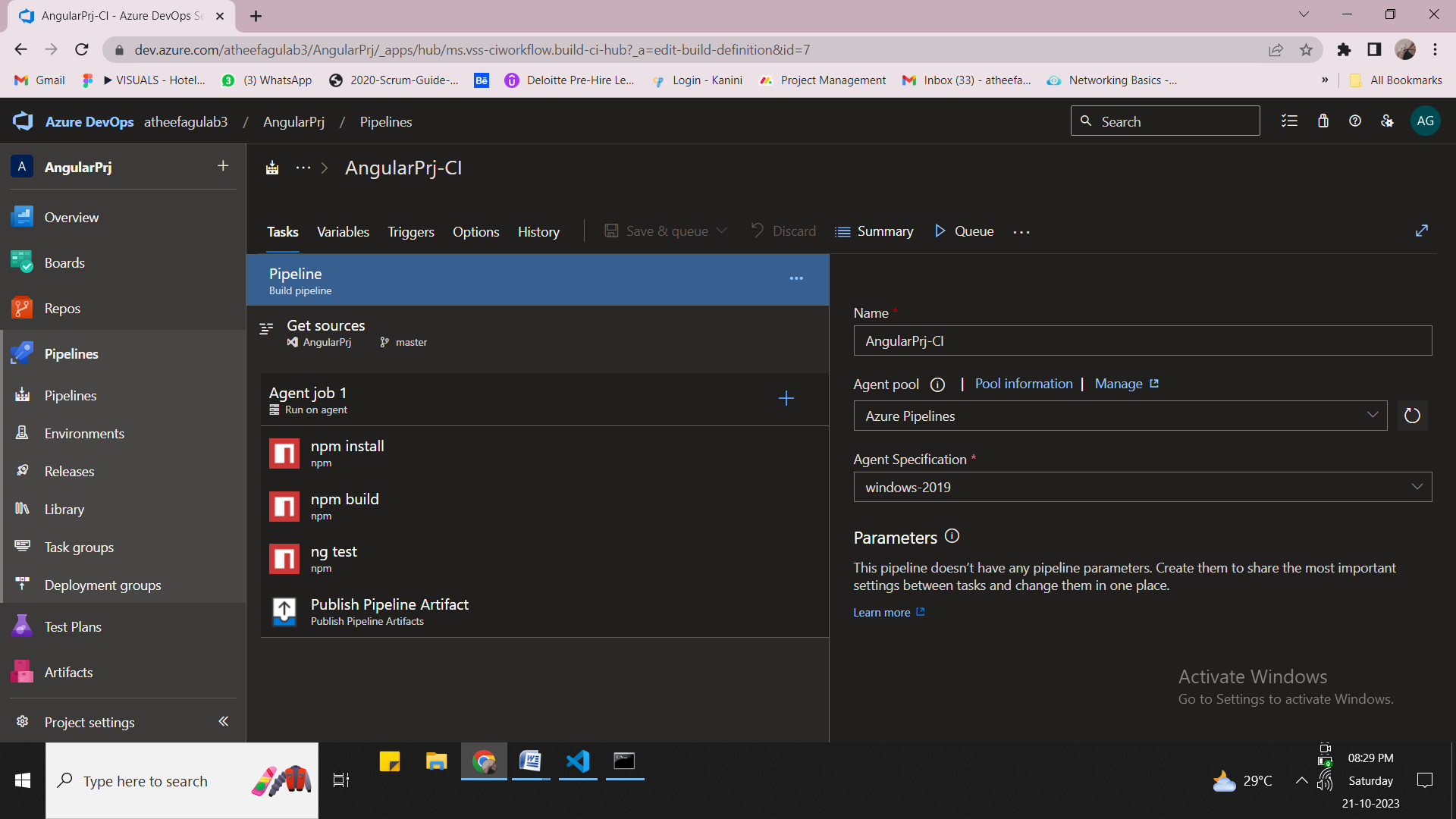
3. Configure the pipeline to use Jasmine and Karma for unit testing.

4. Run the pipeline and validate the test results.

Solution :

Screenshots :





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.

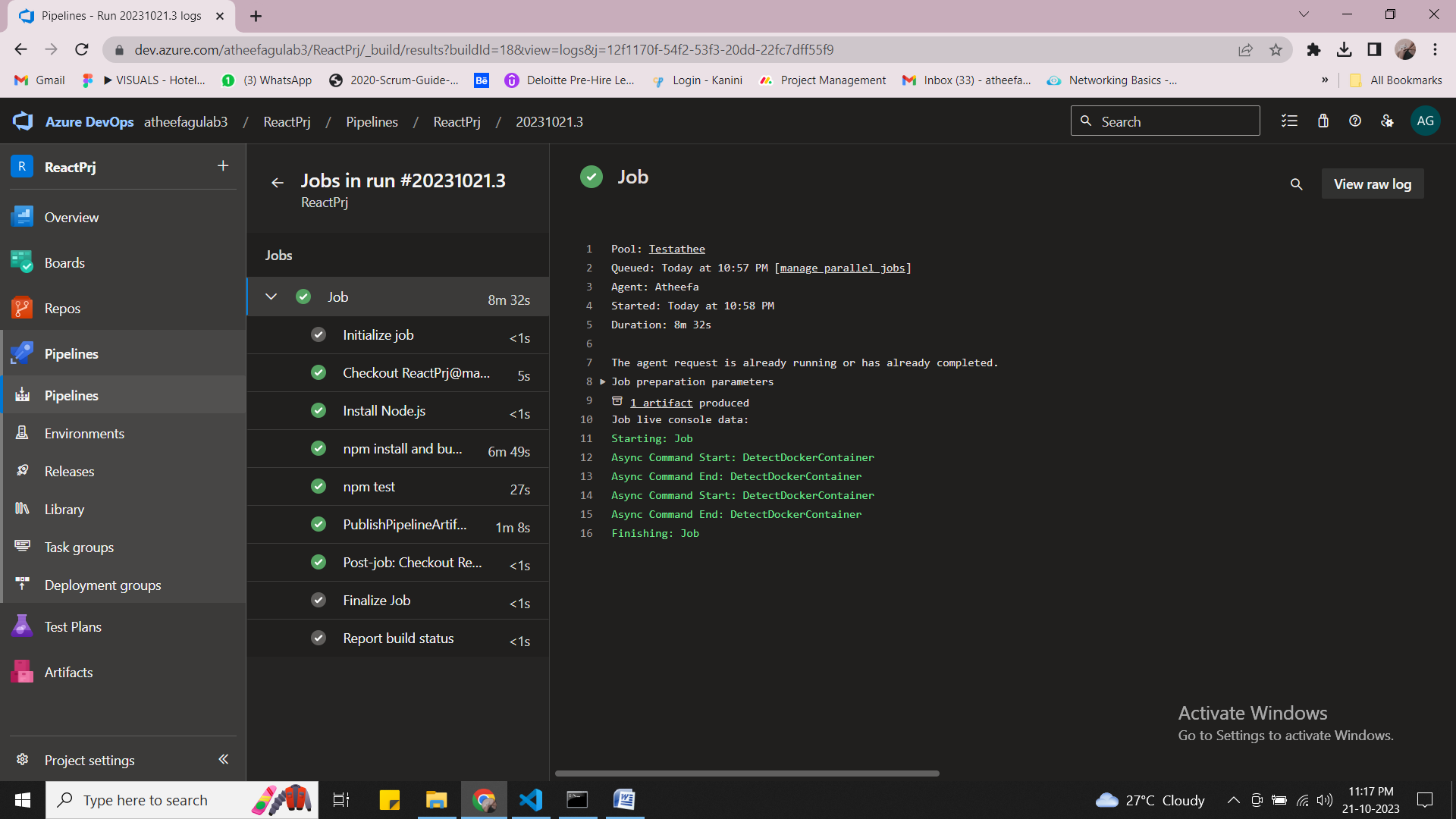
2. Create a YAML-based CI pipeline to build a React application.

3. Configure the pipeline to use Enzyme and Jest for unit testing.

4. Trigger the pipeline and verify the test results.

Solution :

Screenshots :



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.

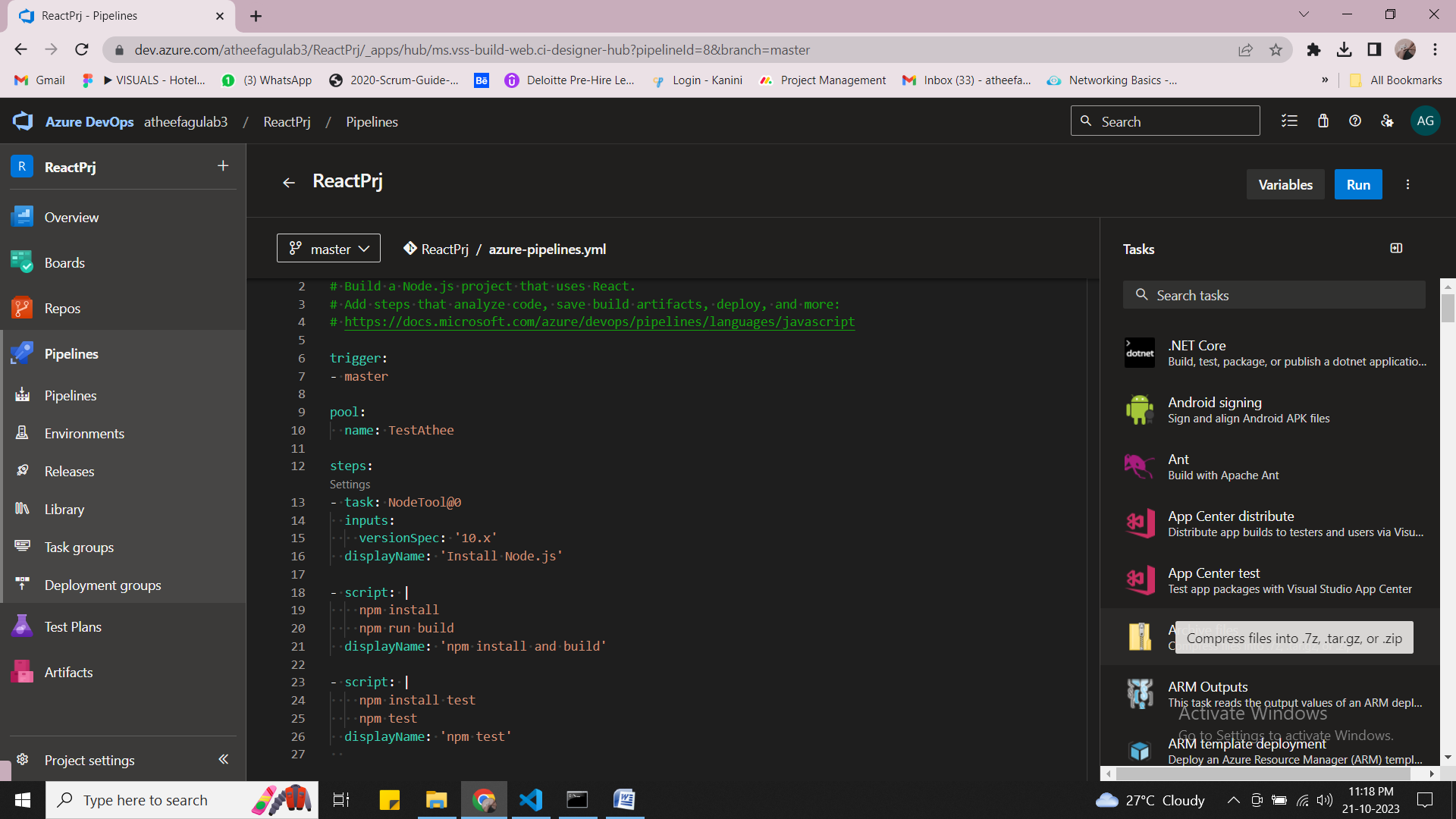
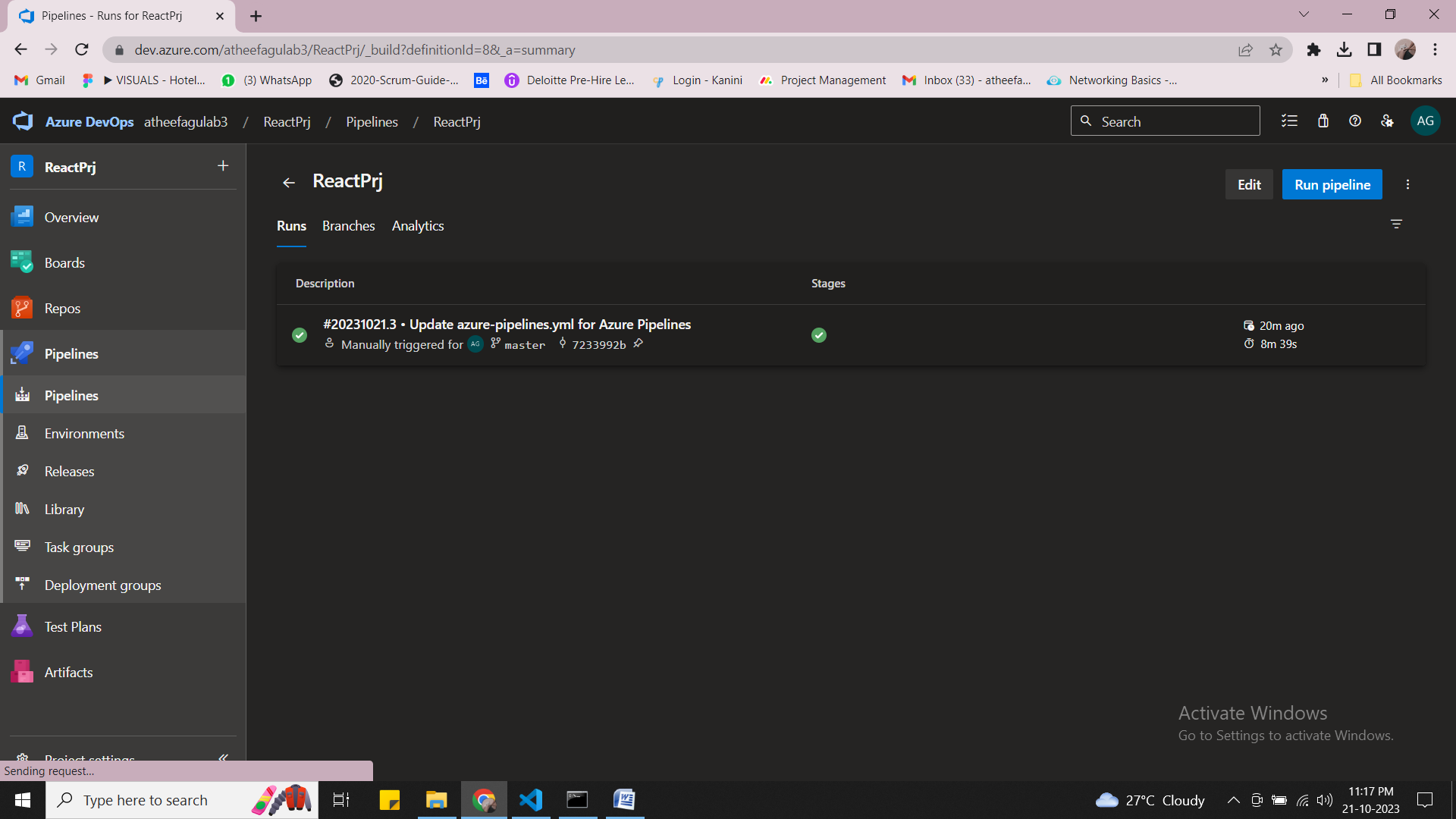
Tasks:

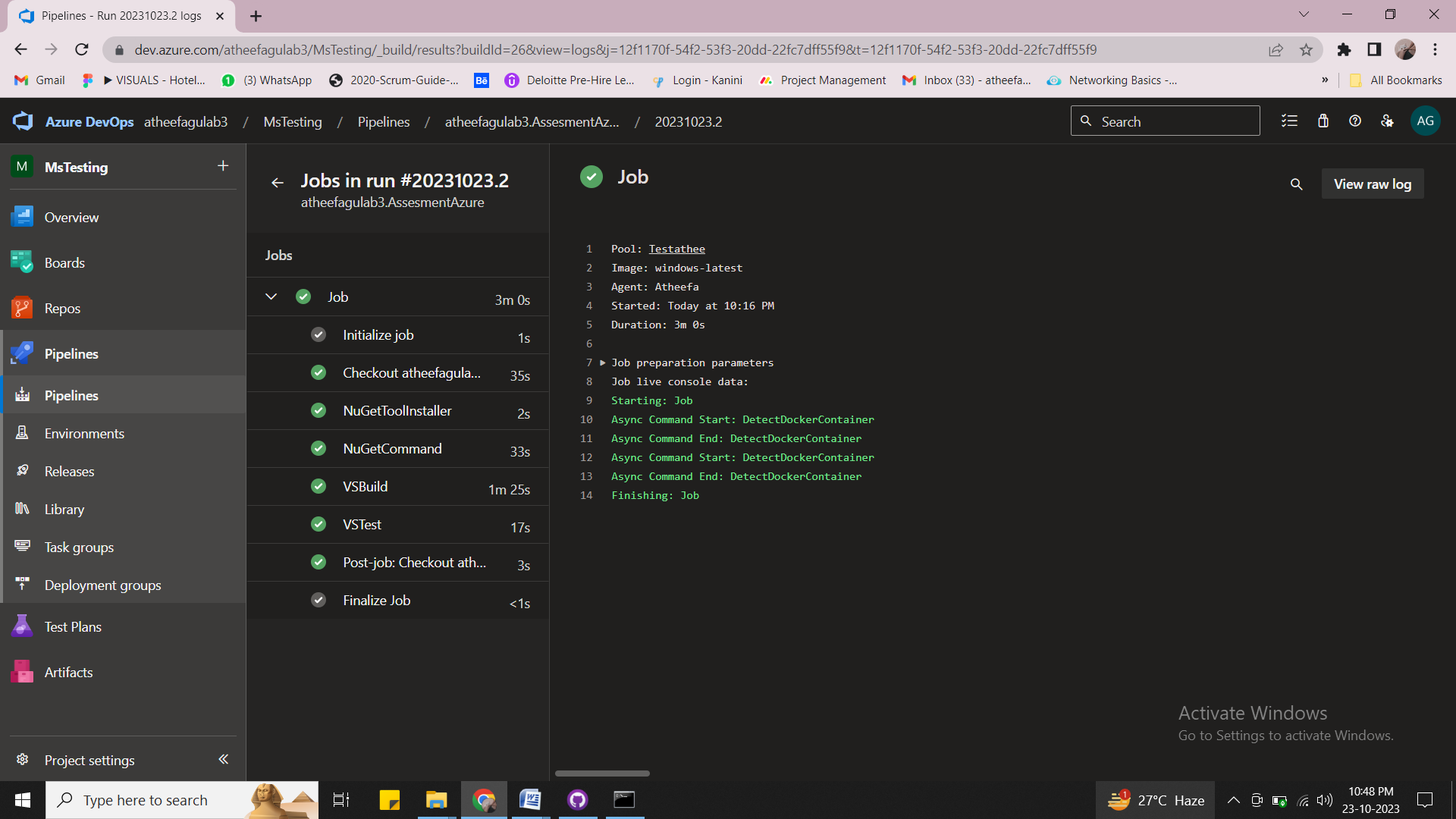
1. Set up a new Azure DevOps project.

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.





10. 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

