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

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

Ans : Option C – Windows Subsystem for Linux

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

Ans : Option b - Frequent integration of code changes

Explanation : CI involves automatically integrating and testing code changes from multiple contributors frequently, usually several times a day.This helps to find the integration issues early in the development process .

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

Ans : Option b – Change the Working directory

Explanation : “cd “ stands for change directory When we use the "cd" command followed by a directory path, it changes your current working directory to the specified directory, allowing you to navigate to the particular file or directory form the command line .

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

Ans : Option c – Docker

Explanation : Docker is a platform for developing , shipping and running the applications in the form of containers , so its not a Linux distribution.

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

Ans : Option c - Packaging and deploying applications in containers

Explanation : Docker is primarily used for packaging applications and their dependencies into containers which can be easily deployed across different environments.

6. What is the primary purpose of Azure DevOps?

Ans : Option b – Software development and delivery

Explanation : Azure DevOps provides development tools and services by Microsoft for software development, testing, and delivery.

7. Which components are part of Azure DevOps?

Ans : Option c - Azure Boards and Azure Pipelines

8. How does Azure DevOps support version control in software development?  
  
Ans : Option b. It tracks changes in source code and manages versions.

Explanation : It offers version control systems like Git and features like branching , merging , and code review .

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

Ans : Option c - Administrative tasks with superuser privileges

Explanation : The root user has complete control over the system and can perform actions that can potentially impact the entire system, such as installing software

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

c. Azure Pipelines

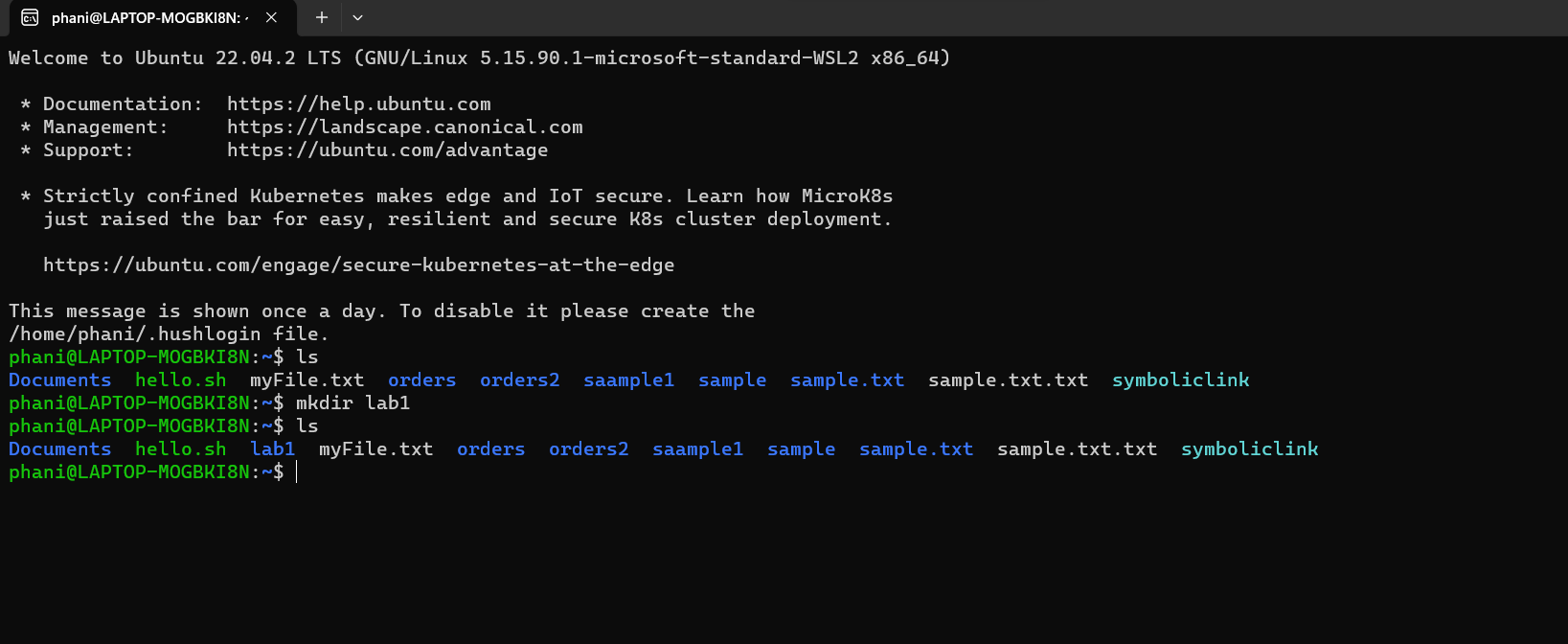
Explanation :   
In Azure DevOps, Azure Pipelines is the component used to define, build, test, and deploy applications.

**Section 2: Labs**

**Lab 1: File and Directory Management**

Task 1 : Create a directory called "lab1" in your home directory.

Ans : we can use the command “mkdir directory\_name” to create a empty directory in a particular location . The cmd “mkdir” stands for Make Directory



Task 2 “ Inside "lab1," create a text file named "sample.txt" with some content.

Ans :

Step 1: using “ cd lab1” you can navigate inside to the lab1 directory

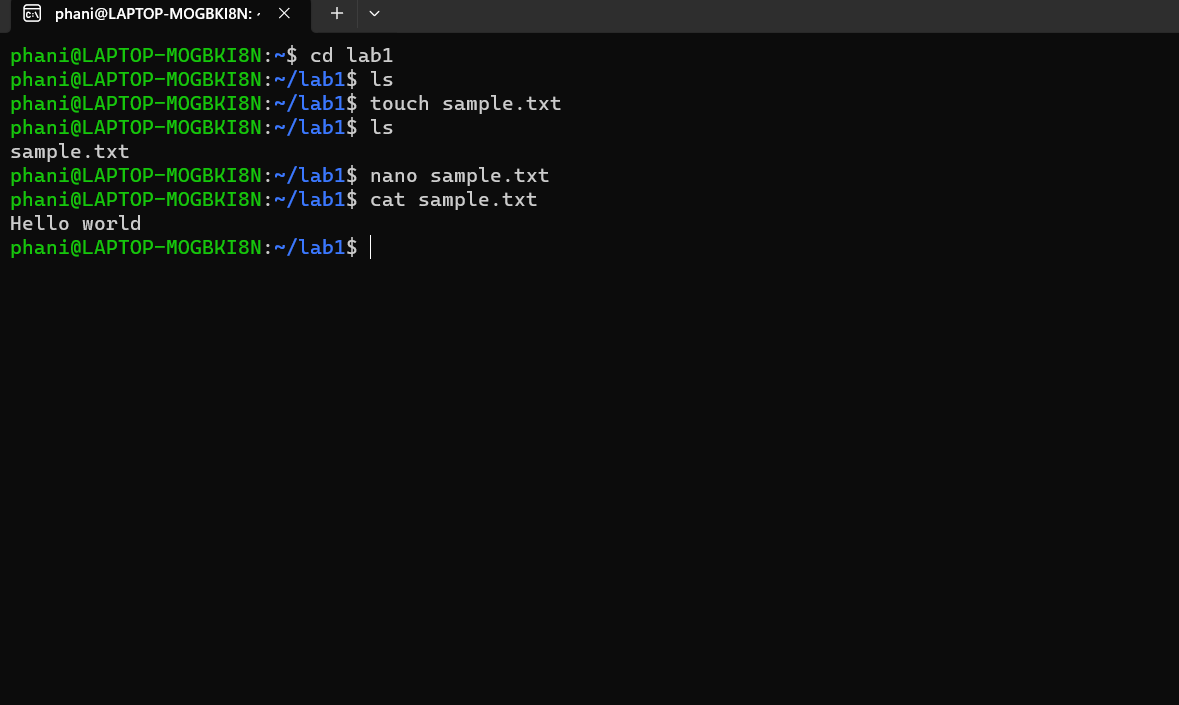
Step 2 : using “touch sample.txt ” you can create a text file named sample .

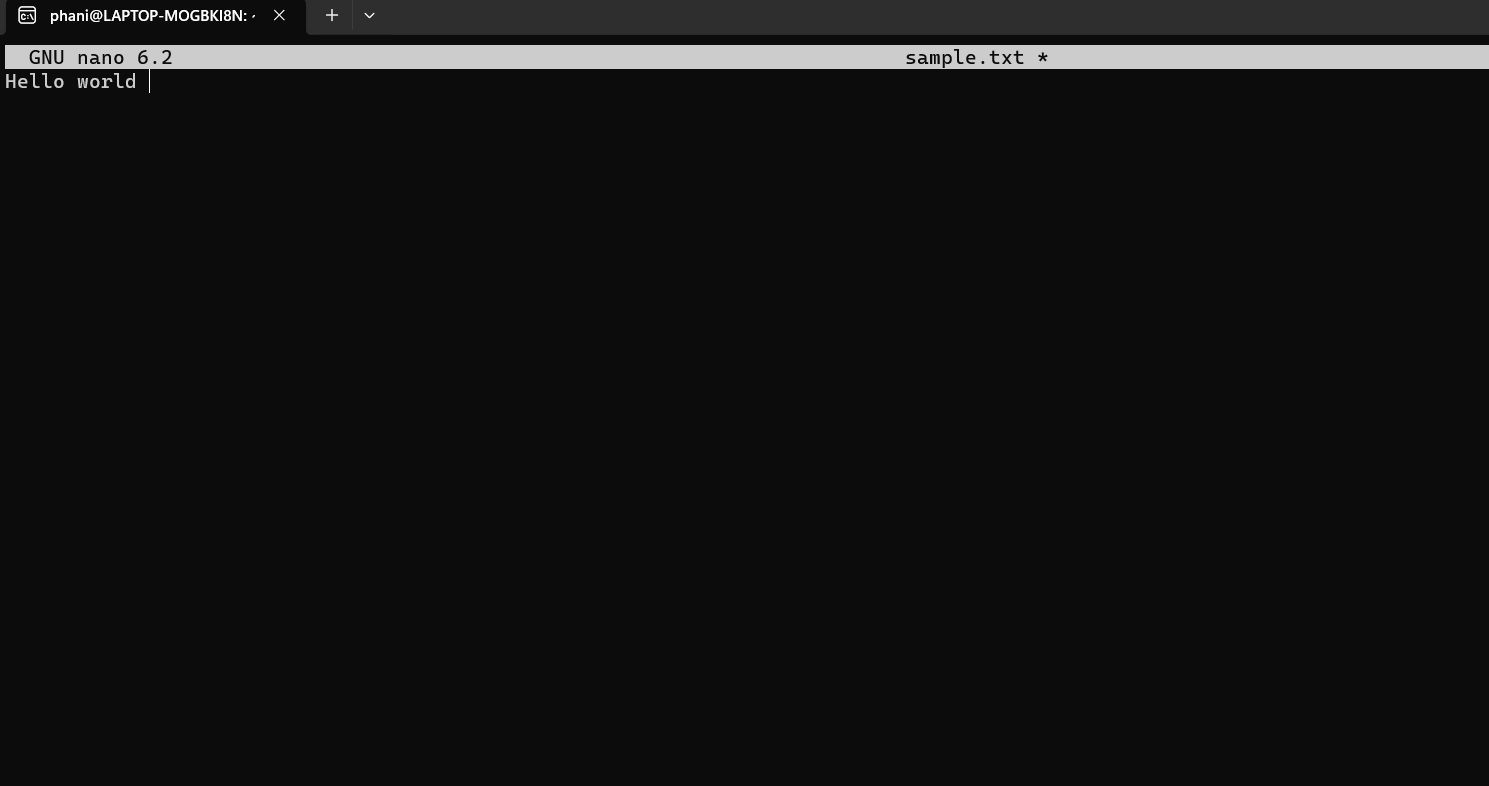
Step 3 :open the file using the nano text editor using “sudo nano sample.txt”

Step 4 : Write the content inside the sample.txt

Step 5 : we can use ctrl+o to save and ctrl+x to exit from the text editor

Step 6 : To view the contents inside the sample text use “cat sample.txt”





Task 3 : Make a copy of "sample.txt" and name it "sample\_copy.txt."

To make a copy of "sample.txt" and name it "sample\_copy.txt" in the "lab1" directory, we can use the **“cp”** command.

Command to execute :

cp sample.txt sample\_copy.txt

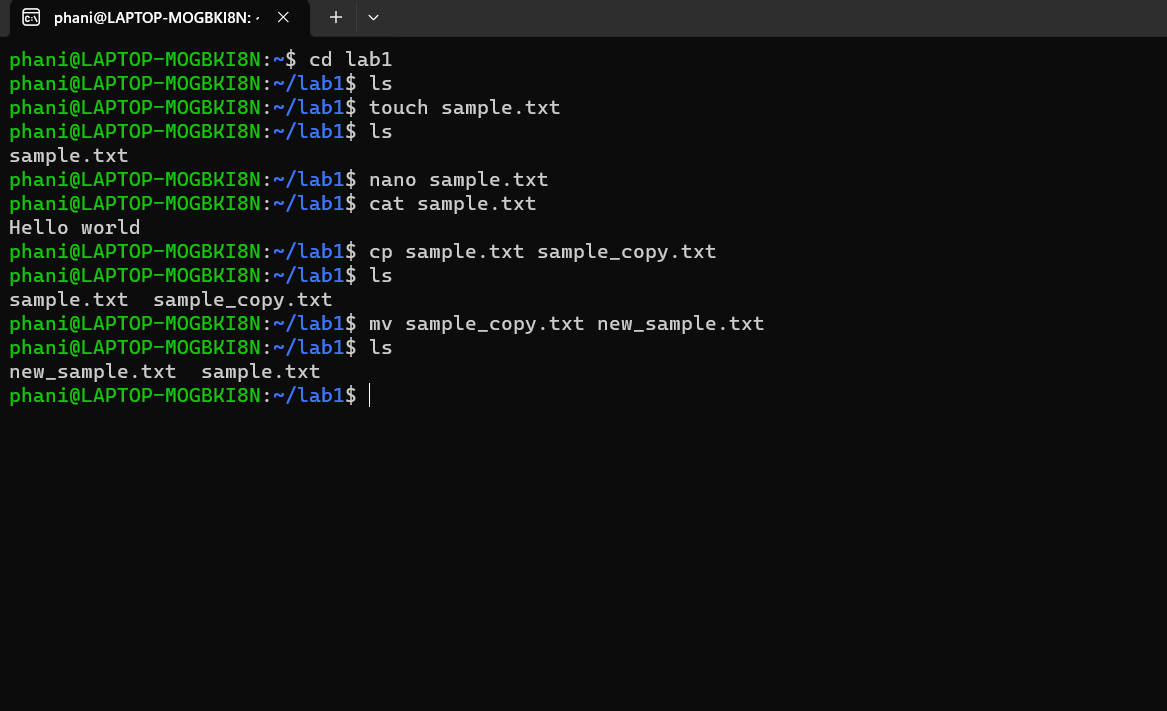
Task 4 :Rename "sample\_copy.txt" to "new\_sample.txt."

To rename a file we can use the following command :

“ mv original\_file\_name new\_file \_name “

Task 5: List the files in the "lab1" directory to confirm their names.

We can use the ls command to list the files inside a particular directory



**Lab 2: Permissions and Ownership**

Objective: Understand and manage file permissions and ownership.

Tasks:

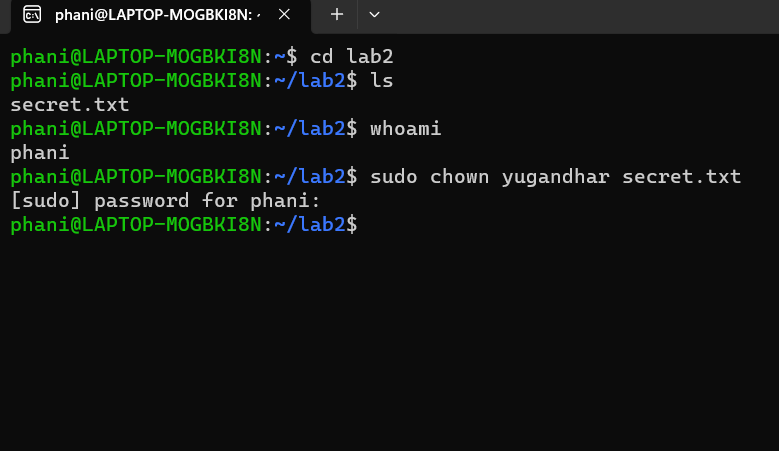
1.Create a new file named "secret.txt" in the "lab2" directory.

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

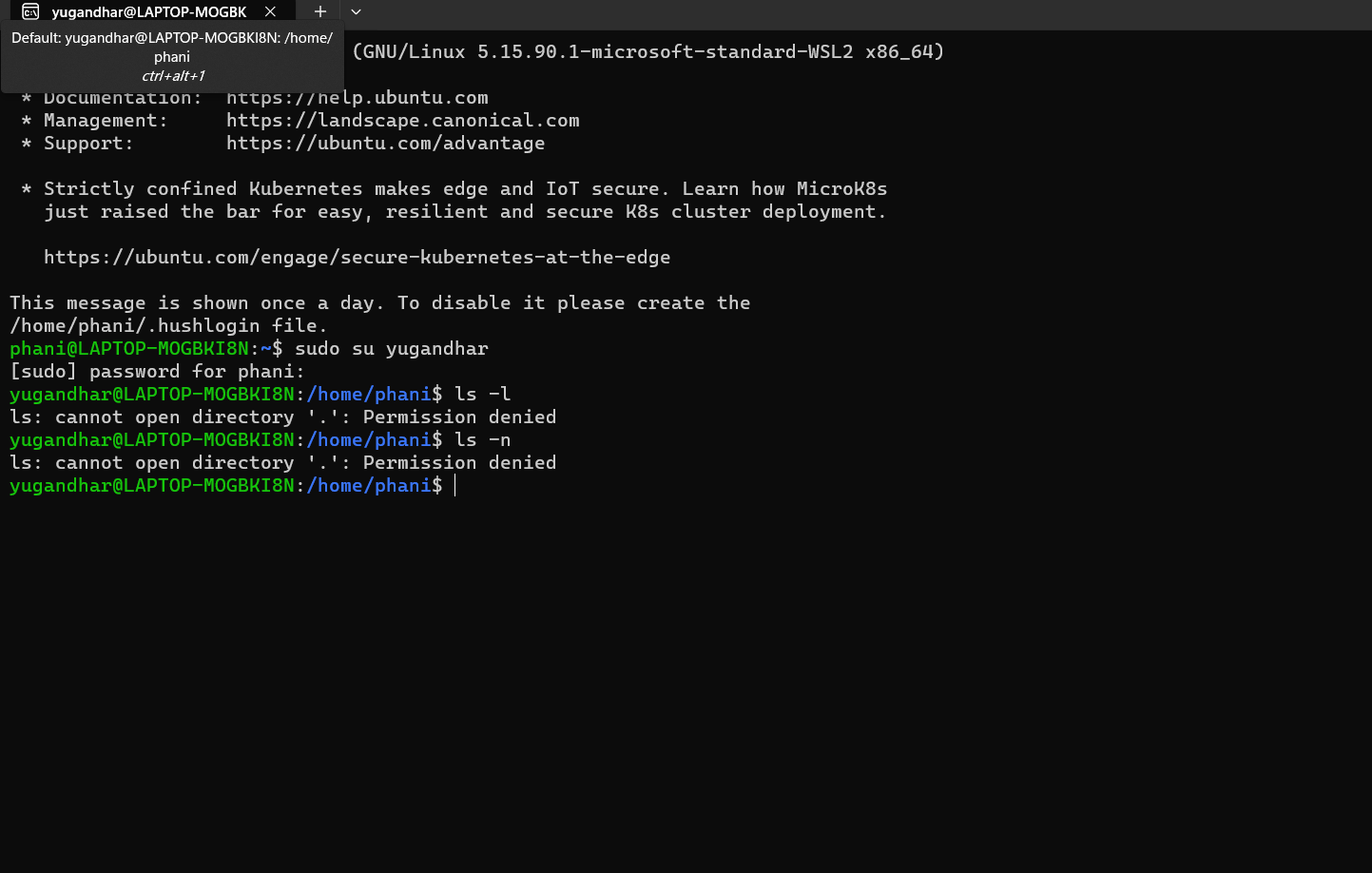
A screenshot of a computer

Description automatically generated

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



4.Ve0rify the new permissions and owner using the **ls -l** and **ls -n** commands.



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

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.

A screenshot of a computer

Description automatically generated

**Lab 4: Creating a Simple YAML File**

Objective: Create a basic YAML configuration file.

Task:

Create a YAML file named "config.yaml."

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

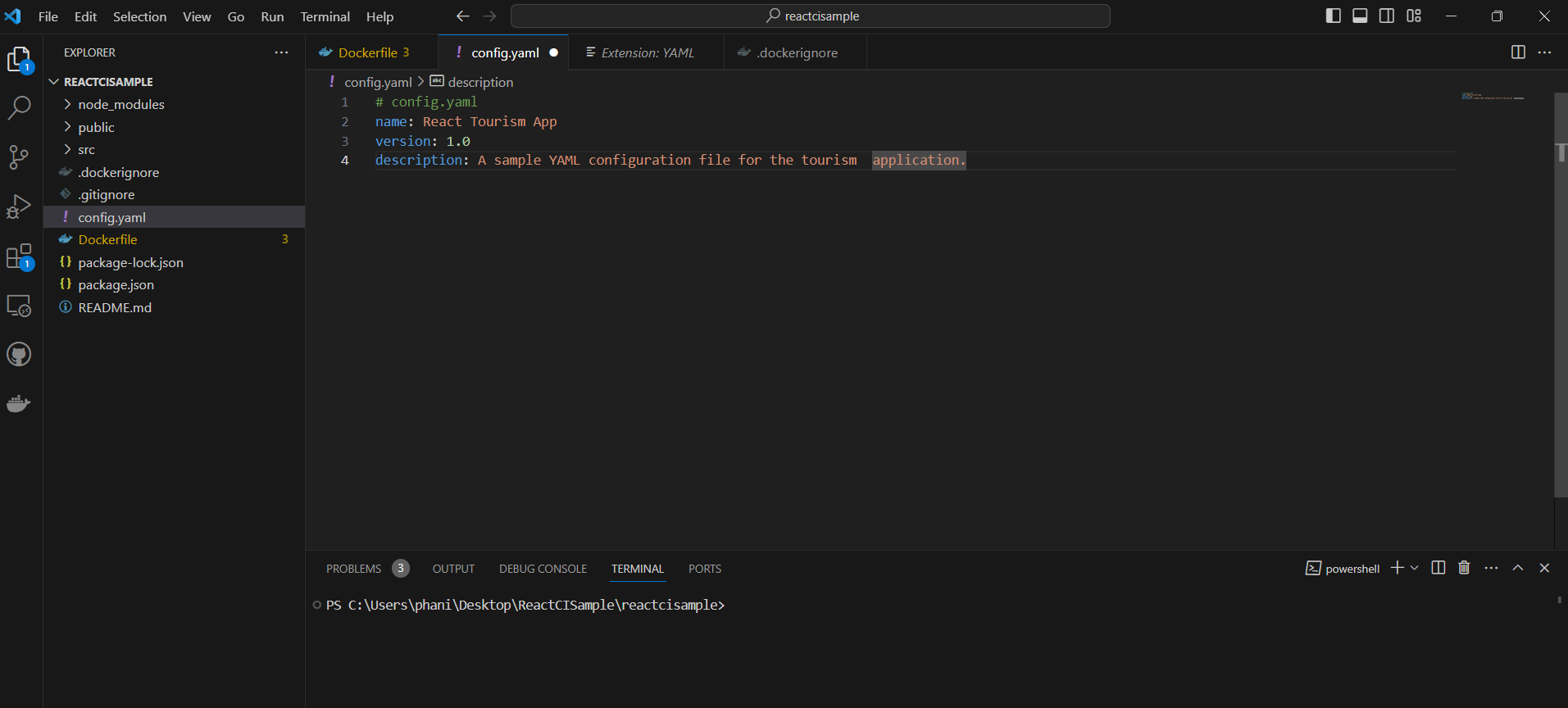
Save the file.

Validate that the YAML file is correctly formatted.

We can install an extension like "YAML" to automatically validate YAML files

While we editing the files itself .

In this case I have installed the extension named “YAML” to validate the YAML cos I have written.



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

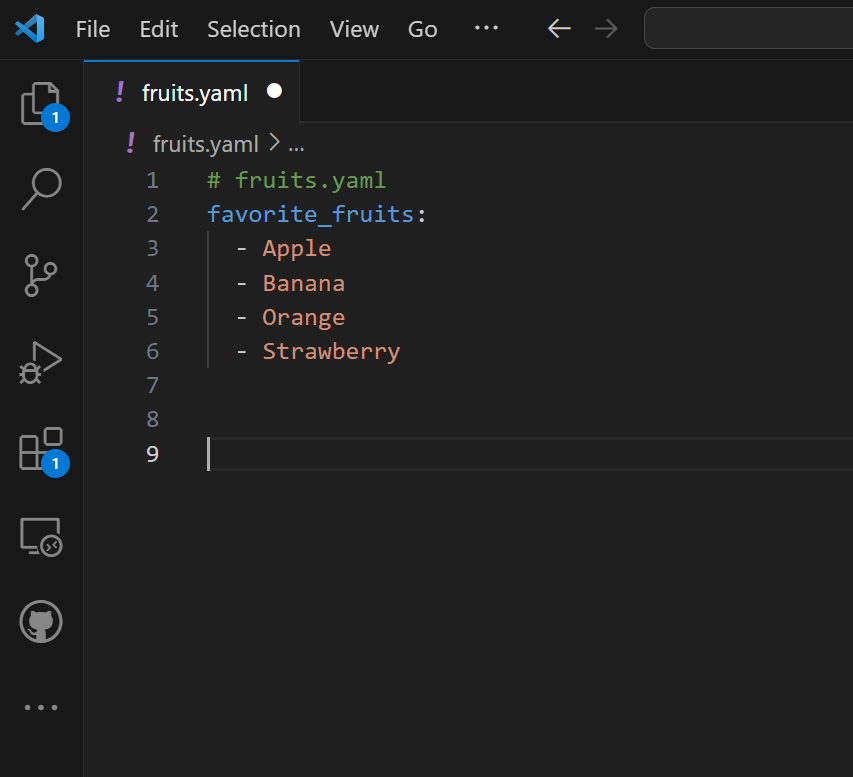
Objective: Practice working with lists (arrays) in YAML.

Task:

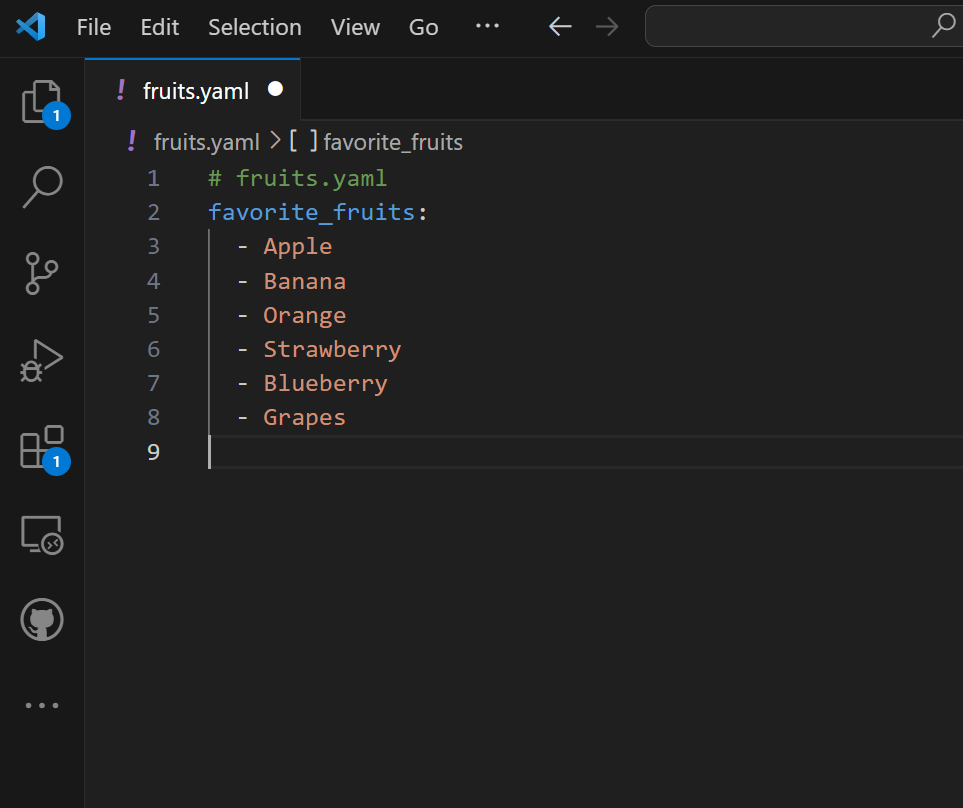
1.Create a YAML file named "fruits.yaml."

2.Define a list of your favorite fruits using YAML syntax.

YAML uses indentation and dashes (**-**) to create lists.



3.Add items from the list.



4..Save and validate the YAML file.

**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.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

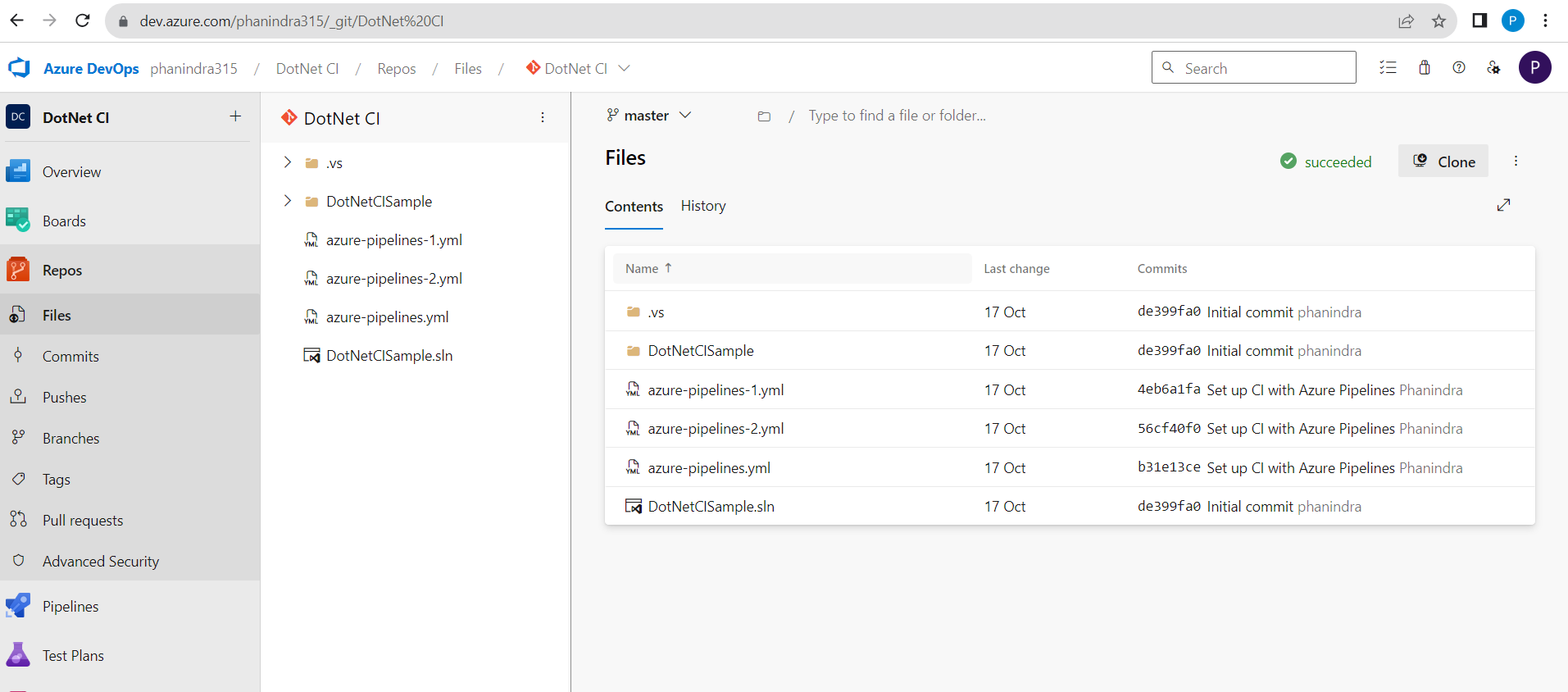
Description automatically generated

A screenshot of a computer

Description automatically generated

**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.



A screenshot of a computer

Description automatically generated

**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.

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated