# 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

   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: 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

   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.

# 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.
2. Now, list the directories using **ls,** move to lab1 using **cd,** create a file using **touch.**
3. Make a copy of sample.txt and name it sample_copy.txt using **cp**
4. Rename it to new_sample.txt using **mv**

5. List using **ls.**

```
piriya@PIRIYADHARSHINI: ~/
piriya@PIRIYADHARSHINI:/mnt/c/WINDOWS/system32$ cd
piriya@PIRIYADHARSHINI:~$ mkdir Dev
piriya@PIRIYADHARSHINI:~$ ls
Dev  File1  Folder1  file1  file1.txt  lab1
piriya@PIRIYADHARSHINI:~$ cd Dev
piriya@PIRIYADHARSHINI:~/Dev$ touch sample.txt
piriya@PIRIYADHARSHINI:~/Dev$ echo "Hello!!" > sample.txt
echo "Hellotouch sample.txt" > sample.txt
piriya@PIRIYADHARSHINI:~/Dev$ cp sample.txt sample_copy.txt
piriya@PIRIYADHARSHINI:~/Dev$ mv sample_copy.txt new_sample.txt
piriya@PIRIYADHARSHINI:~/Dev$ ls
new_sample.txt  sample.txt
piriya@PIRIYADHARSHINI:~/Dev$
```

## 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**
2. Set file permissions to allow read, write access using **chmod 600**
3. Change owner using **chown**
4. Verify new permission using **ls -l , ls -n**

```
piriya@PIRIYADHARSHINI:~/Dev$ mkdir lab2
piriya@PIRIYADHARSHINI:~/Dev$ cd lab2
piriya@PIRIYADHARSHINI:~/Dev/lab2$ cd
piriya@PIRIYADHARSHINI:~$ mkdir lab2
piriya@PIRIYADHARSHINI:~$ cd lab2
piriya@PIRIYADHARSHINI:~/lab2$ touch secret.txt
piriya@PIRIYADHARSHINI:~/lab2$ chmod 600 secret.txt
piriya@PIRIYADHARSHINI:~/lab2$ chown Piriya:Piriya secret.txt
```

```
piriya@PIRIYADHARSHINI:~/lab2$ chown piriya:piriya secret.txt
piriya@PIRIYADHARSHINI:~/lab2$ ls -l secret.txt
-rw------- 1 piriya piriya 0 Oct 23 21:53 secret.txt
piriya@PIRIYADHARSHINI:~/lab2$ ls -n secret.txt
-rw------- 1 1000 1000 0 Oct 23 21:53 secret.txt
piriya@PIRIYADHARSHINI:~/lab2$
```

## 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**
2. Use **grep** to search for a word
3. Use **sed** to replace a word
4. Use **wc** to count no.of lines

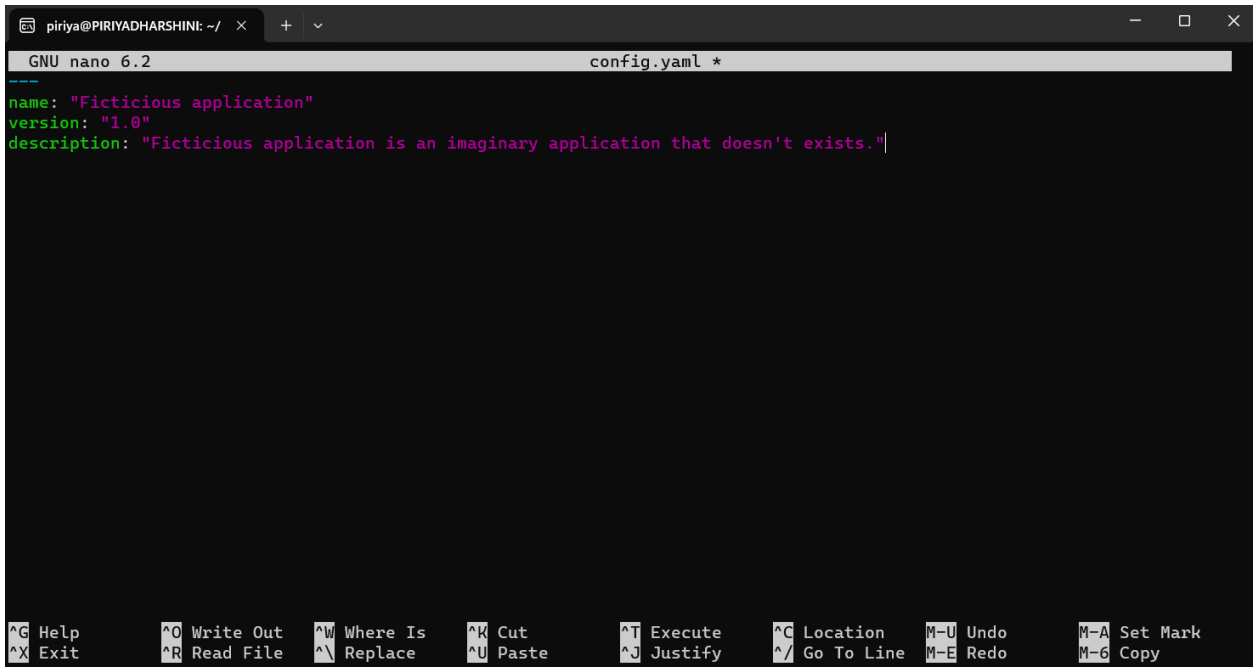
```
piriya@PIRIYADHARSHINI:~/lab2$ touch sample.txt
piriya@PIRIYADHARSHINI:~/lab2$ echo "Heyy Kanini!!"
echo "Heyy Kaninitouch sample.txt"
Heyy Kaninitouch sample.txt
piriya@PIRIYADHARSHINI:~/lab2$ echo "Heyy Kanini!!" > sample.txt
echo "Heyy Kaniniecho "Heyy Kaninitouch sample.txt"" > sample.txt
piriya@PIRIYADHARSHINI:~/lab2$ grep "chennai" sample.txt
piriya@PIRIYADHARSHINI:~/lab2$ grep "Heyy" sample.txt
Heyy Kaniniecho Heyy Kaninitouch sample.txt
piriya@PIRIYADHARSHINI:~/lab2$ sed -i "s/piriya/piri/g" sample.txt
piriya@PIRIYADHARSHINI:~/lab2$ cat sample.txt
Heyy Kaniniecho Heyy Kaninitouch sample.txt
piriya@PIRIYADHARSHINI:~/lab2$ wc sample.txt
 1  5 44 sample.txt
piriya@PIRIYADHARSHINI:~/lab2$
```

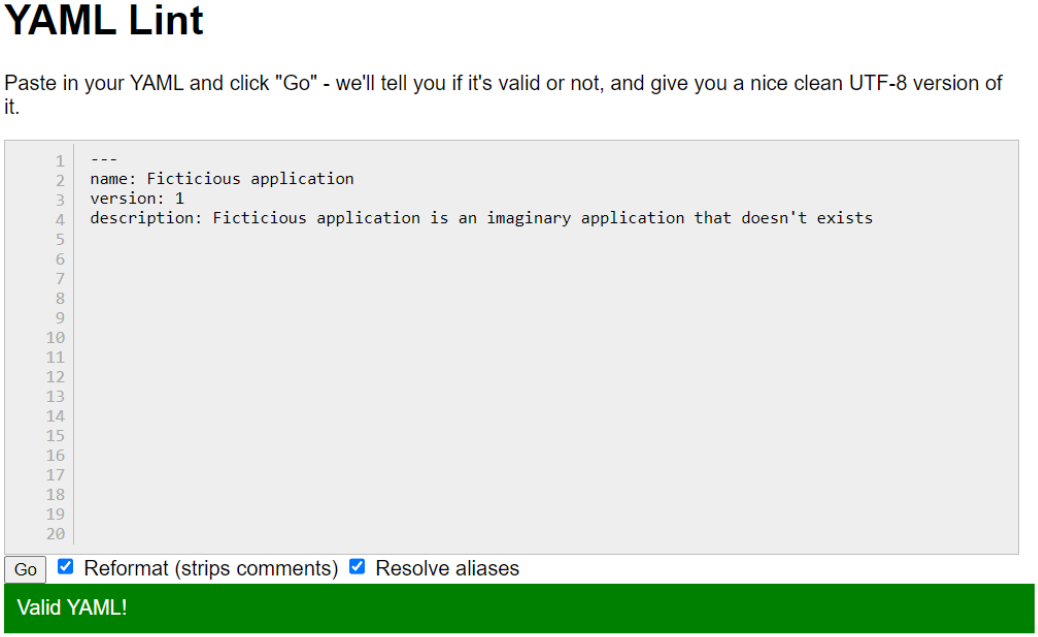
## Lab 4: Creating a Simple YAML File

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.

```
piriya@PIRIYADHARSHINI:~$ sudo mkdir yaml
[sudo] password for piriya:
piriya@PIRIYADHARSHINI:~$ cd yaml
piriya@PIRIYADHARSHINI:~/yaml$ sudo touch config.yaml
piriya@PIRIYADHARSHINI:~/yaml$
```




## 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:

```
piriya@PIRIYADHARSHINI:/home/yaml$ sudo touch fruits.yaml
piriya@PIRIYADHARSHINI:/home/yaml$
```

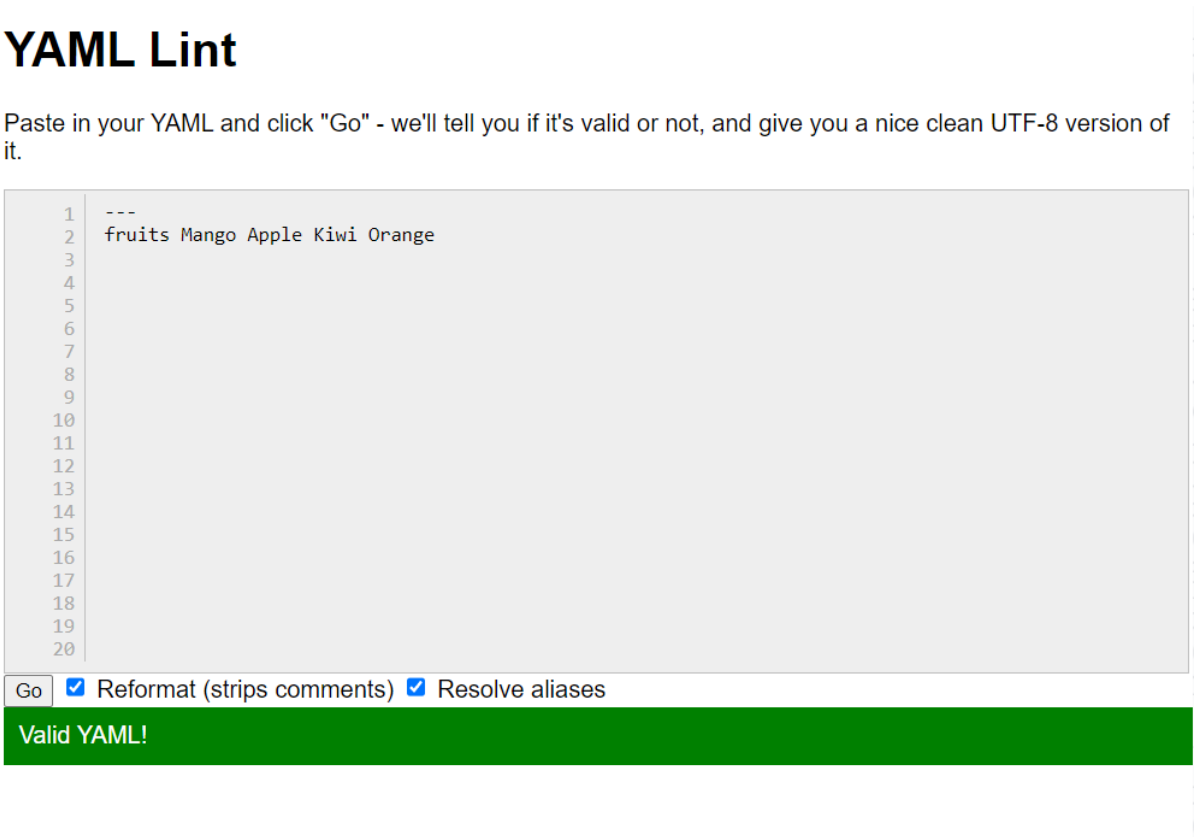
```
piriya@PIRIYADHARSHINI: /h
  GNU nano 6.2                                   fruits.yaml *
---
favourite_fruits:
Mango
Apple
Orange

^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

```
piriya@PIRIYADHARSHINI:/mnt/c/WINDOWS/system32$ cd
piriya@PIRIYADHARSHINI:~$ pwd
/home/piriya
piriya@PIRIYADHARSHINI:~$ cd /home
piriya@PIRIYADHARSHINI:/home$ cd yaml
piriya@PIRIYADHARSHINI:/home/yaml$ cat fruits.yaml
piriya@PIRIYADHARSHINI:/home/yaml$ sudo nano
[sudo] password for piriya:
piriya@PIRIYADHARSHINI:/home/yaml$ cat fruits
Fruits -
Mango
Apple
Orange
piriya@PIRIYADHARSHINI:/home/yaml$
```

- Adding items to the list:

```
piriya@PIRIYADHARSHINI:/home/yaml$ sudo nano fruits.yaml
piriya@PIRIYADHARSHINI:/home/yaml$ cat fruits.yaml
fruits
Mango
Apple
Kiwi
Orange
piriya@PIRIYADHARSHINI:/home/yaml$
```




## 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.

```
piriya@PIRIYADHARSHINI:/home/yaml$ sudo touch data.yaml
piriya@PIRIYADHARSHINI:/home/yaml$ sudo nano data.yaml
piriya@PIRIYADHARSHINI:/home/yaml$ sudo nano data.yaml
piriya@PIRIYADHARSHINI:/home/yaml$ sudo nano
piriya@PIRIYADHARSHINI:/home/yaml$ cat data.yaml
---
organization:
        name: "Fictitious Corp"
        departments:
                - name: "Engineering"
                employees:
                -name: "Alice"
                -position: "Software Engineer"
                -name: "Bob"
                -position: "System Architect"
                - name: "Marketing"
                employees:
                -name: "Charlie"
                -position: "Marketing Specialist"
                - name: "Diana"
                -position: "Social Media Manager"
piriya@PIRIYADHARSHINI:/home/yaml$
```

## 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.

## 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