Devops – Final Assessment

(Shrianshi Kumari)

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

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

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

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

a. Ubuntu

b. CentOS

c. Docker

d. Debian

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

6. What is the primary purpose of Azure DevOps?

a. Infrastructure management

b. Software development and delivery

c. Network security

d. Virtualization

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

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.

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

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

Section 2: Labs

Lab 1: File and Directory Management

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

```
shrianshi@Shri-PC:~$ pwd
/home/shrianshi
shrianshi@Shri-PC:~$ cd /home/shrianshi
shrianshi@Shri-PC:~$ mkdir ~/lab1
shrianshi@Shri-PC:~$ ls
lab1
shrianshi@Shri-PC:~$
```

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

```
shrianshi@Shri-PC:~$ cd ~/lab1
shrianshi@Shri-PC:~/lab1$ nano sample.txt
shrianshi@Shri-PC:~/lab1$ ls
shrianshi@Shri-PC:~/lab1$ touch sample.txt
shrianshi@Shri-PC:~/lab1$ ls
sample.txt
shrianshi@Shri-PC:~/lab1$
```

3. Make a copy of "sample.txt" and name it "sample_copy.txt."

```
shrianshi@Shri-PC:~/lab1$ cp sample.txt sample_copy.txt
shrianshi@Shri-PC:~/lab1$ ls
sample.txt  sample_copy.txt
shrianshi@Shri-PC:~/lab1$
```

4. Rename "sample_copy.txt" to "new_sample.txt."

```
shrianshi@Shri-PC:~/lab1$ mv sample_copy.txt new_sample.txt
shrianshi@Shri-PC:~/lab1$ ls
new_sample.txt  sample.txt
```

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

```
shrianshi@Shri-PC:~/lab1$ ls
new_sample.txt  sample.txt
```

Lab 2: Permissions and Ownership

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

```
shrianshi@Shri-PC:~$ mkdir ~/lab2
shrianshi@Shri-PC:~$ cd ~/lab2
shrianshi@Shri-PC:~/lab2$ touch secret.txt
```

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

```
shrianshi@Shri-PC:~/lab2$ chmod 600 secret.txt
shrianshi@Shri-PC:~/lab2$ ls -l secret.txt
-rw------- 1 shrianshi shrianshi 0 Oct 20 12:12 secret.txt
```

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

```
shrianshi@Shri-PC:~/lab2$ sudo useradd shri
shrianshi@Shri-PC:~/lab2$ sudo chown shri:shri sample.txt
chown: cannot access 'sample.txt': No such file or directory
shrianshi@Shri-PC:~/lab2$ sudo chown shri:shri secret.txt
```

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

```
shrianshi@Shri-PC:~/lab2$ ls -l secret.txt
-rw------- 1 shri shri 0 Oct 20 12:12 secret.txt
shrianshi@Shri-PC:~/lab2$ ls -n secret.txt
-rw------- 1 1001 1001 0 Oct 20 12:12 secret.txt
```

Lab 3: Text Processing with Command Line Tools

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

```
shrianshi@Shri-PC:~$ mkdir ~/lab3
shrianshi@Shri-PC:~$ cd ~/lab3
shrianshi@Shri-PC:~/lab3$ touch random.txt
shrianshi@Shri-PC:~/lab3$ echo "Hello, I am Shri👋." > random.txt
```

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

```
shrianshi@Shri-PC:~/lab3$ grep "👋" random.txt
Hello, I am Shri👋.
```

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

```
shrianshi@Shri-PC:~/lab3$ sed "s/👋/😊/g" random.txt
Hello, I am Shri😊.
```

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

```
shrianshi@Shri-PC:~/lab3$ wc random.txt
 1  4 22 random.txt
```

Lab 4: Creating a Simple YAML File

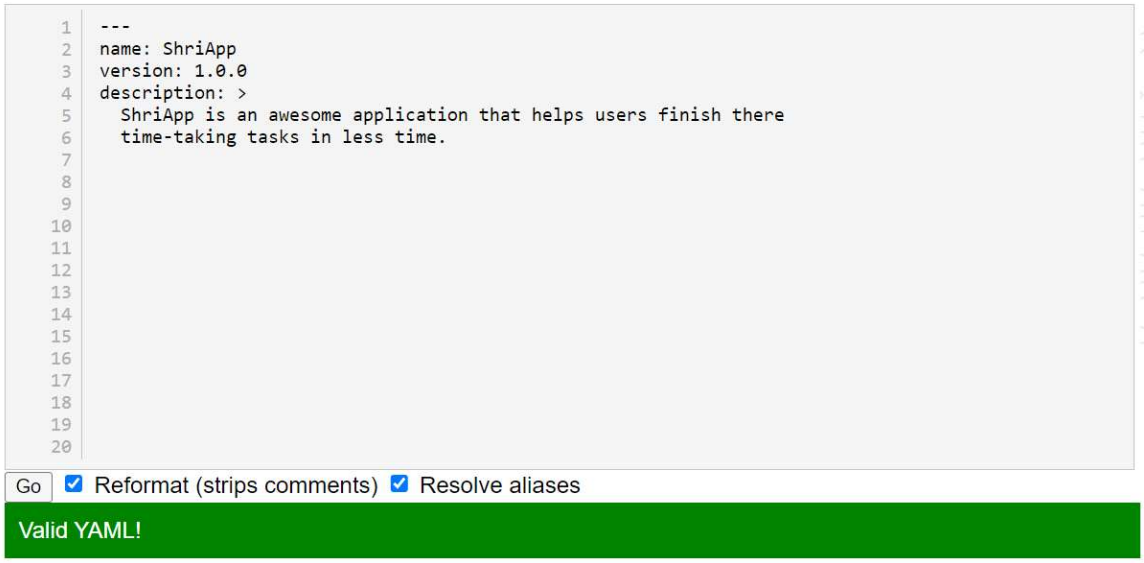
1. Create a YAML file named "config.yaml."

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.



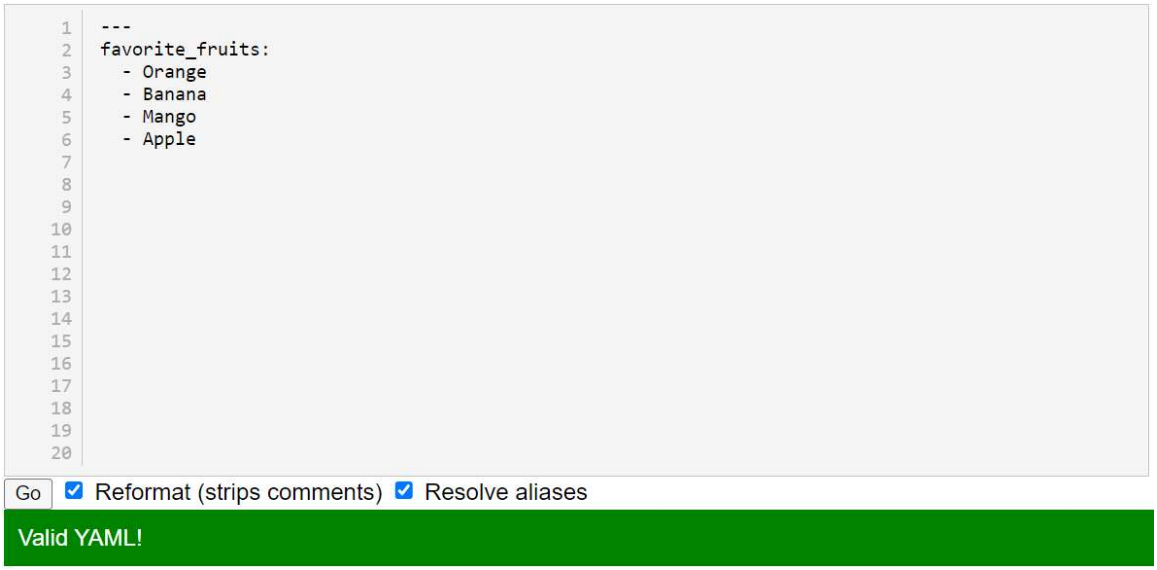
Lab 5: Working with Lists in YAML

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.



Lab 6: Nested Structures in YAML

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.

data.yaml

```yaml
KANINI SOFTWARE SOLUTIONS:
  - department:
    - name: IT
      employees:
      - name: Shri
      - name: Thrupthi

  - department:
    - name: Sales
      employees:
      - name: Pankaj
      - name: Suman

  - department:
    - name: HR
      employees:
      - name: Priya
      - name: Mozhi

```

! data.yaml

```yaml
KANINI SOFTWARE SOLUTIONS:
  - department:
    - name: IT
      employees:
      - name: Shri
      - name: Thrupthi
      - name: Swathi

  - department:
    - name: Sales
      employees:
      - name: Divya
      - name: Suman

  - department:
    - name: HR
      employees:
      - name: Priya
      - name: Mozhi
```

! data.yaml

```yaml
KANINI SOFTWARE SOLUTIONS:
  - department:
    - name: IT
      employees:
      - name: Shri
      - name: Thrupthi
      - name: Swathi

  - department:
    - name: HR
      employees:
      - name: Priya
      - name: Mozhi
```

```yaml
---
KANINI SOFTWARE SOLUTIONS:
  - department:
      - name: IT
        employees:
          - name: Shri
          - name: Thrupthi
          - name: Swathi
  - department:
      - name: HR
        employees:
          - name: Priya
          - name: Mozhi
```

Go ☑ Reformat (strips comments) ☑ Resolve aliases

Valid YAML!

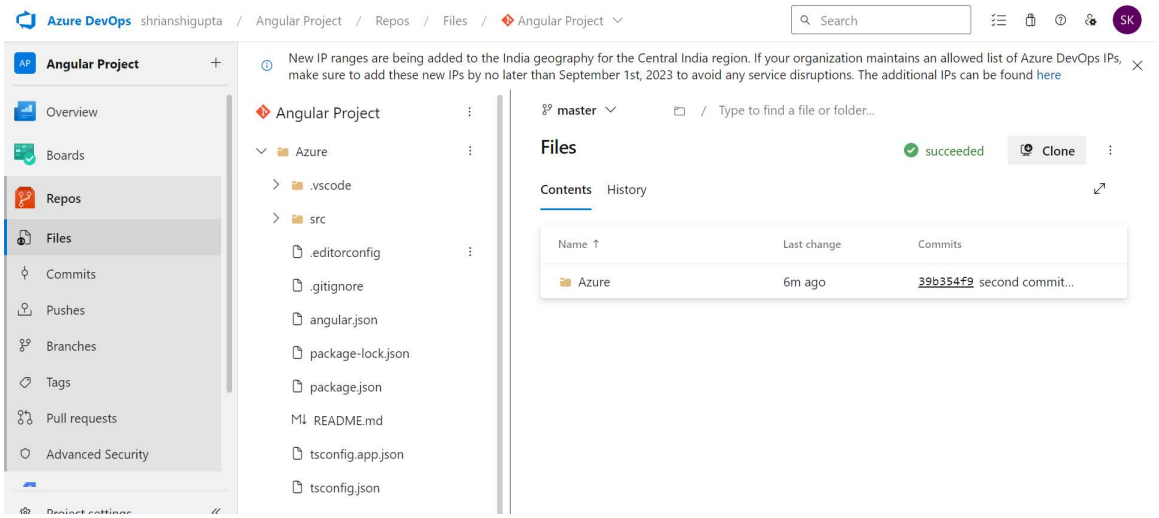
Lab 7: Create Classic Azure CI Pipeline for Angular Application

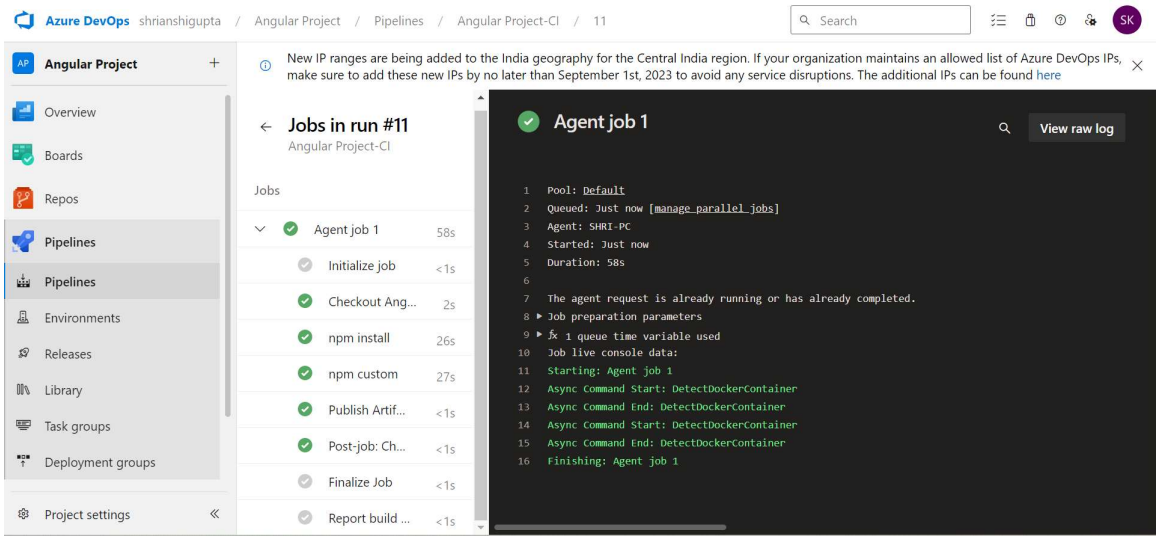
1. Create an Azure DevOps project.

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.




Azure DevOps shrianshigupta / Angular Project / Pipelines / Angular Project-CI / 11
Search
SK
Angular Project
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
Project settings
New IP ranges are being added to the India geography for the Central India region. If your organization maintains an allowed list of Azure DevOps IPs, make sure to add these new IPs by no later than September 1st, 2023 to avoid any service disruptions. The additional IPs can be found here
Jobs in run #11
Angular Project-CI
Jobs
Agent job 1 58s
Initialize job <1s
Checkout Ang... 2s
npm install 26s
npm custom 27s
Publish Artif... <1s
Post-job: Ch... <1s
Finalize Job <1s
Report build ... <1s
Agent job 1
View raw log
1 Pool: Default
2 Queued: Just now [manage parallel jobs]
3 Agent: SHRI-PC
4 Started: Just now
5 Duration: 58s
6
7 The agent request is already running or has already completed.
8 Job preparation parameters
9 1 queue time variable used
10 Job live console data:
11 Starting: Agent job 1
12 Async Command Start: DetectDockerContainer
13 Async Command End: DetectDockerContainer
14 Async Command Start: DetectDockerContainer
15 Async Command End: DetectDockerContainer
16 Finishing: Agent job 1