

DEVOPS FINAL ASSESSMENT

SECTION : 1

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

Answer: Windows Subsystem for Linux

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

Answer: Frequent integration of code changes

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

Answer: Change the working directory

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

Answer: Docker

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

Answer: Packaging and deploying applications in containers

6. What is the primary purpose of Azure DevOps?

Answer: Software development and delivery

7. Which components are part of Azure DevOps?

Answer: Azure Boards and Azure Pipelines

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

Answer: It tracks changes in source code and manages versions

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

Answer: Administrative tasks with superuser privileges

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

Answer: Azure Pipelines

SECTION : 2

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.

```
kbrkannan@KBRKANNAN:~$ pwd
/home/kbrkannan
kbrkannan@KBRKANNAN:~$ mkdir lab1
kbrkannan@KBRKANNAN:~$ cd lab1
kbrkannan@KBRKANNAN:~/lab1$ touch sample.txt
kbrkannan@KBRKANNAN:~/lab1$ ls
sample.txt
kbrkannan@KBRKANNAN:~/lab1$ cp sample.txt sample_copy.txt
kbrkannan@KBRKANNAN:~/lab1$ mv sample_copy.txt new_sample.txt
kbrkannan@KBRKANNAN:~/lab1$ ls
new_sample.txt  sample.txt
kbrkannan@KBRKANNAN:~/lab1$
```

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.

```
kbrkannan@KBRKANNAN:~$ mkdir lab2
kbrkannan@KBRKANNAN:~$ cd lab2
kbrkannan@KBRKANNAN:~/lab2$ touch secret.txt
kbrkannan@KBRKANNAN:~/lab2$ chmod u+rw secret.txt
kbrkannan@KBRKANNAN:~/lab2$ cat > secret.txt
New File Addedkbrkannan@KBRKANNAN:~/lab2$ sudo chown new_user secret.txt
kbrkannan@KBRKANNAN:~/lab2$ ls -l
total 4
-rw-r--r-- 1 new_user kbrkannan 14 Oct 20 11:11 secret.txt
kbrkannan@KBRKANNAN:~/lab2$ ls -n
total 4
-rw-r--r-- 1 1001 1000 14 Oct 20 11:11 secret.txt
kbrkannan@KBRKANNAN:~/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.

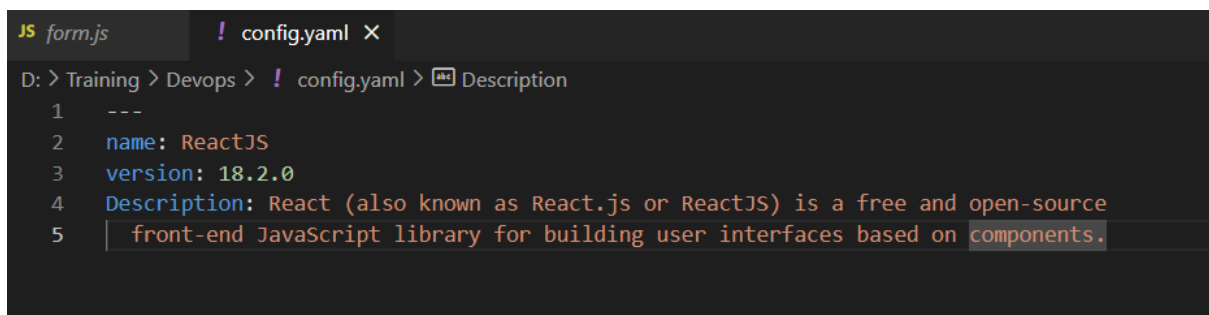
```
kbrkannan@KBRKANNAN:~$ mkdir lab3
kbrkannan@KBRKANNAN:~$ cd lab3
kbrkannan@KBRKANNAN:~/lab3$ touch newtextfile.txt
kbrkannan@KBRKANNAN:~/lab3$ cat > newtextfile.txt
This is the new file added in the new text file.kbrkannan@KBRKANNAN:~/lab3$ grep This newtextfile.txt
This is the new file added in the new text file.
kbrkannan@KBRKANNAN:~/lab3$ sed -i 's/added/replaced/g' newtextfile.txt
kbrkannan@KBRKANNAN:~/lab3$ cat newtextfile.txt
This is the new file replaced in the new text file.kbrkannan@KBRKANNAN:~/lab3$ wc newtextfile.txt
 0 11 51 newtextfile.txt
kbrkannan@KBRKANNAN:~/lab3$
```

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.



```
JS form.js ! config.yaml X
D: > Training > Devops > ! config.yaml > Description
1 ---
2 name: ReactJS
3 version: 18.2.0
4 Description: React (also known as React.js or ReactJS) is a free and open-source
5 front-end JavaScript library for building user interfaces based on components.
```

YAML Lint

Paste in your YAML and click "Go" - we'll tell you if it's valid or not, and give you a nice clean UTF-8 version of it.



```
1 ---
2 name: ReactJS
3 version: 18.2.0
4 Description: React (also known as React.js or ReactJS) is a free and open-source
5 front-end JavaScript library for building user interfaces based on components.
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Go ☒ Reformat (strips comments) ☒ Resolve aliases

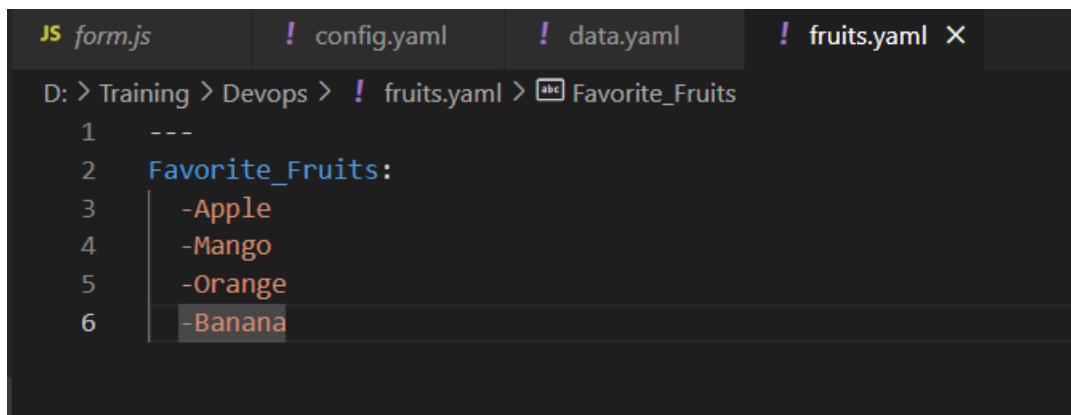
Valid YAML!

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.

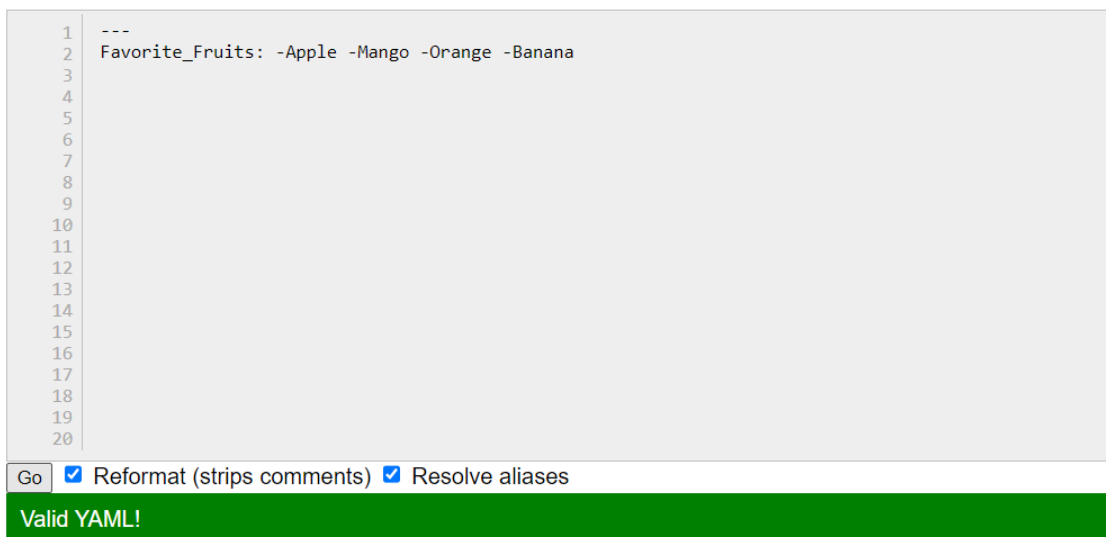


The screenshot shows a code editor with a dark theme. The file explorer on the left shows the path 'D: > Training > Devops > ! fruits.yaml'. The main editor area shows the following YAML content:

```
1 ---
2 Favorite_Fruits:
3   -Apple
4   -Mango
5   -Orange
6   -Banana
```

YAML Lint

Paste in your YAML and click "Go" - we'll tell you if it's valid or not, and give you a nice clean UTF-8 version of it.



The screenshot shows the YAML Lint web interface. The input field contains the following YAML content:

```
1 ---
2 Favorite_Fruits: -Apple -Mango -Orange -Banana
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

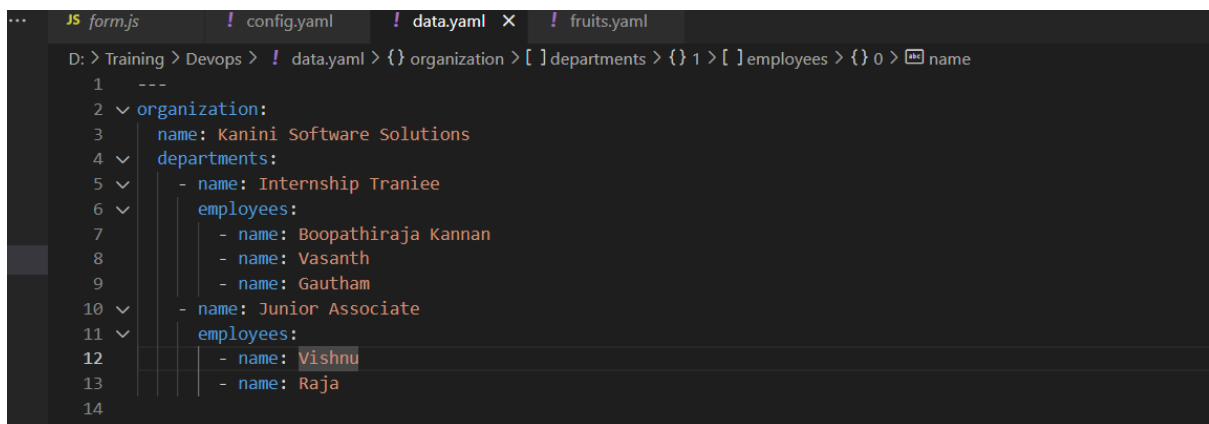
Below the input field, there are two checkboxes: "Reformat (strips comments)" and "Resolve aliases", both of which are checked. A "Go" button is located to the left of the checkboxes. Below the input field, a green banner displays the message "Valid YAML!".

Lab 6 : Nested Structures in yaml

□ Objective: Explore nested structures within YAML.

□ Task:

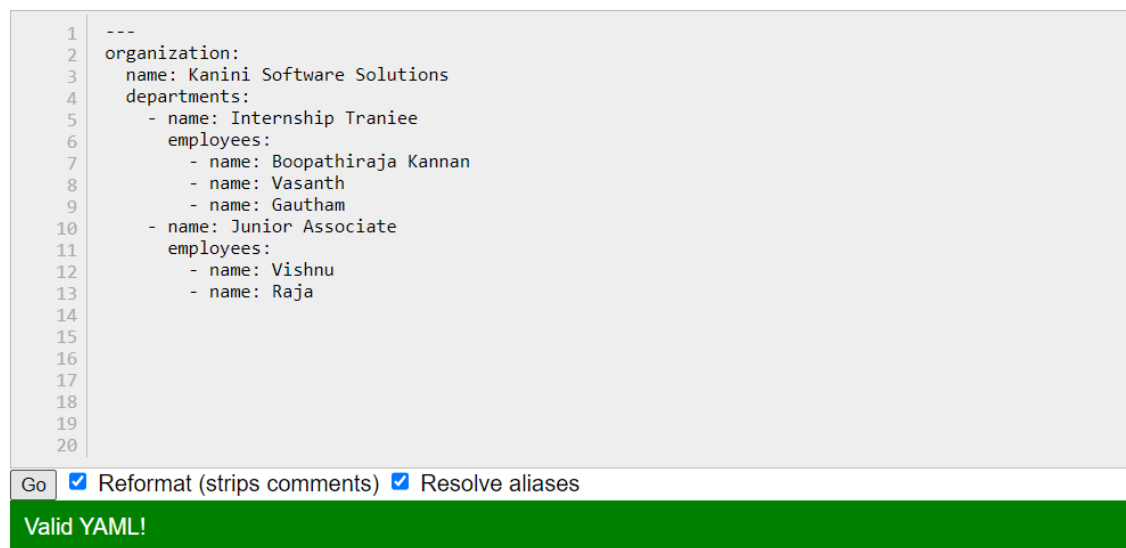
1. Create a YAML file named "data.yaml."
2. Define a nested structure representing a fictitious organization with departments and employees.
3. Use YAML syntax to add, update, or remove data within the nested structure.
4. Save and validate the YAML file.



```
1 ---
2 organization:
3   name: Kanini Software Solutions
4   departments:
5     - name: Internship Trainee
6       employees:
7         - name: Boopathiraja Kannan
8         - name: Vasanth
9         - name: Gautham
10    - name: Junior Associate
11      employees:
12        - name: Vishnu
13        - name: Raja
14
```

YAML Lint

Paste in your YAML and click "Go" - we'll tell you if it's valid or not, and give you a nice clean UTF-8 version of it.



```
1 ---
2 organization:
3   name: Kanini Software Solutions
4   departments:
5     - name: Internship Trainee
6       employees:
7         - name: Boopathiraja Kannan
8         - name: Vasanth
9         - name: Gautham
10    - name: Junior Associate
11      employees:
12        - name: Vishnu
13        - name: Raja
14
```

Go ☒ Reformat (strips comments) ☒ Resolve aliases

Valid 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.













Recently run pipelines

Pipeline	Last run
 Angularcicd-CI	#47 • Deleted azure-pipelines.yml Manually triggered for  master 5m ago 3m 12s

Angularcicd-CI

EditRun pipeline

RunsBranchesAnalytics

Description	Stages	
 #47 • Deleted azure-pipelines.yml Manually triggered for  master 59f81af5		5m ago 3m 12s
 #46 • Deleted azure-pipelines.yml Manually triggered for  master 59f81af5		9m ago 1m 30s
 #45 • Deleted azure-pipelines.yml Manually triggered for  master 59f81af5		12m ago 1m 50s
 #44 • Deleted azure-pipelines.yml Manually triggered for  master 59f81af5		24m ago 2m 18s


#47 • Deleted azure-pipelines.yml

Angularcicd-CI

Run new

This run is being retained as one of 3 recent runs by master (Branch).View retention leases

SummaryCode Coverage

Manually run by  Boopathiraja Kannan

Repository and version

Angularcicd
master 59f81af5

Time started and elapsed

Today at 5:13 PM
3m 12s


Related

0 work items
1 published; 1 consumed

Tests and coverage

Get started

Jobs

Name	Status	Duration
 Agent job 1	Success	3m 7s

← Jobs in run #47

Angularci-cd-CI

Jobs

✓ Agent job 1	3m 7s
Initialize job	1s
Checkout Angularci-cd...	6s
npm install	49s
npm build	27s
npm test	42s
Publish Pipeline Arti...	1m 0s
Post-job: Checkout An...	<1s
Finalize Job	<1s
Report build status	<1s

✓ Agent job 1

View raw log

```

1 Pool: Azure Pipelines
2 Image: windows-2019
3 Agent: Hosted Agent
4 Started: Fri at 5:14 PM
5 Duration: 3m 7s
6
7 ▶ Job preparation parameters
8 ▶ fx 1 queue time variable used
9 1 artifact produced

```

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.

Description	Stages	
<div>✓ #20231020.12 • Update azure-pipelines.yml for Azure Pipelines</div> <div>Manually triggered for master 6d1c330e</div>	✓	<div>15m ago</div> <div>7m 31s</div>
<div>✓ #20231020.9 • Update azure-pipelines.yml for Azure Pipelines</div> <div>Manually triggered for master 21d18f83</div>	✓	<div>21m ago</div> <div>3m 16s</div>
<div>✓ #20231020.1 • Set up CI with Azure Pipelines</div> <div>Individual CI for master 2d153aaf</div>	✓	<div>56m ago</div> <div>4m 37s</div>

#20231020.12 • Update azure-pipelines.yml for Azure Pipelines Run new

React

This run is being retained as one of 3 recent runs by master (Branch). View retention leases

Summary Code Coverage

Manually run by Boopathiraja Kannan View 2 changes

Repository and version React master 6d1c330e	Time started and elapsed Today at 4:11 PM 7m 31s	Related 0 work items 1 published	Tests and coverage Get started
---	---	---	--

Jobs

Name	Status	Duration
Job	Success	7m 23s

Jobs in run #20231020.12 React

Jobs

Name	Status	Duration
Job	Success	7m 23s
Initialize job	Success	<1s
Checkout React@maste...	Success	3s
Install Node.js	Success	<1s
npm install and bu...	Success	1m 26s
npm test	Success	24s
PublishPipelineArti...	Success	5m 27s
Post-job: Checkout Re...	Success	<1s
Finalize Job	Success	<1s
Report build status	Success	<1s

Finalize Job View raw log

```

1 Starting: Finalize Job
2 Cleaning up task key
3 Start cleaning up orphan processes.
4 Finishing: Finalize Job
  
```

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.

Pipelines New pipeline

Recent All Runs Filter pipelines

Recently run pipelines

Pipeline	Last run
PipelineforDotnet-ASP.NET Core-CI	#20231022.1 • Commit Manually triggered for master Just now 29s

#20231022.1 • Commit Run new

PipelineforDotnet-ASP.NET Core-CI

This run is being retained as one of 3 recent runs by master (Branch). View retention leases

Summary Code Coverage

Manually run by Boopathiraja Kannan View change

Repository and version PipelineforDotnet master 5c8b2491	Time started and elapsed Today at 11:02 PM 29s	Related 0 work items 1 published; 1 consumed	Tests and coverage Get started
---	--	--	---

← **Jobs in run #20231022.2** PipelineforDotnet-ASP.NET Core-CI

Jobs

Agent job 1 28s
Initialize job <1s
Checkout PipelineforDo... 2s
Restore 6s
Build 6s
Test <1s
Build solution **/*.sln 6s
Publish 3s
Publish Artifact 1s
Post-job: Checkout Pi... <1s
Finalize Job <1s
Report build status <1s

Agent job 1 View raw log

```

1 Pool: Default
2 Queued: Just now [manage_parallel_jobs]
3 Agent: KBRKANNAN
4 Started: Just now
5 Duration: 28s
6
7 The agent request is already running or has already completed.
8 ▶ Job preparation parameters
9 ▶ fx 3 queue time variables used
10  1 artifact produced
11 Job live console data:
12 Starting: Agent job 1
13 Async Command Start: DetectDockerContainer
14 Async Command End: DetectDockerContainer
15 Async Command Start: DetectDockerContainer
16 Async Command End: DetectDockerContainer
17 Finishing: Agent job 1

```

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

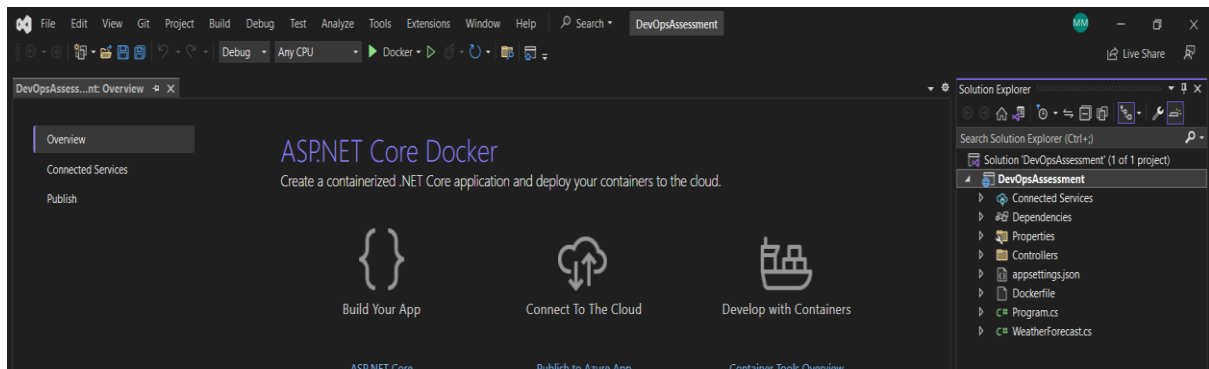
Step 2: Build the .NET Core Web API Project

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

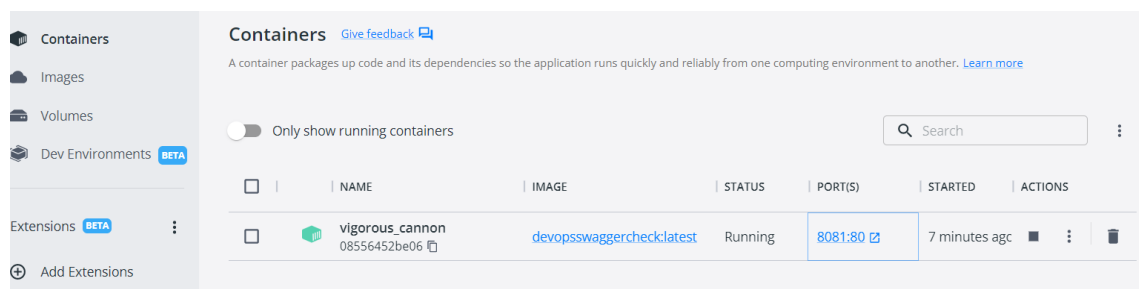
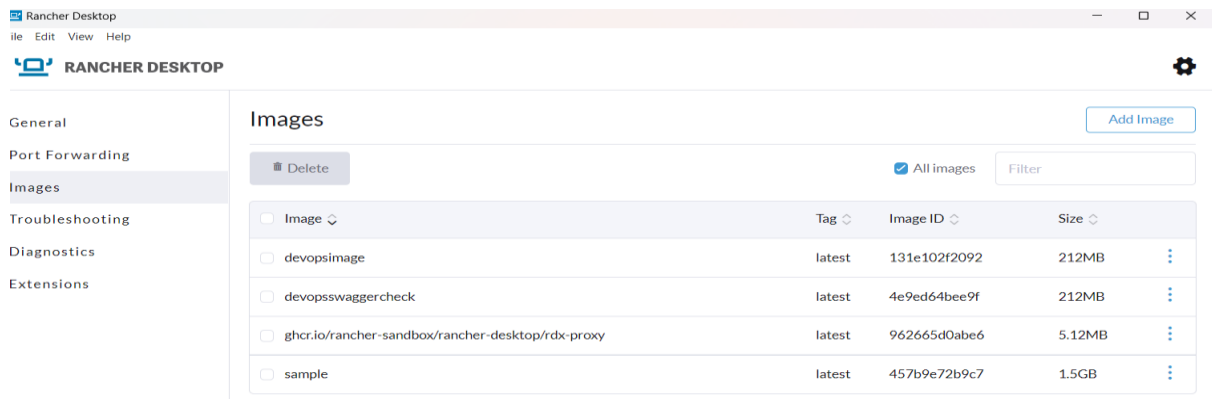


```
Microsoft Windows [Version 10.0.22635.2483]
(c) Microsoft Corporation. All rights reserved.

C:\Manini Software Solutions\ManiniWorkspacePortal\DevOpsAssessment>docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
devopsswagger       latest          131e102f2892   2 days ago     212MB
devopsswaggercheck  latest          4e9ed64bee9f   2 days ago     212MB
sample              latest          457b9e72b9c7   3 weeks ago    1.5GB
ghcr.io/rancher-sandbox/rancher-desktop/rdx-proxy         latest          962665d0abe6   N/A             5.12MB

C:\Manini Software Solutions\ManiniWorkspacePortal\DevOpsAssessment>docker build -t devopsswagger .
[*] Building 17.9s (14/14) FINISHED                                docker:default
=> [internal] load build definition from dockerfile                0.0s
=> => transferring dockerfile: 32B                                0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                      0.0s
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:6.0 0.7s
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:6.0   0.3s
=> [internal] load build context                                  0.1s
=> => transferring context: 5.03MB                                  0.1s
=> [build 1/5] FROM mcr.microsoft.com/dotnet/sdk:6.0@sha256:1167d21bf2d6bbaeb9fca863a73e90defa9233574b74d9deabee 0.0s
=> [stage-1 1/3] FROM mcr.microsoft.com/dotnet/aspnet:6.0@sha256:d4ff6177eb040828c0e2d9d6a4ba813841826e412e5b36c 0.0s
=> => resolve mcr.microsoft.com/dotnet/aspnet:6.0@sha256:d4ff6177eb040828c0e2d9d6a4ba813841826e412e5b36c1a82fc93 0.0s
=> [build 2/5] WORKDIR /source                                    0.0s
=> [build 3/5] COPY . .                                          0.1s
=> [build 4/5] RUN dotnet restore ".\DevOpsAssessment\DevOpsAssessment.csproj" --disable-parallel 12.7s
=> [build 5/5] RUN dotnet publish ".\DevOpsAssessment\DevOpsAssessment.csproj" -c release -o /app --no-restore 3.7s
=> [stage-1 2/3] WORKDIR /app                                    0.0s
=> [stage-1 3/3] COPY --from=build /app ./                        0.0s
=> exporting to image                                             0.0s
=> => exporting layers                                             0.0s
=> writing image sha256:4e9ed64bee9fc1b1e862305148d911a3aaf9e0e4f971415989d1d7b455e03b97 0.0s
=> naming to docker.io/library/devopsswagger                    0.0s

C:\Manini Software Solutions\ManiniWorkspacePortal\DevOpsAssessment>
```



```
C:\Kanini Software Solutions\KaniniWorkspacePortal\DevOpsAssessment>docker run -p 8081:80 devopsswaggercheck:latest
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:80
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /app/
warn: Microsoft.AspNetCore.HttpPolicy.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```

