Academic Year: 2025-26 **Semester: V**
Class/ Div: TE IT
Subject: DevOps Lab (DL)
Subject In-charge: Ms. Sujata Oak

Assignment-2

Lab Outcomes (LO):

ITL503.1: To explain fundamentals of DevOps practices which aims to simplify Softare Development Life Cycle

ITL503.2: Make use of various Git related operations to obtain complete knowledge of the "version control system" to effectively track changes augmented with Git and GitHub

ITL503.3: To build and deploy Software Applications on server environment using Jenkins

ITL503.4: To build and test Software Applications using Selenium

ITL503.5: Analyze the Containerization of OS images and deployment of applications over Docker

ITL503.6: Make use of Ansible tool to implement software configuration management

---

***BL: Blooms Level *LO: Lab Outcomes**

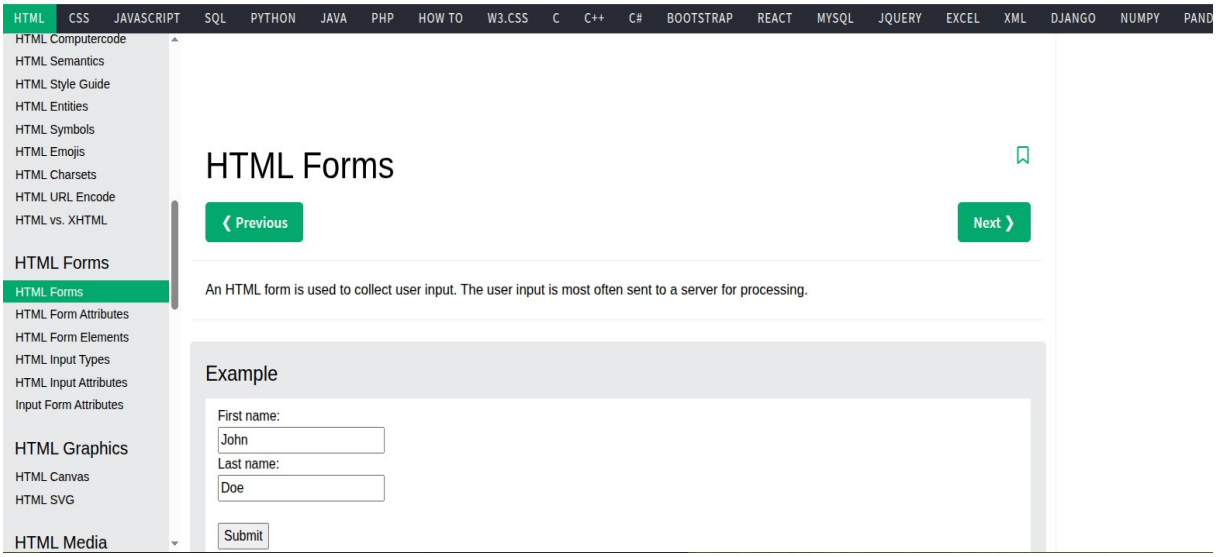
Q.1) Automate Form Filling Using Selenium and Python. **CO:4 BL:2**

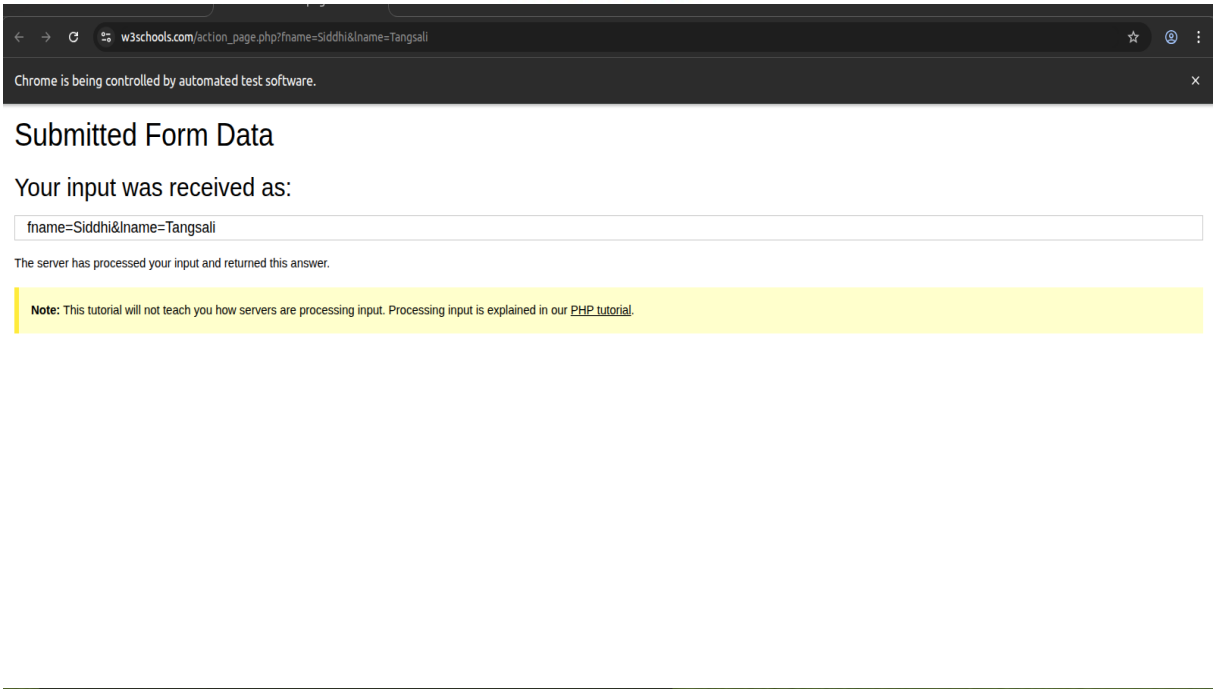
Solution:

```
# form_fill.py
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

# Initialize Chrome driver
driver = webdriver.Chrome()

# Step 1: Open form page
driver.get("https://www.w3schools.com/html/html_forms.asp")
driver.maximize_window()

# Step 2: Find input fields by name
fname = driver.find_element(By.ID, "fname")
```

```
lname = driver.find_element(By.ID, "lname")

# Step 3: Clear and fill the form fields
fname.clear()
fname.send_keys("Sujata")
lname.clear()
lname.send_keys("Oak")

# Step 4: Scroll and click Submit button
driver.find_element(By.XPATH, "//input[@type='submit']").click()

# Step 5: Wait and print the new URL
time.sleep(5)
print("Form submitted successfully!")
print("Redirected URL:", driver.current_url)

# Step 6: Close browser
driver.quit()
```

OUTPUT:



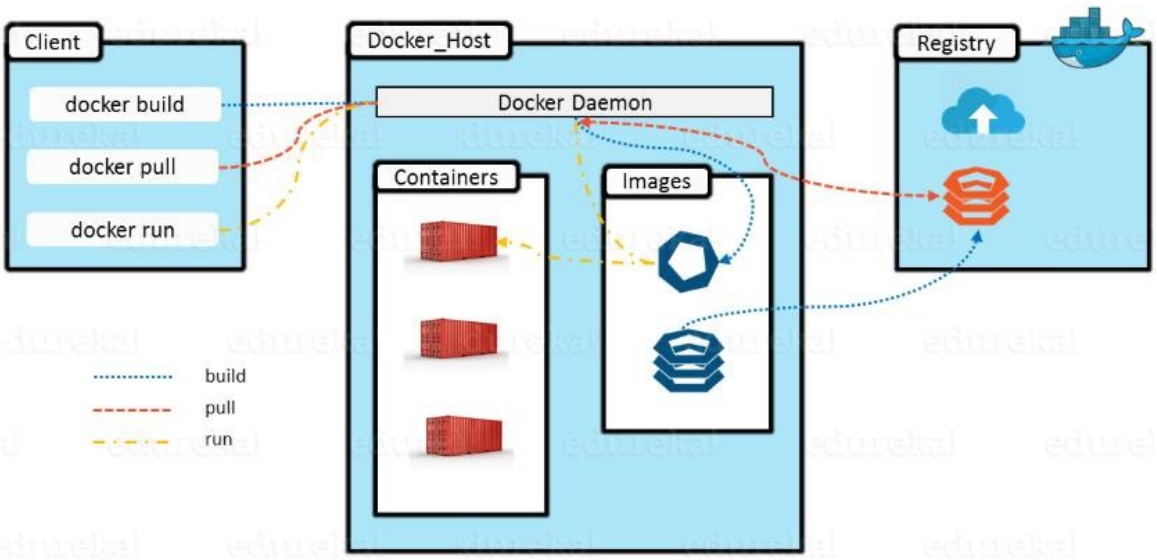


Form submitted successfully!

Redirected URL: https://www.w3schools.com/html/html_forms.asp

Q.2) Explain each component of Docker Architecture? CO:5 BL:2

Solution:



1. Docker Client

- The Docker Client is the interface through which users interact with Docker.
- Commands like docker build, docker run, and docker pull are sent from the client to the Docker Daemon using REST APIs.

2. Docker Daemon (dockerd)

- The Docker Daemon runs in the background on the host system.
- It listens to Docker API requests and manages images, containers, networks, and volumes.
- It does the actual work of building, running, and distributing containers.

3. Docker Images

- A Docker Image is a read-only template used to create containers.
- It contains the application code, libraries, and dependencies required to run the app.
- Images are built from a Dockerfile.

4. Docker Containers

- A Container is a running instance of an image.
- It is lightweight, portable, and isolated from other containers but shares the same OS kernel.
- You can start, stop, move, or delete containers easily.

5. Docker Registry (e.g., Docker Hub)

- A Registry stores and distributes Docker images.
- Docker Hub is the default public registry; you can also have private registries.
- Commands like docker pull and docker push interact with registries.

6. Docker Engine

- The Docker Engine combines the Client, Daemon, and REST API.
- It is the core part of Docker that enables building and running containers.

Q.3) Demonstrate how to deploy "Hello World" java application using a Dockerfile?

CO:5 BL:3

Solution:

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-java# ls
Dockerfile  Hello.java
```

Hello.java

```
class Hello{
public static void main(String[] args){
System.out.println("Hello Evryone!!!");
System.out.println("I am a java app running inside a docker container");
}
}
```

nano Dockerfile:

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-java# nano Dockerfile
```

```
FROM openjdk
LABEL author="Prof. Sujata Oak"
COPY . /var/www/java
WORKDIR /var/www/java
RUN javac Hello.java
CMD ["java", "Hello"]
```

 # docker build -t sujatadocker2025/javaapp .

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~/Desktop/New Folder 3$  docker build -t sujatadocker2025/javaapp .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  4.608kB
Step 1/6 : FROM openjdk
latest: Pulling from library/openjdk
197c1adcd755: Pull complete
57b698b7af4b: Pull complete
95a27dbe0150: Pull complete
Digest: sha256:9b448de897d211c9e0ec635a485650aed6e28d4eca1efbc34940560a480b3f1f
Status: Downloaded newer image for openjdk:latest
 ---> 71260f256d19
Step 2/6 : LABEL author="Siddhi Tangsali"
 ---> Running in b83234b04c64
 ---> Removed intermediate container b83234b04c64
 ---> 119b8a9bb7bd
Step 3/6 : COPY . /var/www/java
```

#docker images

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~/Desktop/New Folder 3$ docker images
REPOSITORY                                TAG       IMAGE ID       CREATED          SIZE
<none>                                    <none>    6259473545f3   24 seconds ago   470MB
sonarqube                                 latest    c23ddf1f0583   5 weeks ago      1.22GB
2a5818839b03/apsitsiddhicontainer25       v1        2a5818839b03   2 months ago     136MB
siddheee/apsitsiddhicontainer25           v1        2a5818839b03   2 months ago     136MB
siddhidocker2025/website25                latest    2a5818839b03   2 months ago     136MB
itsSiddheee/apsitsiddhicontainer25        v1        2a5818839b03   2 months ago     136MB
```

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~/Desktop/New Folder 3$ docker ps -a
CONTAINER ID   IMAGE                 COMMAND                  CREATED          STATUS                      PORTS
               NAMES
b2193f767e24   ada                   "tail -f /dev/null"      34 seconds ago   Up 32 seconds
               siddhi_javacontainer1
a919c4b5d1c8   ada                   "java Hello"             2 minutes ago    Exited (0) 2 minutes ago
               sujata_javacontainer
15a95fa9e7d4   6259473545f3          "/bin/sh -c 'javac H…"   4 minutes ago    Exited (2) 4 minutes ago
```

 # docker run -d --name sujata_javacontainer 03f tail -f /dev/null

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~/Desktop/New Folder 3$ docker run -d --name sujata_javacontainer ada tail -f /dev/null
docker: Error response from daemon: Conflict. The container name "/sujata_javacontainer" is already in use by container "a919c4b5d1c813de6b72fa1b97f96238fcf5c651cf3a352737a2933
ec4807d94". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
```

 # docker run -d --name sujata_javacontainer1 03f tail -f /dev/null

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~/Desktop/New Folder 3$ docker run -d --name siddhi_javacon
ainer1 ada tail -f /dev/null
b2193f767e24da5d40f9593ae8c2c2f0cc87b476a0d203fa8281e4eb8dd52fee
```

 # docker ps -a

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~/Desktop/New Folder 3$ docker ps -a
CONTAINER ID   IMAGE                COMMAND                  CREATED          STATUS
          PORTS
   NAMES
b2193f767e24   ada                  "tail -f /dev/null"      34 seconds ago   Up 32 seconds

   siddhi_javacontainer1
a919c4b5d1c8   ada                  "java Hello"             2 minutes ago    Exited (0) 2 minut
es ago
   sujata_javacontainer
```

root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-java# **docker exec -it 6ec bash**

bash-4.4# javac Hello.java

bash-4.4# java Hello

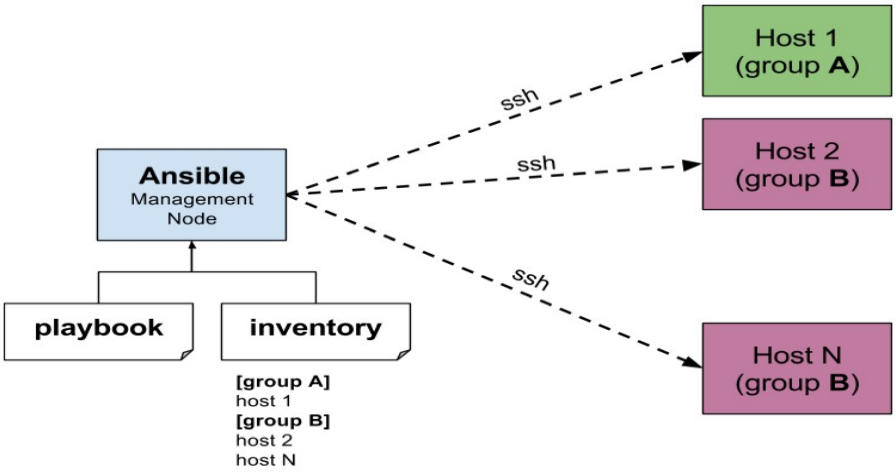
```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~/Desktop/New Folder 3$ docker exec -it b21 bash
bash-4.4# javac Hello.java
bash-4.4# java Hello
Hello Evryone!!!
I am a java app running inside a docker container
bash-4.4#
```

Q.4) Explain Architecture of ansible with its core components? CO:6 BL:3

Solution:

Ansible follows an agentless architecture, meaning no software/agent needs to be installed on target machines. It uses SSH (Linux) or WinRM (Windows) to communicate and execute tasks remotely.

Core Components of Ansible:



Component	Description
Control Node	The main machine where Ansible is installed and from which playbooks are executed.
Managed Nodes	Target machines that Ansible manages (no agent required).
Inventory	A file (usually hosts) that lists all the managed nodes (IP or hostname).
Modules	Reusable units of code (like copy, yum, service) that perform specific tasks on nodes.
Plugins	Extend Ansible's functionality (e.g., connection plugins, callback plugins).
Playbooks	YAML files that define automation tasks in order (what to configure or deploy).
Tasks	Individual actions inside a playbook, executed by modules.
Facts	System information collected from managed nodes (like OS, IP, memory).

Component	Description
Roles	A structured way to organize playbooks into reusable components (tasks, vars, templates, etc.).

Q.5) Deploy a Web Application Using Ansible playbook? **CO:6 BL:3**

Index.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Welcome to My Ansible Deployed Site</title>
</head>

<body style="text-align:center; font-family:sans-serif;">

  <h1> Deployment Successful!</h1>

  <p>This web page was deployed using Ansible Playbook by Prof Sujata Oak.</p>

</body>

</html>
```

EXPLANATION:

Q1] form_fill.py

Your script opens a page, fills two text fields, clicks submit, waits a bit and quits.

```
from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

import time
```

- webdriver: main API to control a browser (Chrome, Firefox, etc.).
- By: locator strategy helper (By.ID, By.NAME, By.XPATH, etc.).

- Keys: for keyboard keys (Enter, TAB).
- time: used for time.sleep()

driver = webdriver.Chrome()

- Starts a new Chrome browser controlled by Selenium.
- How driver is found: modern Selenium uses Selenium Manager to locate/download the right ChromeDriver automatically.

Open form page:

driver.get("https://www.w3schools.com/html/html_forms.asp")

driver.maximize_window()

- get(url) navigates to the URL and waits for the page to load (Selenium waits for the load event by default).
- maximize_window() makes the window fullscreen

Find input fields by ID

fname = driver.find_element(By.ID, "fname")

lname = driver.find_element(By.ID, "lname")

Clear and fill the fields

fname.clear()

fname.send_keys("Sujata")

lname.clear()

lname.send_keys("Oak")

Scroll and click Submit button

driver.find_element(By.XPATH, "//input[@type='submit']").click()

Wait and print the new URL

time.sleep(5)

print("Form submitted successfully!")

```
print("Redirected URL:", driver.current_url)
```

Close browser

```
driver.quit()
```