2. Install Docker, explore its containerization commands, create Docker container using various operating system images and deploy containerized applications using Docker and Docker hub

Step 1: Install Docker on Windows

1. Download Docker Desktop for Windows from the official Docker website.
2. Run the installer and follow the on-screen instructions.
3. Once installed, open Docker Desktop and ensure it is running correctly.

Step 2: Explore Docker commands:

Check Docker version:

docker –version

Get system-wide Docker information:

docker info

List running containers:

docker ps

List all containers (including stopped):

docker ps –a

List Docker images:

docker images

Pull an image from Docker Hub:

docker pull Ubuntu

Run a Docker container interactively:

docker run -it Ubuntu

Stop a running container:

docker stop <container_id>

Remove a container:

docker rm <container_id>

Remove a Docker image:

docker rmi <image_id>

Build a Docker image from a Dockerfile:

docker build -t <image_name> **.**

Step3: Create Docker Containers Using Various Operating System Images

Use Ubuntu Image

1. **Pull the latest Ubuntu image:**

   docker pull Ubuntu

2. **Run a container using Ubuntu image**

   docker run -it Ubuntu

3. **Exit the container:**

   exit

Use Alpine Image (Lightweight OS)

docker pull alpine

docker run -it alpine

exit

Run Containers with Other Operating Systems

docker pull centos

docker pull debian

docker pull fedora

Step4: Deploy Containerized Applications Using Docker

Deploying a Simple Web Application

For example, let's deploy a Python-based web application.

1. **Create a Python web app (app.py):** Create a file named app.py with the following content:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, Docker!'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

2. **Create a Dockerfile:** Create a Dockerfile to containerize this Python application:

```
# Use the official Python image from Docker Hub
FROM python:3.8-slim

# Set the working directory inside the container
WORKDIR /app

# Copy the local app file into the container
COPY app.py .

# Install Flask
RUN pip install flask

# Expose port 5000 for the web app
EXPOSE 5000

# Command to run the app
CMD ["python", "app.py"]
```

3. **Build the Docker image:**

   docker build -t python-flask-app .

4. **Run the Docker container:**

```
docker run -p 5000:5000 python-flask-app
```

5. **Access the application:** Open your browser and go to http://localhost:5000. You should see "Hello, Docker!" displayed.

Step 5: Push Docker Image to Docker Hub

1. **Create a Docker Hub Account:** Go to Docker Hub and sign up for an account.
2. **Login to Docker Hub**
   Use the following command to log in to Docker Hub:
   ```
   docker login
   ```
   Enter your Docker Hub credentials.
3. **Tag and Push the Image**

   Tag the image you want to upload:

   docker tag python-flask-app <your_dockerhub_username>/python-flask-app

   Push the image to Docker Hub:

   docker push <your_dockerhub_username>/python-flask-app

4. Pull Image from Docker Hub

   docker pull <your_dockerhub_username>/python-flask-app