

Here is a **step-by-step guide** to build, run, push, and pull a weather app using your Docker Hub :

1. Install Docker on Your CentOS System

If Docker is not already installed, install it:

```
sudo yum install -y docker
sudo systemctl start docker
sudo systemctl enable docker
```

Verify the Docker installation:

```
docker --version
```

2. Log in to Docker Hub

Log in with your Docker Hub username:

```
docker login
```

Enter your username (faisal991) and password when prompted. If you have two-factor authentication enabled, use a personal access token instead of your password.

3. Create the Weather App

1. Create a project directory:

```
mkdir weather-app
cd weather-app
```

2. Create the Python script: Create a file named `app.py`:

`nano app.py`

Add the following code:

```
from flask import Flask, request, jsonify
import requests

app = Flask(__name__)

API_KEY = "your_openweathermap_api_key" # Replace with your
OpenWeatherMap API key

@app.route('/weather', methods=['GET'])
def get_weather():
    city = request.args.get('city')
    if not city:
        return jsonify({"error": "Please provide a city name"}), 400

    url =
f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_K
EY}&units=metric"
    response = requests.get(url)

    if response.status_code != 200:
        return jsonify({"error": "City not found"}), 404

    data = response.json()
    return jsonify({
        "city": city,
        "temperature": data['main']['temp'],
        "description": data['weather'][0]['description']
    })

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

4. Create the Dockerfile

The Dockerfile specifies how to build your app's Docker image:

```
nano Dockerfile
```

Add the following content:

```
# Use the official Python image
FROM python:3.9-slim

# Set the working directory in the container
WORKDIR /app

# Copy the application code into the container
COPY app.py /app

# Install required Python packages
RUN pip install flask requests

# Expose port 5000
EXPOSE 5000

# Run the application
CMD ["python", "app.py"]
```

5. Build the Docker Image

Build the Docker image and tag it with your Docker Hub username:

```
docker build -t faisal991/weather-app .
```

Verify the image:

```
docker images
```

You should see an image named `faisal991/weather-app`.

6. Run the Docker Container

Run the container locally to test the app:

```
docker run -d -p 5000:5000 faisal991/weather-app
```

Check the running containers:

```
docker ps
```

Test the app by visiting the URL or using `curl`:

```
curl "http://localhost:5000/weather?city=London"
```

Expected response:

```
{
  "city": "London",
  "temperature": 8.5,
  "description": "clear sky"
}
```

7. Push the Image to Docker Hub

1. Tag the image for Docker Hub:

```
docker tag faisal991/weather-app faisal991/weather-app:latest
```

2. Push the image to Docker Hub:

```
docker push faisal991/weather-app:latest
```

Verify it is available on Docker Hub by visiting your repository at:

<https://hub.docker.com/repository/docker/faisal991/weather-app>

8. Pull the Image on Any System

On another system, you can pull and run the app:

1. Pull the image:

```
docker pull faisal991/weather-app:latest
```

2. Run the container:

```
docker run -d -p 5000:5000 faisal991/weather-app:latest
```

3. Test the app:

```
curl "http://localhost:5000/weather?city=London"
```

9. Stop and Remove Containers (Optional)

To stop and clean up running containers:

```
docker ps # Find the container ID
docker stop <container_id>
docker rm <container_id>
```

To remove the image:

```
docker rmi faisal991/weather-app:latest
```

10. Automate with Docker Compose (Optional)

If you want to use Docker Compose:

1. Create a `docker-compose.yml` file:

```
nano docker-compose.yml
```

Add the following:

```
version: '3.8'
```

```
services:
```

```
  weather-app:
```

```
    image: faisal991/weather-app:latest
```

```
    ports:
```

```
      - "5000:5000"
```

2. Start the service:

```
docker-compose up -d
```