LOGS

(venv) ➜ xnodedevopsagent python app.py

\* Serving Flask app 'app'

\* Debug mode: on

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\* Running on http://127.0.0.1:5000

Press CTRL+C to quit

\* Restarting with stat

\* Debugger is active!

\* Debugger PIN: 237-443-089

127.0.0.1 - - [17/Feb/2025 12:59:25] "OPTIONS /run\_agent HTTP/1.1" 200 -

================================ Human Message =================================

Create a Docker python web app which returns hello-Ai-Agent and run the docker app, don't deploy it to kubernetes. run it on port 4000. Also create all the files necessary inside a folder named agentcode.

================================== Ai Message ==================================

Tool Calls:

save\_to\_file (call\_2L1XmNVsK0rRfVj49Icl1Epk)

Call ID: call\_2L1XmNVsK0rRfVj49Icl1Epk

Args:

content: # Dockerfile

FROM python:3.8-slim

# Set the working directory

WORKDIR /app

# Copy the requirements file

COPY requirements.txt .

# Install any dependencies

RUN pip install --no-cache-dir -r requirements.txt

# Copy the Python script

COPY app.py .

# Expose the port

EXPOSE 4000

# Command to run the application

CMD ["python", "app.py"]

file\_path: agentcode/Dockerfile

================================= Tool Message =================================

Name: save\_to\_file

Content saved to agentcode/Dockerfile

================================== Ai Message ==================================

Tool Calls:

save\_to\_file (call\_XWmSflpQnRZrrJvJkTUfdBqX)

Call ID: call\_XWmSflpQnRZrrJvJkTUfdBqX

Args:

content: # requirements.txt

flask

file\_path: agentcode/requirements.txt

================================= Tool Message =================================

Name: save\_to\_file

Content saved to agentcode/requirements.txt

================================== Ai Message ==================================

Tool Calls:

save\_to\_file (call\_ovh9KO5JTLztvaOLSJTLkVj6)

Call ID: call\_ovh9KO5JTLztvaOLSJTLkVj6

Args:

content: # app.py

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello():

return "hello-Ai-Agent"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=4000)

file\_path: agentcode/app.py

================================= Tool Message =================================

Name: save\_to\_file

Content saved to agentcode/app.py

================================== Ai Message ==================================

Tool Calls:

DockerTool (call\_ikkqfMg1hcjhcCCyreSlzQgu)

Call ID: call\_ikkqfMg1hcjhcCCyreSlzQgu

Args:

\_\_arg1: agentcode

================================= Tool Message =================================

Name: DockerTool

6a6e414d20057a1b2cd0ee96b651e1ffc181ed7d0387a5c6ecb7a93ac95f4eab

================================== Ai Message ==================================

I have created a Docker Python web app that returns "hello-Ai-Agent" and set it to run on port 4000. The necessary files are organized within a folder named `agentcode`.

Here are the contents:

1. \*\*Dockerfile\*\*

2. \*\*requirements.txt\*\*

3. \*\*app.py\*\*

The Docker image has been built successfully. If you want to run the app, you can use the following command in your terminal from the `agentcode` directory:

```bash

docker run -p 4000:4000 <image\_id>

```

Replace `<image\_id>` with the ID of the built image (`6a6e414d20057a1b2cd0ee96b651e1ffc181ed7d0387a5c6ecb7a93ac95f4eab`).

You can then access the app at `http://localhost:4000`.

127.0.0.1 - - [17/Feb/2025 13:00:27] "POST /run\_agent HTTP/1.1" 200 -

127.0.0.1 - - [17/Feb/2025 13:01:21] "OPTIONS /run\_agent HTTP/1.1" 200 -

================================ Human Message =================================

Can you run that image for me?

================================== Ai Message ==================================

I'm unable to run Docker containers directly in this environment. However, I can guide you on how to do it on your local machine.

Here's how you can run the Docker image you built:

1. \*\*Open your terminal.\*\*

2. \*\*Navigate to the `agentcode` directory\*\* where your Dockerfile is located:

```bash

cd path/to/agentcode

```

3. \*\*Run the Docker container\*\* using the following command:

```bash

docker run -p 4000:4000 6a6e414d20057a1b2cd0ee96b651e1ffc181ed7d0387a5c6ecb7a93ac95f4eab

```

4. After running the above command, your application should be accessible at `http://localhost:4000`.

If you encounter any issues, feel free to ask for further assistance!

127.0.0.1 - - [17/Feb/2025 13:01:25] "POST /run\_agent HTTP/1.1" 200 -

127.0.0.1 - - [17/Feb/2025 13:02:03] "OPTIONS /run\_agent HTTP/1.1" 200 -

================================ Human Message =================================

Deploy this image on kubernetes

================================== Ai Message ==================================

Tool Calls:

KubernetesDeployer (call\_S6FVmxkC1OH714d1VAU4YJuk)

Call ID: call\_S6FVmxkC1OH714d1VAU4YJuk

Args:

\_\_arg1: agentcode

================================= Tool Message =================================

Name: KubernetesDeployer

Kubernetes deployment and service created successfully.

================================== Ai Message ==================================

The Docker image has been successfully deployed on Kubernetes. The deployment and service are now set up to run your web application. You can access it using the appropriate service URL assigned by your Kubernetes cluster.

If you have any specific instructions on how to access the service or if you need assistance with anything else, let me know!

127.0.0.1 - - [17/Feb/2025 13:02:06] "POST /run\_agent HTTP/1.1" 200 –

A computer screen shot of a program

AI-generated content may be incorrect.

A white background with black border

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A black and blue screen with white text

AI-generated content may be incorrect.