Create two applications in two different docker containers. Push those

applications and run to show the communications between two Dockers. (Hint IPC)

ChatGPT

Inter-process communication (IPC) between two Docker containers can be achieved using various methods. One common approach is to use Docker networking to enable communication between containers. Below is a step-by-step guide to creating two applications in separate Docker containers, pushing the images to a container registry (e.g., Docker Hub), and then running them to demonstrate communication:

Application 1 - Docker Container 1:

Create a Simple Flask Application (app1.py):

python

from flask import Flask

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

@app.route('/')

def hello():

return 'Hello from Container 1!'

Create a Dockerfile for Container 1 (Dockerfile1):

Dockerfile

FROM python:3.8-slim

WORKDIR /app

COPY app1.py /app/

EXPOSE 5000

CMD ["python", "app1.py"]

Build and Push Docker Image for Container 1:

docker build -t your-username/app1 -f Dockerfile1 .

docker push your-username/app1

Application 2 - Docker Container 2:

Create Another Flask Application (app2.py):

from flask import Flask

import requests

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

@app.route('/')

def hello():

response = requests.get('http://container1:5000')

return f'Hello from Container 2! Received response from Container 1: {response.text}'

Note the URL http://container1:5000 in the requests.get call. This is the hostname of Container 1, which will be used for communication.

Create a Dockerfile for Container 2 (Dockerfile2):

FROM python:3.8-slim

WORKDIR /app

COPY app2.py /app/

RUN pip install Flask requests

EXPOSE 5000

CMD ["python", "app2.py"]

Build and Push Docker Image for Container 2:

docker build -t your-username/app2 -f Dockerfile2 .

docker push your-username/app2

Running Containers:

Run Container 1:

bash

docker run -d --name container1 your-username/app1

Run Container 2:

docker run -d --name container2 --link container1 your-username/app2

The --link flag allows Container 2 to communicate with Container 1.

Check Communication:

Visit http://localhost:5000 in your browser or use curl to see the communication between the two containers:

curl http://localhost:5000

You should see a response indicating communication between Container 1 and Container 2.

This example uses Flask for simplicity, but you can adapt the concept to other types of applications and communication methods based on your specific requirements.