# Disclaimer: This output contains AI-generated content; user is advised to review it before consumption.  
  
#\*Start of AI Generated Content\*

# **Docker Deployment for Netflix System Design API**

## **Step-by-Step Guide for Containerized Deployment**

### **Prerequisites**

* Docker Engine (latest version)
* Docker Compose (latest version)
* Python 3.9+ (for building the API image)
* Dependencies:
* Kafka
* Apache Chukwa
* Elasticsearch
* Memcached
* Hystrix

### **Step 1: Create Docker Images for Services**

#### **1.1 Netflix System Design API Image**

\* \*\*Directory:\*\* `/path/to/netflix-system-design-api`  
\* \*\*Dockerfile:\*\*  
```dockerfile  
FROM python:3.9-slim  
  
# Set working directory  
WORKDIR /app  
  
# Copy requirements file  
COPY requirements.txt .  
  
# Install dependencies  
RUN pip install -r requirements.txt  
  
# Copy application code  
COPY . .  
  
# Expose port  
EXPOSE 8000  
  
# Command to run the API  
CMD ["python", "netflix\_system\_design\_api.py"]

* **Build Image:**

docker build -t netflix-api:image .

#### **1.2 Kafka Image**

\* \*\*Use existing image:\*\* `confluentinc/cp-kafka:latest`  
\* \*\*No build required\*\*  
#### \*\*1.3 Apache Chukwa Image\*\*  
```markdown  
\* \*\*Use existing image:\*\* `apache/chukwa:latest`  
\* \*\*No build required\*\*  
#### \*\*1.4 Elasticsearch Image\*\*  
```markdown  
\* \*\*Use existing image:\*\* `elasticsearch:latest`  
\* \*\*No build required\*\*  
#### \*\*1.5 Memcached Image\*\*  
```markdown  
\* \*\*Use existing image:\*\* `memcached:latest`  
\* \*\*No build required\*\*  
#### \*\*1.6 Hystrix Image\*\*  
```markdown  
\* \*\*Note:\*\* Hystrix is a Java-based library. For simplicity, we'll use a pre-built image.  
\* \*\*Use existing image:\*\* `hystrix/hystrix-dashboard:latest`  
\* \*\*No build required\*\*  
  
### \*\*Step 2: Configure Docker Compose\*\*  
-----------------------------  
  
\* \*\*File:\*\* `docker-compose.yml`  
\* \*\*Content:\*\*  
```yml  
version: '3'  
  
services:  
 netflix-api:  
 image: netflix-api:image  
 container\_name: netflix-api  
 depends\_on:  
 - kafka  
 - chukwa  
 - elasticsearch  
 - memcached  
 - hystrix  
 ports:  
 - "8000:8000"  
 environment:  
 - KAFKA\_BOOTSTRAP\_SERVERS=localhost:9092  
 - CHUKWA\_SERVICE\_URL=http://chukwa:8080  
 - ELASTICSEARCH\_URL=http://elasticsearch:9200  
 - EV\_CACHE\_MEMCACHED\_SERVERS=memcached:11211  
 - HYSTRIX\_COMMAND\_KEY=netflix-system-design  
  
 kafka:  
 image: confluentinc/cp-kafka:latest  
 container\_name: kafka  
 ports:  
 - "9092:9092"  
 environment:  
 - KAFKA\_BROKER\_ID=1  
 - KAFKA\_ZOOKEEPER\_CONNECT=zookeeper:2181  
 - KAFKA\_ADVERTISED\_HOST\_NAME=kafka  
 - KAFKA\_ADVERTISED\_PORT=9092  
  
 chukwa:  
 image: apache/chukwa:latest  
 container\_name: chukwa  
 ports:  
 - "8080:8080"  
  
 elasticsearch:  
 image: elasticsearch:latest  
 container\_name: elasticsearch  
 ports:  
 - "9200:9200"  
 environment:  
 - discovery.type=single-node  
  
 memcached:  
 image: memcached:latest  
 container\_name: memcached  
 ports:  
 - "11211:11211"  
  
 hystrix:  
 image: hystrix/hystrix-dashboard:latest  
 container\_name: hystrix  
 ports:  
 - "8081:8081"

### **Step 3: Deploy Containers using Docker Compose**

* **Command:**

docker-compose up -d

* **Verify deployment:**

docker-compose ps

### **Step 4: Test the Netflix System Design API**

* **Send a test request:**

curl -X POST -H "Content-Type: application/json" -d '{"username": "john\_doe", "password": "password123"}' http://localhost:8000/user-service

* **Expected response:**

{  
 "status": "success",  
 "message": "User authenticated"  
}

#\*End of AI Generated Content\*