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

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

## **Deploying a Scalable and Fault-Tolerant Architecture**

#### **Step 1: Create Docker Images for Each Service**

----------------------------------------------- (font-size: 14px)

|  |  |  |
| --- | --- | --- |
| **Service** | **Dockerfile Creation** | **Image Creation Command** |
| **Video Processor** | create Dockerfile.video\_processor | docker build -t netflix-video-processor . |
| **Elastic Load Balancer** | create Dockerfile.elastic\_load\_balancer | docker build -t netflix-elastic-load-balancer . |
| **EV Cache (Memcached)** | use official memcached image | docker pull memcached |
| **Hystrix Service** | create Dockerfile.hystrix\_service | docker build -t netflix-hystrix-service . |
| **Kafka** | use official confluentinc/kafka image | docker pull confluentinc/kafka |
| **Apache Chukwa** | create custom Dockerfile.apache\_chukwa | docker build -t netflix-apache-chukwa . |
| **Elasticsearch** | use official elastic/elasticsearch image | docker pull elastic/elasticsearch |

**Dockerfile Examples:**

* Dockerfile.video\_processor

FROM python:3.9-slim  
WORKDIR /app  
COPY requirements.txt .  
RUN pip install -r requirements.txt  
COPY . .  
CMD ["python", "video\_processor.py"]

* Dockerfile.elastic\_load\_balancer

FROM python:3.9-slim  
WORKDIR /app  
COPY requirements.txt .  
RUN pip install -r requirements.txt  
COPY . .  
CMD ["python", "elastic\_load\_balancer.py"]

* Dockerfile.hystrix\_service

FROM python:3.9-slim  
WORKDIR /app  
COPY requirements.txt .  
RUN pip install -r requirements.txt  
COPY . .  
CMD ["python", "hystrix\_service.py"]

* Dockerfile.apache\_chukwa (create custom, as no official image)

FROM openjdk:8-jdk-alpine  
WORKDIR /app  
COPY apache-chukwa.tar.gz .  
RUN tar -xvf apache-chukwa.tar.gz  
CMD ["sh", "start-chukwa.sh"]

#### **Step 2: Configure Docker Compose for Orchestration**

--------------------------------------------------- (font-size: 14px)

Create a docker-compose.yml file to define and run multi-container Docker applications:

version: '3'  
services:  
 kafka:  
 image: confluentinc/kafka  
 ports:  
 - "9092:9092"  
 environment:  
 - KAFKA\_ZOOKEEPER\_CONNECT=localhost:2181  
 - KAFKA\_ADVERTISED\_LISTENERS=PLAINTEXT://kafka:9092  
  
 apache-chukwa:  
 image: netflix-apache-chukwa  
 ports:  
 - "8080:8080"  
 depends\_on:  
 - kafka  
  
 elasticsearch:  
 image: elastic/elasticsearch  
 ports:  
 - "9200:9200"  
 environment:  
 - discovery.type=single-node  
  
 memcached:  
 image: memcached  
 ports:  
 - "11211:11211"  
  
 video-processor:  
 image: netflix-video-processor  
 depends\_on:  
 - kafka  
 - apache-chukwa  
 - memcached  
 environment:  
 - KAFKA\_BOOTSTRAP\_SERVERS=kafka:9092  
 - CHUKWA\_URL=http://apache-chukwa:8080  
  
 elastic-load-balancer:  
 image: netflix-elastic-load-balancer  
 depends\_on:  
 - elasticsearch  
 environment:  
 - ELASTICSEARCH\_URL=http://elasticsearch:9200  
  
 hystrix-service:  
 image: netflix-hystrix-service  
 depends\_on:  
 - video-processor  
 - elastic-load-balancer  
 - memcached

#### **Step 3: Deploy with Docker Compose**

-------------------------------------- (font-size: 14px)

* **Initialize and start all services**:

docker-compose up -d

* **Verify container statuses**:

docker-compose ps

* **Scale services (optional)**:

docker-compose scale video-processor=3

* **Monitor and maintain your deployment**:

docker-compose logs -f

#\*End of AI Generated Content\*