

Docker Compose

NOTE: ?????? docker compose ?

NOTE: ??????????docker-compose.yml ?????????????????????? docker-compose ???
container??

????????????????

```
# For all services
docker-compose up -d

# For specified service
docker-compose up -d <service-name>
```

Build the image of the service

```
docker-compose build <service-name>
docker-compose up --rebuild <service-name>
```

????????????

```
docker-compose ps
```

???????????????????? container

```
docker-compose stop
docker-compose start
```

“ ?? container ???

???????????????????? container

```
# For all services
docker-compose down

# For a specified service
docker-compose stop <service-name>
docker-compose rm -f <service-name>
```

“ NOTE:

- ????? docker-compose.yml????? down ?????? up -d ?????
- ?????????????????? docker-compose up -d

??????????????

```
docker-compose logs
```

????????????????

```
docker-compose exec <service-name> sh -c "pwd"
```

????? `--env-file`

```
docker compose --env-file .env up -d
```

Labels

- ?????????????? containers ?????????????????? container????????????????????? Prometheus????????? label ???
- ??? `--label my-label=my-value`
- ?????????????????? com.myorg.XX ?????myorg ???????????
- ?????? [How I Use Docker Labels and Compose Tricks to Organize 75+ Containers - Virtualization Howto](#)

docker-compose.yml

```
services:
  web:
    image: myorg/web:2.3.1
```

```
labels:
  com.myorg.project: payment-api
  com.myorg.environment: production
  com.myorg.team: billing
  com.myorg.service-type: backend
  com.myorg.version: 2.3.1
redis:
  image: redis:6.2
  labels:
    com.myorg.project: payment-api
    com.myorg.environment: production
    com.myorg.team: billing
    com.myorg.service-type: cache
    com.myorg.version: "6.2"
```

?? --filter ???

```
docker ps --filter "label=com.myorg.project=payment-api"
docker inspect --format='{{json .Config.Labels}}' $(docker ps -q)
docker ps --filter label=com.myorg.environment=production --format "{{.Names}}: {{.Label
\"com.myorg.version\"}}"
```

.env

????????????API Token?????? .env ????????

```
# Make a '.env' file in the same directory.
env_file:
  - .env
```

.env

```
MYSQL_ROOT_PASSWORD=supersecurepassword
MYSQL_DATABASE=appdb
MYSQL_USER=appuser
MYSQL_PASSWORD=anothersecurepassword
```

docker-compose.yml

```
version: "3.9"
services:
  mysql:
    image: mysql:8
    environment:
      MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
      MYSQL_DATABASE: ${MYSQL_DATABASE}
      MYSQL_USER: ${MYSQL_USER}
      MYSQL_PASSWORD: ${MYSQL_PASSWORD}
```

```
frappe:
  image: frappe/bench:latest
  command: bash /workspace/init.sh
  environment:
    - SHELL=/bin/bash
  working_dir: /home/frappe
  volumes:
    - ./workspace
  ports:
    - 8000:8000
    - 9000:9000
```

HealthCheck

ping

```
healthcheck:
  test: ping -c 1 www.google.com || exit 1
  interval: 20s
  timeout: 10s
  retries: 5
```

environment + healthcheck

```
services:
  db:
    image: pgvector/pgvector:0.6.2-pg15
```

```
restart: always
ports:
  - '5432:5432'
environment:
  POSTGRES_USER: talkdai
  POSTGRES_PASSWORD: talkdai
  POSTGRES_DB: talkdai
volumes:
  - ./etc/db-extensions.sql:/docker-entrypoint-initdb.d/db-extensions.sql
healthcheck:
  test: ["CMD", "pg_isready", "-d", "talkdai", "-U", "talkdai"]
  interval: 10s
  timeout: 5s
  retries: 5
dialog:
  image: ghcr.io/talkdai/dialog:latest
  volumes:
    - ./data:/app/data/
  ports:
    - '8000:8000'
  depends_on:
    db:
      condition: service_healthy
environment:
  - PORT=8000
  - DATABASE_URL=postgresql://talkdai:talkdai@db:5432/talkdai
  - OPENAI_API_KEY=sk-your-openai-api-key
  - STATIC_FILE_LOCATION=/app/static
  - DIALOG_DATA_PATH=./data/your.csv
  - PROJECT_CONFIG=./data/your.toml
```

healthcheck(web) + build + networks

```
networks:
  net:
    driver: bridge

services:
  server:
```

```
image: server
build:
  context: .
  dockerfile: Dockerfile
volumes:
  # Be aware that indexed data are located in "/chroma/chroma/"
  # Default configuration for persist_directory in chromadb/config.py
  # Read more about deployments: https://docs.trychroma.com/deployment
  - chroma-data:/chroma/chroma
command: "--workers 1 --host 0.0.0.0 --port 8000 --proxy-headers --log-config
chromadb/log_config.yml --timeout-keep-alive 30"
environment:
  - IS_PERSISTENT=TRUE
  - CHROMA_SERVER_AUTHN_PROVIDER=${CHROMA_SERVER_AUTHN_PROVIDER}
  - CHROMA_SERVER_AUTHN_CREDENTIALS_FILE=${CHROMA_SERVER_AUTHN_CREDENTIALS_FILE}
  - CHROMA_SERVER_AUTHN_CREDENTIALS=${CHROMA_SERVER_AUTHN_CREDENTIALS}
  - CHROMA_AUTH_TOKEN_TRANSPORT_HEADER=${CHROMA_AUTH_TOKEN_TRANSPORT_HEADER}
  - PERSIST_DIRECTORY=${PERSIST_DIRECTORY:-/chroma/chroma}
  - CHROMA_OTEL_EXPORTER_ENDPOINT=${CHROMA_OTEL_EXPORTER_ENDPOINT}
  - CHROMA_OTEL_EXPORTER_HEADERS=${CHROMA_OTEL_EXPORTER_HEADERS}
  - CHROMA_OTEL_SERVICE_NAME=${CHROMA_OTEL_SERVICE_NAME}
  - CHROMA_OTEL_GRANULARITY=${CHROMA_OTEL_GRANULARITY}
  - CHROMA_SERVER_NOFILE=${CHROMA_SERVER_NOFILE}
restart: unless-stopped # possible values are: "no", "always", "on-failure", "unless-
stopped"
ports:
  - "8000:8000"
healthcheck:
  # Adjust below to match your container port
  test: [ "CMD", "curl", "-f", "http://localhost:8000/api/v1/heartbeat" ]
  interval: 30s
  timeout: 10s
  retries: 3
networks:
  - net

volumes:
  chroma-data:
    driver: local
```

vlan network

???Container ??? IP

?? macvlan ??

```
docker network create -d macvlan \  
  --subnet=192.168.0.0/24 \  
  --gateway=192.168.0.1 \  
  --ip-range=192.168.0.100/28 \  
  -o parent=eth0 vlan
```

?? compose.yml

```
services:  
  windows:  
    container_name: windows  
    ..<snip>..  
    networks:  
      vlan:  
        ipv4_address: 192.168.0.100  
  
networks:  
  vlan:  
    external: true
```

Command

```
frappe:  
  image: frappe/bench:latest  
  command: bash /workspace/init.sh  
  environment:  
    - SHELL=/bin/bash  
  working_dir: /home/frappe  
  volumes:  
    - ./workspace  
  ports:  
    - 8000:8000  
    - 9000:9000
```

```
version: "3.7"
services:
  mariadb:
    image: mariadb:10.8
    command:
      - --character-set-server=utf8mb4
      - --collation-server=utf8mb4_unicode_ci
      - --skip-character-set-client-handshake
      - --skip-innodb-read-only-compressed # Temporary fix for MariaDB 10.6
    environment:
      MYSQL_ROOT_PASSWORD: 123
    volumes:
      - mariadb-data:/var/lib/mysql
```

Misc

- `pull_policy: always` ?? container ???????? image?

Revision #26

Created 2020-06-19 12:05:07 CST by A-Lang (Admin)

Updated 2026-03-30 21:53:48 CST by A-Lang (Admin)