DEVHINTS.IO

Edit

docker-compose cheatsheet

Basic example

```
# docker-compose.yml
version: '2'

services:
  web:
    build: .
    # build from Dockerfile
    context: ./Path
    dockerfile: Dockerfile
    ports:
     - "5000:5000"
    volumes:
     - .:/code
  redis:
    image: redis
```

Commands

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

```
docker-compose pause
docker-compose unpause
```

```
docker-compose ps
docker-compose up
docker-compose down
```

Reference

Building

```
web:
  # build from Dockerfile
  build: .
  args:     # Add build arguments
    APP_HOME: app
```

```
  # build from custom Dockerfile
  build:
    context: ./dir
    dockerfile: Dockerfile.dev
```

Ports

```
ports:
  - "3000"
  - "8000:80"  # host:container
```

```
# expose ports to linked services (not to host)
expose: ["3000"]
```

Commands

```
# command to execute
command: bundle exec thin -p 3000
command: [bundle, exec, thin, -p, 3000]
```

```
# override the entrypoint
entrypoint: /app/start.sh
entrypoint: [php, -d, vendor/bin/phpunit]
```

```
# build from image
image: ubuntu
image: ubuntu:14.04
image: tutum/influxdb
image: example-registry:4000/postgresql
image: a4bc65fd
```

Other options

```
# make this service extend another
extends:
  file: common.yml  # optional
  service: webapp
```

```
volumes:
  - /var/lib/mysql
  - ./_data:/var/lib/mysql
```

Environment variables

```
# environment vars
environment:
  RACK_ENV: development
environment:
  - RACK_ENV=development
```

```
# environment vars from file
env_file: .env
env_file: [.env, .development.env]
```

Dependencies

```
# makes the `db` service available as the hostn
# (implies depends_on)
links:
  - db:database
  - redis
```

```
# make sure `db` is alive before starting
depends_on:
  - db
```

Advanced features

Labels

```
services:
  web:
    labels:
      com.example.description: "Accounting web ap
```

Hosts

```
services:
  web:
    extra_hosts:
      - "somehost:192.168.1.100"
```

External network

DNS servers

```
services:
  web:
    dns: 8.8.8.8
    dns:
      - 8.8.8.8
      - 8.8.4.4
```

Network

```
# creates a custom network called `frontend`
networks:
  frontend:
```

Volume

Devices

```
services:
  web:
    devices:
    - "/dev/ttyUSB0:/dev/ttyUSB0"
```

External links

```
services:
  web:
    external_links:
      - redis_1
      - project_db_1:mysql
```

```yaml
# join a pre-existing network
networks:
  default:
    external:
      name: frontend
```

```yaml
# Mount host paths or named volumes, specified as
  db:
    image: postgres:latest
    volumes:
      - "/var/run/postgres/postgres.sock:/var/run
      - "dbdata:/var/lib/postgresql/data"

volumes:
  dbdata:
```

▶ **5 Comments** for this cheatsheet. Write yours!

devhints.io / Search 356+ cheatsheets

Over 356 curated cheatsheets, by developers for developers.

Devhints home

Other Devops cheatsheets

- AWS CLI cheatsheet
- Chef cheatsheet
- CircleCI cheatsheet
- Deis cheatsheet
- Docker CLI cheatsheet
- Dockerfile cheatsheet

Top cheatsheets

- Elixir cheatsheet
- ES2015+ cheatsheet
- React.js cheatsheet
- Vimdiff cheatsheet
- Vim cheatsheet
- Vim scripting cheatsheet