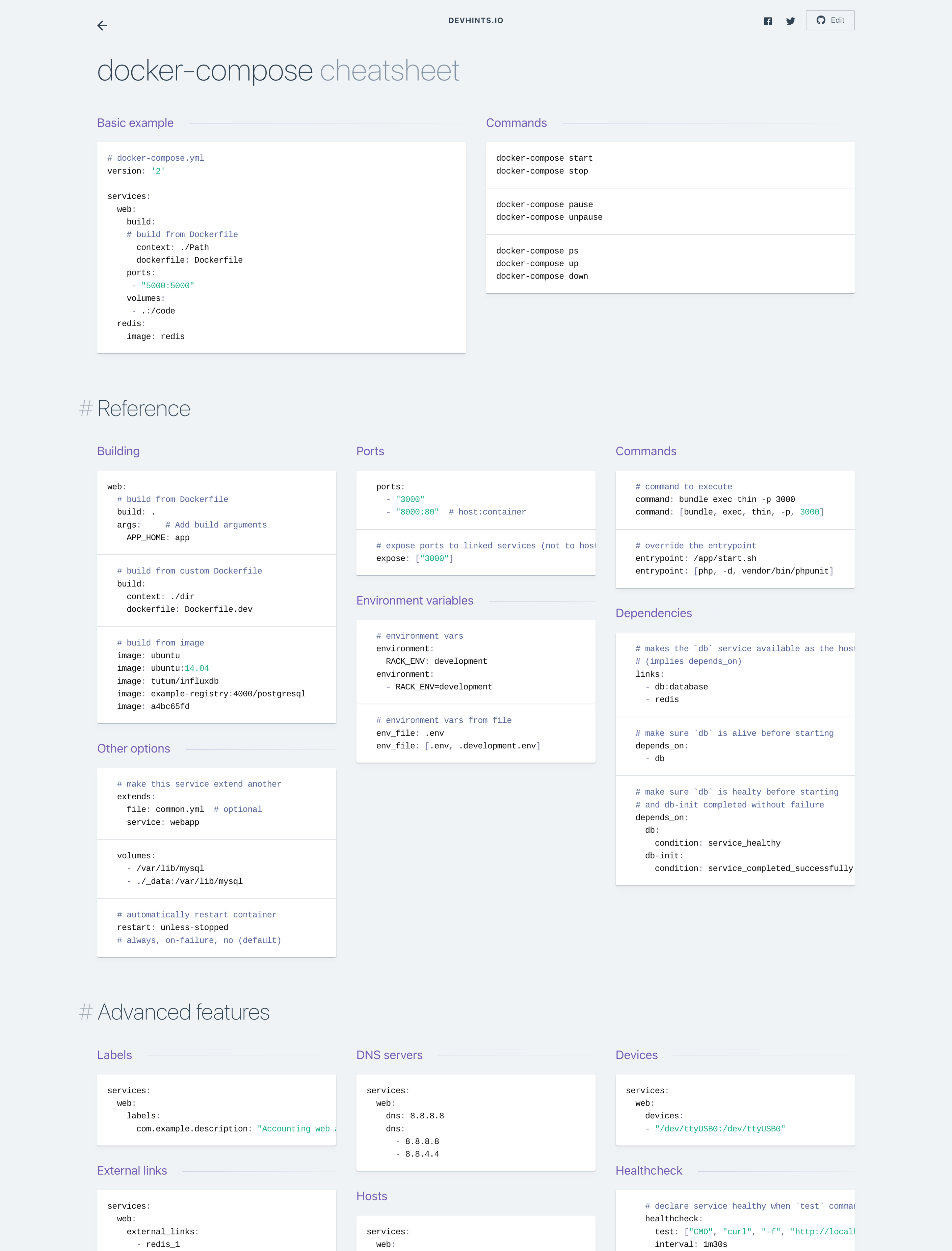

docker-compose cheatsheet

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
```

```
  # 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
```

```
  # automatically restart container
  restart: unless-stopped
  # always, on-failure, no (default)
```

Ports

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

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

Environment variables

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

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

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]
```

Dependencies

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

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

```
  # make sure `db` is healty before starting
  # and db-init completed without failure
  depends_on:
    db:
      condition: service_healthy
    db-init:
      condition: service_completed_successfully
```

Advanced features

Labels

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

External links

```
services:
  web:
    external_links:
      - redis_1
```

DNS servers

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

Hosts

```
services:
  web:
```

Devices

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

Healthcheck

```
    # declare service healthy when `test` comman
    healthcheck:
      test: ["CMD", "curl", "-f", "http://localh
      interval: 1m30s
      - project_db_1:mysql
```

Network

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

User

```
# specifying user
user: root
```

```
# specifying both user and group with ids
user: 0:0
```

```
    extra_hosts:
      - "somehost:192.168.1.100"
```

External network

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

```
        timeout: 10s
        retries: 3
        start_period: 40s
```

Volume

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

volumes:
  dbdata:
```

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

devhints.io / Search 357+ cheatsheets

Over 357 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