

DOCKER COMPOSE CHEAT SHEET

File

structure

```
services:  
  container1:  
    properties: values  
  
  container2:  
    properties: values
```

```
networks:  
  network:
```

```
volumes:  
  volume:
```

Types

value

```
key: value
```

array

```
key:  
  - value  
  - value
```

dictionary

```
master:  
  key: value  
  key: value
```

Properties

build

build image from dockerfile in specified directory

```
container:  
  build: ./path  
  image: image-name
```

image

use specified image

```
image: image-name
```

container_name

define container name to access it later

```
container_name: name
```

volumes

define container volumes to persist data

```
volumes:  
  - /path:/path
```

command

override start command for the container

```
command: execute
```

environment

define env variables for the container

```
environment:  
  KEY: VALUE
```

```
environment:  
  - KEY=VALUE
```

env_file

define a env file for the container to set and override env variables

```
env_file: .env
```

```
env_file:  
  - .env
```

restart

define restart rule (no, always, on-failure, unless-stopped)

```
restart:  
  - "9999"
```

networks

define all networks for the container

```
networks:  
  - network-name
```

ports

define ports to expose to other containers and host

```
ports:  
  - "9999:9999"
```

expose

define ports to expose only to other containers

```
expose:  
  - "9999"
```

network_mode

define network driver (bridge, host, none, etc.)

```
network_mode: host
```

depends_on

define build, start and stop order of container

```
depends_on:  
  - container-name
```

Other

idle container

send container to idle state
> container will not stop

```
command: tail -f /dev/null
```

named volumes

create volumes that can be used in the volumes property

```
services:  
  container:  
    image: image-name  
    volumes:  
      - data-  
volume:/path/to/dir
```

```
volumes:  
  data-volume:
```

networks

create networks that can be used in the networks property

```
networks:  
  frontend:  
    driver: bridge
```

