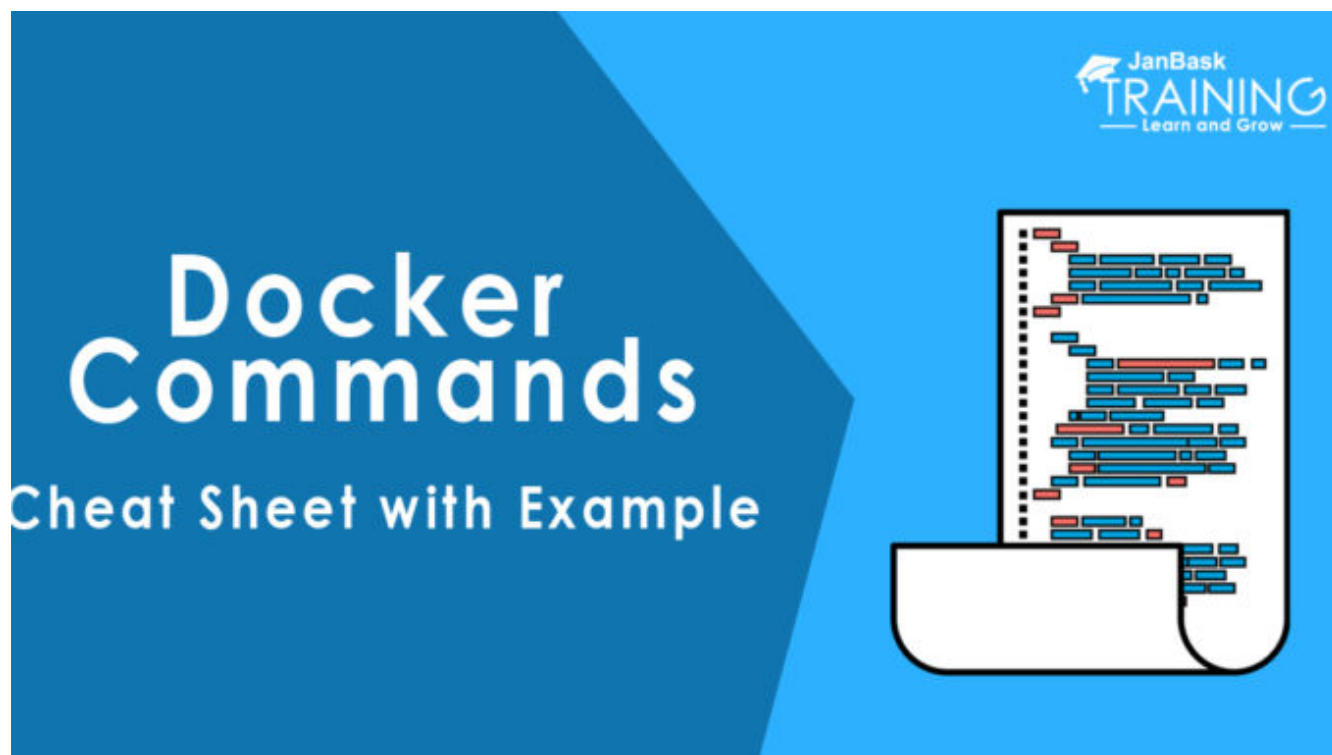


Docker Cheat Sheet

© 05.07.2019  Cinex  Linux, Веб-розробка, Різне  0



ПОШУК

ПОЗНАЧКИ

[APACHE](#)[AUTORUN](#)[BASH](#)[BITCOIN](#)[BLOCKCHAIN](#)[COMPOSER](#)[CRYPTOCURRENCY](#)[CSS](#)[DATABASE](#)[DJANGO](#)[DOCKER](#)[ELOQUENT](#)[ENGLISH](#)[ENUM](#)[GAMES](#)[GIMP](#)[GIT](#)[HTML](#)[JAVA](#)[JAVASCRIPT](#)[JQUERY](#)[LARAVEL](#)[MERCURIAL](#)[PARSING](#)[PHOTOSHOP](#)[PHP](#)[PHPSTORM](#)

Зміст

- Встановлення
- Реєстри та репозиторії Docker
- Перші дії з контейнерами
- Запуск і зупинка контейнерів
- Отримання інформації про контейнери
- Мережа
- Очищення Docker
- Docker Swarm

Встановлення

Linux

Більше інформації [тут](#)

```
1 curl -sSL https://get.docker.com/ | sh
```

Mac

Більше інформації [тут](#)

Завантажте dmg по цьому посиланню.

```
1 https://download.docker.com/mac/stable/Docker.dmg
```

[PLAYONLINUX](#)[PLUGINS](#)[PROBLEM SOLVING](#)[PYTHON](#)[REGEXP](#)[SEO](#)[SOFTWARE ENGINEERING](#)[SPRING](#)[SQL](#)[TELEGRAM](#)[TIPS AND TRICKS](#)[TIPS AND TRICKS](#)[UNIT TESTING](#)[WORDPRESS](#)[ВІРШІ](#)[ШПАРГАЛКИ](#)

ОСТАННІ НОТАТКИ

20 Laravel Eloquent порад і трюків

15.12.2019

Як зробити скріншот сайту по URL на PHP

04.08.2019

Docker Cheat Sheet

05.07.2019

Гарячі клавіші Ubuntu Linux

01.07.2019

Git happens! 6 типових помилок Git і як їх виправити

22.06.2019

НЕДАВНІ КОМЕНТАРІ

Windows

Більше інформації [тут](#)

Використовуйте MSI-інсталятор:

```
1 https://download.docker.com/win/stable/InstallDocker.msi
```

Реєстри та репозиторії Docker

Вхід в реєстр

```
1 docker login
```

```
1 docker login localhost:8080
```

Вихід з реєстра.

```
1 docker logout
```

```
1 docker logout localhost:8080
```

Пошук образу

```
1 docker search nginx
```

```
1 docker search --filter stars=3 --no-trunc nginx
```

Pull (вивантаження з реєстру) образу

```
1 docker image pull nginx
```

```
1 docker image pull eon01/nginx localhost:5000/myadmin/nginx
```

Push (завантаження в реєстр) образу

```
1 docker image push eon01/nginx
```

```
1 docker image push eon01/nginx localhost:5000/myadmin/nginx
```

Перші дії з контейнерами

Створення і запуск простого контейнера

- Запустіть образ **ubuntu:latest**
- Зв'яжіть порт **80 КОНТЕЙНЕРА** з портом **3000 ХОСТА**
- Змонтуйте поточну директорію у **/data** на КОНТЕЙНЕРІ
- Замітка: на **windows** ви повинні змінити **-v \${PWD}:/data** на **-v "C:\Data":/data**

```
1 docker container run --name infinite -it -p 3000:80 -v ${PWD}:/data ubuntu:latest
```

Створення контейнера

```
1 docker container create -t -i eon01/infinite --name infinite
```

Запуск контейнера

```
1 docker container run -it --name infinite -d eon01/infinite
```

Перейменування контейнера

```
1 docker container rename infinite infinity
```

Видалення контейнера

```
1 docker container rm infinite
```

Оновлення контейнера

```
1 docker container update --cpu-shares 512 -m 300M infinite
```

Запуск і зупинка контейнерів

Запуск

```
1 docker container start nginx
```

Зупинка

```
1 docker container stop nginx
```

Перезапуск

```
1 docker container restart nginx
```

Пауза (припинення всіх процесів контейнера)

```
1 docker container pause nginx
```

Зняття паузи

```
1 docker container unpause nginx
```

Блокування (до зупинки контейнера)

```
1 docker container wait nginx
```

Відправка SIGKILL (завершального сигналу)

```
1 docker container kill nginx
```

Відправка іншого сигналу

```
1 docker container kill -s HUP nginx
```

Підключення до існуючого контейнера

```
1 docker container attach nginx
```

Отримання інформації про контейнери

Працюючі контейнери

```
1 docker container ls
```

```
1 docker container ls -a
```

Логи контейнеру

```
1 docker logs infinite
```

Слідувати логам контейнера (висновок логів з оновленнями в реальному часі)

```
1 docker container logs infinite -f
```

Інформація про контейнер

```
1 docker container inspect infinite
```

```
1 docker container inspect --format '{{.NetworkSettings.IPAddress}}' $(docker ps -q)
```

Події контейнера

```
1 docker system events infinite
```

Публічні порти

```
1 docker container port infinite
```

Виконуються процеси

```
1 docker container top infinite
```

Використання ресурсів

```
1 docker container stats infinite
```

Зміни в файлах або директоріях файлової системи контейнера

```
1 docker container diff infinite
```

Управління образами

Список образов

```
1 docker image ls
```

Створення образів

```
1 docker build .
```

```
1 docker build github.com/creack/docker-firefox
```

```
1 docker build - < Dockerfile
```

```
1 docker build - < context.tar.gz
```

```
1 docker build -t eon/infinite .
```

```
1 docker build -f myOtherDockerfile .
```

```
1 curl example.com/remote/Dockerfile | docker build -f - .
```

Видалення способу

```
1 docker image rm nginx
```

Завантаження сховища в tar (з файлу або стандартного вводу)

```
1 docker image load < ubuntu.tar.gz
```

```
1 docker image load --input ubuntu.tar
```

Збереження образу в tar-архів

```
1 docker image save busybox > ubuntu.tar
```

Перегляд історії образу

```
1 docker image history
```

Створення образу з контейнера

```
1 docker container commit nginx
```

Тегування способу

```
1 docker image tag nginx eon01/nginx
```

Push (завантаження до реєстру) способу

```
1 docker image push eon01/nginx
```

Мережа

Створення мережі

```
1 docker network create -d overlay MyOverlayNetwork
```

```
1 docker network create -d bridge MyBridgeNetwork
```

```
1 docker network create -d overlay \  
2 --subnet=192.168.0.0/16 \  
3 --subnet=192.170.0.0/16 \  
4 --gateway=192.168.0.100 \  
5 --gateway=192.170.0.100 \  
6 --ip-range=192.168.1.0/24 \  
7 --aux-address="my-router=192.168.1.5" --aux-address="my-switch=192.168.1.6" \  
8 --aux-address="my-printer=192.170.1.5" --aux-address="my-nas=192.170.1.6" \  
9 MyOverlayNetwork
```

Видалення мережі

```
1 docker network rm MyOverlayNetwork
```

Список мереж

```
1 docker network ls
```

Отримання інформації про мережу

```
1 docker network inspect MyOverlayNetwork
```

Підключення працюючого контейнера до мережі

```
1 docker network connect MyOverlayNetwork nginx
```

Підключення контейнера до мережі при його запуску

```
1 docker container run -it -d --network=MyOverlayNetwork nginx
```

Відключення контейнера від мережі

```
1 docker network disconnect MyOverlayNetwork nginx
```

Exposing Ports

Використовуючи Dockerfile, ви можете розкрити порт в контейнері використовуючи:

```
1 EXPOSE <port_number>;
```

You can also map порт контейнера to порт хоста використовуючи:

Наприклад,

```
1 docker run -p $HOST_PORT:$CONTAINER_PORT --name infinite -t infinite
```

Очищення Docker

Видалення працюючого контейнера

```
1 docker container rm nginx
```

Видалення контейнера і його томи (volume)

```
1 docker container rm -v nginx
```

Видалення всіх контейнерів зі статусом exited

```
1 docker container rm $(docker container ls -a -f status=exited -q)
```

Видалення всіх зупинених контейнерів

```
1 docker container rm $(docker container ls -a -q)
```

Видалення способу

```
1 docker image rm nginx
```

Видалення невикористовуваних (dangling) образів

```
1 docker image rm $(docker image ls -f dangling=true -q)
```

Видалення всіх образів

```
1 docker image rm $(docker image ls -a -q)
```

Видалення всіх образів без тегів

```
1 docker image rm -f $(docker image ls | grep "^&lt;none&gt;" | awk "{print $3}")
```

Зупинка і видалення всіх контейнерів

```
1 docker container stop $(docker container ls -a -q) && docker container rm $(docke
```

Видалення невикористовуваних (dangling) томів

```
1 docker volume rm $(docker volume ls -f dangling=true -q)
```

Removing all unused (containers, images, networks and volumes)

```
1 docker system prune -f
```

Повне очищення

```
1 docker system prune -a
```

Docker Swarm

Установка Docker Swarm

```
1 curl -ssl https://get.docker.com | bash
```

Ініціалізація Swarm

```
1 docker swarm init --advertise-addr 192.168.10.1
```

Підключення робочого вузла (worker) до Swarm

```
1 docker swarm join-token worker
```

Підключення керуючого вузла (manager) до Swarm

```
1 docker swarm join-token manager
```

Список сервісів

```
1 docker service ls
```

Список вузлів

```
1 docker node ls
```

Створення сервісу

```
1 docker service create --name vote -p 8080:80 instavote/vote
```

Список завдань Swarm

```
1 docker service ps
```

Масштабування сервісу

```
1 docker service scale vote=3
```

Оновлення сервісу

```
1 docker service update --image instavote/vote:movies vote
```

```
1 docker service update --force --update-parallelism 1 --update-delay 30s nginx
```

```
1 docker service update --update-parallelism 5--update-delay 2s --image instavote/vote:inde
```

```
1 docker service update --limit-cpu 2 nginx
```

```
1 docker service update --replicas=5 nginx
```

Джерело



DOCKER

TIPS AND TRICKS

ШПАРГАЛКИ



« ПОПЕРЕДНІЙ

Гарячі клавіші Ubuntu Linux

ДАЛІ »

Як зробити скріншот сайту по URL
на PHP



ЗАЛИШТЕ ПЕРШИЙ КОМЕНТАР

Залишити коментар

Вашу адресу електронної пошти не буде опубліковано

Коментувати

Ім'я*

Email *

☐ Збережіть моє ім'я, електронну пошту у цьому веб-переглядачі наступного разу, коли я коментуватиму.

ОПУБЛІКУВАТИ КОМЕНТАР

Copyright © 2020