Docker CMD

nginx	docker run --name my-ninix -d -p 8080:80 nginx
Attach a **shell**	docker exec -it <container-name> /bin/bash
Lister container actif	docker container ls -a
Supprime tous les conteneurs inactifs -q affiche la liste des id des conteneur	docker rm $(docker ps -a -f status=exited -f status=created -q)

Docker File

Build IMAGE *(depuis le répertoire)*	docker build -t site-image .
Write html in doc *(dockerfile)*	RUN echo hello > /var/www/html
Build image + **args**	docker build -t site-args --build-arg ARG_PRENOM=Etienne .

Docker compose

Run compose.yaml file (in directory)	docker compose up -d
Supprimer les conteneur	docker compose stop

Docker network

Create net work	docker network create --driver=bridge --subnet=10.71.122.0/16 --ip-range=10.71.122.0/24 --gateway=10.71.122.14 my-spider-net
Connect	docker network connect my-spider-net web-args
Choose IP / on RUN	docker run --name my-spider-contaner -d -p 8080:80 site-args
Show net config	docker network inspect dcea2b…

Docker Volume

Create	docker volume create 130volt
Run container + volume	docker run --name my-volume-contaner1 -d -p 8080:80 -v 130volt:/usr/share/nginx/html site-args
Choose IP / on RUN	docker run --name my-spider-contaner -d -p 8080:80 nginx
Show net config	docker network inspect dcea2b…

DockerFile

Limiter les couches + limiter erreur sur update / upgrade *(garde en cache les couches précédente)* *RUN === 1 couche*	RUN apt-get update && apt-get install -y \ package-1 \ package-2 \ && rm -rf /var/lib/apt/lists*
run container + volume	docker run --name my-volume-contaner1 -d -p 8080:80 -v 130volt:/usr/share/nginx/html site-args
Gestion des droits CREATE USER	RUN adduser my-ninix-user
WITCH USER	USER my-user
READ ONLY	
RESSOURCE LIMIT	

Surveillance

DOCKER SCOUT	docker scout quickview ubuntu
run container + volume	docker run --name my-volume-contaner1 -d -p 8080:80 -v 130volt:/usr/share/nginx/html
Gestion des droits CREATE USER	RUN useradd my-ninix-user
WITCH USER	USER my-user
GRANT USER	`chown -R appUser:appUser /var/www/html`
RESSOURCE LIMIT	

Système impératif => fonctionne sur des ordre

Système déclarative => j'ai besoins d'une instance , une application, j'ai besoin de toi pour que tu me les apportes.

cf. IMP_DEC.png

CLUSTER => *Dire que 2 server SQL fonctionnent en cluster signifie qu'ils y a un liens entre les deux unité et que notre endPoint pour accéder aux ressource sera le même pour les deux.*

cluster applicatif

cf. cluster.PNG

Kubernetes

POD

CREAT POD	kubeclt run <podName> --image=ngnix

K8S == 2 mod cluster / test *(local)*

config map

CREAT MAP	kubeclt run <podName> --image=ngnix