

### Docker cheat sheet serie



### Orchestration de conteneurs

`docker-compose up` Lance l'ensemble des conteneurs

`docker-compose stop` Stoppe l'ensemble des conteneurs

`docker-compose kill` Tue l'ensemble des conteneurs

`docker-compose exec` Exécute une commande dans le  
<service> <command> conteneur

### Raccourcis clavier en exécution

**Ctrl+c** stoppe les conteneurs (équivalent à `docker-compose stop`)

**Ctrl+c x2** tue les conteneurs (équivalent à `docker-compose kill`)

### Supprimer un volume de données

Supprimer le conteneur MySQL

```
docker-compose rm -v <service name>
ex:docker-compose rm -v db
```

Lister les volumes présents

```
docker volume ls
```

Supprimer le volume de données approprié

```
docker volume rm <volume name>
```

\* La suppression d'un volume est nécessaire en cas de downgrade de base de données par exemple.

⚠ **La suppression d'un volume de données supprime la base de données associée. Penser à effectuer une sauvegarde au préalable !**

### Scalabilité avec swarm

Lancer les conteneurs en arrière plan

```
docker-compose up -d
```

Dimensionner un service (nécessite un service fontal)

```
docker-compose scale <service name>=<qté>
ex:docker-compose scale web=4
```

ⓘ le port 80 ne peut être publié qu'une fois vers l'hôte.

🔗 utiliser un service en reverse proxy vers le service `http://web/`

→ le DNS `web` est un round-robin sur `web-1`, `web-2`, `web-3`, `web-4`

➡ pensez également au partage de sessions (memcached par exemple)

### ⚠ Modifications éphémères !

\* Les modifications structurelles effectuées dans un conteneur par l'intermédiaire de la commande `docker-compose exec` (installation de paquet, etc.) ne sont **jamais** persistées !

\* Lorsque l'on stoppe ou tue le conteneur, **les modifications sont supprimées !**

\* Les modifications doivent-être répercutées dans l'image docker utilisée (dans le `Dockerfile` approprié).

### > Exemple de commandes avec exec

**Retrer** dans un conteneur pour un shell interactif

```
docker-compose exec web bash
```

Lancer les **tests d'acceptance** avec Codeception

```
docker-compose exec web codecept run acceptance
```

Mettre à jour `composer`

```
docker-compose exec web composer -n self-update
```

Installer les dépendances `composer`

```
docker-compose exec web composer -n install
```

Mettre à jour les dépendances `composer`

```
docker-compose exec web composer -n update
```

Installer les dépendances `bower`

```
docker-compose exec web bower -n install
--config.interactive=false --allow-root
```

\* Dans les exemple ci-dessus, nous utilisons `web` comme service depuis lequel nous souhaitons exécuter les commandes.

\* Utiliser `exit` pour sortir d'un conteneur et revenir à l'hôte.

\* Le drapeau `-n` permet d'indiquer à `composer` qu'on l'exécute en mode non interactif

### docker-compose

