

docker  
Compose

# Objectifs

- Notion d'orchestrateur
- Docker-compose
  - Premières commandes
- Présentation du produit Kubernetes (k8s)

# Dans les cours précédents...

- Nous avons vu comment :
  - Créer une image
  - Créer un conteneur
  - Les faire « parler » ensemble
- Les opérations sont faites principalement à la main : on crée chaque conteneur en ligne de commande et non à partir de configurations
- L'utilisation **simultanée de plusieurs conteneurs** à mis en lumière la notion de dépendance :
  - Un conteneur peut **dépendre d'un autre** (ex. Wordpress dépend de MySQL)
  - Un conteneur doit **exister avant l'autre** (ex. MySQL avant le démarrage Wordpress)
  - Docker compose permet de faire cela et plus.

# Fichier docker-compose.yml

- Un fichier Docker Compose est un fichier texte, qui se termine par l'extension yml ou yaml. Vous pouvez l'écrire avec l'éditeur de texte de votre choix.
- À la suite de la rédaction du fichier,
  - exécutez simplement la commande `docker-compose up` pour démarrer l'infrastructure,
  - et `[Ctrl] C` pour l'arrêter.

# Syntaxe du fichier Compose

- **1. Structure d'un fichier Compose**

- Le fichier Docker Compose est composé d'une suite de blocs reprenant les éléments essentiels du CLI. Ainsi on retrouvera, entre autres, les blocs suivants :
- **Version** : définit la version du moteur de Compose à utiliser.
- **Services** : comprend les conteneurs que vous souhaitez mettre en service, avec leurs options.
- **Réseaux** : comprend les réseaux à mettre en place ou à utiliser.
- **Volumes** : comprend les volumes à créer ou à utiliser.

# Exemple de fichier docker-compose

```
version: "3.9"
services:
  web:
    build: .
    ports:
      - "5000:5000"
  redis:
    image: "redis:alpine"
```

```
version: "3.9"
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./code
    environment:
      FLASK_ENV: development
  redis:
    image: "redis:alpine"
```

# Activer Docker-Compose



Upgrade



## Settings

General

Resources

Docker Engine

Experimental Features

Kubernetes

Software Updates

## General

☐ Start Docker Desktop when you log in

☐ Expose daemon on tcp://localhost:2375 without TLS

Exposing daemon on TCP without TLS helps legacy clients connect to the daemon. It also makes yourself vulnerable to remote code execution attacks. Use with caution.

☒ Use the WSL 2 based engine

WSL 2 provides better performance than the legacy Hyper-V backend. [Learn more.](#)

☒ Send usage statistics

Send error reports, system version and language as well as Docker Desktop lifecycle information (e.g., starts, stops, resets).

☒ Show weekly tips

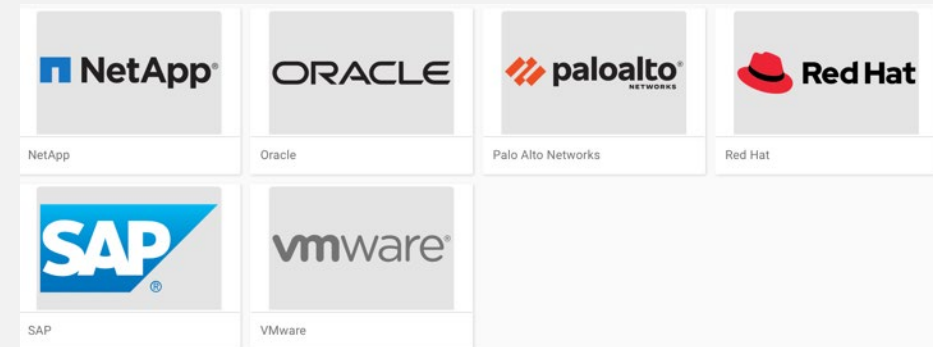
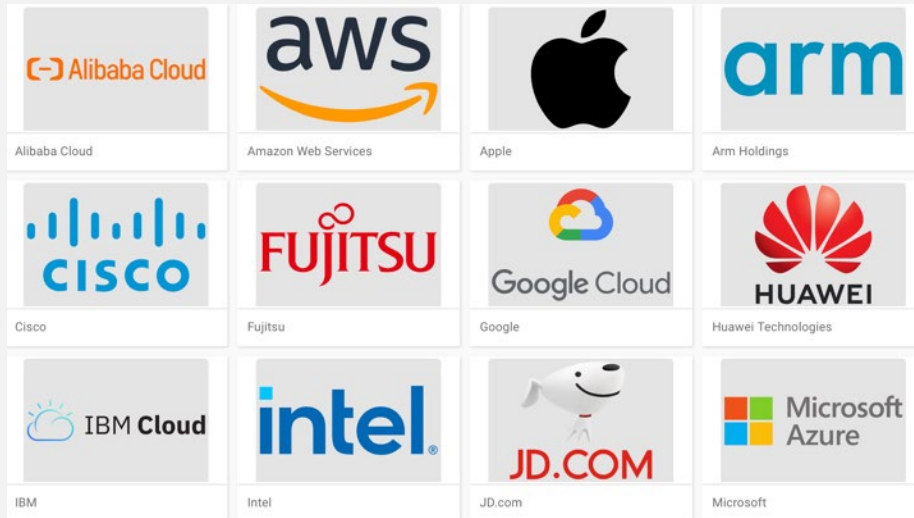
☒ Open Docker Dashboard at startup

☒ Use Docker Compose V2

Enables the docker-compose command to use Docker Compose V2. [Learn More.](#)

# kubernetes

- Projet open source originaire de Google<sup>(1)</sup> :
  - Le nom vient du grec kubernesis : piloter / gouverner
  - Souvent simplifié en k8s (débuté par k, 8 lettres, termine par s)
  - Basé sur Borg 2004, produit en version 1.0 en 2015<sup>(2)</sup>
  - Fait parti des projets du Cloud Native Computing Foundation (CNCF)



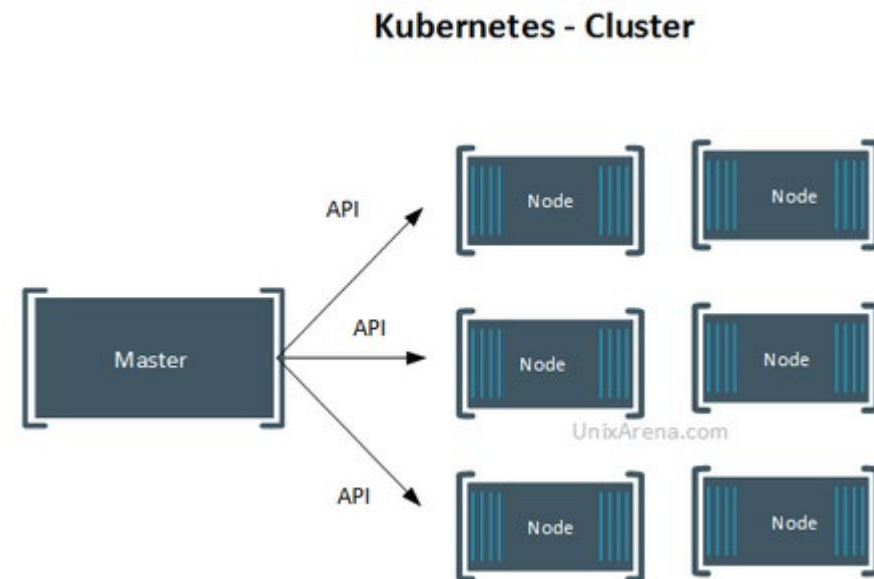
(1) <https://kubernetes.io>

(2) <https://research.google/pubs/pub43438/>



# Notions et vocabulaire k8s

- Nœud (serveur) : physique ou virtuel
  - Maître : gestion des nœuds de calcul, s'assure que les configurations sont respectées au mieux
  - Nœud de calcul : nœud où s'exécute un ensemble de « pods »
- Grappe kubernetes (cluster) : ensemble de nœuds dont des nœuds maîtres et des nœuds de calcul



# Les différents types de nodes

```
$ kubectl get nodes
NAME        STATUS    ROLES    AGE   VERSION
kube-01     Ready     master   32m   v1.13.4
kube-02     Ready     <none>   24m   v1.13.4
kube-03     Ready     <none>   24m   v1.13.4
```

## Master

- Responsable de la gestion du cluster (« Control plane »)
- Expose l'API server
- Schedule les Pods sur les nodes du cluster

## Worker /Node

- Node sur lequel sont lancés les Pods applicatifs
- Communique avec le Master
- Fournit les ressources aux Pods

# Démonstration depuis notre infrastructure

▼ Perm\_ITV\_K8S

- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_m01
- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_w01
- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_w02
- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_w03
- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_w04
- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_w05
- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_w06
- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_w07
- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_w08
- ▶ Perm\_W44\_IV\_JP\_k8s\_c02\_w09

▶ Perm\_W44\_IV\_JP\_k8s\_c02\_m01



ACTIONS ▼

Résumé

Surveiller

Configurer

Autorisations

Banques de données

Réseaux

Mises à jour



[Lancer la console Web](#)

[Lancer Remote Console](#)

SE invité : Ubuntu Linux (64-bit)  
Compatibilité : ESXi 6.7 et versions ultérieures (VM version 14)  
VMware Tools : En cours d'exécution, version :11333 (Invité géré)  
[Plus d'infos](#)  
Nom DNS : ivk8sc02m01  
Adresses IP : 10.100.2.90  
[Afficher toutes les 8 adresses IP](#)  
Hôte : esx-dfc-1.cegep-ste-foy.qc.ca



UTILISATION DU CPU

615 MHz



UTILISATION DE LA MÉMOIRE

1,36 Go



UTILISATION DU STOCKAGE

88,08 Go

# Références

- Site officiel : <https://kubernetes.io/fr/>
- Commandes kubectl :  
<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands>
- Si vous êtes intéressés par les statistiques :  
<https://k8s.devstats.cncf.io/d/12/dashboards?orgId=1&refresh=15m>