

# Kubernetes

## Table of Contents

C'est quoi ? .....	1
Vocabulaires .....	2
Deployer un cluster Kubernetes .....	2
Dépendances requises .....	2
Sur Windows - Vagrant: Deploy 1 cluster on Kubernetes .....	2
Sur Linux : Deployer un mutli cluster .....	3
demonstation .....	4

## C'est quoi ?

Kubernetes est un projet developpé par Google, outil d'orchestration de container avec très performant avec beaucoup de fonctionnalité.

Kubernetes se place au début de la liste des outils les plus performant en 2018 par ses nombreux points fort par rapport à d'autre outils de d'orchestration de container.

Kubernetes ou k8s n'est pas tout de même l'un des outils les plus facile à mettre en place, pas pour autant de bien maitriser. Pour faire fonctionner correctement K8s, il en est d'avoir une bonne pratique et de methode.

K8s contient plusieurs module, l'outil est modulable en fonction de choix de son utilisateur. Les points forts de K8s sont:

### Lightweight

Pour faire tourner l'outil, il n'est pas necessaire d'avoir un monstre puisque en memoire ou en processeur :), l'application ne compte que quelque mega octet dememoire.

### Realability

la fiabilité est aussi l'un de ces points fort contrairement à ces concurant. Kubernetes assure la disponibilité de ces containers en ajustant automatiquement les services déjà disponible.

### Scalable

K8s est modulable, tous ces services sont remplaçables par un autre module. K8s loadbalance automatiquement les nodes orphelins pour optimiser la mise en diponibilité.

Le point fort de kubernetes est 'LA HAUTE DISPONIBILITE' de tous les containers qui eux executes differents charges de travail et different flux de données sur une infrastructure donné tel qu'une VM ou une bare metal ou sur le cloud.

# Vocabulaires

## **cluster**

a Cluster est une collection de host et de ressource.

## **node**

a node est un host qui peut etre un VM, une bare metal.

## **master**

le master est le plan de control de Kubernetes. Il contient l'API server, le Scheduler, et le controle manager

## **pod**

le pod est une unité de travail dans kubernetes.

1. Un pod contient un ou plusieurs containers
2. un pod à le meme adresse IP et même port
3. communique en localhost ou avec un standard inter-procès de communication
4. partage un meme dossier
5. label
6. annotation
7. label selection
8. volume
9. PetSet
10. Secret
11. Name
12. Namespace: le Namespace est un cluser virtuel

# Deployer un cluster Kubernetes

## Dépendances requises

1. Kubectl
2. minikube (pour un seul noeud)
3. cmdr sur Windows (conEmu) ou Terminale sur Linux

## Sur Windows - Vagrant: Deploy 1 cluster on Kubernetes

1. Définir un alias 'k' pour kubectl

```
Set-Alias -Name k -Value kubectl
```

2. Installer 'minikube' pour déployer un seul cluster K8s
3. Définir un alias 'mk' pour Minikube

```
function mk { minikube-windows-amd64 '--show-libmachine-logs' '--alsologtostderr' @args }
```

4. pull and run echo nginx container sur le cluster kubernetes. Image récupérer de google cloud platform

```
k run echo --image=gcr.io/google_containers/echoserver:1.4 --port=8080
```

5. Si le container 'stop' Faire

```
k expose deployment echo --type=NodePort  
NodePort = port of your host
```

6. Commande pour récupérer IP du container 'echo' sur le hôte

```
k get service echo --output='jsonpath="{.spec.ports[0].NodePort}"'  
-> it returns the port of echo container
```

7. Status des pods

```
k get pods
```

8. Commande pour récupérer l'IP du pod

```
mk ip  
-> it returns the IP of pods
```

- Dashboard de Kubernetes is accessible with `mk dashboard`

## Sur Linux : Deployer un multi cluster

Pour commencer, il est plus facile de faire fonctionner correctement K8s sur Linux que sur Windows. J'ai testé Kubernetes sur Vagrant avec Ansible sur Windows mais la rencontre avec les difficultés restant est un sujet pertinent. Exemple, les versions Ansible ou Vagrant présente des bugs, ou encore le port forwarding, ... . En effet, il est mieux d'exploiter Kubernetes sur Linux ou dans le Cloud. == outils sur les cluster . Docker . Kubectl Autres outils importants : . Monitoring

Docker container link: [NetClipper](#)

## **demonstation**