

JEAN-MARC CHORIER



JEAN-BAPTISTE DELPECH

**OOPS ! ON M'A LIVRÉ UN CONTENEUR !
(FOCUS OPS DANS UN ENVIRONNEMENT DEVOPS)**

**TECH
WEEK**



KAIZEN

Quelques citations

Voici la doc pour installer l'application, elle fait environs 20 pages.

Tiens mon image Docker, met la en production !

L'appli ne fonctionne pas en production ? Mais ça marche sur mon poste !

Toute mon infra va devoir être changée !



Introduction

2. Retours d'expérience

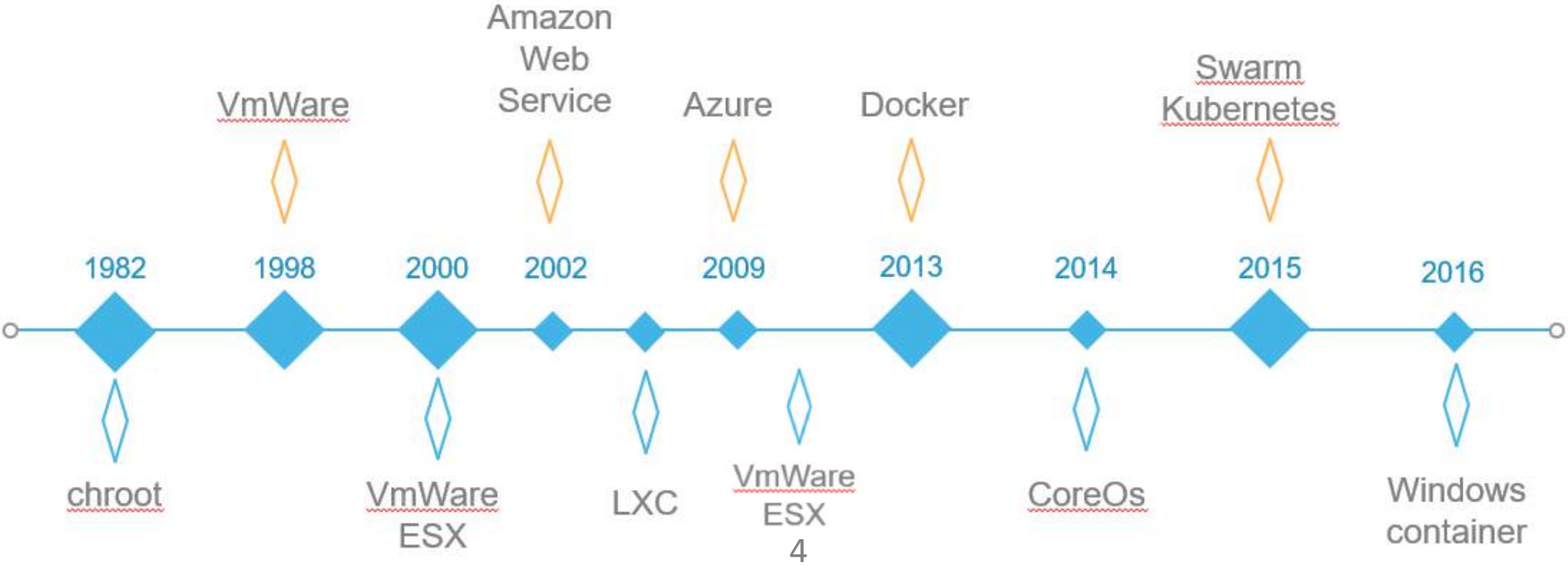
1. Des conteneurs en production

Conclusion

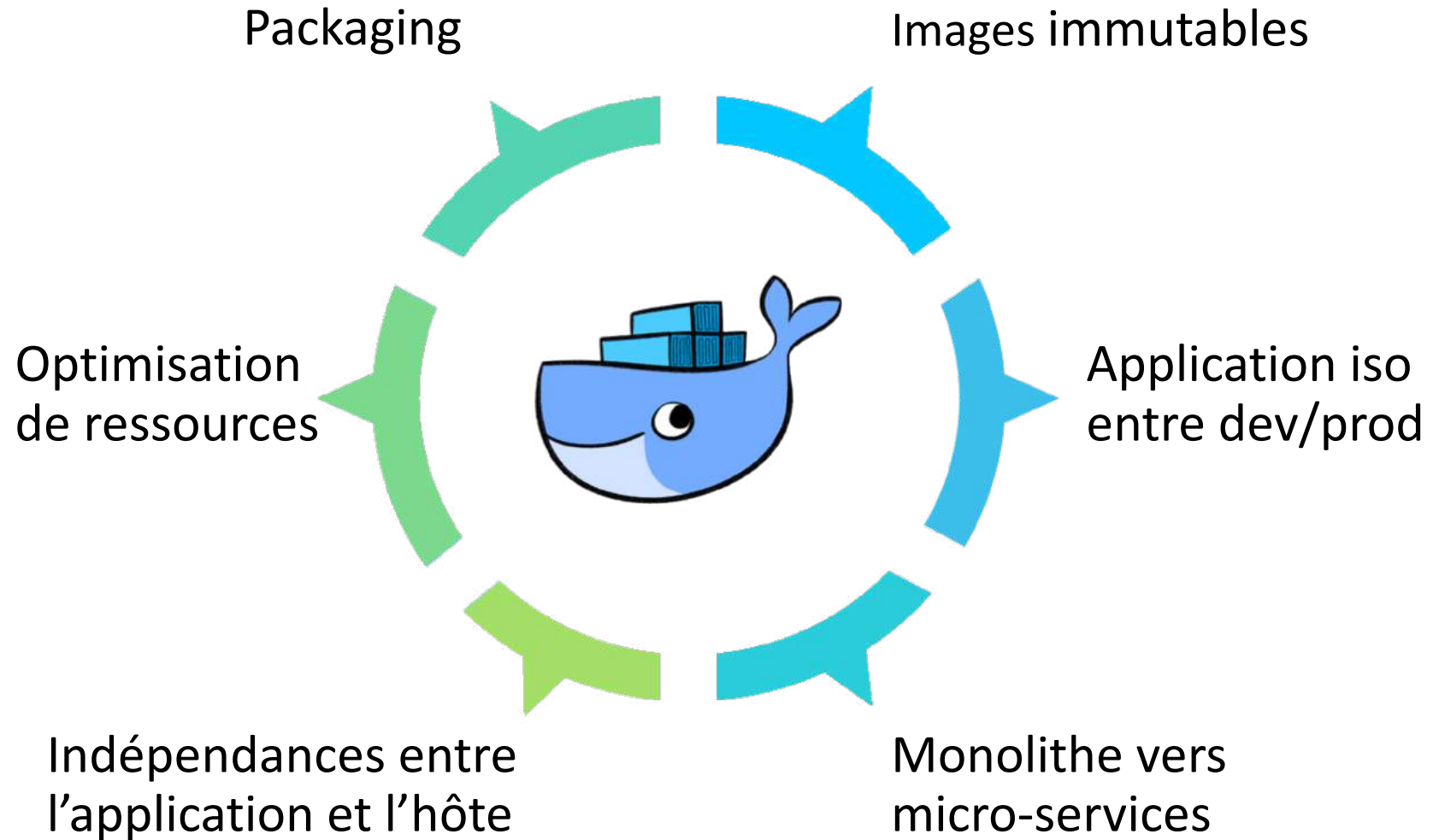


CI/CD
Conteneurs
DevOps Automation
Virtualisation
Hyperviseur
CAMS Docker
Scalabilité Vmware Cloud
Métriques Monitoring
Orchestrateur
Kaizen

Chronologie



Pourquoi Docker (conteneurs) ?



Introduction

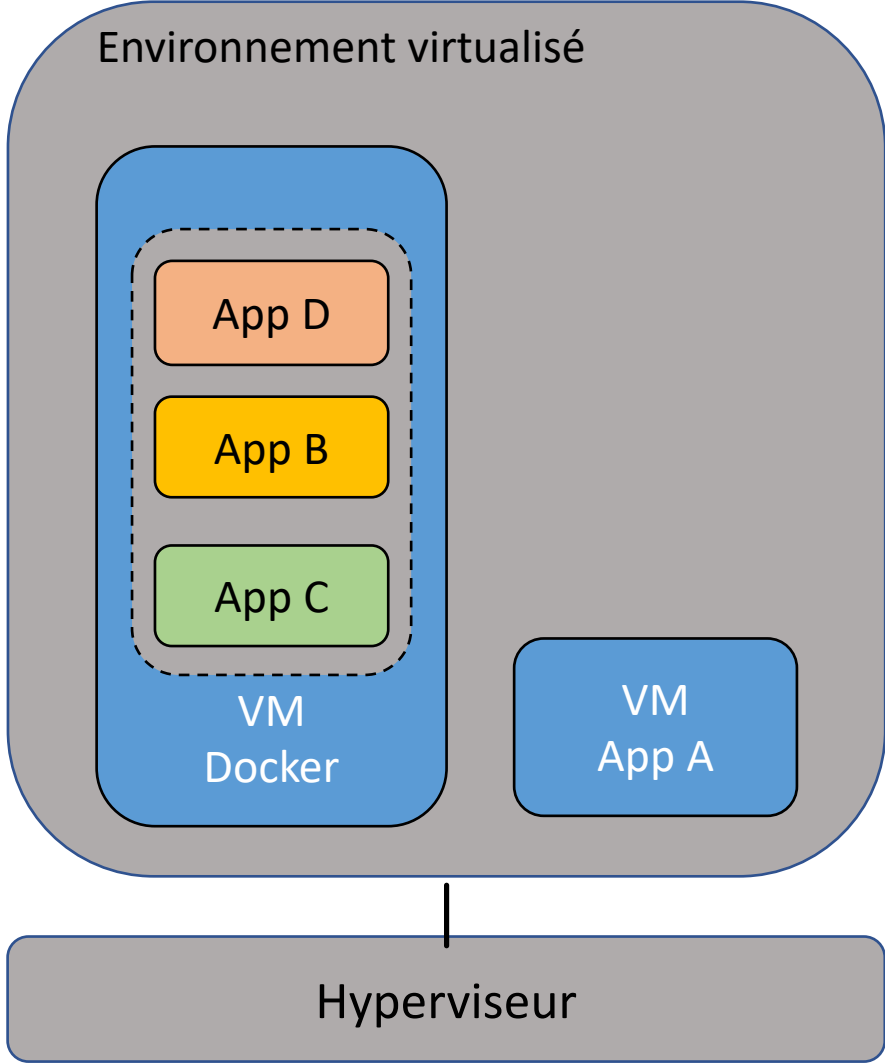
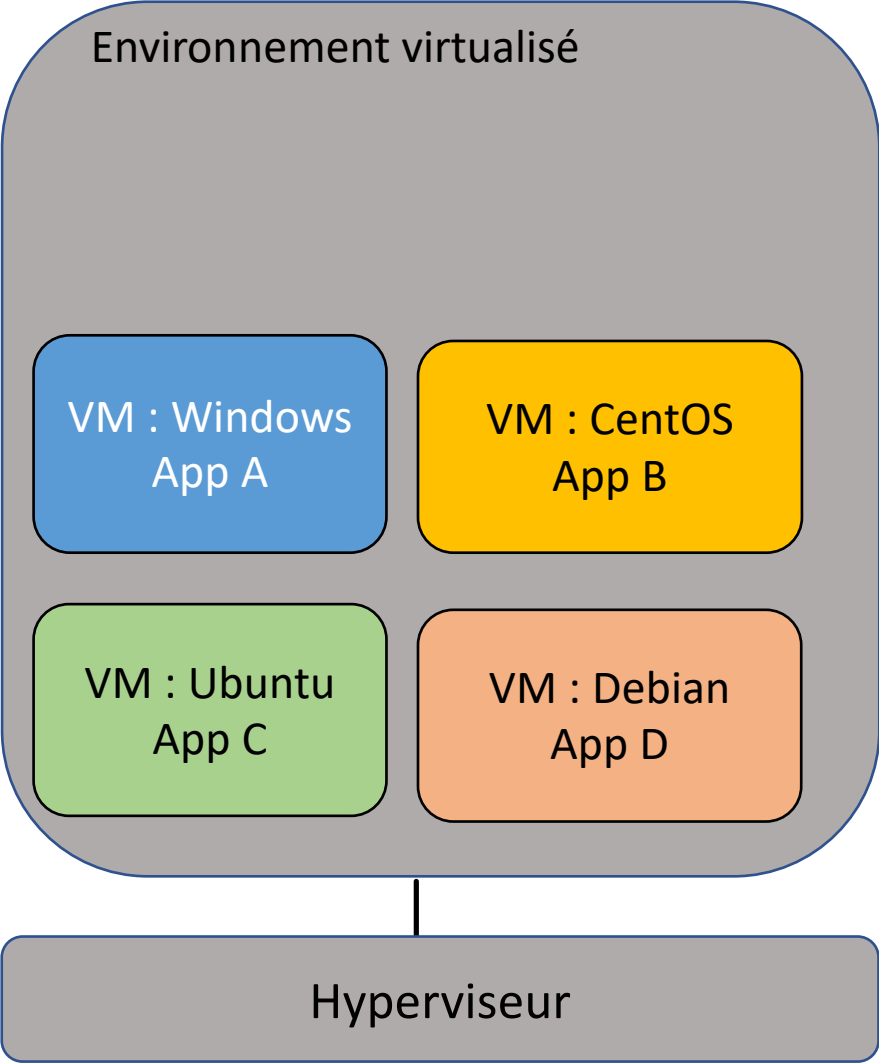
2. Retours d'expérience

1. Des conteneurs en production

Conclusion

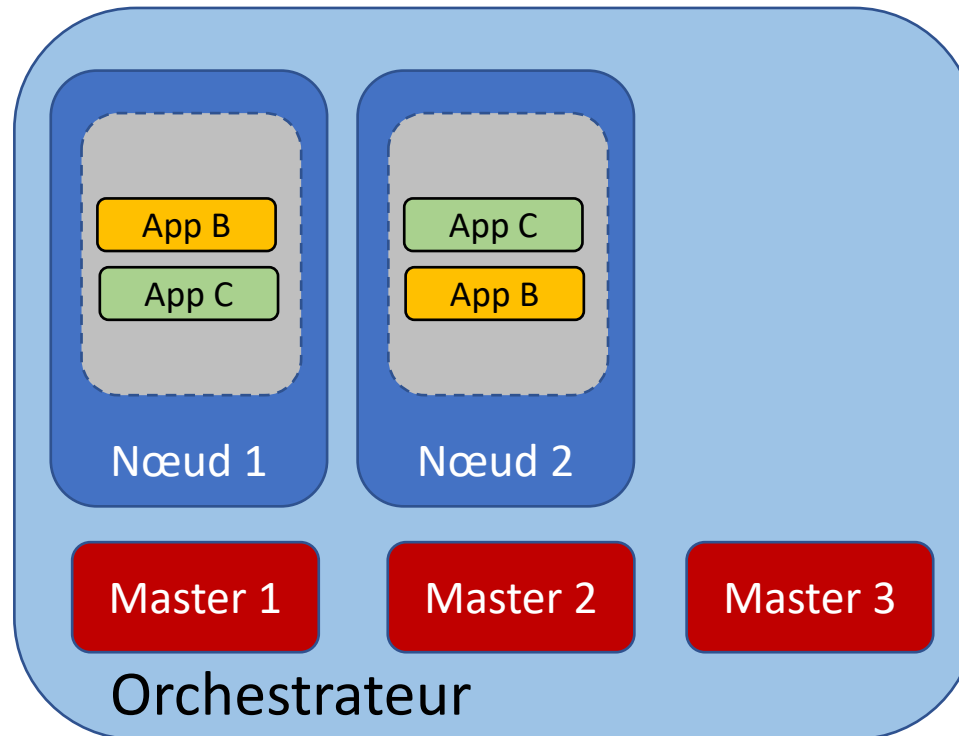


Virtualisation et Docker



L'Orchestrateur

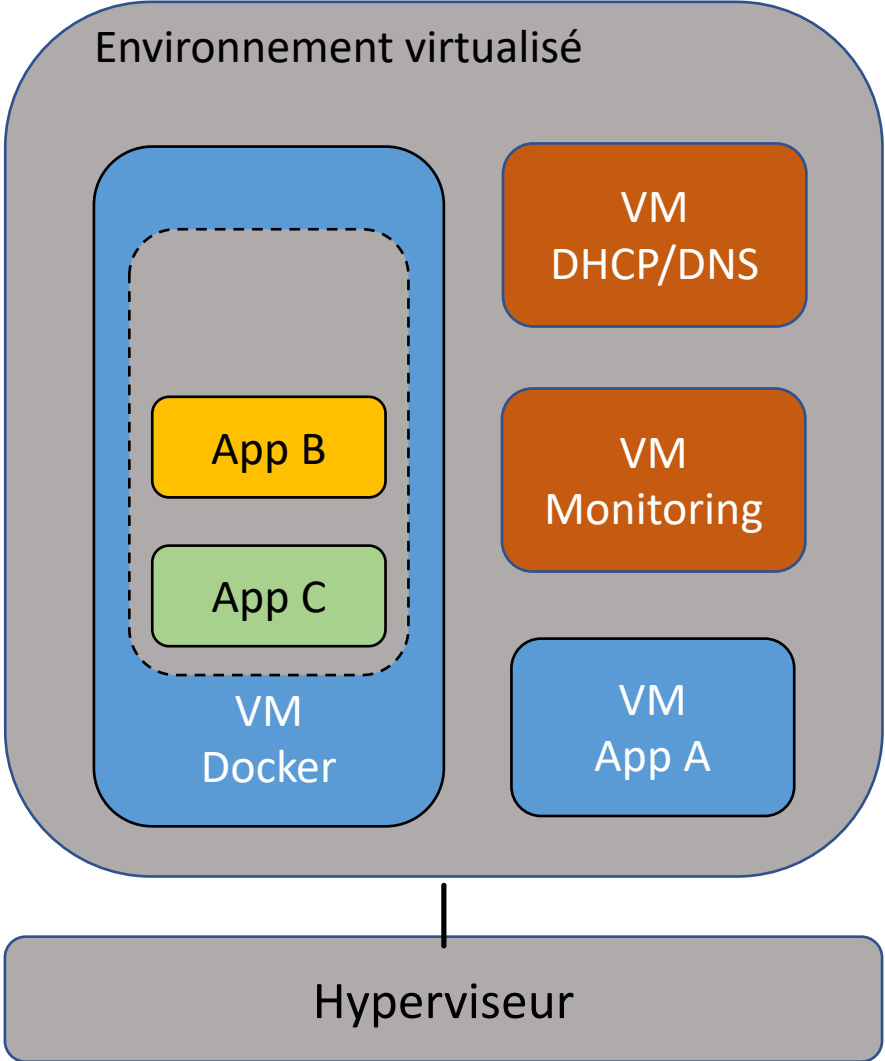
« Faire fonctionner plusieurs serveurs Docker ensemble. »



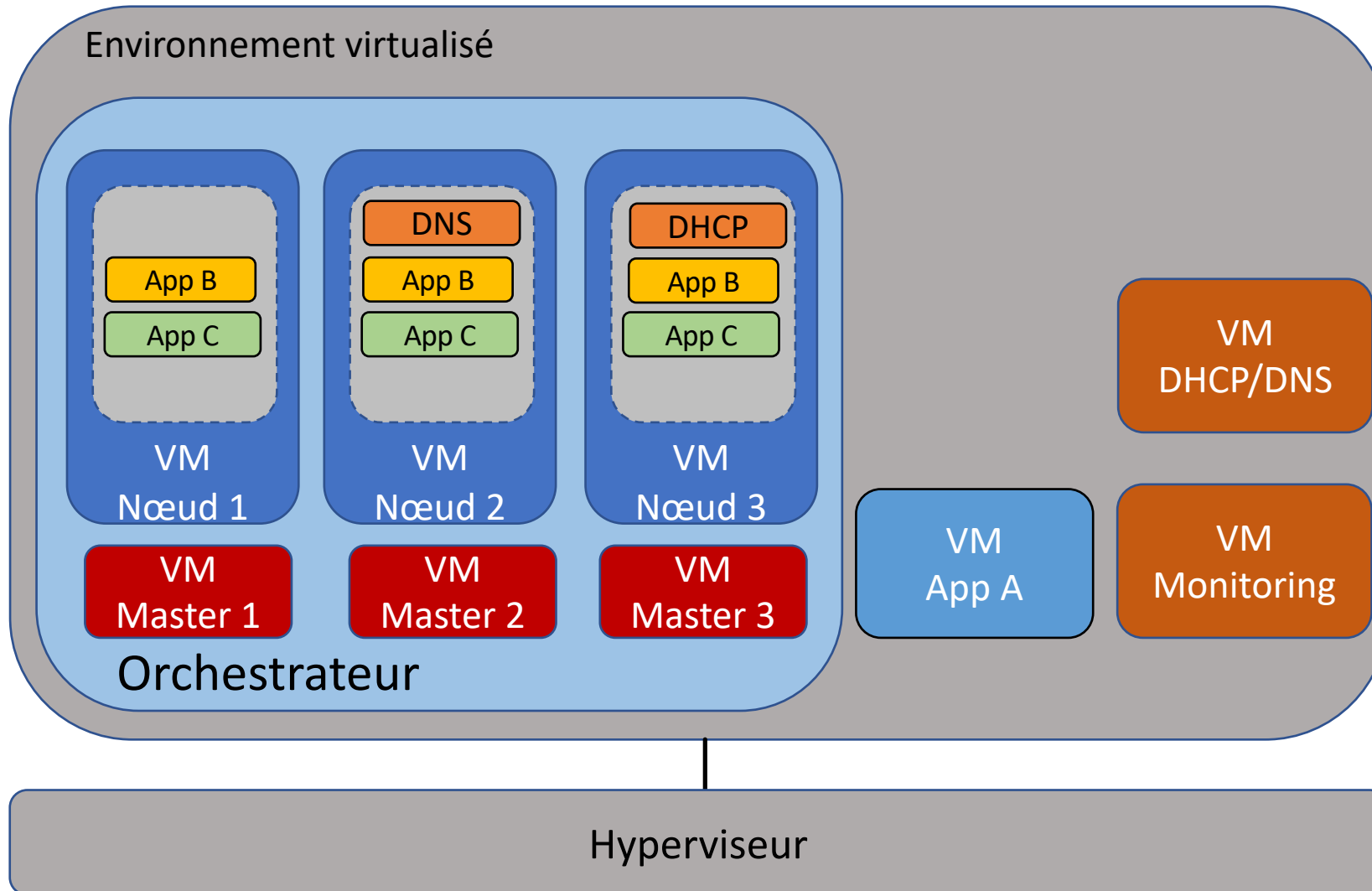
Nœud = Worker

Master = Chef d'orchestre du cluster

Virtualisation et Docker



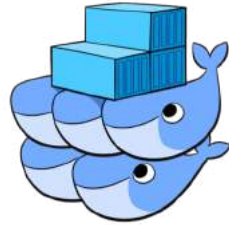
Virtualisation et Orchestration



Synthèse

- Migrer sans changer fondamentalement son infrastructure
- VM moins puissantes & mieux dimensionnées
- Mutualise le système d'exploitation
- Maintenance
 - 1 seul système à maintenir
 - Configuration du système unique (ansible, template VM, packer, etc)
- Déployer nos applications de la même façon dans nos clusters

Quelques orchestrateurs



- Docker Swarm



RANCHER

- Rancher



- Kubernetes



- Apache Mesos



- Nomad

Cycle de vie

Application / Environnement

Application (Image Docker)

V1.0.0

V1.1.0

V2.0.0

V2.0.1

Environnement (stack.yml)

V1.0.0

V1.1.0

V1.1.1



Introduction

1. Des conteneurs en production

2. Retours d'expérience

Conclusion



Docker Swarm

Collaborate

Build

Test

Deploy

Run

Application Lifecycle Mgmt.



SCM/VCS



Testing



Deployment



Cloud / IaaS / PaaS



Communication & ChatOps



CI



Config Mgmt. / Provisioning



Orchestration & Scheduling



Knowledge Sharing



Build



Database Management



Artefact Management



BI / Monitoring / Logging

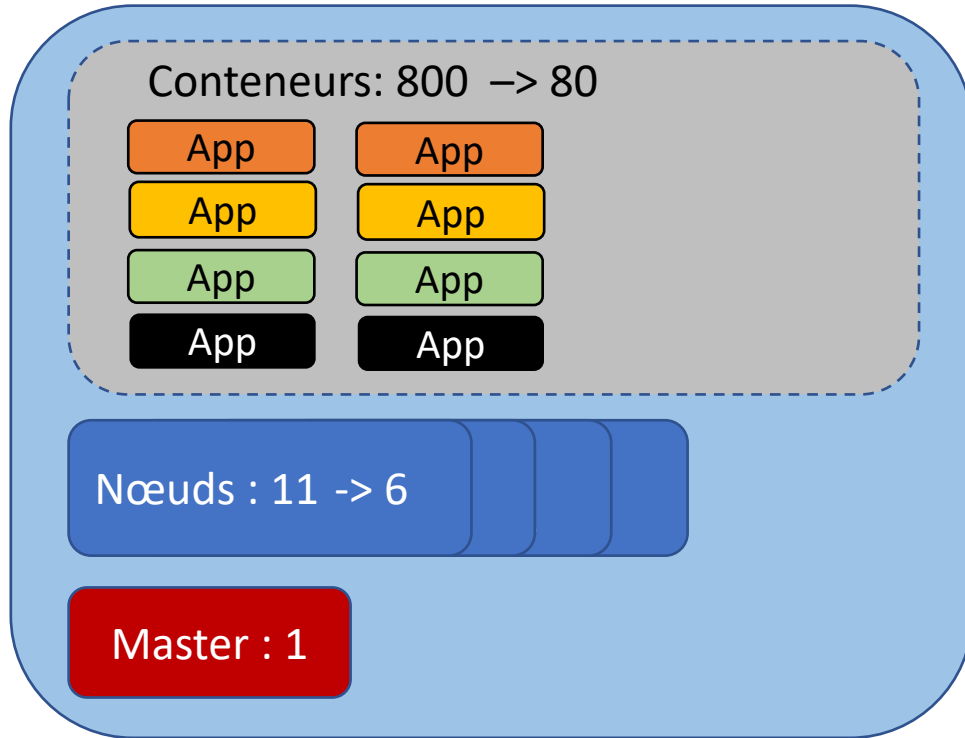


Kubernetes

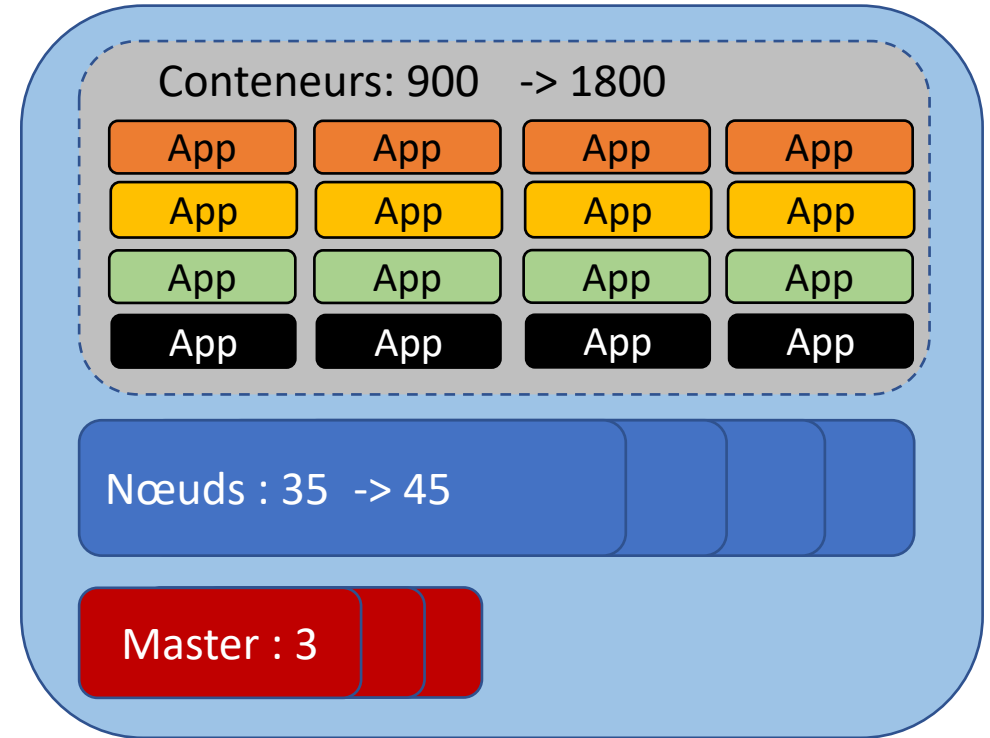


Environnements

Cluster Kubernetes Dev

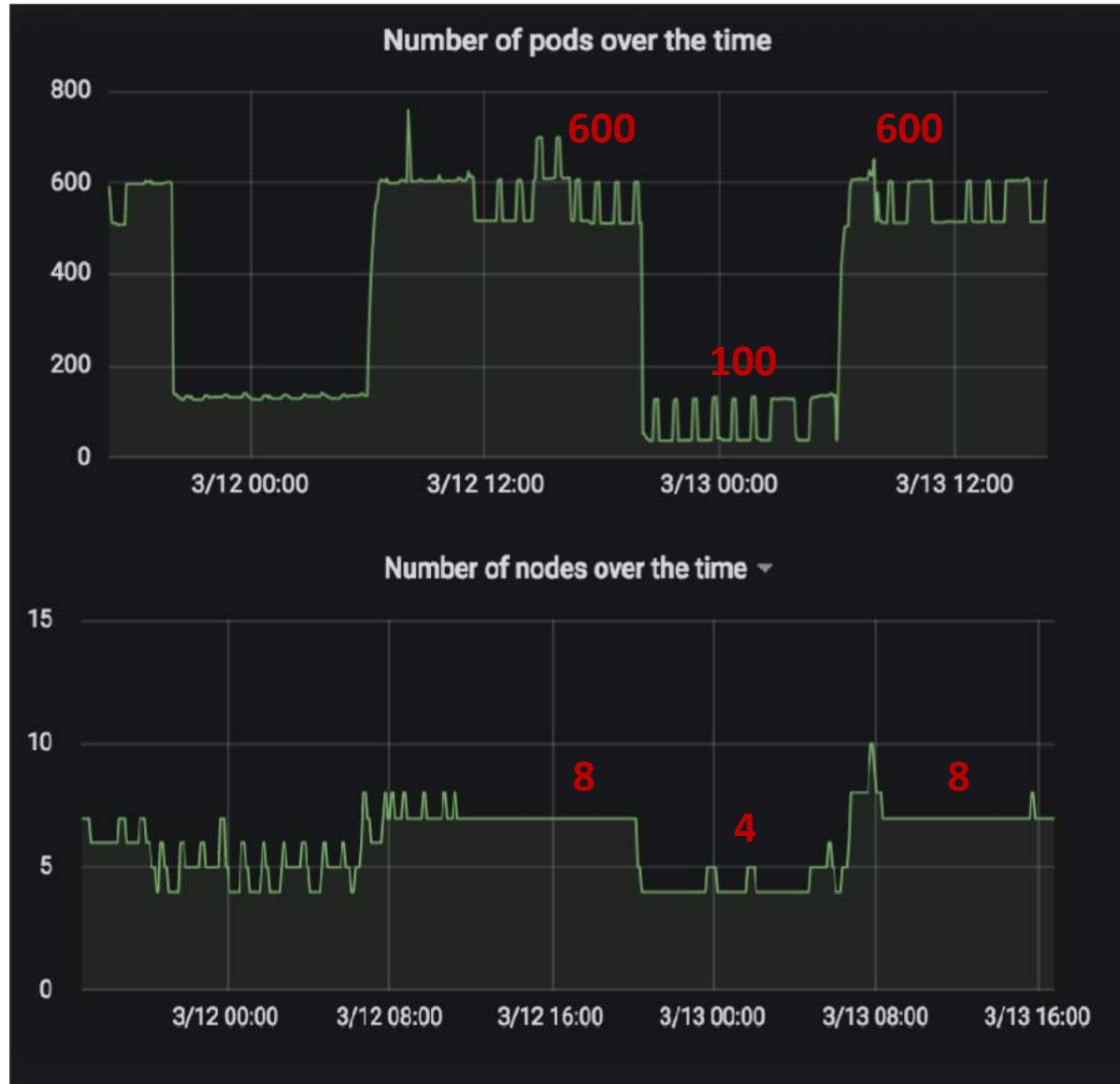


Cluster Kubernetes Prod



2M requêtes / jour

Monitoring : Prometheus / Grafana

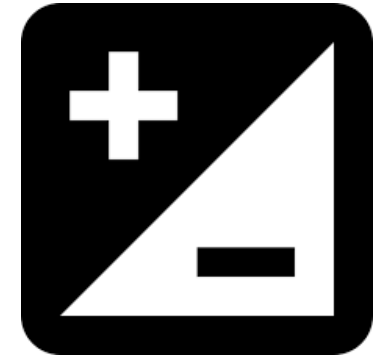


Fonctionne bien



Auto réparation

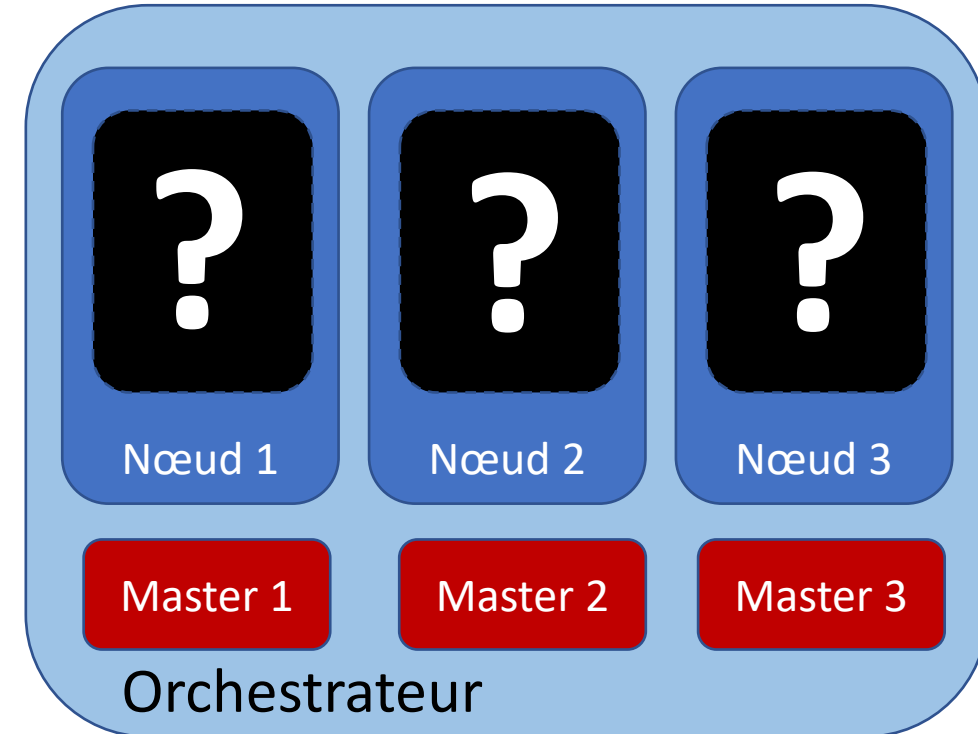
Autoscaling des noeuds



Helm

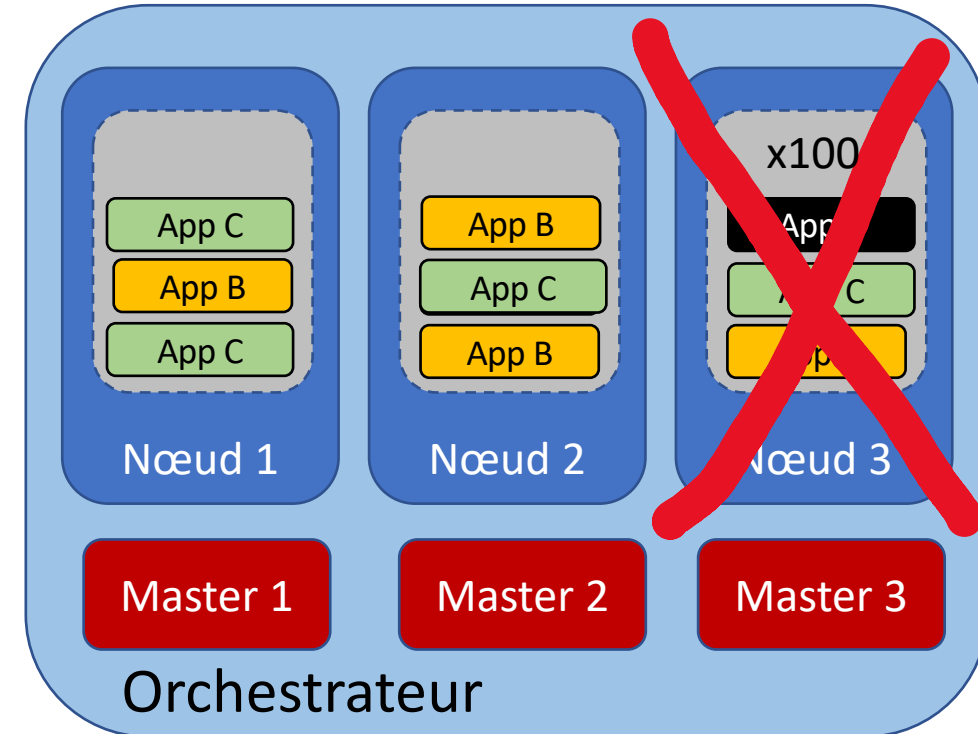
Points de vigilance

- Auto réparation ?!
 - Monitoring (prometheus / grafana / alertManager)
 - Logging (ELK)



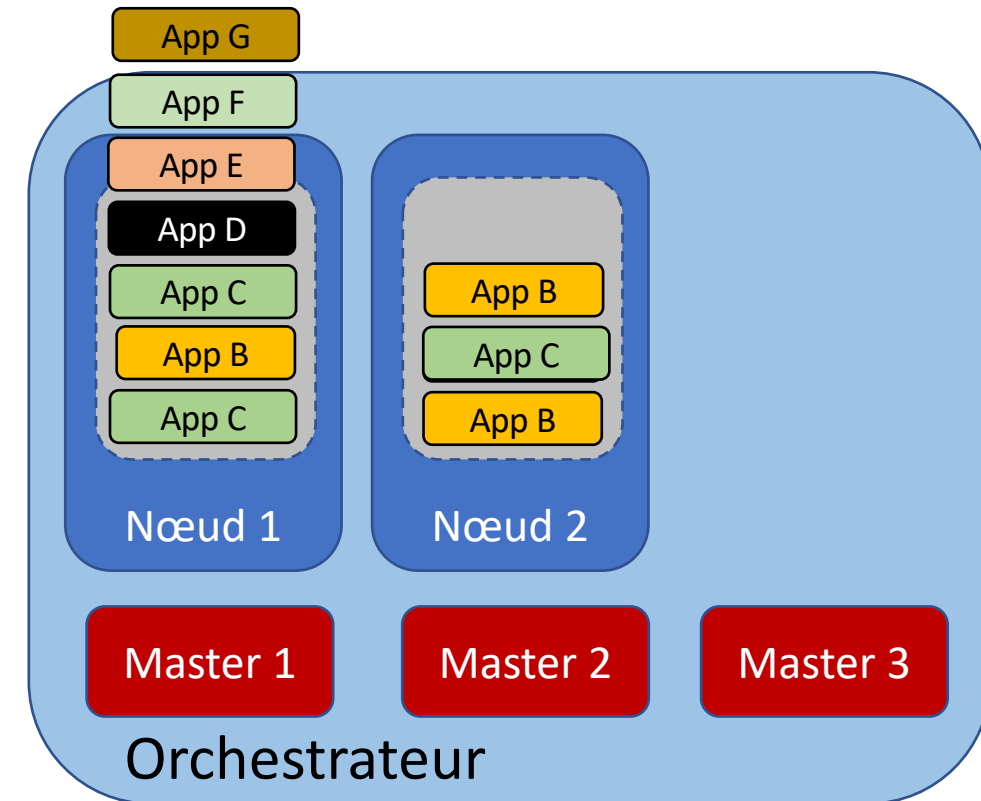
Points de vigilance

- Auto réparation ?!
 - Monitoring (prometheus / grafana / alertManager)
 - Logging (ELK)
- Taille des images docker
 - 500-800Mo
 - 100 x 500Mo = 50Go
 - Suivez les bonnes pratiques Docker



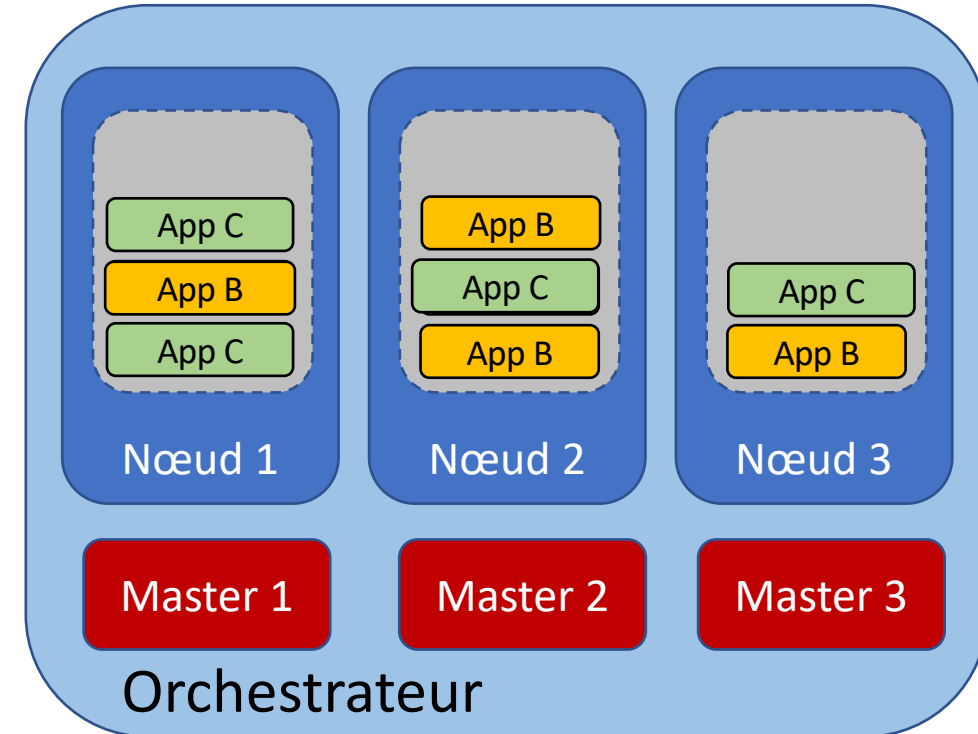
Points de vigilance

- Auto réparation ?!
 - Monitoring (prometheus / grafana / alertManager)
 - Logging (ELK)
- Taille des images docker
 - 500-800Mo
 - 100 x 500Mo = 50Go
 - Suivez les bonnes pratiques Docker
- Comment placer les conteneurs ?
 - Réservation de ressources (resources requests/limit)



Points de vigilance

- Auto réparation ?!
 - Monitoring (prometheus / grafana / alertManager)
 - Logging (ELK)
- Taille des images docker
 - 500-800Mo
 - 100 x 500Mo = 50Go
 - Suivez les bonnes pratiques Docker
- Comment placer les conteneurs ?
 - Réserveation de ressources (resources requests/limit)
 - Répartition des conteneurs (affinity / toleration)



Points de vigilance

- Auto réparation ?!
 - Monitoring (prometheus / grafana / alertManager)
 - Logging (ELK)
- Taille des images docker
 - 500-800Mo
 - 100 x 500Mo = 50Go
 - Suivez les bonnes pratiques Docker
- Comment placer les conteneurs ?
 - Réserveation de ressources (resources requests/limit)
 - Répartition des conteneurs (affinity / toleration)
- Vérifier l'état des conteneurs
 - Readiness probe
 - Liveness probe

Introduction

2. Retours d'expérience

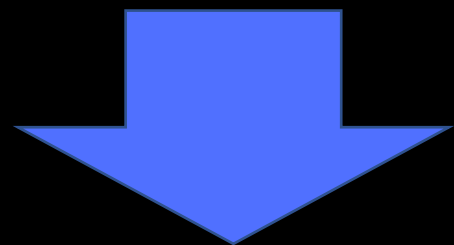
1. Des conteneurs en production

Conclusion



- ✓ Utilisation de l'infrastructure existante
- ✓ Tous les outils existent
- ✓ Allez-y étape par étape
- ✓ Pas de refonte globale applicative
- ✓ Accompagnement par des experts ;)

Libérez du temps pour vous focaliser sur
votre métier et votre application !



Réduction du Time To Market

DES QUESTIONS ?

Pôle Devops @ KAIZEN
pole.devops@kaizen-solutions.net

--

www.kaizen-solutions.net

--

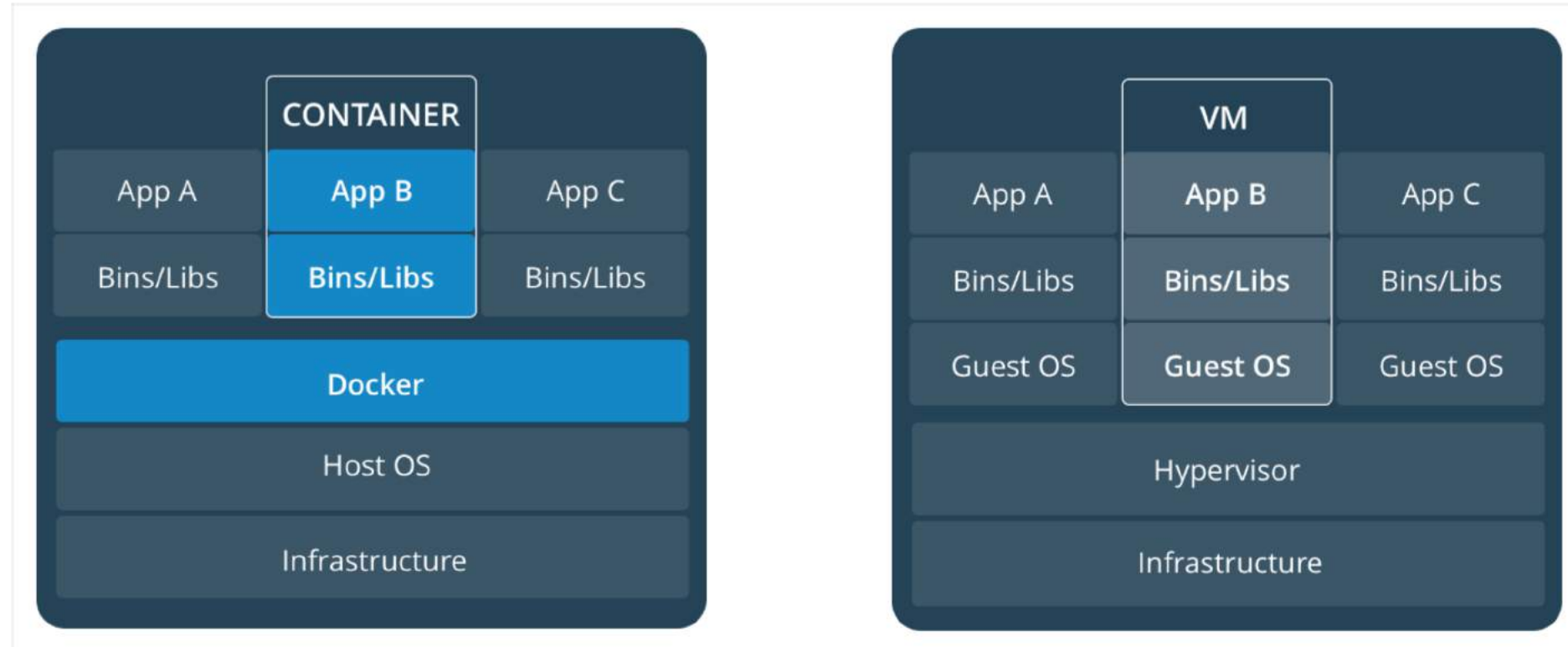
contact@kaizen-solutions.net

TECH
WEEK



KAIZEN

Containers vs Virtualisation



Docker :

- Utilise le noyau Linux
- Est basé sur les cgroup (isolation des ressources)
- Etend le format LXC (Linux Container) avec des capacités de plus haut niveau

Points de vigilance

- Auto réparation ?!
 - Monitoring (prometheus / grafana / alertManager)
 - Logging (ELK)
- Taille des images docker
 - 500-800Mb
 - Respect bonnes pratiques Docker : divisé par 5 voir 10
- Ordonnancement des conteneurs
 - Ressources utilisées (resources requests/limit)
 - Répartition des conteneurs (affinity / toleration)
- Vérifier l'état des conteneurs
 - Readiness probe
 - Liveness probe

