

# Orchestration et automatisation

ESIR 3 – 2021/2022

Djob Mvondo

# Déploiement ???

Automatisation

Choix des nœuds

En continu

Surveiller

Automatiser l'installation des dépendances, des binaires, et le paramétrage de l'application/module.

# Déploiement ???

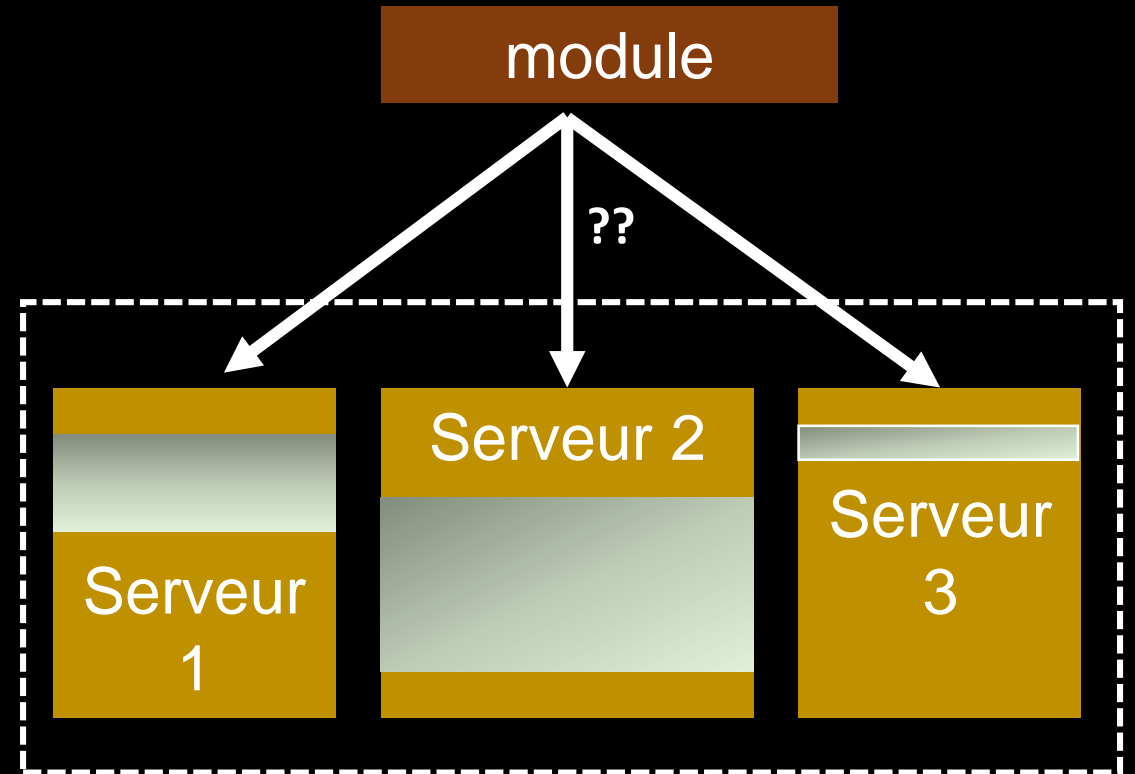
Automatisation

Choix des nœuds

En continu

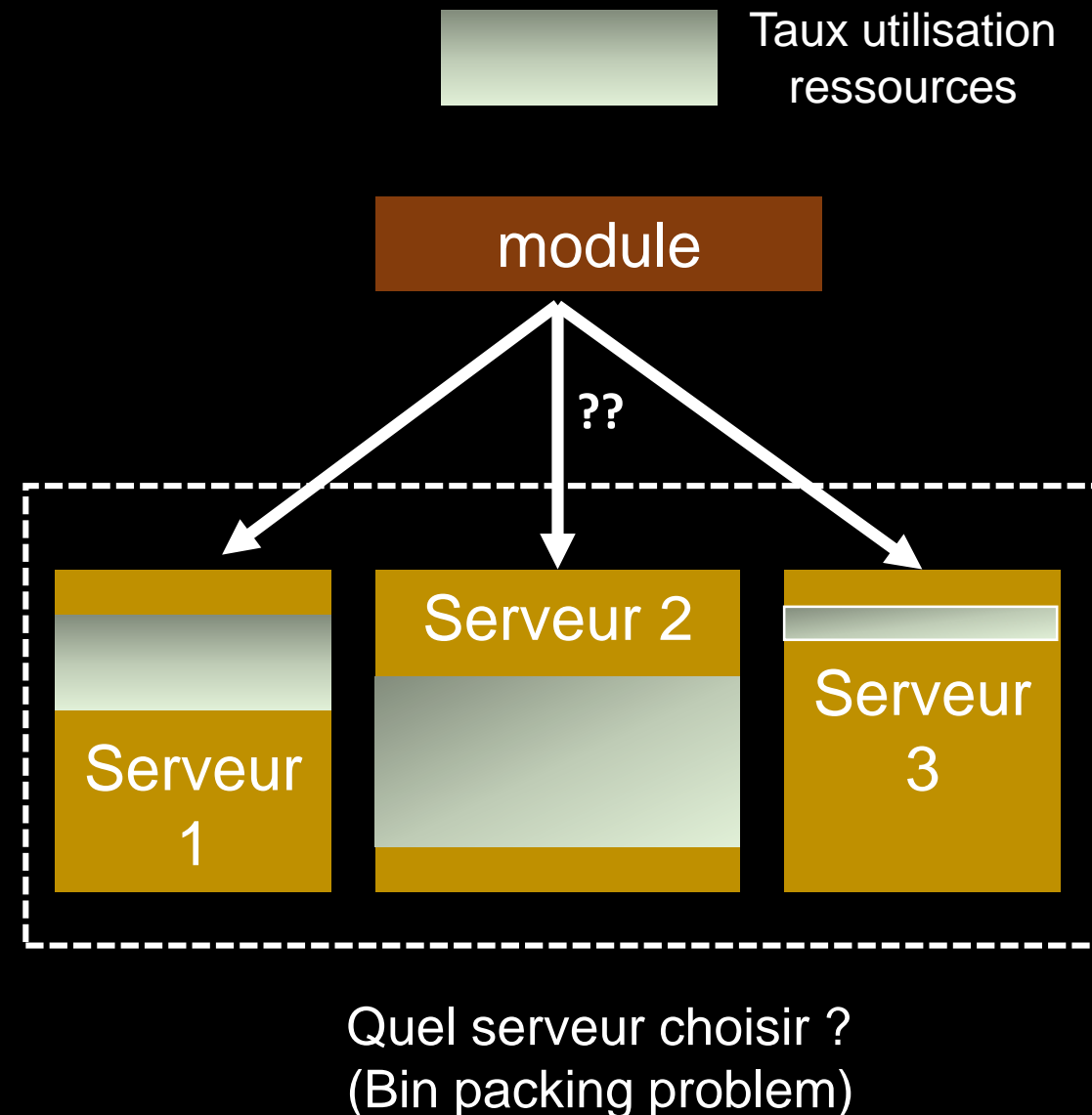
Surveiller

 Taux utilisation  
ressources



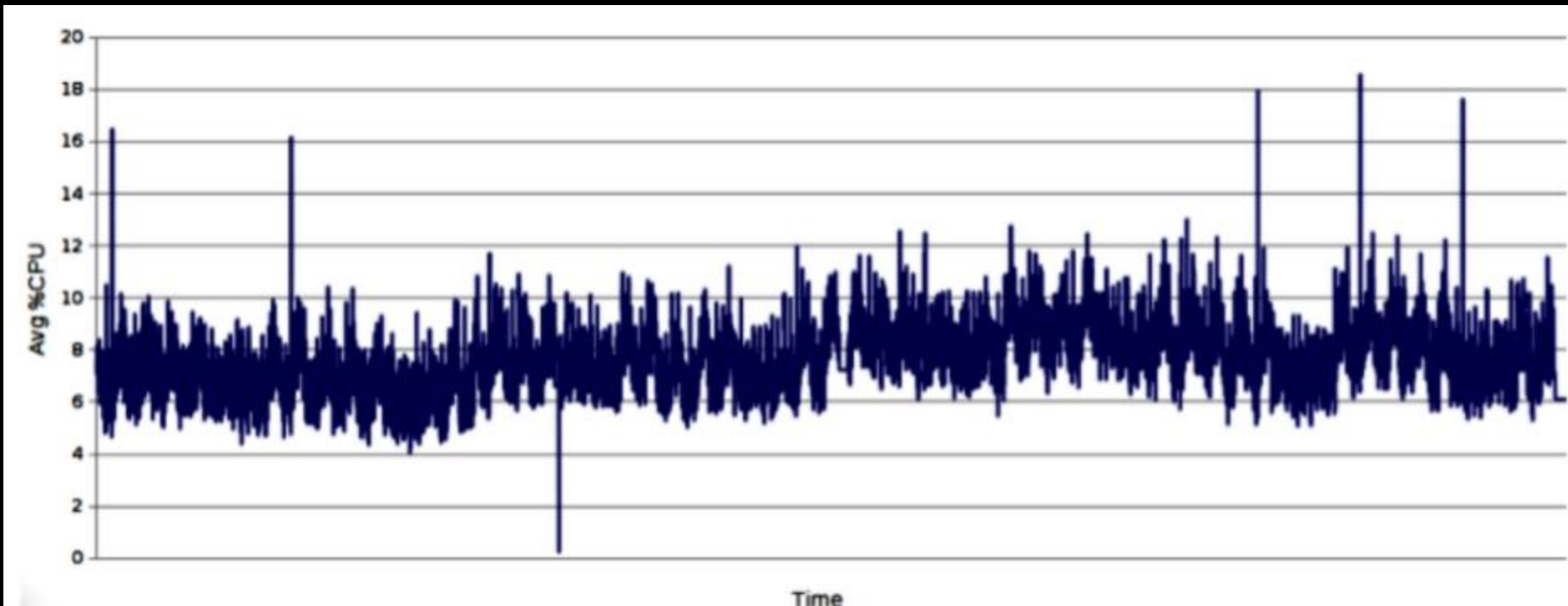
# Choix des nœuds = Consolidation

- ❑ Placer le module en fonction des quantités de mémoire et CPU
- ❑ Optimiser les ressources disponible (éviter les **pitholes**)
- ❑ Les unités d'isolations n'utilisent pas forcément toutes les ressources



# Choix des nœuds = Consolidation

EOLAS CLOUD --- **mean CPU usage of 10%** ---  
**4 months observations**



# Choix des nœuds = Consolidation

## Providing SLOs for Resource-Harvesting VMs in Cloud Platforms – OSDI'20

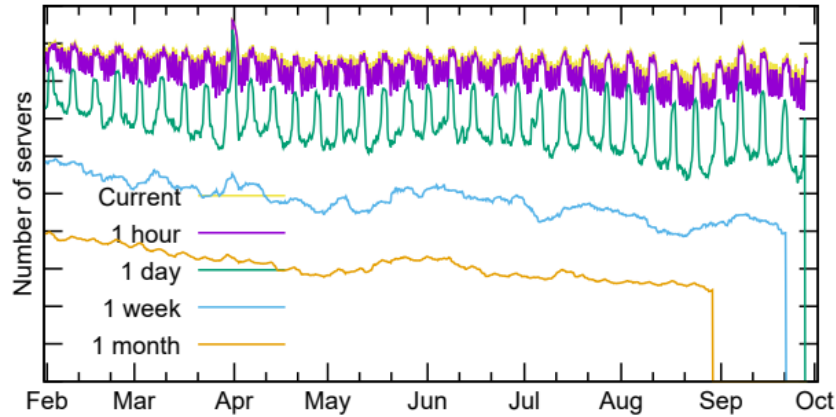


Figure 3: #servers with 1 unallocated core in a region.

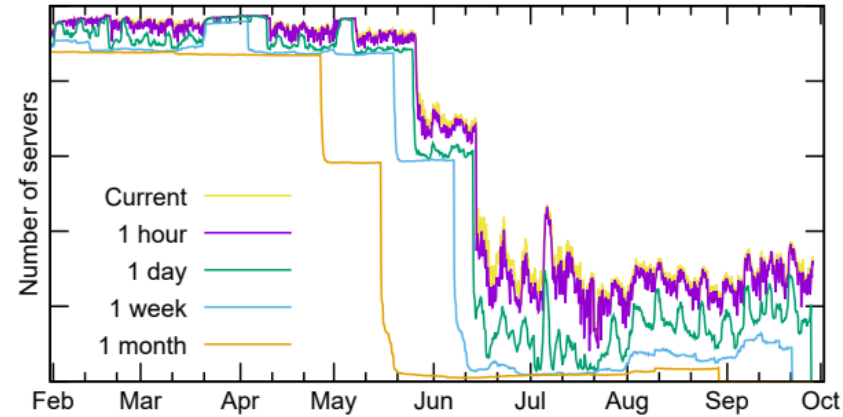


Figure 5: #servers with 1 unallocated core in a cluster.

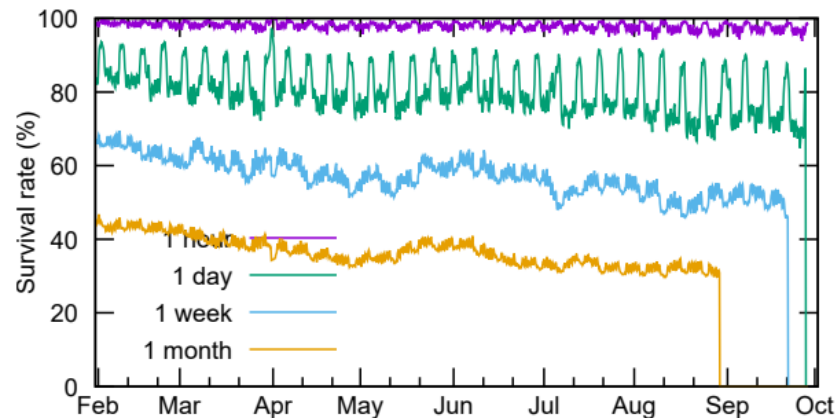


Figure 4: Survival rate with 1 unallocated core in a region.

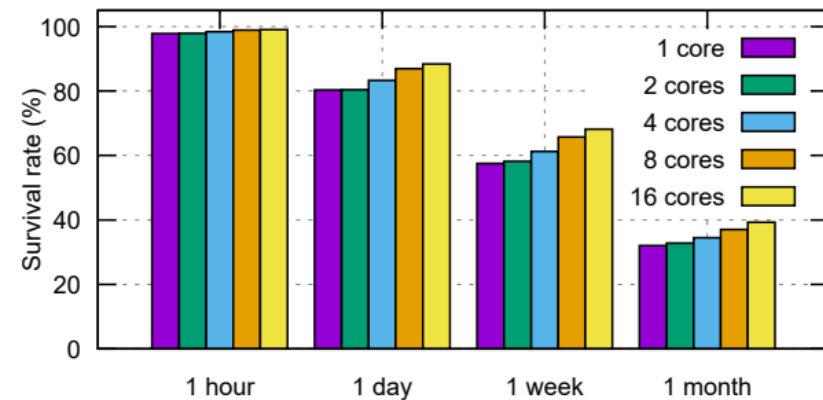
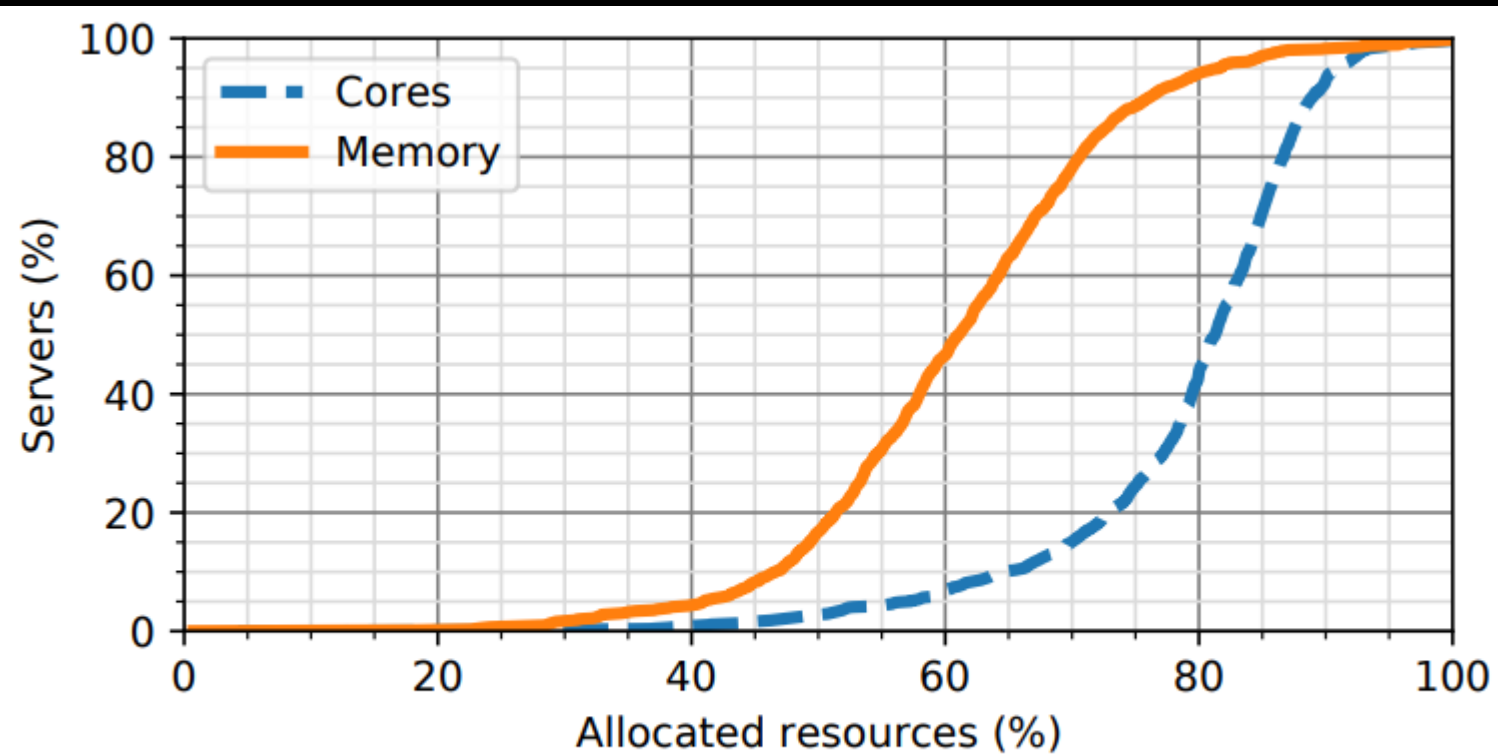


Figure 6: Survival rate of deployable evictable VMs as a function of lifetime and minimum size.

# Choix des nœuds = Consolidation

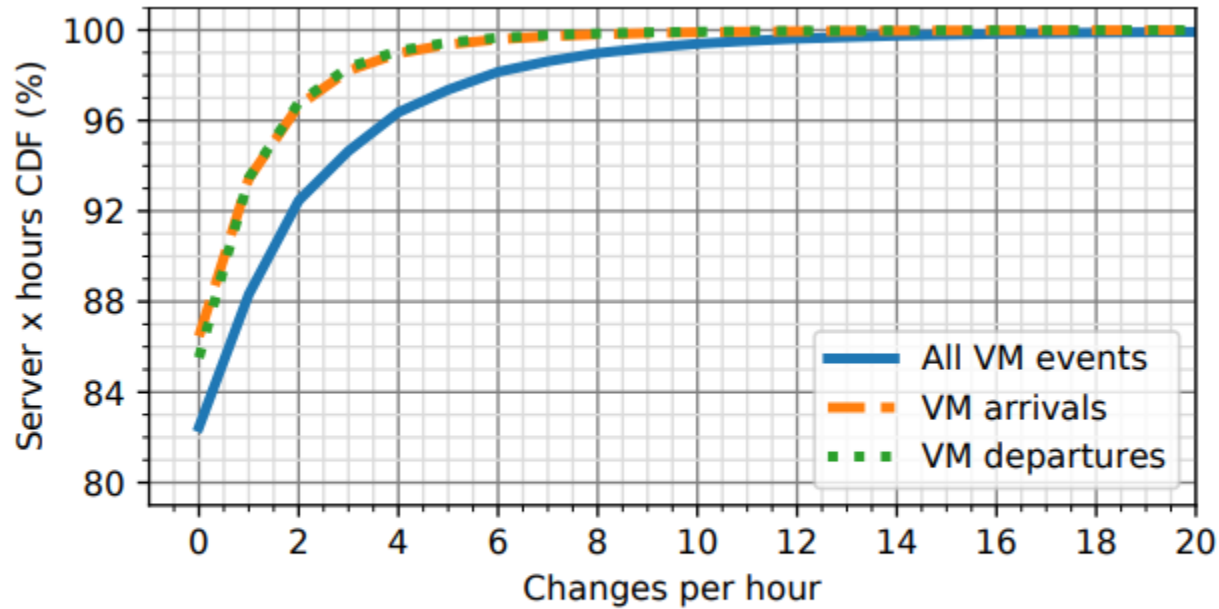
Memory-Harvesting VMs in Cloud Platforms– **ASPLOS'22**



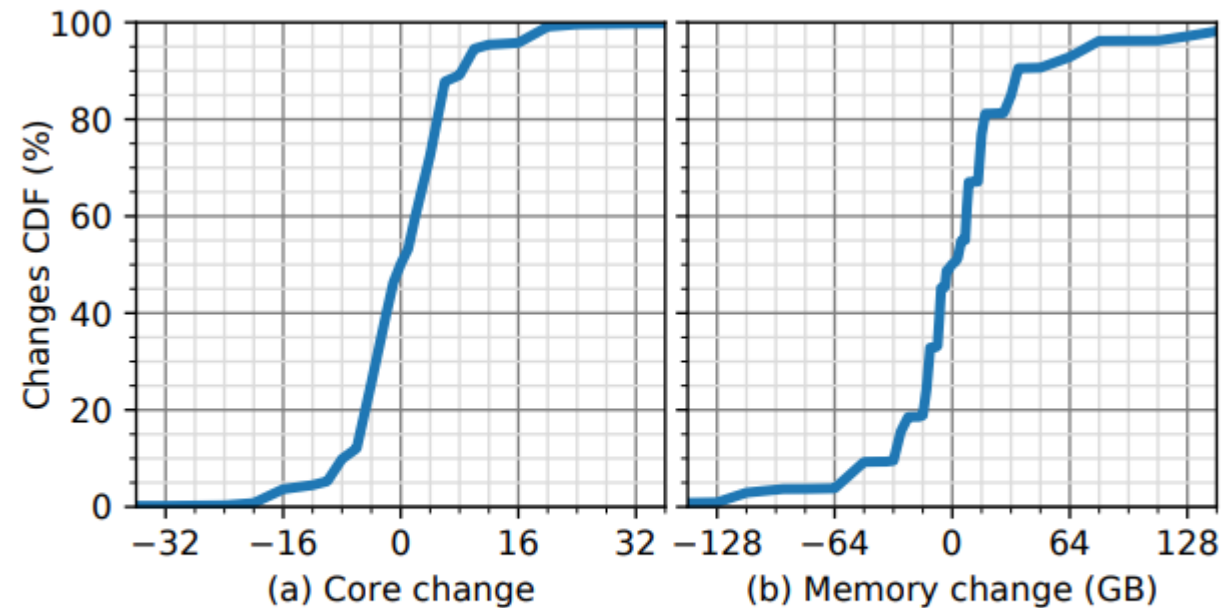
**Figure 1: Allocated resources per server in increasing order. There is room for harvesting both cores and memory.**

# Choix des nœuds = Consolidation

## Memory-Harvesting VMs in Cloud Platforms– ASPLOS'22



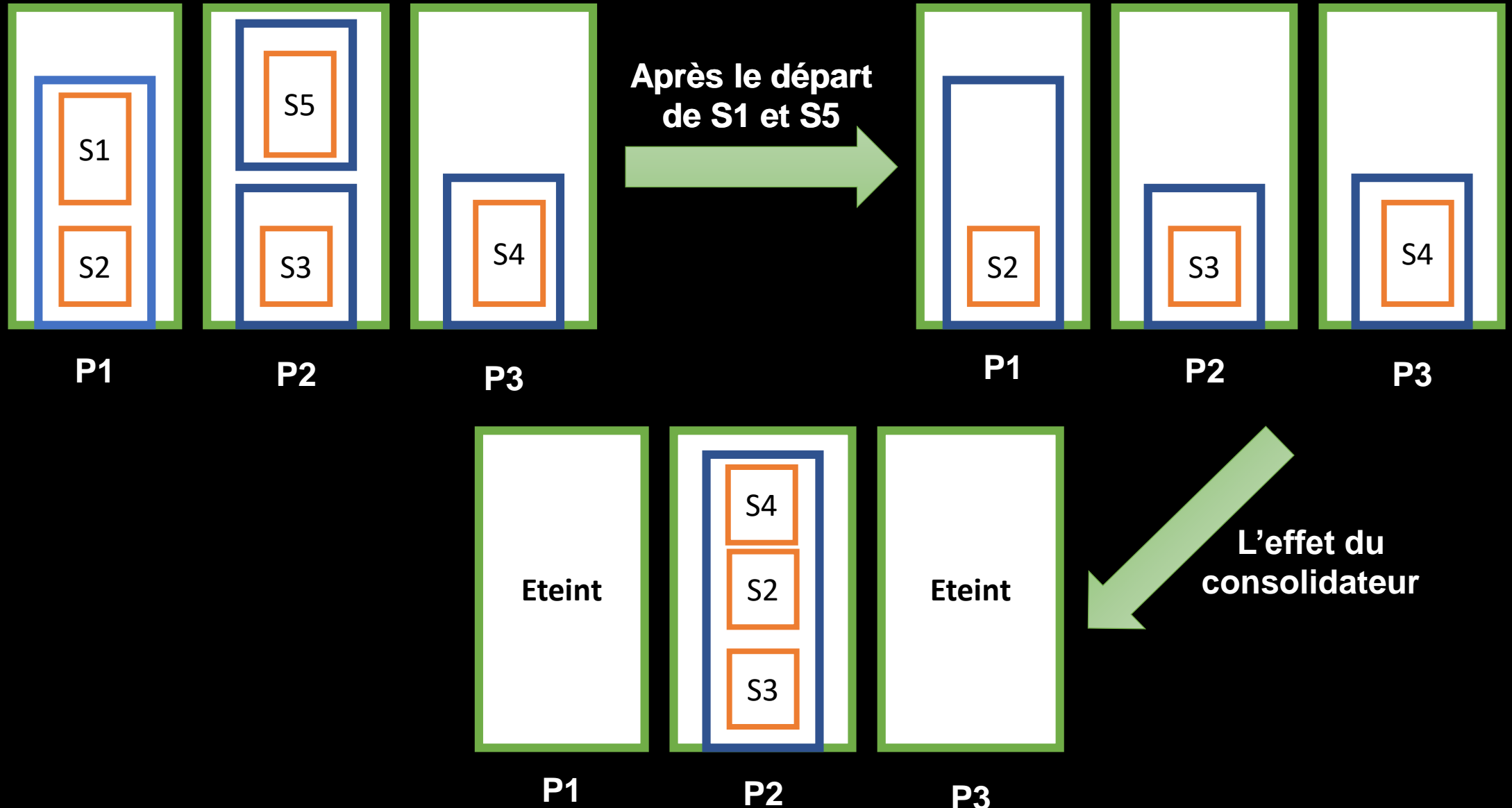
**Figure 2: Resource changes per hour at each server over all hours. 82.5% of the hours involve no VM events, and in 86.5% of them there are no VM arrivals.**



**Figure 3: Core and memory allocation changes. Positive values mean VM arrivals. 90% of the VM creations involve 8 cores or fewer and 32 GB of memory or less.**



# Rôle du consolidateur



# Rôle du consolidateur (NP-complet)

Minimise le nombre de machines physiques

Eteint les machines physiques

Exploite la migration à chaud (faut minimiser l'impact)

# Rôle du consolidateur (NP-complet)

Exploiter la mémoire non-utilisée

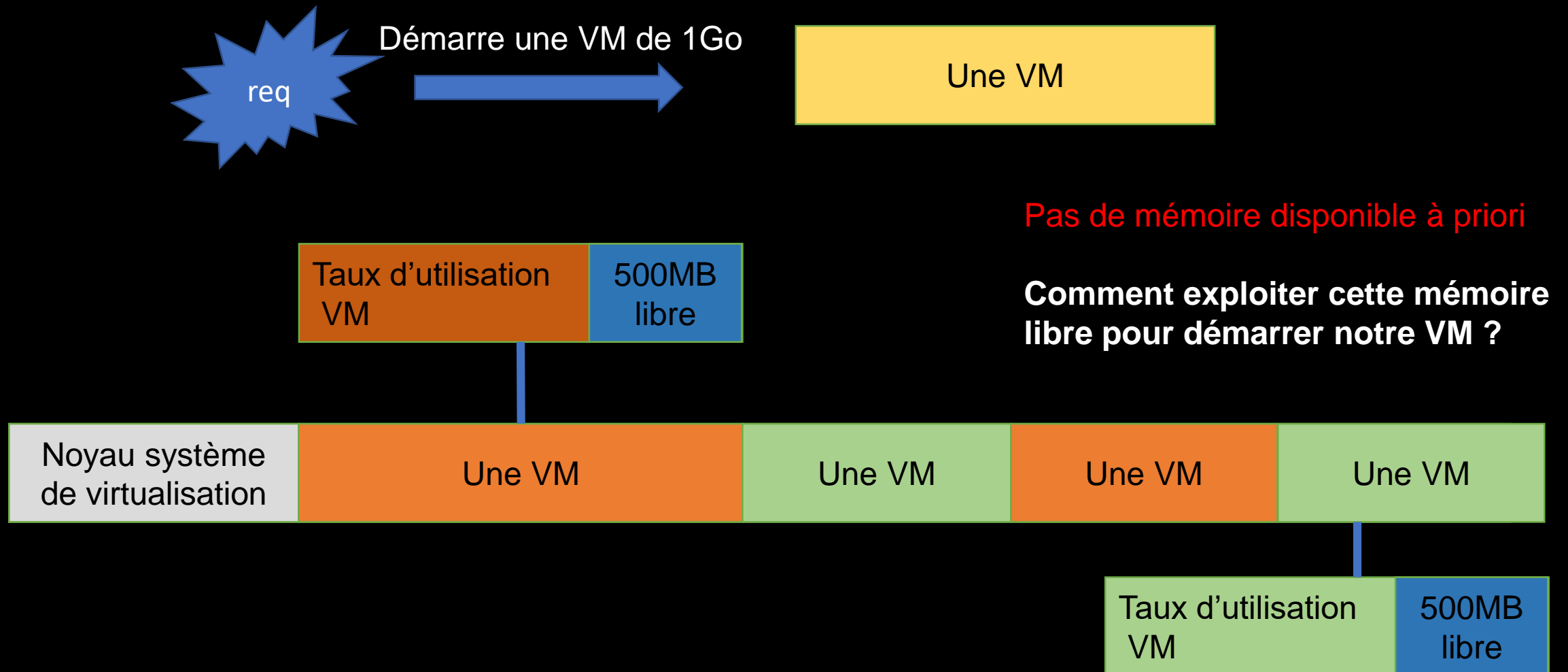


Pas de mémoire disponible à priori



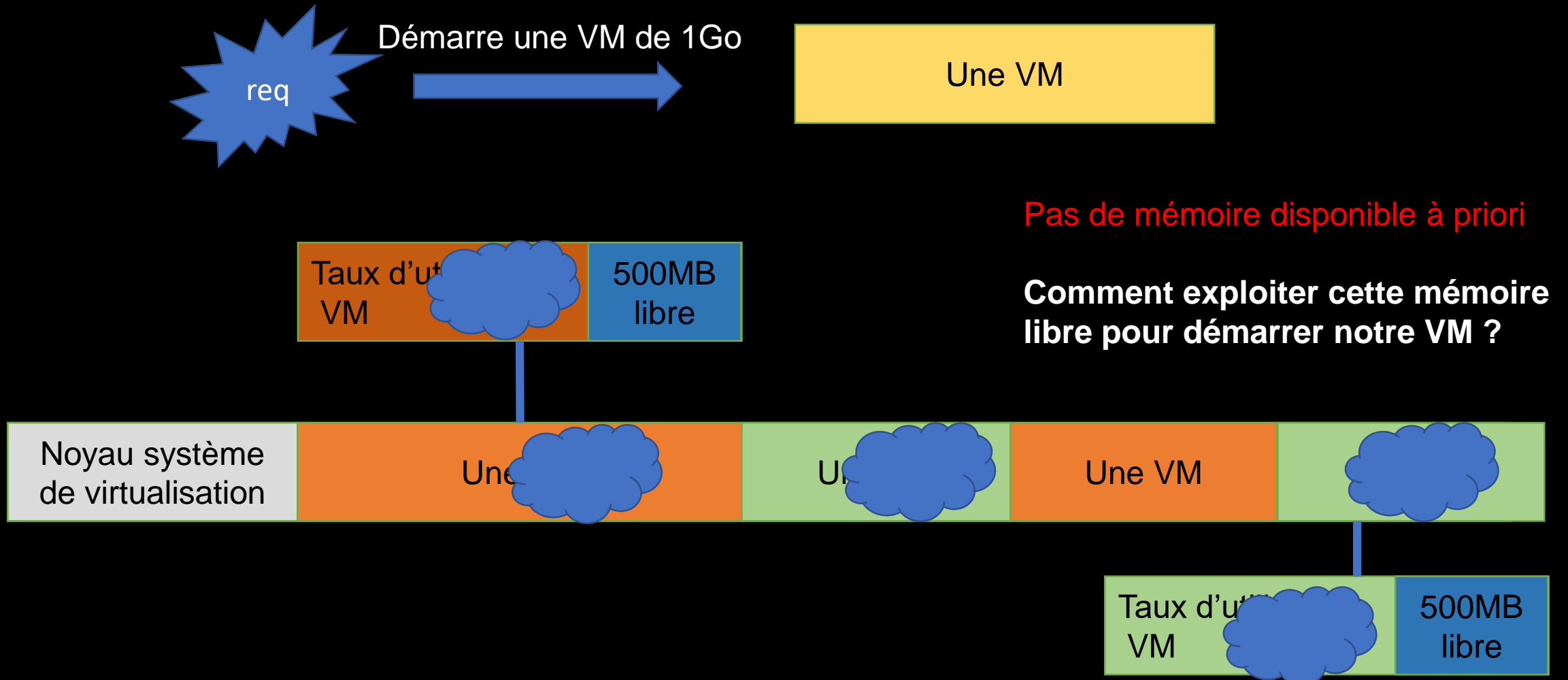
# Rôle du consolidateur (NP-complet)

Exploiter la mémoire non-utilisée



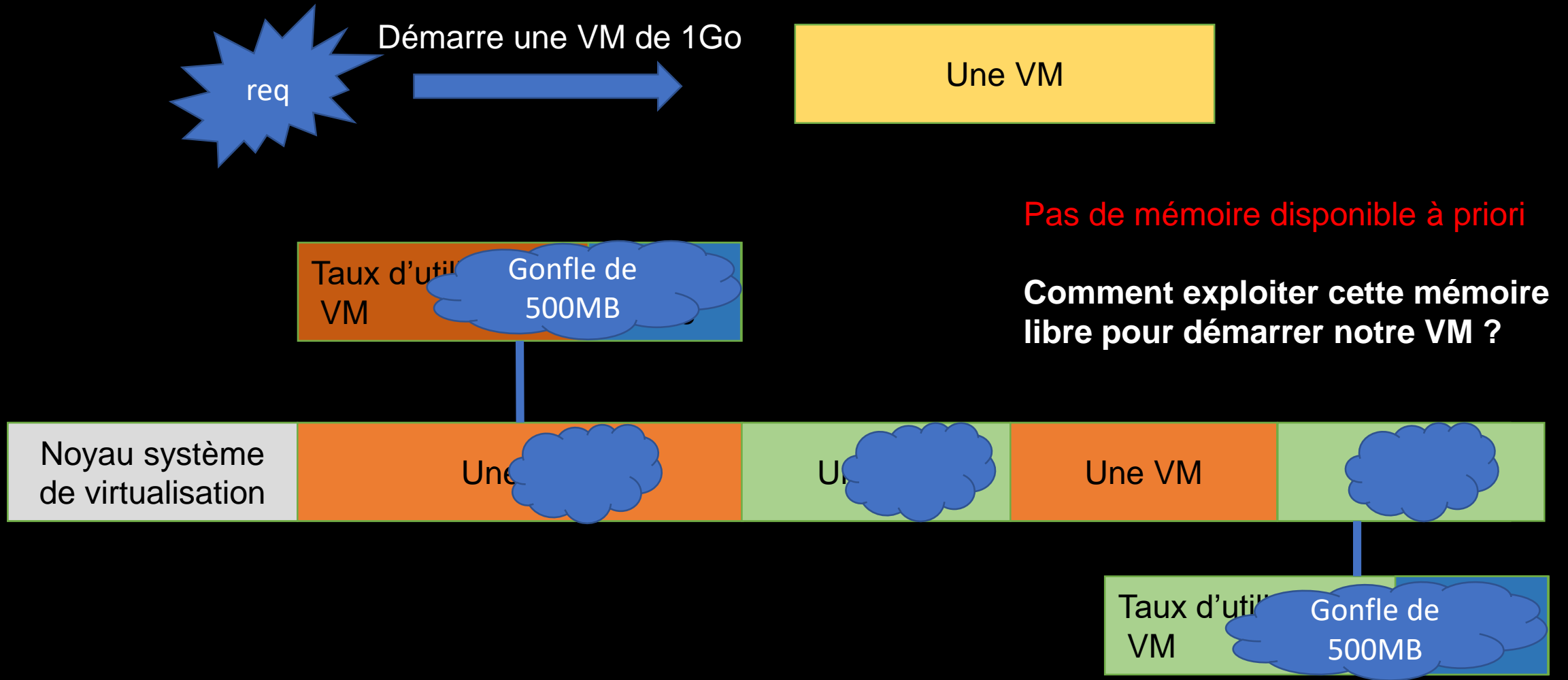
# Rôle du consolidateur (NP-complet)

Exploiter la mémoire non-utilisée – Principe du ballooning



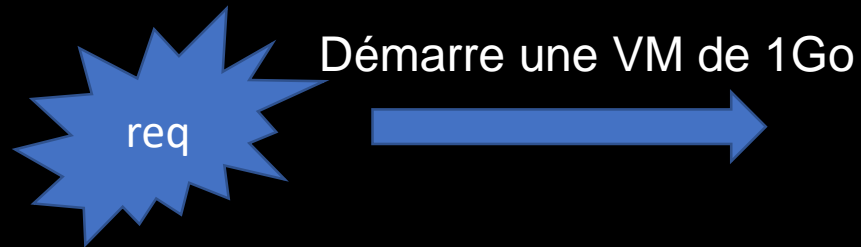
# Rôle du consolidateur (NP-complet)

Exploiter la mémoire non-utilisée – Principe du ballooning



# Rôle du consolidateur (NP-complet)

Exploiter la mémoire non-utilisée – Principe du ballooning



Pas de mémoire disponible à priori

Comment exploiter cette mémoire libre pour démarrer notre VM ?



Noyau système  
de virtualisation

Une

Une

Une VM

Assez de  
place 😊

Taux d'utilisation  
VM

Gonfle de  
500MB

# Déploiement ???

Automatisation

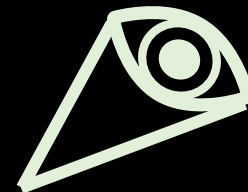
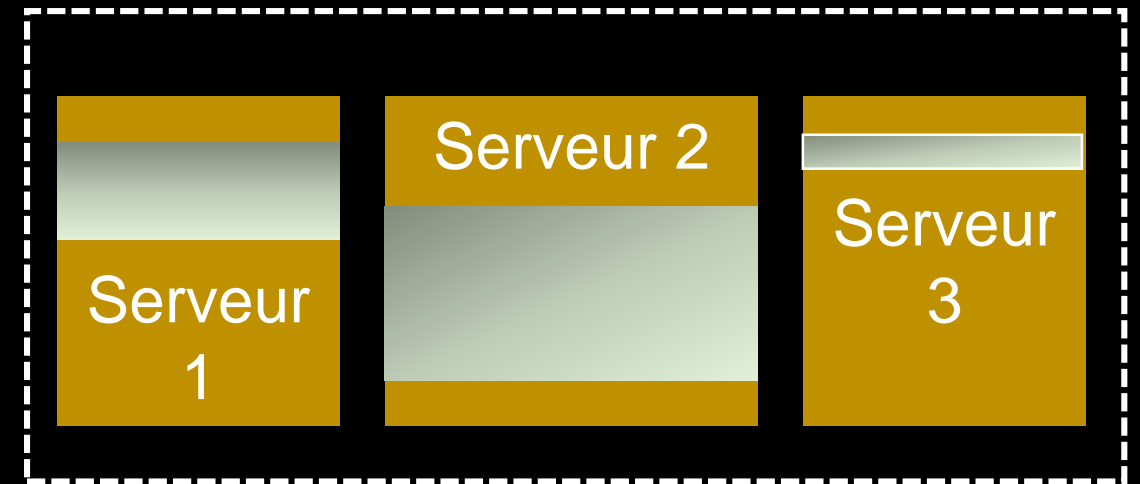
Choix des nœuds

En continu

**Surveiller**



Taux utilisation  
ressources



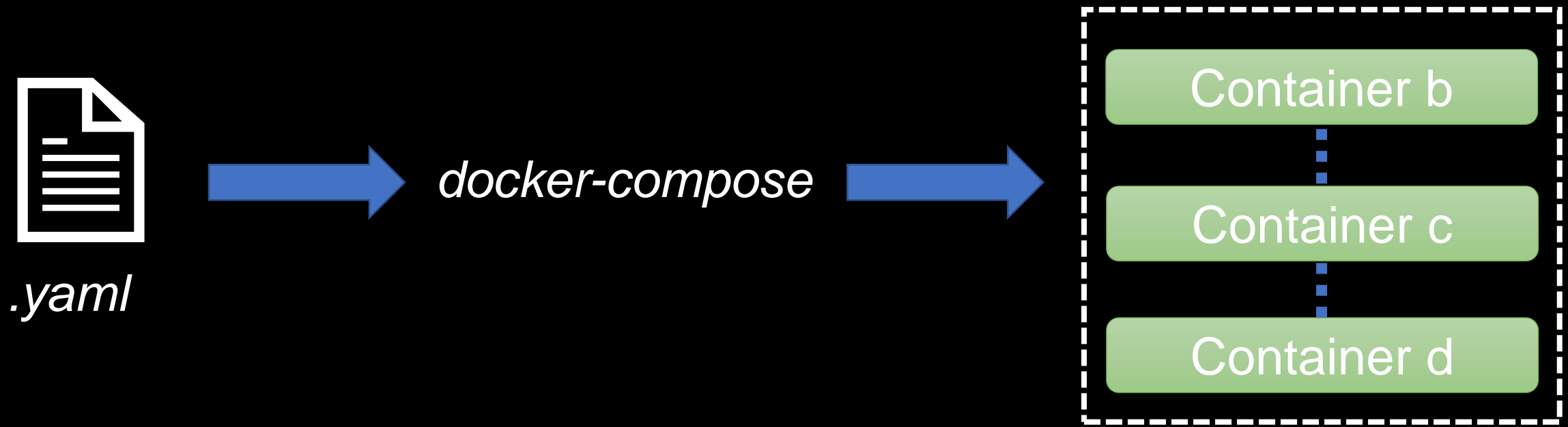
Observer les modules et  
déclencher les actions  
correspondantes





# Automatisation – docker-compose

**docker-compose** permet l'automatisation le déploiement d'une application multi-container --- en automatisant les différentes configuration à effectuer



# Déploiement ???

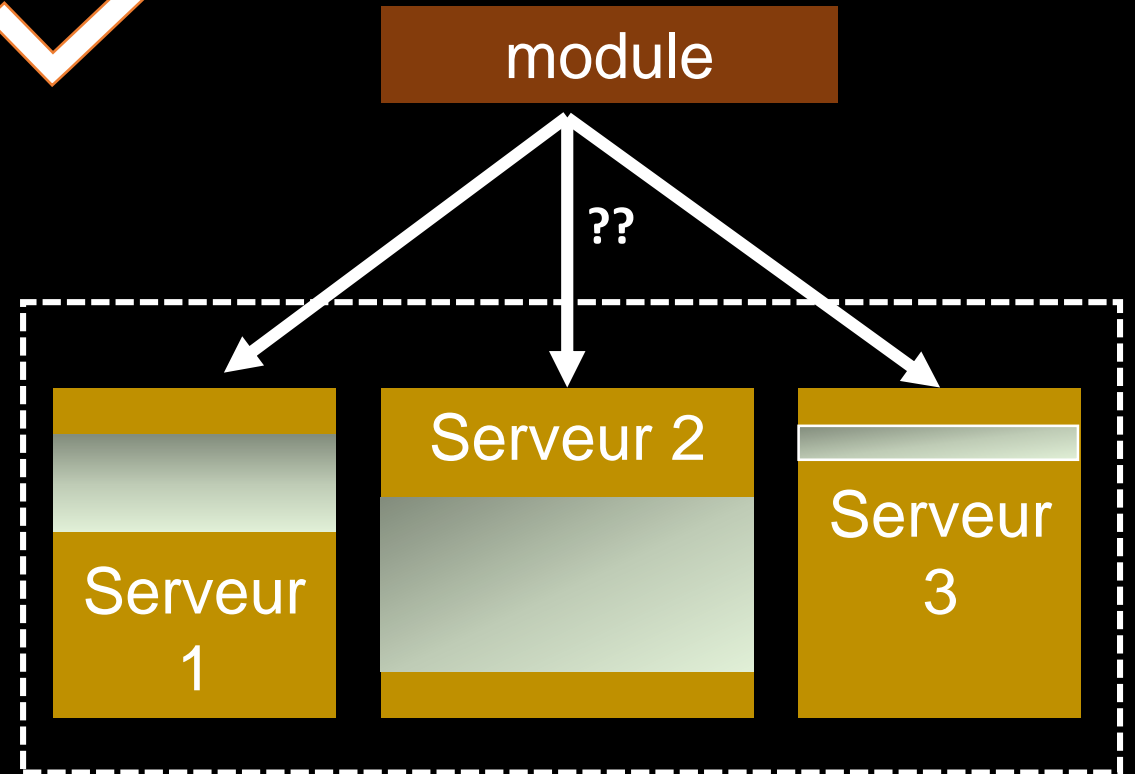
Automatisation

Choix des nœuds

En continu

Surveiller

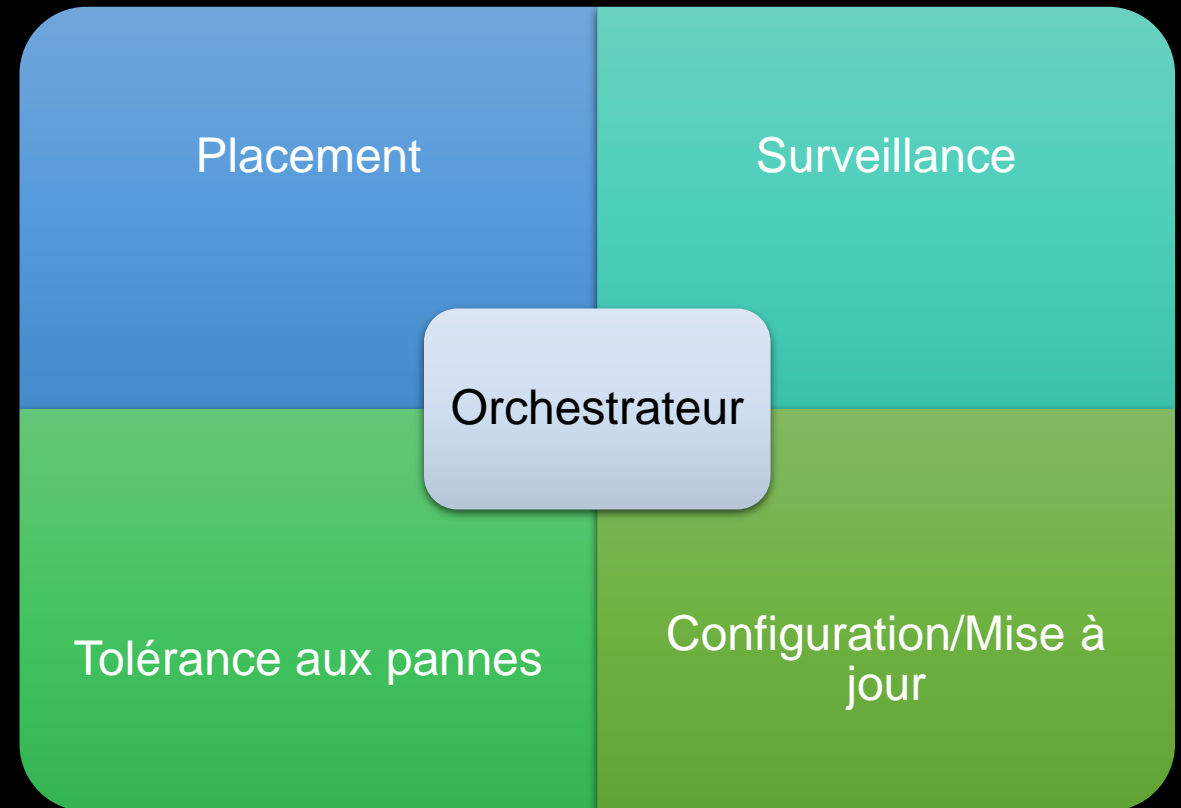
 Taux utilisation  
ressources



# Orchestrateur

L'orchestrateur se charge de la **gestion** des unités d'isolations.

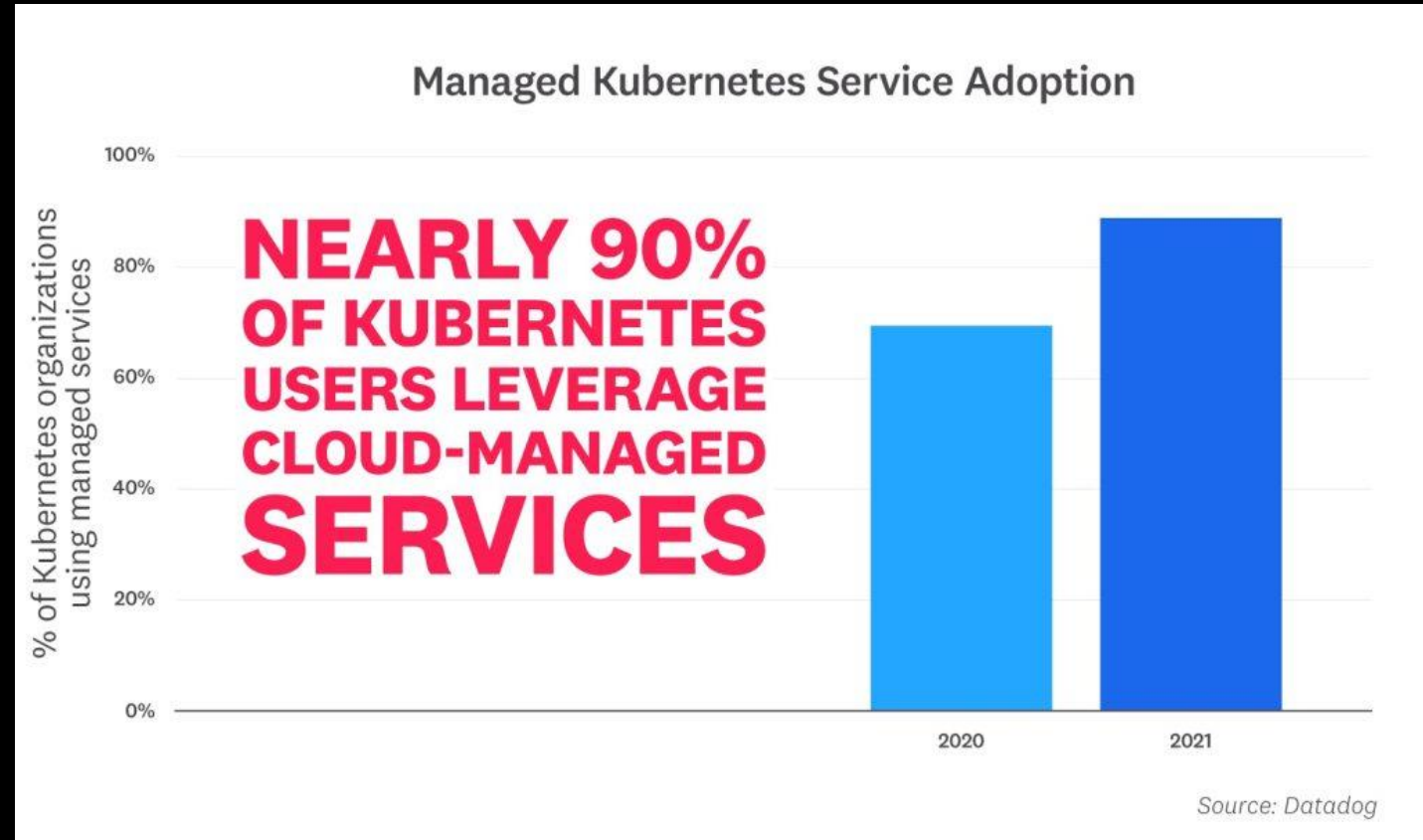
Elle gère le **placement** des unités sur les différents nœuds en fonction du **taux d'utilisation** et **surveille** les unités pour assurer leur **disponibilités**.



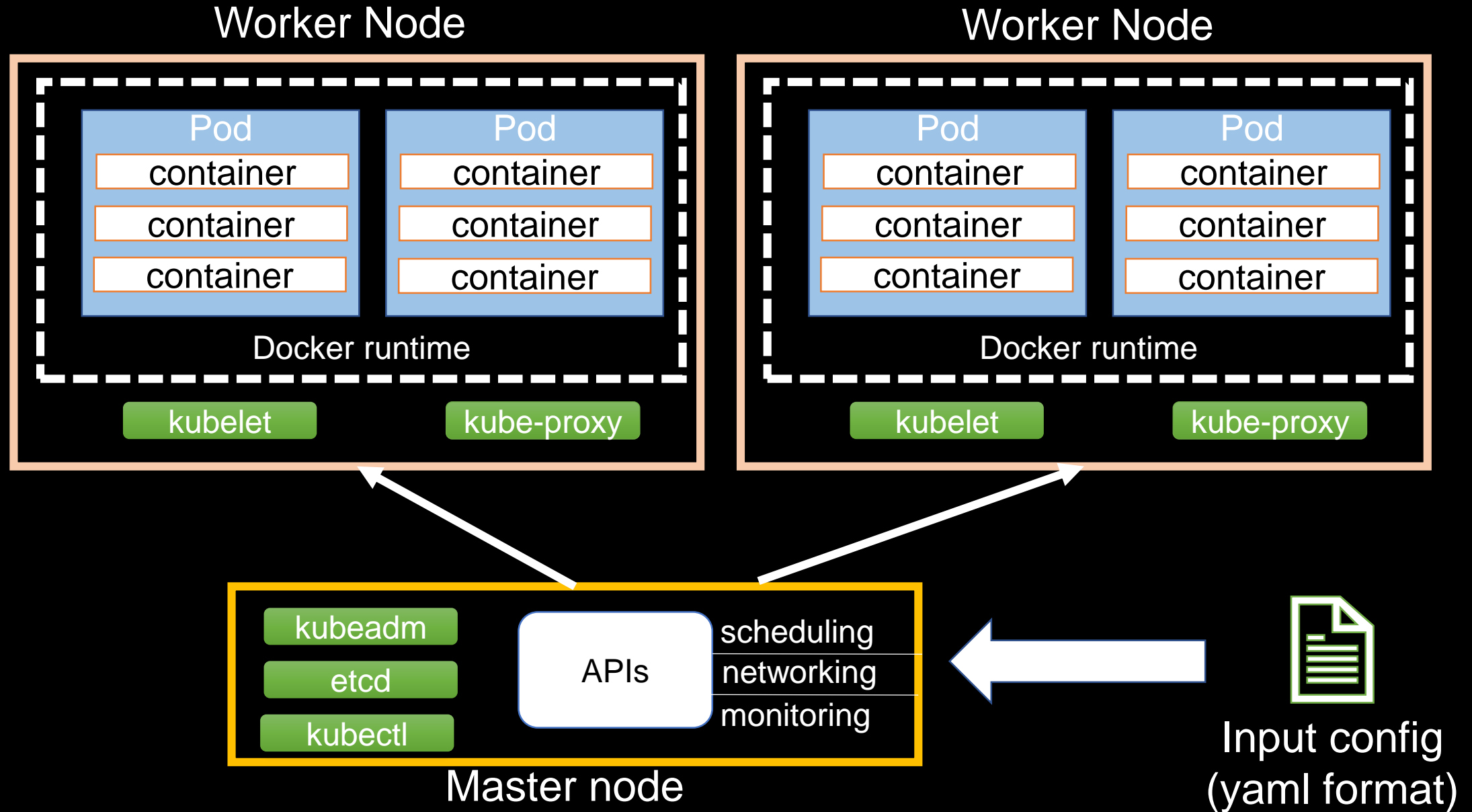
# Orchestrateur - Containers

Plusieurs orchestrateurs open-source et propriétaires existent pour les containers, Docker Swarm, Openshift, Kubernetes, Mesos, Nomad, Rancher, etc.

Nous allons illustrer les concepts d'un orchestrateur avec **Kubernetes**, mais tous reposent sur les même principes.



# Orchestrateur – Kubernetes (k8s)



# Orchestrateur – Kubernetes (k8s)

Kubernetes repose sur une architecture master/workers.

Les nœuds workers possède l'environnement pour l'exécution des containers.

Le nœud master communique avec les workers pour surveiller l'état des containers grâce au service **kubelet** et persiste les données sur **etcd**.

Les nœuds workers créent des containers dans une abstraction appelé **pods** qui symbolise un groupe de containers qui réalisent un objectif précis.

L'isolation réseau est assuré par le composant **kube-proxy**

# Orchestrateur – Kubernetes (k8s)

## On Ubuntu

```
sudo apt-get update
```

```
sudo apt-get install -y apt-transport-https ca-certificates  
curl
```

```
sudo curl -fsSL /usr/share/keyrings/kubernetes-archive-  
keyring.gpg  
https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

```
echo "deb [signed-by=/usr/share/keyrings/kubernetes-  
archive-keyring.gpg] https://apt.kubernetes.io/  
kubernetes-xenial main" | sudo tee  
/etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl
```

## On Windows (kubectl, kubelet)

Installer docker-desktop puis aller dans les paramètres de Docker et assurer vous que Kubernetes est activé (diode verte)

Kubeadm ne supporte pas les hôtes Windows mais les worker peuvent tourner sous Windows

Tester votre installation avec :

```
kubectl --version  
kubectl get all
```

# Orchestrateur – Kubernetes (k8s)

**Format du fichier de configuration**

**Télécharger le fichier deployment.yaml à  
l'adresse suivante :**

**[https://github.com/djobiii2078/cloud\\_course\\_resources/blob/main/deployment.yaml](https://github.com/djobiii2078/cloud_course_resources/blob/main/deployment.yaml)**

**Pouvez-vous expliquer ce qu'il décrit ?**



# Orchestrateur – Kubernetes (k8s)

## Déploiement continu

```
minReadySeconds: 10
strategy:
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 0
type: RollingUpdate
```

Définit la stratégie de mise à jour pour notre déploiement.

MaxSurge conditionne le nombre de containers à instancier lors du passage à l'échelle.

Mettez à jour le déploiement :

```
kubectl apply -f deployment.yaml
```

Qu'observez-vous ? Essayer d'arreter un de vos containers et décrivez ce qui se passe.

# Orchestrateur – Kubernetes (k8s)

## Déploiement continu

```
spec:  
  containers:  
  - name: bb-site:v2  
    image: getting-started
```

Le fichier peut être mis à jour pour faire évoluer l'architecture.

Kubernetes redéploie les composants modifiés pour s'ajuster aux nouvelles directives.

Mettez à jour le déploiement :

```
kubectl apply -f deployment.yaml
```

Vérifier si l'image de vos containers ont bien évolué ...

# Orchestrateur – Kubernetes (k8s)

## Surveillance (Monitoring) : Kubernetes dashboard

