

# Sense4data x Datacraft

## L'atelier Kubernetes

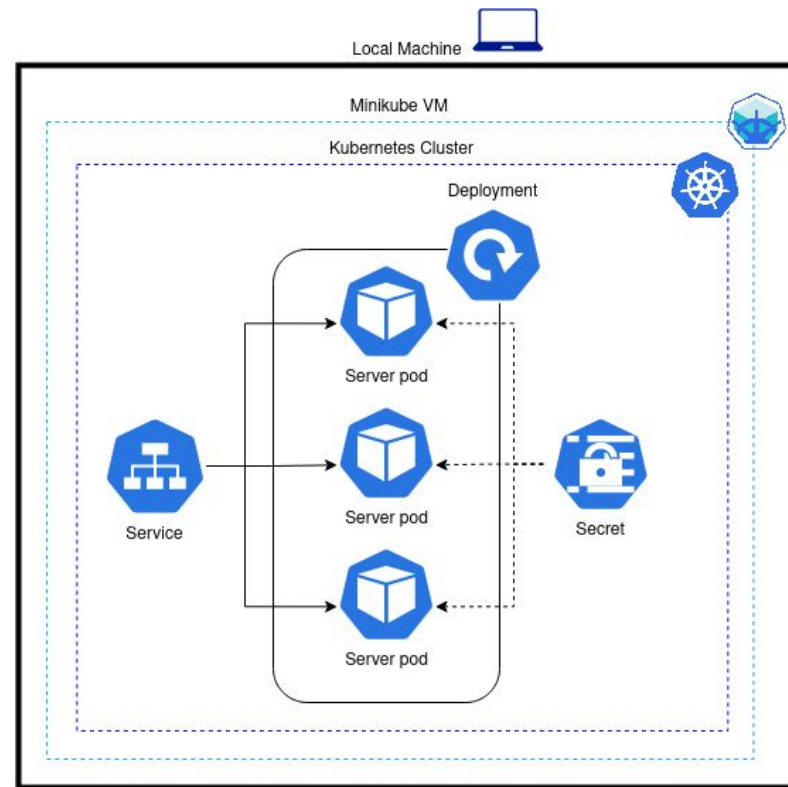
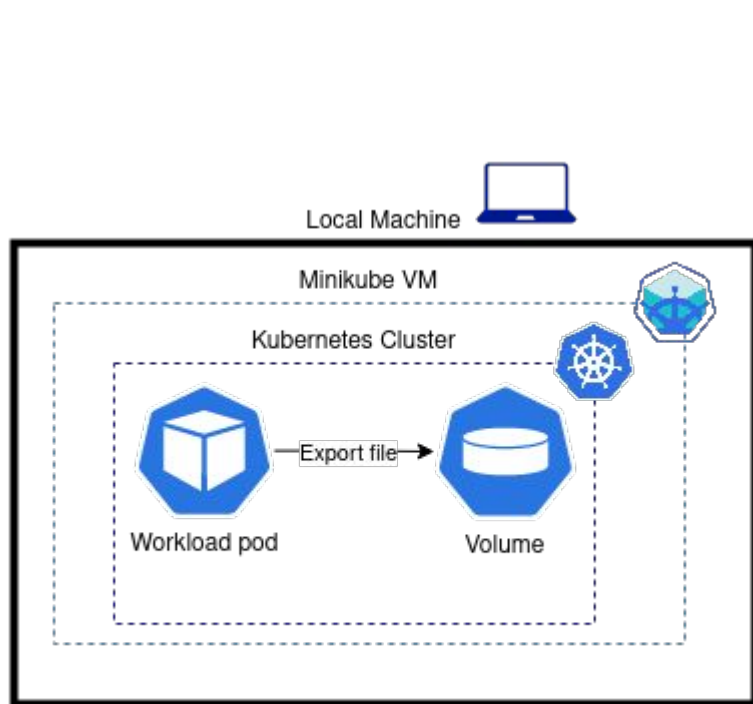


# Aperçu

Introduction	01
Pourquoi Kubernetes?	02
Outils	03
Architecture de Kubernetes	04
Les pods	05
Les volumes	06
Les déploiements	07
Les services	08
Les secrets	09

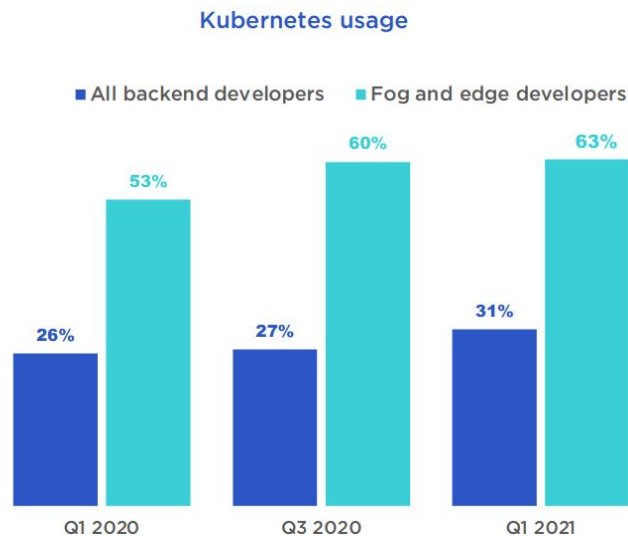
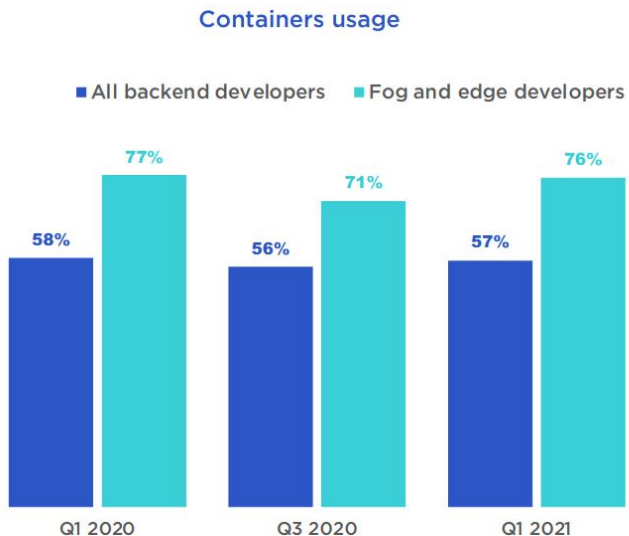
**07/04/2022**

# Objectif de l'atelier



# Kubernetes dans la nature

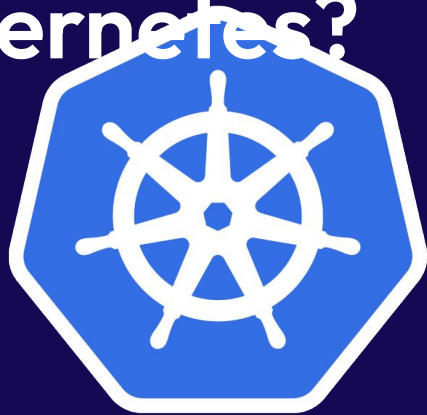
Kubernetes est très répandu et son adoption ne fait que grimper.



Extrait du "The state report of cloud native development" 2021

Pourquoi

Kubernetes?

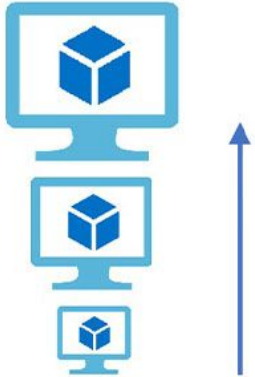


# Pourquoi Kubernetes?

La scalabilité verticale peut rapidement devenir très coûteuse.

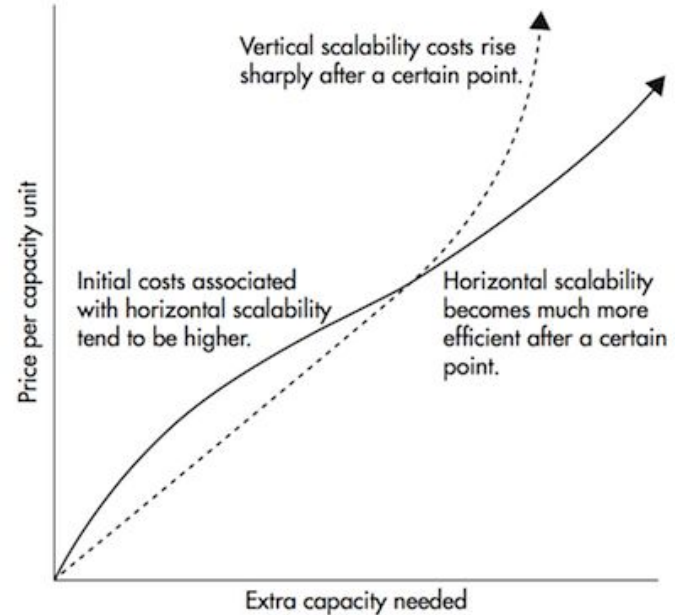
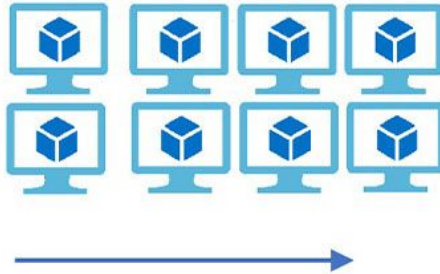
## Vertical Scaling

( Increase size of instance (RAM , CPU etc.) )



## Horizontal Scaling

( Add more instances )



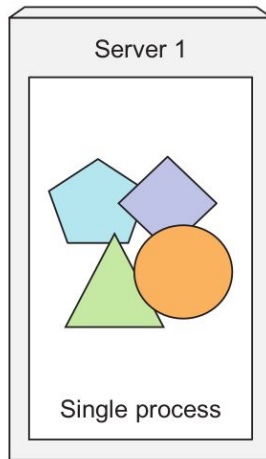
# Pourquoi Kubernetes?

## Le passage vers des architectures microservices

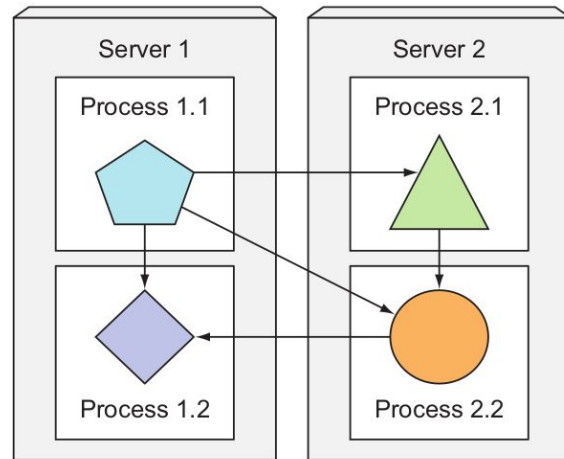
Un microservice est un composant:

- Indépendamment scalable
- Indépendamment déployable
- Développé par une “petite” équipe autonome.

Monolithic application



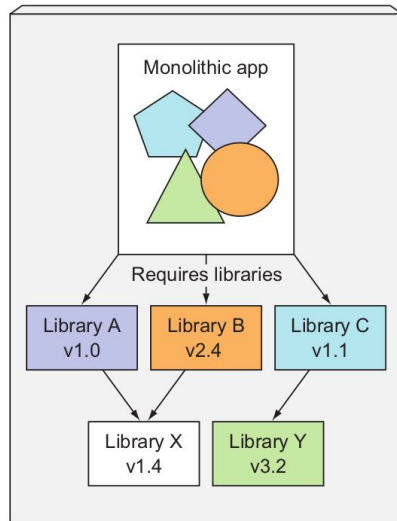
Microservices-based application



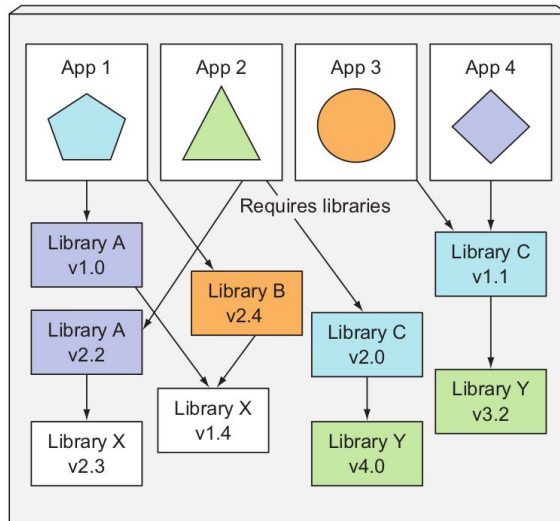
# Pourquoi Kubernetes?

## L'adoption massive des technologies de conteneurisation

Server running a monolithic app

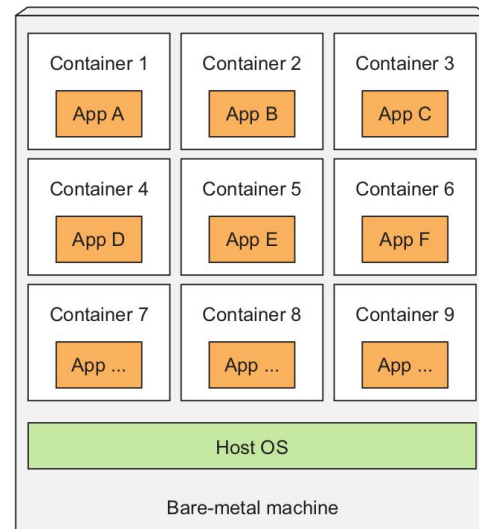


Server running multiple apps



Conteneurisation

Apps running in isolated containers





# Outils

# Outils

Le cluster Kubernetes: Minikube

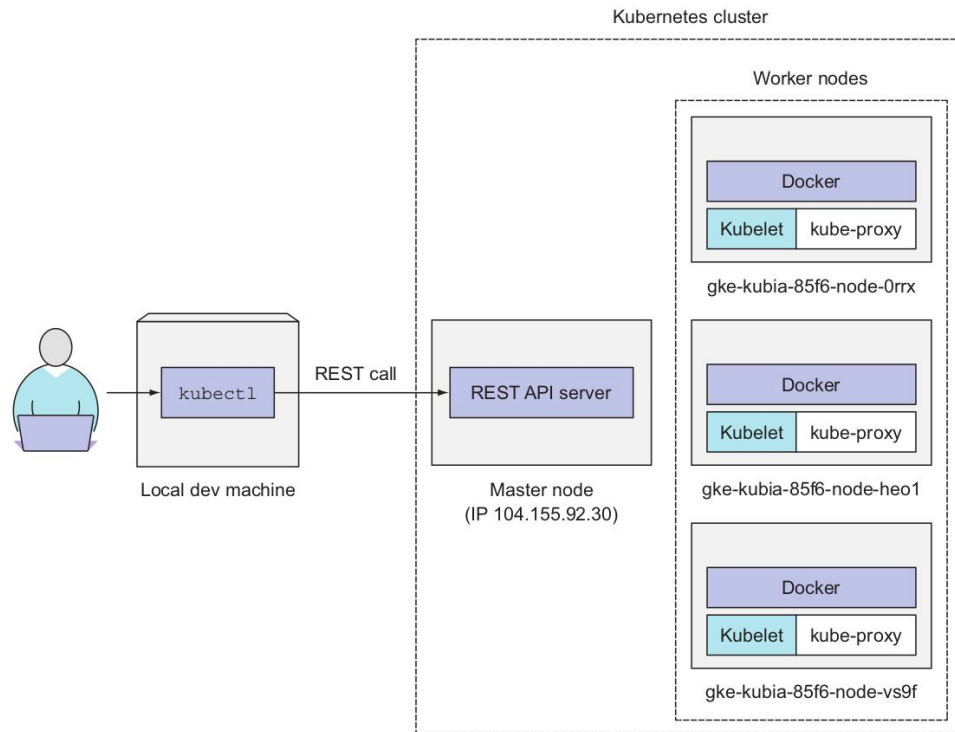


minikube

-> *minikube start*

# Outils

## Le client Kubernetes: Kubectl



# Outils

Le client Kubernetes: Kubectl

`kubectl` [Command] [ResourceType] [ResourceName] [flags]



- create
- get
- describe
- delete
- ...



- pods
- services
- secrets
- deployments
- ...

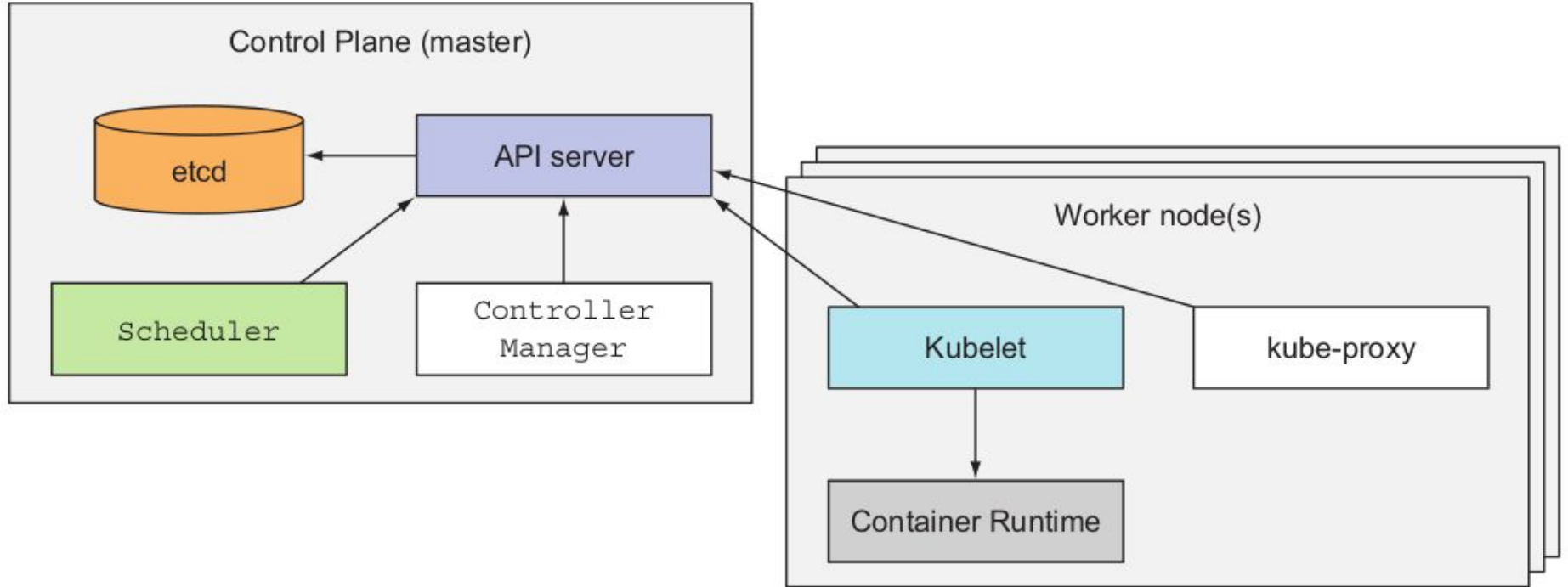
# Outils

Pour commencer: Clonez le code sur votre machine

```
git clone https://github.com/Reda-Bona/datacraft-k8s.git
```

# Architecture de Kubernetes

# Architecture de Kubernetes



# Concepts fondamentaux



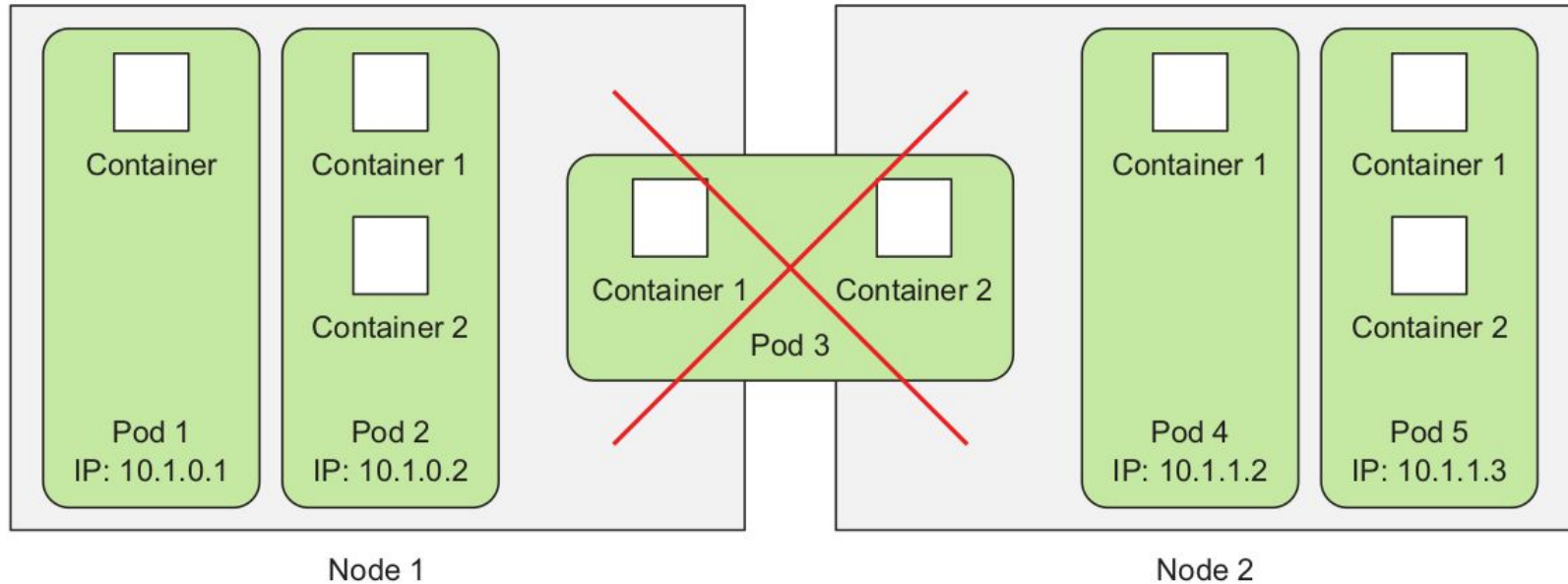
## Concepts fondamentaux

# Les pods



- Qu'est ce qu'un pod?
- Comment créer un pod?
- Comment interagir avec un pod?

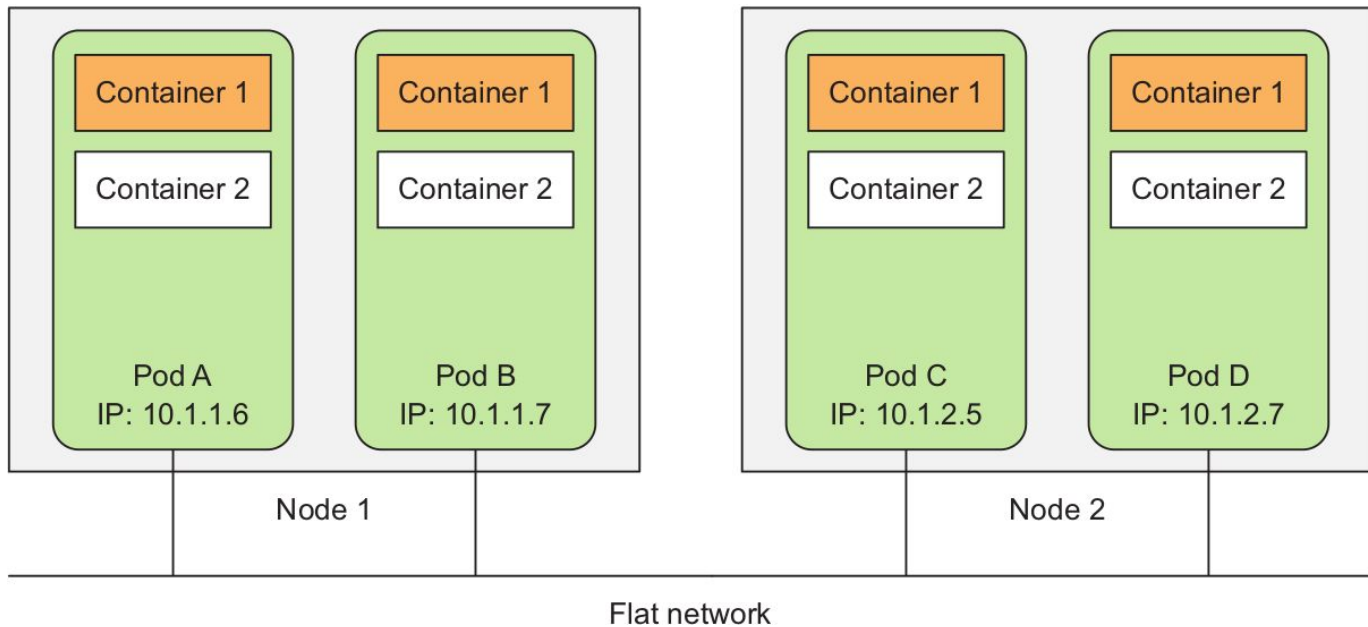
# Qu'est ce qu'un pod?



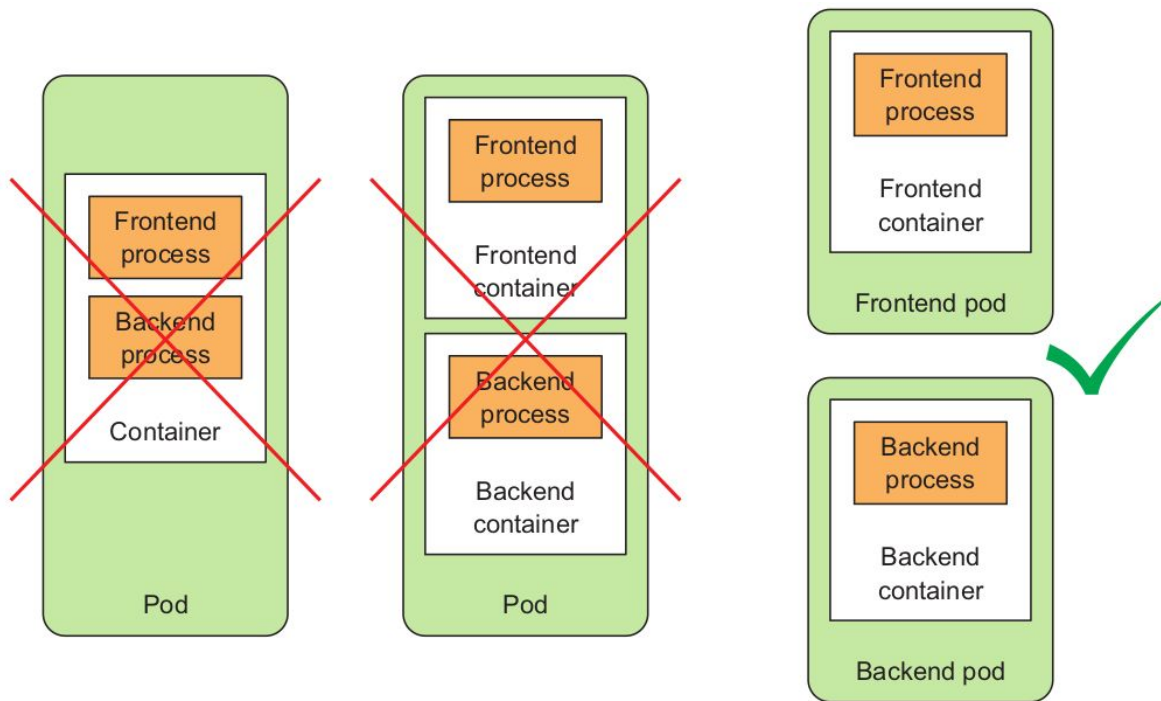
Qu'est ce qu'un pod?

**Un pod est éphémère**

# Qu'est ce qu'un pod?



# Qu'est ce qu'un pod?



# Comment créer un pod?

```
kubectl apply -f k8s/nginx/pod.yml
```

```
k8s > nginx > vim pod.yml
1  apiVersion: v1
2
3  kind: Pod
4
5  metadata:
6    name: webserver
7  spec:
8    containers:
9      - name: webserver
10        image: nginx:latest
11        ports:
12          - containerPort: 80
```

# Comment interagir avec un pod?

- Voir l'état de tous les pods déployés -> `kubectl get pods`
- Voir l'état du pod -> `kubectl get pod ${pod_name}`
- Voir la description du pod -> `kubectl describe pod ${pod_name}`
- Voir les logs du pod -> `kubectl logs ${pod_name}`
- Supprimer le pod. -> `kubectl delete pod ${pod_name}`

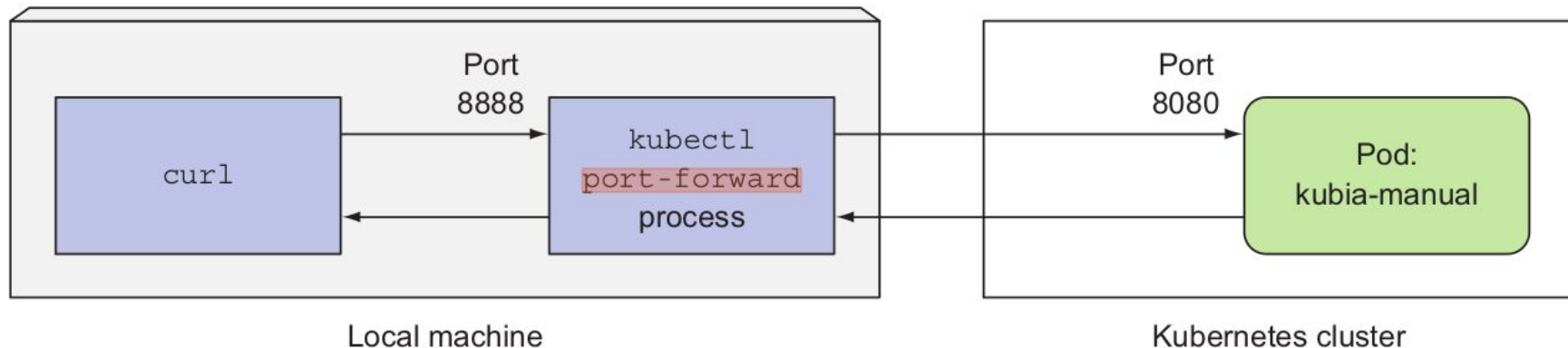
# Comment interagir avec un pod?

Pour exposer le server localement.

Nom définit dans la metadata du pod



La méthode “quick and dirty” → `kubectl port-forward webserver 8888:80`





Concepts fondamentaux

# Les volumes

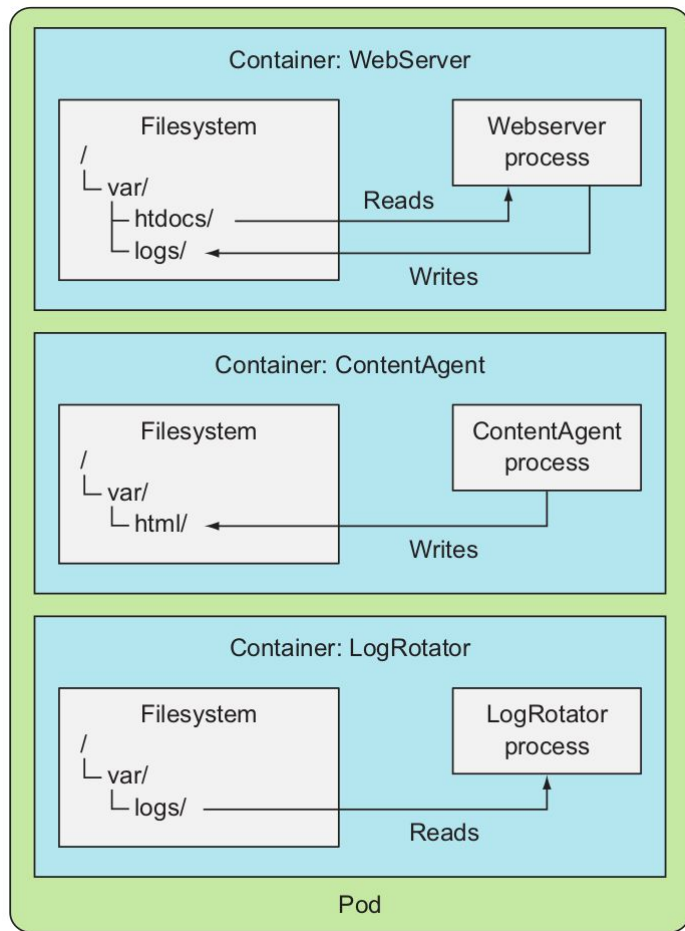


- Pourquoi un volume?
- Comment créer un volume?

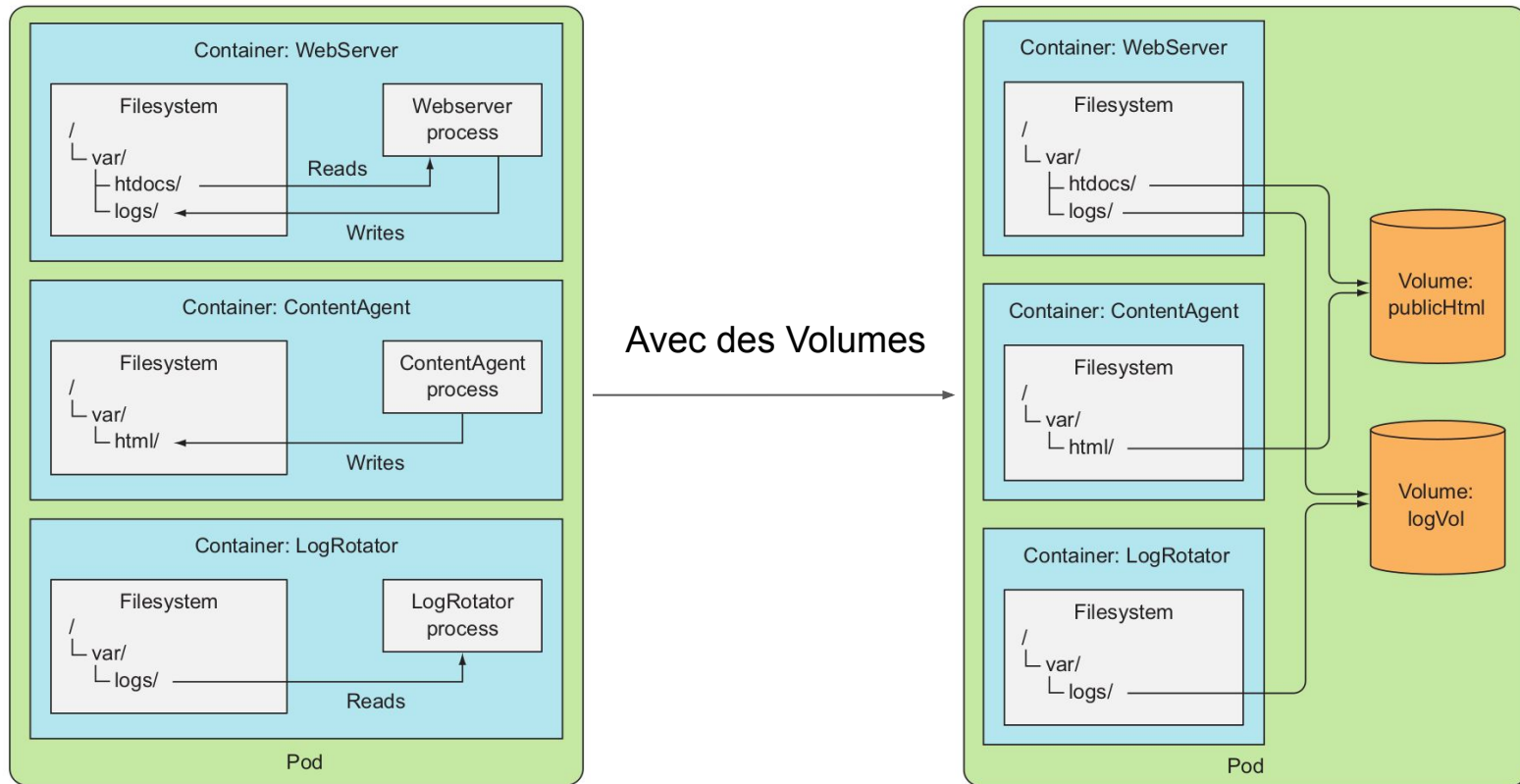
# Pourquoi un volume?

A travers un exemple:

- **Webserver**: renvoie des pages HTML à partir de `/var/htdocs` et logs les accès dans `/var/logs`
- **ContentAgent**: Génère des pages html dans `/var/html`
- **LogRotator**: Lit les logs dans `/var/html` et les traitent: rotation, compression, analyse...



# Pourquoi un volume?



# Comment créer un volume?

NB: Dans le contexte de minikube. Le volume est créé dans la VM de minikube et non pas dans la machine. Pour accéder au volume dans le système de fichier local il faut dire à minikube de le monter.

```
minikube mount ./data:/data
```

```
k8s > workflow_app > pod.yml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: my-workflow
5  spec:
6    containers:
7      - name: app
8        image: redabona/datacraft-k8s-workshop:toy-workflow
9        volumeMounts:
10       - mountPath: /data
11         name: pv0001
12       ports:
13       - containerPort: 80
14       imagePullPolicy: Always
15     volumes:
16       - name: pv0001
17         hostPath:
18           path: /data/pv0001
```

Concepts fondamentaux

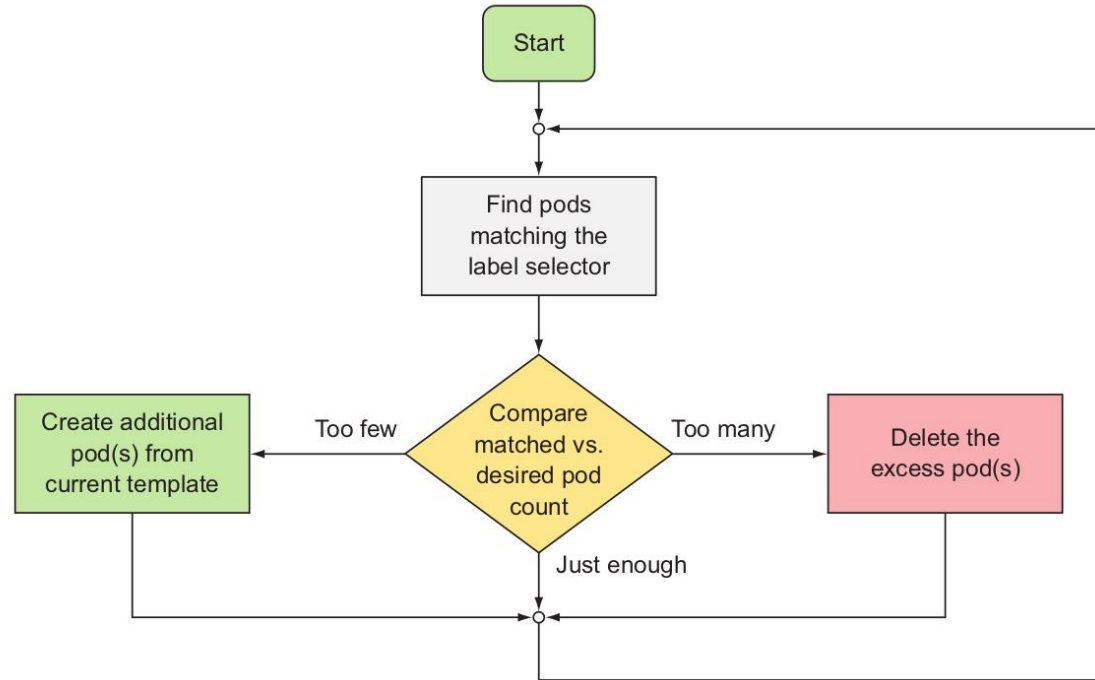
# Les déploiements



- Pourquoi un déploiement?
- Comment créer un déploiement?
- Comment interagir avec un déploiement?

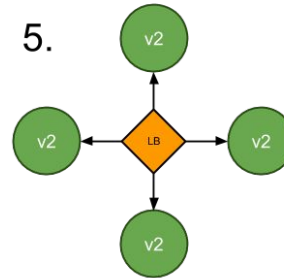
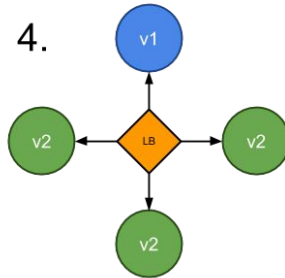
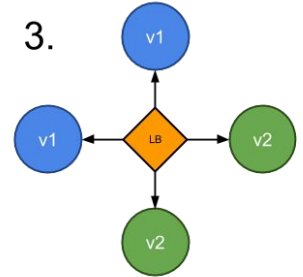
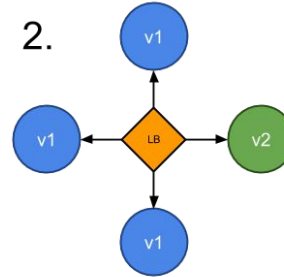
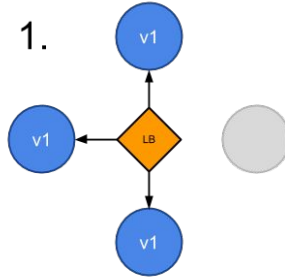
# Pourquoi un déploiement?

## Rôle #1: Replication controller



# Pourquoi un déploiement?

Rôle #2: Mise à jour des pods



# Comment créer un déploiement?

```
kubectl apply -f  
k8s/web_app/deployment.yml
```

```
k8s > web_app > vim deployment.yml  
1  apiVersion: apps/v1  
2  kind: Deployment  
3  metadata:  
4    name: my-app-deployment  
5    labels:  
6      app: my-app-label  
7  spec:  
8    replicas: 3  
9    selector:  
10     matchLabels:  
11       app: my-app-label  
12   template:  
13     metadata:  
14       labels:  
15         app: my-app-label  
16     spec:  
17       containers:  
18         - name: my-app  
19           image: redabona/datacraft-k8s-workshop:toy-server  
20           ports:  
21             - containerPort: 80  
22           env:  
23             - name: APP_ITEM_VALUE  
24               value: "Configurable item"  
25             - name: APP_SECRET_STRING  
26               valueFrom:  
27                 secretKeyRef:  
28                   name: mysecret  
29                   key: APP_SECRET_STRING
```



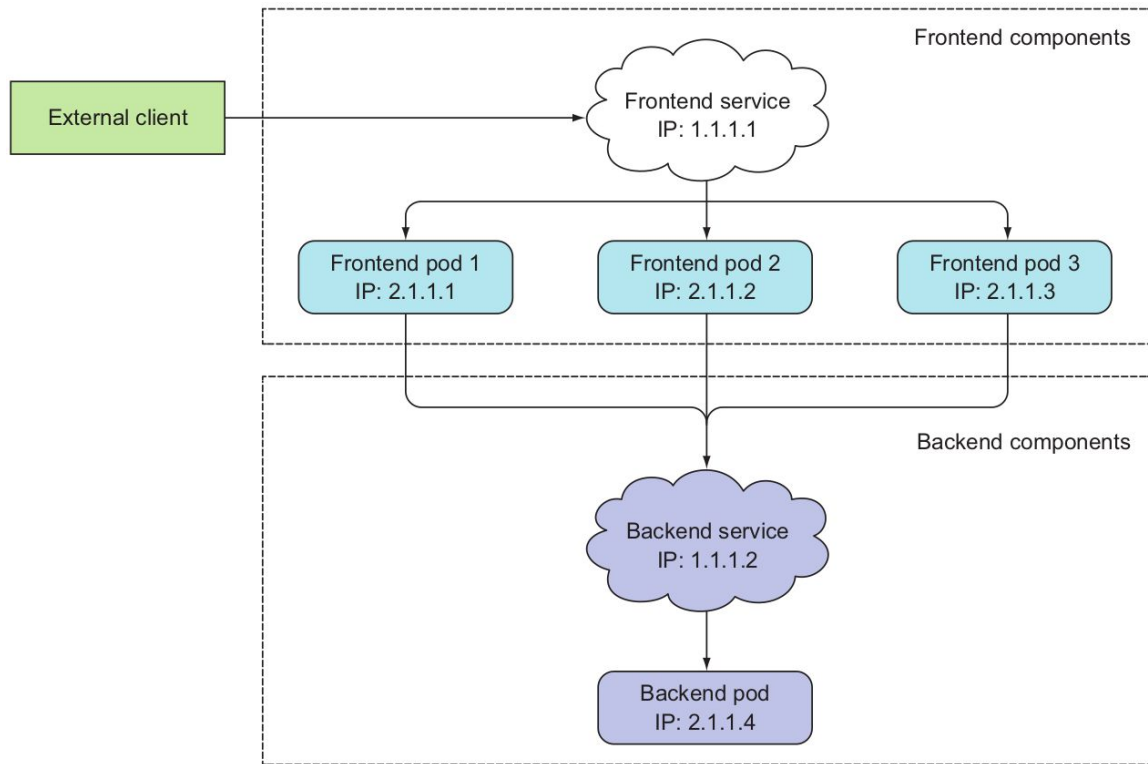
Concepts fondamentaux

# Les services



- Pourquoi un service?
- Comment créer un service?

# Pourquoi un service?



# Comment créer un service?

```
kubectl apply -f  
k8s/web_app/service.yml
```

```
k8s > web_app > vim service.yml  
1  apiVersion: v1  
2  kind: Service  
3  metadata:  
4    name: my-service  
5    labels:  
6      app: my-app-label  
7  spec:  
8    type: NodePort  
9    selector:  
10     app: my-app-label  
11    ports:  
12     - port: 80  
13       targetPort: 80  
14       nodePort: 30007
```

Concepts fondamentaux

# Les secrets



- Pourquoi un secret?
- Comment créer un secret?

# Comment créer un secret?

```
kubectl apply -f k8s/web_app/secret.yml
```

```
k8s > web_app > yaml secret.yml
```

```
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: mysecret
5  type: Opaque
6  stringData:
7    APP_SECRET_STRING: "My secret string!!!"
```



**Fin**

**Des questions?**

