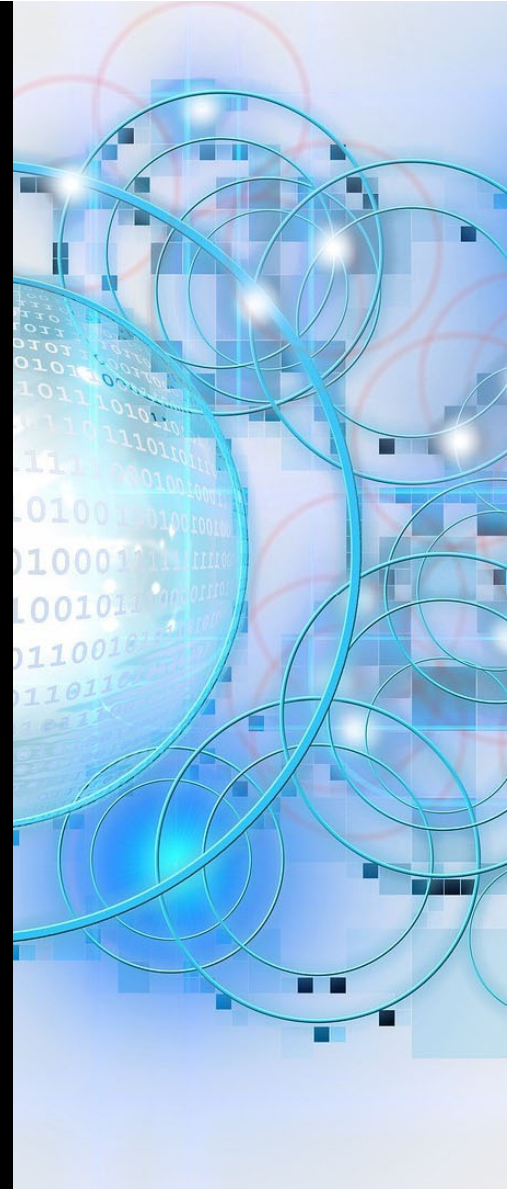


# Explication de l'exercice 15 partie 3

---



# Service:

Les Pods Kubernetes sont mortels. Ils naissent et lorsqu'ils meurent, ils ne ressuscitent pas. Si vous utilisez un Déploiement pour exécuter votre application, il peut créer et détruire dynamiquement des pods.

Chaque pod obtient sa propre adresse IP, mais dans un déploiement, l'ensemble de pods s'exécutant en un instant peut être différent de l'ensemble de pods exécutant cette application un instant plus tard.

Le **service** est une manière abstraite **d'exposer une application** s'exécutant sur un ensemble de Pods **en tant que service réseau**.

Kubernetes attribue à ce service une adresse IP (parfois appelé l'"IP cluster"), qui est utilisé par les proxies Service

Chaque nœud d'un cluster Kubernetes exécute un kube-proxy. kube-proxy est responsable de l'implémentation d'une forme d'IP virtuelle pour les Services qui ne sont pas de type [ExternalName](#). (Exemple: my.database.example.com)

Le contrôleur de service recherche en continu les pods qui correspondent à son sélecteur, puis POST toutes les mises à jour d'un objet Endpoint également appelé "my-service" (mon-service dans le schéma qui suit).

Source : <https://kubernetes.io/fr/docs/concepts/services-networking/service>

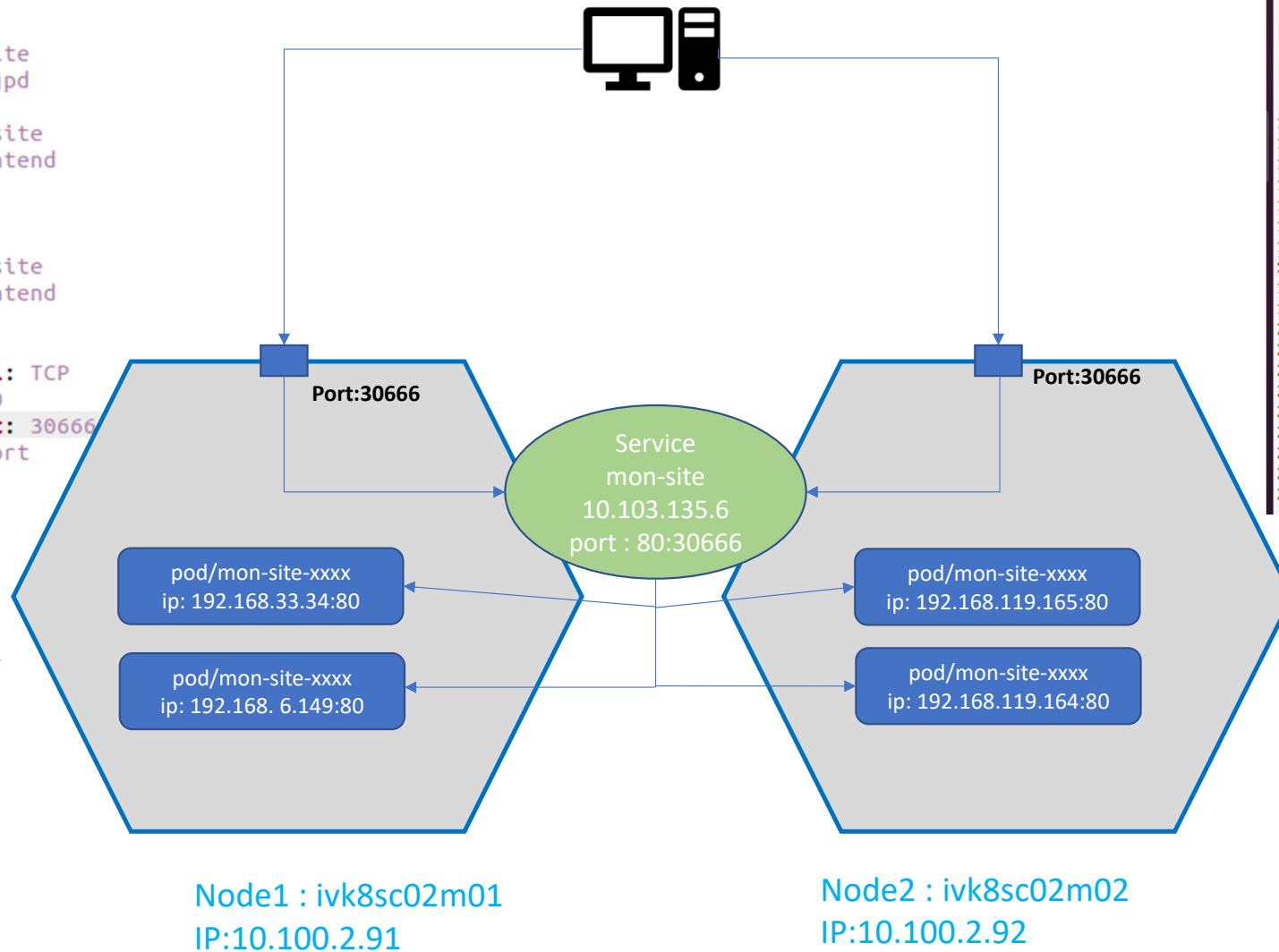
```

1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: mon-site
5   namespace: jpd
6   labels:
7     app: mon-site
8     tier: frontend
9     env: unit
10 spec:
11   selector:
12     app: mon-site
13     tier: frontend
14     env: unit
15   ports:
16   - protocol: TCP
17     port: 80
18     nodePort: 30666
19   type: NodePort

```

**NodePort:** Expose le service sur l'IP de chaque nœud sur un port statique (le NodePort). Un service ClusterIP, vers lequel le service NodePort est automatiquement créé. Vous pourrez contacter le service NodePort, depuis l'extérieur du cluster, en demandant <NodeIP>: <NodePort>

Lors de l'utilisation du type NodePort vous devez utiliser la plage 30000 à 32767.



```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: mon-site-deploiment
5   namespace: jpd
6   labels:
7     app: mon-site
8     tier: frontend
9     env: unit
10 spec:
11   replicas: 10
12   selector:
13     matchLabels:
14       app: mon-site
15       tier: frontend
16       env: unit
17   template:
18     metadata:
19       labels:
20         app: mon-site
21         tier: frontend
22         env: unit
23     spec:
24       containers:
25       - name: nginx
26         image: nginxdemos/hello
27         ports:
28         - containerPort: 80

```

```
jpduches@VM-DevOpsJPD:~/k8s/exer15part3$ kubectl get all -o wide
```

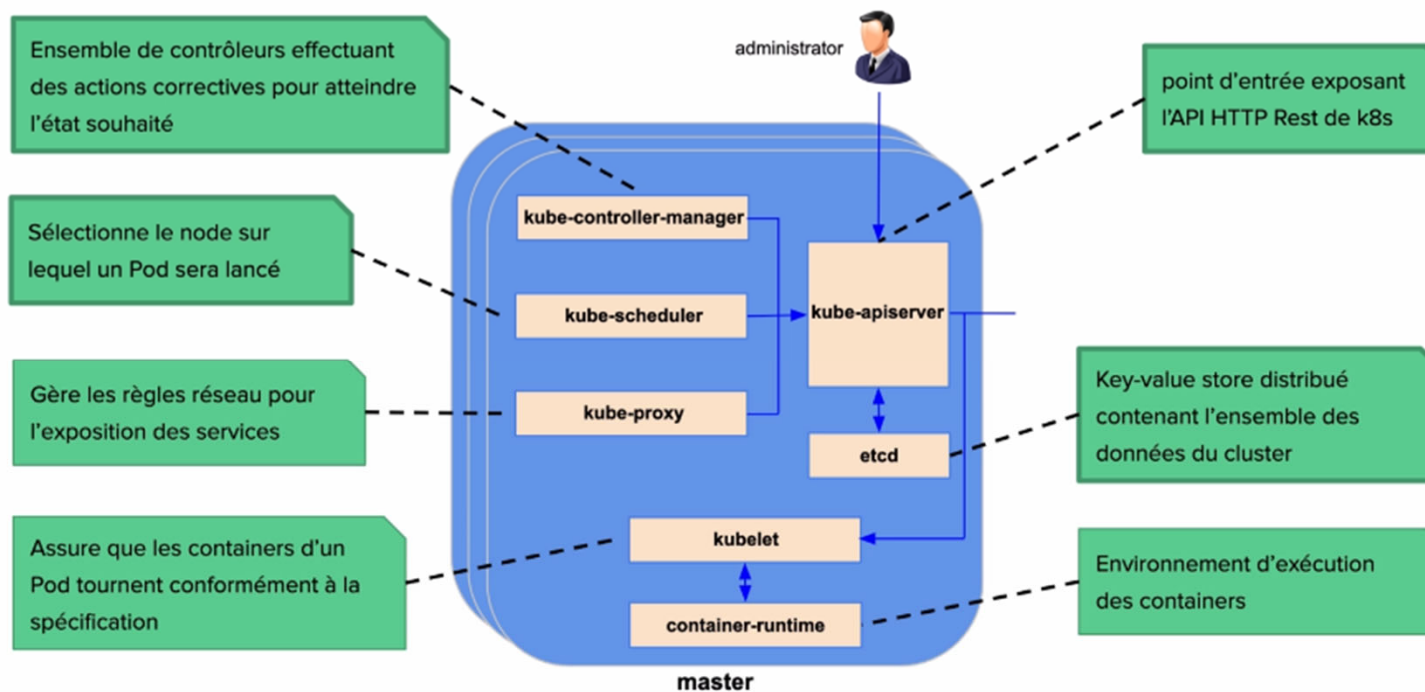
NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE
pod/mon-site-deploiment-78769cc56f-6fxqk	1/1	Running	0	71m	192.168.33.34	ivk8sc02w09	<none>
pod/mon-site-deploiment-78769cc56f-c57hc	1/1	Running	0	71m	192.168.24.49	ivk8sc02w04	<none>
pod/mon-site-deploiment-78769cc56f-dck2m	1/1	Running	0	71m	192.168.6.149	ivk8sc02w01	<none>
pod/mon-site-deploiment-78769cc56f-fbbfm	1/1	Running	0	71m	192.168.102.13	ivk8sc02w03	<none>
pod/mon-site-deploiment-78769cc56f-gzx6m	1/1	Running	0	71m	192.168.6.150	ivk8sc02w01	<none>
pod/mon-site-deploiment-78769cc56f-m6dqm	1/1	Running	0	71m	192.168.119.165	ivk8sc02w02	<none>
pod/mon-site-deploiment-78769cc56f-ptnv9	1/1	Running	0	71m	192.168.33.33	ivk8sc02w09	<none>
pod/mon-site-deploiment-78769cc56f-txjlm	1/1	Running	0	71m	192.168.24.47	ivk8sc02w04	<none>
pod/mon-site-deploiment-78769cc56f-vlpxt	1/1	Running	0	71m	192.168.119.164	ivk8sc02w02	<none>
pod/mon-site-deploiment-78769cc56f-vsp79	1/1	Running	0	71m	192.168.24.48	ivk8sc02w04	<none>
pod/monbash	1/1	Running	0	97m	192.168.119.227	ivk8sc02w05	<none>

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
service/mon-site	NodePort	10.103.135.6	<none>	80:30666/TCP	71m	app=mon-site,env=unit,tier=frontend

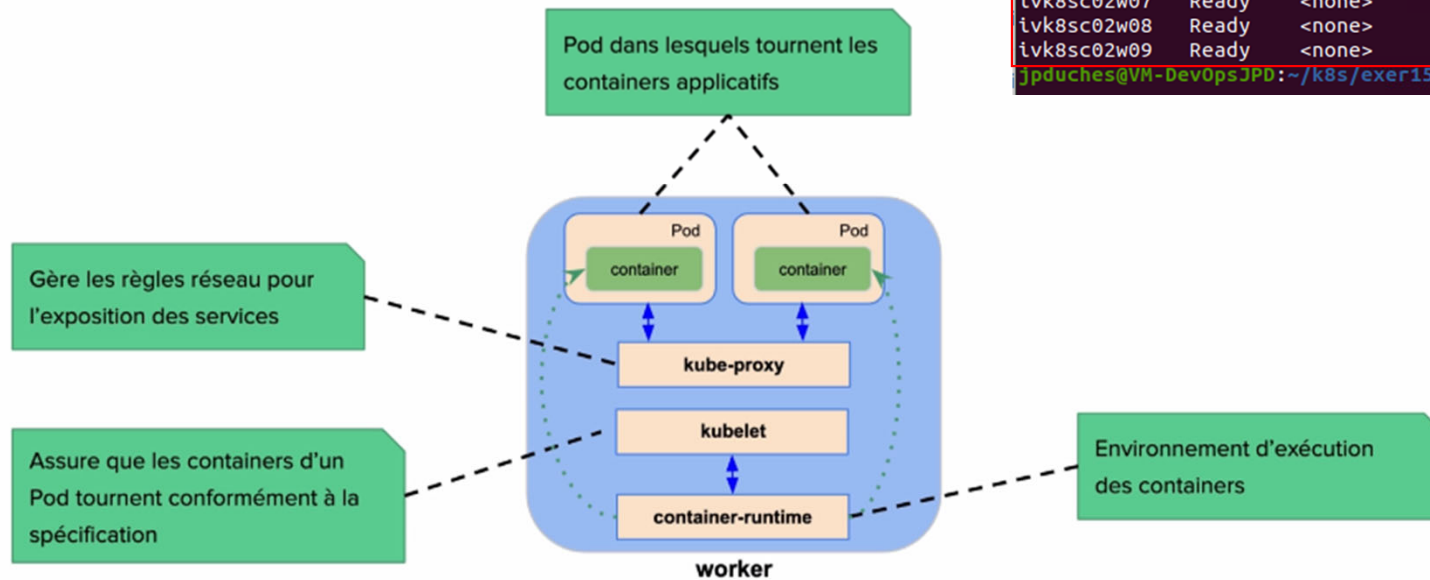
NAME	READY	UP-TO-DATE	AVAILABLE	AGE	CONTAINERS	IMAGES	SELECTOR
deployment.apps/mon-site-deploiment	10/10	10	10	71m	nginx	nginxdemos/hello	app=mon-site,env=unit,tier=frontend

# Les processus : node master

```
jpduches@VM-DevOpsJPD:~/k8s/exer15part3$ kubectl get node
NAME                STATUS    ROLES    AGE   VERSION
ivk8sc02m01         Ready     control-plane,master   12d   v1.20.10
ivk8sc02w01         Ready     <none>    12d   v1.20.10
ivk8sc02w02         Ready     <none>    12d   v1.20.10
ivk8sc02w03         Ready     <none>    12d   v1.20.10
ivk8sc02w04         Ready     <none>    12d   v1.20.10
ivk8sc02w05         Ready     <none>    12d   v1.20.10
ivk8sc02w06         Ready     <none>    12d   v1.20.10
ivk8sc02w07         Ready     <none>    12d   v1.20.10
ivk8sc02w08         Ready     <none>    12d   v1.20.10
ivk8sc02w09         Ready     <none>    12d   v1.20.10
jpduches@VM-DevOpsJPD:~/k8s/exer15part3$
```



# Les processus : node worker



```
jpduches@VM-DevOpsJPD:~/k8s/exer15part3$ kubectl get node
NAME                STATUS    ROLES    AGE   VERSION
ivk8sc02m01        Ready    control-plane,master   12d   v1.20.10
ivk8sc02w01        Ready    <none>    12d   v1.20.10
ivk8sc02w02        Ready    <none>    12d   v1.20.10
ivk8sc02w03        Ready    <none>    12d   v1.20.10
ivk8sc02w04        Ready    <none>    12d   v1.20.10
ivk8sc02w05        Ready    <none>    12d   v1.20.10
ivk8sc02w06        Ready    <none>    12d   v1.20.10
ivk8sc02w07        Ready    <none>    12d   v1.20.10
ivk8sc02w08        Ready    <none>    12d   v1.20.10
ivk8sc02w09        Ready    <none>    12d   v1.20.10
jpduches@VM-DevOpsJPD:~/k8s/exer15part3$
```