



kubernetes



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Kubernetes est une technologie de gestion de conteneurs développée dans le laboratoire Google pour gérer les applications conteneurisées dans différents types d'environnements tels que l'infrastructure physique, virtuelle et cloud. C'est un système open source qui aide à créer et à gérer la conteneurisation des applications. Avec la sortie de la première version « version 1,0 » google fait un don du projet Kubernetes en 2015 à la **Cloud Native Computing Foundation**.

Kubernetes peut fonctionner sur des plateformes variées: sur votre PC portable, sur des VMs d'un fournisseur de cloud, ou un rack de serveurs bare-metal.

A- Vous pouvez gérer kubernetes vous-même

- Nous pouvons installer kubernetes à partir de zéro sur deux machines de la manière traditionnelle
- Nous pouvons automatiser l'installation de kubernetes sur deux machines avec vagrant pour gagner du temps
- Nous pouvons utiliser Minikube : est une méthode pour créer un cluster Kubernetes local à noeud unique pour le développement et le test. L'installation est entièrement automatisée et ne nécessite pas de compte de fournisseur de cloud.
- Nous pouvons utiliser Microk8s : fournit une commande unique d'installation de la dernière version de Kubernetes sur une machine locale pour le développement et les tests.

B- Les fournisseurs de cloud géreront Kubernetes pour vous. (kubernetes as a service)

les principaux fournisseurs de cloud proposent de nombreuses offres Kubernetes-as-a-Service. **Amazon EKS** , **Google Cloud Kubernetes Engine** , **Azure Kubernetes Service (AKS)** , **Red Hat OpenShift** et **IBM Cloud Kubernetes Service**, fournissent tous une gestion complète de la plateforme Kubernetes, afin que vous puissiez vous concentrer sur ce qui compte le plus pour vous: expédier des applications qui ravissent vos utilisateurs.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Architecture du cluster Kubernetes

Lorsque vous commencez à travailler avec Kubernetes (K8), de nombreuses nouvelles terminologies sont à apprendre, à comprendre et à mémoriser.

Master node (like the manager node with docker swarm)

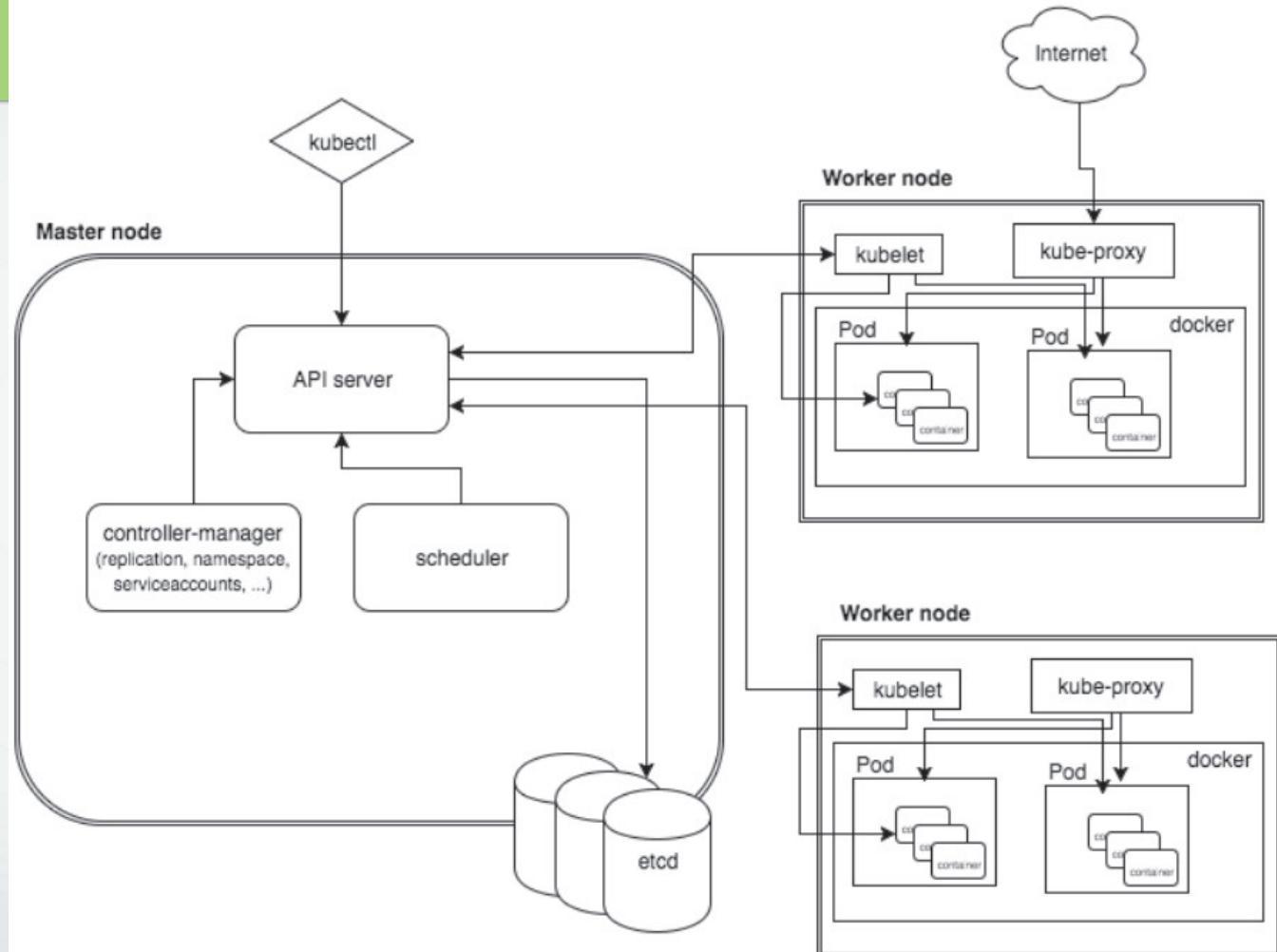
Work node (like the node with docker swarm)

We have a **pod** (in which we have n containers , its like a task with docker swarm)

To be able to expose a container to the outside world, we create a **service**

Replicas set like the replicas with docker swarm

Kubectl command to interact with the api server



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

I : Les composants des nodes

Kubernetes suit l'architecture **maître-esclave**, le maître plus communément appelé **master** existe principalement pour gérer votre cluster Kubernetes. Les **esclaves** sont quant à eux plus connus sous le nom de **workers** (on les appellent aussi **minions**) et ne sont là que pour fournir de la capacité et n'ont pas le pouvoir d'ordonner à une autre node ce qu'il peut ou ne peut pas faire. Les composants clés du **master** et **worker** sont définis dans la section suivante.

1- Les composants du Master :

Sur un node de type master, nous aurons les composants suivants :

- **kube-apiserver**: point d'entrée exposant l'API HTTP Rest de k8s depuis le maître du cluster Kubernetes. Différents outils et bibliothèques peuvent facilement communiquer avec l'API. [\[expose k8s API\]](#)
- **kube-scheduler**: Il est responsable de la répartition et l'utilisation de la charge de travail sur les nœuds du cluster selon les ressources nécessaires et celles disponibles. [\[selects Nodes for Pods to run on\]](#)
- **kube-controller-manager** : ce composant est responsable de la plupart des collecteurs qui récupèrent des informations du cluster tout en effectuant des actions de correctives en cas de besoin, en apportant des modifications pour amener l'état actuel du serveur à l'état souhaité. Il est composé de plusieurs contrôleurs, on peut par exemple retrouver un contrôleur de réPLICATION qui va s'assurer que vous avez le nombre désiré de répliques sur vos pods, mais aussi d'autres contrôleurs clés comme , le contrôleur de Endpoints, le contrôleur d'espace de noms et le contrôleur de compte de service. [\[several controllers to manage cluster\]](#)
- **etcd** : il stocke les informations de configuration pouvant être utilisées par chacun des nœuds du cluster. Ces informations sont conservées sous forme de clé et valeurs à haute disponibilité, [\[store k8s cluster data\]](#)



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

2- Les composants du Workers

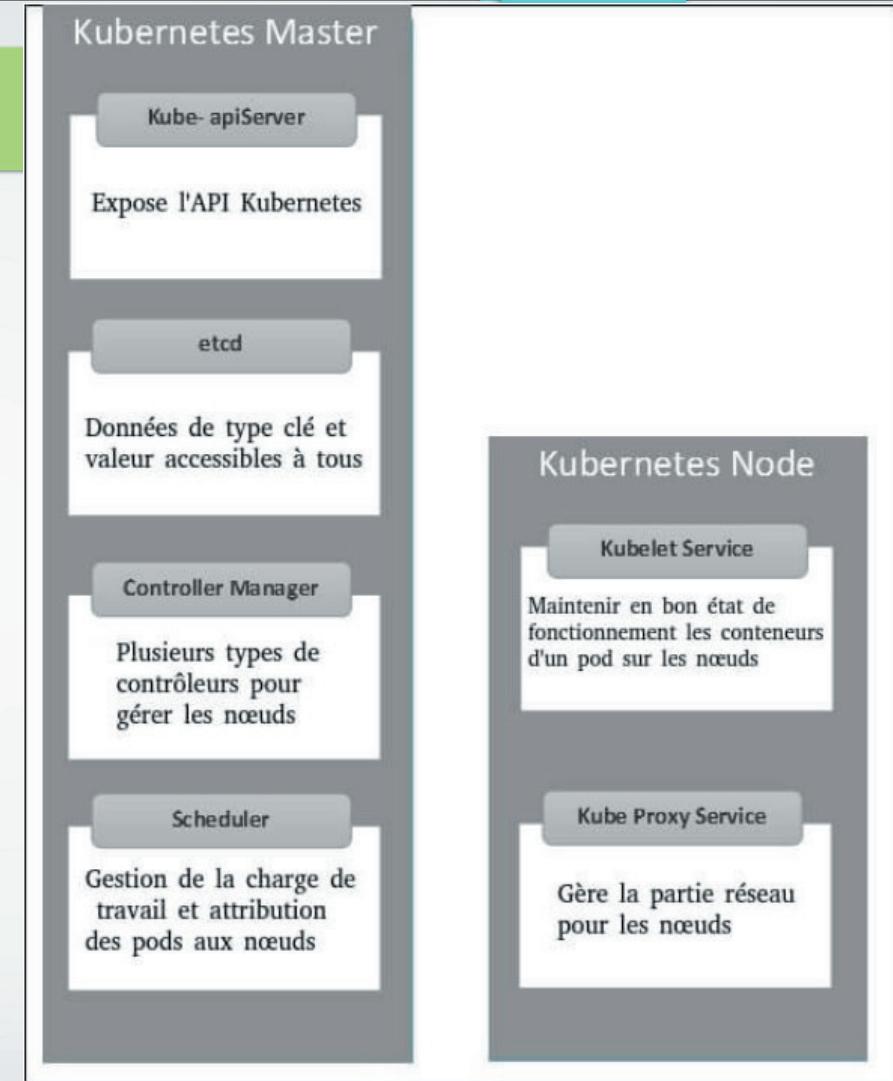
Sur chaque worker, nous aurons les composants suivants :

- **kubelet** : Il s'agit d'un agent qui s'exécute dans chaque nœud chargé de relayer les informations au Master. Il interagit avec la base de données etcd du Master pour récupérer des informations afin de connaître les tâches à effectuer. Il assume la responsabilité de maintenir en bon état de fonctionnement les conteneurs d'un pod et s'assure qu'ils tournent conformément à la spécification. Il ne gère pas les conteneurs qui n'ont pas été créés par Kubernetes. Il communique avec le Master et redémarre le conteneur défaillant en cas de crash. [\[node agent that manages assigned pods\]](#)
- **kube-proxy** : il active l'abstraction réseau du Service Kubernetes en maintenant les règles du réseau et permet l'exposition des services vers l'extérieur.

[\[Networking, connection forwarding\]](#)

- **Docker** : environnement d'exécution de conteneurs.

[\[for running containers « not part of kubernetes »\]](#)



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

II : Communication avec le cluster

1- kubectl (ligne de commande)

L'outil en ligne de commande Kubernetes, kubectl, vous permet d'exécuter des commandes sur les clusters Kubernetes afin de créer et gérer des objets k8s et interagir avec l'API Kubernetes. Vous pouvez utiliser kubectl pour déployer des applications, inspecter et gérer les ressources du cluster et afficher les logs.

2- Dashboard (interface web)

Dashboard est l'interface utilisateur Web officielle de Kubernetes. Il fournit une vue d'ensemble des applications en cours d'exécution sur votre cluster Kubernetes. Il permet aux administrateurs de surveiller et gérer les opérations de leurs clusters Kubernetes, dépanner les applications conteneurisées et gérer les ressources du cluster.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Déploiement d'un cluster k8s avec kubeadm

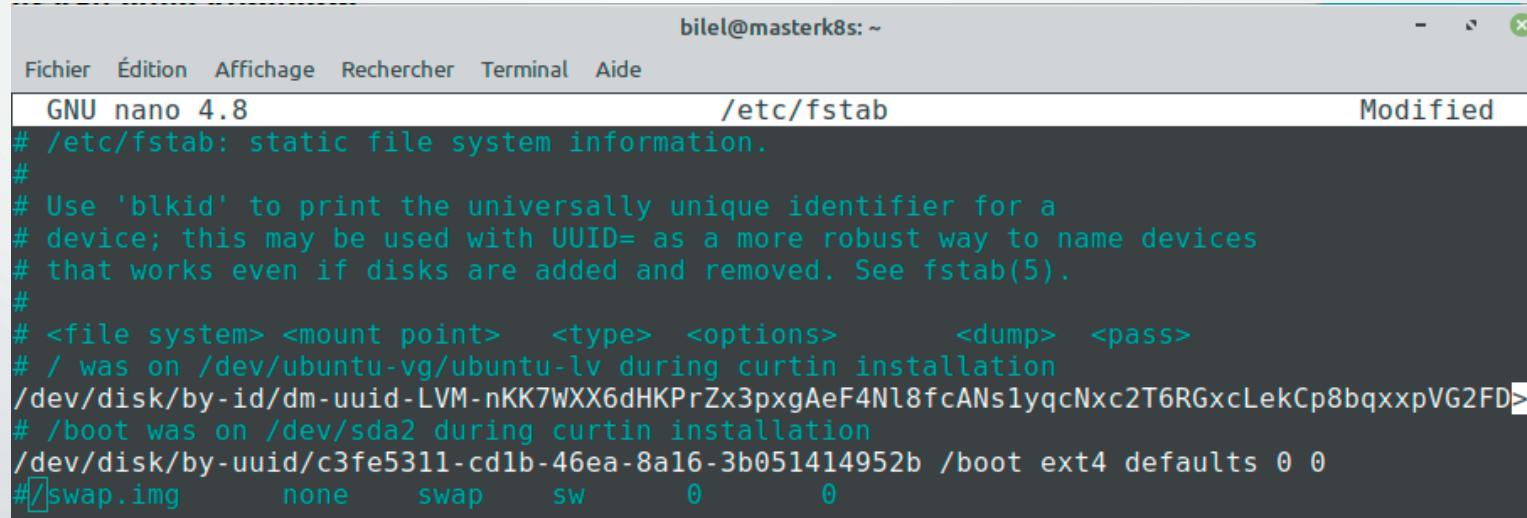
On va créer un cluster k8s avec un Master unique: <https://kubernetes.io/fr/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

1- installer Docker

```
$ sudo apt install docker.io  
$ sudo systemctl enable docker
```

2- désactiver le Swap : commenter la ligne Swap défini dans la fichier fstab

```
$ sudo nano /etc/fstab  
$ sudo swapoff -a
```



bilel@masterk8s: ~

Fichier Édition Affichage Rechercher Terminal Aide

GNU nano 4.8	/etc/fstab	Modified
# /etc/fstab: static file system information.		
#		
# Use 'blkid' to print the universally unique identifier for a		
# device; this may be used with UUID= as a more robust way to name devices		
# that works even if disks are added and removed. See fstab(5).		
#		
# <file system> <mount point> <type> <options> <dump> <pass>		
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation		
/dev/disk/by-id/dm-uuid-LVM-nKK7WXX6dHKPrZx3pxgAeF4Nl8fcANS1yqcNxc2T6RGxcLekCp8bqxxpVG2FD>		
# /boot was on /dev/sda2 during curtin installation		
/dev/disk/by-uuid/c3fe5311-cd1b-46ea-8a16-3b051414952b /boot ext4 defaults 0 0		
#/swap.img none swap sw 0 0		



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

3- Redémarrer le serveur

4- Installation de kubeadm, kubelet et kubectl

```
$ sudo apt-get update && sudo apt-get install -y apt-transport-https curl
```

```
$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
```

```
$ cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install -y kubelet kubeadm kubectl
```

```
$ sudo apt-mark hold kubelet kubeadm kubectl
```

5- Initialiser votre master

```
$ sudo kubeadm init
```

```
Your Kubernetes control-plane has initialized successfully!
```

```
To start using your cluster, you need to run the following as a regular user:
```

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
Alternatively, if you are the root user, you can run:
```

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

```
You should now deploy a pod network to the cluster.
```

```
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/
```

```
Then you can join any number of worker nodes by running the following on each as root:
```

```
kubeadm join 192.168.1.152:6443 --token uz8e75.9o2sbyyvuh2cg1bv \
  --discovery-token-ca-cert-hash sha256:1fb0bec63ca45e68dc9b10f88df78a29818984fe4e4378f2df73c2ae4b6944d0
bilel@masterk8s:~$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Pour que kubectl fonctionne pour votre utilisateur non root, exécutez ces commandes, qui font également partie du résultat de la commande kubeadm init

```
$ mkdir -p $HOME/.kube  
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Alternativement, si vous êtes root, vous pouvez exécuter:

```
$ export KUBECONFIG=/etc/kubernetes/admin.conf
```

6- Installation d'un add-on réseau : Vous devez installer un add-on réseau pour pod afin que vos pods puissent communiquer les uns avec les autres. Le réseau doit être déployé avant toute application. De plus, CoreDNS ne démarrera pas avant l'installation d'un réseau.

Plusieurs projets fournissent des réseaux de pod Kubernetes utilisant CNI, dont certains supportent les network policies. Allez voir la page des add-ons pour une liste complète des add-ons réseau disponibles. <https://kubernetes.io/docs/concepts/cluster-administration/addons/>

On choisie Weave : <https://www.weave.works/docs/net/latest/kubernetes/kube-addon/>

```
kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"
```

```
bilel@masterk8s:~$ kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64  
| tr -d '\n')"  
serviceaccount/weave-net created  
clusterrole.rbac.authorization.k8s.io/weave-net created  
clusterrolebinding.rbac.authorization.k8s.io/weave-net created  
role.rbac.authorization.k8s.io/weave-net created  
rolebinding.rbac.authorization.k8s.io/weave-net created  
daemonset.apps/weave-net created  
bilel@masterk8s:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Faites un enregistrement du retour de la commande **kubeadm join** que **kubeadm init** génère. Vous avez besoin de cette commande pour joindre des noeuds à votre cluster.

```
Then you can join any number of worker nodes by running the following on each as root:
```

```
kubeadm join 192.168.1.152:6443 --token uz8e75.9o2sbyyvuh2cg1bv \
--discovery-token-ca-cert-hash sha256:1fb0bec63ca45e68dc9b10f88df78a29818984fe4e4378f2df73c2ae4b6944d0
```

Le jeton est utilisé pour l'authentification mutuelle entre le master et les nœuds qui veulent le rejoindre. Le jeton est secret. Gardez-le en sécurité, parce que n'importe qui avec ce jeton peut ajouter des nœuds authentifiés à votre cluster. Ces jetons peuvent être listés, créés et supprimés avec la commande **kubeadm token**.

⇒ Maintenant notre Master kubernetes est prêt. récupérerons alors la liste des nœuds disponibles dans notre cluster : a l'aide de commande **kubectl get nodes**

```
bilel@masterk8s:~$ kubectl get node
NAME      STATUS   ROLES          AGE      VERSION
masterk8s  Ready    control-plane,master  7h24m   v1.20.4
bilel@masterk8s:~$
```

⇒ Notre Cluster k8s contient un seul nœud « Master »

Par défaut, votre cluster ne déploie pas de pods sur le master pour des raisons de sécurité. Si vous souhaitez pouvoir déployer des pods sur le master, par exemple, pour un cluster Kubernetes mono-machine pour le développement, exécutez:

```
kubectl taint nodes --all node-role.kubernetes.io/master-
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Les objets Kubernetes

Kubernetes a pour objectif principal de dissimuler la complexité de la gestion d'un parc de conteneurs, et pour cela, différents objets existent pour vous faciliter la vie, dont notamment :

- **Node** : Un node ("nœud" en fr) est une machine de travail du cluster Kubernetes. Ce sont des unités de travail qui peuvent être physiques, virtuelles mais aussi des instances cloud.
- **Pod** : Il s'agit de l'unité la plus petite de K8s, un pod encapsule le ou les conteneur(s) formant votre application conteneurisée partageant ainsi la même stack réseau (chaque pod se voit attribuer une adresse IP unique) et le même stockage, plus précisément un volume partagé (tous les conteneurs du pod peuvent accéder aux volumes partagés, ce qui permet à ces conteneurs de partager plus facilement des données).
- **Replicas** : c'est le nombre d'instances d'un Pod ("réplique" en fr)
- **ReplicaSet** : s'assure que les répliques spécifiés sont actifs
- **Deployment** : définit l'état désiré et fournit des mises à jour déclaratives de vos Pods et ReplicaSets.
- **Service** : Un service peut être défini comme un ensemble logique de pods exposés en tant que service réseau. C'est un niveau d'abstraction au-dessus du pod, qui fournit une adresse IP et un nom DNS unique pour un ensemble de pods. Avec les Services, il est très facile de gérer la configuration de Load Balancing (équilibrage de charge) permettant ainsi aux pods de scaler plus facilement.
- **Endpoint** : Représente l'adresse IP et le port d'un service, il est automatiquement créé lors de la création d'un service avec les pods correspondants.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

1- Gérer et manipuler les Pods Kubernetes

Dans ce chapitre, nous verrons comment créer, gérer, supprimer, déboguer nos Pods k8s.

Nous utiliserons un template au format YAML afin de créer la configuration de notre pod, puis on exécutera la commande **kubectl create** qui permet de créer un objet Kubernetes, dans notre cas ça sera un objet de type Pod.

Primo, il faut savoir que pour chaque objet Kubernetes que vous souhaitez créer, vous devez définir des valeurs pour les champs suivants :

- **apiVersion** : la version de l'API Kubernetes que vous utilisez pour créer cet objet .
- **kind** : le type d'objet k8s que vous comptez créer.
- **metadata** : données de type clé valeur permettant d'identifier de manière unique l'objet.
- **spec** : contient la spécification avec des champs imbriqués propres à chaque objet k8S. Le format est donc différent pour chaque objet Kubernetes.

Dans l'exemple ci-dessous nous créerons un Pod contenant un conteneur de l'image nginx.

Commençons donc par créer un fichier yaml nommé **nginx-pod.yaml** et remplissons dedans les champs requis pour la création de notre Pod :

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    type: web
  name: nginx
spec:
  containers:
    - image: nginx
      name: nginx
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Quelques explications s'imposent, de façon à comprendre l'étendue du fichier :

```
apiVersion: v1
kind: Pod
```

Ici, nous utilisons l'api en **version 1** et nous créons un objet Kubernetes de type **Pod**. Pour connaître la version exacte pour chaque objet, vous pouvez utiliser la commande suivante : **kubectl explain « objet »**

```
bilel@masterk8s:~$ kubectl explain pod
KIND:     Pod
VERSION:  v1

DESCRIPTION:
  Pod is a collection of containers that can run on a host. This resource is
  created by clients and scheduled onto hosts.

FIELDS:
  apiVersion  <string>
    APIVersion defines the versioned schema of this representation of an
    object. Servers should convert recognized schemas to the latest internal
    value, and may reject unrecognized values. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

  kind <string>
    Kind is a string value representing the REST resource this object
    represents. Servers may infer this from the endpoint the client submits
    requests to. Cannot be updated. In CamelCase. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

  metadata    <Object>
    Standard object's metadata. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

  spec <Object>
    Specification of the desired behavior of the pod. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

```
metadata:
  labels:
    type: web
  name: nginx
```

Dans cette partie, nous désignons un label avec une clé nommée **type** et sa valeur **web** (vous pouvez rajouter n'importe quelle clé et valeur), cette donnée n'est pas obligatoire mais sachez juste qu'elle peut nous servir plus tard dans d'autres objets Kubernetes afin de sélectionner et manipuler comme bon nous semble tous les pods contenant ce label. Par la suite on ne fait que nommer notre Pod **nginx**.

```
spec:
  containers:
    - image: nginx
      name: nginx
```

Dans cette phase, nous indiquons la configuration de notre Pod, plus précisément nous spécifions la configuration du conteneur du Pod en question. Très exactement nous désignons un conteneur utilisant l'image **nginx** et qu'on nomme **nginx**

Maintenant, nous allons créer notre pod Kubernetes avec la commande suivante : **kubectl create -f nginx-pod.yaml**

```
bilel@masterk8s:~$ vi nginx-pod.yaml
bilel@masterk8s:~$ kubectl create -f nginx-pod.yaml
pod/nginx created
bilel@masterk8s:~$
```

Vérifions maintenant la liste des pods disponibles dans notre cluster Kubernetes avec la commande **kubectl get pods**

```
bilel@masterk8s:~$ kubectl get pods
NAME     READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          2m2s
bilel@masterk8s:~$
```

Bon ok, nous voyons bien qu'il existe seulement un seul pod disponible dans notre cluster k8s. Cela dit, il serait plus intéressant d'en plus, récupérer l'IP du pod mais aussi le nœud qui héberge ce pod. Pour cela, nous rajouterais l'option **-o wide** :

```
bilel@masterk8s:~$ kubectl get pods -o wide
NAME     READY   STATUS    RESTARTS   AGE       IP           NODE
nginx   1/1     Running   0          4m48s   10.32.0.4   masterk8s
bilel@masterk8s:~$
```

D'après le résultat, notre Pod nginx est situé dans le nœud **masterk8s** et possède l'ip **10,32,0,4**



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Il faut savoir que l'ip du Pod n'est accessible qu'à partir du notre noeud sur lequel il est hébergé (dans notre cas c'est le noeud Masterk8s). Nous verrons dans la partie dédiée aux Services, à comment rendre notre pod accessible depuis l'extérieur de notre cluster.

Essayons alors d'accéder à la page web de notre pod nginx depuis notre noeud

```
bilel@masterk8s:~$ kubectl get pods -o wide
NAME    READY   STATUS    RESTARTS   AGE     IP          NODE    NOMINATED-NODE
nginx   1/1     Running   0          9m51s   10.32.0.4   masterk8s   <none>
bilel@masterk8s:~$ curl http://10.32.0.4
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
bilel@masterk8s:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Vous pouvez aussi exécuter une commande directement dans le conteneur de votre Pod, grâce à la commande suivante :

```
bilel@masterk8s:~$ kubectl exec nginx cat /usr/share/nginx/html/index.html
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD]
-- [COMMAND] instead.
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
bilel@masterk8s:~$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel,
il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Comme sur Docker vous pouvez utiliser les options -t et -i afin de vous connecter directement sur le shell du conteneur de votre Pod :

```
bilel@masterk8s:~$ kubectl -ti exec nginx sh
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD]
-- [COMMAND] instead.
# pwd
/
# bash
root@nginx:/# cat /usr/share/nginx/html/index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Voici un commande utile, qui peuvent notamment vous aider à debuger vos pods.

la commande `kubectl logs nginx` permet de vérifier les logs des conteneurs de vos pods

```
bilel@masterk8s:~$ kubectl logs nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
10.32.0.1 - - [01/Mar/2021:12:52:44 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.68.0" "-"
bilel@masterk8s:~$
```

- Cette commande vient avec quatre options très utiles :
- **-f** : suivre en permanence les logs du conteneur (correspond à un tail -f sur Linux).
- **--tail** : nombre de lignes les plus récentes à afficher.
- **--since=1h** : afficher tous les logs du conteneur au cours de la dernière heure .
- **--timestamps** : afficher la date et l'heure de réception des logs d'un conteneur.

Nous allons supprimer notre pod nginx avec la commande suivante : `kubectl delete pod nginx`

```
bilel@masterk8s:~$ kubectl get po
NAME    READY    STATUS    RESTARTS   AGE
nginx   1/1     Running   0          27h
bilel@masterk8s:~$ kubectl delete po nginx
pod "nginx" deleted
bilel@masterk8s:~$ kubectl get po
No resources found in default namespace.
bilel@masterk8s:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Appliquer des changements sans détruire le Pod

Dans cet exemple, j'ai volontairement rajouté une image invalide dans mon template YAML, ce qui nous donne l'état du pod suivant :

```
bilel@masterk8s:~$ kubectl create -f nginx-pod.yaml
pod/nginx created
bilel@masterk8s:~$ kubectl get po
NAME    READY    STATUS          RESTARTS   AGE
nginx  0/1    ContainerCreating  0          5s
bilel@masterk8s:~$ kubectl get po
NAME    READY    STATUS          RESTARTS   AGE
nginx  0/1    ErrImagePull   0          10s
bilel@masterk8s:~$ 
```

Voici une commande utile, qui peuvent notamment vous aider à debuger vos pods.

la commande **kubectl describe <KUBERNETES OBJECT> <OBJECT NAME>** permet de décrire et de récupérer les informations détaillées de votre pod :

```
node.kubernetes.io/unreachable/noexecutecap-exists-for-500s
Events:
  Type      Reason     Age           From            Message
  ----      -----     --           ----            -----
  Normal    Scheduled  19s          default-scheduler  Successfully assigned default/nginx to masterk8s
  Warning   Failed     13s          kubelet         Failed to pull image "nginx:155": rpc error: code = Unknown desc = Error response from daemon: manifest for nginx:155 not found: manifest unknown: manifest unknown
  Warning   Failed     13s          kubelet         Error: ErrImagePull
  Normal    BackOff    12s          kubelet         Back-off pulling image "nginx:155"
  Warning   Failed     12s          kubelet         Error: ImagePullBackOff
  Normal    Pulling    1s (x2 over 16s)  kubelet         Pulling image "nginx:155"
bilel@masterk8s:~$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Sans la nécessité de supprimer et recréer votre Pod, vous pouvez appliquer des changements d'un Pod depuis votre template YAML avec la commande ci-dessous :

Kubectl apply -f nginx-pod.yaml

```
bilel@masterk8s:~$ kubectl get po
NAME    READY    STATUS      RESTARTS   AGE
nginx  0/1     ErrImagePull  0          2m7s
bilel@masterk8s:~$ nano nginx-pod.yaml
bilel@masterk8s:~$ kubectl apply -f nginx-pod.yaml
Warning: resource pods/nginx is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
pod/nginx configured
bilel@masterk8s:~$ kubectl get po
NAME    READY    STATUS      RESTARTS   AGE
nginx  1/1     Running    0          2m38s
bilel@masterk8s:~$
```

Dans ce chapitre, nous avons vu comment créer, gérer, supprimer, déboguer nos Pods. Mais c'est encore insuffisant, car nous devons étudier dans les chapitres suivants d'autres objets Kubernetes dans l'intention d'exploiter au maximum nos Pods.

Je partage avec vous, un aide-mémoire des commandes de manipulation des Pods :



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

```
# Afficher la liste des pods
kubectl get pods [En option <POD NAME>]
    -o : format de sortie
        wide : afficher l'ip du pod et le nœud qui l'héberge
        yaml : afficher encore plus d'informations sur un pod sous format YAML
        json : afficher encore plus d'informations sur un pod sous format JSON
    --template : récupérer des informations précises de la sortie de la commande
        Récupérer l'IP d'un pod : kubectl get pod <POD NAME> --template={{.status.podIP}}
# Créer un Pod
kubectl create -f <filename.yaml>
# Supprimer un pod
kubectl delete pods <POD NAME>
# Appliquer des nouveaux changements à votre Pod sans le détruire
kubectl apply -f <filename.yaml>
# Afficher les détails d'un pod
kubectl describe pods <POD NAME>
# Exécuter une commande d'un conteneur de votre pod
kubectl exec <POD NAME> -c <CONTAINER NAME> <COMMAND>
    -t : Allouer un pseudo TTY
    -i : Garder un STDIN ouvert
# Afficher les logs d'un conteneur dans un pod
kubectl logs <POD NAME> -c <CONTAINER NAME>
    -f : suivre en permanence les logs du conteneur
    --tail : nombre de lignes les plus récentes à afficher
    --since=1h : afficher tous les logs du conteneur au cours de la dernière heure
    --timestamps : afficher la date et l'heure de réception des logs d'un conteneur
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel,
il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

2- Gérer et manipuler les ReplicaSets Kubernetes

Dans le tuto précédent nous avions étudié la plus petite unité de Kubernetes, à savoir les Pods. C'est bien joli tout ça, mais si vous souhaitez allouer plus de puissance à votre Pod, comment allez-vous vous y prendre ?

Une des solutions reste de scaler votre Pod. Mais pensez-vous que nous allons exécuter la commande permettant de créer un Pod autant de fois que le nombre de répliques voulues ?

La réponse est évidemment NON. Dans ce cas, nous utiliserons un objet Kubernetes, prévue à cet effet, soit les **ReplicaSets**. Les **ReplicaSets** garantissent que le nombre de répliques souhaitées de votre Pod, est bel et bien respecté.

2.1 : Création du ReplicaSet

Dans cet exemple, nous allons créer un ReplicaSet contenant trois répliques du Pod nginx. Comme d'habitude pour créer un Pod, nous passerons par la création d'un template au format YAML, ce qui nous donne ;

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs
  labels:
    app: web
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: nginx
          image: nginx
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Passons à l'explication de ces quelques nouvelles lignes de notre template :

`apiVersion: apps/v1` Dans ce template nous spécifions la version apps/v1 qui est la version de l'API k8s qui prend en charge le concept des **ReplicasSets**.

`replicas: 3`
`selector:`
`matchLabels:`
`app: web`

Ici, nous indiquons le nombre de répliques voulues, soit trois répliques. Et nous définissons aussi la condition de correspondance qui consiste à répliquer les pods ayant le label **app: web**.

Par la suite, on ne fait que définir la configuration de notre Pod, comme vu dans le tuto précédent dédié aux Pods.

Une fois le template fini, on peut commencer à créer notre ReplicaSet avec la commande suivante : `kubectl create -f nginx-rs.yaml`

```
bilel@masterk8s:~$ nano nginx-rs.yaml
bilel@masterk8s:~$ kubectl create -f nginx-rs.yaml
replicaset.apps/nginx-rs created
bilel@masterk8s:~$ █
```

Vérifons ensuite la liste des pods disponibles dans notre cluster

```
bilel@masterk8s:~$ kubectl get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE      IP          NODE
nginx-rs-6k947 1/1     Running   0          64s     10.32.0.5  masterk8s
nginx-rs-dnl9v  1/1     Running   0          64s     10.32.0.4  masterk8s
nginx-rs-tgz88  1/1     Running   0          64s     10.32.0.6  masterk8s
bilel@masterk8s:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

D'après le résultat, nous avons bien nos trois répliques du Pod nginx avec une adresse IP spécifique à chaque Pod.

Nous pouvons dès lors, accéder à la page d'accueil de nginx depuis ces 3 IPS :

```
curl http://10.32.0.5
Curl http://10.32.0.4
curl http://10.32.0.6
```

2.2 : Scaler vos Pods

Il existe différentes manières pour scaler vos Pods. Vous pouvez par exemple scaler les répliques de vos Pods sans changer votre template YAML avec les commandes suivantes :

```
kubectl scale --replicas=4 replicaset nginx-rs OU kubectl scale --replicas=4 -f nginx-rs.yaml
```

Après avoir exécuté la commande de scaling , nous allons cette fois-ci, vérifier la liste des ReplicaSets disponibles dans notre cluster k8s :

```
kubectl get replicaset OU kubectl get rs
```

```
bilel@masterk8s:~$ kubectl scale --replicas=4 -f nginx-rs.yaml
replicaset.apps/nginx-rs scaled
bilel@masterk8s:~$ kubectl get replicaset
NAME      DESIRED   CURRENT   READY   AGE
nginx-rs   4         4         4       16m
bilel@masterk8s:~$ kubectl get rs
NAME      DESIRED   CURRENT   READY   AGE
nginx-rs   4         4         4       18m
bilel@masterk8s:~$ █
```

Le nombre de répliques du Pod nginx est bien passé de trois à quatre.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Contrairement à la première méthode, celle prochaine commande permet d'un côté de diminuer le nombre de répliques de notre Pod, mais nécessite un changement depuis notre template YAML.

Depuis notre template, je vais changer la ligne **replicas: 3** en **replicas: 2**, et lancer la commande suivante : `kubectl apply -f nginx-pod.yaml`

```
bilel@masterk8s:~$ nano nginx-rs.yaml
bilel@masterk8s:~$ kubectl apply -f nginx-rs.yaml
Warning: resource replicaset/nginx-rs is missing the
        annotation required by kubectl apply. kubectl apply should only be used
        with resources annotated with the kubectl.kubernetes.io/last-updated-by annotation or kubectl apply. The missing annotation will be
        replicated to the new replicaset.apps/nginx-rs configured
bilel@masterk8s:~$ kubectl get rs
NAME      DESIRED   CURRENT   READY   AGE
nginx-rs   2         2         2       23m
bilel@masterk8s:~$ 
```

Information

Si jamais vous avez perdu votre template, vous pouvez alors utiliser la commande `kubectl edit rs nginx-rs`, ce qui aura pour effet d'ouvrir l'éditeur de texte de votre terminal contenant la configuration de votre ReplicaSet au format YAML. Pour chaque modification et sauvegarde de ce fichier, les changements seront immédiatement appliqués sur votre ReplicaSet.

```
bilel@masterk8s:~$ kubectl edit rs nginx-rs
replicaset.apps/nginx-rs edited
bilel@masterk8s:~$ kubectl get rs
NAME      DESIRED   CURRENT   READY   AGE
nginx-rs   3         3         2       25m
bilel@masterk8s:~$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

2.3 : Récupérer des informations sur votre ReplicaSet

Comme pour les Pods, vous pouvez récupérer des informations détaillées sur votre ReplicaSet. Plus exactement, ce qui va nous intéresser le plus, ce sont les différents événements subis par notre ReplicaSet, soit la commande suivante : **kubectl describe rs nginx-rs**

Events:					
Type	Reason	Age	From	Message	
Normal	SuccessfulCreate	5s	replicaset-controller	Created pod: nginx-rs-ptwk	
Normal	SuccessfulCreate	5s	replicaset-controller	Created pod: nginx-rs-sxr22	

2.4 : Supprimer votre ReplicaSet

Voici la commande qui permet de supprimer votre ReplicaSet :

```
kubectl delete ReplicaSet nginx-rs
```

Ou

```
kubectl delete rs nginx-rs
```

```
bilel@masterk8s:~$ kubectl delete rs nginx-rs
replicaset.apps "nginx-rs" deleted
bilel@masterk8s:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Information

Pour afficher les SHORTNAMES et les APIVERSION des tous les ressources k8s, il faut utiliser la commande suivante : **kubectl api-resources**

NAME	SHORTNAMES	APIVERSION	NAMESPACED	KIND
bindings		v1	true	Binding
componentstatuses	cs	v1	false	ComponentStatus
configmaps	cm	v1	true	ConfigMap
endpoints	ep	v1	true	Endpoints
events	ev	v1	true	Event
limitranges	limits	v1	true	LimitRange
namespaces	ns	v1	false	Namespace
nodes	no	v1	false	Node
persistentvolumeclaims	pvc	v1	true	PersistentVolumeClaim
persistentvolumes	pv	v1	false	PersistentVolume
pods	po	v1	true	Pod
podtemplates		v1	true	PodTemplate
replicationcontrollers	rc	v1	true	ReplicationController
resourcequotas	quota	v1	true	ResourceQuota
secrets		v1	true	Secret
serviceaccounts	sa	v1	true	ServiceAccount
services	svc	v1	true	Service
mutatingwebhookconfigurations		admissionregistration.k8s.io/v1	false	MutatingWebhookConfiguration
validatingwebhookconfigurations		admissionregistration.k8s.io/v1	false	ValidatingWebhookConfiguration
customresourcedefinitions	crd,crds	apiextensions.k8s.io/v1	false	CustomResourceDefinition
apiservices		apiregistration.k8s.io/v1	false	APIService
controllerrevisions		apps/v1	true	ControllerRevision
daemonsets	ds	apps/v1	true	DaemonSet
deployments	deploy	apps/v1	true	Deployment
replicasets	rs	apps/v1	true	ReplicaSet
statefulsets	sts	apps/v1	true	StatefulSet



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

```
# Afficher la liste des Replicasets
kubectl get replicsets [En option <REPLICASET NAME>]
    --template : récupérer des informations précises de la sortie de la commande
        Ex: Récupérer le nombre de réplique : kubectl get rs <REPLICASET NAME> --template={{.status.replicas}}
# Créer un ReplicaSet
kubectl create -f <template.yaml>
# Supprimer un ReplicaSet
kubectl delete replicsets <REPLICASET NAME>
# Appliquer des nouveaux changements à votre ReplicaSet sans le détruire
kubectl apply -f <template.yaml>
# Modifier et appliquer les changements de votre ReplicaSet instantanément sans le détruire
kubectl edit rs <REPLICASET NAME>
# Afficher les détails d'un replicaset
kubectl describe replicsets <REPLICASET NAME>
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

3 : Gérer et manipuler les Deployments Kubernetes

Jusqu'ici, nous avions étudié deux types d'objets Kubernetes, à savoir les **Pods** et les **ReplicaSets**. Mais, il existe encore un autre niveau plus haut d'abstraction qui permet de gérer ces deux derniers en un seul même objet. Ce nouvel objet Kubernetes se nomme le **Deployment** (en FR : "déploiement"), que nous allons étudier à travers ce chapitre.

3.1 : Créer un Deployment

Créé un **Deployment** depuis un template YAML, afin d'exécuter un **ReplicaSet** de trois **Pods** de type **nginx** :

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
```

Le squelette du template YAML, reste le même que celui des **ReplicaSets** avec le champ **replicas** où nous spécifions le nombre de Pods et le champ **selector** où nous définissons comment le **Deployment** trouve les Pods à gérer. Dans ce cas, il ne générera que les Pods ayant le **label app: nginx**.

Créons maintenant notre Deployment, à l'aide de la commande ci-dessous :

```
kubectl create -f nginx-deploy.yaml
```

Revérifions ensuite la liste de nos Deployments :

```
bilel@masterk8s:~$ nano nginx-deploy.yaml
bilel@masterk8s:~$ kubectl create -f nginx-deploy.yaml
deployment.apps/nginx-deployment created
bilel@masterk8s:~$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  3/3     3           3           26s
bilel@masterk8s:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

```
bilel@masterk8s:~$ kubectl create -f nginx-deploy.yaml
deployment.apps/nginx-deployment created
bilel@masterk8s:~$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           26s
bilel@masterk8s:~$ kubectl get po
NAME                           READY   STATUS    RESTARTS   AGE
nginx-deployment-7848d4b86f-rnfnk   1/1     Running   0          103s
nginx-deployment-7848d4b86f-wgn8g   1/1     Running   0          103s
nginx-deployment-7848d4b86f-x4m74   1/1     Running   0          104s
bilel@masterk8s:~$ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
nginx-deployment-7848d4b86f      3         3         3        112s
bilel@masterk8s:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

3.2 : Mise ajours & Retour en arrière Update & rollback

Vous avez la possibilité de revenir sur une version précédente de votre **Deployment** en utilisant la méthode **kubectl rollout**. Dans cet exemple nous allons mettre à jour notre Deployment nginx-deployment, en mettant le nom de l'image des Pods de notre Deployment en nginx **version 1.9.1** au lieu de la version **latest** :

```
kubectl set image deployment/nginx-deployment nginx=nginx:1.9.1
```

```
bilel@masterk8s:~$ kubectl describe deployments.apps nginx-deployment | grep Image
  Image:      nginx
bilel@masterk8s:~$ kubectl set image deployment/nginx-deployment nginx=nginx:1.9.1
deployment.apps/nginx-deployment image updated
bilel@masterk8s:~$ 
```

Vérifions ensuite l'historique de ce Deployment : `kubectl rollout history deployment/nginx-deployment`

```
bilel@masterk8s:~$ kubectl describe deployments.apps nginx-deployment | grep Image
  Image:      nginx
bilel@masterk8s:~$ kubectl set image deployment/nginx-deployment nginx=nginx:1.9.1
deployment.apps/nginx-deployment image updated
bilel@masterk8s:~$ kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1        <none>
2        <none>

bilel@masterk8s:~$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Nous allons vérifier si les Pods de notre Deployment possèdent la nouvelle version 1.9.1 de l'image nginx :

```
kubectl describe deploy nginx-deployment | grep Image
```

```
bilel@masterk8s:~$ kubectl describe deployments.apps nginx-deployment | grep Image
  Image:      nginx
bilel@masterk8s:~$ kubectl set image deployment/nginx-deployment nginx=nginx:1.9.1
deployment.apps/nginx-deployment image updated
bilel@masterk8s:~$ kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1          <none>
2          <none>

bilel@masterk8s:~$ kubectl describe deploy nginx-deployment | grep Image
  Image:      nginx:1.9.1
bilel@masterk8s:~$ █
```

Supposons, que nous décidons maintenant d'annuler le Deployment actuel et revenir à la révision précédente, soit à la révision numéro 1. Pour ce faire, nous exécuterons la commande suivante :

```
kubectl rollout undo deployment/nginx-deployment --to-revision=1
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Revérifions une nouvelle fois la version de notre image nginx :

```
kubectl describe deploy nginx-deployment | grep Image
```

```
bilel@masterk8s:~$ kubectl describe deployments.apps nginx-deployment | grep Image
  Image:      nginx
bilel@masterk8s:~$ kubectl set image deployment/nginx-deployment nginx=nginx:1.9.1
deployment.apps/nginx-deployment image updated
bilel@masterk8s:~$ kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1          <none>
2          <none>

bilel@masterk8s:~$ kubectl describe deploy nginx-deployment | grep Image
  Image:      nginx:1.9.1
bilel@masterk8s:~$ kubectl rollout undo deployment/nginx-deployment --to-revision=1
deployment.apps/nginx-deployment rolled back
bilel@masterk8s:~$ kubectl describe deploy nginx-deployment | grep Image
  Image:      nginx
bilel@masterk8s:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

3.3 : Mise à l'échelle d'un Deployment

Vous pouvez mettre à l'échelle un Deployment à l'aide de la commande suivante :

```
kubectl scale deployment nginx-deployment --replicas=5
```

```
bilel@masterk8s:~$ kubectl get deployments.apps
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
nginx-deployment  3/3    3           3          52m
bilel@masterk8s:~$ kubectl get po
NAME                               READY  STATUS   RESTARTS  AGE
nginx-deployment-7848d4b86f-psz9z  1/1    Running  0          8m32s
nginx-deployment-7848d4b86f-qr75v  1/1    Running  0          8m51s
nginx-deployment-7848d4b86f-zdg44  1/1    Running  0          8m41s
bilel@masterk8s:~$ kubectl scale deployment nginx-deployment --replicas=5
deployment.apps/nginx-deployment scaled
bilel@masterk8s:~$ kubectl get deployments.apps
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
nginx-deployment  3/5    5           3          53m
bilel@masterk8s:~$ kubectl get deployments.apps
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
nginx-deployment  4/5    5           4          53m
bilel@masterk8s:~$ kubectl get deployments.apps
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
nginx-deployment  4/5    5           4          53m
bilel@masterk8s:~$ kubectl get po
NAME                               READY  STATUS   RESTARTS  AGE
nginx-deployment-7848d4b86f-ksjj9  1/1    Running  0          21s
nginx-deployment-7848d4b86f-p4wjx  1/1    Running  0          21s
nginx-deployment-7848d4b86f-psz9z  1/1    Running  0          9m36s
nginx-deployment-7848d4b86f-qr75v  1/1    Running  0          9m55s
nginx-deployment-7848d4b86f-zdg44  1/1    Running  0          9m45s
bilel@masterk8s:~$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Vous pouvez configurer une mise à l'échelle horizontale automatique des Pods de votre Deployment et choisir le nombre minimal et maximal de Pods que vous souhaitez exécuter en fonction de l'utilisation du processeur de vos Pods existants.

Dans cet exemple nous allons créer un autoscaler (**objet k8s HorizontalPodAutoscaler**) d'un Deployment avec 3 Pods au minimum et 6 Pods au maximum.

```
kubectl autoscale deployment nginx-deployment --min=3 --max=6
```

```
bilel@masterk8s:~$ kubectl autoscale deployment nginx-deployment --min=3 --max=6
horizontalpodautoscaler.autoscaling/nginx-deployment autoscaled
bilel@masterk8s:~$ kubectl get hpa
NAME          REFERENCE          TARGETS      MINPODS   MAXPODS   REPLICAS   AGE
nginx-deployment   Deployment/nginx-deployment   <unknown>/80%   3          6          3          19s
bilel@masterk8s:~$
```

Exemple: <https://kubernetes.io/docs/tasks/run-application/vertical-pod-autoscale-walkthrough/>

3.4 : Supprimer votre Deployment

Voici la commande qui permet de supprimer votre Deployment :

```
kubectl delete deploy nginx-deployment
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

```
# Afficher la liste des Deployments
kubectl get deployment [En option <DEPLOYMENT NAME>]
    -o wide : récupérer en plus, le nom de l'image et le sélecteur
# Créer un Deployment
kubectl create -f <template.yaml>
# Supprimer un Deployment
kubectl delete deployment <DEPLOYMENT NAME>
# Appliquer des nouveaux changements à votre Deployment sans le détruire
kubectl apply -f <template.yaml>
# Modifier et appliquer les changements de votre Deployment instantanément sans le détruire
kubectl edit deployment <DEPLOYMENT NAME>
# Afficher les détails d'un Deployment
kubectl describe deployment <DEPLOYMENT NAME>
# Mettre à jour l'image des Pods de votre Deployment
kubectl set image deployment/<DEPLOYMENT NAME> <CONTAINER NAME>=<NEW IMAGE NAME>
# Afficher le status du rolling update de votre Deployment
kubectl rollout status deployment/<DEPLOYMENT NAME>
# Afficher l'historique des révisions de votre Deployment
kubectl rollout history deployment/<DEPLOYMENT NAME>
# Revenir à une version précédente
kubectl rollout undo deployment/nginx-deployment --to-revision=<REVISION NUMBER>
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

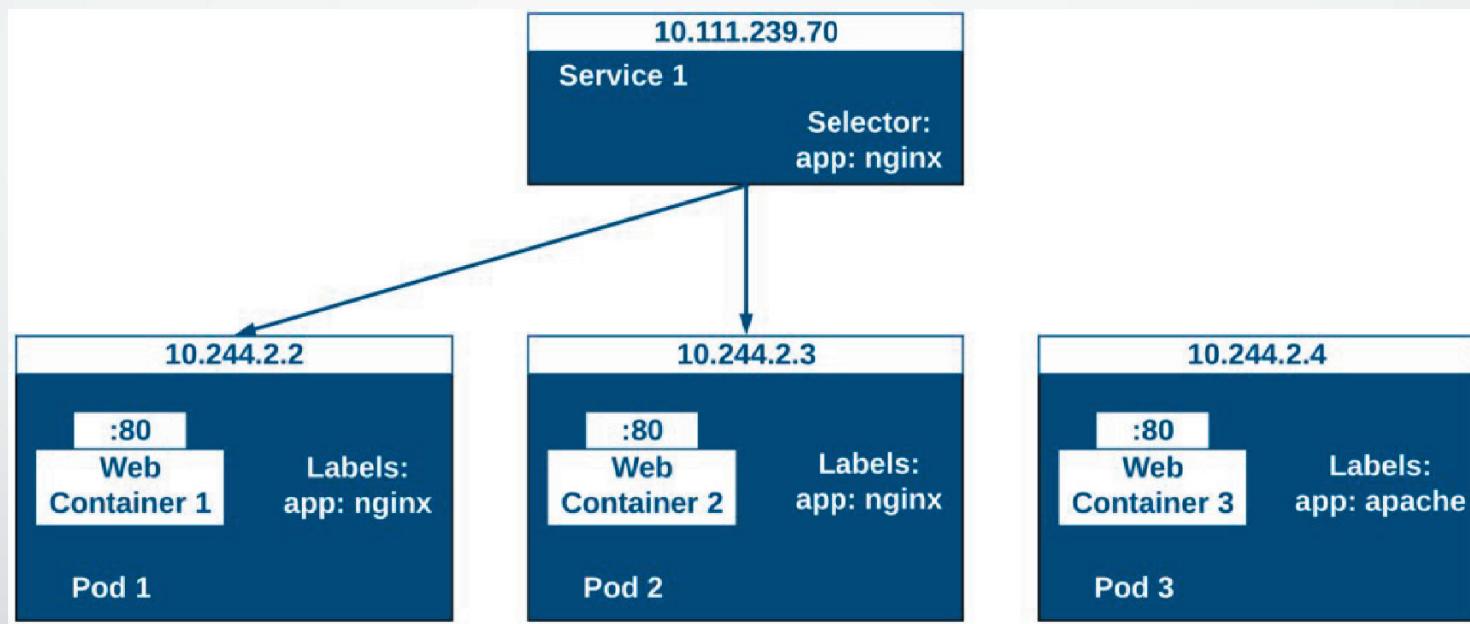
Kubernetes

4 : Gérer et manipuler les Services Kubernetes

Dans ce chapitre, nous verrons comment gérer plus efficacement le réseau de vos Pods grâce aux Services.

Un service peut être défini comme une abstraction par-dessus les Pods, qui attribue aux pods leurs propres adresses IP et un nom DNS unique, et peut aussi équilibrer la charge entre eux,

En effet, les pods ont une durée de vie très limité et sont à maintes reprises créés et détruits, ce qui par conséquent altère continuellement leurs adresses IP. Interviennent alors les services qui vont quant à eux permettre aux clients d'échanger de manière plus fiable avec les conteneurs s'exécutant dans le Pod à l'aide d'une adresse IP virtuelle statique grâce au travail du composant **kube-proxy**.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Il existe plusieurs types de services pour une utilisation particulière :

ClusterIP : C'est le type par défaut. Il expose le Service sur une adresse IP interne du cluster. De ce fait, le service n'est accessible que depuis l'intérieur du cluster.

NodePort : Il expose le service vers l'extérieur du cluster à l'aide du NAT (la plage de ports autorisés est entre 30000 et 32767).

LoadBalancer : Il utilise l'équilibrEUR de charge des fournisseurs de cloud. Ainsi, les services NodePort et ClusterIP sont créés automatiquement et sont acheminés par l'équilibrEUR de charge externe.

Dans ce chapitre nous n'étudierons que le type ClusterIP et NodePort qui sont les plus communément utilisés.

4.1 : ClusterIP

Nous allons commencer par manipuler le service ClusterIP, utilisé par défaut sur Kubernetes.

Pour ce chapitre, j'utiliserai ma propre image **issaoui/b swarm** avec le tag **k8s** qui permet d'afficher depuis une requête http, le nom du conteneur ce qui aura pour but d'analyser plus facilement l'acheminement du trafic effectué par l'agent **kube-proxy**.

Créons d'abord notre template sous le format YAML de manière à construire notre propre service, avec le contenu suivant :

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: ClusterIP
  selector:
    app: myapp
  ports:
  - port: 5000
    targetPort: 5000
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Place maintenant aux explications :

type: ClusterIP c'est ici que nous définissons le type de notre service.

selector:
app: myapp Ici, nous indiquons que le service myapp-service ne sera utilisé que par les Pods ayant le label app: myapp.

- port: 5000
targetPort: 5000 Dans cette phase, le **targetPort** est le port qui est utilisé par les Pods de notre Deployment et le **port** est le numéro de port qui sera utilisé par notre service afin de communiquer avec lui depuis l'intérieur de notre cluster K8.

nous allons exécuter la commande suivante dans le but de créer notre Service Kubernetes : **kubectl create -f service.yaml**

Vérifions ensuite la liste des services disponibles dans notre cluster Kubernetes : **kubectl get svc**

```
bilel@masterk8s:~$ nano service.yaml
bilel@masterk8s:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP     8d
bilel@masterk8s:~$ kubectl apply -f service.yaml
service/myapp-service created
bilel@masterk8s:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP     8d
myapp-service   ClusterIP   10.104.77.160  <none>        5000/TCP    3s
bilel@masterk8s:~$
```

Nous avons donc le service **kubernetes** qui est créé par défaut par K8s et notre service **myapp-service** qui écoute sur l'IP **10.104.77.160** et sur le port **5000**.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Ensuite, nous construisons notre Deployment qui utilisera mon image **issaouib/swarm:k8s** en deux répliques :

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
spec:
  selector:
    matchLabels:
      app: myapp
  replicas: 2
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
        - name: myapp
          image: issaouib/swarm:k8s
```

Comme vous pouvez le constater, nous avons spécifié le label **app: myapp**, afin qu'il soit pris en compte par notre Service **myapp-service**. Dès lors, créons notre Deployment : **kubectl create -f my-deployment.yaml** récupérons le nom des Pods de notre Deployment : **kubectl get po -o wide**

```
bilel@masterk8s:~$ nano my-deployment.yaml
bilel@masterk8s:~$ kubectl create -f my-deployment.yaml
deployment.apps/my-deployment created
bilel@masterk8s:~$ kubectl get po -o wide
NAME                               READY   STATUS    RESTARTS   AGE     IP
S GATES
my-deployment-687ff59748-249jw   1/1    Running   0          9s     10.32.0.4
my-deployment-687ff59748-ct9w9   1/1    Running   0          9s     10.32.0.5
bilel@masterk8s:~$ █
```

Comme je l'ai précisé plus haut, le service de type ClusterIP n'est pas accessible depuis l'extérieur de notre cluster, je vais donc communiquer avec mon service comme suite : **curl http://[CLUSTER_IP]:5000**



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Soit, dans mon cas :

```
bilel@masterk8s:~$ kubectl get svc
NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes     ClusterIP  10.96.0.1      <none>          443/TCP      8d
myapp-service  ClusterIP  10.104.77.160   <none>          5000/TCP    25m
bilel@masterk8s:~$ kubectl get po -o wide
NAME           READY   STATUS      RESTARTS   AGE   IP           NODE
S GATES
my-deployment-687ff59748-249jw  1/1     Running     0          11m   10.32.0.4   masterk8s
my-deployment-687ff59748-ct9w9   1/1     Running     0          11m   10.32.0.5   masterk8s
bilel@masterk8s:~$ curl http://10.104.77.160:5000
You are connected to the Pod : my-deployment-687ff59748-249jw
bilel@masterk8s:~$
```

On remarque que c'est le Pod nommé **my-deployment-687ff59748-249jw** qui répond à notre requête depuis le service **myapp-service** sur le port **5000**.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Maintenant utilisons une boucle for afin de vérifier comment l'agent **kube-proxy** achemine notre trafic :

```
for i in {1..5}; do curl -s http://Cluster-IP:5000 && echo ""; done
```

```
bilel@masterk8s:~$ kubectl get svc
NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
kubernetes     ClusterIP  10.96.0.1      <none>          443/TCP       8d
myapp-service  ClusterIP  10.104.77.160  <none>          5000/TCP      29m
bilel@masterk8s:~$ kubectl get po -o wide
NAME                           READY   STATUS    RESTARTS   AGE     IP           NODE
S GATES
my-deployment-687ff59748-249jw  1/1     Running   0          14m    10.32.0.4   masterk8s
my-deployment-687ff59748-ct9w9   1/1     Running   0          14m    10.32.0.5   masterk8s
bilel@masterk8s:~$ for i in {1..5}; do curl -s http://10.104.77.160:5000 && echo ""; done
You are connected to the Pod : my-deployment-687ff59748-249jw
You are connected to the Pod : my-deployment-687ff59748-ct9w9
You are connected to the Pod : my-deployment-687ff59748-ct9w9
You are connected to the Pod : my-deployment-687ff59748-249jw
You are connected to the Pod : my-deployment-687ff59748-249jw
bilel@masterk8s:~$
```

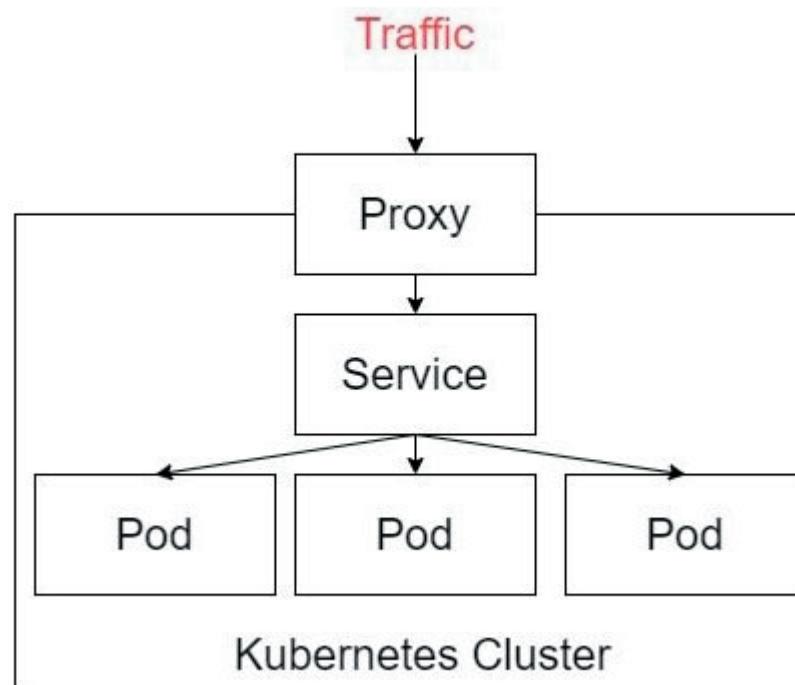
On peut s'apercevoir qu'un LoadBalancer très basique est adopté par l'agent kube-proxy afin de rediriger le trafic sur les différents Pods de notre Deployment.

À partir de maintenant , vous pouvez détruire et recréer vos Pods et communiquer avec eux sereinement, sans vous soucier de leur nouvelle adresse IP, puisque dorénavant vous utilisez l'adresse IP de votre ClusterIP, offrant ainsi une meilleure stabilité.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

4.2 : NodePort

Imaginons, que nous avons besoin cette fois-ci d'accéder à notre Pod depuis l'extérieur de notre cluster Kubernetes. Nous exploiterons ainsi cette le type **NodePort**.

Sans la nécessité de détruire notre ancien service, nous allons modifier et appliquer les changements de notre ancien template YAML, en suivant les étapes suivantes :

- Premièrement modifions notre ancien template YAML, comme suit :

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  selector:
    app: myapp
  ports:
  - port: 5000
    targetPort: 5000
    nodePort: 30080
```

Voici quelques explications sur nos nouveaux changements :

- **type: NodePort** Ici, Nous changeons d'abord le type de notre service.

Ensuite, dans cette étape, nous translatons l'IP du service vers l'IP du nœud qui accueillera nos Pods et le port du service (ici 5000) vers le port 30080.

Information

Pour rappel la plage d'IP dans un service de type NodePort se situe entre 30000 et 32767. On ne peut donc pas dépasser cette limite dans le champ nodePort.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

- Deuxièmement, appliquons les changements de notre service : `kubectl apply -f service.yaml`

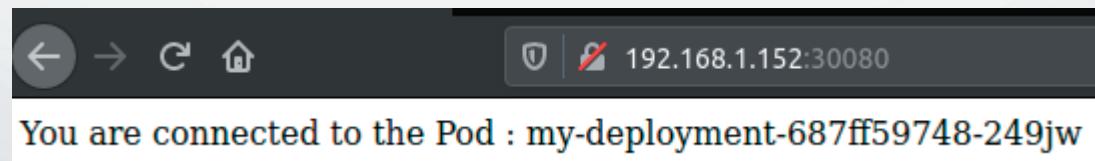
```
bilel@masterk8s:~$ nano service.yaml
bilel@masterk8s:~$ kubectl apply -f service.yaml
service/myapp-service configured
bilel@masterk8s:~$ █
```

Vérifions ensuite la liste des services disponibles dans notre cluster :

```
bilel@masterk8s:~$ kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
kubernetes     ClusterIP  10.96.0.1    <none>        443/TCP       9d
myapp-service  NodePort   10.104.77.160 <none>        5000:30080/TCP 28h
bilel@masterk8s:~$ █
```

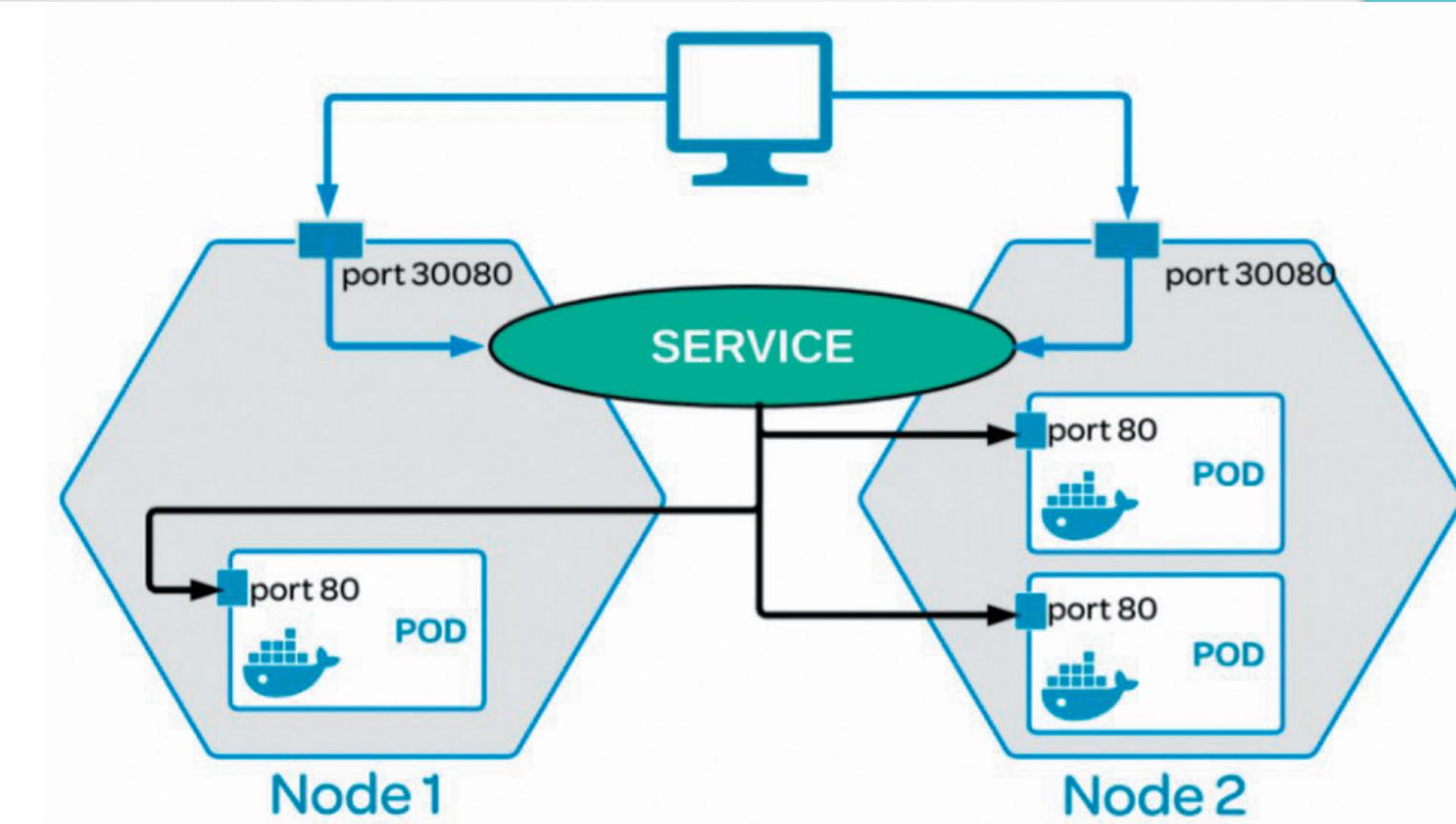
Désormais, nous sommes capables d'accéder à nos Pods depuis l'extérieur de notre cluster à travers l'IP de notre nœud comme suit :

```
bilel@bilel-G3-3590:~$ curl http://192.168.1.152:30080
You are connected to the Pod : my-deployment-687ff59748-249jw
bilel@bilel-G3-3590:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

4.3 : La commande kubectl expose

Plutôt que de rédiger un template YAML de notre service, on peut créer un service à l'aide de la commande **kubectl expose** afin d'exposer les pods de notre Deployment.

Par exemple pour exposer notre Deployment **my-deployment** à l'aide d'un service de type **ClusterIP**, comme vu précédemment, nous suivrons les étapes suivantes :

- D'abord je commence par supprimer mon service Kubernetes **myapp-service**. Pour ce faire nous lançons la commande suivante :

kubectl delete svc myapp-service

```
bilel@masterk8s:~$ kubectl get svc
NAME           TYPE      CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
kubernetes     ClusterIP  10.96.0.1    <none>        443/TCP       9d
myapp-service  NodePort   10.104.77.160 <none>        5000:30080/TCP 28h
bilel@masterk8s:~$ kubectl delete svc myapp-service
service "myapp-service" deleted
bilel@masterk8s:~$ kubectl get svc
NAME           TYPE      CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
kubernetes     ClusterIP  10.96.0.1    <none>        443/TCP       9d
bilel@masterk8s:~$ █
```

Après ceci, on expose les pods de notre Deployment :

```
kubectl expose deployment my-deployment --name myapp-service \
--type ClusterIP --protocol TCP --port 5000 --target-port 5000 --selector='app=myapp'
```

Voici la commande pour exposer notre Deployment avec un service de type NodePort :

```
kubectl expose deployment my-deployment --name myapp-service \
--type NodePort --protocol TCP --port 5000 --target-port 5000 --selector='app=myapp'
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

```
bilel@masterk8s:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes  ClusterIP  10.96.0.1   <none>        443/TCP   9d
bilel@masterk8s:~$ kubectl expose deployment my-deployment --name myapp-service --type ClusterIP --protocol TCP --port 5000 --target-port 5000 --selector='app=myapp'
service/myapp-service exposed
bilel@masterk8s:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes  ClusterIP  10.96.0.1   <none>        443/TCP   9d
myapp-service  ClusterIP  10.96.101.40  <none>        5000/TCP  4s
bilel@masterk8s:~$ kubectl get ep
NAME            ENDPOINTS          AGE
kubernetes      192.168.1.152:6443  9d
myapp-service   10.32.0.2:5000,10.32.0.4:5000  55s
bilel@masterk8s:~$ kubectl get po -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP          NODE     NOMINATED NODE   READINESS GATES
my-deployment-687ff59748-249jw  1/1     Running   1          28h    10.32.0.2  masterk8s  <none>           <none>
my-deployment-687ff59748-ct9w9  1/1     Running   1          28h    10.32.0.4  masterk8s  <none>           <none>
bilel@masterk8s:~$ 
```

```
bilel@masterk8s:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes  ClusterIP  10.96.0.1   <none>        443/TCP   9d
bilel@masterk8s:~$ kubectl expose deployment my-deployment --name myapp-service --type NodePort --protocol TCP --port 5000 --target-port 5000 --selector='app=myapp'
service/myapp-service exposed
bilel@masterk8s:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes  ClusterIP  10.96.0.1   <none>        443/TCP   9d
myapp-service  NodePort   10.102.122.148  <none>        5000:31405/TCP  3s
bilel@masterk8s:~$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

On peut constater que les services restent à bon moyen de gérer plus efficacement la partie réseau de nos Pods. Comme pour chaque fin d'article, voici un antisèche des commandes des services kubernetes :

```
# Afficher la liste des Services
kubectl get service [En option <SERVICE NAME>]
# Créer un Service depuis un template
kubectl create -f <template.yaml>
# Créer un Service depuis sans template
kubectl expose deployment <DEPLOYMENT NAME>
  --name : nom du service
  --type : type du service
  --protocol : protocole à utiliser (TCP/UDP)
  --port : port utilisé par le service
  --target-port : port utilisé utilisé par le Pod
  --selector='clé=valeur': le sélecteur utilisé par service
# Supprimer un Service
kubectl delete service <SERVICE NAME>
# Appliquer des nouveaux changements à votre Service sans le détruire
kubectl apply -f <template.yaml>
# Modifier et appliquer les changements de votre Service instantanément sans le détruire
kubectl edit service <SERVICE NAME>
# Afficher les détails d'un Service
kubectl describe service <SERVICE NAME>
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

5 : Gérer et manipuler les namespaces

Kubernetes prend en charge plusieurs clusters virtuels sauvegardés par le même cluster physique. Ces clusters virtuels sont appelés namespaces.

Un namespace fournit une étendue pour les noms des ressources Kubernetes. Ceci est utile lorsque de nombreux utilisateurs répartis dans plusieurs équipes ou projets utilisent le même cluster et qu'il existe un risque de collision de noms.

5.1 : Les namespaces par défauts

Vous ne le saviez peut-être pas, mais vous utilisez des namespaces depuis le début de cours sans pour autant en être conscient. Afin de vérifier cela, nous allons répertorier la liste des namespaces actuels de votre cluster, en lançant la commande suivante :

```
kubectl get namespace ou kubectl get ns
```

```
bilel@masterk8s:~$ kubectl get ns
NAME      STATUS   AGE
default   Active   9d
kube-node-lease   Active   9d
kube-public   Active   9d
kube-system   Active   9d
bilel@masterk8s:~$ █
```

Comme vous pouvez le constater, notre cluster k8s utilise 4 namespaces par défaut dans Kubernetes, soit :

- **default** : le namespace utilisé par défaut pour les objets Kubernetes dont on ne spécifie pas de namespace.
- **kube-system** : le namespace pour les objets créés et utilisés par le système Kubernetes.
- **kube-public** : ce namespace est créé automatiquement au cas où vous souhaitez que certaines de vos ressources soient visibles publiquement dans l'ensemble du cluster par tous les utilisateurs (y compris ceux non authentifiés).
- **kube-node-lease** : Ce namespace contient les objets de bail associés à chaque nœud, ce qui améliore les performances des pulsations (heartbeats) du nœud à mesure que le cluster évolue.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

5.2 : Créer des namespaces

Il existe deux façons pour créer notre namespace, soit en utilisant un template au format YAML, soit sans.

Dans cet exemple, nous allons créer un namespace nommé **dev-team** sans utiliser de template : **kubectl create namespace dev-team**

```
bilel@masterk8s:~$ kubectl get ns
NAME        STATUS   AGE
default     Active   9d
kube-node-lease  Active   9d
kube-public   Active   9d
kube-system   Active   9d
bilel@masterk8s:~$ kubectl create namespace dev-team
namespace/dev-team created
bilel@masterk8s:~$ kubectl get ns
NAME        STATUS   AGE
default     Active   9d
dev-team    Active   3s
kube-node-lease  Active   9d
kube-public   Active   9d
kube-system   Active   9d
bilel@masterk8s:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Cette fois-ci, nous utiliserons la méthode avec le template au format YAML. Créez donc un fichier yaml et copiez le contenu suivant :

```
apiVersion: v1
kind: Namespace
metadata:
  name: prod-team
```

Ensuite, exécutez la commande suivante : kubectl create -f namespace.yaml

```
bilel@masterk8s:~$ nano namespace.yaml
bilel@masterk8s:~$ kubectl create -f namespace.yaml
namespace/prod-team created
bilel@masterk8s:~$ kubectl get ns
NAME          STATUS   AGE
default        Active   9d
dev-team       Active   2m40s
kube-node-lease Active   9d
kube-public    Active   9d
kube-system    Active   9d
prod-team      Active   42s
bilel@masterk8s:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

5.3 : Utilisation des namespaces dans les objets Kubernetes

Là aussi, il existe deux manières pour utiliser vos ressources Kubernetes dans un namespace. Soit vous définissez dans la liste metadata le champ namespace, comme suit :

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  namespace: prod-team
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
```

Après cela, créez votre objet Kubernetes : **kubectl create -f deploy-prod.yaml**

```
bilel@masterk8s:~$ nano deploy-prod.yaml
bilel@masterk8s:~$ kubectl get all -n prod-team
No resources found in prod-team namespace.
bilel@masterk8s:~$ kubectl create -f deploy-prod.yaml
deployment.apps/nginx created
bilel@masterk8s:~$ kubectl get all -n prod-team
NAME                               READY   STATUS             RESTARTS   AGE
pod/nginx-7848d4b86f-bk9mq       0/1     ContainerCreating   0          6s
pod/nginx-7848d4b86f-s9ls2       0/1     ContainerCreating   0          6s

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx            0/2     2           0           6s

NAME                               DESIRED  CURRENT  READY   AGE
replicaset.apps/nginx-7848d4b86f 2        2        0        6s
bilel@masterk8s:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Soit, il suffit juste de rajouter l'option **-n** ou **--namespace** dans vos commandes kubectl, comme par exemple : cree un template yaml pour un deployments sans spécifier le champ namespace,

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
```

Après cela, créez votre objet Kubernetes : **kubectl -n dev-team create -f deploy-dev.yaml**

```
bilel@masterk8s:~$ nano deploy-dev.yaml
bilel@masterk8s:~$ kubectl -n dev-team create -f deploy-dev.yaml
deployment.apps/nginx created
bilel@masterk8s:~$ kubectl get all -n dev-team
NAME                      READY   STATUS    RESTARTS   AGE
pod/nginx-7848d4b86f-98nfz  1/1    Running   0          15s
pod/nginx-7848d4b86f-hkpr7  1/1    Running   0          15s

NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx  2/2     2           2           15s

NAME                DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-7848d4b86f  2         2         2         15s
bilel@masterk8s:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

5.4 : Définition de la préférence d'un namespace

Vous pouvez enregistrer de manière permanente un namespace pour toutes les commandes kubectl, sans la nécessité d'utiliser l'option **-n** :

```
kubectl config set-context --current --namespace=prod-team
```

```
bilel@masterk8s:~$ kubectl config set-context --current --namespace=prod-team
Context "kubernetes-admin@kubernetes" modified.
bilel@masterk8s:~$
```

Vous pouvez examiner le namespace par défaut grâce à la commande suivante :

```
kubectl config view | grep namespace
```

```
bilel@masterk8s:~$ kubectl config set-context --current --namespace=prod-team
Context "kubernetes-admin@kubernetes" modified.
bilel@masterk8s:~$ kubectl config view | grep namespace
  namespace: prod-team
bilel@masterk8s:~$
```

Voici la commande pour revenir au namespace par défaut :

```
kubectl config set-context --current --namespace=default
```

Si vous n'avez plus besoin de votre namespace, alors vous pouvez le supprimer avec la commande suivante : `kubectl delete namespaces prod-team`



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Tous les objets ne sont pas dans un namespace

La plupart des objets Kubernetes se trouvent dans certains namespaces. Cependant, les objets de bas niveau, tel que les nodes ou les persistentVolumes, ne figurent dans aucun namespace.

Voici la commande qui permet de lister les objets Kubernetes qui sont et ne sont pas dans un namespace : `kubectl api-resources --namespaced=true`

NAME	SHORTNAMES	APIVERSION	NAMESPACED	KIND
bindings		v1	true	Binding
configmaps	cm	v1	true	ConfigMap
endpoints	ep	v1	true	Endpoints
events	ev	v1	true	Event
limitranges	limits	v1	true	LimitRange
persistentvolumeclaims	pvc	v1	true	PersistentVolumeClaim
pods	po	v1	true	Pod

`kubectl api-resources --namespaced=false`

NAME	SHORTNAMES	APIVERSION	NAMESPACED	KIND
componentstatuses	cs	v1	false	ComponentStatus
namespaces	ns	v1	false	Namespace
nodes	no	v1	false	Node
persistentvolumes	pv	v1	false	PersistentVolume



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

6 : La supervision dans kubernetes

Dans ce chapitre nous allons parler du monitoring d'un cluster Kubernetes. Nous allons plus précisément voir comment il est possible de surveiller la consommation des ressources sur Kubernetes.

Il est possible avec kubernetes de surveiller les métriques au niveau des nœuds et pods. On peut par exemple inspecter le nombre de nœuds/pods dans le cluster, le nombre de nœuds/pods sains ainsi que les métriques de performances telles que la consommation de processeur, utilisation de la mémoire, du réseau et du disque.

Nous aurons besoin d'une solution qui surveillera toutes ces métriques et les stockera dans une base de données, tout en fournissant des analyses statistiques autour de ces données. Par défaut, Kubernetes ne propose pas de solution de monitoring intégrée complète. Cependant, il existe de nombreuses solutions open sources disponibles aujourd'hui, telles que Prometheus, Elastic Stack .

Heapster était l'un des projets originaux qui permettait d'activer la supervision et l'analyse de fonctionnalités pour Kubernetes. Vous verrez d'ailleurs beaucoup de références à Hipster lorsque vous recherchez des références sur la surveillance dans kubernetes. Cependant Heapster est maintenant obsolète, une version allégée a été conçue, plus connue sous le nom de Metrics Server.

6.1 : Comment ça marche ?

Le serveur de métriques récupère les mesures de chacun des nœuds et des pods Kubernetes, il les regroupe ensuite et les stocke en mémoire. Notez que le Metric Server n'est qu'une solution de surveillance en mémoire et ne stocke en aucun cas les données récupérées dans le disque. Par conséquent, vous ne pouvez pas voir l'historique des données de performances. Pour cela, vous devez compter sur l'une des solutions de monitoring avancées dont nous avons parlé au tout début de ce chapitre.

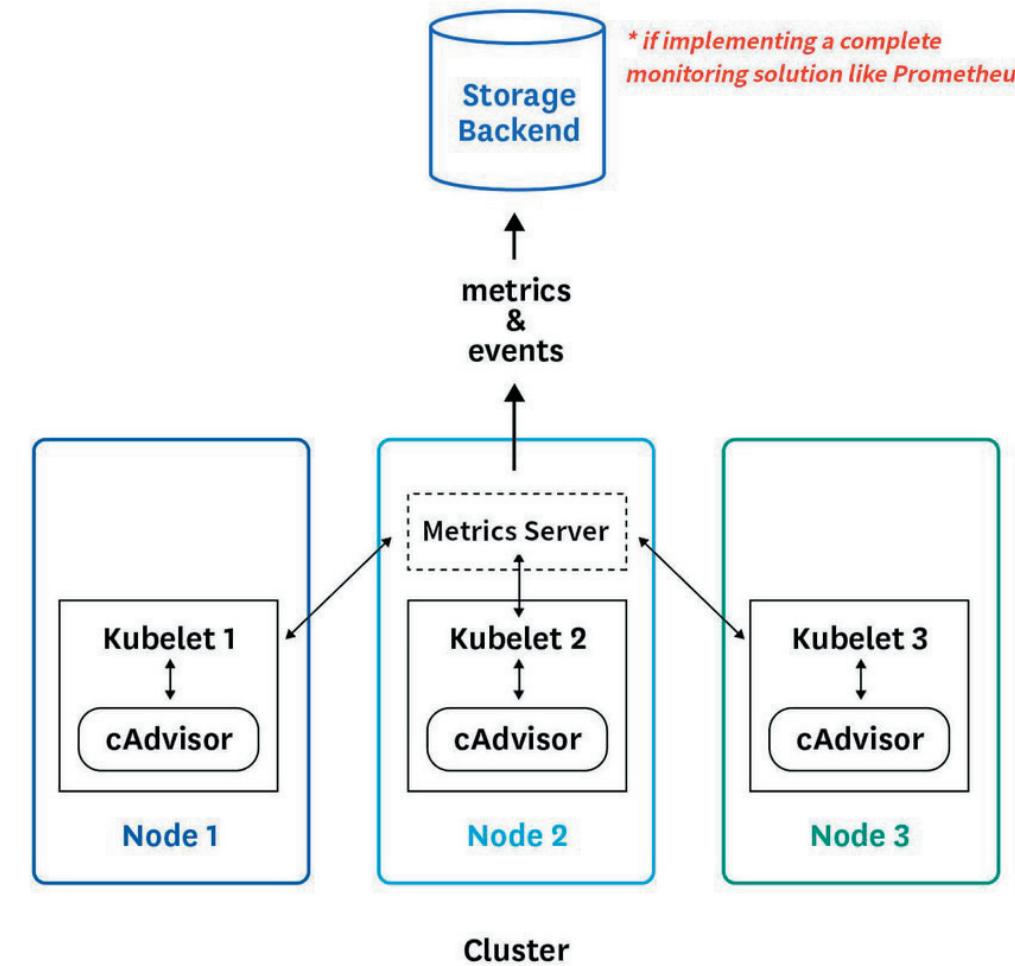
On peut d'ores et déjà se poser la question suivante ? Comment les métriques sont-elles générées pour les POD sur ses nœuds ? En réalité Kubernetes exécute un agent sur chaque nœud connu sous le nom de kubelet. Cet agent est responsable de la réception des ordres du serveur API kubernetes, qui lui demande d'exécuter des pods sur les nœuds disponibles dans le cluster.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

L'agent kubelet contient également un sous composant appelé cAdvisor ou Container Advisor, qui est responsable de récupérer les métriques de performance des pods et les exposer via l'API de kubelet afin de les rendre disponibles pour le serveur de métriques.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

6.2 : Implémentation d'un Metrics Server

- Si vous utilisez minikube, il suffit alors d'exécuter la commande suivante: **minikube addons enable metrics-server**
- Si vous utilisez un cluster k8s avec kubeadm, il suffit alors d'exécuter la commande suivante (appliquée la template yaml du projet git Metrics Server directement sur le nœud master) :

```
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/download/v0.3.6/components.yaml
```

Vérifiez que votre pod de métriques est bien à l'état running :

```
bilel@masterk8s:~$ kubectl get po -n kube-system |grep metrics
metrics-server-6f74f9d985-xgv6h    1/1   Running   0          161m
bilel@masterk8s:~$ kubectl get po -n kube-system
NAME                           READY   STATUS    RESTARTS   AGE
coredns-74ff55c5b-tmgkr        1/1   Running   10         11d
coredns-74ff55c5b-z4dlz        1/1   Running   10         11d
etcd-masterk8s                 1/1   Running   11         11d
kube-apiserver-masterk8s       1/1   Running   15         11d
kube-controller-manager-masterk8s 1/1   Running   17         11d
kube-proxy-wms7f                1/1   Running   11         11d
kube-scheduler-masterk8s       1/1   Running   17         11d
metrics-server-6f74f9d985-xgv6h  1/1   Running   0          161m
weave-net-26ht2                  2/2   Running   22         10d
bilel@masterk8s:~$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

6.2 : Implémentation d'un Metrics Server

- Télécharger le template yaml du projet git Metrics Server

```
wget https://github.com/kubernetes-sigs/metrics-server/releases/download/v0.3.6/components.yaml
```

- Rajoutez les informations ci-dessous au fichier manifeste yaml dans la section **args** sous le **Kind Deployment**

```
--kubelet-insecure-tls=true  
--kubelet-preferred-address-types=InternalIP
```

```
args:  
  - --cert-dir=/tmp  
  - --secure-port=4443  
  - --kubelet-insecure-tls=true  
  - --kubelet-preferred-address-types=InternalIP
```

Voici la définitions des différentes options de metrics-server que vous allez ajouter :

kubelet-insecure-tls=true : ignorer la vérification des certificats Kubelet CA.

kubelet-preferred-address-types=InternalIP : type de communication avec les différents nœuds, dans notre cas on utilise les adresses IP interne.

- vous allez par la suite, créer un ensemble de ressources disponibles dans le template yaml **components** afin de permettre au serveur de métriques d'interroger les métriques de performance des nœuds du cluster.

```
kubectl create -f components.yaml
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

```
bilel@masterk8s:~/monitoring-metric$ nano components.yaml
bilel@masterk8s:~/monitoring-metric$ kubectl create -f components.yaml
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
Warning: apiregistration.k8s.io/v1beta1 APIService is deprecated in v1.19+, unavailable in v1.22+; use apiregistration.k8s.io/v1 APIService
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
serviceaccount/metrics-server created
deployment.apps/metrics-server created
service/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
bilel@masterk8s:~/monitoring-metric$ 
```

Vérifiez que votre pod de métriques est bien à l'état running :

```
bilel@masterk8s:~/monitoring-metric$ kubectl -n kube-system get po
NAME                               READY   STATUS    RESTARTS   AGE
coredns-74ff55c5b-tmgkr          1/1     Running   10         11d
coredns-74ff55c5b-z4dlz          1/1     Running   10         11d
etcd-masterk8s                   1/1     Running   11         11d
kube-apiserver-masterk8s        1/1     Running   15         11d
kube-controller-manager-masterk8s 1/1     Running   17         11d
kube-proxy-wms7f                 1/1     Running   11         11d
kube-scheduler-masterk8s        1/1     Running   17         11d
metrics-server-67c589fb95-mfd88  1/1     Running   0          112s
weave-net-26ht2                  2/2     Running   22         10d
bilel@masterk8s:~/monitoring-metric$ kubectl get po -n kube-system | grep metrics
metrics-server-67c589fb95-mfd88  1/1     Running   0          2m1s
bilel@masterk8s:~/monitoring-metric$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Attendez un peu, le temps de collecter et traiter les données. Une fois traitées, les performances du cluster peuvent être visualisées par les commandes suivantes :

Dans le cas d'une supervision des nœuds : `kubectl top nodes`

```
bilel@masterk8s:~/monitoring-metric$ kubectl top nodes
NAME          CPU(cores)   CPU%    MEMORY(bytes)   MEMORY%
masterk8s     280m        14%    1259Mi          32%
bilel@masterk8s:~/monitoring-metric$ 
```

Dans le cas d'une supervision des pods : Je vais premièrement créer un Deployment avec une réplique de trois pods :

`kubectl create deployment nginx --image=nginx -r=3`

```
bilel@masterk8s:~/monitoring-metric$ kubectl create deployment nginx --image=nginx -r=3
deployment.apps/nginx created
bilel@masterk8s:~/monitoring-metric$ kubectl get deployments.apps
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx    3/3     3           3           75s
bilel@masterk8s:~/monitoring-metric$ kubectl get po
NAME                  READY   STATUS    RESTARTS   AGE
nginx-6799fc88d8-4psxs 1/1     Running   0          80s
nginx-6799fc88d8-52lqn 1/1     Running   0          80s
nginx-6799fc88d8-kf9q8  1/1     Running   0          80s
bilel@masterk8s:~/monitoring-metric$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Deuxièmement, je les supervise avec la commande suivante : `kubectl top pod`

```
bilel@masterk8s:~/monitoring-metric$ kubectl get po
NAME           READY   STATUS    RESTARTS   AGE
nginx-6799fc88d8-4psxs  1/1     Running   0          80s
nginx-6799fc88d8-52lqn  1/1     Running   0          80s
nginx-6799fc88d8-kf9q8  1/1     Running   0          80s
bilel@masterk8s:~/monitoring-metric$ kubectl top po
NAME           CPU(cores)   MEMORY(bytes)
nginx-6799fc88d8-4psxs  0m          2Mi
nginx-6799fc88d8-52lqn  0m          2Mi
nginx-6799fc88d8-kf9q8  0m          2Mi
bilel@masterk8s:~/monitoring-metric$ █
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

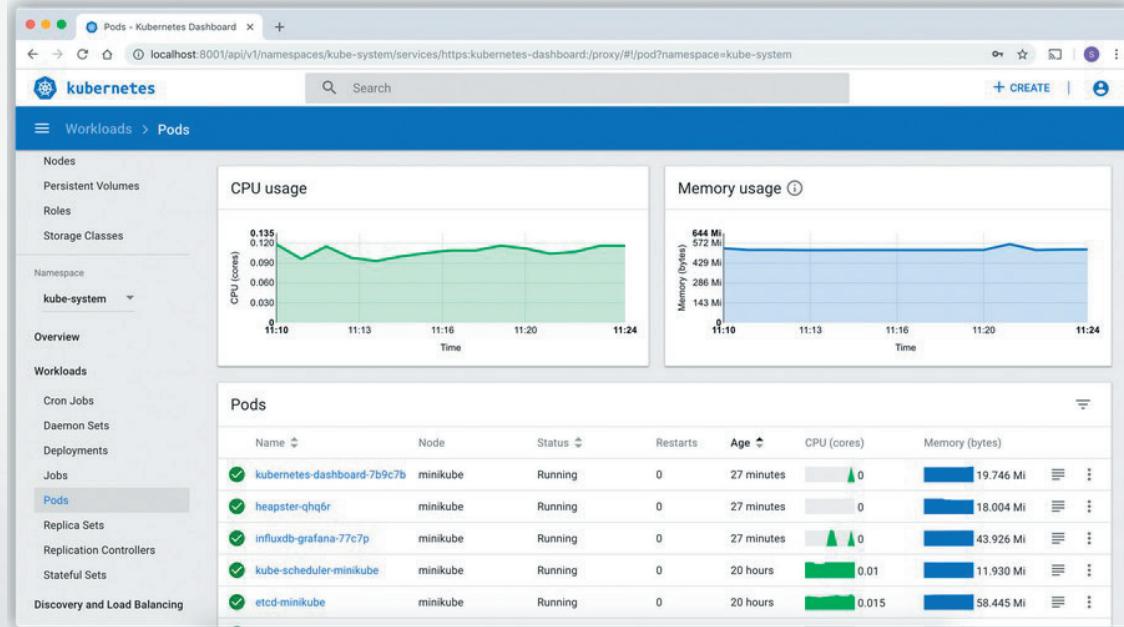
Kubernetes

6.3 : Implémentation du Dashboard K8s (Tableau de Bord)

Le tableau de bord (Dashboard) est une interface web pour Kubernetes. Vous pouvez utiliser ce tableau de bord pour déployer des applications conteneurisées dans un cluster Kubernetes, dépanner votre application conteneurisée et gérer les ressources du cluster. Vous pouvez utiliser le tableau de bord pour obtenir une vue d'ensemble des applications en cours d'exécution dans votre cluster, ainsi que pour créer ou modifier des ressources Kubernetes individuelles. (comme des Deployments, Jobs, DaemonSets, etc). Par exemple, vous pouvez redimensionner un Deployment, lancer une mise à jour progressive, recréer un pod ou déployez de nouvelles applications à l'aide d'un assistant de déploiement.

Le tableau de bord fournit également des informations sur l'état des ressources Kubernetes de votre cluster et sur les erreurs éventuelles.

Tableau de bord Kubernetes



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Déploiement du tableau de bord

L'interface utilisateur du tableau de bord n'est pas déployée par défaut. Pour le déployer, exécutez la commande suivante:

(appliqué le template yaml du projet git Dashboard k8s directement sur le nœud master)

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/master/aio/deploy/recommended.yaml
```

```
bilel@masterk8s:~/monitoring-metric$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/master/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
bilel@masterk8s:~/monitoring-metric$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Affichez les services actuels dans tous les **namespaces**. Notez que le service **kubernetes-dashboard** est de type **ClusterIP**

```
bilel@masterk8s:~/monitoring-metric$ kubectl get svc --all-namespaces
NAMESPACE      NAME          TYPE        CLUSTER-IP      EXTERNAL-IP    PORT(S)        AGE
default        kubernetes   ClusterIP  10.96.0.1     <none>        443/TCP       11d
default        nginx         NodePort   10.104.23.198  <none>        80:31522/TCP  6h57m
kube-system    kube-dns     ClusterIP  10.96.0.10    <none>        53/UDP,53/TCP,9153/TCP 11d
kube-system    metrics-server ClusterIP  10.101.53.165 <none>        443/TCP       4h9m
kubernetes-dashboard dashboard-metrics-scraper ClusterIP  10.105.54.17  <none>        8000/TCP     18m
kubernetes-dashboard kubernetes-dashboard ClusterIP  10.111.147.103 <none>        443/TCP       18m
bilel@masterk8s:~/monitoring-metric$ kubectl get svc -n kubernetes-dashboard
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP    PORT(S)        AGE
dashboard-metrics-scraper ClusterIP  10.105.54.17  <none>        8000/TCP     19m
kubernetes-dashboard     ClusterIP  10.111.147.103 <none>        443/TCP       19m
bilel@masterk8s:~/monitoring-metric$ 
```

Vous ne pouvez accéder à l'interface du tableau de bord k8s que depuis la machine la machine locale à l'aide de commande **kubectl proxy**

Kubectl mettra le tableau de bord à disposition à l'adresse suivante:

<http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/>.

Donc, pour mettre le tableau de bord k8s accessible depuis l'extérieur, il faut faire comme suite :

a- Modifiez le service **kubernetes-dashboard** et changez le type en NodePort : **kubectl -n kubernetes-dashboard edit svc kubernetes-dashboard**

```
bilel@masterk8s:~/monitoring-metric$ kubectl -n kubernetes-dashboard edit svc kubernetes-dashboard
service/kubernetes-dashboard edited
bilel@masterk8s:~/monitoring-metric$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Vérifiez à nouveau le service kubernetes-dashboard. Le type doit apparaître comme **NodePort**. Prenez note du nombre de port, qui est **31931** dans l'exemple ci-dessous. Le vôtre sera différent

```
bilel@masterk8s:~/monitoring-metric$ kubectl get svc -n kubernetes-dashboard
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
dashboard-metrics-scraper  ClusterIP  10.105.54.17 <none>       8000/TCP    19m
kubernetes-dashboard     ClusterIP  10.111.147.103 <none>       443/TCP    19m
bilel@masterk8s:~/monitoring-metric$ kubectl -n kubernetes-dashboard edit svc kubernetes-dashboard
service/kubernetes-dashboard edited
bilel@masterk8s:~/monitoring-metric$ kubectl get svc -n kubernetes-dashboard
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
dashboard-metrics-scraper  ClusterIP  10.105.54.17 <none>       8000/TCP    35m
kubernetes-dashboard     NodePort   10.111.147.103 <none>       443:31931/TCP 35m
bilel@masterk8s:~/monitoring-metric$ 
```

Pour protéger vos données dans le cluster, le tableau de bord prend uniquement en charge la connexion avec un jeton.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

https://192.168.1.152:31931/#/login 130 %

Kubernetes Dashboard

Jeton

Chaque compte de service a un Secret associé avec un jeton porteur (Bearer Token) valide qui peut être utilisé pour se connecter au Dashboard. Pour en savoir plus sur la façon de configurer et utiliser des jetons porteurs, veuillez vous référer à la section [Authentification](#).

Kubeconfig

Veuillez sélectionner le fichier kubeconfig que vous avez créé pour accéder au cluster. Pour en savoir plus sur la façon de configurer et utiliser un fichier kubeconfig, veuillez vous référer à la section [Configurer l'accès à plusieurs clusters](#).

Saisissez un jeton *

Connexion



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Il y a eu quelques problèmes avec RBAC et les autorisations du tableau de bord pour voir les objets. Afin de garantir l'accès à afficher diverses ressources, donnent au tableau de bord un accès administrateur.

```
kubectl create clusterrolebinding dashaccess \
--clusterrole=cluster-admin \
--serviceaccount=kubernetes-dashboard:kubernetes-dashboard
```

```
bilel@masterk8s:~/monitoring-metric$ kubectl create clusterrolebinding dashaccess \
> --clusterrole=cluster-admin \
> --serviceaccount=kubernetes-dashboard:kubernetes-dashboard
clusterrolebinding.rbac.authorization.k8s.io/dashaccess created
bilel@masterk8s:~/monitoring-metric$ kubectl get clusterrolebindings.rbac.authorization.k8s.io
NAME                                     ROLE
cluster-admin                           ClusterRole/cluster-admin
dashaccess                               ClusterRole/cluster-admin
                                         AGE
                                         11d
                                         40s
```

Nous utiliserons la méthode Token pour accéder au tableau de bord. Avec RBAC, nous devons utiliser le jeton **kubernetes-dashboard-token**. Trouvez le jeton, copiez-le puis collez-le dans la page de connexion.

```
bilel@masterk8s:~/monitoring-metric$ kubectl -n kubernetes-dashboard get secrets
NAME                           TYPE           DATA   AGE
default-token-rqfbf          kubernetes.io/service-account-token 3       72m
kubernetes-dashboard-certs   Opaque          0       72m
kubernetes-dashboard-csrf    Opaque          1       72m
kubernetes-dashboard-key-holder Opaque          2       72m
kubernetes-dashboard-token-fx2vr kubernetes.io/service-account-token 3       72m
bilel@masterk8s:~/monitoring-metric$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

```
kubectl -n kubernetes-dashboard describe secrets kubernetes-dashboard-token-
```

```
bilel@masterk8s:~/monitoring-metrics$ kubectl -n kubernetes-dashboard describe secrets kubernetes-dashboard-token-fx2vr
Name:      kubernetes-dashboard-token-fx2vr
Namespace: kubernetes-dashboard
Labels:    <none>
Annotations: kubernetes.io/service-account.name: kubernetes-dashboard
             kubernetes.io/service-account.uid: d9a5a57f-816e-49a5-8640-0bf0fc954c63

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1066 bytes
namespace:  20 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IkUzOTE4dkZRM19xYnhLSGdfUk5wUUFnaxLhRzJfWmN60Xluako4bmthMMif0.eyJpc3MiOiJrdWJlcmlgdGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRLcy5pb9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOjJrdWJlcml5ldGVzLWRhc2hib2FyZCIsImt1YmVybmV0ZXMuaw8vc2VydmljZWfjY291bnQvc2VjcmV0Lm5hbWUiOjJrdWJlcml5ldGVzLWRhc2hib2FyZC10b2tlbiimeDJ2ciIsImt1YmVybmV0ZXMuaw8vc2VydmljZWfjY291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOjJrdWJlcml5ldGVzLWRhc2hib2FyZCIsImt1YmVybmV0ZXMuaw8vc2VydmljZWfjY291bnQvc2VydmljZS1hY2NvdW50LnVpZCI6ImQ5YTVhNTdmLTgxNmUtNDlhNS04NjQwLTBiZjBmYzk1NGM2MyIsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdWJlcml5ldGVzLWRhc2hib2FyZDprdWJlcml5ldGVzLWRhc2hib2FyZCJ9.Rp96sOfCh08wbwQwB5VmP3Dj24tyD5GcIGn4qITK6UFxSYNCdy8gxtPBvob0TOM35dvxnOKCPDvlTU45LR_HBK_rbxU5ErBeXV5KmaP0r8YLy5Vx-f_0aHtt1G5xv-Wc9SvyGqfrbHU8evUK0MMv6e0n3VGELFrGh460TEpm1cvlxqakg63YqEWbm6tKMZw1Pv8h_ETYt-g6_AgsQpD9VXAYu_jZAT7PoDATFXSVUVe-J0ho4TPeedBpobHLFc7Zey-H1H0IKxeAkUsuDAAPxcjmTvv8s5gBXqa1bR3i5xy533YpweXexMScvNrwYHU0HW40ViVRhotgJNh0COVA
bilel@masterk8s:~/monitoring-metrics$
```

Kubernetes Dashboard

Jeton
Chaque compte de service a un Secret associé avec un jeton porteur (Bearer Token) valide qui peut être utilisé pour se connecter au Dashboard. Pour en savoir plus sur la façon de configurer et utiliser des jetons porteurs, veuillez vous référer à la section [Authentification](#).

Kubeconfig
Veuillez sélectionner le fichier kubeconfig que vous avez créé pour accéder au cluster. Pour en savoir plus sur la façon de configurer et utiliser un fichier kubeconfig, veuillez vous référer à la section [Configurer l'accès à plusieurs clusters](#).

Saisissez un jeton *

.....

[Connexion](#)



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes dashboard interface showing the default namespace.

Workloads

- Déploiements: 3/3
- Pods: 2
- Replica Sets: 2

Déploiements

Nom	Espace de nom	Étiquettes	Pods	Date de création	Images
nginx	default	app: nginx	3 / 3	5 hours ago	nginx

Pods

Nom	Espace de nom	Étiquettes	Noeud	Statut	Redémarrages	Utilisation CPU (coeurs)	Utilisation mémoire (octets)	Date de création
nginx-6799fc88d8-4psxs	default	app: nginx, pod-template-hash: 6799fc88d8	masterk8s	Running	1	0,00m	7,29Mi	5 hours ago
nginx-6799fc88d8-52lqn	default	app: nginx, pod-template-hash: 6799fc88d8	masterk8s	Running	1	0,00m	2,89Mi	5 hours ago

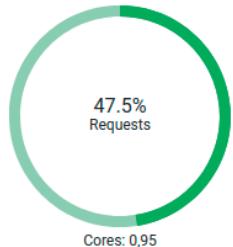


Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Allocation

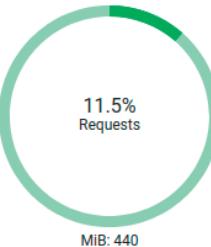
CPU



0.0%
Limits

Cores: 0

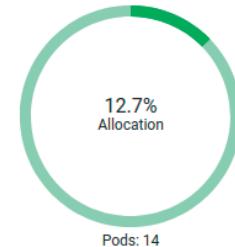
Mémoire



11.5%
Requests

MIB: 440

Pods



12.7%
Allocation

Pods: 14

Conditions

Type	Statut	Dernière sonde	Dernière transition	Motif	Message
NetworkUnavailable	False	4 hours ago	4 hours ago	WeavelsUp	Weave pod has set this
MemoryPressure	False	4 minutes ago	11 days ago	KubeletHasSufficientMemory	kubelet has sufficient memory available
DiskPressure	False	4 minutes ago	11 days ago	KubeletHasNoDiskPressure	kubelet has no disk pressure
PIDPressure	False	4 minutes ago	11 days ago	KubeletHasSufficientPID	kubelet has sufficient PID available
Ready	True	4 minutes ago	11 days ago	KubeletReady	kubelet is posting ready status. AppArmor enabled



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Superviser son cluster Kubernetes avec Prometheus et Grafana

Prometheus est un logiciel de supervision open-source créé par SoundCloud. En 2013, SoundCloud a décidé d'utiliser Prometheus pour ses infrastructures de Production et a publié la version 1.0 en Juillet 2016. [https://fr.wikipedia.org/wiki/Prometheus_\(logiciel\)](https://fr.wikipedia.org/wiki/Prometheus_(logiciel))

Prometheus, écrit en GO, s'impose depuis comme la solution de référence pour superviser une infrastructure de type Cloud, SaaS/Openstack, OKD, K8S.

Plusieurs autres solutions de supervision sont disponibles sur le marché :

- Zabbix ;
- Nagios ;
- Centreon ;
- Sensu ;
- InfluxData.

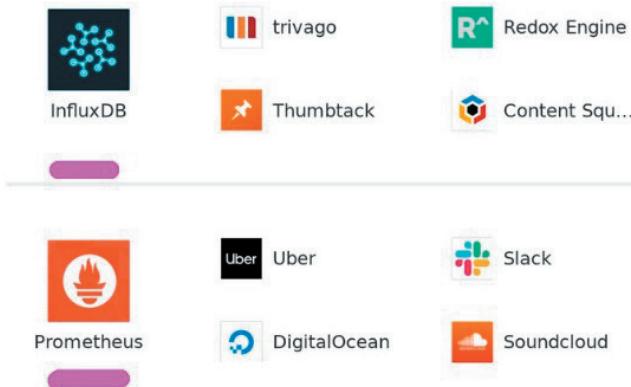
Chaque solution offre ses avantages et ses inconvénients. Prometheus est fourni par défaut comme outil de supervision pour OKD et K8S.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Prometheus vs influxdb(microk8s)



InfluxDB

InfluxDB is a push based system, i.e. your running application needs to actively push data into the monitoring system.

Prometheus

Prometheus is a pull based system, the Prometheus server fetches the metrics values from the running application periodically.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Architecture de Prometheus

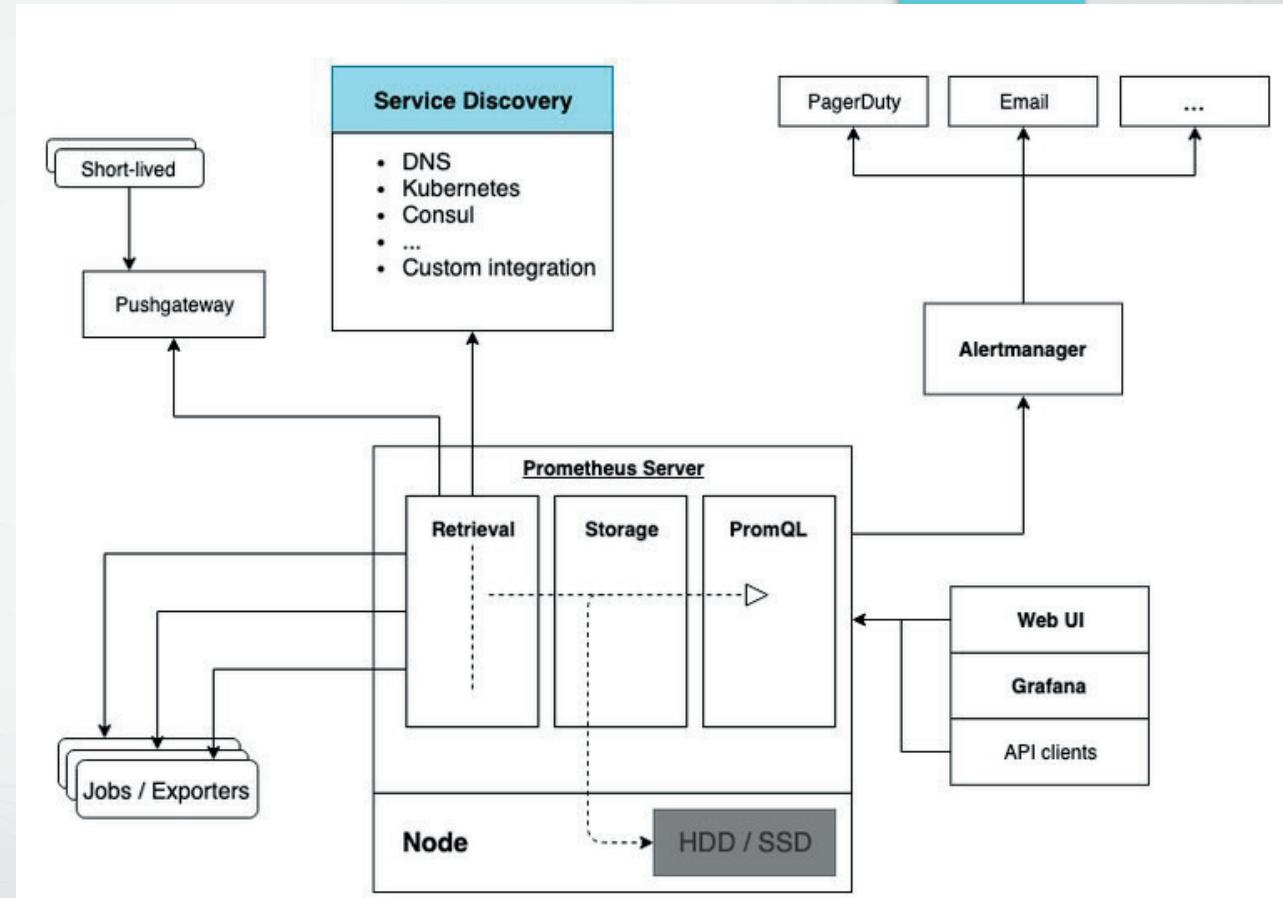
Le modèle de fonctionnement de Prometheus est extensible par construction, basé avant tout sur un modèle de type pull : Prometheus, au travers d'une configuration statique et / ou dynamique (via les service discoveries), interroge régulièrement des exporters qui fournissent des métriques. Il persiste par la suite les données collectées dans une base locale sur disque. Réalisant à la fois les fonctions de collecte, de stockage, et d'exposition des métriques par une API, il est plus simple à installer qu'une architecture basée du InfluxDB, laquelle nécessite des collecteurs en amont.

Prometheus permet, via son API, à des outils de dashboarding tiers (Grafana est l'option la plus répandue) de se connecter. Un langage de requête nommé PromQL permet de sélectionner et traiter des métriques.

Prometheus va se connecter à des targets comme kubernetes, il va récupérer auprès de k8s les k8s metrics

l'opération de récupération de donnée s'appelle **scraping**

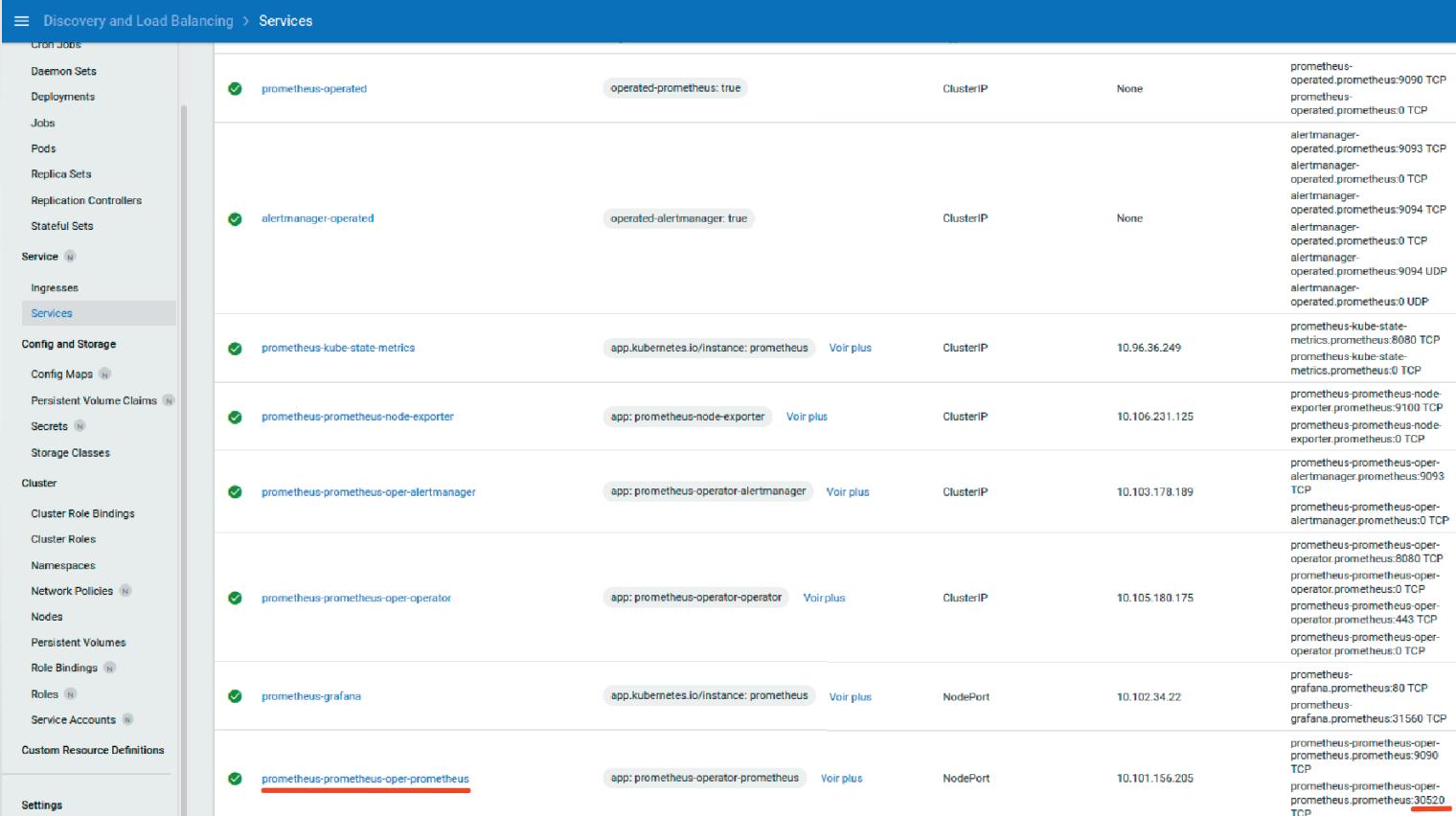
PromQL : prometheus query language



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Après l'installation du Prometheus, on se connecte sur le dashboard k8s a fin de récupérer le port du service `prometheus`



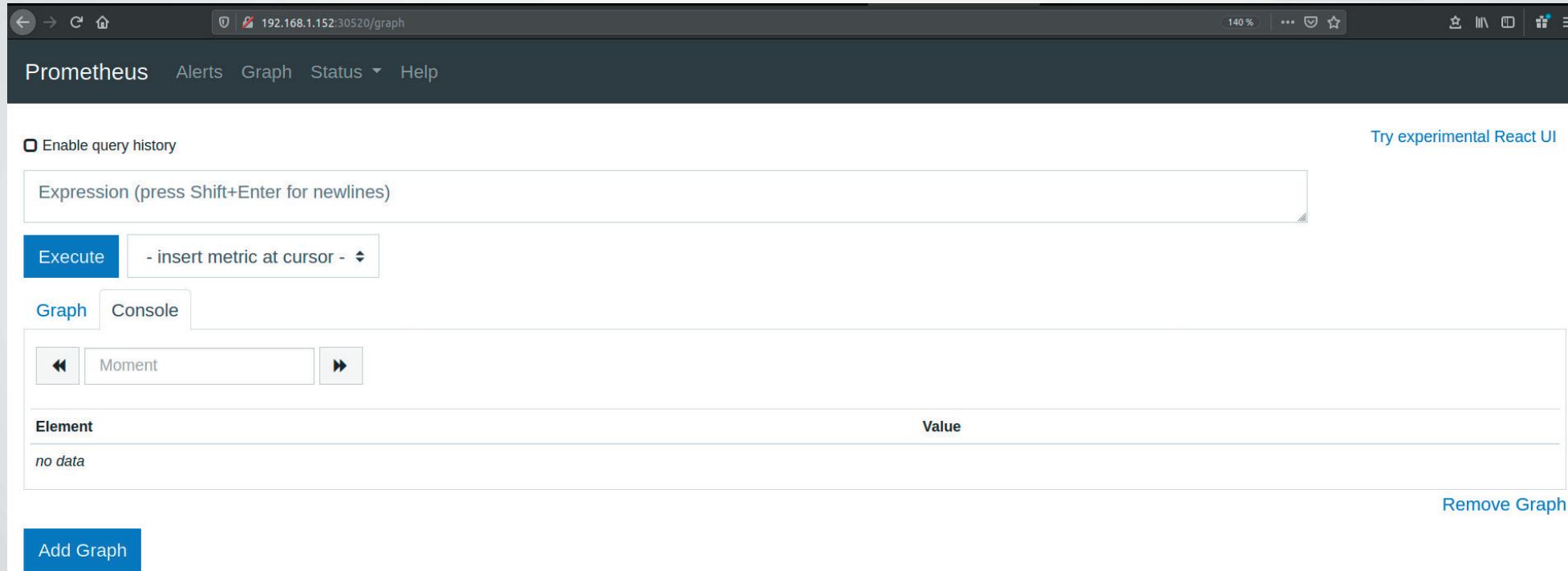
Service	Description	Type	External IP	Ports
<code>prometheus-operated</code>	<code>operated-prometheus: true</code>	ClusterIP	None	<code>prometheus-operated.prometheus:9090 TCP</code> <code>prometheus-operated.prometheus:0 TCP</code>
<code>alertmanager-operated</code>	<code>operated-alertmanager: true</code>	ClusterIP	None	<code>alertmanager-operated.prometheus:9093 TCP</code> <code>alertmanager-operated.prometheus:0 TCP</code> <code>alertmanager-operated.prometheus:9094 TCP</code> <code>alertmanager-operated.prometheus:0 TCP</code> <code>alertmanager-operated.prometheus:9094 UDP</code> <code>alertmanager-operated.prometheus:0 UDP</code>
<code>prometheus-kube-state-metrics</code>	<code>app.kubernetes.io/instance: prometheus</code> Voir plus	ClusterIP	10.96.36.249	<code>prometheus-kube-state-metrics.prometheus:8080 TCP</code> <code>prometheus-kube-state-metrics.prometheus:0 TCP</code>
<code>prometheus-prometheus-node-exporter</code>	<code>app: prometheus-node-exporter</code> Voir plus	ClusterIP	10.106.231.125	<code>prometheus-prometheus-node-exporter.prometheus:9100 TCP</code> <code>prometheus-prometheus-node-exporter.prometheus:0 TCP</code>
<code>prometheus-prometheus-oper-alertmanager</code>	<code>app: prometheus-operator-alertmanager</code> Voir plus	ClusterIP	10.103.178.189	<code>prometheus-prometheus-oper-alertmanager.prometheus:9093 TCP</code> <code>prometheus-prometheus-oper-alertmanager.prometheus:0 TCP</code>
<code>prometheus-prometheus-oper-operator</code>	<code>app: prometheus-operator-operator</code> Voir plus	ClusterIP	10.105.180.175	<code>prometheus-prometheus-oper-operator.prometheus:8080 TCP</code> <code>prometheus-prometheus-oper-operator.prometheus:0 TCP</code> <code>prometheus-prometheus-oper-operator.prometheus:443 TCP</code> <code>prometheus-prometheus-oper-operator.prometheus:0 TCP</code>
<code>prometheus-grafana</code>	<code>app.kubernetes.io/instance: prometheus</code> Voir plus	NodePort	10.102.34.22	<code>prometheus-grafana.prometheus:80 TCP</code> <code>prometheus-grafana.prometheus:31560 TCP</code>
<code>prometheus-prometheus-oper-prometheus</code>	<code>app: prometheus-operator-prometheus</code> Voir plus	NodePort	10.101.156.205	<code>prometheus-prometheus-oper-prometheus.prometheus:9090 TCP</code> <code>prometheus-prometheus-oper-prometheus.prometheus:30520 TCP</code>



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Accéder à l'interface web de Prometheus avec l'adresse ip du serveur k8s et le port du service



The screenshot shows the Prometheus web interface. At the top, there is a navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below the navigation bar, there is a search bar labeled "Expression (press Shift+Enter for newlines)" with a "Execute" button. To the right of the search bar, there is a link to "Try experimental React UI". Below the search bar, there is a "Graph" tab selected, followed by a "Console" tab. Under the "Graph" tab, there is a timestamp "Moment" and a "Remove Graph" button. A table below the timestamp has two columns: "Element" and "Value", both currently showing "no data". At the bottom left of the interface, there is a "Add Graph" button.

Prometheus dispose d'une interface Web simpliste mais toujours utile dans des cas de requêtsages PromQL manuels et graphes simples et pour vérifier le paramétrage du serveur.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Interroger la liste des namespace dans notre cluster kubernetes

```
bilel@masterk8s:~$ kubectl get ns
NAME           STATUS  AGE
default        Active  13d
kube-node-lease  Active  13d
kube-public     Active  13d
kube-system     Active  13d
kubernetes-dashboard  Active  45h
prometheus      Active  26h
bilel@masterk8s:~$
```

Load time: 20ms
 Resolution: 14s
 Total time series: 6

kube_namespace_labels

Execute kube_namespace_labels ↴

Graph Console

Moment

Element	Value
kube_namespace_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-node-lease",pod="prometheus-kube-state-metrics-95d956569-5rmjx",service="prometheus-kube-state-metrics"}	1614253903
kube_namespace_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-public",pod="prometheus-kube-state-metrics-95d956569-5rmjx",service="prometheus-kube-state-metrics"}	1614253903
kube_namespace_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="prometheus-kube-state-metrics-95d956569-5rmjx",service="prometheus-kube-state-metrics"}	1614253903
kube_namespace_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kubernetes-dashboard",pod="prometheus-kube-state-metrics-95d956569-5rmjx",service="prometheus-kube-state-metrics"}	1615230776
kube_namespace_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="prometheus",pod="prometheus-kube-state-metrics-95d956569-5rmjx",service="prometheus-kube-state-metrics"}	1615299806
kube_namespace_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="default",pod="prometheus-kube-state-metrics-95d956569-5rmjx",service="prometheus-kube-state-metrics"}	1614253914



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel,
 il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Lister les pods qui sont dans le namespace kube-system

kube_pod_created

Execute - insert metric at cursor -

Graph **Console**

◀ Moment ▶

Element

```
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="coredns-74ff55c5b-z4dlz",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="weave-net-26ht2",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="prometheus",pod="prometheus-kube-state-metrics-95d956569-5rmjx",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kubernetes-dashboard",pod="kubernetes-dashboard-9f9799597-gcpq8",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="prometheus",pod="prometheus-prometheus-prometheus-oper-prometheus-0",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="kube-controller-manager-masterk8s",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="kube-scheduler-masterk8s",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="kube-proxy-wms7f",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="default",pod="nginx-6799fc88d8-52lqn",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="metrics-server-67c589fb95-mfd88",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="prometheus",pod="prometheus-prometheus-node-exporter-59ghv",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="etcd-masterk8s",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="kube-apiserver-masterk8s",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="prometheus",pod="prometheus-prometheus-oper-operator-6d9c4bdb9f-cqwp5",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="prometheus",pod="prometheus-grafana-7789c77d9d-5h2kb",service="prometheus-kube-state-metrics"]
kube_pod_created[endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="coredns-74ff55c5b-tmgkr",service="prometheus-kube-state-metrics"]
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

`bilel@masterk8s:~$ kubectl get po -n kube-system`

NAME	READY	STATUS	RESTARTS	AGE
coredns-74ff55c5b-tmgkr	1/1	Running	11	13d
coredns-74ff55c5b-z4dlz	1/1	Running	11	13d
etcd-masterk8s	1/1	Running	12	13d
kube-apiserver-masterk8s	1/1	Running	16	13d
kube-controller-manager-masterk8s	1/1	Running	18	13d
kube-proxy-wms7f	1/1	Running	12	13d
kube-scheduler-masterk8s	1/1	Running	18	13d
metrics-server-67c589fb95-mfd88	1/1	Running	1	2d1h
weave-net-26ht2	2/2	Running	24	12d

`bilel@masterk8s:~$`

`kube_pod_created{namespace="kube-system"}`

`Execute`

`Graph`
`Console`


Moment


`Element`

`kube_pod_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="coredns-74ff55c5b-z4dlz",service="prometheus-kube-state-metrics"}`

`kube_pod_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="weave-net-26ht2",service="prometheus-kube-state-metrics"}`

`kube_pod_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="kube-controller-manager-masterk8s",service="prometheus-kube-state-metrics"}`

`kube_pod_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="kube-scheduler-masterk8s",service="prometheus-kube-state-metrics"}`

`kube_pod_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="kube-proxy-wms7f",service="prometheus-kube-state-metrics"}`

`kube_pod_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="metrics-server-67c589fb95-mfd88",service="prometheus-kube-state-metrics"}`

`kube_pod_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="etcd-masterk8s",service="prometheus-kube-state-metrics"}`

`kube_pod_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="kube-apiserver-masterk8s",service="prometheus-kube-state-metrics"}`

`kube_pod_created{endpoint="http",instance="10.32.0.11:8080",job="kube-state-metrics",namespace="kube-system",pod="coredns-74ff55c5b-tmgkr",service="prometheus-kube-state-metrics"}`

`Add Graph`



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Intégration avec Grafana

Prometheus fournit par défaut une interface Web qui permet d'effectuer des requêtes pour voir les données présentes en base, mais cette interface n'est absolument pas pratique pour exploiter et superviser l'infrastructure.

Grafana est un logiciel Open Source pour la visualisation et la supervision d'une infrastructure. Ce logiciel propose une connexion native à Prometheus et propose une liste de dashboards pré-générés pour récupérer les informations en provenance de Prometheus.

connecte sur le dashboard k8s a fin de récupérer le port du service Grafana

 prometheus-grafana	app.kubernetes.io/instance: prometheus	Voir plus	NodePort	10.102.34.22	prometheus-grafana.prometheus:80 TCP prometheus-grafana.prometheus:31560 TCP
--	--	-----------	----------	--------------	---

Accéder a l'interface web de Grafana avec l'adresse ip du serveur k8s et le port du service.

<http://192.168.1.152:31560>

Login par defaul : admin
Password : prom-operator

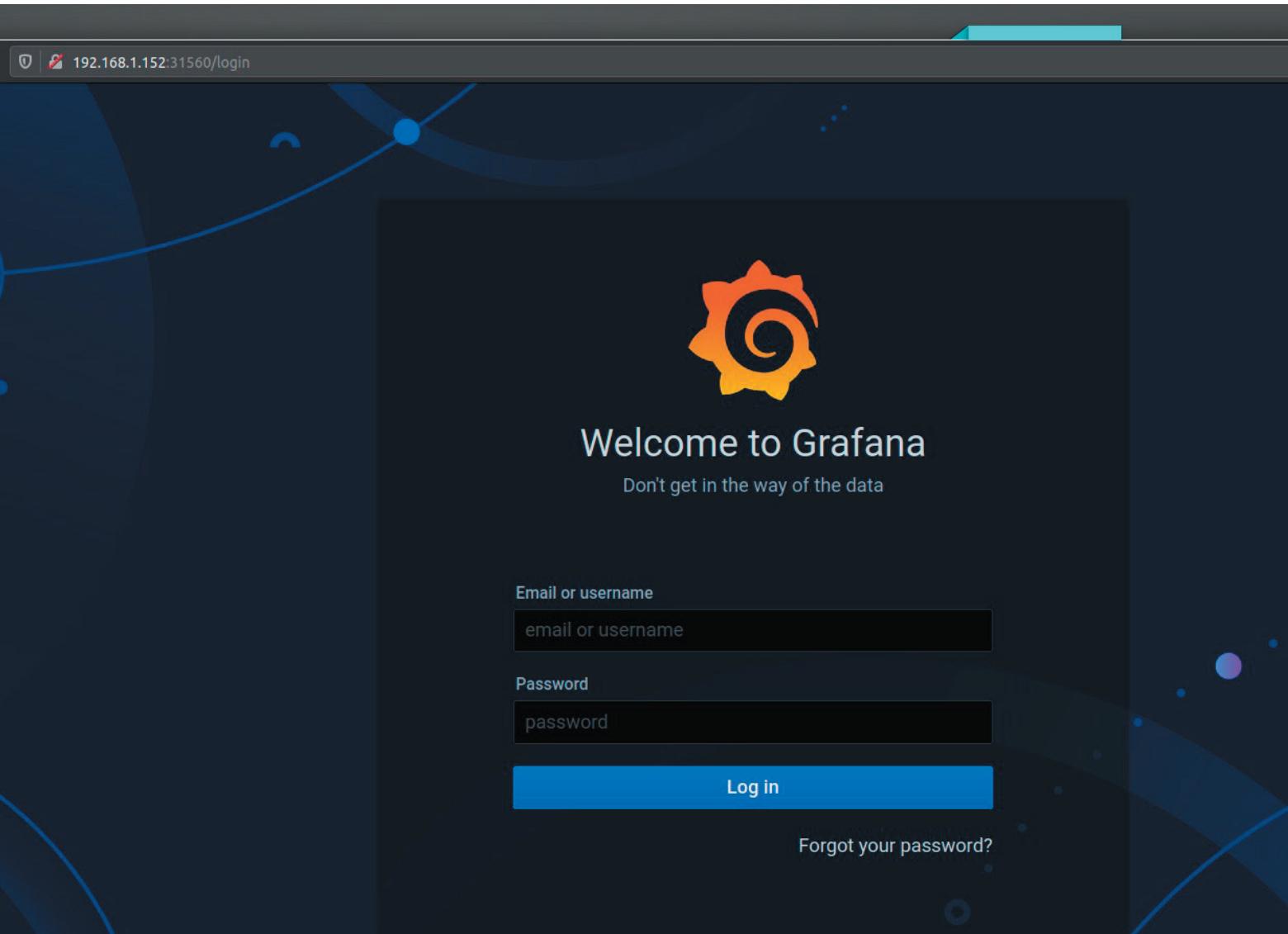
kubectl get secret --namespace prometheus prometheus-grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo

```
bilel@masterk8s:~$ kubectl get secret --namespace prometheus prometheus-grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo  
prom-operator  
bilel@masterk8s:~$ 
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



192.168.1.152:31560/login

Welcome to Grafana

Don't get in the way of the data

Email or username

email or username

Password

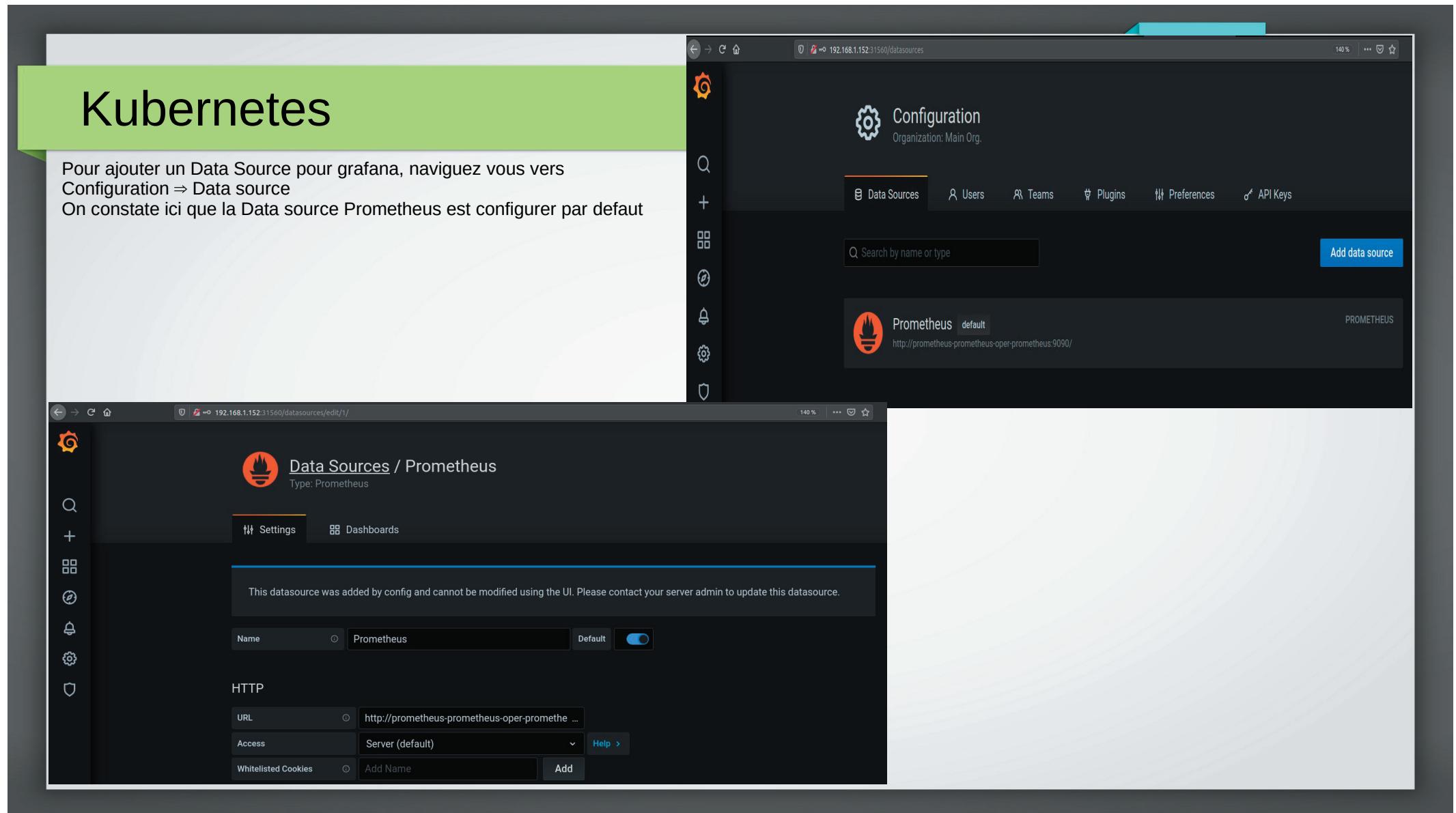
password

Log in

Forgot your password?



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155



The image shows two screenshots of the Grafana interface. The top screenshot is the 'Configuration' page at <http://192.168.1.152:31560/datasources>. It displays a 'Data Sources' tab and a search bar. A single entry for 'Prometheus' is listed, with the URL <http://prometheus-prometheus-oper-prometheus:9090/>. The bottom screenshot is the 'Data Sources / Prometheus' configuration page at <http://192.168.1.152:31560/datasources/edit/1/>. It shows the 'Settings' tab selected, with a note: 'This datasource was added by config and cannot be modified using the UI. Please contact your server admin to update this datasource.' The 'Name' field is set to 'Prometheus', and the 'Default' toggle is on. Under the 'HTTP' section, the 'URL' is set to <http://prometheus-prometheus-oper-promethe...>, 'Access' is set to 'Server (default)', and there are sections for 'Whitelisted Cookies' and 'Add'.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Dashboard par défaut grafana

Dashboards: Manage - Grafana - Mozilla Firefox

192.168.1.152:31560/dashboards

Dashboard

Manage

Playlists

Snapshots

Search dashboards by name

New Dashboard

New Folder

Import

General

- CoreDNS
- etcd
- Kubernetes / API server
- Kubernetes / Compute Resources / Cluster
- Kubernetes / Compute Resources / Namespace (Pods)
- Kubernetes / Compute Resources / Namespace (Workloads)
- Kubernetes / Compute Resources / Node (Pods)
- Kubernetes / Compute Resources / Pod
- Kubernetes / Compute Resources / Workload
- Kubernetes / Controller Manager



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



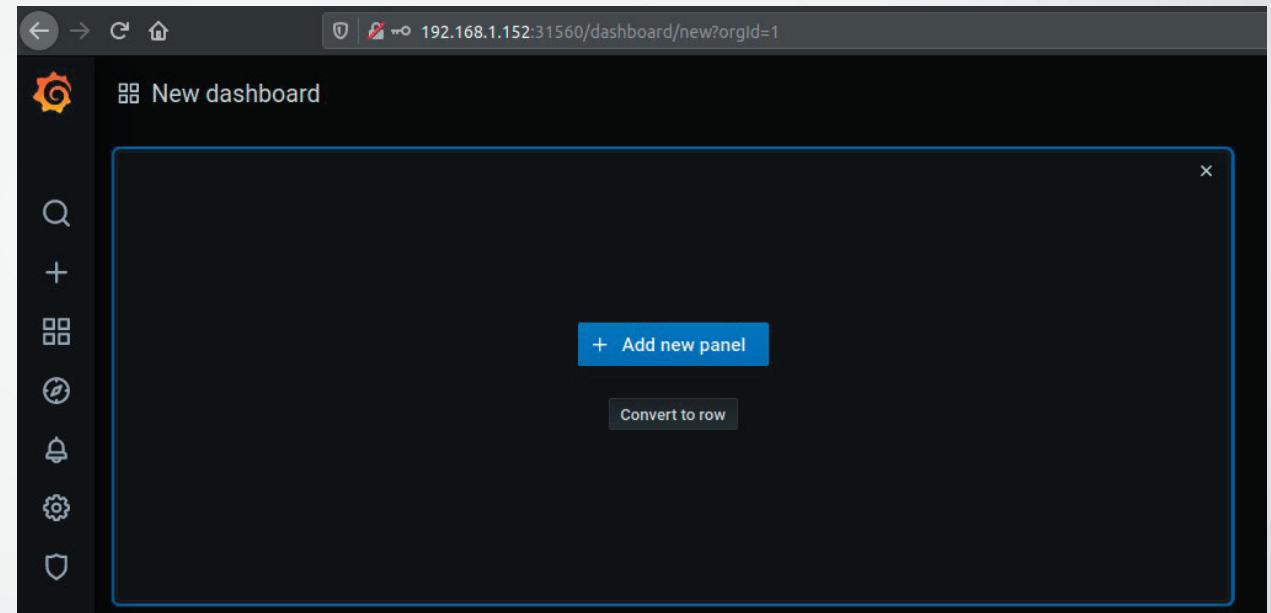
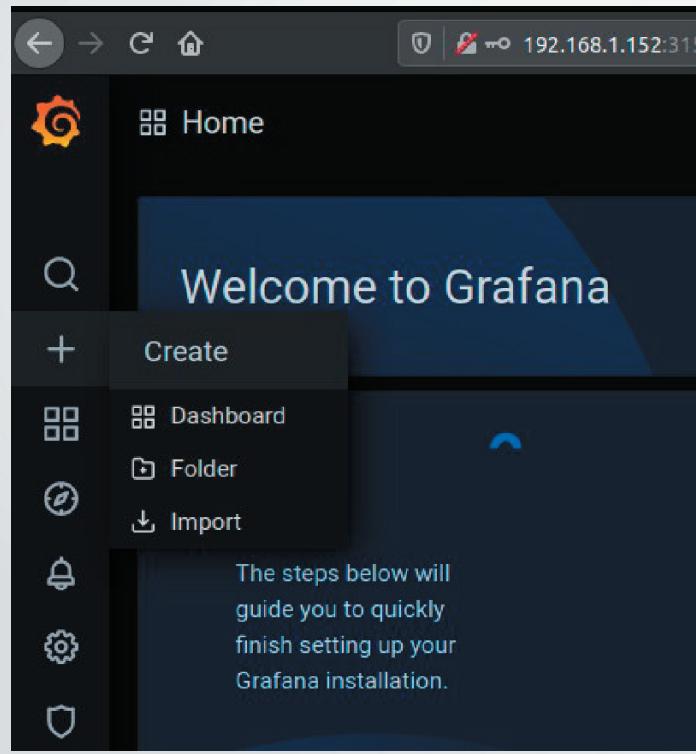
Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Créez vos propres tableaux de bord

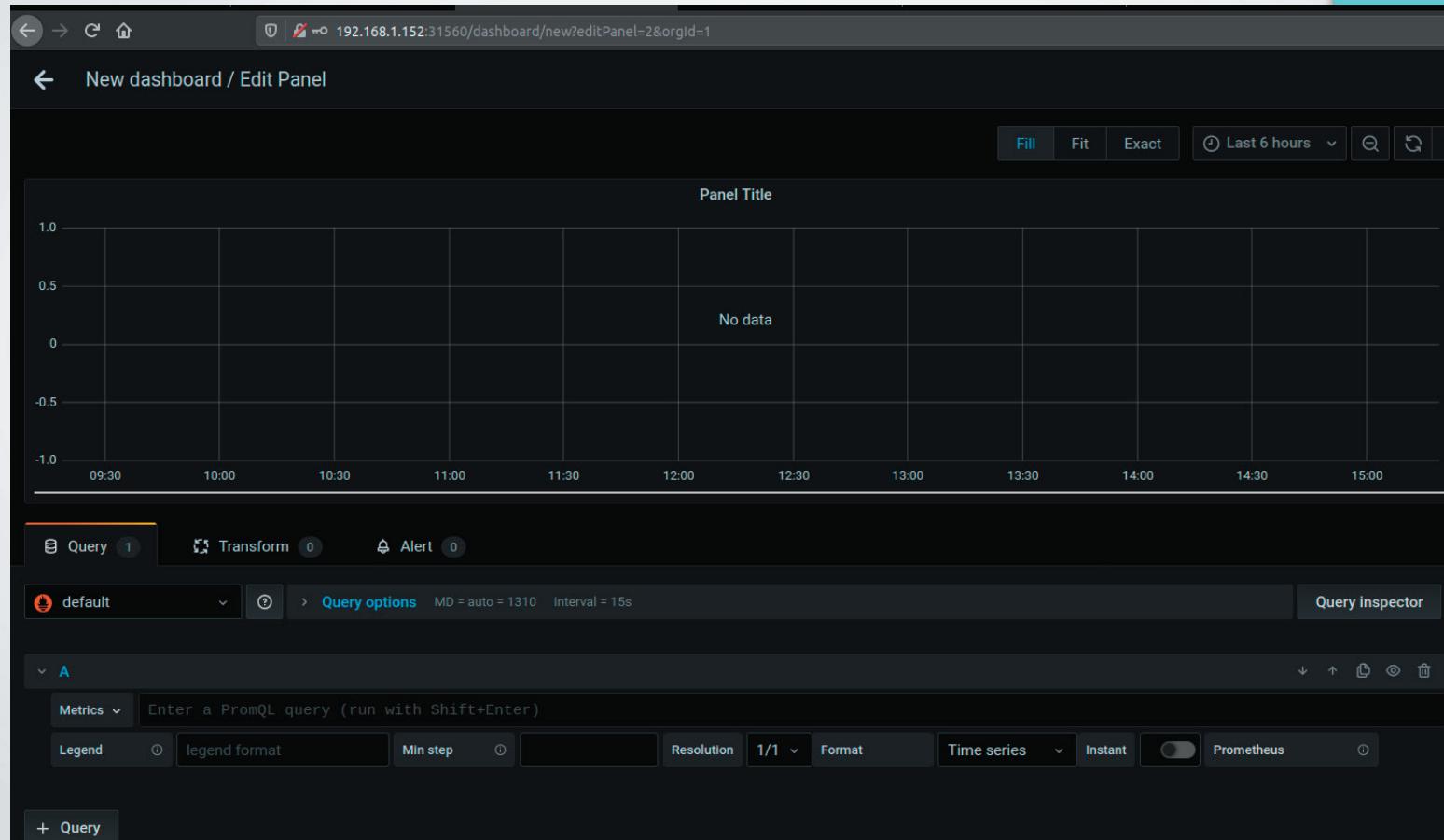
- Explorez avec Grafana la charge du node
- 1- + ==> Create ==> Dashboard

2- Add new Panel



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

3-Promql : node_load1



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

3-legend : {{instance}}



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Nom Dashboard + Auto refresh



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Enregistrer le tableau de bord

New dashboard / Settings

General

Name: my-dashboard

Description:

Tags: add tags

Folder: General

Editable:

Time Options

Timezone: Default

Auto-refresh: 30s

Now delay now: 0m

Hide time picker:

Panel Options

Graph Tooltip: Default

Delete Dashboard



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



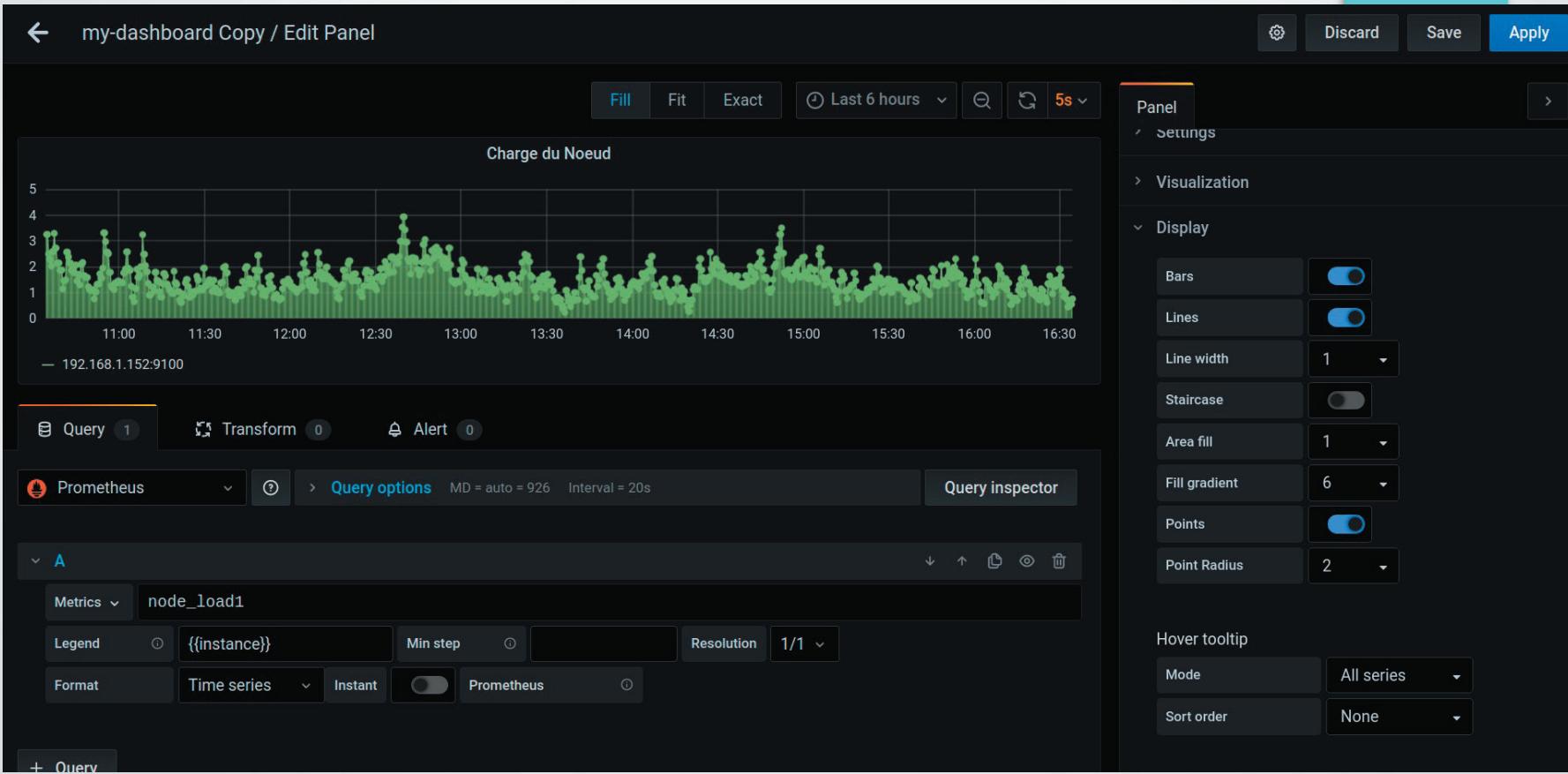
Affichage de l'auto refresh du panel à savoir 5 s



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

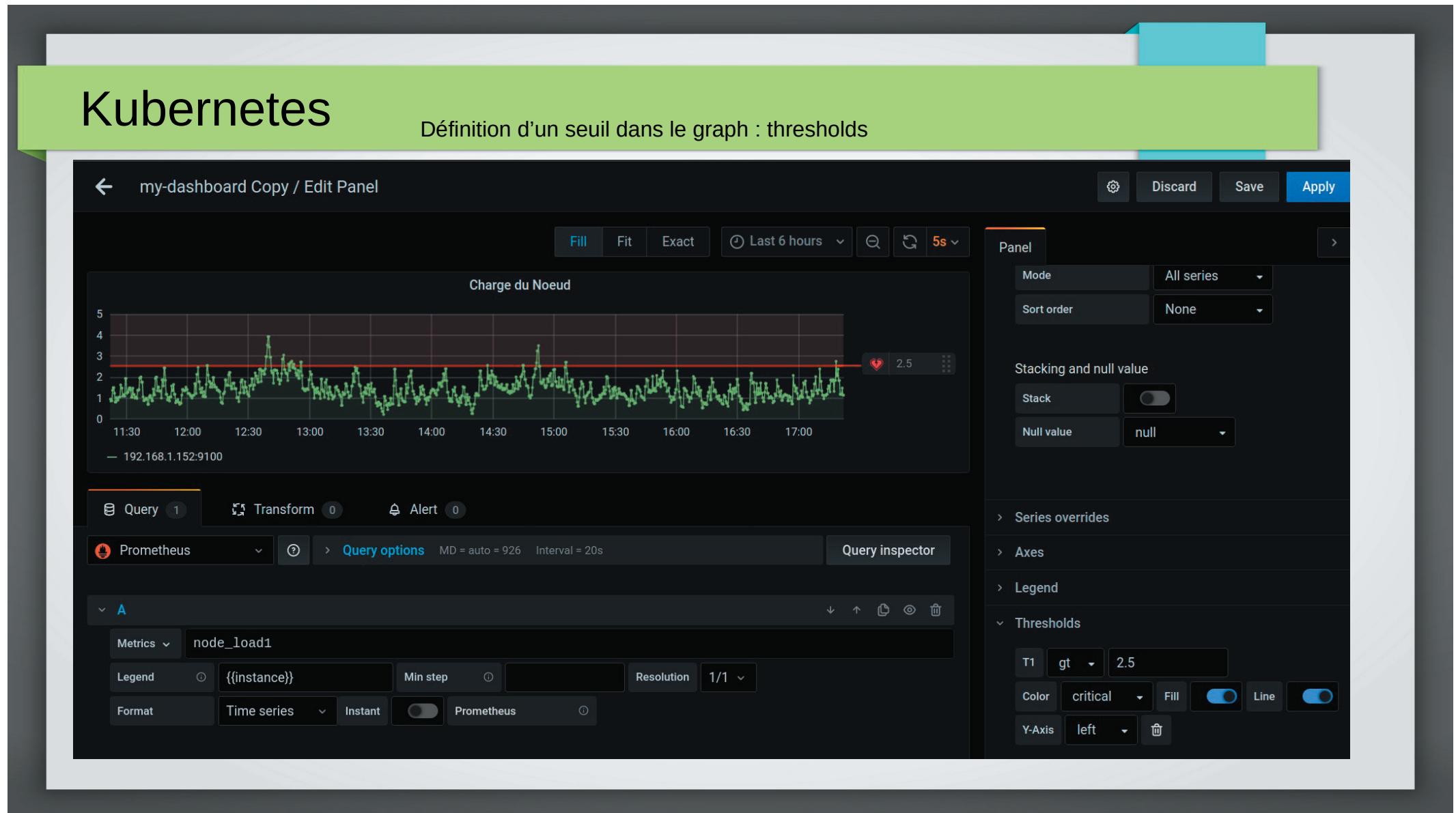
Paramétrage du graph par point, ligne ..



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Définition d'un seuil dans le graph : thresholds



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

PromQL (<https://prometheus.io/docs/prometheus/latest/querying/basics/>) est un langage de requête pour le système de surveillance Prometheus (<https://prometheus.io/>). Il est conçu pour créer des requêtes puissantes mais simples pour des graphiques, des alertes ou séries chronologiques dérivées (règles d'enregistrement https://prometheus.io/docs/prometheus/latest/configuration/recording_rules/). PromQL est conçu à partir de scratch et n'a aucun point commun avec les autres langages de requête

PromQL tutorial for beginners and humans :

<https://valyala.medium.com/promql-tutorial-for-beginners-9ab455142085>

Suivi du système de fichiers

Cree un panel pour surveiller les systèmes de fichiers, L'exportateur de nœuds exporte deux métriques pour que nous puissions récupérer ces statistiques: **node_filesystem_avail_bytes** et **node_filesystem_size_bytes** .

L'un divisé par l'autre nous donnera l'utilisation globale du système de fichiers par périphérique ou par point de montage, comme vous préférez.

Formule : $(1 - \text{node_filesystem_avail_bytes} / \text{node_filesystem_size_bytes}) * 100$

```
ubuntu@prod:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G   0    1.9G  0% /dev
tmpfs           385M  2.8M  382M  1% /run
/dev/sda1        78G  6.7G   71G  9% /
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Prometheus container metric

<https://github.com/google/cadvisor/blob/master/docs/storage/prometheus.md>

Prometheus container metrics

The table below lists the Prometheus container metrics exposed by cAdvisor (in alphabetical order by metric name):

Metric name	Type	Description	Unit (where applicable)	-disable_metrics parameter
<code>container_accelerator_duty_cycle</code>	Gauge	Percent of time over the past sample period during which the accelerator was actively processing	percentage	accelerator
<code>container_accelerator_memory_total_bytes</code>	Gauge	Total accelerator memory	bytes	accelerator
<code>container_accelerator_memory_used_bytes</code>	Gauge	Total accelerator memory allocated	bytes	accelerator
<code>container_blkio_device_usage_total</code>	Counter	Blkio device bytes usage	bytes	diskIO
<code>container_cpu_cfs_periods_total</code>	Counter	Number of elapsed enforcement period intervals		
<code>container_cpu_cfs_throttled_periods_total</code>	Counter	Number of throttled period intervals		
<code>container_cpu_cfs_throttled_seconds_total</code>	Counter	Total time duration the container has been throttled	seconds	
<code>container_cpu_load_average_10s</code>	Gauge	Value of container cpu load average over the last 10 seconds		
<code>container_cpu_schedstat_run_periods_total</code>	Counter	Number of times processes of the cgroup have run on the cpu		sched



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

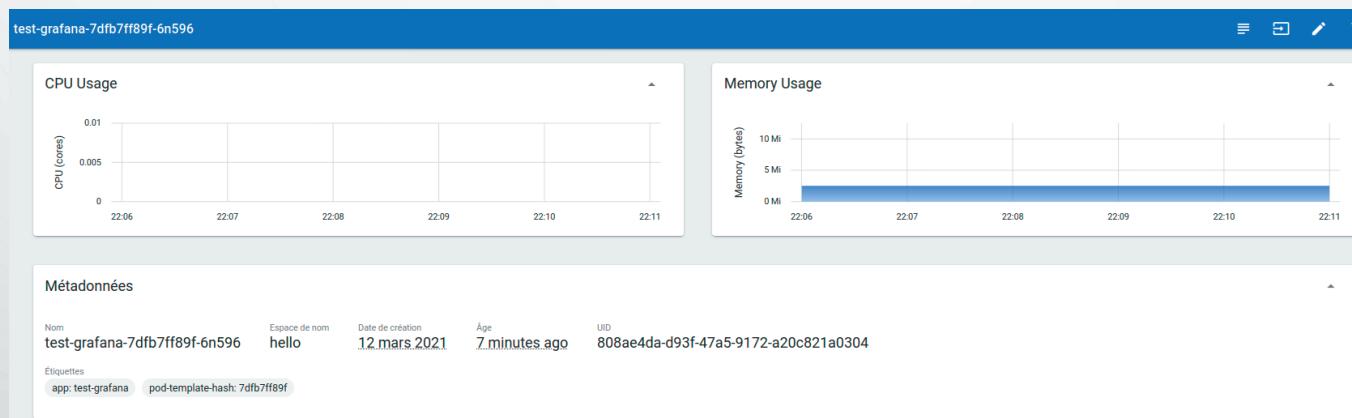
Crée un deployment **test-grafana** avec un container **nginx** dans le namespace **hello**

```
bilel@masterk8s:~$ kubectl create deployment test-grafana --image=nginx -n hello
deployment.apps/test-grafana created
bilel@masterk8s:~$ █
```

Afficher l'utilisation de la mémoire du pod test-grafana avec commande cli

```
bilel@masterk8s:~$ kubectl get po -n hello
NAME           READY   STATUS    RESTARTS   AGE
test-grafana-7dfb7ff89f-6n596  1/1     Running   0          2m40s
bilel@masterk8s:~$ kubectl top po test-grafana-7dfb7ff89f-6n596 -n hello
NAME           CPU(cores)   MEMORY(bytes)
test-grafana-7dfb7ff89f-6n596  0m          2Mi
bilel@masterk8s:~$ █
```

Afficher l'utilisation de la mémoire du pod test-grafana avec dashboard kubernetes



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Afficher l'utilisation de la mémoire du pod test-grafana avec dashboard Prometheus

Prometheus Alerts Graph Status ▾ Help

Enable query history

Try experimental React UI

Load time: 13ms
Resolution: 14s
Total time series: 1

```
container_memory_usage_bytes{namespace="hello",pod="test-grafana-7dfb7ff89f-6n596",container="nginx"}
```

Execute - insert metric at cursor - ↴

Graph Console

Moment

Element	Value
container_memory_usage_bytes(container="nginx",endpoint="https-metrics",id="/kubepods/besteffort/pod808ae4da-d93f-47a5-9172-a20c821a0304 /3cd7edcc762f4380570c298dce91749ed55eb4f12cb8c3262c8b59b644f490d7",image="nginx@sha256:d2925188effb4ddca9f14f162d6fba9b5fab232028aa07ae5c1dab764dca8f9f",instance="192.168.1.152:10250",job="kubelet",metrics_path="/metrics/cadvisor",name="k8s_nginx_test-grafana-7dfb7ff89f-6n596_hello_808ae4da-d93f-47a5-9172-a20c821a0304_0",namespace="hello",node="masterk8s",pod="test-grafana-7dfb7ff89f-6n596",service="prometheus-prometheus-oper-kubelet")	2641920

Remove Graph

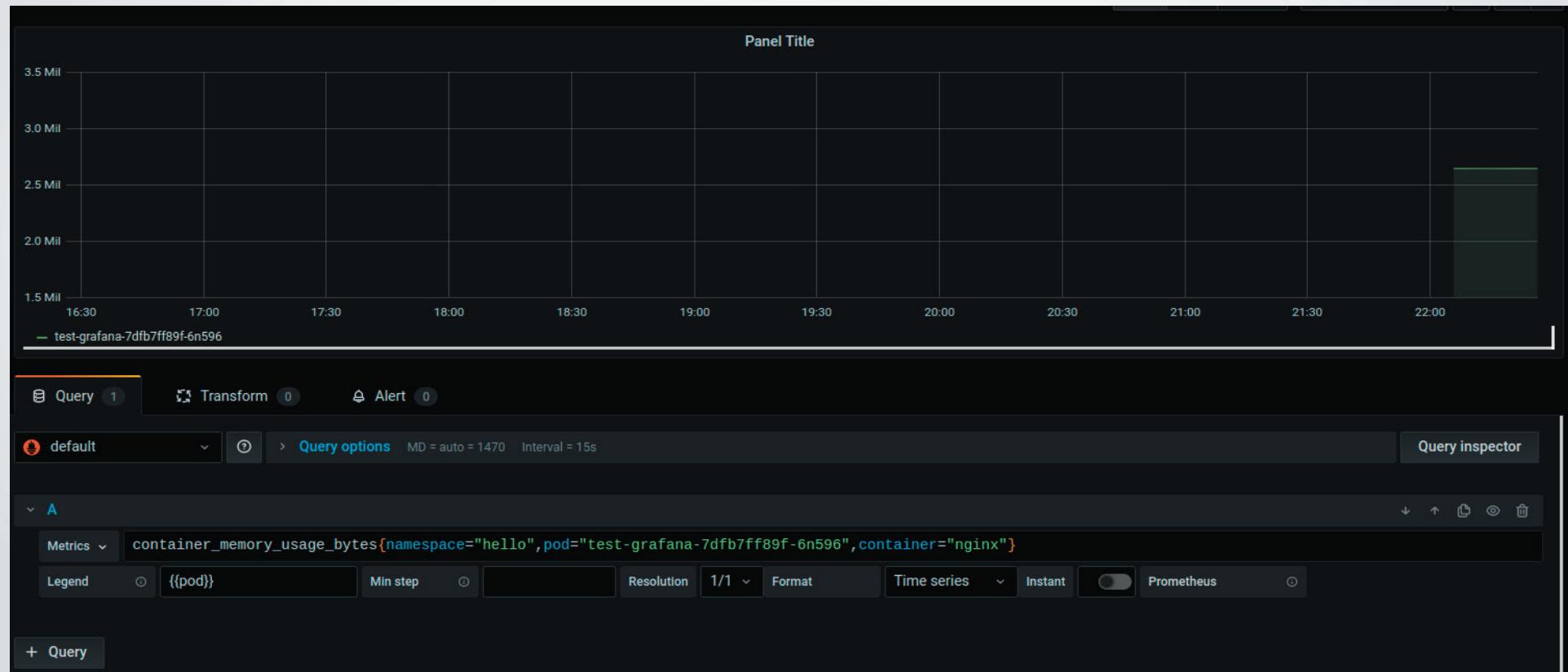
Add Graph



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

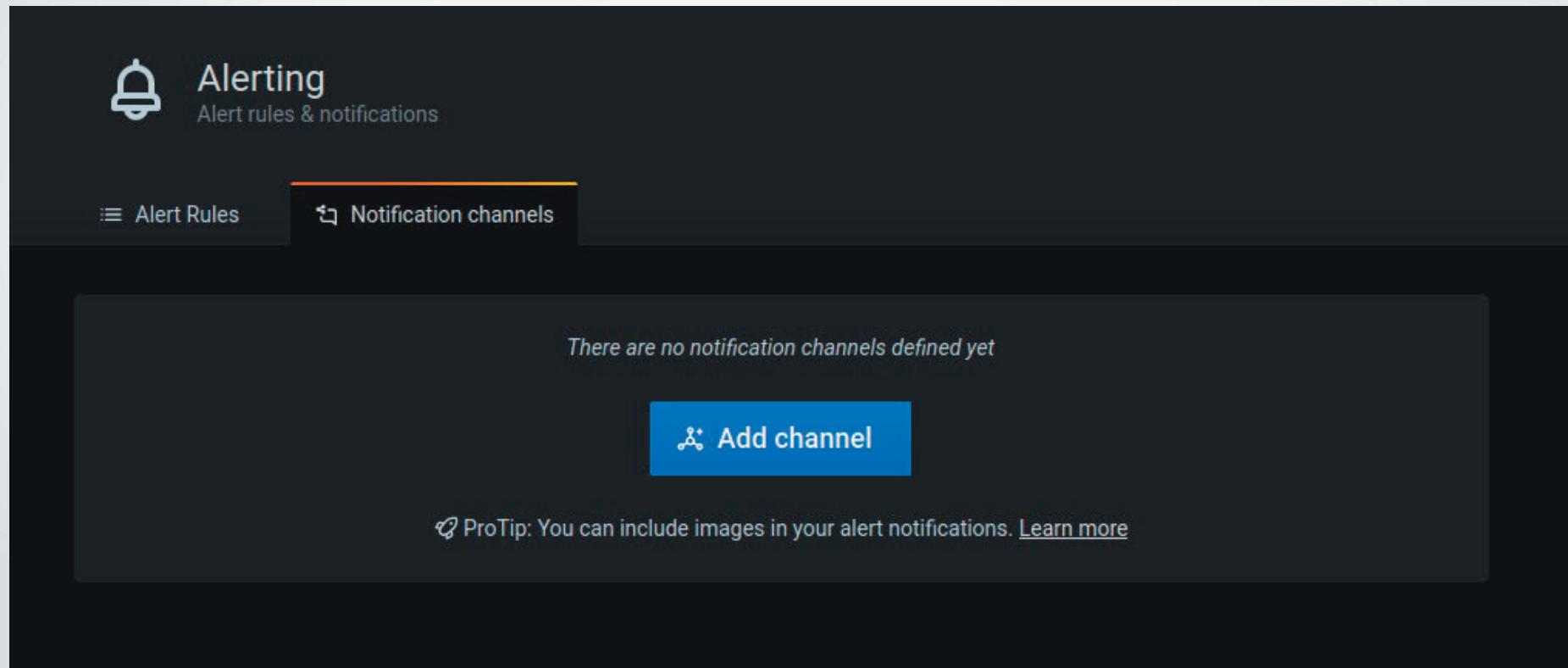
Afficher l'utilisation de la mémoire du pod test-grafana avec dashboard grafana



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Alerte et notification



The screenshot shows the Kubernetes Alerting interface. At the top, there is a header with a bell icon and the text "Alerting" followed by "Alert rules & notifications". Below the header, there are two tabs: "Alert Rules" and "Notification channels". The "Notification channels" tab is currently selected, indicated by an orange underline. A message "There are no notification channels defined yet" is displayed in a central box. Below this message is a blue button with the text "Add channel" and an icon of a person. At the bottom of the screen, there is a ProTip message: "ProTip: You can include images in your alert notifications. [Learn more](#)".



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Definition d'une channel de notification via email

A screenshot of an email from Grafana. The subject is "Send test email". The recipient is "Grafana <issaoulios2015@gmail.com>" with a "À moi" button. The email body contains a Grafana alert notification. It features the Grafana logo at the top. Below it, the title "[Alerting] Test notification" is displayed. A message states "Someone is testing the alert notification within grafana." Under the heading "Error message", there is the text "This is only a test". A table shows two metrics: "High value" with a value of "100.000" and "Higher Value" with a value of "200.000". At the bottom, there are two buttons: "View your Alert rule" and "Go to the Alerts page".

 Alerting
Alert rules & notifications

≡ Alert Rules ↳ Notification channels

Edit Notification Channel

Name	send to me
Type	Email ▼
Default (send on all alerts)	 ⓘ <input checked="" type="checkbox"/>
Include image	 ⓘ <input checked="" type="checkbox"/>
Disable Resolve Message	 ⓘ <input checked="" type="checkbox"/>
Send reminders	 ⓘ <input checked="" type="checkbox"/>

Email settings

Single email ⓘ 	 <input checked="" type="checkbox"/>
Addresses	issaouiios2015@gmail.com

You can enter multiple email addresses using a ";" separator

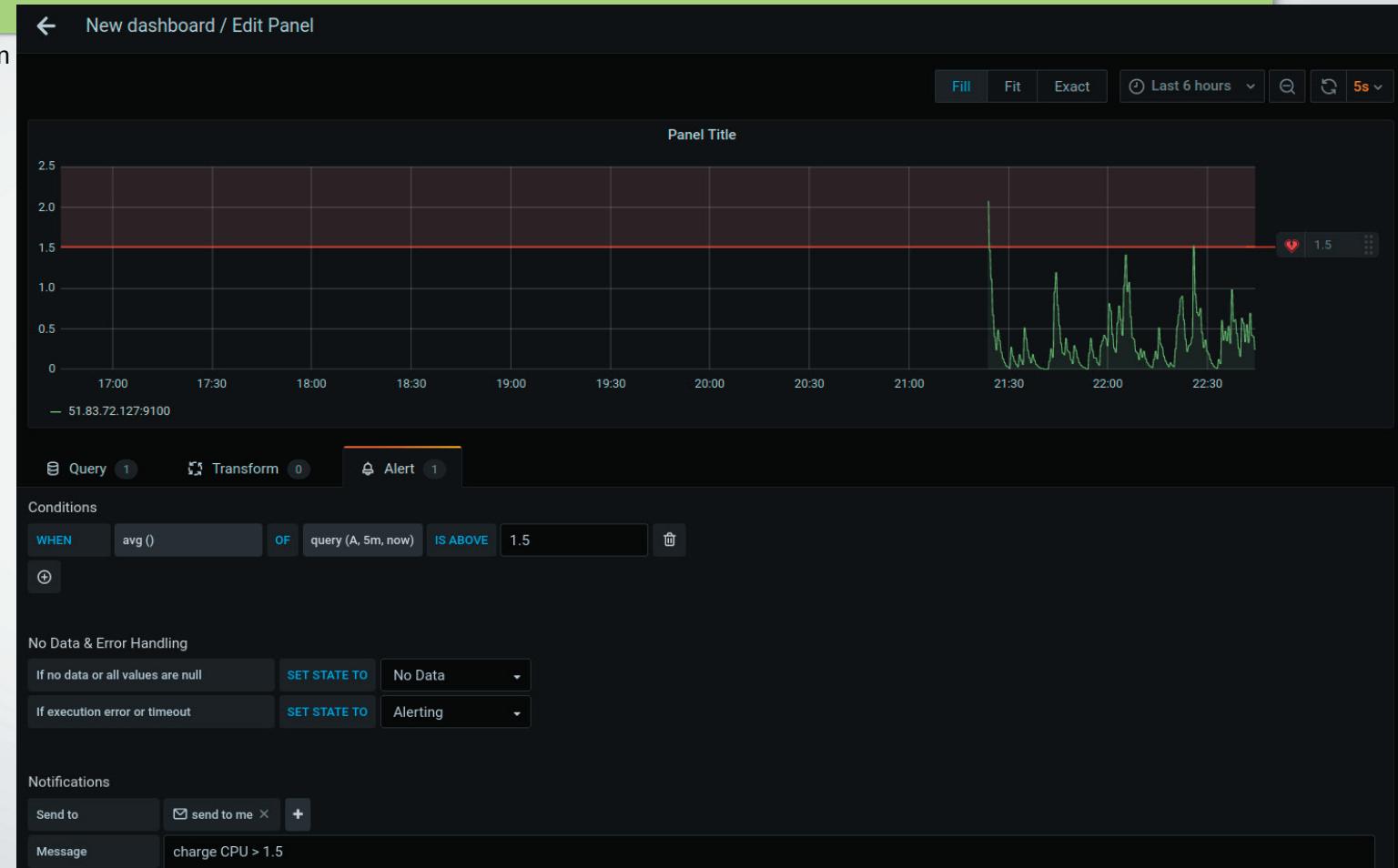
Save Send Test Delete Back



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

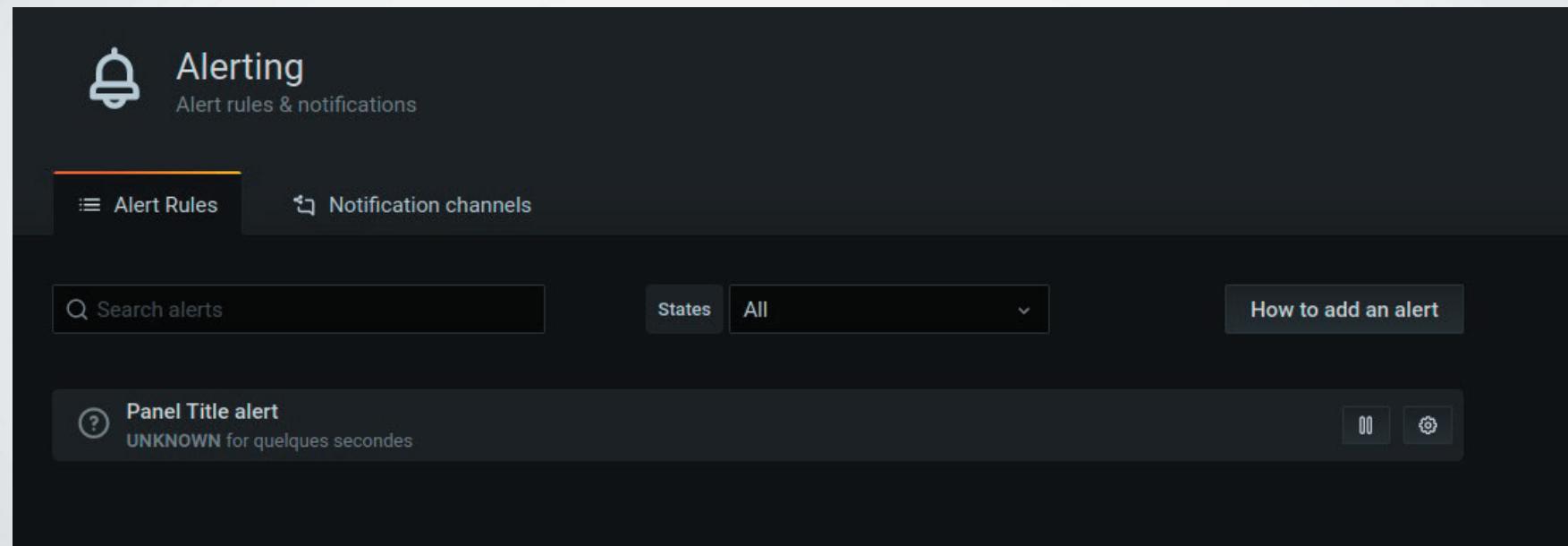
Kubernetes

Notification par e-mail pour l'utilisation du CPU



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



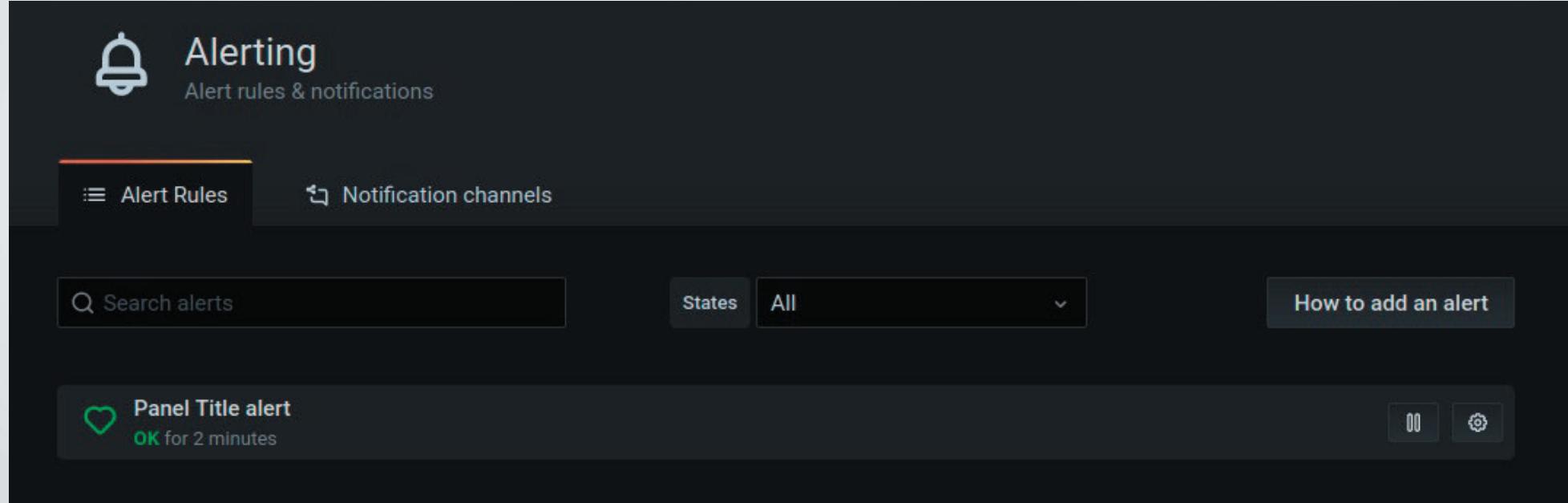
The screenshot shows the Grafana Alerting interface. The title bar says "Alerting" with the subtitle "Alert rules & notifications". Below the title are two tabs: "Alert Rules" (selected) and "Notification channels". There is a search bar labeled "Search alerts", a dropdown for "States" set to "All", and a button "How to add an alert". A single alert card is displayed, titled "Panel Title alert" with a question mark icon, showing the status "UNKNOWN for quelques secondes". To the right of the alert card are two small icons: a square with three dots and a gear.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Surcharger un peut le CPU pour dépasser le seuil 1,5

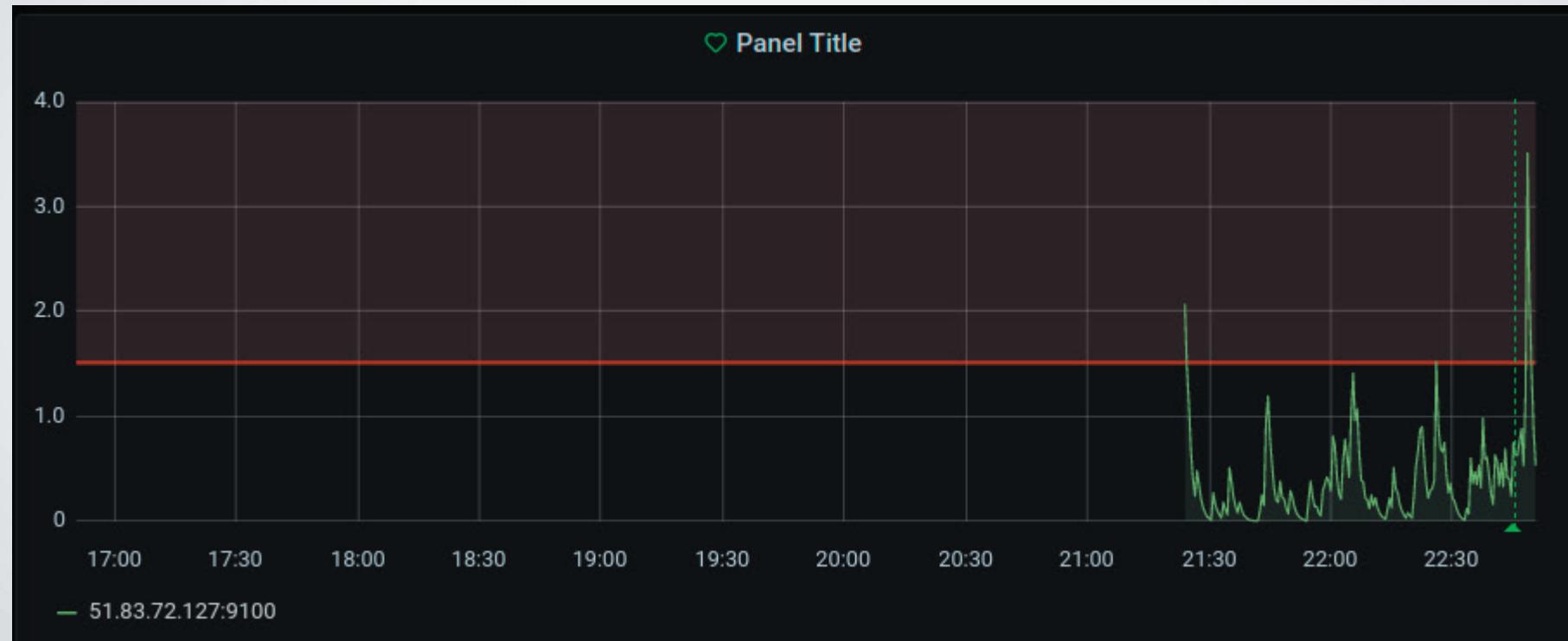


The screenshot shows a dark-themed user interface for managing alert rules and notifications. At the top left is a bell icon and the title "Alerting". Below it are two tabs: "Alert Rules" (selected) and "Notification channels". A search bar labeled "Search alerts" is positioned below the tabs. To the right are filters for "States" (set to "All") and a "How to add an alert" button. A prominent alert message at the bottom left reads "Panel Title alert" with a green heart icon and "OK for 2 minutes".



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

[Alerting] Panel Title alert Boîte de réception ×

 **Grafana** <issaouios2015@gmail.com>
À moi ▾



[Alerting] Panel Title alert
charge CPU > 1.5

Metric name	Value
51.83.72.127:9100	1.787

[View your Alert rule](#) [Go to the Alerts page](#)

Sent by **Grafana v7.0.3**
© 2019 Grafana Labs



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes

Lien exemple prometheus-grafana avec docker

<https://finestructure.co/blog/2016/5/16/monitoring-with-prometheus-grafana-docker-part-1>

<https://grafana.com/grafana/dashboards/893>



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Kubernetes



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155