

Formation Kubernetes

-

Pratique



kubernetes

Atelier 2

-

Administration Kubernetes

Table des matières

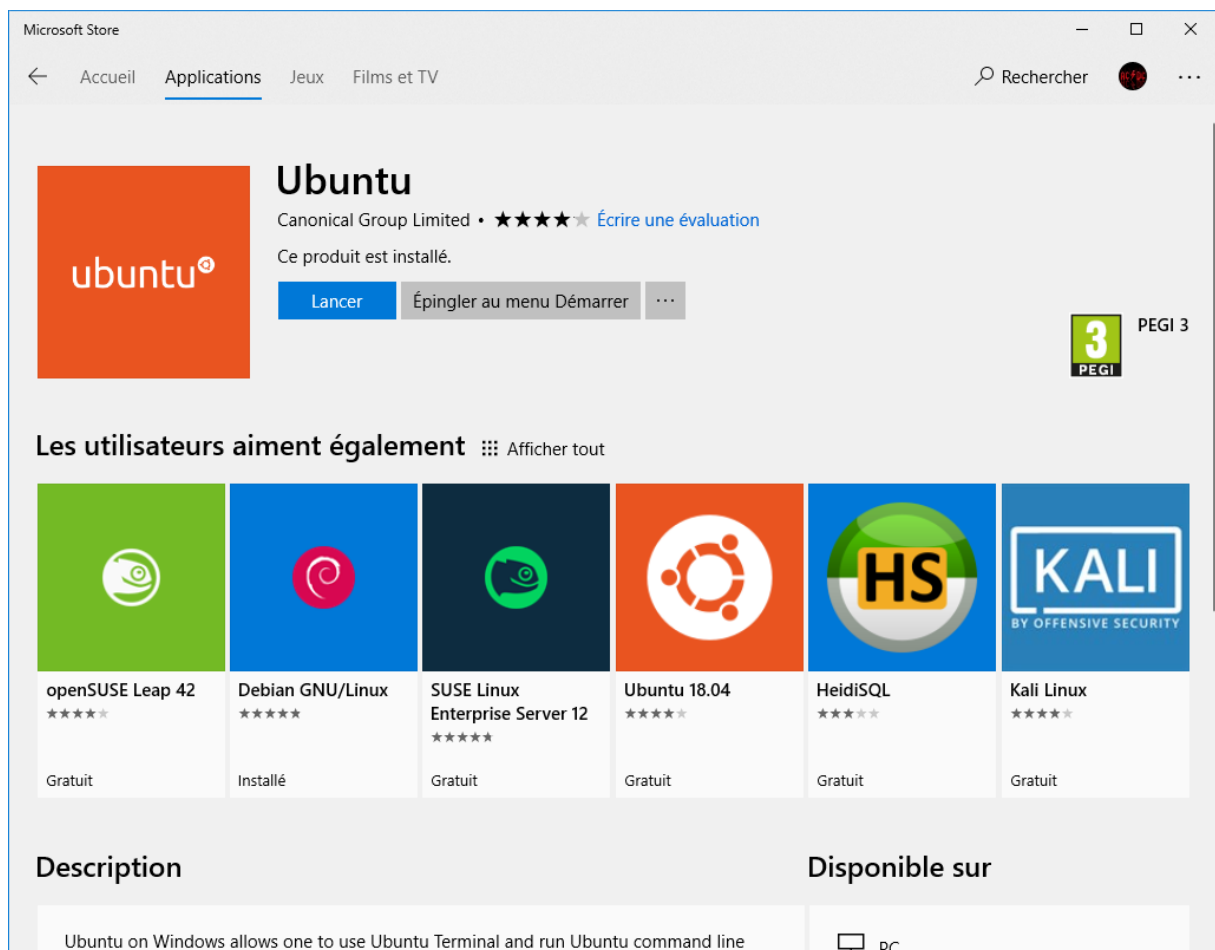
| | | |
|--------|---|----|
| 1. | Installation de Kubectl (Bash Ubuntu)..... | 3 |
| 1.1. | Installation du Bash Ubuntu sur Windows..... | 3 |
| 1.2. | Installation de gcloud | 5 |
| 1.3. | Connexion au projet GCP | 7 |
| 1.4. | Autocomplétion | 11 |
| 1.5. | Installation et configuration de Docker | 11 |
| 1.5.1. | Installation de Docker pour Windows..... | 11 |
| 1.5.2. | Installation de la ligne de commande Docker dans le Bash Ubuntu | 13 |
| 1.5.3. | Connexion au Google Container Registry | 14 |
| 2. | Installation de kubectl pour Windows..... | 16 |
| 3. | Utilisation du kubectl | 19 |
| 3.1. | Informations sur le cluster | 20 |
| 3.1. | Informations sur les nœuds | 20 |
| 3.2. | Informations sur les services système | 21 |
| 3.3. | Communication avec l'API Server | 22 |
| 3.4. | Découvrir les champs de configurations API..... | 25 |
| 4. | Tour d'horizon de l'interface graphique | 29 |
| 4.1. | Dashboard Kubernetes | 29 |
| 4.2. | Dashboard GKE | 34 |
| 4.2.1. | Clusters..... | 34 |
| 4.2.2. | Workloads | 34 |
| 4.2.3. | Services | 36 |
| 4.2.4. | Applications | 36 |
| 4.2.5. | Configuration | 39 |
| 4.2.6. | Storage..... | 40 |

1. Installation de Kubectl (Bash Ubuntu)

1.1. Installation du Bash Ubuntu sur Windows

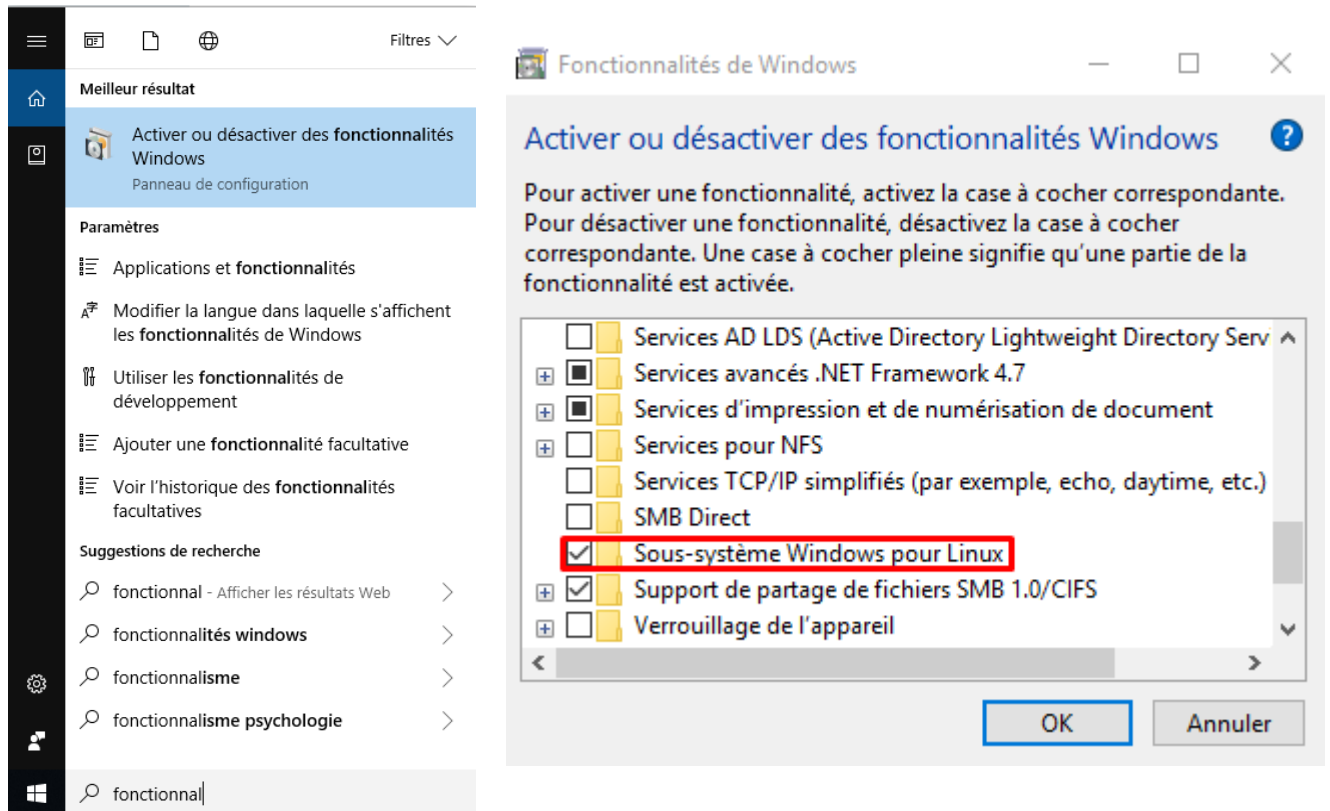
- 1) Commencez par télécharger l'application Ubuntu sur le store de Windows 10 à l'adresse suivante :

<https://www.microsoft.com/store/productId/9NBLGGH4MSV6>

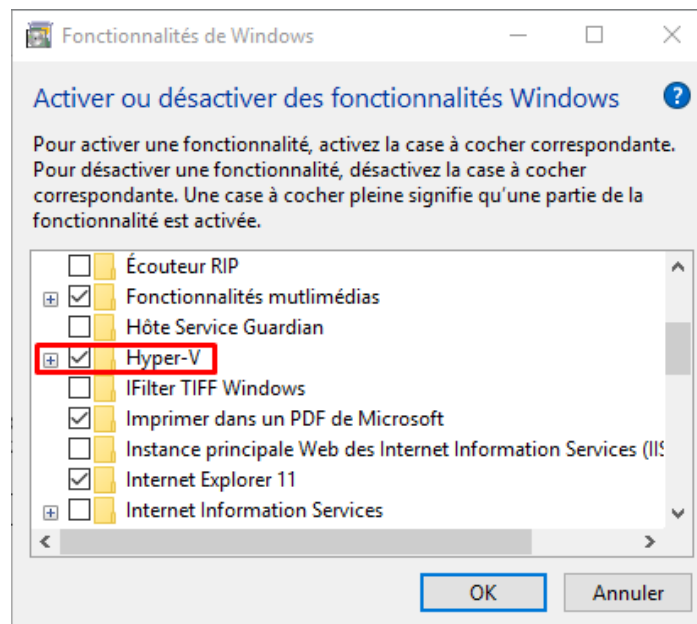


Vous utiliserez par la suite ce Bash afin de vous connecter à votre plateforme GCP et interagir avec.

- 2) Une fois le téléchargement terminé, activez la fonctionnalité de Windows pour le sous-système Windows pour Linux comme indiqué sur la capture ci-dessous :



- 3) Activez également la fonctionnalité Hyper-V qui vous sera utile par la suite :



- 4) Redémarrez votre ordinateur pour appliquer les modifications.

1.2. Installation de gcloud

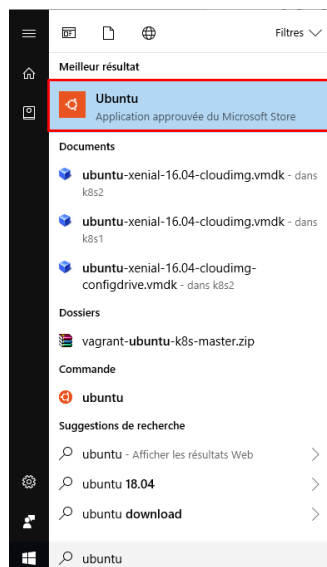
gcloud est un outil qui fournit l'interface de ligne de commande principale de Google Cloud Platform. Vous pouvez utiliser cet outil pour effectuer de nombreuses tâches courantes sur la plate-forme, à partir de la ligne de commande ou dans des scripts et autres moyens d'automatisation.

Par exemple, vous pouvez utiliser **gcloud** pour créer et gérer :

- Instances et autres ressources de Google Compute Engine
- Instances de Google Cloud SQL
- Les clusters de Google Kubernetes Engine
- Clusters et jobs de Google Cloud Dataproc
- Zones managées et ensembles d'enregistrement de Google Cloud DNS
- Déploiements de Google Cloud Deployment

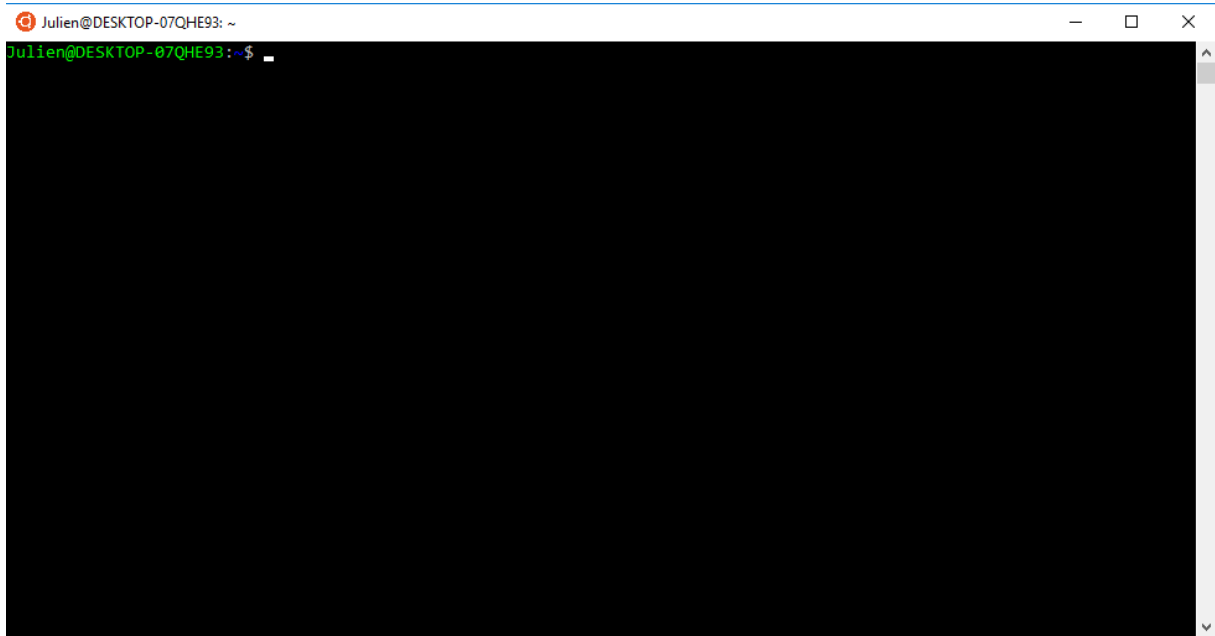
gcloud fait partie du **Google Cloud SDK**. Nous allons donc procéder à l'installation de ce SDK.

1) Pour cela, commencez par lancer le Bash Ubuntu fraîchement installé.



Note : Une fois exécuté, vous devez attendre plusieurs minutes pour qu'il finisse sa configuration automatique.

Voici à quoi ressemble votre Bash Ubuntu une fois installé et configuré :



Pour information, il s'agit d'un environnement Linux (Ubuntu 18.04 « Bionic ») avec son propre kernel et arborescence de fichiers.

2) Dans le but d'installer le SDK vous devez commencer par télécharger python2.7.

a. Ajoutez le repository de python 2.7 :

```
$ sudo add-apt-repository ppa:deadsnakes/ppa
```

b. Mettez à jour vos repositories :

```
$ sudo apt update
```

c. Installez python 2.7 :

```
$ sudo apt install python2.7 python-pip
```

3) Téléchargez le **Google Cloud SDK** puis décompressez-le :

```
$ wget install  
https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-  
cloud-sdk-214.0.0-linux-x86_64.tar.gz  
$ tar zxvf [ARCHIVE_FILE] google-cloud-sdk
```

4) Rendez-vous dans le dossier « **google-cloud-sdk/** » et lancez le script d'installation :

```
$ ./google-cloud-sdk/install.sh
#répondez non
n
#puis oui
Y
#puis validez le bashrc
```

5) Avant de pouvoir utiliser la commande **gcloud** vous devez l'importer dans votre bashrc:

```
$ export PATH=$PATH:[google-cloud-sdk/bin]
$ sudo echo "export PATH=$PATH:[google-cloud-sdk/bin]" >>
/etc/bash.bashrc
```

Fermez la session et réouvrez en une autre. Vous pouvez maintenant utiliser la commande **gcloud** dans votre Bash.

6) Vous allez ajouter les commandes **kubectl** qui vous permettront d'interagir avec **Kubernetes** :

```
$ gcloud components install kubectl
```

1.3. Connexion au projet GCP

1) Maintenant, vous pouvez utiliser la commande **gcloud init** qui permet de se connecter à un projet GCP.

Un certain nombre d'informations vous seront demandées. Elles sont représentées en vert ci-dessous :

```
$ gcloud init

Welcome! This command will take you through the configuration of
gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
```



```
gcloud init --skip-diagnostics
```

```
Network diagnostic detects and fixes local network connection
issues.
```

```
Checking network connection...done.
```

```
Reachability Check passed.
```

```
Network diagnostic (1/1 checks) passed.
```

```
You must log in to continue. Would you like to log in (Y/n)?  Y
```

```
Go to the following link in your browser:
```

```
https://...
```

```
Enter verification code:
```

```
4/SgAZpZy06zA4bJsl_SdvKRjajkzefuhXbWbn4izozadkokfrkOt-o
```

```
You are logged in as: [username@gmail.com].
```

```
Pick cloud project to use:
```

```
[1] k8s-training01-214314
```

```
[2] Create a new project
```

```
Please enter numeric choice or text value (must exactly match list
item):  1
```

```
Your current project has been set to: [k8s-training01-214314].
```

```
Do you want to configure a default Compute Region and Zone? (Y/n)?y
```

```
Which Google Compute Engine zone would you like to use as project
default?
```

```
If you do not specify a zone via a command line flag while working
```



```
with Compute Engine resources, the default is assumed.
...
[17] europe-west1-b
[18] europe-west1-d
[19] europe-west1-c
...
Did not print [3] options.
Too many options [53]. Enter "list" at prompt to print choices
fully.
Please enter numeric choice or text value (must exactly match list
item):17

Your project default Compute Engine zone has been set to [europe-
west1-d].

You can change it by running [gcloud config set compute/zone NAME].

Your project default Compute Engine region has been set to [europe-
west1].

You can change it by running [gcloud config set compute/region
NAME].

Created a default .boto configuration file at
[/home/username/.boto]. See this file and
[https://cloud.google.com/storage/docs/gsutil/commands/config] for
more
information about configuring Google Cloud Storage.
Your Google Cloud SDK is configured and ready to use!

* Commands that require authentication will use username@gmail.com
by default
* Commands will reference project `k8s-training01-214314` by default
* Compute Engine commands will use region `europe-west1` by default
```

* Compute Engine commands will use zone `europe-west1-d` by default

Run `gcloud help config` to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional configurations if you work with multiple accounts and/or projects.

Run `gcloud topic configurations` to learn more.

Some things to try next:

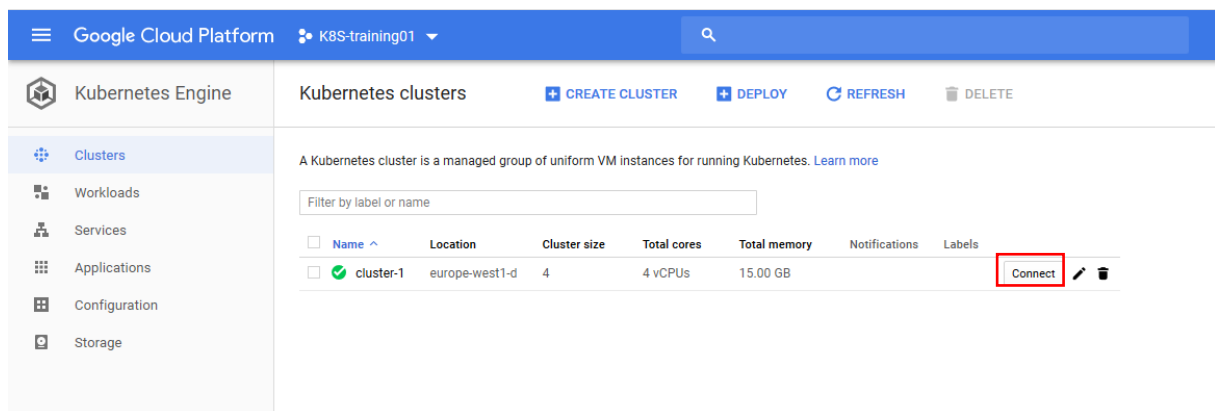
* Run `gcloud --help` to see the Cloud Platform services you can interact with. And run `gcloud help COMMAND` to get help on any gcloud command.

* Run `gcloud topic -h` to learn about advanced features of the SDK like arg files and output formatting

- 2) Maintenant que vous êtes connecté au projet GCP, vous pouvez vous connecter à votre cluster **Kubernetes** avec la commande suivante (modifiez les valeurs en vert par les vôtres) :

```
$ gcloud container clusters get-credentials <k8s-cluster-name> --zone
<zone> --project <project-ID>
```

Note : Cette commande est également disponible dans GKE. Pour la visualiser, allez à la page de visualisation de votre cluster et cliquez sur le bouton « *Se connecter* ».





Google Cloud Platform K8S-training01

Kubernetes Engine

Kubernetes clusters CREATE CLUSTER DEPLOY REFRESH DELETE

A Kubernetes cluster is a managed group of uniform VM instances for running Kubernetes. [Learn more](#)

Filter by label or name

| <input type="checkbox"/> Name ^ | Location | Cluster size | Total cores | Total memory | Notifications | Labels | |
|---|----------------|--------------|-------------|--------------|---------------|--------|---|
| <input checked="" type="checkbox"/> cluster-1 | europe-west1-d | 4 | 4 vCPUs | 15.00 GB | | | Connect   |

- 3) Validez la bonne configuration de **kubectl** en lançant la commande suivante qui affiche les informations de votre cluster :

```
$ kubectl cluster-info
```

```
kubernetes master is running at https://35.195.5.123
Kubelet is running at https://35.195.5.123/api/v1/namespaces/kube-system/services/default-http-backend:http/proxy
Heapster is running at https://35.195.5.123/api/v1/namespaces/kube-system/services/heapster/proxy
KubeDNS is running at https://35.195.5.123/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Kubernetes-dashboard is running at https://35.195.5.123/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy
Metrics-server is running at https://35.195.5.123/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy
```

Votre CLI **kubectl** est désormais configurée et fonctionnelle.

1.4. Autocomplétion

Il est possible de configurer l'autocomplétion de **kubectl**.

- 1) Pour activer l'autocomplétion, vous pouvez exécuter les commandes suivantes (attention ces commandes ne fonctionnent pas pour l'utilisateur root) :

```
$ echo "source <(kubectl completion bash)" >> ~/.bashrc
# ajoute l'autocomplétion de façon permanente au Shell pour
l'utilisateur en question.
$ source ~/.bashrc
# recharge le bashrc pour prendre en compte la modification
```

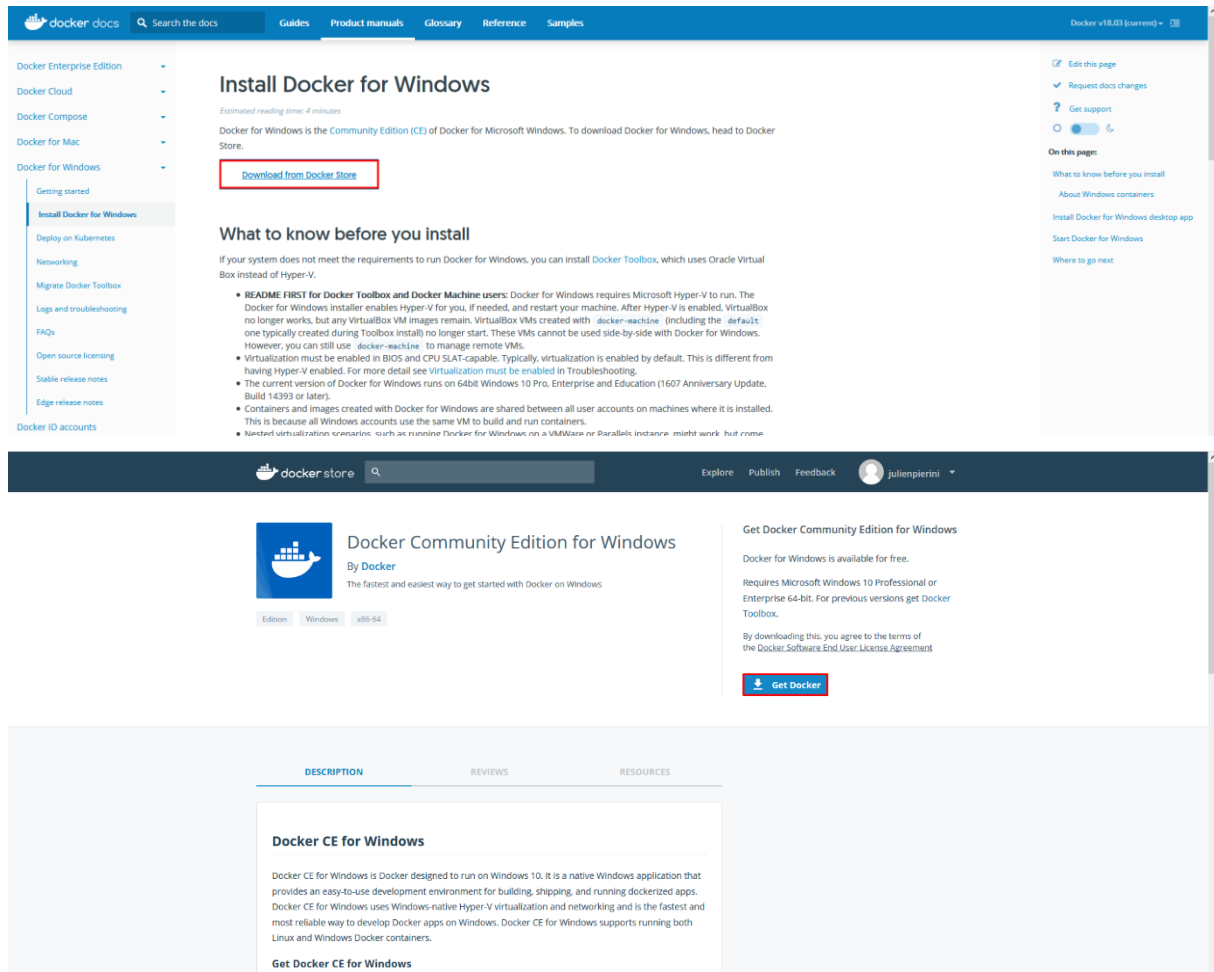
- 2) Validez la bonne configuration de l'autocomplétion de **kubectl**.

1.5. Installation et configuration de Docker

1.5.1. Installation de Docker pour Windows

- 1) Afin de pouvoir utiliser les commandes **Docker** et le **Docker Engine**, vous devez installer **Docker** sur votre machine Windows. Pour cela rendez-vous sur l'adresse suivante :

<https://docs.docker.com/docker-for-windows/install/>



Install Docker for Windows

Estimated reading time: 4 minutes

Docker for Windows is the Community Edition (CE) of Docker for Microsoft Windows. To download Docker for Windows, head to Docker Store.

[Download from Docker Store](#)

What to know before you install

If your system does not meet the requirements to run Docker for Windows, you can install Docker Toolbox, which uses Oracle Virtual Box instead of Hyper-V.

- **README FIRST for Docker Toolbox and Docker Machine users:** Docker for Windows requires Microsoft Hyper-V to run. The Docker for Windows installer enables Hyper-V for you, if needed, and restart your machine. After Hyper-V is enabled, VirtualBox no longer works, but any VirtualBox VM images remain. VirtualBox VMs created with `docker-machine` (including the default one typically created during Toolbox install) no longer start. These VMs cannot be used side-by-side with Docker for Windows. However, you can still use `docker-machine` to manage remote VMs.
- Virtualization must be enabled in BIOS and CPU SLAT-capable. Typically, virtualization is enabled by default. This is different from having Hyper-V enabled. For more detail see [Virtualization must be enabled in Troubleshooting](#).
- The current version of Docker for Windows runs on 64bit Windows 10 Pro, Enterprise and Education (1607 Anniversary Update, Build 14393 or later).
- Containers and images created with Docker for Windows are shared between all user accounts on machines where it is installed. This is because all Windows accounts use the same VM to build and run containers.
- Nested virtualization scenarios, such as running Docker for Windows on a VMWare or Parallels instance, might work, but come

Docker Community Edition for Windows

By Docker

The fastest and easiest way to get started with Docker on Windows

Get Docker Community Edition for Windows

Docker for Windows is available for free.

Requires Microsoft Windows 10 Professional or Enterprise 64-bit. For previous versions get Docker Toolbox.

By downloading this, you agree to the terms of the Docker Software End User License Agreement

[Get Docker](#)

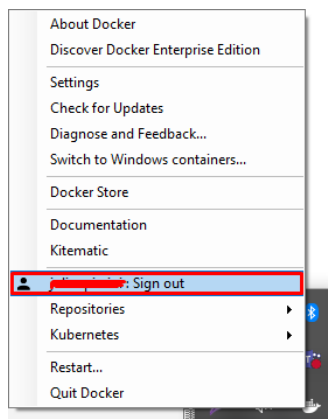
DESCRIPTION

Docker CE for Windows

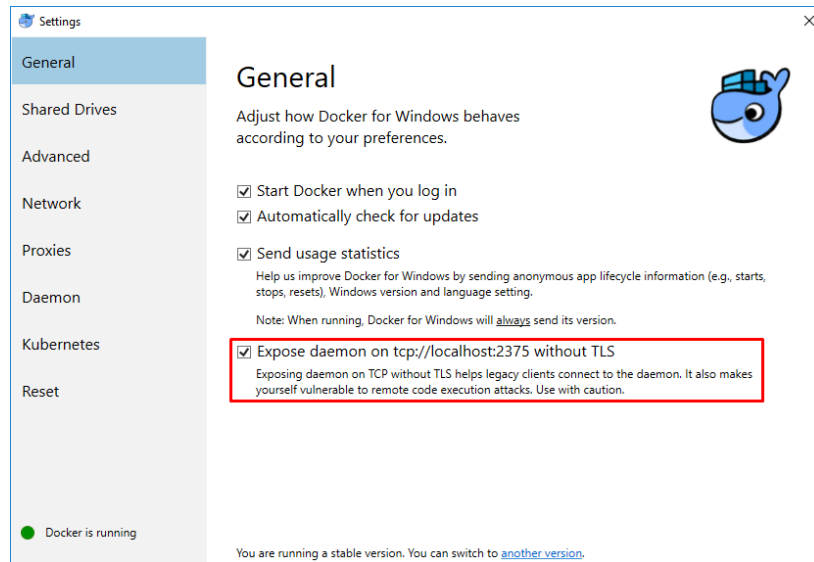
Docker CE for Windows is Docker designed to run on Windows 10. It is a native Windows application that provides an easy-to-use development environment for building, shipping, and running dockerized apps. Docker CE for Windows uses Windows-native Hyper-V virtualization and networking and is the fastest and most reliable way to develop Docker apps on Windows. Docker CE for Windows supports running both Linux and Windows Docker containers.

Get Docker CE for Windows

2) Une fois téléchargé, suivez les instructions d'installation et créez un compte si cela n'est pas déjà fait.



- 3) Lorsque l'installation est terminée, allez dans les paramètres de l'application puis activez la fonctionnalité suivante qui vous permettra d'exécuter des commandes **Docker** depuis votre bash Ubuntu :



1.5.2. Installation de la ligne de commande Docker dans le Bash Ubuntu

- 1) Ouvrez votre terminal Ubuntu puis suivez les instructions suivantes :
- Mettez à jour la liste de vos repositories :

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

- Installez les paquets suivants qui sont nécessaires au fonctionnement de **Docker** :

```
$ sudo apt-get install apt-transport-https ca-certificates curl  
software-properties-common
```

- Ajoutez la clef officielle de **Docker** pour poursuivre le téléchargement par la suite :

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-  
key add -
```

- Vérifiez bien que la clef termine bien par 0EBFCD88 :

```
$ sudo apt-key fingerprint 0EBFCD88
```

e. Ajoutez le repository de **Docker** :

```
$ sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

f. Remettez à jour la liste de vos repositories

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

g. Enfin, installez Docker Community Edition :

```
$ sudo apt-get install docker-ce
```

2) Essayez la commande suivante pour vérifier que le daemon **Docker** pour Windows communique bien avec votre sous-système Ubuntu :

```
$ docker -H localhost:2375 images
```

3) Afin de ne plus utiliser l'option -H à chaque commande, il suffit d'ajouter la variable d'environnement suivante :

```
$ export DOCKER_HOST=localhost:2375
```

Ou de façon permanente en l'ajoutant dans votre bashrc :

```
# echo "export DOCKER_HOST=localhost:2375" >> /etc/bash.bashrc
```

4) Redémarrez votre Bash.

```
$ source ~/.bashrc
```

1.5.3. Connexion au Google Container Registry

1) Tout d'abord, installez le composant suivant :

```
$ gcloud components install docker-credential-gcr
```

Ce composant permet de configurer Docker pour qu'il puisse utiliser le service **Container Registry** de Google Cloud Platform.

2) Maintenant, vous pouvez configurer **Docker** pour se connecter au registre avec la commande qui suit :

```
$ docker-credential-gcr configure-docker
```

Ceci permettra par la suite d'utiliser les commandes « *pull* » et « *push* » de **Docker** en ciblant le registre GCP.

3) Validez que l'utilisation du registry est désormais fonctionnelle

```
$ docker pull nginx:latest
$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
nginx                latest       06144b287844     8 days ago      109MB

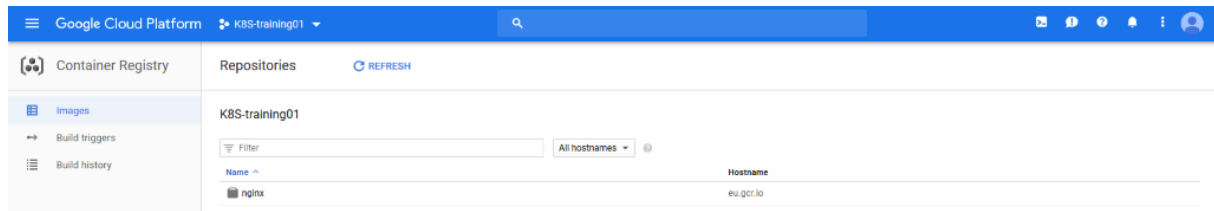
$ docker tag nginx eu.gcr.io/k8s-training01-214314/test:v1
# remplacez `k8s-training01-214314/test:v1` par vos valeurs
$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
nginx                latest       06144b287844     8 days ago      109MB
eu.gcr.io/<id-pjt>/<name> <tag>    06144b287844     8 days ago      109MB

$ docker push eu.gcr.io/k8s-training01-214314/test:v1
```

us.gcr.io indique le bucket vers lequel l'image doit être poussée dans votre cas. Il existe néanmoins plusieurs registres géo-localisés :

| | |
|-------------|--|
| gcr.io | États-Unis (risque de changer dans le futur) |
| us.gcr.io | États-Unis (2) (séparé du 1 ^{er}) |
| eu.gcr.io | Europe |
| asia.gcr.io | Asie |

- a. Vous pouvez retrouver l'image poussée dans l'interface en vous rendant dans l'onglet « **Container Registry** ».



2. Installation de kubectl pour Windows

Dans le cas où vous ne souhaitez pas utiliser le bash, voici la méthode pour manipuler votre cluster directement depuis le shell gcloud sur Windows.

- 1) Pour utiliser **kubectl** sur Windows et gérer un environnement GCP, nous allons installer la CLI GCP « gcloud ». Le programme d'installation pour Windows est disponible à l'adresse suivante : <https://cloud.google.com/sdk/>.
- 2) Lors de l'installation, décochez l'option permettant à Google de récolter vos données et laissez les autres options par défaut. Lorsque l'installation est terminée « gcloud » lance automatiquement la commande « gcloud init » pour la configuration de « gcloud » à un compte GCE.
- 3) Indiquez les informations sur la base de l'exemple ci-dessous en vert. A remplacer par vos informations de projet.

```
$ gcloud init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic (1/1 checks) passed.

You must log in to continue. Would you like to log in (Y/n)?  Y

Go to the following link in your browser:

  https://...

Enter verification code:
4/SgAZpZy06zA4bJsl_SdvKRjaJkzefuhXbWbn4izozadkokfrkOt-o
You are logged in as: [username@gmail.com].
```



```
Pick cloud project to use:
[1] k8s-training01-214314
[2] Create a new project
Please enter numeric choice or text value (must exactly match list
item): 1

Your current project has been set to: [k8s-training01-214314].

Do you want to configure a default Compute Region and Zone? (Y/n)?y

Which Google Compute Engine zone would you like to use as project
default?
If you do not specify a zone via a command line flag while working
with Compute Engine resources, the default is assumed.
...
[17] europe-west1-b
[18] europe-west1-d
[19] europe-west1-c
...
Did not print [3] options.
Too many options [53]. Enter "list" at prompt to print choices fully.
Please enter numeric choice or text value (must exactly match list
item):17

Your project default Compute Engine zone has been set to [europe-west1-d].
You can change it by running [gcloud config set compute/zone NAME].

Your project default Compute Engine region has been set to [europe-west1].
You can change it by running [gcloud config set compute/region NAME].

Created a default .boto configuration file at [/home/username/.boto]. See
this file and
[https://cloud.google.com/storage/docs/gsutil/commands/config] for more
information about configuring Google Cloud Storage.
Your Google Cloud SDK is configured and ready to use!

* Commands that require authentication will use username@gmail.com by
default
* Commands will reference project `k8s-training01-214314` by default
* Compute Engine commands will use region `europe-west1` by default
* Compute Engine commands will use zone `europe-west1-d` by default

Run `gcloud help config` to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional
configurations if you work with multiple accounts and/or projects.
Run `gcloud topic configurations` to learn more.

Some things to try next:

* Run `gcloud --help` to see the Cloud Platform services you can interact
with. And run `gcloud help COMMAND` to get help on any gcloud command.
* Run `gcloud topic -h` to learn about advanced features of the SDK like
arg files and output formatting
```

- 4) Utilisez la commande suivante pour installer **kubectl** dans « gcloud ». Laissez les options par défaut lors de l'installation.

```
gcloud components install kubectl
```

- 5) Maintenant que vous êtes connectés au projet GCP, vous pouvez vous connecter à notre cluster **Kubernetes** avec la commande suivante.

```
$ gcloud container clusters get-credentials {NOM_DU_CLUSTER} --zone {ZONE}
```

- 6) Pour vérifier que **kubectl** est correctement installé et configuré, lancez la commande « **kubectl cluster-info** ».

```
>kubectl cluster-info
Kubernetes master is running at https://35.205.154.81
GLBCDefaultBackend is running at
https://35.205.154.81/api/v1/namespaces/kube-system/services/default-http-
backend:http/proxy
Heapster is running at https://35.205.154.81/api/v1/namespaces/kube-
system/services/heapster/proxy
KubeDNS is running at https://35.205.154.81/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy
kubernetes-dashboard is running at
https://35.205.154.81/api/v1/namespaces/kube-
system/services/https:kubernetes-dashboard:/proxy
Metrics-server is running at https://35.205.154.81/api/v1/namespaces/kube-
system/services/https:metrics-server:/proxy
```

3. Utilisation du kubectl

Kubectl est le client qui permet d'interagir avec votre cluster **Kubernetes** (par l'intermédiaire de l'API Server).

1) Pour débiter, vous pouvez afficher l'aide de **kubectl** grâce à la commande suivante :

```
$ kubectl -h
Find more information at:
https://kubernetes.io/docs/reference/kubectl/overview/

Utilisation :
  kubectl [flags] [options]

Liste des [flags] disponibles par catégorie. Pour afficher les
options, utilisez la commande kubectl [flags] -h :

Basic Commands (Beginner):
  create / expose / run / set / run-container

Basic Commands (Intermediate):
  get / explain / edit / delete

Ces commandes sont toujours suivies d'une <ressource> et
éventuellement le <nom-de-la-ressource>

Deploy Commands:
  rollout / rolling-update / scale / autoscale

Cluster Management Commands:
  certificate / cluster-info / top / cordon / uncordon / drain /
  taint

Troubleshooting and Debugging Commands:
  describe / logs / attach / exec / port-forward / proxy / cp / auth

Advanced Commands:
  apply / patch / replace / convert
```

Settings Commands:

label / annotate / completion

Other Commands:

api-versions / config / help / plugin / version

3.1. Informations sur le cluster

Vous retrouverez ci-dessous une série de commandes permettant d'afficher l'état du cluster **Kubernetes** :

```
$ kubectl cluster-info
```

#Affiche les adresses de l'API server et des services associés au cluster Kubernetes

```
$ kubectl cluster-info dump
```

#Affiche l'ensemble des logs liés au cluster. Utile lors du troubleshooting

```
$ kubectl config view
```

Affiche le fichier de configuration utilisé pour s'authentifier avec le(s) serveur(s) API

3.1. Informations sur les nœuds

Vous retrouverez ci-dessous une série de commandes permettant lister les nodes composant votre cluster :

```
$ kubectl get nodes
```

Affiche les nœuds qui appartiennent au cluster Kubernetes avec les informations suivantes : leur état, leur rôle, leur âge et leur version

```
$ kubectl get nodes -o wide
```

Affiche des informations supplémentaires sur les nœuds du cluster (version d'OS, version du kernel...)

Vous pouvez également afficher des informations plus détaillées sur vos **workers** grâce à la commande suivante :

```
$ kubectl describe node <nom-du-worker>  
# affiche l'état de la machine k8s-worker du cluster
```

Pour rappel, dans votre contexte, vous ne pouvez pas afficher les informations associées aux **masters**, car ceux-ci sont entièrement managés par GKE.

3.2. Informations sur les services système

Lors de l'initialisation de notre cluster, certaines ressources indispensables à son fonctionnement ont été créées. Nous allons les explorer afin que vous puissiez vous familiariser avec l'interpréteur **kubectl**.

Après l'initialisation du cluster, 3 namespaces sont créés à savoir :

- default
- kube-public
- kube-system

Ci-après, la commande permettant d'afficher ces namespaces :

```
$ kubectl get namespaces  
# affiche tous les namespaces du cluster kubernetes
```

Certains services sont aussi déployés lors de l'initialisation afin de les afficher utiliser la commande suivante :

```
$ kubectl get service --all-namespaces  
# affiche les services du namespaces kube-system
```

Certains pods sont aussi déployés lors de l'initialisation. Afin de les afficher, utilisez la commande suivante :

```
$ kubectl get pods --all-namespaces
```

L'option « **--all-namespaces** » permet d'afficher toutes les ressources déployées au sein du cluster **Kubernetes**.

Pour obtenir plus d'informations sur les pods, utilisez l'option suivante :

```
$ kubectl get pods --all-namespaces -o wide
```

Vous pouvez utiliser l'option « **-n** » qui permet de sélectionner un namespace particulier. Si l'option n'est pas spécifiée, la commande s'exécutera pour le namespace « default ».

```
$ kubectl get pods -n kube-system -o wide
```

La commande suivante permet d'afficher les événements survenus sur le cluster :

```
$ kubectl get events --sort-by=.metadata.creationTimestamp
```

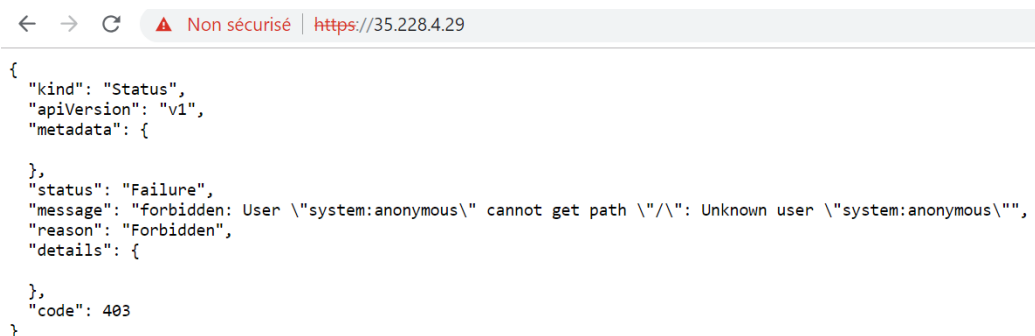
3.3. Communication avec l'API Server

La commande **kubectl cluster-info** retourne des informations utiles concernant les différents services **Kubernetes** déployés.

```
root@DESKTOP-BT28R5N:/home/pierre# kubectl cluster-info
Kubernetes master is running at https://35.228.4.29
GLBCDefaultBackend is running at https://35.228.4.29/api/v1/namespaces/kube-system/services/default-http-backend:http/proxy
Heapster is running at https://35.228.4.29/api/v1/namespaces/kube-system/services/heapster/proxy
KubeDNS is running at https://35.228.4.29/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
kubernetes-dashboard is running at https://35.228.4.29/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy
Metrics-server is running at https://35.228.4.29/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy
```

Vous pouvez notamment voir dans la capture d'écran ci-dessus que l'API Server écoute à l'adresse <https://35.228.4.29>.

Cependant, il n'est pas possible d'effectuer une requête directement sur cette URL, car il s'agit d'une connexion sécurisée HTTPS avec authentification.



```
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {
  },
  "status": "Failure",
  "message": "Forbidden: User \"system:anonymous\" cannot get path \"/\": Unknown user \"system:anonymous\"",
  "reason": "Forbidden",
  "details": {
  },
  "code": 403
}
```

Pour communiquer directement avec l'API Server il est possible d'utiliser un proxy grâce à la commande **kubectl proxy**.

Cette commande démarre un proxy vers l'API Server en s'occupant de l'authentification. De plus, il s'assure que l'on communique bien à l'API en vérifiant le certificat et sa signature à chaque requête.

- 1) Dans une nouvelle fenêtre de ligne de commande, exécutez donc la commande suivante pour démarrer un le proxy :

```
# le & permet d'exécuter la commande en fond de tâche fermer le terminal stoppera la commande
$ kubectl proxy &
Starting to serve on 127.0.0.1:8001
```

Note : Si vous travaillez avec **kubectl** sur Windows le symbole « & » n'est pas reconnu. Utilisez la commande « *start kubectl proxy* »

- 2) Il est maintenant possible de dialoguer avec l'API Server depuis l'adresse du proxy. Ouvrez un navigateur et effectuer une requête sur l'URL <http://127.0.0.1:8001> :



The screenshot shows a web browser window with the address bar set to `127.0.0.1:8001`. The page content is a JSON array of API paths:

```
{
  "paths": [
    "/api",
    "/api/v1",
    "/apis",
    "/apis/",
    "/apis/admissionregistration.k8s.io",
    "/apis/admissionregistration.k8s.io/v1beta1",
    "/apis/apiextensions.k8s.io",
    "/apis/apiextensions.k8s.io/v1beta1",
    "/apis/apiregistration.k8s.io",
    "/apis/apiregistration.k8s.io/v1beta1",
    "/apis/apps",
    "/apis/apps/v1",
    "/apis/apps/v1beta1",
    "/apis/apps/v1beta2",
    "/apis/authentication.k8s.io",
    "/apis/authentication.k8s.io/v1",
    "/apis/authentication.k8s.io/v1beta1"
  ]
}
```

Chaque ressource est catégorisée en fonction de son « *apiversion* ». Par exemple les pods, services se trouveront dans la catégorie `/api/v1`.

L'« *apiversion* » est le champ que vous renseignerez lors de la création de nouvelles ressources pour indiquer à l'API Server l'endroit où il doit trouver les informations dont il a besoin.

3) Pour connaître la version d'API utilisée, vous pouvez effectuer la requête suivante :

```
$ curl http://localhost:8001/api/
```

4) Ensuite, vous pouvez obtenir la liste des pods déployés dans le namespaces kube-system de votre cluster, en utilisant cette commande :

```
$ curl localhost:8001/api/v1/namespaces/kube-system/pods
{
  "kind": "PodList",
  "apiVersion": "v1",
  "metadata": {
    "selfLink": "/api/v1/namespaces/kube-system/pods",
    "resourceVersion": "12299"
  },
  "items": [
    {
      "metadata": {
        "name": "etcd-k8s1",
        "namespace": "kube-system",
        "selfLink": "/api/v1/namespaces/kube-system/pods/etcd-k8s1",
        "uid": "640ad255-8f11-11e8-9ba5-02f6928eb0b4",
        "resourceVersion": "1816",
        "creationTimestamp": "2018-07-24T07:15:49Z",
        "labels": {
          "component": "etcd",
          "tier": "control-plane"
        }
      },
      ...
    }
  ]
}
```

5) Cette liste de pods correspond à la liste obtenue grâce à la commande suivante :

```
$ kubectl get -n kube-system pod
```

| NAME | READY | STATUS | RESTARTS | AGE |
|------------------------------|-------|---------|----------|-----|
| etcd-k8s1 | 1/1 | Running | 1 | 2h |
| kube-apiserver-k8s1 | 1/1 | Running | 2 | 2h |
| kube-controller-manager-k8s1 | 1/1 | Running | 1 | 2h |

| | | | | |
|---------------------------|-----|---------|---|----|
| kube-dns-86f4d74b45-nl9fx | 3/3 | Running | 0 | 2h |
| kube-proxy-7f64p | 1/1 | Running | 1 | 2h |
| kube-proxy-vx62s | 1/1 | Running | 1 | 2h |
| kube-scheduler-k8s1 | 1/1 | Running | 1 | 2h |
| kubernetes-dashboard-k8s1 | 1/1 | Running | 0 | 2h |
| weave-net-www5t | 2/2 | Running | 0 | 2h |
| weave-net-zxft2 | 2/2 | Running | 2 | 2h |

3.4. Découvrir les champs de configurations API

Lorsque vous souhaitez découvrir une ressource, vous pouvez utiliser la commande « **kubectl explain** ».

- 1) Commencez par afficher l'aide, qui liste toutes les ressources qui sont supportées par l'API server :

```
$ kubectl explain -h
```

- 2) Vous pouvez par exemple afficher tous les champs que l'on peut utiliser lors de la définition d'un pod dans un fichier *yaml* ou *json* :

```
$ kubectl explain pod
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/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/api-
conventions.md#types-kinds

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

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

status        <Object>
  Most recently observed status of the pod. This data may not be
  up to date.
  Populated by the system. Read-only. More info:
  https://git.k8s.io/community/contributors/devel/api-
  conventions.md#spec-and-status
```

Comme nous pouvons le voir, cette commande retourne la version API que nous devons indiquer à l'API server lors de la création d'un pod (« **VERSION: v1** »). Puis, une description de la ressource est détaillée.

Enfin, nous retrouvons les champs évoqués tout à l'heure à savoir :

- **Metadata**
 - Contient les métadonnées de la ressource (nom, namespace...)
- **Spec**
 - Contient la description du contenu de la ressource
- **Status**
 - Contient les informations courantes associées à la ressource (statut, IP...)

Cette structure décomposée en 3 sections est la structure type d'un objet API **Kubernetes**.

Dans la pratique, l'écriture des manifests est plus légère. Lors de la création de l'objet, **Kubernetes** ajoute lui-même plusieurs informations complémentaires.

Ces champs sont suivis d'une description. À ce stade, les paramètres des différents champs ne sont pas encore indiqués. Pour cela, il faut utiliser la commande suivante :

```
$ kubectl explain pod.spec
KIND:      Pod
VERSION:   v1

RESOURCE: spec <Object>

DESCRIPTION:
    Specification of the desired behavior of the pod. More info:
    https://git.k8s.io/community/contributors/devel/api-
    conventions.md#spec-and-status

    PodSpec is a description of a pod.

FIELDS:
    activeDeadlineSeconds          <integer>
        Optional duration in seconds the pod may be active on the node
        relative to
        StartTime before the system will actively try to mark it failed
        and kill
        associated containers. Value must be a positive integer.

    affinity                       <Object>
        If specified, the pod's scheduling constraints

    automountServiceAccountToken <boolean>
        AutomountServiceAccountToken indicates whether a service account
        token
        should be automatically mounted.

    containers                     <[]Object> -required-
```

```
List of containers belonging to the pod. Containers cannot
currently be
    added or removed. There must be at least one container in a Pod.
Cannot be
    updated.
...
```

Vous pouvez aller ainsi dans le détail des arguments :

```
$ kubectl explain pod.spec.containers
$ kubectl explain pod.spec.containers.image
```

4. Tour d'horizon de l'interface graphique

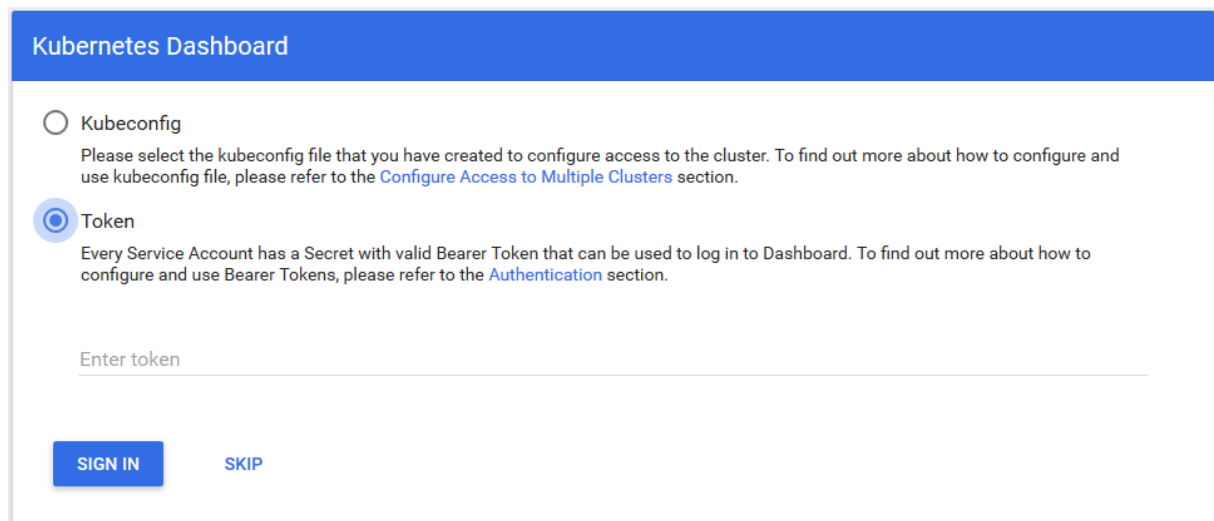
4.1. Dashboard Kubernetes

- 1) Pour configurer l'accès au dashboard natif de **Kubernetes**, utilisez la commande suivante pour lancer un nouveau proxy :

```
$ kubectl proxy &
```

- 2) Vous pouvez maintenant joindre le dashboard sur l'adresse URL suivante :

<http://localhost:8001/ui>



Vous arrivez sur une page de login permettant de sécuriser l'accès au dashboard.

Cependant, il est possible de désactiver la connexion sécurisée au dashboard. Dans un premier temps, nous allons utiliser cette méthode, qui est évidemment déconseillée.

- 3) Pour cela, vous allez donc créer un fichier yaml contenant le code suivant :

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: kubernetes-dashboard
  labels:
    k8s-app: kubernetes-dashboard
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
```

```
name: cluster-admin
subjects:
- kind: ServiceAccount
  name: kubernetes-dashboard
  namespace: kube-system
```

Le contenu de ce fichier permet de gérer les autorisations, sujet traité plus tard dans la formation.

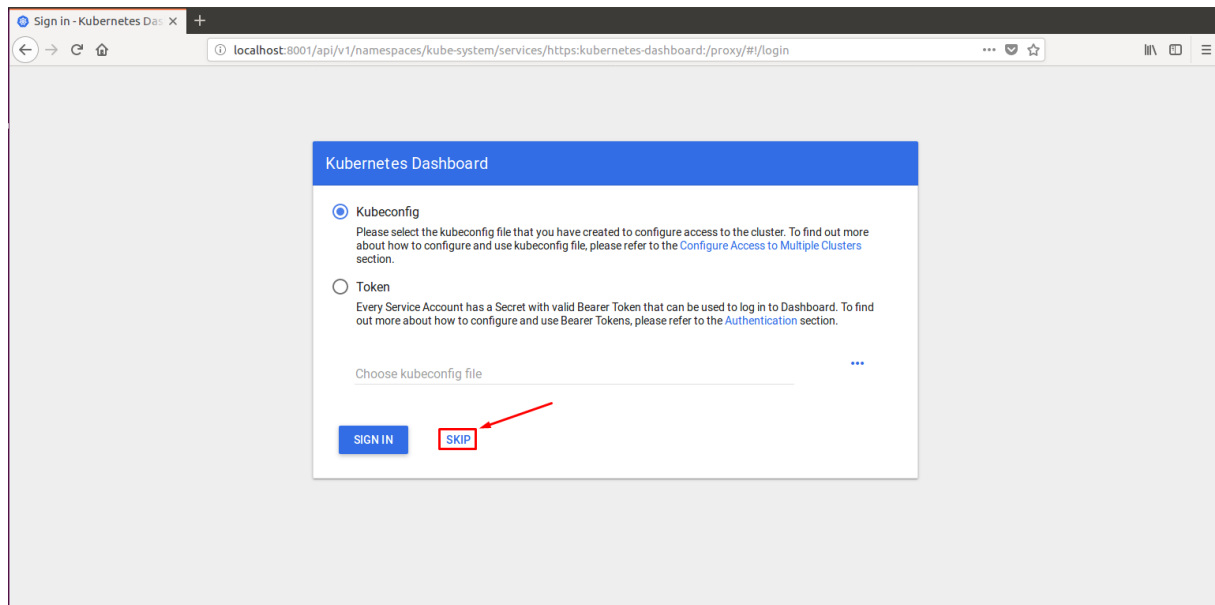
- 4) Une fois le fichier créé, utilisez la commande suivante pour appliquer les modifications auprès de l'API Server :

```
$ kubectl create -f unsecure.yaml
```

- 5) Reconnectez-vous au dashboard par le biais d'un navigateur web. Pour rappel, l'URL à utiliser est la suivante :

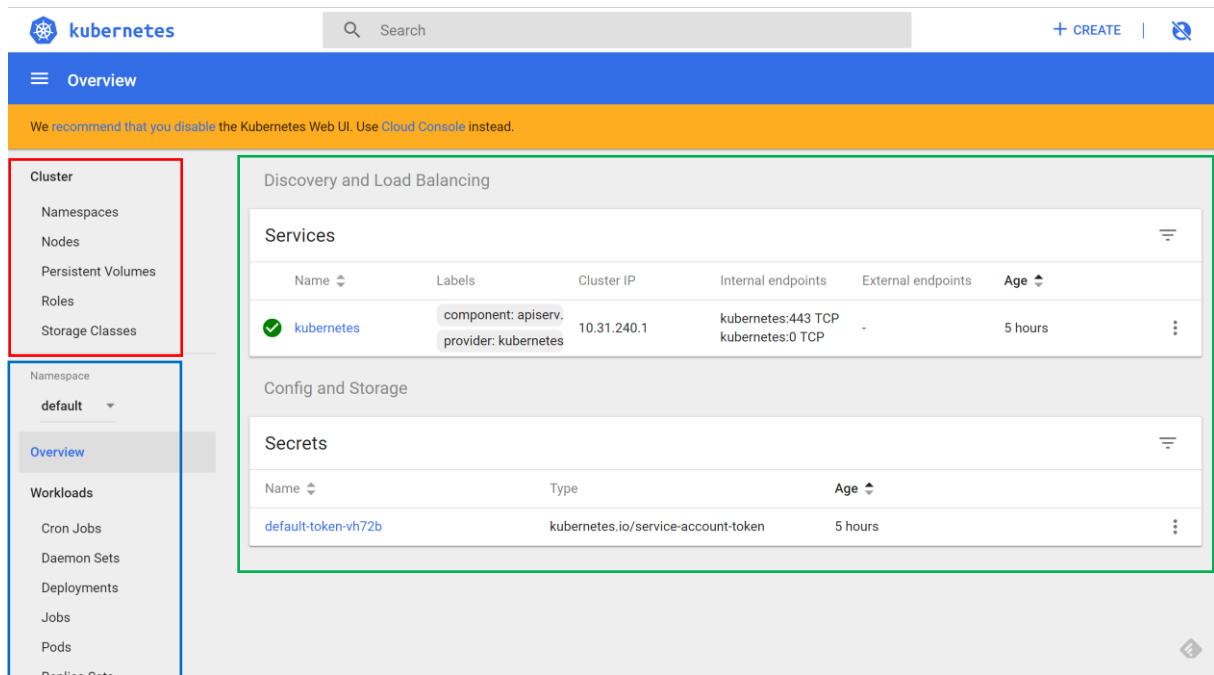
<http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#/login>

Il est désormais possible de cliquer sur « *Skip* » et accéder au dashboard sans identifiants :



Remarque :

- Cette méthode n'est évidemment pas recommandée dans un environnement de production, car tout utilisateur qui dispose d'un accès à l'URL peut se connecter avec toutes les autorisations (créer, supprimer, afficher, interagir ...).



The screenshot shows the Kubernetes dashboard interface. On the left, there is a sidebar with a red box highlighting the 'Cluster' and 'Namespaces' categories, and a blue box highlighting the 'default' namespace dropdown and the 'Overview' link. The main content area is divided into sections: 'Discovery and Load Balancing' containing a 'Services' table, and 'Config and Storage' containing a 'Secrets' table.

| Name | Labels | Cluster IP | Internal endpoints | External endpoints | Age |
|------------|---|-------------|--|--------------------|---------|
| kubernetes | component: apiserv. provider: kubernetes | 10.31.240.1 | kubernetes:443 TCP kubernetes:0 TCP | - | 5 hours |

| Name | Type | Age |
|---------------------|-------------------------------------|---------|
| default-token-vh72b | kubernetes.io/service-account-token | 5 hours |

Lorsque vous vous êtes connecté au dashboard, vous avez pu observer un bandeau à gauche qui permet de naviguer entre les différents menus (encadré en rouge et bleu) et au centre (en vert) les informations contenues dans ces catégories.

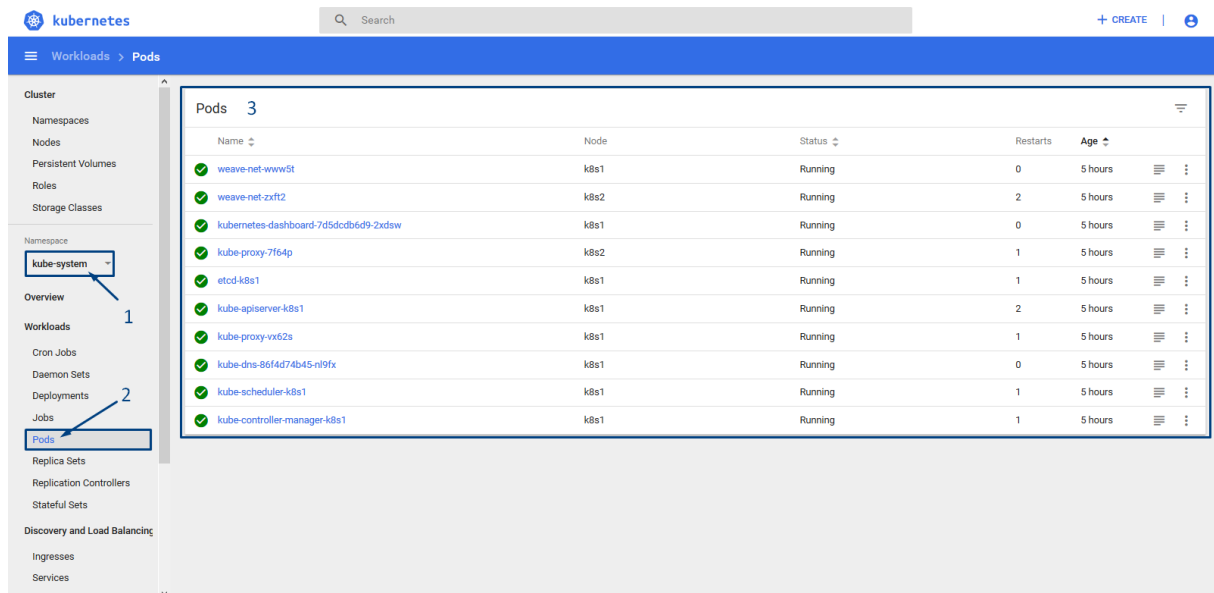
En rouge, nous avons les catégories n'appartenant pas à un namespace **Kubernetes**.

En bleu, nous avons toutes les catégories appartenant à un namespace et qui appartiennent au namespace « **default** ». Il est possible de changer de namespace en cliquant dessus.

Lors de la connexion, nous arrivons dans l'onglet « **overview** » qui liste toutes les ressources du namespace sélectionné et le statut du namespace.

6) Changez de namespace et sélectionnez le namespace « **kube-system** ».

7) Allez dans la catégorie « **Pods** ». Vous retrouvez ici une liste de tous les pods se trouvant à ce namespace :



Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

Namespace: kube-system

Overview

Workloads

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Discovery and Load Balancing

Ingresses

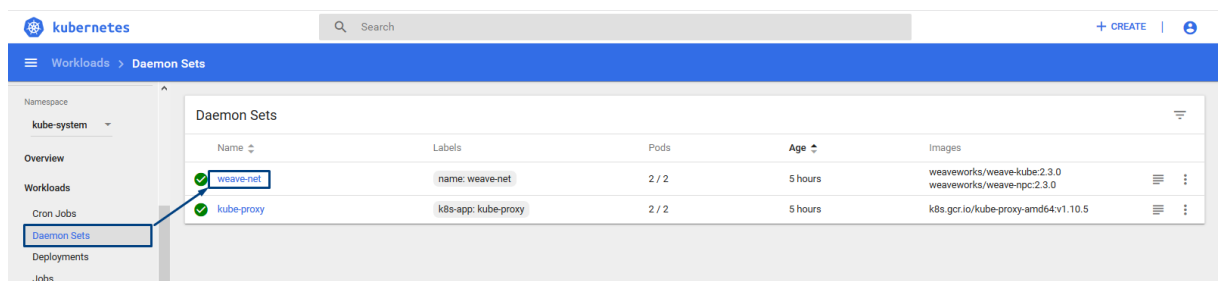
Services

Pods 3

| Name | Node | Status | Restarts | Age |
|--------------------------------------|------|---------|----------|---------|
| weave-net-www5t | k8s1 | Running | 0 | 5 hours |
| weave-net-zxf2 | k8s2 | Running | 2 | 5 hours |
| kubernetes-dashboard-7d5dcb6d9-2xdsw | k8s1 | Running | 0 | 5 hours |
| kube-proxy-7f64p | k8s2 | Running | 1 | 5 hours |
| etcd-k8s1 | k8s1 | Running | 1 | 5 hours |
| kube-apiserver-k8s1 | k8s1 | Running | 2 | 5 hours |
| kube-proxy-vx62s | k8s1 | Running | 1 | 5 hours |
| kube-dns-86f4d74b45-nl9fx | k8s1 | Running | 0 | 5 hours |
| kube-scheduler-k8s1 | k8s1 | Running | 1 | 5 hours |
| kube-controller-manager-k8s1 | k8s1 | Running | 1 | 5 hours |

Note : Il est possible de cliquer sur chacun de ces pods pour avoir de plus amples détails sur chacun.

- Par exemple, allez dans la catégorie « *DaemonSets* » et affichez la liste des ressources présentes.



Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

Namespace: kube-system

Overview

Workloads

Cron Jobs

Daemon Sets

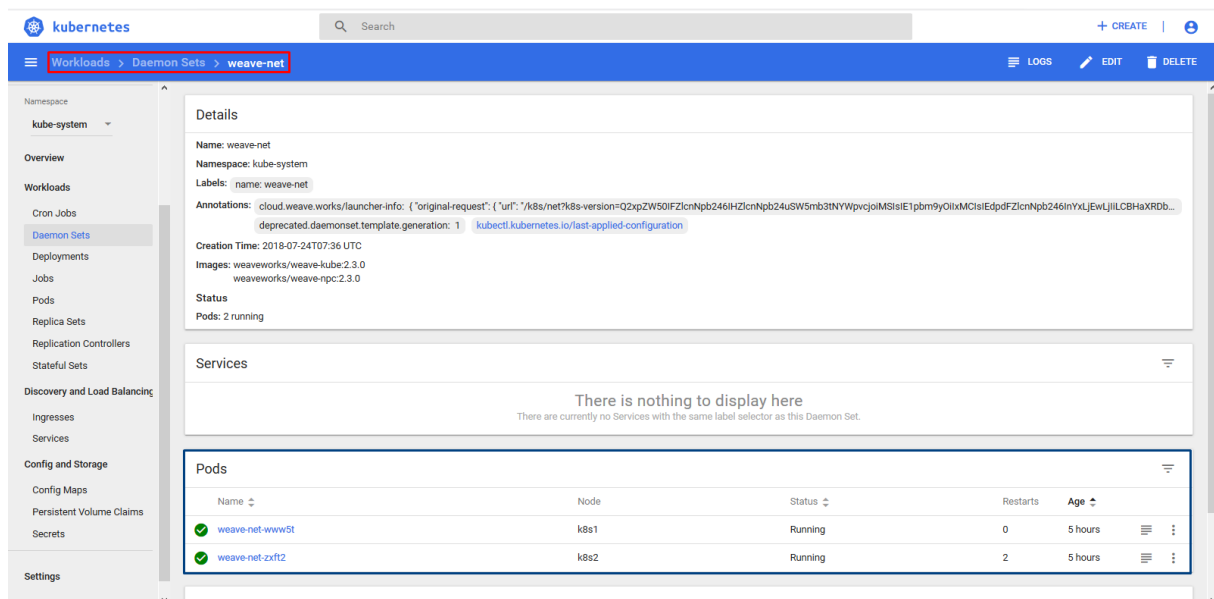
Deployments

Jobs

Daemon Sets

| Name | Labels | Pods | Age | Images |
|------------|---------------------|-------|---------|---|
| weave-net | name: weave-net | 2 / 2 | 5 hours | weaveworks/weave-kube:2.3.0 weaveworks/weave-npc:2.3.0 |
| kube-proxy | k8s-app: kube-proxy | 2 / 2 | 5 hours | k8s.gcr.io/kube-proxy-amd64:v1.10.5 |

- Cliquez sur une des ressources pour apercevoir les « *pods* » qui appartiennent au « *DaemonSet* » en question.



Details

Name: weave-net
Namespace: kube-system
Labels: name: weave-net
Annotations: cloud.weave.works/launcher-info: {"url": "https://k8s.net?k8s-version=Q2xpZW50IFZlcjNpb246H2ZlcjNpb24uSW5mb3RNYWpvcjoMSisE1pbm9yOliXMCisEdpdZlcjNpb246inYxLjEwLjILCBHaXRDb... deprecated.daemonset.template.generation: 1 kubecti.kubernetes.io/last-applied-configuration

Creation Time: 2018-07-24T07:36 UTC
Images: weaveworks/weave-kube:2.3.0
weaveworks/weave-npc:2.3.0

Status
Pods: 2 running

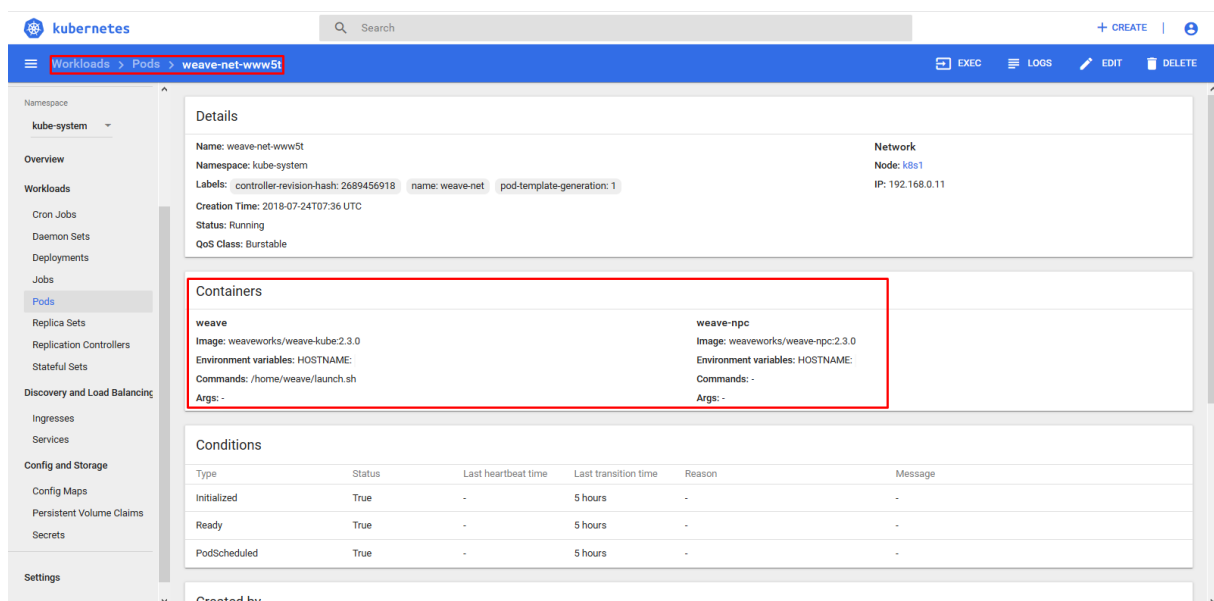
Services

There is nothing to display here
There are currently no Services with the same label selector as this Daemon Set.

Pods

| Name | Node | Status | Restarts | Age |
|-----------------|------|---------|----------|---------|
| weave-net-www5t | k8s1 | Running | 0 | 5 hours |
| weave-net-cxft2 | k8s2 | Running | 2 | 5 hours |

c. Enfin, cliquez sur un des « *Pods* » pour apercevoir les « *Containers* » le composant.



Details

Name: weave-net-www5t
Namespace: kube-system
Labels: controller-revision-hash: 2689456918 name: weave-net pod-template-generation: 1
Creation Time: 2018-07-24T07:36 UTC
Status: Running
QoS Class: Burstable

Network
Node: k8s1
IP: 192.168.0.11

Containers

weave
Image: weaveworks/weave-kube:2.3.0
Environment variables: HOSTNAME:
Commands: /home/weave/launch.sh
Args: -

weave-npc
Image: weaveworks/weave-npc:2.3.0
Environment variables: HOSTNAME:
Commands: -
Args: -

Conditions

| Type | Status | Last heartbeat time | Last transition time | Reason | Message |
|--------------|--------|---------------------|----------------------|--------|---------|
| Initialized | True | - | 5 hours | - | - |
| Ready | True | - | 5 hours | - | - |
| PodScheduled | True | - | 5 hours | - | - |

Note :

- Il est possible d'utiliser la « *barre de recherche* » pour rechercher une chaîne de caractères parmi toutes les ressources du « *namespace* » sélectionnés.
- L'icône « + » permet de créer de nouvelles ressources à partir de *yaml*, *json*, fichiers ou formulaire.

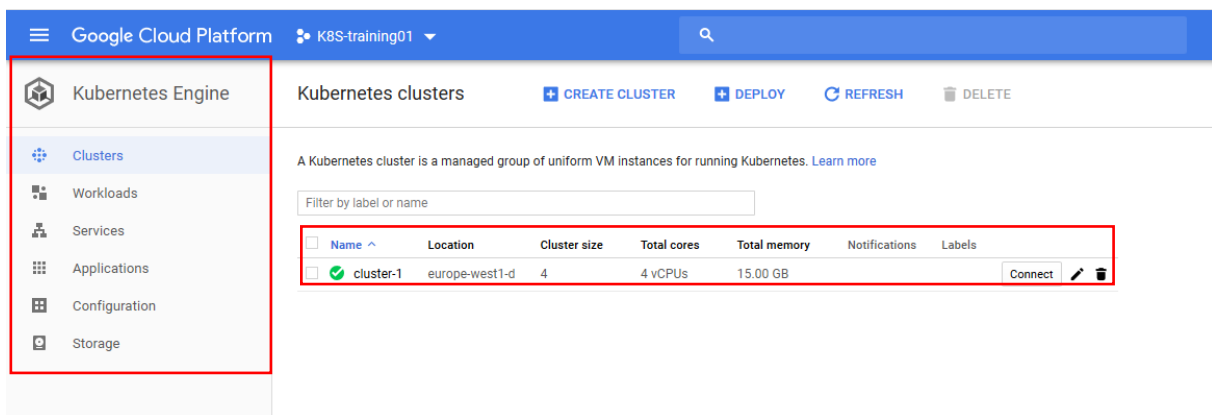
4.2. Dashboard GKE



Cette interface est le dashboard conseillé pour la gestion de votre cluster K8S déployé sur GKE. Google a développé son propre dashboard en y intégrant notamment ses outils de monitoring, logging et sa marketplace.

4.2.1. Clusters

L'onglet « **Workloads** » est l'onglet contenant tous les clusters **Kubernetes** déployés au sein de votre projet GCP.

Au centre, vous devriez donc voir apparaître votre cluster ainsi que les informations principales qui lui sont associées. De plus, vous pouvez apercevoir sur la gauche plusieurs onglets liés à l'utilisation de **Kubernetes Engine**.




| <input type="checkbox"/> | Name ^ | Location | Cluster size | Total cores | Total memory | Notifications | Labels |
|-------------------------------------|-----------|----------------|--------------|-------------|--------------|---------------|---|
| <input checked="" type="checkbox"/> | cluster-1 | europa-west1-d | 4 | 4 vCPUs | 15.00 GB | | Connect   |

Dans les parties suivantes, vous naviguerez au travers de ces onglets pour un rapide tour d'horizon.

Note : Vous pouvez toujours utiliser le dashboard par défaut de **Kubernetes** qui est déployé à la création de votre cluster. Cependant, celui-ci ne dispose pas de la même intégration avec le reste de GCP.

4.2.2. Workloads

L'onglet « **Workloads** » est l'onglet contenant toutes les charges de travail qui s'exécutent sur votre cluster. Autrement dit, vous retrouverez ici, toutes les ressources **Kubernetes** déployées comme par exemple des **Pods**, **Namespaces**, **Deployments**, **ReplicatSet**, ...


Kubernetes Engine

Clusters

Workloads

Services

Applications

Configuration

Storage

Marketplace

Workloads

REFRESH

DEPLOY

Workloads are deployable units of computing that can be created and managed in a cluster.

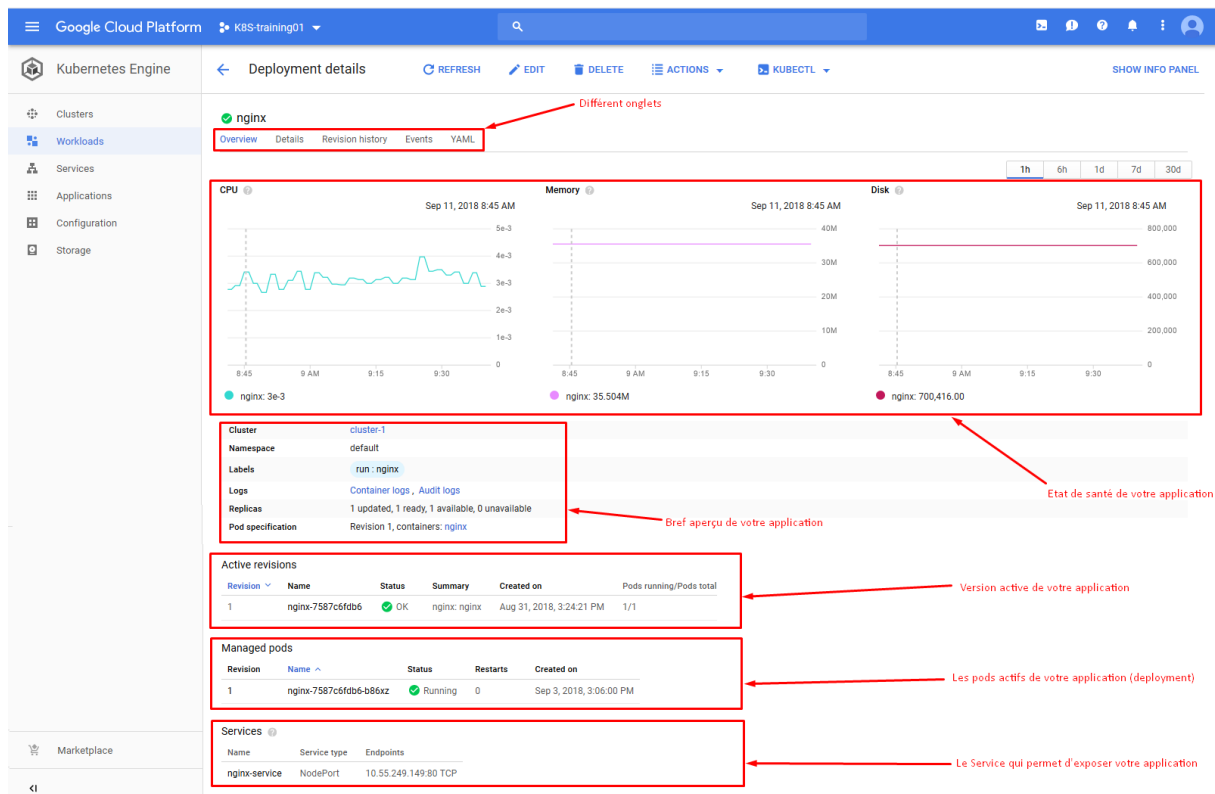
Filter workloads

Columns

| Name | Status | Type | Pods | Namespace | Cluster |
|---|---------|------------|------|-------------|---------|
| event-exporter-v0.1.9 | OK | Deployment | 1/1 | kube-system | k8s-lab |
| fluentd-gcp-v2.0.17 | OK | Daemon Set | 3/3 | kube-system | k8s-lab |
| heapster-v1.5.2 | OK | Deployment | 1/1 | kube-system | k8s-lab |
| kube-dns | OK | Deployment | 2/2 | kube-system | k8s-lab |
| kube-dns-autoscaler | OK | Deployment | 1/1 | kube-system | k8s-lab |
| kube-proxy-gke-k8s-lab-default-pool-4fd48f3f-pgnc | Running | Pod | 1/1 | kube-system | k8s-lab |
| kube-proxy-gke-k8s-lab-default-pool-4fd48f3f-r0w2 | Running | Pod | 1/1 | kube-system | k8s-lab |
| kube-proxy-gke-k8s-lab-default-pool-4fd48f3f-t5p0 | Running | Pod | 1/1 | kube-system | k8s-lab |
| kubernetes-dashboard | OK | Deployment | 1/1 | kube-system | k8s-lab |
| l7-default-backend | OK | Deployment | 1/1 | kube-system | k8s-lab |
| metadata-proxy-v0.1 | OK | Daemon Set | 0/0 | kube-system | k8s-lab |

Vous noterez la présence de la barre de recherche permettant simplifier et d'affiner les résultats affichés.

Si vous cliquez sur une des ressources, vous pouvez apercevoir un certain nombre d'informations sur celle-ci, ainsi que différentes métriques la concernant :

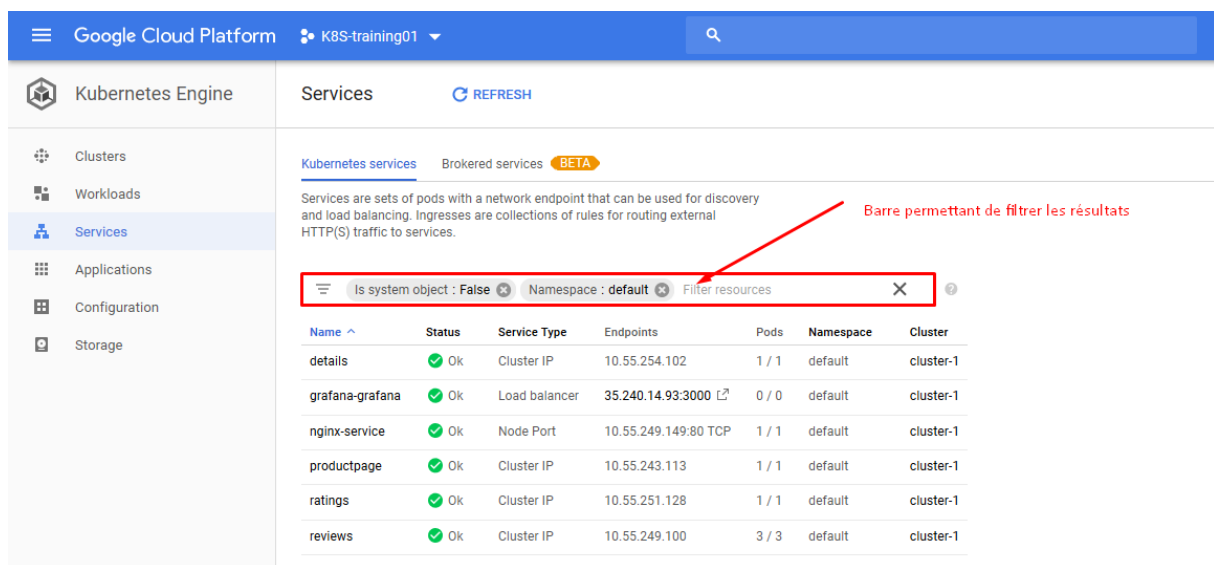


Note : Ce panneau ne remplace pas l'utilisation d'un outil de monitoring/alerting tiers.

4.2.3. Services

Dans cet onglet, vous pouvez observer l'ensemble des applications exposées par le biais d'un **Service**. La notion de **Services** vous sera expliquée lors d'un des ateliers de cette formation.

Cet onglet donne donc un aperçu des services déployés sur votre cluster ainsi que les informations associées.



Google Cloud Platform K8S-training01

Kubernetes Engine Services REFRESH

Kubernetes services Brokered services BETA

Services are sets of pods with a network endpoint that can be used for discovery and load balancing. Ingresses are collections of rules for routing external HTTP(S) traffic to services.

Is system object : False Namespace : default Filter resources

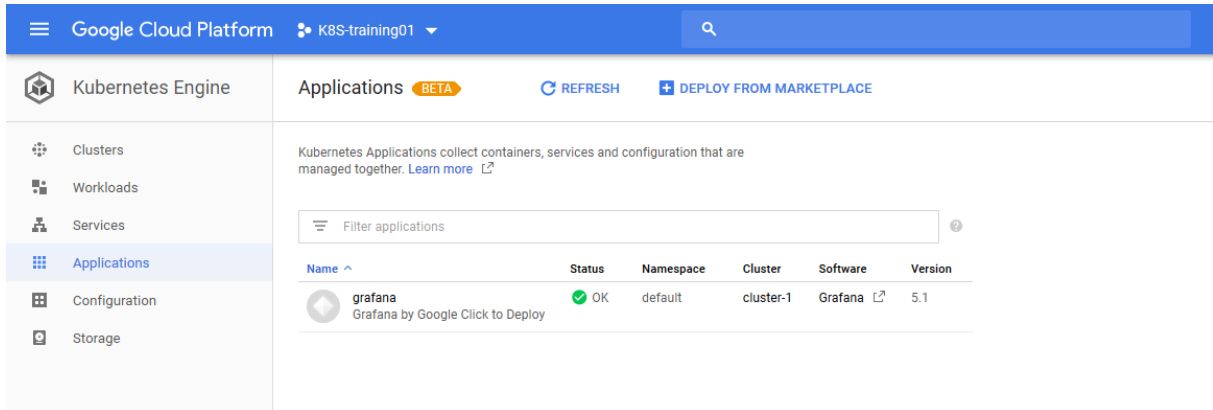
| Name | Status | Service Type | Endpoints | Pods | Namespace | Cluster |
|-----------------|--------|---------------|----------------------|-------|-----------|-----------|
| details | Ok | Cluster IP | 10.55.254.102 | 1 / 1 | default | cluster-1 |
| grafana-grafana | Ok | Load balancer | 35.240.14.93:3000 | 0 / 0 | default | cluster-1 |
| nginx-service | Ok | Node Port | 10.55.249.149:80 TCP | 1 / 1 | default | cluster-1 |
| productpage | Ok | Cluster IP | 10.55.243.113 | 1 / 1 | default | cluster-1 |
| ratings | Ok | Cluster IP | 10.55.251.128 | 1 / 1 | default | cluster-1 |
| reviews | Ok | Cluster IP | 10.55.249.100 | 3 / 3 | default | cluster-1 |

Comme dans l'onglet « **Workloads** », vous pouvez voir qu'il est possible d'utiliser la barre de recherche pour affiner les résultats.

4.2.4. Applications

L'onglet « **Applications** » regroupe toutes les applications que vous avez déployées depuis la marketplace de GCP.

Dans cet exemple, une application « *grafana* » a été déployée :



Google Cloud Platform K8S-training01

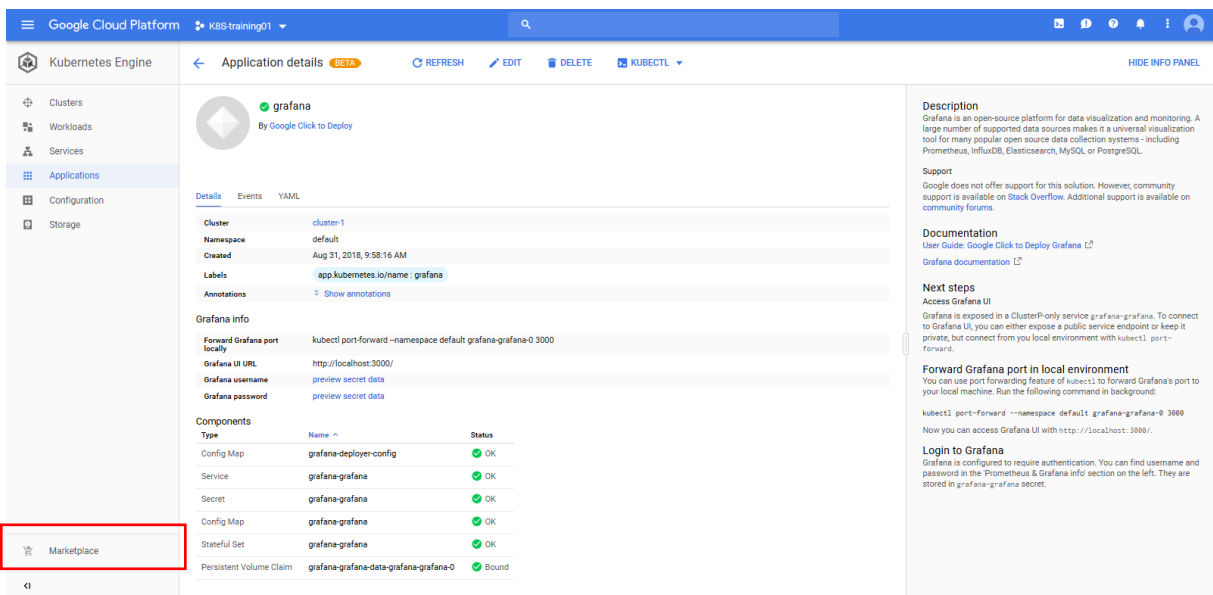
Kubernetes Engine Applications **BETA** REFRESH DEPLOY FROM MARKETPLACE

Kubernetes Applications collect containers, services and configuration that are managed together. [Learn more](#)

Filter applications

| Name | Status | Namespace | Cluster | Software | Version |
|--|--------|-----------|-----------|----------|---------|
| grafana Grafana by Google Click to Deploy | OK | default | cluster-1 | Grafana | 5.1 |

Si vous cliquez dessus, vous obtenez de nombreuses informations à son sujet comme les ressources **Kubernetes** (pods, services, ...) qui la constitue :



Google Cloud Platform K8S-training01

Kubernetes Engine Application details **BETA** REFRESH EDIT DELETE KUBECTL

grafana
By Google Click to Deploy

Details Events YAML

Cluster: cluster-1
Namespace: default
Created: Aug 31, 2018, 9:58:16 AM
Labels: app.kubernetes.io/name: grafana
Annotations: Show annotations

Grafana info

Forward Grafana port locally: kubectl port-forward --namespace default grafana-grafana-0 3000
Grafana UI URL: http://localhost:3000/
Grafana username: preview secret data
Grafana password: preview secret data

Components

| Type | Name | Status |
|-------------------------|--|--------|
| Config Map | grafana-deployer-config | OK |
| Service | grafana-grafana | OK |
| Secret | grafana-grafana | OK |
| Config Map | grafana-grafana | OK |
| Stateful Set | grafana-grafana | OK |
| Persistent Volume Claim | grafana-grafana-data-grafana-grafana-0 | Bound |

Marketplace

Description
Grafana is an open-source platform for data visualization and monitoring. A large number of supported data sources makes it a universal visualization tool for many popular open source data collection systems - including Prometheus, InfluxDB, Elasticsearch, MySQL or PostgreSQL.

Support
Google does not offer support for this solution. However, community support is available on [Stack Overflow](#). Additional support is available on [community forums](#).

Documentation
[User Guide: Google Click to Deploy Grafana](#)
[Grafana documentation](#)

Next steps
Access Grafana UI
Grafana is exposed in a ClusterIP-only service grafana-grafana. To connect to Grafana UI, you can either expose a public service endpoint or keep it private, but connect from you local environment with kubectl port-forward.

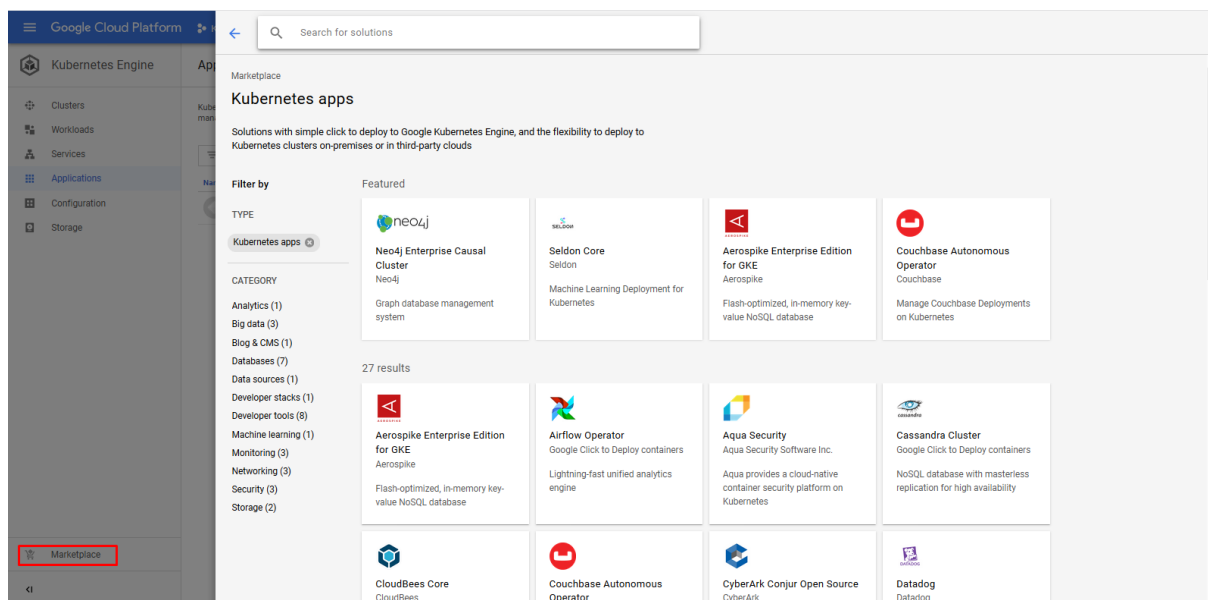
Forward Grafana port in local environment
You can use port forwarding feature of kubectl to forward Grafana's port to your local machine. Run the following command in background:
kubectl port-forward --namespace default grafana-grafana-0 3000
Now you can access Grafana UI with http://localhost:3000/.

Login to Grafana
Grafana is configured to require authentication. You can find username and password in the 'Prometheus & Grafana info' section on the left. They are stored in grafana-grafana secret.

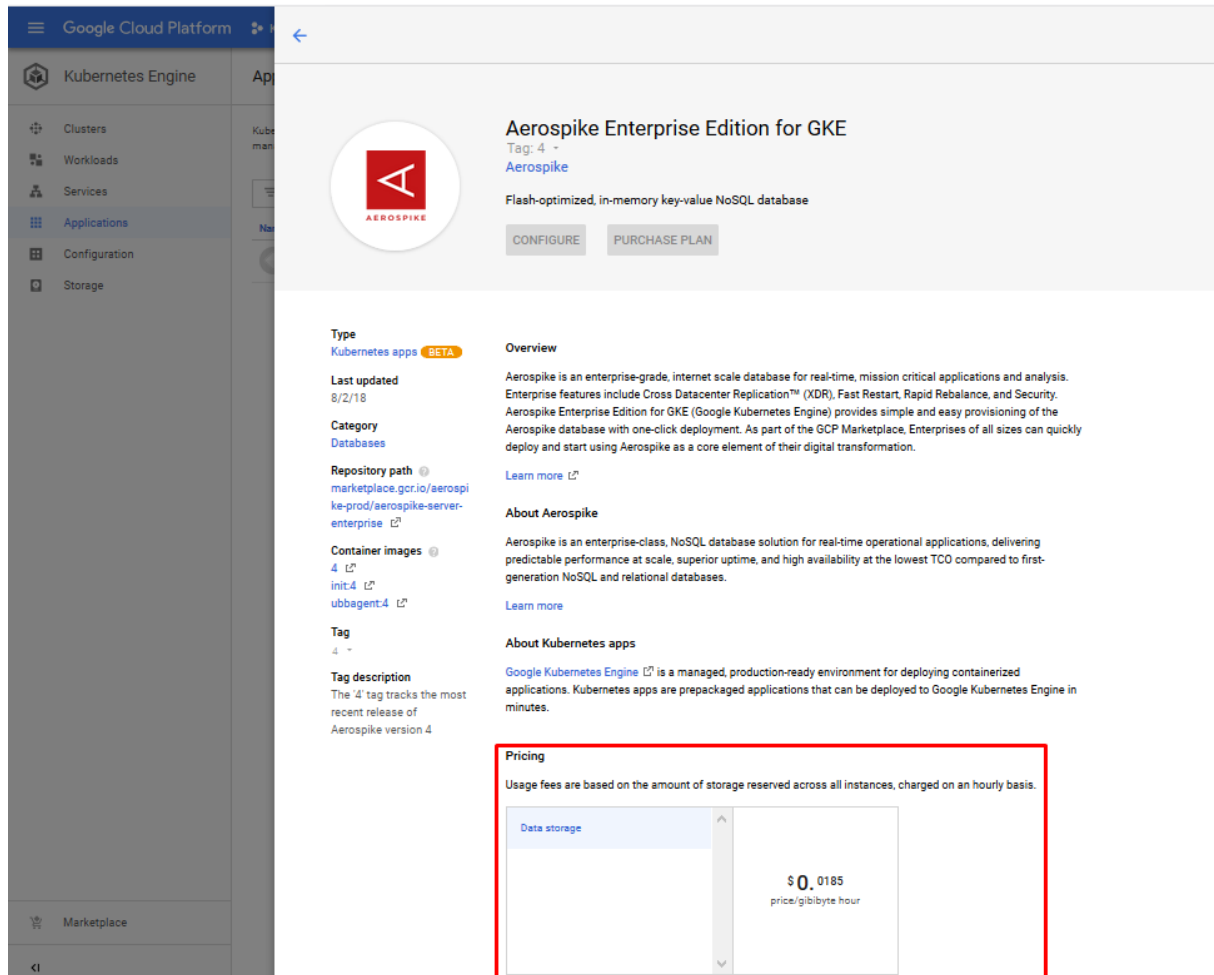
Note :

- Vous pouvez accéder à la marketplace de GCP à tout moment depuis l'onglet « **Marketplace** » (en bas). Pour rappel, la marketplace contient de nombreuses applications déjà prêtes à l'emploi et facilement déployables.
- N'hésitez pas à tester le déploiement d'une application de votre choix pour bien comprendre le fonctionnement du dashboard.

Vous retrouverez à la fois des applications gratuites ou payantes. Si celles-ci sont payantes, le type de facturation est spécifié dans les détails de l'application :



Exemple d'une application payante :



Aerospike Enterprise Edition for GKE
Tag: 4
Aerospike
Flash-optimized, in-memory key-value NoSQL database

[CONFIGURE](#) [PURCHASE PLAN](#)

Type
Kubernetes apps **BETA**

Last updated
8/2/18

Category
Databases

Repository path
marketplace.gcp.io/aerospike-prod/aerospike-server-enterprise

Container images
4
init:4
ubbagent:4

Tag
4

Tag description
The '4' tag tracks the most recent release of Aerospike version 4

Overview
Aerospike is an enterprise-grade, internet scale database for real-time, mission critical applications and analysis. Enterprise features include Cross Datacenter Replication™ (XDR), Fast Restart, Rapid Rebalance, and Security. Aerospike Enterprise Edition for GKE (Google Kubernetes Engine) provides simple and easy provisioning of the Aerospike database with one-click deployment. As part of the GCP Marketplace, Enterprises of all sizes can quickly deploy and start using Aerospike as a core element of their digital transformation.

[Learn more](#)

About Aerospike
Aerospike is an enterprise-class, NoSQL database solution for real-time operational applications, delivering predictable performance at scale, superior uptime, and high availability at the lowest TCO compared to first-generation NoSQL and relational databases.

[Learn more](#)

About Kubernetes apps
[Google Kubernetes Engine](#) is a managed, production-ready environment for deploying containerized applications. Kubernetes apps are prepackaged applications that can be deployed to Google Kubernetes Engine in minutes.

Pricing
Usage fees are based on the amount of storage reserved across all instances, charged on an hourly basis.

| Data storage | Price |
|--------------|---------------------------------|
| | \$0.0185 price/gibibyte hour |

4.2.5. Configuration

Cet onglet permet de lister l'ensemble des **Secrets** et **Configmaps** déployés au sein de votre cluster. Ces informations sont potentiellement des informations sensibles, notamment si celles-ci sont des informations d'authentification et autorisations.

Les notions de **Secrets** et **Configmaps** vous seront expliquées lors d'un des ateliers de cette formation.

Google Cloud Platform K8S-training01

Kubernetes Engine Configuration REFRESH

Secrets are sensitive pieces of information, like passwords, keys and tokens. Config maps are designed to store information that is not sensitive, like environment variables, command line arguments, and configuration files.

Secrets respect access control and are not visible to users without read permissions Dismiss

Is system object : False Filter secrets and config maps

| Name ^ | Type | Namespace | Cluster |
|-------------------------------------|-------------------------|--------------|-----------|
| default-token-25bc2 | Secret: service account | cloud-apps | cluster-1 |
| default-token-2gmhz | Secret: service account | google-oauth | cluster-1 |
| default-token-sm8nr | Secret: service account | istio-system | cluster-1 |
| gcp-svc-account-secret | Secret | google-oauth | cluster-1 |
| gitlab-gitlab-secret | Secret | default | cluster-1 |
| gitlab-gitlab-initial-root-password | Secret | default | cluster-1 |
| gitlab-gitlab-runner-secret | Secret | default | cluster-1 |
| gitlab-gitlab-shell-host-keys | Secret | default | cluster-1 |
| gitlab-gitlab-shell-secret | Secret | default | cluster-1 |
| gitlab-minio-secret | Secret | default | cluster-1 |
| gitlab-postgresql-password | Secret | default | cluster-1 |
| gitlab-rails-secret | Secret | default | cluster-1 |
| gitlab-redis-secret | Secret | default | cluster-1 |
| gitlab-registry-secret | Secret | default | cluster-1 |

4.2.6. Storage

Cet onglet permet d'afficher l'ensemble des **PersistentVolumeClaim** et **StorageClass** déployés au sein de votre cluster. Autrement dit, vous retrouverez ici l'ensemble des configurations associées au stockage.

Les notions de **PersistentVolumeClaim** et **StorageClass** vous seront expliquées lors d'un des ateliers de cette formation.

Google Cloud Platform K8S-training01

Kubernetes Engine Storage REFRESH

Persistent volume claims Storage classes

Persistent volume claims are requests for storage of specific size and access mode. [Learn more](#)

Filter persistent volume claims

| Name ^ | Phase | Volume | Storage class | Namespace | Cluster |
|------------------------|-------|--|---------------|-----------|-----------|
| grafana-data-grafana-0 | Bound | pvc-a99822b5-acf3-11e8-81cd-42010a8400b7 | standard | default | cluster-1 |