English Report – 2nd Lesson

José MASSE

LP A2SR - Communication en langue anglaise

Mathieu Perochon

2023

2nd course

Kubernetes

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It is good because it allows you to easily manage and scale your containerized applications in a cluster of nodes, provides fault-tolerance, self-healing, and load balancing features, and enables you to abstract away the underlying infrastructure, making it easier to develop and deploy applications in a consistent and efficient manner.

Mission 1 : First stop with Kubernetes

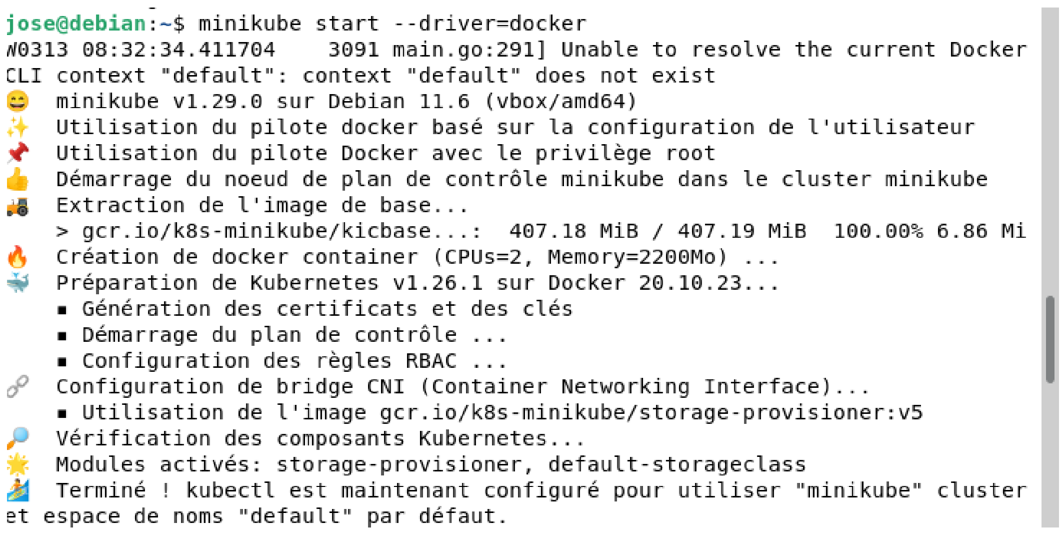
The purpose of this section was to install Kubernetes and deploy a dashboard to manage pods more easily.

The problem I encountered is that I am using my company's computer and the GPOs blocked me from connecting to Windows Update to download the necessary features.

That's why I decided to do this work using Minikube on Linux.

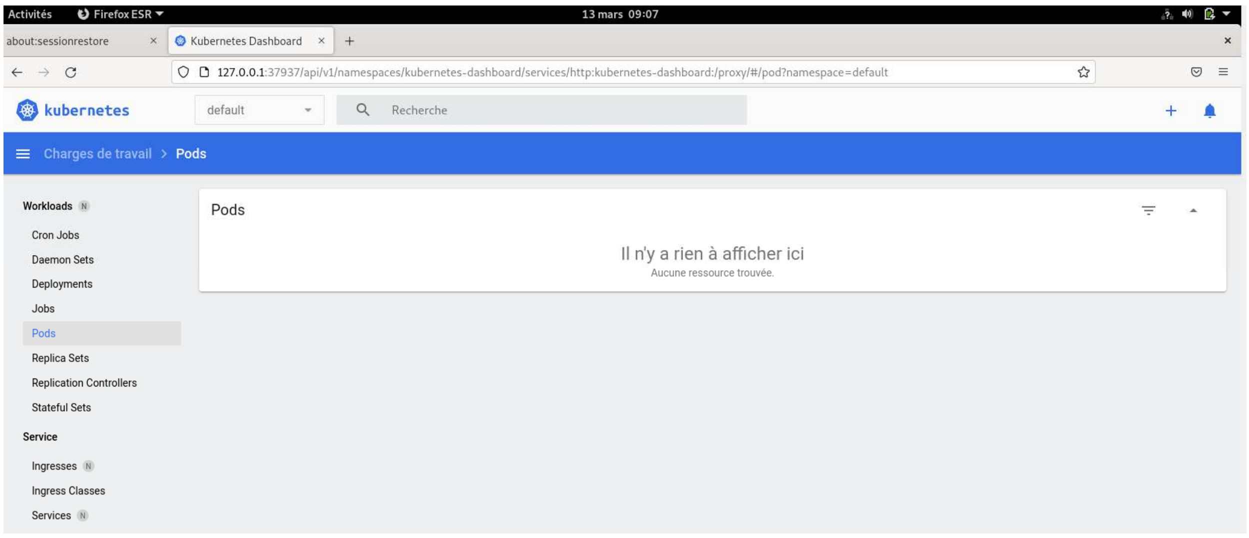
Firstly I installed minikube with docker driver because by default it tried to launch with virtualization driver and it did'nt work for me.

Then I started the minikube.



```
jose@debian:~$ minikube start --driver=docker
W0313 08:32:34.411704    3091 main.go:291] Unable to resolve the current Docker
CLI context "default": context "default" does not exist
😄  minikube v1.29.0 sur Debian 11.6 (vbox/amd64)
✨  Utilisation du pilote docker basé sur la configuration de l'utilisateur
📌  Utilisation du pilote Docker avec le privilège root
👍  Démarrage du noeud de plan de contrôle minikube dans le cluster minikube
🚜  Extraction de l'image de base...
    > gcr.io/k8s-minikube/kicbase...:  407.18 MiB / 407.19 MiB  100.00% 6.86 Mi
🔥  Création de docker container (CPUs=2, Memory=2200Mo) ...
🐳  Préparation de Kubernetes v1.26.1 sur Docker 20.10.23...
    ▪ Génération des certificats et des clés
    ▪ Démarrage du plan de contrôle ...
    ▪ Configuration des règles RBAC ...
🔗  Configuration de bridge CNI (Container Networking Interface)...
    ▪ Utilisation de l'image gcr.io/k8s-minikube/storage-provisioner:v5
🔎  Vérification des composants Kubernetes...
🌟  Modules activés: storage-provisioner, default-storageclass
🏄  Terminé ! kubectl est maintenant configuré pour utiliser "minikube" cluster
et espace de noms "default" par défaut.
```

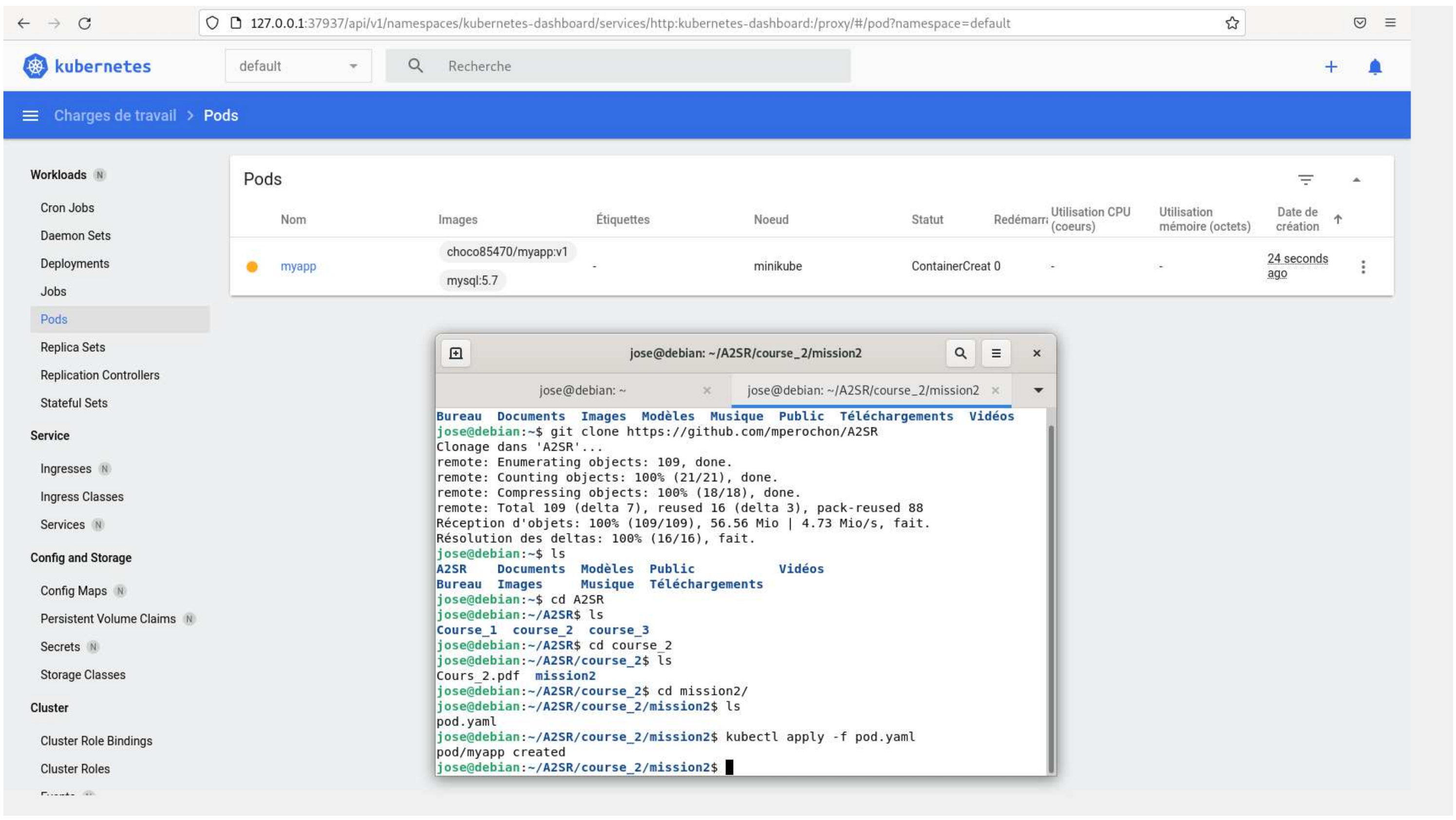
After that, I was able to launch a Kubernetes Dashboard using command Minikube Dashboard. I accessed my dashboard page in localhost using Firefox web browser.



Second mission : Deploy your first application

I cloned Mathieu's github repository and launch the pod.yml file with the kubectl command.

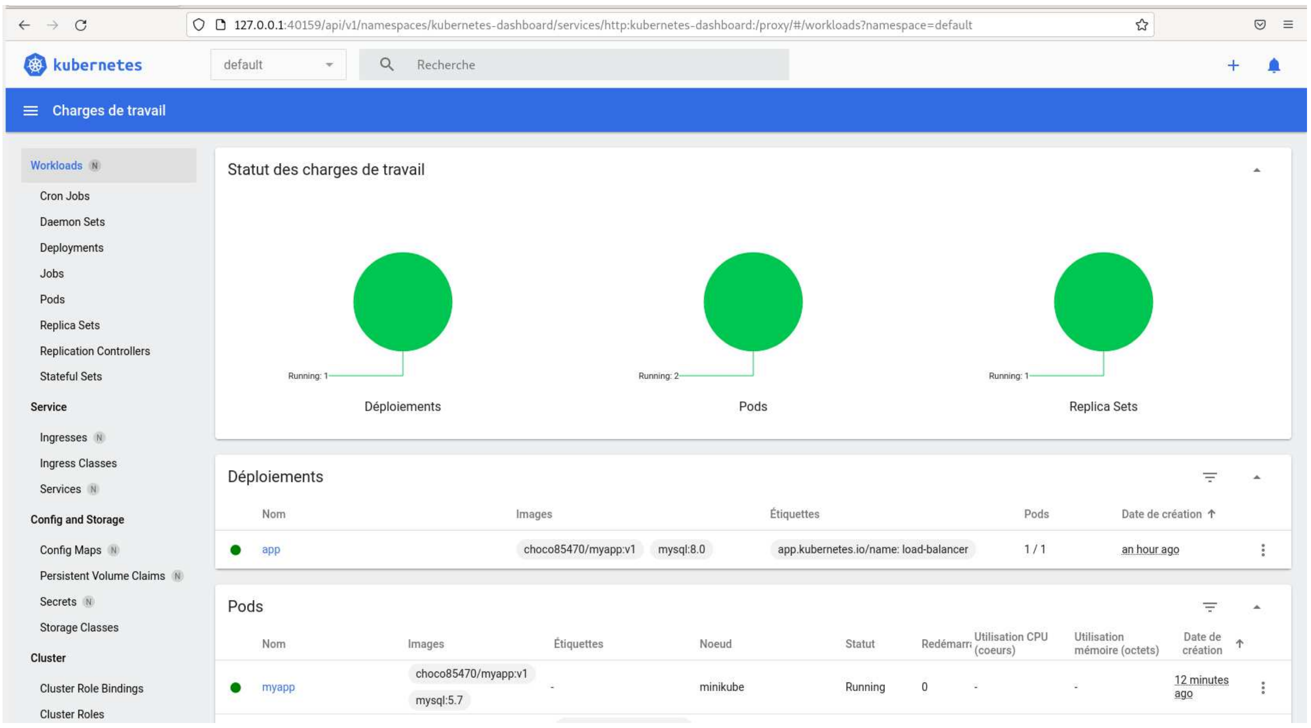
It didn't work because there was an error in the syntax of the document. Thanks to ChatGPT (which is very usefull in this case), I modified the document and succeded to launch my first application.



Pods

	Nom	Images	Étiquettes	Noeud	Statut	Redémarr	Utilisation CPU (coeurs)	Utilisation mémoire (octets)	Date de création
●	myapp	choco85470/myapp:v1 mysql:5.7	-	minikube	ContainerCreat	0	-	-	24 seconds ago

```
Bureau  Documents  Images  Modèles  Musique  Public  Téléchargements  Vidéos
jose@debian:~$ git clone https://github.com/mperochon/A2SR
Clonage dans 'A2SR'...
remote: Enumerating objects: 109, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 109 (delta 7), reused 16 (delta 3), pack-reused 88
Réception d'objets: 100% (109/109), 56.56 Mio | 4.73 Mio/s, fait.
Résolution des deltas: 100% (16/16), fait.
jose@debian:~$ ls
A2SR    Documents  Modèles  Public           Vidéos
Bureau  Images     Musique  Téléchargements
jose@debian:~$ cd A2SR
jose@debian:~/A2SR$ ls
Course_1  course_2  course_3
jose@debian:~/A2SR$ cd course_2
jose@debian:~/A2SR/course_2$ ls
Cours_2.pdf  mission2
jose@debian:~/A2SR/course_2$ cd mission2/
jose@debian:~/A2SR/course_2/mission2$ ls
pod.yaml
jose@debian:~/A2SR/course_2/mission2$ kubectl apply -f pod.yaml
pod/myapp created
jose@debian:~/A2SR/course_2/mission2$
```



Déploiements

	Nom	Images	Étiquettes	Pods	Date de création
●	app	choco85470/myapp:v1 mysql:8.0	app.kubernetes.io/name: load-balancer	1 / 1	an hour ago

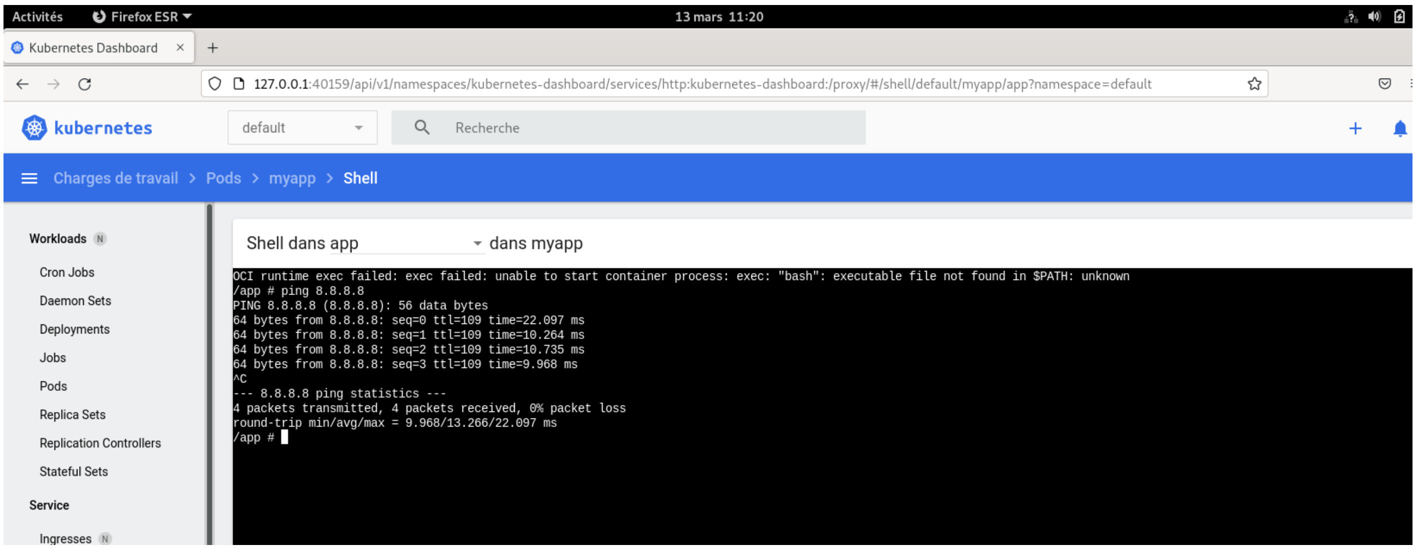
Pods

	Nom	Images	Étiquettes	Noeud	Statut	Redémarr	Utilisation CPU (coeurs)	Utilisation mémoire (octets)	Date de création
●	myapp	choco85470/myapp:v1 mysql:5.7	-	minikube	Running	0	-	-	12 minutes ago

I also did several test, that's why we can see in the screenshot there is 2 pods running.

I tried to check the logs of the pod, but I didn't really understand where I can have the ping return. I found that I can run the pod in a shell and tried a ping command. Therefore, I'm not sure that was the result expected by the teacher.

After that, we had to complete the DockerFile :



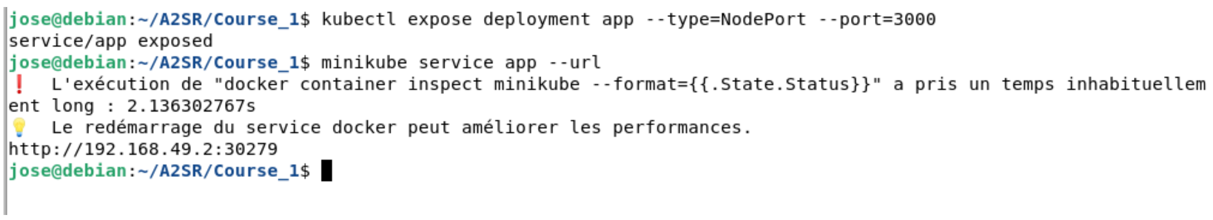
```
OCI runtime exec failed: exec failed: unable to start container process: exec: "bash": executable file not found in $PATH: unknown
/app # ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=109 time=22.097 ms
64 bytes from 8.8.8.8: seq=1 ttl=109 time=10.264 ms
64 bytes from 8.8.8.8: seq=2 ttl=109 time=10.735 ms
64 bytes from 8.8.8.8: seq=3 ttl=109 time=9.968 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 9.968/13.266/22.097 ms
/app #
```

Finally, I deleted the pod, using the dashboard interface, which is very convenient.

Mission 3 : Deploy your multi-container application

I created a deployment yml file to create a pod with the app TODO List running on it.

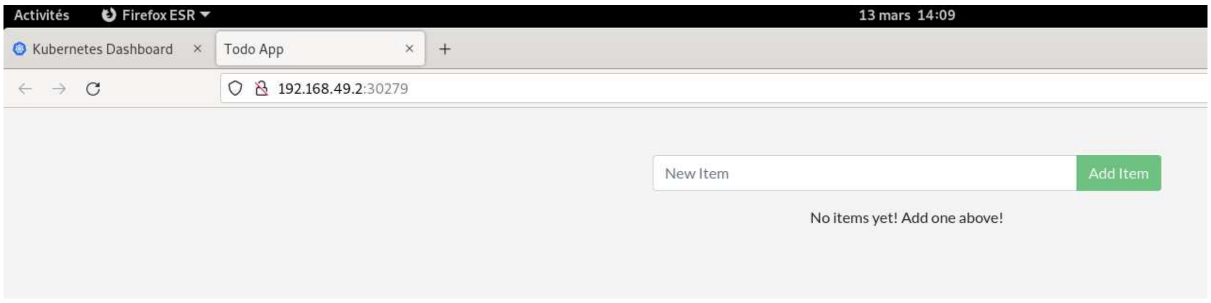
Then I exposed the app, that way i twill be accessible.



```
jose@debian:~/A2SR/Course_1$ kubectl expose deployment app --type=NodePort --port=3000
service/app exposed
jose@debian:~/A2SR/Course_1$ minikube service app --url
❗  L'exécution de "docker container inspect minikube --format={{.State.Status}}" a pris un temps inhabituellem
ent long : 2.136302767s
💡  Le redémarrage du service docker peut améliorer les performances.
http://192.168.49.2:30279
jose@debian:~/A2SR/Course_1$
```

Now I have my app listening on port 3000, and with the other command, I ask the url and the binded port to access the app. As you can see, it takes time because my computer is slow, as I run a Debian VM in virtualbox.

I checked the url, it worked !



No items yet! Add one above!

I pushed my deployement-app.yml on my github : https://github.com/josemasse/A2SR

Conclusion

To summarize, in this course we covered the use of Kubernetes. We were supposed to do it on Windows, but due to my work computer for which I don't have all the permissions, I couldn't. Instead, I used Minikube on Linux which I found relatively easy to use. We deployed pods and tested the first commands on this technology.