



Atelier 12 : Conteneurisation avec Docker et Orchestration avec Kubernetes

Module : Virtualisation

Master M1 : Sécurité IT et Big Data

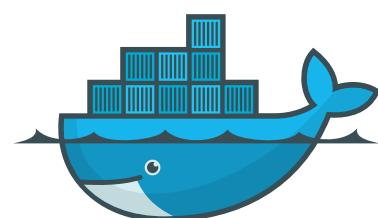


Préparé par :

Bouassab Chaimae

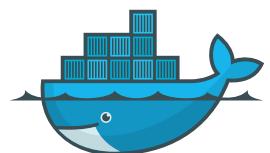
Encadré par :

Pr Mohamed BEN AHMED



Conteneurisation avec Docker et Orchestration avec Kubernetes

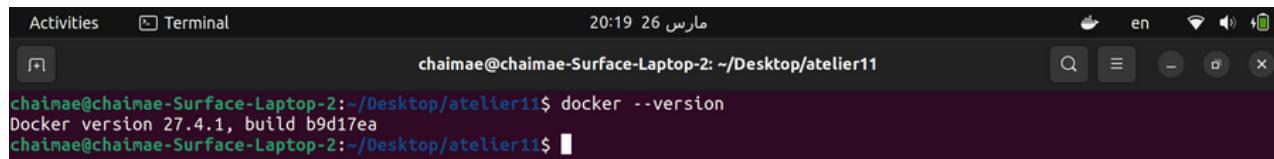
L'atelier sur la conteneurisation avec Docker et l'orchestration avec Kubernetes m'a permis de comprendre les concepts fondamentaux et les outils nécessaires pour déployer des applications de manière efficace dans un environnement virtuel. Docker m'a offert une plateforme permettant de packager des applications et leurs dépendances dans des conteneurs légers, tandis que Kubernetes m'a fourni une solution d'orchestration puissante pour gérer et déployer ces conteneurs à grande échelle. En suivant cet atelier, j'ai appris à créer des images Docker, lancer des conteneurs, et orchestrer des déploiements dans un cluster Kubernetes, ce qui m'a permis de renforcer mes compétences en gestion d'applications dans des environnements cloud ou distribués.



Partie 1 : Conteneurisation avec Docker

1.1 Vérification de l'installation de Docker

Avant de commencer, il est nécessaire de vérifier si Docker est installé sur ta machine.



```
Activities Terminal 20:19 26 مارس chaimae@chaimae-Surface-Laptop-2: ~/Desktop/atelier11$ docker --version
Docker version 27.4.1, build b9d17ea
chaimae@chaimae-Surface-Laptop-2: ~/Desktop/atelier11$
```

Création du fichier app.js :

Ce fichier contient le code de l'application Node.js que nous allons conteneuriser.

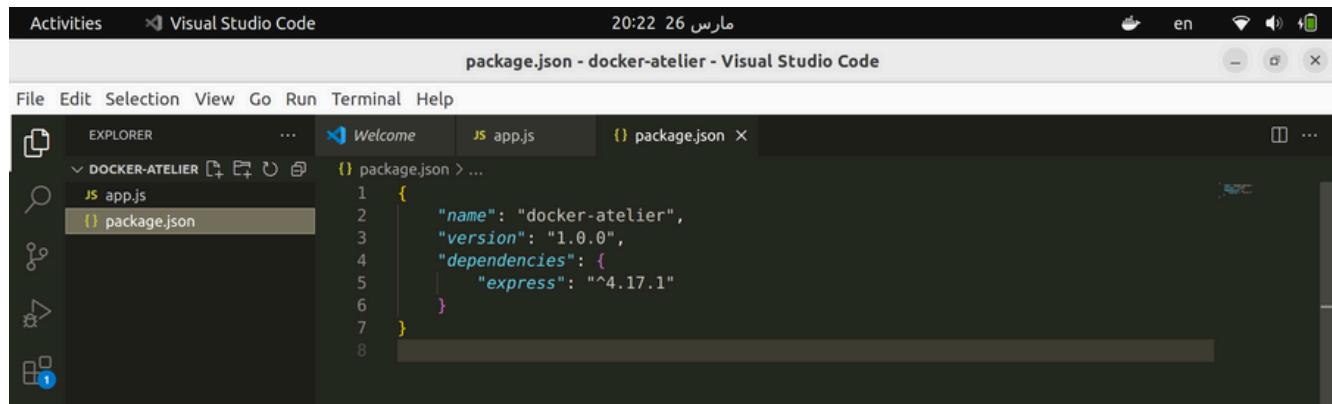
Voici un serveur Express simple :



```
Activities Visual Studio Code 20:21 26 مارس app.js - docker-atelier - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER ... Welcome JS app.js ...
JS app.js > ...
1 const express = require('express');
2 const app = express();
3 app.get('/', (req, res) => {
4   res.send('Bonjour depuis Docker !');
5 });
6 app.listen(3000, () => {
7   console.log('Serveur démarré sur le port 3000');
8 });
```

Création du fichier package.json :

Ce fichier gère les dépendances de l'application Node.js.



```
Activities Visual Studio Code 20:22 26 مارس package.json - docker-atelier - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER ... Welcome JS app.js {} package.json ...
{} package.json > ...
1 {
2   "name": "docker-atelier",
3   "version": "1.0.0",
4   "dependencies": {
5     "express": "^4.17.1"
6   }
7 }
```

```

FROM node:16
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
EXPOSE 3000
CMD ["node", "app.js"]

```

Construction de l'image Docker :

```

chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$ code .
chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$ docker build -t mon-app-node .
[+] Building 22.1s (5/10) docker:desktop-linux
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 144B 0.0s
=> [internal] load metadata for docker.io/library/node:16 3.6s
=> [auth] library/node:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerrcignore 0.1s
=> => transferring context: 2B 0.0s
=> [1/5] FROM docker.io/library/node:16@sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fe8bbfb8c6e4246 18.3s
=> => resolve docker.io/library/node:16@sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fe8bbfb8c6e4246c 0.1s
=> => sha256:ee7d78be1eb92caf6ae84fc3af736b23eca018d5dedc967ae5bdee6d7082403b 450B / 450B 0.4s
=> => sha256:ca266fd6192108b67fb57b74753a8c4ca5d8bd458baae3d4df7ce9f42dedcc1d 2.27MB / 2.27MB 2.1s
=> => sha256:0e421f66aff42bb069dff26af6d132194b22a1082b08c5ef7cd69c627783c04 10.49MB / 34.79MB 17.6s
=> => sha256:ae3b95bbaa61ce24cefdd89e7c74d6fdb7713b2bcae93af47063d06bd7e02172 4.20kB / 4.20kB 1.5s
=> => sha256:513d779256048c961239af5f500589330546b072775217272e19ffae1635e98e 3.15MB / 191.90MB 17.3s
=> => sha256:ffd9397e94b74abcb54e514f1430e00f604328d1f895eadbd482f08cc02444e5 4.19MB / 51.89MB 16.1s
=> => sha256:7e9bf114588c05b2df612b083b96582f3b8dbf51647aa6138a50d09d42df2454 12.58MB / 17.58MB 15.6s
=> [internal] load build context 0.2s
=> => transferring context: 554B 0.0s

```

Vérification de la création de l'image : Commande : docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mon-app-node	latest	49853ab226a2	About a minute ago	1.35GB
docker-demo-app	1.0	2fca14b1cff4	57 minutes ago	1.36GB
node-app_node-app	latest	859f5820ff85	7 days ago	201MB
mysql	latest	9b9d0aab4860	2 months ago	1.09GB
gcr.io/k8s-minikube/kicbase	v0.0.46	cef9f3c2e399	2 months ago	1.86GB
gcr.io/k8s-minikube/kicbase	<none>	fd2d445ddcc3	2 months ago	1.86GB

1.3 Lancement du conteneur Docker

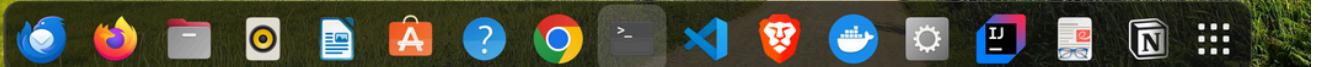
Lancement du conteneur

cette commande lance un conteneur basé sur l'image créée, en exposant le port 3000 de la machine hôte vers le port 3000 du conteneur

Vérification des conteneurs en cours d'exécution :

docker ps

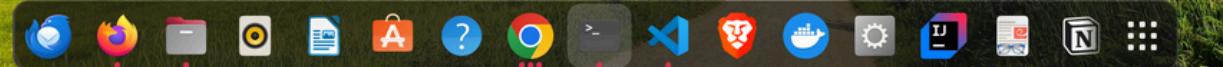
```
vae5d76483620cc360c16e5e40598ec0e1ce2c34ca1237ca0230c): bind for 0.0.0.0:3000 failed: port is already allocated.  
chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$ docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS  
NAMES  
bc3aef011b84 docker-demo-app:1.0 "docker-entrypoint.s..." 57 minutes ago Up 57 minutes 0.0.0.0:3000->3000/tcp  
admiring_shaw  
7039db31179b mysql:latest "docker-entrypoint.s..." 7 days ago Up About an hour 3306/tcp, 0.0.0.0:3307->  
3306/tcp node-app_mysql-container_1  
chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$
```



1.4 Gestion des conteneurs

Lister tous les conteneurs (même arrêtés)

```
3306/tcp node-app_mysql-container_1  
chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$ sudo lsof -i :3000  
[sudo] password for chainae:  
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME  
com.docke 7724 chainae 67u IPv6 503523 0t0 TCP *:3000 (LISTEN)  
chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$ docker ps -a  
CONTAINER ID IMAGE COMMAND CREATED STATUS  
PORTS NAMES  
9d04f1ed9c05 mon-app-node "docker-entrypoint.s..." About a minute ago Created  
a0d05a30f28c mon-app-node intelligent_tu "docker-entrypoint.s..." 2 minutes ago Created  
4bc541242bea gcr.io/k8s-minikube/kicbase:v0.0.46 "/usr/local/bin/entr..." 39 minutes ago Exited (130) 25 minutes  
ago minikube  
bc3aef011b84 docker-demo-app:1.0 "docker-entrypoint.s..." 58 minutes ago Up 58 minutes  
0.0.0.0:3000->3000/tcp admiring_shaw  
d3510d8d0519 node-app_node-app "docker-entrypoint.s..." 7 days ago Exited (255) About an ho  
ur ago 0.0.0.0:3000->3000/tcp node-app_node-app_1  
7039db31179b mysql:latest "docker-entrypoint.s..." 7 days ago Up About an hour  
3306/tcp, 0.0.0.0:3307->3306/tcp node-app_mysql-container_1  
chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$
```



1 Installation et vérification de Minikube

Démarrage de Minikube :

```
See 'docker run --help'.
chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$ minikube start
 INFO minikube v1.35.0 on Ubuntu 22.04
+ Using the docker driver based on existing profile

✖ Exiting due to PROVIDER_DOCKER_NOT_RUNNING: "docker version --format <no value>-<no value>:<no value>" exit status 1: C
annot connect to the Docker daemon at unix:///home/chainae/.docker/desktop/docker.sock. Is the docker daemon running?
💡 Suggestion: Start the Docker service
💡 Documentation: https://minikube.sigs.k8s.io/docs/drivers/docker/

chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$ kubectl get nodes
E0326 20:33:30.873418 80702 memcache.go:265] "Unhandled Error" err=
```

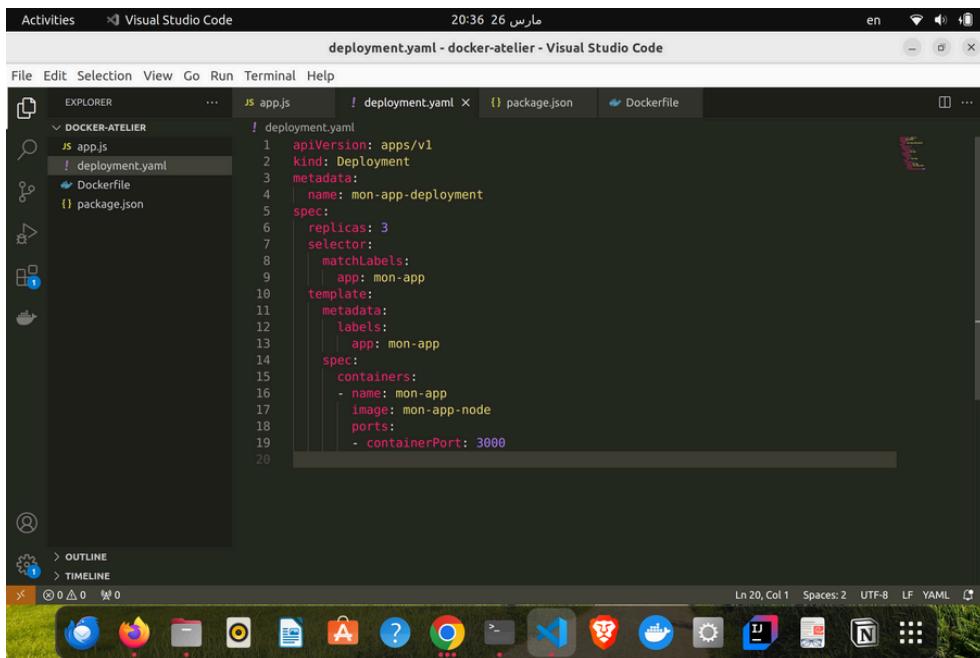
Vérification du cluster Kubernetes :

```
chainae@chainae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$ kubectl get nodes
E0326 20:33:30.873418 80702 memcache.go:265] "Unhandled Error" err=<
    couldn't get current server API group list: <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/c9828ae0/script/redirect.js'></script></head><body style='background-color:white; color:white;'>
        Authentication required
    <!--
    -->
</body></html>
>
E0326 20:33:30.886097 80702 memcache.go:265] "Unhandled Error" err=
```

2.2 Déploiement de l'application sur Kubernetes

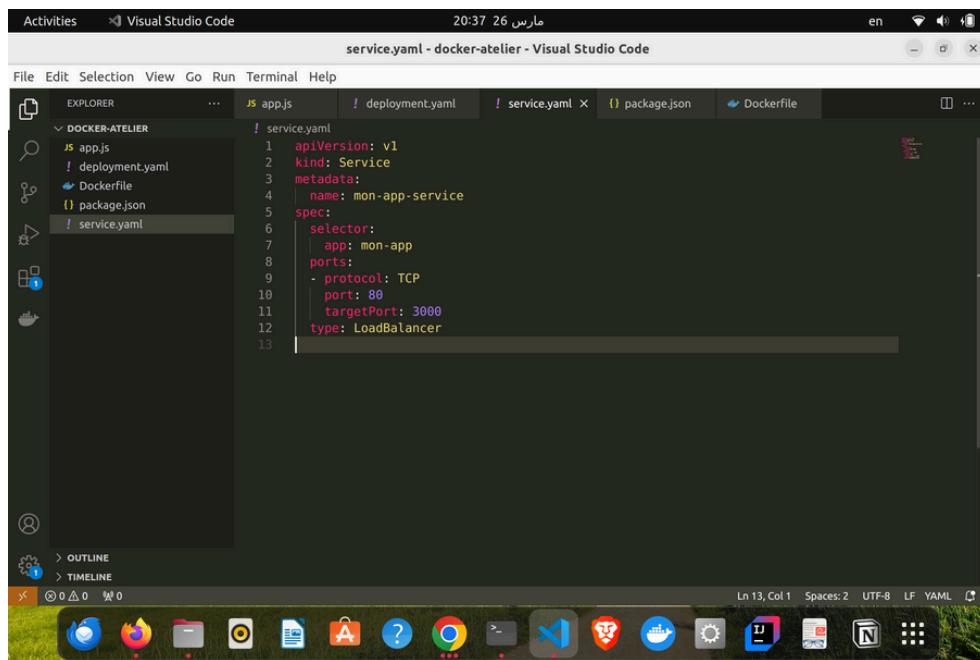
Création du fichier deployment.yaml :

Ce fichier YAML contient la configuration du déploiement de l'application dans Kubernetes.



Création du fichier service.yaml :

Ce fichier YAML expose l'application à l'extérieur du cluster.



```
apiVersion: v1
kind: Service
metadata:
  name: mon-app-service
spec:
  selector:
    app: mon-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
  type: LoadBalancer
```

Sauvegarde et chargement de l'image Docker dans Minikube :

```
docker save mon-app-node | minikube ssh -- docker load
```

Application des configurations Kubernetes :

```
kubectl apply -f deployment.yaml
```

```
kubectl apply -f service.yaml
```

Vérification des Pods :

```
kubectl get pods
```

Accès à l'application :

```
minikube service mon-app-service --url
```

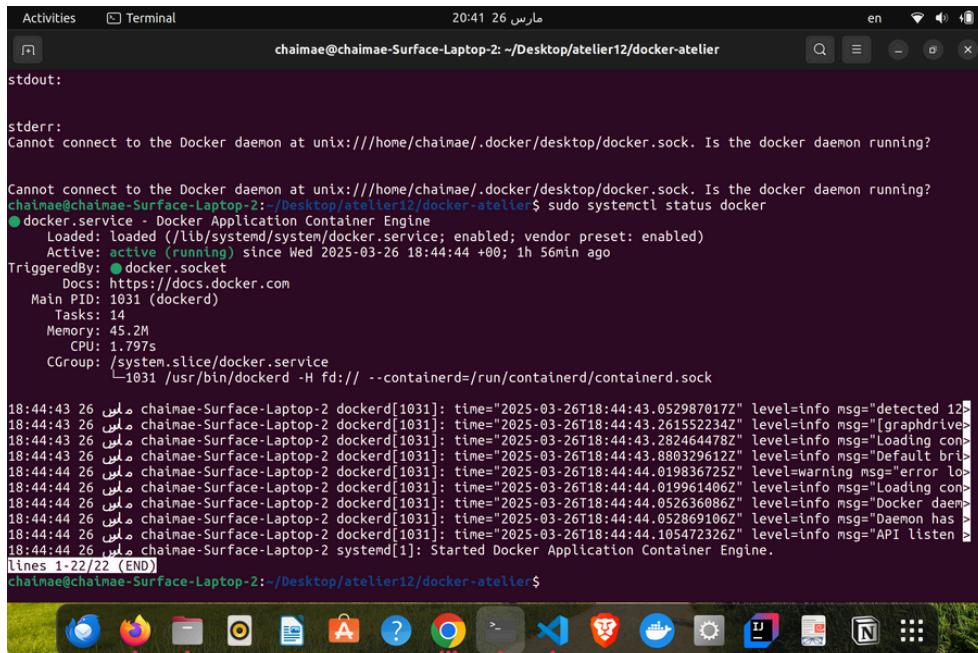
2.3 Mise à l'échelle du déploiement

```
kubectl scale deployment mon-app-deployment --replicas=5
```

Suppression des ressources Kubernetes :

kubectl delete -f deployment.yaml

kubectl delete -f service.yaml



```
Activities Terminal 20:41 26 مارس chaimae@chaimae-Surface-Laptop-2: ~/Desktop/atelier12/docker-atelier
chaimae@chaimae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2025-03-26 18:44:44 +00; 1h 56min ago
TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 1031 (dockerd)
     Tasks: 14
    Memory: 45.2M
      CPU: 1.797s
     CGroup: /system.slice/docker.service
             └─1031 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

18:44:43 26 مارس chaimae-Surface-Laptop-2 dockerd[1031]: time="2025-03-26T18:44:43.052987017Z" level=info msg="detected 128 CPUs"
18:44:43 26 مارس chaimae-Surface-Laptop-2 dockerd[1031]: time="2025-03-26T18:44:43.261552234Z" level=info msg="[graphdriver] using native storage driver"
18:44:43 26 مارس chaimae-Surface-Laptop-2 dockerd[1031]: time="2025-03-26T18:44:43.282464478Z" level=info msg="Loading configuration from /etc/docker/config.json"
18:44:43 26 مارس chaimae-Surface-Laptop-2 dockerd[1031]: time="2025-03-26T18:44:43.880379612Z" level=info msg="Default bridge driver"
18:44:44 26 مارس chaimae-Surface-Laptop-2 dockerd[1031]: time="2025-03-26T18:44:44.019836725Z" level=warning msg="error loading config file: /etc/docker/config.json: failed to open file: No such file or directory"
18:44:44 26 مارس chaimae-Surface-Laptop-2 dockerd[1031]: time="2025-03-26T18:44:44.019961406Z" level=info msg="Loading configuration from /etc/docker/daemon.json"
18:44:44 26 مارس chaimae-Surface-Laptop-2 dockerd[1031]: time="2025-03-26T18:44:44.052636086Z" level=info msg="Docker daemon initialized"
18:44:44 26 مارس chaimae-Surface-Laptop-2 dockerd[1031]: time="2025-03-26T18:44:44.052869106Z" level=info msg="Daemon has started successfully"
18:44:44 26 مارس chaimae-Surface-Laptop-2 dockerd[1031]: time="2025-03-26T18:44:44.105472326Z" level=info msg="API listen on 0.0.0.0:2375"
18:44:44 26 مارس chaimae-Surface-Laptop-2 systemd[1]: Started Docker Application Container Engine.
lines 1-22/22 (END)
chaimae@chaimae-Surface-Laptop-2:~/Desktop/atelier12/docker-atelier$
```

Cet atelier m'a permis d'acquérir des compétences pratiques dans la conteneurisation et l'orchestration d'applications. En utilisant Docker, j'ai pu créer et gérer des images et des conteneurs, facilitant ainsi le déploiement et la gestion des applications. Grâce à Kubernetes, j'ai appris à déployer et à mettre à l'échelle une application conteneurisée dans un environnement clusterisé. L'exercice de gestion de Pods et de Services m'a également permis d'expérimenter avec l'auto-scaling et le monitoring des applications. Ces connaissances me permettront de mieux gérer les applications à grande échelle, avec une gestion simplifiée et un contrôle total sur le déploiement et la scalabilité dans des environnements complexes. Ce travail m'a montré l'importance de l'automatisation et de l'orchestration dans les processus de développement et de déploiement d'applications modernes.