



UNIVERSITÉ MOHAMMED V DE RABAT

FACULTÉ DES SCIENCES

**MASTER TRAITEMENT INTELLIGENT DES
SYSTEMES**

Compte rendu Projet 2:

**MISE EN PLACE D'UN CLOUD PaaS
PRIVE AVEC OPENSIFT**

Réalisé par :

Lamia Oualili

CNE : 130014913

Encadré par :

Mr. Ahmed EL-YAHYAOUI

Année universitaire :

2022-2023

Table des matières :

| | |
|---|----|
| Déploiement et exploitation d'une solution Cloud IaaS privée avec OpenStack.. | 1 |
| Table des figures : | 3 |
| I. Objectif de ce TP : | 4 |
| II. Installation du cloud openshift : | 4 |
| 1. Mise à jour du système : | 4 |
| 2. Installer Docker CE : | 4 |
| 3. Télécharger OpenShift Origin : | 5 |
| 4. Démarrer le cluster OpenShift Origin : | 6 |
| 5. Créer un projet sur OpenShift..... | 7 |
| 6. Accédez à la console Web OpenShift..... | 8 |
| III. Déploiement d'applications sur Openshift : | 9 |
| 1. Déployer une application sur OpenShift Origin : | 9 |
| 2. Vérifier l'accès externe : | 13 |
| IV. Conclusion : | 14 |
| □ Lien de la vidéo explicative | 14 |

Table des figures :

| | |
|--|----|
| Figure 1: Mise à jour | 4 |
| Figure 2: Démarrer Docker | 4 |
| Figure 3: Télécharger openshift | 5 |
| Figure 4: Création du fichier et redémarrer le service | 6 |
| Figure 5: Démarrage du cluster..... | 6 |
| Figure 6: Connecter au cluster | 7 |
| Figure 7: Vérification de l'état du projet | 7 |
| Figure 8: Connecter à Openshift | 8 |
| Figure 9: Login d'openshift | 8 |
| Figure 10: Création du projet | 9 |
| Figure 11: Se connecter avec developper..... | 9 |
| Figure 12: Basculer vers le projet | 10 |
| Figure 13: vérifié l'état | 10 |
| Figure 14: Image d'application..... | 10 |
| Figure 15: Déploiement d'application | 11 |
| Figure 16: Informations sur le service..... | 11 |
| Figure 17: Informations détaillées..... | 12 |
| Figure 18: Vérification des pods | 12 |
| Figure 19: Exposer l'application..... | 13 |
| Figure 20: Des informations sur les itinéraires | 13 |
| Figure 21: Application prête | 14 |

TP 1 : MISE EN PLACE D'UN CLOUD PaaS PRIVE AVEC OPENSIFT:

I. Objectif de ce TP :

L'objectif de ce TP est :

- Mettre en place une solution de Cloud privé PaaS open source avec OpenShift.
- OpenShift est une plateforme en tant que service PaaS, permet aux développeurs de développer et déployer leurs applications sur une infrastructure cloud.

II. Installation du cloud opensift :

1. Mise à jour du système :

- Connecter à l'Ubuntu 20.4
- Redémarrer le système par Reboot.

```
root@ubuntu:~# apt update -y && apt upgrade -y
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
```

Figure 1: Mise à jour

2. Installer Docker CE :

- Après avoir installé Docker dans le système pour exécuter les services OKD dans les conteneurs Docker, il faut démarrer Docker et l'activer puis vérifier son état.

```
lamia@ubuntu: ~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit/
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ ls
kubectl LICENSE oc openshift.local.clusterup README.md
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo systemctl status docker
[sudo] password for lamia:
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-10-30 09:04:09 PDT; 1min 24s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 1177 (dockerd)
      Tasks: 14
     Memory: 106.9M
    CGroup: /system.slice/docker.service
            └─1177 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Figure 2: Démarrer Docker

3. Télécharger OpenShift Origin :

- On a téléchargé la version v3.11.0 à partir du git hub et puis extraire le fichier téléchargé.
- Ensuite, on a changé le répertoire vers le répertoire extrait et copié les binaires kubectl et oc dans le répertoire /usr/local/bin

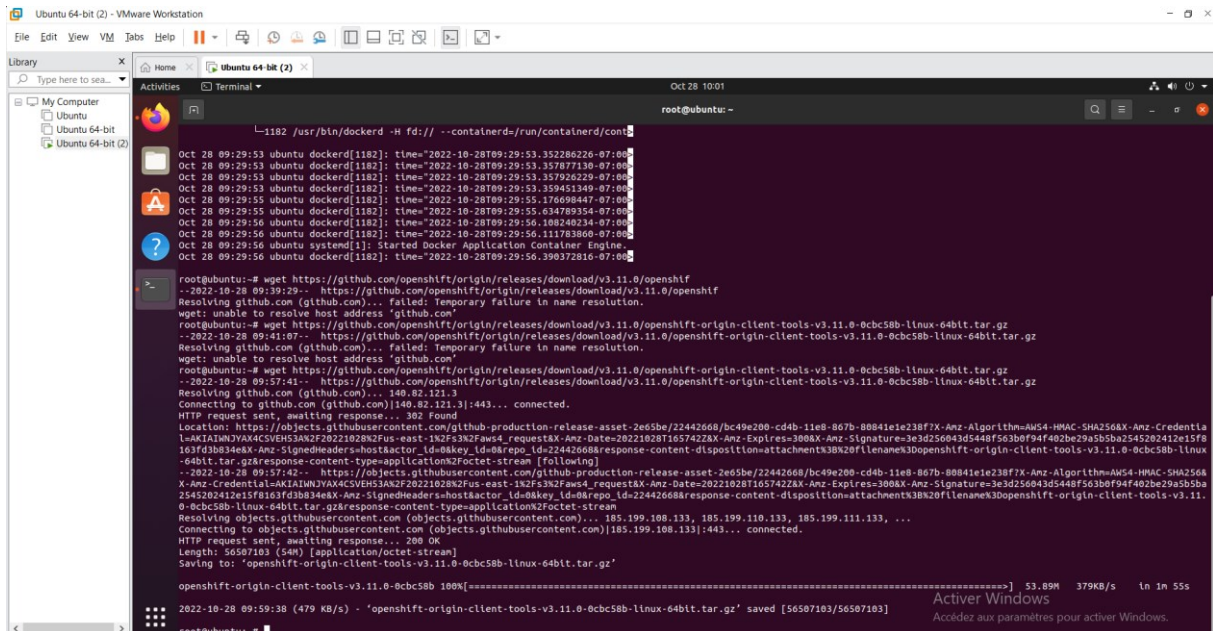


Figure 3: Télécharger openshift

- Puis on a vérifié l'installation de l'utilitaire client OpenShift.
- Ensuite on a créé un nouveau fichier daemon.json et autorisé l'utilisation du registre Insecure Docker.
- On a enregistré et fermé le fichier, puis redémarrer le service Docker pour implémenter les modifications

```

root@ubuntu:~# wget https://github.com/openshift/origin/releases/download/v3.11.0/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit.tar.gz
--2022-10-28 09:57:41-- https://github.com/openshift/origin/releases/download/v3.11.0/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit.tar.gz
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/22442668/bc49e200-cd4b-11e8-b163fd3b834e&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=22442668&response-content-disposition=attachment%3Bfilename%3Dopenshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit.tar.gz&response-content-type=application%2Foctet-stream [following]
--2022-10-28 09:57:42-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/22442668/bc49e200-cd4b-11e8-b163fd3b834e&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=22442668&response-content-disposition=attachment%3Bfilename%3Dopenshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.110.133, 185.199.112.133, 185.199.114.133
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 56507103 (54M) [application/octet-stream]
Saving to: 'openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit.tar.gz'

openshift-origin-client-tools-v3.11.0-0cbc58b 100%[=====]
2022-10-28 09:59:38 (479 KB/s) - 'openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit.tar.gz' saved [56507103/56507103]

root@ubuntu:~# tar -xvzf openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit.tar.gz
tar: Ignoring unknown extended header keyword 'LIBARCHIVE.xattr.security.selinux'
openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit/oc
tar: Ignoring unknown extended header keyword 'LIBARCHIVE.xattr.security.selinux'
openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit/kubectl
tar: Ignoring unknown extended header keyword 'LIBARCHIVE.xattr.security.selinux'
openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit/README.md
tar: Ignoring unknown extended header keyword 'LIBARCHIVE.xattr.security.selinux'
openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit/LICENSE
root@ubuntu:~# cd openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit
root@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit# cp oc kubectl /usr/local/bin/
root@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit# oc version
oc v3.11.0-0cbc58b
kubernetes v1.11.0+d4cacc0
features: Basic-Auth GSSAPI Kerberos SPNEGO
root@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit#

```

Figure 4: Création du fichier et redémarrer le service

4. Démarrer le cluster OpenShift Origin :

- On démarré le cluster OpenShift Origin en spécifiant l'adresse IP de votre système:

```

lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc cluster up --public-hostname=192.168.163.130
Getting a Docker client ...
Checking if image openshift/origin-control-plane:v3.11 is available ...
Creating shared mount directory on the remote host ...
Determining server IP ...
Using public hostname IP 192.168.163.130 as the host IP
Checking if OpenShift is already running ...
Checking for supported Docker version (>=1.22) ...
Checking if insecure registries is configured properly in Docker ...

```

Figure 5: Démarrage du cluster

- Puis se connecter au cluster en tant qu'utilisateur administrateur.

```
lamia@ubuntu: ~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit
I1030 09:09:10.009494 2352 apply_list.go:67] Installing "sample-templates/sample pipeline"
I1030 09:09:10.009584 2352 apply_list.go:67] Installing "sample-templates/dancer quickstart"
I1030 09:09:10.033847 2352 apply_list.go:67] Installing "centos-imagestreams"
I1030 09:09:10.040573 2352 apply_template.go:81] Installing "openshift-web-console-operator"
I1030 09:09:38.192697 2352 interface.go:41] Finished installing "sample-templates/rails quickstart" "sample-templates/jenkins pipeline-templates/mariadb" "sample-templates/mysql" "sample-templates/postgresql" "sample-templates/cakephp quickstart" "sample-templates/django quickstart" "sample-templates/dancer quickstart" "sample-templates/nodejs quickstart"
I1030 09:10:36.387478 2352 interface.go:41] Finished installing "sample-templates" "openshift-image-registry" "persistent-volumes" "centos-imagestreams" "openshift-router"
Login to server ...
Creating initial project "myproject" ...
Server Information ...
OpenShift server started.

The server is accessible via web console at:
https://192.168.163.130:8443

You are logged in as:
User: developer
Password: <any value>

To login as administrator:
oc login -u system:admin

lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc login -u system:admin
Logged into "https://192.168.163.130:8443" as "system:admin" using existing credentials.

You have access to the following projects and can switch between them with 'oc project <projectname>':

    default
    kube-dns
    kube-proxy
    kube-public
    kube-system
    * myproject
```

Figure 6: Connecter au cluster

- On a passé au projet par défaut et vérifié l'état actuel de votre projet.

```
openshift-web-console

Using project "myproject".
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc project default
Now using project "default" on server "https://192.168.163.130:8443".
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc status
In project default on server https://192.168.163.130:8443

svc/docker-registry - 172.30.1.1:5000
  dc/docker-registry deploys docker.io/openshift/origin-docker-registry:v3.11
  deployment #1 deployed 3 minutes ago - 1 pod

svc/kubernetes - 172.30.0.1:443 -> 8443

svc/router - 172.30.150.214 ports 80, 443, 1936
  dc/router deploys docker.io/openshift/origin-haproxy-router:v3.11
  deployment #1 deployed 3 minutes ago - 1 pod

View details with 'oc describe <resource>/<name>' or list everything with 'oc get all'.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc login
Authentication required for https://192.168.163.130:8443 (openshift)
Username: developer
Password:
```

Figure 7: Vérification de l'état du projet

5. Créer un projet sur OpenShift

- On a connecté à OpenShift avec l'utilisateur developer.
- Ensuite fournir le nom d'utilisateur en tant que developer et le mot de passe en tant que developer.
- Puis on a créé un nouveau projet.

```

View details with 'oc describe <resource>/<name>' or list everything with 'oc get all'.
lanla@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc login
Authentication required for https://192.168.163.130:8443 (openshift)
Username: developer
Password:
Login successful.

You have one project on this server: "myproject"

Using project "myproject".
lanla@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc new-project dev --display-name="Project - Dev" --description="My Projec"
Now using project "dev" on server "https://192.168.163.130:8443".

You can add applications to this project with the 'new-app' command. For example, try:

  oc new-app centos/ruby-25-centos7-https://github.com/sclorg/ruby-ex.git

to build a new example application in Ruby.
lanla@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc cluster down

```

Figure 8: Connecter à Openshift

6. Accédez à la console Web OpenShift

- Maintenant, on a ouvert le navigateur Web et tapé l'URL depuis la requête précédente.
- Puis on a fourni le nom d'utilisateur et le mot de passe de developer.

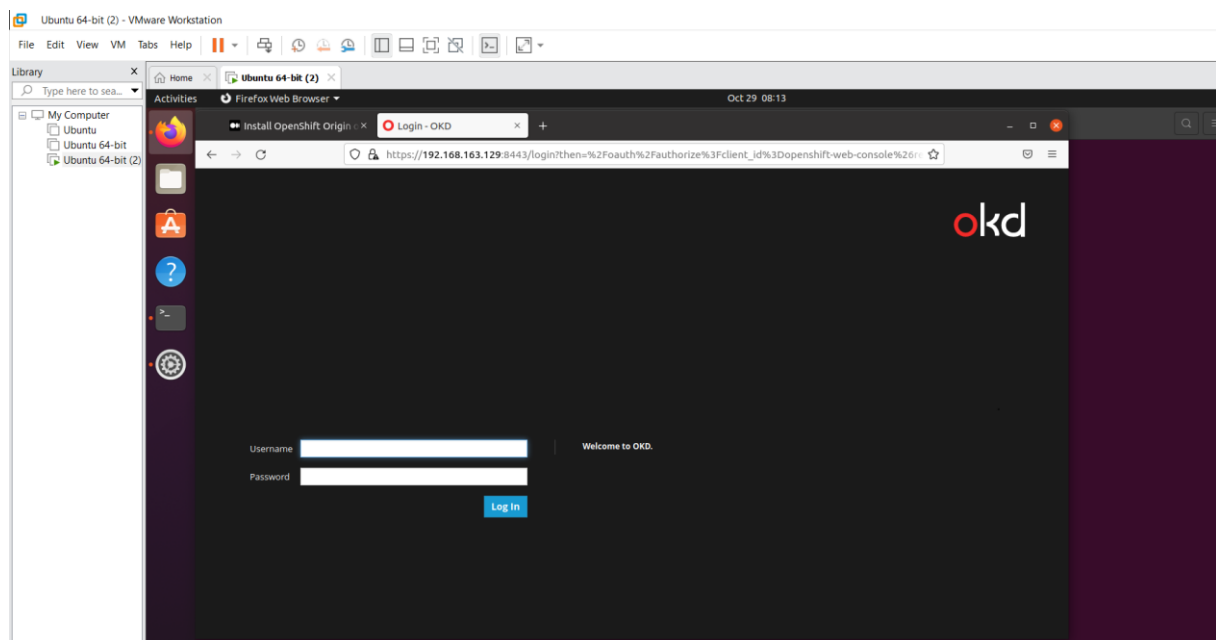


Figure 9: Login d'openshift

- Sur cette interface on remplit le formulaire suivant pour créer un projet.

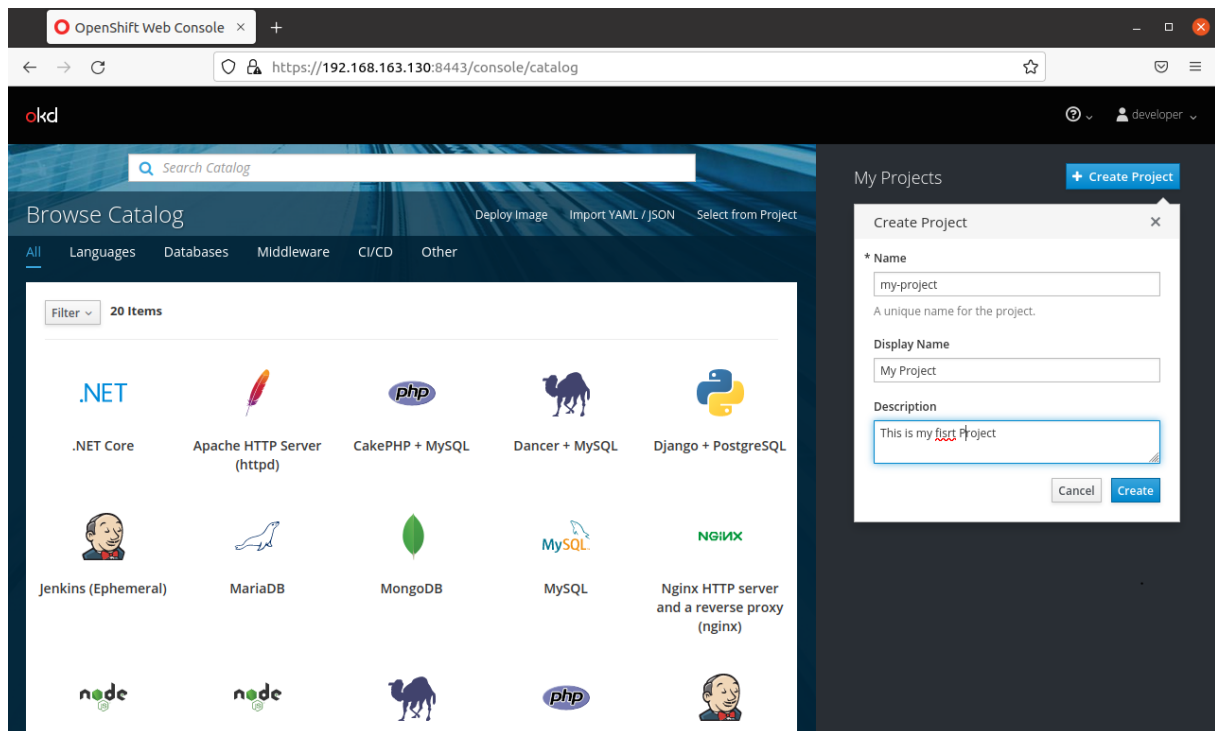


Figure 10: Création du projet

III. Déploiement d'applications sur OpenShift :

1. Déployer une application sur OpenShift Origin :

- Premièrement on a connecté à OpenShift avec l'utilisateur developer.

```

lamia@ubuntu: ~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit
start" "sample-templates/dancer quickstart" "sample-templates/nodejs quickstart" "sample-templates/rails quickstart" "sample-templates/mongodb" "sample-templates/mariadb" "sample-templates/mysql"
I1030 09:25:27.110551 18100 interface.go:41] Finished installing "sample-templates" "openshift-image-registry" "persistent-volumes" "openshift-web-console-operator" "centos-imagestreams" "openshift-router"
Server Information ...
OpenShift server started.

The server is accessible via web console at:
https://192.168.163.130:8443

lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc login
Authentication required for https://192.168.163.130:8443 (openshift)
Username: developer
Password:
Login successful.

You have access to the following projects and can switch between them with 'oc project <projectname>':

* dev
  my-project
  myproject

Using project "dev".
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$

```

Figure 11: Se connecter avec developer

- Puis on a basculé le projet vers le my-project que nous avons créé.

```
You have access to the following projects and can switch between them with 'oc project <projectname>':
* dev
  my-project
  myproject

Using project "dev".
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc project my-project
Now using project "my-project" on server "https://192.168.163.130:8443".
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$
```

Figure 12: Basculer vers le projet

- Ensuite la vérification de l'état du projet actuel.

```
lamia@ubuntu: ~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit
The server is accessible via web console at:
https://192.168.163.130:8443

lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc login
Authentication required for https://192.168.163.130:8443 (openshift)
Username: developer
Password:
Login successful.

You have access to the following projects and can switch between them with 'oc project <projectname>':
* dev
  my-project
  myproject

Using project "dev".
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc project my-project
Now using project "my-project" on server "https://192.168.163.130:8443".
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc status
In project My Project (my-project) on server https://192.168.163.130:8443

You have no services, deployment configs, or build configs.
Run 'oc new-app' to create an application.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$
```

Figure 13: vérifié l'état

- Ici on a marqué une image d'application du registre Docker Hub :

```
You have no services, deployment configs, or build configs.
Run 'oc new-app' to create an application.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc tag --source=docker openshift/deployment-
example:v2 deployment-example:
Error from server (BadRequest): ImageStreamTags must be retrieved with <name>:<tag>
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc tag --source=docker openshift/deployment-
example:v2 deployment-example:latest
Tag deployment-example:latest set to openshift/deployment-example:v2.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$
```

Figure 14: Image d'application

- Puis on a déployé une application sur Openshift :

```
lamia@ubuntu: ~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit
example:v2 deployment-example:
Error from server (BadRequest): ImageStreamTags must be retrieved with <name>:<tag>
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc tag --source=docker openshift/deployment-
example:v2 deployment-example:latest
Tag deployment-example:latest set to openshift/deployment-example:v2.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc new-app deployment-example
--> Found image da61bb2 (7 years old) in image stream "my-project/deployment-example" under tag "latest" for "deployment-ex
ample"

* This image will be deployed in deployment config "deployment-example"
* Port 8080/tcp will be load balanced by service "deployment-example"
* Other containers can access this service through the hostname "deployment-example"
* WARNING: Image "my-project/deployment-example:latest" runs as the 'root' user which may not be permitted by your clus
ter administrator

--> Creating resources ...
deploymentconfig.apps.openshift.io "deployment-example" created
service "deployment-example" created
--> Success
Application is not exposed. You can expose services to the outside world by executing one or more of the commands below
:
'oc expose svc/deployment-example'
Run 'oc status' to view your app.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$
```

Figure 15:Déploiement d'application

- Maintenant on a obtenu des informations sur le service.

```
lamia@ubuntu: ~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit
* WARNING: Image "my-project/deployment-example:latest" runs as the 'root' user which may not be permitted by your clus
ter administrator

--> Creating resources ...
deploymentconfig.apps.openshift.io "deployment-example" created
service "deployment-example" created
--> Success
Application is not exposed. You can expose services to the outside world by executing one or more of the commands below
:
'oc expose svc/deployment-example'
Run 'oc status' to view your app.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc status
In project My Project (my-project) on server https://192.168.163.130:8443

svc/deployment-example - 172.30.137.12:8080
dc/deployment-example deploys istag/deployment-example:latest
deployment #1 deployed 7 minutes ago - 1 pod

2 infos identified, use 'oc status --suggest' to see details.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc get svc
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
deployment-example  ClusterIP   172.30.137.12 <none>         8080/TCP   8m
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$
```

Figure 16:Informations sur le service

- Pour obtenir des informations détaillées sur service.

```
lamia@ubuntu: ~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit
svc/deployment-example - 172.30.137.12:8080
dc/deployment-example deploys istag/deployment-example:latest
deployment #1 deployed 7 minutes ago - 1 pod

2 infos identified, use 'oc status --suggest' to see details.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc get svc
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
deployment-example  ClusterIP   172.30.137.12 <none>         8080/TCP   8m
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc describe svc deployment-example
Name:                deployment-example
Namespace:            my-project
Labels:               app=deployment-example
Annotations:          openshift.io/generated-by=OpenShiftNewApp
Selector:             app=deployment-example,deploymentconfig=deployment-example
Type:                 ClusterIP
IP:                   172.30.137.12
Port:                 8080-tcp 8080/TCP
TargetPort:           8080/TCP
Endpoints:            172.17.0.10:8080
Session Affinity:     None
Events:               <none>
```

Figure 17: Informations détaillées

- Ici on a vérifié l'état des pods.

```
lamia@ubuntu: ~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit
deployment #1 deployed 7 minutes ago - 1 pod

2 infos identified, use 'oc status --suggest' to see details.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc get svc
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
deployment-example  ClusterIP   172.30.137.12 <none>         8080/TCP   8m
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc describe svc deployment-example
Name:                deployment-example
Namespace:            my-project
Labels:               app=deployment-example
Annotations:          openshift.io/generated-by=OpenShiftNewApp
Selector:             app=deployment-example,deploymentconfig=deployment-example
Type:                 ClusterIP
IP:                   172.30.137.12
Port:                 8080-tcp 8080/TCP
TargetPort:           8080/TCP
Endpoints:            172.17.0.10:8080
Session Affinity:     None
Events:               <none>
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc get pods
NAME                READY    STATUS    RESTARTS    AGE
deployment-example-1-d2d6t  1/1     Running   0           11m
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$
```

Figure 18: Vérification des pods

- Ensuite on a exposé l'application pour un accès externe

```

lamia@ubuntu: ~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit
2 infos identified, use 'oc status --suggest' to see details.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
deployment-example ClusterIP      172.30.137.12 <none>         8080/TCP    8m
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc describe svc deployment-example
Name:
Namespace:    my-project
Labels:
Annotations:  openshift.io/generated-by=OpenShiftNewApp
Selector:     app=deployment-example,deploymentconfig=deployment-example
Type:         ClusterIP
IP:           172.30.137.12
Port:         8080-tcp  8080/TCP
TargetPort:   8080/TCP
Endpoints:    172.17.0.10:8080
Session Affinity: None
Events:       <none>
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc get pods
NAME                                READY    STATUS    RESTARTS    AGE
deployment-example-1-d2d6t          1/1      Running   0            11m
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc expose service/deployment-example
route.route.openshift.io/deployment-example exposed
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$

```

Figure 19: Exposer l'application

- Maintenant on a affiché les informations sur les itinéraires.

```

2 infos identified, use 'oc status --suggest' to see details.
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
deployment-example ClusterIP      172.30.137.12 <none>         8080/TCP    8m
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc describe svc deployment-example
Name:
Namespace:    my-project
Labels:
Annotations:  openshift.io/generated-by=OpenShiftNewApp
Selector:     app=deployment-example,deploymentconfig=deployment-example
Type:         ClusterIP
IP:           172.30.137.12
Port:         8080-tcp  8080/TCP
TargetPort:   8080/TCP
Endpoints:    172.17.0.10:8080
Session Affinity: None
Events:       <none>
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc get pods
NAME                                READY    STATUS    RESTARTS    AGE
deployment-example-1-d2d6t          1/1      Running   0            11m
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc expose service/deployment-example
route.route.openshift.io/deployment-example exposed
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$ sudo oc get routes
NAME                                HOST/PORT                                PATH    SERVICES    PORT    TERMINATION
--                                --                                --    --         --    --
deployment-example                  deployment-example-my-project.192.168.163.130.nip.io  /      deployment-example  8080-tcp
lamia@ubuntu:~/openshift-origin-client-tools-v3.11.0-0cbc58b-linux-64bit$

```

Figure 20: Des informations sur les itinéraires

2. Vérifier l'accès externe :

En fin, l'application est prête pour un accès externe.

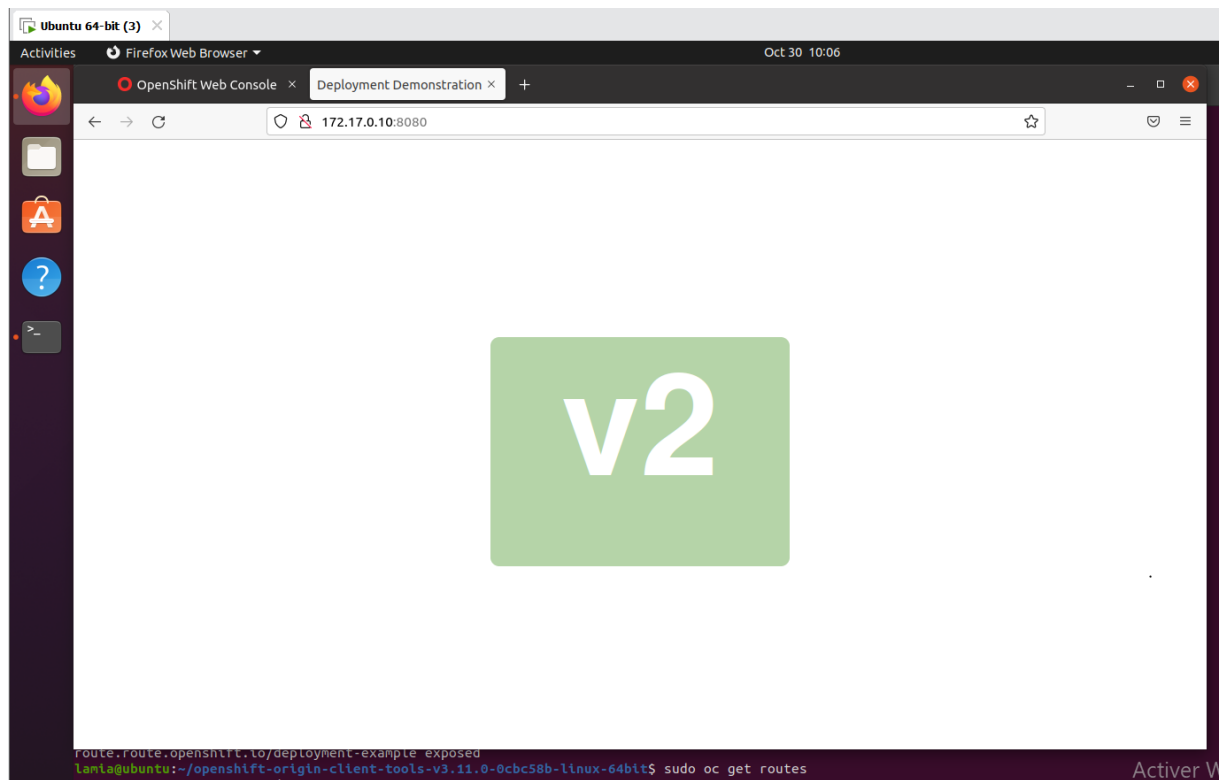


Figure 21: Application prête

IV. Conclusion :

Dans ce TP :

- Nous avons installé et configuré avec succès un nœud unique OpenShift Origin sur Ubuntu 18.04.
- Maintenant, on peut commencer à développer votre première application dans l'environnement OpenShift.

✓ Lien de la vidéo explicative :
<https://clipchamp.com/watch/4fBWVtuEwyf>