Despliegue de un Apache HTTP Server en Oracle Kubernetes Engine (OKE)

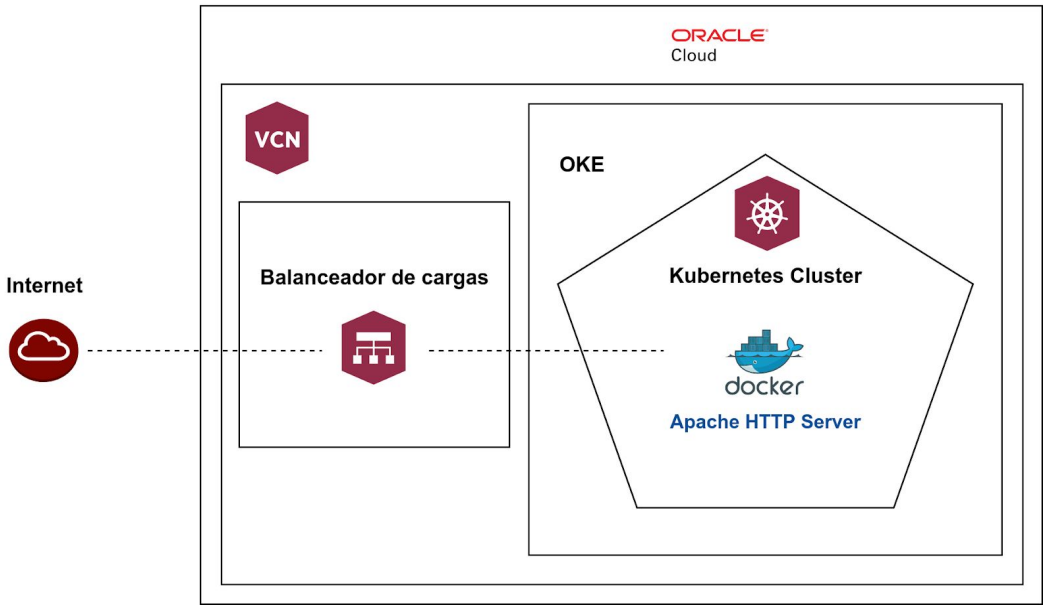
Introducción

En este documento se expone una serie de pasos para realizar el despliegue de un Apache HTTP Server en Oracle Kubernetes Engine (OKE), con el fin de obtener un servicio contenerizado y orquestado por la solución de Kubernetes disponible en Oracle Cloud Infrastructure (OCI).

Obs: Para el correcto entendimiento del ejercicio se debe tener claro conceptos básicos sobre: Oracle Cloud Infrastructure (OCI), Apache HTTP Server, Microservicios, Docker y Kubernetes.

Visión General

En el siguiente gráfico se puede observar la arquitectura a implementar a alto nivel:



Todos los componentes deben ser creados en la infraestructura cloud proporcionada por Oracle.

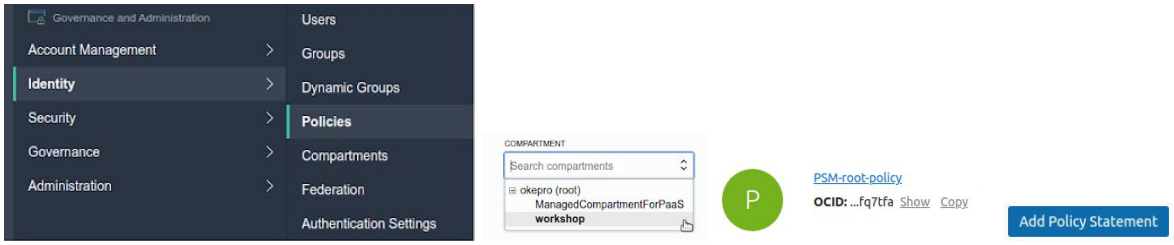
A continuación, se presentan las secciones en las que está dividido este artículo:

- **Aprovisionamiento y configuración del OKE en OCI**
- **Despliegue del Apache HTTP Server en el clúster Kubernetes (Cloud Shell)**

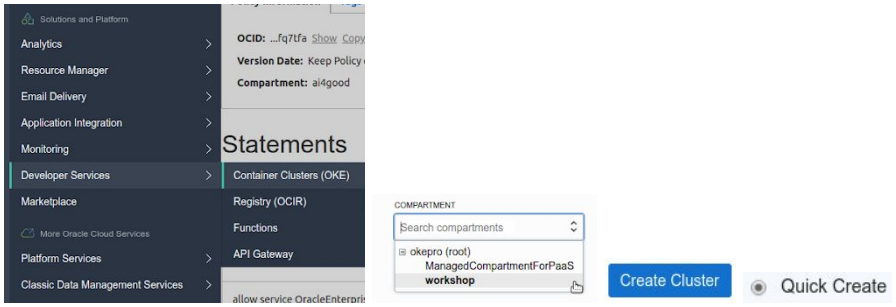
Desarrollo

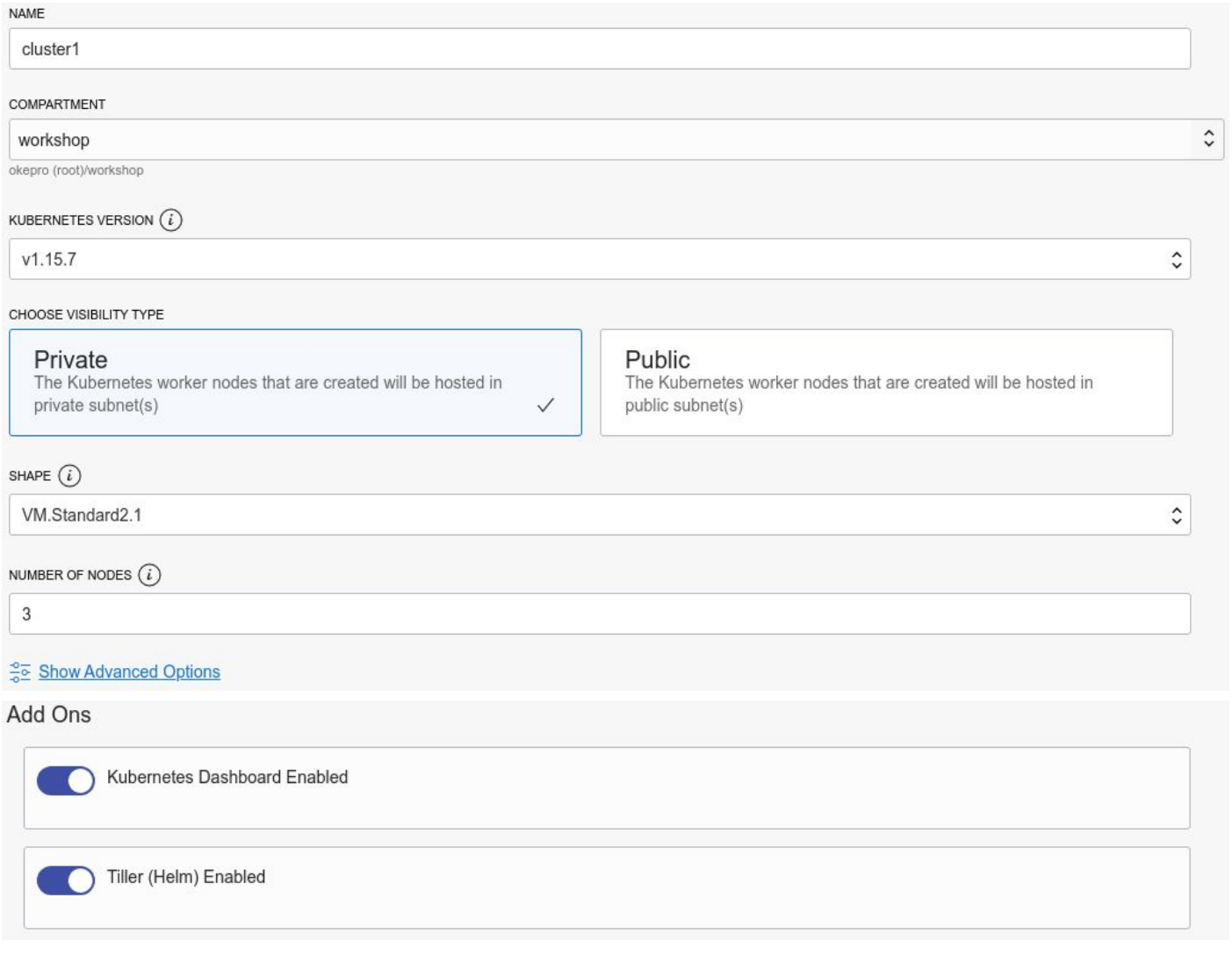
1. Aprovisionamiento y configuración del OKE en OCI

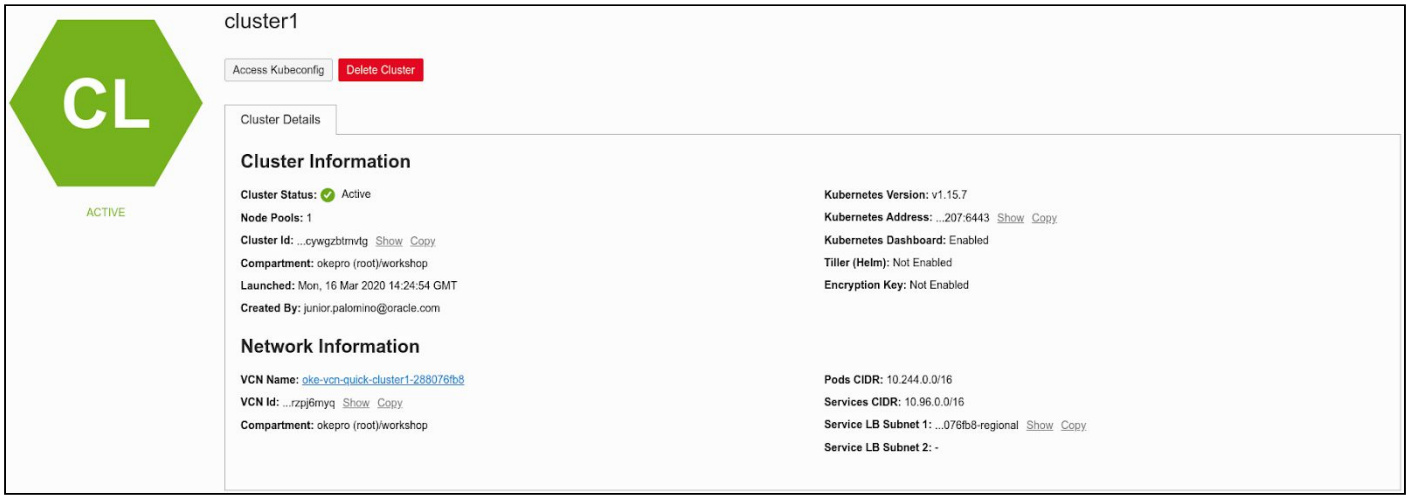
Primero, se debe agregar la siguiente política en el compartimiento raíz:
allow service OKE to manage all-resources in tenancy



Después, se crea una instancia de OKE.

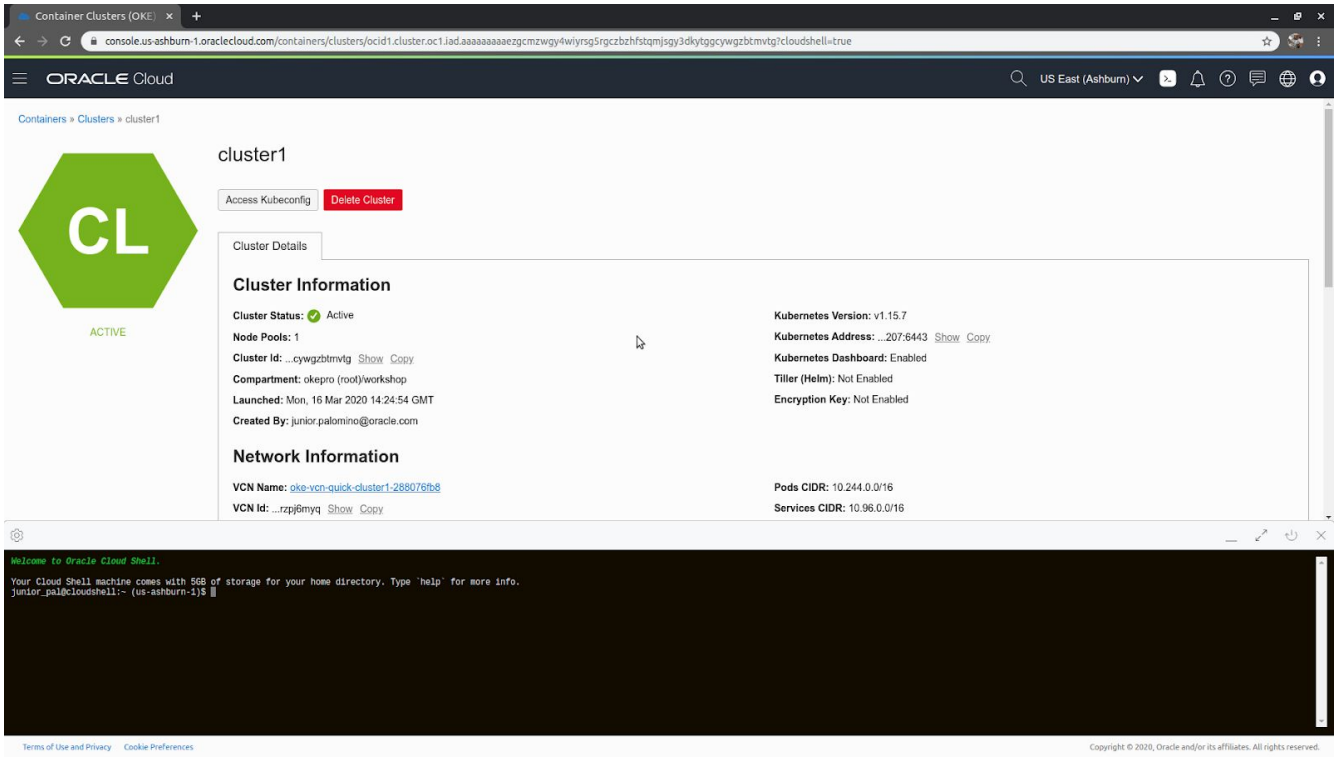






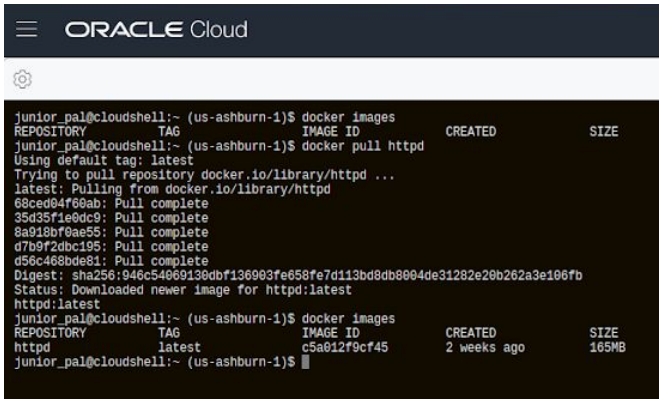
2. Despliegue del Apache HTTP Server en el clúster Kubernetes (Cloud Shell)

Para el despliegue del servidor de aplicaciones en el clúster de OKE aprovisionado, se usa la nueva función de línea de comandos de OCI, llamado Cloud Shell. Cloud Shell básicamente es el terminal de una máquina virtual que tiene instalado *oci cli*, *git*, *docker* y *kubectl*.



En el terminal se procede a bajar la imagen docker ***httpd*** y se verifican que images de docker se tiene.

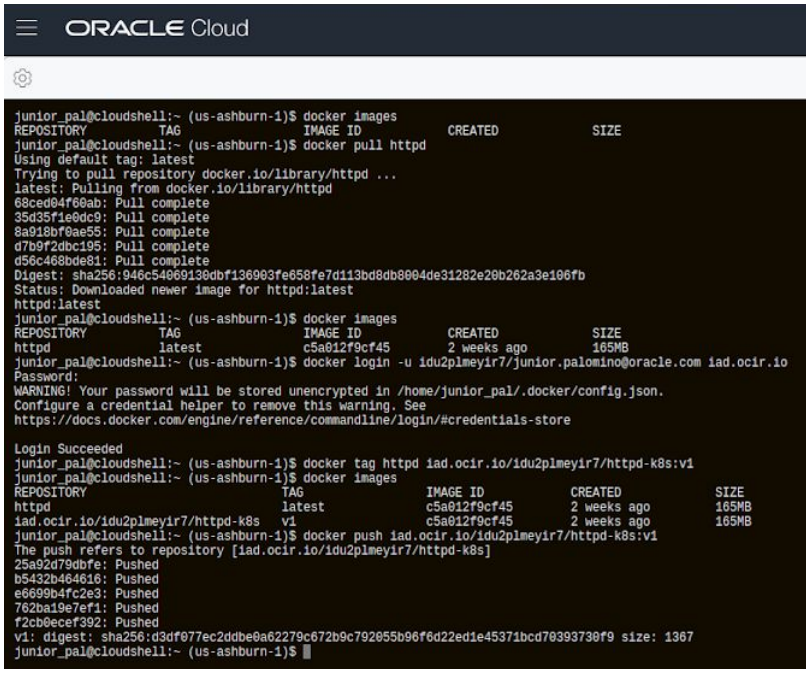
$ docker pull httpd
$ docker images



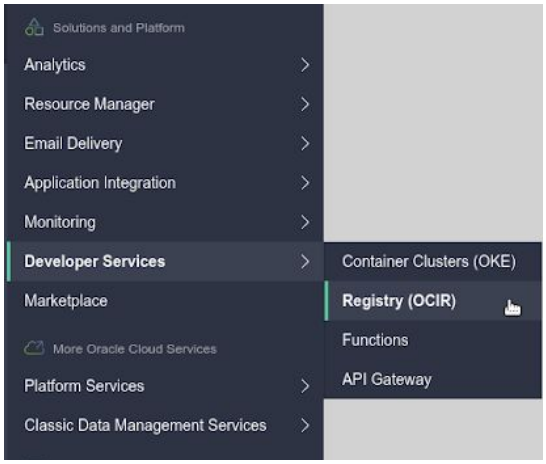
Seguidamente, se sube la imagen al repositorio local imágenes docker en OCI, conocido com Oracle Cloud Infrastructure Registry (OCIR).

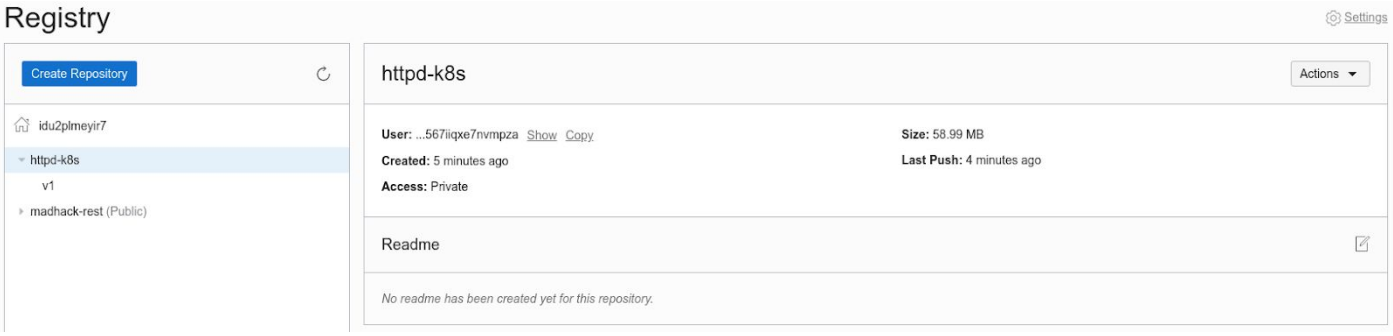
Object Storage Namespace: *idu2plmeyir7*
Auth Token: *W<Qoe#QNs}OdOvA5BCD4*

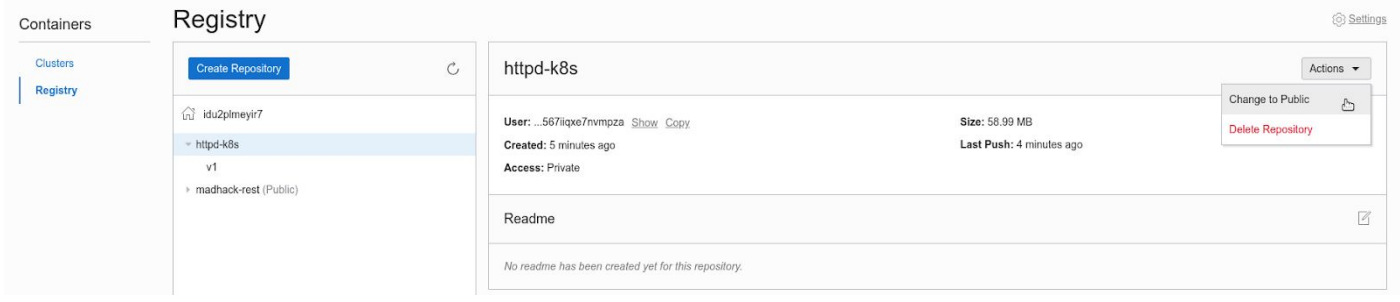
$ docker login -u idu2plmeyir7/junior.palomino@oracle.com iad.ocir.io ***(password es el auth token)***
$ docker tag httpd iad.ocir.io/idu2plmeyir7/httpd-k8s:v1
$ docker push iad.ocir.io/idu2plmeyir7/httpd-k8s:v1



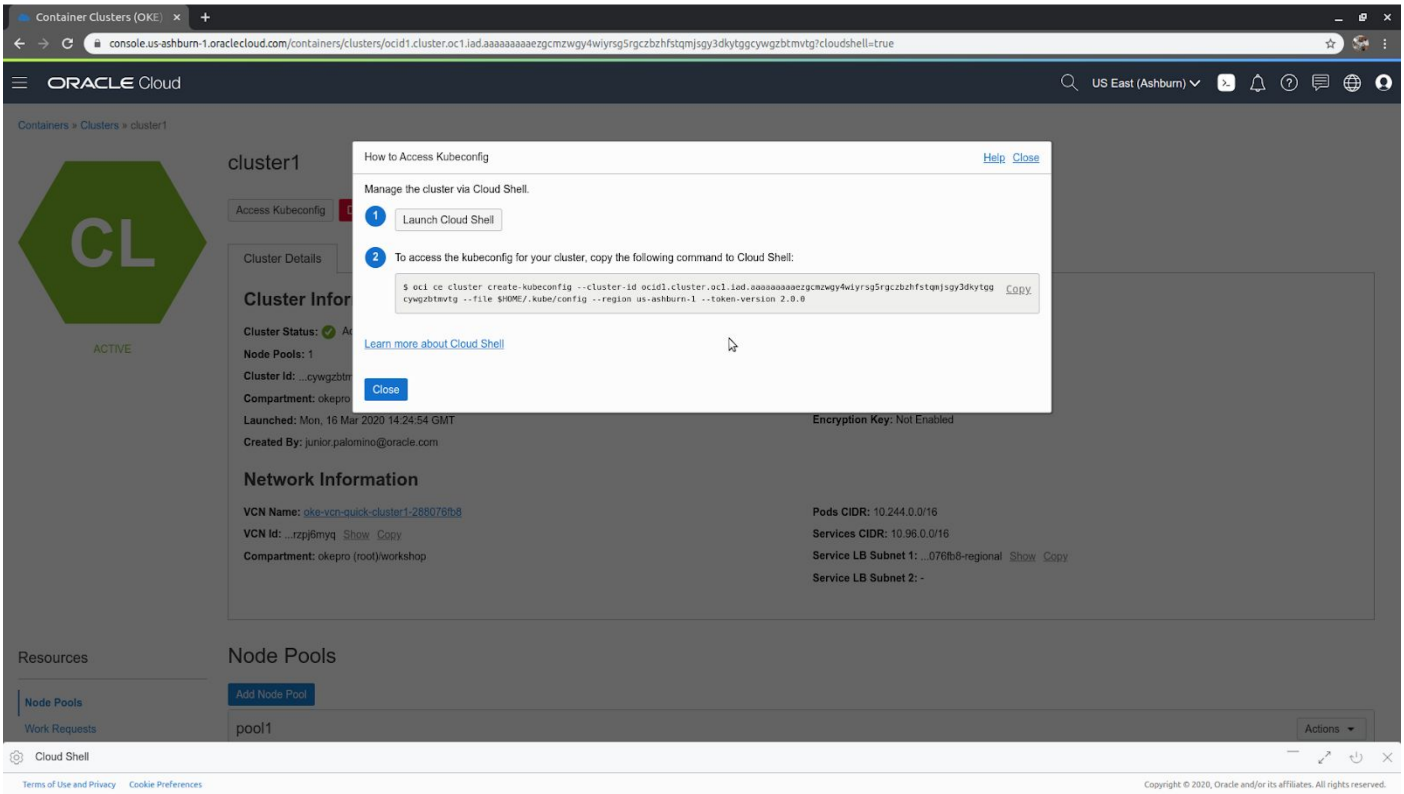
Se verifica que la imagen se subió correctamente al OCIR y se cambia a público.







Después, se tiene que crear el archivo ***Kubeconfig***, archivo de acceso al clúster de Kubernetes. El instructivo se encuentra en la propia consola, específicamente en la interfaz *de* la instancia OKE creada.



Se descarga el repositorio de github que alberga los archivos *.yaml* necesarios para desplegar la aplicación.

$ git clone https://github.com/obi10/httpd_k8s.git
$ cd http _k8s

```
junior_pal@cloudshell:~ (us-ashburn-1)$ git clone https://github.com/obi10/httpd_k8s.git
Cloning into 'httpd_k8s'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
junior_pal@cloudshell:~ (us-ashburn-1)$ ls
httpd_k8s
junior_pal@cloudshell:~ (us-ashburn-1)$ cd httpd_k8s/
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$ ls
app_kubernetes.yaml  service.yaml
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$
```

Se debe modificar el nombre del namespace (idu2plmeyir7) en el siguiente archivo:
app_kubernetes.yaml

```
 GNU nano 2.3.1

apiVersion: apps/v1
kind: Deployment
metadata:
  name: httpd-k8s-deployment
  labels:
    app: httpd-k8s
spec:
  replicas: 3
  selector:
    matchLabels:
      app: httpd-k8s
  template:
    metadata:
      labels:
        app: httpd-k8s
    spec:
      containers:
      - name: httpd-k8s
        image: iad.ocir.io/idu2plmeyir7/httpd-k8s:v1
        ports:
        - containerPort: 80
```

Luego ejecutar los siguientes comandos:
$ kubectl get all
$ kubectl apply -f app_kubernetes.yaml

```
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$ kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   8h
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$ kubectl apply -f app_kubernetes.yaml
deployment.apps/httpd-k8s-deployment created
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$ kubectl get all
NAME                                     READY   STATUS              RESTARTS   AGE
pod/httpd-k8s-deployment-f65c8dfb-9s5qr  0/1     ContainerCreating   0          7s
pod/httpd-k8s-deployment-f65c8dfb-l7n5f  0/1     ContainerCreating   0          7s
pod/httpd-k8s-deployment-f65c8dfb-qv4tw  0/1     ContainerCreating   0          7s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   8h

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/httpd-k8s-deployment   0/3     3            0           7s

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/httpd-k8s-deployment-f65c8dfb     3         3         0       7s
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$ kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/httpd-k8s-deployment-f65c8dfb-9s5qr  1/1     Running   0          56s
pod/httpd-k8s-deployment-f65c8dfb-l7n5f  1/1     Running   0          56s
pod/httpd-k8s-deployment-f65c8dfb-qv4tw  1/1     Running   0          56s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   8h

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/httpd-k8s-deployment   3/3     3            3           57s

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/httpd-k8s-deployment-f65c8dfb     3         3         3       57s
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$
```

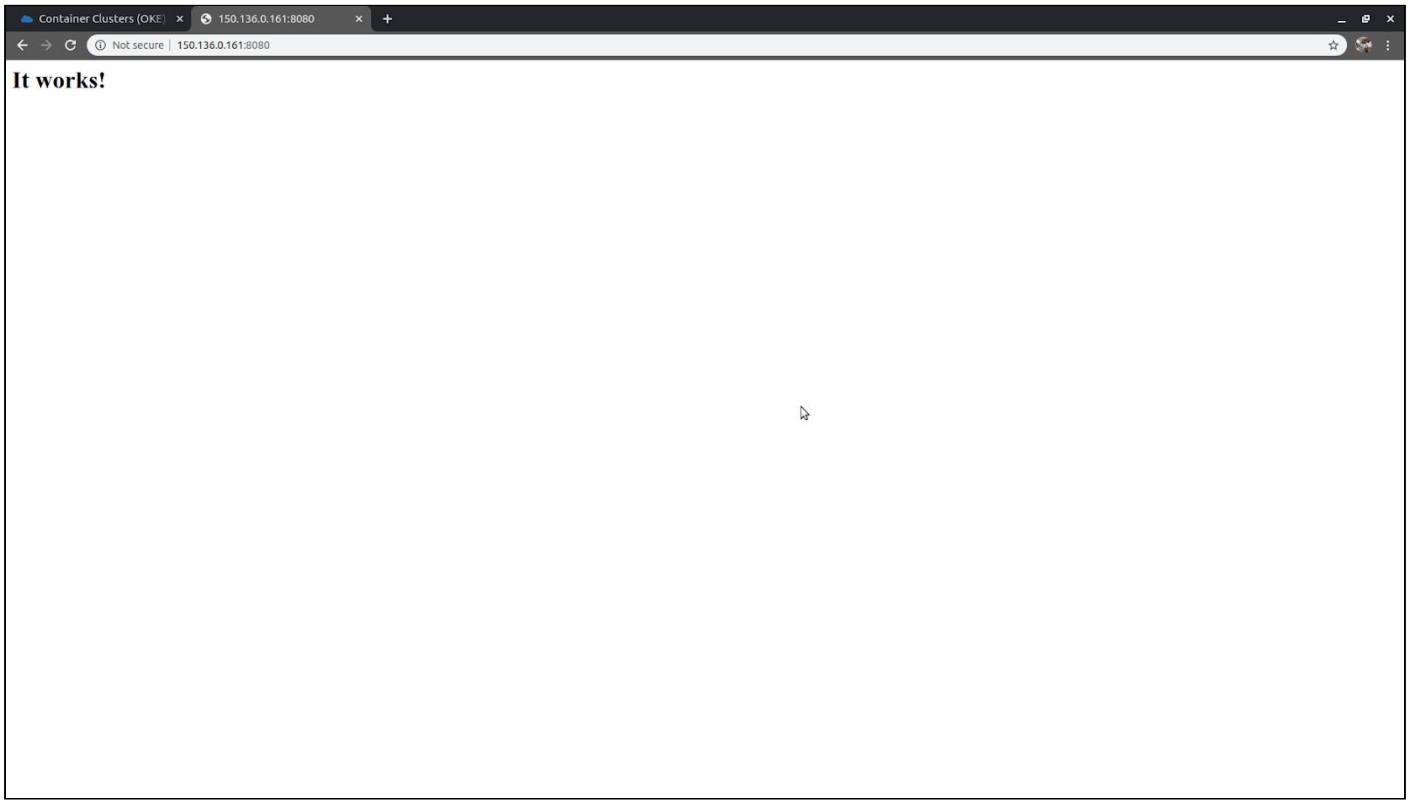
$ kubectl apply -f service.yaml
$ kubectl get all

```
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$ kubectl apply -f service.yaml

service/httpd-k8s-service created
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$
junior_pal@cloudshell:httpd_k8s (us-ashburn-1)$ kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/httpd-k8s-deployment-f65c8dfb-9s5qr   1/1     Running   0          4m31s
pod/httpd-k8s-deployment-f65c8dfb-l7n5f   1/1     Running   0          4m31s
pod/httpd-k8s-deployment-f65c8dfb-qv4tw   1/1     Running   0          4m31s

NAME                         TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/httpd-k8s-service    LoadBalancer   10.96.181.22   <pending>     8080:30949/TCP   11s
service/kubernetes           ClusterIP      10.96.0.1      <none>        443/TCP          8h

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/httpd-k8s-deployment    3/3     3            3           4m31s

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/httpd-k8s-deployment-f65c8dfb      3         3         3       4m31s
```

```
junior_pal@cloudshell:~ (us-ashburn-1)$ kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/httpd-k8s-deployment-f65c8dfb-9s5qr   1/1     Running   0          7m44s
pod/httpd-k8s-deployment-f65c8dfb-l7n5f   1/1     Running   0          7m44s
pod/httpd-k8s-deployment-f65c8dfb-qv4tw   1/1     Running   0          7m44s

NAME                         TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)          AGE
service/httpd-k8s-service    LoadBalancer   10.96.181.22   150.136.0.161   8080:30949/TCP   3m24s
service/kubernetes           ClusterIP      10.96.0.1      <none>          443/TCP          8h

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/httpd-k8s-deployment    3/3     3            3           7m44s

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/httpd-k8s-deployment-f65c8dfb      3         3         3       7m44s
junior_pal@cloudshell:~ (us-ashburn-1)$
```

ip pública del servicio: 150.136.0.161

Introducir "*<ip pública de su servicio>:8080*" en el browser para visualizar el mensaje de "It works".



Autor: Ygnacio Junior Palomino Reyes, Bachiller en Ing. Mecatrónica - Pontificia Universidad Católica del Perú (PUCP), actualmente me desempeño como GenO en Oracle Perú. **Contacto: y.palomino@pucp.pe**