Instrucciones, según mi experiencia para comenzar a utilizar Oracle Cloud Kubernetes

Una vez que tenemos el acceso a Oracle Cloud Infrastructure, es necesario crear una estructura de objetos de y para el cluster de kubernetes.

Aqui hay una página con ayuda acerca de cómo crear el Cluster

[https://docs.oracle.com/es/solutions/build-rest-java-application-with-oke/configure-your-kubernetes-cluster-oracle-cloud1.html#GUID-D1832637-FFF9-4875-9408-4F28320511E1](https://docs.oracle.com/es/solutions/build-rest-java-application-with-oke/configure-your-kubernetes-cluster-oracle-cloud1.html" \l "GUID-D1832637-FFF9-4875-9408-4F28320511E1)

Aunque de todos modos les voy a copiar en imágenes como quedaron los objetos que yo estoy usando.

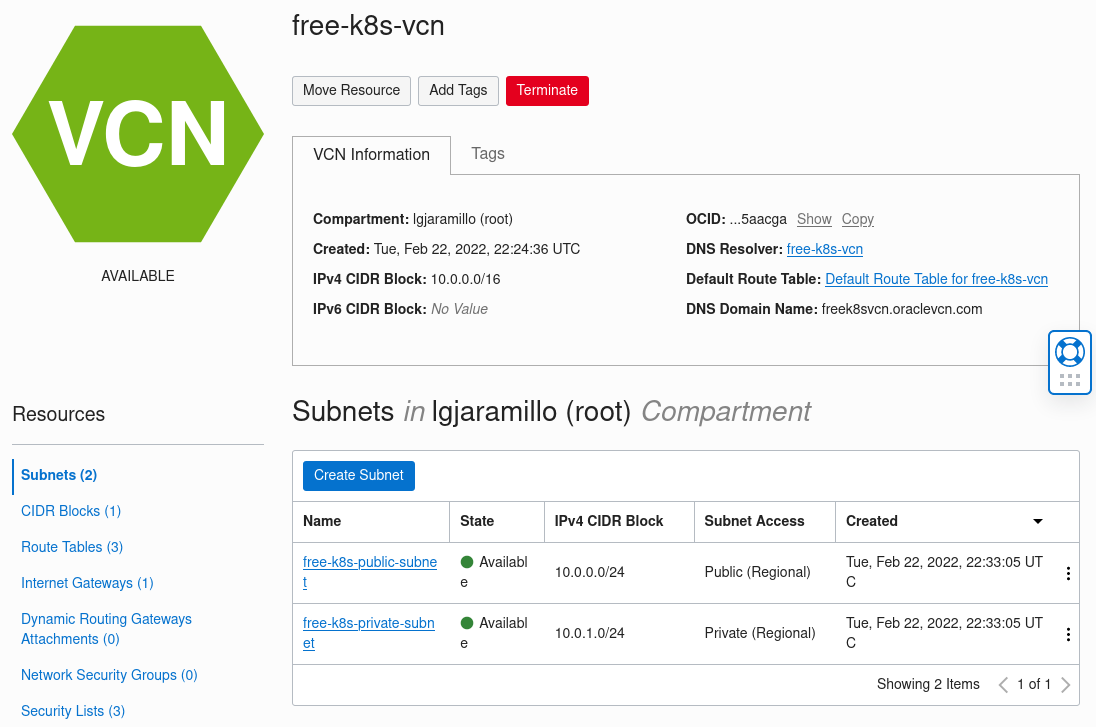
Como les mencioné la idea es que los nombres de los objetos puedan ser iguales, en la medida de lo posible, aunque estoy consciente que varios de Ustedes ya tengan experiencia utilizando la nube de Oracle y tendrán acceso a elegir componentes mas óptimos.

Finalmente, está en su decisión la asignación de los nombres en su espacio de cluster, comprendiendo bien los conceptos no habrá problema en asignar los valores de nombres cuando sea necesario.

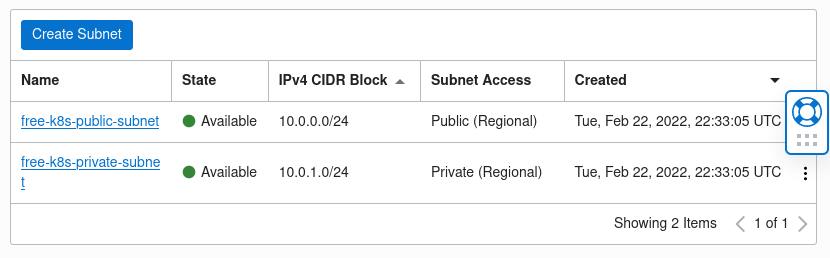
A lo largo de las páginas donde se pueden crear los componentes suele usarse un “asistente” para facilitar la creación.

1.- Opcional, es posible crear ( se recomienda ) un Compartment de modo que todo lo que se cree usando éste compartimiento se mantenga organizado ( y aislado ), yo no he realizado este paso, lo dejo a su consideración. Yo he creado los objetos en la raíz (root) de mi espacio.

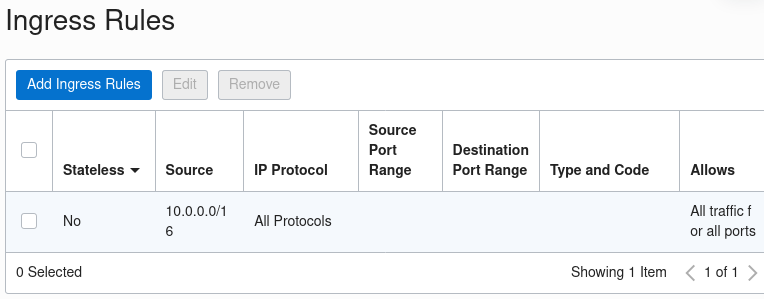
2.- Crear un Virtual Cloud Network, les muestro como están los detalles de la que he creado

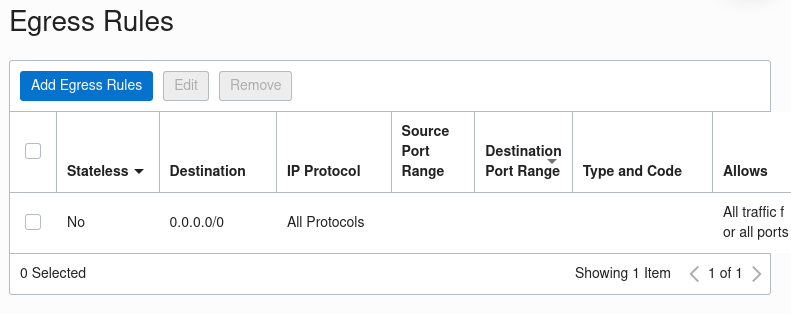


2.1.- Crear una subred pública y otra privada

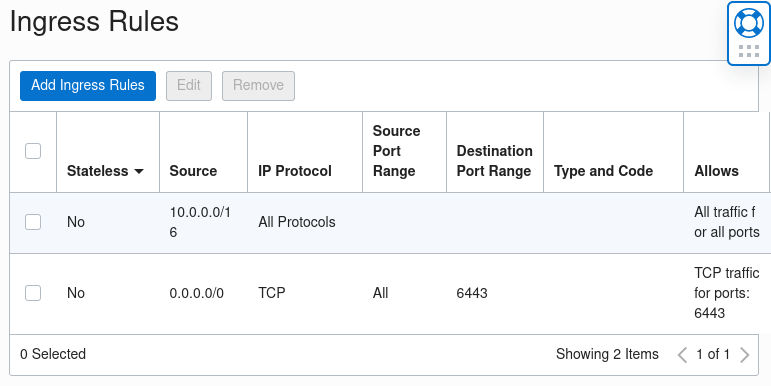


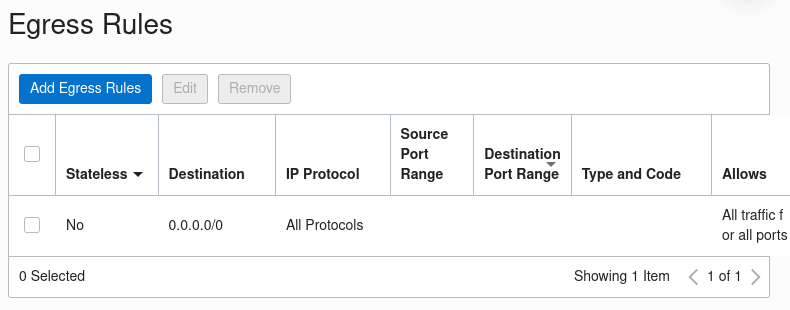
2.2.- Crear una lista de seguridad para la red privada se crean reglas de ingress y de egress



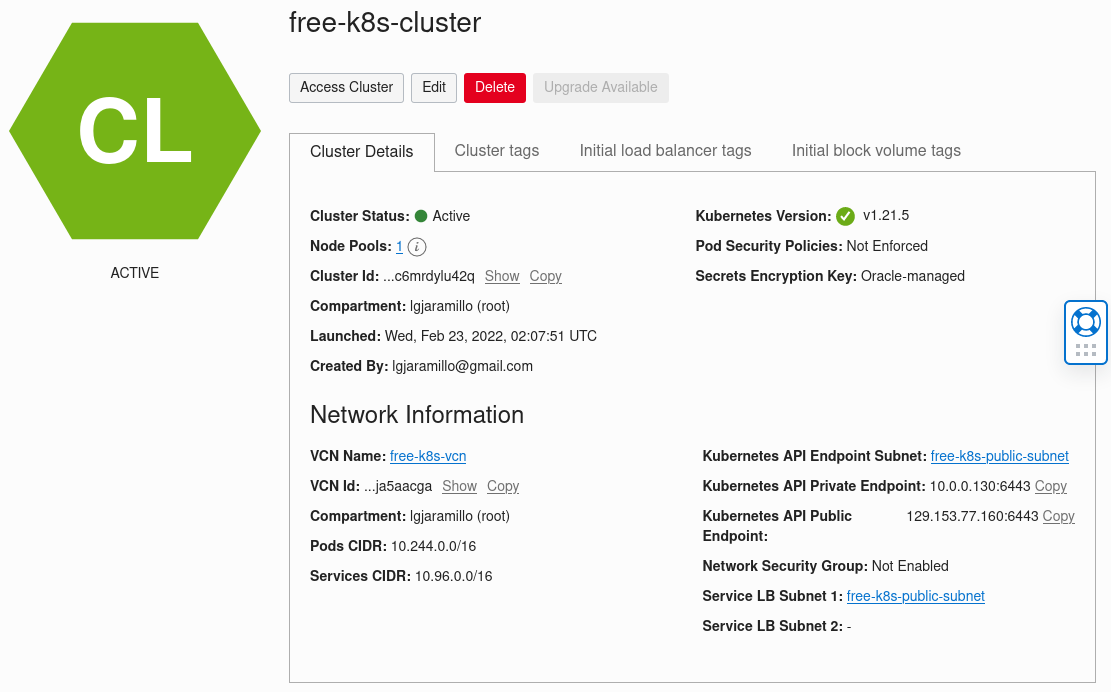


2.3.- Crear las lista de seguridad para la red pública, reglas de ingress y de egress

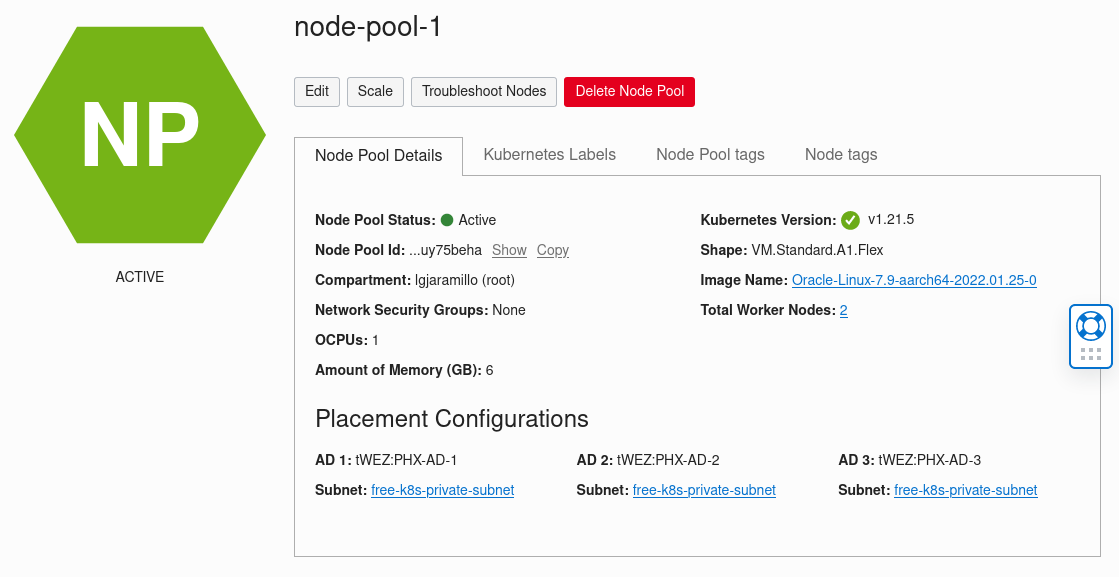




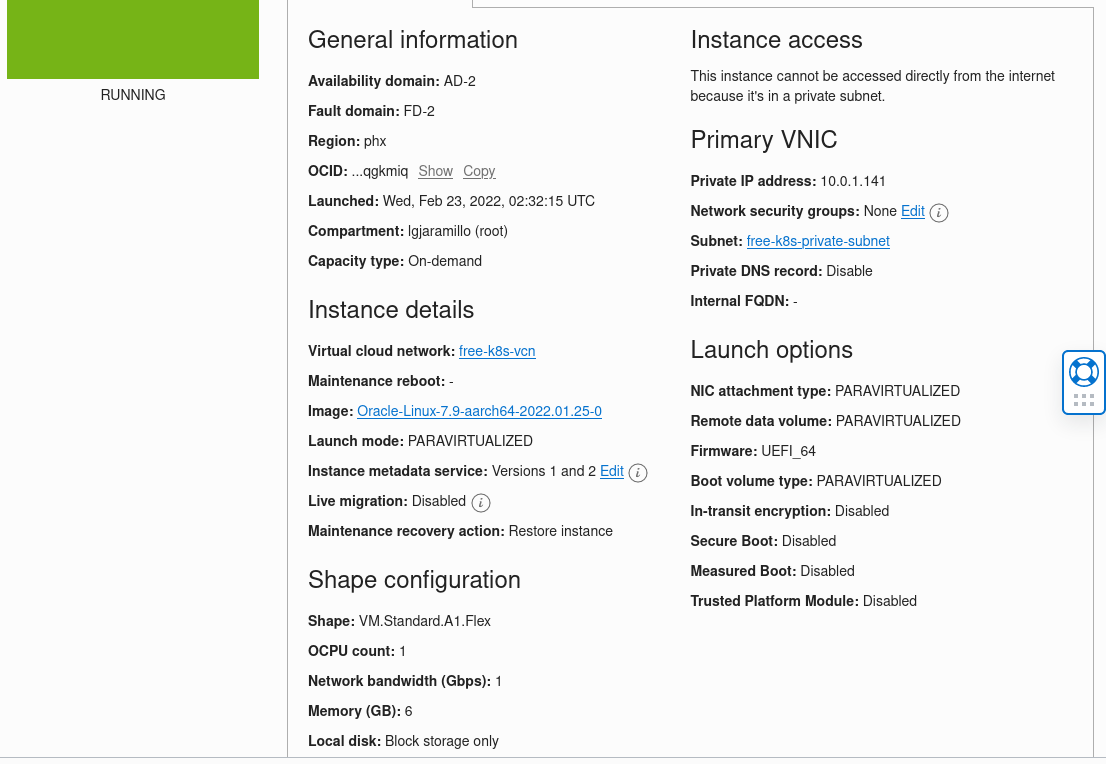
3.- Crear un cluster de kubernetes



4.- Crear un pool de nodos



5.- Dentro del pool de nodos, he creado inicialmente 2 worker nodes, les adjunto las características que usé, cada nodo representa una instancia de cómputo



Para poder tener acceso al cluster de Oracle Cloud en la línea de comandos es necesario vamos a hacer algunas modificaciones, instalaciones y configuraciones en la maquina de Oracle Linux

Deneter minikube para que no esté robando capacidad de procesamiento

minikube stop

minikube status

minikube delete --all

Instalar el OCI Cli

sudo dnf -y install oraclelinux-developer-release-el8

sudo dnf install python36-oci-cli

* Verificar la instalación

oci --version

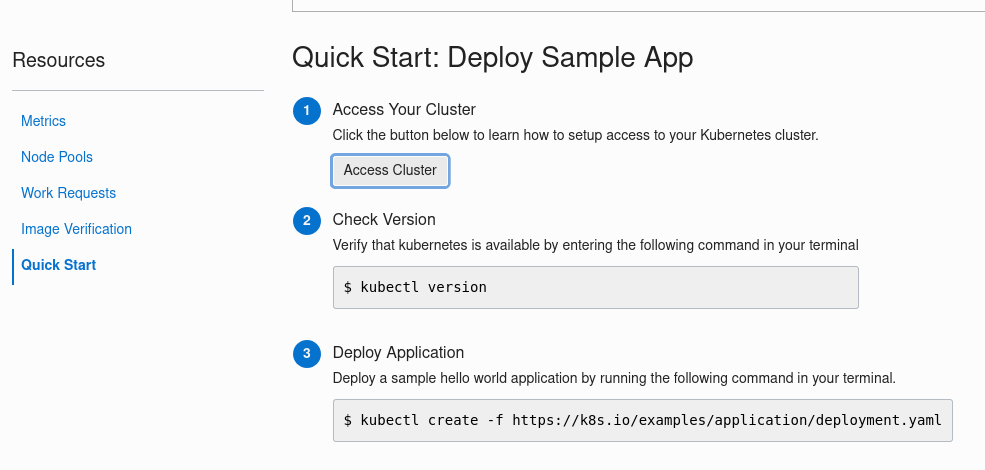
Establecer el archivo de configuración

oci setup config

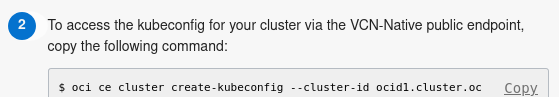
* establecer la ubicación
* User OCID: copiarlo de la pagina de [cloud.oracle.com](http://cloud.oracle.com/) - Menú superior derecho - My Profile
* Tenancy OCID: copiarlo de la misma ubicación anterior Tenancy debajo de la opción My Profile
* Seleccionar la region mediante indice de las que están en la lista y que concuerde con la que tengo configurada en [cloud.oracle.com](http://cloud.oracle.com/)
* Que si se generen las llaves de API Signing RSA key pair

Para configurar kubectl de modo que se establezca el acceso al cluster en la línea de comandos de nuestra maquina Oracle Linux

* Vamos a la página de administración de nuestro espacio en oracle cloud y buscamos los detalles del cluster. Menú (izquierdo ) Compute - Kubernetes Clusters - Seleccionamos nuestro cluster, luego vamos hacia abajo donde dice Quick Start, se abrirá una pantalla donde hay un botón que dice Acess Cluster



Se abrirá una ventana donde la opción 2 nos funcionar para ejecutar en la terminal de nuestra maquina de Oracle Linux, copiamos el texto y lo pegamos en la terminal



Al ejecutar el comando, aparecerá un mensaje que menciona que se ha guardado la congiguración en *home*/USUARIO/.kube/config a partir de este momento podremos ejecutar los comandos de kubectl para administrar el cluster de la nuber de Oracle Cloud.

kubectl get nodes

kubectl get all

Yo he creado la configuración usando una herramienta llamada Terraform, en caso de tener problemas pueden apoyarse entre sus compañeros que tengan mas experiencia o bien me pueden enviar mensajes a mi correo si desean que hagamos una práctica acerca de como se establecería la configuración usando terraform

Suerte y buen fin de semana.

Crear la infraestreuctura usando Terraform, la cual es una herramienta que permite generar la infraestructura de servidores en la nube sin tener que hacerlo a través de los menús del proveedor.

Para hacer esto vamos a tener que tener instalado Terraform, OCI-CLI ( en las paginas anteriores hay comandos para instalar el cliente de la nube de oracle ) y también tener instalado el kubect. Éste último ya está instalado en nuestra maquina con Oracle Linux

Liga hacia las imágenes de SO para los nodos del cluster:

https://docs.oracle.com/en-us/iaas/images/oraclelinux-7x/

Voy a dejar una copia de los archivos en el repositorio del curso\_kubernetes bajo la carpeta oci-infra

1.- Instalar Terraform

sudo yum install -y yum-utils

sudo yum-config-manager --add-repo <https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo>

sudo yum -y install terraform

2.- Ejecuta OCI – CLI Setup , ( ver la página anterior )

3.- Configurar las variables de Terraform

Es conveniente crear una carpeta para los archivos de terraform por ejemplo oci-infra, con estos archivos:

* variables-tf Para guardaremos las variables que se usarán en terraform
* infra.tf El archivo que describe la infraestructura
* outputs.tf Archivo que usaremos para obtener alguna salida de información

El archivo variables sería así:

variable "compartment\_id" {

type = string

description = "The compartment to create the resources in"

}

variable "region" {

type = string

description = "The region to provision the resources in"

}

variable "ssh\_public\_key" {

type = string

description = "The SSH public key to use for connecting to the worker nodes"

}

Para cada variable es posible establecer un valor por default especificando el valor entre comillas, debajo de “decription”, por ejemplo para el valor de región yo he puesto:

default = “us-phoenix-1”

Aunque esto está hecho para tomar los valores usando variables de entorno de linux, si les es complicado, es necesario llenar estos datos con un valor por default desde este archivo.

El compartment\_id, solo se requiere si no desean usar el root y separar el proyecto en un compartmente que ya hayan creado previamente en la nube de oracle.

Yo tuve que poner el valor por default de ssh\_public\_key ya que no me dejaba establecerlo como variable de entorno, el valor lo obtuve abrien el archivo que se graba en *HOME/USUARIO/.oci/oci\_api\_key\_public.pem* me quedó mas o menos así:

variable "ssh\_public\_key" {

type = string

description = "The SSH public key to use for connecting to the worker nodes"

default = "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwKTA4gtn3NGHon24cPh/B3m3XFvUp5wKVRjVxfoPdNKfaDlqSktgYMImfrpIr+I+KeNw4TTh+1Z4Oop18VpWke2vWnITp9SUdDiRHl6156QR+hFJAqcE7rgkhveriuh34nfjk3rfnkuwnekwjntjTwqR9NOzolNaPZCOOyO2ohlwmNNABmY5ujlqQVueSR0fOnr4TZRs6I7UiNiIrkvjbNxHS6yzr5Elr+LJ56+slkQri2zg8pv7P0Ld1G/auhAd+x3RIg5jkcwIDAQAB"

}

La forma de expecificar los valores de estas variables sin tener que usar valores por default grabados en el archivo sería usando el comando de linux export:

$ export TF\_VAR\_compartment\_id=<your compartment ocid>

$ export TF\_VAR\_region=<your region>

$ export TF\_VAR\_ssh\_public\_key=<your public key>

4.- Especificar los recursos para el cluster de kubernetes:

Se edita el archivo infra.tf a mi me quedó con este contenido:

provider "oci" {

region = var.region

}

module "vcn" {

source = "oracle-terraform-modules/vcn/oci"

version = "3.1.0"

compartment\_id = var.compartment\_id

region = var.region

internet\_gateway\_route\_rules = null

local\_peering\_gateways = null

nat\_gateway\_route\_rules = null

vcn\_name = "free-k8s-vcn"

vcn\_dns\_label = "freek8svcn"

vcn\_cidrs = ["10.0.0.0/16"]

create\_internet\_gateway = true

create\_nat\_gateway = true

create\_service\_gateway = true

}

resource "oci\_core\_security\_list" "private\_subnet\_sl" {

compartment\_id = var.compartment\_id

vcn\_id = module.vcn.vcn\_id

display\_name = "free-k8s-private-subnet-sl"

egress\_security\_rules {

stateless = false

destination = "0.0.0.0/0"

destination\_type = "CIDR\_BLOCK"

protocol = "all"

}

ingress\_security\_rules {

stateless = false

source = "10.0.0.0/16"

source\_type = "CIDR\_BLOCK"

protocol = "all"

}

}

resource "oci\_core\_security\_list" "public\_subnet\_sl" {

compartment\_id = var.compartment\_id

vcn\_id = module.vcn.vcn\_id

display\_name = "free-k8s-public-subnet-sl"

egress\_security\_rules {

stateless = false

destination = "0.0.0.0/0"

destination\_type = "CIDR\_BLOCK"

protocol = "all"

}

ingress\_security\_rules {

stateless = false

source = "10.0.0.0/16"

source\_type = "CIDR\_BLOCK"

protocol = "all"

}

ingress\_security\_rules {

stateless = false

source = "0.0.0.0/0"

source\_type = "CIDR\_BLOCK"

protocol = "6"

tcp\_options {

min = 6443

max = 6443

}

}

}

resource "oci\_core\_subnet" "vcn\_private\_subnet" {

compartment\_id = var.compartment\_id

vcn\_id = module.vcn.vcn\_id

cidr\_block = "10.0.1.0/24"

route\_table\_id = module.vcn.nat\_route\_id

security\_list\_ids = [oci\_core\_security\_list.private\_subnet\_sl.id]

display\_name = "free-k8s-private-subnet"

prohibit\_public\_ip\_on\_vnic = true

}

resource "oci\_core\_subnet" "vcn\_public\_subnet" {

compartment\_id = var.compartment\_id

vcn\_id = module.vcn.vcn\_id

cidr\_block = "10.0.0.0/24"

route\_table\_id = module.vcn.ig\_route\_id

security\_list\_ids = [oci\_core\_security\_list.public\_subnet\_sl.id]

display\_name = "free-k8s-public-subnet"

}

resource "oci\_containerengine\_cluster" "k8s\_cluster" {

compartment\_id = var.compartment\_id

kubernetes\_version = "v1.21.5"

name = "free-k8s-cluster"

vcn\_id = module.vcn.vcn\_id

endpoint\_config {

is\_public\_ip\_enabled = true

subnet\_id = oci\_core\_subnet.vcn\_public\_subnet.id

}

options {

add\_ons {

is\_kubernetes\_dashboard\_enabled = false

is\_tiller\_enabled = false

}

kubernetes\_network\_config {

pods\_cidr = "10.244.0.0/16"

services\_cidr = "10.96.0.0/16"

}

service\_lb\_subnet\_ids = [oci\_core\_subnet.vcn\_public\_subnet.id]

}

}

data "oci\_identity\_availability\_domains" "ads" {

compartment\_id = var.compartment\_id

}

resource "oci\_containerengine\_node\_pool" "k8s\_node\_pool" {

cluster\_id = oci\_containerengine\_cluster.k8s\_cluster.id

compartment\_id = var.compartment\_id

kubernetes\_version = "v1.21.5"

name = "free-k8s-node-pool"

node\_config\_details {

placement\_configs {

availability\_domain = data.oci\_identity\_availability\_domains.ads.availability\_domains[0].name

subnet\_id = oci\_core\_subnet.vcn\_private\_subnet.id

}

placement\_configs {

availability\_domain = data.oci\_identity\_availability\_domains.ads.availability\_domains[1].name

subnet\_id = oci\_core\_subnet.vcn\_private\_subnet.id

}

placement\_configs {

availability\_domain = data.oci\_identity\_availability\_domains.ads.availability\_domains[2].name

subnet\_id = oci\_core\_subnet.vcn\_private\_subnet.id

}

size = 1

}

node\_shape = "VM.Standard.A1.Flex"

node\_shape\_config {

memory\_in\_gbs = 6

ocpus = 1

}

node\_source\_details {

image\_id = "ocid1.image.oc1.phx.aaaaaaaavahyixgct7zsbidfsive2npkamdb4lmztoux52t35fymijanjsdq"

source\_type = "image"

}

initial\_node\_labels {

key = "name"

value = "free-k8s-cluster"

}

}

El archivo outputs.tf tendría el siguiente contenido:

output "k8s-cluster-id" {

value = oci\_containerengine\_cluster.k8s\_cluster.id

}

5.- Ejecutar terraform

Solo serían 2 comandos para crear la infraestructura, primero la inicialización y después la creación:

terraform init

terraform apply

El proceso puede llevar mucho tiempo, podremos ir comprobando el avance en la pantalla del símbolo de comandos o bien en la página de infraestructura de oracle.

6.- Configurar el acceso al cluster de kubernetes.

El último paso seria configurar el acceso al cluster de kubernetes, ese paso ya se ha descrito en la página 7 de este documento.

Finalmente, en mi experiencia, solo creaba 1 nodo de trabajo, no creaba 2, el segundo nodo lo he creado manualmente en la página de oracle cloud.