



COMPUTACIÓN EN LA NUBE

OSCAR HERNAN MONDRAGON MARTINEZ

MÍLMAX JEFFREY DUQUE ASPRILLA
CÓDIGO: 2220164

SEBASTIAN AMILKAR MURILLO HURTADO
CÓDIGO:2151016

GRUPO: 1

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
SANTIAGO DE CALI
13 DE MAYO DE 2022

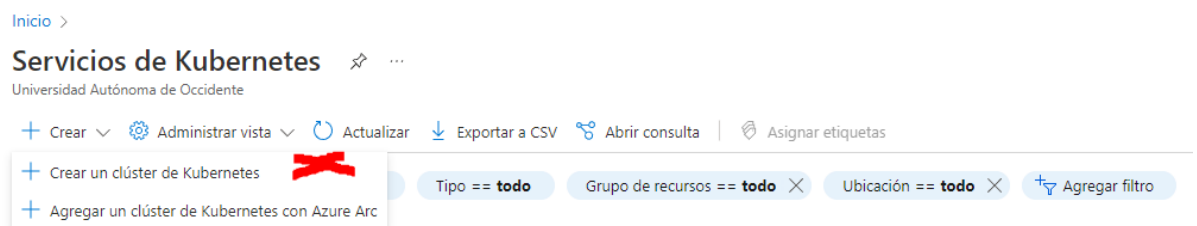
IMPLEMENTACIÓN DE AZURE KUBERNETES SERVICE

A continuación presentamos cómo crear un cluster de kubernetes en la plataforma de azure. Cabe resaltar que debemos estar registrados para poder realizar dicha actividad.

- En primer lugar debemos dirigirnos a <https://portal.azure.com/#home>
- En la página principal vamos a seleccionar Servicios de Kubernetes. En caso de que no esté disponible en el panel. Podemos hacer uso del buscador y digitamos el servicio que deseamos crear.



- Al haber seleccionado el servicio, se nos desplegará una nueva ventana. donde procederemos con la creación de nuestro kubernetes. para esto debemos seleccionar el botón **Crear**.



- En la página Datos básicos, configure las siguientes opciones:
 - **Detalles del proyecto:**
Seleccione una suscripción de Azure.
Seleccione o cree un **grupo de recursos** de Azure, como *myResourceGroup*.
 - **Detalles del clúster:**
Asegúrese de que el valor de **Configuración preestablecida** es *Estándar (\$\$)* . Para obtener más información sobre las

configuraciones preestablecidas, vea Valores preestablecidos de configuración de clúster en Azure Portal.

Escriba un **Nombre del clúster de Kubernetes**, como *myAKSCluster*.

Seleccione una Región para el clúster de AKS y deje el valor predeterminado seleccionado para **Versión de Kubernetes**.

Seleccione **99,5 %** para **Disponibilidad del servidor de API**.

- **Grupo de nodos principal:**

Deje los valores predeterminados seleccionados.

Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Inicio > Servicios de Kubernetes >

Crear un clúster de Kubernetes

Datos básicos

Grupos de nodos

Acceso

Redes

Integraciones

Opciones avanzadas

Etiquetas

Revisar y crear

Azure Kubernetes Service (AKS) administra el entorno de Kubernetes hospedado, a la vez que facilita y agiliza la implementación y la administración de aplicaciones en contenedores sin necesidad de experiencia relativa. También elimina la carga de las operaciones en curso y el mantenimiento mediante el aprovisionamiento, la actualización y el escalado de los recursos a petición, sin tener que desconectar las aplicaciones. [Más información sobre Azure Kubernetes Service](#)

Detalles del proyecto

Seleccione una suscripción para administrar los recursos implementados y los costos. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción *

Azure for Students

Grupo de recursos *

(Nuevo) Grupo de recursos

[Crear nuevo](#)

Detalles del clúster

Configuración preestablecida del clúster

Estándar (\$\$)

Para personalizar rápidamente el clúster de Kubernetes, elija una de las configuraciones preestablecidas anteriores. Puede modificar estas configuraciones en cualquier momento.
[Obtener más información y comparar valores preestablecidos](#)

Nombre del clúster de Kubernetes *

Región *

(Asia Pacific) Australia East

Zonas de disponibilidad

Zonas 1,2,3

Se recomienda la alta disponibilidad para la configuración estándar.

Versión de Kubernetes *

1.22.6 (predeterminado)

Disponibilidad del servidor de API

99,95%

Optimice la disponibilidad.

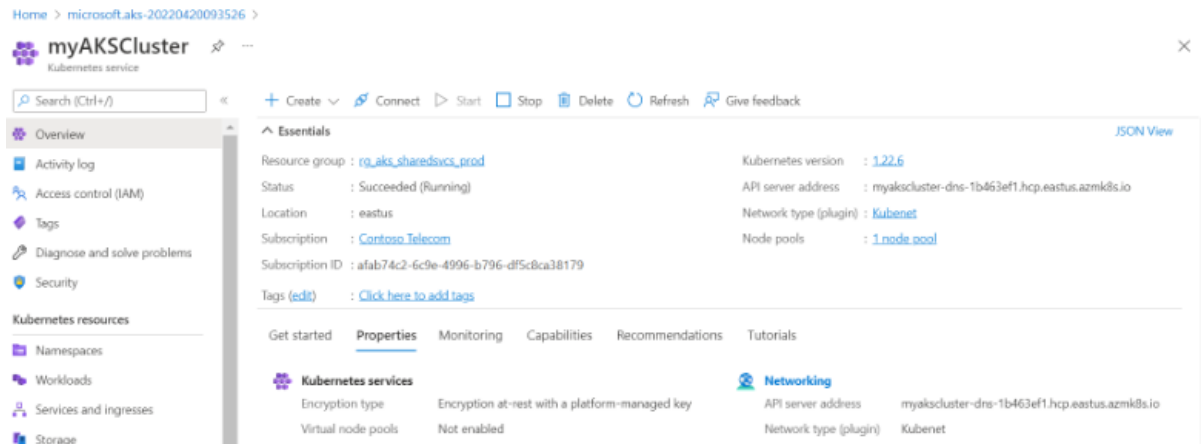
99,5%

Revisar y crear

< Anterior

Siguiente: Grupos de nodos >

- Ya que tenemos nuestra configuración, podemos presionar el botón, **Revisar y crear**. El proceso tardará unos minutos. Al haberse completado la creación kubernetes ya podemos seleccionar el botón **ir al recurso**.



- Una vez creado nuestro cluster, procedemos a verificar su estado realizando la conexión desde nuestra máquina virtual. la cual se encuentra configurada localmente.

para conectarnos ejecutamos el comando **az login**. El cual nos proporciona un enlace en el cual debemos digitar el token para poder sincronizar nuestra máquina con el portal de azure.

```
vagrant@machine1: ~$ az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code A5ECAZGXB to authenticate.
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "693cbea0-4ef9-4254-8977-76e05cb5f556",
    "id": "62f81e13-66dc-404b-9aeb-74e1bfdee2f4",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "693cbea0-4ef9-4254-8977-76e05cb5f556",
    "user": {
      "name": "sebastian.murillo@uao.edu.co",
      "type": "user"
    }
  }
]
vagrant@machine1: ~$
```

Luego procedemos a conectarnos con nuestro **servicio de AKS**

```
vagrant@machine1: ~  
vagrant@machine1:~$ az aks get-credentials --resource-group compuNube --name mi_kubern  
et  
Merged "mi_kubernetes" as current context in /home/vagrant/.kube/config  
vagrant@machine1:~$
```

En donde digitamos el código mostrado en consola y reemplazamos **compuNube** por nombre del grupo y **mi_kubern** por el nombre establecido del cluster. Esto nos permitirá conectarnos con nuestro servicio, descargar las credenciales y posteriormente usarlas en la configuración del CLI.

- Ahora procedemos a realizar la configuración necesaria para desplegar nuestra **aplicación de clasificación de imágenes en el cluster de AKS**
- Vamos a hacer uso de la carpeta **kubermatic-dl** del repositorio: una vez clonado el repositorio vamos a crear la imagen de nuestro contenedor con el siguiente comando:

```
sudo docker build -t kubermatic-dl:latest .
```

Recordemos que debemos de ejecutar este comando en el directorio **kubermatic-dl**. dicho comando creará la imagen con la configuración correspondiente al archivo **Dockerfile**

Ahora procederemos a iniciar nuestro contenedor con la imagen ya creada y especificamos un puerto de comunicación.

```
sudo docker run -d -p 5000:5000 kubermatic-dl
```

Luego verificamos que nuestro contenedor esté funcionando con el comando:

```
sudo docker ps -a
```

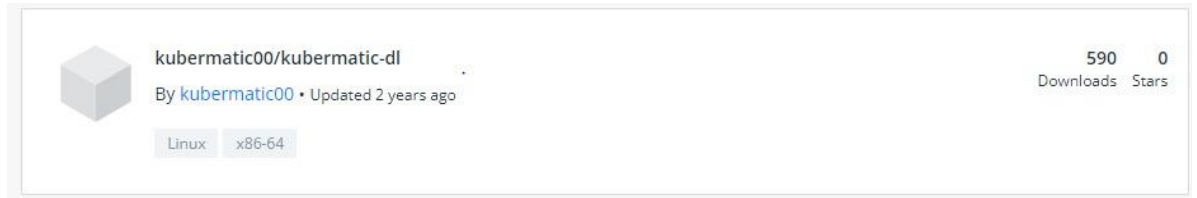
```
vagrant@machine1: ~/proyecto  
vagrant@machine1:~/proyecto/kubermatic-dl$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED NAMES	STATUS	PORTS
7984a30efbe4	kubermatic-dl	"python app.py"	2 days ago	Exited (1) 2 days ago	cranky_greider

como ya tenemos nuestro contenedor corriendo, procederemos a ejecutar la aplicación en el **cluster de AKS**. para esto vamos a ejecutar el siguiente comando

```
kubectl apply -f image_classifier.yaml
```

Nota: Se hace uso de la imagen original de la aplicación. debido a que cuando se estaba creando nuestra propia imagen de la implementación. esta se creaba con errores debido a que no era posible conectarse a internet y los paquetes no se descargan. a continuación imagen real de la aplicación

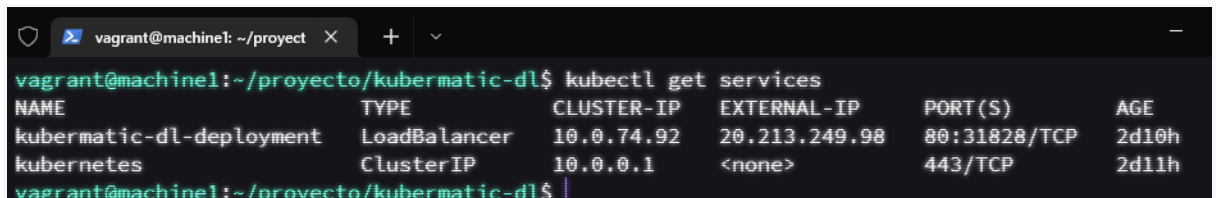


Ahora, podemos seguir con nuestra implementación ,y debemos desplegar nuestro servicio para uso de los usuarios. para esto vamos a exponer nuestro servicio de tipo loadbalancer y especificando un puerto con el siguiente comando:

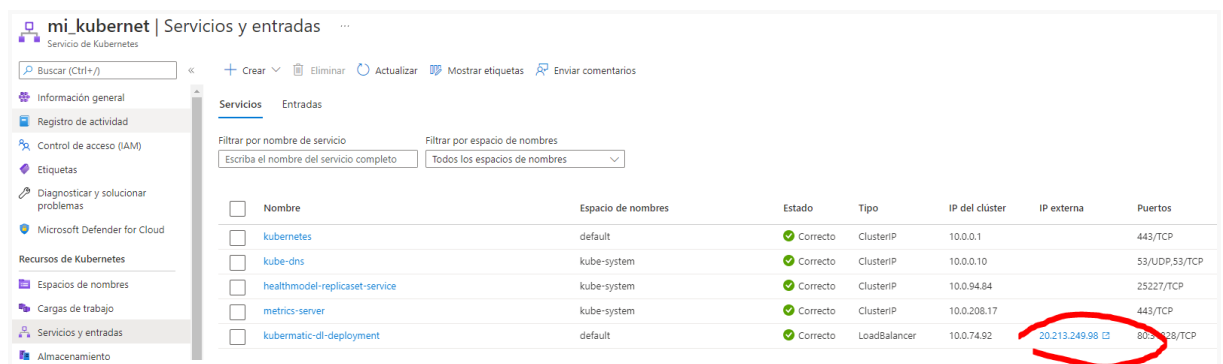
```
kubectl expose deployment kubernatic-dl-deployment
--type=LoadBalancer --port 80 --target-port 5000
```

Luego vamos a verificar el estado de nuestra implementación y conocer su dirección ip, la cual vamos a usar para poder realizar la predicción con nuestras imágenes. para esto usamos el comando

```
kubectl get services
```

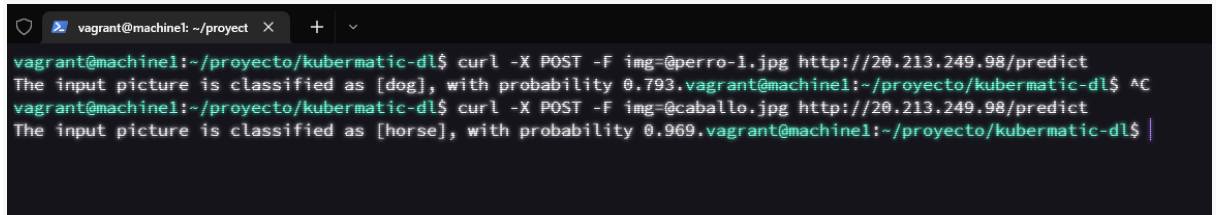


También podemos verificar nuestra IP en el portal de azure. Nos dirigimos en la sesión de **Servicios y entradas**



Como se puede observar en el apartado de ip externa, ya tenemos una dirección para usar nuestro modelo. para esto usaremos el siguiente comando:

```
curl -X POST -F img=@caballo.jpg http://20.213.249.98/predict
```



```
vagrant@machine1: ~/proyecto x + v
vagrant@machine1:~/proyecto/kubermatic-dl$ curl -X POST -F img=@perro-1.jpg http://20.213.249.98/predict
The input picture is classified as [dog], with probability 0.793.vagrant@machine1:~/proyecto/kubermatic-dl$ ^C
vagrant@machine1:~/proyecto/kubermatic-dl$ curl -X POST -F img=@caballo.jpg http://20.213.249.98/predict
The input picture is classified as [horse], with probability 0.969.vagrant@machine1:~/proyecto/kubermatic-dl$ |
```