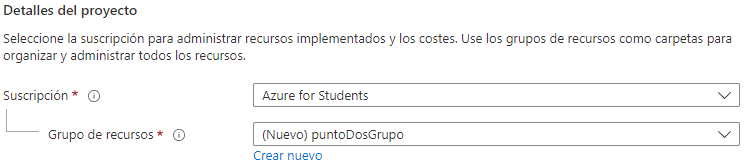
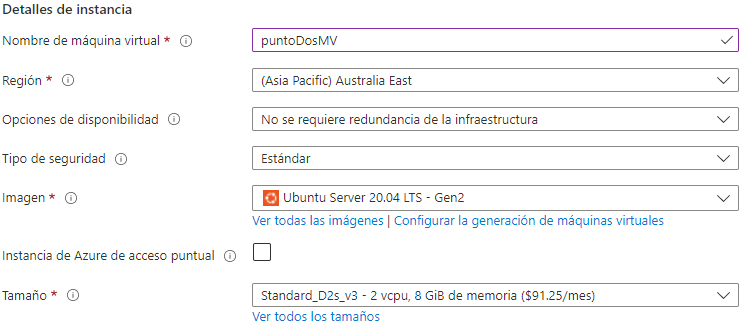
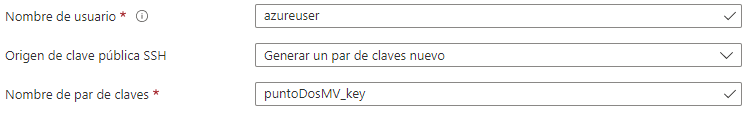
2. Aplicación de clasificación de imágenes en Azure

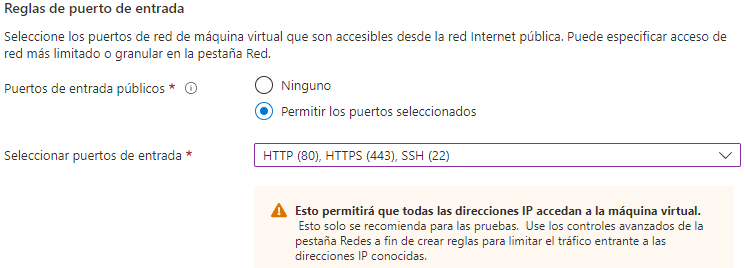
<https://opensource.com/article/20/9/deep-learning-model-kubernetes>

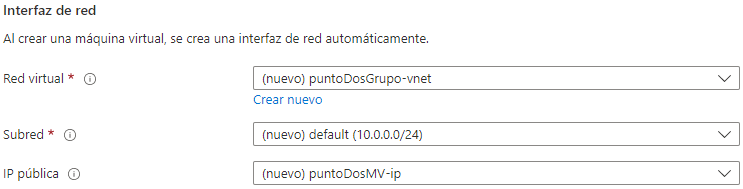
MV



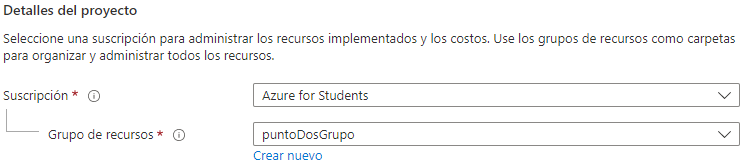


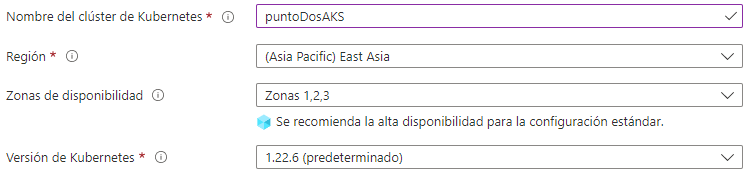


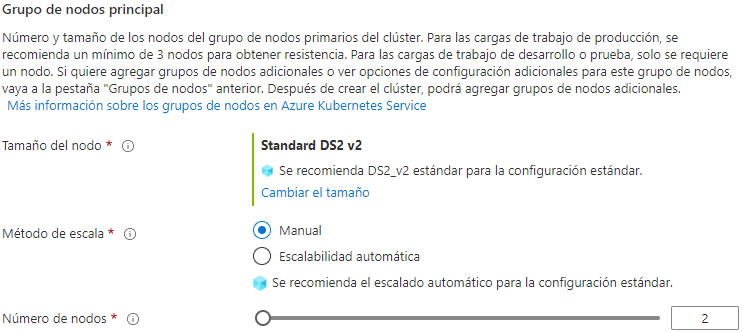




AKS







<https://docs.microsoft.com/en-us/azure/aks/learn/quick-kubernetes-deploy-cli>

Nos conectamos al cloud shell

**NOTA:** Esta parte no funciona

Cuando ejecutamos comandos de kubectl directamente nos aparece el siguiente error



Según foros se debe a que el cluster AKS tiene la privacidad activada

**NOTA:** Esta parte si funciona

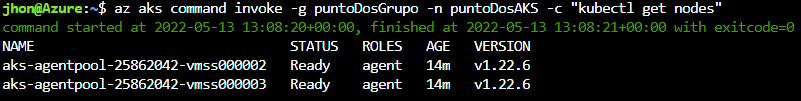
Para ejecutar comandos kubectl hacemos

| az aks command invoke -g puntoDosGrupo -n puntoDosAKS -c "kubectl get nodes" |
| --- |

1. Verify the connection to your cluster using the kubectl get command. This command returns a list of the cluster nodes

| kubectl get nodes |
| --- |

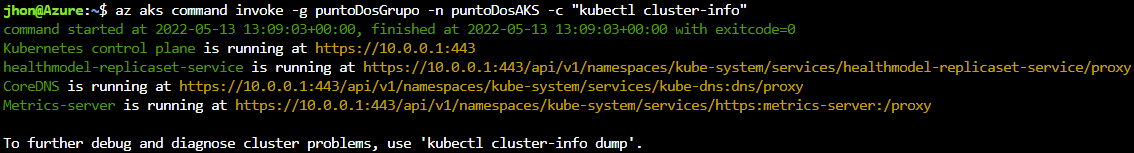
| az aks command invoke -g puntoDosGrupo -n puntoDosAKS -c "kubectl get nodes" |
| --- |



1. Using kubectl, check the cluster information, such as the services that kube-system starts on your cluster:

| kubectl cluster-info |
| --- |

| az aks command invoke -g puntoDosGrupo -n puntoDosAKS -c "kubectl cluster-info" |
| --- |



1. To run the container in the cluster, you need to create a deployment (deployment.yaml) and apply it to the cluster:

| apiVersion: apps/v1  kind: Deployment  metadata:  name: kubermatic-dl-deployment  spec:  selector:  matchLabels:  app: kubermatic-dl  replicas: 3  template:  metadata:  labels:  app: kubermatic-dl  spec:  containers:  - name: kubermatic-dl  image: kubernetespytorch/kubermatic  imagePullPolicy: Always  ports:  - containerPort: 8080 |
| --- |

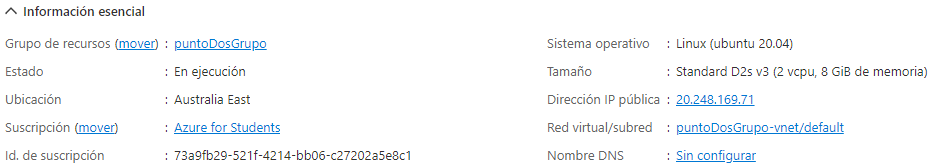
kubernetespytorch/kubermatic esta imagen está en mi docker hub

| kubectl apply -f deployment.yaml |
| --- |

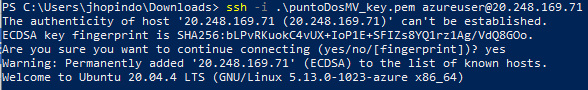
En nuestro caso, primero debemos crear el archivo YAML en la máquina virtual del servidor

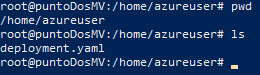
Para ellos nos conectamos por ssh en la MV y creamos el archivo

La MV tiene ip publica 20.248.169.71



| ssh -i .\puntoDosMV\_key.pem azureuser@20.248.169.71 |
| --- |





| az aks command invoke -g puntoDosGrupo -n puntoDosAKS -c "kubectl apply -f deployment.yaml -n default" --file deployment.yaml |
| --- |

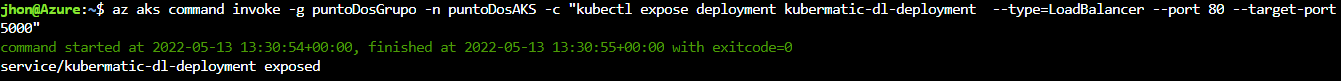
<https://docs.microsoft.com/en-us/azure/aks/command-invoke>



1. To expose your deployment to the outside world, you need a service object that will create an externally reachable IP for your container:

| kubectl expose deployment kubermatic-dl-deployment --type=LoadBalancer --port 80 --target-port 5000 |
| --- |

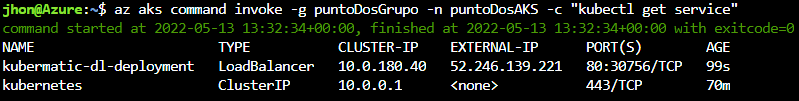
| az aks command invoke -g puntoDosGrupo -n puntoDosAKS -c "kubectl expose deployment kubermatic-dl-deployment --type=LoadBalancer --port 80 --target-port 5000" |
| --- |



1. You're almost there! Check your services to determine the status of your deployment and get the IP address to call your image recognition API:

| kubectl get service |
| --- |

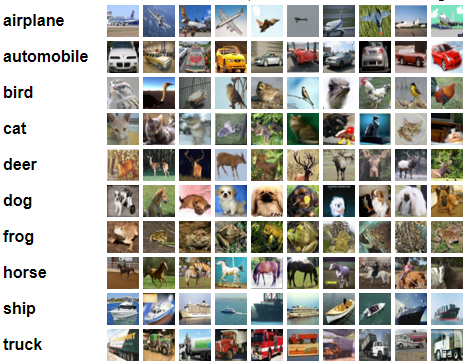
| az aks command invoke -g puntoDosGrupo -n puntoDosAKS -c "kubectl get service" |
| --- |



Ip publica: 52.246.139.221

1. Test your API with these two images using the external IP

Here are the classes in the dataset



| curl -X POST -F img=@airplane\_1.jpg http://52.246.139.221/predict |
| --- |

## Borrar el servicio

Assuming you have a deployment called hello-world, and do a kubectl expose as follows:

| kubectl expose deployment hello-world --type=ClusterIP --name=my-service |
| --- |

his will create a service called my-service, which makes your deployment accessible for debugging, as you described.

To display information about the Service:

| kubectl get services my-service |
| --- |

To delete this service when you are done debugging:

| kubectl delete service my-service |
| --- |

| az aks command invoke -g puntoDosGrupo -n puntoDosAKS -c "kubectl delete service kubermatic-dl-deployment" |
| --- |

## Testing

Creamos una máquina virtual local

