PASO A PASO MICROPROYECTO 2

PUNTO 1:

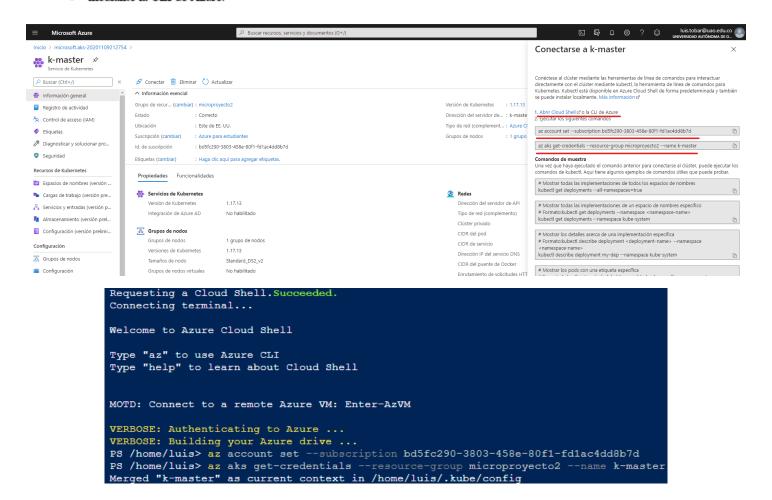
Para este Micro proyecto implementaremos un cluster de Kubernetes en Azure, desplegaremos algunas aplicaciones y exploraremos algunos servicios disponibles.

1. Implementación de cluster Kubernetes en Azure

En este punto realizara la implementación de un clúster de Azure Kubernetes Service (AKS) de al menos dos nodos, mediante Azure Portal. Para esto deberá crear una cuenta para estudiantes en Azure.

Compruebe el funcionamiento de su cluster de dos formas:

- mediante Cloud Shell
- mediante la CLI de Azure.



Para probar desde máquina local con Azure CLI:

Se instala el CLI de Azure:

```
PS C:\Users\Lenovo> Invoke-WebRequest -Uri https://aka.ms/installazurecliwindows -OutFile .\AzureCLI.msi; Start-Process
msiexec.exe -Wait -ArgumentList '/I AzureCLI.msi /quiet'; rm .\AzureCLI.msi
```

Se actualiza:

Se hace login:

A correr az login se va abrir el navegador y pide ingresar a la cuenta azure, si se hace por Ubuntu(máquina virtual) va a dar una url y un clave para copiarlas en el navegador, y luego logearse igual.

Ahora se conecta a AKS:

```
PS C:\Users\Lenovo> az account set --subscription bd5fc290-3803-458e-80f1-fd1ac4dd8b7d
PS C:\Users\Lenovo> az aks get-credentials --resource-group microproyecto2 --name k-master
```

Para subar kubernetes toca instalarlo con choco, pero primero se instala choco (se verifica la versión de PoweShell):

@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

```
Administration Simbolo del sistems

**Aicrosoft Mindows [Versión 10.0.18362.720]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

**Li Windows ksystem 22/9" **Ssystem Nott Noystem 22 Windows Power Shell V1.0 Nower Shell. exe" - NoProfile - Input Format None - Execution Policy Bypass - Command 'iex (New-Object System. Net. NebClient). DownloadString('https://chocolatey.org/install.psi'))"

**Setting Latest version of the Chocolatey package for download.

**Setting Chocolatey from https://chocolatey.org/api/V2/package/chocolatey/0.10.15.

**Downloading 7-21p commandline tool prior to extraction.

**Extracting Chocolatey from https://chocolatey.org/api/V2/package/chocolatey/0.10.15.

**Downloading 7-21p commandline tool prior to extraction.

**Extracting Chocolatey from https://chocolatey.org/api/V2/package/chocolatey.2ip to C:\Users\Lenovo\AppData\Local\Tem
2\textracting Chocolatey from https://chocolatey.org/api/V2/package/chocolatey.2ip to C:\Users\Lenovo\AppData\Local\Tem
2\textracting Chocolatey.org.atall to 'C:\ProgramData\chocolatey\Chocolatey\Chocolatey.2ip to C:\Users\Lenovo\AppData\Local\Tem
2\textracting Chocolatey.org.atall to 'C:\ProgramData\chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Chocolatey\Cho
```

```
PS C:\Windows\system32> choco install kubernetes-cli
Chocolatey v0.10.15
Installing the following packages:
cubernetes-cli
By installing you accept licenses for the packages.
cubernetes-cli v1.19.3 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
The package kubernetes-cli wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): Y
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar.gz to C:\Pro
gramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar to C:\Progra
mData\chocolatey\lib\kubernetes-cli\tools..
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
ShimGen has successfully created a shim for kubectl.exe
 The install of kubernetes-cli was successful.
 Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-cli\tools'
Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32>
```

```
PS C:\Users\Lenovo> az login
You have logged in. Now let us find all the subscriptions to which you have access...
   "cloudName": "AzureCloud",
"homeTenantId": "693cbea0-4ef9-4254-8977-76e05cb5f556",
"id": "bd5fc290-3803-458e-80f1-fd1ac4dd8b7d",
   "isDefault": true,
"managedByTenants": [],
"name": "Azure para estudiantes",
    "state": "Enabled",
    "tenantId": "693cbea0-4ef9-4254-8977-76e05cb5f556",
    "user": {
   "name": "luis.tobar@uao.edu.co",
      "type": "user
PS C:\Users\Lenovo> kubectl get services
                      NAME
                                                                                                                         EXTERNAL-IP
                                                                                                       CLUSTER-IP
                                                                                                                                           PORT(S)
NAMESPACE
                                                                                      TYPE
default
                                                                                      ClusterIP
                                                                                                       10.0.0.1
                                                                                                                                           443/TCP
                      kubernetes
                                                                                                                         <none>
                                                                                                       10.0.130.78
                                                                                                                                           80:31794/TCP
example
                      image-classifier
                                                                                      NodePort
                                                                                                                         <none>
                                                                                                                                                                             69m
69m
                      gatekeeper-webhook-service
                                                                                      ClusterIP
                                                                                                       10.0.248.92
                                                                                                                                           443/TCP
gatekeeper-system
                                                                                                                         <none>
ube-system
                       addon-http-application-routing-default-http-backend
                                                                                     ClusterIP
                                                                                                       10.0.172.146
                                                                                                                                           80/TCP
                                                                                                                                           80:30124/TCP,443:31106/TCP
                                                                                                                                                                             69m
69m
69m
                                                                                                                         40.76.150.116
kube-system
                      addon-http-application-routing-nginx-ingress
                                                                                      LoadBalancer
                                                                                                       10.0.117.231
                                                                                                       10.0.29.54
10.0.125.145
kube-system
                      azure-policy-webhook-service
                                                                                     ClusterIP
                                                                                                                         <none>
                                                                                                                                           443/TCP
                                                                                                                                           5473/TCP
                      calico-typha
                                                                                      ClusterIP
kube-system
                                                                                                                         <none>
                                                                                                                                                                             69m
kube-system
                      dashboard-metrics-scraper
                                                                                      ClusterIP
                                                                                                       10.0.22.247
                                                                                                                                           8000/TCP
                                                                                                                         <none>
                      healthmodel-replicaset-service
                                                                                      ClusterIP
                                                                                                       10.0.10.60
                                                                                                                                           25227/TCP
                                                                                                                                                                              69m
ube-system
                                                                                                                                                                             69m
69m
                                                                                                                                           53/UDP,53/TCP
443/TCP
                                                                                      ClusterIP
                                                                                                       10.0.0.10
 ube-system
                      kube-dns
                                                                                                                         <none>
                                                                                                       10.0.150.79
ube-system
                      kubernetes-dashboard
                                                                                     ClusterIP
                                                                                                                         <none>
ube-system
S C:\Users\Lenovo>
                      metrics-server
                                                                                     ClusterIP
                                                                                                       10.0.2.134
                                                                                                                         <none>
                                                                                                                                           443/TCP
```

PUNTO 2:

2. Aplicación de clasificación de imágenes en Azure

Despliegue la aplicación "image classifier" en AKS. Compruebe su funcionamiento.

PRIMERA FORMA DESDE LA INTERFAZ WEB DE AZURE Y CLOUD SHELL:

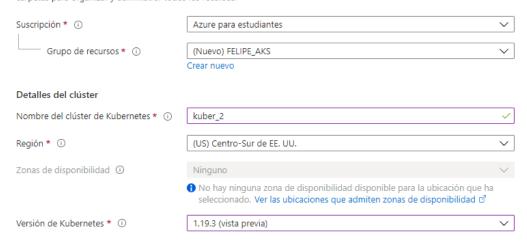
Inicio > Servicios de Kubernetes >

Crear un clúster de Kubernetes

impiementación y la administración de aplicaciones en contenedores sin necesidad de experiencia relativa, lambien 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.



Grupo de nodos principal

Número y tamaño de los nodos del grupo de nodos primarios del clúster. Para las cargas de trabajo de producción, se recomienda un mínimo de 3 nodos para obtener resistencia. Para las cargas de trabajo de desarrollo o prueba, solo se requiere un nodo. Si quiere agregar grupos de nodos adicionales o ver opciones de configuración adicionales para este grupo de nodos, vaya a la pestaña "Grupos de nodos" anterior. Después de crear el clúster, podrá agregar grupos de nodos adicionales. Más información sobre los grupos de nodos en Azure Kubernetes Service



Datos básicos	Grupos de nodos	Autenticación	Redes	Integraciones	Etiquetas	Revisar y crear
Grupos de nodo	os					
de nodos opciona	o de nodos principal ne ales para controlar una acerca de varios grupos	gran variedad de c			sicos, también	puede agregar grupos
+ Agregar gru	upo de nodos 🗓 Elin	ninar				
Nombre	Tipo de S	50 N	úmero de no	odos Tamai	ño del nodo	Zonas de disponik
agentpool ((principal) Linux	2		Stand	ard_B2s	Ninguno
← Habilitar nodos	virtuales)
Los nodos virtual	es permiten el escalado sobre los nodos virtuale	•	por Azure	Container Instance	es sin servidor.	
Habilitar nodos vi	irtuales 🛈					
Habilitar conjur	ntos de escalado de m	náquinas virtuales	5			
máquinas virtuale escenarios que in	njuntos de escalado de es individuales para los icluyen escalado autom sobre los conjuntos de	nodos del clúster. L ático, varios grupos	os conjunto de nodos y	s de escalado de i compatibilidad c	máquinas virtu	
Habilitar conjunto máquinas virtuale	os de escalado de es ①					
Datos básicos	Grupos de nodos	Autenticación	Redes I	ntegraciones	Etiquetas	Revisar y crear
	e clúster ervice usa la autenticac clúster. Puede ser una					
Método de autentio	cación	Entidad de se	ervicio 🔘	Identidad admini	strada asignad	a por el sistema
Entidad de servicio	* (i)	Entidad de servi				
		Configurar la entid	dad de servi	cio		
El clúster de Kuberr	ticación y autorizació netes usa la autenticació er una vez autenticado.	n y la autorización	•			er, así como lo que el
Control de acceso b	pasado en roles (RBAC)	Habilitado (Deshabi	litado		
Azure Active Director AKS ①	ory administrado con	Habilitado (Deshabi	litado		
De forma predetern obtener más contro	de SO del grupo de n ninada, todos los discos ol sobre el cifrado, pued t. El conjunto de cifrado ación d	de AKS están cifra e proporcionar sus	propias clav	es con un conjun	to de cifrado d	e disco respaldado
Tipo de cifrado		(Predeterminado	o) Cifrado en	reposo con una	clave administr	ada por la plat 🗸

Datos básicos Redes Grupos de nodos Autenticación Integraciones Etiquetas Revisar v crear Puede cambiar la configuración de red del clúster, por ejemplo, habilitar el enrutamiento de aplicaciones HTTP y configurar la red mediante las opciones "Kubenet" o "Azure CNI": El complemento de redes kubenet crea una red virtual para el clúster con los valores predeterminados. El complemento de redes de Azure CNI permite a los clústeres usar una red virtual nueva o existente con direcciones que se pueden personalizar. Los pods de aplicación se conectan directamente a la red virtual, lo que permite la integración nativa con las características de redes virtuales. Más información sobre las redes en Azure Kubernetes Service Kubenet Conexión de red (i) Azure CNI Prefijo de nombre DNS * ① kuber-dns Enrutamiento de tráfico Standard Equilibrador de carga (i) Habilitar enrutamiento de solicitudes HTTP 🗹 Seguridad Habilitar clúster privado ① Establecer intervalos IP autorizados (i) Ninguno Directiva de red (i) Calico Azure 1 La directiva de red de Azure no es compatible con las redes de kubenet. Integraciones Etiquetas Datos básicos Grupos de nodos Autenticación Redes Revisar y crear Conecte el clúster de AKS con servicios adicionales. Azure Container Registry Conecte el clúster a una instancia de Azure Container Registry para permitir que las implementaciones se realicen sin problemas desde un registro de imagen privada. Puede crear un registro nuevo o elegir uno existente. Más información sobre Azure Container Registry de Más información sobre Azure Container Registry Registro de contenedor Ninguno 🌓 Se debe usar el método de autenticación de identidad administrada asignada por el sistema para asociar una instancia de Azure Container Registry. Azure Monitor Además de las métricas de la CPU y la memoria incluidas en AKS de forma predeterminada, puede habilitar Container Insights para obtener datos más completos sobre el rendimiento y el estado globales del clúster. La facturación se basa en las opciones de configuración de la ingesta y la retención de datos. Más información acerca de la supervisión del rendimiento y del estado de contenedores Obtenga más información sobre precios Supervisión de contenedores Área de trabajo de Log Analytics 🛈 (Nuevo) DefaultWorkspace-bd5fc290-3803-458e-80f1-fd1ac4dd8b7d-SCUS Crear Azure Policy

Aplique cumplimientos y medidas de seguridad a escala para los clústeres de AKS de forma centralizada y coherente mediante Azure Policy. Más información sobre Azure Policy para AKS ♂

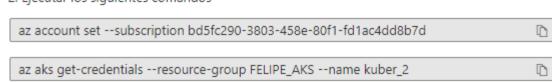
Azure Policy

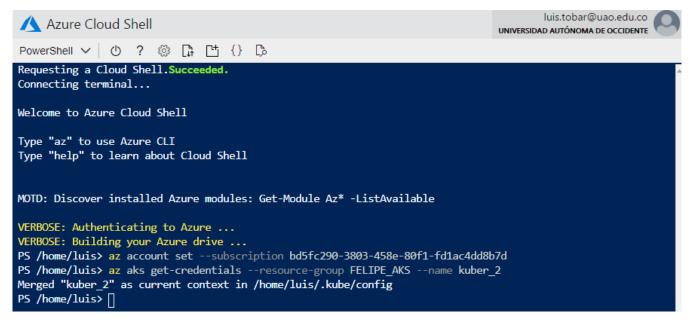
Conectarse a kuber_2

X

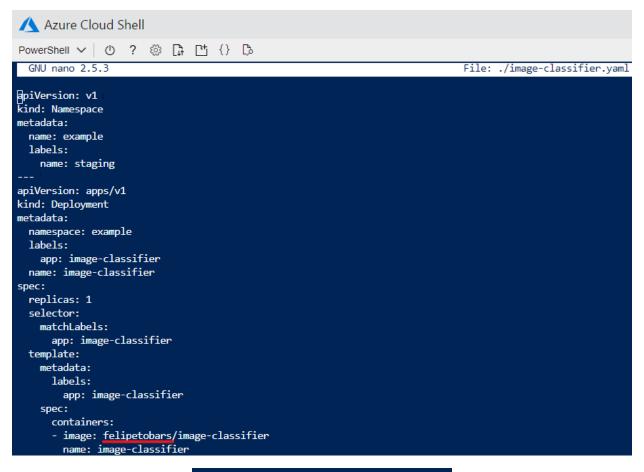
Conéctese al clúster mediante las herramientas de línea de comandos para interactuar directamente con el clúster mediante kubectl, la herramienta de línea de comandos para Kubernetes. Kubectl está disponible en Azure Cloud Shell de forma predeterminada y también se puede instalar localmente. Más información 🗹

- 1. Abrir Cloud Shell do la CLI de Azure
- 2. Ejecutar los siguientes comandos

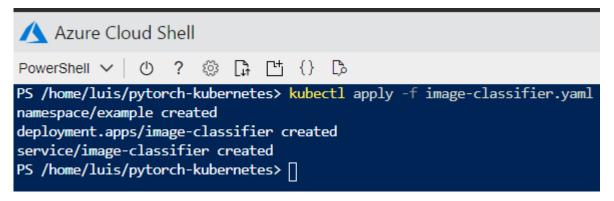




```
PS /home/luis> git clone https://github.com/omondragon/pytorch-kubernetes
Cloning into 'pytorch-kubernetes'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
PS /home/luis> cd ./pytorch-kubernetes/
PS /home/luis/pytorch-kubernetes> nano
clouddrive/ pytorch-kubernetes> ls
image-classifier.yaml
PS /home/luis/pytorch-kubernetes> [
```



```
apiVersion: v1
kind: Service
metadata:
   namespace: example
   labels:
    app: image-classifier
   name: image-classifier
spec:
   ports:
   - port: 80
     protocol: TCP
     targetPort: 5000
   selector:
    app: image-classifier
type: LoadBalancer
```



NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
example	image-classifier-866757868b-k8m88	0/1	ContainerCreating	0	27s
gatekeeper-system	gatekeeper-audit-67cd687775-6g4nt	1/1	Running	0	13m
gatekeeper-system	gatekeeper-controller-65bcdc6f5b-qvxll	1/1	Running	1	13m
gatekeeper-system	gatekeeper-controller-65bcdc6f5b-tdzwx	1/1	Running	0	13m
kube-system	addon-http-application-routing-default-http-backend-6d9d88sm4w5	1/1	Running	0	13m
kube-system	addon-http-application-routing-external-dns-76874f85dc-cc4dj	1/1	Running	0	13m
kube-system	addon-http-application-routing-nginx-ingress-controller-c8m4tqm	1/1	Running	0	13m
kube-system	azure-policy-648b5fc6d8-cqbcc	1/1	Running	0	13m
kube-system	azure-policy-webhook-55bc4976f-wjpbg	1/1	Running	0	13m
kube-system	coredns-85dbd7c745-8cbp2	1/1	Running	0	9m57s
kube-system	coredns-85dbd7c745-vs7qp	1/1	Running	0	13m
kube-system	coredns-autoscaler-68f5df579d-jdkht	1/1	Running	0	13m
kube-system	kube-proxy-74zld	1/1	Running	0	11m
kube-system	kube-proxy-hbvzv	1/1	Running	0	9m44s
kube-system	metrics-server-777689c987-crzph	1/1	Running	0	13m
kube-system	omsagent-g7xfm	1/1	Running	0	9m44s
kube-system	omsagent-rs-866dfcc6f9-6qt8t	1/1	Running	0	13m
kube-system	omsagent-xcmzw	1/1	Running	1	11m
kube-system	tunnelfront-7cdcf65949-189bm	1/1	Running	0	12m
PS /home/luis/pyton	rch-kubernetes> [

PS /home/luis/pytor	ch-kubernetes> kubectl get podsall-namespaces				
NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
example	image-classifier-866757868b-k8m88	1/1	Running	0	6m55s
gatekeeper-system	gatekeeper-audit-67cd687775-6g4nt	1/1	Running	0	19m
gatekeeper-system	gatekeeper-controller-65bcdc6f5b-qvxll	1/1	Running	1	19m
gatekeeper-system	gatekeeper-controller-65bcdc6f5b-tdzwx	1/1	Running	0	19m
kube-system	addon-http-application-routing-default-http-backend-6d9d88sm4w5	1/1	Running	0	19m
kube-system	addon-http-application-routing-external-dns-76874f85dc-cc4dj	1/1	Running	0	19m
kube-system	addon-http-application-routing-nginx-ingress-controller-c8m4tqm	1/1	Running	0	19m
kube-system	azure-policy-648b5fc6d8-cqbcc	1/1	Running	0	19m
kube-system	azure-policy-webhook-55bc4976f-wjpbg	1/1	Running	0	19m
kube-system	coredns-85dbd7c745-8cbp2	1/1	Running	0	16m
kube-system	coredns-85dbd7c745-vs7qp	1/1	Running	0	19m
kube-system	coredns-autoscaler-68f5df579d-jdkht	1/1	Running	0	19m
kube-system	kube-proxy-74zld	1/1	Running	0	17m
kube-system	kube-proxy-hbvzv	1/1	Running	0	16m
kube-system	metrics-server-777689c987-crzph	1/1	Running	0	19m
kube-system	omsagent-g7xfm	1/1	Running	0	16m
kube-system	omsagent-rs-866dfcc6f9-6qt8t	1/1	Running	0	19m
kube-system	omsagent-xcmzw	1/1	Running	1	17m
kube-system	tunnelfront-7cdcf65949-189bm	1/1	Running	0	19m
PS /home/luis/pytor	rch-kubernetes> [

PS /home/luis/pytor	rch-kubernetes> kubectl get servicesall-namespaces					
NAMESPACE	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
default	kubernetes	ClusterIP	10.0.0.1	<none></none>	443/TCP	20m
example	image-classifier	LoadBalancer	10.0.125.206	40.124.232.234	80:31222/TCP	7m26s
gatekeeper-system	gatekeeper-webhook-service	ClusterIP	10.0.112.200	<none></none>	443/TCP	20m
kube-system	addon-http-application-routing-default-http-backend	ClusterIP	10.0.100.237	<none></none>	80/TCP	20m
kube-system	addon-http-application-routing-nginx-ingress	LoadBalancer	10.0.220.131	20.64.25.231	80:31764/TCP,443:31230/TCP	20m
kube-system	azure-policy-webhook-service	ClusterIP	10.0.79.66	<none></none>	443/TCP	20m
kube-system	healthmodel-replicaset-service	ClusterIP	10.0.67.114	<none></none>	25227/TCP	20m
kube-system	kube-dns	ClusterIP	10.0.0.10	<none></none>	53/UDP,53/TCP	20m
kube-system	metrics-server	ClusterIP	10.0.77.48	<none></none>	443/TCP	20m

SEGUNDA FORMA DESDE CLI DE AZURE EN MÁQUINA LINUX:

https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-apt

```
ragrant@maquina1:~$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [368 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [652 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [163 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-updates/restricted i386 Packages [12.0 kB]
vagrant@maquina1:~$ sudo apt-get install ca-certificates curl apt-transport<u>-https lsb-release gnupg</u>
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2).
lsb-release is already the newest version (11.1.0ubuntu2).
lsb-release set to manually installed.
The following additional packages will be installed:
 libcurl4
The following NEW packages will be installed:
 apt-transport-https
The following packages will be upgraded:
 ca-certificates curl libcurl4
3 upgraded, 1 newly installed, 0 to remove and 129 not upgraded.
Need to get 549 kB of archives.
After this operation, 176 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
vagrant@maquina1:~$ curl -sL https://packages.microsoft.com/keys/microsoft.asc
> gpg --dearmor |
> sudo tee /etc/apt/trusted.gpg.d/microsoft.gpg > /dev/null
```

```
uina1:~$ AZ_REPO=$(lsb_release -cs)
echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ $AZ_REPO main" |
sudo tee /etc/apt/sources.list.d/azure-cli.listvagrant@maquina1:~$ echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ $AZ_REPO main" |
       sudo tee /etc/apt/sources.list.d/azure-cli.list
deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ focal main
 /agrant@maquina1:~$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://packages.microsoft.com/repos/azure-cli focal InRelease [10.4 kB]
Get:6 https://packages.microsoft.com/repos/azure-cli focal/main amd64 Packages [3,162 B]
Fetched 13.6 kB in 15s (915 B/s)
Reading package lists... Done
vagrant@maquinal:~$ sudo apt-get install azure-cli
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 azure-cli
0 upgraded, 1 newly installed, 0 to remove and 129 not upgraded.
Need to get 51.4 MB of archives.
After this operation, 615 MB of additional disk space will be used.
Get:1 https://packages.microsoft.com/repos/azure-cli focal/main amd64 azure-cli all 2.14.2-1~focal [51.4 MB] Fetched 51.4 MB in 21s (2,412 kB/s)
Selecting previously unselected package azure-cli.
(Reading database ... 74648 files and directories currently installed.)
Preparing to unpack .../azure-cli_2.14.2-1~focal_all.deb ...
Unpacking azure-cli (2.14.2-1~focal) ...
Setting up azure-cli (2.14.2-1~focal) ...
 /agrant@maquina1:∼$ .
```

vagrant@maquina1:~\$ az login To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code RL2FJKFA9 to auth enticate.



Especificar el código

Escriba el código que se muestra en su aplicación o dispositivo.

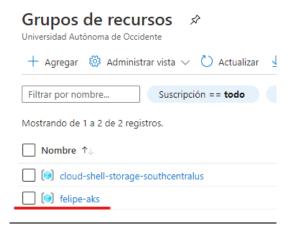
RL2FJKFA9

Siguiente

https://www.youtube.com/watch?v=TzjWqBD5Vng&ab channel=TechSnips

```
vagrant@maquina1:~$ az provider register -n Microsoft.ContainerService
vagrant@maquina1:~$ az provider register -n Microsoft.Compute
vagrant@maquina1:~$ az provider register -n Microsoft.Network
vagrant@maquina1:~$ az provider register -n Microsoft.Storage
vagrant@maquina1:~$
```

```
vagrant@maquina1:~$ az group create -n felipe-aks --location eastus
{
    "id": "/subscriptions/bd5fc290-3803-458e-80f1-fd1ac4dd8b7d/resourceGroups/felipe-aks",
    "location": "eastus",
    "managedBy": null,
    "name": "felipe-aks",
    "properties": {
        "provisioningState": "Succeeded"
    },
    "tags": null,
    "type": "Microsoft.Resources/resourceGroups"
}
vagrant@maquina1:~$
```



az aks create --resource-group felipe-aks --name proyecto-cn --node-count 2 --generate-ssh-keys --kubernetes-version 1.19.3 --node-vm-size=Standard_B2s

```
"count": 1
    "outboundIpPrefixes": null,
     "outboundIps": null
  },
"loadBalancerSku": "Standard",
  "networkMode": null,
  "networkPlugin": "kubenet",
  "networkPolicy": null,
  "outboundType": "loadBalancer",
  "podCidr": "10.244.0.0/16",
  "serviceCidr": "10.0.0.0/16"
 "nodeResourceGroup": "MC felipe-aks proyecto-cn eastus",
"powerState": {
  "code": "Running"
 privateFqdn": null,
"provisioningState": "Succeeded",
"resourceGroup": "felipe-aks",
"servicePrincipalProfile": {
  "clientId": "cc8a2a5f-f9e2-4879-8943-41d894a22ac0",
  "secret": null
 sku": {
   "name": "Basic",
  "tier": "Free"
 "tags": null,
 "type": "Microsoft.ContainerService/ManagedClusters",
 "windowsProfile": null
agrant@maquina1:∼$
```

Universidad Autónoma de Occidente 🕂 Agregar 🗸 🥨 Administrar vista 🗸 🖰 Actualizar 🔟 Exportar a CSV 😽 Abrir consulta Filtrar por nombre... Suscripción == todo Tipo == todo Grupo de recursos = Mostrando de 1 a 1 de 1 registros. Nombre ↑↓ Tipo ↑↓ Grupo de recursos ↑↓ proyecto-cn Servicio de Kubernetes felipe-aks Para poder apagar y encender al AKS: https://docs.microsoft.com/en-gb/azure/aks/start-stop-cluster#code-try-4 vagrant@maquina1:~\$ az extension add --name aks-preview The installed extension 'aks-preview' is in preview. /agrant@maquina1:~\$ az extension update --name aks-preview No updates available for 'aks-preview'. Use --debug for more information. agrant@maquina1:∼\$ az feature register --namespace "Microsoft.ContainerService" --name "StartStopPreview" Once the feature 'StartStopPreview' is registered, invoking 'az provider register -n Microsoft.ContainerService' is required to get the change propagated "id": "/subscriptions/bd5fc290-3803-458e-80f1-fd1ac4dd8b7d/providers/Microsoft.Features/providers/Microsoft.Contai nerService/features/StartStopPreview", "name": "Microsoft.ContainerService/StartStopPreview", 'properties": { "state": "Registered" type": "Microsoft.Features/providers/features" uinal:~\$ az feature list -o table --query "[?contains(name, 'Microsoft.ContainerService/StartStopPreview)].{Name:name,State:properties.state}" State Microsoft.ContainerService/StartStopPreview Registered vagrant@maquina1:~\$ az provider register --namespace Microsoft.ContainerService agrant@maquina1:~\$ az aks stop --name proyecto-cn --resource-group felipe-aks Running ... agrant@maquina1:~\$ az aks show --name proyecto-cn --resource-group felipe-aks

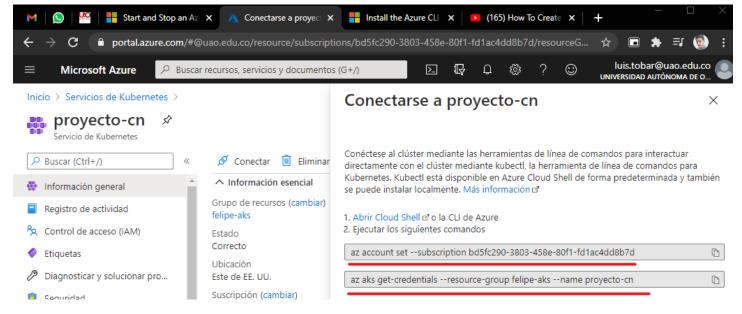
Servicios de Kubernetes 🛷

```
},
"nodeResourceGroup": "MC_felipe-aks_proyecto-cn_eastus",
"powerState": {
    "code": "Stopped"
},
    "privateFqdn": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "felipe-aks",
    "servicePrincipalProfile": {
        "clientId": "cc8a2a5f-f9e2-4879-8943-41d894a22ac0"
},
    "sku": {
        "name": "Basic",
        "tier": "Free"
},
    "type": "Microsoft.ContainerService/ManagedClusters",
    "windowsProfile": null
}
```

vagrant@maquina1:~\$ az aks start --name proyecto-cn --resource-group felipe-aks _- Running ..

vagrant@maquina1:~\$ az aks show --name proyecto-cn --resource-group felipe-aks

```
},
"nodeResourceGroup": "MC_felipe-aks_proyecto-cn_eastus",
"powerState": {
    "code": "Running"
},
"privateFqdn": null,
"provisioningState": "Succeeded",
"resourceGroup": "felipe-aks",
"servicePrincipalProfile": {
    "clientId": "cc8a2a5f-f9e2-4879-8943-41d894a22ac0"
},
"sku": {
    "name": "Basic",
    "tier": "Free"
},
"type": "Microsoft.ContainerService/ManagedClusters",
"windowsProfile": null
}
```



Para no tener que copiar y pegar las credenciales se usa:

```
/agrant@maquina1:~$ kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
/agrant@maquina1:~$ az aks get-credentials --resource-group felipe-aks --name proyecto-cn
The behavior of this command has been altered by the following extension: aks-preview
Merged "proyecto-cn" as current context in /home/vagrant/.kube/config
vagrant@maquina1:~$ kubectl get nodes
NAME
                                    STATUS
                                             ROLES
                                                      AGE
                                                            VERSION
aks-nodepool1-26759458-vmss000002
                                    Ready
                                                      12m
                                                            v1.19.3
                                              agent
aks-nodepool1-26759458-vmss000003
                                                            v1.19.3
                                                      12m
                                    Ready
                                              agent
/agrant@maquina1:∼$
```

```
vagrant@maquina1:~$ git clone https://github.com/omondragon/pytorch-kubernetes
Cloning into 'pytorch-kubernetes'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 473 bytes | 43.00 KiB/s, done.
vagrant@maquina1:~$ cd pytorch-kubernetes/
vagrant@maquina1:~/pytorch-kubernetes$
```

vagrant@maquina1:~/pytorch-kubernetes\$ nano image-classifier.yaml

```
apiVersion: v1
kind: Service
metadata:
  namespace: example
  labels:
    app: image-classifier
  name: image-classifier
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 5000
selector:
    app: image-classifier
type: LoadBalancer
```

```
vagrant@maquina1:~/pytorch-kubernetes$ kubectl apply -f image-classifier.yaml
namespace/example created
deployment.apps/image-classifier created
service/image-classifier created
```

vagrant@maqui	<pre>ina1:~/pytorch-kubernetes\$ kubectl get</pre>	pods	all-namespaces		
NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
example	image-classifier-866757868b-gktdr	0/1	ContainerCreating	0	67s
kube-system	coredns-85dbd7c745-2pxpg	1/1	Running	0	17m
kube-system	coredns-85dbd7c745-1svb9	1/1	Running	0	17m
kube-system	coredns-autoscaler-68f5df579d-svktl	1/1	Running	0	17m
kube-system	kube-proxy-4vzj8	1/1	Running	0	16m
kube-system	kube-proxy-fh81n	1/1	Running	0	16m
kube-system	metrics-server-777689c987-9k9m7	1/1	Running	0	17m
kube-system	tunnelfront-d98d8ddb-7w7qt	1/1	Running	0	17m

```
vagrant@maquina1:~/pytorch-kubernetes$ kubectl get pods --all-namespaces
NAMESPACE
              NAME
                                                               STATUS
                                                                                     AGE
                                                       READY
                                                                          RESTARTS
                                                       1/1
                                                                                     7m3s
example
              image-classifier-866757868b-gktdr
                                                               Running
                                                                          0
                                                      1/1
kube-system
              coredns-85dbd7c745-2pxpg
                                                               Running
                                                                          0
                                                                                     23m
kube-system
              coredns-85dbd7c745-1svb9
                                                       1/1
                                                               Running
                                                                          0
                                                                                     23m
kube-system
              coredns-autoscaler-68f5df579d-svktl
                                                       1/1
                                                               Running
                                                                          0
                                                                                     23m
kube-system
              kube-proxy-4vzi8
                                                       1/1
                                                               Running
                                                                          0
                                                                                     22m
                                                                         0
                                                                                     22m
kube-system
              kube-proxy-fh81n
                                                       1/1
                                                               Running
              metrics-server-777689c987-9k9m7
                                                      1/1
                                                               Running
                                                                          0
                                                                                     23m
kube-system
              tunnelfront-d98d8ddb-7w7qt
                                                                                     23m
                                                       1/1
                                                               Running
                                                                          0
kube-system
vagrant@maquina1:~/pytorch-kubernetes$
```

```
tes$ kubectl get services --all-namespaces
NAMESPACE
              NAME
                                  TYPE
                                                  CLUSTER-IP
                                                                  EXTERNAL-IP
                                                                                 PORT(S)
                                                                                                  AGE
default
              kubernetes
                                  ClusterIP
                                                  10.0.0.1
                                                                                 443/TCP
                                                                                                  50m
                                                                  <none>
              image-classifier
                                  LoadBalancer
                                                  10.0.191.133
                                                                  20.62.191.63
                                                                                 80:30707/TCP
                                                                                                  7m27s
example
                                                                                 53/UDP,53/TCP
                                  ClusterIP
              kube-dns
                                                  10.0.0.10
                                                                  <none>
                                                                                                  49m
kube-system
                                  ClusterIP
                                                  10.0.35.98
                                                                                 443/TCP
                                                                                                  49m
kube-system
              metrics-server
                                                                  <none>
ragrant@maquina1:~/pytorch-kubernetes$
```

```
vagrant@maquina1:~/pytorch-kubernetes$ curl -X POST -d '{"url": "https://i.imgur.com/jD2hDMc.jpg"}' -H 'Content-Type
: application/json' http://20.62.191.63/predict
[["Persian_cat", 80.17660522460938]]
vagrant@maquina1:~/pytorch-kubernetes$
```

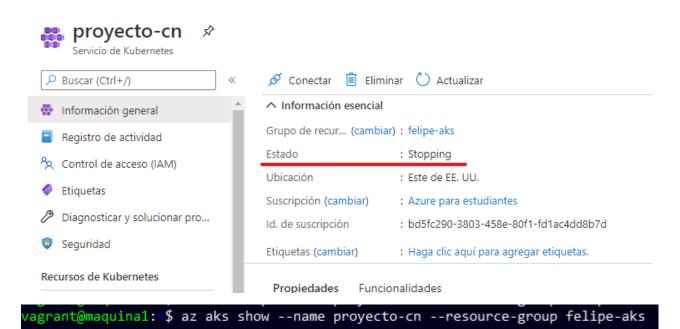
curl -X POST -d '{"url": "https://i.imgur.com/jD2hDMc.jpg"}' -H 'Content-Type: application/json' http://52.191.18.213/predict

Apagar todo y dejar bonito si consumir créditos:

vagrant@maquina1:~/pytorch-kubernetes\$ kubectl delete service,deployment image-classifier -n example
service "image-classifier" deleted
deployment.apps "image-classifier" deleted

```
vagrant@maquina1:~/pytorch-kubernetes$ kubectl get pods -n example
No resources found in example namespace.
vagrant@maquina1:~/pytorch-kubernetes$
```

```
vagrant@maquina1:~/pytorch-kubernetes$ cd ...
vagrant@maquina1:~$ kubectl get nodes
                                             ROLES
                                                      AGE
                                                            VERSION
NAME
                                    STATUS
aks-nodepool1-26759458-vmss000002
                                                      28m
                                                            v1.19.3
                                    Ready
                                              agent
aks-nodepool1-26759458-vmss000003
                                    Ready
                                                      28m
                                                            v1.19.3
                                              agent
vagrant@maquina1:~$ az aks stop --name proyecto-cn --resource-group felipe-aks
```



```
"nodeResourceGroup": "MC_felipe-aks_proyecto-cn_eastus",
"powerState": {
    "code": "Stopped"
},
    "privateFqdn": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "felipe-aks",
    "servicePrincipalProfile": {
        "clientId": "cc8a2a5f-f9e2-4879-8943-41d894a22ac0"
},
    "sku": {
        "name": "Basic",
        "tier": "Free"
},
    "type": "Microsoft.ContainerService/ManagedClusters",
    "windowsProfile": null
}
```

PUNTO 3:

3. Aplicación de su interés en Azure

Despliegue una aplicación de su interés en AKS. Compruebe su funcionamiento.

https://kubernetes.io/docs/tutorials/stateless-application/guestbook/

PRIMERO SE HIZO LA PRUEBA DE LA APLICACIÓN CON LAS MÁQUINAS VIRTUALES QUE SE TIENEN CON VAGRANT:

```
vagrant@kmaster:~$ mkdir application
vagrant@kmaster:~$ cd application/
vagrant@kmaster:~/application$ mkdir guestbook
vagrant@kmaster:~/application$
vagrant@kmaster:~/application$ cd guestbook/
```

vagrant@kmaster:~/application/guestbook\$ nano redis-master-deployment.yaml

```
💹 vagrant@kmaster: ~/application/guestbook
 GNU nano 4.8
                                            redis-master-deployment.yaml
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
 name: redis-master
  labels:
    app: redis
spec:
 selector:
    matchLabels:
      app: redis
      role: master
      tier: backend
  replicas: 1
  template:
   metadata:
      labels:
        app: redis
        role: master
        tier: backend
    spec:
      containers:
      - name: master
        image: k8s.gcr.io/redis:e2e # or just image: redis
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        ports:
        - containerPort: 6379
```

```
vagrant@kmaster:~/application$ tree
L guestbook
L redis-master-deployment.yaml
1 directory, 1 file
```

vagrant@kmaster:~/application/guestbook\$ nano redis-master-service.yaml

```
💹 vagrant@kmaster: ~/application/guestbook
 GNU nano 4.8
apiVersion: v1
kind: Service
metadata:
 name: redis-master
 labels:
   app: redis
   role: master
   tier: backend
spec:
 ports:
  name: redis
   port: 6379
   targetPort: 6379
 selector:
   app: redis
    role: master
    tier: backend
    agrant@kmaster:~/application$ tree

    redis-master-deployment.yaml

    redis-master-service.yaml

    directory, 2 files
```

```
ion/guestbook$ kubectl get_services
/agrant@kmaster:∼/applicat
NAME
                            CLUSTER-IP
               TYPE
                                             EXTERNAL-IP
                                                           PORT(S)
                                                                       AGE
               ClusterIP
                            10.96.0.1
                                                            443/TCP
                                                                       3d11h
kubernetes
                                             <none>
               ClusterIP
                            10.104.67.158
                                                                       2m31s
redis-master
                                             <none>
                                                            6379/TCP
```

vagrant@kmaster:~/application/guestbook\$ nano redis-slave-deployment.yaml

```
💹 vagrant@kmaster: ~/application/guestbook
 GNU nano 4.8
                                                                           redis-slave-deployment.yaml
 labels:
    app: redis
spec:
 selector:
   matchLabels:
      app: redis
     role: slave
     tier: backend
 replicas: 2
 template:
   metadata:
      labels:
        app: redis
        role: slave
        tier: backend
    spec:
     containers:
      - name: slave
        image: gcr.io/google samples/gb-redisslave:v3
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        - name: GET HOSTS FROM
          value: dns
          # access an environment variable to find the master
        ports:
        - containerPort: 6379
```

vagrant@kmaster:~/application/guestbook\$ kubectl apply -f https://k8s.io/examples/application/guestbook/redis-slave-deployment.yaml
deployment.apps/redis-slave created

```
/agrant@kmaster:~/applic
                                        $ kubect1 get pods
NAME
                                READY
                                        STATUS
                                                             RESTARTS
                                                                         AGE
redis-master-f46ff57fd-15hr2
                                1/1
                                        Running
                                                             1
                                                                         22m
redis-slave-bbc7f655d-7k9t8
                                0/1
                                        ContainerCreating
                                                             0
                                                                         85
                                0/1
                                        ContainerCreating
redis-slave-bbc7f655d-glkbn
                                                                         8s
```

vagrant@kmaster:~/application/guestbook\$ nano redis-slave-service.yaml

```
💹 vagrant@kmaster: ~/application/guestbook
  GNU nano 4.8
apiVersion: v1
kind: Service
metadata:
  name: redis-slave
  labels:
    app: redis
    role: slave
    tier: backend
spec:
  ports:
  - port: 6379
  selector:
    app: redis
    role: slave
    tier: backend
```

vagrant@kmaster:~/application/guestbook\$ kubectl apply -f https://k8s.io/examples/application/guestbook/redis-slave-service.yaml

```
vagrant@kmaster:~/application/guestbook$ kubectl get services
NAME
               TYPE
                           CLUSTER-IP
                                            EXTERNAL-IP
                                                           PORT(S)
                                                                      AGE
                                                                      3d11h
kubernetes
               ClusterIP
                            10.96.0.1
                                                           443/TCP
                                            <none>
               ClusterIP
redis-master
                           10.104.67.158
                                            <none>
                                                           6379/TCP
                                                                      17m
               ClusterIP
redis-slave
                            10.107.129.44
                                                           6379/TCP
                                                                      20s
                                            <none>
```

vagrant@kmaster:~/application/guestbook\$ nano frontend-deployment.yaml

```
💹 vagrant@kmaster: ~/application/guestbook
 GNU nano 4.8
                                                                             frontend-deployment.yaml
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
 name: frontend
  labels:
    app: guestbook
spec:
 selector:
    matchLabels:
      app: guestbook
      tier: frontend
  replicas: 3
  template:
   metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
      name: php-redis
        image: gcr.io/google-samples/gb-frontend:v4
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        env:
        - name: GET_HOSTS_FROM
          value: dns
          # Using `GET HOSTS FROM=dns` requires your cluster to provide a dns service.
        - containerPort: 80
```

vagrant@kmaster:~/application/guestbook\$ kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-deployment.yaml
deployment.apps/frontend created

```
agrant@kmaster:~/application/guestbook$ kubectl get pods -l app=guestbook -l tier=frontend
                                                         RESTARTS
NAME
                            READY
                                    STATUS
                            0/1
                                    ContainerCreating
frontend-6c6d6dfd4d-c6cdz
                                                         0
                                                                     36s
frontend-6c6d6dfd4d-ds25q
                            0/1
                                                         0
                                                                     36s
                                    ContainerCreating
frontend-6c6d6dfd4d-t6h24
                            0/1
                                    ContainerCreating
                                                         0
                                                                     36s
```

vagrant@kmaster:~/application/guestbook\$ nano frontend-service.yaml

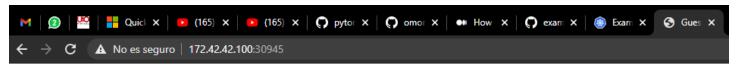
```
💹 vagrant@kmaster: ~/application/guestbook
 GNU nano 4.8
                                                                              frontend-service.yaml
apiVersion: v1
cind: Service
netadata:
 name: frontend
 labels:
   app: guestbook
   tier: frontend
spec:
 # comment or delete the following line if you want to use a LoadBalancer
 type: NodePort
 # if your cluster supports it, uncomment the following to automatically create
 # an external load-balanced IP for the frontend service.
 ports:
 - port: 80
 selector:
   app: guestbook
   tier: frontend
```

vagrant@kmaster:~/application/guestbook\$ kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-service.yaml
service/frontend created

```
/agrant@kmaster:~/
                                     book$ kubectl get services
NAME
                            CLUSTER-IP
                                             EXTERNAL-IP
               TYPE
                                                            PORT(S)
                                                                            AGE
frontend
               NodePort
                            10.97.170.134
                                             <none>
                                                            80:30945/TCP
                                                                            26s
               ClusterIP
                                                                            3d12h
kubernetes
                            10.96.0.1
                                                            443/TCP
                                             <none>
redis-master
               ClusterIP
                            10.104.67.158
                                                            6379/TCP
                                                                            24m
                                             <none>
redis-slave
                            10.107.129.44
                                                                            7m53s
               ClusterIP
                                                            6379/TCP
                                             <none>
```

vagrant@kmaster:~/application/	/guestbo	ok\$ kubectl	get pods	
NAME	READY	STATUS	RESTARTS	AGE
frontend-6c6d6dfd4d-c6cdz	1/1	Running	1	24m
frontend-6c6d6dfd4d-ds25q	1/1	Running	1	24m
frontend-6c6d6dfd4d-t6h24	1/1	Running	1	24m
redis-master-f46ff57fd-15hr2	1/1	Running	2	53m
redis-slave-bbc7f655d-7k9t8	1/1	Running	1	30m
redis-slave-bbc7f655d-glkbn	1/1	Running	1	30m

Se usa la ip de la máquina virtual (kmaster) pero el puerto redireccionado



Guestbook

Messages

hola probando el guestbook

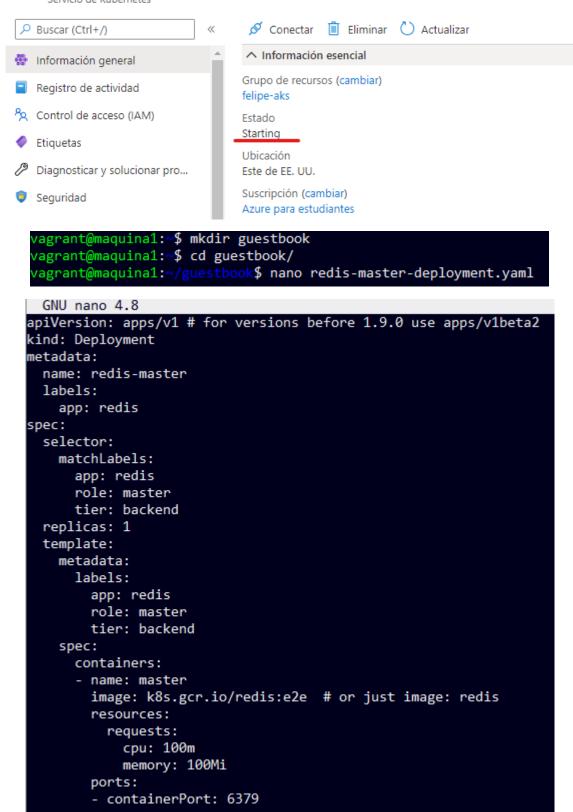
```
vagrant@kmaster:~/application/guestbook$ kubectl delete deployment -l app=redis
deployment.apps "redis-master" deleted
deployment.apps "redis-slave" deleted
vagrant@kmaster:~/application/guestbook$ kubectl delete service -l app=redis
service "redis-master" deleted
service "redis-slave" deleted
vagrant@kmaster:~/application/guestbook$ kubectl delete deployment -l app=guestbook
deployment.apps "frontend" deleted
vagrant@kmaster:~/application/guestbook$ kubectl delete service -l app=guestbook
service "frontend" deleted
vagrant@kmaster:~/application/guestbook$
```

AHORA EN AKS:

Para esto de hace az login en la terminal y se inicializa el aks: https://kubernetes.io/docs/tutorials/stateless-application/guestbook/

vagrant@maquina1:~\$ az aks start --name proyecto-cn --resource-group felipe-aks _- Running ..





```
GNU nano 4.8
apiVersion: v1
kind: Service
metadata:
  name: redis-master
  labels:
    app: redis
    role: master
    tier: backend
spec:
  ports:
  - name: redis
    port: 6379
    targetPort: 6379
  selector:
    app: redis
    role: master
    tier: backend
```

vagrant@maquina1:~/guestbook\$ nano redis-slave-deployment.yaml

```
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
 name: redis-slave
 labels:
   app: redis
spec:
 selector:
   matchLabels:
     app: redis
      role: slave
      tier: backend
  replicas: 2
  template:
   metadata:
      labels:
       app: redis
       role: slave
       tier: backend
   spec:
      containers:
      name: slave
        image: gcr.io/google_samples/gb-redisslave:v3
       resources:
          requests:
            cpu: 100m
            memory: 100Mi
        - name: GET HOSTS FROM
         value: dns
          # Using `GET HOSTS FROM=dns` requires your cluster to
        ports:
        - containerPort: 6379
```

```
apiVersion: v1
kind: Service
metadata:
   name: redis-slave
   labels:
    app: redis
    role: slave
    tier: backend
spec:
   ports:
   - port: 6379
   selector:
   app: redis
   role: slave
   tier: backend
```

vagrant@maquina1:~/guestbook\$ nano frontend-deployment.yaml

```
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
 name: frontend
  labels:
    app: guestbook
spec:
  selector:
    matchLabels:
      app: guestbook
      tier: frontend
  replicas: 3
  template:
   metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
      - name: php-redis
        image: gcr.io/google-samples/gb-frontend:v4
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        env:

    name: GET HOSTS FROM

          value: dns
          # Using `GET HOSTS FROM=dns` requires your cluster to
          # service launched automatically. However, if the clus
          # access an environment variable to find the master sen
        ports:
        - containerPort: 80
```

OJO A LO SIGUIENTE (MODIFICAR):

```
GNU nano 4.8
apiVersion: v1
kind: Service
metadata:
 name: frontend
 labels:
   app: guestbook
   tier: frontend
spec:
 type: LoadBalancer
 # if your cluster supports it, uncomment the following to au
 # an external load-balanced IP for the frontend service.
 ports:
  port: 80
  selector:
    app: guestbook
   tier: frontend
```

vagrant@maquina1:~/guestbook\$ ls frontend-deployment.yaml frontend-service.yaml redis-master-deployment.yaml redis-master-service.yaml redis-slave -deployment.yaml redis-slave-service.yaml

vagrant@maquina1:~/guestbook\$ kubectl apply -f redis-master-deployment.yaml
deployment.apps/redis-master created

```
vagrant@maquina1:~/guestbook$ kubectl get pods
NAME READY STATUS RESTARTS AGE
redis-master-f46ff57fd-r1764 1/1 Running 0 82s
```

vagrant@maquina1:~/guestbook\$ kubectl apply -f redis-master-service.yaml service/redis-master created

```
vagrant@maquina1:~/guestbook$ kubectl get service
NAME
              TYPE
                          CLUSTER-IP
                                         EXTERNAL-IP
                                                       PORT(S)
                                                                  AGE
                          10.0.0.1
                                                       443/TCP
                                                                  19h
kubernetes
              ClusterIP
                                         <none>
redis-master ClusterIP 10.0.204.213 <none>
                                                       6379/TCP
                                                                  26s
```

vagrant@maquina1:~/guestbook\$ kubectl apply -f redis-slave-deployment.yaml
deployment.apps/redis-slave created

```
vagrant@maquina1:~/guestbook$ kubectl get pods
NAME
                                READY
                                        STATUS
                                                  RESTARTS
                                                             AGE
redis-master-f46ff57fd-r1764
                                                              3m36s
                                1/1
                                        Running
                                                  0
redis-slave-bbc7f655d-ghvv7
                                1/1
                                        Running
                                                  0
                                                              26s
                               1/1
redis-slave-bbc7f655d-pp9bb
                                        Running
                                                  0
                                                              26s
```

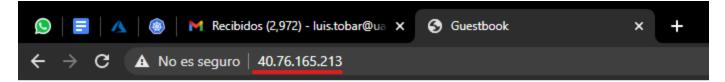
vagrant@maquina1:~/guestbook\$ kubectl apply -f redis-slave-service.yaml
service/redis-slave created

```
vagrant@maquina1:~/guestbook$ kubectl get services
NAME
               TYPE
                            CLUSTER-IP
                                           EXTERNAL-IP
                                                          PORT(S)
                                                                     AGE
               ClusterIP
kubernetes
                            10.0.0.1
                                           <none>
                                                          443/TCP
                                                                     19h
redis-master
               ClusterIP
                            10.0.204.213
                                           <none>
                                                          6379/TCP
                                                                     2m57s
               ClusterIP
                            10.0.86.205
redis-slave
                                           <none>
                                                          6379/TCP
                                                                     27s
```

vagrant@maquina1:~/guestbook\$ kubectl apply -f frontend-deployment.yaml
deployment.apps/frontend created

```
/agrant@maquina1:~/guestbook$ kubectl get pods -l app=guestbook -l tier=frontend
NAME
                                               RESTARTS
                            READY
                                    STATUS
                                                          AGE
frontend-6c6d6dfd4d-jcz7f
                            1/1
                                                          62s
                                    Running
                                              0
frontend-6c6d6dfd4d-n6pgm
                            1/1
                                                          62s
                                    Running
                                              0
frontend-6c6d6dfd4d-w2rhb
                            1/1
                                    Running 0
                                                          62s
```

vagrant@maquina1:~/guestbook\$ kubectl apply -f frontend-service.yaml
service/frontend created



Guestbook

Messages

Submit

hola prueba de questbook

```
vagrant@maquina1:~/guestbook$ kubectl delete deployment -l app=redis
deployment.apps "redis-master" deleted
deployment.apps "redis-slave" deleted
vagrant@maquina1:~/guestbook$ kubectl delete service -l app=redis
service "redis-master" deleted
service "redis-slave" deleted
vagrant@maquina1:~/guestbook$ kubectl delete deployment -l app=guestbook
deployment.apps "frontend" deleted
vagrant@maquina1:~/guestbook$ kubectl delete service -l app=guestbook
service "frontend" deleted
vagrant@maquina1:~/guestbook$ kubectl get pods
No resources found in default namespace.
```

PUNTO 4:

4. Supervisión y monitoreo en Azure

Demuestre el uso de los servicios de supervisión y monitoreo que provee AKS, para esto puede hacer uso de alguna de las aplicaciones que desplegó en los puntos anteriores.

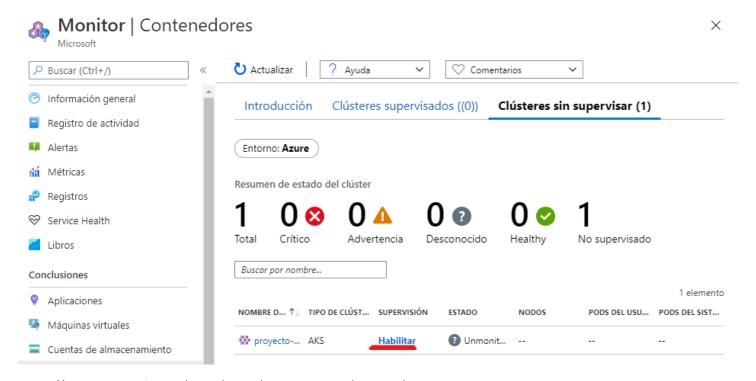
Desde que se crea se puede habilitar el monitoreo:

az aks create --resource-group felipe-aks --name proyecto-cn --node-count 2 **--enable-addons** monitoring --generate-ssh-keys --kubernetes-version 1.19.3 --node-vm-size=Standard B2s

También si se tiene desactivado, se puede activar con el comando:

az aks enable-addons -a monitoring -n MyExistingManagedCluster -g MyExistingManagedClusterRG

Se puede también por la web de Azure:



https://docs.microsoft.com/es-es/azure/azure-monitor/insights/container-insights-analyze

Azure cuenta con una interfaz de los clústeres AKS llamada Azure monitor en la cual podemos observar cuantos cluster se encuentras en estado crítico o incorrecto y cuanto en buen estado.

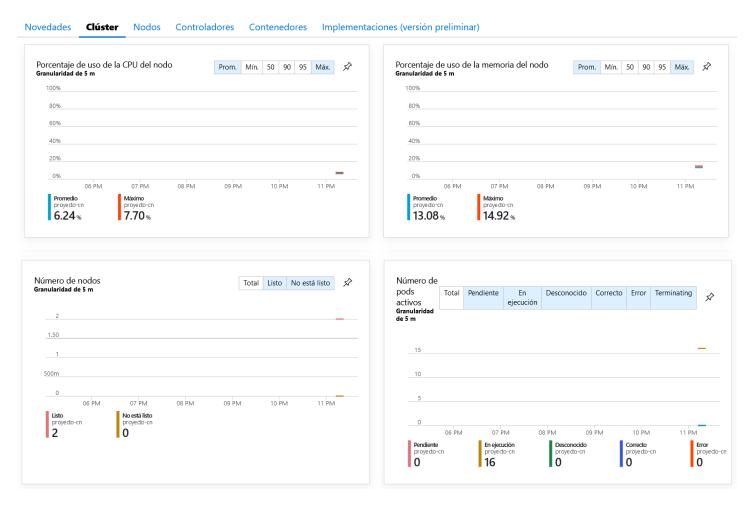
Los estados de mantenimiento incluidos son:

- Correcto: no se detectó ningún problema para la VM y funciona adecuadamente.
- **Crítico** : se detectan uno o varios problemas críticos que deben solucionarse para restaurar el estado de funcionamiento normal según lo previsto.
- **Advertencia** : se detectan uno o varios problemas que deben solucionarse o el estado puede volverse crítico.

- **Desconocido**: si el servicio no pudo establecer ninguna conexión con el nodo o el pod, el estado se cambia a un estado desconocido.
- **No se encuentra** : se eliminó el área de trabajo, el grupo de recursos o la suscripción que contiene el área de trabajo para esta solución.
- No autorizado : el usuario no tiene los permisos necesarios para leer los datos del área de trabajo.
- Error : al intentar leer los datos del área de trabajo.
- **Configuración errónea** : Azure Monitor para contenedores no se configuró correctamente en el área de trabajo especificada.
- No hay datos : no se han notificado los datos en el área de trabajo durante los últimos 30 minutos.

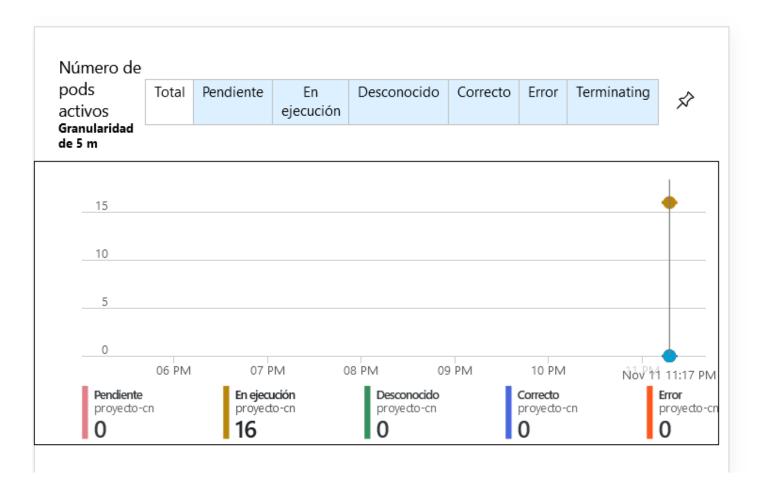
La informacion sobre el cluster esta organizada en 4 perspectivas

• Clúster

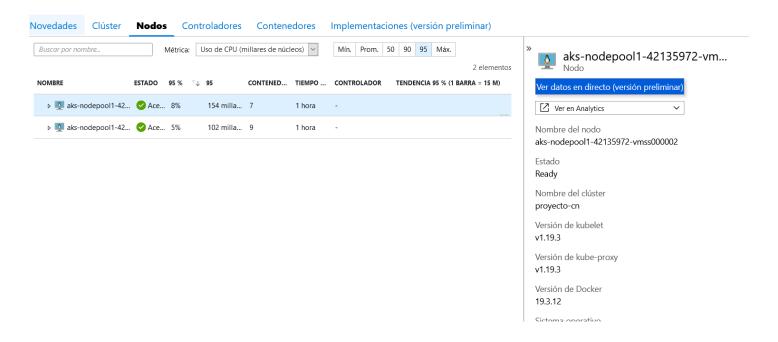


Podemos observar graficas de uso de CPU del nodo en porcentaje para todo el cluster y la utilizacion de memoria del nodo. Los resultados pueden ser filtrados segun intervalos de tiempo acorde al selector de percentiles situado encima del grafico.

Las graficas inferiores muestran el numero de nodos y su estado como ready(listo) y not ready(no esta listo) ademas de el numero de pods activos representados como total , pendiente, en ejecucion , desconocido, correcto o con errores.



• Nodos

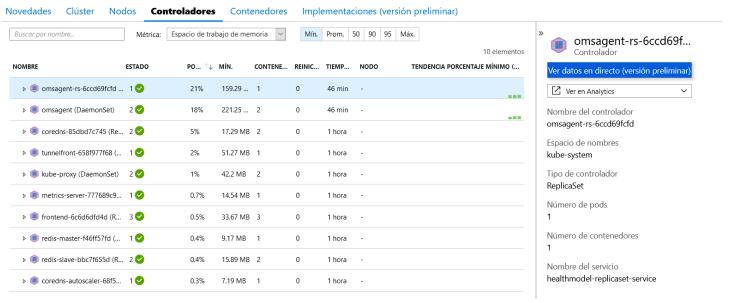


▲ Local Disk Capacity

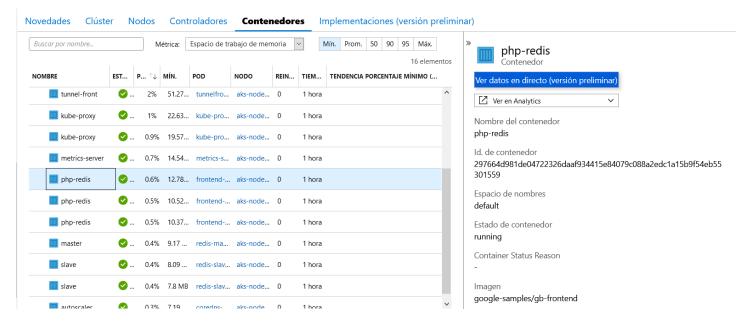
Device	Path	Used %	Free
sda1	/	9%	112.2 GB
sda15	/boot/efi	3%	100.76 MB
sdb1	/mnt/docker	0.5%	7.36 GB

Controladores

Aquí puede ver el estado de rendimiento de los controladores y a los controladores del nodo virtual de Container Instances o los pods de nodo virtual no conectados a un controlador.



Contenedores



En un contenedor, puede explorar en profundidad un pod o un nodo para ver los datos de rendimiento filtrados por ese objeto