# Implementación, Supervisión y Monitoreo de Aplicaciones en Amazon Kubernetes Service AKS

Fierro, Rubén Dario<sup>1</sup>, Velasco Rodriguez, Daniel<sup>2</sup>
ruben.fierro@uao.edu.co<sup>1</sup>
daniel.velasco\_rod@uao.edu.co<sup>2</sup>
Computación en la Nube
Universidad Autónoma de Occidente

# I. INTRODUCCIÓN

Azure Kubernetes Services (AKS) es un servicio de Microsoft de contenerización que nos permite de una manera sencilla implementar, administrar y usar Kubernetes, como un servicio de orquestador de contenedores ofreciendo el uso y administración de clúster sobre máquinas virtuales y recursos de almacenamiento.

La orquestación de contenedores hace referencia a enlazar contenedores uno con otros permitiendo exponer servicios, escalar, organizar y desplegar aplicaciones sin afectar la disponibilidad del servicio de una manera automatizada en conjuntos de máquinas virtuales.

Kubernetes fue liberada por Google desde el año 2014 y es uno de los orquestadores más usados actualmente. Es un sistema open-source para la automatización de despliegue, el escalado y la gestión de aplicaciones en contenedores permite hacer orquestación y aportar muchas soluciones en el mercado actual.

# A. Contenedores Docker

Docker es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. Con Docker, puede implementar

y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará. [1].

#### B. Kubernetes

Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios [2]. Kubernetes ofrece un entorno de administración centrado en contenedores. Kubernetes orquesta la infraestructura de cómputo, redes y almacenamiento para que las cargas de trabajo de los usuarios no tengan que hacerlo. Esto ofrece la simplicidad de las Plataformas como Servicio (PaaS) con la flexibilidad de la Infraestructura como Servicio (IaaS) y permite la portabilidad entre proveedores de infraestructura.

# C. Microsoft Azure

Azure es un conjunto de servicios en la nube de la empresa Microsoft. Con Azure es posible almacenar información y crear, administrar e implementar aplicaciones en cloud [3]. Entre los beneficios que podemos obtener se encuentran:

- Almacenamiento en la nube.
- Reducción de costos.

- Integración con otras plataformas en la nube.
- Escalable y accesible.
- Trabajo híbrido.

# II. PLANTEAMIENTO DEL PROBLEMA

Se requiere implementar un clúster de Azure Kubernetes Service (AKS) de al menos dos nodos, mediante Azure Portal. Para esto se debe crear una cuenta para estudiantes en Azure, cuenta que nos ofrece la Universidad Autónoma de Occidente con una suscripción para estudiantes. A diferencia del Microproyecto 1, no se utilizarán archivos de aprovisionamiento Shell para aprovisionar las máquinas virtuales y crear el cluster, se tendrá que hacer uso de las herramientas que proporciona Microsoft Azure en su portal para crear el cluster de Kubernetes (AKS).

Como requerimientos para el desarrollo de este Microproyecto 2, se necesita crear al clúster tanto desde el Portal Azure y su interfaz como por CLI de Azure. También se debe acceder a dicho cluster por medio de Cloud Shell y por CLI de Azure a través de otra máquina virtual.

Una vez que se ha creado y realizado el acceso al clúster se deben implementar dos aplicaciones:

- La primera aplicación es un clasificador de imágenes que utiliza algoritmos de inteligencia artificial. Se debe desplegar el clasificador y comprobar su funcionamiento [4].
- La segunda aplicación es un guestbook, es una aplicación web que se utiliza como un libro de anotaciones. De igual manera se debe hacer el despliegue y comprobar su funcionamiento [5].

Por último, se necesita hacer uso de las herramientas de monitoreo que nos ofrece el Portal Azure, aquí se mostraran algunas métricas que se utilizan para la supervisión del clúster implementado.

# III. SOLUCIÓN

# A. Creación del clúster AKS desde Portal Azure

Desde la web de Portal Azure se puede acceder al AKS (Azure Kubernetes Services) e iniciar a configurar todos los parámetros que se requieren para crearlo, se hace la implementación de la siguiente manera:

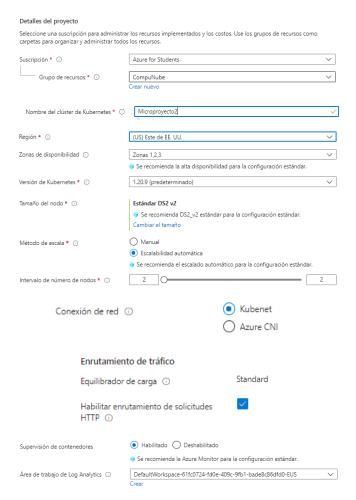


Fig. 1. Configuración clúster AKS desde el Portal Azure.

Una vez configurados los parámetros para la implementación del clúster debemos darle a la opción revisar y crear. Finalmente creamos el cluster llamado 'Microproyecto2' en el grupo de recursos llamado 'CompuNube'.

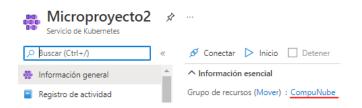


Fig. 2. Creación clúster AKS desde el Portal Azure.

# B. Creación del clúster AKS desde CLI Azure

Para la creación del clúster mediante el CLI de Azure, hacemos uso de una máquina virtual sencilla administrada por Vagrant. Como primer paso una vez ingresamos por ssh a la máquina virtual creada es necesario hacer un apt update & upgrade e instalar CLI mediante las siguientes líneas:

```
sudo apt-get install ca-certificates curl apt-transport-https lsb-release gnupg

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

Una vez instalado CLI podemos hacer el login a nuestra cuenta de Azure

Fig. 3. Ingreso a Azure por medio del CLI

# Instalamos algunos proveedores Microsoft

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

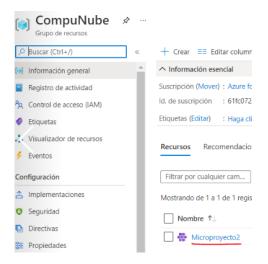
Procedemos a crear el grupo de recursos, donde en la siguiente línea vemos que se llama CompuNube, localizado al este de EEUU. También creamos el clúster AKS, de igual manera en la segunda línea donde configuramos en que grupo de recursos se creará, el nombre, ubicación, versión de Kubernetes y el tamaño de nuestra máquina virtual.

```
vagrant@maquina1:~$ az group create -n CompuNube --location eastus
{
    "id": "/subscriptions/61fc0724-fd0e-409c-9fb1-bade8c86dfd0/resour
    "location": "eastus",
    "managedBy": null,
    "name": "CompuNube",
    "properties": {
        "provisioningState": "Succeeded"
    },
    "tags": null,
    "type": "Microsoft.Resources/resourceGroups"

vagrant@maquina1:~$ az aks create --resource-group CompuNube
    --name Microproyecto2 --node-count
2 --generate-ssh-keys --kubernetes-version 1.20.9
    --node-vm-size=Standard_B2s
```

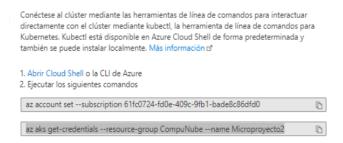
Fig. 4. Creación del clúster mediante CLI Azure.

Finalmente accedemos al Portal Azure para verificar si nuestro clúster AKS se creó correctamente.



# C. Acceso a clúster AKS mediante Cloud Shell en Portal Azure

Para realizar el acceso mediante Cloud Shell es necesario hacer uso de las credenciales que el Portal Azure nos entrega al momento de conectar con el clúster Microproyecto2



Accedemos a Cloud Shell en la parte superior derecha de nuestra pantalla y seleccionamos el terminal de PowerShell, donde ingresamos las credenciales.

```
PowerShell V O ? () [] [] [] () []
Welcome to Azure Cloud Shell

Type "az" to use Azure CLI

Type "help" to learn about Cloud Shell

MOTD: Cmdlet help is available: help <cmdlet name>

VERBOSE: Authenticating to Azure ...

VERBOSE: Building your Azure drive ...

PS /home/daniel> az account set --subscription 61fc0724-fd0e-409c-9fb1-bade8c86dfd0

PS /home/daniel> az aks get-credentials --resource-group CompuNube --name Microproyecto

Merged "Microproyecto2" as current context in /home/daniel/.kube/config
```

# D. Acceso a clúster AKS mediante CLI Azure desde máquina local

El proceso es muy similar, únicamente debemos instalar algunas dependencias y herramientas que vamos a necesitar.

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

[DanielV][ ]
[-]> choco install kubernetes-cli
Chocolatey v0.11.3
2 validations performed. 1 success(es), 1 warning(s), and 0 error(s).
```

# Hacemos login:

```
[DanielV][ ]

[-]> az login

The default web browser has been opened at https://log
he login in the web browser. If no web browser is avai
n --use-device-code .
[
```

Y utilizamos las credenciales del clúster Microproyecto2.

```
[DanielV][f]

[a]> az account set —subscription 61fc0724-fd0e-409c-9fb1-bade8c86dfd0

[DanielV][f]

[a]> az aks get-credentials —resource-group CompuNube —name Microproyecto2

Merged *Microproyecto2* as current context in C:\Users\DanielV\.kube\config
```

 E. Implementación del clasificador de imágenes mediante CLI Azure.

Para la implementación del clasificador de imágenes seguimos la guía suministrada para el desarrollo del Microproyecto [6].

Seguimos los pasos de la guía y creamos todos los archivos necesarios para realizar el despliegue de la aplicación de clasificación de imágenes, estos archivos los guardamos en nuestro Github para tener mayor facilidad para trabajar.

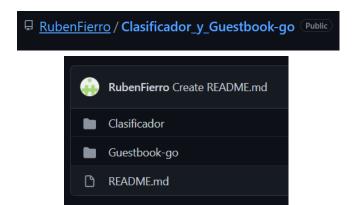


Fig. 5. Repositorio Github utilizado.

Como podemos observar tenemos dos carpetas, una para el clasificador de imágenes y la segunda para la implementación de un servicio web que mostraremos más adelante en el despliegue de Guesbook-go.

Lo primero que se hace es hacer az login, usar las credenciales del clúster Microproyecto2 y finalmente clonar el repositorio para obtener las carpetas con todos los archivos necesarios para el despliegue de la aplicación.

```
vagrant@maquina1:-$ git clone https://github.com/RubenFierro/Clasificador_imagenes_.git
Cloning into 'Clasificador_imagenes_' ...
vagrant@maquina1:-$ ls
Clasificador_imagenes_ Clasificador_y_Guestbook-go pytorch-kubernetes
```

Accedemos a la carpeta que contiene los archivos del clasificador de imágenes y ejecutamos las siguientes líneas:

# \$ kubectl apply -f deployment.yaml

Para no tener que especificar la ip del nodo que contiene el servicio junto con el puerto (como se haría si el tipo de exposición del servicio fuera NodePort) se expone el servicio de forma "LoadBalancer", lo que generará una ip externa que servirá como frontend para el servicio y es la que se usará para hacer requerimientos por parte de los usuarios.

```
$ kubectl expose deployment kubermatic-dl-deployment
--type=LoadBalancer --port 80
--ta rget-port 5000
```

Verificamos mediante Kubectl get pods, services y deployment que todo esté correctamente configurado e inicializado.

<pre>na1:~\$ kubectl get servicesal</pre>	l-namespaces			
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
guestbook	LoadBalancer	10.0.92.227	52.226.249.219	3000:30663/TCP
kubermatic-dl-deployment	LoadBalancer	10.0.224.46	20.85.251.81	80:31949/TCP
kubernetes	ClusterIP	10.0.0.1	<none></none>	443/TCP
redis-master	ClusterIP	10.0.234.116	<none></none>	6379/TCP
redis-slave	ClusterIP	10.0.78.109	<none></none>	6379/TCP
image-classifier	LoadBalancer	10.0.24.217	52.226.244.46	80:30763/TCP
healthmodel-replicaset-service	ClusterIP	10.0.249.126	<none></none>	25227/TCP
kube-dns	ClusterIP	10.0.0.10	<none></none>	53/UDP,53/TCP
metrics-server	ClusterIP	10.0.29.154	<none></none>	443/TCP

nal:~\$ kubectl get podsall-namespaces		-		
NAME	READY	STATUS	RESTARTS	AGE
guestbook-5dltw	1/1	Running	Ø	11h
guestbook-5×68q	1/1	Running	Ø	11h
guestbook-z7b5z	1/1	Running	Ø	11h
<pre>kubermatic-dl-deployment-bfdcb9898-7t6rv</pre>	1/1	Running	0	11h
kubermatic-dl-deployment-bfdcb9898-fp5rv	1/1	Running	0	11h
kubermatic-dl-deployment-bfdcb9898-vp2kk	1/1	Running	Ø	11h

nal:~\$ kubectl get deploymentsall-namespaces=true					
NAME READY UP-TO-DATE AVAILABLE AGE					
kubermatic-dl-deployment	3/3	3	3	2d22h	
image-classifier	1/1	1	1	3d16h	

Finalmente hacemos dos pruebas con curl pasándole primero una imagen local de un perro para que la aplicación realice la predicción. La segunda prueba es otro curl, pero esta vez le pasamos una imagen de un caballo como vemos a continuación:

\$ curl -X POST -F img=@dog.jpg http://20.85.251.81/predict



The input picture is classified as [dog], with probability 0.97

Fig. 7. Respuesta 1 aplicación de clasificación de imágenes.



The input picture is classified as [horse], with probability 0.864.

Fig. 8. Respuesta 2 aplicación de clasificación de imágenes.

F. Implementación del servicio web Guestbook mediante CLI Azure.

Para la implementación del servicio web, se utiliza la guía que proporciona Kubernetes [7].

Seguimos los pasos de la guía y creamos todos los archivos necesarios para realizar el despliegue del servicio, hacemos uso del mismo repositorio Github utilizado anteriormente para guardar los archivos creados.

El primer paso es crear el pod maestro de Redis, utilizamos el archivo redis-master-controller.json, que es un archivo para crear un controlador de replicación.

\$ kubectl apply -f redis-master-controller.j

Creamos el servicio maestro de Redis, este es un servicio de Kubernetes configurado como balanceador de carga que envía el tráfico a uno o más pods.

\$ kubectl apply -f redis-master-service.json

Creamos los pods esclavos de Redis, estos esclavos son pods replicados, un controlador de replicación es responsable de administrar las múltiples instancias de un pod replicado.

```
kubectl apply -f redis-slave-controller.js
```

Ahora creamos el servicio esclavo de Redis, al igual que el servicio maestro, este servicio esclavo proporciona un balanceo de carga a los clientes

```
$ kubectl apply -f redis-slave-service.json
```

Creamos los pods del guestbook, que esta configurado para hablar con los servicios maestro o esclavo dependiendo de si la solicitud es de lectura o escritura. Estos pods también son administrados por un controlador de replicación.

```
$ kubectl apply -f guestbook-controller.json
```

Finalmente creamos el servicio guestbook, la función de este servicio es agrupar los pods del guestbook para que sea visible externamente, se configura "type: LoadBalancer"

```
$ kubectl apply -f guestbook-service.json
```

Una vez implementados todos los pods, servicios y controladores validamos que estén correctamente inicializados mediante 'kubectl'.

# \$ kubectl get services

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
guestbook	LoadBalancer	10.0.237.219	20.88.177.244	3000:31911
kubernetes	ClusterIP	10.0.0.1	<none></none>	443/TCP
redis-master	ClusterIP	10.0.234.16	<none></none>	6379/TCP
redis-slave	ClusterIP	10.0.107.142	<none></none>	6379/TCP
\$ kubect	l get pod	S		

NAME	READY	STATUS	RESTARTS	AGE
guestbook-8vhz9	1/1	Running	0	12m
guestbook-smnqp	1/1	Running	0	12m
guestbook-z6qr9	1/1	Running	0	12m
redis-master-dhk9b	1/1	Running	0	10m
redis-slave-8m4th	1/1	Running	0	14m
redis-slave-z888q	1/1	Running	0	14m

\$ kubectl g	et :	rc
--------------	------	----

NAME	DESIRED	CURRENT	READY	AGE
guestbook	3	3	3	12m
redis-master	1	1	1	10m
redis-slave	2	2	2	15m

Fig. 9. Verificación de los pods, servicios y controladores para Guestbook-go.

Como se observa en la anterior figura, mediante la ip externa creada por el servicio para frontend, se puede acceder a la aplicación y realizar pruebas de funcionamiento mediante el navegador. Accedemos al guestbook a través del navegador usando la ip externa '20.88.177.244:3000'.



Fig. 10. Aplicación Guestbook corriendo en el navegador con la ip externa.

# IV. MONITOREO Y SUPERVISIÓN

Azure Monitor ayuda a maximizar la disponibilidad y el rendimiento de las aplicaciones y los servicios. Ofrece una solución completa para recopilar, analizar y actuar en la telemetría desde los entornos local y en la nube. Esta información le ayudará a conocer el rendimiento de las aplicaciones y a identificar de manera proactiva los problemas que les afectan y los recursos de los que dependen [8].

Se puede hacer uso de la herramienta de monitoreo, el primer paso es tener la opción de supervisión activado al momento de la creación del clúster AKS. Teniendo esto en cuenta se procede a analizar todas las

herramientas que nos ofrece el Portal Azure ingresando a la ventana de monitoreo y buscando la opción contenedores.



Al ingresar al monitor vamos a ver la siguiente información, aquí nos dice el total de cluster que tenemos creados y cuales de ellos se encuentran supervisados por la herramienta y cuales no. También podemos ver otros estados que deben tenerse en cuenta cuando algún cluster presente estado crítico, advertencia, desconocido, healthy y no supervisado.



Fig. 12. Azure monitor para supervisión del cluster implementado.

Dentro de los clústeres supervisados tenemos habilitada la supervisión para el Microproyecto2, además podemos observar un informe general que nos permite ver el estado del cluster, en este caso Healthy, tipo de cluster AKS, versión de Kubernetes utilizada, número de nodos y pods.

Ingresando a Microproyecto2 podemos ver todas las opciones de monitoreo que nos ofrece el Portal Azure. Haciendo uso de algunos filtros e intervalo de tiempo podemos ver:



#### A. Monitoreo Clúster

Podemos observar gráficas de uso del porcentaje de uso de CPU para el nodo para todo el cluster, también podemos ver la cantidad de memoria que está utilizando el nodo. Los resultados pueden ser filtrados según intervalos de tiempo.

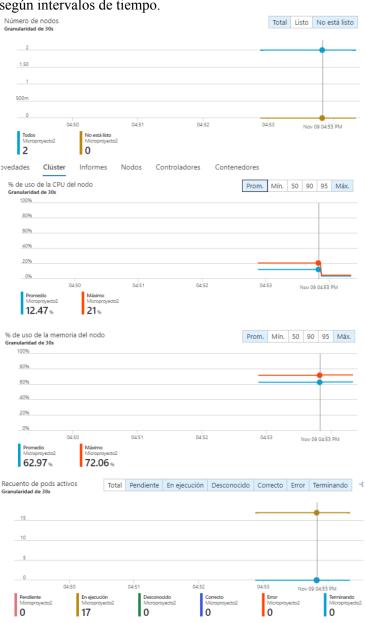
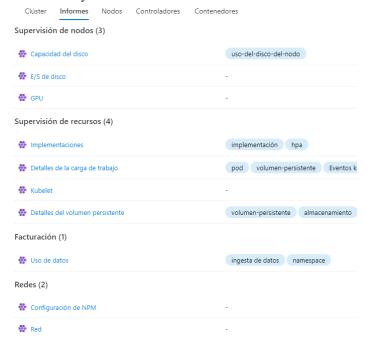


Fig. 13. Monitoreo del Cluster, número de nodos, uso de CPU, uso de memoria y recuento de pods.

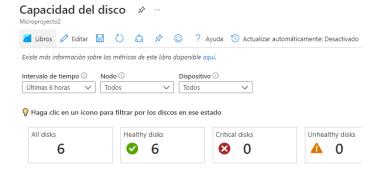
Se observa en los gráficos que el consumo es mínimo debido a que las peticiones no fueron demasiadas, por el contrario, se realizaron pocas pruebas ya que el consumo de créditos para la suscripción de estudiante es limitado.

#### B. Informes

El monitor de contenedores también nos permite ver algunos informes acerca de los nodos, recursos, facturación y redes como vemos a continuación



Si accedemos por ejemplo a los discos, podemos observar la cantidad total de discos que estamos usando, la capacidad de cada uno de ellos y el máximo uso que se le ha dado.



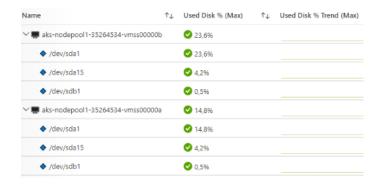
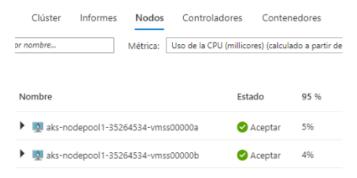


Fig. 14. Informes para supervisión de nodos, capacidad de los discos en uso.

# C. Nodos

Si seguimos navegando en las pestañas que nos ofrece el monitor Azure, para los nodos podemos ver la cantidad de nodos que tiene nuestro cluster, para este caso tenemos 2 nodos.



Podemos ver el estado de cada nodo y si accedemos a cualquiera de ellos podemos ver la siguiente información:

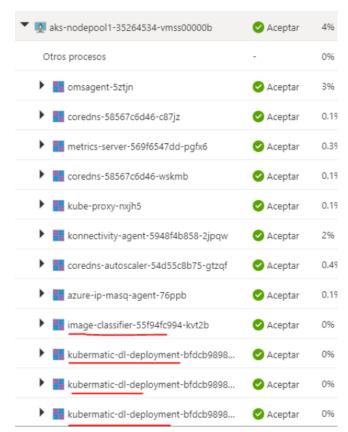


Fig. 14. Monitoreo de nodos.

#### D. Controladores

En la pestaña de controladores vemos todos los que implementamos, podemos observar kubermatic-dll-deployment, corresponde a la aplicación de clasificación de imágenes implementada.

Nombre	Estado	95 % ↓	95	Conten
• (ReplicaSet)	1 🕢	22%	222.16 MB	1
	2 🕢	1796	594.71 MB	4
lacksquare (ReplicaSet)	2 🕢	1596	38.89 MB	2
▶ 📵 kubermatic-dl-deployment-bfdcb9898 (Rep	3 🕢	9%	586.98 MB	3
image-classifier-55f94fc994 (ReplicaSet)	1 🕢	7%	157.59 MB	1
	2 🕢	5%	48.61 MB	2
	2 🕢	5%	23.45 MB	2
	1 🕢	3%	16.67 MB	1
metrics-server-569f6547dd (ReplicaSet)	1 🕢	196	25.44 MB	1
lackube-proxy (DaemonSet)	2 🕢	1%	47.8 MB	2

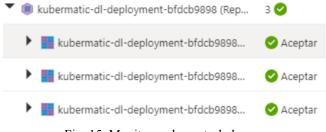
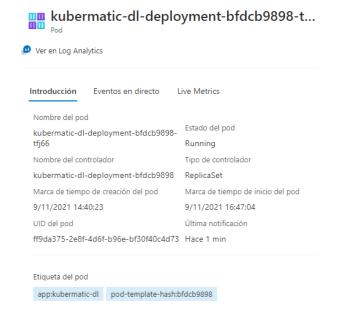


Fig. 15. Monitoreo de controladores.

Accediendo a cualquiera de los controladores nos brinda información más detallada como estado del pod, tipo de controlador, eventos en directo y también nos ofrece la posibilidad de ver métricas en vivo en el caso de que se esté ejecutando.



# E. Contenedores

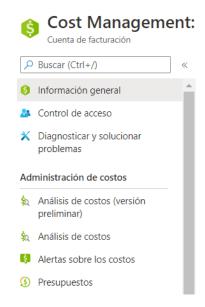
Finalmente podemos monitorear los contenedores que estamos usando en el clúster, accediendo a cualquiera de ellos podemos ver información más detallada, como el estado del contenedor, limite de CPU, límite de memoria. Al igual que para el apartado de controladores nos permite ver eventos en directo y la posibilidad de ver métricas en vivo en el caso de que se esté ejecutando.



Fig. 15. Monitoreo de contenedores.

# V. MANEJO DE COSTOS

Cost Management es un conjunto de herramientas que proporciona Microsoft para ayudarle a analizar, administrar y optimizar los costos de sus cargas de trabajo. El uso del conjunto de herramientas ayuda a garantizar que su organización está aprovechando las ventajas que proporciona la nube [9].

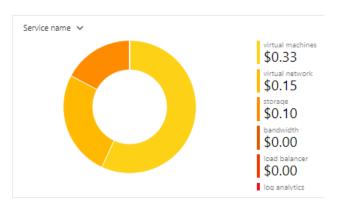


Para acceder a esta herramienta debemos ingresar a 'Cost Management' y luego 'Análisis de costos' aquí escogemos la suscripción para la cual queremos ver los costos y podemos ver diferentes gráficas con el historial de costos en el intervalo de tiempo que uno desee analizar.



Como se observa en la imagen anterior, escogemos el intervalo de días en el cual se creó y se trabajó con el clúster Microproyecto2 y el grupo de recursos CompuNube.

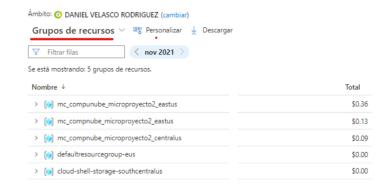
Por otro lado, también podemos generar distintas gráficas de manera que nos permite ver los gastos ya sea por servicios, localización, las suscripciones que tengamos activas, entre otras.



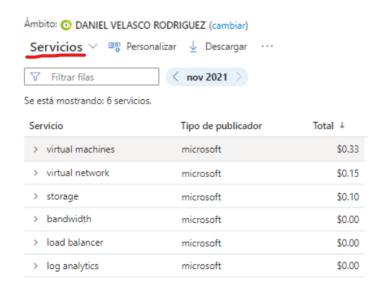




Podemos ver las estadísticas de costos para cada uno de los grupos de recursos creados.



De igual manera podemos cambiar la pestaña a 'Servicios' de esta forma nos permite ver las estadísticas de costos para los diferentes servicios que estamos usando en todos los grupos de recursos.



# A. Alarmas de presupuesto

Una última herramienta interesante que nos proporciona tener un control sobre los costos y para evitar posibles cargos adicionales de ciertos servicios que olvidamos apagar o destruir es la creación de presupuestos, donde podemos asignar un valor límite para el cual queremos recibir una notificación al correo asignado si nuestro grupo de recursos con los servicios implementados alcanzan dicho valor.



Como podemos observar el presupuesto 'prueba' tiene un límite de 3usd\$ y se tiene un progreso de 74.6% de dicho límite, al momento de acercarse y al momento de alcanzar ese límite se recibirá notificación al correo electrónico para poder tomar acciones al respecto.

# VI. CONCLUSIONES

- El uso de clústeres y orquestadores nos permite administrar todos los servicios desde un solo clúster, esto nos proporciona más robustez ante fallas a los servicios que se proporcionen, por otro lado, el uso de Kubernetes para desplegar aplicaciones brinda facilidad de implementación.
- Gracias a AKS podemos administrar de una manera más simplificada e implementar aplicaciones basadas en contenedores Docker, esto gracias a todo su desarrollo de entorno en la nube mediante interfaces gráficas para implementar y seleccionar recursos y características diferentes.
- Gracias al Portal Azure y AKS podemos realizar monitoreo y supervisión completa sin necesidad de configuración compleja para nuestros servicios, Azure monitor también nos proporciona un sistema de notificaciones y programación automática para soluciones ante situaciones que puedan causar problemas como fallos.
- La documentación tan completa que nos brinda Microsoft Azure nos permitió aun sin tener mucha experiencia en el Portal Azure poder entender, implementar y analizar los resultados de las implementaciones realizadas para el Microproyecto sin ninguna complicación, y aun cuando se presentaba, acudiendo a los documentos fue suficiente para seguir trabajando sin problemas.

#### REFERENCIAS

- [1] AWS, ¿Qué es Docker? [En línea]. Disponible en: https://aws.amazon.com/es/docker/
- [2] ¿Qué es Kubernetes?, junio 2020. [En línea]. Disponible en:

https://kubernetes.io/es/docs/concepts/overview/what-is-kubernetes/

- [3]Ambit, Microsoft Azure, el cloud de Microsoft ¿Qué es y para qué sirve? [En línea]. Disponible en: <a href="https://www.ambit-bst.com/blog/microsoft-azure-el-cloud-de-microsoft-qu%C3%A9-es-y-para-qu%C3%A9-es-y-para-qu%C3%A9-sirve">https://www.ambit-bst.com/blog/microsoft-azure-el-cloud-de-microsoft-qu%C3%A9-es-y-para-qu%C3%A9-es-y-
- [4] Opensource; How to deploy a Deep Learning model on Kubernetes, septiembre 2020 [En línea]. Disponible en: <a href="https://opensource.com/article/20/9/deep-learning-model-kubernetes">https://opensource.com/article/20/9/deep-learning-model-kubernetes</a>
- [5] Github; Kubernetes examples, noviembre 2020 [En línea]. Disponible en: <a href="https://github.com/kubernetes/examples/tree/master/guestbook-go#step-zero">https://github.com/kubernetes/examples/tree/master/guestbook-go#step-zero</a>
- [6] Opensource; How to deploy a Deep Learning model on Kubernetes, septiembre 2020 [En línea]. Disponible en: <a href="https://opensource.com/article/20/9/deep-learning-model-kubernetes">https://opensource.com/article/20/9/deep-learning-model-kubernetes</a>
- [7] Github; Kubernetes examples, noviembre 2020 [En línea]. Disponible en: <a href="https://github.com/kubernetes/examples/tree/master/guestbook-go#step-zero">https://github.com/kubernetes/examples/tree/master/guestbook-go#step-zero</a>
- [8] Microsoft; Introducción a Azure Monitor, abril 2021 [En línea] Disponible en: <a href="https://docs.microsoft.com/es-es/azure/azure-monitor/overview">https://docs.microsoft.com/es-es/azure/azure-monitor/overview</a>
- [9] Microsoft; ¿Qué es administración de costos + facturación?, octubre 2021 [En línea] Disponible en: <a href="https://docs.microsoft.com/es-es/azure/cost-management-billing/cost-management-billing-overview">https://docs.microsoft.com/es-es/azure/cost-management-billing-overview</a>