UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Despliegue de un cluster Kubernetes altamente disponible en Alibaba Cloud

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Jaramillo Sizalima, Mario Miguel

Tutor/a: Acebrón Linuesa, Floreal

CURSO ACADÉMICO: 2022/2023

Resumen

Este trabajo de fin de grado va a poner de manifiesto las distintas posibilidades existentes para realizar un despliegue de un clúster altamente disponible basado en Kubernetes haciendo uso de la plataforma de servicios basados en cloud de Alibaba Cloud.

Se realizan pruebas que permitirán comprobar como el entorno desplegado se puede adaptar a las necesidades del servicio prestado según los niveles de carga al que sea sometido, permitiendo optimizar el coste del despliegue. Esto se lleva a cabo mediante la implementación de un pequeño servicio web haciendo uso de una página web básica mediante HTML y un pequeño script PHP que nos permite saber a qué nodo del clúster estamos accediendo.

También se pretende mostrar las distintas opciones de almacenamiento persistente y se lleva a cabo una recopilación de las herramientas proporcionadas por Alibaba Cloud para poder realizar una monitorización del clúster mediante métricas, análisis de logs y tracing.

Palabras clave: Kubernetes, Alibaba, Cloud, Alta disponibilidad.

Abstract

This thesis will show the different possibilities to deploy a highly available clúster based on Kubernetes using Alibaba Cloud's cloud-based services platform.

Tests will be performed to verify how the deployed environment can be adapted to the needs of the service provided according to the load levels to which it is subjected, allowing to optimize the cost of the deployment. This is done by implementing a small web service using a basic HTML web page and a small PHP script that allows us to know which node of the clúster we are accessing.

It is also intended to show the different persistent storage options and a compilation of the tools provided by Alibaba Cloud to monitor the clúster through metrics, log analysis and tracing.

Keywords: Alibaba, Kubernetes, Cloud, High availability.

Índice general

Índice de ilustraciones

Índice de tablas

1. Introducción

Desde hace unas décadas se viene haciendo uso de un tipo de sistemas distribuidos que permiten unir a una serie de ordenadores mediante una red de alta velocidad para llevar a cabo una tarea común, comportarse como si de un único sistema se tratase, esto es lo que comúnmente conocemos como un clúster.

El origen del clúster, aunque no está confirmado, viene de la necesidad de poder llevar a cabo tareas muy complejas en términos de tiempo de cálculo, así pues, los orígenes se pueden remontar al físico estadounidense Eugene Myron Amdahl, que gracias a su desarrollo en 1967 de la, ya famosa, ley de Amdahl, fue posible comprender que mediante la paralelización de ciertas tareas podría reducirse el tiempo empleado para su finalización,

Actualmente, mediante el uso de un clúster, se busca responder a cuatro principales necesidades de la computación:

- Alta disponibilidad
- Rendimiento
- Equilibrado de la carga
- Escalabilidad

Durante mucho tiempo, la única solución existente para la mayoría de las organizaciones para poder satisfacer estas necesidades era mediante el despliegue de un clúster propio en un entorno físico propiedad de la entidad que lo necesitase. Dado que se trata de desplegar una serie de dispositivos que tienen un coste muy elevado, no solo en adquisición, sino también en cuanto a mantenimiento, no todas las empresas, entidades educativas o gubernamentales se podían permitir hacer uso de este tipo de clústeres.

Sin embargo, desde hace unos años, se ha podido observar que existen una serie de empresas que, debido a su alto poder adquisitivo, pueden permitirse desplegar una gran cantidad de estas máquinas para poder llevar a cabo una estrategia de negocio que no se había implementado aún. De esta forma, dado que las empresas más pequeñas u otras entidades que, anteriormente no se podían permitir el despliegue de un clúster, ahora pueden tener acceso a este tipo de herramientas a un precio más razonable, pues no tienen que realizar la compra de hardware ni llevar a cabo su mantenimiento, simplemente pagar por el uso real que den a dichas herramientas.

1.1 Motivación

Cada vez son más las empresas que desean implantar sus servicios en plataformas cloud en vez de tener su propia infraestructura para dicha implantación, no se trata únicamente de pequeñas y medianas empresas, sino también de grandes empresas que buscan ofrecer sus servicios a través de la infraestructura proporcionada por otras grandes empresas que sí disponen de la capacidad de ofrecer un servicio con una alta disponibilidad y rendimiento a un menor coste.

La tendencia alcista en este tipo de soluciones hace que se perciba que este va a ser el camino que adoptar de cara al futuro, y dado que se trata de un camino que falta por recorrer nos hace pensar que es necesario realizar estudios sobre como encaminar este tipo de soluciones para poder ofrecerlas de la forma más eficiente para las empresas.

Dado que ya se dispone de estudios anteriores relacionados con la virtualización de infraestructuras y servicios, la elección de este tema resulta interesante para poder seguir profundizando en el tema y adquiriendo los conocimientos necesarios para tener un mejor entendimiento de la materia.

1.2 Objetivos

El principal objetivo de este trabajo de final de grado es dar a conocer las distintas posibilidades existentes para el despliegue de un clúster Kubernetes altamente disponible en una plataforma de servicios cloud que no es tan conocida como sus principales competidores, este es el caso de la plataforma del gigante asiático Alibaba Cloud.

En este caso, no se van a exponer únicamente las formas de despliegue automatizadas disponibles en la plataforma, sino también dar a conocer una forma breve y sencilla para el despliegue de un clúster Kubernetes de forma manual, realizando un despliegue de máquinas virtuales en la plataforma Alibaba Cloud para posteriormente pasar a formar parte del clúster Kubernetes.

Otro objetivo del proyecto consiste en mostrar las distintas herramientas disponibles en Alibaba Cloud para el correcto mantenimiento de un clúster altamente disponible, como herramientas de monitorización, tracing, manejo de logs, almacenamiento persistente, etc.

De esta forma, se puede tener un marco de referencia para poder tener una alternativa adicional a las principales y más conocidas plataformas de servicios cloud como son Amazon Web Services (AWS), Microsoft Azure y Google Cloud. Permitiendo expandir los horizontes a otro terreno que aún no está suficientemente explorado. Para ello, se realiza una comparativa con los principales proveedores de servicios cloud a nivel global.

1.3 Estructura de la memoria

La presente memoria se va a dividir en siete apartados.

El actual y primer apartado, se encuentra una breve introducción, así como la motivación para la realización de este proyecto y los objetivos que se persiguen en el proyecto.

El segundo apartado proporciona información sobre el panorama previo y el actual del tema que se abarca en este proyecto.

El tercer apartado se centra en la realización de un análisis práctico del problema planteado para la elaboración del proyecto.

A lo largo del cuarto apartado se encuentra un análisis del estado actual del mercado, así como una comparación con los principales competidores de Alibaba Cloud.

En el quinto apartado se halla una descripción de los tipos de despliegues ofrecidos por Alibaba Cloud para un clúster Kubernetes, así como una serie de herramientas que permiten gestionar dicho clúster.

El sexto apartado se centra en un ejemplo de despliegue mediante las herramientas mostradas en el apartado anterior, de forma que se lleva a la práctica el apartado quinto.

El séptimo apartado muestra las conclusiones a las que se han llegado gracias a la realización de este proyecto, también se muestra cuáles de los objetivos se han cumplido y los posibles trabajos futuros.

En el octavo apartado se encuentran una serie de agradecimientos.

Por último, en el noveno apartado se documentan las fuentes que han sido consultadas para la elaboración del presente documento.

2. Estado del arte

Durante este apartado se va a introducir al lector al contexto actual sobre el que se encuentra el estado del arte, para de esta forma comprender la materia que se trata durante el desarrollo del presente proyecto.

2.1 Situación previa

Como se ha mencionado durante la introducción a este trabajo, la necesidad del uso de clústeres viene dada por la necesidad de realizar tareas que conllevan un coste computacional más elevado del que se había tenido hasta la fecha, bien por la realización de tareas más pesadas o bien por la necesidad de hacer frente a una carga de trabajo más amplia que necesita un correcto equilibrado de la misma para no crear cuellos de botella y posteriormente tener una saturación de la infraestructura del sistema.

Esto deriva en la necesidad de crear un sistema distribuido que sea altamente disponible y que pueda hacer frente a una gran cantidad de carga, que sea por lo tanto altamente eficiente.

Para solventar estos problemas se dieron a conocer los primeros clústeres, formados por la unión de varios equipos unidos entre sí mediante una red de alta velocidad, formando granjas de ordenadores.

El uso de este tipo de sistema distribuido le permite comportarse como si de un único equipo servidor se tratase. Sin embargo, cada equipo o nodo, realiza la misma tarea, para así, conseguir mediante la distribución de la carga entre todos ellos un mayor rendimiento del conjunto. Por otro lado, se consigue una alta disponibilidad del sistema en su conjunto de dos formas:

- Alta disponibilidad hardware: en el caso de fallo de uno de los nodos del sistema, el resto de los nodos pueden absorber la carga de trabajo del nodo que ha fallado.
- Alta disponibilidad software: en el caso de fallo de una de las aplicaciones, la misma puede ser arrancada en otro nodo distinto.

Esta solución, si bien es cierto que solventa los principales problemas, tiene ciertos elementos que complican su implantación para el público mayoritario.

Uno de los problemas que implican el uso de este tipo de soluciones es su alto coste de implantación. Al tratarse del despliegue de una serie de elementos hardware que tienen unas características específicas y de cierta calidad y rendimiento, se trata de componentes con un elevado coste económico, lo que complica enormemente su uso por un público muy amplio. También es necesario tener en cuenta que no se trata únicamente del coste de despliegue de los componentes, sino también su mantenimiento, ya que, al tratarse de componentes especiales, también lo es el entorno en el que tienen que estar ubicados, bien sea por seguridad, por climatización, consumo energético, etc.

Otro de los inconvenientes que tiene esta solución es el relacionado con la escalabilidad. Aunque el propio principio del clúster permite adaptar los recursos que se ofrecen para la realización de una tarea concreta, no podemos decir que esto implique una total escalabilidad del sistema. Esto es debido a que, dado que se trata de un sistema físico, cuando no está sometido a mucha carga de trabajo, existen recursos que se pueden estar desaprovechando, por el contrario, ante un aumento masivo de la carga de trabajo, los recursos del clúster son limitados, lo cual también limita la carga que este puede aceptar.

2.2 Situación actual

Desde la implantación de los primeros clústeres, se ha producido un avance significativo de las soluciones que se han ido desarrollando con este fin.

Actualmente hay una gran cantidad de entidades, tanto públicas como privadas, que tienen la necesidad de implementar una solución para poder ofrecer sus servicios al público o incluso para poder satisfacer necesidades internas de la propia organización. Sin embargo, no todas ellas pueden o desean realizar una inversión tan grande en implementar un clúster físico en propiedad.

Para solventar esta necesidad, desde hace unos años, se ha podido observar cómo han surgido empresas especializadas que ofrecen un nuevo tipo de servicio, el denominado cloud computing o IaaS (Infrastructure as a Service). Estas empresas, con una gran capacidad económica, invierten grandes cantidades de dinero y recursos en implantar grandes centros de datos a lo largo de todo el mundo, pero no únicamente para hacer un uso propio de dichos centros de datos, sino para ofrecer de forma temporal o de largo plazo de las soluciones que estos centros de datos pueden ofrecer.

Dicho de otro modo, existen grandes empresas que “levantan” grandes centros de datos para alquilar los recursos de estos o para ofrecer una virtualización de servicios a otras empresas u organizaciones.

Entre estas grandes empresas, las que más destacan por su cuota de mercado son: Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud e IBM Cloud. Teniendo una cuota de mercado del: 33%, 20%, 10%, 6% y 4% respectivamente en el tercer cuatrimestre de 2021 (Herranz, 2021). Como se puede observar, se trata de grandes compañías tecnológicas con un fuerte poder adquisitivo.

La forma de ofrecer las soluciones por parte de estas compañías viene dada, principalmente, por dos formas de pago por los servicios prestados:

- En forma de suscripción
- Pago por uso

En el mercado actual, aun cuando se trata de empresas que tienen cierta capacidad de inversión en IT, cada vez son más las empresas que, a pesar de poder realizar dicha inversión o ya haberla realizado, deciden migrar su infraestructura a estos proveedores cloud. Esto lleva a pensar que el futuro de este tipo de soluciones pasa, cada día más, por la elección de servicios cloud.

De todas las soluciones que se ofrecen actualmente en el mercado, para el desarrollo de este proyecto se ha elegido la opción de Alibaba Cloud, dado que se trata de una plataforma que no es muy conocida, pero tiene el potencial de ponerse al mismo nivel que el resto de grandes tecnológicos.

3. Análisis del problema

Durante este apartado se va a realizar un análisis del problema planteado. Para ello se va a realizar una implantación de un clúster Kubernetes en un entorno local, de esta forma se podrá entender más adelante los objetivos a perseguir en el entorno de la plataforma Alibaba Cloud.

3.1 Herramientas y definiciones

Para la realización de la implantación en un entorno local se va a hacer uso de una serie de herramientas que nos permitan simular el clúster a desarrollar. Para ello se va a contar con:

- Contenedor. Sistema de virtualización que se ejecuta a sobre el núcleo del sistema operativo, permitiendo ejecutar aplicaciones directamente sobre dicho espacio virtualizado de forma ágil y segura.
- VirtualBox como herramienta de virtualización. Permitirá crear máquinas virtuales que simularán los servidores que componen el clúster.
- Imagen del Sistema Operativo Red Hat en su versión 8.5.0 que serán el sistema operativo base de los servidores.
- MetalLB. Herramienta que permite realizar un balance de carga entre distintos servidores.
- Script en Python que permite realizar pruebas de carga sobre servidores web HTTP, realizando múltiples peticiones en serie.
- Kubernetes. Como herramienta que permite el despliegue de contenedores que se ofrecerán desde el clúster, así como su gestión.
- Docker. Como herramienta que permite el despliegue de aplicaciones dentro de los contenedores ofrecidos por Kubernetes

3.2 Despliegue en sistema local

En este apartado se lleva a cabo un pequeño despliegue de un clúster Kubernetes virtualizado sobre VirtualBox en máquinas virtuales. La virtualización del clúster se realiza sobre un equipo de forma local.

3.2.1 Creación de máquinas virtuales

Para poder llevar a cabo un despliegue en el sistema local se ha elegido el software de virtualización VirtualBox, dado que es uno de los principales softwares con este fin. Dado que se trata de un entorno local de pruebas se busca la forma más simple de implementarlo, por ello, se va a realizar en primer lugar, la creación de cuatro máquinas virtuales que tendrán asignados distintos roles:

- Una máquina virtual como *master*.
- Dos máquinas virtuales que actuarán como *workers*.
- Una máquina virtual que actuará como servidor de almacenamiento persistente.

Las máquinas virtuales contarán con la siguiente configuración:

	Master	**Worker**	**Almacenamiento persistente**
Sistema Operativo	Red Hat 8.50	Red Hat 8.50	Red Hat 8.50
Memoria principal	4 Gb	4 Gb	1,5 Gb
Almacenamiento	10 Gb	10 Gb	20 Gb
Tarjeta de red	Adaptador puente Intel PRO/1000 T Server	Adaptador puente Intel PRO/1000 T Server	Adaptador puente Intel PRO/1000 T Server

Tabla 1. Configuración de las máquinas virtuales

3.2.2 Despliegue del clúster

Una vez creadas las máquinas virtuales, se procede al despliegue del clúster en sí, que estará formado por Kubernetes como orquestador de contenedores, Docker como herramienta de gestión de contenedores, MetalLB como balanceador de carga y NFS como protocolo para establecer la conexión con el sistema de almacenamiento compartido.

En primer lugar, se procede a la implantación del *master*, para ello, una vez instalado el sistema operativo base, es necesario añadir el repositorio desde el cual se procederá a descargar e instalar los paquetes necesarios para Kuberntes (versión 7 en este caso).

```
sudo tee /etc/yum.repos.d/kubernetes.repo<<EOF
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
```

Ilustración 1. Despliegue de Kubernetes en local I

Una vez actualizados los repositorios se procede a instalar los paquetes que compondrán Kubernetes.

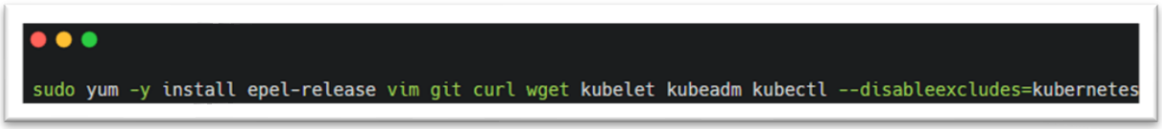


Ilustración 2. Despliegue de Kubernetes en local III

A continuación, se procede a rebajar los mecanismos de seguridad incorporados en Linux, para ello, se cambia el modo de funcionamiento del módulo de seguridad del *kernel* de Linux (Security-Enhaced Linux o SELinux), y se deshabilita la memoria de intercambio *swap* se activan los módulos *overlay* y *br_netfilter* para permitir que se realice un *forwarding* IPv4 y que el módulo *iptables* pueda observar el tráfico *bridge*.

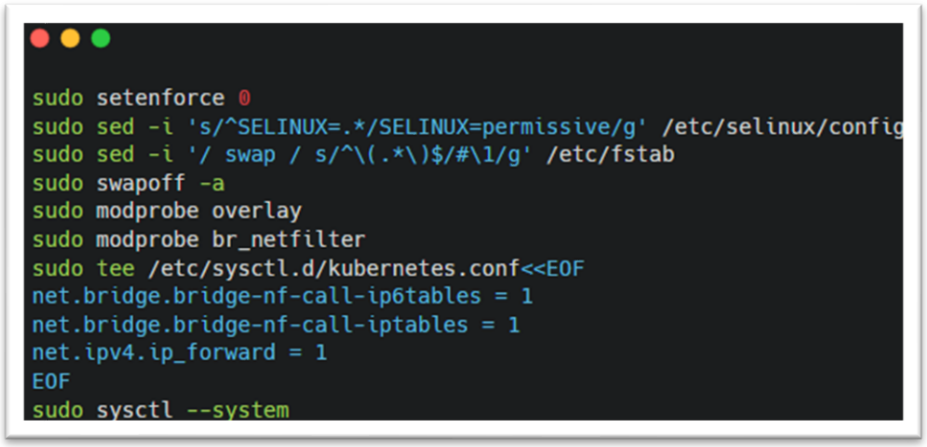


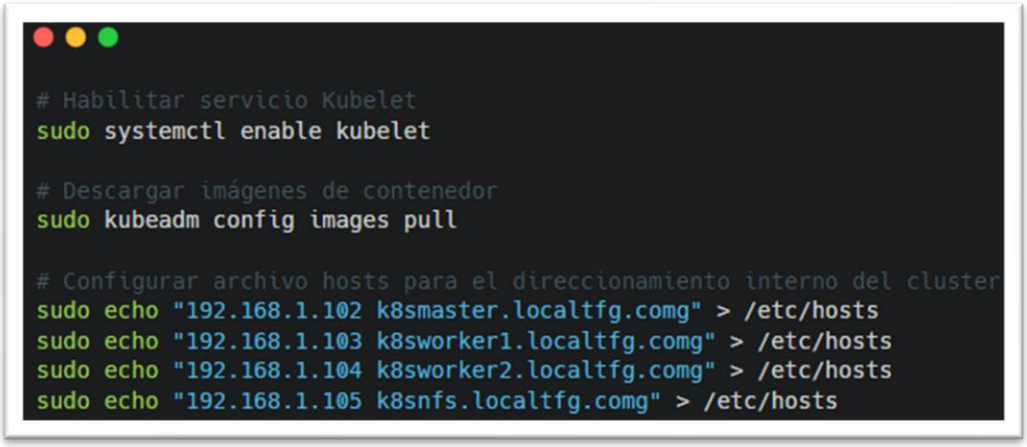
Ilustración 3. Despliegue de Kubernetes en local IV

Kubernetes necesita de un *runtime* para la ejecución de los contenedores, de entre los varios existentes, se hace uso de uno de los más conocidos, Docker. Por simplicidad, se procede a deshabilitar el firewall del sistema.

```
# Instalar paquetes Docker
sudo yum install -y yum-utils device-mapper-persistent-data lvm2
sudo yum-config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
sudo yum install docker-ce docker-ce-cli containerd.io

# Crear directorios necesarios para Docker
sudo mkdir /etc/docker
sudo mkdir -p /etc/systemd/system/docker.service.d

# Inicializar y habilitar servicios
sudo systemctl daemon-reload
sudo systemctl restart docker
sudo systemctl enable docker

# Deshabilitar firewall
sudo systemctl disable --now firewalld
```

Ilustración 4. Despliegue de Kubernetes en local V

A continuación, es necesario habilitar el servicio Kubelet, descargar las imágenes de contenedor y configurar el archivo *hosts* para el enrutamiento interno del clúster.



```
# Habilitar servicio Kubelet
sudo systemctl enable kubelet

# Descargar imágenes de contenedor
sudo kubeadm config images pull

# Configurar archivo hosts para el direccionamiento interno del cluster
sudo echo "192.168.1.102 k8smaster.localtfg.comg" > /etc/hosts
sudo echo "192.168.1.103 k8sworker1.localtfg.comg" > /etc/hosts
sudo echo "192.168.1.104 k8sworker2.localtfg.comg" > /etc/hosts
sudo echo "192.168.1.105 k8snfs.localtfg.comg" > /etc/hosts
```

Ilustración 5. Despliegue de Kubernetes en local VI

Para el resto de los nodos *worker* se deben realizar los mismos pasos descritos anteriormente. A continuación, es posible inicializar el nodo *master*.

Ilustración 6. Despliegue de Kubernetes en local VII

Una vez inicializado el nodo *master*, hay que tener en cuenta la salida que proporciona, pues en la misma se proporcionan los pasos a seguir para poder poner en funcionamiento el clúster. También se indican los pasos para poder añadir los nodos *worker* más adelante, se toma nota de esta.

Ilustración 7. Despliegue de Kubernetes en local VIII

Internamente, Kubernetes necesita de un plugin que gestione la red interna del clúster, se utiliza uno de los más utilizados y con un mayor soporte por parte de la comunidad, Calico.

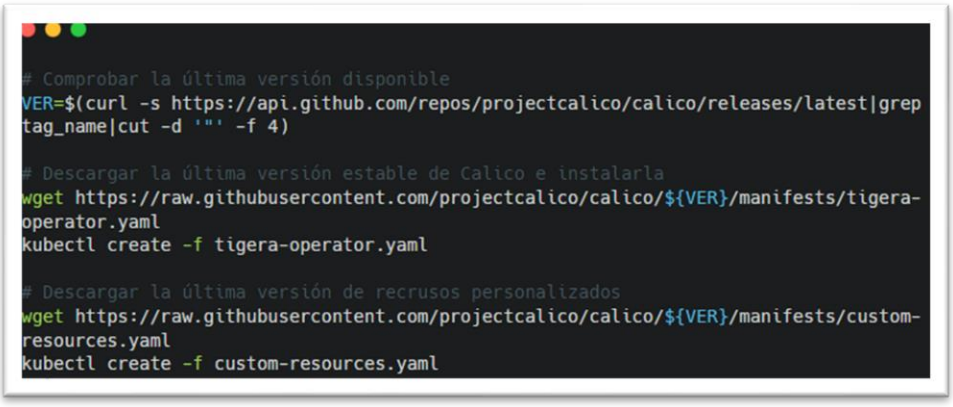


Ilustración 8. Despliegue de Kubernetes en local IX

Se comprueba que los *pods* correspondientes a Calico están funcionando correctamente.

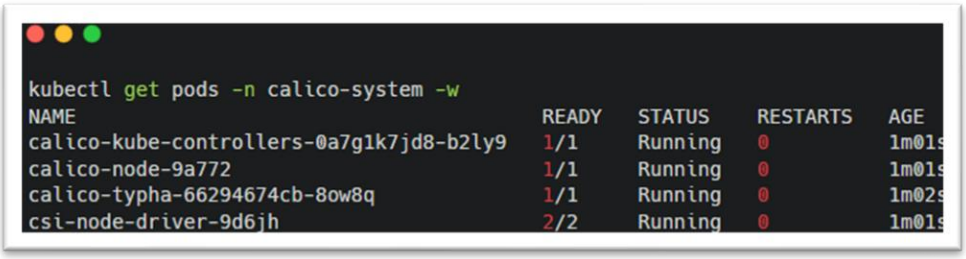


Ilustración 9. Despliegue de Kubernetes en local X

Tal como se indica anteriormente, para añadir los nodos *worker*, es necesario ejecutar el siguiente comando en cada uno de los *worker*.

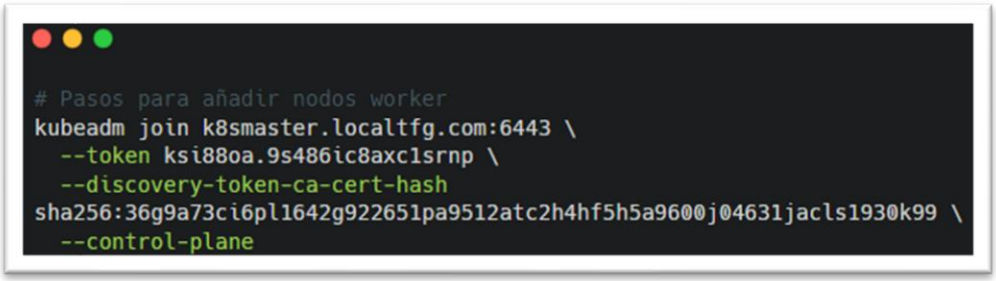


Ilustración 10. Despliegue de Kubernetes en local XI

Ya es posible comprobar que el clúster se ha iniciado correctamente.

Ilustración 11. Despliegue de Kubernetes en local XII

Con los pasos descritos anteriormente ya se dispone de un clúster Kubernetes básico en el que desplegar aplicaciones, sin embargo, son necesarios más elementos para poder disponer de un cierto nivel de disponibilidad en el mismo, entre ellos un balanceador de carga y almacenamiento persistente, ya que, en caso contrario, al destruirse los *pods* que forman parte del servicio ofertado se perderá la información de estos o incluso mostrarán información distinta.

En primer lugar, se propone poner en funcionamiento el servidor de almacenamiento persistente. Para poder ofrecer cierto grado de disponibilidad es necesario que el almacenamiento no se encuentre en las mismas máquinas que ofrecen el servicio Kubernetes.

En el servidor propuesto en la preparación del clúster se instalan y preparan los paquetes y servicios necesarios para el funcionamiento del almacenamiento compartido.

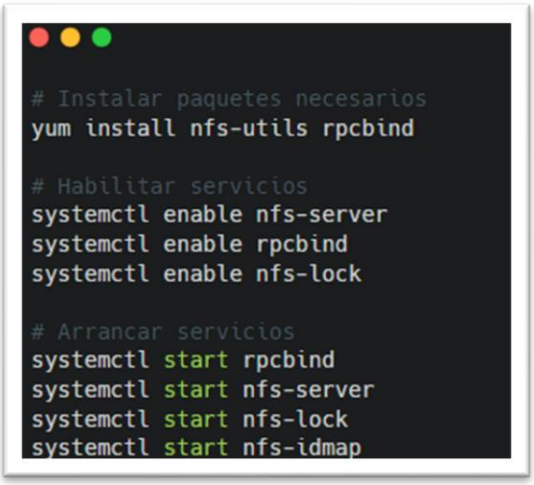


Ilustración 12. Despliegue de Kubernetes en local XIII

A continuación, se crean los directorios que van a ser compartidos y se añaden al fichero */etc/exports* con las opciones que se consideren oportunas, por simplicidad, en este caso se crea un directorio en la raíz y es compartido con permiso de escritura y lectura por todos los clientes.

Ilustración 13. Despliegue de Kubernetes en local XIV

Dado que los clientes que van a mapear el directorio compartido van a ser los servicios ofrecidos dentro del clúster, estos van a ser usado por los *pods*, por lo tanto, serán montados de forma dinámica, gracias a la definición del servicio en los archivos de despliegue YAML.

Para poder simular el uso de un balanceador de carga como el ofrecido por los proveedores cloud, se va a hacer uso del balanceador MetalLB para realizar dichas funciones. Para ello, es necesario crear un espacio de nombres que contendrá el balanceador de carga, se crea un archivo YAML para poder aplicarlo en el clúster.

```
apiVersion: v1
kind: Namespace
metadata:
 name: metallb-system
 labels:
 app: metallb
```

Ilustración 14. Despliegue de Kubernetes en local XV

Se aplica el archivo YAML que contiene el balanceador de carga a partir de su manifiesto original.

```
# Aplicar el balanceador de carga en el cluster
kubectl apply -f https://raw.githubusercontent.com/google/metallb
/v0.12.0/manifests/metallb.yaml
```

Ilustración 15. Despliegue de Kubernetes en local XVI

A partir de un fichero YAML se definen las especificaciones del balanceador de carga, como el protocolo utilizado y el rango de direcciones IP que conforman el *pool* de direccionamiento.

```
apiVersion: v1
kind: ConfigMap
metadata:
 namespace: metallb-system
 name: config
data:
 config: |
 address-pools:
 - name: my-ip-space
 protocol: layer2
 addresses:
 - 192.168.2.240-192.168.2.250
```

Ilustración 16. Despliegue de Kubernetes en local XVII

Realizadas estas acciones ya está disponible el clúster para ofrecer servicios-

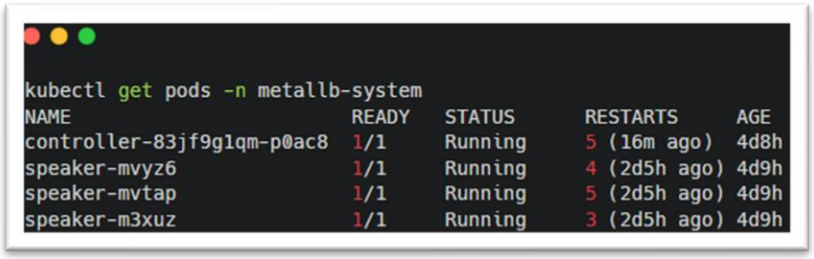


Ilustración 17. Despliegue de Kubernetes en local XVIII

De la misma forma que se ha realizado el despliegue del clúster Kubernetes de forma local, es posible realizar el despliegue de un clúster en máquinas de una plataforma cloud.

3.2.3 Pruebas de despliegue local

Una vez configurado el clúster se procede a realizar una serie de pruebas para comprobar que el funcionamiento de este es correcto. Mediante el uso de ficheros YAML se procede a desplegar un servicio HTML con el que se realizan las pruebas de funcionamiento descritas en el apartado siguiente. El detalle del fichero YAML se encuentra en el Anexo A.

Para ello, se utiliza el servicio HTML implantado anteriormente y, aplicando una serie de pruebas se observa cómo reacciona el clúster a los cambios y cómo se adapta ante los mismos para ofrecer un servicio altamente disponible.

Objetivos por conseguir durante la ejecución de las pruebas:

- El servidor HTML trabaja en distintos *pods*.
- Se utiliza el almacenamiento persistente definido en el servidor NFS.
- Kubernetes aplica auto escalado horizontal en los *pods* del servicio según necesidades.
- Ante la eliminación de algún *pod*, se sigue ofreciendo el servicio mediante el resto de *pods*.

Otro objetivo interesante para comprobar la alta disponibilidad del clúster es el auto escalado del propio clúster, sin embargo, al no estar realizando las pruebas en un entorno cloud, sino con máquinas virtuales, no es posible conseguir este objetivo.

En el servidor NFS configurado anteriormente se ha dispuesto una página web (HTML + PHP) que se sirve a los clientes, mediante la cual se muestra la dirección IP del *pod* que está sirviendo la página en cada momento.

En el clúster Kubernetes se disponen de un fichero YAML denominado persistentVolume.yaml para poder conectar con el servidor NFS, en dicho fichero se configuran las opciones de montaje, así como la ruta en la que se comparte y el servidor que lo comparte. Mediante otro fichero llamado persistentVolumeClaim.yaml se configura la petición, mediante la cual, la aplicación desplegada hará la petición del volumen anteriormente creado.

A continuación, mediante el fichero nginxDeployment.yaml se procede a crear el despliegue de la aplicación HTTP, la cual cuenta con una imagen NGINX para funcionar con PHP, se asigna el *volumen claim* y se define la ruta de los *pods* donde se procede al montaje del volumen persistente.

Finalmente, se configura el servicio que hará las veces de balanceador de carga y definirá el puerto que conecta la aplicación con el exterior, esto se realiza mediante un fichero llamado nginxService.yaml

Ilustración 18. Página servida por un pod con la IP 192.168.26.69

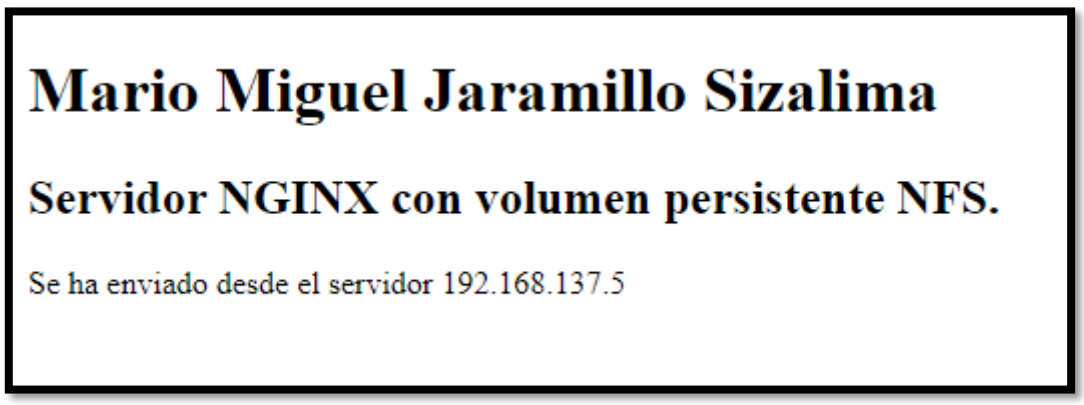


Ilustración 19. Página servida por un pod con la IP 192.168.137.5

Con esto, se comprueba que el primer objetivo se ha cumplido, pues diferentes peticiones son servidas por distintos *pods* y la página servida es la almacenada en el servidor NFS.

```
[root@k8s-cluster ~]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-789c5d5f97-cb2xw   1/1     Running   0          162m
nginx-deployment-789c5d5f97-gjgvw   1/1     Running   0          162m
```

Ilustración 20. Pods creados tras el despliegue de la aplicación

A continuación, se configura el auto escalado horizontal, que permitirá que al aumentar la carga de trabajo de la CPU por encima del 10% (por simplicidad), se creen más réplicas para abastecer la demanda del despliegue. Para configurarlo, se hace uso del fichero horizontalPodAutoscaler.yaml

```
[root@k8s-cluster ~]# kubectl get hpa
NAME        REFERENCE                     TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
nginx-hpa   Deployment/nginx-deployment   0%/10%    2         5         2          8m23s
```

Ilustración 21. HPA configurado

Para simular una alta carga de trabajo, se hace uso de un script en Python que realizará múltiples conexiones a la aplicación.

```python
import http.client

while (True):
    connection = http.client.HTTPConnection('192.168.1.139', 30826, timeout=10)
    connection.request("GET", "/")
    response = connection.getresponse()
    print("Status: {} and reason: {}".format(response.status, response.reason))
```

Ilustración 22. Script Python para peticiones HTTP

Al aumentar la carga de trabajo, es posible observar cómo se crean más réplicas. Por otro lado, al disminuir la carga de trabajo, y pasado un tiempo prudencial, se destruyen réplicas hasta volver al estado original del despliegue. De esta forma, se pueden dar por cumplidos los objetivos de auto escalado horizontal y prestación del servicio ante la eliminación de pods.

```
[root@k8s-cluster ~]# kubectl get hpa -w
NAME        REFERENCE                     TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
nginx-hpa   Deployment/nginx-deployment   0%/10%     2         5         2          10m
nginx-hpa   Deployment/nginx-deployment   0%/10%     2         5         2          10m
nginx-hpa   Deployment/nginx-deployment   2%/10%     2         5         2          11m
nginx-hpa   Deployment/nginx-deployment   8%/10%     2         5         2          11m
nginx-hpa   Deployment/nginx-deployment   11%/10%    2         5         2          11m
nginx-hpa   Deployment/nginx-deployment   8%/10%     2         5         3          11m
nginx-hpa   Deployment/nginx-deployment   9%/10%     2         5         3          12m
nginx-hpa   Deployment/nginx-deployment   18%/10%    2         5         3          12m
nginx-hpa   Deployment/nginx-deployment   15%/10%    2         5         5          12m
nginx-hpa   Deployment/nginx-deployment   13%/10%    2         5         5          12m
```

Ilustración 23. Creación de nuevas réplicas mediante HPA

```
[root@k8s-cluster ~]# kubectl get hpa -w
NAME        REFERENCE                     TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
nginx-hpa   Deployment/nginx-deployment   0%/10%     2         5         5          18m
nginx-hpa   Deployment/nginx-deployment   0%/10%     2         5         2          18m
```

Ilustración 24. Eliminación de réplicas mediante HPA

La implementación en local de esta aplicación es un claro ejemplo de cómo se va a llevar a cabo el despliegue de una aplicación igual en un clúster Kubernetes en la plataforma Alibaba Cloud.

4. Alibaba Cloud

A lo largo de este apartado se procede a realizar un análisis de la plataforma de servicios la cual es objeto de estudio en este trabajo. Para ello se procede a realizar una comparativa con sus principales competidores, un análisis de las opciones disponibles, su facilidad de uso y una comparativa económica respecto a sus principales rivales en el mercado.

4.1 Análisis del mercado actual

A lo largo de los últimos años se ha podido observar un enorme crecimiento en la demanda de soluciones TIC (Horcajo, 2021) tanto a nivel de empresarial como social. Para poder hacer frente a esta elevada demanda se ha podido observar, a su vez, un crecimiento en las plataformas que ofrecen servicios cloud que intentan innovar y a su vez proporcionar una mayor cantidad de soluciones a las necesidades que se identifican.

Al hablar de proveedores de servicios cloud, es casi de obligado cumplimiento mencionar las tres grandes plataformas que dominan el mercado actual, ya que entre las tres copan más del 60% del mercado actual. Estas tres grandes corporaciones son (Richter, 2023):

- **Amazon Web Services (AWS).** Perteneciente a la multinacional Amazon y con un 32% de cuota de mercado.
- **Microsoft Azure.** Perteneciente a la multinacional Microsoft y con un cerca de un 23% de cuota de mercado.
- **Google Cloud.** De la multinacional Google y con un 10% de cuota de mercado.

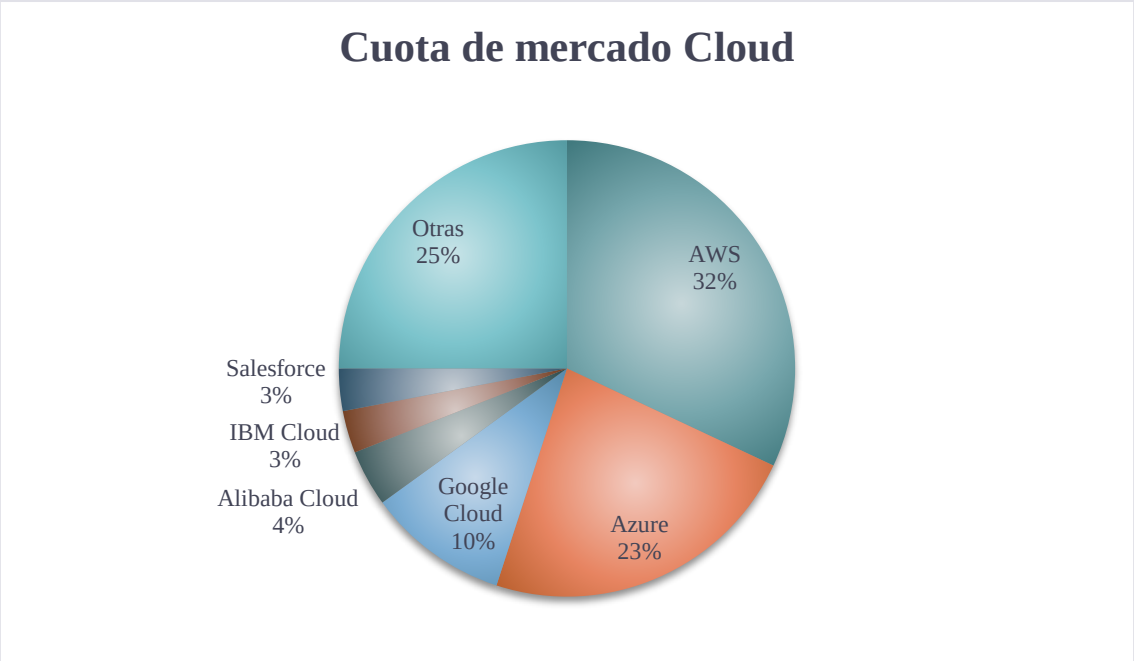


Ilustración 25. Cuota de mercado Cloud

A pesar de que estas tres grandes plataformas se disputan la mayoría del mercado, también se dispone de otras alternativas en crecimiento, que poco a poco se están dando a conocer y están obteniendo su respectiva cuota de mercado.

Una de estas plataformas emergentes es Alibaba Cloud, que actualmente se disputa el 4% del mercado con otras plataformas como IBM Cloud o Salesforce. A pesar de ser una plataforma en crecimiento, se trata de una compañía fundada en septiembre de 2009, casi al mismo tiempo que Microsoft Azure o Google Cloud.

Esta compañía, respaldada por el gigante tecnológico Alibaba Group, no ha tenido una presencia relevante en el mercado mundial debido, entre otros factores, al desconocimiento de la plataforma y al tratarse de una compañía procedente de un país con el que existen ciertas tensiones políticas. A pesar de esto, cada vez cuenta con una mayor presencia y visibilidad.

4.2 Alibaba y AWS

Tal como se ha mostrado en el apartado anterior, actualmente el mayor proveedor de servicios cloud a nivel mundial es Amazon Web Services (Amazon Web Services, 2023), con más de un 30% de cuota de mercado. En comparación, el proveedor Alibaba Cloud, únicamente dispone de una décima parte que el gigante de Amazon.

Ante esta enorme diferencia de tamaño, es previsible encontrar diferencias en otros aspectos, tales como la oferta de servicios, coste de estos, implantación y expansión con respecto a Amazon Web Services. Para abordar esto, se va a realizar una comparativa con Amazon Web Services, centrándose especialmente en la diferencia de los servicios que ofrecen ambas compañías con respecto a aspectos principales como los mencionados anteriormente y que pueden ser de especial relevancia a la hora de tomar la decisión del despliegue de un clúster Kubernetes en una u otra plataforma.

A la hora de decantarse por una u otra plataforma de servicios cloud. Es necesario tener en cuenta las zonas de disponibilidad de las que dispone dicho proveedor y su ubicación física, ya que estas indican la proximidad desde la que se prestan los servicios a los usuarios finales y por lo tanto influenciando en la calidad de los servicios ofertados.

En el caso de Alibaba Cloud, el proveedor distingue principalmente dos regiones, centro de datos que se encuentran en China continental; y aquellos que se encuentran fuera de China continental. Una vez hecha esta distinción, se debe hacer una mención especial a la localización de los centros de datos que se encuentran fuera de China continental, ya que la mayor parte de estos se encuentran situados en el continente asiático, contando únicamente con dos centros de datos en Europa (Londres y Frankfurt), dos en América (Los Ángeles y Silicon Valley) y dos en Oriente Próximo (Riad y Dubái), ofreciendo cada centro de datos entre una y tres zonas de disponibilidad. Por su lado, dentro de la región de China continental se cuenta con 13 centros con hasta once zonas de disponibilidad (Alibaba Cloud, 2023).

Si se observa la distribución de los centros de datos de Amazon Web Services, su enfoque está más distribuido a nivel global. Así pues, en América cuentan con ocho centros de datos (incluido uno en América del Sur), ocho centros en Europa, dos en Oriente Próximo, uno en Oceanía y 10 centros en Asia (incluidos tres centros de datos dentro de China). Sus centros de datos cuentan con hasta cinco zonas de disponibilidad en América y hasta cuatro zonas para el resto de los centros de datos (Amazon Web Services, 2023).

A continuación, se observan las distintas soluciones que ofrecen tanto la plataforma de Amazon como la de Alibaba. Para este caso, se hace referencia únicamente a aquellas soluciones destinadas a satisfacer la necesidad del despliegue de un clúster altamente disponible basado en contenedores, sin entrar a distinguir las opciones disponibles para otro tipo de necesidades. Para ello se realizará una comparación entre los servicios Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Kubernetes Services (Amazon EKS) por parte de Amazon Web Services; por parte de Alibaba Cloud se encuentran los servicios Alibaba Cloud Container Service for Kubernetes (ACK) y Alibaba Elastic Compute Service (ECS).

A la hora de desplegar un clúster Kubernetes en los servicios de Amazon Web Services, se proporcionan principalmente dos servicios dependiendo de las necesidades que se necesiten cubrir en el proyecto.

Por un lado, haciendo uso del servicio Amazon EKS, se dispone de la posibilidad de realizar un despliegue de forma que es posible utilizar el propio entorno de la nube de Amazon o hacer uso de una infraestructura propia o una infraestructura híbrida. En caso de hacer un uso de la nube de Amazon, es el propio servicio de Amazon el que se encarga de administrar los nodos *master*, así como del *control plane* de Kubernetes, permitiendo al cliente centrarse en exclusiva de los nodos *worker* y en el despliegue de las aplicaciones. También dispone de la posibilidad de no hacer uso de nodos *worker*, para ello, Amazon propone el uso de su servicio Amazon Web Services Fargate de forma que se despliegan aplicaciones *serverless*, contabilizando únicamente el tiempo y carga de ejecución, sin necesidad de disponer de una infraestructura de forma fija. Esto permite el despliegue de un clúster de forma más fácil y rápida, sin embargo, se elimina la posibilidad de tener un control total sobre las características con las que se desea que cuente el clúster, pues los nodos encargados de la planificación y despliegue de contenedores están bajo el control del servicio del proveedor (Amazon EKS, 2023).

Por otro lado, en caso de necesitar tener un control total sobre toda la infraestructura del clúster Kubernetes, es posible hacer uso del servicio Amazon EC2, mediante el cual se pueden contratar instancias según las necesidades a cubrir. Es sobre estas instancias contratadas sobre las cuales se ha de realizar el despliegue de todos los nodos con los que se desea que cuente el clúster, desde los nodos *master* a los *worker*. Hacer uso de este tipo de servicios requiere una planificación previa de forma precisa, ya que, según las instancias contratadas, su intensidad de trabajo y sus características, el precio de estas puede oscilar de forma notable, sin embargo, permite tener un control total sobre el clúster, pudiendo definir de forma precisa cómo se desea que se comporte en cada momento (Amazon EC2, 2023).

Por su parte, Alibaba Cloud también ofrece distintas soluciones para el despliegue de un clúster Kubernetes, que también permiten adaptarse a las necesidades concretas de cada tipo de usuario u organización.

De esta forma, gracias al servicio Elastic Compute Service (ECS), se pueden crear instancias vacías en las cuales realizar el despliegue de forma manual de un clúster Kubernetes, permitiendo tener un control total y absoluto de la administración de este, de forma similar al servicio ofrecido por Amazon EC2 (Alibaba ECS, 2023). Mediante el uso de este servicio, al igual que en Amazon Web Services, es posible realizar un despliegue mediante instancias elásticas individuales haciendo uso de herramientas de despliegue automatizadas como Terraform (Alibaba Terraform, 2019). Sin embargo, no todas las herramientas están disponibles en esta plataforma, por ejemplo, la herramienta de despliegues y administración kOps no proporciona soporte para esta plataforma debido a la falta de colaboradores (kOps 1.20 Deprecations, 2020).

Alternativamente a esta solución, Alibaba cuenta con otro servicio más centrado y adaptado al despliegue de un clúster Kubernetes, se trata de Alibaba Cloud Container Service for Kubernetes o Alibaba ACK. Dentro de este servicio se distinguen tres alternativas que ofrece una adaptabilidad a las necesidades del cliente. De forma análoga a las ofertas de Amazon EKS, se permite la creación de un clúster de forma que es el propio servicio de Alibaba el que controla y administra los nodos *master*, centrándose el usuario únicamente en la creación de los nodos *worker* y el resto de la infraestructura del clúster, lo que dentro del servicio se denomina *Managed Kubernetes*. Esta opción permite una relativa facilidad de uso, un coste más bajo y alta disponibilidad, ya que la creación y planificación de los contenedores viene dada por el propio servicio. Otra alternativa que ofrece es la denominada *Dedicated Kubernetes*, en la cual es el usuario el que tiene el control de todos los nodos que constituyen el clúster, desde los nodos *master* a los nodos *worker*, así como toda la infraestructura del clúster, según se indica por la propia Alibaba, esta solución está pensada para aquellas organizaciones que tienen en mente realizar una migración de servicios *on premise* a la nube y para la transformación empresarial digital. La tercera alternativa que ofrece Alibaba ACK es, al igual que Amazon EKS, la posibilidad de hacer uso de servicios contenerizados sin necesidad de crear, propiamente dicho, un clúster dedicado. De forma que se permite la ejecución de tareas por lotes, actividades de escalado rápido o test CI/CD. Esta alternativa permite que se haga uso de la capacidad *cloud* del servicio de contenedores de Alibaba, pero facturando por el tiempo de uso de cada actividad o ejecución, gracias a esto, es posible realizar la ejecución de dichas tareas de forma más económica que si se realizase el despliegue de un clúster dedicado en caso de no realizarse de forma continua (Alibaba ACK, 2023).

Cabe destacar que ambas opciones, tanto Amazon como Alibaba, permiten realizar un despliegue híbrido de un clúster Kubernetes. En ambas opciones la propuesta es muy similar, pues ambas proponen que lo servidores físicos en los que esté alojado el clúster Kuberntes estén en posesión de la organización, haciéndose cargo esta de su administración y características. En ambas opciones el servicio ofertado es la administración y gestión del sistema Kubernetes por parte del servicio de cada proveedor.

Un clúster no está compuesto únicamente por las instancias sobre las que opera la plataforma Kubernetes ni por la propia plataforma Kubernetes, también debe contar con otros elementos igual de importantes para poder funcionar de forma correcta, algunos de estos elementos pueden ser balanceadores de carga, almacenamiento compartido y persistente, infraestructura de red, etc. A continuación, se describe de qué forma trata cada proveedor de servicios cloud estos elementos y qué alternativas ofrece cada uno, en ambos casos se procede a realizar un análisis teniendo como punto de partida la creación de un clúster altamente disponible con componentes básicos, de forma análoga a como se ha desplegado el clúster de forma local en el apartado 3.2.2.

En el caso de Amazon, se dispone de servicios dedicados a cada componente que necesita el clúster para estar operativo.

Para el caso de un balanceador de carga, cuenta con el denominado Elastic Load Balancing, que puede ser integrado para realizar dichas tareas sin necesidad de implementar uno de forma manual en una instancia separada, se distinguen cuatro tipos de balanceadores: balanceador de carga de aplicaciones (como indica su nombre, permite realizar el balanceo de carga dependiendo de la aplicación a la que va dirigida el tráfico, haciendo uso de la capa 7 del modelo OSI), balanceador de carga de red (en este caso, el balanceo se realiza haciendo uso de la capa 4 del modelo OSI, ,dependiendo del protocolo de red utilizado), balanceador de carga Gateway (permite hacer uso del Gateway para realizar un escalado de las aplicaciones en función del tráfico dirigido a cada red virtual) y balanceador de carga clásico (Amazon Networking, 2023).

Para el almacenamiento, Amazon ofrece su servicio Amazon Simple Storage Service (Amazon S3), mediante el cual se puede implementar un almacenamiento básico de datos,, aunque también cuenta con otros servicios más dedicados a la implementación de un clúster altamente disponible que deben ser puestos de manifiesto, estos son Amazon Elastic Block Storage (Amazon EBS) que permite utilizar almacenamiento en bloque de forma altamente disponible y compartido por todas aquellas instancias que lo necesiten; y Amazon Elastic File System (Amazon EFS) que permite crear un almacenamiento de forma que se adapta a la cantidad de datos que almacena de forma automática, dependiendo de su crecimiento o decrecimiento,

optimizando de esta forma el coste de dicho almacenamiento, esta opción también cuenta con alta disponibilidad y persistencia en el tiempo (Amazon Storage, 2023).

Respecto al resto de los elementos de red, Amazon ofrece la creación de redes VPC mediante Amazon Virtual Private Cloud (Amazon VPC), gracias a la cual es posible la creación de una red privada virtual mediante la cual se pueden interconectar las distintas instancias y los distintos servicios que se ejecutarán en el clúster.

Por parte de Alibaba, también ofrece una serie de servicios, tanto de forma individual para otros usos, así como en la propia implantación del clúster Kubernetes.

En lo que respecta a los balanceadores de carga, Alibaba ofrece el servicio de forma similar a la oferta por parte de Amazon. En este caso ofrece tres tipos de balanceadores de carga, entre los que encontramos el caso de balanceador de carga de red, balanceador de carga de aplicaciones y el balanceador de carga clásico, contando con las mismas características que los soportados por Amazon. Cabe destacar que todos los balanceadores de carga ofrecen auto escalado dependiendo de las necesidades del tráfico de red al que se enfrenten, lo cual es deseado a la hora de ofrecer un servicio altamente disponible. Fuera de estas opciones, en ambos proveedores se encuentran soluciones muy similares o incluso idénticas para ofertar soporte de red (Alibaba Networking, 2023).

De forma similar a los productos ofrecidos por Amazon, Alibaba ofrece a su vez tres soluciones de almacenamiento cloud para la implantación del clúster buscado. En este caso, para obtener un almacenamiento simple, ofrecen el denominado Apsara File NAS Storage, el cual ya ofrece características buscadas para implantar un clúster altamente disponible. Estas características ofrecidas son: 99,999999999% de fiabilidad, disponibilidad de un 99,99%, escalabilidad según necesidades incluso a nivel horizontal, seguridad y un bajo coste. Otra de las opciones que ofrece Alibaba para el almacenamiento es el denominado Object Storage Service (OSS), mediante el cual, gracias a su integración con el servicio CDN permite obtener un espacio de almacenamiento de objetos escalable, flexible y fiable, según sus propios datos, especialmente útil para distribución de contenido, backups y recuperación, hosting web o almacenamiento virtual. La tercera opción ofrecida por Alibaba es Block Storage, que al igual que en el caso de Amazon, permite el almacenamiento de datos en bloque de forma que se obtiene una baja latencia y un alto rendimiento (Alibaba Storage, 2023).

En lo que respecta el resto de los elementos de red necesarios para el correcto funcionamiento del clúster, Alibaba también ofrece un servicio de red privada virtual (Alibaba VPC) mediante la cual permite la creación de subredes virtuales mediante las que interconectar el resto de los

componentes, así como la creación de otros elementos necesarios como switches, routers o incluso VPN para un acceso más seguro.

Tanto Alibaba como Amazon cuentan con un servicio para ofrecer direcciones IP elásticas mediante las cual acceder a los servicios del clúster a través de Internet mediante el mapeo con una dirección IP pública. En ambos casos es posible utilizar una dirección IP pública propia.

	Amazon Cloud Services	**Alibaba Cloud**
Áreas geográficas	Hong Kong, Melbourne, Mumbai, Seúl, Singapur, Sídney, Tokio, Osaka, Pekín, Ningxia, Yakarta, Hyderbad, Norte de Virginia, Ohio, Norte de California, Oregón, EEUU Este, EEUU Oeste, EEUU Central, São Paulo, Bahréin, Ciudad del Cabo, Frankfurt, Irlanda, Londres, Milán, París, España, Estocolmo, Zúrich, EUA	Pertenecientes a China: Qingdao, Beijing, Zhangiako Hohhot, Ulanqab, Hangzhou, Shanghái, Nanjing, Fuzhou, Shenzhen, Heyuan, Guangzhou, Chengdu, Hong Kong. Fuera de China: Singapur, Sídney, Kuala Lumpur, Yakarta, Manila, Bangkok, Mumbai, Tokio, Seúl, Silicon Valley, Virginia, Frankfurt, Londres, Dubái, Riad
Zonas de disponibilidad	99 entre todas las regiones	72 entre todas las regiones
Servicios Kubernetes ofrecidos	• Amazon Elastic Compute Cloud (Amazon EC2) • Amazon Elastic Kubernetes Service (Amazon EKS)	• Alibaba Elastic Compute Service (Alibaba ECS) • Alibaba Container Service for Kubernetes (Alibaba ACK)
Tipos de gestión	• Gestión total (nodos *master* y *worker*) • Gestión, únicamente, de nodos *worker*	• Gestión total (nodos *master* y *worker*) • Gestión, únicamente, de nodos *worker*
Balanceadores de carga	• De aplicación • De red • Gateway • Clásico	• De aplicación • De red • Clásico
Red virtual privada (VPC)	• Elastic IP • Alibaba VPC ○ Subred ○ Switch ○ Router ○ Enrutamiento	• Elastic IP • Alibaba VPC ○ Subred ○ Switch ○ Router ○ Enrutamiento
Almacenamiento	• Amazon Simple Storage Service • Amazon Elastic Block Storage • Amazon Elastic File System	• Apsara File NAS Storage • Alibaba Block Storage Service • Alibaba Object Storage Service

Tabla 2. Amazon Cloud Services VS Alibaba Cloud

4.3 Microsoft Azure y Google Cloud

Si bien Amazon Web Services está catalogado actualmente como el mayor proveedor de servicios cloud a nivel global, y Alibaba Cloud es el objeto de estudio del presente Trabajo de Fin de Grado, no hay que olvidar que existen otros proveedores que también cuentan con una gran cuota de mercado, siendo la misma muy superior a la de Alibaba Cloud. Por ello, es necesario realizar una mención a estos dos proveedores que tienen una especial relevancia en el mercado actual. De esta forma, se procede a realizar un breve análisis de la implantación de un clúster Kubernetes en Microsoft Azure y en Google Cloud de forma nativa.

Empezando por Microsoft Azure, sin duda se trata de uno de los proveedores destacados gracias a que forma parte de la familia Microsoft, una de las grandes tecnológicas a nivel mundial y con un gran recorrido a lo largo del tiempo. Microsoft Azure ofrece el servicio Azure Kubernetes Service (AKS) (Microsoft AKS, 2023) para la implantación de clústeres Kubernetes en su plataforma de una forma fácil y rápida en sus más de cuarenta regiones a lo largo del globo.

A la hora de crear un clúster Kubernetes, ofrece la posibilidad de crearlo mediante el dimensionamiento de este en hasta cinco posibilidades (Estándar, Desarrollo/Test, Optimizado para costes, Procesamiento por lotes, Acceso protegido), mediante este dimensionamiento, es posible crear un clúster al que se le asignan una cantidad de nodos con unas características prefijadas dependiendo del uso que se le vaya a dar. Aunque es posible cambiar algunas características de los nodos, es necesario dejar claro que, al contrario que en el caso de Alibaba, no es posible tener acceso a todos los nodos, ya que el nodo *master* está restringido, de forma que su administración está establecida por la plataforma del proveedor, únicamente es posible administrar los nodos *worker*.

Si bien es cierto que esto puede suponer un reticente a la hora de gestionar completamente un servicio de este calibre, también es necesario poner de manifiesto otras propiedades con las que cuenta esta plataforma. Al pertenecer a la familia Microsoft, tiene una ventaja muy superior al resto de plataformas, su integración con el resto de los servicios ofrecidos por la compañía. De esta forma, resulta sencillo integrar otra serie de servicios empresariales de uno de los mayores desarrolladores de sistemas operativos, entre ellos Microsoft Active Directory, mediante el cual es posible asignar roles de seguridad a los usuarios, permitiendo un mayor control sobre el acceso al clúster y a las acciones que pueden ser llevadas a cabo dentro del mismo. Otro de los componentes con los que también hay que contar es con el uso de una de las mayores plataformas para el almacenamiento y tratamiento de código, como es GitHub, recientemente

adquirido por el gigante tecnológico, permite su integración de forma sencilla; de la misma forma procede con el editor de código Microsoft Visual Studio Code y Microsoft Visual Studio.

Al empezar a utilizar esta plataforma, se cuenta con un crédito inicial gratuito de $200 para emplear dentro de la plataforma durante el plazo de treinta días, además de más de 50 servicios gratuitos, una vez acabado este crédito inicial es posible pasar a un plan de pago por uso, conservando los servicios gratuitos y disponiendo de una cantidad de recursos gratuitos al mes.

Por otro lado, se encuentra la plataforma perteneciente a los creadores de Kubernetes, Google Cloud. Al tratarse de los creadores de Kubernetes, es lógico pensar que lleva más tiempo que el resto de los proveedores mejorando sus servicios, lo cual no deja de ser cierto, pues su lanzamiento es un año posterior al lanzamiento de Kubernetes.

Google Cloud (Google GKE, 2023) ofrece dos posibilidades a la hora de implementar un clúster basado en Kubernetes: modo estándar y modo Autopilot. Ambas opciones permiten la implantación de un clúster administrador por Google, sin embargo, la diferencia viene dada en cuan gestionado está por Google. Por un lado contamos con el modo estándar, en el cual el proveedor se encarga de gestionar el plano de control y los nodos master (al igual que en otras plataformas no tenemos acceso a los mismos para poder gestionarlos), permitiendo al usuario gestionar el resto de configuraciones del clúster, como el tipo de instancias que van a ejecutarse, sus características el auto escalado o la creación y eliminación de grupos de nodos, siendo facturado por las características que el usuario dote al clúster.

La segunda opción que ofrece el proveedor es una de las más interesantes, puesto que es el propio proveedor el que se encarga de gestionar el clúster al completo. Esto es, desde las instancias que corren los nodos *master*, los nodos *worker* y las características de ambos, el auto escalado de los nodos, etc. Por lo que el usuario únicamente se encarga de administrar las aplicaciones desplegadas en el clúster. Así pues, la facturación de este tipo de despliegue viene dada por la cantidad de *pods* desplegados y en ejecución, ya que el resto de infraestructura está gestionada por el proveedor.

	Microsoft Azure	**Google Cloud**
Áreas geográficas	Johannesburgo, Doha, Dubái, Frankfurt, Irlanda, Países Bajos, París, Oslo, Varsovia, Cardiff, Londres, Gävle, Zúrich, Hong Kong, Singapur, Canberra, Nueva Gales del Sur, Victoria, Shanghái, Beijing, Seúl, Pune, Chennai, Osaka, Tokio, São Paulo, Toronto, Quebec, Iowa, Illinois, Wyoming, Texas, Virginia, California, Arizona, Washington	Oregón, Los Ángeles, Salt Lake, Las Vegas, Iowa, Carolina del Sur, Norte de Virginia, Columbus, Dallas, Montreal, Toronto, Santiago de Chile, São Paulo, Londres, Bélgica, Países Bajos, Zúrich, Frankfurt, Varsovia, Milán, Madrid, París, Turín, Bombay, Deli, Singapur, Yakarta, Hong Kong, Taiwán, Osaka, Sídney, Melbourne, Seúl, Tel Aviv, Doha.
Zonas de disponibilidad	87 entre todas las regiones	112 entre todas las regiones
Servicios Kubernetes ofrecidos	• Microsoft Azure Kubernetes Service (Microsoft AKS)	• Google Kubernetes Engine con Autopilot • Google Kubernetes Engine sin Autopilot
Tipos de gestión	• Gestión, únicamente, de nodos *worker*	• Gestión no disponible (ni *master* ni *worker*) • Gestión, únicamente, de nodos *worker*
Balanceadores de carga	• Gateway • Clásico	• De aplicación • De red • Clásico
Red virtual privada (VPC)	• Elastic IP • Azure VPC ○ Subred ○ Switch ○ Router ○ Enrutamiento	• Elastic IP • Alibaba VPC ○ Subred ○ Switch ○ Router ○ Enrutamiento
Almacenamiento	• Azure Disk Storage • Azure Archive Storage • Azure Elastic SAN • Almacenamiento de contenedor de Azure	• Google Cloud Storage • Google Persistent Disk • Google Filestore

Tabla 3. Microsof Azure VS Google Cloud

5. Despliegue de un clúster Kubernetes en Alibaba Cloud

En este apartado se exponen los distintos tipos de despliegue de un clúster altamente disponible que tenemos disponibles en la plataforma de Alibaba. Además, se realiza una descripción de las distintas herramientas de gestión del clúster ofrecidas por Alibaba.

Para el despliegue de cualquier tipo de servicio que ofrece Alibaba, es necesario en primer lugar iniciar sesión en la consola de administración. Una vez iniciada la sesión, se puede acceder a los distintos tipos de servicios que ofrece Alibaba, En el caso de querer desplegar el clúster de Servicio de Contenedores para Kubernetes, solo hay que seleccionar la opción "*Container Service for Kubernetes*".

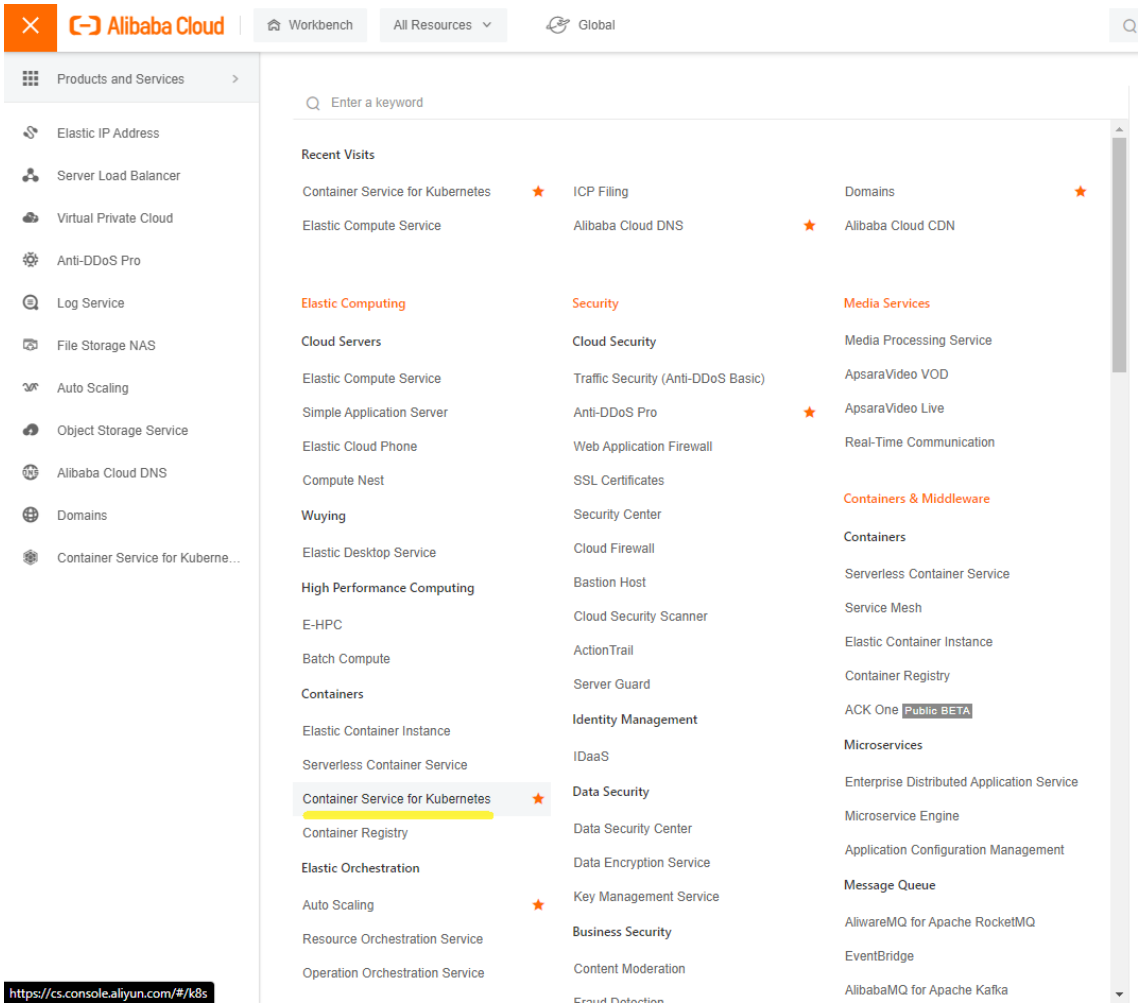


Ilustración 26. Consola de administración de Alibaba

Para continuar con cualquier despliegue, una vez se accede al *Container Service for Kuberntes*, se encuentra la página que permite administrar todos los clústeres que se tengan en activo.

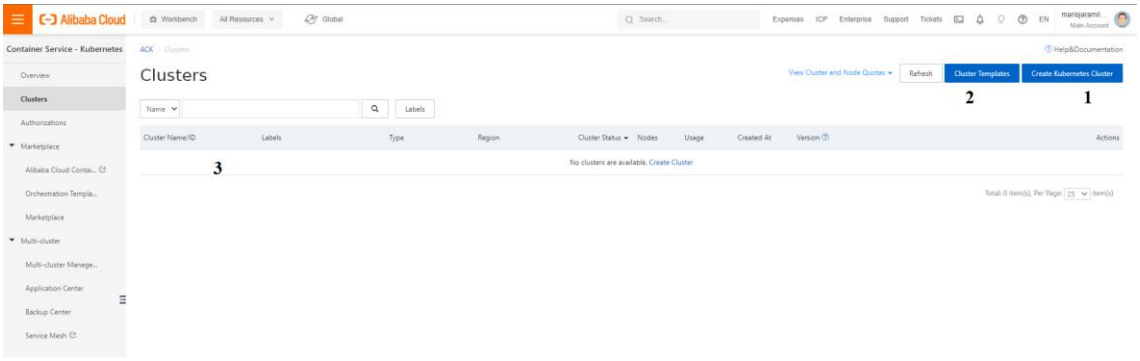


Ilustración 27. Consola de administración de clústeres

Entre las posibilidades ofrecidas, hay que destacar:

1. Crear un clúster Kubernetes.
2. Plantillas de clústeres.
3. Espacio en el que se muestran todos los clústeres creados.

Una vez se accede a la opción *Create Kubernetes Clúster*, se puede observar que Alibaba ofrece hasta cinco probabilidades de despliegue: *Managed Kubernetes*, *Dedicated Kubernetes*, *Serverless Kubernetes*, *Managed Edge Kubernetes* y *Register Clúster*. Estas opciones van a ser comentadas a continuación.

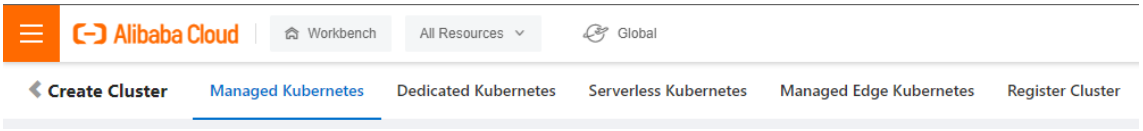


Ilustración 28. Tipos de despliegue de clústeres en Alibaba Cloud

5.1 Managed Kubernetes

Este tipo de despliegue que ofrece Alibaba permite la creación de un clúster Kubernetes de forma que, en este caso, solo se crean y administran los nodos *worker*, de los nodos *master* se hace cargo el Servicio de Contenedores de Alibaba. Para llevar a cabo este tipo de despliegue, una vez se accede a la página de creación de clústeres, hay que seleccionar la opción "*Managed Kubernetes"* y, a continuación, empezar a detallar las características del clúster a crear a través de las distintas opciones que se van ofreciendo (Alibaba ACK Managed, 2023).

Clúster Configurations. Se eligen las opciones básicas respecto al clúster, como son: nombre del clúster, región en la que se encuentra, versión de Kubernetes, red y seguridad.

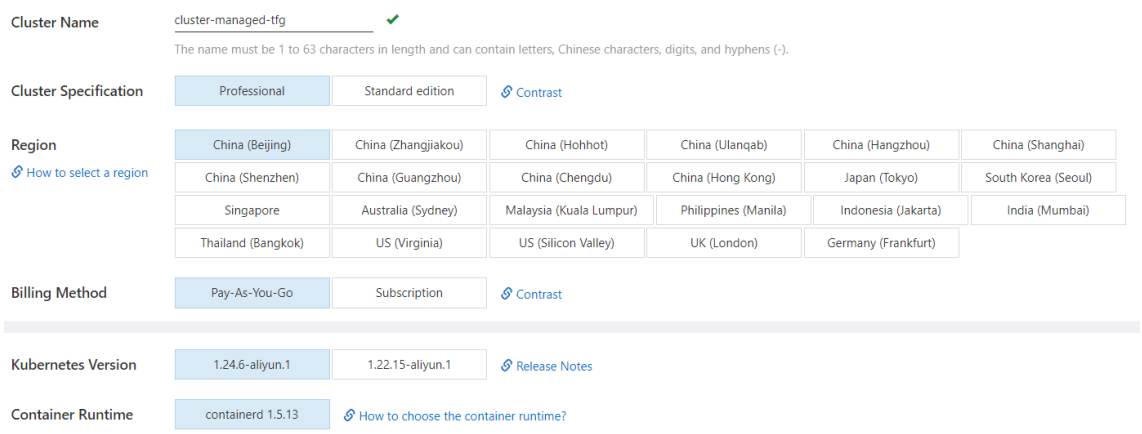


Ilustración 29. Configuraciones del clúster tipo Managed Kubernetes 1

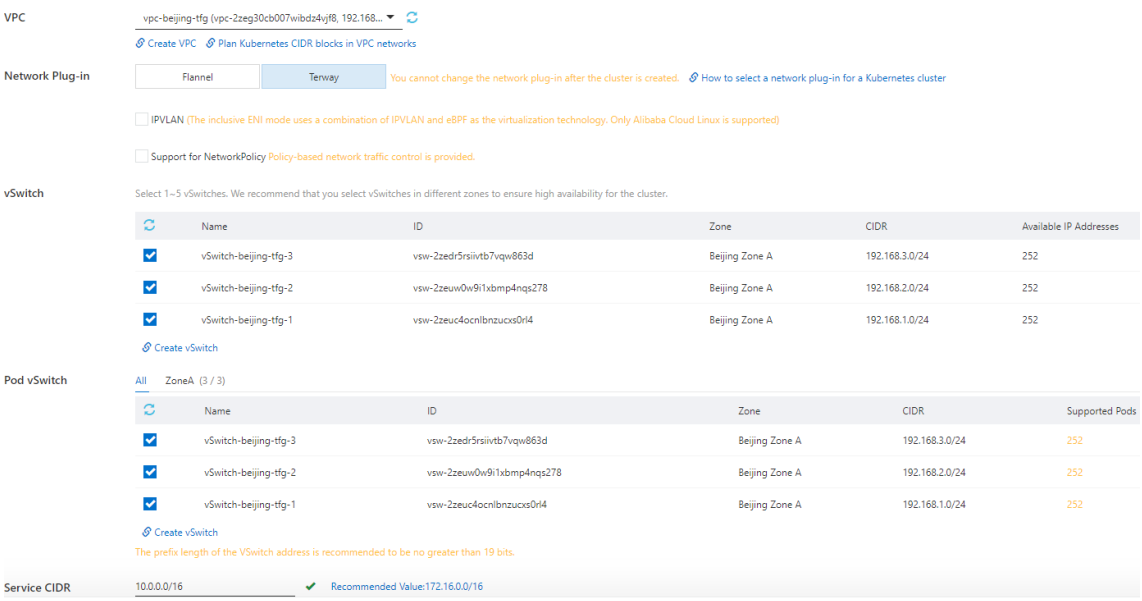


Ilustración 30. Configuraciones del clúster tipo Managed Kubernetes 2

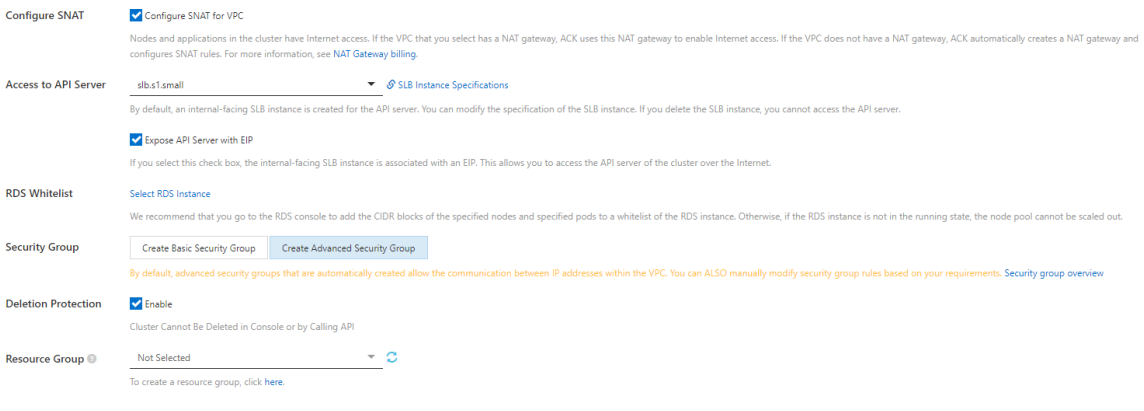


Ilustración 31. Configuraciones del clúster tipo Managed Kubernetes 3

Ilustración 32. Configuraciones avanzadas del clúster tipo Managed Kubernetes

Node Pool Configurations. Durante esta fase se van a elegir las características correspondientes a los nodos *worker* del clúster que estamos creando. Algunas de estas características son: nombre del *pool* de nodos, tipo de instancias que componen el *pool* (familia, vCPU, Memoria principal, zona, *pods* soportados de forma simultánea, ancho de banda), cantidad de instancias a carear, disco duro que van a albergar y su capacidad, sistema operativo y forma de acceso (par de claves, contraseña o decidir luego).

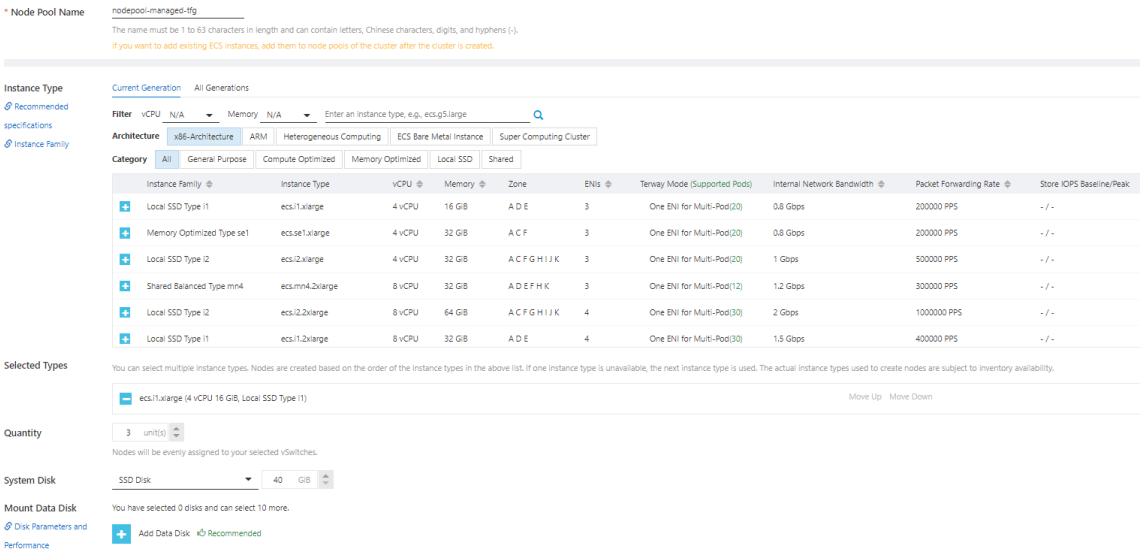


Ilustración 33. Configuración del pool de nodos worker 1

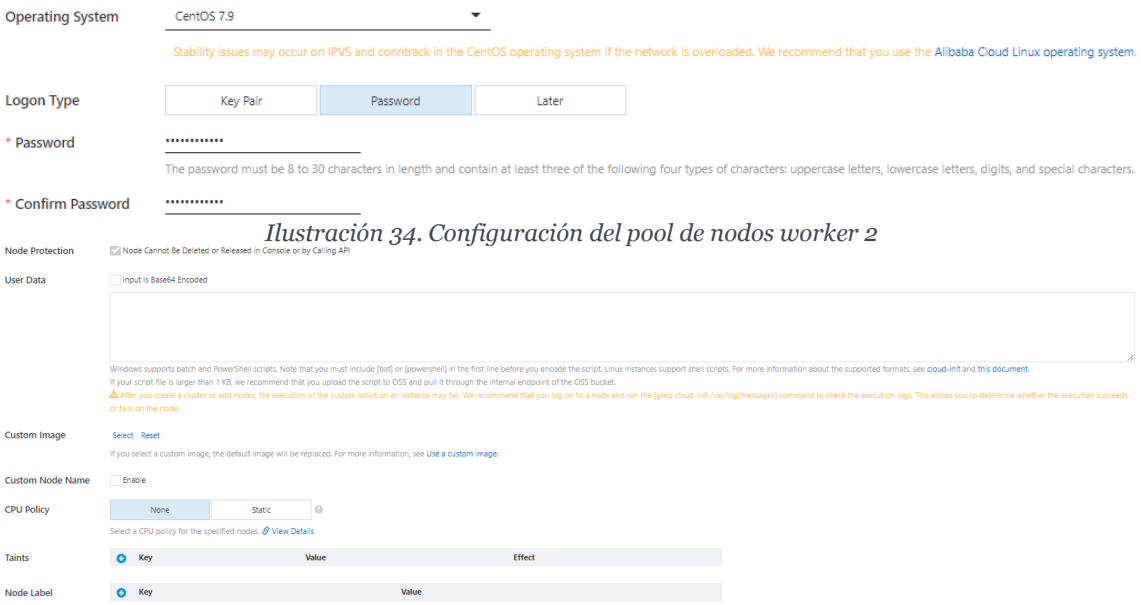


Ilustración 34. Configuración del pool de nodos worker 2

Ilustración 35. Configuración avanzada del pool de nodos worker

Component Configurations. A lo largo de esta fase se encuentran más opciones que se pueden añadir al clúster, aunque en este caso no se trata de características técnicas, sino de características de gestión en caso de que deseemos contar tras crear el clúster. Estas características son: sistema Ingress, sistema de monitorización, alertas y gestión de logs.

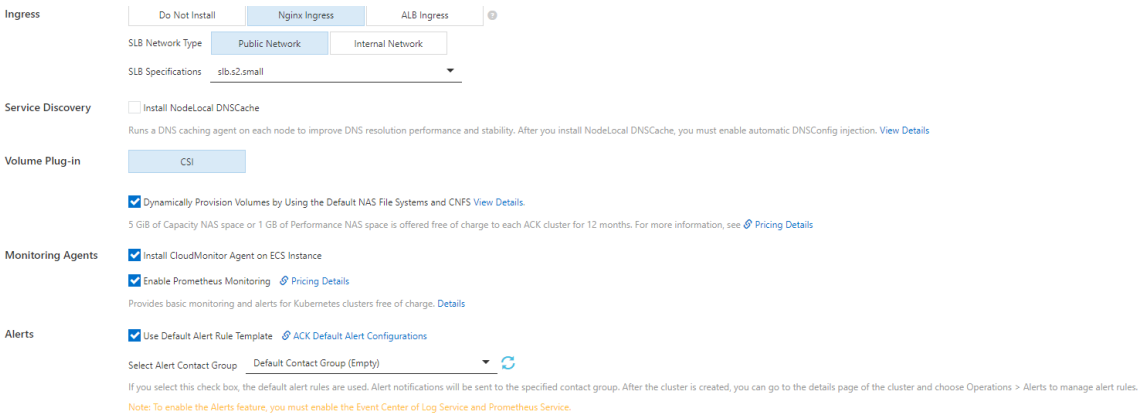


Ilustración 36. Configuración adicional del clúster 1

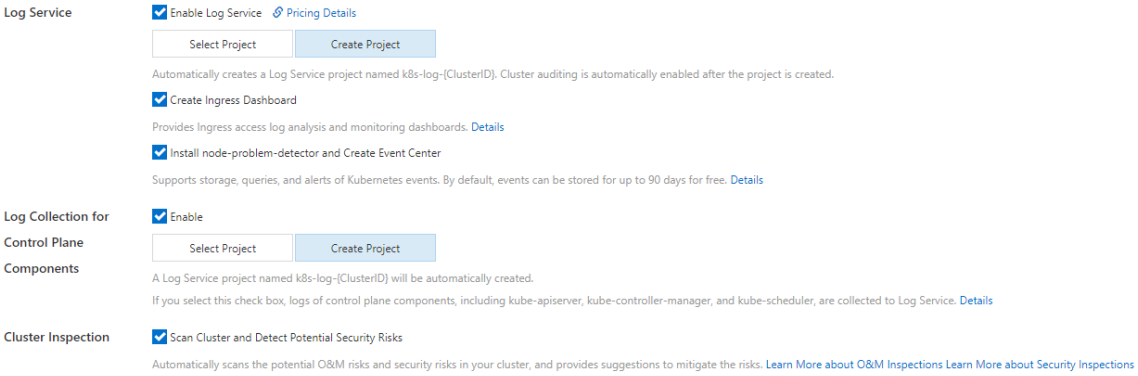


Ilustración 37. Configuración adicional del clúster 2

Confirm Order. Finalmente muestra un resumen de aquello que hemos seleccionado para confirmar la orden de creación del clúster, confirmando el precio de dicho clúster.

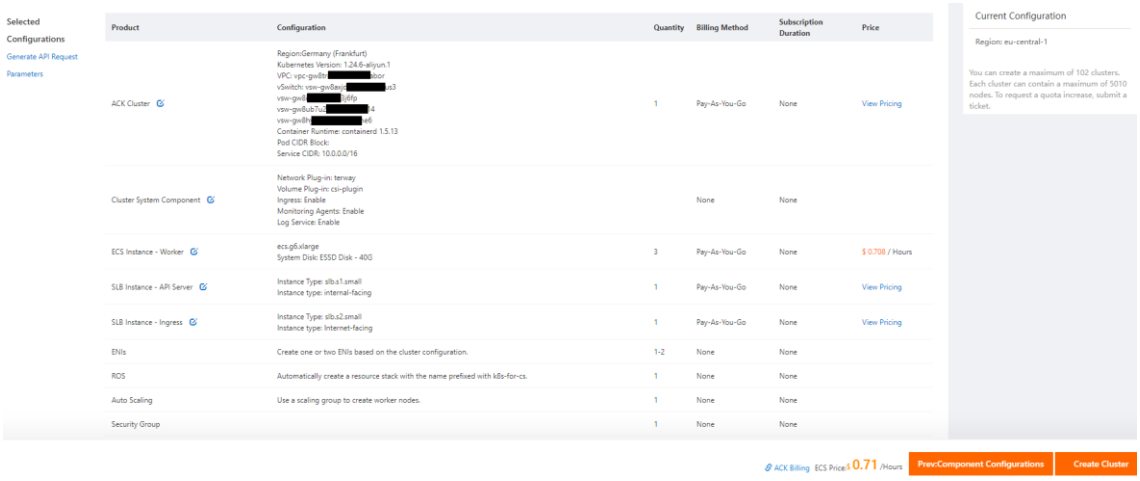


Ilustración 38. Confirmar creación clúster Managed Kubernetes

Una vez se ha confirmado la creación del clúster, este empezará a desplegarse. Al final del proceso, aparecerá en la tabla de clústeres activos para proceder a la gestión de este.

Ilustración 39. Managed Kubernetes desplegado

5.2 Dedicated Kubernetes

A diferencia del anterior tipo de despliegue, en el caso de un despliegue del tipo *Dedicated Kubernetes* en Alibaba permite crear y administrar, no solo los nodos *worker*, sino también los nodos *master* que conformarán el clúster (Alibaba ACK Dedicated, 2023).

El proceso de despliegue de este tipo de clúster es muy similar al despliegue del clúster tipo *Managed Kubernetes*. De hecho, los pasos a seguir durante las fases de *Clúster Configurations*, *Node Pool Configurations*, *Component Configurations* y *Confirm Order*, son los mismos que en el caso anterior.

Dado que la diferencia entre el despliegue anterior y este radica en que en este despliegue se crean los nodos *master*, es de esperar que haya un paso intermedio (entre *Clúster Configurations* y *Node Pool Configurations*) que permite especificar los detalles de los nodos *worker*, esta es la fase *Master Configurations.*

Durante esta fase de especificación de las características de los nodos *master* cabe destacar que:

- Hay que elegir entre disponer de 3 o 5 nodos *master*.
- Cada nodo *master* estará en una zona distinta de la región escogida.
- Cada nodo *master* puede tener características distintas de vCPU, memoria principal, pods soportados, tipo de instancia, etc.

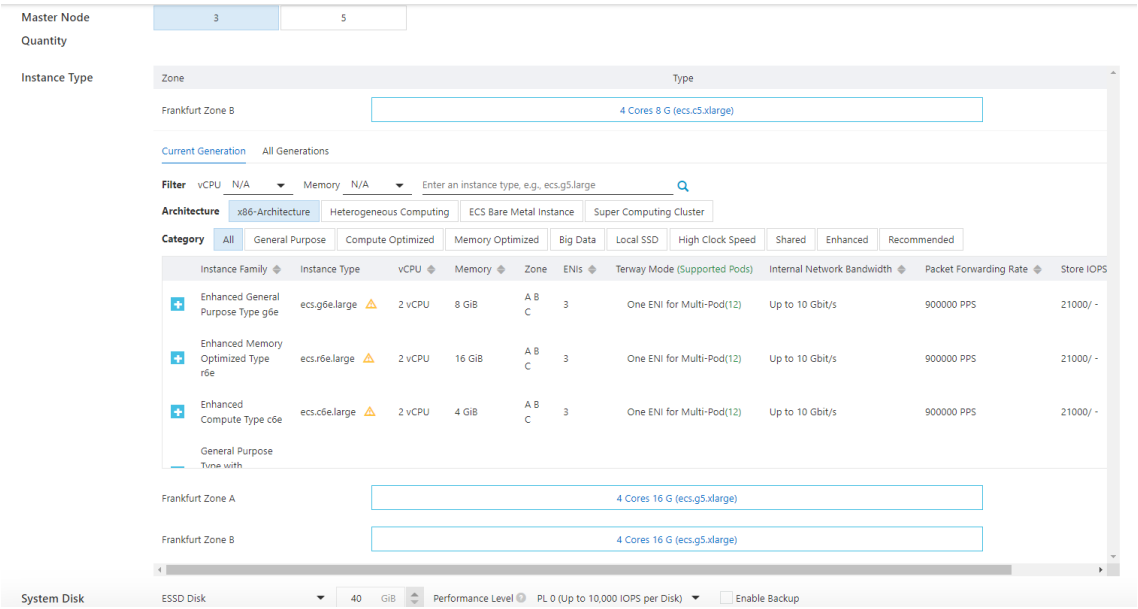


Ilustración 40. Master Configurations en Dedicated Kubernetes

5.3 Serverless Kubernetes

El tipo de despliegue *Serverless Kubernetes* se caracteriza por ser un despliegue enfocado en el simple hecho de desplegar aplicaciones, sin necesidad de tener que realizar ninguna configuración técnica de un clúster, pues el propio Servicio de Contenedores se encarga de realizar las tareas necesarias para desplegar y mantener tanto los nodos *master* como los nodos *worker*, por lo que estos no son accesibles. El servicio Kubernetes únicamente estará activo durante la ejecución de las aplicaciones desplegadas, por lo que su facturación depende también del tiempo de ejecución de la aplicación desplegada (Alibaba ACK Serverless, 2023).

Los *pods* desplegados son creados en nodos virtuales, por lo que se asegura una alta escalabilidad y aislamiento de las aplicaciones que se desplieguen. También permiten un mayor ahorro de costes, pues la facturación del servicio viene dada por el uso de las aplicaciones desplegadas, no por los recursos que se puedan contratar.

Para realizar este tipo de despliegue se tienen que seguir unos pasos similares a los tipos de despliegue vistos anteriormente, pero de forma más simple. Las fases del despliegue son dos: *Clúster Configurations*, *Component Configurations* y *Confirm Order*.

Durante la fase de *Clúster Configurations* se puede elegir el nombre del clúster y su región (y zona) de despliegue, la versión de Kubernetes, la subred del servicio, el tipo de balanceador de carga y el dominio entre otras.

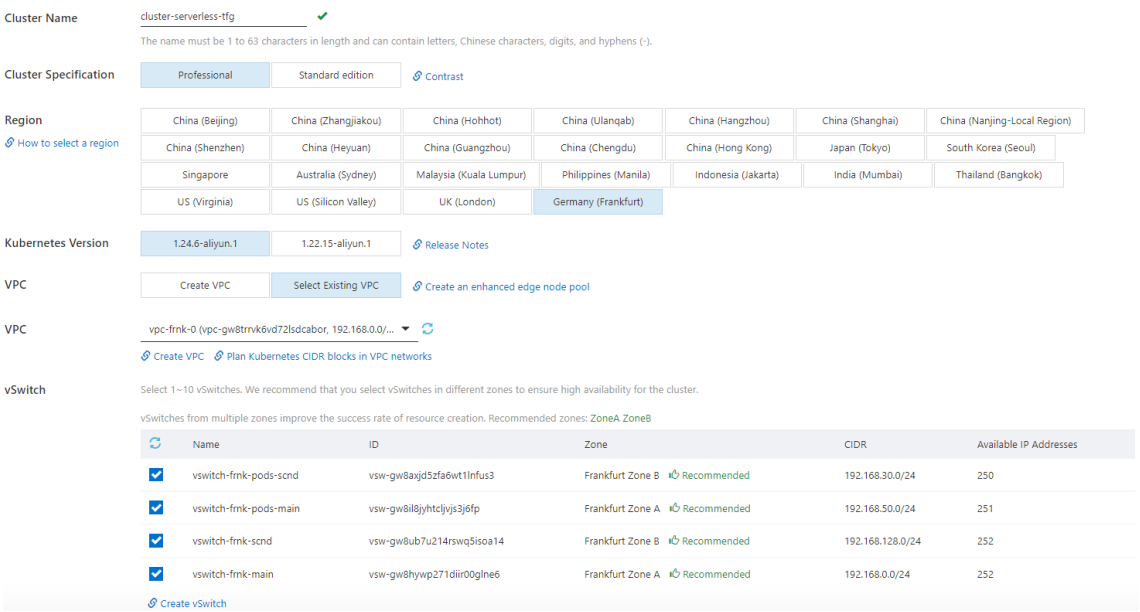


Ilustración 41. Clúster Configurations en Serverless Kubernetes 1

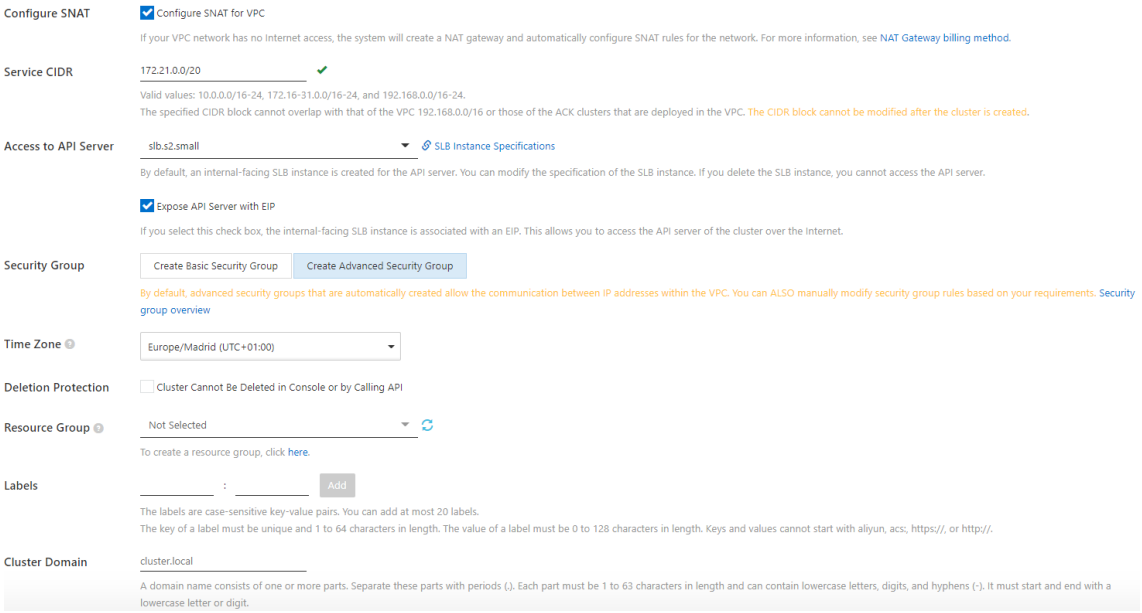


Ilustración 42. Clúster Configurations en Serverless Kubernetes 2

Para la fase de *Component Configurations* se observa que las configuraciones que se permiten hacen referencia a la creación de servicios DNS, servicio *Ingress*, métricas y logs, y la implementación de un *framework* para aplicaciones basadas en clústeres Kubernetes *serverless* denominado "Knative".

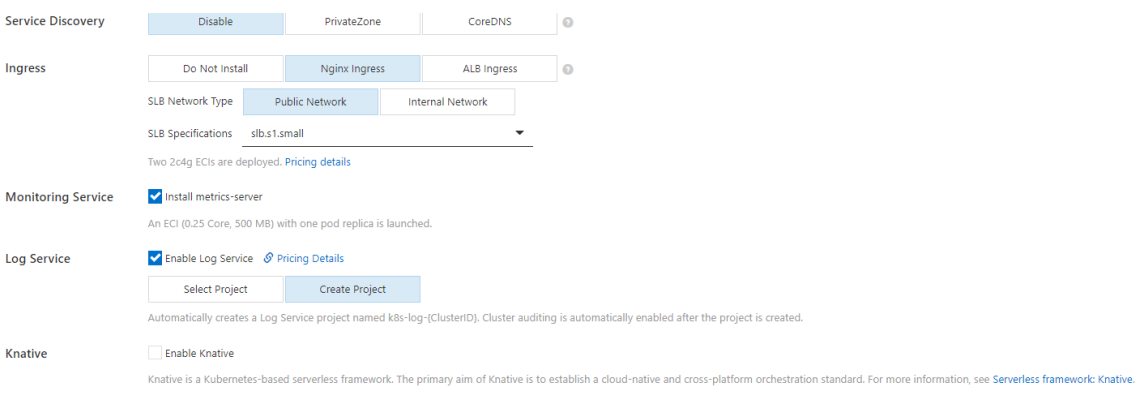


Ilustración 43. Component Configurations en Serverless Configurations

Por último, se pasa a la fase *Confirm Order*, en la que, de nuevo, se obtiene un resumen de la configuración seleccionada para proceder a su creación.

5.4 Managed Edge Kubernetes

Cada vez es mayor el número de dispositivos IoT que se despliegan en todo el mundo, estos dispositivos generan una gran cantidad de datos que es necesario procesar, sin embargo, realizar la transferencia de todos estos datos a un único centro de datos, teniendo en cuenta que los dispositivos IoT pueden estar muy descentralizados, significaría tener que realizar una gran inversión en transferencia de datos, necesitarían una gran cantidad de ancho de banda (Alibaba ACK Edge, 2023).

Una de las soluciones a este problema que se han propuesto es el uso de Kubernetes. Dado que este tipo de clústeres pueden ser desplegados en distintas zonas geográficas, es más fácil y eficiente realizar estos despliegues en zonas más cercanas a dónde se encuentren ubicados estos dispositivos, reduciendo el ancho de banda necesario para transferir todos estos datos.

No es necesario realizar el despliegue de un clúster en cada zona cercana al *Edge* de los dispositivos, otra solución disponible es desplegar una parte del clúster en una determinada zona, y para acercarse a las distintas zonas en las que se encuentran los dispositivos, desplegar un nodo *worker*, por ejemplo, en cada zona deseada.

En Alibaba se ofrece la posibilidad de realizar un despliegue del primer tipo, un clúster entero en una determinada zona, pero adecuando sus características a este tipo de conexiones. Para ello se ofrece la posibilidad de utilizar hasta 256 nodos en el clúster creado y 256 *pods* en cada nodo.

Para este tipo de despliegue se encuentran cuatro fases: *Clúster Configurations*, *Worker Configurations*, *Component Configurations* y *Confirm Order*. Las fases *Worker Configurations* y *Confirm Order* son iguales a las ya mencionadas en el despliegue del *Managed Kubernetes*, por lo que se detallan las fases restantes.

La fase *Clúster Configurations* tiene una estructura similar al *Managed Kubernetes*, la mayor diferencia radica en la configuración de la red, que en este caso permite elegir, aparte de la VPC y el vSwitch a usar, la cantidad de direcciones IP por nodo y las subredes que serán utilizadas por los *pods* y por el servicio.

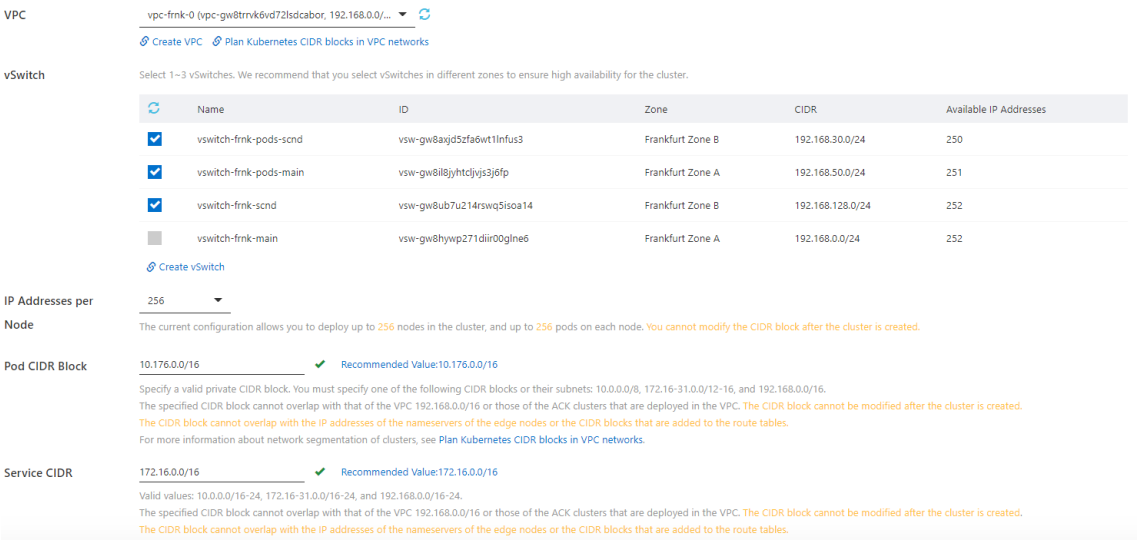


Ilustración 44. Configuración de red en Managed Edge Kubernetes

5.5 Register Clúster

Finalmente, el servicio Alibaba ACK ofrece la posibilidad de registrar y administrar otros clústeres Kubernetes desplegados en distintas regiones, en proveedores distintos a Alibaba Cloud o incluso clústeres locales desplegados *on premise*.

Para configurar este tipo de despliegue, únicamente se solicita el nombre del clúster con el que se conocerá el despliegue, la región donde se despliega la red que la compone y el balanceador de carga que se usará para gestionar el despliegue.

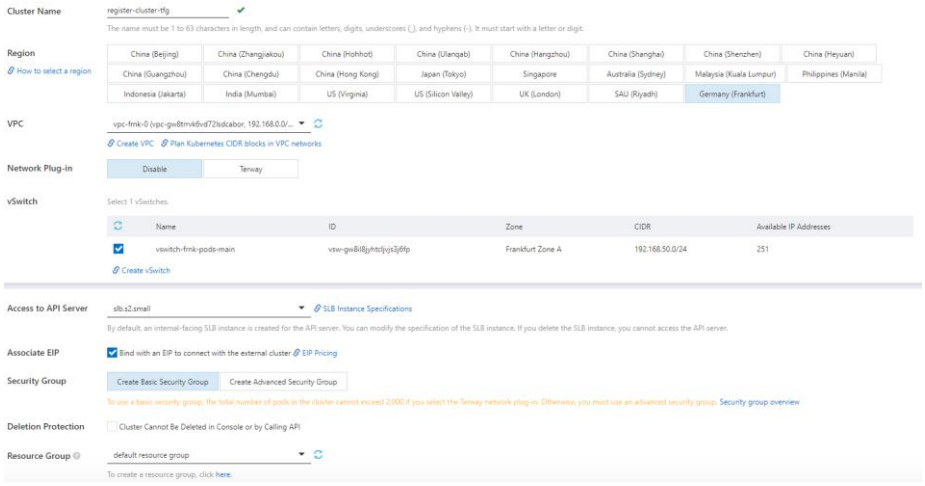


Ilustración 45. Register Cluster Alibaba Cloud

5.6 Comparación de despliegues

A continuación, se muestra una comparación de los distintos despliegues ofrecidos por Alibaba ACK para ofrecer una comparativa de los mismos de forma más resumida.

	Managed Kubernetes	**Dedicated Kubernetes**	**ACK Serverless**
Descripción	Clúster parcialmente administrador por Alibaba	Clúster NO administrado por Alibaba	Clúster totalmente administrado por Alibaba sin nodos desplegados
Administrar master	NO	SI	NO
Administrar worker	SI	SI	NO
Método de facturación	Suscripción Pago por uso	Suscripción Pago por uso	Pago por uso
Necesidad VPC	SI	SI	SI
Necesidad de balanceador de carga	SI	SI	NO
Versión de Kubernetes ofrecida	1.22.15-aliyun.1 1.24.6-aliyun.1 1.26.3-aaliyun.1	1.22.15-aliyun.1 1.24.6-aliyun.1 1.26.3-aaliyun.1	1.22.15-aliyun.1 1.24.6-aliyun.1 1.26.3-aaliyun.1
SO ofrecido	Alibaba Cloud Linux CentOS 7.9	N/A	N/A
Controlador Ingress ofrecido	Nginx ALB (Application Load Balancer) MSE (Microservices Engine)	Nginx ALB (Application Load Balancer) MSE (Microservices Engine)	Nginx ALB (Application Load Balancer) MSE (Microservices Engine)
Monitorización	SI	SI	SI
Servicio Log	SI	SI	SI

Tabla 4. Comparación de despliegues Alibaba ACK I

	ACK Edge	**Register Kubernetes**
Descripción	Clúster especializado en redes perimetrales y gestión de IoT	Registrar y administrar clústeres externos a Alibaba
Administrar master	NO	N/A
Administrar worker	SI	N/A
Método de facturación	Pago por uso	Pago por uso
Necesidad VPC	SI	SI
Necesidad de balanceador de carga	NO	NO
Versión de Kubernetes ofrecida	1.22.15-aliyun.1 1.24.6-aliyun.1 1.26.3-aaliyun.1	N/A
SO ofrecido	N/A	N/A
Controlador Ingress ofrecido	N/A	N/A
Monitorización	NO	NO
Servicio Log	SI	NO

Tabla 5. Comparación de despliegues Alibaba ACK II

5.7 Almacenamiento persistente

Como se ha mencionado anteriormente, para que la información que reside en el clúster no se pierda con la destrucción de los *pods*, y asegurar su alta disponibilidad, es necesario que sea almacenada en un espacio persistente y externo al propio clúster. Alibaba ofrece principalmente dos posibilidades para esta solución: Object Storage Services y Apsara File Storage NAS (Alibaba ACK Storage, 2023).

	Apsara NAS	**Object Storage Service**
Descripción	Servicio de almacenamiento distribuido que ofrece acceso compartido, escalable, altamente persistente	Almacenamiento distribuido altamente persistente y fiable para el almacenamiento de objetos y su acceso a través de Internet. Recomendado para datos que no es necesario modificar frecuentemente
Escenario	Persistencia de datos, aplicaciones científicas, aplicaciones empresariales en producción	Aplicaciones basadas en APIs de objetos con acceso a través de Internet
Modo de acceso	Miles o decenas de miles de instancias con acceso aleatorio concurrente mediante POSIX	Millones de instancias con acceso aleatorio concurrente mediante SDKs o APIS RESTful
Capacidad	Desde Gibibytes a Pebibytes	Ilimitado
Modo de almacenamiento	Los archivos se organizan en directorios	Los archivos se organizan en el mismo directorio
Protocolo usado	NFS SMB	HTTP HTTPS
Latencia	Milisegundos (mediante NFS o SMB) Microsegundos (mediante clúster ACK)	Milisegundos

Tabla 6. Almacenamiento persistente en Alibaba Cloud

5.8 Auto escalado

A la hora de configurar un clúster, se definen una serie de características con las que cuentan los nodos que prestan el servicio, tales como su vCPU, memoria principal, almacenamiento, etc. Una vez desplegado un servicio, los *pods* en los que se despliega también cuentan con una serie de características que se definen en el momento del despliegue. Según la demanda del servicio, es posible que las características con las que se cuentan no sean suficientes para abastecer dicha demanda, lo cual deriva en una disminución de la calidad del servicio o incluso la pérdida del propio servicio (Alibaba ACK Autoscaling, 2023).

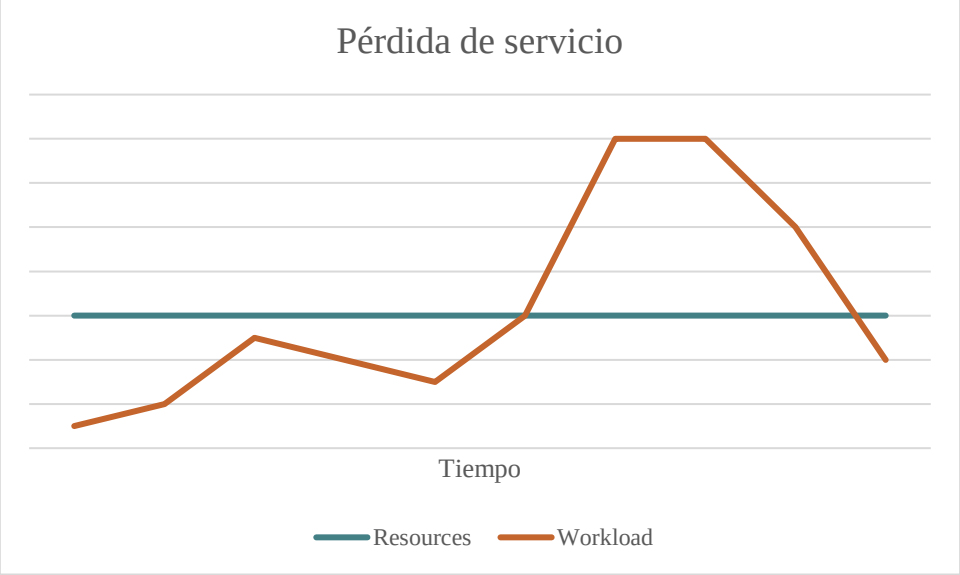


Ilustración 46. Auto escalado no activo

Para afrontar este problema, desde Alibaba Cloud se ofrece una solución de auto escalado integrado en Alibaba ACK que permite ajustar las características del clúster o incluso de los *pods* que prestan el servicio, de tal manera que se puedan ajustar a las necesidades del servicio según la demanda en cada momento. Para entender el servicio de auto escalado ofrecido por Alibaba, en primer lugar, hay que entender los distintos tipos de auto escalado existentes en Kubernetes.

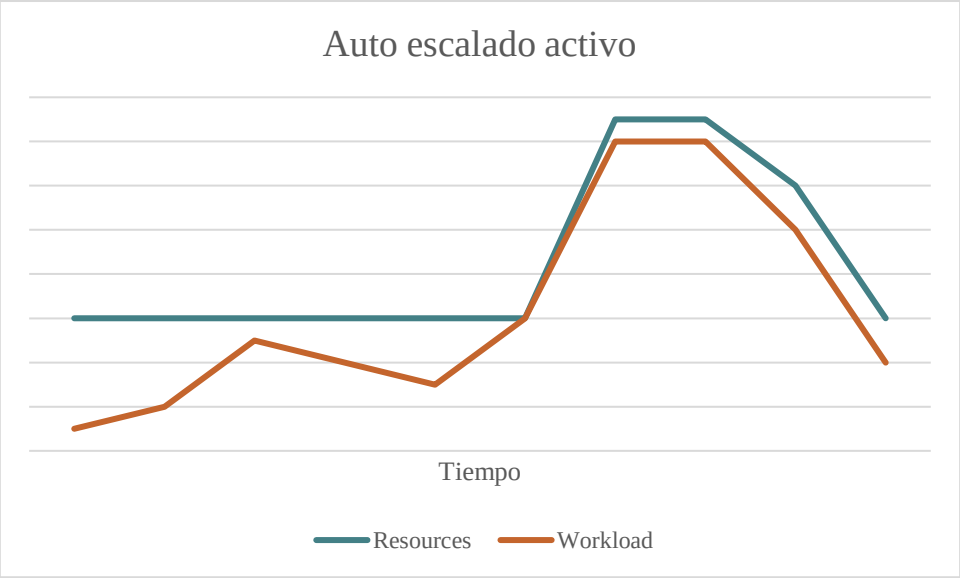


Ilustración 47. Auto escalado activo

Uno de los tipos de auto escalado más conocidos es el llamado *Horizontal Pod Autoscaler (HPA)*, el cual consiste en la replicación de los *pods* que prestan el servicio para dar abasto a la carga de trabajo a la que están sometidos en cada momento, de modo que entre mayor sea la carga de trabajo, más *pods* se crean, por el contrario, si la carga de trabajo desciende, el número de *pods* se decrementa para no malgastar recursos.

Otro de los tipos de auto escalado es el *Vertical Pod Autoscaler (VPA)*, el cual se diferencia de HPA en el hecho de no crear nuevas réplicas de los *pods* sino en aumentar sus especificaciones para poder hacer frente a una mayor carga de trabajo, permitiendo que, al igual que HPA, las especificaciones de los *pods* vuelvan a su estado inicial una vez la carga de trabajo va disminuyendo.

De forma oficial por parte de Kubernetes, también existe el denominado *Cluster Autoscaler*, siendo este el que más impacto puede llegar a tener en el despliegue de un clúster Kubernetes. En este caso, al aumentar la carga de trabajo, los componentes que se ven afectados son los propios nodos del clúster, ya que, a una mayor carga de trabajo, se aumenta el número de nodos desplegados en el clúster, para posteriormente, disminuir en función de la carga de trabajo.

Por parte de Alibaba Cloud, se ofrecen otra serie de opciones de auto escalado de forma complementaria a las anteriores, de esta forma, se obtiene una mayor adaptabilidad a las necesidades del negocio.

Entre las opciones ofrecidas por Alibaba Cloud, se encuentra *CronHPA*. Esta opción permite hacer uso de un auto escalado de tipo HPA de forma periódica y programada, de forma que, en caso de tener conocimiento que existen ciertos intervalos de tiempo o que la carga de trabajo a la que se tiene que enfrentar el servicio aumenta o disminuye de forma periódica, se pueda programar el escalado HPA para poder hacer frente de forma más precisa

Otro de los tipos de escalado ofrecidos es *Elastic-Workload*. Este escalado se encuentra enfocado a un escenario en el que se debe realizar un escalado más preciso. Permite que, para una misma carga de trabajo de un servicio en concreto, se puedan desplegar *pods* en un cluster ACK mientras que otros *pods* son desplegados en instancias ECS de forma separada, para poder hacer frente a la carga de trabajo de forma más efectiva y precisa.

Por último, también se ofrece el servicio *Virtual-Node*, este escalado está enfocado a aplicaciones *serverless* y a hacer frente a una subida de la carga de trabajo de forma que no se crean más instancias para nuevos nodos en el clúster, ya que los nuevos *pods* generados, son almacenados en un nodo virtual, de forma que únicamente se factura por el tiempo de ejecución de estos. El tiempo de creación de los *pods* en estos nodos virtuales es muy reducido, ya que, por ejemplo, para la creación de 1000 *pods* se tarda entre 15 a 30 segundos.

En lo respecta al ámbito económico, la facturación se ve afectada dependiendo del tipo de escalado que se aplique en cada momento, pues no se factura de igual forma un escalado HPA, VPA o el escalado del clúster o la utilización de un nodo virtual. En el caso de los escalados que afectan únicamente a los *pods* (sin hacer uso de un nodo virtual) la facturación no se ve afectada, pues los recursos contratados son los iniciales con los que se contrata el clúster. En el caso de un escalado del clúster, la facturación aumenta según los nuevos nodos creados, al igual que en el caso de *Elasstic-Workload*, pues se crean nuevas instancias ECS para asumir la nueva carga de trabajo. En el caso de la creación de un nodo virtual, la facturación aumenta según el tiempo de uso que haga del nodo virtual. Cabe destacar que una vez se dejen de utilizar los nuevos recursos creados, la facturación vuelve a su estado inicial, pues los nuevos recursos creados son destruidos y se dejan de facturar.

5.9 Observabilidad

Una vez desplegado cualquier servicio en el clúster, es necesario comprobar el correcto funcionamiento de estos, para ello es necesaria una solución que permita observar cómo se comportan los servicios, qué parte de estos tienen un correcto funcionamiento o dónde se puede encontrar un posible problema o deficiencia que altere el correcto funcionamiento de este.

Para abordar esta problemática, desde Alibaba Cloud se ofrecen varias herramientas que se encargan de recolectar y presentar la información de todos los elementos que conforman el clúster, así como sus componentes, aplicaciones desplegadas, e incluso una herramienta para analizar el impacto de las aplicaciones en el público (Alibaba ACK Observability, 2023).

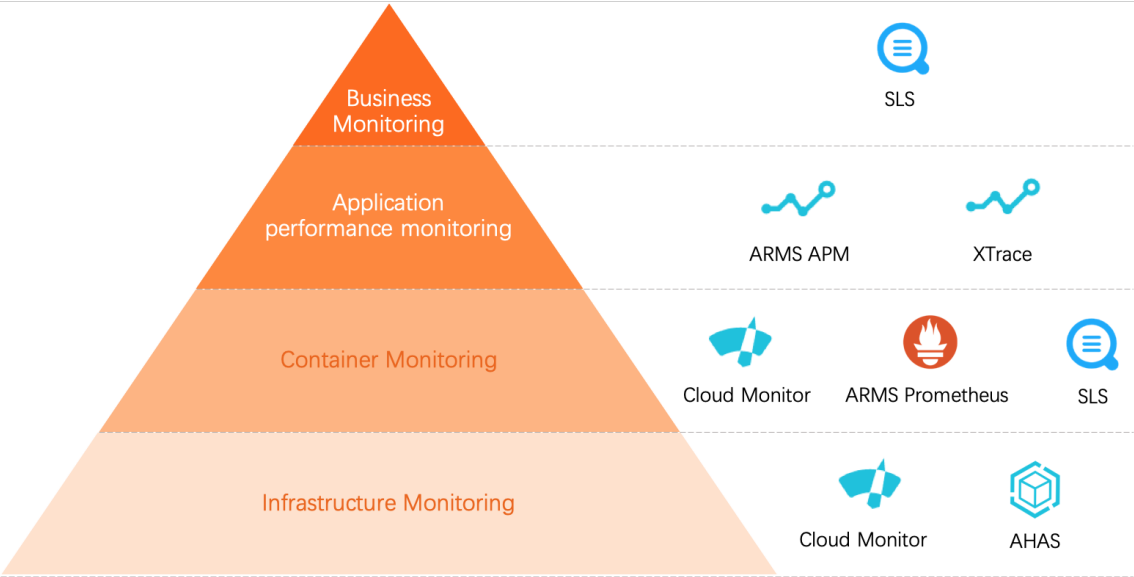


Ilustración 48. Observabilidad en Alibaba Cloud

El modelo de observabilidad de Alibaba Cloud está formado por cuatro capas, estructuradas según el nivel de profundidad del recurso que se analice, para cada capa existen diferentes herramientas de monitorización, como se ha indicado anteriormente. La capa del nivel más profundo es la de infraestructura, pasando por los contenedores Kubernetes, la aplicación desplegada y el negocio.

La capa más externa es la capa de infraestructura. En esta capa se pretende llevar a cabo una monitorización y ofrecer una visión de la arquitectura del clúster Kubernetes contratado con la plataforma. Para llevar a cabo estas acciones, Alibaba ofrece dos herramientas que se encuentran de forma nativa en el servicio *Alibaba Container Service for Kubernetes* (Alibaba ACK). En primer lugar, cuenta con la herramienta *Application High Availability Service* (AHAS) que permite hacer un descubrimiento de forma automática de la arquitectura del clúster, así como de un servicio de alerta según métricas, registro de logs del funcionamiento del propio clúster, enlace con aplicaciones de terceros (Jira, GitHub, GitLab, Trello, etc); por otro lado, ofrece una herramienta incluida en el despliegue del clúster que permite llevar a cabo una monitorización de los elementos que conforman el clúster, pero de forma más superficial, esta herramienta es conocida como CloudMonitor.

En la siguiente capa, el objetivo a monitorizar son los componentes que se ejecutan en el clúster, los contenedores. Para ello, eso posible hacer uso de la misma herramienta CloudMonitor mencionada para la capa anterior. En caso de necesitar un análisis en más profundidad del rendimiento de los contenedores, así como de su comportamiento o las métricas que puedan arrojar, es posible hacer uso de una de las herramientas más conocidas en el ámbito de Kubernetes, el conocido como servicio Prometheus. Siendo posible el uso de un servicio Protmetheus integrado en Alibaba Cloud, como conectar el clúster a un servicio propio de monitorización mediante Prometheus para tener un control total de la monitorización que se desee llevar a cabo. Al igual que en la capa anterior, también existe una herramienta nativa que permite notificar alertas sobre eventos de los contenedores mediante herramientas de terceros.

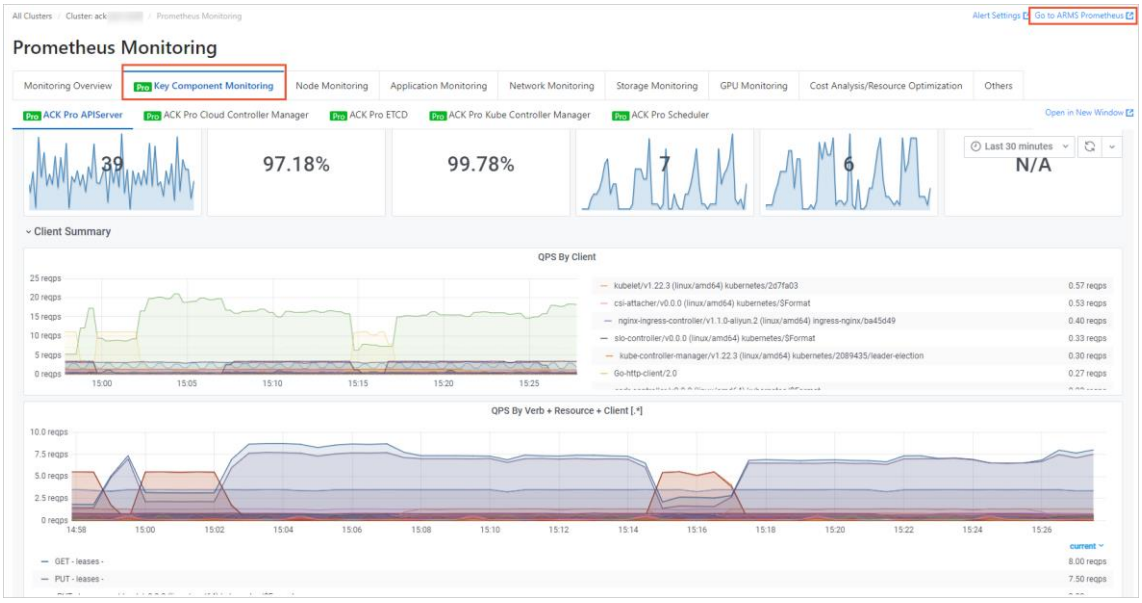


Ilustración 49. Servicio Prometheus

La siguiente capa es, una de las más interesantes para comprender el estado de las aplicaciones desplegadas, así como su comportamiento en tiempo real y análisis para mejoras de estas, siendo también una de las más novedosas. Para comprender la importancia de esta capa de observabilidad es necesario tener en cuenta el funcionamiento de una aplicación basada en microservicios.

Las aplicaciones basadas en microservicios disponen de varios componentes que se interconectan para hacer funcionar una aplicación más compleja, de forma que esta sea más fácil de mantener, actualizar e implementar. Este es un caso de uso muy común en la computación en la nube mediante el uso de Kubernetes, dado que cada componente, o microservicio, puede ser implementado en un contenedor diferente, de forma que una vez interconectados todos, puedan hacer funcionar de forma eficiente la aplicación, reduciendo los costes de implementación.

Este tipo de implementación tiene unas claras ventajas a la hora de implementar una aplicación, sin embargo, también conlleva una serie de riesgos para tener en cuenta, uno de estos riesgos es la posibilidad que el rendimiento de la aplicación no sea el deseado. Dada la naturaleza del uso de microservicios para la implementación de una aplicación, puede resultar bastante complicado saber cuál de estos microservicios es el que está produciendo esta pérdida de rendimiento, ya que, a más microservicios, más aumentan los posibles puntos de fallo. Para poder realizar un análisis de la arquitectura de una aplicación de este tipo, así como de un análisis de su funcionamiento, Alibaba Cloud pone a disposición de los desarrolladores una serie de herramientas que permiten realizar un análisis detallado de su funcionamiento, el estado en tiempo real, la arquitectura, etc. Esta serie de herramientas conforman un análisis de *tracing*.

En primer lugar, se dispone de una herramienta que permite realizar un autodescubrimiento de la arquitectura de la aplicación, permitiendo realizar un análisis topográfico de la arquitectura de dicha aplicación, definiendo cada microservicio que la conforma. De esta forma es posible tener una visión global de cada elemento que conforma la aplicación.

Ilustración 50. Descubrimiento topológico

Al igual que el resto de los servicios de observabilidad, también permite la recolección y presentación de logs de cada componente de la aplicación, así como de estadísticas de estos, diferenciando los resultados de cada componente, permitiendo averiguar si alguno de estos componentes no está desempeñando un correcto funcionamiento y de esta forma haciendo que la aplicación no se comporte de la forma en que se espera que deba funcionar.

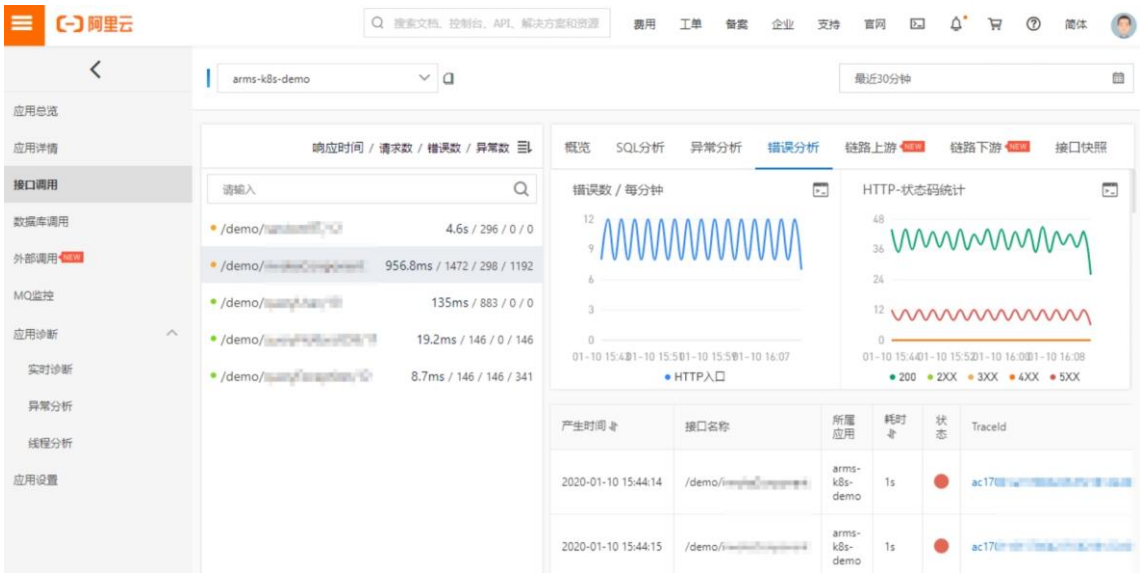


Ilustración 51. Datos por microservicio

Otra posibilidad que ofrece Alibaba Cloud, mediante el análisis de *tracing*, es una funcionalidad que permite simular una petición a la aplicación. Gracias a esta funcionalidad, se permite que esta petición pase por cada componente de la aplicación, permitiendo de esta forma recolectar estadísticas y datos sobre el funcionamiento de cada componente de forma individual, gracias a lo cual, es posible realizar los ajustes necesarios en el componente que no presente el funcionamiento adecuado.

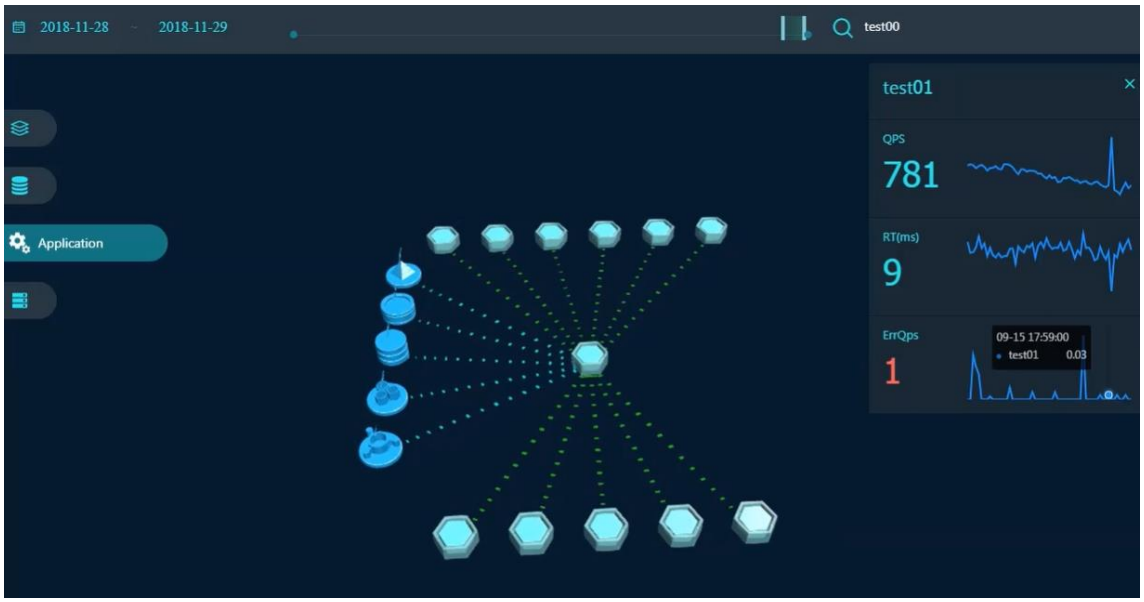


Ilustración 52. Visión por microservicio

La última capa de observabilidad es aquella dedicada a realizar un análisis de las estadísticas de negocio de las aplicaciones desplegadas. Para ello, Alibaba ofrece una herramienta integrada en Alibaba ACK que permite extraer datos de acceso a cada aplicación desplegada en el clúster, permitiendo tener una visión del uso que se le está dando a cada aplicación, tanto por visitas totales a la aplicación, como midiendo cuáles de estas son visitas por visitantes únicos. De esta forma, es posible llevar a cabo un análisis de los costes que genera cada aplicación desplegada, así como de su rendimiento e impacto en el mercado.

6. Pruebas de despliegue en Alibaba Cloud

Para realizar el despliegue en la plataforma Kubernetes se ha elegido hacer uso del servicio Alibaba ACK, concretamente mediante la opción Dedicated Kubernetes siguiendo los pasos mencionados en el punto 5.2. En este caso, el despliegue dispone del clúster está conformado con la siguiente configuración.

Nombre del clúster	cluster-tfg
Región	Alemania (Frankfurt)
Método de facturación	Pago por uso
Versión de Kubernetes	1.26.3-aliyun.1
CIDR	172.16.0.0/16
Grupo de seguridad	Básico
Dominio del clúster	k8s-majasi.cluster
Cantidad de nodos master	3
Tipo de instancia nodos master	ecs.g7a.large (2vCPU, memoria principal 8GiB, almacenamiento 50GiB)
Nombre del pool *de nodos*	k8s-nodepool
Servicio de contenedores	Containerd 1.6.20
Cantidad de nodos worker	2
Tipo de instancia nodos worker	ecs.g5.xlarge (4 vCPU, memoria principal 16GiB, almacenamiento 50GiB)
Sistema operativo	Alibaba Cloud Linux 3.2.104
Ingress	ALB (Gestionado)
Monitorización de contenedores	Prometheus
Servicio de Logs	Activado
Seguridad	Inspección de Clúster

Tabla 7. Configuración clúster en Alibaba Cloud

Una vez especificadas las características con las que va a contar el clúster que se desea crear, se muestra un precio aproximado, tanto en la opción de pago por uso, como en la opción de suscripción

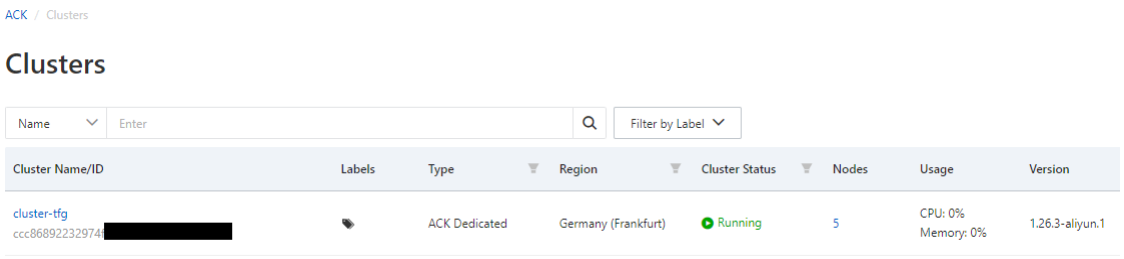


Ilustración 53. Clúster creado en Alibaba Cloud

Se procede a la creación del volumen persistente, en este caso, mediante interfaz gráfica de la propia consola de Alibaba Cloud-

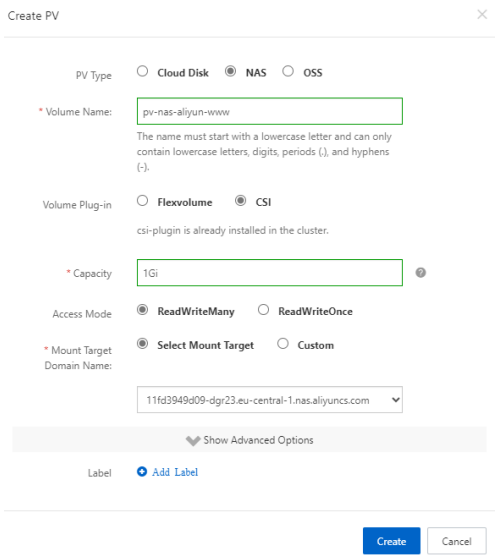


Ilustración 54. Creación del Volumen Persistente

A partir del volumen creado, se crea la petición del volumen persistente (*Persistent Volume Claim*).

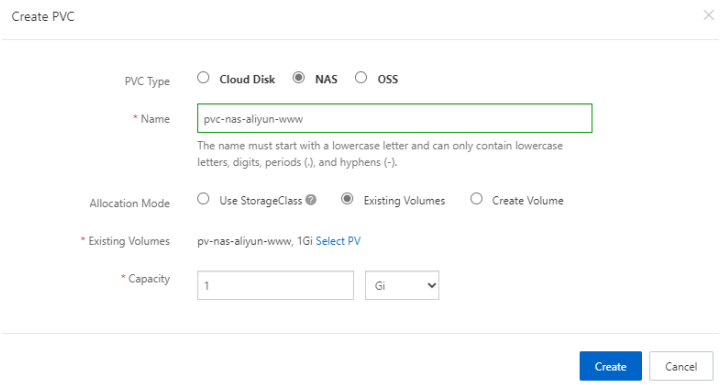


Ilustración 55. Creación de la Petición de Volumen Persistente

Como siguiente paso, se crea el *deployment* mediante el fichero YAML del Anexo 1.

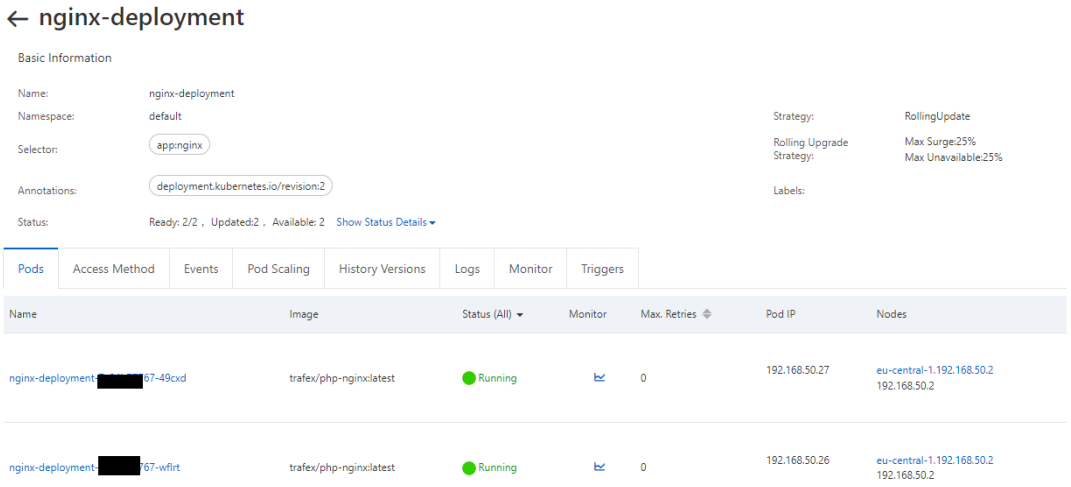


Ilustración 56. Deployment creado en el clúster

Una vez creado el *deployment* y verificado que está operativo, se procede a crear el *Service* que permitirá acceder a la aplicación web.

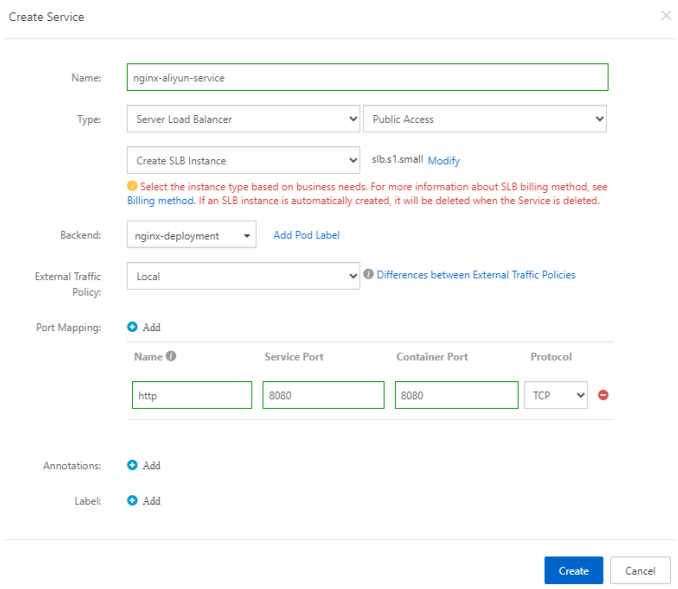


Ilustración 57. Creación del servicio Ingress

Como se puede observar, se han creado dos *pods* para el uso del *deployment*. Al crear el servicio *Ingress*, se proporciona una dirección IP pública con su respectivo puerto mediante el que acceder a través de Internet al *deployment*. Por lo que al acceder desde un navegador a dicha dirección IP, es posible visualizar la página web en cuestión.

Ilustración 58. Deployment en funcionamiento (2 pods)

Al acceder desde dos instancias distintas del navegador (similar a usar dos dispositivos), se observa que se accede a dos *pods* distintos gracias al servicio *Ingress* que actúa como balanceador de carga para el *deployment* realizado, cada uno de los pods carga la misma web a través del volumen persistente que ha sido montado a través del componente de *Persistent Volume* y *Persistent Volume Claim* de Kubernetes.

Como se ha indicado en el punto 5.9, Alibaba Cloud permite realizar un seguimiento del funcionamiento del clúster a través de su propia consola. Entre alguna de las opciones ofrecidas para realizar la observabilidad del clúster, se encuentra la herramienta Prometheus, como se observa a continuación, permite conocer el estado del clúster en tiempo real.

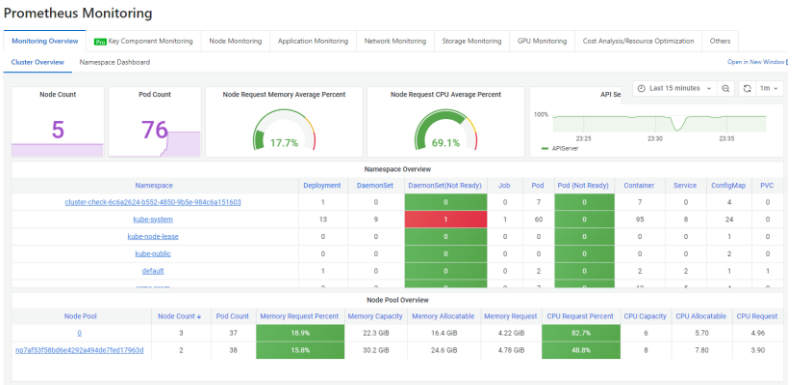


Ilustración 59. Alibaba Prometheus Monitoring

Esto no se limita únicamente a una visión general del clúster, sino que va más allá y permite monitorizar los nodos, las aplicaciones, red, almacenamiento, etc. Cabe destacar que, en la vista de aplicaciones, se puede obtener una visión topográfica del despliegue. A continuación, se muestran ejemplos de monitorización de nodos, aplicaciones y red.

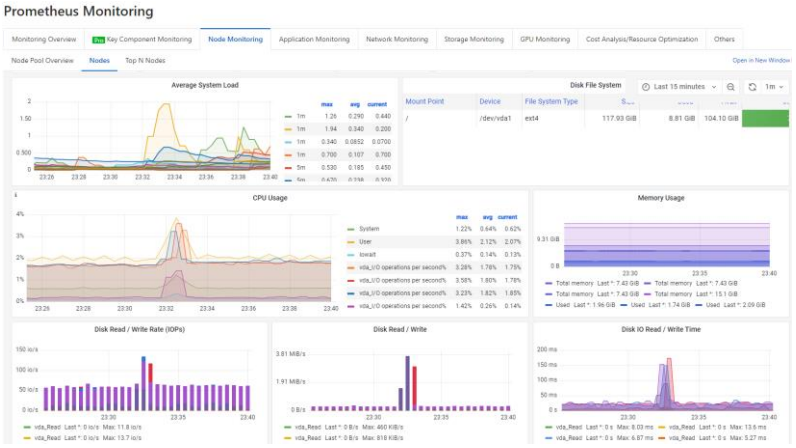


Ilustración 60. Monitorización de nodos

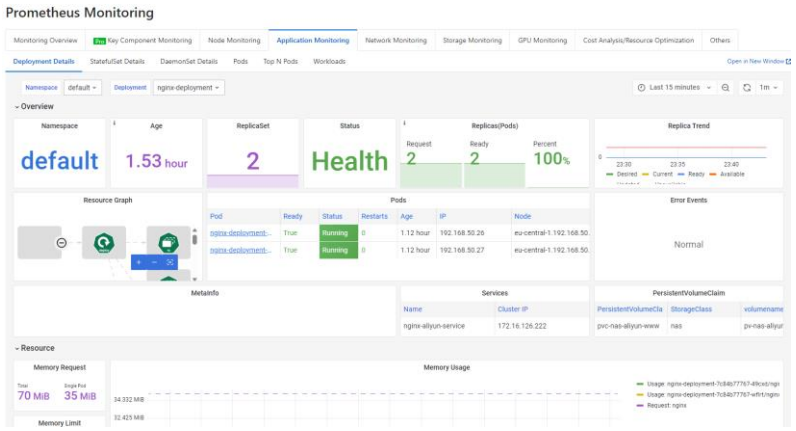


Ilustración 61. Monitorización de aplicaciones

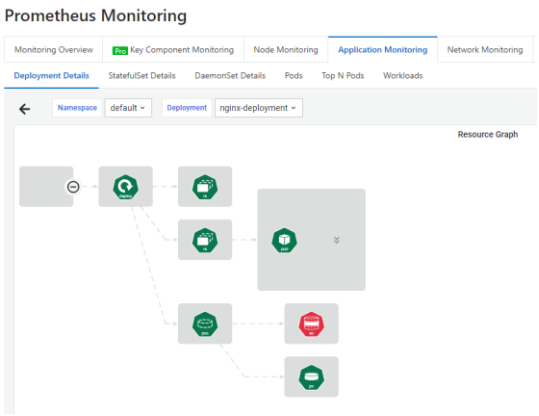


Ilustración 62. Vista topográfica del deployment

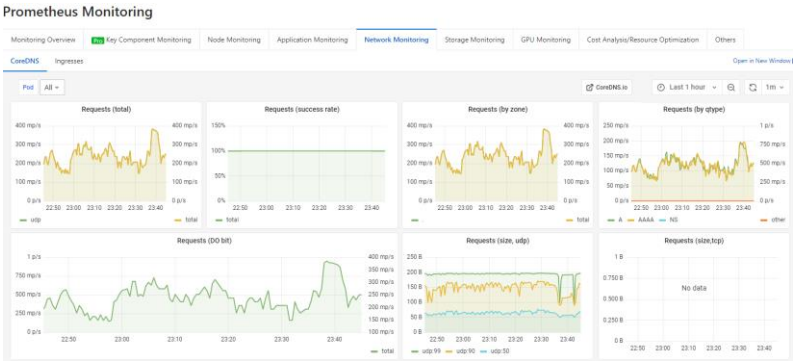


Ilustración 63. Monitorización de red

De la misma forma, es posible tener acceso a los logs del clúster, red y aplicación desde el Centro de Logs como se muestra a continuación para el caso de logs de red.

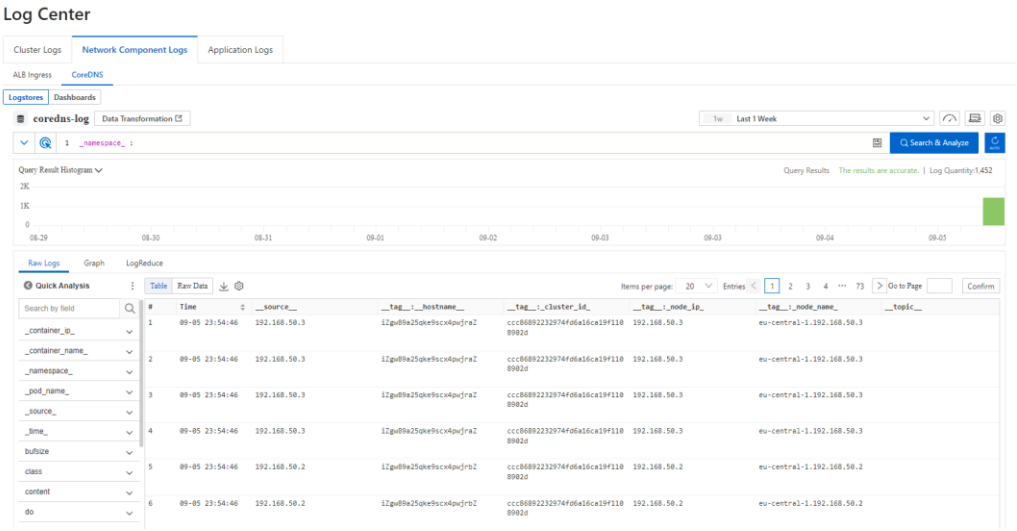


Ilustración 64. Alibaba Log Center

A continuación, se procede a configurar un auto escalado HPA para el *deployment* creado anteriormente. Para ello, es necesario acceder a los detalles del *deployment* y crear un auto escalado, es posible elegir entre los tipos HPA y CronHPA. En este caso se selecciona HPA, dada la capacidad de uso de CPU, se configura con un 4% de umbral para realizar el escalado, con un máximo de 5 réplicas y un mínimo de 2 réplicas.

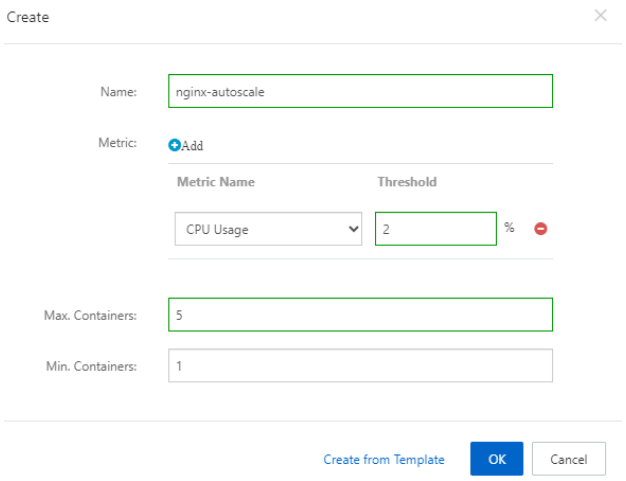


Ilustración 65. Creación del auto escalado

Al ejercer una mayor carga de trabajo en los *pods* existentes, se comprueba que el auto escalado HPA se encarga de aumentar el número de *pods* hasta la cantidad indicada, para una vez reducida la carga de trabajo y pasado un tiempo de enfriamiento, reducir los *pods* hasta su cantidad inicial.

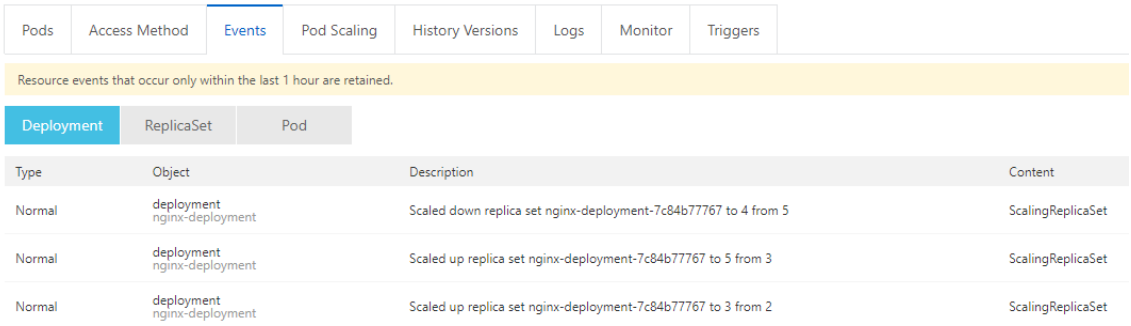


Ilustración 66. Auto escalado HPA en funcionamiento

De igual manera es posible proceder para el escalado VPA o el escalado de los nodos del clúster en caso de resultar necesario debido a la alta demanda del servicio prestado o por su complejidad.

7. Conclusiones y trabajos futuros

Gracias al desarrollo del presente trabajo se puede concluir que se ha cubierto el principal objetivo. Mostrar una alternativa factible al uso de las ya conocidas plataformas de computación en la nube y exponer sus ventajas frente a sus competidores, así como las características que comparte con el resto de las plataformas existentes.

Actualmente el mercado está principalmente centrado en el uso de plataformas occidentales bien conocidas y dominadas por multinacionales que abarcan una gran cantidad del mercado actual, bien por desconocimiento o bien por desconfianza de plataformas que conforman una alternativa real.

Alibaba Cloud, perteneciente al Grupo Alibaba, cuenta con capacidad para poder posicionarse como una alternativa al resto de plataformas, permitiendo ofrecer una serie de herramientas y servicios capaces de abastecer las necesidades que el mercado necesita, siendo estas similares a las que cuentan el resto de las plataformas.

Siendo que Alibaba cuenta también con el respaldo económico de una economía emergente que cada vez cuenta con mayor público y una mayor capacidad de influencia y adquisición, es lógico pensar que cada vez consiga una mayor capacidad de operar y ofrecer mayores y mejores servicios para poder abastecer las necesidades que el mercado demanda.

Como trabajos futuros, es posible ahondar más en todas las posibilidades que ofrece la plataforma Alibaba, ya que esta ofrece una infinidad de posibilidades que permiten el desarrollo de diversos proyectos, sin limitarse únicamente al despliegue de un clúster Kubernetes.

Cada vez son más las empresas que solicitan alojar sus servicios en la nube, aumentando también la cantidad y diversidad de los servicios que necesitan alojarse en la nube. En el caso del presente trabajo, solo se ha ahondado en un clúster Kubernetes, sin embargo, también es posible implementar otra serie de servicios, entre ellos proyectos de inteligencia artificial, IoT, big data, etc. Para lo cual Alibaba Cloud, ofrece una serie de servicios que se adaptan a estas necesidades.

8. Agradecimientos

En este apartado me gustaría mostrar mi más grande y sincero agradecimiento a todas las personas que me han hecho posible llegar hasta aquí.

En primer lugar y de forma muy especial, a mis padres, que me han proporcionado el apoyo a lo largo de toda mi educación, dándome las oportunidades necesarias para poder llegar a este punto, y seguir brindándome su apoyo para superarme cada día y animarme a plantearme nuevos retos y a conseguirlos.

A mis amigos “de toda la vida”, por animarme a continuar con los estudios y por “darme la chapa” para finalizar y avanzar en mi vida universitaria y personal. En especial a vosotros: Edu, Nina, Diana, Gabi, Laura… Sin duda me habéis dado mucho apoyo y en parte, esto es gracias a vosotros.

A todos los compañeros que me han acompañado a lo largo de la carrera, aunque a algunos se les haya perdido la pista, sin duda no habría sido lo mismo sin su apoyo y los buenos momentos que me han proporcionado en esta etapa.

En último lugar, pero con bastante importancia, al Centro de Formación Aula Campus y a su personal docente. Gracias a aquellos que fomentaron mi curiosidad en el mundo de la informática y me proporcionaron una base sólida para afrontar este reto.

9. Bibliografía

Alibaba ACK. (Consulta: 04 de 05 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/es/product/kubernetes

Alibaba ACK Autoscaling. (Consulta: 15 de 07 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/help/en/ack/ack-managed-and-ack-dedicated/user-guide/auto-scaling

Alibaba ACK Dedicated. (Consulta: 14 de 05 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/help/en/ack/ack-managed-and-ack-dedicated/user-guide/create-an-ack-dedicated-cluster

Alibaba ACK Edge. (Consulta: 14 de 05 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/help/en/ack/ack-edge

Alibaba ACK Managed. (Consulta: 14 de 05 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/help/en/ack/ack-managed-and-ack-dedicated/user-guide/create-an-ack-managed-cluster-2

Alibaba ACK Observability. (Consulta: 1 de 08 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/help/en/ack/ack-managed-and-ack-dedicated/user-guide/observability

Alibaba ACK Serverless. (Consulta: 15 de 05 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/help/en/ack/serverless-kubernetes

Alibaba ACK Storage. (Consulta: 01 de 07 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/help/en/ack/ack-managed-and-ack-dedicated/user-guide/storage-1

Alibaba Clean Energy. (Consulta: 13 de 08 de 2022). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/blog/how-alibaba-cloud-data-centers-will-reach-100%25-clean-energy-by-2030_598748

Alibaba Cloud. (Consulta: 05 de 04 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/es/global-locations

Alibaba ECS. (Consulta: 15 de 05 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/es/product/ecs

Alibaba EGS. (Consulta: 15 de 08 de 2023). *Alibaba Cloud*. Obtenido de https://data.alibabagroup.com/ecms-files/1509739361/fcaefa3d-0989-48fb-b003-fa96aa04880e/2023%20Alibaba%20ESG%20Report-Final.pdf

Alibaba Networking. (Consulta: 23 de 06 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/es/product/networking

Alibaba Storage. (Consulta: 03 de 07 de 2023). *Alibaba Cloud*. Obtenido de https://www.alibabacloud.com/es/product/storage

Amazon EC2. (Consulta: 24 de 05 de 2023). *Amazon Web Services*. Obtenido de https://aws.amazon.com/es/ec2/

Amazon EKS. (Consulta: 24 de 05 de 2023). *Amazon Web Services*. Obtenido de https://aws.amazon.com/es/eks/

Amazon Networking. (Consulta: 24 de 05 de 2023). *Amazon Web Services*. Obtenido de https://aws.amazon.com/es/products/networking

Amazon Storage. (Consulta: 24 de 05 de 2023). *Amazon Web Services*. Obtenido de https://aws.amazon.com/es/products/storage

Amazon Web Services. (Consulta: 25 de 05 de 2023). *Amazon Web Services*. Obtenido de https://aws.amazon.com/es/containers/?nc2=h_ql_sol_use_con

Amazon Web Services. (Consulta: 25 de 05 de 2023). *Amazon Web Services*. Obtenido de https://aws.amazon.com/es/about-aws/global-infrastructure/regions_az/

Google GKE. (Consulta: 27 de 05 de 2023). *Google Cloud*. Obtenido de https://cloud.google.com/kubernetes-engine

Herranz, A. (Consulta: 01 de 04 de 2023). *Xataka*. Obtenido de https://www.xataka.com/pro/asi-se-reparte-mercado-cloud-tres-grandes-tienen-63-siguientes-10-22

Horcajo, A. (Consulta: 01 de 04 de 2023). *elEconomista*. Obtenido de https://www.investinspain.org/content/icex-invest/es/sectors/tic.html

Microsoft AKS. (Consulta: 30 de 05 de 2023). *Microsoft Azure*. Obtenido de https://azure.microsoft.com/es-es/products/kubernetes-service/

Richter, F. (Consulta: 20 de 04 de 2023). *Statista*. Obtenido de https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/

Anexo 1. Ficheros de despliegue

persistentVolume.yaml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-www
spec:
  storageClassName: ""
  capacity:
    storage: 1Gi
  accessModes:
  - ReadOnlyMany
  persistentVolumeReclaimPolicy:
  mountOptions:
  - hard
  - nfsvers=4.1
  nfs:
    path: /var/web
    server: 192.168.2.200
    readOnly: false
```

persistentVolumeClaim.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-www-pvc
spec:
  storageClassName: ""
  volumeName: nfs-www
  accessModes:
  - ReadOnlyMany
  volumeMode: Filesystem
  resources:
    requests:
      storage: 1G
```

nginxDeployment.yaml

```yaml
apiVersion: /v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: default
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: trafex/php-nginx:latest
          ports:
          - containerPort: 8080
          volumeMounts:
          - name: nfs-vol
            mountPath: /var/www/html
            readOnly: true
          resources:
            requests:
              cpu: 30m
              memory: 35Mi
        volumes:
        - name: nfs-vol
          persistentVolumeClaim:
            claimName: nfs-www-pvc
```

nginxService.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
 name: nginx-service
 namespace: default
spec:
  selector:
    app: nginx
  type: LoadBalancer
  ports:
  - name: http
    port: 8080
    protocol: TCP
    targetPort: 8080
```

horizontalPodAutoscaler.yaml

```yaml
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
 name: nginx-hpa
spec:
 scaleTargetRef:
  apiVersion: apps/v1
  kind: Deployment
  name: nginx-deployment
 minReplicas: 2
 maxReplicas: 5
 metrics:
  - type: Resource
    resource:
     name: cpu
     target:
      type: Utilization
      averageUtilization: 30
```

<

Anexo 2. Objetivos de Desarrollo Sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	**Alto**	**Medio**	**Bajo**	**No Procede**
ODS 1. **Fin de la pobreza.**				**X**
ODS 2. **Hambre cero.**				**X**
ODS 3. **Salud y bienestar.**				**X**
ODS 4. **Educación de calidad.**				**X**
ODS 5. **Igualdad de género.**		**X**		**X**
ODS 6. **Agua limpia y saneamiento.**				**X**
ODS 7. **Energía asequible y no contaminante.**	**X**			
ODS 8. **Trabajo decente y crecimiento económico.**	**X**			
ODS 9. **Industria, innovación e infraestructuras.**		**X**		
ODS 10. **Reducción de las desigualdades**.				**X**
ODS 11. **Ciudades y comunidades sostenibles**.				**X**
ODS 12. **Producción y consumo responsables.**	**X**			
ODS 13. **Acción por el clima.**	**X**			
ODS 14. **Vida submarina.**				**X**
ODS 15. **Vida de ecosistemas terrestres.**				**X**
ODS 16. **Paz, justicia e instituciones sólidas.**				**X**
ODS 17. **Alianzas para lograr objetivos.**				**X**

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados (Alibaba EGS, 2023) (Alibaba Clean Energy, 2022).

- **Energía asequible y no contaminante,** en mi opinión, la tecnología que se usa para el despliegue de un clúster Kubernetes en Alibaba tiene una relación elevada con el presente ODS. Debido a que el Grupo Alibaba, está realizando una gran inversión en energía limpia.
- **Trabajo decente y crecimiento económico,** creo que al tratarse de un ámbito que hace uso de una tecnología en constante evolución y con un enorme impacto en las empresas a nivel internacional, proporciona una gran cantidad de puestos de trabajo decente, a la vez que promociona un crecimiento económico para sus empleados como para las empresas que se relacionan con Alibaba.
- **Industria, innovación e infraestructuras,** creo que Alibaba Cloud es una de las empresas líderes en el desarrollo de infraestructuras y tecnología puntera, a la vez que se preocupa del impacto ambiental que genera. A su vez, al crear infraestructuras en diversas partes el globo, tiene un impacto en otras regiones del planeta.
- **Producción y consumo responsables,** en mi opinión, Alibaba Cloud, así como el Grupo Alibaba, tiene como objetivo reducir su huella de carbono, llegando a neutralizar por completo la huella de carbono en el desarrollo de sus operaciones internas, reduciendo también el consumo de energías no sostenibles para la producción de servicios del resto de clientes que contratan sus servicios.
- **Acción por el clima,** también es necesario tener en cuenta que el uso de servicios cloud ayuda a reducir la cantidad de energía no renovable que se usa para proporcionar servicios a través de Internet, reduciendo no solo la cantidad de energía que consumen durante su operación, sino también la cantidad de residuos electrónicos, pues son menos los dispositivos que se usan, así como la energía usada para su producción.