



UNIVERSIDAD
NEBRIJA

DESARROLLO DEVOPS & CLOUD

JAIME TELLO SÁNCHEZ

**UNIVERSIDAD NEBRIJA
GRADO EN INGENIERÍA INFORMÁTICA
MEMORIA PRÁCTICAS EN EMPRESA**



UNIVERSIDAD
NEBRIJA

DESARROLLO DEVOPS & CLOUD

JAIME TELLO SÁNCHEZ

**UNIVERSIDAD NEBRIJA
GRADO EN INGENIERÍA INFORMÁTICA
MEMORIA PRÁCTICAS EN EMPRESA**

Profesor asignado de prácticas: Javier García Arcal

PÁGINA DE DERECHOS

D./Dña. JAIME TELLO SÁNCHEZ autoriza a que el presente trabajo se guarde y custodie en los repositorios de la Universidad Nebrija y además SI autoriza a su disposición en abierto.

ÍNDICE

1	Resumen.....	7
2	Objetivos	9
3	ONBOARDING.....	10
4	REVISIÓN BIBLIOGRÁFICA.....	14
4.1	¿Qué es Amazon Web Service?.....	14
4.2	¿Qué es Microsoft Azure?	15
4.3	Diferencias y similitudes entre AWS y Microsoft Azure.....	16
4.4	Proceso de migración de servicios <i>on Premise</i> a AWS según Amazon	23
5	Casos de uso de AWS	27
5.1	Gestión de operaciones centralizada	27
5.2	Monitoreo y observación de los servicios y aplicaciones	27
5.3	Configuración, cumplimientos y auditorías	28
5.4	Gobernanza y control.....	28
5.5	Aprovisionamiento y orquestación del contenido	28
5.6	Administración financiera de la plataforma Cloud	29
5.6.1	Organización y registro del costo y el uso en función de los métodos definidos por el usuario	29
5.6.2	Administración de la facturación y control de costos.....	30
5.6.3	Planificación Optimizada con Previsiones y Presupuestos Flexibles.....	31
5.6.4	Optimización de los costos con recomendaciones sobre recursos y precios	32
5.6.5	Recursos destacados	32
6	Principales dificultades en el proceso de migración en la nube	33
7	Creación de un clúster EKS mediante la interfaz gráfica.....	34
8	Creación de un clúster EKS mediante Terraform.....	38
9	Conclusiones.....	48
10	Líneas futuras	49
11	Presupuesto	50
12	Aplicabilidad	51
13	Anexo.....	52
13.1	Glosario	52
13.2	Diagramas.....	55
13.2.1	Infraestructura general de AWS.....	55
13.2.2	Infraestructura clúster EKS.....	56
13.2.3	Infraestructura VPC	57

14	Agradecimientos	58
15	Bibliografía	59

ÍNDICE DE ILUSTRACIONES

4.3.1	Red de centros de datos de AWS.....	18
4.3.2	Red de centros de datos de Azure	19
4.3.3	Encriptación de datos	21
4.3.4	Modelo de responsabilidad compartida de AWS	22
4.3.5	Modelo de responsabilidad compartida Azure	23
4.4.1	Modelo flujo de datos	23
6.1.1	Modelo migración on premise a cloud	33
7.1	Estructura AWS	34
7.2	Interfaz gráfica de clústeres en AWS	36
7.3	Estructura de nodos clúster EKS de AWS	37
8.1	Creación de las políticas AWS	38
8.2	Configuración de cuentas de AWS mediante terminal	39
8.3	Código yaml, para la creación del clúster	41
8.4	Continuación del código yaml, para la creación del clúster	42
8.5	Continuación del código yaml, para la creación del clúster	44
8.6	Kubernetes dentro del clúster con el despliegue de un servicio	47
11.1	Estimación de costes de AWS	50

1 RESUMEN

Se trata de un trabajo relacionado con las funciones que estoy desempeñando en las prácticas. Esta relación se amplía con las comparaciones entre plataformas *cloud* para albergar los servicios e información de las empresas. En mi caso comparo, Amazon Web Service con Microsoft Azure. Ambas plataformas como se observarán en el trabajo son muy similares, tanto en rendimiento como en costes para los clientes. En mi caso, en la empresa donde estoy realizando la asignatura de prácticas en empresa, trabajamos solo con Amazon Web Service, realizando la migración de servicios de local a *cloud*. Estas funciones de migración las automatizamos utilizando Jenkins. Con los conocimientos adquiridos y complementándolos con información de internet he podido conseguir información suficiente sobre la plataforma, no solo para realizar la comparativa sino, que también se realiza una explicación teórica de cómo crear un clúster EKS con todos sus componentes y dentro del clúster los elementos que el usuario se encontrará. Estos conceptos se complementan con una definición teórica de AWS, de cómo migrar los servicios albergados en un sistema en local a la nube de AWS y con los casos de uso que tiene esta plataforma con los recursos destacados de la misma. Esto suele tener a la hora de migrar una serie de dificultades, como el transporte de los datos o la operatividad continua de ciertas aplicaciones o bases de datos a la hora de realizar el proceso de migración. Para ello se exponen las distintas soluciones que propone AWS y los métodos más utilizados por las empresas.

No nos podemos olvidar del gran beneficio que tienen estas plataformas y por el que las empresas los contratan. Que es el ahorro de costes. Ya que las empresas no necesitan tener recursos físicos ni adquirirlos, ya que, con estas plataformas, cobran por la cantidad de servicios y recursos que utilices. Pudiendo hacer que las empresas optimicen al máximo sus gastos. Además, no solo optimizan sus gastos, sino que también hacen que a nivel económico-administrativo sean unos gastos predecibles y la empresa se pueda hacer a la idea de lo que va a gastar, sin llevarse una gran cantidad de sobrecostos inesperados.

Este trabajo es el correspondiente al primer cuatrimestre de prácticas, siendo un trabajo teórico de investigación, quedando abierto al trabajo del segundo cuatrimestre, en el cual, se llevará a cabo la migración de servicios de un sistema local a AWS (Amazon Web Service), pudiendo observar en casos prácticos las ventajas e inconvenientes que

tiene esta plataforma y los conceptos teóricos que se desarrollan en este trabajo se verán puestos en práctica.

2 OBJETIVOS

Se trata de un trabajo teórico de investigación, sobre temas relacionado con la asignatura de prácticas en el cual, se ponen en práctica los conocimientos que se han ido adquiriendo en los últimos meses en la empresa. Este trabajo está relacionado con los entornos *cloud*, desde una comparativa entre plataformas de distintas compañías, casos de uso de la plataforma de Amazon Web Service, con los distintos servicios que te otorgan y requisitos que se necesitan. También se explica teóricamente como se crea un clúster EKS y los distintos objetos que el cliente se encontrará dentro.

Este trabajo se ha llevado a cabo para poner en práctica los conocimientos, así como adquirir nuevos realizando el proceso de investigación y ponerlos en práctica en la empresa. Respecto a la finalidad del trabajo se trata de mostrar las diferencias entre las plataformas, para que ya sea el propio cliente el que pueda elegir entre Azure o AWS.

En caso de que el cliente elija, AWS, se muestra un modelo teórico de los motivos para realizar una migración y si se decide a realizar dicha migración se le otorgará una serie de directrices, para que pueda migrar los servicios de su sistema en local a la infraestructura de AWS en la nube. Además de proporcionarle una serie de pasos e instrucciones, se le proporciona un modelo de envío de datos a AWS y que fallos y vulnerabilidades se han cometido en el pasado por otras empresas, para que no se vuelvan a repetir dichos fallos.

Respecto al tema que más suele interesar a las empresas, son los costes, que también explicamos el sistema de cobro que tienen dichas plataformas poniendo algunos ejemplos y explicando los beneficios económicos y administrativos que nos encontramos en este tipo de plataformas.

Otros de los objetivos, a nivel personal, por lo que he elegido esta temática de trabajo es por la continuidad y la posibilidad de poder llevarlo a la práctica pudiendo observar las problemáticas y los beneficios de estas tecnologías de forma práctica y pudiendo también demostrar dicha aplicabilidad.

3 ONBOARDING

El primer día de trabajo entré con otro compañero de la universidad. Al principio, el departamento de recursos humanos nos estuvo explicando todo el tema de las políticas de empresa y la privacidad, ya que, para acceder a las cuentas relacionadas con la empresa solo se puede realizar desde los dispositivos que nos han proporcionado. También nos estuvieron informando de todos los servicios que nos proporciona la empresa, como pueden ser las retribuciones en especies o seguros médicos a los que podemos optar cuando terminemos la beca y entremos en plantilla.

Una vez informados de la política de empresa y los servicios que proporcionan, se nos fue explicando el tema de cuentas en diversas plataformas y configuración de contraseñas, que hasta que no nos llegaron los equipos a casa no pudimos configurar.

Tras esa explicación pasamos a otra segunda reunión con B. G., quien iba a ser nuestro instructor directo en la empresa y nos iba a ayudar a aprender estos conceptos nuevos. En esta segunda reunión se trataron aspectos técnicos a los que nos íbamos a dedicar. Cuando nos aclaró como iba a ser la formación y recalcar que una vez terminábamos la formación que podríamos entrar en la plantilla de la empresa como empleados, nos aclaró los horarios y se puso a disposición para cualquier duda que tuviésemos.

Para poder comunicarnos y que nos puedan ayudar utilizamos la herramienta Teams. En la cual todos los días nos conectamos todo el equipo de *interns* de Devops llamado Dailys, en el cual por la mañana se nos asigna tarea y tenemos que explicar a B.G que hicimos el día anterior y que problemas y complicaciones tuvimos. Las complicaciones y problemas en la mayoría de las ocasiones no esperamos al día siguiente para decirlos y que nos ayuden, sino, que los exponemos cuando surgen y no somos capaces de solucionarlos, entonces algún otro miembro del equipo, que no tenga excesiva carga de trabajo o el propio Borja nos echan una mano, para poder continuar realizando la tarea.

Una vez nos llegaron los equipos en esta primera semana nos dedicamos a configurar las cuentas y una vez que ya las teníamos configuradas, empezamos con la formación técnica. Al principio, nos hicieron conectar las máquinas virtuales con un terminal externo que es el MobaXterm y más adelante nos ayudaron a configurar el Bitbucket con la cuenta de la empresa con Eclipse. Como estábamos comenzando, esa primera semana no nos introdujeron mucha carga de trabajo y solo empezamos a ver los conceptos relacionados con los Dockerfiles.

Más adelante ya empezamos a introducirnos en realizar las actividades correlacionadas y sustentadas en los primeros conceptos adquiridos y ya estas actividades dejaron de ser teóricas y ya pasamos a realizarlos nosotros y que cada Dockerfile tuviese una función determinada.

Cuando ya había adquirido los conocimientos y lógica de los Dockerfile, procedimos a la instalación de Minikube y del kubectl y con la definición de los distintos objetos en kubernetes, para saber que realiza cada objeto y en qué estado debe estar para que la aplicación esté operativa.

Dentro de Minikube, lo primero que me tocó realizar es levantar un entorno de WordPress con un entorno para poder programar las bases de datos en MySQL. Esto se llevó a cabo como una actividad para afianzar los conceptos y después se prosiguió con el levantamiento de servicios de Jenkins, PhpMyAdmin, OpenLDAP, Nexus, Sonarqube y Velero.

El levantamiento de cada servicio y configuración de cada uno de ellos llevó bastante tiempo, ya que se deben introducir URLs muy específicas de usuarios y la conectividad entre las diferentes aplicaciones lleva tiempo como saber conectarlas y que, con un mismo usuario, puedas logearte en todos los servicios.

Una vez que terminamos de levantar todos los servicios en local, pasamos a trabajar con AWS. Lo primero que realizamos es la configuración de roles del cada usuario en el apartado IAM. Una vez que teníamos ya configurados los usuarios, pasamos a levantar instancias y grupos de seguridad, configurando las IPs y que solo se tuviera acceso desde las IPs que permitíamos. Una vez que teníamos los grupos de seguridad y las instancias, empezamos a levantar las máquinas de tipo t2 micro, y creando subnets donde levantar dichas máquinas. Para ahorrar en costes, nos enseñaron, a realizar grupos de autoescalado, con el fin que las máquinas estuvieran operativas los días y horas que nosotros deseásemos.

Una vez que teníamos todo esto creado, nos enseñaron que es terraform y nos hicieron repetir toda esta tarea, mediante terraform. Terraform es una herramienta basada en código que te permite acceder y crear todos los servicios de AWS sin usar la interfaz gráfica. Es decir, que mediante código tuvimos que ir realizando toda la tarea que anteriormente realizamos con la interfaz.

Una vez que ya controlábamos esos conceptos, al departamento nos mandaron montar un clúster eks, en el cual, albergase tres máquinas en 2 subnets diferentes y en él levantásemos todos los servicios, que levantamos anteriormente en minikube, utilizando

diferentes *namespaces* para cada servicio y en vez de acceder por *ingress*, accediésemos por *load balancer*.

Una vez que ya tuvimos todos los servicios levantados en el clúster y funcionando correctamente, nos empezaron a enseñar Ansible. Es decir, el aprovisionar desde una máquina padre a dos distintas máquinas hijas con distintas distribuciones de Ubuntu. Este aprovisionamiento, tiene el fin, de que, a partir de cualquier máquina introduciendo una IP y puerto específico, se muestre un texto modificado, mediante HTML.

A partir de esa actividad hemos ido realizando tareas para afianzar conceptos con automatizando el lanzamiento de diferentes *pipelines* en Jenkins. Una vez realizado las tareas de los pipelines, se nos encomendó otra tarea relativa a la seguridad de este, ya que, cualquier usuario podía acceder y ver y modificar todos los proyectos. Para ello estuvimos creando los roles correspondientes dentro de Jenkins y los permisos para que no pudiese acceder nadie que nosotros quisiéramos a la plataforma nuestra.

Otra actividad que se nos encargó fue el que uno de los usuarios del departamento de *Devops Interns*, solo pudiese ser espectador en el clúster, que tiene el departamento. Para ello tuvimos que modificar el fichero de autorizaciones del clúster de AWS (aws auth) y crear un rol específico en otro fichero. Para ello dentro del fichero de autorizaciones tuvimos que modificar el grupo y hacer una llamada al fichero del rol, para que ese usuario solo fuese espectador.

Una vez finalizado todas las actividades y el periodo de formación la empresa contactó conmigo para introducirme ya en un proyecto y así poder poner en práctica todos los conocimientos adquiridos y también para aprender a afrontar problemas en la empresa de una forma más práctica.

Para ponerse en contacto conmigo, me escribió un compañero de RRHH para que le especificase en que campo había estado formándome y me pidió que le enviase mi CV, uno específico que nos hace rellenar la empresa. Una vez que le envié los documentos y algunos datos personales que también me pidió, se puso en contacto conmigo por teléfono y me estuvo especificando en que empresa iba a entrar, a que departamento y quien iba a ser mi responsable dentro del proyecto.

Una vez que ya sabía a la empresa del sector banca y que iba a estar en el departamento de *Blockchain*, me pidieron otra vez, mis datos personales, para poder otorgarme las credenciales de la empresa y las claves para acceder tanto a escritorio virtual, como a mi propia sesión. Este proceso, hasta que me dieron las credenciales

llevó varios días, hasta que me enviaron un email, con las claves y el identificador, para poder acceder a la plataforma de la empresa.

El mismo día que me llegó el correo, me tuve que poner en contacto con el departamento de soporte técnico, ya que, ellos me tenían que dar una guía de como montar el escritorio virtual de la empresa y como me tenía que conectar a él. Como tras diversos intentos de intentar conectar, me salía fallo de IP y que no se había podido establecer una conexión, un compañero de soporte técnico, se conectó conmigo a una llamada y ya conseguimos configurarlo.

Una vez que accedí al escritorio virtual, por primera vez, en el Teams ya tenía un mensaje de mi jefe de proyectos, diciéndome que le llamase cuando tuviese todo configurado y eso es lo que hice. Le llamé, aunque tuvimos problemas de audio, ya que al Teams había accedido por el Internet Explorer y era de Office 365, por lo que para subsanar el problema, procedía descargarme la versión de escritorio, y ya con ella se solucionaron los problemas de audio.

En la llamada que tuvimos, me estuvo informando un poco sobre el papel que iba a desempeñar dentro del proyecto, a que departamento iba a pertenecer y quien iba a ser mi responsable directo y mis compañeros. Y acto seguido tuvimos una *Day/Is* con todos los integrantes del grupo de DevOps. En la cual, nos presentaron a los dos becarios que entrábamos en el proyecto.

Desde entonces me han ido dando permisos a las distintas plataformas como son el Jira o el Confluence o Azure DevOps para poder ir ya realizando tareas y documentándolas.

La primera tarea que se me encomendó fue el actualizar una red Ethereum. Por lo que tuve que ir actualizando cada uno de los nodos hasta por último actualizar el nodo regular.

4 REVISIÓN BIBLIOGRÁFICA

4.1 ¿QUÉ ES AMAZON WEB SERVICE?

Amazon Web Service se trata de una plataforma *cloud*, la cual combina ofertas de Infraestructura como Servicio (IaaS), Plataforma como servicio (PaaS) y herramientas informáticas, abarcando las categorías de *Cloud Computing* con el escalado y el mantenimiento de las instancias EC2, Bases de datos con servicios como Postgress o Wordpress, creación de redes virtuales privadas a través de la nube, aplicaciones empresariales como puede ser Amazon *Workmail*, almacenamiento y gestión de contenido, cuyo servicio principal es el S3, *Bussiness Intelligent*, que sirve para realizar estudios de datos a gran escala y poder analizarlos y poder analizar su flujo, gestión de aplicaciones móviles, ya que, con Amazon Mobile Hub, permiten la creación, testeo y mantenimiento de las mismas, también proporcionan servicios de IOT (*Internet of Things*), para generar conexión y análisis de los dispositivos conectados, herramientas de desarrollo, que proporcionan la autogeneración de código o publicarlo, mediante una un sistema de entrega continua y por último, una herramienta con la que se diferencia de la competencia, que es la seguridad, ya que se puede personalizar desde dentro de la nube en el apartado de IAM. [1][2]

La gran novedad que introdujo al mercado esta plataforma fue el modelo de pago a escala, ya que, cada usuario deberá pagar en función de los recursos que utilizase, para poder proporcionar un almacenamiento, procesamiento o rendimiento acorde a lo que el usuario necesite. Por lo tanto, esto hizo posible que múltiples pequeñas y medianas empresas pudiesen acceder a este tipo de servicios, ya que, económicamente les saldrá más barato a las Pymes que a las multinacionales, debido a un menor uso de recursos.[2]

Actualmente se trata de la plataforma *cloud* más grande del mercado, ya que, tiene más de 175 servicios disponibles, servidores en 81 zonas geográficas y está disponible para 245 países. A esto ha de sumarle que ostenta el 32% de la cuota del mercado en la nube. [3]

Uno de los grandes factores por lo que tiene tanto éxito esta plataforma es, por la seguridad, ya que, tiene servidores por todo el mundo, que se encargan de mantener y estar monitorizando en todo momento la aplicación en la nube. Esto hace que todos los

servicios que tengan contratados los clientes no se caigan nunca. Ya que, como he dicho antes, al tener servidores en todo el mundo, si hay una catástrofe en una situación geográfica específica, no hay problema, ya que tienen una copia de seguridad en otros servidores en otras zonas geográficas.

Otro factor de la seguridad es el tema de la seguridad y privacidad de los datos de los clientes, ya que, las claves, se pueden gestionar desde dentro de la aplicación de cada usuario mediante la función IAM, aunque también se puede dejar administrar automáticamente por la propia Amazon. [11]

Entre los usuarios más reconocidos que tiene estas plataformas, son Netflix, la NASA o la CIA. [5]

4.2 ¿QUÉ ES MICROSOFT AZURE?

Microsoft Azure se trata de una plataforma de computación en la nube de Microsoft, la cual proporciona servicios en la nube. Estos servicios permiten que sus clientes puedan gestionar los datos de su negocio de forma ágil y flexible, estén donde estén, creando una plataforma que se adapta a las necesidades específicas del cliente y que solo, al igual que AWS, cobra a escala de los recursos que se utilizan en la plataforma, por lo tanto, podrás gestionar sencillamente la capacidad de almacenamiento y procesamiento que necesitas. Además, podrá, el cliente, desarrollar aplicaciones web y móviles de forma segura, ya que se basa en los estándares ISO 27001, ISO 27018, SOC 1, SOC 2, SOC3, FedRAMP, HITRUST, MTCS, IRAP y ENS [6] [7].

Azure proporciona servicios tanto IaaS (Infraestructura como Servicio), como PaaS (Plataforma como Servicio) y SaaS (Software como servicio). [7]

La infraestructura Azure como servicio, ayuda a crear rápidamente una infraestructura, reduciendo tiempos e inversiones dedicadas a la planificación, despliegue y gestión. Este despliegue puede ser compatible de forma híbrida compatibles con una infraestructura *on-premise*. [8]

La plataforma Azure como servicio, permite cualquier tipo de servicio, desde una simple aplicación en la nube hasta complejas y sofisticadas aplicaciones de negocio. Adquiere los recursos que necesites con un modelo de pago por uso, y accede a ellos mediante una conexión segura a Internet. [8]

Azure como servicio permiten a los usuarios conectarse a aplicaciones que residen en la nube y acceder a ellas a través de Internet. Ejemplos habituales son las herramientas de mensajería y de calendario, así como las herramientas de ofimática como Microsoft Office 365. [8]

En cuanto a seguridad, las empresas necesitan los mismos estándares de seguridad en local, como en la nube. Además, tiene un sistema de detección de fugas de datos, permitiendo restringir a cada usuario a la información que puede acceder. Esto indirectamente proporciona un servicio de seguridad contra programas maliciosos que intentan suplantar la identidad, para poder acceder a la información. [8]

Esta plataforma al igual, que AWS tiene distintas características y herramientas que proporcionan a los clientes como son las máquinas virtuales, aplicaciones web y móviles, distintos tipos de almacenamiento en la nube y procesamiento en la nube, bases de datos y herramientas de gestión de flujo y estudio de dichos datos, Internet de las cosas para poder controlar todos los dispositivos que están conectados a la red, subredes privadas de dispositivos, servicios multimedia y CDN, herramientas para autenticación de acceso, sincronización de directorios, gestión de identidades y accesos y servicios para poder realizar *backups* en los servidores. [8]

Entre los clientes que utilizan esta plataforma nos podemos encontrar empresas de renombre mundial, como la aseguradora AXA, Ebay, Boeing, Samsung, Media Markt o BMW. [9]

4.3 DIFERENCIAS Y SIMILITUDES ENTRE AWS Y MICROSOFT AZURE

Se tratan de ambas plataformas de computación en la nube, que llegan a ofrecer, una gran variedad de ventajas a los usuarios, desde una potencia de cálculo, escalabilidad y seguridad superiores hasta una rentabilidad y una reducción de la huella de carbono inigualables. [10]

Actualmente, según el último estudio de Canalys y Synergy Research Group, entre Amazon y Microsoft se reparten más de la mitad de los gastos de infraestructuras en la nube. Pese a la pandemia que ha sufrido el mundo a nivel global, ambas empresas han seguido obteniendo un crecimiento económico, incluso con una tendencia mayor de crecimiento, debido al teletrabajo.

Ambas plataformas comparten multitud de características, que son prácticamente idénticas. Sobre todo, me voy a centrar en las máquinas virtuales, que se tratan de una de las partes principales, ya que, emplean las funcionalidades y llevan a cabo la potenciación de las cargas computacionales de trabajo.

La primera diferencia entre ambas plataformas es el acceso a las máquinas, ya que con Azure se accede mediante la consola Powershell de Windows, en cambio, AWS proporciona acceso adicional a máquinas Windows a través de su dirección IPv6 y Session Manager [3]. Para acceder a dichas máquinas virtuales, ambas plataformas, requieren que incluyas, tus propias claves SSH.

Respecto al tipo de instancias de máquinas virtuales ambas plataformas, ya te ofrecen una gran variedad de instancias predefinidas, aunque se pueden modificar, en función de las necesidades que tenga el cliente, pudiendo modificar, la memoria RAM y el número de recursos. Además de poder modificar a gusto del consumidor los recursos, también ofrecen una amplia gama de imágenes para las instancias, pudiéndolas adquirir en sus respectivos *marketplaces* (Amazon Marketplace y Azure Marketplace), aunque aparte de las imágenes de los terceros proveedores, se pueden crear las propias imágenes. Una diferencia en la gestión de imágenes es que Amazon en 2020, introdujo una funcionalidad específica, para llevar un almacenamiento y registro de las imágenes, que se denomina, Amazon Container Elastic Registry.

Al ser ambas plataformas *cloud*, en las cuales se paga dependiendo de las funcionalidades y la cantidad de recursos que se necesitan, las dos plataformas han introducido un sistema de autoescalado de las instancias, a partir del cual, puedes programar automáticamente cuando se levanten y se apaguen las mismas. También te da opción a crear políticas para escalar las instancias en función de métricas específicas como la utilización de la CPU o la longitud de la cola de mensajes y a poder gestionar manualmente cuando creas o eliminas una instancia. En el caso de Amazon se realiza en el apartado de AWS Auto Scaling y en caso de Azure en Azure Autoscale.

Las dos plataformas ofrecen instancias temporales, que son instancias con máquinas que están disponibles de forma imprevisible y puedes reasignar los recursos en cualquier momento, por lo que tienen precios muy reducidos. Las dos plataformas comparten las características del control de las instancias temporales cuando se están utilizando, limitación de los tipos de instancia y de las imágenes de máquina disponibles, en comparación con las instancias bajo demanda y acceder al mismo rendimiento que las instancias bajo demanda cuando se ejecutan instancias temporales.

Para concluir con las máquinas de ambas plataformas, Cockroach, realizó un estudio entre dos máquinas virtuales, con las mismas características e imágenes, en el cual, concluyó que las máquinas de Azure en igualdad de condiciones tenían un rendimiento de un 10% mayor, que las de AWS.

Al ser dos plataformas *cloud* y sin restar importancia a las máquinas virtuales, también se expondrán las distintas características, similitudes y diferencias entre las funciones de red de ambas plataformas.

Lo primero en lo que me voy a centrar es en las ubicaciones donde la red de AWS está disponible. Son más de 24 regiones, 77 zonas, 210 ubicaciones de borde de red y 245 países y territorios. Como resultado, AWS tiene la mayor huella global de infraestructura en la nube que cualquier otro proveedor. Esto provoca que tenga una menor latencia entre los nodos de la red. [10]

Actualmente Amazon tiene ambiciosos planes para seguir ampliando su infraestructura mundial con la puesta en marcha de nuevos centros de datos, como se puede observar en la siguiente ilustración.[10]



4.3,1 Red de centros de datos de AWS

Aunque Azure no tiene una red tan amplia como Amazon, también dispone de múltiples ubicaciones de la red que abarcan más de 60 regiones y 170 puntos de red, aunque Microsoft tiene planes de abrir nuevos centros de datos en distintas ubicaciones, como se puede observar en la ilustración posterior. [10]



4.3.2 Red de centros de datos de Azure

Para convertir los nombres de dominio legibles para el ser humano en las direcciones IP que utilizan los servidores para comunicarse, se utilizan servicios DNS. Ambas plataformas utilizan sus respectivos servicios (Amazon Route 53 y Azure DNS), aunque en este apartado Amazon, cuenta con el enrutamiento basado en la geografía, que le permite restringir el contenido a lugares geográficos específicos, enrutamiento basado en la latencia, o la capacidad de dirigir el tráfico según los niveles de latencia medidos por los servicios de DNS y firma DNSSEC, que proporciona la validación de que una respuesta DNS proviene de Amazon Route 53 y no ha sido manipulada. Estos servicios Azure DNS aun no los soporta.

El equilibrio de carga entre los nodos de las redes en ambas plataformas se realiza de forma automática.

En el caso de AWS se denomina Elastic Load Balancing. Esta es encargada de balancear las cargas entre las instancias del clúster del cliente. Estas instancias, se deben comprobar, para ver el estado en el que se encuentran. El enrutamiento basado en el contenido y SSL disponible a través de un *Application Load Balancer*. Para tener un mayor rendimiento y una menor latencia se realizan las conexiones a nivel 4 de alto

rendimiento a través de un *network load balancer*. Este equilibrador de cargas permite que las direcciones IP sean de versión 6.

Respecto al balanceador de cargas de Microsoft Azure, presenta el Azure Load Balancer, que ofrece, el equilibrio de cargas para tráfico interno y externo a las máquinas virtuales de Azure, mayor capacidad para la distribución del tráfico entre las instancias de las máquinas virtuales dentro y fuera de las zonas, supervisión de las máquinas virtuales, soporta el equilibrio de cargas entre los diferentes puertos, direcciones IP o ambas y por último, también soporta direcciones IP versión 6, utilizando PowerShell y Azure Cloud Shell. [10]

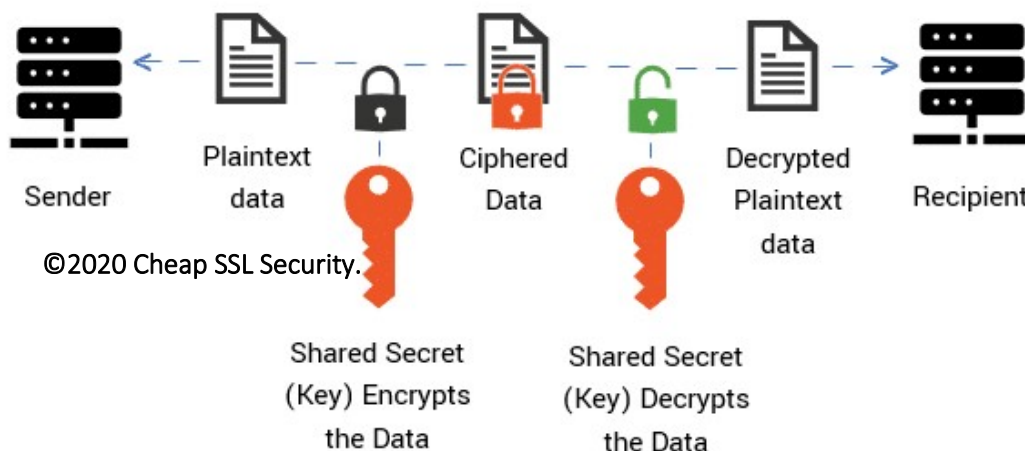
En el apartado de conexión entre redes de diferentes regiones AWS, utiliza AWS Direct Connect, que permite reducir tus costes de ancho de banda al transferir los datos directamente a AWS, minimizando el uso de datos a través de tu actual ISP. Mientras que Azure utiliza Azure ExpressRoute, que es idéntica a la que nos proporciona AWS.

Respecto a la seguridad, se trata de una de las características, de las que ambas plataformas pueden presumir, debido a las medidas que proporcionan a los clientes. Basando las medidas de seguridad en tres pilares principales, que son, en la protección de datos, infraestructuras y sistemas basados en la nube. [10]

Actualmente los gobiernos estatales han ido aumentando las medidas de seguridad, aplicándolas en el almacenamiento, la transferencia y el intercambio de información. Por lo tanto, ambas plataformas cumplen con éxitos los requisitos de seguridad pasando los controles más estrictos como son, CSA STAR, GDPR, HIPPA, PCI-DSS y una serie de normas ISO [10].

Para intentar evitar la fuga de información y la protección de datos, esta se somete a una encriptación casi imposible de descifrar. Para ello se cifran los datos con AES 256 bits, que se trata de uno de los cifrados más potentes y difíciles de descifrar que existen actualmente. Para ello cada plataforma utilizan AWS Key Management y Azure Key Vault que permiten crear claves centralizadas y firmar digitalmente los documentos, permitiendo gestionar toda la gama de cifrados en la nube. [10]

Private Key Encryption (Symmetric)



4.3.3 Encriptación de los datos.

Los servicios de encriptación de ambos proveedores cumplen la norma federal de procesos de información 140-2 (FIPS 140-2), que valida que sus módulos criptográficos cumplen unas normas de seguridad bien definidas. [10]

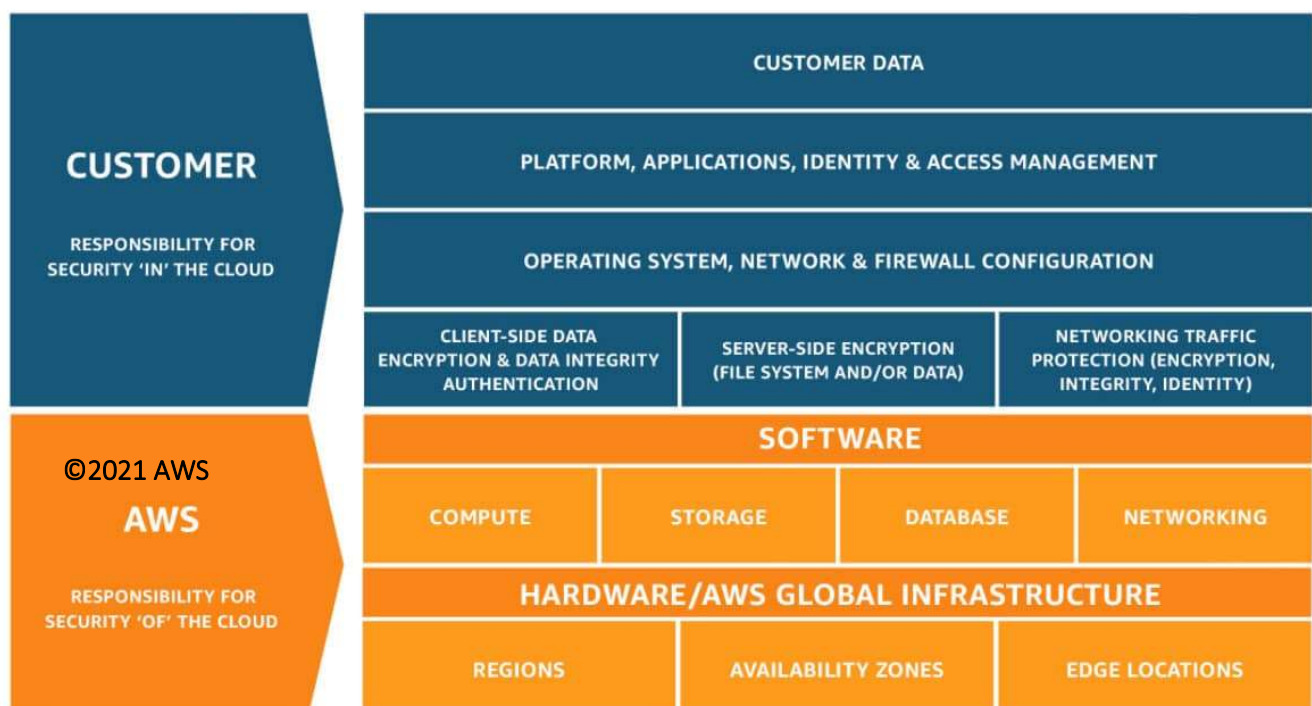
Aunque la encriptación de los datos no es la única medida que presentan, las plataformas, también nos encontramos con el cortafuegos, que sirve para proteger a tus aplicaciones web de los ataques más comunes, sin que tengas que comprometer el rendimiento en exceso. Estos cortafuegos son configurables bajo unas reglas. No solamente hay cortafuegos para las aplicaciones web, sino que también dan la posibilidad de tenerlo para las redes privadas virtuales, centraliza la creación, la aplicación y el registro de tus políticas de aplicación y conectividad de red.

Otro de los principales pilares de la seguridad en la seguridad es la gestión de identidades de quien accede a la información y a que información accede. Para ello ambas plataformas utilizan una consola central a través de la cual se pueden gestionar y asegurar las identidades [10]. Crea y gestiona usuarios y grupos, aplicando permisos para permitir o impedir su acceso a los recursos basados en la nube [10]. En el caso de Amazon, se realiza en el apartado de AWS Identity and Access Management (IAM), mientras que en Azure se realiza en Azure Active Directory (AD). Además, se puede implementar la gestión de identidades y accesos para controlar el registro de usuarios, el inicio de sesión y el acceso a tus aplicaciones móviles y web orientadas al cliente.

Esto se puede controlar en los apartados de Amazon Cognito y Azure Active Directory B2C. Para gestionar y controlar el acceso de los usuarios a varias cuentas y aplicaciones en las instalaciones y en la nube, se utiliza un inicio de sesión único (SSO), denominado en Amazon AWS Single Sign-On y en Azure Active Directory Single sign-on.

Por muchas medidas de seguridad que pongan ambas plataformas, no solamente la responsabilidad es de Amazon y Azure, sino que también el cliente tiene una gran parte de responsabilidad. Para ello, proporcionan una documentación clara sobre su modelo de responsabilidad compartida, lo que le proporciona la información que necesita para entender dónde se encuentra la responsabilidad de la seguridad [10].

En las siguientes ilustraciones se muestran los modelos de responsabilidades compartidas de ambos proveedores.



4.3.4 Modelo de responsabilidad compartida de AWS

Shared responsibility model

©2021 Microsoft



4.3.5 Modelo de responsabilidad compartida de Azure

4.4 PROCESO DE MIGRACIÓN DE SERVICIOS *ON PREMISE* A AWS SEGÚN AMAZON

El proceso de migración de tanto aplicaciones como servicios, se divide en tres fases que están relacionadas entre sí. Para ello, Amazon, pone a servicio de sus clientes, múltiples herramientas, para que los procesos se realicen de forma sencilla y correcta.



4.4.1 Modelo flujo de datos

La primera de las fases es la **evaluación**. En esta fase se debe identificar, cuáles van a ser los resultados que esperan las empresas y que caso de negocio se va a desarrollar. Para ayudar a los clientes Amazon proporciona la herramienta Migration Evaluator, que nos proporciona una proyección del costo total de propiedad (TCO) para AWS en base al uso real de los recursos y a los años de experiencia ayudando a clientes a optimizar su cómputo, almacenamiento, bases de datos, redes y licencias de software en AWS. Cuando solo se necesiten recomendaciones de instancias EC2 de un tamaño adecuado con el fin de ejecutar cargas de trabajo en las instalaciones en AWS, la empresa utilizará la herramienta AWS Migration Hub [11].

Además, en el dentro del Centro de Ahorro en la nube de AWS, se proporcionará ayuda a las empresas, para que puedan desarrollar un caso empresarial, eficiente y fácilmente.

Todas, las guías y herramientas proporcionadas en esta fase se basan en el historial de clientes que ha tenido Amazon y en algoritmos matemáticos y analíticos. Por lo que se ha ido aprendiendo de los clientes y se juega con la experiencia en el sector para poder proporcionar ayuda de forma óptima.

Si las herramientas no han sido de utilidad al cliente, Amazon, también dispone de una serie de guías y catálogos con una serie de recomendaciones.

La segunda fase es la **movilización**. En esta fase se abordarán aquellos aspectos de la preparación de su organización que no se trataron en la fase de evaluación, con hincapié en la creación de su entorno de referencia (la “zona de almacenamiento”), el impulso de la preparación operativa y el desarrollo de habilidades para la nube.[11]

En esta fase se trata de la comprensión de las aplicaciones que vamos a subir a la nube y evaluar las estrategias, para cumplir los objetivos en el caso empresarial. Un aspecto crítico del desarrollo de su estrategia de migración es la recopilación de datos del catálogo de aplicaciones y la racionalización de aplicaciones mediante las siguientes estrategias de migración: reubicación, realojo, reforma de la plataforma, refactorización, readquisición, retiro o retención [11].

Para facilitar la movilización del código al cliente Amazon pone a disposición herramientas. La primera es AWS Application Discovery Service, la cual, recopila y presenta de manera automática información detallada sobre las dependencias y el uso de aplicaciones para ayudarlo a tomar decisiones más fundamentadas al planificar la

migración [4]. Para complementar esta herramienta, AWS, posee soluciones de socios de migración, que proporcionan herramientas de detección exhaustiva y planificación. Para automatiza la planificación y el seguimiento de migraciones de aplicaciones en varias herramientas de socios y de AWS, lo cual le permite elegir las herramientas de migración que mejor se adaptan a sus necesidades, Amazon nos proporciona AWS Migration Hub.

Para sacar el máximo partido de la migración a la nube, necesita darles a sus equipos libertad y agilidad para innovar, pero también aplicar controles que protejan a su organización de riesgos. Gracias a los servicios de administración y gobernanza de AWS, los clientes no tienen que elegir entre innovación y control y pueden mantener ambos aspectos. Los clientes eligen AWS para que les ayude a administrar y gobernar sus recursos, sean o no de AWS. AWS brinda servicios para la administración del ciclo de vida de TI de extremo a extremo, de modo que ayuda a los clientes a controlar y garantizar la seguridad de sus entornos, reducir costos, simplificar la conformidad y mejorar la eficiencia operativa. No solamente AWS, nos proporciona ese servicio, sino que también nos brinda la posibilidad de automatizar la configuración de un entorno de referencia para la ejecución de cargas de trabajo seguras y escalables en AWS, según prácticas recomendadas prescriptivas que habilitan medidas de protección para seguridad, operaciones y conformidad, con AWS Control Tower. Si fuese necesario crear una referencia más personalizada, se puede utilizar AWS Landing Zone, un servicio que brindan los arquitectos de soluciones de AWS o servicios profesionales. [11]

La última fase se trata del proceso de migración y modernización de los servicios que queremos pasar a *Cloud*. Durante esta fase, se diseñan, migran y validan cada una de las aplicaciones que vamos a subir a la plataforma. Para ello, AWS, nos pone a disposición multitud de herramientas.

La primera se trata de Migration Hub, que es utilizada para comprobar si todas las aplicaciones funcionan correctamente y si se encuentran actualizadas.

También nos encontramos herramientas para realizar la migración del código. Para poder realojar servidores, desde una infraestructura local a AWS, utilizamos AWS Application Migration Service, la cual, estandariza el formato de los servidores automáticamente a un formato estándar de AWS. Esto facilita al usuario la migración de la aplicación. Aunque en ocasiones no se puede instalar un servicio de migración basado en agentes en el servidor. Para ello deberemos utilizar AWS Server Migration Service, un servicio sin agente que facilita y acelera la migración de miles de cargas de

trabajo en las instalaciones a AWS desde una instantánea del servidor existente. Si cuenta con entornos basados en VMware Cloud Foundation, VMware Cloud on AWS le permite reubicar rápidamente cientos de aplicaciones virtualizadas en vSphere a la nube de AWS en solo días, y al mismo tiempo mantiene operaciones consistentes con su entorno en las instalaciones.

Si en vez de migrar un servicio necesitamos migrar una base de datos, utilizaremos el servicio AWS Database Migration Service, que nos permite que la base de datos siga estando operativa en el momento de la migración. Por lo tanto, para empresas que necesitan estar con las bases de datos constantemente activas, no le supondrá un impedimento, la migración de esta.

Además, disponemos de AWS Marketplace, donde podemos comprar, implementar y administrar más de 7000 listas de software de terceros y servicios de más de 1500 proveedores de software independientes para crear soluciones para los clientes.[11]

Para la transferencia de datos en la nube encontramos diferentes formas de realizarlos, según la cantidad o el tipo de datos que queramos transferir. Para realizar una transferencia de datos de forma rápida entre las soluciones de almacenamiento en las instalaciones y Amazon S3, Amazon Elastic File System (Amazon EFS) o Amazon FSx for Windows File Server, utilizaremos AWS DataSync. Mientras que para transferir archivos directamente hacia y desde Amazon S3 mediante el Protocolo de transferencia segura de archivos (SFTP), el Protocolo de transferencia de archivos sobre SSL (FTPS) y el Protocolo de transferencia de archivos (FTP), tenemos AWS Transfer Family. Además, AWS ofrece los productos Snow, que se tratan de dispositivos físicos y puntos de capacidad, que la mayoría ya tienen las capacidades informáticas integradas.[4]

Para optimizar todos los servicios, ir actualizando los servicios contratados y eliminando y desactivando los servicios que ya no se utilizan, disponemos de AWS Managed Services, con lo que nos proporcionará una correcta optimización tanto en costes como en rendimiento de nuestros servicios contratados en AWS.

A medida que sus aplicaciones se transfieren a la nube, puede utilizar Service Catalog para crear y administrar catálogos de servicios de TI aprobados para su uso en AWS. En estos servicios de TI se incluye todo lo relacionado con imágenes de máquinas virtuales, servidores, software y bases de datos para completar las arquitecturas de aplicaciones multinivel. AWS Service Catalog permite administrar de forma centralizada los servicios de TI más implementados y ayuda a lograr una gobernanza uniforme, así como a cumplir los requisitos de conformidad, a la vez que permite a los usuarios implementar con rapidez solo los servicios aprobados de TI que necesitan. [11]

5 CASOS DE USO DE AWS

5.1 GESTIÓN DE OPERACIONES CENTRALIZADA

Este caso se da en empresas que tienen una infraestructura híbrida en varios equipos. Por lo que Amazon, da la oportunidad de administrar controlar y operar, con todos ellos bajo una única interfaz. La automatización y centralización de las operaciones, conlleva un mayor rendimiento en el desarrollo de estas y mejorar indirectamente los resultados comerciales de la compañía. No solo mejora el rendimiento y consecuentemente los resultados comerciales, sino que al tener un control centralizado mejora la seguridad y la administración de acceso a cada una de las instancias [12].

5.2 MONITOREO Y OBSERVACIÓN DE LOS SERVICIOS Y APLICACIONES

Con los servicios que proporciona AWS se puede mejorar el rendimiento y la disponibilidad de los recursos, pudiendo supervisar, automatizar y poner recordatorios a los servicios y aplicaciones que estén alojadas en la nube [12].

Con este servicio se intenta detectar cuando un servicio o aplicación esta fuera de los umbrales del rendimiento óptimo, ya que, o lo observa el propio cliente o salta una alarma para avisar. Estos avisos irán ligados a unos perfiles que se pueden crear y modificar a gusto del consumidor. Amazon también pone a disposición herramientas de telemetría para cada perfil, pudiendo recopilar los datos analíticos de los servicios y aplicaciones.[13]

Amazon permite realizar un monitoreo tanto de las aplicaciones y servicios en la nube, como también de su propia infraestructura y de las instancias en la que están albergados. [14]

Con todas estas medidas, se trata de evitar anomalías en los datos de los clientes y antes de que puedan ser un potencial problema para el cliente, se puedan subsanar.

5.3 CONFIGURACIÓN, CUMPLIMIENTOS Y AUDITORÍAS

Este caso de uso está relacionado, en gran parte con las legislaciones de cada país. Por lo tanto, Amazon automatiza la gestión del cumplimiento y supervisa sus recursos, puede proteger sus datos y salvaguardar sus procesos comerciales. [14]

Con ello nos aseguramos una supervisión continua de la gestión de riesgos y del cumplimiento de permisos, que deberemos cumplir, así como una actualización de estos cuando sea necesario, ya que, nos permite la opción de llevar un monitoreo de estos de forma centralizada y agilizar dichos procesos cuando corresponda, ya que, Amazon utiliza procesos de auditoría con plantillas basadas en PCI-DSS, HIPAA y FedRAMP; registro automatizado y protección criptográfica de la actividad de los usuarios y registros de API; además de una recopilación continua de pruebas de cumplimiento en toda la organización del cliente. [15]

5.4 GOBERNANZA Y CONTROL

Amazon nos permite crear un entorno de varias cuentas, seguro y administrado de forma centralizada. Aumentando la agilidad para controlar la gobernanza de forma más automatizada. Con la creación de los entornos y grupos, se pueden establecer y crear políticas de empresa a sus necesidades en AWS. Cada entorno grupo o cuenta se puede personalizar aplicando las políticas creadas y automatizaciones de AWS correspondientes. [16]

5.5 APROVISIONAMIENTO Y ORQUESTACIÓN DEL CONTENIDO

AWS nos permite crear, aprovisionar y compartir recursos de terceros clientes. Para ello ofrece servicios de escalado mediante el aprovisionamiento de recursos de forma coherente y repetible [17]. Para la gestión del contenido Amazon ofrece un catálogo de servicios (AWS Service Catalog), donde podemos encontrar herramientas en forma de código.

Con AWS CloudFormation se le proporciona al cliente la posibilidad de crear políticas sobre el contenido de terceros, teniendo así la gobernanza de este contenido y poder restringir tanto el acceso como la lectura y escritura en los mismos. Como se comenta antes, al ser de forma centralizada, se agiliza en los procesos de control sobre la información.

5.6 ADMINISTRACIÓN FINANCIERA DE LA PLATAFORMA CLOUD

Con la aparición de plataformas como AWS o Azure, las empresas han pasado de tener unos costes fijos por unos servidores y centros de datos físicos a unos costos variables, pagando solo por los servicios que utilizan en la nube. Por lo tanto, esto para la empresa supone una optimización en los costes y permite una mejor planificación de estos. Para ello Amazon, pone a disposición un conjunto de herramientas, servicios y recursos para organizar y rastrear los datos de costo y uso, mejorar el control mediante la facturación unificada y el permiso de acceso, permitir una mejor planificación mediante el presupuesto y las previsiones, y reducir aún más el costo con la optimización de los recursos y los precios.

Para administrar los costes, AWS propone diferentes **soluciones**:

5.6.1 Organización y registro del costo y el uso en función de los métodos definidos por el usuario

AWS proporciona herramientas para organizar los recursos en función de las necesidades, visualizando y analizando los datos de coste y uso en un único panel de vidrio, y cargarlos con precisión a las entidades adecuadas. En vez de fiscalizar el coste de forma centralizada, el cliente puede proporcionar datos de costes en tiempo real que son de interés para sus equipos de ingeniería, de aplicaciones y de negocios. Los datos de costos detallados y asignables permiten a los equipos disponer de la visibilidad y los detalles para rendir cuentas de sus propios gastos. [18]

Los beneficios que nos proporcionan son los siguientes:

- **Visión consolidada:** Permite administrar de manera centralizada la facturación y pago para múltiples cuentas, y la monitorización de las tendencias para comprender que nos está generando los costos y cualquier factor preocupante. También nos permite analizar datos en una vista agregada o por varias dimensiones, y descargar informes o mostrar datos de costos y nos permite el uso de las herramientas de *Business Intelligence*, para obtener más información consumible.
- **Datos significativos:** Define estrategias de etiquetado y asignación que se relacionen con la estructura de su negocio y el proceso de gestión financiera, y vincula los datos de costos y los relaciona con los resultados comerciales. Utiliza una variedad de recursos organizativos y de etiquetado de AWS para estructurar sus cuentas y recursos, y habilitar informes de uso y costos de nivel más detallados.
- Nos permite que se atribuya el costo y el uso a las entidades adecuadas para evaluar el rendimiento y la rentabilidad, priorizar y ajustar las decisiones de inversión en consecuencia. También se puede modificar su diseño arquitectónico, la estrategia de precios de los productos y el modelo de asignación de costos para maximizar los márgenes de beneficio teniendo en cuenta la información sobre los costos del usuario final. [18]

5.6.2 Administración de la facturación y control de costos

Los clientes de las organizaciones necesitan una forma sencilla y fácil de acceder a la información de facturación de AWS, incluyendo un resumen de gastos, un desglose de todos los costes de servicios incurridos por las cuentas de toda la organización, junto con descuentos y créditos. El cliente puede optar por unificar sus facturas y aprovechar los descuentos de mayor volumen basados en el uso agregado de todas sus facturas. [18]

Esto genera diferentes beneficios, como pueden ser:

- **Modelo de responsabilidad compartida:** Centraliza la propiedad de las políticas de gobierno de gestión de facturación y control de costos, al tiempo que designa equipos para implementar los mecanismos de gobierno e impulsar las mejores prácticas.
- **Umbrales financieros:** Ya sea que se trate del permiso para usar los recursos de AWS, acceder a la información de facturación y costos, o establecer los límites

objetivo de gasto, uso y la progresión de la utilización del plan de compra, puede definir sus umbrales financieros deseados.

- Autogobernanza: AWS se encargará de supervisar el gasto y la utilización de los recursos y le avisará cuando se superen los límites presupuestarios o se produzcan anomalías. Esto permitirá cumplir las políticas de gobierno y evitará gastos no intencionales autorizando acciones automáticas de control de costos.[18]

5.6.3 Planificación Optimizada con Previsiones y Presupuestos Flexibles

La plataforma de AWS permite a los clientes adquirir y dar de baja los recursos de manera continua, sin depender de los equipos para aprobar, adquirir e instalar la infraestructura. Sin embargo, esta flexibilidad requiere que las organizaciones se adapten al nuevo y dinámico proceso de previsiones y presupuestos. AWS proporciona previsiones basadas en su historial de costes y uso y le permite establecer un límite de presupuesto y alertas, para que pueda estar informado siempre que el coste y el uso se prevea o sobrepase el límite de presupuesto. Además, puede establecer objetivos de utilización de reservas y/o cobertura para sus Instancias Reservadas y Savings Plans y supervisar el progreso de estos hacia su objetivo. [18]

Los beneficios que nos proporcionan son los siguientes:

- Guía del ciclo de la vida de los recursos de planificación: Las herramientas y los recursos están disponibles a lo largo de su ciclo de planificación de IT. Se puede elegir entre una variedad de herramientas de planificación de autoservicio o consultar con expertos de AWS para que los usuarios puedan crear sus propias estrategias de migración y previsión que mejor se adapten a sus necesidades comerciales específicas.
- Decisiones basadas en datos: Nos permite observar los datos en vivo y poder analizarlos en función de su situación actual o para el uso que se necesite y los escenarios futuros. Por lo tanto, las mediciones económicas serán mucho más fiables y exactas.
- Planificación dinámica: El cliente podrá adaptar sus procesos de planificación y presupuestación con configuraciones dinámicas y personalizables para satisfacer sus necesidades continuas de adquisición y desaprobación de IT. Esto conlleva la mejora de la precisión de la previsión y el control de la presupuestación con la agilidad que necesita para su gasto variable en IT. [18]

5.6.4 Optimización de los costos con recomendaciones sobre recursos y precios

Gracias a las plataformas como AWS o Azure, los costos se han reducido drásticamente, debido a que se paga por los recursos que se utilizan de la plataforma. Para que el cliente pueda elegir los recursos óptimos, Amazon, aconseja la adquisición de una serie de recursos en función de las necesidades del cliente, aunque si ve que los recursos que se han contratado no son óptimos para la empresa, ya sea, porque sobran recursos o porque faltan, también le da la posibilidad al cliente de modificar los recursos contratados. Estas modificaciones las basan en los modelos de clientes anteriores que han ido teniendo a lo largo de los años. Consecuentemente, esto se ve reflejado en una optimización de costes, haciendo que las empresas se puedan ahorrar el dinero de los recursos que no utilizan o son demasiado potentes y costosos para ella.

[18]

5.6.5 Recursos destacados

Para que los clientes no tengan unos gastos que no esperen, Amazon pone a disposición herramientas, para que en todo momento el cliente pueda ver lo que lleva gastado y lo que se va a gastar con los recursos que tiene en ese momento. Además, presenta un programa llamado Saving Plans con precios flexibles que permite que el cliente se pueda ahorrar de media hasta un 72% en el uso de herramientas informáticas. Para ello se utilizará el servicio que pone Amazon a disposición que es el AWS Cost Explorer.

Para que el usuario aprenda a cómo sacar el máximo partido a los recursos al menor precio posible, AWS, también pone a disposición una serie de guías y blogs en los cuales va explicando cómo puedes optimizar el costo en la factura y como maximizar el uso de los recursos que tienes contratados.

6 PRINCIPALES DIFICULTADES EN EL PROCESO DE MIGRACIÓN EN LA NUBE



6.1.1 Modelo de migración on premise a cloud

En el proceso de migración de una plataforma *on premise* a Cloud, encontramos ciertas dificultades al migrar bases de datos de gran capacidad, ya que, se puede extraviar alguno de los datos. Para ello hay algunos clientes, que prefieren enviar a la plataforma en un dispositivo físico en el cual, se encuentra la base de datos, para asegurarse la integridad de los datos. Este proceso se complica aún más cuando la base de datos debe estar activa constantemente. [19] Para ello AWS nos proporciona un servicio específico, aunque descartamos la opción de migración, de enviar un dispositivo hardware al proveedor. Otra solución al necesitar un servicio operativo continuo es la previa copia de los datos, para posteriormente, realizar ya la migración de estos, superponiendo los servicios del cliente y del proveedor. Estos son la minoría de los casos, ya que, las empresas suelen realizar las migraciones de forma escalonada, trasladando pequeñas cantidades de información, en vez de, todo de una única vez.

Otra de las principales dificultades es la seguridad e integridad de los datos, ya que, se debe asegurar tanto el cliente como el proveedor que los datos llegan al destino de forma segura e intacta. Esta seguridad se ve afectada en el caso de enviar un dispositivo físico al proveedor, ya que, dejamos expuestos los datos a un posible hurto o copia de estos, dejando en una situación muy expuesta a la empresa. Este método, debido a lo expuesto que se dejan los datos, solo se pudo llevar a cabo con información que no sea muy relevante para la empresa.

7 CREACIÓN DE UN CLÚSTER EKS MEDIANTE LA INTERFAZ GRÁFICA

Para la creación de un clúster EKS mediante a interfaz de Amazon antes debemos tener configuradas las VPCs y los grupos de seguridad.

Una VPC es una red privada, en la cual, la empresa tiene libertad de configuración, brindando control pleno sobre su entorno de redes virtuales, incluidas la ubicación de los recursos, la conectividad y la seguridad. Además, se puede definir cómo se comunican sus VPC entre sí, entre cuentas, zonas de disponibilidad (AZ) o regiones [20].



7.1 Estructura AWS

Para crear el clúster con la consola en primer lugar deberemos acceder al apartado de EKS y dentro de este menú, deberemos seleccionar la opción de clúster.

Una vez el cliente se encuentre dentro de la venta del clúster, deberá elegir la opción en la que se indica crear clúster. Cuando elija dicha opción deberá rellenar los campos correspondientes al nombre, versión de kubernetes, cifrado de secretos, que es opcional, rol de servicio, que vendrá dado por los roles que tenga el usuario en el apartado de IAM y las etiquetas, que es opcional.

Cuando el cliente acciona la opción de siguiente, Amazon le da la opción de especificación de red. Dentro de esta opción podremos seleccionar las opciones de

VPC, subredes, grupos de seguridad, rango de las direcciones IP y puntos de acceso al clúster.

Dentro de las VPCs solo le dará la opción Amazon de seleccionar las existentes. Mientras que las subredes por defecto se asignan predeterminadamente las correspondientes a las VPCs. Puede desactivarse cualquier subred que no quiera que aloje recursos del clúster, como nodos de trabajo o balanceadores de carga.

Respecto al grupo de seguridad, este se generó, cuando se crearon las VPCs, este grupo de seguridad tiene `ControlPlaneSecurityGroup` en el nombre desplegable [21].

Para poder configurar el rango de servicios de direcciones IP, deberemos especificar un rango de servicios IPv4. Para ello debe de cubrir el bloque CIDR los siguientes requisitos:

Direcciones IP dentro de los rangos: 10.0.0/8, 172.16.0.0/12 o 192.168.0.0/16

Debe estar incluido entre /24 y /12.

No se superpone con ningún bloque CIDR especificado en su VPC. [21]

Se recomienda especificar un bloque CIDR que no se superponga con otras redes que estén conectadas o emparejadas con la VPC. Si no habilita esto, Kubernetes asigna direcciones IP del servicio de desde los bloques de CIDR 10.100.0.0/16 o 172.20.0.0/16. [21].

Para la configuración del punto de acceso del clúster Amazon nos brinda dos opciones, pública, privada o pública y privada.

La pública permite sólo el acceso público al punto de enlace del servidor de Kubernetes del clúster. Las solicitudes de la API de Kubernetes que se originan desde fuera de la VPC del clúster utilizan el punto de enlace público. De forma predeterminada, se permite el acceso desde cualquier dirección IP de origen. Opcionalmente, puede restringir el acceso a uno o más rangos CIDR, como 192.168.0.0/16. [21]

La privada permite solo el acceso privado al punto de enlace del servidor de la API de Kubernetes del clúster. Las solicitudes de la API de Kubernetes que provengan de dentro de la VPC del clúster utilizan el punto de enlace de la VPC privada. [21]

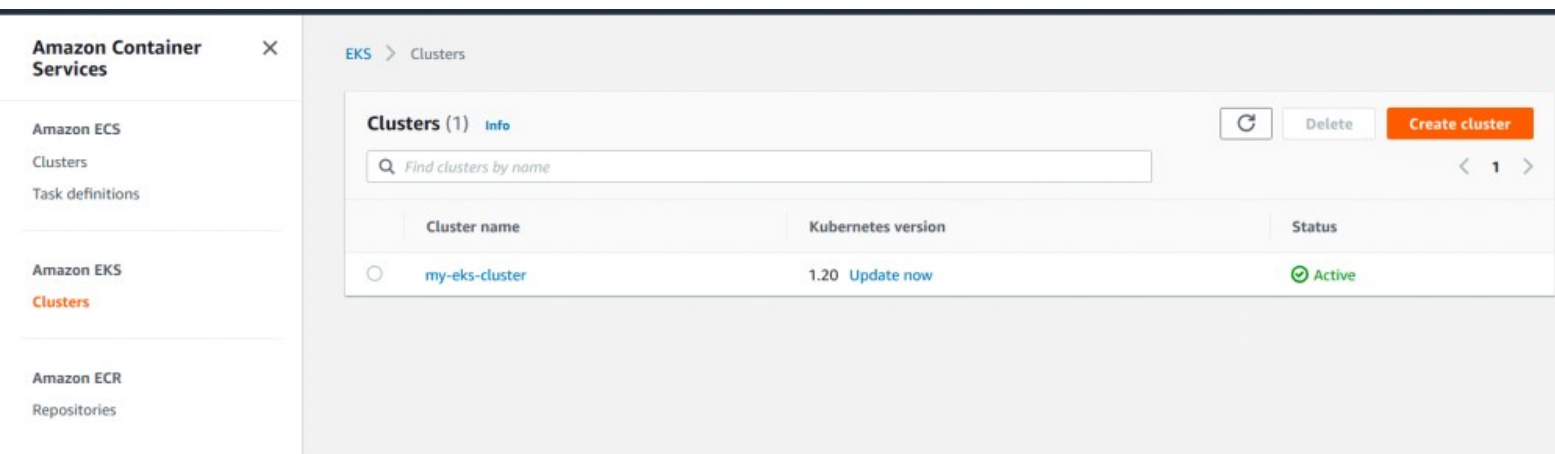
La opción pública y privada permite el acceso público y privado. [21]

Una vez que el cliente finaliza de configurar el punto de acceso, procede al siguiente menú, para configurar el registro. El cliente puede elegir opcionalmente los tipos de

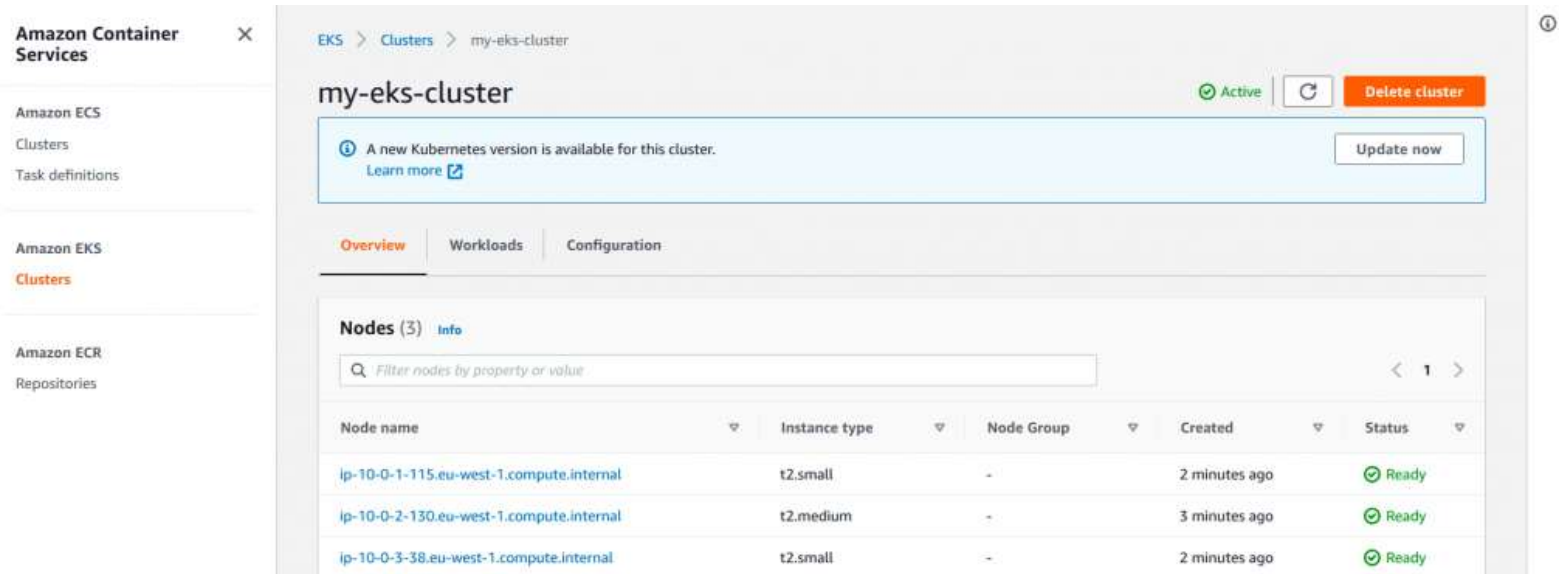
registro que desea habilitar [21]. De forma predeterminada, los tipos de registro están deshabilitados.

Una vez que el cliente prosigue al siguiente menú, ya se encontrará en el último, el cual, es el de revisar y crear. En el cual, el cliente revisa que la configuración que ha ido realizando del clúster es correcta. Si desea cambiar algo, debe pulsar la opción de editar y lo modifica a su gusto, mientras que si está satisfecho elegirá la opción de crear. La creación del clúster llevará varios minutos.

Para crear dentro del clúster los nodos el usuario deberá tener configurados los roles en el apartado de IAM de AWS con una política asociada al rol, que permita que cree los distintos nodos del clúster.



7.2 Interfaz gráfica de clústeres en AWS



7.3 Estructura de nodos clúster EKS de AWS

Como se puede apreciar en las imágenes superiores se puede apreciar en la primera, los clústeres EKS que el cliente tiene, la versión y si los tiene activos o si están desactivados. Mientras que en la segunda el cliente ya se encuentra dentro del propio clúster. En el cual podemos ver la cantidad de nodos que hemos configurado, el tipo de instancia de cada nodo, que esa la habrá configurado el cliente en función de sus necesidades, al grupo de nodos que pertenece, que es opcional, cuando se creó cada instancia y el estado si está activa o está caída.

8 CREACIÓN DE UN CLÚSTER EKS MEDIANTE TERRAFORM

Como he comentado en ocasiones anteriores un EKS (Elastic Kubernetes System) se trata de un servicio de kubernetes alojado en un servidor de AWS. La creación mediante Terraform, conlleva el realizar previamente ciertos requisitos, para que se pueda llevar a cabo correctamente.

Lo primero que debe asegurarse el usuario es que el servicio de creación de clúster EKS está incluido en el plan que tiene contratado con AWS, ya que las plantillas gratuitas no lo incluyen.

Para crear el clúster lo primero que tenemos que realizar es la creación de una política mediante la interfaz de Amazon, en la cual, nos aseguraremos de ponernos como *developers*, para poder ejecutar las funciones correspondientes. Cuando el cliente proceda, le aparecerá una ventana como la de la siguiente imagen, mostrándole los permisos y sobre que tiene premisos. También en esta venta deberá asignar un nombre a la política.

The screenshot shows the AWS IAM 'Create policy' console. The 'Review policy' step is active, showing a policy named 'eks-terraform-user-policy'. Below the name field is a description field. A summary table lists the permissions granted by the policy:

Service	Access level	Resource	Request condition
Allow (6 of 285 services) Show remaining 279			
CloudWatch Logs	Limited: List, Write	All resources	None
EC2	Full: Tagging Limited: List, Read, Write	All resources	None
EC2 Auto Scaling	Full: List, Read, Tagging Limited: Write	All resources	None
EKS	Full: Tagging Limited: List, Read, Write	All resources	None
IAM	Limited: List, Read, Write, Permissions management, Tagging	All resources	None
KMS	Limited: List, Read, Write, Permissions management	All resources	None

At the bottom, there are buttons for 'Cancel', 'Previous', and 'Create policy'.

8.1 Creación de las políticas AWS

Una vez que tenemos la política creada se la podemos asignar directamente al usuario o crear un rol, asignársela al rol y ese rol asignárselo al usuario. La ventaja es que, la política si la asignas directamente solo la puedes asignar a personas, mientras que los roles se los puedes asignar directamente a un grupo. [22]

Una vez que tenemos configurados los usuarios, roles y políticas, el usuario deberá configurar el AWS CLI, esto se trata de códigos internos de AWS, para poder ejecutar comando y poder trabajar con Terraform. Este proceso se realizará mediante la terminal ejecutada en modo administrador y los siguientes comandos:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

Este primer comando sirve para descarga la versión de AWS CLI.

```
unzip awscliv2.zip
```

Este comando sirve para descomprimir el archivo.

```
sudo ./aws/install
```

Este comando sirve para instalar el paquete descomprimido de AWS CLI

Una vez tengamos ya instalado AWS CLI ejecutaremos el comando *aws configure* y se pedirá el ID, la clave de acceso, la región y el formato en el que vaya a trabajar el cliente, tal y como aparece en la ilustración posterior.

```
$ aws configure
AWS Access Key ID [None]: <your-AWS-USER-ACCESS-ID>
AWS Secret Access Key [None]: <YOUR-SECRET-KEY>
Default region name [None]: eu-west-1
Default output format [None]: json
```

8.2 Configuración de cuentas de AWS mediante terminal

Una vez que ya están configurados los comandos de AWS, el cliente deberá instalar el comando *kubectl* en su terminal. Para ello desde el terminal ejecutado como administrador, deberemos introducir los siguientes comandos.

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

Este primer comando sirve para descargar el paquete de *kubectl*.

El siguiente comando se utilizará para instalar el paquete descargado con el anterior.

```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

Una vez que ya tenemos descargado y configurado el comando para poder administrar nuestros servicios en el clúster EKS, deberemos instalar el paquete de Terraform para que se reconozcan los archivos. Esto también se realizará a través de la terminal mediante comandos. [22]

```
TER_VER=`curl -s https://api.github.com/repos/hashicorp/terraform/releases/latest |  
grep tag_name | cut -d: -f2 | tr -d '\n,\v' | awk '{\$1=\$1};1`
```

```
wget
```

```
https://releases.hashicorp.com/terraform/${TER_VER}/terraform_${TER_VER}_linux_a  
md64.zip
```

Estos dos comandos sirven el primero para clonarse el repositorio de GitHub, mientras que el segundo es para descargar el paquete directamente del repositorio.

Con el siguiente comando procederemos a la extracción del archivo.

```
unzip terraform_${TER_VER}_linux_amd64.zip
```

Y por último lo almacenaremos en la ruta que deseemos dentro de nuestra máquina física.

```
sudo mv terraform /usr/local/bin/
```

Ya con todos los pasos anteriores completados el cliente podrá trabajar ya directamente con los archivos .tf, que son los de Terraform.

Para crear un proyecto, lo primero que deberá realizar el cliente es crearse un directorio con el siguiente comando: *mkdir -p ~/terraform-deployments && cd ~/terraform-deployments*

Una vez que se encuentre creado el directorio procederemos a crear el propio clúster determinando la versión del clúster la configuración de los nodos y los módulos de Terraform. Se realizará creando una serie de ficheros en un editor de texto (el que utilizo yo es Eclipse) guardándolo con la terminación .tf, para que tenga el formato de Terraform. Todos los objetos que se generen con Terraform deberán llevar dicha terminación, ya sean clústeres, instancias, grupos de seguridad, etc.

Para ello lo primero que el cliente debe realizar es la creación de la VPC y la conexión entre ellas. Para ello se deberá utilizar un script de Terraform para ir conectando y configurando dichas VPCs, como se especifica en las siguientes imágenes.

```
provider "aws" {
  region = "eu-west-1"
}

#Crear una VPC
resource "aws_vpc" "vpc_main" {
  cidr_block = "0.0.0.0/16" #IP de la VPC
  instance_tenancy = "default"
  enable_dns_hostnames = true #Para que nos de la ip publica
  enable_dns_support = true

  tags = {
    Name = "Terraform"
  }
}

#Crear la subnet para nuestra maquina
resource "aws_subnet" "Subred_Terraform" {
  vpc_id = aws_vpc.vpc_main #cogemos el id de nuestra vpc
  cidr_block = "0.0.0.0/24"
  map_public_ip_on_launch = true
  availability_zone = "eu-west-1a"

  tags = {
    Name = "Subnet-Terra"
  }
}

resource "aws_subnet" "Subred_Terraform2" {
  vpc_id = aws_vpc.vpc_main #cogemos el id de nuestra vpc
  cidr_block = "0.0.0.0/24"
  map_public_ip_on_launch = true
  availability_zone = "eu-west-1b"

  tags = {
    Name = "Subnet-Terra-B-2"
  }
}

# Main Internal Gateway for VPC(Nuestro router para acceder a fuera)
# https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/internet_gateway
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.vpc_main_

  tags = {
    Name = "Main IGW"
  }
}
```

8.3 Código yaml, para la creación del clúster

```

#conecta del puerto al amazon, siempre que haya un gateway necesitamos la tabla de rutas
resource "aws_route_table" "public" {
  vpc_id = aws_vpc.vpc_main_id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
  }
  tags = {
    Name = "Public Route Table"
  }
}

#Conectamos la subred id a la tabla de rutas
resource "aws_route_table_association" "public" {
  subnet_id      = aws_subnet.Subred_Terraform.id
  route_table_id = aws_route_table.public.id
}

#Lanzamos la instancia
resource "aws_instance" "instance_ejemplo" {
  ami = "ami-0ec00000000000000"
  instance_type = "t2.micro"
  subnet_id = "subnet-09eb000000000000"
  associate_public_ip_address = true
  tags = {
    Name = "firstInstance"
  }
  key_name = "clave-aws"
}

#Grupo de seguridad
resource "aws_security_group" "Grupo" {
  name = "SecurityGroup"
  description = "Allow all inbound traffic"
  vpc_id = "vpc-0000000000000000"

  ingress {
    from_port = 80
    to_port = 80
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port = 0
    to_port = 0
    protocol = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

#Conecta el grupo de seguridad con la instancia
resource "aws_network_interface_sg_attachment" "sg_attachment" {
  security_group_id = aws_security_group.Grupo_id
  network_interface_id = aws_instance.instance_network_interface_id
}

```

8.4 Continuación del código yaml, para la creación del clúster

En estas imágenes se puede observar los distintos módulos para poder crear, conectar y configurar una red virtual privada (VPC). El primero de los módulos especifica la región en la cual se encontrará la VPC. El siguiente módulo es el de creación de la propia VPC en la que asignamos un nombre, una IP y la dirección IP la habilitamos para poder obtener la propia dirección IP pública.

Una vez que ya tenemos la VPC creada, creamos un par de *subnets* cada una en una región diferente, la primera está en la región europea oeste 1A mientras que la segunda está en la región 1B.

Si proseguimos analizando el script se puede apreciar en el siguiente módulo, la configuración del *Gateway* que sirve para poder acceder desde fuera a las instancias que tenemos, asociando el id de la VPC.

El siguiente bloque es la tabla de rutas. Que nos permite conectar el *Gateway* con la dirección IP pública de la VPC. Una vez que tenemos conectada la VPC deberemos conectar a la tabla de rutas la subred, con el ID de esta. Una vez que ya tenemos realizadas todas las conexiones. Procederemos a crear la instancia, donde especificaremos sus características, así como su clave.

Por último, creamos un grupo de seguridad para restringir el acceso a la instancia, en la imagen al ser un ejemplo está abierta a todas las IP, ya que la dirección 0.0.0.0/0, permite el acceso de todos los públicos.

En la siguiente imagen podemos apreciar los distintos módulos que tiene el *script* de creación del propio clúster EKS. En el primer módulo se puede apreciar que introducimos los datos del clúster, la versión y las distintas *subnets* que tengamos.

Si se sigue analizando el script encontramos el módulo de *tags*. Se tratan de las etiquetas y las referencias que tendremos después de nuestro clúster. Mientras que en el siguiente módulo hacemos referencia a las VPC que tenemos en AWS.

El último grupo se trata de los nodos o instancias que tengan dentro el propio clúster. En este caso tiene 2 instancias. La primera es de tipo t2.small, mientras que la segunda, es una instancia con mayor capacidad y más potente, que es la t2.medium.

Como el cliente se puede imaginar, el costo de la primera instancia y la segunda, a nivel económico no será el mismo. Ya que AWS es una plataforma a la que el cliente paga por servicios y al utilizar mayor capacidad y más recursos consecuentemente produce una elevación del coste.

El último bloque que se puede apreciar en la imagen se trata del id del clúster, que se puede rellenar, aunque si no se rellena Amazon, le otorgará un identificador por defecto.

Estos nodos de las instancias serán los encargados de realizar los procesos computacionales que le exija el usuario. Amazon ya se encargará de realizar la distribución de la carga computacional entre los diferentes nodos del clúster. [22]

```

module "eks" {
  source          = "terraform-aws-modules/eks/aws"
  cluster_name    = local.cluster_name
  cluster_version = "1.20"
  subnets        = module.vpc.private_subnets

  tags = {
    Environment = "development"
    GithubRepo  = "terraform-aws-eks"
    GithubOrg   = "terraform-aws-modules"
  }

  vpc_id = module.vpc.vpc_id

  workers_group_defaults = {
    root_volume_type = "gp2"
  }

  cluster_endpoint_private_access = "true"
  cluster_endpoint_public_access = "true"

  write_kubeconfig      = true
  manage_aws_auth       = true

  worker_groups = [
    {
      name                  = "worker-group-1"
      instance_type         = "t2.small"
      additional_userdata   = "echo foo bar"
      asg_desired_capacity  = 2
      additional_security_group_ids = [aws_security_group.worker_group_mgmt_one.id]
    },
    {
      name                  = "worker-group-2"
      instance_type         = "t2.medium"
      additional_userdata   = "echo foo bar"
      additional_security_group_ids = [aws_security_group.worker_group_mgmt_two.id]
      asg_desired_capacity  = 1
    },
  ]
}

data "aws_eks_cluster" "cluster" {
  name = module.eks.cluster_id
}

```

8.5 Continuación del código yaml, para la creación del clúster

Una vez que ya se encuentre montado el clúster nos dispondremos a acceder a kubernetes mediante el comando *kubectl proxy*. Que nos devuelve la dirección donde se encuentra nuestra interfaz gráfica del clúster. Posteriormente deberemos utilizar un token que lo obtendremos introduciendo en la terminal de nuestro dispositivo el comando *Sudo kubectl -n kube-system describe secret \$(Sudo kubectl -n kube-system get secret | (grep k8sadmin || echo "\$_") | awk '{print \$1}')* | *grep token:* | *awk '{print \$2}'*

Una vez que ya hayamos introducido el token ya nos encontraremos dentro de la interfaz gráfica de nuestro clúster con los servicios desplegados y los *namespaces* que tengamos.

Dentro de kubernetes nos encontraremos con los siguientes objetos:

- **Namespace:** clústeres virtuales que respaldan a un mismo clúster físico. Es una forma de dividir los recursos del clúster entre múltiples usuarios. Los espacios de nombres están pensados para utilizarse en entornos con muchos usuarios distribuidos entre múltiples equipos, o proyecto.

- **Deployment:** es un objeto que puede representar una aplicación en tu clúster. En el *spec*, puedes especificar qué quieres correr tres réplicas de la *app*, pudiendo establecer también un estado específico. ¿Qué es *spec* y *status*? La API de Kubernetes diferencia la especificación (campo de objeto anidado “*spec*”) del estado (campo de objeto anidado “*status*”). El *spec* es una descripción completa del *status* deseado, que incluye los ajustes de configuración proporcionados por el usuario, los valores predeterminados expandidos por el sistema y las propiedades inicializadas o cambiadas después de la creación por otros componentes del ecosistema.

- Un **pod** de Kubernetes es un conjunto de uno o varios contenedores de Linux y constituye la unidad más pequeña de las aplicaciones de Kubernetes. Puede estar compuesto por un solo contenedor, en un caso de uso común, o por varios con conexión directa, en un caso de uso avanzado. Los contenedores dentro de un *Pod* comparten dirección IP y puerto, y pueden encontrarse a través de localhost. También pueden comunicarse entre sí mediante comunicaciones estándar entre procesos, como semáforos de SystemV o la memoria compartida POSIX.

- **Service:** una forma abstracta de mostrar una aplicación que se ejecuta en un conjunto de *Pods* como un servicio de red. Se establece también en el servicio la política de acceso a los Pods (también llamado patrón microservicio).

- **Ingress:** controlador que nos permite acceder a nuestros *Pods* desde fuera del clúster. Este recurso nos permite acceder a servicios a través de HTTP(s) y podemos definir un conjunto de reglas para controlar el tráfico en la comunicación.

- **DaemonSet:** es un controlador que nos permite que, automáticamente, se despliegue un *pod* en cada uno o en algunos de los nodos de un clúster. Es muy útil para lanzar procesos de monitorización o de control del estado de nuestros nodos.

DaemonSet lanza una unidad de cada uno de estos *Pods* que definimos aquí abajo en cada uno de los nodos del clúster. Este comportamiento lo podemos modificar utilizando '*tolerations*'.

Con el *daemonset* podemos consultar la métrica del microservicio en cada nodo

- **Configmap:** es un objeto de la API utilizado para almacenar datos no confidenciales en el formato clave-valor. Los *Pods* pueden utilizar los ConfigMaps como variables de entorno, argumentos de la línea de comandos o como ficheros de configuración en un Volumen.

Es para hacer persistencia a un archivo, con la clave-valor hacia la información propia del archivo.

- **ReplicaSet:** es un recurso de Kubernetes que asegura que siempre se ejecute un número de réplicas de un *pod* determinado. Por lo tanto, nos asegura que un conjunto de *Pods* siempre están funcionando y disponibles. Que no haya caída del servicio.

- **PersistenceVolume (PV):** es un recurso de almacenamiento en el clúster que ha sido provisto por un administrador o provisto dinámicamente mediante clases de almacenamiento. Un administrador de clúster crea varios PV. Llevan los detalles del almacenamiento real, que está disponible para que lo utilicen los usuarios del clúster.

PV es el espacio físico del clúster del cliente que le asigno a un microservicio donde se guardan los datos. Es un volumen persistente(físico).

- **PersistenceVolume (PVC):** es una solicitud de almacenamiento por parte de un usuario desde un PV. Las reclamaciones pueden solicitar tamaños y modos de acceso específicos (por ejemplo: se pueden montar una o muchas veces de lectura/escritura: ReadWriteOnce, ReadOnlyMany o ReadWriteMany).

Si ninguno de los volúmenes persistentes estáticos coincide con la solicitud de PVC del usuario, el clúster puede intentar crear dinámicamente un PV que coincida con la solicitud de PVC según la clase de almacenamiento.

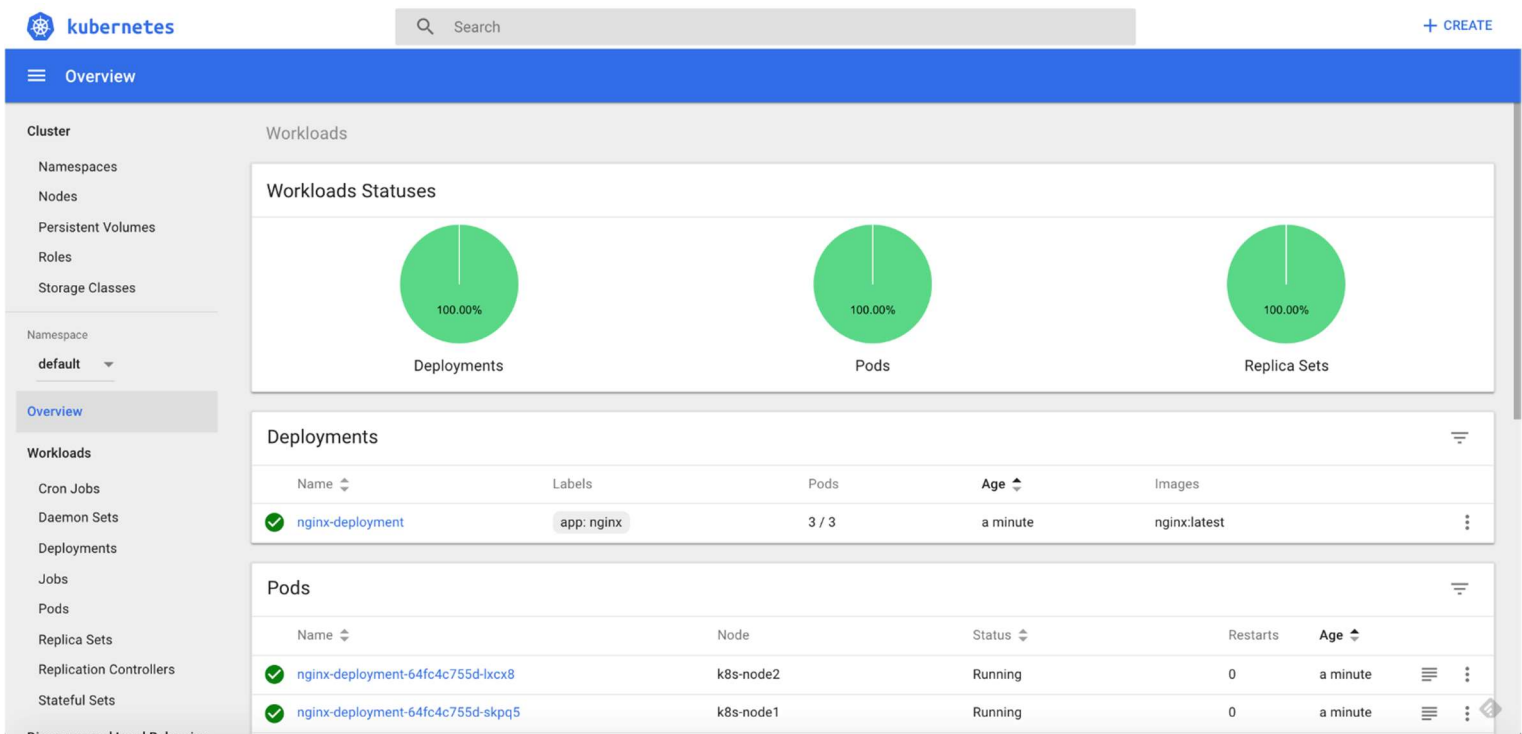
- **StorageClass:** proporciona a los administradores una forma de describir las "clases" de almacenamiento que ofrecen. Las diferentes clases pueden correlacionarse con niveles de calidad de servicio, o con políticas de respaldo, o con políticas arbitrarias determinadas por los administradores del clúster. El propio Kubernetes no tiene opiniones sobre lo que representan las clases. Este concepto a veces se denomina "perfiles" en otros sistemas de almacenamiento.

- **Statefullset:** es el objeto de la API workload que se usa para gestionar aplicaciones con estado. [13]

Gestiona el despliegue y escalado de un conjunto de *Pods*, y garantiza el orden y unicidad de dichos *Pods*. [13]

Al igual que un *deployment*, un *StatefulSet* gestiona *Pods* que se basan en una especificación idéntica de contenedor. A diferencia de un *deployment*, un *StatefulSet* mantiene una identidad asociada a sus *Pods*. Estos *Pods* se crean a partir de la misma especificación, pero no pueden intercambiarse; cada uno tiene su propio identificador persistente que mantiene a lo largo de cualquier reprogramación. [23]

Un *StatefulSet* opera bajo el mismo patrón que cualquier otro controlador. Se define el estado deseado en un objeto *StatefulSet*, y el controlador del *StatefulSet* efectúa las actualizaciones que sean necesarias para alcanzarlo a partir del estado actual. [23]



The screenshot shows the Kubernetes dashboard interface. The top navigation bar includes the Kubernetes logo, a search bar, and a '+ CREATE' button. The left sidebar contains a menu with 'Overview' selected. The main content area is titled 'Workloads' and features three circular progress indicators for 'Deployments', 'Pods', and 'Replica Sets', all showing '100.00%'. Below these, there are two tables: 'Deployments' and 'Pods'. The 'Deployments' table has columns for Name, Labels, Pods, Age, and Images. The 'Pods' table has columns for Name, Node, Status, Restarts, and Age.

Name	Labels	Pods	Age	Images
nginx-deployment	app: nginx	3 / 3	a minute	nginx:latest

Name	Node	Status	Restarts	Age
nginx-deployment-64fc4c755d-lxcx8	k8s-node2	Running	0	a minute
nginx-deployment-64fc4c755d-skpg5	k8s-node1	Running	0	a minute

8.6 Kubernetes dentro del clúster con el despliegue de un servicio

9 CONCLUSIONES

Tras el exhaustivo que he realizado, se han podido sacar varias conclusiones, por las cuales, dependiendo del poder adquisitivo y tamaño puede compensar o no hacer una migración de los servicios *on premise*, una plataforma *cloud* como pueden ser AWS o Azure.

La primera conclusión que sacamos es que, si la empresa tiene unos ingresos asegurados y fijos o tiene un gran poder adquisitivo, no le vale la pena tener los servidores y la información en local. Ya que con estas plataformas *cloud* el cliente tendrá unos gastos fijos, podrá beneficiarse de beneficios fiscales como, el ahorro del IVA o desgravaciones de impuestos sobre los servicios *cloud* que utilizan. Además, el cliente se encuentra con unos gastos mensuales predecibles, en función de la cantidad de recursos que haya consumido. Otra de las principales ventajas es el ahorro de espacios y el desembolso que tiene que realizar el cliente para obtener la cantidad de servicios que te ofrece AWS o Azure. Ya que ese espacio lo puedes tener disponible para realizar otras funciones, por lo tanto, le estarías sacando una rentabilidad extra y el desembolso inicial, para conseguir todos los servicios, se puede invertir en otros servicios que le convenga a la empresa.

Respecto a una empresa pequeña que necesita un número bastante limitado de ordenadores y de recursos, en mi opinión no le sale rentable el estar desembolsando una cantidad fija al mes, ya que, a los pocos meses, la cantidad invertida en dichas plataformas superaría con creces al desembolso inicial.

Respecto a la comparativa que se realiza en el trabajo entre ambas plataformas, obtengo como conclusión que son dos plataformas muy similares que se reparten más de la mitad del mercado, con servicios muy similares y métodos de pago y precios también a la par. La única diferencia por la que decantaría por AWS es que la red que tiene global es algo mayor que la de Azure y gracias a eso tiene una mayor cobertura y un mayor rendimiento. Aunque por el resto de las funcionalidades y en la forma en la que se utilizan las plataformas son similares.

En la temática de migración servicios *on premise* a *cloud* con AWS, mi recomendación por sencillez es que el montaje del clúster, así como todas las subredes y VPCs se creen bajo la interfaz que te proporciona Amazon, aunque en el entorno de las empresas se suele utilizar más Terraform, ya que puedes entrar más en detalle en cada una de las funciones de AWS.

10 LÍNEAS FUTURAS

El proyecto se pasará a la práctica en el trabajo del segundo cuatrimestre, en el cual, se levantará un clúster EKS con los nodos correspondientes, pasando diferentes servicios de una máquina virtual a un clúster EKS que nos proporcionará AWS. Para trabajar con los dockerfiles para automatizar algunas funciones o patrones y explicar la base del funcionamiento de kubernetes utilizaremos una distribución Ubuntu en una máquina virtual. Esta será una máquina con 4 gigas de memoria base interna. Mientras que de espacio de almacenamiento le he asignado a la máquina 10 gigas. Con dichas prestaciones, obtendremos rendimiento suficiente para crear los dockerfile.

Una vez que se hayan explicado con ejemplos como funcionan los dockerfiles, se utilizará el propio sistema operativo Windows 10, que viene ya instalado en el portátil y se llevará a cabo en el propio equipo la instalación de paquetes para poder manejar los comandos en el Powershell tanto de kubectl como los de AWS CLI.

Una vez que hayamos preparado nuestro equipo con las instalaciones oportunas y se hayan explicado los conceptos bases. Comenzaremos a configurar el clúster con un nodo y la estructura más básica, la t2.micro, que es las instancias que Amazon proporciona sin coste adicional. Esta instancia tiene solo una CPU, que puede trabajar a 6 créditos la hora. Tiene también un giga de RAM. Con un rendimiento de red entre bajo y moderado.

También, aunque en este caso es se trata de una instancia gratuita, realizar una plantilla de autoescalado para que se levanten y se apaguen las instancias a las horas que configuremos, para optimizar en costes.

En conclusión, este proyecto se entregará en enero será solamente la parte teórica de la migración de servicios a AWS. La continuación se llevará a cabo en el segundo cuatrimestre llevando a la práctica y realizando la migración de los servicios. Para la demostración se realizará un video demostrando el resultado de la migración que se acompañará con una documentación que justifique que se ha ido realizando en cada caso. Con estos dos trabajos serán la conclusión de las asignaturas de prácticas en la empresa.

11 PRESUPUESTO

Tras haber realizado una estimación de costes en la página web de Amazon, hemos configurado un entorno similar al que utilizaré en el siguiente trabajo de prácticas. Con los servicios de dos VPCs situadas en la región de Irlanda dentro de Europa. Estas con un terabyte de capacidad las instancias y con transferencia de datos entre las VPC se queda por un precio en torno a 20 \$ al mes.

El servicio del gateway para controlar quien accede a las distintas VPCs que tenemos Amazon nos lo incluye en el precio anterior, por lo tanto, nos los ofrece de forma gratuita, si le añadimos las opciones de API Rest, para que ciertas páginas no puedan captar información nos incluyen un pequeño sobrecosto.

Las instancias EC2 que se que se encontrarán dentro del clúster y de las VPC con una capacidad de 8GB de memoria RAM y 2 núcleos de procesamiento, con una estimación de uso de 8 horas diarias, una empresa con 35 empleados saldría entorno a los 35\$ mensuales.

Y por último para completar los servicios que se necesitan y se trata del más importantes es el propio clúster EKS que sale a 0,10 \$ la hora que se traduce en un gasto mensual de 73\$.

Si realizamos la suma de todos los servicios se nos quedan en torno a unos 123\$ dólares al mes, que prorrateado a un año se traduce en 1541\$.

My Estimate Editar		Añadir servicio	Adición de compatibilidad	Agregar grupo	Borrar estimación	Exportar estimación	Compartir
Amazon EC2							Editar Acción
Región: UE (Irlanda)							
Estimación rápida							
Sistema operativo (Linux), Cantidad (2), Estrategia de precios (EC2 Instance Savings Plans 1 año Sin pagos iniciales), Cantidad de almacenamiento (100 GB), Tipo de instancia (t2.micro)		Mensual:					34,99 USD
Amazon Virtual Private Cloud (VPC)							Editar Acción
Región: UE (Irlanda)							
Conexión de VPN							
Días laborables al mes (22), DT Entrada: Internet (0 TB al mes), DT Salida: Internet (0 TB al mes), DT Intra-región: (1 TB al mes), Costo por transferencia de datos (20.48)		Mensual:					20,48 USD
Amazon API Gateway							Editar Acción
Región: UE (Irlanda)							
Unidades de solicitudes de la API HTTP (millones), Tamaño promedio de cada solicitud (34 KB), Unidades de solicitud de la API REST (millones), Tamaño de memoria caché (GB) (Ninguno), Unidades de mensaje WebSocket (miles), Tamaño promedio del mensaje (32 KB)							
Amazon EKS							Editar Acción
Región: UE (Irlanda)							
Amazon EKS							
Número de clústeres de EKS (1)		Mensual:					73,00 USD

12 APLICABILIDAD

Las plataformas *cloud* se han creado para que las empresas no tengan que invertir ellas mismas en tener las infraestructuras con los servidores en los propios centros de trabajo. Para ello las grandes compañías como Amazon o Microsoft han desplegado una gran red con servidores físicos distribuidos en diferentes partes del mundo. Estas compañías que tienen tantos recursos ponen a disposición de otras empresas esos recursos y los servicios que ellas mismas han desplegado. Por estos recursos al mes los clientes deberán pagar una cantidad fija de dinero, que podrá ir modificándose en función cuantos recursos utilicen, en caso de necesitar en un momento dado más recursos o servicios ese mes se le cobrará más y la misma situación en caso inverso, si no utilizas tantos recursos como los que habías contratado en un principio, se te pasará una factura de menor cantidad.

Estos servicios las empresas los contratan en la mayoría de los casos, porque les resulta insostenible el poder tener y mantener una cantidad de servicios tan amplia como tienen AWS o Azure. Además, tienen los beneficios de que el espacio que debería estar ocupado para tener esos recursos, lo pueden utilizar para otra finalidad. Esto tiene como finalidad una mayor eficacia y rendimiento que en caso de tener todos los *servicios on premise*. Además, el cliente no deberá estar al tanto de todas las reparaciones en los servidores ni el mantenimiento de estos, por lo que siempre tendrá los servicios que ofrece las plataformas *cloud* correspondiente disponibles, independientemente de que estén en reparación o mantenimiento.

Para manejar este tipo de plataformas no se necesita una formación específica, debido a su sencillez y las interfaces gráficas que aportan las podrá manejar cualquier usuario sin ningún problema, aun así, se proporcionan manuales en los cuales se explica en detalle cada una de las funciones y funcionalidades de cada herramienta proporcionada.

13 ANEXO

13.1 GLOSARIO

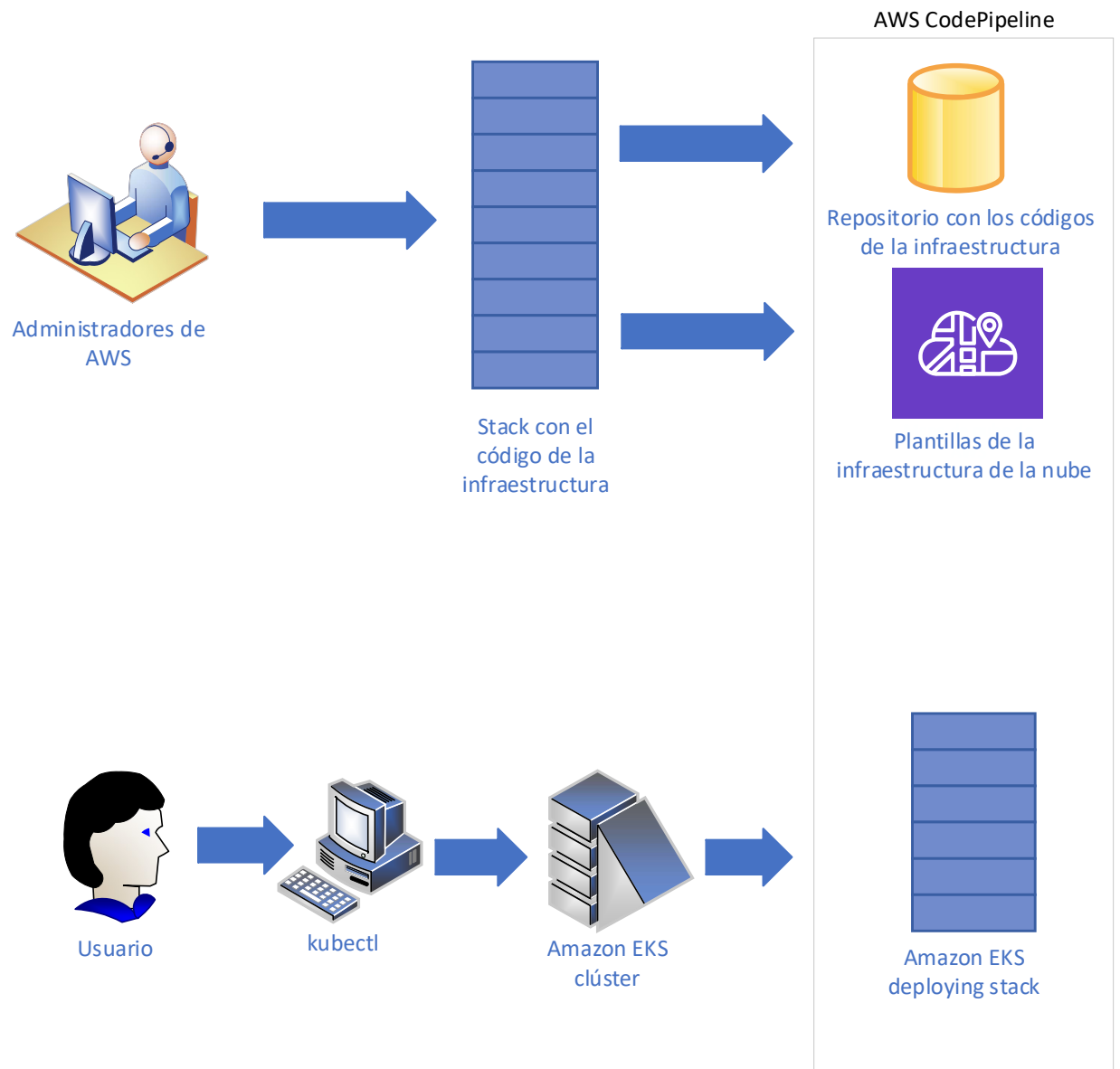
1	Onboarding:	Incorporación (en el caso del trabajo en la empresa).
2	Ongoing:	Continuación (en el caso del trabajo en la empresa).
3	Interns:	Becarios.
4	Devops:	Desarrollo de operaciones.
5	Dockerfiles:	Archivos de texto plano que contienen una serie de instrucciones necesarias para crear una imagen que, posteriormente, se convertirá en una sola aplicación utilizada para un determinado propósito.
6	IP:	Protocolo de Internet. Puede ser de versión 4 o 6.
7	Instancias:	Servidor virtual en la nube de AWS.
8	Subnets:	Redes internas en las que se encuentran las instancias. Es decir, rango de direcciones IP de las VPCs.
9	VPC:	Red virtual dedicada para su cuenta de AWS.
10	Clúster:	Grupo de máquinas interrelacionadas que trabajan en un mismo sector y que colaboran estratégicamente para obtener beneficios comunes.

11	<i>Namespace:</i>	Clústeres virtuales que respaldan a un mismo clúster físico. Es una forma de dividir los recursos entre múltiples usuarios. Los espacios de nombres están pensados para utilizarse en entornos con muchos usuarios distribuidos entre múltiples equipos o proyectos.
12	<i>Ingress:</i>	Recurso que permite a los usuarios acceder a los servicios desplegados a través de HTTPs. Pudiendo controlar el tráfico con las reglas impuestas por el usuario.
13	<i>Load Balancer:</i>	Balanceador de carga entre las VPC del clúster.
14	<i>Pipeline:</i>	Conjunto de instrucciones del proceso que siga una aplicación desde el repositorio de control de versiones hasta que llega a los usuarios.
15	<i>Cloud:</i>	Servicios de computación a través de una red, que usualmente es internet.
16	<i>Computing:</i>	Relativo a la informática.
17	<i>Workmail:</i>	Correo electrónico relativo al oficio que desempeña el usuario.
18	<i>Bussiness Intelligent:</i>	Habilidad para transformar los datos en información, y la información en conocimiento, de forma que se pueda optimizar el proceso de toma de decisiones en los negocios.
19	<i>Internet of Things:</i>	Se refiere a una interconexión digital de objetos cotidianos con internet.
20	<i>PYMES:</i>	Pequeña y mediana empresa.

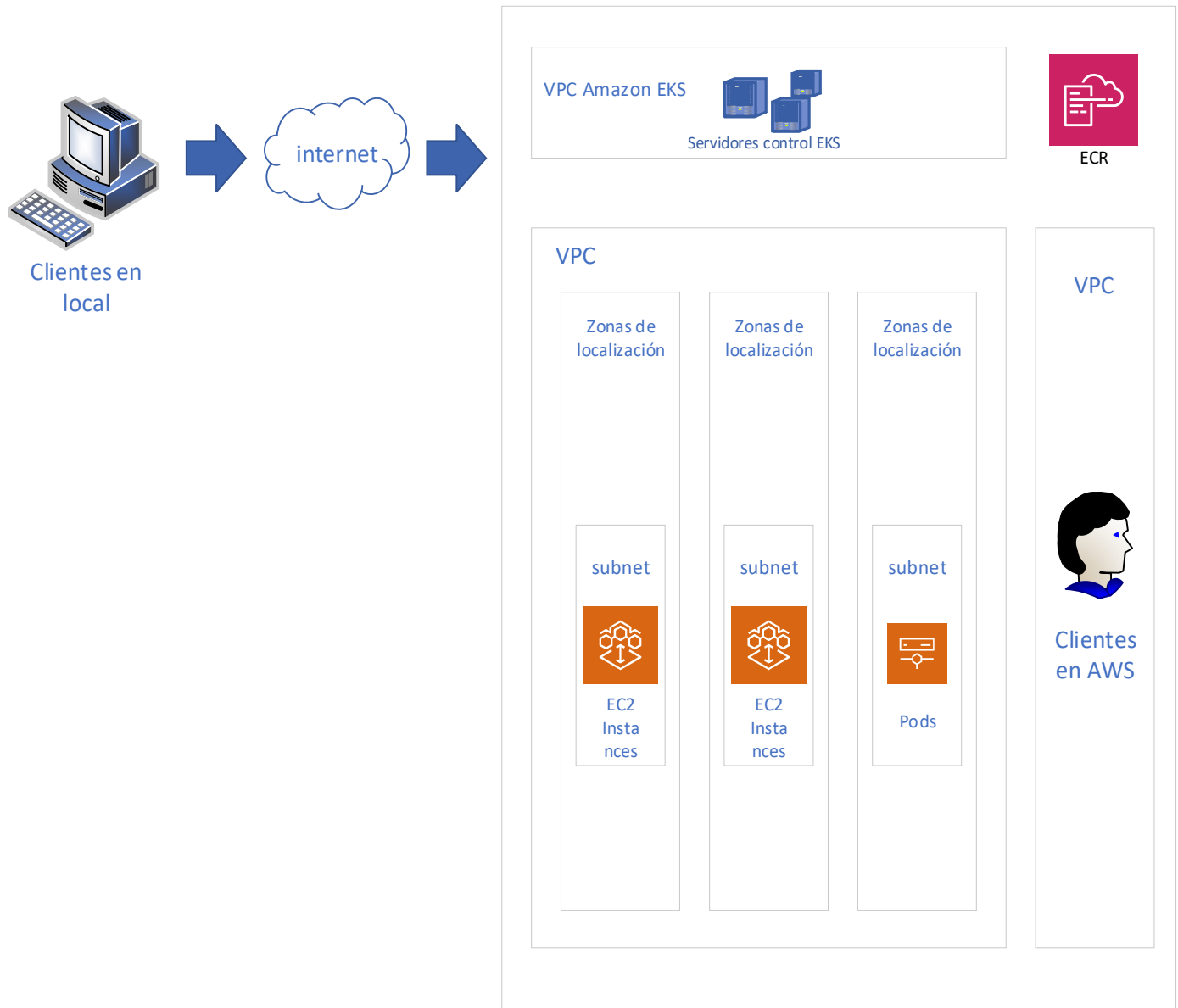
21	On-premise:	Local. Albergado en la máquina física del usuario.
22	Backups:	Copias de seguridad.
23	Marketplaces:	Mercados dentro de las plataformas, en los cuales se pueden obtener servicios. Algunos gratuitos y otros son de pago.
24	DNS:	Método de denominación empleado para nombrar a los dispositivos que se conectan a una red a través del IP.
25	DNSSEC:	Extensiones de seguridad para el sistema de nombres de dominio son un conjunto de especificaciones de la Internet Engineering Task Force para asegurar cierto tipo de información proporcionada por el sistema de nombres de dominio que se usa en el protocolo de Internet.
26	Network:	Conjunto de equipos nodos y software conectados entre sí por medio de dispositivos físicos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos, con la finalidad de compartir información, recursos y ofrecer servicios.
27	ISP:	Conecta a sus usuarios a Internet a través de diferentes tecnologías.
28	CIDR:	Enrutamiento entre dominios sin clases.
29	Developers:	Desarrolladores
30	Script:	Documento que contiene instrucciones, escritas en códigos de programación.
31	EKS	Servicio de kubernetes adaptable de AWS

13.2 DIAGRAMAS

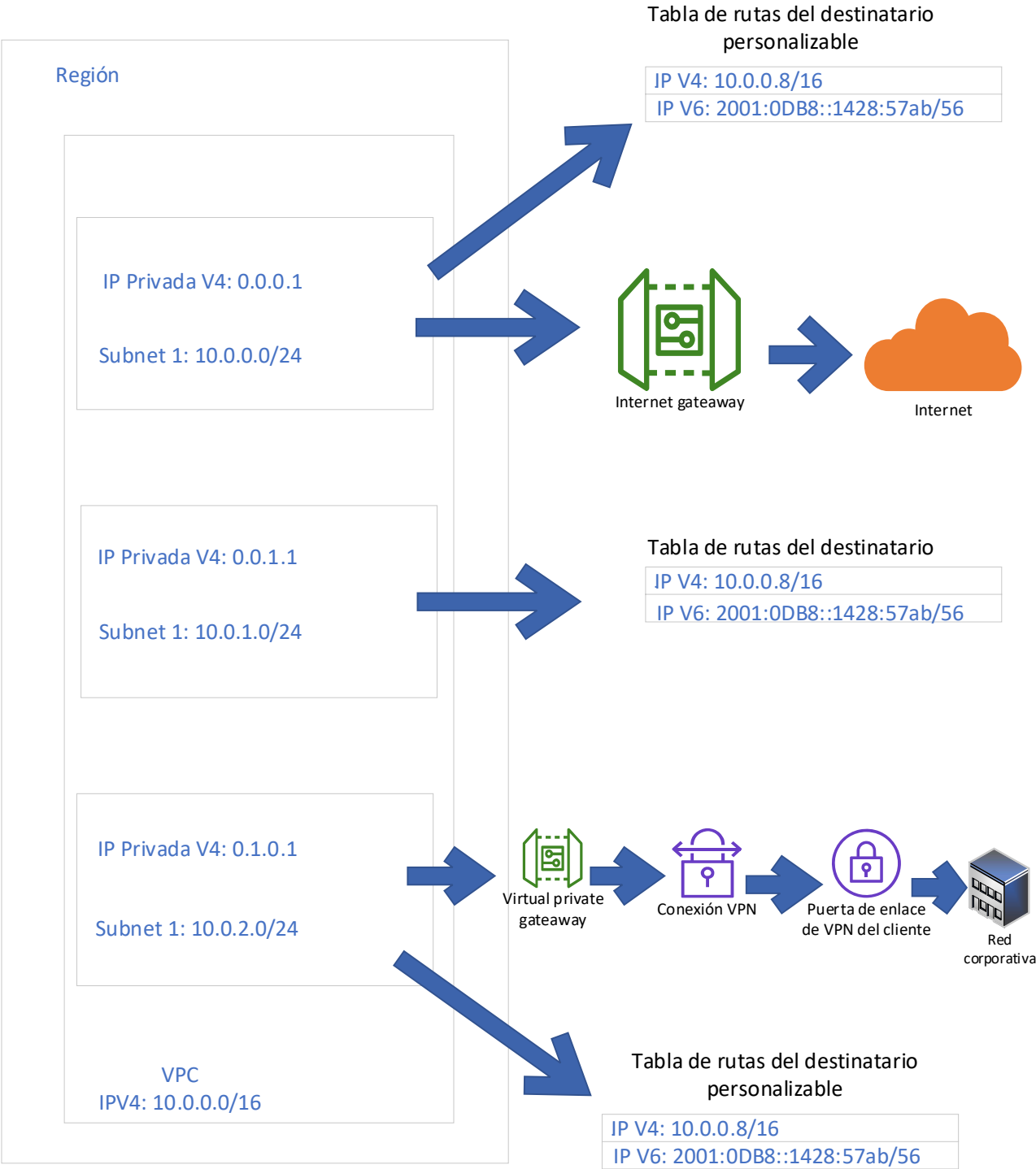
13.2.1 Infraestructura general de AWS



13.2.2 Infraestructura clúster EKS



13.2.3 Infraestructura VPC



14 AGRADECIMIENTOS

En primer lugar, quería agradecer a mis padres el hecho de poder estar realizando las prácticas, ya que, han sido quienes me han apoyado e incentivado siempre a continuar estudiando y a decantarme por la carrera que actualmente estoy cursando. Además, no solo por eso sino por la cantidad económica que están invirtiendo en mi día a día para que la formación y las prácticas que estoy realizando en la empresa sean posibles.

En segundo lugar, me gustaría agradecer a la universidad Nebrija los servicios que me han dado durante estos años. Que sin esos servicios no hubiese sido posible adquirir los conocimientos que he ido y sigo obteniendo día a día. Otro de los servicios por el que estoy agradecido es por la oportunidad que se nos dan desde la propia universidad para realizar prácticas. Aunque especialmente dentro de la universidad, me quiero centrar en el claustro de profesores, ya que, son con quienes compartes el tiempo día a día dentro de las aulas y son quienes te imparten las lecciones y de quienes aprendes. Agradecerles el trato que nos brindan en el día a día, así como, la disposición de la mayoría de ellos a la hora de resolver algún problema o duda que te ha surgido.

También quiero agradecerles a mis amigos más cercanos que son quienes en un momento de bajón (porque te ha salido mal algún examen o por el estrés causado por la cantidad de trabajo) están ahí para apoyarte.

Por último, no me podía olvidar de los compañeros de la carrera, aunque a algunos de ellos, ya están incluidos en el párrafo anterior, ya que, los considero amigos. Gracias por ayudarme y hacer que el día a día fuese muy agradable y ameno. Gracias a ellos (y a los profesores) en el futuro, cuando termine la universidad, podré tener un gran recuerdo de ella.

15 BIBLIOGRAFÍA

Definición AWS

[1] What Is Amazon Web Services and Why Is It So Successful?

Vanessa Page (agosto 2021)

<https://www.investopedia.com/articles/investing/011316/what-amazon-web-services-and-why-it-so-successful.asp>

[2] Amazon Web Services (AWS)

Alexander S. Gillis (abril 2020)

<https://searchaws.techtarget.com/definition/Amazon-Web-Services>

[3] Carrier selecciona a AWS de Amazon como proveedor en la nube para impulsar la transformación digital

DeNoticias (febrero 2020)

<https://www.denoticias.es/tecnologia/carrier-aws-amazon.html>

[4] AWS lidera el mercado de servicios de infraestructura *cloud*

Muycanal E.E.G (febrero de 2019)

<https://www.muycanal.com/2019/02/28/aws-mercado-cloud>

[5] Amazon Web Services

TIC Portal (2021)

<https://www.ticportal.es/temas/cloud-computing/amazon-web-services>

Definición Azure

[6] Microsoft Azure

Stephen J. Bigelow (agosto 2021)

<https://www.computerweekly.com/es/definicion/Microsoft-Azure>

[7] Cumplimiento de Azure

Microsoft Ibérica (2019)

<https://azure.microsoft.com/es-es/overview/trusted-cloud/compliance/>

[8] Los Servicios AZURE

Prodware España (Madrid) (2020)

<https://www.prodwaregroup.com/es-es/soluciones/microsoft-azure/>

[9] ¿Quiénes son los principales clientes de la plataforma Microsoft Azure?

Sudhanshu D. (2019)

<https://www.cisin.com/coffee-break/es/enterprise/who-are-biggest-customers-of-the-microsoft-azure-platform.html>

Diferencias y similitudes entre AWS y Microsoft Azure

[10] AWS vs Azure en 2021 (Comparación de los gigantes de la computación en la nube)

2021 Kinsta Inc, Edward Jones, marzo 25, 2021

<https://kinsta.com/es/blog/aws-vs-azure/>

Proceso de migración de servicios on Premise a AWS según Amazon

[11] ¿Cómo migrar?

Amazon Web Services, Inc. o sus empresas afiliadas (2021)

<https://aws.amazon.com/es/cloud-migration/how-to-migrate/>

[12] How to operate

Amazon Web Services, Inc. o sus empresas afiliadas (2021)

<https://aws.amazon.com/es/cloud-migration/how-to-operate/>

[13] Gestión de operaciones centralizada

Amazon Web Services, Inc. o sus empresas afiliadas (2021)

<https://aws.amazon.com/es/products/management-and-governance/use-cases/centralized-operations-management/>

[14] Observabilidad

Amazon Web Services, Inc. o sus empresas afiliadas (2021)

<https://aws.amazon.com/es/products/management-and-governance/use-cases/monitoring-and-observability/>

[15] Configuración, cumplimiento y auditoría

Amazon Web Services, Inc. o sus empresas afiliadas (2021)

<https://aws.amazon.com/es/products/management-and-governance/use-cases/configuration-compliance-and-auditing/>

[16] Gobernanza y control empresarial

Amazon Web Services, Inc. o sus empresas afiliadas (2021)

<https://aws.amazon.com/es/products/management-and-governance/use-cases/enterprise-governance-and-control/>

[17] Aprovisionamiento y organización

Amazon Web Services, Inc. o sus empresas afiliadas (2021)

<https://aws.amazon.com/es/products/management-and-governance/use-cases/provisioning-and-orchestration/>

[18] Administración financiera en la nube con AWS

Amazon Web Services, Inc. o sus empresas afiliadas (2021)

<https://aws.amazon.com/es/aws-cost-management/>

Dificultades en la migración de local a cloud

[19] What is cloud migration? | Cloud migration strategy

Cloudflare, Inc. (2021)

<https://www.cloudflare.com/es-es/learning/cloud/what-is-cloud-migration/>

Creación de un clúster EKS mediante la interfaz gráfica

[20] Amazon Virtual Private Cloud (VPC)

Amazon Web Services, Inc o sus afiliados (2021)

<https://aws.amazon.com/es/vpc/>

[21] Creación de un clúster de Amazon EKS

Amazon Web Services, Inc o sus afiliados (2021)

https://docs.aws.amazon.com/es_es/eks/latest/userguide/create-cluster.html

[22] How To Deploy EKS Cluster on AWS using Terraform

Victor Shamallah (agosto 2021)

<https://computingforgeeks.com/how-to-deploy-eks-cluster-on-aws-using-terraform/>

[23] StatefulSets

Los autores de Kubernetes (2021)

<https://kubernetes.io/es/docs/concepts/workloads/controllers/statefulset/>