Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Telecomunicación

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Trabajo Fin de Grado

Desarrollo de un sistema de automatización del despliegue y gestión de nubes híbridas

Luis Travé Reneses

2023

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Trabajo Fin de Grado

Título: Desarrollo de un sistema de automatización del despliegue y gestión de nubes híbridas

Autor: D. Luis Travé Reneses

Tutor: D. Sergio-Nabil Khayyat Arranz

Ponente: D. Alejandro Antonio Alonso Muñoz

Departamento: Departamento de Ingeniería de Sistemas Telemáticos

Miembros del Tribunal

Presidente: D. . . . . .

Vocal: D. . . . . ..

Secretario: D. . . . . ..

Suplente: D. . . . . ..

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de: . . . . . .

Madrid, a de de 20...

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Telecomunicación

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Trabajo Fin de Grado

Desarrollo de un sistema de automatización del despliegue y gestión de nubes híbridas

Luis Travé Reneses

2023

Resumen

Durante la última década la adopción de la computación en la nube por parte de usuarios y empresas ha sufrido un crecimiento masivo, convirtiéndose en uno de los pilares de la tecnología actual. En concreto para las empresas, la computación en la nube ha pasado de ser una simple alternativa a los centros de datos tradicionales, a convertirse en un componente esencial de su lógica de negocios.

Los proveedores de servicios en la nube ofrecen acceso a nubes públicas y privadas según las necesidades de los clientes. Sin embargo, existe una opción que combina los beneficios de ambas: las nubes híbridas. Estas nubes híbridas brindan entornos con la escalabilidad y flexibilidad característica de la nube pública, al mismo tiempo que proporcionan la seguridad y control que ofrecen las nubes privadas. Sin embargo, es importante tener en cuenta que las nubes híbridas tienden a ser entornos de alta complejidad, lo que puede requerir una gestión y configuración cuidadosa para aprovechar al máximo sus beneficios.

Con el fin de afrontar dicha complejidad, en el presente proyecto se propone un sistema que automatice el despliegue y gestión de entornos de nube híbrida interconectando el entorno local empresarial y entornos de nube pública como si fuesen uno solo. Este sistema pretende agilizar y simplificar el alojamiento de cargas de trabajo en la nube, pudiendo desplegar de forma completamente automática recursos de TI en la nube que estén plenamente interconectados de forma segura con los recursos de las instalaciones locales.

Como proveedor de servicios en la nube se utilizará Amazon Web Services (AWS), siendo la principal herramienta utilizada a la hora de conformación y aprovisionamiento de las nubes AWS CloudFormation.

El presente trabajo se realiza en colaboración con ADIC-IIC, un centro español de investigación, desarrollo e innovación centrado en proyectos de análisis Big Data e Inteligencia Artificial.

Summary

Throughout the last decade, adoption of cloud computing by users and companies has increased massively, turning cloud computing into one of current technologies cornerstones. Specifically for companies, cloud computing has gone from being a simple alternative for traditional data centers to an essential component of their business logic.

Cloud service providers offer access to both public and private clouds based on their clients needs. However, there is an option that combines the benefits of both: hybrid clouds. These hybrid clouds provide environments with the scalability and flexibility characteristic of public clouds, while also offering the security and control provided by private clouds. However, it is important to take into consideration that hybrid clouds tend to be complex environments, which may require careful management and configuration to fully leverage

their benefits.

To address this complexity, the present project proposes a system that automates the deployment and management of hybrid cloud environments, seamlessly connecting the on-premises environment with public cloud environments. This system aims to facilitate and simplify the hosting of workloads in the cloud, enabling fully automated deployment of IT resources in the cloud that are securely interconnected with on-premises resources.

The cloud service provider whose services will be used will be Amazon Web Services (AWS), being the main tool used in the modeling and provisioning of the clouds AWS CloudFormation.

The development of this system will be made in collaboration with ADIC-IIC, a spanish research center focused on Big Data analysis and Artificial Intelligence projects.

Palabras Clave

Computación en la nube, nubes híbridas, Amazon Web Services, AWS CloudFormation

Keywords

Cloud computing, hybrid clouds, Amazon Web Services, AWS CloudFormation

Agradecimientos

Este trabajo no habría sido posible sin la ayuda de muchísimas personas y quiero agradecérselo a todas las que no están mencionadas a continuación.

Antes de nada, me gustaría expresar mi más sincero agradecimiento al Instituto de Ingeniería del Conocimiento, y en especial al área de DEA, por darme la posibilidad de trabajar y desarrollar este proyecto con ellos. Su apoyo, experiencia y orientación han sido vitales para hacer de este trabajo algo de lo que estoy muy orgulloso.

También quiero dar las gracias a Alejandro Alonso, por su gran contribución en la supervisión y dirección de mi trabajo. Su experiencia y conocimientos han sido clave para mejorar significativamente la calidad de mi trabajo.

No se ni cómo expresar lo agradecido que estoy a mi familia, y en especial a mis padres. Su constante apoyo, cariño y paciencia han sido mi mayor motivación y fuerza en este camino, y sin ellos no habría sido capaz de llegar a donde he llegado.

A Cris, la persona que más me apoya y que más tiene que soportarme, no tengo forma de agradecer lo mucho que haces por mí. Te quiero, y no se que haría sin ti.

Gracias a mis amigos de *Ruleco* y *Planchazo*, lo mejor que me llevo de estos años en la UPM. Aunque a veces parezca que habéis sido más bien un obstáculo en mi vida académica, de alguna forma habéis conseguido que disfrute esta barbaridad de carrera. Gracias también a mis amigos del *Montessori*, con vosotros nada es imposible.

Índice

Índice de figuras

Índice de tablas

Lista de acrónimos

ACL *Access Control List*

API *Application Programming Interfaces*

AMI *Amazon Machine Image*

AS *Autonomous System*

ASN *Autonomous System Number*

AWS *Amazon Web Services*

AZ *Availability Zone*

BGP *Border Gateway Protocol*

CGW *Customer Gateway*

CIDR *Classless Inter-Domain Routing*

CLI *Command Line Interface*

CSP *Cloud Service Provider*

DNS *Domain Name System*

EBS *Elastic Block Storage*

EC2 *Elastic Compute Cloud*

EFS *Elastic File System*

EIP *Elastic IP Address*

ENI *Elastic Network Interface*

IaaS *Infrastructure as a Service*

IaC *Infrastructure as a Code*

IIC *Instituto de Ingeniería del Conocimiento*

IKE *Internet Key Exchange*

IP *Internet Protocol*

IPsec *Internet Protocol security*

LAN *Local Area Network*

NAT *Network Address Translation*

PaaS *Platform as a Service*

PSK *Pre-Shared Key*

SaaS *Software as a Service*

SO *Sistema Operativo*

TGW *Transit Gateway*

TI *Tecnologías de la Información*

VGW *Virtual Private Gateway*

VPC *Virtual Private Cloud*

VPN *Virtual Private Network*

1. Introducción y objetivos

1.1. Introducción

La computación en la nube se ha convertido en una de las tendencias tecnológicas más influyentes del siglo XXI por su total cambio de paradigma con respecto al enfoque tradicional de las tecnologías de la información. Según Gartner, la computación en la nube es ¨un estilo de computación en el que servicios de TI elásticos y escalables se ofrecen como un servicio a través de Internet".

Es decir, la computación en la nube ofrece recursos de TI tales como computación, almacenamiento, redes y aplicaciones a los que los usuarios pueden acceder con el único requisito de tener conexión a Internet. Por lo tanto, el cambio de paradigma viene del hecho de que para que un usuario o una empresa tenga acceso a, por ejemplo, capacidad de computación, ya no tiene que adquirir, implementar y gestionar un servidor propio. Con la computación en la nube, tiene acceso a recursos TI con escalabilidad ilimitada, con un modelo bajo demanda en el que sólo pagará por lo que utilice, y en el que la administración y gestión de la infraestructura la lleva a cabo el proveedor de servicios en la nube.

Dentro del paradigma de la computación en la nube, se pueden distinguir dos tipos de nubes principales: las nubes privadas y las nubes públicas.

Una nube privada es una infraestructura de TI dedicada y reservada exclusivamente a un usuario u organización, que puede estar alojada en un centro de datos perteneciente al usuario o a través de un proveedor de servicios en la nube. Por lo tanto, ofrecen más control y seguridad sobre la infraestructura, además de rendimiento y costes más predecibles, a cambio de costes iniciales mucho más altos y menos flexibilidad y escalabilidad que con la nube pública.

Las nubes públicas ofrecen recursos de TI a través de Internet por proveedores de servicios en la nube disponibles para cualquier usuario u organización con acceso a la nube. Por lo tanto, la nube pública provee una escalabilidad ilimitada a ojos del consumidor, con costes mucho más eficientes y una accesibilidad y gestión muy simples, a cambio de menos seguridad y privacidad, recursos menos personalizados a las necesidades específicas de cada cliente, una mayor latencia y el hecho de que se depende del proveedor de servicios en la nube.

Por estas diferencias, cada tipo de nube atrae a tipos de empresas diferentes: las nubes privadas son utilizadas por empresas que necesitan tener un control total sobre sus datos y que tienen altos requisitos de privacidad y seguridad como empresas gubernamentales, del sector financiero y del sector de la salud, mientras que las nubes públicas atraen a

empresas que necesitan acceso a infraestructura flexible y escalable con bajas inversiones, como son *startups*, empresas de comercio electrónico o empresas del sector tecnológico.

Sin embargo, existe un tercer tipo de nube cuya adopción está ya muy establecida a día de hoy y que permite aprovechar las ventajas de los dos tipos de nube previos: las nubes híbridas.

Una nube híbrida es una combinación de nube privada y pública, cuyo principal beneficio es que permite combinar en una misma nube la escalabilidad y flexibilidad de las nubes públicas con la seguridad y control de las nubes privadas. Así, es posible dividir las cargas de trabajo dentro de una misma nube: las aplicaciones y datos críticos que requieran de un alto nivel de seguridad se pueden alojar en nubes privadas a la vez que otras aplicaciones menos críticas o con mayores picos de tráfico se alojan en nubes públicas.

Por lo tanto, las nubes híbridas ofrecen una alternativa atractiva a empresas que necesitan de una solución de nube personalizada a sus requisitos y necesidades, pudiendo escalar rápidamente sus aplicaciones para adaptarse a las demandas cambiantes del mercado y a la vez mantener un alto nivel de seguridad y control sobre sus datos.

Sin embargo, la utilización de nubes híbridas también trae consigo algunas desventajas. Al conjugar en una misma nube distintas plataformas y tecnologías, cada una con sus configuraciones, políticas de seguridad y herramientas distintas, la integración y gestión de una nube híbrida puede ser mucho más compleja que la de otro tipo de nube. Pueden surgir complicaciones en la integración como problemas de compatibilidad o fallos en la migración entre nubes, haciendo del despliegue y mantenimiento de una nube híbrida un proceso complejo.

Además, a pesar de que una nube híbrida ofrece un nivel superior de seguridad que la nube pública, el hecho de que se utilicen instancias públicas sigue suponiendo un riesgo. A pesar de que los proveedores de servicios en la nube cuentan con fuertes sistemas de seguridad, una configuración incorrecta, por ejemplo, en las interfaces API del lado público de la nube, puede suponer un riesgo de seguridad para toda la nube híbrida.

1.2. Objetivos

La finalidad de este trabajo es desarrollar un sistema que permita disfrutar de todos los beneficios de las nubes híbridas, distribuyendo las cargas de trabajo críticas en nubes privadas y las cargas menos importantes en nubes públicas, a la vez que reduciendo la complejidad típica del despliegue y la gestión de dichos entornos híbridos.

Los objetivos de este trabajo se han dividido en los siguientes:

- El primer objetivo es abordar la simplificación del despliegue de la nube híbrida utilizando la herramienta ofrecida por AWS, AWS CloudFormation. Se pretende automatizar completamente el despliegue de una infraestructura inicial compuesta por una nube de AWS que simule las instalaciones *on-premise* de una empresa, otra nube de AWS y una conexión mediante un túnel VPN.
- A continuación, se estudiarán distintas herramientas y servicios ofrecidos por AWS con la finalidad de poder dotar a la nube híbrida de más flexibilidad y adaptabilidad a las necesidades concretas que le puedan surgir a una empresa. Así, se podrá desplegar una infraestructura más completa que la propuesta en el primer objetivo.

- Como objetivo final, se pretende desarrollar un sistema capaz de desplegar automáticamente una arquitectura híbrida completa y funcional que pueda ser utilizada para albergar verdaderas cargas de trabajo.

1.3. Motivación

Es indudable que las tecnologías en la nube han revolucionado como las personas y las empresas utilizan la tecnología. Y a pesar de que la nube está más que instaurada en el mundo hoy en día, su potencial es enorme. McKinsey habla de las tecnologías *cloud* como una de las tendencias tecnológicas que están transformando el futuro de las TI y del mundo empresarial en general, y en concreto pronostica que para 2025 un 70 % de las empresas utilizarán tecnologías, herramientas y procesos de gestión híbridos o *multicloud*[1].

Y no es sólo como han cambiado el mundo ya, si no la cantidad de posibilidades que la computación en la nube aún guarda. La gran mayoría de las consideradas "tecnologías del futuro", como pueden ser el Internet de las Cosas o IoT, los vehículos autónomos, la realidad aumentada o el metaverso necesitan de las tecnologías en la nube para avanzar. El Multi-Access Edge Computing, o MEC, una de las tecnologías claves para el paso a la sexta generación de las comunicaciones móviles y el *Internet of Everything* necesita de una infraestructura de nube que proporcione los recursos necesarios para poder entregar servicios al borde de la red[2].

Es por la inmensa cantidad de puertas que abre la computación en la nube al mundo tecnológico que personalmente me motiva este trabajo. Se trata de un tema del que sólo tengo un conocimiento teórico, y considero que se trata de una gran oportunidad para familiarizarme y aprender sobre la que sin duda es una de las tecnologías que habilitarán los grandes progresos tecnológicos del futuro.

2. Marco teórico

2.1. Computación en la nube

La computación en la nube, o *cloud computing*, es el acceso bajo demanda y a través de Internet a recursos informáticos como aplicaciones, servidores (físicos y virtuales), almacenamiento de datos, herramientas de desarrollo, capacidades de red y más, hospedados en centros de datos remotos y gestionados por proveedores de servicios en la nube, o CSP[3]. Por lo tanto, la computación en la nube no es una tecnología en sí: es la acción de ejecutar cargas de trabajo en una nube, es decir, un entorno de TI que agrupa y comparte recursos informáticos en una red[4].

El Instituto Nacional de Estándares y Tecnología, o NIST, establece en *"The NIST Definition of Cloud Computing"* cinco características esenciales de la computación en la nube[5]:

1. Auto-servicio bajo demanda: con la computación en la nube se tiene acceso a servicios de computación automáticamente, sin necesidad de interactuar con el CSP. Los clientes pueden acceder a sus cuentas a través de portales web, ver sus servicios en la nube, monitorizarlos y aprovisionarlos de forma independiente.

2. Amplio acceso a la red: una característica esencial es que los servicios en la nube deben ser accesibles desde cualquier tipo de dispositivo y desde cualquier lugar.

3. *Pooling* de recursos: permite que múltiples clientes compartan recursos físicos utilizando un modelo multiusuario, el cual permite asignar y liberar recursos físicos y virtuales según la demanda.

4. Elasticidad: una de las principales características de la computación en la nube es que los recursos se pueden aprovisionar y liberar de forma elástica, por lo que los clientes pueden escalar según su demanda. A ojos del consumidor, la escalabilidad es ilimitada, pudiendo hacerlo en cualquier momento y en cualquier magnitud.

5. Servicio medido: con los servicios en la nube, el sistema de cobro se hace a medida y a un nivel de abstracción acorde con el tipo de servicio. Así, se monitoriza y controla la utilización de recursos, y los pagos se establecen acorde a esto siguiendo un modelo *pay-as-you-go*.

Por lo tanto, la computación en la nube presenta un modelo de negocio que rompe con la forma tradicional en la que las empresas desempeñan sus labores de TI, ofertando acceso a todo tipo de recursos de forma flexible y escalable, a precios ajustados a lo que se consume y de forma accesible y segura. Microsoft Azure, uno de los tres principales proveedores de servicios en la nube, identifica los siguientes beneficios del *cloud computing* [6]:

- Precio: elimina la necesidad de adquirir hardware y software, de configurarlo y ejecutarlo con los costes de mantenimiento que eso conlleva (personal TI especializado, alimentación, refrigeración y más).
- Velocidad: por el auto-servicio bajo demanda que ofrece la nube, es posible desplegar gran cantidad de recursos en minutos a través de Internet. Así, recursos de computación se pueden desplegar y liberar de forma rápida y flexible, liberando a las empresas de tener que planear con antelación y detalle.
- Escala global: la nube ofrece recursos con escalabilidad flexible e ilimitada con posibilidad de desplegarlos en la localización geográfica requerida.
- Productividad: dado que las empresas tienen en la nube acceso a todo tipo de recursos sin necesidad de mantener y gestionarlos, se libera de esa responsabilidad a los equipos de TI.
- Rendimiento: los proveedores de servicios en la nube mantienen y actualizan sus recursos a la última generación de hardware disponible, lo que da lugar a servicios más eficientes y con menores latencias y a mayores economías de escala.
- Fiabilidad: la computación en la nube ofrece soluciones asequibles para el *backup* de datos, la recuperación tras desastres y la continuidad de negocio gracias a que los datos se pueden almacenar de forma redundante en varios servidores del CSP ubicados físicamente en lugares distintos.
- Seguridad: la mayor parte de CSP ofrecen una amplia gama de políticas, medidas y tecnologías de seguridad que, de ser bien aplicados, mejoran la ciberseguridad de sus clientes.

2.2. Arquitectura de la computación en la nube

La computación en la nube sigue un modelo de arquitectura basado en 4 capas que se encargan de proporcionar diferentes funcionalidades y servicios para la infraestructura de la nube. Estas cuatro capas son la capa de hardware, la capa de infraestructura, la capa de plataforma y la capa de aplicación[7].

La capa de hardware se refiere a la infraestructura física de las nubes, como servidores, dispositivos de almacenamiento, routers, switches y sistemas de alimentación y de refrigeración. Esta capa es la inicial de la arquitectura de una nube, siendo la base para la virtualización y la automatización de los recursos TI.

A continuación se encuentra la capa de infraestructura, también conocida como capa de virtualización. Esta capa utiliza tecnologías de virtualización sobre la infraestructura física de la capa anterior para construir un *pool* de recursos de almacenamiento y computación. Es gracias a esta capa que surgen los principales beneficios de la computación en la nube, ya que gracias a dividir servidores físicos en servidores virtuales que pueden ser compartidos y rápidamente asignados a diferentes usuarios y aplicaciones, se obtiene la asignación dinámica de recursos que da lugar a la flexibilidad y escalabilidad de la nube.

La siguiente capa es la de plataforma, y proporciona una plataforma de servicios construida a partir de sistemas operativos y *frameworks* de aplicación donde los desarrolladores pueden construir y desplegar aplicaciones en la nube. Consta de servicios como bases de

datos, herramientas de desarrollo, de análisis de datos y sistemas de gestión de contenido cuyo objetivo es facilitar el despliegue de aplicaciones en la nube.

Finalmente, está la capa de aplicación. En esta capa superior se ejecutan las aplicaciones construidas y desplegadas en la nube, dando lugar a aplicaciones auto-escalables con mejores rendimiento, accesibilidad y costes de operación que las aplicaciones tradicionales.

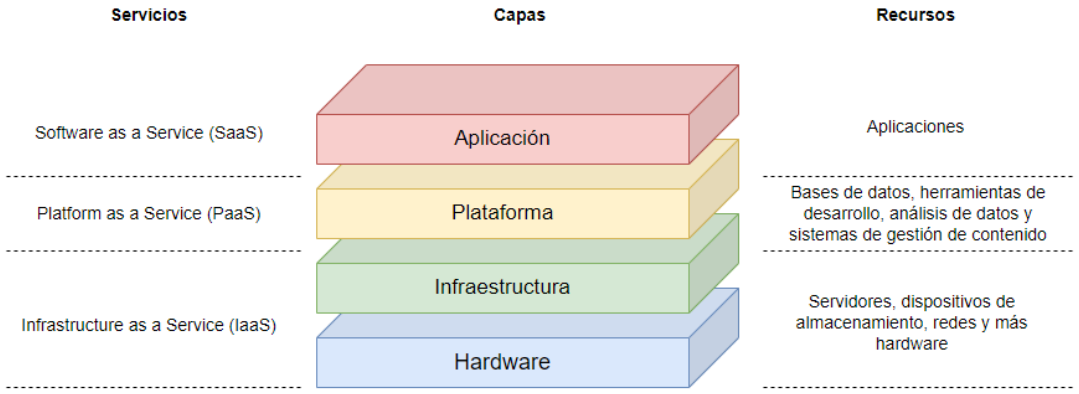

Figura 2.1: Capas de la arquitectura de la computación en la nube

2.3. Modelos de servicios de computación en la nube

La computación en la nube presenta un modelo de negocio orientado hacia los servicios en el que destacan tres tipos: la Infraestructura como Servicio, o IaaS, la Plataforma como Servicio, o Paas, y el Software como Servicio, o SaaS[4].

La Infraestructura como Servicio comprende las dos capas inferiores de la arquitectura de la computación en la nube explicadas en la sección anterior. Se trata de un modelo de servicio en la nube en el cual se ofrecen recursos informáticos virtualizados a través de Internet. Así, el usuario tiene acceso a infraestructura sin tener que invertir en la adquisición de esta: el CSP ofrece la virtualización, el almacenamiento, las redes y los servidores mientras que el usuario se encarga de las aplicaciones, los datos, el sistema operativo o SO, el *middleware* y los tiempos de ejecución.

La Plataforma como Servicio ofrece recursos de la capa de plataforma a los usuarios. Estos recursos incluyen sistemas operativos, entornos de programación, herramientas de desarrollo, bases de datos y demás servicios necesarios para desarrollar y ejecutar aplicaciones. Así, se le ofrece a los usuarios plataformas de software donde desarrollar y ejecutar aplicaciones sin tener que administrar la infraestructura subyacente, por lo que el usuario sólo tiene que proporcionar los datos y la propia aplicación para desplegarla.

El Software como Servicio supone el modelo de servicio en el que se ofrecen recursos de la capa superior de la arquitectura de la computación en la nube. Es decir, un proveedor de SaaS ofrece aplicaciones de software accesibles a través de Internet donde el usuario no tiene que preocuparse ni por la creación de la aplicación ni por su instalación, mantenimiento o actualización, y es capaz de acceder a ella y ejecutarla desde su propio navegador.

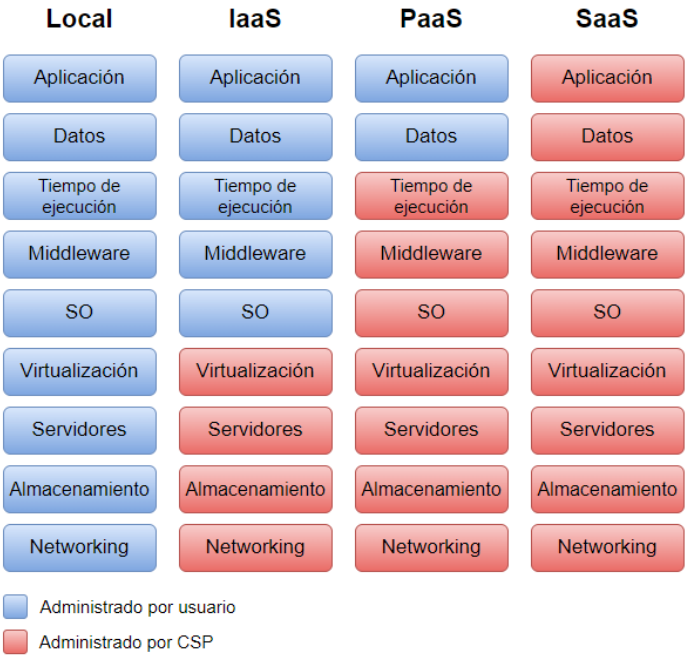

Figura 2.2: Comparativa modelos de servicios en la nube

2.4. Tipos de nubes

Dentro del concepto de la nube existen varios tipos de nubes distintas, de los cuales se van a explicar los tres más importantes para este trabajo. Todas ellas tienen una infraestructura que agrupa recursos de TI como servidores, almacenamiento, bases de datos, redes y entrega de contenido. Sus arquitecturas basadas en la automatización y la virtualización dan lugar a una gran escalabilidad, permitiendo crear entornos de servidores virtuales que se pueden adaptar a lo que se precise (capacidad de procesamiento, almacenamiento, ancho de banda de red, etc) y que pueden ser creados o eliminados rápida y eficientemente.

Sin embargo, los tipos de nube difieren en un aspecto principal: la propiedad.

2.4.1. Las nubes privadas

Una nube privada es un modelo de computación en la nube en el que toda la infraestructura que la compone está dedicada a una organización o empresa, por lo que combina los beneficios de la computación en la nube con la seguridad y control de la infraestructura local u *on-premise*. Por lo tanto, todos los recursos informáticos están dedicados a una sola organización, y son utilizados exclusivamente para ejecutar y alojar las aplicaciones y servicios específicos de esta organización.

El responsable de gestionar y controlar una nube privada es comúnmente la organización propietaria, lo que se traduce en un control total sobre los datos y aplicaciones que se alojan en la nube. Esto, unido a que se pueden implementar medidas de seguridad personalizadas para la protección de los datos de la organización, da lugar a una mayor seguridad y privacidad que en el resto de tipos de nubes. Sin embargo, que la gestión y seguridad sea responsabilidad de la organización propietaria supone que esta debe tener equipos de TI

especializados en áreas como virtualización, automatización y seguridad capaces de implementar, configurar y mantener la infraestructura de forma eficiente y de asegurar que la nube cumple con los requisitos de escalabilidad, rendimiento, disponibilidad y seguridad de la organización.

2.4.2. Las nubes públicas

Una nube pública es una nube cuyo dueño, un CSP, ofrece sus recursos como servicios a través de Internet. La infraestructura reside en las instalaciones y es mantenida por el CSP, y este la pone a disposición de los usuarios a través de una interfaz de autoservicio de forma automática. Por lo tanto, la infraestructura de la nube es compartida por todos los usuarios que contratan recursos.

Los CSP agrupan grandes cantidades de recursos en centros de datos y los distribuyen por todo el mundo, buscando alquilar estos recursos a multitud de empresas y usuarios a través de Internet. Esto conlleva muchas ventajas para los usuarios: al haber tales cantidades de recursos disponibles, la escalabilidad y flexibilidad a ojos del usuario es ilimitada sin necesidad de grandes inversiones iniciales ni de tener equipos especializados en mantener infraestructura.

Esta escalabilidad, flexibilidad y accesibilidad sin embargo también conllevan la pérdida de control sobre los datos y aplicaciones que se despliegan. No significa que en las nubes públicas los datos estén desprotegidos, ya que los CSP implementan medidas de seguridad para proteger los datos de sus clientes, pero si que puede suponer un problema para empresas que traten datos confidenciales.

Para solventar este problema los CSP ofrecen una alternativa que se estudiará en detalle en este trabajo: las nubes privadas virtuales o VPC. Este servicio permite desplegar redes privadas virtuales completamente aisladas del resto de la infraestructura compartida de una nube pública. Así, un usuario puede desplegar una red virtual completamente privada y personalizada, pudiendo definir sus propias subredes, reglas de firewall, tablas de rutas y direcciones IP ajustadas a sus necesidades de rendimiento y seguridad[8].

La principal ventaja que se estudiará en este trabajo ofrecida por las VPC es la posibilidad de que los usuarios conecten sus redes locales con la nube pública a través de una conexión VPN (Virtual Private Network) segura, dando lugar a una hibridación en la que la VPC es una extensión de la infraestructura local, por lo que se tiene acceso a los beneficios de la nube pública de forma más segura y controlada.

2.4.3. Las nubes híbridas

Una nube híbrida es un entorno de computación en la nube mixto en el que se combinan servicios de distintos entornos, como pueden ser entornos de nubes privadas, públicas o instalaciones locales. La combinación de distintos entornos, cada uno con sus ventajas y desventajas, puede proporcionar una solución *cloud* integral en el que las cargas de trabajo se destinan al entorno que mejor encaja con sus características.

Así, se obtiene el entorno más versátil y flexible posible, permitiendo alojar cargas de trabajo que precisan de gran escalabilidad y flexibilidad en la nube pública sin perder el control de datos confidenciales, que serán tratados en entornos de nube privada u *on-premise*. Es por esto que los entornos híbridos son los más adoptados a día de hoy por las

empresas: según un informe llevado a cabo por Cisco en 2022, un 82 % de las empresas tecnológicas líderes adoptan soluciones híbridas para sus entornos en la nube[9].

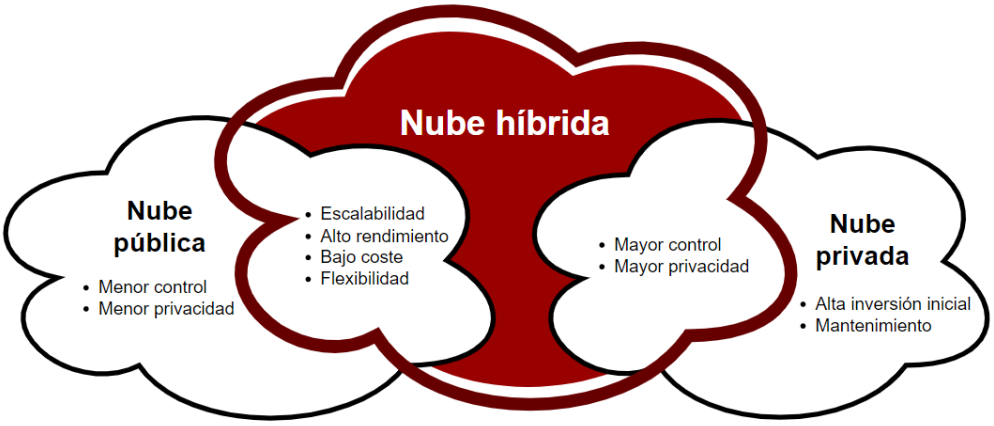

Figura 2.3: Esquema nubes híbridas frente a nubes públicas y privadas

Para que una nube sea verdaderamente híbrida, todos sus entornos deben estar interconectados completamente, de forma que a pesar de las diferencias en tecnologías, tenencia y ubicación, todos los entornos funcionen como una infraestructura única y combinada[10]. Esto supone también una desventaja: la complejidad. La implementación de entornos híbridos supone un incremento de complejidad con respecto a entornos normales en todos los pasos del desarrollo y mantenimiento de infraestructura: modelado, despliegue, gestión, monitorización y seguridad.

La arquitectura de una nube híbrida puede ser muy variada, ya que cada usuario la personaliza según sus necesidades: puede incorporar instalaciones *on-premise*, varias nubes públicas y varias nubes privadas.

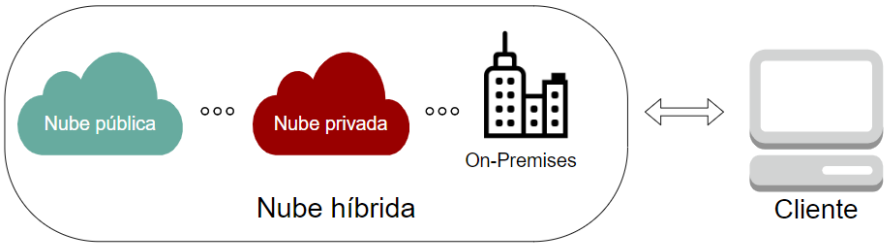

Figura 2.4: Composición de nubes híbridas

Sin embargo, sin importar la conformación específica, todas las nubes híbridas tienen rasgos en común[10]:

- Integración de datos: todos los datos de una misma organización deberán estar sincronizados en los entornos públicos y privados, lo cual puede suponer implementar una solución técnica que asegura la consistencia de los datos de forma automática.
- Conexión de red: debe haber conectividad entre las nubes privadas, las nubes públicas y los entornos locales, ya sea a través de la Internet pública o de una red privada.

- Gestión unificada: una buena implementación de nube híbrida cuenta con una herramienta global que lleva a cabo la gestión, sin necesidad de gestionar las nubes individualmente.

2.5. Amazon Web Services

Un proveedor de servicios en la nube, o CSP, es una empresa que ofrece recursos de computación a través de Internet de forma escalable y bajo demanda. Un CSP ofrece como servicios recursos informáticos a través de los tres modelos de servicio de computación en la nube estudiados previamente: IaaS, PaaS y SaaS.

La elección de CSP debe hacerse atendiendo a una gran cantidad de factores, en función de los requisitos funcionales, de precio, de seguridad y logísticos que pueda tener la empresa o el proyecto. Para este proyecto se eligió Amazon Web Services, AWS a partir de ahora, como CSP debido a los siguientes motivos:

- AWS es el mayor proveedor de servicios de IaaS en el mundo, con una cuota de mercado de 40,8 % en 2022 según un estudio de Gartner[11]. Dado que este proyecto está principalmente orientado al modelo de la infraestructura como servicio (IaaS), se busca el proveedor con más fuerza en este sector.
- AWS es también el CSP que más servicios de IaaS ofrece, contando con más de 200 a día de hoy[12], y con constantes novedades. Al investigar distintas modalidades de hibridación de entornos, se busca el proveedor que más variedad tenga.
- La documentación de AWS es muy completa, compuesta por extensas explicaciones, ejemplos y *workshops* para todos sus servicios
- Dado que este trabajo se realiza en colaboración con el Instituto de Ingeniería del Conocimiento, o IIC, se escogió a AWS por ser su principal proveedor de servicios en la nube.

Como se ha explicado previamente, AWS cuenta con una extensa cantidad de servicios, varios de los cuales serán utilizados en este proyecto, por lo que serán explicados a continuación.

2.5.1. AWS CloudFormation

La Infraestructura como Código, o IaC, es una práctica que consiste en utilizar *scripts* de código para describir y gestionar infraestructura de TI en lugar de hacerlo de forma manual. Así, se abstrae la configuración y administración de infraestructura para tratarla como cualquier otro tipo de software, permitiendo definir y aprovisionar recursos de forma automatizada.

La herramienta que AWS ofrece para la IaC se llama AWS CloudFormation, que se define como un servicio de IaC que permite modelar, aprovisionar y administrar recursos de AWS de forma automatizada, segura y repetible[13]. Por lo tanto, permite definir y desplegar infraestructura de AWS a través de una plantilla en formato YAML o JSON en la que se describan los recursos requeridos, y sus propiedad, relaciones y configuraciones específicas.

Una vez el usuario monta la plantilla de la infraestructura que desea desplegar, la despliega a través de AWS CloudFormation. Esto se hace desplegando una pila o *stack* en

CloudFormation, que es una instancia de la plantilla implementada en AWS, y CloudFormation se encarga de coordinar, aprovisionar y gestionar las dependencias de los recursos en el orden adecuado, lo que garantiza que se creen correctamente.

Además de la simplificación del despliegue de infraestructura a través de la automatización, CloudFormation también interviene en el resto de ciclo de vida de los recursos: facilita la gestión, actualización y eliminación de un conjunto de recursos tratándolos como una sola unidad, en vez de hacerlo de manera individual. Por lo tanto, se trata de una herramienta orientada a la gestión de recursos en bloque: al definir un conjunto de recursos dentro de un *stack*, el usuario puede desplegarlos, administrarlos, escalarlos, actualizarlos y eliminarlos de forma predecible, repetible y automática[13].

2.5.2. AWS Command Line Interface

AWS Command Line Interface, AWS CLI a partir de ahora, es la interfaz de línea de comandos proporcionada por AWS. Se trata de una herramienta que unifica todos los recursos de AWS y ofrece una interfaz consistente para interactuar con ellos. AWS CLI es una herramienta de línea de comandos multiplataforma que permite interactuar de todas las formas que permite la consola manual de AWS, pero a través de comandos de terminal.

AWS CLI funciona a través de la API de AWS, por lo que los comandos que se ejecutan a través de la interfaz de línea de comandos lanzan solicitudes a la API de los servicios de AWS, proporcionando una manera programática y automática de administrar recursos e interactuar con servicios de AWS[14].

En concreto para este proyecto, AWS CLI posibilita la creación, administración y gestión de todo tipo de recursos de IaaS a través de línea de comandos, ofreciendo una herramienta más eficaz y automática que la consola manual de AWS, la AWS Management Console.

2.5.3. Amazon VPC

Amazon Virtual Private Cloud, Amazon VPC en adelante, es un servicio de AWS que permite crear una nube privada virtual dentro de la infraestructura de AWS, ofreciendo un entorno aislado del resto de nube pública. Al desplegar una VPC, el usuario puede construir su propia red privada dentro del entorno público de AWS, por lo que puede personalizar y controlar completamente su entorno de nube.

Por lo tanto, Amazon VPC es un recurso extremadamente útil a la hora de crear entornos de nube híbrida, ya que permite establecer un puente seguro y fluido entre la infraestructura local de una empresa y la nube de AWS. Este servicio brinda a las empresas la capacidad de desplegar todo tipo de servicios de computación en la nube de forma segura y controlada a la vez que aprovechando la escalabilidad y flexibilidad de esta, y conectarlos con sus instalaciones locales a través de una VPN[15].

Dentro de la red privada virtual que crea una VPC se puede controlar y personalizar el entorno de red, pudiendo seleccionar los rangos de direcciones IP, la cantidad de subredes, la configuración de tablas de rutas y la aplicación de reglas de seguridad, conocidas como Access Control Lists o ACL.

Al ser un servicio indispensable para el despliegue de soluciones en el entorno de AWS, se ofrecen muchos servicios para dotar a las VPC de conectividad con el exterior: otras

VPC, entornos *on-premise* o Internet. A continuación se describen los servicios utilizados en este trabajo para conectar VPC´s con entornos *on-premise*.

2.5.4. AWS Site-to-Site VPN

AWS Site-to-Site VPN permite establecer conexiones seguras y encriptadas a través de la utilización de túneles VPN IPsec. Por lo tanto, es una solución muy útil para conformar arquitecturas de nube híbrida, ya que permite establecer una conexión segura y fluida a través de Internet entre instalaciones locales y servicios de AWS[16].

Al configurar una conexión Site-to-Site VPN, se despliegan túneles VPN IPsec (protocolo de seguridad de Internet) entre el dispositivo VPN configurado localmente y un dispositivo VPN virtual de AWS. Se despliegan dos túneles redundantes para aumentar la disponibilidad de la conexión.

La seguridad de los túneles IPsec se hace siguiendo tres funciones de criptografía: autenticación, confidencialidad e integridad.

Para generar el túnel IPsec, se utiliza el protocolo Internet Key Exchange(IKE). Este protocolo se encarga de establecer y negociar los parámetros de seguridad y las claves de cifrado entre los dos extremos o *endpoints* de la VPN. AWS Site-to-Site permite el uso de IKEv1 e IKEv2, siendo esta última la recomendada por su mayor eficiencia y seguridad.

Una vez generado el túnel, AWS ofrece dos métodos para que los *endpoints* se autentiquen entre sí: o a través de una clave compartida previamente, o PSK, o a través de autenticación basada en certificados. PSK ofrece una autenticación más simple y rápida, mientras que la autenticación basada en certificados es más fuerte, al utilizar certificados digitales emitidos a través de AWS Certificate Manager (un servicio de AWS de emisión y gestión de certificados SSL/TLS).

Para el encriptado de la conexión VPN, primero se utiliza el algoritmo Diffie-Hellman para generar claves de cifrado compartidas entre los dos *endpoints*. Este algoritmo permite establecer una clave de cifrado compartida entre los dos extremos del túnel VPN sin transmitir directamente la clave, ya que se hace a través de Internet. AWS Site-to-Site VPN admite varios grupos de Diffie-Hellman, que ofrecen diferentes longitudes de claves y niveles de seguridad. La posibilidad de elegir grupos de Diffie-Hellman proporciona flexibilidad para adaptarse a los niveles de seguridad requeridos, obteniendo una mayor seguridad con los grupos de mayor índice, a cambio de pérdida de rendimiento.

Una vez se han generado claves de cifrado compartidas, se utilizan para encriptar el tráfico utilizando Advanced Encryption Standard o AES. También se aplican algoritmos de *hashing*, como SHA-2, para asegurar la integridad de los datos transmitidos. Para estos dos algoritmos se ofrecen varias versiones, entre las cuales se debe elegir teniendo en cuenta los requisitos de seguridad y rendimiento del sistema[17].

Al configurar una conexión Site-to-Site VPN es necesario establecer a ambos lados de la conexión *endpoints*. Un VPN *endpoint* actúa como punto final de una conexión VPN, permitiendo la transferencia de datos cifrados: configura y gestiona los parámetros de seguridad de la VPN, autentica y autoriza las conexiones, cifra el tráfico de salida y descifra y encamina el tráfico de entrada.

En las conexiones Site-to-Site VPN de AWS es el endpoint local el que debe establecer la conexión VPN, por lo que debe ser un dispositivo VPN dedicado, un enrutador compatible

	Algoritmos y protocolos de seguridad admitidos por AWS Site-to-Site VPN			
	Negociación de claves	**Grupos Diffie-Hellman**	**Encriptación**	**Integridad**
Fase 1	IKEv1, IKEv2	2, 14-24	AES128, AES256, AES128-GCM-16, AES256-GCM-16	SHA1, SHA2-256, SHA2-384, SHA2-512
Fase 2	IKEv1, IKEv2	2, 5, 14-24	AES128, AES256, AES128-GCM-16, AES256-GCM-16	SHA1, SHA2-256, SHA2-384, SHA2-512

Tabla 2.1: Algoritmos y protocolos de seguridad admitidos por AWS Site-to-Site VPN

con VPN u otro tipo de software VPN capaz de iniciar y establecer la conexión local hacia la infraestructura de AWS.

Por lo tanto, el *endpoint* del lado de AWS se encargará de autenticar la conexión, descifrar y enrutar el tráfico entrante y cifrar el tráfico saliente. En este trabajo se utilizan dos recursos de AWS distintos como *endpoint* del lado de AWS, que se describirán a continuación.

2.5.4.1. Virtual Private Gateway

Un Virtual Private Gateway, VGW a partir de ahora, es un enrutador lógico distribuido y completamente redundante que se sitúa al borde de una VPC. Dado que tiene capacidad para actuar como *endpoint* de conexiones VPN, se utiliza como concentrador VPN en el lado de AWS para conexiones Site-to-Site VPN.

Por lo tanto, se puede asociar un VGW a una VPC para actuar como el punto de entrada y salida de tráfico dicha VPC hacia la conexión VPN. Es importante destacar que un VGW sólo se puede asociar a una VPC, por lo que empleando este recurso solo se pueden formar una conexión *one-to-one* entre un entorno local y una VPC. Para generar el túnel VPN, también hace falta establecer un Customer Gateway (CGW), para dar información a AWS del dispositivo VPN dedicado del lado local de la conexión.

Al establecer una conexión Site-to-Site VPN, el VGW admite enrutado dinámico a través del protocolo Border Gateway Protocol (BGP), pero el dispositivo VPN local también debe soportarlo[16]. El protocolo BGP habilita el intercambio de información de enrutado entre Sistemas Autónomos (AS) a través de Internet. Un AS es un conjunto de redes controladas por una única entidad, por lo que BGP es el protocolo de enrutamiento utilizado para intercambiar información de enrutado entre de estas redes.

Al habilitar la utilización de BGP, el protocolo permite que el dispositivo VPN local y el VGW intercambien información de enrutado de forma dinámica y automática. La utilización de BGP supone una serie de ventajas que facilitan el establecimiento y administración de la conexión, y aumentan su eficiencia:

- Intercambio dinámico de rutas: en vez de tener que configurar las rutas estáticas manualmente en ambos extremos de la conexión, BGP propaga las rutas de forma automática en el momento de creación del túnel, y si ocurre cualquier cambio en algún extremo.
- Convergencia rápida: como se ha explicado previamente, BGP aplica un mecanismo

de intercambio de información basado en eventos, por lo que si ocurre cualquier cambio en las rutas, BGP propaga esta información rápidamente a través de la VPN. Esto supone una mejora de la resiliencia y la disponibilidad de la conexión, ya que permite una convergencia rápida de las rutas en casos de cambios o fallos de conectividad.

- Aumento de escalabilidad: BGP es un protocolo escalable y flexible que permite el intercambio de rutas en redes grandes. Esto puede ser útil para grandes entornos empresariales con múltiples ubicaciones, ya que BGP es capaz de manejar de forma eficiente una gran cantidad de rutas, y adaptarse a cambios en la topología de la red sin necesidad de grandes reconfiguraciones manuales.
- Funcionalidad de gestión de políticas de enrutamiento: BGP ofrece control preciso sobre cómo anuncia y recibe rutas de enrutamiento, lo cual habilita la aplicación de controles de flujo de tráfico.

2.5.4.2. AWS Transit Gateway

AWS Transit Gateway, TGW a partir de ahora, es un servicio de red altamente escalable y flexible diseñado para extender la funcionalidad del Virtual Private Gateway explicado previamente. Además de ofrecer todas las funcionalidades que ofrece el VGW, suma una larga lista de funcionalidades nueva que hace del TGW un recurso muy valioso a la hora de hibridar entornos de nube.

Transit Gateway actúa como un concentrador de tráfico de recursos de AWS, permitiendo la interconexión de VPCs y redes locales al actuar como un eje central de las comunicaciones, por lo que elimina la necesidad de utilizar varias conexiones punto a punto. Es decir, en vez de tener que desplegar un túnel VPC para cada VPC como habría que hacer con un VGW, AWS Transit Gateway permite centralizar el paso de todo el tráfico perteneciente a una organización por un mismo punto, y al poder actuar como *endpoint* VPN en conexiones Site-to-Site VPN, enrutarlo hacia las instalaciones locales de dicha organización. Para conectar una VPC a un TGW simplemente hay que desplegar en ella un Transit Gateway Attachment y asociarlo al TGW deseado, y el enrutamiento y conectividad entre el TGW y las demás redes conectadas se resolverá de forma automática.

Esto supone el primer y principal beneficio de TGW, el poder conectar hasta 10,000 VPC a un único TGW si se dispone de una cuenta de AWS de soporte empresarial. Esta centralización de la conectividad supone una gran simplificación de la arquitectura y de su gestión, eliminando la necesidad de establecer y administrar conexiones individuales al centralizar todo el tráfico en un único punto de entrada. Esto da pie también a una gran flexibilidad y escalabilidad, al poder alojar cargas de trabajo en nuevas VPC e interconectarlas con las instalaciones locales u otras VPCs de forma rápida y sencilla.

AWS Transit Gateway ofrece otras funcionalidades muy útiles para la gestión de entornos híbridos. Un Transit Gateway puede ser compartido a través de AWS RAM, lo que permite que VPCs de distintas cuentas de una misma organización se conecten, facilitando en gran medida la gestión, consolidando la conectividad y facilitando la colaboración y el intercambio de recursos entre múltiples cuentas de AWS. Dado que empresas alojan sus cargas de trabajo en distintas cuentas de AWS por motivos organizativos, de aislamiento y seguridad o de gestión de costos y facturación, esta funcionalidad es idónea para dichos entornos híbridos. Otra funcionalidad útil para entornos híbridos es que además de actuar

como un punto central de comunicación a nivel 3 de red, TGW también ofrece soporte de DNS, resolviendo los DNS de los distintos *attachments*[18].

Por lo tanto, se identifican los siguientes beneficios de centralizar todo el tráfico de AWS de una organización a través de Transit Gateway :

- Conectividad simplificada: al actuar como enrutador central al que se pueden conectar todas las VPC y conexiones VPN deseadas, Transit Gateway permite consolidar y configurar el enrutamiento de toda una red híbrida en un único punto.
- Gestión simplificada: elimina la necesidad de establecer y administrar conexiones VPN punto a punto con cada recurso desplegado en AWS, simplificando en gran medida la gestión de entornos híbridos. Ofrece resolución de DNS entre los recursos de AWS conectados, facilitando la interoperabilidad, y facilita la gestión de organizaciones con múltiples cuentas de AWS al poder ser compartido a través de AWS RAM.
- Control y monitorización: dado que el Transit Gateway está diseñado para que lo atraviese todo el tráfico de una organización, supone el punto idóneo para monitorizar el tráfico. Ofrece el servicio de Transit Gateway Network Manager, que permite dicha monitorización desde una consola central.
- Seguridad: la utilización de Transit Gateway como enrutador central supone también una mejora de seguridad, ya que permite que todos los recursos de una organización se comuniquen sin abandonar la red privada global de AWS, por lo que no se expone al Internet público.

2.5.5. Amazon EC2

Amazon Elastic Compute Cloud, Amazon EC2 de ahora en adelante, es la principal plataforma de computación ofrecida por AWS, siendo uno de los primeros 3 que sacaron al mercado en su fundación en 2006.

Cualquier tipo de aplicación necesita de servidores de computación, y EC2 es la principal herramienta ofrecida por AWS para esto. En vez de tener que enfrentarse a las altas inversiones iniciales y los problemas logísticos que supone adquirir servidores propios, EC2 ofrece capacidad de computación accesible y de bajo coste que se puede adquirir, escalar y liberar muy rápidamente.

Por lo tanto, EC2 ofrece la posibilidad de lanzar servidores virtuales con facilidad[19]:

1. Primero, se debe elegir el tipo de instancia que lanzar en función de las capacidades necesitadas por las cargas de trabajo que se vayan a alojar. Una instancia es un servidor virtual en ejecución, y sus capacidades de computación y memoria vienen dadas por su tipo. Por lo tanto, dependiendo de los requisitos de rendimiento que tengan las cargas de trabajo, se pueden elegir que tipos de instancia son más adecuados, existiendo instancias de uso genérico, orientadas a la computación, a la memoria o al almacenamiento.
2. A continuación, se debe seleccionar que Amazon Machine Image (o AMI) se utiliza. La AMI es la plantilla de la configuración que tiene cada instancia en el momento de su ejecución, preconfigurando el sistema operativo, software adicional, configuraciones personalizadas y datos asociados a la instancia. AWS ofrece gran variedad de

AMIs que se pueden seleccionar por diversos filtros: región, sistema operativo (Linux, Windows u opciones personalizadas), arquitectura (32-bit o 64-bit), permisos de despliegue y almacenamiento.

3. El tercer paso es seleccionar que tipo de almacenamiento se va a utilizar para las instancias, existiendo varias opciones con diferentes rendimientos, durabilidad y precio:

   - Amazon Elastic Block Storage, o EBS: sistema de almacenamiento en bloque que ofrece volúmenes de almacenamiento que se pueden conectar a instancias, siendo estos duraderos (los datos persisten incluso si se elimina la instancia) y flexibles (se pueden modificar y escalar de forma dinámica).
   - Amazon EC2 instance store: almacenamiento local en la instancia EC2. Al estar ubicado en la misma instancia, ofrece alto rendimiento, pero no ofrece persistencia de los datos en el caso de que se detenga la instancia. Al no aplicar cargos adicionales por almacenamiento, tiende a ser menos costoso.
   - Amazon Elastic File System, o EFS: servicio de almacenamiento de archivos administrado, utilizado para almacenar y compartir archivos en la nube y que se puede integrar con EC2. Es escalable y permite conexiones simultáneas ilimitadas, por lo que es adecuado para crear aplicaciones empresariales de alta demanda, por lo que las instancias pueden acceder a sistemas de archivos compartidos como si estuvieran en una red de área local, o LAN.

4. Posteriormente, se debe seleccionar la configuración de red de la instancia. Por defecto, las instancias se despliegan en una VPC, por lo que su configuración de red será la misma que la definida por la VPC en la que se despliegue. Además, cada instancia tiene una o varias Interfaces de Red Elástica, o ENI, que le permite conectarse a una o varias subredes de su VPC y tener asignadas direcciones IP públicas o privadas. Por lo tanto, esta ENI permite el enrutamiento de la instancia con los demás elementos de su VPC y con el exterior.

2.5.6. Amazon Route 53

Amazon Route 53 es un servicio web de Sistema de Nombres de Dominio (DNS) altamente escalable proporcionado por AWS. Su principal objetivo es la administración y control de la resolución de nombres de dominio en la web, traduciendo los nombres de dominio de servicios alojados en el entorno de AWS en las direcciones IP reales de los servidores donde está alojado el contenido web asociado.

Route 53 ofrece una gran variedad herramientas y servicios para distintos modelos de resolución DNS, ofreciendo también la posibilidad de que actúe como *firewall* o balanceador de carga o con posibles herramientas de monitorización y seguridad. Sin embargo, sus tres principales funciones son las siguientes[20]:

- Registro de nombres de dominio: permite registrar nuevos nombres de dominio como puede ser *ejemplo.com* desde el mismo servicio, simplificando en gran medida el proceso de compra y gestión de nombres de dominio.
- Resolución de nombres de dominio: proporciona la capacidad de traducir nombres de dominio a direcciones IP y viceversa, ofreciendo una infraestructura global que asegura una resolución rápida y una alta disponibilidad. Ofrece posibilidades de

redireccionamiento y enrutamiento avanzado, con opciones de redirección de tráfico inteligente en función de reglas predefinidas.

- Registro de salud de los recursos: envía peticiones automáticas a los recursos asociados a los nombres de dominio registrados, comprobando si son alcanzables y su disponibilidad y estado. En el caso de que un recurso deje de estar disponible, puede redirigir automáticamente el tráfico a recursos alternativos.

2.5.6.1. Amazon Route 53 Resolver

En este proyecto se va a utilizar en especial un servicio de Route 53 en concreto, Route 53 Resolver. Este servicio es una extensión del servicio de resolución de DNS orientada en concreto a entornos de red híbridos complejos, al ofrecer capacidades avanzadas como la resolución de DNS entre VPCs y entornos locales o entre diferentes VPCs y la integración con otros servicios de AWS como Transit Gateway.

Route 53 Resolver es capaz de resolver peticiones DNS de no solo nombres de dominio internos de AWS, si no que también es capaz de resolver ambos nombres de dominio públicos de internet y, de configurarse, nombres de dominio definidos por una empresa en su servidor de DNS interno.

Es esta última capacidad la que hace de AWS Route 53 Resolver un servicio especialmente interesante para este proyecto, dado que permite la resolución de DNS centralizada de un entorno híbrido en su plenitud. Es decir, permite que máquinas en el entorno local resuelvan nombres de dominio en la nube y viceversa, obteniendo así un entorno híbrido mucho más manejable e intuitivo, al permitir la comunicación entre entornos a través de nombres de dominio en vez de direcciones IP[21].

3. Requisitos

3.1. Requisitos generales del sistema

Como se ha visto en el Marco Teórico, la implementación de una arquitectura de nube híbrida conlleva una larga lista de beneficios para una empresa u organización, permitiendo acceder a la escalabilidad, flexibilidad y al extenso *pool* de recursos que ofrece la nube pública además de reducir las inversiones iniciales en equipos físicos, sin perder el control, seguridad y más sencillo cumplimiento normativo que permite el alojar cargas de trabajo localmente.

Sin embargo, la flexibilidad y gran variedad de configuraciones para el alojamiento de cargas de trabajo que permite todas estas ventajas también supone la mayor desventaja de la implementación de arquitecturas híbridas, la complejidad. El modelaje, despliegue, gestión y monitorización en dichas arquitecturas puede suponer un gran reto, y de hacerse mal, puede dar lugar a graves problemas logísticos y de seguridad.

Es por esto que el principal requisito del sistema propuesto en este trabajo es la agilización de dichos procesos mediante la automatización de todas estas labores a la hora de implementar y mantener una arquitectura híbrida. Por lo tanto, se pueden establecer los dos requisitos principales de este proyecto:

- Hibridación pura: el objetivo definitivo de este proyecto es lograr una hibridación pura de la infraestructura desplegada, es decir, que todos los recursos desplegados, sin importar si están ubicados en las instalaciones locales o en la nube pública, deben poder verse y comunicarse como si se encontrasen en el mismo entorno. Por lo tanto, el sistema debe ser capaz de integrar y conectar de forma eficiente ambos entornos, debiendo ofrecer una integración absoluta e imperceptible de los servicios de cada lado. Este requisito también incluye la necesidad de una gestión unificada de toda la infraestructura híbrida, pudiendo monitorizar y gestionar el estado de todos los recursos e implementar cambios, ajustes y actualizaciones en la arquitectura entera.
- Automatización: el requisito que aporta verdadero valor a este sistema es la automatización. A través de los métodos descritos en el Desarrollo se debe obtener un sistema que agilice y libere de carga técnica el despliegue, la gestión y la monitorización de infraestructuras de nube híbrida. Procesos como el de despliegue, de hacerse de forma manual, resultan en procesos lentos y complejos técnicamente, además de ser más propensos a errores y más difícilmente repetibles. Mediante la automatización que provee este sistema, se deben obtener procesos de despliegue, gestión y monitorización más eficientes, consistentes y reproducibles, además de ser menos propensos a errores, más flexibles y más escalables.

Es en estos dos requisitos donde se ubica la base del valor que aporta este proyecto,

ya que son en los que se centra su verdadera utilidad. Sin embargo, se deben establecer otra serie de requisitos para el sistema para abordar las diversas necesidades y retos que surgen al operar en entornos complejos, dinámicos y de alta demanda como pueden ser infraestructuras de TI corporativas. Por lo tanto, se establecen los siguientes requisitos:

- Escalabilidad: como una de las principales ventajas que ofrece la nube pública, la escalabilidad debe ser un requisito imprescindible de este sistema. Para una aplicación empresarial, poder adaptarse a la demanda cambiante a través de una escalabilidad eficiente es de vital importancia. Por lo tanto, este sistema debe desarrollar arquitecturas híbridas capaces de ampliarse y reducirse rápida y eficientemente para ofrecer una distribución óptima de los recursos, no sólo para ser capaces de ofrecer servicio en demandas crecientes, pero también para evitar infrautilizar recursos.
- Seguridad: para una empresa u organización, la información es su activo más valioso. Por lo tanto, en la utilización de la nube por parte de empresas u organizaciones la seguridad se convierte en un requisito crítico. Esto se potencia en arquitecturas de nube híbrida, donde la información se distribuyen entre múltiples entornos. Por lo tanto, este sistema debe asegurar la confidencialidad, integridad y disponibilidad de los recursos que contiene, debiendo contar con mecanismos de autenticación, autorización y cifrado en todas las comunicaciones.
- Resiliencia: dado lo críticos que pueden ser las cargas de trabajo y recursos alojados en este sistema, debe ser capaz de resistir y recuperarse ante fallos o interrupciones de una forma eficaz y eficiente. Por lo tanto, el sistema debe contar con estrategias de tolerancia a fallos y redundancia y métodos de recuperación para minimizar el tiempo de inactividad en caso de cualquier incidencia.
- Orquestación: para lograr una gestión coherente y eficiente de toda la infraestructura híbrida, el sistema debe permitir la ejecución de flujos de trabajo a lo largo de toda la infraestructura sin necesidad de intervención manual. Por lo tanto, el sistema debe coordinar los diversos componentes que componen la infraestructura híbrida de forma automática para permitir la ejecución automática de dichos flujos de trabajo.

Estos requisitos adicionales complementan los requisitos principales y abordan aspectos críticos claves.

4. Desarrollo

El desarrollo de este proyecto está dividido en dos módulos. La primera fase estará enfocada en el despliegue automático de arquitecturas híbridas, y la segunda se centrará en incorporar herramientas de gestión que faciliten la implementación y utilización de dichas arquitecturas híbridas.

4.1. Fase de despliegue

El desarrollo de la primera fase de este proyecto se centra en la aplicación de herramientas y servicios de AWS y externos para lograr un sistema capaz de desplegar de forma completamente automática arquitecturas híbridas.

El principal objetivo de esta fase es lograr una solución eficiente y ágil para la implementación de infraestructuras, teniendo como enfoque principal la automatización. Dadas la flexibilidad y escalabilidad que permiten las nubes híbridas, este sistema pretende simplificar una de las principales desventajas que estas presentan: la complejidad.

Por lo tanto, el enfoque de la fase de despliegue está centrado en la simplificación y automatización del despliegue de distintas arquitecturas de nube híbrida. Para lograr esta automatización, se ha aplicado un enfoque basado en varias capas de abstracción sobre la infraestructura, para así ir obteniendo cada vez un mayor grado de automatización.

Estas capas de abstracción se refieren a niveles de simplificación y generalización que se aplican a la infraestructura base subyacente. Por lo tanto, con cada capa de abstracción, se busca encapsular y simplificar aspectos técnicos de la capa inferior para buscar ofrecer interfaces y herramientas más simples de utilizar. A través de esta metodología de separación de detalles técnicos de implementación y configuración a través de abstracción por capas se logran varios beneficios para el sistema:

- Simplificación: cada capa de abstracción encapsula detalles técnicos complejos de la capa inferior, simplificando la interacción con la infraestructura. Esto posibilita facilitar el proceso de modelado, implementación y configuración de infraestructura.
- Reproducibilidad: al separar los detalles técnicos en diferentes capas, se posibilita la reutilización de componentes y configuraciones en distintas arquitecturas. Esto no sólo facilita el proceso de desarrollo, si no que permite adaptar el sistema final a las necesidades del usuario de una forma más eficaz y simple.
- Estandarización: de manera similar a la reusabilidad de las capas de abstracción, estas también ofrecen un conjunto de modelos y patrones predefinidos para el despliegue de distintas infraestructuras. La estandarización y repetibilidad de las capas

permite el despliegue de grandes y complejas infraestructuras de manera homogénea, con una mayor consistencia y con menor probabilidad de errores puntuales.

- Automatización: como principal enfoque de esta fase, la creación de herramientas simplificadas a partir de una compleja infraestructura subyacente, se facilita la automatización de estas. Al obtener una herramienta más simple, es más sencillo automatizar la siguiente capa. Así, los procesos se pueden ir simplificando, reduciendo la necesidad de intervención manual para ejecutarlos de manera programática. Esto ahorra tiempo y recursos además de reducir la probabilidad de errores.

Por lo tanto, se aplican diferentes herramientas en cada capa de abstracción, para conseguir con cada capa simplificar y agilizar el despliegue de infraestructura con respecto a la capa anterior. A continuación se detallan las capas de abstracción y las herramientas utilizadas para obtenerlas:

1. La primera capa de abstracción se trata de la propia consola de usuario de AWS. Esta primera herramienta presenta el mayor nivel de automatización de todo el proceso, eliminando gran parte de la complejidad técnica que conlleva la configuración y despliegue manual de infraestructura a través de una interfaz de usuario intuitiva y fácil de usar.

   Poniendo el ejemplo del despliegue de un servidor, en vez de tener que lidiar con todas las tareas que supone configurar un servidor, la consola de AWS establece un flujo de trabajo predefinido en el cual el usuario especifica los parámetros necesarios (como la región, el tipo de instancia o el almacenamiento) y luego se encarga de crear y configurar la instancia automáticamente según esos parámetros.

   Sin embargo, a pesar de que la consola simplifica en gran parte ciertos procesos, el grado de automatización que proporciona es limitado. La necesidad de usar la consola de forma manual conlleva limitaciones a la hora de desplegar infraestructuras más complejas, ya que implica un mayor tiempo de implementación, una difícil repetibilidad y una mayor probabilidad de errores humanos.

2. A causa de estas limitaciones surge la necesidad de utilizar herramientas de IaC como AWS CloudFormation que permiten modelar y desplegar infraestructura de forma programática y reproducible. Por lo tanto, siendo el objetivo de esta fase la automatización y agilización del modelado y despliegue de infraestructura, el uso de AWS CloudFormation constituye un adecuado siguiente nivel de abstracción.

   Al proporcionar una forma declarativa de describir el total de una infraestructura deseada, AWS CloudFormation eleva la automatización de recursos individuales a bloques complejos de infraestructura. En otras palabras, con esta capa de abstracción, se pasa de automatizar el despliegue de recursos uno a uno a poder desplegar arquitecturas enteras (por ejemplo, una VPC con todos los recursos deseados dentro) de forma automática, repetible y eficaz.

   Sin embargo, pese a que AWS CloudFormation posibilita el despliegue de una arquitectura compleja con la utilización de un solo *stack*, desplegar distintos módulos de infraestructura en un mismo *stack* puede resultar perjudicial de cara a la escalabilidad, reutilización, control de cambios y gestión de la infraestructura. En otras palabras, desplegar grandes y complejos bloques de infraestructura utilizando un solo *stack* no es eficiente en el caso de querer modificar, escalar o reutilizar partes de dicha infraestructura.

Por lo tanto, en esta fase de despliegue se propone la utilización de distintos *stacks* de CloudFormation para el despliegue de los varios módulos que componen las infraestructuras deseadas.

3. La última capa de automatización de cara a lograr un despliegue completamente automático conlleva la utilización de dos herramientas: comandos del AWS CLI automatizados a través de *scripts* de Python. La utilización de Python como lenguaje de programación para la automatización de procesos debido a su legibilidad y facilidad de uso, su gran variedad de bibliotecas y módulos, y al conocimiento previo existente en el uso de este lenguaje.

   Al haber dividido el despliegue de la infraestructura en varios *stacks* de CloudFormation, se utilizan los scripts de Python para automatizar la ejecución de comandos del AWS CLI que despliegan cada uno de los *stacks*, además de gestionar las dependencias entre ellos. Por lo tanto, con esta última herramienta, se obtiene un sistema que despliega de forma completamente automática una infraestructura híbrida: con la simple ejecución del *script* de Python, se desplegaría todo.

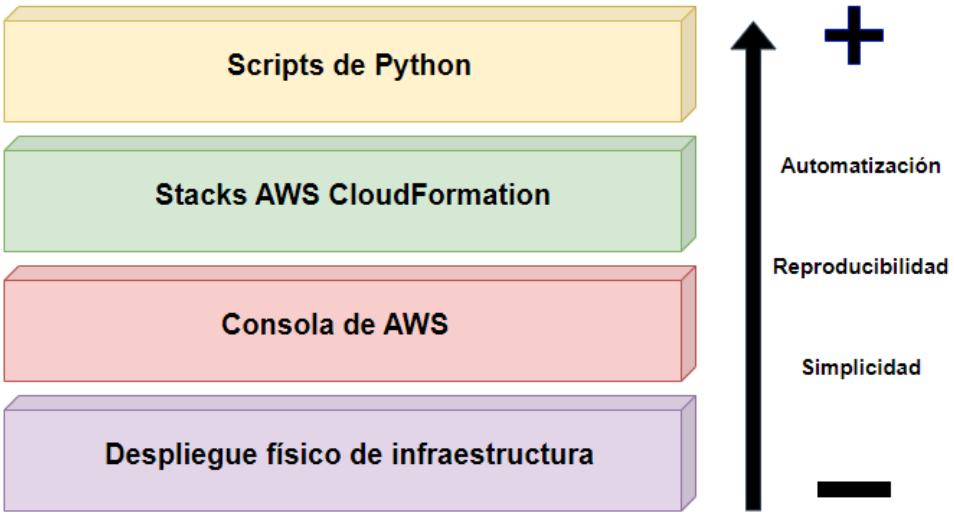

Figura 4.1: Diagrama de capas de abstracción

4.1.1. Arquitectura inicial *one-to-one*

Inicialmente, se propuso el despliegue de la arquitectura híbrida más sencilla posible, compuesta por las instalaciones locales, una VPC desplegada en el entorno de AWS y una conexión AWS Site-to-Site VPN entre ellas. El motivo de empezar con una arquitectura simple es empezar por familiarizarse con las herramientas para conseguir un despliegue completamente automático y una hibridación pura entre las instalaciones locales y la VPC de AWS.

Por lo tanto, un esquema de alto nivel de la arquitectura inicial desplegada en la figura 4.2.

Debido a la dificultad de simular un centro de datos corporativo físico, el entorno local u *on-premise* se simula mediante otra VPC. Al ser el objetivo de una VPC generar un

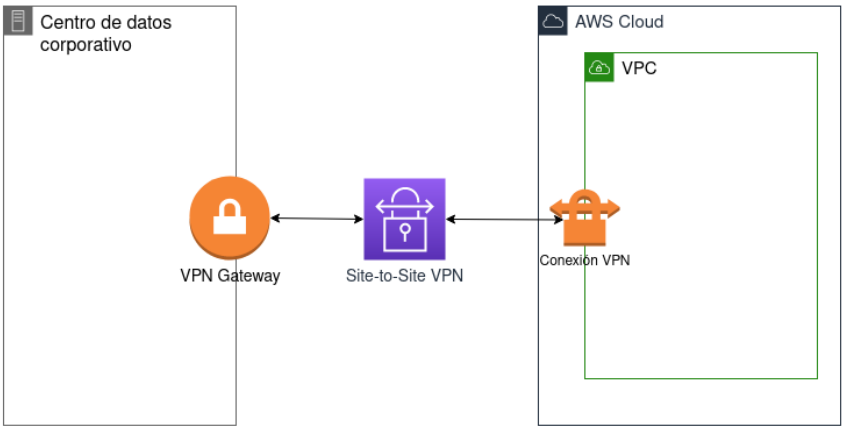

Figura 4.2: Esquema de alto nivel de arquitectura inicial

entorno completamente aislado dentro de la nube pública de AWS, el uso de una VPC para simular un entorno *on-premise* resulta idóneo: se utilizarás instancias EC2 desplegadas dentro de dicha VPC para simular servidores físicos locales, y las políticas de seguridad y reglas de acceso de la VPC para simular los *firewalls* que pudiese tener un centro corporativo real.

Por lo tanto, para desplegar esta infraestructura, se divide el despliegue en tres *stacks* de CloudFormation, que se pueden observar en la figura 4.3:

1. VPC-OnPremise: un *stack* para la VPC de simulación del entorno local y todos los recursos asociados a este entorno.
2. VPC-Cloud: un segundo *stack* para desplegar la verdadera VPC, que alojará los recursos mínimos que pueda necesitar una VPC para alojar una carga de trabajo cualquiera.
3. VPN-Server: Un último *stack* que desplegará una instancia EC2 dentro de la VPC de simulación del *on-premise* que actuará como servidor VPN contra el que se realizará la conexión Site-to-Site VPN

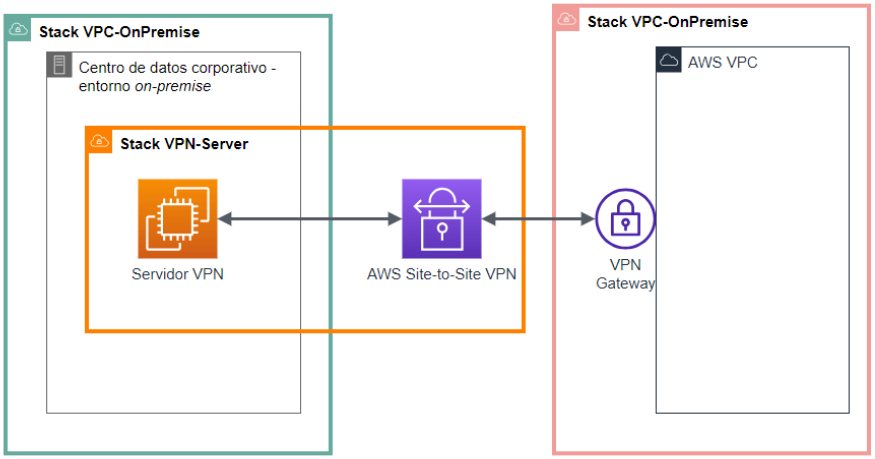

Figura 4.3: Arquitectura inicial según los *stacks* de CloudFormation que la conforman

4.1.1.1. Configuración de las VPC

Como se ha explicado en el Marco Teórico, una VPC puede albergar cualquier servicio de AWS, aislándolo del resto de nube pública. Por lo tanto, las posibilidades de configuración de una VPC son muy extensas, pudiendo adaptarlas a los requisitos específicos de cada momento del usuario. Dado que el objetivo de este sistema no es desplegar servicios en VPCs, sino desplegar de forma automática una arquitectura puramente híbrida, en cada VPC se desplegará una configuración básica provista por AWS en su GitHub[22].

A continuación se puede observar un esquema de dicha configuración de VPC:

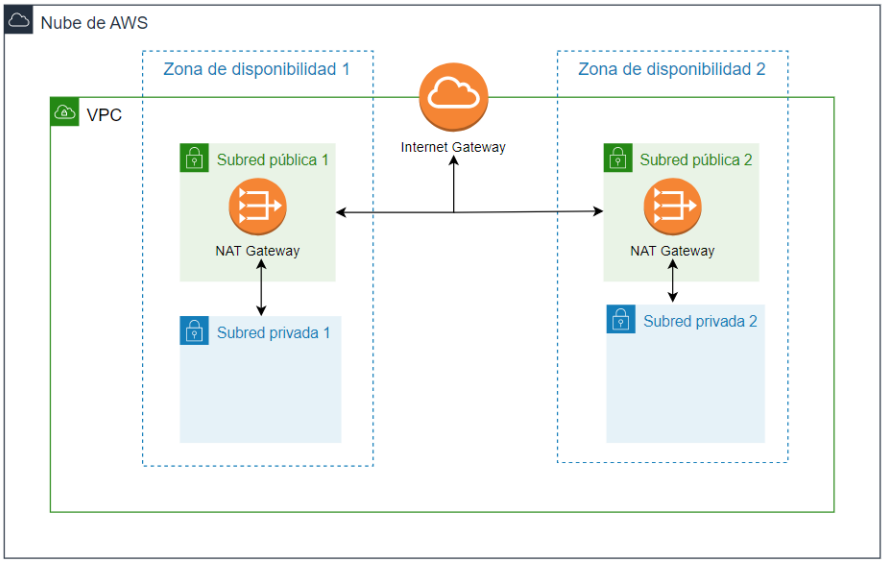

Figura 4.4: Modelo de VPC básico

En la figura 4.4 se puede observar la composición de esta VPC básica, formada por dos subredes, una pública y una privada. Estas dos subredes están redundadas en dos Zonas de Disponibilidad o Availability Zones siguiendo las guías de buenas prácticas de AWS[23] para aumentar la disponibilidad y la tolerancia a fallos de la VPC.

La subred pública tiene asociada una tabla de rutas que encamina el tráfico dirigido a Internet hacia la puerta de enlace de Internet o Internet Gateway. También cuenta con una Access Control List (ACL), un conjunto de reglas de seguridad que definen los permisos de acceso para el tráfico entrante y saliente a nivel de red. Este ACL permite el tráfico de protocolo HTTP entrante y saliente desde cualquier dirección IP, por lo que la subred pública está completamente abierta al tráfico con Internet.

La conectividad de la subred privada se hace a través de pasarelas NAT, o NAT gateways. Se trata de un servicio gestionado que permite que recursos en subredes privadas se conecten con servicios externos a la VPC, sin que el exterior pueda conectarse con ellos. El NAT Gateway se aloja en la subred pública, y cuando la subred privada encamina tráfico hacia el exterior desde su dirección IP privada, el NAT Gateway la traduce a una dirección IP pública. Esta IP pública se trata de un Elastic IP Address, o EIP, ofrecido por AWS, que es una dirección estática y pública que se puede asociar a recursos y que permanece constante después de reinicios o cambios en la infraestructura que la rodea[24].

La configuración de las dos VPCs desplegadas en este escenario se hará siguiendo esta plantilla como base, para luego alojar dentro los recursos necesarios para lograr la hibridación.

4.1.1.2. Hibridación mediante Site-to-Site VPN con Virtual Private Gateway

Una vez desplegadas las dos VPC, una simulando un entorno local en un centro de datos corporativo, y otra actuando como una VPC real, se procede a desplegar el túnel Site-to-Site VPN que las conectará de manera cifrada y segura, logrando la hibridación.

El primer paso para desplegar el túnel VPN se hará siguiendo el procedimiento más común en nubes híbridas, desplegando el túnel desde las instalaciones locales. Para hacer esto, una empresa puede utilizar varios métodos: dispositivos VPN dedicados, routers con soporte VPN, soluciones de software VPN comerciales, o la opción utilizada en este trabajo, software VPN de código abierto. Se ha seleccionado esta opción por su ágil y gratuito despliegue y por la extensa documentación que existe para las soluciones *open source*.

Existen gran cantidad de softwares VPN de código abierto, siendo la más popular OpenVPN. Sin embargo, se optó por la utilización de strongSwan, ya que ofrece mucha más compatibilidad con AWS Site-to-Site VPN. StrongSwan es una implementación de código abierto de los protocolos IKE que posibilita el tráfico IP seguro en escenarios IPsec principalmente para sistemas Linux, por lo que es muy compatible con AWS Site-to-Site VPN y con las AMI más comúnmente utilizadas en instancias de Amazon EC2, de Linux.

StrongSwan ofrece métodos sólidos de cifrado y autenticación con énfasis en la simplicidad de la configuración a través de un diseño modular que permite gran capacidad de extensión y portabilidad, por lo que resulta idóneo para el despliegue de arquitecturas híbridas que serán muy cambiantes dependiendo de las cargas de trabajo alojadas[25].

Para desplegar esta solución de VPN, se debe desplegar un servidor en el que se aloje el software de strongSwan, que actuará como VPN Gateway desplegando el túnel. Para el despliegue del software se sigue una plantilla de CloudFormation provista por AWS, que despliega dicha instancia con strongSwan configurado concretamente para AWS Site-to-Site VPN[26]. Dicha plantilla lanza una instancia EC2 ubicada en la VPC que se le especifica, y se le debe pasar como parámetros las configuraciones necesarias del lado de AWS para levantar el túnel.

Para obtener los valores de dichas configuraciones, se despliegan los recursos de AWS necesarios para conectarse al servidor VPN:

- Un AWS CustomerGateway, o CGW, que actúa como punto de conexión de red del entorno local y se encarga de encaminar el tráfico de red originado localmente hacia la VPC de AWS. Dicho recurso establece una interfaz para configurar y administrar la conexión VPN, especificando aquí los datos relevantes del servidor VPN. Aquí se debe establecer la dirección IP del servidor VPN (que será una dirección IP elástica desplegada previamente en la VPC-OnPremise que se importará). Además, se debe especificar el número de Sistema Autónomo, o ASN, que AWS utilizará para identificar los dispositivos involucrados en la conexión VPN en el intercambio de rutas BGP.

```
1 VPNCustomerGateway:
2   Type: AWS::EC2::CustomerGateway
```

```
  Properties:
    BgpAsn: 65000
    IpAddress: !ImportValue vpn-eip
    Type: ipsec.1
```

Programación 4.1: Código CloudFormation para despliegue de AWS CustomerGateway

- Un VGW, que actuará como punto de entrada y salida para el tráfico de red de la conexión VPN, especificando también su ASN y el protocolo que utilizará la conexión VPN. Al utilizar strongSwan como software VPN, se debe utilizar *ipsec.1*. Una vez desplegado este recurso, se debe asociar a la VPC en la que se ubicará, es decir, VPC-Cloud.

```
VPNGateway:
      Type: AWS::EC2::VPNGateway
      Properties:
        AmazonSideAsn: 64512
        Type: ipsec.1

  VPCGatewayAttachment:
      Type: AWS::EC2::VPCGatewayAttachment
      Properties:
        VpnGatewayId:
          Ref: VPNGateway
        VpcId:
          Ref: VPC-Cloud
```

Programación 4.2: Código CloudFormation para despliegue del VPG y del *attachment* a VPC-Cloud

- Una vez desplegados el CGW y el VGW, se debe desplegar un VPNConnection, que especifica la conexión VPN entre estos dos. Al desplegar este recurso, se especifica que CGW y VGW va a conectar, además de habilitar el protocolo BGP al poner el parámetro *StaticRoutesOnly* a *false*.

```
VPNConnection:
      Type: AWS::EC2::VPNConnection
      Properties:
        CustomerGatewayId:
          Ref: VPNCustomerGateway
        StaticRoutesOnly: false
        Type: ipsec.1
        VpnGatewayId:
          Ref: VPNGateway
```

Programación 4.3: Código CloudFormation para despliegue del VPNConnection

Una vez desplegados todos estos recursos, la VPC-Cloud tiene todos lo necesario para conectarse mediante un túnel a un servidor VPN. Del recurso VPNConnection se obtienen las configuraciones necesarias para el túnel VPN a través de un fichero de configuración. Se debe especificar el proveedor VPN o fabricante del dispositivo VPN local, y dado que se está utilizando el software *open source* strongSwan y no un dispositivo dedicado, se selecciona un proveedor genérico.

Cabe destacar que el software strongSwan despliega dos túneles VPN iguales por motivos de redundancia, para aumentar la disponibilidad y la tolerancia a fallos de la conexión VPN. De dicho fichero de configuración se extraen todos los parámetros necesarios para desplegar el servidor VPN en la VPC-OnPremise y los dos túneles VPN que se conectarán al VGW de la VPC-Cloud:

- Se puede seleccionar que método de autenticación utilizar, a través de PSK o autenticación basada en certificados. Debido a la simpleza de la conexión (solo una conexión a cada extremo) se utiliza autenticación basada en PSK. Dentro del fichero de configuración se pueden obtener los valores de las PSK de ambos túneles, que se deberán almacenar en AWS SecretsManager (el servicio de AWS utilizado para almacenar y gestionar secretos) en formato clave-valor, para luego pasarle la clave de los secretos como parámetro a la plantilla de CloudFormation del servidor VPN.
- Las direcciones IP de cada extremo de ambos túneles:
  - Las direcciones IP externas del CGW y del VGW, es decir, las direcciones IP públicas utilizadas para establecer la conexión entre el CGW y el VGW que se utilizarán para el intercambio de tráfico cifrado a través de la conexión VPN.
  - Las direcciones IP internas del CGW y del VGW, es decir, las direcciones IP privadas que han sido asignadas a la interfaz del túnel VPN y se utilizarán para enrutar el tráfico interno de la conexión VPN.
- Por último se obtienen los valores necesarios para la configuración del protocolo BGP: el ASN del VGW y la dirección IP del vecino BGP, es decir, la dirección IP del siguiente salto que se utilizará para establecer el intercambio de rutas con BGP, que en este caso será una dirección IP interna del VGW.

Además de los datos obtenidos del fichero de configuración para la conexión VPN, hay otros parámetros que se deben pasar como parámetro al *stack* del servidor VPN para su correcto despliegue:

- El ID de la VPC en la que se desplegará el servidor VPN, es decir, el ID de VPC-OnPremise.
- El bloque CIDR de la VPC en la que se desplegará el servidor VPN.
- El ID de la subred donde se desplegará el servidor VPN.
- La EIP reservada en la VPC-OnPremise que actuará como dirección IP pública de la instancia EC2 que actuará como VPN Gateway.
- Por último, se debe seleccionar que tipo de instancia y con que AMI se va a desplegar la instancia de EC2 que actuará como servidor VPN. Por la compatibilidad de strongSwan con Linux, se selecciona la última AMI disponible de Amazon Linux, *amzn2-ami-hvm-x86-64-ebs*. Además, debido a que se trata de una simulación y el servidor VPN tendrá que soportar un volumen de tráfico muy bajo, se selecciona un tipo de instancia EC2 barata y de bajo rendimiento: *t3a.micro*.

Una vez obtenidos todos los parámetros necesarios para desplegar el *stack* de CloudFormation del servidor VPN, se puede desplegar dicho *stack*. Según se ha especificado, esto despliega una instancia EC2 en la subred pública de la VPC-OnPremise, que actúa como servidor VPN. Entre este servidor y el VGW ubicado en la VPC-Cloud el software de strongSwan despliega los dos túneles VPN y realiza el intercambio de rutas BGP, lo que

permite una hibridación pura a nivel IP. El intercambio de rutas BGP permite que servidores ubicados en la VPC-OnPremise puedan ver las IP privadas de servidores ubicados en la VPC-Cloud y viceversa, lo que significa que, unido al cifrado y seguridad que ofrecen los túneles VPN, ambos entornos se pueden ver entre ellos como si fuesen uno sólo.

En la figura 4.5 se puede observar la arquitectura desplegada. Gracias a la hibridación, se pueden alojar cargas de trabajo localmente y en la nube y estas se podrán comunicar entre ellas como si formasen parte de un mismo entorno, dando lugar a una hibridación completa.

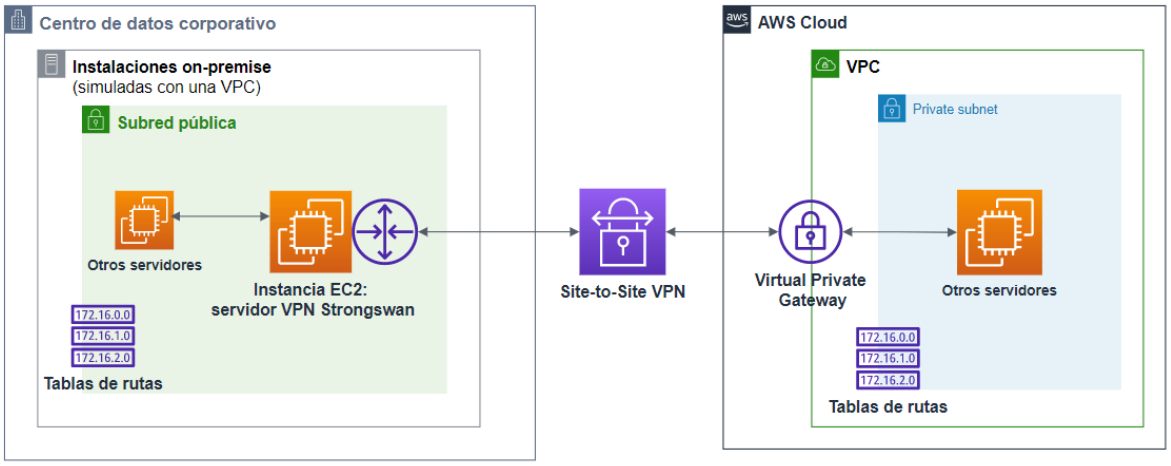

Figura 4.5: Arquitectura de despliegue inicial

4.1.1.3. Automatización del despliegue

En los apartados anteriores se han descrito los pasos necesarios para desplegar las dos VPC que componen esta arquitectura y el túnel VPN entre ellas que permite la hibridación del entorno. Para lograr el despliegue automático de dicha arquitectura se aplican las tres capas de abstracción explicadas previamente.

Primero se comprueba que la arquitectura cumple los requisitos establecidos desplegándola de forma manual a través de la consola de AWS. Este proceso además de como prueba, sirve como manual de guía para establecer todos los pasos que posteriormente se van a automatizar.

Una vez probado el despliegue manual, se configuran *stacks* de CloudFormation que despliegan de forma automática la VPC-OnPremise, la VPC-Cloud y el servidor VPN. Estos *stacks* permiten el despliegue automático de módulos predefinidos de la arquitectura, en este caso la VPC de simulación del entorno local, la VPC real y el servidor EC2 que alojará el software VPN strongSwan que establecerá la conexión VPN. La organización de los tres *stacks* se puede observar en la figura 4.3.

Sin embargo, estos *stacks* de CloudFormation no son suficiente para lograr la automatización deseada, ya que son dependientes de parámetros de otros *stacks*. El CGW desplegado en la VPC-Cloud es dependiente de recursos creados en las VPC-OnPremise, y el servidor VPN depende de los múltiples parámetros extraídos del fichero de configuración VPN para poder establecer el túnel VPN y el intercambio de rutas BGP correctamente. Además, la existencia de 3 *stacks* distintos también conlleva la necesidad de desplegarlos manualmente desde la consola de CloudFormation en el orden correcto.

Por lo tanto, para gestionar las dependencias de recursos y lograr el despliegue automático de la arquitectura, se utiliza un *script* de Python. Dicho *script* ejecuta de forma automática los comandos del AWS CLI necesarios para:

1. Desplegar el *stack* VPC-OnPremise.
2. Una vez terminada de crearse la VPC-OnPremise, despliega el *stack* VPC-Cloud.
3. Obtiene el fichero de configuración de la conexión VPN asociado a la VPC-Cloud para un proveedor genérico. Extrae los parámetros necesarios de dicho fichero para el establecimiento del túnel VPN y sube las PSK de ambos túneles a AWS Secrets Manager.
4. Una vez se obtienen todos los parámetros necesarios, despliega el *stack* VPN-Server. Así, se despliegan los dos túneles VPN.
5. Una vez están todos los recursos desplegados, añade a las tablas de rutas de cada VPC las rutas necesarias para encaminar tráfico de la VPC-OnPremise a la VPC-Cloud y viceversa. Una vez están establecidas dichas rutas, se establece el intercambio de rutas BGP y se obtiene la hibridación pura de ambos entornos.

A continuación se puede observar un pseudocódigo que resume el *script* de Python de automatización del despliegue.

```
#Se despliega primero la VPC-OnPremise
command = "aws cloudformation create-stack --stack-name VPC-OnPremise --template-body file://VPCsimulacionOnPremise.yml"
subprocess.run(command, shell=True, check=True)

#Se despliega la VPC-Cloud
command = "aws cloudformation create-stack --stack-name VPC-Cloud --template-body file://VPCnube.yml"
subprocess.run(command, shell=True, check=True)

#Obtiene el fichero de configuracion de la conexion VPN para el vendor y la VPNConnection dados
Funcion saca_fichero_config(vpn_conexion_id, dispositivo_vpn, ike_version, aws_region):
    Intenta hacer lo siguiente:
        Ejecutar el comando "aws ec2 get-vpn-connection-device-sample-configuration" con los siguientes parametros:
            --vpn_conexion_id: ID de la conexion VPN
            --dispositivo_vpn: Tipo de dispositivo de conexion VPN (en este caso, Generico)
            --internet-key-exchange-version: Version del intercambio de claves de Internet (IKE)
            --region: Region de AWS
            --output: Formato de salida (en este caso, 'text')
fichero_config = saca_fichero_config(vpn_conexion_id, vpn_connection_device_type_id, ike_version, aws_region)

#Extrae los parametros necesarios del fichero de configuracion y los guarda en un JSON
Funcion extraer_parametros(fichero_config):
```

```
    Abrir el fichero de configuracion "fichero_config"

    Leer el contenido del fichero y cargarlo en una variable
       llamada "configuracion"

    Crear una lista vacia llamada "parametros"

    Para cada parametro en la lista de configuracion:
        Crear un nuevo objeto de parametro con las siguientes
           propiedades:
            "ParameterKey": Clave del parametro
            "ParameterValue": Valor del parametro

        Agregar el objeto de parametro a la lista JSON
        "parametros"

    Devolver la lista de "parametros"

# Uso de la funcion
lista_parametros = extraer_parametros("fichero_config")

#Despliega el stack de la conexion VPN con los parametros
   obtenidos
command = "aws cloudformation create-stack --stack-name VPN-
   Server --template-body file://vpn-gateway-strongswan.yml --
   parameters file://lista_parametros.json"
subprocess.run(command, shell=True, check=True)
```

Programación 4.4: Pseudocódigo del *script* de Python de automatización del despliegue

Así, se obtiene un *script* capaz de desplegar dicha arquitectura de forma completamente automática, siendo solamente necesario ejecutar el *script* desde un terminal.

4.1.2. Arquitectura *one-to-many*

Una vez comprobado el correcto funcionamiento del despliegue automático de la arquitectura inicial, se procede al desarrollo del despliegue automático de una arquitectura híbrida más compleja y cercana a lo que una empresa real precisa.

La arquitectura inicial *one-to-one*, pese a su hibridación pura, presenta ciertas limitaciones en cuanto a rendimiento y logística: no es recomendable albergar todos los servicios de una empresa en una sola VPC, y de querer desplegar múltiples VPC, el sistema previo precisa del despliegue de un túnel VPN para conectarse con cada una. Esto representa un problema para una empresa que necesite alojar cargas de trabajo en distintas VPC por motivos organizativos (distintos departamentos, proyectos o productos) o por motivos operativos (separación de entornos de desarrollo, pruebas, pre-producción y producción o separación de cargas de trabajo por motivos de seguridad), ya que de querer desplegar múltiples VPC sería necesario establecer y mantener múltiples conexiones VPN, lo que supone una considerable complejidad de despliegue y gestión, una escalabilidad limitada y unos mayores costes.

Para resolver estas limitaciones se propone el despliegue automático de una arquitectura

one-to-many, en la que el entorno local se conecta mediante un sólo túnel VPN a tantas VPC como necesite desplegar el usuario. En esta arquitectura se volverá a utilizar una VPC aislada para simular en entorno *on-premise* de una empresa.

El elemento clave de esta nueva arquitectura es el Transit Gateway, que actuará a la vez como punto de entrada y salida de tráfico del túnel VPN hacia la nube de AWS y como enrutador central de todo el tráfico de la red. Es decir, el Transit Gateway actuará como eje central de toda el tráfico de la red, desencriptando el tráfico proveniente del entorno local y enrutándolo a la VPC destino, encriptando y transmitiendo por el túnel VPN tráfico dirigido al entorno local, o simplemente enrutando el tráfico entre VPCs.

Así, se muestra en la figura 4.6 un diagrama a alto nivel de la arquitectura a desplegar.

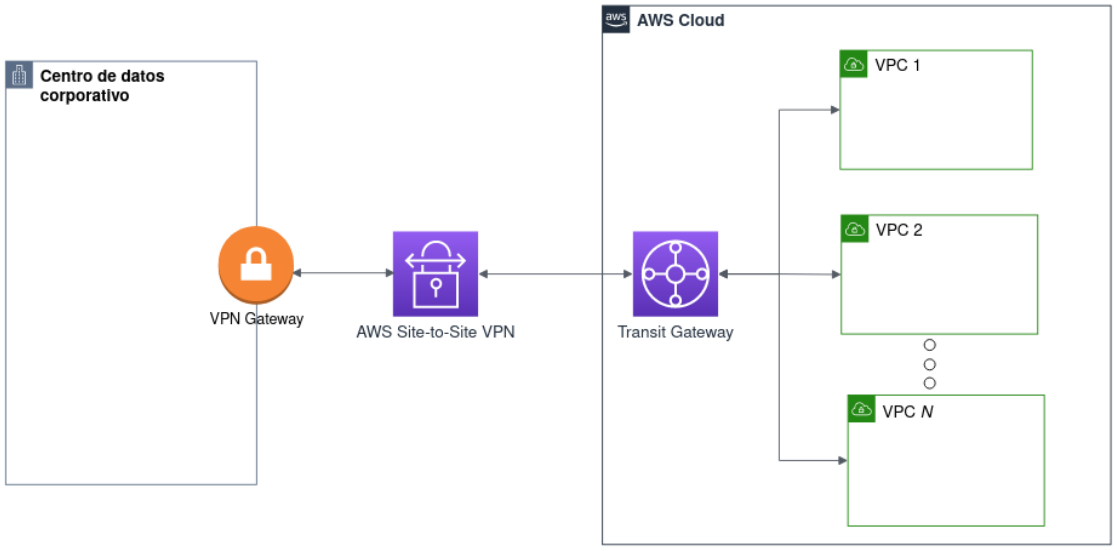

Figura 4.6: Esquema de alto nivel de arquitectura *one-to-many*

Por lo tanto, la conformación de esta arquitectura se divide en cuatro módulos distintos, a partir de los cuales se desarrollan los *stacks* de CloudFormation:

- VPC-OnPremise: la simulación del entorno *on-premise*, formada por una VPC y los recursos mínimos necesarios para simular un entorno local real.
- VPC-Cloud: la mínima VPC que desplegará el sistema, en el caso de que solo se desee desplegar una. Este módulo se aprovecha para desplegar el Transit Gateway y todos los recursos necesarios para interconectarlo con el entorno local a través de la conexión VPN y con las posibles demás VPC.
- VPN-Server: el servidor VPN, que se desplegará en la VPC-OnPremise y establecerá la conexión Site-to-Site VPN entre el entorno local y las demás VPC conectadas al Transit Gateway.
- VPC-ExtraCloud: este módulo desplegará todas las VPC adicionales que se deseen desplegar y las interconectará con el Transit Gateway.

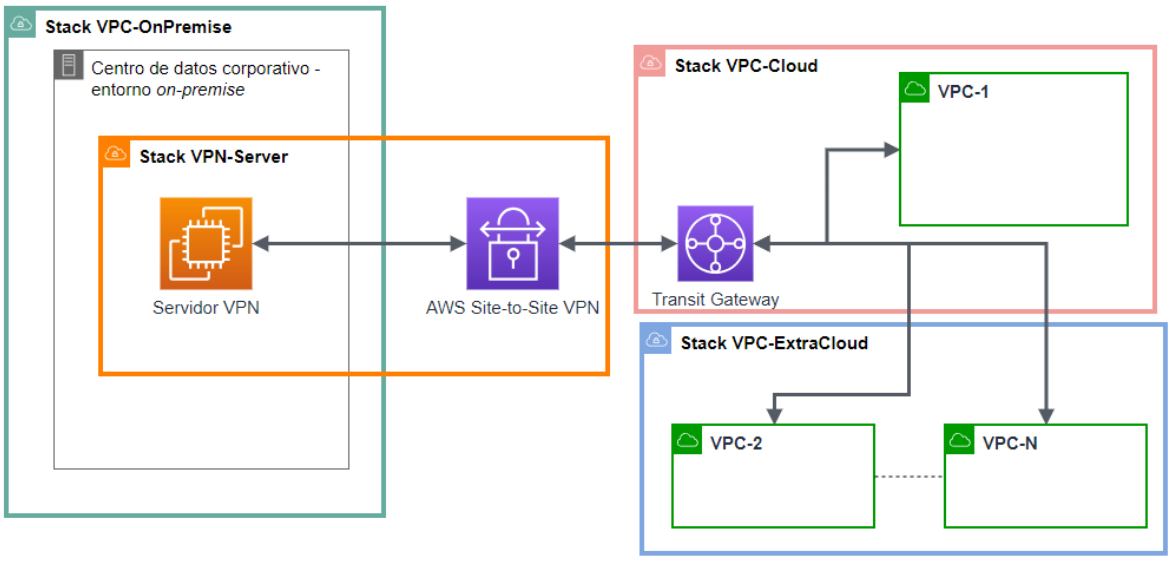

Figura 4.7: Arquitectura *one-to-many* según los *stacks* que la conforman

4.1.2.1. Configuración de las VPC

Como se ha explicado previamente, esta arquitectura está formada por tres tipos de VPC distintas: la VPC de simulación de un entorno local, la VPC que actuará como VPC mínima y que desplegará el Transit Gateway, y las demás VPC que se desee desplegar.

La VPC que simulará el entorno *on-premise*,VPC-OnPremise, estará conformada exactamente igual que en la arquitectura inicial del apartado 4.1.1, por lo que seguirá el modelo de VPC básico mostrado en la figura 4.4. Esto se debe a que al simular un entorno local, no es necesario añadir más recursos que los definidos previamente, sin importar cómo está conformada la arquitectura en la nube.

En cuanto a la VPC-Cloud, estará conformada de forma similar a la VPC-Cloud de la arquitectura inicial. Sin embargo, se aprovechará el despliegue de esta VPC para desplegar el Transit Gateway, ya que al desplegar esta arquitectura como mínimo se desplegará una VPC. Es importante señalar que aunque el Transit Gateway se despliegue dentro de la plantilla de CloudFormation de esta VPC, no estará ubicado dentro de ella. El Transit Gateway actuará como un elemento independiente de cualquier VPC, para desempeñar correctamente su labor de enrutador central de toda la red.

Por lo tanto, VPC-Cloud desplegará por un lado el Transit Gateway, y por otro lado la VPC y los recursos mínimos necesarios dentro. Dichos recursos serán iguales a los mostrados en la figura 4.4, ya que conforman la VPC más básica posible. Además de dichos recursos, se despliega un Transit Gateway Attachment, el recurso que actúa como enlace entre una VPC y un Transit Gateway.

Finalmente, en el caso de que se desee desplegar más de una VPC, se configuran las VPCs adicionales a través del módulo VPC-ExtraCloud. Estas VPC están diseñadas igual que la VPC-Cloud, con los recursos necesarios para conformar una VPC básica funcional mostrados en la figura 4.4. Además, se desplegará en cada VPC adicional un Transit Gateway Attachment, permitiendo la interconexión de cada VPC con el entorno local y con las demás VPC.

4.1.2.2. Hibridación mediante Site-to-Site VPN con Transit Gateway

Gracias a que el Transit Gateway actúa como punto único de interconexión entre todas las VPC y como *endpoint* de la conexión VPN con el entorno local, solo es necesario establecer un túnel VPN para obtener la hibridación entre el entorno local y todas las VPC desplegadas. Por lo tanto, el establecimiento del Site-to-Site VPN se hace de forma muy similar al realizado en la arquitectura inicial.

Observado el buen rendimiento, simpleza de despliegue y teniendo en cuenta los medios disponibles, se vuelve a utilizar el software VPN de código abierto strongSwan. Dicho software se despliega en una instancia EC2 que actúa como VPN Gateway en las instalaciones locales (simuladas mediante una VPC). Dicho servidor VPN establecerá los túneles VPN entre él mismo y el *endpoint* que se le especifique, que en este caso será el Transit Gateway. Por lo tanto, se reutiliza el mismo *stack* de CloudFormation provisto por AWS en su GitHub, que lanza una instancia EC2 con el software de strongSwan configurado concretamente para conexiones Site-to-Site VPN.

Antes de desplegar dicho *stack*, se despliegan los recursos necesarios para establecer la conexión VPN y obtener los parámetros necesarios para desplegar el *stack*:

- El CGW, que actuará como interpretación de AWS del punto de conexión de red del lado local de la conexión VPN. Se establece la dirección IP del servidor VPN, que será la EIP que actuará como dirección IP pública del servidor VPN y el ASN necesario para el intercambio de rutas BGP.

```
VPNCustomerGateway:
      Type: AWS::EC2::CustomerGateway
      Properties:
        BgpAsn: 65000
        IpAddress: !ImportValue vpn-eip
        Type: ipsec.1
```

Programación 4.5: Código CloudFormation para despliegue del Customer Gateway

- El Transit Gateway ya desplegado actuará como punto de entrada y salida para el tráfico de red de la conexión VPN, estableciendo *ipsec.1* como protocolo de seguridad de la conexión VPN.

```
TransitGateway:
    Type: AWS::EC2::TransitGateway
    Properties:
      AmazonSideAsn: 64512
      MulticastSupport: enable
      Description: VPC transit gateway
      AutoAcceptSharedAttachments: enable
      DefaultRouteTableAssociation: enable
      DefaultRouteTablePropagation: enable
      DnsSupport: enable
      VpnEcmpSupport: enable

TransitGatewayAttachment:
    Type: AWS::EC2::TransitGatewayAttachment
    Properties:
      SubnetIds:
```

```
17          - !Ref PublicSubnet0
18          - !Ref PrivateSubnet0
19        TransitGatewayId: !Ref TransitGateway
20        VpcId: !Ref VPC
```

Programación 4.6: Código CloudFormation para despliegue del Transit Gateway y de un Transit Gateway Attachment

- Con el CGW y el Transit Gateway desplegados, se despliega la conexión VPN entre ellos a través del recurso VPNConnection, habilitando el intercambio de rutas BGP al poner el parámetro *StaticRoutesOnly* a *false*.

```
1 VPNConnection:
2       Type: AWS::EC2::VPNConnection
3       Properties:
4         CustomerGatewayId:
5           Ref: VPNCustomerGateway
6         StaticRoutesOnly: false
7         Type: ipsec.1
8         TransitGatewayId:
9           Ref: TransitGateway
```

Programación 4.7: Código CloudFormation para despliegue del VPNConnection

Con estos recursos desplegados, se debe extraer el fichero de configuración para un proveedor VPN genérico del que se obtendrán todos los parámetros necesarios para desplegar el software strongSwan en un servidor EC2. De dicho fichero se extraen prácticamente los mismos parámetros que los extraídos en la arquitectura inicial, por lo que no se vuelven a detallar. Sin embargo, hay algunos parámetros que si es interesante explicar:

- Como método de autenticación se vuelve a seleccionar PSK. Dado que toda esta arquitectura está diseñada para solo tener que establecer un único túnel VPN, se elige la simplicidad y fácil de implementación de PSK, ya que su delicada gestión de claves y la falta de escalabilidad no serán un problema.
- En cuanto al tipo de instancia EC2 que se va a desplegar, dado que esta arquitectura cuenta con múltiples VPC, de aplicarse en un entorno de trabajo real tendrá que soportar mayores volúmenes de tráfico. Se recomienda utilizar una instancia de tipo *m5*, que ofrecen equilibrio entre capacidad de computación, recursos de memoria, rendimiento de red y precio. Dado que este servidor debe estar en constante funcionamiento y el número de conexiones y caudal de las conexiones es variable, este equilibrio es idóneo. Sin embargo, dado que en este proyecto el volumen de tráfico va a ser únicamente el generado para pruebas, se sigue utilizando una instancia de bajo coste de tipo *t3a.micro*.

Una vez obtenidos estos parámetros, se utilizan en el despliegue del servidor VPN para establecer el túnel entre el lado local y el Transit Gateway. El servidor se despliega en la subred pública de la VPC-OnPremise, estableciendo la conexión VPN redundada a través de dos túneles VPN cifrados con IPsec. Al habilitar el protocolo BGP en el Transit Gateway, se realizará el intercambio de rutas BGP en el momento del establecimiento de la conexión VPN, lo que permitirá al entorno local resolver las direcciones IP privadas de las VPC, y viceversa.

Al poder cada entorno resolver las direcciones IP privadas del otro a través de una conexión VPN segura, cargas de trabajo alojadas en un entorno podrán comunicarse con el otro extremo de forma segura y eficiente, por lo que se obtiene la hibridación completa buscada. En la figura 4.8 se muestra la arquitectura *one-to-many* desplegada, con la comunicación entre cargas de trabajo alojadas localmente y en cualquiera de las VPC desplegadas.

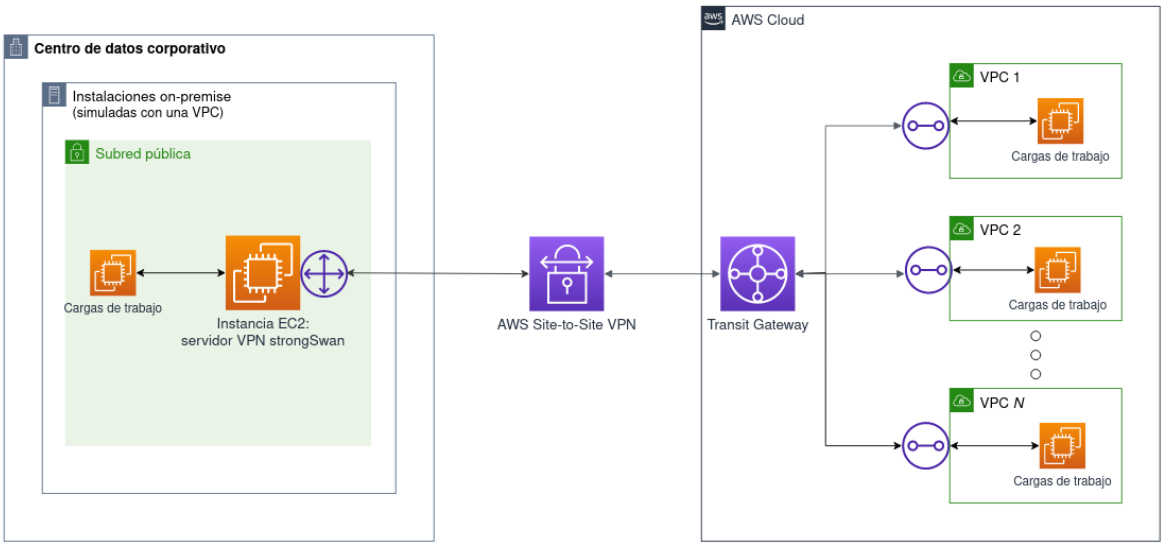

Figura 4.8: Arquitectura *one-to-many* con Transit Gateway

4.1.2.3. Automatización del despliegue

La automatización de esta arquitectura se realiza de manera muy similar a la realizada en la arquitectura inicial en el apartado 4.1.1.3. Los *stacks* de CloudFormation se conforman como se indica en la figura 4.7, con la particularidad de que el *stack* VPC-ExtraCloud se puede desplegar tantas veces como VPCs se quieran, proporcionando mucha más flexibilidad a la arquitectura.

Por lo tanto, el *script* de automatización del despliegue de los *stacks* actuará de forma muy similar al desarrollado para la arquitectura inicial, siguiendo los mismos pasos que los mostrados en el pseudocódigo de dicho apartado para desplegar la VPC-OnPremise primero, la VPC-Cloud y el VPN-Server. Sin embargo, a partir de este punto el *script* funciona de forma distinta.

Dado que la intención de esta arquitectura es lograr una mayor flexibilidad y adaptabilidad a las necesidades del usuario, este script permite al usuario que en el momento de ejecutar el comando de despliegue, se pase como parámetro el número de VPCs a desplegar. Así, de ejecutar el comando sin pasar el número de VPCs deseado, se desplegará por defecto una sola VPC conectada al Transit Gateway (la que despliega el *stack* VPC-Cloud). Sin embargo, si en la ejecución se indica el despliegue de, por ejemplo, 4 VPCs, el *script* se encarga de desplegar tres instancias del *stack* VPC-ExtraCloud.

Con motivo de obtener la máxima simplicidad posible a la hora de ejecutar el *script*, el número de VPCs deseado simplemente se debe indicar en el argumento del comando de ejecución del *script* en el terminal. A continuación se muestra el código necesario para desplegar el número adicional de VPCs deseado, que se añade al final del pseudocódigo mostrado en el apartado de la arquitectura inicial.

```python
#Genera un bloque CIDR para cada VPC conectada al Transit Gateway
def generate_cidr_blocks(numero):
    cidr_blocks = []
    public_subnet_cidr = []
    private_subnet_cidr = []

    for i in range(numero):
        cidr_block = f"192.168.{i * 16}.0/20"
        cidr_blocks.append(cidr_block)

        network = ipaddress.ip_network(cidr_block)
        subnets = list(network.subnets(prefixlen_diff=1))
        private_subnet_cidr.append(str(subnets[0]))
        public_subnet_cidr.append(str(subnets[1]))

    return cidr_blocks, private_subnet_cidr, public_subnet_cidr

#Despliega tantas VPC extras conectadas al Transit gateway como se hayan introducido en el comando de ejecucion
def deploy_vpc_stacks(num_vpcs: int):

    cidr_blocks, private_subnet_cidrs, public_subnet_cidrs = generate_cidr_blocks(num_vpcs)

    for i in range(num_vpcs - 1):
        stack_name_extra = f"{nombre_stack_nube}{i+2}"
        cidr = cidr_blocks[i]
        private_subnet_cidr = private_subnet_cidrs[i]
        public_subnet_cidr = public_subnet_cidrs[i]

        cmd = f"aws cloudformation create-stack --stack-name stack_name_extra --template-body file://VPCnubeExtra.yml --parameters ParameterKey=VpcCIDR,ParameterValue=cidr ParameterKey=PublicSubnetCIDR,ParameterValue=public_subnet_cidr ParameterKey=PrivateSubnetCIDR,ParameterValue=private_subnet_cidr"
        subprocess.run(cmd, shell=True, check=True)

        wait_command = f"aws cloudformation wait stack-create-complete --stack-name {stack_name_extra}"
        subprocess.run(wait_command, shell=True, check=True)

        print(f"Exito desplegando {stack_name_extra}")

if __name__ == "__main__":
    try:
        num_vpcs = int(sys.argv[1])
    except ValueError:
        print("Error: el argumento debe ser un numero entero.")
    else:
        deploy_vpc_stacks(num_vpcs)
```

Programación 4.8: Script de Python de automatización del despliegue de N VPCs

Por lo tanto, el *script* de Python es capaz de desplegar una arquitectura completamente híbrida compuesta por tantas VPCs como se le especifique en el comando de ejecución, que se desplegarán conectadas con un Transit Gateway que a su vez se conecta con el entorno local mediante un túnel VPN.

4.2. Fase de gestión

Una vez se obtiene un sistema capaz de desplegar arquitecturas completamente híbridas de forma automática, se procede a desarrollar elementos en dichas arquitecturas que simplifiquen y agilicen su gestión. Dado que un entorno híbrido puede ser realmente complejo, toda arquitectura híbrida debe contar con herramientas de gestión centralizadas capaces de controlar todos los recursos dispersos.

La implementación de herramientas de gestión centralizadas pueden traer consigo gran cantidad de ventajas para arquitecturas híbridas, como una mayor eficiencia operativa resultado de la automatización de procesos y la consecuente reducción de carga de trabajo manual, una mas rápida y efectiva escalabilidad de los recursos, y una mayor seguridad, resiliencia y tolerancia a fallos.

Sin embargo, la cantidad de posibilidades en cuanto a herramientas de gestión para entornos híbridos es enorme, existiendo todo tipo de posibles soluciones a los problemas administrativos que pueda tener una organización en concreto. Por lo tanto, en este trabajo se va a desarrollar una herramienta de gestión que resuelve un problema de las arquitecturas desplegadas previamente y que agiliza la interconexión y reduce la carga administrativa de dichas arquitecturas.

4.2.1. Resolución automática y centralizada de DNS

Para esta segunda fase, se va a partir de la segunda arquitectura planteada en la fase de despliegue debido a su mayor versatilidad, escalabilidad y flexibilidad. Sin embargo, pese a que todos los entornos de dicha arquitectura si que están plenamente interconectados a nivel 3, la comunicación entre servicios se debe hacer a través de direcciones IP. Esto plantea problemas de operabilidad por el simple hecho de que si una máquina local desea conectarse a un servidor alojado en una VPC en la nube, debe consultar su dirección IP privada y conectarse.

Las direcciones IP, al ser identificadores numéricos, pueden ser difíciles de recordar y administrar cuando se tratan muchos recursos, lo que reduce la escalabilidad y flexibilidad de la infraestructura de red. Además, también puede resultar en problemas de mantenimiento y resolución de problemas, como por ejemplo, de necesitar migrar un servicio de un servidor a otro, haría falta actualizar manualmente todas las referencias a la antigua dirección IP del servicio.

La utilización de nombres de dominio DNS resuelve dichos problemas y ofrece una serie de ventajas para la administración de una infraestructura de red:

- Legibilidad y facilidad de uso: los nombres de dominio son más fáciles e intuitivos de utilizar para cualquier usuario que una dirección IP, al ser mucho más fácil

entender y recordar un registro del tipo *"www.ejemplo.com"* que una dirección del tipo "192.168.0.1".

- Gestión centralizada: utilizar un DNS centralizado para toda la infraestructura de red permite administrar todos los registros en un sólo lugar, facilitando la configuración y mantenimiento de la red.

- Flexibilidad y escalabilidad: dado que DNS permite la asignación de nombres de dominio a diferentes direcciones IP, es posible agregar, escalar o cambiar recursos sin afectar directamente a los nombres de dominio. Esto facilita en gran medida la gestión al evitar la necesidad de actualizar manualmente todas las referencias a direcciones IP en caso de cambios.

Por lo tanto, esta fase del proyecto se enfoca en el desarrollo de una herramienta de gestión automática y centralizada de DNS que se añadirá a la arquitectura *one-to-many* desplegada previamente en el apartado 4.1.2.

4.2.1.1. Resolución de DNS mediante Route 53 Resolver

Para lograr dicha resolución automática y centralizada de DNS en una arquitectura híbrida se utilizará el servicio de AWS, Route 53 Resolver. Dicho servicio resulta idóneo para esta fase del proyecto, al ofrecer capacidades avanzadas para la resolución de DNS en entornos híbridos complejos. Además, este servicio es compatible con AWS Transit Gateway, por lo que dicha sinergia permite una fácil integración con la arquitectura *one-to-many* desarrollada en el apartado 4.1.2.

La integración de Route 53 Resolver con dicha arquitectura se puede realizar de diversas formas. Dada la arquitectura de la que se parte, representada en la figura 4.8, resulta lógico aprovechar la posibilidad que ofrece el Transit Gateway para interconectar múltiples VPCs, por lo que se plantea la utilización de una VPC aparte conectada al Transit Gateway cuya única finalidad sea alojar el Route 53 Resolver. Aunque este servicio se podría alojar en cualquier VPC ya desplegada, se considera beneficioso para la infraestructura global la utilización de una nueva VPC, ya que ofrece mayor aislamiento, simplicidad, resiliencia y seguridad el alojar una herramienta de gestión en una VPC cuyo único propósito sea la gestión en vez de en una VPC que pueda estar albergando otros servicios.

Así, se añadirá una nueva VPC llamada VPC-DNS conectada al Transit Gateway, cuya función será centralizar la resolución de DNS de toda la infraestructura de forma automática. Por lo tanto, la arquitectura resultaría como se muestra en la figura 4.9.

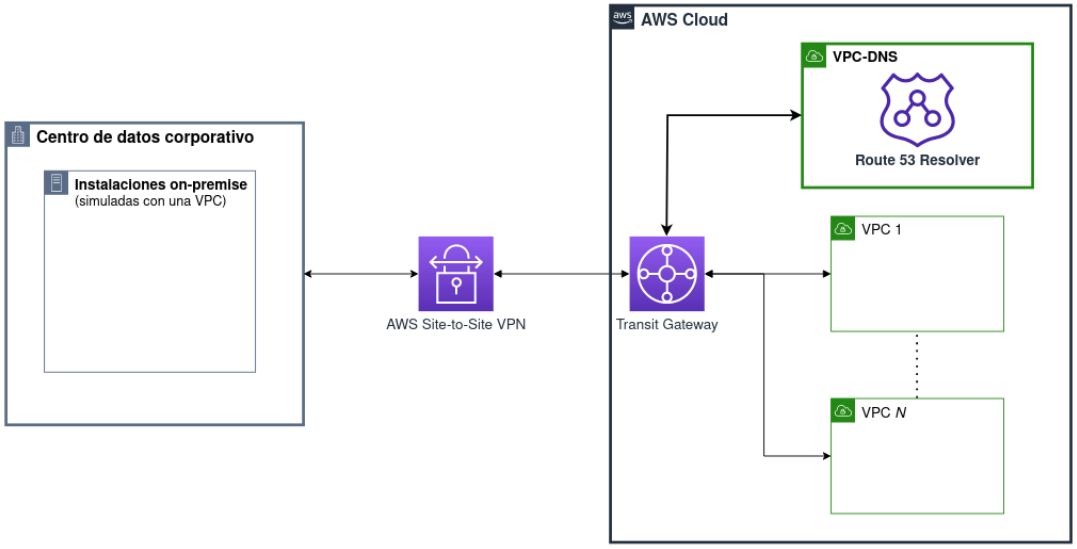

Figura 4.9: Arquitectura *one-to-many* con Route 53 Resolver

4.2.1.2. Configuración de la VPC-DNS

Como se ha explicado previamente, la herramienta de gestión de DNS desarrollada en esta fase se desplegará en una única VPC aislada, que se llamará VPC-DNS. Esta VPC estará formada inicialmente por el modelo de VPC básico mostrado en la figura 4.4, que la dotará de los recursos mínimos para tener conectividad. A continuación, se añaden los elementos de Route 53 Resolver necesarios para configurar la resolución de DNS.

El primer paso para configurar Route 53 Resolver es desplegar dos recursos llamados Resolver Endpoints, que son dos puntos finales accesibles a través de una dirección IP a través de los cuales se pueden dirigir consultas de DNS desde diferentes fuentes. Por lo tanto, cada Endpoint resolverá un tipo de consulta DNS:

- El Endpoint para consultas entrantes, o Inbound Resolver Endpoint, permite al servidor DNS local resolver los nombres de dominio de recursos alojados en AWS. Se establecen dos direcciones IP, una en la subred privada y una en la subred pública, que se reservarán en la VPC-DNS y a las que se dirigirán las consultas DNS originadas en el entorno local.

```
InboundDnsResolver:
    Type: AWS::Route53Resolver::ResolverEndpoint
    Properties:
      Direction: INBOUND
      IpAddresses:
        - SubnetId:
            Ref: PublicSubnet
          Ip: 10.1.0.7
        - SubnetId:
            Ref: PrivateSubnet
          Ip: 10.1.2.7
```

Programación 4.9: Código CloudFormation para despliegue del Inbound Resolver Endpoint

- El endpoint para consultas salientes, o Outbound Resolver Endpoint, que permite encaminar las consultas DNS originadas en AWS hacia el servidor DNS local. Se deben indicar donde se van a originar las consultas DNS, y dado que se pueden originar en varios puntos de la infraestructura, se establece que se pueden originar en ambas subredes de la VPC-DNS. Así, cualquier consulta que le llegue a la VPC proveniente de AWS será encaminada al entorno local.

```
OutboundDnsResolver:
    Type: AWS::Route53Resolver::ResolverEndpoint
    Properties:
      Direction: OUTBOUND
      IpAddresses:
        - SubnetId:
            Ref: PublicSubnet
        - SubnetId:
            Ref: PrivateSubnet
```

Programación 4.10: Código CloudFormation para despliegue del Outbound Resolver Endpoint

Una vez se han creado los dos Endpoints, la VPC-DNS tiene establecido un punto de entrada y un punto de salida para las consultas DNS. Sin embargo, falta por configurar un par de reglas que establezcan en la infraestructura que las consultas originadas localmente y dirigidas a la nube se deben encaminar al Inbound Endpoint, y las consultas originadas en la nube y dirigidas al entorno local al Outbound Endpoint. Para hacer esto, se establecen dos dominios generales: *onpremise.domain* para el entorno local y *awscloud.domain* para el entonrno de nube:

- La primera regla encaminará las consultas dirigidas al dominio de recursos en la nube *awscloud.domain* hacia las dos direcciones IP reservadas en la VPC-DNS para el Inbound Endpoint. Se dirigen al puerto 53, ya que ese puerto ha sido abierto previamente para permitir el paso de las consultas DNS (que son de tipo TCP y UDP).

```
ResolverRuleAWS:
    Type: AWS::Route53Resolver::ResolverRule
    Properties:
      DomainName: awscloud.domain
      RuleType: FORWARD
      TargetIps:
        - Ip: 10.1.0.7
          Port: 53
        - Ip: 10.1.2.7
          Port: 53
      ResolverEndpointId: !Ref OutboundDnsResolver
```

Programación 4.11: Código CloudFormation para establecimiento de la regla *awscloud.domain*

- La segunda regla encamina las consultas DNS dirigidas al dominio local *onpremise.domain* a la dirección IP del servidor DNS de las instalaciones locales. En este caso, dado que las instalaciones locales están siendo simuladas mediante una VPC, se encaminan a la dirección IP .2 de la VPC, que es la que AWS reserva en cada

VPC para la resolución de DNS. Como en este caso la VPC de simulación del entorno local tiene CIDR 172.0.0.0/16, su resolución de DNS interna se lleva a cabo en la dirección IP 172.0.0.2.

```
ResolverRuleToOnPremise:
    Type: AWS::Route53Resolver::ResolverRule
    Properties:
      DomainName: onpremise.domain
      RuleType: FORWARD
      TargetIps:
        -
          Ip: 172.0.0.2
          Port: 53
      ResolverEndpointId: !Ref OutboundDnsResolver
```

Programación 4.12: Código CloudFormation para establecimiento de la regla *onpremise.domain*

Con los Endpoints y las reglas establecidas, la VPC-DNS está preparada para resolver todas las consultas DNS que se generen en cualquier punto de la infraestructura. Cualquier petición dirigida a *onpremise.domain* o cualquier subdominio asociado (como podría ser *servidor.onpremise.domain* será encaminada al servidor DNS local, y cualquier petición dirigida a *awscloud.domain* será encaminada por el Inbund Endpoint a su recurso asociado.

4.2.1.3. Configuración del entorno para la resolución de DNS

Con la VPC-DNS preparada para resolver cualquier consulta DNS sin importar donde de la infraestructura se origine o se dirija, el último paso es configurar la infraestructura para que cada elemento en ella esté asociado a un dominio.

Para este proyecto se plantea que cada VPC desplegada en la nube tenga un subdominio de *awscloud.domain* propio. Es decir, si se despliegan dos VPCs, existirían dos subdominios: *zona1.awscloud.domain* y *zona2.awscloud.domain*. Así, todos los entornos donde se pueden alojar cualquier tipo de recurso o servicio tendrán un dominio propio, simplificando la gestión y el encaminamiento de las consultas DNS.

Así, en la figura 4.10 se muestra como se plantea el establecimiento de los nombres de dominio en la arquitectura global.

Por lo tanto, la gestión de DNS pasa a ser extremadamente eficaz: cada vez que se necesite desplegar una nueva VPC, simplemente hay que crearle un nombre de dominio (o Hosted Zone en AWS Route 53) y asociarlo primero a dicha VPC y luego a la VPC-DNS. Esta última asociación permite que el nuevo dominio tenga asociadas las dos reglas de resolución de DNS creadas en el apartado 4.2.1.2, por lo que sabrá como encaminar todas las peticiones de DNS que origine o reciba.

Una vez se ha hecho esto, el Route 53 Resolver es capaz de encaminar cualquier consulta DNS con destino a dicho dominio a la VPC asociada. Por lo tanto, si se decide desplegar una VPC nueva, digamos una VPC-Ejemplo, simplemente habría que crearle un dominio asociado *ejemplo.awscloud.domain* y asociarlo con la VPC-Ejemplo y con la VPC-DNS.

Así, si por ejemplo se despliega en la VPC-Ejemplo un servidor y se desea asociarle un subdominio, simplemente habría que crear un registro para dicho servidor, como podría ser

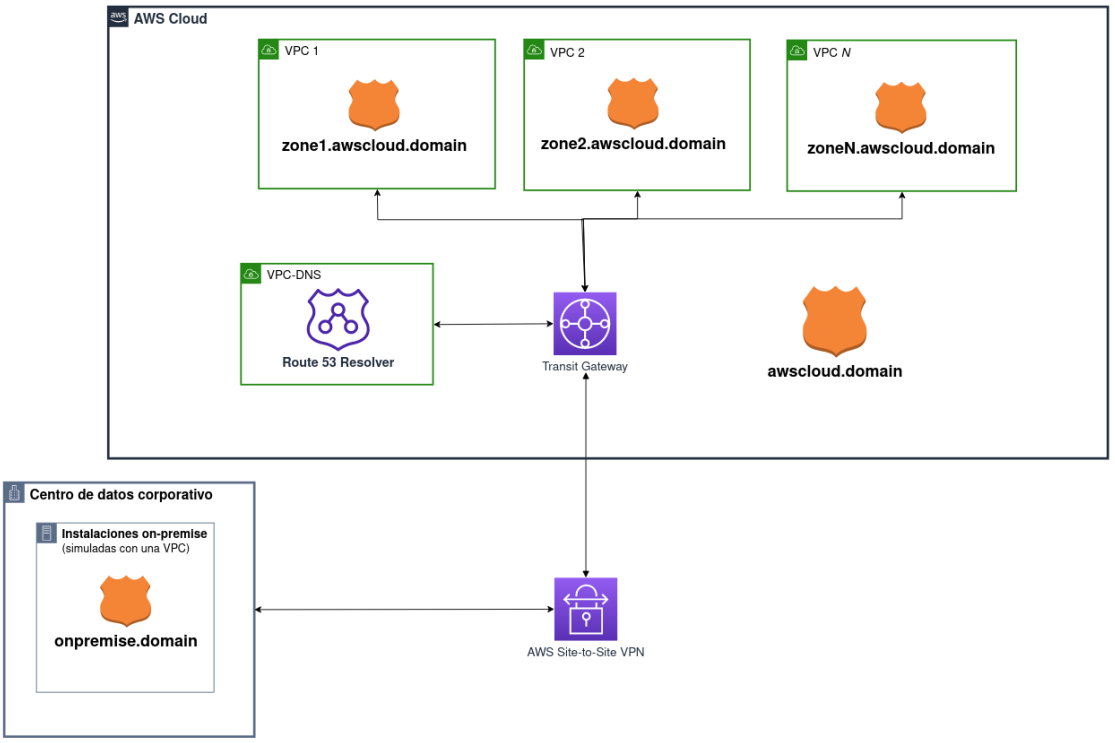

Figura 4.10: Arquitectura con los nombres de dominio establecidos

servidor.ejemplo.awscloud.domain. Con esto hecho, cualquier usuario o recurso en cualquier punto de la arquitectura podrá buscar el dominio *servidor.ejemplo.awscloud.domain* y acceder al servidor.

4.2.1.4. Ejemplo de funcionamiento de resolución de DNS centralizada y automática

Con las configuraciones realizadas previamente, la arquitectura planteada es capaz de resolver cualquier consulta DNS. Si en cualquier momento se despliega un servicio en cualquier VPC o en el entorno local y se le crea un registro de dominio asociado, el Route 53 Resolver lo recoge y es capaz de encaminar cualquier consulta dirigida a él de forma automática.

Por lo tanto, la gestión se reduce al momento de creación de una VPC y de sus servicios asociados, quedando toda la resolución de DNS posterior completamente centralizada y automatizada por el Route 53 Resolver.

Para explicar de forma más clara como funciona la resolución de DNS a través del Route 53 Resolver, se plantea el siguiente caso: en la arquitectura mostrada en la figura 4.10, se alojan primero dos servidores en VPC-1 y VPC-2 y se les crea los registros de dominio *servidor.zona1.awscloud.domain* y *servidor.zona2.awscloud.domain*. A continuación se despliega en las instalaciones locales otro servidor, y en el servidor de DNS local se le asocia el registro de dominio *servidor.onpremise.domain*.

El primer caso, ilustrado en la figura 4.11, es que el servidor alojado en la VPC-1 necesite comunicarse con el servidor del entorno local, por lo que lanzará una petición DNS a *servidor.onpremise.domain*. Dicha petición irá al servidor DNS de la VPC-1 (salto 1), que al tener asociadas las reglas de resolución de DNS creadas previamente, la encamina al

Outbound Endpoint en la VPC-DNS (salto 2). Como la petición DNS va dirigida a un dominio del entorno local, el Outbound Endpoint dirige la consulta al servidor DNS del entorno local (salto 3), el cual dirige la consulta consecuentemente al servidor buscado. Así, la petición encuentra la dirección IP del servidor asociado al dominio buscado, y la consulta del dominio se resuelve.

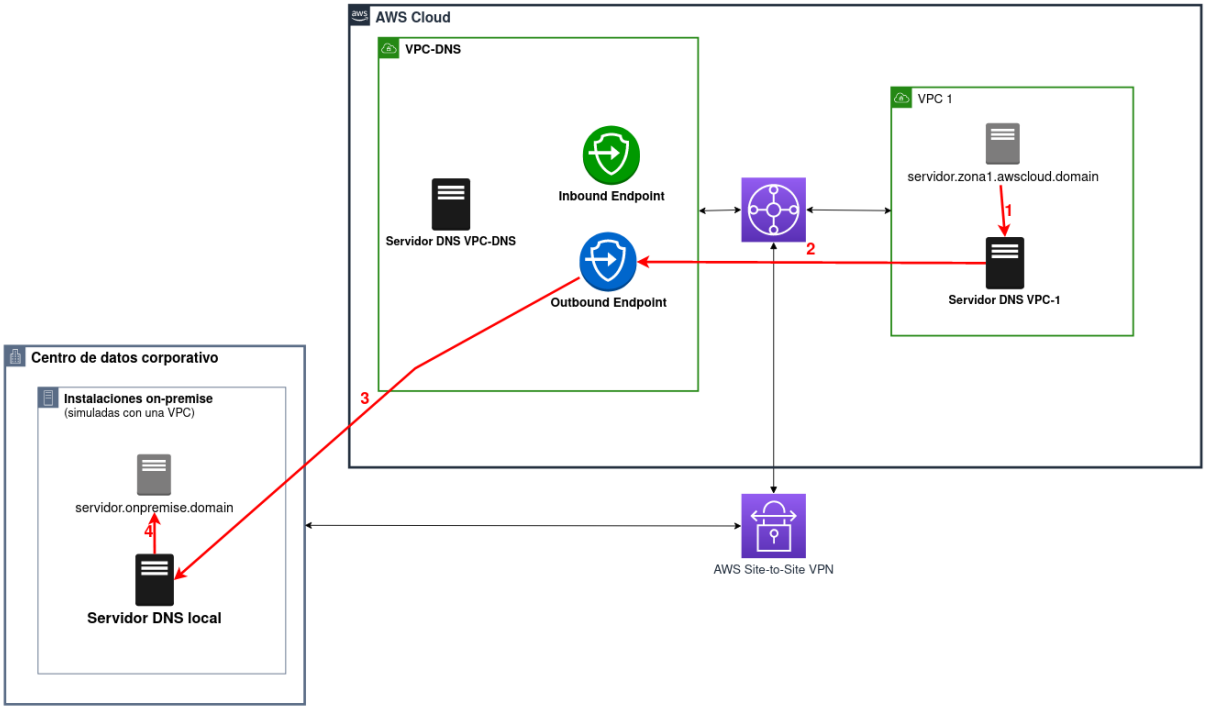

Figura 4.11: Consulta DNS desde VPC hacia entorno local

El segundo caso , ilustrado en la figura 4.12, es que el servidor del entorno local necesite comunicarse con el servidor de la VPC-1, lanzando una petición DNS al dominio *servidor.zona1.awscloud.domain*. Dicha petición se encaminará al servidor DNS local (salto 1), que al tener configurado que peticiones al dominio *awscloud.domain* se deben encaminar a la dirección IP del Inbound Endpoint, lo hace (salto 2). Dado que la consulta ya se encuentra en el entorno de AWS y está dirigida a un dominio de AWS, puede ser resuelta por el servidor DNS de la VPC-DNS (salto 3). Dicho servidor, al tener asociado el dominio destino, sabe como encaminar hacia él (salto 4). Así, el servidor DNS de la VPC destino recibe la consulta (salto 5) y la encamina al servidor correspondiente, por lo que se resuelve.

Finalmente se plantea el tercer caso posible, mostrado en la figura 4.13, en el que el servidor ubicado en la VPC-1 se necesite comunicar con el servidor ubicado en la VPC-2. Primero, la petición pasa del servidor de VPC-1 al servidor DNS de dicha VPC (salto 1). Como una de la reglas definidas 4.2.1.2 indica que el tráfico dirigido a *awscloud.domain* debe encaminarse al Inbound Endpoint, esto ocurre a través del Outbound Endpoint (salto 2). Posteriormente, el Inbound Endpoint encamina la consulta al servidor DNS de la VPC-DNS (salto 3), y dado que la VPC-DNS está asociada con el dominio *zona2.awscloud.domain*, se encamina la consulta al servidor DNS de la VPC-2 (salto 4). Así, se encamina finalmente al servidor y la consulta DNS se resuelve.

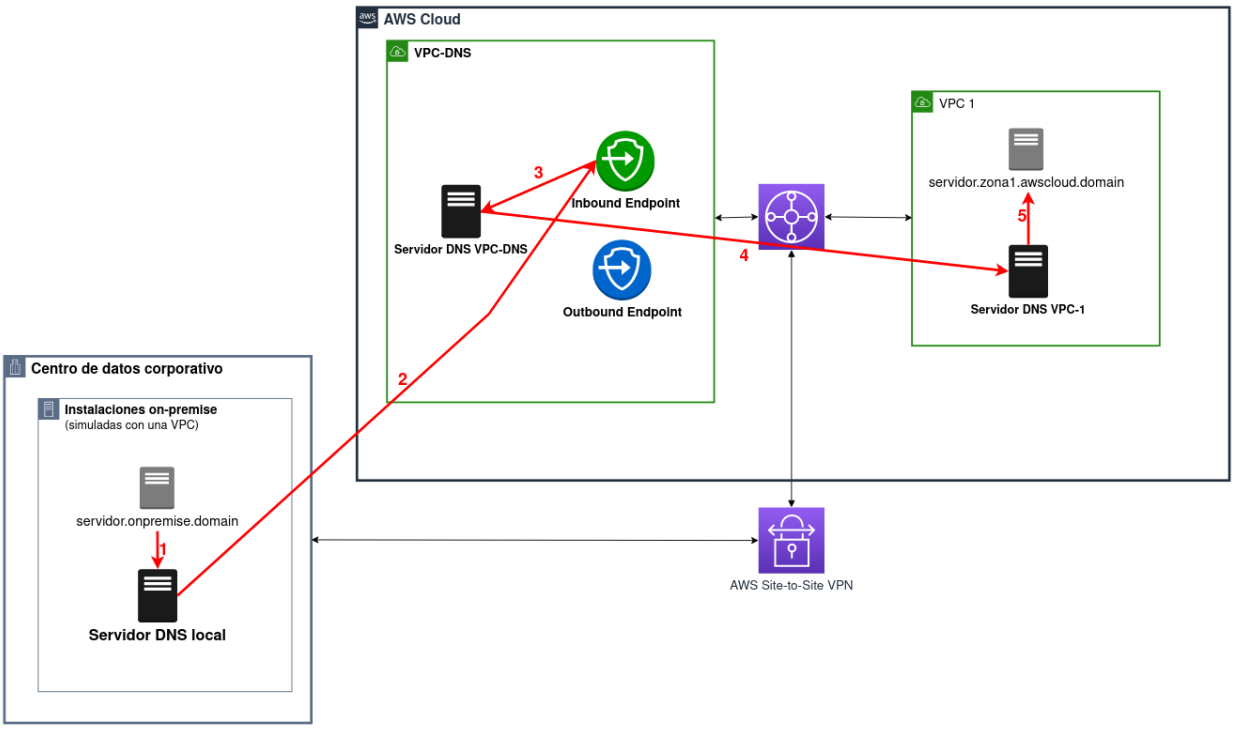

Figura 4.12: Consulta DNS desde entorno local hacia VPC

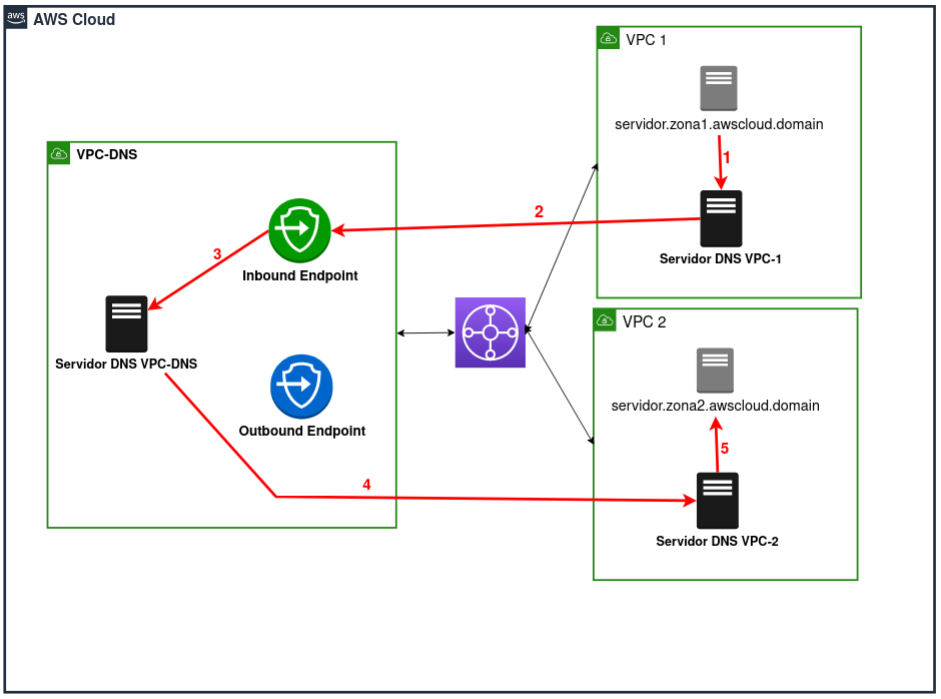

Figura 4.13: Consulta DNS desde VPC hacia VPC

5. Resultados

5.1. Pruebas realizadas

Dado que en este proyecto el principal objetivo era el despliegue de arquitecturas híbridas funcionales, las pruebas realizadas para la comprobación del correcto funcionamiento del sistema han consistido en un conjunto de comprobaciones de conectividad a nivel de red.

En la primera fase del proyecto, en las arquitecturas de los apartados 4.1.1 y 4.1.2, dado que se buscaba la conectividad a nivel de red, el correcto funcionamiento del sistema fue comprobado a través del comando *ping* entre todas las posibles combinaciones de comunicación: entorno local con entorno de nube y viceversa, y entorno de nube con entorno de nube. Para la realización de esta prueba se desplegaron servidores EC2 en los distintos entornos y se lanzó el comando, lo que de funcionar bien, demostraba el correcto funcionamiento de la arquitectura.

Dicho comando demuestra la correcta conectividad a nivel de red, por lo que demuestra que el sistema cumple con los objetivos impuestos.

En cuanto a las pruebas realizadas para la comprobación del correcto funcionamiento de la herramienta de gestión y resolución centralizada y automática de DNS, se utilizó el comando *nslookup*, una herramienta de línea de comandos utilizada para realizar consultas de DNS y obtener información relacionada con los nombres de dominio.

Se desplegaron servidores EC2 siguiendo los tres casos explicados en el apartado 4.2.1.4, y para cada servidor se creó un registro de dominio. A continuación, se comprobó con el comando *nslookup* que la consulta de DNS lanzada a cada dominio devolvía la dirección IP del servidor correspondiente, demostrando que la resolución de DNS entre dominios de distintos entornos funcionaba.

5.2. Resumen del sistema desarrollado

Durante el desarrollo del presente proyecto se ha logrado el objetivo de obtener un sistema capaz de desplegar de forma completamente automática arquitecturas de nube híbrida ofreciendo una total interconexión entre los entornos locales y los entornos de nube.

No sólo eso, si no que se ha desarrollado un sistema capaz de desplegar una infraestructura de nube compleja, escalable y flexible de forma plenamente automática, empleando nuevos servicios de AWS que ofrecen una hibridación más completa y versátil. Además, se ha incluido la implementación de una herramienta de gestión y resolución de DNS centralizada y automática, que simplifica en gran medida la gestión de dicha arquitectura, haciendo más intuitiva su utilización y dotándola de mayor flexibilidad y escalabilidad.

Recopilando los distintos módulos desarrollados, se ha obtenido finalmente un sistema que despliega de forma automática entornos de nube híbrida con la posibilidad de personalizar la arquitectura en función de las necesidades del usuario al permitir desplegar el número de VPCs deseadas automáticamente. Además, dicha arquitectura incluye la implementación de una herramienta que centraliza y automatiza la resolución de DNS en cualquier punto de la arquitectura.

5.3. Cumplimiento de requisitos

Una vez explicado el sistema que se ha desarrollado finalmente, se procede a analizar hasta que punto se han cumplido los requisitos que se le habían puesto al sistema, con objetivo de discutir si los resultados son satisfactorios.

El primer requisito impuesto al sistema era la necesidad de la hibridación pura entre el entorno local y el entorno de nube en las arquitecturas desplegadas. Considerando los resultados, se ha comprobado que la conectividad a nivel de red es completa entre todos los entornos, por lo que el resultado es satisfactorio. Además, la resolución de DNS ofrece una hibridación más intuitiva y sencilla de utilizar, mejorando la operación del sistema con respecto a los requisitos impuestos.

El segundo gran requisito impuesto era la automatización de los procesos llevados a cabo. Dicho requerimiento ha sido el pilar central del sistema, logrando no sólo la automatización del despliegue de las distintas arquitecturas, si no también el despliegue automático de la herramienta de resolución de DNS. Además, dicha herramienta resuelve el DNS de toda la arquitectura de forma completamente automática, por lo que dicho requisito se considera satisfactoriamente cubierto.

En cuanto al requisito de aseguración de la escalabilidad del sistema, también ha sido un eje central del trabajo. La implementación del Transit Gateway, al que se pueden conectar gran cantidad de VPCs de forma automática, ofrece una escalabilidad casi ilimitada a la arquitectura desplegada. Además, la herramienta de resolución de DNS aporta a la escalabilidad del sistema, al simplificar la utilización y gestión de este en caso de llegar a altas complejidades.

Finalmente, en cuanto a los requisitos de seguridad y resiliencia, se han aprovechado las recomendaciones propuestas por AWS para este propósito. La seguridad de la infraestructura se ha garantizado a través de la utilización de VPCs completamente aisladas del exterior y que se interconectan o a través de servicios de AWS (que se consideran seguros) o a través de conexiones VPN cifradas. La resiliencia de la arquitectura se ha asegurado con la implementación de elementos redundantes dentro de cada VPC (con subredes redundadas en distintas zonas de disponibilidad) y en el túnel VPN, asegurando la resiliencia del sistema ante caídas de servicios.

En resumen, se puede concluir que se han cumplido con creces los requisitos impuestos al sistema. Se ha asegurado el cumplimiento de cada requerimiento, y se han añadido funcionalidades que dan lugar a una mejor operabilidad del sistema obtenido, mejorando la experiencia del usuario en su utilización. Por lo tanto, se concluye que los resultados obtenidos son satisfactorios.

6. Conclusiones y líneas futuras

6.1. Conclusiones

La extensa cantidad de ventajas que aporta la utilización de entornos de nube híbrida en empresas u organizaciones ha provocado un gran crecimiento en su implementación durante los últimos años. La hibridación ofrece entornos de gran escalabilidad y flexibilidad a la vez que seguros y altamente controlables, pero tiende a conllevar un aumento en la complejidad de las arquitecturas empresariales. La implementación de dichas arquitecturas, por lo tanto, plantea problemas para las organizaciones a la hora de ser desplegadas y gestionadas.

Por lo tanto, en el presente trabajo se ha desarrollado un sistema de despliegue y gestión de arquitecturas de nube híbrida de gran escalabilidad, flexibilidad, seguridad, control y resiliencia de forma completamente automática y repetible. Este sistema trae consigo una reducción tanto del tiempo como de la probabilidad de fallos de implementación de dichas arquitecturas, además de ofrecer una experiencia de usuario simplificada e intuitiva en la utilización de dichos entornos al habilitar la resolución de DNS centralizada y automática.

Dicho sistema se ha desarrollado mediante la aplicación de herramientas y servicios principalmente provistos por Amazon Web Services, aunque también se han utilizado ciertas herramientas externas. La primera fase del desarrollo ha consistido en la familiarización y documentación de dichos servicios y herramientas, obteniendo una visión global de las posibilidades para modelar la arquitectura a desarrollar.

Una vez hecho esto, se propuso el despliegue inicial de una arquitectura simplificada pero completa y funcional, con objetivo de analizar la efectividad del proyecto. Una vez esta arquitectura inicial era completamente funcional y sus resultados satisfactorios, se propuso el despliegue de una arquitectura realista modelada según las necesidades de una empresa u organización real. Así, se obtuvo la capacidad de desplegar de manera completamente automática una arquitectura relativamente compleja muy similar a la que una organización podría utilizar.

Con el modelado y despliegue de dicha arquitectura finalizado, se propuso la mejora de la experiencia del usuario al utilizarla. La cantidad de mejoras que se pueden implementar en una arquitectura empresarial es muy extensa, variada y muy específica acorde a las necesidades concretas de cada empresa. Sin embargo, se identificó la posibilidad de facilitar la comunicación entre cargas de trabajo en distintos puntos del entorno híbrido con la implementación de una herramienta de resolución automática y centralizada de DNS.

Con dicha herramienta, se completó el desarrollo del sistema. Así, se obtuvo una arquitectura desplegable de forma automática y reproducible, capaz de interconectar el entorno local de una empresa u organización con tantos entornos aislados en la nube de AWS

como se desee. Al alojar cargas de trabajo en cualquier punto del entorno, es posible la creación de nombres de dominio que hacen de la comunicación dentro de la arquitectura empresarial un proceso simple e intuitivo.

Por lo tanto, este sistema ofrece una posibilidad rápida, eficiente y reproducible para la implementación de cargas de trabajo en la nube de AWS. En el caso de necesitar desplegar una en un momento puntual para cubrir una demanda creciente, este sistema posibilita la creación de un entorno privado, escalable y flexible sin tener que ocuparse del despliegue y configuración, eliminando la posibilidad de errores y reduciendo considerablemente el tiempo consumido.

Todo esto lleva a concluir que en un mercado tan estable y en crecimiento como es el de los entornos empresariales de nube híbrida, la posibilidad de implementar entornos aislados rápida y eficazmente resulta muy atractiva. Toda herramienta que automatice procesos manuales, ahorrando tiempo y recursos y reduciendo la probabilidad de fallos, es de gran valor para empresas y organizaciones. Por lo tanto, este sistema presenta una herramienta, potencialmente, de gran utilidad para las organizaciones que la implementen.

Sin embargo, el presente sistema es capaz de desplegar arquitecturas generalistas poco adaptadas a las necesidades puntuales de una empresa real, lo que da paso a las posibles líneas futuras de desarrollo y mejoría del sistema.

6.2. Líneas futuras

El presente sistema ofrece la capacidad de desplegar arquitecturas de nube híbridas completamente funcionales pero muy generales. Por lo tanto, su principal línea de desarrollo es incrementar la capacidad de personalizar las arquitecturas desplegadas, adaptándolas a las necesidades específicas de cada organización.

Existe una gran variedad de posibilidades a implementar:

- Herramientas de monitorización: al centralizar el paso de todo el tráfico de red, el Transit Gateway ofrece un punto idóneo para la monitorización de este. La implementación de servicios de AWS como CloudWatch posibilitaría la monitorización del tráfico en función de ciertas métricas preestablecidas y el lanzamiento de alertas en caso de que se incumpliesen ciertos umbrales impuestos.
- Seguridad mejorada: con respecto a la ofrecida por defecto por AWS agregando capas adicionales de seguridad al sistema. Se sugiere la utilización de AWS Security Hub para monitorizar y gestionar la postura de seguridad y de AWS Web Application firewall para proteger las aplicaciones desplegadas.
- Automatización de backups y recuperación ante desastres: implementación de servicios como AWS Backup y AWS Disaster Recovery para automatizar y gestionar las copias de seguridad y los planes de recuperación de las VPC, Transit Gateway y otros componentes críticos desplegados, aumentando la resiliencia y seguridad del sistema.
- Escalabilidad y optimización de recursos: para mejorar la capacidad del sistema de escalar de manera eficiente y de optimizar el uso de recursos, se propone la integración de servicios como AWS Auto Scaling o AWS Cost Explorer para ajustar automáticamente la capacidad de tus recursos y controlar los costos.

Estas son simplemente algunas sugerencias a la hora de querer mejorar y adaptar el presente sistema a las posibles necesidades de una organización concreta.

Bibliografía

[1] McKinsey. «Tech at the edge: Trends reshaping the future of IT and business», [en línea]. Available: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/tech-at-the-edge-trends-reshaping-the-future-of-it-and-business.

[2] Shivam Arora. «Emerging Technologies Enabled by the Cloud», [en línea]. Available: https://www.simplilearn.com/emerging-technologies-enabled-by-the-cloud-article.

[3] IBM. «¿Qué es computación en la nube?», [en línea]. Available: https://www.ibm.com/es-es/topics/cloud-computing.

[4] Red Hat. «Cloud Computing», [en línea]. Available: https://www.redhat.com/es/topics/cloud.

[5] P. M. Mell and T. Grance. The NIST definition of cloud computing. Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology, 2011.

[6] Microsoft Azure. «What is cloud computing?», [en línea]. Available: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing.

[7] L. Cheng Q. Zhang and R. Boutaba. Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[8] AWS Documentation. «What is Amazon VPC?», [en línea]. Available: https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html.

[9] Cisco. «2022 Global Hybrid Cloud Trends Report», [en línea]. Available: https://www.cisco.com/c/en_au/solutions/hybrid-cloud/2022-trends-report-cte.html.

[10] CloudFlare. «How does hybrid cloud architecture work?», [en línea]. Available: https://www.cloudflare.com/es-es/learning/cloud/how-does-hybrid-cloud-architecture-work/.

[11] Gartner. «Gartner Says Worldwide IaaS Public Cloud Services Market Grew 41.4 % in 2021», [en línea]. Available: https://www.gartner.com/en/newsroom/press-releases/2022-06-02-gartner-says-worldwide-iaas-public-cloud-services-market-grew-41-percent-in-2021.

[12] AWS Documentation. «What is IaaS», [en línea]. Available: https://aws.amazon.com/es/what-is/iaas/.

[13] AWS Documentation. «AWS Cloudformation», [en línea]. Available: https://aws.amazon.com/es/cloudformation/.

[14] AWS Documentation. «AWS CLI Documentation», [en línea]. Available: https://docs.aws.amazon.com/cli.

[15] Gurudatt Kulkarni, Ramesh Sutar, and Jayant Gambhir. Cloud computing-infrastructure as service-amazon ec2. *International journal of Engineering research and applications*, 2(1):117–125, 2012.

[16] AWS Documentation. «What is AWS Site-to-Site VPN?», [en línea]. Available: https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.html#s2svpn-features.

[17] AWS Documentation. «AWS Site-to-Site VPN, choosing the right options to optimize performance», [en línea]. Available: https://aws.amazon.com/es/blogs/networking-and-content-delivery/aws-site-to-site-vpn-choosing-the-right-options-to-optimize-performance.

[18] AWS Documentation. «AWS Transit Gateway», [en línea]. Available: https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html.

[19] AWS Documentation. «What is Amazon EC2», [en línea]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html.

[20] AWS Documentation. «Amazon Route 53», [en línea]. Available: https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html.

[21] AWS Documentation. «Amazon Route 53 Resolver», [en línea]. Available: https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resolver.html.

[22] AWS Labs. VPC with managed NAT and private subnet, [Source code] . Repository: https://github.com/awslabs/aws-cloudformation-templates/blob/master/aws/services/VPC/VPC_With_Managed_NAT_And_Private_Subnet.yaml, 2019.

[23] AWS Documentation. «Availability with redundancy», [en línea]. Available: https://docs.aws.amazon.com/whitepapers/latest/availability-and-beyond-improving-resilience/availability-with-redundancy.html.

[24] AWS Documentation. «NAT Gateways», [en línea]. Available: https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html.

[25] strongSwan. «About strongSwan», [en línea]. Available: https://www.strongswan.org/about.

[26] AWS Samples. VPN gateway - strongSwan, [Source code] . Repository: https://github.com/aws-samples/vpn-gateway-strongswan, 2019.

Anexo A: Aspectos éticos, económicos, sociales y ambientales

A1. Introducción

En este Trabajo de Fin de Grado se ha estudiado el desarrollo de un sistema de automatización del despliegue y gestión de arquitecturas de nube híbrida en el entorno de AWS. Dicho sistema se ha desarrollado mediante la implementación de servicios y herramientas propios de AWS principalmente, además de ciertas herramientas externas. A continuación se exponen los impactos relevantes de este Trabajo de Fin de Grado en ciertos contextos.

A2. Descripción de impactos relevantes relacionados con el proyecto

Dado que en este trabajo se ha desarrollado un sistema de despliegue automático de entornos de nube híbrida, no tiene un impacto relevante en el contexto ni social ni ético. Sin embargo, se considera que dicho sistema si puede causar cierto impacto a nivel medioambiental y económico.

A nivel medioambiental, el *cloud computing* como tecnología puede tener un impacto positivo. Al utilizar centros de datos especializados y de gran tamaño, con una mayor consolidación de recursos y la aplicación de tecnologías más eficientes, los proveedores de servicios en la nube logran una mayor eficiencia energética en comparación con las implementaciones locales. Además, la escalabilidad adaptable de los recursos *cloud*, al permitir ajustar dinámicamente los recursos en función de la demanda de usuarios, evita el desperdicio de recursos en momentos de baja demanda.

Dado que en el presente trabajo se busca facilitar el alojamiento de cargas de trabajo en la nube mediante el desarrollo de un sistema que hace de esto un proceso automático, simple y eficiente, pretende tener como resultado una mayor utilización por parte de empresas de entornos *cloud*, potenciando las ventajas medioambientales expuestas previamente.

Sin embargo, el principal impacto que pretende tener este sistema es a nivel económico, por lo que será explorado en más profundidad en la sección A3.

A3. Análisis del impacto económico

La implementación del sistema desarrollado en este trabajo por parte de empresas y organizaciones tiene el potencial de generar ahorros de costes significativos a través de

varios factores.

En primer lugar, la agilización y simplificación del despliegue de entornos híbridos facilita en gran medida el proceso de alojamiento de cargas de trabajo en la nube. Esto supone una eficiencia operativa significativamente mejorada, ya que al eliminar los procesos de despliegue y mantenimiento manuales, se reduce el tiempo de inactividad y se aumenta la productividad del personal, liberando tiempo y recursos para tareas más estratégicas. Como consecuencia de esto, la automatización y simplificación del despliegue de nubes híbridas también puede evitar que las organizaciones tengan la necesidad de invertir en hardware costoso, suponiendo ahorros significativos en términos de costos de capital y gastos operativos.

La implementación de este sistema también supone un acceso aumentado a la escalabilidad y flexibilidad que ofrece la nube, permitiendo a las organizaciones y empresas ajustarse de forma más precisa a la demanda, utilizando los mínimos recursos necesarios y por lo tanto permitiendo optimizar costos y evitar gastos innecesarios.

Por último, y desde un punto de vista más global, la implementación de la nube por parte de las empresas supone un gran facilitador de la innovación y el emprendimiento. La nube ha democratizado el acceso a recursos informáticos y ha eliminado una gran barrera de entrada para nuevas empresas y emprendedores, fomentando la innovación al permitir a las empresas desarrollar y lanzar productos y servicios de manera más ágil y rentable. Unido a esto, la nube también ofrece a las organizaciones un impulso hacia la transformación digital, permitiéndoles la adopción de gran variedad de nuevas tecnologías que generan ventajas competitivas y beneficios económicos significativos.

A4. Conclusiones

Dado que el presente proyecto desarrolla una herramienta de utilización interna para empresas u organizaciones, su impacto social y ético puede ser limitado. Sin embargo, puede generar impactos económicos y medioambientales significativos para las organizaciones que implementen el sistema desarrollado.

El principal valor añadido que ofrece este proyecto es la automatización de procesos y la reducción de la carga de trabajo para el personal. Esto puede generar eficiencia operativa y ahorros de costos, lo que impacta positivamente en el aspecto económico de las organizaciones. Además, al facilitar el acceso a la nube, el sistema puede tener un impacto económico beneficioso para las organizaciones. La adopción de servicios en la nube puede proporcionar flexibilidad, escalabilidad y reducción de costos en comparación con las infraestructuras locales.

En términos medioambientales, este proyecto puede aportar un impacto positivo al promover la eficiencia energética y la reducción de residuos electrónicos. Al evitar la necesidad de adquirir y mantener infraestructura física costosa, se contribuye a la conservación de recursos naturales y a la disminución de emisiones de carbono.

El uso de criterios de sostenibilidad puede aportar valor añadido al proyecto al promover prácticas empresariales responsables y sostenibles. A través de la eficiencia energética, la reducción de residuos electrónicos y la consideración de aspectos sociales y económicos, el proyecto puede generar beneficios a largo plazo tanto para las organizaciones que lo implementen como para el entorno en general. La integración de criterios de sostenibilidad no solo puede mejorar la imagen y reputación de las organizaciones, sino también reducir

costos operativos, minimizar impactos ambientales y fomentar la responsabilidad social. Además, al permitir un acceso más eficiente y responsable a la nube, el proyecto contribuye a una transformación digital más sostenible y alinea a las organizaciones con los desafíos y demandas actuales relacionados con la sostenibilidad.

Anexo B: Presupuesto económico

COSTE DE MANO DE OBRA (coste directo)	Horas	Precio/hora	Total
	400	15 €	**6.000 €**

COSTE DE RECURSOS MATERIALES (coste directo)	Precio de compra	Uso en meses	Amortización (en años)	Total
Ordenador personal (Software incluido)	1.100,00 €	6	5	110,00 €

COSTE TOTAL DE RECURSOS MATERIALES	**110,00 €**

GASTOS GENERALES (costes indirectos)	15 %	sobre CD	**915,50 €**
BENEFICIO INDUSTRIAL	6 %	sobre CD + CI	**366,60 €**

MATERIAL FUNGIBLE	
Infraestructura y servicios de AWS	**156,35 €**

SUBTOTAL PRESUPUESTO		**7.548,45 €**
IVA APLICABLE	21 %	**1.585,18 €**

TOTAL PRESUPUESTO	**9.133,63 €**

Tabla 6.1: Presupuesto económico