PROYECTO FIN DE GRADO

TÍTULO: Despliegue y securización de un núcleo de 5G en la nube

AUTOR/A: Marta González Lucía

TITULACIÓN: Ingeniería Telemática

DIRECTOR/A: Diego Rodríguez Pérez

TUTOR/A: Pedro Castillejo Parrilla

DEPARTAMENTO: Ingeniería telemática y Electrónica (DTE)

VºBº TUTOR/A

Miembros del Tribunal Calificador:

PRESIDENTE/A: José David Osés del Campo

TUTOR/A: Pedro Castillejo Parrilla

SECRETARIO/A: Jesús Rodríguez Molina

Fecha de lectura:

Calificación:

El Secretario/La Secretaria,

You can't control the wind

but you can adjust your sails.

Choose Resilience

Agradecimientos

A mi familia, por ser mi sitio seguro siempre.

A mi tía Sonia, por transmitirme la pasión por esta bonita profesión que compartimos.

A Sergio, mi compañero de vida, por motivarme siempre a ser mejor y ser un pilar fundamental en todo lo que hago.

A mis amigos, Diego, Isabel y Beatriz, por apoyarme y ayudarme en todo lo posible.

A Isabel, por tus chistes malos.

A todos los profesores que durante este largo camino han confiado en mí, en especial a Pedro, por ayudarme siempre a que esto siguiera adelante.

Resumen

La evolución de las redes móviles ha sido un motor clave en el progreso tecnológico y económico, transformando nuestra manera de comunicarnos y acceder a la información. Cada nueva generación, desde el 2G hasta el 5G, ha marcado hitos en términos de capacidad, velocidad y funcionalidad.

El 5G representa un cambio de paradigma, ofreciendo velocidades de datos sin precedentes, latencias mínimas y capacidad para conectar una cantidad masiva de dispositivos simultáneamente. Estas características son cruciales para satisfacer las demandas de una variedad de casos de uso y tecnologías emergentes como el Internet de las Cosas (IoT).

La implementación de redes 5G puede requerir inversiones significativas, pero la virtualización de redes y la computación en la nube ofrecen un despliegue más eficiente y rentable. Estas tecnologías también transforman las redes 5G en entidades altamente escalables y flexibles, mejorando la eficiencia en costos.

En este proyecto se realiza una implementación de una red 5G, desplegándolo en la nube de Azure a través de una máquina virtual que hace la función de un nodo 5G. Se usan varias máquinas en local que simulan la red de acceso radio. Uno de los terminales móviles está securizado con Cloudflare a fin de proteger contra ataques como fishing o malware, creándose en todas sus conexiones una VPN con todo el tráfico cifrado.

Palabras Clave

Redes móviles, virtualización redes, computación en la nube, máquina virtual, 5G, acceso radio, comunicación, VPN, Cloudflare, seguridad.

Abstract

The evolution of mobile networks has been a key driver in technological and economic progress, transforming the way we communicate and access information. Each new generation, from 2G to 5G, has marked milestones in terms of capacity, speed, and functionality.

5G represents a paradigm shift, offering unprecedented data speeds, minimal latencies, and the ability to connect a massive number of devices simultaneously. These features are crucial to meet the demands of a variety of use cases and emerging technologies like the Internet of Things (IoT).

Implementing 5G networks may require significant investments, but network virtualization and cloud computing offer a more efficient and cost-effective deployment. These technologies also transform 5G networks into highly scalable and flexible entities, improving cost efficiency.

In this project, a 5G network implementation is carried out, deploying it in the Azure cloud through a virtual machine that functions as a 5G node. Several local machines are used to simulate the radio access network. One of the mobile terminals is secured with Cloudflare to protect against attacks such as phishing or malware, creating a VPN with encrypted traffic for all its connections.

Keywords

Mobile networks, network virtualization, cloud computing, virtual machine, 5G, radio access, communication, VPN, Cloudflare, security.

Índice de figuras

Índice de tablas

Lista de acrónimos

3GPP	*3rd Generation Partnership Project*
5GC	*5th Generation Core*
AF	*Application Function*
AMF	*Access and Mobility Management Function*
AR	*Augmented Reality*
AUSF	*Authentication Server Function*
AWS	*Amazon Web Services*
CA	*Certificate Authority*
CDN	*Content Delivery Network*
CLI	*Command Line Interface*
CU	*Central Unit*
DN	*Data Network*
GSM	*Global System for Mobile*
IaaS	*Infrastructure as a Service*
IoT	*Internet of Things*
LTE	*Long Term Evolution*
MAC	*Media Access Control*
MEC	*Multi-access Edge Computing*
MIMO	*Multiple-Input Multiple-Output*
NEF	*Network Exposure Function*
NRF	*Network Repository Function*
NSSF	*Network Slice Selection Function*
PCF	*Policy Control Function*
PKI	*Public Key Infrastructure*
PaaS	*Platform as a Service*

QoS	*Quality of Service*
RAN	*Radio Access Network*
SCTP	*Stream Control Transmission Protocol*
SMF	*Session Management Function*
SaaS	*Software as a Service*
TCP	*Transmission Control Protocol*
UDM	*Unified Data Management*
UDP	*User Datagram Protocol*
UE	*User Equipment*
UPF	*User Plane Function*
VPN	*Virtual Private Network*
VR	*Virtual Reality*

Índice

1. Introducción

1.1 Marco y motivación del proyecto

Las redes móviles han experimentado una evolución exponencial en los últimos años, avanzando desde 2G hasta 5G. La tecnología 5G representa un avance revolucionario en el ámbito de las telecomunicaciones, ofreciendo mejoras drásticas en términos de ancho de banda y latencia en comparación con sus predecesores. Esta nueva generación de redes móviles puede alcanzar velocidades superiores a los 10 Gbps y latencias tan bajas como un milisegundo, facilitando la comunicación en tiempo real y el manejo eficiente de grandes volúmenes de datos.

Aunque la tecnología 4G ha sido un avance significativo, ha tenido que evolucionar para hacer frente al aumento en la demanda de velocidad y capacidad. Este incremento ha sido impulsado por nuevas aplicaciones que requieren una transmisión de datos rápida y confiable, como la realidad aumentada (AR), la realidad virtual (VR) y los vehículos autónomos. La necesidad de soportar estas aplicaciones ha llevado al desarrollo y adopción de la tecnología 5G, que proporciona la infraestructura necesaria para satisfacer estas demandas crecientes y permitir el avance de la conectividad global. [1]

El impacto del 5G se extiende a múltiples sectores industriales, transformando la manufactura, la logística y la agricultura mediante la automatización avanzada. En el sector de la salud, el 5G facilita la telemedicina, las cirugías remotas y la monitorización en tiempo real de pacientes, mejorando la calidad de la atención médica y contribuyendo a salvar vidas. En el ámbito automotriz, la comunicación instantánea entre vehículos e infraestructuras, posibilitada por el 5G, es clave para el desarrollo de sistemas de transporte inteligente y vehículos autónomos, lo que mejora la seguridad vial y la eficiencia del tráfico.

No obstante, debido a su arquitectura menos centralizada y a la mayor dependencia de los programas informáticos, las redes 5G presentan más puntos de acceso para los atacantes, por lo que garantizar su seguridad es primordial. [2]

El despliegue de un núcleo de red 5G en la nube se perfila como una solución eficaz. Además de abordar la creciente demanda mediante la simplificación de su implementación, esta estrategia mejora la calidad del servicio, gracias a la escalabilidad y resiliencia propias de la computación en la nube. Al mismo tiempo, asegurar las conexiones con Cloudflare fortalece la seguridad de la infraestructura, protegiéndola contra posibles amenazas.

1.2 Objetivos

El objetivo principal de este proyecto es desplegar un núcleo de red 5G en la nube, aprovechando las capacidades y ventajas que ofrece la computación en la nube. Además, se busca reforzar la seguridad de este núcleo y sus conexiones mediante servicios de red que mejoran la protección de las comunicaciones.

Se definen los siguientes objetivos específicos:

- Creación del entorno en el que se va a desplegar el núcleo de red 5G, apoyándose en el entorno elegido en cloud.
- Proporcionar conectividad al núcleo de red para la posterior conexión con el resto de elementos de la red.
- Proporcionar la seguridad requerida a los terminales de la arquitectura a través de la herramienta de seguridad elegida.

1.3 Estructura de la memoria

En este apartado se proporciona la información contenida en cada uno de los apartados en los que se divide este proyecto, ofreciendo una vista general sobre el mismo.

Se divide en los siguientes capítulos:

1. **Introducción**

En este capítulo se describe la motivación del proyecto y un breve contexto de dónde se sitúa, analizando la evolución de las redes a lo largo del tiempo y la necesidad de securización de las mismas. Se detallan los objetivos que se pretenden alcanzar con este proyecto y se finaliza el capítulo con una explicación de cómo se divide la memoria.

2. **Estado del arte**

En este capítulo se detallan las tecnologías que se van a utilizar para su desarrollo, realizándose un estudio de las posibilidades que hay para llevarlo a cabo y justificando el por qué las seleccionadas.

3. **Especificaciones y restricciones de la solución**

En este apartado se incluye el diseño y arquitectura de la solución propuesta para este proyecto.

4. **Implementación de la solución**

En este apartado se detalla paso a paso todo el procedimiento para poder llegar a la solución propuesta, definiendo las especificaciones y restricciones del mismo.

5. **Validación de resultados**

En este apartado se evalúa el funcionamiento de la solución propuesta a través de un análisis de los resultados obtenidos.

6. **Presupuesto**

En esta sección se detalla el presupuesto requerido para la realización de este proyecto.

7. **Impacto**

En este capítulo se indicarán las implicaciones sociales, seguridad, ambientales y económicas relacionadas con el proyecto, aportando a los Objetivos de Desarrollo Sostenible.

8. **Conclusiones**

En este capítulo se lleva a cabo un análisis de los objetivos previamente establecidos y cómo se han cumplido.

9. **Referencias**

Este apartado contiene las referencias con la información que ha ayudado a poder realizar este proyecto

2. Marco tecnológico

En esta sección se llevará a cabo una investigación exhaustiva sobre las tecnologías que se utilizarán en el proyecto. Se realizará un análisis detallado para determinar cuál de estas tecnologías será seleccionada para la implementación, fundamentando esta elección en criterios específicos y justificando por qué es la más adecuada para los objetivos y requerimientos del proyecto.

2.1 Redes de comunicaciones móviles

En las últimas décadas, uno de los principales impulsores de la conectividad global ha sido el avance tecnológico en el ámbito de las comunicaciones móviles. La evolución de las redes móviles, desde los primeros sistemas analógicos hasta tecnologías modernas como el 5G, ha transformado la manera en que se comunica, se accede a la información y se utilizan los servicios digitales en la vida cotidiana.

En este apartado se llevará a cabo un análisis exhaustivo de la trayectoria histórica y el estado actual de las redes de comunicaciones móviles, con especial énfasis en la tecnología más reciente, el 5G. Al explorar esta evolución, se podrá comprender mejor cómo se ha llegado a la era actual de la conectividad móvil y entender el impacto del 5G en la sociedad y la economía.

2.1.1 Evolución de las tecnologías 2G, 3G y 4G

La evolución de las comunicaciones móviles ha estado marcada por hitos significativos con cada generación tecnológica. La introducción del 2G [3], con el sistema GSM como uno de sus estándares más importantes, supuso una revolución al digitalizar las redes y mejorar tanto la eficiencia espectral como la calidad de voz en comparación con los sistemas analógicos. Esta tecnología permitió la transmisión de datos y servicios como los mensajes de texto, sentando las bases para futuras aplicaciones y servicios móviles.

La transición hacia el 3G estuvo impulsada por la creciente demanda de servicios de datos móviles, a medida que un número cada vez mayor de usuarios comenzó a utilizar dispositivos móviles para acceder a Internet y realizar actividades en línea más complejas. El 3G ofreció una experiencia de usuario mejorada con mayores velocidades de transmisión de datos, facilitando la navegación web, la descarga de contenido multimedia y aplicaciones avanzadas como la videoconferencia y la transmisión de video en tiempo real. Este avance abrió nuevas oportunidades tanto para consumidores como para empresas, consolidando la convergencia digital.

El paso del 3G al 4G representó un avance aún mayor, impulsado por la necesidad de mayor velocidad, capacidad y eficiencia en la transmisión de datos. El 4G ofreció velocidades de conexión significativamente más rápidas y un rendimiento mejorado gracias a tecnologías de modulación más eficientes y a una mayor capacidad de procesamiento en dispositivos móviles. Esto permitió una experiencia de usuario más fluida y el acceso a servicios de datos

intensivos, como la transmisión de video en alta definición, juegos en línea de alta calidad y aplicaciones multimedia avanzadas. La adopción del 4G se aceleró debido a la creciente competencia entre proveedores de servicios móviles y los avances en los estándares de la industria, como el Long-Term Evolution (LTE).

La transición al 5G ha sido impulsada por la necesidad de una conectividad extremadamente rápida y confiable para soportar una amplia variedad de aplicaciones emergentes, como el Internet de las Cosas (IoT), la realidad aumentada (RA), la realidad virtual (RV), los vehículos autónomos y la automatización industrial. El 5G promete velocidades de datos mucho más altas, menor latencia y mayor capacidad de red, lo que permitirá una experiencia de usuario más rica y la viabilidad de aplicaciones y servicios avanzados. Además, la competencia global por liderar en el desarrollo e implementación del 5G ha sido un factor clave en su despliegue, impulsando la innovación y la competitividad económica a nivel mundial.

2.1.2 Arquitectura 5G

La quinta generación de tecnología móvil, conocida como 5G, representa un avance significativo en las comunicaciones inalámbricas, ofreciendo velocidades de conexión ultrarrápidas, menor latencia y mayor capacidad de red en comparación con las generaciones anteriores. Este avance tecnológico habilita una serie de aplicaciones y servicios avanzados, incluyendo el Internet de las Cosas (IoT), la realidad virtual (RV), la realidad aumentada (RA) y la automatización industrial.

El 5G está transformando diversos sectores, desde la atención médica hasta la producción, al facilitar la conectividad instantánea y confiable de dispositivos y máquinas. La creciente demanda de mayor velocidad, junto con la competencia global por liderar la innovación tecnológica y el desarrollo económico, impulsa su despliegue a nivel mundial.

A continuación se describen los componentes que forman la red 5G [4]:

- **UPF:** transporta el tráfico de datos IP entre el usuario final UE y la red externa. También puede llegar a conectarse a otros UPF.

- **UE:** son los dispositivos finales que se conectan a través del gNB, al núcleo 5G.

- **SMF**: gestiona las sesiones con el usuario final, se encarga del establecimiento de conexión, modificación y liberación de recursos una vez finalizada la comunicación.

- **AMF**: gestiona el acceso y la movilidad, actuando como único punto de entrada para la conexión del equipo usuario. También participa en la autenticación y gestiona el contexto de seguridad. Intercambia información entre el gNB y el UE.

- **AUSF**: gestiona la autenticación de usuarios, se comunica con el UDM para poder realizar este proceso.

- **UDM**: se encarga de generar credenciales de autenticación, además gestiona la identificación de los usuarios.

- **PCF**: guarda el marco de políticas que se deben de cumplir en la red a la hora de hacer uso de la misma.

- **NRF**: encargado de proporcionar las funciones de red disponibles en un momento determinado, con los servicios que ofrece, además soporta su descubrimiento.

- **NEF**: muestra las capacidades de la red a aplicaciones externas.

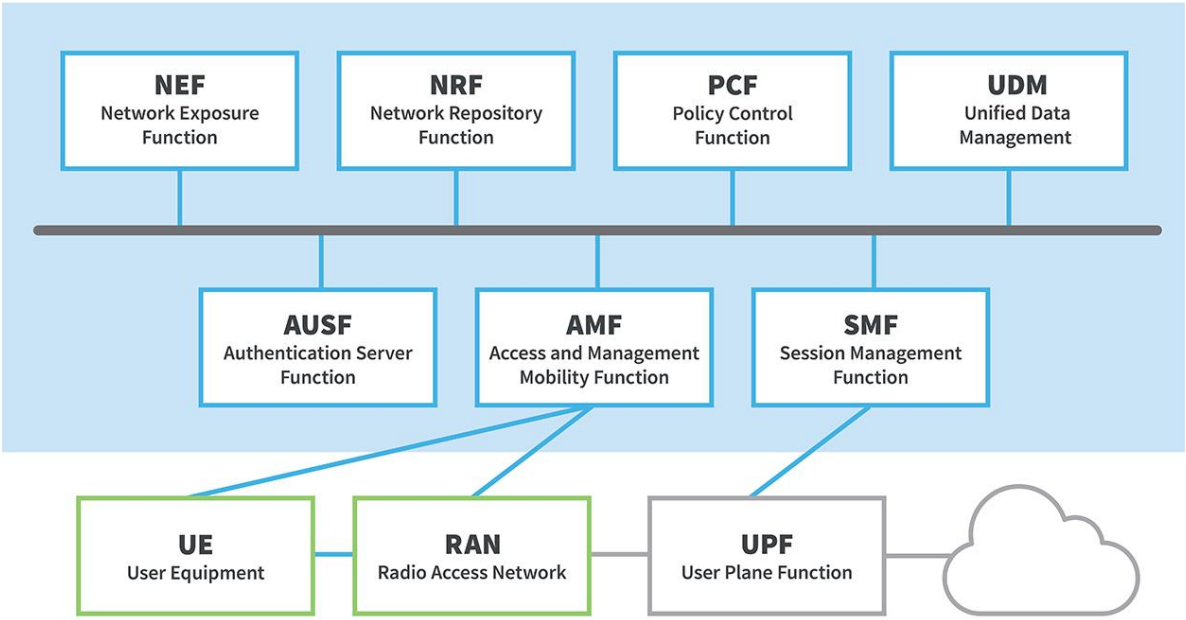

Ilustración 1: Arquitectura de red 5G [4]

La red 5G cuenta con estaciones base gNB (gNodeB) responsables de la transmisión y recepción de señales de radio entre el dispositivo final UE y el núcleo de red.

Las funciones del gNB son [5]:

- **Gestión de la interfaz radio:** el gNB es el encargado de la transmisión y recepción de señales radio a través de la interfaz de radio NR, incluyendo modulación, codificación y la gestión de recursos de radiofrecuencia.
- **Control de recursos radio:** establece y mantiene las conexiones de los dispositivos, gestiona la movilidad de los usuarios y garantiza la calidad del servicio (QoS).

- **Gestión de la movilidad:** supervisa la movilidad de los dispositivos conectados dentro de su área de cobertura y coordina el traspaso en caso de que el dispositivo se desplace a otra celda (handover).

- **Gestión de calidad de servicio:** implementa políticas de QoS para garantizar que el tráfico reciba el nivel de servicio adecuado, ya sea voz, vídeo o datos.

- **Manejo de MIMO:** usa técnicas como la de MIMO para dirigir de forma precisa y eficiente las señales a los dispositivos móviles conectados.

- **Seguridad:** participa en la autenticación de los dispositivos móviles y en medidas de seguridad como puede ser el cifrado de los datos, asegurando integridad y confidencialidad en sus comunicaciones.

- **Gestión de interferencias:** coordina las posibles interferencias con celdas adyacentes para mejorar la calidad de la señal.

- **Soporte para Edge computing:** permite que ciertos datos y servicios se procesen más cerca del usuario para poder reducir la latencia.

2.1.2.1 Tecnologías para el despliegue de red 5G

2.1.2.1.1 UERANSIM

UERANSIM [6] es un simulador de código abierto que proporciona un entorno de prueba y desarrollo para la interfaz de acceso por radio y la comunicación entre equipos de usuario (UE) y estaciones base (gNodeB). Sus principales características son:

- **Cumplimiento con estándares 3GPP**: Este software está diseñado para cumplir con los estándares definidos por el 3rd Generation Partnership Project (3GPP) [6] para redes 5G.

- **Implementación de 5G-SA (Standalone):** Ofrece una implementación de código abierto para 5G-SA (Standalone), lo que significa que soporta el modo independiente de 5G, sin depender de redes 4G existentes. Esta capacidad es esencial para el desarrollo y pruebas de redes 5G en modo autónomo.

- **Interoperabilidad:** Proporciona alta flexibilidad en la configuración de distintos escenarios de prueba, permitiendo ajustar los parámetros para evaluar diversos casos de uso y comportamientos de red. Además, puede integrarse fácilmente con otras plataformas y herramientas, ofreciendo una visión más completa del rendimiento de la red.

2.1.2.1.2 OpenAirInterface

OpenAirInterface [7] (OAI) es un conjunto de implementaciones de software diseñado para experimentar con redes móviles, incluyendo 4G y 5G. Proporciona una plataforma abierta y accesible que facilita la simulación, prueba y validación de tecnologías de redes móviles.

Entre sus características destacan:

- **Cumplimiento con estándares 3GPP**: Se basa en las especificaciones técnicas y los estándares desarrollados por el 3GPP.

- **Flexibilidad:** La plataforma permite modificar una amplia variedad de parámetros, lo que posibilita la creación de distintos escenarios para su posterior análisis.

- **Compatibilidad con hardware:** Puede integrarse tanto con hardware físico como con simulaciones, y también con otras herramientas y sistemas para realizar evaluaciones más completas.

2.1.2.1.3 Free5GC

Free5GC [8] es un software de código abierto que ofrece una plataforma flexible y accesible para experimentación con redes 5G permitiendo probar diversas funcionalidades y configuraciones de la red 5G.

Entre sus características destacan:

- **Cumplimiento con estándares 3GPP**: Se basa en las especificaciones técnicas y los estándares desarrollados por el 3GPP.

- **Arquitectura completa de Core 5G:** proporciona una implementación completa de los componentes del core de la red 5G.

- **Modularidad y escalabilidad:** permite a los usuarios adaptar y escalar diferentes componentes según sus necesidades específicas, facilitando la personalización de distintos escenarios de prueba.

- **Interoperabilidad:** ofrece la capacidad de interoperar con diferentes simuladores y herramientas, facilitando la integración con otros sistemas.

2.1.2.1.4 Open5GS

Open5GS [9] es una implementación de código abierto del núcleo de la red 5G que proporciona una solución completa para el core de la red 5G, permitiendo experimentar con diversas funcionalidades en entornos de investigación y desarrollo.

Entre sus características destacan:

- **Cumplimiento con estándares 3GPP**: se basa en las especificaciones técnicas y los estándares desarrollados por el 3GPP.

- **Arquitectura modular**: está compuesto por varios módulos separados representando diferentes funciones del core de la red, permitiendo flexibilidad en la configuración y escalabilidad.

- **Interoperabilidad e integración:** ofrece capacidad para integrarse con diferentes herramientas y plataformas, así como con otros componentes de la red, como estaciones base y simuladores de red.

2.1.2.1.4.1 Conclusión

Se ha seleccionado UERANSIM debido a su destacada capacidad para simular tanto los equipos de usuario (UE) como las estaciones base (gNodeB), lo que permite una evaluación detallada de la interacción entre estos componentes. Además, su alta flexibilidad en la configuración de diversos escenarios de prueba lo hace especialmente adecuado para los requisitos del proyecto.

También se ha optado por Free5GC debido a su robustez en la implementación del núcleo de la red 5G. Este software cumple con los estándares definidos por el 3GPP y proporciona una plataforma abierta que facilita la experimentación con distintas funcionalidades del núcleo de la red 5G.

2.2 Cloud computing

2.2.1 Introducción a la computación en la nube

La comercialización de servidores físicos conlleva una inversión económica considerable. Dichos servidores requieren ubicarse en espacios adecuadamente acondicionados y supervisados, y su mantenimiento debe ser efectuado por personal especializado. [10]

La nube se configura como una red de servidores distribuidos a nivel mundial, operativos de manera ininterrumpida, que funcionan como un único ecosistema para el almacenamiento de diversos tipos de datos, programas y plataformas tecnológicas. Así, se elimina la necesidad de guardar toda esta información, software o sistemas en el disco duro local, ya que, al estar hospedados en esta red interconectada de servidores, se facilita el acceso a la información desde cualquier ubicación o dispositivo.

Por consiguiente, la nube representa una mejora significativa en la modalidad de trabajo, así como en la eficacia y productividad tanto para usuarios individuales como para empresas. Además del almacenamiento en la nube, esta tecnología ha facilitado otros usos comunes y ventajosos para las empresas, tales como aplicaciones en la nube, máquinas virtuales que simulan el funcionamiento de un ordenador e infraestructuras en la nube, así como progresos como el Business Intelligence [11].

El avance de la computación en nube ha posibilitado la creación de ecosistemas virtuales que facilitan el almacenamiento de una extensa variedad de datos, archivos y programas sin la necesidad de instalarlos en dispositivos propios. El usuario se beneficia al no tener que alojar este software y datos en sus equipos locales, lo que reduce los costos asociados con el mantenimiento, la gestión y la seguridad de estos servicios. Actualmente, existe una amplia gama de proveedores de servicios en la nube disponibles para datos y aplicaciones.

Las empresas más destacadas en este sector son [12] [13] [14]:

1. **Amazon Web Services (AWS)**: AWS se posiciona como una de las plataformas líderes en el sector de la computación en la nube, ofreciendo una amplia gama de servicios que incluyen desde almacenamiento y procesamiento hasta bases de datos, análisis, aprendizaje automático, Internet de las cosas (IoT) y una variedad de soluciones adicionales.

3. **Microsoft Azure**: Azure constituye la plataforma en la nube proporcionada por Microsoft, la cual ofrece un abanico de servicios equiparables a los de AWS, que incluyen desde infraestructura como servicio (IaaS), plataforma como servicio (PaaS) hasta software como servicio (SaaS). Azure se caracteriza por su integración con las herramientas y tecnologías desarrolladas por Microsoft.

4. **Google Cloud Platform (GCP)**: GCP es la plataforma de computación en la nube ofrecida por Google, que proporciona una extensa gama de servicios que incluyen almacenamiento y procesamiento, bases de datos, aprendizaje automático, análisis de

datos, IoT, entre otros. Google se distingue por su experiencia en infraestructura y su enfoque en herramientas de inteligencia artificial y aprendizaje automático.

5. **IBM Cloud**: IBM ofrece una diversidad de servicios en la nube, que incluyen infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS). IBM Cloud se distingue por su enfoque en soluciones para empresas y su capacidad para integrarse con tecnologías híbridas y entornos multicloud.

2.2.2 Almacenamiento en la nube

Dentro de la computación en la nube, uno de los servicios más demandados es el del almacenamiento. Existen varias empresas especializadas en ofrecer el servicio de almacenamiento en la nube, este servicio permite:

- Guardar grandes volúmenes de datos e información a un coste inferior que si fuesen servidores físicos.
- Mantener tus archivos seguros ya que estas empresas cuentan con sistemas de seguridad más avanzados y actualizados.
- Compartir archivos en tu almacenamiento en la nube con otros usuarios y mantener los archivos sincronizados.
- Fácil acceso a tus archivos a través de una web, aplicación para pc o móvil.

2.2.3 Tipos de nube

Para poder realizar una buena elección a la hora de elegir un entorno Cloud, es necesario conocer los distintos tipos de nube que existen y cómo se clasifican.
Esto permitirá que escojamos el entorno que más se adapte a nuestras necesidades y podamos reducir costos y recursos al comprender cuales son nuestras necesidades.
Así, se pueden distinguir cuatro tipos de nube:

Nube privada

Es un conjunto de servidores conectados entre sí que están dedicados exclusivamente a una única organización, lo que significa que todos sus recursos informáticos son utilizados únicamente por esta organización y no se comparte con usuarios que no pertenezcan a esta.
Esta interconexión de recursos se realiza a través de una red privada.
Puede ser implementada y mantenida en las instalaciones de la empresa (on-premise) o puede ser gestionada por un proveedor de servicios externo.
Ofrece mayor control sobre los recursos y la seguridad, ya que la infraestructura está dedicada exclusivamente a la organización.

Esta opción se suele utilizar cuando se necesita un mayor control sobre los datos o aplicaciones que puedan tener requisitos de seguridad específicos y que no se podían cumplir en la nube pública. [15]

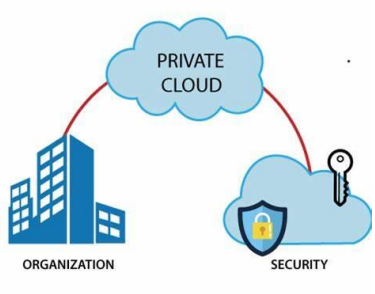

Ilustración 2: Nube privada [15]

Nube pública

Es un conjunto de servidores conectados entre sí que están gestionados por un proveedor de servicios en la nube.
Esta infraestructura es compartida por múltiples organizaciones y cada usuario paga por el almacenamiento y recursos que necesite.
Esto permite una mayor eficiencia y economía de escala ya que las organizaciones pueden aumentar o reducir la capacidad de computación según sea necesario sin tener que realizar inversiones en infraestructura física.
Además, estos únicamente pagan por los recursos que se utilicen que serán accesibles a través de internet desde cualquier lugar del mundo. [16]

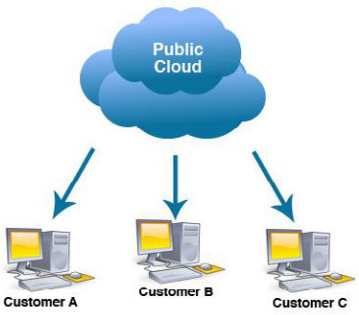

Ilustración 3: Nube pública [16]

Nube híbrida

Se trata de la combinación de infraestructura de nube privada y nube pública de forma que algunos de los sistemas o recursos están en infraestructura local (on-premise) y otras en nubes públicas con servicios ofrecidos por un proveedor, esto permite que las organizaciones puedan mover datos y aplicaciones entre ambos entornos según sea necesario.

En muchas ocasiones la arquitectura principal se encuentra en la nube privada pero se utiliza la nube pública como backup y tener una copia de seguridad.
Esto proporciona flexibilidad para aprovechar los beneficios de la nube pública, como la escalabilidad y la economía, mientras se permite a las organizaciones cumplir con las regulaciones específicas sobre datos y aplicaciones críticas. [17]

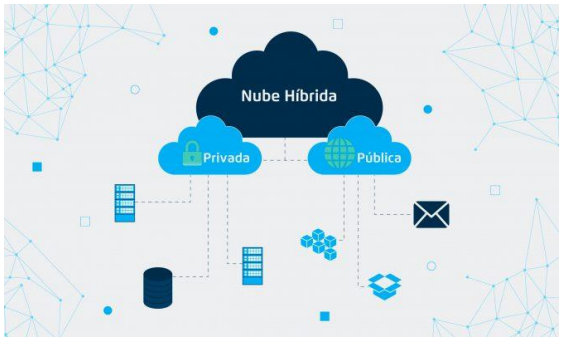

Ilustración 4: Nube híbrida [17]

Nube Comunitaria

Una nube comunitaria es una infraestructura en la nube compartida por varias organizaciones que tienen intereses comunes, estos pueden ser requisitos de cumplimiento normativo, seguridad o necesidades de la industria.
Las organizaciones pertenecientes a una misma comunidad comparten infraestructura de nube centralizada y pueden acceder a recursos informáticos compartidos, como servidores, almacenamiento y aplicaciones, a través de una red privada.
Estas organizaciones colaboran para construir y gestionar la nube, compartiendo recursos y costos de manera más eficiente mientras mantienen un alto nivel de seguridad y control.
Aunque los recursos son compartidos también se pueden satisfacer las necesidades específicas de cada organización que lo forma permitiendo que sea más flexible y ágil a la hora de implementar soluciones. [18]

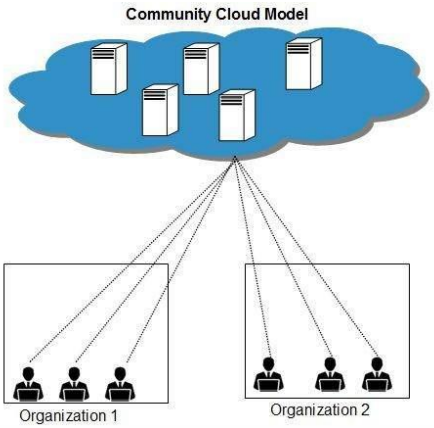

Ilustración 5: Nube comunitaria [19]

Mulitinube

Es una arquitectura que consiste en el uso de varios servicios o plataformas de computación por parte de una organización de distintos proveedores de nube.
Estos servicios pueden incluir servicios de nube pública, privada e híbrida, esto depende de las necesidades o requisitos que deba cumplir la organización.
De esta manera las organizaciones pueden reducir su dependencia con un único proveedor y diversificar el riesgo con cualquier interrupción del servicio, además pueden elegir de cada proveedor los servicios que más les interesen y se adapten a sus necesidades, adaptando también los costos. [20]

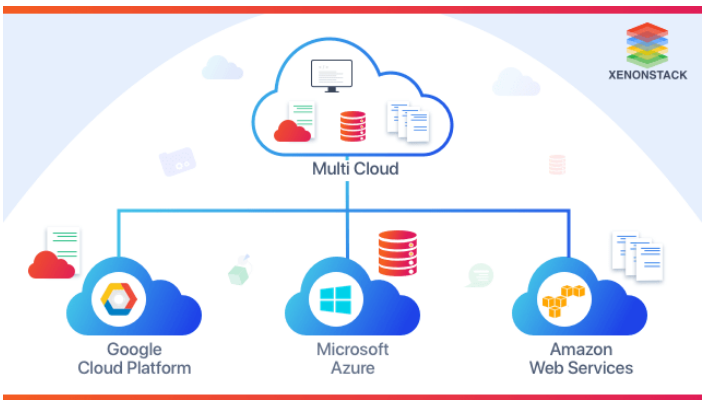

Ilustración 6: Multinube [20]

2.2.4 Modelos de servicios en la nube

Infraestructura como Servicio (IaaS - Infrastructure as a Service):

En este modelo, los usuarios tienen acceso a través de internet a distintos recursos como pueden ser máquinas virtuales, almacenamiento y redes.

La Infraestructura como Servicio (IaaS) permite ajustar rápidamente los recursos según la demanda, lo que implica pagar solo por lo utilizado. Elimina la necesidad de adquirir y gestionar servidores físicos, lo que reduce costos y complejidad. Los recursos se ofrecen como servicios separados, lo que permite alquilar solo lo necesario por el tiempo requerido. Sus ventajas incluyen la eliminación de gastos de capital, reducción de costos operativos, capacidad de innovar rápidamente y mejorar la seguridad.

Plataforma como Servicio (PaaS - Platform as a Service):

En este modelo los usuarios tienen un entorno para la creación y el lanzamiento de aplicaciones, incluyendo herramientas de desarrollo, sistemas operativos, middleware y servicios de base de datos.

Esto permite poder desarrollar, desplegar y gestionar aplicaciones sin tener que lidiar con la infraestructura técnica subyacente como el hardware o el sistema operativo.

Software como Servicio (SaaS - Software as a Service):

En este modelo, los proveedores de la nube ofrecen aplicaciones completas de software alojadas y accesibles a través de Internet.

Los usuarios pueden acceder a estas aplicaciones utilizando un navegador web y desde sus dispositivos, sin necesidad de instalar o mantener software adicional en el propio dispositivo desde el que se accede.

Algunos ejemplos de servicios típicos SaaS son el correo electrónico como el Gmail, Microsoft Office 365, etc.

Estos servicios suelen darse a modo de subscripción, en el que los usuarios pagan una tarifa por el uso de ese software.

Cada tipo de servicio en la nube ofrece diferentes niveles de control y responsabilidad compartida entre el proveedor de la nube y el usuario, lo que permite a las organizaciones elegir el modelo que mejor se adapte a sus necesidades específicas de negocio y de tecnología. [21] [17]

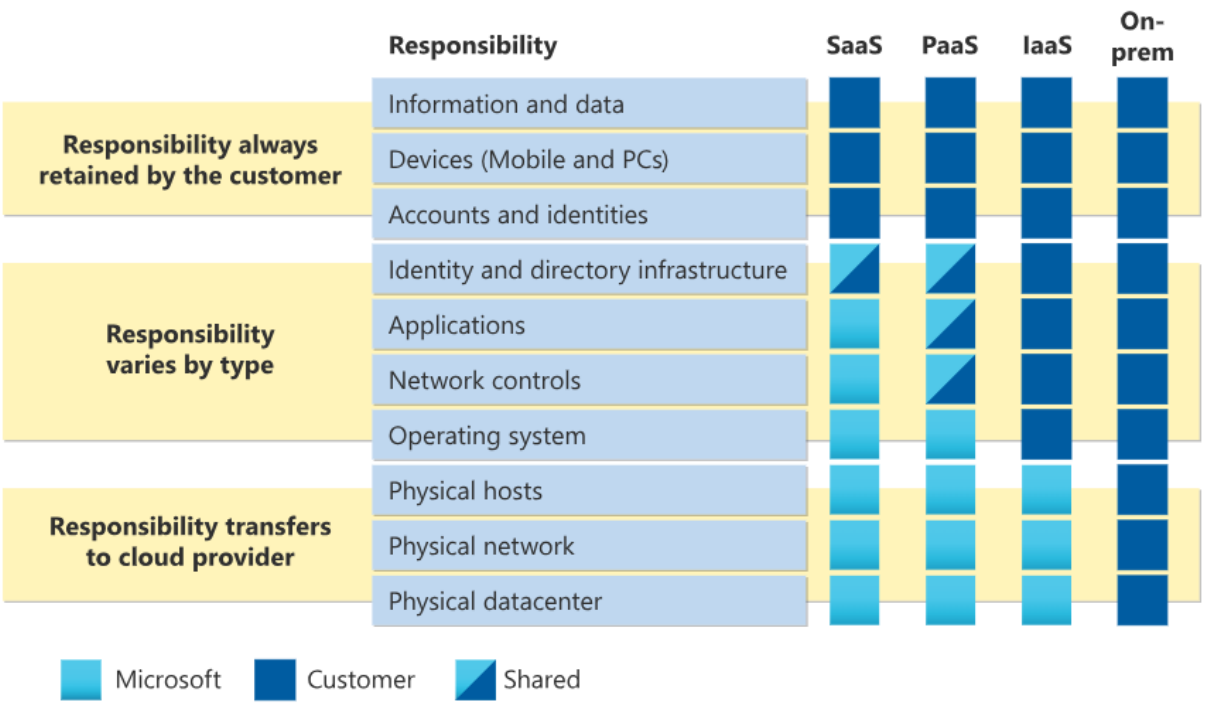

Ilustración 7: Modelo de responsabilidad compartida Microsoft [22]

También existe otro modelo de responsabilidad compartida que es:

Funciones como servicio (FaaS) [23]

Este modelo también se le conoce por "serverless computing", en este modelo los usuarios son capaces de ejecutar código en respuesta a eventos específicos sin tener que preocuparse por la infraestructura subyacente ya que esta es gestionada por el proveedor de la nube y los usuarios solo pagan por el tiempo de ejecución de su código.

El modelo de responsabilidad compartida es un concepto fundamental en la computación en la nube que define las responsabilidades tanto del proveedor de la nube como del usuario. El espacio físico, la seguridad y el mantenimiento o reemplazo de los servidores son responsabilidades típicas de una empresa en un centro de datos corporativo. El departamento de TI está a cargo de mantener toda la infraestructura y el software necesarios para que el centro de datos funcione correctamente.

El modelo de responsabilidad compartida divide estas responsabilidades entre el consumidor y el proveedor de la nube. El proveedor de la nube es responsable de la seguridad física, la energía, la refrigeración y la conectividad de la red. Mientras tanto, el consumidor es responsable de la seguridad de los datos y la información almacenados en la nube.

La situación también determina la responsabilidad. Por ejemplo, si se utiliza una base de datos SQL en la nube, el consumidor sería responsable de los datos ingresados en la base de datos, pero el proveedor de la nube sería responsable del mantenimiento de la base de datos en sí. El consumidor sería responsable de actualizar la base de datos y mantener los datos almacenados en ella en el caso de una máquina virtual con una base de datos SQL instalada.

Los tipos de servicios en la nube infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS) están estrechamente relacionados con el modelo de responsabilidad compartida. IaaS se encarga principalmente del consumidor, mientras que SaaS se encarga principalmente del proveedor de la nube. PaaS, como término intermedio entre IaaS y SaaS, divide las responsabilidades entre el proveedor de la nube y el consumidor de manera equitativa.

2.2.5 Ventajas de la nube

Son muchas las ventajas que tienen los usuarios de la nube y el cloud computing, algunas de ellas son [24]:

- Reducción de gastos: con el uso de la nube, las organizaciones ya no necesitan comprar sus propios servidores y mantener la arquitectura necesaria, además que se les permite almacenar sus datos, programas y recursos.

Con los servicios de cloud computing, las empresas no necesitan contratar infraestructuras cuando hay picos de demanda porque pagan a los proveedores solo por los recursos utilizados.

- Escalabilidad: dependiendo de las necesidades del usuario o de la organización en el momento, estos servicios permiten la escalada rápida y fácil de los recursos.
  Al escalar, puede aumentar o reducir la capacidad de almacenamiento, la potencia de procesamiento u otros recursos sin tener que adquirir hardware adicional por parte del usuario u organización. Esto les permite adaptarse rápidamente a los cambios y la demanda del momento.

- Siempre accesibles: además de que los proveedores de servicio en la nube suelen hacer copias de seguridad regularmente, se puede acceder a programas y aplicaciones siempre que se esté conectado a internet, independientemente del dispositivo desde el que se acceda y su ubicación.

- Trabajo colaborativo: Permite que distintos usuarios estén realizando cambios en un mismo documento de manera simultánea.

- Mayor seguridad: Las actualizaciones se realizan de forma inmediata por lo que ni la empresa ni el usuario tienen que preocuparse por actualizar estos programas.

2.2.6 Plataformas comerciales de Cloud Computing

Una correcta elección del entorno Cloud para llevar a cabo un proyecto es una decisión muy importante ya que una correcta elección afecta a los costos operativos, la disponibilidad, rendimiento, seguridad y funcionalidades de sus aplicaciones.

A través de un análisis detallado de los precios y características de los servicios ofrecidos por AWS, Microsoft Azure, Google Cloud Platform (GCP) e IBM Cloud, este estudio busca proporcionar una visión comprensiva de las opciones disponibles para las organizaciones que consideran la adopción o la optimización de su infraestructura en la nube. Además de evaluar los costos asociados con diferentes categorías de servicios en la nube, como cómputo, almacenamiento y bases de datos, se explorará cómo el modelo de responsabilidad compartida entre el proveedor y el consumidor puede influir en los costos totales y en la estrategia de adopción.

Además, este estudio pretende sentar las bases para futuras investigaciones y análisis en el campo en constante evolución de la computación en la nube.

2.2.6.1 Azure Cloud

Es una plataforma de servicios en la nube proporcionada por Microsoft. Los modelos de precios por servicios pueden variar dependiendo del mismo y la región geográfica en la que se encuentre, algunos de los modelos de precios más comunes son:

- Pago por uso: en este modelo, los usuarios pagan únicamente por recursos que consumen de modo que pagan por el tiempo de uso de una máquina virtual, cantidad de almacenamiento usado o cantidad de datos transferidos.

- Suscripciones mensuales o anuales: ofrece servicios de suscripción que permiten a los usuarios pagar una tarifa fija mensual por los recursos que consumen o por un grupo de recursos. Este modelo puede llegar a tener descuentos o beneficios en comparación con el pago por uso.

- Precios fijos: algunos servicios de Azure pueden tener una tarifa fija lo que significa que si un usuario quiere utilizar esos recursos tiene que pagar un precio fijado independientemente de las horas de uso que se le vaya a dar.

- Precios basados en transacciones: algunos recursos de Azure como por ejemplo las bases de datos se cobran a los usuarios dependiendo del número de consultas ejecutadas o cantidad de datos almacenados.

- Descuentos por compromisos: Azure ofrece descuentos si los usuarios se ofrecen a largo plazo a utilizar ciertos servicios de Azure durante un determinado periodo de tiempo específico.

2.2.6.1.1 Redes 5G privadas en Azure

Actualmente las ofertas de 5G privado para empresas al ser orientado para robótica, Inteligencia artificial de vídeo IoT a gran escala, etc van más allá de la arquitectura de red de telecomunicaciones tradicional, se usa la nube de hiperescala y el MEC [25].

- **Nube de hiperescala** [26]: se refiere a la infraestructura en cloud pensada para poder ser escalable horizontalmente pudiendo llegar de este modo a altos rendimientos, procesamientos y redundancia para potenciar la tolerancia a fallos y alta disponibilidad.
- **MEC** [27]: son extensiones de Azure y se usan para ejecutar cargas de trabajo que requieren una latencia baja ya que se conectan a la red móvil.

Para el despliegue de un nodo 5G, se requiere como mínimo una máquina virtual con 8 GiB de RAM y 2 vCPU. Los costos mensuales por cada máquina virtual pueden consultarse utilizando la calculadora de precios de Azure.

Instancia	vCPU	RAM	Almacenamiento temporal	Pago por uso con AHB	Plan de ahorro de 1 año con AHB	Plan de ahorro de 3 año con AHB	Al contado con AHB	Agregar a estimación
B2ts v2	2	1 GiB	0 GiB	€7.6715/mes	€5.8276/mes ~24% savings	€4.0645/mes ~47% savings	€1.9179/mes ~75% savings	+
B2ls v2	2	4 GiB	0 GiB	€30.6858/mes	€23.3239/mes ~23% savings	€16.2648/mes ~46% savings	€7.6715/mes ~75% savings	+
B2s v2	2	8 GiB	0 GiB	€61.3717/mes	€46.6411/mes ~24% savings	€32.5297/mes ~46% savings	€15.3429/mes ~75% savings	+
B4ls v2	4	8 GiB	0 GiB	€109.0155/mes	€82.8518/mes ~24% savings	€57.7782/mes ~47% savings	€27.2539/mes ~75% savings	+

Ilustración 8: Coste máquina virtual Azure

2.2.6.2 IBM Cloud

Combina una plataforma como servicio PaaS con la infraestructura como servicio IaaS para proporcionar una experiencia integrada. Disponible en centro de datos de todo el mundo, con regiones multizona en América del Norte y del Sur, Europa, Asia y Australia, está habilitado para desplegar localmente con escalabilidad global.

IBM cloud tiene dos tipos de facturación [14]:

- Pago por uso: Se cobrará al usuario cada mes dependiendo del gasto de recursos. Cuando se desea explorar el catálogo de IBM Cloud y hay cargas de trabajo pequeñas, este tipo de facturación es ideal.

- Cuentas de subscripción: Al pagar una subscripción, se le permite usar los recursos de la nube de IBM durante un período de tiempo específico y recibe un descuento en el costo de uso. Cuando hay grandes cantidades de trabajo en la nube, es mejor tener una planificación financiera para este tipo de facturación.

2.2.6.2.1 Redes 5G privadas en IBM

IBM está trabajando para incorporar la tecnología 5G a su oferta de servicios en la nube para brindar a sus clientes capacidades más avanzadas.

Algunas de las medidas que están tomando incluyen la integración de la computación de borde para reducir la latencia y mejorar la velocidad de respuesta de las aplicaciones; implementar el corte de red para ofrecer servicios de red personalizados a diferentes clientes; y desarrollar aplicaciones comerciales y de consumo habilitadas para 5G.

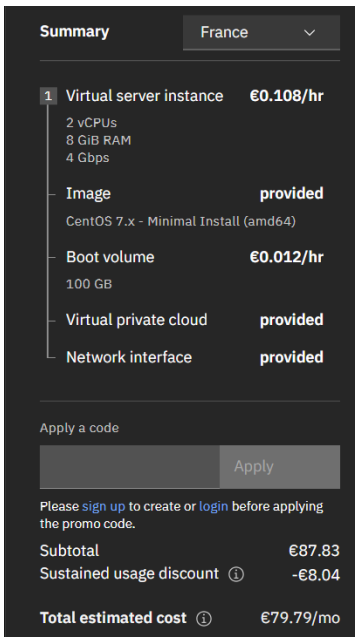

Ilustración 9: Coste máquina virtual IBM Cloud

2.2.6.3 Google Cloud

Es una plataforma que ofrece a sus usuarios más de 90 servicios y está disponible en más de 200 países y territorios. Esta amplia gama de herramientas incluye aplicaciones de inteligencia artificial, procesamiento de datos y hosting. Estos servicios se pueden utilizar según las necesidades de cada usuario o empresa [28].

Las ventajas de GCP incluyen una infraestructura escalable y flexible que permite ajustar los recursos según las necesidades del negocio, herramientas de innovación avanzadas como análisis de datos y aprendizaje automático, y un modelo de pago por uso que permite a las empresas optimizar sus recursos y probar los servicios de forma gratuita.

Google Cloud tiene dos tipos de facturación [29]:

- Plan flexible: solo se cobra a los usuarios por las cuentas que utilicen cada mes.

  En cualquier momento se pueden agregar y quitar cuentas de usuario, también cancelar la suscripción sin pagar nada.

- Plan anual o de duración fija: El usuario se compromete a usar el servicio durante un año o más, el número de licencias será lo que determine el precio.
  Se pueden llegar a comprar más licencias si el equipo creciese, lo que aumentaría el precio, solo se podría reducir el número de licencias cuando el plan se renovase al final del contrato.

No se recibe ningún reembolso si se cancela durante el periodo de compromiso.

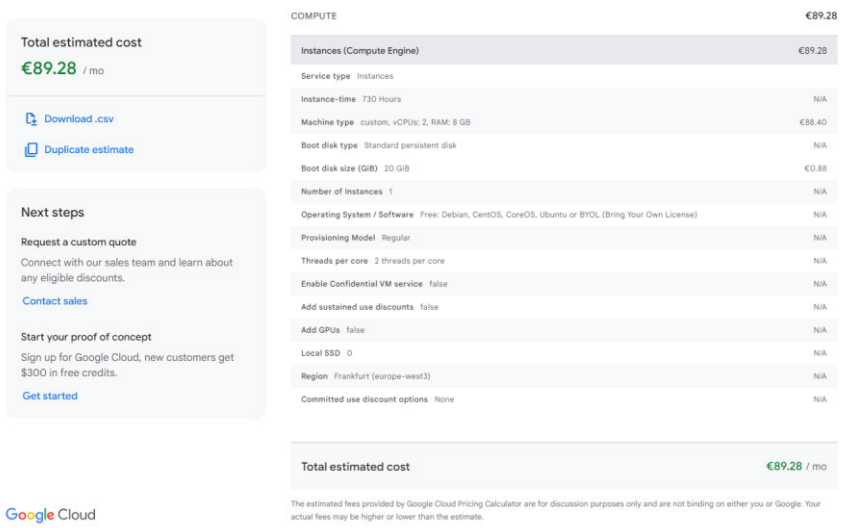

Ilustración 10: Coste máquina virtual Google Cloud

2.2.6.4 AWS Cloud

AWS [30] tiene una amplia gama de servicios y funcionalidades que superan a las de otros proveedores de la nube. Proporciona desde tecnologías de infraestructura básicas como cómputo, almacenamiento y bases de datos hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, así como análisis de datos, lagos de datos e Internet de las cosas. Esto hace que la migración de aplicaciones existentes a la nube sea más fácil y rápida, al mismo tiempo que lo hace más rentable, lo que permite la creación de una variedad casi ilimitada de soluciones innovadoras.

Además, AWS se destaca por brindar una amplia gama de funcionalidad en sus servicios. Por ejemplo, ofrece una amplia gama de bases de datos diseñadas para diferentes tipos de aplicaciones, lo que permite a los usuarios elegir la herramienta que mejor se adapte a sus necesidades en términos de costo y rendimiento.

Los métodos de facturación que se usan en AWS son:

- Pago por hora o por segundo: en el que los servicios de AWS utilizados se cobran por hora o segundo de uso.
- Pago por capacidad: algunos servicios específicos como por ejemplo el de almacenamiento de datos, pueden llegar a facturarse en función de la capacidad que se esté utilizando o el número de transacciones.
- Modelos de precios basados en consumo: AWS ofrece opciones de precios basadas en consumo para servicios específicos como pueden ser bases de datos, transferencia de datos, análisis y servicios de inteligencia artificial.

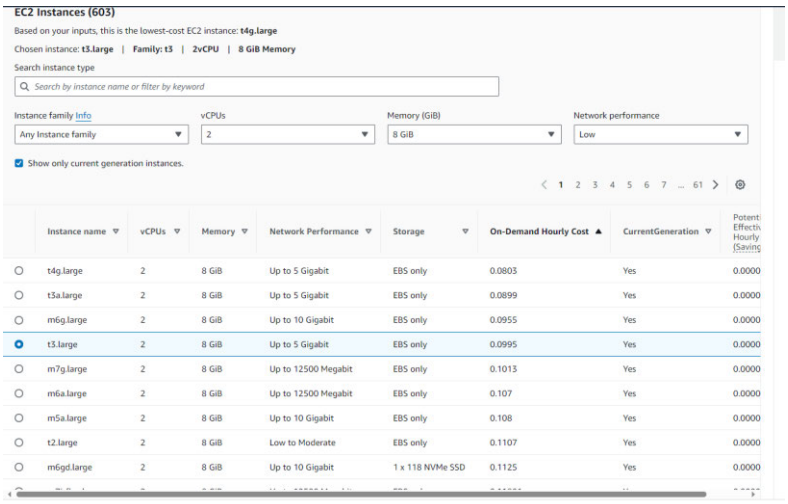

Ilustración 11: Coste máquina virtual AWS

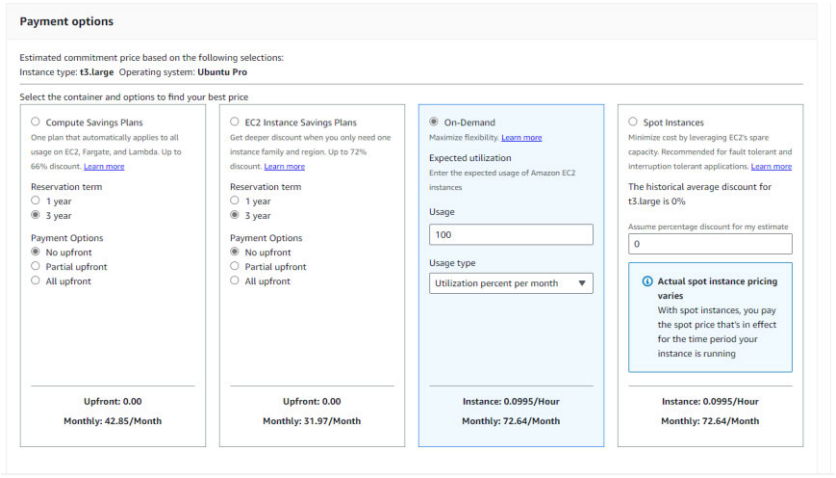

Ilustración 12: Opciones de pago AWS

2.2.7 Conclusión

Para el desarrollo de este proyecto se ha decidido utilizar Azure Cloud por varios motivos. Azure se presenta como la opción más económica en comparación con otros proveedores de servicios, ya que ofrece descuentos y créditos anuales al utilizar una cuenta de estudiantes, lo que permite optimizar los costos del proyecto. Por otro lado, IBM Cloud se orienta más hacia la automatización de redes 5G que al despliegue de estas. AWS, a su vez, muestra un enfoque dirigido a empresas y ofrece precios considerablemente más altos. Finalmente, Google Cloud, aunque habilita algunos casos de uso para 5G, tiene varias limitaciones y no proporciona una solución integral y específica para la creación y gestión completa de una red 5G.

2.3 Soluciones auxiliares de Cloud computing

2.3.1 CDN (Content Delivery Network)

Al conectarse a una página, servicio o aplicación web, se realizan peticiones a un servidor remoto. Sin embargo, las velocidades de conexión pueden ser lentas y pueden surgir problemas de disponibilidad o acceso geográfico desde ciertos países. El CDN (Content Delivery Network o Red de Distribución de Contenido) es una herramienta formada por redes de servidores distribuidos geográficamente que albergan aplicaciones, servicios o datos a los que los usuarios acceden de forma remota. [31]

2.3.2 Funcionamiento de los CDN

Los CDN funcionan almacenando copias de contenido estático, como imágenes, vídeos, archivos CSS y JavaScript, en servidores distribuidos estratégicamente en diferentes ubicaciones geográficas. Cuando se solicita un recurso, el CDN redirige la petición al servidor más cercano a la ubicación del usuario, lo que reduce la latencia y mejora el rendimiento de la carga del sitio web. Además de mejorar la velocidad y el rendimiento, los CDN también pueden proporcionar características adicionales, como la optimización de imágenes para una carga más rápida, la compresión de datos para reducir el ancho de banda utilizado y la seguridad contra ataques DDoS [32] para proteger los sitios web de amenazas maliciosas.

2.3.3 Cloudflare

Es una de las redes CDN más grandes que opera en Internet. Fundada en 2009, su misión es mejorar la experiencia del usuario al proporcionar seguridad, rendimiento y privacidad a sitios web, aplicaciones, entre otros, acercando los recursos a los usuarios finales de manera rápida y eficiente.

Su función principal es actuar como intermediario entre el servidor de un sitio web y sus visitantes, lo que implica que todos los datos que ingresan a la plataforma pasan por un proceso de filtrado y optimización antes de llegar al servidor del servicio web al cual se acceder. [33]

Las solicitudes de Internet para millones de sitios web pasan por esta red, que maneja un promedio de 50 millones de solicitudes HTTP por segundo.

Además, prioriza la privacidad mediante el cifrado de extremo a extremo y se cumple con las normativas locales sobre la localización y almacenamiento de datos. La red global es más rápida que Internet en general, con centros de datos en más de 310 ciudades en todo el mundo para ofrecer contenido rápidamente [33].

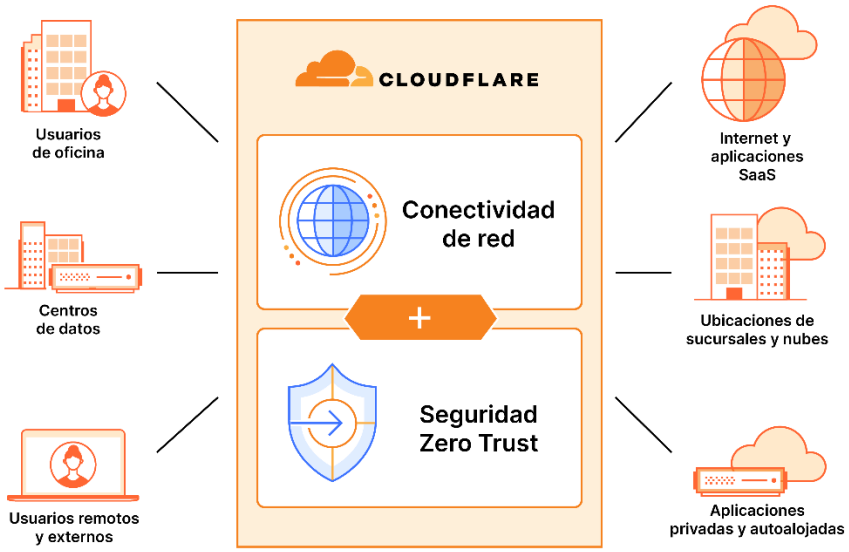

Ilustración 13: Arquitectura de Cloudflare [33]

3. Especificaciones y restricciones de diseño

3.1 Introducción

En esta sección se presenta una visión general la propuesta, enfocándose en las especificaciones y restricciones de diseño que guiarán el desarrollo del proyecto.

Se describen los requisitos funcionales que deben cumplirse, así como las limitaciones técnicas y operativas que deben tenerse en cuenta durante el diseño.

3.2 Especificaciones de diseño

A continuación, se listan las especificaciones más relevantes para la realización de la propuesta de diseño:

- El diseño propuesto se implementará utilizando Free5GC, una solución de código abierto para la arquitectura de redes 5G.

- Se usará el software UERANSIM para las tres máquinas virtuales que tendrán la función de terminales.

- Se usará el software de virtualización VirtualBox para el desarrollo y gestión de las máquinas virtuales sobre las que se implementará la red de acceso radio.

- El despliegue del núcleo 5G se realiza en una máquina virtual en Azure mediante el software Free5GC.

- La red de acceso radio se implementa en local, formado por 5 máquinas virtuales donde dos de ellas representan dos estaciones base y las otras tres son terminales.

- La conexión entre el núcleo de red y las estaciones base se realiza a través de una VPN (Virtual Private Network o Red Virtual Privada), para ello se usa el software OpenVPN. Siendo las estaciones base los clientes y el núcleo de red el servidor.

- En uno de los terminales se intalará el clienteWARP para poder usar los servicios de Cloudflare.

3.3 Restricciones de diseño

A continuación, se especifican las restricciones [34] [35] [36]:

- Para la implementación del núcleo de 5G, el número de imágenes disponibles por Azure es limitado.

- UERANSIM requiere un sistema operativo Linux Ubuntu 16.04 o superior., una versión de CMake 3.17 o superior, versión de gcc 9.0.0 o superior y versión de g++ 9.0.0 o superior.

- Para la instalación del cliente WARP se requiere una versión de Linux: CentOS, RHEL 8, Ubuntu 20.4,Ubuntu 22.04, Ubuntu 24.04, Debian 10, Debian 11, Debian 12.

- Para el correcto funcionamiento del UPF de Free5GC es necesario una versión del kernel de Linux 5.0.0-23-generic o una versión 5.4.x.

- Se requiere un sistema con al menos 8 GB de RAM (Random Access Memory o Memoria Aleatoria de Acceso) y 160 GB de memoria libre en disco para la implementación de Free5GC. Además, requiere un procesador Intel® Core™ i5 o superior.

3.4 Diseño propuesto

A continuación se muestra la arquitectura de red de la cual se parte para realizar la solución propuesta.

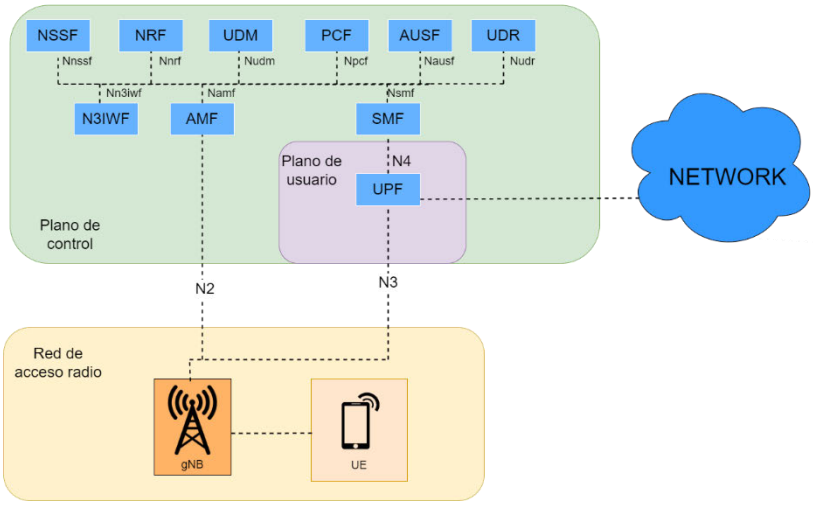

Ilustración 14: Diagrama de componentes de la red 5G

- Plano de usuario: se encarga de gestionar y transmitir los datos de usuario en tiempo real para aplicaciones como streaming, llamadas VoIP y navegación. Su función principal es transportar los datos entre los Equipos de Usuario (UE) y las redes externas. Está compuesto por la Función del Plano de Usuario (UPF), que enruta y reenvía los paquetes de datos entre la red de acceso (gNB) y la red de datos externa. La interfaz que se utiliza para la comunicación con el gNB es la N3, la cual se encarga del transporte de los datos de usuario entre el núcleo de la red y los gNB. Esta interfaz emplea el protocolo GTP-U (GPRS Tunneling Protocol – User Plane) para encapsular los paquetes de datos de usuario.

- Plano de control: El plano de control es responsable de gestionar y controlar las conexiones y el tráfico entre los terminales (UE) y el núcleo de la red. Su función incluye el establecimiento, mantenimiento y finalización de conexiones, así como la gestión de movilidad, autenticación y políticas de red. Los componentes del plano de control son: AMF, SMF, UDM, PCF, AUSF y NRF. Algunas de las interfaces involucradas son:

  -**N1**: conecta el UE con el AMF para la gestión de señalización y control entre el dispositivo y la red.

  -**N2**: conecta el gNB con el AMF para gestionar la señalización relacionada con el acceso y la movilidad.

  -**N3**: aunque forma parte del plano de usuario, es importante señalar que la interfaz N3 conecta el gNB con el UPF para transportar datos de usuario.

  -**N4**: conecta el SMF con el UPF para gestionar las políticas de sesión y la asignación de recursos en el plano de usuario.

  -**N6**: conecta el UPF con las redes externas, como internet o redes privadas, para enrutar el tráfico de datos del usuario.

  -**N8**: conecta el AMF con el UDM para obtener los datos de suscripción y perfil del usuario necesarios para la autenticación y la gestión de movilidad.

  -**N10**: conecta el UDM con el AUSF para la autenticación del usuario.

  -**N15**: conecta el PCF con el AMF y el SMF para aplicar políticas de control sobre la QoS, acceso a la red y otros parámetros.

- Red de acceso radio (RAN): es la parte encargada de conectar los dispositivos de usuario con el núcleo de la red 5G. Su función principal es gestionar las comunicaciones a través de la interfaz inalámbrica, utilizando los gNB para transmitir y recibir señales

de radiofrecuencia, permitiendo la interacción entre los dispositivos móviles y la red central. En este bloque participan las siguientes interfaces:

- **Uu**: interfaz que conecta el terminal móvil con el gNB, gestiona la transmisión de datos y señalización entre dispositivos de usuario y red.

- **F1**: interfaz que conecta la DU (Distributed Unit) con la CU (Central Unit) dentro del mismo gNB, es responsable de coordinar la transmisión de datos y control.

- **N2**: interfaz que conecta el gNB con la AMF (Accesss Mobility Management Function) en el núcleo de 5G, coordina la señalización para la gestión de la movilidad.

- **N3**: conecta el gNB con el UPF (User Plane Function), transporta los datos de usuario desde el UE hasta la red y viceversa.

4. Descripción de la solución propuesta

4.2 Introducción

En este apartado se detalla cómo se ha realizado el despliegue de la solución propuesta, mostrándose una visión general de la arquitectura de la solución, de cómo se conectan los distintos componentes entre ellos y, por último, todos los pasos hasta llegar a la solución propuesta.

4.3 Diseño general

En este apartado se describen los aspectos clave de la solución diseñada. Se ha implementado una red 5G, desplegando el núcleo de red en Azure Cloud utilizando la tecnología Free5GC. Para la conexión entre los distintos equipos, se emplea el software OpenVPN, asegurando una comunicación segura y eficiente entre los componentes de la infraestructura.

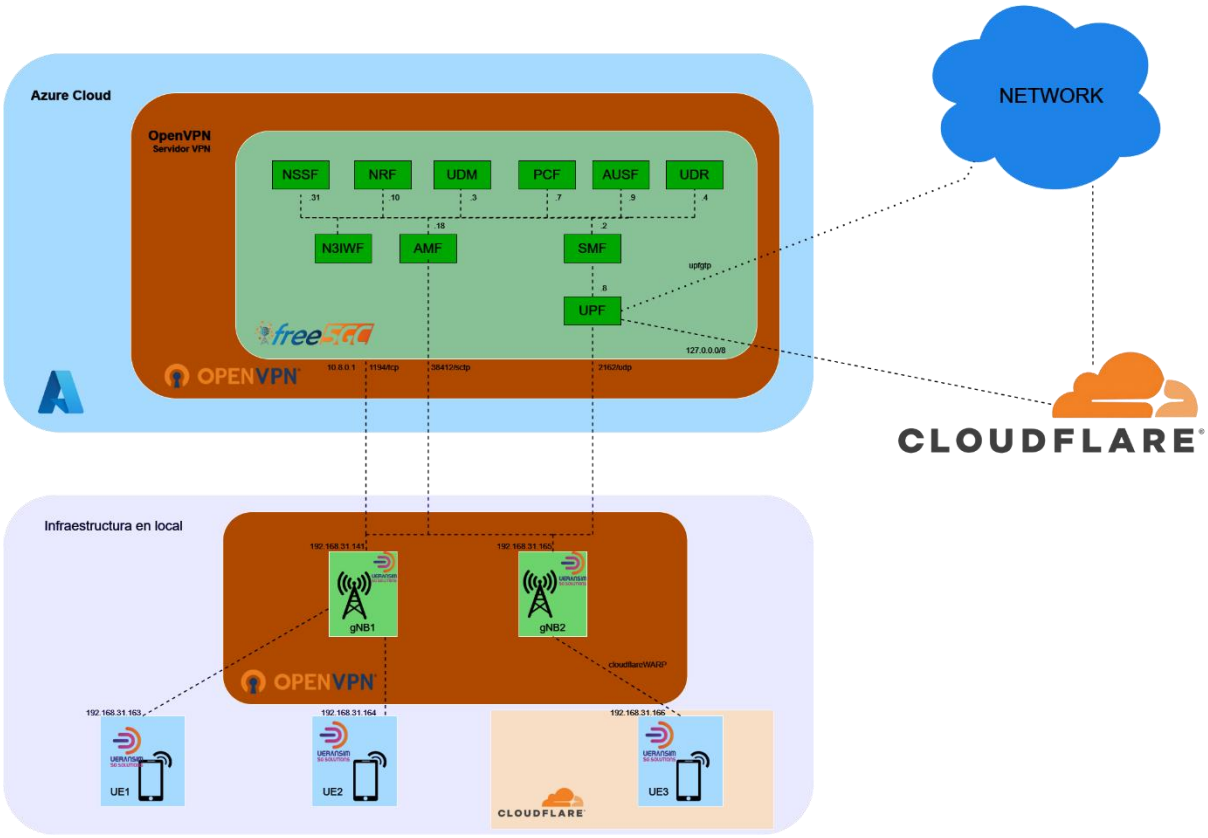

Ilustración 15: Arquitectura de la solución propuesta

A continuación se describen los elementos presentes en la arquitectura:

- Free5GC: es el núcleo de red de 5G, se divide en dos planos, el plano de usuario en el que interviene el UPF y el plano de control en el que intervienen el resto de componentes de la red.

- UERANSIM: interviene en la red de acceso radio y los terminales. Está formado por máquinas virtuales que actúan como gNB y máquinas virtuales que actúan como UEs.

- OpenVPN: las conexiones entre los gNB y el núcleo de red se hace a través de una VPN proporcionando una conexión segura y encriptada entre dispositivos. Además, facilita la creación de túneles seguros que permiten el acceso remoto a los recursos de la red, garantizando la integridad y confidencialidad de las comunicaciones.

A continuación se muestra una tabla con las direcciones IP, puertos y protocolos que se utilizan:

Tabla 1: Direcciones IP, puertos y protocolos

Nombre de la máquina virtual	Dirección IP	Puerto	Protocolo
Free5GC	10.8.0.1	38412	SCTP
		2152	UDP
gNB1	10.8.0.10	1194	TCP
gNB2	10.8.0.6	1194	TCP
UE1	192.168.31.163	1194	TCP
UE2	192.168.31.164	1194	TCP
UE3	192.168.31.166	1194	TCP

4.4 Configuraciones previas

En esta sección se detallan las configuraciones necesarias antes del despliegue de la máquina virtual que actuará de núcleo de 5G.

Para el despliegue de la máquina virtual en Azure se pueden usar varios métodos [37]:

- **Azure CLI (Command Line Interface):** mediante esta herramienta se puede crear una máquina virtual utilizando la línea de comandos. El primer paso es descargar la última versión de Azure CLI que sea compatible con el sistema operativo desde el cual se creará el recurso, y luego proceder a la instalación.

- **Azure PowerShel**l: similar a Azure CLI, es una interfaz de línea de comandos que permite administrar recursos en Azure. Se utilizar para escribir scripts que automaticen la creación y administración de máquinas virtuales.

- **Portal de Azure:** a través de este portal web se pueden crear máquinas virtuales. Es necesario iniciar sesión en la cuenta de Azure y navegar por los distintos paneles para seguir las instrucciones de creación de la máquina.

Para el desarrollo de este proyecto se utilizará el portal de Azure[1] para la creación de la máquina virtual y el resto de recursos, ya que el portal proporciona una interfaz gráfica intuitiva, lo que resulta más accesible en comparación con el uso de la línea de comandos. Esto facilita la administración de todos los recursos

Además, esta interfaz ofrece descripciones detalladas sobre los diferentes parámetros y opciones de configuración disponibles al crear una máquina virtual.

El primer paso para la creación de una máquina virtual es acceder al portal de Azure y autenticarse con el usuario y la contraseña. A continuación, se accederá a un menú de servicios donde se muestran los principales servicios más solicitados.

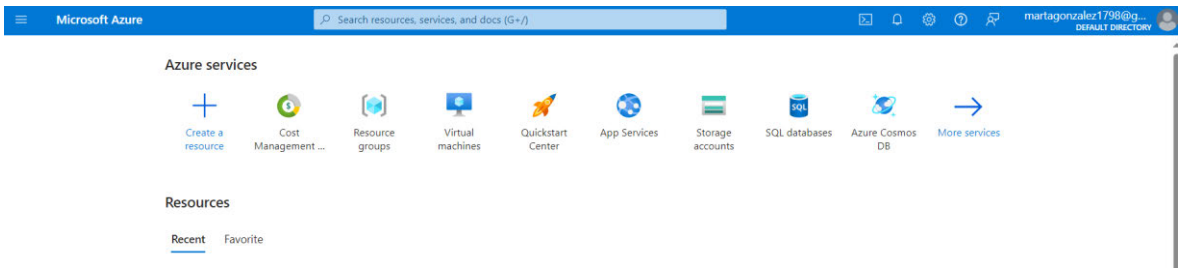

Ilustración 16: Menú servicios de Azure

Antes de crear la máquina virtual, es necesario establecer un grupo de recursos al cual asociarla.

Un grupo de recursos [38] es un contenedor lógico que agrupa recursos relacionados dentro de una solución de Azure. Su uso principal es facilitar la administración, el seguimiento y la eliminación de esos recursos como una unidad.

Para crear un grupo de recursos, se debe seleccionar el icono de "*Resource groups"* en el menú mostrado en la ilustración anterior.

[1] https://portal.azure.com/

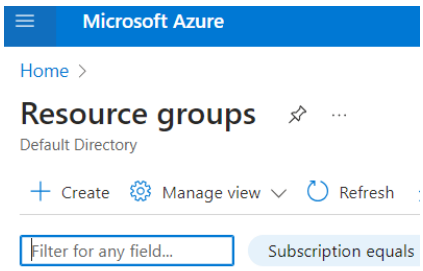

Ilustración 17: Menú Resource Groups

A continuación se selecciona la opción "*Create*", lo que despliega una pestaña en la que se define el concepto de "*Resource groups*" y se muestran los campos que se deben completar para la creación del mismo.

El primer paso es seleccionar la suscripción de Azure con la que se creará este grupo de recursos.

Se debe elegir el tipo de suscripción, en este caso, "*Azure for Students*" y un nombre para el grupo de recursos, que en éste caso se llamara "*5g*"

Asimismo, se debe seleccionar la región en la que se creará el grupo de recursos; por proximidad, el despliegue se realizará en "*France Central*".

Al crear cualquier tipo de recurso en Azure y seleccionar una región, es importante considerar que una región cercana al despliegue minimiza la latencia y mejora el rendimiento. Además, no todos los servicios de Azure están disponibles en todas las regiones.

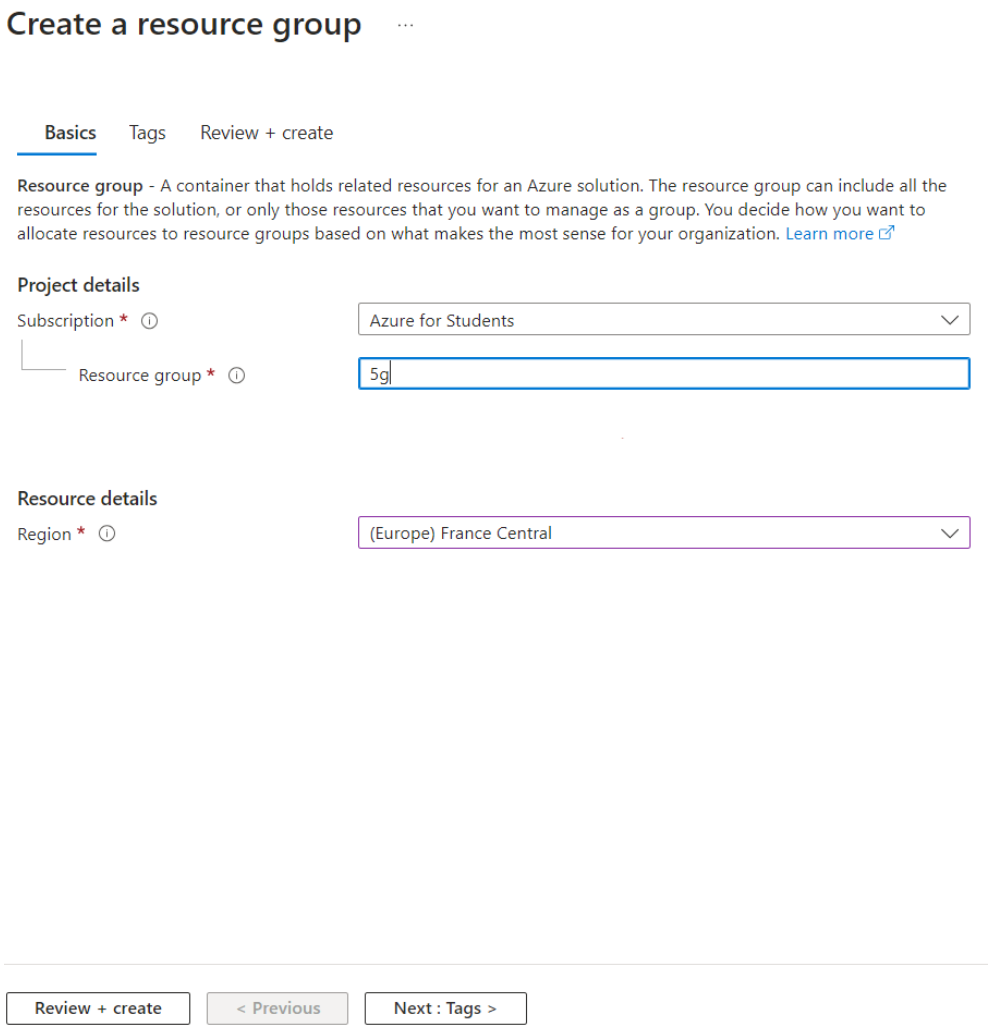

Ilustración 18: Creación de un grupo de recursos

Se pueden aplicar etiquetas (tags) para asociar recursos a categorías, lo que facilita su organización, el manejo de costos, y la búsqueda y filtrado más rápidos de recursos. Esto resulta especialmente útil en sistemas complejos con numerosos recursos. Sin embargo, en esta solución, al ser más simple, no es necesario utilizar etiquetas.

Para finalizar la creación del recurso, se selecciona "*Review + Create*".

A continuación, se crea una red virtual, ya que las máquinas virtuales requieren una red segura para comunicarse entre sí y acceder a internet, simulando que están en la misma red física.

Para crear la red virtual, se debe buscar "*Virtual Network*" en el buscador de recursos de Azure y seleccionar "*Create*".

Se mostrará un menú en el que se puede asociar el grupo de recursos a la red virtual que se está creando en la misma región. El nombre asignado será "*5g-virtual-network*".

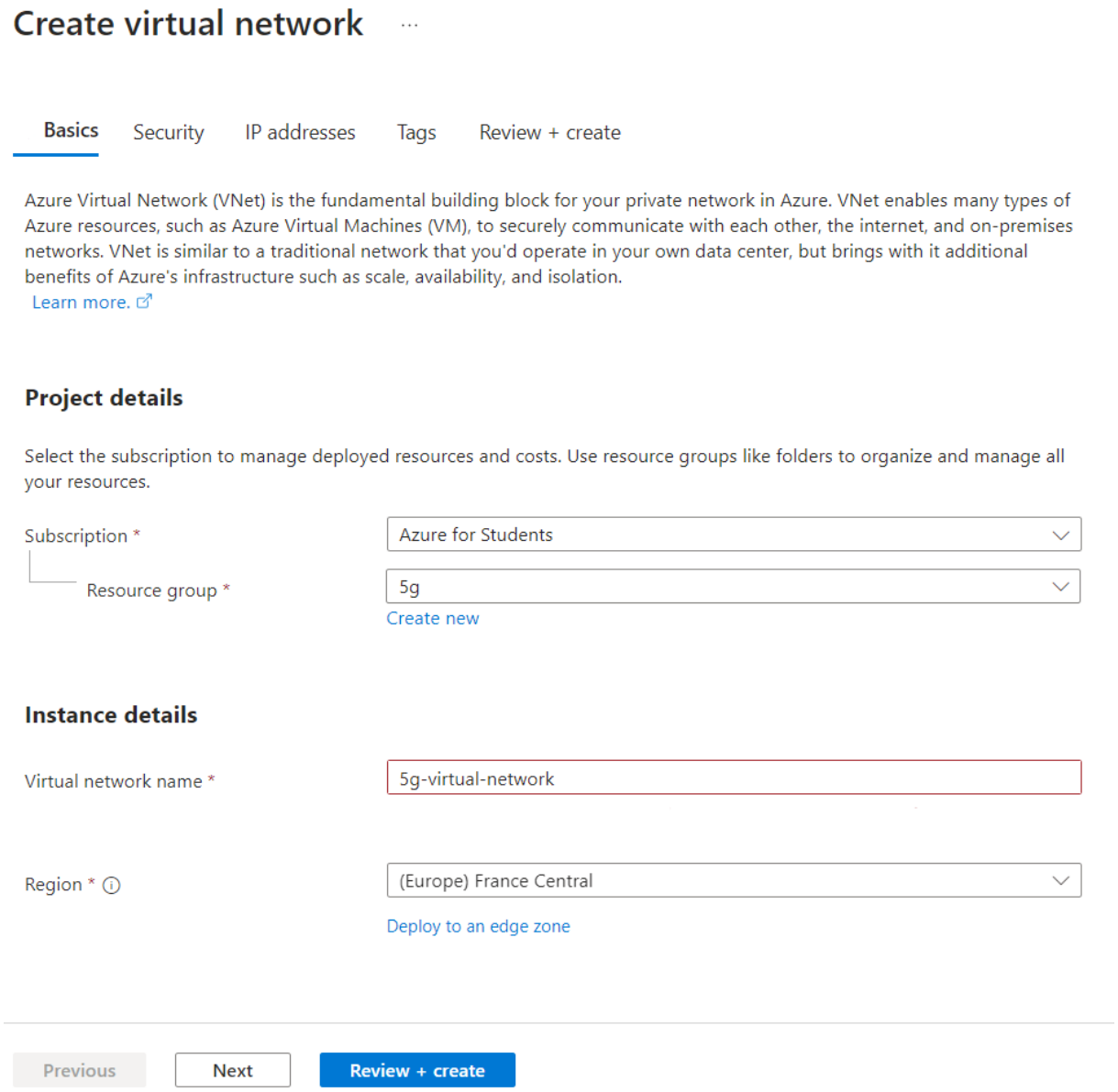

Ilustración 19: Creación de una red virtual

Se debe pulsar "*Next*" y, a continuación, se mostrará una sección relacionada con las IP. Se Luego, se selecciona "*Review + Create*".

4.5 Creación de una máquina virtual en Azure

A continuación, se procederá a crear una máquina virtual. Primero, se debe seleccionar "*Virtual Machines*" en el menú de recursos de Azure.

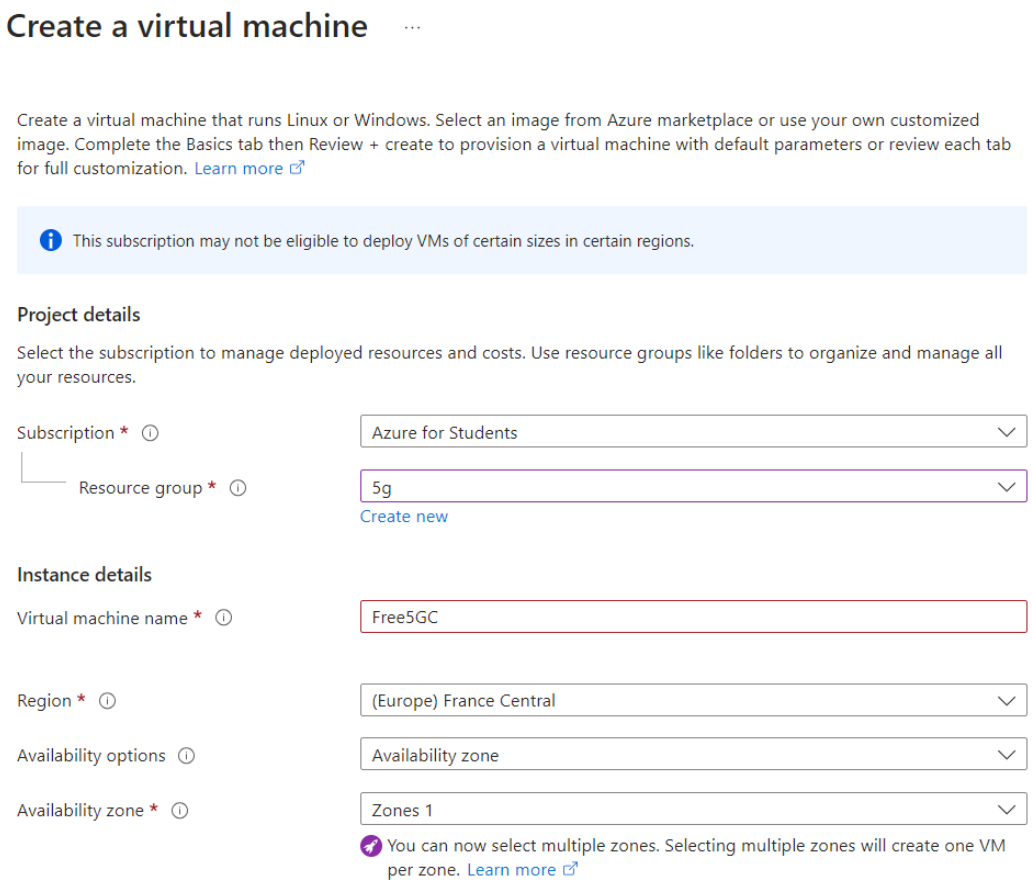

Ilustración 20: Creación de una máquina virtual

Se desplegará un menú como el de la ilustración anterior, en el cual se indicará que los sistemas operativos aceptados son Linux o Windows. Dependiendo del sistema operativo que se desee instalar, se debe seleccionar un tipo de imagen de entre las opciones disponibles. A continuación, se selecciona el grupo de recursos al que pertenece la máquina virtual, que en este caso es el grupo creado anteriormente llamado "*5g*".

El nombre de la máquina virtual será "*Free5GC*" y se asociará a la región de "*France Central*", al igual que el resto de los recursos creados anteriormente.

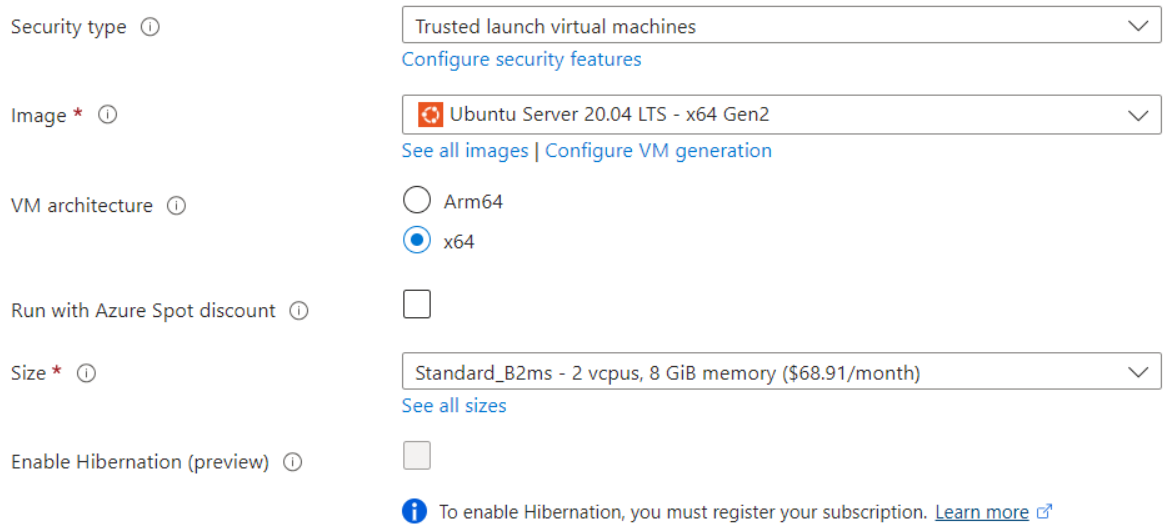

Ilustración 21: Selección de la imagen de la máquina virtual

Para que la máquina virtual que se cree sea compatible con el software Free5GC [39] es necesario que el sistema operativo instalado sea *Linux Ubuntu 18.04* o una versión superior. Además, Free5GC requiere una versión del kernel de Linux igual a 5.0.0-23-generic o 5.4.x.

Para verificar qué versiones de Ubuntu incluyen la versión de kernel necesaria, se puede consultar la página de Canonical[2], donde se encuentra el CHANGELOG, un documento que registra los cambios en el sistema operativo Ubuntu, incluidas las modificaciones, actualizaciones y mejoras realizadas.

En este documento se proporcionan detalles sobre las versiones de Ubuntu lanzadas, las nuevas características agregadas y los errores corregidos. Al revisar este documento, se observa que la última versión de Ubuntu compatible con la versión de kernel requerida para implementar Free5GC es Ubuntu 20.04.2.

En este caso, el kernel preinstalado en esta versión no es compatible con lo que se va a utilizar para implementar la solución, por lo que más adelante será necesario cambiarlo.

Por lo tanto, al crear la máquina virtual, se seleccionarán los siguientes parámetros:

- Imagen: teniendo en cuenta los requisitos necesarios, se seleccionará "*Ubuntu Server 20.04 LTS – x64 Gen2*".

- Tamaño: según el estudio anterior sobre los entornos en la nube, se necesitará una máquina virtual con 8 GiB de RAM y 2 vCPU, por lo que se seleccionará "*Standard_B2ms – 2 vcpus, 8 GiB memory*".

[2] https://canonical.com/

Al pulsar "*Next*", se cambiará de pantalla, donde se ofrecerá la opción de añadir un disco adicional a la máquina virtual que se está creando. En este caso, se añadirá un disco adicional con redundancia local de 256 GiB.

Para el resto de las configuraciones, se mantendrán las opciones predeterminadas.

4.6 Despliegue de máquinas virtuales locales

Para completar la arquitectura, se requieren cinco máquinas virtuales que se desplegarán localmente utilizando VirtualBox.

Para garantizar la compatibilidad con el software UERANSIM, el sistema operativo debe ser igual o superior a Linux Ubuntu 16.04 [40].

Por lo tanto, se descargará la versión más reciente, Linux Ubuntu Server 20.04.2 LTS[3], y se creará una máquina virtual con 2 GB de memoria RAM y 10 GB de almacenamiento, manteniendo las demás características por defecto.

Más adelante, se incrementará el espacio de almacenamiento en dos de las máquinas virtuales para realizar pruebas de seguridad en la red.

A partir de la máquina principal, denominada UERANSIM, se clonarán las demás máquinas, asignándoles los nombres correspondientes. Es importante asegurarse de que, durante el proceso de clonación, se generen nuevas direcciones MAC para evitar conflictos en la red.

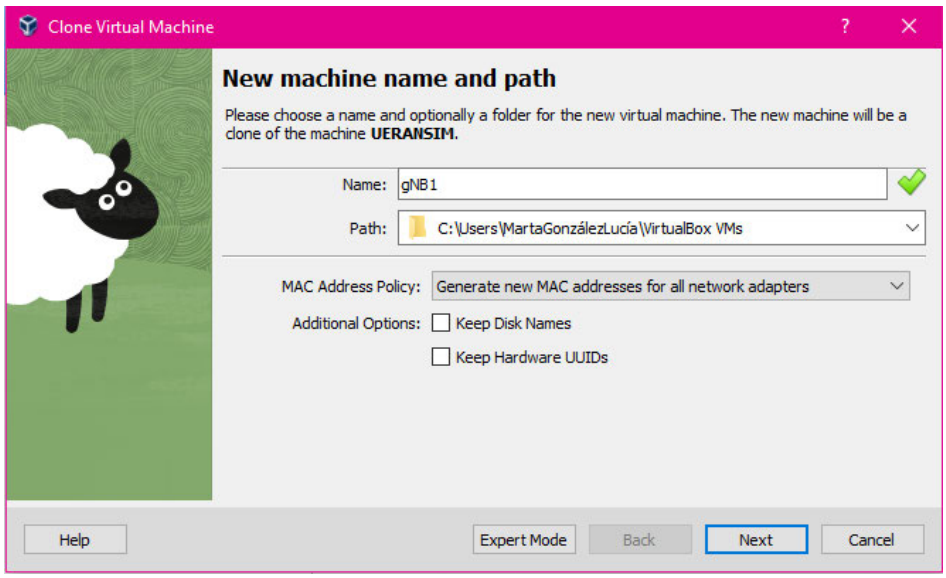

Ilustración 22: Clonación máquinas virtuales

[3] https://ubuntu.com/download/server

De esta manera hay un total de 5 máquinas virtuales, siendo una de ellas de backup como se muestra en la siguiente ilustración.

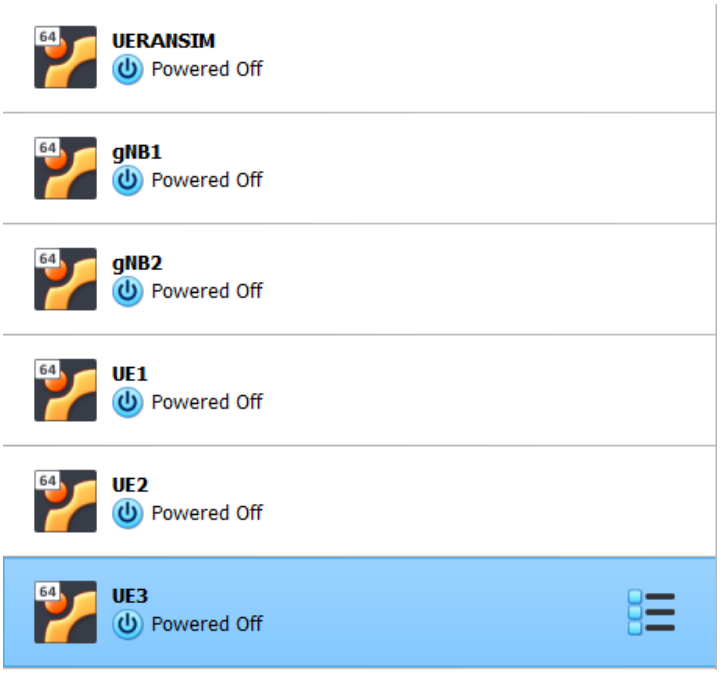

Ilustración 23: Máquinas virtuales en local

A continuación, se procederá a la actualización del software en cada máquina para asegurar que el dispositivo esté al día y protegido.

Primero, se sincroniza la lista local de paquetes con la lista de paquetes disponibles en los repositorios configurados, para garantizar que se disponga de la información más reciente.

Posteriormente, se actualiza la lista de paquetes y sus versiones, sin proceder a la instalación o actualización de ningún paquete en este paso.

```
sudo apt update
```

Luego, se instalarán las versiones más recientes de todos los paquetes instalados en el sistema que tengan actualizaciones disponibles.

```
sudo apt upgrade
```

El primer paso consiste en cambiar los nombres de los servidores de red (hostname) para que coincidan con el nombre asignado a la máquina virtual. Esto evitará confusiones, facilitará la gestión y permitirá una identificación más precisa de los servidores dentro de la red.

Para modificar el hostname de cada máquina, se debe ejecutar el siguiente comando:

```
sudo nano etc/hostname
```

Después de ejecutar el comando, es recomendable reiniciar el sistema para asegurarse de que los cambios en el hostname se apliquen correctamente.

4.6.1 Ampliación de espacio en UE3

A continuación, se ampliará el espacio de almacenamiento en la máquina UE3, ya que se instalarán el cliente WARP de Cloudflare y un navegador para realizar pruebas, lo que requerirá más espacio de almacenamiento.

Primero, en la pantalla principal de VirtualBox, se selecciona la máquina a la que se desea añadir el espacio. Luego, se accede a "*Archivo*" -> "*Herramientas*"-> "*Administrador de Medios Virtuales*".

Se selecciona la memoria que se añadirá a la máquina.

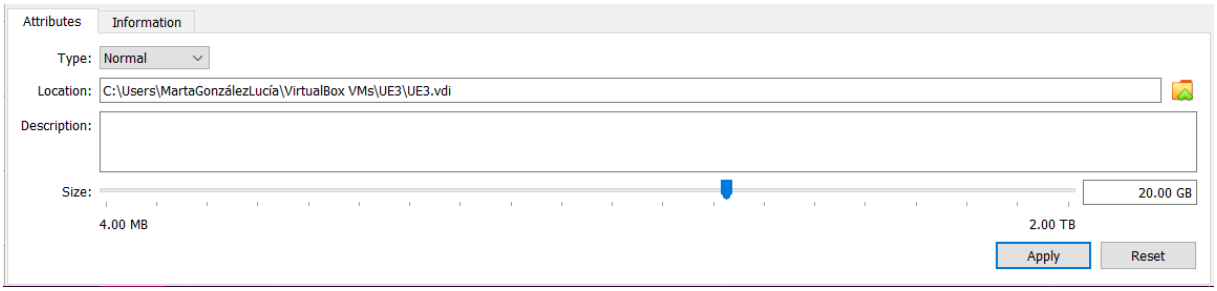

Ilustración 24: Ampliación del espacio de almacenamiento en UE3

Una vez ampliado el espacio en la máquina, será necesario asignar el nuevo espacio a la partición correspondiente para poder utilizarlo. Las particiones se pueden visualizar con el siguiente comando:

```
lsblk
```

De esta manera, se pueden visualizar todas las particiones del disco de la máquina en un formato de lista en árbol. Además, se pueden consultar detalles como el nombre del dispositivo, el tipo de disco y el tamaño.

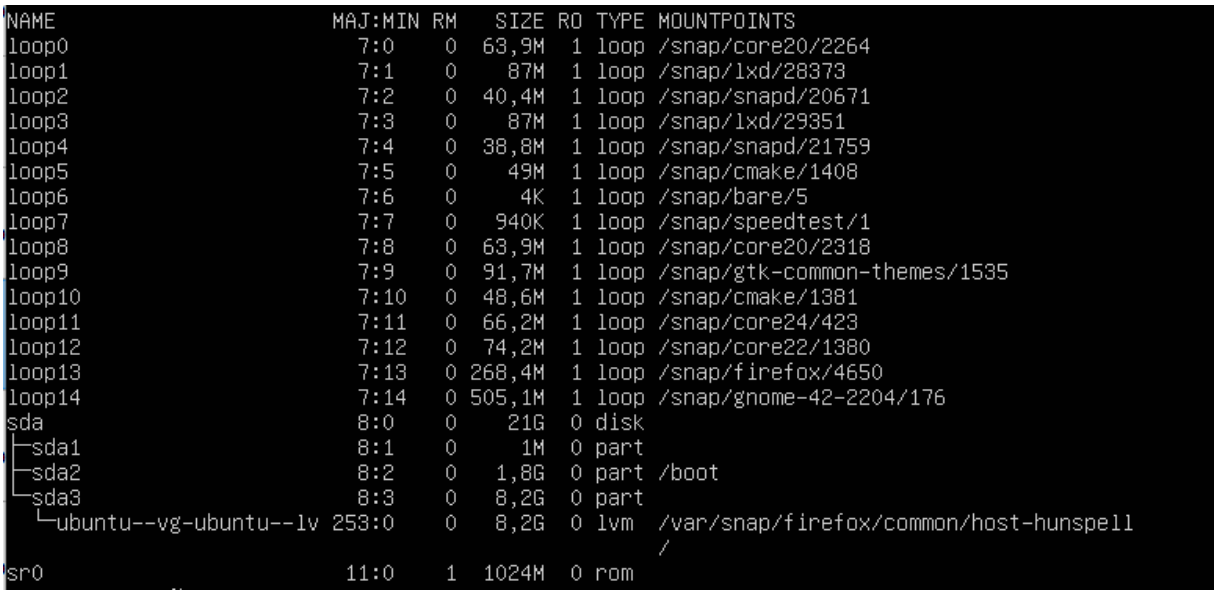

Ilustración 25: Particiones de disco y espacio asignado

La partición en uso es "*Ubuntu—vg-ubuntu—lv*". Se han ejecutado los comandos necesarios para aumentar su tamaño, incrementando la capacidad de la partición "*sda*". Ahora, es necesario asignar el espacio adicional de *sda* a la partición que se está utilizando. Para ello, se ejecuta el siguiente comando:

```
sudo growpart /dev/sda 3
sudo lvextend -L +5G /dev/mapper/ ubuntu--vg-ubuntu--lv
```

De este modo, se puede verificar que la partición deseada ha aumentado de tamaño.

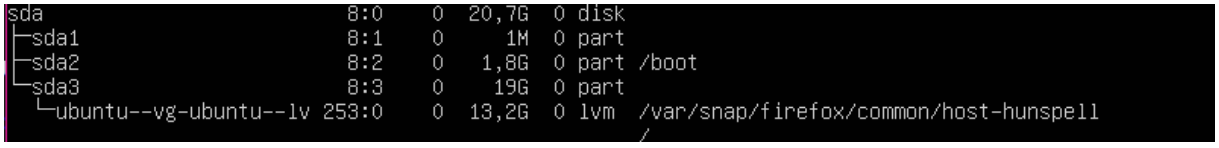

Ilustración 26: Aumento de partición de disco

4.7 Instalación y configuración de OpenVPN

Para permitir la comunicación entre los distintos equipos, se instalará el software OpenVPN en las máquinas locales que actúan como gNB1 y gNB2, las cuales funcionarán como clientes. Asimismo, se instalará en la máquina Free5gc, que actuará como servidor.

Para realizar la instalación y configuración, se seguirá el tutorial de OpenVPN [41].

4.7.1 Servidor OpenVPN

El servidor se instalará en la máquina virtual desplegada en Azure. Antes de proceder con la instalación, es importante actualizar la lista de paquetes y sus versiones disponibles en los repositorios. Esto asegura que se trabajará con las versiones más recientes y seguras del software, evitando posibles problemas de compatibilidad o dependencias. Para ello, se ejecuta el siguiente comando:

```
sudo apt update
```

A continuación, se procede a la instalación de Easy-RSA y OpenVPN. Easy-RSA [42] es una herramienta ampliamente utilizada para gestionar la infraestructura de Clave Pública basada en certificados X.509. Esta herramienta es fundamental para la creación y gestión de certificados digitales y claves RSA, para establecer conexiones seguras en redes, especialmente en servicios como VPNs.

Por otro lado, OpenVPN es tanto un protocolo VPN como un software que se encarga de manejar las comunicaciones cliente-servidor, permitiendo la creación de túneles seguros entre distintos equipos a través de la red.

Para instalar estas dos herramientas y así habilitar la gestión de certificados y la configuración de la VPN, se ejecutan los siguientes comandos:

```
sudo apt install easy-rsa
sudo apt install openvpn
```

Si la instalación se ha realizado correctamente, ambas herramientas estarán disponibles en el sistema. A continuación, se procederá a copiar el directorio Easy-RSA desde su ubicación predeterminada en *'/usr/share/easy-rsa'* a la carpeta de configuración de OpenVPN en *'/etc/openvpn'*. Esta acción asegura que, en caso de errores o problemas, el archivo original se pueda recuperar fácilmente desde su ubicación inicial.

Se configurará una infraestructura de clave pública en la máquina Free5gc, que funcionará como servidor. Esta infraestructura es crucial para la gestión de certificados TLS necesarios para que los clientes puedan establecer conexiones seguras con la VPN. La máquina Free5gc

generará y administrará estos certificados, facilitando así la autenticación y la seguridad de la conexión.

Para copiar el directorio a la nueva ubicación, se ejecutará el siguiente comando:

```
sudo cp -r /usr/share/easy-rsa /etc/openvpn
```

Para asegurar que los permisos de lectura, escritura y ejecución están correctamente configurados, se utiliza el siguiente comando:

```
sudo chown -R free5gc /etc/openvpn/easy-rsa
```

El comando chown se utiliza para cambiar el propietario y el grupo asociado de los archivos y directorios en el sistema. El parámetro *'-R'* indica que el cambio de propietario debe aplicarse de manera recursiva a todos los archivos y subdirectorios dentro del directorio especificado, garantizando que todos los elementos del directorio tengan los permisos adecuados para el usuario designado.

Una vez que se han configurado los permisos adecuados, se procede a ejecutar el siguiente comando. Este comando ejecutará el script Easy-RSA, que inicializa el directorio de la infraestructura de clave pública (PKI) y configura el entorno necesario para la gestión de certificados y claves.

```
free5gc@Free5GC:/etc/openvpn/easy-rsa$ ./easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/easy-rsa/pki
```

Ilustración 27: Inicialización de easy-rsa

A continuación, se procederá a crear el certificado de la Autoridad de Certificación (CA). Este certificado es necesario para verificar la identidad tanto del servidor como de los clientes, asegurando que ambas partes sean quienes dicen ser. De esta manera, se establece una conexión segura entre el servidor y los clientes, garantizando que los datos no sean modificados durante la transmisión.

El primer paso es generar datos aleatorios en un archivo *".rnd"*, el cual se creará para inicializar un generador de números aleatorios con OpenSSL.

OpenSSL [43] es una biblioteca de código abierto que ofrece herramientas para implementar criptografía y protocolos de seguridad en redes, facilitando la gestión de certificados digitales, claves criptográficas y la protección de las comunicaciones.

Este archivo es necesario para la generación de certificados y claves, ya que proporciona la entropía necesaria para crear elementos criptográficos seguros.

```
openssl rand -writerand .rnd
```

Si ha ido bien, se habrá creado un archivo llamado *".rnd"*. Para continuar con la creación del certificado de la Autoridad de Certificación, se debe ejecutar el siguiente comando:

```
./easyrsa build-ca
```

Este comando crea una nueva autoridad de certificación (CA) dentro de una infraestructura de clave pública (PKI). Este proceso genera una clave privada para la CA, la cual se utiliza para firmar los certificados de los servidores y clientes. También se crea un certificado para la CA, el cual sirve para validar la autenticidad de los certificados emitidos por esta CA. El certificado de la CA actúa como un vínculo de confianza que permite a los clientes verificar que los certificados de los servidores son legítimos y emitidos por una fuente confiable.

Durante la ejecución del comando, se solicita una "*CA Key passphrase*", que protege la clave privada de la CA, y un "*Common Name*", que identifica la entidad a la que se emite el certificado. En este proyecto, se ha utilizado la **Passphrase:** '*cakey5gc*' y el **Common Name:** '*free5gc*'.

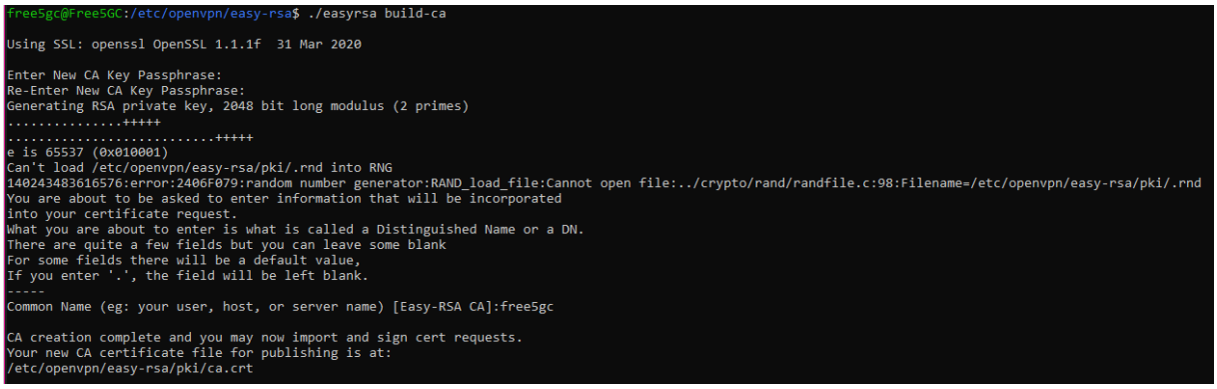

Ilustración 28: Creación del certificado de la CA

A continuación, se genera una solicitud de certificado (CSR) con la información necesaria para que la CA pueda emitir un certificado para el servidor. Además, se crea una clave privada que se utilizará para la firma del certificado. Para realizar este proceso, se ejecuta el siguiente comando:

```
./easyrsa gen-req serverVPN nopass
```

Este comando solicitará al usuario que proporcione un *"Common Name"* para el servidor, en este caso, se ha asignado *'serverVPN'*. Como resultado de la ejecución del comando, se generarán dos archivos: *'serverVPN.req'*, que contiene la solicitud de certificado, y *'serverVPN.key'*, que es la clave privada asociada.

Una vez que se han creado estos archivos, el siguiente paso es firmar la solicitud de certificado con la Autoridad de Certificación (CA). Este proceso de firma convierte la solicitud en un certificado válido, que será utilizado para autenticar el servidor en la infraestructura de clave pública.

El siguiente paso es firmar el certificado con la Autoridad de Certificación.

```
./easyrsa sign-req server serverVPN
```

Con este comando se firma la solicitud de certificado previamente generada para el servidor VPN. Este proceso convierte la solicitud de certificado (CSR) en un certificado válido. Al firmar la solicitud, la Autoridad de Certificación (CA) verifica y autentica la identidad del servidor, garantizando que el servidor es legítimo y permitiendo que los clientes confíen en él.

Si el comando se ejecuta correctamente, se habrá generado un archivo llamado serverVPN.crt. Este archivo contiene el certificado del servidor, que se usa para establecer conexiones seguras.

El siguiente paso es generar los parámetros Diffie-Hellman [44]. Estos parámetros se usan en el protocolo de intercambio de claves Diffie-Hellman, el cual permite a dos partes establecer una clave compartida de manera segura a través de un canal inseguro. El protocolo, se basa en la matemática de los números primos y los logaritmos discretos para permitir el intercambio seguro de claves sin necesidad de compartir la clave directamente.

Estos parámetros aseguran que las comunicaciones entre servidor y clientes sean cifradas, añadiendo una capa adicional de seguridad a la infraestructura de clave pública y permitiendo a dos partes establecer una clave compartida de manera segura a través de un canal inseguro.

Para ello ejecutamos el siguiente comando:

```
./easyrsa gen-dh
```

Este comando genera un archivo de parámetros Diffie-Hellman utilizando EasyRSA. Este archivo, llamado *'dh.pem'*, se guarda en el mismo directorio en el que se ha ejecutado el comando.

Cuando un cliente se conecta al servidor VPN, el contenido del archivo *'dh.pem'* se comparte para permitir la generación de una clave de cifrado común entre el cliente y el servidor. Este proceso asegura que ambos extremos de la conexión puedan cifrar y descifrar los datos enviados a través de la VPN. La clave de cifrado generada a partir de estos parámetros se establece de manera independiente en cada conexión cliente-servidor, lo que significa que incluso si una clave se ve comprometida, los datos de otras sesiones permanecen seguros y no pueden ser descifrados.

Después de la generación de los parámetros Diffie-Hellman, se procede a configurar los parámetros del servidor. Para mayor seguridad, se recomienda hacer una copia de seguridad del archivo de configuración antes de realizar cualquier modificación. Esto permite recuperar la configuración inicial en caso de que ocurra algún problema durante el proceso de configuración. Para realizar la copia de seguridad, se ejecuta el siguiente comando:

```
sudo cp
/usr/share/doc/openvpn/examples/sample-
config-files/server.conf.gz
/etc/openvpn/server/server.conf.gz
sudo gzip -d
/etc/openvpn/server/server.conf.gz
```

Se edita el fichero con el siguiente comando:

```
sudo nano /etc/openvpn/server/server.conf
```

En el Anexo B, se puede consultar de manera más detallada el contenido de este archivo de configuración. Entre los parámetros más importantes del archivo se incluyen:

- **Puerto:** en el archivo especifica el puerto 1194, que es el puerto por el cual el servidor OpenVPN escucha las conexiones entrantes.

- **Protocolo:** se define el protocolo utilizado, que en este caso es TCP.

- **Rutas a Archivos:** se indican las rutas para los archivos esenciales generados anteriormente: *'ca.crt'*, *'serverVPN.crt'*, *'serverVPN.key'*, y *'dh.pem'*.

- **Subred VPN:** Se establece la subred que se utilizará para asignar direcciones IP a los clientes VPN. En este caso, se utiliza la subred 10.8.0.0 255.255.255.0.

Una vez que se ha modificado el archivo de configuración, se procede a iniciar el servicio de OpenVPN y a configurarlo para que se inicie automáticamente cada vez que la máquina se arranque. Para ello, se ejecutan los siguientes comandos:

```
sudo systemctl start openvpn-server@server
sudo systemctl enable openvpn-server@server
sudo systemctl status openvpn-server@server
```

Primero se inicia el servicio OpenVPN para la configuración especificada, después se habilita el servicio OpenVPN para que se inicie en cada arranque de forma automática.

Finalmente, se ejecuta el último comando que permite visualizar el estado del servicio OpenVPN, mostrando si está activo o inactivo. Esto proporciona información sobre el estado actual del servicio y ayuda a confirmar que se ha iniciado correctamente.

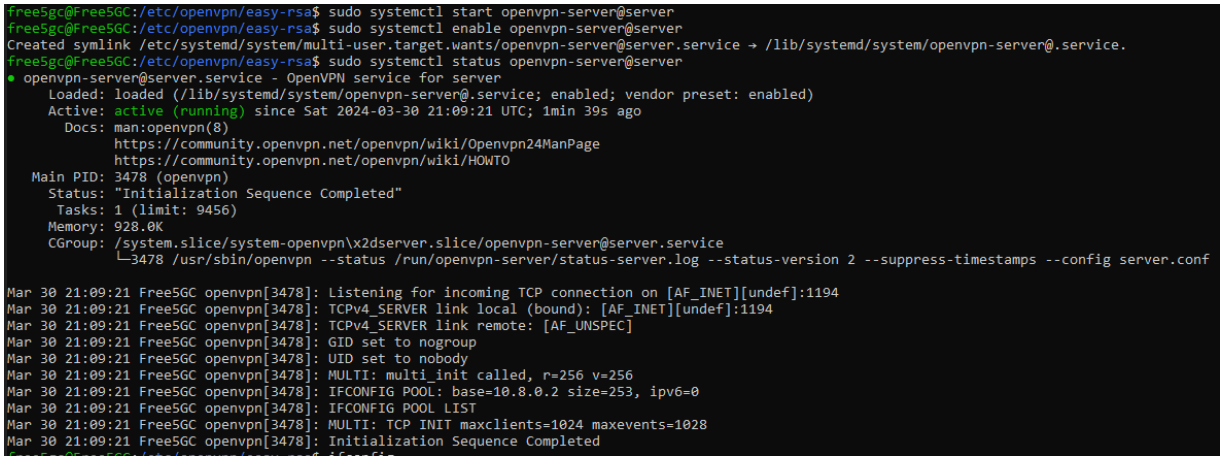

Ilustración 29: Estado servicio OpenVPN en Free5gc

En caso de que todo haya ido bien, al revisar el estado deberíamos ver que el campo *"Active"* tiene el valor *"active (running)"*. Esto indica que el servicio está funcionando correctamente. Además, el campo *"Status"* proporcionará información sobre el estado del servicio, incluyendo cualquier posible problema ocurrido durante la inicialización.

Con el servicio en funcionamiento, podemos verificar que se ha creado una nueva interfaz de red. En este caso, la interfaz creada es tap0.

```
free5gc@Free5GC:/etc/openvpn/easy-rsa$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.4  netmask 255.255.255.0  broadcast 10.0.1.255
        inet6 fe80::20d:3aff:fe78:ee50  prefixlen 64  scopeid 0x20<link>
        ether 00:0d:3a:78:ee:50  txqueuelen 1000  (Ethernet)
        RX packets 27428  bytes 7229123 (7.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29875  bytes 5749519 (5.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 144  bytes 17816 (17.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 144  bytes 17816 (17.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tap0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.8.0.1  netmask 255.255.255.0  broadcast 10.8.0.255
        inet6 fe80::ecaf:69ff:fe5a:9f6d  prefixlen 64  scopeid 0x20<link>
        ether ee:af:69:5a:9f:6d  txqueuelen 100  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 866 (866.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Ilustración 30: Interfaces de Free5gc

4.7.2 Clientes OpenVPN

El siguiente paso es configurar correctamente los clientes que forman parte de la red, en este caso, las máquinas que actúan como GNB1 y GNB2. Para ello, es necesario que el software OpenVPN esté instalado en estas máquinas. Se procede a la instalación del software ejecutando el siguiente comando:

```
sudo apt install openvpn
```

Este comando instalará OpenVPN en las máquinas GNB1 y GNB2, permitiéndoles conectarse a la red VPN y comunicarse de manera segura con el servidor OpenVPN.

El siguiente paso es generar los certificados y claves necesarios para cada uno de los clientes. Para hacerlo, primero se debe acceder a la ruta *'/etc/openvpn/easy-rsa'* en la máquina cliente. A continuación, se ejecuta el siguiente comando para generar los certificados y claves para el cliente, en este caso, para gNB1:

```
./easyrsa gen-req clientVPNgNB1 nopass
```

En el caso del gNB2 sería:

```
./easyrsa gen-req clientVPNgNB2 nopass
```

Este comando genera una solicitud de certificado (CSR) para un cliente y especifica que la clave privada generada no está protegida por una contraseña. Durante el proceso, se solicitará al usuario que proporcione un *"Common Name"*.

En este caso, los *"Common Name"* elegidos son *"clientVPNgNB1"* para el primer cliente y *"clientVPNgNB2"* para el segundo. Estos nombres aseguran que cada cliente tenga un identificador único dentro de la red VPN.

```
./easyrsa sign-req client clientVPNgNB1
```

Para el caso del gNB2 se ejecuta este comando:

```
./easyrsa sign-req client clientVPNgNB2
```

A la salida si ha ido bien debería salir un resultado similar al de la siguiente figura, tanto para el gNB1 como para el gNB2.

```
Free5GC@Free5GC:/etc/openvpn/easy-rsa$ ./easyrsa sign-req client clientVPNgNB2

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 1080 days:

subject=
    commonName                = vpnclientgNB2


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'vpnclientgNB2'
Certificate is to be certified until Jul  7 16:23:17 2027 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /etc/openvpn/easy-rsa/pki/issued/clientVPNgNB2.crt
```

Ilustración 31: Firma de certificado clientVPNgNB2

Se tendrán en total dos certificados y dos claves que hay que enviar a cada uno de los clientes locales además del certificado de la Autoridad de Certificación.

Se generarán un total de dos certificados y dos claves para los clientes, además del certificado de la Autoridad de Certificación (CA). Estos archivos deben ser enviados a las máquinas locales correspondientes.

Para transferir los archivos de una máquina a otra, se puede utilizar MobaXterm, una herramienta que facilita la transferencia de archivos entre máquinas virtuales y sistemas locales. En el Anexo A se detalla el uso de MobaXterm. Esta herramienta permite extraer los archivos de la máquina virtual en Azure y almacenarlos en el ordenador local, desde donde podrán ser transferidos a las máquinas virtuales locales.

Una vez que los archivos estén en el ordenador local, deben ser copiados a las máquinas virtuales locales. Para realizar esta transferencia desde la línea de comandos del propio ordenador, se utiliza el siguiente comando, especificando la ruta donde se encuentran los archivos descargados:

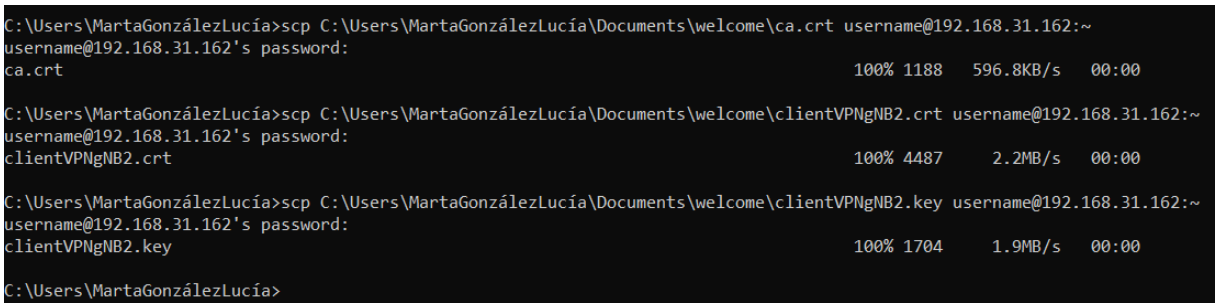

Ilustración 32: Transferencia de ficheros

Estos pasos deben repetirse para cada uno de los clientes. Una vez transferidos los archivos, se procede a modificar los archivos de configuración de cada cliente.

Antes de realizar cualquier cambio en los parámetros de configuración, es importante hacer una copia de seguridad del archivo original. Esto garantiza que, en caso de cometer algún error durante la modificación, se pueda recuperar la configuración inicial sin dificultad.

Para realizar la copia de seguridad del archivo de configuración, se ejecuta el siguiente comando:

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-
files/client.conf
/etc/openvpn/client/clientgNB1.conf
```

A continuación, se edita el fichero de configuración con el siguiente comando:

```
sudo nano /etc/openvpn/client/clientgNB1.conf
```

El contenido del archivo de configuración se encuentra en el Anexo B. A continuación, se detallan los parámetros más importantes de este archivo:

- **Puerto y Dirección IP:** se especifica el puerto 1194, por el cual el servidor OpenVPN escuchará las conexiones entrantes, y la dirección IP pública del servidor.
- **Protocolo:** se define el protocolo TCP, igual que en la configuración del servidor.
- **Rutas a Archivos:** se indican las rutas para los archivos esenciales generados anteriormente, como *"ca.crt"*, *"clientVPNgNB1.crt"*, y *"clientVPNgNB1.key"*.
- **Servidor:** Se especifica el servidor al que se conectará el cliente, que cuenta con un certificado de autenticación de servidor web TLS (TLS Web Server Authentication). Esto asegura que el servidor es legítimo y permite una conexión segura.

Es importante ajustar algunos valores en los archivos de configuración según si se está configurando gNB1 o gNB2. Una vez modificado el archivo, se puede verificar su funcionamiento ejecutando el siguiente comando:

```
sudo openvpn --config /etc/openvpn/client/clientgNB1.conf
```

Esto inicia una conexión VPN y si todo ha ido bien podemos visualizar una nueva interfaz, en el caso del gNB1 la interfaz es la tun0.

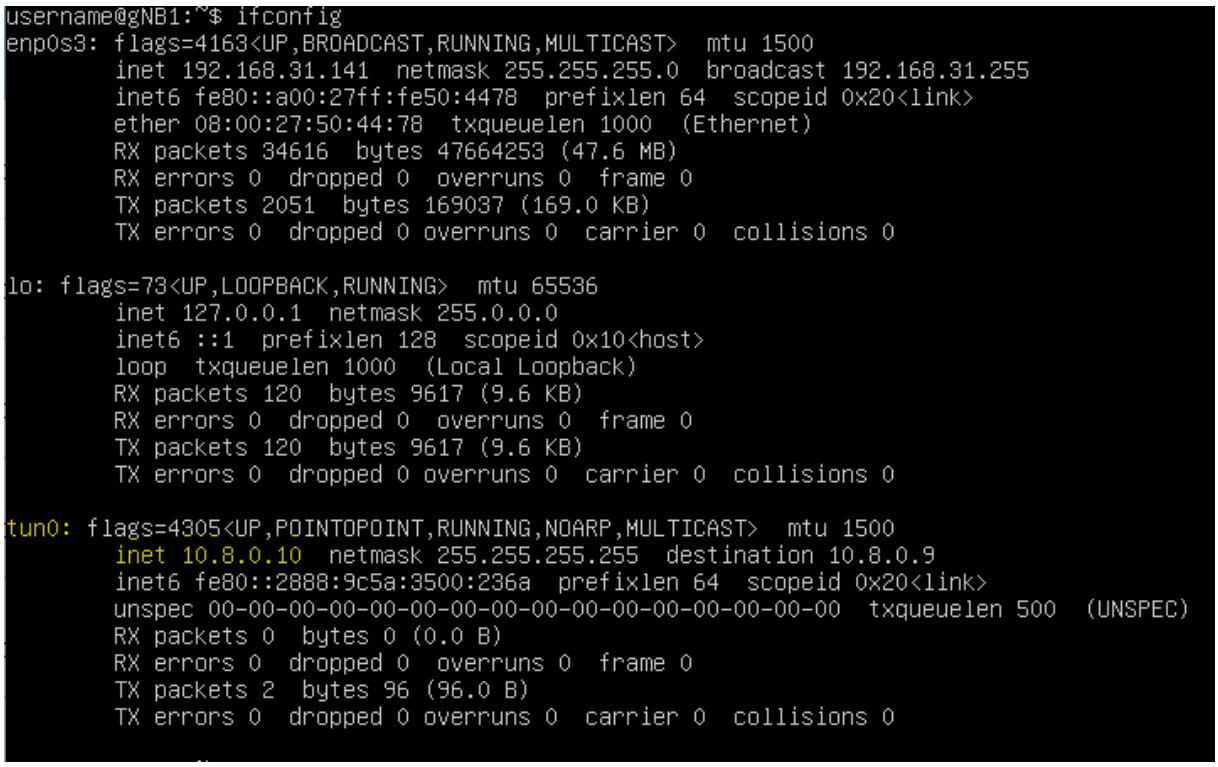

Ilustración 33: Interfaces de gNB1

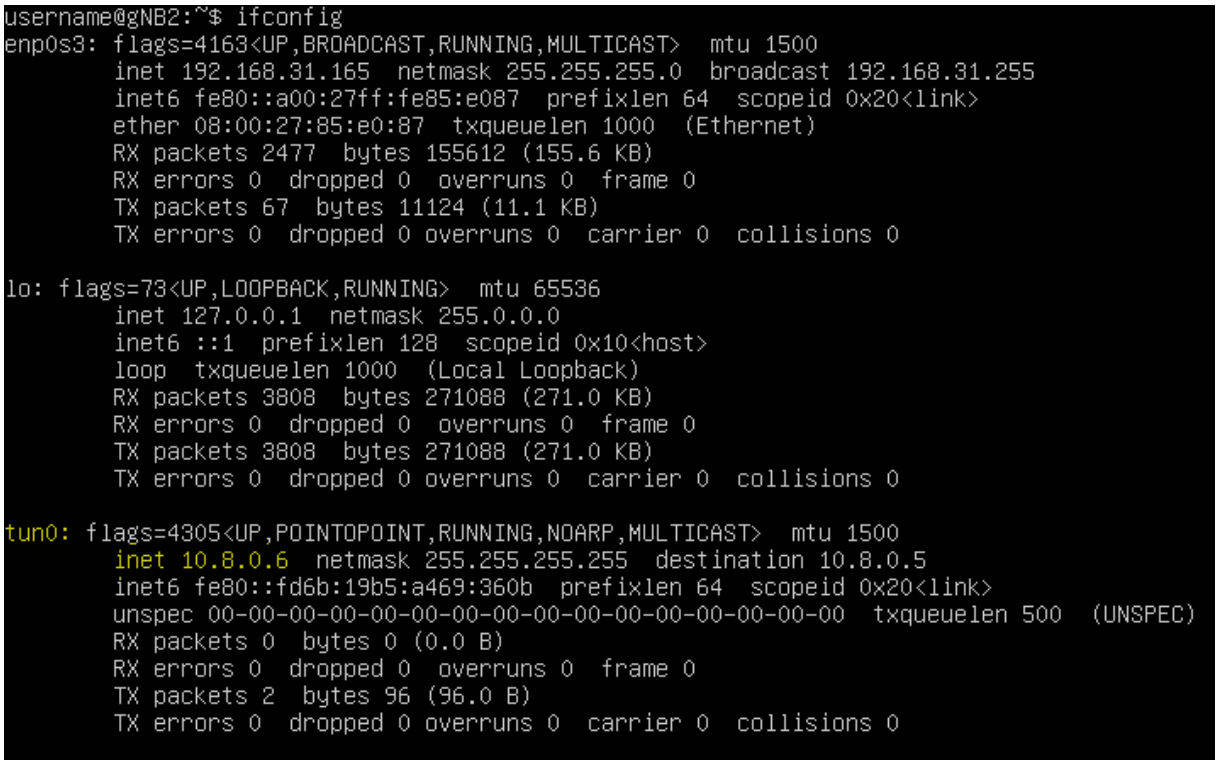

Ilustración 34: Interfaces de gNB2

Para verificar que el servicio OpenVPN está funcionando correctamente, se pueden revisar los registros generados en la máquina de Azure. La información relevante se encuentra al final del archivo *'openvpn-status.log'*, por lo que es útil mostrar solo las últimas líneas del archivo.

Este archivo guarda el estado actual de las conexiones VPN, incluyendo detalles como las direcciones IP de los clientes conectados y el tiempo de conexión.

Para visualizar las últimas líneas de este archivo en el servidor, se debe ejecutar el siguiente comando:

```
sudo tail -n 20 /etc/openvpn/server/openvpn-status.log
```

Este comando muestra las últimas líneas del archivo y se actualiza en tiempo real para reflejar los cambios a medida que se generan nuevos registros. De esta manera se puede monitorizar la actividad de las conexiones VPN y verificar su correcto funcionamiento.

```
Free5GC@Free5GC:~$ sudo tail -n 20 /etc/openvpn/server/openvpn-status.log
TITLE,OpenVPN 2.4.12 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Jun 2
7 2024
TIME,Tue Sep  3 16:27:55 2024,1725380875
HEADER,CLIENT_LIST,Common Name,Real Address,Virtual Address,Virtual IPv6 Address,Bytes Received,Bytes Sent,Connected Sin
ce,Connected Since (time_t),Username,Client ID,Peer ID
CLIENT_LIST,vpnclientgNB1,212.9.72.28:33266,10.8.0.10,,4033,3793,Tue Sep  3 16:25:13 2024,1725380713,UNDEF,1,0
HEADER,ROUTING_TABLE,Virtual Address,Common Name,Real Address,Last Ref,Last Ref (time_t)
ROUTING_TABLE,10.8.0.10,vpnclientgNB1,212.9.72.28:33266,Tue Sep  3 16:25:13 2024,1725380713
GLOBAL_STATS,Max bcast/mcast queue length,0
END
```

Ilustración 35: Estado conexión VPN de free5gc con gNB1

En este caso se puede visualizar como es el gNB1 con el que tiene conexión.

4.8 Instalación y configuración de Free5GC

A continuación se procede a la instalación en la máquina virtual de Azure Free5GC, para ello se seguirá la guía oficial de instalación [36] .

4.8.1 Requisitos previos a la instalación de Free5GC

Antes de proceder con la instalación de Free5GC, es importante cumplir con ciertos requisitos previos para asegurar su correcto funcionamiento. Los requisitos incluyen:

- **Instalación de Golang**: es un lenguaje de programación de código abierto en el que está desarrollado el núcleo de 5G de Free5GC. Para su correcto funcionamiento se requiere que la versión instalada sea “Go 1.21.8”.
  Para poder instalarlo se ejecutan los siguientes comandos:

```
wget https://dl.google.com/go/go1.14.4.linux-amd64.tar.gz
sudo tar -C /usr/local -zxvf go1.14.4.linux-amd64.tar.gz
mkdir -p ~/go/{bin,pkg,src}
# The following assume that your shell is bash
echo 'export GOPATH=$HOME/go' >> ~/.bashrc
echo 'export GOROOT=/usr/local/go' >> ~/.bashrc
echo 'export PATH=$PATH:$GOPATH/bin:$GOROOT/bin' >> ~/.bashrc
echo 'export GO111MODULE=auto' >> ~/.bashrc
source ~/.bashrc
```

- **Instalación de paquetes para el soporte del plano de usuario:** estos paquetes se usan ya que proporcionan las herramientas para compilar, configurar y operar los componentes del plano de usuario de la red 5G.
  En este plano se maneja el tráfico de datos de los usuarios de la red.

```
apt -y install git gcc g++ cmake autoconf libtool pkg-config
libmnl-dev libyaml-dev
```

- **Instalación de paquetes para el soporte del plano de control (MongoDB):** consiste en la instalación de paquetes de software que proporcionan soporte y funcionalidades adicionales necesarias para el correcto funcionamiento de la capa de control de Free5GC. Esta capa es responsable de gestionar la señalización, autenticación, movilidad y otros aspectos del funcionamiento de la red.
  Para operar correctamente, Free5GC requiere una base de datos NoSQL, específicamente MongoDB. Esta base de datos se utiliza para almacenar y gestionar la información esencial para el funcionamiento de Free5GC.

```
sudo apt -y update
sudo apt -y install mongodb wget git
sudo systemctl start mongodb
```

- **Configuraciones de red de Linux:** se requiere el ajuste de ciertos parámetros de red y firewall en el sistema operativo, para asegurar que el tráfico de red se maneje de manera adecuada.

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -p tcp -m tcp --tcp-flags SYN,RST SYN -j
TCPMSS -
-set-mss 1400
sudo systemctl stop ufw
```

- **Versión de kernel de Linux:** para el correcto funcionamiento de la red UPF, se requiere una versión específica del kernel de *Linux: 5.0.0-23-generic o 5.4.x*. Si la versión instalada no coincide con estos requisitos, es necesario realizar un cambio.
  Primero, se debe verificar la versión actual del kernel. Para ello, se ejecuta el siguiente comando:

```
sudo dkpg --list |grep linux-image
```

Este comando permite visualizar un listado de las versiones del kernel instaladas en el sistema. Si la versión requerida no se encuentra en este listado, se puede descargar e instalar la versión necesaria utilizando el siguiente comando:

```
sudo apt install linux-image-5.4.0-1130-azure
```

Una vez que se haya realizado la instalación, al listar las versiones del kernel, debería aparecer la nueva versión instalada.

Ilustración 36: Listado versiones de kernel

Aunque la nueva versión del kernel aparezca en el listado, la máquina virtual puede no estar utilizándola aún. Para asegurarse de que se utiliza la nueva versión, es necesario eliminar las versiones anteriores y conservar únicamente la deseada. Para eliminar las versiones no necesarias, se debe ejecutar el siguiente comando:

```
sudo apt-get purge linux-image-xxx
```

En el comando anterior, *'xxx'* debe ser reemplazado por la versión del kernel que se desea eliminar.

4.8.2 Instalación de Free5GC

Una vez que se han instalado los requisitos previos, se puede proceder con la descarga e instalación de Free5GC. El primer paso consiste en clonar el repositorio que contiene el código fuente de Free5GC. Para ello, se ejecuta el siguiente comando:

```
git clone --recursive -b v3.2.1 -j `nproc`
https://github.com/free5gc/free5gc.git
```

Tras descargar el código, es necesario compilarlo. Este proceso se realiza utilizando el comando *'make'*, que gestiona las dependencias entre los archivos del código fuente. Este comando determina qué partes del programa necesitan ser recompiladas y ejecuta los comandos necesarios para realizar dicha recompilación.

```
cd ~/free5gc
make
```

Si la compilación se ha realizado correctamente, se puede proceder al siguiente paso: la instalación del UPF. Primero, se debe clonar el repositorio de gtp5g desde GitHub. Este repositorio contiene el código fuente para un componente del plano de usuario que implementa el protocolo GTP (GPRS Tunneling Protocol) para el transporte de datos.

A continuación, se compila el código fuente para convertirlo en un formato ejecutable, y finalmente, se procede con la instalación.

```
git clone https://github.com/free5gc/gtp5g.git
cd gtp5g
make
sudo make install
```

4.9 Instalación de Web Console

Web Console es una herramienta de interfaz gráfica diseñada para la gestión y configuración de redes 5G implementadas con Free5GC. Esta herramienta permite realizar diversas operaciones administrativas, como la adición, modificación y eliminación de suscriptores en la red 5G. En este contexto, se emplea para añadir nuevos abonados a la red.

Antes de proceder con la instalación, es necesario preparar el entorno eliminando versiones previas de *"cmdtest"* y *"yarn"* que puedan existir en el sistema, con el fin de evitar conflictos durante el proceso de instalación.

A continuación, se agrega la clave pública del repositorio de Yarn a la lista de claves confiables del sistema. Este paso garantiza la autenticidad y seguridad de los paquetes durante su instalación.

Seguidamente, se descarga e instala el script de configuración para el repositorio de *Node.js* en su versión 12.x. Node.js es una plataforma para el desarrollo de aplicaciones de red escalables y eficientes, ya que permite ejecutar código JavaScript del lado del servidor, optimizando la gestión de múltiples conexiones simultáneas.

```
sudo apt remove cmdtest
sudo apt remove yarn
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key
add -
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee
/etc/apt/sources.list.d/yarn.list
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
sudo apt-get update
sudo apt-get install -y nodejs yarn
```

Una vez se tiene el software necesario descargado se compila Web Console a través de *'make'*.

```
cd ~/free5gc
make webconsole
```

Una vez se ha compilado se puede iniciar el servidor

```
cd ~/free5gc/webconsole
go run server.go
```

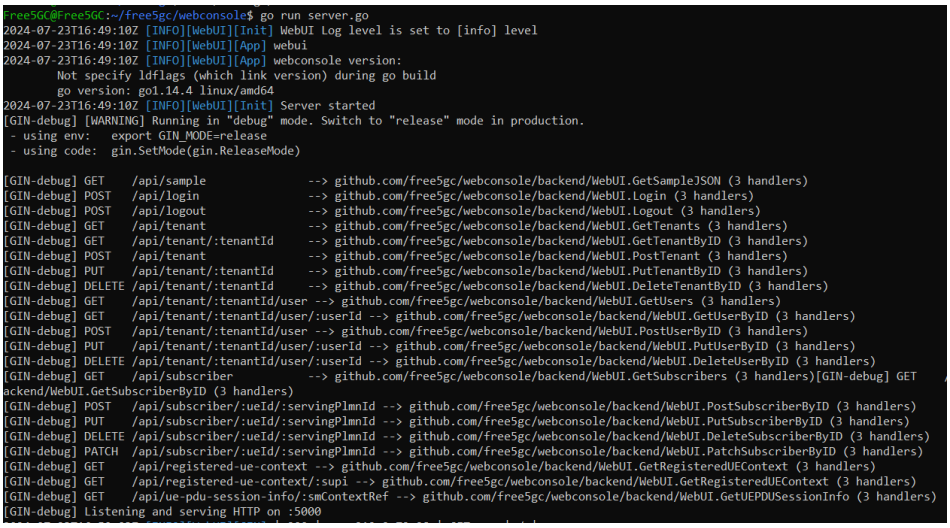

Ilustración 37: Inicio servidor Web Console

El siguiente paso consiste en conectarse a la consola web (Web Console) e iniciar sesión a través de un navegador. Para ello, se debe introducir en la barra de direcciones una URL que incluya la dirección IP pública de la máquina virtual de Azure, seguida del puerto utilizado para la comunicación, en este caso *'http://98.66.160.104:5000'*.

Una vez que la página haya cargado, se debe proceder a la autenticación. El nombre de usuario predeterminado es *"admin"* y la contraseña es *"free5gc"*.

4.10 Registro de abonados

Para registrar abonados en la red se debe seleccionar la pestaña *"SUBSCRIBERS"* y después seleccionamos *"NEW SUBSCRIBER"*. A continuación, se abrirá una ventana donde se muestran los datos de configuración del abonado.

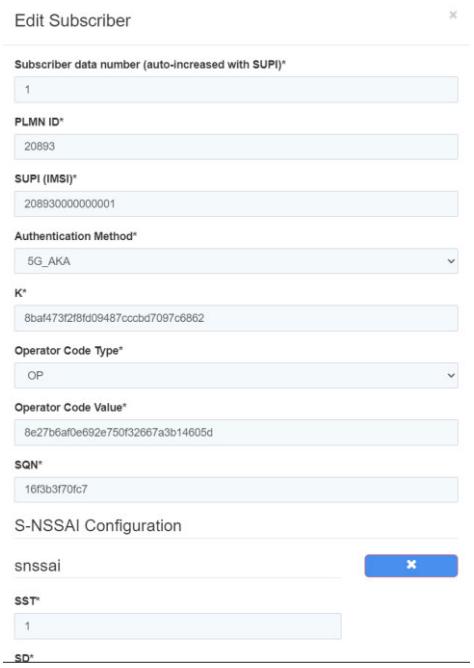

Ilustración 38: Parámetros de configuración UE1

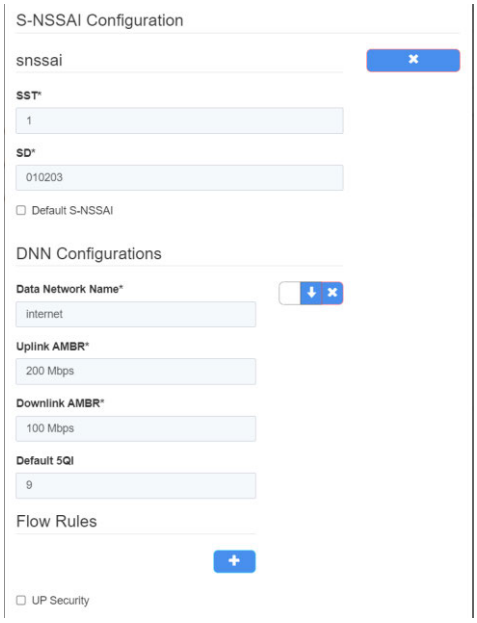

Ilustración 39: Parámetros de configuración UE1

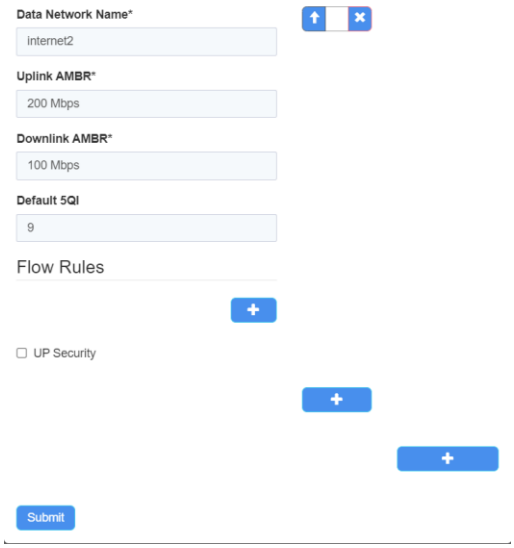

Ilustración 40: Parámetros de configuración UE1

Los valores de los parámetros que se muestran se dejarán por defecto excepto dos que se modifican:

- IMSI (International Mobile Subscriber Identity o Identidad Internacional de Abonado Móvil), este valor tiene que ser único para cada abonado, ya que este se utiliza en el proceso de autenticación y autorización del usuario en la red.

- Operator Code Type: se cambiará al *'OP'* este parámetro es un identificador para diferenciar entre distintos operadores de redes, además se utiliza en procesos de autenticación y autorización para el acceso a ciertos servicios o recursos dentro de la red.

A continuación se muestran los valores de IMSI para cada uno de los abonados:

Tabla 2: Valores de IMSI y terminales asociados

Abonado	IMSI
UE1	208930000000001
UE2	208930000000002
UE3	208930000000003

4.11 Configuración de ficheros Free5g

A continuación, se procederá a la configuración de los archivos necesarios para la instalación de Free5GC, con el fin de establecer la arquitectura requerida para la solución. Para ello, es necesario editar los archivos de configuración correspondientes a cada una de las funciones del núcleo de red.

En esta sección, se modificarán los archivos relacionados con los componentes AMF, SMF y UPF, ya que estos componentes participan en la gestión de las conexiones con las estaciones base y los terminales.

- **AMF**

El primer paso consiste en modificar el archivo de configuración del AMF, ajustando la dirección IP para asegurar que pueda comunicarse correctamente con la estación base. Para llevar a cabo esta modificación, es necesario ubicarse en la carpeta correspondiente a las configuraciones, que se encuentra en el directorio *'config'*.

Para acceder a esta carpeta, se debe ejecutar el siguiente comando:

```
cd free5gc/config
```

Para modificar el fichero correspondiente con la configuración de AMF se ejecuta:

```
sudo nano amfcfg.yaml
```

Una vez en el directorio *'config'*, se debe editar el archivo de configuración del AMF para cambiar el parámetro *'ngapIpList'*, que especifica la interfaz N2. Este parámetro define las direcciones IP que el AMF utilizará para interactuar con las estaciones base.

```
info:
  version: 1.0.3
  description: AMF initial local configuration

configuration:
  amfName: AMF # the name of this AMF
  ngapIpList:  # the IP list of N2 interfaces on this AMF
    - 10.8.0.1 #127.0.0.18
  sbi: # Service-based interface information
    scheme: http # the protocol for sbi (http or https)
    registerIPv4: 127.0.0.18 # IP used to register to NRF
    bindingIPv4: 127.0.0.18  # IP used to bind the service
    port: 8000 # port used to bind the service
    tls: # the local path of TLS key
      pem: config/TLS/amf.pem # AMF TLS Certificate
      key: config/TLS/amf.key # AMF TLS Private key
  serviceNameList: # the SBI services provided by this AMF, refer to TS 29.518
    - namf-comm # Namf_Communication service
```

Ilustración 41: Configuración del archivo AMF

- **SMF**

Para modificar este archivo, similar al proceso del AMF, es necesario ubicarse de nuevo en el directorio *'config'*. Una vez en esta carpeta, se debe ejecutar el siguiente comando para editar el contenido del archivo:

```
sudo nano smfcfg.yaml
```

En este archivo, se debe modificar el parámetro *"endpoints"*. Este parámetro se refiere al enlace de la interfaz N3 o N9, se utiliza para la comunicación entre el componente SMF y los otros elementos de la red, como el UPF y los dispositivos de usuario.

```
            - cidr: 10.60.0.0/16
        - sNssai: # S-NSSAI (Single Network Slice Selection Assistance Information)
            sst: 1 # Slice/Service Type (uinteger, range: 0~255)
            sd: 112233 # Slice Differentiator (3 bytes hex string, range: 000000~FFFFFF)
          dnnUpfInfoList: # DNN information list for this S-NSSAI
            - dnn: internet
              pools:
                - cidr: 10.61.0.0/16
      interfaces: # Interface list for this UPF
        - interfaceType: N3 # the type of the interface (N3 or N9)
          endpoints: # the IP address of this N3/N9 interface on this UPF
            - 10.8.0.1 #127.0.0.8
          networkInstance: internet # Data Network Name (DNN)
  links: # the topology graph of userplane, A and B represent the two nodes of each link
    - A: gNB1
      B: UPF
nrfUri: http://127.0.0.10:8000 # a valid URI of NRF
```

Ilustración 42: Configuración del archivo SMF

- **UPF**

Para modificar este archivo, primero debemos ubicarnos en la carpeta *'config'* y luego ejecutar el siguiente comando:

```
sudo nano upfcfg.yaml
```

En este archivo, se debe modificar el parámetro *'gtpu'*. Este parámetro define la dirección IP que se utilizará para establecer túneles GTP-U con los gNB, a través de los cuales se encapsulan

los datos de usuario. Ajustar correctamente este parámetro es crucial para asegurar la correcta transmisión de los datos entre el UPF y las estaciones base en la red 5G.

```
version: 1.0.3
description: UPF initial local configuration

# The listen IP and nodeID of the N4 interface on this UPF (Can't set to 0.0.0.0)
pfcp:
  addr: 127.0.0.8   # IP addr for listening
  nodeID: 127.0.0.8 # External IP or FQDN can be reached
  retransTimeout: 1s # retransmission timeout
  maxRetrans: 3 # the max number of retransmission

gtpu:
  forwarder: gtp5g
  # The IP list of the N3/N9 interfaces on this UPF
  # If there are multiple connection, set addr to 0.0.0.0 or list all the addresses
  ifList:
    - addr: 10.8.0.1 #127.0.0.8
      type: N3
      # name: upf.5gc.nctu.me
      # ifname: gtpif
```

Ilustración 43: Configuración del archivo UPF

Se genera un script dentro del directorio *"free5gc"* llamado *'configNet.sh'* que simplifica la modificación de las configuraciones tras el reinicio o apagado de la máquina virtual. El contenido del fichero es el siguiente:

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo systemctl stop ufw
sudo iptables -I FORWARD 1 -j ACCEPT
```

Para poder tener permisos sobre el fichero es necesario cambiarlos, para ello:

```
sudo chmod u+x configNet.sh
```

4.12 Instalación y configuración de UERANSIM

En este apartado se procederá a la instalación y configuración de UERANSIM en las máquinas desplegadas en local. Para llevar a cabo esta tarea, se seguirá la guía proporcionada por UERANSIM [45]:

4.12.1 Requisitos previos

El primer paso es actualizar la lista de paquetes disponibles y sus versiones e instalar las versiones más recientes de los paquetes ya instalados.

```
sudo apt update
sudo apt upgrade
```

Para instalar UERANSIM y utilizarlo, el segundo paso es instalar las dependencias necesarias. Para ello, se deben ejecutar los siguientes comandos:

```
sudo apt install make
sudo apt install gcc
sudo apt install g++
sudo apt install libsctp-dev lksctp-tools
sudo apt install iproute2
sudo snap install cmake --classic
```

Se instalan distintas herramientas de compilación, bibliotecas para SCTP (Stream Control Transmission Protocol) y herramientas de red, para poder construir y ejecutar UERANSIM.

4.12.2 Instalación UERANSIM

Una vez que se han instalado los requisitos previos, el siguiente paso es clonar el repositorio de UERANSIM para poder proceder con la instalación. Tras clonar el repositorio, el proyecto debe ser compilado. Una vez completada la compilación, los binarios resultantes se almacenarán en la carpeta *'~/UERANSIM/build'*.

```
git clone https://github.com/aligungr/UERANSIM
cd ~/UERANSIM
make
```

En esta carpeta, se encontrarán los siguientes archivos:

- nr-gnb: ejecutable principal para el gNB (Nodo de la Red de Acceso Radio 5G).
- nr-ue: ejecutable principal para el UE (Equipo de Usuario 5G).
- nr-cli: herramienta de línea de comandos (CLI) para gestionar tanto el gNB como el UE.
- nr-binder: herramienta para habilitar la conectividad a Internet del UE.
- libdevbnd.so: biblioteca dinámica para nr-binder.

4.12.3 Configuración UERANSIM

Para permitir la conexión entre los terminales y los gNB, es necesario configurar los archivos correspondientes en cada máquina virtual. En particular:

- El archivo de configuración para el gNB se encuentra en *'~/UERANSIM/config/free5gc-nb.yaml'*.
- El archivo de configuración para el UE se encuentra en *'~/UERANSIM/config/free5g-ue.yaml'*.

El nombre de estos archivos puede variar según el terminal o nodo que se esté configurando.

Primero, se procederá a configurar los archivos del gNB. Para ello, el primer paso es cambiar el nombre del archivo que se va a modificar para poder identificar cada uno de los gNB. Se asignarán los nombres *'free5gc-gnb1.yaml'* y *'free5gc-gnb2.yaml'* a los archivos correspondientes.

```
cd ~/UERANSIM/config
cp free5gc-gnb.yaml free5gc-gnb1.yaml
```

A continuación, procedemos a editar el primer fichero:

```
sudo nano free5gc-gnb1.yaml
```

```
mcc: '208'          # Mobile Country Code value
mnc: '93'           # Mobile Network Code value (2 or 3 digits)

nci: '0x000000010'  # NR Cell Identity (36-bit)
idLength: 32        # NR gNB ID length in bits [22...32]
tac: 1              # Tracking Area Code

linkIp: 192.168.31.141   # gNB's local IP address for Radio Link Simulation (Usually same with local
 IP)
ngapIp: 10.8.0.10   # gNB's local IP address for N2 Interface (Usually same with local IP)
gtpIp: 10.8.0.10    # gNB's local IP address for N3 Interface (Usually same with local IP)

# List of AMF address information
amfConfigs:
  - address: 10.8.0.1
    port: 38412

# List of supported S-NSSAIs by this gNB
slices:
  - sst: 0x1
    sd: 0x010203

# Indicates whether or not SCTP stream number errors should be ignored.
ignoreStreamIds: true
```

Ilustración 44: Configuración free5gc-gnb1.yaml

```
mcc: '208'          # Mobile Country Code value
mnc: '93'           # Mobile Network Code value (2 or 3 digits)

nci: '0x000000010'  # NR Cell Identity (36-bit)
idLength: 32        # NR gNB ID length in bits [22...32]
tac: 1              # Tracking Area Code

linkIp: 192.168.31.165   # gNB's local IP address for Radio Link Simulation (Usually same with local
 IP)
ngapIp: 10.8.0.6   # gNB's local IP address for N2 Interface (Usually same with local IP)
gtpIp: 10.8.0.6    # gNB's local IP address for N3 Interface (Usually same with local IP)

# List of AMF address information
amfConfigs:
  - address: 10.8.0.1
    port: 38412

# List of supported S-NSSAIs by this gNB
slices:
  - sst: 0x1
    sd: 0x010203

# Indicates whether or not SCTP stream number errors should be ignored.
ignoreStreamIds: true
```

Ilustración 45: Configuración free5gc-gnb2.yaml

A continuación, se procederá a la modificación de los archivos copiados. Es necesario ajustar los siguientes parámetros en los archivos de configuración de los gNB:

- linkIp: dirección IP local de gNB.
- ngapIp: dirección IP de la interfaz N2 para la conexión con el AMF.
- gtpIp: dirección IP de la interfaz N3 para la conexión con el UPF.

Estos parámetros deben configurarse con las direcciones IP correspondientes del gNB: 10.8.0.6 para gNB1 y 10.8.0.10 para gNB2. Además, se debe actualizar el parámetro *'amfConfigs'* con el valor 10.8.0.1, que especifica la dirección IP del AMF.

Una vez configurados los gNB, se procederá a la configuración de los terminales. Se crearán copias de los archivos de configuración para los terminales, asignándoles los nombres correspondientes *'free5gc-ue1.yaml'*, *'free5gc-ue2.yaml'*, o *'free5gc-ue3.yaml'*, dependiendo del terminal que se esté configurando, en este caso se hará para UE1.

```
cd ~/UERANSIM/config
cp free5gc-ue.yaml free5gc-ue1.yaml
```

Una vez copiado procedemos a cambiar el contenido de los ficheros:

```
sudo nano free5gc-ue1.yaml
```

```
  GNU nano 6.2                                  free5gc-ue1.yaml
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits)
supi: 'imsi-208930000000001'
# Mobile Country Code value of HPLMN
mcc: '208'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '93'
# SUCI Protection Scheme : 0 for Null-scheme, 1 for Profile A and 2 for Profile B
protectionScheme: 0
# Home Network Public Key for protecting with SUCI Profile A
homeNetworkPublicKey: '5a8d38864820197c3394b92613b20b91633cbd897119273bf8e4a6f4eec0a650'
# Home Network Public Key ID for protecting with SUCI Profile A
homeNetworkPublicKeyId: 1
# Routing Indicator
routingIndicator: '0000'

# Permanent subscription key
key: '8baf473f2f8fd09487cccbd7097c6862'
# Operator code (OP or OPC) of the UE
op: '8e27b6af0e692e750f32667a3b14605d'
# This value specifies the OP type and it can be either 'OP' or 'OPC'
opType: 'OP'
# Authentication Management Field (AMF) value
amf: '8000'
# IMEI number of the device. It is used if no SUPI is provided
imei: '356938035643803'
# IMEISV number of the device. It is used if no SUPI and IMEI is provided
imeiSv: '4370816125816151'

# List of gNB IP addresses for Radio Link Simulation
gnbSearchList:
  - 192.168.31.141

# UAC Access Identities Configuration
```

Ilustración 46: Configuración de free5gc-ue1.yaml

```
  GNU nano 6.2                                  free5gc-ue2.yaml
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits)
supi: 'imsi-208930000000002'
# Mobile Country Code value of HPLMN
mcc: '208'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '93'
# SUCI Protection Scheme : 0 for Null-scheme, 1 for Profile A and 2 for Profile B
protectionScheme: 0
# Home Network Public Key for protecting with SUCI Profile A
homeNetworkPublicKey: '5a8d38864820197c3394b92613b20b91633cbd897119273bf8e4a6f4eec0a650'
# Home Network Public Key ID for protecting with SUCI Profile A
homeNetworkPublicKeyId: 1
# Routing Indicator
routingIndicator: '0000'

# Permanent subscription key
key: '8baf473f2f8fd09487cccbd7097c6862'
# Operator code (OP or OPC) of the UE
op: '8e27b6af0e692e750f32667a3b14605d'
# This value specifies the OP type and it can be either 'OP' or 'OPC'
opType: 'OP'
# Authentication Management Field (AMF) value
amf: '8000'
# IMEI number of the device. It is used if no SUPI is provided
imei: '356938035643803'
# IMEISV number of the device. It is used if no SUPI and IMEI is provided
imeiSv: '4370816125816151'

# List of gNB IP addresses for Radio Link Simulation
gnbSearchList:
  - 192.168.31.141

# UAC Access Identities Configuration
```

Ilustración 47: Configuración de free5gc-ue2.yaml

```
  GNU nano 6.2                                free5gc-ue3.yaml
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits)
supi: 'imsi-208930000000003'
# Mobile Country Code value of HPLMN
mcc: '208'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '93'
# SUCI Protection Scheme : 0 for Null-scheme, 1 for Profile A and 2 for Profile B
protectionScheme: 0
# Home Network Public Key for protecting with SUCI Profile A
homeNetworkPublicKey: '5a8d38864820197c3394b92613b20b91633cbd897119273bf8e4a6f4eec0a650'
# Home Network Public Key ID for protecting with SUCI Profile A
homeNetworkPublicKeyId: 1
# Routing Indicator
routingIndicator: '0000'

# Permanent subscription key
key: '8baf473f2f8fd09487cccbd7097c6862'
# Operator code (OP or OPC) of the UE
op: '8e27b6af0e692e750f32667a3b14605d'
# This value specifies the OP type and it can be either 'OP' or 'OPC'
opType: 'OP'
# Authentication Management Field (AMF) value
amf: '8000'
# IMEI number of the device. It is used if no SUPI is provided
imei: '356938035643803'
# IMEISV number of the device. It is used if no SUPI and IMEI is provided
imeiSv: '4370816125816151'

# List of gNB IP addresses for Radio Link Simulation
gnbSearchList:
  - 192.168.31.165

# UAC Access Identities Configuration
```

Ilustración 48: Configuración de free5gc-ue3.yaml

A continuación, es necesario ajustar los siguientes parámetros en los archivos de configuración de los terminales:

- **supi**: debe coincidir con el IMSI del terminal, según los valores especificados en la *Tabla 2*.
- **opType**: debe configurarse de acuerdo con el tipo de código de operador, en este caso, *'OP'*.
- **gnbSearchList**: Debe actualizarse con la dirección IP del gNB al que el UE se conectará.

Finalmente, se debe verificar que el resto de los parámetros del archivo de configuración del terminal estén alineados con los datos del abonado registrados previamente a través de la Web Console, para asegurar que el terminal funcione correctamente dentro de la red 5G.

4.13 Instalación y configuración de Cloudflare WARP

El siguiente paso, una vez que la arquitectura 5G está en funcionamiento, es configurar Cloudflare en uno de los terminales, en este caso, en el UE3. Este proceso se realiza con el objetivo de incrementar la seguridad y protección del terminal, así como optimizar el tráfico entre el terminal y la infraestructura de red.

Para instalar Cloudflare WARP, se deben seguir los pasos indicados en la guía [46].

El primer paso consiste en descargar y almacenar la clave pública GPG (GNU Privacy Guard) de Cloudflare WARP, lo cual permite verificar que los paquetes descargados provienen de Cloudflare y no han sido alterados por terceros. Para ello, se ejecuta el siguiente comando:

```
curl -fsSL https://pkg.cloudflareclient.com/pubkey.gpg | sudo gpg --
yes --dearmor --output /usr/share/keyrings/cloudflare-warp-archive-
keyring.gpg
```

A continuación, se agrega el repositorio de Cloudflare WARP a la lista de fuentes de paquetes del sistema, lo que facilita la instalación de Cloudflare. Esto se realiza mediante el siguiente comando:

```
echo "deb [signed-by=/usr/share/keyrings/cloudflare-warp-archive-
keyring.gpg] https://pkg.cloudflareclient.com/ $(lsb_release -cs)
main" | sudo tee /etc/apt/sources.list.d/cloudflare-client.list
```

Si todo ha ido bien podemos proceder a la instalación de Cloudflare Warp:

```
sudo apt-get update && sudo apt-get install cloudflare-warp
```

El siguiente paso consiste en inscribir este dispositivo en la cuenta de Cloudflare Zero Trust de la organización, en este caso, Kyndryl, para que esta máquina virtual funcione bajo las políticas de seguridad establecidas por la organización. Esto se lleva a cabo con el siguiente comando:

```
warp-cli teams-enroll kyndryliberia
```

A continuación, se procede a registrar un nuevo usuario, lo que crea una cuenta de registro para ese dispositivo en el servicio WARP y genera las credenciales necesarias para la conexión. Para hacer este procedimiento más visual, se ha decidido realizar estos pasos mediante una interfaz gráfica y un navegador web. En el anexo C se detalla el proceso de descarga e instalación de la interfaz, así como del navegador Firefox.

En este caso, es necesario registrar al usuario en la plataforma Cloudflare de la empresa. Por lo tanto, junto con la configuración anterior, se ejecuta el siguiente comando:

```
warp-cli registration new
```

Esto abrirá una página en el navegador para iniciar sesión en la plataforma de Cloudflare con un usuario existente, proporcionado por la empresa. Al completar este paso, el dispositivo UE3 quedará vinculado al usuario, lo que permitirá aplicar las políticas y reglas de seguridad correspondientes.

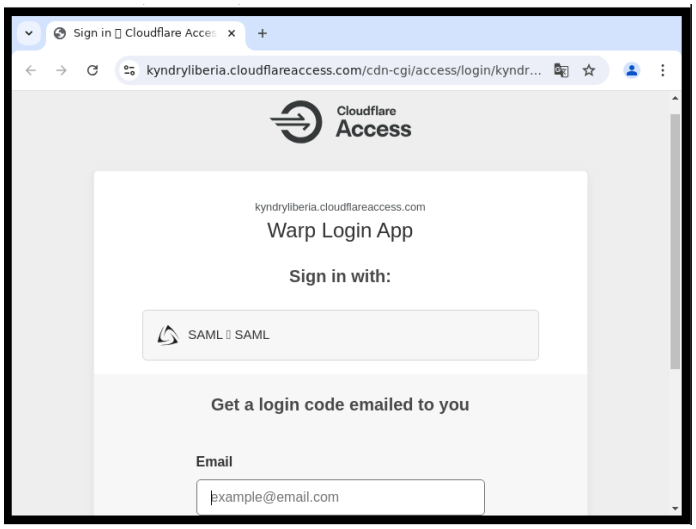

Ilustración 49: Login Cloudflare desde UE3

Para que este inicio de sesión sea efectivo, el registro del nuevo usuario deberá ser aprobado por una segunda persona encargada de la monitorización de la plataforma de Cloudflare.

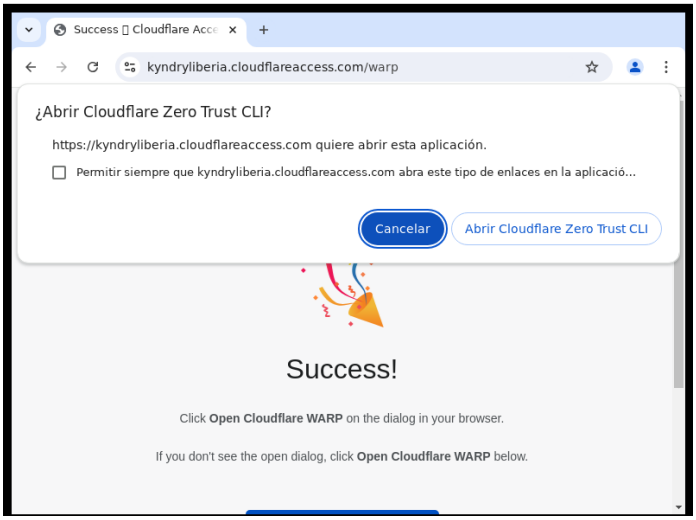

Ilustración 50: Registro correcto con Cloudflare

Si el proceso ha sido exitoso, se mostrará un mensaje emergente indicando que el usuario se ha registrado correctamente, lo que permitirá comenzar a utilizar la herramienta.

A continuación, se debe descargar el certificado de Cloudflare para establecer conexiones seguras y autenticadas entre el UE3 y el servidor de Cloudflare. Sin este certificado, el acceso a través de navegadores o aplicaciones no será posible, ya que no se podrá verificar la autenticidad de la conexión.

Se ejecuta el siguiente comando:

```
sudo wget
https://developers.cloudflare.com/cloudflare-
one/static/Cloudflare_CA.crt
```

Si el proceso se ha completado con éxito, el siguiente paso es añadir este certificado a la lista de certificados del buscador. Para ello, se debe reiniciar Firefox desde la interfaz del UE3 y comprobar que el certificado está instalado y operativo.

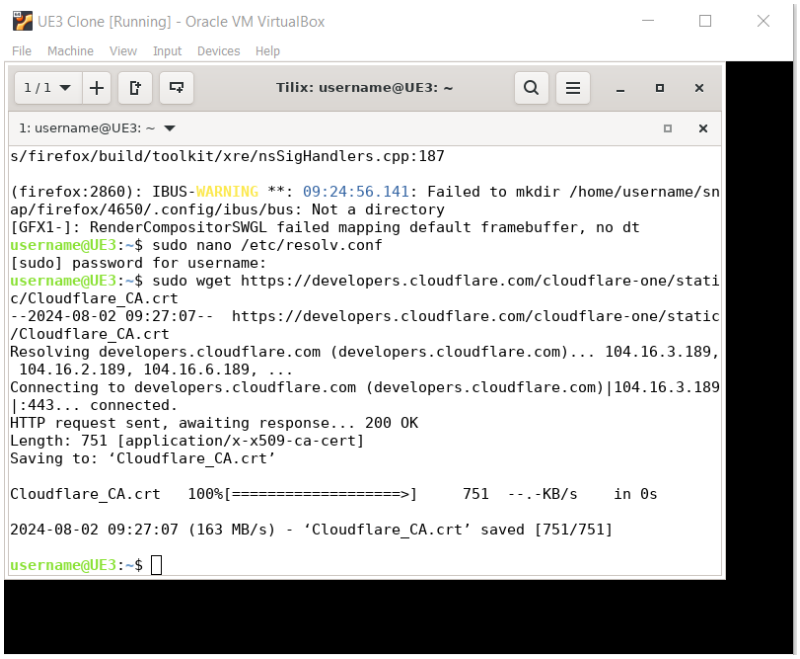

Ilustración 51: Descarga certificado Cloudflare

A partir de ahora, las conexiones entre usuarios y servidores realizadas con el cliente WARP activo estarán cifradas y protegidas contra interceptaciones.

Para añadir el certificado al navegador Firefox, se debe hacerlo manualmente mediante el siguiente comando:

```
sudo certutil -A -n "Cloudflare for Teams ECC Certificate Authority
- Cloudflare" -t "CT,c," -i /home/Cloudflare_CA.crt -d
sql:/home/username/snap/firefox/common/.mozilla/firefox/agozqg7d.de
fault
```

Para verificar que el certificado se ha añadido correctamente, primero hay que asegurarse de que Firefox esté en ejecución. Luego, haz clic en el ícono de menú en la esquina superior derecha (representado por tres líneas horizontales) para desplegar el menú, y selecciona la opción *"Ajustes"*. En la nueva pestaña que se abre, utiliza el campo de búsqueda para buscar *"certificados"*. A continuación, se selecciona la opción *"Ver certificados"* para visualizar la lista de certificados.

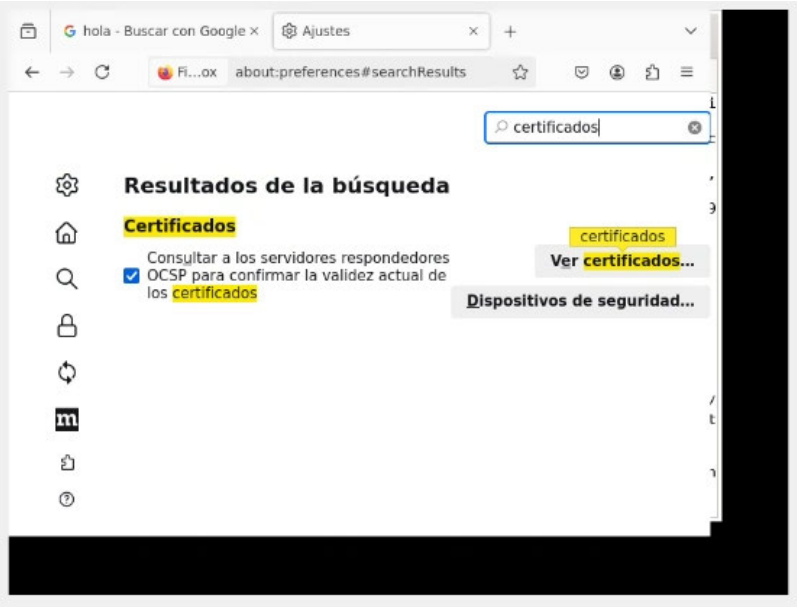

Ilustración 52: Certificados Firefox

A continuación se nos muestra una lista con todos los certificados que se encuentran en Firefox.

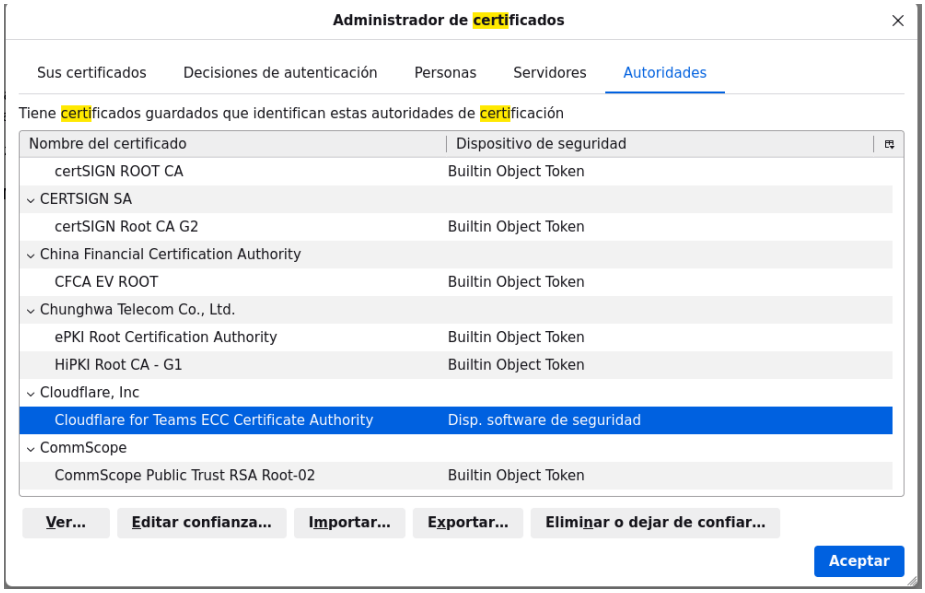

Ilustración 53: Administrador de certificados de Firefox

Si el certificado se ha añadido correctamente, aparecerá el certificado de Cloudflare en el listado de certificados de Firefox, permitiendo su uso en el navegador. Para obtener más detalles, se puede hacer clic en "*Ver*", lo que mostrará la información completa del certificado.

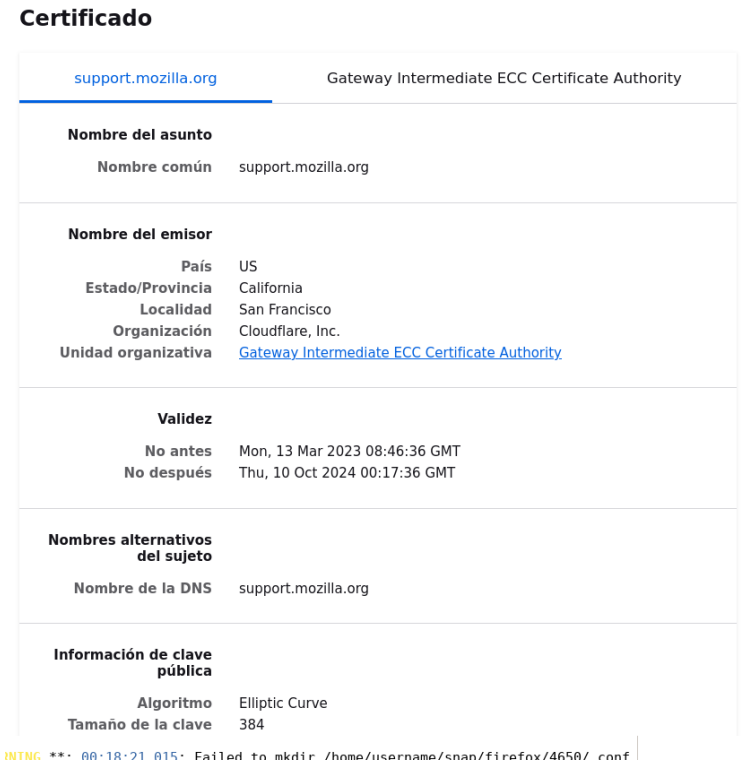

Ilustración 54: Contenido certificado Cloudflare

Una vez registrados procedemos a conectarnos:

```
warp-cli connect
```

Para saber en qué estado se encuentra la conexión se puede ejecutar el siguiente comando:

```
warp-cli status
```

De esta manera sabremos si está conectado o desconectado.

```
username@UE3:~$ warp-cli status
Status update: Connected
Success
username@UE3:~$
```

Ilustración 55: Estado conexión WARP UE3

Una vez estemos conectados, en la propia plataforma de Cloudflare que tiene Kyndryl, se puede ver el nuevo usuario registrado.

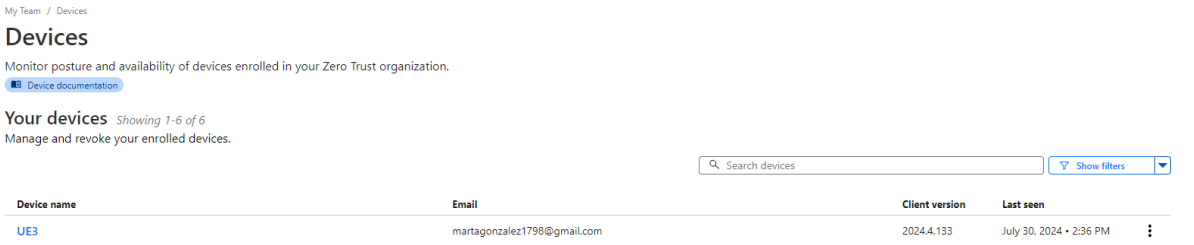

Ilustración 56: Usuarios activos en Cloudflare

4.14 Ejecución

En este apartado se llevará a cabo una prueba del funcionamiento de toda la arquitectura montada, cuyos resultados se analizarán posteriormente. El primer paso consiste en ejecutar en el servidor de Azure el script previamente creado para aplicar las configuraciones de red necesarias para el despliegue de la arquitectura.

```
sudo ./configNet.sh
```

```
Free5GC@Free5GC:~/free5gc$ sudo ./configNet.sh
net.ipv4.ip_forward = 1
```

Ilustración 57: Respuesta de ejecución configNet.sh

El siguiente paso es ejecutar el núcleo de red. Para ello, hay que situarse en la carpeta correspondiente de Free5GC.

```
cd ~/free5gc
./run.sh
```

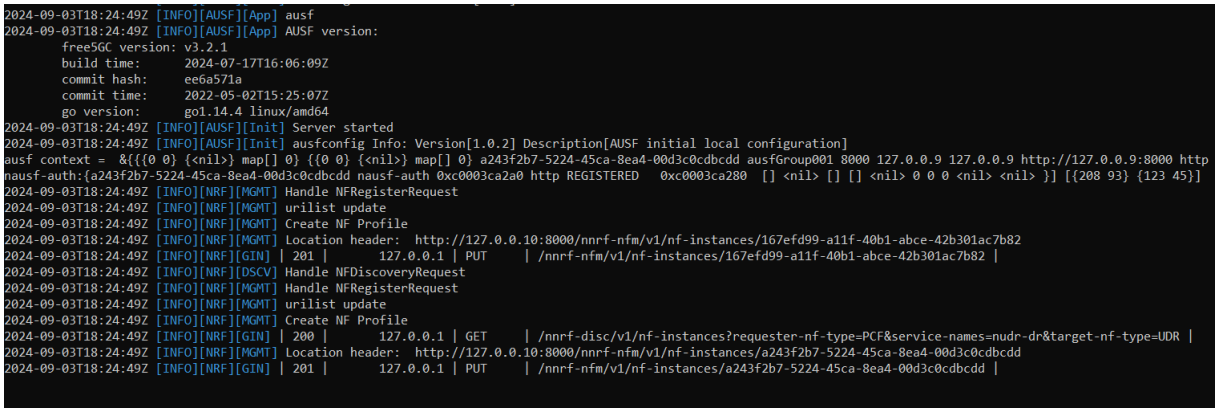

Ilustración 58: Parte de la ejecución del núcleo de red

En el proceso de arranque, los componentes de la red se arrancan en un orden determinado:

- UPF (User Plane Function): primero se inicia el UPF que maneja el plano de usuario y es responsable del enrutamiento.
- NRF: se inicia después del UPF para proporcionar el registro y descubrimiento de funciones en la red.
- AMF (Access and Mobility Management Function): se inicia a continuación, ya que este se encarga de gestionar el acceso y la movilidad de los usuarios, al igual de la autenticación y autorización.
- SMF (Session Management Function): se inicia después, su función principal es gestionar las sesiones de datos y solicitudes de conectividad.
- UDM y UDR: se inician para la gestión y el repositorio de datos del usuario.
- PCF: se inicia para gestionar políticas.
- AUSF se inicia al final para la autenticación.

A continuación ejecutamos en los dos gNB los clientes de OpenVPN y estableciendo el túnel:

```
sudo openvpn --config /etc/openvpn/client/clientgNB1.conf
```

Recalcar, que en este caso se indica el comando para clientgNB1.conf pero varía en función del gNB en el que se quiera ejecutar.

```
cipher AES-256-GCM'
2024-09-03 18:28:41 OPTIONS IMPORT: timers and/or timeouts modified
2024-09-03 18:28:41 OPTIONS IMPORT: --ifconfig/up options modified
2024-09-03 18:28:41 OPTIONS IMPORT: route options modified
2024-09-03 18:28:41 OPTIONS IMPORT: peer-id set
2024-09-03 18:28:41 OPTIONS IMPORT: adjusting link_mtu to 1626
2024-09-03 18:28:41 OPTIONS IMPORT: data channel crypto options modified
2024-09-03 18:28:41 Data Channel: using negotiated cipher 'AES-256-GCM'
2024-09-03 18:28:41 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2024-09-03 18:28:41 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2024-09-03 18:28:41 net_route_v4_best_gw query: dst 0.0.0.0
2024-09-03 18:28:41 net_route_v4_best_gw result: via 192.168.31.1 dev enp0s3
2024-09-03 18:28:41 ROUTE_GATEWAY 192.168.31.1/255.255.255.0 IFACE=enp0s3 HWADDR=08:00:27:50:44:78
2024-09-03 18:28:41 TUN/TAP device tun0 opened
2024-09-03 18:28:41 net_iface_mtu_set: mtu 1500 for tun0
2024-09-03 18:28:41 net_iface_up: set tun0 up
2024-09-03 18:28:41 net_addr_ptp_v4_add: 10.8.0.10 peer 10.8.0.9 dev tun0
2024-09-03 18:28:41 net_route_v4_add: 10.0.1.0/24 via 10.8.0.9 dev [NULL] table 0 metric -1
2024-09-03 18:28:41 net_route_v4_add: 10.8.0.0/24 via 10.8.0.9 dev [NULL] table 0 metric -1
2024-09-03 18:28:41 Initialization Sequence Completed
```

Ilustración 59: Ejecución OpenVPN en gNB1

```
ipher AES-256-GCM'
2024-09-03 18:54:20 OPTIONS IMPORT: timers and/or timeouts modified
2024-09-03 18:54:20 OPTIONS IMPORT: --ifconfig/up options modified
2024-09-03 18:54:20 OPTIONS IMPORT: route options modified
2024-09-03 18:54:20 OPTIONS IMPORT: peer-id set
2024-09-03 18:54:20 OPTIONS IMPORT: adjusting link_mtu to 1626
2024-09-03 18:54:20 OPTIONS IMPORT: data channel crypto options modified
2024-09-03 18:54:20 Data Channel: using negotiated cipher 'AES-256-GCM'
2024-09-03 18:54:20 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2024-09-03 18:54:20 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2024-09-03 18:54:20 net_route_v4_best_gw query: dst 0.0.0.0
2024-09-03 18:54:20 net_route_v4_best_gw result: via 192.168.31.1 dev enp0s3
2024-09-03 18:54:20 ROUTE_GATEWAY 192.168.31.1/255.255.255.0 IFACE=enp0s3 HWADDR=08:00:27:85:e0:87
2024-09-03 18:54:20 TUN/TAP device tun0 opened
2024-09-03 18:54:20 net_iface_mtu_set: mtu 1500 for tun0
2024-09-03 18:54:20 net_iface_up: set tun0 up
2024-09-03 18:54:20 net_addr_ptp_v4_add: 10.8.0.6 peer 10.8.0.5 dev tun0
2024-09-03 18:54:20 net_route_v4_add: 10.0.1.0/24 via 10.8.0.5 dev [NULL] table 0 metric -1
2024-09-03 18:54:20 net_route_v4_add: 10.8.0.0/24 via 10.8.0.5 dev [NULL] table 0 metric -1
2024-09-03 18:54:20 Initialization Sequence Completed
```

Ilustración 60: Ejecución OpenVPN en gNB2

El siguiente paso es ejecutar los gNB que han sido implementados con UERANSIM, se ejecutan los siguientes comandos con el nombre correspondiente al gNB que queramos ejecutar.

```
cd ~/UERANSIM
build/nr-gnb -c config/free5gc-gnb.yaml
```

Esto hará que se establezca la conexión entre núcleo de red y gNB, siendo una conexión SCTP con la función de red AMF.

Se puede observar en el proceso de *"NG Setup"* cómo se realiza un proceso para iniciar la configuración de la interfaz del gNB, formando parte del protocolo NGAP (Next Generation Application Protocol)

```
username@gNB1:~/UERANSIM$ build/nr-gnb -c config/free5gc-gnb1.yaml
UERANSIM v3.2.6
[2024-08-02 21:08:49.564] [sctp] [info] Trying to establish SCTP connection... (10.8.0.1:38412)
[2024-08-02 21:08:49.637] [sctp] [info] SCTP connection established (10.8.0.1:38412)
[2024-08-02 21:08:49.638] [sctp] [debug] SCTP association setup ascId[3]
[2024-08-02 21:08:49.638] [ngap] [debug] Sending NG Setup Request
[2024-08-02 21:08:49.728] [ngap] [debug] NG Setup Response received
[2024-08-02 21:08:49.728] [ngap] [info] NG Setup procedure is successful
```

Ilustración 61: Ejecución UERANSIM desde gNB1

```
username@gNB2:~/UERANSIM$ build/nr-gnb -c config/free5gc-gnb2.yaml
UERANSIM v3.2.6
[2024-09-03 18:57:22.824] [sctp] [info] Trying to establish SCTP connection... (10.8.0.1:38412)
[2024-09-03 18:57:22.889] [sctp] [info] SCTP connection established (10.8.0.1:38412)
[2024-09-03 18:57:22.889] [sctp] [debug] SCTP association setup ascId[3]
[2024-09-03 18:57:22.890] [ngap] [debug] Sending NG Setup Request
[2024-09-03 18:57:22.981] [ngap] [debug] NG Setup Response received
[2024-09-03 18:57:22.981] [ngap] [info] NG Setup procedure is successful
```

Ilustración 62: Ejecución UERANSIM desde gNB2

A continuación, se muestra la información que se muestra en free5GC donde podemos ver cómo se ha hecho el establecimiento de conexión.

```
2024-09-03T18:48:49Z [INFO][AMF][NGAP] [AMF] SCTP Accept from: 10.8.0.10:48124
2024-09-03T18:48:49Z [INFO][AMF][NGAP] Create a new NG connection for: 10.8.0.10:48124
2024-09-03T18:48:49Z [INFO][AMF][NGAP][10.8.0.10:48124] Handle NG Setup request
2024-09-03T18:48:49Z [INFO][AMF][NGAP][10.8.0.10:48124] Send NG-Setup response
```

Ilustración 63: Establecimiento de conexión con gNB1

```
2024-09-04T20:36:06Z [INFO][AMF][NGAP] [AMF] SCTP Accept from: 10.8.0.6:40011
2024-09-04T20:36:07Z [INFO][AMF][NGAP] Create a new NG connection for: 10.8.0.6:40011
2024-09-04T20:36:07Z [INFO][AMF][NGAP][10.8.0.6:40011] Handle NG Setup request
2024-09-04T20:36:07Z [INFO][AMF][NGAP][10.8.0.6:40011] Send NG-Setup response
```

Ilustración 64: Establecimiento de conexión con gNB2

El siguiente paso es establecer conexión con el núcleo 5G procediendo a la ejecución de los UEs. Se ejecutan los siguientes comandos, ajustando el nombre del fichero al terminal en el que queramos ejecutarlo.

```
cd ~/UERANSIM
sudo build/nr-ue -c config/free5gc-ue.yaml
```

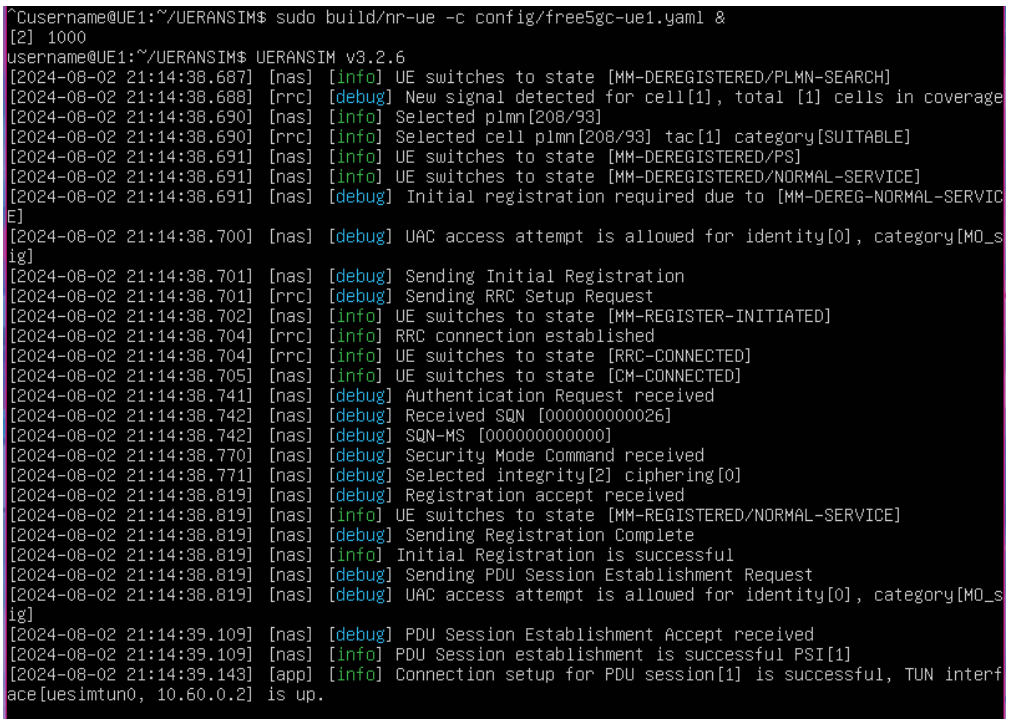

Ilustración 65: Ejecución UERANSIM en UE1

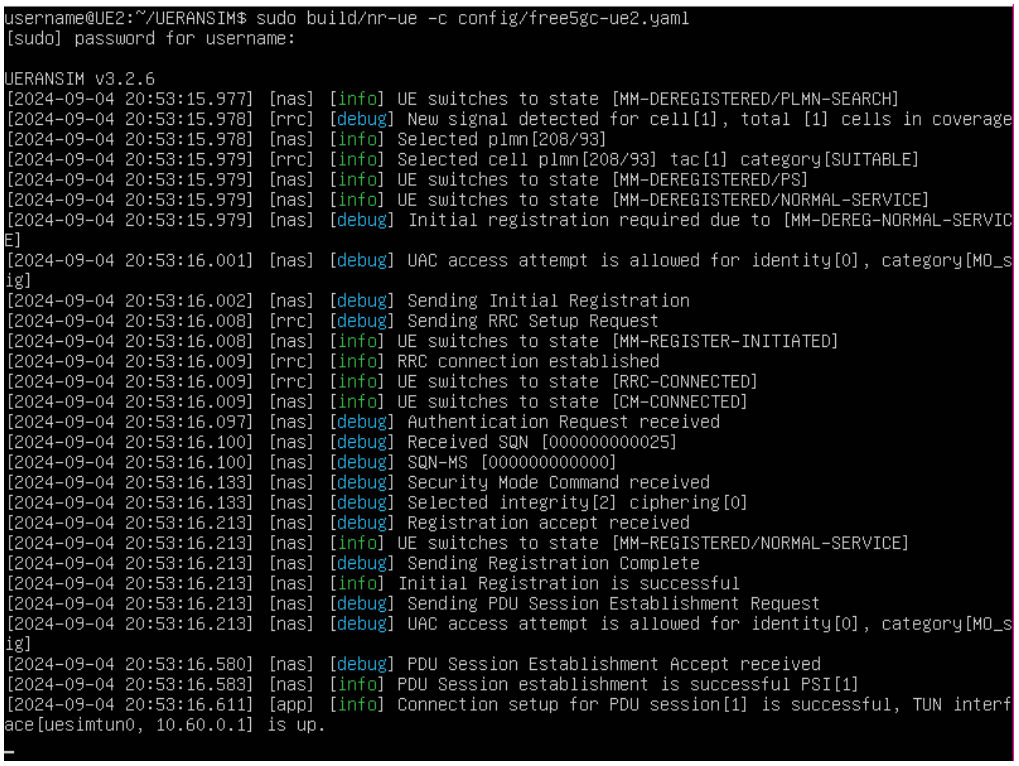

Ilustración 66: Ejecución UERANSIM en UE2

En el UE3 arrancamos la interfaz y ejecutamos el mismo archivo que en UE1 y UE2.

```
[2024-08-02 11:41:48.575] [rrc] [debug] New signal detected for cell[2], total [1] cells in coverage
[2024-08-02 11:41:48.575] [nas] [info] Selected plmn[208/93]
[2024-08-02 11:41:48.575] [rrc] [info] Selected cell plmn[208/93] tac[1] category[SUITABLE]
[2024-08-02 11:41:48.575] [nas] [info] UE switches to state [MM-DEREGISTERED/PS]
[2024-08-02 11:41:48.575] [nas] [info] UE switches to state [MM-DEREGISTERED/NORMAL-SERVICE]
[2024-08-02 11:41:48.575] [nas] [debug] Initial registration required due to [MM-DEREG-NORMAL-SERVICE]
[2024-08-02 11:41:48.580] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2024-08-02 11:41:48.580] [nas] [debug] Sending Initial Registration
[2024-08-02 11:41:48.581] [rrc] [debug] Sending RRC Setup Request
[2024-08-02 11:41:48.581] [nas] [info] UE switches to state [MM-REGISTER-INITIATED]
[2024-08-02 11:41:48.582] [rrc] [info] RRC connection established
[2024-08-02 11:41:48.582] [rrc] [info] UE switches to state [RRC-CONNECTED]
[2024-08-02 11:41:48.582] [nas] [info] UE switches to state [CM-CONNECTED]
[2024-08-02 11:41:48.673] [nas] [debug] Authentication Request received
[2024-08-02 11:41:48.685] [nas] [debug] Received SQN [16F3B3F70FC2]
[2024-08-02 11:41:48.685] [nas] [debug] SQN-MS [000000000000]
[2024-08-02 11:41:48.685] [nas] [debug] Sending Authentication Failure due to SQN out of range
[2024-08-02 11:41:48.718] [nas] [debug] Authentication Request received
[2024-08-02 11:41:48.719] [nas] [debug] Received SQN [000000000021]
[2024-08-02 11:41:48.719] [nas] [debug] SQN-MS [000000000000]
[2024-08-02 11:41:48.755] [nas] [debug] Security Mode Command received
[2024-08-02 11:41:48.755] [nas] [debug] Selected integrity[2] ciphering[0]
[2024-08-02 11:41:48.841] [nas] [debug] Registration accept received
[2024-08-02 11:41:48.841] [nas] [info] UE switches to state [MM-REGISTERED/NORMAL-SERVICE]
[2024-08-02 11:41:48.841] [nas] [info] UE switches to state [5U1-UPDATED]
[2024-08-02 11:41:48.841] [nas] [debug] Sending Registration Complete
[2024-08-02 11:41:48.841] [nas] [info] Initial Registration is successful
[2024-08-02 11:41:48.841] [nas] [debug] Sending PDU Session Establishment Request
[2024-08-02 11:41:48.847] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2024-08-02 11:41:49.163] [nas] [debug] PDU Session Establishment Accept received
[2024-08-02 11:41:49.163] [nas] [info] PDU Session establishment is successful PSI[1]
[2024-08-02 11:41:49.212] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[u
esimtun0, 10.60.0.1] is up.
```

Ilustración 67: Ejecución UERANSIM en UE3

Cuando se conectan los terminales, también se puede observar información en las terminales de los gNB .

```
UERANSIM v3.2.6
[2024-09-03 18:35:02.994] [sctp] [info] Trying to establish SCTP connection... (10.8.0.1:38412)
[2024-09-03 18:35:03.062] [sctp] [info] SCTP connection established (10.8.0.1:38412)
[2024-09-03 18:35:03.063] [sctp] [debug] SCTP association setup ascId[3]
[2024-09-03 18:35:03.063] [ngap] [debug] Sending NG Setup Request
[2024-09-03 18:35:03.156] [ngap] [debug] NG Setup Response received
[2024-09-03 18:35:03.157] [ngap] [info] NG Setup procedure is successful
[2024-09-03 18:36:54.241] [rrc] [debug] UE[1] new signal detected
[2024-09-03 18:37:03.036] [rrc] [info] RRC Setup for UE[1]
[2024-09-03 18:37:03.038] [ngap] [debug] Initial NAS message received from UE[1]
[2024-09-03 18:37:03.260] [ngap] [debug] Initial Context Setup Request received
[2024-09-03 18:37:03.647] [ngap] [info] PDU session resource(s) setup for UE[1] count[1]
[2024-09-03 18:38:21.999] [rrc] [debug] UE[2] new signal detected
[2024-09-03 18:38:23.040] [rrc] [info] RRC Setup for UE[2]
[2024-09-03 18:38:23.042] [ngap] [debug] Initial NAS message received from UE[2]
[2024-09-03 18:38:23.171] [ngap] [debug] Initial Context Setup Request received
[2024-09-03 18:38:23.577] [ngap] [info] PDU session resource(s) setup for UE[2] count[1]
```

Ilustración 68: Establecimiento de conexión UE1 y UE2 a través de gNB1

```
username@gNB2:~/UERANSIM$ build/nr-gnb -c config/free5gc-gnb2.yaml
UERANSIM v3.2.6
[2024-08-02 11:52:03.864] [sctp] [info] Trying to establish SCTP connection... (10.8.0.1:38412)
[2024-08-02 11:52:03.914] [sctp] [info] SCTP connection established (10.8.0.1:38412)
[2024-08-02 11:52:03.914] [sctp] [debug] SCTP association setup ascId[4]
[2024-08-02 11:52:03.915] [ngap] [debug] Sending NG Setup Request
[2024-08-02 11:52:04.002] [ngap] [debug] NG Setup Response received
[2024-08-02 11:52:04.003] [ngap] [info] NG Setup procedure is successful
[2024-08-02 11:52:04.733] [rrc] [debug] UE[1] new signal detected
[2024-08-02 11:52:04.740] [rrc] [info] RRC Setup for UE[1]
[2024-08-02 11:52:04.741] [ngap] [debug] Initial NAS message received from UE[1]
[2024-08-02 11:52:04.999] [ngap] [debug] Initial Context Setup Request received
[2024-08-02 11:52:05.321] [ngap] [info] PDU session resource(s) setup for UE[1] count[1]
```

Ilustración 69: Establecimiento de conexión UE3 a través de gNB2

Si la ejecución ha ido bien se observar una nueva interfaz llamada ‘uesimtun0’ en los tres terminales.

```
ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.31.163  netmask 255.255.255.0  broadcast 192.168.31.255
        inet6 fe80::a00:27ff:fe00:322  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:00:03:22  txqueuelen 1000  (Ethernet)
        RX packets 11259  bytes 681315 (681.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 209  bytes 15460 (15.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 17816  bytes 1268680 (1.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17816  bytes 1268680 (1.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

uesimtun0: flags=369<UP,POINTOPOINT,NOTRAILERS,RUNNING,PROMISC>  mtu 1400
        inet 10.60.0.3  netmask 255.255.255.255  destination 10.60.0.3
        inet6 fe80::96ee:477e:5610:9ed1  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

username@UE1:~/UERANSIM$
```

Ilustración 70: Interfaz uesimtun0 en UE1

```
ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.31.164  netmask 255.255.255.0  broadcast 192.168.31.255
        inet6 fe80::a00:27ff:fed4:aef5  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d4:ae:f5  txqueuelen 1000  (Ethernet)
        RX packets 11406  bytes 700077 (700.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 479  bytes 35654 (35.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 17392  bytes 1236144 (1.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17392  bytes 1236144 (1.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

uesimtun0: flags=369<UP,POINTOPOINT,NOTRAILERS,RUNNING,PROMISC>  mtu 1400
        inet 10.60.0.4  netmask 255.255.255.255  destination 10.60.0.4
        inet6 fe80::be63:69f3:de15:8fe7  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

username@UE2:~/UERANSIM$
```

Ilustración 71: Interfaz uesimtun0 en UE2

En el UE3 desde la interfaz gráfica arrancamos el cliente WARP:

```
warp-cli connect
```

De ésta manera podremos observar que a parte de la interfaz *'uesimtun0'*, cuenta con otra nueva interfaz, *'CloudflareWARP'* la cual se mostrará y se podrá usar cuando esté activo.

```
username@UE3:~$ ifconfig
CloudflareWARP: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1280
        inet 172.16.0.2  netmask 255.255.255.255  destination 172.16.0.2
        inet6 2606:4700:110:89e6:dde1:98a7:89f1:a0eb  prefixlen 128  scopeid 0x0<global>
        inet6 fe80::560:623e:e339:b850  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 9  bytes 4933 (4.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15  bytes 1492 (1.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.31.166  netmask 255.255.255.0  broadcast 192.168.31.255
        inet6 fe80::a00:27ff:fe90:ed26  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:90:ed:26  txqueuelen 1000  (Ethernet)
        RX packets 18391  bytes 11886936 (11.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13060  bytes 3150849 (3.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 7396  bytes 701902 (701.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7396  bytes 701902 (701.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

uesimtun0: flags=369<UP,POINTOPOINT,NOTRAILERS,RUNNING,PROMISC>  mtu 1400
        inet 10.60.0.2  netmask 255.255.255.255  destination 10.60.0.2
        inet6 fe80::8434:8596:6375:d2cc  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 688 (688.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

username@UE3:~$
```

Ilustración 72: Interfaces uesimtun0 y CloudflareWARP

Para comprobar el correcto funcionamiento de la red se realizan una serie de pruebas que se detallan en profundidad en el siguiente apartado.

4.14.1 Liberación de recursos

A continuación se muestra el proceso de liberación de recursos, primero se muestra desde los gNB se pierde la señal de los UE.

```
19:11:02.652] [rls] [debug] UE[1] signal lost
```

Ilustración 73: Fin de la ejecución de UE1 desde gNB1

```
19:11:29.401] [rls] [debug] UE[2] signal lost
```

Ilustración 74: Fin de la ejecución de UE2 desde gNB1

```
19:14:39.366] [rls] [debug] UE[1] signal lost
```

Ilustración 75: Fin de ejecución de UE3 desde gNB3

También se puede observar información en los gNB:

```
-09-03T19:00:25Z [INFO][AMF][NGAP] Handle SCTP Notification[addr: 10.8.0.6:59360]
-09-03T19:00:25Z [INFO][AMF][NGAP][10.8.0.6:59360] SCTP_SHUTDOWN_EVENT notification, close the connection
-09-03T19:00:25Z [INFO][AMF][NGAP][10.8.0.6:59360] Remove RAN Context[ID: <PlmnID: {Mcc:208 Mnc:93}, GNbID: 00000001>]
```

Ilustración 76: Pérdida de conexión en Free5gc de gNB2

```
2024-09-03T18:47:56Z [INFO][AMF][NGAP] Handle SCTP Notification[addr: 10.8.0.10:37705]
2024-09-03T18:47:56Z [INFO][AMF][NGAP][10.8.0.10:37705] SCTP_SHUTDOWN_EVENT notification, close the connection
2024-09-03T18:47:56Z [INFO][AMF][NGAP][10.8.0.10:37705] Remove RAN Context[ID: <PlmnID: {Mcc:208 Mnc:93}, GNbID: 00000001>]
```

Ilustración 77: Pérdida de conexión en Free5gc de gNB1

5. Resultados

En este apartado se realizan una serie de pruebas para verificar el correcto funcionamiento de la arquitectura. Inicialmente, se realiza un ping a través de la interfaz configurada que es *uesimtun0.*

```
username@UE1:~$ ping -c 4 -I uesimtun0 www.google.com
PING www.google.com (142.250.200.68) from 10.60.0.2 uesimtun0: 56(84) bytes of data.
64 bytes from mad07s24-in-f4.1e100.net (142.250.200.68): icmp_seq=1 ttl=110 time=43.8 ms
64 bytes from mad07s24-in-f4.1e100.net (142.250.200.68): icmp_seq=2 ttl=110 time=42.7 ms
64 bytes from mad07s24-in-f4.1e100.net (142.250.200.68): icmp_seq=3 ttl=110 time=45.8 ms
64 bytes from mad07s24-in-f4.1e100.net (142.250.200.68): icmp_seq=4 ttl=110 time=42.8 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 42.652/43.762/45.848/1.279 ms
```

Ilustración 78: Ping desde UE1 por interfaz uesimtun0

```
username@UE2:~/UERANSIM$ ping -c 4 -I uesimtun0 www.google.com
PING www.google.com (142.250.200.100) from 10.60.0.8 uesimtun0: 56(84) bytes of data.
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=1 ttl=111 time=42.6 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=2 ttl=111 time=42.0 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=3 ttl=111 time=43.4 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=4 ttl=111 time=43.0 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 41.958/42.749/43.439/0.553 ms
```

Ilustración 79: Ping desde UE2 por interfaz uesimtun0

```
username@UE3:~$ ping -c 4 -I uesimtun0 www.google.com
PING www.google.com (142.250.200.100) from 10.60.0.1 uesimtun0: 56(84) bytes of data.
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=1 ttl=111 time=41.4 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=2 ttl=111 time=43.8 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=3 ttl=111 time=48.5 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=4 ttl=111 time=43.8 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 41.377/44.370/48.520/2.590 ms
```

Ilustración 80: Ping desde UE3 por interfaz uesimtun0

En las imágenes se muestra como se realiza el envío de 4 paquetes a Google por la interfaz *uesimtun0,* creada por UERANSIM.

A continuación, se intenta ejecutar un ping utilizando la interfaz configurada para Cloudflare sin estar conectado al cliente de Cloudflare. Si la configuración es correcta, el ping no debería completarse.

```
username@UE3:~$ ping -c 4 -I CloudflareWARP www.google.com
ping: SO_BINDTODEVICE CloudflareWARP: No such device
```

Ilustración 81: Ping fallido desde UE3 por interfaz CloudflareWARP

A continuación procedemos a conectar el cliente de Cloudflare.

```
username@UE3:~$ warp-cli connect
Success
```

Ilustración 82: Conexión a Cloudflare WARP

A continuación, comprobamos que al realizar de nuevo el ping, se envían los paquetes por la interfaz *CloudflareWARP.*

```
username@UE3:~$ ping -c 4 -I CloudflareWARP www.google.com
PING www.google.com(mad07s25-in-x04.1e100.net (2a00:1450:4003:811::2004)) from 2606:4700:110:84ce:9b16:cb37:6dec:193b Cloudflare
WARP: 56 data bytes
64 bytes from mad07s25-in-x04.1e100.net (2a00:1450:4003:811::2004): icmp_seq=1 ttl=117 time=18.1 ms
64 bytes from mad07s25-in-x04.1e100.net (2a00:1450:4003:811::2004): icmp_seq=2 ttl=117 time=7.85 ms
64 bytes from mad07s25-in-x04.1e100.net (2a00:1450:4003:811::2004): icmp_seq=3 ttl=117 time=7.63 ms
64 bytes from mad07s25-in-x04.1e100.net (2a00:1450:4003:811::2004): icmp_seq=4 ttl=117 time=7.91 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 7.633/10.380/18.128/4.474 ms
```

Para continuar con las pruebas de Cloudflare y demostrar que el certificado funciona en Firefox de manera adecuada, procedemos en la plataforma de Kyndryl a bloquear el tráfico de las siguientes dos páginas *netflix.com* y *marca.com*.

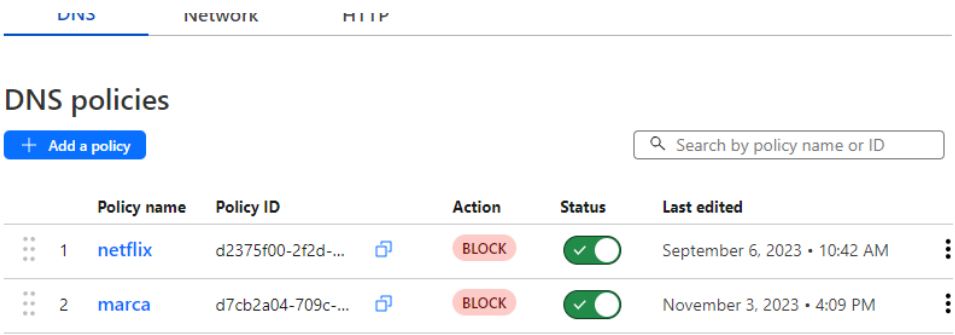

Ilustración 83: Políticas de DNS Cloudflare

Desde la máquina virtual, tras ejecutar Firefox, se acceden a dos sitios específicos y se verifica que el acceso está bloqueado. La pantalla que aparece es completamente editable mediante la aplicación de Cloudflare de Kyndryl.

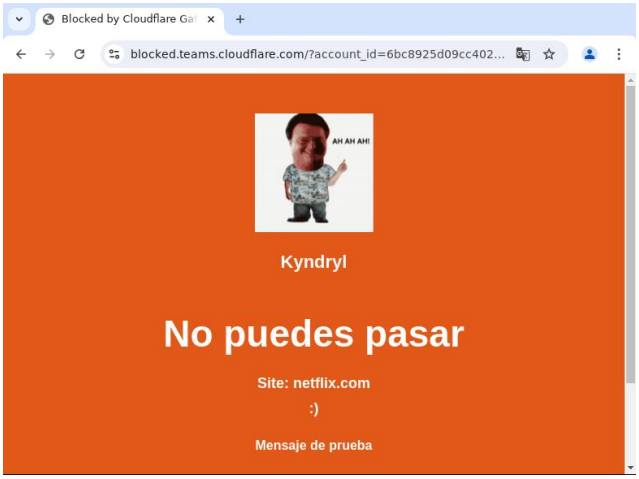

Ilustración 84: Conexión fallida con netflix.com

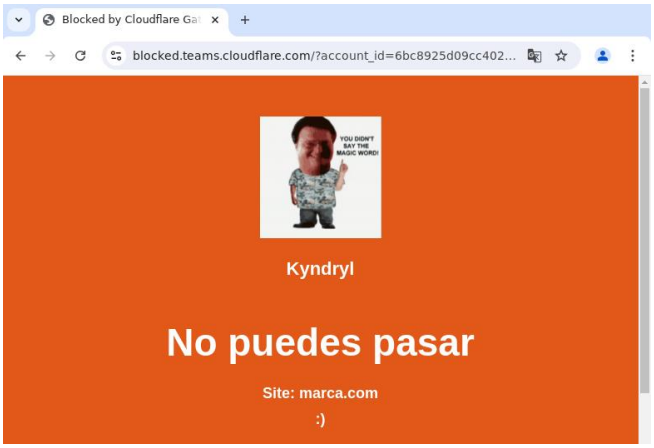

Ilustración 85: Conexión fallida con marca.com

6. Presupuesto

En este apartado de muestra el presupuesto empleado para el desarrollo de este proyecto, incluyendo desde el coste de los recursos para desplegar la infraestructura, software utilizado, la capa de seguridad y recursos humanos

6.1 Coste núcleo 5G en la nube

Para el despliegue del núcleo 5G, se ha utilizado una máquina virtual Linux para implementar la infraestructura. En el portal de Azure, en la sección *"Cost Management + Billing"*, se pueden visualizar estadísticas detalladas de los gastos asociados a la suscripción. Estos gastos incluyen todos los recursos vinculados a la suscripción, como la máquina virtual, el grupo de recursos y la red virtual.

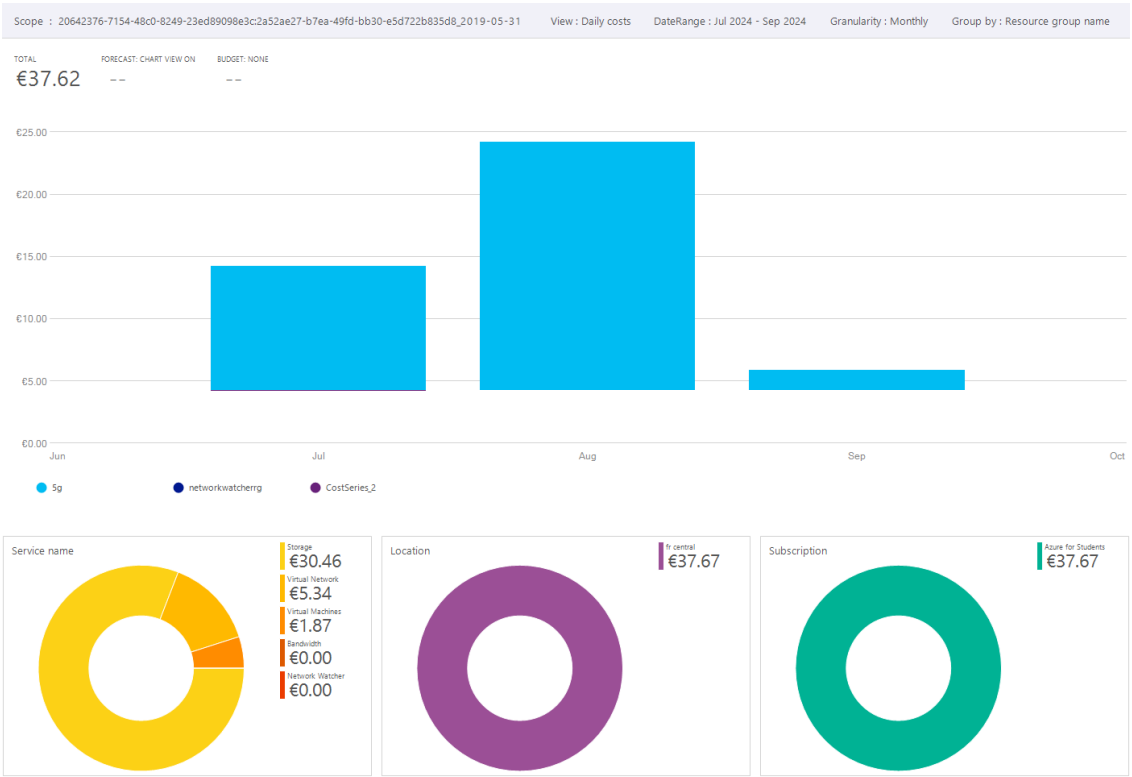

Ilustración 86: Parte de los costes mensuales de Azure

En la imagen se presentan las estadísticas generadas automáticamente por la plataforma para los últimos tres meses. El proyecto comenzó en enero y finalizó en septiembre, lo que permite observar el gasto mensual de los recursos utilizados. Este gasto varía en función del uso que se haya realizado cada mes. En promedio, el coste mensual ha sido de 23,81 €, cifra que se empleará como referencia para calcular el coste total del desarrollo del proyecto a lo largo de los 9 meses de duración.

Para la suscripción de Azure, se ha utilizado un crédito para estudiantes proporcionado a los alumnos de la UPM, el cual incluye 200 $ destinados al uso de los servicios ofrecidos por la plataforma.

6.2 Coste capa de seguridad

En este proyecto se ha utilizado la tecnología de Cloudflare para añadir una capa adicional de seguridad, sobre la cual se podrán implementar soluciones más avanzadas en el futuro. Es importante señalar que los costos relacionados con el uso de Cloudflare no se han contemplado dentro del presupuesto de este proyecto, ya que el servicio fue suministrado y financiado por la empresa Kyndryl, la cual, en su calidad de partner de Cloudflare, no ha proporcionado acceso a dichos gastos.

Para servicios más personalizados, se pueden consultar los planes disponibles, tanto para particulares como para empresas, en la página web de Cloudflare [47].

6.3 Coste de recursos software y hardware

Todo el software utilizado en este proyecto es gratuito y en su mayoría proviene de proyectos de código abierto. No se ha utilizado hardware especializado; sin embargo, para la ejecución del proyecto se ha requerido un ordenador personal, cuyo costo se incluirá en el presupuesto.

6.4 Coste de recursos humanos

Para calcular el coste de mano de obra, se ha considerado que el coste aproximado de un ingeniero de telecomunicaciones en España es de 20 euros por hora [48].

El número total de horas trabajadas, estimado en el anteproyecto, es de aproximadamente 315 horas. Con estos datos, el coste total de la mano de obra asciende a 6.300 euros.

6.5 Coste total

Recurso	Descripción	Coste total
Hardware	Portatil Intel Core i7-1360P/32GB/1T	2.083€
Recursos de Azure	Precio estimado al mes 23,81€ x 9 meses	214.29€
Free5GC	implementación	0€
OpenVPN	implementación	0€
Oracle VirtualBox	Herramienta de virtualización	0€
Ubuntu 20.04.2 TLS	5 máquinas virtuales con ese sistema operativo	0€
MobaXterm	Herramienta de escritorio remoto	0€
Cloudflare WARP Client	Cliente de cloudflare para tener acceso a sus servicios	0€
TOTAL		2.297,29€

7. Impacto del proyecto

En este capítulo se identifican las implicaciones sociales, seguridad, ambientales, económicas y tecnológicas que están asociadas a este proyecto.

7.2 Identificación de impactos

Este proyecto se enmarca en el ámbito de las telecomunicaciones, específicamente en computación en la nube, 5G y seguridad. Aunque estas tecnologías han avanzado considerablemente en los últimos años, se anticipa que su expansión global continuará, abarcando también áreas menos desarrolladas en el futuro. Los posibles impactos asociados a estas tecnologías y al proyecto son diversos e incluyen:

- **Implicaciones sociales:** el despliegue de tecnología 5G tiene el potencial de reducir la brecha digital al mejorar el acceso a conectividad avanzada en áreas remotas o desfavorecidas, aunque pueden persistir barreras económicas o geográficas que limiten este acceso. La implementación de 5G en la nube optimiza la eficiencia y velocidad de la red, beneficiando a los usuarios con menor latencia y mayor capacidad en los servicios de telecomunicaciones.

- **Implicaciones de seguridad**: El despliegue de un núcleo de red 5G en la nube conlleva riesgos de seguridad si no se implementan medidas adecuadas. Aunque la integración de Cloudflare contribuye a mitigar estos riesgos, es muy importante proteger los datos sensibles de los usuarios para evitar brechas de seguridad y ciberataques.

- **Implicaciones ambientales**: presenta implicaciones ambientales significativas, como el incremento del consumo energético en centros de datos y la generación de residuos electrónicos debido a la obsolescencia de dispositivos. No obstante, al optimizar recursos en la nube, se puede mitigar el impacto ambiental asociado con la fabricación de hardware. Además, la adopción de 5G puede fomentar la eficiencia en otras industrias, reduciendo emisiones y promoviendo el teletrabajo, lo cual disminuye los desplazamientos y, en consecuencia, las emisiones de gases contaminantes.

- **Implicaciones económicas:** el despliegue de la tecnología 5G genera nuevas oportunidades de empleo y fomenta el desarrollo de negocios en diversas industrias, desde el Internet de las Cosas (IoT) hasta las telecomunicaciones, lo que contribuye al crecimiento económico.

- **Implicaciones tecnológicas**: la implementación de tecnología 5G mejora la escalabilidad y flexibilidad de la infraestructura de red, facilitando un manejo más eficiente de grandes volúmenes de tráfico de datos. La virtualización de funciones de red (NFV) y el uso de tecnologías en la nube optimizan tanto la gestión como el despliegue de redes. Sin embargo, este avance presenta desafíos, como la necesidad

de garantizar baja latencia y conectividad confiable, así como la integración con equipos locales como gNBs y terminales. Además, la incorporación de capas de seguridad adicionales, como Cloudflare para mitigar riesgos de posibles ataques.

8. Conclusiones

En este apartado se detallan las conclusiones del proyecto y la forma en que se han alcanzado los objetivos establecidos. También se incluyen las propuestas para trabajos futuros que permitirán la continuación y ampliación del proyecto.

8.1 Fases del proyecto y objetivos

En este proyecto se ha implementado una red 5G con el núcleo de red alojado en la nube y el resto de la infraestructura en local. La implementación se ha llevado a cabo en cuatro fases:

- **Selección del entorno**: Se identificó y optó por una solución en la nube para el núcleo 5G utilizando Free5GC.
- **Implementación de la infraestructura local**: Se desplegó la infraestructura necesaria en las instalaciones locales.
- **Conexión entre infraestructuras:** Se estableció la conexión entre la infraestructura en la nube y la local, asegurando la interoperabilidad entre ambos entornos.
- **Mejora de la seguridad**: Se añadió una capa adicional de seguridad, con la opción de ampliar funcionalidades mediante la plataforma Cloudflare.

Los objetivos que se establecieron para este proyecto fueron los siguientes:

- **Proporcionar los servicios básicos de un núcleo de red 5G desde un despliegue en un proveedor cloud:** este objetivo se ha logrado mediante la implementación de Free5GC en una máquina virtual desplegada en Azure.
  Algunas de las principales dificultades de esta implementación implican la correcta configuración de todos los ficheros para que sean capaces de poder comunicarse con el plano de usuario que se encuentra en local, teniendo en cuenta la compatibilidad de versiones de las tecnologías utilizadas para el despliegue así como la versión de la propia máquina Linux en la que se despliega el núcleo.

- **Asegurar la visibilidad de dichos servicios desde fuera del perímetro de la nube para garantizar su acceso por parte de otros servicios externos**: se ha conseguido que los dispositivos locales, implementados en el ordenador local, puedan acceder a internet y utilizar los servicios alojados en la nube.

- **Ofrecer los servicios del núcleo de una forma que garantice su seguridad**: esto se consigue añadiendo una capa adicional con Cloudflare en uno de los terminales, UE3.

  Sin embargo, esto conlleva dificultades adicionales, ya que es necesario asegurar que la comunicación entre el núcleo y el terminal esté completamente protegida sin comprometer el rendimiento. Cuando los terminales son de uso corporativo, el riesgo es aún mayor, ya que se maneja información sensible y confidencial, lo que requiere

un enfoque más riguroso en la gestión de permisos, autenticación y protección de los datos frente a amenazas externas.

- **Diseñar un conjunto de pruebas para validar el despliegue y su seguridad**: en el apartado 5 se presentan pruebas que demuestran el correcto funcionamiento de la red 5G creada y de la capa adicional de seguridad proporcionada por Cloudflare.

8.2 Trabajos futuros

A continuación se describen los posibles trabajos futuros para este proyecto:

- **Despliegue de más UPFs en máquinas virtuales**: se contempla la expansión de la infraestructura con la adición de más UPFs (User Plane Functions) para soportar funciones adicionales y mejorar la capacidad de la red.
- **Mejora de la seguridad**: mediante la incorporación de un sistema de autenticación de doble factor para los terminales, lo que permitirá un control más riguroso de los accesos de los usuarios. Esta mejora se puede implementar a través de Cloudflare, que también proporcionará registros detallados de los accesos y usuarios para un control más exhaustivo.
- **Automatización del despliegue y configuración**: con la automatización de la implementación y configuración de la infraestructura, tanto local como en la nube, mediante el uso de plataformas de automatización como Ansible. Esto optimizará la gestión y reducirá los posibles errores manuales.

9. Referencias

[1] R. Hat, «5G: ¿qué es, cómo funciona y por qué es importante?,» 22 03 2021. [En línea]. Available: https://www.redhat.com/es/topics/5g-networks/what-is-5g. [Último acceso: 16 07 2024].

[2] E. Commission, «Seguridad de las redes 5G: Preguntas y respuestas sobre el conjunto de instrumentos de la UE,» 29 01 2020. [En línea]. Available: https://ec.europa.eu/commission/presscorner/api/files/document/print/es/qanda_20_127/QANDA_20_127_ES.pdf. [Último acceso: 16 07 2024].

[3] V. España, «2G, sistema digital en redes móviles,» 06 11 2023. [En línea]. Available: https://www.universidadviu.com/es/actualidad/nuestros-expertos/2g-sistema-digital-en-redes-moviles.

[4] H. Remmert, «What is 5G network architecture?,» 19 03 2021. [En línea]. Available: https://es.digi.com/blog/post/5g-network-architecture.

[5] «¿Cuál es la función de gNB en 5G? | TELETOPIX.ORG,» 31 03 2024. [En línea]. Available: https://teletopix.org/es/cual-es-la-funcion-de-gnb-en-5g/.

[6] «GitHub,» [En línea]. Available: https://github.com/aligungr/UERANSIM.

[7] «OpenAirInterface,» [En línea]. Available: https://openairinterface.org/.

[8] «free5GC,» [En línea]. Available: https://free5gc.org/.

[9] «Open5gs,» [En línea]. Available: https://open5gs.org/.

[10] M. Alonso, «Qué es la nube en Internet y cómo sacarle partido - Asana,» 7 02 2024. [En línea]. Available: https://asana.com/es/resources/what-is-the-cloud.

[11] M. Alonso, «Qué es Business Intelligence (BI) en la gestión de proyectos [2024] • Asana,» asana, 10 02 2024. [En línea]. Available: https://asana.com/es/resources/business-intelligence.

[12] I. Amazon Web Services, «what is aws,» [En línea]. Available: https://aws.amazon.com/es/what-is-aws/.

[13] M. (. Gualda, «¿Qué es Microsoft Azure? ¿Cómo funciona? | Tecon,» 13 01 2021. [En línea]. Available: https://www.tecon.es/que-es-microsoft-azure-como-funciona.

[14] «IBM Cloud Docs,» IBM Cloud, [En línea]. Available: https://cloud.ibm.com/docs/billing-usage?topic=billing-usage-overview&locale=es.

[15] «Private Cloud - Programming Trends,» 11 08 2020. [En línea]. Available: https://www.programmingtrends.com/2020/08/private-cloud.html.

[16] H. Safa, «What is Public Cloud Computing? – InspirationSeek.com,» 11 02 2016. [En línea].

[17] D. d. m. d. r. compartida, «Microsoft Learn: Build skills that open doors in your career,» [En línea]. Available: https://learn.microsoft.com/es-es/training/modules/describe-cloud-compute/4-describe-shared-responsibility-model.

[18] Kali, «Nube comunitaria - Techinfo,» [En línea]. Available: https://techinfo.wiki/nube-comunitaria/.

[19] «Modelo de nube comunitaria - Stack,» 10 12 2020. [En línea]. Available: https://isolution.pro/es/t/cloud-computing/cloud-computing-community-cloud-model/modelo-de-nube-comunitaria.

[20] G. Singh, «What is Multi-Cloud and Hybrid Cloud | Use-Cases,» 03 06 2022. [En línea]. Available: https://www.xenonstack.com/insights/multi-cloud-vs-hybrid-cloud.

[21] D. Pardo, «Pandora FMS - The Monitoring Blog,» 03 08 2020. [En línea]. Available: https://pandorafms.com/blog/es/servicios-en-la-nube.

[22] A. R. TerryLanfear, «Microsoft Learn: Build skills that open doors in your career,» 20 10 2023. [En línea]. Available: https://learn.microsoft.com/es-es/azure/security/fundamentals/shared-responsibility.

[23] M. Duo, «Función como Servicio (FaaS): Todo lo que necesitas saber,» 13 06 2022. [En línea]. Available: https://kinsta.com/es/blog/funcion-como-servicio/.

[24] L. 1. p. v. d. l. c. e. l. nube, «Oracle | Cloud Applications and Cloud Platform,» [En línea]. Available: https://www.oracle.com/es/cloud/what-is-cloud-computing/top-10-benefits-cloud-computing/.

[25] «Implementación de redes 5G privadas en Azure,» 07 06 2024. [En línea]. Available: https://learn.microsoft.com/es-es/azure/architecture/industries/telecommunications/deploy-private-mobile-network.

[26] I. d. r. 5. p. e. A. -. A. A. Center, Microsoft Learn: Build skills that open doors in your career, 11 07 2023. [En línea]. Available: https://learn.microsoft.com/es-

es/azure/architecture/industries/telecommunications/deploy-private-mobile-network.

[27] «¿Qué es el MEC público de Azure?,» 22 11 2022. [En línea]. Available: https://learn.microsoft.com/es-ES/azure/public-multi-access-edge-compute-mec/overview.

[28] «¿Qué es Google Cloud y cuáles son sus ventajas? - IMMUNE,» Immune Technology Institute, [En línea]. Available: https://immune.institute/blog/que-es-google-cloud-y-sus-ventajas/.

[29] «Compare Flexible & Annual/Fixed-Term payment plans - Google Workspace Admin Help,» [En línea]. Available: https://support.google.com/a/answer/1247360?sjid=2589514600744740708-EU.

[30] «¿Qué es AWS? - Computación en la nube con Amazon Web Services,» Amazon Web Services, Inc., [En línea]. Available: https://aws.amazon.com/es/what-is-aws/.

[31] AWS, «¿Qué es una red de entrega de contenido (CDN)?,» [En línea]. Available: https://aws.amazon.com/es/what-is/cdn/.

[32] «¿Qué es un ataque DDoS? | IBM,» IBM, [En línea]. Available: https://www.ibm.com/es-es/topics/ddos.

[33] «Descubriendo Cloudflare: ¿Qué es y para qué sirve? | Kuex,» Guías y tutoriales tecnológicos para personas y empresas | Kuex, [En línea]. Available: https://kuex.es/descubriendo-cloudflare-que-es-y-para-que-sirve/#:~:text=CloudFlare%20es%20una%20plataforma%20de%20seguridad%2C%20rendimiento%20y,mejora%20la%20velocidad%20y%20la%20eficiencia%20del%20sitio..

[34] «Download WARP | Cloudflare Zero Trust docs,» [En línea]. Available: https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/download-warp/.

[35] Ubuntu, «http://ubuntu-manual.org/,» [En línea].

[36] «Install free5gc - free5GC,» [En línea]. Available: https://free5gc.org/guide/3-install-free5gc/#a-prerequisites.

[37] Microsoft, «Create and deploy VM application packages - Azure Virtual Machines,» 22 08 2022. [En línea]. Available: https://learn.microsoft.com/es-es/azure/virtual-machines/vm-applications-how-to?tabs=portal.

[38] Microsoft, «Cómo funciona Azure Resource Manager - Cloud Adoption Framework,» [En línea]. Available: https://learn.microsoft.com/es-es/azure/cloud-adoption-framework/get-started/how-azure-resource-manager-works. [Último acceso: 2023].

[39] «Environment - free5GC,» [En línea]. Available: https://free5gc.org/guide/Environment/.

[40] A. Güngör, «Installaton UERANSIM - GitHub,» 24 06 2021. [En línea]. Available: https://github.com/aligungr/UERANSIM/wiki/Installation.

[41] «How To Guide: Set Up & Configure OpenVPN Client/server VPN | OpenVPN,» [En línea]. Available: https://openvpn.net/community-resources/how-to/#openvpn-quickstart.

[42] «Home - Easy RSA,» [En línea]. Available: https://easy-rsa.readthedocs.io/en/latest/.

[43] D. Gitlan, «¿Qué es OpenSSL y cómo funciona? - SSL Dragon,» 04 04 2024. [En línea]. Available: https://www.ssldragon.com/es/blog/que-es-openssl/#what-is-openssl.

[44] Qué es el intercambio de claves Diffie-Hellman y cómo funciona, [En línea]. Available: https://ciberseguridad.com/guias/recursos/intercambio-claves-diffie-hellman/#%C2%BFQue_es_el_intercambio_de_claves_Diffie-Hellman.

[45] «UERANSIM installation,» 24 06 2021. [En línea]. Available: https://github.com/aligungr/UERANSIM/wiki/Installation.

[46] GitHub - How to install Cloudflare WARP, [En línea]. Available: https://github.com/MasterCode112/How-To-Install-and-use-Cloudflare-WARP-on-Kali-Linux.

[47] «Planes de Precios | Cloudflare,» [En línea]. Available: https://www.cloudflare.com/es-es/plans/.

[48] «Sueldo de Ingeniero/a de Telecomunicaciones (2023) | Kiwi remoto,» [En línea]. Available: https://www.kiwiremoto.com/sueldo/ingeniero-de-telecomunicaciones/#:~:text=%C2%BFCu%C3%A1nto%20gana%20por%20hora%3F%20Un%2Fa%20Ingeniero%2Fa%20de,Telecomunicaciones%20cobra%20de%20media%2020%E2%82%AC%20brutos%20por%20hora..

[49] J. M. Ousmane Diallo, «Real-time data management on wireless sensor networks: A survey,» *Journal of Network and Computer Applications,* 2011.

[50] Ó. Ortiz, A. B. Garcia, R. Capilla, J. Bosch y M. Hinchey, *Runtime Variability for Dynamic Reconfiguration in Wireless Sensor Network Product Lines.*

[51] E. C. -. I. S. a. M. DG, «Internet of Things. Strategic Research Roadmap,» 2009.

[52] «Mono Documentation,» [En línea]. Available: http://docs.go-mono.com/.

[53] IEEE, «IEEE 802.15.4g™-2012,» [En línea]. Available: http://standards.ieee.org/getieee802/download/802.15.4g-2012.pdf.

[54] R. M. Castejón, «Interconexión de redes de sensores inalámbricos 802.15.4 en localizaciones remotas,» *PFC,* pp. 8,9,10, 2011.

[55] F. B. J. S. A. C. Diego Martínez, «Redes de Sensores y Actuadores Inalámbricas: Una Caracterización y Caso de Estudio para Aplicaciones Médicas en Espacios Cerrados,» *Universidad Autónoma de Occidente, Universidad Politécnica de Valencia.*

[56] E. International, «ECMAScript Language Specification. Standar 262,» [En línea]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf.

[57] Apache.org, «Manual Apache Ant,» [En línea]. Available: http://ant.apache.org/manual/intro.html.

[58] W. i. A. -. C. C. w. A. -. A. W. Services, «Amazon Web Services, Inc.,» [En línea]. Available: https://aws.amazon.com/what-is-aws/.

[59] ¿. e. M. A. ¿. f. |. Tecon, «Tecon,» [En línea]. Available: https://www.tecon.es/que-es-microsoft-azure-como-funciona/.

[60] «Qué es y para qué sirve Google Cloud Platform,» 20 08 2020. [En línea]. Available: https://www.incentro.com/es-ES/blog/que-es-google-cloud-platform.

[61] «IBM Cloud,» 15 02 2024. [En línea]. Available: https://cloud.ibm.com/docs/overview?topic=overview-whatis-platform&locale=es.

[62] «Nube comunitaria – Techinfo,» Techinfo, [En línea]. Available: https://techinfo.wiki/nube-comunitaria/.

[63] «Conoce los tipos de servicios en La Nube: SaaS, PaaS, IaaS,» Pandora FMS - The Monitoring Blog, [En línea]. Available: https://pandorafms.com/blog/es/servicios-en-la-nube/.

[64] «Función como Servicio (FaaS): Todo lo que necesitas saber,» Kinsta®, [En línea]. Available: https://kinsta.com/es/blog/funcion-como-servicio/.

[65] «Hewlett Packard Enterprise,» [En línea]. Available: https://www.hpe.com/es/es/what-is/hyperscale.html.

[66] «Microsoft Learn: Build skills that open doors in your career,» [En línea]. Available: https://learn.microsoft.com/es-ES/azure/public-multi-access-edge-compute-mec/overview.

[67] «¿Qué es una Nube Híbrida? - Sapia,» 21 09 2021. [En línea]. Available: https://www.sapia.com.pe/sin-categoria/que-es-una-nube-hibrida/.

[68] M. Alain Sultan, «3GPP – The Mobile Broadband Standard,» 08 08 2022. [En línea]. Available: https://www.3gpp.org/technologies/5g-system-overview.

Anexo A – Herramienta MobaXterm

MobaXterm es una herramienta de software para Windows utilizada en la administración de sistemas remotos y el acceso a servidores mediante una variedad de protocolos de red. En este contexto, permite establecer conexiones SSH utilizando autenticación con claves públicas y privadas.

Para utilizar MobaXterm, es necesario descargar la última versión disponible. Para iniciar una nueva sesión SSH, se debe hacer clic en el botón *"Session"* y seleccionar la opción *"SSH"*. Luego, es necesario ingresar la dirección IP pública del servidor, asignar un nombre a la conexión y especificar la ruta del archivo de la clave privada almacenada en el ordenador local. Una vez establecida la conexión, se facilita la transferencia de archivos mediante el árbol de directorios integrado.

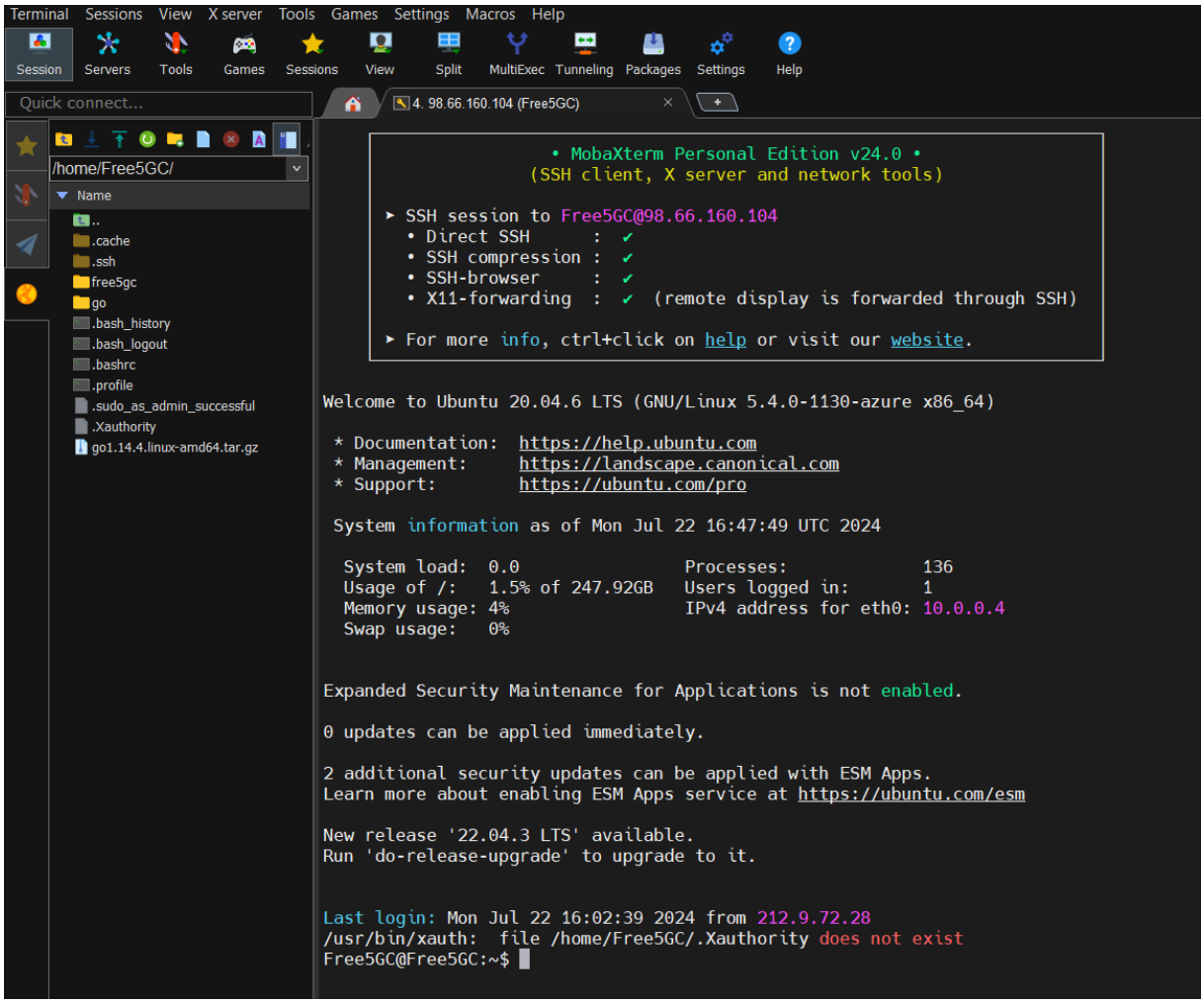

Ilustración 87: Conexión SSH a Free5GC a través de MobaXterm

Anexo B – Ficheros de configuración

A continuación se muestra el contenido de los ficheros de configuración para OpenVPN.

Fichero de configuración d**el** servidor vpnserver: *'~/etc/openvpn/server/server.conf'*

```
port 1194
proto tcp
dev tun
ca /etc/openvpn/easy-rsa/pki/ca.crt
cert /etc/openvpn/easy-
rsa/pki/issued/serverVPN.crt
key /etc/openvpn/easy-
rsa/pki/private/serverVPN.key
dh /etc/openvpn/easy-rsa/pki/dh.pem
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "route 10.0.1.0 255.255.255.0"
keepalive 10 120
cipher AES-256-CBC
user nobody
group nogroup
persist-key
persist-tun
client-to-client
status openvpn-status.log
verb 3
```

Fichero de configuración del cliente vpnclientgNB1:*'~/etc/openvpn/cliente/clientegNB1.conf'*

```
client
dev tun
proto tcp
remote 98.66.160.104 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert clientVPNgNB1.crt
key clientVPNgNB1.key
remote-cert-tls server
cipher AES-256-CBC
verb 3
```

Fichero de configuración del cliente vpnclientgNB1*:'~/etc/openvpn/cliente/clientegNB2.conf'*

```
client
dev tun
proto tcp
remote 98.66.160.104 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert clientVPNgNB2.crt
key clientVPNgNB2.key
remote-cert-tls server
cipher AES-256-CBC
verb 3
```

Fichero de configuración del AMF:

```yaml
info:
  version: 1.0.3
  description: AMF initial local configuration

configuration:
  amfName: AMF # the name of this AMF
  ngapIpList:  # the IP list of N2 interfaces on this AMF
    - 10.8.0.1 #127.0.0.18
  sbi: # Service-based interface information
    scheme: http # the protocol for sbi (http or https)
    registerIPv4: 127.0.0.18 # IP used to register to NRF
    bindingIPv4: 127.0.0.18  # IP used to bind the service
    port: 8000 # port used to bind the service
    tls: # the local path of TLS key
      pem: config/TLS/amf.pem # AMF TLS Certificate
      key: config/TLS/amf.key # AMF TLS Private key
  serviceNameList: # the SBI services provided by this AMF, refer
to TS 29.518
    - namf-comm # Namf_Communication service
    - namf-evts # Namf_EventExposure service
    - namf-mt   # Namf_MT service
    - namf-loc  # Namf_Location service
    - namf-oam  # OAM service
  servedGuamiList: # Guami (Globally Unique AMF ID) list supported
by this AMF
    # <GUAMI> = <MCC><MNC><AMF ID>
    - plmnId: # Public Land Mobile Network ID, <PLMN ID> = <MCC>
<MNC>
        mcc: 208 # Mobile Country Code (3 digits string, digit:
0~9)
        mnc: 93 # Mobile Network Code (2 or 3 digits string, digit:
0~9)
      amfId: cafe00 # AMF identifier (3 bytes hex string, range:
000000~FFFFFF)
  supportTaiList:  # the TAI (Tracking Area Identifier) list
supported by this AMF
    - plmnId: # Public Land Mobile Network ID, <PLMN ID> = <MCC>
<MNC>
        mcc: 208 # Mobile Country Code (3 digits string, digit:
0~9)
        mnc: 93 # Mobile Network Code (2 or 3 digits string, digit:
0~9)
      tac: 1 # Tracking Area Code (uinteger, range: 0~16777215)
  plmnSupportList: # the PLMNs (Public land mobile network) list
supported by this AMF
    - plmnId: # Public Land Mobile Network ID, <PLMN ID> = <MCC>
```

```
<MNC>
        mcc: 208 # Mobile Country Code (3 digits string, digit:
0~9)
        mnc: 93 # Mobile Network Code (2 or 3 digits string, digit:
0~9)
      snssaiList: # the S-NSSAI (Single Network Slice Selection
Assistance Information) list supported by this AMF
        - sst: 1 # Slice/Service Type (uinteger, range: 0~255)
          sd: 010203 # Slice Differentiator (3 bytes hex string,
range: 000000~FFFFFF)
        - sst: 1 # Slice/Service Type (uinteger, range: 0~255)
          sd: 112233 # Slice Differentiator (3 bytes hex string,
range: 000000~FFFFFF)
  supportDnnList:  # the DNN (Data Network Name) list supported by
this AMF
    - internet
  nrfUri: http://127.0.0.10:8000 # a valid URI of NRF
  security:  # NAS security parameters
    integrityOrder: # the priority of integrity algorithms
      - NIA2
      # - NIA0
    cipheringOrder: # the priority of ciphering algorithms
      - NEA0
      # - NEA2
  networkName:  # the name of this core network
    full: free5GC
    short: free
  locality: area1 # Name of the location where a set of AMF, SMF
and UPFs are located
  networkFeatureSupport5GS: # 5gs Network Feature Support IE, refer
to TS 24.501
    enable: true # append this IE in Registration accept or not
    length: 1 # IE content length (uinteger, range: 1~3)
    imsVoPS: 0 # IMS voice over PS session indicator (uinteger,
range: 0~1)
    emc: 0 # Emergency service support indicator for 3GPP access
(uinteger, range: 0~3)
    emf: 0 # Emergency service fallback indicator for 3GPP access
(uinteger, range: 0~3)
    iwkN26: 0 # Interworking without N26 interface indicator
(uinteger, range: 0~1)
    mpsi: 0 # MPS indicator (uinteger, range: 0~1)
    emcN3: 0 # Emergency service support indicator for Non-3GPP
access (uinteger, range: 0~1)
    mcsi: 0 # MCS indicator (uinteger, range: 0~1)
  t3502Value: 720  # timer value (seconds) at UE side
  t3512Value: 3600 # timer value (seconds) at UE side
  non3gppDeregistrationTimerValue: 3240 # timer value (seconds) at
UE side
  # retransmission timer for paging message
  t3513:
    enable: true     # true or false
    expireTime: 6s   # default is 6 seconds
    maxRetryTimes: 4 # the max number of retransmission
  # retransmission timer for NAS Deregistration Request message
  t3522:
```

```
    enable: true     # true or false
    expireTime: 6s   # default is 6 seconds
    maxRetryTimes: 4 # the max number of retransmission
  # retransmission timer for NAS Registration Accept message
  t3550:
    enable: true     # true or false
    expireTime: 6s   # default is 6 seconds
    maxRetryTimes: 4 # the max number of retransmission
  # retransmission timer for NAS Authentication Request/Security
Mode Command message
  t3560:
    enable: true     # true or false
    expireTime: 6s   # default is 6 seconds
    maxRetryTimes: 4 # the max number of retransmission
  # retransmission timer for NAS Notification message
  t3565:
    enable: true     # true or false
    expireTime: 6s   # default is 6 seconds
    maxRetryTimes: 4 # the max number of retransmission
  # retransmission timer for NAS Identity Request message
  t3570:
    enable: true     # true or false
    expireTime: 6s   # default is 6 seconds
    maxRetryTimes: 4 # the max number of retransmission

# the kind of log output
# debugLevel: how detailed to output, value: trace, debug, info,
warn, error, fatal, panic
# ReportCaller: enable the caller report or not, value: true or
false
logger:
  AMF:
    debugLevel: info
    ReportCaller: false
  NAS:
    debugLevel: info
    ReportCaller: false
  FSM:
    debugLevel: info
    ReportCaller: false
  NGAP:
    debugLevel: info
    ReportCaller: false
  Aper:
    debugLevel: info
    ReportCaller: false
```

Fichero de configuración del SMF:

```
info:
  version: 1.0.2
  description: SMF initial local configuration

configuration:
  smfName: SMF # the name of this SMF
  sbi: # Service-based interface information
    scheme: http # the protocol for sbi (http or https)
    registerIPv4: 127.0.0.2 # IP used to register to NRF
    bindingIPv4: 127.0.0.2  # IP used to bind the service
    port: 8000 # Port used to bind the service
    tls: # the local path of TLS key
      key: config/TLS/smf.key # SMF TLS Certificate
      pem: config/TLS/smf.pem # SMF TLS Private key
  serviceNameList: # the SBI services provided by this SMF, refer
to TS 29.502
    - nsmf-pdusession # Nsmf_PDUSession service
    - nsmf-event-exposure # Nsmf_EventExposure service
    - nsmf-oam # OAM service
  snssaiInfos: # the S-NSSAI (Single Network Slice Selection
Assistance Information) list supported by this AMF
    - sNssai: # S-NSSAI (Single Network Slice Selection Assistance
Information)
        sst: 1 # Slice/Service Type (uinteger, range: 0~255)
        sd: 010203 # Slice Differentiator (3 bytes hex string,
range: 000000~FFFFFF)
      dnnInfos: # DNN information list
        - dnn: internet # Data Network Name
          dns: # the IP address of DNS
            ipv4: 8.8.8.8
    - sNssai: # S-NSSAI (Single Network Slice Selection Assistance
Information)
        sst: 1 # Slice/Service Type (uinteger, range: 0~255)
        sd: 112233 # Slice Differentiator (3 bytes hex string,
range: 000000~FFFFFF)
      dnnInfos: # DNN information list
```

```
        - dnn: internet # Data Network Name
          dns: # the IP address of DNS
            ipv4: 8.8.8.8
  plmnList: # the list of PLMN IDs that this SMF belongs to
(optional, remove this key when unnecessary)
    - mcc: "208" # Mobile Country Code (3 digits string, digit:
0~9)
      mnc: "93" # Mobile Network Code (2 or 3 digits string, digit:
0~9)
  locality: area1 # Name of the location where a set of AMF, SMF
and UPFs are located
  pfcp: # the IP address of N4 interface on this SMF (PFCP)
    addr: 127.0.0.1
  userplaneInformation: # list of userplane information
    upNodes: # information of userplane node (AN or UPF)
      gNB1: # the name of the node
        type: AN # the type of the node (AN or UPF)
      UPF:  # the name of the node
        type: UPF # the type of the node (AN or UPF)
        nodeID: 127.0.0.8 # the IP/FQDN of N4 interface on this UPF
(PFCP)
        sNssaiUpfInfos: # S-NSSAI information list for this UPF
          - sNssai: # S-NSSAI (Single Network Slice Selection
Assistance Information)
              sst: 1 # Slice/Service Type (uinteger, range: 0~255)
              sd: 010203 # Slice Differentiator (3 bytes hex
string, range: 000000~FFFFFF)
            dnnUpfInfoList: # DNN information list for this S-NSSAI
              - dnn: internet
                pools:
                  - cidr: 10.60.0.0/16
          - sNssai: # S-NSSAI (Single Network Slice Selection
Assistance Information)
              sst: 1 # Slice/Service Type (uinteger, range: 0~255)
              sd: 112233 # Slice Differentiator (3 bytes hex
string, range: 000000~FFFFFF)
            dnnUpfInfoList: # DNN information list for this S-NSSAI
```

```
            - dnn: internet
              pools:
                - cidr: 10.61.0.0/16
        interfaces: # Interface list for this UPF
          - interfaceType: N3 # the type of the interface (N3 or
N9)
            endpoints: # the IP address of this N3/N9 interface on
this UPF
              - 10.8.0.1 #127.0.0.8
            networkInstance: internet # Data Network Name (DNN)
    links: # the topology graph of userplane, A and B represent the
two nodes of each link
      - A: gNB1
        B: UPF
  nrfUri: http://127.0.0.10:8000 # a valid URI of NRF

# the kind of log output
# debugLevel: how detailed to output, value: trace, debug, info,
warn, error, fatal, panic
# ReportCaller: enable the caller report or not, value: true or
false
logger:
  SMF:
    debugLevel: info
    ReportCaller: false
  NAS:
    debugLevel: info
    ReportCaller: false
  NGAP:
    debugLevel: info
    ReportCaller: false
  Aper:
    debugLevel: info
    ReportCaller: false
  PFCP:
    debugLevel: info
    ReportCaller: false
```

Fichero de configuración del UPF:

```yaml
version: 1.0.3
description: UPF initial local configuration

# The listen IP and nodeID of the N4 interface on this UPF (Can't
set to 0.0.0.0)
pfcp:
  addr: 127.0.0.8   # IP addr for listening
  nodeID: 127.0.0.8 # External IP or FQDN can be reached
  retransTimeout: 1s # retransmission timeout
  maxRetrans: 3 # the max number of retransmission

gtpu:
  forwarder: gtp5g
  # The IP list of the N3/N9 interfaces on this UPF
  # If there are multiple connection, set addr to 0.0.0.0 or list
all the addresses
  ifList:
    - addr: 10.8.0.1 #127.0.0.8
      type: N3
      # name: upf.5gc.nctu.me
      # ifname: gtpif

# The DNN list supported by UPF
dnnList:
  - dnn: internet # Data Network Name
    cidr: 10.60.0.0/24 # Classless Inter-Domain Routing for
assigned IPv4 pool of UE
    # natifname: eth0

logger: # log output setting
  enable: true # true or false
  level: info # how detailed to output, value: trace, debug, info,
warn, error, fatal, panic
  reportCaller: false # enable the caller report or not, value:
true or false
```

Anexo C – Instalación Firefox

En éste apartado se proporciona la información para descargarse e instalar Firefox. Para ello procedemos a la descarga de la siguiente manera:

```
sudo apt update
sudo apt install snapd
sudo snap install firefox
```

Si todo ha ido bien, el navegador ya está instalado en la máquina virtual, para poder usarlo primero se debe arrancar la interfaz gráfica de la máquina virtual de la siguiente manera:

```
startx
```

De esta manera podremos ver una interfaz como se muestra en la figura X. esto nos permite poder ejecutar el navegador y tener varias terminales abiertas a la vez.

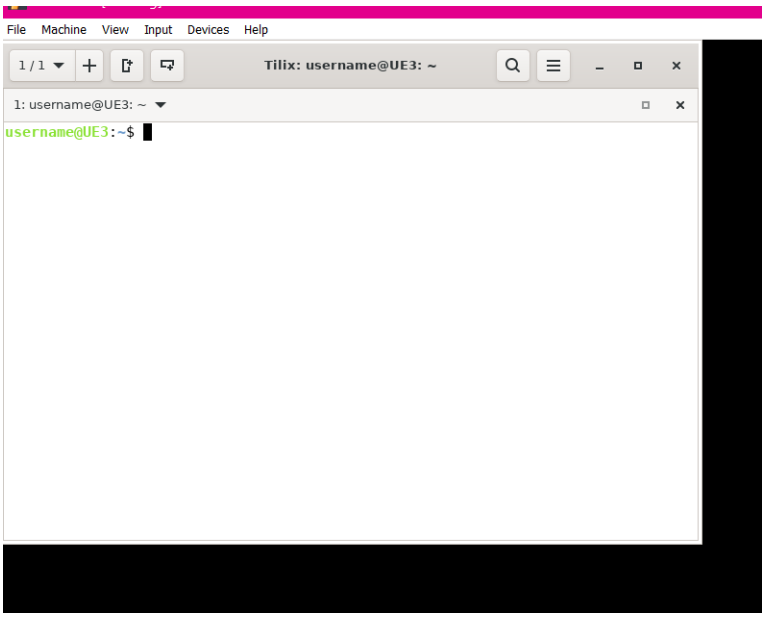

Ilustración 88: Interfaz UE3

Una vez arrancada la interfaz, se podrá ejecutar desde la misma el comando para arrancar el navegador:

```
sudo firefox
```