UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Recuperación ante desastres del directorio activo

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Taberner Cervera, Andrés

Tutor/a: Pons Terol, Julio

Cotutor/a externo: NAVARRO BO, VICTOR

CURSO ACADÉMICO: 2021/2022

Resumen

En los últimos años, varias empresas españolas y del mundo en general han sufrido ataques de *ransomware* que han paralizado su actividad parcial o completamente. El concepto de "hacker" ha evolucionado en los últimos años; los ataques ya no son realizados por personas únicas cuyo propósito es demostrar su potencial técnico, hacer daño u obtener una recompensa económica, sino que grandes empresas de hackers invierten muchísimos recursos para preparar ataques focalizados, cuyo propósito es conseguir paralizar la empresa víctima y pedir un rescate millonario para liberarla.

De tal modo que para solucionar este imprevisto necesitamos llevar a cabo un protocolo llamado comúnmente como “Disaster and Recovery”, un manual que en un 33% de los casos, según un estudio, está creado, pero nunca se ha testeado y que para el 37% de las empresas tienen un impacto catastrófico.

Por ello, nuestra intención ha sido crear un procedimiento para llevar a cabo este manual de manera automatizada, incluso llegar a desplegar la infraestructura en nuevos entornos, como es la Nube, este nuevo servicio que está en auge y que permite infinidad de herramientas.

Los resultados han sido concluyentes en entornos de laboratorio, los cuales, disponemos del entorno On-premise y el entorno de nube, en ambos ha sido testeado el procedimiento.

Palabras clave: Directorio Activo; recuperación ante desastres; controladores de dominio; automatización; *ransomware*.

Abstract

In recent years, several Spanish companies and the world in general have suffered ransomware attacks that have partially or completely paralyzed their activity. The concept of "hacker" has evolved in recent years; Attacks are no longer carried out by single people whose purpose is to demonstrate their technical potential, do damage or obtain financial reward, but large hacker companies invest a lot of resources to prepare targeted attacks, whose purpose is to paralyze the victim company and request a millionaire ransom to free her.

In such a way that to solve this unforeseen event we need to carry out a protocol commonly called “Disaster and Recovery”, a manual that in 33% of cases, according to a study, is created, but has never been tested and that for the 37% of companies have a catastrophic impact.

For this reason, our intention has been to create a procedure to carry out this manual in an automated manner, including deploying the infrastructure in new environments, such as the Cloud, this new service that is on the rise and that allows an infinite number of tools.

The results have been conclusive in laboratory environments, which, we have the On-premise environment and the cloud environment, in both the procedure has been tested.

Keywords: Active Directory; Disaster and Recovery; domain controllers; automation; ransomware.

Resum

En els últims anys, diverses empreses espanyoles i del món en general han patit atacs de *ransomware* que han paralitzat la seua activitat parcial o completament. El concepte de "hacker" ha evolucionat en els últims anys; els atacs ja no són realitzats per persones úniques el propòsit de les quals és demostrar el seu potencial tècnic, fer mal o obtindre una recompensa econòmica, sinó que grans empreses de hackers inverteixen moltíssims recursos per a preparar atacs focalitzats, el propòsit dels quals és aconseguir paralitzar l'empresa víctima i demanar un rescat milionari per a alliberar-la.

De tal manera que per a solucionar aquest imprevist necessitem dur a terme un protocol anomenat comunament com “Disaster and Recovery”, un manual que en un 33% dels casos, segons un estudi, està creat, però mai s'ha testat i que per al 37% de les empreses tenen un impacte catastròfic.

Per això, la nostra intenció ha sigut crear un procediment per a dur a terme aquest manual de manera automatitzada, fins i tot arribar a desplegar la infraestructura en nous entorns, com és el Núvol, aquest nou servei que està en auge i que permet infinitat d'eines.

Els resultats han sigut concloents en entorns de laboratori, els quals, disposem de l'entorn On-premise i l'entorn de núvol, en tots dos ha sigut testat el procediment.

Paraules clau: Directori Actiu; recuperació davant desastres; controladors de domini; automatització; *ransomware*.

Tabla de contenidos

1. Introducción

El Directorio Activo es un servicio creado por Microsoft, establecido en uno o más servidores de una organización, donde se crean objetos tales como usuarios, equipos, grupos, con el objetivo de administrar los inicios de sesión en los equipos conectados en red.

El problema por resolver viene cuando esta herramienta es encriptada mediante un ransomware[1], un ataque intencionado, el cual deniega el servicio a toda la red donde pertenece. Así pues, todos nuestros objetos en red, ordenadores de oficina, servidores, impresoras, incluso soluciones creadas para la producción como puede ser una caja registradora, dejan de ser operativas y por esta razón, nuestra empresa dejaría de producir.

Los ataques están pensados concienzudamente, no son ataques que busquen hacer daño en cualquier punto, sino que los "malos" van infectando poco a poco la empresa (lo que se conoce como movimiento lateral), escalando privilegios desde usuarios sin permisos a usuarios más privilegiados, hasta conseguir las joyas de la corona: usuarios administradores del dominio. El ataque no queda ahí; el último paso es conseguir acceso a las copias de seguridad de la compañía. Cuando todos los puntos están comprometidos, es entonces cuando se lanza el ataque final; se cifra el Directorio Activo de la compañía (pieza central clave de cualquier organización en el funcionamiento de casi todas las demás) y los backups o copias de seguridad, para que su recuperación sea imposible.

1.1. Objetivos

El principal objetivo es conseguir que la empresa se pueda recuperar ante un desastre, ya sea natural o intencionado. Cuando decimos recuperación, nos referimos a intentar restablecer el funcionamiento de la empresa, en nuestro caso recuperar el directorio activo.

Para conseguir el objetivo habrá que solventar la automatización de la recuperación del primer controlador de dominio y el despliegue del resto, además calcular cuántos controladores de domino son necesario para satisfacer los servicios mínimos de la empresa.

[1] Un ransomware o 'secuestro de datos' en español, es un tipo de programa dañino que restringe el acceso a determinadas partes o archivos del sistema operativo infectado y pide un rescate a cambio de quitar esta restricción.

2. Contexto Tecnológico

Los expertos en tecnología están de acuerdo en que este año proseguirá el auge del suministro de *cloud o nube*, de la inteligencia artificial y de la conectividad, de mano, fundamentalmente del despegue definitivo de 5G. (Chen, 2022)[1] También brillarán las tecnologías que permiten extraer más valor del dato, el nuevo petróleo de las organizaciones, las que mejoran la experiencia de los clientes, los empleados y los ciudadanos, las que permiten trabajar desde cualquier lugar facilitando el nuevo puesto de trabajo híbrido que se está consolidando tras la crisis sanitaria, el uso de *blockchain*, de IoT, el incremento del foco en la ciberseguridad y un largo etcétera. (Redacción, 2022)[2]

Situándonos en nuestro ámbito de trabajo, el directorio activo, no hay novedades destacadas, más allá de cambios en la experiencia de usuario, configuraciones y medidas que facilitan el uso. En lo que respecta a seguridad, que cada vez tiene más peso, se están utilizando nuevas tecnologías como ADFS[2], que trabaja con OAuth 2.0 y OpenID Connect. (Dalbera's, 2021)[4]

Esto ayuda a que se mejore la seguridad respecto a hace unos años. Directorio activo anteriormente incluía mecanismos de seguridad para las autenticaciones como Kerberos[3], pero ahora con la aparición de esta nueva tecnología nuestro directorio activo es “simplemente” una base de datos.

La problemática principal de esa base de datos es que debe estar siempre operativa, ya que, si hay alguna inconsistencia, los recursos o aplicaciones no estarán disponibles y esto puede afectar gravemente al funcionamiento de la empresa incluso parar la producción. Esto provocaría pérdida de miles o millones de euros dependiendo del tiempo de la detención. Por eso es muy importante tener muchas alternativas ante cualquier problema, una de ellas es la recuperación ante desastres, un manual que se lleva a cabo para recuperar el dominio o dominios del directorio activo.

Según una encuesta realizada por Semperis, empresa pionera de la resiliencia cibernética impulsada por la identidad para entornos híbridos y de nube cruzada, sobre la recuperación ante desastres, el directorio activo está en el punto de mira de todos los hackers, ya que, es la puerta de entrada al 90% de las aplicaciones de una empresa.

[2] ADFS (Active Directory Federation Services), componente software desarrollado por Microsoft, pueden ejecutarse en sistemas operativos Windows Server para proporcionar a los usuarios acceso de inicio de sesión único a sistemas y aplicaciones ubicados a través de los límites de la organización.

[3] Kerberos, es un protocolo de autenticación de redes de ordenador creado por el MIT que permite a dos ordenadores en una red insegura demostrar su identidad mutuamente de manera segura.

Estamos hablando de que el 47% de las empresas encuestadas utiliza esta herramienta de identidad, entonces podemos decir que es una de las herramientas más críticas que pueda tener una empresa, por lo tanto, si tu directorio activo no es seguro, nada lo es. (Mr. Hawkins, 2019)[5]

En relación con el impacto que puede crear este tipo de situaciones, en el 37% tiene un impacto catastrófico, es decir, no se recuperaron o si se recuperaron tuvieron graves consecuencias, Maersk CISO, Andy Powel, lo compara con navegar con un motor roto.

Las malas noticias son que sólo el 37% de las organizaciones son conscientes de la complejidad de una recuperación de un bosque y lo único que Microsoft aporta es un documento guía como el que vamos a seguir, que mayoritariamente son pasos manuales. Además, en el momento de la recuperación, más del 50% de los encuestados nunca han testeado una recuperación ante ciber-ataques y pocos logran recuperar el directorio de manera rápida.

En lo que a planes de recuperación se refiere, el 33% tiene creado el plan, pero nunca lo han testeado y sólo el 15% ha testeado dentro de los últimos 6 meses.

Las principales preocupaciones de la recuperación ante un ciberataque son, que nunca se ha testeado la recuperación; no hay un plan de recuperación; las copias de seguridad están encriptadas o borradas; no hay una rápida recuperación y la posibilidad de recuperación no está bien definida. (Semperis, 2020)[7]

Una de las conclusiones de este artículo es la incorporación de la recuperación basada en la nube, según un director de IT, es una fortaleza en el caso te tener corrupto nuestro centro de datos. Los datos de esta encuesta corresponden a la ilustración 1.

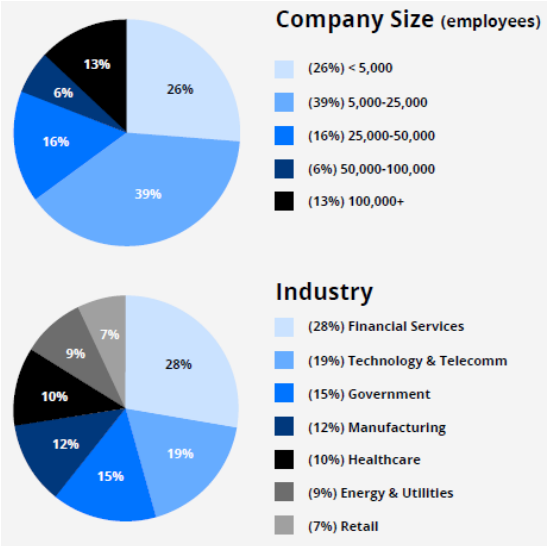

Ilustración 1, empresas colaboradoras en la encuesta

2.1 Prácticas en Mercadona

Mercadona, empresa de capital familiar, es una de las principales compañías de supermercados físicos y online en España que tiene por objetivo asumir la responsabilidad de prescribirle a "El Jefe" (cliente) la mejor opción para satisfacer sus necesidades de alimentación, cuidado del hogar, cuidado personal y cuidado de animales. Gracias a ello, 5,6 millones de hogares depositan diariamente su confianza en la compañía.

Misión, "Llenar la tripa", es decir, prescribir al consumidor final productos / soluciones que cubran sus necesidades de comer, beber, cuidado personal, cuidado del hogar y cuidado de animales...

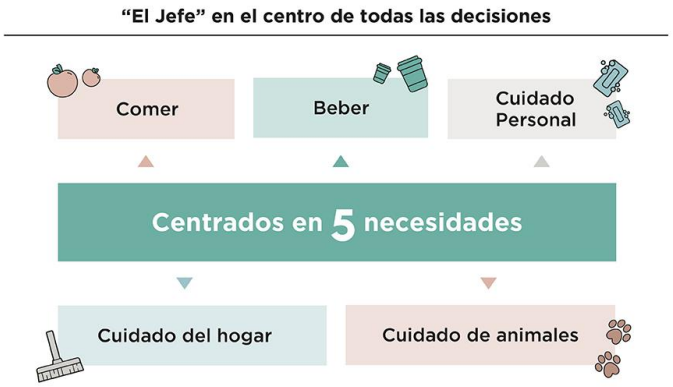

Ilustración 2, 5 necesidades como objetivo de misión

Visión, conseguir una Cadena Agroalimentaria Sostenible, que la gente quiera que exista y sienta orgullo de ella, liderada por Mercadona y teniendo a "El Jefe" como faro.

Historia, supermercado nacido en Tabernes Blanques por el Grupo Cárnicas Roig en 1977 del matrimonio formado por D. Francisco Roig Ballester y Da Trinidad Alfonso Mocholí, padres de Juan Roig y sus 4 hermanos. Como acciones destacadas, fue la primera empresa en España en utilizar el escáner para la lectura del código de barras en los puntos de venta en 1982 o la puesta en marcha de la primera fase del bloque logístico Almacén Siglo XXI de Ciempozuelos en 2007, almacén automatizado según el estudio del Reputation Institute de Nueva York.

Actualmente dispone de 1.632 tiendas en toda España y 29 en Portugal, y una plantilla de 95.800 personas orientadas a la excelencia, 2.500 de ellas en Portugal.

Ilustración 3, Almacén Ciempozuelos (Madrid), construido en 2007

Modelo, desde 1993 Mercadona basa todas sus decisiones en su Modelo de Calidad Total que busca satisfacer por igual y con la misma intensidad a los cinco componentes de la empresa: "El Jefe", como internamente denomina al cliente, El Trabajador, El Proveedor, La Sociedad y El Capital. (Mercadona)

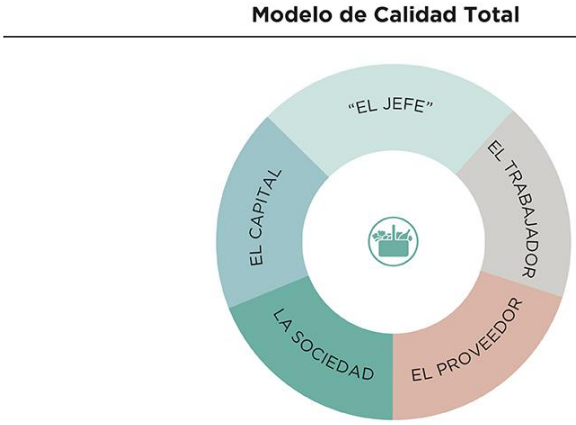

Ilustración 4, Modelo en el que se basan todas las decisiones

La vacante a la que opté se trata de técnico de Sistemas Microsoft y directorio activo. Durante las practicas, los propios técnicos de la división han estado formándome en las diferentes tecnologías que utilizan día a día, como por ejemplo Directorio Activo, ADFS, Windows Server, etc. Mientras tanto mi tutor va encargándome trabajos para practicar las tecnologías. (Mercadona, 2020)[3]

2.2 Propuesta

Nuestra propuesta se basa en uno de los protocolos de seguridad más importantes que una empresa media o de gran tamaño debería tener. La posibilidad de que en la historia de una empresa ocurra este tipo de desastres, es bajo, pero no por eso es necesario no tenerlo, incluso realizarlo como simulacro una vez al año. Hablamos de la recuperación de un directorio activo

con todos sus componentes necesarios para poder recuperar la actividad. Este protocolo ya está vigente en la empresa, pero no de una manera optimizada y ágil, para ello, utilizaremos scripts en algunos casos.

Requisitos para la propuesta:

- Recuperar el primer controlador de dominio del directorio activo.
- Conocer cuántos controladores de dominio son necesarios para cubrir el caudal mínimo necesario para que la empresa recupere su actividad.
- Desplegar una infraestructura en cualquier plataforma nube o interna.
- Promocionar a controlador de dominio los servidores necesarios para la infraestructura.
- Configuraciones adicionales necesarias para poder recuperar el dominio de forma segura, como cambiar la contraseña a todos los usuarios si se trata de un ataque.

2.3 Conceptos

Para poder seguir una lectura fluida, vamos a explicar unos conceptos relacionados con el directorio activo.

- Controlador de Dominio (DC), los controladores de dominio son servidores de Windows que contienen la base de datos de Active Directory y ejecutan funciones relacionadas con AD, como la autenticación y la autorización.
- Roles FSMO, son una serie de funciones de Active Directory para prevenir conflictos de replicación.
- Directorio Activo (AD), es una base de datos y un conjunto de servicios que conectan a los usuarios con los recursos de red que necesitan para realizar su trabajo. La base de datos (o el directorio) contiene información crítica sobre su entorno, incluidos los usuarios y las computadoras que hay y quién puede hacer qué.
- Global Catalog (GC), permite a los usuarios y aplicaciones encontrar objetos en un árbol de dominio de Active Directory, dado uno o más atributos del objeto de destino.

3. Análisis del problema

Directorio Activo es un servicio de identidad hecho por Microsoft que sirve para construir una organización lógica de una empresa y poder administrarla, cuya característica principal es autenticar.

Para poner en contexto, hay que aclarar que para trabajar con normalidad nosotros necesitamos acceder a un ordenador, el cual se autentica contra nuestro directorio activo. También las aplicaciones, internas o externas, que usamos para trabajar, el correo, la intranet, aplicación de reserva de salas, etc. La gran mayoría de ellas requieren de directorio activo para ser utilizadas.

La principal vulnerabilidad del directorio activo es la base de datos, ya que, sin ella el servicio dejaría de funcionar. Para poder acceder a esta base de datos hay que tener unos privilegios que de normal no son de fácil acceso. Si se trata de un ataque intencionado o *ransomware*, lo que intentarán conseguir los "hackers" es tener un usuario con acceso al directorio activo. Con el acceso a nuestro directorio seguirán buscando información para vulnerar y ejecutar comandos y poder hacer movimiento lateral [4], escalar verticalmente obteniendo mayores privilegios y poder vulnerar nuestra base de datos.

Si se trata de una catástrofe natural, como podría ser un incendio del centro de procesamiento de datos (CPD), deberemos tener en cuenta si nuestros datos continúan a salvo, es decir que tengamos una copia de seguridad para poder restaurar. En la mayoría de los casos, las empresas que cuentan con un CPD propio con certificación tienen otro centro de procesamiento con replicación y consiguen redundancia de datos. Es decir, no necesitaran en la mayoría de los casos una copia para poder restaurar. En cualquier caso, no es tarea fácil, porque puede haber inconsistencia de datos y es casi imposible poder recuperar el cien por cien del estado.

[4] El movimiento lateral, es parte de una de las etapas que utiliza un hacker ético y cracker en un ataque dirigido, que típicamente son; la recolección de información y escaneo, acceso y escalamiento de privilegios, exfiltración, sostenimiento, asalto y ofuscación

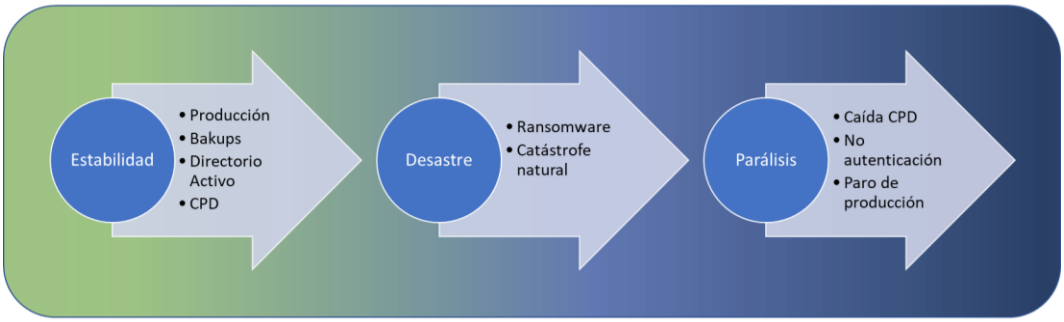

Ilustración 5, problemática

3.1 Análisis de soluciones posibles

Como hemos visto antes hay dos posibles inconvenientes donde, primero; nuestros datos están corruptos por un ataque intencionado, o que nuestro centro de procesamiento haya sufrido un desastre no intencionado y nuestros datos no hayan sido expuestos.

Para el caso en el que nuestros datos hayan sido expuestos, necesitaremos aislar y mantener fuera de internet lo máximo posible. Una vez aislado, necesitaremos desplegar la infraestructura en un lugar donde no se haya expuesto la vulnerabilidad, como podría ser otro centro de procesamiento o la nube.

Por otra parte, si tenemos la posibilidad de solucionar el problema del desastre no intencionado, que la inundación no haya causado graves desperfectos, se consiga volver al funcionamiento normal. En caso de haber perdido nuestros servidores, deberíamos poder recuperar nuestra infraestructura gracias a nuestro CPD pasivo o en caso de no tener disponible otro centro de procesamiento, recuperar mediante copias de seguridad.

3.2 Solución propuesta

En este proyecto vamos a enfocarnos en la posibilidad de que nos hayan atacado mediante un *ransomware*. Para ello, he desarrollado un modelado, ilustración 6, con todas las fases que vamos a aplicar

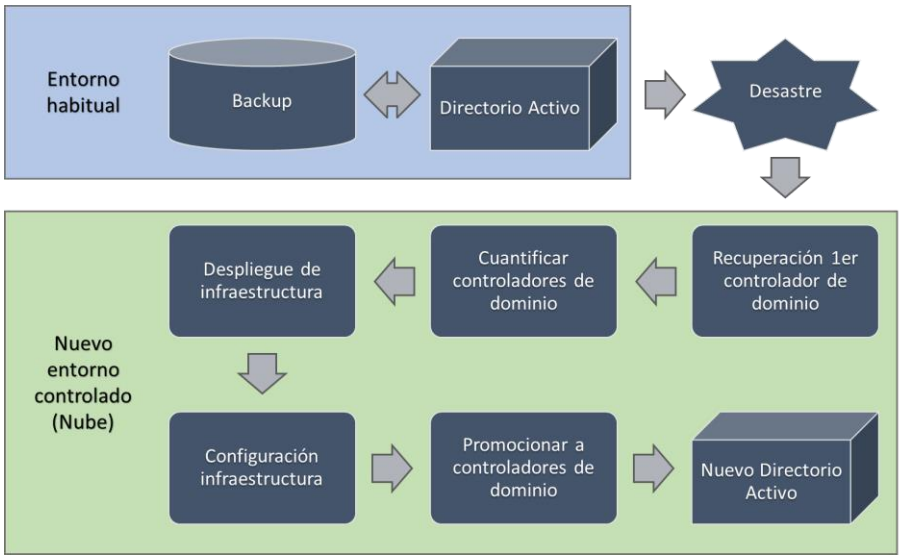

Ilustración 6, solución

Para planificar el proyecto, lo vamos a dividir en tres fases, definición, diseño e implementación.

- Definición: Determinar el alcance del proyecto, conocer el estado actual del cambio y buscar los recursos necesarios.
- Diseño: Visualizar el diseño técnico detallado y empezar a desarrollarlo en un laboratorio.
- Implementación: Construir pruebas de concepto y preparar la documentación para la entrega.

Con una estimación horaria de 15 semanas y 4 horas al día, que serían unas 300 horas, compaginando con las prácticas.

Para la realización del proyecto se van a necesitar diferentes recursos. Licencias Windows Server 2019, sistema de virtualización para el laboratorio y hardware necesario donde virtualizar en caso de desplegar en servidores propios. Si desplegamos el laboratorio, con similares características en la nube el presupuesto sería el de la ilustración 7.

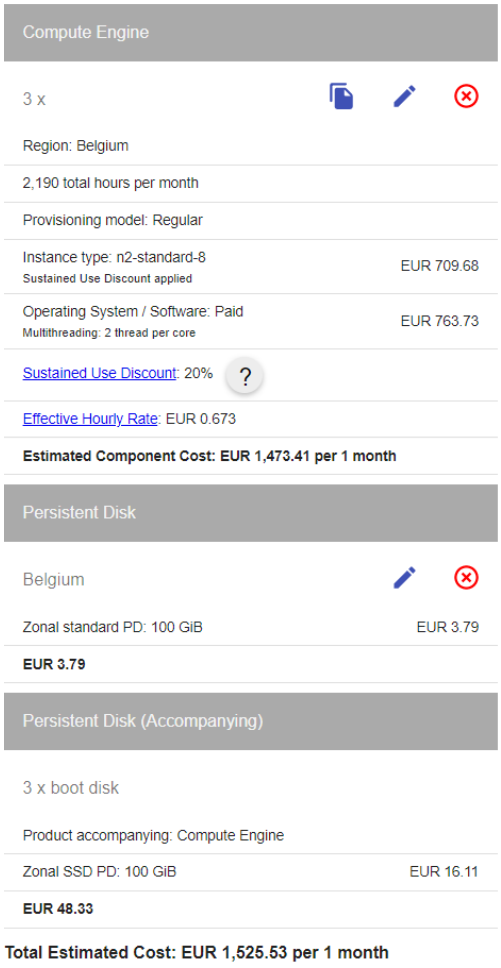

Ilustración 7, estimación laboratorio en la nube

En mi caso, gracias a la empresa donde realizo las practicas, nos han proporcionado un laboratorio para desarrollar el proyecto con todos los recursos, tanto *On-premise*[5] como en la nube.

5 On-premise se refiere a que la instalación del programa se ha realizado de manera local, en las instalaciones de la empresa y obligando a esta a crear una infraestructura informática compleja con servidores que requieren mantenimiento.

4. Diseño de la solución

Para tener una visión funcional de como trabajaremos, hemos formado una estructura en los diferentes ambientes donde desarrollaremos la solución. Habrá diferentes diseños acordes a los pasos anteriormente anunciados de la elaboración del proyecto.

4.1 Diseño funcional *On-premise*

Para el desarrollo del primer paso, automatización de la recuperación del dominio, he necesitado el entorno de laboratorio. El programa para la realización de los scripts es PowerShell ISE (Microsoft, s. f.)[6], el cual se puede ejecutar tanto en el entorno local como directamente en las máquinas virtuales. La conexión con nuestro CPD es interna, es decir, no necesitamos una VPN u otra manera de enlace. La virtualización es proporcionada por un equipo interno. En cuanto a las máquinas, podemos moldear sus características a nuestro gusto acorde a la aplicación que vamos a desplegar, gracias a el sistema operativo administrador del CPD, VMware ESXi y vSphere.

Una vez desplegadas las máquinas virtuales, podremos acceder a ellas mediante escritorio remoto desde nuestro ordenador y poder administrarlas. Cuando terminemos de escribir los scripts llegaremos al momento de la ejecución. El lenguaje PowerShell nos permite la ejecución de forma remota o en local, es decir, podemos tener los scripts en el entorno local y ejecutar el script en cualquier máquina virtual, sabiendo la dirección IP y el usuario administrador.

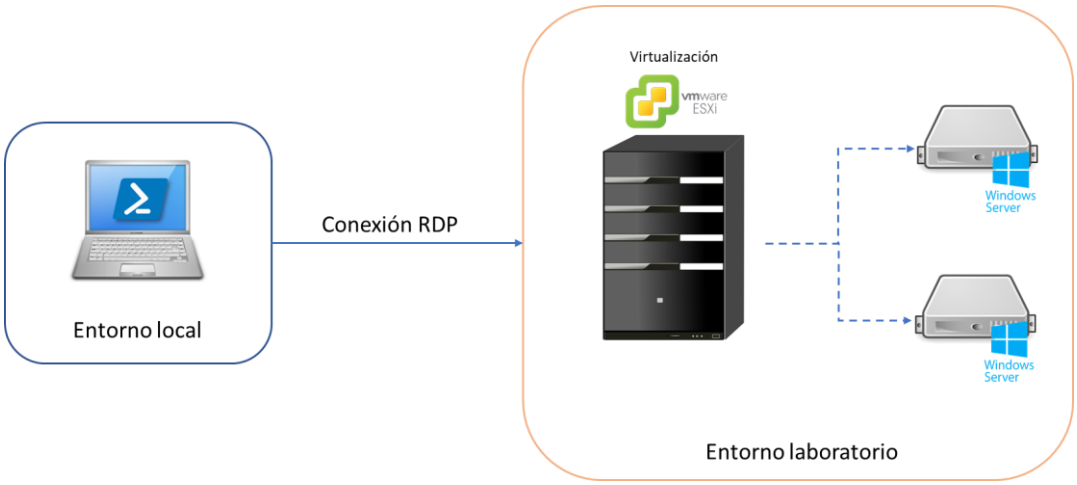

Ilustración 8, diseño funcional On-premise

4.2 Diseño funcional de la Nube

El siguiente objetivo del proyecto es el despliegue de manera ágil en la nube, para ello vamos a utilizar el protocolo SSH para la conexión al bastión, que es nuestro enlace con la nube, en este caso Google Cloud Platform (GCP).

Necesitaremos generar el par de claves SSH, pública y privada, para la autenticación de la conexión SSH al bastión, la generaremos con la aplicación puTTYgen y subiremos a GCP la clave pública.

Una vez configuramos la conexión SSH al bastión, podremos acceder a nuestro directorio, en él es donde deberemos crear un nuevo directorio el cual lo utilizaremos para nuestra automatización del despliegue sobre Terraform (HashiCorp. s. f.). Con la ayuda de GitLab (GitLab, 2022)[10] crearemos nuestro proyecto Terraform, para el forjado y versionado del proyecto, y además tener una copia de seguridad de los archivos.

Cuando hayamos desplegado y configurado las máquinas virtuales en la nube, nos conectaremos mediante RDP a las maquinas como si se tratara de nuestro CPD, y es ahora cuando deberemos recuperar nuestro directorio activo con la copia de seguridad del controlador de dominio objetivo.

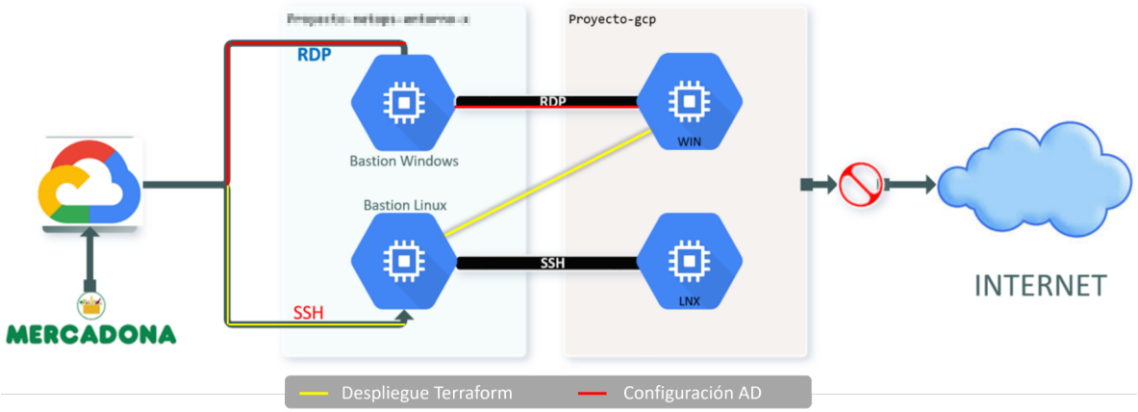

Ilustración 9, Diseño funcional en la nube

Ahora, desplegado y configurado el primer controlador de dominio, continuaremos con el siguiente paso, que es la promoción del resto de controladores de dominio, los que sean necesarios y previo despliegue de estos. Esta acción también se realizará mediante scripts.

Una vez consigamos tener una infraestructura estable, procederemos a terminar la configuración del dominio y la conexión con la empresa.

4.3 Tecnología utilizada

En cuanto a las tecnologías empleadas:

- Protocolos utilizados son RDP y SSH para las conexiones, RDP para los *host* Windows y SSH para la conexión al bastión para el despliegue.
- Lenguajes como Powershell, Terraform, para scripting y configuración. PowerShell para automatizar la recuperación del primer controlador de dominio y del resto, Terraform para el despliegue de infraestructura como código.
- Sistemas, VMware, Windows, Linux, en virtualización y desarrollo. VMware utilizado para administrar las máquinas virtuales del laboratorio, Windows es el sistema operativo principal del proyecto, ya que, sobre el corre nuestra herramienta, y Linux lo hemos utilizado para la nube.
- Plataformas de *cloud*, Google Cloud Platform y Microsoft Azure, para el despliegue en infraestructura ajena. Las plataformas seleccionadas para desplegar nuestra infraestructura.

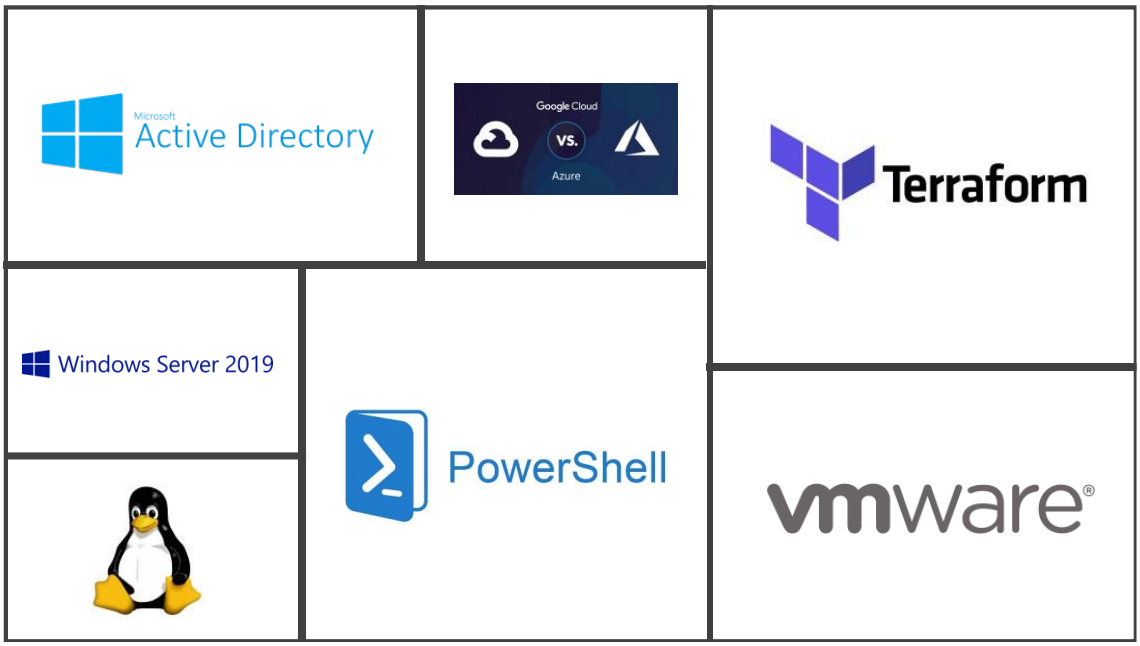

Ilustración 10, tecnologías utilizadas en el proyecto

5. Desarrollo de la solución propuesta

5.1 Recuperación automatizada

Como mencionamos anteriormente, este paso está basado en el manual "Disaster and Recovery" del directorio activo ya implementado por la empresa. Teniendo en cuenta esto utilizaremos el mismo índice para la recuperación.

5.1.1 Consideraciones previas

Para llevar a cabo la restauración del controlador de dominio (DC) seleccionado en el nuevo servidor candidato, habrá que seleccionar una buena copia de seguridad, porque en ello dependerá de cómo haya que proceder en el restablecimiento, seguiremos estos pasos:

- El controlador de dominio debe estar accesible para poder aislarlo y desconectarlo de la red, en nuestro caso estará en el entorno de virtualización y no llevará muchos problemas.
- El DC debe ser servidor DNS y poseer las particiones con todos los dispositivos del dominio, "ForestDNSZones" y "DomainDNSZones" para evitar la pérdida de datos.
- La copia de seguridad no deberá tener más de 180 días, por política de seguridad de Microsoft, más allá no se podrán restablecer los objetos, se llama "tombstone lifetime attribute".
- El equipo candidato debe ser uno de los controladores de dominio anteriores (formateado y limpio) y si no es así, debe tener idéntico hardware.
- A ser posible, que el DC de la copia de seguridad, tenga los roles FSMO.
- Hay que aislar por completo el servidor, o apagar todos los demás controladores del bosque infectados.

En nuestro caso, recuperar la copia de seguridad en la nube ha sido un problema, por ello, hemos decidido tener dispuesto un controlador de dominio, de nuestro directorio activo, en la nube, ya que, la copia de seguridad de nuestro CPD no es posible migrarla a la nube sin que Microsoft lo soporte. Más adelante veremos posibles alternativas.

5.1.2 Restauración de sistema

Para hacer la restauración de sistema, necesitamos una imagen ISO desde donde cargar la reparación del equipo y una copia de seguridad ubicada en un disco aparte al del sistema operativo, en nuestra máquina virtual. En principio, este será el único paso que realizaremos manualmente.

- Una vez iniciado, se nos presentará el diálogo que se presenta en la Ilustración 11.

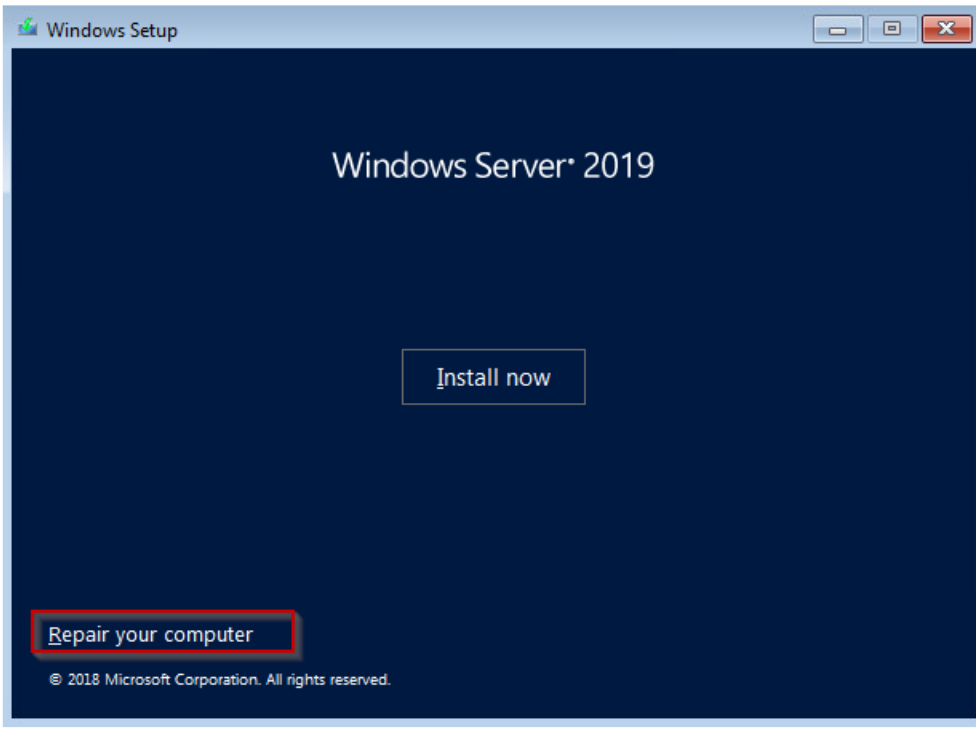

Ilustración 11, inicio restauración

- Seleccionamos la opción: “Troubleshoot” y luego: “System Image Recovery”. (ver ilustración 12)

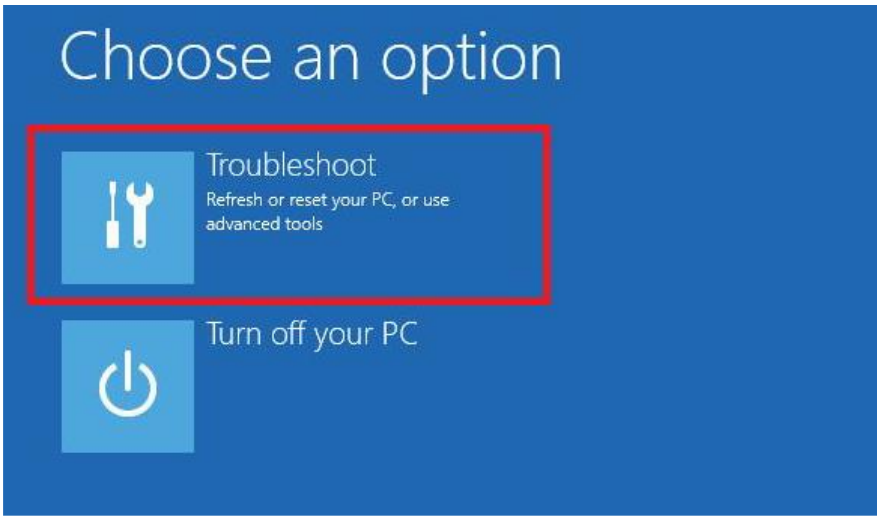

Ilustración 12, paso 2 de restauración

- Seleccionamos la opción: "Select a system image". (ver ilustración 13)

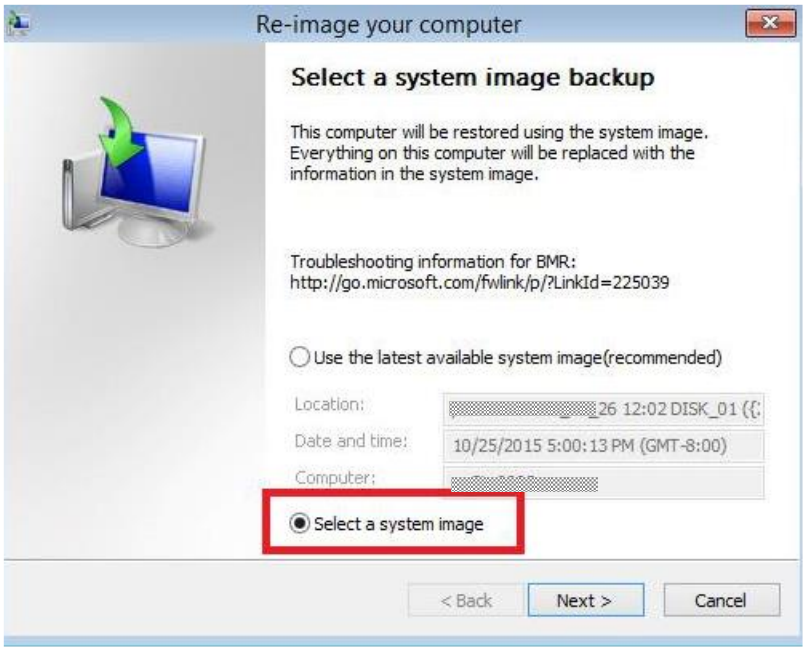

Ilustración 13, paso 3 de restauración

- Seleccionamos el "backup" que queremos restaurar y pulsamos: "Next". (ver ilustración 14)

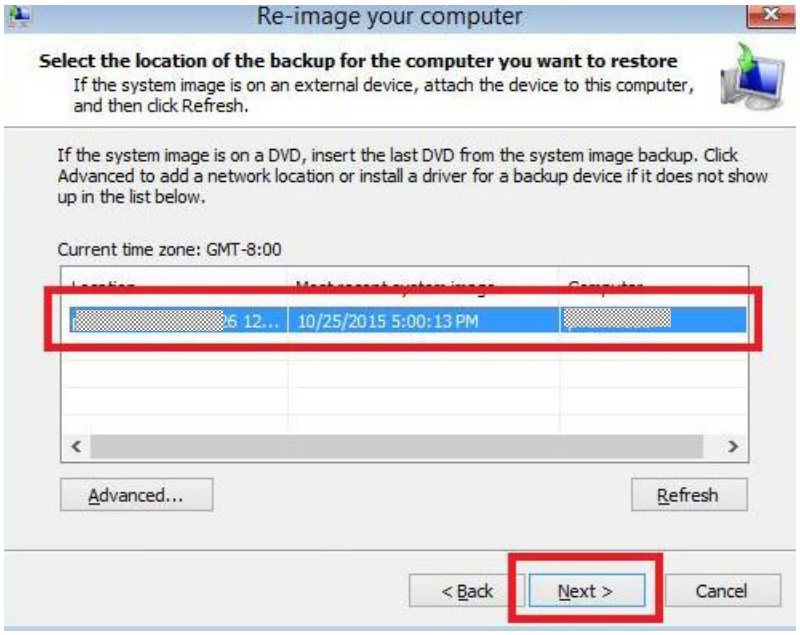

Ilustración 14, paso 4 de restauración

- Seleccionamos la fecha y hora del "backup" que queremos restaurar y pulsamos: "Next".
- Pulsamos: "Advanced", desmarcamos la casilla "Automatically restart this computer after the restore is complete" y pulsamos "Next". (ver ilustración 15 y 16)

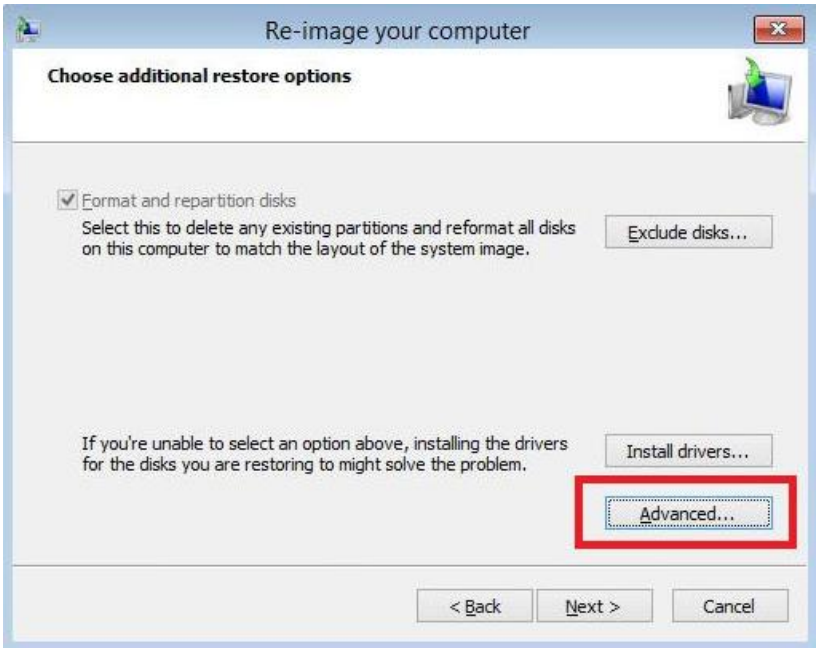

Ilustración 15, paso 5 de restauración

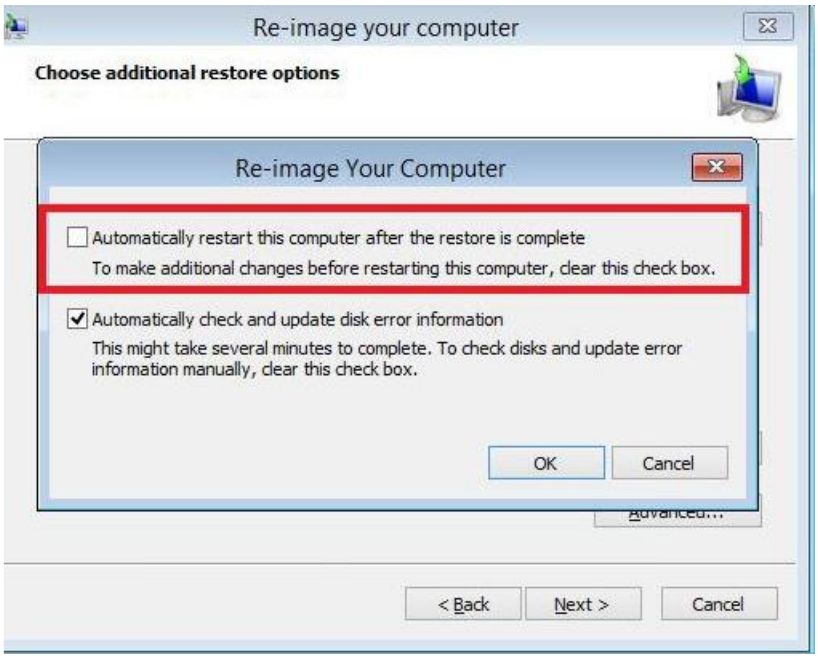

Ilustración 16, paso 6 de restauración

- Comprobamos el resumen de las opciones seleccionadas para realizar la restauración del sistema, y pulsamos “Finish” para iniciar el proceso.

5.1.3 Paso a paso

En este apartado iremos revisando cada proceso del índice y transformando a lenguaje PowerShell, además de los procedimientos necesarios para restablecer el primer controlador de dominio.

5.1.3.1 Revisar configuración de red:

Dado que ahora solo hay operativo un único DC en el dominio, el DC seleccionado para iniciar la restauración, se debería de revisar que la configuración IP y DNS está definida correctamente. La configuración del servidor DNS primario debería apuntar al propio equipo. (ver ilustración 17)

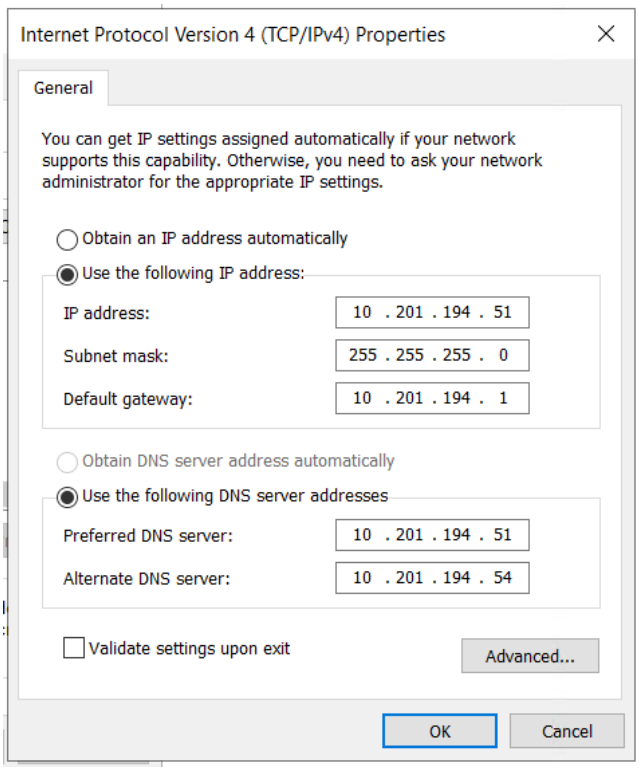

Ilustración 17, preferencias de IPv4

La transformación a lenguaje PowerShell para automatizar este paso es el siguiente:

```
$ip = (Get-NetIPAddress | Where-Object {$_.InterfaceAlias -match
"Ethernet"}).IPv4Address
$ip = $ip -join ""

$dns = (Get-DnsClientServerAddress).ServerAddresses[0]

if ($dns -ne $ip) {
   Set-DnsClientServerAddress -InterfaceAlias ethernet0 -ServerAddresses
$ip
}
```

Código 1, revisar configuración de red

5.1.3.2 Identificar la red:

Comprobamos que la red está bien identificada, en nuestro dominio, si no es así, reiniciamos el servicio “Network Location Awareness”, este servicio permite que las aplicaciones de Windows identifiquen la red lógica a la que está conectada una computadora con Windows, es decir, el dominio. (ver ilustración 18 y 19)

Ilustración 18, red no identificada

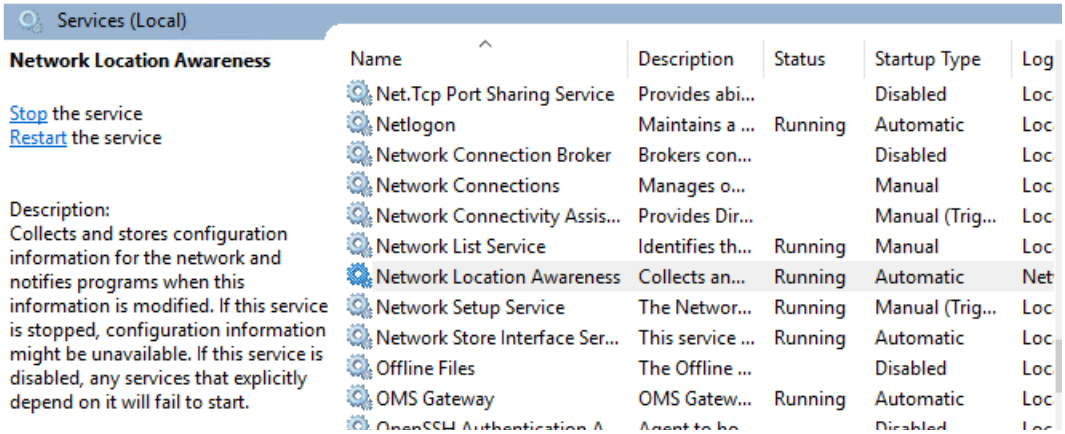

Ilustración 19, administrador de servicios

La transformación a lenguaje PowerShell para automatizar este paso es el siguiente:

```
$doms = (Get-WmiObject Win32_NetworkAdapterConfiguration).DNSDomain
$doms = $doms -join ""
$dom = Get-WMIObject Win32_ComputerSystem | Select-Object -ExpandProperty
Domain
$eth = Get-NetAdapter

if ($doms -ne $dom) {
    Restart-Service -Name NlaSvc
       Disable-NetAdapter -Name $eth.name -Confirm:$false
       Enable-NetAdapter -Name $eth.name -Confirm:$false
} else {Write-Host "Done" -ForegroundColor Green}
```

Código 2, identificar red

5.1.3.3 Comprobamos registro del sistema "Repl Perform Initial Synchronizations":

En las claves de registro del sistema, ejecutando "regedit", bajo la ruta: [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters] (ver ilustración 20)

Añadir el DWORD: "Repl Perform Initial Synchronizations" con valor: "0".

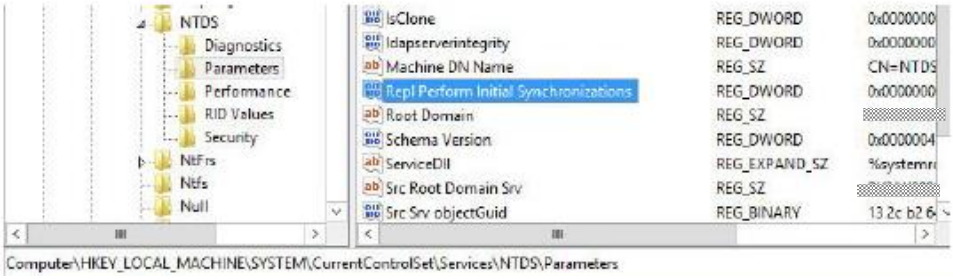

Ilustración 20, parámetros de sistema

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
#Cambiar Repl Perform Initial Synchronizations
##Si es 1 cambiar a 0
if ( $Repl -ne 0)
{
    New-ItemProperty -Path $path1 -Name 'Repl Perform Initial
Synchronizations' -Value 0 -PropertyType DWord
    #Si modifoca o crea reinicia la máquina
    Write-Host "La máquina se va a reiniciar"
       Restart-Computer
} else {Write-Host "Done" -ForegroundColor Green}
```

Código 3, comprobar registros de sistema

Esta clave de registro nos permite indicar que el controlador de dominio no espere a el resto de los controladores de dominio, ya que en este momento sólo estamos recuperando uno, para sincronizar la base de datos SYSVOL. En caso de no existir esta clave de registro DWORD, la crearemos con el valor a 0.

5.1.3.4 Habilitar el usuario administrador y grupos:

Necesitaremos habilitar el usuario administrador, para realizar cambios con mayor privilegio. Además, tendremos que añadir este usuario a los grupos “Schema Admins” y “Enterprise Admins”, en el caso de que aún no pertenezcan.

Clic derecho sobre el usuario, que en nuestro caso es Andrés, y habilitamos el usuario. En la mayoría de los casos el usuario administrador no tiene el nombre de “Administrator”, por seguridad se cambia el nombre y contraseña, además de deshabilitarlo. Para hacer operaciones donde necesitemos mayores privilegios, crearemos un usuario para esta acción.

En caso de no pertenecer a alguno de estos grupos, clic derecho y añadimos al grupo que no pertenezca. (ver ilustración 21)

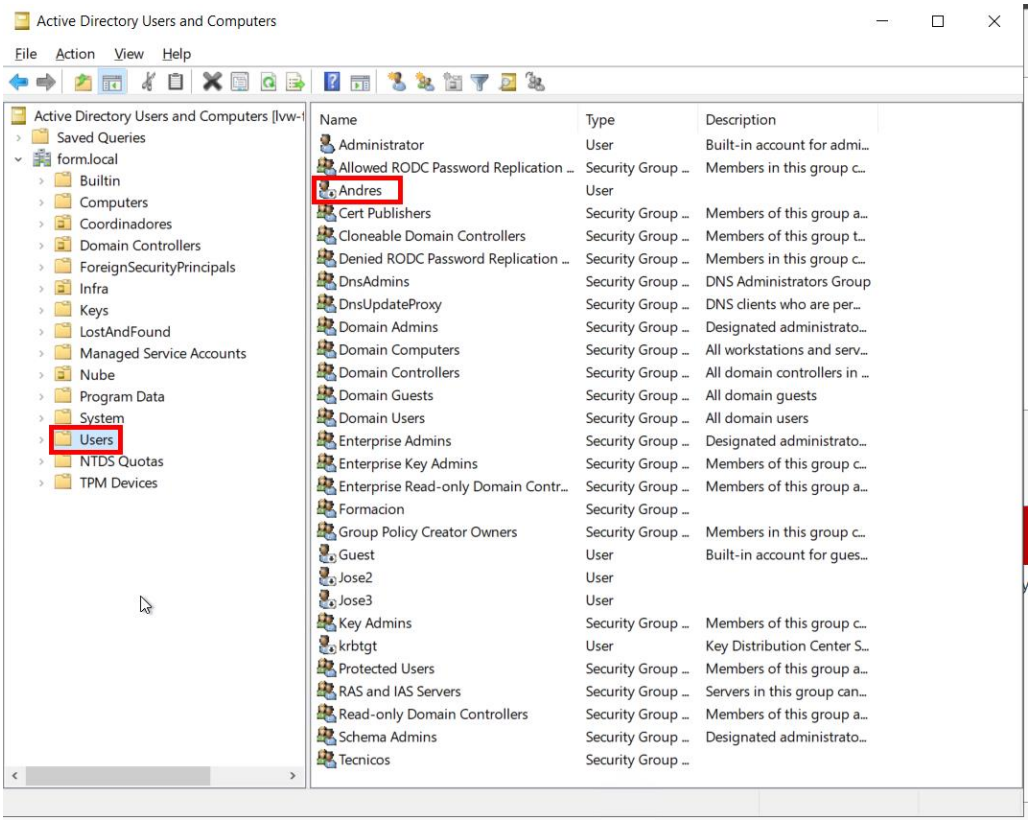

Ilustración 21, administrador de usuarios y computadores de AD

Para comprobar si pertenece a los grupos con el comando “whoami /groups”. (ver ilustración 22)

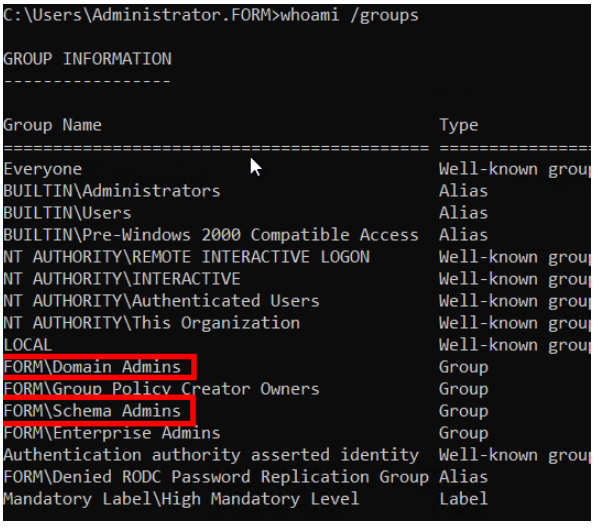

Ilustración 22, grupos a los que pertenece el administrador del dominio

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
#Habilitar Usuario administrator
$BA = (Get-ADDomain).domainsid
$BA = $BA.ToString() + "-500"
$usr = Get-ADUser -Identity $BA
$usr | Enable-ADAccount

#Añadimos usuario a schema admins y Enterprise Admins (sino pertenecemos
aun)
$MemSA = (Get-ADGroupMember -Identity "Schema Admins")
if (!$MemSA | Where {$_.name -eq $usr.Name}) {
    Write-Host not ok
    Add-ADGroupMember -Identity "Schema Admins" -Members $usr.name
}
$MemEA = (Get-ADGroupMember -Identity "Enterprise Admins")
if (!$MemEA | Where {$_.name -eq $usr.Name}) {
    Add-ADGroupMember -Identity "Enterprise Admins" -Members $usr.name
}
```

Código 4, habilitar usuario administrador y a que grupos pertenece

5.1.3.5 Modificar contraseña del administrador:

Por seguridad e integridad de datos, necesitaremos cambiar la contraseña del administrador. También se podrá identificar por tener el RID-500 (ver ilustración 23), este valor es un identificador único del dominio, y este valor siempre pertenece al administrador. El proceso de cambio se debe procesar dos veces seguidas.

```
PS C:\Users\Administrator.FORM\Desktop> get-aduser Andres


DistinguishedName : CN=Andres,CN=Users,DC=form,DC=local
Enabled           : True
GivenName         : Andres
Name              : Andres
ObjectClass       : user
ObjectGUID        : d83d6b97-f83f-4ca3-b41d-084d0de37f7b
SamAccountName    : andres
SID               : S-1-5-21-2072063333-644844303-4078280733-500
Surname           :
UserPrincipalName :andres@form.local
```

Ilustración 23, buscar administrador de dominio

Abrir una ventana de línea de comandos y escribir el siguiente comando “NET USER <cuenta_administrator> * /domain”, esta operación la realizaremos dos veces. (ver ilustración 24)

```
C:\Users\Administrator.FORM>NET USER Andres * /domain
Type a password for the user:
Retype the password to confirm:
The command completed successfully.
```

Ilustración 24, reiniciar contraseña del administrador del dominio

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
#función que genera una contraseña aleatoria
function Get-RandomPassword {
param (
[Parameter(Mandatory)]
[int] $length,
[int] $amountOfNonAlphanumeric = 1
)
Add-Type -AssemblyName 'System.Web'
return [System.Web.Security.Membership]::GeneratePassword($length,
$amountOfNonAlphanumeric)
}

##Cambiar contraseña administrador
$pass = Get-RandomPassword 20
write-host $pass  ####MOSTRAR POR PANTALLA LA CONTRASEÑA GENERADA####
Set-ADAccountPassword -Identity ($usr.Name) -NewPassword (ConvertTo-
SecureString -AsPlainText $pass -Force) -Reset
```

Código 5, modificar contraseña del administrador del domino

5.1.3.6 Comprobar el recurso compartido SYSVOL y NETLOGON:

Este paso es importante ya que es necesario para las réplicas entre controladores de dominio.

Para asegurar que los recursos están compartidos lanzaremos el comando “net share”. (ver ilustración 25)

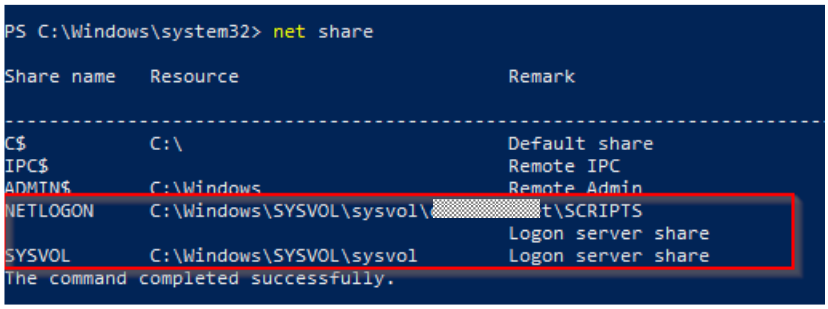

Ilustración 25, comprobar recursos compartidos

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
$net = @((Get-SmbShare -Name "NETLOGON").Description, (Get-SmbShare -Name
"SYSVOL").Description)
$net = $net -match "Logon server share"

if ($net.Length -eq 2) {
    Write-Host "NETLOGON Y SYSVOL correctos" -ForegroundColor Green
} else { write-error "No estan compartidos NETLOGON O SYSVOL" }
```

Código 6, comprobar recursos compartidos, SYSVOL Y NETLOGON

5.1.3.7 Marcar SYSVOL como autoritativo:

Abrimos la consola de usuarios y computadoras en las herramientas de "Server Manager", seleccionamos opciones avanzadas en vista (ver ilustración 26). Desplegamos el controlador de dominio restaurado hasta "SYSVOL Suscription" y propiedades. (ver ilustración 27)

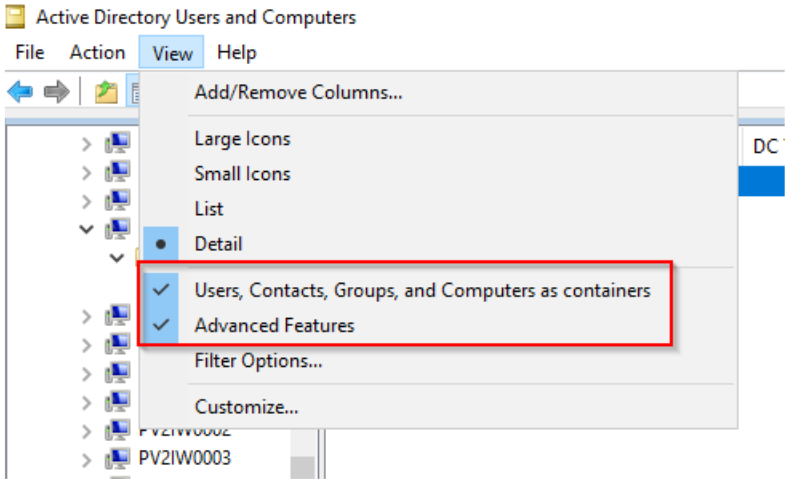

Ilustración 26, opciones avanzadas de vista

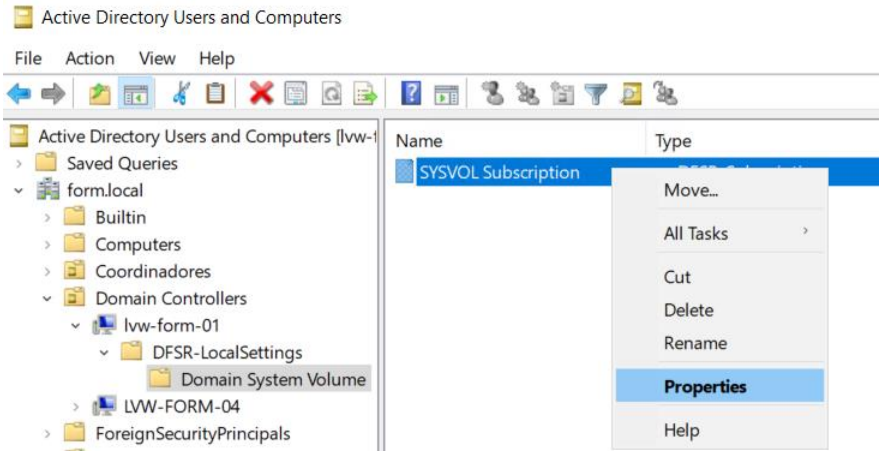

Ilustración 27, recurso SYSVOL Suscription

Verificar/Editar los siguientes atributos de la suscripción SYSVOL:

- Modificar el valor del atributo: "msDFSR-Enabled" = "false".
- Modificar el valor del atributo: "msDFSR-options" = 1.
- Ejecutamos "dfsrdiag pollad".

- Comprobamos que se ha generado un evento “DFSR Replication” con id 4114. (ver ilustración 28)

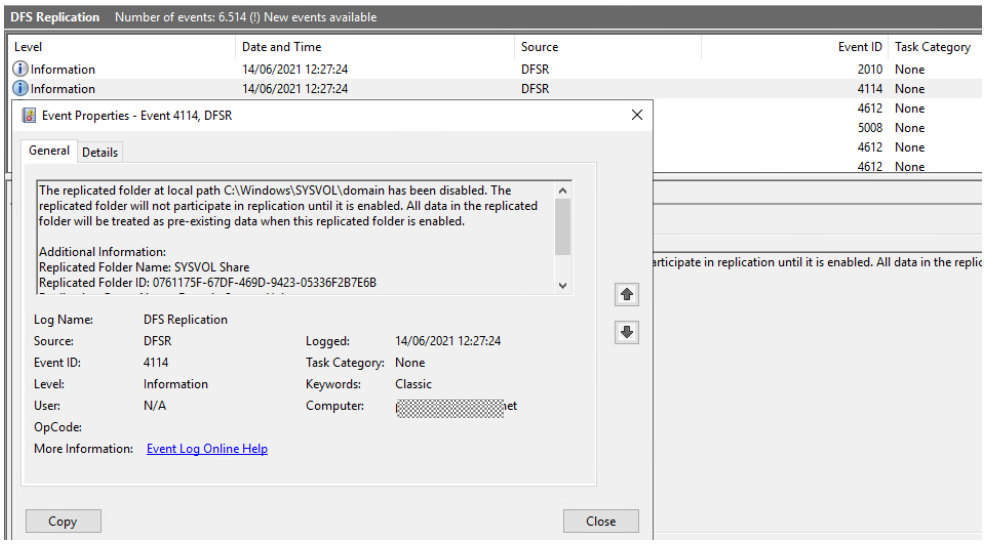

Ilustración 28, evento 4114 DFSR

- Volvemos a modificar el atributo: “msDFSR-Enabled” = “true”. (ver ilustración 29)

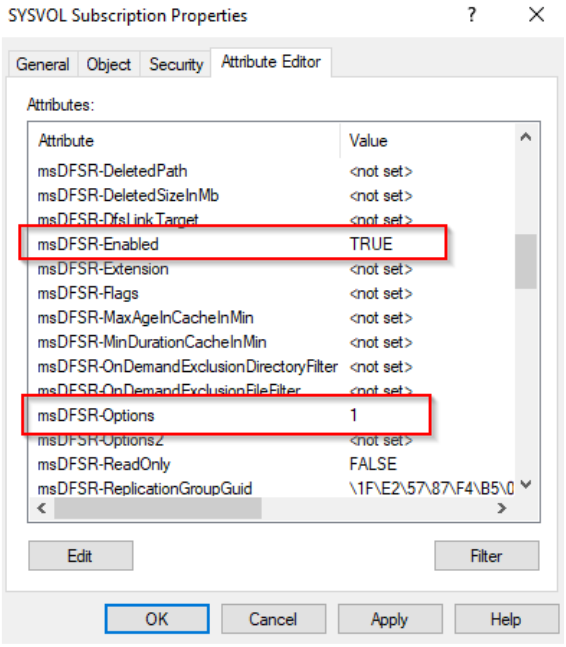

Ilustración 29, modificar atributo msDFSR-Enabled

- Y ejecutamos otra vez el comando “dfsrdiag pollad”.
- Comprobamos que se ha generado un evento “DFSR Replication” con id 4602. (ver ilustración 30)

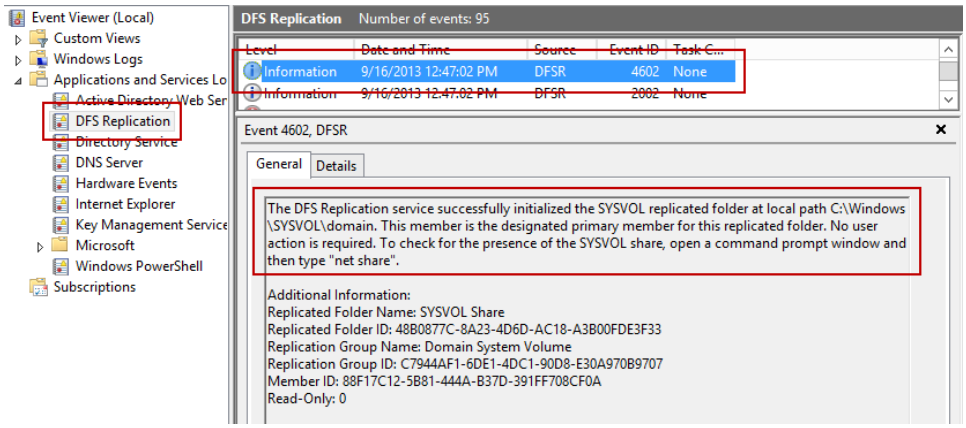

Ilustración 30, evento 4602 DFSR

- Reiniciamos la replicación con el comando "Restart-Service dfsr".

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
#Marcar SYSVOL como autoritativo
$as = (Get-Date).AddSeconds(-15) //tomamos tiempo

Set-ADObject -Identity $Doma -Replace @{"msDFSR-Enabled" = $false }
Set-ADObject -Identity $Doma -Replace @{"msDFSR-options" = 1 }


#Comprobar evento 4114 si aparece
$eventos = Get-Eventlog -LogName 'DFS Replication' | Where {$_.EventId -eq
4114}
$last = $eventos[0].TimeGenerated
if ($as -gt $last) {
      write-error "Evento DFS Replication no generado"
} else { write-host "Evento DFS Replication generado" -ForegroundColor
Green }

$as = (Get-Date).AddSeconds(-60) //tomamos tiempo para otro evento
#Volvemos a activar la replicacion
Set-ADObject -Identity $Doma -Replace @{"msDFSR-Enabled" = $true }

#Comprobar evento 4602
$eventos = Get-Eventlog -LogName 'DFS Replication' | Where {$_.EventId -eq
4602}
$last = $eventos[0].TimeGenerated
if ($as -gt $last) {
      write-error "Evento DFS Replication no generado"
} else { write-host "Evento DFS Replication generado" -ForegroundColor
Green }
```

Código 7, marcar SYSVOL como autoritativo

Cuando marcamos un recurso como autoritativo nos referimos a que, en el proceso de replicación, nuestros datos prevalezcan sobre el resto, es decir, si hay algún cambio en el recurso, éste no modificará sus datos.

5.1.3.8 Limpiar información de cuentas de máquina y metadata:

En este apartado vamos a comprimir tres pasos en uno, limpiaremos los objetos servidor, es decir todos los controladores de dominio que pertenecían al dominio y si en alguno de ellos hubiera algún rol FSMO se transferirán al controlador restaurado, la *metadata* DNS o registros de datos DNS de los servidores infectados.

Esto asegura que posteriormente no se tengan problemas de red entre los controladores de dominio.

Procedemos a eliminar información residual que puede verse desde la consola de Sitios y Servicios:

- Abrir la consola de Sitios y Servicios.
- Desplegar el contenedor de Sitios.
- Para cada uno de los Sitios.
    - Desplegar el contenedor Server.
    - Borrar cada DC que ya no exista excepto el que se está recuperando. (ver ilustración 31)

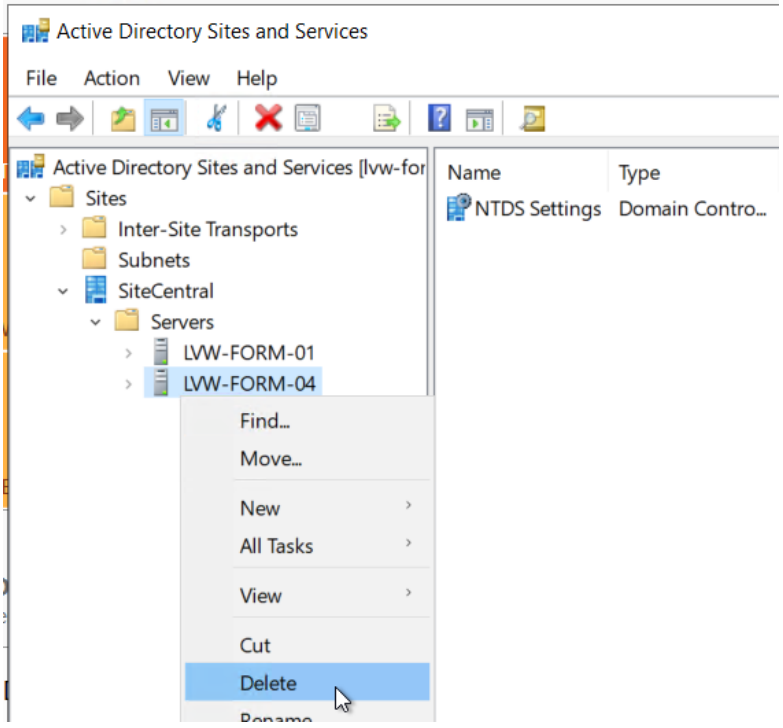

Ilustración 31, eliminamos DC's

Otro proceso de limpieza que debemos realizar es mantener la información mínima indispensable para los controladores de dominio que van a ser restaurados y eliminando toda la información anterior de los DC's que serán promocionados nuevamente, permitiendo que se registren con sus nuevos valores evitando inconsistencia de datos.

- Verificar en DNS que en la zona "_msdcs" aparece el GUID del DC que está recuperando y en la zona directa o dominio, aparece el Registro A del Servidor.
- Limpiar los registros de DNS de los controladores de dominio (Registros A, SRV, CNAME y NS) y dejar solo el del DC recuperado.
- Reiniciaremos con el comando "Restart-Service netlogon", para posibles errores. (ver ilustración 32)

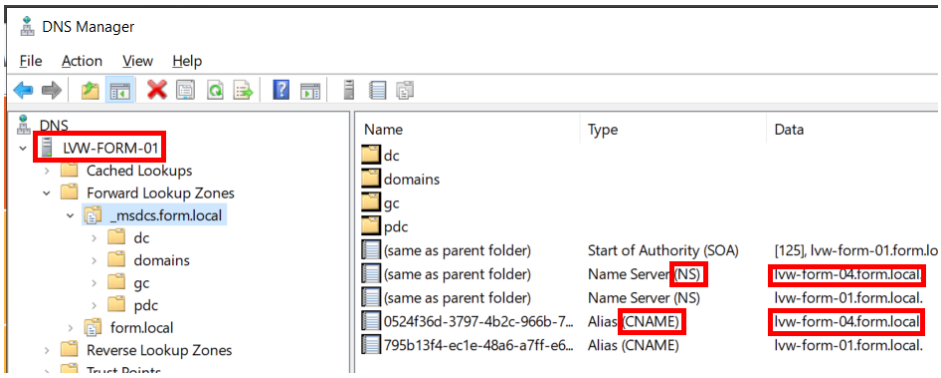

Ilustración 32, eliminación de datos DNS

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
#Limpieza de los objetos Server y DNS
$recu = hostname
$DCs = Get-ADDomainController -Filter *
$DNSZones = (Get-DnsServerZone).ZoneName

foreach ($DC in $DCs) {
    if ($recu -ne $DC.Name) {

        Get-ADObject -Identity "CN="$($DC.Name)",OU=Domain
Controllers,DC=form,DC=local" | Remove-ADObject -Recursive -Confirm $false
        Get-ADObject -Identity
"CN="$($DC.Name)",CN=Servers,CN="$($DC.Site)",CN=Sites,CN=Configuration,DC=
form,DC=local" | Remove-ADObject -Recursive -Confirm $false
        foreach ($zone in $DNSZones) {
            Get-DnsServerResourceRecord -ZoneName $zone -RRType SRV | where
{ $_.RecordData.DomainName -match $DC.Name } | Remove-
DnsServerResourceRecord -Force -ZoneName $zone
            Get-DnsServerResourceRecord -ZoneName $zone -RRType Ns | where
{$_.RecordData.NameServer -match $DC.Name } | Remove-
DnsServerResourceRecord -Force -ZoneName $zone
            Get-DnsServerResourceRecord -ZoneName $zone -RRType A | where
{$_.RecordData.IPv4Address.IPAddressToString -match $DC.IPv4Address } |
Remove-DnsServerResourceRecord -Force -ZoneName $zone
            Get-DnsServerResourceRecord -ZoneName $zone -RRType Cname |
where {$_.RecordData.HostNameAlias -match $DC.Name } | Remove-
DnsServerResourceRecord -Force -ZoneName $zone
        }}}
```

Código 8, eliminación de cunetas de máquina y metadata

5.1.3.9 Forzar sobre el controlador de dominio todos los roles FSMO:

Los roles FSMO son roles que los adquiere un controlador de dominio para que mande sobre el resto en algún aspecto, como por ejemplo el rol de "RID Pool Manager", es el responsable de suministrar al resto el pool de ID's necesario para crear usuarios u objetos de directorio.

Es necesario que cuando restauremos el dominio un controlador de dominio deba poseer el rol de PDC que es el más importante.

Este paso se realiza también con la consola PowerShell, así que es igual que en el script.

```
#Forzar sobre el DC todos los roles FSMO
Move-ADDirectoryServerOperationMasterRole -Identity $server -
OperationMasterRole 0,1,2,3,4 -force
#Para confirmar
netdom query fsmo
```

Código 9, forzar roles FSMO a nuevo DC

5.1.3.10 Corregir el propietario del rol FSMO en las particiones "ForestDNSZones" y "DomainDNSzones":

Nos dirigimos al registro "CN=Infrastructure de domain naming context" en el "ADSI Edit" y modificamos el atributo "FSMORoleOwner" y cambiamos el texto donde se introduce el nombre del *host*, cambiando al nombre del controlador de dominio que restauremos.

Este cambio sirve para que las particiones DNS "Forest" y "Domain", tengan claro cuál es el controlador de dominio que administra este rol.

Para corregir/verificar el propietario de estos "Naming Context" seguiremos los siguientes pasos:

- Abrimos la consola "ADSI Edit".
- Pulsamos botón derecho sobre "ADSI Edit" y seleccionamos "Connect To…".
- Seleccionamos "Select or type a Distinguished Name or Naming Context" y escribimos el DN de la partición "ForestDNSZones" en el caso del dominio form.local es "DC=ForestDNSZones,DC=form,DC=local". (ver ilustración 33)

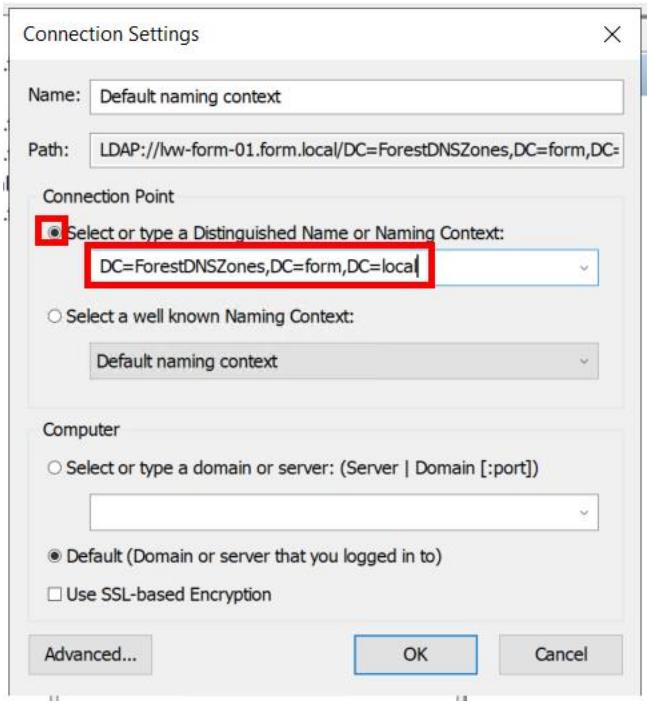

Ilustración 33, nueva conexión a "ForestDNSZones"

- Navegamos hasta el registro: "CN=Infrastructure" que está en la raíz de la partición y seleccionamos propiedades. (ver ilustración 34)

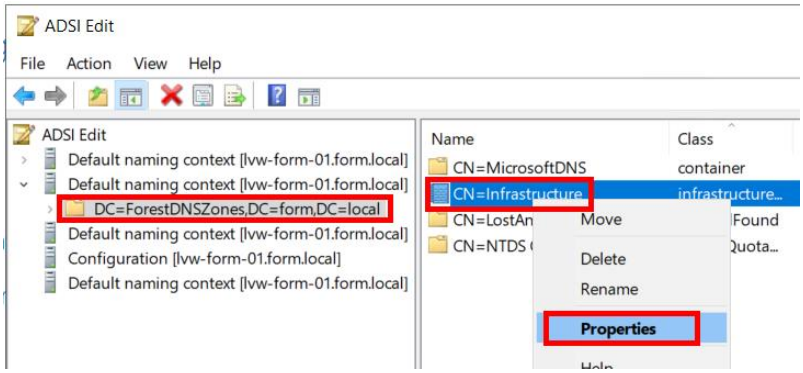

Ilustración 34, selección de carpeta infrastructura

- Editamos el atributo: "fSMORoleOwner".
- Corregimos el valor con los datos del nuevo propietario de la zona. (ver ilustración 35)

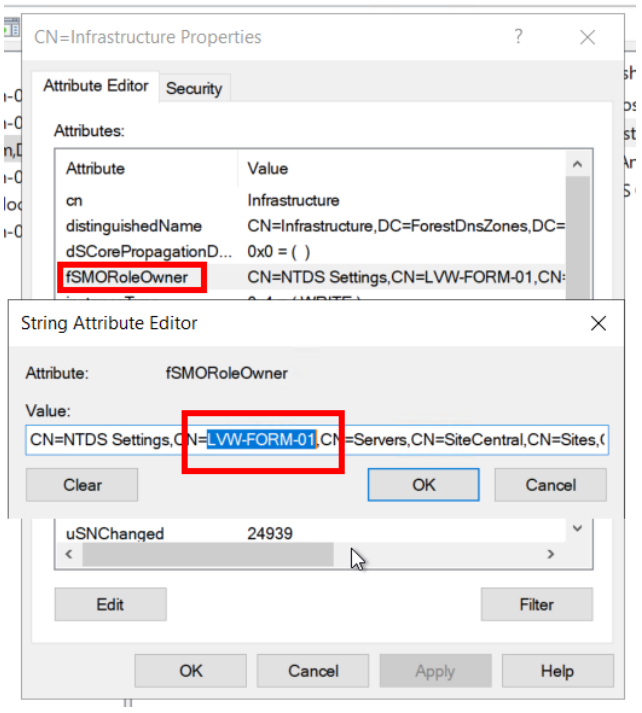

Ilustración 35, cambio de propietario fsmo a nuevo DC

- Corregimos el DN para que apunte al nuevo propietario del rol, para tener en cuenta hay que verificar el nombre del servidor y también el Site.
- Repetimos el proceso para la partición "DomainDNSZones"

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
#Corergir propietario ForestDNSZones y DomainDNSZones
$DomainFQDN = Get-ADDomain
Set-ADObject -Identity
"CN=Infrastructure,DC=DomainDnsZones,$($DomainFQDN.DistinguishedName)" -
Replace @{fSMORoleOwner = $((Get-ADDomainController -Identity
($DomainFQDN).PDCEmulator).NTDSSettingsObjectDN)}
Set-ADObject -Identity
"CN=Infrastructure,DC=ForestDnsZones,$($DomainFQDN.DistinguishedName)" -
Replace @{fSMORoleOwner = $((Get-ADDomainController -Identity
($DomainFQDN).PDCEmulator).NTDSSettingsObjectDN)}
```

Código 10, Corregir el propietario del rol FSMO en las particiones "ForestDNSZones" y "DomainDNSzones

5.1.3.11 Elevar el valor disponible del pool de RID y verificar:

Los pasos para seguir son los siguientes:

- Ejecutar “ADSIEdit.msc” para abrir la consola “ADSI Edit”.
- Desplegar el servidor para que muestre la partición de dominio “Default Naming Context”.
- Expandir “Default Naming Context”.
- Expandir “DC=form,DC=local”.
- Ir a “CN=System”.
- En el panel derecho, hacer clic derecho sobre “CN=RID Manager$” para abrir su ventana de Propiedades. (ver ilustración 36)

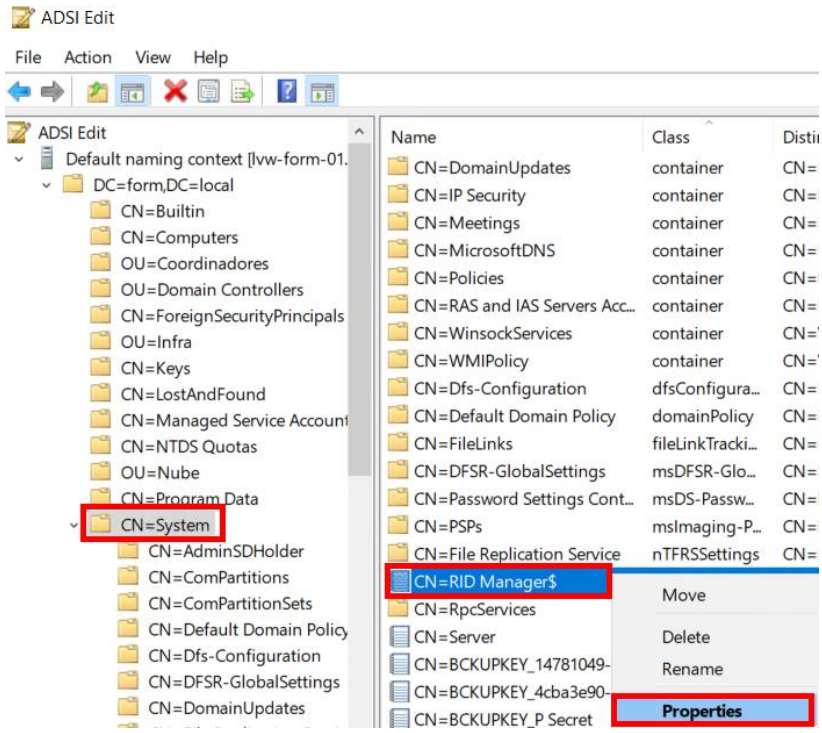

Ilustración 36, selección de las propiedades del RID

- En la lista “Select a property to view”, seleccionar “RidAvailablePool”, Copiar el valor del campo “Value(s)” sobre “Edit Attribute” y sumar 100.000 al valor, en el momento que se ha realizado la simulación el valor antiguo es: 4.611.686.014.132.955.717 y por tanto el nuevo será: 4.611.686.014.13**3.0**55.717, cambiamos el valor en el atributo y “OK”.
- Para verificar que se ha cambiado el valor del “RID pool” hay que ejecutar una serie de comandos y crear un usuario prueba. (ver ilustración 37)

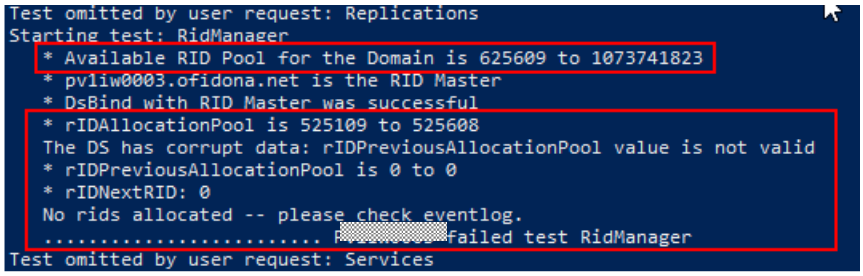

Ilustración 37, RID Pool 1

- Observamos que el pool disponible ya ha subido en 100.000 unidades respecto a la comprobación inicial, pero que el pool disponible es inválido y el siguiente RID disponible es 0. Debemos crear un usuario ahora. (ver ilustración 38)

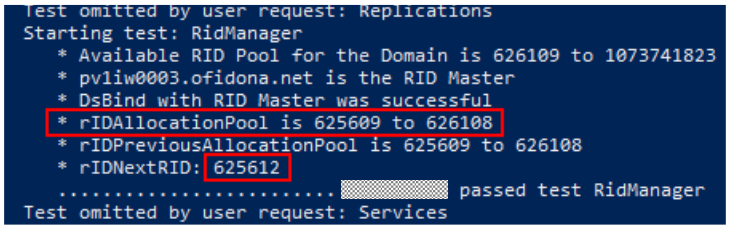

Ilustración 38, RID Pool 2

- Volvemos a verificar el "riDAllocationPool" y este ya estará de forma correcta.

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
#Elevar valor del RID en 100000 y verificar
function raiseRIDPool($amount=100000){
    $domain = get-addomain
    $currentRidPool = get-adobject "CN=RID
Manager$,CN=System,$($domain.DistinguishedName)" -properties
rIDAvailablePool | select -expand rIDAvailablePool
    Write-verbose "RidPool is currently $currentRidPool will be raised to
$($currentRidPool + $amount)"
    set-adobject "CN=RID Manager$,CN=System,$($domain.DistinguishedName)" -
replace @{ridavailablePool=($currentRidPool + $amount)}
    $Domain = New-Object System.DirectoryServices.DirectoryEntry
    $DomainSid = $Domain.objectSid
    $RootDSE = New-Object
System.DirectoryServices.DirectoryEntry("LDAP://RootDSE")
    $RootDSE.UsePropertyCache = $false
    Write-Verbose "Invalidating the rid pool for the current domain
$($domain.name)"
    $RootDSE.Put("invalidateRidPool", $DomainSid.Value)
    $RootDSE.SetInfo()
}
raiseRIDPool
New-ADUser -Name "test" -Enabled $false
```

Código 11, elevar el valor del RID Pool

Cuando necesitamos restablecer un dominio, uno de los problemas que nos encontraremos es que el valor del RID será incorrecto y habrá conflicto de ID's a la hora de crear objetos. Por eso es necesario incrementar en un valor, preferiblemente 100.000, para que no se produzcan más conflictos asignando ID's.

5.1.3.12 Reiniciar la contraseña del equipo y de la cuenta "krbtgt":

Por seguridad y porque seguramente las contraseñas están corruptas, necesitaremos reiniciar las contraseñas del equipo que estamos restaurando y de la cuenta "krbtgt" que se dedica a asignar tiques Kerberos para el dominio.

Para resetear la contraseña del equipo que restauramos procedemos bajo el comando siguiente "Netdom resetpwd /server:< domain controller name> /userD:Andres /passwordD:*", donde <domain controller name> es el DC local que estamos recuperando, este proceso lo deberemos lanzar dos veces. (ver ilustración 39)

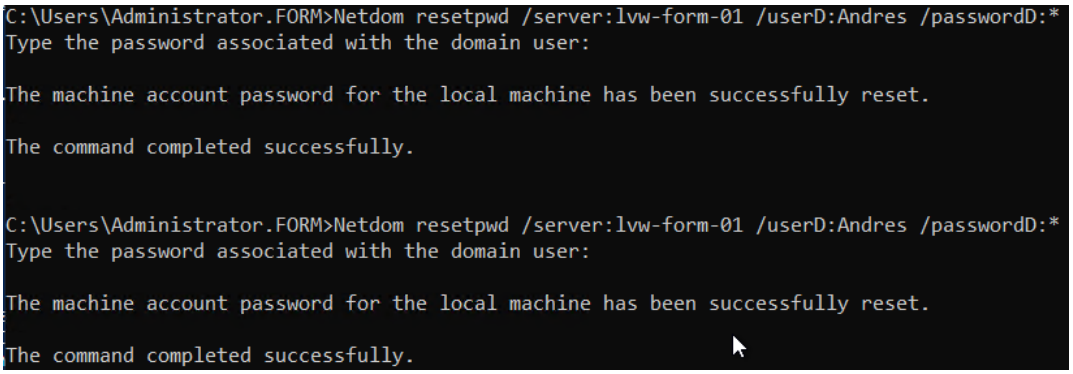

Ilustración 39, reinicio de contraseña administrador y Kerberos

Para cambiar la contraseña de la cuenta "krbtgt" se procede de igual manera.

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
$hostn = hostname
$DomainFQDN = Get-ADDomain
$i=1
for (;$i -le 2;$i++) {
      Reset-ComputerMachinePassword -Server $hostn -Credential
($DomainFQDN).DNSRoot\Administrator
      $passkrbtgt = Get-RandomPassword 20
      Set-ADAccountPassword -Identity krbtgt -Reset -NewPassword
(ConvertTo-SecureString -AsPlainText $passkrbtgt -Force)
}
```

Código 12, reinicio de contraseña cuenta administrador y krbtgt

5.1.3.13 Habilitar el controlador de dominio como "Global Catalog":

"Global Catalog" es un diccionario el cual, obtiene información de cualquier objeto del bosque, es necesario habilitarlo, pero no es importante cuando restauremos. En un principio si hemos hecho una buena restauración del controlador de dominio objetivo más adecuado, no será necesario habilitarlo ya que este controlador de dominio ya tendrá esta característica.

Para habilitar el DC como catálogo global:

- Abrir "Active Directory Sites and Services".
- En la consola hacer un doble-clic sobre el DC recuperado.
- Hacer clic de botón derecho sobre el "NTDS Settings" y escoger "Properties".
- Marcar la opción de "Global Catalog". (ver ilustración 40)

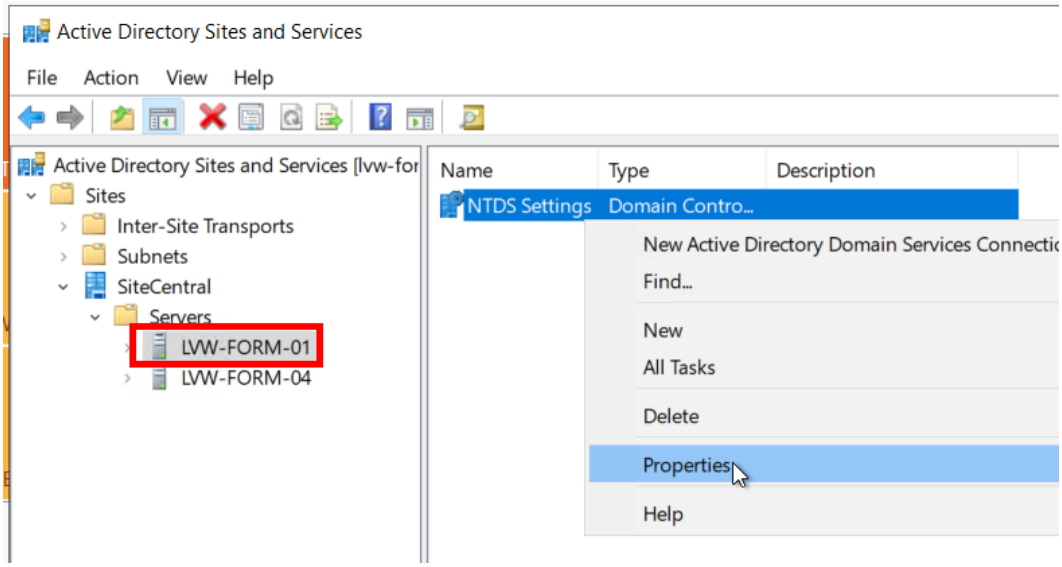

Ilustración 40, propiedades de "NTDS settings"

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
$hostn = Get-ADDomainController -filter * | where {$_.Name -eq $(hostname)}
$path1 = 'HKLM:\SYSTEM\CurrentControlSet\Services\NTDS\Parameters'
$Repl = (Get-ItemProperty -Path $path1)."Global Catalog Promotion Complete"
if (!$hostn.IsGlobalCatalog) {
    Set-ADObject -Identity $hostn.NTDSSettingsObjectDN -Replace
@{options='1'}
}
if ( $Repl -eq 0){
    New-ItemProperty -Path $path1 -Name 'Global Catalog Promotion Complete'
-Value 1 -PropertyType DWord
    #Si modifoca o crea reinicia máquina
    write-host "Global Catalog activado" -ForegroundColor Green
} else {write-host "Global Catalog activado" -ForegroundColor Green}
```

Código 13, habilitar GC en el DC

5.1.3.14 Verificar la configuración horaria en el nuevo PDC:

Este cambio se debe a que el rol PDC posee la característica del reloj primario, es decir, este debe configurar su hora con respecto a un servidor externo de Microsoft para sincronizar su hora, y el resto de los controladores deben hacerlo sobre este controlador para así funcionar todos de manera síncrona con el mismo horario, ya que pueden provocar conflictos en las réplicas.

Revisar con el siguiente comando que aplica la configuración correcta, desde un CMD lanzamos "W32tm /dumpreg /subkey:parameters".

La traducción a lenguaje PowerShell para automatizar este paso es el siguiente:

```
$path2 = 'HKLM:\SYSTEM\CurrentControlSet\Services\W32Time\Parameters'
$ntp = (Get-ItemProperty -Path $path2)."NtpServer"

if ($ntp -ne "URL-servidor-NTP") {
    Set-ItemProperty -Path $path2 -Name "NtpServer" -Value " URL-servidor-
NTP "
    Set-ItemProperty -Path $path2 -Name "Type" -Value "NTP"
}
```

Código 14, verificar configuración horaria

Éste será el último paso para restaurar el primer controlador de dominio. A partir de ahora hay que conseguir ordenar y configurar de manera adecuada los scripts para su ejecución en el controlador de dominio deseado.

En este caso hemos configurado los scripts en tres documentos, ya que, hay dos momentos clave donde es posible que el servidor se deba reiniciar. Para lanzar los scripts deberemos conectarnos por RDP al servidor que hemos restaurado de la copia de seguridad y ejecutamos los scripts de manera secuencial.

- Primer documento "D&R_1.ps1", en este documento agruparemos los scripts de la configuración de la red y verificación del dominio. Si se han efectuado cambios deberemos reiniciar el servidor.
- Segundo documento "D&R_2.ps1", habilitaremos la cuenta de administrador y lo incluiremos en los grupos necesarios, si se han efectuado cambios necesitaremos lanzar el próximo documento autenticado como administrador.
- Tercer documento "D&R_3.ps1", ahora ejecutaremos el resto del script.

Una vez ejecutados estos scripts en el servidor, ya tendremos recuperado el primer controlador de dominio, y podremos continuar con el procedimiento de recuperación del dominio.

5.2 Cuantificar DC's

La idea de este apartado es averiguar cuantos controladores de dominio son necesarios para mantener a la empresa produciendo, pero con los servicios mínimos.

5.2.1 Caso Microsoft

Para resolver esta duda tuvimos que abrir un caso con Microsoft para intentar guiarnos hasta la solución que buscábamos. Tras varias semanas de discusión, el técnico asignado a nuestro caso nos hizo llegar una documentación en la que se calculaba la capacidad de los controladores de dominio según la carga, no era exactamente lo que buscábamos, pero nos pudo resolver algunas dudas al respecto.

Esta documentación "Capacity planning for Active Directory Domain Services", habla sobre cinco componentes esenciales para determinar la carga y en qué modo debemos modificarla.

- Memoria, se refiera a la memoria RAM del DC, la capacidad y frecuencia.
- Red, habla sobre la arquitectura de la red, la banda ancha, tipo de conexión, etc.
- Almacenamiento, espacio disponible, operaciones de entrada/salida, tecnología NAS y SAS, distribución RAID preferente, memoria caché y operaciones de lectura y escritura
- Procesador, pone especial énfasis en el cuello de botella generado por el almacenamiento, cantidad de núcleos y velocidad del procesador
- NetLogon, tipo de autenticación y concurrencia a la hora de autenticar.

Estos serán algunos de los aspectos a tener en cuenta a la hora de planificar nuestros controladores de dominio, que características obtendrán y la cuestión principal cuantos son necesarios.

Siguiendo la documentación aportada por el técnico de Microsoft, hay que analizar en cada DC una cantidad inmensa de datos que actualmente no están a nuestra disposición, pero con un estudio estimado facilitado por los técnicos, consideran oportuno que diez controladores de

dominio serían suficientes para continuar en servicios mínimos y funcionar con la herramienta recuperada, Directorio Activo.

5.3 Despliegue en la nube

En este apartado desplegaremos en una plataforma nube, GCP o Azure, toda la infraestructura necesaria para formar un dominio, asimismo, el lenguaje utilizado para ello será Terraform.

Terraform es un software de infraestructura como código desarrollado por HashiCorp. Permite a los usuarios definir y configurar la infraestructura de un centro de datos en un lenguaje de alto nivel. Este lenguaje sirve para desplegar en cualquier nube una infraestructura a nuestro gusto, y nosotros lo utilizaremos para desplegar servidores y crear nuestro directorio activo.

Vamos a explicar algunos conceptos necesarios para poder interpretar nuestro código y averiguar los que ocasionara en nuestra nube:

- Variables: También se utiliza como variables de entrada, es un par clave-valor utilizado por los módulos Terraform para permitir la personalización.
- Proveedor: Es un complemento para interactuar con las API de servicio y acceder a sus recursos relacionados.
- Módulo: Es una carpeta con plantillas Terraform donde se definen todas las configuraciones
- Estado: Consiste en información en caché sobre la infraestructura administrada por Terraform y las configuraciones relacionadas.
- Recursos: Se refiere a un bloque de uno o más objetos de infraestructura (instancias de cómputo, redes virtuales, etc.), que se utilizan para configurar y administrar la infraestructura.
- Fuente de datos: Los proveedores lo implementan para devolver información sobre objetos externos a Terraform.
- Valores de salida: Estos son valores de retorno de un módulo Terraform que pueden ser utilizados por otras configuraciones.
- Planificar: Es una de las etapas en las que determina qué se debe crear, actualizar o destruir para pasar del estado real / actual de la infraestructura al estado deseado.

- Aplicar: Es una de las etapas donde se aplican los cambios de estado real / actual de la infraestructura para pasar al estado deseado. (Avi, 2022)[8]

Así es como funciona Terraform, independientemente de dónde se despliegue:

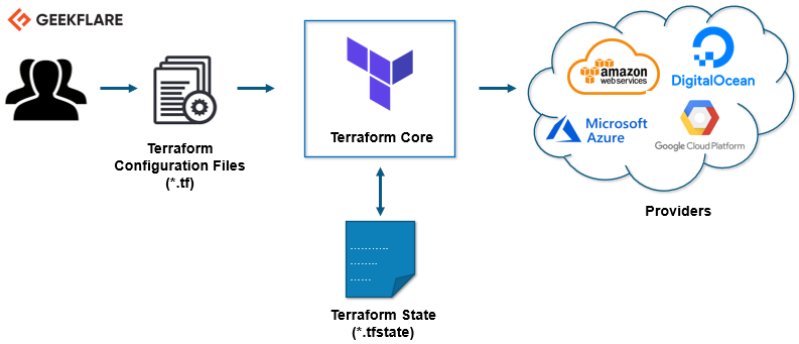

Ilustración 41, modelado del funcionamiento de Terraform

Una vez entendido estos conceptos, mostraremos el código principal de nuestro Terraform:

```
################################################
#DESPLIEGUE DEL PRIMER CONTROLADOR DE DOMINIO
################################################
#Creacion de maquina windows server modulo
module "vm-windows-main" {
  count               = var.instance_count
  source              = "git@gitlab.URL-Module"
  network             = ""
  project_id          = var.project_id
  subnetwork          = var.subnetwork
  subnetwork_project  = var.subnetwork_project
  hostname                        = "lvw-gcpad-main"
  network_tags          = ["zds-labpri-vm-windows"]
  service_account     = ""
  metadata            = [
    {key = "windows-startup-script-url", value = "gs://bucket-andres-
pruebas/age.ps1"},
    {key = "enable-oslogin", value = "TRUE"}
  ]}
################################################
#DESPLIEGUE DEL RESTO DE CONTROLADORES DE DOMINIO
################################################
#Creacion de maquina windows server modulo
module "vm-windows-resto-dc" {
  count               = "2"
  source              = " git@gitlab.URL-Module"
  network             = ""
  project_id          = var.project_id
  subnetwork          = var.subnetwork
  subnetwork_project  = var.subnetwork_project
  hostname                        = "lvw-gcpad-${count.index + 1}"
  network_tags            = ["zds-labpri-vm-windows"]
  service_account     = ""
  metadata            = [
    {key = "windows-startup-script-url", value = "gs://bucket-andres-
pruebas/start${count.index + 1}.ps1"},
    {key = "enable-oslogin", value = "TRUE"}
  ]}
```

Código 15, despliegue de máquinas virtuales en GCP

Para generar nuestra infraestructura necesitaremos diferenciar dos módulos diferentes, uno servirá para desplegar el primer controlador de dominio y el otro para desplegar el resto de los controladores de dominio. (HashiCorp, 2022)[9]

El fragmento inicial describe como debería ser ese primer servidor mediante el uso de variables, que tenemos definidas en el archivo ".tfvars", las cuales están declaradas en "vars.tf". Para comprender mejor el código vamos a explicar las variables:

- "count" la necesitamos para cuantificar el número de instancias a desplegar, por defecto está declarada a 1.
- "source", URI de GitLab dónde se encuentra el módulo que vamos a utilizar, en nuestro caso es "vm-windows".
- "network", no definiremos nada ya que utilizaremos una subred definida en VPC de Google Cloud.
- "project_id", contiene el nombre del proyecto en GCP donde vamos a desplegar las máquinas virtuales.
- "subnetwork y subnetwork_project", estas variables hacen referencia a la red VPC compartida de otro proyecto, por eso debemos declarar también el proyecto donde se comparte la subred.
- "hostname", el nombre de la instancia.
- "network_tags", aquí vamos a definir las reglas firewall, es decir, que puertos tendremos abiertos, por ejemplo, RDP o SSH. Se hace mediante tags para que sea más fácil administrar las reglas.
- "service_account", es una bolsa de privilegios para poder manejar diferentes aspectos en Google Cloud, por ejemplo, el permiso de desplegar instancias. Pero no es una cuenta vinculada a un usuario, es impersonal. En nuestro caso la dejaremos vacía porque ya tenemos privilegio total en nuestro proyecto.
- "metadata", esta variable es una lista de objetos clave/valor, la cual se declara para añadir opciones adicionales a nuestra instancia. El objeto que debemos tener en cuenta en "windows-startup-script-url", esta clave la utilizaremos para definir un script una vez creemos la instancia, que para el despliegue del primer controlador de dominio será el script definido anteriormente como "D&R_1.ps1", y para el resto de los controladores será un script, véase código 16.

```
#Despliegue del resto de controladores de dominio
Import-Module ServerManager
#Instalar el rol de AD
Add-WindowsFeature "AD-Domain-Services" -IncludeManagementTools -Restart
##Configuramos DNS
Set-DnsClientServerAddress -InterfaceAlias ethernet0 -ServerAddresses
$ip_dc_recuperado
shutdown /r
```

Código 16, script de despliegue para el resto de DC's

Este script instala el servicio de "AD-Domain-Services", que es el servicio de directorio activo. Además, tendremos que configurar el DNS para que apunte al nuevo controlador de dominio recuperado, ya que éste tendrá instalado el servicio de servidor DNS, para así identificar el dominio.

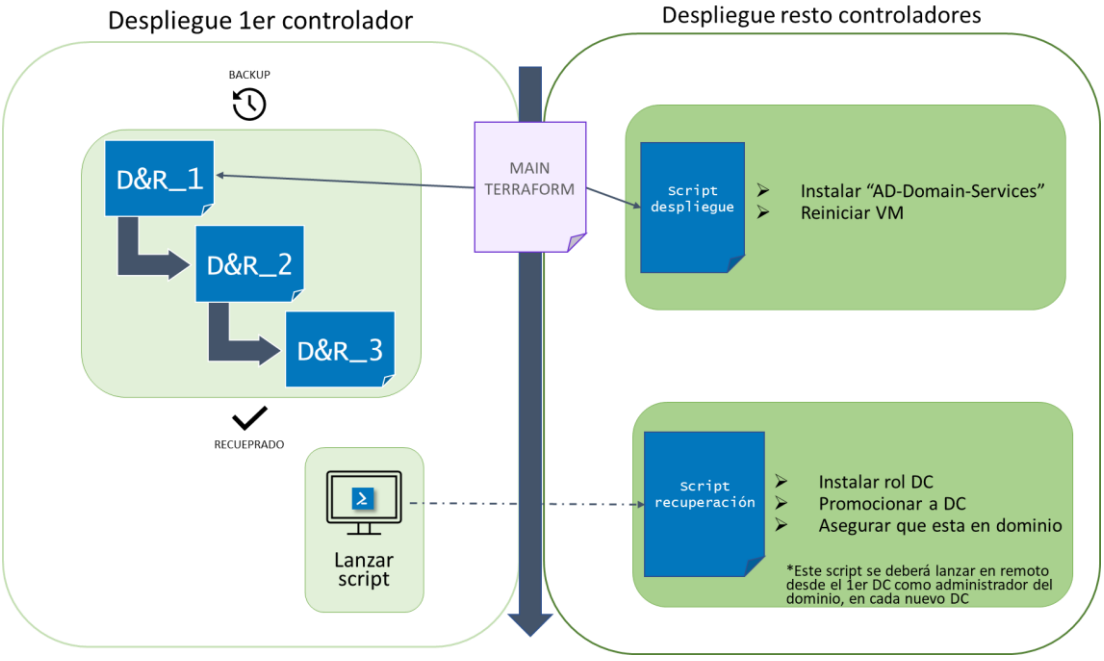

Ilustración 42, Despliegue en la nube

Una vez desplegados los servidores y la infraestructura conformada, procederemos a la inserción de los nuevos servidores desplegados al dominio. Para conseguir este estado necesitaremos ejecutar un script que, deberemos lanzar desde nuestro primer controlador de manera remota con el comando "invoke-command" de PowerShell, este comando puede ejecutar cualquier comando en un servidor, con el argumento "-computerName" para identificar el nombre del servidor, y el argumento "-scriptBlock" para el comando que deseemos lanzar.

Lanzamos los comandos necesarios para asegurarnos de que ya pertenezca al dominio, le instalamos el rol de controlador de dominio y lo promocionamos para que forme parte de los controladores del dominio. Incluiremos en el anexo todos los scripts necesarios para el despliegue y configuración de los servidores.

5.4 Acciones tras la recuperación

Tras el restablecimiento del directorio activo y todos sus componentes necesarios para recuperar la producción, necesitaremos aplicar configuraciones adicionales. En el caso de haber sufrido un ataque de *ransomware*:

- Reiniciaremos todas las contraseñas de los usuarios, esto se puede hacer de manera automática aplicando la opción de cuenta, que en el siguiente inicio de sesión cambie la contraseña, y para aplicar esto de manera global necesitaremos crear un script.

```
#Cambiar la contraseña en el siguiente inicio de sesión para las cuentas
que no sean administrador o hayan sido

$nadmin = $nadmin = Get-ADUser -filter * -Properties adminCount | where
{$_.admincount -lt 0}

foreach ($user in $nadmin) {
    try {
        if (Get-ADUser -filter { PasswordNeverExpires -eq $true } | where
{$_.Name -eq $user.Name}) {
            Set-ADUser -Identity $user.Name -PasswordNeverExpires $false
        }
        Set-ADUser -Identity $user.Name -ChangePasswordAtLogon $true
        Write-Host "Cambio para $($user.SamAccountName) efectuado con
exito" -ForegroundColor Green
    } catch {
        Write-Host "Fallo al efectuar el cambio a $($user.SamAccountName)"
-ForegroundColor Red
    }
}
```

Código 17, script para el cambio de contraseñas

  Seleccionamos las cuentas que no sean administrador porque en la recuperación del primer controlador de dominio se haría este proceso, y se podría sobrescribir.

- Lanzar una réplica completa a los DC's del bosque y comprobar el estado de la replicación. Esto se podrá conseguir con la ejecución de estos comandos.

```
Repadmin /syncall Dcname /S /A /e /P
Repadmin /replsummary
```

Código 18, llamada para reiniciar replica

Este estado en el que nos encontramos, y si no ha surgido ningún imprevisto, podremos funcionar con nuestro nuevo directorio activo en la nube después de haber sufrido un ataque de *ransomware*, y no lo hemos recuperado de cualquier manera, se ha hecho de manera automatizada y reduciendo el tiempo de recuperación en un 74%, aunque se ha simulado en un entorno de laboratorio.

6. Implantación y pruebas

6.1 Despliegue en laboratorio

Para simular un entorno parecido al de una empresa, a muy pequeña escala, crearemos un directorio activo desde la creación hasta la copia de seguridad. Para probar la efectividad de nuestros scripts y que después probaremos en la nube.

6.1.1 Despliegue On-premise

El primer paso que realizamos fue establecer un entorno donde poder hacer las pruebas, en nuestro caso nos facilitaron un entorno de laboratorio virtualizado. Con la herramienta vSphere client de VMWare, desplegamos las máquinas virtuales necesarias para el proceso de recuperación. Con ayuda del equipo de virtualización nos desplegaron las máquinas virtuales.

6.1.2 Creación del directorio activo

Para proceder con la recuperación primero debíamos crear un dominio nuevo, que llamamos “form.local”. Añadimos nuevos servidores al entorno para simular un dominio, que más adelante les instalaremos el rol de directorio activo y el de controlador de dominio.

Ahora promocionamos los nuevos controladores de dominio y realizaremos comprobaciones con herramientas de diagnóstico del directorio activo para ver el estado de este, además, los roles FSMO asignarlos al servidor que le vayamos a realizar la copia de seguridad.

Creamos objetos de prueba, unidades organizativas usuarios, equipos, GPO (Directivas de grupo), etc. Los podremos crear mediante la consola grafica o por comandos PowerShell. (ver ilustración 43)

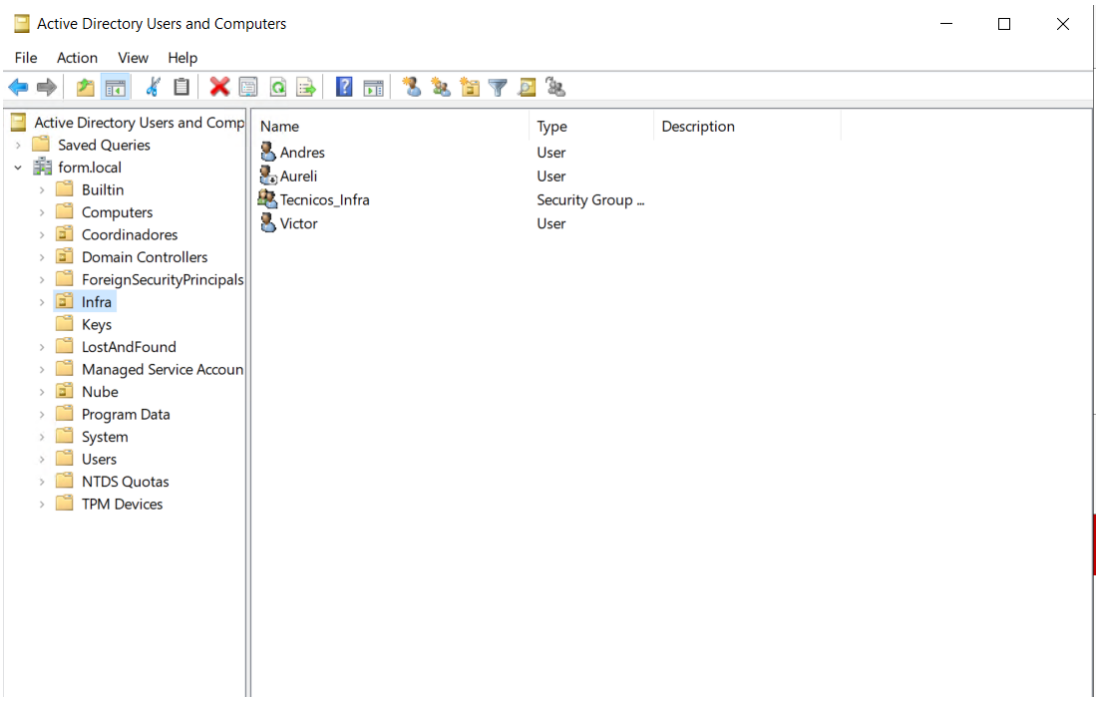

Ilustración 43, creación manual del directorio activo

6.1.3 Realización de la copia de seguridad

En el momento que tengamos nuestro dominio completamente construido, pasaremos a la realización de la copia de seguridad. Si aun no teníamos el rol de "Windows Server Backup", lo instalaremos. Hay que tener en cuenta el tipo de copia de seguridad, para este caso será de tipo BMR (Bare Metal Recovery). (ver ilustración 44)

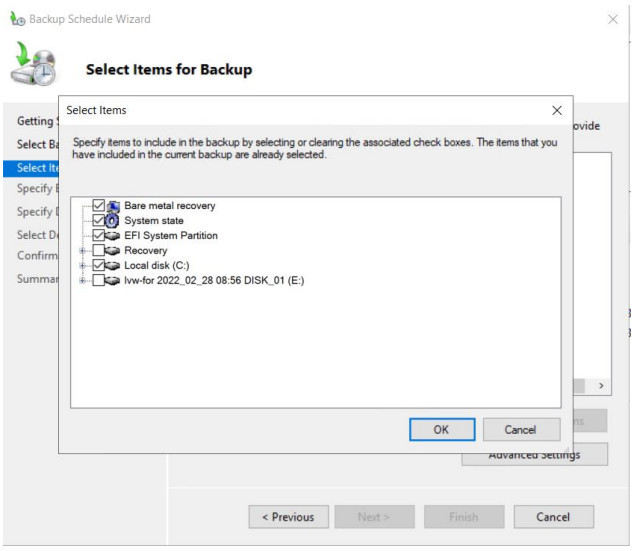

Ilustración 44, asistente para la copia de seguridad

El siguiente paso será, habilitar en la máquina virtual un disco duro con capacidad suficiente para albergar la copia de seguridad, habilitaremos un disco de 100GB.

- Reiniciamos el servidor en modo DSRM (Directory Services Restore Mode)
- En nuestro caso restauraremos la copia en modo autoritativo, para que no pida replicas al resto de DC's en cuanto se restaure.
- Copiamos completamente de la copia de seguridad y no una copia parcial.
- Y comprobamos que se ha efectuado bien la copia "repadmin/showbackup". (ver ilustración 45)

```
PS C:\Users\Administrator.FORM\Desktop> repadmin /showbackup

Repadmin: running command /showbackup against full DC localhost


Loc.USN                         Originating DSA  Org.USN  Org.Time/Date        Ver Attribute
=======                         =============== ========= =============        === =========
DC=ForestDnsZones,DC=form,DC=local
  48404       a4dbed44-5afc-45d4-a02a-b8843de7824d     48404 2022-03-24 09:35:31   19 dSASignature
DC=DomainDnsZones,DC=form,DC=local
  48403       a4dbed44-5afc-45d4-a02a-b8843de7824d     48403 2022-03-24 09:35:31   19 dSASignature
CN=Schema,CN=Configuration,DC=form,DC=local
  48402       a4dbed44-5afc-45d4-a02a-b8843de7824d     48402 2022-03-24 09:35:31   19 dSASignature
CN=Configuration,DC=form,DC=local
  48401       a4dbed44-5afc-45d4-a02a-b8843de7824d     48401 2022-03-24 09:35:31   19 dSASignature
DC=form,DC=local
  48400       a4dbed44-5afc-45d4-a02a-b8843de7824d     48400 2022-03-24 09:35:31   19 dSASignature
```

Ilustración 45, ver estado de replicas

6.2 Despliegue en la nube

Google Cloud Platform la plataforma en la nube de Google, en ella vamos a desplegar de manera automatizada los recursos necesarios para la implantación y pruebas.

6.2.1 Creación de las instancias

A la hora de crear las instancias o máquinas virtuales sobre terraform, hay múltiples maneras, vamos a explicar las que hemos utilizado y las que finalmente utilizaremos para la automatización:

- Mediante la consola Google Cloud Console, por interfaz web, no es la manera que buscamos de crear máquinas virtuales. (ver ilustración 46)

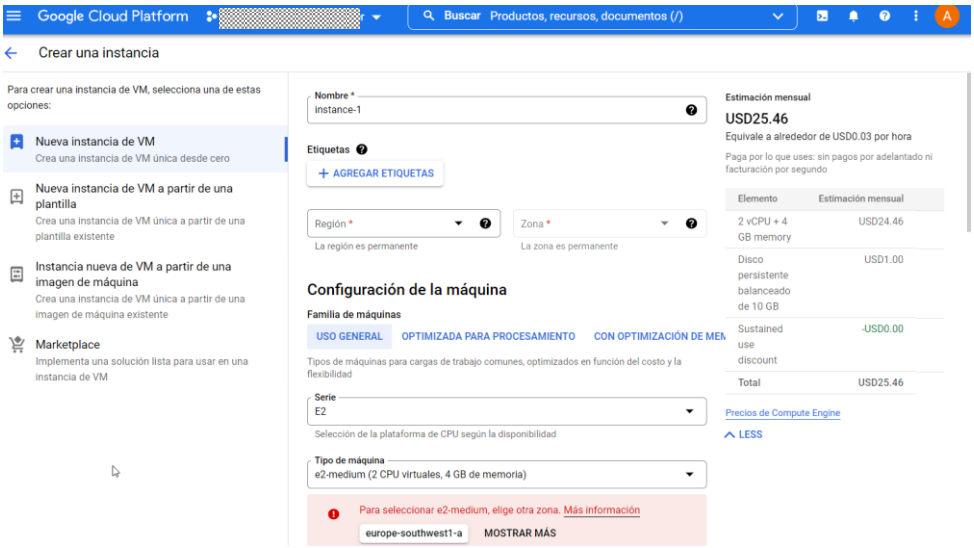

Ilustración 46, creación manual de VM

- Con la utilización de un lenguaje de infraestructuras, como es terraform, u otras posibilidades por código, la consola Shell de google, API, C#, Go, Java, Node.js, PHP, Python o Ruby. En nuestro caso vamos a utilizar Terraform. Esta será la mejor manera de desplegar maquina en la nube.

```
#Recurso para crear instacia
resource "google_compute_instance" "dev" {
name         = "pruebaserver" # name of the server
project      = var.project_id
machine_type = "e2-medium" # machine type refer google machine
types
zone         = var.zone # `a` zone of the selected region
tags         = ["zds-labpri-automata", "zds-labpri-bastion"]

boot_disk {
        initialize_params {
        image = "debian-cloud/debian-10"
        }
}
network_interface {
      subnetwork          = var.subnetwork
      subnetwork_project  = var.subnetwork_project
}
}
```

Código 19, creación automatizada con Terraform de VM en GCP

Con Terraform también hay múltiples maneras de crear una máquina virtual, y para este lenguaje se llaman recursos o módulos, que ya explicamos anteriormente. Para probar como funcionaban los despliegues empecé utilizando recursos de creación de instancias de Linux, y después lo hice mediante plantillas de instancias.

Esta instancia se llamará prueba server, con un sistema operativo debian-10 como disco de arranque y la red predeterminada definida en el documento ".tfvars", el recurso se llamará "dev".

Para su despliegue necesitaremos tener acceso a nuestro bastión por SSH, este será el proceso para desplegar el recurso:

```
#comando para lanzar terraform
Terraform init
#comando para ver que vamos a lanzar
Terraform plan -tarjet Google_compute instance.dev.id
#comando para aplicar el plan
Terraform apply -tarjet Google_compute instance.dev.id
```

Código 20, instrucciones de preparación y ejecución de Terraform

El argumento "target" lo utilizaremos para aplicar el terrafrom a sólo ese recurso seleccionado, en el caso de que tengamos más recursos descritos en el "main" de terraform.

La salida de las instrucciones del código 20 las visualizamos en la ilustración 47.

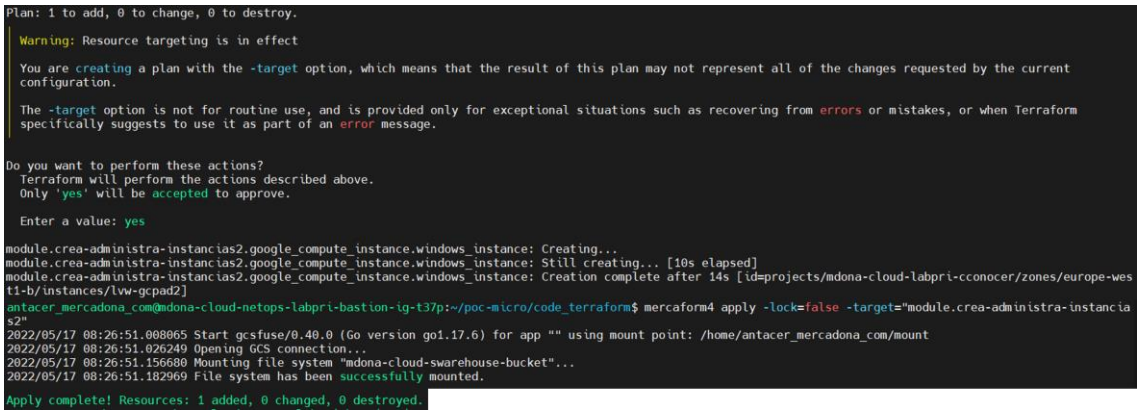

Ilustración 47, salida de las llamadas con Terraform

Finalmente nos decantamos por utilizar módulos, ya que, existía uno para el despliegue de instancias Windows en el GitLab de la compañía, creado internamente por técnicos del equipo de nube. Esto nos facilitará el proceso de creación de instancias Windows, con algunos requisitos ahí parametrizados invisibles a nuestra declaración.

Como visualizamos en el código de arriba, en desarrollo, seguiremos ese código para la creación de la maquina Windows en el laboratorio de *cloud*. Una vez desplegada la máquina, podremos conectarnos por RDP configurando la contraseña Windows para el primer inicio de sesión, desde la consola de la instancia.

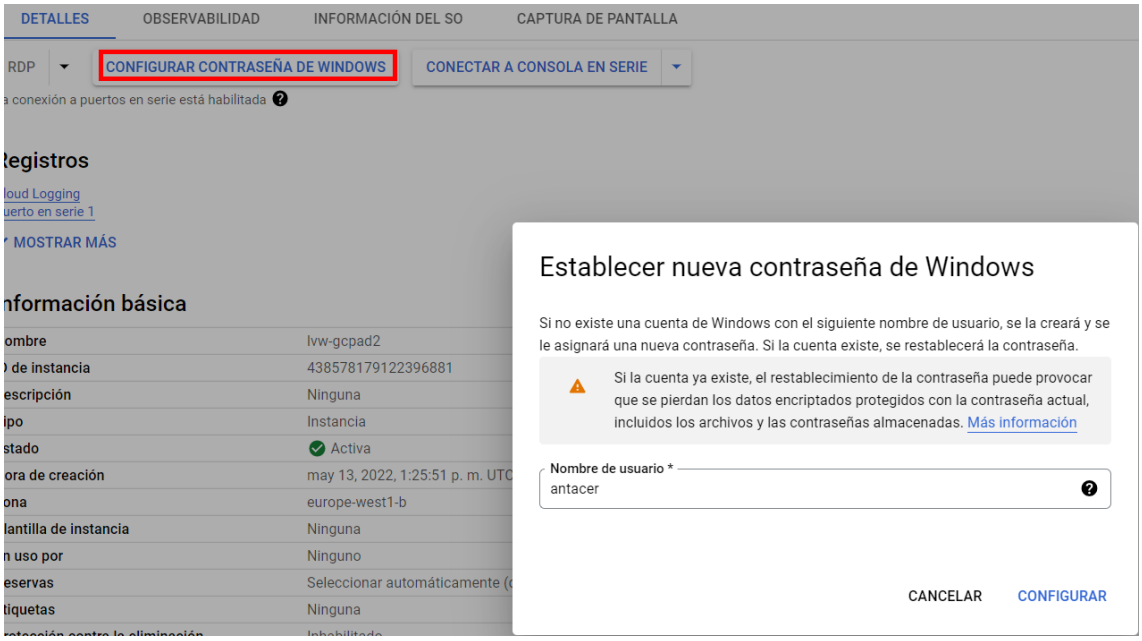

Ilustración 48, creación de usuario administrador para VM en GCP

También se puede establecer nueva contraseña mediante comandos gcloud, desde el Shell.

Finalmente, para la simulación de la recuperación ante un desastre, decidimos desplegar 11 instancias, una como primer controlador de dominio y las diez restantes para los controladores de dominio necesarios para mantener la carga. (ver ilustración 49)

```
# module.crea-administra-instancias.google_compute_instance.windows_instance will be created
+ resource "google_compute_instance" "windows_instance" {
...
# module.crea-administra-instancias2[9].google_compute_instance.windows_instance will be created
+ resource "google_compute_instance" "windows_instance" {
Plan: 11 to add, 0 to change, 0 to destroy.
```

Ilustración 49, salida de las múltiples instancias desplegadas en GCP

6.2.2 Conexión a la instancia

La conexión que vamos a utilizar será mediante el protocolo RDP de Windows. En el caso de tener una red privada con Google, le asignaremos una subred interna. Con nuestro usuario y contraseña generado, iniciaremos sesión y podremos empezar a configurar nuestro directorio activo de prueba. Ilustración 9

6.2.3 Problema de recuperación

A causa de un problema con la conexión al puerto de serie en máquinas Windows de Google Cloud, no será posible restablecer la máquina desde una copia de seguridad generada *On-premise*, por lo menos en este caso, ya que en la documentación de Google si dan soporte a este método de recuperación. Por esta razón, no va a ser posible que Microsoft pueda dar soporte a estas máquinas, que de cualquier otra manera sean restablecidas en Google Cloud.

Una de las alternativas para restablecer una máquina virtual *On-premise*, es con la herramienta proporcionada por Google llamada “Migrate Connect”, esta herramienta facilita la migración de una máquina virtual gracias a la virtualización, mediante una conexión directa con Google. De esta manera sería posible restablecer una máquina, pero como hemos comentado no da soporte Microsoft, por consiguiente, no es recomendable esta práctica.

Por esta razón, tendremos que pasar a trabajar con Azure, que proporciona los mismos recursos y el despliegue también se puede automatizar mediante Terraform.

6.2.4 Cambio de nube

Azure va a ser la nueva nube seleccionada para las pruebas, necesitaremos consultar diferentes librerías, respecto a Google Cloud, por lo tanto, el código de despliegue será diferente. Sin embargo, el proceso de despliegue y recuperación será el mismo.

Para empezar el proceso de despliegue mediante Terraform, necesitamos un *host* que tenga instalado Azure CLI y Terraform, en el caso de Google ya teníamos un entorno preparado para esto, el bastión. Pero para Azure no estaba preparado, así que, utilizaremos un ordenador de nuestro laboratorio para hacer las instalaciones precisas y empezar a desplegar.

Para comprobar que funciona bien, lanzamos “terraform init” e iniciamos sesión en Azure con “az login”. (ver ilustración 50 y 51)

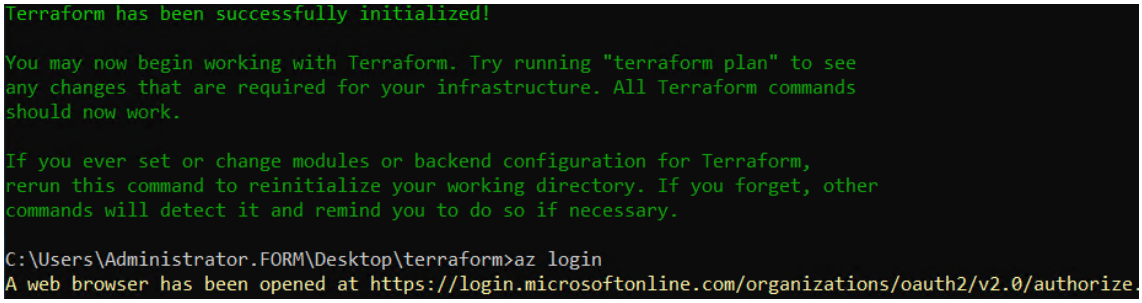

Ilustración 50, inicialización de Terraform y login en Azure

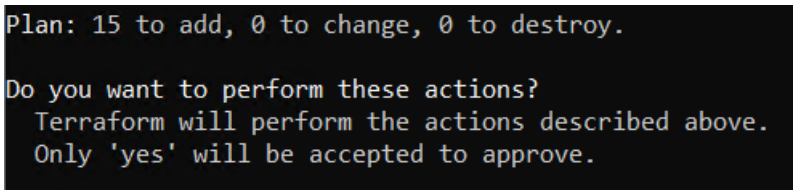

Ilustración 51, salida de creación de un plan en Terraform

Aplicamos e insertamos los recursos necesarios para nuestras maquinas. A simple vista se añaden más recursos que en Google, pero al fin y al cabo son las mismas instancias, 3. La diferencia es que en Azure las tarjetas de red, los grupos de recursos, la declaración de las nuevas redes se divide y forman más recursos en vez de comprimirlos en una sola instancia. También es verdad que, al no utilizar módulos, necesitamos declarar más recursos que en Google, que en nuestro caso ya están definidos en otros proyectos y compartidos en Google.

6.2.5 Comprobar copia de seguridad

Para empezar, había que configurar el puerto de serie en Azure, con ayuda de la documentación se pudo hacer sin imprevistos. Una vez configurado, tenemos que reiniciar el servidor en modo opciones avanzadas de inicio para poder recuperar la copia de seguridad.

Antes de recuperar desde una copia de seguridad, hay que tener en cuenta que debemos ubicar en otro disco la copia del servidor que queramos recuperar, para ello necesitamos añadir en nuestra declaración Terraform un disco de datos. Una vez declarado tenernos que conectarnos mediante RDP para transferir nuestra copia a la nube, este proceso va a tardar unos minutos ya que se transferirán varios gigas de datos.

Ahora ya tenemos la instancia en estado óptimo para la recuperación, reiniciamos la maquina y entramos en el modo avanzado, seleccionamos la opción “System Image Recovery” y seguimos los pasos de recuperación.

Una vez finalizada la instalación de la recuperación comprobamos los scripts que preparamos para restaurar el primer controlador de dominio.

6.2.6 Alternativas a la copia de seguridad

Para la plataforma que deseemos utilizar deberían aportarnos documentación relacionada a la recuperación de una máquina virtual en caso de error.

En nuestro caso, para la plataforma en la nube de Google necesitaremos utilizar una herramienta llamada “Migrate Connect” explicada en el punto 6.2.3, dónde explicamos cómo deberíamos proceder en ese caso.

Por otra parte, en Azure hay una documentación la cual explican como recuperar una máquina virtual, el inconveniente es que no será posible recuperar una maquina On-premise, es decir, deberemos tener previamente la máquina virtual desplegada en Azure para poder recuperarla. (Azure Backup., s. f.)[11]

6.3 Comparación de resultados

La base de datos del directorio activo creado *On-premise* para la simulación del entorno, tiene un peso aproximado de 20 megabytes, en comparación con la base de datos de una empresa con más de 20.000 objetos es de alrededor de 4 gigabytes, es decir, un 95% más grande que nuestra simulación.

Un mal mantenimiento y actualización del directorio puede provocar tener una base de datos desmedida, por eso, puede que el directorio activo sea más grande de lo esperado.

Teniendo en cuenta los datos obtenidos en la simulación, construiremos la tabla 1 de resultados para comparar los datos recogidos.

Tabla 1, comparación de resultados

	Automatizado	Manual
Despliegue (11 servidores)	1’	30’
Recuperación 1er DC	10’	60’
Recuperación resto DC’s	20’	60’
Total	**31’**	**150’**
Decremento	**79,33%**	

Se prevé un decremento de un 76.67% en el tiempo, al automatizar la recuperación ante desastres del Directorio Activo en nuestra simulación. Esto supone que, sin haberlo probado en un entorno realista, podamos decrementar el tiempo de recuperación y poder mitigar las pérdidas que suponen las horas de parada para una gran empresa.

7. Conclusiones

Como consecuencia de lo aquí expuesto, el objetivo del proyecto se ha podido llevar a cabo con algunos inconvenientes, pero al fin y al cabo se finalizó con éxito.

Se ha podido efectuar todas las pruebas y tomado los tiempos necesarios para poder sacar conclusiones de que el objetivo se ha alcanzado, hemos aprendido mucho y también cosas nuevas, de las cuales, algunas dadas en la universidad.

La primera parte del proyecto que estaba relacionada con la confección de scripts para el directorio activo, teníamos algunas nociones, pero no a tan gran escala. Respecto al servicio Directorio Activo, se dan pequeñas pinceladas en la asignatura Administración de sistemas de tercero en la rama de tecnologías de la información. La segunda parte del proyecto, donde desplegamos la infraestructura necesaria en la nube, esto es totalmente nuevo y he aprendido mucho, gracias a los técnicos de la empresa, me han ayudado a desarrollar el objetivo desde cero hasta llegar a la automatización del despliegue.

A grandes rasgos, el porcentaje de aprendizaje ha sido del 80%, el resto son conocimientos aprendidos en la carrera o por cuenta propia.

En cuanto a los inconvenientes encontrados, he de decir que, he podido solucionarlos sin muchas consecuencias. El primero fue en el momento de cómo y de qué manera lanzar los scripts, porque dependiendo desde donde se lancen debemos tener diferentes cosas en cuenta. Con ayuda de los compañeros de Microsoft he podido alcanzar la solución sin mayores imprevistos.

Otro de los inconvenientes surgió cuando nos planteamos recuperar nuestra copia de seguridad en la nube, fue un gran tropiezo ya que, incluso teniendo ya toda la infraestructura creada sobre Google, nos tuvimos que cambiar a Azure. En este momento tuve la oportunidad de volver a aprender, aunque a grandes rasgos, es muy parecida a Google, pero como comentamos anteriormente, necesitamos utilizar otra librería y por tanto el código es diferente. Aun teniendo estos imprevistos hemos conseguido llegar a la solución.

Otro problema para solucionar fue sobre el caso que abrimos a Microsoft intentando resolver cuantos controladores de dominio hacían falta desplegar en la nube para que la producción de la empresa siga su curso. acaba

En síntesis, el objetivo de poder recuperar el directorio activo ante un ataque de *ransomware*, y poder continuar la producción sin necesitar más de un día o incluso una mañana para la

recuperación, ha sido satisfactorio, aunque para poder probar nuestra mejora, habría sido conveniente poder implantarlo en un dominio a gran escala.

Referencias

[1] CHEN, Brian X. La tecnología que invadirá nuestras vidas en 2022. The New York Times [en línea]. 10 de enero de 2022 [consultado el 25 de marzo de 2022]. Disponible en: https://www.nytimes.com/es/2022/01/10/espanol/tecnologia-tendencias-2022.html#:~:text=Algunas%20tendencias%20de%202022%20que,para%20manipular%20juegos%20en%203D

[2] REDACCIÓN. Estas son las grandes tendencias tecnológicas de 2022, según la industria TIC española. ComputerWorld | Innovación, negocio y tecnología [en línea]. 31 de enero de 2022 [consultado el 25 de marzo de 2022]. Disponible en: https://www.computerworld.es/tendencias/estas-son-las-grandes-tendencias-tecnologicas-de-2022-segun-la-industria-tic-espanola#:~:text=El%20director%20general%20de%20Intel,el%20digital%20como%20nunca%20antes

[3] Mercadona, Conócenos. Mercadona - Supermercados de Confianza [en línea]. [sin fecha] [consultado el 4 de abril de 2022]. Disponible en: https://info.mercadona.es/es/conocenos

[4] DALBERA'S, Jacques. What is OAuth? OAuth versus Kerberos! ADFS and OAuth! Jacques Dalbera's IT world [en línea]. [sin fecha] [consultado el 4 de abril de 2022]. Disponible en: https://itworldjd.wordpress.com/2021/11/19/what-is-oauth/

[5] Mr. Hawkins, When *Ransomware* Cripples a City, Who's to Blame? This I.T. Chief Is Fighting Back | The New York Times | August, 2019 | [consultado el 20 de abril de 2022]. Disponible en: https://www.nytimes.com/2019/08/22/us/florida-ransomware-hacking-it.html

[6] Microsoft. (s. f.). Documentación de PowerShell - PowerShell. Developer tools, technical documentation and coding examples | [consultado el 18 de mayo de 2022] Disponible en: Microsoft Docs. https://docs.microsoft.com/es-es/powershell/

[7] Semperis. (2020). Recovering Active Directory from Cyber Disasters. Semperis News, 2–16. | [consultado el 12 de mayo de 2022]. Disponible en: https://www.semperis.com/resources/report-recovering-ad-from-cyber-disasters/

[8] Avi. Una introducción a Terraform para principiantes - Tutorial de Terraform. Geekflare. [en línea].16 de febrero de 2022 [consultado el 26 de abril de 2022]. Disponible en: https://geekflare.com/es/terraform-for-beginners/#:~:text=Terraform%20es%20una%20infraestructura%20de,lenguaje%20declarativo%20fácil%20de%20aprender

[9] HashiCorp. (s. f.). Terraform registry GCP/Azure. Terraform Registry. [en línea]. [sin fecha] [consultado el 5 de abril de 2022]. Disponible en: https://registry.terraform.io/namespaces/hashicorp

[10] GitLab. (2022, 3 de enero). The One DevOps Platform | GitLab. The One DevOps Platform | [consultado el 8 de abril de 2022]. GitLab. https://about.gitlab.com/

[11] Azure Backup. (s. f.). Developer tools, technical documentation and coding examples [consultado el 13 de abril de 2022] | Microsoft Docs. https://docs.microsoft.com/en-us/azure/backup/backup-azure-vms-automation#restore-an-azure-vm

Anexos

1. Scripts PowerShell

Script que será ejecutado en el primer controlador de dominio, después de haber recuperado la copia de seguridad. Se deberá ejecutar en local, conectando al ordenador por RDP.

```
######################
#Primera parte del D&R
######################

Import-Module ServerManager

#Variables
$ip = Get-NetIPAddress | Where-Object {$_.InterfaceAlias -match "Ethernet"}
$dns = (Get-DnsClientServerAddress).ServerAddresses[0]

$doms = (Get-WmiObject Win32_NetworkAdapterConfiguration).DNSDomain
$dom = Get-WMIObject Win32_ComputerSystem | Select-Object -ExpandProperty
Domain

$path1 = 'HKLM:\SYSTEM\CurrentControlSet\Services\NTDS\Parameters'
$Repl = (Get-ItemProperty -Path $path1)."Repl Perform Initial
Synchronizations"


#Cambiar DNS y apuntar a si mismo
if ($dns -ne $ip.IPv4Address) {
   Set-DnsClientServerAddress -InterfaceAlias ethernet0 -ServerAddresses
$ip.IPv4Address
}


#Reiniciamos Network Location Awareness
if ($doms -ne $dom) {
    Restart-Service -Name NlaSvc -force
       Disable-NetAdapter -Name $ip.InterfaceAlias -Confirm:$false
       Enable-NetAdapter -Name $ip.InterfaceAlias -Confirm:$false
}


#Cambiar Repl Perform Initial Synchronizations
##Si es 1 cambiar a 0
if ( $Repl -ne 0)
{
    New-ItemProperty -Path $path1 -Name 'Repl Perform Initial
Synchronizations' -Value 0 -PropertyType DWord
    #Si modifoca o crea reinicia máquina
    Write-Host "La maquina se va a reiniciar"
       Restart-Computer

} else {Write-Host "Done" -ForegroundColor Green}
#FIN PRIMERA PARTE
```

Segundo Script de ejecutaremos después de la primera parte, ya que, es posible que deba reiniciar el servidor. Esperamos que reinicie y ejecutamos en local.

```
######################
#Segunda parte del D&R
######################

#Variables
$SIDDom = (Get-ADDomain).domainsid
$SIDAdmin = $SIDDom.ToString() + "-500"
$MemSA = (Get-ADGroupMember -Identity "Schema Admins")
$MemEA = (Get-ADGroupMember -Identity "Enterprise Admins")


#Habilitamos cuenta administrador
$usr = Get-ADUser -Identity $SIDAdmin | Enable-ADAccount


#Añadimos usuario a schema admins y Enterprise Admins (sino pertenecemos
aun)
if (!$MemSA | where {$_.name -eq $usr.Name}) {
    Add-ADGroupMember -Identity "Schema Admins" -Members $usr.name
}

if (!$MemEA | where {$_.name -eq $usr.Name}) {
    Add-ADGroupMember -Identity "Enterprise Admins" -Members $usr.name
}
#FIN SEGUNDA PARTE
```

Tercera parte de la recuperación del primer controlador de domino,

```
######################
#Tercera parte del D&R
######################

#Cambiar nombre de dominio
$d = "form"
$l = "local"

#Variables
$net = @((Get-SmbShare -Name "NETLOGON").Description, (Get-SmbShare -Name
"SYSVOL").Description)
$hostn = hostname
$DomainFQDN = Get-ADDomain
$Doma = "CN=SYSVOL Subscription,CN=Domain System Volume,CN=DFSR-
LocalSettings,CN=" + $hostn + ",OU=Domain Controllers," +
$DomainFQDN.DistinguishedName
$DCs = Get-ADDomainController -Filter *
$DNSZones = (Get-DnsServerZone).ZoneName
$DC1 = $DCs | where {$_.Name -eq $(hostname)}
$path1 = 'HKLM:\SYSTEM\CurrentControlSet\Services\NTDS\Parameters'
$GC = (Get-ItemProperty -Path $path1)."Global Catalog Promotion Complete"

#creamos fichero para almacenar contraseña
New-Item ".\pass.txt"

#Generador de contraseñas
function Get-RandomPassword {
param (
[Parameter(Mandatory)]
[int] $length,
[int] $amountOfNonAlphanumeric = 1
) Add-Type -AssemblyName 'System.Web'
return [System.Web.Security.Membership]::GeneratePassword($length,
$amountOfNonAlphanumeric)}
```

```powershell
##Modificar contraseña administrador
$pass = Get-RandomPassword 20
Set-Content ".\pass.txt" "Contraseña de administrador: $($pass)"
write-host "Contraseña de administrador: $($pass)"  ####MOSTRAR POR
PANTALLA LA CONTRASEÑA GENERADA y genera un file pass.txt####
Set-ADAccountPassword -Identity ($usr.Name) -NewPassword (ConvertTo-
SecureString -AsPlainText $pass -Force) -Reset


#Comprobar recurso compartido SYSVOL y NETLOGON
if ($net.Length -eq 2) {
    Write-Host "NETLOGON Y SYSVOL correctos" -ForegroundColor Green
} else { write-error "No estan compartidos NETLOGON O SYSVOL" }

#Marcar SYSVOL como autoritativo
$as = (Get-Date).AddSeconds(-60)

Set-ADObject -Identity $Doma -Replace @{"msDFSR-Enabled" = $false }
Set-ADObject -Identity $Doma -Replace @{"msDFSR-options" = 1 }

#Comprobar evento 4114
$eventos = Get-Eventlog -LogName 'DFS Replication' | Where {$_.EventId -eq
4114}
$last = $eventos[0].TimeGenerated
if ($as -gt $last) {
      write-error "Evento DFS Replication no generado"
} else { write-host "Evento DFS Replication generado" -ForegroundColor
Green }

$as = (Get-Date).AddSeconds(-60)
#Volvemos a activar la replicacion
Set-ADObject -Identity $Doma -Replace @{"msDFSR-Enabled" = $true }

#Comprobar evento 4602
$eventos = Get-Eventlog -LogName 'DFS Replication' | Where {$_.EventId -eq
4602}
$last = $eventos[0].TimeGenerated
if ($as -gt $last) {
      write-error "Evento DFS Replication no generado"
} else { write-host "Evento DFS Replication generado" -ForegroundColor
Green }

#Limpieza de los objetos Server y metadata DNS
foreach ($DC in $DCs) {
    if ($hostn -ne $DC.Name) {

        Get-ADObject -Identity "CN="$($DC.Name)",OU=Domain
Controllers,DC=$($l.Split(".")[0]),DC=$($l.Split(".")[1])" | Remove-
ADObject -Recursive -Confirm $false
        Get-ADObject -Identity
"CN="$($DC.Name)",CN=Servers,CN="$($DC.Site)",CN=Sites,CN=Configuration,DC=
$($l.Split(".")[0]),DC=$($l.Split(".")[1])" | Remove-ADObject -Recursive -
Confirm $false
        foreach ($zone in $DNSZones) {
            Get-DnsServerResourceRecord -ZoneName $zone -RRType SRV | where
{ $_.RecordData.DomainName -match $DC.Name } | Remove-
DnsServerResourceRecord -Force -ZoneName $zone
            Get-DnsServerResourceRecord -ZoneName $zone -RRType Ns | where
{$_.RecordData.NameServer -match $DC.Name } | Remove-
DnsServerResourceRecord -Force -ZoneName $zone
            Get-DnsServerResourceRecord -ZoneName $zone -RRType A | where
{$_.RecordData.IPv4Address.IPAddressToString -match $DC.IPv4Address } |
Remove-DnsServerResourceRecord -Force -ZoneName $zone
            Get-DnsServerResourceRecord -ZoneName $zone -RRType Cname |
where {$_.RecordData.HostNameAlias -match $DC.Name } | Remove-
DnsServerResourceRecord -Force -ZoneName $zone
        }
    }
}

Restart-Service netlogon


#Forzar sobre el DC todos los roles FSMO
Move-ADDirectoryServerOperationMasterRole -Identity $hostn -
OperationMasterRole 0,1,2,3,4 -force
#Para confirmar
netdom query fsmo
```

```
#Corergir propietario ForestDNSZones y DomainDNSZones (si esta mal)
Set-ADObject -Identity
"CN=Infrastructure,DC=DomainDnsZones,$($DomainFQDN.DistinguishedName)" -
Replace @{fSMORoleOwner = $((Get-ADDomainController -Identity
($DomainFQDN).PDCEmulator).NTDSSettingsObjectDN)}

Set-ADObject -Identity
"CN=Infrastructure,DC=ForestDnsZones,$($DomainFQDN.DistinguishedName)" -
Replace @{fSMORoleOwner = $((Get-ADDomainController -Identity
($DomainFQDN).PDCEmulator).NTDSSettingsObjectDN)}


#Elevar valor del RID en 100000 y verificar
function raiseRIDPool($amount=100000){
    $domain = get-addomain
    $currentRidPool = get-adobject "CN=RID
Manager$,CN=System,$($domain.DistinguishedName)" -properties
rIDAvailablePool | select -expand rIDAvailablePool
    Write-verbose "RidPool is currently $currentRidPool will be raised to
$($currentRidPool + $amount)"
    set-adobject "CN=RID Manager$,CN=System,$($domain.DistinguishedName)" -
replace @{ridavailablePool=($currentRidPool + $amount)}
    $Domain = New-Object System.DirectoryServices.DirectoryEntry
    $DomainSid = $Domain.objectSid
    $RootDSE = New-Object
System.DirectoryServices.DirectoryEntry("LDAP://RootDSE")
    $RootDSE.UsePropertyCache = $false
    Write-Verbose "Invalidating the rid pool for the current domain
$($domain.name)"
    $RootDSE.Put("invalidateRidPool", $DomainSid.Value)
    $RootDSE.SetInfo()
}

raiseRIDPool
New-ADUser -Name "test" -Enabled $false


#Resetear la password de equipo del controlador de dominio y de la cuenta
krbtgt
$i=1
for (;$i -le 2;$i++) {
       Reset-ComputerMachinePassword -Server $hostn -Credential
($DomainFQDN).DNSRoot\Administrator
       $passkrbtgt = Get-RandomPassword 20
       Set-ADAccountPassword -Identity krbtgt -Reset -NewPassword
(ConvertTo-SecureString -AsPlainText $passkrbtgt -Force)
}
#Y añade al archivo pass la contraseña de krbtgt
Add-Content ".\pass.txt" "Contraseña de krbtgt : $($passkrbtgt)"
write-host "Contraseña de krbtgt : $($passkrbtgt)"


#Habilitar el DC como Global Catalog (Solo en caso de haberlo
deshabilitado)

if (!$DC1.IsGlobalCatalog) {
    Set-ADObject -Identity $DC1.NTDSSettingsObjectDN -Replace
@{options='1'}
}
if ( $GC -eq 0)
{
    New-ItemProperty -Path $path1 -Name 'Global Catalog Promotion Complete'
-Value 1 -PropertyType DWord
    #Si modifoca o crea reinicia máquina
    write-host "Global Catalog activado" -ForegroundColor Green
} else {write-host "Global Catalog activado" -ForegroundColor Green}
#FIN TERCERA PARTE
```

Una vez finalizado estos Scripts tendremos el primer controlador de dominio recuperado y limpio para continuar con la recuperación del resto DC's.

Script de inicio para instalar los recursos necesarios en los servidores, y prepararlos para su promoción a controladores de dominio. Este Script lo deberemos tener en la nube para que las instancias puedan ejecutarlo, por ejemplo, en GCP añadirlo a un bucket y con startup-script añadirlo.

```
#####################
# Script Despliegue #
#####################

#Para el resto de controladores de dominio
Import-Module ServerManager

#Instalar el rol de AD
Add-WindowsFeature "AD-Domain-Services" -IncludeManagementTools -Restart

#Reiniciamos para aplicar cambios
shutdown /r
#Fin Script
```

Una vez reiniciadas todas las instancias, volveremos al primer DC y ejecutamos este Script, que a su vez ejecutará el siguiente. Debemos tener bien ubicados los archivos necesarios.

```
#################
# Script Invoke #
#################

#invokecommand desde el primer dc recuperado
#utilizaremos para el usuario de dominio, uno habilitado par ala
recuperacion y un usuario local creado por GCP

#path del script recuperar.ps1 (ubicarse en el mismo sitio que este script)
$pathScript = ".\Recuperar.ps1"
#path del archivo se servidores desplegados (ubicarse en el mismo sitio que
este script)
$pathServers = ".\servers.csv"
#Nombre de la Interfaz
Get-NetIPAddress -InterfaceAlias * -AddressFamily IPv4 | Format-Table
{$_.InterfaceAlias}
$iface = read-host "Nombre de la interfaz?"

#Variables
$hostip = (Get-NetIPAddress -AddressFamily IPV4 -InterfaceAlias
$iface).IPAddress
$domain = (Get-ADDomain).DNSRoot
$credenciales = $host.ui.PromptForCredential("Credencial Dominio", "Por
favor introduce el usuario y contraseña del dominio $($domain)", "",
"NetBiosUserName")
$lcredential = $host.ui.PromptForCredential("Credencial Local", "Por favor
introduce el usuario y contraseña Local", "", "NetBiosUserName")

#Leer el fichero de servidores
$DonaADServers = Import-Csv $pathServers -Delimiter ','

foreach($server in $DonaADServers){
      Add-Computer -ComputerName $($server.NAME) -LocalCredential
$lcredential -DomainName $domain -Credential $credenciales -Restart -Force
}
```

```
#Promocionar a controlador de dominio
foreach($server in $DonaADServers){
	try {
	#Invocar recuperacion en servidores
	write-host "Servidor: $($server.NAME)"
	Invoke-Command -FilePath $pathScript -ComputerName $($server.NAME) -
ErrorAction Stop -Credential $lcredential -ArgumentList $hostip, $domain,
$credenciales
	} catch {
		Write-Host "Error: $($_.Exception.Message)" -ForegroundColor
Red
	}
}
#Fin Script invoke
```

Script colateral lanzado por el Script invoke.

```
################################
# Script promoción resto DC's #
################################

#Acciones a ejecutar en cada DC a promocionar
#Parametro de entrada del invokecommand
[cmdletbinding()]
param(
[parameter(mandatory=$false)]
[string]$hostip
[string]$domain
[System.Management.Automation.PSCredential]$credenciales
)

##Configuramos IP y DNS
Set-DnsClientServerAddress -InterfaceAlias ethernet0 -ServerAddresses
$hostip

##Promocionar DC
$SafeModeAdminPassword =  (ConvertTo-SecureString -AsPlainText
"xxxxcambiar-pass-DSRMxxxx" -Force)
Install-ADDSDomainController -NoGlobalCatalog:$false -
CreateDnsDelegation:$false -CriticalReplicationOnly:$true -DatabasePath
"C:\windows\NTDS" -DomainName $domain -InstallDns:$true -LogPath
"C:\Windows\NTDS" -NoRebootOnCompletion:$false -SiteName "SiteCentral" -
SYSVOLPath "C:\Windows\SYSVOL" -Force:$true -SafeModeAdministratorPassword
$SafeModeAdminPassword -Credential $credenciales

##Comprobar que es GC, sino activar
if (!(Get-ADDomainController).IsGlobalCatalog) {
    Set-ADObject -Identity (Get-ADDomainController).ntdssettingsobjectdn -
Replace@{options='1'}
}
#Fin Script promoción
```

2. Scripts Bash

Estos scripts nos servirán para exportar los nombres de las instancias desplegadas en la nube, de GCP o de Azure, para que posteriormente las utilicemos en Script invoke.

```
########################
# Script nombres Azure #
########################

#!/bin/bash
# Script to retrieve compute engine names.

list=$(az vm list --resource-group Win-AD --output table --query "[].{Name:name}")

printf '%s\n' $list >> servers.csv
```

```
######################
# Script nombres GCP #
######################

#!/bin/bash
# Script to retrieve compute engine names.

list=$(gcloud compute instances list --format="table(name)" --filter='name~lvw')

printf '%s\n' $list >> servers.csv
```

3. Scripts Terraform

3.1. Google Cloud Platform

Para el despliegue de máquinas virtuales en la plataforma nube GCP seguiremos este código Terraform para su descripción. Lo dividiremos en varios archivos.

```
####################
#archivo versions.tf#
####################
terraform {
  required_version = ">= 0.13"
  required_providers {
      google = {
          source = "hashicorp/google"
      }
      google-beta = {
          source = "hashicorp/google-beta"
      }}}

terraform {
  backend "gcs" {
    bucket      = "bucket "
    prefix      = "terraform"
//    credentials = "../../mount/credentials.json"
  }}
```

```
#################
#archivo vars.tf#
#################

variable "environment_acronym" {
  type        = string
  description = "Acronimo del entorno de desarrollo para crear nombres compuestos"
}

variable "project_id" {
  type        = string
  description = "Project ID of project"
}

variable "netops_project_id" {
  type        = string
  description = "Identificador del proyecto de Netops"
}

variable "network-name" {
  type        = string
  description = "Network del proyecto"
}

variable "subnetwork" {
  type        = string
  description = "Subnetwork del proyecto"
}

variable "subnetwork_project" {
  type        = string
  description = "Subnetwork del proyecto"
}

variable "region" {
  type        = string
  description = "region del proyecto"
}

variable "zone" {
  type        = string
  description = "zona del proyecto"
}

variable "instance_count" {
  type        = number
  default     = "1"
}
```

```
##########################
#archivo terraform.tfvars#
##########################

#Var instances
environment_acronym = "acrónimo-proyecto"
project_id          = "nombre-proyecto"
netops_project_id   = "nombre-proyecto-compartido"
network-name        = "network-vpc"
subnetwork          = "network-subnet"
subnetwork_project  = "nombre-proyecto-compartido"
region              = "europe-west1"
zone                = "europe-west1-b"
```

```
################################################
#DESPLIEGUE DEL RESTO DE CONTROLADORES DE DOMINIO#
################################################

#Creacion de maquina windows server modulo
module "vm-windows-resto-dc" {
  count               = 10
  source              = "git@gitlab.url.module"
  network             = ""
  project_id          = var.project_id
  subnetwork          = var.subnetwork
  subnetwork_project  = var.subnetwork_project
  hostname            = "lvw-gcpad-r${count.index + 1}"
  network_tags        = ["tag-network"]
  service_account     = ""
  metadata            = [
    {key = "windows-startup-script-url", value = "gs://bucket/despliegue.ps1"},
    {key = "enable-oslogin", value = "TRUE"}
  ]
}
############################################
#DESPLIEGUE DEL PRIMER CONTROLADOR DE DOMINIO
############################################

#Creacion de maquina windows server modulo
module "vm-windows-main" {
  count               = var.instance_count
  source              = "git@gitlab.url.module"
  network             = ""
  project_id          = var.project_id
  subnetwork          = var.subnetwork
  subnetwork_project  = var.subnetwork_project
  hostname            = "lvw-gcpad-main"
  network_tags        = ["tag-network"]
  service_account     = ""
  metadata            = [
    {key = "windows-startup-script-url", value = "gs://bucket/D&R_1.ps1"},
    {key = "enable-oslogin", value = "TRUE"}
  ]
}
```

3.2. Microsoft Azure

Este va a ser nuestra disposición de archivos para el despliegue en Azure.

```
####################
#archivo az_data.tf#
####################

#Nombre del grupo de recursos
resource "azurerm_resource_group" "AD" {
  name     = "Win-AD"
  location = "Location"
}

#Definimos las redes para el proyecto Azure
resource "azurerm_virtual_network" "AD" {
  name                = "rg-network"
  address_space       = ["10.0.0.0/16"]
  location            = azurerm_resource_group.AD.location
  resource_group_name = azurerm_resource_group.AD.name
}

resource "azurerm_subnet" "AD" {
  name                 = "internal"
  resource_group_name  = azurerm_resource_group.AD.name
  virtual_network_name = azurerm_virtual_network.AD.name
  address_prefixes     = ["10.0.2.0/24"]
}

# Create public IPs
resource "azurerm_public_ip" "myterraformpublicip" {
  name                = "myPublicIP"
  location            = azurerm_resource_group.AD.location
  resource_group_name = azurerm_resource_group.AD.name
  allocation_method   = "Dynamic"
}

# Create Network Security Group and rule
resource "azurerm_network_security_group" "myterraformnsg" {
  name                = "myNetworkSecurityGroup"
  location            = azurerm_resource_group.AD.location
  resource_group_name = azurerm_resource_group.AD.name

  security_rule {
    name                       = "RDP"
    priority                   = 1001
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "3389"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }}

resource "azurerm_storage_account" "storage-account" {
  name                     = "antacer"
  resource_group_name      = azurerm_resource_group.AD.name
  location                 = azurerm_resource_group.AD.location
  account_tier             = "Standard"
  account_replication_type = "LRS"
}
```

```
####################
#archivo az_main.tf#
####################

#Interfaz de red para VM
resource "azurerm_network_interface" "AD" {
  name                = "AD-nic"
  location            = azurerm_resource_group.AD.location
  resource_group_name = azurerm_resource_group.AD.name

  ip_configuration {
    name                          = "internal"
    primary                       = "true"
    subnet_id                     = azurerm_subnet.AD.id
       public_ip_address_id          = azurerm_public_ip.myterraformpublicip.id
    private_ip_address_allocation = "Dynamic"
  }
}

#Configuracion de VM
resource "azurerm_virtual_machine" "AD" {
  name                  = "VM-main-AD"
  resource_group_name   = azurerm_resource_group.AD.name
  location              = azurerm_resource_group.AD.location
  vm_size               = "Standard_B1s"
  network_interface_ids = [
    azurerm_network_interface.AD.id,
  ]

  delete_os_disk_on_termination = true
  delete_data_disks_on_termination = true

  storage_os_disk {
    name              = "myosdisk"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
   }
  os_profile {
    computer_name  = "main-AD"
    admin_username = "antacer"
    admin_password = "xxxxxxxxx"
  }

  boot_diagnostics {
    enabled     = true
    storage_uri = "https://url-cuenta-almacenamiento/"
  }

  storage_image_reference {
    publisher = "MicrosoftWindowsServer"
    offer     = "WindowsServer"
    sku       = "2019-Datacenter"
    version   = "latest"
  }

  os_profile_windows_config {
  }
}

# Connect the security group to the network interface
resource "azurerm_network_interface_security_group_association" "AD-link" {
  network_interface_id      = azurerm_network_interface.AD.id
  network_security_group_id = azurerm_network_security_group.myterraformnsg.id
}
```

```
#####################
#archivo az_resto.tf#
#####################

#Interfaz de red para VM
resource "azurerm_network_interface" "AD-r" {
  name                = "AD-nic-${count.index + 1}"
  location            = azurerm_resource_group.AD.location
  resource_group_name = azurerm_resource_group.AD.name

  ip_configuration {
    name                          = "internal"
    primary                       = "true"
    subnet_id                     = azurerm_subnet.AD.id
  }
}

#Configuracion de VM
resource "azurerm_virtual_machine" "AD-r" {
  count                 = 10
  name                  = "VM-resto-AD-${count.index + 1}"
  resource_group_name   = azurerm_resource_group.AD.name
  location              = azurerm_resource_group.AD.location
  vm_size               = "Standard_B1s"
  network_interface_ids = [
    element(azurerm_network_interface.AD-r.*.id, count.index + 1),
  ]

  delete_os_disk_on_termination = true
  delete_data_disks_on_termination = true

  storage_os_disk {
    name              = "myosdisk"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
   }
  os_profile {
    computer_name  = "resto-AD-${count.index + 1}"
    admin_username = "antacer"
    admin_password = "xxxxxxxxx"
  }

  boot_diagnostics {
    enabled     = true
    storage_uri = "https://url-cuenta-almacenamiento/"
  }

  storage_image_reference {
    publisher = "MicrosoftWindowsServer"
    offer     = "WindowsServer"
    sku       = "2019-Datacenter"
    version   = "latest"
  }

  os_profile_windows_config {
  }
}
```

Listado de abreviaturas, siglas y acrónimos

5G	Quinta generación de tecnología móvil
AD	Active Directory
A, SRV, CNAME, NS	Tipos de registro de datos DNS
BBDD	Base de datos
CMD	Command Prompt
CPD	Centro de procesamiento de datos
DC	Controlador de dominio
DN	“Distinguished Names”
DNS	Sistema de nombres de dominio
DSRM	“Directory Services Restore Mode”
DWORD	“double word” unidad de datos
FSMO	“Flexible Single Master Operations” Roles del AD
GCP	Google Cloud Platform
GUID	Identificador único global
ID	Identificador
IoT	“Internet of Things”
IP	Dirección IP, etiqueta numérica de identificación de la red
ISO	Archivo que almacena la imagen de un sistema de archivos
NETLOGON	Protocolo de autenticación de Windows Server
NTDS	BBDD que almacena archivos del AD
PDC	Primary Domain Controller Emulator
RDP	Remote desktop protocol
RID	Relative ID operations master role
SSH	Secure Shell protocol
SYSVOL	Almacena archivos públicos del dominio
VPC	Nube privada virtual
VPN	Red privada virtual

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

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

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

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

De los objetivos anteriormente mencionados, podemos relacionar algunos de ellos con nuestro proyecto:

- **Energía asequible y no contaminante**, como uno de los objetivos del trabajo es desplegar una infraestructura en un entorno seguro, la alternativa es la nube y en concreto Google Cloud Platform, de los cuales, la mayoría de sus centros repartidos por Europa tienen un certificado de emisiones bajas de carbono, es decir se suministran de centrales eléctricas con bajas emisiones.

- **Trabajo decente y crecimiento económico**, los procedimientos llevados a cabo en el proyecto incentivan el crecimiento y el buen trabajo, la automatización en uno de los fuertes en este caso.

- **Industria, innovación e infraestructuras**, el hecho de querer desplegar una infraestructura en un entorno nuevo como la nube es un avance de innovación por parte de las empresas, que cada vez utilizan estas herramientas que incentivan el crecimiento de las nuevas tecnologías y el desarrollo de nuevos métodos en el día a día de un empresa.