IMMUNE
TECHNOLOGY INSTITUTE

Solución de Recuperación ante Desastres Multi-plataforma

Daniel García | José Carbonell

>ÍNDICE_

1. DESCRIPCIÓN DEL PROYECTO

DESCRIPCIÓN

El proyecto se centra en el diseño, implementación y testeo de una arquitectura de tipo DR (Disaster Recovery) en modo activo-pasivo para la infraestructura crítica de una empresa.

Esta arquitectura busca garantizar la continuidad del negocio y la protección de los datos como recurso crítico ante eventos adversos y no previstos, como desastres naturales, fallos tecnológicos o errores humanos que puedan poner en peligro la integridad y persistencia de los mismos.

La estrategia se basa en la duplicidad de la infraestructura en dos entornos cloud: Azure y GCP. Azure se ha designado como el entorno activo, mientras que GCP actúa como el entorno pasivo de respaldo.

Se ha elaborado un protocolo de recuperación y actuación que aborda todos los aspectos necesarios tanto para garantizar la preservación de los datos como la restauración de datos y servicios de manera rápida y eficiente en caso de desastre. Este protocolo incluye la replicación continua de datos críticos entre los entornos activo y pasivo, la monitorización constante de la salud de los servicios, la automatización de los procesos de conmutación por error y la realización periódica de pruebas de recuperación ante desastres para validar la efectividad del protocolo.

Microsoft Azure

Google Cloud

1. DESCRIPCIÓN DEL PROYECTO

MOTIVACIÓN

Poner en valor la potencialidad de los entornos cloud para garantizar la continuidad del negocio y la protección de los datos críticos ante eventos adversos no previstos, minimizando el tiempo de inactividad y la pérdida de datos.

Todo asegurando la integridad y disponibilidad de los datos y recursos y aspirando a fortalecer la resiliencia empresarial y a fortalecer la protección continua de los activos críticos de la empresa en todo momento.

OBJETIVOS

> OBJETIVO 1:
Diseñar una arquitectura DR activo-pasivo entre Azure y GCP escalable y resistente a fallos.

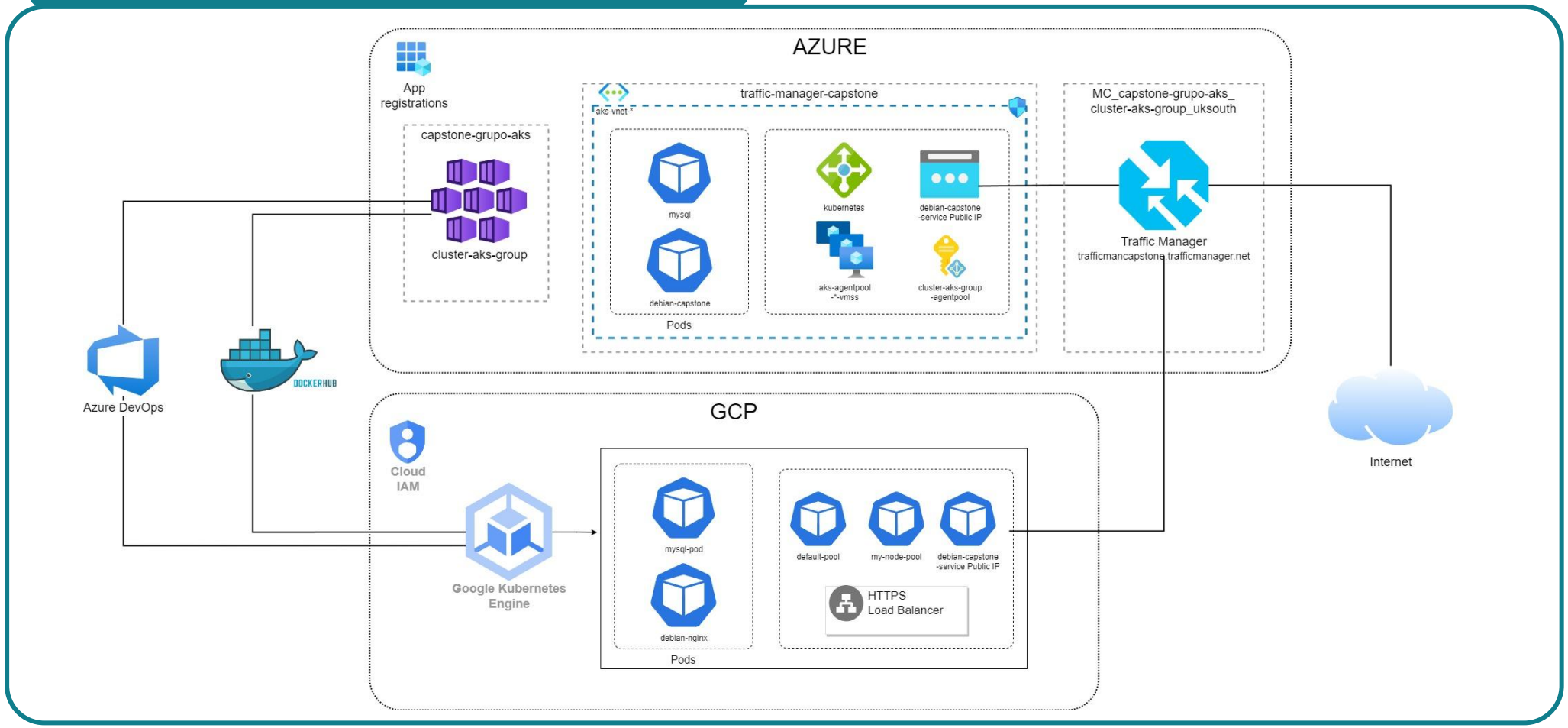
> OBJETIVO 2:
Tiempo de Recuperación (**RTO**) de **30 minutos** para infraestructura y recursos críticos del negocio.

> OBJETIVO 3:
Punto de Recuperación (**RPO**) de **30 minutos** para la base de datos del negocio.

> OBJETIVO 4:
Diseñar una estrategia de despliegue y gestión automatizada.

2. ARQUITECTURA Y SEGURIDAD

DISEÑO DE LA ARQUITECTURA



2. ARQUITECTURA Y SEGURIDAD

SEGURIDAD

Dado que el proyecto se basa en el diseño de una arquitectura con estrategia de recuperación ante desastres para infraestructura crítica, las medidas de seguridad son uno de los factores clave.

Dado el corto plazo de tiempo para la ejecución del proyecto, se han implementado unas medidas de seguridad iniciales que, en una segunda fase de despliegue del proyecto, se deben complementar con otras que se especificarán en el apartado final de la presentación.

Medidas implementadas:

- Respecto a la **seguridad del dato**, se cuenta con una estrategia de backup de la base de datos a través de automatización por pipeline, teniendo Azure DevOps programada la realización de copias de seguridad cumpliendo con los objetivos RTO y RPO marcados.
- Finalmente se dispone de un **protocolo de acción** que relaciona los procesos para la recuperación rápida y eficiente del servicio en caso de interrupción o fallo.

2. ARQUITECTURA Y SEGURIDAD

HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

Para desplegar la arquitectura del proyecto se han usado tanto herramientas y tecnologías para su diseño, despliegue y gestión como diferentes tipos de recursos para su funcionalidad.

Herramientas y Tecnologías:

- **Draw.io** para diseño de la arquitectura.
- **Visual Studio Code** para trabajar con código.
- **Azure DevOps** para gestión del ciclo de vida del proyecto, automatización y despliegue de pipelines.
- **Terraform** para trabajar con Infraestructura como Código (IaC).
- **Docker y Docker Hub** para la creación de la configuración del cluster.
- **Kubernetes** para contenerización.
- **Azure Cloud** para despliegue de infraestructura Active.
- **Google Cloud Platform** para despliegue de infraestructura Passive.

- **Azure**:
  - Resource Group
  - Azure Kubernetes Service (AKS)
  - Network Security Group
  - App Registrations
  - Traffic Manager

- **Google Cloud Platform**:
  - Kubernetes Engine
  - Storage Bucket

3. PROTOCOLO DE RECUPERACIÓN Y ACTUACIÓN ANTE DESASTRES Y SITUACIONES DE EMERGENCIA

ESCENARIOS:

- **Desastres naturales**: fallo de suministro en distintas regiones del proveedor cloud.
- **Desastres técnicos:** fallo en el proveedor cloud (errores de configuración, ataques DDoS, interrupción del servicio, fallo en múltiples regiones), fallo en el proveedor de servicios de red.
- **Desastres humanos:** eliminación accidental o corrupción de datos, actualizaciones defectuosas.

PLAN DE RECUPERACIÓN:

- **Healthchecks automatizados** en entorno Azure DevOps a través de Continuous Integration > Scheduled según objetivos
- **Automated mysql database backups**
- **Activación de recursos de respaldo en el entorno pasivo**
- **Restauración de datos**
- **Conmutación de acceso público a la web en el entorno de respaldo**

Realización periódica de pruebas de recuperación ante desastres para validar la efectividad del plan y garantizar su ejecución de manera adecuada en caso de emergencia.

Métricas de rendimiento y criterios de éxito para evaluar e identificar áreas de mejora.

Un factor importante es la **formación y concienciación del personal** sobre la importancia del plan para poder dar la respuesta necesaria y de manera efectiva en situaciones de crisis.

4. RESULTADOS Y GESTIÓN

GESTIÓN DEL PROYECTO

1. Fase de planteamiento y distribución de tareas.
2. Despliegue manual de infraestructura para pruebas.
3. Despliegue de la infraestructura final de forma automatizada en ambos clouds.
4. Desarrollo de pipelines de despliegue, configuración y backup para cumplir el protocolo establecido.
5. Testeo y comprobación de resultados.

DESAFÍOS Y SOLUCIONES

El principal **desafío** de este proyecto ha sido desarrollar una metodología autónoma y automatizada. La práctica habitual de recuperación ante desastres suele hacerse a través de recursos nativos de los diferentes proveedores cloud, existiendo además herramientas no nativas de migración entre clouds.

La **solución** implementada para este proyecto ha sido utilizar un sistema 100% controlado y automatizado basado en el uso de tecnología IaC y pipeline en Azure DevOps para realizar tanto el despliegue de la infraestructura, su configuración y la automatización de los backups y su transferencia de Azure a GCP, en este proyecto concretamente.

4. RESULTADOS Y GESTIÓN

RESULTADOS Y BENEFICIOS

El principal **resultado** del proyecto ha sido conseguir transferir, de forma automática y periódica según lo establecido en el plan de recuperación, backups de la base de datos de Azure a GCP, minimizando el coste de infraestructura en Google Cloud Platform con simplemente un bucket de almacenamiento.

Los **beneficios** más interesantes del proyecto son:

- Automatización de todo el proceso a través de pipelines.
- Bajo coste económico para la implementación del proyecto, asumiendo los costes en cada momento solo de la infraestructura en funcionamiento, dado que en el cloud Passive solo se mantiene desplegado un bucket de almacenamiento de backups.
- Escalabilidad futura del mismo, requiriendo llevar a cabo nuevas fases de desarrollo en materia de seguridad.

ESCALABILIDAD Y MANTENIMIENTO

La infraestructura desplegada en Azure opera sobre un clúster de Kubernetes, que por sí mismo ya permite escalabilidad y mantenimiento autogestionado.
Además, cuenta con ciertos servicios de monitoreo que facilitan el control sobre posibles fallos, tanto presentes como futuros. Ante un posible fallo o cualquier otra circunstancia que dejara fuera de servicio la infraestructura en Azure, la infraestructura que se despliega en GCP, basada de la misma forma en un cluster de Kubernetes, cuenta con las mismas características y beneficios que la original.

5. FUNCIONALIDAD

La funcionalidad del proyecto está diseñada para ser gestionada íntegramente desde Azure DevOps, si bien es cierto que por cuestión de tiempo no se ha podido finalizar la automatización de configuración del cluster GKE como sí se ha realizado del cluster AKS.

Así, el funcionamiento es:

1. Lanzamiento del **pipeline Deploy Azure Terraform**, que despliega la infraestructura de Azure según arquitectura mostrada.
2. Autoejecución del **pipeline Config Cluster AKS**, al estar configurado con Build Completion respecto al pipeline anterior. Este pipeline realiza lo siguiente:
    - abre los puertos del NSG creado automáticamente por AKS tras su despliegue;
    - despliega el contenido del cluster:
        - deployment con servidor Debian en el que corre un servidor Nginx el cual tiene un index.html a modo de prueba que muestra una tabla de productos de la base de datos del cluster. Incluye un servicio Load Balancer con IP pública.
        - pod con una base de datos mysql que contiene la tabla que muestra la web.
    - creación de la base de datos, la tabla y los productos iniciales a modo de ejemplo
    - creación de un endpoint en el Traffic Manager con la IP Pública creada en el despliegue del servidor.

5. FUNCIONALIDAD

Nota: Como el pipeline segundo cuenta con ejecución automática si el pipeline de despliegue finaliza correcto, aunque sea destroy, se ha creado un stage inicial para verificación de existencia de la infraestructura, en caso de que no exista se cancela y se corta el proceso de despliegue.

3. Autoejecución del **pipeline Backup db AKS,** , al estar configurado con Build Completion respecto al pipeline anterior y contar con programación de ejecución en Continuous Integration > Scheduled según objetivos.
   Este pipeline tiene como función realizar un backup de la base de datos contenerizada y mover el archivo, a través del agende pipeline, a un bucket creado de manera permanente en GCP.
4. **Pipeline de Healthcheck y despliegue de GCP**, que tiene la función de realizar revisión del estado de salud de la infraestructura de Azure. En caso de fallo o caída del servicio en Azure, el pipeline lo detecta y automáticamente despliega la infraestructura de GCP sobre la cual cargar el backup.

5. *NOTA: El pipeline anterior incluye dos stages " ConfigClusterGKE" y "CargaBackupMysql" para el **despliegue del cluster GKE**. Este proceso el objetivo es que esté automatizado a través de un pipeline (que no se ha podido finalizar en tiempo y forma para estar funcional e implementado en el proyecto) cuya funcionalidad es automatizar, al igual que en Azure, la creación del deployment y pod, la carga del archivo backup en la base de datos y la creación en el Traffic Manager de un endpoint con la IP del servidor Debian para que vuelva a estar operativa la web. Sin haber puesto en marcha el pipeline este proceso se realiza manualmente de manera exitosa.*

6. EVOLUCIÓN Y MEJORA DEL PROYECTO

El desarrollo y resultado obtenido con el proyecto se considera muy positivo, tanto por la curva de aprendizaje como por la funcionalidad obtenida.

Sin embargo, el equipo tiene identificadas y claras cuáles serían las acciones de mejora del mismo que se deberían implementar, siendo éstas las siguiente:

MEDIDAS A IMPLEMENTAR

- **Securización de datos sensibles** mediante secrets, como de datos de acceso a la base de datos.
- Uso de **deployment para la base de datos** en lugar de solo pod.
- A nivel **seguridad en cuanto a la gestión del tráfico y el enrutado** mediante:
  - Conexión privada entre Azure DevOps y el cluster, mediante Private EndPoint o VPN, por ejemplo.
  - Habilitación de enrutado a los clusters mediante Application Gateway en Ingress en Azure y un HTTPS Load Balancer en GCP que aportan balanceo y filtrado de seguridad en la conexión con el servidor web.
- **Encriptación de los backups de la base de datos** en origen y desencriptación en destino.
- **Automatización del despliegue y configuración del cluster GKE**, al igual que se realiza del cluster AKS. Para ello se dispone del pipeline de despliegue en AKS, pero se debe completar con los requerimientos de instalación de paquetes y complementos necesarios (SDK, Google Cli, Kubectl y Kubectl-Oidc) para que el agente pueda interactuar con el cluster de GCP.

IMMUNE
TECHNOLOGY INSTITUTE