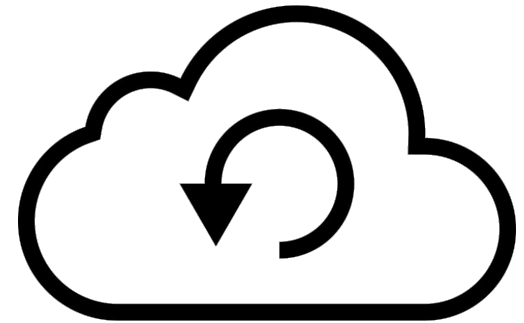


Recalculando los backups en tiempos de nube



Filosofemos



Algunas definiciones

*“A copy of information held on a computer that is **stored separately** from the computer”*

*“A copy of computer data taken and **stored elsewhere** so that it may be used to restore the original after a data loss event”*

*“Create a copy of data that can be recovered in the event of a primary data failure. Backup copies allow data **to be restored from an earlier point in time** to help the business recover from an unplanned event”*

Entonces...

- La idea es **copiar datos en un lugar diferente**.
- Para asegurar que es posible lograr un restore en caso de un problema de hardware, o alguna **catástrofe**.
- Las copias deben permitir restauraciones **a un tiempo determinado**.
- No siempre queremos restaurar por un desastre.
 - *Podríamos querer crear/inicializar un ambiente con datos de otro.*

Backups o replicación de datos

- **Backups:** focaliza en recuperar datos a un tiempo determinado.
- **Replicación:** focaliza en dar continuidad ante fallas, sin necesidad de un restore. Esto significa mantener sincronizados los datos de cada réplica y ante la falla de un servicio principal, poder reemplazarlo por la réplica.

¿Por qué la replicación no alcanza?

Porque si por error se ejecuta una acción indebida (por ejemplo un truncate table), la acción se replicará. No hay forma de volver atrás con la replicación.

Tipos de backup

Qué tipos de backups conocemos

- **Full:** todos los datos se copian.
- **Diferencial:** cada backup copia los datos modificados desde el último full backup.
- **Incremental:** cada backup copia los datos modificados desde el último backup.

TYPES OF BACKUP: FULL, DIFFERENTIAL, AND INCREMENTAL

Full Backups: Entire data set, regardless of any previous backups or circumstances.



Differential Backups: Additions and alterations since the most recent full backup.



Incremental Backups: Additions and alterations since the most recent incremental backup.



Initial Full Backup



1st Backup

2nd Backup

3rd Backup

4th Backup

5th Backup



Data subject to backup

Políticas

Políticas claras y realistas

- Los backups no serán *infinitos*.
- Las políticas deben ser claras para cada dato:
 - ¿Qué retención debe tener un dato? un día, una semana, un año, 30 años.
 - ¿Es necesario recuperar un dato de hace un año, cinco meses, tres días y dos horas? En general se requiere esta precisión para un período corto respecto del hoy.
 - ¿Qué tolerancia de tiempo puedo aceptar para un restore? Pierdo datos y quiero un restore de ayer a la madrugada o de hace 5 minutos.
- Si podemos responder las preguntas anteriores, podemos definir una política clara.
- Una base de datos, suele mantener la historia en la propia base.
 - ¿Es necesario almacenar una copia de hace 10 años?

**¿Qué podemos
afirmar?**

Afirmaciones

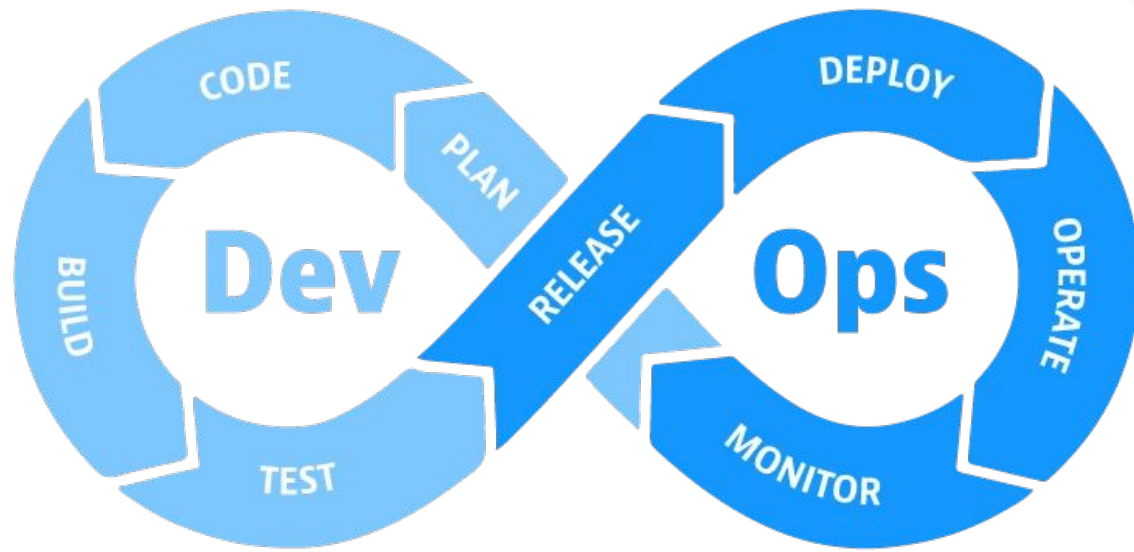
- Necesitamos storage considerando un uso **creciente**.
- Se necesita **mover a un lugar diferente** del lugar de origen.
- Soluciones al espacio se dan con nuevas estrategias:
 - Compresión
 - Deduplicación
- Estas estrategias requieren discos de acceso directo, pero además:
 - Son **más veloces** que medios secuenciales.
 - **No son secuenciales** como las cintas.
 - **No se degradan fácilmente** como cintas o medios ópticos.
 - Es posible **extender su espacio y obtener redundancia** con técnicas de RAID.

Virtualización

Vicios de virtualizar

Se hacen snapshots como backups:

- Un snapshot es un backup de **todo el sistema**.
- Si bien el snapshot en general optimiza el uso de disco, es necesario **moverlo** del mismo hypervisor.
- No hay una política clara: ¿snapshots diarios son suficientes?
- Es una solución de backup que considera **todo un sistema** en vez del dato que necesita resguardarse.



¿Qué tiene que ver con backups?

- Considerando que estemos automatizando ambientes utilizando IaC, podemos afirmar que se **recrean servidores en poco tiempo**.
 - Phoenix Servers vs Snowflake Servers.
- Si además, nuestros desarrollos aplican 12 factores.
- **La conclusión** es que nuestros servidores **solamente deben backupear menos información**:
 - *Los datos de las aplicaciones que hostean.*
 - *El estado de ellas, que según los 12 factores están en bases de datos o filesystem compartidos.*

Contenedores



Contenedores y devops

- Al igual que con IaC obtenemos una infraestructura replicable, pero además **inmutable**.
- El **estado** de un contenedor estará en sus **volúmenes**.
- Los volúmenes, al igual que se mencionó para DevOps corresponden a:
 - *Filesystem compartido*
 - *Bases de datos*



Backups en esta era

Tipos de datos a backupear

- Bases de datos:
 - Tienen la particularidad de modificarse constantemente.
 - **¿Alcanza con un backup diario?**
 - **Pueden ser relacionales:** MySQL, MariaDB, Postgres, Oracle, SQLServer, etc.
 - **Pueden ser NoSQL:** redis, Elastic, MongoDB, etc
- Archivos en Filesystem:
 - Uploads de aplicaciones
 - Archivos que generan aplicaciones

Bases de datos

- Los proveedores de cloud ofrecen servicios de **DB as a Service** que ya proveen backups con posibilidad de restaurar a un momento en el tiempo:
 - **AWS RDS:** Mysql, Postgres, SQLServer, Oracle
 - **AWS DynamoDB:** noSQL DB de AWS
 - **Mongo Atlas:** MongoDB as a Service
 - **Google SQL:** Mysql, Postgres, SQLServer
- ¿Y si no puedo usar la nube?
- ¿Y si en la nube no puedo usar una base de datos como servicio?

Bases de datos autogestionadas

- Operadores de Kubernetes simplifican la gestión:
 - [Oracle DB Operator](#)
 - [Oracle El Carro Operator](#)
 - [Postgres Operator](#)
 - [Mysql Operator](#)
- La mayoría de los operadores, no solamente **ofrecen la gestión de clusters de nodos principales y de lectura, sino además backups.**
- Si no se utilizan operadores de kubernetes, entonces se recomienda crear un cluster de base de datos **manualmente** y **contratar al menos un DBA.**

¿Y dónde se almacenan los backups de las DBs?

- En un **Object Store**:
 - AWS S3
 - GCP Cloud Storage
 - On Prem:
 - Minio
 - Ceph
- Es un **servicio económico**. Y las bases de datos lo utilizan para ir guardando réplicas de sus binary logs, redo logs o wal.
- Garantizan la replicación de datos en diferentes datacenters del mundo.
 - *O en diferentes servidores ubicados en diferentes racks, por ej.*
- **Son seguros**: proveen encriptación de buckets.

¿Y los dumps?

- Todo motor, además de proveer mecanismos de Point-In-Time recovery, permite realizar un dump.
- Un dump tiene las siguientes desventajas:
 - Requieren lockear tablas
 - Ocupan más espacio
 - Representan un full backup
 - Restores lentos
- Un dump, puede comprimirse y considerarse un archivo a backupear en un filesystem, como se verá más adelante.

¿Y los archivos?

- Si trabajamos con DevOps o contenedores, podemos usar volúmenes para estos datos.
 - O directorios aislados para los datos. Preferiblemente discos
- Si se utilizan discos para los datos, podemos mantener el uso de snapshots de los mismos.
- **Servicios del proveedor de cloud:**
 - AWS backup
 - GCP Backup and disaster recovery
- **Algunas herramientas:**
 - [restic](#) (deduplica)
 - [rsnapshot](#) (utiliza links duros)
 - [VMWare velero](#) (utiliza snapshots si se soportan o restic)
 - [Bacula](#) (hoy soporta cloud backup storage)

Conclusiones

Conclusiones

- Aplicar devops, usar contenedores e implementar los 12 factores, **minimiza la cantidad de información a resguardar.**
 - *Otra información es fácilmente regenerable.*
- Usar DB (sql o noSQL) como servicio u operadores de K8S **simplifica la gestión**, replicación de datos y promoción de nodos principales automáticamente. Además ofrecen **backups y recuperación a un punto en el tiempo.**
- Consideraciones de **ancho de banda si estamos on-Prem** y usamos backups en la nube.

Y lo más importante...

No importa la estrategia, herramienta, si aplicamos devops, contenedores o un servicio de cloud...

Un backup se considera válido si su restore es válido.

*Por lo tanto, si podemos agregar a nuestra solución de backups, **métricas de observabilidad**, y **pruebas automatizadas de restore**, vamos a poder garantizar **backups válidos**.*

¿Preguntas?