Dockerfile Avanzado

Arturo Silvelo

Try New Roads

¿Qué es un backup?

Un backup (o copia de seguridad) es una copia de los datos originales que se realiza para poder recuperarlos en caso de pérdida, corrupción, borrado accidental o desastre. En el contexto de Docker, los backups suelen centrarse en los volúmenes y bases de datos, ya que los contenedores son efímeros y fácilmente reemplazables. Un buen sistema de backups permite restaurar la información y minimizar el impacto de cualquier incidente.

1. Backups incrementales, diferenciales y completos

- **Un backup completo** copia todos los datos cada vez.
- **Un backup incremental** solo copia los cambios desde el último backup (completo o incremental).
- **Un backup diferencial** copia los cambios desde el último backup completo.

Ventajas:

- **Ahorro de espacio y tiempo:** Los backups incrementales y diferenciales solo copian los archivos que han cambiado desde el último backup relevante. Esto reduce el espacio ocupado y acelera el proceso, ya que no se duplican datos innecesarios.
- **Restauración a diferentes puntos en el tiempo:** Al tener una secuencia de backups (completo + incrementales o diferenciales), puedes restaurar el sistema no solo al último estado, sino también a cualquier punto intermedio para el que tengas un backup. Esto es útil si necesitas recuperar datos de un día específico antes de un error o corrupción.

Herramientas:

- `rsync`: ideal para backups incrementales, ya que solo copia los archivos nuevos o modificados desde el último backup. Muy útil para sincronizar directorios de forma eficiente.
- `restic`, `borg`, `duplicity`: soluciones avanzadas que permiten realizar backups completos, incrementales y diferenciales, con deduplicación, cifrado y gestión de versiones. Son recomendadas para estrategias de backup más complejas y seguras.

Consideraciones:

- Los backups incrementales requieren todos los backups anteriores para una restauración completa.
- Los diferenciales requieren solo el último completo y el último diferencial.
- Es importante planificar la rotación y limpieza de backups antiguos.

2. Backups consistentes de bases de datos

¿Por qué no es suficiente copiar archivos de bases de datos en caliente?

- Las bases de datos suelen mantener archivos abiertos y estructuras en memoria.
- Copiar los archivos directamente mientras la base está en uso puede provocar corrupción o backups incompletos.
- Es fundamental garantizar la consistencia de los datos en el momento del backup.

Uso de herramientas nativas:

- Cada motor de base de datos tiene su propia herramienta para realizar backups consistentes:
  - `mysqldump` para MySQL/MariaDB
  - `pg_dump` para PostgreSQL
  - `mongodump` para MongoDB
  - Otras: `sqlite3 .dump`, `redis-cli --rdb`, etc.

3. Automatización de backups

Ventajas de automatizar backups

- **Evita errores y olvidos humanos:** Los backups se realizan siempre, sin depender de la intervención manual.
- **Asegura la recuperación ante fallos:** Permite restaurar datos fácilmente en caso de pérdida o corrupción.
- **Facilita el cumplimiento de políticas:** Ayuda a cumplir normativas y auditorías sobre retención y protección de datos.
- **Optimiza el tiempo:** Libera al equipo de tareas repetitivas, permitiendo centrarse en otras prioridades.

Estrategias de automatización

- **Tareas programadas:**
  Utiliza cron jobs en el host o en contenedores dedicados para ejecutar scripts de backup periódicamente (por ejemplo, cada noche).
- **Contenedores especializados:**
  Usa imágenes Docker diseñadas para realizar backups automáticos y gestionarlos, como **docker-volume-backup**.
- **Monitorización y alertas:**
  Configura notificaciones ante fallos en los backups o falta de espacio en disco.

4. Verificación y restauración de backups

¿Por qué verificar los backups?

- Un backup no verificado puede estar corrupto o incompleto.
- La única forma de asegurar que un backup sirve es restaurándolo y comprobando los datos.

Estrategias de verificación

- **Verificación de integridad:** Usa checksums (`sha256sum`, `md5sum`) para comprobar que los archivos no se han corrompido.
- **Pruebas de restauración:** Restaura backups en un entorno de pruebas y valida que los servicios y datos funcionan correctamente.
- **Automatización:** Programa verificaciones periódicas y registra los resultados.

Ejemplo: Verificar integridad de un backup

```
sha256sum backup-completo-2025-07-29.tar.gz > backup-completo-2025-07-29.tar.gz.sha256
sha256sum -c backup-completo-2025-07-29.tar.gz.sha256
```

Ejemplo: Restaurar un volumen Docker desde un backup

```
docker volume create datos_app_restaurado
docker run --rm \
  -v datos_app_restaurado:/data \
  -v $(pwd):/backup \
  alpine \
  tar xzvf /backup/backup-completo-2025-07-29.tar.gz -C /data
```

Ejemplo: Restaurar una base de datos PostgreSQL

```
cat pg_backup-2025-07-29.sql | docker exec -i mi_postgres psql -U postgres mi_db
```