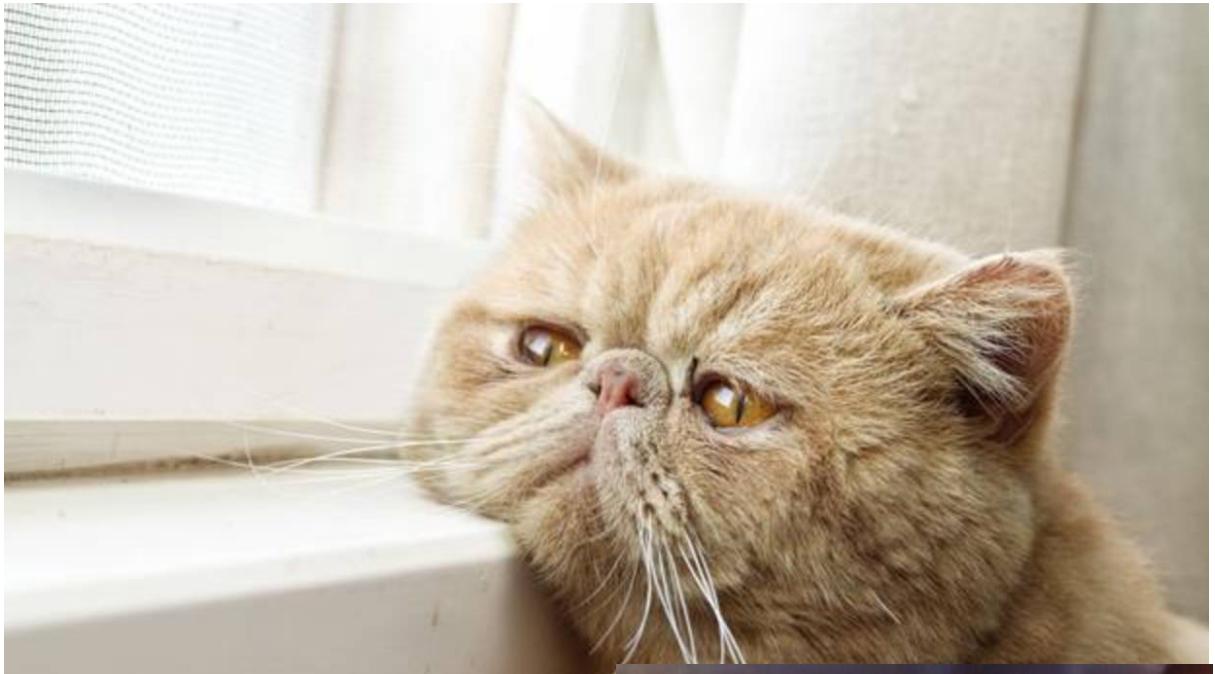


PROYECTO FINAL DE LA MUERTE

EVIL CORP



Alumno: Marcel Setó Gómez
Profesor: Rafa Laguna

ÍNDICE

1. DIAGRAMA, SQL Y SCRIPTS.....	2
1.1 TABLA USUARIOS CAMPOS.....	2
1.2 SQL SENTENCES Y SCRIPTS.....	2
1.2.1 SCRIPT 1: CREAR LA BASE DE DATOS Y LA TABLA E INSERTAR EL USUARIO ROOT.....	2
1.2.2 SCRIPT 2: CREACIÓN DE USUARIOS.....	3
2. HARDWARE DE LA BASE DE DATOS DE LA EMPRESA.....	4
2.1 ¿QUE HARDWARE NECESITAREIS?.....	4
2.1.1 RAM.....	4
2.1.2 DISCOS DUROS.....	4
2.1.3 PROCESADORES.....	5
2.1.4 PLACAS BASE.....	5
2.1.5 CAJAS / RACKS.....	6
2.2 CLUSTERING.....	6
2.3 PLAN DE SEGURIDAD.....	7
2.3.1 ¿QUÉ HARDWARE COMPRARÉIS PARA AYUDAROS?.....	7
2.3.2 ¿CÓMO Y CUÁNDO SE REALIZAN LAS COPIAS DE SEGURIDAD?.....	7
2.3.3 ¿QUÉ OCURRE SI FALLA UN DISCO DURO Y DEBEMOS RECUPERAR? ¿CÓMO ENTRA EN JUEGO LA CONFIGURACIÓN DE LOS DISCOS DUROS?.....	7
2.3.4 ¿Y SI FALLA UNA MÁQUINA ENTERA?.....	8
2.4 PRESUPUESTO APROXIMADO.....	8
3. CRON.....	9
4. ANEXO.....	9
4.1 TAR.GZ.....	9

1. DIAGRAMA, SQL Y SCRIPTS

1.1 TABLA USUARIOS CAMPOS

id_user (id_user) INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT

username VARCHAR(32) NOT NULL

password VARCHAR(32) NOT NULL (guardado en hash md5)

name VARCHAR(16) NOT NULL

surname VARCHAR(32) NOT NULL

email VARCHAR(64) NOT NULL

scontact_number VARCHAR(13) NOT NULL

register DATETIME NOT NULL

country VARCHAR(32) NOT NULL

gender CHAR (1)

birthdate DATE

Users	
* id_user	UINT
* username	VARCHAR (32)
* password	VARCHAR (32)
* name	VARCHAR (16)
* surname	VARCHAR (32)
* email	VARCHAR (64)
* contact_number	VARCHAR (13)
* register	DATETIME
* country	VARCHAR (32)
* gender	CHAR (1)
* birthdate	DATE

JUSTIFICACIÓN:

He elegido los siguientes campos ya que me parecen los más indicados y relevantes para almacenar en una base de datos que trata la información de la diferente población mundial.

1.2 SQL SENTENCES Y SCRIPTS

1.2.1 SCRIPT 1: CREAR LA BASE DE DATOS Y LA TABLA E INSERTAR EL USUARIO ROOT

```
CREATE DATABASE EvilCorp;
```

```
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE users (id_user INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT, username VARCHAR(32) NOT NULL, password VARCHAR(32) NOT NULL, name VARCHAR(16) NOT NULL, surname VARCHAR(32) NOT NULL, email VARCHAR(64) NOT NULL, contact_number VARCHAR(13) NOT NULL, register DATETIME NOT NULL, country VARCHAR(32) NOT NULL, gender CHAR(1) NOT NULL, birthdate DATE NOT NULL);
```

```
INSERT INTO users(username, password, name, surname, email, contact_number, register, country, gender, birthdate) VALUES ("root", "fb226127d0e2e623fb639a1b38bb88eb", "Hilon", "Musgo Muskiz", "hilonamusgoking@gmail.com", "+3465401345", now(), "Spain", 'N', "1971-06-28");
```

JUSTIFICACIÓN:

A través de las sentencias mencionadas anteriormente, se ejecutará una serie de acciones en nuestra base de datos llamada "EvilCorp". En primer lugar, se comprobará si la base de datos ya existe. En caso de que no exista, se creará.

Luego, se procederá a eliminar la tabla "users" si ya existe en la base de datos. Una vez que se haya eliminado la tabla, se creará una nueva tabla "users" con los parámetros y campos necesarios para almacenar información de los usuarios. Es importante destacar que el campo "DATE" de la tabla se establecerá por defecto para que registre la hora exacta en que se registre un nuevo usuario en la base de datos, esto lo haremos con la función de mysql now().

1.2.2 SCRIPT 2: CREACIÓN DE USUARIOS

```
CREATE USER 'insertupdate'@'localhost' IDENTIFIED BY 'enti';
CREATE USER 'select'@'localhost' IDENTIFIED BY 'enti';
GRANT INSERT, UPDATE ON evilcorp.* TO
'insertupdate'@'localhost';
GRANT SELECT ON evilcorp.* TO 'select'@'localhost';

FLUSH PRIVILEGES;
```

JUSTIFICACIÓN:

Primero de todo creamos dos usuarios uno para insertar+actualizar contenido y el otro para leerlo (select). Ambos usuarios tendrán como contraseña enti.

A continuación les damos los permisos correspondientes, por un lado el de insertar y actualizar a uno, y por otro lado el de select al otro, les damos únicamente permisos a la base de datos de la empresa (EvilCorp) y dentro de la propia base de datos a todas las tablas. A continuación con el flush privileges, actualizamos y refrescamos los cambios. Finalmente en terminal añadiremos la siguiente instrucción en root para poder ejecutar el script. *cat nombreDelScript.sql | mysql*

2. HARDWARE DE LA BASE DE DATOS DE LA EMPRESA

2.1 ¿QUE HARDWARE NECESITAREIS?

Como es obvio para poder almacenar los datos de toda la población mundial y unos cuantos miles de millones más, necesitaremos un hardware muy potente y escalable, con una capacidad de almacenamiento enorme y una alta disponibilidad. Podemos decir que en este caso los servidores de alta gama suelen ser los más adecuados para este tipo de proyectos.

2.1.1 RAM

En cuanto a la memoria RAM he considerado que se necesitará un mínimo de 8GB de RAM para almacenar los datos de cada persona. Por lo tanto si hacemos los cálculos aproximados podemos ver que necesitaremos al menos (7,888 miles de millones de personas X 2000 bytes) 19,776 TB para almacenar los datos de toda la población del planeta además le añadiré 10TB más para las próximas generaciones que vendrán.

El modelo de RAM que usará la empresa será la **Samsung**

DDR4-3200 ECC LRDIMM

capaz de almacenar 64 GB de memoria ya que es la que más se adapta al tipo de base de datos que queremos generar.



2.1.2 DISCOS DUROS

La empresa usará aproximadamente 110 discos duros del modelo **Seagate Exos X16 3.5" 16TB SATA3** capaces de almacenar 16 TB cada una.

En cuanto a la configuración para asegurar que no fallarán, creo que la mejor técnica de almacenamiento redundante de las que hemos visto en clase para un servidor de estas características es la RAID (Redundant Array of Inexpensive Disks). Las raids usan varios discos duros conectados a la vez y por lo tanto distribuyen los datos en múltiples unidades para minimizar la posibilidad de fallos. En nuestro caso usariamos la RAID 6 ya

que proporciona una alta tolerancia a fallos y redundancia de datos. Sabemos que RAID 6 es similar a RAID 5, pero los diferentes datos se escriben en dos unidades en lugar de una, lo que proporciona una mayor capacidad de recuperación en caso de que fallen simultáneamente dos unidades. Aunque es poco probable que esto suceda, a diferencia de la RAID 5, la RAID 6 proporciona una mayor seguridad durante la reconstrucción de una unidad fallida. Esto lo que nos ayuda es a evitar la pérdida de datos en caso de fallos simultáneos de varias unidades.

Las RAIDs son muy importantes ya que hay que tener en cuenta que los discos duros y las SSD pueden fallar, lo que significa que eventualmente pueden dejar de funcionar y perder datos. Con las RAIDs nos evitamos estas pérdidas creando copias de seguridad de la información y los diferentes datos.

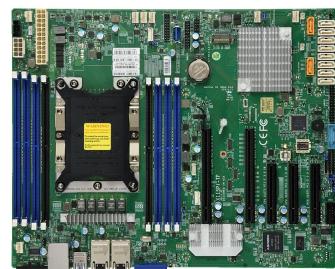


2.1.3 PROCESADORES

El procesador más adecuado para servidores de alta gama es actualmente el **Intel Xeon Platinum 8380**. Aunque estos tienen una capacidad de proceso menor que los procesadores Intel Core, están mejor optimizados para la ejecución en paralelo y de los diferentes hilos, lo que los hace ideales para servidores que tienen una gran exigencia de ejecución de múltiples procesos simultáneos como el nuestro. Además estos también tienen un mayor rendimiento de sus núcleos, memoria y ancho de banda para acelerar la ejecución de procesos y agilizar las cargas de trabajo.

2.1.4 PLACAS BASE

Las placas base que usaría sería la **Supermicro X11SPi-TF** ya que se trata de una placa base de alto rendimiento diseñada para servidores, con soporte para procesadores Xeon Scalable (el que usamos nosotros) de hasta 28 núcleos y 56 hilos. Además esta, cuenta con una amplia variedad de puertos de entrada/salida y una gran cantidad de opciones de expansión, lo que la hace ideal para cargas de trabajo intensivas como en el caso de nuestra empresa.



2.1.5 CAJAS / RACKS

Se necesitan cajas/racks que permitan una buena ventilación y gestión del cableado. Por lo tanto para un proyecto de este calibre, se requeriría un rack de servidor de alta densidad que pueda alojar un gran número de unidades de almacenamiento y servidores en un espacio reducido. Finalmente tras informarme sobre diferentes modelos he decidido que las cajas/racks que utilizaremos para nuestro servidor serán las **Dell PowerEdge R740xd**. Estas cajas/racks están especialmente diseñadas para una gran cantidad de almacenamiento y una buena gestión térmica y de cables, lo que es fundamental para un proyecto de esta escala.



Además, el servidor también contará con fuentes de alimentación redundantes y unidades de refrigeración para mantener el sistema funcionando de manera óptima y evitar fallos en el hardware.

2.2 CLUSTERING

Para diseñar el sistema de clustering, la empresa ha decidido que necesitaremos alrededor de 8 máquinas para almacenar y procesar la gran cantidad de datos e información correspondiente a la base de datos de manera óptima. Para distribuir toda esta carga de trabajo de manera equitativa, podemos subdividir las 8 máquinas en dos clusters, cada uno con control sobre cuatro máquinas, y conectar estos clusters entre sí. De esta manera, cada máquina tendrá una carga de trabajo equilibrada y la base de datos podrá rendir correctamente. Además, configuraremos un almacenamiento compartido con los clusters de balanceo para garantizar que la integridad de la base de datos no se vea comprometida si falla alguna máquina. También podemos implementar una mecánica de recuperación en caso de fallo de una máquina, de manera que otra máquina asuma la responsabilidad de la fallida, lo que nos permitirá evitar la pérdida de información valiosa.

En definitiva con este sistema de clustering podremos estar siempre activos pese a que existan muchas consultas

2.3 PLAN DE SEGURIDAD

2.3.1 ¿QUÉ HARDWARE COMPRARÉIS PARA AYUDAROS?

El Hardware que usaré para realizar las diferentes copias de seguridad serán diferentes discos duros externos en concreto **Seagate Exos X16 3.5" 16TB SATA3**. También otra opción que había pensado inicialmente era la de utilizar el almacenamiento en la nube para mantener las copias de seguridad fuera del sitio físico y proteger contra desastres naturales o robos, aún así esto último al estar conectado a la red sería más vulnerable a diferentes ciberataques.

2.3.2 ¿CÓMO Y CUÁNDO SE REALIZAN LAS COPIAS DE SEGURIDAD?

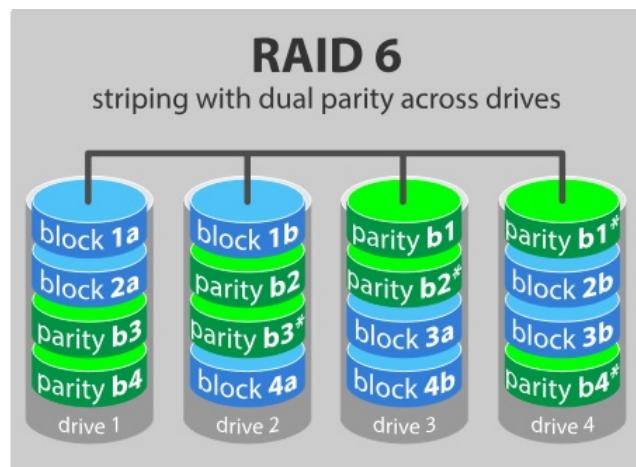
Al ser una empresa tan importante que acapara la gente de todo el globo terráqueo, se deben establecer políticas de copias de seguridad rigurosas que aseguren la protección de los datos y permitan la recuperación en caso de desastres.

Las copias de seguridad se realizarán de manera regular ya que en el caso de nuestra empresa constantemente se trabaja con un gran flujo de datos. Por lo tanto hemos decidido que se harán copias de seguridad diariamente. En el caso de la empresa estas copias de seguridad diarias se realizarán en horas de menor actividad para minimizar el impacto en el rendimiento de la base de datos y las aplicaciones.. También cabe recalcar que una copia de seguridad anterior será borrada una vez hecha la nueva.

2.3.3 ¿QUÉ OCURRE SI FALLA UN DISCO DURO Y DEBEMOS RECUPERAR? ¿CÓMO ENTRA EN JUEGO LA CONFIGURACIÓN DE LOS DISCOS DUROS?

Como empresa tenemos un plan de recuperación ante desastres para garantizar que se pueda recuperar la base de datos en caso de fallas en el hardware o software. Nuestro plan incluye tanto la recuperación de datos de las copias de seguridad como la restauración de la base de datos en un servidor de reemplazo.

En este caso la configuración de los discos duros (RAIDS) juega un papel importantísimo en la recuperación de los datos. En nuestro caso al ser una gran empresa como he dicho anteriormente hemos utilizado la RAID 6, lo que nos permite recuperar los datos en caso de falla de dos discos duros.



2.3.4 ¿Y SI FALLA UNA MÁQUINA ENTERA?

Si falla una máquina entera lo primero que haremos en la empresa es identificar la causa de la falla y evaluar los daños para determinar si es posible reparar la máquina o si se debe reemplazar. Si la máquina no se puede reparar, lo que haremos será reemplazarla por una nueva compatible con el sistema ya presente y la configuración de RAID 6.

Una vez que se haya reemplazado la máquina, restauraremos el sistema y los diferentes datos desde las copias de seguridad hechas. Esto será un proceso largo, que dependerá del tamaño de los datos y la frecuencia de las copias de seguridad. Es por eso que es importante lo comentado en los puntos anteriores de realizar copias de seguridad frecuentes y pruebas de recuperación para garantizar la integridad de los datos y la rapidez de recuperación en caso de un desastre. Por lo tanto en el presupuesto final habrá que sumar la mitad de las máquinas que tenemos.

2.4 PRESUPUESTO APROXIMADO

PROCESADORES Intel Xeon Platinum 8380: 6,847.67€ x 40 PROCESADORES = 273,906.80€
 RAM'S Samsung DDR4-3200 ECC LRDIMM: 150€ x 500 RAM'S = 75,000€
 DISCOS DUROS Seagate Exos X16 3.5" 16TB SATA3: 450€ x 110 DISCOS DUROS = 49,500€
 PLACAS BASE Supermicro X11SPi-TF: 600€ x 40 PLACAS BASE = 24,000€
 PRECIO POR SI FALLA UNA MÁQUINA ENTERA (mitad de las máquinas que tenemos) = 422,406.80 / 2 = 211,203.40€

 PRECIO TOTAL = 422,406.80€ + 211,203.40€ = 633,610.20€

3. CRON

```
#!/bin/bash

DIA=`echo $(date + %F) `

RUTA_COPIADIARIA="/root/backup/diario/"

rm $RUTA_COPIADIARIA*
mysqldump EvilCorp > ${RUTA_COPIADIARIA}copia.sql
tar -czvd ${RUTA_COPIADIARIA}copia_${DIA}.tar.gz
${RUTA_COPIADIARIA}copia.sql
rm ${RUTA_COPIADIARIA}copia.sql
```

JUSTIFICACIÓN:

Realizo una copia de seguridad diaria en la base de datos llamada "EvilCorp" y la comprimo en un archivo tar.gz, eliminando la copia anterior. La copia se guarda en la ruta /root/backup/diario/ y tiene el nombre "copia_fecha.tar.gz", donde la fecha se obtiene a través del comando "date" y se guarda en la variable \$DIA. Primero, se elimina la copia anterior en la ruta de copia diaria con el comando "rm". Luego, se utiliza el comando "mysqldump" para crear una copia de seguridad de la base de datos "EvilCorp" en un archivo llamado "copia.sql" en la ruta de copia diaria. A continuación, se utiliza el comando "tar" para comprimir el archivo "copia.sql" en un archivo "copia_fecha.tar.gz" en la misma ruta, utilizando la opción "-czvd" para comprimir el archivo y mostrar la salida de la compresión en la pantalla. Finalmente, se elimina el archivo "copia.sql".

4. ANEXO

4.1 TAR.GZ

Primero de todo creamos un directorio en el home del usuario donde almacenar los diferentes scripts que queremos comprimir, en mi caso el directorio se llamará entregaFinalBD. Dentro del directorio se encontrarán tanto los dos scripts SQL (1. Crear base de datos + insertar a Elon Musgo | 2. Dar permisos a diferentes usuarios a nuestra base de datos) como el de BASH (el cron (copia de seguridad)).

```
marcel@enti:~$ pwd  
/home/marcel  
marcel@enti:~$ ls  
adivina_el_numero.sh  cowsay          keysitas      -M0  
añadirCambiosBD.sh    entregaFinalBD  keysitas.pub   M04  
buscapacoscript.sql   exportpaco.sql  lrwxrwxrwx    M10
```

```
marcel@enti:~/entregaFinalBD$ pwd  
/home/marcel/entregaFinalBD  
marcel@enti:~/entregaFinalBD$ ls  
cron.sh  scriptSQLCreacionBD.sql  scriptSQLPermisos.sql
```

Con la siguiente instrucción comprimimos el directorio que deseamos al formato tar.gz.

```
tar -czvf entregaFinalBD.tar.gz entregaFinalBD
```

tar → Formato en que se comprime, en nuestro caso tar.gz

-czvd → Significa que la instrucción es de comprimir el archivo o el directorio en nuestro caso

entregaFinalBD.tar.gz → Nombre final del comprimido

entregaFinalBD → Directorio que queremos comprimir

```
marcel@enti:~$ ls  
adivina_el_numero.sh  entregaFinalBD      keysitas.pub  M10UF01  
añadirCambiosBD.sh    entregaFinalBD.tar.gz  lrwxrwxrwx  MarcelSeto8.github  
buscapacoscript.sql   exportpaco.sql       -M04UF2      Práctica1_MARCEL  
cowsay                keysitas            M04UF2      procedure.sql
```