DICIEMBRE 2020

EXAMEN PARCIAL #3

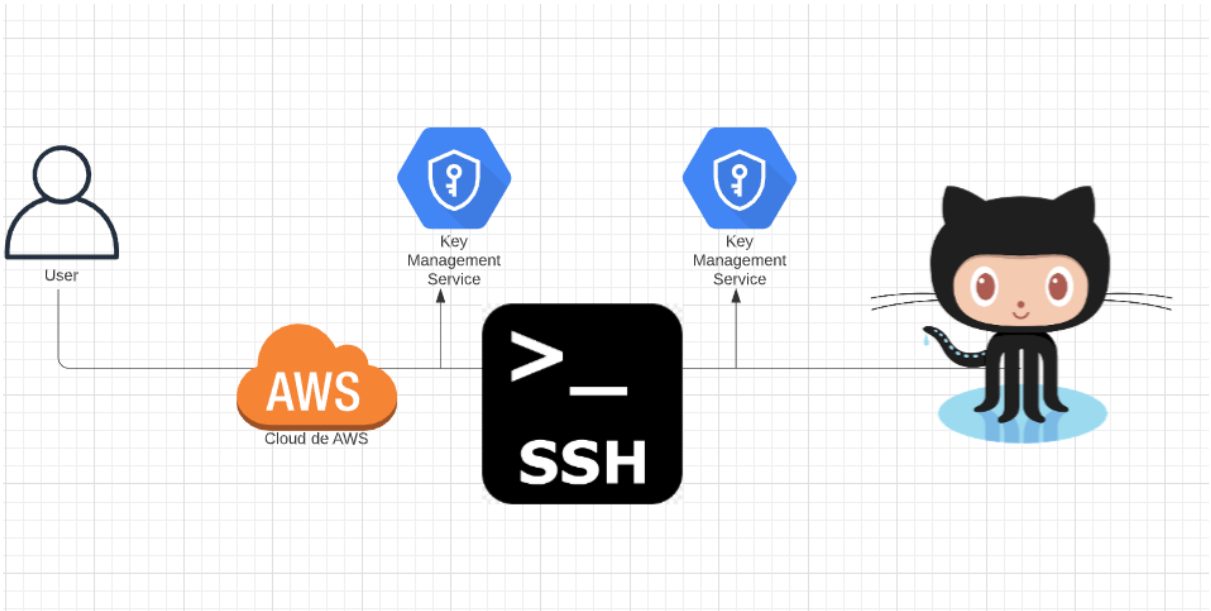
ADMINISTRACIÓN DE BASE DE DATOS

Preparado por: Rafael Alejandro Raygosa Mena

01

RESPALDOS AUTOMATIZADOS

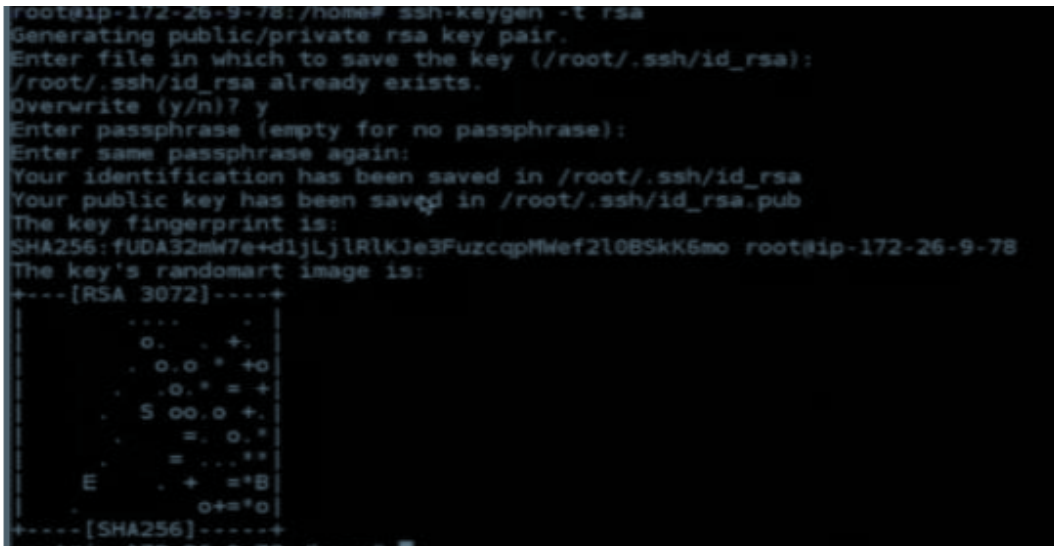
DIAGRAMA DESCRIPTIVO



02

PROCESO

Para realizar el respaldo automatizado primero deberemos entrar a la plataforma de github y crearemos un nuevo repositorio privado , este repositorio será el que va a recibir los respaldos automatizados.
Posteriormente dentro de nuestra instacia creada en la nube de aws ejecutaremos el comando **ssh-keygen -t rsa** este lo que hará es crear una llave publica y una llave primara ssh para obtener los permisos necesarios y poder establecer la conexión entre la instancia y git.



LLAVES REALIZADAS

Una vez generadas nuestra llaves la agregaremos a nuestro repositorio accediendo a settings, posteriormente Deploy keys y por ultimo Add deploy key, al ingresar a esta ultima ventana se nos pedira un nombre y el contenido de nuestra llave publica , para obtener dicho contenido vamos a ejecutar el comando **vi nombre de la ruta de la llave** este comando se utiliza para editar el contenido de los archivos y nos mostrara el contenido del nuestro, entonces copiaremos y pegaremos este contenido en el apartado que nos indica en github y añadimos la llave.
IMPORTANTE: debemos de seleccionar la opción allow key access para que los respaldo se realizan de manera automatizada

03

Teniendo ya nuestras llaves vamos a clonar nuestro repositorio git en la instancia de la nueve con el comando **git clone link de ru** repositorio que nos permitirá clonar nuestro repositorio y ojo el link del repositorio debe ser el tipo ssh, posteriormente se deberá crear el archivo uploadgit.sh pero antes debemos cambiar la ruta del los respaldos en la instancia de la nube a la ruta de la carpeta de nuestro repositorio clonado y ahora si crearemos el archivo uploadgit.sh con los comandos:

- **git add .**
- **git commit -m 'Dialy backup'**
- **git push origin master**

Este primer comando tendrá la funcion de añadir todos los nuevos archivos cargados, el segundo indica el registro del cambio y nuestro tercer y ultimo comando simplemente indica que se suban los nuevos respaldos a la nube.

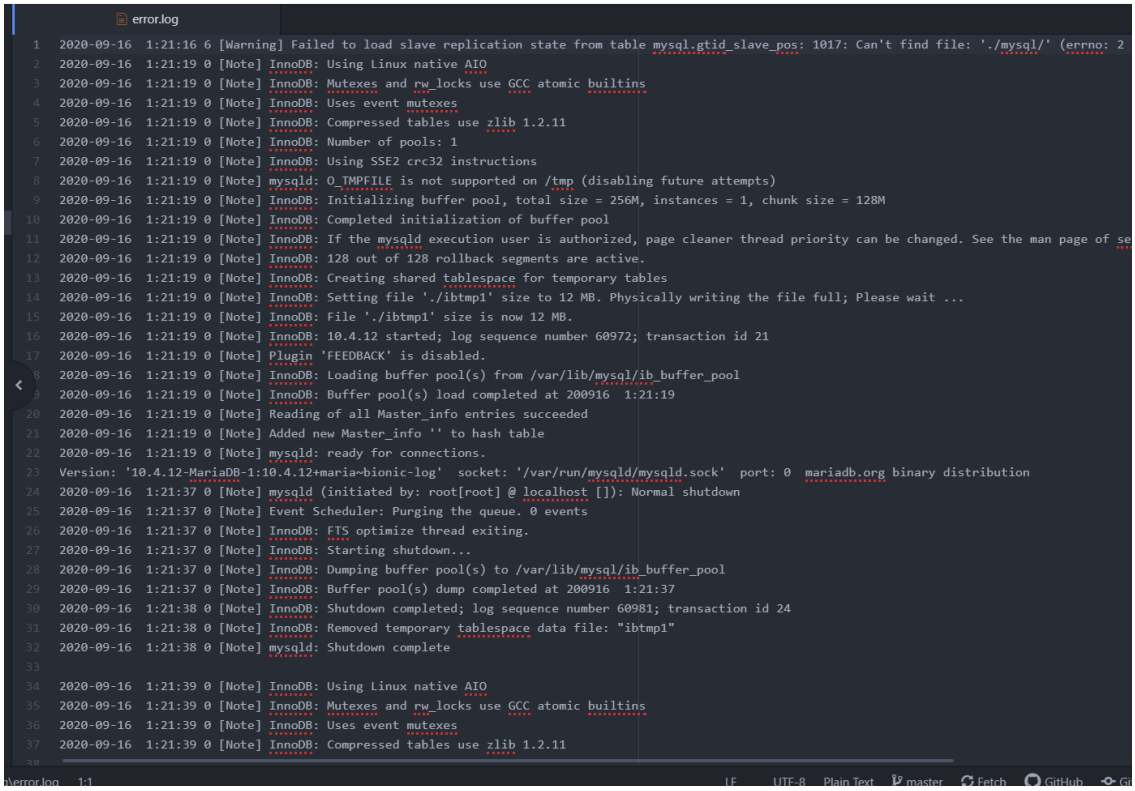
Ya casi por terminar ejecutaremos el comando **- chmod +x** que tiene como funcion otorgar permisos de ejecución y en este caso se lo otorgaremos al archivo uploadgit.sh, tambien agregaremos el comando **crontab -e** para que nos permita determinar el tiempo en el que se ejecutara el archivo de uploadgit (el que subira nuestros respaldos a git).

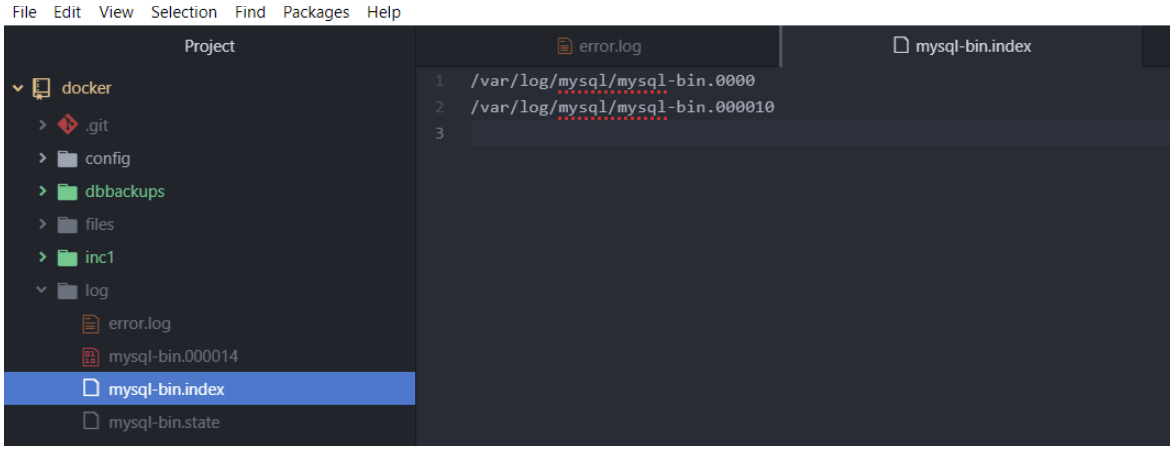
Podremos verificar que todo este funcionando con el comando **./uploadgit.sh** que nos permitirá ejecutar el archivo upload y en nuestro repositorio en git ya se visualizaran los respaldos.

04

LOGS

Evidencia de archivos error.log y mysql-bin.index





05

LOGS

Al intentar acceder a nuestra base de datos con los datos incorrectos se mostrara en el archivo erro.log líneas de código como las que se muestran a continuación

```
2450    2020-12-17 23:47:51 2277 [Warning] Access denied for user 'root'@'172.18.0.1' (using password: YES)
```

Esto nos indicara un intento de inicio sospechoso en lo cual podremos decir que estan intentando atacarnos con robo de información por lo que se recomienda bloquear la ip del usuario que esta intentando atacar.

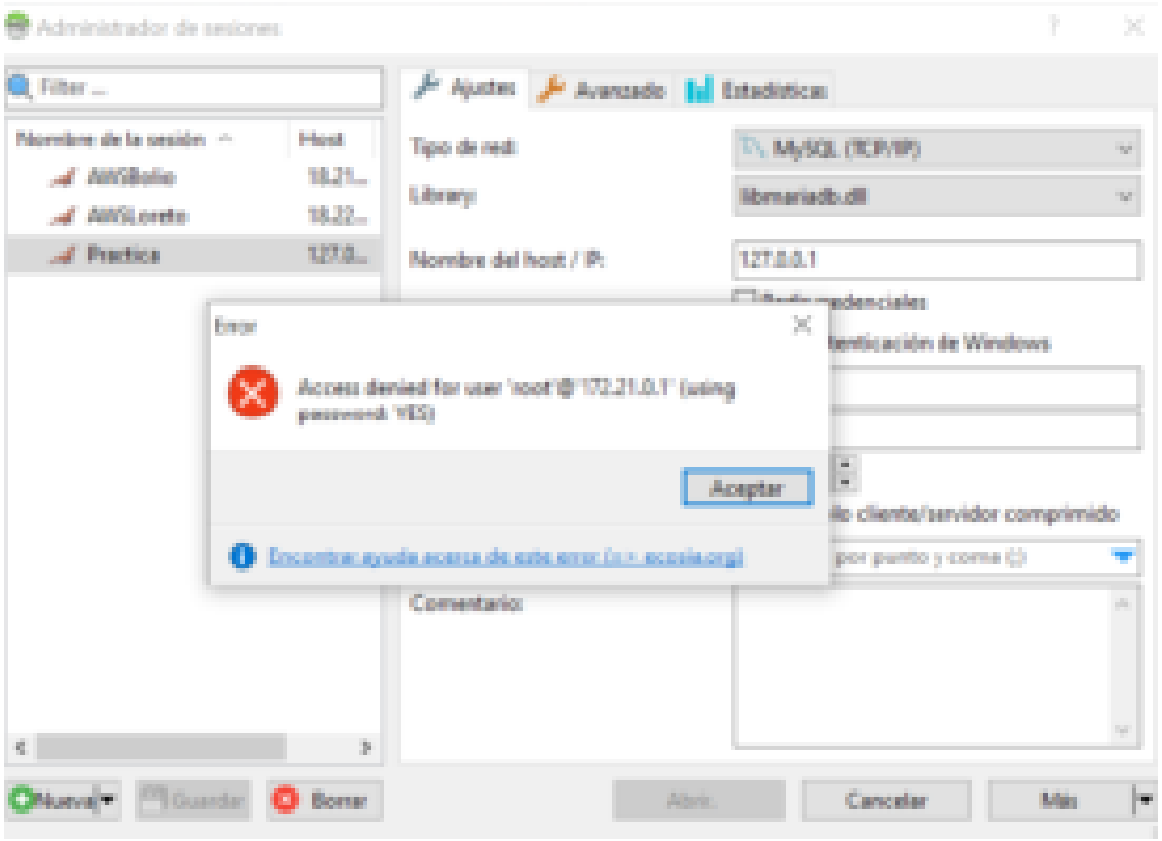
Ejecutaremos el siguiente comando para poder obtener la traducción de nuestro archivo binario sql , que sirven con restauración de un gestor de base de datos

```
C:\docker>docker exec docker_db_1 mysqlbinlog /var/log/mysql/mysql-bin.000014 > binary.sql
```

ARCHIVO TRADUCIDO

```
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!40019 SET @@session.max_insert_delayed_threads=0*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 4
#201218 20:04:54 server id 1  end_log_pos 256 CRC32 0x212297a2  Start: binlog v 4, server v 10.4.12-MariaDB-1:10.4.12+maria~bionic-log created 201218 
# Warning: this binlog is either in use or was not closed properly.
ROLLBACK/*!*/;
BINLOG '
ZgvdXw8BAAAA/AAAAAABAAABAAQAMTAuNC4xMi1NYXJpYURCLTE6MTAuNC4xMittYXJpYX5iaW9u
aWMtbG9nAAAAAAAAAABmC91fEzgNAAgAEgAEBAQEEgAA5AAEGggAAAAICAgCAAAACgoKAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAEEwQADQgICAoKCgGilyIh
'/*!*/;
# at 256
#201218 20:04:54 server id 1  end_log_pos 299 CRC32 0xe646ec35  Gtid list [0-1-7288]
# at 299
#201218 20:04:54 server id 1  end_log_pos 342 CRC32 0x22e36454  Binlog checkpoint mysql-bin.000014
DELIMITER ;
# End of log file
ROLLBACK /* added by mysqlbinlog */;
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
```

06 EVENTO DE SEGURIDAD



Como evento de seguridad intentaremos entrar a nuestro gestor de base de datos con otra contraseña al menos 3 veces y como se menciona anteriormente en nuestro archivo log podremos observar al menos 3 inicios fallidos por lo que deduciremos que estan intentando realizar un ataque, entonces bloquearemos la ip del usuario que esta intentando acceder. Estos errores o advertencias seran faciles de identificar por que justo tienen una "etiqueta" llamada warning.

```
429 2020-12-18 20:06:57 9 [Warning] Access denied for user 'root'@'160.15.0.1' (using password: YES)
430 2020-12-18 20:18:32 10 [Warning] Access denied for user 'root'@'160.15.0.1' (using password: YES)
431 2020-12-18 20:18:43 11 [Warning] Access denied for user 'root'@'160.15.0.1' (using password: YES)
432 2020-12-18 20:31:52 0 [Note] mysqld (initiated by: unknown): Normal shutdown
```