

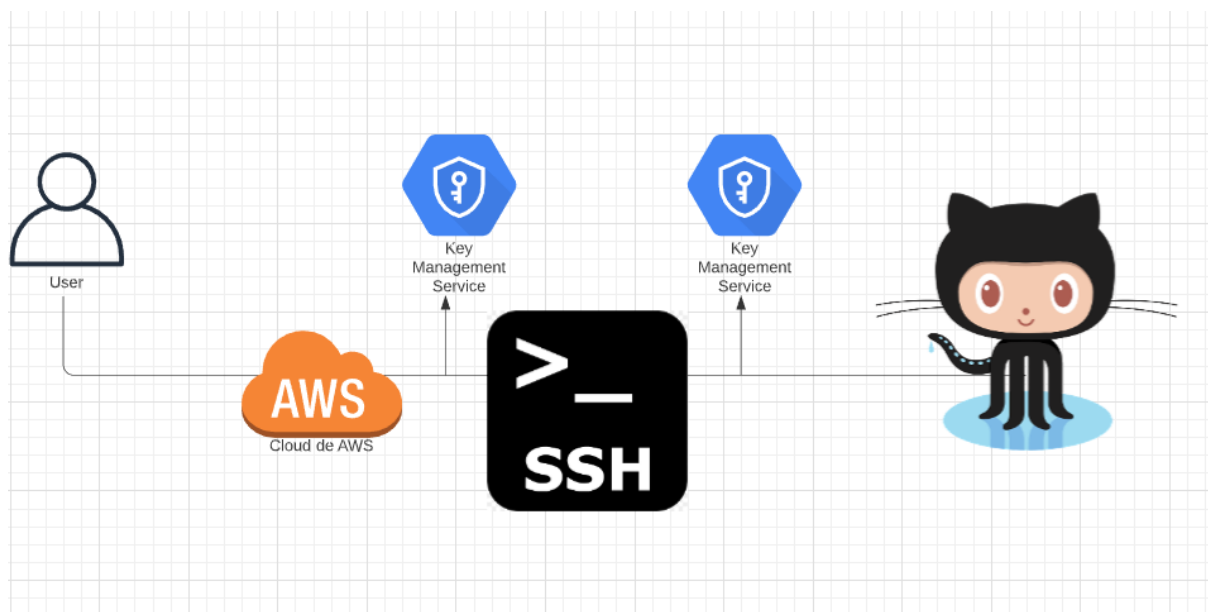
EXAMEN PARCIAL #3

ADMINISTRACIÓN DE BASE DE DATOS

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 instancia 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.

```
root@ip-172-26-9-78:/home# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:fUDAS2mW7e+d1jLj1RlKJe3FuzcqpMwef2l0BSkK6mo root@ip-172-26-9-78
The key's randomart image is:
+----[RSA 3072]----+
  . . . . .
  o. . . +.
  . o.o * +o
  . .o.* = +
  . 5 oo.o +.
  . =. o.*
  . = ...**
  E . + =*B
  . o+*o
+----[SHA256]-----+
```

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 nube 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 de 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 función de añadir todos los nuevos archivos cargados, el segundo indica el registro del cambio y nuestro tercer y último comando simplemente indica que se suban los nuevos respaldos a la nube.

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

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

04

LOGS

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

```
error.log
1 2020-09-16 1:21:16 6 [Warning] Failed to load slave replication state from table mysql.gtid_slave_pos: 1017: Can't find file: './mysql/' (errno: 2
2 2020-09-16 1:21:19 0 [Note] InnoDB: Using Linux native AIO
3 2020-09-16 1:21:19 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
4 2020-09-16 1:21:19 0 [Note] InnoDB: Uses event mutexes
5 2020-09-16 1:21:19 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
6 2020-09-16 1:21:19 0 [Note] InnoDB: Number of pools: 1
7 2020-09-16 1:21:19 0 [Note] InnoDB: Using SSE2 crc32 instructions
8 2020-09-16 1:21:19 0 [Note] mysqld: O_TMPFILE is not supported on /tmp (disabling future attempts)
9 2020-09-16 1:21:19 0 [Note] InnoDB: Initializing buffer pool, total size = 256M, instances = 1, chunk size = 128M
10 2020-09-16 1:21:19 0 [Note] InnoDB: Completed initialization of buffer pool
11 2020-09-16 1:21:19 0 [Note] InnoDB: If the mysqld execution user is authorized, page cleaner thread priority can be changed. See the man page of se
12 2020-09-16 1:21:19 0 [Note] InnoDB: 128 out of 128 rollback segments are active.
13 2020-09-16 1:21:19 0 [Note] InnoDB: Creating shared tablespace for temporary tables
14 2020-09-16 1:21:19 0 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...
15 2020-09-16 1:21:19 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
16 2020-09-16 1:21:19 0 [Note] InnoDB: 10.4.12 started; log sequence number 60972; transaction id 21
17 2020-09-16 1:21:19 0 [Note] Plugin 'FEEDBACK' is disabled.
18 2020-09-16 1:21:19 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
19 2020-09-16 1:21:19 0 [Note] InnoDB: Buffer pool(s) load completed at 200916 1:21:19
20 2020-09-16 1:21:19 0 [Note] Reading of all Master_info entries succeeded
21 2020-09-16 1:21:19 0 [Note] Added new Master_info '' to hash table
22 2020-09-16 1:21:19 0 [Note] mysqld: ready for connections.
23 Version: '10.4.12-MariaDB-1:10.4.12+maria-bionic-log' socket: '/var/run/mysqld/mysqld.sock' port: 0 mariadb.org binary distribution
24 2020-09-16 1:21:37 0 [Note] mysqld (initiated by: root[root] @ localhost []) : Normal shutdown
25 2020-09-16 1:21:37 0 [Note] Event Scheduler: Purging the queue. 0 events
26 2020-09-16 1:21:37 0 [Note] InnoDB: FTS optimize thread exiting.
27 2020-09-16 1:21:37 0 [Note] InnoDB: Starting shutdown...
28 2020-09-16 1:21:37 0 [Note] InnoDB: Dumping buffer pool(s) to /var/lib/mysql/ib_buffer_pool
29 2020-09-16 1:21:37 0 [Note] InnoDB: Buffer pool(s) dump completed at 200916 1:21:37
30 2020-09-16 1:21:38 0 [Note] InnoDB: Shutdown completed; log sequence number 60981; transaction id 24
31 2020-09-16 1:21:38 0 [Note] InnoDB: Removed temporary tablespace data file: "ibtmp1"
32 2020-09-16 1:21:38 0 [Note] mysqld: Shutdown complete
33
34 2020-09-16 1:21:39 0 [Note] InnoDB: Using Linux native AIO
35 2020-09-16 1:21:39 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
36 2020-09-16 1:21:39 0 [Note] InnoDB: Uses event mutexes
37 2020-09-16 1:21:39 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
38
```

File Edit View Selection Find Packages Help

Project	error.log	mysql-bin.index
<ul style="list-style-type: none">docker<ul style="list-style-type: none">.gitconfigdbbackupsfilesinc1log<ul style="list-style-type: none">error.logmysql-bin.000014mysql-bin.indexmysql-bin.state	<pre>1 /var/log/mysql/mysql-bin.0000 2 /var/log/mysql/mysql-bin.000010 3</pre>	

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

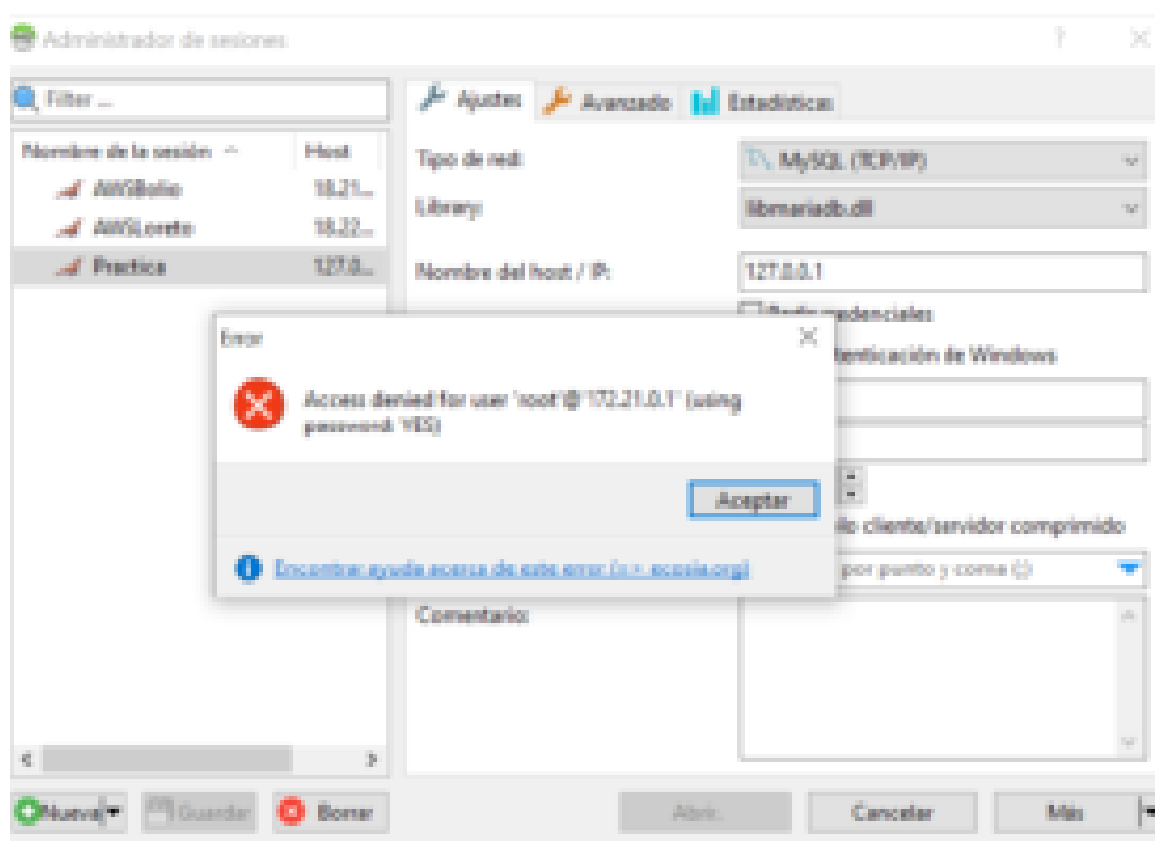
```
C:\docker>docker exec docker_db_1 mysqlbinlog /var/log/mysql/mysql-bin.000014 > binary.sql
```

ARCHIVO TRADUCIDO

```
1 /*150530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
2 /*140019 SET @@session.max_insert_delayed_threads=0*/;
3 /*150003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
4 DELIMITER /*!*/;
5 # at 4
6 #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 ;
7 # Warning: this binlog is either in use or was not closed properly.
8 ROLLBACK/*!*/;
9 BINLOG '
10 ZgvdXw8BAAAA/AAAAABAAABAAQAMTAuNC4xMi1NYXJpYURCLTE6MTAuNC4xMittYXJpYX5iaW9u
11 aWMTbG9nAAAAAABmC91FEzgNAAgAEgAEBAQEegAA5AAEGggAAAIICAgCAAAACgoKAAAAA
12 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
13 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
14 AAAAAAAAAAAAEwQADQgICAoKCgilyIh
15 '/*!*/;
16 # at 256
17 #201218 20:04:54 server id 1 end_log_pos 299 CRC32 0xe646ec35 Gtid list [0-1-7288]
18 # at 299
19 #201218 20:04:54 server id 1 end_log_pos 342 CRC32 0x22e36454 Binlog checkpoint mysql-bin.000014
20 DELIMITER ;
21 # End of log file
22 ROLLBACK /* added by mysqlbinlog */;
23 /*150003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
24 /*150530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
25 |
```

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] IP address '172.16.0.1' could not be resolved: Name or service not known
430 2020-12-18 20:06:57 9 [Warning] Access denied for user 'root'@'160.15.0.1' (using password: YES)
431 2020-12-18 20:18:32 10 [Warning] Access denied for user 'root'@'160.15.0.1' (using password: YES)
432 2020-12-18 20:18:43 11 [Warning] Access denied for user 'root'@'160.15.0.1' (using password: YES)
433 2020-12-18 20:31:52 0 [Note] mysqld (initiated by: unknown): Normal shutdown
```