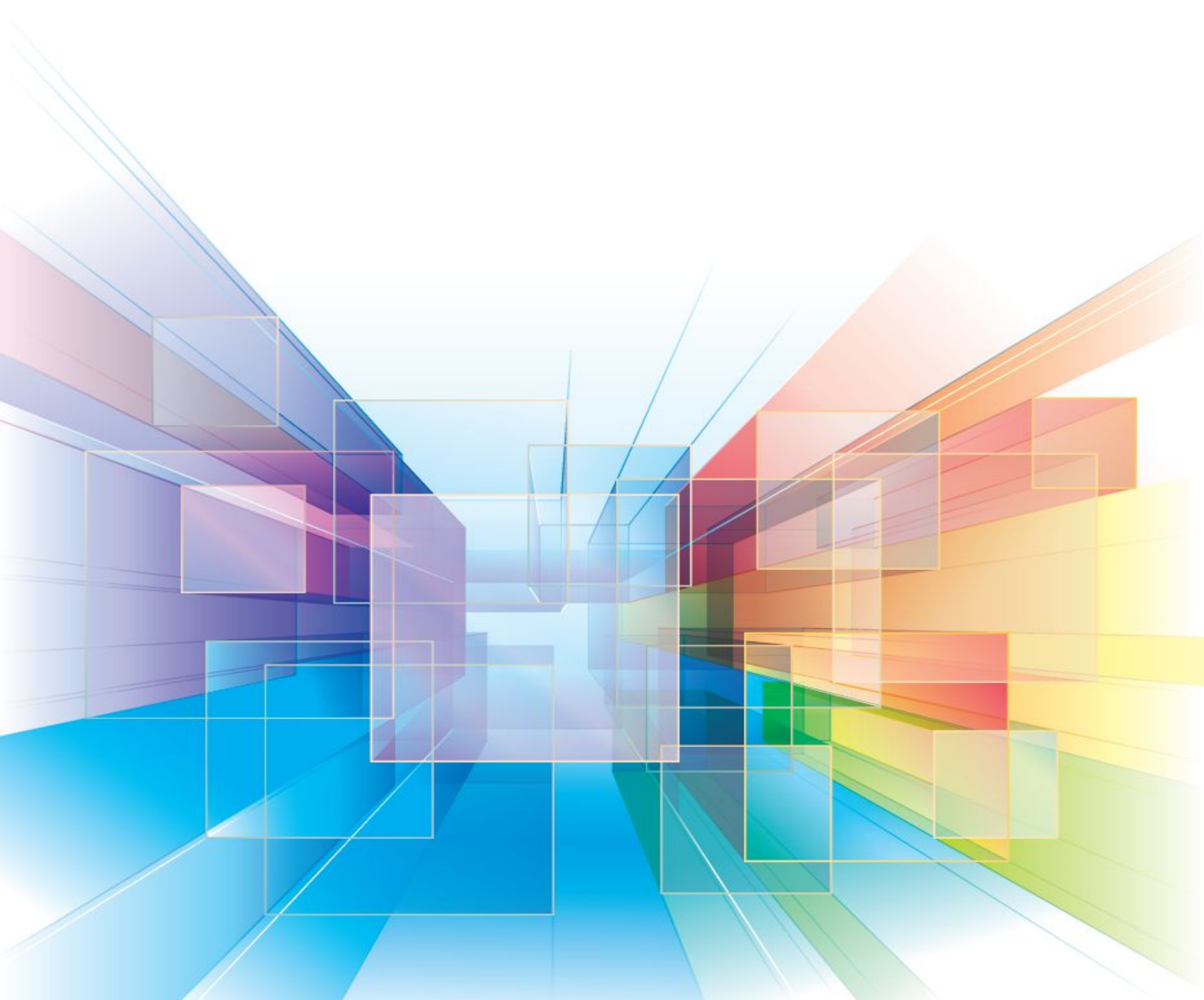


Bitácoras de base de datos

Por Jorge Said Serrano Soto



Métodos automatizados de respaldos de bases de datos	3
Respaldos automatizados con GitHub	3
Explicación del funcionamiento	3
Instalación y preparación de la instancia para el respaldo automatizado	3
Procesos de administración y operativos	4
Manejo de incidentes de seguridad	4
Ataque de fuerza bruta	4
Simulación de ejemplo	4
Contraataque de protección	4
Simulación de ejemplo	4
Contramedida de protección	4
Anexos	6
Diagrama de conexión de github a la nube mediante SSH img.1.0	6
Comandos de ejecución en la instancia en amazon para generar el SSH img.1.1	6
Comandos de ejecución en la instancia en amazon para generar el SSH segunda parte img.1.2	7
creación de la conexión por SSH mediante el uso de la clave generada en la instancia mediante la declaración de la clave img.1.3	7
demostración de clave SSH creada en git img.1.4	8
ejemplo del script que se ejecutara en el crontab para la subida de los respaldos a la nube de git img.1.5	8
ejemplo de la escritura dentro del crontab de la instancia para los scripts de la instancia que se encargan de los respaldos dum y de la subida de estos a un repositorio de git img.1.6	9
evidencia del resultado de los respaldos automatizados mediante el crontab y la escritura de un script con terminación sh img.1.6	9
evidencia de ejemplo real de la modificación del crontab para la declaración de los scripts img.1.7	10
contenido de un script para el respaldo dum que sera agregado en el crontab para su automatización img.1.8	10
generación de la clave ssh mediante el uso de comandos de ssh-keygen -t rsa img.1.9	11
evidencia del contenido de un script para la subida automatizada de archivos a un repositorio en git y su resultado en ejecución debido a su añadidura en el crontab img.1.10	11

llave ya instaurada en el repositorio img.2.1	12
ejemplo de ataque de fuerza bruta por acceso de claves de base de datos img.2.2	12
ejemplo de analisis de logs del ataque de fuerza bruta con claves de base de datos img.2.3	13
ejemplo de borrado accidental de datos de una tabla img.2.4	13
archivo proveniente del repositorio en donde se encuentran los respaldos img.2.5	14
ejemplo de restauración simple de una base de datos afectada img.2.6	14
instauración de conexiones en la instancia de AWS img.2.7	15
personalización de las conexiones para que la instancia se conecte a una sola ip a fin de fortalecer la conexión y bloquear las direcciones ip externas img.2.8	15
resultado de establecimiento de una sola dirección ip img.2.9	15

Métodos automatizados de respaldos de bases de datos

Respaldos automatizados con GitHub

Explicación del funcionamiento

Se toma en cuenta de que la automatización de los respaldos de bases de datos se pueden llegar a realizar de diferentes maneras, para este caso conectaremos a la instancia que tiene al gestor de la base de datos a un repositorio de GitHub mediante el uso de SSH key a fin de subir los respaldos de bases de datos a este repositorio, para esto explicaremos esto mediante un diagrama **img.1.0** en donde se muestra que están presentes tres elementos importantes, primero la instancia en AWS se hace presente como un servidor el cual tiene un gestor de base de datos que hace respaldos el cual se hace presente como el segundo elemento en el diagrama, como tercer elementos está el repositorio de GitHub el cual se conecta mediante el uso de SSH el cual permite un acceso seguro con el fin de poder hacer que la transmisión de datos sea mucho más fácil.

Instalación y preparación de la instancia para el respaldo automatizado

Primero se toma en cuenta que para este tipo de respaldos automatizado se toma en cuenta de que se debe de instalar docker en la instancia antes de realizar el procedimiento, después se deben de ejecutar la lista de comandos en los anexos **img.1.1** y **img.1.2** los cuales generarán una SSH key la cual se utilizara para conectar la instancia de AWS a el repositorio en GitHub, una vez generada la SSH key se procede a conectarse en el repositorio en el apartado de deploy keys **img.1.3** y **img.1.4**, se toma en cuenta de que una vez declarada la SSH key en git se debe de iniciar a la preparación de los scripts para la añadidura de estos en el crontab **img.1.5**, **img.1.8** siendo estos ejemplos manejados en el contenido de los scripts llamados uploadgit.sh y backup.sh, una vez redactados los scripts se agregan en el crontab con un tiempo predefinido con el fin de que estos procesos sean automatizados **img.1.7** una vez declarados estos scripts se generarán los resultados tanto en la instancia, así como en el repositorio **img.1.6**, **img.1.10** finalizando así el proceso de automatización de archivos.

Procesos de administración y operativos

Manejo de incidentes de seguridad

Ataque de fuerza bruta

Simulación de ejemplo

Aquí se simula un ataque de fuerza bruta mediante el uso de un gestor gráfico de base de datos en donde se trata de adivinar la contraseña de una instancia de base de datos **img.2.2** aquí se ve que el acceso fue denegado al usuario mal intencionado.

Contraataque de protección

A continuación se ve el proceso en donde se bloquean las ips no autorizadas mediante el uso de la administración de correcciones **img.2.7** se verá que de manera predeterminada hay una conexión mediante el uso del protocolo ssh en el puerto 22, para bloquear todas las ips no autorizadas procederemos a agregar una ip que si es autorizada a conexión **img.2.8** con esto se logra que cualquier otra ip que no esté autorizada no pueda conectarse de cualquier manera, el incidente dejó en el apartado de logs los datos de la conexión no autorizada **img.2.3** aquí es en donde se analizan los logs de conexiones los cuales nos indican la dirección de la ip del atacante y la manera de identificarlo para verificar que no haya hecho un mayor daño en alguna otra instancia, esto se realiza mediante el uso de la documentación de este evento en donde se registran las especificaciones de dicho suceso.

Borrado de tabla de manera incidental

Simulación de ejemplo

Aquí se muestra como un empleado conectado a la base de datos mediante un gestor gráfico, al final borrando los datos el usuario ve como la tabla ha sido vaciada teniendo un incidente de seguridad que compromete la integridad de los datos **img.2.4**.

Contramedida de protección

Como contramedida de protección para este incidente se toma en cuenta de que utilizan los respaldos en el repositorio el cual está conectado a la base de datos **img.1.10** con esto se procede a descargar el primer respaldo que se encuentra el cual contiene un script que genera la base de datos que se encontraba en la instancia de

base de datos ***img.2.5*** al final el usuario como un método común ejecuta este script en el gestor gráfico de la base de datos logrando la integración de la contramedida para la restauración completa de la base de datos.

Anexos

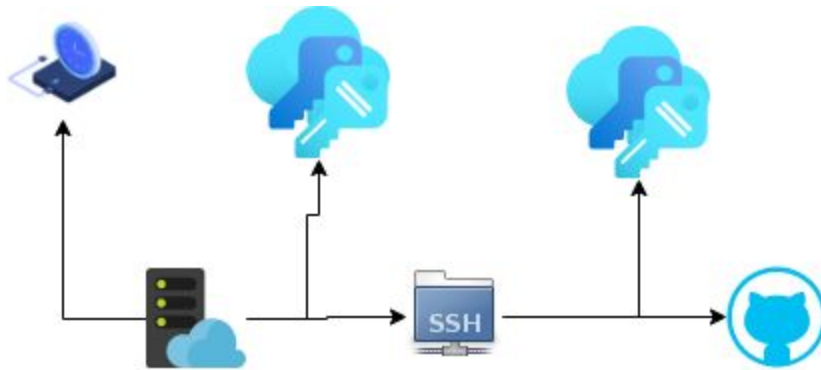


Diagrama de conexión de github a la nube mediante SSH img.1.0

```

Respaldo_comandos.md
--- Una vez planteada la instancia de Ubuntu en con el contenedor de mariadb ejecutar los siguientes comandos
--- Inicializar la SSH key en la instancia de Ubuntu
ssh-keygen -t rsa

--- Aparecera una serie de preguntas en conjunt con arte ascii de una aleta

Generating public/private rsa key pair.
Enter file in which to save the key (/home/demo/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/demo/.ssh/id_rsa.
Your public key has been saved in /home/demo/.ssh/id_rsa.pub.
The key fingerprint is:
4a:dd:0a:c6:35:4e:3f:ed:27:38:8c:74:44:4d:93:67 demo@a
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .oo.      |
|      . o.E     |
|      + . o     |
|      . = .     |
|      = S =     |
|      o + = +    |
|      . o + o .  |
|      . o        |
|      +-----+

```

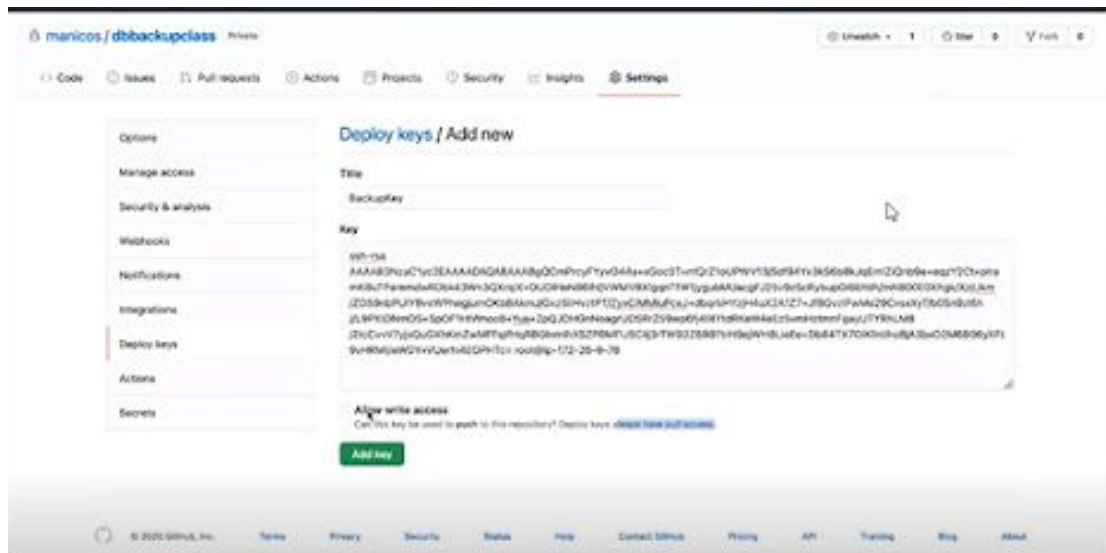
Comandos de ejecución en la instancia en amazon para generar el SSH img.1.1

```

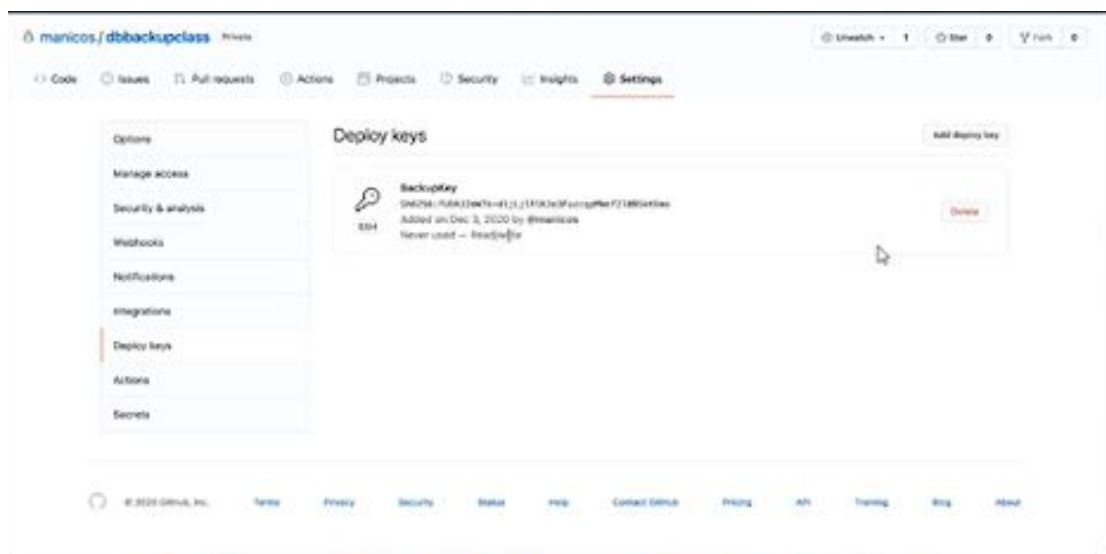
27 +-----+
28
29 --- En la linea de comandos podremos copiar la ssh key que se encuentra en la carpeta /home/demo/.ssh/id_rsa.pub
30 -- Se copia esta llave mediante el comando que marca la copia de la key y la ip de la instancia
31 ssh-copy-id demo@198.51.100.0
32
33 --- como alternativa se puede tomar en cuenta de que se puede abrir el archivo de terminación .pub abriendolo con cat
34 cat ~/.ssh/id_rsa.pub | ssh demo@198.51.100.0 "mkdir -p ~/.ssh && chmod 700 ~/.ssh && cat >> ~/.ssh/authorized_keys"
35
36
37 -- Aparecera el siguiente mensaje en la linea de comandos
38 The authenticity of host '198.51.100.0 (198.51.100.0)' can't be established.
39 RSA key fingerprint is b1:2d:33:67:ce:35:4d:5f:f3:a8:cd:c0:c4:48:86:12.
40 Are you sure you want to continue connecting (yes/no)? yes
41 Warning: Permanently added '198.51.100.0' (RSA) to the list of known hosts.
42 user@198.51.100.0's password:

```

Comandos de ejecución en la instancia en amazon para generar el SSH segunda parte img.1.2



creación de la conexión por SSH mediante el uso de la clave generada en la instancia mediante la declaración de la clave img.1.3



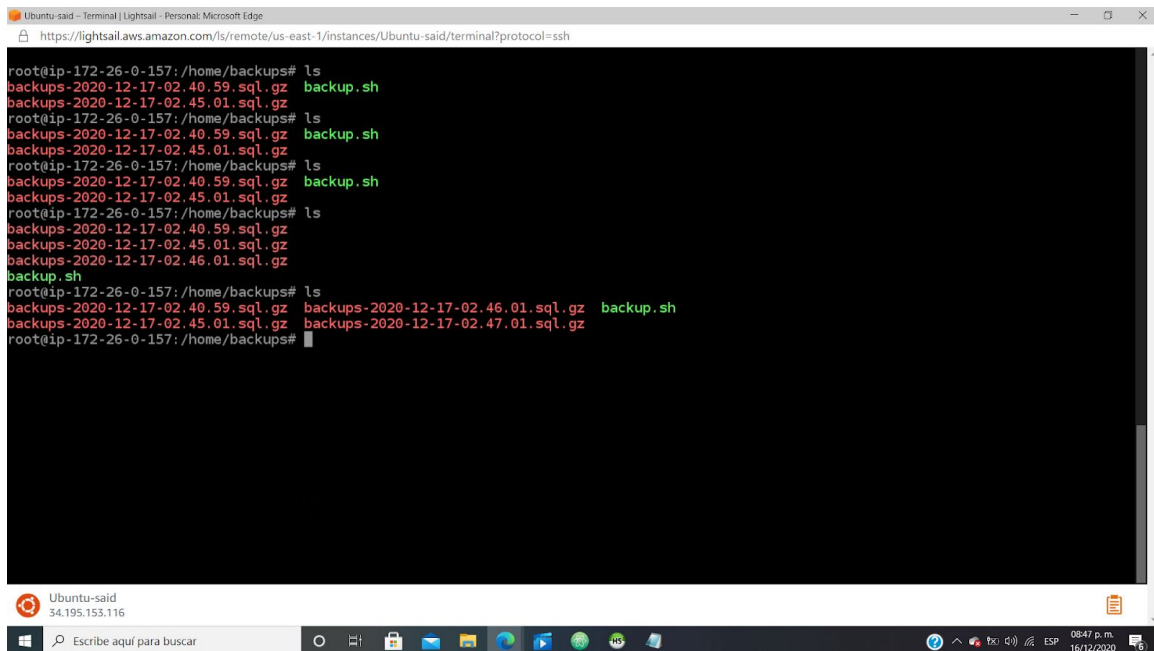
demonstración de clave SSH creada en git img.1.4

```
git.sh
1  cd /home/dbbackup
2  git add .
3  git commit -m 'Daily backup'
4  git push origin master
5
```

ejemplo del script que se ejecutara en el crontab para la subida de los respaldos a la nube de git img.1.5

```
crontab.md
1  */1 * * * * /home/backups/backup.sh
2  */2 * * * * /home/backups/uploadgit.sh
3
```

ejemplo de la escritura dentro del crontab de la instancia para los scripts de la instancia que se encargan de los respaldos dum y de la subida de estos a un repositorio de git img.1.6

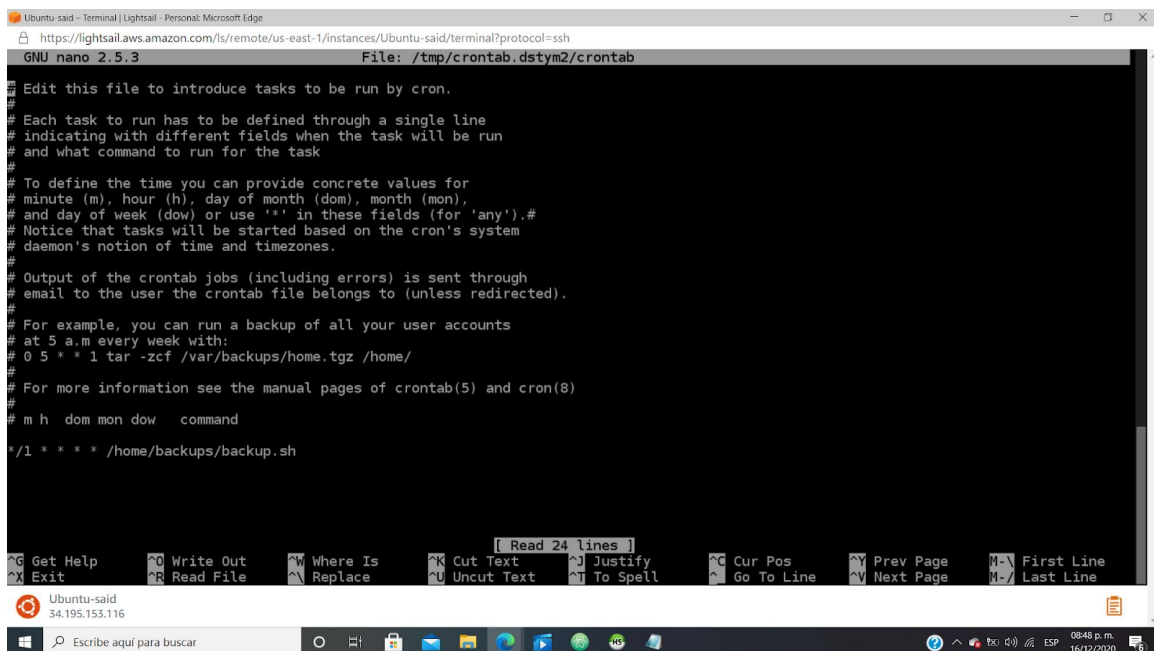


```

root@ip-172-26-0-157: /home/backups# ls
backups-2020-12-17-02.40.59.sql.gz backup.sh
backups-2020-12-17-02.45.01.sql.gz
root@ip-172-26-0-157: /home/backups# ls
backups-2020-12-17-02.40.59.sql.gz backup.sh
backups-2020-12-17-02.45.01.sql.gz
root@ip-172-26-0-157: /home/backups# ls
backups-2020-12-17-02.40.59.sql.gz backup.sh
backups-2020-12-17-02.45.01.sql.gz
root@ip-172-26-0-157: /home/backups# ls
backups-2020-12-17-02.40.59.sql.gz
backups-2020-12-17-02.45.01.sql.gz
backups-2020-12-17-02.46.01.sql.gz
backup.sh
root@ip-172-26-0-157: /home/backups# ls
backups-2020-12-17-02.40.59.sql.gz backups-2020-12-17-02.46.01.sql.gz backup.sh
backups-2020-12-17-02.45.01.sql.gz backups-2020-12-17-02.47.01.sql.gz
root@ip-172-26-0-157: /home/backups#

```

evidencia del resultado de los respaldos automatizados mediante el crontab y la escritura de un script con terminación sh img.1.6



```

GNU nano 2.5.3 File: /tmp/crontab.dstym2/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/1 * * * * /home/backups/backup.sh

```

evidencia de ejemplo real de la modificación del crontab para la declaración de los scripts img.1.7

```

GNU nano 2.5.3 File: backup.sh
/usr/bin/docker exec dockermariadb_db_1 mysqldump --user root --password=123456 e-commerce | gzip > /home/backups/backups-$(date +%Y-%m-%$

```

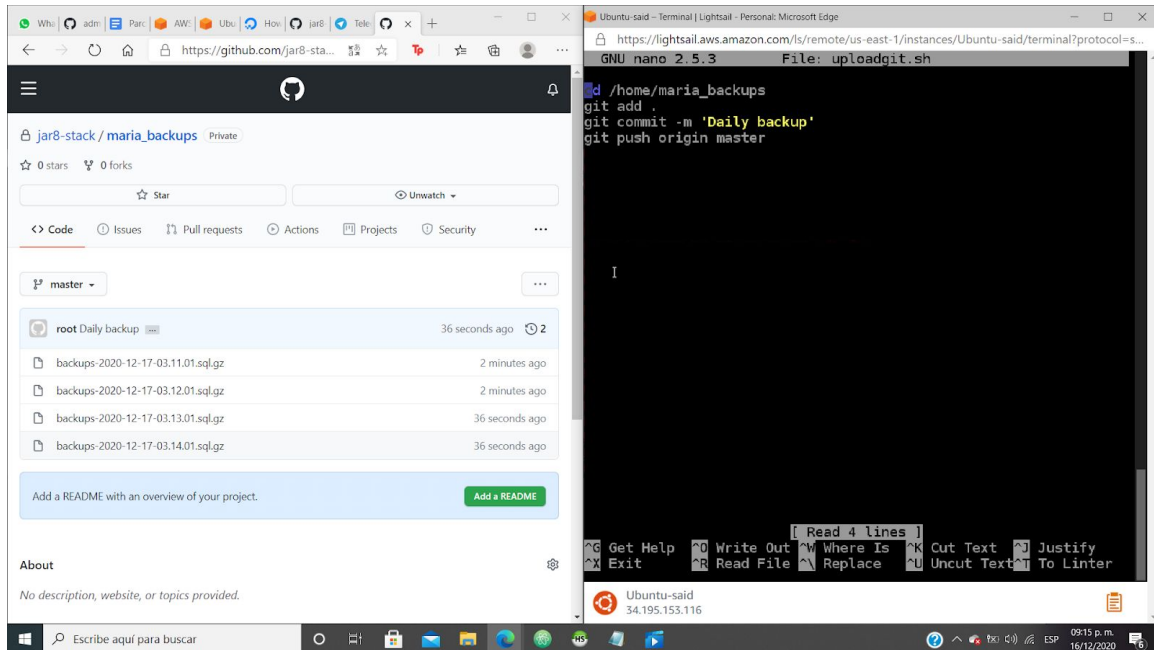
contenido de un script para el respaldo dum que sera agregado en el crontab para su automatización img.1.8

```

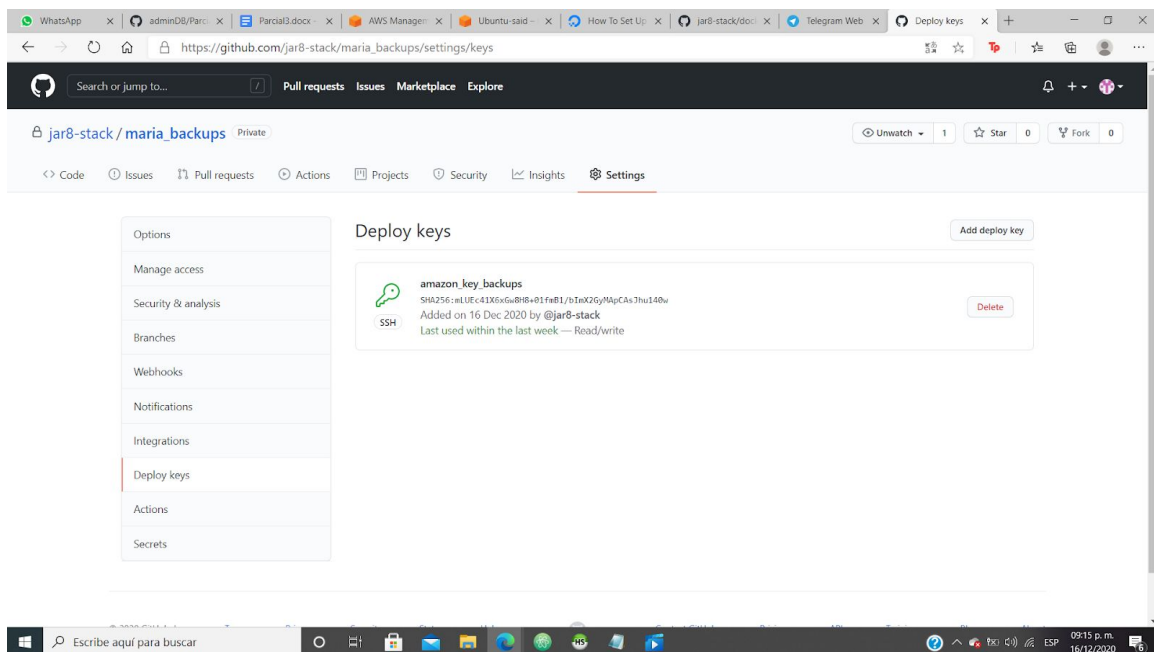
root@ip-172-26-0-157: /home/backups# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): Enter pas
sphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:mLUEc4lX6xGw8H8+0lfbB1/bImX2GyMApCAsJhu140w root@ip-172-26-
0-157
The key's randomart image is:
+---[RSA 2048]---+
|.o..o o+.oo|
|oo o..+oo. o|
|ooE..ooo o|
|.+. =.o.|
|o oS.o.*+|
|..*B=|
|..=+0|
|..+*|
|.
+---[SHA256]-----+
root@ip-172-26-0-157: /home/backups#

```

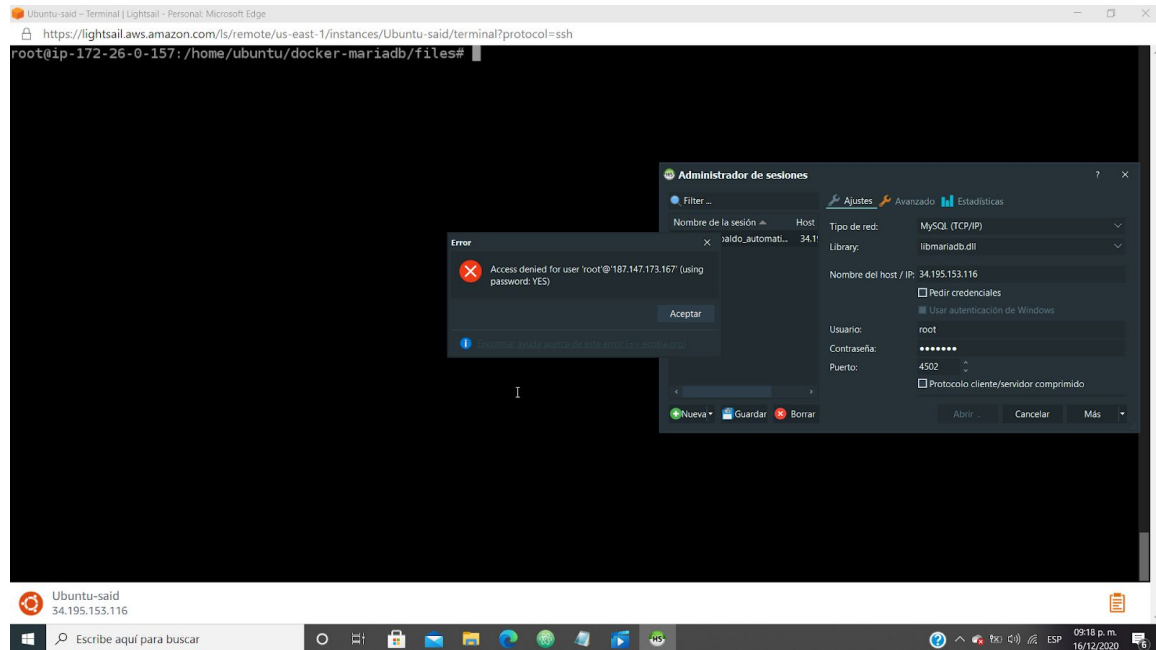
generación de la clave ssh mediante el uso de comandos de ssh-keygen -t rsa img.1.9



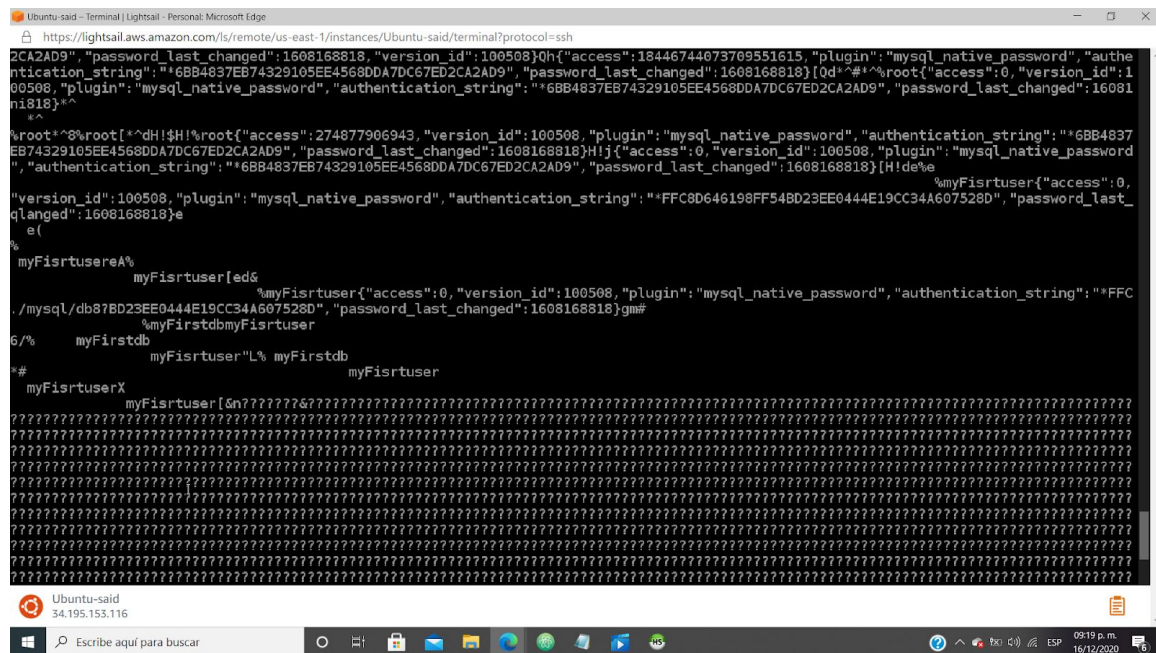
evidencia del contenido de un script para la subida automatizada de archivos a un repositorio en git y su resultado en ejecución debido a su añadidura en el crontab img.1.10



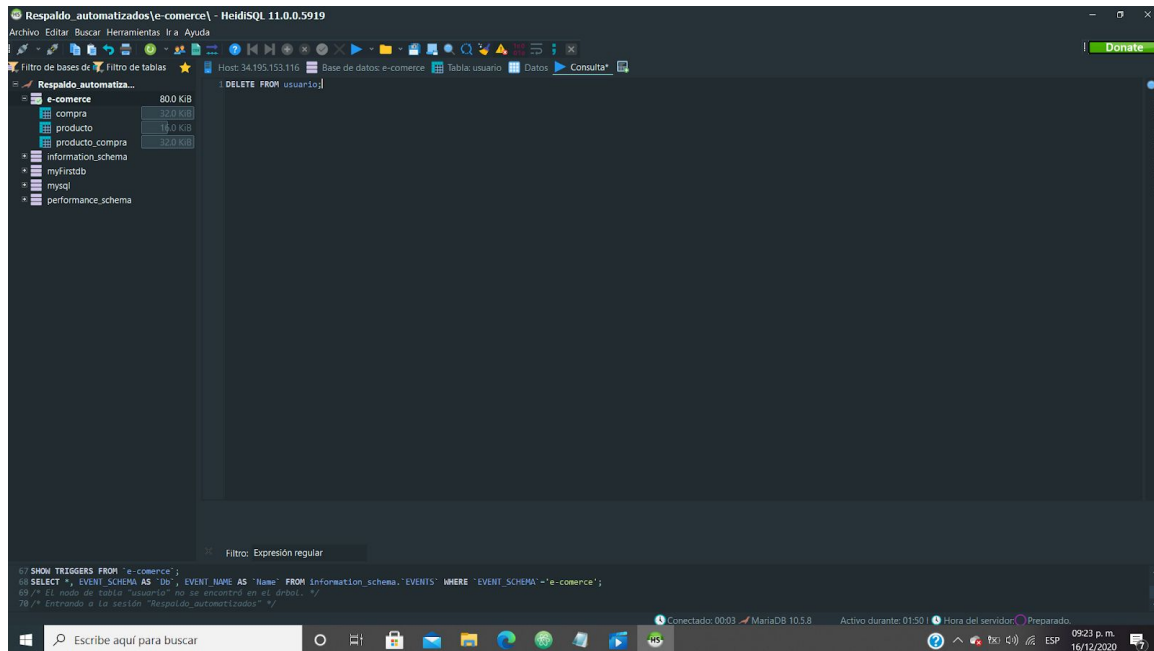
llave ya instaurada en el repositorio img.2.1



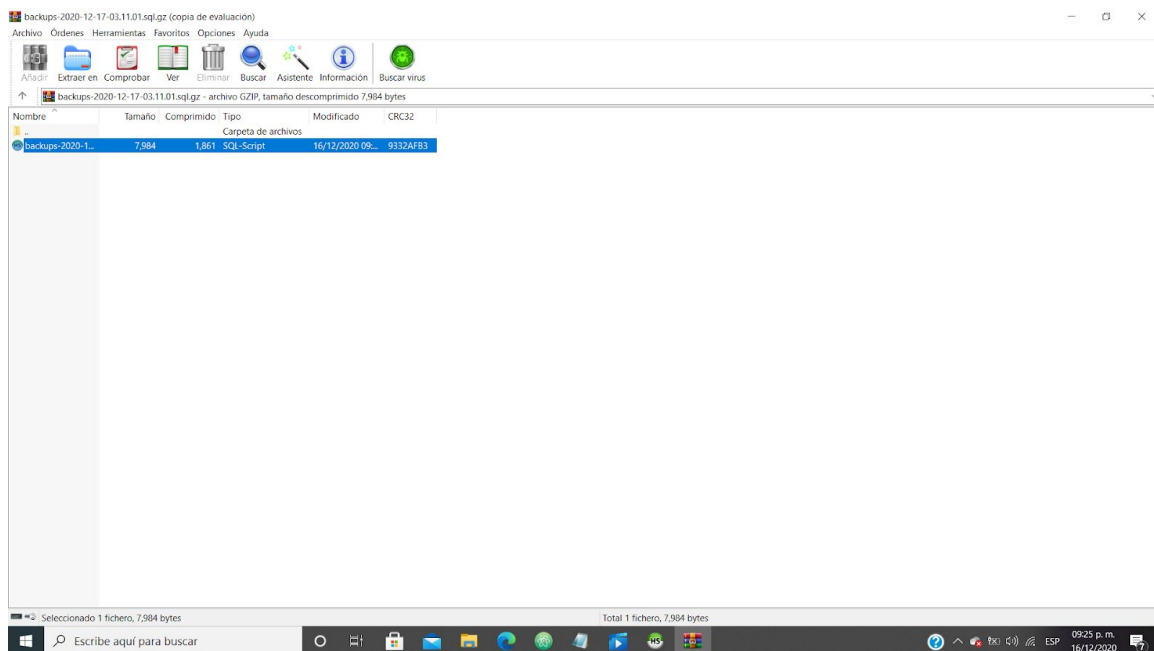
ejemplo de ataque de fuerza bruta por acceso de claves de base de datos img.2.2



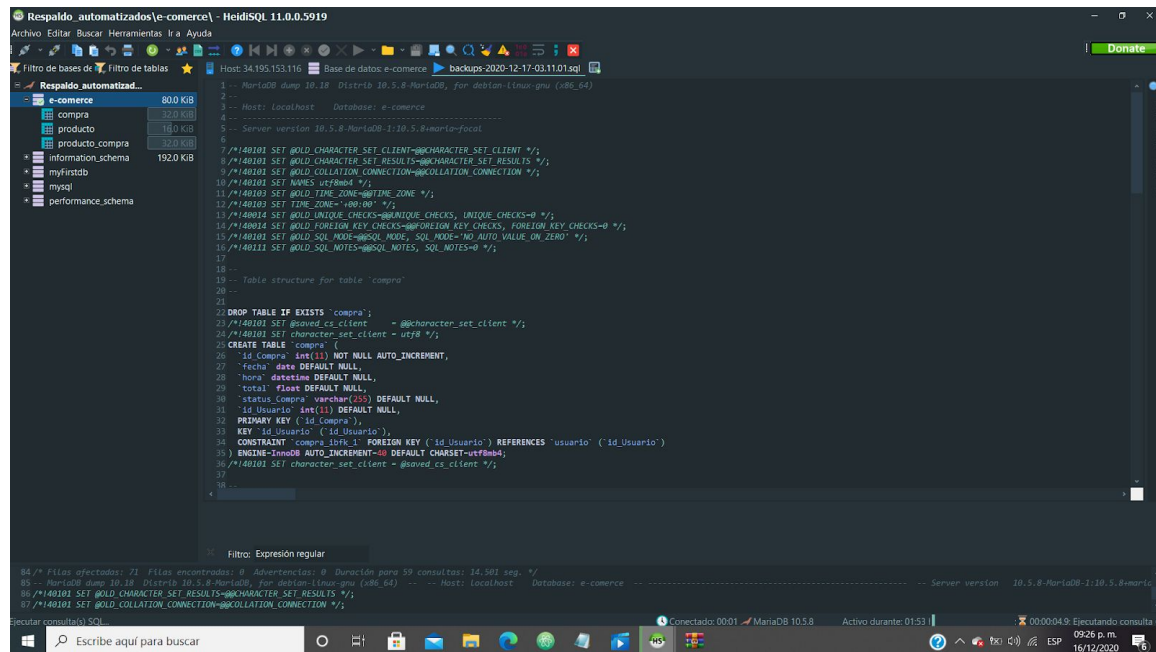
ejemplo de analisis de logs del ataque de fuerza bruta con claves de base de datos
img.2.3




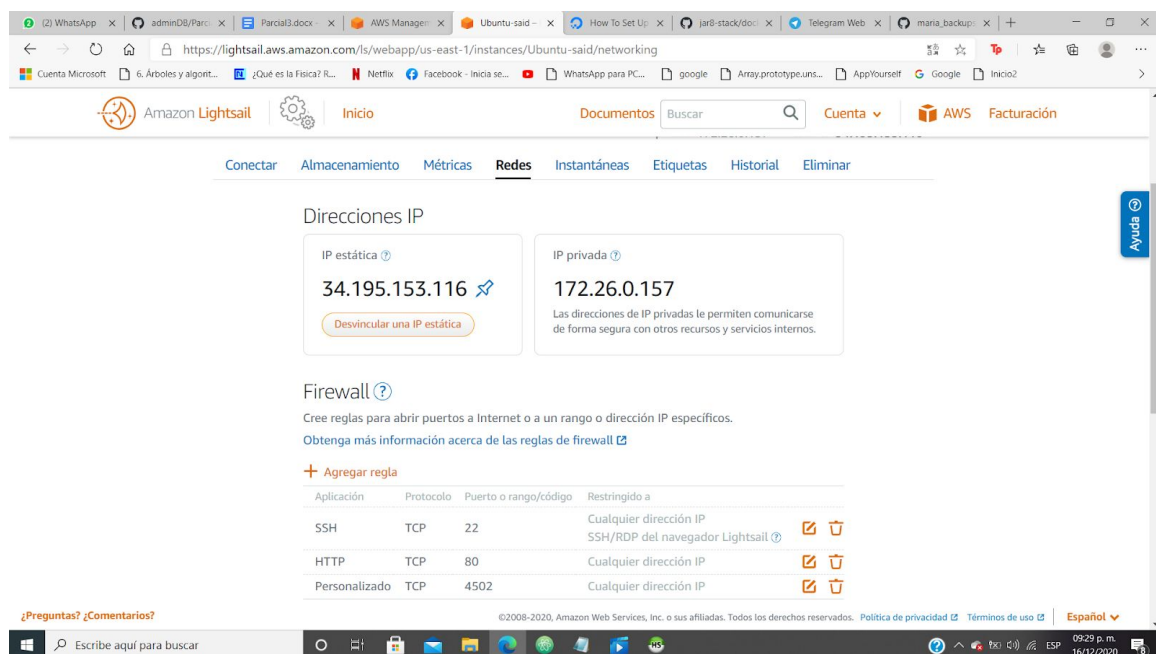
ejemplo de borrado accidental de datos de una tabla img.2.4



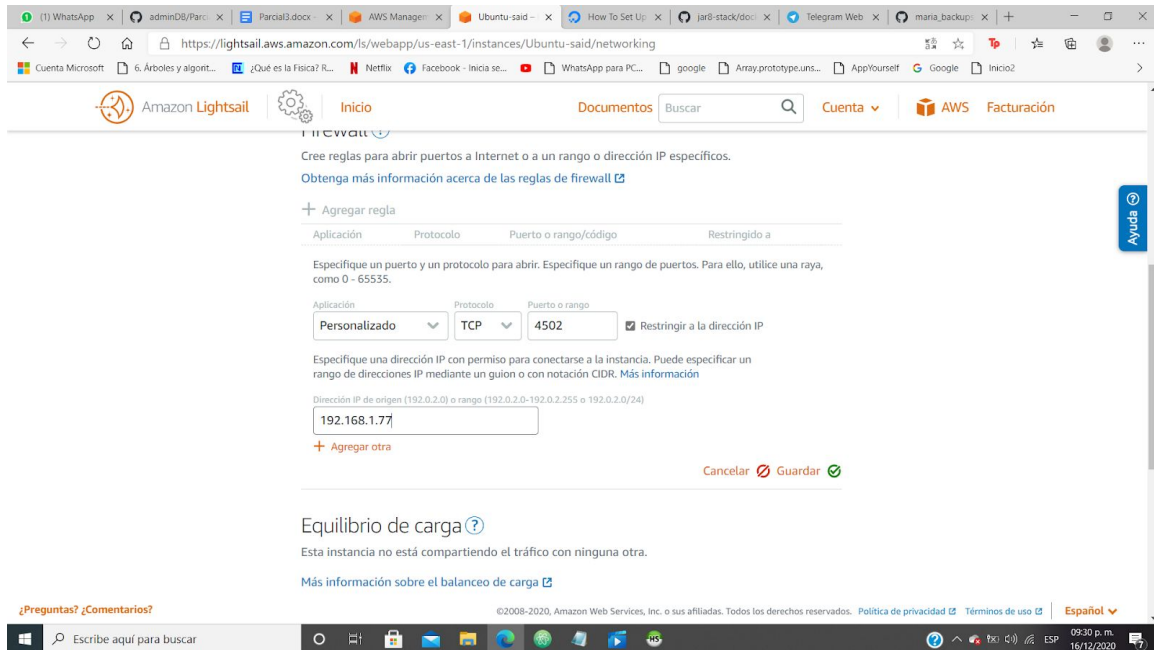
archivo proveniente del repositorio en donde se encuentran los respaldos img.2.5



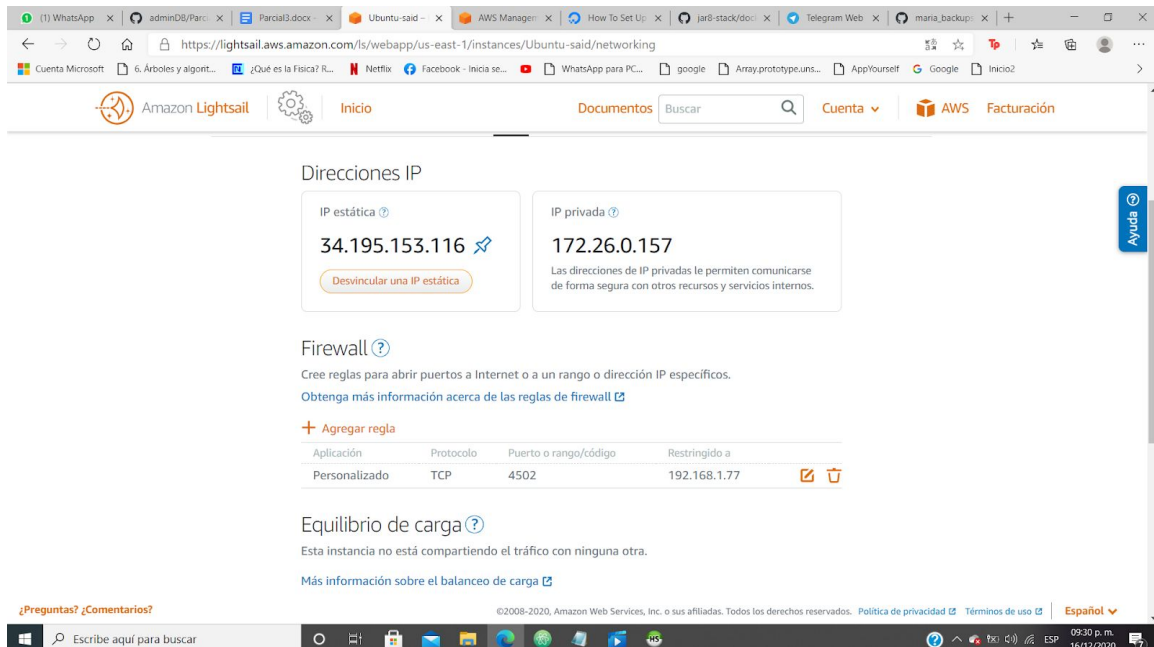
ejemplo de restauración simple de una base de datos afectada



instauración de conexiones en la instancia de AWS img.2.7



personalización de las conexiones para que la instancia se conecte a una sola ip a fin de fortalecer la conexión y bloquear las direcciones ip externas img.2.8



resultado de establecimiento de una sola dirección ip img.2.9