Chirpstack Backup and Restoration:
Redis & Postgres

NOTE: It is recommended to go through the main Chirpstack Guide on setting up for ubuntu before going through this backup and restore document to better understand the procedure. Installation and procedure was set up for Debian or Ubuntu servers.

//before doing any backups make sure the appropriate "backups" directory exists and its 2 sub directories "redisbackups" and "postgresbackups", if they do not, create them using the mkdir command

PostgreSQL:

https://www.postgresql.org/docs/12/backup.html

//before running the command using chron a '.pgpass' file must be made so the passwords can be used without a user having to enter them

To begin create a new file called .pgpass

```
bash$ vi .pgpass // use vim or nano to create then edit the file
//then add the contents to the file like this
#hostname:port:database:username:password  //the passwords will be that of the postgres user

localhost:5432:chirpstack_as:postgres:postgres_user_password
localhost:5432:chirpstack_ns:postgres:postgres_user_password
localhost:5432:postgres:postgres:postgres_user_password
localhost:5432:template0:postgres:postgres_user_password
localhost:5432:template1:postgres:postgres_user_password
```

//once the file is made with the correct passwords save and exit using ctrl X C then :wq in vim

//now you can put the command into cron
//open cron using :
bash$ sudo crontab -e //edits the current crontab

//if it is the first time opening chron it will ask which editor you prefer, so select the one normally used

//then add this command to the crontab:

```
0 0 * * * pg_dumpall -c --no-password -h localhost -U postgres > /root/backups/postgresbackup/pgbackup.out
```

//the 0 0 * * * indicates to run the command at 0 minutes on the 0 hour (aka midnight) every day
Do this command to allow cron to run in the background:

```
sudo systemctl enable cron
```

File can now be found in their associated directories

//once the backup is made you can restore it using the command:

```
$ psql -h localhost -U postgres < /root/backups/postgresbackups/pgbackup.out
```

Command Lines:

Update Database:

```
$ psql -h localhost -U (postgres user) < (file)
```

example:

```
$ psql -h localhost -U postgres < /root/backups/postgresbackups/pgbackup.out
```

Additional Command Lines If Needed:

Create Single Database Dump:

```
$ pg_dump -C -h localhost -U (role) (dbname) -f (filename.sql)
```

//dumps only one specified database, -C cleans the old databases out when you do a restore, -h specifies the host, and -U specifies the logon user

example:

```
$ pg_dump -C -h localhost -U chirpstack_as chirpstack_as -f chirpstack_asdump.sql
```

Enter Postgres:

```
$ sudo -i -u postgres
```

Enter Postgres Command Line:

```
$ sudo -i -u postgres psql
```

Redis:

https://redis.io/topics/persistence
https://www.digitalocean.com/community/tutorials/how-to-back-up-and-restore-your-redis-data-on-ubuntu-14-04#:~:text=So%2C%20you%20can%20back%20up,sammy%2Fredis%2Dbackup%2D001

//redis is not like postgres where you have to use a postgres command to create a dump file, redis literally runs off of its rdb.dump file (in our configuration, others use .rdb and .aof) and the backup one will replace the current dump file when restored

Command Lines:

Open cron :

```
sudo crontab -e
```

Add these commands:

```
0 0 * * * cp /var/lib/redis/dump.rdb /root/backups/redisbackup/dump.rdb
```

Do this command to allow cron to run in the background:

```
sudo systemctl enable cron
```

File can now be found in their associated directories

Redis restore procedure

```
root@hostname:~# cd backups //change dir to backups
root@hostname:~/backups# cd redisbackups //change dir to redisbackups to verify
root@hostname:~/backups/redisbackups# ls //redis dump file is there
dump.rdb
root@hostname:~/backups/redisbackups# vi dump.rdb //check contents of file and verify it's not empty and exit vim using :q
root@hostname:~/backups/redisbackups# sudo service redis-server stop //stop redis db
root@hostname:~/backups/redisbackups# redis-cli //confirm it is stopped
Could not connect to Redis at IPADDRESS: Connection refused
not connected>
root@hostname:~# sudo mv /var/lib/redis/dump.rdb /var/lib/redis/dump.rdb.old //use the mv command to change the current dump file name to something else(dump.rdb.old for example)
root@hostname:~# sudo cp -p /root/backups/redisbackups/dump.rdb /var/lib/redis //copy the backup redis dump file into the redis directory
root@hostname:~# ls -la /var/lib/redis/ //check the file permissions of the redis files
total 436
drwxr-x---  2 redis redis   4096 Jun  3 21:56 .
drwxr-xr-x 42 root  root    4096 Jun  2 23:17 ..
-rw-r-----  1 root  root  146545 Jun  3 20:35 dump.rdb //notice how the new dump.rdb has root root it needs to be redis redis
-rw-rw----  1 redis redis     92 Jun  2 19:49 dump.rdb.old
-rw-rw----  1 redis redis 146160 Jun  3 21:10 dump.rdb.old2
-rw-rw----  1 redis redis 138615 Jun  3 21:53 dump.rdb.old6-3
root@hostname:~# sudo chown redis:redis /var/lib/redis/dump.rdb //this command updates the permission
root@hostname:~# sudo chmod 660 /var/lib/redis/dump.rdb //and this command allows redis to write to the new file
root@hostname:~# ls -la /var/lib/redis/ //check the file permissions now and verify dump.rdb has redis redis not root root
total 436
drwxr-x---  2 redis redis   4096 Jun  3 21:56 .
drwxr-xr-x 42 root  root    4096 Jun  2 23:17 ..
-rw-rw----  1 redis redis 146545 Jun  3 20:35 dump.rdb //this one here
-rw-rw----  1 redis redis     92 Jun  2 19:49 dump.rdb.old
-rw-rw----  1 redis redis 146160 Jun  3 21:10 dump.rdb.old2
-rw-rw----  1 redis redis 138615 Jun  3 21:53 dump.rdb.old6-3
root@hostname:~# sudo service redis-server start //restart the redis server
root@hostname:~# redis-cli //try logging on to make sure its running
IPADDRESS> exit //no errors and the ip address shown means it's started
```

```
root@hostname:~#
```

Additional Command Lines:

Move File To New Server:

Push = $ scp ‘filename’ user@remotehost:~

Pull = $ scp user@remotehost:filename filenamechange

example:

```
$ scp backup.out root@remotehost: ~
```

```
$ scp root@remotehost:backups/postgresbackup/pgbackup.out pgbackup.out
```

```
$ scp root@remotehost:backups/redisbackup/dump.rdb dump.rdb
```

Additional Links:

Main Chirpstack Guide:
https://www.chirpstack.io/guides/debian-ubuntu/

Chirpstack Forum:
https://forum.chirpstack.io/t/backup-postgresql-and-redis-databases/1061

Cron:
https://www.linode.com/docs/tools-reference/tools/schedule-tasks-with-cron/

Drop Database:
https://www.postgresqltutorial.com/postgresql-drop-database/

Postgres Role and Password Change:
https://chartio.com/resources/tutorials/how-to-set-the-default-user-password-in-postgresql/

SSH Without Password
https://unix.stackexchange.com/questions/214594/how-do-i-scp-a-file-from-server-a-to-server-b-from-cron