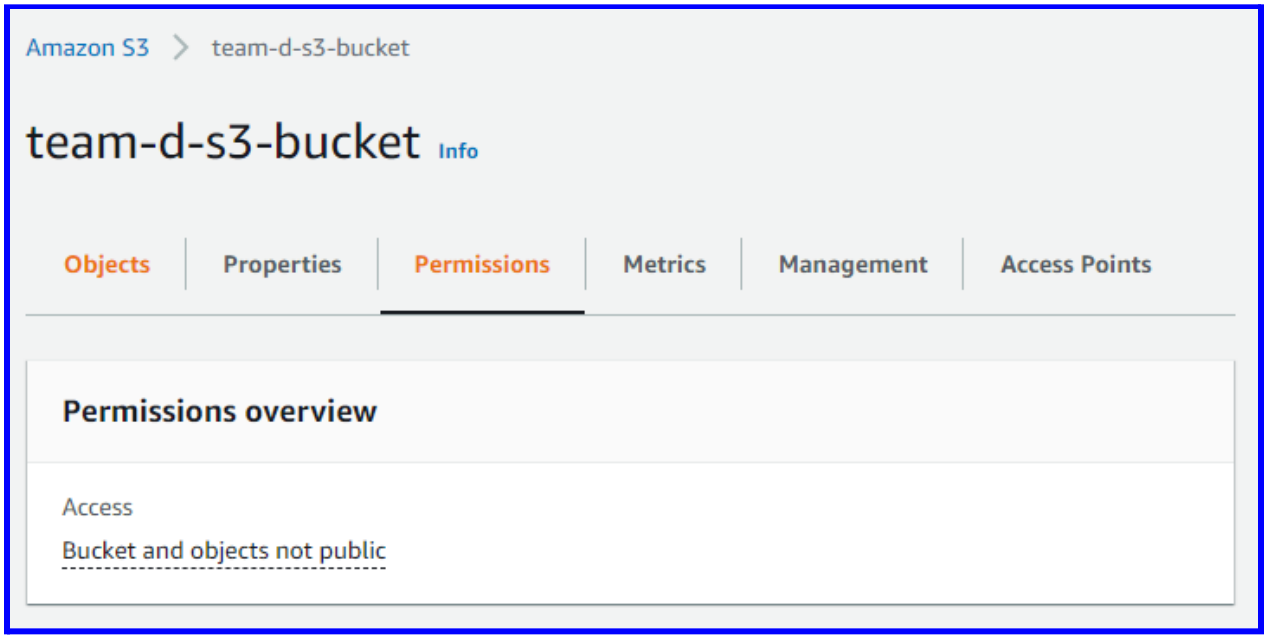
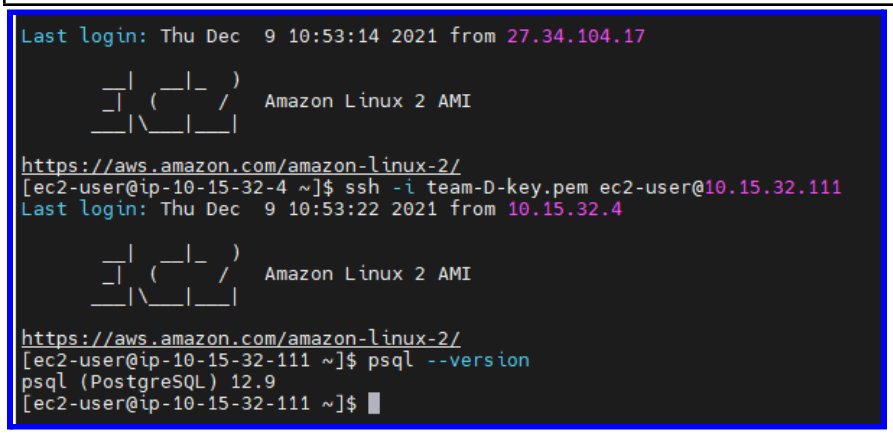
Write a script that backs up an SQL dump and uploads it to an S3 Bucket.
The contents of the S3 bucket should not be accessible via public.

S3 Bucket
We have already created ***S3-bucket*** *(***Private***)* **- team-d-s3-bucket** *Using this bucket to store our* **SQL Dump File.**



Bash Script to backup a PostgreSQL database
We have already installed **PostgreSQL 12** in **EC2(10.15.32.111)** (Private Subnet)



Adding some data into lf_technology database

```
[ec2-user@ip-10-15-32-111 ~]$ psql -U postgres -d lf_technology
Password for user postgres:
psql (12.9)
Type "help" for help.

lf_technology=# CREATE TABLE team_d( ID serial PRIMARY KEY, NAME CHAR(250) NOT NULL );
CREATE TABLE
lf_technology=# \dt
         List of relations
 Schema |  Name  | Type  |  Owner
--------+--------+-------+----------
 public | team_d | table | postgres
(1 row)

lf_technology=# select * from team_d;
 id | name
----+------
(0 rows)

lf_technology=#
```

```
lf_technology=# INSERT INTO team_d (name) VALUES ('Bibek');
INSERT 0 1
lf_technology=# select * from team_d;
 id |                                                    name
----+------------------------------------------------------------------
  1 | Bibek

(1 row)

lf_technology=#
```

```
lf_technology=# INSERT INTO team_d (name) VALUES ('Prerit');
INSERT 0 1
lf_technology=# \q
```

Creating Profile lft-training and using it with AWS cli

aws configure --profile lft-training

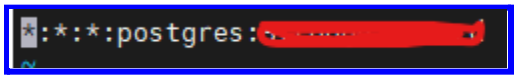
```
[ec2-user@ip-10-15-32-111 ~]$ aws configure --profile lft-training
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]
Default region name [None]: us-east-2
Default output format [None]:
[ec2-user@ip-10-15-32-111 ~]$
```

Creating .pgpass in the home directory of the ec2-user to pass password

cd ~

vi .pgpass

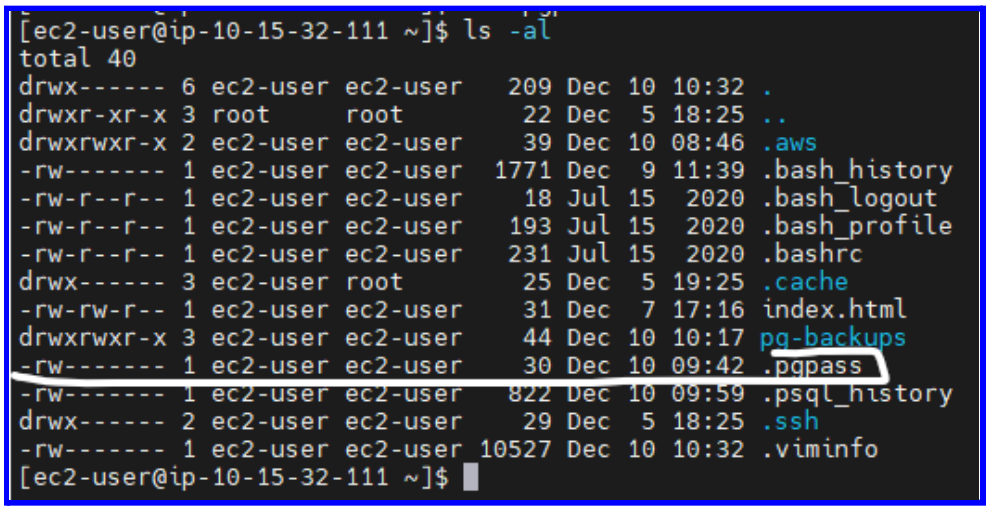
In format ***host:port:dbname:user:password***



Now while using ***pg_dump*** *trough script, it will* ***fetch password from .pgpass***

Giving .pgpass 600 permission so the owner can only read and write to this file

chmod 600 .pgpass



Creating pg-dump-S3.sh file to take dump file and store them in AWS S3

cd pg-backups

mkdir dumpfiles - *to store dumpfiles in this directory*

vi pg-dump-S3.sh

```
#!/bin/bash

#assigning a name to sql file with date (Month,day,Hour,Minute and seconds)
dump=pgdump_EC2_$( date +"%m%d_%H%M_%S")

#taking dump file of lf_technology
pg_dump -U postgres -d lf_technology >> /home/ec2-user/pg-backups/dumpfiles/$dump.sql

#uploading to aws s3 team-d-s3-bucket
aws s3 cp /home/ec2-user/pg-backups/dumpfiles/$dump.sql s3://team-d-s3-bucket/$dump.sql --profile lft-training
```

Save & Exit

Giving .sh file executive permission
chmod +x pg-dump-S3.sh

```
[ec2-user@ip-10-15-32-111 pg-backups]$ ls
dumpfiles  pg-dump-S3.sh
[ec2-user@ip-10-15-32-111 pg-backups]$ ll
total 4
drwxrwxr-x 2 ec2-user ec2-user  41 Dec 10 10:19 dumpfiles
-rwxrwxr-x 1 ec2-user ec2-user 379 Dec 10 10:46 pg-dump-S3.sh
[ec2-user@ip-10-15-32-111 pg-backups]$
```

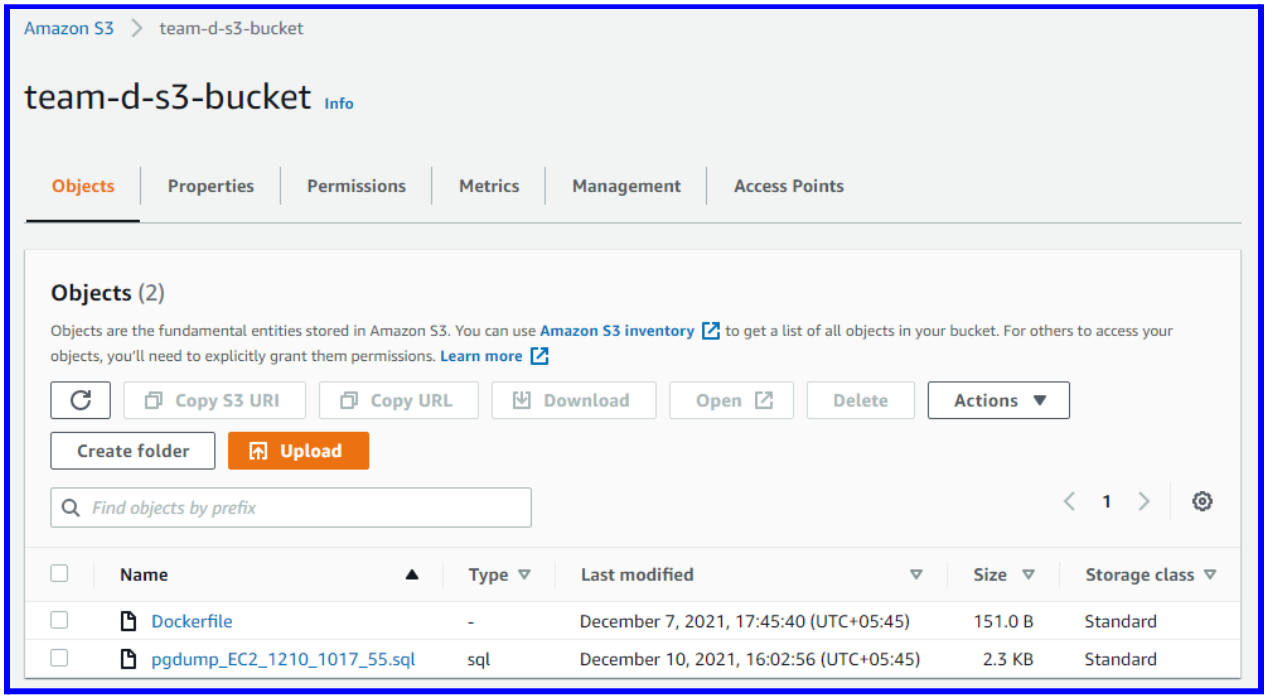
Running pg-dump-S3.sh file
./pg-dump-S3.sh

```
[ec2-user@ip-10-15-32-111 pg-backups]$ vi pg-dump-S3.sh
[ec2-user@ip-10-15-32-111 pg-backups]$ ./pg-dump-S3.sh
upload: dumpfiles/pgdump_EC2_1210_1017_55.sql to s3://team-d-s3-bucket/pgdump_EC2_1210_1017_55.sql
```

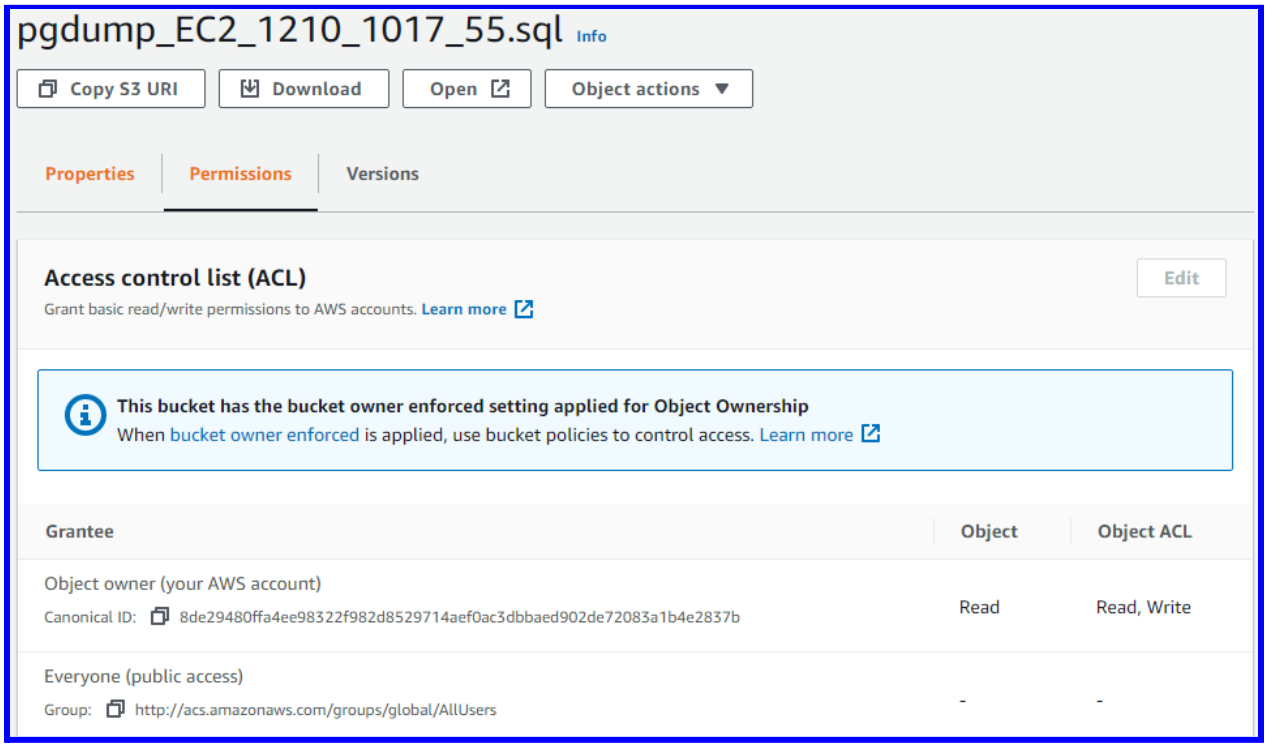
We can see a dump file in the dumfiles directory

```
[ec2-user@ip-10-15-32-111 dumpfiles]$ ls
pgdump_EC2_1210_1017_55.sql
[ec2-user@ip-10-15-32-111 dumpfiles]$
```

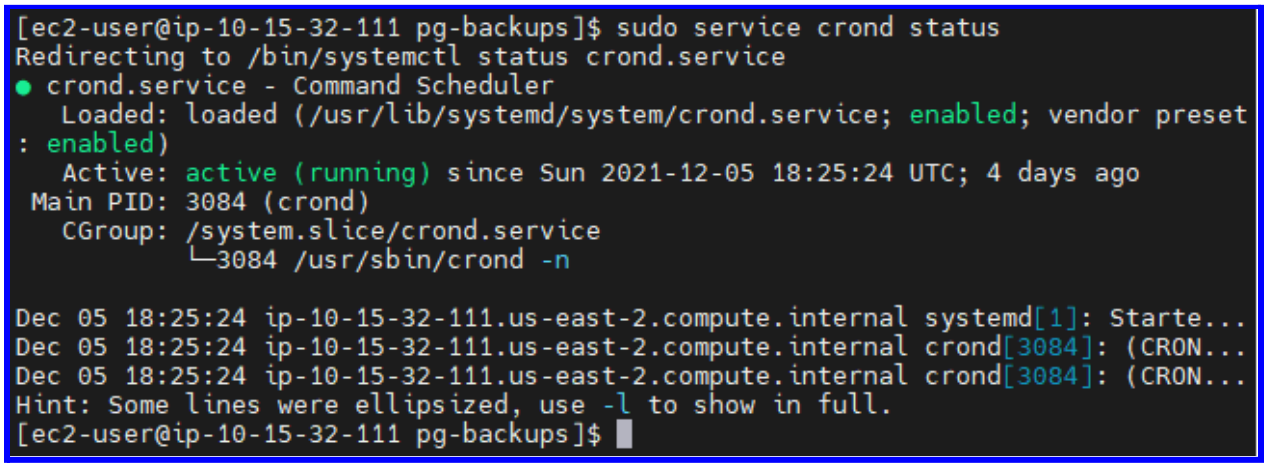
Also it has stored dumpfile to the team-d-s3-bucket



And the dumpfile is not publicly accessible

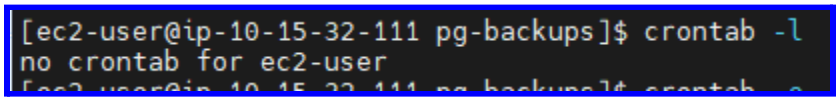


Testing cron job to perform backup

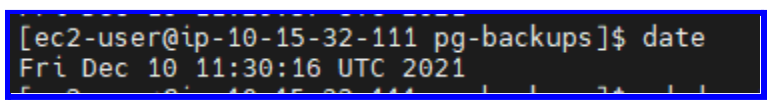


crond is successfully installed and running

Initially there is no job assigned



Checking the date



Creating cron job to run script at 11:36

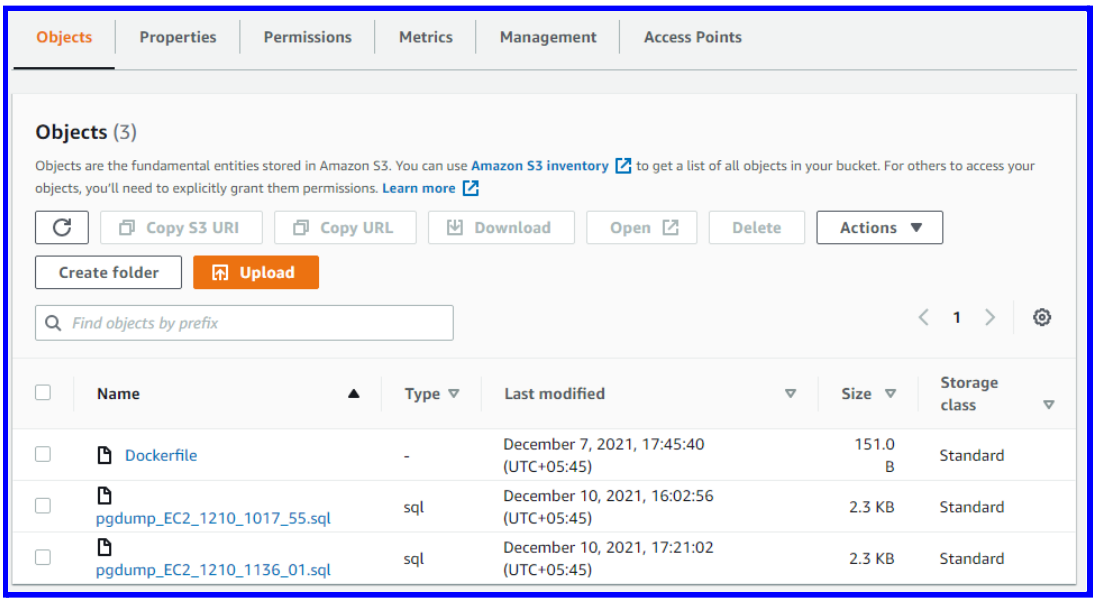
crontab -e

```
#min(0-59),Hour(0-23),Day_of_Month(1-31),Month(1-12),Day_of_Week(0-6)
36 11 * * * ~/pg-backups/pg-dump-S3.sh
```

At 11 36 script file runs automatically and store a dump file in local

```
[ec2-user@ip-10-15-32-111 pg-backups]$ ls
dumpfiles  pg-dump-S3.sh
[ec2-user@ip-10-15-32-111 pg-backups]$ cd dumpfiles/
[ec2-user@ip-10-15-32-111 dumpfiles]$ ls
pgdump_EC2_1210_1017_55.sql  pgdump_EC2_1210_1130_01.sql  pgdump_EC2_1210_1136_01.sql
[ec2-user@ip-10-15-32-111 dumpfiles]$
```

And also stores the dump file to s3 bucket



To set the cron job to run 3 times a day

crontab -e

```
#min(0-59),Hour(0-23),Day_of_Month(1-31),Month(1-12),Day_of_Week(0-6)

0 3,11,23 * * * ~/pg-backups/pg-dump-S3.sh
```

It will run the script at <u>3 AM, 11 AM and 11 PM daily</u>

```
[ec2-user@ip-10-15-32-111 dumpfiles]$ crontab -e
crontab: installing new crontab
[ec2-user@ip-10-15-32-111 dumpfiles]$ crontab -l
#min(0-59),Hour(0-23),Day_of_Month(1-31),Month(1-12),Day_of_Week(0-6)

0 3,11,23 * * * ~/pg-backups/pg-dump-S3.sh
[ec2-user@ip-10-15-32-111 dumpfiles]$
```