

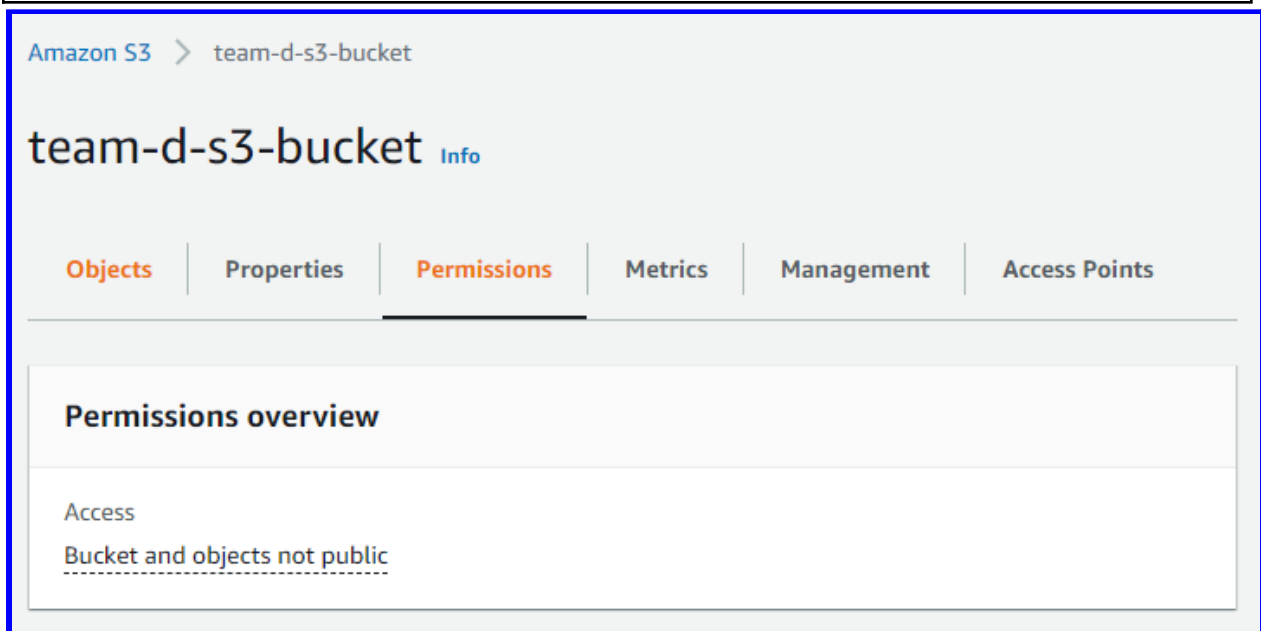
Write a script that backs up an SQL dump and uploads it to an S3 Bucket.

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)

```
Last login: Thu Dec  9 10:53:14 2021 from 27.34.104.17

 _ _ | ( _ _ | _ )
 _ _ | \ _ _ | _ _ | Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-15-32-4 ~]$ ssh -i team-D-key.pem ec2-user@10.15.32.111
Last login: Thu Dec  9 10:53:22 2021 from 10.15.32.4

 _ _ | ( _ _ | _ )
 _ _ | \ _ _ | _ _ | Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-15-32-111 ~]$ psql --version
psql (PostgreSQL) 12.9
[ec2-user@ip-10-15-32-111 ~]$
```

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

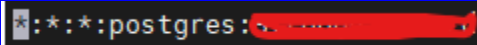
```
[ec2-user@ip-10-15-32-111 ~]$ aws configure --profile lft-training
AWS Access Key ID [None]: 
AWS Secret Access Key [None]: 
Default region name [None]: us-east-2
Default output format [None]: 
[ec2-user@ip-10-15-32-111 ~]$
```

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

cd ~

vi .pgpass

In format host:port:dbname:user:password

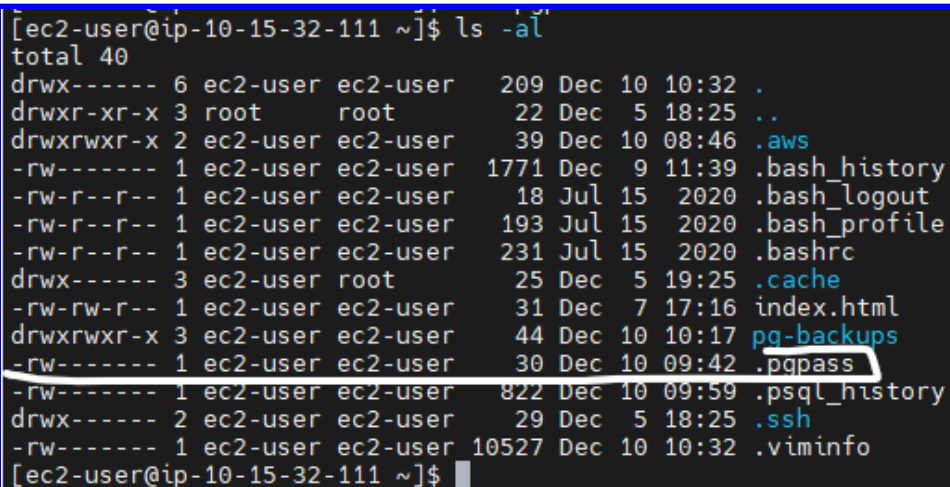


```
host:port:postgres:ec2-user:password
```

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



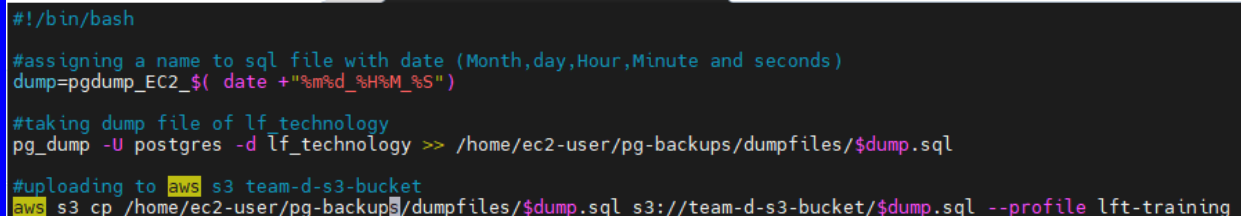
```
[ec2-user@ip-10-15-32-111 ~]$ ls -al
total 40
drwx----- 6 ec2-user ec2-user 209 Dec 10 10:32 .
drwxr-xr-x 3 root root 22 Dec 5 18:25 ..
drwxrwxr-x 2 ec2-user ec2-user 39 Dec 10 08:46 .aws
-rw----- 1 ec2-user ec2-user 1771 Dec 9 11:39 .bash_history
-rw-r--r-- 1 ec2-user ec2-user 18 Jul 15 2020 .bash_logout
-rw-r--r-- 1 ec2-user ec2-user 193 Jul 15 2020 .bash_profile
-rw-r--r-- 1 ec2-user ec2-user 231 Jul 15 2020 .bashrc
drwx----- 3 ec2-user root 25 Dec 5 19:25 .cache
-rw-rw-r-- 1 ec2-user ec2-user 31 Dec 7 17:16 index.html
drwxrwxr-x 3 ec2-user ec2-user 44 Dec 10 10:17 pg-backups
-rw----- 1 ec2-user ec2-user 30 Dec 10 09:42 .pgpass
-rw----- 1 ec2-user ec2-user 822 Dec 10 09:59 .psql_history
drwx----- 2 ec2-user ec2-user 29 Dec 5 18:25 .ssh
-rw----- 1 ec2-user ec2-user 10527 Dec 10 10:32 .viminfo
[ec2-user@ip-10-15-32-111 ~]$
```

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-backup[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

Amazon S3 > team-d-s3-bucket

team-d-s3-bucket Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI Copy URL Download Open Delete

Create folder Upload

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Dockerfile	-	December 7, 2021, 17:45:40 (UTC+05:45)	151.0 B	Standard
<input type="checkbox"/>	pgdump_EC2_1210_1017_55.sql	sql	December 10, 2021, 16:02:56 (UTC+05:45)	2.3 KB	Standard

And the dumpfile is not publicly accessible

pgdump_EC2_1210_1017_55.sql [Info](#)

[Copy S3 URI](#) [Download](#) [Open](#) [Object actions](#) ▼

[Properties](#) [Permissions](#) [Versions](#)

Access control list (ACL) [Edit](#)

Grant basic read/write permissions to AWS accounts. [Learn more](#)

This bucket has the bucket owner enforced setting applied for Object Ownership
When **bucket owner enforced** is applied, use bucket policies to control access. [Learn more](#)

Grantee	Object	Object ACL
Object owner (your AWS account) Canonical ID: 8de29480ffa4ee98322f982d8529714aef0ac3dbbaed902de72083a1b4e2837b	Read	Read, Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	-	-

Testing cron job to perform backup

```
[ec2-user@ip-10-15-32-111 pg-backups]$ sudo service crond status
Redirecting to /bin/systemctl status crond.service
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-12-05 18:25:24 UTC; 4 days ago
     Main PID: 3084 (crond)
        CGroup: /system.slice/crond.service
                └─3084 /usr/sbin/crond -n

Dec 05 18:25:24 ip-10-15-32-111.us-east-2.compute.internal systemd[1]: Starte...
Dec 05 18:25:24 ip-10-15-32-111.us-east-2.compute.internal crond[3084]: (CRON...
Dec 05 18:25:24 ip-10-15-32-111.us-east-2.compute.internal crond[3084]: (CRON...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-10-15-32-111 pg-backups]$
```

crond is successfully installed and running

Initially there is no job assigned

```
[ec2-user@ip-10-15-32-111 pg-backups]$ crontab -l
no crontab for ec2-user
[ec2-user@ip-10-15-32-111 pg-backups]$ crontab -e
```

Checking the date

```
[ec2-user@ip-10-15-32-111 pg-backups]$ date
Fri Dec 10 11:30:16 UTC 2021
```

Creating cron job to run script at 11:36

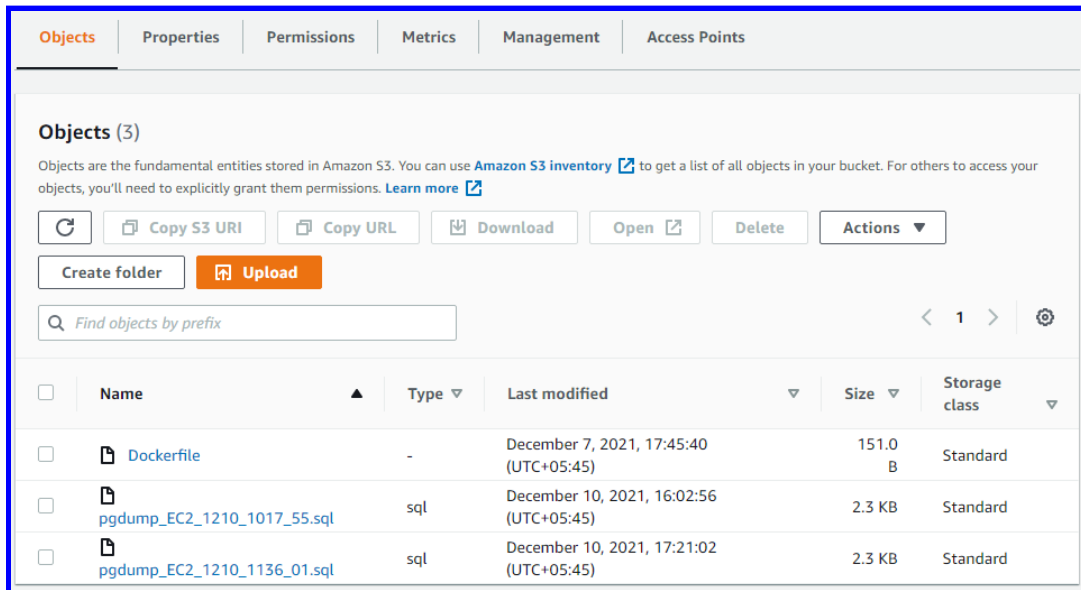
crontab -e

```
#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



Objects	Properties	Permissions	Metrics	Management	Access Points
Objects (3) Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more					
<div> Copy S3 URI Copy URL Download Open Delete </div> <div> Create folder Upload</div> <div><input type="text" value="Find objects by prefix"/></div> <div>< 1 > </div>					
<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Dockerfile	-	December 7, 2021, 17:45:40 (UTC+05:45)	151.0 B	Standard
<input type="checkbox"/>	pgdump_EC2_1210_1017_55.sql	sql	December 10, 2021, 16:02:56 (UTC+05:45)	2.3 KB	Standard
<input type="checkbox"/>	pgdump_EC2_1210_1136_01.sql	sql	December 10, 2021, 17:21:02 (UTC+05:45)	2.3 KB	Standard

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 3 AM, 11 AM and 23 AM daily

```
[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]$
```