

Postgres-Backup

Backing up of stage Prod3 Postgres periodically.

We are doing regular backups to azure blob storage using the following script

Pre-flight checks

1. To authenticate with azure we should give azure environment variables in the script as shown below
1. Copy the key from azure UI by following this path
1. Install the Azure CLI on Linux

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

```
echo -e "[azure-cli]
name=Azure CLI
baseurl=https://packages.microsoft.com/yumrepos/azure-cli
enabled=1
gpgcheck=1
gpgkey=https://packages.microsoft.com/keys/microsoft.asc" | sudo tee
/etc/yum.repos.d/azure-cli.repo
```

```
sudo yum install azure-cli
```

To fetch the access keys.

To fetch the access keys we need to login to azure portal Storage accounts argoidinfrabackupstorage container.

Click on Access keys, we can view the access keys by clicking on show keys.

Dashboard > Storage accounts > argoidinfrabackups

argoidinfrabackups | Access keys

Storage account

Search (Cmd+/) << Hide keys Set rotation reminder Refresh

Events

Storage browser (preview)

Data storage

- Containers
- File shares
- Queues
- Tables

Security + networking

- Networking
- Azure CDN
- Access keys**
- Shared access signature
- Encryption
- Security

Data management

- Geo-replication

Access keys authenticate your applications' requests to this storage account. Keep your keys in a secure location like Azure Key Vault, and replace them often with new keys. The two keys allow you to replace one while still using the other.

Remember to update the keys with any Azure resources and apps that use this storage account. [Learn more](#)

Storage account name
argoidinfrabackups

key1

Last rotated: 04/08/2021 (92 days ago)

Rotate key

Key
UillmJy48TPldgOCWXQAI9ovSEJCg5lqzJw24ZBra4VFYgOeo458HB+OdkLy43w0l9...

Connection string
DefaultEndpointsProtocol=https;AccountName=argoidinfrabackups;AccountKey=...

key2

Last rotated: 04/08/2021 (92 days ago)

Rotate key

Key
jHvcEyaySb4S+1q9CYqaaj/Yql7Z7oHVUULLm99O/syZWyl8ykApPMb8TamXOva7g...

Connection string

Home>argoidinfrabackupstorage >Access keys>Show keys

1. The key found in it must be stored in the

AZURE_STORAGE_KEY property

1. keep the environment variables in crontab as shown below

```
#SHELL=/bin/bash

##export azure properties
AZURE_STORAGE_KEY="DtaTVqgLGLEcT/x4
/FvOn8Bc7ygzg5KbGLCKE2j7ga4G82Y9qUtkZdEAhgcM0ORjR5rFgtdLUdzQ+AStM30EYQ==
"
AZURE_STORAGE_AUTH_MODE="key"
AZURE_STORAGE_ACCOUNT="argoidinfrabackupstorage"
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;
AccountName=argoidinfrabackupstorage;AccountKey=DtaTVqgLGLEcT/x4
/FvOn8Bc7ygzg5KbGLCKE2j7ga4G82Y9qUtkZdEAhgcM0ORjR5rFgtdLUdzQ+AStM30EYQ==
;EndpointSuffix=core.windows.net"
```

Step1:-Scripts to backup

Placed the scripts in /opt/postgres_backup_scripts/postgres_backup.sh in stage Prod3 instance 10.1.0.5 VM

Here we are backing up the database Config_Service_SaaS_Console_2_1_0 while logging in as a user postgres

```
#!/bin/bash
#created by vasu
##created on 2021-09-06 ist

date=$(date '+%Y-%m-%d_%H:%M:%S')
pg_dump_backup_dir=/opt/postgres_backup/pg_dump/pg_dump_$date.sql
postgres_container=postgres-backup
azure_backup=/opt/postgres_backup/
docker_container_postgres=debezium_postgres-12
logrotate_script=/opt/postgres_backup/scripts/logrotate_postgres.sh

##"Backup the postgres database"
docker exec $docker_container_postgres pg_dump -U postgres
Config_Service_SaaS_Console_2_1_0 >> $pg_dump_backup_dir

## "azure syncing with azure blob"
az storage azcopy blob sync --container $postgres_container -s
$azure_backup

## "calling the log rotation script"
bash "$logrotate_script"
```

Step2:-Logrotation script

Also configured log rotation script and calling the script at the end

```
#!/bin/bash
##created by vasu
##created on 28-05-2021
#filepath=<path>
delfilepath=/opt/postgres_backup/
#deldirpath=<path>

#To delete the .sql files older than 5 days
find $delfilepath -mtime +5 -type f -delete
```

The backups will be copied into the azure blobs(argoind-infra-backups)

Step3:-New instance restoration

we have to install the postgres by using the following documentation in new instance

[PostgreSQL Installation](#)

Step4:-Download the backups

we can download the backups from Home>Storage accounts>argoindinfrabackupstorage>

Step5:-Postgres data upgradation

Go to the latest file and download the Postgres database and data
we can use the following command to restore the Postgres data

First, we should clear the Postgres database using the following commands

```
DROP SCHEMA postgres CASCADE;  
drop schema public cascade;  
CREATE SCHEMA public;
```

then,

we need to copy the latest `pg_dump.sql` file into the docker container

```
sudo docker cp pg_dump.sql debezium_postgres-12:./
```

we should be care ful while executing this command sice It will dump all the backup data in the database `Config_Service_SaaS_Console_2_1_0` we can change the database name accordingly

```
sudo docker exec -it debezium_postgres-12 psql -U postgres -d  
Config_Service_SaaS_Console_2_1_0 -f pg_dump.sql
```

Step6:-Restart the docker containers

We need to restart the docker containers

Step7:- Final checks

1. We need to check whether the data and schema is copied into the new Postgres
2. Check for the data sources