

Airflow-Backup

Backing up of Prod3 Airflow 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
UillmJy48TPIgdwOCWXQAI9ovSEJCg5lqzJw24ZBra4VFYgOeo458HB+OdkLy43w0I9...

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"

30 02 * * * root /opt/airflow_backup_scripts/airflow_backup.sh
```

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 /opt/airflow and /data/airflow directories

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

date=$(date '+%Y-%m-%d_%H_%M_%S')
airflow_backup_opt_dir=/opt/airflow_backup/airflow_dir
/airflow_opt_${date}.gz
airflow_backup_data_dir=/opt/airflow_backup/data_dir/airflow_data_${date}.
gz
airflow_container=airflow-backup
azure_backup=/opt/airflow_backup/
docker_container_secret=airflow_server_1
docker_container_postgres=airflow_postgres_1
logrotate_script=/opt/airflow_backup_scripts/logrotate_script

##"Backup the /opt/airflow directory"
cd /opt/; tar -C /opt/ -cvzf $airflow_backup_opt_dir airflow

##"Backup the airflow data directory"
cd / && tar -cvzf $airflow_backup_data_dir data

## "azure syncing with azure blob"
az storage azcopy blob sync --container $airflow_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)