

Redash-backup

Backing up of stage fun2 redash 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 argoidinfrabackups 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
UillmJy48TPlgdwOCWXQAI9ovSEJCg5lqzJw24ZBra4VFYgOeo458HB+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>argoidinfrabackups >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="Uil*****D2PwlwpoOVovoRiA=="
AZURE_STORAGE_AUTH_MODE="key"
AZURE_STORAGE_ACCOUNT="argoidinfrabackups"
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;
AccountName=argoidinfrabackups;
AccountKey=UilI*****0I9LEhD2PwlwpoOVovoRiA==;
EndpointSuffix=core.windows.net"
```

Step1:-Scripts to backup

Placed the scripts in /opt/redash_backup_scripts in stage fun2 redash instance 192.0.1.28 VM

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

date=$(date '+%Y-%m-%d_%H:%M:%S')
pg_dump_backup_dir=/opt/redash_backup/pg_dump/pg_dump_${date}.sql
redash_secret_key_dir=/opt/redash_backup/redash_secret_key
/redash_secret_key_${date}
redash_container=redash-stage-backups
azure_backup=/opt/redash_backup/
docker_container_secret=redash_server_1
docker_container_postgres=redash_postgres_1
logrotate_script=/root/logrotate_redash.sh

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

## "Backup redash secret key"
docker inspect $docker_container_secret | grep REDASH_SECRET_KEY >>
$redash_secret_key_dir

## "azure syncing with azure blob"
az storage azcopy blob sync --container $redash_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/redash_backup/
#deldirpath=<path>

#To gzip the files older than 2 days
for f in $(find $filepath -name "*.log" -mtime +2 -type f) ;
do
    # filename=`echo $f | awk -F '/' '{print $NF}'`;
    gzip $f
    echo "$f"
done

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

#To delete the directories older than 9 days
#find $deldirpath -mtime +9 -type d -empty -delete
```

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

Step3:-New instance restoration

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

[Redash Installation](#)

Step4:-Download the backups

we can download the backups from Home>Storage accounts>argoidinfrabackups>redash-stage-backups

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 redash_postgres_1:./
```

```
sudo docker exec -it redash_postgres_1 psql -U postgres -d postgres -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