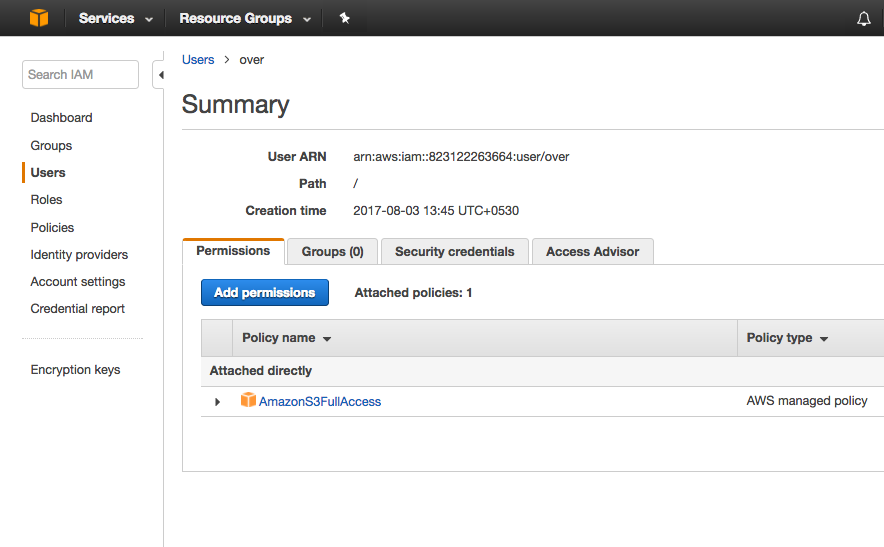
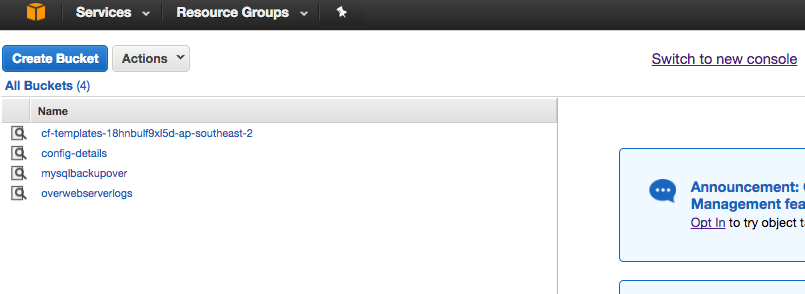
**s3-Upload-Apache2-Logs and Mysql-Database and Logs**

Step 1. First Create AWS IAM User with Limited Access to Only S3 ( Limited Access for Security ) and download it.



Step 2. Create Bucket



**Example: mysqlbackupover , overwebserverlogs**

**Step 3. Go to docker container**

**root@2dafe8e6402d:~# apt-get install s3cmd**

**Configure S3cmd Environment**

**root@2dafe8e6402d:~# s3cmd --configure**

Access key and Secret key are your identifiers for Amazon S3  
Access Key: xxxxxxxxxxxxxxxxxxxxxx  
Secret Key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
  
Encryption password is used to protect your files from reading  
by unauthorized persons while in transfer to S3  
Encryption password: xxxxxxxxxx  
Path to GPG program [/usr/bin/gpg]:  
  
When using secure HTTPS protocol all communication with Amazon S3  
servers is protected from 3rd party eavesdropping. This method is  
slower than plain HTTP and can't be used if you're behind a proxy  
Use HTTPS protocol [No]: Yes  
  
New settings:  
 Access Key: xxxxxxxxxxxxxxxxxxxxxx  
 Secret Key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
 Encryption password: xxxxxxxxxx  
 Path to GPG program: /usr/bin/gpg  
 Use HTTPS protocol: True  
 HTTP Proxy server name:  
 HTTP Proxy server port: 0  
  
Test access with supplied credentials? [Y/n] Y  
Please wait, attempting to list all buckets...  
Success. Your access key and secret key worked fine :-)  
  
Now verifying that encryption works...  
Success. Encryption and decryption worked fine :-)  
  
Save settings? [y/N] y  
Configuration saved to '/root/.s3cf

**#### BASH SCRIPT TO Sync & Upload Apache2 Logs on S3 Bucket #####**

**root@2dafe8e6402d:~# cat apache2logss3.sh**

**#!/bin/bash**

**###### THIS WILL SYNC Apache2 logs to S3 ###########**

**/usr/bin/s3cmd sync -r /var/log/apache2/access.log s3://overwebserverlogs/**

**/usr/bin/s3cmd sync -r /var/log/apache2/error.log s3://overwebserverlogs/**

**#### BASH SCRIPT TO Take MySQL Database Dump on localserver #####**

**root@2dafe8e6402d:~# cat mysqldump.sh**

**#!/bin/bash**

**# Database credentials**

**user="root"**

**password="root@123"**

**host="127.0.0.1"**

**db\_name="icinga"**

**# Other options**

**backup\_path="/var/backups/mysql"**

**date=$(date +"%d-%b-%Y")**

**# Set default file permissions**

**umask 177**

**# Dump database into SQL file**

**mysqldump --user=$user --password=$password --host=$host $db\_name > $backup\_path/$db\_name-$date.sql**

**##### BASH SCRIPT TO Upload Mysqldump Database on S3 Bucket #####**

**root@2dafe8e6402d:~# cat mysqllogS3.sh**

**#!/bin/bash**

**###### THIS WILL SYNC logs to S3 ###########**

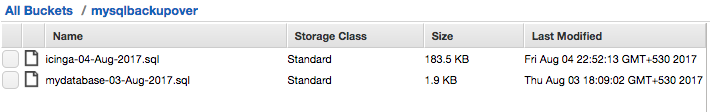
**/usr/bin/s3cmd sync -r /var/backups/mysql/ s3://mysqlbackupover/**

**####### CRONJOB TO RUN 7pm Daily ########3**

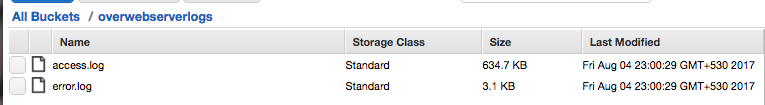
**0 19 \* \* \* sh ~/apache2logss3.sh [ Will upload apache2 logs on s3 ]**

**0 18 \* \* \* sh ~/mysqldump.sh [ Will take mysql Database Dump ]**

**0 19 \* \* \* sh~/mysqllogS3.sh [ Will upload mysql database dump on S3 ]**

****

**Apache2 LOGS On S3 Bucket**

****