**Automated Data backup by using crontab**

**Script NO. 1]**

Reference link :- https://www.folio3.com/5-easy-steps-on-scheduling-mysql-database-backup-using-cron/

Step 1] Folder and file creation

Create a folder anywhere, let’s say: /var/**backup**/, preferably outside the web root to avoid accidental access by anyone besides you.

Now, create a file with executable permission with the name “**script.sh”** in the folder created earlier.

You need to create another folder, let’s say “**db**”,where you will be storing your database backups using the following command:

sudo mkdir /var/backup/db

Step 2] Contenets in the file

# (1) set up all the mongodump variables

DATE=`date +"%d\_%b\_%Y\_%H%M"`

MONGOFILE=/var/lib/mongo/backup/db/db\_backup\_${DATE}.db

DATABASE=intelliconvo

USER=root

PASSWORD=rootpswd

# (2) in case you run this more than once a day,

# remove the previous version of the file

unalias rm 2> /dev/null

rm ${MONGOFILE} 2> /dev/null

rm ${MONGOFILE}.gz 2> /dev/null

# (3) do the mongodb database backup (dump)

sudo mongodump -u ${USER} -p${PASSWORD} ${DATABASE}|gzip > ${MONGOFILE}.gz

Step 3] verify script working correctly

cd /var/backup/

And execute the following command:

sudo ./script.sh

Step 4]scheduling execution with cron

Edit the server’s cron with the undermentioned command:

***sudo crontab -e***

Please note: you can exclude the ‘sudo’ command if you are logged in with ***root*** itself or the user you are logged in with has enough permissions to execute cron and create backup.

Enter the following line at the bottom of this file:

0 \* \* \* \* for every hours update automatically

**Script NO.2] simple script to take backup inside the instance**

#!/bin/bash

DATE=`date +"%d\_%b\_%Y\_%H%M"`

MONGOFILE='/home/ubuntu/workspace/devops/mongodb/db\_backup\_${DATE}.db'

HOST='localhost'

DATABASE='intelliconvo'

USER='root'

AUTH\_DB='admin'

PASSWORD='rootpswd'

CONNECTION\_STRING="mongodb://root:rootpswd@13.126.209.38:37017/?authMechanism=SCRAM-SHA-1&authSource=admin"

echo $CONNECTION\_STRING

mongodump -h $HOST -u ${USER} -p${PASSWORD} --authenticationDatabase $AUTH\_DB -d ${DATABASE} --out=${MONGOFILE}.gz

**Script NO 3] simple script to take backup inside the instance**

#!/bin/sh

DIR=`date +%y%m%d`

DEST=/home/ubuntu/workspace/devops/mongodb/$DIR

mkdir $DEST

mongodump -h localhost -d intelliconvo -u root -p rootpswd --authenticationDatabase admin -o $DEST

**Script NO 4] Taking backup local to aws s3 bucket**

#!/bin/bash

#Force file syncronization and lock writes

mongo admin --eval "printjson(db.fsyncLock())"

MONGODUMP\_PATH="/usr/bin/mongodump"

MONGO\_HOST="13.126.209.38" #replace with your server ip

MONGO\_PORT="37017"

MONGO\_DATABASE="intelliconvo" #replace with your database name

TIMESTAMP=`date +%F-%H%M`

S3\_BUCKET\_NAME="intelliconvo-qa" #replace with your bucket name on Amazon S3

S3\_BUCKET\_PATH="mongodb-backups"

# Create backup

$MONGODUMP\_PATH -h $MONGO\_HOST:$MONGO\_PORT -d $MONGO\_DATABASE

# Add timestamp to backup

mv dump mongodb-$HOSTNAME-$TIMESTAMP

tar cf mongodb-$HOSTNAME-$TIMESTAMP.tar mongodb-$HOSTNAME-$TIMESTAMP

# Upload to S3

s3cmd put mongodb-$HOSTNAME-$TIMESTAMP.tar

   s3://$S3\_BUCKET\_NAME/$S3\_BUCKET\_PATH/mongodb-$HOSTNAME-$TIMESTAMP.tar

#Unlock database writes

mongo admin --eval "printjson(db.fsyncUnlock())"

#Delete local files

rm -rf mongodb-\*

**Script NO 5] Taking backup of database from instance to AWS S3 bucket**

NOTE:- it’s working properly

Reference link :- <https://youtu.be/EXvRwOuGpQo>

**Step 1] install a AWS CLI on instance or server**

* sudo apt-get update
* sudo apt install awscli
* aws --version

Output

aws-cli/1.16.301 Python/3.7.6 Linux/5.4.0-9-generic botocore/1.13.37

* aws configure

Output

AWS Access Key ID [None]:

AWS secrete Access Key [None]:

Default region name [ap-south-1]:

Default output format [table]:

**Step 2] Create a folder for script like**

* mkdir mongo-backup-s3.sh
* vi mongo-backup-s3.sh

inside the file paste following script

# S3 bucket name

BUCKET=intelliconvo-qa/backup/ #required to change backet name

BACKUPBUCKET=intelliconvo-qa/backup/ #required to change backet name

# Linux user account

USER=ubuntu

# Backup directory

DEST=/home/$USER/workspace/devops/mongodb/

# Dump z2p & poststodos

DATE=`date +"%d\_%b\_%Y\_%H%M"`

MONGOFILE='/home/ubuntu/workspace/devops/mongodb/db\_backup\_${DATE}.db'

HOST='localhost'

DATABASE='intelliconvo' #required to change database name

USER='root' #required to change user name

AUTH\_DB='admin' # required to change auth name

PASSWORD='rootpswd' # required to change field

CONNECTION\_STRING="mongodb://root:rootpswd@13.126.209.38:37017/?authMechanism=SCRAM-SHA-1&authSource=admin" # required to change this field

echo $CONNECTION\_STRING

mongodump -h $HOST -u ${USER} -p${PASSWORD} --authenticationDatabase $AUTH\_DB -d ${DATABASE} --out=${MONGOFILE}.gz

# File name

TIME=`/bin/date --date='+5 hour 30 minutes' '+%d-%m-%Y-%I-%M-%S-%p'`

# Tar file of backup directory

TAR=$DEST/../$TIME.tar

# Create tar of backup directory

/bin/tar cvf $TAR -C $DEST .

# Upload tar to s3

/usr/bin/aws s3 cp $TAR s3://$BUCKET

#/usr/bin/aws s3 cp $TAR s3://$BACKUPBUCKET

# Remove tar file locally

/bin/rm -f $TAR

# Remove backup directory

/bin/rm -rf $DEST

# All done

echo "Backup available at <https://s3.amazonaws.com/$BUCKET/$TIM>..."

**Step 3] permissions change of the file**

* sudo chmod 777 mongo-backup-s3.sh # need to change the permission of file

**Step 5] run the script by using following command**

* sh mongo-backup-s3.sh OR
* ./mongo-backup-s3.sh OR
* bash mongo-backup-s3.sh

**Step 6] applying cron to the script to run automatically**

* crontab -e
* at the last of the page add following cron expression

0 \* \* \* \* /bin/bash /home/ubuntu/workspace/devops/mongodb/autogenarate-database-backup/mongo-backup-s3.sh

**Note :- for taking backup of database in every hrs used above cron expression otherwise you can change as per your requirement**