**SCENARIO-3**

**Aim:** Create a script in shell that verifies user credentials and take mysql backup from the cloud, archives it and move it to the backup folder in a scheduler that works every 12 hours.

**Pseudocode:**

* **Read Arguments:** Location of the backup directory and the MySQL Credentials file (Credentials file should have a hostname, port, username and password of the database server).
* **Check:** The Location of the MySQL Credentials file is valid. If true, read the credentials as local variables, else throw an error saying credentials not found and write to the log.
* **Run:** MySQL Dump command to retrieve all the data from MySQL. If it returns a non-zero exit code, then exit the program and write the error to the log.
* Redirect the stdout returned above to a new file in the backup directory. This file can be used to restore the database.
* **Check:** The script is scheduled in the crontab. If not, add the script to crontab such that it gets executed every 12 hours.

**Code:**

#!/bin/bash

if [ -z "$1" ]; then //checks whether the user has input the credentials or not

echo "ERROR: Credentials file not specified" >&2; exit 1; //&2 refers to stderr

elif [ -z "$2" ]; then //checks whether the user has input the backup directory or not

echo "ERROR: Backup directory not specified" >&2; exit 1;

fi

credentials\_file=$(realpath $1) //realpath prints the resolved absolute file name

backup\_directory=$(realpath $2)

if [ ! -f "$credentials\_file" ]; then //-f is used to check for the file name following it

echo "ERROR: Credentials file does not exist" >&2; exit 1;

elif [ ! -d "$backup\_directory" ]; then //-d is used to refer the directory

echo "ERROR: Backup directory does not exist" >&2; exit 1;

fi

source $credentials\_file //the keyword source executes the file passed as an argument to it

if [ -z ${hostname:+word} ]; then //checks whether the hostname is there or not

echo "ERROR: hostname is not set" >&2; exit 1;

elif [ -z ${username:+word} ]; then //checks whether the username is there or not

echo "ERROR: username is not set" >&2; exit 1;

elif [ -z ${password:+word} ]; then //checks whether the password is given or not

echo "ERROR: password is not set" >&2; exit 1;

fi

mysqldump -h$hostname -u$username -p$password --all-databases > backup.sql

//the above statement passes the entered arguments to the mysql dump file

//the mysqldump stores the backup of the database sessions at the specified location

if [[ $? != 0 ]]; then //$? Checks for the returned value of last executed statement

echo "ERROR: Error in taking mysql backup" >&2; exit 1;

fi

mv backup.sql $backup\_directory/$(date +%F\_%R).sql

//moves the backup to a new directory, %F and %R are used the display date

path\_to\_script=$(realpath "$0")

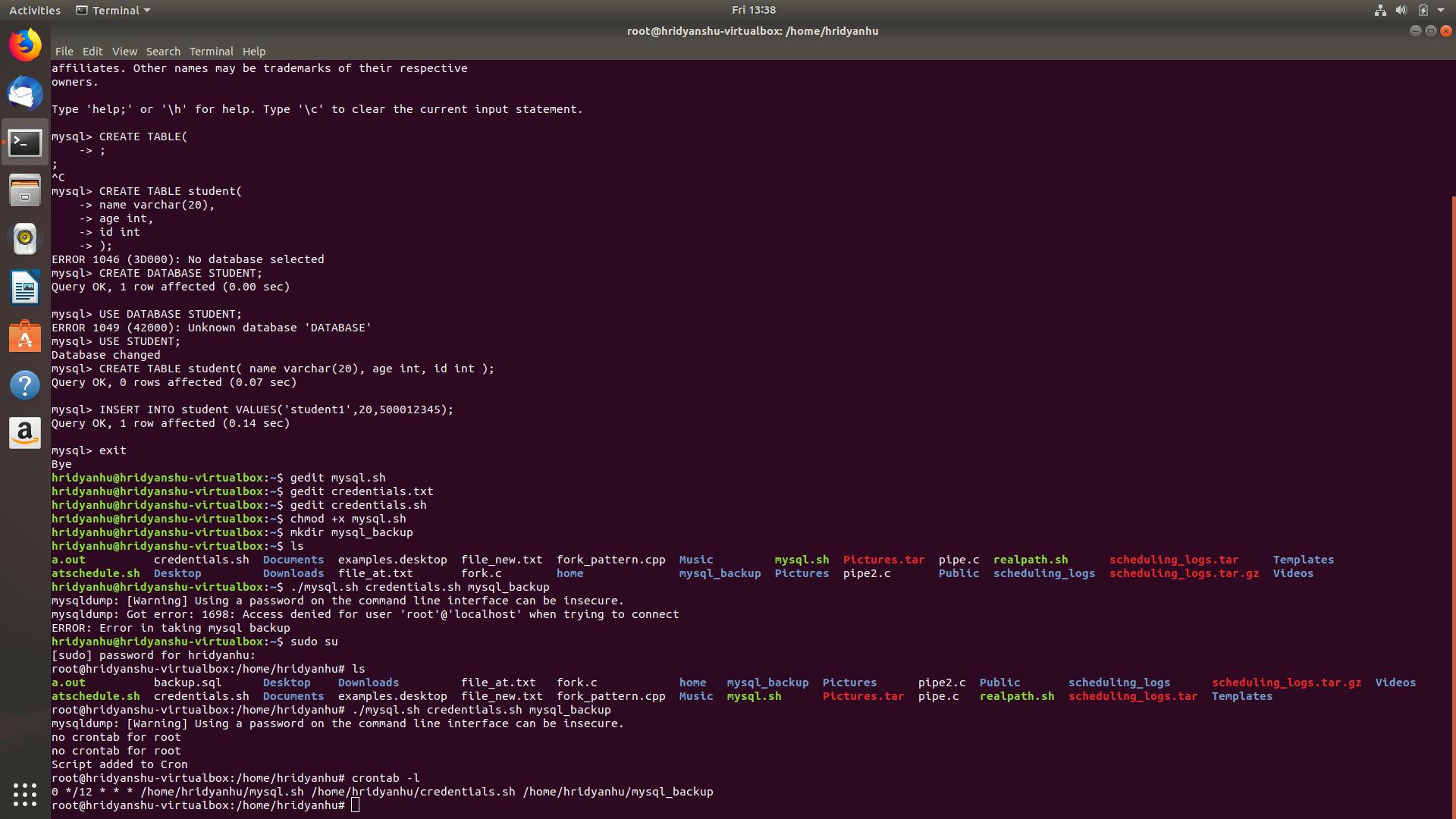
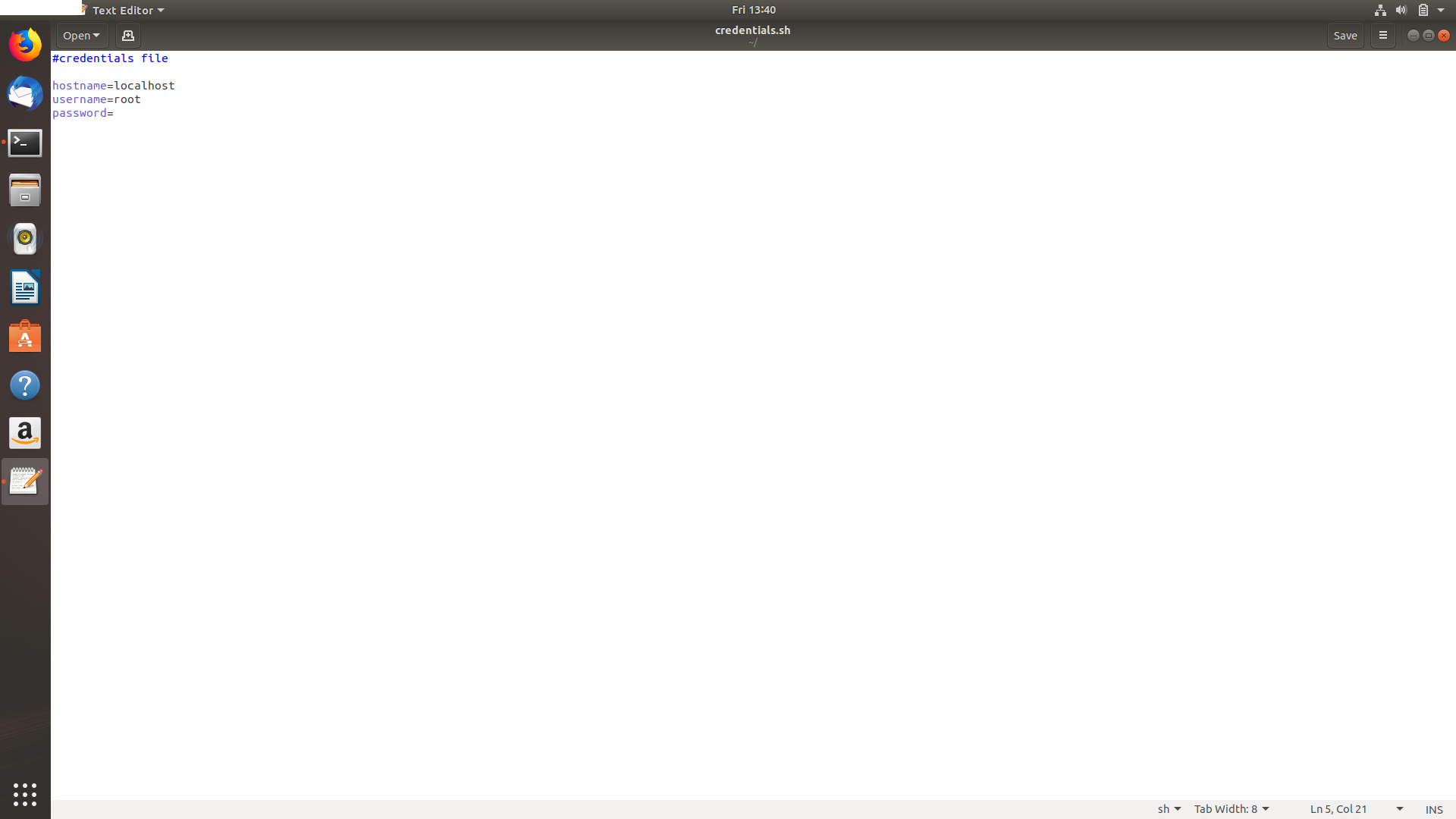
if ! (crontab -l | grep -Fxq "0 \*/12 \* \* \* $path\_to\_script $credentials\_file $backup\_directory"); then

crontab -l | { cat; echo "0 \*/12 \* \* \* $path\_to\_script $credentials\_file $backup\_directory"; } | crontab -

echo "Script added to Cron"

fi

exit 0



**SHELL SCRIPTS**

1. Write a Shell script to convert Hexadecimal to Binary.

**Code:**

#!/bin/bash

echo "Type a hex number"

read hexNum

echo -n "The binary value of $hexNum="

echo "obase=2; ibase=16; $hexNum" | bc

2. Write a Shell script to convert Octal to Binary.

**Code:**

#!/bin/bash

echo "Type an octal number"

read octalNum

echo -n "The binary value of $octalNum="

echo "obase=2; ibase=8; $octalNum" | bc

3. Write a Shell script to check for Palindrome.

**Code:**

#!/bin/bashecho -n "Enter number : "read n# store single digitsd=0# store number in reverse orderrev=""# store original numberon=$nwhile [ $n -gt 0 ]dosd=$(( $n % 10 )) # get Remainder n=$(( $n / 10 )) # get next digit# store previous number and current digit in reverse rev=$( echo ${rev}${sd} ) doneif [ $on -eq $rev ];thenecho "Number is palindrome"elseecho "Number is NOT palindrome"fi

4. Write a Shell script to check for Perfect Number.

**Code:**

# !/bin/bash

echo "Enter the number"

read num

s=0

i=1

while [ $i -lt $(expr "$num" - "1") ]

do

temp=$(( num % i ))

if (( $temp == 0 ))

then

s=$(( s + i ))

fi

i=$(( i + 1 ))

done

if [ $s == $num ]

then

echo "Perfect number"

else

echo "Not a perfect number"

fi

5. Write a Shell script to check for ArmStrong number.

echo ”enter the number”

read n

q=$n

a=0

while [ $q – gt 0 ]

do

r= `expr $q % 10 `

q= `expr $q / 10 `

a=`expr $a + $r /\* $r /\*$r `

done

if [ $a=$n ]

then

echo “the number $n is armstrong number”

else

echo “the number $n is not armstrong number”

fi

8. Write a Shell script to convert Hexadecimal to Octal.

#!/bin/bash

echo "Type a hex number"

read hexNum

echo -n "The octal value of $hexNum="

echo "obase=8; ibase=16; $hexNum" | bc

9. Write a Shell script to convert Octal to Hexadecimal.

#!/bin/bash

echo "Type an octal number"

read octalNum

echo -n "The binary value of $octalNum="

echo "obase=2; ibase=8; $octalNum" | bc

10. Write a Shell script to convert Decimal to Hexadecimal.

#!/bin/bash

echo "Type a decimal number"

read decimalNum

echo -n "The hexadecimal value of $decimalNum="

echo "obase=16; ibase=10; $hexNum" | bc