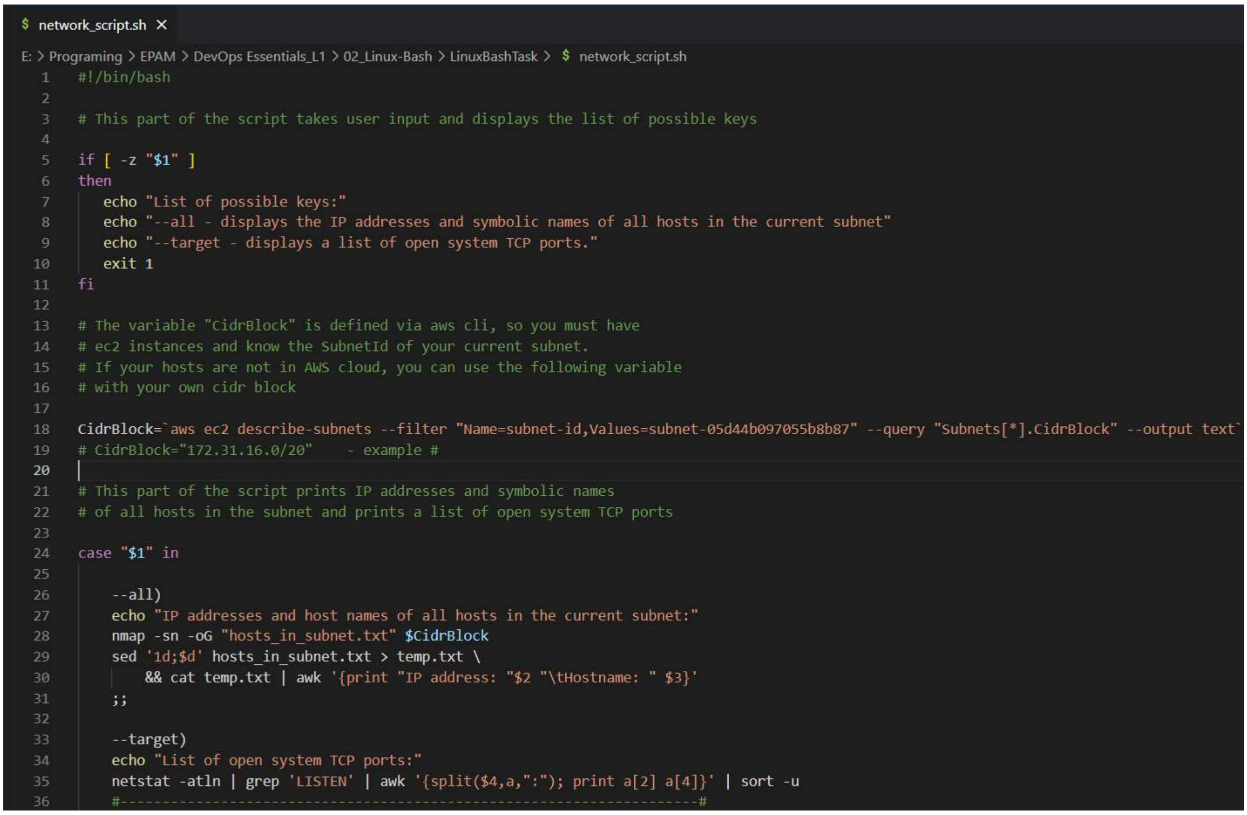
Bash

1. Task A.

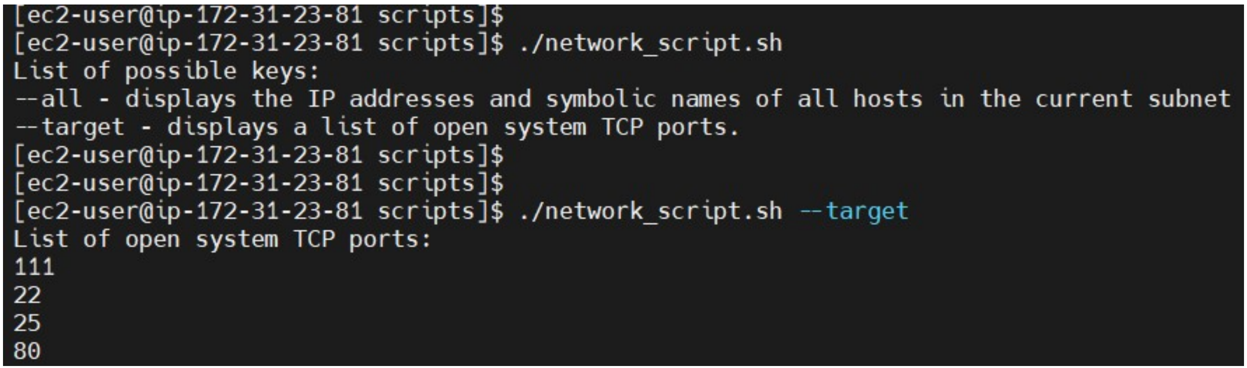
Create a script that uses the following keys:

1. When starting **without parameters**, it will display a list of possible keys and their description.
2. The **--all** key displays the IP addresses and symbolic names of all hosts in the current subnet.
3. The **--target** key displays a list of open system TCP ports.

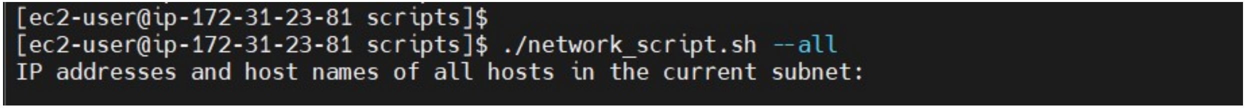


$ network_script.sh ×
E: > Programing > EPAM > DevOps Essentials_L1 > 02_Linux-Bash > LinuxBashTask > $ network_script.sh

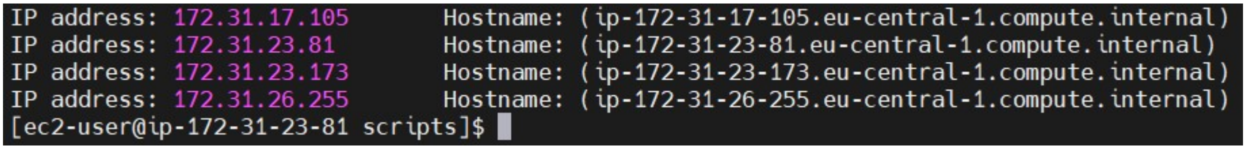
```bash
#!/bin/bash

# This part of the script takes user input and displays the list of possible keys

if [ -z "$1" ]
then
    echo "List of possible keys:"
    echo "--all - displays the IP addresses and symbolic names of all hosts in the current subnet"
    echo "--target - displays a list of open system TCP ports."
    exit 1
fi

# The variable "CidrBlock" is defined via aws cli, so you must have
# ec2 instances and know the SubnetId of your current subnet.
# If your hosts are not in AWS cloud, you can use the following variable
# with your own cidr block

CidrBlock=`aws ec2 describe-subnets --filter "Name=subnet-id,Values=subnet-05d44b097055b8b87" --query "Subnets[*].CidrBlock" --output text`
# CidrBlock="172.31.16.0/20"    - example #

# This part of the script prints IP addresses and symbolic names
# of all hosts in the subnet and prints a list of open system TCP ports

case "$1" in

    --all)
    echo "IP addresses and host names of all hosts in the current subnet:"
    nmap -sn -oG "hosts_in_subnet.txt" $CidrBlock
    sed '1d;$d' hosts_in_subnet.txt > temp.txt \
        && cat temp.txt | awk '{print "IP address: "$2 "\tHostname: " $3}'
    ;;

    --target)
    echo "List of open system TCP ports:"
    netstat -atln | grep 'LISTEN' | awk '{split($4,a,":"); print a[2] a[4]}' | sort -u
    #--------------------------------------------------------------------#
```



```
[ec2-user@ip-172-31-23-81 scripts]$
[ec2-user@ip-172-31-23-81 scripts]$ ./network_script.sh
List of possible keys:
--all - displays the IP addresses and symbolic names of all hosts in the current subnet
--target - displays a list of open system TCP ports.
[ec2-user@ip-172-31-23-81 scripts]$
[ec2-user@ip-172-31-23-81 scripts]$
[ec2-user@ip-172-31-23-81 scripts]$ ./network_script.sh --target
List of open system TCP ports:
111
22
25
80
```



```
[ec2-user@ip-172-31-23-81 scripts]$
[ec2-user@ip-172-31-23-81 scripts]$ ./network_script.sh --all
IP addresses and host names of all hosts in the current subnet:
```



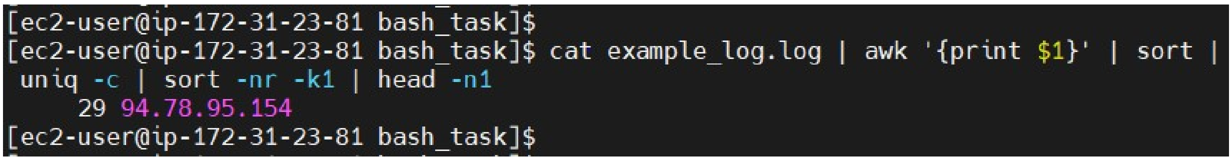
```
IP address: 172.31.17.105       Hostname: (ip-172-31-17-105.eu-central-1.compute.internal)
IP address: 172.31.23.81        Hostname: (ip-172-31-23-81.eu-central-1.compute.internal)
IP address: 172.31.23.173       Hostname: (ip-172-31-23-173.eu-central-1.compute.internal)
IP address: 172.31.26.255       Hostname: (ip-172-31-26-255.eu-central-1.compute.internal)
[ec2-user@ip-172-31-23-81 scripts]$ █
```

2. Task B.

Using Apache log example create a script to answer the following questions:

1. From which ip were the most requests?

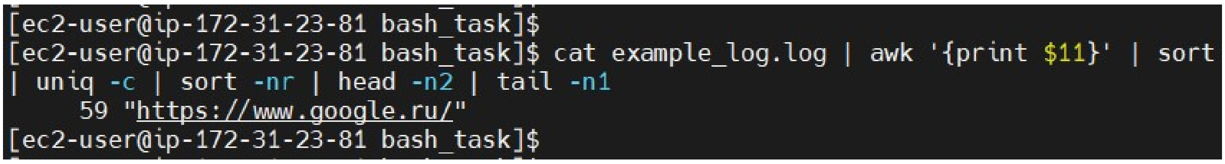
```
$  cat example_log.log | awk '{print $1}' | sort | uniq -c |
sort -nr -k1 | head -n1
```



```
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$ cat example_log.log | awk '{print $1}' | sort |
 uniq -c | sort -nr -k1 | head -n1
     29 94.78.95.154
[ec2-user@ip-172-31-23-81 bash_task]$
```

2. What is the most requested page?

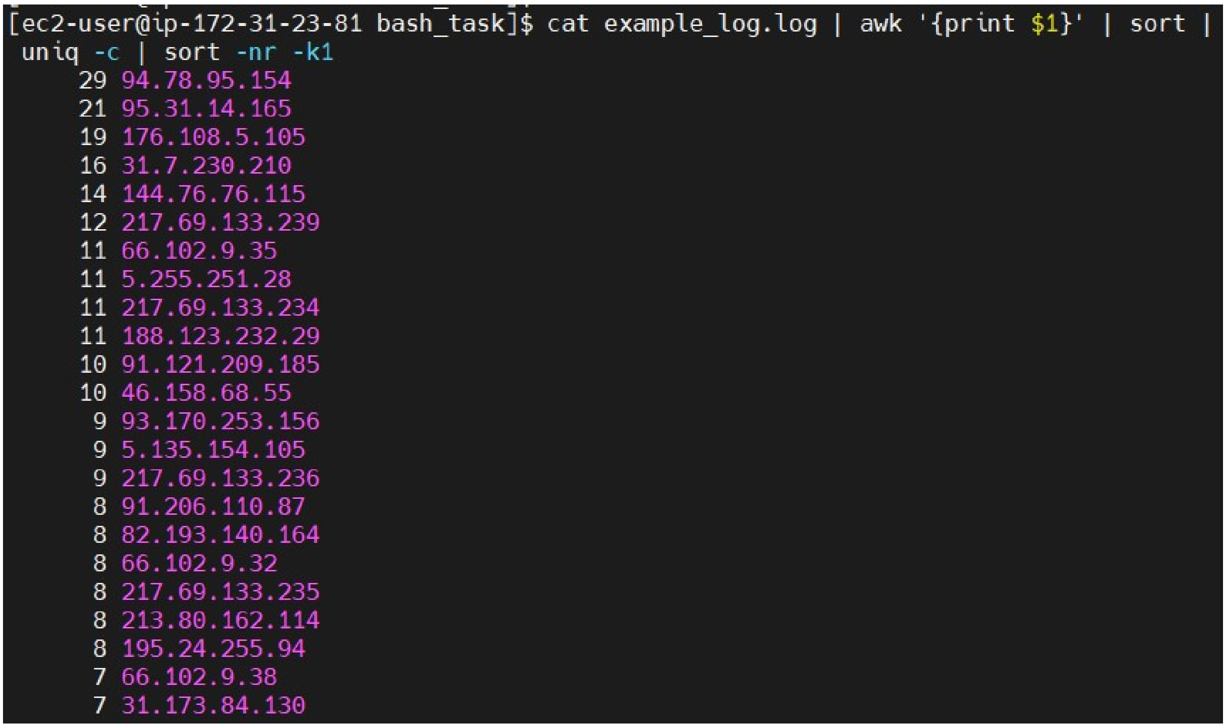
```
$  cat example_log.log | awk '{print $11}' | sort | uniq -c |
sort -nr | head -n2 | tail -n1
```



```
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$ cat example_log.log | awk '{print $11}' | sort
| uniq -c | sort -nr | head -n2 | tail -n1
     59 "https://www.google.ru/"
[ec2-user@ip-172-31-23-81 bash_task]$
```

3. How many requests were there from each ip?

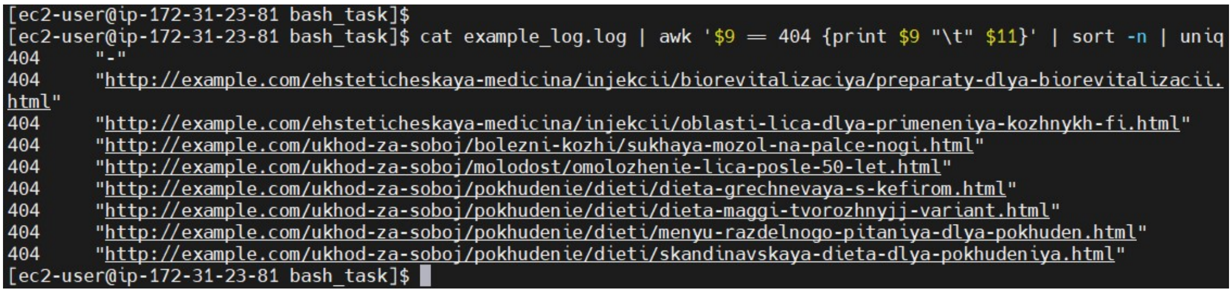
```
$  cat example_log.log | awk '{print $1}' | sort | uniq -c |
sort -nr -k1
```



```
[ec2-user@ip-172-31-23-81 bash_task]$ cat example_log.log | awk '{print $1}' | sort |
 uniq -c | sort -nr -k1
     29 94.78.95.154
     21 95.31.14.165
     19 176.108.5.105
     16 31.7.230.210
     14 144.76.76.115
     12 217.69.133.239
     11 66.102.9.35
     11 5.255.251.28
     11 217.69.133.234
     11 188.123.232.29
     10 91.121.209.185
     10 46.158.68.55
      9 93.170.253.156
      9 5.135.154.105
      9 217.69.133.236
      8 91.206.110.87
      8 82.193.140.164
      8 66.102.9.32
      8 217.69.133.235
      8 213.80.162.114
      8 195.24.255.94
      7 66.102.9.38
      7 31.173.84.130
```

4. What non-existent pages were clients referred to?

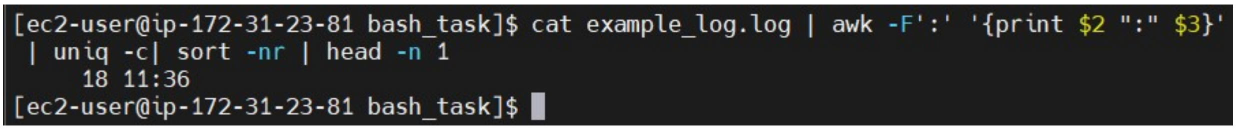
```
$  cat example_log.log | awk '$9 == 404 {print $9 "\t" $11}' |
sort -n | uniq
```



```
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$ cat example_log.log | awk '$9 == 404 {print $9 "\t" $11}' | sort -n | uniq
404     "-"
404     "http://example.com/ehsteticheskaya-medicina/injekcii/biorevitalizaciya/preparaty-dlya-biorevitalizacii.
html"
404     "http://example.com/ehsteticheskaya-medicina/injekcii/oblasti-lica-dlya-primeneniya-kozhnykh-fi.html"
404     "http://example.com/ukhod-za-soboj/bolezni-kozhi/sukhaya-mozol-na-palce-nogi.html"
404     "http://example.com/ukhod-za-soboj/molodost/omolozhenie-lica-posle-50-let.html"
404     "http://example.com/ukhod-za-soboj/pokhudenie/dieti/dieta-grechnevaya-s-kefirom.html"
404     "http://example.com/ukhod-za-soboj/pokhudenie/dieti/dieta-maggi-tvorozhnyjj-variant.html"
404     "http://example.com/ukhod-za-soboj/pokhudenie/dieti/menyu-razdelnogo-pitaniya-dlya-pokhuden.html"
404     "http://example.com/ukhod-za-soboj/pokhudenie/dieti/skandinavskaya-dieta-dlya-pokhudeniya.html"
[ec2-user@ip-172-31-23-81 bash_task]$
```

5. What time did site get the most requests?

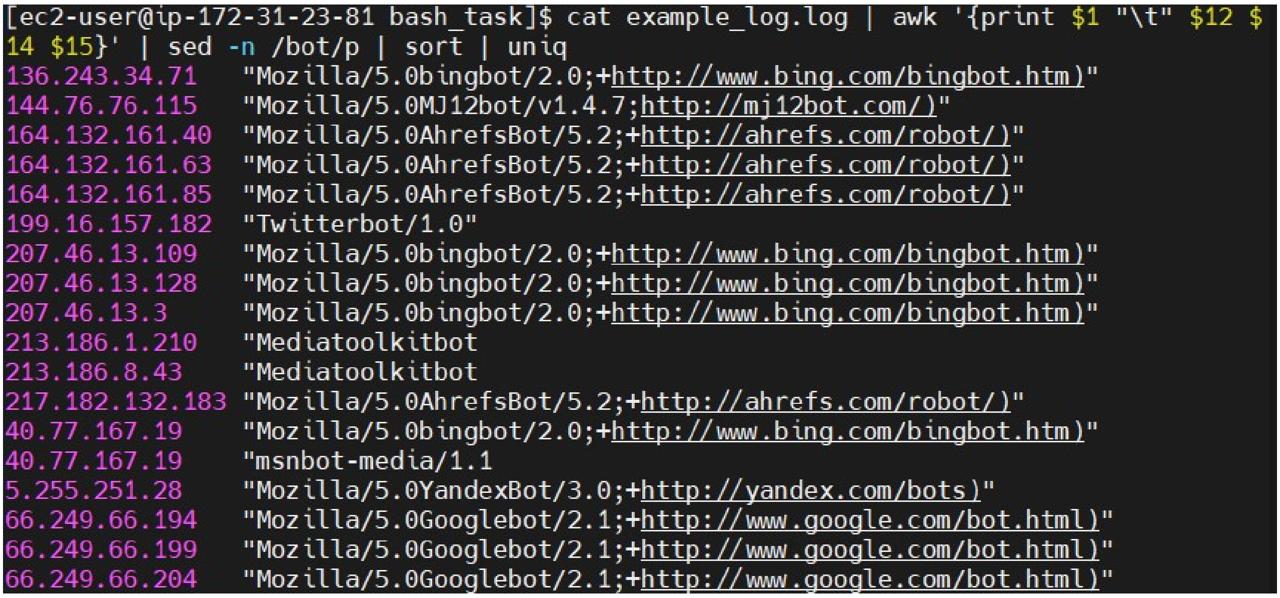
```
$  cat example_log.log | awk -F':' '{print $2 ":" $3}' | uniq -c
| sort -nr | head -n1
```



```
[ec2-user@ip-172-31-23-81 bash_task]$ cat example_log.log | awk -F':' '{print $2 ":" $3}'
 | uniq -c| sort -nr | head -n 1
     18 11:36
[ec2-user@ip-172-31-23-81 bash_task]$
```

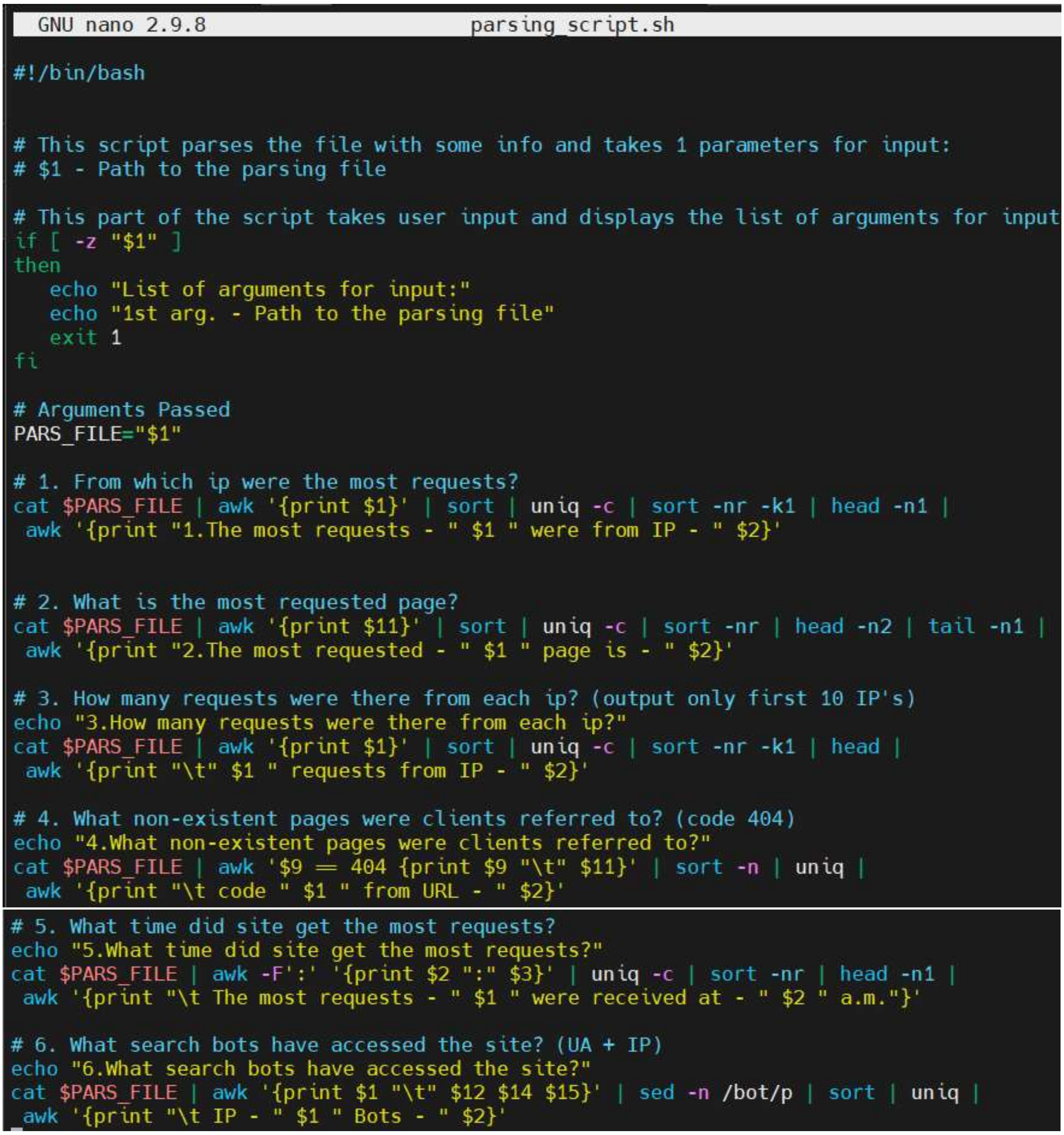
6. What search bots have accessed the site? (UA + IP)

```
$  cat example_log.log | awk '{print $1 "\t" $12 $14 $15}' | sed
-n /bot/p | sort | uniq
```



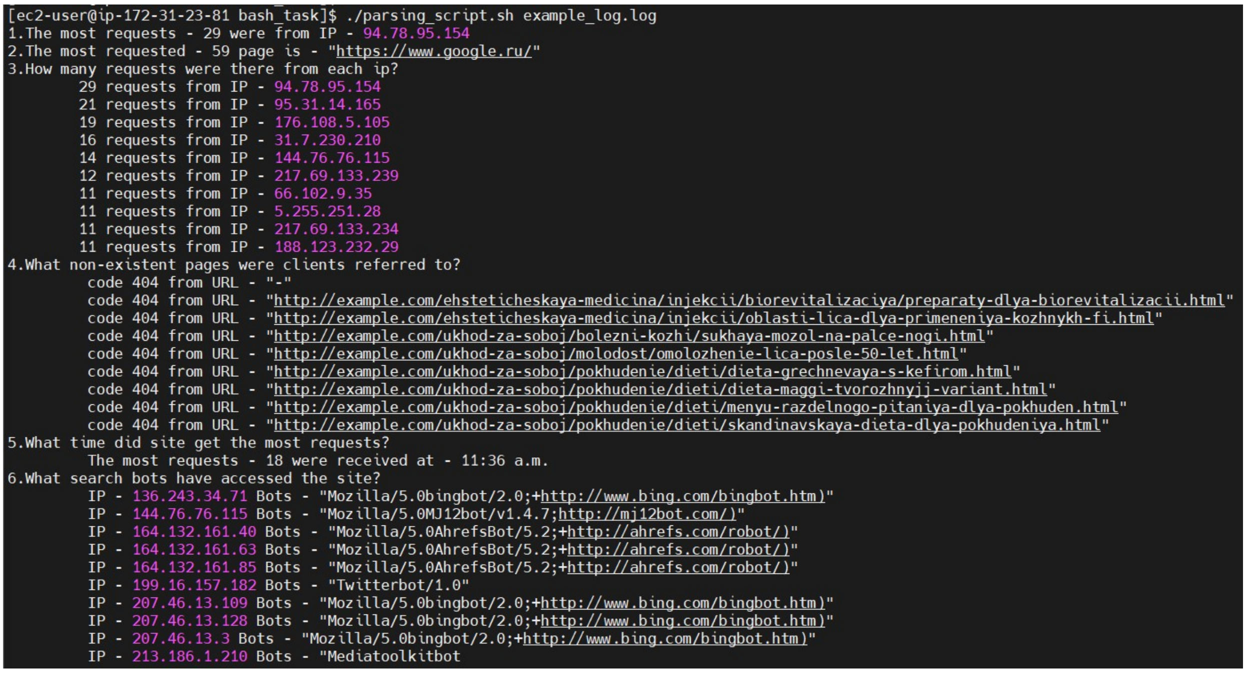
```
[ec2-user@ip-172-31-23-81 bash_task]$ cat example_log.log | awk '{print $1 "\t" $12 $
14 $15}' | sed -n /bot/p | sort | uniq
136.243.34.71   "Mozilla/5.0bingbot/2.0;+http://www.bing.com/bingbot.htm)"
144.76.76.115   "Mozilla/5.0MJ12bot/v1.4.7;http://mj12bot.com/)"
164.132.161.40  "Mozilla/5.0AhrefsBot/5.2;+http://ahrefs.com/robot/)"
164.132.161.63  "Mozilla/5.0AhrefsBot/5.2;+http://ahrefs.com/robot/)"
164.132.161.85  "Mozilla/5.0AhrefsBot/5.2;+http://ahrefs.com/robot/)"
199.16.157.182  "Twitterbot/1.0"
207.46.13.109   "Mozilla/5.0bingbot/2.0;+http://www.bing.com/bingbot.htm)"
207.46.13.128   "Mozilla/5.0bingbot/2.0;+http://www.bing.com/bingbot.htm)"
207.46.13.3     "Mozilla/5.0bingbot/2.0;+http://www.bing.com/bingbot.htm)"
213.186.1.210   "Mediatoolkitbot
213.186.8.43    "Mediatoolkitbot
217.182.132.183 "Mozilla/5.0AhrefsBot/5.2;+http://ahrefs.com/robot/)"
40.77.167.19    "Mozilla/5.0bingbot/2.0;+http://www.bing.com/bingbot.htm)"
40.77.167.19    "msnbot-media/1.1
5.255.251.28    "Mozilla/5.0YandexBot/3.0;+http://yandex.com/bots)"
66.249.66.194   "Mozilla/5.0Googlebot/2.1;+http://www.google.com/bot.html)"
66.249.66.199   "Mozilla/5.0Googlebot/2.1;+http://www.google.com/bot.html)"
66.249.66.204   "Mozilla/5.0Googlebot/2.1;+http://www.google.com/bot.html)"
```

- **Bash script for task B:**



```
GNU nano 2.9.8                        parsing_script.sh

#!/bin/bash


# This script parses the file with some info and takes 1 parameters for input:
# $1 - Path to the parsing file

# This part of the script takes user input and displays the list of arguments for input
if [ -z "$1" ]
then
   echo "List of arguments for input:"
   echo "1st arg. - Path to the parsing file"
   exit 1
fi

# Arguments Passed
PARS_FILE="$1"

# 1. From which ip were the most requests?
cat $PARS_FILE | awk '{print $1}' | sort | uniq -c | sort -nr -k1 | head -n1 |
 awk '{print "1.The most requests - " $1 " were from IP - " $2}'


# 2. What is the most requested page?
cat $PARS_FILE | awk '{print $11}' | sort | uniq -c | sort -nr | head -n2 | tail -n1 |
 awk '{print "2.The most requested - " $1 " page is - " $2}'

# 3. How many requests were there from each ip? (output only first 10 IP's)
echo "3.How many requests were there from each ip?"
cat $PARS_FILE | awk '{print $1}' | sort | uniq -c | sort -nr -k1 | head |
 awk '{print "\t" $1 " requests from IP - " $2}'

# 4. What non-existent pages were clients referred to? (code 404)
echo "4.What non-existent pages were clients referred to?"
cat $PARS_FILE | awk '$9 == 404 {print $9 "\t" $11}' | sort -n | uniq |
 awk '{print "\t code " $1 " from URL - " $2}'
# 5. What time did site get the most requests?
echo "5.What time did site get the most requests?"
cat $PARS_FILE | awk -F':' '{print $2 ":" $3}' | uniq -c | sort -nr | head -n1 |
 awk '{print "\t The most requests - " $1 " were received at - " $2 " a.m."}'

# 6. What search bots have accessed the site? (UA + IP)
echo "6.What search bots have accessed the site?"
cat $PARS_FILE | awk '{print $1 "\t" $12 $14 $15}' | sed -n /bot/p | sort | uniq |
 awk '{print "\t IP - " $1 " Bots - " $2}'
```

- **Run script for task B:**



```
[ec2-user@ip-172-31-23-81 bash_task]$ ./parsing_script.sh example_log.log
1.The most requests - 29 were from IP - 94.78.95.154
2.The most requested - 59 page is - "https://www.google.ru/"
3.How many requests were there from each ip?
        29 requests from IP - 94.78.95.154
        21 requests from IP - 95.31.14.165
        19 requests from IP - 176.108.5.105
        16 requests from IP - 31.7.230.210
        14 requests from IP - 144.76.76.115
        12 requests from IP - 217.69.133.239
        11 requests from IP - 66.102.9.35
        11 requests from IP - 5.255.251.28
        11 requests from IP - 217.69.133.234
        11 requests from IP - 188.123.232.29
4.What non-existent pages were clients referred to?
         code 404 from URL - "-"
         code 404 from URL - "http://example.com/ehsteticheskaya-medicina/injekcii/biorevitalizaciya/preparaty-dlya-biorevitalizacii.html"
         code 404 from URL - "http://example.com/ehsteticheskaya-medicina/injekcii/oblasti-lica-dlya-primeneniya-kozhnykh-fi.html"
         code 404 from URL - "http://example.com/ukhod-za-soboj/bolezni-kozhi/sukhaya-mozol-na-palce-nogi.html"
         code 404 from URL - "http://example.com/ukhod-za-soboj/molodost/omolozhenie-lica-posle-50-let.html"
         code 404 from URL - "http://example.com/ukhod-za-soboj/pokhudenie/dieti/dieta-grechnevaya-s-kefirom.html"
         code 404 from URL - "http://example.com/ukhod-za-soboj/pokhudenie/dieti/dieta-maggi-tvorozhnyjj-variant.html"
         code 404 from URL - "http://example.com/ukhod-za-soboj/pokhudenie/dieti/menyu-razdelnogo-pitaniya-dlya-pokhuden.html"
         code 404 from URL - "http://example.com/ukhod-za-soboj/pokhudenie/dieti/skandinavskaya-dieta-dlya-pokhudeniya.html"
5.What time did site get the most requests?
         The most requests - 18 were received at - 11:36 a.m.
6.What search bots have accessed the site?
         IP - 136.243.34.71 Bots - "Mozilla/5.0bingbot/2.0;+http://www.bing.com/bingbot.htm)"
         IP - 144.76.76.115 Bots - "Mozilla/5.0MJ12bot/v1.4.7;http://mj12bot.com/)"
         IP - 164.132.161.40 Bots - "Mozilla/5.0AhrefsBot/5.2;+http://ahrefs.com/robot/)"
         IP - 164.132.161.63 Bots - "Mozilla/5.0AhrefsBot/5.2;+http://ahrefs.com/robot/)"
         IP - 164.132.161.85 Bots - "Mozilla/5.0AhrefsBot/5.2;+http://ahrefs.com/robot/)"
         IP - 199.16.157.182 Bots - "Twitterbot/1.0"
         IP - 207.46.13.109 Bots - "Mozilla/5.0bingbot/2.0;+http://www.bing.com/bingbot.htm)"
         IP - 207.46.13.128 Bots - "Mozilla/5.0bingbot/2.0;+http://www.bing.com/bingbot.htm)"
         IP - 207.46.13.3 Bots - "Mozilla/5.0bingbot/2.0;+http://www.bing.com/bingbot.htm)"
         IP - 213.186.1.210 Bots - "Mediatoolkitbot
```

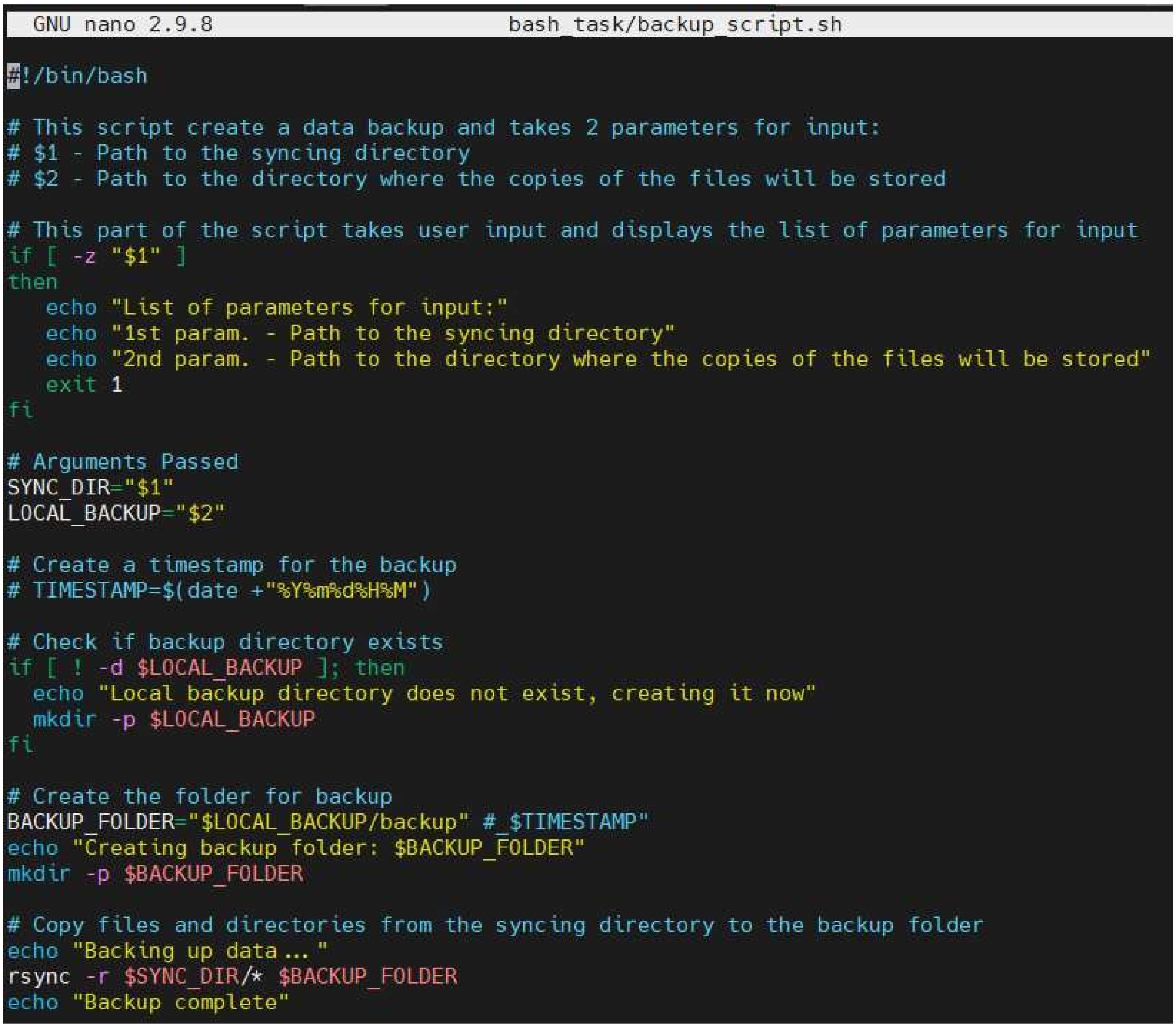
3. Task C.

Create a data backup script that takes the following data as parameters:

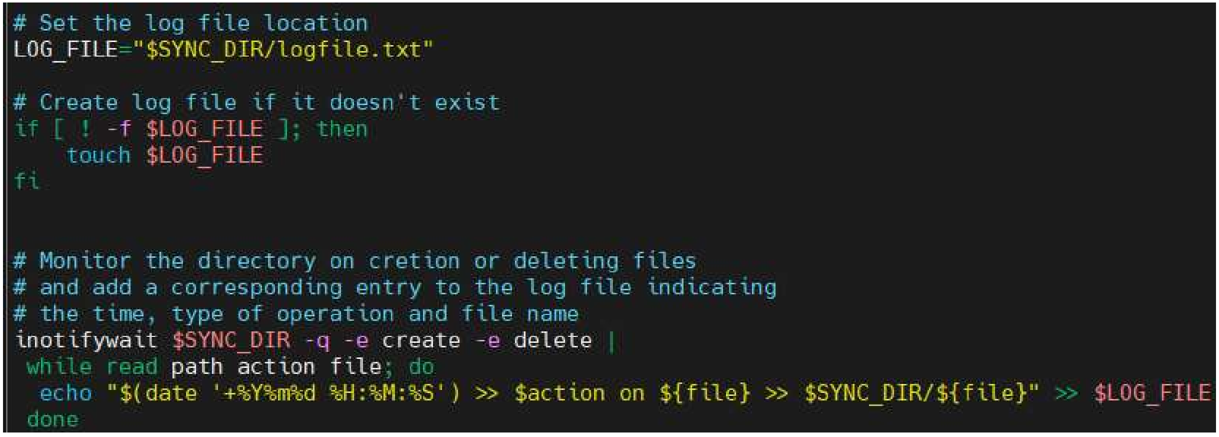
1. Path to the syncing directory.
2. The path to the directory where the copies of the files will be stored.

In case of adding new or deleting old files, the script must add a corresponding entry to the log file cat. [The command to run the script must be added to crontab with a run frequency of one minute].

- **Script**

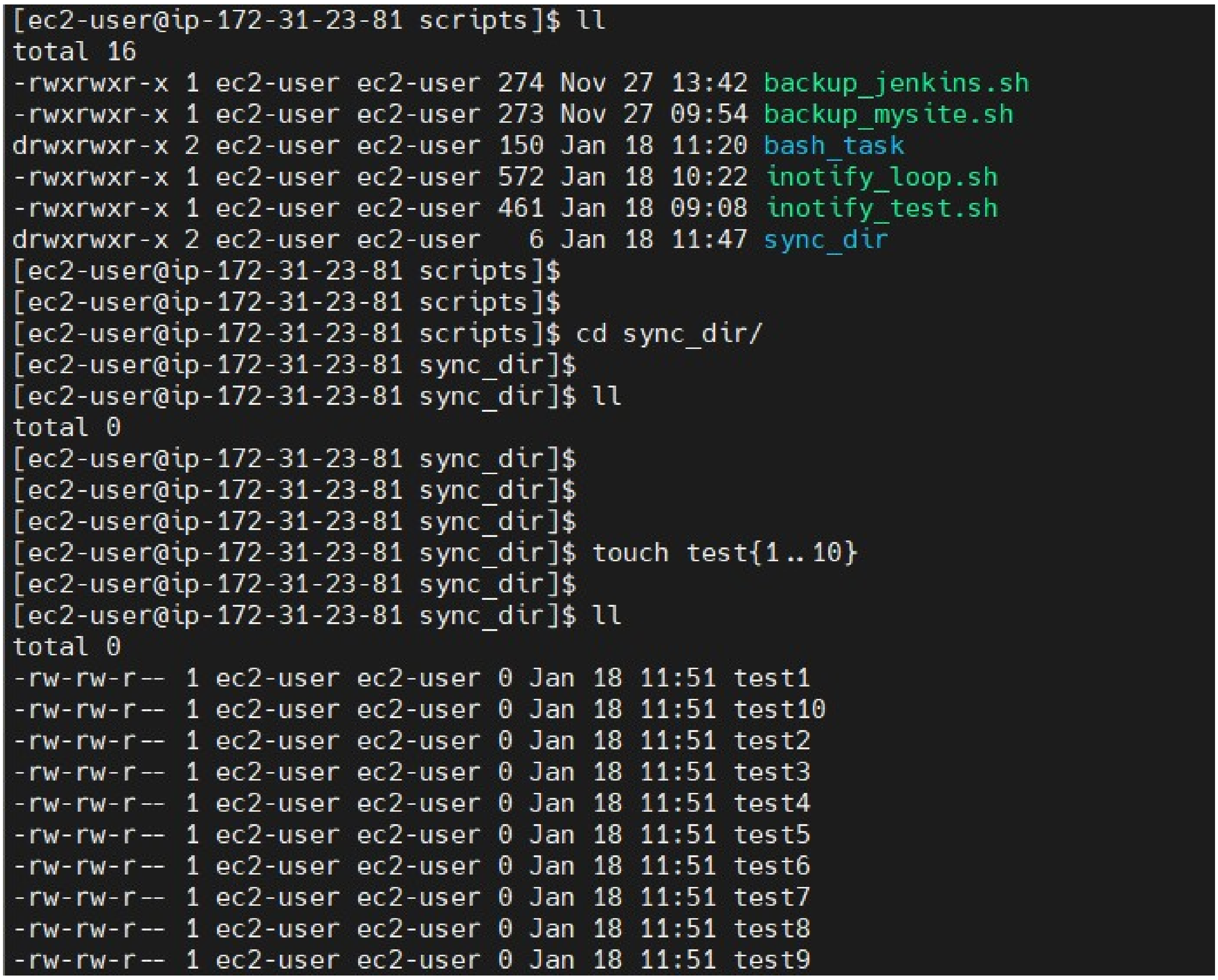


```
  GNU nano 2.9.8                      bash_task/backup_script.sh

#!/bin/bash

# This script create a data backup and takes 2 parameters for input:
# $1 - Path to the syncing directory
# $2 - Path to the directory where the copies of the files will be stored

# This part of the script takes user input and displays the list of parameters for input
if [ -z "$1" ]
then
   echo "List of parameters for input:"
   echo "1st param. - Path to the syncing directory"
   echo "2nd param. - Path to the directory where the copies of the files will be stored"
   exit 1
fi

# Arguments Passed
SYNC_DIR="$1"
LOCAL_BACKUP="$2"

# Create a timestamp for the backup
# TIMESTAMP=$(date +"%Y%m%d%H%M")

# Check if backup directory exists
if [ ! -d $LOCAL_BACKUP ]; then
  echo "Local backup directory does not exist, creating it now"
  mkdir -p $LOCAL_BACKUP
fi

# Create the folder for backup
BACKUP_FOLDER="$LOCAL_BACKUP/backup" #_$TIMESTAMP"
echo "Creating backup folder: $BACKUP_FOLDER"
mkdir -p $BACKUP_FOLDER

# Copy files and directories from the syncing directory to the backup folder
echo "Backing up data ... "
rsync -r $SYNC_DIR/* $BACKUP_FOLDER
echo "Backup complete"
```

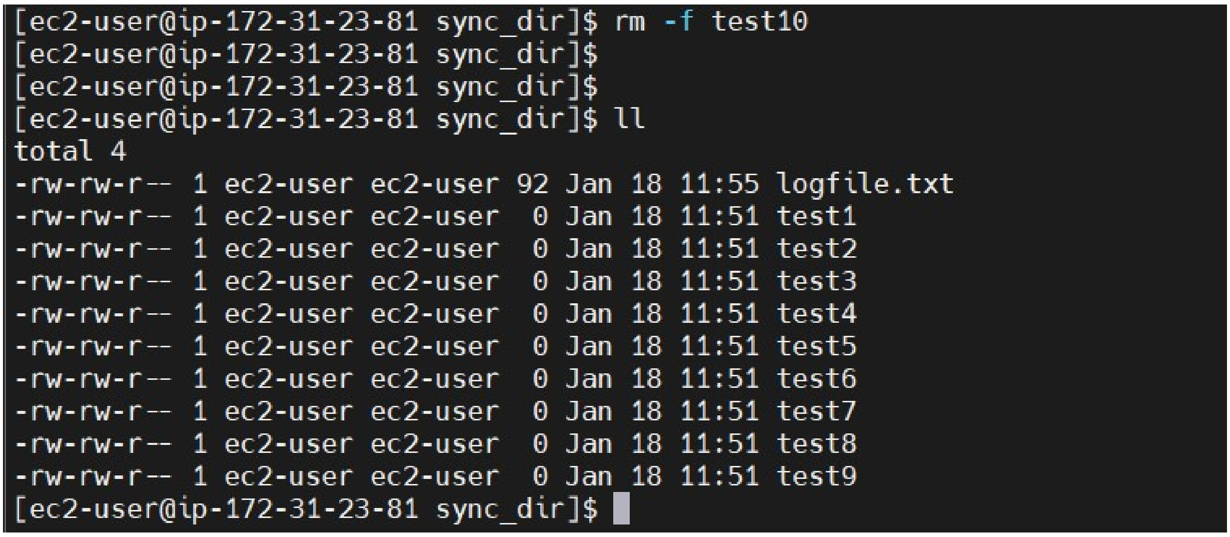


```
# Set the log file location
LOG_FILE="$SYNC_DIR/logfile.txt"

# Create log file if it doesn't exist
if [ ! -f $LOG_FILE ]; then
    touch $LOG_FILE
fi


# Monitor the directory on cretion or deleting files
# and add a corresponding entry to the log file indicating
# the time, type of operation and file name
inotifywait $SYNC_DIR -q -e create -e delete |
 while read path action file; do
  echo "$(date '+%Y%m%d %H:%M:%S') >> $action on ${file} >> $SYNC_DIR/${file}" >> $LOG_FILE
 done
```

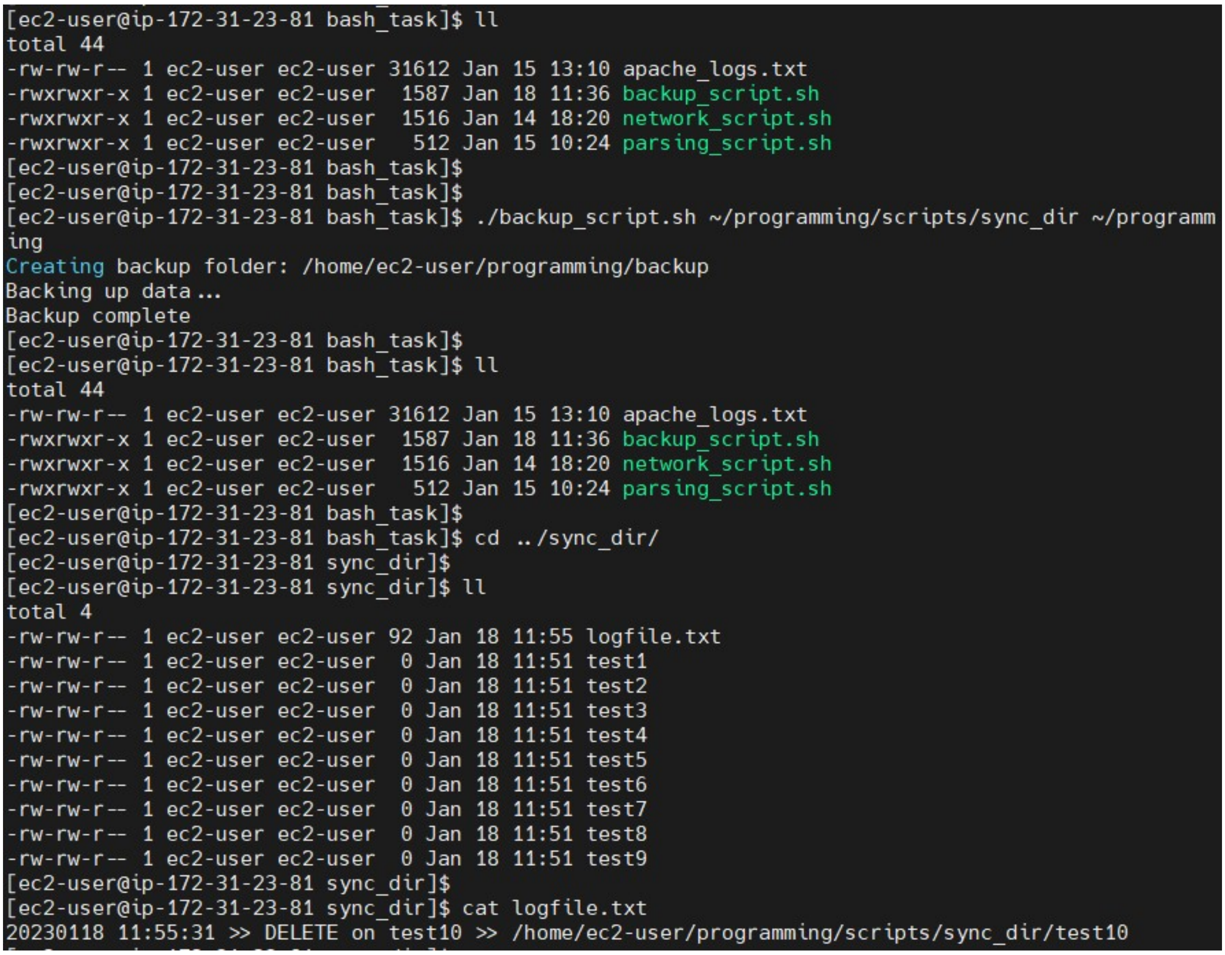
- **Example of the script working**



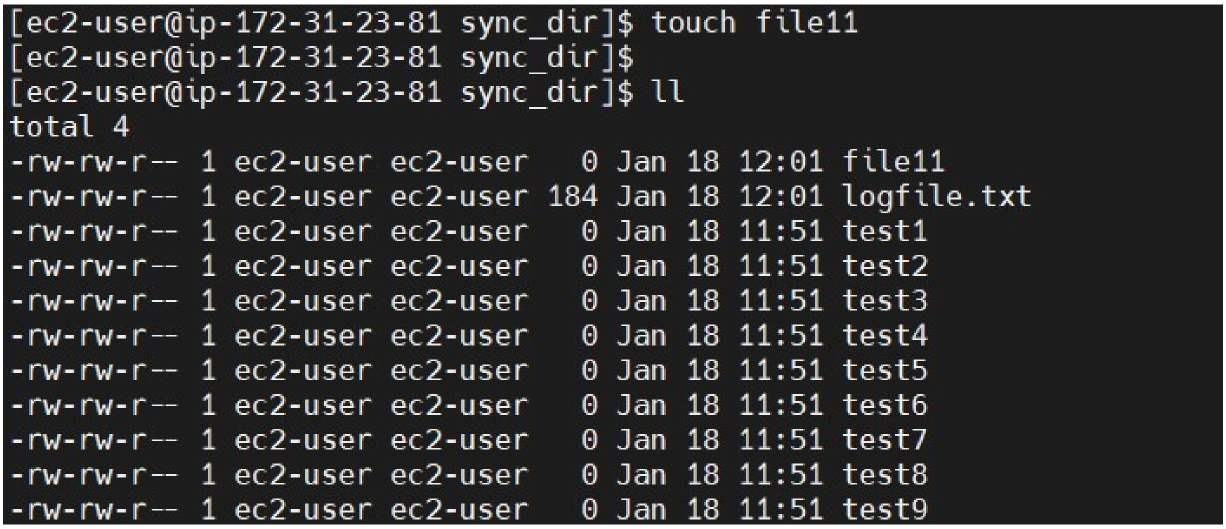
```
[ec2-user@ip-172-31-23-81 scripts]$ ll
total 16
-rwxrwxr-x 1 ec2-user ec2-user 274 Nov 27 13:42 backup_jenkins.sh
-rwxrwxr-x 1 ec2-user ec2-user 273 Nov 27 09:54 backup_mysite.sh
drwxrwxr-x 2 ec2-user ec2-user 150 Jan 18 11:20 bash_task
-rwxrwxr-x 1 ec2-user ec2-user 572 Jan 18 10:22 inotify_loop.sh
-rwxrwxr-x 1 ec2-user ec2-user 461 Jan 18 09:08 inotify_test.sh
drwxrwxr-x 2 ec2-user ec2-user   6 Jan 18 11:47 sync_dir
[ec2-user@ip-172-31-23-81 scripts]$
[ec2-user@ip-172-31-23-81 scripts]$
[ec2-user@ip-172-31-23-81 scripts]$ cd sync_dir/
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$ ll
total 0
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$ touch test{1..10}
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$ ll
total 0
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test1
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test10
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test2
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test3
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test4
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test5
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test6
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test7
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test8
-rw-rw-r-- 1 ec2-user ec2-user 0 Jan 18 11:51 test9
```



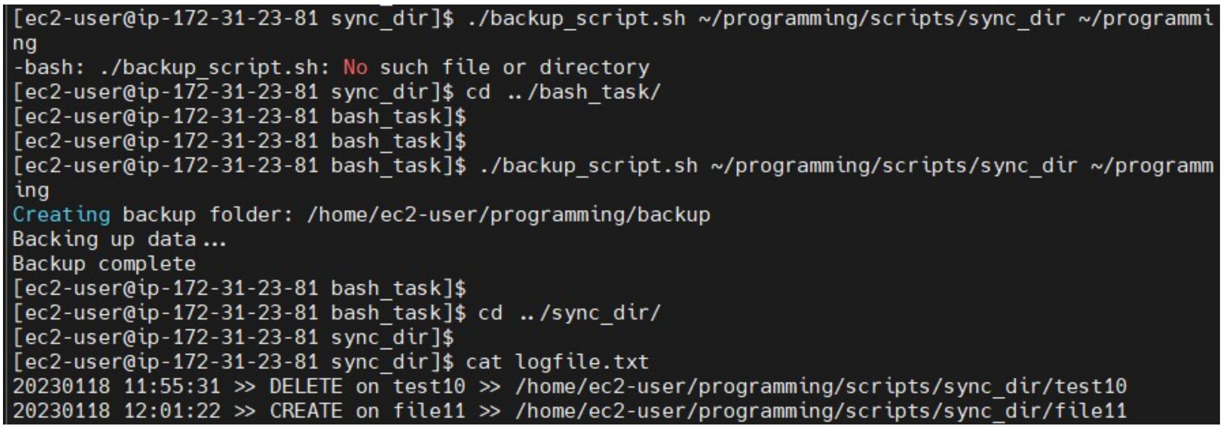
```
[ec2-user@ip-172-31-23-81 sync_dir]$ rm -f test10
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$ ll
total 4
-rw-rw-r-- 1 ec2-user ec2-user 92 Jan 18 11:55 logfile.txt
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test1
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test2
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test3
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test4
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test5
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test6
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test7
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test8
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test9
[ec2-user@ip-172-31-23-81 sync_dir]$
```



```
[ec2-user@ip-172-31-23-81 bash_task]$ ll
total 44
-rw-rw-r-- 1 ec2-user ec2-user 31612 Jan 15 13:10 apache_logs.txt
-rwxrwxr-x 1 ec2-user ec2-user  1587 Jan 18 11:36 backup_script.sh
-rwxrwxr-x 1 ec2-user ec2-user  1516 Jan 14 18:20 network_script.sh
-rwxrwxr-x 1 ec2-user ec2-user   512 Jan 15 10:24 parsing_script.sh
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$ ./backup_script.sh ~/programming/scripts/sync_dir ~/programm
ing
Creating backup folder: /home/ec2-user/programming/backup
Backing up data...
Backup complete
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$ ll
total 44
-rw-rw-r-- 1 ec2-user ec2-user 31612 Jan 15 13:10 apache_logs.txt
-rwxrwxr-x 1 ec2-user ec2-user  1587 Jan 18 11:36 backup_script.sh
-rwxrwxr-x 1 ec2-user ec2-user  1516 Jan 14 18:20 network_script.sh
-rwxrwxr-x 1 ec2-user ec2-user   512 Jan 15 10:24 parsing_script.sh
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$ cd ../sync_dir/
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$ ll
total 4
-rw-rw-r-- 1 ec2-user ec2-user 92 Jan 18 11:55 logfile.txt
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test1
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test2
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test3
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test4
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test5
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test6
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test7
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test8
-rw-rw-r-- 1 ec2-user ec2-user  0 Jan 18 11:51 test9
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$ cat logfile.txt
20230118 11:55:31 >> DELETE on test10 >> /home/ec2-user/programming/scripts/sync_dir/test10
```

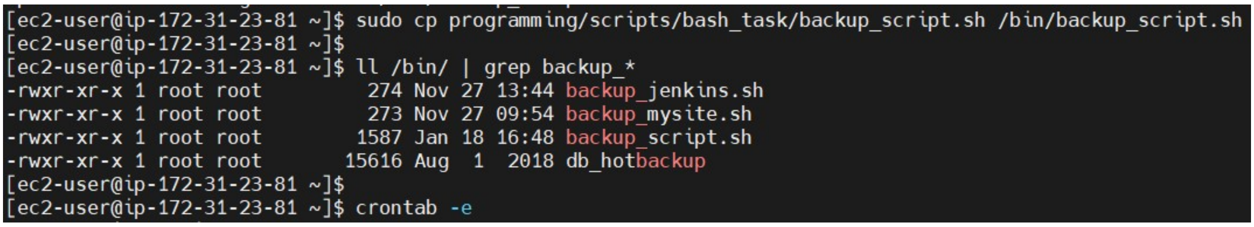


```
[ec2-user@ip-172-31-23-81 sync_dir]$ touch file11
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$ ll
total 4
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 12:01 file11
-rw-rw-r-- 1 ec2-user ec2-user 184 Jan 18 12:01 logfile.txt
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 11:51 test1
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 11:51 test2
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 11:51 test3
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 11:51 test4
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 11:51 test5
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 11:51 test6
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 11:51 test7
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 11:51 test8
-rw-rw-r-- 1 ec2-user ec2-user   0 Jan 18 11:51 test9
```

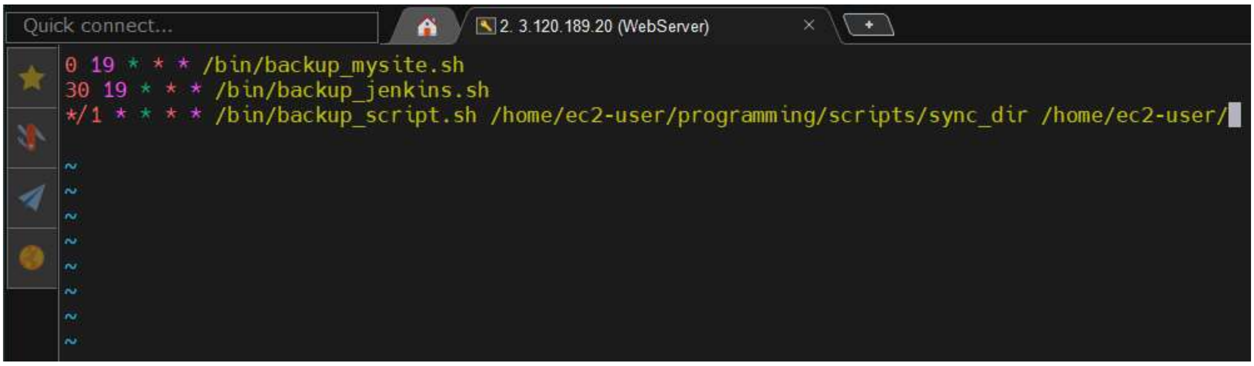


```
[ec2-user@ip-172-31-23-81 sync_dir]$ ./backup_script.sh ~/programming/scripts/sync_dir ~/programmi
ng
-bash: ./backup_script.sh: No such file or directory
[ec2-user@ip-172-31-23-81 sync_dir]$ cd ../bash_task/
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$ ./backup_script.sh ~/programming/scripts/sync_dir ~/programm
ing
Creating backup folder: /home/ec2-user/programming/backup
Backing up data...
Backup complete
[ec2-user@ip-172-31-23-81 bash_task]$
[ec2-user@ip-172-31-23-81 bash_task]$ cd ../sync_dir/
[ec2-user@ip-172-31-23-81 sync_dir]$
[ec2-user@ip-172-31-23-81 sync_dir]$ cat logfile.txt
20230118 11:55:31 >> DELETE on test10 >> /home/ec2-user/programming/scripts/sync_dir/test10
20230118 12:01:22 >> CREATE on file11 >> /home/ec2-user/programming/scripts/sync_dir/file11
```

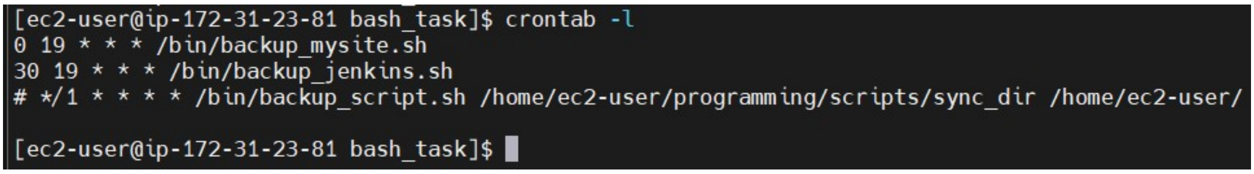
- **Crontab (run the script frequency of one minute)**



```
[ec2-user@ip-172-31-23-81 ~]$ sudo cp programming/scripts/bash_task/backup_script.sh /bin/backup_script.sh
[ec2-user@ip-172-31-23-81 ~]$
[ec2-user@ip-172-31-23-81 ~]$ ll /bin/ | grep backup_*
-rwxr-xr-x 1 root root         274 Nov 27 13:44 backup_jenkins.sh
-rwxr-xr-x 1 root root         273 Nov 27 09:54 backup_mysite.sh
-rwxr-xr-x 1 root root        1587 Jan 18 16:48 backup_script.sh
-rwxr-xr-x 1 root root       15616 Aug  1  2018 db_hotbackup
[ec2-user@ip-172-31-23-81 ~]$
[ec2-user@ip-172-31-23-81 ~]$ crontab -e
```



```
0 19 * * * /bin/backup_mysite.sh
30 19 * * * /bin/backup_jenkins.sh
*/1 * * * * /bin/backup_script.sh /home/ec2-user/programming/scripts/sync_dir /home/ec2-user/

~
~
~
~
~
~
~
~
```



```
[ec2-user@ip-172-31-23-81 bash_task]$ crontab -l
0 19 * * * /bin/backup_mysite.sh
30 19 * * * /bin/backup_jenkins.sh
# */1 * * * * /bin/backup_script.sh /home/ec2-user/programming/scripts/sync_dir /home/ec2-user/

[ec2-user@ip-172-31-23-81 bash_task]$
```