

## Assignment -6: Linux and Bash by Hemanth

1.Set a cron job at host sh01to start your individual instance at the defined work period and stop it out of your work time. Turn in host, cron job line and screenshot to validate the implementation.

Answer:

1.Create a new bash script by running following command

```
nano start-stop-ec2.sh
```

2.Connect to the instance using SSH and create a shell script that starts and stops the instance based on a given schedule. For example, you can create a script named **start-stop-ec2.sh** with the following contents:

```
hemanth@sh01:~$ cat start-stop-ec2.sh
INSTANCE_ID="i-018896f6b27c70563"
REGION="us-east-1"

# Get the current hour (in 24-hour format)
HOUR=$(date +%H)

# Start the instance if the current hour is between 8am and 6pm (inclusive)
if [ "$HOUR" -ge 8 ] && [ "$HOUR" -le 18 ]; then
    ssh -i "~/CSYE7374Spring23.pem" ubuntu@ec2-18-212-162-236.compute-1.amazo
naws.com
    echo "EC2 instance started at $(date)"
# Stop the instance if the current hour is outside of the work period
else
    exit
    echo "EC2 instance stopped at $(date)"
fi
```

3.Make the script executable by running the following command:

```
chmod +x start-stop-ec2.sh
./start-stop-ec2.sh
```

This will start or stop the specified EC2 instance based on the current time, and print a message to the console indicating whether the instance was started or stopped.

6. In the crontab editor, add a new line at the bottom of the file with the following syntax:

```
hemanth@sh01:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 * * * 1-5 /home/hemanth/start-stop-ec2.sh
```

This cron job will run the **start-stop-ec2.sh** script at the top of every hour on weekdays (Monday through Friday), which will start or stop the EC2 instance based on the current time.

2. The 1000 Genomes Project is a collaboration among research groups in the US, UK, China and Germany to produce an extensive catalog of human genetic variation that will support future medical research studies. write a bash script with following requirements  
Input: the set of the file names (SRR062634\_1 SRR062634\_2 SRR062635\_1 SRR062635\_2)  
Function and Output: oDownload the data to subdirectory at host sh01 under /scratch/[user]/hg38  
oPrint the base name of the file and the base name with no extension from the absolute path of file  
i.e. downloaded data at /scratch/[user]/hg38/xyz.fastq.gz, output as xyz.fastq.gz and xyzSRR062634\_1.filt.fastq.gz SRR062634\_2.filt.fastq.gz SRR062635\_1.filt.fastq.gz SRR062635\_2.filt.fastq.gz  
https://ftp.ncbi.nih.gov/1000genomes/ftp/phase3/data/HG00096/sequence\_read/  
Turn in the location of your script, the command to run your script and screenshot of the result.

Answer:

genome.sh

```
#!/bin/bash
```

```
# Set the base URL for the data
```

```
URL="ftp://ftp.ncbi.nih.gov/1000genomes/ftp/phase3/data/HG00096/sequence_read/"
```

```
# Set the destination directory for the downloaded data
```

```
DEST_DIR="/scratch/hemanth/hg38/"
```

```
# Make sure the destination directory exists
```

```
mkdir -p "$DEST_DIR"
```

```
# Loop over the input file names
```

```
for FILENAME in SRR062634_1 SRR062634_2 SRR062635_1 SRR062635_2; do
```

```
    # Construct the URL for the file
```

```
    FILE_URL="${URL}${FILENAME}.filt.fastq.gz"
```

```
    # Download the file to the destination directory
```

```
    wget -P "$DEST_DIR" "$FILE_URL"
```

```
    # Extract the base name and the base name with no extension from the absolute path of the file
```

```
    BASENAME=$(basename "$DEST_DIR/${FILENAME}.filt.fastq.gz")
```

```
    BASENAME_NO_EXT="${BASENAME%. *}"
```

```
    # Print the base name and the base name with no extension
```

```
    echo "$BASENAME"
```

```
    echo "$BASENAME_NO_EXT"
done
```

```
hemanth@sh01:~$ nano genome.sh
hemanth@sh01:~$ cat genome.sh
#!/bin/bash

# Set the base URL for the data
URL="ftp://ftp.ncbi.nih.gov/1000genomes/ftp/phase3/data/HG00096/sequence_read/"

# Set the destination directory for the downloaded data
DEST_DIR="/scratch/hemanth/hg38/"

# Make sure the destination directory exists
mkdir -p "$DEST_DIR"

# Loop over the input file names
for FILENAME in SRR062634_1 SRR062634_2 SRR062635_1 SRR062635_2; do
    # Construct the URL for the file
    FILE_URL="${URL}${FILENAME}.filt.fastq.gz"

    # Download the file to the destination directory
    wget -P "$DEST_DIR" "$FILE_URL"

    # Extract the base name and the base name with no extension from the absolute path of the file
    BASENAME=$(basename "$DEST_DIR/${FILENAME}.filt.fastq.gz")
    BASENAME_NO_EXT="${BASENAME%.*}"

    # Print the base name and the base name with no extension
    echo "$BASENAME"
    echo "$BASENAME_NO_EXT"
done
```

This will download the files to the subdirectory **/scratch/hemanth/hg38** and print the base name and base name with no extension for each file. we can verify that the files have been downloaded by checking the target directory.\

```

[hemanth@sh01:~]$ chmod u+x genome.sh
[hemanth@sh01:~]$ ./genome.sh
--2023-02-23 22:06:57-- ftp://ftp.ncbi.nih.gov/1000genomes/ftp/phase3/data/HG00096/sequence_read/SRR062634_1.filt.fastq.gz
=> '/scratch/hemanth/hg38/SRR062634_1.filt.fastq.gz.1'
Resolving ftp.ncbi.nih.gov (ftp.ncbi.nih.gov)... 130.14.250.7, 165.112.9.229, 2607:f220:41e:250::12, ...
Connecting to ftp.ncbi.nih.gov (ftp.ncbi.nih.gov)[130.14.250.7]:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done. ==> PWD ... done.
==> TYPE I ... done. ==> CWD (1) /1000genomes/ftp/phase3/data/HG00096/sequence_read ... done.
==> SIZE SRR062634_1.filt.fastq.gz ... 1938959399
==> PASV ... done. ==> RETR SRR062634_1.filt.fastq.gz ... done.
Length: 1938959399 (1.8G) (unauthoritative)

SRR062634_1.filt.fastq.gz.1 100%[=====] 1.81G 41.5MB/s in 23s

2023-02-23 22:07:21 (80.6 MB/s) - '/scratch/hemanth/hg38/SRR062634_1.filt.fastq.gz.1' saved [1938959399]

SRR062634_1.filt.fastq.gz
SRR062634_1.filt.fastq
--2023-02-23 22:07:21-- ftp://ftp.ncbi.nih.gov/1000genomes/ftp/phase3/data/HG00096/sequence_read/SRR062634_2.filt.fastq.gz
=> '/scratch/hemanth/hg38/SRR062634_2.filt.fastq.gz.1'
Resolving ftp.ncbi.nih.gov (ftp.ncbi.nih.gov)... 130.14.250.7, 165.112.9.229, 2607:f220:41e:250::12, ...
Connecting to ftp.ncbi.nih.gov (ftp.ncbi.nih.gov)[130.14.250.7]:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done. ==> PWD ... done.
==> TYPE I ... done. ==> CWD (1) /1000genomes/ftp/phase3/data/HG00096/sequence_read ... done.
==> SIZE SRR062634_2.filt.fastq.gz ... 1921928972
==> PASV ... done. ==> RETR SRR062634_2.filt.fastq.gz ... done.
Length: 1921928972 (1.8G) (unauthoritative)

SRR062634_2.filt.fastq.gz.1 100%[=====] 1.79G 256MB/s in 7.0s

2023-02-23 22:07:45 (262 MB/s) - '/scratch/hemanth/hg38/SRR062634_2.filt.fastq.gz.1' saved [1921928972]

SRR062634_2.filt.fastq.gz
SRR062634_2.filt.fastq
--2023-02-23 22:07:45-- ftp://ftp.ncbi.nih.gov/1000genomes/ftp/phase3/data/HG00096/sequence_read/SRR062635_1.filt.fastq.gz
=> '/scratch/hemanth/hg38/SRR062635_1.filt.fastq.gz.1'
Resolving ftp.ncbi.nih.gov (ftp.ncbi.nih.gov)... 130.14.250.7, 165.112.9.229, 2607:f220:41e:250::12, ...
Connecting to ftp.ncbi.nih.gov (ftp.ncbi.nih.gov)[130.14.250.7]:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done. ==> PWD ... done.
==> TYPE I ... done. ==> CWD (1) /1000genomes/ftp/phase3/data/HG00096/sequence_read ... done.
==> SIZE SRR062635_1.filt.fastq.gz ... 1961900395
==> PASV ... done. ==> RETR SRR062635_1.filt.fastq.gz ... done.
Length: 1961900395 (1.8G) (unauthoritative)

SRR062635_1.filt.fastq.gz.1 100%[=====] 1.83G 292MB/s in 6.8s

2023-02-23 22:08:09 (274 MB/s) - '/scratch/hemanth/hg38/SRR062635_1.filt.fastq.gz.1' saved [1961900395]

SRR062635_1.filt.fastq.gz
SRR062635_1.filt.fastq
--2023-02-23 22:08:09-- ftp://ftp.ncbi.nih.gov/1000genomes/ftp/phase3/data/HG00096/sequence_read/SRR062635_2.filt.fastq.gz
=> '/scratch/hemanth/hg38/SRR062635_2.filt.fastq.gz.1'
Resolving ftp.ncbi.nih.gov (ftp.ncbi.nih.gov)... 130.14.250.7, 165.112.9.229, 2607:f220:41e:250::12, ...
Connecting to ftp.ncbi.nih.gov (ftp.ncbi.nih.gov)[130.14.250.7]:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done. ==> PWD ... done.
==> TYPE I ... done. ==> CWD (1) /1000genomes/ftp/phase3/data/HG00096/sequence_read ... done.
==> SIZE SRR062635_2.filt.fastq.gz ... 1946604084
==> PASV ... done. ==> RETR SRR062635_2.filt.fastq.gz ... done.
Length: 1946604084 (1.8G) (unauthoritative)

SRR062635_2.filt.fastq.gz.1 100%[=====] 1.81G 239MB/s in 6.9s

2023-02-23 22:08:32 (267 MB/s) - '/scratch/hemanth/hg38/SRR062635_2.filt.fastq.gz.1' saved [1946604084]

SRR062635_2.filt.fastq.gz
SRR062635_2.filt.fastq

```