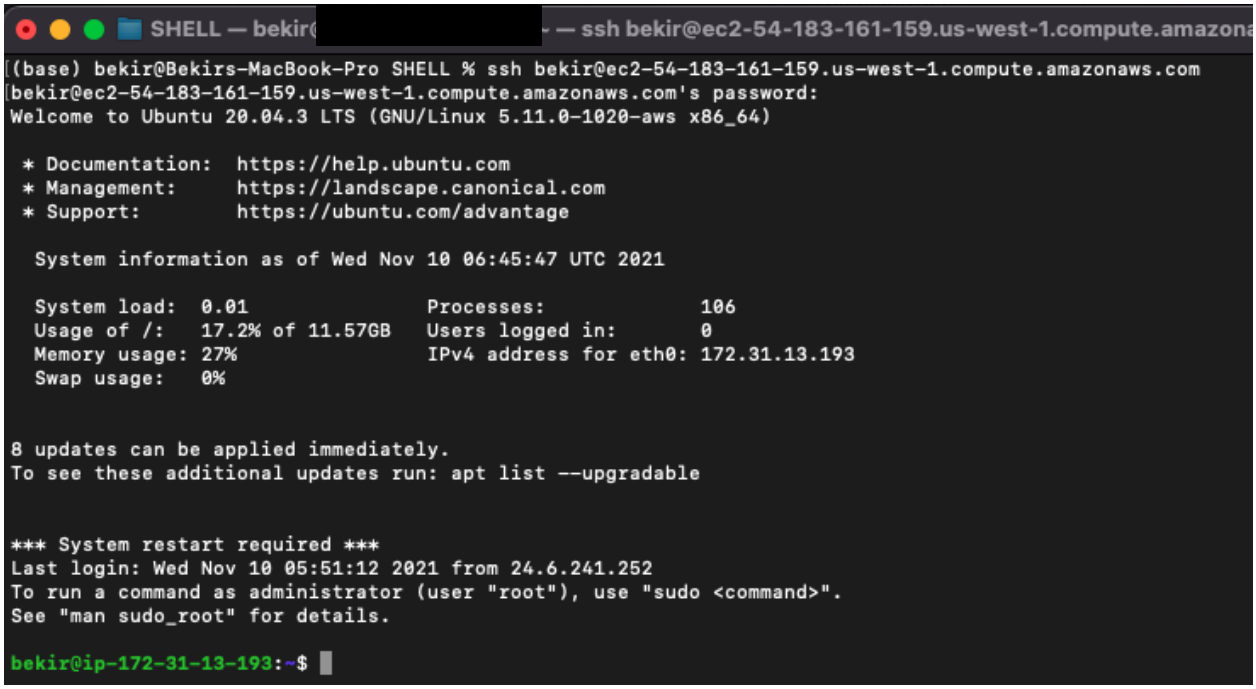
Running a bash script on a remote machine and interacting with it

This document shows how to run a script on a remote server.

Step-1 : Connecting to remote server with ssh

```
ssh bekir@ec2-54-183-161-159.us-west-1.compute.amazonaws.com
```



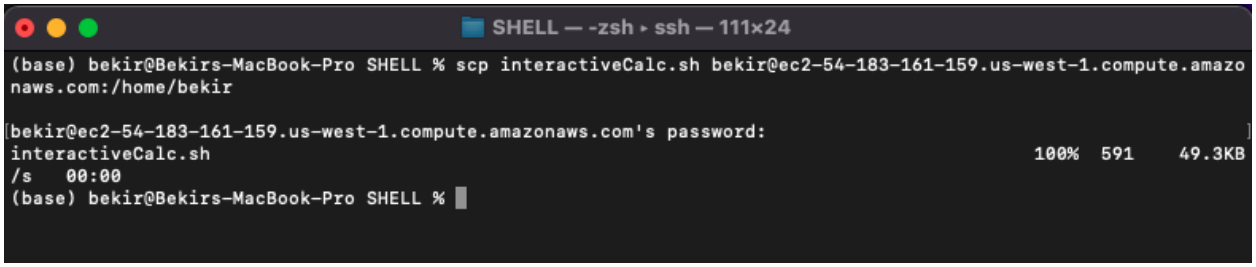
Step-2 : Testing the connection by copy a file to remote desktop

I copied a bash file by using scp to test the connection.

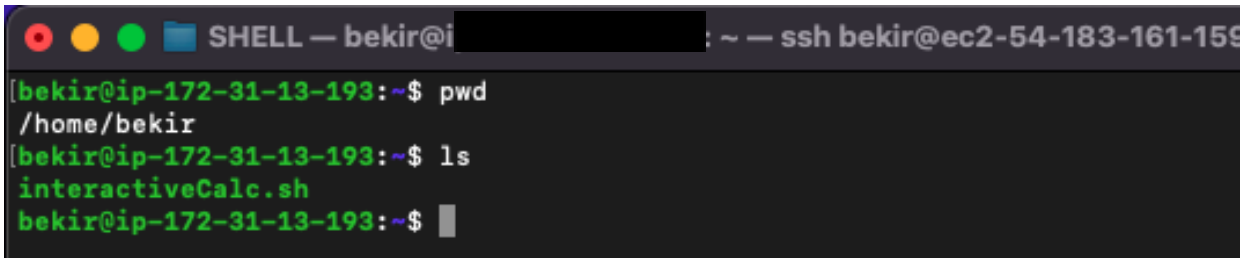
Command:

```
scp interactiveCalc.sh
bekir@ec2-5-18-161-159.us-west-1.compute.amazonaws.com:/home/
bekir
```

Local Machine:



Remote Machine:



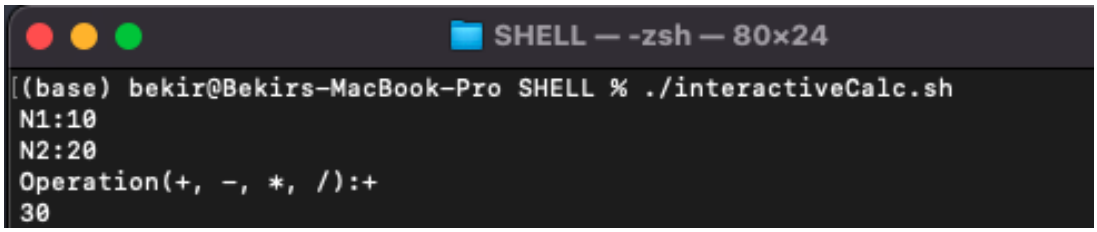
Step-3 : Calculator Script

InteractiveCalc.sh:

```
##!/bin/bash
echo -n "N1:"
read N1

echo -n "N2:"
read N2

echo -n "Operation(+, -, *, /):"
read Operation

if [ "$Operation" = "+" ]
then
   (( result = N1 + N2 ))
elif [ "$Operation" = "-" ]
then
   if [ "$N1" -ge "$N2" ]
   then
```

```
      (( result = N1 - N2 ))
   else
      (( result = N2 - N1 ))
      result="-${result}"
   fi
elif [ "$Operation" = "*" ]
then
   (( result = N1 * N2 ))
elif [ "$Operation" = "/" ]
then
   (( result1 = N1 / N2 ))
   (( result2 = N1 * 100 / N2 ))
   result="${result1}.${result2}"
else
   echo "Error: wrong operation $Operation"
   exit 1
fi

echo $result
```

Testing the InteractiveCalc.sh on the local machine:



Step-4 : Making Calculator Script Interactive by using Expect Script

In this step, I created two files on the local machine. One file called ServerTriger.sh for generating random numbers and operations. Then called runScriptOnServer.exp that sends a command to the remote machine and provides credentials to ssh to the remote machine. Also interact with the calculator.

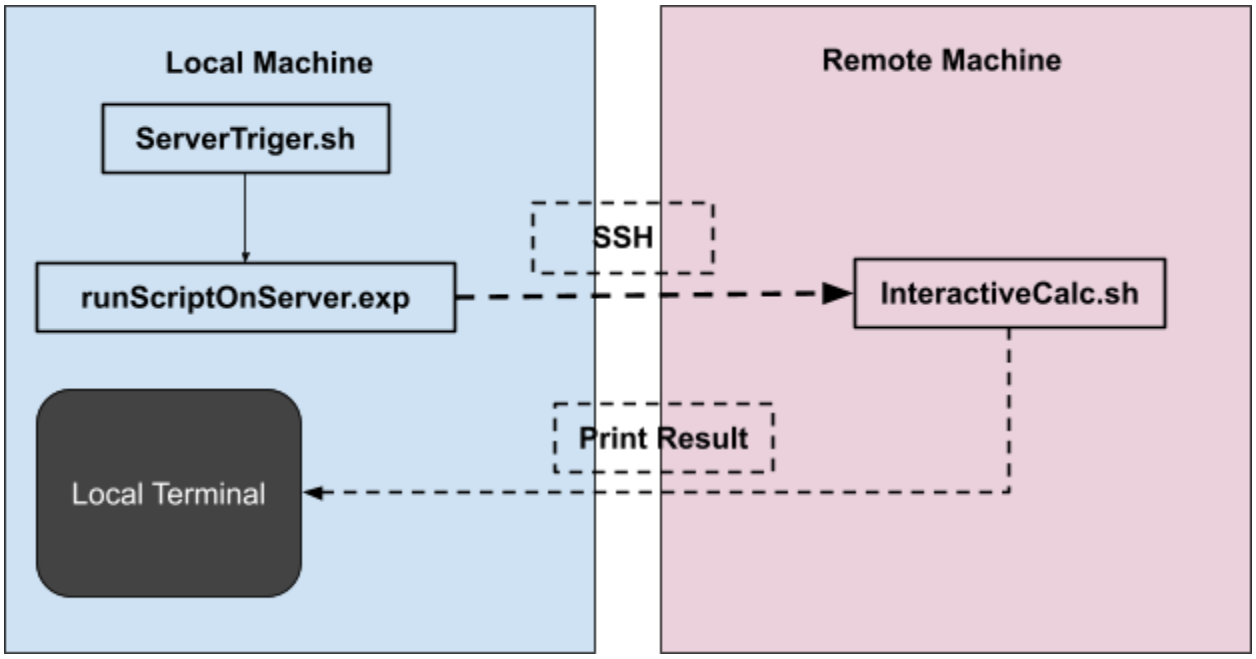
ServerTriger.sh

```
#!/bin/bash
N1=$(shuf -i 1-1000 -n 1)
N2=$(shuf -i 1-1000 -n 1)
ans=$(shuf -i 1-4 -n 1)

if [ "$ans" -eq 1 ]
then
  operation="+"
elif [ "$ans" -eq 2 ]
then
  operation="\-"
elif [ "$ans" -eq 3 ]
then
  operation="\*"
else
  operation="/"
fi
expect runScriptOnServer.exp $N1 $N2 "$operation"
```

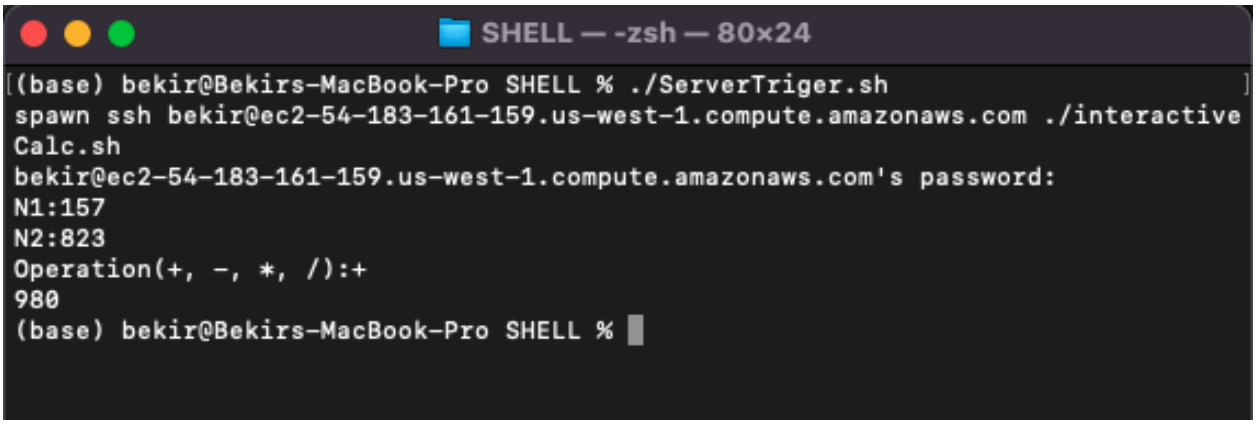
runScriptOnServer.exp

```
set timeout -1
set N1 [lindex $argv 0]
set N2 [lindex $argv 1]
set operation [lindex $argv 2]
spawn ssh bekir@ec2-5-13-161-159.us-west-1.compute.amazonaws.com
./interactiveCalc.sh
expect "bekir@ec2-5-13-161-159.us-west-1.compute.amazonaws.com's
password:" { send "Password\r" }

expect "N1:" { send "$N1\r" }
expect "N2:" { send "$N2\r" }
expect "Operation(+, -, *, /):" { send "$operation\r" }
interact
```

System looks like the figure below:



Here is the result if we run ./ServerTriger.sh manually on local machine:



Step-5 : Setting up a cron job to automate the process

Cron job is needed to be created to automate the process at 1 am everyday and send an email notice.

Creating a cron job command:

```
crontab -e
```

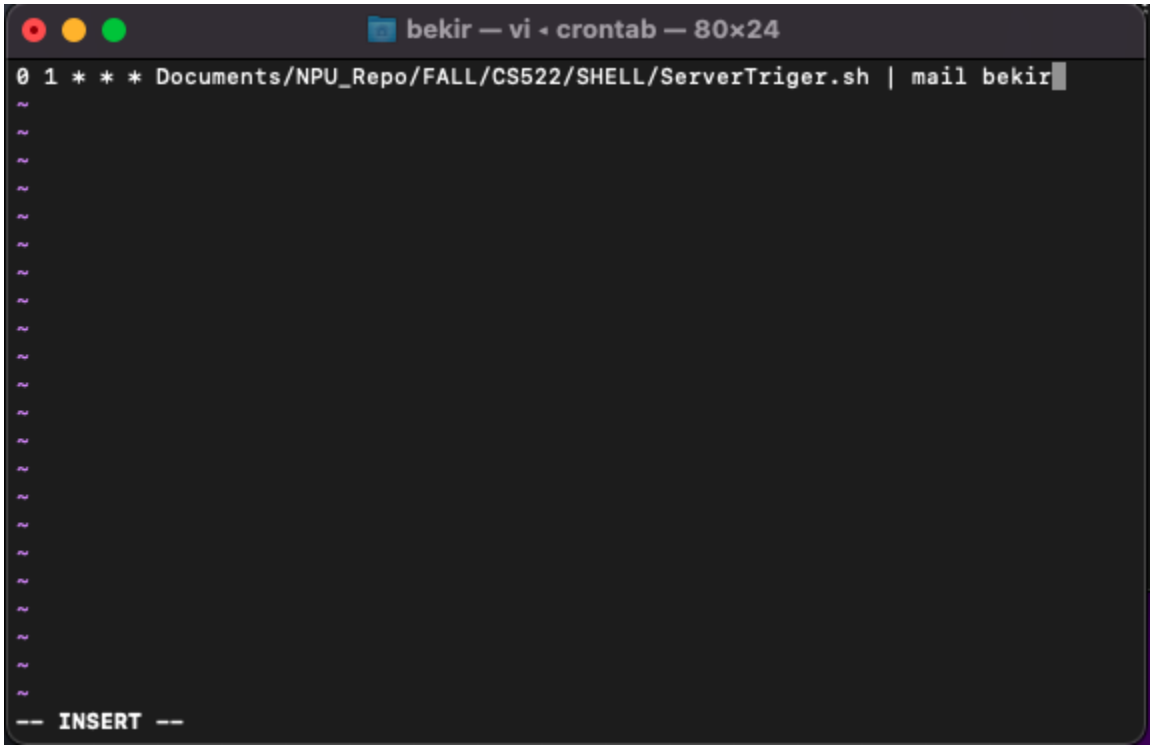
Listing the current cron jobs command:

```
crontab -l
```

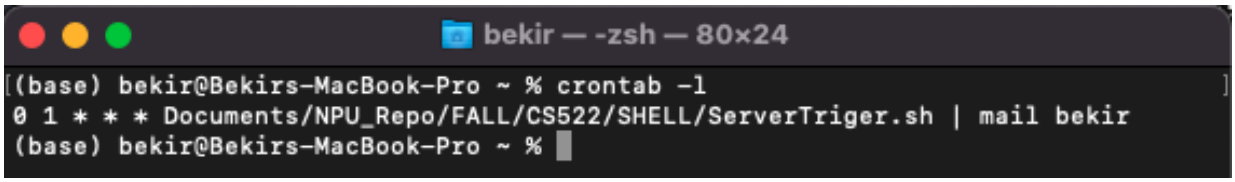
Creating cron job at 1 am everyday and send mail to a local user

```
0 1 * * * Documents/NPU_Repo/FALL/CS522/SHELL/ServerTriger.sh
| mail bekir
```

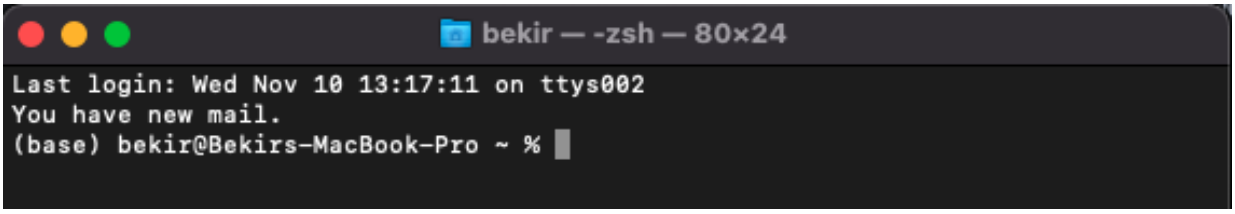
Creating a cron job:



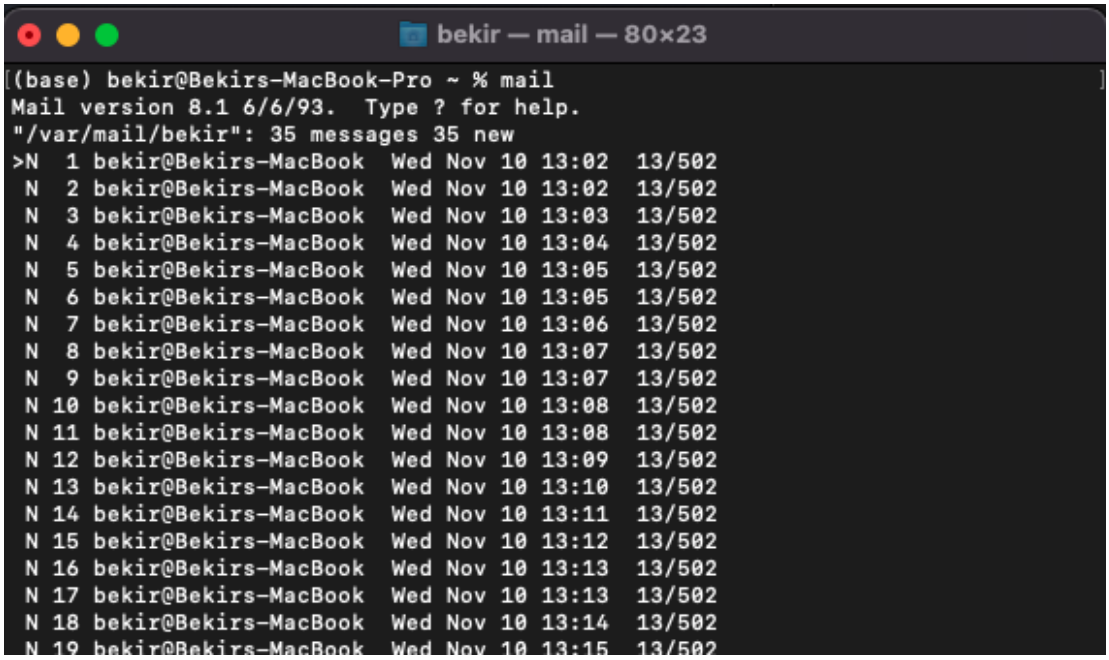
Listing cron jobs:

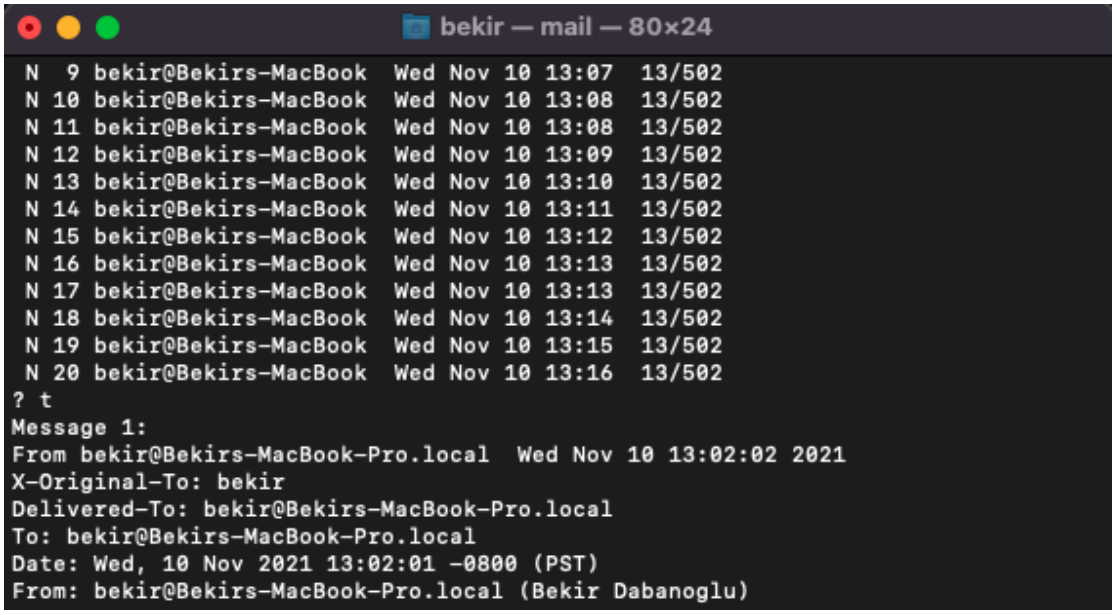


After the job ran. It will send an email to the user.



Command mail will bring all system mails. If we want to read the content of the mail we need to type -t after mail command:





Deleting specific message:

```
d [message number]
```