Bash Scripting

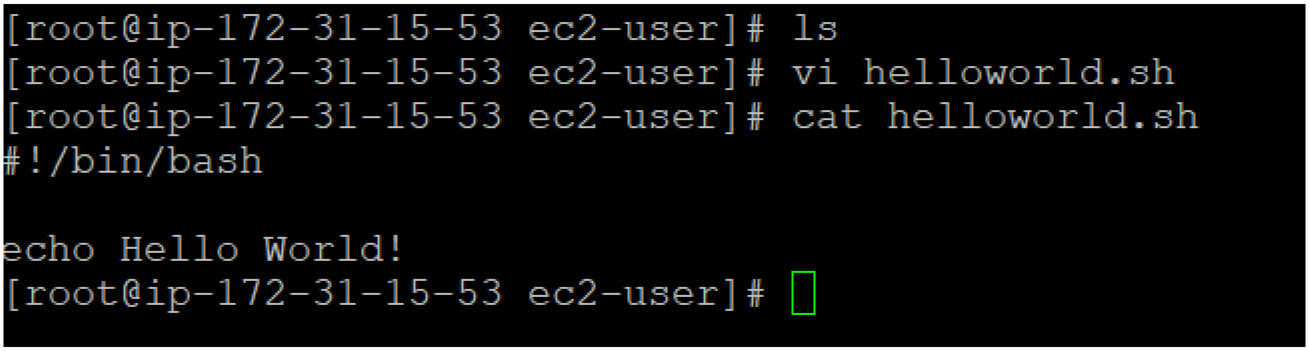
Shell Scripting is nothing but **a series of command in a simple text file ending with an extension '. sh'**. The '. sh' determines that it is a shell executable file. After writing a shell script, one would need to change the executable permission using the 'chmod function'.

1. To print hello world

```
#!/bin/bash
#shebang comment
echo Hello World!
```

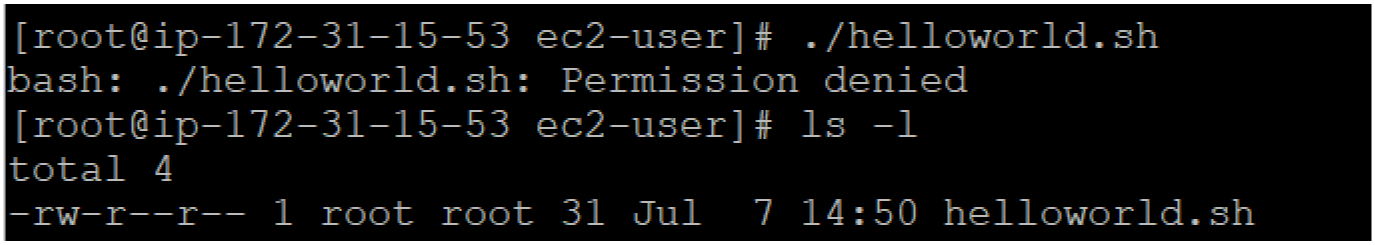
save this code as a file name “helloworld.sh”

```
vi helloworld.sh
cat helloworld.sh
./helloworld.sh
chmod 755 helloworld.sh
./helloworld.sh
```



```
[root@ip-172-31-15-53 ec2-user]# ls
[root@ip-172-31-15-53 ec2-user]# vi helloworld.sh
[root@ip-172-31-15-53 ec2-user]# cat helloworld.sh
#!/bin/bash

echo Hello World!
[root@ip-172-31-15-53 ec2-user]#
```

To run this command : ./helloworld.sh

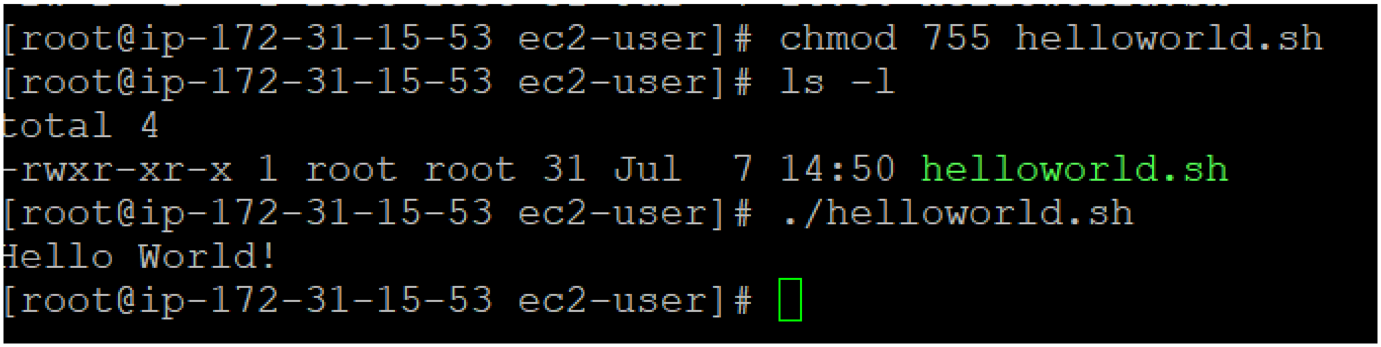


```
[root@ip-172-31-15-53 ec2-user]# ./helloworld.sh
bash: ./helloworld.sh: Permission denied
[root@ip-172-31-15-53 ec2-user]# ls -l
total 4
-rw-r--r-- 1 root root 31 Jul  7 14:50 helloworld.sh
```

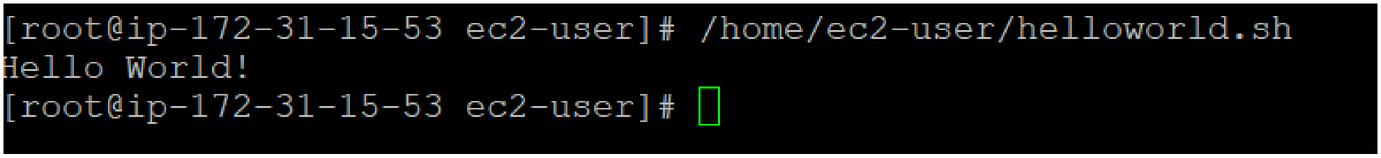
It will say permission denied as this script file don’t have execute permission.

so for providing execute permission : chmod 755 helloworld.sh

and then run command: ./helloworld.sh



```
[root@ip-172-31-15-53 ec2-user]# chmod 755 helloworld.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 4
-rwxr-xr-x 1 root root 31 Jul  7 14:50 helloworld.sh
[root@ip-172-31-15-53 ec2-user]# ./helloworld.sh
Hello World!
[root@ip-172-31-15-53 ec2-user]#
```



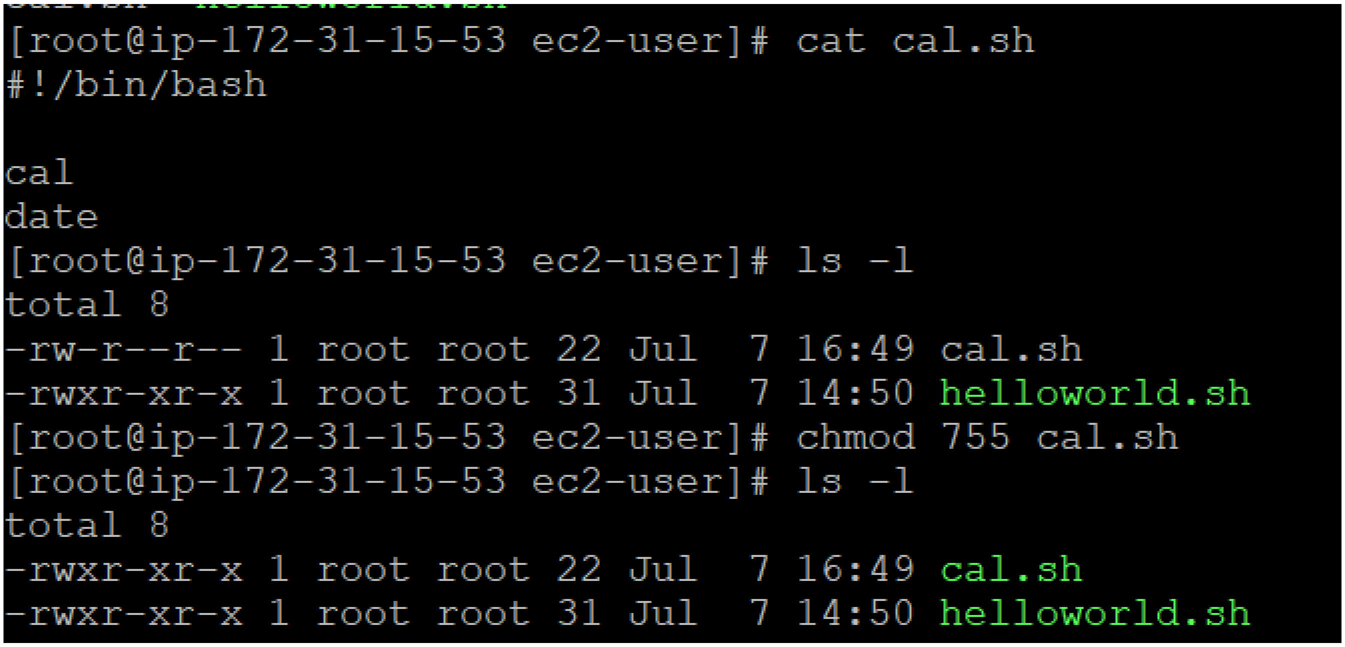
```
[root@ip-172-31-15-53 ec2-user]# /home/ec2-user/helloworld.sh
Hello World!
[root@ip-172-31-15-53 ec2-user]#
```

2. To print calender and date.

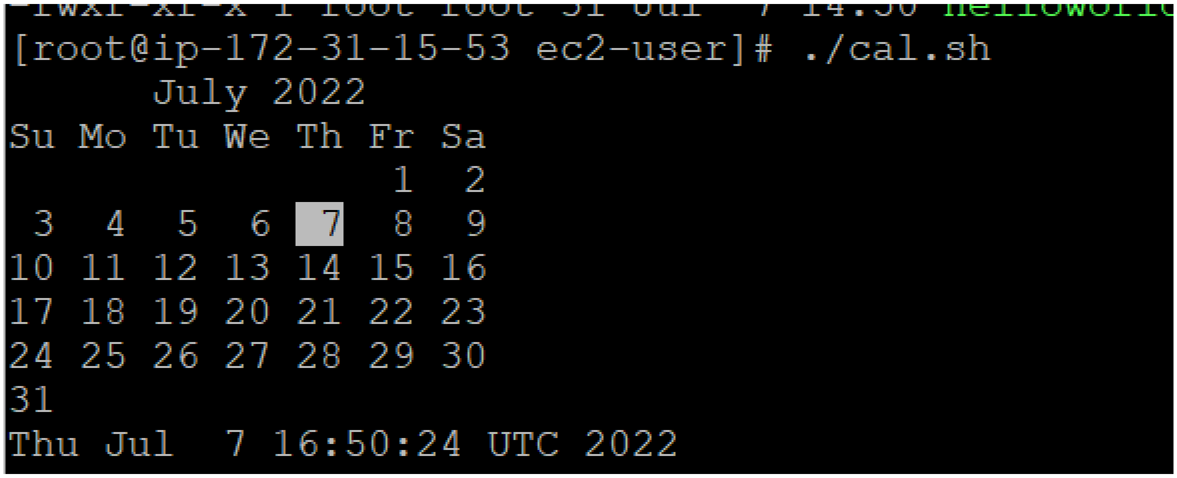
```
#!/bin/bash

cal
date
```

save this code in cal.sh file

```
vi cal.sh
cat cal.sh
ls -l
chmod 755 cal.sh
./cal.sh
```



```
[root@ip-172-31-15-53 ec2-user]# cat cal.sh
#!/bin/bash

cal
date
[root@ip-172-31-15-53 ec2-user]# ls -l
total 8
-rw-r--r-- 1 root root 22 Jul  7 16:49 cal.sh
-rwxr-xr-x 1 root root 31 Jul  7 14:50 helloworld.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 cal.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 8
-rwxr-xr-x 1 root root 22 Jul  7 16:49 cal.sh
-rwxr-xr-x 1 root root 31 Jul  7 14:50 helloworld.sh
```



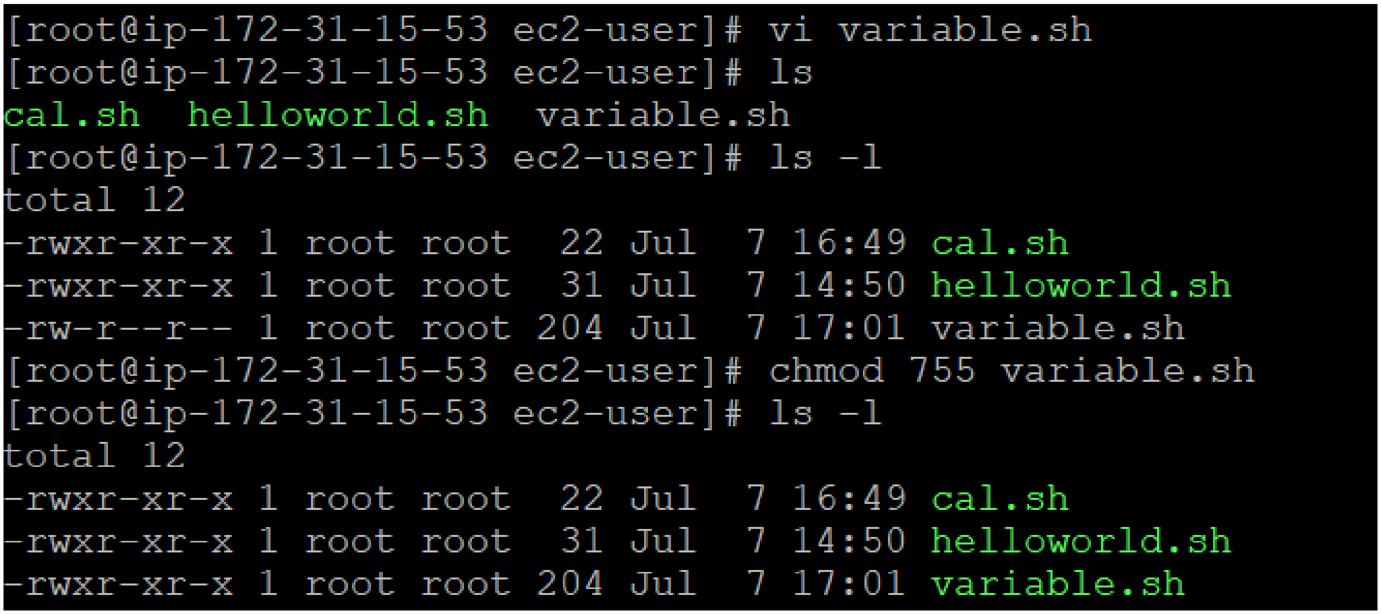
```
-rwxr-xr-x 1 root root 31 Jul  7 14:50 helloworl
[root@ip-172-31-15-53 ec2-user]# ./cal.sh
      July 2022
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
Thu Jul  7 16:50:24 UTC 2022
```

3. Variable

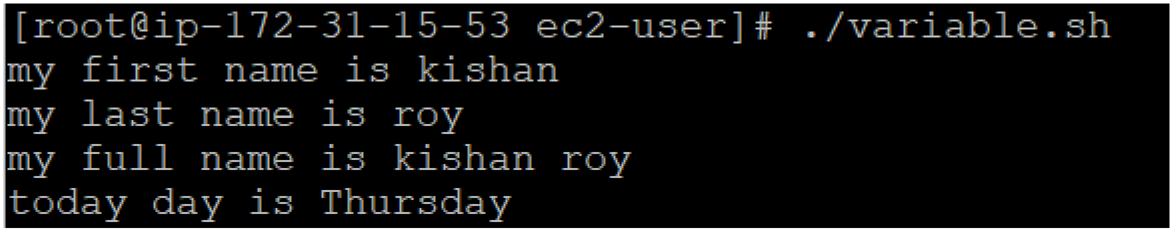
```
#!/bin/bash

firstname="kishan"
lastname="roy"
day=$(date +%A)


echo "my first name is $firstname"
echo "my last name is $lastname"
echo "my full name is $firstname $lastname"
echo "today day is $day"
```

save this as a variable.sh file

```
vi variable.sh
ls -l
chmod 755 variable.sh
ls -l
./variable.sh
```



```
[root@ip-172-31-15-53 ec2-user]# vi variable.sh
[root@ip-172-31-15-53 ec2-user]# ls
cal.sh  helloworld.sh  variable.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 12
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rw-r--r-- 1 root root 204 Jul  7 17:01 variable.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 variable.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 12
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rwxr-xr-x 1 root root 204 Jul  7 17:01 variable.sh
```



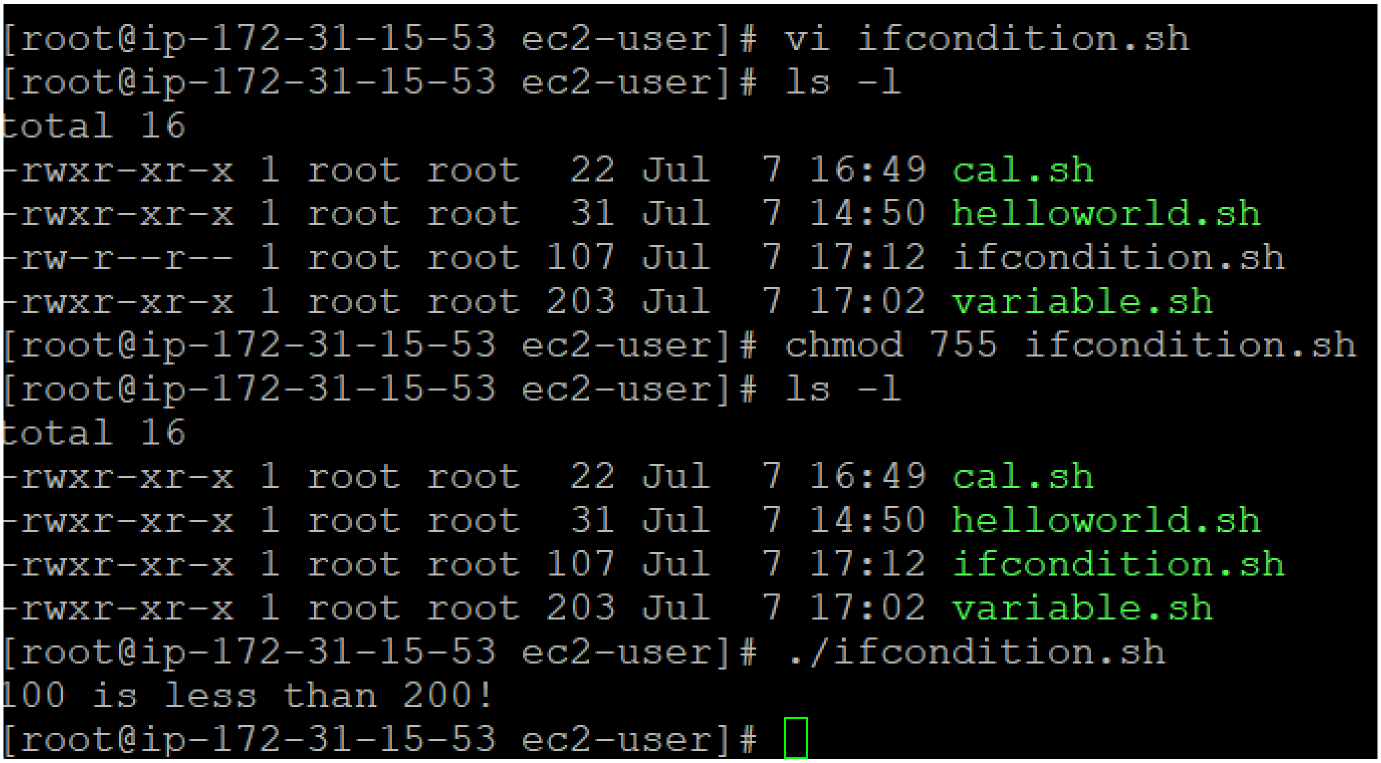
```
[root@ip-172-31-15-53 ec2-user]# ./variable.sh
my first name is kishan
my last name is roy
my full name is kishan roy
today day is Thursday
```

4. IF condition

```
#!/bin/bash

num_a=100
num_b=200

if [ $num_a -lt $num_b ]; then
    echo "$num_a is less than $num_b!"
fi
```

save this file with name ifcondition.sh, it will check where num_a is less than num_b or not.

```
vi ifcondition.sh
ls -l
chmod 755 ifcondition.sh
ls -l
./ifcondition.sh
```



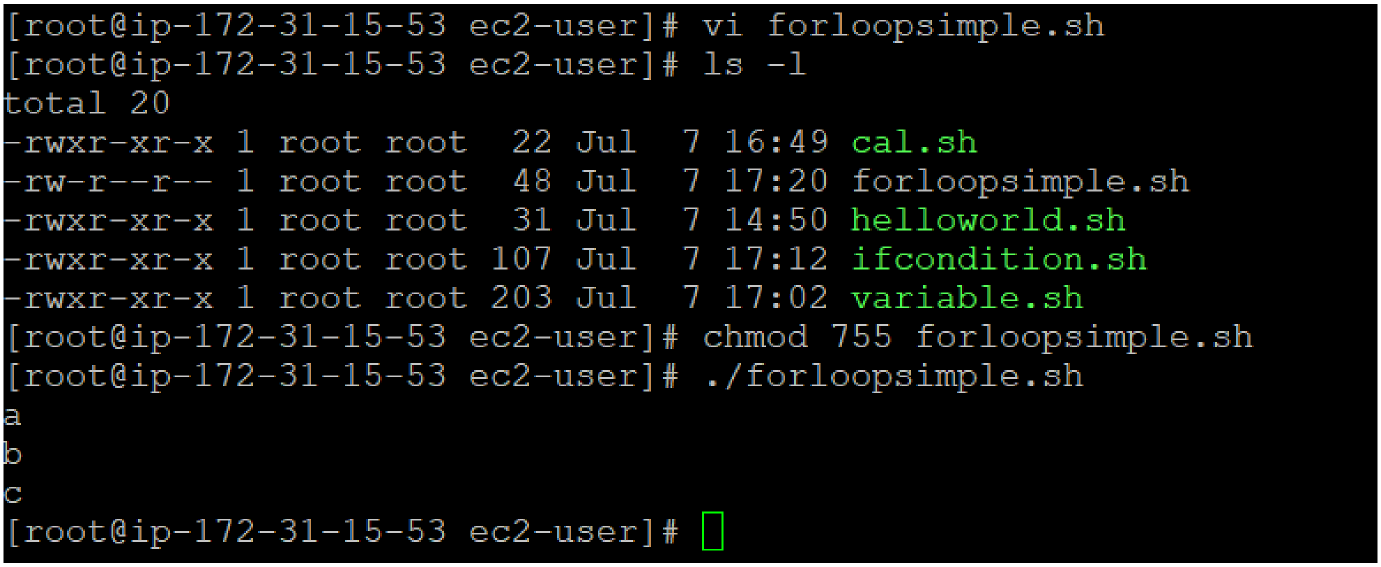
```
[root@ip-172-31-15-53 ec2-user]# vi ifcondition.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 16
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rw-r--r-- 1 root root 107 Jul  7 17:12 ifcondition.sh
-rwxr-xr-x 1 root root 203 Jul  7 17:02 variable.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 ifcondition.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 16
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rwxr-xr-x 1 root root 107 Jul  7 17:12 ifcondition.sh
-rwxr-xr-x 1 root root 203 Jul  7 17:02 variable.sh
[root@ip-172-31-15-53 ec2-user]# ./ifcondition.sh
100 is less than 200!
[root@ip-172-31-15-53 ec2-user]#
```

5. For loop simple

```
#!/bin/bash

for n in a b c;
do
   echo $n
done
```

save this file with name “forloopsimple.sh”

```
vi forloopsimple.sh
ls -l
chmod 755 forloopsimple.sh
ls -l
./forloopsimple.sh
```



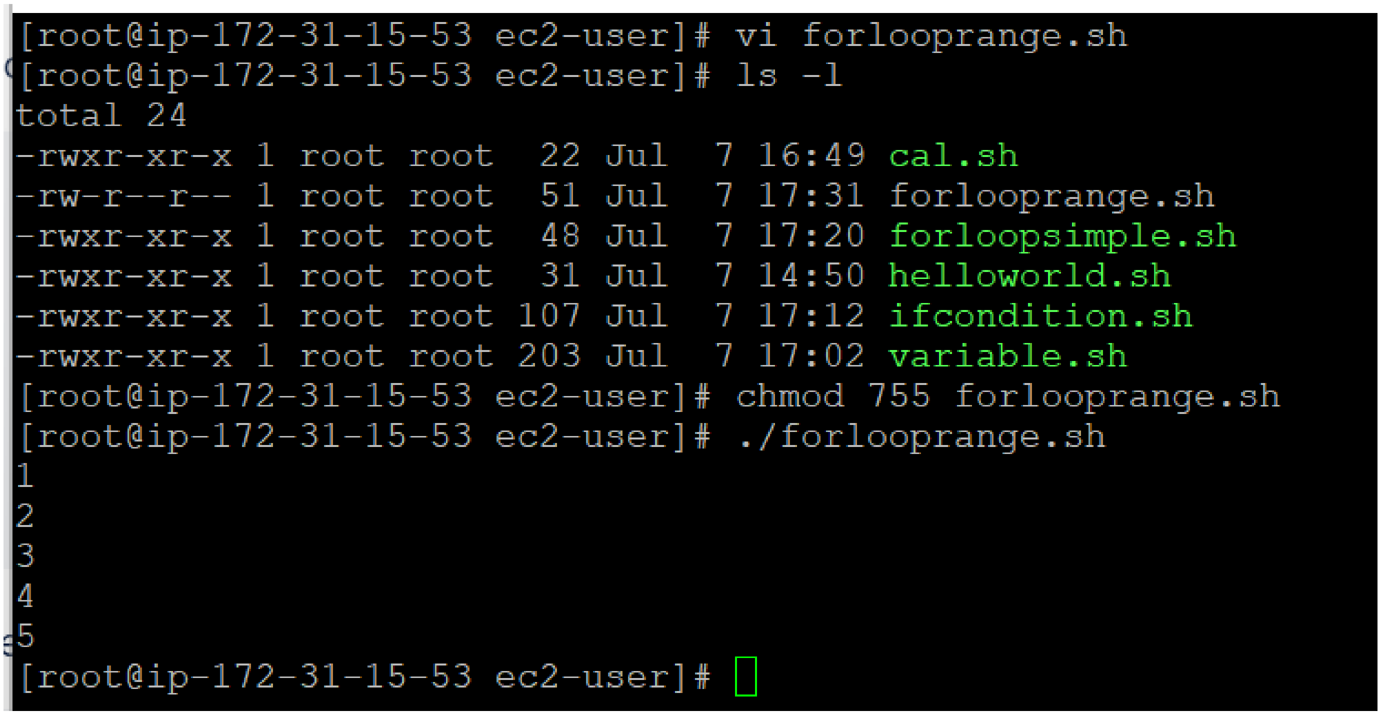
```
[root@ip-172-31-15-53 ec2-user]# vi forloopsimple.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 20
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rw-r--r-- 1 root root  48 Jul  7 17:20 forloopsimple.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rwxr-xr-x 1 root root 107 Jul  7 17:12 ifcondition.sh
-rwxr-xr-x 1 root root 203 Jul  7 17:02 variable.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 forloopsimple.sh
[root@ip-172-31-15-53 ec2-user]# ./forloopsimple.sh
a
b
c
[root@ip-172-31-15-53 ec2-user]# 
```

6. For loop range based (by default increment is 1)

```
#!/bin/bash

for n in {1..5};
do
    echo $n
done
```

save this file with name forlooprange.sh

```
vi forlooprange.sh
ls -l
chmod 755 forlooprange.sh
ls -l
./forlooprange.sh
```



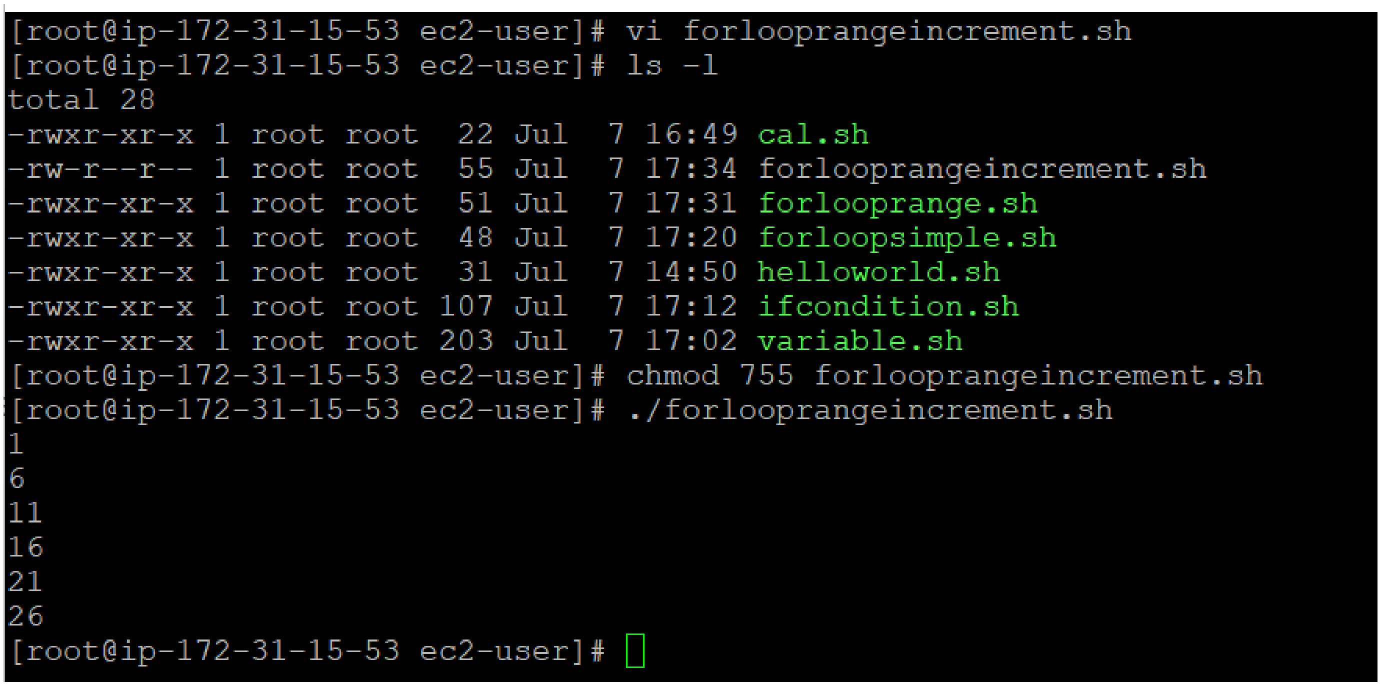
```
[root@ip-172-31-15-53 ec2-user]# vi forlooprange.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 24
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rw-r--r-- 1 root root  51 Jul  7 17:31 forlooprange.sh
-rwxr-xr-x 1 root root  48 Jul  7 17:20 forloopsimple.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rwxr-xr-x 1 root root 107 Jul  7 17:12 ifcondition.sh
-rwxr-xr-x 1 root root 203 Jul  7 17:02 variable.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 forlooprange.sh
[root@ip-172-31-15-53 ec2-user]# ./forlooprange.sh
1
2
3
4
5
[root@ip-172-31-15-53 ec2-user]#
```

7. For loop range based with increment as 5

```
#!/bin/bash

for n in {1..30..5};
do
    echo $n
done
```

save this file with name forlooprangeincrement.sh

```
vi forlooprangeincrement.sh
ls -l
chmod 755 forlooprangeincrement.sh
ls -l
./forlooprangeincrement.sh
```



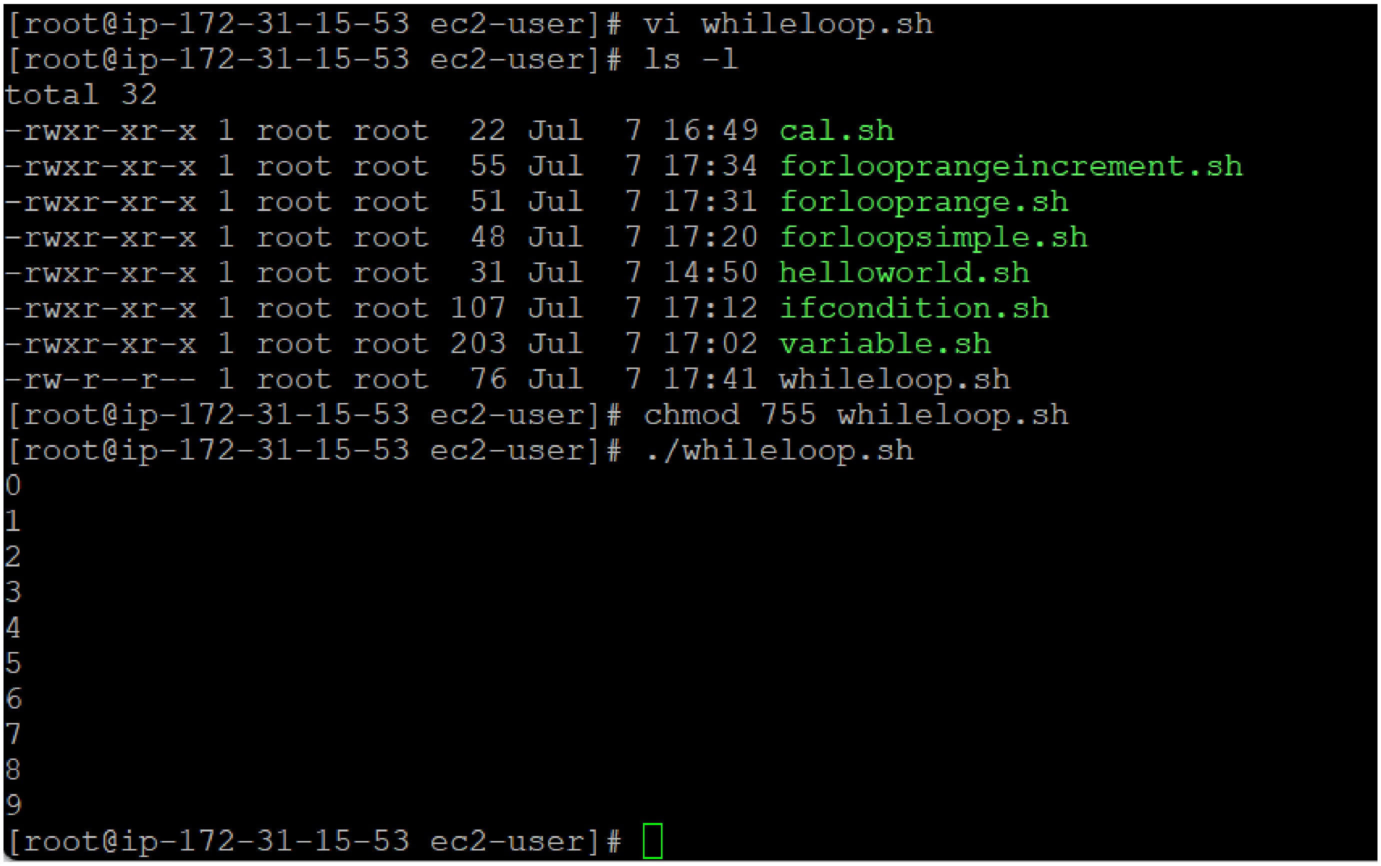
```
[root@ip-172-31-15-53 ec2-user]# vi forlooprangeincrement.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 28
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rw-r--r-- 1 root root  55 Jul  7 17:34 forlooprangeincrement.sh
-rwxr-xr-x 1 root root  51 Jul  7 17:31 forlooprange.sh
-rwxr-xr-x 1 root root  48 Jul  7 17:20 forloopsimple.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rwxr-xr-x 1 root root 107 Jul  7 17:12 ifcondition.sh
-rwxr-xr-x 1 root root 203 Jul  7 17:02 variable.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 forlooprangeincrement.sh
[root@ip-172-31-15-53 ec2-user]# ./forlooprangeincrement.sh
1
6
11
16
21
26
[root@ip-172-31-15-53 ec2-user]# 
```

8. while loop

```
#!/bin/bash

a=0

while [ $a -lt 10 ]
do
   echo $a
   a=`expr $a + 1`
done
```

save the file with name whileloop.sh

```
vi whileloop.sh
ls -l
chmod 755 whileloop.sh
ls -l
./whileloop.sh
```



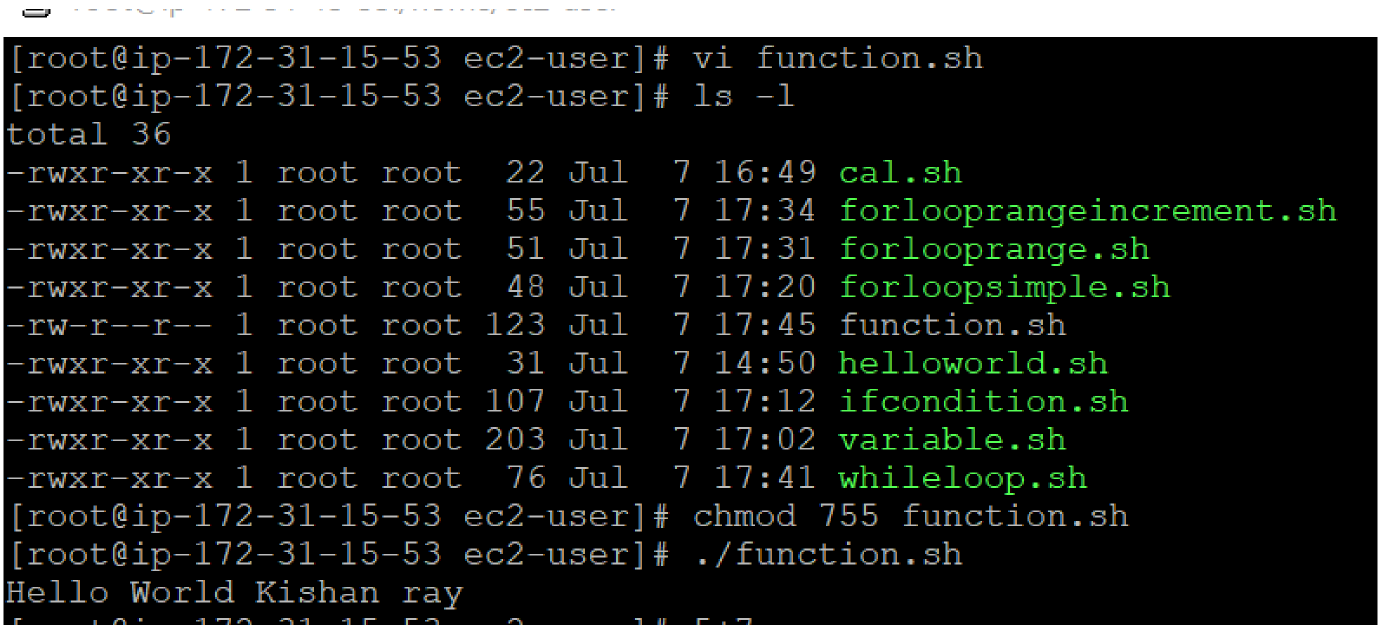
```
[root@ip-172-31-15-53 ec2-user]# vi whileloop.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 32
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rwxr-xr-x 1 root root  55 Jul  7 17:34 forlooprangeincrement.sh
-rwxr-xr-x 1 root root  51 Jul  7 17:31 forlooprange.sh
-rwxr-xr-x 1 root root  48 Jul  7 17:20 forloopsimple.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rwxr-xr-x 1 root root 107 Jul  7 17:12 ifcondition.sh
-rwxr-xr-x 1 root root 203 Jul  7 17:02 variable.sh
-rw-r--r-- 1 root root  76 Jul  7 17:41 whileloop.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 whileloop.sh
[root@ip-172-31-15-53 ec2-user]# ./whileloop.sh
0
1
2
3
4
5
6
7
8
9
[root@ip-172-31-15-53 ec2-user]#
```

9. Function

```
#!/bin/bash

# Define your function here
Hello () {
   echo "Hello World $1 $2"
}

# Invoke your function
Hello Kishan ray
```

save this file with name function.sh

```
vi function.sh
ls -l
chmod 755 function.sh
ls -l
./function.sh
```



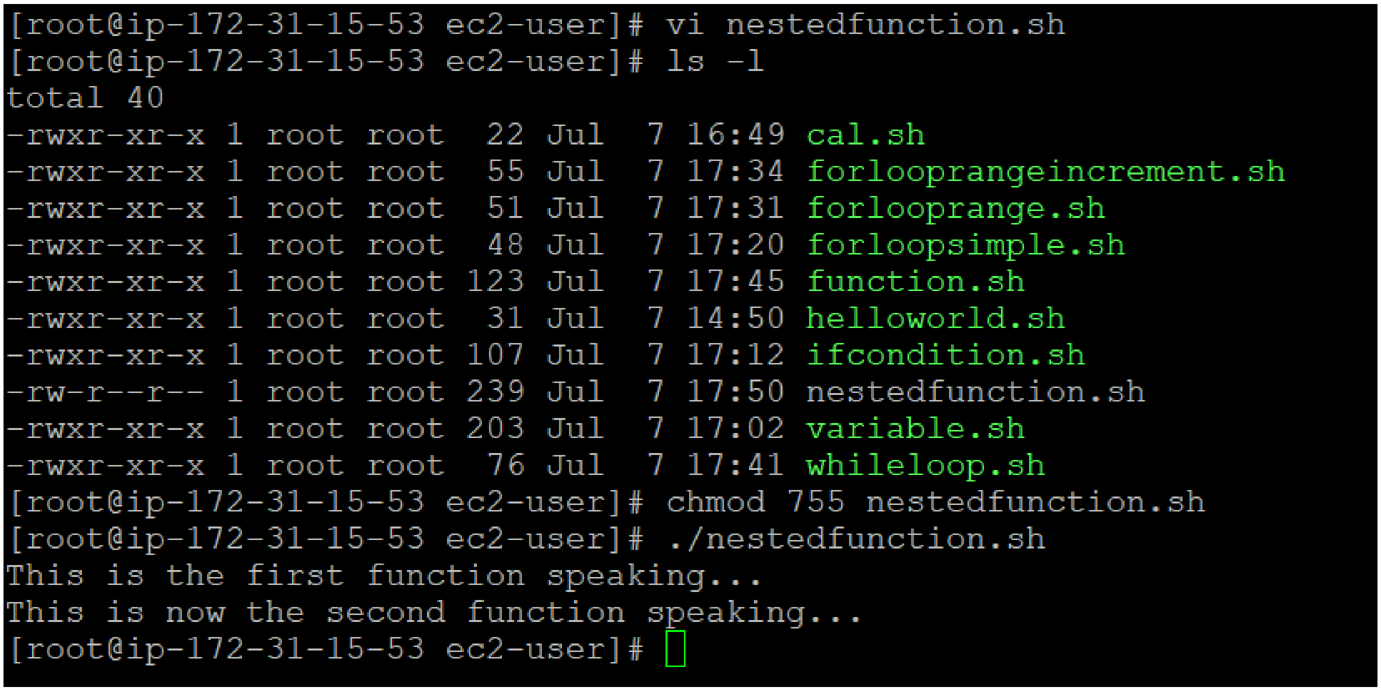
```
[root@ip-172-31-15-53 ec2-user]# vi function.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 36
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rwxr-xr-x 1 root root  55 Jul  7 17:34 forlooprangeincrement.sh
-rwxr-xr-x 1 root root  51 Jul  7 17:31 forlooprange.sh
-rwxr-xr-x 1 root root  48 Jul  7 17:20 forloopsimple.sh
-rw-r--r-- 1 root root 123 Jul  7 17:45 function.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rwxr-xr-x 1 root root 107 Jul  7 17:12 ifcondition.sh
-rwxr-xr-x 1 root root 203 Jul  7 17:02 variable.sh
-rwxr-xr-x 1 root root  76 Jul  7 17:41 whileloop.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 function.sh
[root@ip-172-31-15-53 ec2-user]# ./function.sh
Hello World Kishan ray
```

10. Nested Function

```
#!/bin/bash

# Calling one function from another
number_one () {
   echo "This is the first function speaking..."
   number_two
}

number_two () {
   echo "This is now the second function speaking..."
}

# Calling function one.
number_one
```

save this file with name nestedfunction.sh

```
vi nestedfunction.sh
ls -l
chmod 755 nestedfunction.sh
ls -l
./nestedfunction.sh
```



```
[root@ip-172-31-15-53 ec2-user]# vi nestedfunction.sh
[root@ip-172-31-15-53 ec2-user]# ls -l
total 40
-rwxr-xr-x 1 root root  22 Jul  7 16:49 cal.sh
-rwxr-xr-x 1 root root  55 Jul  7 17:34 forlooprangeincrement.sh
-rwxr-xr-x 1 root root  51 Jul  7 17:31 forlooprange.sh
-rwxr-xr-x 1 root root  48 Jul  7 17:20 forloopsimple.sh
-rwxr-xr-x 1 root root 123 Jul  7 17:45 function.sh
-rwxr-xr-x 1 root root  31 Jul  7 14:50 helloworld.sh
-rwxr-xr-x 1 root root 107 Jul  7 17:12 ifcondition.sh
-rw-r--r-- 1 root root 239 Jul  7 17:50 nestedfunction.sh
-rwxr-xr-x 1 root root 203 Jul  7 17:02 variable.sh
-rwxr-xr-x 1 root root  76 Jul  7 17:41 whileloop.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 nestedfunction.sh
[root@ip-172-31-15-53 ec2-user]# ./nestedfunction.sh
This is the first function speaking...
This is now the second function speaking...
[root@ip-172-31-15-53 ec2-user]# 
```

11. Reading file content using bash script

Create a bash file and add the following code to read the content of a particular file. Here, an existing filename is stored in **$filename** variable and **$n** variable is used to keep the value of the line number of that file. Like previous example, **while** loop is used to read this file with line number.

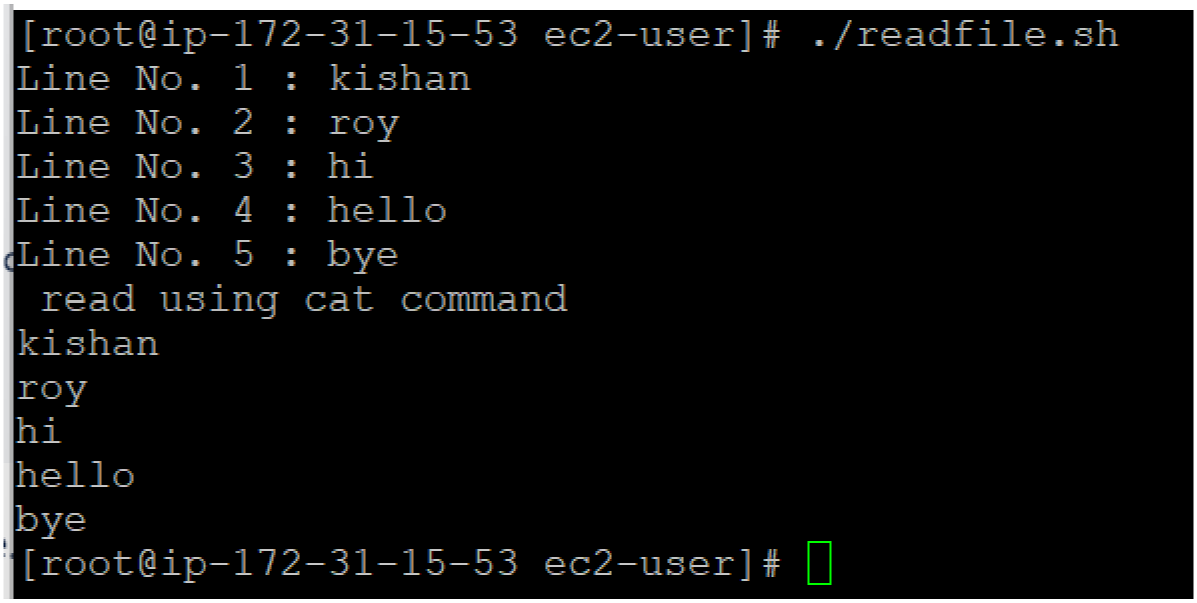
add this lines in abc.txt file first.

```
kishan
roy
hi
hello
bye
```

```
#!/bin/bash
filename='abc.txt'
n=1
while read line; do
# reading each line
echo "Line No. $n : $line"
n=$((n+1))
done < $filename

echo " read using cat command"
cat $filename
```

save this file with name readfile.sh

```
vi readfile.sh
ls -l
chmod 755 readfile.sh
./readfile.sh
```



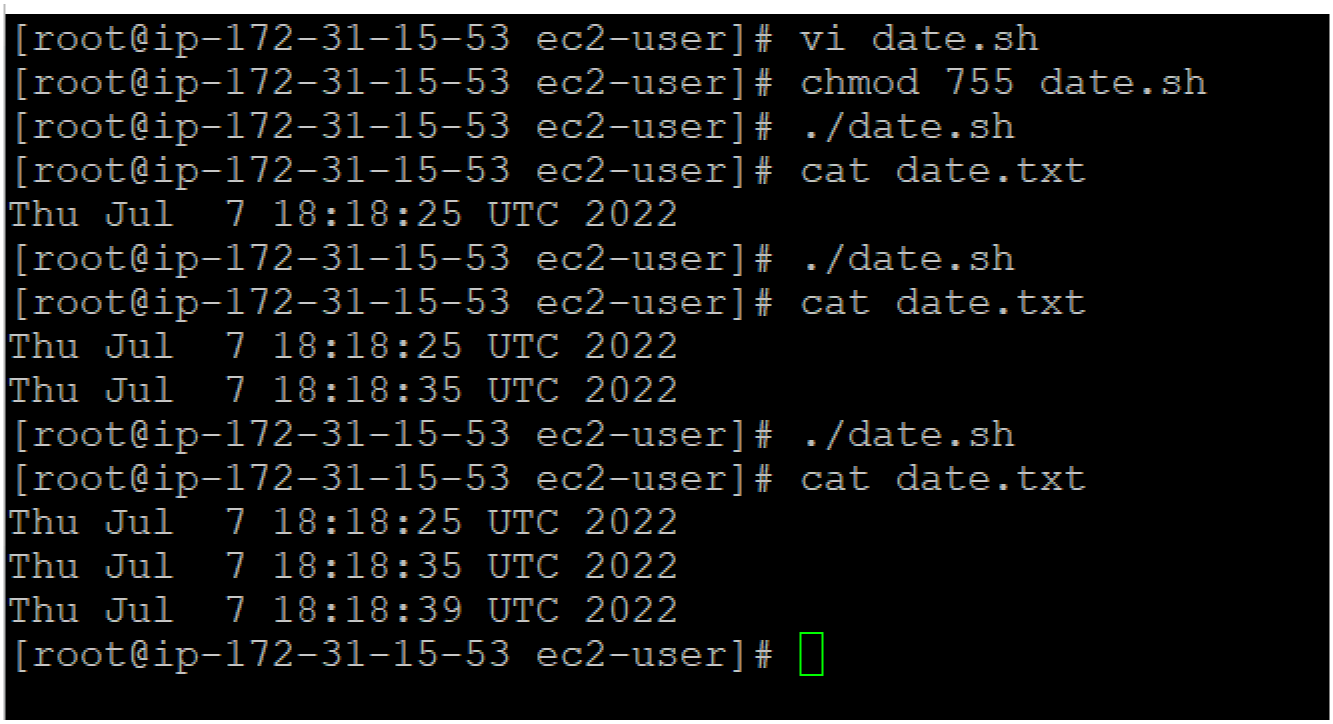
```
[root@ip-172-31-15-53 ec2-user]# ./readfile.sh
Line No. 1 : kishan
Line No. 2 : roy
Line No. 3 : hi
Line No. 4 : hello
Line No. 5 : bye
 read using cat command
kishan
roy
hi
hello
bye
[root@ip-172-31-15-53 ec2-user]# 
```

12. writing file content using script

```
#!/bin/bash

date >> date.txt
```

save this file with name date.sh, it will create a file with name date.txt and will keep adding the date to this file whenever we will execute the script.

```
vi date.sh
ls -l
chmod 755 date.sh
./date.sh
cat date.txt
```



```
[root@ip-172-31-15-53 ec2-user]# vi date.sh
[root@ip-172-31-15-53 ec2-user]# chmod 755 date.sh
[root@ip-172-31-15-53 ec2-user]# ./date.sh
[root@ip-172-31-15-53 ec2-user]# cat date.txt
Thu Jul  7 18:18:25 UTC 2022
[root@ip-172-31-15-53 ec2-user]# ./date.sh
[root@ip-172-31-15-53 ec2-user]# cat date.txt
Thu Jul  7 18:18:25 UTC 2022
Thu Jul  7 18:18:35 UTC 2022
[root@ip-172-31-15-53 ec2-user]# ./date.sh
[root@ip-172-31-15-53 ec2-user]# cat date.txt
Thu Jul  7 18:18:25 UTC 2022
Thu Jul  7 18:18:35 UTC 2022
Thu Jul  7 18:18:39 UTC 2022
[root@ip-172-31-15-53 ec2-user]#
```