

Assignment

Assignment Date: 05-11-2022

Submitted by: Govarthan A(govasana07@gmail.com)

Assignment 1 :- How to upload HTML web pages on Apache2 web server in EC-2 Instance?

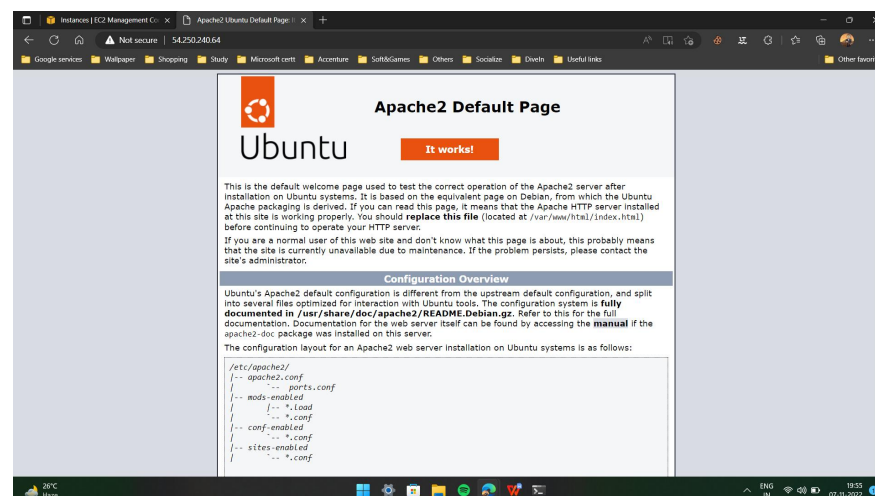
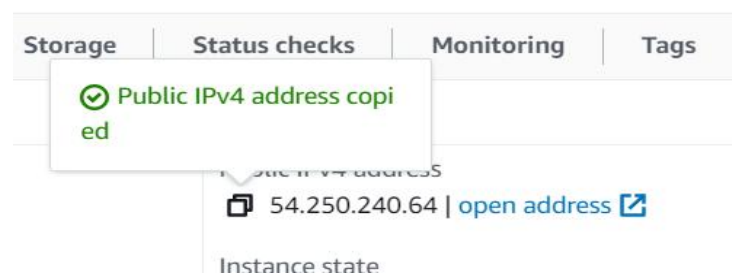
Step1: Initially I created a Linux server using Ubuntu image under EC2 section once the server is up then I connected to that server via SSH

```
ubuntu@ip-172-31-32-38: ~  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Mon Nov  7 14:12:19 UTC 2022  
  
System load:  0.18798828125    Processes:            102  
Usage of /:   19.5% of 7.57GB   Users logged in:     0  
Memory usage: 20%              IPv4 address for eth0: 172.31.32.38  
Swap usage:   0%  
  
0 updates can be applied immediately.  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-172-31-32-38:~$
```

Step2: Once connected I updated the Linux server using **sudo apt-get update** command

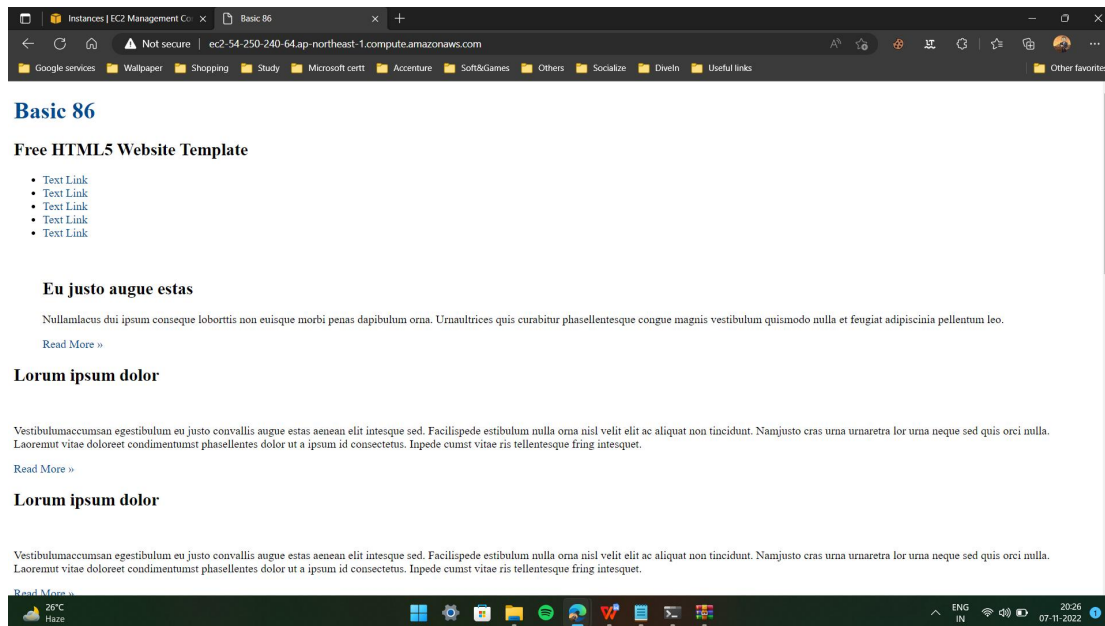
Step3: After updating the machine I installed the apache2 server using **sudo apt-get install apache2**

Step4: Now I went to my instance and took the public ip of it and opened it



I got default page of Apache server now I'm going to put new HTML file to load new page for this IP address

Step5: This default page HTML will be located in `/var/www/html/index.html` so I moved to this path and started editing `index.html` to load new page



Step6: once edited the page I reloaded the tab to see updated page

Assignment 2 : Create readfile.sh in which you can read the information of PWD like size, permission, date time etc.

So, I created a file called readfile.sh and provided permission to it to execute using `chmod +x readfile.sh`

And added below lines to get the Present working directory's size, permission and date&time

```

gova@GovaAnand: ~
GNU nano 4.8 readfile.sh
#!/bin/sh

echo "Size of this directory is $(du -sh)"
echo "Permission in this directory is $(ls -l) "
echo "Date and time is $(date)"

```

Output is:

```

gova@GovaAnand: ~
gova@GovaAnand:~$ ls
downloads readfile.sh shell.sh shell.sh.save
gova@GovaAnand:~$ nano readfile.sh
gova@GovaAnand:~$ ./readfile.sh
Size of this directory is 84K
Permission in this directory is total 16
drwxr-xr-x 5 gova gova 4096 Nov  7 11:51 downloads
-rwxr-xr-x 1 gova gova 135 Nov 11 11:47 readfile.sh
-rwxr-xr-x 1 gova gova  88 Nov  6 19:42 shell.sh
-rwxr-xr-x 1 gova gova  80 Nov  6 19:43 shell.sh.save
Date and time is Fri Nov 11 11:48:28 IST 2022
gova@GovaAnand:~$

```

Assignment 3 : Take an input of name from user and print Have a great day ahead {name}

Created new sh file called name.sh and added the below commands

```
gova@GovaAnand: ~  
GNU nano 4.8 name.sh  
#!/bin/sh  
  
echo "What is your name?"  
read name  
echo "Have a Great day Ahead, $name"
```

And the output is:

```
gova@GovaAnand: ~  
gova@GovaAnand:~$ touch name.sh  
gova@GovaAnand:~$ chmod +x name.sh  
gova@GovaAnand:~$ nano name.sh  
gova@GovaAnand:~$ nano name.sh  
gova@GovaAnand:~$ ./name.sh  
What is your name?  
Govarthan Anandhan  
Have a Great day Ahead, Govarthan Anandhan  
gova@GovaAnand:~$
```

Assignment 4 : Let's take a scenario of fintech app program in which we want to have three separate outputs for 3 different situations:

I created the fintech.sh file and added the below condition inside that

```
gova@GovaAnand: ~  
GNU nano 4.8 fintech.sh  
#!/bin/bash  
echo "Enter the balance number"  
read balance  
if [ $balance -lt 0 ]; then  
    echo "Balance is less than zero so add some fund to avoid penalty"  
elif [ $balance -eq 0 ]; then  
    echo "Balance is zero, please add funds"  
else  
    echo "Your Balance is above zero"  
fi
```

And the output is:

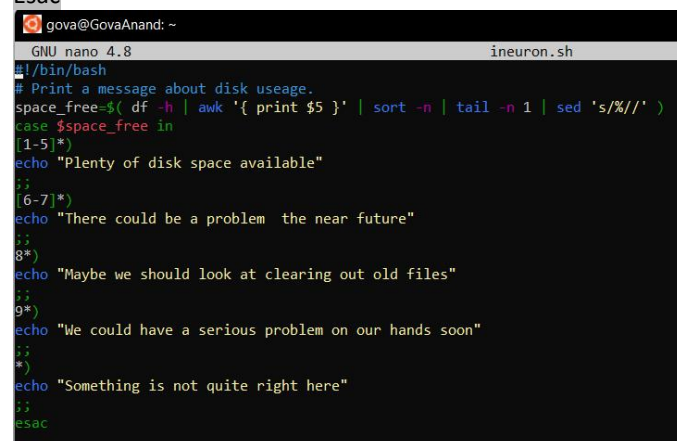
```
gova@GovaAnand: ~  
gova@GovaAnand:~$ ls  
downloads fintech.sh name.sh readfile.sh shell.sh shell.sh.save  
gova@GovaAnand:~$ nano fintech.sh  
gova@GovaAnand:~$ ./fintech.sh  
Enter the balance number  
0  
Balance is zero, please add funds  
gova@GovaAnand:~$ ./fintech.sh  
Enter the balance number  
5  
Your Balance is above zero  
gova@GovaAnand:~$ ./fintech.sh  
Enter the balance number  
-5  
Balance is less than zero so add some fund to avoid penalty  
gova@GovaAnand:~$
```

Assignment 5 : Debug and define briefly about the following program :-

```
#!/bin/bash
# Print a message about disk useage.space_free=$( df -h | awk '{ print $5 }' | sort -n | tail -n
1 | sed 's/%/' )
case $space_free in
[1-5]*)
echo Plenty of disk space available
[6-7]*)
echo There could be a problem in the near future
8*)
echo Maybe we should look at clearing out old files
9*)
echo We could have a serious problem on our hands soon
*)
echo Something is not quite right here
;;
Esac
```

Since the Above program has some syntax error I resolved those and correct program is

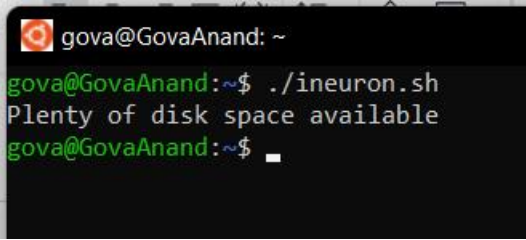
```
#!/bin/bash
# Print a message about disk useage.
space_free=$( df -h | awk '{ print $5 }' | sort -n | tail -n 1 | sed 's/%/' )
case $space_free in
[1-5]*)
echo "Plenty of disk space available"
;;
[6-7]*)
echo "There could be a problem the near future"
;;
8*)
echo "Maybe we should look at clearing out old files"
;;
9*)
echo "We could have a serious problem on our hands soon"
;;
*)
echo "Something is not quite right here"
;;
Esac
```



The screenshot shows a terminal window with the prompt 'gova@GovaAnand: ~'. The nano editor is open, editing a file named 'ineuron.sh'. The script content is the corrected version of the shell script, with syntax errors fixed. The script uses double quotes for echo statements and includes a final 'Esac' keyword. The terminal background is black, and the nano editor interface is visible at the top and bottom of the window.

```
gova@GovaAnand: ~
GNU nano 4.8      ineuron.sh
#!/bin/bash
# Print a message about disk useage.
space_free=$( df -h | awk '{ print $5 }' | sort -n | tail -n 1 | sed 's/%/' )
case $space_free in
[1-5]*)
echo "Plenty of disk space available"
;;
[6-7]*)
echo "There could be a problem the near future"
;;
8*)
echo "Maybe we should look at clearing out old files"
;;
9*)
echo "We could have a serious problem on our hands soon"
;;
*)
echo "Something is not quite right here"
;;
Esac
```

In this program actually it is checking for the disk free space here it takes last line (tail -n 1) and print fifth field (awk '{print \$5}'
And in each case condition is added based on the condition falls we will get output
My output is :



```
gova@GovaAnand: ~  
gova@GovaAnand:~$ ./ineuron.sh  
Plenty of disk space available  
gova@GovaAnand:~$
```