Introduction to Cloud Computing (CS 524)

(Lab Assignment 1)
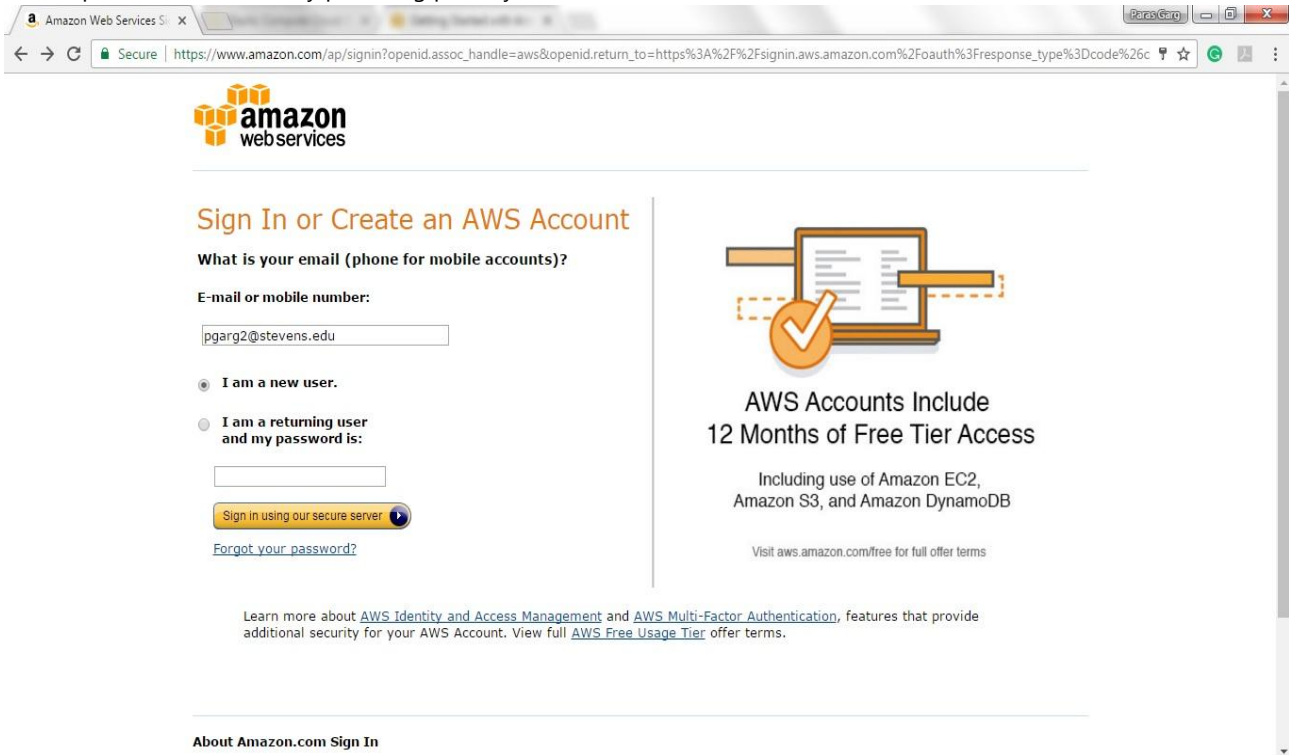
Prof. Igor Faynberg

Student Name: **Paras Garg**

Course Section: **CS 524-A**

Step for Creating an AWS account

- First step to create account by providing primary contact detail.

- Second step is to provide login credentials

- Third step is to provide full contact information

- After contact information, we have to provide payment information.

- Step after payment set up is to verify the identity by getting one time password.

- Next step is to select the support plan.

- After choosing plan, the final step is the confirmation.

- After creating account, we have to sign in to access the resources.

Step for Creating and Launching AWS instance

- After successfully login into AWS console, we would land on the user dashboard from where you find all options.
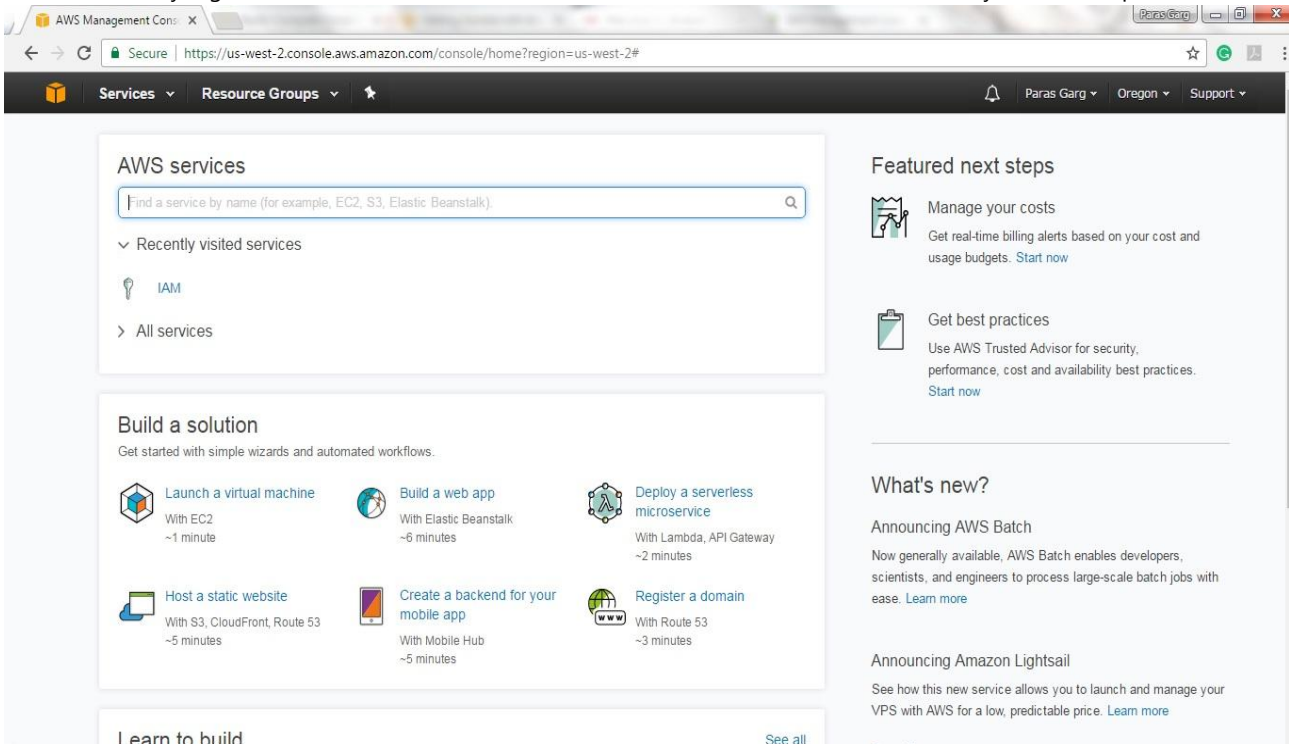
- On clicking EC2 service, EC2 Dashboard will open. Now we have to click on "Launch Instance" under Create Instance tab.
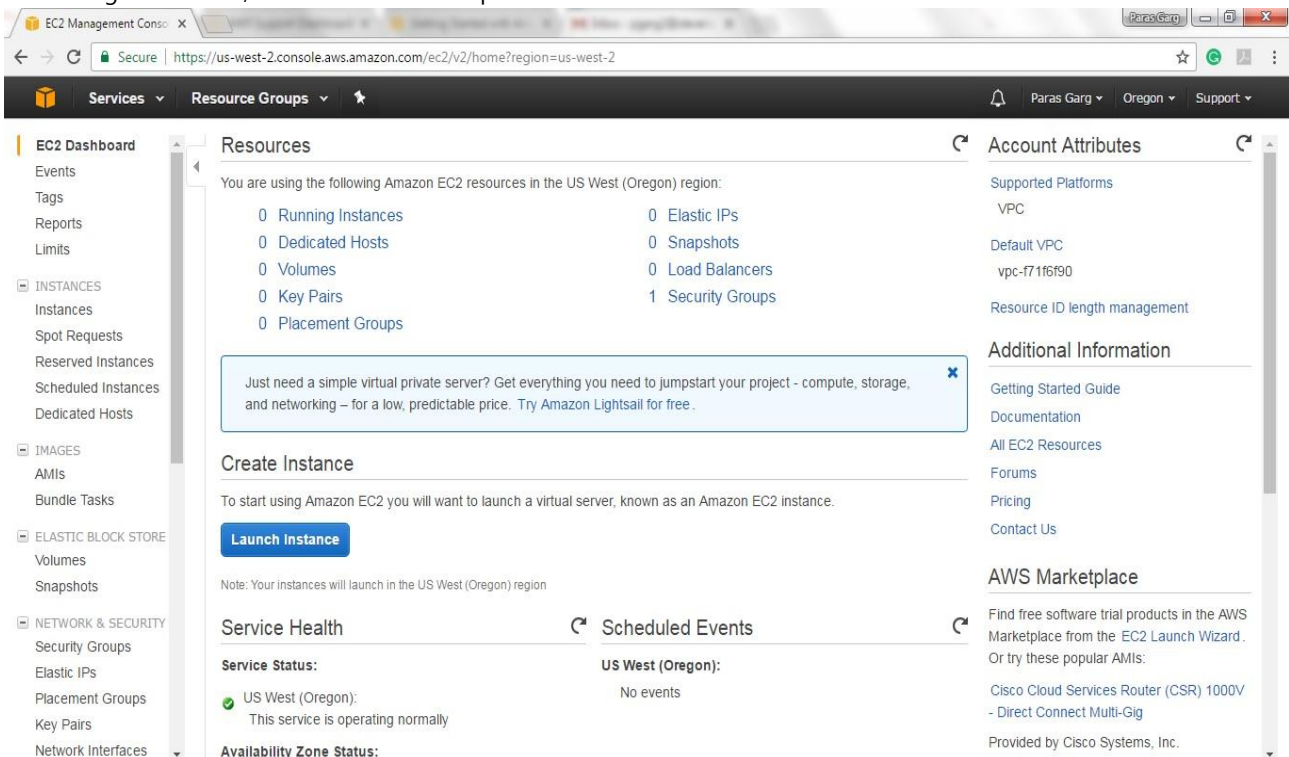
- After Launching Instance, first, we have to choose an Amazon Machine Image (AMI). Selecting Amazon Linux AMI (64-bit).

- After selecting AMI, we have the options to choose an Instance Type. Selecting **t2.micro** (free tier eligibility).

- As we have selected free tier instance type, we would jump to final step to Review Instance Launch.

- After reviewing, we have to create a new Key-Pair and have to download it into our local machine.

- After creating key-pair, our instance is ready to launch and we can view it by clicking View Instance.

- Now, we can view description about your instance and other modules. Now viewing Instances and its description.

- Here, we are checking Security Group and its description.

Step for Accessing AWS instance

- Ensuring read write permission on instance by executing below command
  $ chmod 400 ParasGarg_AW.pem

```
MINGW64:/c/Users/Paras Garg/documents/aws

Paras Garg@ParasGarg-HP MINGW64 ~/documents/aws
$ ls -lrt
total 8
-rw-r--r-- 1 Paras Garg 197121 1692 Mar  7 14:01 ParasGarg_AWS.pem
-r--r--r-- 1 Paras Garg 197121 1486 Mar  7 14:59 ParasGarg_AWS.ppk

Paras Garg@ParasGarg-HP MINGW64 ~/documents/aws
$ chmod 400 ParasGarg_AWS.pem

Paras Garg@ParasGarg-HP MINGW64 ~/documents/aws
$ ls -lrt
total 8
-r--r--r-- 1 Paras Garg 197121 1692 Mar  7 14:01 ParasGarg_AWS.pem
-r--r--r-- 1 Paras Garg 197121 1486 Mar  7 14:59 ParasGarg_AWS.ppk

Paras Garg@ParasGarg-HP MINGW64 ~/documents/aws
$
```
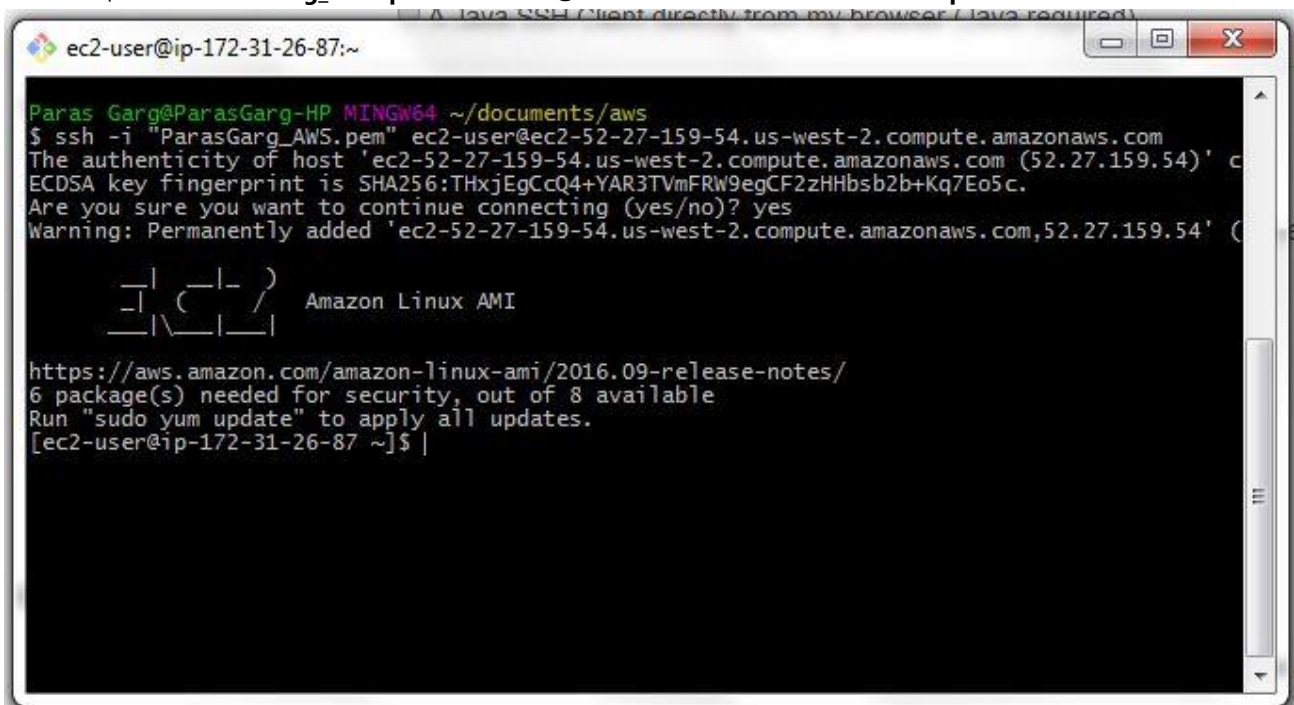
- Establishing connection with EC2 Instance by executing below command
  $ ssh –I "ParasGarg_AWS.pem" ec2-user@ec2-52-27-159-54.us-west-2.compute.amazonaws.com

```
ec2-user@ip-172-31-26-87:~

Paras Garg@ParasGarg-HP MINGW64 ~/documents/aws
$ ssh -i "ParasGarg_AWS.pem" ec2-user@ec2-52-27-159-54.us-west-2.compute.amazonaws.com
The authenticity of host 'ec2-52-27-159-54.us-west-2.compute.amazonaws.com (52.27.159.54)' c
ECDSA key fingerprint is SHA256:THxjEgCcQ4+YAR3TVmFRW9egCF2zHHbsb2b+Kq7Eo5c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-52-27-159-54.us-west-2.compute.amazonaws.com,52.27.159.54' (

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
6 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-87 ~]$
```
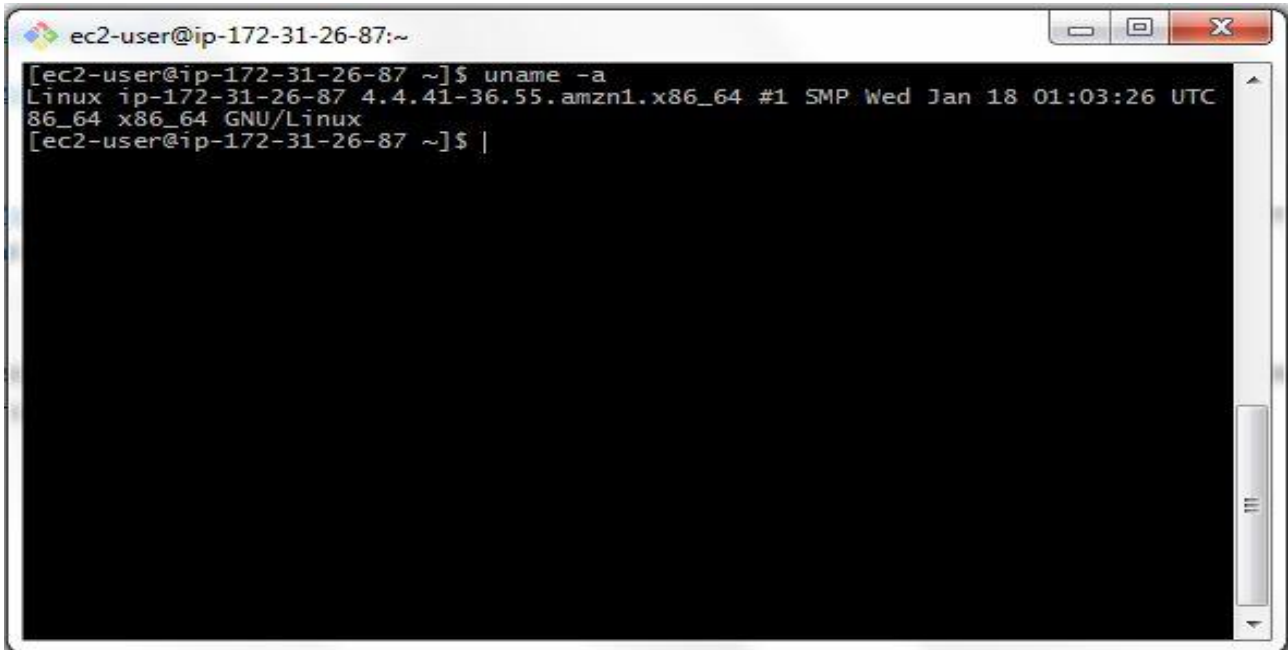
AWS instance Commands
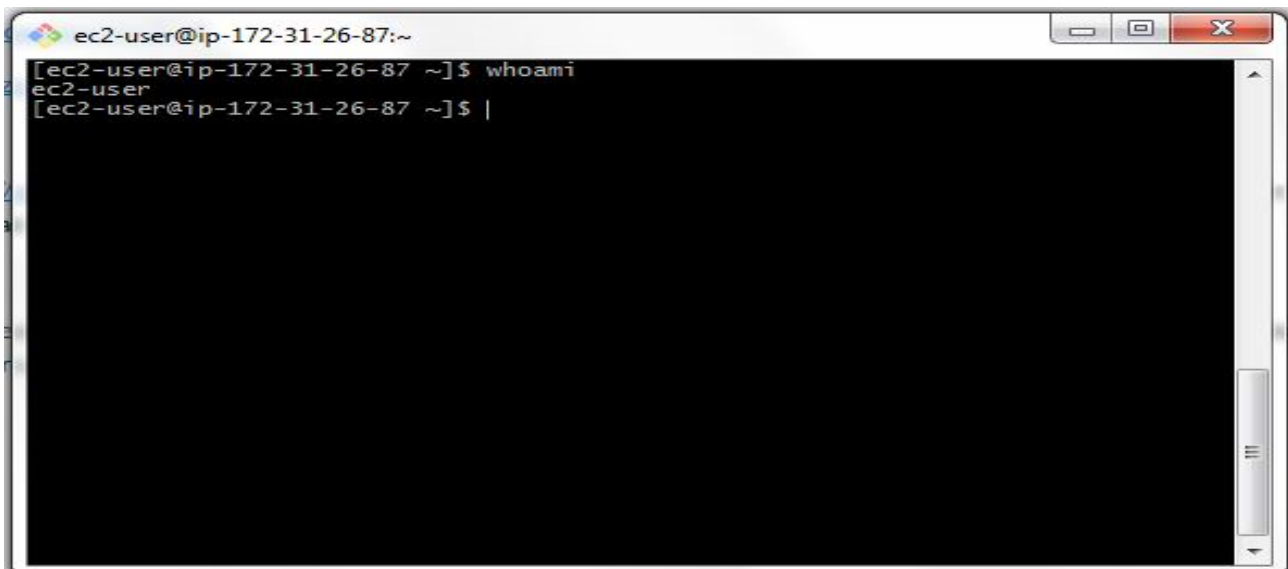
- **uname –a**

```
ec2-user@ip-172-31-26-87:~

[ec2-user@ip-172-31-26-87 ~]$ uname -a
Linux ip-172-31-26-87 4.4.41-36.55.amzn1.x86_64 #1 SMP Wed Jan 18 01:03:26 UTC
86_64 x86_64 GNU/Linux
[ec2-user@ip-172-31-26-87 ~]$
```

- Print information about the current system

- Print certain system information. If no OPTION is specified, **uname** assumes the -s option.
    - -a, --all   print all information, in the following order, except omit -p and -i if unknown:
    - -s, --kernel-name   print the kernel name
    - -n, --nodename   print the network node hostname
    - -r, --kernel-release   print the kernel release
    - -v, --kernel-version   print the kernel version
    - -m, --machine   print the machine hardware name
    - -p, --processor   print the processor type or "unknown"
    - -i, --hardware-platform   print the hardware platform or "unknown"
    - -o, --operating-system   print the operating system

- **whoami**
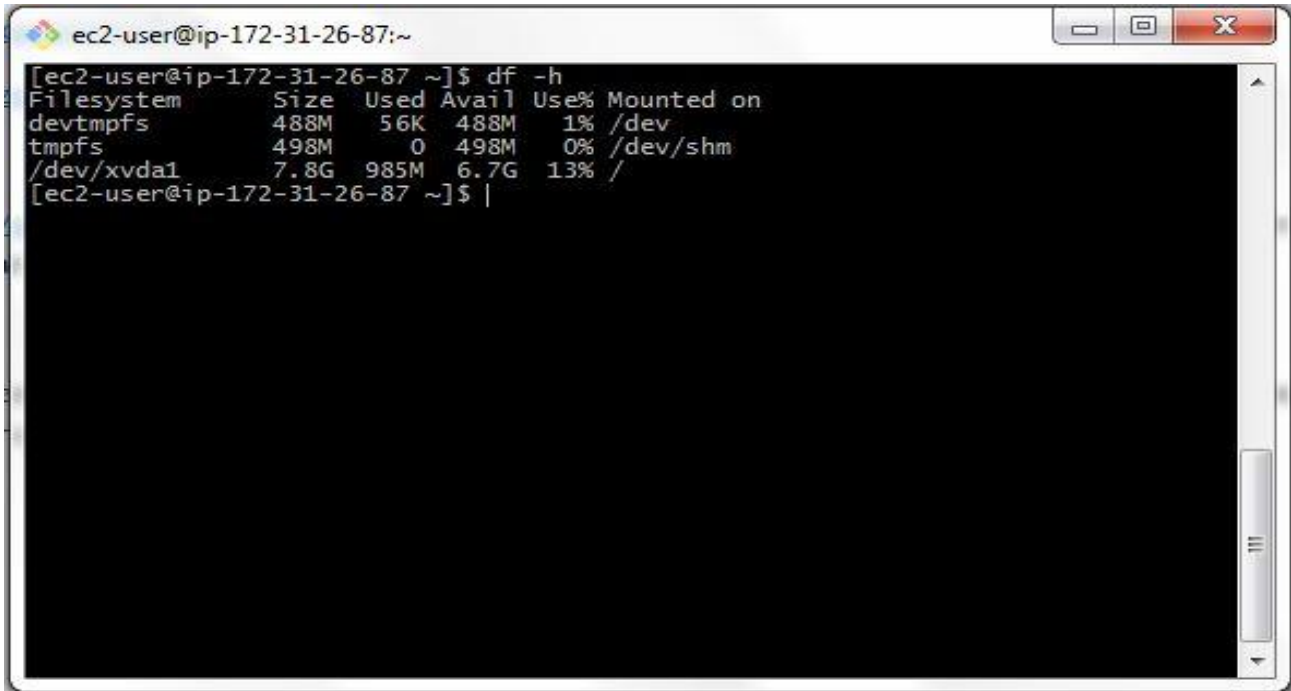
```
ec2-user@ip-172-31-26-87:~

[ec2-user@ip-172-31-26-87 ~]$ whoami
ec2-user
[ec2-user@ip-172-31-26-87 ~]$
```

whoami prints the effective user ID.
This command prints the username associated with the current effective user ID.
Running **whoami** is the same as running the id command with the options -un.

- **df -h**

df displays the amount of disk space available on the file system containing each file name argument. If no file name is given, the space available on all currently mounted file systems is shown. Disk space is shown in 1Kblocks by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

If an argument is the absolute file name of a disk device node containing a mounted file system, **df** shows the space available on that file system rather than on the file system containing the device node. **df** cannot show the space available on un-mounted file systems, because on most kinds of systems doing so requires very system-specific knowledge of file system structures.

Options
-h, --human-readable---- print sizes in human readable format (e.g., 1K 234M 2G)

-a, --all	include dummy file systems.
-B, --block-size=SIZE	scale sizes by SIZE before printing them. E.g., '-BM' prints sizes in units of 1,048,576 bytes. See "SIZE Format" below for more information.
--total	display a grand total.
-h, --human-readable	print sizes in human readable format (e.g., 1K 234M 2G)
-H, --si	same as -h, but use powers of 1000 instead of 1024.
-i, --inodes	list inode information instead of block usage.
-k	like --block-size=1K.
-l, --local	limit listing to local file systems

- **ifconfig –a**

- **ifconfig** is to configure a network interface
- **ifconfig** is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.
- If no arguments are given, **ifconfig** displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only.
- **ifconfig –a:** if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface

-
- **netstat**

The netstat command is used to print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.

Netstat "network statistics" is a command-line tool that displays network connections (both incoming and outgoing), routing tables, and a number of network interface (network interface controller or software-defined network interface) and network protocol statistics.

--route, -r	Display the kernel routing tables. See the description in route for details. netstat -r and route -e produce the same output.
--groups, -g	Display multicast group membership information for IPv4 and IPv6.
--interfaces, -i	Display a table of all network interfaces.
--masquerade, -M	Display a list of masqueraded connections.
--statistics, -s	Display summary statistics for each protocol.

(Reference: http://aws.amazon.com/ec2/, http://docs.amazonwebservices.com/AWSEC2/2009-11-30/GettingStartedGuide/)