Lab Assignment 1

(Solution)

These are the following steps which I followed to create the instance and then executing the commands:

PART - 1

1. **Choose an Amazon Machine Image:**
   - An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. I selected **Amazon Linux 2 AMI (HVM), SSD Volume Type** to launch my instance.

2. **Choose an Instance type:**
   - Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. I selected the below instance type for the running the applications.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General-purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes

3. **Configure Instance:**
   - In this step I kept the default configuration and moved towards the step 4.
4. **Add Storage:**
   - In here, the default storage size is 8 GiB. I can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume.
5. **Add Tag:**
   - A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. As this step was not mandatory, I passed over to next step.

6. **Configure Security Setup:**
   - A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. I selected the existing security group.
7. **Review Instance Launch:**
   - At this point, one needs to review the instances. I review my created instance and then I launched the created instance

8. **Choosing Key-pair:**
    - At this point, I created a key-pair to connect to the instance securely. One can also choose existing pair if they have created previously. After that a ".pem" file will be downloaded.

9. **Launch Instance**:
    - After that launching of the actual instance takes place and it comes in the running state.

Now, I downloaded **putty** and **puttygen** software.

10. After installing the puttygen software, start the respective application. In the home page, select the "Conversions" tab and select import key. Import the ".pem" file saved previously and select "Save private key". It will save a ".ppk" file for you.
11. Now copy the "Public IP" from the bottom of the Amazon web services page where you can see your instance running and start the putty software and paste the IP in the Host Name in the Session tab.
12. Now go to the Connection tab and select SSH. In that select "Auth" and browse to the ".ppk" file previously created. Go to Session tab again and save the instance giving it some meaningful name and click open.

PART-2

A command prompt gets opened and I logged into it using the username of root device and executed the following commands:

1. ***uname –a****:*

    - It prints system information. By default it will print –s.
    - **-a**: print all information in the following order – kernel name, node name, kernel release, kernel version, machine, operating system.

    - **Options**:
        - -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
        - --help : display this help and exit

- --version : output version information and exit

- **Output**: Here it is showing all the information in the order explained above.

```
[ec2-user@ip-172-31-27-240 ~]$ uname -a
Linux ip-172-31-27-240.us-east-2.compute.internal 4.14.97-90.72.amzn2.x86_64 #1
SMP Tue Feb 5 20:46:19 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
```

2. ***whoami:***

- It prints effective user id.

- **Output:** It is showing the current username I have logged into with.

```
[ec2-user@ip-172-31-27-240 ~]$ whoami
ec2-user
```

3. ***df –h:***

- This manual page documents the GNU version of df. 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 1K blocks 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. This version of df cannot show the space available on unmounted file systems, because on most kinds of systems doing so requires very nonportable intimate knowledge of file system structures.

- **Options**:
  - -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
  - --direct : show statistics for a file instead of mount point
  - --total : produce a grand total
  - -h, --human-readable : print sizes in human readable format (e.g., 1K 234M 2G)
  - -H, --si : likewise, but use powers of 1000 not 1024
  - -i, --inodes : list inode information instead of block usage

- **Output**: It is showing the sizes of the respective objects.

```
[ec2-user@ip-172-31-27-240 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        476M     0  476M   0% /dev
tmpfs           493M     0  493M   0% /dev/shm
tmpfs           493M  444K  493M   1% /run
tmpfs           493M     0  493M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.3G  6.8G  16% /
tmpfs            99M     0   99M   0% /run/user/0
tmpfs            99M     0   99M   0% /run/user/1000
```

4. ***ifconfig –a:***
   - 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; if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.

   - **Options**:
     - -a display all interfaces which are currently available, even if down
     - -s display a short list (like netstat -i)
     - -v be more verbose for some error conditions

   - **Output:** It is displaying all the interfaces which are currently available.

```
[ec2-user@ip-172-31-27-240 ~]$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 172.31.27.240  netmask 255.255.240.0  broadcast 172.31.31.255
        inet6 fe80::424:2cff:fe05:498a  prefixlen 64  scopeid 0x20<link>
        ether 06:24:2c:05:49:8a  txqueuelen 1000  (Ethernet)
        RX packets 43008  bytes 61357915 (58.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2497  bytes 247391 (241.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 648 (648.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 648 (648.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5. ***netstat***:
   - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

   - **Output**: It is showing all the network connections with the additional informtions.

```
[ec2-user@ip-172-31-27-240 ~]$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0    304 ip-172-31-27-240.us:ssh 155.246.179.162:61030   ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    17185    /var/run/chrony/chronyd.sock
unix  2      [ ]         DGRAM                    14158    /run/systemd/shutdownd
unix  3      [ ]         DGRAM                    9053     /run/systemd/notify
unix  2      [ ]         DGRAM                    9055     /run/systemd/cgroups-agent
unix  5      [ ]         DGRAM                    9067     /run/systemd/journal/socket
unix  15     [ ]         DGRAM                    9069     /dev/log
unix  3      [ ]         STREAM     CONNECTED     22531
unix  3      [ ]         STREAM     CONNECTED     18929
unix  3      [ ]         STREAM     CONNECTED     16928
unix  3      [ ]         STREAM     CONNECTED     18902
unix  3      [ ]         STREAM     CONNECTED     17274
unix  3      [ ]         STREAM     CONNECTED     22530
unix  3      [ ]         STREAM     CONNECTED     18930
unix  3      [ ]         STREAM     CONNECTED     16929    /run/systemd/journal/stdout
unix  2      [ ]         DGRAM                    14300
unix  2      [ ]         DGRAM                    18980
unix  3      [ ]         STREAM     CONNECTED     18903
unix  3      [ ]         STREAM     CONNECTED     17273
unix  3      [ ]         STREAM     CONNECTED     18927
unix  3      [ ]         STREAM     CONNECTED     16932
unix  3      [ ]         STREAM     CONNECTED     18900
unix  3      [ ]         STREAM     CONNECTED     15009    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     16987    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     18926
unix  3      [ ]         STREAM     CONNECTED     16952
unix  3      [ ]         STREAM     CONNECTED     18899
unix  3      [ ]         STREAM     CONNECTED     14575    /run/systemd/journal/stdout
unix  2      [ ]         DGRAM                    22118
unix  3      [ ]         STREAM     CONNECTED     18923
unix  3      [ ]         STREAM     CONNECTED     18953
unix  3      [ ]         STREAM     CONNECTED     18896
unix  3      [ ]         STREAM     CONNECTED     14574
unix  3      [ ]         STREAM     CONNECTED     18924
unix  2      [ ]         DGRAM                    16945
unix  3      [ ]         STREAM     CONNECTED     18954
unix  3      [ ]         STREAM     CONNECTED     18897
unix  2      [ ]         DGRAM                    17836
unix  3      [ ]         STREAM     CONNECTED     16782
```

Conclusion: I fairly studied how to launch the AWS instance and after launching it I understood the commands which I executed using the created instance.