

# 239-[LX]-Lab - Managing Processes

## Managing Processes

### Note

All labs rely on previous courseware and lab information.

## Duration

This lab will require approximately **45 minutes** to complete.

## Objectives

In this lab, you will:

- Create a new log file for process listings
- Use the top command
- Establish a repetitive task that runs your previous auditing commands once a day

## AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

## Accessing the AWS Management Console

1. At the top of these instructions, choose **Start Lab** to launch your lab.

A **Start Lab** panel opens, and it displays the lab status.

**Tip:** If you need more time to complete the lab, choose the Start Lab button again to restart the timer for the environment.

2. Wait until you see the message *Lab status: ready*, then close the **Start Lab** panel by choosing the **X**.

3. At the top of these instructions, choose **AWS**.

This opens the AWS Management Console in a new browser tab. The system will automatically log you in.

**Tip:** If a new browser tab does not open, a banner or icon is usually at the top of your browser with a message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon and then choose **Allow pop ups**.

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time so that you can follow the lab steps more easily.

## Task 1: Use SSH to connect to an Amazon Linux EC2 instance

In this task, you will connect to a Amazon Linux EC2 instance. You will use an SSH utility to perform all of these operations. The following instructions vary slightly depending on whether you are using Windows or Mac/Linux.

### Windows Users: Using SSH to Connect

● These instructions are specifically for Windows users. If you are using macOS or Linux, [skip to the next section](#).

5. Select the `Details` drop-down menu above these instructions you are currently reading, and then select `Show`. A Credentials window will be presented.
6. Select the **Download PPK** button and save the **labsuser.ppk** file.  
*Typically your browser will save it to the Downloads directory.*
7. Make a note of the **PublicIP** address.
8. Then exit the Details panel by selecting the **X**.
9. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer, [download it here](#).
10. Open **putty.exe**
11. Configure your PuTTY session by following the directions in the following link: [Connect to your Linux instance using PuTTY](#)
12. Windows Users: [Select here to skip ahead to the next task](#).

## Task 2: Exercise - Create List of Processes

In this exercise, you will create a log file from the `ps` command. This log file should be added to the SharedFolders section:

Create a log file named `processes.csv` from `ps -aux` and omit any processes that contain root user or contain "[\"or\"]" in the COMMAND section.

*Note:*

There is a space following the command followed by a period to represent the current location.

21. To validate that you are in the `/home/ec2-user/companyA` folder, enter `pwd` and press Enter.  
If you are not in this folder, enter `cd companyA` and press Enter.
22. View all processes running on the machine and filter out the word root by typing `sudo ps -aux | grep -v root | sudo tee SharedFolders/processes.csv` and pressing ENTER.
23. Validate your work by typing `cat SharedFolders/processes.csv` and pressing ENTER.

```
[ec2-user@ip-10-0-10-21 companyA]$ sudo ps -aux | grep -v root | sudo tee SharedFolders/processes.csv
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
libstor+ 1693  0.0  0.1  12620  1864 ?        Ss   03:26   0:00 /usr/bin/lsmd -d
dbus      1699  0.3  0.4  60352  4120 ?        Ss   03:26   0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopid
file --systemd-activation
rpc        1702  0.0  0.3  69348  3300 ?        Ss   03:26   0:00 /sbin/rpcbind -w
chrony     1713  0.0  0.3  122388 3712 ?        S    03:26   0:00 /usr/sbin/chronyd
rngd       1718  0.0  0.4  94044  4512 ?        Ss   03:26   0:00 /sbin/rngd -f --fill-watermark=0 --exclude=jitter
postfix    2128  0.0  0.6  90392  6804 ?        S    03:26   0:00 pickup -l -t unix -u
postfix    2129  0.0  0.6  90468  6744 ?        S    03:26   0:00 qmgr -l -t unix -u
ec2-user   6411  0.0  0.3  152644 3304 ?        S    03:27   0:00 sshd: ec2-user@pts/0
ec2-user   6412  0.0  0.4  124860 4032 pts/0    Ss   03:27   0:00 -bash
[ec2-user@ip-10-0-10-21 companyA]$ cat SharedFolders/processes.csv
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
libstor+ 1693  0.0  0.1  12620  1864 ?        Ss   03:26   0:00 /usr/bin/lsmd -d
dbus      1699  0.3  0.4  60352  4120 ?        Ss   03:26   0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopid
file --systemd-activation
rpc        1702  0.0  0.3  69348  3300 ?        Ss   03:26   0:00 /sbin/rpcbind -w
chrony     1713  0.0  0.3  122388 3712 ?        S    03:26   0:00 /usr/sbin/chronyd
rngd       1718  0.0  0.4  94044  4512 ?        Ss   03:26   0:00 /sbin/rngd -f --fill-watermark=0 --exclude=jitter
postfix    2128  0.0  0.6  90392  6804 ?        S    03:26   0:00 pickup -l -t unix -u
postfix    2129  0.0  0.6  90468  6744 ?        S    03:26   0:00 qmgr -l -t unix -u
ec2-user   6411  0.0  0.3  152644 3304 ?        S    03:27   0:00 sshd: ec2-user@pts/0
ec2-user   6412  0.0  0.4  124860 4032 pts/0    Ss   03:27   0:00 -bash
```

Figure: The command `sudo ps -aux | grep -v root | sudo tee SharedFolders/processes.csv` shows all the current processes running on your machine. This is also validated by using the command `cat SharedFolders/processes.csv`.

## Task 3: Exercise - List the processes using the top command

In this exercise, you will use the top command:

- Run the **top** command to display processes and threads that are active in the system.
- Observe the outputs of the top command.

24. In the main terminal run the command top and press ENTER:

```
top
```

The top command is used to display the system performance and lists the processes and threads active in the system. The output of the top command should look similar to the picture below:

```
top - 03:31:43 up 5 min, 1 user, load average: 0.07, 0.12, 0.07
Tasks: 93 total, 1 running, 48 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.2 st
KiB Mem : 979184 total, 299116 free, 98108 used, 581960 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 736388 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	125728	5616	3968	S	0.0	0.6	0:01.87	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H
5	root	20	0	0	0	0	I	0.0	0.0	0:00.06	kworker/u4:0
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0.0	0.0	0:00.03	ksoftirqd/0
8	root	20	0	0	0	0	I	0.0	0.0	0:00.06	rcu_sched
9	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/1
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.10	migration/1
16	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/1
17	root	20	0	0	0	0	I	0.0	0.0	0:00.06	kworker/1:0
18	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
21	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
22	root	20	0	0	0	0	I	0.0	0.0	0:00.13	kworker/u4:1
30	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/1:1
34	root	20	0	0	0	0	I	0.0	0.0	0:00.02	kworker/0:1
117	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
122	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper

Figure: The output of the top command gives the system performance and gives you the following information: Total number of tasks, how many are running, how many are sleeping, how many are stopped, zombie state. It gives the percentage of CPU used, the KiB memory used, and KiB swap.

25. While observing the output of top, the second line below the command top, we can see the Tasks (outlined in red). Tasks in top either have a running, sleep, stopped or zombie state. How many running tasks do you see?

```

top - 03:31:43 up 5 min,  1 user,  load average: 0.07, 0.12, 0.07
Tasks:  93 total,   1 running,  48 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.2 st
KiB Mem :  979184 total,  299116 free,   98108 used,   581960 buff/cache
KiB Swap:   0 total,    0 free,    0 used.  736388 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root        20   0  125728    5616   3968 S   0.0   0.6   0:01.87 systemd
    2 root        20   0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root        20   0      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0
    4 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H
    5 root        20   0      0      0      0 I   0.0   0.0   0:00.06 kworker/u4:0
    6 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    7 root        20   0      0      0      0 S   0.0   0.0   0:00.03 ksoftirqd/0
    8 root        20   0      0      0      0 I   0.0   0.0   0:00.06 rcu_sched
    9 root        20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_bh
   10 root        rt    0      0      0      0 S   0.0   0.0   0:00.00 migration/0
   11 root        rt    0      0      0      0 S   0.0   0.0   0:00.00 watchdog/0
   12 root        20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   13 root        20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   14 root        rt    0      0      0      0 S   0.0   0.0   0:00.00 watchdog/1
   15 root        rt    0      0      0      0 S   0.0   0.0   0:00.10 migration/1
   16 root        20   0      0      0      0 S   0.0   0.0   0:00.02 ksoftirqd/1
   17 root        20   0      0      0      0 I   0.0   0.0   0:00.06 kworker/1:0
   18 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H
   20 root        20   0      0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
   21 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 netns
   22 root        20   0      0      0      0 I   0.0   0.0   0:00.13 kworker/u4:1
   30 root        20   0      0      0      0 I   0.0   0.0   0:00.00 kworker/1:1
   34 root        20   0      0      0      0 I   0.0   0.0   0:00.02 kworker/0:1
  117 root        20   0      0      0      0 S   0.0   0.0   0:00.00 khungtaskd
  122 root        20   0      0      0      0 S   0.0   0.0   0:00.00 oom_reaper

```

Figure: The outline in red is the Tasks output from the top command. The command prompt shows 93 total tasks, 1 running, 48 sleeping, 0 stopped, and 0 zombie tasks.

26. To quit top, hit **q** and press ENTER.

27. You can also run top with the following options to find the usage and version information:

```
top -hv
```

## Task 4: Exercise - Create a Cron Job

In this exercise, you will create a cron job that will create an audit file with ##### to cover all csv files:

*Note:*

You may have to use `sudo` to complete this exercise if you are not root.

Remember that **cron** is a command that runs a task on a regular basis at a specified time. This command maintains the list of tasks to run in a crontab file, which you create in this task. You create a job that creates the audit file with ##### in order to cover all .csv files. When you enter the **crontab -e** command, you are taken to an editor where you then enter a list of steps of what the cron daemon will run. The crontab file includes six fields: minutes, hour, day of month (DOM), month (MON), day of Week (DOW), and command (CMD). These fields can also be denoted with asterisks. Once this command runs, you can verify your work.

28. To validate that you are in the `/home/ec2-user/companyA` folder, enter `pwd` and press Enter.

29. To create a cron job that creates the audit file with ##### to cover all .csv files, enter `sudo crontab -e` and press Enter to enter the default text editor.

30. Press `i` to enter insert mode, and press Enter.

31. For the first line, enter `SHELL=/bin/bash` and press the Space bar.

32. For the second line, enter `PATH=/usr/bin:/bin:/usr/local/bin` and press Enter.

33. For the third line, enter `MAILTO=root` and press Enter.

34. For the last line, enter `0 * * * * ls -la $(find .) | sed -e 's/..csv/#####.csv/g' > /home/ec2-user/companyA/SharedFolders/filteredAudit.csv`

Your terminal should look like the following image:

![The terminal window shows the establishment of a cron job. The cron job is used to create an audit file.](images/cron.jpg)

\*Figure: In the terminal, it shows how the cron job with the SHELL, PATH, MAILTO, and a script that was referenced earlier in the lab.\*


35. To save and close the file, press ESC. Then enter `:wq` and press Enter.

36. To validate your work, enter `sudo crontab -l` and press Enter. Inspect the crontab file to ensure that it matches the text exactly, as the following output shows:

![The terminal window show output from the command sudo crontab -l ](images/installed-cron.jpg)

\*Figure: A validated cron job is shown by entering the command sudo crontab -l. The output of the command will be from the file that was entered from earlier in the lab.\*

## Lab Complete

 Congratulations! You have completed the lab.

37. Select **End Lab** at the top of this page and then select **Yes** to confirm that you want to end the lab.

A panel will appear, indicating that "DELETE has been initiated... You may close this message box now."

38. Select the **X** in the top right corner to close the panel.