실습 - 프로세스 관리

소요 시간

본 실습을 완료하는 데는 약 **45 분**이 소요된다.

목표

본 실습에서는 다음을 수행한다.

- 프로세스 리스팅을 위한 새 로그 파일 생성
- top 명령 사용
- 하루에 한 번 이전 감사 명령을 실행하는 반복 태스크 구축

태스크 1: SSH 를 사용하여 Linux VM 에 연결

이 태스크에서는 이전 Lab 에서 설치한 Ubuntu VM 에 연결한다. SSH 유틸리티를 사용하여 이 모든 작업을 수행한다. 다음 지침은 Windows 를 사용하는지 Mac/Linux 를 사용하는지 여부에 따라 다소 차이가 있을 수 있다.

이번 실습은 어떤 OS 에서도 SSH 연결로 원격으로 커넥션이 가능한 Tabby 툴을 사용하기로 한다.

다음 내용은 해당 OS 사용자가 참고용으로 확인한다.

Windows 사용자: SSH 를 사용하여 연결

1. Windows 사용자들은 보통 PuTTY 를 사용한다.
2. PuTTY 를 사용하지 않을 경우 Tabby 를 사용한다.

macOS 및 Linux 사용자

이 지침은 Mac/Linux 사용자에게만 적용된다.

3. 터미널을 오픈한 후, 다음과 같은 순서로 연결한다.

   $ ssh -p ubuntu_portnumber ubuntu_user@ubuntu_ipaddress

태스크 2: 연습 - 프로세스 목록 생성

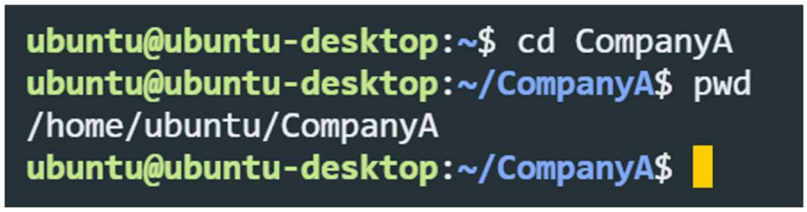
이 연습에서는 ps 명령에서 로그 파일을 생성한다. 이 로그 파일은 SharedFolders 폴더에 추가해야 한다.

ps -aux 에서 processes.csv 라는 이름의 로그 파일을 생성하고 COMMAND 섹션에서 루트 사용자를 포함하거나 "[" 또는 "]"를 포함하는 프로세스를 생략한다.

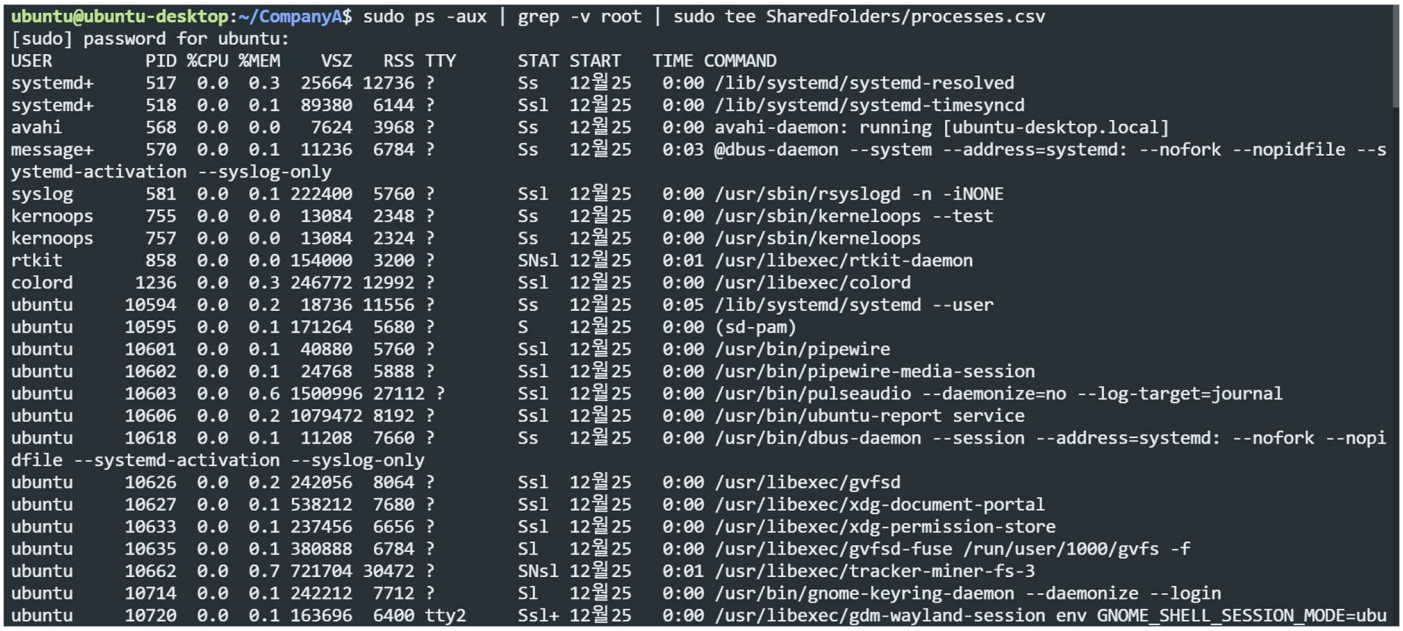
주: 현재 위치를 나타내기 위해 명령 끝의 마침표 뒤에 공백이 있다.

4. 현재 위치가 **/home/ubuntu/CompanyA** 폴더라는 것을 확인하려면 `pwd` 를 입력하고 Enter 키를 누른다.

현재 위치가 이 폴더가 아닌 경우 `cd CompanyA` 를 입력하고 Enter 키를 누른다.



5. `sudo ps -aux | grep -v root | sudo tee SharedFolders/processes.csv` 를 입력하고 ENTER 키를 눌러 기기에서 실행되는 모든 프로세스를 표시하고 단어 root 를 필터링한다.
6. `cat SharedFolders/processes.csv` 를 입력하고 ENTER 키를 눌러 작업을 검증한다.



sudo ps -aux | grep -v root | sudo tee SharedFolders/processes.csv 명령의 출력.

태스크 3: 연습 - top 명령을 사용하여 프로세스 나열

이 연습에서는 top 명령을 사용한다.

- **top** 명령을 실행하여 시스템에서 활성 상태인 프로세스와 스레드 표시.
- top 명령의 출력 관찰.

7. 기본 터미널에서 top 명령을 실행하고 ENTER 키를 누른다.

```
top
```

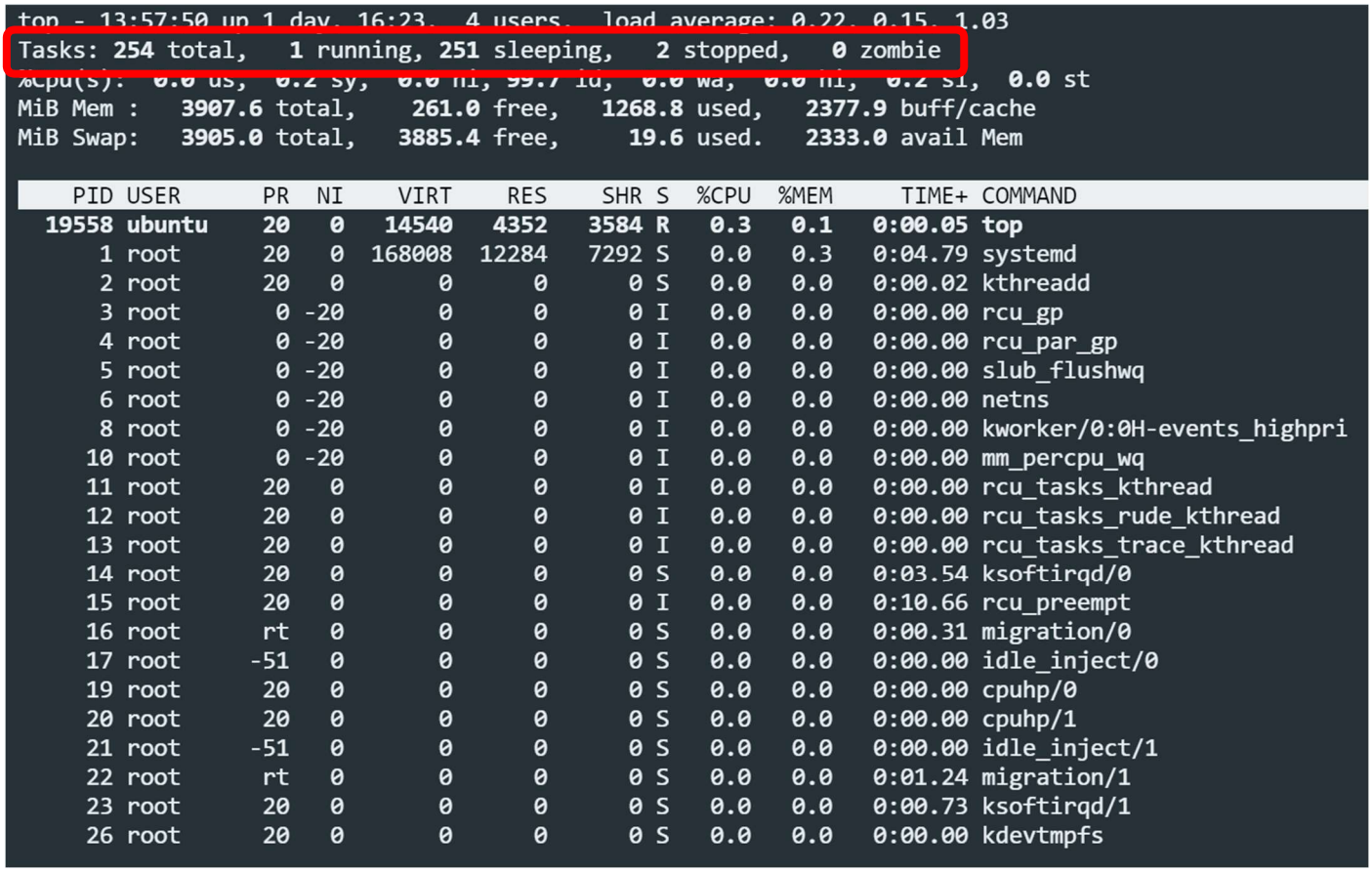
top 명령은 시스템 성능을 표시하고 시스템에서 활성 상태인 프로세스와 스레드를 나열한다. top 명령의 출력은 아래 그림과 유사해야 한다.

```
top - 13:57:50 up 1 day, 16:23,  4 users,  load average: 0.22, 0.15, 1.03
Tasks: 254 total,   1 running, 251 sleeping,   2 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.2 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :   3907.6 total,    261.0 free,   1268.8 used,   2377.9 buff/cache
MiB Swap:   3905.0 total,   3885.4 free,     19.6 used.   2333.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  19558 ubuntu    20   0   14540   4352   3584 R   0.3   0.1   0:00.05 top
      1 root      20   0  168008  12284   7292 S   0.0   0.3   0:04.79 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.02 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
     12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
     14 root      20   0       0      0      0 S   0.0   0.0   0:03.54 ksoftirqd/0
     15 root      20   0       0      0      0 I   0.0   0.0   0:10.66 rcu_preempt
     16 root      rt   0       0      0      0 S   0.0   0.0   0:00.31 migration/0
     17 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     21 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     22 root      rt   0       0      0      0 S   0.0   0.0   0:01.24 migration/1
     23 root      20   0       0      0      0 S   0.0   0.0   0:00.73 ksoftirqd/1
     26 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
```

콘솔에 top 명령의 결과가 표시된다.

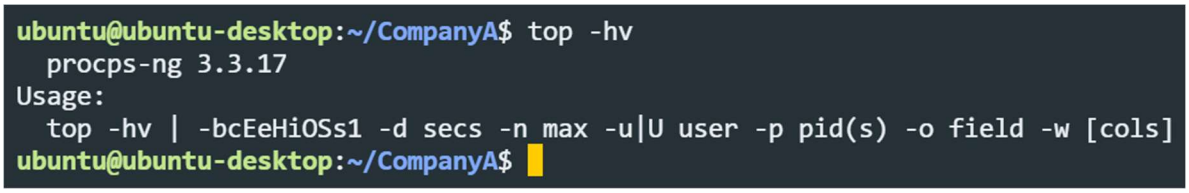
8. top 명령 아래의 두 번째 줄인 top 의 출력을 관찰하는 동안 과제(빨간색으로 표시)를 볼 수 있다. top 의 과제는 실행 중, 휴면, 중지 또는 좀비 상태이다. 실행 중인 태스크가 몇 개 있는가?



과제는 실행 중, 휴면, 중지 또는 좀비 상태일 수 있다.

9. top 을 종료하려면 **q** 를 누르고 ENTER 키를 누른다.
10. 또한 다음 옵션으로 top 을 실행하여 사용량 및 버전 정보를 찾을 수도 있다.

```
top -hv
```



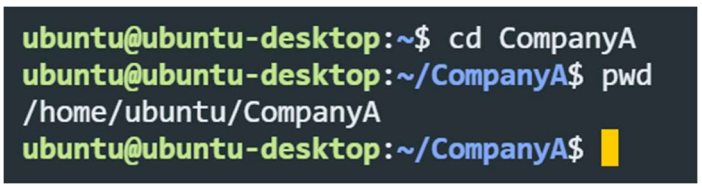
태스크 4: 연습 - Cron 작업 생성

이 연습에서는 모든 csv 파일을 포함하도록 #####이 포함된 감사 파일을 만드는 cron 작업을 생성한다.

주: 루트 사용자가 아닌 경우 sudo 를 사용하여 이 연습을 완료해야 할 수 있다.

cron 은 지정된 시간에 태스크를 정기적으로 실행하는 명령이다. 이 명령은 이 태스크에서 생성한 crontab 파일에서 실행할 태스크 목록을 유지 관리한다. 모든 .csv 파일을 포함하도록 #####이 포함된 감사 파일을 만드는 작업을 생성한다. **crontab -e** 명령을 입력하면 cron 대몬(daemon)이 실행되는 단계 목록을 입력하는 편집기로 이동한다. crontab 파일에는 분, 시간, 일(DOM), 월(MON), 요일(DOW) 및 명령(CMD)의 6 개 필드가 있다. 이러한 필드는 별표로 표시할 수도 있다. 이 명령이 실행되면 작업을 확인할 수 있다.

11. 현재 위치가 **/home/ubuntu/CompanyA** 폴더라는 것을 확인하려면 `pwd` 를 입력하고 Enter 키를 누른다.



12. 모든 .csv 파일을 포함하도록 #####이 포함된 감사 파일을 만드는 cron 작업을 생성하려면 `sudo crontab -e` 를 입력하고 Enter 키를 눌러 기본 텍스트 편집기로 들어간다.

```
ubuntu@ubuntu-desktop:~/CompanyA$ sudo crontab -e
no crontab for root - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]: 2
```

13. i 를 눌러 삽입 모드로 들어가고 Enter 키를 누른다.
14. 첫 번째 줄에 `SHELL=/bin/bash` 를 입력하고 Enter 키를 누른다.
15. 두 번째 줄에 `PATH=/usr/bin:/bin:/usr/local/bin` 을 입력하고 Enter 키를 누른다.
16. 세 번째 줄에 `MAILTO=root` 를 입력하고 Enter 키를 누른다.
17. 마지막 줄에 `0 * * * * ls -la $(find .) | sed -e 's/..csv/#####.csv/g' > /home/ubuntu/companyA/SharedFolders/filteredAudit.csv` 를 입력한다.

터미널은 다음 이미지와 같이 표시된다.

```
SHELL=/bin/bash
PATH=/usr/bin:/bin:/usr/local/bin
MAILTO=root
0 * * * * ls -la $(find .) | sed -e 's/..csv/#####.csv/g' > /home/ubuntu/CompanyA/SharedFolders/filteredAudit.csv

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
-- INSERT --
```

cron 작업의 예:

18. 파일을 저장하고 닫으려면 ESC 키를 누른다. 그런 다음 `:wq` 를 입력하고 Enter 키를 누른다.
19. 작업을 검증하려면 `sudo crontab -l` 를 입력하고 Enter 키를 누른다. crontab 파일을 검사하여 다음 출력과 같이 텍스트와 정확히 일치하는지 확인한다.

```
ubuntu@ubuntu-desktop:~/CompanyA$ sudo crontab -l
SHELL=/bin/bash
PATH=/usr/bin:/bin:/usr/local/bin
MAILTO=root
0 * * * * ls -la $(find .) | sed -e 's/..csv/#####.csv/g' > /home/ubuntu/CompanyA/SharedFolders/filteredAudit.csv

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
```

cron 작업 검증의 예.