

실습 - 서비스 관리

목표

본 실습에서는 다음을 수행한다.

- 서비스 **apache2**의 상태를 검토하여 실행되고 있는지 확인 및 로컬 호스트 IP 주소에 대한 http 연결을 설정할 수 있는지 확인
- Linux **top** 명령 사용

소요 시간

30 분

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: apache2 서비스의 상태 확인

apache2 는 호스트에 설치된 Apache http 서버용 서비스다. 이는 자주 사용하는 웹사이트(예: naver.com)를 실행하는 것과 같은 경량 웹 서버이다. 이 연습에서는 **apache2** 서비스의 상태를 확인하고 **systemctl** 명령을 사용하여 시작하고 서비스가 작동 중인지 확인한다.

유용한 힌트

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

4. 아래와 같이 **systemctl** 명령을 사용하고 ENTER 키를 눌러 **apache2** 서비스의 상태를 확인한다.

```
sudo systemctl status apache2
```

예상 출력:

```
ubuntu@ubuntu-desktop:~$ sudo systemctl status apache2
Unit apache2.service could not be found.
ubuntu@ubuntu-desktop:~$
```

여기에는 **apache2** 서비스가 설치되지 않았음을 의미한다. 따라서 다음 단계는 해당 서비스를 설치하는 것부터 시작해야 한다.

5. 아래와 같이 **apt** 명령을 사용하여 **apache2** 서비스를 설치한다.

```
sudo apt install apache2
```

```
ubuntu@ubuntu-desktop:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 5 not upgraded.
Need to get 1,919 kB of archives.
After this operation, 7,718 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

6. 이제 **systemctl** 명령을 사용하고 ENTER 키를 눌러 apache2 서비스의 상태를 확인한다.

```
sudo systemctl status apache2
```

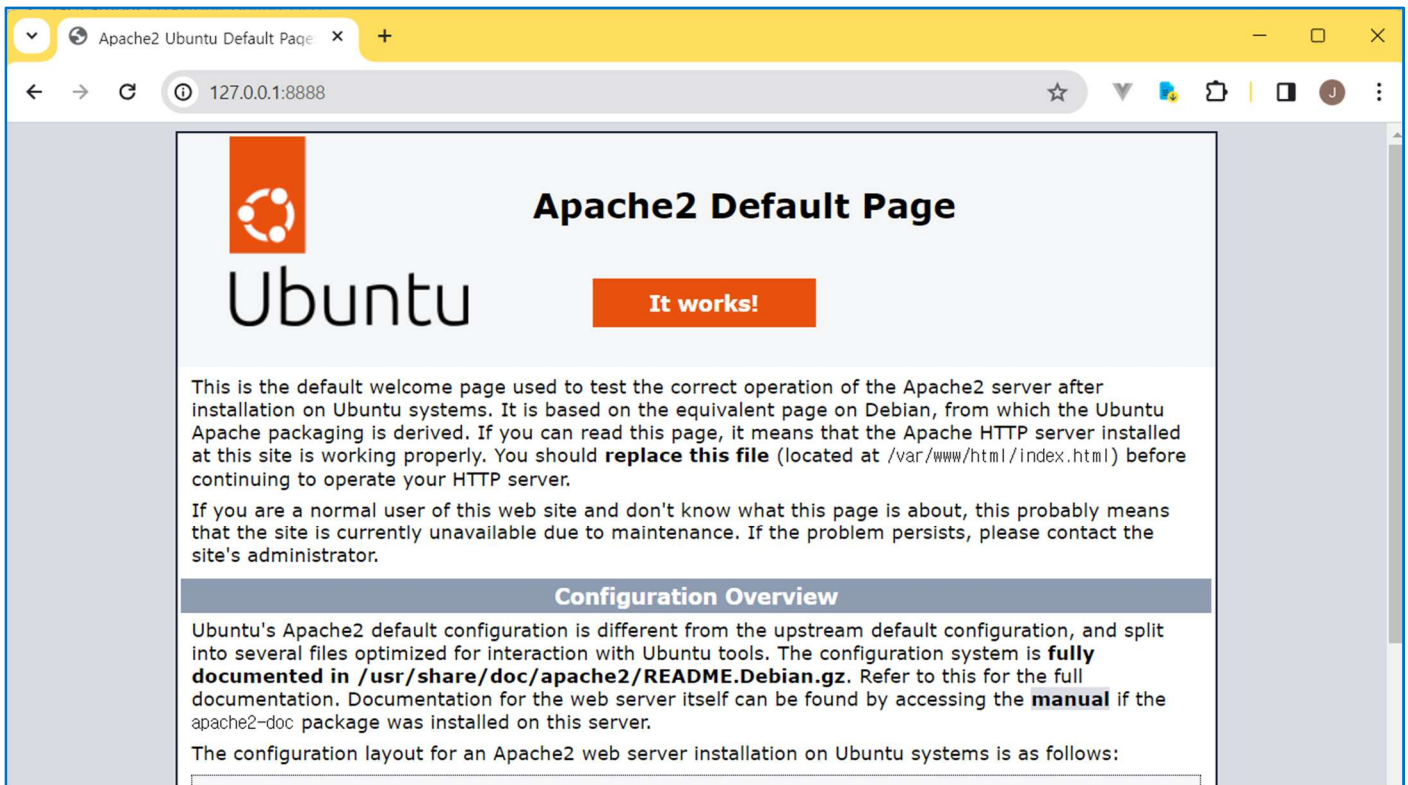
예상 출력

```
ubuntu@ubuntu-desktop:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-12-28 13:26:10 KST; 33s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 3514 (apache2)
    Tasks: 55 (limit: 4599)
   Memory: 4.9M
      CPU: 77ms
   CGroup: /system.slice/apache2.service
           └─3514 /usr/sbin/apache2 -k start
             └─3515 /usr/sbin/apache2 -k start
               └─3516 /usr/sbin/apache2 -k start

12월 28 13:26:10 ubuntu-desktop systemd[1]: Starting The Apache HTTP Server...
12월 28 13:26:10 ubuntu-desktop apachectl[3513]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please edit the 'ServerName' line in the configuration file to reflect the actual server name
12월 28 13:26:10 ubuntu-desktop systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

7. **apache2** 가 실행 중이므로 이제 제대로 작동하는지 확인해보자. 브라우저에서 새 탭을 열고 **http://** 을 입력하고, IP 를 입력한다.

예상 출력



8. 이제 아래 명령을 입력하고 Enter 키를 눌러 서비스를 중지할 수 있다.

```
sudo systemctl stop apache2
```

```
ubuntu@ubuntu-desktop:~$ sudo systemctl start apache2
ubuntu@ubuntu-desktop:~$ sudo systemctl stop apache2
ubuntu@ubuntu-desktop:~$ sudo systemctl restart apache2
ubuntu@ubuntu-desktop:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-12-28 13:28:13 KST; 7s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 3877 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 3881 (apache2)
    Tasks: 55 (limit: 4599)
   Memory: 4.8M
      CPU: 32ms
   CGroup: /system.slice/apache2.service
           └─3881 /usr/sbin/apache2 -k start
             └─3882 /usr/sbin/apache2 -k start
               └─3883 /usr/sbin/apache2 -k start

12월 28 13:28:13 ubuntu-desktop systemd[1]: Starting The Apache HTTP Server...
12월 28 13:28:13 ubuntu-desktop apachectl[3880]: AH00558: apache2: Could not reliably determine the server's fu
12월 28 13:28:13 ubuntu-desktop systemd[1]: Started The Apache HTTP Server.
lines 1-17/17 (END)
```

태스크 3: Linux VM 모니터링

이 연습에서는 Linux VM 을 모니터링하기 위해 Linux 명령을 사용한다.

유용한 힌트

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

- 아래 명령을 입력하고 Enter 키를 눌러 실행 중인 프로세스 목록을 표시한다.

top

예상 출력

```
top - 14:21:21 up 1:15, 3 users, load average: 0.07, 0.10, 0.15
Tasks: 202 total, 2 running, 198 sleeping, 2 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.3 sy, 0.0 ni, 97.8 id, 1.5 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3907.6 total, 395.3 free, 922.5 used, 2589.9 buff/cache
MiB Swap: 3905.0 total, 3905.0 free, 0.0 used, 2685.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6986	netdata	20	0	270040	47972	8704	S	0.7	1.2	0:19.03	netdata
7177	netdata	20	0	54348	4224	2048	S	0.7	0.1	0:15.28	apps.plugin
631	message+	20	0	10984	6400	3968	S	0.3	0.2	0:06.13	dbus-daemon
7172	netdata	20	0	277528	58980	19840	S	0.3	1.5	0:07.17	python.d.plugin
7656	ubuntu	20	0	14404	4352	3584	R	0.3	0.1	0:00.01	top
1	root	20	0	168064	13148	8284	S	0.0	0.3	0:05.25	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	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	0:01.08	ksoftirqd/0
15	root	20	0	0	0	0	I	0.0	0.0	0:02.22	rcu_preempt
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.05	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

상단에는 현재 실행 중인 프로세스와 CPU 사용량 및 메모리 사용량과 같은 리소스 사용량이 표시된다. **q** 를 눌러 종료하고 셸로 돌아간다. 다음 단계에서는 CPU 에서 워크로드를 시뮬레이션하는 스크립트를 실행한다.

- 다음 명령을 입력하고 ENTER 키를 눌러 Linux VM 에서 많은 워크로드를 시뮬레이션하는 **stress.sh** 스크립트를 설치 및 실행한다.


```
$ sudo apt install stress
```

```
$ ./stress.sh & top
```

```
ubuntu@ubuntu-desktop:~$ sudo apt install stress
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  stress
0 upgraded, 1 newly installed, 0 to remove and 5 not upgraded.
Need to get 18.4 kB of archives.
After this operation, 52.2 kB of additional disk space will be used.
Get:1 https://mirror.kakao.com/ubuntu jammy/universe amd64 stress amd64 1.0.5-1 [18.4 kB]
Fetched 18.4 kB in 0s (51.9 kB/s)
Selecting previously unselected package stress.
(Reading database ... 242891 files and directories currently installed.)
Preparing to unpack .../stress_1.0.5-1_amd64.deb ...
Unpacking stress (1.0.5-1) ...
Setting up stress (1.0.5-1) ...
Processing triggers for man-db (2.10.2-1) ...
ubuntu@ubuntu-desktop:~$
```

11.1 단계에서와 같이 **top** 명령을 입력하고 Enter 키를 눌러 실행 중인 프로세스 목록을 표시한다.

예상 출력

```
top - 14:29:35 up 1:23, 3 users, load average: 2.29, 0.63, 0.32
Tasks: 220 total, 11 running, 207 sleeping, 2 stopped, 0 zombie
%Cpu(s): 51.2 us, 23.5 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 25.3 si, 0.0 st
MiB Mem : 3907.6 total, 186.5 free, 1129.2 used, 2592.2 buff/cache
MiB Swap: 3905.0 total, 3905.0 free, 0.0 used, 2478.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7844	ubuntu	20	0	3704	256	256	R	43.1	0.0	0:02.82	stress
7836	ubuntu	20	0	3704	256	256	R	23.4	0.0	0:01.66	stress
7835	ubuntu	20	0	134780	14872	128	R	20.4	0.4	0:01.67	stress
7846	ubuntu	20	0	3704	256	256	R	17.8	0.0	0:01.94	stress
7841	ubuntu	20	0	3704	256	256	R	14.8	0.0	0:01.74	stress
7839	ubuntu	20	0	3704	256	256	R	14.5	0.0	0:01.55	stress
7845	ubuntu	20	0	3704	256	256	R	14.5	0.0	0:01.50	stress
7833	ubuntu	20	0	3704	256	256	R	12.5	0.0	0:01.30	stress
7838	ubuntu	20	0	134780	114660	128	R	12.5	2.9	0:01.44	stress
7843	ubuntu	20	0	3704	256	256	R	12.2	0.0	0:01.41	stress
60	root	0	-20	0	0	0	I	3.6	0.0	0:02.73	kworker/0:1H-kblockd
265	root	0	-20	0	0	0	I	3.6	0.0	0:02.97	kworker/1:2H-kblockd
7842	ubuntu	20	0	3704	128	128	D	1.3	0.0	0:00.09	stress
7177	netdata	20	0	54348	4224	2048	S	1.0	0.1	0:18.75	apps.plugin
7834	ubuntu	20	0	3704	128	128	D	1.0	0.0	0:00.09	stress
7837	ubuntu	20	0	3704	128	128	D	1.0	0.0	0:00.08	stress
6986	netdata	20	0	273364	51352	8704	S	0.7	1.3	0:23.51	netdata
7840	ubuntu	20	0	3704	128	128	D	0.7	0.0	0:00.08	stress
527	systemd+	20	0	14828	6784	6016	S	0.3	0.2	0:05.42	systemd-oomd

12. 추가적으로 **htop** 명령어로 모니터링 해보자.

13. **http** 가 첫 번째 열에 표시되는 **stress.sh** 프로세스의 **PID** 를 복사한다.

14. **q**를 눌러 **htop** 애플리케이션을 종료한다. 이제 아래 명령을 입력하고 Enter 키를 눌러 스트레스 프로세스를 종료할 수 있다(7906를 터미널에서 읽은 실제 PID로 바꿈).

```
ubuntu@ubuntu-desktop:~$ sudo kill 7906  
ubuntu@ubuntu-desktop:~$ stress: FAIL: [7899] (416) <-- worker 7906 got signal 15  
stress: WARN: [7899] (418) now reaping child worker processes  
stress: FAIL: [7899] (422) kill error: No such process  
^C  
ubuntu@ubuntu-desktop:~$ sudo kill 7899  
Terminated  
[2]- Exit 143                               ./stress.sh  
ubuntu@ubuntu-desktop:~$
```