

128- [PF] - 실습 - Python으로 시스템 관리

Python으로 시스템 관리 소개

실습 개요

Linux 를 사용하여 터미널 또는 Bash 명령줄에서 많은 관리 태스크를 수행할 수 있습니다. Python 은 명령줄에서 명령을 실행하는 데 사용할 수 있는 여러 모듈을 제공합니다. 이 실습에서는 `os.system()`과 `subprocess.run()`을 사용하여 Python 에서 Bash 명령을 실행합니다.

본 실습에서는 다음을 수행합니다.

- `os.system()`을 사용하여 Bash 명령 실행
- `subprocess.run()`을 사용하여 Bash 명령 실행

예상 완료 시간

30 분

AWS Cloud9 IDE 액세스

1. 이 지침의 상단으로 이동한 다음 **Start Lab** 을 선택하여 실습 환경을 시작합니다.

Start Lab 패널이 열리고 실습 상태가 표시됩니다.

2. *Lab status: ready* 라는 메시지가 표시되면 **X** 를 선택하여 **Start Lab** 패널을 닫습니다.
3. 지침의 맨 위에서 **AWS** 를 선택합니다.

새 브라우저 탭에서 AWS 관리 콘솔이 열립니다. 시스템에 자동으로 로그인됩니다.

참고: 새 브라우저 탭이 열리지 않는 경우 일반적으로 브라우저에서 팝업 창을 열 수 없음을 나타내는 배너 또는 아이콘이 브라우저 상단에 표시됩니다. 배너 또는 아이콘을 선택하고 **Allow pop ups** 를 선택합니다.

4. AWS 관리 콘솔에서 **Services > Cloud9** 을 선택합니다. **Your environments** 패널에서 **reStart-python-cloud9** 카드를 찾아 **Open IDE** 를 선택합니다.

AWS Cloud9 환경이 열립니다.

참고: *.c9/project.settings have been changed on disk* 라는 메시지가 담긴 팝업 창이 표시되면 **Discard** 를 선택하여 무시합니다. 마찬가지로, *Show third-party content* 라는 대화 창이 나타나면 **No** 를 선택하여 거절합니다.

Python 연습 파일 생성

5. 메뉴 모음에서 **File > New From Template > Python File** 을 선택합니다.

이 작업은 제목이 없는 파일을 생성합니다.

6. 템플릿 파일에서 샘플 코드를 삭제합니다.
7. **File > Save As...**를 선택하고, 연습 파일에 적절한 이름(예: *sys-admin.py*)을 입력한 다음 **/home/ec2-user/environment** 디렉터리에 저장합니다.

터미널 세션에 액세스

8. AWS Cloud9 IDE 에서 + 아이콘을 선택하고 **New Terminal** 을 선택합니다.

터미널 세션이 열립니다.

9. 현재 작동 중인 디렉터리를 표시하려면 **pwd** 를 입력합니다. 이 명령은 **/home/ec2-user/environment** 를 가리킵니다.
10. 이 디렉터리에서 이전 섹션에서 생성한 파일을 찾을 수 있어야 합니다.

연습 1: os.system 사용

Python 에는 Python 에서 Bash 명령을 실행할 수 있는 여러 가지 모듈이 있습니다. 이 연습에서는 **os.system()**을 사용하여 디렉터리 콘텐츠를 보여주는 Bash 명령 **ls** 를 실행합니다.

11. IDE 의 탐색 창에서 이전 **Python 연습 파일 생성** 섹션에서 생성한 파일을 선택합니다.

12. os 모듈 가져오기:

```
import os
```

13. 모듈에는 다른 개발자가 작성한 함수가 포함되어 있다는 사실을 기억하십시오. `os.system()` 함수는 문자열 인수를 취합니다. Bash 명령을 실행하려면 다음 명령을 입력합니다.

```
os.system("ls")
```

14. Cloud 9 IDE 에 파일을 저장하고 **Run** 을 선택하여 파일을 실행합니다.
15. 현재 디렉터리의 콘텐츠가 출력에 표시됩니다. 출력이 다음 예시와 비슷한지 확인합니다. 디렉터리의 콘텐츠는 다를 수 있습니다.

```
sys-admin.py README.md
```

연습 2: subprocess.run 사용

`os.system()`은 문자열 인수를 취하기 때문에 사용이 간편하지만, 보다 강력한 `subprocess.run()` 함수를 사용하는 것이 권장됩니다. `subprocess` 모듈을 사용하여 새 프로세스를 생성하고, 입력/출력/오류 파이프에 연결하고, 오류 코드를 확보할 수 있습니다. `subprocess.run()` 함수는 새로운 여러 인수를 취할 수 있지만, 이러한 추가 인수는 선택 사항입니다.

`subprocess.run()`에 대한 인수의 전체 목록은 다음 목록과 같습니다.

```
subprocess.run(args, *, stdin=None, input=None, stdout=None, stderr=None,
capture_output=False, shell=False, cwd=None, timeout=None, check=False, encoding=None,
errors=None, text=None, env=None, universal_newlines=None)
```

16. 이 실습에서는 코드를 단순하게 유지합니다.
17. 이 실습을 위해 생성한 파일에 `subprocess` 모듈을 가져옵니다.

```
import subprocess
```

18. `ls` Bash 명령을 실행하려면 다음 명령을 입력합니다.

```
subprocess.run(["ls"])
```

19. Cloud 9 IDE 에 파일을 저장하고 **Run** 을 선택하여 파일을 실행합니다.

20. 다음 예와 유사한 디렉터리의 파일이 출력에 나열되는지 확인합니다. (디렉터리의 콘텐츠는 다를 수 있습니다.)

```
sys-admin.py  sys-admin_2.py  README.md
```

출력은 연습 1 에서의 `os.system()`의 출력과 동일하지만, `subprocess` 모듈이 아닌 `os` 모듈을 사용합니다.

연습 3: 인수를 2 개 사용하여 `subprocess.run` 사용

Python 에서 대괄호는 목록 데이터 유형이며, `run()`이 인수 목록을 취할 수 있음을 나타냅니다. Python 스크립트를 계속해서 추가합니다.

21. 이 연습의 실습 파일에서 추가 인수를 포함하도록 스크립트의 마지막 줄을 수정합니다.

```
subprocess.run(["ls", "-l"])
```

22. `"-l"`은 `ls` 명령에 긴 리스팅 형식을 사용하라고 알리는 인수입니다.

23. Cloud 9 IDE 에 파일을 저장하고 **Run** 을 선택하여 파일을 다시 실행합니다.

24. 출력이 다음 예시와 비슷한지 확인합니다.

```
total 12
-rw-r--r-- 1 ec2-user ec2-user  55 Apr 16 20:20 sys-admin.py
-rw-r--r-- 1 ec2-user ec2-user 343 Apr 16 19:07 sys-admin_2.py
-rw-r--r-- 1 ec2-user ec2-user 569 Apr  6 02:17 README.md
```

연습 4: 인수를 3 개 사용하여 `subprocess.run` 사용

이제 인수를 3 개 사용하여 `subprocess.run()`을 호출합니다. 세 번째 인수는 디렉터리 이름입니다.

25. Python 파일로 돌아가 스크립트의 마지막 줄을 수정합니다.

```
subprocess.run(["ls","-l","README.md"])
```

26. Cloud 9 IDE 에 파일을 저장하고 **Run** 을 선택하여 파일을 실행합니다.

27. 예상한 출력이 다음 예시와 비슷한지 확인합니다.

```
-rw-r--r-- 1 ec2-user ec2-user 569 Apr  6 02:17 README.md
```

연습 5: 시스템 정보 검색

`subprocess.run()` 함수는 모든 Bash 명령을 실행할 수 있기 때문에 강력합니다. 이 연습에서는 `uname` 명령을 호출하여 시스템 정보를 가져옵니다.

28. Python 파일로 돌아가 다음 코드를 입력합니다.

```
command="uname"
commandArgument="-a"
print(f'Gathering system information with command: {command} {commandArgument}')
subprocess.run([command,commandArgument])
```

29. Cloud 9 IDE 에 파일을 저장하고 **Run** 을 선택하여 파일을 실행합니다.

30. 예상한 출력이 다음 예시와 비슷한지 확인합니다.

```
Gathering system information with command: uname -a
Linux ip-172-31-29-181 4.4.0-139-generic #165-Ubuntu SMP Wed Oct 24 10:58:50
UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
```

연습 6: 디스크 공간에 대한 정보 검색

`subprocess.run()`으로 모든 명령을 실행할 수 있음을 강조하기 위해, **df** 명령을 실행하여 디스크 정보를 가져옵니다.

31. Python 파일로 돌아가 다음 코드를 입력합니다.

```
command="ps"
commandArgument="-x"
print(f'Gathering active process information with command: {command}
{commandArgument}')
subprocess.run([command,commandArgument])
```

32. Cloud 9 IDE 에 파일을 저장하고 **Run** 을 선택하여 파일을 실행합니다.

33. 예상한 출력이 다음 예시와 비슷한지 확인합니다.

```
Gathering active process information with command: ps -x
  PID TTY          STAT       TIME COMMAND
18976 pts/459    S+          0:00 python3.6 lab_15_2.py
18977 pts/459    R+          0:00 ps -x
21139 pts/459    S           0:00 /bin/bash -c export OLD_HOME=/home/ccc_4dfa91ec5a_
21164 pts/459    S           0:00 bash --rcfile /home/ccc_4dfa91ec5a_45122/.termrc -
```

축하합니다! Python 에서 Bash 명령을 호출했습니다.

실습 종료

축하합니다! 실습을 마치셨습니다.

34. 이 페이지의 상단에서 **End Lab** 을 선택한 다음 Yes 를 선택하여 실습 종료를 확인합니다.

*DELETE has been initiated... You may close this message box now.*라는 내용의 패널이 표시됩니다.

35. *Ended AWS Lab Successfully* 라는 메시지가 잠시 표시되어 실습이 종료되었음을 나타냅니다.

추가 리소스

AWS Training and Certification 에 대한 자세한 내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다. 제안이나 수정 사항을 공유하려면 [AWS Training and Certification Contact Form](#)에서 세부 정보를 제공해 주십시오.

© 2022 Amazon Web Services, Inc. 및 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 대여 또는 판매는 금지됩니다.