

04. 이미지 저장 장소 준비하기(S3)

[Amazon S3\(Simple Storage Services\)](#)

[EBS](#)

[S3](#)

[프리티어](#)

[스토리지](#)

[S3](#)

[S3의 작업 범위와 비용](#)

[가용성\(Availability\)](#)

[S3와 VPC의 관계](#)

[역할 적용](#)

[S3 시스템](#)

[S3 버킷 생성하기](#)

[생성 내용](#)

[버킷 조작 순서](#)

[AWS CLI로 S3 관리](#)

[시스템 백업 스크립트](#)

[버킷 만들기](#)

[예제](#)

[버킷 및 파일 나열](#)

[예제](#)

[버킷에 업로드\(백업\)](#)

[셸스크립트로 작성](#)

[버킷 삭제](#)

[예제](#)

[객체 삭제](#)

[객체 이동](#)

[예제](#)

[예제](#)

[CRON으로 정기 백업 스케줄링하기 \(/etc/crontab\)](#)

[예제](#)

Amazon S3(Simple Storage Services)

확장성이 뛰어나며, 무한대로 저장 가능하고, 사용한 만큼만 지불하는 인터넷 기반 스토리지 서비스.

버킷(Bucket)이라는 리전(Region) 내에서 유일한 영역을 생성하고 데이터를 키-값 형식의 객체(Object)로 저장한다.

S3 서비스는 스토리지 기술을 근간으로 하며, 파일 단위의 접근만 지원하기 때문에 EBS(Elastic Block Storage) 서비스를 대체할 수 없다.

EBS

- 블록 스토리지 (따라서 포맷해야 함) 이는 원하는 파일 시스템 유형을 선택할 수 있다.
- 블록 스토리지이므로 여러 블록 스토리지에 Raid 1 (또는 0 또는 10)을 사용할 수 있다.
- 속도가 빠르고 비교적 저렴하다.
- Amazon의 새로운 발표로 SSD 당 스토리지 당 최대 16TB 데이터를 저장할 수 있습니다.
- 백업 이유로 EBS 스냅 샷 (아직 실행 중)을 스냅 샷 할 수 있다.
- 그러나 특정 지역에만 존재한다. 다른 리전으로 마이그레이션 할 수는 있지만 리전간에 액세스 할 수는 없다. (EC2를 통해 공유하는 경우에만 해당되며 파일 서버가 있음을 의미 함).
- 연결하려면 EC2 인스턴스가 필요합니다
- 새로운 기능 (2017 년 2 월 15 일) : 이제 볼륨을 사용하는 동안 볼륨 크기를 늘리거나 성능을 조정하거나 볼륨 유형을 변경할 수 있다. 변경 사항이 적용되는 동안 응용 프로그램을 계속 사용할 수 있다.

S3

- 파일 시스템이 아닌 객체 저장소
- 파일과 “폴더”를 저장할 수 있지만 기존 파일 시스템에서와 같이 잠금, 권한 등을 가질 수는 없다.
- 즉, 기본적으로 S3를 마운트하여 웹 서버로 사용할 수는 없다.
- 그러나 웹 사이트의 이미지와 비디오를 저장하는 데 적합하다.
- 단기 보관에 적합하다 (예 : 몇 주). 장기 보관에도 적합하지만 Glacier가 더 비용 효율적입니다.
- 로그 저장에 적합하다.
- 모든 지역의 데이터에 액세스 할 수 있다. (추가 비용이 적용될 수 있음)

- 고 가용성, 중복성. 기본적으로 데이터 손실이 불가능하다 (99.999999999 % 내구성, 99.9 가동 시간 SLA)
- EBS보다 훨씬 저렴하다.
- 콘텐츠를 인터넷에 직접 제공 할 수 있으며 EC2 인스턴스 없이 S3에서 직접 작동하는 전체 (정적) 웹 사이트를 보유 할 수도 있다.

프리티어

- 5GB Amazon S3 표준 스토리지
- Get 요청 20,000건, Put요청 2,000건

스토리지

스토리지는 데이터를 오랫동안 저장하는 것을 목적으로 제공되는 데이터 저장 장소다. 주위에서 가장 쉽게 찾아볼 수 있는 스토리지로는 PC에 설치된 SSD 또는 하드 디스크 등을 들 수 있다.

스토리지와 메모리의 비교

항목	스토리지	메모리
속도	느림	빠름
용량당 비용	낮음	높음
데이터 저장 기간	영구적	전원을 끄면 사라짐
사용방법	우선 메모리에 로딩해서 사용	직접 사용
용도	텍스트나 이미지 등을 장기간 저장하는 장소	프로그램 작동에 필요한 정보를 일시적으로 보존하는 장소(변수 등)

S3

EC2 인스턴스는 SSD나 하드 디스크에 해당하는 스토리지를 제공한다. EBS(Awazon Elastic Block Store)라는 이름으로 제공되는 서비스다. 다만 EBS를 스토리지로 이용하면 다음과 같은 문제가 발생한다.

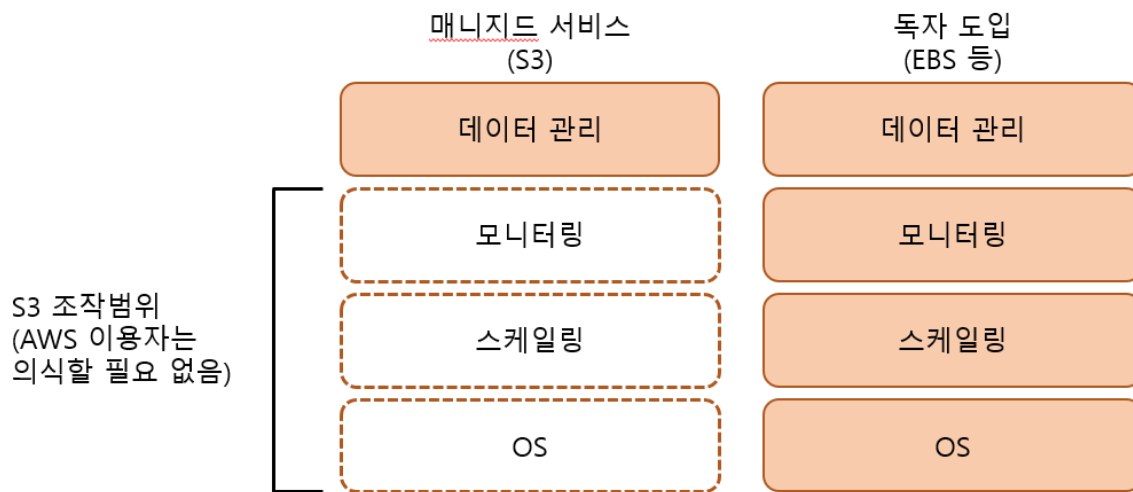
- EC2 인스턴스의 OS 자체도 관리해야 한다. (보안 대응 등)
- 예측하지 못한 장애 발생 시 대응의 필요한 경우에 대한 준비를 해야 한다.

- EC2 인스턴스를 사용하지 못할 가능성이 있다(SLA에 의하면 1년에 약 5분 정도 서비스가 정지될 가능성이 있다.)

이런 문제를 해결하고자 AWS에서 S3(Amazon Simple Storage Service) 라는 서비스를 제공한다.

S3의 작업 범위와 비용

S3는 스토리지 관리를 수행하는 매니지드 서비스다.



가용성(Availability)

가용성이란, 해당 시스템이나 서비스가 가동 및 실행 되는 시간의 비율을 말한다.

가용성이 높다는 것은 서비스의 가동률이 높다는 것을 의미하며, 이러한 가용성은 보통 '9'로 측정한다.

예를 들어, 가용성 수준이 '세 개의 9(Three Nines)'인 서비스는 가동되어 실행되는 **99.9%**의 시간만큼 지원 할 수 있다는 것을 말하며, **24 x 7 x 356(하루 24시간/일주일 7일/일년 365일)**을 기준으로 했을 때 1년 동안 서비스 중지가 **8.76시간**이라는 것을 의미한다.

다음은 인프라의 가용성 비용에 따른 서비스 가동 중지 시간을 정리한 내용이다.

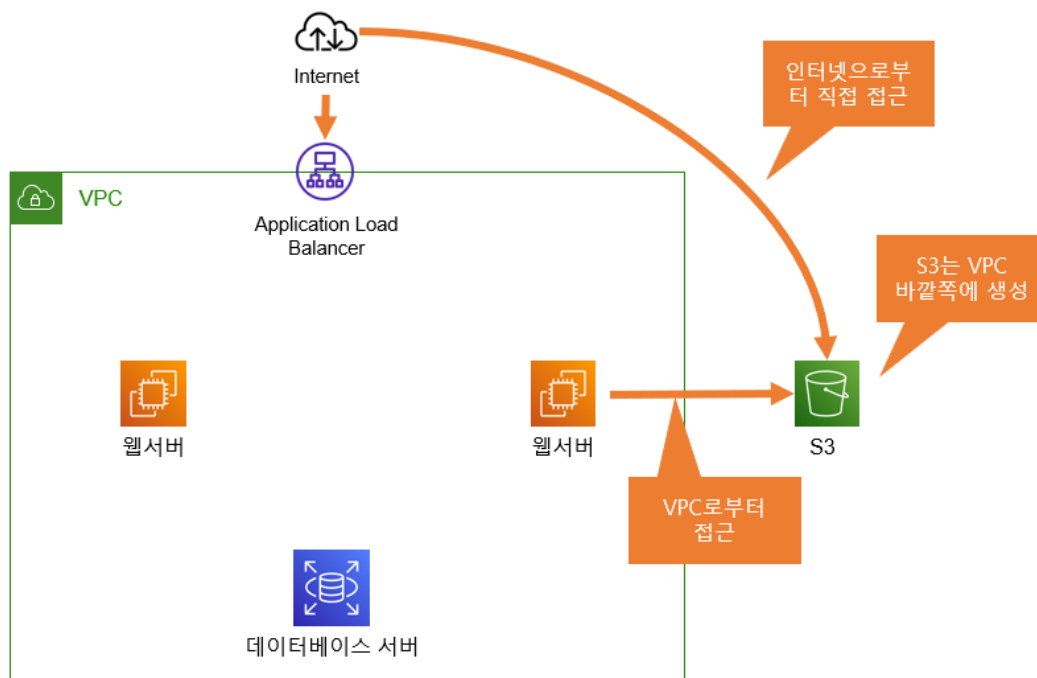
가용성 비율	가동 중지 시간(24시간 기준)
90%	876시간(36.5일)
95%	438시간(18.25일)

가용성 비율	가동 중지 시간(24시간 기준)
99%	87.6시간(3.65일)
99.9%	8.76시간
99.99%	52.56분
99.999%	5.256분

S3와 VPC의 관계

- 인터넷으로부터 직접 접근
- VPC로부터 접근

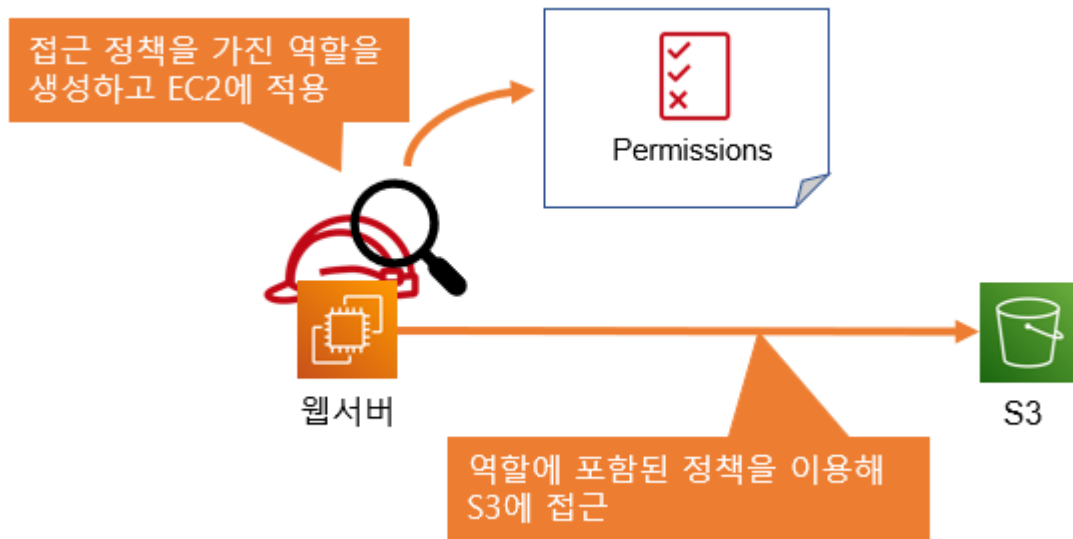
VPC와 S3의 관계




VPC안의 리소스로부터 접근할 때는 S3 버킷에 대한 접근 권한이 필요하다. 이 권한은 일반적으로 IAM의 역할이라는 개념을 이용해서 적용한다.

먼저 대상이 되는 S3 버킷에 접근하기 위한 정책을 가진 역할을 생성한다. 그리고 그 역할을 EC2에 적용한다.

역할 적용



 기술적으로는 S3에 접근할 수 있는 정책을 가진 사용자를 만든 뒤 해당 사용자의 권한과 액세스 키를 조합해 S3 버킷에 접근할 수도 있다. 단, 역할을 이용하는 방법을 보통 더 많이 이용한다.

S3 시스템

S3에 데이터를 저장할 때는 먼저 버킷을 생성해야 한다. 버킷을 S3에서 관리하는 데이터를 하나로 모은 단위다.

버킷 안에는 서비스에서 이용하는 데이터인 객체를 저장한다. 객체는 파일에 해당하며 텍스트, 이미지, 음성, 동영상 등 파일로 다룰 수 있는 것은 대부분 S3 객체로 취급할 수 있다. 대량의 객체가 존재할 때는 폴더를 이용해 시스템적으로 관리할 수도 있다.

S3 버킷 생성하기

생성 내용

항목	값	설명
버킷 이름	aws-intro-sample-upload * 중복되지 않는 이름으로 저장	S3 버킷 이름
리전	ap-northeast-2	S3 버킷을 생성할 리전
퍼블릭 접근	모두 블록 * '퍼블릭 접근을 모두 블록'에 체크	외부로부터의 접근에 대해 S3 버킷의 공개 정도를 설정

버킷 조작 순서

[서비스] - [S3] - [버킷]

The screenshot displays the Amazon S3 console interface. On the left, a sidebar contains navigation links: '버킷' (Buckets), 'Storage Lens', and 'S3용 AWS Marketplace'. The main area is titled 'Amazon S3 > 버킷'. It features a '계정 스냅샷' (Account Snapshot) section with a table showing storage metrics: 총 스토리지 (541.4KB), 객체 수 (33), and 평균 객체 크기 (16.4KB). Below this is the '버킷 정보' (Bucket Information) section, which includes a search bar and a '버킷 만들기' (Create Bucket) button. The bottom part of the console shows a table with columns for '이름' (Name), 'AWS 리전' (AWS Region), '액세스' (Access), and '생성 날짜' (Creation Date), currently displaying '버킷 없음' (No buckets).

[버킷 만들기] 클릭 후 '일반 구서' 카테고리에서 버킷 이름과 리전을 설정한다.

Amazon S3 > 버킷 > 버킷 만들기

버킷 만들기 정보

버킷은 S3에 저장되는 데이터의 컨테이너입니다. 자세히 알아보기 [🔗](#)

일반 구성

버킷 이름

aws00-intro-sample-upload

버킷 이름은 전역에서 고유해야 하며 공백 또는 대문자를 포함할 수 없습니다. 버킷 이름 지정 규칙 [참조](#) [🔗](#)

AWS 리전

아시아 태평양(서울) ap-northeast-2 ▼

기존 버킷에서 설정 복사 - 선택 사항
다음 구성의 버킷 설정만 복사됩니다.

버킷 선택

‘ 이 버킷의 퍼블릭 액세스 차단 설정’ 카테고리로 이동해 데이터로의 접근 권한을 설정한다.
이 설정은 데이터의 의도치 않은 열람 또는 변경을 방지하기 위해 매우 중요하다.

기본값으로 ‘모든 퍼블릭 액세스 차단’에 체크되어 있으며 데이터는 공개되지 않는다. 따라서
의도하지 않은 열람이나 변경을 방지할 수 있다.

객체 소유권 정보

다른 AWS 계정에서 이 버킷에 작성한 객체의 소유권 및 액세스 제어 목록(ACL)의 사용을 제어합니다. 객체 소유권은 객체에 대한 액세스를 지정할 수 있는 사용자를 결정합니다.

☒ ACL 비활성화됨(권장)

이 버킷의 모든 객체는 이 계정이 소유합니다. 이 버킷과 그 객체에 대한 액세스는 정책을 통해서만 지정됩니다.

☐ ACL 활성화됨

이 버킷의 객체는 다른 AWS 계정에서 소유할 수 있습니다. 이 버킷 및 객체에 대한 액세스는 ACL을 사용하여 지정할 수 있습니다.

객체 소유권

버킷 소유자 적용

이 버킷의 퍼블릭 액세스 차단 설정

퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지점 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 이 버킷 및 해당 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 모든 퍼블릭 액세스 차단을 활성화합니다. 이 설정은 이 버킷 및 해당 액세스 지점에만 적용됩니다. AWS에서는 모든 퍼블릭 액세스 차단을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 이 버킷 또는 내부 객체에 대한 어느 정도 수준의 퍼블릭 액세스가 필요한 경우 특정 스토리지 사용 사례에 맞게 아래 개별 설정을 사용자 지정할 수 있습니다. [자세히 알아보기](#)

☒ 모든 퍼블릭 액세스 차단

이 설정을 활성화하면 아래 4개의 설정을 모두 활성화한 것과 같습니다. 다음 설정 각각은 서로 독립적입니다.

☒ 새 ACL(액세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 새로 추가된 버킷 또는 객체에 적용되는 퍼블릭 액세스 권한을 차단하며, 기존 버킷 및 객체에 대한 새 퍼블릭 액세스 ACL 생성을 금지합니다. 이 설정은 ACL을 사용하여 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 권한을 변경하지 않습니다.

☒ 임의의 ACL(액세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 모든 ACL을 무시합니다.

☒ 새 퍼블릭 버킷 또는 액세스 지점 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 새 버킷 및 액세스 지점 정책을 차단합니다. 이 설정은 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 정책을 변경하지 않습니다.

☒ 임의의 퍼블릭 버킷 또는 액세스 지점 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 및 교차 계정 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 정책을 사용하는 버킷 또는 액세스 지점에 대한 퍼블릭 및 교차 계정 액세스를 무시합니다.

AWS CLI로 S3 관리

시스템 백업 스크립트

```
#!/bin/bash
```

```
# ||는 '만약 실패하면'
```

```
cd /var/backups || exit 0
```

```
for FILE in passwd group shadow gshadow; do
```

```
# for 다음의 파일이 없으면 done
```

```
test -f /etc/$FILE || continue
# 원본과 백업을 비교해서 같으면 다음, 아니면 밑으로
cmp -s $FILE.bak /etc/$FILE && continue
# -p : 원본파일의 기존 소유권과 타임스탬프를 유지한다.
cp -p /etc/$FILE $FILE.bak && chmod 600 $FILE.bak
done
```

버킷 만들기

```
$ aws s3 mb <target> [--options]
```

예제

```
# Bucket명은 전체 s3 시스템에서 고유한 이름이어야 한다.
$ aws s3 mb s3://aws00-bucket
```

버킷 및 파일 나열

```
$ aws s3 ls <target> [--options]
```

예제

```
# 모든 Amazon S3 버킷을 나열한다.
$ aws s3 ls
2018-12-11 17:08:50 aws00-bucket

# 버킷에 있는 모든 객체와 접두사를 나열한다.
$ aws s3 ls s3://aws00-bucket
2018-12-04 19:05:48          3 MyFile1.txt

# 명령에 특정 접두사를 포함하여 출력을 필터링 할 수 있다.
$ aws s3 ls s3://aws00-bucket/M
2018-12-06 18:59:32          3 MyFile1.txt
```

버킷에 업로드(백업)

```
$ aws s3 sync <원본 디렉토리> s3://aws00-bucket

# 서버의 로그들을 s3의 버킷에 동기화
$ aws s3 sync ./storage/logs s3://aws00-bucket/logs
$ aws s3 sync /home/ubuntu/dir2backup s3://aws00-bucket

# 서버의 특정파일을 s3의 버킷에 복사 (이름지정해서 복사가능)
$ aws s3 cp ./storage/logs/mylog.log s3://aws00-bucket/logs/mylog-cp.log

# 현재날짜 넣어서 복사
$ aws s3 cp ./storage/logs/mylog.log s3://aws00-bucket/logs/mylog-`date +%Y%m%d`.log

# 현재날짜 넣어서 복사 (위와 동일함)
$ aws s3 cp ./storage/logs/mylog.log s3://aws00-bucket/logs/mylog-`date +%F`.log
```

셸스크립트로 작성

```
#!/bin/bash
/usr/local/bin/aws s3 sync /home/ubuntu/dir2backup s3://aws00-bucket
```

aws의 위치는 where aws 로 검색

버킷 삭제

```
$ aws s3 rb <target> [--options]
```

예제

```
# 버킷을 삭제한다.
$ aws s3 rb s3://aws00-bucket

# 버킷의 모든 객체와 접두사를 삭제한 다음 버킷을 삭제한다.
$ aws s3 rb s3://aws00-bucket --force
```

객체 삭제

버킷이나 로컬 디렉토리의 파일을 삭제한다.

```
$ aws s3 rm <target> [--options]

$ aws s3 rm s3://aws00-bucket/hello.txt

# 하위 디렉토리를 포함한 모든 객체 삭제
$ aws s3 rm s3://aws00-bucket --recursive
```

객체 이동

```
$ aws s3 mv <source> <target> [--options]
```

예제

```
# s3://std22-bucket/에서 모든 객체를 s3://my-bucket/으로 이동합니다.
$ aws s3 mv s3://aws00-bucket s3://my-bucket/
$ aws s3 mv filename.txt s3://aws00-bucket
$ aws s3 mv s3://std22-bucket/hello.txt ./
```

예제

```
# filename.txt.에서 s3:bucket-name/example 파일을 삭제한다.
$ aws s3 rm s3://aws00-bucket/hello.txt --recursive

# s3://bucket-name/example 옵션을 사용하여 --recursive 에서 모든 객체를 삭제한다.
$ aws s3 rm s3://aws00-bucket --recursive
```

CRON으로 정기 백업 스케줄링하기 (/etc/crontab)

```
# 매주 월요일 아침 5시 21분에 업그레이드 한다.
# 분 시 일 월 요일 소유자 명령, 1(월요일) ~ 7(또는 0 일요일)
21 5 * * 1 root apt update && apt upgrade
```

예제

```
# 매분 test.sh 실행
* * * * * /home/script/test.sh

# 매주 금요일 오전 5시 45분에 test.sh 를 실행
45 5 * * 5 /home/script/test.sh >> /home/test_sh_cron.log
# >> 명령어를 활용해 실행 Output을 log파일 형태로 append하게 만들 수 있다.

# 매일 매시간 0분, 20분, 40분에 test.sh 를 실행
0,20,40 * * * * * /home/script/test.sh

# 매일 1시 0분부터 30분까지 매분 test.sh 를 실행
0-30 1 * * * /home/script/test.sh

# 매 10분마다 test.sh 를 실행
*/10 * * * * /home/script/test.sh

# 5일에서 6일까지 2시,3시,4시에 매 10분마다 test.sh 를 실행
*/10 2,3,4 5-6 * * /home/script/test.sh

# 1월부터 12월까지 2개월마다 1일날 오전 4시 10분에 test.sh 를 실행
10 4 1 1-12/2 * /home/script/test.sh
```

- *: 각 필드 자리에 별기호가 오면 해당 필드의 모든 값을 의미한다.
- -: 그 사이의 모든 값 (정규식)
- , : 지정한 모든 값을 의미한다. 불규칙적인 값 지정시 주로 사용한다.
- / : 연결된 설정 값 범위에서 **특정 주기로 나눌 때** 사용

파이썬을 이용한 S3 사용

```
$ import boto3

# S3 Client 생성
$ s3 = boto3.client('s3')

# S3에있는 현재 버킷리스트의 정보를 가져온다.
$ response = s3.list_buckets()
```

```
$ print(response)

# response에 담겨있는 Buckets의 이름만 가져와 buckets 변수에 배열로 저장.
$ buckets = [bucket['Name'] for bucket in response['Buckets']]

# S3 버킷 리스트를 출력.
$ print("Bucket List: %s" % buckets)
```