

S H O U T . O U R . P A S S I O N . T O G E T H E R

21st U N I S O P T

06.
Multer
AWS S3
Scheduler

21st UNI SOPT

6th Seminar

1. AWS S3

- AWS S3 소개
- S3 버킷 만들기
- IAM 설정
- gitignore 파일 설정

2. Multer

- multer 모듈 소개
- 단일 파일 in Local
- AWS-SDK & multer-s3
- 다중 파일 in S3

3. Scheduler

- Scheduler의 필요성
- CRON 표현식

6 차 세 미 나

01. AWS S3

AWS S3 소개

S3 - Simple Storage Service

- AWS에서 제공하는 파일 스토리지 서비스
- 클라우드 스토리지와 비슷한 역할
- 언제 어디서든 파일 전송, 저장 및 읽기 가능
- 프리티어에서 20000건의 Get, 2000건의 Put, 15GB 전송의 무료사용 가능

01. AWS S3

S3 버킷 만들기

The screenshot displays the AWS Management Console interface. On the left sidebar, the 'Storage' category is selected. The main content area shows a grid of AWS services. The 'Storage' category is expanded, and the 'S3' service is highlighted with a red box. The console header shows the AWS logo, navigation tabs for '서비스' (Services) and '리소스 그룹' (Resource Groups), and a user profile for 'Blake Kim'.

이름 또는 기능(예: EC2, S3 또는 VM, 스토리지)으로 서비스를 찾습니다.

그룹 A-Z

- 컴퓨팅**
 - EC2
 - Elastic Container Service
 - Lightsail
 - Elastic Beanstalk
 - Lambda
 - Batch
- 스토리지**
 - S3**
 - EFS
 - Glacier
 - Storage Gateway
- 데이터베이스**
 - RDS
 - DynamoDB
 - ElastiCache
 - Amazon Redshift
- 네트워킹 및 콘텐츠 전송**
 - VPC
 - CloudFront
 - Direct Connect
 - Route 53
- 마이그레이션**
 - AWS Migration Hub
- 개발자 도구**
 - CodeStar
 - CodeCommit
 - CodeBuild
 - CodeDeploy
 - CodePipeline
 - X-Ray
- 관리 도구**
 - CloudWatch
 - CloudFormation
 - CloudTrail
 - Config
 - OpsWorks
 - Service Catalog
 - Trusted Advisor
 - Managed Services
- 보안, 자격 증명 및 규정 준수**
 - IAM
 - Inspector
 - Certificate Manager
 - Directory Service
 - WAF & Shield
 - Artifact
 - Amazon Macie
 - CloudHSM
- 분석**
 - Athena
 - EMR
 - CloudSearch
 - Elasticsearch Service
 - Kinesis
 - Data Pipeline
 - QuickSight
 - AWS Glue
- 인공 지능**
 - Lex
 - Amazon Polly
 - Rekognition
 - Machine Learning
- 사물 인터넷**
 - AWS IoT
 - AWS Greengrass
- 문의 센터**
 - Amazon Connect
- 게임 개발**
 - Amazon GameLift
- 모바일 서비스**
- 애플리케이션 서비스**
 - Step Functions
 - SWF
 - API Gateway
 - Elastic Transcoder
- 메시징**
 - Simple Queue Service
 - Simple Notification Service
 - Simple Email Service
- 기업 생산성**
 - WorkDocs
 - WorkMail
 - Amazon Chime
- 데스크톱 및 앱 스트리밍**
 - WorkSpaces
 - AppStream 2.0

AWS 콘솔에서 서비스 - S3 선택

S3 버킷 만들기

Amazon S3

새 콘솔 검색 | 간편 도움말

버킷 검색

+ 버킷 만들기 | 버킷 삭제 | 버킷 비우기

5 버킷 1 퍼블릭 1 리전 ↺

버킷 이름 ↑	액세스 ⓘ ↑	리전 ↑	생성 날짜 ↑
befreshcommunity	퍼블릭 아님 *	아시아 태평양(서울)	2017. 6. 28. 오후 9:00:39
befreshrecipes	퍼블릭 아님 *	아시아 태평양(서울)	2017. 6. 26. 오전 4:13:41
everyfoody	퍼블릭 아님 *	아시아 태평양(서울)	2017. 9. 26. 오전 4:30:06
unithonprac	퍼블릭	아시아 태평양(서울)	2017. 7. 21. 오후 5:25:57
ytkimsopt	퍼블릭 아님 *	아시아 태평양(서울)	2017. 5. 20. 오후 3:12:04

* 객체 ACL로 인해 객체는 여전히 퍼블릭 액세스될 수 있습니다. [세부 정보](#)

버킷만들기 선택

S3 버킷 만들기

버킷 만들기

1 이름 및 리전

2 속성 설정

3 권한 설정

4 검토

이름 및 리전

버킷 이름 ⓘ
uniquename

리전
아시아 태평양(서울) ▾

기존 버킷에서 설정 복사
버킷을 선택합니다(선택 사항). 5 버킷 ▾

생성

취소

다음

버킷 이름 및 리전 선택 후 다음

버킷이름에 대문자 사용불가, AWS에서 유니크한 이름이어야 함

S3 버킷 만들기

버킷 만들기

이름 및 리전

2 속성 설정

3 권한 설정

4 검토

버전 관리

동일 버킷 내에 한 객체의 여러 버전을 보관합니다.

세부 정보

☐ 비활성

서버 액세스 로깅

액세스 요청에 대한 세부 정보를 기록하는 액세스 로그를 설정합니다.

세부 정보

☐ 비활성

태그

프로젝트나 기타 기준에 대해 비용을 추적하려면 태그를 사용합니다.

세부 정보

☐ 0태그

객체 수준 로깅

CloudTrail 데이터 이벤트 기능을 사용한 객체 수준 API 활동을 기록합니다(추가 비용).

세부 정보

☐ 비활성

기본 암호화

이전

다음

속성설정 - 지금은 크게 건들것 없으므로 다음

S3 버킷 만들기

버킷 만들기

이름 및 리전

속성 설정

3 권한 설정

4 검토

사용자 관리

사용자 ID

객체

객체 권한

urbanscenery(소유자)

☒ 읽기 ☒ 쓰기

☒ 읽기 ☒ 쓰기

×

다른 AWS 계정에 대한 액세스

+ 계정 추가

계정

객체

객체 권한

퍼블릭 권한 관리

이 버킷에 퍼블릭 읽기 액세스를 부여하지 말 것(권장사항)

시스템 권한 관리

이 버킷에 Amazon S3 Log Delivery 그룹 쓰기 액세스 권한을 부여하지 말 것

이전

다음

사용자 및 외부 권한 설정

퍼블릭 권한 관리는 버킷별 혹은 파일별 관리도 가능

=> 버킷에는 가급적이면 퍼블릭권한 주지 않는걸 권장

S3 버킷 만들기

버킷 만들기

이름 및 리전

속성 설정

권한 설정

4 검토

이름 및 리전

버킷 이름 uniqueness1234 리전 아시아 태평양(서울)

속성

버전 관리

비활성

서버 액세스 로깅

비활성

태그 지정

0 태그

객체 수준 로깅

비활성

기본 암호화

None

권한

사용자

1

퍼블릭 권한

비활성

시스템 권한

비활성

이전

버킷 만들기

버킷 만들기 완료

IAM 설정

버킷 만들기

이름 및 리전

속성 설정

권한 설정

4 검토

이름 및 리전

버킷 이름 uniqueness1234 리전 아시아 태평양(서울)

속성

버전 관리

비활성

서버 액세스 로깅

비활성

태그 지정

0 태그

객체 수준 로깅

비활성

기본 암호화

None

권한

사용자

1

퍼블릭 권한

비활성

시스템 권한

비활성

이전

버킷 만들기

버킷 만들기 완료

SOPT

IAM 설정

Identity and Access Management

- AWS의 사용자 접근권한을 관리
- 한 AWS 계정 내에 여러 IAM 유저 생성 및 권한설정 가능
- Access Key와 Secret Key 는 AWS의 CLI, API를 사용할때 인증하는 수단
- 모든권한을 가진 Key들이 털리면 AWS의 모든 권한을 획득
=> 매우 위험하다
- CLI, API를 사용할 때 IAM User별로 권한을 따로 주고 관리

01. AWS S3

IAM 설정

The screenshot displays the AWS Management Console interface. On the left sidebar, the '기록' (History) menu is expanded, showing 'S3', 'RDS', 'EC2', '콘솔 홈' (Console Home), and '결제' (Billing). The main content area shows a grid of AWS services categorized into: 컴퓨팅 (Computing), 개발자 도구 (Developer Tools), 분석 (Analytics), 애플리케이션 서비스 (Application Services), 스토리지 (Storage), 관리 도구 (Management Tools), 인공지능 (Artificial Intelligence), 기업 생산성 (Enterprise Productivity), 데이터베이스 (Database), 보안, 자격 증명 및 규정 준수 (Security, Identity, and Compliance), 사물 인터넷 (Internet of Things), 데스크톱 및 앱 스트리밍 (Desktop and App Streaming), 네트워킹 및 콘텐츠 전송 (Networking and Content Delivery), and 마이그레이션 (Migration). The 'IAM' service is highlighted with a red box under the '보안, 자격 증명 및 규정 준수' category. The top navigation bar includes the AWS logo, '서비스' (Services) dropdown, '리소스 그룹' (Resource Groups) dropdown, and a user profile 'Blake Kim'.

AWS 콘솔에서 서비스 - IAM 선택

IAM 설정

Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

Add user

Delete user

Find users by username or access key

☐

User name ▼

Groups

☐

testUser

None

User - Add User 선택

IAM 설정

Add user



Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* ☒ **Programmatic access**

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ **AWS Management Console access**

Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

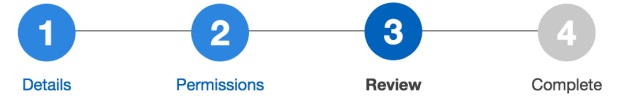
[Cancel](#)

[Next: Permissions](#)

User name 설정 및 Access Type 설정
Access type은 access key 와 secret key를 사용하게 설정

IAM 설정

Add user



Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name sopt21th
AWS access type Programmatic access - with an access key

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AmazonS3FullAccess

[Cancel](#) [Previous](#) [Create user](#)

IAM 유저 생성

IAM 설정

Add user



Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://458782526410.signin.aws.amazon.com/console>

Download .csv

User		Access key ID	Secret access key
▶	✔ sopt21th	AKIAJS2KDSQ2K7PVFKVA	***** Show

Close

Access key ID 및 Secret access key 확인

특히나 Secret access key는 이 창을 지나게 되면 다시 확인 불가하므로 다른곳에 메모!

gitignore 설정

.gitignore 파일

- Git에 커밋, 푸시를 할 때 어떤 파일을 자동으로 업로드 하지 않을지 설정
- config 파일 등 중요정보가 담긴 파일을 커밋할때마다 제외시키기 힘들
=> 한번 실수하면 다른 커밋으로 해당파일을 지워도 로그가 남음
- 포트폴리오용으로 Repository를 공개하고싶은데 AWS키가 털리네...?
- 새로 만들어서 올리자니 커밋이력이랑 개발과정 공개가 아쉬워....

.gitignore 파일을 처음부터 잘 설정하면 해결!

gitignore 설정

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: urbanscenery / Repository name: helloSOPT ✓

Great repository names are short and memorable. Need inspiration? How about **animated-barnacle**.

Description (optional): gitignore 파일 실습

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

☒ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None Add a license: None ⓘ

.gitignore

Filter ignores...

- Magento
- Maven
- Mercury
- MetaProgrammingSystem
- Nim
- Node**
- OCaml
- Objective-C

© 2017 GitHub, Inc. Te Contact GitHub API Training Shop Blog About

Github에서 새 레퍼지토리 만들 때 Add .gitignore - Node 선택

gitignore 설정

```
1 # Logs
2 logs
3 *.log
4 npm-debug.log*
5 yarn-debug.log*
6 yarn-error.log*
7
8 # Runtime data
9 pids
10 *.pid
11 *.seed
12 *.pid.lock
13
14 # Directory for instrumented libs generated by jscoverage/JSCover
15 lib-cov
16
17 # Coverage directory used by tools like istanbul
18 coverage
19
20 # nyc test coverage
21 .nyc_output
22
23 # Grunt intermediate storage (http://gruntjs.com/creating-plugins#storing-task-files)
24 .grunt
25
26 # Bower dependency directory (https://bower.io/)
27 bower_components
28
29 # node-waf configuration
30 .lock-wscript
31
32 # Compiled binary addons (http://nodejs.org/api/addons.html)
33 build/Release
34
35 # Dependency directories
36 node_modules/
37 jspm_packages/
38 config/
39
40 # Typescript v1 declaration files
41 typings/
42
43 # Optional npm cache directory
44 .npm
45
```

Node를 위한 기본적인 ignore 설정이 되어있다!

=> 적당한 곳에 /config 추가 (config폴더를 커밋시키지 않겠다)

6 차 세 미 나

02. Multer

multer 모듈 소개

Multer

- 기존 x-www-urlencoded의 방법으로는 파일 전송이 불가능하다
- multipart/form-data 의 방법으로 파일전송을 수행
- multipart/form-data 의 파일을 처리하는 모듈이 multer
- 클라이언트가 파일을 전송하면 지정한 곳에 파일을 업로드
- multipart/form-data 가 아닌 방법은 처리하지 못한다

단일파일 in Local

```
1  const express = require('express');
2  const router = express.Router();
3  const multer = require('multer');
4  const upload = multer({
5    storage: multer.diskStorage({
6      destination: './saved',
7      filename: function(req, file, cb) {
8        cb(null, file.originalname);
9      }
10   })
11 });
12
13 router.post('/', upload.single('image'), function(req, res) {
14   res.status(201).send("successful save image file!");
15 });
16
17 module.exports = router;
```

- 한개의 파일을 처리하기 위해 single 사용
- 서버 local에 저장하기 위해 diskStorage 옵션 사용

aws-sdk & multer-s3

AWS-SDK

- Node.js 에서 자바스크립트용 AWS SDK를 사용하기 위한 모듈
- S3, EC2, Dynamo DB 등의 서비스 사용가능

multer-s3

- AWS S3에 multer 를 사용해 파일을 업로드하기 위한 모듈
- 버킷과 권한, 업로드할 파일 이름 등을 설정
- 서버 로컬에 업로드하기 위해 diskStorage 옵션을 사용했다면 S3에 업로드 하기위해서는 aws-s3를 사용한 옵션을 사용한다

다중파일 in S3

```
1  const express = require('express');
2  const router = express.Router();
3  const multer = require('multer');
4  const multerS3 = require('multer-s3');
5  const aws = require('aws-sdk');
6  aws.config.loadFromPath('./config/aws_config.json');
7  const s3 = new aws.S3();
8  const upload = multer({
9    storage: multerS3({
10      s3: s3,
11      bucket: 'uniquename1234',
12      acl: 'public-read',
13      key: function(req, file, cb) {
14        cb(null, Date.now() + '.' + file.originalname.split('.').pop());
15      }
16    })
17  });
18
19  router.post('/', upload.array('image', 2), function(req, res){
20    res.status(201).send({
21      msg : "successful save image file!",
22      data : [req.files[0].location, req.files[1].location]
23    });
24  });
25
26  module.exports = router;
```

- 여러개의 파일을 업로드 하기 위해 array 사용
- 두번째 인자로 파일의 갯수를 명시
- S3에 업로드 하기 위해 multer-s3를 require한 옵션 사용
- single은 req.file.location 으로 파일링크 받아옴
- array는 req.files[i].location 으로 파일링크 받아옴

실습

1. 사진 업로드 API를 구현해주세요 (POST)
2. DB에 들어갈 내용은 다음과 같습니다
 - 1) 사진 구분용 ID (INT)
 - 2) S3에 저장된 사진 링크
3. multer 모듈을 사용해 AWS S3에 사진 1장을 업로드합니다
4. EC2에 express 프로젝트를 올려 pm2로 돌려주세요

11월 26일까지 소스파일은 해당 API의 자바스크립트 파일만
이름_training6.js 의 양식으로 드라이브에 올려주세요!

6 차 세 미 나

03. Scheduler

Scheduler의 필요성

- 서버에서 반복적인 일을 처리하는 경우가 많다
ex) 일주일마다 유저의 포인트 업데이트
일정기간이 지난 데이터 삭제
- 관리자가 일정시간마다 해당 작업을 수행하는 API를 사용해야만 한다
=> 매우 비효율적, 그냥 넘어가는 경우가 생길수도
- Scheduler를 사용해 작업을 자동화
- Node.js에서는 node-schedule 모듈을 사용
- CRON 표현식을 사용해 반복주기 시간을 정한다

CRON 표현식

- 왼쪽에서 오른쪽으로 최소 6개, 최대 7개의 파라미터를 사용해 표현

필드이름	값
seconds	0~59
minutes	0~59
hours	0~23
day of month	1~31, - * ? / L W
month	1~12 or JAN-DEC, - * /
day of week	1~7(1-일,7-토) or SUN-SAT, - * ? / L #
years (optional)	1970~2099, - * /

- <http://www.cronmaker.com/> 에서 쉽게 CRON 표현식을 만들 수 있다

- 기호의 의미

CRON 표현식

기호	의미	사용 예
*	모든 수를 의미	seconds에서 사용하면 매초, minutes에서 사용하면 매분, hours에서 사용하면 매시간
?	해당 항목을 사용하지 않음	day of month에서 사용하면 월중 날짜를 지정하지 않음. day of week에서 사용하면 주중 요일을 지정하지 않음
-	기간을 설정	hours에서 10-12이면 10시, 11시, 12시에 동작 minutes에서 58-59이면 58분, 59분에 동작
,	특정 시간을 지정	day of week에서 2,4,6이면 월,수,금에만 동작함
/	시작시간과 반복 간격 설정	seconds위치에 0/15로 설정하면 0초에 시작해서 15초 간격으로 동작 minutes위치에 5/10으로 설정하면 5분에 시작해서 10분 간격으로 동작
L	마지막 기간에 동작	day of month에서 사용하면 해당월 마지막 날에 수행 day of week에서 사용하면 토요일에 수행
W	가장 가까운 평일 동작	15W로 설정하면 15일이 토요일이면 가장 가까운 14일 금요일에 실행 15W로 설정하고 15일이 일요일이면 16일에 실행 15W로 설정하고 15일이 평일이면 15일에 실행
LW	L과 W의 조합	그달의 마지막 평일에 동작
#	몇 번째 주와 요일 설정	6#3이면 3번째 주 금요일에 동작 4#2이면 2번째 주 수요일에 동작

과제

1. 뉴스 콘텐츠 업로드 API를 구현해주세요 (POST)
2. 뉴스 콘텐츠의 구성요소는 다음과 같습니다.
 - 1) 뉴스 타이틀
 - 2) 뉴스 업로드 시간
 - 3) 뉴스 제작사(ex. MBC뉴스, 조선일보...)
 - 4) 뉴스 썸네일 이미지
 - 5) 뉴스 본문
 - 6) 뉴스 콘텐츠 이미지
3. 뉴스 업로드 시간의 경우 내림차순, 오름차순 정렬이 정상적으로 수행되게 저장합니다.
hint) string 형식의 2017.11.25 16:30:34 등의 양식은 정렬이 수행되지 않을 수 있습니다.
moment 모듈을 사용해 보도록 합시다.
4. 뉴스 이미지는 multer 모듈을 사용해 AWS S3에 저장합니다. - 썸네일, 콘텐츠 총 2장
저장 후 AWS S3에 존재하는 파일링크를 DB에 저장합니다.

12월 3일까지 소스파일은 해당 API의 자바스크립트 파일만
이름_homework6.js 의 양식으로 드라이브에 올려주세요!

다음주는 신나는 안드, 아요와 합동세미나!!