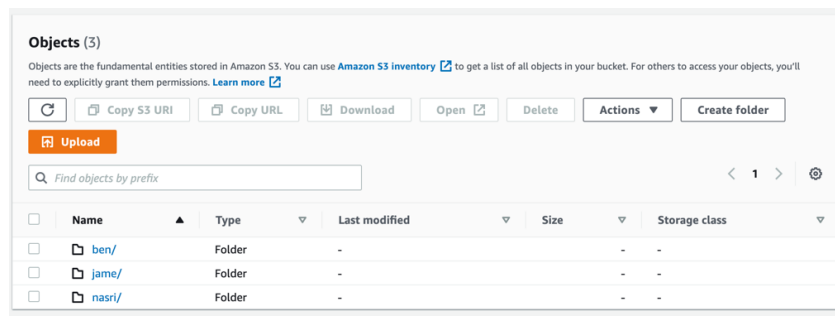


S3 design

1. How your many users will store their files in the same S3 bucket

ผู้ใช้งานแต่ละคน (user) จะเก็บไฟล์บนบัคเก็ต (AWS S3 Bucket) เดียวกัน โดยแต่ละผู้ใช้งานจะมีโฟลเดอร์ (Folder) สำหรับเก็บไฟล์เป็นของตนเอง โดยเมื่อส่ง POST request จะทำการแนบชื่อของผู้ใช้งานมาพร้อมกับไฟล์ ทำให้สามารถใช้ชื่อนั้นในการแยกไฟล์เข้าไปในแต่ละโฟลเดอร์ได้



รูปภาพที่ 1.1 แสดงโฟลเดอร์ของแต่ละผู้ใช้งานบน AWS S3 bucket

2. How you will know which file belongs to which user

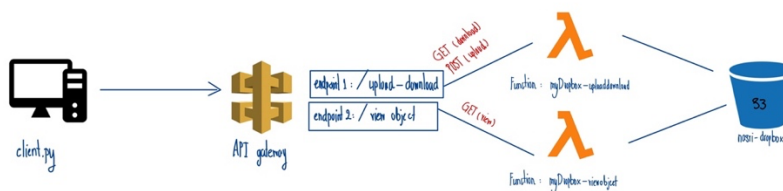
จากที่อธิบายในข้อที่ 1 เราสามารถรู้ว่าไฟล์ใดเป็นของใครโดยดูจากโฟลเดอร์ที่ไฟล์นั้นเก็บอยู่ อย่างไรก็ตามการกำหนดสิทธิการเข้าถึงไฟล์จะถูกกำหนดด้วย metadata บน DynamoDB ซึ่งจะอธิบายในส่วนถัดไป

3. How you will ensure that if user A uploads a file, and user B uploads another file with the same name, that they are NOT THE SAME object in your bucket

ตามที่อธิบายไปในข้อที่ 1 และข้อที่ 2 หากผู้ใช้งาน 2 คนอัปโหลดไฟล์ด้วยชื่อเดียวกัน ขณะส่ง POST request ตัว client จะแนบชื่อของผู้ใช้งานมาด้วยอัตโนมัติ โดย AWS lambda จะใช้ชื่อนั้นในการนำไฟล์ไปแยกเก็บตามโฟลเดอร์ของผู้ใช้งานนั้น ๆ ทำให้ถึงแม้ผู้ใช้งานหลายคนจะส่งไฟล์ด้วยชื่อเดียวกัน แต่ไฟล์จะไม่ถูกเก็บบนโฟลเดอร์เดียวกัน

หมายเหตุ - ชื่อผู้ใช้งาน (username) จะต้องไม่ซ้ำกันเลยในระบบ (unique)

4. System design



dynamoDB design

บน DynamoDB จะมีด้วยกัน 2 tables คือ myDropbox_auth และ myDropbox_objects โดยมีรายละเอียดดังนี้

1. myDropbox-auth ใช้สำหรับการทำ login – logout โดยมีรายละเอียดดังต่อไปนี้

Attributes:

username (partition key) - email ของผู้ใช้งาน

password - password ของผู้ใช้งาน

<input type="checkbox"/>	username	password
<input type="checkbox"/>	ben@gmail.com	qwerty123
<input type="checkbox"/>	ken@gmail.com	qwerty
<input type="checkbox"/>	nasri@gmail.com	mypassword

รูปภาพที่ 2.1 แสดงตารางของ myDropbox-auth table บน dynamoDB

2. myDropbox-objects ใช้สำหรับเก็บ metadata ของไฟล์ รวมถึงการแชร์ไฟล์และควบคุมสิทธิการเข้าถึงไฟล์ โดยมีรายละเอียดดังต่อไปนี้

Attributes:

username (partition key) – ชื่อผู้ใช้งานที่มีสิทธิเข้าถึงไฟล์ที่ระบุไว้บน attribute path

path (sort key) - ที่อยู่ (path) ของไฟล์บน AWS s3

contentLength - ขนาดไฟล์

lastModified - เวลาแก้ไขไฟล์ล่าสุด

owner - เจ้าของไฟล์ (ผู้สร้างไฟล์)

<input type="checkbox"/>	username	path	contentLength	lastModified	owner
<input type="checkbox"/>	1 ken@gmail.com	ken@gmail.com/ken01.txt	195	Tue, 01 Mar 202...	ken@gmail.com
<input type="checkbox"/>	ken@gmail.com	ken@gmail.com/ken02.txt	196	Tue, 01 Mar 202...	ken@gmail.com
<input type="checkbox"/>	ken@gmail.com	ken@gmail.com/ken03.txt	195	Tue, 01 Mar 202...	ken@gmail.com
<input type="checkbox"/>	2 nasri@gmail.com	ken@gmail.com/ken01.txt	195	Tue, 01 Mar 202...	ken@gmail.com
<input type="checkbox"/>	nasri@gmail.com	nasri@gmail.com/nasri01.txt	203	Tue, 01 Mar 202...	nasri@gmail.com
<input type="checkbox"/>	nasri@gmail.com	nasri@gmail.com/nasri02.txt	198	Tue, 01 Mar 202...	nasri@gmail.com
<input type="checkbox"/>	nasri@gmail.com	nasri@gmail.com/nasri03.txt	197	Tue, 01 Mar 202...	nasri@gmail.com

รูปภาพที่ 2.2 แสดงตารางของ myDropbox-objects table บน DynamoDB

โดยทั้ง 2 table จะใช้ eventual consistency

โดยหลักการทำงานเบื้องต้นคือ

put - เมื่อผู้ใช้งาน ken@gmail.com ทำการอัปโหลดไฟล์ ken01.txt ขึ้นไปยังระบบด้วยคำสั่ง "put ken01.txt" ระบบจะทำการอัปโหลดไฟล์ไปเก็บบน AWS S3 ที่ตำแหน่ง ken@gmail.com/ken01.txt และสร้างข้อมูลบน DynamoDB โดยเพิ่ม username และ owner เป็นชื่อของผู้ใช้งาน(ken@gmail.com) และข้อมูล metadata อื่นๆ ของไฟล์คือ path (ken@gmail.com/ken01.txt), contentLength และ lastModified (รูปที่ 2.2 เลข 1 สีแดง)

Share - ต่อมาเมื่อผู้ใช้งาน ken@gmail.com ต้องการแชร์ไฟล์ให้แก่ nasri@gmail.com ด้วยคำสั่ง "share ken01.txt nasri@gmail.com" ระบบจะทำการเพิ่มข้อมูลเหมือนกับข้อมูลก่อนหน้า (รูปที่ 2.2 เลข 1 สีแดง) แต่แก้ไขผู้มีสิทธิ์เข้าถึง (username) เป็น nasri@gmail.com โดยจะได้ข้อมูลแถวที่ 3 (รูปที่ 2.2 เลข 2 สีแดง)

get - เมื่อผู้ใช้งาน nasri@gmail.com ต้องการดาวน์โหลดไฟล์ ken01.txt ระบบจะตรวจสอบสิทธิการเข้าถึงโดยการค้นหาข้อมูลบน DynamoDB ที่มี username ตรงกับ nasri@gmail.com และดึง path จากข้อมูล path จาก metadata คือ ken@gmail.com/ken01.txt จากนั้นจะไปดึงไฟล์จาก AWS S3 และส่งไฟล์ให้เครื่องผู้ใช้งาน (client machine)

View - เมื่อผู้ใช้งานขอดูไฟล์ที่ตัวเองมีสิทธิ์เข้าถึงด้วยคำสั่ง view. ระบบจะดึงข้อมูลจาก DynamoDB โดย query เอาทุกช่องที่ username ตรงกับชื่อของผู้ใช้งานมาแสดง

Signup - ระบบเอา username และ password ไปเพิ่มบน DynamoDB หากมี username อยู่แล้วให้ปฏิเสธการขอสมัคร (signup)

Login - ระบบเอา username และ Password ไปเช็คกับข้อมูลบน database ว่าตรงไหม

A README for all of your code that tells me the name and functionality of each of the source code files you submit.

Lambda – myDropbox-uploaddownload

ไฟล์นี้จะเป็นไฟล์ source code ที่รันบน AWS Lambda ซึ่งรองรับ endpoint ดังต่อไปนี้

- GET /upload-download – ซึ่งทำงานบนฟังก์ชัน download(event)

บน endpoint นี้จะต้องแนบ username, filename และ fileowner มาด้วย โดยระบบจะเอา username, filename และ fileowner ไปตรวจสอบสิทธิ์การเข้าถึงไฟล์บน database ก่อนหากผู้ใช้งานมีสิทธิ์เข้าถึงระบบ จึงจะไป download ไฟล์

- POST /upload-download – ซึ่งทำงานบนฟังก์ชัน upload(event)

บน endpoint นี้จะต้องแนบ filename, content และ username มาด้วย โดยระบบจะเอา filename, content และ username ไปเพิ่มสิทธิ์การเข้าถึงของไฟล์และ metadata บน database ก่อนจากนั้นจึงอัปโหลดไฟล์ไปยัง AWS S3

Lambda – myDropbox-viewshare

ไฟล์นี้จะเป็นไฟล์ source code ที่รันบน AWS Lambda ซึ่งรองรับ endpoint ดังต่อไปนี้

- GET /viewobject – ซึ่งทำงานบนฟังก์ชัน viewobject(username)

บน endpoint นี้จะต้องแนบ username มากับ request ด้วย ระบบจะเอา username ไปเช็คข้อมูลบน database (dynamoDB) แล้วดึงเอา metadata ที่ผู้ใช้งานมีสิทธิ์เข้าถึงทั้งหมดมาแสดง

- POST /shareobject – ซึ่งทำงานบนฟังก์ชัน shareobject(username, objectName, targetUser)

บน endpoint นี้จะต้องแนบ username, filename, targetUser มาด้วย ระบบจะเอา username และ filename ไปดึงข้อมูล metadata ของไฟล์จาก database มา จากนั้นจะเอาข้อมูล metadata ของไฟล์นั้นไปเพิ่มบน database โดยข้อมูลที่เพิ่มนี้จะใส่ชื่อผู้ที่มีสิทธิ์เข้าถึงเป็นชื่อของ targetUser

Lambda – myDropbox-auth

ไฟล์นี้จะเป็นไฟล์ source code ที่รันบน AWS Lambda ซึ่งรองรับ endpoint ดังต่อไปนี้

- POST /signup – ซึ่งทำงานบนฟังก์ชัน signup(username, password)

บน endpoint นี้จะต้องแนบ username, password มาด้วย โดยระบบ (AWS Lambda) จะเอา username ไปเช็คบน myDropbox-auth table บน database (dynamoDB) ว่าเคยมี username นี้ปรากฏมาก่อนหรือไม่ หากไม่เคยจะเอา username, password เพิ่มเข้าไปยัง database

- POST /login – ซึ่งทำงานบนฟังก์ชัน login(username, password)

บน endpoint นี้จะต้องแนบ username, password มาด้วย โดยระบบ (AWS Lambda) จะเอา username, password ไปเช็คบน myDropbox-auth table บน database (dynamoDB) ว่าตรงหรือไม่

Client – client.py

ไฟล์นี้จะเป็น source code ที่รันบนเครื่อง client โดยจะสร้างหน้าต่างเป็น command line เพื่อรับคำสั่งจากผู้ใช้งานและส่ง request ไปยัง endpoint ของ API-gateway ที่กำหนดไว้ตามแต่ละฟังก์ชัน

A HOWTO for your API (need to specify http endpoints and request/ response formats)

Put object – POST request to endpoint - <https://oaa60ku0pj.execute-api.ap-southeast-1.amazonaws.com/prod/upload-download>

Request format

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	nasri.txt			
<input checked="" type="checkbox"/>	username	nasri		
	Key	Value		Description

Response format

```
{
  "statusCode": 200,
  "body": "OK"
}
```

Get object -- GET request to endpoint - <https://oaa60ku0pj.execute-api.ap-southeast-1.amazonaws.com/prod/upload-download>

Request format

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	nasri		
<input checked="" type="checkbox"/>	filename	nasri.txt		
<input checked="" type="checkbox"/>	fileOwner	nasri		

Response format

```
{'ResponseMetadata': {'RequestId': '54QAQWSSW1VAVKEW', 'HostId':
'GRxpJVNIpca+Njn7A4v1pCyFz1DuWZ7tCFrH4cPmde9vDZ/kB7/mlC78LBxUglhRPvtrIs7eqz4=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2':
'GRxpJVNIpca+Njn7A4v1pCyFz1DuWZ7tCFrH4cPmde9vDZ/kB7/mlC78LBxUglhRPvtrIs7eqz4=', 'x-amz-request-id': '54QAQWSSW1VAVKEW', 'date': 'Mon, 21
Feb 2022 19:18:20 GMT', 'last-modified': 'Mon, 21 Feb 2022 18:47:17 GMT', 'etag': '"05580f71afbd3f03d67f9eeb1d1b1c35"', 'accept-ranges': 'bytes', 'content-
type': 'binary/octet-stream', 'server': 'AmazonS3', 'content-length': '27'}, 'RetryAttempts': 0}, 'AcceptRanges': 'bytes', 'LastModified': '2022-02-21
18:47:17+00:00', 'ContentLength': 27, 'ETag': '"05580f71afbd3f03d67f9eeb1d1b1c35"', 'ContentType': 'binary/octet-stream', 'Metadata': {}, 'Body':
'b'aGVsbG8gdGhpcyBpcyBteSB0ZXN0IGZpbGUu'}
```

Demo - <https://youtu.be/ju0p-LzfoUw>

View object - GET request to endpoint - <https://oaa60ku0pj.execute-api.ap-southeast-1.amazonaws.com/prod/viewobject>

Request format

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	nasri	
Key	Value	Description

Response format

```
{
  "statusCode": 200,
  "Body": "list of metadata of object for example - ['mytest.txt 27 Mon, 21 Feb 2022 19:19:20 GMT nasri', 'nasri.txt 13 Mon, 21 Feb 2022 18:47:27 GMT nasri']"
}
```

Share object – POST request to endpoint - <https://oaa60ku0pj.execute-api.ap-southeast-1.amazonaws.com/prod/shareobject>

Request format

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	ken	
<input checked="" type="checkbox"/> filename	mytest.txt	
<input checked="" type="checkbox"/> targetuser	nasri	
Key	Value	Description

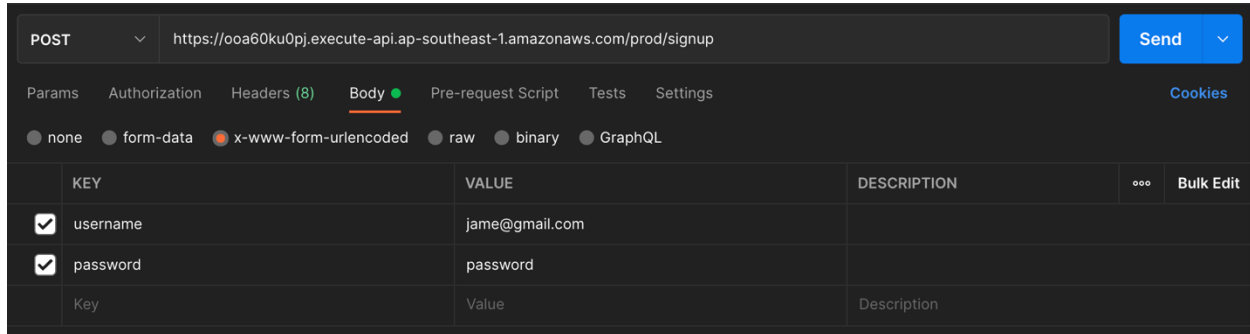
Response format

```
{
  "statusCode": 200,
  "Body": "OK"
}
```

Demo - <https://youtu.be/ju0p-LzfoUw>

Signup – POST request to endpoint - <https://oaa60ku0pj.execute-api.ap-southeast-1.amazonaws.com/prod/signup>

Request format



POST <https://oaa60ku0pj.execute-api.ap-southeast-1.amazonaws.com/prod/signup> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data **x-www-form-urlencoded** raw binary GraphQL

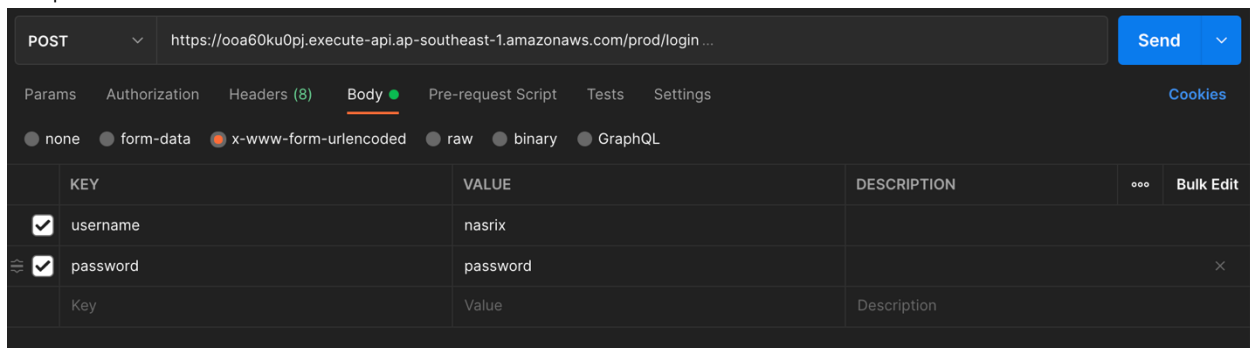
	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	jame@gmail.com			
<input checked="" type="checkbox"/>	password	password			
	Key	Value	Description		

Response format

```
{
  "statusCode": 200,
  "Body": "OK"
}
```

login – POST request to endpoint - <https://oaa60ku0pj.execute-api.ap-southeast-1.amazonaws.com/prod/login>

Request format



POST <https://oaa60ku0pj.execute-api.ap-southeast-1.amazonaws.com/prod/login> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data **x-www-form-urlencoded** raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	nasrix			
<input checked="" type="checkbox"/>	password	password			×
	Key	Value	Description		

Response format

```
{
  "statusCode": 200,
  "Body": "OK"
}
```

Demo - <https://youtu.be/ju0p-LzfoUw>