

S3 design



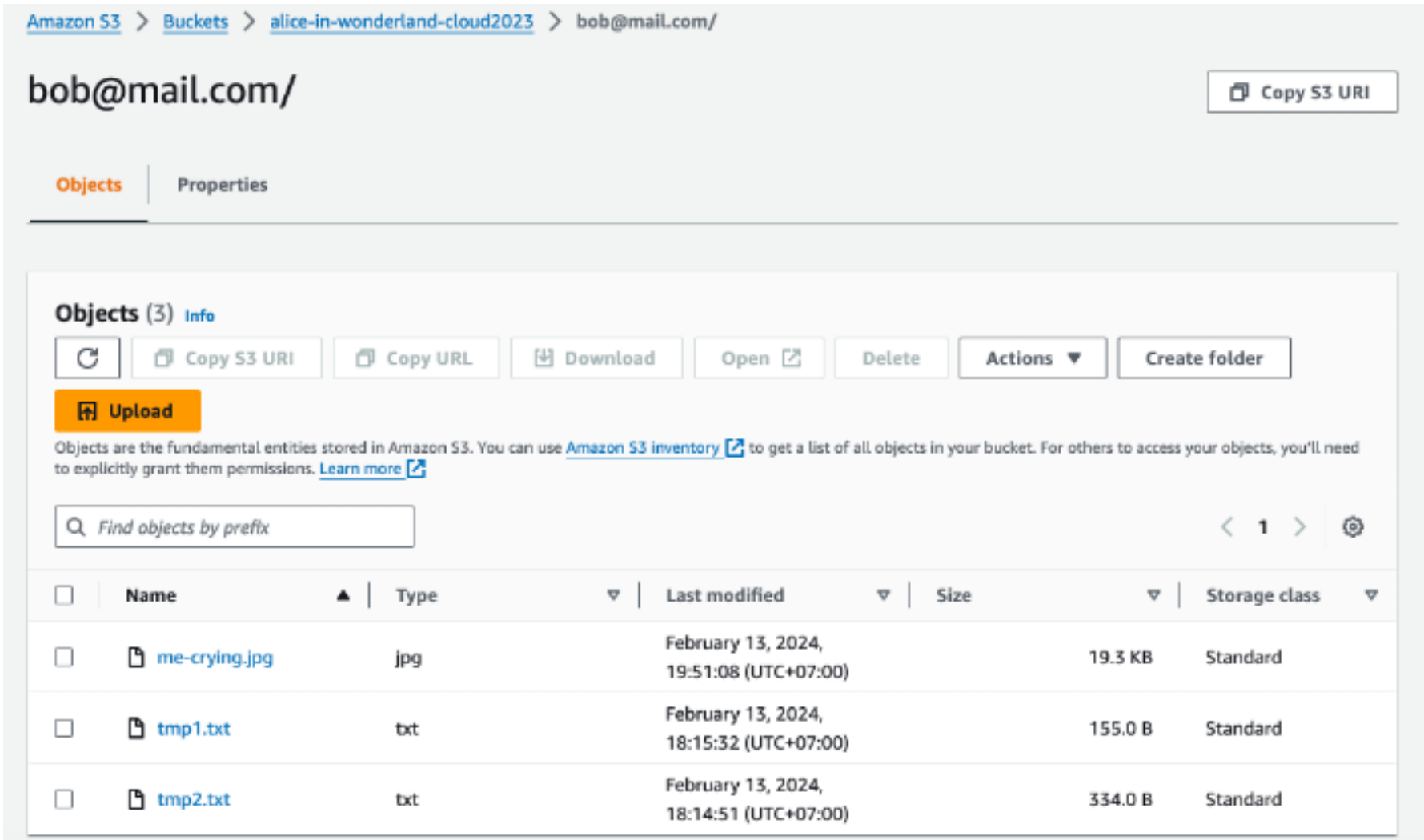
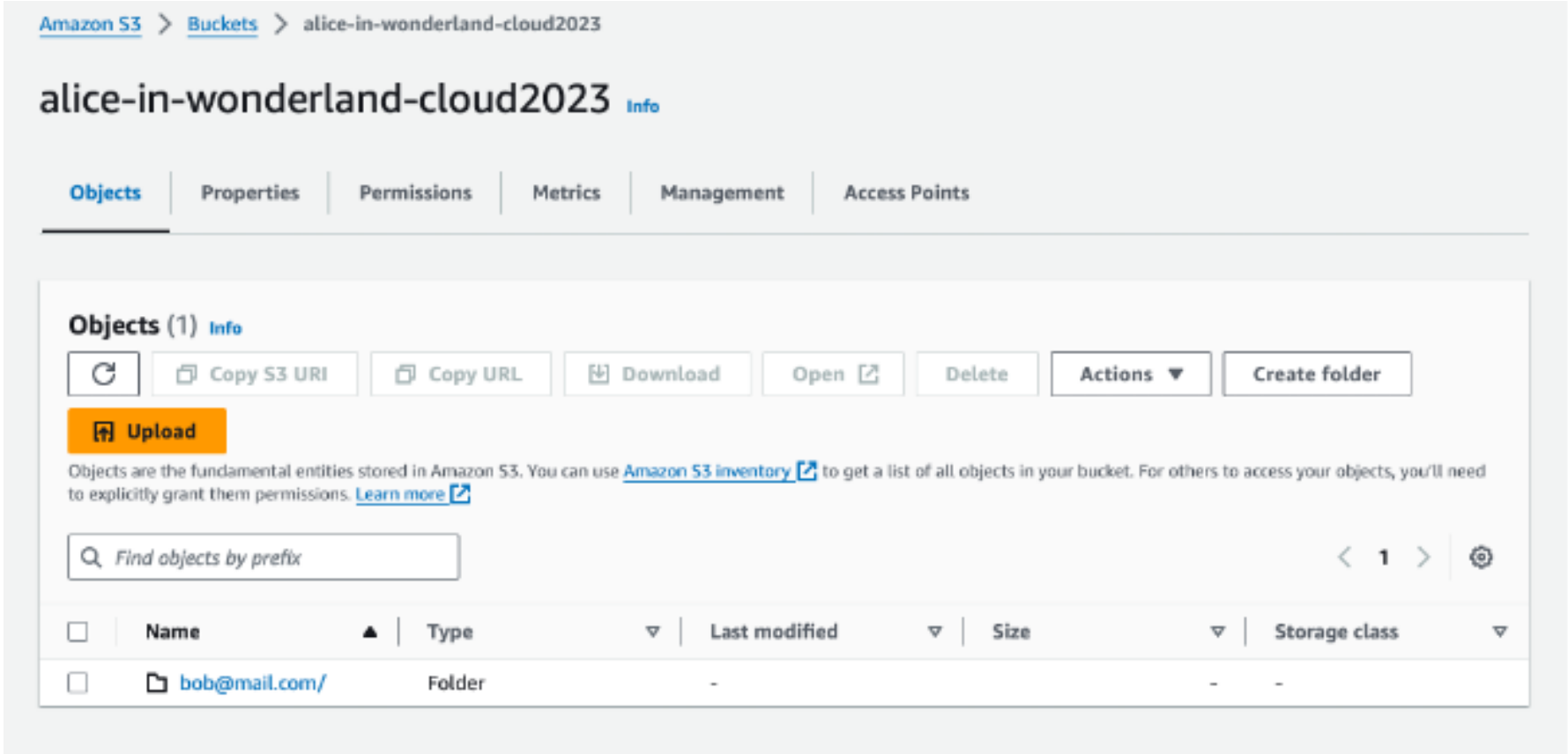
S3 Bucket

I have decided to organize my S3 bucket to store folders named after users' emails because myDropbox uses email for login authentication, ensuring uniqueness. Each user's files will be stored within a folder named after their email address, allowing **multiple users to store their files in the same S3 bucket** while **easily discerning which files belong to whom**. Additionally, this structure **guarantees that if User A uploads a file and User B uploads another file with the same name, they are treated as separate objects in the bucket**.

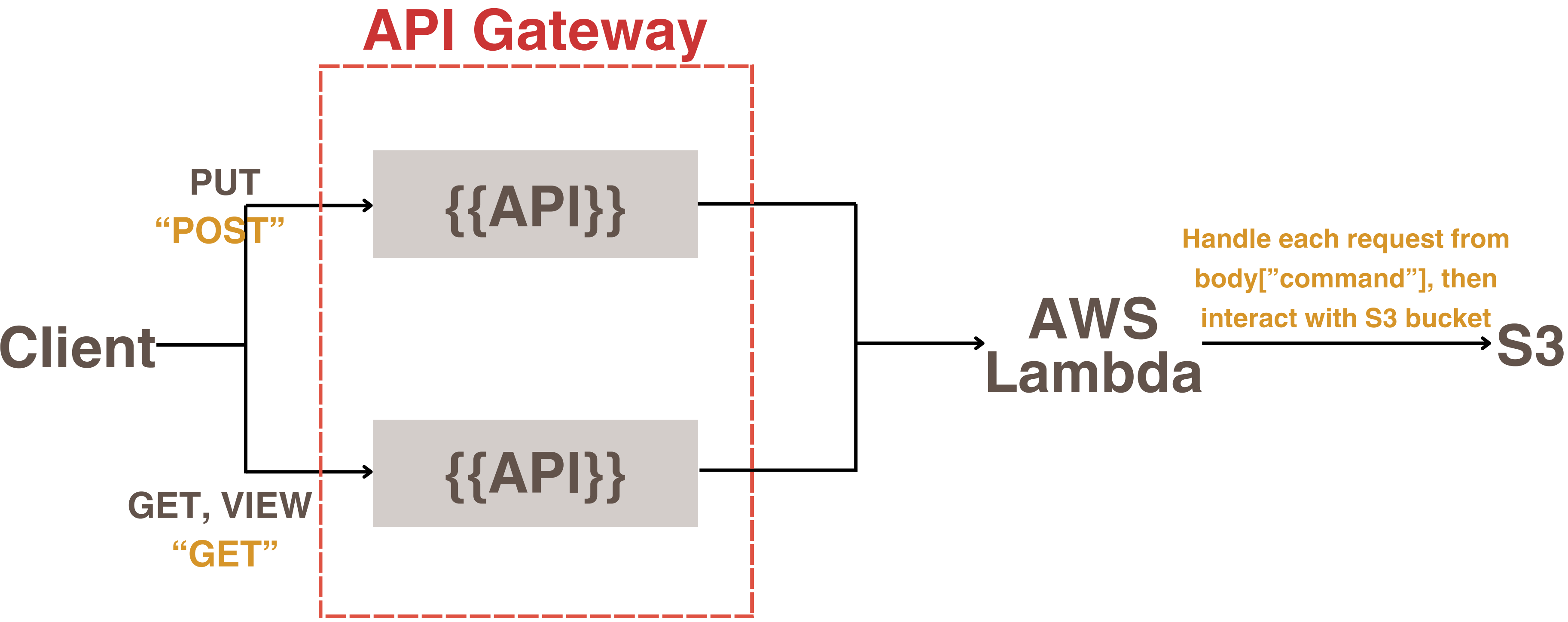


bob@mail.com folder

Assignment 5 myDropbox



Overview System



myDropbox Application

README.md

When running myDropboxClient file, you will see the result as following

Welcome to myDropbox Application

=====

Please input command (newuser username password password, login
username password, put filename, get filename, view, or logout).

If you want to quit the program just type quit.

=====

After that, you can use the myDropbox Application, which supports the following commands:

PUT

- The 'put' command is used to upload a file from local to S3:

```
> put tmp1.txt
```

GET

- The 'get' command is used to download user's file from S3 to local:

```
> get tmp1.txt bob@mail.com
```

VIEW

- The 'view' command is used to display all files of the user, including filename, file size, last modified date, and the owner:

```
> view
```

QUIT

- The 'quit' command is used to exit the application:

```
> quit
```

HOW TO for API

PUT

- HTTP method: POST
- headers: {"Content-Type": "application/json"}
- body: body = {"command": "put",
"user": user,
"filename": filename,
"file": data}
- successful response: {"statusCode": 200,
"body": {"success": True,
"message": "PUT successful"}}

GET

- HTTP method: GET
- headers: {"Content-Type": "application/json"}
- body: body = {"command": "get",
"user": user,
"filename": filename,
"shareuser": shareuser}
- successful response: {"statusCode": 200,
"body": {"success": True,
"message": "GET successful",
"content": content}}

VIEW

- HTTP method: GET
- headers: {"Content-Type": "application/json"}
- body: body = {"command": "view",
"user": user}
- successful response: {"statusCode": 200,
"body": {"success": True,
"message": "VIEW successful",
"contents": contents}}

Result

```
Welcome to myDropbox Application
```

```
=====
```

```
Please input command (newuser username password password, login  
username password, put filename, get filename, view, or logout).  
If you want to quit the program just type quit.
```

```
=====
```

```
myDropbox>> put tmp2.txt
```

```
OK
```

```
myDropbox>> get tmp2.txt bob@mail.com
```

```
OK
```

```
myDropbox>> view
```

```
OK
```

```
tmp1.txt 155 2024-02-13 09:47:14+00:00 bob@mail.com
```

```
tmp2.txt 334 2024-02-13 09:55:02+00:00 bob@mail.com
```

```
myDropbox>> quit
```

```
=====
```


myDropbox command	HTTP method	body
PUT	POST	<pre>{ "command": "put", "user": user, "filename": filename, "file": data }</pre>
GET	GET	<pre>{"command": "get", "user": user, "filename": filename, "shareuser": shareuser}</pre>
VIEW	GET	<pre>{"command": "view", "user": user}</pre>