Upload photos to S3 bucket

Family's holiday 2015'

Samsung AC

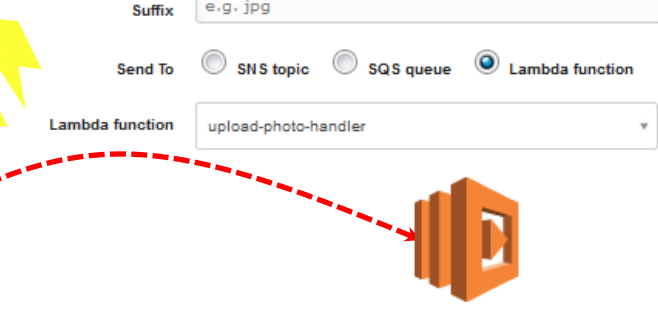
Samsung Galaxy 3

iPhone

AWS Cloud

Amazon S3

upload-photos-ext



Amazon Lambda

```
/**
 * @author smolnik
 *
 */
public class UploadPhotoHandler {

    public void handle(S3Event s3Event, Context context) {
        LambdaLogger log = context.getLogger();
        ExecutorService es = Executors.newCachedThreadPool();
        try {
            s3Event.getRecords().forEach(record -> {
                try {
                    process(new S3ObjectStream(record.getS3(), record.getUserIdentity()), log, es);
                } catch (IOException e) {
                    throw new UploadPhotoHandlerException(e);
                }
            });
        } finally {
            es.shutdownNow();
        }

    }
```

2

Amazon Lambda

JPEG?

```
/**
 * @author asmolnik
 *
 */
public class UploadPhotoHandler {

    public void handle(S3Event s3Event, Context context) {
        LambdaLogger log = context.getLogger();
        ExecutorService es = Executors.newCachedThreadPool();
        try {
            s3Event.getRecords().forEach(record -> {
                try {
                    process(new S3ObjectStream(record.getS3(), record.getUserIdentity()), log, es);
                } catch (IOException e) {
                    throw new UploadPhotoHandlerException(e);
                }
            });
        } finally {
            es.shutdownNow();
        }

    }
```

All Buckets / smolnik.photos / photos / default

Name
2015-06-16-10-47-43-38863000-927fe43c.jpg
2015-06-16-10-58-54-39534000-12828b29.jpg
2015-06-16-11-01-47-39707000-b67db998.jpg
2015-06-16-11-03-12-39792000-7061e124.jpg

Amazon S3

arch.smolnik.photos

smolnik.photos

Upload photo handler job to accomplish:

- ✓ *Image detection*
- ✓ *Renaming to a common format*
- ✓ *Image resizing*
- ✓ *JPEG metadata extraction*
- ✓ *Storing image data in a DB table for searching*
- ✓ *Archiving*

photos

Amazon DynamoDB Explore Table: photos

List Tables | Browse Items

Scan | Query | Go | Create Item | Edit Item | Copy to New | Details | Delete Item | Export to .csv

Scan On: [Table] photos: userId, photoKey | Add Filter | Start New Scan

userId	photoKey	bucket	madeBy	model	photoTakenDate
default	photos/default/2015-0	smolnik.photos	SAMSUNG	SM-G357FZ	2015-06-16
default	photos/default/2015-0	smolnik.photos	SAMSUNG	SM-G357FZ	2015-06-16

DynamoDB

Browse your photo gallery

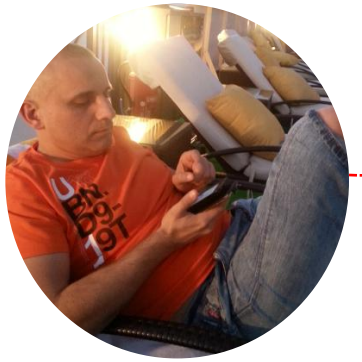

Route 53

+

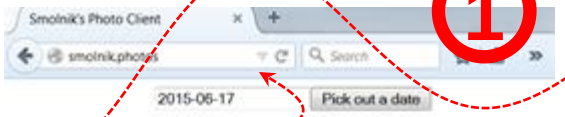

Amazon S3

=

static website hosting, but with JS in place, it's not so static...



Fetch the list of photos for a given date

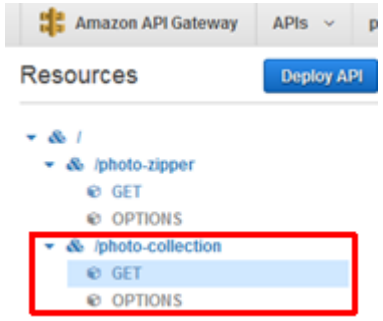


1

2


Amazon Lambda


DynamoDB

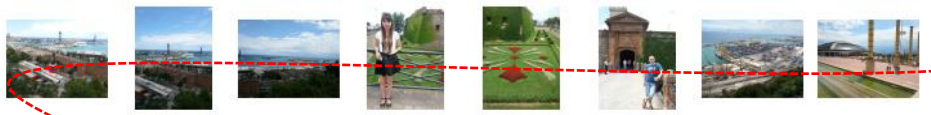


Both services can work in tandem

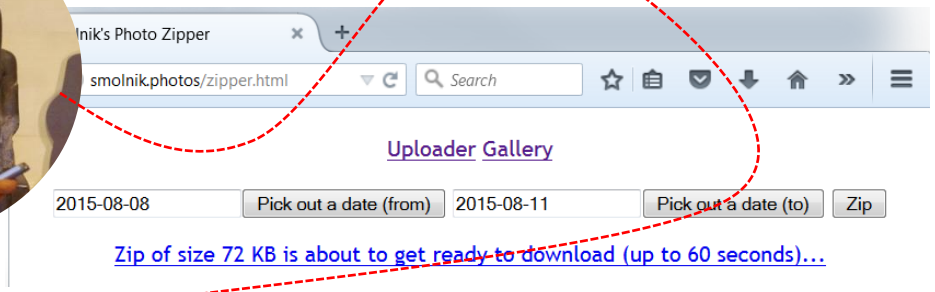

query for a given date
from table photos

Retrieve photos based on the list received


smolnik.photos



Download zipped photos for a date range



DynamoDB

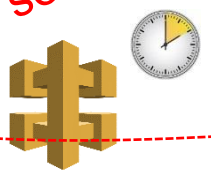
Amazon S3

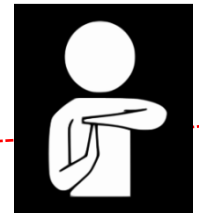
Fetch the list of photos

query for a given date
from table photos

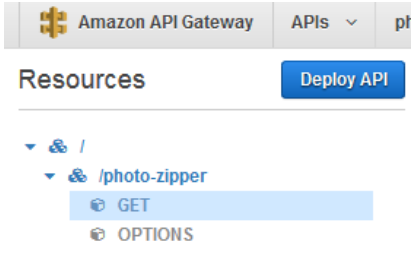
smolnik.photos

max. 10 seconds





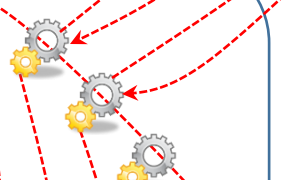
max. 60 seconds

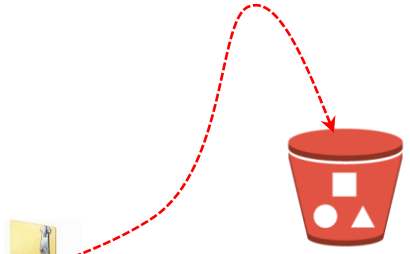



```
/**
 * Experimental and questionable usage of AWS Lambda service
 * ...
 *
 * @author asmolnik
 *
 */
public class PhotoZipperHandler extends PhotoHandler {

    private static final String ZIP_BUCKET = "zip.smolnik.phc

    public PhotoZipperResponse handle(PhotoZipperRequest requ
        Logger log = new Logger(context);
```





zip.smolnik.photos

Send the compressed file to S3 bucket

Retrieve photos (simultaneously) and then zip them