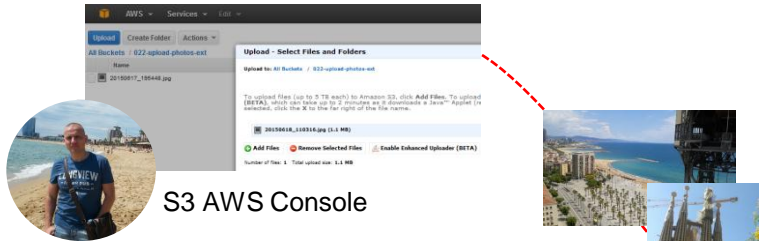


Upload photos to S3 bucket

1

Family's holiday 2015'



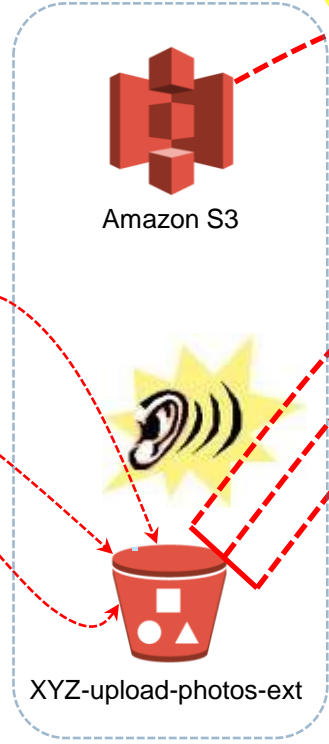
Samsung Galaxy 3



iPhone



AWS Cloud



Name	Event(s)
upload-photo-event	Put

Name:

Events:

Prefix:

Suffix:

Send To: SNS topic SQS queue Lambda function

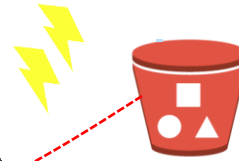
Lambda function:

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

    private static final String STUDENT_PREFIX = "001";
    private static final String PHOTOS_TABLE_NAME = STUDENT_PREFIX + "-codepot-photos";
    private static final String DEST_BUCKET = STUDENT_PREFIX + "-codepot-photos";
    ...

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

2



XYZ-upload-photos-ext



Amazon S3

All Buckets / 022-codepot-photos / photos / default

	Name
<input type="checkbox"/>	2015-06-17-19-54-48-71688000-ea00f149.jpg
<input type="checkbox"/>	thumbnails



XYZ-codepot-photos

```

* @author asmolnik
*/
public class UploadPhotoHandler {
    private static final String STUDENT_PREFIX = "001";
    private static final String PHOTOS_TABLE_NAME = STUDENT_PREFIX + "-codepot-photos";
    private static final String DEST_BUCKET = STUDENT_PREFIX + "-codepot-photos";
    ...
    public void handle(S3Event s3Event, Context context) {
        LambdaLogger log = context.getLogger();
        s3Event.getRecords().forEach(record -> {
            try {
                process(new S3ObjectStream(record.getS3(), record.getUserIdentity()), log);
            } catch (IOException e) {
                throw new UploadPhotoHandlerException(e);
            }
        });
    }
}

```



XYZ-codepot-photos

Amazon DynamoDB Explore Table: photos

Scan On: [Table] photos: userID, photoKey

userId	photoKey	bucket	madeBy	model	photoTakenDate	p
default	photos/default/2015-0	samolnik.photos	SAMSUNG	SM-G357FZ	2015-06-16	1
default	photos/default/2015-0	samolnik.photos	SAMSUNG	SM-G357FZ	2015-06-16	1



DynamoDB

Upload photo handler job to accomplish:

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

Browse your photo gallery



=



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

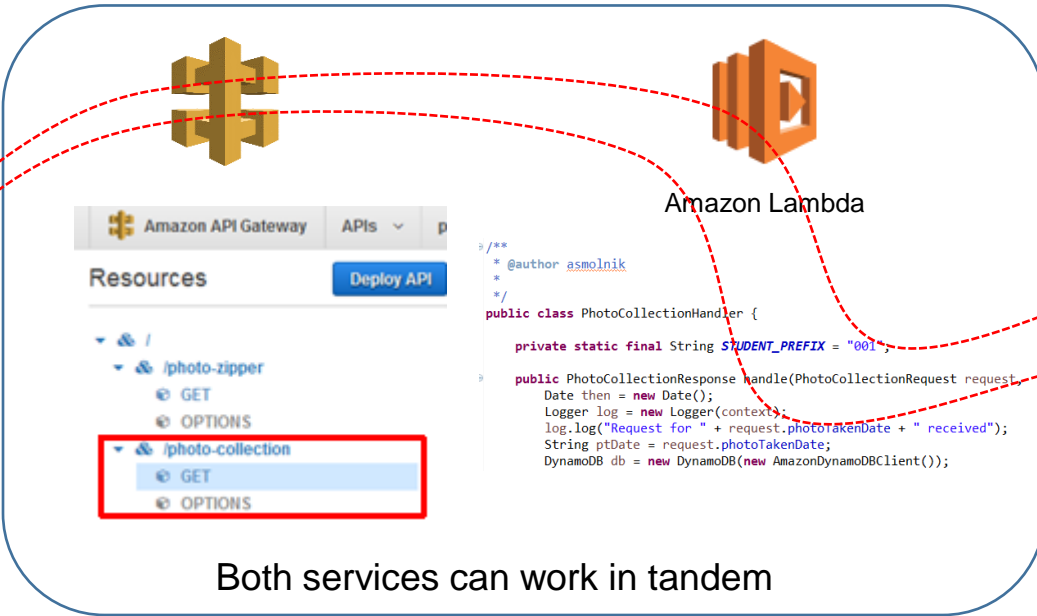
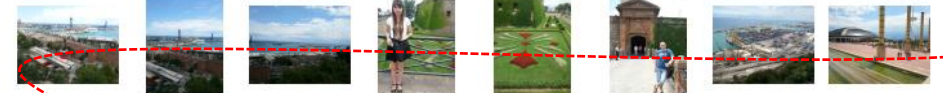


2

Retrieve photos based on the list received



XYX-codepot-photos



```
/**  
 * @author asmolnik  
 */  
public class PhotoCollectionHandler {  
    private static final String STUDENT_PREFIX = "001";  
    public PhotoCollectionResponse handle(PhotoCollectionRequest request,  
    Date then = new Date();  
    Logger log = new Logger(context);  
    log.log("Request for " + request.photoTakenDate + " received");  
    String ptDate = request.photoTakenDate;  
    DynamoDB db = new DynamoDB(new AmazonDynamoDBClient());
```



DynamoDB



query for a given date from table XYZ-codepot-photos

Set the appropriate AWS Region!

