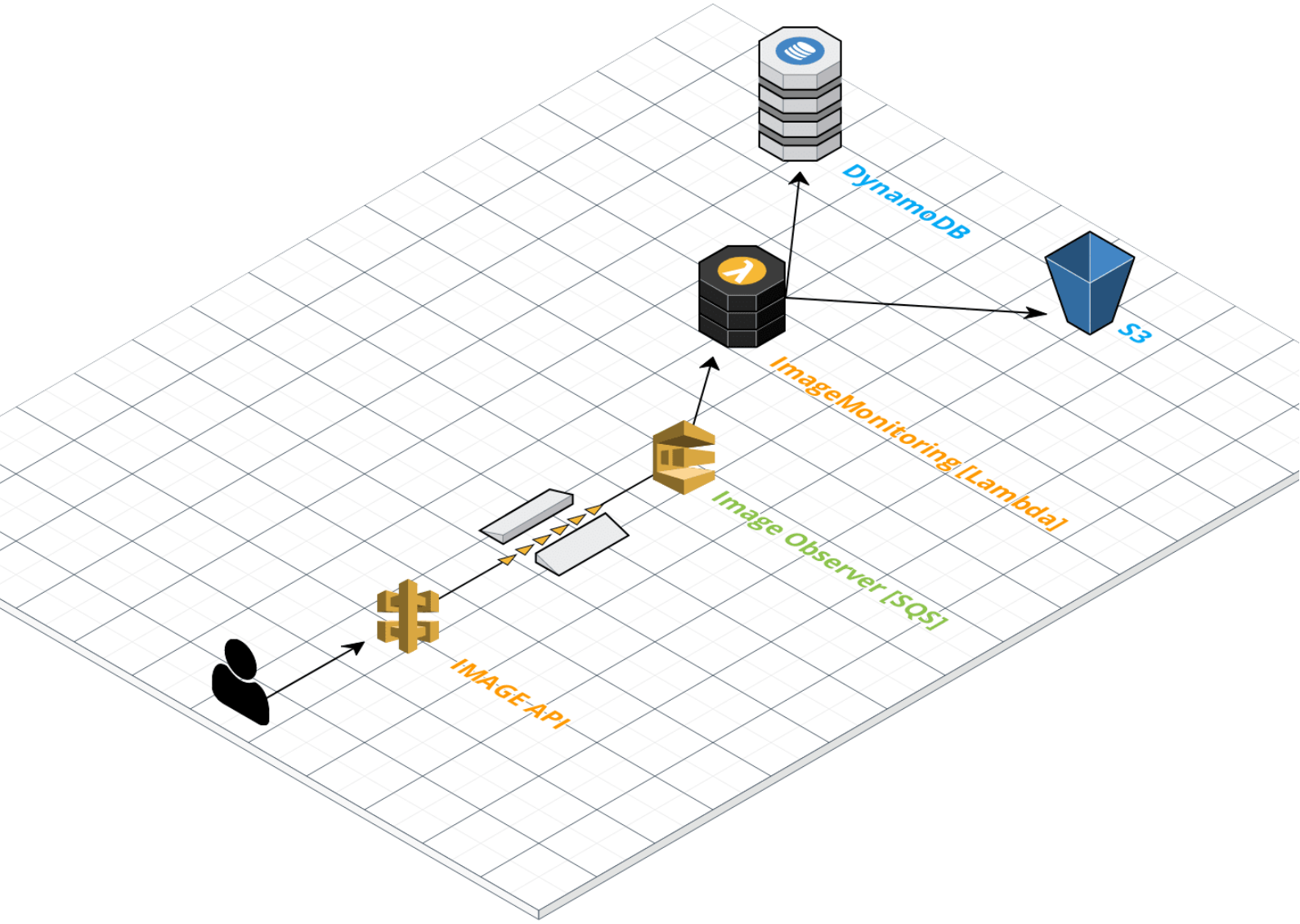
# BI Engineer Challenge for FashionCloud
Fernand Ramat - Sunday 11/07
“Version 1”




- **Image API**: Where the user can request/download Images. integrated with the “Image Observer” SQS queue.
- **Image Observer**: Service responsible of receiving raw payload from Image API and transfer it (via lambda subscription) to ImageMonitoring lambda function
- **ImageMonitoring**: Lambda function which is responsible of archiving the payload into S3 bucket and store the image information in dynamoDB

I assume the API (“IMAGE API”) is built under Amazon API gateway and we can easily integrate it with the image Observer queue (amazon SQS service); if not, we will have to build a SQS API endpoint.

I decided to use cloudformation to implement the solution, you will find the infrastructure described in *template.yaml* file.

There is one lambda function (ImageMonitoring, stored in *./handlers* folder) which have two responsibilities and so two python functions:

- archive the raw payload in s3 bucket (*archive_image()* function)
- store the image information in dynamoDB from payload (*store_image()* function)

To test:

- Create a “**bi-source**” S3 bucket manually and upload *./handlers/imageMonitoring.zip* in *bi-source/lambda/* folder.
- In AWS cloudformation service, create a new stack from an existing template and use *./template.yaml* file
- In AWS SQS service, select “Send and receive messages”, and send the following raw payload (also available in *./example.json* file:

```
{
   "event_name": "IMAGE_REQUESTED",
   "user_id": "42",
   "image_id": "b9f08c7f-089e-4cbf-ba78-a89a6e586e86",
   "timestamp": "2021-07-11T09:17:13+00:00"
}
```

Possible improvements:

- I attached a version 2 where there is a proxy lambda function in between the “ImageObserver” queue and a new aws step function. The version 2 is a bit more complicated to implement and I prefered focus on this version to not lose too much time. But version 2 has the advantage of allowing us to separate the responsibility bundled in the current “ImageMonitoring” lambda function to two lambda functions, managed by one step function.
- The deployment and the testing is really basic and I already apologize for that, I honestly missed some time to offer you a more convenient version for now. But we can think of sam command line to pack the handler/python code, and deploy it to the AWS ecosystem. Then on top of that we could implement some CI/CD pipeline through github. I have more experience with bitbucket pipeline & aws-sam-deploy but it should be fairly with github pipeline.
- About the python code: For now it’s very basic, and implementing some unit tests will not hurt the same as adding  exception handling.
- Getting notified if something went wrong with sentry alerts for example. It Would also be nice to get notified if something went well with something like AWS SNS notification for example.