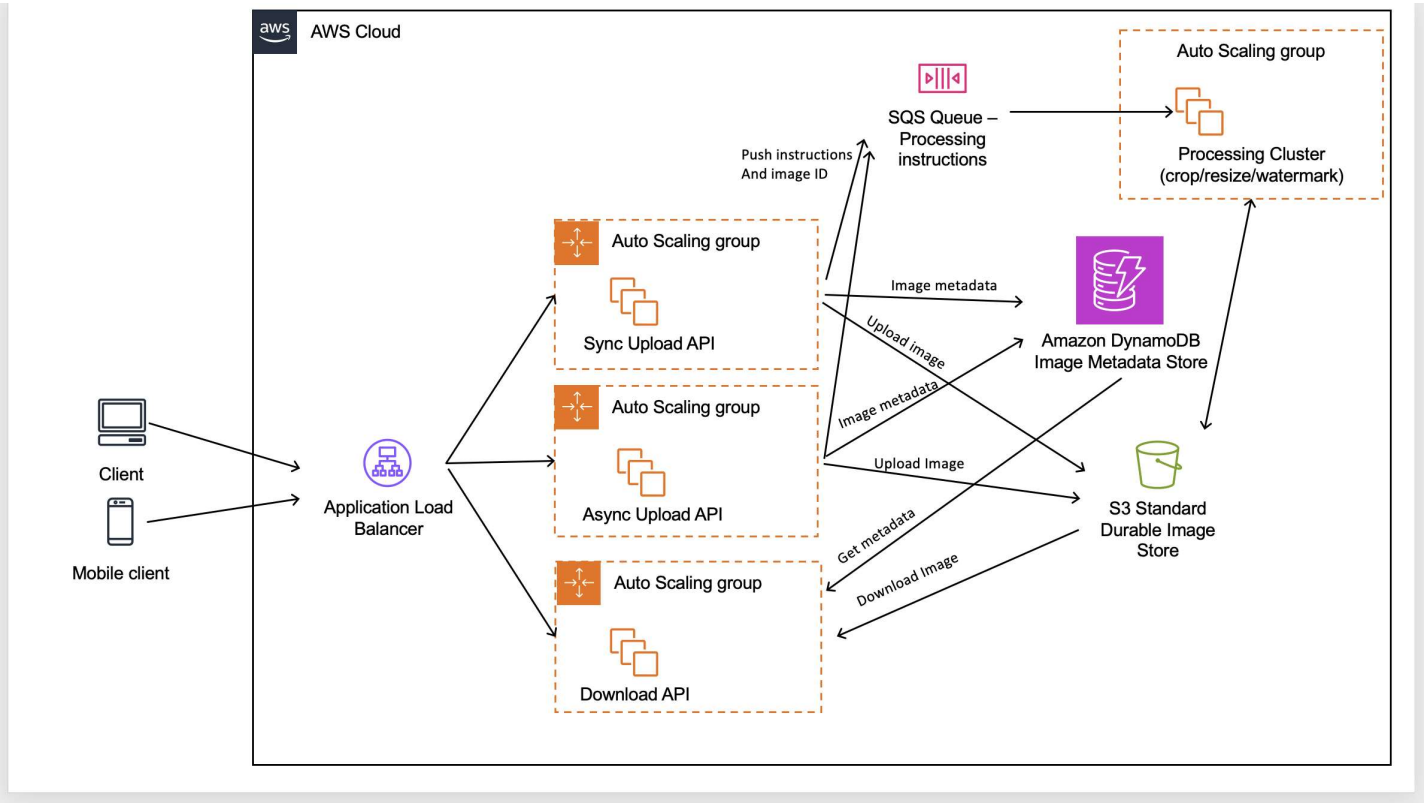

Image service:

- Image upload occurs through load balancer and goes to either Sync Upload API (for small workloads, e.g. up to 100 MB) or to Async API.

- Async API might be similar to a multipart upload API of S3 – CreateMultipartUpload, UploadPart, CompleteMultipartUpload methods/endpoints

- Sync and Async API are hosted by difference servers (and scaling groups), because they have different workloads and might use different instance types.

- Download API is hosted separately for the same reason.

- Image upload is done by Upload API servers. Image metadata gets stored in DynamoDB. I use NoSQL DB here, because we require High Availability. Eventual consistency should be enough for this use case, so no need to relation databases.

- API servers store images on S3 because it’s durable and cheap. It’s also possible to apply different features (e.g. life cycle policies) to decrease costs.

- Image processing (crop, watermark, etc), is done by a processing cluster that scales separately from API servers. To avoid bottlenecks and back pressure issues, I've added SQS queue between APIs and Processing Cluster.

- SQS Queue does not process images. It only passes image ID and commands. After processing cluster consumes the message, it downloads image from S3, does necessary operations and uploads it back to S3.

Notes:
- This architecture is based on AWS, but is not bound to it. It's possible to replace AWS services, such as EC2, Load Balancer, DynamoDB, etc by the similar services of other providers (e.g. Blob storage, MongoDB etc).

- It's also possible to be cloud agnostic and use orchestration platforms such as Kubernetes to deploy the required servers and scale them.

- If sticking to AWS, there are additional options for reducing costs. For example, we might use Spot instances for Processing Cluster. Also, if some APIs are not used too often/steadily, we might replace instances by Lambda + API Gateway.

- CDN (e.g. CloudFront) can be added to decrease costs of image download in case there are images that get often downloaded and might be cached.