Section 10: Other Compute Services: EC2, Lambda, Batch, Lightsail

What is Docker ?

- Docker is a software development platform to deploy app.
- Apps are packeged in containers that can be run on any os.
- Apps run the same, regardless of where they're run
  - Any machine
  - No Compatibility issues
  - Predictable behaviour
  - Less work
  - Easier to maintain and deploy
  - Works with any language, any OS, any technology.
- Scale containers up & down very quickly (seconds)

Where Docker images are stored?

- Docker images are stored in Docker Reposi-
  -tories.

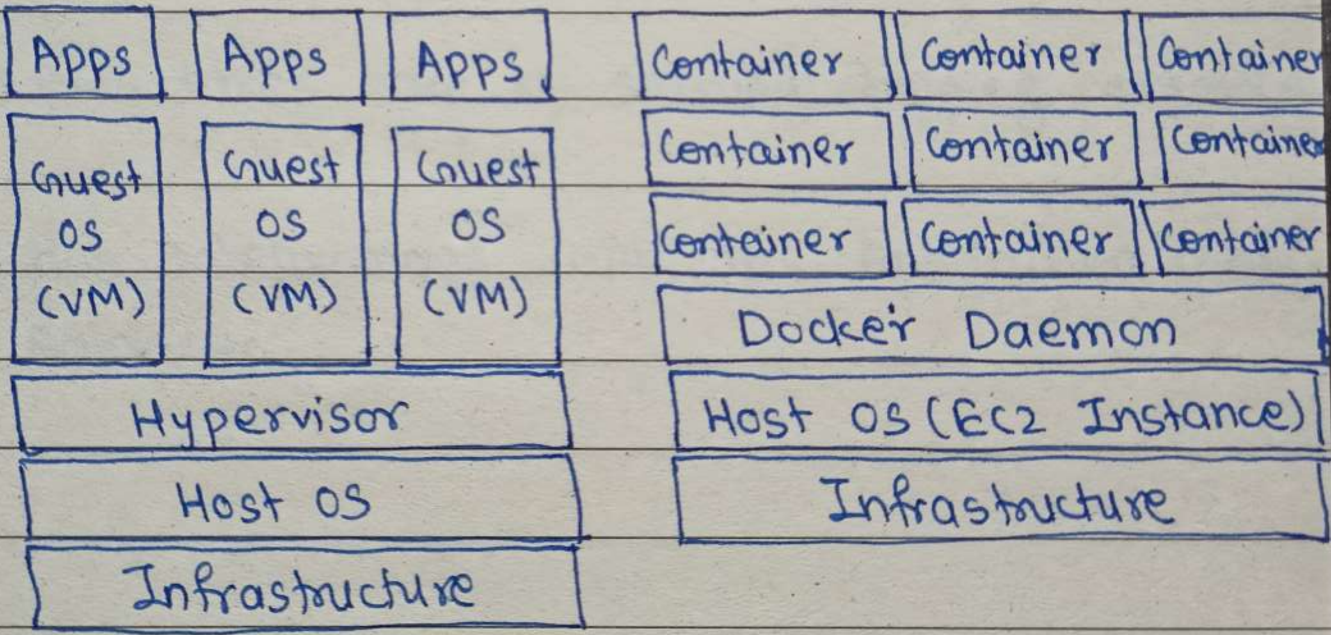
- Public : Docker Hub https://hub.docker.com/
  - Find base images for many technologies

*   or OS:
  - Ubuntu
  - MySQL
  - NodeJs, Java,......

- Private : Amazon ECR (Elastic Container Registry)

Docker versus Virtual Machines

- Docker is "sort of" a virtualization technology, but not exactly
- Resources are shared with the host ⇒ many containers on one server



ECS

- ECS = Elastic Container Service.

- Launch Docker containers on AWS.

- You must provision & maintain the infrastructure (the EC2 instances)

- AWS takes care of starting/stopping containers.

- Has integrations with the Application Load Balancer.

Fargate

- Launch Docker containers on AWS.

- You do not provisione the infrastructure (no EC2 instances to manage) - simpler!

- Serverless offering

- AWS just runs containers for you based on the RCPU/RAM you need.

ECR

- Elastic Container Registry.
- Private docker Registry on AWS.
- This is where you store your Docker Images so they can be run by ECS or Fargate.

What is Serverless?

· Serverless is a new paradigm in which the developer's dont have to manage servers anymore. --

· They just deploy code.

· They just deploy .... functions!

· Intially...serverless == Faas (Function as a service)

· Serverless was pioneered by AWS Lambda but now also includes anything that's managed : "databases, messaging, storage, etc".

° Serverless does not mean there are no servers ... it means you just don't manage / provision / see them example of Serverless Service in AWS:-

1] Amazon S3
2] Fargate
3] DynamoDB
4] Lambda

Lambda Overview

Why AWS Lambda?

1] Amazon EC2 :- Virtual servers in the cloud
- Limited by RAM & CPU
- Continuously running
- Scaling means intervation to add/remove servers.

2] Amazon lambda :- Virtual functions - no server to manage!
- Limited by time - short executions
- Run on-demand
- Scaling is automated.

Benefits of AWS Lambda

- Easy Pricing:-
  - Pay per request & compute time
  - Free tier of 1000,000 AWS Lambda requests & 4000,000 GBs of compute ~~service~~ time.

- Integrated with the whole AWS suite of services.

- Event Driven: functions get invoked by AWS when needed.
  This makes it reactive service in AWS
- Integrated with many programming lang.

- Easy monitoring through AWS CloudWatch.

- Easy to get more resources per functions (upto 10GB of RAM)

• Increasing RAM will ~~be~~ also improve CPU & network!

• Lambda Container Image

- The container image ~~ss~~ must implement the lambda Runtime API.
- ECS/Fargate is preffered for running arbitrary Docker images.

ex:- ① Creating serverless thumbnail
② Serverless cron Job

AWS Lambda Pricing : example

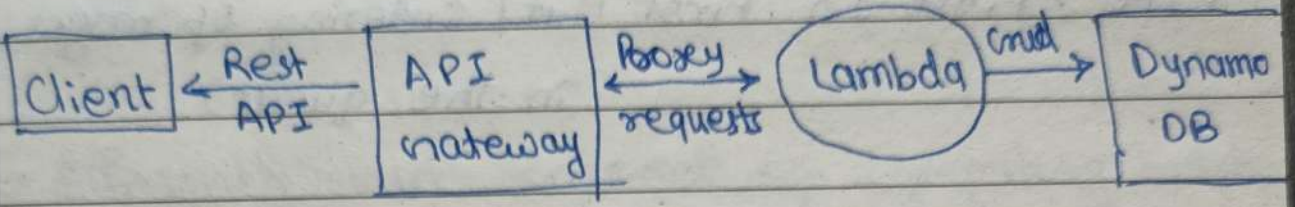
① Pay per calls :-
- first 1,000,000 requests are free
- $0.20 per 1 million request there after

② Pay per duration :-
- 4,00,000 GB - seconds of compute time per month if free.
- == 4,00,000 seconds if function is 1 GB RAM.
- == 3,200,000 seconds if function is 128 MB RAM
- After that $ 0.01 for 600,000 GB-seconds.

- It usually very cheap to run AWS Lambda so it's very popular.

Amazon API Gateway

ex:- Building a serverless API



- Fully managed service for developers to easily create, public, maintain, monitor, and secure APIs.

- Serverless & scalable

- Supports RESTFul APIs & WebSocket APIs.

- Support for security, user Authentication, API throttling, API keys, monitoring....-

AWS Batch

- Fully managed batch processing at any
- scale.

- Efficiently run 100,000s of computing batch jobs on AWS.

- A "batch" job is a job with a start & an end (opposed to continuous)

- Batch will dynamically launch EC2 instances or Spot Instances.

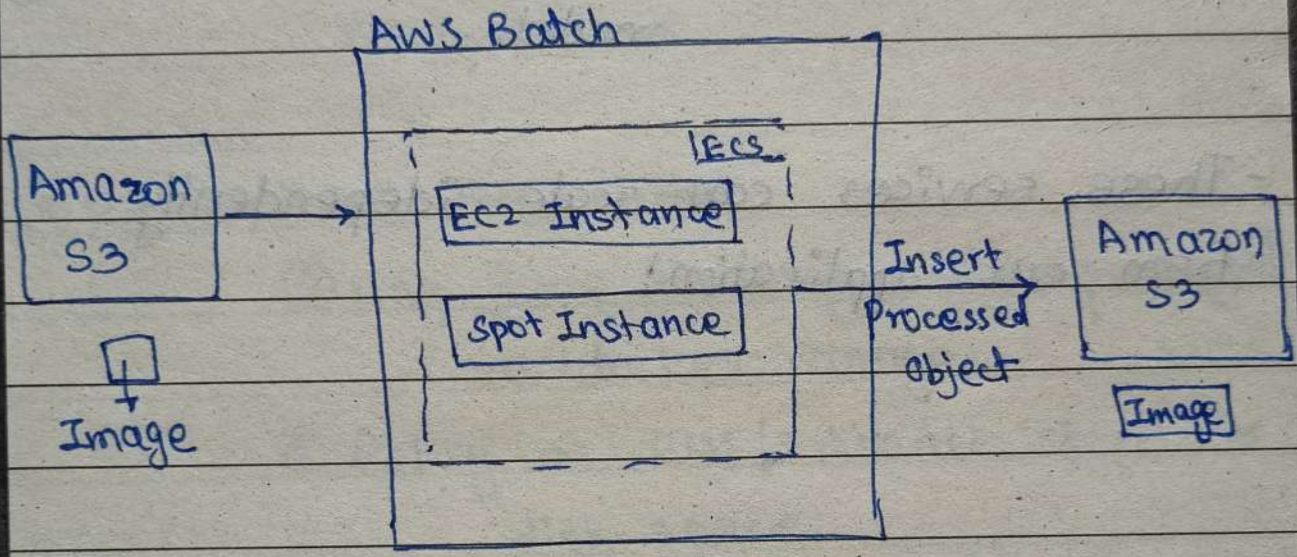
- AWS Batch provisions at right amount of compute/memory.

- You submit or schedule batch jobs & AWS Batch does the rest!

-

-Batch jobs are defined as Docker images & run on ECS.

-Helpful for cost optimizations & focusing less on the infrastructure.



Batch vs Lamba

① Lambda :→ ① Time Limit (15 min)
② Limited runtimes
③ Limited Temporary disk space
④ Serverless

② Batch :- No time limit
② any time as long as its packaged as docker image
③ Relay on EBS/instance store for disk space
④ Relies on Ec2 (can be managed by AWS).
↳ for creating new instances but this is also serverless.

Amazon Lightsail

- Virtual serverless, storage, dabases & n/w.

- Low & Predictable pricing.

- Simpler alternative to using EC2, RDS, ELB, EBS, Route 53......

- Great for people with little cloud exp.

- Can setup notifications & monitoring of your lightsail & resources.

- High availability but no auto-scaling, limited AWS integration.

* Use cases :-

1] Simple web applications (has templates for LAMP, Nginx, MEAN, Node, js....)

2] Websites (templates for wordpress, ~~Mgn~~ Magnto, Plesk, Joomla)

3] Dev/Test environment.

Other Compute - Summary

① Docker : container technology to run appl^{n}s.

② ECS :- run Docker containers on EC2 instances

③ Fargate :- Run Docker containers without provisioning the infrastructure.
- serverless offering (no EC2 instances)

④ ECR :- Private Docker Images Repository

⑤ Batch :- run batch jobs on AWS Across managed EC2 instances.

⑥ Lightsail :- Predictable & low pricing for simpl~~er~~ appl^n & DB stacks.

Lambda Summary

lamba is serverless, function as a service, seamless scaling, reactive.

- Lambda Billing :-
  1] By time run x by the RAM Provisioned
  2] By number of invocations.

- lambda supports many programming lang. except (arbitary) docker.

- Invocation time :- upto 15 minutes.

- Use Cases :- 1] Create thumbnail for images uploaded onto S3.
  2] Run a serverless cron job.

+ API Gateway :- exepose lambda functions as HTTPAPI