# Other Compute Section.

## What is Docker?

Software Development platform to deploy apps.
Apps: Packaged in containers, run on any OS.
Run the same, regardless of where they run.

1) Any machine
2) No compatibility issues.
3) Predictable behavior
4) less work
5) Easier to maintain and deploy
6) Works with any lang, os, tech.

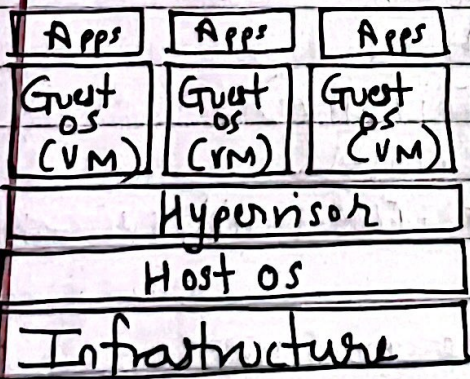
Scale containers up/down very quickly.

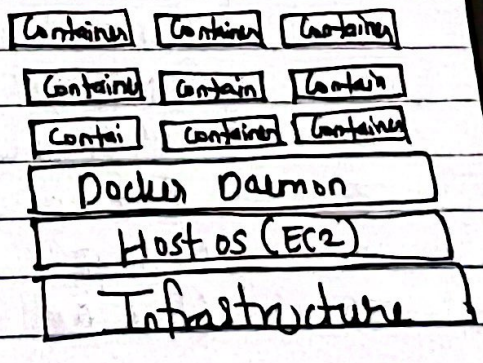
## Where Docker images are stored?

Docker images: Docker Repositories.
Private: Amazon ECR (Elastic Container Registry)

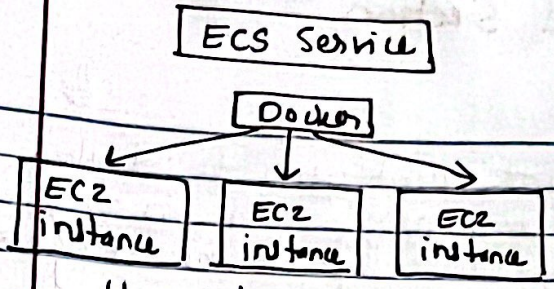
## Docker vs Virtual Machines

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



## Elastic Container Service (ECS)



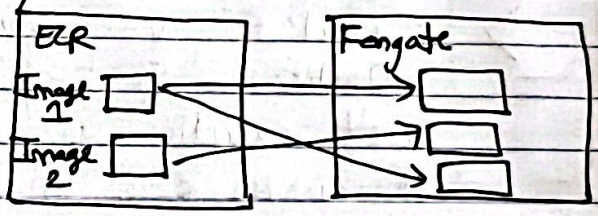

- Launch Docker containers on AWS
- You must provision & maintain the infra (EC2)
- AWS takes care of starting / stopping containers.
- Has integrations with the application LB.

## Fargate


- Launch Docker containers on AWS.
- You do not provision the infrastructure (no EC2 instances to manage)
- Serverless offerings
- AWS just runs containers for you based on the CPU/RAM you need.

## Elastic Container Registry (ECR):

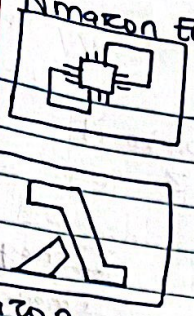
- Private docker registry.
- Store Docker images.
- Run by ECS / Fargate




## What is Serverless?

- No need to manage servers.
- Just deploy code, functions.
- Serverless ≠ FaaS (Initially)
- Pioneered by AWS Lambda
- But now also includes: Databases, Messaging, Storage.
- Does not mean there are no servers.

## Why AWS Lambda?




- Virtual servers in the cloud
- Limited by RAM and CPU
- Continuously running
- Scaling means intervention to add/remove servers.




- Virtual functions - no servers to manage
- Limited by time - short executions
- Run on-demand
- Scaling is automated

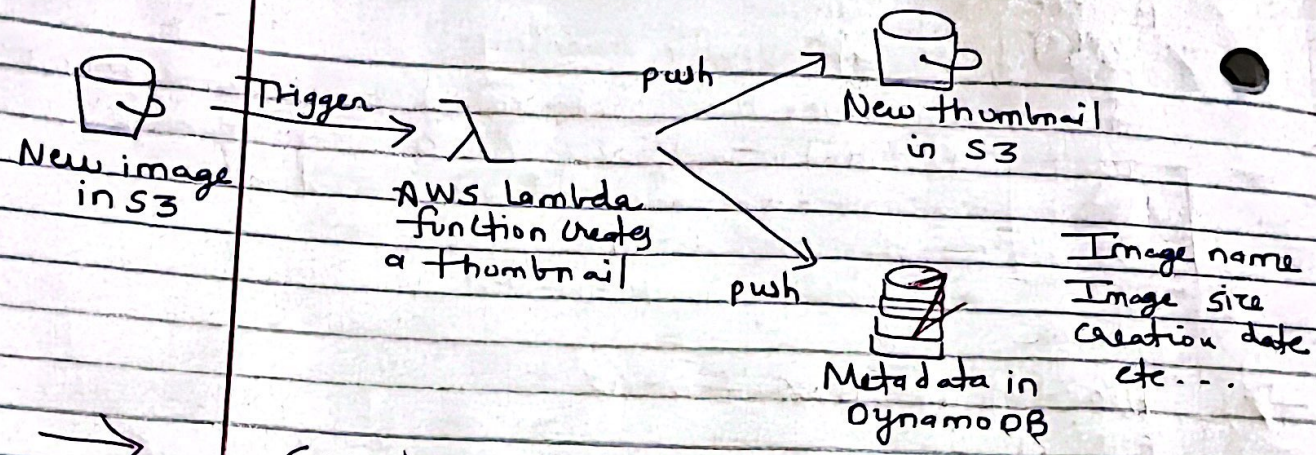
## Benefits of AWS Lambda:

- Easy Pricing:
  - Pay per request and compute time
  - Free tier: 1,000,000 AWS Lambda request, 400,000 GBs of compute time.
- Integrate with whole suite of AWS services.
- Event driven: Functions get invoked by AWS when needed.
- Integrated with many programming languages.
- Easy monitoring through cloudWatch
- More resources per function (up to 10GB of RAM)
  - ↑ RAM ⇒ ↑ CPU, Network ↑

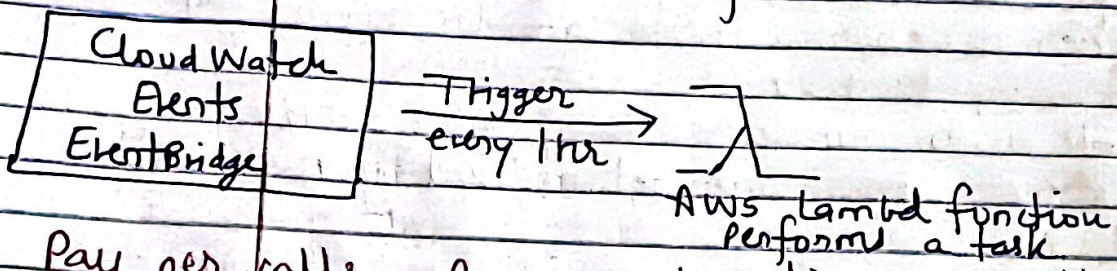
## AWS Lambda language Support.

- Node.js (Javascript)
- Python
- Java (Java 8 compatible)
- C# (.Net core)
- Golang
- C# & Powershell
- Ruby
- Custom Runtime API
- Lambda Container Image: Lambda Runtime API.

## Serverless Thumbnail Creation.

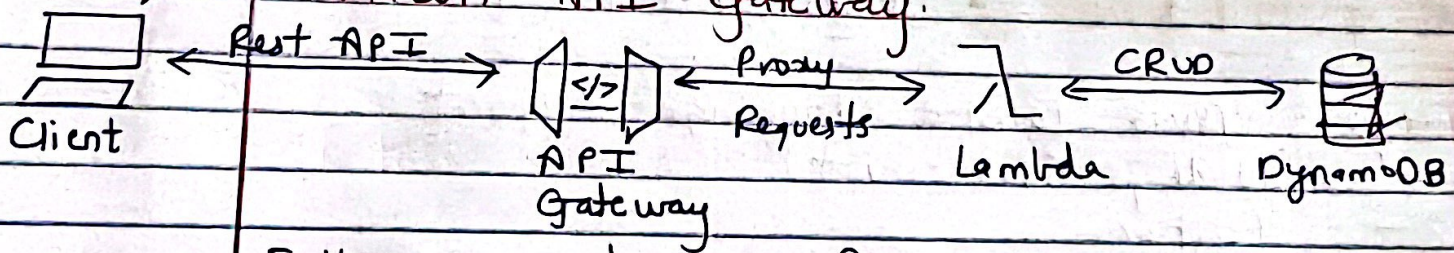



## Serverless CRON job




Pay per calls, Pay per duration, usually very cheap to run AWS Lambda.

## Amazon API Gateway.

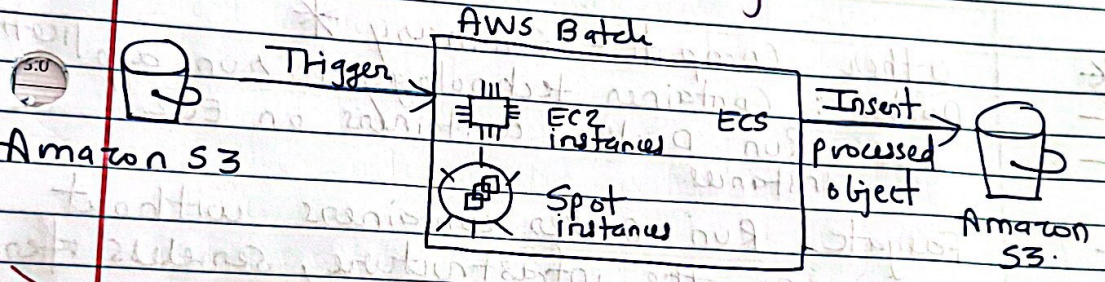



- Fully managed service for developers to easily create, publish, maintain, monitor, and secure APIs.
- Serverless and scalable.
- Supports: RESTful APIs and Websocket APIs
- Support: Security, User authentication, API throttling, API keys, monitoring ----

# AWS Batch

Fully managed batch processing at any scale.
Efficiently run 100,000s of computing batch jobs on AWS.
"Batch": Job with an start and end.
AWS Batch provisions the right amount of compute / memory.
You: Submit / Schedule batch jobs
AWS: Batch: Does the rest
Batch jobs are defined as Docker images and run on ECS
Helpful for cost optimizations and fowwing less on the infrastructure.

* AWS Batch:- Simplified eg*.




→ Batch vs Lambda.

Lambda:
1) Time limit
2) Limited runtimes
3) Limited temporary disk space
4) Serverless

Batch:
1) No time limit
2) Any runtime as long as its packaged as a Docker image
3) Rely on EBS / instance store for disk space
4) Relies on EC2 (can be managed by AWS)

## Amazon Lightsail.

- Virtual servers, storage, databases, networking.
- Low & predictable pricing
- Simpler alternative to using EC2, RDS, ELB, EBS, Route 53.
- Great for people with little cloud exp.
- Can setup notifications and monitoring of your lightsail resources.
- Use cases:
  1) Simple web applications
  2) Websites (templates for Wordpress, Magento, Plesk, Joomla)
  3) Dev / Test environment
- Has high availability but no auto-scaling, limited AWS integrations

## * Other Compute - Summary *

- Docker: Container technology to run applications
- ECS: Run Docker containers on EC2 instances.
- Fargate: Run Docker containers without provisioning the infrastructure, serverless offering (no EC2 instances)
- ECR: Private Docker Images Repository.
- Batch: Run batch jobs on AWS across managed EC2 instances.
- Lightsail: Predictable low pricing for simple application & DB stacks.

## * Lambda Summary *

- Serverless, FaaS, seamless scaling, reactive
- Lambda Biling: By the time run x by the RAM provisioned, By the no. of invocations.
- Language support - Many programming languages except (arbitrary) Docker.
- Invocation time - Up to 15 mins.

Use cases: Create Thumbnails for images uploaded onto S3
- Run a serverless cron job

API Gateway: Expose lambda functions as HTTP API