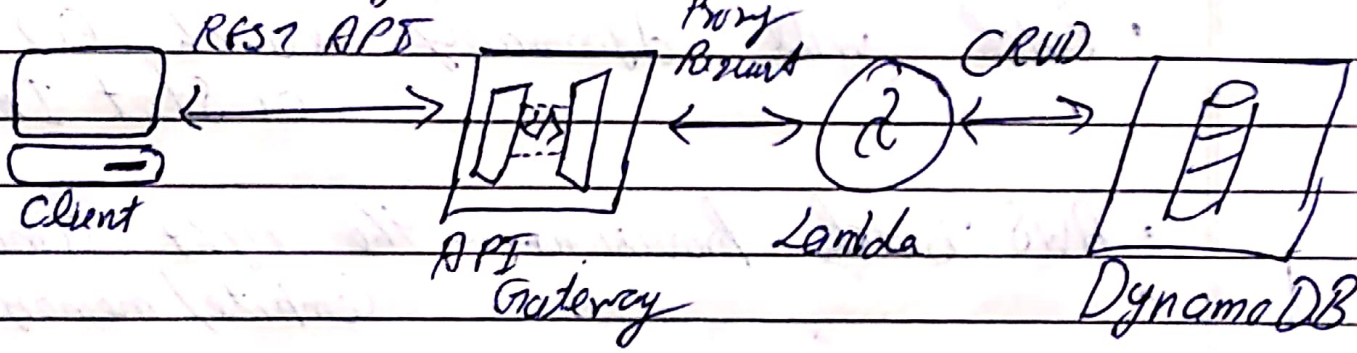
- It is usually very cheap to run AWS Lambda so its very popular

Day 17/100 → Date: 20/12/21

Amazon API Gateway

- Example: building a serverless API



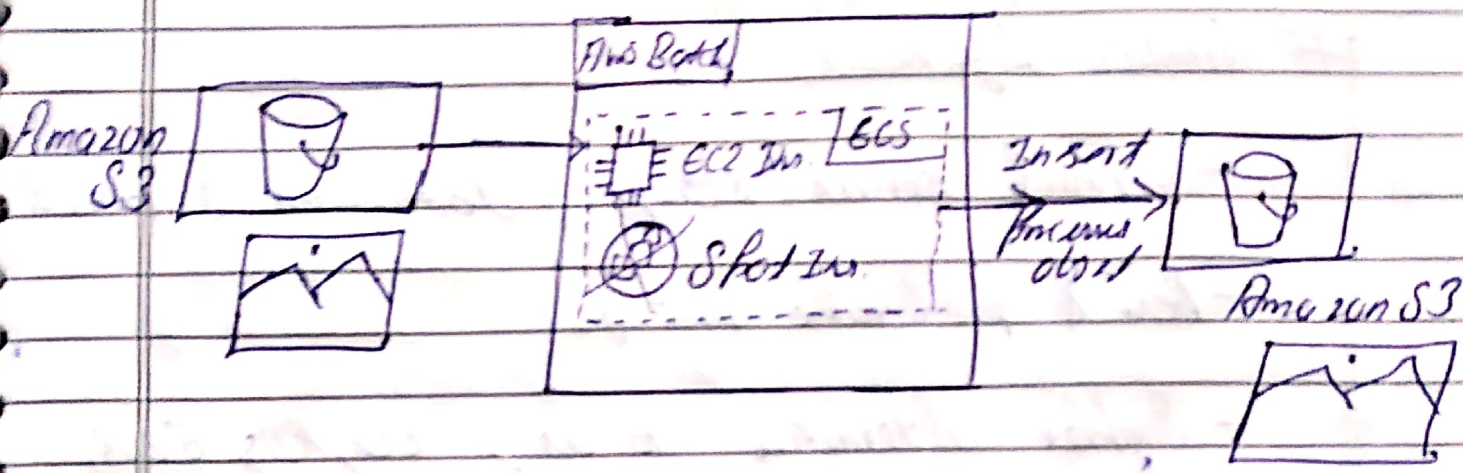
- Fully managed service for developers to easily create, publish, maintain, monitor and secure APIs.
- Serverless and scalable.
- Supports RESTful APIs and WebSocket APIs.
- Support for security, user authentication, API throttling, API keys, monitoring.

DATE 20/12/21

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 and and end (opposed to continuous)
- Batch will dynamically launch EC2 instance or Spot Instance.
- AWS Batch provisions the right amount of compute/memory.
- You submit or schedule batch jobs & AWS Batch does the rest!
- Batch jobs are defined as Docker images and run on ECS
- Helpful for cost optimizations and focusing on the infrastructure

DATE 20/12/21



Batch vs Lambda

- Lambda:
  - Time limit
  - Limited runtimes
  - Limited temporary disk space
  - Serverless

- Batch:
  - NO time limit
  - Any runtime as long as it's packaged as Docker image
  - Rely on EBS/instance store for disk space
  - Relies on EC2 (can be managed by AWS)

DATE 20/12/21

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 experience
- Can setup notifications and monitoring of your Lightsail resources.
- Use Cases:
  - Simple web applications (has templates for LAMP, Nginx, MEAN, Node.js)
  - Websites (templates for WordPress, Magento, Plesk, Joomla)
  - Dev/test environment
- Has high available but ~~not~~ no auto-scaling
  limited AWS integration

DATE 20/12/21

Deploying And Managing Infrastructure at Scale Section

CloudFormation:

- CloudFormation is declarative way of outlining your AWS Infrastructure, for any resources (most of them are supported)
- For example, with in CloudFormation template, you say:
  - I want a security group
  - I want two EC2 instances using this security group.
  - I want an S3 bucket
  - I want a load balancer (ELB) in front of these machines.
- Then CloudFormation creates those for you, in the right order, with the exact Configuration that you specify.

Benefits of AWS CloudFormation (1/2)

- Infrastructure as Code
  - No resources are manually created, which is excellent for Control
  - Changes to the infrastructure are reviewed through Code.
- Cost
  - Each resource within the stack is tagged with an identifier so you can easily see how much a stack costs you

DATE 20/12/21

- You can estimate the costs of your resources using the CloudFormation template

- Saving Strategy: In dev, you could automation deletion of templates at 5PM and recreated at 8AM safely

Benefits of AWS CloudFormation (2/2)

- Productivity
  - Ability to destroy and re-create an infrastructure on the cloud on the fly
  - Automated generation of Diagram for your templates!
  - Declarative programming (no need to figure out ordering and orchestration)

- Don't re-invent the wheel
  - Leverage existing templates on the web.
  - Leverage the documentation

- Supports (almost) all AWS resources!
  - Everything we'll see in this course is supported
  - You can use "Custom resources" for resource that are not supported

AWS Cloud Development Kit (CDK)

- Define your Cloud Infrastructure using a familiar language!
  - JavaScript/TypeScript, Python, Java and .NET
- The Code is "Compiled" into a cloudformation template (JSON/YAML)
- You can therefore deploy infrastructure and application runtime code together
  - Great for lambda functions
  - Great for Docker Containers in ECS/EKS

BeanStalk

- Elastic BeanStalk is developer centric view of deploying an application on AWS.
- It uses all the components we've seen before EC2, ASG, ELB, RDS etc.
- But it's all in one view that's easy to make sense of
- We still have full control over the configuration

EKS

DATE 20/12/21

- Beanstalk = Platform as a Service (PaaS)
- Beanstalk is free but you pay for the underlying instances.

Elastic Beanstalk

- Managed Service
  - Instance Configuration/OS is handled by Beanstalk
  - Deployment Strategy is configurable but performed by Elastic Beanstalk
  - Capacity provisioning
  - Load balancing & auto-scaling
  - Application Health-monitoring & responsiveness
- Just the application code is the responsibility of the developer
- Three [3] architecture models:
  - Single Instance deployment: good for dev
  - LB + ASG: great for production or pre-production web application
  - ASG only: great for non-web apps in production (worker etc).

- Support for many platforms:
  - GO
  - Java SE
  - Java with Tomcat
  - .NET on Windows Server with IIS
  - Node.js
  - PHP
  - Python
  - Ruby
  - Packer Builder
  - Single Container Docker
  - Multi-Container Docker
  - Preconfigured Docker

- If not supported, you can write your Custom platform (advanced)

- Elastic Beanstalk - Health Monitoring
  - Health agent pushes metrics to CloudWatch
  - Checks for app health, publishes health events