Other Compute

What is Docker?

- Docker is a software development platform to deploy apps
- Apps are packaged 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 behavior
  - Less work
  - Easier to maintain and deploy
  - Works with any language, any OS, any technology
- Scale containers up and down very quickly (seconds)

Where are Docker images stored?

- **Docker Hub**: Centralized public repository for storing Docker images.
- Public: Docker Hub https://hub.docker.com/
- Private: **Amazon ECR (Elastic Container Registry)**: AWS service for storing, managing, and deploying container images.

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

Docker Containers	Virtual Machines (VMs)
Lightweight, shares the host OS kernel	Heavier, includes full OS
Starts in seconds	Slower startup (minutes)
Portable, fast scaling	Not as portable, more resource-intensive
Best for microservices & modern apps	Best for running multiple OS environments

ECS (Elastic Container Service)

- Fully managed container orchestration service.
- Supports Docker containers.
- Launch Docker containers on AWS
- AWS takes care of starting / stopping containers
- **Two launch modes**: **EC2** (self-managed instances) and **Fargate** (serverless).
- Provides integration with IAM, VPC, ELB, and ECR.

Fargate

- Serverless compute engine for containers, works with ECS and EKS.
- No need to manage EC2 instances.
- Pay for resources used (vCPU and memory).
- AWS just runs containers for you based on the CPU / RAM you need

ECR (Elastic Container Registry)

- Fully managed Docker container registry.
- Stores, manages, and secures Docker images.
- Integrated with **ECS**, **EKS**, and **Fargate** for easy deployment.
- This is where you store your Docker images so they can be run by ECS or Fargate

What's Serverless?

- No need to provision, scale, or manage servers.
- Resources are automatically provisioned and scaled by AWS.
- Serverless is a new paradigm in which the developers don't have to manage servers anymore…
- They just deploy code
- They just deploy… functions !
- Initially… 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
- Ideal for event-driven and stateless applications.

Why AWS Lambda?

- Serverless compute service to run code without managing infrastructure.
- Executes code in response to events (e.g., API calls, file uploads).
- Scales automatically and you only pay for usage.

EC2	Lambda
Virtual Servers in the Cloud	Virtual functions – no servers to manage!
Limited by RAM and CPU	Limited by time - short executions
Continuously running	Run on-demand
Scaling means intervention to add / remove servers	Scaling is automated!

Benefits of AWS Lambda

- **No server management**: AWS handles the infrastructure.
- **Automatic scaling**: Scales based on event triggers.
- **Flexible scaling**: Runs from a few requests per day to thousands per second.
- **Event-driven architecture**: Ideal for apps that need to respond to events.
- Easy Pricing:
  - Pay per request and compute time
  - Free tier of 1,000,000 AWS Lambda requests and 400,000 GBs of compute time
- Integrated with the whole AWS suite of services
- Event-Driven: functions get invoked by AWS when needed
- Integrated with many programming languages
- Easy monitoring through AWS CloudWatch
- Easy to get more resources per functions (up to 10GB of RAM!)
- Increasing RAM will also improve CPU and network!

AWS Lambda Language Support

- Node.js
- Python
- Ruby
- Java
- Go
- .NET Core
- custom runtime (via container images) (community supported, example Rust)
- Lambda Container Image
  - The container image must implement the Lambda Runtime API
  - ECS / Fargate is preferred for running arbitrary Docker images

AWS Lambda Pricing: Example

- Based on number of requests and execution time.
- You can find overall pricing information here: https://aws.amazon.com/lambda/pricing/
- First **1 million requests/month** are free.
- After that, **$0.20 per million requests**.
- **Execution duration**: $0.00001667 for every GB-second used (first 400,000 GB-seconds free per month).
-
  - Pay per duration: (in increment of 1 ms)
  - 400,000 GB-seconds of compute time per month for FREE
  - == 400,000 seconds if function is 1GB RAM
  - == 3,200,000 seconds if function is 128 MB RAM
  - After that $1.00 for 600,000 GB-seconds
- It is usually **very cheap** to run AWS Lambda so it's **very popular**

Amazon API Gateway

- Managed service for creating, publishing, and monitoring REST, HTTP, and WebSocket APIs.

- Integrates with AWS Lambda for fully serverless APIs.
- Serverless and scalable
- Support for security, user authentication, API throttling, API keys, monitoring.
- **Throttling**, **caching**, and **authorization** features built-in.

AWS Batch

- Fully managed service for running batch processing workloads.
- Dynamically provisions compute resources based on job requirements.
- Suitable for large-scale data processing, such as machine learning and rendering tasks.
- Efficiently run 100,000s of computing batch jobs on AWS
- A "batch" job is a job with a start and an end (opposed to continuous)
- Batch will dynamically launch EC2 instances or Spot Instances
- AWS Batch provisions the right amount of compute / memory
- You submit or schedule batch jobs and AWS Batch does the rest!
- Batch jobs are defined as Docker images and run on ECS
- Helpful for cost optimizations and focusing less on the infrastructure

Batch vs Lambda

AWS Batch	**AWS Lambda**
Designed for **batch processing**	Designed for **event-driven** architecture
Handles large-scale compute jobs	Executes short-lived functions
Custom EC2 instances or Fargate tasks	Fully serverless, no server management
Jobs may take minutes to hours	Max execution time of 15 minutes
Rely on EBS / instance store for disk space	Limited temporary disk space

Amazon Lightsail

- Virtual servers, storage, databases, and 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 availability but no auto-scaling, limited AWS integrations

Lambda Summary

- Lambda is Serverless, Function as a Service, seamless scaling, reactive
- Lambda Billing:
  - By the time run x by the RAM provisioned
  - By the number of invocations
- Language Support: many programming languages except (arbitrary) Docker
- Invocation time: up to 15 minutes
- Use cases:
  - Create Thumbnails for images uploaded onto S3
  - Run a Serverless cron job
- API Gateway: expose Lambda functions as HTTP API

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