Day - 08 | Other Compute Services | AWS Cloud Practitioner Certification CLF-C02

Created on 2024-07-29 13:31
Published on 2024-08-03 08:32

Other Compute

- What is Docker?

➺ Where Docker images are stored?

➺ Docker versus Virtual Machines

- ECS

- Fargate

- ECR

- What’s serverless?

- Why AWS Lambda ?

➺ Benefits of AWS Lambda

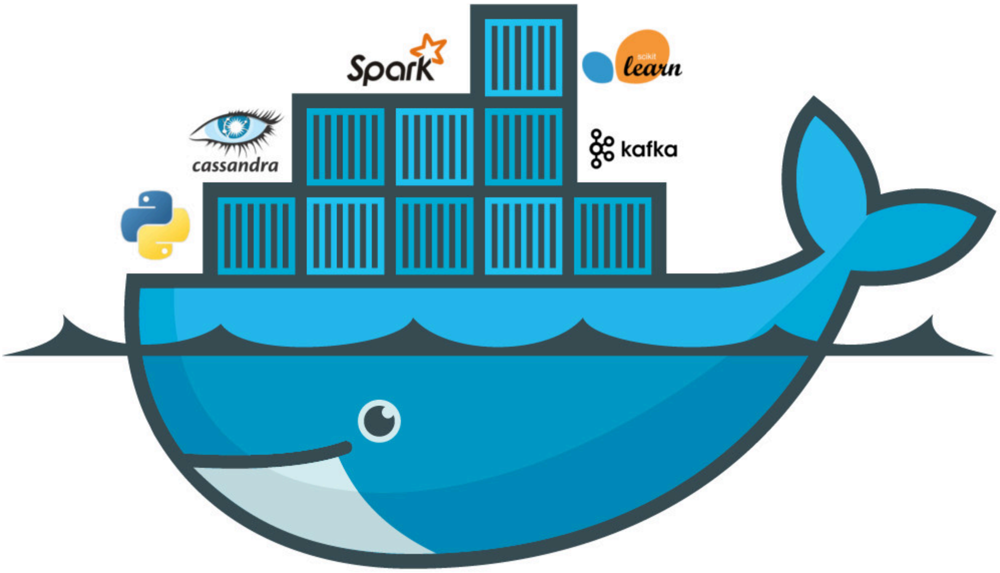
➺ AWS Lambda language support

➺ AWS Lambda Pricing: example

- Amazon API Gateway
- AWS Batch
- Batch vs Lambda
- Amazon Lightsail
- Lambda Summary
- Other Compute Summary

What is Docker?

Docker is an open platform for developing, shipping, and running applications. It enables you to separate your applications from your infrastructure, ensuring that your software runs consistently across different environments. Docker uses containerization to package an application and its dependencies into a single, portable container that can run on any system with Docker installed. These containers can be easily scaled up and down very quickly within seconds.



Docker

Where Docker images are stored?

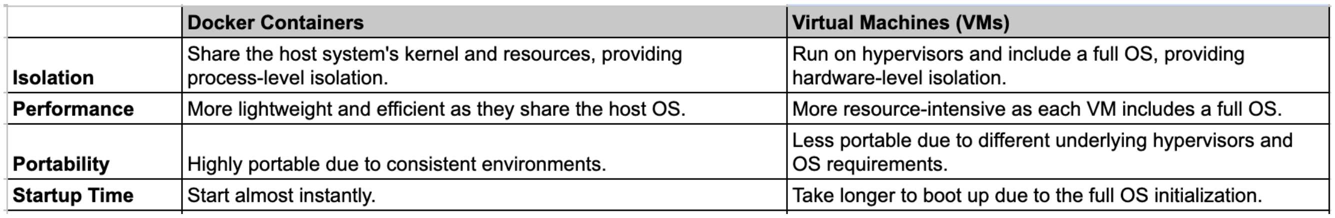
Docker images are stored in repositories. These repositories can be:

- **Docker Hub:** The default public registry where users can store and share Docker images.

- **Private Registries:** Hosted by organizations to securely store and manage Docker images internally. Examples include:

➻ **Elastic Container Registry (ECR):** A fully managed Docker container registry on AWS.

Docker versus Virtual Machines



	Docker Containers	Virtual Machines (VMs)
Isolation	Share the host system's kernel and resources, providing process-level isolation.	Run on hypervisors and include a full OS, providing hardware-level isolation.
Performance	More lightweight and efficient as they share the host OS.	More resource-intensive as each VM includes a full OS.
Portability	Highly portable due to consistent environments.	Less portable due to different underlying hypervisors and OS requirements.
Startup Time	Start almost instantly.	Take longer to boot up due to the full OS initialization.

Docker versus Virtual Machines

ECS

Amazon Elastic Container Service (ECS) is a fully managed container orchestration service. ECS allows you to run and manage Docker containers on a cluster of EC2 instances. AWS takes care of starting / stopping containers. It has integrations with the Application Load Balancer.

Key features include:

- **Task Definitions:** Define how Docker containers should be deployed.

- **Service Definitions:** Maintain and scale applications.

- **Cluster Management:** Automatic scaling and management of EC2 instances in the cluster.

Fargate

AWS Fargate is a serverless compute engine for containers that works with ECS and EKS (Elastic Kubernetes Service). Fargate eliminates the need to manage servers or clusters. AWS just runs containers for you based on the CPU / RAM you need.

Key benefits include:

- **Serverless:** No infrastructure management.

- **Scalability:** Automatically scales based on workload.

- **Resource Efficiency:** Pay only for the resources you use.

ECR

Amazon Elastic Container Registry (ECR) is a fully managed Docker container registry that makes it easy to store, manage, and deploy Docker container images. This is where you store your Docker images so they can be run by ECS or Fargate.

It integrates seamlessly with ECS and Fargate. Key features include:

- **Security:** Fine-grained access control using IAM.

- **Reliability:** Highly available and durable.

- **Integration:** Supports Docker CLI and native integration with AWS services.

What’s serverless?

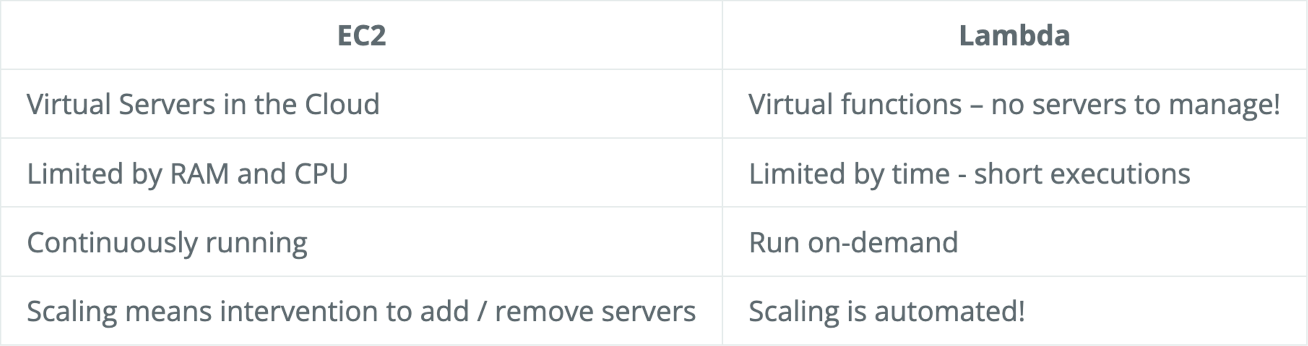
Serverless computing allows you to build and run applications and services without managing infrastructure. In a serverless architecture, the cloud provider dynamically manages the allocation and provisioning of servers. Serverless was pioneered by AWS Lambda but now also includes anything that's managed: "databases, messaging, storage, etc."

Key characteristics include:

- **Automatic Scaling:** Scales automatically with demand.
- **No Server Management:** Developers focus on code, not infrastructure.
- **Pay-per-Use:** Costs are based on actual usage.

Serverless does not mean there are no servers. It means you just don't manage / provision / see them.

Why AWS Lambda ?



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!

EC2 vs Lambda

Benefits of AWS Lambda

- **Easy Pricing:** (a) Pay per request and compute time (b) 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 (JavaScript)

- Python

- Java (Java 8 compatible)

- C# (.NET Core)

- Golang

- C# / Powershell

- Ruby

- Custom Runtime API (community supported, example Rust) for other languages

- Lambda Container Image -> The container image must implement the Lambda Runtime. APIECS / Fargate is preferred for running arbitrary Docker images.

AWS Lambda Pricing: example

- You can find overall pricing information here: https://aws.amazon.com/lambda/pricing/

- Pay per calls:

➺ First 1,000,000 requests are free

➺ $0.20 per 1 million requests thereafter ($0.0000002 per request)

- 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

Amazon API Gateway is a fully managed service for creating, publishing, maintaining, monitoring, and securing APIs at any scale.

Example: building a serverless API

Key features include:

- **Support for RESTful and WebSocket APIs**

- **Integration with Lambda, EC2, and other AWS services**

- **Throttling, security, and monitoring capabilities**

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 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

- **Use Case:**

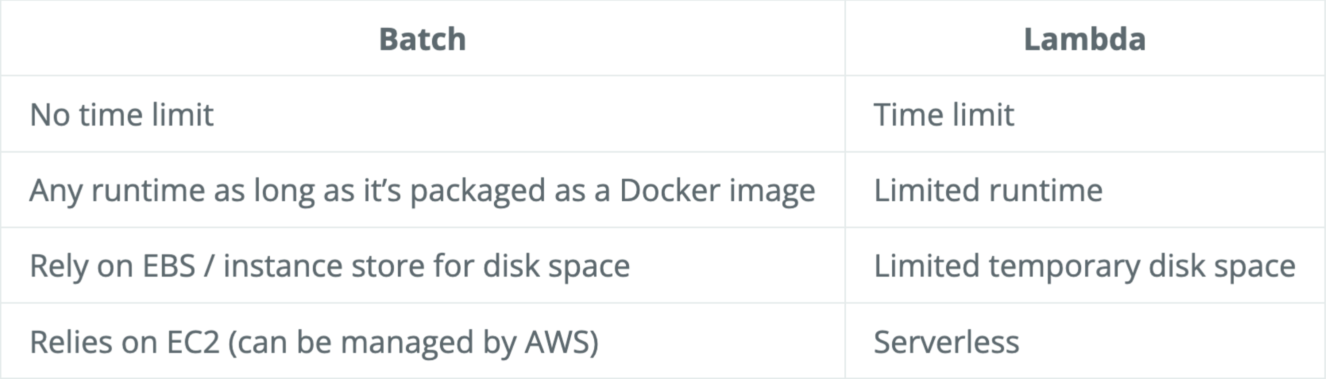
➻ **Batch:** Suitable for long-running, large-scale batch processing jobs.

➺ **Lambda:** Ideal for short-duration, event-driven functions.

- **Execution Time:**

➻ **Batch:** Can run for hours or days.

➺ **Lambda:** Limited to 15 minutes per execution.



Batch	Lambda
No time limit	Time limit
Any runtime as long as it's packaged as a Docker image	Limited runtime
Rely on EBS / instance store for disk space	Limited temporary disk space
Relies on EC2 (can be managed by AWS)	Serverless

Batch vs Lambda

Amazon Lightsail

Amazon Lightsail is a simplified cloud platform that provides easy-to-use cloud resources for smaller workloads. It offers:

- **Pre-configured instances:** For web applications, websites, and small databases.
- **Fixed pricing:** Simple, low & predictable pricing for compute, storage, and networking resources.
- **Ease of use:** User-friendly interface and fast setup.
- Virtual servers, storage, databases, and networking
- 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

AWS offers a wide range of compute services to meet various needs, from container orchestration (ECS, Fargate) and serverless computing (Lambda) to simplified cloud resources (Lightsail). These services provide flexibility, scalability, and cost-efficiency, allowing developers to focus on building and running applications without managing underlying infrastructure.

- 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

---

Happy Learning !