Cloud Computing Introduction

CS516 – Cloud Computing

Computer Science Department

Maharishi International University

Maharishi International University - Fairfield, Iowa

Main concepts

- Cloud computing
- Cloud services models
- Core AWS services

What is Cloud Computing?

Cloud computing is the on-demand delivery of the all types of resources as a web service such as compute, database, big data, AI, VR, IoT, block chain, quantum technologies, robotics, satellite, you name it.

Think of the cloud as software that helps you build your infrastructure and app components. You will use cloud services rather than developing them yourself.

Compute
- EC2
- Lightsail
- Lambda
- Batch
- Elastic Beanstalk
- Serverless Application Repository
- AWS Outposts
- EC2 Image Builder
- AWS App Runner

Containers
- Elastic Container Registry
- Elastic Container Service
- Elastic Kubernetes Service
- Red Hat OpenShift Service on AWS

Storage
- S3
- EFS
- FSx
- S3 Glacier
- Storage Gateway
- AWS Backup

Customer Enablement
- AWS IQ
- Support
- Managed Services
- Activate for Startups

Robotics
- AWS RoboMaker

Blockchain
- Amazon Managed Blockchain

Satellite
- Ground Station

Quantum Technologies
- Amazon Braket

Management & Governance
- AWS Organizations
- CloudWatch
- AWS Auto Scaling
- CloudFormation
- CloudTrail

Machine Learning
- Amazon SageMaker
- Amazon Augmented AI
- Amazon CodeGuru
- Amazon DevOps Guru
- Amazon Comprehend
- Amazon Forecast
- Amazon Fraud Detector
- Amazon Kendra
- Amazon Lex
- Amazon Personalize
- Amazon Polly
- Amazon Rekognition
- Amazon Textract
- Amazon Transcribe
- Amazon Translate
- AWS DeepComposer
- AWS DeepLens
- AWS DeepRacer
- AWS Panorama
- Amazon Monitron
- Amazon HealthLake
- Amazon Lookout for Vision
- Amazon Lookout for Equipment

AWS Cost Management
- AWS Cost Explorer
- AWS Budgets
- AWS Marketplace Subscriptions
- AWS Application Cost Profiler

Front-end Web & Mobile
- AWS Amplify
- Mobile Hub
- AWS AppSync
- Device Farm
- Amazon Location Service

AR & VR
- Amazon Sumerian

Application Integration
- Step Functions
- Amazon AppFlow
- Amazon EventBridge
- Amazon MQ
- Simple Notification Service
- Simple Queue Service
- SWF

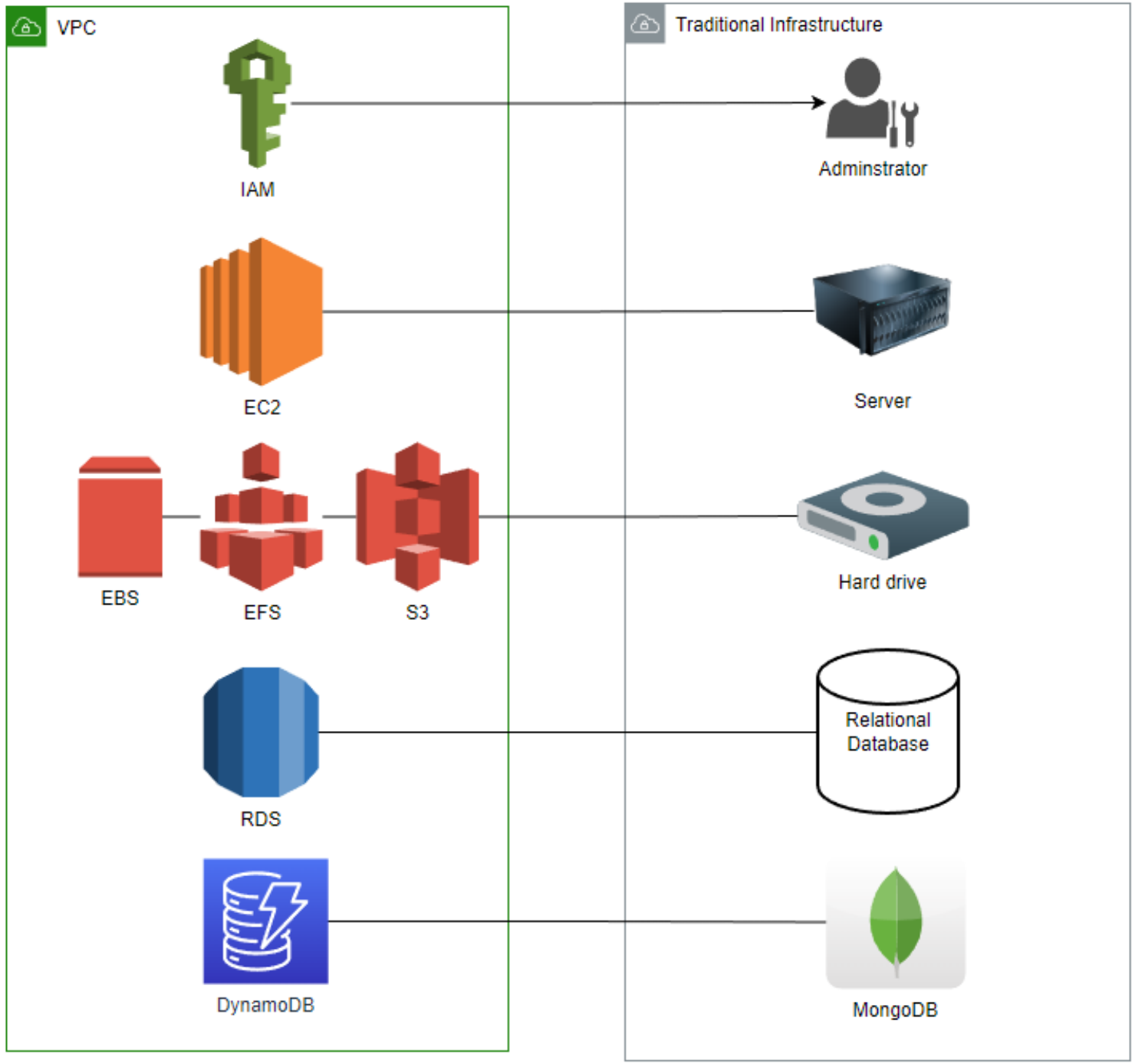
What are services?

Amazon Web Services (AWS) is the biggest cloud provider.

Web services are just HTTP endpoints (RESTful or SOAP). We can call and use AWS services in 3 ways.

1. AWS console – Calling AWS services from the web app. It is just a front-end app that is built on top of the Amazon's web services.
2. CLI - Calling the same AWS services from terminal. You need to install it and provide tokens. Great for quick experiments.
3. SDK. – Calling the same AWS services from your application. For example, storing data in the database.

Cloud is not very different from traditional infrastructure.



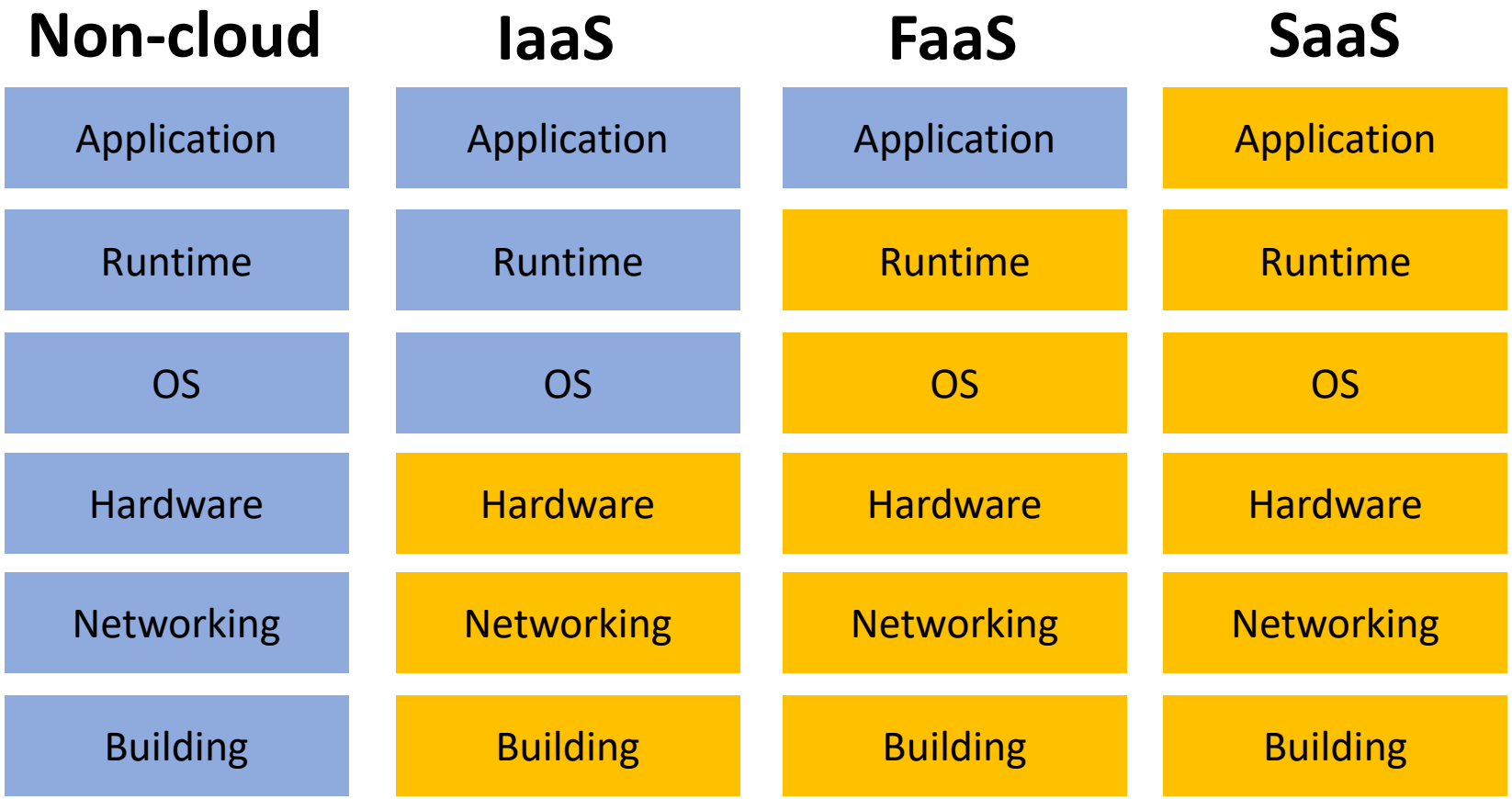
Benefits of the cloud

- **Do less and achieve more** – The cloud provider deals with technical problems. It let’s developers to focus on the application.
- **Cost-effective** – The cloud is cost-effective in most cases, especially serverless services.
- **Secure** – There are many security services that you can stack on top of your applications that protect against attacks at all layers.
- **Reliable** – Because the app runs and data is stored in multiple data centers, even multiple regions. That improves high availability and fault tolerance of the application and durability of the data. The cloud also helps your app to scale.

Benefits of the cloud

- **Performant** – The globe is in your hand with the cloud. You can serve users all over the world without losing performance. There are many services in the cloud that improve the performance of the application.
- **Agility** – Agility is crucial in business that gives advantages. You can deploy your application in multiple regions globally in minutes. There are also tools like Amazon Amplify that helps developers to build full-stack web and mobile applications in minutes.
- **You don’t have to guess capacity** – In the traditional infrastructure, you have to guess the server size that satisfies the need. But that could be too much or too low. If the server is too big, it will cost more. If the server is too small, the application goes down or gets slower due to full utilization. Cloud resources are elastic.
- **Built-in metrics** – Metrics are created along with the resource in the cloud. Metrics are useful information about the resource for monitoring and troubleshooting purpose. For example, when you create EC2 virtual machines in the AWS cloud, CPU utilization metrics are also created in the CloudWatch.

Models of Cloud Services



Infrastructure as a Service (IaaS)

IaaS means you rent a server from the cloud provider. You choose the operating system, memory, hard drive, and CPU size. You will receive a key pair to log in to your server after the instance is created. Once the server is provisioned, you can do whatever you want with the server such as hosting a website you developed.

You still have a lot of work to do on your side. I recommended you use other cloud service models if you want to do less and don’t need much OS-level configurations.

Platform as a Service (PaaS)

PaaS is where you don't know what resources you need but you've just got your code and you will then use Platform as a Service to go in and provision those resources for you.

You still have to look after the underlying assets but you don't have to worry about the provisioning of it.

Function as a Service (FaaS)

FaaS is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage application functionalities **without the complexity of building and maintaining the infrastructure**.

Building an application following this model is one way of achieving a **serverless** architecture and is typically used when building event-driven and microservices applications.

Serverless

Serverless computing is a cloud computing execution model in which the cloud provider allocates machine resources on demand, taking care of the servers on behalf of their customers.

When an app is not in use, there are no computing resources allocated to the app. Pricing is based on the actual amount of resources consumed by an application.

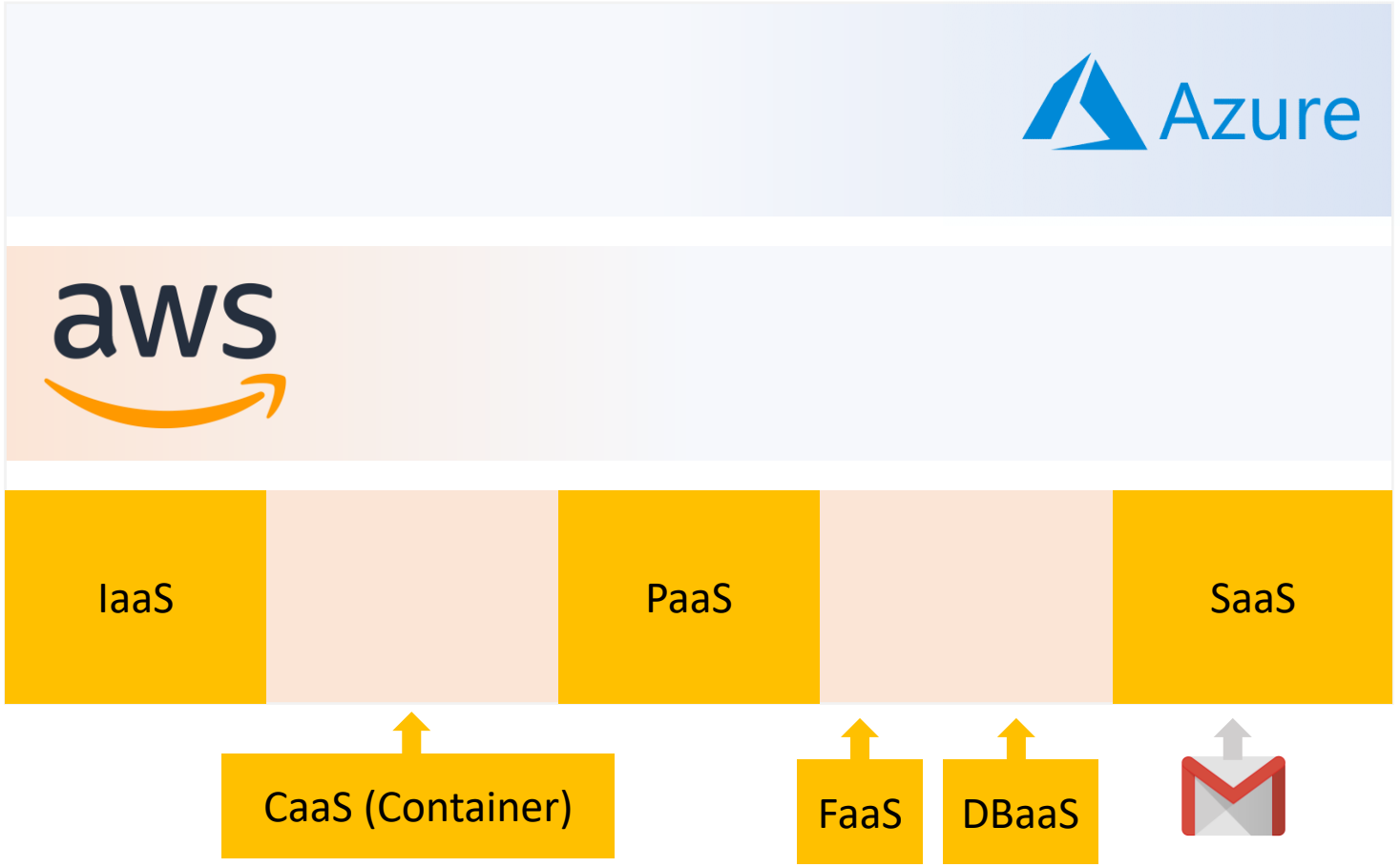
Software as a Service (SaaS)

SaaS is the app you develop and costumers use it via the internet. For example, Gmail, all you worry about is using the actual software, about creating messages, filtering spam filters. You're not worried about the underlying servers, how they are load balanced, high availability, DNS resolving etc.

SaaS apps are delivered by the IaaS, PaaS, and/or FaaS.

As a Service!

- **Container** as a service (CaaS)
- **Data** as a service
- **Desktop** as a service
- **Function** as a service
- **Infrastructure** as a service
- **Integration** as a service
- **Network** as a service
- **Platform** as a service
- **Security** as a service
- **Software** as a service



Container as a Service

CaaS allows you to run containerized applications in the cloud. Containerized deployments took over deployments on virtual machines. Because it is much lighter and faster to deploy apps.

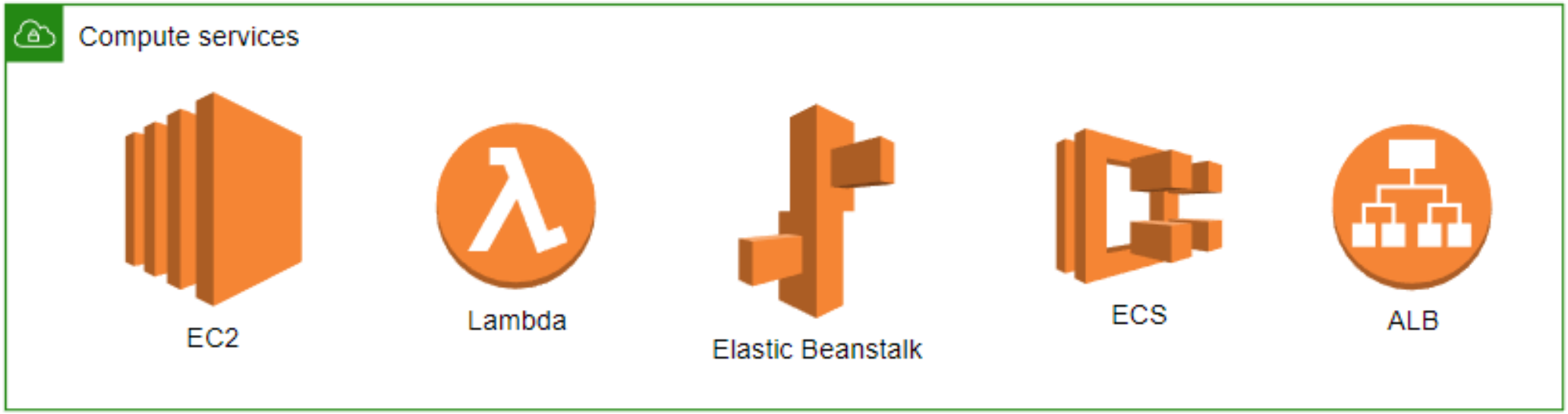
Containerized applications are platform-agnostic.

Docker is the most popular containerization technology. In AWS, there 2 ways to run containerized applications, on servers (ECS on EC2 or EKS) or serverless (ECS Fargate).

Compute services

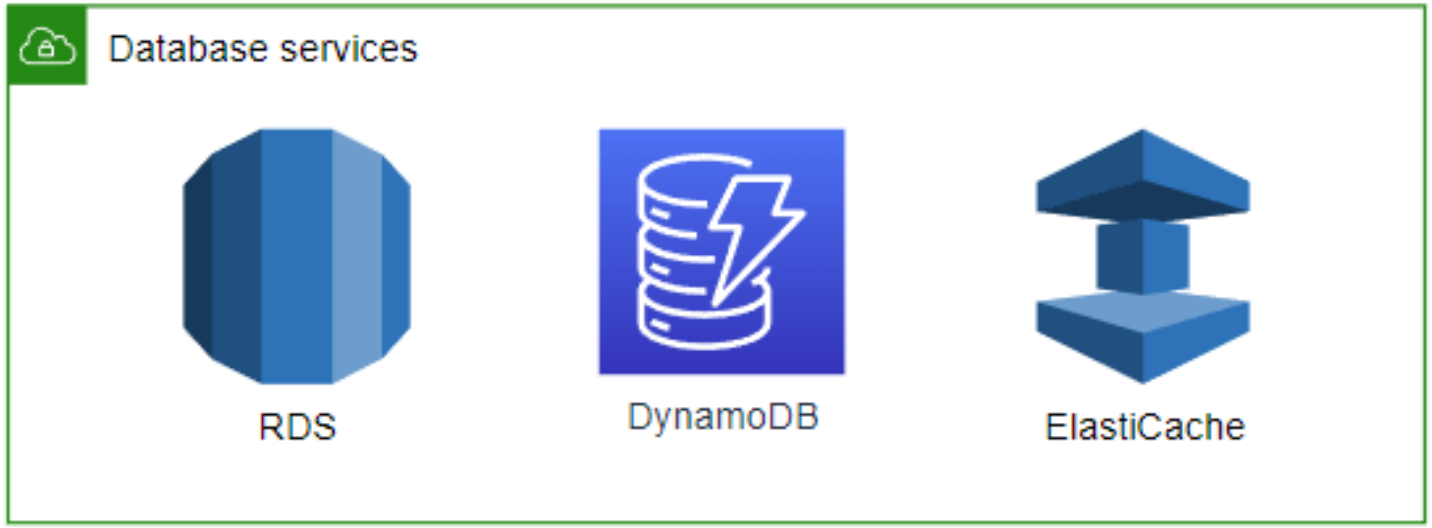
- **EC2** (Elastic Compute Cloud – IaaS model) – Virtual servers in the cloud.
- **Lambda** (FaaS model) – Run code without thinking of servers.
- **Elastic Beanstalk** (PaaS model) – Run and manage web applications.
- **ECS** (Elastic Compute Cloud - CaaS) – Run container applications.
- **EKS** (Elastic Kubernetes Service) – Run Kubernetes applications. The technology used in your application stack doesn’t have to be cloud-native technology. Cloud providers also give you the option to run other popular technologies in the cloud such as Kubernetes, Kafka, ActiveMQ, and MongoDB.

Compute services



Database services

- **RDS** (Relational Database Service) – Relational database.
- **DynamoDB** – Hyper scaling NoSQL database fully managed by AWS.
- **ElastiCache** (Redis and MemCache) – In memory cache that you can put in front of the RDS or use it as a database.
- **DocumentDB** – MongoDB in the AWS cloud.



Networking and content delivery services

- **VPC** (Virtual Private Cloud) – Isolated cloud network. Similar to a private network for an organization or home.
- **Route53** – Scalable DNS and domain name registration.
- **Global Accelerator** – It improves the application's availability and performance using the AWS Global Network.
- **Direct Connect** – It connects the on-premises data center with the AWS cloud.
- **CloudFront** – Global content delivery network.
- **API Gateway** – Helps build, deploy, and manage APIs.

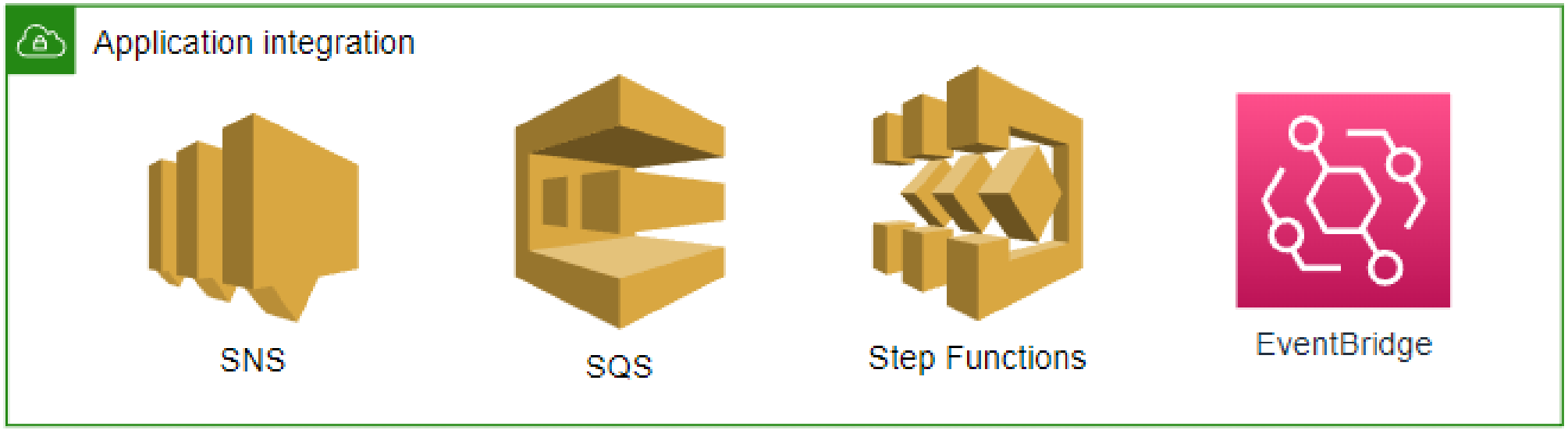
Networking and content delivery services

Application integration services

Application integration services are important. They make applications asynchronous. So latency is decreased drastically, and throughput is increased.

- **SNS** (Simple Notification Service) – Send messages to applications or people.
- **SQS** (Simple Queue Service) – Used to decouple applications. A message is sent to a queue from an application. Another app picks app the messages in the queue.
- **Step Functions** – Coordinates distributed applications.
- **EventBridge** – Serverless event bus that connects application data from other apps and AWS services.

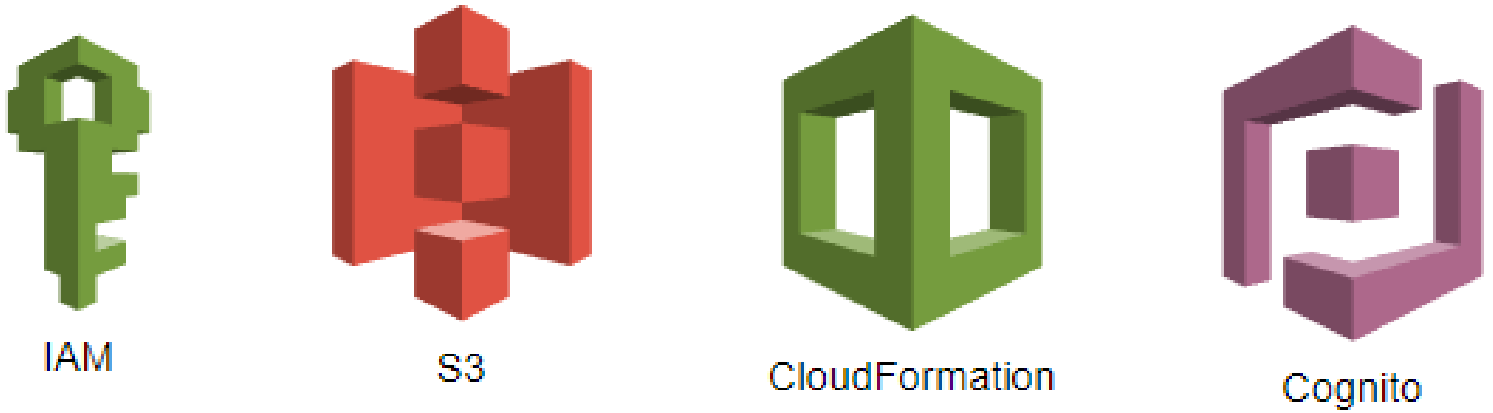
Application integration services



Other important services

- **IAM** (Identity and Access Management) – Access management to AWS.
- **S3** (Simple Storage Service) – Scalable object storage service.
- **CloudFormation** and CDK (Cloud Development Kit) – It automates cloud resource creation and management.
- **Cognito** – Application user management

There are many other types of services you can check out such as security services, data analytical services, etc.



Amazon Web Services