**Lecture Notes for AWS Developer**

**Module-4**

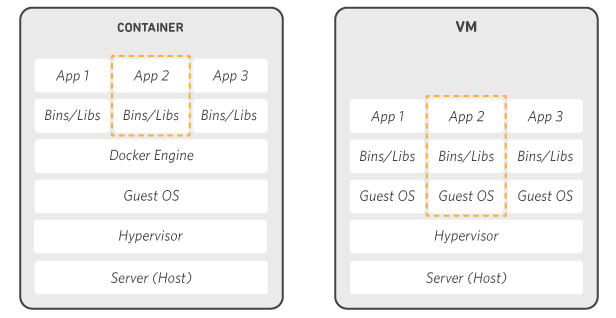
**AWS Docker**

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

Running Docker on AWS provides developers and admins a highly reliable, low-cost way to build, ship, and run distributed applications at any scale. Docker collaborates with AWS to help developers speed delivery of modern apps to the cloud. This collaboration helps developers use Docker Compose and Docker Desktop to leverage the same local workflow they use today to seamlessly deploy apps on Amazon ECS and AWS Fargate.

**How Docker works**

Docker works by providing a standard way to run your code. Docker is an operating system for containers. Similar to how a virtual machine virtualizes (removes the need to directly manage) server hardware, containers virtualize the operating system of a server. Docker is installed on each server and provides simple commands you can use to build, start, or stop containers. AWS services such as AWS Fargate, Amazon ECS, Amazon EKS, and AWS Batch make it easy to run and manage Docker containers at scale.



**Why to use Docker**

**SHIP MORE SOFTWARE FASTER**

Docker users on average ship software 7x more frequently than non-Docker users. Docker enables you to ship isolated services as often as needed.

**STANDARDIZE OPERATIONS**

Small containerized applications make it easy to deploy, identify issues, and roll back for remediation.

**SEAMLESSLY MOVE**

Docker-based applications can be seamlessly moved from local development machines to production deployments on AWS.

**SAVE MONEY**

Docker containers make it easier to run more code on each server, improving your utilization and saving you money.

**Amazon Elastic Container Service (Amazon ECS)**

Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service. Customers such as Duolingo, Samsung, GE, and Cookpad use ECS to run their most sensitive and mission critical applications because of its security, reliability, and scalability.

ECS is a great choice to run containers for several reasons. First, you can choose to run your ECS clusters using AWS Fargate, which is serverless compute for containers. Fargate removes the need to provision and manage servers, lets you specify and pay for resources per application, and improves security through application isolation by design. Second, ECS is used extensively within Amazon to power services such as Amazon SageMaker, AWS Batch, Amazon Lex, and Amazon.com’s recommendation engine, ensuring ECS is tested extensively for security, reliability, and availability.

**AWS Elastic Beanstalk**

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time. There is no additional charge for Elastic Beanstalk - you pay only for the AWS resources needed to store and run your applications.

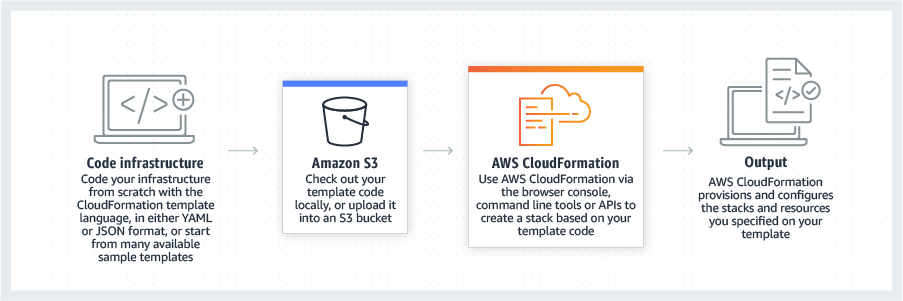
**Benefits**

* Fast and simple to begin
* Developer productivity
* Impossible to outgrow
* Complete resource control

**AWS CloudFormation**

AWS CloudFormation gives you an easy way to model a collection of related AWS and third-party resources, provision them quickly and consistently, and manage them throughout their lifecycles, by treating infrastructure as code. A CloudFormation template describes your desired resources and their dependencies so you can launch and configure them together as a stack. You can use a template to create, update, and delete an entire stack as a single unit, as often as you need to, instead of managing resources individually. You can manage and provision stacks across multiple AWS accounts and AWS Regions.

**How it works**



**Benefits**

* Automate best practices
* Scale your infrastructure worldwide
* Integrate with other AWS services
* Manage third-party and private resources
* Extend CloudFormation with the community