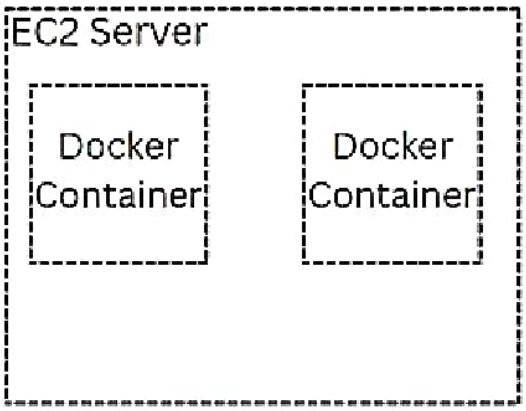
SESSION 14

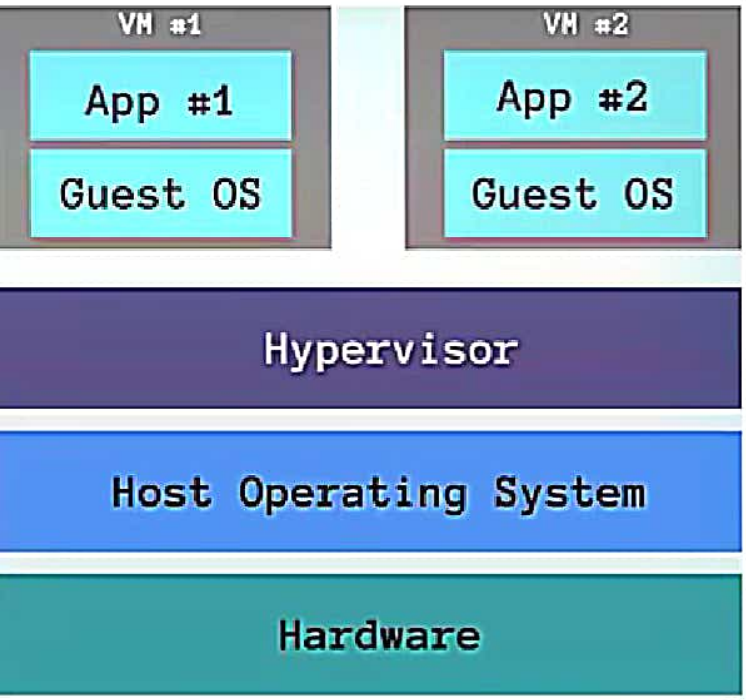
AWS CONTAINERS

Docker

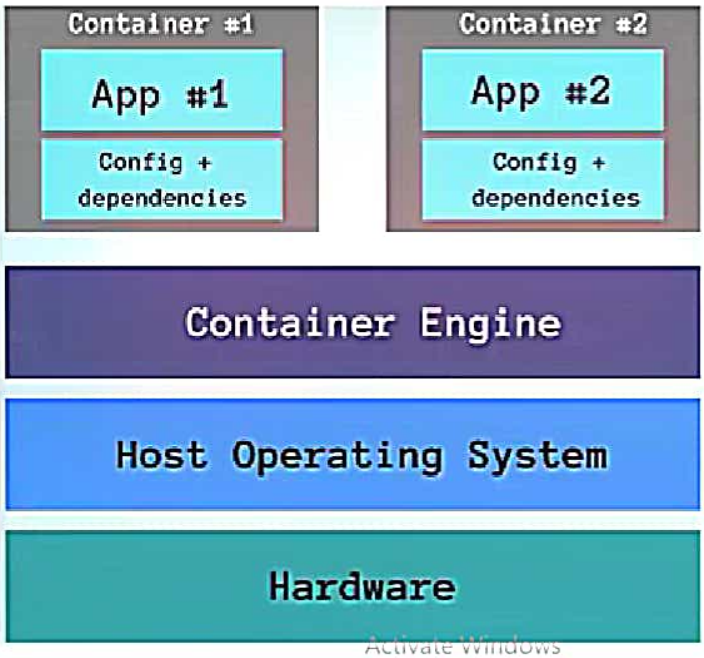
- Docker is a technology that allows you to incorporate and store your code and its dependencies into a neat little package – an image.
- Apps are packed in containers that cna run on any OS
- Apps run the same regardless of where they are run: Any machine, no compatible issue, predictable behaviour, work with any language, any OS, any technology
- Use cases: lift- and- shift apps from on-premises to the AWS cloud



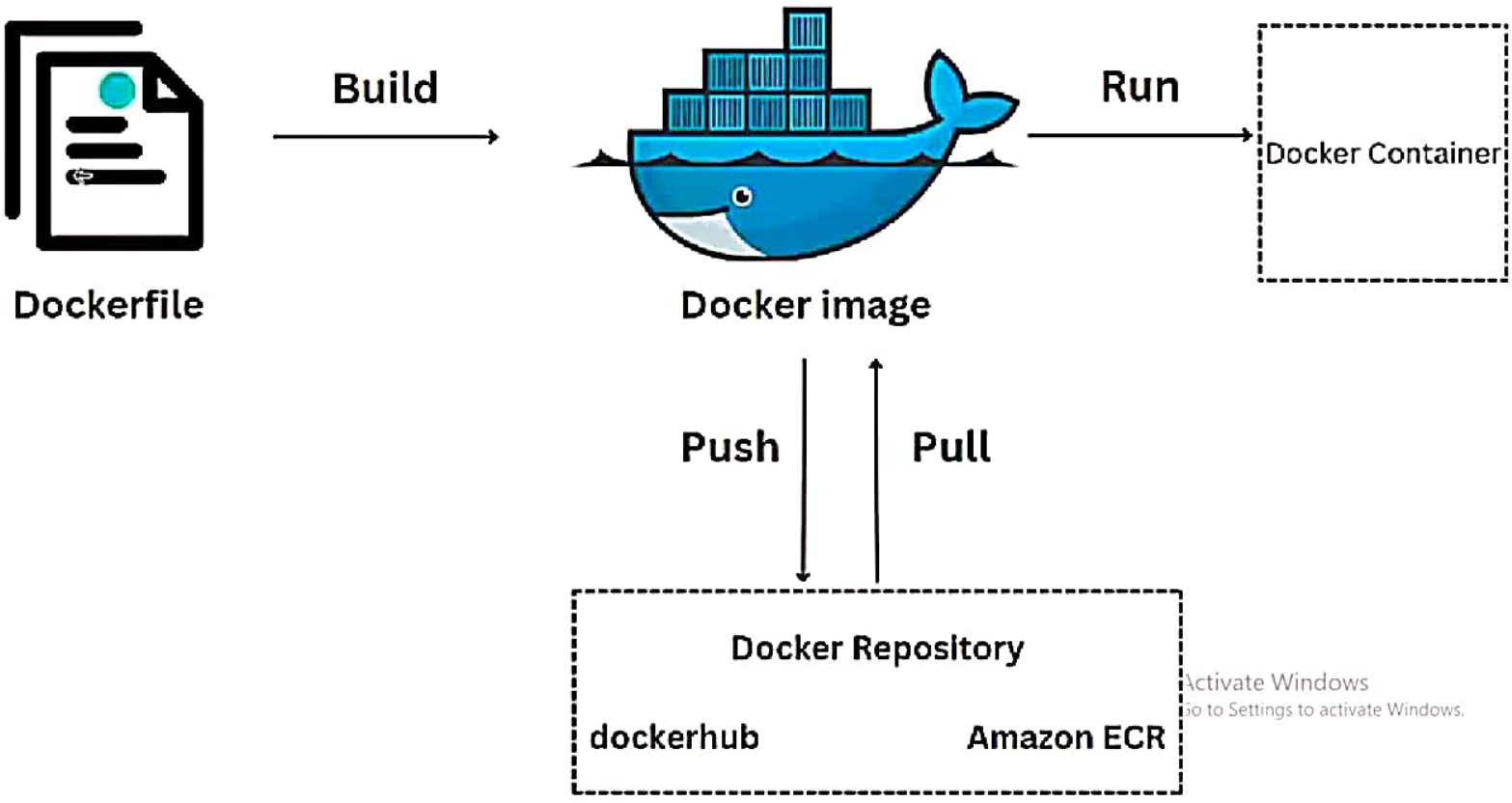
Docker VS Virtual Machine



Virtual Machine



Docker



Docker Container Management in AWS

- **Amazon Elastic Container Service(Amazon ECS)):** Amazon own container platform
- **Amazon Elastic Kubernetes Service(Amazon EKS)**
- **Amazon Fargate:** Amzon's own serverless container platform. It works with ECS and EKS
- **Amazon ECR:** Store container images

Amazon ECS

- Amazon Elastic Container Service (Amazon ECS) is a container management service that can quickly launch, exit, and manage docker containers in a cluster.
- It maintains the availability of the application and allows every user to sell containers when necessary.
- It meets the availability of the application: one container hosting your application should be running all the time, to meet that high availability, therefore it becomes important for the container to make sure that the service is running 24/7.
- Amazon ECS makes it a lot **easier by automatically scaling the number of containers needed to meet the demand.**
- With Amazon ECS, developers can pull the necessary Docker images and resources from Amazon Elastic Container Registry (ECR), or other repositories, to define their application. The ECS service then ingests container images and arranges or composes containers and resources into an application. Once all the appropriate containers are gathered and services implemented, the containers are deployed either on EC2 or AWS Fargate. Finally, Amazon ECS scales the application and continuously manages the availability of containers.

Amazon ECS launches containers through Fargate or EC2.

Fargate: You do not provision the infrastructure(no EC2 instance t manage). It is all serverless. AWS runs ECS tasks for you based on the CPU/RAM you need.
The Fargate launch type offers a serverless computing alternative that provisions, launches and runs containers without the need to manage the underlying infrastructure. Fargate is best for small, batch or highly scalable -- that is, burstable -- workloads with relatively short time-to-live (TTL) requirements.

EC2. The EC2 launch type is a more traditional deployment option. Users can provision and deploy EC2 instances to run containers while the service manages the related infrastructure and services. EC2 is well-suited for larger and more demanding workloads, applications that require persistent storage, applications that benefit from careful resource tuning or configuration and where direct management of the AWS infrastructure is desired.

Amazon Fargate

AWS Fargate is an Amazon Elastic Container Service (ECS) compute engine that enables you to run containers without needing to create, manage and scale clusters of

AWS Fargate eliminates the requirement for customers to manage their own EC2 instances. Users are not required to utilize AWS EC2 instances at all. Fargate will serve as the computational engine. It allows you to concentrate on aspects such as planning and building your application rather than maintaining the infrastructure that supports it. All you have to do with the Fargate launch type is package your application in containers, select the memory and CPU requirements, establish IAM policies, and start your application. VMs that host container applications.

AWS Fargate also makes scaling your apps simple. AWS Fargate provides all of the scalability and infrastructure required to operate your containers in a highly available way once you declare all of your application needs. It interfaces with Amazon ECS and EKS, launching and managing your containers for you.

Amazon ECR

- Elastic Container Registry
- It is fully managed container registry that makes it easy to store, manage, share and deploy your container images.
- Store and manage Docker images on AWS
- Private and public repository
- Fully integrated with ECS , allowing easy retrieval of container images from ECR while managing containers using ECS.
- IAM permissions are required

Amazon EKS

Amazon EKS is a managed service that is used to run Kubernetes on AWS. Using EKS users don’t have to maintain a Kubernetes control plan on their own. It is used to automate the deployment, scaling, and maintenance of the containerized application. It works with most operating systems.
EKS is integrated with various AWS services:

- ECR (Elastic Container Registry) for container images.
- Elastic Load Balancer for distributing traffic.
- IAM for providing authentication and authorization.
- VPC (Virtual Private Cloud) for isolating resources.

Questions:

1. You have multiple Docker-based applications hosted on-premises that you want to migrate to AWS. You don't want to provision or manage any infrastructure, you just want to run your containers on AWS. Which AWS service should you choose?
2. Amazon ECS has two launch type: _____ and _____
3. You are migrating your on-premises Docker- based applications to Amazon ECS. You were using Docker Hub Container Image Library as your container image repository. Which is an alternative AWS service which is fully integrated with Amazon ECS?

1. AWS Fargate on ECS 2. Amazon EC2 launch type and Amazon Fargate Launch Type
4. Elastic Container Registry(ECR)

AWS Glue

- AWS Glue is simply a serverless ETL tool.
- ETL refers to three processes that are commonly needed in most Data Analytics / Machine Learning processes: **Extraction, Transformation, Loading**. *Extracting* data from a source, *transforming* it in the right way for applications, and then *loading it* back to the data warehouse. And AWS helps us to make the magic happen. AWS console UI offers straightforward ways for us to perform the whole task to the end. No extra code scripts are needed.
-

Components of AWS Glue

- **Data catalog**: The data catalog holds the metadata and the structure of the data.
- **Database:** It is used to create or access the database for the sources and targets.
- **Table**: Create one or more tables in the database that can be used by the source and target.
- **Crawler and Classifier:** A crawler is used to retrieve data from the source using built-in or custom classifiers. It creates/uses metadata tables that are pre-defined in the data catalog.
- **Job**: A job is business logic that carries out an ETL task. Internally, Apache Spark with python or scala language writes this business logic.
- **Trigger:** A trigger starts the ETL job execution on-demand or at a specific time.
- **Development endpoint**: It creates a development environment where the ETL job script can be tested, developed, and debugged.

- Create an IAM role to access AWS Glue + S3+Athena
- Upload source files to Amazon S3
- Start the AWS Glue Database
- Create and Run Glue Crawlers
- Define Glue Jobs
- https://towardsdatascience.com/aws-glue-101-all-you-need-to-know-with-a-real-world-example-f34af17b782f