1. **How do you deploy Docker containers on AWS?**

* You can deploy Docker containers on AWS using services like **Amazon ECS (Elastic Container Service), Amazon EKS (Elastic Kubernetes Service), or AWS Fargate**.
* The typical deployment steps include:
  + Building a Docker image and pushing it to **Amazon Elastic Container Registry (ECR)**.
  + Creating a **task definition** in ECS.
  + Launching the container on an **ECS cluster** (EC2 or Fargate) or an **EKS cluster** (Kubernetes).
  + Configuring networking, scaling, and security policies.

2. **What is the role of Amazon ECR (Elastic Container Registry)?**

* Amazon ECR is a fully managed **container registry** for storing, managing, and deploying Docker container images.
* It integrates with AWS services like ECS, EKS, and Fargate.
* Provides security features like **IAM permissions and encryption**.
* Supports **image scanning** for vulnerabilities.

3.**What is the difference between ECS and EKS in AWS?**

* **Amazon ECS (Elastic Container Service):**
  + AWS-managed container orchestration service.
  + Works with AWS-native infrastructure like EC2 and Fargate.
  + Simpler to set up compared to Kubernetes.
* **Amazon EKS (Elastic Kubernetes Service):**
  + Managed **Kubernetes** service on AWS.
  + Provides full Kubernetes functionality.
  + Suitable for workloads requiring Kubernetes flexibility.

4. **How does Docker integrate with AWS CI/CD pipelines?**

* Docker integrates with AWS CI/CD tools like **AWS CodePipeline and AWS CodeBuild**.
* Typical pipeline:
  + **CodeCommit/GitHub** → **CodeBuild (build Docker image)** → **Push to Amazon ECR** → **Deploy to ECS/EKS using CodeDeploy**.
  + AWS also supports **Jenkins, GitHub Actions, and GitLab CI/CD** for Docker-based deployments.

5.**What is the role of AWS Fargate in Docker container deployment?**

* AWS Fargate is a **serverless compute engine** for containers.
* Eliminates the need to manage EC2 instances.
* Automatically scales containers.
* Works with **ECS and EKS**, making deployment easier and more cost-efficient.

6. **How do you manage Docker container scaling on AWS?**

* Use **AWS Auto Scaling** for ECS services (EC2 or Fargate).
* In Kubernetes (EKS), use **Horizontal Pod Autoscaler (HPA)** and **Cluster Autoscaler**.
* Scale based on metrics like **CPU, memory, and request rates** using **Amazon CloudWatch**.

7.**What is a Docker Compose file, and how can it be used with AWS?**

* A **Docker Compose file (docker-compose.yml)** defines multi-container applications.
* AWS offers **AWS ECS CLI** support for deploying docker-compose.yml configurations to ECS.
* Example:

yaml

CopyEdit

version: '3'

services:

web:

image: myapp:latest

ports:

- "80:80"

depends\_on:

- db

db:

image: postgres

environment:

POSTGRES\_PASSWORD: example

* Can be deployed to **ECS using AWS Copilot or ECS CLI**.

8. **How do you monitor Docker containers in AWS?**

* Use **Amazon CloudWatch** for logs and metrics.
* Enable **AWS X-Ray** for distributed tracing.
* For Kubernetes, use **Prometheus + Grafana**.
* AWS offers **Container Insights** to monitor ECS and EKS performance.

9. **How do you secure Docker containers on AWS?**

* Use **IAM roles and policies** to restrict access.
* Enable **image scanning in Amazon ECR** for vulnerability detection.
* Use **security groups and VPC network isolation**.
* Implement **AWS Secrets Manager** for managing sensitive data like passwords and API keys.
* Enable **runtime security tools like AWS GuardDuty or Falco** for monitoring threats.

10. **How do you optimize cost while running Docker containers on AWS?**

* Use **AWS Fargate Spot Instances** for cost savings.
* Choose **EC2 Spot Instances** for ECS clusters.
* Optimize **resource allocation** (CPU & memory) in task definitions.
* Use **Amazon Auto Scaling** to scale down containers when traffic is low.
* Monitor cost using **AWS Cost Explorer** and **Compute Optimizer**.