Interview Tips Day 7: AWS for DevOps – 12 Common Advance Questions (Part-2)

1. What is AWS Lambda and how can it be utilized in DevOps practices?

Example: "AWS Lambda is a serverless compute service that allows running code without provisioning or managing servers. In DevOps, Lambda can automate various tasks like triggering deployments, running tests, or managing infrastructure events. For instance, I could create a Lambda function that automatically scales EC2 instances based on specific metrics like CPU usage."

2. Describe AWS Elastic Beanstalk and its benefits.

Example: "AWS Elastic Beanstalk simplifies deploying and managing web applications by handling the underlying infrastructure automatically. For a fresher, let's consider deploying a Python Flask application. Instead of configuring servers and load balancers, I'd use Elastic Beanstalk, which automates the setup, scaling, and monitoring, allowing me to focus on the application's code."

3. What are the differences between Amazon S3 and EBS (Elastic Block Store)?

Example: "Amazon S3 is an object storage service suitable for storing and retrieving large amounts of unstructured data, while EBS is block-level storage attached to EC2 instances, used for persistent data storage. For instance, if I'm building a media storage system, I'd use S3 for storing videos and EBS to hold the database files required to manage those videos within an EC2 instance."

4. Explain the concept of AWS Identity and Access Management (IAM) and its importance in DevOps.

Example: "AWS IAM manages user access to AWS services and resources. In a DevOps environment, IAM ensures secure and controlled access to AWS resources. For example, I would use IAM to create roles and policies that define which team members can deploy infrastructure or access sensitive services, maintaining a secure and manageable environment."

5. How does AWS CodePipeline contribute to a DevOps workflow?

Example: "AWS CodePipeline is a continuous integration and continuous delivery (CI/CD) service that automates the software release process. As a fresher, I could use CodePipeline to orchestrate build, test, and deployment stages for a web application. For instance, connecting CodePipeline with a GitHub repository triggers automated testing on every code commit and deploys to an AWS environment upon successful tests."

6. Discuss the benefits of using AWS CloudWatch in a DevOps environment.

Example: "AWS CloudWatch is a monitoring service that collects and tracks metrics, logs, and events from AWS resources and applications. As a fresher, I'd utilize CloudWatch to set up alarms for specific metrics. For instance, I could create an alarm to notify the team if CPU utilization on EC2 instances exceeds a certain threshold, ensuring proactive responses to potential issues."

7. Explain the difference between AWS RDS and DynamoDB.

Example: "AWS RDS (Relational Database Service) is a managed service offering relational databases like MySQL, PostgreSQL, etc., whereas DynamoDB is a fully managed NoSQL database service. As a fresher, if I need to store structured data requiring ACID transactions, I'd choose RDS, while for highly scalable and flexible data models, I'd opt for DynamoDB, suitable for applications needing high-performance and scalability."

8. How can AWS Auto Scaling be advantageous in a DevOps setup?

Example: "AWS Auto Scaling automatically adjusts the number of instances in response to demand or based on defined metrics. For a fresher, I'd use Auto Scaling to ensure application availability and cost optimization. For instance, during peak traffic periods, Auto Scaling can dynamically add EC2 instances to handle increased load, and it can scale down during low traffic to minimize costs."

9. Discuss the significance of AWS ECS (Elastic Container Service) in container orchestration for DevOps.

Example: "AWS ECS is a container orchestration service that simplifies deploying, managing, and scaling containerized applications using Docker. For instance, in a DevOps context, I'd utilize ECS to deploy microservices as containers. It provides flexibility in managing resources, deploying updates, and maintaining the availability of services through load balancing and task scheduling."

10. What is AWS CloudTrail, and how does it contribute to security and auditing in AWS?

Example: "AWS CloudTrail is a service that records AWS API calls for auditing, compliance, and security analysis. As a fresher, I'd use CloudTrail to monitor actions performed on AWS resources. For example, it helps in identifying who made specific changes or accessed sensitive data, ensuring accountability and enabling compliance with security policies."

11. Explain the concept of VPC (Virtual Private Cloud) and its relevance in a DevOps environment.

Example: "VPC enables users to create a logically isolated virtual network within the AWS cloud. In a DevOps role, I'd use VPC to securely launch resources like EC2 instances or RDS databases in a private network with controlled access. For instance, setting up multiple subnets, route tables, and security groups within a VPC ensures a secure and segregated environment for deploying applications and managing traffic flow."

12. Explain the use of AWS CloudFormation.

Example: "AWS CloudFormation is an infrastructure-as-code service used for provisioning and managing AWS resources through templates. For instance, imagine needing to create a consistent environment for a web application. Instead of manually configuring resources like EC2 instances, load balancers, and databases, I'd create a CloudFormation template defining these resources and their configurations. This template can be version-controlled and reused to deploy the same environment reliably."