AWS DevOps Scenario-Based Questions

I. CI/CD Pipeline & Automation (40 Questions)

1. **Scenario:** You need to set up a fully automated CI/CD pipeline for a new microservices application hosted on AWS. The application uses Docker containers.
   - How would you design the pipeline using AWS CodePipeline, CodeBuild, CodeDeploy, and ECR?
   - What considerations would you have for unit testing, integration testing, and end-to-end testing within this pipeline?
   - How would you ensure that only thoroughly tested code reaches production?
2. **Scenario:** Your current CI/CD pipeline built with Jenkins is becoming a bottleneck due to scaling issues and maintenance overhead. You want to migrate to a fully managed AWS CI/CD solution.
   - Describe your migration strategy.
   - Which AWS services would you use to replace Jenkins' functionalities (source control, build, deploy, orchestration)?
   - How would you handle existing Jenkinsfiles and build scripts?
3. **Scenario:** A critical bug fix needs to be deployed to production as quickly as possible, bypassing some non-essential stages of the CI/CD pipeline.
   - How would you design your pipeline to allow for expedited deployments for hotfixes while maintaining proper controls?
   - What automated gates or approvals would you still recommend?
4. **Scenario:** Your development team frequently pushes small code changes, leading to many build and deployment cycles. You want to optimize the CI/CD pipeline for faster feedback and reduced build times.
   - What strategies would you employ to speed up the build process in AWS CodeBuild?
   - How can you parallelize testing within the pipeline?
5. **Scenario:** You need to implement a blue/green deployment strategy for a web application hosted on Amazon EC2 to minimize downtime during deployments.
   - Explain how you would achieve this using AWS CodeDeploy and an Application Load Balancer (ALB).
   - What steps would you include for health checks and traffic shifting?
   - How would you handle a rollback if the new "green" environment has issues?
6. **Scenario:** Your application is serverless, built with AWS Lambda and API Gateway. You need to automate the deployment process with canary releases.
   - How would you implement canary deployments for Lambda functions using AWS CodeDeploy and API Gateway?
   - What metrics would you monitor during the canary release, and how would you automate rollbacks based on these metrics?

7. **Scenario:** A new developer joins the team and needs access to the CI/CD pipeline. You want to ensure least privilege access and proper security.
   - How would you manage IAM roles and policies for different stages of the CI/CD pipeline (e.g., source, build, deploy)?
   - What considerations would you have for secret management within the pipeline (e.g., database credentials)?
8. **Scenario:** Your CI/CD pipeline needs to trigger deployments based on events from a third-party Git repository (e.g., GitHub Enterprise).
   - How would you integrate this external source with AWS CodePipeline?
   - What security measures would you put in place for the integration?
9. **Scenario:** You want to implement automated security scans (static analysis, dependency scanning) as part of your CI/CD pipeline.
   - At which stages of the pipeline would you integrate these scans?
   - Which AWS services or third-party tools would you consider for this purpose?
   - How would you handle identified vulnerabilities (e.g., failing the build, reporting)?
10. **Scenario:** Your application build process generates large artifacts that need to be stored and versioned efficiently.
    - Where would you store these build artifacts in AWS?
    - How would you ensure proper versioning and lifecycle management for these artifacts?
11. **Scenario:** You need to build and deploy multiple microservices from a single monorepo.
    - How would you configure your CodeBuild projects and CodePipeline stages to handle independent builds and deployments for each microservice?
12. **Scenario:** A new feature branch is created, and you want to deploy a temporary environment for testing this feature before merging to `main`.
    - How would you automate the provisioning of a temporary testing environment using your CI/CD pipeline?
    - How would you ensure these temporary environments are torn down after use?
13. **Scenario:** Your CI/CD pipeline is experiencing intermittent failures in the build stage.
    - What AWS services and strategies would you use for troubleshooting and debugging these failures?
    - How would you implement better logging and observability for your CodeBuild projects?
14. **Scenario:** You need to ensure that every code commit goes through a formal approval process before deployment to production.
    - How would you incorporate manual approval steps into your AWS CodePipeline?
    - What kind of notifications would you set up for these approval steps?
15. **Scenario:** Your application requires custom build environments with specific tools and libraries not available in standard CodeBuild images.
    - How would you create and use a custom build environment in AWS CodeBuild?
16. **Scenario:** You want to perform performance testing on your application as part of the CI/CD pipeline.
    - How would you integrate a performance testing tool (e.g., Apache JMeter, Locust) into your CodePipeline?

    - How would you analyze and report on the performance test results?

17. **Scenario:** Your team wants to adopt GitFlow branching strategy.
    - How would you configure your CodeCommit repositories and CodePipeline to support the GitFlow branching model?
18. **Scenario:** You need to trigger your CI/CD pipeline based on changes in an Amazon S3 bucket (e.g., configuration files).
    - How would you set up this S3-triggered pipeline?
19. **Scenario:** Your deployment to a staging environment keeps failing due to missing environment variables.
    - How would you manage and inject environment-specific variables into your CodeBuild and CodeDeploy processes securely?
20. **Scenario:** You're migrating an existing application that uses specific version of Python and Node.js.
    - How do you ensure CodeBuild uses the exact versions required for your application?
21. **Scenario:** Your pipeline needs to deploy to multiple AWS accounts (e.g., Dev, Staging, Prod).
    - How would you structure your CodePipeline to handle cross-account deployments securely?
    - What IAM roles and trust policies would be necessary?
22. **Scenario:** You want to implement infrastructure testing (e.g., using Terratest or InSpec) as part of your CI/CD pipeline.
    - Where in the pipeline would these tests be executed, and what AWS services would facilitate this?
23. **Scenario:** Your development team wants to receive Slack notifications for successful and failed deployments.
    - How would you integrate AWS Chatbot with CodePipeline for these notifications?
24. **Scenario:** You need to enforce specific code quality standards (e.g., linting, cyclomatic complexity checks) before code can be merged.
    - How would you integrate code quality gates into your CI/CD pipeline using AWS services or third-party tools?
25. **Scenario:** You want to implement A/B testing for a new feature.
    - How could your CI/CD pipeline help in deploying and managing different versions for A/B testing?
    - What AWS services would you leverage for traffic routing and monitoring?
26. **Scenario:** Your application has a complex database schema that needs to be migrated with each deployment.
    - How would you incorporate automated database migrations into your CI/CD pipeline, ensuring reversibility and data integrity?
27. **Scenario:** You have a mono-repository with multiple services, and you only want to trigger a build for a specific service when its code changes.
    - How would you configure your CodeBuild and CodePipeline to achieve this selective triggering?

28. **Scenario:** You need to automate the creation of new user accounts and their permissions in an application after deployment.
    - How would you integrate a post-deployment script for user provisioning into your CI/CD pipeline?
29. **Scenario:** Your organization requires detailed audit trails for every deployment.
    - Which AWS services would you use to track and log all CI/CD pipeline activities, including who deployed what and when?
30. **Scenario:** You are using AWS Elastic Beanstalk for application deployment, and you need to automate updates and rollbacks.
    - How would you integrate Elastic Beanstalk into a CodePipeline for continuous deployment?
    - What deployment policies would you configure in Elastic Beanstalk to minimize downtime?
31. **Scenario:** You want to incorporate immutable infrastructure principles into your deployments.
    - How would your CI/CD pipeline create new EC2 AMIs with each build and deploy them without modifying existing instances?
    - What AWS services would be central to this approach?
32. **Scenario:** A new security vulnerability is discovered in a common library used by your application. You need to rapidly redeploy all affected services with the patched version.
    - How would you use your CI/CD pipeline to identify affected services and orchestrate a mass redeployment?
33. **Scenario:** Your CI/CD pipeline needs to deploy containerized applications to Amazon EKS.
    - Describe the stages involved in building a Docker image, pushing it to ECR, and deploying it to EKS using CodePipeline.
    - How would you manage Kubernetes manifests?
34. **Scenario:** You need to manage environment-specific configurations (e.g., API endpoints, database names) for your application across Dev, UAT, and Production environments.
    - How would you store and inject these configurations into your application at deployment time using AWS services?
35. **Scenario:** Your team wants to shift left on security by integrating security scanning tools directly into the development workflow and CI/CD.
    - What specific types of security scans would you integrate and at what points in the pipeline?
    - How would you ensure developers receive timely feedback on security issues?
36. **Scenario:** Your application uses a polyglot architecture (multiple programming languages).
    - How would your CI/CD pipeline handle building and testing code written in different languages (e.g., Python, Java, Node.js)?
37. **Scenario:** You need to deploy static website content to Amazon S3 and serve it via CloudFront.

  - How would you automate the CI/CD pipeline for this, ensuring invalidation of CloudFront cache on new deployments?

38. **Scenario:** You want to implement a "push-button" rollback mechanism for your deployments.
    - How would you design your CodePipeline to enable easy and fast rollbacks to a previous successful version?
39. **Scenario:** Your CI/CD pipeline needs to build and deploy a desktop application to end-users (e.g., via S3 for download).
    - How would the deployment stage of your pipeline differ from a web application deployment?
    - How would you handle versioning and notification of new releases?
40. **Scenario:** You are implementing a feature flag system to control feature rollout.
    - How would your CI/CD pipeline integrate with and deploy changes related to feature flags?

II. Infrastructure as Code (IaC) & Configuration Management (30 Questions)

41. **Scenario:** Your team is currently provisioning AWS resources manually through the console, leading to inconsistencies and errors. You want to adopt Infrastructure as Code.
    - Which AWS IaC service would you recommend (CloudFormation vs. CDK) and why?
    - How would you handle existing manually provisioned resources? (Hint: Drift detection)
42. **Scenario:** You need to deploy a complex, multi-tier application stack (VPC, subnets, EC2, RDS, ALB) in a repeatable manner across different environments (Dev, Staging, Prod).
    - How would you structure your CloudFormation templates (e.g., nested stacks, parameters) to manage this complexity?
    - How would you ensure consistency while allowing for environment-specific variations?
43. Scenario: Your CloudFormation stack update failed, and it's stuck in UPDATE_ROLLBACK_FAILED state.
    * What steps would you take to recover the stack and identify the root cause of the failure?
44. **Scenario:** You need to manage configuration drift for your EC2 instances.
    - How would you use AWS Systems Manager State Manager or Ansible/Chef/Puppet to ensure instances remain in their desired state?
    - What's the difference between using Systems Manager State Manager and traditional configuration management tools for this purpose?
45. **Scenario:** Your team is using Terraform for IaC, and you need to manage Terraform state files securely and collaboratively.
    - Where would you store your Terraform state files, and how would you ensure remote backend locking?
    - How would you handle sensitive data within your Terraform configurations?

46. **Scenario:** You need to deploy a serverless application consisting of AWS Lambda functions, API Gateway, and DynamoDB tables.
    - How would you define this infrastructure using AWS Serverless Application Model (SAM) or AWS CloudFormation?
    - What are the benefits of using SAM over raw CloudFormation for serverless applications?
47. **Scenario:** You have a common set of network resources (VPC, subnets, security groups) that need to be shared across multiple application stacks.
    - How would you design your IaC to create and manage these shared resources, and then reference them in application-specific templates?
48. **Scenario:** You need to automate the patching of operating systems on your EC2 instances.
    - How would you use AWS Systems Manager Patch Manager to achieve this, ensuring minimal downtime and proper reporting?
49. **Scenario:** Your organization has a strict naming convention for all AWS resources.
    - How would you enforce this naming convention using IaC (e.g., CloudFormation tags, Terraform local values)?
50. **Scenario:** You need to ensure that all EC2 instances launched comply with specific security configurations (e.g., no public IP, specific security groups).
    - How would you use AWS Config Rules to monitor and enforce these compliance policies?
    - How would you integrate this with your IaC deployments?
51. **Scenario:** You are tasked with migrating an existing on-premises application that relies heavily on Windows Server and Active Directory to AWS.
    - How would you provision the necessary Windows EC2 instances and integrate with AWS Directory Service using IaC?
52. **Scenario:** You need to implement a "golden AMI" strategy for your EC2 instances.
    - How would you automate the creation and updating of these golden AMIs using AWS services like Packer, EC2 Image Builder, or a custom CodeBuild pipeline?
53. **Scenario:** Your IaC templates (CloudFormation/Terraform) need to consume secrets (e.g., database passwords) without hardcoding them.
    - How would you integrate AWS Secrets Manager or AWS Systems Manager Parameter Store with your IaC for secret injection?
54. **Scenario:** You need to perform a "dry run" or validate your IaC templates before actual deployment to catch errors early.
    - How would you achieve this for CloudFormation and Terraform?
55. **Scenario:** Your development team frequently needs to spin up new environments for testing new features, and then tear them down.
    - How would you automate the provisioning and de-provisioning of these ephemeral environments using IaC?
56. **Scenario:** You're managing a large number of EC2 instances, and you need to ensure they have the latest application configuration applied automatically.
    - How would you use AWS Systems Manager Distributor and Run Command to deploy configuration updates across your fleet?

57. **Scenario:** You need to define a consistent security baseline for all new EC2 instances, including security groups, NACLs, and instance roles.
    - How would you codify this security baseline using CloudFormation or Terraform and ensure it's applied to all new deployments?
58. **Scenario:** You are refactoring a monolithic CloudFormation template into smaller, modular components.
    - How would you approach this refactoring, and what are the benefits of doing so?
59. **Scenario:** You need to audit changes made to your AWS infrastructure.
    - How does CloudFormation/Terraform help in providing an audit trail for infrastructure changes?
    - What other AWS services would you leverage for comprehensive auditing?
60. **Scenario:** You want to implement a tagging strategy across all your AWS resources for cost allocation and resource identification.
    - How would you enforce mandatory tagging using IaC (CloudFormation/Terraform) and AWS Config?
61. **Scenario:** Your CloudFormation stack deployment fails due to a dependency issue (e.g., a resource trying to reference a non-existent resource).
    - How would you troubleshoot and resolve such dependency-related failures in CloudFormation?
62. **Scenario:** You need to grant specific, temporary access to an AWS resource for a maintenance task, without hardcoding credentials.
    - How would you use AWS Systems Manager Session Manager and IAM roles to provide secure, temporary access to EC2 instances?
63. **Scenario:** You are using CloudFormation, and a critical resource (e.g., an S3 bucket) was accidentally deleted outside of CloudFormation.
    - How would you detect this drift and reconcile your CloudFormation stack with the actual state?
64. **Scenario:** You want to ensure that all S3 buckets created in your AWS account have encryption enabled by default.
    - How would you enforce this using AWS CloudFormation and AWS Config?
65. **Scenario:** You need to orchestrate a complex deployment that involves launching resources in a specific order and waiting for them to be healthy before proceeding.
    - How would you use CloudFormation wait conditions or custom resources to manage these dependencies?
66. **Scenario:** Your team needs to share common IaC modules (e.g., a standard VPC module) across multiple projects.
    - How would you achieve this reusability with CloudFormation (e.g., nested stacks, macros) or Terraform (e.g., modules)?
67. **Scenario:** You're using Ansible to configure your EC2 instances after they are launched by CloudFormation.
    - How would you integrate Ansible playbooks into your CloudFormation template or a post-launch script?
68. **Scenario:** You want to use AWS CloudFormation StackSets to deploy the same CloudFormation template to multiple AWS accounts and regions.

    - Describe a use case for StackSets and how you would set it up.

69. **Scenario:** You need to define custom security groups that allow traffic only from specific VPCs or IP ranges.
    - How would you define these security group rules in your IaC templates?
70. **Scenario:** You're managing stateful applications (e.g., databases) with IaC.
    - What precautions and strategies would you employ to manage changes to these resources without data loss during IaC updates?

III. Monitoring, Logging & Alerting (30 Questions)

71. **Scenario:** Your production web application is experiencing intermittent slow response times, but you're not getting any alerts.
    - What AWS monitoring and logging services would you use to investigate this issue?
    - How would you set up proactive alerts for similar issues in the future?
72. **Scenario:** You need to collect application logs from EC2 instances and centralize them for analysis and troubleshooting.
    - How would you use AWS CloudWatch Logs to achieve this?
    - How would you implement log parsing and filtering?
73. **Scenario:** Your application runs on AWS Lambda, and you need to monitor its performance, invocations, and errors.
    - What CloudWatch metrics would you focus on for Lambda, and how would you set up alarms?
    - How would you use CloudWatch Logs Insights to analyze Lambda function logs?
74. **Scenario:** You want to create a comprehensive dashboard to visualize the health and performance of your entire application stack.
    - How would you use CloudWatch Dashboards to aggregate metrics from various AWS services (EC2, RDS, ALB, Lambda, etc.)?
75. **Scenario:** Your team needs to be notified via Slack or PagerDuty when critical application errors occur.
    - How would you integrate CloudWatch Alarms with SNS and then with a third-party notification service?
76. **Scenario:** You need to audit all API calls made to your AWS account for security and compliance purposes.
    - How would you use AWS CloudTrail to achieve this?
    - How would you store and analyze CloudTrail logs effectively?
77. **Scenario:** Your database (Amazon RDS) is experiencing high CPU utilization during peak hours.
    - How would you use CloudWatch metrics for RDS to identify the bottleneck?
    - What actions would you recommend based on your findings (e.g., scaling, query optimization)?
78. **Scenario:** You are implementing a new microservice, and you need to set up distributed tracing to understand request flow and latency across services.

    - How would you use AWS X-Ray for this purpose, and how would you instrument your application code?

79. **Scenario:** You need to monitor the cost of your AWS resources and receive alerts if costs exceed a certain threshold.
    - How would you use AWS Budgets and Cost Explorer for cost monitoring and alerting?
80. **Scenario:** Your application logs contain sensitive data (e.g., PII). You need to redact or mask this data before it's stored in CloudWatch Logs.
    - How would you implement log sanitization or redaction as part of your logging strategy?
81. **Scenario:** You want to analyze log data from multiple sources (EC2, Lambda, VPC Flow Logs) to identify security threats or anomalies.
    - How would you use AWS Athena or a third-party SIEM solution (e.g., Splunk) with your centralized logs?
82. **Scenario:** You need to collect custom application metrics (e.g., number of user sign-ups, successful API calls) and push them to CloudWatch.
    - How would you implement custom metrics collection for your application?
83. **Scenario:** Your application is generating a high volume of logs, leading to increased CloudWatch costs.
    - What strategies would you employ to optimize log ingestion and storage costs? (e.g., log retention, filtering)
84. **Scenario:** You need to set up an automated response to a specific alarm, such as stopping an unhealthy EC2 instance.
    - How would you use CloudWatch Alarms to trigger an EC2 action or an SNS topic that invokes a Lambda function for automated remediation?
85. **Scenario:** Your team requires real-time dashboards for operational metrics during major deployments.
    - How would you leverage CloudWatch Dashboards and widgets to provide immediate visibility into deployment health?
86. **Scenario:** You want to ensure that all critical security groups have specific inbound/outbound rules.
    - How would you use AWS Config to monitor for non-compliant security group configurations and generate alerts?
87. **Scenario:** Your development team wants to debug issues in a shared development environment without direct SSH access to instances.
    - How would you enable secure and audited debugging using AWS Systems Manager Run Command and CloudWatch Logs?
88. **Scenario:** You need to predict future resource utilization trends for capacity planning.
    - How can historical data from CloudWatch metrics assist in capacity planning?
89. **Scenario:** You suspect an external attack on your web application.
    - What logging and monitoring sources would you immediately check (e.g., CloudFront access logs, ALB access logs, VPC Flow Logs)?
90. **Scenario:** You need to implement log archival for compliance purposes, storing logs for several years in a cost-effective manner.

    - How would you configure CloudWatch Logs to automatically export logs to S3 and manage their lifecycle?

91. **Scenario:** You are receiving too many "false positive" alarms from CloudWatch.
    - How would you fine-tune your CloudWatch alarm thresholds and metric definitions to reduce alert fatigue?
92. **Scenario:** Your application's performance varies significantly throughout the day.
    - How would you use CloudWatch's anomaly detection or composite alarms to capture these unusual patterns effectively?
93. **Scenario:** You need to analyze user activity within your AWS account to detect suspicious behavior.
    - How would you utilize CloudTrail events and integrate them with a security information and event management (SIEM) system?
94. **Scenario:** You want to monitor the health of your Amazon SQS queues, including messages in flight and message age.
    - What CloudWatch metrics are relevant for SQS, and how would you set up alerts for potential bottlenecks?
95. **Scenario:** Your application relies on external APIs, and you need to monitor the latency and error rates of these calls.
    - How would you use CloudWatch metrics and logs to track the performance of external API integrations?
96. **Scenario:** You need to perform log retention for different log groups based on their criticality.
    - How would you configure varied retention policies for CloudWatch Log Groups?
97. **Scenario:** You're deploying an application that uses Amazon Kinesis for real-time data streaming.
    - How would you monitor Kinesis stream utilization, put/get records, and errors using CloudWatch?
98. **Scenario:** You need to set up a "single pane of glass" for your operations team, integrating metrics, logs, and traces.
    - How would you use CloudWatch to bring together data from CloudWatch Metrics, CloudWatch Logs, and AWS X-Ray?
99. **Scenario:** Your application utilizes Amazon DynamoDB.
    - How would you monitor DynamoDB's read/write capacity units, throttled events, and latency using CloudWatch?
100. Scenario: You need to ensure that specific security configurations (e.g., encryption for RDS instances) are always applied and alerted if not.
    * How would you use AWS Config rules and remediation actions to enforce and monitor these configurations?

IV. Security & Compliance (30 Questions)

101. Scenario: Your company handles sensitive customer data, and you need to ensure strong security and compliance (e.g., GDPR, HIPAA) for your AWS environment.
    * What are the foundational AWS security services you would implement (IAM, VPC,

Security Groups, NACLs)?
* How would you automate compliance checks and reporting?

102. Scenario: You need to implement the principle of least privilege for all IAM users and roles in your AWS account.
* How would you approach defining granular IAM policies for different roles (developers, operations, auditors)?
* What tools would you use to review and refine existing IAM policies?

103. Scenario: Your organization requires encryption for all data at rest and in transit.
* How would you ensure data encryption for S3 buckets, RDS databases, EBS volumes, and inter-service communication (e.g., ALB to EC2)?

104. Scenario: You need to protect your web application from common web exploits like SQL injection and cross-site scripting.
* How would you deploy and configure AWS WAF with an Application Load Balancer (ALB) or CloudFront?

105. Scenario: You want to prevent unauthorized access to your S3 buckets.
* What S3 bucket policies and public access block settings would you configure?
* How would you regularly audit S3 bucket permissions?

106. Scenario: You need to securely manage database credentials, API keys, and other secrets for your applications.
* How would you use AWS Secrets Manager or AWS Systems Manager Parameter Store to store and rotate these secrets?
* How would applications retrieve these secrets securely?

107. Scenario: You need to establish secure connectivity between your on-premises data center and your AWS VPC.
* What are the options (VPN, Direct Connect), and when would you choose each?
* How would you configure the networking and security for this hybrid setup?

108. Scenario: Your security team wants to receive alerts for any suspicious activity in your AWS account, such as root user login or unusual API calls.
* How would you configure CloudTrail and CloudWatch Alarms to detect and notify on these events?

109. Scenario: You need to ensure that all EC2 instances are launched with a specific, hardened AMI and follow a security baseline.
* How would you enforce the use of golden AMIs and specific security group rules using AWS Config and IaC?

110. Scenario: You want to implement Multi-Factor Authentication (MFA) for all IAM users accessing the AWS Management Console.
* How would you enforce this across your organization?

111. Scenario: Your compliance requirements dictate that all network traffic within your VPC must be logged and monitored.
* How would you enable and configure VPC Flow Logs for central logging and analysis?

112. Scenario: You need to prevent accidental deletion of critical AWS resources (e.g., production databases, S3 buckets).
* How would you enable termination protection and S3 versioning, and what other preventative measures would you take?

113. Scenario: Your development team frequently needs temporary access to production EC2 instances for troubleshooting.
* How would you provide secure, time-limited, and auditable access without sharing SSH keys or opening inbound SSH ports? (Hint: SSM Session Manager)

114. Scenario: You need to conduct regular security vulnerability assessments of your EC2 instances.
* How would you use Amazon Inspector to automate vulnerability scanning and reporting?

115. Scenario: You want to centralize security findings and track remediation efforts across your AWS environment.
* How would you use AWS Security Hub to aggregate findings from various security services (Inspector, GuardDuty, Macie, WAF)?

116. Scenario: Your application requires encryption of data in transit between microservices within your VPC.
* How would you implement this using TLS/SSL and potentially AWS Certificate Manager (ACM)?

117. Scenario: You need to restrict access to an S3 bucket to only specific VPC endpoints, not public internet.
* How would you configure the S3 bucket policy and VPC endpoint policy to achieve this?

118. Scenario: You're concerned about potential data exfiltration from your VPC.
* How would you use VPC Flow Logs and AWS GuardDuty to detect and respond to suspicious network activity?

119. Scenario: You need to ensure that all sensitive data stored in S3 is discovered and classified.
* How would you use Amazon Macie to automate sensitive data discovery and protection?

120. Scenario: You want to manage your security group rules as code and integrate them with your CI/CD pipeline.
* How would you define security group rules in CloudFormation or Terraform, and ensure changes go through a review process?

121. Scenario: You need to prevent users from accidentally creating public S3 buckets.
* How would you implement an AWS Organizations Service Control Policy (SCP) to enforce this?

122. Scenario: Your application interacts with external APIs, and you need to securely manage the API keys for these integrations.
* How would you use Secrets Manager for storing these keys and ensure applications retrieve them dynamically?

123. Scenario: You're building a multi-tenant application and need to ensure strong isolation between tenants' data and resources.
* What security patterns and AWS services would you employ to achieve tenant isolation?

124. Scenario: You need to restrict IAM user permissions based on their IP address.
* How would you apply IP-based conditions to IAM policies?

125. Scenario: You want to implement a strong password policy for all IAM users in your AWS account.
* How would you configure the IAM account password policy?

126. Scenario: Your security team requires regular reports on your AWS environment's compliance posture.
* How would you use AWS Config and AWS Audit Manager to generate these reports automatically?

127. Scenario: You need to securely store Docker images and scan them for vulnerabilities before deployment.
* How would you use Amazon ECR and ECR Image Scanning?

128. Scenario: Your application is publicly accessible, and you need to protect it from large-scale DDoS attacks.
* How would you use AWS Shield Advanced to mitigate these attacks?

129. Scenario: You need to provide cross-account access for a third-party auditor to review CloudTrail logs without granting full administrative access.
* How would you set up an IAM role with a trust policy for cross-account access?

130. Scenario: You are establishing a new AWS account structure for your organization.
* How would you use AWS Organizations and AWS Control Tower to set up a secure and compliant multi-account environment from the start?

V. High Availability, Scalability & Disaster Recovery (30 Questions)

131. Scenario: Your single-instance web application is experiencing downtime during traffic spikes. You need to improve its availability and scalability.
* How would you re-architect the application to be highly available across multiple Availability Zones (AZs)?
* Which AWS services would you use for load balancing and automatic scaling?

132. Scenario: Your Amazon RDS instance is a single point of failure. You need to ensure database high availability.
* How would you configure RDS for multi-AZ deployment?
* What are the implications for failover and application connectivity?

133. Scenario: Your application's traffic is highly unpredictable, with sudden, massive spikes. You need to ensure your EC2 instances can handle these spikes.
* How would you configure Auto Scaling Groups with predictive scaling or target tracking policies?
* What metrics would you use to drive scaling decisions?

134. Scenario: Your company requires a disaster recovery plan with a low Recovery Time Objective (RTO) and Recovery Point Objective (RPO) for a critical application.
* Describe a multi-region active-passive (pilot light or warm standby) or active-active disaster recovery strategy using AWS.
* Which AWS services (Route 53, S3 Cross-Region Replication, RDS Read Replicas/Multi-AZ) would be involved?

135. Scenario: You need to ensure your application can recover quickly from a regional outage.

* How would you implement cross-region data replication for your S3 buckets and DynamoDB tables?

136. Scenario: Your application is read-heavy, and your RDS instance is struggling to keep up with the query load.
* How would you use RDS Read Replicas to offload read traffic and improve performance?
* What considerations would you have for eventual consistency?

137. Scenario: You are designing a serverless application using AWS Lambda, and you need to ensure its scalability under high load.
* How does Lambda inherently scale?
* What are cold starts, and how would you mitigate their impact (e.g., provisioned concurrency)?

138. Scenario: Your application serves global users, and you need to reduce latency and improve content delivery speed.
* How would you use Amazon CloudFront (CDN) to cache static and dynamic content closer to your users?
* What are the benefits of using CloudFront with S3?

139. Scenario: Your application requires a shared file system accessible by multiple EC2 instances. You need high availability for this file system.
* How would you use Amazon EFS (Elastic File System) across multiple Availability Zones?

140. Scenario: You are migrating a stateful application from on-premises to AWS. You need to ensure session stickiness for load-balanced traffic.
* How would you configure session stickiness on an Application Load Balancer (ALB)?
* What are the pros and cons of using session stickiness?

141. Scenario: You need to distribute incoming application traffic across multiple regions for global availability and disaster recovery.
* How would you use Amazon Route 53 with latency-based routing, geolocation routing, or failover routing policies?

142. Scenario: Your application stores large files (e.g., videos, images) that need to be highly available and durable.
* How would you use Amazon S3 for this purpose, including its various storage classes and replication options?

143. Scenario: You have a batch processing application that runs only during off-peak hours and can tolerate interruptions.
* How would you leverage AWS Spot Instances within an Auto Scaling Group to reduce costs for this workload while maintaining availability guarantees?

144. Scenario: Your application uses a message queue for asynchronous processing. You need to ensure the queue is highly available and scalable.
* How would you use Amazon SQS (Standard vs. FIFO) and integrate it with your application?

145. Scenario: You need to design a highly available Kubernetes cluster on AWS.
* How would you configure Amazon EKS to span multiple Availability Zones and ensure control plane and data plane high availability?

146. Scenario: Your application experiences sudden failures of individual EC2 instances.
* How would Auto Scaling Groups automatically replace unhealthy instances?
* What health checks would you configure?

147. Scenario: You need to build a caching layer for your read-heavy database to improve application performance and reduce database load.
* How would you use Amazon ElastiCache (Redis or Memcached) for this purpose?
* What are the considerations for cache invalidation?

148. Scenario: You are running a mission-critical legacy application on a single EC2 instance that cannot be easily containerized or refactored.
* How would you ensure its high availability using EC2 auto-recovery or other methods, despite its limitations?

149. Scenario: Your application has a microservices architecture, and you need a highly available and scalable service discovery mechanism.
* How would you use AWS Cloud Map for service discovery?

150. Scenario: You need to handle distributed cron jobs or scheduled tasks reliably and scalably.
* How would you use AWS EventBridge (CloudWatch Events) to trigger Lambda functions or other targets on a schedule?

151. Scenario: Your application needs to store highly available and durable object data that is frequently accessed.
* Which S3 storage class would you choose (Standard, Standard-IA, One Zone-IA) and why?

152. Scenario: You're using an Application Load Balancer, and you need to route traffic to different target groups based on URL paths or host headers.
* How would you configure ALB listener rules for path-based or host-based routing?

153. Scenario: You need to ensure zero downtime during database schema changes for your RDS instance.
* What strategies (e.g., blue/green deployment for RDS, logical replication, specific migration tools) would you consider?

154. Scenario: Your application processes real-time streaming data, and you need a highly available and scalable streaming data service.
* How would you use Amazon Kinesis Data Streams for this purpose, including sharding and consumer groups?

155. Scenario: You need to recover specific files or directories from an EBS volume snapshot after an accidental deletion.
* How would you attach the snapshot as a new volume and recover the data?

156. Scenario: You want to perform load testing on your application to identify bottlenecks and ensure it can handle expected traffic.
* What AWS services or third-party tools would you use for load generation and performance monitoring?

157. Scenario: Your application has varying workloads throughout the day, and you want to scale your database dynamically.
* How would you leverage Amazon Aurora Serverless for automatic database scaling?

158. Scenario: You need to provide a highly available and scalable managed DNS service for your public-facing applications.
* How would you use Amazon Route 53, including health checks and different routing policies?

159. Scenario: You need to distribute large software updates to a fleet of edge devices or IoT devices.
* How would you use AWS IoT Greengrass or AWS Device Farm for this, ensuring reliable and secure delivery?

160. Scenario: You are designing a new application that needs extreme low-latency access to data, with high throughput.
* How would you consider using a distributed caching layer (e.g., ElastiCache for Redis) and data partitioning strategies?

VI. Cost Optimization (15 Questions)

161. Scenario: Your AWS bill is increasing rapidly, and you need to identify areas for cost optimization for your EC2 instances.
* What strategies would you employ (e.g., right-sizing, reserved instances, spot instances, scheduling)?
* Which AWS tools would you use for cost analysis?

162. Scenario: You have a large number of S3 buckets storing various types of data. You want to optimize S3 storage costs.
* How would you use S3 Intelligent-Tiering or S3 lifecycle policies to move data to lower-cost storage classes (e.g., S3 Standard-IA, S3 Glacier)?

163. Scenario: Your RDS database is running 24/7, but it's only heavily utilized during business hours.
* How would you optimize its cost (e.g., Aurora Serverless, Reserved Instances, stopping/starting non-production instances)?

164. Scenario: You have many EC2 instances that are idle during nights and weekends.
* How would you automate the stopping and starting of these non-production instances to save costs? (Hint: AWS Lambda, CloudWatch Events)

165. Scenario: You are seeing high data transfer costs between your EC2 instances and other AWS services.
* How would you analyze and optimize inter-service data transfer costs (e.g., VPC endpoints, colocation within AZs)?

166. Scenario: Your Lambda function costs are higher than expected due to long execution times.
* How would you optimize Lambda function performance and memory allocation to reduce costs?

167. Scenario: Your organization is committed to using a certain amount of EC2 capacity over the next year.
* How would you use EC2 Reserved Instances or Savings Plans to reduce costs for this committed usage?

168. Scenario: You have multiple AWS accounts, and you want to get a consolidated view of your costs and implement cost allocation.
* How would you use AWS Organizations, Cost Explorer, and tagging for consolidated billing and cost allocation?
169. Scenario: Your EBS volumes have high I/O operations but are attached to low-performance instance types, leading to underutilization of provisioned IOPS.
* How would you right-size your EBS volumes and instance types to optimize cost and performance?
170. Scenario: You are using Amazon ECR for Docker image storage, and you have many old, unused images.
* How would you implement ECR lifecycle policies to automatically delete old images and reduce storage costs?
171. Scenario: Your Application Load Balancer (ALB) is configured, but you suspect it's over-provisioned or not being utilized efficiently.
* How would you monitor ALB metrics and potentially optimize its configuration or consider alternative load balancing strategies?
172. Scenario: You want to identify and terminate unused or idle AWS resources across your accounts to save costs.
* What strategies and tools (e.g., AWS Config, custom scripts, third-party tools) would you use for resource cleanup?
173. Scenario: Your CloudFront distribution has high data transfer out costs.
* How would you optimize CloudFront costs (e.g., caching strategies, compression, origin shield)?
174. Scenario: You need to convince management to invest in a cost optimization initiative.
* How would you demonstrate the potential cost savings using AWS Cost Explorer reports and historical data?
175. Scenario: Your application uses DynamoDB, and you're seeing high costs for read/write capacity.
* How would you optimize DynamoDB costs (e.g., on-demand vs. provisioned capacity, right-sizing capacity, leveraging DAX)?

VII. Containerization & Orchestration (15 Questions)

176. Scenario: You have a legacy application running on EC2 instances, and you want to containerize it and move it to a managed container service.
* Would you choose Amazon ECS or EKS, and why?
* How would you containerize the application (Dockerfile creation)?
177. Scenario: You need to deploy a microservices application using Docker containers on AWS, and you want a serverless compute option for your containers.
* How would you use AWS Fargate with either ECS or EKS?
* What are the benefits of Fargate in terms of operational overhead?
178. Scenario: Your containerized application needs to store persistent data that can be accessed by multiple containers.

* How would you manage persistent storage for containers on ECS or EKS (e.g., EFS, EBS CSI driver)?

179. Scenario: You are deploying a containerized application to Amazon ECS, and you need to manage secret injection into your containers.
    * How would you use Secrets Manager or Parameter Store with ECS task definitions for secure secret delivery?
180. Scenario: You need to scale your containerized application automatically based on custom metrics (e.g., messages in an SQS queue).
    * How would you configure Auto Scaling for ECS services or EKS deployments based on custom metrics using CloudWatch?
181. Scenario: You want to implement a service mesh for your microservices running on EKS to gain capabilities like traffic management, mTLS, and observability.
    * How would you deploy AWS App Mesh or Istio on your EKS cluster?
182. Scenario: Your containerized application on ECS needs to communicate with an RDS database in a private subnet.
    * How would you configure the networking (VPC, security groups) for your ECS tasks to securely access the database?
183. Scenario: You are troubleshooting a containerized application that is failing to start on ECS.
    * What steps would you take to diagnose the issue (e.g., checking task logs, task events, security group rules)?
184. Scenario: You need to automate the build and push of Docker images to a private registry.
    * How would you use AWS CodeBuild to build your Docker images and push them to Amazon ECR?
185. Scenario: Your Kubernetes pods on EKS need to assume specific IAM roles to access other AWS services.
    * How would you implement IAM Roles for Service Accounts (IRSA) on EKS?
186. Scenario: You have a legacy application that can't be easily containerized due to its dependencies on the host OS.
    * What alternatives would you consider besides containers (e.g., EC2, Elastic Beanstalk)?
187. Scenario: You need to run scheduled tasks or batch jobs as containers.
    * How would you use ECS Scheduled Tasks or Kubernetes CronJobs on EKS?
188. Scenario: You are moving from a single Dockerfile to a multi-stage Dockerfile for optimizing image size.
    * How would this impact your CodeBuild configuration for building images?
189. Scenario: Your EKS cluster needs to integrate with AWS Load Balancer Controller (formerly ALB Ingress Controller) for exposing services via an ALB.
    * How would you deploy and configure the AWS Load Balancer Controller in your EKS cluster?
190. Scenario: You want to implement container health checks to ensure only healthy containers receive traffic.

* How would you define readiness and liveness probes in your Kubernetes deployments or ECS task definitions?
