## Module 1: Advanced Introduction to ELK Stack

1. Explain the CAP theorem as it relates to Elasticsearch clustering, and discuss how you would mitigate the risks of partition tolerance impacting availability.
2. For a high-throughput log analytics system, what are the possible bottlenecks in the ELK stack, and how would you monitor and resolve them?
3. In a regulated environment (e.g., PCI DSS), what ELK architectural decisions must be made regarding data retention and auditability?
4. Design a multi-region ELK deployment: what are the primary concerns, and how would you address cross-region data consistency?
5. How can you secure inter-component (Logstash, Elasticsearch, Kibana) communications in an untrusted network? List specific protocols and configuration mechanisms.

## Module 2: ELK Stack Advanced Setup and Administration

6. Describe the process of automating full ELK stack deployment (Elasticsearch, Logstash, Kibana) using configuration management tools (e.g., Ansible, Puppet, Chef).
7. When performing a rolling upgrade of Elasticsearch, what are the steps and precautions to ensure zero downtime and data consistency?
8. Discuss best practices for configuring Elasticsearch heap space and JVM options for nodes with differing roles (master, data, ingest, etc.).
9. Explain the implications of split-brain scenarios in Elasticsearch clusters and detail your recovery strategy.
10. How would you implement centralized authentication and access control across the ELK stack components, compatible with Active Directory/LDAP?
11. Which monitoring tools or APIs would you use to detect and diagnose cluster performance degradation before it becomes a crisis?
12. How would you automate hot-warm-cold index lifecycle management, and what benefits does this bring to company operations?
13. DevOps often use Docker and Kubernetes. Outline the key configuration steps for deploying a resilient, auto-scaling ELK stack on Kubernetes.
14. After a sudden disk failure on an Elasticsearch node, what steps would you take to restore service integrity and avoid further data loss?
15. Discuss how index sharding and replication factors should be tuned for a cluster expected to handle petabyte-scale log ingestion.

## Module 3: Logstash – Advanced Data Ingestion & Parsing

16. Given a scenario where Logstash requires ingesting logs from distributed, unreliable sources, how would you ensure pace control and data integrity?
17. Compare and contrast persistent queues and dead letter queues in Logstash. When would you use each, and how would you configure them?
18. Describe the impact and mitigation strategies of a memory or CPU spike caused by a poorly written Logstash filter plugin.
19. When ingesting terabytes of daily log data with both Logstash and Filebeat, how would you architect the data flow for maximum resiliency and efficiency?
20. How would you design a Logstash pipeline to handle multi-format log ingestion (e.g., mixed JSON, CSV, and syslog), and what is the trade-off in pipeline complexity vs. maintainability?
21. Explain how to use conditionals and mutation filters in Logstash to normalize data fields across different input formats.
22. A third-party Logstash plugin fails during pipeline startup. What diagnostic steps do you perform, and how can you safeguard production pipelines?
23. You suspect a Logstash pipeline introduces latency. Detail a step-by-step method for isolating and resolving bottlenecks.
24. Describe how to design Logstash pipelines for at-least-once and exactly-once data delivery semantics.
25. Discuss plugin version compatibility; how would you manage and test plugin upgrades to avoid downtime or data loss?

## Module 4: Elasticsearch – Scaling, Performance, and Operations

26. How do you mitigate mapping explosion in dynamic log environments, and what are the operational impacts if not managed?
27. Elasticsearch is reporting frequent circuit breaker exceptions. What do these mean, and how do you address them short and long term?
28. Describe a backup and disaster recovery plan for production Elasticsearch indices under 24/7 SLA constraints.
29. Given high cardinality aggregations on large datasets, what query, index, and hardware optimizations can you perform?
30. Explain the role and configuration of ILM (Index Lifecycle Management) for indices with unpredictable data retention requirements.
31. A developer complains that search results are stale. How could index refresh settings or replica lag contribute, and how do you adjust these settings?
32. Explain how you would reindex a petabyte-scale index with zero downtime, and the challenges you anticipate.

33. Describe how to enable encryption-at-rest for Elasticsearch indices and the operational implications.
34. Under log surges (e.g., DDoS attack), what measures protect indexing throughput and ensure search performance?

## Module 5: Kibana – Security, Visualizations, and Dashboards

35. If Kibana's dashboards are timing out due to slow Elasticsearch queries, how do you debug and optimize both the query and the dashboard?
36. A sensitive dashboard must be accessible only to select users. How would you implement role-based access in Kibana?
37. Describe the best way to create parameterized (DRY) visualizations in Kibana for use in multiple dashboards.
38. DevOps teams want a real-time infrastructure health overview. How would you architect Kibana dashboards for actionable monitoring and alerting?
39. Custom plugins are needed for new visualization types. Describe the process and deployment pipeline you would set up for Kibana plugin development and updates.
40. How do you securely expose Kibana to external users while protecting Elasticsearch from direct public access?
41. A compliance auditor requests a change history for dashboards and visualizations. What options does Kibana offer for auditability, and how would you supplement if native options are limited?
42. Explain the use of Kibana's Canvas or Lens for visualizing custom KPIs, and how you'd integrate them with external data sources.

## Module 6: Troubleshooting, Upgrades, and Advanced Operations

43. Describe your strategy for blue-green or canary deployments of new ELK configurations or plugins without user disruption.
44. Elasticsearch's JVM heap usage keeps growing and triggers out-of-memory errors. Walk through your diagnostic and remediation steps.
45. Network partition occurs between Kibana and some Elasticsearch nodes. What symptoms appear in user dashboards, and how do you resolve?
46. A recurring complaint is slow ingest or delayed dashboards during specific business hours. How would you investigate and present findings to management?
47. During an upgrade, an Elasticsearch index becomes read-only due to disk watermark. What steps do you take to resume indexing?

48. Describe how you would set up CI/CD pipelines for Logstash configuration management and rollout, handling secrets securely.
49. If a developer wants production-like ELK stack test data for local debugging, how do you provide it safely and cost-effectively?
50. For compliance, you're required to export all ELK configurations and mappings regularly. How do you automate this task?