14 Advanced Topics & Integrations

Advanced Topics & Integrations

Basic Questions (10 Questions)

CI/CD with Azure DevOps

1. **What is CI/CD and why is it important for data engineering projects?**
   - Expected: Understanding of Continuous Integration/Continuous Deployment, automated testing, deployment pipelines, and reduced manual errors.
2. **What are Azure DevOps Pipelines and how do they differ from traditional deployment methods?**
   - Expected: YAML-based pipelines, automated triggers, integration with Git repositories, and environment management.
3. **Name three key components of Azure DevOps that are essential for data engineering workflows.**
   - Expected: Azure Repos (Git), Azure Pipelines, Azure Artifacts, with brief explanations of each.

Terraform with Databricks

4. **What is Infrastructure as Code (IaC) and why would you use Terraform for Databricks deployments?**
   - Expected: Version control for infrastructure, reproducible environments, declarative configuration management.
5. **What is a Terraform provider and which provider would you use for Databricks?**
   - Expected: Understanding of providers as plugins, specifically databricks/databricks provider for managing Databricks resources.
6. **What are the basic Terraform commands you would use in a typical workflow?**
   - Expected: terraform init, terraform plan, terraform apply, terraform destroy, and their purposes.

Observability, Logging and Monitoring

7. **What is the difference between logging, monitoring, and observability?**
   - Expected: Logging captures events, monitoring tracks metrics, observability provides insights into system behavior through logs, metrics, and traces.

8. **Name three types of data you would typically monitor in a data pipeline.**
   - Expected: Data quality metrics, pipeline performance metrics, resource utilization, error rates, data freshness.
9. **What is a dashboard and why is it important for data engineering operations?**
   - Expected: Visual representation of metrics, real-time monitoring, alerting capabilities, stakeholder communication.
10. **What are some common log levels and when would you use each?**
    - Expected: DEBUG, INFO, WARN, ERROR, FATAL with appropriate use cases for each level.

Intermediate Questions (10 Questions)

CI/CD with Azure DevOps

11. **How would you implement automated testing for data pipelines in Azure DevOps?**
    - Expected: Unit tests, integration tests, data quality tests, using pytest or similar frameworks, test stages in YAML pipelines.
12. **Explain the concept of environment promotion in Azure DevOps for data engineering projects.**
    - Expected: Dev/Test/Prod environments, deployment gates, approval processes, environment-specific configurations.
13. **How would you handle secrets and sensitive configuration in Azure DevOps pipelines?**
    - Expected: Azure Key Vault integration, variable groups, secure variables, service connections.

Terraform with Databricks

14. **How would you structure a Terraform project for managing multiple Databricks environments?**
    - Expected: Workspaces, modules, environment-specific tfvars files, remote state management.
15. **Explain how you would manage Databricks cluster configurations using Terraform.**
    - Expected: databricks_cluster resource, node types, autoscaling, spot instances, cluster policies.
16. **What are Terraform modules and how would you create a reusable module for Databricks job deployment?**

- Expected: Module structure, input/output variables, versioning, calling modules from root configurations.

Observability, Logging and Monitoring

17. **How would you implement end-to-end monitoring for a data pipeline that processes data from source to analytics?**
    - Expected: Data lineage tracking, SLA monitoring, data quality checks, alerting thresholds, custom metrics.
18. **Explain how you would set up centralized logging for distributed data processing jobs.**
    - Expected: Log aggregation tools (ELK stack, Splunk), structured logging, correlation IDs, log retention policies.
19. **What metrics would you track for a Databricks cluster and how would you collect them?**
    - Expected: CPU/Memory utilization, job duration, queue times, cost metrics, using Spark UI, Databricks metrics, custom instrumentation.
20. **How would you implement alerting for data pipeline failures and what information should be included?**
    - Expected: Alert channels (email, Slack, PagerDuty), escalation policies, contextual information, runbook links.

Difficult Questions (10 Questions)

CI/CD with Azure DevOps

21. **Design a complete CI/CD strategy for a complex data platform with multiple data sources, transformation layers, and analytics outputs. Include branching strategy, testing approach, and deployment methodology.**
    - Expected: GitFlow or GitHub Flow, feature branches, automated testing pyramid, blue-green deployments, canary releases, rollback strategies.
22. **How would you implement cross-environment data validation and reconciliation in your CI/CD pipeline?**
    - Expected: Data diff tools, statistical validation, schema comparison, automated reconciliation reports, integration with pipeline gates.
23. **Explain how you would handle database schema migrations and data transformations in a CI/CD pipeline with zero downtime requirements.**
    - Expected: Backward compatible changes, database versioning, feature flags, gradual rollouts, rollback procedures.

Terraform with Databricks

24. **Design a Terraform architecture for managing a multi-tenant Databricks platform with proper isolation, cost allocation, and governance.**
    - Expected: Workspace per tenant, shared infrastructure, RBAC implementation, cost tracking tags, policy enforcement.
25. **How would you implement Terraform state management for a large-scale Databricks deployment across multiple regions and environments?**
    - Expected: Remote state backends, state locking, workspace separation, state file encryption, disaster recovery.
26. **Explain how to implement drift detection and remediation for Databricks infrastructure managed by Terraform.**
    - Expected: terraform plan automation, drift detection scripts, automated remediation vs manual review, compliance reporting.

Observability, Logging and Monitoring

27. **Design a comprehensive observability strategy for a real-time streaming data platform that processes millions of events per second.**
    - Expected: Distributed tracing, sampling strategies, metrics aggregation, real-time alerting, capacity planning, SLI/SLO definition.
28. **How would you implement cost optimization monitoring and automated cost control for cloud-based data processing workloads?**
    - Expected: Resource tagging, cost allocation, automated scaling policies, spot instance management, cost anomaly detection.
29. **Explain how you would build a data lineage and impact analysis system that integrates with your monitoring and alerting infrastructure.**
    - Expected: Metadata extraction, dependency graphs, impact propagation, automated data quality assessment, stakeholder notification.
30. **Design a disaster recovery and business continuity monitoring system for critical data pipelines. Include RTO/RPO considerations and automated failover mechanisms.**
    - Expected: Multi-region deployment, data replication strategies, automated health checks, failover procedures, communication protocols, compliance requirements.