04 Databricks Basics

Databricks Workspace, Clusters & Notebooks

Basic Level Questions (1-8)

1. What is Databricks and how does it differ from traditional Apache Spark installations?

What to look for: Understanding of managed Spark service, collaborative workspace, auto-scaling, and integrated analytics platform.

2. Explain the main components of a Databricks workspace.

What to look for: Workspace browser, notebooks, clusters, jobs, libraries, and data exploration tools.

3. What are the different types of clusters available in Databricks?

What to look for: All-purpose clusters, job clusters, pools, and their respective use cases and cost implications.

4. What is the difference between a driver node and worker nodes in a Databricks cluster?

What to look for: Driver coordination role, worker task execution, resource allocation, and communication patterns.

5. How do you create and configure a basic cluster in Databricks?

What to look for: Cluster creation process, node types, autoscaling settings, and runtime selection.

6. What are Databricks notebooks and what languages do they support?

What to look for: Interactive development environment, Python, Scala, SQL, R support, and collaborative features.

7. Explain the concept of Databricks Runtime and its versions.

What to look for: Pre-configured Spark environments, ML runtime, genomics runtime, and version compatibility.

8. How do you share notebooks and collaborate with team members in Databricks?

What to look for: Workspace permissions, notebook sharing, version control integration, and commenting features.

Intermediate Level Questions (9-17)

9. How would you optimize cluster performance for different types of workloads?

What to look for: Instance types selection, cluster sizing, memory optimization, and workload-specific configurations.

10. Explain the difference between Standard and High Concurrency clusters and when to use each.

What to look for: Isolation levels, security features, resource sharing, and multi-user scenarios.

11. How do you manage libraries and dependencies in Databricks clusters?

What to look for: Cluster libraries, init scripts, Maven coordinates, PyPI packages, and environment management.

12. What are cluster pools and how do they help with cost and performance optimization?

What to look for: Pre-warmed instances, reduced startup time, cost savings, and pool management strategies.

13. How would you implement proper logging and monitoring for Databricks clusters?

What to look for: Spark UI, cluster logs, metrics collection, and integration with monitoring tools.

14. Explain how you would handle secrets management in Databricks notebooks.

What to look for: Databricks secrets, Azure Key Vault integration, scope management, and security best practices.

15. How do you debug performance issues in Databricks notebooks and clusters?

What to look for: Spark UI analysis, query execution plans, bottleneck identification, and optimization techniques.

16. What are the best practices for organizing notebooks and workspace structure?

What to look for: Folder hierarchy, naming conventions, notebook templates, and project organization.

17. How would you implement version control and CI/CD for Databricks notebooks?

What to look for: Git integration, Databricks CLI, automated deployments, and environment promotion strategies.

Advanced/Difficult Level Questions (18-25)

18. Design a multi-environment Databricks setup (dev, staging, production) with proper governance.

What to look for: Workspace separation, access controls, deployment automation, and configuration management.

19. How would you implement auto-scaling strategies for unpredictable workloads in Databricks?

What to look for: Cluster policies, autoscaling algorithms, cost optimization, and performance monitoring.

20. Explain how you would design a notebook architecture for complex ETL pipelines with error handling.

What to look for: Modular design, exception handling, logging frameworks, and pipeline orchestration patterns.

21. How would you implement fine-grained access control and security policies across multiple teams?

What to look for: Workspace access control, cluster policies, table access control, and audit logging.

22. Design a solution for managing compute resources across multiple business units with cost allocation.

What to look for: Cluster policies, resource tagging, cost monitoring, chargeback mechanisms, and governance frameworks.

23. How would you optimize Databricks for streaming workloads with low latency requirements?

What to look for: Structured streaming configuration, cluster optimization, checkpointing strategies, and monitoring.

24. Explain how you would implement disaster recovery and high availability for critical Databricks workloads.

What to look for: Multi-region deployment, backup strategies, failover mechanisms, and data replication.

25. How would you design a notebook-based ML pipeline with proper model lifecycle management?

What to look for: MLflow integration, experiment tracking, model versioning, and automated deployment patterns.

Technical Deep-Dive Scenarios

Scenario A: Performance Troubleshooting

"Your Databricks job is running slowly and consuming excessive memory. Walk me through your diagnostic and optimization approach."

Scenario B: Cost Optimization

"Your monthly Databricks bill has increased by 300%. How would you investigate and optimize costs?"

Scenario C: Multi-Team Environment

"Design a Databricks environment for 50+ data scientists and engineers across 5 different teams with varying security requirements."

Scenario D: Production Pipeline

"Your production ETL pipeline in Databricks needs 99.9% availability. How would you architect this?"

Integration Questions

Azure Integration

- "How would you integrate Databricks with Azure Data Factory and Azure Synapse?"
- "Explain your approach to accessing Azure Storage from Databricks securely."

Data Lake Integration

- "How would you optimize Databricks for processing data stored in Delta Lake?"
- "Explain your strategy for handling schema evolution in notebook-based pipelines."

Follow-Up Questions:

- "How would you handle notebook timeouts in long-running processes?"
- "What's your approach to testing notebooks before production deployment?"
- "How would you implement data quality checks within notebooks?"
- "Explain your strategy for managing notebook dependencies across environments."
- "How would you monitor and alert on notebook execution failures?"

Real-World Problem Solving:

- "A data scientist's notebook works in their environment but fails in production. How do you troubleshoot?"
- "Your cluster keeps running out of memory during peak hours. What's your approach?"
- "Management wants to reduce Databricks costs by 40% without impacting performance. How do you achieve this?"
- "You need to provide Databricks access to external contractors while maintaining security. How do you set this up?"