1. Arquitetura e Configuração:

Descreva a arquitetura geral de uma implementação típica do Databricks. Como você configuraria um ambiente Databricks para otimizar o desempenho e a escalabilidade?

Arquitetura típica do Databricks inclui armazenamento distribuído (DBFS), clusters de computação, camadas de armazenamento de dados (Bronze, Silver, Gold) e hospedagem em provedores de nuvem.

Componentes principais: Cluster de Computação, Apache Spark, Databricks Runtime, Workspace.

Para otimizar desempenho e escalabilidade:

Dimensionamento do cluster conforme necessidade de processamento.

Ajuste de executores e memória no Databricks Runtime.

Particionamento de dados adequado, preferencialmente por data.

Utilização de Cache e Persistência do Spark para melhorar o desempenho.

2. Apache Spark e Databricks:

Como o Databricks se integra ao Apache Spark? Quais são as principais vantagens do uso do Databricks em comparação com uma instalação padrão do Apache Spark?

Databricks se integra ao Apache Spark como uma camada adicional que facilita desenvolvimento, implantação e gerenciamento de big data.

Principais vantagens do Databricks em comparação com uma instalação padrão do Apache Spark incluem Databricks Runtime, integração em notebook, gerenciamento automatizado de clusters.

3. Notebooks e Linguagens de Programação:

Explique como você usaria Notebooks no Databricks para criar e executar código. Além disso, como o Databricks suporta várias linguagens de programação, e como você decidiria qual linguagem usar em um projeto específico?

Eu criaria notebooks usando a interface do usuário ou usando comandos específicos, escreveria o código conforme linguagem adequada, a execução executaria as células do Notebook, para testar e validar código, analisaria os resultados e iteraria sobre o código conforme necessidade.

A escolha da linguagem depende do projeto específico, incluindo requisitos do projeto, compatibilidade com bibliotecas e ferramentas necessárias, experiência da equipe, sobre desempenho e escalabilidade eu gosto particularmente de trabalhar com pyspark, que você já deixa a arquitetura pronta para o BIGDATA se não for, mas por exemplo, podemos usar python puro tendo em vista prototipagem rápida, enquanto Scala pode ser mais adequado para processamento em larga scala, por isso para evitar esses cenários específicos eu uso com frequência pyspark.

4. Integração de Fontes de Dados:

Como o Databricks facilita a integração com diferentes fontes de dados, como Data Lakes, bancos de dados relacionais e fontes externas? Você pode fornecer um exemplo prático de como lidar com essas integrações?

Por meio de integrações nativas ou terceira, datalakes tradicionais (Azure lakestorage, S3), BD relacionais (SQL Database, Redshift), fontes externas como GCP ou salesforce, libraries e API's, conectores personalizados.

Para uma projeto de dados que usa o Azure Data Lake Storage e o SQL Database, faria uma conexão no ambiente do Databricks para acessar o DW e DL, usaria os conectores para ler os dados, dito isso processaria os dados um ETL, salvava os dados transformados, e realizaria análise exploratória, modelagem e armazenaria o resultado dos processamentos e análise de volta no DL ou DW ou outro sistema de armazamento.

5. Machine Learning no Databricks:

Descreva a abordagem que você seguiria para desenvolver e treinar modelos de machine learning no Databricks. Quais são as principais ferramentas e bibliotecas que você usaria para esse fim?

NÃO É MINHA ESPECIALIDADE, MAS NA MINHA EXPERIÊNCIA, segue a preparação dos dados, normalmente com Pandas, pyspark, exploração dos dados com matplotlib, seaborn e SparkSQL, seleção do modelo para determinado problema de negócio, treinamento do modelo, avaliação do desempenho com precisão, F1-Score, ajustar hipeparâmetros validação cruzada e por fim a implantação no próprio ambiente do Databricks, eu particularmente salvo os resultados em um DW para facilitar a análise.

6. Segurança e Controle de Acesso:

Como o Databricks aborda questões de segurança e controle de acesso? Quais são as práticas recomendadas para garantir a proteção dos dados e ambientes de desenvolvimento?

IAM ou AAD, controles de acessos por função, Políticas de acessos, permissões granulares, auditoria e monitoramento (Uso muito detecção de anomalias para ajudar), dados criptografados em transito e em repouso, configurar VPC, atualização dos patches de SI.

7. Desafios e Soluções:

Pergunta: Conte-nos sobre um desafio específico que você enfrentou ao trabalhar com Databricks e como o resolveu. Qual foi a solução implementada e quais foram os resultados alcançados?

No projeto com Databricks, enfrentamos o desafio de lidar com grandes volumes de dados de forma eficiente. Implementamos uma arquitetura distribuída utilizando clusters gerenciados pelo Databricks, otimizando consultas e garantindo confiabilidade com pipelines automatizados pelo Databricks Delta para ETL (Extração, Transformação e Carga). Os resultados incluíram processamento eficiente de grandes volumes de dados, tomada de decisão mais rápida e precisa, redução de custos e aumento da satisfação do cliente.

Conjunto de Dados do Kaggle: Não usei o Kaggle gerei os dados através do python (FAKER)

Escolha um conjunto de dados do Kaggle relacionado a vendas. Certifique-se de que o conjunto

de dados inclui informações como datas, produtos, quantidades vendidas, etc.

Projeto de Engenharia de Dados:

https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/999120970326964/3273897128718224/2452831531558393/latest.html

Ingestão e Carregamento de Dados:

Carregue o conjunto de dados no Databricks.

Explore o esquema dos dados e faça ajustes conforme necessário.

Transformações de Dados:

Realize transformações necessárias, como tratamento de valores nulos, conversões de tipos,

etc.

Adicione uma coluna calculada, por exemplo, o valor total de cada transação.

Agregue os dados para obter estatísticas de vendas, por exemplo, o total de vendas por produto ou por categoria.

Introduza uma regra mais complexa, como identificar padrões de comportamento de compra ao longo do tempo ou criar categorias personalizadas de produtos com base em determinados critérios.

Saída em Parquet e Delta:

Grave os dados transformados e agregados em um formato Parquet para persistência eficiente.

Grave os mesmos dados em formato Delta Lake para aproveitar as funcionalidades de versionamento e transações ACID.

Exploração Adicional (Opcional):

Execute consultas exploratórias para entender melhor os dados e validar as transformações.

Crie visualizações ou relatórios para comunicar insights.

Agende o notebook para execução automática em intervalos regulares para garantir a atualização contínua dos dados.