### Mindset/Brainstorm para Redução de Custos em Jobs de Streaming no Azure e Databricks

#### \*\*1. Otimização de Armazenamento:\*\*

- \*\*Compactação de Dados:\*\*

- Utilizar formatos compactados (Parquet, ORC)

- Implementar compactação diretamente no código do Databricks

- Automatizar o processo de compactação para dados de streaming

- \*\*Tiered Storage:\*\*

- Mover dados antigos para camadas de armazenamento mais baratas

- Configurar políticas de ciclo de vida no Azure Blob Storage

- Automação da movimentação de dados conforme envelhecem

#### \*\*2. Gerenciamento de Dados:\*\*

- \*\*Políticas de Retenção de Dados:\*\*

- Definir e aplicar políticas de retenção de dados

- Utilizar Azure Data Lake Storage para automação de retenção

- Revisão periódica das políticas de retenção para otimização contínua

- \*\*Deduplicação de Dados:\*\*

- Implementar deduplicação em pipelines de streaming

- Ferramentas e scripts para identificação e remoção de duplicatas

- Monitorar e ajustar processos de deduplicação regularmente

#### \*\*3. Otimização de Consultas:\*\*

- \*\*Particionamento de Dados:\*\*

- Implementar particionamento em dados de streaming

- Escolher colunas de partição adequadas para melhorar desempenho

- Monitorar e ajustar partições conforme necessário

- \*\*Cache de Consultas:\*\*

- Utilizar caching no Databricks para melhorar eficiência

- Configurar e gerenciar cache de forma automática

- Monitorar uso de cache e ajustar estratégias conforme necessário

#### \*\*4. Gerenciamento de Recursos:\*\*

- \*\*Auto-Scaling:\*\*

- Configurar auto-scaling para clusters no Databricks

- Ajustar parâmetros de auto-scaling para otimização de custos

- Monitorar e revisar a eficácia do auto-scaling regularmente

- \*\*Cluster Size Optimization:\*\*

- Ajustar tamanho de clusters de acordo com a carga de trabalho

- Revisão contínua das necessidades de recursos

- Implementar automação para ajustes de tamanho de cluster

#### \*\*5. Governança e Monitoramento:\*\*

- \*\*Tagging e Monitoramento:\*\*

- Adicionar tags para monitoramento e controle de custos

- Implementar políticas de tagging em toda a infraestrutura

- Monitorar recursos usando tags e ajustar conforme necessário

- \*\*Alertas de Custo:\*\*

- Configurar alertas de custo no Azure Portal

- Revisão e ajuste contínuo dos limites de alerta

- Implementar automação para resposta a alertas de custo

#### \*\*6. Otimização de Workloads:\*\*

- \*\*Job Scheduling:\*\*

- Agendar jobs de streaming para horários de menor custo

- Ajustar agendamento conforme a carga de trabalho e custos

- Monitorar a eficiência do agendamento e otimizar continuamente

- \*\*Parallelism and Concurrency:\*\*

- Ajustar paralelismo e concorrência em jobs de streaming

- Implementar políticas de ajuste automático conforme a demanda

- Monitorar e ajustar configurações regularmente

#### \*\*7. Utilização de Serviços Gerenciados:\*\*

- \*\*Utilização de Spot Instances:\*\*

- Utilizar instâncias spot para reduzir custos

- Configurar e gerenciar instâncias spot no Databricks

- Implementar tolerância a falhas para instâncias spot

- \*\*Azure Databricks Pools:\*\*

- Criar e utilizar pools de instâncias

- Configurar pools para reduzir tempo de inicialização e custos

- Monitorar e ajustar a utilização dos pools

#### \*\*8. Licenciamento e Planos:\*\*

- \*\*Revisão de Planos e Licenças:\*\*

- Revisar planos de assinatura e licenciamento regularmente

- Ajustar planos conforme necessário para otimização de custos

- Implementar revisões periódicas para manter a eficiência

#### \*\*9. Auditoria e Revisão Regular:\*\*

- \*\*Revisão Periódica de Custos:\*\*

- Utilizar ferramentas de custo para revisão regular

- Gerar e analisar relatórios de custo periodicamente

- Implementar mudanças baseadas nas análises de custo

- \*\*Feedback de Utilização:\*\*

- Coleção de feedback regular das equipes

- Realização de reuniões periódicas para discutir uso de recursos

- Implementar melhorias baseadas no feedback das equipes

Esses tópicos estruturados fornecem um guia claro e abrangente para a redução de custos em jobs de streaming no Azure e Databricks.

### Implementação Técnica das Ações de Redução de Custos em Jobs de Streaming no Azure e Databricks

#### \*\*1. Otimização de Armazenamento:\*\*

- \*\*Compactação de Dados:\*\*

- \*\*Código Databricks:\*\*

```python

stream\_df = spark.readStream.format("json").load("/path/to/input")

stream\_df.writeStream.format("parquet").option("path", "/path/to/output").start()

```

- \*\*Tiered Storage:\*\*

- \*\*Azure Portal:\*\*

1. Navegue até "Storage Accounts".

2. Selecione a conta de armazenamento desejada.

3. Vá para "Blobs" e selecione o contêiner.

4. Clique em "Change tier" e selecione "Cool" ou "Archive".

- \*\*Lifecycle Management:\*\*

1. Acesse "Storage Accounts".

2. Selecione a conta de armazenamento.

3. Clique em "Lifecycle Management" e adicione uma regra para mover dados antigos para Cool ou Archive.

#### \*\*2. Gerenciamento de Dados:\*\*

- \*\*Políticas de Retenção de Dados:\*\*

- \*\*Azure Portal:\*\*

1. Acesse "Storage Accounts".

2. Vá para "Lifecycle Management".

3. Crie regras para deletar ou mover dados com base na idade.

- \*\*Deduplicação de Dados:\*\*

- \*\*Código Databricks:\*\*

```python

stream\_df = spark.readStream.format("delta").load("/path/to/input")

deduplicated\_df = stream\_df.dropDuplicates(["key\_column"])

deduplicated\_df.writeStream.format("delta").option("path", "/path/to/output").start()

```

#### \*\*3. Otimização de Consultas:\*\*

- \*\*Particionamento de Dados:\*\*

- \*\*Código Databricks:\*\*

```python

stream\_df.writeStream.partitionBy("partition\_column").format("parquet").option("path", "/path/to/output").start()

```

- \*\*Cache de Consultas:\*\*

- \*\*Configuração Databricks:\*\*

```python

spark.conf.set("spark.databricks.io.cache.enabled", "true")

stream\_df = spark.readStream.format("delta").load("/path/to/input")

```

#### \*\*4. Gerenciamento de Recursos:\*\*

- \*\*Auto-Scaling:\*\*

- \*\*Databricks:\*\*

1. Acesse "Clusters".

2. Clique em "Create Cluster".

3. Ative "Enable Auto-Scaling" e configure os parâmetros mínimos e máximos.

- \*\*Cluster Size Optimization:\*\*

- \*\*Databricks:\*\*

1. Acesse "Clusters".

2. Ajuste o tamanho e as configurações do cluster conforme a carga de trabalho.

#### \*\*5. Governança e Monitoramento:\*\*

- \*\*Tagging e Monitoramento:\*\*

- \*\*Azure Portal:\*\*

1. Acesse "Cost Management + Billing".

2. Vá para "Tags" e adicione tags aos recursos.

- \*\*Monitoramento:\*\*

1. Use "Cost Management" > "Cost analysis" para monitorar custos por tag.

- \*\*Alertas de Custo:\*\*

- \*\*Azure Portal:\*\*

1. Acesse "Cost Management + Billing".

2. Vá para "Budgets".

3. Clique em "Create Budget" e configure alertas.

#### \*\*6. Otimização de Workloads:\*\*

- \*\*Job Scheduling:\*\*

- \*\*Databricks:\*\*

1. Acesse "Jobs".

2. Clique em "Create Job".

3. Configure o job e defina a agenda de execução para horários de menor custo.

- \*\*Parallelism and Concurrency:\*\*

- \*\*Código Databricks:\*\*

```python

spark.conf.set("spark.sql.shuffle.partitions", "num\_partitions")

stream\_df.writeStream.format("delta").option("path", "/path/to/output").start()

```

#### \*\*7. Utilização de Serviços Gerenciados:\*\*

- \*\*Utilização de Spot Instances:\*\*

- \*\*Databricks:\*\*

1. Ao criar um cluster, vá para "Advanced Options".

2. Selecione "Spot Instances" e configure a quantidade desejada.

- \*\*Azure Databricks Pools:\*\*

- \*\*Databricks:\*\*

1. Acesse "Clusters".

2. Vá para "Instance Pools".

3. Crie um pool e configure os clusters para utilizá-lo.

#### \*\*8. Licenciamento e Planos:\*\*

- \*\*Revisão de Planos e Licenças:\*\*

- \*\*Azure Portal:\*\*

1. Acesse "Cost Management + Billing".

2. Vá para "Subscriptions".

3. Revise os planos e ajuste conforme necessário.

#### \*\*9. Auditoria e Revisão Regular:\*\*

- \*\*Revisão Periódica de Custos:\*\*

- \*\*Azure Portal:\*\*

1. Utilize "Cost Management + Billing" para gerar relatórios periódicos.

2. Configure dashboards para monitoramento contínuo.

- \*\*Feedback de Utilização:\*\*

- \*\*Internamente:\*\*

1. Realize reuniões periódicas com as equipes.

2. Colete feedback e implemente melhorias contínuas.

Implementando essas técnicas, você poderá reduzir custos de forma eficaz em jobs de streaming no Azure e Databricks, garantindo uma gestão eficiente e sustentável dos recursos.

### Ações para Redução de Custos em Data Lake usando Azure e Databricks

1. \*\*Otimização de Armazenamento:\*\*

- \*\*Compactação de Dados:\*\* Utilize formatos de arquivo compactados como Parquet ou ORC.

- \*\*Tiered Storage:\*\* Movimentação de dados menos acessados para armazenamento de custo mais baixo, como Azure Blob Storage Cool ou Archive tiers.

2. \*\*Gerenciamento de Dados:\*\*

- \*\*Políticas de Retenção de Dados:\*\* Defina políticas de retenção para eliminar dados obsoletos.

- \*\*Deduplicação de Dados:\*\* Remova dados redundantes para economizar espaço.

3. \*\*Otimização de Consultas:\*\*

- \*\*Particionamento de Dados:\*\* Particione os dados para melhorar a eficiência das consultas.

- \*\*Cache de Consultas:\*\* Utilize caching para acelerar consultas repetitivas.

4. \*\*Gerenciamento de Recursos:\*\*

- \*\*Auto-Scaling:\*\* Configure clusters com auto-scaling para ajustar automaticamente os recursos conforme a demanda.

- \*\*Cluster Size Optimization:\*\* Utilize clusters de tamanho adequado para workloads específicos.

5. \*\*Governança e Monitoramento:\*\*

- \*\*Tagging e Monitoramento:\*\* Utilize tags para monitorar o uso de recursos e identificar áreas de desperdício.

- \*\*Alertas de Custo:\*\* Configure alertas de custo para monitorar e controlar gastos em tempo real.

6. \*\*Otimização de Workloads:\*\*

- \*\*Job Scheduling:\*\* Agende jobs para horários de menor custo (por exemplo, fora do horário de pico).

- \*\*Parallelism and Concurrency:\*\* Ajuste o paralelismo e a concorrência das tarefas para otimizar o uso de recursos.

7. \*\*Utilização de Serviços Gerenciados:\*\*

- \*\*Utilização de Spot Instances:\*\* Utilize instâncias spot/preemptibles para workloads que toleram interrupções, reduzindo significativamente os custos.

- \*\*Azure Databricks Pools:\*\* Utilize pools de instâncias para reduzir o tempo de inicialização de clusters e aproveitar descontos.

8. \*\*Licenciamento e Planos:\*\*

- \*\*Revisão de Planos e Licenças:\*\* Revise e ajuste os planos de assinatura e licenciamento para garantir que a organização esteja utilizando a opção mais econômica.

9. \*\*Auditoria e Revisão Regular:\*\*

- \*\*Revisão Periódica de Custos:\*\* Realize auditorias regulares dos custos de uso e identifique oportunidades de otimização contínua.

- \*\*Feedback de Utilização:\*\* Colete feedback contínuo das equipes para identificar e resolver ineficiências no uso de recursos.

Seguir essas ações ajudará a manter os custos sob controle enquanto se beneficia da flexibilidade e escalabilidade das soluções Azure e Databricks.

Claro, vou detalhar mais cada uma das ações no portal da Azure e Databricks:

### 1. Otimização de Armazenamento

- \*\*Compactação de Dados:\*\*

- \*\*Databricks:\*\*

- Utilize a interface de notebooks do Databricks.

- Ao salvar dados, use comandos como `df.write.format("parquet").save("/path")`.

- \*\*Tiered Storage:\*\*

- \*\*Azure Portal:\*\*

- Vá para "Storage Accounts".

- Selecione a conta de armazenamento desejada.

- Navegue até "Blobs".

- Selecione o contêiner ou blob específico.

- Escolha "Change tier" e selecione a opção "Cool" ou "Archive".

### 2. Gerenciamento de Dados

- \*\*Políticas de Retenção de Dados:\*\*

- \*\*Azure Portal:\*\*

- Acesse sua conta de armazenamento.

- Selecione "Lifecycle Management" no menu à esquerda.

- Configure regras de ciclo de vida para mover ou excluir dados com base na idade dos dados.

- \*\*Deduplicação de Dados:\*\*

- \*\*Databricks:\*\*

- Use notebooks para criar scripts que detectem e removam duplicatas.

- Exemplo:

```python

df = df.dropDuplicates()

df.write.format("delta").save("/path")

```

### 3. Otimização de Consultas

- \*\*Particionamento de Dados:\*\*

- \*\*Databricks:\*\*

- Particione dados ao criar tabelas:

```python

df.write.partitionBy("column\_name").format("parquet").save("/path")

```

- \*\*Cache de Consultas:\*\*

- \*\*Databricks:\*\*

- Use Delta Cache:

```python

spark.conf.set("spark.databricks.io.cache.enabled", "true")

df = spark.read.format("delta").load("/path")

```

### 4. Gerenciamento de Recursos

- \*\*Auto-Scaling:\*\*

- \*\*Databricks:\*\*

- Acesse "Clusters" no Databricks.

- Clique em "Create Cluster".

- Em "Auto-Scaling", ative a opção e configure os parâmetros mínimos e máximos de nós.

- \*\*Cluster Size Optimization:\*\*

- \*\*Databricks:\*\*

- Em "Clusters", ajuste o tamanho e a configuração do cluster conforme necessário para a carga de trabalho específica.

### 5. Governança e Monitoramento

- \*\*Tagging e Monitoramento:\*\*

- \*\*Azure Portal:\*\*

- Acesse "Cost Management + Billing".

- Selecione "Cost Management" > "Cost analysis".

- Adicione tags aos recursos navegando até o recurso específico e clicando em "Tags".

- \*\*Alertas de Custo:\*\*

- \*\*Azure Portal:\*\*

- Vá para "Cost Management + Billing".

- Selecione "Budgets".

- Clique em "Add" para criar um novo orçamento e configurar alertas de custo.

### 6. Otimização de Workloads

- \*\*Job Scheduling:\*\*

- \*\*Databricks:\*\*

- Acesse "Jobs".

- Clique em "Create Job".

- Configure o job e defina a agenda de execução.

- \*\*Parallelism and Concurrency:\*\*

- \*\*Databricks:\*\*

- Configure a quantidade de paralelismo e concorrência nas configurações de cluster.

### 7. Utilização de Serviços Gerenciados

- \*\*Utilização de Spot Instances:\*\*

- \*\*Databricks:\*\*

- Ao criar um cluster, vá para "Advanced Options".

- Selecione "Spot Instances" e configure a quantidade desejada.

- \*\*Azure Databricks Pools:\*\*

- \*\*Databricks:\*\*

- Vá para "Clusters".

- Clique em "Instance Pools".

- Crie um novo pool e utilize-o na criação de clusters para reduzir o tempo de inicialização.

### 8. Licenciamento e Planos

- \*\*Revisão de Planos e Licenças:\*\*

- \*\*Azure Portal:\*\*

- Acesse "Cost Management + Billing".

- Revise os planos de assinatura em "Subscriptions".

- Ajuste para um plano mais adequado se necessário.

### 9. Auditoria e Revisão Regular

- \*\*Revisão Periódica de Custos:\*\*

- \*\*Azure Portal:\*\*

- Utilize "Cost Management + Billing" para gerar relatórios e revisar custos regularmente.

- \*\*Feedback de Utilização:\*\*

- \*\*Internamente:\*\*

- Realize reuniões regulares com as equipes para discutir o uso de recursos e identificar áreas de melhoria.

Essas ações detalhadas devem ajudá-lo a implementar estratégias eficazes de redução de custos usando o portal Azure e Databricks.

### Ações para Redução de Custos em Jobs de Stream no Azure e Databricks

1. \*\*Otimização de Armazenamento:\*\*

- \*\*Compactação de Dados:\*\*

- Utilize formatos de arquivo compactados (Parquet, ORC) para armazenar dados de streaming no Databricks.

- Exemplo:

```python

stream\_df.writeStream.format("parquet").option("path", "/path/to/output").start()

```

- \*\*Tiered Storage:\*\*

- Configure políticas de ciclo de vida no Azure Blob Storage para mover dados antigos para camadas de armazenamento mais baratas.

- Azure Portal: Vá para sua conta de armazenamento > "Lifecycle Management" > Configure regras para mover dados baseados na idade.

2. \*\*Gerenciamento de Dados:\*\*

- \*\*Políticas de Retenção de Dados:\*\*

- Use Azure Data Lake Storage ou Blob Storage para definir políticas de retenção automáticas.

- Azure Portal: Configure regras em "Lifecycle Management".

- \*\*Deduplicação de Dados:\*\*

- Implementar deduplicação em seu pipeline de streaming.

- Exemplo no Databricks:

```python

deduplicated\_df = stream\_df.dropDuplicates(["key\_column"])

deduplicated\_df.writeStream.format("delta").start("/path/to/output")

```

3. \*\*Otimização de Consultas:\*\*

- \*\*Particionamento de Dados:\*\*

- Particione os dados de streaming para melhorar a eficiência.

- Exemplo:

```python

stream\_df.writeStream.partitionBy("partition\_column").format("parquet").option("path", "/path/to/output").start()

```

- \*\*Cache de Consultas:\*\*

- Utilize caching no Databricks para consultas repetitivas.

- Exemplo:

```python

spark.conf.set("spark.databricks.io.cache.enabled", "true")

cached\_df = spark.readStream.format("delta").load("/path")

```

4. \*\*Gerenciamento de Recursos:\*\*

- \*\*Auto-Scaling:\*\*

- Configure clusters com auto-scaling para ajustar automaticamente os recursos.

- Databricks: Acesse "Clusters" > "Create Cluster" > Ative "Enable Auto-Scaling".

- \*\*Cluster Size Optimization:\*\*

- Ajuste o tamanho do cluster de acordo com a carga de trabalho de streaming.

- Databricks: Em "Clusters", ajuste o tamanho conforme necessário.

5. \*\*Governança e Monitoramento:\*\*

- \*\*Tagging e Monitoramento:\*\*

- Adicione tags aos recursos para monitoramento e controle de custos.

- Azure Portal: Adicione tags em "Cost Management + Billing" > "Tags".

- \*\*Alertas de Custo:\*\*

- Configure alertas de custo para monitorar gastos em tempo real.

- Azure Portal: Vá para "Cost Management + Billing" > "Budgets" > "Create Budget".

6. \*\*Otimização de Workloads:\*\*

- \*\*Job Scheduling:\*\*

- Otimize o agendamento dos jobs de streaming para horários de menor custo.

- Use Azure Stream Analytics para agendar jobs.

- \*\*Parallelism and Concurrency:\*\*

- Ajuste o paralelismo e a concorrência para otimizar o uso de recursos.

- Databricks: Configure essas opções nas configurações de cluster.

7. \*\*Utilização de Serviços Gerenciados:\*\*

- \*\*Utilização de Spot Instances:\*\*

- Use instâncias spot para reduzir custos em workloads que podem tolerar interrupções.

- Databricks: Configure "Spot Instances" em "Advanced Options" ao criar um cluster.

- \*\*Azure Databricks Pools:\*\*

- Utilize pools de instâncias para reduzir o tempo de inicialização e custos.

- Databricks: Vá para "Clusters" > "Instance Pools" > Crie e use um pool.

8. \*\*Licenciamento e Planos:\*\*

- \*\*Revisão de Planos e Licenças:\*\*

- Revise os planos de assinatura e licenciamento regularmente.

- Azure Portal: Acesse "Cost Management + Billing" > "Subscriptions".

9. \*\*Auditoria e Revisão Regular:\*\*

- \*\*Revisão Periódica de Custos:\*\*

- Utilize "Cost Management + Billing" para gerar relatórios de custo e revisar periodicamente.

- Configure relatórios e dashboards para monitoramento contínuo.

- \*\*Feedback de Utilização:\*\*

- Colete feedback das equipes regularmente para identificar áreas de melhoria.

- Realize reuniões periódicas para discutir o uso de recursos e otimizações.

Implementando essas ações específicas para jobs de streaming, você poderá reduzir custos enquanto mantém a eficiência e a performance de seu data lake.