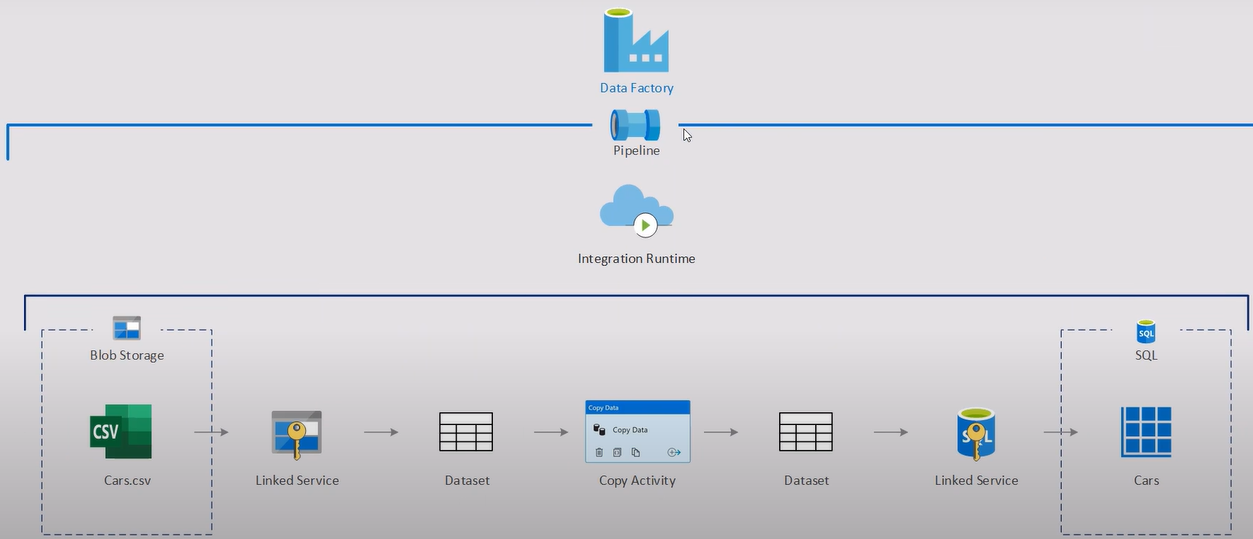
Arquitetura Azure Data Factory | Introdução a pipeline de dados



Serviços que iremos precisar para esse pipe:

- Azure Subscritption;

- Resource Group;

- Azure Data Factory;

- Azure SQL Database;

- Azure Serve Database

- Storage Account.

1. Crie a Azure Subscription(Assinatura) e o Resource group(Grupo de recursos).

2. Depois crie o Azure Data Factrory.

3. Crie um server Sql database.

4. Instancie o serviço SQL Database.

5. Storage Account

\* Usar o serviço Blob Service Dados do blob service:

- Storage account name: andreiastorageaccount

\* Criar um container chamado input e deixar como privado(para não acesso não autorizado ao serviço de armazenamento de blobs)

- Vá p/ cointainer e faça upload do arquivo csv.

- Clique no arquivo que vc fez upload e clique em edit blob.

6. Vá p/ o seu Resource Group e fixe o storage account p/ ficar mais fácil de navegar

7. Vá p/ o SQL Dtabase

\* Vá para a ferramenta Query editor --> Permite consultar o banco de dados sem precisar de nenhuma ferramenta externa. Use p/ logar no banco de dados.

8. Vá para o Data Factory

\* Clique em Open Azure Data Factory

- Lado esquerdo clique em autor

- Clique em newpipeline

\* Chame de pipeline1\_Blob\_SQL

- Agora faça a integração

\* Conectar um armazenamento de blobs.

OBS: Para conectar o armazenamento de blobs, precisamos de um serviço de link.

9. Ciar um linked Service

- Lado esquerdo vá para o botão Manage

\* Parte inferior create linked service

- Busca por Azure Blob Storage - continue

\* Name: inputblob

\* Depois vc precisa definir o tempo de integração em execução

- Nesse caso vamos deixá-lo com tempo de execução de integração de

resolução automática, ou seja, meu tempo de execução de integração

será dado p/ que eu não tenha que criá-lo por conta própria.

\* Depois para que possamos saber que iremos nos conectar corretamente. Existem várias maneiras de autenticar. Porém a maneira mais adequada e segura é através da Account Key.

\* Manter em From Azure Subscription - Que mantém em sua própria assinatura. Se vc acessar o armazenamento de blobs.

\* Storage account name: selecionar a própria assinatura de demonstração.

\* Testing connection

- Agora vamos definir dataset --> Definir um conjunto de dados p/ extrairmos os dados do armazenamento de blobs.

\* Lado esquerdo botão Author --> Dataset --> new dataset --> Procurar por Azure Blob Storage. --> continue.

\* Selecione o formato --> csv --> continue

\* Name: fake\_customersCSV

\* Linked Service: inputblob

\* filepath(localização do arquivo): Procure pelo browser

\* First row as header: selecione a primeira linha como cabeçalho.

\* Mantenha o import schema em FROM connection/store. --> OK

- Na guia de conexão, você pode alterar, caso tenha marcado algo errado.

- No delimitador de coluna, verifique se o seu está correto. No caso

de csv por ponto e vírgula. Verifique sempre.

- Você tmbm consegue visualizar seu schema em preview data.

- Agora vamos nos conectar ao SQL Database. Para isso iremos usar outro serviço.

\* Lado esquerdo na Fábrica --> Manage --> New --> Procure p/ Azure SQL Database

\* Selecione o serviço --> continue -->

\* Name: OutputSQL

\* Connect via integration runtime --> o msmo que usamos no anterior.

\* Azure subscription: a mesma

\* Server name: Usar o server que vc já tiver instanciado.

\* Database name: O data base que vc instanciou

\* Authentication type: SQL Authentication

\* User name: Outputsql

\* Password: \*\*\*\*\*\*\*\*\*

\* Testing connection

- Vamos criar outro dataset --> Author --> New Dataset --> Azure SQL Database --> continue

\* Name: fake\_customersTable

\* Linked Service: Outpusql

\* Table: dbo.fake\_customers

\* Import Schema: From connection/store --> OK

\* Save all

- No campo pipeline --> Move & Transform -- > Selecione e arraste Copy Data

\* No campo General --> Renomeie o campo Name p/ algo que sugere a atividade desse M.T. Nesse caso:

\* fake\_customers CSV to SQL.

\* No campo Source --> Source dataset --> selecione o dataset fake\_customers.csv

\* No campo Sink --> Sink dataset --> selecione o conjunto de dados fake\_customersTable

\* No campo Mapping --> Import schemas --> Você pode verificar o mapeamento dos dados e vê se os campos correspondentes estão corretos. Caso não esteja vc pode alterá-los.

\* Feito todos esses estepes, você pode verificar se de fato tudo está funcionando com debug ou publicar.

- Para vc verificar se todo esse esquema realmente funcionou:

- Sem sair da fabrica, vamos ao resource group.

- Indo ao SQL Database que vc usou --> Query editor --> Faça o logon para acessar o editor

\* Dê um "select \* from fake\_customers" --> "select count(\*) from fake\_customers"

- Uma vez que o fluxo funciona, vc poderá add uma Trigger para o fluxo de trabalho e monitorá-lo.

Vc pode agendar ou chamar o evento dentro da pipeline.

\* Na parte superior botão Add Trigger --> Trigger now(ACIONA A TRIGGER manualmente) --> clique em OK.

- Enviando uma nova execução da pipeline. Onde vc poderá monitorá-la através do botão Monitor

\* Portanto se voltarmos e consultarmos novamente na tabela, teremos aproximadamente 1/2 do dobro de linhas na nossa tabela. Esse não é o cenário perfeito prq apenas drobamos os dados.

O que significa que temos duplicatas em nosso banco de dados. Para lidar com essa situação.

No campo Sink em Author --> Pre-copy script --> delete from fake\_customers. Isso apagará todos os dados da tabela.