**Objetivo:** Neste exercício vamos realizar algumas atividades que permitam tomar familiaridade com a ferramenta de ingestão de dados NiFi.

**Inicio: Iniciando o NiFi**

1. Inicie o serviço NiFi:

nifi.sh start

1. Acesse o NiFi através do firefox no endereço:

http://hdpdemo.local:18080/nifi/

1. Dica: aguarde alguns minutos antes de acessar pois o serviço leva certo tempo para ficar disponível. Para verificar que o serviço está rodando, mesmo antes de estar com a interface disponível, rode o comando:

nifi.sh status

**Demonstração: Importação do primeiro arquivo para HDFS através do NiFi**

1. Na interface do NiFi, entre no grupo **input csv** e pare todos os processors
2. Volte ao terminal e renomeie o arquivo /data/input/airline\_safety/airline-safety.csv com o nome airline-safety2.csv (esta tarefa também pode ser feita através da interface gráfica)

cp /data/input/airline\_safety/processed/airline-safety.csv /data/input/airline\_safety/airline-safety\_2.csv

**Obs.:** a cada vez que for inserir novamente este arquivo, incremente o valor do arquivo no destino

1. Faça uma análise da configuração de cada processor iniciando-os um a um após a análise. Aproveite esta análise para estudar também o que ocorre com o flowfile a cada um dos processamentos.
2. Verifique os arquivos ingeridos no HDFS.
3. Faça alterações no grupo input csv para que os arquivos passem a ser salvos no formato avro, com compressão snappy, no diretório /data/airplane do HDFS.

**Exercício 1: Importar os arquivos CSV do diretório /data/marketplace/sales para o diretório /data/marketplace/sales no HDFS**

1. Copie o fluxo criado no item anterior.
2. Faça as alterações no fluxo do NiFi para que os arquivos sejam ingeridos no HDFS de acordo com as seguintes especificações:

* Os arquivos deverão ser ingeridos no formato CSV
* Os arquivos ingeridos deverão seguir o seguinte diretório:

**/data/marketplace/sales**

1. Verifique que a ingestão foi realizada com sucesso.

**Exercício 2: Importar dados do tópico Kafka para o HDFS**

1. Verifique no Ambari que os serviços Zookeeper e Kafka estão rodando, caso necessário suba os serviços
2. Instale os pacotes necessários para garantir que o simulador de logs para Kafka rode corretamente na VM

pip install confluent-kafka tzlocal faker

1. Copie o conteúdo do script log\_generator-kafka.py para a VM Inicie o script:

python log\_generator-kafka.py

**observação**: o arquivo pode ser obtido em

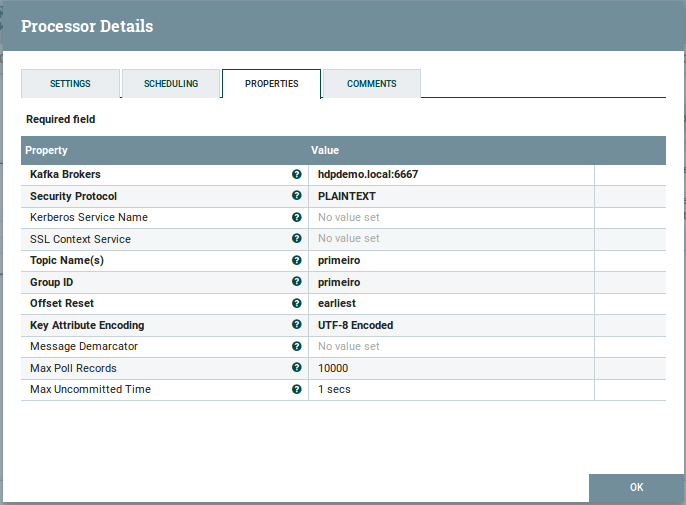
<https://github.com/thiagonogueira/fiap/blob/main/abd/ingestao%20de%20dados/nifi_kafka/log_generator-kafka.py>

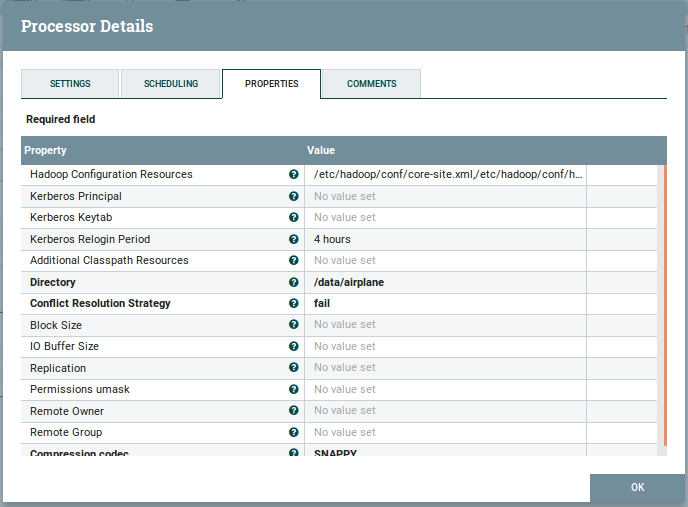
1. Valide a atividade no tópico Kafka log-generator:

kafka-console-consumer.sh --zookeeper localhost:2181 --topic log-generator

1. Utilize os processos ConsumeKafka e PutHDFS para realizar a tarefa. Ajuste os valores conforme a necessidade. O destino no HDFS deverá ser /data/kafka/kafka\_tail.

Utilize as imagens abaixo como guia, notando que alguns valores precisarão ser ajustados:



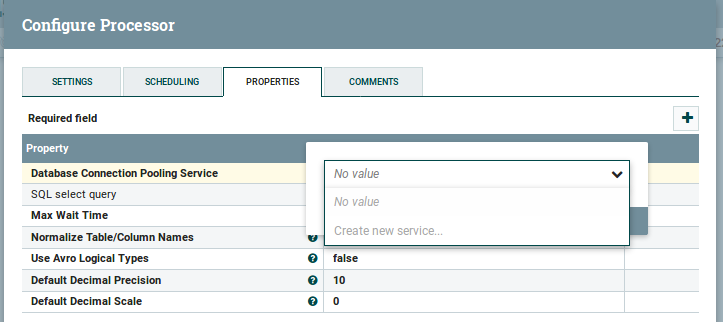


**Exercício 3: Ingestão de arquivos através de TailFile**

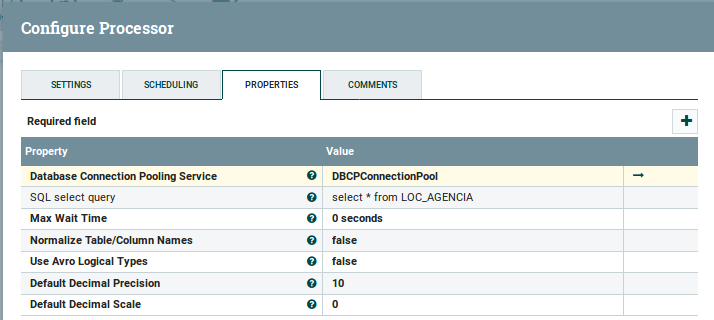
1. Utilize o processo TailFile para fazer a ingestão dos arquivos gerador pelo script gerador de logs diretamente no HDFS no diretório /data/tailfile/access\_log
2. **Desafio**: Procure adicionar um processor ao fluxo criado no item anterior para que escreva os eventos em nova fila kafka chamada access\_log?

**Desafio: Export do Banco de Dados para HDFS**

1. Utilize o processador ExecuteSQL para obter os dados da tabela LOC\_AGENCIA para o HDFS.
2. Ao configurar o ExecuteSQL, você deverá configurar um novo serviço para conexão ao banco de dados. Na configuração de **Database Connection Pooling Service** você deverá criar um novo serviço:



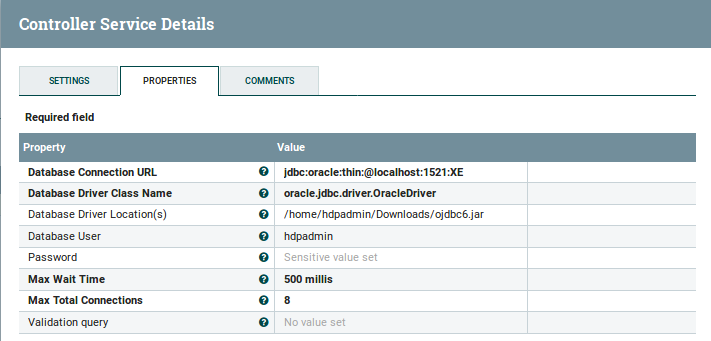
1. Após a criação, abra a tela de configuração clicando na seta que aparece à direita:



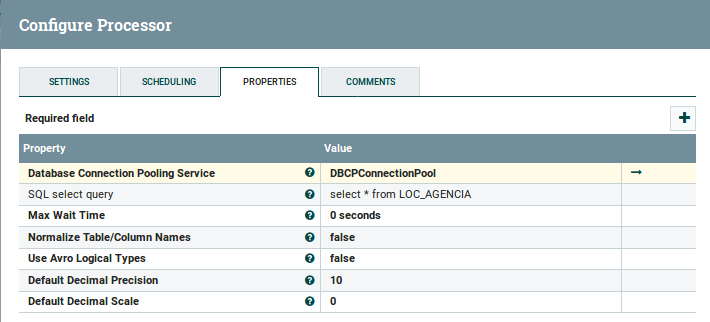
1. Clique na engrenagem à direita para entrar na configuração:



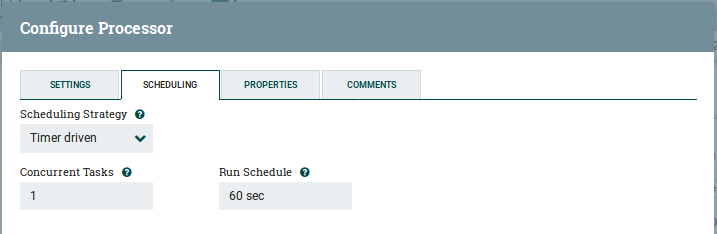
1. Configure conforme a imagem abaixo **Não se esqueça de adicionar também a senha do usuário hdpadmin (hadoop)**:



1. Ao terminar, clique no ícone de raio para iniciar o serviço:
2. Volte ao processador e configure conforme a imagem abaixo:



1. Na aba Scheduling, ajuste o parâmetro Run Schedule para 60s:



1. Utilize o processo PutHDFS para inserir no diretório /data/export\_db/stage