## ! STOP!

## You are required to watch the walkthrough video for this week's assignment. In the video, I guide you through each command, showing both the input and the expected output. Since you are working with new technology for the first time, it’s important not to just run a command

## Week 9 Assignment: Exploring Apache NiFi

### Objective: Mastering Apache NiFi for Dataflow Automation and Integration

In this assignment, you will explore **Apache NiFi**, a powerful tool for automating and managing data flows between systems. NiFi is widely used for moving and transforming data across different platforms, offering real-time control over data flow, routing, transformation, and system integration. You will design and implement dataflows, working with processors, parameter contexts, and integrating with **Apache Solr** for data indexing and querying.

By the end of this assignment, you will:

* Understand how to create and configure processor groups in NiFi.
* Gain experience in designing simple and advanced data flows using NiFi processors.
* Learn how to integrate NiFi with Solr for processing and indexing log data.
* Use Solr’s interface to query and analyze processed data.

#### **1. Environment Initialization**

* Change into the nifi directory with

cd dsc650-infra/bellevue-bigdata/nifi

* Start NiFi using the command:

/bin/bash nifi-\*/bin/nifi.sh start

* Access the NiFi User Interface using the instructions in the Week 1 assignment.

[Access NiFi UI](https://localhost:8443/nifi)

**Note:** Use the username and password obtained in Week 1 to access the interface.

**Deliverable:** Screenshot confirming successful access to the NiFi UI.

#### **2. Creating a Processor Group**

In this section, you will create your first **Processor Group** in NiFi, which serves as a logical container for grouping processors that will work together to form a data flow.

***Please follow the video tutorial which covers creating a process group and parameter context.***

**Deliverable:** Screenshot of the defined parameter within the Parameter Context.

#### **3. Designing a Simple Flow**

In this step, you will design a simple data flow using NiFi processors to generate files and log their attributes. This introduces you to how processors can be linked and configured to automate data processing.

***Please follow the video tutorial which covers creating designing your first flow.***

**Deliverable:** Screenshot of the simple flow (GenerateFlowFile to LogAttribute).

#### **4. Setting Up Solr Collection**

In this section, you’ll set up a Solr collection to store logs. This will integrate your NiFi flow with Solr, allowing you to push data into a searchable index.

* Navigate to the Solr directory and initiate the Solr Docker container, as instructed in the Solr assignment:

cd solr  
docker-compose up -d

Create a topic named nifi-syslog.

* Enter the Solr docker container:

docker exec -it solr\_solr\_1 bash

* If you can’t access the Solr container, it could be due to a container name change. In this cause use:

docker exec -it solr-solr-1 bash

* Create the Solr Collection

/opt/solr/bin/solr create -c syslog

**Deliverable:** Screenshot confirming the successful creation of the syslog collection.

#### **5. NiFi Advanced Flow with Solr**

Now, you’ll implement a more advanced data flow that integrates NiFi with Solr for log data processing and storage. You’ll import a pre-built flow to generate, filter, and index logs in Solr.

***Please follow the video tutorial which covers importing and starting your flow.***

**Deliverable:** Screenshot of the advanced NiFi flow processing and sending data to Solr.

#### **6. Querying Solr Data**

Finally, you will query the processed data stored in the Solr syslog collection. Use the Solr Web Interface to craft and execute queries, exploring different query parameters and filters.

Access the Solr Web Interface for querying:

[Query ‘syslog’ in Solr Web Interface](http://localhost:8983/solr/#/syslog/query)

Follow the link, and you’ll be presented with a user-friendly interface to craft and execute your queries. Experiment with different parameters and filters.

**Deliverable:** Screenshot of the Solr query results.