**Required Work**

This week does not include lecture or walkthrough videos. Instead, you will complete the **final project** by building an end-to-end data pipeline that integrates at least six major technologies: **NiFi, HDFS, YARN, Hive, HBase, and Spark**. Within Spark, you must also demonstrate a machine learning workflow using MLlib.

**Important Warning:** Using grades.csv or any provided example dataset will result in an automatic 0. You must use your own dataset hosted on GitHub.

**Submission Guidelines**

* Submit a **single, comprehensive report** (Word or PDF). Do not upload multiple raw screenshots.
* Your submission must include:
  + A detailed write-up of what you attempted, what worked, and what did not.
  + A discussion of challenges and issues encountered.
  + The GitHub direct URL for your dataset.
  + Screenshots showing successful execution at each stage (NiFi, HDFS, Hive, Spark, HBase).
  + The full code used (NiFi templates, PySpark scripts, Hive queries, etc.).
  + A well-structured format with labeled sections: Introduction, Dataset, Pipeline Overview, Issues Encountered, Screenshots, Code, Conclusion.
* Manage VM resources carefully. After loading data into HDFS with NiFi, stop NiFi before running Hive, Spark, or HBase.
* This is a **master’s level course** – clarity, professionalism, and technical completeness are expected.

**Week 11 Final Project – Objectives and Points (660 pts)**

**Objective 1 – NiFi Data Ingestion into HDFS (220 pts)**

* Submit a screenshot of your NiFi flow design showing all processors used.
* Submit a screenshot of the flow running with data visible in queues.
* Submit a screenshot of hdfs dfs -ls confirming that the dataset was successfully ingested into HDFS.
* Provide a brief written explanation of the processors used and their roles in the flow.

**Objective 2 – Hive Managed Table (100 pts)**

* Submit your CREATE TABLE statement.
* Submit a screenshot showing data successfully loaded into Hive.
* Submit query results verifying that the data is populated (include at least one aggregation query to confirm schema correctness).
* Provide a short explanation of your schema design choices.

**Objective 3 – Environment Setup (40 pts)**

* Submit a screenshot of package install commands (e.g., numpy, happybase) along with successful installation output.
* Submit a screenshot showing the HBase Thrift server running.
* Provide a short explanation of why these steps are necessary for the project pipeline.

**Objective 4 – HBase Table Creation (40 pts)**

* Submit the create command for your HBase table.
* Submit a screenshot of an empty scan command result before inserting any data.
* Provide an explanation of your row key and column family design.

**Objective 5 – PySpark ML Code (220 pts)**

* Submit your complete PySpark ML code in the document and provide a link to your GitHub repository.
* Submit screenshots showing the code running successfully with training output.
* Submit a screenshot of evaluation metrics (e.g., accuracy, RMSE, or another appropriate metric).
* Provide a written explanation of the algorithm you chose, why you selected it, and what the results show.

**Objective 6 – Spark Submit & Output (20 pts)**

* Submit a screenshot of the spark-submit command used to run your job.
* Submit screenshots of the Spark job logs and final output.

**Objective 7 – HBase Scan Verification (20 pts)**

* Submit a screenshot of a populated scan command result after Spark has written metrics into HBase.
* Provide a short explanation of what metrics were written and how they validate that your pipeline worked end-to-end.

**Total: 660 points**

## Weeks 11: Final Project Assignment

### Objective

For the final project, you will design and implement a complete end-to-end pipeline that demonstrates your ability to work across the major components of the Hadoop and Spark ecosystem.

**Your pipeline will integrate at least six major components from the following ecosystem: NiFi, HDFS, YARN, Hive, HBase, and Spark. Within Spark, you must also demonstrate machine learning (MLlib).**

### Requirements

#### Pipeline Composition

Your pipeline will utilize six components learned in class:

* **NiFi** for data ingestion
* **HDFS** for data storage
* **YARN for resource management.**
* **Hive** for data warehousing
* **HBase** for NoSQL storage
* **Spark** for data processing, querying, and machine learning

#### Dataset

* Use a dataset of your choice — **do not reuse any course-provided datasets**.
* The dataset must be hosted on **GitHub**, and you must use the **direct file URL** as demonstrated throughout the semester.
* Update the NiFi HTTP processor to point to your dataset’s GitHub URL.

#### Component Integration

The components must be fully connected and work together in sequence. A typical flow might look like this:

1. Use NiFi to ingest data and load it into HDFS.
2. Move the dataset into Hive for querying.
3. Apply **Spark MLlib** to the dataset (refer to SparkMachineLearning.docx).
4. Store performance metrics (accuracy, precision, recall, etc.) into HBase (see WriteHBasePySpark.docx).
5. Retrieve stored metrics from HBase.

### Critical System Resource Tip – Stop NiFi After Data Load

Your VM has **8 CPUs and 16 GB RAM**, which is extremely limited to keep cloud costs low. You cannot run all services simultaneously.

* After loading your dataset into HDFS with NiFi, **stop NiFi before continuing** with Hive, Spark, or HBase.
* In real-world production environments, organizations operate at a vastly larger scale with hundreds of servers and terabytes of memory.
* In this academic environment, you must **manage limited resources efficiently** to prevent system crashes.

### Evaluation Criteria

* Completeness and correctness of your pipeline implementation.
* Effective use of the technologies taught in the course.
* Uniqueness of your dataset and schemas.
* Effort and depth of your analysis.
* Clarity and organization of your report.

### Project Guides

You are required to refer to the project guides and templates in the following repository:  
<https://github.com/bellevue-university/dsc650-updated/tree/main/week11-12/assignment/examples>

You are required to use these resources as guides, but you cannot use the same datasets or data schemas provided in these examples. The Docker networking has already been configured and tested for seamless connectivity between containers.

* **Final\_Project.json**: A NiFi template designed for downloading your own dataset and loading it into HDFS. You are required to use this template.
* **HDFS\_Hive\_Spark\_HBase\_Pipeline.docx**: A step-by-step walkthrough of an end-to-end pipeline using a sample dataset. You are required to reverse engineer this process for your own dataset. Submitting work with the sample dataset will result in an automatic zero.