**Apache NiFi Advanced Lab Guide**

**Topics Covered:**  
FlowFile/Provenance Chain · Core Processors · Back-pressure & Prioritizer · Template/Version-Control Registry · Site-to-Site (HTTPS) · Cluster Admin Basics

**Lab 1: Understanding FlowFiles & Provenance**

**Objective:**

Track the journey of a FlowFile through the NiFi dataflow and analyze provenance.

**Steps:**

1. **Launch NiFi UI** (e.g., <http://localhost:8080>).
2. Drag and configure GenerateFlowFile.
   * Set Custom Text to "Hello NiFi".
   * Set Run Schedule to 10 sec.
3. Add LogAttribute processor.
4. Connect GenerateFlowFile → LogAttribute.
5. Enable and run both processors.
6. After data flows, go to the **Provenance** tab:
   * Filter events by Component ID.
   * Explore the life cycle of the FlowFile.
   * Download the content and check metadata.

**Lab 2: Using Core Processors for File Flow**

**Objective:**

Build a basic file ingestion and routing pipeline.

**Steps:**

1. Use GetFile to pick up files from an input directory.
2. Add RouteOnAttribute to classify files based on filename/size.
3. Add two PutFile processors to write outputs to separate folders (e.g., /processed/small, /processed/large).
4. Run a test file through the system.
5. Observe flow movement, failure handling, and attributes.

**Lab 3: Configure Back-Pressure and Queue Prioritizers**

**Objective:**

Throttle flow using queue limits and prioritize older FlowFiles.

**Steps:**

1. Use GenerateFlowFile → LogAttribute.
2. Right-click connection → Configure:
   * Set back pressure to **10 FlowFiles**.
   * Add **AgePrioritizer**.
3. Observe that after the queue reaches the limit, GenerateFlowFile halts.
4. Manually drain flow files and see processing resume.

**Lab 4: Export/Import Flows as Templates & Use Version Control**

**Objective:**

Modularize a flow and manage changes with NiFi Registry.

**Steps:**

1. Select an existing flow, click **Create Template**.
2. Export and re-import into another canvas.
3. Set up NiFi Registry (install & configure if not already):
   * Register a flow.
   * Make changes (e.g., add a processor).
   * Commit changes and observe version history.
4. Try **reverting** to an older version.

**Lab 5: Setup Site-to-Site Data Transfer (HTTPS)**

**Objective:**

Configure data exchange between two NiFi instances.

**Pre-requisites:**

* Two NiFi instances (e.g., nifi1, nifi2)
* SSL certificates and secured NiFi setup (or HTTP mode for testing)

**Steps:**

1. On **nifi2**, create an input port: ReceiveData.
2. On **nifi1**, drag Remote Process Group (RPG) and set target URL.
3. Connect local processors to the RPG.
4. Bind data to ReceiveData port.
5. Validate transmission and check remote logs.

**Lab 6: Perform Cluster Admin Tasks**

**Objective:**

Deploy a basic NiFi cluster and monitor node behavior.

**Steps:**

1. Set up 2+ NiFi nodes with shared nifi.properties cluster settings.
2. Enable cluster mode:
   * Set nifi.cluster.is.node=true
   * Set unique ports per node
   * Shared Zookeeper config (or dummy ZK for learning)
3. Run NiFi and access UI:
   * Observe **Cluster UI**
   * View node health
   * Designate **Primary Node Tasks**
4. Add processor marked **Primary Node Only** and test if it only runs on one node.

**Lab 7: Secure NiFi with HTTPS and Users**

**Objective:**

Secure a NiFi instance and configure basic access control.

**Steps:**

1. Generate a TLS keystore and truststore using tls-toolkit.sh.
2. Modify nifi.properties to enable HTTPS.
3. Restart NiFi and access via HTTPS.
4. Add users and roles:
   * Enable login with certificates or LDAP
   * Assign policy (read/write/view provenance)
5. Try logging in as different users and see access differences.