Installing HADOOP on Intel based desktop PC with ANSI C Compiler and Supporting Editors

General Guidelines:

To install Hadoop on an Intel-based desktop PC using ANSI C compiler and supporting editors, the process will deviate slightly from typical Java-based installations. Hadoop is fundamentally written in Java, but you can work with its C bindings using Hadoop's **native libraries** (for performance-critical tasks). Here's how to set up Hadoop with an emphasis on C-based components:

Prerequisites

1. **Intel-based Desktop PC**: Ensure your PC meets Hadoop's minimum requirements:
   - 4 GB RAM (8 GB recommended).
   - At least 20 GB of free storage.
2. **Operating System**: Use a Linux distribution (e.g., Ubuntu, CentOS).
3. **ANSI C Compiler**: Install `gcc` or `clang`.
4. **Hadoop Source Code**: Download the Hadoop source for native library compilation.

Step 1: Install Linux OS

- Install a Linux OS if it's not already installed. You can use Ubuntu or CentOS.
- Set up SSH for communication (Hadoop uses SSH for cluster management).

Step 2: Install System Dependencies

1. Update your system:

   ```
   sudo apt update && sudo apt upgrade -y
   ```

2. Install essential tools:

   ```
   sudo apt install build-essential gcc g++ make ssh rsync
   wget -y
   ```

Step 3: Install Java

Hadoop requires Java for most operations, even if you're using native libraries.

```
sudo apt install openjdk-11-jdk -y
```

Step 4: Download and Compile Hadoop Source

1. Download the Hadoop source code:

   ```
   wget https://downloads.apache.org/hadoop/common/hadoop-
   X.Y.Z/hadoop-X.Y.Z-src.tar.gz
   ```

   Replace `X.Y.Z` with the desired version number.

2. Extract the source code:

```
tar -xvzf hadoop-X.Y.Z-src.tar.gz
cd hadoop-X.Y.Z-src
```

3. Compile native libraries:

```
mvn package -Pdist,native -DskipTests -Dtar
```

Note: You need Maven installed (`sudo apt install maven -y`). This command compiles the Hadoop distribution with native libraries.

Step 5: Set Up Hadoop Configuration

1. Move the compiled Hadoop distribution:

```
sudo mv hadoop-X.Y.Z /usr/local/Hadoop
```

2. Set environment variables: Edit `~/.bashrc`:

```
nano ~/.bashrc
```

Add:

```
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
export JAVA_HOME=$(readlink -f /usr/bin/java | sed
"s:/bin/java::")
```

Apply the changes:

```
source ~/.bashrc
```

Step 6: Enable C Libraries

1. Confirm the native library is built:

```
ls $HADOOP_HOME/lib/native
```

This directory should contain `.so` (shared object) files for Hadoop's native operations.

2. Set the library path: Add this to your `~/.bashrc`:

```
export
LD_LIBRARY_PATH=$HADOOP_HOME/lib/native:$LD_LIBRARY_PATH
```

Reload:

```
source ~/.bashrc
```

Step 7: Test Hadoop with C-based Tools

- Hadoop comes with **pipes** (C++ applications) and **native tasks**. Here's how to test them:

1. **Run Hadoop Pipes Example**: Pipes allow you to write MapReduce jobs in C++.
   - Compile the example job:

```
g++ -o wordcount wordcount.cc -lhadooppipes -
lhadooputils
```

- Submit the job:

  ```
  hadoop pipes -input <input_path> -output
  <output_path> -program ./wordcount
  ```

2. **Test HDFS with Native Libraries**: Run:

   ```
   hadoop checknative -a
   ```

   It should list available native features (e.g., compression codecs).

Step 8: Install Supporting Editors

To make development easier:

1. Install text editors:

   ```
   sudo apt install vim nano -y
   ```

2. Install IDEs or tools for C development:

   ```
   sudo apt install codeblocks geany –y
   ```

Step 9: Validate the Setup

1. Start Hadoop services:

   ```
   start-dfs.sh
   start-yarn.sh
   ```

2. Verify:

   ```
   jps  # Ensure NameNode, DataNode, etc., are running.
   hadoop version
   ```

Key Points

- Hadoop's native libraries provide optimized performance but depend heavily on the Java layer.
- Use tools like **Hadoop Pipes** for integrating C/C++ code into Hadoop jobs.
- For deep C-level customization, explore **libhdfs** (C API for HDFS).

****

WEEK-1: HADOOP SETTING AND INSTALLATION

1. Installation of VMWare to set up the Hadoop open environment and its ecosystems
2. Perform setting up and Installing Hadoop in its following nodes.
3. Single node, multi node

Ans: A step-by-step guide to set up and install Hadoop on VMware for single-node and multi-node configurations:

1. **Installation of VMware:**
   1. **Download VMware:**
      - Download VMware Workstation Player or VMware Workstation Pro from the VMware official website.
   2. **Install VMware:**
      - Run the installer and follow the installation wizard.
      - After installation, open VMware.
   3. **Set Up the Virtual Machine:**
      - Download a Linux distribution (e.g., Ubuntu, CentOS) ISO file.
      - In VMware, create a new virtual machine:
        - Select "Install OS from ISO" and attach the Linux ISO file.
        - Allocate resources:
          - **RAM:** Minimum 4GB (preferably 8GB or higher for multi-node setups).
          - **Disk:** At least 20GB for single-node or 50GB for multi-node configurations.
        - Finish setup and install Linux on the VM.

****

2. **Setting up and Installing Hadoop in Linux Nodes:**

   Preparation:

   1. **Update the System:**

      ```
      sudo apt update && sudo apt upgrade -y  # For Ubuntu
      ```

      or

      ```
      sudo yum update -y  # For CentOS
      ```

   2. **Install Required Packages:**

      ```
      sudo apt install openjdk-11-jdk ssh rsync -y  # For Ubuntu
      ```

      or

      ```
      sudo yum install java-11-openjdk-devel ssh rsync -y  # For CentOS
      ```

3. **Verify Java Installation:**

```
java –version
```

4. **Download Hadoop:**

   - Visit the [Apache Hadoop download page](https://hadoop.apache.org/releases.html).
   - Download the latest stable release and extract it:

```
wget https://downloads.apache.org/hadoop/common/hadoop-X.Y.Z/hadoop-X.Y.Z.tar.gz
tar -xvzf hadoop-X.Y.Z.tar.gz
sudo mv hadoop-X.Y.Z /usr/local/Hadoop
```

5. **Set Hadoop Environment Variables: Add the following to ~/.bashrc:**

```
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:/bin/java::")
```

Reload the profile:

```
source ~/.bashrc
```

****

3. Single-Node Setup:

1. **Configure Hadoop:** Edit the following files in `/usr/local/hadoop/etc/hadoop/`:

- **`core-site.xml`:**

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

- **hdfs-site.xml :**

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

- **mapred-site.xml (Create if missing):**

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

- **yarn-site.xml:**

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

Format the Namenode:

2. **Format the Namenode:**

   ```
   hdfs namenode -format
   ```

3. **Start Hadoop Services:** Edit the following files in /usr/local/hadoop/etc/hadoop/:

   start-dfs.sh

   start-yarn.sh

4. **Verify the Setup:**
   - **Access the Hadoop web UI:**
     - HDFS: `http://localhost:9870`
     - YARN: `http://localhost:8088`

****

4. Multi-Node Setup:

1. **Set Up Additional Virtual Machines:**
   - Clone or create additional VMs for each node (e.g., Master, Worker1, Worker2).
2. **Configure Hostnames:** Edit `/etc/hosts` to include all nodes' IPs and hostnames:

   ```
   192.168.1.100 master
   192.168.1.101 worker1
   192.168.1.102 worker2
   ```

3. **Enable SSH Key-Based Authentication:** On the master node:

   ```
   ssh-keygen -t rsa
   ssh-copy-id worker1
   ssh-copy-id worker2
   ```

4. **Update Hadoop Configuration:**
   - **`core-site.xml`**:

     ```
     <configuration>
       <property>
     ```

```
            <name>fs.defaultFS</name>
            <value>hdfs://master:9000</value>
        </property>
    </configuration>
```

- **hdfs-site.xml:**

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>2</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>

<value>/usr/local/hadoop/data/namenode</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>

<value>/usr/local/hadoop/data/datanode</value>
    </property>
</configuration>
```

5. **Distribute Hadoop Configuration Files:** Copy the Hadoop folder from the master to worker nodes:

   ```
   scp -r /usr/local/hadoop worker1:/usr/local/hadoop
   scp -r /usr/local/hadoop worker2:/usr/local/hadoop
   ```

6. **Start Hadoop Services:** On the master node:

   ```
   start-dfs.sh
   start-yarn.sh
   ```

7. **Verify Multi-Node Setup:**
   - Access the same Hadoop web UIs as in single-node setup.
   - Verify worker nodes are listed under the "Nodes" section.