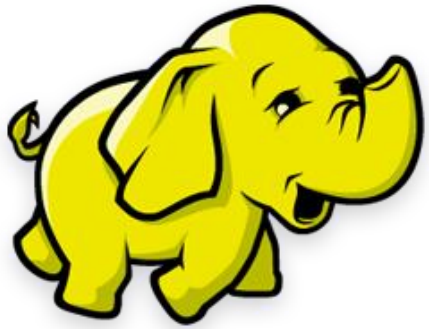
Apache Hadoop Installation and Configuration

COMP9313: Big Data Management

Installing Hadoop



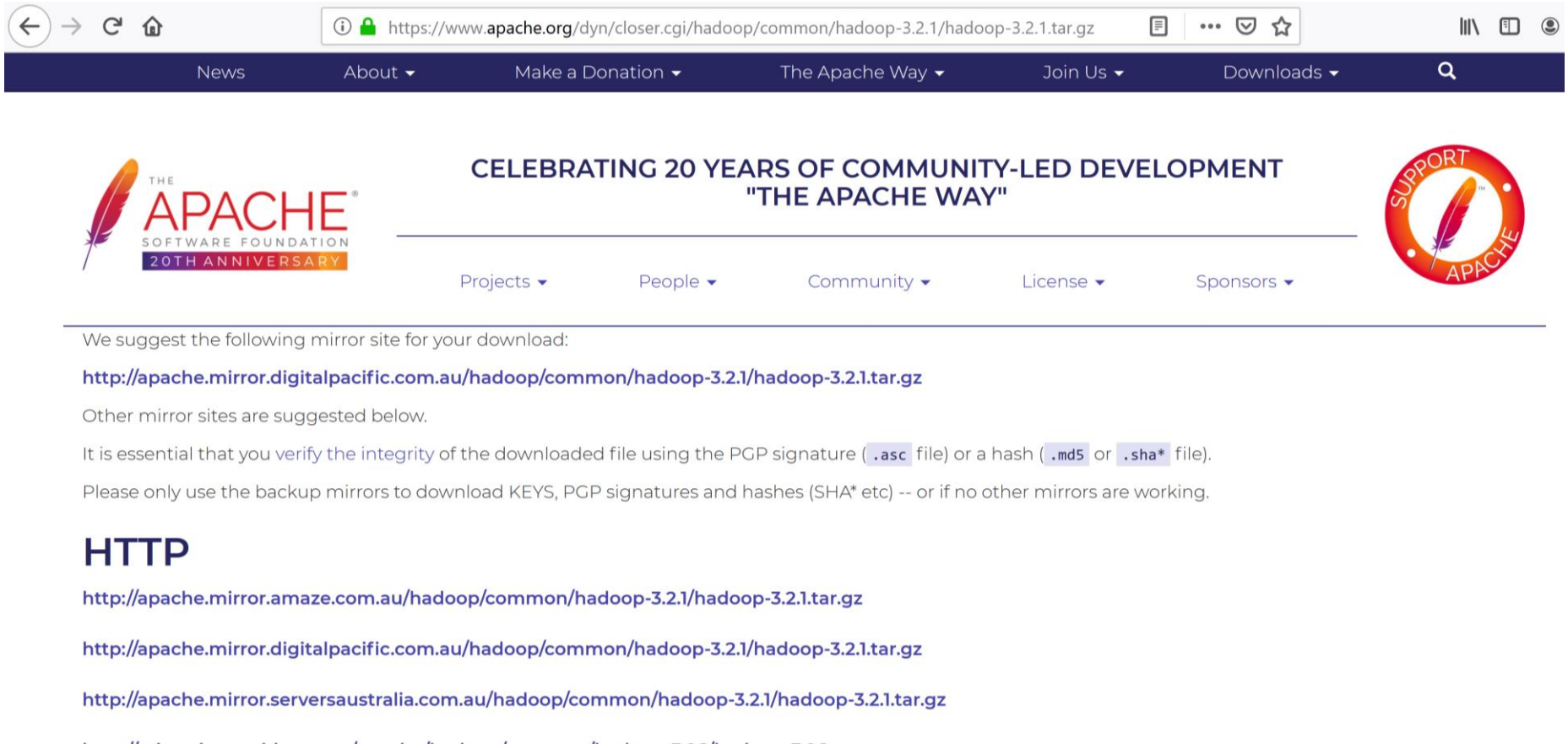
source

- Do I need a supercomputer to have a taste of Hadoop? **No**
- Do I need more than one computer to install Hadoop? **No, at least not for testing**
- Can I install Hadoop on a Windows Machine? **Yes, But…**
- How do I debug my MapReduce program?

What you need to do before installing and Configuring Hadoop

- Create a User for Hadoop, Why?
- Make sure SSH is installed
  - Generate Keys (More relevant for Distributed installation)
- Install Java
- Add Java_HOME in your default shell (e.g., bashrc)
  - export JAVA_HOME=/usr/local/[Java Folder]
  - export PATH=$PATH:$JAVA_HOME/bin

Download Hadoop



tar xzf hadoop-3.2.1.tar.gz
mv hadoop-3.2.1/* to hadoop/

Hadoop Operation Modes

- **Local/Standalone Mode** − After downloading Hadoop in your system, by default, it is configured in a standalone mode and can be run as a single java process.
- **Pseudo Distributed Mode** − It is a distributed simulation on single machine. Each Hadoop daemon such as hdfs, yarn, MapReduce etc., will run as a separate java process. This mode is useful for development.
- **Fully Distributed Mode** − This mode is fully distributed with minimum two or more machines as a cluster.

Installing Hadoop in Standalone Mode

- This is how Hadoop run by default
- There are no daemons running and everything runs in a single JVM.
- Standalone mode is suitable for running MapReduce programs during development, since it is easy to test and debug them.

  - Just add HADOOP_HOME to your ~/.bashrc and you are good to go

export HADOOP_HOME=/usr/local/hadoop

Installing Hadoop in Pseudo Distributed Mode

- Single Node cluster
- You need to have HDFS and YARN running

You can set Hadoop environment variables by appending the following commands to **~/.bashrc** file.

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
```

Where can I find the Configuration files?

- You can find all the Hadoop configuration files in the location “$HADOOP_HOME/etc/hadoop”.
- It is required to make changes in those configuration files according to your Hadoop infrastructure
- In order to develop Hadoop programs in java, you have to reset the java environment variables in **hadoop-env.sh** file by replacing **JAVA_HOME** value with the location of java in your system.

What configuration files do I need to change

- **core-site.xml**
- **hdfs-site.xml**
- **yarn-site.xml**
- **mapred-site.xml**

“core-site.xml” Configuration file

- The **core-site.xml** file contains information such as the port number used for Hadoop instance, memory allocated for the file system, memory limit for storing the data, and size of Read/Write buffers.

“core-site.xml” Configuration file

Specify the name and port

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

“hdfs-site.xml” Configuration file

- The **hdfs-site.xml** file contains information such as the value of replication data, namenode path, and datanode paths of your local file systems.
- It means the place where you want to store the Hadoop infrastructure.

“hdfs-site.xml” Configuration file

- Define the replication factor and the path for the namenode and datanode

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

   <property>
      <name>dfs.name.dir</name>
      <value>file:///home/hadoop/hadoopinfra/hdfs/namenode </value>
   </property>

   <property>
      <name>dfs.data.dir</name>
      <value>file:///home/hadoop/hadoopinfra/hdfs/datanode </value>
   </property>
</configuration>
```

“yarn-site.xml” Configuration file

- This file is used to configure yarn into Hadoop
- You can also specify what YARN services you are supporting

“yarn-site.xml” Configuration file

- For the purpose of illustration let’s allow shuffling and sorting in the node manager

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

“mapred-site.xml” Configuration file

- specify which MapReduce framework we are using

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

“mapred-site.xml” Configuration file

Parameter	Value	Description
mapreduce.framework.name	yarn	Execution framework set to Hadoop YARN.
mapreduce.map.memory.mb	1024	Larger resource limit for maps.
mapreduce.map.java.opts	-Xmx1024M	Larger heap-size for child jvms of maps.
mapreduce.reduce.memory.mb	3072	Larger resource limit for reduces.
mapreduce.reduce.java.opts	-Xmx2560M	Larger heap-size for child jvms of reduces.
mapreduce.task.io.sort.mb	512	Higher memory limit while sorting data for efficiency.
mapreduce.task.io.sort.factor	100	More streams merged at once while sorting files.
mapreduce.reduce.shuffle.parallelcopies	50	Higher number of parallel copies run by reduces to fetch outputs from very large number of maps.

Questions?

Notes

- Thursday lecture time we'll have a hands-on activity
  - Bring Laptop
  - We'll run a step by step activity
  - We can't expect for everyone to complete in time that is why we have Labs (starting week4)