+ Java IO Streams:
    + Binary Streams -- InputStream & OutputStream
        - DataInputStream & DataOutputStream
    + Character Streams -- Reader & Writer
        - Handling character encoding/charset

+ To install Hadoop -- Local Installation:
    - unzip hadoop-x.y.z.tar.gz
    - <hadoop>/etc/hadoop-env.sh --> JAVA_HOME
    - in ~/.bashrc
        export HADOOP_HOME=/path/to/hadoop
        export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
    - Hadoop commands:
        hadoop fs -help
        hadoop fs -ls /
        hadoop fs -cat /home/atlas/.bashrc
    - In local mode we can access LocalFileSystem only (not HDFS).
    - In local mode MR framework is "local" (not YARN).

==========

+ To install Hadoop -- Pseudo Distrubution Installation:

1. Prepare machine.
    Install java-1.8-64 bit & ssh on machine.

2. Disable ipv6 on all machines -- /etc/sysctl.conf (Optional)
    net.ipv6.conf.all.disable_ipv6 = 1
    net.ipv6.conf.default.disable_ipv6 = 1
    net.ipv6.conf.lo.disable_ipv6 = 1

3. In /etc/hosts ensure entry of standalone hostname.
    127.0.0.1   localhost

4. Enable password-less login for SSH:
        ssh-keygen -t rsa -P ""
        cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
        chmod 600 $HOME/.ssh/authorized_keys
        ssh localhost

5. Download & Extract into $HOME Hadoop 2.7.3.
    cd ~
    tar xvf hadoop-2.7.3.tar.gz

6. In $HOME/.bashrc setup HADOOP_HOME and PATH:

```
export HADOOP_HOME=$HOME/hadoop-2.7.3
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

7. In $HADOOP_HOME/etc/hadoop/hadoop-env.sh:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

8. In $HADOOP_HOME/etc/hadoop/core-site.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
        <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
        </property>
</configuration>
```

9. In $HADOOP_HOME/etc/hadoop/hdfs-site.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
        <property>
        <name>dfs.name.dir</name>
        <value>${user.home}/bigdata/hd-data/nn</value>
        </property>
        <property>
        <name>dfs.replication</name>
        <value>1</value>
        </property>
        <property>
        <name>dfs.data.dir</name>
        <value>${user.home}/bigdata/hd-data/dn</value>
        </property>
</configuration>
```

10. In $HADOOP_HOME/etc/hadoop/mapred-site.xml:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
        <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
```

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

11. In $HADOOP_HOME/etc/hadoop/yarn-site.xml:

```
    <?xml version="1.0"?>
    <configuration>
        <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>localhost</value>
        </property>
        <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
        </property>
        <property>
        <name>yarn.nodemanager.local-dirs</name>
        <value>${user.home}/bigdata/hd-data/yarn/data</value>
        </property>
        <property>
        <name>yarn.nodemanager.logs-dirs</name>
        <value>${user.home}/bigdata/hd-data/yarn/logs</value>
        </property>
        <property>

<name>yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage</name
>
            <value>99.9</value>
        </property>
        <property>
            <name>yarn.nodemanager.vmem-check-enabled</name>
            <value>false</value>
        </property>
    </configuration>
```

12.

```
    $HADOOP_HOME/etc/hadoop/slaves
        localhost
```

13. Format namenode:

```
        hdfs namenode -format
```

14. Start HDFS & YARN:

```
        start-dfs.sh
```

start-yarn.sh

Verify daemons using "jps" command.

15. HDFS commands:

```
hadoop fs -ls /
hadoop fs -mkdir -p /user/data
hadoop fs -put localfilepath /user/data
hadoop fs -get /user/data/filepath localfilepath
```

16. Stop HDFS & YARN:

```
stop-yarn.sh
stop-dfs.sh
```

Verify daemons using "jps" command.

==========

Multi-Node cluster setup:

- Identify machines on which cluster to be developed. Connect them in a network -- 1(master-NN,SNN,RM) + 3(slaves-DN,NM).
- On each machine /etc/hosts make entries of machines:

```
192.168.56.1   master
192.168.56.101      vm1
192.168.56.102      vm2
192.168.56.103      vm3
```

- Create user on each machine for running hadoop processes – "hduser".
- The master should be able to access each slave machine & itself over ssh without password.

```
ssh-keygen -t rsa -P ""
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
chmod 600 $HOME/.ssh/authorized_keys
ssh localhost
```

- Then copy key to all slaves:

```
ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@vmX
```

- Copy Hadoop distribution on all machines. Extract there. Set HADOOP_HOME & PATH in $HOME/.bashrc.
- hadoop-env.sh (Master+Slaves):

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

- core-site.xml (Master+Slaves):

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

- hdfs-site.xml

◦ Master (NameNode)

```
<property>
        <name>dfs.name.dir</name>
        <value>${user.home}/bigdata/nn</value>
</property>
<property>
        <name>dfs.replication</name>
        <value>2</value>
</property>
```

◦ Slaves (DataNodes)

```
<property>
        <name>dfs.replication</name>
        <value>2</value>
</property>
<property>
        <name>dfs.data.dir</name>
        <value>${user.home}/bigdata/dn</value>
</property>
```

• mapred-site.xml (Master+Slaves):

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

• yarn-site.xml

◦ Master (ResourceManager)

```
<property>
        <name>yarn.resourcemanager.hostname</name>
        <value>master</value>
</property>
```

◦ Slaves (NodeManagers)

```
<property>
        <name>yarn.resourcemanager.hostname</name>
        <value>master</value>
</property>
<property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
</property>
<property>
        <name>yarn.nodemanager.local-dirs</name>
        <value>${user.home}/bigdata/yarn/data</value>
```

```
                </property>
                <property>
                        <name>yarn.nodemanager.logs-dirs</name>
                        <value>${user.home}/bigdata/yarn/logs</value>
                </property>
                <property>

<name>yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage</name
>
                                <value>99.9</value>
                </property>
                <property>
                <name>yarn.nodemanager.vmem-check-enabled</name>
                        <value>false</value>
                </property>
```

- slaves (Master):

```
vm1
vm2
vm3
```

- On Master:

```
hdfs namenode -format
start-dfs.sh
```

==========

+ HDFS Java APIs -- *.jar <-- hadoop-2.7.3/share/hadoop/hdfs/*, hadoop-2.7.3/share/hadoop/command/*

```
FileSystem
        |- LocalFileSystem              --> represent local fs (local mode)
        |- DistributedFileSystem--> represent hdfs (distributed mode)

FileStatus

DataInputStream
        |- FsDataInputStream

DataOutputStream
        |- FsDataOutputStream
```