Hadoop

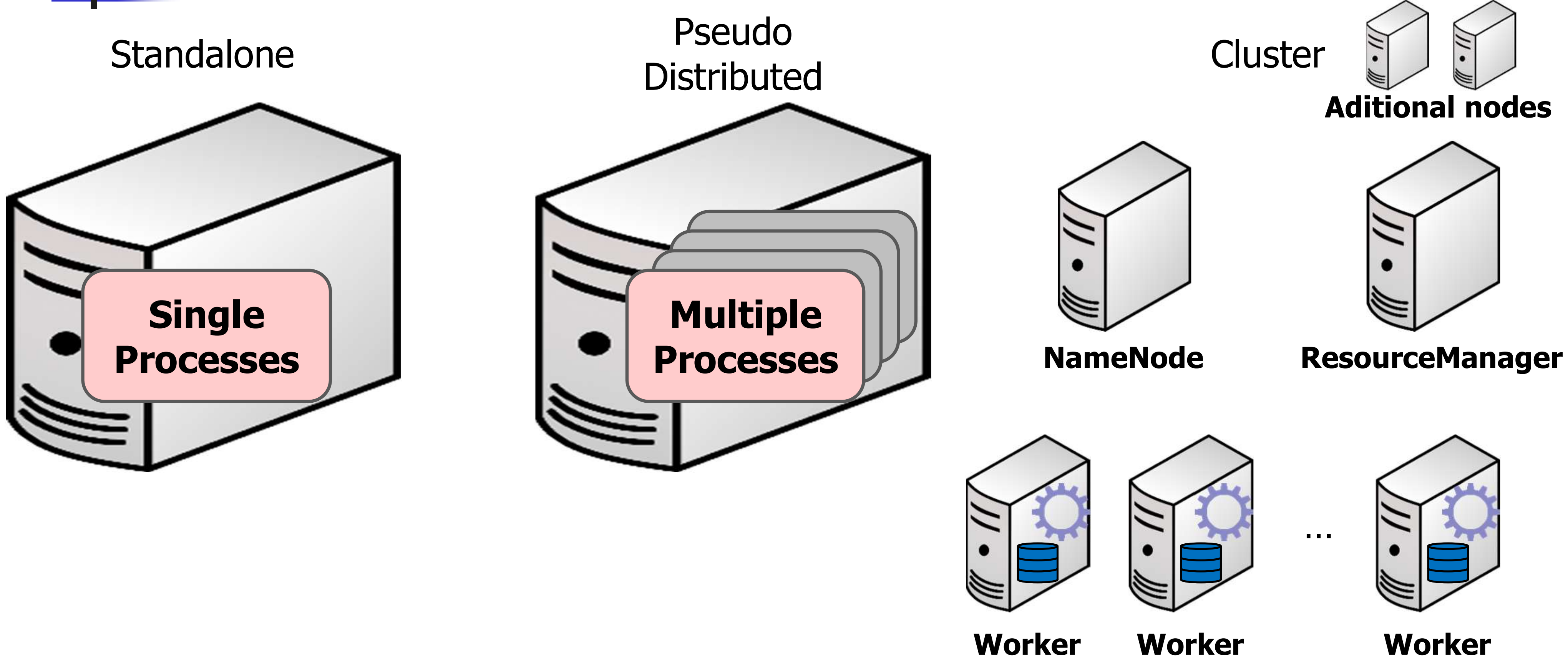
Installation and Configuration

ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Hadoop Modes

- Standalone
  - Run in a non-distributed mode, as a single Java process
- Pseudo Distributed
  - Each Hadoop daemon/service runs in a separate Java process
- Cluster
  - Computer clusters ranging from a few nodes to thousands
    - 1 node for the **NameNode**
    - 1 node for the **ResourceManager**
    - Aditional nodes (Web App Proxy Server or MapReduce Job History)
    - Remaining nodes act both as **DataNodes** and **NodeManager** (workers)

Hadoop Modes



Standalone

Single Processes

Pseudo Distributed

Multiple Processes

Cluster

Aditional nodes

NameNode

ResourceManager

Worker

Worker

…

Worker

Environment Used

- Linux Operating System
  - Examples presented for Ubuntu 18.04.3 LTS
- Java virtual machine
  - Java™ SE Runtime Environment (build 1.8.0_161-b12)
  - Maven
- Remote login using SSH (**S**ecure **SH**ell)
  - Login for cluster nodes without a passphrase
- Optionally **pdsh** (run multiple remote commands in parallel)

Additional Tools/Software

- wget
  - A non-interactive network downloader
    - wget [option]... [URL]...

```
wget https://archive.apache.org/dist/hadoop/common/hadoop-3.1.2/hadoop-3.1.2.tar.gz
```

- tar
  - An archiving utility with command line interface
    - tar {A|c|d|r|t|u|x}[GnSkUWOmpsMBiajJzZhPlRvwo] [ARG...]

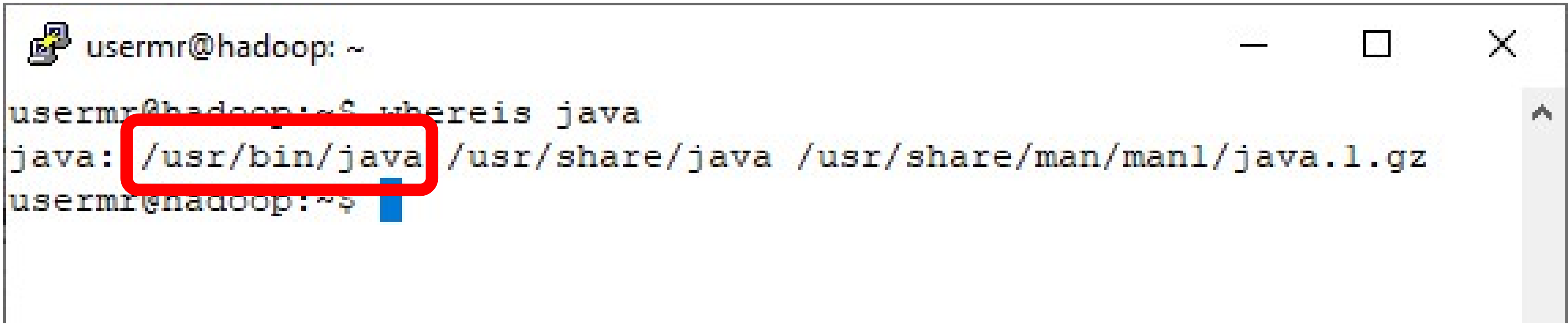
```
tar -xzf hadoop-3.1.2.tar.gz
```

Install Required Software

- Ensure system is update
  - `sudo apt update`
- Install ssh
  - `sudo apt install openssh-server`
  - `sudo service ssh start`
- Install
  - `sudo apt install openjdk-8-jdk-headless`

Install Required Software

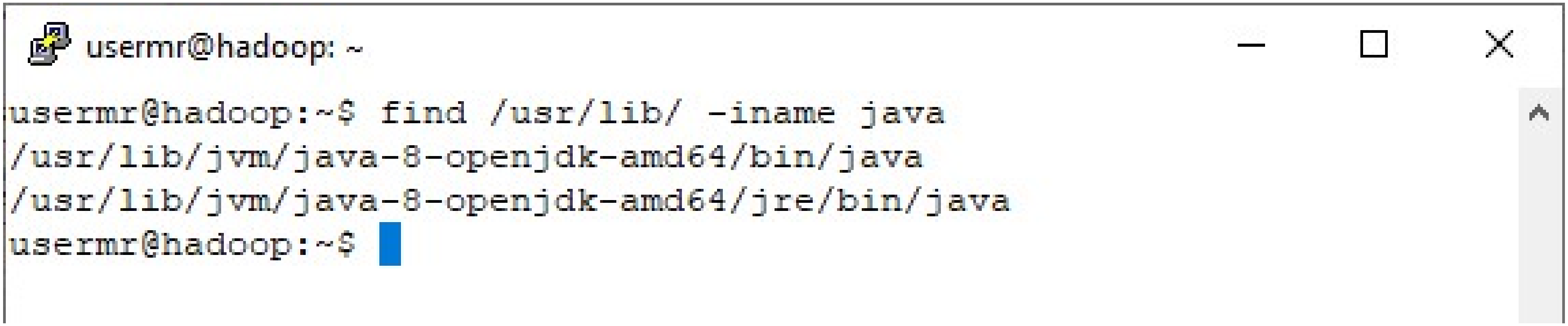
- Each user/process should have an environment variable named `JAVA_HOME` that represents the directory where java is installed



```
usermr@hadoop:~$ whereis java
java: /usr/bin/java /usr/share/java /usr/share/man/man1/java.1.gz
usermr@hadoop:~$
```

Install Required Software

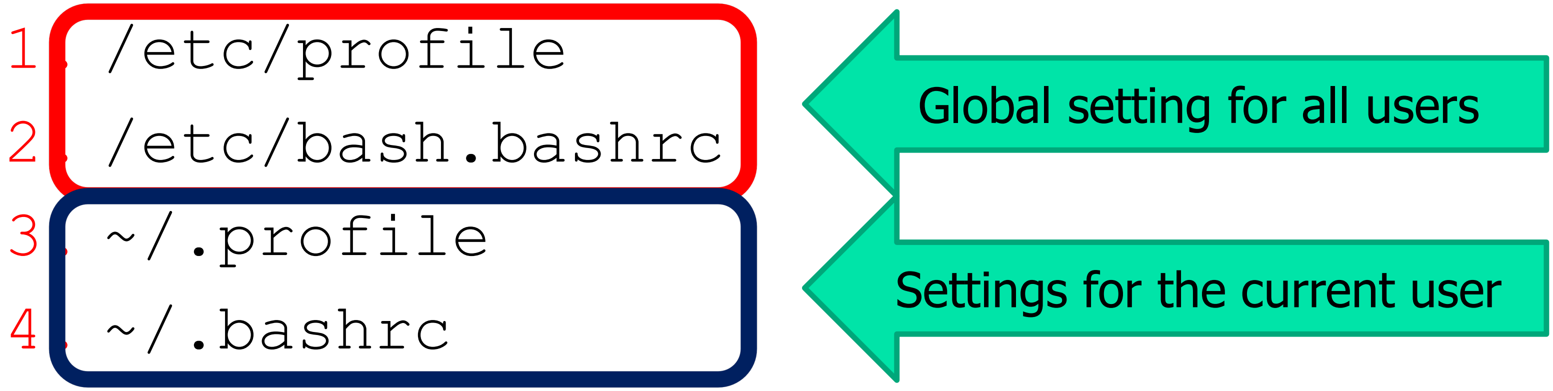
- Finding the location of a specific file can also be done with the `find` command:



```
usermr@hadoop:~$ find /usr/lib/ -iname java
/usr/lib/jvm/java-8-openjdk-amd64/bin/java
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
usermr@hadoop:~$
```

Install Required Software

- In Linux system environment variables can be configured for all users or for a particular user
- Every time a user performs an interactive login the following scripts are executed:



1. /etc/profile
2. /etc/bash.bashrc
3. ~/.profile
4. ~/.bashrc

Global setting for all users

Settings for the current user

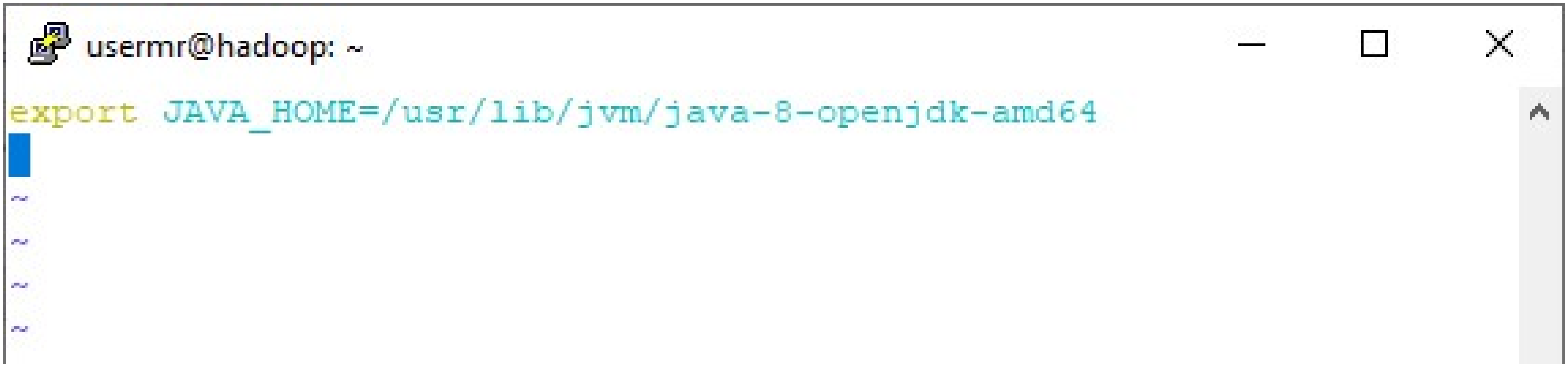
Install Required Software

- The execution of script `/etc/profile` execute all the files contained in directory `/etc/profile.d/`

- Specific settings, e.g. for defining an environment variable, that should be made available to all users, can be made in a file placed in the above directory

Install Required Software

- An example for Java
  - /etc/profile.d/jdk.sh



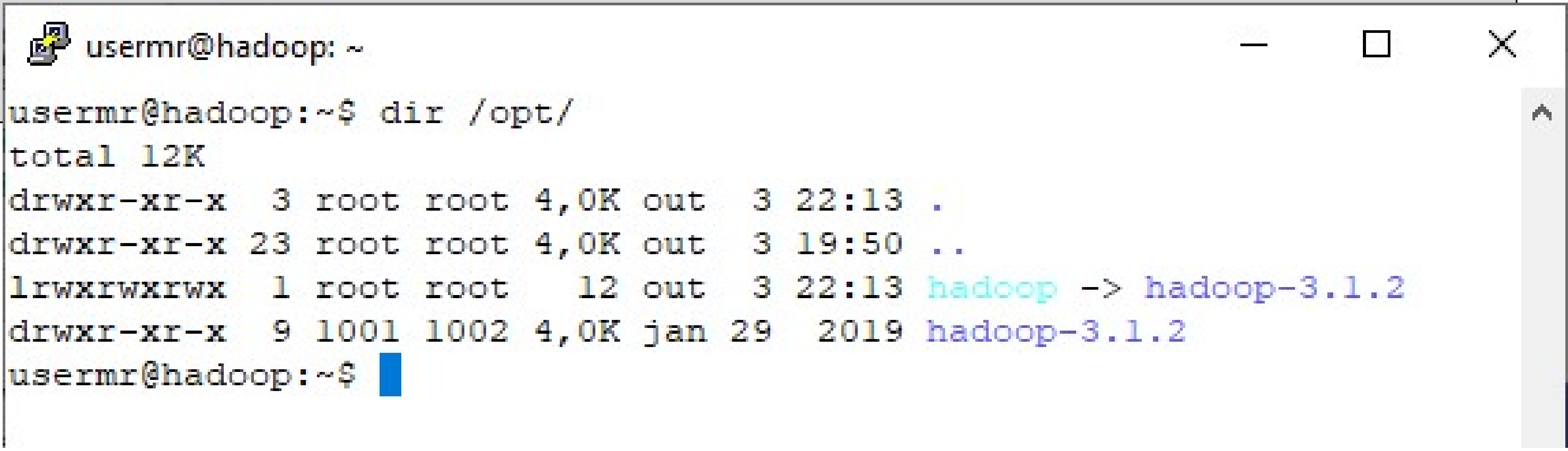
```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

~
~
~
~
```

Hadoop Standalone Mode

- Download Hadoop
  - Hadoop location
    - https://archive.apache.org/dist/hadoop/common/
  - Download
    - wget https://archive.apache.org/dist/hadoop/common/...
- Decompress downloaded file
  - tar -xzf …
- Define environment variable `HADOOP_HOME`
- Configure Hadoop with the location of `JAVA_HOME`

Hadoop
Standalone Mode

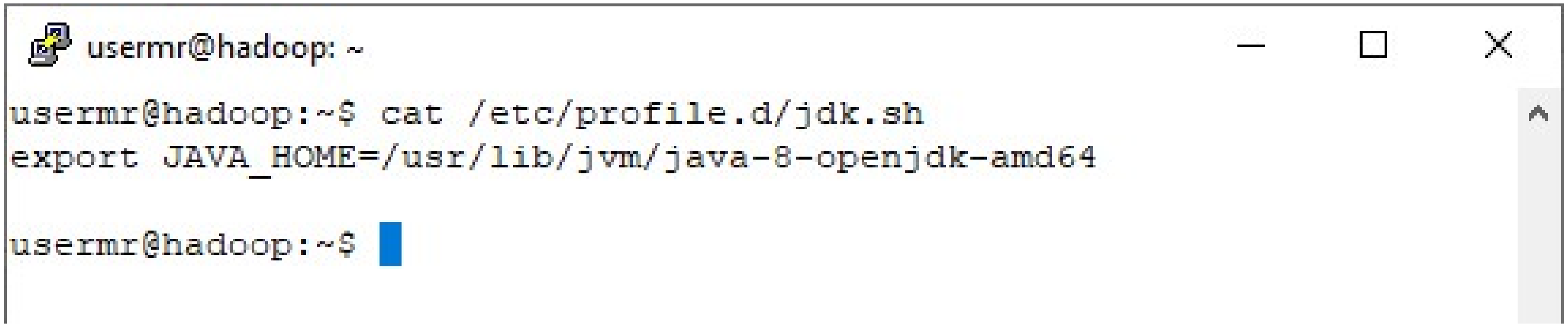
```
wget https://archive.apache.org/dist/hadoop/common/hadoop-3.1.2/hadoop-3.1.2.tar.gz

cd /opt

sudo tar -xzvf ~/hadoop-3.1.2.tar.gz

sudo ln -s hadoop-3.1.2 hadoop

dir /opt
```



```
usermr@hadoop:~$ dir /opt/
total 12K
drwxr-xr-x  3 root root 4,0K out  3 22:13 .
drwxr-xr-x 23 root root 4,0K out  3 19:50 ..
lrwxrwxrwx  1 root root   12 out  3 22:13 hadoop -> hadoop-3.1.2
drwxr-xr-x  9 1001 1002 4,0K jan 29  2019 hadoop-3.1.2
usermr@hadoop:~$
```

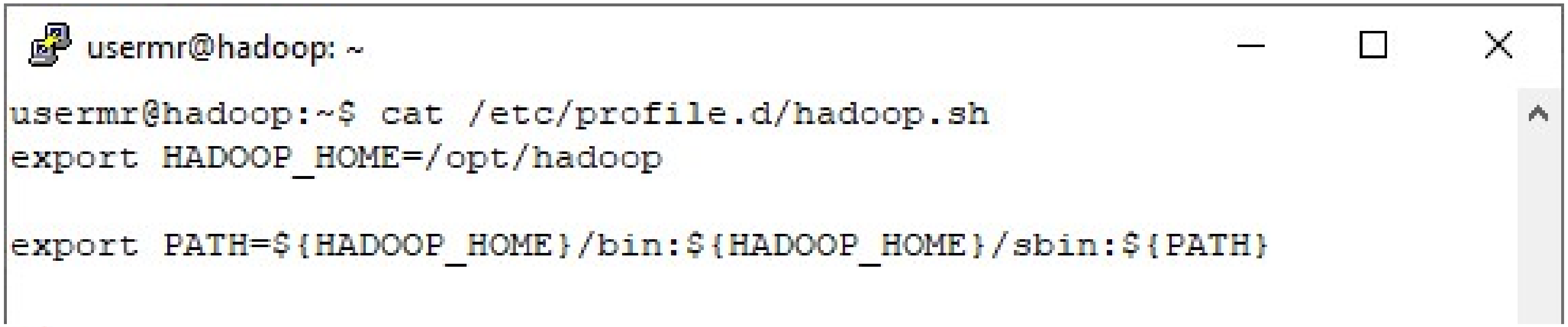
Hadoop Standalone Mode

- /etc/profile.d/jdk.sh



```
usermr@hadoop:~$ cat /etc/profile.d/jdk.sh
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

usermr@hadoop:~$ 
```

- /etc/profile.d/Hadoop.sh



```
usermr@hadoop:~$ cat /etc/profile.d/hadoop.sh
export HADOOP_HOME=/opt/hadoop

export PATH=${HADOOP_HOME}/bin:${HADOOP_HOME}/sbin:${PATH}
```

Testing Hadoop
Word Count – Map function

```
public class WordCountMapper extends Mapper<Object, Text, Text, IntWritable> {
  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(Object key, Text value, Context context)
    throws IOException, InterruptedException {

    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
    }
  }
}
```

Ex10

Testing Hadoop
Word Count – Reduce function

```
public class WordCountReducer extends Reducer<Text, IntWritable, Text,
IntWritable> {
  private IntWritable result = new IntWritable();

  public void reduce(Text key, Iterable<IntWritable> values, Context context)
    throws IOException, InterruptedException {

    int sum = 0;
    for (IntWritable val : values) {  sum += val.get();  }
    result.set(sum);
    context.write(key, result);
  }
}
```

Ex10

Testing Hadoop
Word Count – Application

```
public class WordCountApplication {

  public static void main(String[] args) throws Exception {
    if ( args.length!=2 ) {
     System.err.printf(
        "Usage: %s <input path> <output path>\n",
        WordCountApplication.class.getCanonicalName()       );
     System.exit( -1 );
    }

    Job job = new Job();
    job.setJarByClass( WordCountApplication.class );
    job.setJobName( "Word Count Ver 1" );
    ...
```

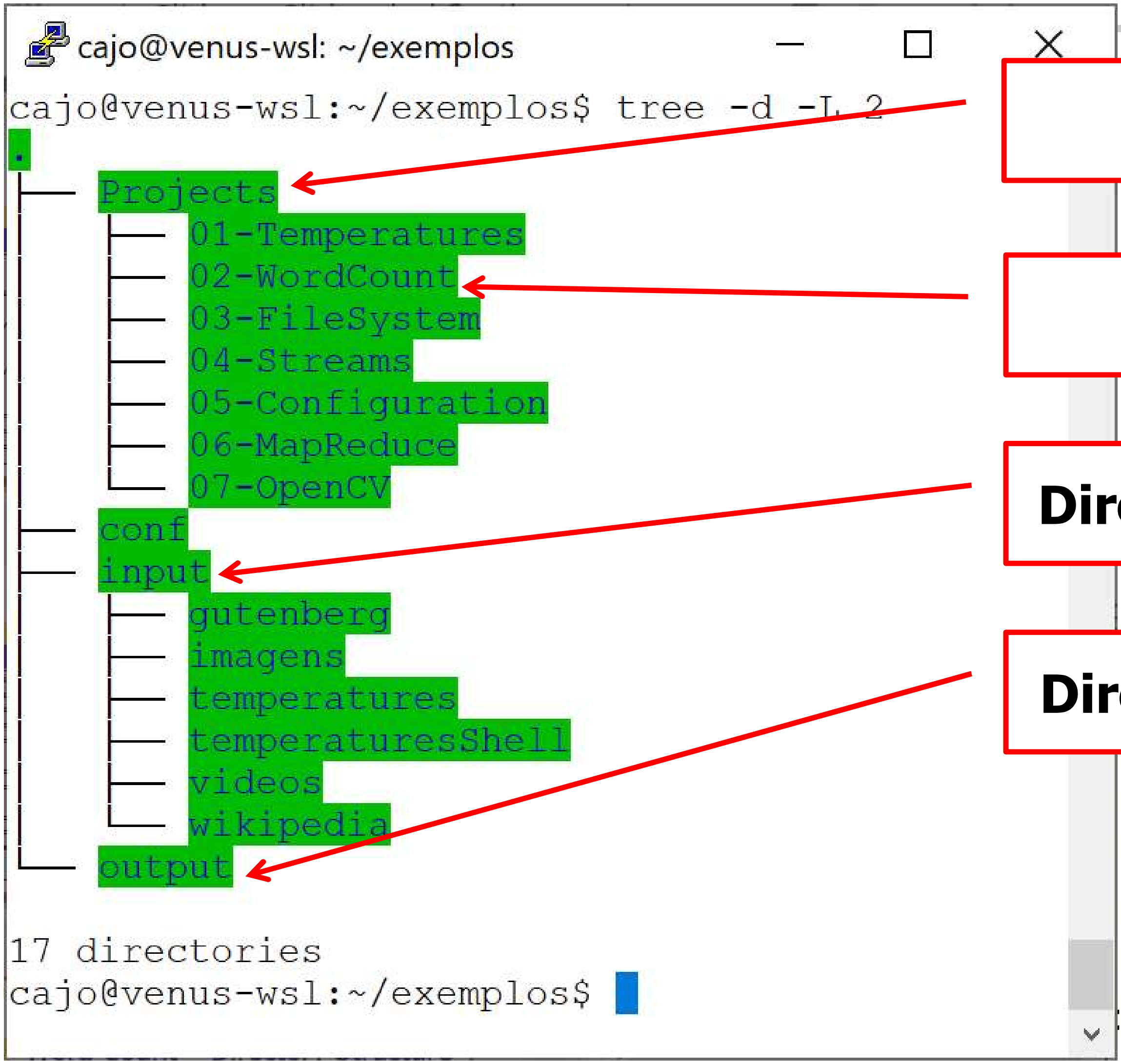
Ex10

Testing Hadoop Word Count – Application

```
    FileInputFormat.addInputPath(job, new Path(args[0]) );
    FileOutputFormat.setOutputPath(job, new Path(args[1]) );
    job.setMapperClass( WordCountMapper.class );
    job.setCombinerClass( WordCountReducer.class );
    job.setReducerClass( WordCountReducer.class );
    // Output types of map function
    job.setMapOutputKeyClass( Text.class );
    job.setMapOutputValueClass( IntWritable.class );
    // Output types of reduce function
    job.setOutputKeyClass( Text.class );
    job.setOutputValueClass( IntWritable.class );
    System.exit( job.waitForCompletion(true) ? 0 : 1 );
  }
}
```

Ex10

Examples
Directory Structure



```
cajo@venus-wsl:~/exemplos$ tree -d -L 2
.
├── Projects
│   ├── 01-Temperatures
│   ├── 02-WordCount
│   ├── 03-FileSystem
│   ├── 04-Streams
│   ├── 05-Configuration
│   ├── 06-MapReduce
│   └── 07-OpenCV
├── conf
├── input
│   ├── gutenberg
│   ├── imagens
│   ├── temperatures
│   ├── temperaturesShell
│   ├── videos
│   └── wikipedia
└── output

17 directories
cajo@venus-wsl:~/exemplos$
```

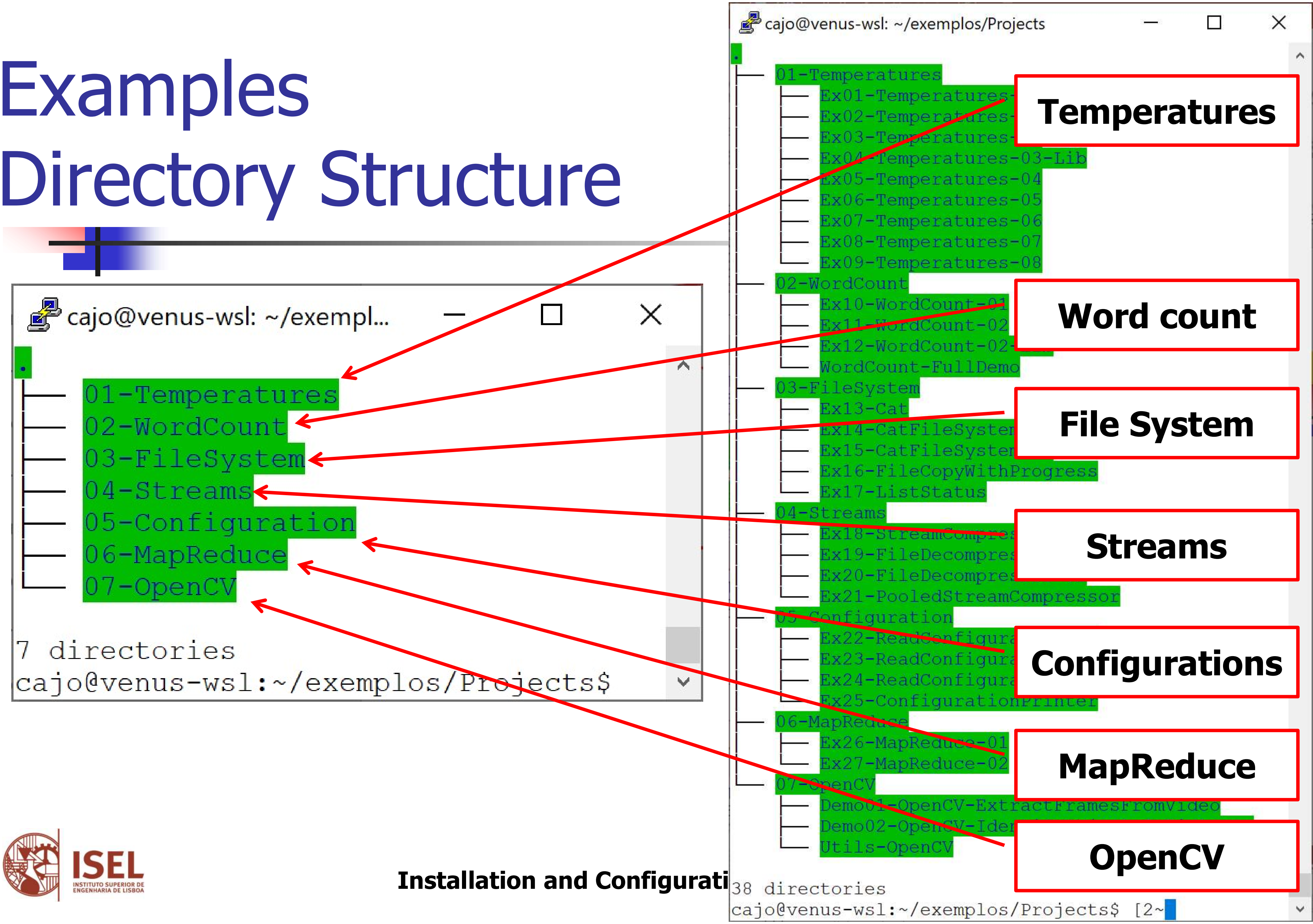
Examples

Word count examples

Directory with input data files

Directory for output data files

Examples Directory Structure



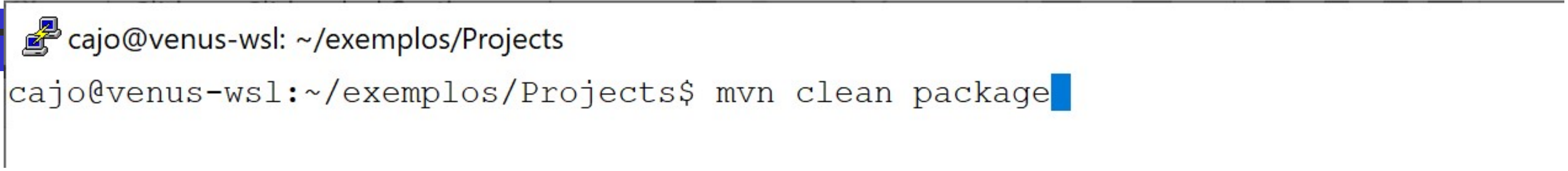
cajo@venus-wsl: ~/exempl...

```
.
├── 01-Temperatures
├── 02-WordCount
├── 03-FileSystem
├── 04-Streams
├── 05-Configuration
├── 06-MapReduce
└── 07-OpenCV

7 directories
cajo@venus-wsl:~/exemplos/Projects$
```

cajo@venus-wsl: ~/exemplos/Projects

```
.
├── 01-Temperatures
│   ├── Ex01-Temperatures-
│   ├── Ex02-Temperatures-
│   ├── Ex03-Temperatures-
│   ├── Ex04-Temperatures-03-Lib
│   ├── Ex05-Temperatures-04
│   ├── Ex06-Temperatures-05
│   ├── Ex07-Temperatures-06
│   ├── Ex08-Temperatures-07
│   └── Ex09-Temperatures-08
├── 02-WordCount
│   ├── Ex10-WordCount-01
│   ├── Ex11-WordCount-02
│   ├── Ex12-WordCount-02-
│   └── WordCount-FullDemo
├── 03-FileSystem
│   ├── Ex13-Cat
│   ├── Ex14-CatFileSyster
│   ├── Ex15-CatFileSyster
│   ├── Ex16-FileCopyWithProgress
│   └── Ex17-ListStatus
├── 04-Streams
│   ├── Ex18-StreamCompres
│   ├── Ex19-FileDecompres
│   ├── Ex20-FileDecompres
│   └── Ex21-PooledStreamCompressor
├── 05-Configuration
│   ├── Ex22-ReadConfigura
│   ├── Ex23-ReadConfigura
│   ├── Ex24-ReadConfigura
│   └── Ex25-ConfigurationPrinter
├── 06-MapReduce
│   ├── Ex26-MapReduce-01
│   └── Ex27-MapReduce-02
└── 07-OpenCV
    ├── Demo01-OpenCV-ExtractFramesFromVideo
    ├── Demo02-OpenCV-Ider
    └── Utils-OpenCV

38 directories
cajo@venus-wsl:~/exemplos/Projects$ [2~
```

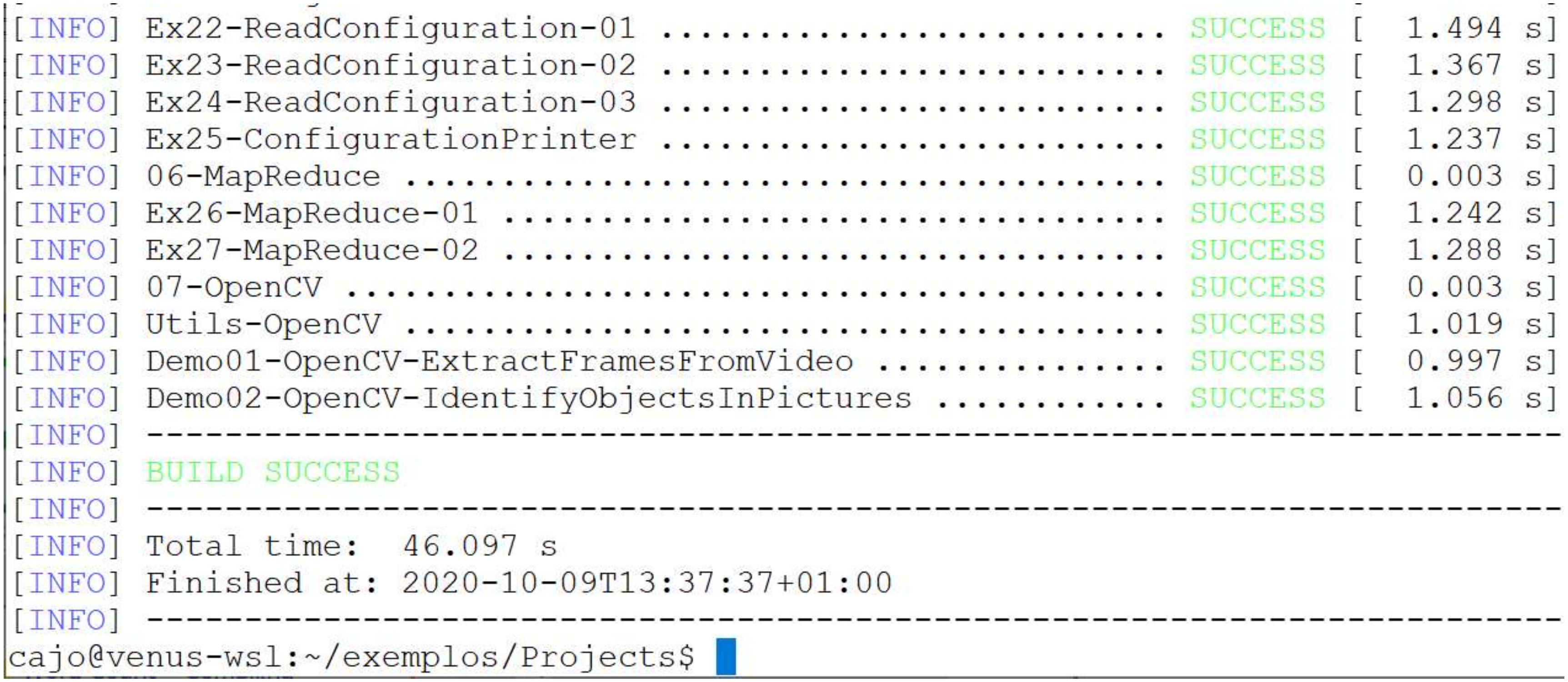
Examples Compiling



cajo@venus-wsl: ~/exemplos/Projects

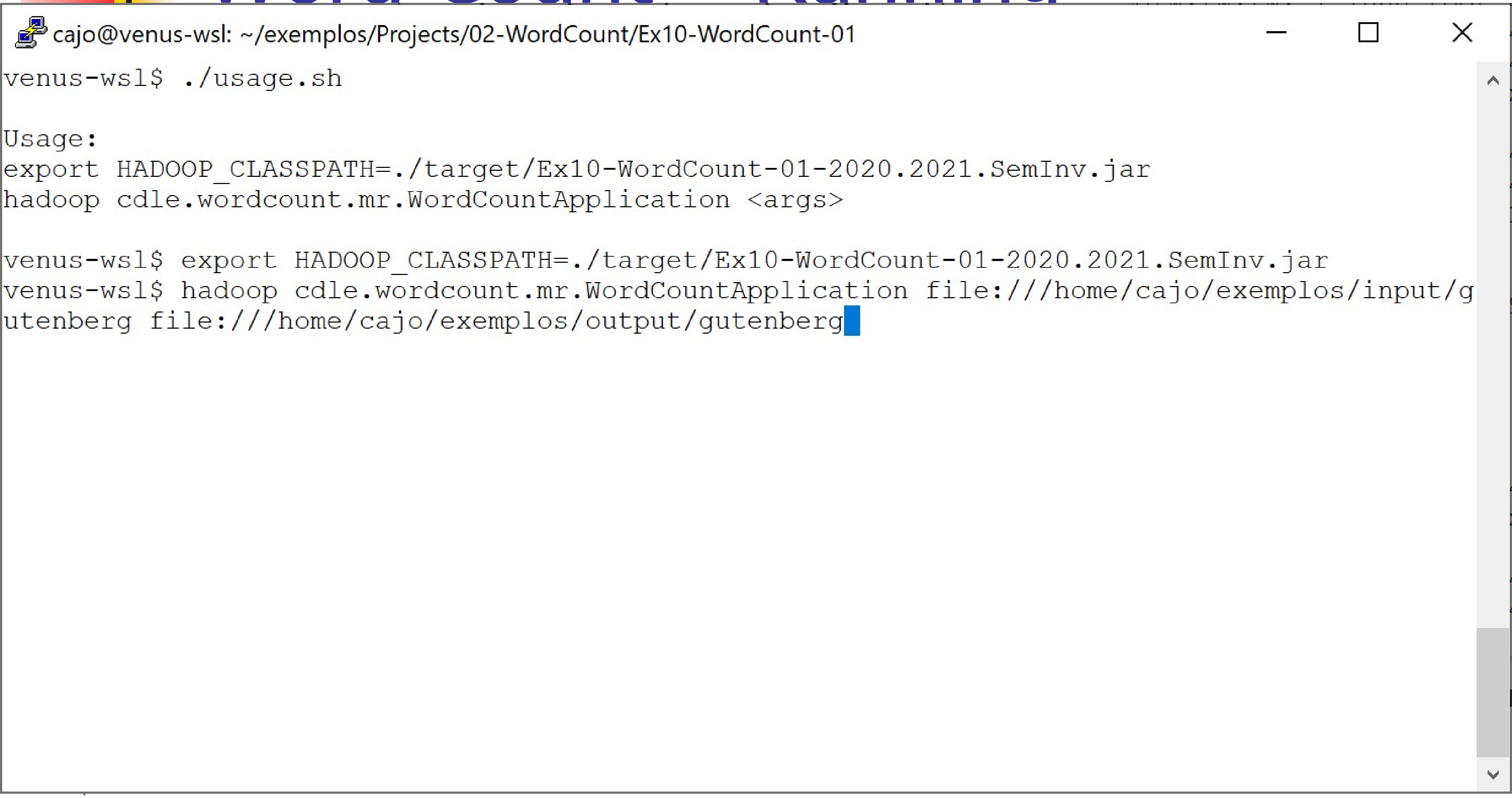
```
cajo@venus-wsl:~/exemplos/Projects$ mvn clean package
```

…



```
[INFO] Ex22-ReadConfiguration-01 ........................... SUCCESS [  1.494 s]
[INFO] Ex23-ReadConfiguration-02 ........................... SUCCESS [  1.367 s]
[INFO] Ex24-ReadConfiguration-03 ........................... SUCCESS [  1.298 s]
[INFO] Ex25-ConfigurationPrinter ........................... SUCCESS [  1.237 s]
[INFO] 06-MapReduce ........................................ SUCCESS [  0.003 s]
[INFO] Ex26-MapReduce-01 ................................... SUCCESS [  1.242 s]
[INFO] Ex27-MapReduce-02 ................................... SUCCESS [  1.288 s]
[INFO] 07-OpenCV ........................................... SUCCESS [  0.003 s]
[INFO] Utils-OpenCV ........................................ SUCCESS [  1.019 s]
[INFO] Demo01-OpenCV-ExtractFramesFromVideo ................ SUCCESS [  0.997 s]
[INFO] Demo02-OpenCV-IdentifyObjectsInPictures ............. SUCCESS [  1.056 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  46.097 s
[INFO] Finished at: 2020-10-09T13:37:37+01:00
[INFO] ------------------------------------------------------------------------
cajo@venus-wsl:~/exemplos/Projects$
```

Testing Hadoop Word Count – Running

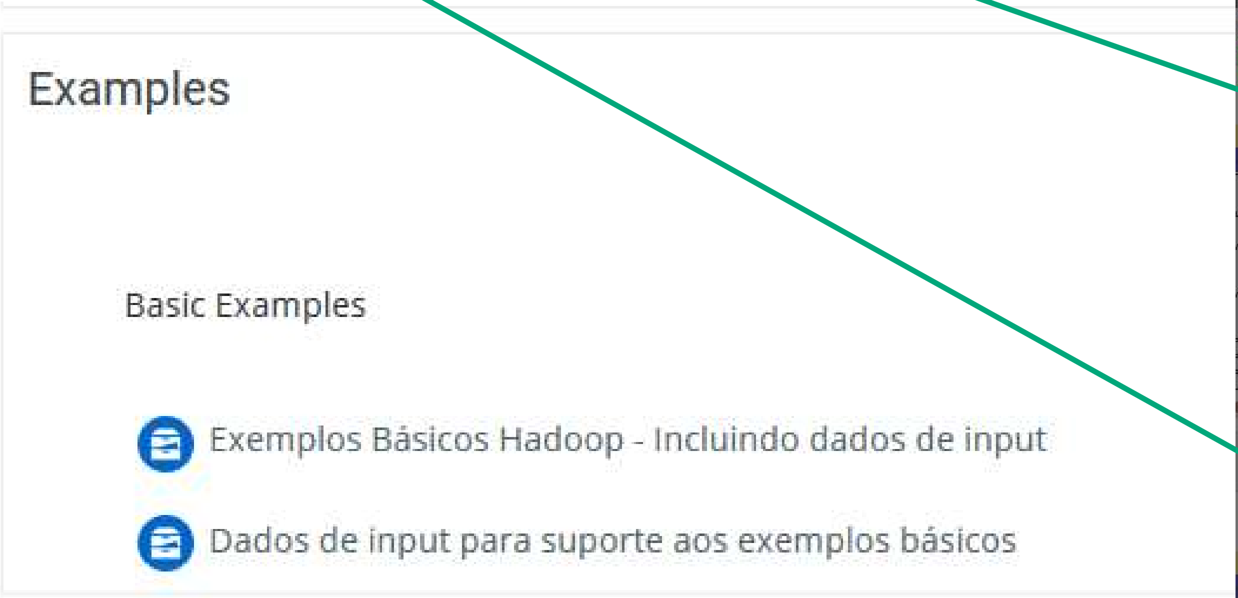


cajo@venus-wsl: ~/exemplos/Projects/02-WordCount/Ex10-WordCount-01

```
venus-wsl$ ./usage.sh

Usage:
export HADOOP_CLASSPATH=./target/Ex10-WordCount-01-2020.2021.SemInv.jar
hadoop cdle.wordcount.mr.WordCountApplication <args>

venus-wsl$ export HADOOP_CLASSPATH=./target/Ex10-WordCount-01-2020.2021.SemInv.jar
venus-wsl$ hadoop cdle.wordcount.mr.WordCountApplication file:///home/cajo/exemplos/input/g
utenberg file:///home/cajo/exemplos/output/gutenberg
```

Testing Hadoop Examples and Input Data

- Input data and examples are available in Moodle

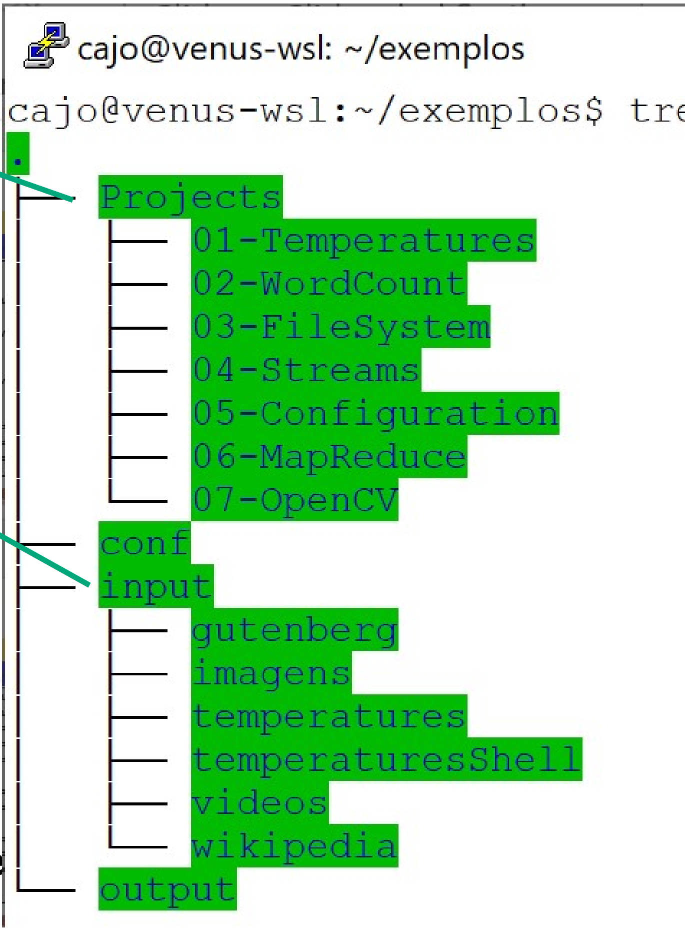


Examples

Basic Examples

Exemplos Básicos Hadoop - Incluindo dados de input

Dados de input para suporte aos exemplos básicos



cajo@venus-wsl: ~/exemplos

```
cajo@venus-wsl:~/exemplos$ tre
.
├── Projects
│   ├── 01-Temperatures
│   ├── 02-WordCount
│   ├── 03-FileSystem
│   ├── 04-Streams
│   ├── 05-Configuration
│   ├── 06-MapReduce
│   └── 07-OpenCV
├── conf
├── input
│   ├── gutenberg
│   ├── imagens
│   ├── temperatures
│   ├── temperaturesShell
│   ├── videos
│   └── wikipedia
└── output
```

Hadoop
Pseudo Distributed Mode

- When running in pseudo distributed mode each component of Apache Hadoop is executed as a different process/service.
- Each process/service is executed using different users:
  - Service HDFS – user `hdfs`
  - Service Resource Manager (YARN) – user `yarn`
  - Service for executing Map Reduce – user `hadoop`

Hadoop
Pseudo Distributed Mode

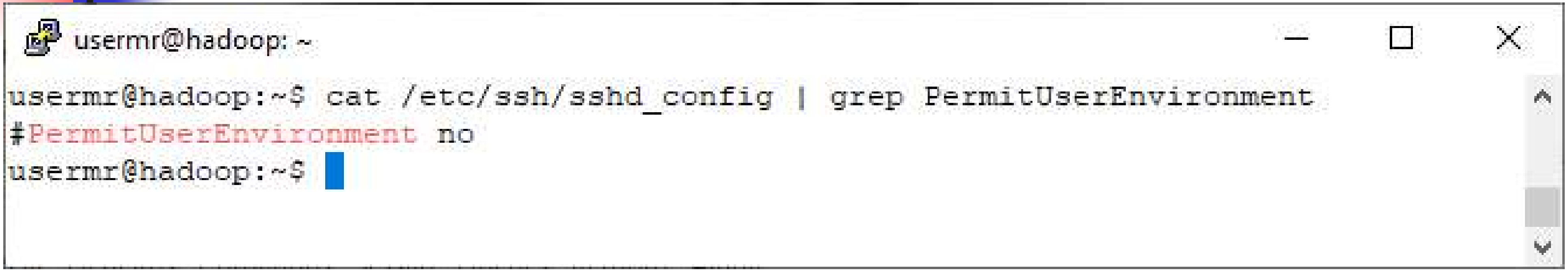
- In pseudo distributed mode it is recommended to separate the configurations settings used by the daemons/services from the default installation settings
- Location of these configurations settings is identified by the environment variable `HADOOP_CONF_DIR`
- In our case we are setting this variable to `/etc/hadoop`

Hadoop
Pseudo Distributed Mode

- Because pseudo distributed mode is a special case of a distributed installation we are going to use the service `SSH` (configured in a password less mode)

- Also because we are going to perform non interactive logins using `SSH` we need to configure `SSH` to enable de definition of user environment variables.

Hadoop
Pseudo Distributed Mode

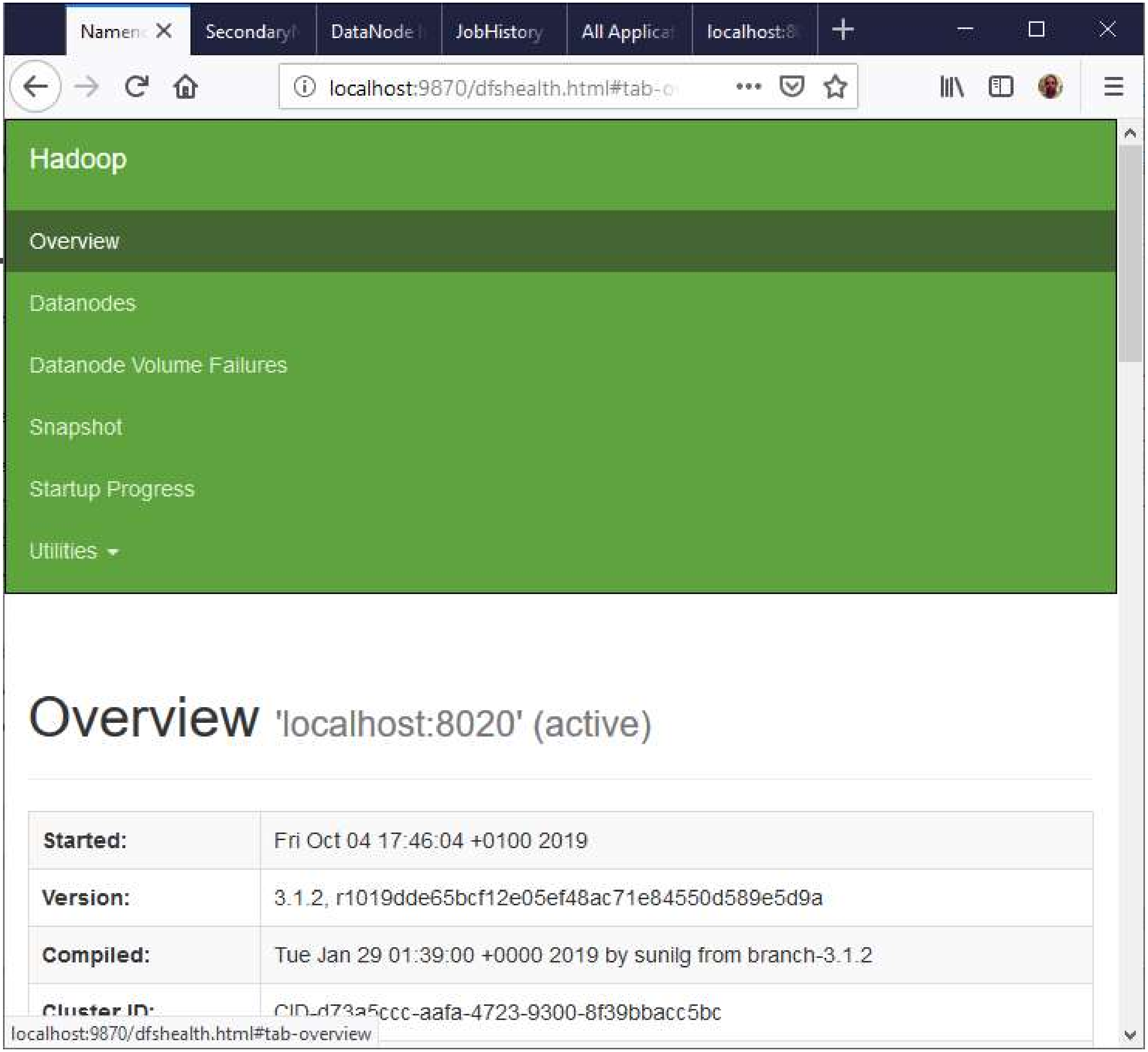


```
usermr@hadoop:~$ cat /etc/ssh/sshd_config | grep PermitUserEnvironment
#PermitUserEnvironment no
usermr@hadoop:~$
```

- Change the option `PermitUserEnvironment` to `yes` (and remove the # char from the beginning of the line) and restart the ssh service
  - `sudo service ssh restart`

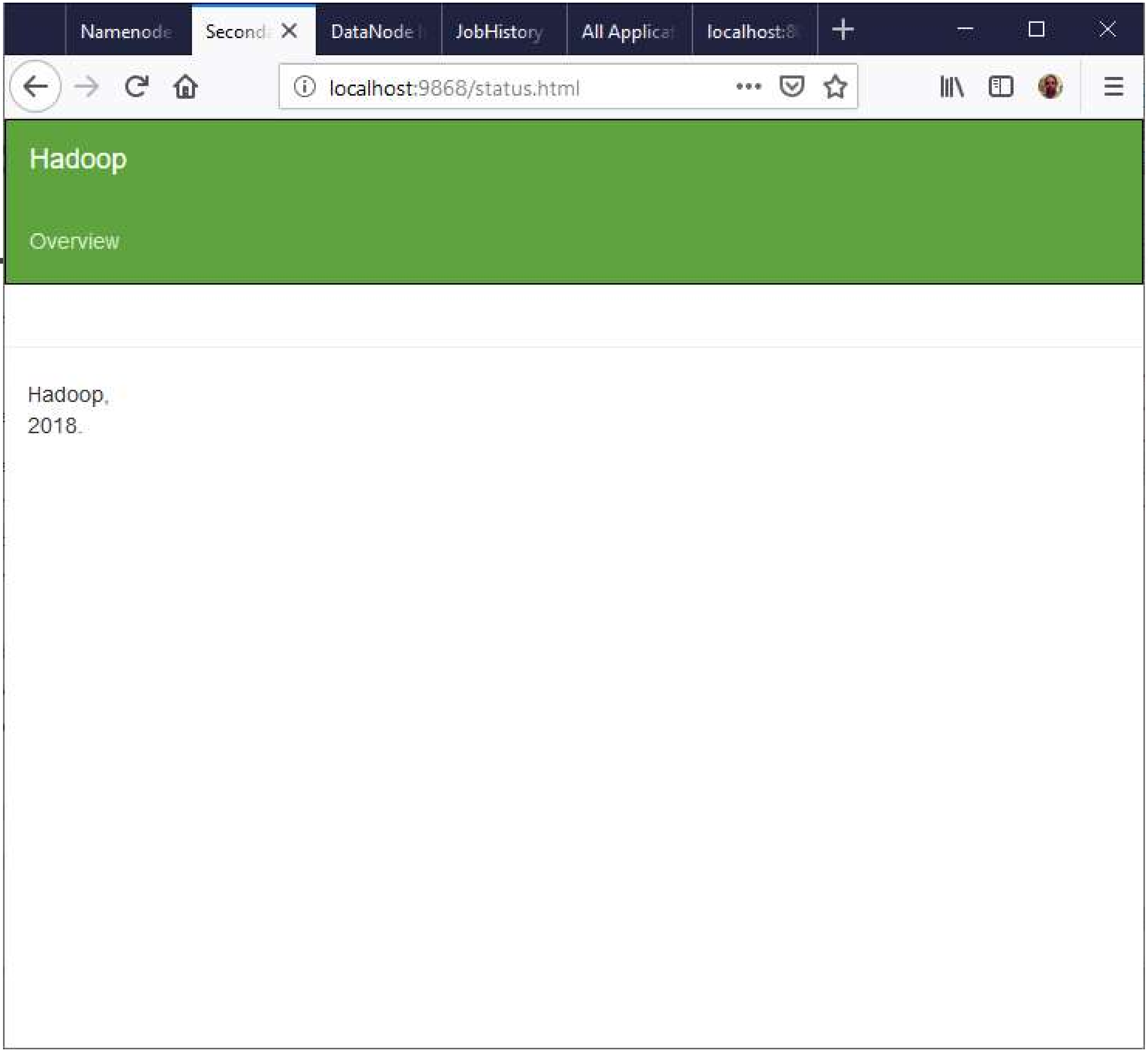
Hadoop
Pseudo Distributed Mode

- The scripts available in Moodle allow the installation of Hadoop in pseudo distributed modes using the followings steps
  1. Download the tar.gz file
  2. Extract to a directory
  3. Add that directory to the environment variable `PATH`
  4. Execute script `installHadoop.sh`



Overview 'localhost:8020' (active)

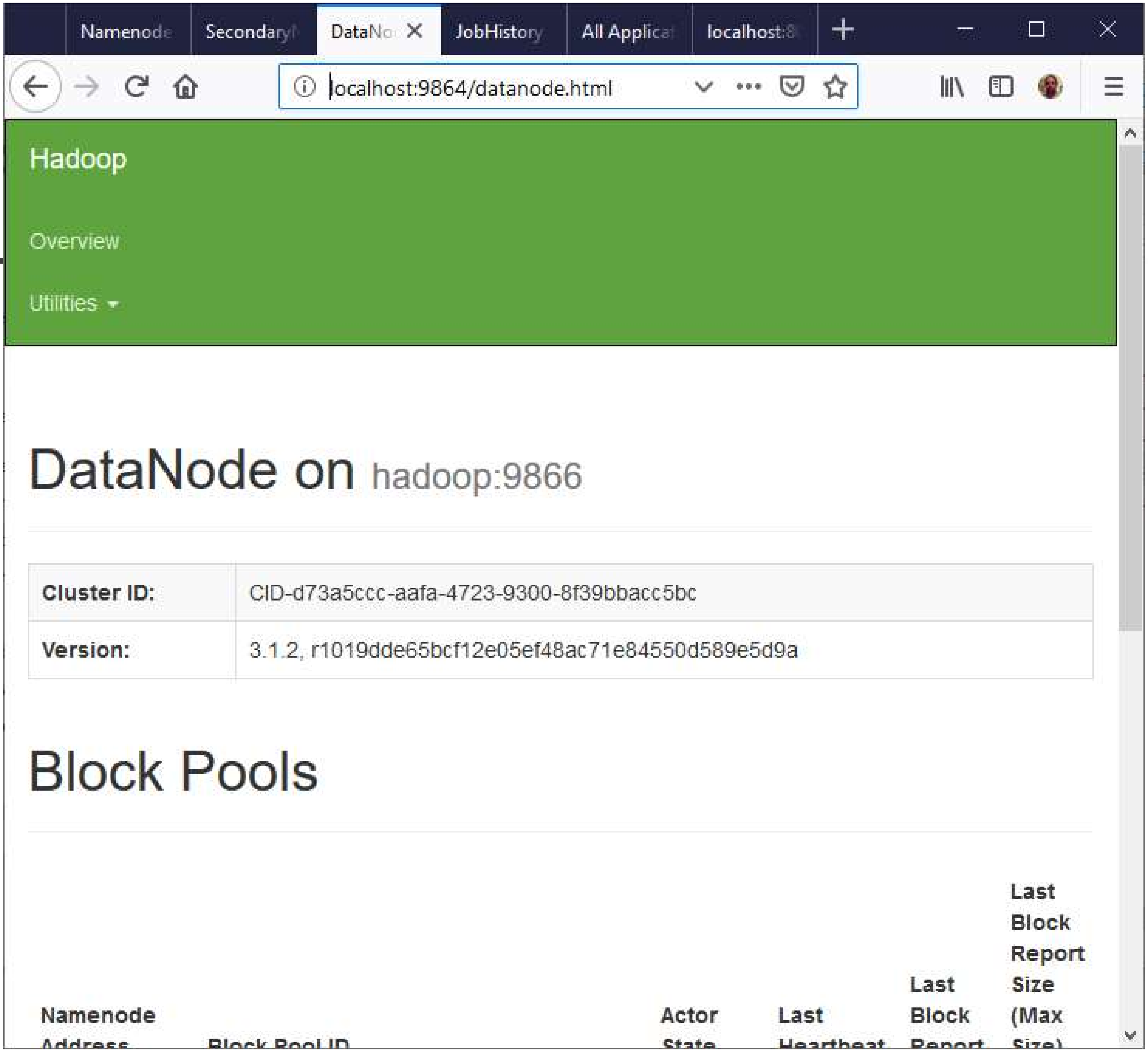
Started:	Fri Oct 04 17:46:04 +0100 2019
Version:	3.1.2, r1019dde65bcf12e05ef48ac71e84550d589e5d9a
Compiled:	Tue Jan 29 01:39:00 +0000 2019 by sunilg from branch-3.1.2
Cluster ID:	CID-d73a5ccc-aafa-4723-9300-8f39bbacc5bc



Hadoop

Overview

Hadoop,
2018.



Hadoop

Overview

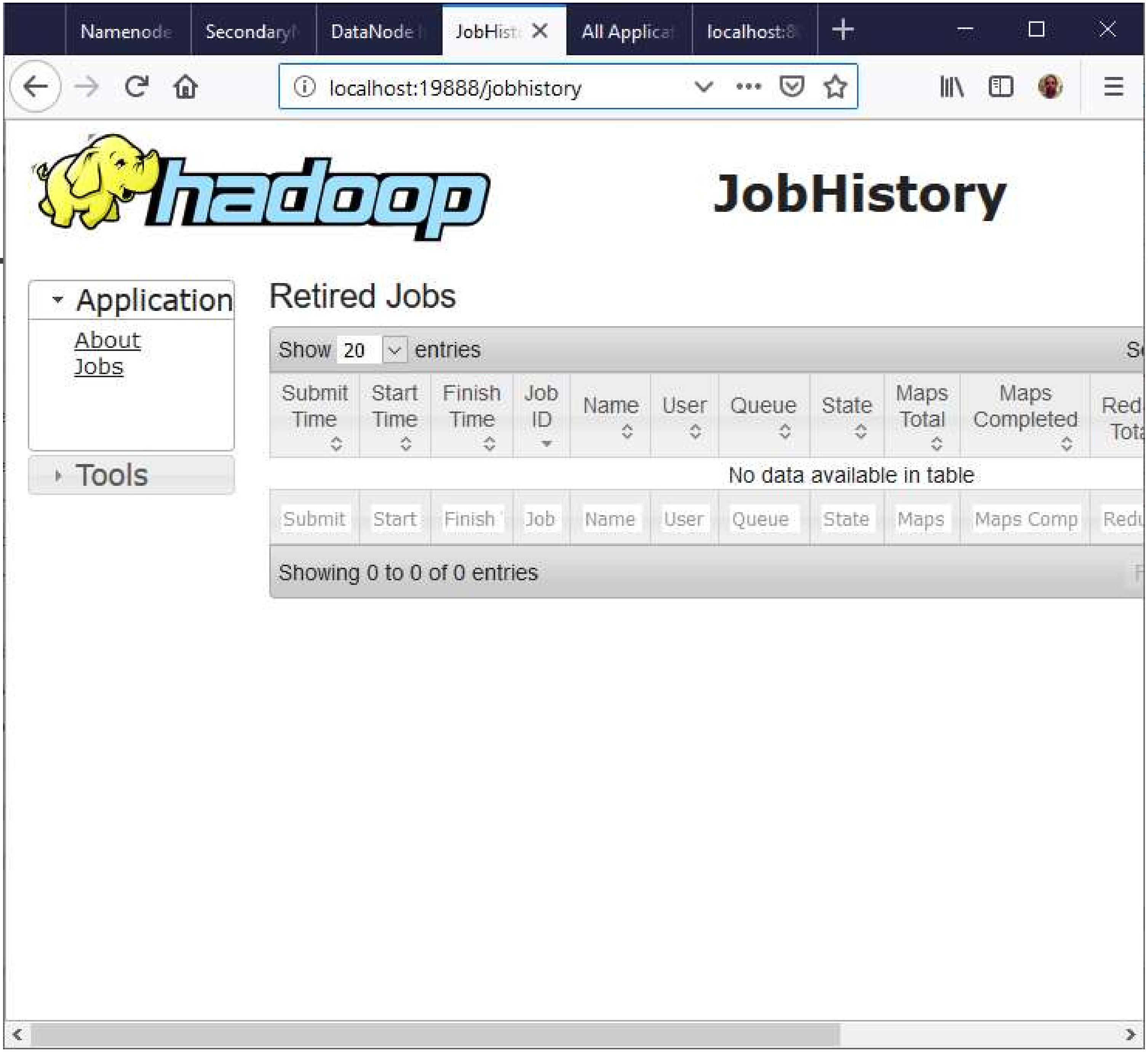
Utilities ▾

DataNode on hadoop:9866

Cluster ID:	CID-d73a5ccc-aafa-4723-9300-8f39bbacc5bc
Version:	3.1.2, r1019dde65bcf12e05ef48ac71e84550d589e5d9a

Block Pools

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)



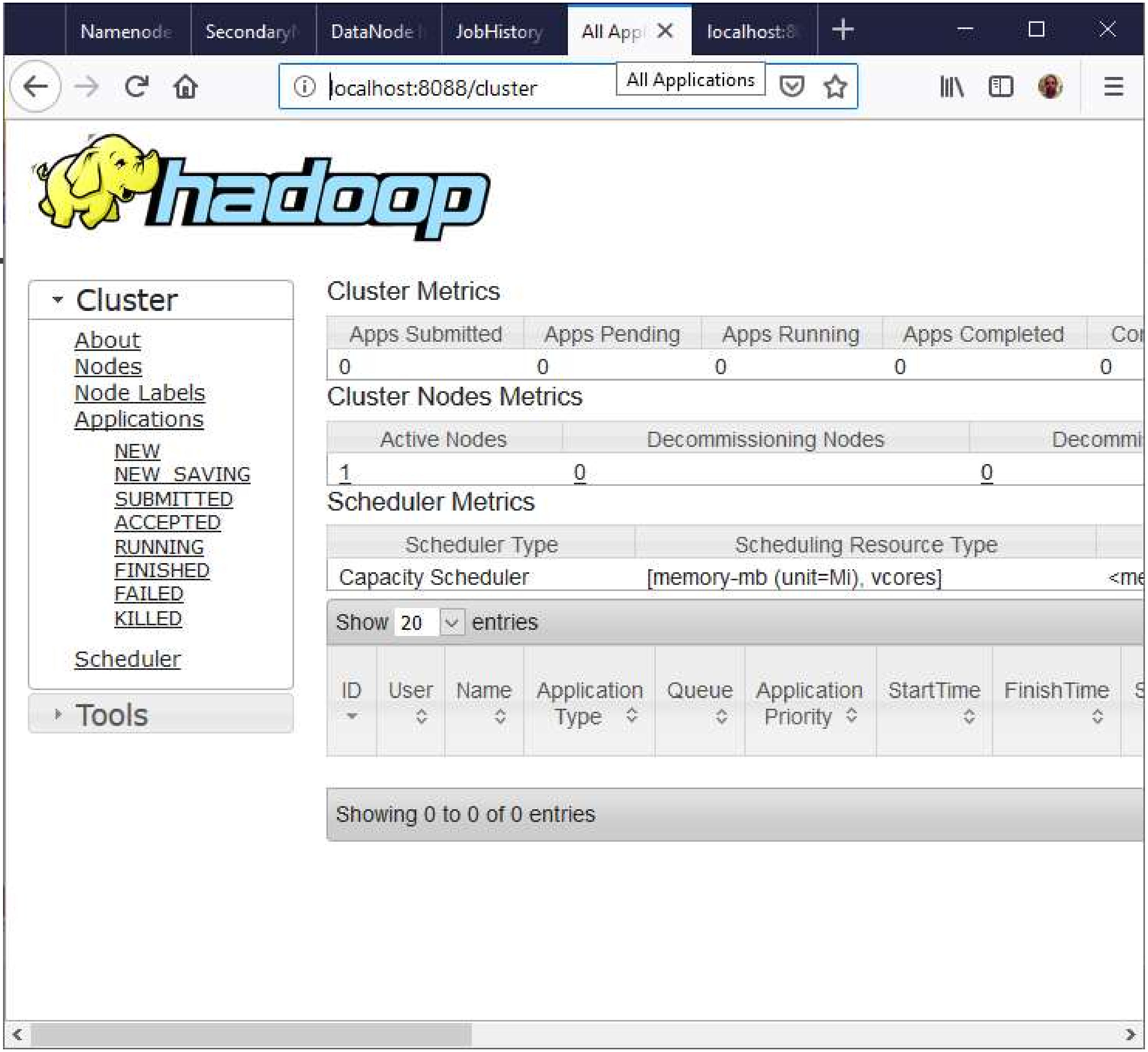
JobHistory

Retired Jobs

Show 20 entries

Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Red Tota
No data available in table										
Submit	Start	Finish	Job	Name	User	Queue	State	Maps	Maps Comp	Redu

Showing 0 to 0 of 0 entries



- Cluster
  - About
  - Nodes
  - Node Labels
  - Applications
    - NEW
    - NEW_SAVING
    - SUBMITTED
    - ACCEPTED
    - RUNNING
    - FINISHED
    - FAILED
    - KILLED
  - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Co
0	0	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommi
1	0	0

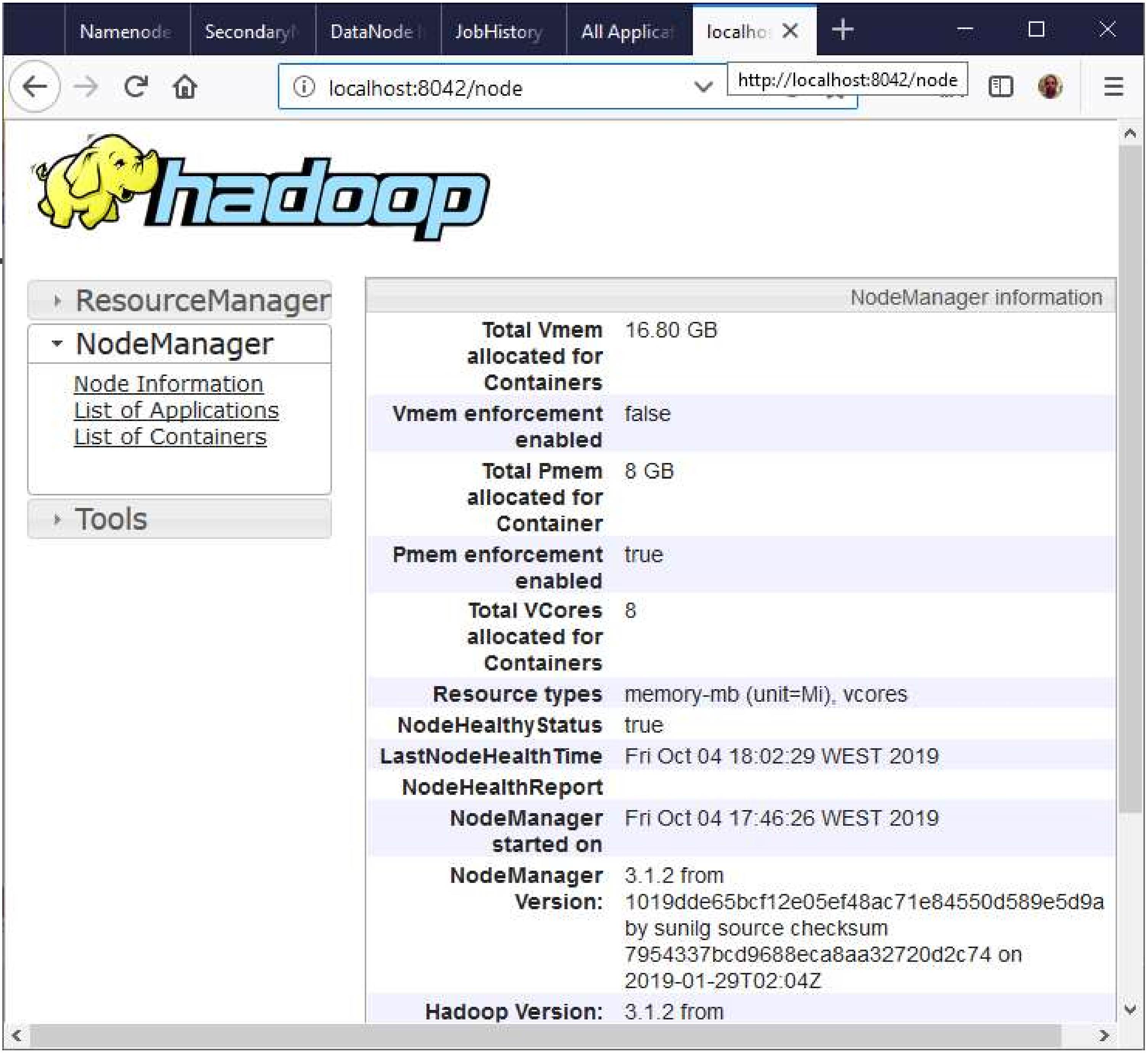
Scheduler Metrics

Scheduler Type	Scheduling Resource Type	
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<me

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	S

Showing 0 to 0 of 0 entries



NodeManager information

Total Vmem allocated for Containers	16.80 GB
Vmem enforcement enabled	false
Total Pmem allocated for Container	8 GB
Pmem enforcement enabled	true
Total VCores allocated for Containers	8
Resource types	memory-mb (unit=Mi), vcores
NodeHealthyStatus	true
LastNodeHealthTime	Fri Oct 04 18:02:29 WEST 2019
NodeHealthReport	
NodeManager started on	Fri Oct 04 17:46:26 WEST 2019
NodeManager Version:	3.1.2 from 1019dde65bcf12e05ef48ac71e84550d589e5d9a by sunilg source checksum 7954337bcd9688eca8aa32720d2c74 on 2019-01-29T02:04Z
Hadoop Version:	3.1.2 from

Next steps

- To access the Hadoop cluster @ ISEL:
  - Follow instructions in “AppendixA-HadoopCluster-ISEL”

- To install Hadoop in a pseudo distributed mode using Docker:
  - Follow instructions in “AppendixB-HadoopWithDocker”