

# The ContactSunny Blog

Tech from one dev to another

**FLAT 50% OFF**

ON SELLING FEE\* ON  
ALL CATEGORIES!

Crores of  
Customers Bhi  
Aadhi Selling  
Fees\* Bhi



You are here ►

Home > Data Science >

## Installing Hadoop On Windows 11 With WSL2

Data Science by Sunny Srinidhi - November 1, 2021  3



In the previous post, we saw how to install a Linux distro on Windows 11 using WSL2 and then how to install Zsh and on-my-zsh to make the terminal more customizable. In this post, we'll see how we can install the complete Hadoop environment on the same Windows 11 machine using WSL.

---

### Installing Dependencies

There are two important dependencies that you'll need to install to make Hadoop work. These aren't optional, unless you have them installed already. So make sure you install these dependencies.

# Download

Fast Download

MediaGet

Download

## Installing JDK

The first dependency is java development kit, or JDK. It's recommended to go with Java 8, or Java 1.8 for Hadoop. This recommendation is coming from me, as I have had problems with newer versions of Java. But you can definitely give newer versions a shot.

Also, it doesn't matter if you install Oracle JDK or Open JDK, or any other version of JDK. You just need to have it installed. I used the following command to install JDK 8 on the Debian Linux that I've installed on Windows 11:

```
sudo apt install adoptopenjdk-8-hotspot
```

For this package to be available in the apt repository, you'll first need to add the PPA. For that run the following command:

```
sudo add-apt-repository --yes https://adoptopenjdk.jfrog.io/adoptopenjdk/deb/
```

Once you have JDK installed, make sure you export the path of the JDK with the env variable name JAVA\_HOME. The export command looks like this:

# Download

MediaGet

Downr

```
export JAVA_HOME=/usr/lib/jvm/adoptopenjdk-8-hotspot-amd64/bin/java
```

---

If you just run this command in your terminal, the variable will be exported only for the current session. To make it permanent, you'll have to add this command to the `.zshrc` file.

## Installing OpenSSH

The next dependency to install is OpenSSH so that Hadoop can SSH into the localhost. This also is a necessary dependency. Without SSH into localhost, most components of Hadoop wouldn't work. To install OpenSSH, run the following commands in the terminal:

```
sudo apt install openssh-server openssh-client -y
```

Once we have both the server and the client installed for SSH, we have to generate keys for authentication. For this, run the following command and go through the instructions that you'll get:

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
```

Once the keys are generated, you have to copy them over to the list of authorized keys so that you don't have to enter a password each time you SSH into the machine. This is especially important because this is what Hadoop expects. At least I haven't seen an option to change this behavior. So, run the following command to *cat* the file contents of the key file we just created, and then copy that to the *authorized\_keys* file:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Now, make sure that the public key file has the right permission. This is because if the key file has more public access than needed, the system would think that the key can be copied or duplicated or tampered with, which would mean the key is not secure. This will make the system refuse the key and not allow SSH login. So run the following command to set the right permissions:

```
chmod 0600 ~/.ssh/id_rsa.pub
```

Next, start the SSH service so that we can test if the the server is working fine. For this, run the following command:

### Ahmedabad - Dubai

₹20,450

BOOK NOW

### Mumbai - Dubai

₹16,296

BO

### Chennai - Singapore

₹17,175

BOOK NOW

### Bengaluru - Shillong

₹18,772

BO

```
sudo service ssh start
```

If everything till here worked as expected, you're golden. Finally, run the following command to make sure that SSH is working as expected:

```
ssh localhost
```

If everything works as expected, you should see something similar to the screenshot below:

```
+ sunny sudo service ssh start
[sudo] password for sunny:
Starting OpenBSD Secure Shell server: sshd.
+ sunny ssh localhost
Linux Sunny-Zephyrus-Windows 5.10.60.1-microsoft-standard-WSL2 #1 SMP Wed Aug 25 23:20:18 UTC 2021 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Oct 27 21:17:41 2021 from 127.0.0.1
+ ~
```

Awesome. You can exit back to your previous session by hitting the key combination CTRL + d. This concludes the dependency installation phase. Let's now move on to installing Hadoop.

## Installing Hadoop

### Downloading Hadoop



First step to installing Hadoop is to actually download it. As of this writing, the latest version of Hadoop is version 3.3.1, and you can download it from [here](#). You will be downloading a *.tar.gz* file from there. To decompress that file, use the following command:

```
tar xzf hadoop-3.3.1.tar.gz
```

This will create a directory named `hadoop-3.3.1` and place all files and directories inside that directory. Because we're installing Hadoop on our local machine, we're going to do a single-node deployment, which is also known as pseudo-distributed mode deployment.

## Setting the environment variables

We have to set a bunch of environment variables. The best part is, you have to customize only one variable. The others are just copy-paste. Anyway, following are the variables I'm talking about:

<b>Ahmedabad - Dubai</b>	
₹20,450	BOOK NOW
<b>Mumbai - Dubai</b>	
₹16,296	BO
<b>Chennai - Singapore</b>	
₹17,175	BOOK NOW
<b>Bengaluru - Shillong</b>	
₹18,772	BO

```
export HADOOP_HOME=/mnt/d/bigdata/hadoop-3.3.1
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$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:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/nativ"
```

As you can see, you only have to change the value of the first environment variable, *HADOOP\_HOME*. Set it to reflect the path where you have placed the Hadoop directory. Also, it is a good idea to place these *export* statements in the *.zshrc* file so that these variables are exported every time automatically instead of you having to do it. Once you place it in the file, make sure you source it so that it takes effect immediately:

```
source ~/.zshrc
```

## Configuring Hadoop

Next, we'll have to edit a few files to change the config for various Hadoop components. Let's start that with the file *hadoop-env.sh*. Run the following command to open the file in the editor:

```
sudo vim $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

Next, find the line that is exporting the `$JAVA_HOME` variable and uncomment it. Here, you have to provide the same path that you did when you installed Java earlier. For me, that's the following:

```
export JAVA_HOME=/usr/lib/jvm/adoptopenjdk-8-hotspot-amd64/bin/java
```

Next, we have to edit the *core-site.xml* file. Here we have to provide the temporary directory for Hadoop and also the default name for the Hadoop file system. Open the file in the editor using the following command:

```
sudo vim $HADOOP_HOME/etc/hadoop/core-site.xml
```

You'll find an empty file here with a few comments and an empty configuration block. You can delete everything and replace it with the following:

```
<configuration>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/mnt/d/hdfs/tmp</value>
  </property>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://127.0.0.1:9000</value>
  </property>
</configuration>
```

Make sure you create the temp directory that you configure here. Next, we have to edit the HDFS config file *hdfs-site.xml*. To do this, open the file in the editor using the following command:

```
sudo vim $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

In this configuration file, we are setting the HDFS data node directory, HDFS name node directory, and the HDFS replication factor. Here again you should get a file with an empty configuration block. Replace that with the following:

## Download the Whitepaper

Learn Why Antivirus Solutions Are Designed For Yesterday's Attacks.

CrowdStrike®

```
<configuration>
  <property>
    <name>dfs.data.dir</name>
    <value>/mnt/d/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/mnt/d/hdfs/datanode</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

And again, make sure you create the data node and name node directories. Next, we have the MapReduce config file. To open this in the editor, run the following command:

```
sudo vim $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

You can replace the configuration block with the following:

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

As you can see, it's a simple configuration which specifies the MapReduce framework name. And finally, we have the YARN configuration file, *yarn-site.xml*. Open the file in the editor using:

```
sudo vim $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

Add the following configuration to the file:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>127.0.0.1</value>
  </property>
  <property>
    <name>yarn.acl.enable</name>
    <value>0</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
  </property>
</configuration>
```

There's nothing to change in this configuration. And finally, we're done configuring Hadoop. We can now move on to formatting the name node and starting Hadoop.

<b>Ahmedabad - Dubai</b>		<b>Mumbai - Dubai</b>	
₹20,450	BOOK NOW	₹16,296	BO
<b>Chennai - Singapore</b>		<b>Bengaluru - Shillong</b>	
₹17,175	BOOK NOW	₹18,772	BO

## Formatting the HDFS name node

It's important to first format the HDFS name node before starting the Hadoop service the first time. This, obviously, makes sure there's no junk anywhere in the name node. And once you start using HDFS more frequently, you'll realize that you're formatting the name node more often than you thought you would, at least on your development machine. Anyway, to format the name node, use the following command:



```
hdfs namenode -format
```

Once you get the shutdown notification for name node, the formatting is complete.

## Starting all of Hadoop

Finally, we're at the best part of this activity, starting and using Hadoop. Now, there are many ways of starting Hadoop depending on what components you actually want to use. For example, you can start only YARN, or HDFS along with it, etc. For this activity, we'll just start everything. To do this, the Hadoop distribution provides a handy script. And because you have already exported a bunch of environment variables earlier, you don't even have to search for that script, it's already in your path. Just run the following command and wait for it to finish:

```
start-all.sh
```

This will take a few seconds, as the script just waits for the first 10 seconds without doing anything to give you an option to cancel the operation if you started it by mistake. Just hold on and you should see output similar to the following screenshot:

```
→ programs start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as sunny in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [Sunny-Zephyrus-Windows]
2021-11-01 16:51:27,982 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting resourcemanager
Starting nodemanagers
→ programs |
```

This tells us that all components of Hadoop are up and running. To make sure, if you want to, you run the *jps* command to get a list of all the processes running. You should see at least the following services:

```
→ programs jps
977 ResourceManager
353 NameNode
438 DataNode
615 SecondaryNameNode
1499 Jps
→ programs |
```

And that's it. You're now running Hadoop on your Windows 11 PC using a Linux distro on WSL 1 or 2. To make sure, you can use the following simple HDFS command:

```
hdfs dfs -ls /
```

This command will list all files and directories at the root HDFS directory. If it's a brand new deployment, you shouldn't find much there. You'll get a list similar to the one shown below:

```

→ programs hdfs dfs -ls /
2021-11-01 16:54:44,132 WARN util.NativeCodeLoader: Unable to load
Found 2 items
drwx-wx-wx   - sunny supergroup          0 2021-10-10 11:14 /tmp
drwxr-xr-x   - sunny supergroup          0 2021-10-10 11:13 /user
→ programs |

```

That's pretty much it. We're done!

And if you like what you see here, or on my Medium blog and Dev.To blog, and would like to see more of such helpful technical posts in the future, consider supporting me on Patreon and Github.

● Become a patron

## About the author



### Sunny Srinidhi

Coding, reading, sleeping, listening, watching, potato. INDIAN.

"If you don't have time to do it right, when will you have time to do it over?" – John Wooden

See author's posts



**Get Hands-on  
Training on  
Salesforce Dev &  
Admin. Become a  
Certified  
Salesforce Expert.**

Edureka

Op

Tagged big data bigdata data science hadoop how to install hdoop install hadoop install hadoop on widows 11 install hadoop on windows 11 install hadoop on wsl install hadoop windows 11 wsl install hadoop wsl2 install windows wsl 2 installing hadoop machine learning programming tech technology windows 11 windows 11 hadoop windows11 windows11 hadoop



## Sunny Srinidhi

Coding, reading, sleeping, listening, watching, potato. INDIAN. "If you don't have time to do it right, when will you have time to do it over?" – John Wooden  
<https://blog.contactsunny.com>

< **Previous article**

Installing Zsh and Oh-my-zsh on Windows 11 with WSL2

**Next article** >

Installing Hadoop on the new M1 Pro and M1 Max MacBook Pro

### *3 thoughts on “Installing Hadoop on Windows 11 with WSL2”*

Pingback: Installing Hadoop on the new M1 Pro and M1 Max MacBook Pro



February 6, 2022 at 6:54 PM

Hello Sunny,

**Andreas Wickenhäuser** thank you for your instructions. I have two comments:

1. The last line of setting the environment variables looks weird: `export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/nativ"`

The equals sign is missing and I suppose the last character of the path is missing.

2. My Java path is: `"/usr/lib/jvm/java-8-openjdk-amd64"`. So compared to your setting, `"bin/java"` is missing at the end.  
Andreas.

Reply



April 24, 2022 at 4:34 PM

**Rachel** Hi Sunny, Great writeup... However i think adoptjdk has changed now to adoptium now. Could you please help with the changed post. And what updates should be done.

Thanks

Reply

**Leave a Reply**

Comment \*



Name \*

Email Address \*

Website

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.

Post Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

## Enjoy the limited-time Mega Offer

Sell on Amazon

### MY UPCOMING O'REILLY LIVE ONLINE COURSES

O'REILLY®

Next Course:

## Getting Started With Amazon Aurora

Dates:

- September 3rd, 2021
- December 10th, 2021



FOLLOW ME



### SUBSCRIBE TO BLOG VIA EMAIL

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Join 23 other subscribers

### RECENT POSTS

» [Optimising Hive Queries with Tez Query Engine](#)

- » Cleaning and Normalizing Data Using AWS Glue DataBrew
- » The Dunning-Kruger Effect In Tech
- » Understanding Apache Hive LLAP
- » Installing Hadoop on the new M1 Pro and M1 Max MacBook Pro

## CATEGORIES

- 📁 Data Science
- 📁 General
- 📁 Rants
- 📁 Smartphones
- 📁 Tech



## ARCHIVES

- 📅 June 2022
- 📅 January 2022
- 📅 November 2021
- 📅 October 2021
- 📅 September 2021
- 📅 June 2021
- 📅 March 2021
- 📅 February 2021
- 📅 January 2021
- 📅 December 2020

- October 2020
- August 2020
- July 2020
- June 2020
- May 2020
- April 2020
- March 2020
- February 2020
- January 2020
- December 2019
- November 2019
- October 2019
- September 2019
- June 2019
- May 2019
- April 2019
- November 2018
- August 2018
- July 2018
- August 2017
- July 2017
- June 2017
- April 2017
- March 2017
- February 2017
- January 2017
- September 2016
- August 2016
- March 2016

## Pet Accessories Shop Near Me

Augie Pet Store Bangalore

### TAGS

ai amazon apache apache kafka apache spark apis artificial intelligence aws best practices big data bigdata coding data  
science datascience data structure implementation in java data structures feature reduction feature selection hadoop  
java java data structures java data structures implementation javascript kafka laravel linked list in java linked lists machine learning  
machine learning models ml natural language processing nlp php programming python scikit python sklearn rants scikit scikit  
learn sklearn spring spring boot tech technology the fasttext series

## Pet Accessorie Shop Near Me

Shop now



**BSV ACADEMY**

**Learn about  
the original  
Satoshi protocol  
and why it matters.**



© 2022

Powered by WordPress | Theme: AccessPress Mag

[Data Science](#)

[Tech](#)

[General](#)

[Proof of Concepts \(POCs\)](#)

[About Me / Products](#)

[Must Watch Videos](#)