**Data Analytics with Spark Using Python**

# **Preface**

* Spark is at the heart of the disruptive Big Data and open source software revolution
* The interest in and use of Spark have grown exponentially, with no signs of abating

## **The focus of the Book**

* The fundamentals of the Spark project (including the runtime, cluster, and application architecture)
* The Python API for Spark (PySpark)
* The scope of this book is quite broad, covering aspects of Spark from core Spark programming to Spark SQL, Spark Streaming, machine learning, and more

## **How to Use This Book**

|  |  |
| --- | --- |
| Part 1: Spark Foundations | Chapter 1 Introducing Big Data, Hadoop, and Spark |
| Chapter 2 Deploying Spark |
| Chapter 3 Understanding the Spark Cluster Architecture |
| Chapter 4 Learning Spark Programming Basics |
| Part 2: Beyond the basic | Chapter 5 Advanced Programming Using the Spark Core |
| Chapter 6 SQL and NoSQL Programming with Spark |
| Chapter 7 Stream Processing and Messaging Using Spark |
| Chapter 8 Introduction to Data Science and Machine Learning Using Spark |

## **About the Author**

* Jeffrey Aven is an independent Big Data, open-source software, and cloud computing professional based out of Melbourne, Australia.
* Jeffrey is a highly regarded consultant and instructor and has authored several other books including Teach Yourself Apache Spark in 24 Hours and Teach Yourself Hadoop in 24 Hours

*Sample data and source code for each of the exercises in this book, which is available at http://sparkusingpython.com. You can also view or clone the GitHub repository for this book at https://github.com/sparktraining/spark\_using\_python.*

# **Chapter 2 – Deploying Spark**

|  |  |
| --- | --- |
| In This Chapter:   * Overview of the different Spark deployment modes * How to install Spark * The contents of a Spark installation * Overview of the various methods available for deploying Spark in the cloud   Chương này trình bày những kiến ​​thức cơ bản về cách triển khai Spark, cách cài đặt Spark và cách khởi động và chạy các cụm Spark.  Nó thảo luận về các chế độ triển khai và bộ lập lịch khác nhau có sẵn cho các cụm Spark, cũng như các tùy chọn để triển khai Spark trên đám mây. |  |

Apache Spark là một khung máy tính phân tán có hỗ trợ tích hợp để xử lý hàng loạt và luồng dữ liệu lớn, hầu hết quá trình xử lý đó xảy ra trong bộ nhớ mang lại hiệu suất tốt hơn. Nó có các mô-đun tích hợp cho SQL, học máy, xử lý đồ thị, v.v.

## **Spark Deployment Modes**

|  |  |
| --- | --- |
| Local mode | * Local mode cho phép tất cả các quy trình Spark chạy trên một máy duy nhất, tùy chọn sử dụng bất kỳ số core nào trên hệ thống cục bộ. * Sử dụng Local mode thường là một cách nhanh chóng để kiểm tra cài đặt Spark mới và nó cho phép nhanh chóng kiểm tra các quy trình Spark dựa trên các tập dữ liệu nhỏ. * Khi chạy Spark Local mode, bạn có thể truy cập bất kỳ dữ liệu nào trên local filesystem. * Mặc dù Local mode cho phép bạn thiết lập và chạy nhanh chóng, nhưng nó bị hạn chế về khả năng mở rộng và hiệu quả đối với các trường hợp sử dụng sản xuất. |
| Spark Standalone | * Spark Standalone đề cập đến bộ lập lịch tích hợp hoặc "độc lập" * Nhiều quy trình hoặc dịch vụ máy chủ lưu trữ, chạy trong một cụm Spark Standalone và mỗi dịch vụ đóng một vai trò trong việc lập kế hoạch, điều phối và quản lý một ứng dụng Spark nhất định đang chạy trên cụm      * Với Spark Standalone, có thể bắt đầu và vận hành nhanh chóng mà không cần phải phụ thuộc hoặc cân nhắc về môi trường. * Mỗi bản phát hành Spark bao gồm mọi thứ cần để bắt đầu, bao gồm tệp nhị phân và tệp cấu hình cho bất kỳ máy chủ lưu trữ nào để đảm nhận bất kỳ vai trò cụ thể nào trong một cụm Spark Standalone. |
| Spark on YARN (Hadoop) | * Phương pháp triển khai phổ biến nhất cho Spark là sử dụng khung quản lý tài nguyên YARN được cung cấp cùng với Hadoop. YARN là thành phần cốt lõi của Hadoop cho phép bạn lập lịch và quản lý khối lượng công việc trên một cụm Hadoop. * Các quy trình Spark như Driver, Master, and Executors được lưu trữ hoặc hỗ trợ bởi các quy trình YARN (Resource Manager, Node Manager, and Application Master) * Có hai chế độ triển khai cụm khi sử dụng YARN làm bộ lập lịch: cluster and client |
| Spark on Mesos | * Apache Mesos là open source cluster manager developed được phát triển tại Đại học California, Berkeley * Mesos có khả năng lập lịch các loại ứng dụng khác nhau, cung cấp chia sẻ tài nguyên chi tiết giúp sử dụng cụm hiệu quả hơn. |

## **Preparing to Install Spark**

### Getting Spark

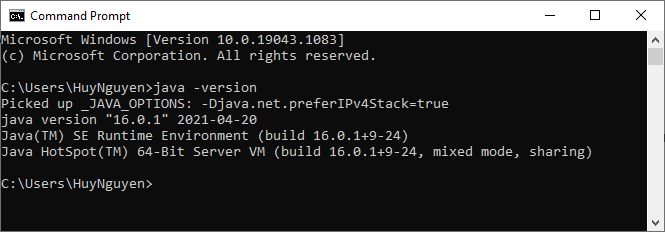
* Using a Spark release is often the easiest way to install Spark on a given system. Spark releases are downloadable from <http://spark.apache.org/downloads.html>.
* These releases are cross-platform: They target a JVM environment, which is platform agnostic.

|  |  |  |
| --- | --- | --- |
| * Linux * Windows * Mac OS X | * 8 GB RAM and more * 8 or more CPU core * 10 GB or greater network speed * Sufficient local disk space for storage | * Java (JDK preferably) (8./11. ~~16.~~) * Python/R |

### Installing Spark on Linux or Mac OS X

### Installing Spark on Windows

1. Download và cài đặt Java

* Download và cài đặt JDK  [tại đây](https://www.oracle.com/technetwork/java/javase/downloads/index.html)
* Thiết lập biến môi trường JAVA\_HOME với đường dẫn tới thư mục jdk (ví dụ: C:\Program Files\Java\jdk1.8.0\_191) trong Enviroment Variable (vào Control Panel, chọn System rồi chọn Advanced System settings ở góc bên góc trái)
* Thêm thư mục bin của jdk (%JAVA\_HOME%\bin) vào trong biến môi trường Path.
* Lưu lại các thiết lập rồi mở cửa sổ command để kiểm tra bằng lệnh java -version.

2. Download và cài đặt Scala/Python

* Download và cài đặt Scala [tại đây](https://www.scala-lang.org/download/) (kéo xuống chọn Binary for windows)
* Thiết lập biến môi trường SCALA\_HOME và Path tương tự như phần cài đặt Java
* Kiểm tra thiết lập bằng lệnh scala -version.

3. Download Spark

* Download Spark [tại đây](https://spark.apache.org/downloads.html) rồi tiến hành tiến hành giải nén file
* Thiết lập biến môi trường SPARK\_HOME và Path tương tự như trên

4. Download Winutils

* + Tiến hành download Winutils [tại đây](https://github.com/steveloughran/winutils), chú ý lựa trọn phiên bản Hadoop phù hợp với phiên bản đã lựa trọn trong phần download Spark
  + Click vào phiên bản Hadoop phù hợp, kéo xuống bên dưới để download file winutils.exe
  + Lưu file winutils.exe vào một thư mục bất kỳ và tạo biến môi trường HADOOP\_HOME với đường dẫn tới thư mục đó (nếu khi chạy spark bị lỗi không load được Winutils thì xem lại đường dẫn của Winutils vì có thể nó cần được để trong thư mục bin)

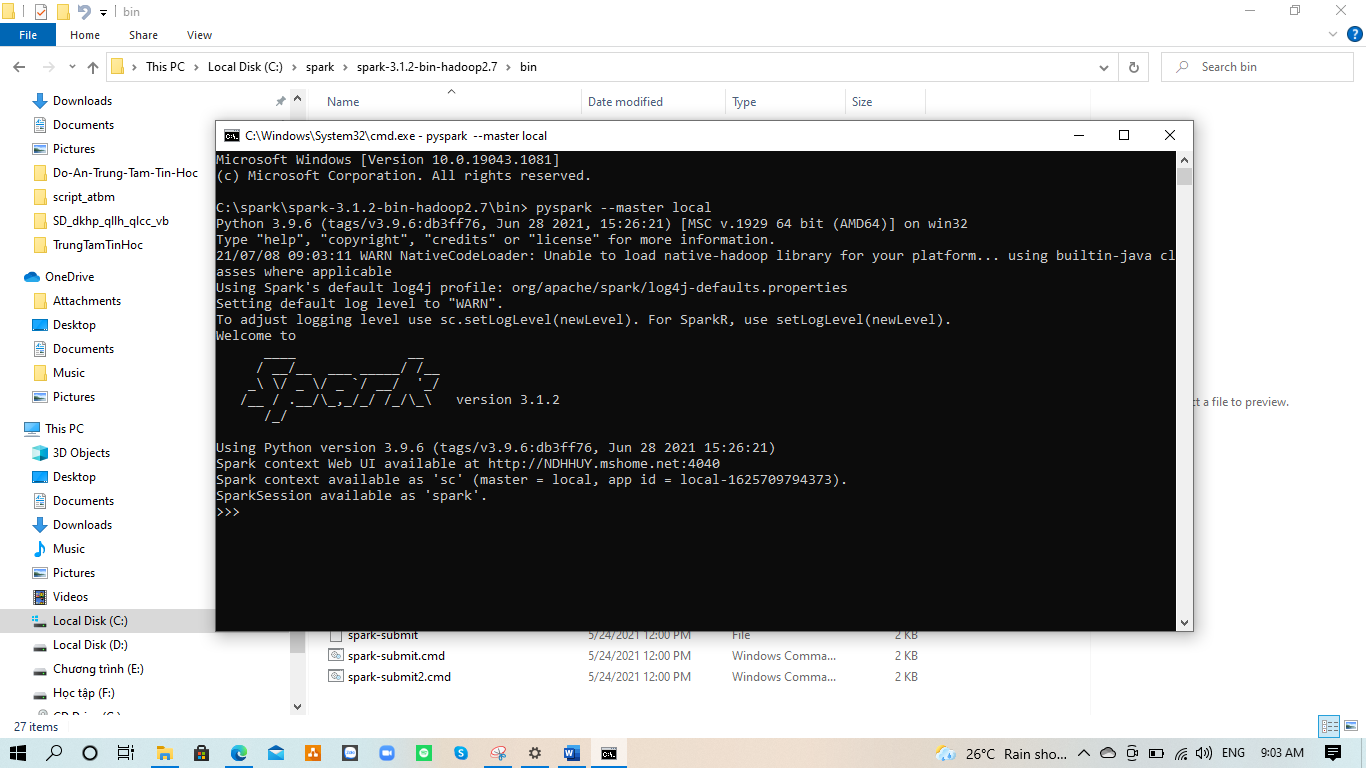
5. Thay đổi quyền truy cập cho thư mục tmp/hive

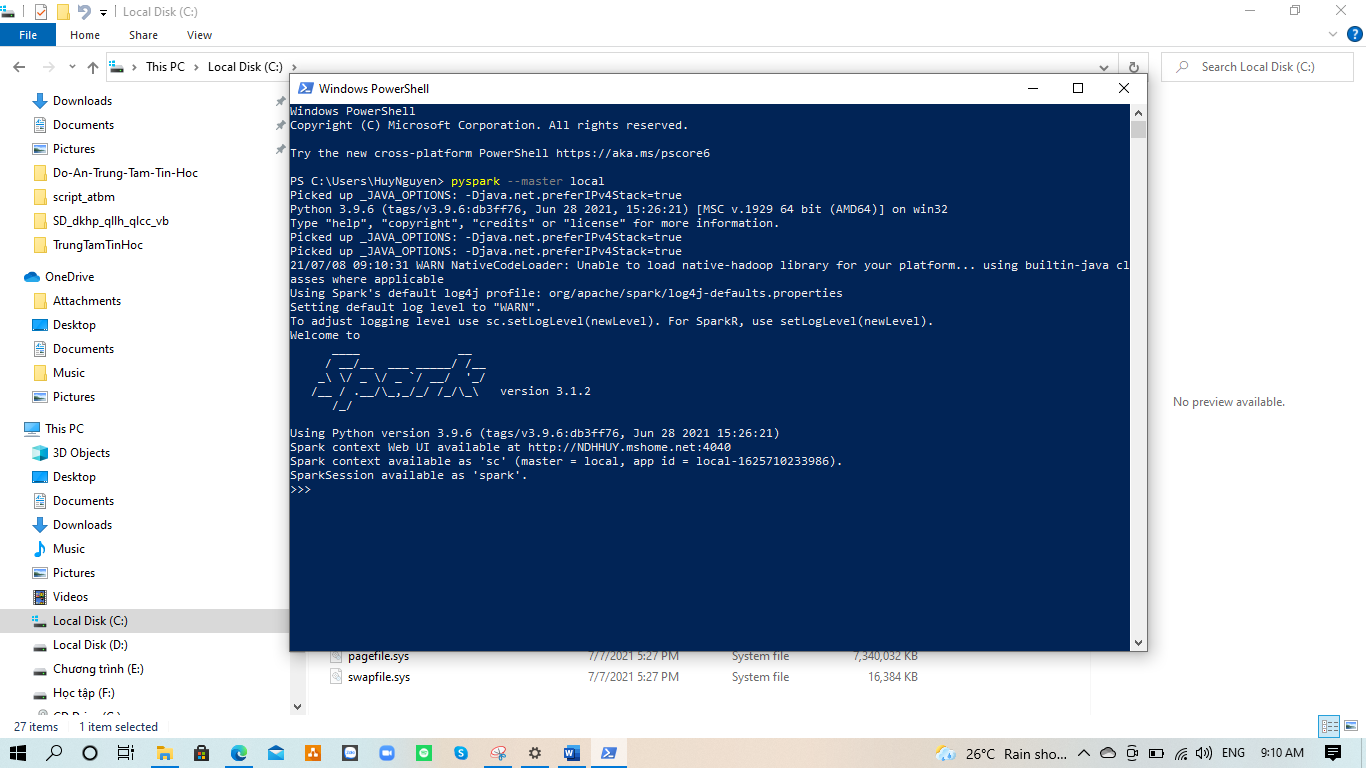
* + Sau khi hoàn thành các bước trên, sẽ có thư mục tmp/hive được tạo ra trong ổ C. Ta cần phải thay đổi quyền truy cập cho thư mục này nếu không khi chạy Spark sẽ báo lỗi
  + Để thực hiện việc này, ta mở một cửa số command rồi dùng Winutils để thay đổi permission như sau (chmod 777: cấp quyền read, write và execute

|  |  |
| --- | --- |
| 1 | winutils.exe chmod 777 C:\tmp\hive |

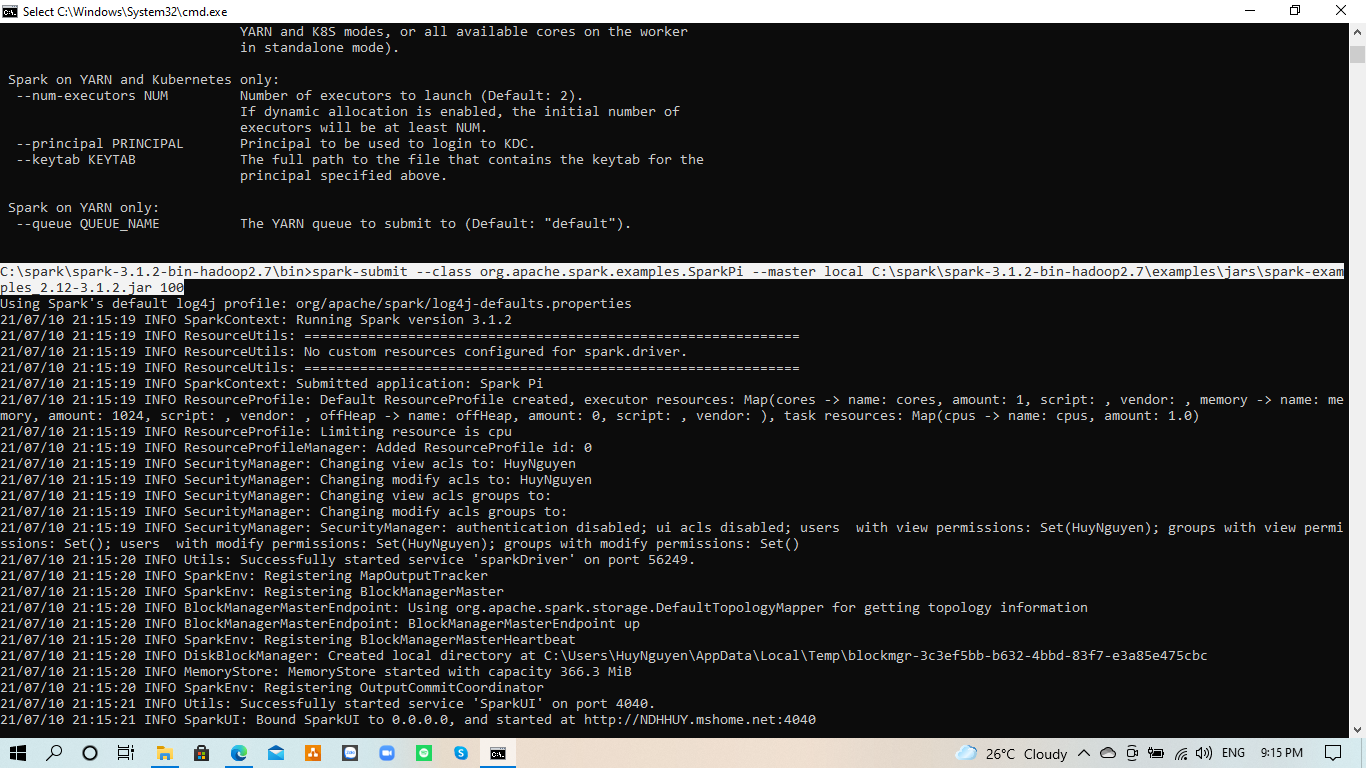
6. Kiểm tra hoàn thiện cài đặt

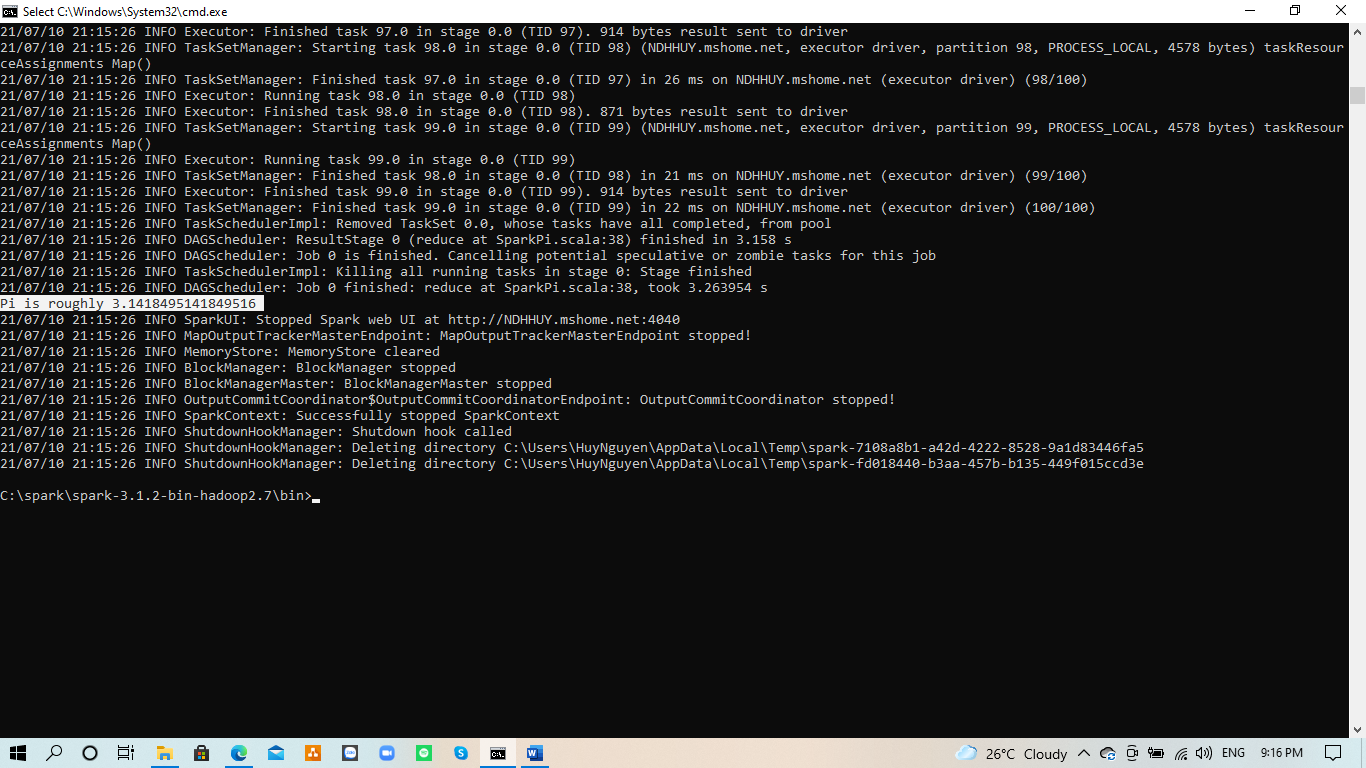
* Chạy thử một chương trình Spark nhỏ để kiểm tra xem việc cài đặt đã hoàn tất hay chưa.
* Dùng lệnh cmd để mở một của sổ command.
* Gõ lệnh spark-shell
* Gõ một chương trình nhỏ để chạy thử val





spark-submit --class org.apache.spark.examples.SparkPi --master local C:\spark\spark-3.1.2-bin-hadoop2.7\examples\jars\spark-examples\_2.12-3.1.2.jar 100





[PySpark with jupyter notebook | Install Apache Spark , PySpark, Java, Python and Anaconda on windows - YouTube](https://www.youtube.com/watch?v=DznteGdeJoA)

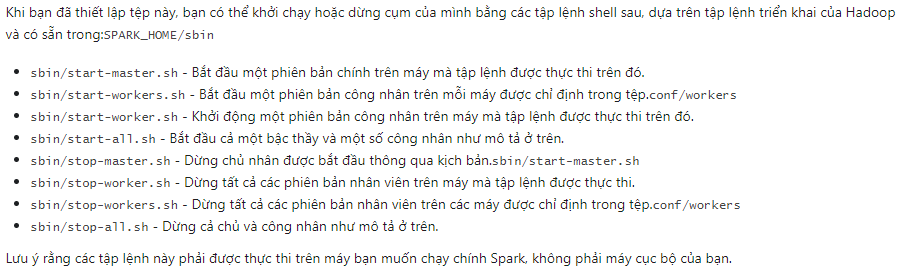
### Exploring the Spark Installation

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | bin/ | Chứa tất cả các lệnh / tập lệnh để chạy các ứng dụng Spark tương tác thông qua các chương trình shell như pyspark, spark-shell, spark-sql và sparkR hoặc ở chế độ hàng loạt bằng cách sử dụng spark-submit |
| conf/ | Chứa các mẫu cho các tệp cấu hình Spark mà bạn có thể sử dụng để đặt các giá trị cấu hình Spark (spark-defaults.conf.template), cũng như tập lệnh shell được sử dụng để đặt các biến môi trường cần thiết cho các quy trình Spark (spark-env.sh.template). Ngoài ra còn có cấu hìnhcác mẫu để kiểm soát việc ghi nhật ký (log4j.properties.template), một bộ sưu tập số liệu (metrics.properties.template) và một mẫu cho tệp nô lệ (slave.template), kiểm soát tệp nàocác nút nô lệ có thể tham gia cụm Spark đang chạy ở chế độ Độc lập. |
| data/ | Chứa tập dữ liệu mẫu được sử dụng để kiểm tra mllib, graphx,và phát trực tuyến các thư viện trong dự án Spark (tất cả đềusẽ được thảo luận sau trong cuốn sách này). |
| example/ | Chứa mã nguồn và các tập hợp đã biên dịch (tệp jar) cho tất cả các ví dụ đi kèm với bản phát hành Spark, bao gồm cả ứng dụng Công cụ ước tính Pi được sử dụng trong các ví dụ trước. Các chương trình mẫu được bao gồm trong Java, Python, R và Scala. Bạn cũng có thể tìm thấy mã mới nhất cho các ví dụ được bao gồm tại <https://github.com/apache/spark/tree/master/examples>. |
| jars/ | Chứa các cụm chính cho Spark cũng như các cụm cho các dịch vụ hỗ trợ được Spark sử dụng, chẳng hạn như snappy, py4j, parquet, và hơn thế nữa. Thư mục này được bao gồm trong CLASSPATH cho Spark theo mặc định |
| kubernetes/ |  |
| licenses/ | Bao gồm các tệp giấy phép bao gồm các dự án bao gồm khác, chẳng hạn như Scala và JQuery. Các tệp này dành cho mục đích tuân thủ pháp luật duy nhất và không bắt buộc phải chạy Spark |
| python/ | Chứa tất cả các thư viện Python cần thiết để chạy PySpark. Bạnnói chung không cần phải truy cập trực tiếp vào các tệp này. |
| R/ | Chứa gói SparkR và các thư viện liên quan và tài liệu. 🡪 Chapter 8 |
| sbin/ | Chứa các tập lệnh quản trị để bắt đầu và dừng master và slave dịch vụ cho các cụm Spark chạy ở chế độ Độc lập, cục bộhoặc từ xa, cũng như bắt đầu các quy trình liên quan đến YARN và Mesos. Bạn sẽ sử dụng một số đoạn mã này trong phần tiếp theokhi bạn triển khai một cụm nhiều nút ở chế độ Độc lập. |
| yarn/ | Chứa các thư viện hỗ trợ cho các ứng dụng Spark đang chạy trênYARN. Điều này bao gồm dịch vụ xáo trộn, một dịch vụ hỗ trợSpark sử dụng để di chuyển dữ liệu giữa các quy trình trong một cụm YARN. |

### Deploying a Multi-Node Spark Standalone Cluster

Hiện tại, Apache Spark supports Standalone, Apache Mesos, YARN và Kubernetes với tư cách là người quản lý tài nguyên. Standalone là trình quản lý tài nguyên của spark, dễ thiết lập có thể được sử dụng để bắt đầu mọi thứ nhanh chóng.

Có rất nhiều bài viết và đủ thông tin về cách bắt đầu một cụm độc lập trên môi trường Linux. Nhưng, không có nhiều thông tin về việc bắt đầu một cụm độc lập trên Windows.



Bây giờ bạn đã cài đặt và thử nghiệm cài đặt Spark ở chế độ Cục bộ, nó thời gian để giải phóng sức mạnh thực sự của Spark bằng cách tạo ra một Spark được phân phối đầy đủ cụm. Đối với bài tập này, bạn sẽ sử dụng bốn máy chủ Linux để tạo một ba-cụm nút bằng cách sử dụng bộ lập lịch biểu độc lập. Làm theo các bước sau:

1. Lập kế hoạch cấu trúc liên kết cụm và cài đặt Spark trên nhiều hệ thống. Bởi vì đây là một hệ thống phân tán, bạn cần cài đặt Spark, như được hiển thị trong các bài tập trước, trên ba máy chủ bổ sung. Ngoài ra, bạn cần phải chỉ định một máy chủ là Spark Master và máy chủ khác là Người lao động. Đối với bài tập này, máy chủ đầu tiên được đặt tên là sparkmaster và các máy chủ bổ sung là các tên sparkworker1, sparkworker2 và sparkworker3.

2. Cấu hình mạng. Tất cả các nút trong cụm Spark cần giao tiếp tất cả các máy chủ khác trong cụm. Cách dễ nhất để thực hiện điều này là bằng cách sử dụng tệp máy chủ (mục nhập cho tất cả máy chủ trong / etc / hosts trên mỗi hệ thống). Đảm bảo rằng mỗi nút có thể giải quyết nút kia. Lệnh ping có thể được sử dụng cho điều này; ví dụ: đây là cách bạn sử dụng nó từ máy chủ sparkmaster:

$ ping sparkworker1

3. Tạo và chỉnh sửa tệp spark-defaults.conf trên mỗi máy chủ. Để tạo và định cấu hình tệp spark-defaults.conf trên mỗi nút, chạy các lệnh sau trên máy chủ sparkmaster và sparkworker:

$ cd $ SPARK\_HOME / conf

$ sudo cp spark-defaults.conf.template spark-defaults.conf

$ sudo sed -i "\ $ aspark.master \ tspark: // sparkmaster: 7077" spark-

defaults.conf

4. Tạo và chỉnh sửa tệp spark-env.sh trên mỗi máy chủ. Để tạo và định cấu hình tệp spark-env.sh trên mỗi nút, hoàn thành các bước sau nhiệm vụ trên máy chủ sparkmaster và sparkworker:

$ cd $ SPARK\_HOME / conf

$ sudo cp spark-env.sh.template spark-env.sh

$ sudo sed -i "\ $ aSPARK\_MASTER\_IP = sparkmaster" spark-env.sh

5. Khởi động Spark Master. Trên máy chủ sparkmaster, hãy chạy như sau chỉ huy:

$ sudo $ SPARK\_HOME / sbin / start-master.sh

Kiểm tra quy trình Spark Master bằng cách xem giao diện người dùng web Spark Master tại

http: // sparkmaster: 8080 / .

6. Khởi động Công nhân Tia lửa. Trên mỗi nút sparkworker, hãy chạy như sau chỉ huy:

$ sudo $ SPARK\_HOME / sbin / start-slave.sh spark: // sparkmaster: 7077

Kiểm tra giao diện người dùng Spark Worker trên http: // sparkslaveN: 8081 / .

7. Kiểm tra cụm nhiều nút. Chạy ví dụ về Công cụ ước tính Pi được tích hợp sẵn từ đầu cuối của bất kỳ nút nào trong cụm:

$ spark-submit --class org.apache.spark.examples.SparkPi \

--master spark: // sparkmaster: 7077 \

--driver-memory 512m \

--executor-memory 512m \

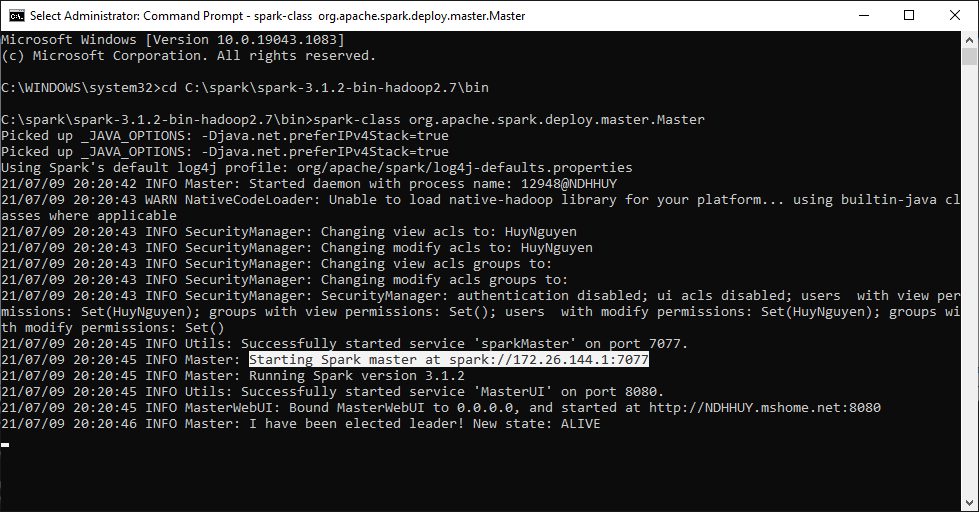
--executor-core 1 \

$ SPARK\_HOME / ví dụ / lọ / ví dụ tia lửa \* .jar 10000

Bạn sẽ thấy đầu ra tương tự như kết quả từ các bài tập trước. Bạn cũng có thể bật SSH không mật khẩu (Secure Shell) cho Spark Master để workers Yarn. Điều này là bắt buộc để kích hoạt đăng nhập từ xa cho daemon nô lệ các hành động khởi động và tắt máy.

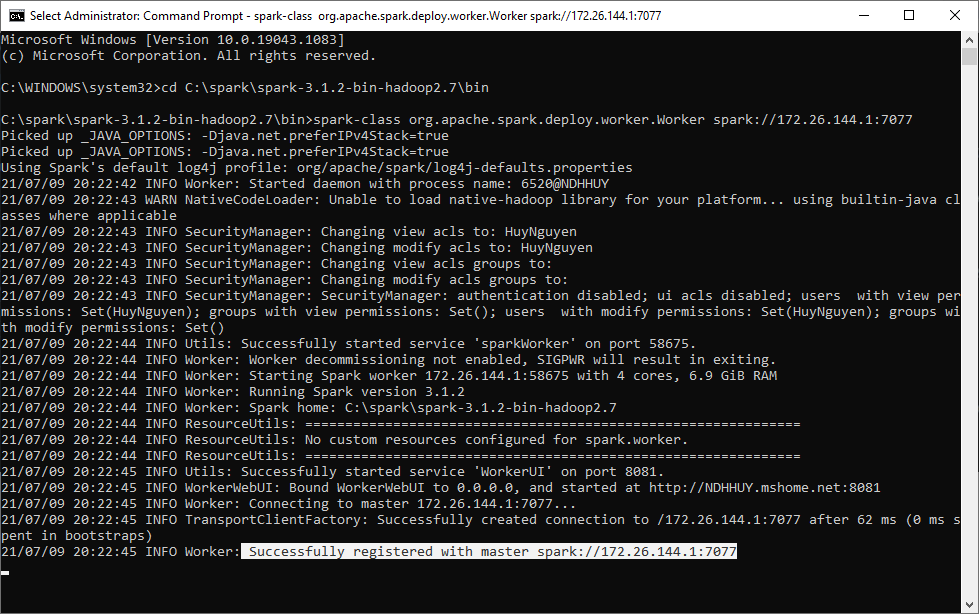
**Set up Master Node**

spark-class org.apache.spark.deploy.master.Master

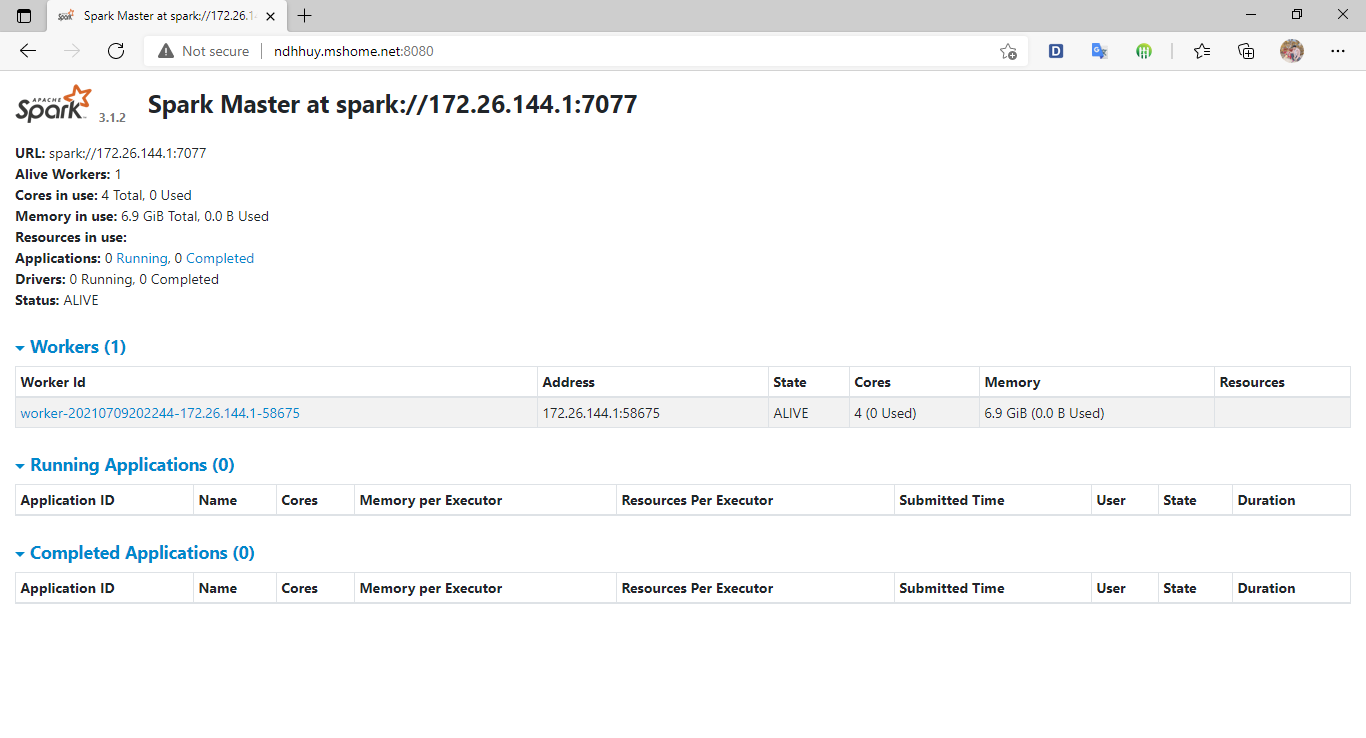


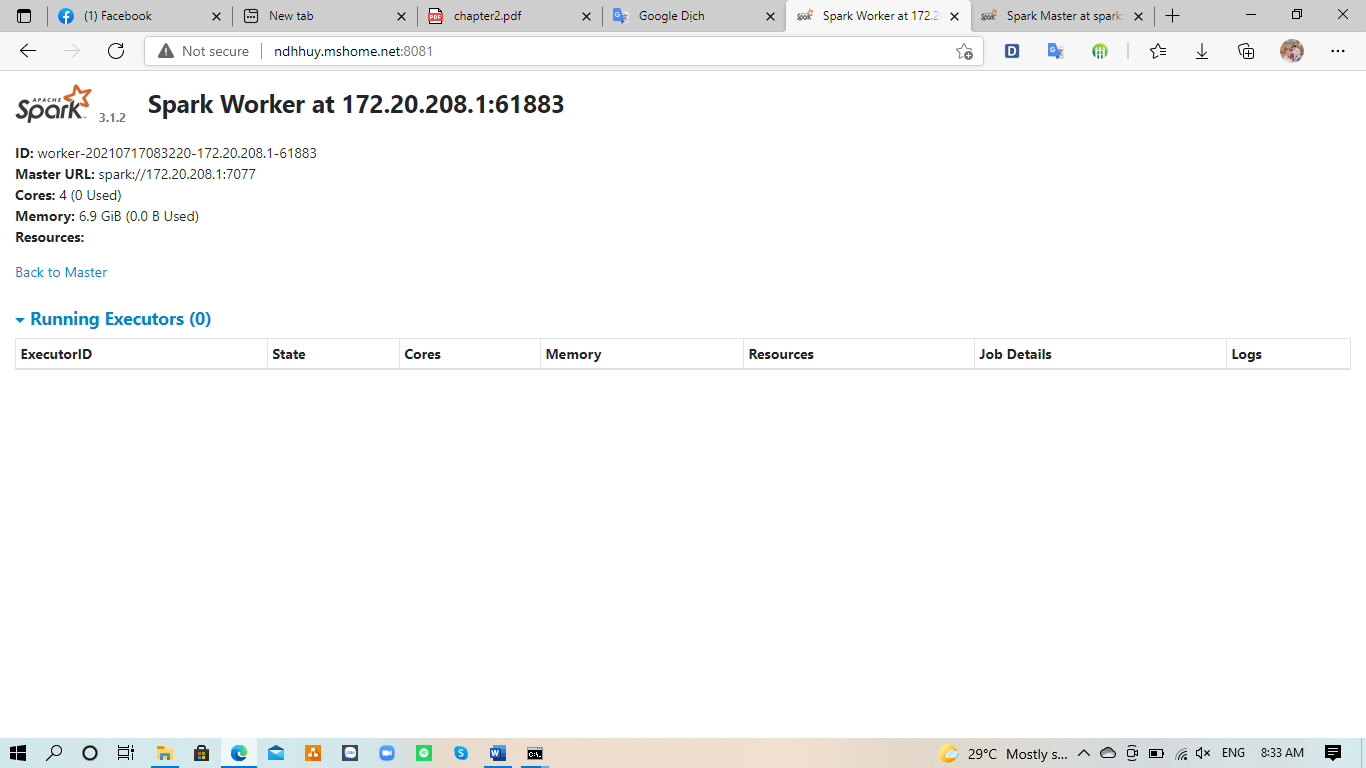
**Set up Worker Node**

spark-class org.apache.spark.deploy.worker.Worker spark://172.26.144.1:7077



**Spark UI**





[Apache Spark standalone cluster on Windows | by Amar Gajbhiye | Medium](https://aamargajbhiye.medium.com/apache-spark-setup-a-multi-node-standalone-cluster-on-windows-63d413296971)

### Deploying Spark in the Cloud

* Software-as-a-Service (SaaS)
* Infrastructure-as-a-Service (IaaS) (Cơ sở hạ tầng như dịch vụ)
* Platform-as-a-Service (PaaS) (Nền tảng như dịch vụ)

#### Amazon Web Services (AWS)

* Amazon Web Services (AWS) là nền tảng đám mây toàn diện và được sử dụng rộng rãi nhất, cung cấp trên 200 dịch vụ đầy đủ tính năng từ các trung tâm dữ liệu trên toàn thế giới. Hàng triệu khách hàng, bao gồm các công ty khởi nghiệp tăng trưởng nhanh nhất, các tập đoàn lớn nhất cũng như các cơ quan hàng đầu của chính phủ, đều tin tưởng vào AWS để giảm chi phí, trở nên linh hoạt hơn và đổi mới nhanh hơn.
* Đầy đủ chức năng: từ các công nghệ cơ sở hạ tầng như máy tính, ổ lưu trữ và cơ sở dữ liệu, đến các công nghệ mới nổi, như machine learning và trí tuệ nhân tạo, kho dữ liệu và phân tích, và Internet of Things
* Bảo mật nhất: được thiết kế để trở thành môi trường điện toán đám mây bảo mật và linh hoạt nhất hiện nay

#### Google Cloud Platform (GCP)

* Google Cloud hay còn gọi là Google Cloud Platform (GCP) chính là một nền tảng của kỹ thuật điện toán đám mây cho phép các cá nhân, tổ chức, các doanh nghiệp, các cơ quan có thể xây dựng, phát triển, và hoạt động các ứng dụng của mình trên hệ thống phần mềm do google tạo ra. Các ứng dụng rất phổ biến hiện nay được mọi người sử dụng rất nhiều như: Trình duyệt Chrome, ứng dụng bản đồ Google Map, Google Apps, kênh Youtube…
* Google Cloud cung cấp tất cả các giải pháp quản lý cho doanh nghiệp, giúp doanh nghiệp có thể phát triển hệ thống công nghệ của mình một cách chính xác, hiện đại. Bên cạnh đó, GC còn giúp người dùng và doanh nghiệp giải quyết các vấn đề như: Developer (phát triển), Management (Quản lý), Computer Engine, Mobile, Storage, Big Data…
* Một điểm khác biệt nữa mà GC mang lại so với các dịch vụ đám mây khác đó chính là hệ thống DataCenter luôn ổn định và có độ bảo mật dữ liệu cực cao, giúp bảo vệ dữ liệu người dùng và khách hàng trước sự dòm ngó và xâm nhập trái phép của các hacker công nghệ.

#### Databricks

* Azure Databricks là giải pháp phần mềm Big Data Analytics Software với chức năng và chi phí phù hợp cho các doanh nghiệp từ nhỏ và vừa tới các doanh nghiệp lớn.
* Phần mềm Azure Databricks được đánh giá cao bởi cả người dùng lẫn chuyên gia trong lĩnh vực Database Software.

### Summary