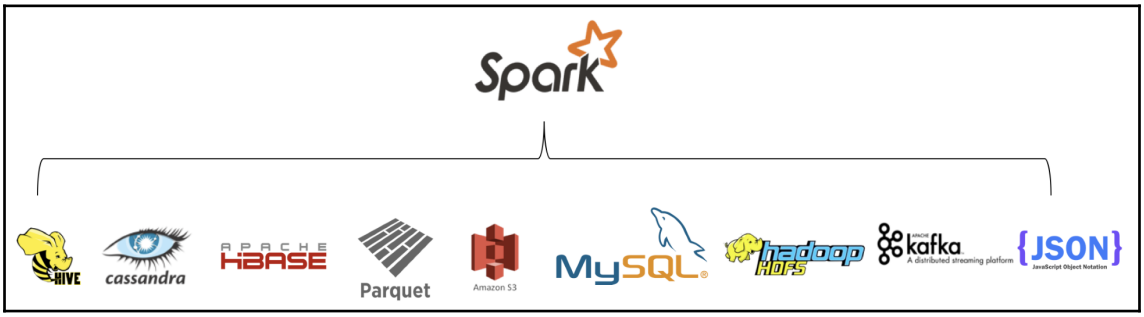
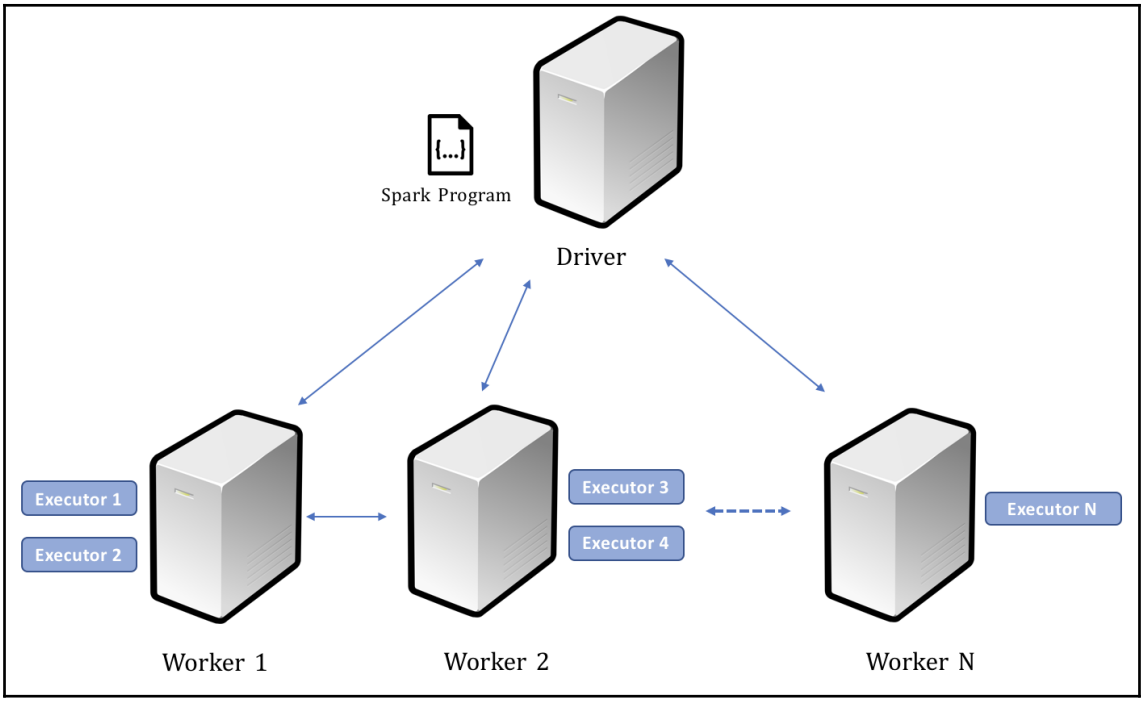
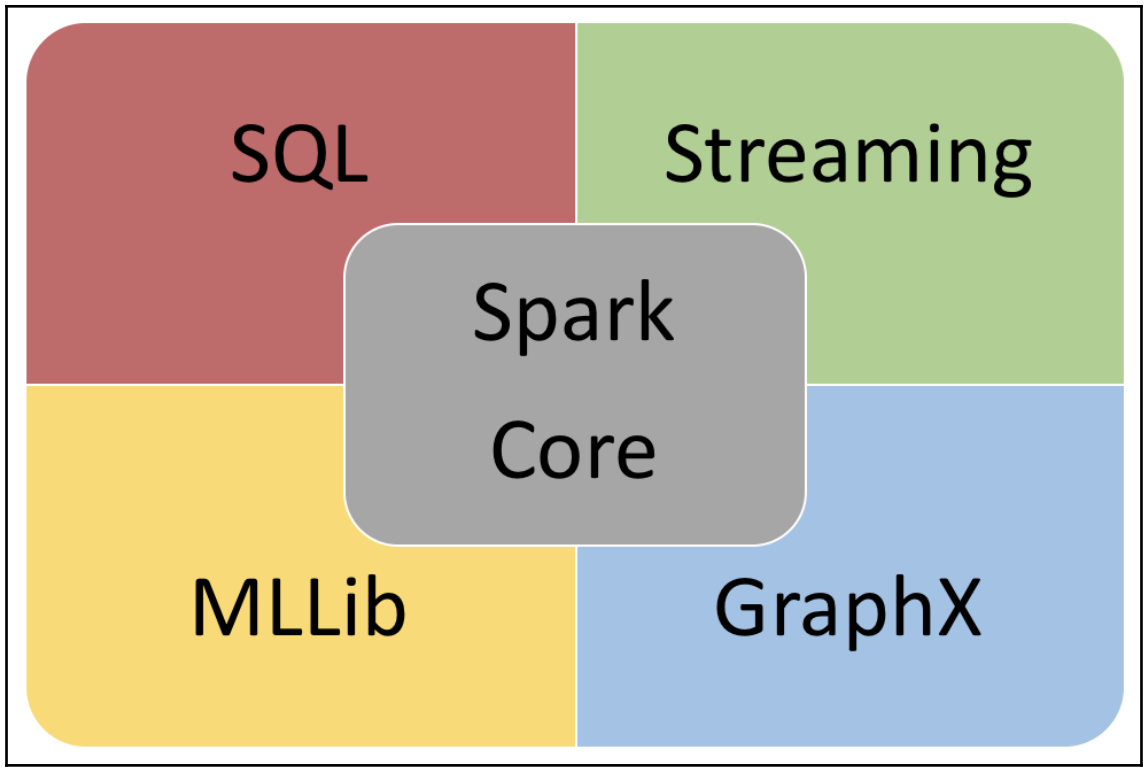
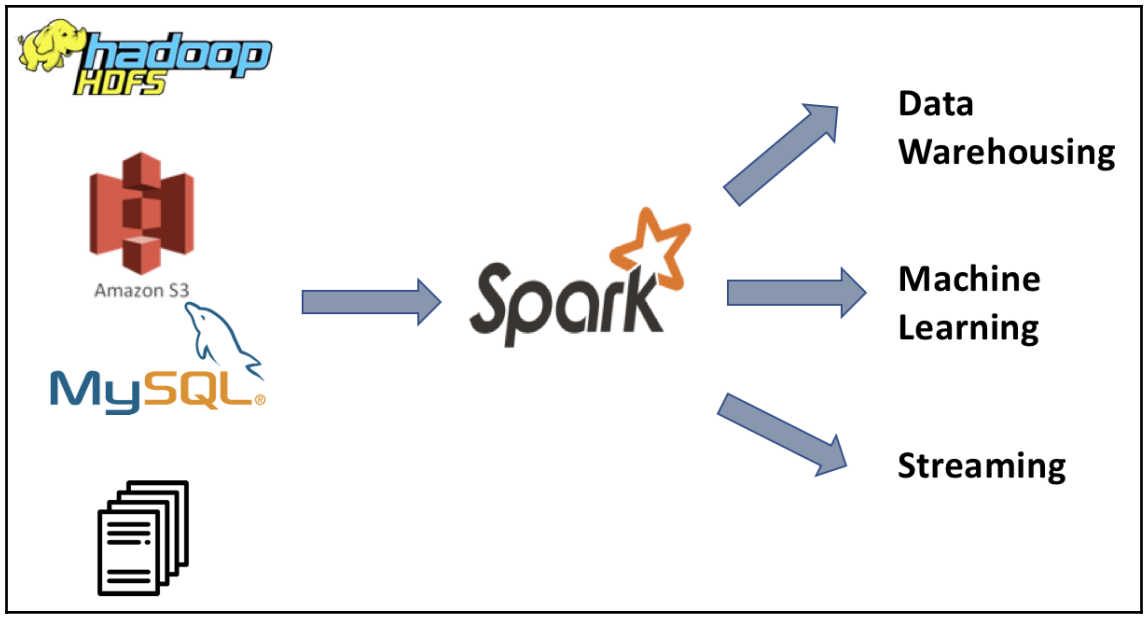
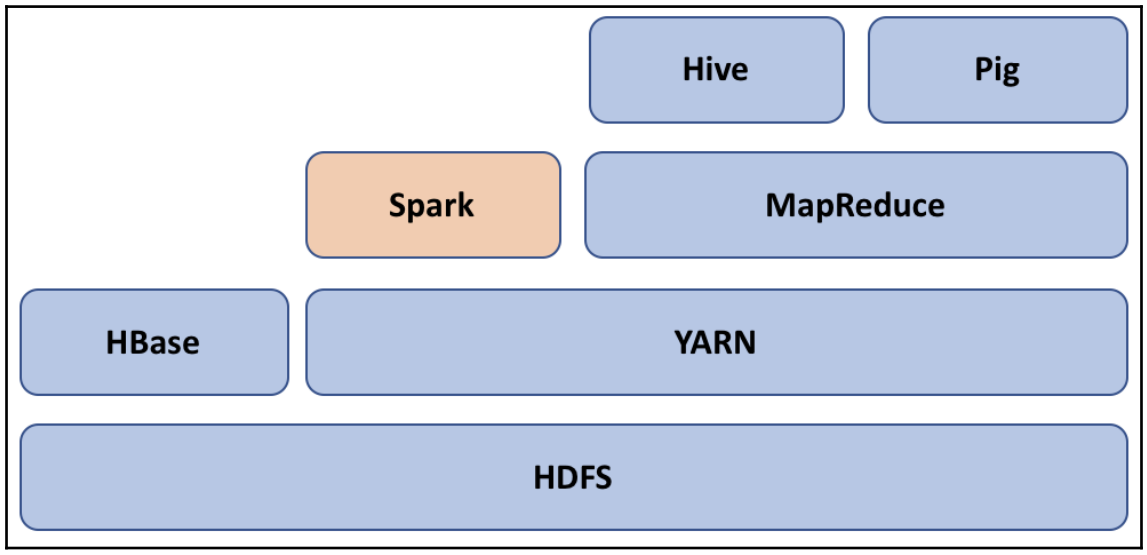
Chapter 1: Introduction to Apache Spark



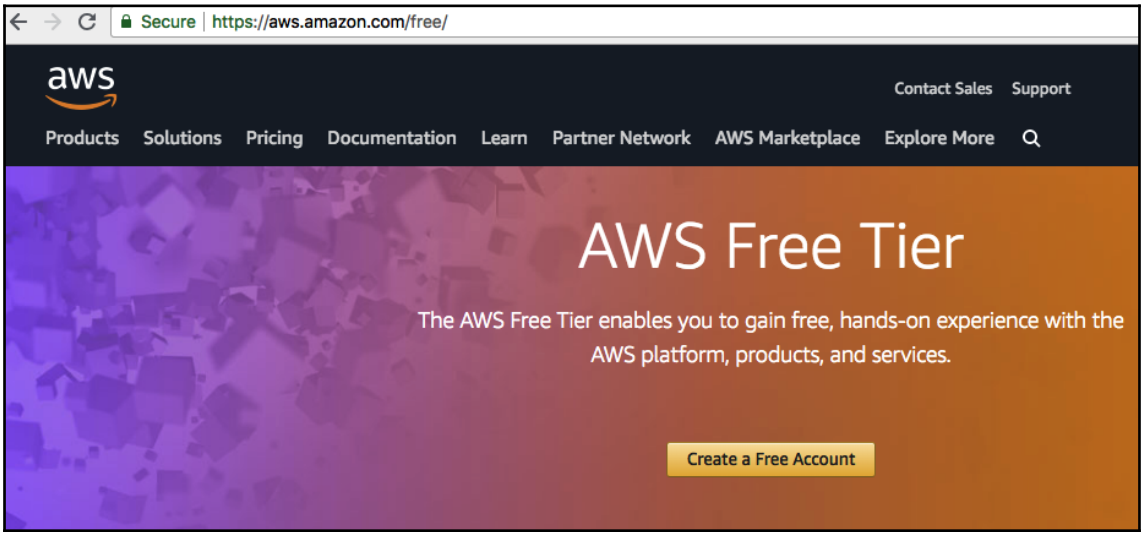


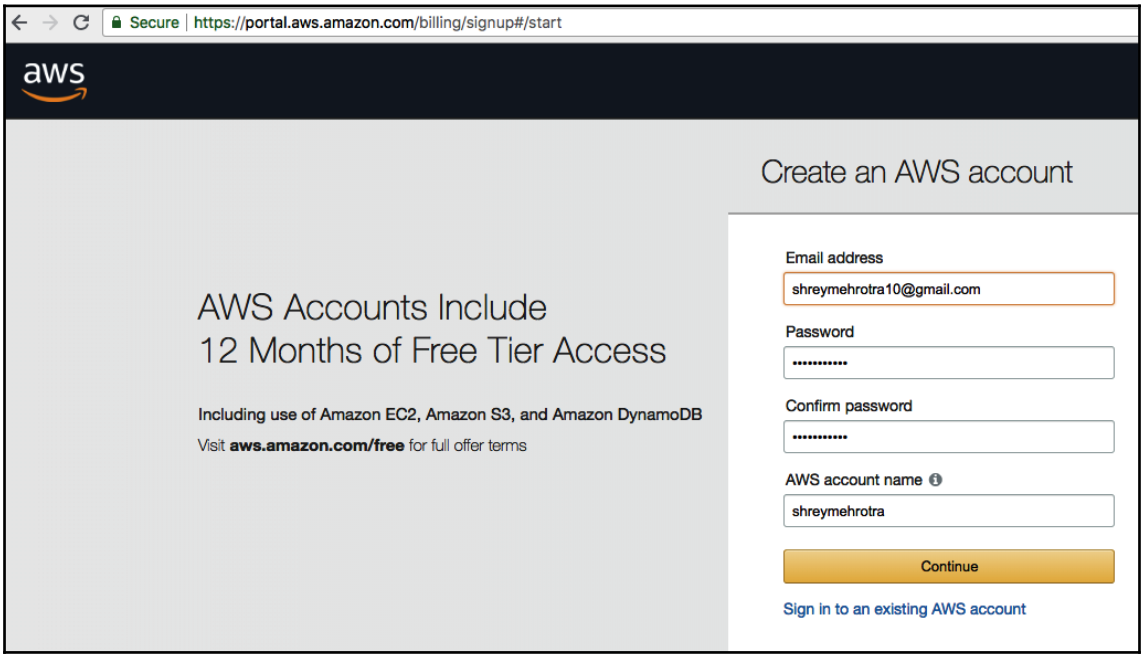

SQL
Streaming
Spark
Core
MLLib
GraphX

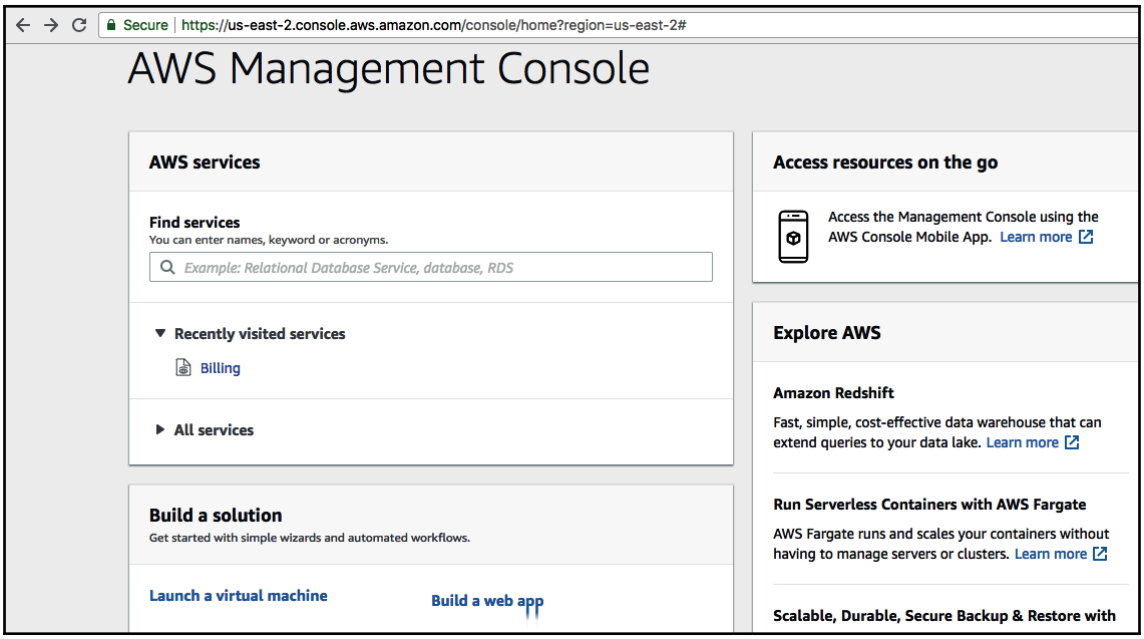

hadoop
HDFS
Amazon S3
MySQL
Spark
Data
Warehousing
Machine
Learning
Streaming

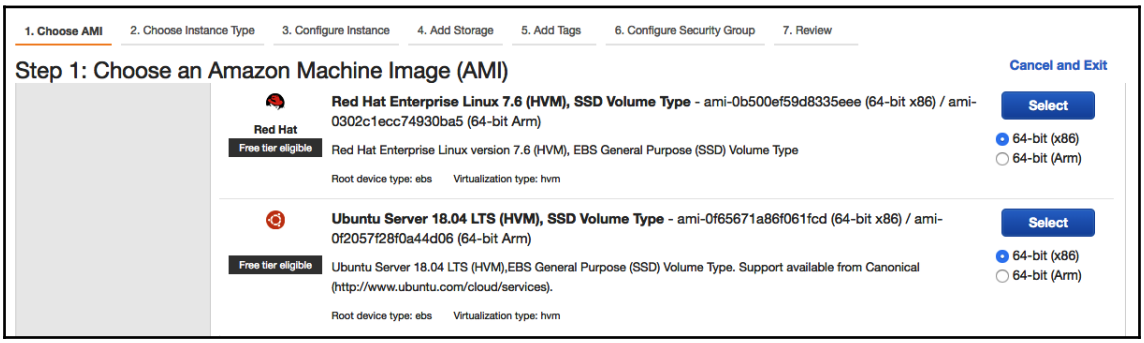

Hive
Pig
Spark
MapReduce
HBase
YARN
HDFS

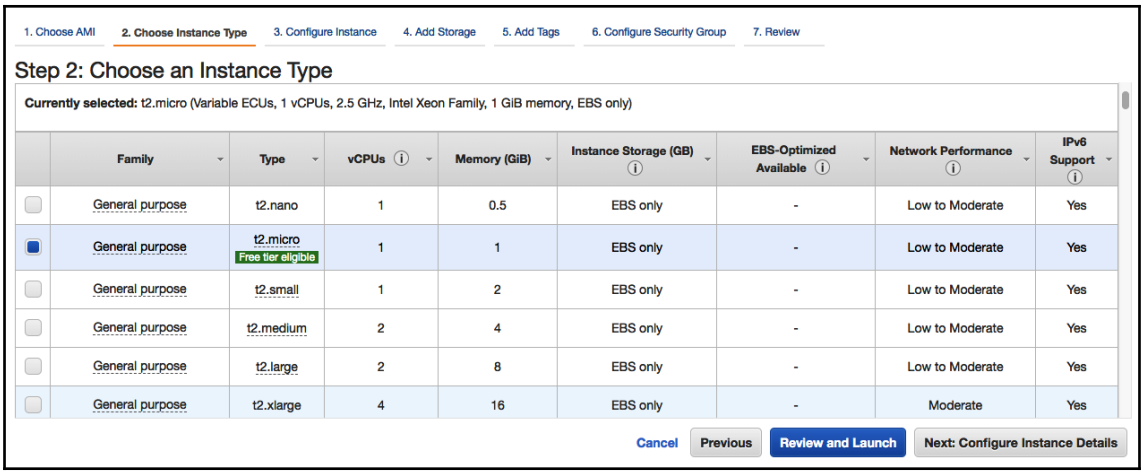
Chapter 2: Apache Spark Installation

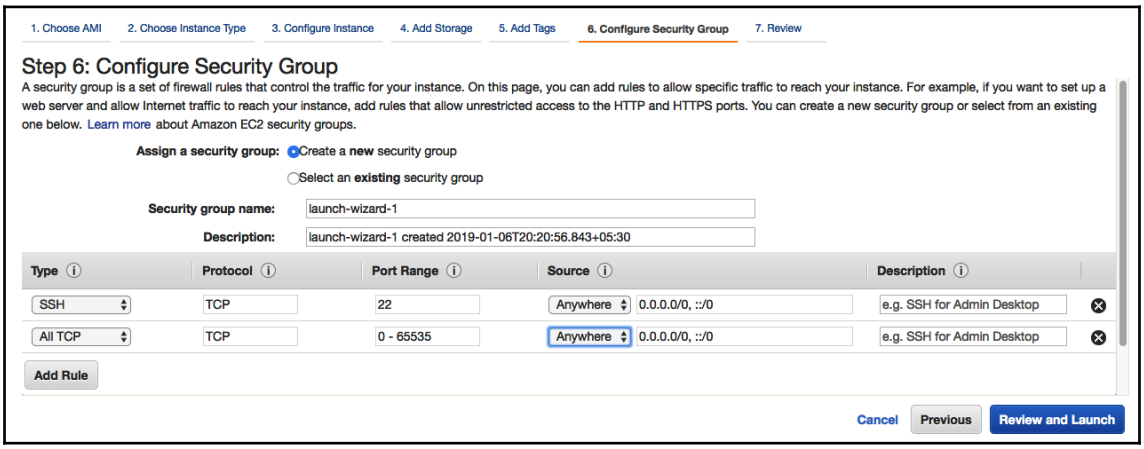


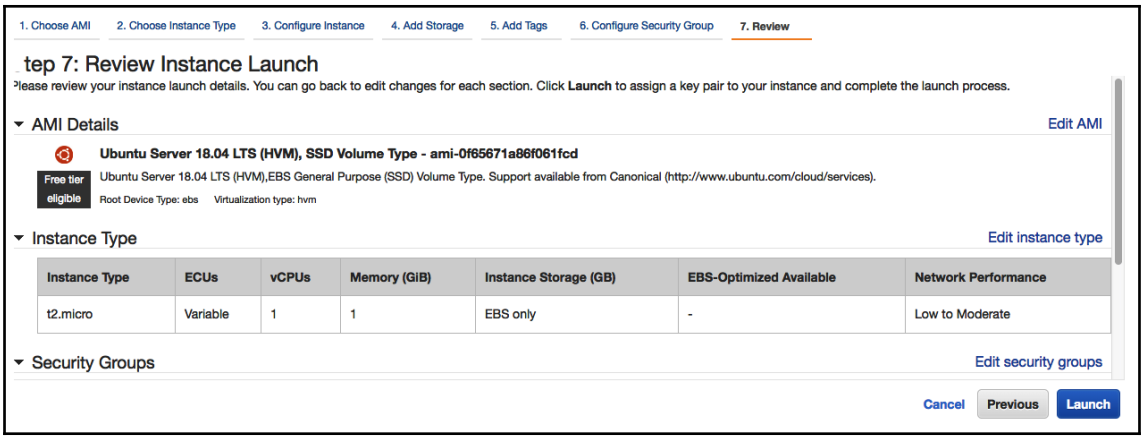


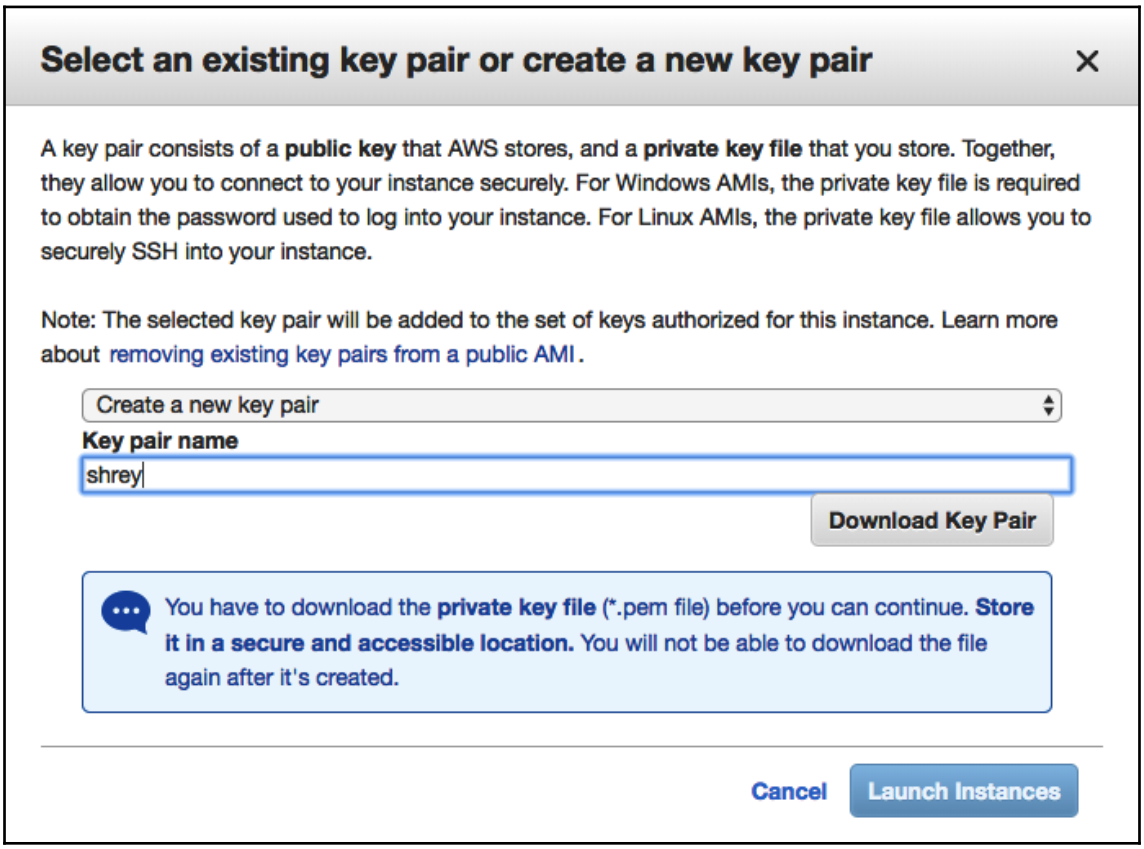

Secure | https://us-east-2.console.aws.amazon.com/console/home?region=us-east-2#
AWS Management Console
AWS services
Find services
You can enter names, keyword or acronyms.
Example: Relational Database Service, database, RDS
Recently visited services
Billing
All services
Build a solution
Get started with simple wizards and automated workflows.
Launch a virtual machine
Build a web app
Access resources on the go
Access the Management Console using the AWS Console Mobile App. Learn more
Explore AWS
Amazon Redshift
Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more
Run Serverless Containers with AWS Fargate
AWS Fargate runs and scales your containers without having to manage servers or clusters. Learn more
Scalable, Durable, Secure Backup & Restore with

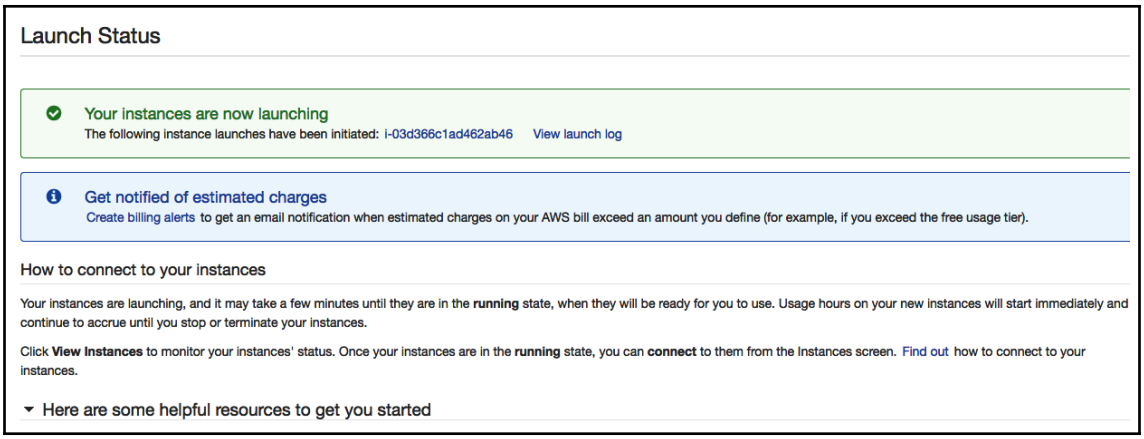

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
Red Hat
Free tier eligible
Red Hat Enterprise Linux 7.6 (HVM), SSD Volume Type - ami-0b500ef59d8335eee (64-bit x86) / ami-0302c1ecc74930ba5 (64-bit Arm)
Red Hat Enterprise Linux version 7.6 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm
Select
64-bit (x86)
64-bit (Arm)
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0f65671a86f061fcd (64-bit x86) / ami-0f2057f28f0a44d06 (64-bit Arm)
Free tier eligible
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm
Select
64-bit (x86)
64-bit (Arm)

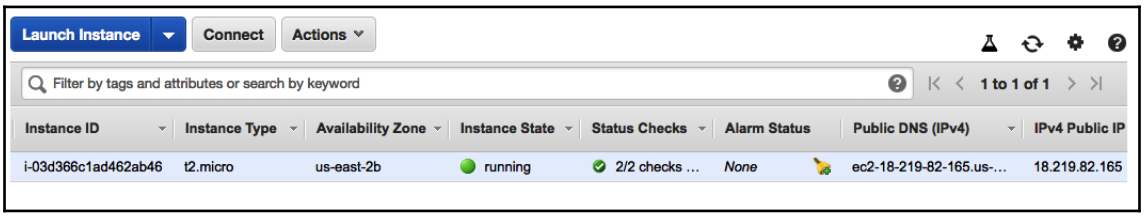

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 2: Choose an Instance Type
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)
Family
Type
vCPUs
Memory (GiB)
Instance Storage (GB)
EBS-Optimized Available
Network Performance
IPv6 Support
General purpose t2.nano 1 0.5 EBS only - Low to Moderate Yes
General purpose t2.micro Free tier eligible 1 1 EBS only - Low to Moderate Yes
General purpose t2.small 1 2 EBS only - Low to Moderate Yes
General purpose t2.medium 2 4 EBS only - Low to Moderate Yes
General purpose t2.large 2 8 EBS only - Low to Moderate Yes
General purpose t2.xlarge 4 16 EBS only - Moderate Yes
Cancel
Previous
Review and Launch
Next: Configure Instance Details

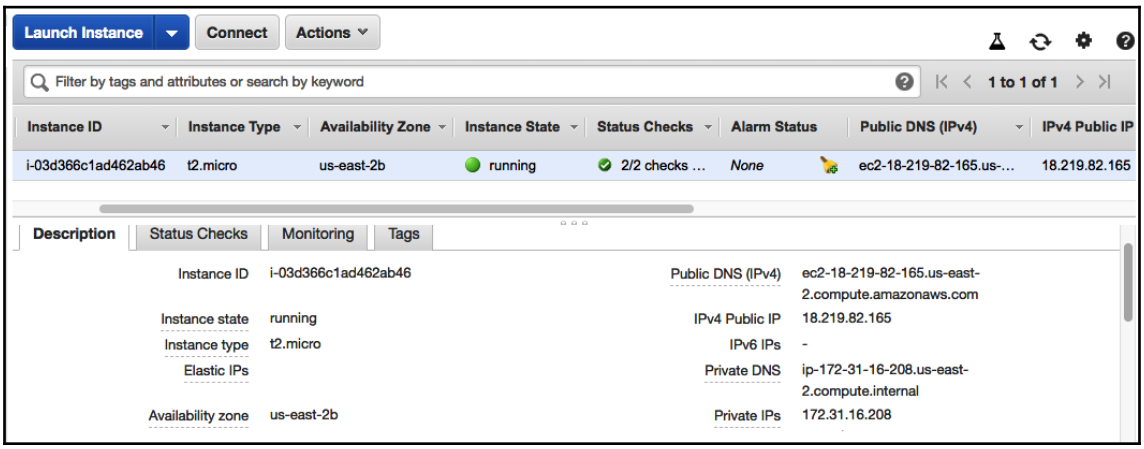

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.
Assign a security group: Create a new security group
Select an existing security group
Security group name: launch-wizard-1
Description: launch-wizard-1 created 2019-01-06T20:20:56.843+05:30
Type
Protocol
Port Range
Source
Description
SSH TCP 22 Anywhere 0.0.0.0/0, ::/0 e.g. SSH for Admin Desktop
All TCP TCP 0 - 65535 Anywhere 0.0.0.0/0, ::/0 e.g. SSH for Admin Desktop
Add Rule
Cancel
Previous
Review and Launch


1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
tep 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.
AMI Details
Edit AMI
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0f65671a86f061fcd
Free tier eligible
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root Device Type: ebs
Virtualization type: hvm
Instance Type
Edit instance type
Instance Type
ECUs
vCPUs
Memory (GiB)
Instance Storage (GB)
EBS-Optimized Available
Network Performance
t2.micro
Variable
1
1
EBS only
-
Low to Moderate
Security Groups
Edit security groups
Cancel
Previous
Launch


Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Create a new key pair
Key pair name
shrey
Download Key Pair
You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.
Cancel
Launch Instances


Launch Status
Your instances are now launching
The following instance launches have been initiated: i-03d366c1ad462ab46 View launch log
Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).
How to connect to your instances
Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.
Click View Instances to monitor your instances' status. Once your instances are in the running state, you can connect to them from the Instances screen. Find out how to connect to your instances.
Here are some helpful resources to get you started


Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
IPv4 Public IP
i-03d366c1ad462ab46
t2.micro
us-east-2b
running
2/2 checks ...
None
ec2-18-219-82-165.us-...
18.219.82.165


Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
IPv4 Public IP
i-03d366c1ad462ab46
t2.micro
us-east-2b
running
2/2 checks ...
None
ec2-18-219-82-165.us-...
18.219.82.165
Description
Status Checks
Monitoring
Tags
Instance ID
i-03d366c1ad462ab46
Instance state
running
Instance type
t2.micro
Elastic IPs
Availability zone
us-east-2b
Public DNS (IPv4)
ec2-18-219-82-165.us-east-2.compute.amazonaws.com
IPv4 Public IP
18.219.82.165
IPv6 IPs
-
Private DNS
ip-172-31-16-208.us-east-2.compute.internal
Private IPs
172.31.16.208

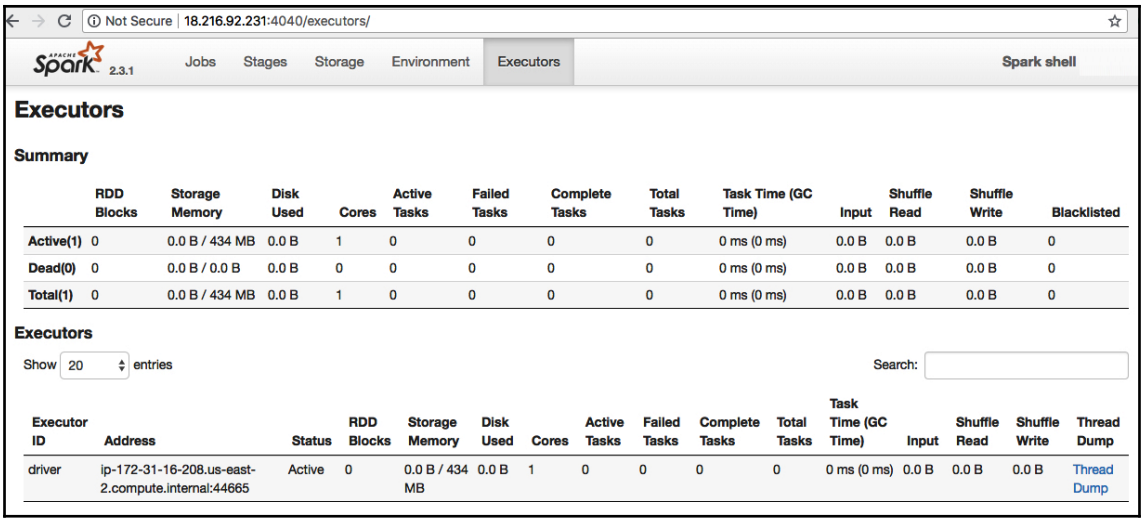
```
shrey:Downloads shrey$ ssh -i "shrey.pem" ubuntu@ec2-18-219-82-165.us-east-2.compute.amazonaws.com
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Jan  6 17:41:26 UTC 2019

  System load:  0.0               Processes:           87
  Usage of /:   13.4% of 7.69GB   Users logged in:     1
  Memory usage: 14%               IP address for eth0: 172.31.16.208
  Swap usage:   0%

 * MicroK8s is Kubernetes in a snap. Made by devs for devs.
   One quick install on a workstation, VM, or appliance.

   - https://bit.ly/microk8s

 * Full K8s GPU support is now available!

   - https://blog.ubuntu.com/2018/12/10/using-gpgpus-with-kubernetes


  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.


Last login: Sun Jan  6 15:46:16 2019 from 223.225.84.26
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-16-208:~$
```

```
ubuntu@ip-172-31-16-208:/opt$ spark-shell
2019-01-18 21:31:47 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://ip-172-31-16-208.us-east-2.compute.internal:4040
Spark context available as 'sc' (master = local[*], app id = local-1547847120337).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.3.1
      /_/

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_201)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

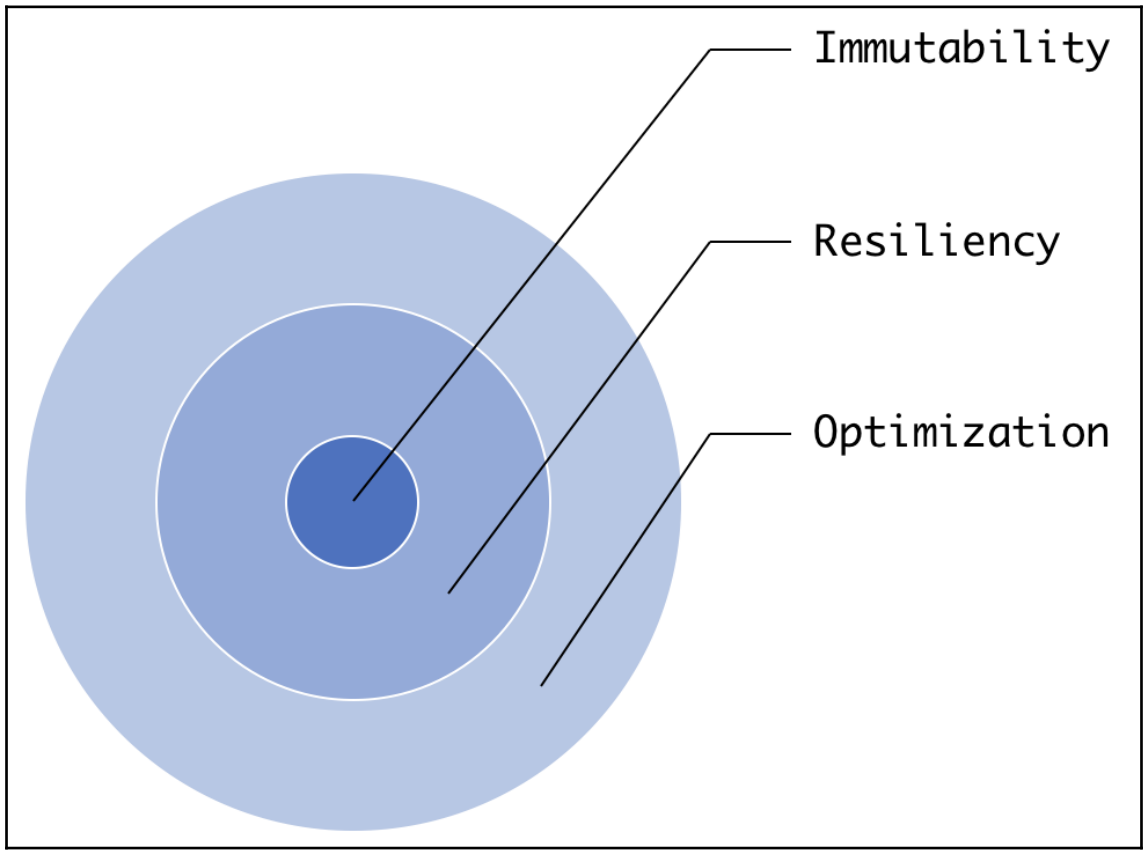
```
ubuntu@ip-172-31-16-208:/opt$ pyspark
/opt/spark-2.3.1-bin-hadoop2.7/bin/pyspark: line 45: python: command not found
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
2019-01-18 22:18:58 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.3.1
      /_/

Using Python version 3.6.7 (default, Oct 22 2018 11:32:17)
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>> spark = SparkSession.builder.appName("Python Spark dataframe example").config("spark.some.config.option", "some-value").getOrCreate()
>>> sales_df = spark.read.option("sep", "\t").option("header", "true").csv("/opt/sample_10000.txt")
2019-01-18 22:19:50 WARN  ObjectStore:6666 - Version information not found in metastore. hive.metastore.schema.verification is not enabled so recording
2019-01-18 22:19:50 WARN  ObjectStore:568 - Failed to get database default, returning NoSuchObjectException
2019-01-18 22:19:51 WARN  ObjectStore:568 - Failed to get database global_temp, returning NoSuchObjectException
>>> sales_df.show()
+---+---------+---------+--------------------+--------------------+----------+-----+---------------+----------+---------+----+
| id|firstname| lastname|             address|                city|     state|  zip|             ip|product_id|      dop|_c10|
+---+---------+---------+--------------------+--------------------+----------+-----+---------------+----------+---------+----+
|  0|     Zena|     Ross|41228 West India Ln.|              Powell|Tennessee|21550|192.168.56.127|     PI_09|13/6/2014|null|
|  1|   Elaine|   Bishop|15903 North North...|Hawaiian Gardens|    Alaska|06429|192.168.56.105|     PI_03| 8/6/2014|null|
|  2|     Sage|  Carroll| 6880  Greenland Ct.|       Guayanilla|    Nevada|08899| 192.168.56.40|     PI_03|13/6/2014|null|
|  3|     Cade|Singleton|64021 South Bulga...|              Derby|  Missouri|11233|192.168.56.171|     PI_06|14/6/2014|null|
|  4|     Abra|   Wright|50155 South Mongo...|      Port Jervis| New Jersey|17751| 192.168.56.52|     PI_09|11/6/2014|null|
|  5|    Stone|   Palmer|12191 West Armeni...|          Henderson|  Nebraska|03560| 192.168.56.85|     PI_08| 8/6/2014|null|
|  6|   Regina|   Bryant|29073  Henderson Ct.|        Texarkana|Tennessee|18224|  192.168.56.5|     PI_10|11/6/2014|null|
|  7| Donovan|  Aguirre|77718 East Farmin...|   Winston-Salem|  New York|95234|192.168.56.114|     PI_05| 9/6/2014|null|
|  8|   Aileen|  Mendoza|46855 East Russia...|      Schenectady|  Illinois|68284| 192.168.56.51|     PI_02|13/6/2014|null|
|  9|   Mariam|   Henson|   84567  Gambia Ct.|         Owensboro|    Hawaii|09146|192.168.56.214|     PI_07|13/6/2014|null|
| 10|    Silas|   Hughes|86635 North Ghana...|           Beverly|  Virginia|08642|192.168.56.253|     PI_03|12/6/2014|null|
| 11|    Robin|    Vance|88311 West Austri...|         Aberdeen|West Virginia|23155| 192.168.56.14|     PI_09|14/6/2014|null|
| 12|   Galvin|     Lane|41866 East Niue Ave.|    East Lansing|  Mississippi|16347|192.168.56.198|     PI_04|13/6/2014|null|
| 13|    Alexa| Hammond|43093 East Tokela...|           Parker| South Dakota|89666| 192.168.56.68|     PI_02|14/6/2014|null|
| 14|  Tatyana|   Sparks|65672 West Detroi...|  Jeffersonville|  Kentucky|83750| 192.168.56.95|     PI_04|12/6/2014|null|
```

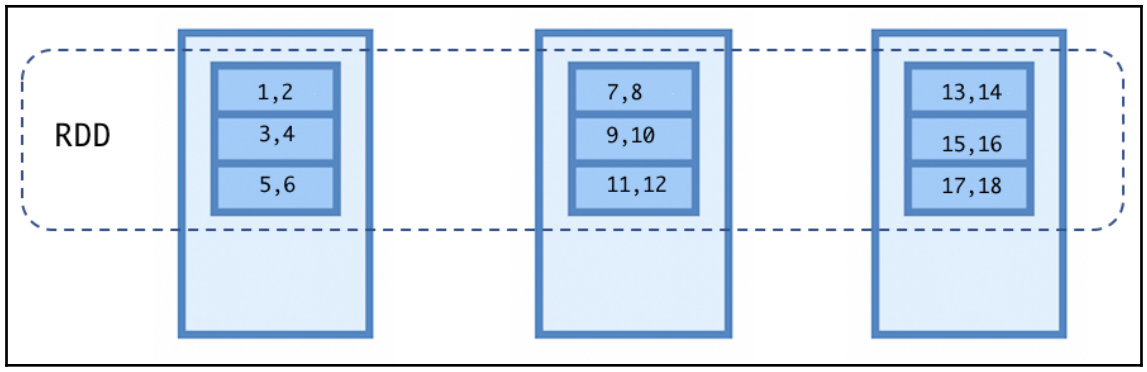
```
ubuntu@ip-172-31-16-208:/opt$ spark-sql
2019-01-18 22:33:52 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classe
2019-01-18 22:33:54 INFO  HiveMetaStore:589 - 0: Opening raw store with implemenation class:org.apache.hadoop.hive.metastore.Object
2019-01-18 22:33:54 INFO  ObjectStore:289 - ObjectStore, initialize called
2019-01-18 22:33:54 INFO  Persistence:77 - Property hive.metastore.integral.jdo.pushdown unknown - will be ignored
2019-01-18 22:33:54 INFO  Persistence:77 - Property datanucleus.cache.level2 unknown - will be ignored
2019-01-18 22:33:56 INFO  ObjectStore:370 - Setting MetaStore object pin classes with hive.metastore.cache.pinobjtypes="Table,Stora
ieldSchema,Order"
2019-01-18 22:33:59 INFO  Datastore:77 - The class "org.apache.hadoop.hive.metastore.model.MFieldSchema" is tagged as "embedded-onl
2019-01-18 22:33:59 INFO  Datastore:77 - The class "org.apache.hadoop.hive.metastore.model.MOrder" is tagged as "embedded-only" so
2019-01-18 22:33:59 INFO  Datastore:77 - The class "org.apache.hadoop.hive.metastore.model.MFieldSchema" is tagged as "embedded-onl
2019-01-18 22:33:59 INFO  Datastore:77 - The class "org.apache.hadoop.hive.metastore.model.MOrder" is tagged as "embedded-only" so
2019-01-18 22:33:59 INFO  Query:77 - Reading in results for query "org.datanucleus.store.rdbms.query.SQLQuery@0" since the connecti
2019-01-18 22:33:59 INFO  MetaStoreDirectSql:139 - Using direct SQL, underlying DB is DERBY
```

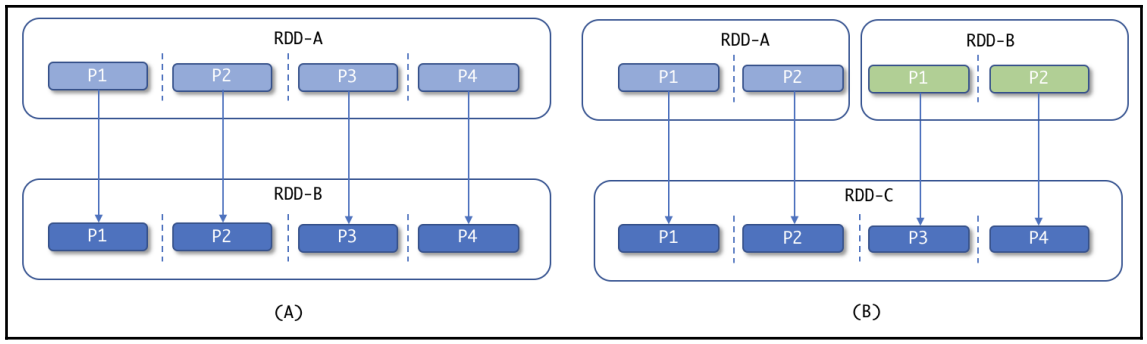
```
2019-01-18 22:34:02 INFO  Query:77 - Reading in results for query "org.datanucleus.store.rdbms.query.SQLQuery@0" since the connection used is closin
2019-01-18 22:34:02 INFO  MetaStoreDirectSql:139 - Using direct SQL, underlying DB is DERBY
2019-01-18 22:34:02 INFO  ObjectStore:272 - Initialized ObjectStore
2019-01-18 22:34:03 INFO  StateStoreCoordinatorRef:54 - Registered StateStoreCoordinator endpoint
spark-sql>
```

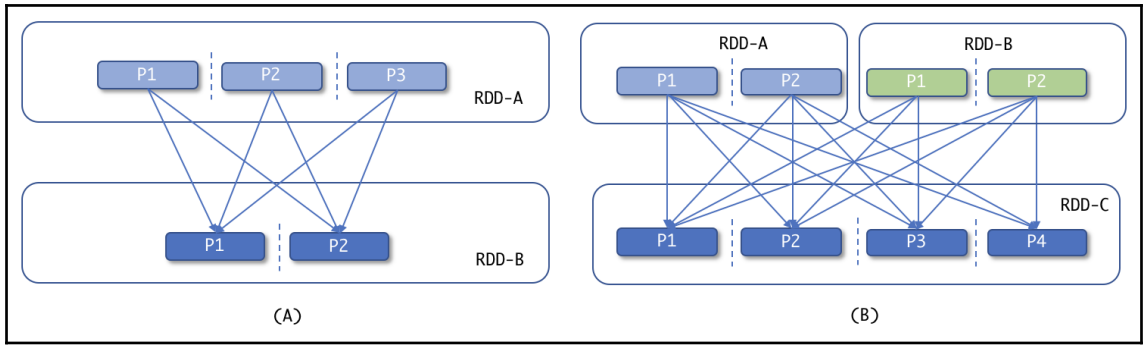

Not Secure 18.216.92.231:4040/executors/
Spark 2.3.1
Jobs
Stages
Storage
Environment
Executors
Spark shell
Executors
Summary
RDD Blocks
Storage Memory
Disk Used
Cores
Active Tasks
Failed Tasks
Complete Tasks
Total Tasks
Task Time (GC Time)
Input
Shuffle Read
Shuffle Write
Blacklisted
Active(1) 0 0.0 B / 434 MB 0.0 B 1 0 0 0 0 0 ms (0 ms) 0.0 B 0.0 B 0.0 B 0
Dead(0) 0 0.0 B / 0.0 B 0.0 B 0 0 0 0 0 0 ms (0 ms) 0.0 B 0.0 B 0.0 B 0
Total(1) 0 0.0 B / 434 MB 0.0 B 1 0 0 0 0 0 ms (0 ms) 0.0 B 0.0 B 0.0 B 0
Executors
Show 20 entries
Search:
Executor ID
Address
Status
RDD Blocks
Storage Memory
Disk Used
Cores
Active Tasks
Failed Tasks
Complete Tasks
Total Tasks
Task Time (GC Time)
Input
Shuffle Read
Shuffle Write
Thread Dump
driver ip-172-31-16-208.us-east-2.compute.internal:44665 Active 0 0.0 B / 434 MB 0.0 B 1 0 0 0 0 0 ms (0 ms) 0.0 B 0.0 B 0.0 B Thread Dump

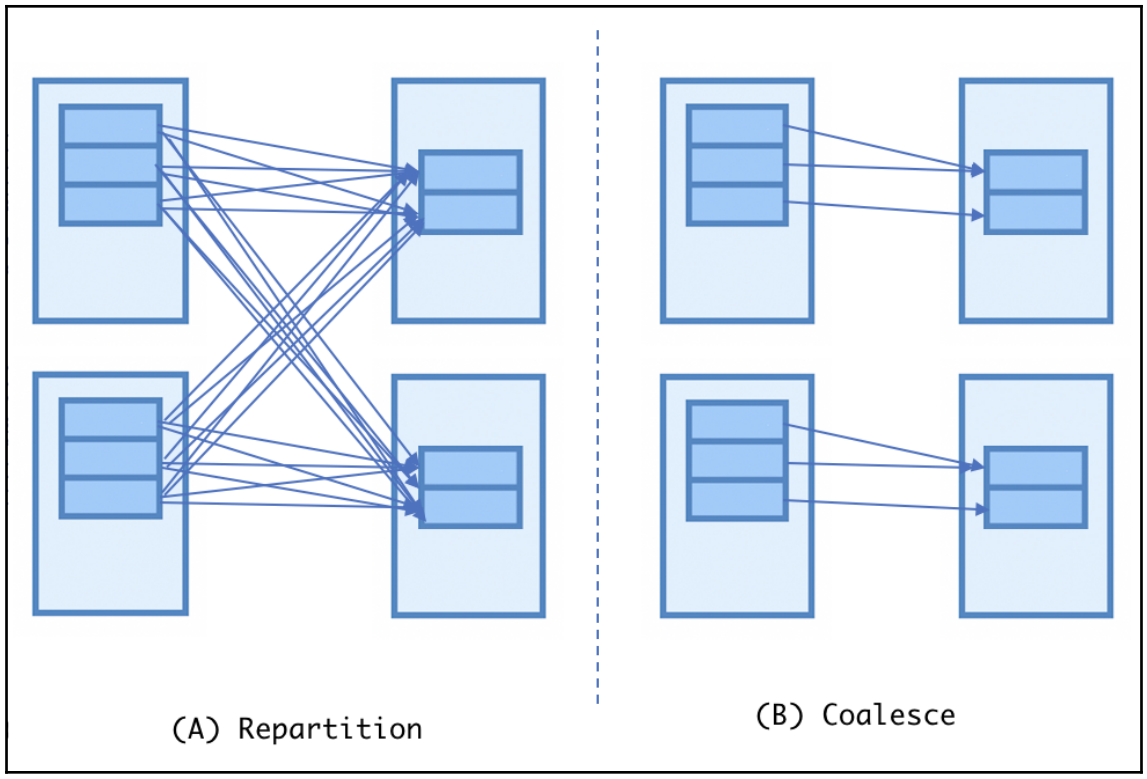
Chapter 3: Spark RDD






RDD-A
P1
P2
P3
P4
RDD-B
P1
P2
P3
P4
(A)
RDD-A
P1
P2
RDD-B
P1
P2
RDD-C
P1
P2
P3
P4
(B)


P1
P2
P3
RDD-A
P1
P2
RDD-B
(A)
RDD-A
P1
P2
RDD-B
P1
P2
RDD-C
P1
P2
P3
P4
(B)


(A) Repartition
(B) Coalesce

```
scala> sales_df.printSchema()
root
 |-- id: string (nullable = true)
 |-- firstname: string (nullable = true)
 |-- lastname: string (nullable = true)
 |-- address: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- zip: string (nullable = true)
 |-- ip: string (nullable = true)
 |-- product_id: string (nullable = true)
 |-- date_of_purchase: string (nullable = true)
```

```
scala> sales_df.select("firstname").show()
+---------+
|firstname|
+---------+
|     Zena|
|   Elaine|
|     Sage|
|     Cade|
|     Abra|
|    Stone|
|   Regina|
|  Donovan|
|   Aileen|
|   Mariam|
|    Silas|
|    Robin|
|   Galvin|
|    Alexa|
|  Tatyana|
|     Yuri|
|     Raya|
|  Ulysses|
|   Edward|
|  Emerald|
+---------+
only showing top 20 rows
```

```
scala> sales_df.filter($"id" < 15).show()
+---+---------+---------+--------------------+----------------+-------------+-----+--------------+----------+----------------+
| id|firstname| lastname|             address|            city|        state|  zip|            ip|product_id|date_of_purchase|
+---+---------+---------+--------------------+----------------+-------------+-----+--------------+----------+----------------+
|  0|     Zena|     Ross|41228 West India Ln.|          Powell|    Tennessee|21550|192.168.56.127|     PI_09|       13/6/2014|
|  1|   Elaine|   Bishop|15903 North North...|Hawaiian Gardens|       Alaska|06429|192.168.56.105|     PI_03|        8/6/2014|
|  2|     Sage|  Carroll| 6880  Greenland Ct.|      Guayanilla|       Nevada|08899| 192.168.56.40|     PI_03|       13/6/2014|
|  3|     Cade|Singleton|64021 South Bulga...|          Derby|     Missouri|11233|192.168.56.171|     PI_06|       14/6/2014|
|  4|     Abra|   Wright|50155 South Mongo...|     Port Jervis|   New Jersey|17751| 192.168.56.52|     PI_09|       11/6/2014|
|  5|    Stone|   Palmer|12191 West Armeni...|       Henderson|     Nebraska|03560| 192.168.56.85|     PI_08|        8/6/2014|
|  6|   Regina|   Bryant|29073  Henderson Ct.|       Texarkana|    Tennessee|18224|  192.168.56.5|     PI_10|       11/6/2014|
|  7|  Donovan|  Aguirre|77718 East Farmin...|   Winston-Salem|     New York|95234|192.168.56.114|     PI_05|        9/6/2014|
|  8|   Aileen|  Mendoza|46855 East Russia...|     Schenectady|     Illinois|68284| 192.168.56.51|     PI_02|       13/6/2014|
|  9|   Mariam|   Henson|   84567  Gambia Ct.|       Owensboro|       Hawaii|09146|192.168.56.214|     PI_07|       13/6/2014|
| 10|    Silas|   Hughes|86635 North Ghana...|         Beverly|     Virginia|08642|192.168.56.253|     PI_03|       12/6/2014|
| 11|    Robin|    Vance|88311 West Austri...|       Aberdeen|West Virginia|23155| 192.168.56.14|     PI_09|       14/6/2014|
| 12|   Galvin|     Lane|41866 East Niue Ave.|    East Lansing|  Mississippi|16347|192.168.56.198|     PI_04|       13/6/2014|
| 13|    Alexa| Hammond|43093 East Tokela...|          Parker| South Dakota|89666| 192.168.56.68|     PI_02|       14/6/2014|
| 14|  Tatyana|   Sparks|65672 West Detroi...|  Jeffersonville|     Kentucky|83750| 192.168.56.95|     PI_04|       12/6/2014|
+---+---------+---------+--------------------+----------------+-------------+-----+--------------+----------+----------------+
```

```
scala> sales_df.groupBy("ip").count().show()
+--------------+-----+
|            ip|count|
+--------------+-----+
|192.168.56.141|   35|
| 192.168.56.30|   41|
|192.168.56.129|   36|
| 192.168.56.91|   39|
| 192.168.56.36|   40|
|192.168.56.170|   46|
|192.168.56.173|   44|
| 192.168.56.54|   45|
|192.168.56.100|   32|
| 192.168.56.70|   35|
|192.168.56.190|   39|
|  192.168.56.7|   33|
|192.168.56.119|   40|
| 192.168.56.34|   29|
| 192.168.56.56|   39|
| 192.168.56.57|   49|
|192.168.56.104|   42|
| 192.168.56.15|   41|
| 192.168.56.13|   36|
|192.168.56.216|   47|
+--------------+-----+
only showing top 20 rows
```

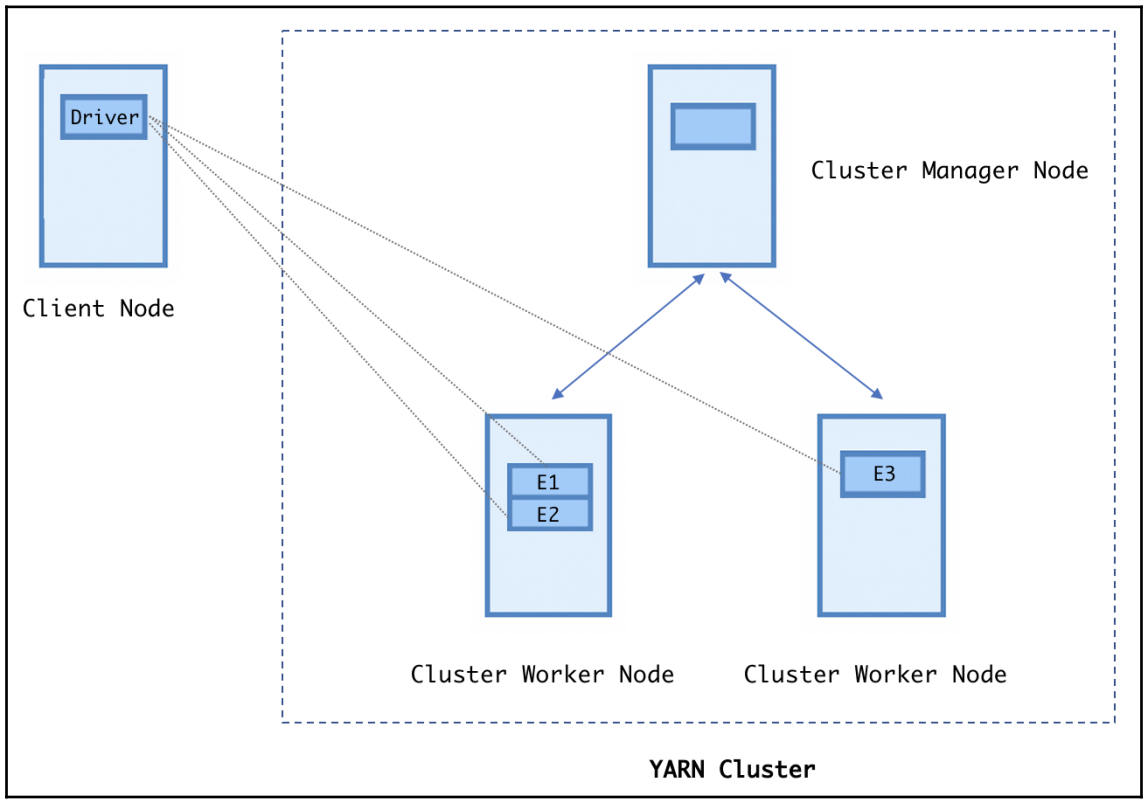
```
[scala> import org.apache.spark.sql.types._
 import org.apache.spark.sql.types._

[scala> import org.apache.spark.sql.Encoders
 import org.apache.spark.sql.Encoders

[scala> import org.apache.spark.sql.SparkSession
 import org.apache.spark.sql.SparkSession

[scala> val spark = SparkSession.builder().appName("Spark DataSet example").config("spark.config.option", "value").getOrCreate()
 19/01/23 20:12:20 WARN SparkSession$Builder: Using an existing SparkSession; some configuration may not take effect.
 spark: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@733fa95c

[scala> case class Sales (id: Int, firstname: String,lastname: String,address: String,city: String,state: String,zip: String,ip: String,product_id: String,dop: String)
 defined class Sales

[scala> val sales_ds = spark.read.option("sep", "\t").option("header", "true").csv("file:///opt/data/sample_10000.txt").withColumn("id", 'id.cast(IntegerType)).as[Sales]
 sales_ds: org.apache.spark.sql.Dataset[Sales] = [id: int, firstname: string ... 8 more fields]

[scala> sales_ds.show()
 +---+---------+---------+--------------------+--------------------+-------------+-----+--------------+----------+---------+
 | id|firstname| lastname|             address|                city|        state|  zip|            ip|product_id|      dop|
 +---+---------+---------+--------------------+--------------------+-------------+-----+--------------+----------+---------+
 |  0|     Zena|     Ross|41228 West India Ln.|             Powell|    Tennessee|21550|192.168.56.127|     PI_09|13/6/2014|
 |  1|   Elaine|   Bishop|15903 North North...|Hawaiian Gardens|       Alaska|06429|192.168.56.105|     PI_03| 8/6/2014|
 |  2|     Sage|  Carroll| 6880  Greenland Ct.|          Guayanilla|       Nevada|08899| 192.168.56.40|     PI_03|13/6/2014|
 |  3|     Cade|Singleton|64021 South Bulga...|              Derby|     Missouri|11233|192.168.56.171|     PI_06|14/6/2014|
 |  4|     Abra|   Wright|50155 South Mongo...|         Port Jervis|   New Jersey|17751| 192.168.56.52|     PI_09|11/6/2014|
 |  5|    Stone|   Palmer|12191 West Armeni...|           Henderson|     Nebraska|03560| 192.168.56.85|     PI_08| 8/6/2014|
 |  6|   Regina|   Bryant|29073  Henderson Ct.|           Texarkana|    Tennessee|18224|  192.168.56.5|     PI_10|11/6/2014|
 |  7| Donovan|  Aguirre|77718 East Farmin...|    Winston-Salem|     New York|95234|192.168.56.114|     PI_05| 9/6/2014|
 |  8|   Aileen|  Mendoza|46855 East Russia...|         Schenectady|     Illinois|68284| 192.168.56.51|     PI_02|13/6/2014|
 |  9|   Mariam|   Henson|   84567  Gambia Ct.|           Owensboro|       Hawaii|09146|192.168.56.214|     PI_07|13/6/2014|
 | 10|    Silas|   Hughes|86635 North Ghana...|             Beverly|     Virginia|08642|192.168.56.253|     PI_03|12/6/2014|
```

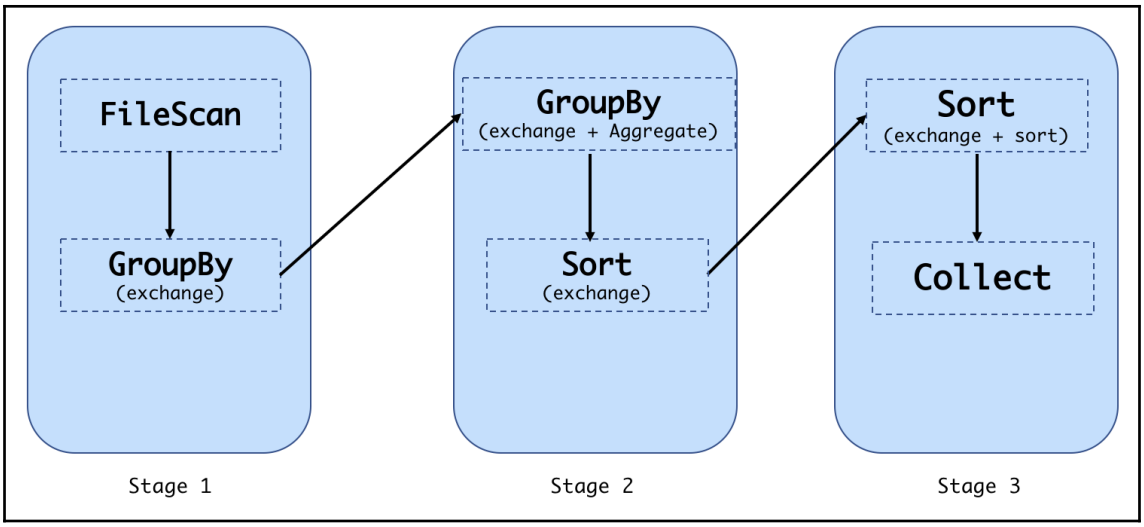
```
10 // Loading dataset from CSV
11 import org.apache.spark.sql.types._
12 import org.apache.spark.sql.Encoders
13 var sales_ds = spark.read.option("sep", "\t").option("header", "true").csv("/FileStore/tables/sample_10000.txt").withColumn("id",
   'id.cast(IntegerType)).as[Sales]
14 // Displays the content of the Dataset to stdout
15 sales_ds.explain
```

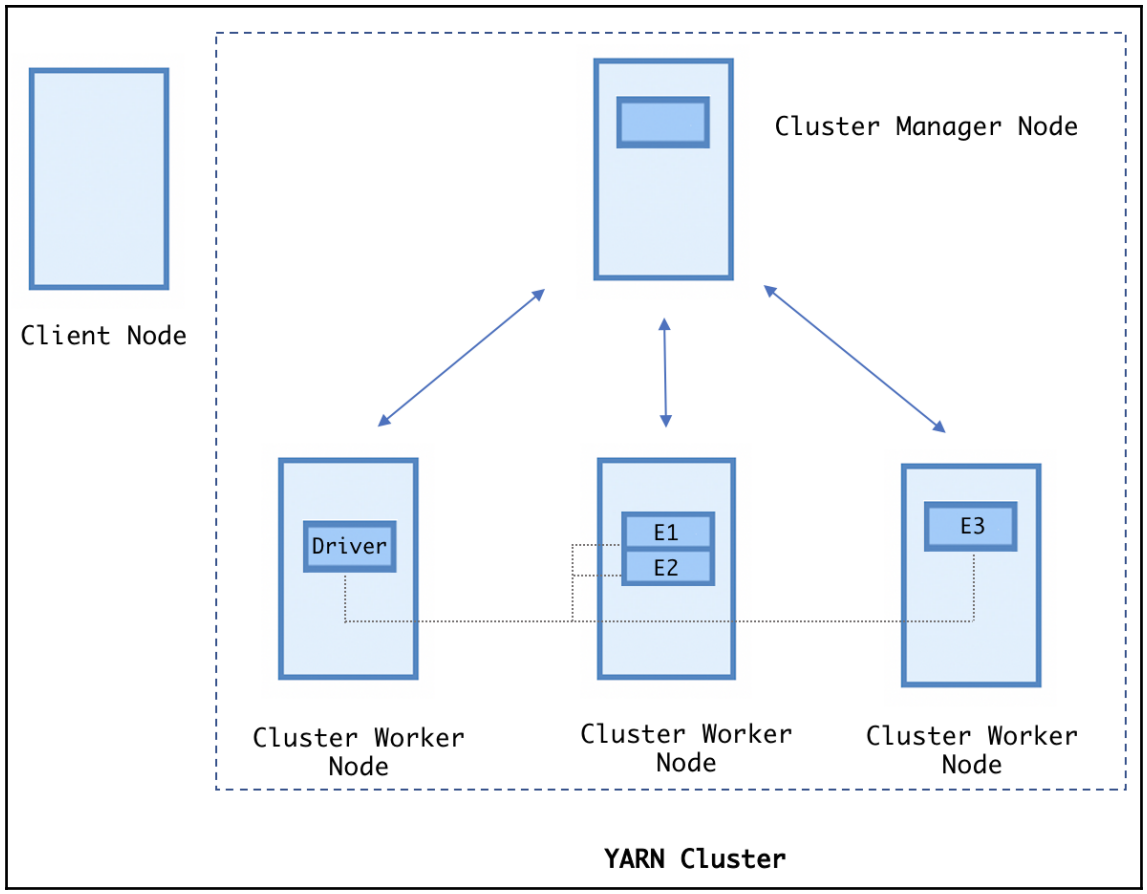
▾ (1) Spark Jobs
▸ Job 219 View (Stages: 1/1)

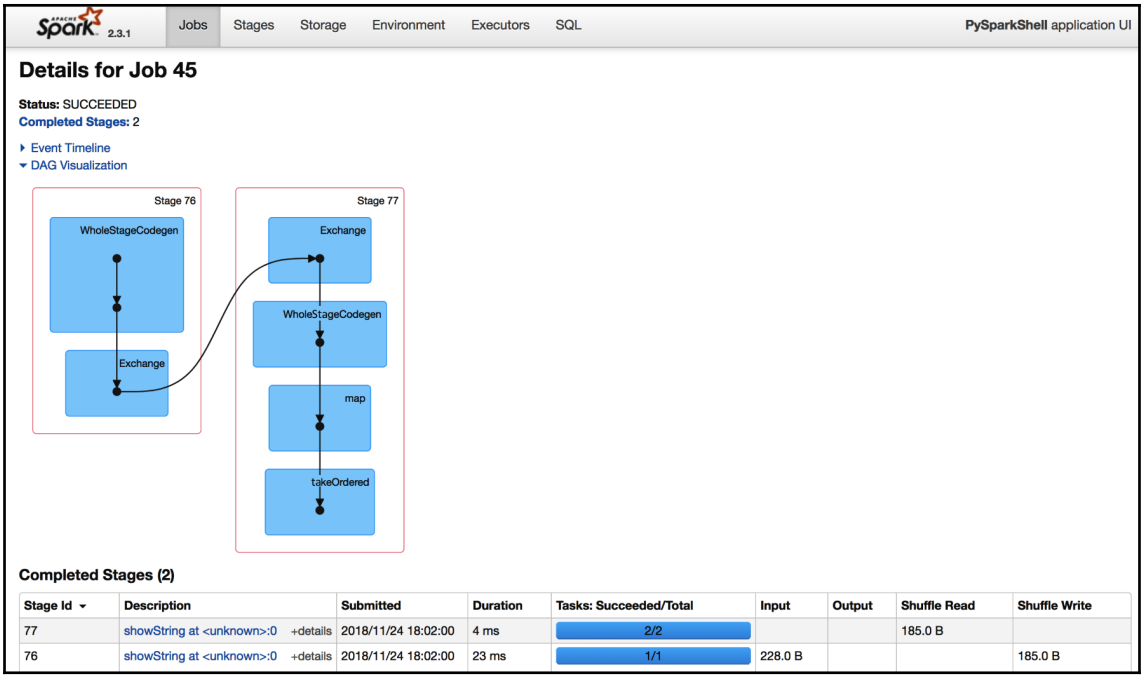
```
== Physical Plan ==
*(1) Project [cast(id#6994 as int) AS id#7016, firstname#6995, lastname#6996, address#6997, city#6998, state#6999, zip#7000, ip#7001, product_id#70
02, dop#7003, _c10#7004]
+- *(1) FileScan csv [id#6994,firstname#6995,lastname#6996,address#6997,city#6998,state#6999,zip#7000,ip#7001,product_id#7002,dop#7003,_c10#7004] B
atched: false, DataFilters: [], Format: CSV, Location: InMemoryFileIndex[dbfs:/FileStore/tables/sample_10000.txt], PartitionFilters: [], PushedFilt
ers: [], ReadSchema: struct<id:string,firstname:string,lastname:string,address:string,city:string,state:string,zip:str...
import org.apache.spark.sql.SparkSession
spark: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@55083c90
import spark.implicits._
defined class Sales
import org.apache.spark.sql.types._
import org.apache.spark.sql.Encoders
sales_ds: org.apache.spark.sql.Dataset[Sales] = [id: int, firstname: string ... 9 more fields]
```

Chapter 5: Spark Architecture and Application Execution Flow




FileScan
GroupBy
(exchange)
GroupBy
(exchange + Aggregate)
Sort
(exchange)
Sort
(exchange + sort)
Collect
Stage 1
Stage 2
Stage 3


Client Node
Cluster Manager Node
Driver
E1
E2
E3
Cluster Worker Node
Cluster Worker Node
Cluster Worker Node
YARN Cluster


Spark 2.3.1
Jobs
Stages
Storage
Environment
Executors
SQL
PySparkShell application UI
Details for Job 45
Status: SUCCEEDED
Completed Stages: 2
Event Timeline
DAG Visualization
Stage 76
WholeStageCodegen
Exchange
Stage 77
Exchange
WholeStageCodegen
map
takeOrdered
Completed Stages (2)
Stage Id
Description
Submitted
Duration
Tasks: Succeeded/Total
Input
Output
Shuffle Read
Shuffle Write
77
showString at <unknown>:0
+details
2018/11/24 18:02:00
4 ms
2/2
185.0 B
76
showString at <unknown>:0
+details
2018/11/24 18:02:00
23 ms
1/1
228.0 B
185.0 B

Chapter 6: Spark SQL

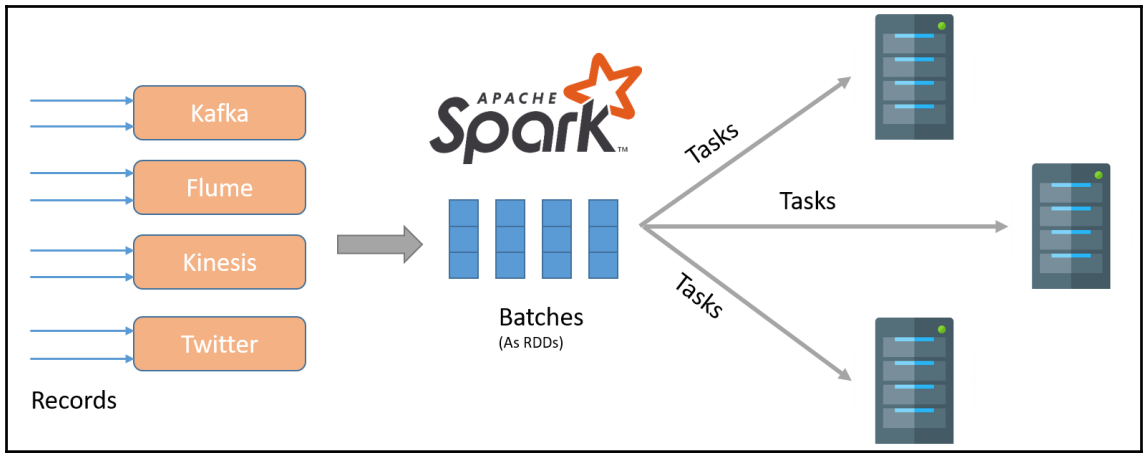
```
spark-sql> show databases;
default
Time taken: 2.584 seconds, Fetched 1 row(s)
spark-sql> Create Database if not exists mydb
         > location '/opt/sparkdb';
chgrp: changing ownership of 'file:///opt/sparkdb': chown:
Time taken: 0.342 seconds
spark-sql> show databases;
default
mydb
Time taken: 0.061 seconds, Fetched 2 row(s)
spark-sql>
```

```
spark-sql> Describe Database mydb;
Database Name   mydb
Description
Location        file:/opt/sparkdb
Time taken: 0.089 seconds, Fetched 3 row(s)
spark-sql> Describe Database Extended mydb;
Database Name   mydb
Description
Location        file:/opt/sparkdb
Properties
Time taken: 0.063 seconds, Fetched 4 row(s)
```

```
spark-sql> CREATE TABLE mytable (id String, firstname String,address String, city String, State String, zip String, ip String, product_id String)
         > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
         > LOCATION '/opt/data'
         > Stored as TEXTFILE;
Time taken: 0.338 seconds
spark-sql> select * from mytable limit 2;
id      firstname       lastname        address city    state   zip     ip
0       Zena    Ross    41228 West India Ln.    Powell  Tennessee       21550   192.168.56.127
Time taken: 2.969 seconds, Fetched 2 row(s)
spark-sql>
```

```
spark-sql> LOAD DATA LOCAL INPATH '/opt/data/sample_10000.txt' INTO TABLE mytable;
Time taken: 0.491 seconds
```

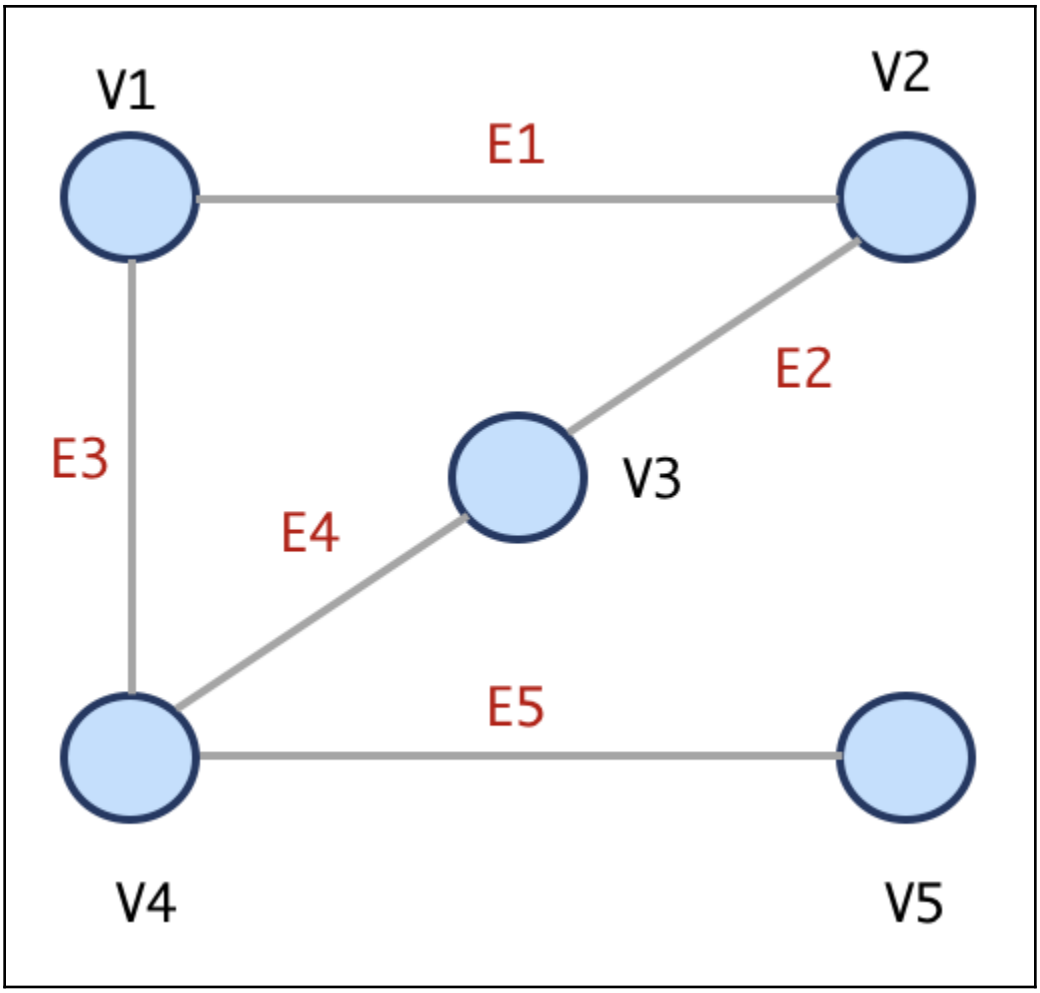
Chapter 7: Spark Streaming, Machine Learning, and Graph Analysis




```
ubuntu@ip-172-31-16-208:/opt$ spark-submit --num-executors 1 --executor-memory 1g --total-executor-cores 1 streaming.py
19/01/19 20:15:59 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
-------------------------------------------
Time: 2019-01-19 20:16:04
-------------------------------------------

-------------------------------------------
Time: 2019-01-19 20:16:05
-------------------------------------------

-------------------------------------------
Time: 2019-01-19 20:16:06
-------------------------------------------

-------------------------------------------
```

```
ubuntu@ip-172-31-16-208:~$ nc -lk 8888
this is spark streaming application. this count streaming words
```

```
-------------------------------------------
Time: 2019-01-19 20:16:22
-------------------------------------------

-------------------------------------------
Time: 2019-01-19 20:16:23
-------------------------------------------
('this', 2)
('is', 1)
('streaming', 2)
('count', 1)
('spark', 1)
('application.', 1)
('words', 1)
```


V1
V2
E1
E2
E3
V3
E4
E5
V4
V5

Chapter 8: Spark Optimizations