ВСТУП В SPARK ТА HADOOP

КАЛЬЧЕНКО ІЛЛЯ

Data Engineer в NIX

ПЛАН:

- ЗАГАЛЬНИЙ ОГЛЯД
- ВСТАНОЛЕННЯ ТА КОНФІГУРАЦІЯ
- СТРУКТУРИ ДАНИХ
- ОРОБКА ДАНИХ
- ПРАКТИКА НА AWS

Вступ в Spark та Hadoop

Ілля Кальченко
Data Engineer в Nix

План лекції

Загальний огляд

Hadoop та Spark

Встановлення

SparkSession та SparkConf

Структури даних

RDD, Dataframe та Dataset

Обробка даних

SQL та функції, трансформації та дії

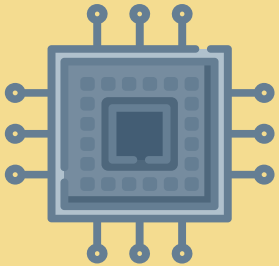
Практика

Запуск Spark програм на AWS

01

Spark та Hadoop

Загальний огляд

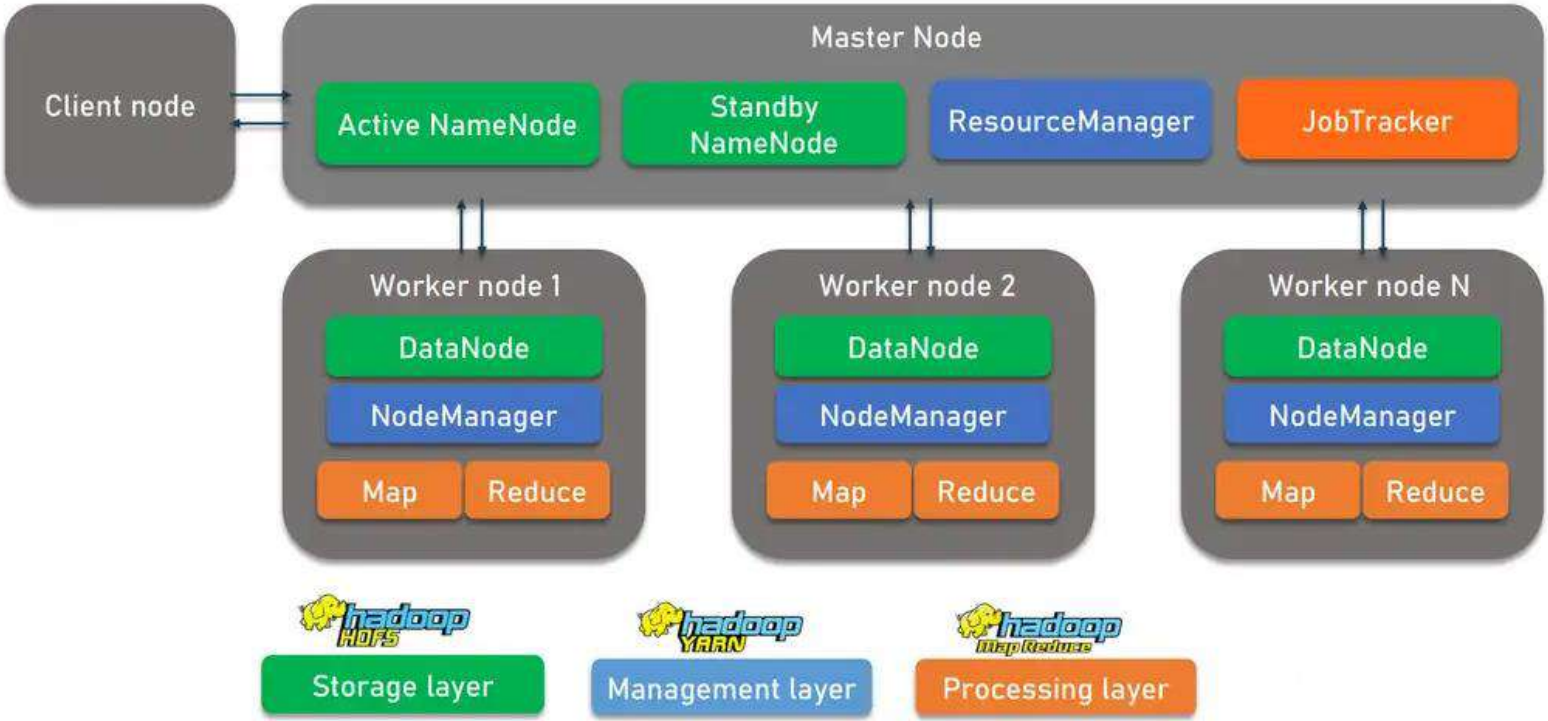


Cluster, Driver, Executor, Task, Job, Partitions

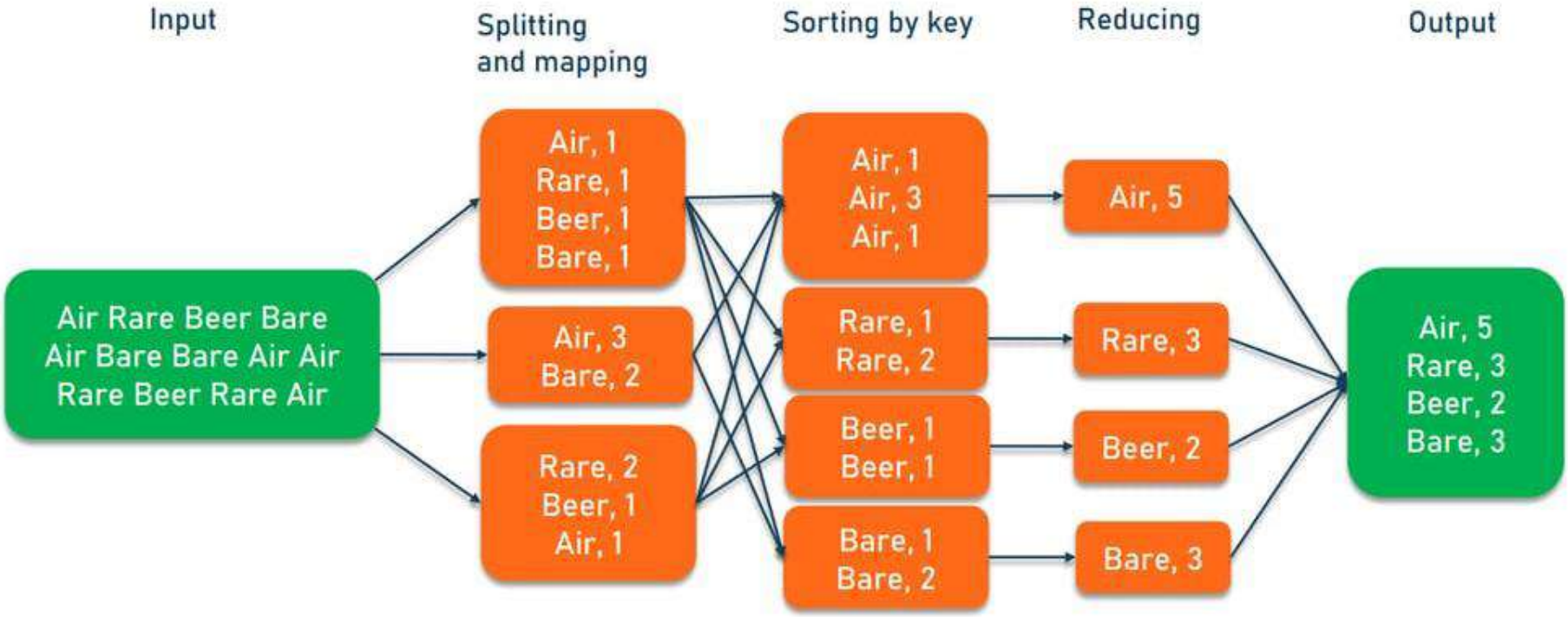
Екосистема Hadoop

Hadoop Core	2008-2009	2010-2012	2013 and later
Processing engine	NoSQL database	Event streaming	SQL on Hadoop
File system	Coordination	Log data ingesting	Real-time data analytics
Resource management	SQL-like scripting	SQL-like scripting	SQL on HBase
	Machine learning	Scheduling	Graph processing
	RDBMS to HDFS data transfer	Stream and batch processing	Stream processing

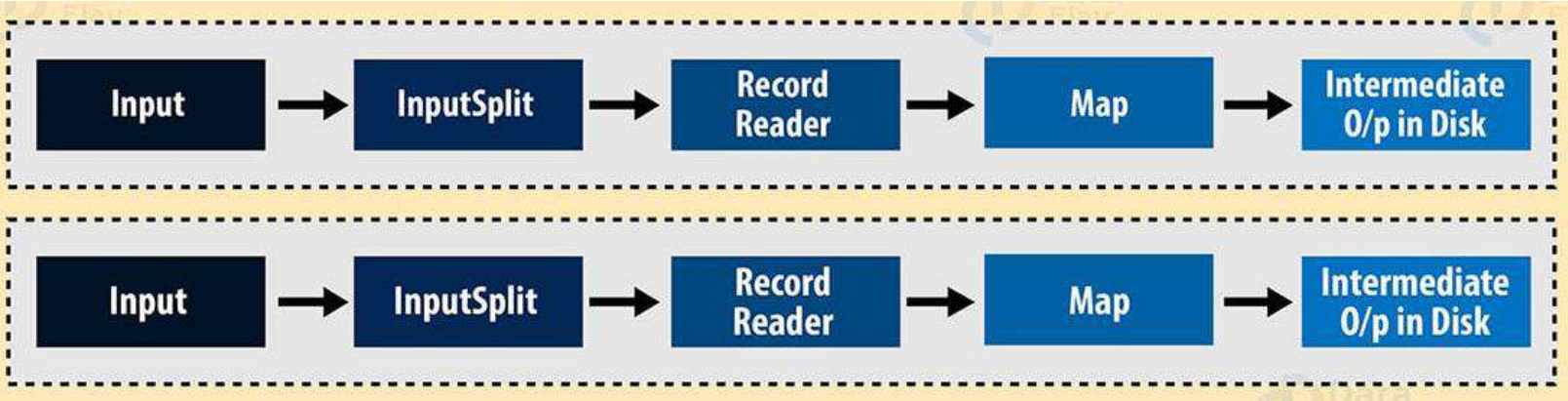
Архітектура кластера Hadoop



Алгоритм MapReduce



Етап Map

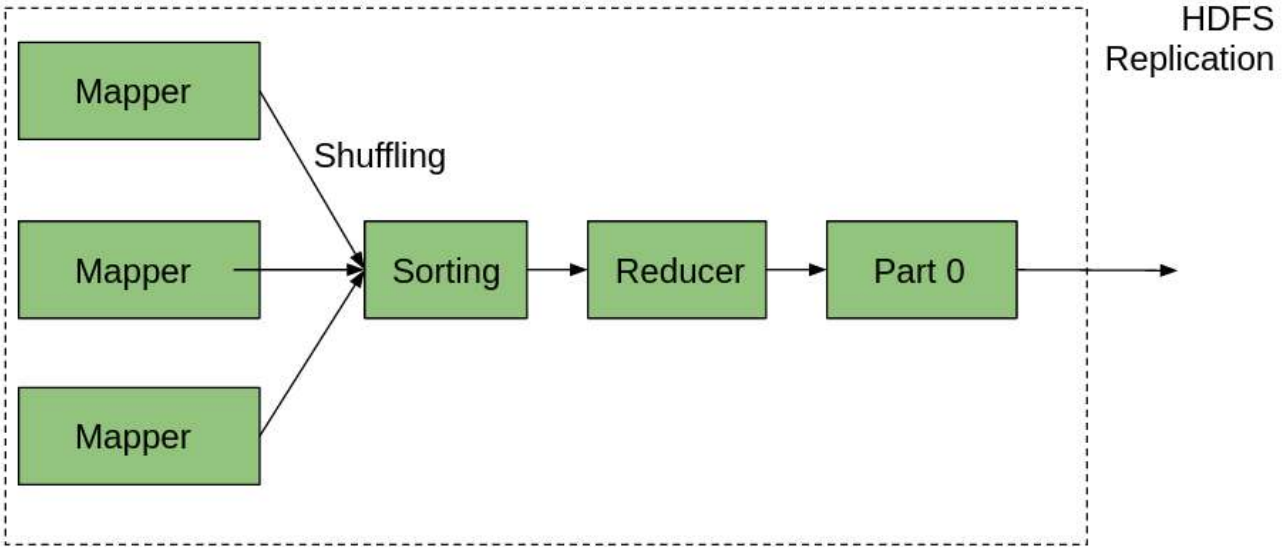


Node 1: Hello from Hadoop
Node 2: Goodbye from Hadoop

Node 1
<Hello, 1>
<from, 1>
<Hadoop, 1>

Node 2
<Goodbye, 1>
<from, 1>
<Hadoop, 1>

Етап Reduce



Node 1
<Hello, 1>
<from, 1>
<Hadoop, 1>

Node 2
<Goodbye, 1>
<from, 1>
<Hadoop, 1>

Node 1
<from, 1>
<from, 1>
<Hello, 1>
(alphabet. order)

Node 2
<Goodbye, 1>
<Hadoop, 1>
<Hadoop, 1>

Node 1
<from, 2>
<Hello, 1>

Node 2
<Goodbye, 1>
<Hadoop, 2>

Приклад етапу Map

```python
def map(input_string: str) -> List[tuple[str, int]]:
    words_tokenized = input_string.lower().replace(".", "").split(" ")
    result_collection = []
    for word in words_tokenized:
        result_collection.append((word, 1))
    return result_collection
```

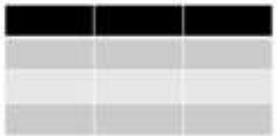
```python
test_str = """I like dogs and cycling. I also like going for a walk with dogs."""
mapped_str = map(test_str)
# [('i', 1), ('like', 1), ('dogs', 1), ('and', 1), ('cycling', 1), ('i', 1), ('also', 1)
# ('like', 1), ('going', 1), ('for', 1), ('a', 1), ('walk', 1), ('with', 1), ('dogs', 1)
```

Приклад етапу Reduce

```python
def reduce(mapped_words: List[tuple[str, int]]) -> tuple[str, int]:
    word_sum = 0
    for word, one in mapped_words:
        word_sum += one
    return word, word_sum
```

```python
test_mapped_words = [('like', 1), ('like', 1), ('like', 1), ('like', 1), ('like', 1)]
reduced_word = reduce(test_mapped_words)
# ('like', 5)
```

Основні модулі Spark



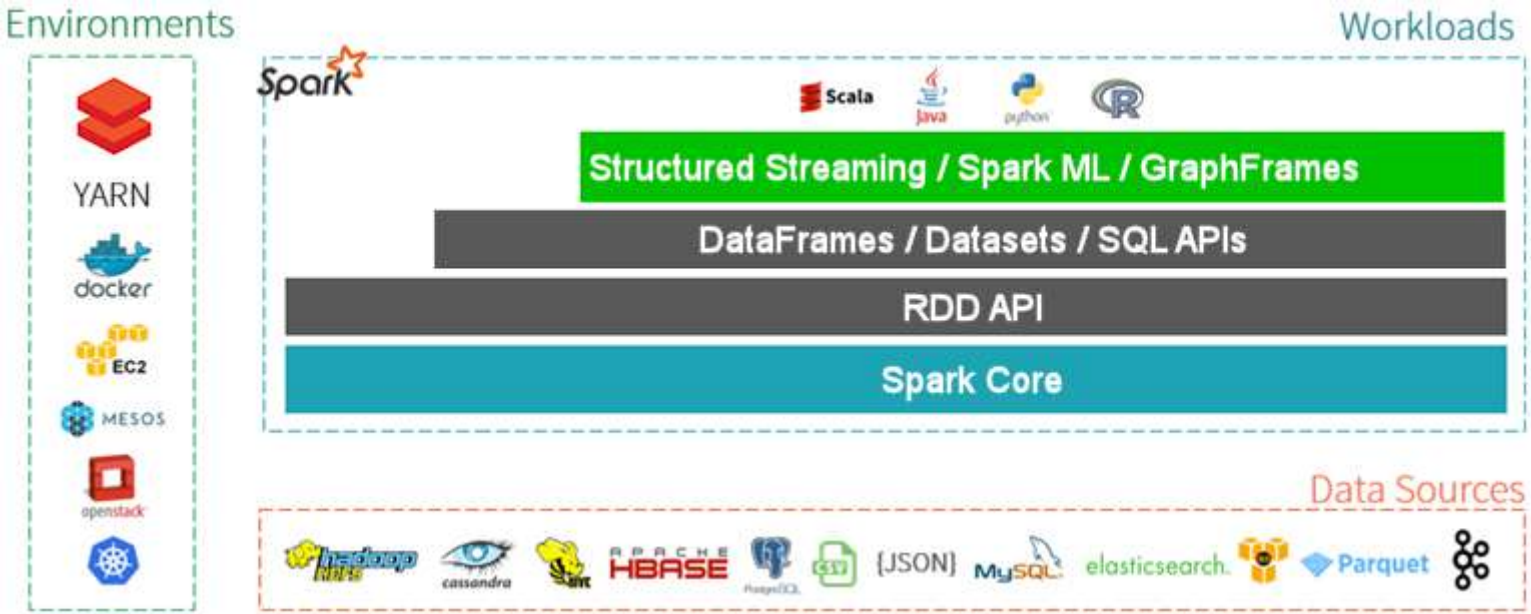
SQL / DataFrames

Machine Learning

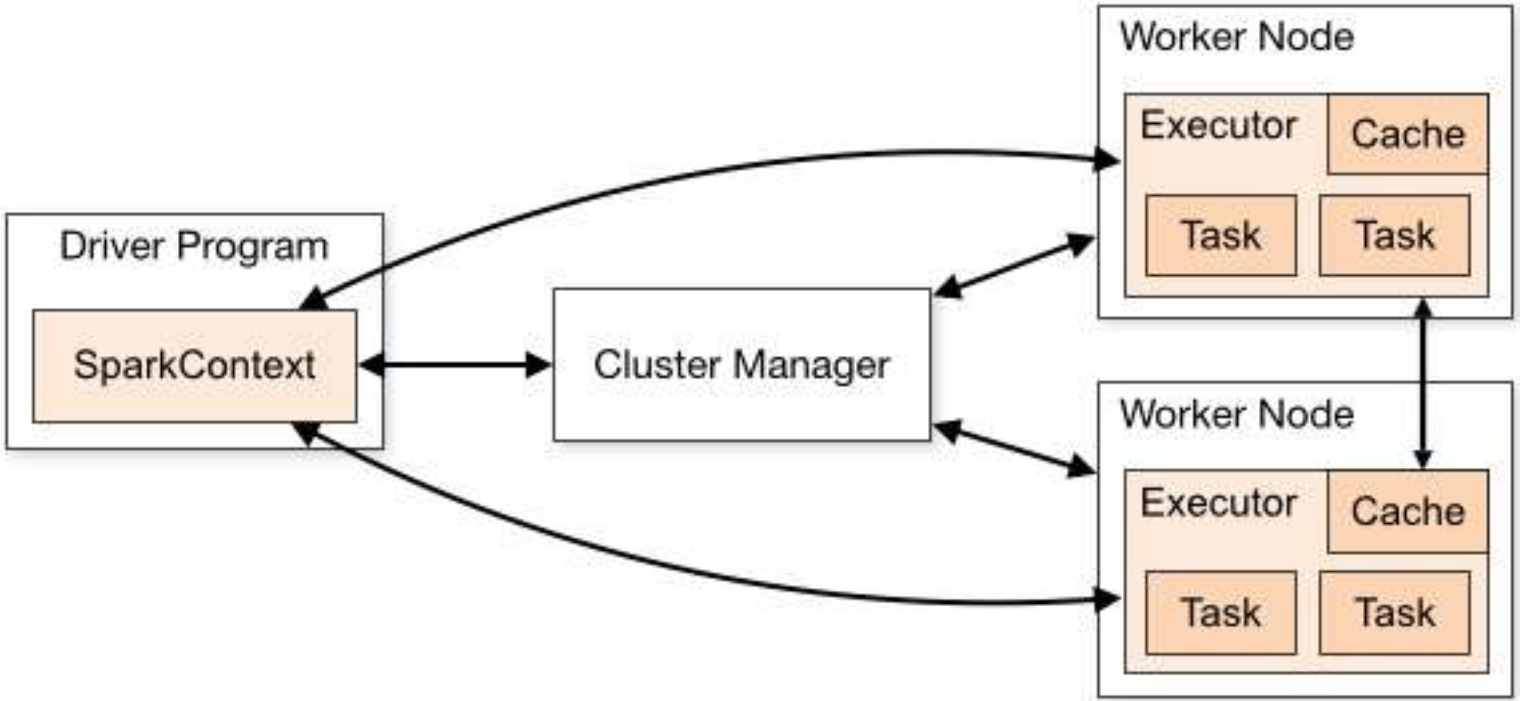
Graph

Streaming

Рушій Spark

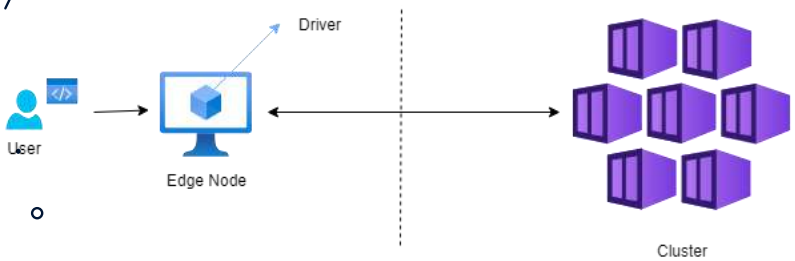

Environments
YARN
docker
EC2
MESOS
openstack
Spark
Workloads
Scala
Java
python
Structured Streaming / Spark ML / GraphFrames
DataFrames / Datasets / SQL APIs
RDD API
Spark Core
Data Sources
hadoop
cassandra
HBASE
{JSON}
MySQL
elasticsearch
Parquet

Кластер Spark

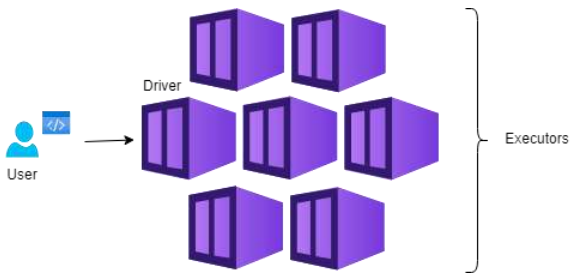


Кластер Spark

Client mode



Cluster mode



Standalone

Підготовка до виконання

Початок

Створюючи завдання, Driver призначає одиниці роботи на слоти для паралельного виконання.

Driver вирішує, як розділити дані для паралельної обробки.

Driver призначає партицію даних кожному завданню.

Після запуску кожне завдання отримуватиме з оригінального джерела даних партицію даних, призначену йому

Виконання

Типи операцій

actions

- `count()`
- `show()`
- `collect()`
- `toPandas()`
- `take(n)`
- `write()`

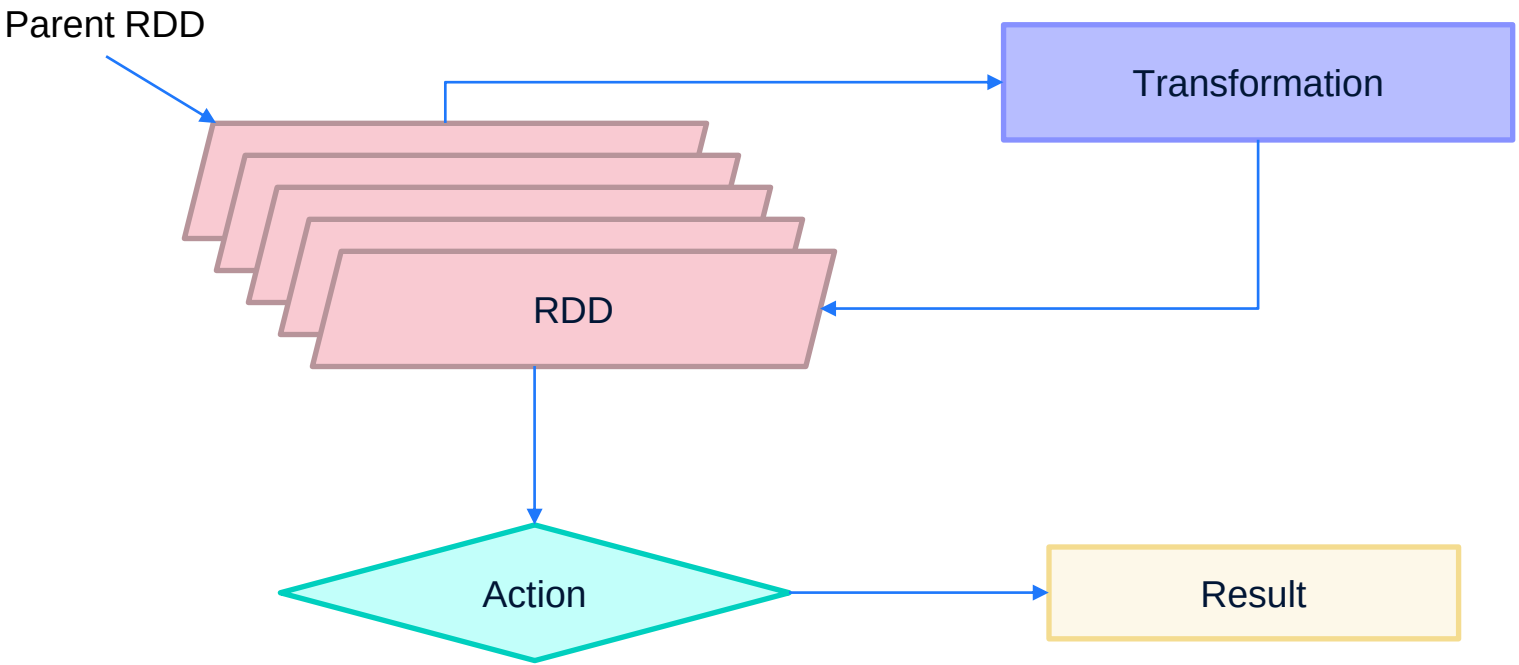
Дані завантажуються у пам'ять

transformations

- `select()`
- `filter()`
- `sort()`
- `distinct()`
- `join ()`
- `groupBy ()`

Дані НЕ завантажуються у пам'ять

«Ліниве» виконання



Типи трансформацій

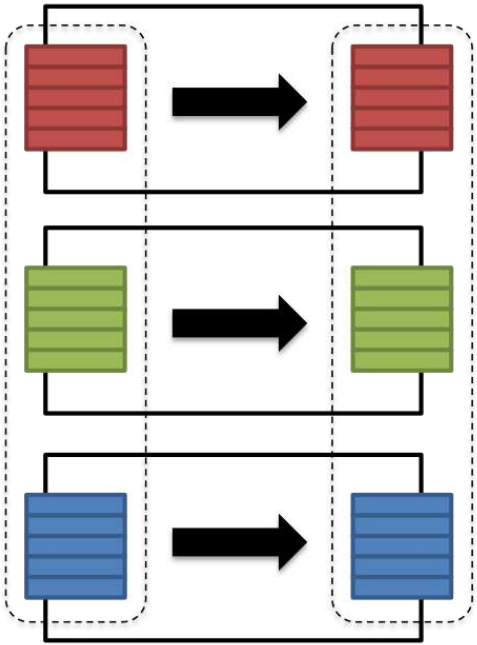
Narrow трансформація

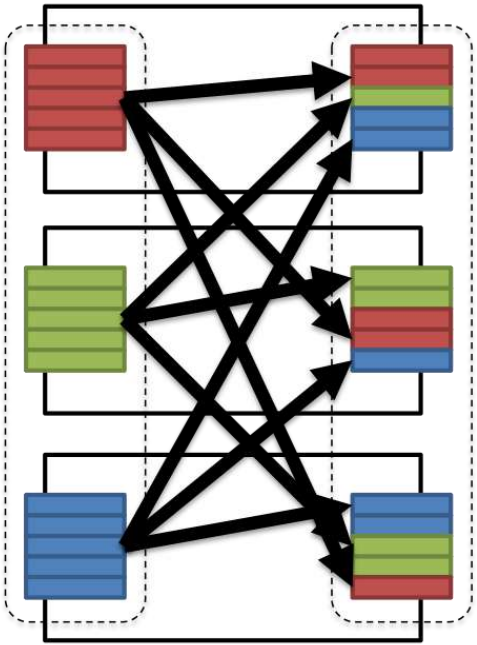
- Вхідні та вихідні дані залишаються в тій самій партиції
- Переміщення даних не потрібне

Wide трансформація

- Потрібні вхідні дані з інших партицій
- Потрібне перемішування даних перед обробкою

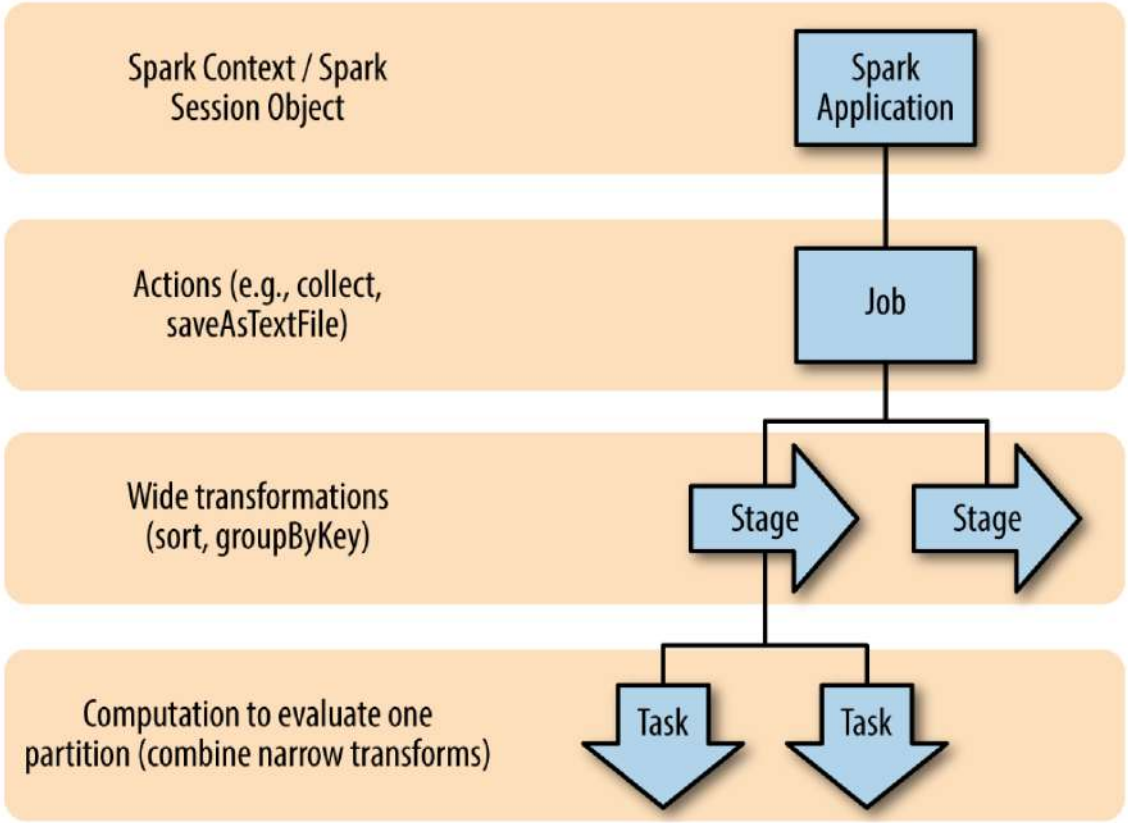
MAP





Reduce

Application, Jobs, stages та tasks



02

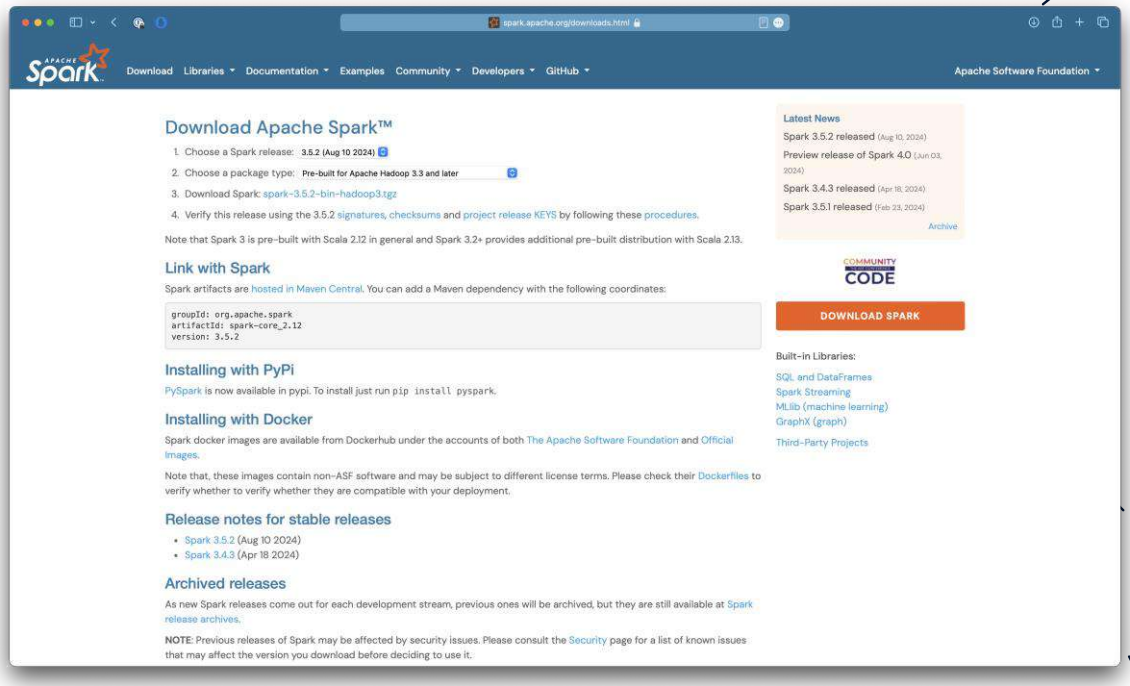
Встановлення Spark

SparkSession та SparkConf

Встановлення

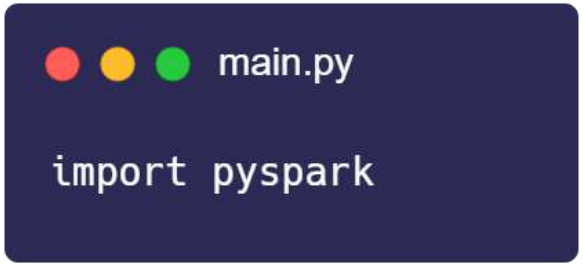
Передумови:

- JRE/JDK 8/11/17
- Python 3.6+



https://spark.apache.org/downloads.html

Встановлення



https://pypi.org/project/pyspark/

SparkSession та SparkConf

SparkSession є точкою входу для програмування в Spark з використанням API Dataset та DataFrame.

```
from pyspark import SparkConf
from pyspark.sql import SparkSession

spark_session = (SparkSession.builder
                             .master("local")
                             .appName("task app")
                             .config(conf=SparkConf())
                             .getOrCreate())
```

03

Структури даних

RDD та Dataframe

RDD та Dataframe

Resilient Distributed Dataset

- Колекція незмінних партицій, розподілених по кластеру. Основні об’єкти Java.
- Вимагає попередньо визначену схему. Немає оптимізації. Обмеження продуктивності.

Підходить для:
неструктурованих даних, низькорівневих операцій

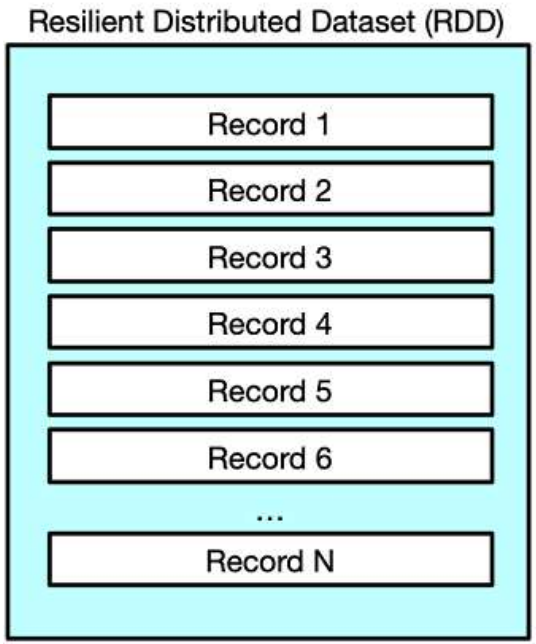
Dataframe

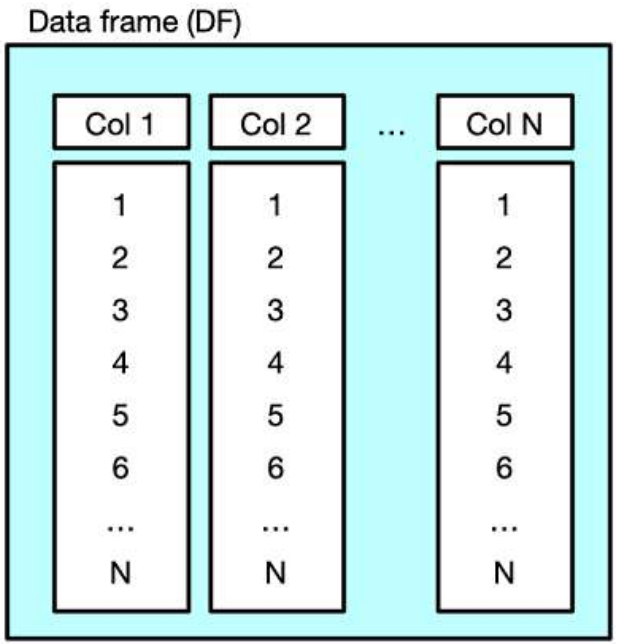
- Розподілена колекція даних, організованих у названі колонки. Концептуально рівнозначна таблиці.
- Може визначати схему. Має Catalyst для оптимізації. Абстракція над RDD.

Підходить для:
структурованих даних, SQL, високорівневих і колонкових операцій

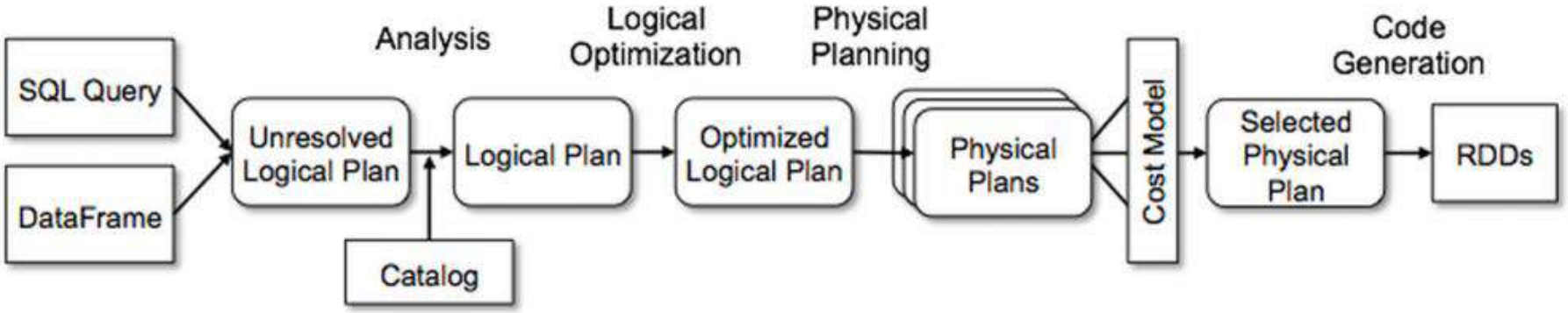
https://techvidvan.com/tutorials/apache-spark-rdd-vs-dataframe/

Представлення даних





Оптимізатор Catalyst

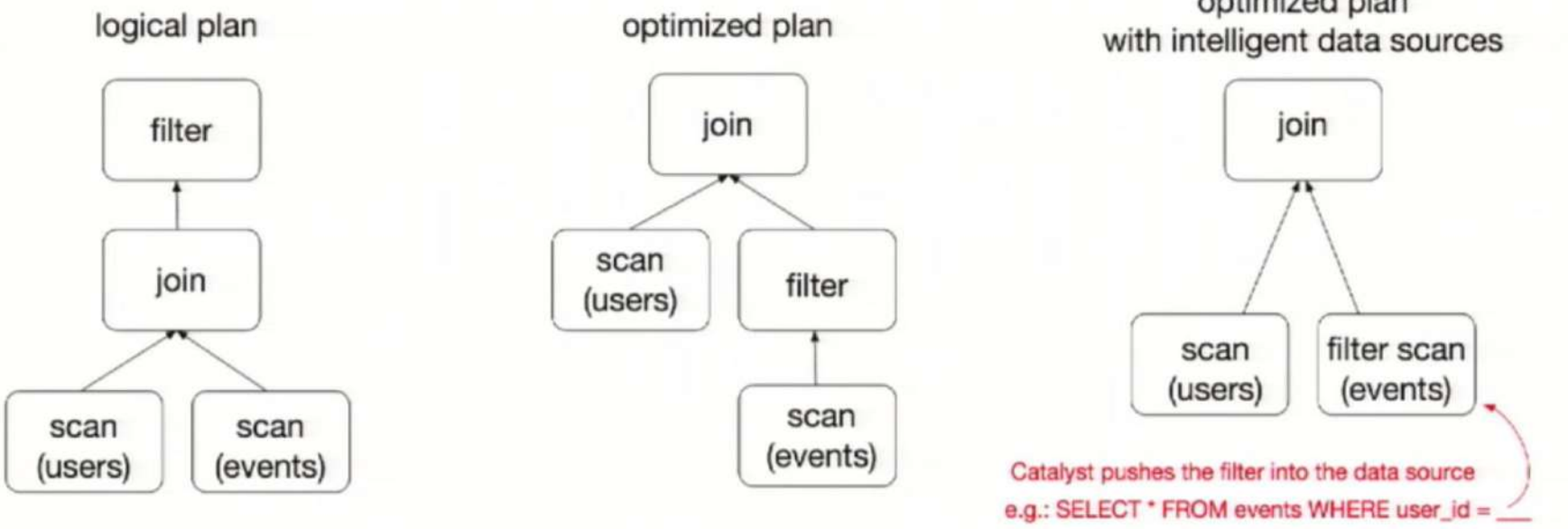



```
df.explain()
```

– отримати план виконання обробки даних
схоже на EXPLAIN ANALYZE з SQL

https://databricks.com/glossary/catalyst-optimizer

Приклад


logical plan
filter
join
scan (users)
scan (events)
optimized plan
join
scan (users)
filter
scan (events)
optimized plan with intelligent data sources
join
scan (users)
filter scan (events)
Catalyst pushes the filter into the data source
e.g.: SELECT * FROM events WHERE user_id =

Створення RDD та DF

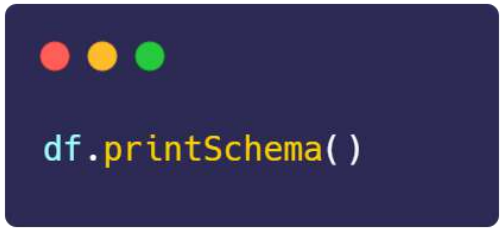
RDD

```
data = [1, 2, 3, 4, 5]
rdd = spark_session.sparkContext.parallelize(data)
```

DataFrame

```
data = [("Ivan", 18), ("Maria", 44)]
schema = t.StructType([
   t.StructField("name", t.StringType(), True),
   t.StructField("age", t.IntegerType(), True)])
df = spark_session.createDataFrame(data, schema)
df.show()
```

Схема Dataframe




```
root
 |-- name: string (nullable = true)
 |-- age: integer (nullable = true)
```

https://spark.apache.org/docs/latest/sql-ref-datatypes.html

Створення Dataframe з файлу

```
from_csv_df = spark_session.read.csv(path)
from_csv_df.show()
```

Може бути parquet, json, avro, orc або text

_c0	_c1	_c2	_c3	_c4	_c5	_c6	_c7	
Summons Number	Plate ID	Registration State	Plate Type	Issue Date	Violation Code	Vehicle Body Type	Vehicle Make	Is
1457617912	JEB5683	NY	PAS	06/25/2021	40	VAN	FORD	
1457617924	JAN2986	NY	PAS	06/25/2021	20	SUBN	DODGE	
1457622427	FJH6630	TX	PAS	06/17/2021	98	SDN	AUDI	
1457638629	RD1Y5N	MO	PAS	06/16/2021	98	SDN	TOYOT	
1457639580	T503814C	NY	OMT	07/04/2021	40	TAXI	HONDA	
1457643042	JLN5490	NY	PAS	06/28/2021	98	SDN	HONDA	
1457663909	UMB4505	VA	PAS	07/02/2021	98	SDN	SUBAR	
1457670471	JPS7544	NY	PAS	06/19/2021	40	SDN	NISSA	
1457670537	UPS7544	NY	PAS	06/19/2021	70	SDN	NISSA	
1457677623	07027R5	TX	PAS	07/03/2021	74	SUBN	LEXUS	
1457677635	07027R5	TX	PAS	07/03/2021	40	SUBN	LEXUS	
1457732816	T649424C	NY	PAS	07/08/2021	20	SDN	TOYOT	
1457732830	T671970C	NY	OMT	07/08/2021	21	SDN	TOYOT	
1457733353	914991C	99	PAS	07/06/2021	24	SDN	DODGE	
1457733365	HGR2844	NY	PAS	07/06/2021	40	SDN	HONDA	
1457734000	KPF6862	NY	PAS	07/02/2021	14	SUBN	JEEP	
1457734011	FGE8882	NY	PAS	07/02/2021	14	SDN	ACURA	
1457734242	KJW2815	NY	PAS	07/04/2021	98	SUBN	HONDA	
1457736044	KLE9108	NY	PAS	07/02/2021	98	SUBN	HONDA	

only showing top 20 rows

Створення Dataframe з файлу

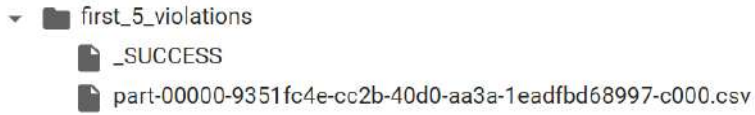
```
from_csv_df = spark_session.read.csv(path,
                                     header=True,
                                     nullValue='null',
                                     dateFormat='MM/dd/yyyy',
                                     schema=violations_schema)

from_csv_df.show()
```

```
+--------------+--------+------------------+----------+----------+--------------+-----------------+------------+-
|Summons Number|Plate ID|Registration State|Plate Type|Issue Date|Violation Code|Vehicle Body Type|Vehicle Make|I
+--------------+--------+------------------+----------+----------+--------------+-----------------+------------+-
|    1457617912| JEB5683|                NY|       PAS|2021-06-25|            40|              VAN|        FORD|
|    1457617924| JAN2986|                NY|       PAS|2021-06-25|            20|             SUBN|       DODGE|
|    1457622427| FJH6630|                TX|       PAS|2021-06-17|            98|              SDN|        AUDI|
|    1457638629|  RD1Y5N|                MO|       PAS|2021-06-16|            98|              SDN|       TOYOT|
|    1457639580|T503814C|                NY|       OMT|2021-07-04|            40|             TAXI|       HONDA|
|    1457643042| JLN5490|                NY|       PAS|2021-06-28|            98|              SDN|       HONDA|
|    1457663909| UMB4505|                VA|       PAS|2021-07-02|            98|              SDN|       SUBAR|
|    1457670471| JPS7544|                NY|       PAS|2021-06-19|            40|              SDN|       NISSA|
|    1457670537| UPS7544|                NY|       PAS|2021-06-19|            70|              SDN|       NISSA|
|    1457677623| 07027R5|                TX|       PAS|2021-07-03|            74|             SUBN|       LEXUS|
|    1457677635| 07027R5|                TX|       PAS|2021-07-03|            40|             SUBN|       LEXUS|
|    1457732816|T649424C|                NY|       PAS|2021-07-08|            20|              SDN|       TOYOT|
|    1457732830|T671970C|                NY|       OMT|2021-07-08|            21|              SDN|       TOYOT|
|    1457733353| 914991C|                99|       PAS|2021-07-06|            24|              SDN|       DODGE|
|    1457733365| HGR2844|                NY|       PAS|2021-07-06|            40|              SDN|       HONDA|
|    1457734000| KPF6862|                NY|       PAS|2021-07-02|            14|             SUBN|        JEEP|
|    1457734011| FGE8882|                NY|       PAS|2021-07-02|            14|              SDN|       ACURA|
|    1457734242| KJW2815|                NY|       PAS|2021-07-04|            98|             SUBN|       HONDA|
|    1457736044| KLE9108|                NY|       PAS|2021-07-02|            98|             SUBN|       HONDA|
|    1457736070| 14400E9|                NJ|       PAS|2021-06-25|            67|             SUBN|         BMW|
+--------------+--------+------------------+----------+----------+--------------+-----------------+------------+-
only showing top 20 rows
```

Збереження в файл

```
first_5_from_csv_df = from_csv_df.limit(5)
path_to_save = '/content/drive/MyDrive/first_5_violations'
first_5_from_csv_df.write.csv(path_to_save)
```



part-00000-9351fc4e-cc2b-40d0-aa3a-1eadfbd68997-c000.csv

Summons Number	Plate ID	Registration State	Plate Type	Issue Date	Violation Code	Vehicle Body Type	Vehicle Make	Issuing Agency	Street Code1	Street Code2	Street Code3
1457617912	JEB5683	NY	PAS	2021-06-25	40	VAN	FORD	P	63430	69230	13490
1457617924	JAN2986	NY	PAS	2021-06-25	20	SUBN	DODGE	P	13490	40404	40404
1457622427	FJH6630	TX	PAS	2021-06-17	98	SDN	AUDI	P	79430	47130	11750
1457638629	RD1Y5N	MO	PAS	2021-06-16	98	SDN	TOYOT	P	53130	23230	23930
1457639580	T503814C	NY	OMT	2021-07-04	40	TAXI	HONDA	P	81030	23930	40030

04

Обробка даних

SQL та функції Dataframe API,
трансформації та дії

Основні трансформації (transformations)

```
violation_df = violation_df.select('plate_id', 'registration_state', 'issue_date', 'violation_code',
                                   'vehicle_make', 'vehicle_color', 'violation_post_code', 'violation_time')
```

plate_id	registration_state	issue_date	violation_code	vehicle_make	vehicle_color	violation_post_code	violation_time
KZH2758	NY	06/09/2023	67	HONDA	BLUE	NULL	0911A
JPD8746	NY	06/30/2023	87	LINCO	GRAY	NULL	0717A
JPD8746	NY	06/30/2023	31	LINCO	GRAY	NULL	0823A
MBH9245	99	07/06/2023	20	KIA	WHITE	NULL	1150P
MBH9245	PA	07/08/2023	40	KIA	WHITE	NULL	1150P
LBZ7486	NY	07/01/2023	14	HONDA	BK	NULL	0618P
JRB4166	FL	06/07/2023	46	HONDA	NULL	NULL	0246P
LCA7360J	NY	06/21/2023	46	HONDA	GRY	NULL	0620P
HNE3840	NY	06/21/2023	14	MAZDA	WHT	NULL	0625P
LAG7093	NY	06/03/2023	46	HONDA	BLUE	NULL	0230P
KXG5820	NY	07/03/2023	14	HONDA	GREY	NULL	0834P
KKX4930	NY	06/29/2023	50	HONDA	WHITE	NULL	0439P
LCV1407	NY	06/29/2023	98	TOYOT	BLK	NULL	0629P
KWG4794	NY	06/29/2023	40	KIA	RED	NULL	1150P
GTK9218	NY	06/30/2023	98	VOLKS	GOLD	NULL	0600P

Основні трансформації (transformations)

```python
violation_df = violation_df.filter(f.col('vehicle_make') == 'HONDA')
# or
violation_df = violation_df.where(f.col('vehicle_make') == 'HONDA')
```

plate_id	registration_state	issue_date	violation_code	vehicle_make	vehicle_color	violation_post_code	violation_time
KZH2758	NY	06/09/2023	67	HONDA	BLUE	NULL	0911A
LBZ7486	NY	07/01/2023	14	HONDA	BK	NULL	0618P
JRB4166	FL	06/07/2023	46	HONDA	NULL	NULL	0246P
LCA7360J	NY	06/21/2023	46	HONDA	GRY	NULL	0620P
LAG7093	NY	06/03/2023	46	HONDA	BLUE	NULL	0230P
KXG5820	NY	07/03/2023	14	HONDA	GREY	NULL	0834P
KKX4930	NY	06/29/2023	50	HONDA	WHITE	NULL	0439P
KTY1801	NY	06/06/2023	40	HONDA	WHT	NULL	1018P
JFV5593	NY	06/26/2023	17	HONDA	RED	NULL	0850A
LCJ3694	NY	06/09/2023	40	HONDA	GRY	NULL	0927P
KLX5185	NY	06/05/2023	71	HONDA	RED	NULL	1229A
LYL9087	NY	01/02/2023	14	HONDA	NULL	NULL	0855A
K86PYJ	NJ	06/12/2023	40	HONDA	BL	NULL	0555A

Основні трансформації (transformations)

vehicle_make	violation_cnt
NIU	10565
PETER	13419
PSE	1
LIBE	2
JAJ	8
YAMAH	5316
POLAR	340
WHC	1
STRAC	12
GLAVA	18
INFTI	4
AGTRU	10
FL/BO	4
Chevr	8068
IC	5944
AZ	7
RED	4
TRE	3
BLK	2
BUZ	1

```
violation_df = violation_df.groupBy('vehicle_make').agg(f.count('*').alias('violation_cnt'))
```

Основні трансформації (transformations)

```
violation_df = violation_df.join(ref_df.select('violation_code', 'violation_description'),
                                 on='violation_code',
                                 how='left')
```

violation_code	plate_id	registration_state	issue_date	vehicle_make	vehicle_color	violation_post_code	violation_time	violation_description
67	KZH2758	NY	06/09/2023	HONDA	BLUE	NULL	0911A	PEDESTRIAN RAMP
87	JPD8746	NY	06/30/2023	LINCO	GRAY	NULL	0717A	FRAUDULENT USE PA...
31	JPD8746	NY	06/30/2023	LINCO	GRAY	NULL	0823A	NO STANDING-COMM ...
20	MBH9245	99	07/06/2023	KIA	WHITE	NULL	1150P	NO PARKING-DAY/TI...
40	MBH9245	PA	07/08/2023	KIA	WHITE	NULL	1150P	FIRE HYDRANT
14	LBZ7486	NY	07/01/2023	HONDA	BK	NULL	0618P	NO STANDING-DAY/T...
46	JRB4166	FL	06/07/2023	HONDA	NULL	NULL	0246P	DOUBLE PARKING
46	LCA7360J	NY	06/21/2023	HONDA	GRY	NULL	0620P	DOUBLE PARKING
14	HNE3840	NY	06/21/2023	MAZDA	WHT	NULL	0625P	NO STANDING-DAY/T...
46	LAG7093	NY	06/03/2023	HONDA	BLUE	NULL	0230P	DOUBLE PARKING
14	KXG5820	NY	07/03/2023	HONDA	GREY	NULL	0834P	NO STANDING-DAY/T...
50	KKX4930	NY	06/29/2023	HONDA	WHITE	NULL	0439P	CROSSWALK
98	LCV1407	NY	06/29/2023	TOYOT	BLK	NULL	0629P	OBSTRUCTING DRIVEWAY
40	KWG4794	NY	06/29/2023	KIA	RED	NULL	1150P	FIRE HYDRANT
98	GTK9218	NY	06/30/2023	VOLKS	GOLD	NULL	0600P	OBSTRUCTING DRIVEWAY
21	HTM79881	NY	06/30/2023	TOYOT	BLUE	NULL	0634P	NO PARKING-STREET...
6	JD32DX	FL	06/30/2023	VOLVO	NULL	NULL	0933P	OVERNIGHT TRACTOR...

Основні дії (actions)

```
violation_df.count()
```

```
violation_df.show()
```

```
violation_df.limit(5).collect()
```

```
24/09/05 13:09:16 WARN NativeCodeLoader:
2060282
(.venv) → ~/education/spark_lecture $
```

```
[Row(plate_id='KZH2758', registration_state='NY', issue_date='06/09/2023', violation_code='67', vehicle
_make='HONDA', vehicle_color='BLUE', violation_post_code=None, violation_time='0911A'), Row(plate_id='L
BZ7486', registration_state='NY', issue_date='07/01/2023', violation_code='14', vehicle_make='HONDA', v
ehicle_color='BK', violation_post_code=None, violation_time='0618P'), Row(plate_id='JRB4166', registrat
ion_state='FL', issue_date='06/07/2023', violation_code='46', vehicle_make='HONDA', vehicle_color=None,
 violation_post_code=None, violation_time='0246P'), Row(plate_id='LCA7360J', registration_state='NY', i
ssue_date='06/21/2023', violation_code='46', vehicle_make='HONDA', vehicle_color='GRY', violation_post_
code=None, violation_time='0620P'), Row(plate_id='LAG7093', registration_state='NY', issue_date='06/03/
2023', violation_code='46', vehicle_make='HONDA', vehicle_color='BLUE', violation_post_code=None, viola
tion_time='0230P')]
```

Завдання

Дано:

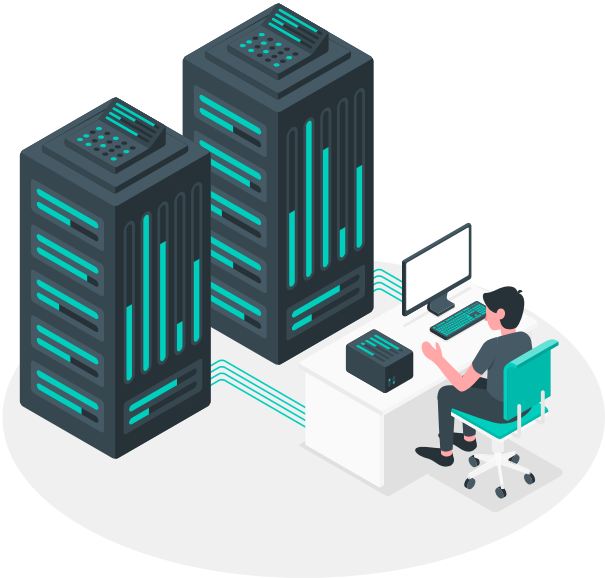
- набір даних Parking Violations Issued - Fiscal Year 2024
- Spark Cluster на AWS EMR

Завдання:

Знайти найбільш часті порушення по кожній марці авто, відсортувати за спаданням кількості таких порушень, та додати колонку з описом порушення

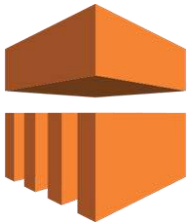
05

Практика Spark



Запуск Spark програм на AWS EMR

Розгортання Spark у хмарному середовищі

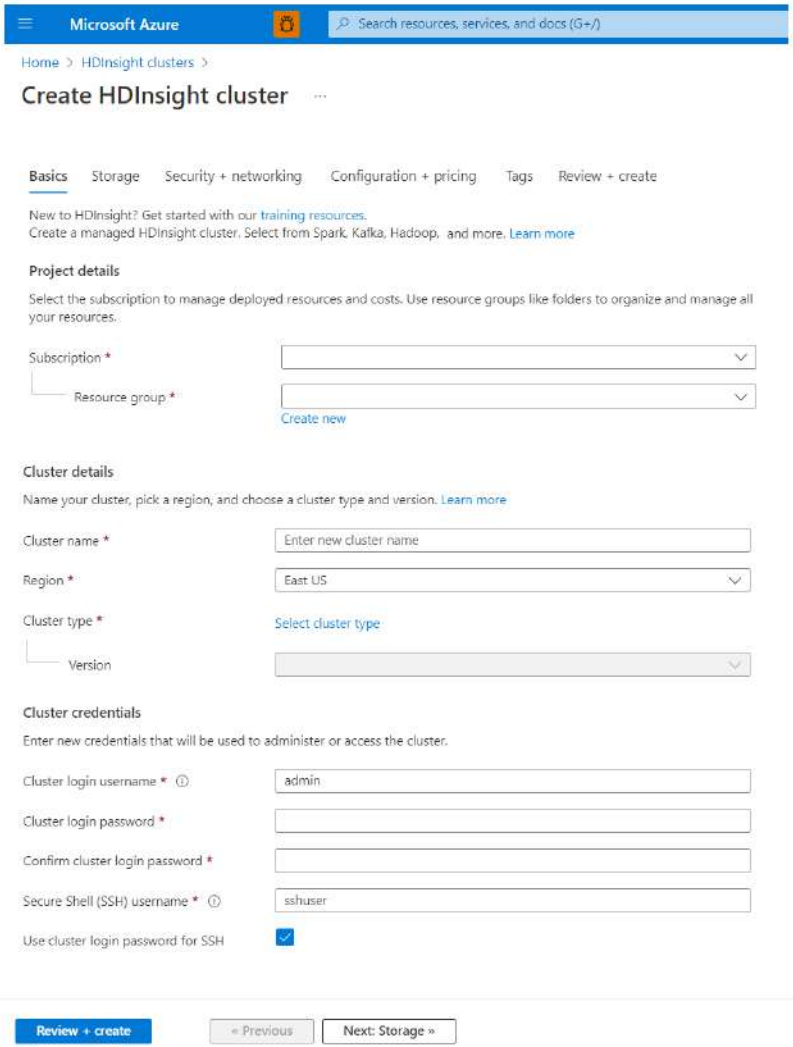


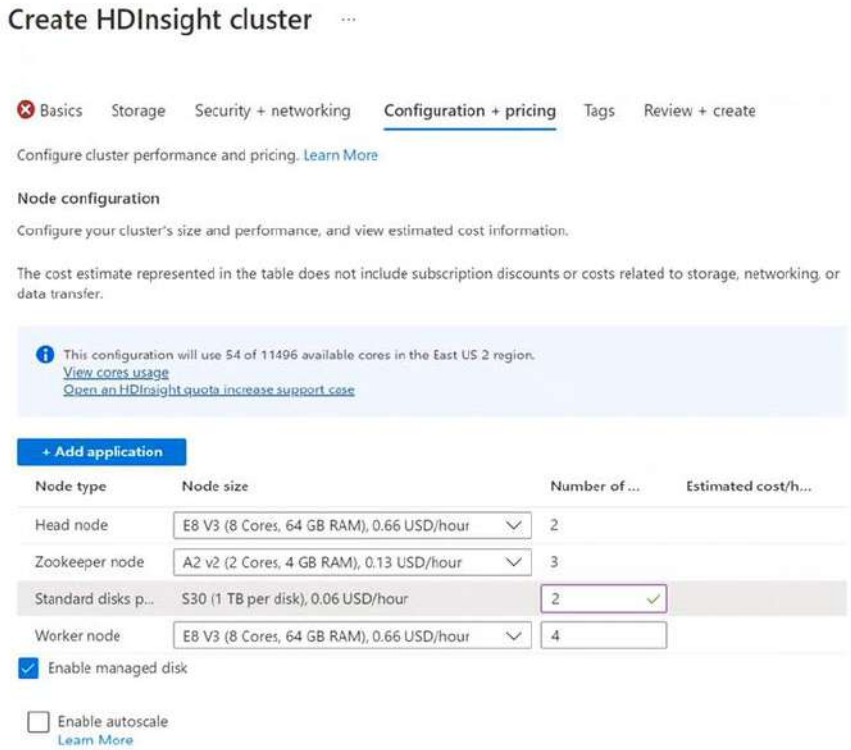
amazon
EMR

Azure HDInsight

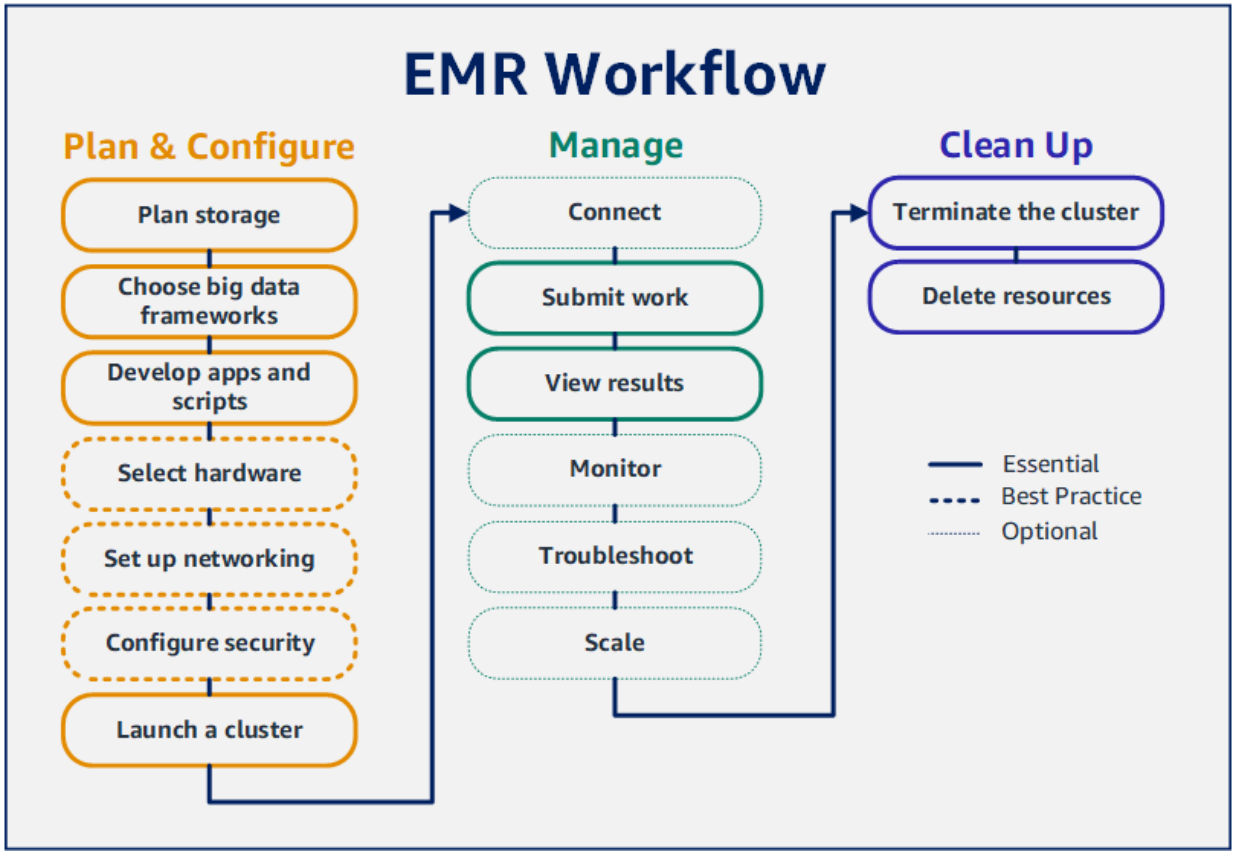
Cloud
Dataproc

Azure HDInsights

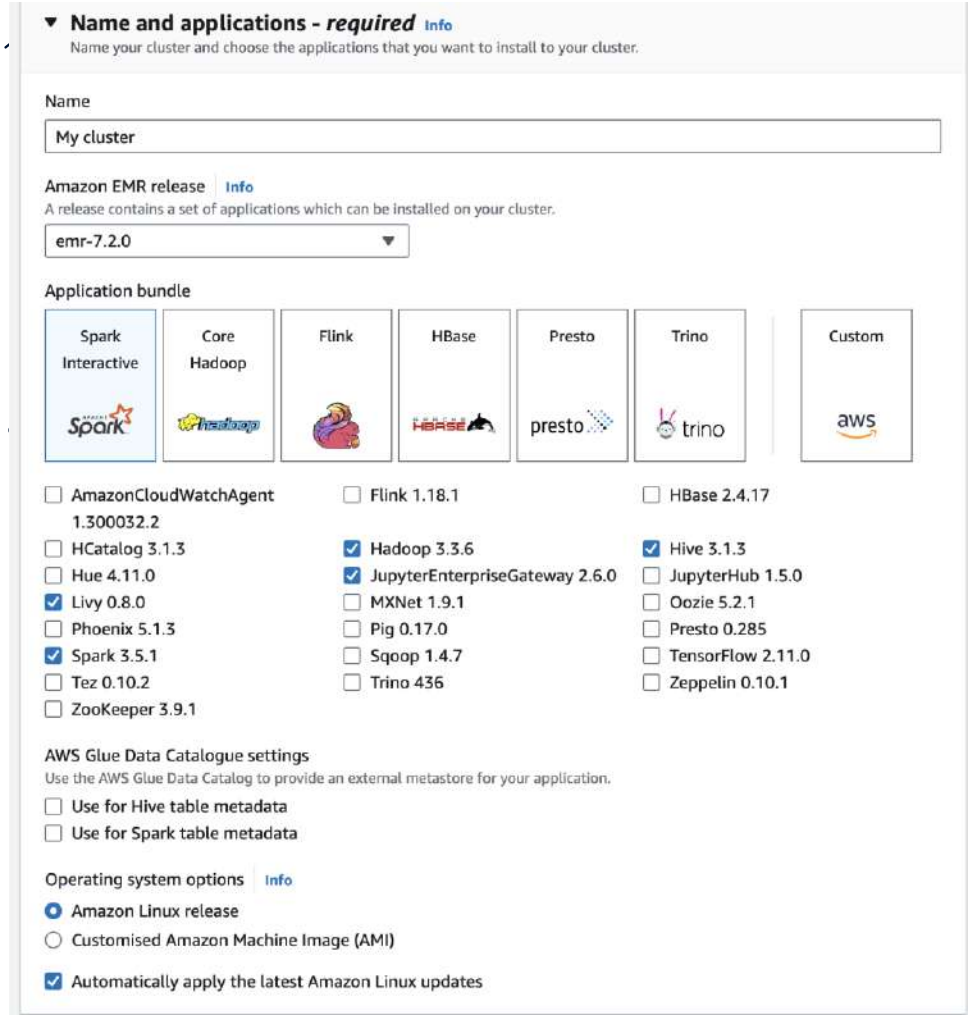


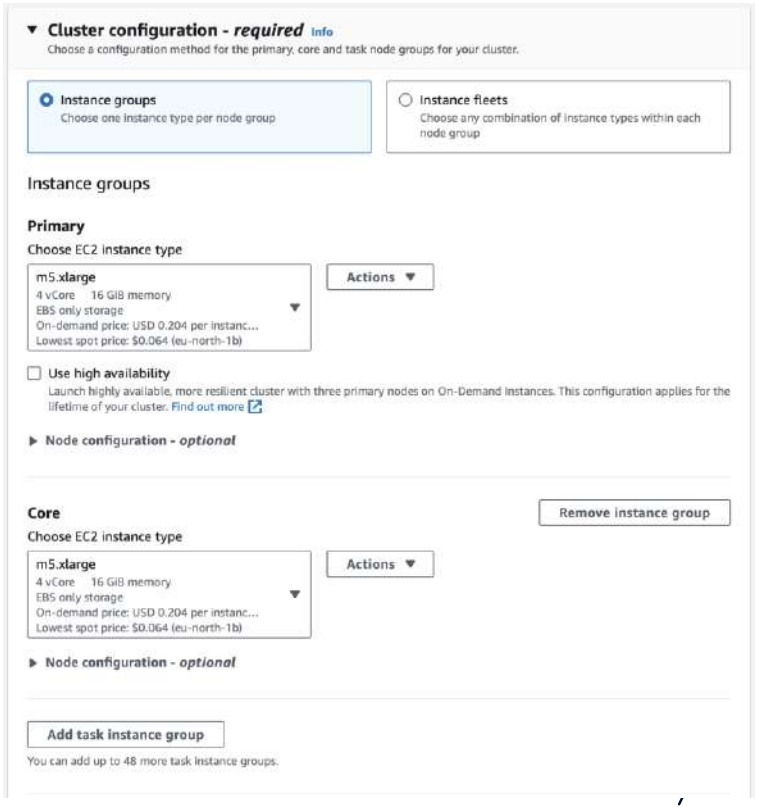


Розгортання Spark у AWS

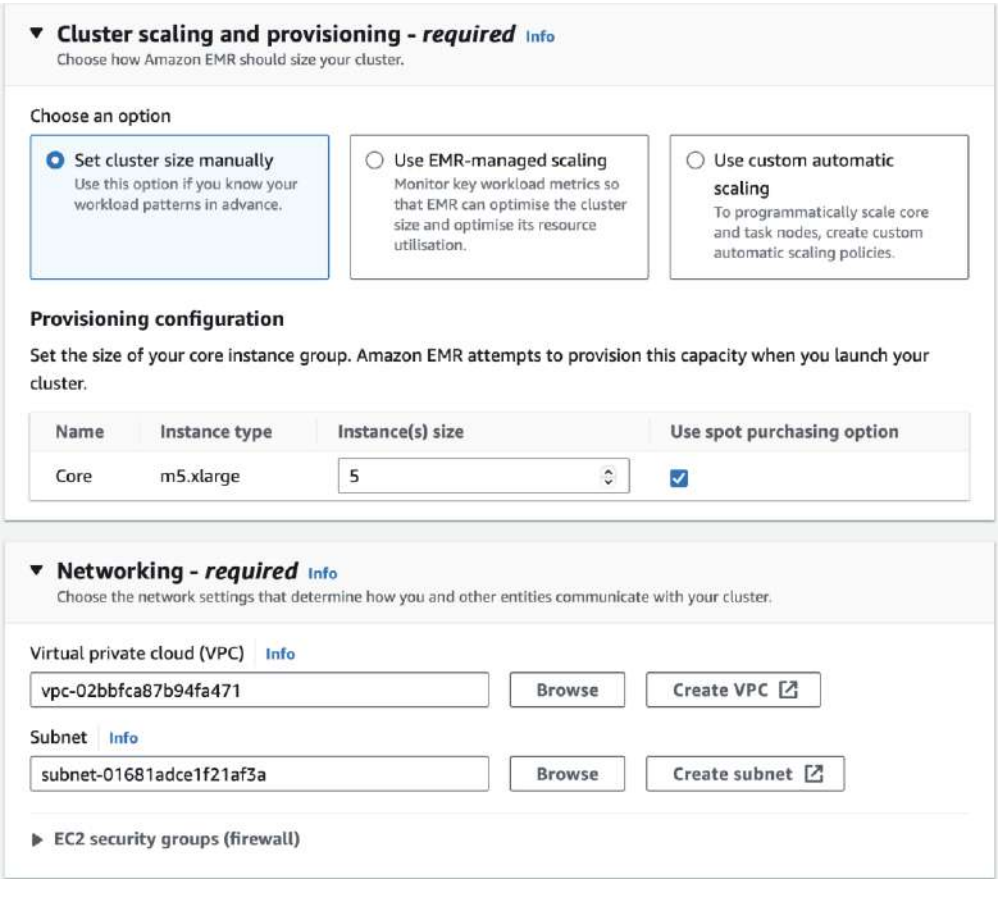


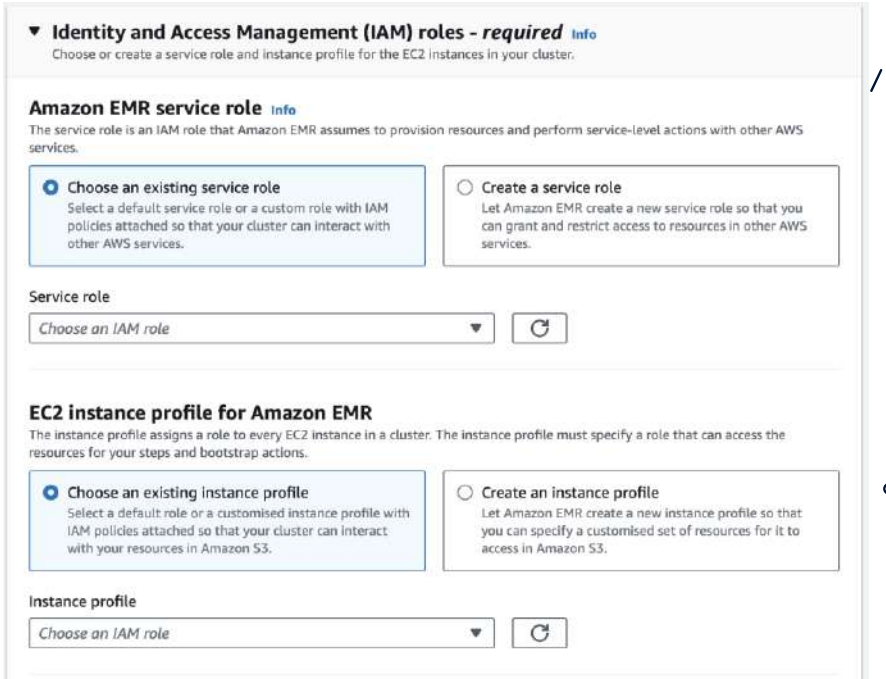
AWS EMR


Name and applications - required Info
Name your cluster and choose the applications that you want to install to your cluster.
Name
My cluster
Amazon EMR release Info
A release contains a set of applications which can be installed on your cluster.
emr-7.2.0
Application bundle
Spark Interactive
Core Hadoop
Flink
HBase
Presto
Trino
Custom
AmazonCloudWatchAgent 1.300032.2
Flink 1.18.1
HBase 2.4.17
HCatalog 3.1.3
Hadoop 3.3.6
Hive 3.1.3
Hue 4.11.0
JupyterEnterpriseGateway 2.6.0
JupyterHub 1.5.0
Livy 0.8.0
MXNet 1.9.1
Oozie 5.2.1
Phoenix 5.1.3
Pig 0.17.0
Presto 0.285
Spark 3.5.1
Sqoop 1.4.7
TensorFlow 2.11.0
Tez 0.10.2
Trino 436
Zeppelin 0.10.1
ZooKeeper 3.9.1
AWS Glue Data Catalogue settings
Use the AWS Glue Data Catalog to provide an external metastore for your application.
Use for Hive table metadata
Use for Spark table metadata
Operating system options Info
Amazon Linux release
Customised Amazon Machine Image (AMI)
Automatically apply the latest Amazon Linux updates


Cluster configuration - required Info
Choose a configuration method for the primary, core and task node groups for your cluster.
Instance groups
Choose one instance type per node group
Instance fleets
Choose any combination of instance types within each node group
Instance groups
Primary
Choose EC2 instance type
m5.xlarge
4 vCore 16 GiB memory
EBS only storage
On-demand price: USD 0.204 per instanc...
Lowest spot price: $0.064 (eu-north-1b)
Actions
Use high availability
Launch highly available, more resilient cluster with three primary nodes on On-Demand instances. This configuration applies for the lifetime of your cluster. Find out more
Node configuration - optional
Core
Remove instance group
Choose EC2 instance type
m5.xlarge
4 vCore 16 GiB memory
EBS only storage
On-demand price: USD 0.204 per instanc...
Lowest spot price: $0.064 (eu-north-1b)
Actions
Node configuration - optional
Add task instance group
You can add up to 48 more task instance groups.

AWS EMR


Cluster scaling and provisioning - required Info
Choose how Amazon EMR should size your cluster.
Choose an option
Set cluster size manually
Use this option if you know your workload patterns in advance.
Use EMR-managed scaling
Monitor key workload metrics so that EMR can optimise the cluster size and optimise its resource utilisation.
Use custom automatic scaling
To programmatically scale core and task nodes, create custom automatic scaling policies.
Provisioning configuration
Set the size of your core instance group. Amazon EMR attempts to provision this capacity when you launch your cluster.
Name
Instance type
Instance(s) size
Use spot purchasing option
Core
m5.xlarge
5
Networking - required Info
Choose the network settings that determine how you and other entities communicate with your cluster.
Virtual private cloud (VPC) Info
vpc-02bbfca87b94fa471
Browse
Create VPC
Subnet Info
subnet-01681adce1f21af3a
Browse
Create subnet
EC2 security groups (firewall)


Identity and Access Management (IAM) roles - required Info
Choose or create a service role and instance profile for the EC2 instances in your cluster.
Amazon EMR service role Info
The service role is an IAM role that Amazon EMR assumes to provision resources and perform service-level actions with other AWS services.
Choose an existing service role
Select a default service role or a custom role with IAM policies attached so that your cluster can interact with other AWS services.
Create a service role
Let Amazon EMR create a new service role so that you can grant and restrict access to resources in other AWS services.
Service role
Choose an IAM role
EC2 instance profile for Amazon EMR
The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.
Choose an existing instance profile
Select a default role or a customised instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.
Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a customised set of resources for it to access in Amazon S3.
Instance profile
Choose an IAM role

Дякую за увагу!

RESOURCES

- https://spark.apache.org/
- https://databricks.com/
- https://spark.apache.org/docs/latest/api/python/index.html
- https://www.wisewithdata.com/2020/05/rdds-vs-dataframes-vs-datasets-the-three-data-structures-of-spark/
- https://data.cityofnewyork.us/City-Government/Parking-Violations-Issued-Fiscal-Year-2022/pvqr-7yc4
- https://www.altexsoft.com/blog/hadoop-pros-cons/
- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-gs.html
- https://learn.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-provision-linux-clusters