

Setting up Spark Environment



Mohit Batra

Founder, Crystal Talks

linkedin.com/in/mohitbatra

Overview



Understand Spark environments

- Development options, cluster managers, execution modes etc.

Install Spark and configure environment

Monitor Spark applications

Development options

- Command line / Shark shell
- Jupyter notebooks
- PyCharm IDE
- Spark-submit command

Setup multi-node cluster

Understanding Spark Environments

Spark Development Options

Command line / Spark Shell

Notebooks

- Jupyter, Zeppelin
- Can run locally or in cloud platforms like Azure HDInsight & Google Dataproc

IDEs to build projects

- IntelliJ, PyCharm, VS Code
- Run code locally, or build & run in Spark cluster

Spark Submit command to run jobs

Cloud platform notebooks

- Offered by Databricks, Azure Synapse Analytics etc.

Cluster Managers

Manages & allocates resources to applications on the cluster

Local

Standalone

YARN

Mesos

Kubernetes

Application Execution Modes

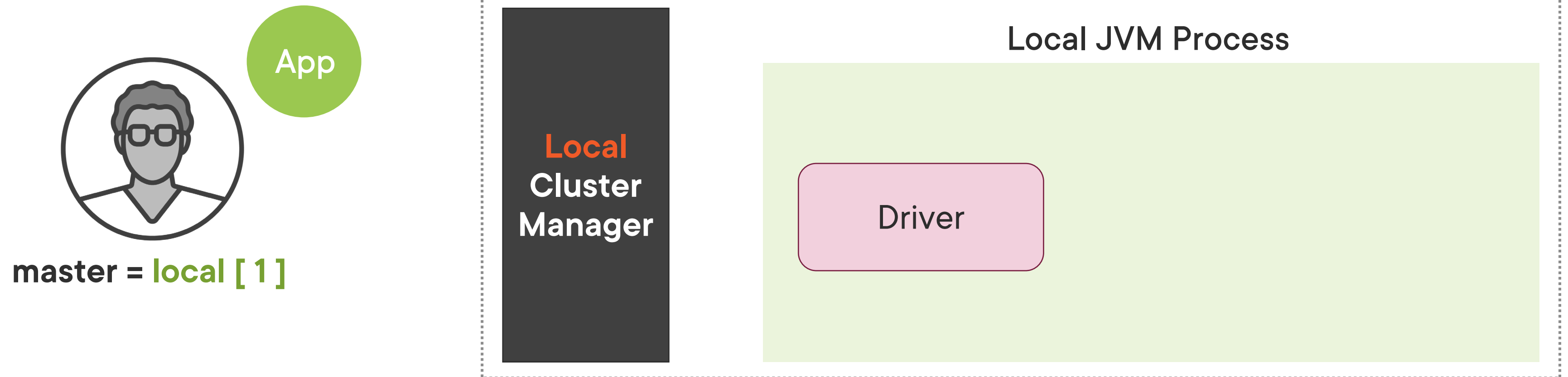


Local Mode

Client Mode

Cluster Mode

Local Execution Mode

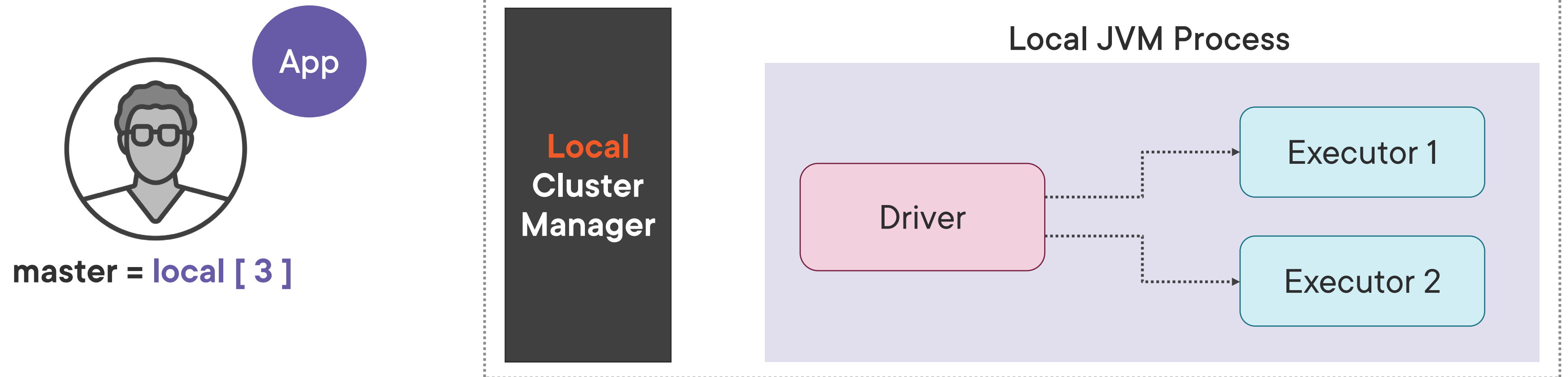


Single thread is used

Driver runs all the jobs

No executors are created

Local Execution Mode



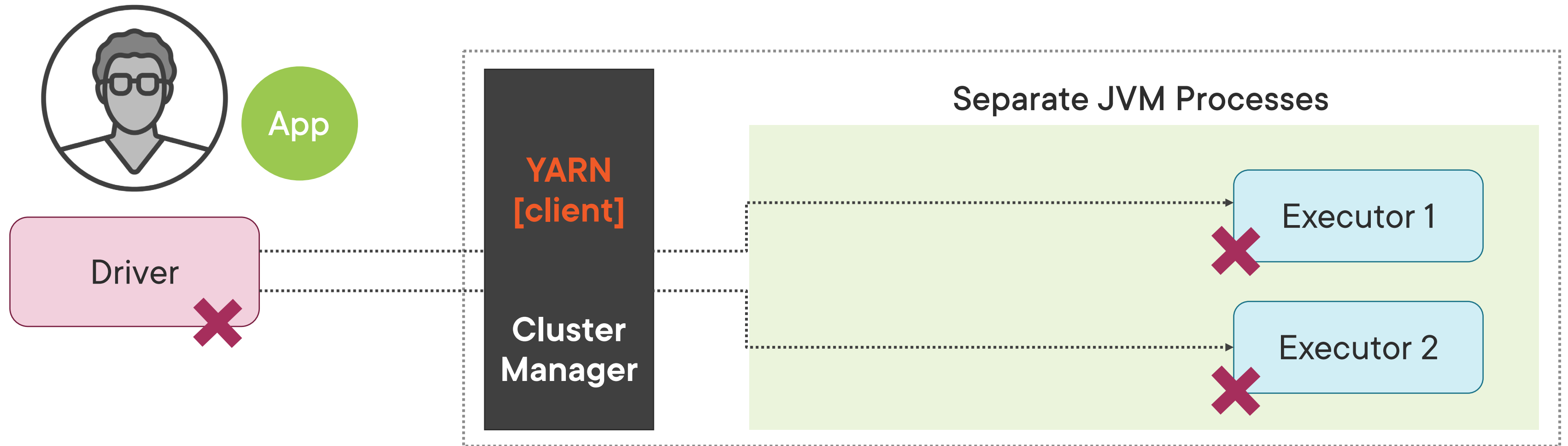
Three threads are created in same JVM

Executors run tasks

Simulation of production cluster environment

| Execution Mode | How it works? | Cluster Managers | Development Tools | Purpose |
|----------------|--------------------------------|------------------|----------------------------------|-----------------------------|
| Local | Driver & executors on same JVM | Local | Spark Shell Notebooks IDEs | Run locally for dev/testing |

Client Execution Mode

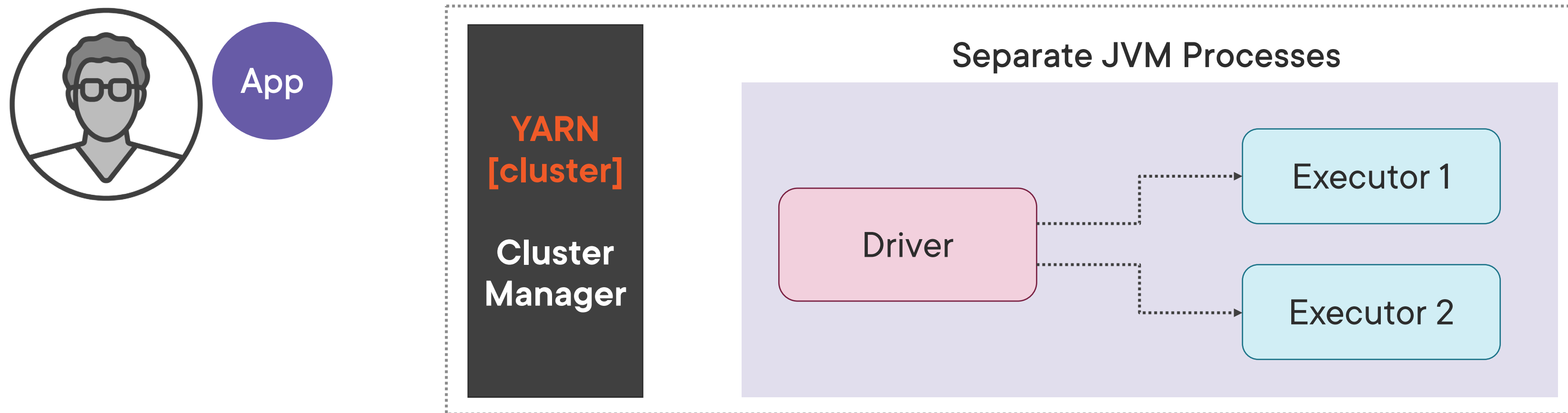


Driver runs on client machine

Each executor runs as a separate JVM process in cluster

| Execution Mode | How it works? | Cluster Managers | Development Tools | Purpose |
|----------------|---|-----------------------------|--|-------------------------------------|
| Local | Driver & executors on same JVM | Local | Spark Shell Notebooks IDEs | Run locally for dev/testing |
| Client | Driver on client, Executors on cluster | YARN [client] Standalone | Spark Shell Notebooks Spark-submit | Interactive querying, Data analysis |

Cluster Execution Mode



Driver & executors run on cluster

Each process consumes resources on cluster

Used to run production workloads

| Execution Mode | How it works? | Cluster Managers | Development Tools | Purpose |
|----------------|--|---|--|-------------------------------------|
| Local | Driver & executors on same JVM | Local | Spark Shell Notebooks IDEs | Run locally for dev/testing |
| Client | Driver on client, Executors on cluster | YARN [client] Standalone | Spark Shell Notebooks Spark-submit | Interactive querying, Data analysis |
| Cluster | Driver & executors in separate JVMs on cluster | YARN [cluster] Standalone Kubernetes Mesos | Spark-submit Cloud notebooks | Long running jobs in production |

Installing Spark

Spark 3+ Minimum Requirements

Windows, Linux, MacOS

Java v8 (8u92+) / v11 / v17

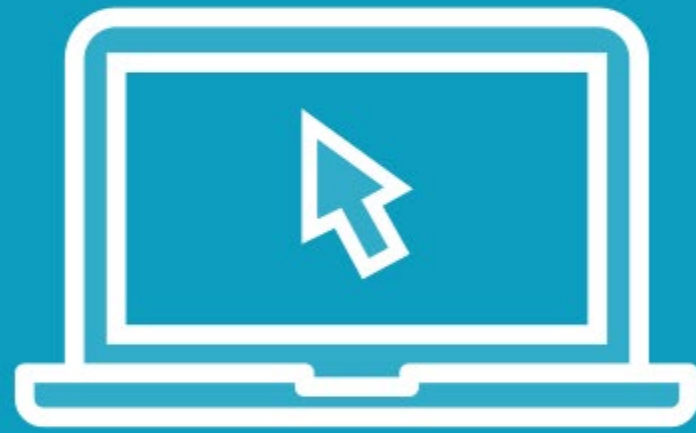
Scala 2.12 and above

Python 3.6 and above

Unzip utility like 7-Zip

Winutils.exe *(for Windows only)*

Demo



Setup Spark on a Windows machine

Installation on macOS is similar





- Check Setup document in Exercise Files

Steps

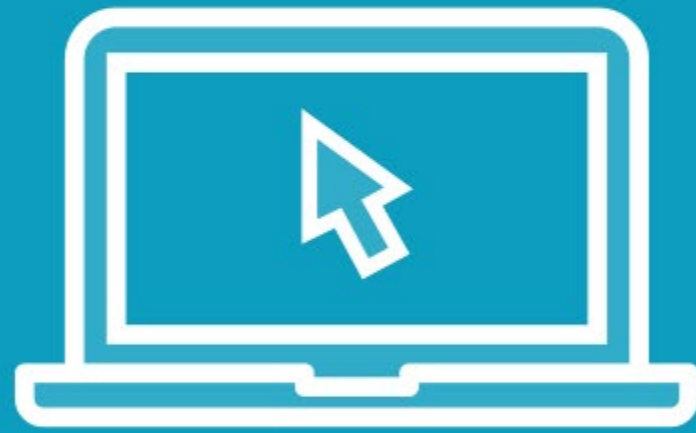
1. Download & install JDK v17
2. Download Apache Spark 3.3+ (with Hadoop)
3. Download winutils.exe (Windows only)
4. Set environment variables
 - a. Java_Home, Spark_Home, Hadoop_Home, Path
5. Run Spark Shell to verify

Monitoring Spark with Web UI

Option 1: Running Spark in Command Line

| Execution Mode | How it works? | Cluster Managers | Development Tools | Purpose |
|--|--|--|--|-------------------------------------|
| Local  | Driver & executors on same JVM | Local  | <div>Spark Shell</div> Notebooks IDEs | Run locally for dev/testing |
| Client  | Driver on client, Executors on cluster | YARN [client] Standalone  | <div>Spark Shell</div> Notebooks Spark-submit | Interactive querying, Data analysis |
| Cluster | Driver & executors in separate JVMs on cluster | YARN [cluster] Standalone Kubernetes Mesos | Spark-submit Cloud notebooks | Long running jobs in production |

Demo



Install Python on a Windows machine

Installation on macOS is similar

- Check Setup document in Exercise Files





Multiple ways to install Python

- Using python.org (*in this clip*)
- Using Anaconda distribution (*in next clip*)

Steps

1. Download & install Python 3.7 and above
2. Set environment variables
 - a. Path, Pyspark_Python, Pyspark_Driver_Python, PythonPath
3. Run PySpark Shell to verify

Option 2: Running Spark with Jupyter Notebooks

| Execution Mode | How it works? | Cluster Managers | Development Tools | Purpose |
|--|--|--|---|-------------------------------------|
| Local  | Driver & executors on same JVM | Local  | Spark Shell <div>Notebooks</div> IDEs | Run locally for dev/testing |
| Client  | Driver on client, Executors on cluster | YARN [client] Standalone  | Spark Shell <div>Notebooks</div> Spark-submit | Interactive querying, Data analysis |
| Cluster | Driver & executors in separate JVMs on cluster | YARN [cluster] Standalone Kubernetes Mesos | Spark-submit Cloud notebooks | Long running jobs in production |

Demo



Install Anaconda distribution

- Comes along with Python and Jupyter notebooks



Installation on macOS is similar

- Check Setup document in Exercise Files

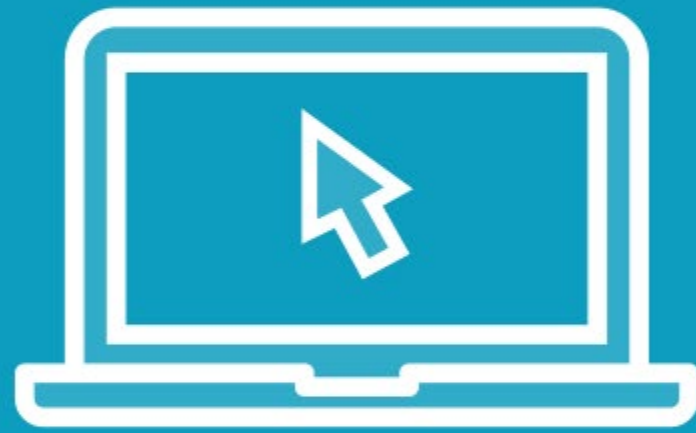
Steps

1. Download & install Anaconda distribution
2. Setup conda environment
 - a. Add Python, FindSpark, Hive, Delta Lake
 - b. Add Jupyter notebooks
3. Launch a Jupyter notebook

Option 3: Creating Project with PyCharm IDE

| Execution Mode | How it works? | Cluster Managers | Development Tools | Purpose |
|---|--|---|---|-------------------------------------|
| Local  | Driver & executors on same JVM | Local  | Spark Shell Notebooks <div>IDEs</div> | Run locally for dev/testing |
| Client | Driver on client, Executors on cluster | YARN [client] Standalone | Spark Shell Notebooks Spark-submit | Interactive querying, Data analysis |
| Cluster | Driver & executors in separate JVMs on cluster | YARN [cluster] Standalone Kubernetes Mesos | Spark-submit Cloud notebooks | Long running jobs in production |

Demo



Prerequisites

- PyCharm Community Edition
(<https://www.jetbrains.com/pycharm/download/>)
- Complete steps in clip – Option 1: Running Spark in Command Line

Steps

1. Create new project in PyCharm
2. Add pyspark package dependency
3. Write code and run the project

Option 4: Running Jobs with Spark Submit

Spark-Submit



SUBMIT

Utility to submit a Spark Application / Job to a cluster

Code can be in any language

- Scala, Python or Java

Can be submitted to any supported Cluster Manager

- Local, Standalone, YARN, Mesos, Kubernetes

Provide configuration and dependencies

Both Spark Shell & Spark-Submit are command line

- Spark Shell is typically used for interactive querying
- Spark-Submit is typically used to submit long running jobs

| Execution Mode | How it works? | Cluster Managers | Development Tools | Purpose |
|----------------|--|---|---|-------------------------------------|
| Local | Driver & executors on same JVM | Local | Spark Shell Notebooks IDEs | Run locally for dev/testing |
| Client | Driver on client, Executors on cluster | YARN [client] Standalone | Spark Shell Notebooks Spark-submit | Interactive querying, Data analysis |
| Cluster | Driver & executors in separate JVMs on cluster | YARN [cluster] Standalone Kubernetes Mesos | Spark-submit Cloud notebooks | Long running jobs in production |

Setting Up Multi-Node Cluster

Demo



Prerequisites

- Clip – Option 1: Running Spark in Command Line
- Clip – Option 2: Running Spark with Jupyter
- Same Python version installed in both clips

1. Setup Master Node

- Generate master URL

2. Setup 2 Worker Nodes

- Use master URL to connect to master node

3. Launch applications on the cluster

- Launch application using Jupyter
- Submit application using Spark-submit

Summary



Can work with Spark in various ways

- Several development options and cluster managers

Understood application execution modes

- Local mode – Single JVM to run driver & executors
- Client mode – Driver on client & executors on cluster
- Cluster mode – Driver & executors run on cluster

Spark can be installed on local machine or on cluster

Development Options

- Command Line can be used for dev/testing
- Jupyter for interactive analysis or development
- PyCharm IDE for project development
- Spark-submit to run long running production jobs

Web UI can be used for monitoring

Up Next:
Working with RDDs –
Resilient Distributed Datasets
