APACHE
Spark
TM

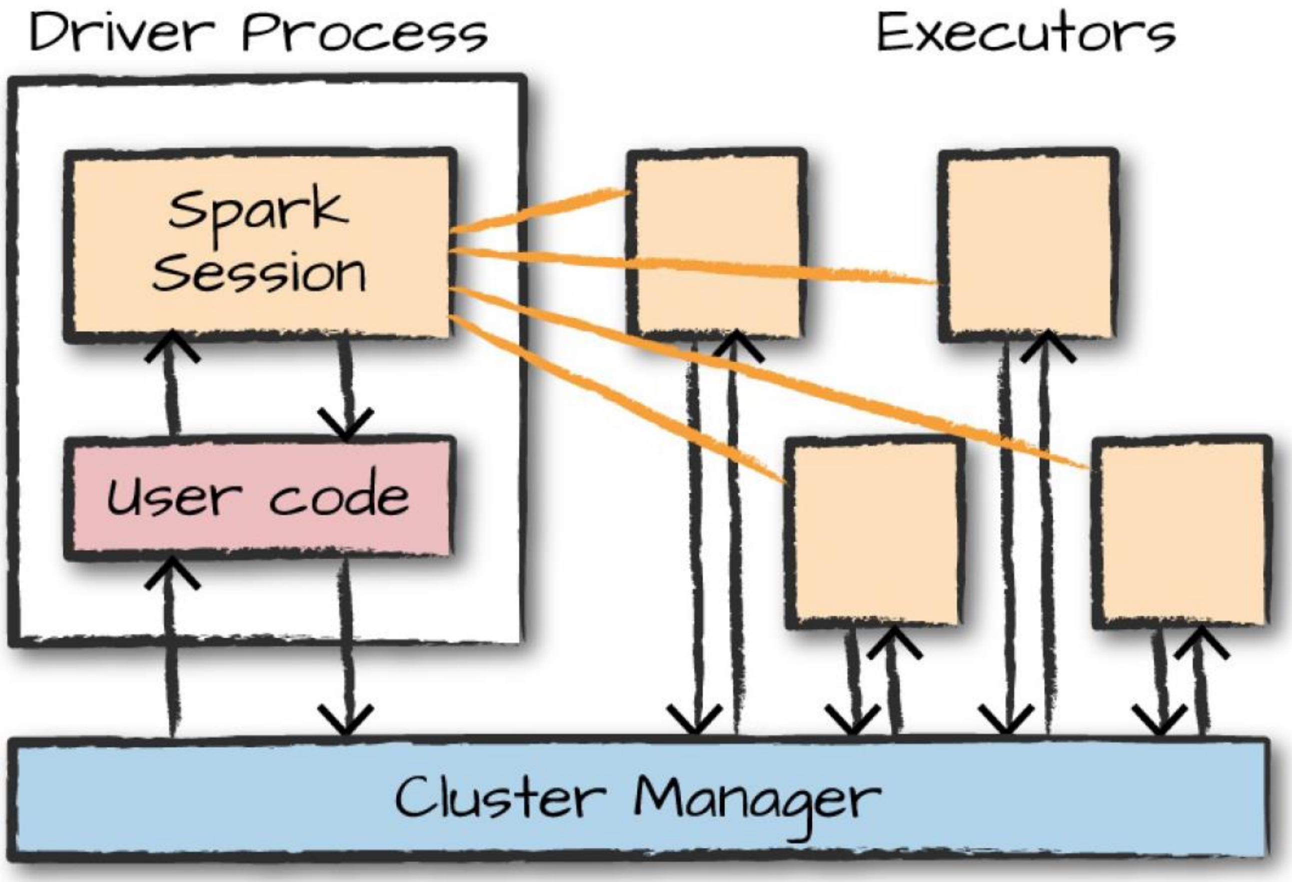
Introduction

- Apache Spark is a unified computing engine and a set of libraries for parallel data processing.
- Distributed, Highly scalable, In-memory data analytics system.
- Why In-memory system is needed ?

Spark's Concepts

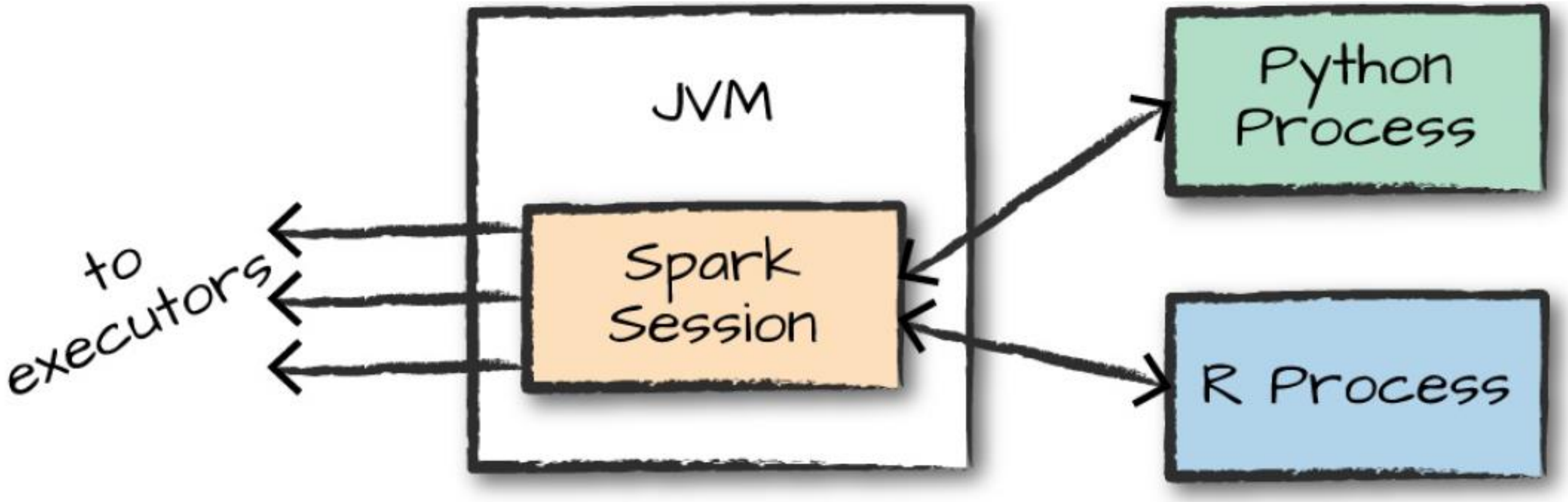
Spark Application –

High Level Architecture



Spark's Concepts

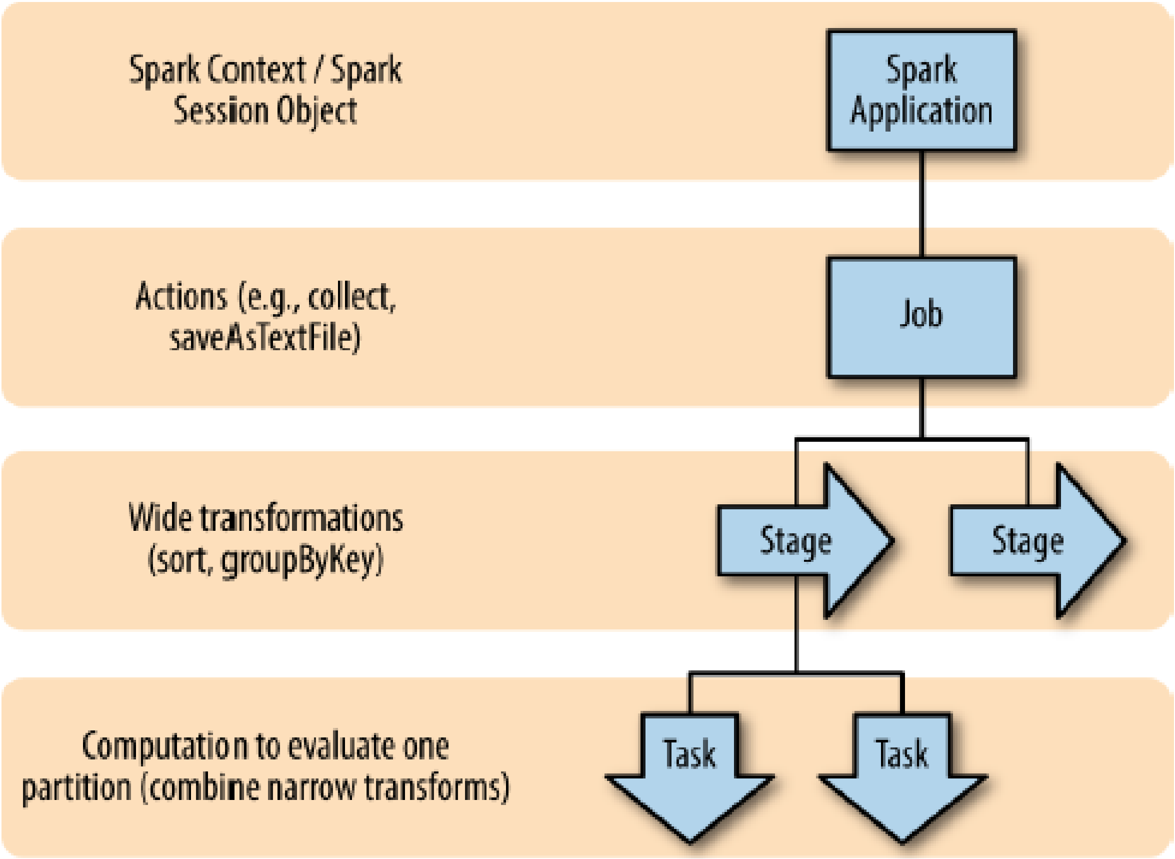
- Spark's Language API
  - Scala, Java, Python, SQL, R.



Spark's Concepts

- Spark Session – the driver process
- Data Frames
  - Most common structured API
- RDD (Resilient Distributed Datasets
- Partitions
- Transformations
  - Narrow transformation
  - Wide transformation
- Lazy Evaluation
- Actions

Anatomy of a Spark Job


Spark Context / Spark Session Object
Spark Application
Actions (e.g., collect, saveAsTextFile)
Job
Wide transformations (sort, groupByKey)
Stage
Stage
Computation to evaluate one partition (combine narrow transforms)
Task
Task

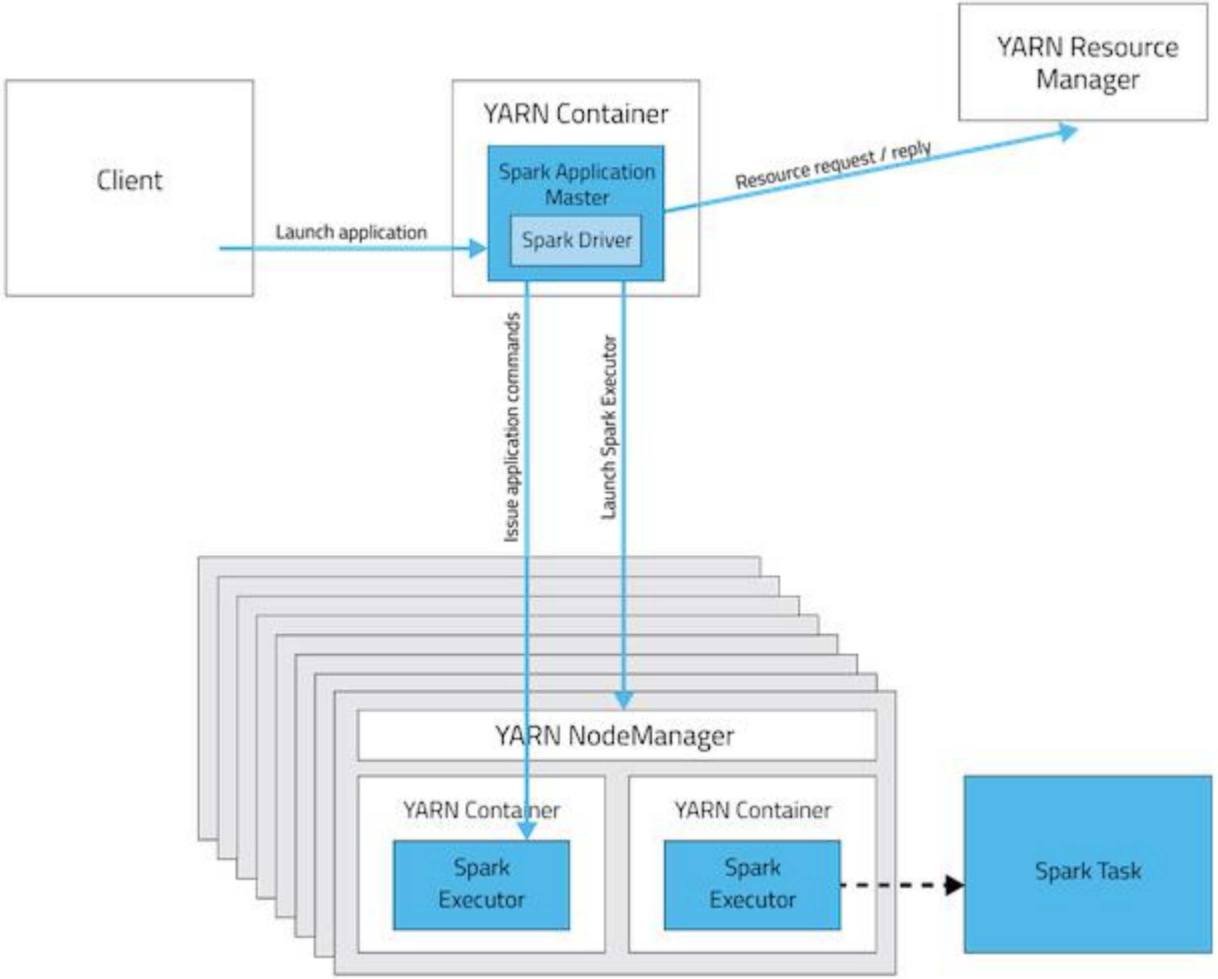
Cloudlytics

Anatomy of a Spark Job

- Execution details
  - Pipelining
  - Shuffle persistence

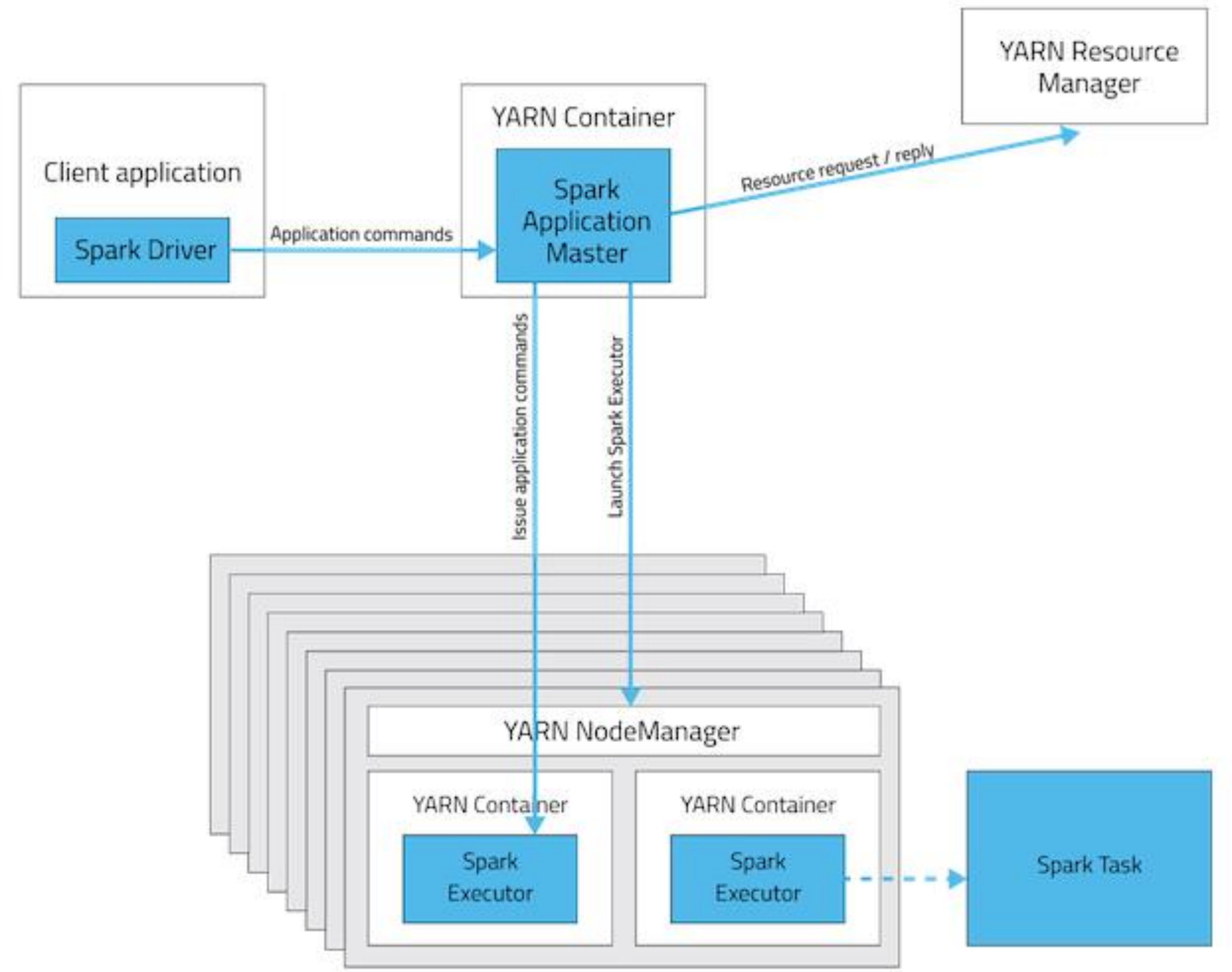
Spark on Yarn

- **Cluster Deployment Mode**



Spark on Yarn

- **Client Deployment Mode**



Configurations

- Dynamic Executor Allocation
  - Benefit
  - Limitation

Hive on Spark

	Memory	CPU
Hive on Spark	Minimum: 16 GB Recommended: 32 GB for larger data sizes Individual executor heaps should be no larger than 16 GB so machines with more RAM can use multiple executors.	Minimum: 4 cores Recommended: 8 cores for larger data sizes

Thank you

Cloudlytics