# Introduction

YARN is a resource manager which allocates resources (CPU, RAM) to run our processes. It can be used for example for running Spark jobs.

YARN runs on a cluster of nodes like Spark or HDFS.

# Containers

Containers are isolated chunks of CPU & memory allocated by YARN to run our processes.

# Components

We have the following processes (daemons, Java processes) running on nodes related to YARN:

* ResourceManager
* NodeManager
* ApplicationMaster

## ResourceManager

It is a global scheduler and resource allocator. It allocates containers on nodes which will be used by a process we want to run.

This process runs only on one node.

## NodeManager

Runs on each node, launches containers, monitors resources.

## ApplicationMaster

Every process we want to run with YARN (for example Spark jobs) gets its own ApplicationMaster.

It is responsible for:

* Requesting containers from the ResourceManager
* Requesting NodeManagers to launch containers

More information about how this works in case of Spark is in the ‘spark/spark on YARN notes’ document.

# Monitoring and logs

YARN provides a **web UI** (typically at http://<resourcemanager>:8088) to view:

* Spark applications
* Resource usage
* Logs for debugging

# High Availability

In the High Availability mode we have one active ResourceManager and multiple Standby ones.

When the active ResourceManager crashes, then one of the Standby ones takes over.