Cloudera Runtime 7.1.3

Apache Hadoop YARN Overview

Date published: 2020-02-11
Date modified: 2020-08-07

https://docs.cloudera.com/

Legal Notice

Contents

Introduction

Apache Hadoop YARN is the processing layer for managing distributed applications that run on multiple machines in a network.

YARN Features

YARN enables you to manage resources and schedule jobs in Hadoop.

YARN provides the following features:

Multi-tenancy

You can use multiple open-source and proprietary data access engines for batch, interactive, and real-time access to the same dataset. Multi-tenant data processing improves an enterprise's return on its Hadoop investments.

Cluster utilization

You can dynamically allocate cluster resources to improve resource utilization.

Multiple resource types

You can use multiple resource types such as memory, CPU, and GPU.

Scalability

Significantly improved data center processing power. YARN's ResourceManager focuses exclusively on scheduling and keeps pace as clusters expand to thousands of nodes managing petabytes of data.

Compatibility

MapReduce applications developed for Hadoop 1 runs on YARN without any disruption to existing processes. YARN maintains API compatability with the previous stable release of Hadoop.

Understanding YARN architecture

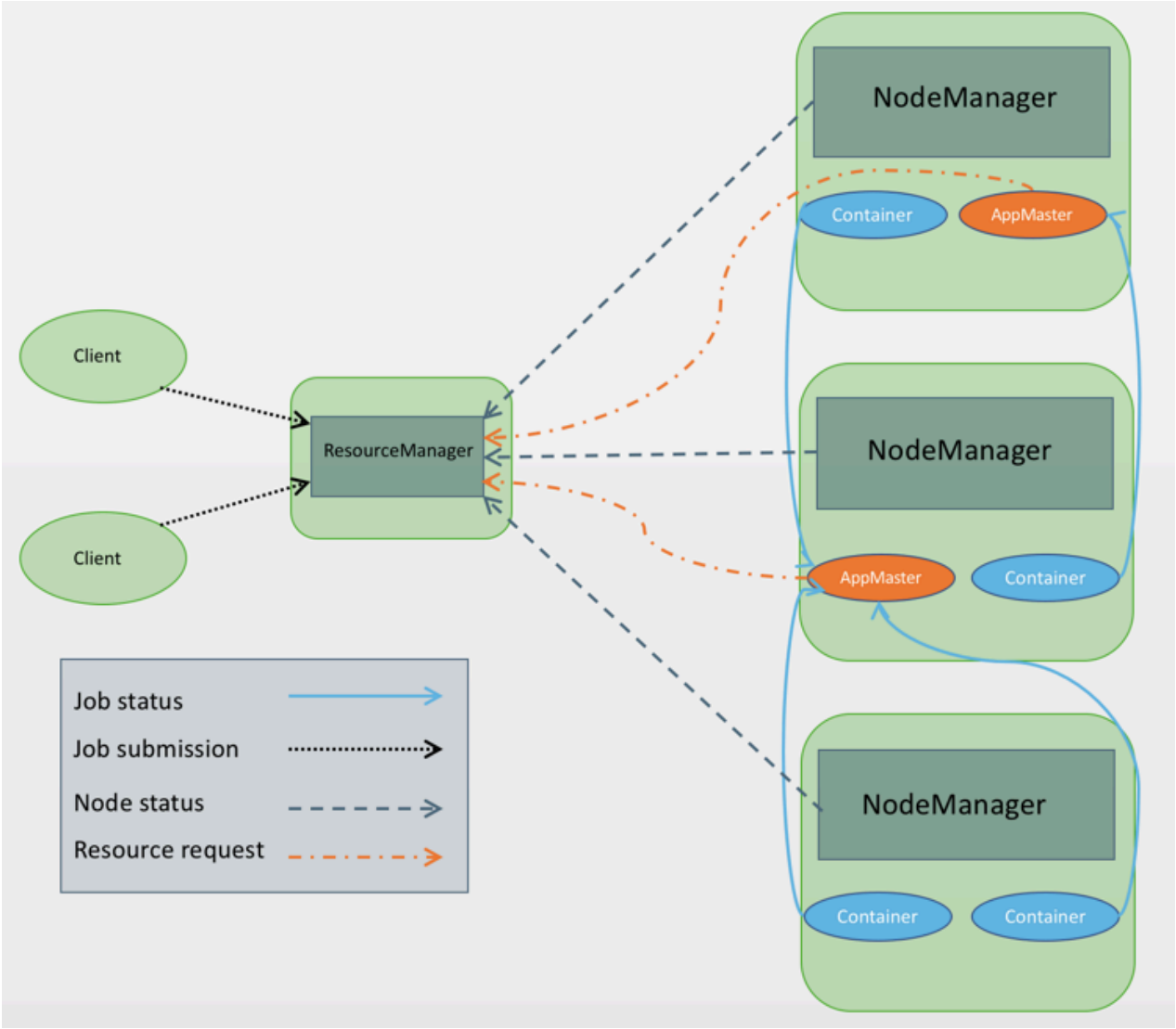
YARN allows you to use various data processing engines for batch, interactive, and real-time stream processing of data stored in HDFS or cloud storage like S3 and ADLS. You can use different processing frameworks for different use-cases, for example, you can run Hive for SQL applications, Spark for in-memory applications, and Storm for streaming applications, all on the same Hadoop cluster.

YARN extends the power of Hadoop to new technologies found within the data center so that you can take advantage of cost-effective linear-scale storage and processing. It provides independent software vendors and developers a consistent framework for writing data access applications that run in Hadoop.

YARN architecture and workflow

YARN has three main components:

- ResourceManager: Allocates cluster resources using a Scheduler and ApplicationManager.
- ApplicationMaster: Manages the life-cycle of a job by directing the NodeManager to create or destroy a container for a job. There is only one ApplicationMaster for a job.
- NodeManager: Manages jobs or workflow in a specific node by creating and destroying containers in a cluster node.


NodeManager
Container
AppMaster
Client
ResourceManager
NodeManager
Client
AppMaster
Container
Job status
Job submission
Node status
Resource request
NodeManager
Container
Container