# Apache Hadoop YARN Overview

**Date published:**
**Date modified:**

**https://docs.cloudera.com/**

# Legal Notice

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# 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

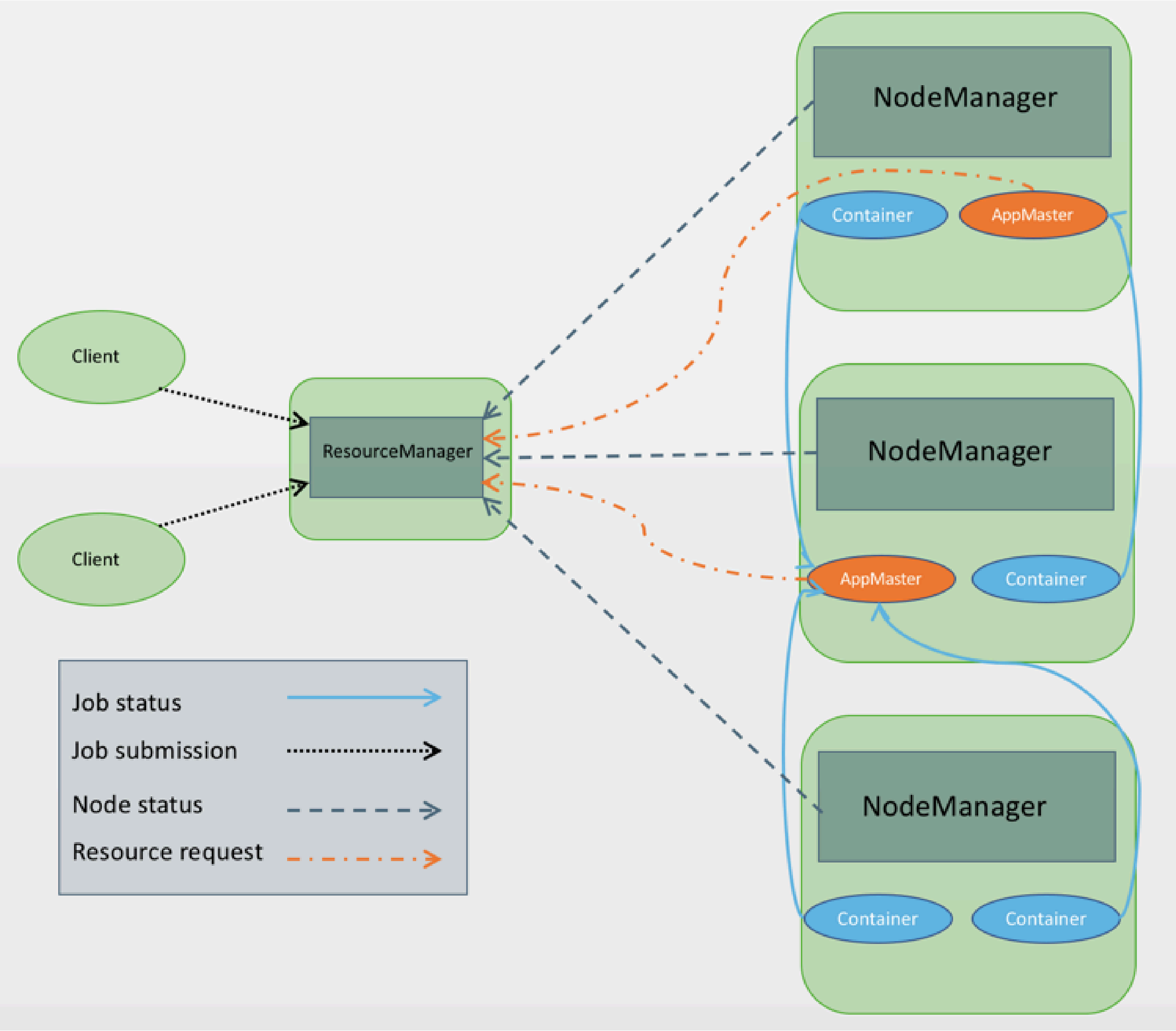
YARN allows you to use various data processing engines for batch, interactive, and real-time stream processing of data stored in HDFS or cloud storage like S3 and ADLS. You can use different processing frameworks for different use-cases, for example, you can run Hive for SQL applications, Spark for in-memory 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
Client
NodeManager
AppMaster
Container
Job status
Job submission
Node status
Resource request
NodeManager
Container
Container