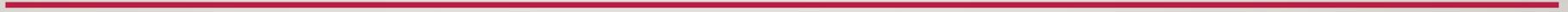
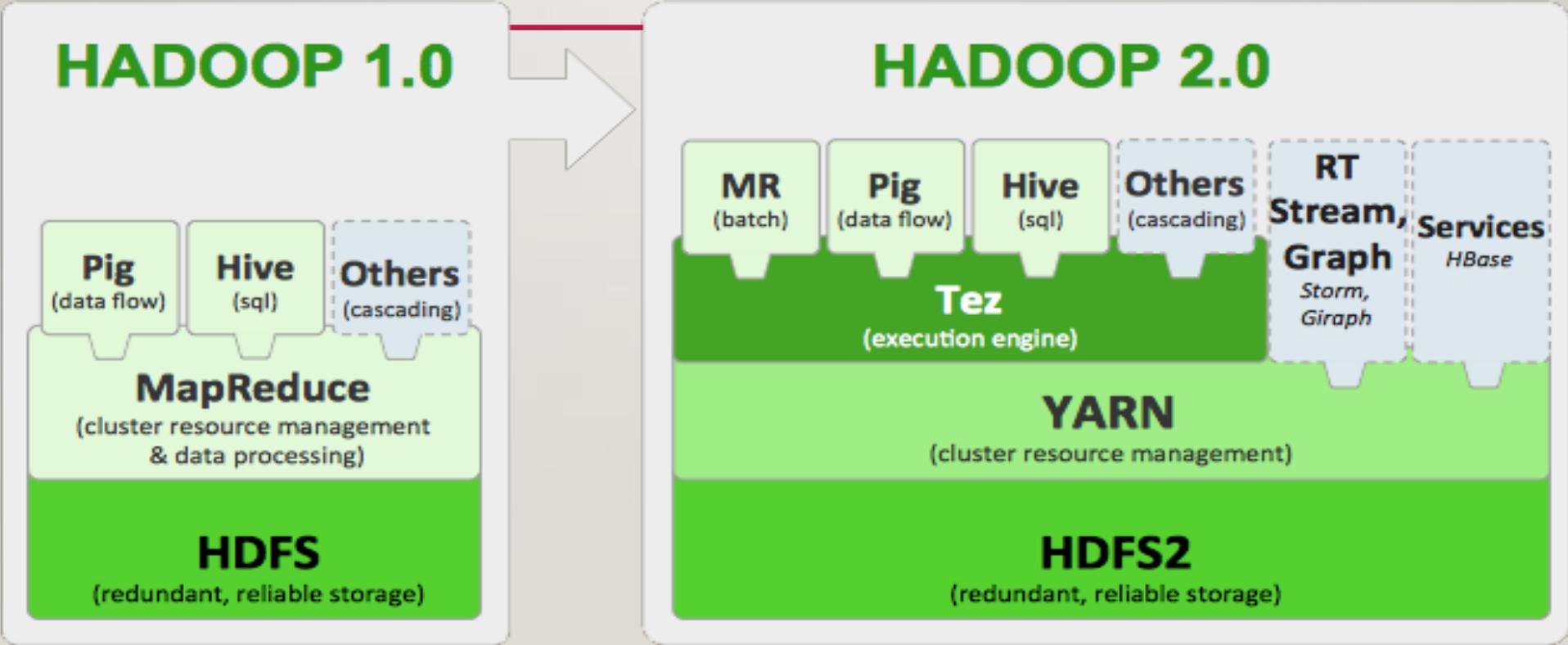


YARN



YARN(YET ANOTHER RESOURCE NEGOTIATOR)



HADOOP1.0

- 1. Data storage framework
- 2. Data processing framework

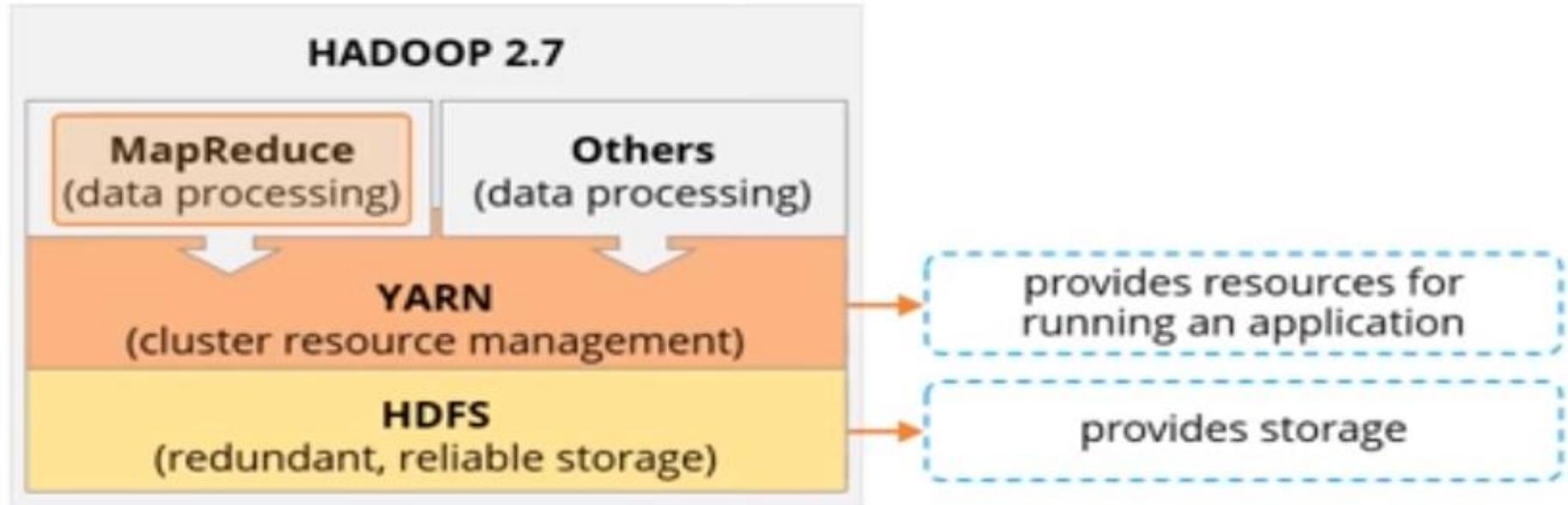
HADOOP 2.0

- HDFS continues to be the data storage framework.
- Resource management framework yarn added.

- Performs all your processing activities by allocating resources and scheduling tasks
- Two services: ResourceManager and NodeManager
- ResourceManager: Manages resources and schedule applications running on top of YARN
- NodeManager: Manages containers and monitors resource utilization in each container

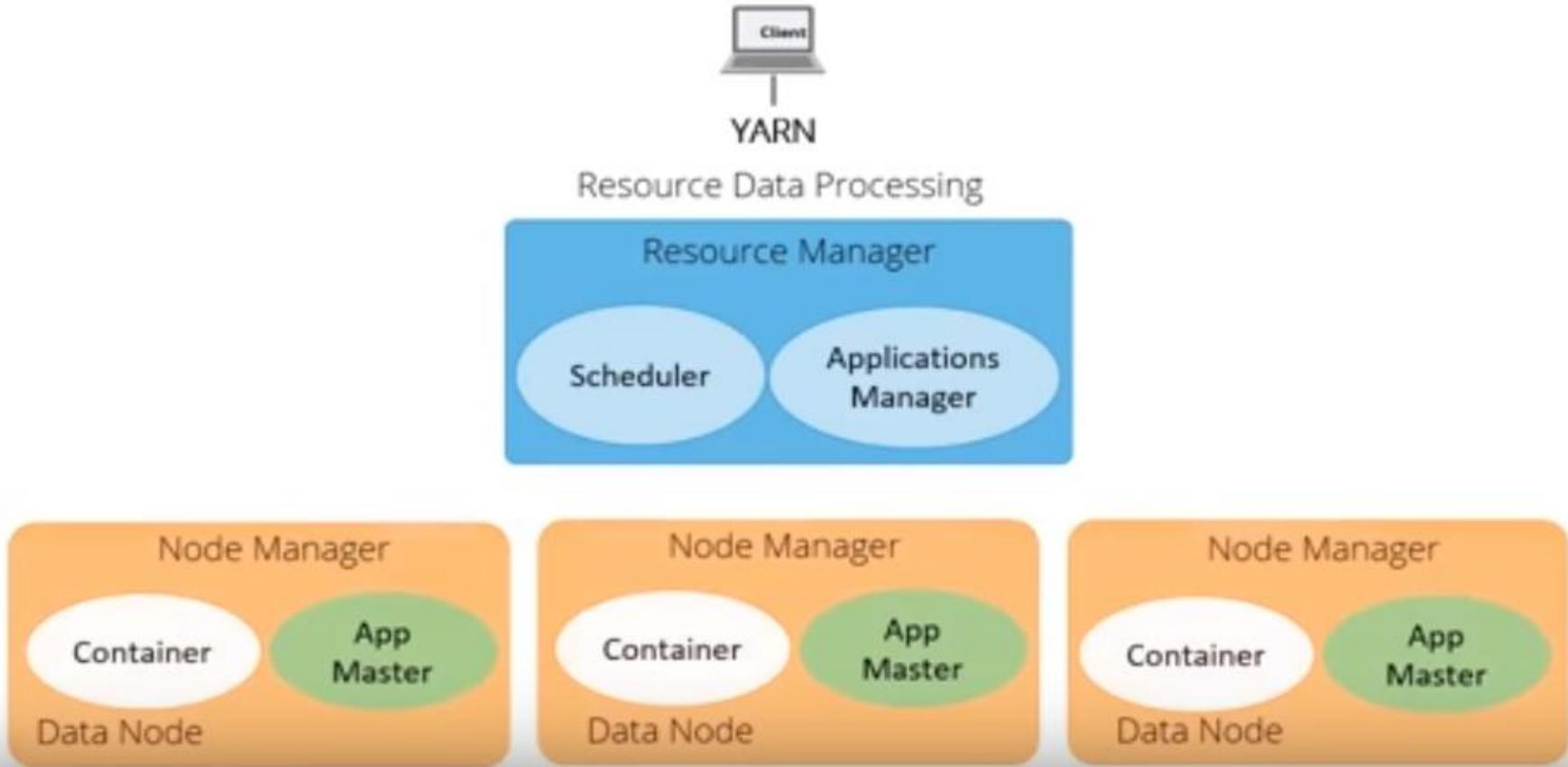
YARN INFRASTRUCTURE

The YARN Infrastructure is responsible for providing computational resources for application executions.



THREE ELEMENTS OF YARN ARCHITECTURE:-

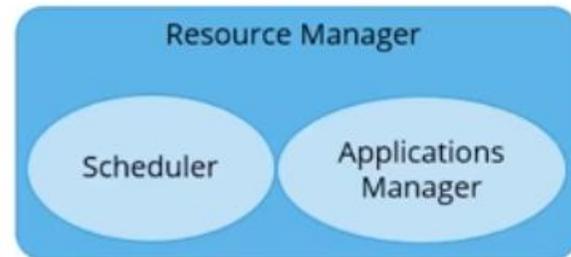
The three important elements of the YARN architecture are the Resource Manager, Application Master, and Node Manager.



-
- The resource manager(RM) is the master. It runs several services including the resource scheduler.
 - The application master negotiates resources for a single application. The application runs in the first container allotted to it.
 - Each application master requests resources from resource manager then works with containers provided by node managers.
 - Node manager(NM) is the slave ,when it starts it announces itself to the RM and offer resources to the cluster.
 - Each NM takes instructions from RM, reports and handles containers on single node.
 - Container is fraction of node manager capacity and is used by client to run programs

RESOURCE MANAGER:-

The RM mediates the available resources in the cluster among competing applications—to maximum cluster utilization.



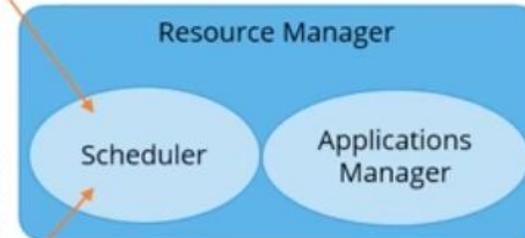
It includes a pluggable scheduler called the YarnScheduler, which allows different policies for managing constraints such as capacity, fairness, and Service Level Agreements.

RESOURCE MANAGER COMPONENT:-SCHEDULER

The Scheduler is responsible for allocating resources to various running applications.

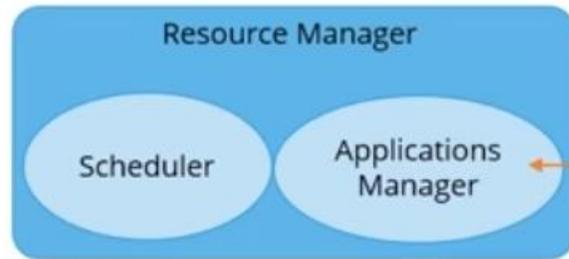
The Scheduler does not monitor or track the status of the application; nor does it restart failed tasks.

The Scheduler has a policy plug-in to partition cluster resources among various applications. Examples: CapacityScheduler, FairScheduler.



RESOURCE MANAGER COMPONENT:- APPLICATION MANAGER

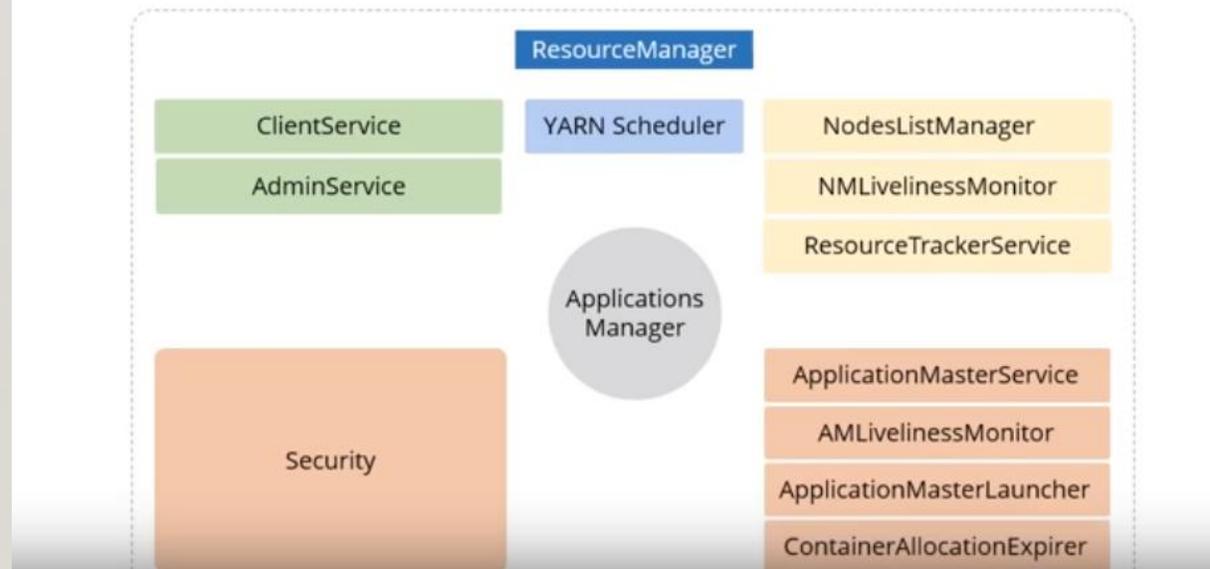
The ApplicationsManager is an interface which maintains a list of applications that have been submitted, currently running, or completed.



The ApplicationsManager accepts job submissions, negotiates the first container for executing the application, and restarts the ApplicationMaster container on failure.

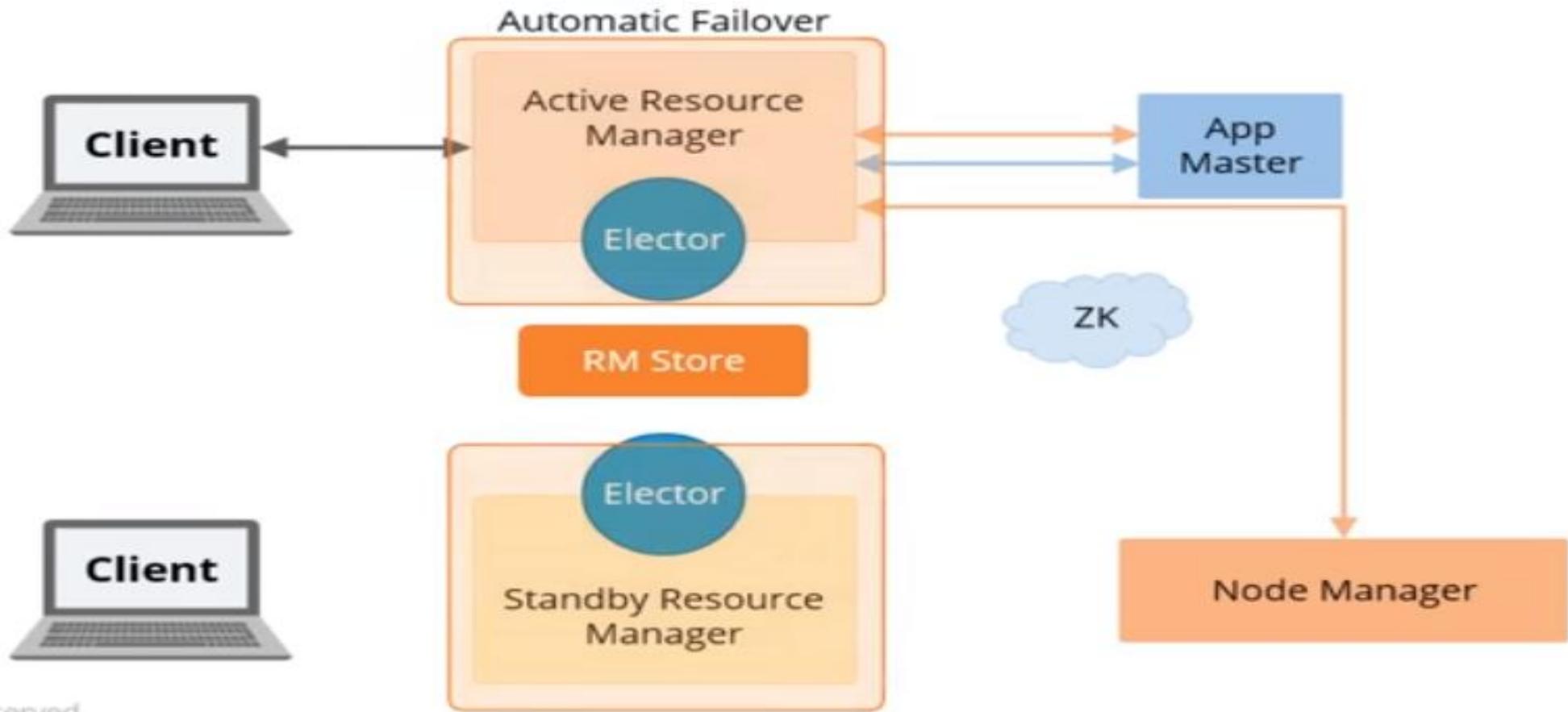
HOW A RESOURCE MANAGER OPERATES:-

The figure shown here displays all the internal components of the Resource Manager.



RESOURCE MANAGER IN HIGH AVAILABILITY MODE:-

- Before Hadoop 2.4 resource manager was single point of failure in yarn cluster.
- The high availability or HA feature an active/standby resource manager pair to remove this single point of failure.



YARN ARCHITECTURE ELEMENT:-APPLICATION MASTER

The ApplicationMaster in YARN is a framework-specific library, which negotiates resources from the RM and works with the NodeManager or Managers to execute and monitor containers and their resource consumption.



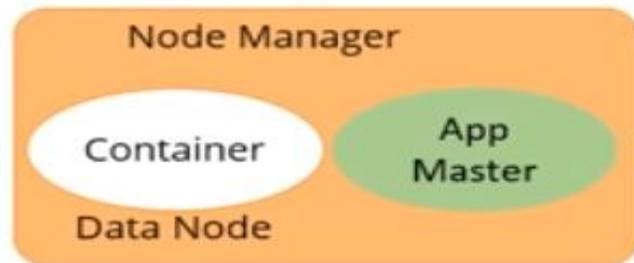
The ApplicationMaster:

- manages the application lifecycle
- makes dynamic adjustments to resource consumption
- manages execution flow
- manages faults
- provides status and metrics to the RM
- interacts with NodeManager and RM using extensible communication protocols
- Is not run as a trusted service

While every application has its own instance of an AppMaster, it is possible to implement an AppMaster for a set of applications as well.

YARN ARCHITECTURE ELEMENT:- NODE MANAGER

When a container is leased to an application, the NodeManager sets up the container's environment, including the resource constraints specified in the lease and any dependencies.

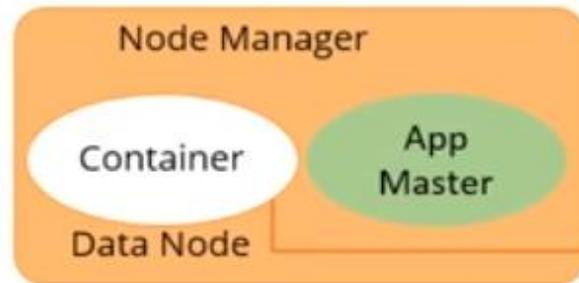


The NodeManager runs on each node and manages the following:

- Container lifecycle management
- Container dependencies
- Container leases
- Node and container resource usage
- Node health
- Log management
- Reporting node and container status to the RM

YARN CONTAINER:-

A YARN container is a result of a successful resource allocation, that is, the RM has granted an application a lease to use specified resources on a specific node.

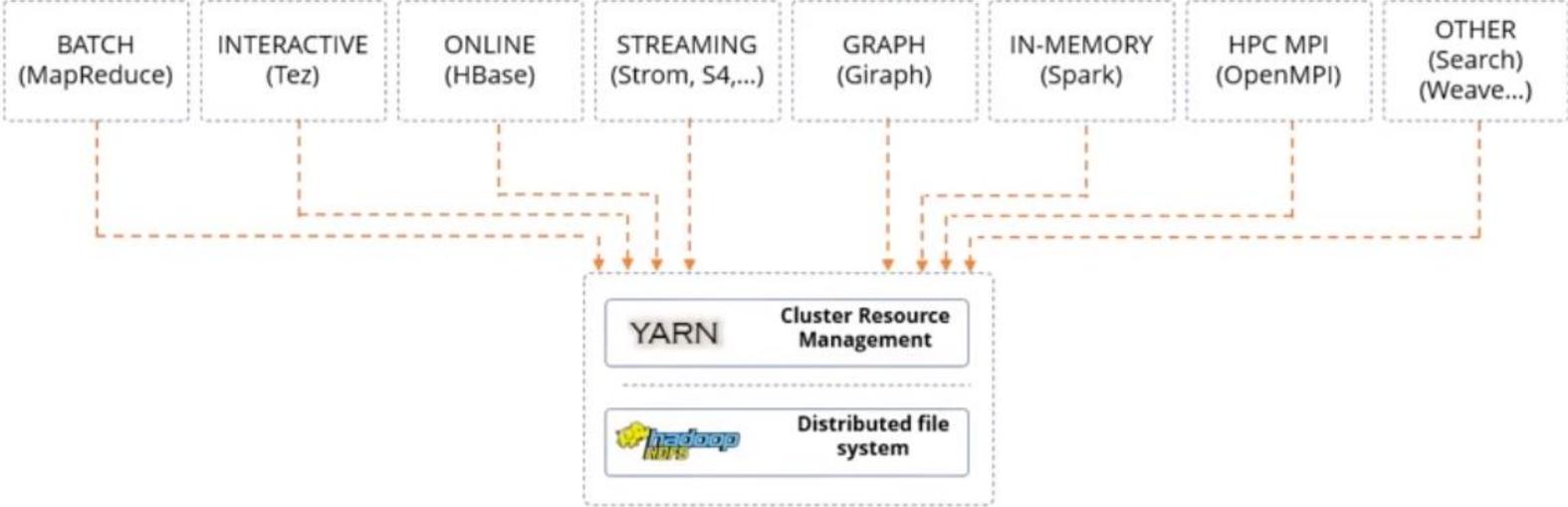


To launch the container, the ApplicationMaster must provide a container launch context (CLC) that includes the following information:

- Environment variables
- Dependencies, that is, local resources such as data files or shared objects needed prior to launch
- Security tokens
- The command necessary to create the process the application wants to launch

APPLICATIONS ON YARN

There can be many different workloads running on a Hadoop YARN cluster.



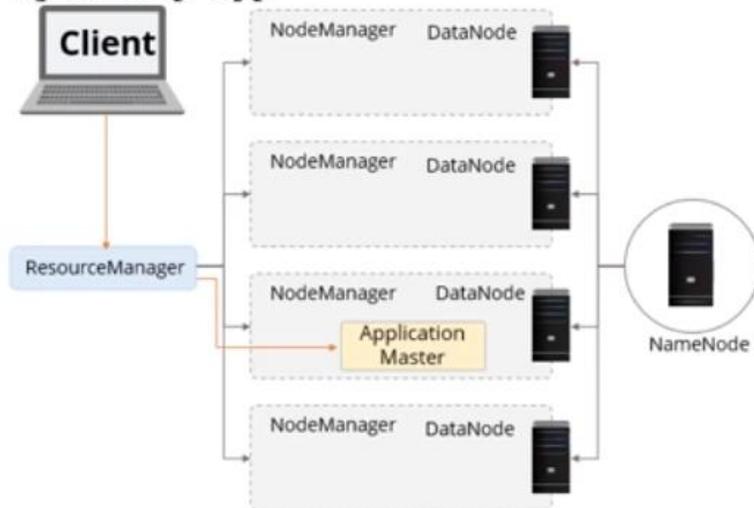
HOW YARN RUNS AN APPLICATION:-

- 1) The client submits an application to the resource manager.
- 2) The resource manager allocates a container.
- 3) The application master contacts the related node manager.
- 4) The node manager launches the container.
- 5) The container executes the application master.

STEP1 :-APPLICATION SUBMITTED TO RESOURCE MANAGER

Users submit applications to the ResourceManager by typing the hadoop jar command.

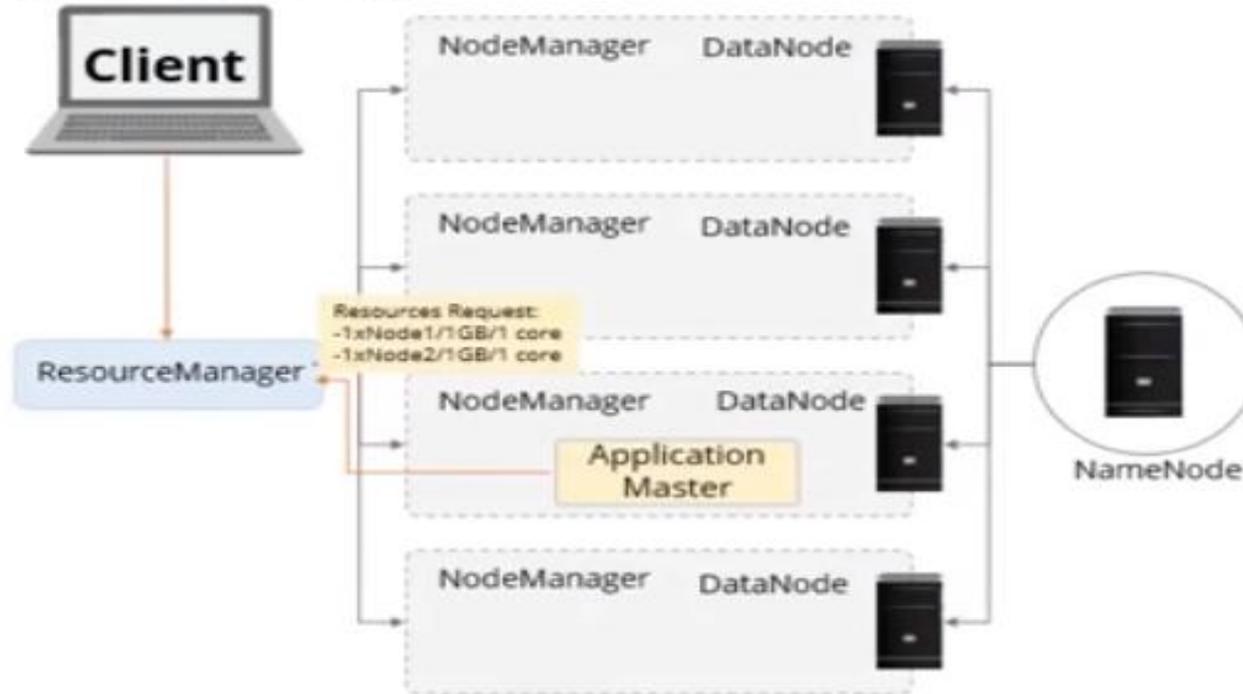
```
$ my-Hadoop-app
```



STEP 2:- RESOURCE MANAGER ALLOCATES A CONTAINER

When the ResourceManager accepts a new application submission, one of the first decisions the Scheduler makes is selecting a container.

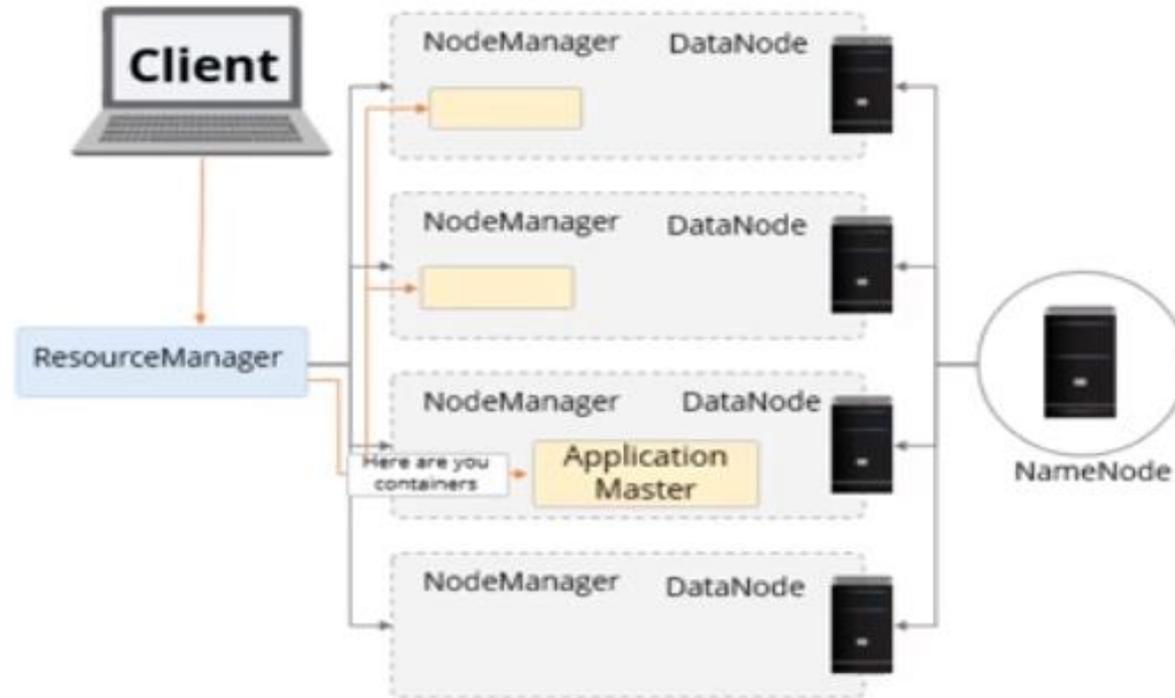
```
$ my-Hadoop-app
```



STEP3:- APPLICATION MASTER CONTACTS NODE MANAGER

After a container is allocated, the ApplicationMaster asks the NodeManager managing the host on which the container was allocated to use these resources to launch an application-specific task.

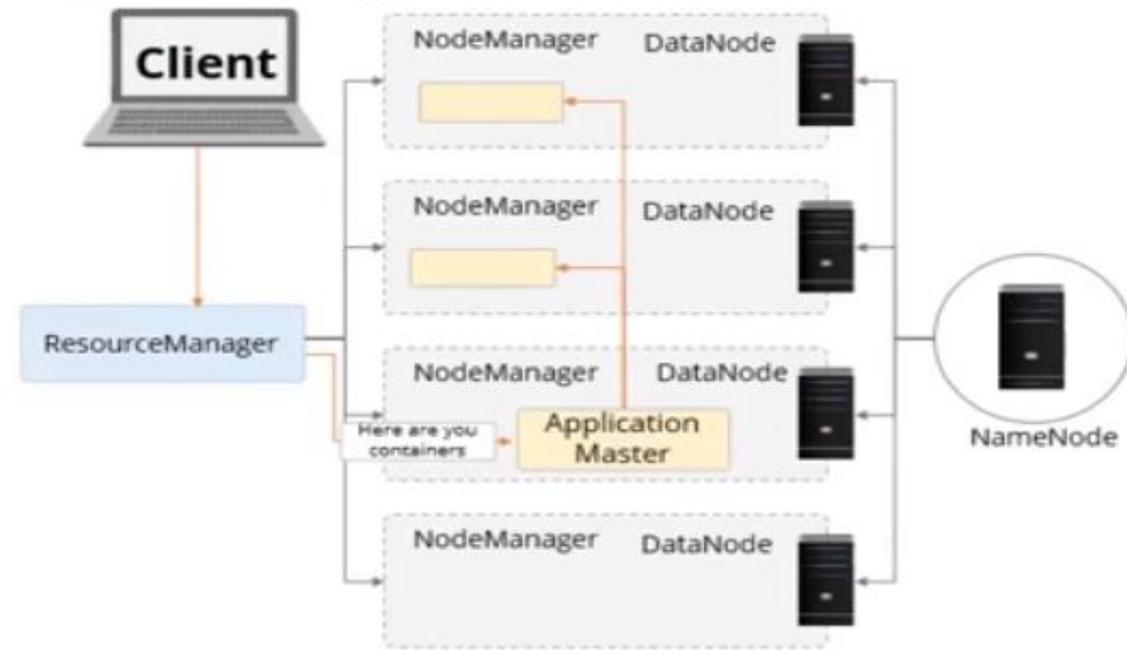
```
$ my-Hadoop-app
```



STEP 4:- RESOURCE MANAGER LAUNCHES A CONTAINER

The NodeManager does not monitor tasks; it only monitors the resource usage in the containers.

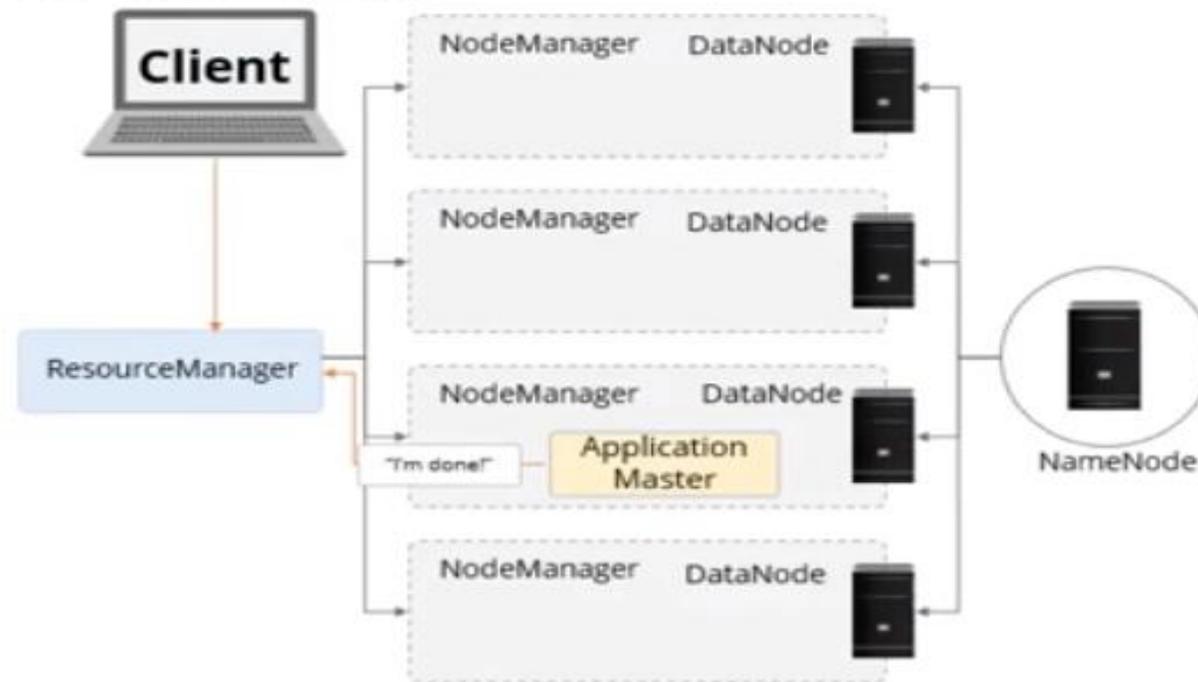
```
$ my-Hadoop-app
```



STEP5:- CONTAINER EXECUTES THE APPLICATION MASTER

After the application is complete, the ApplicationMaster shuts itself and releases its own container.

```
$ my-Hadoop-app
```



WORKFLOW:-

1. Client submits an application
2. RM allocates a container to start AM
3. AM registers with RM
4. AM asks containers from RM
5. AM notifies NM to launch containers
6. Application code is executed in container
7. Client contacts RM/AM to monitor application's status
8. AM unregisters with RM

