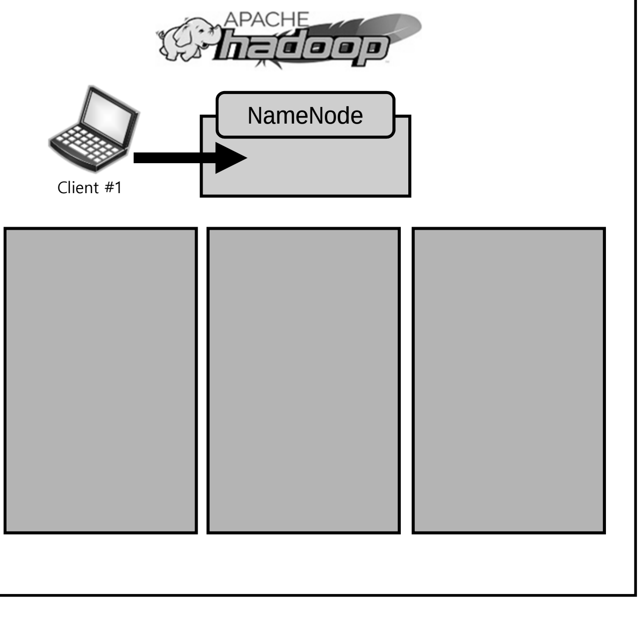
Big Data

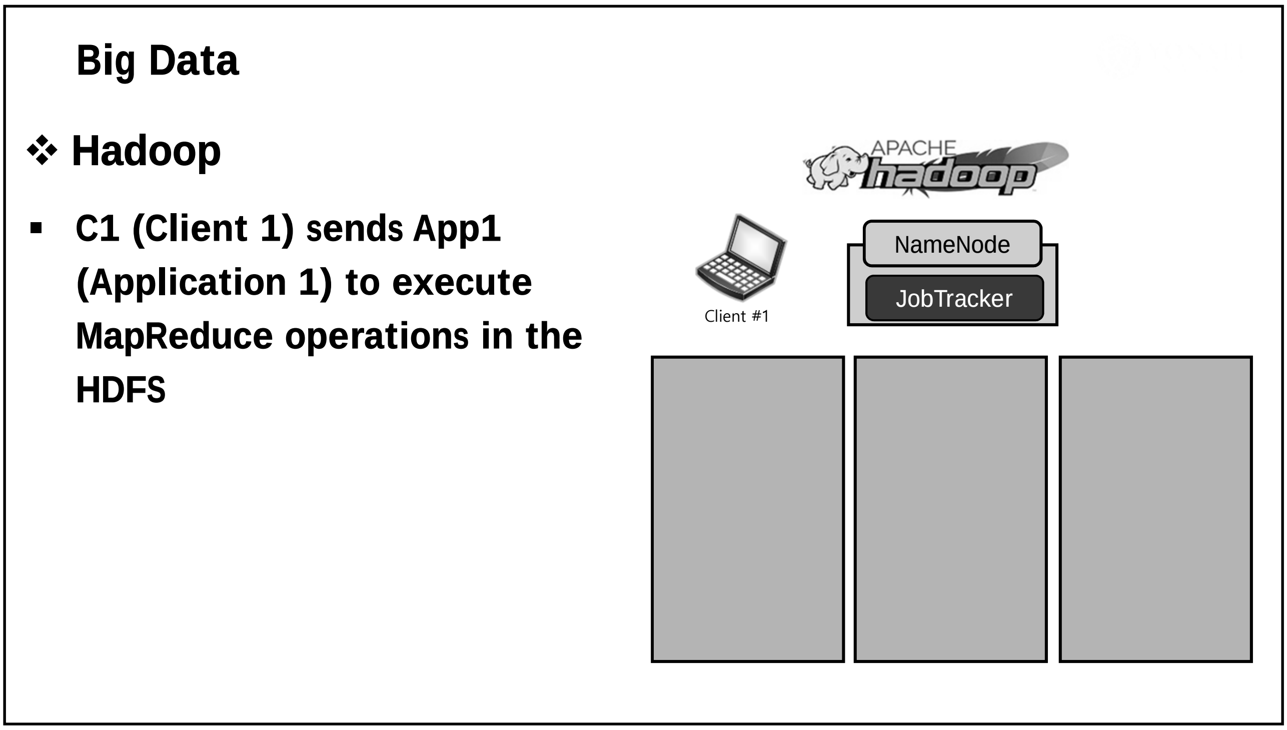
Hadoop vs. Hadoop YARN

Big Data

❖ Hadoop

- C1 (Client 1) sends App1 (Application 1) to execute MapReduce operations in the HDFS

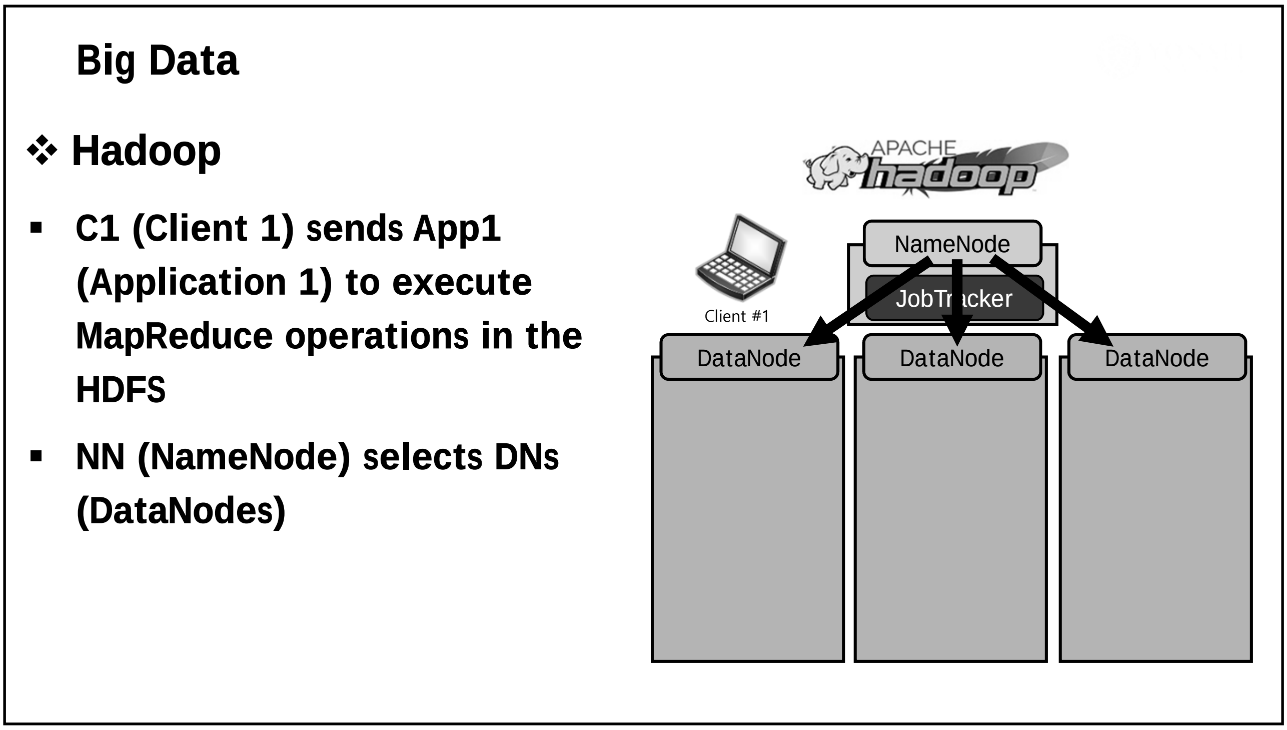




Big Data

❖ Hadoop

- C1 (Client 1) sends App1 (Application 1) to execute MapReduce operations in the HDFS



Big Data

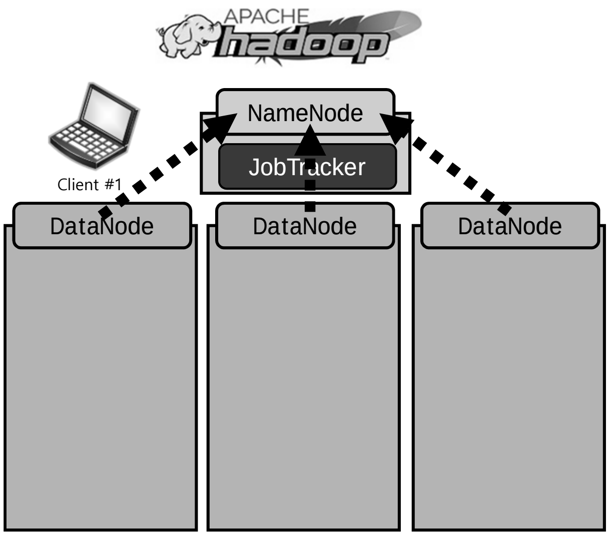
❖ Hadoop

- C1 (Client 1) sends App1 (Application 1) to execute MapReduce operations in the HDFS
- NN (NameNode) selects DNs (DataNodes)

Big Data

❖ Hadoop

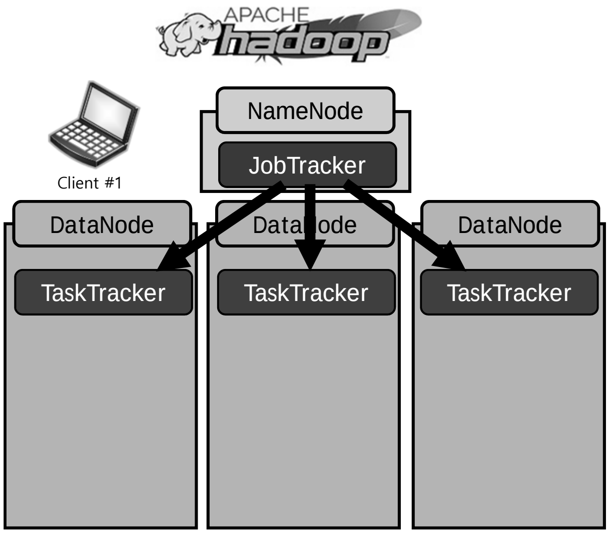
- C1 (Client 1) sends App1 (Application 1) to execute MapReduce operations in the HDFS
- NN (NameNode) selects DNs (DataNodes)
- DNs send Heartbeat signals to the NN every 3 seconds



Big Data

❖ Hadoop

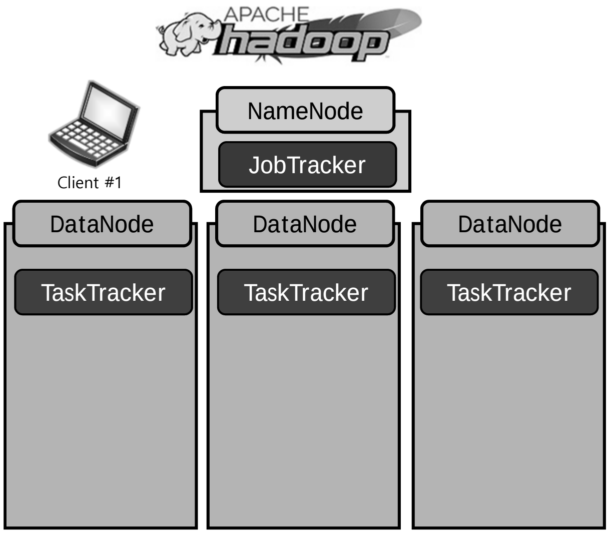
- JT (JobTracker) (brain, master) sets up TTs (TaskTrackers) (workhorse, slave)



Big Data

❖ Hadoop

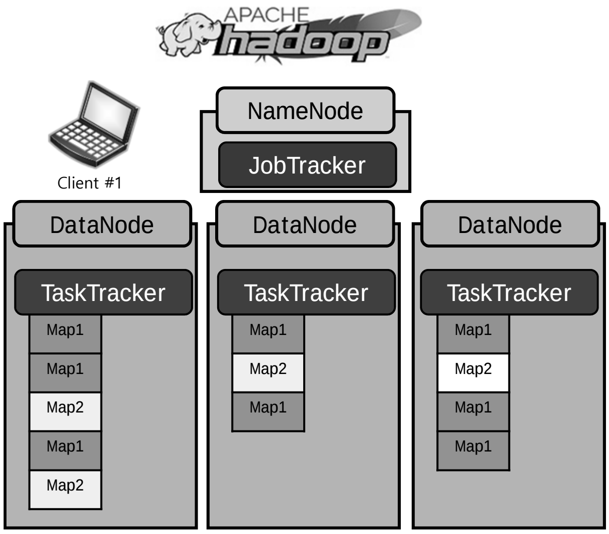
- JT (JobTracker) (brain, master) sets up TTs (TaskTrackers) (workhorse, slave)
- Each TT assigns Slots to be either a Map slot or Reduce slot



Big Data

❖ Hadoop

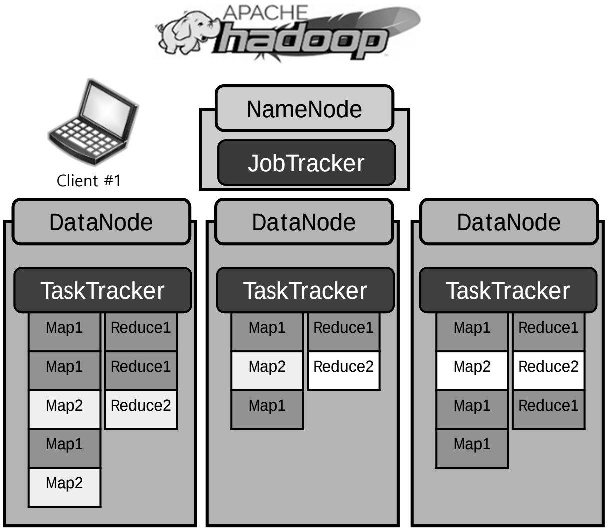
- JT (JobTracker) (brain, master) sets up TTs (TaskTrackers) (workhorse, slave)
- Each TT assigns Slots to be either a Map slot or Reduce slot
- Map Slots are assigned Map functions (JVMs)



Big Data

❖ Hadoop

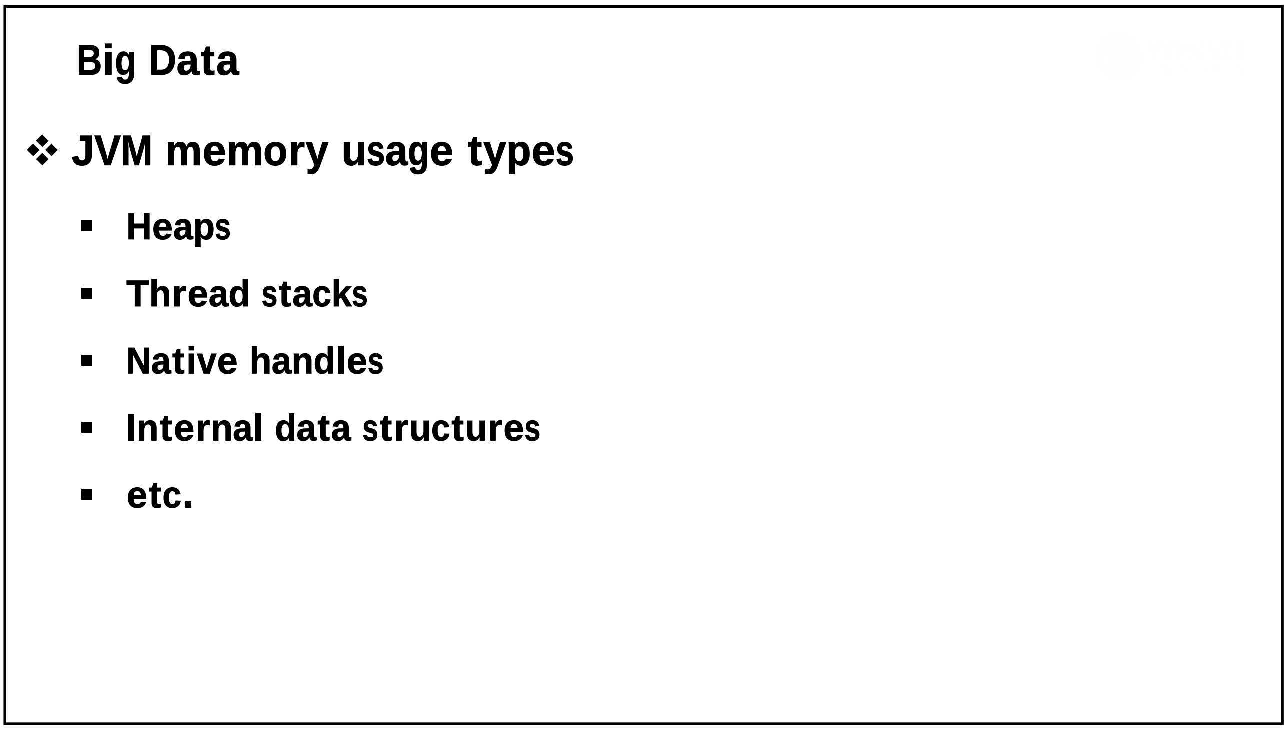
- Reduce Slots are assigned Reduce functions (JVMs)
- Parallel processing is operated based on simultaneously controlling multiple Process IDs



Big Data

❖ Java

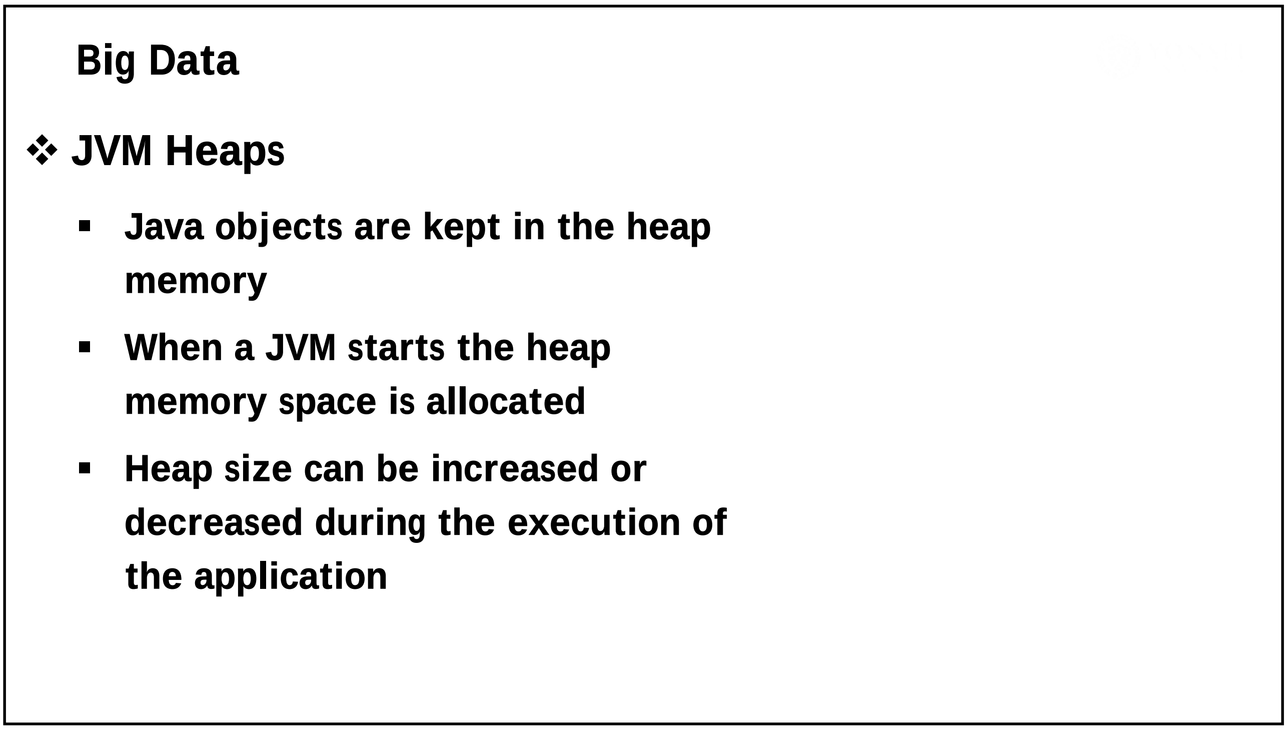
- Trademark is owned by Oracle
- JVM (Java Virtual Machine)
  - Virtual (abstract) computing system on a computer used to execute Java programs
- JRE (Java Runtime Environment)
  - Software package that contains a JVM called HotSpot and JCL (Java Class Library)



Big Data

❖ JVM memory usage types

- Heaps
- Thread stacks
- Native handles
- Internal data structures
- etc.



Big Data

❖ JVM Heaps

- Java objects are kept in the heap memory
- When a JVM starts the heap memory space is allocated
- Heap size can be increased or decreased during the execution of the application

Big Data

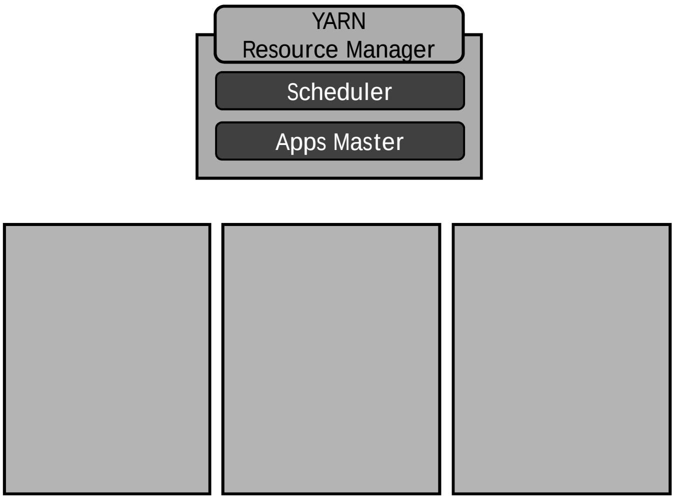
❖ JVM Heaps

- Heap memory is allocated by the JVM using the OS (Operating System)
- JVM conducts heap management of the Java application

Big Data

❖ Hadoop with YARN Example

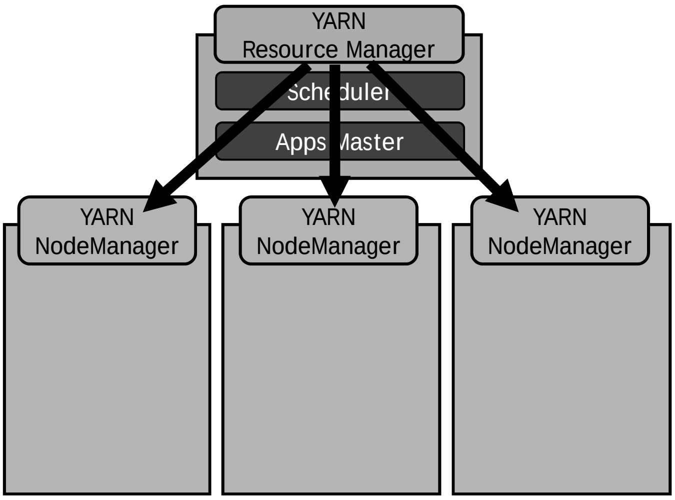
- RM (Resource Manager) has a Scheduler and AM (Apps Master) inside



Big Data

❖ Hadoop with YARN Example

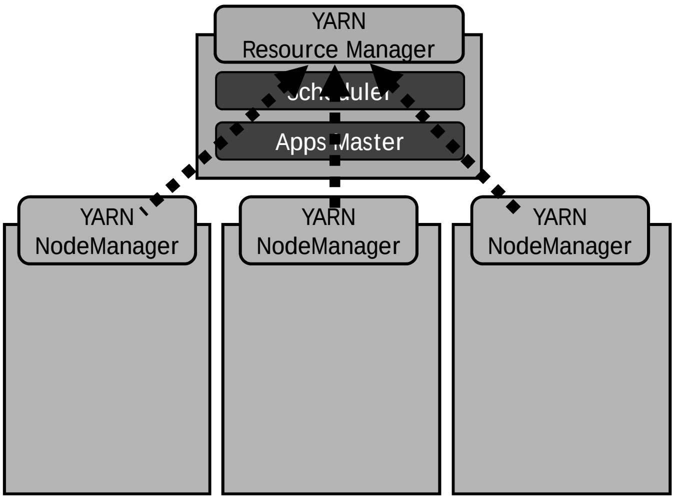
- RM (Resource Manager) has a Scheduler and AM (Apps Master) inside
- RM prepares NMs (Node Managers) on multiple nodes in the cluster



Big Data

❖ Hadoop with YARN Example

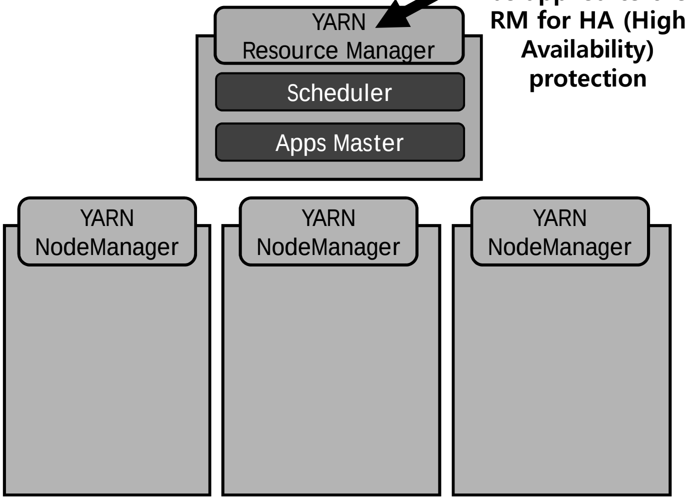
- RM (Resource Manager) has a Scheduler and AM (Apps Master) inside
- RM prepares NMs (Node Managers) on multiple nodes in the cluster
- NMs send Heartbeats to the RM



Big Data

❖ Hadoop with YARN Example

- RM (Resource Manager) has a Scheduler and AM (Apps Master) inside
- RM prepares NMs (Node Managers) on multiple nodes in the cluster
- RM and NMs exchange Heartbeats

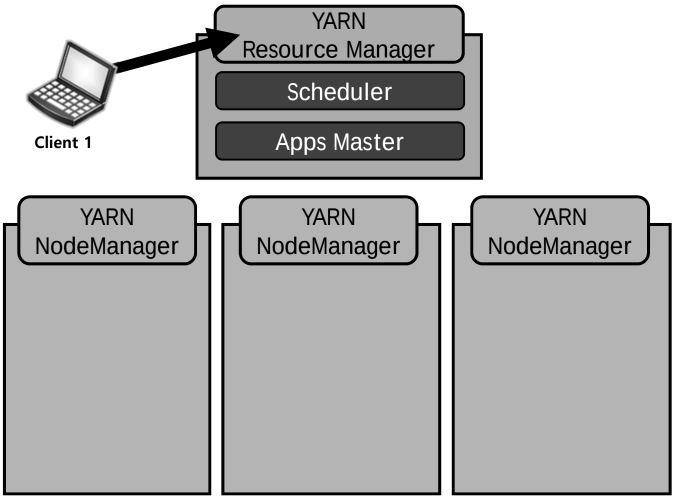


ZooKeeper can be applied to the RM for HA (High Availability) protection

Big Data

❖ Client 1 setup

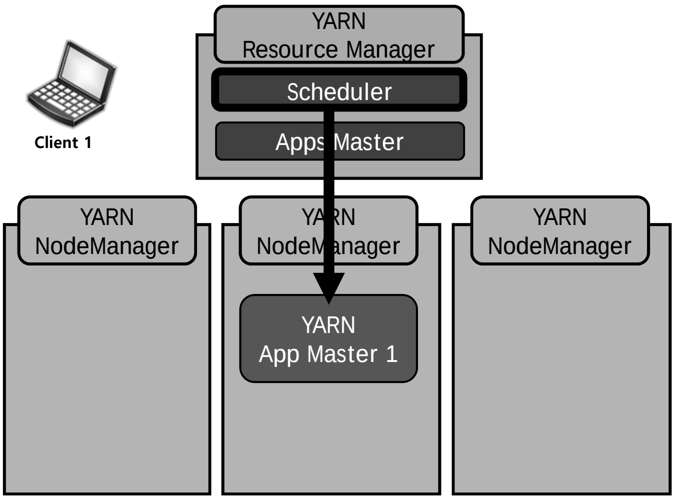
1. Client 1 submits App1 (Application 1) to the RM



Big Data

❖ Client 1 setup

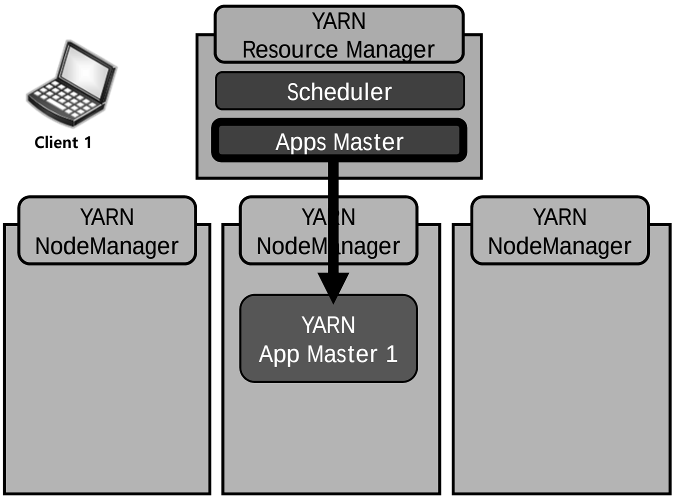
1. Client 1 submits App1 (Application 1) to the RM
2. Scheduler selects a node with sufficient resources to setup AM1 (App Master 1)



Big Data

❖ Client 1 setup

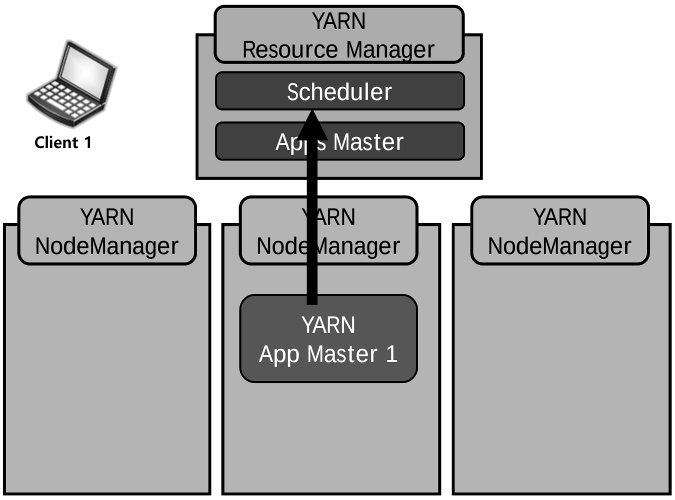
1. Client 1 submits App1 (Application 1) to the RM
2. Scheduler selects a node with sufficient resources to setup AM1 (App Master 1)
3. AM (Apps Master) starts to monitor AM1 (to check if a failure occurs)



Big Data

❖ Client 1 setup

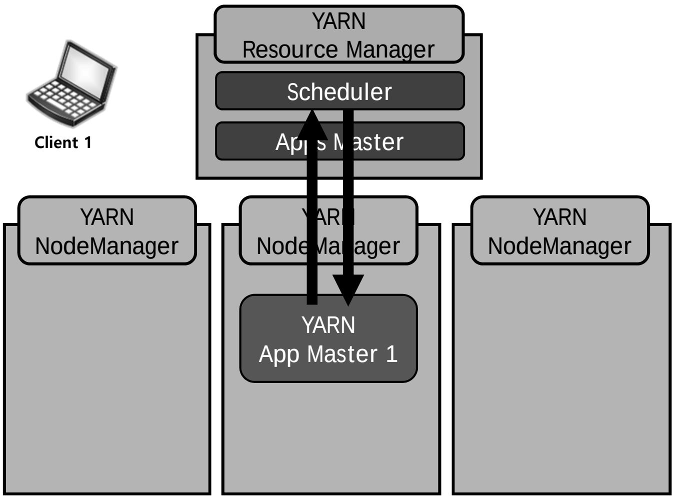
4. AM1 communicates with the Scheduler requesting for Containers to be set on the nodes

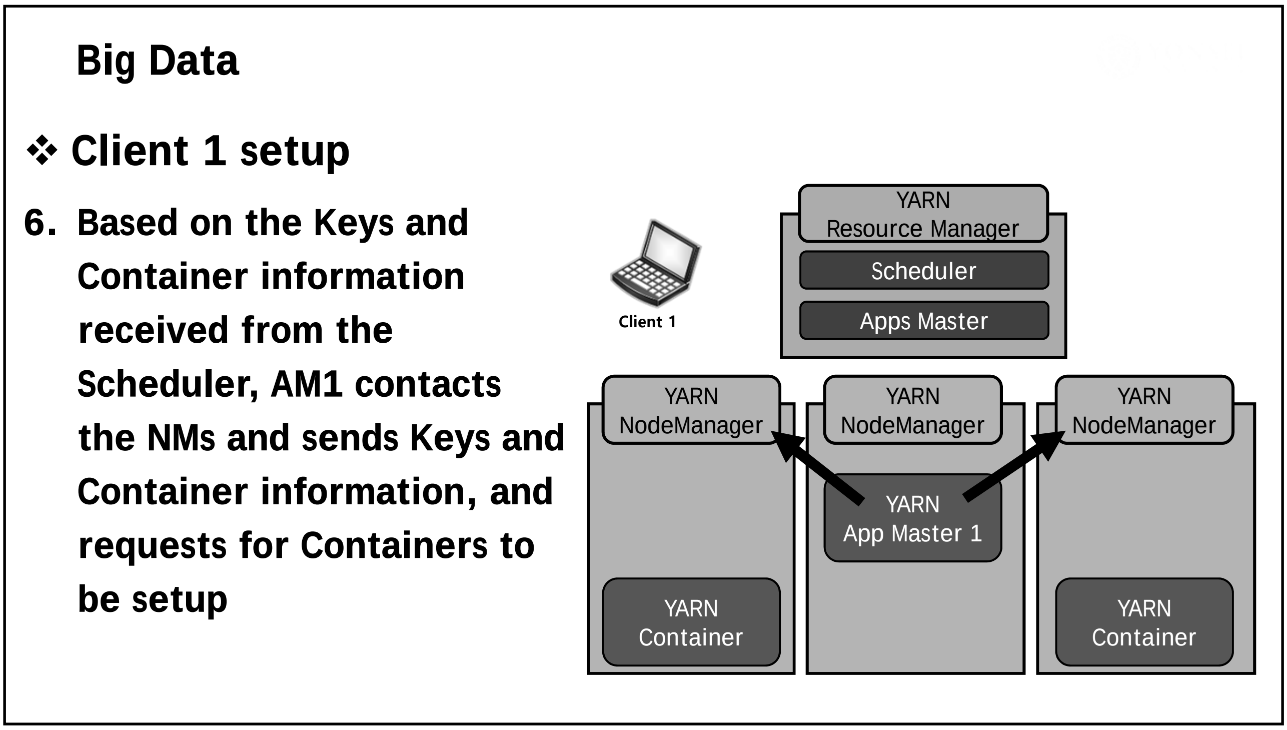


Big Data

❖ Client 1 setup

4. AM1 communicates with the Scheduler requesting for Containers to be set on the nodes
5. Scheduler sends Keys and Container information to AM1 for the Containers to be setup

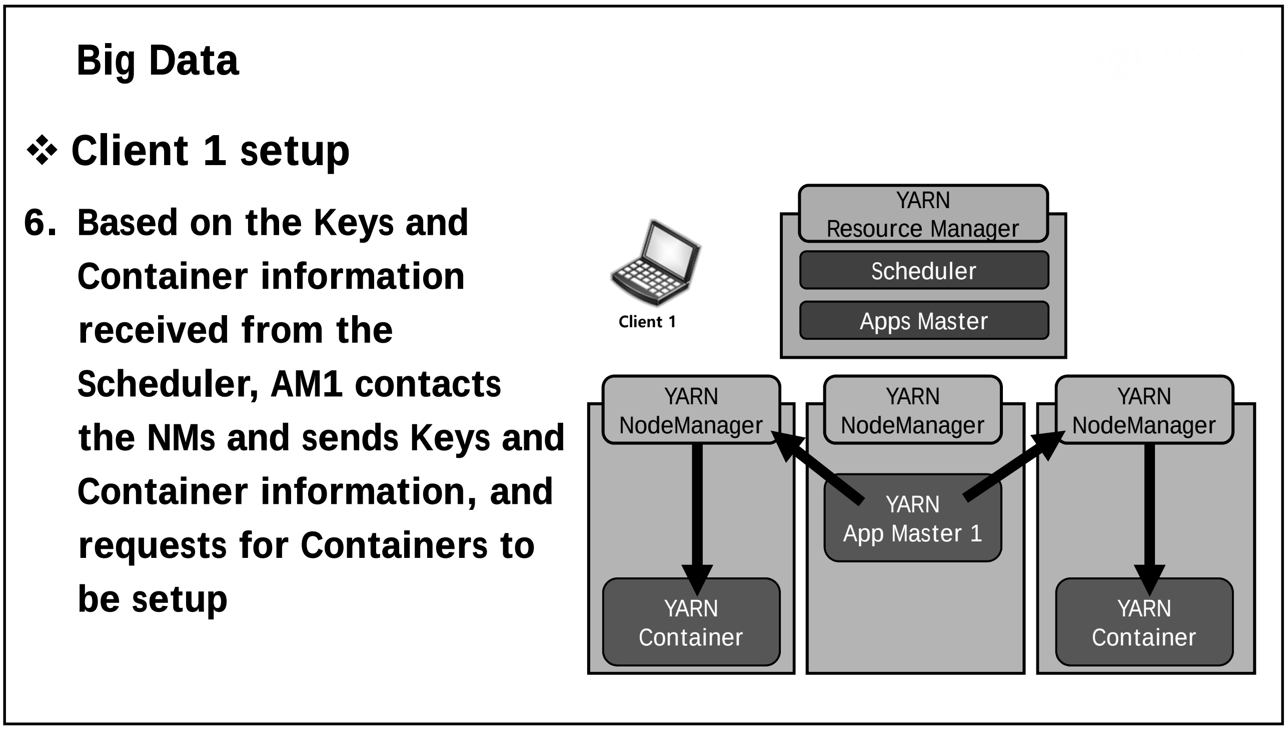




Big Data

❖ Client 1 setup

6. Based on the Keys and Container information received from the Scheduler, AM1 contacts the NMs and sends Keys and Container information, and requests for Containers to be setup



Big Data

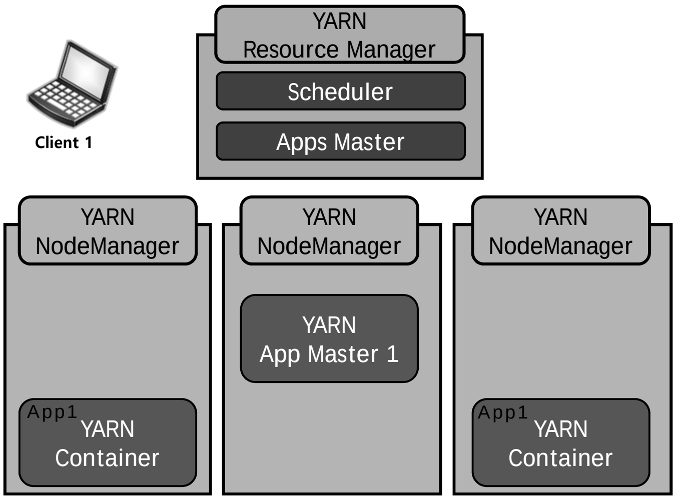
❖ Client 1 setup

6. Based on the Keys and Container information received from the Scheduler, AM1 contacts the NMs and sends Keys and Container information, and requests for Containers to be setup

Big Data

❖ Client 1 setup

7. Each NM contacted by AM1 will setup a Container to run App1 on their node



Client 1

YARN Resource Manager

Scheduler

Apps Master

YARN NodeManager

App1 YARN Container

YARN NodeManager

YARN App Master 1

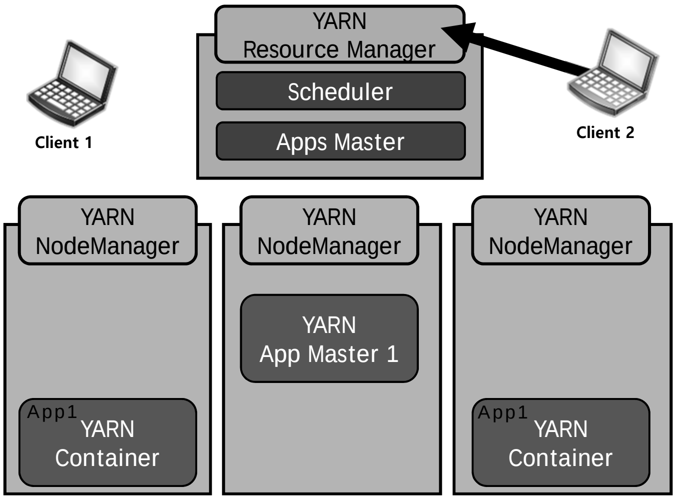
YARN NodeManager

App1 YARN Container

Big Data

❖ Client 2 setup

8. Client 2 submits App2 (Application 2) to the RM



Client 1

YARN Resource Manager

Scheduler

Apps Master

Client 2

YARN NodeManager

App1 YARN Container

YARN NodeManager

YARN App Master 1

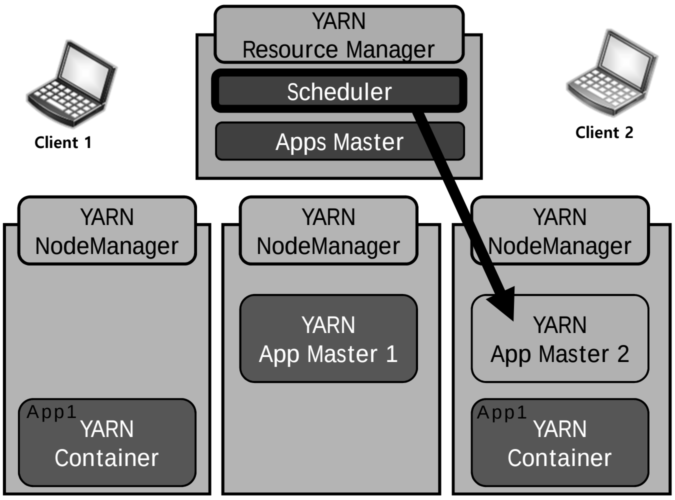
YARN NodeManager

App1 YARN Container

Big Data

❖ Client 2 setup

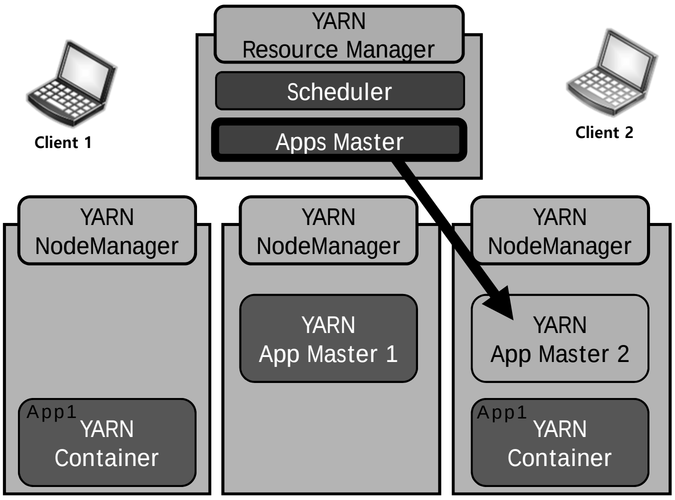
8. Client 2 submits App2 (Application 2) to the RM
9. Scheduler selects a node to setup AM2 (App Master 2)

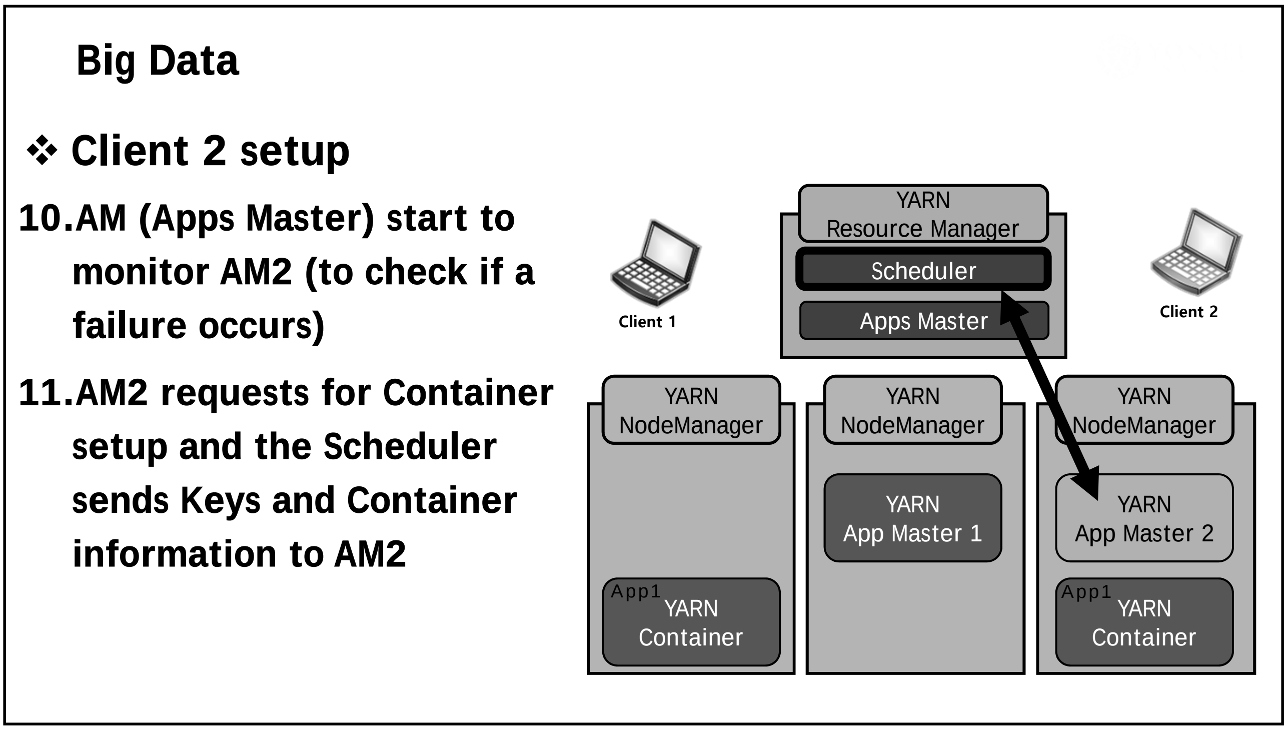


Big Data

❖ Client 2 setup

10. AM (Apps Master) start to monitor AM2 (to check if a failure occurs)



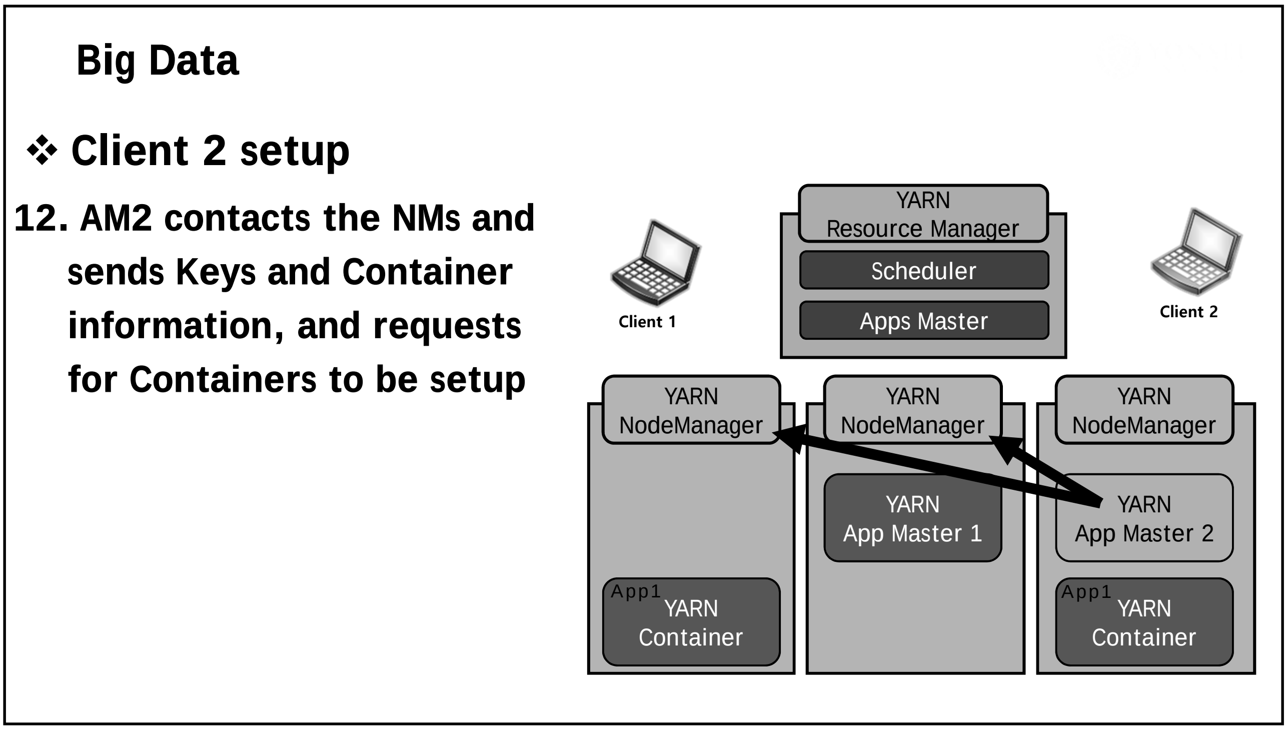


Big Data

❖ Client 2 setup

10. AM (Apps Master) start to monitor AM2 (to check if a failure occurs)

11. AM2 requests for Container setup and the Scheduler sends Keys and Container information to AM2



Big Data

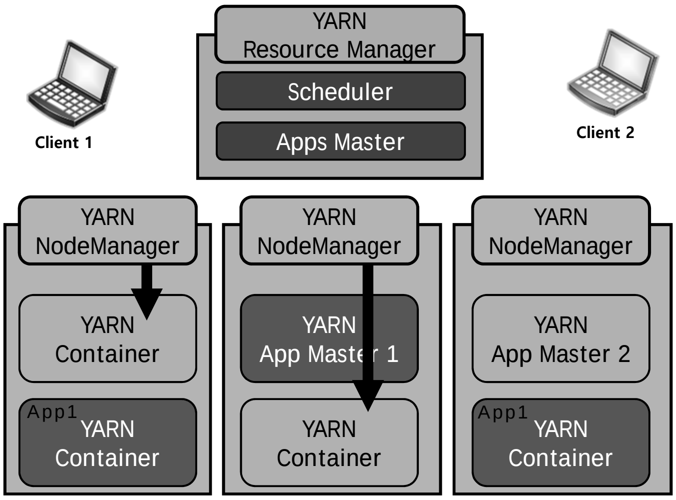
❖ Client 2 setup

12. AM2 contacts the NMs and sends Keys and Container information, and requests for Containers to be setup

Big Data

❖ Client 2 setup

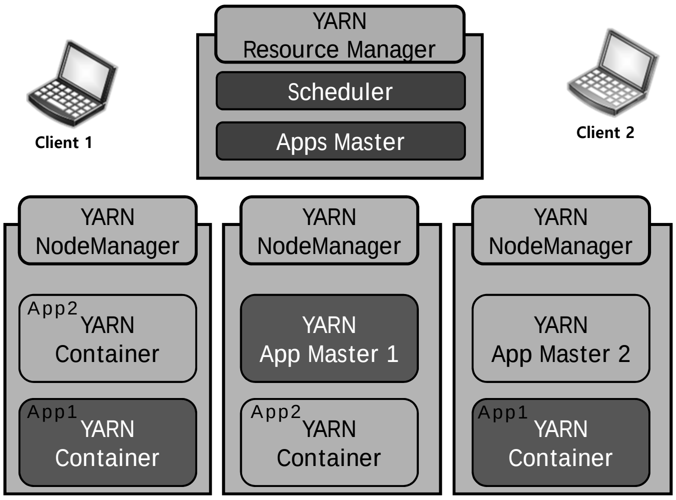
12. AM2 contacts the NMs and sends Keys and Container information, and requests for Containers to be setup
13. Each NM contacted by AM2 will setup a Container to run App2 on their node



Big Data

❖ Client 2 setup

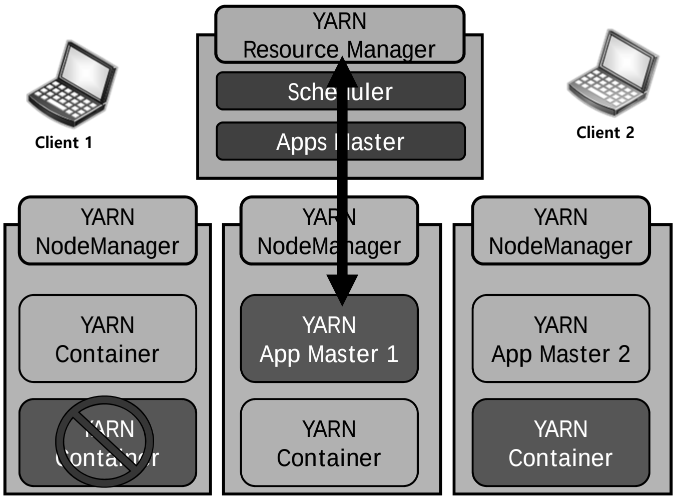
12. AM2 contacts the NMs and sends Keys and Container information, and requests for Containers to be setup
13. Each NM contacted by AM2 will setup a Container to run App2 on their node
14. App2 will run on the new Containers



Big Data

❖ Fault Tolerance in YARN

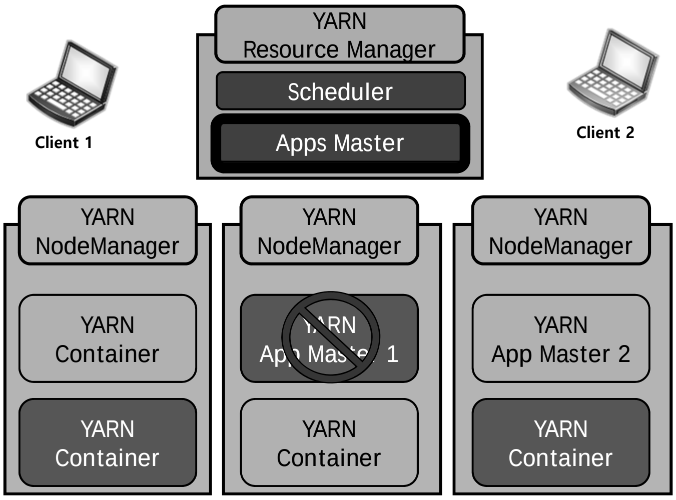
- If a Container crashes, AM1 will communicate with the RM and will setup a new Container to replace the crashed Container



Big Data

❖ Fault Tolerance in YARN

- If AM1 crashes, the AM in the RM will setup a new AM1 (on the same or on another node)

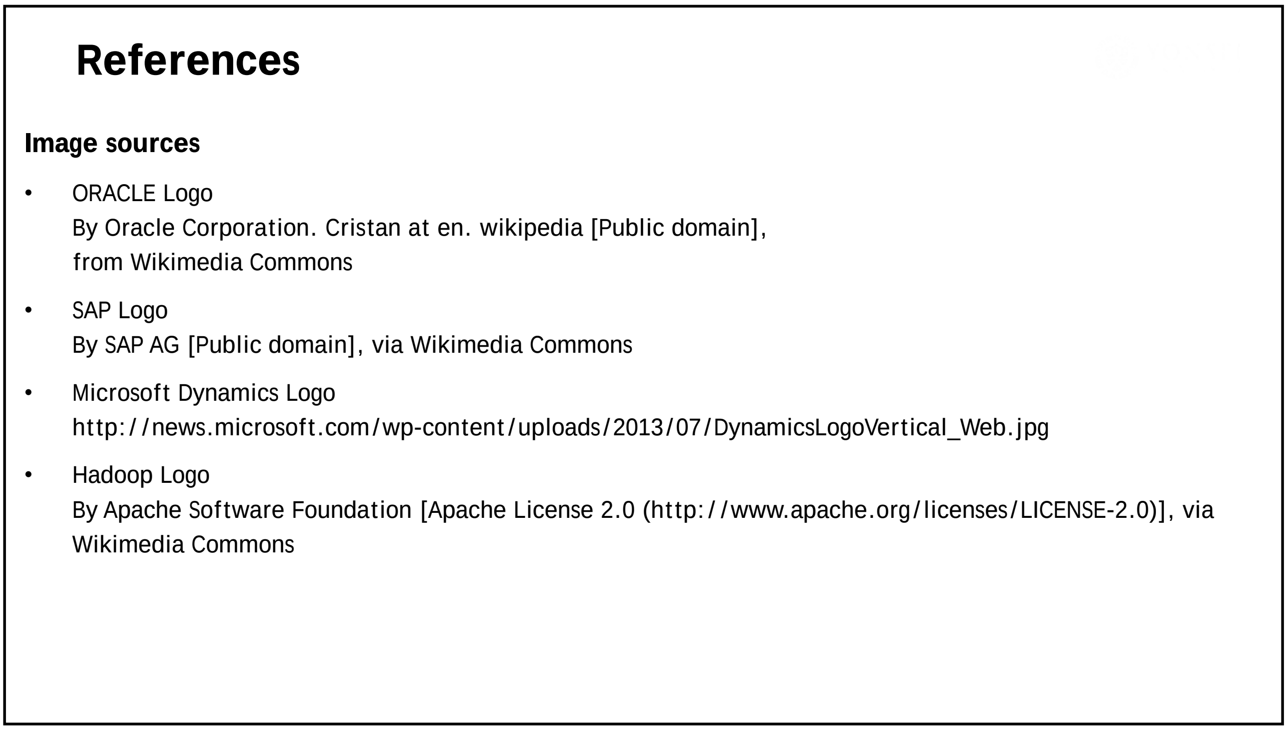


Big Data

References

References

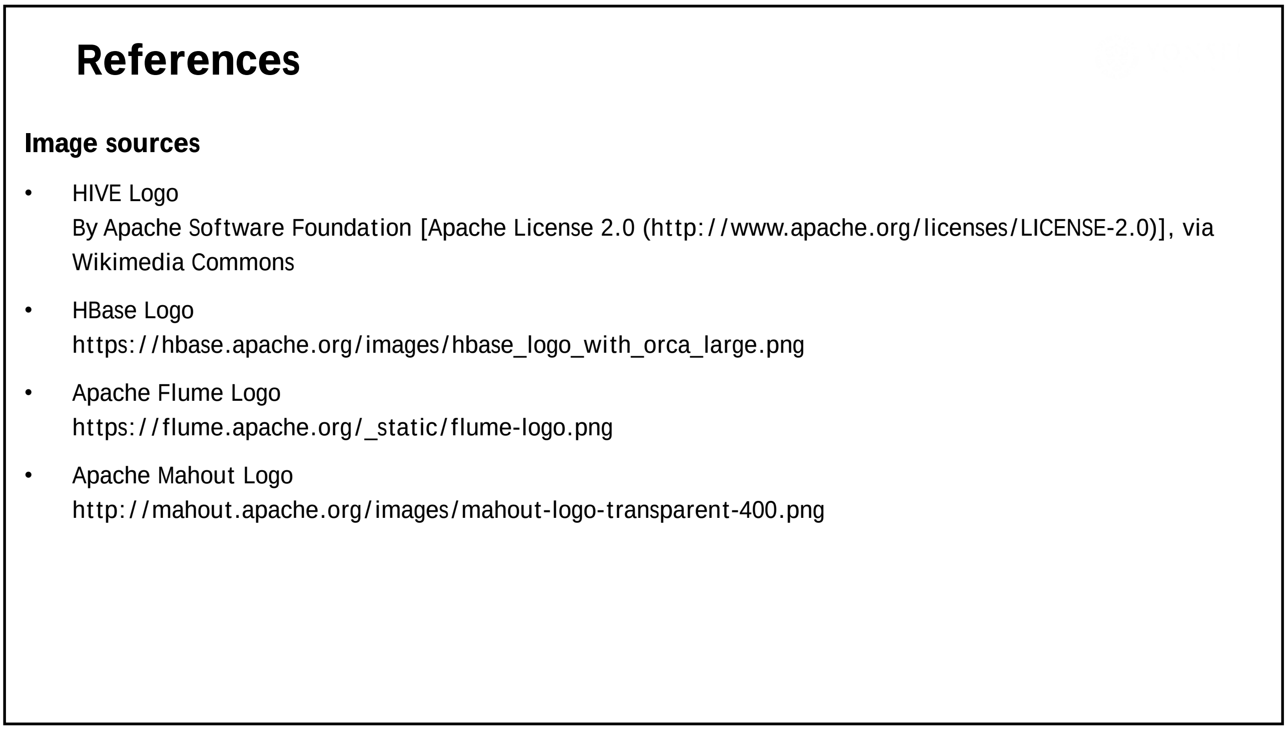
- I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. book in preparation, MIT Press, www.deeplearningbook.org, 2016.
- D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel & S. Dieleman, “Mastering the game of Go with deep neural networks and tree search,” Nature, vol. 529, no. 7587, pp. 484-489, 28 Jan. 2016.
- N. Buduma, Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, O’Reilly Media, Jun. 2015.
- J. Heaton, Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks, Heaton Research, Inc., Nov. 2015.
- Jared Hillam, "What is Hadoop?: SQL Comparison," YouTube, https://www.youtube.com/watch?v=MfF750YVDxM
- Wikipedia, http://www.wikipedia.org



References

Image sources

- ORACLE Logo
  By Oracle Corporation. Cristan at en. wikipedia [Public domain],
  from Wikimedia Commons
- SAP Logo
  By SAP AG [Public domain], via Wikimedia Commons
- Microsoft Dynamics Logo
  http://news.microsoft.com/wp-content/uploads/2013/07/DynamicsLogoVertical_Web.jpg
- Hadoop Logo
  By Apache Software Foundation [Apache License 2.0 (http://www.apache.org/licenses/LICENSE-2.0)], via Wikimedia Commons



References

Image sources

- HIVE Logo
  By Apache Software Foundation [Apache License 2.0 (http://www.apache.org/licenses/LICENSE-2.0)], via Wikimedia Commons
- HBase Logo
  https://hbase.apache.org/images/hbase_logo_with_orca_large.png
- Apache Flume Logo
  https://flume.apache.org/_static/flume-logo.png
- Apache Mahout Logo
  http://mahout.apache.org/images/mahout-logo-transparent-400.png