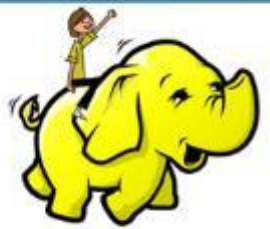
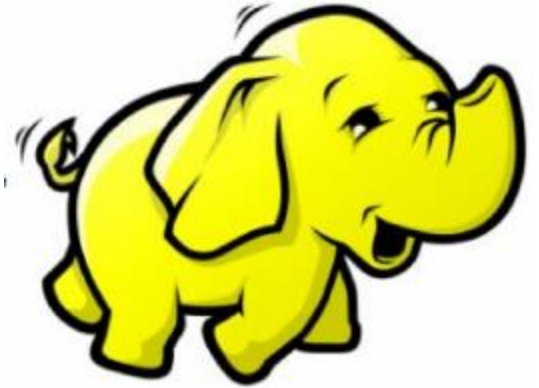
edureka!

Hadoop Administration



Hadoop Administration



Module 5: Hadoop 2.0 and High Availability

Course Topics

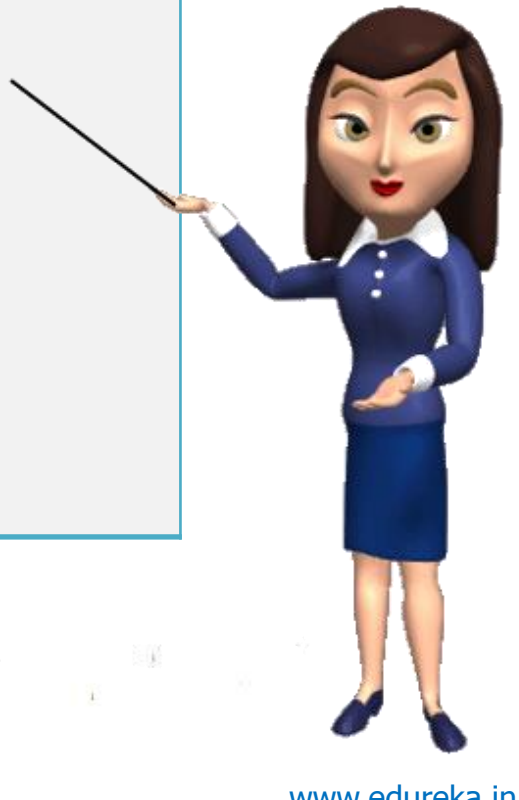
edureka!

- **Module 1**
  - Understanding Big Data
  - Hadoop Components
- **Module 2**
  - Different Hadoop Server Roles
  - Hadoop Cluster Configuration
- **Module 3**
  - Hadoop Cluster Planning
  - Job Scheduling
- **Module 4**
  - Securing your Hadoop Cluster
  - Backup and Recovery
- **Module 5**
  - **Hadoop 2.0 New Features**
  - **HDFS High Availability**
- **Module 6**
  - Quorum Journal Manager (QJM)
  - Hadoop 2.0 - YARN
- **Module 7**
  - Oozie Workflow Scheduler
  - Hive and Hbase Administration
- **Module 8**
  - Hadoop Cluster Case Study
  - Hadoop Implementation

Topics of the Day

edureka!

- Hadoop Architecture
- Problems with Hadoop 1.0
- Solution: Hadoop 2.0 and YARN
- Hadoop 2.0 New Features
  - HDFS High Availability
  - HDFS Federation
- YARN and Hadoop ecosystem
- Hadoop 2.0 Configuration Files
- Hadoop 2.0 Cluster Setup



http://www.pvsd.k12.ca.us/site/default.aspx?PageID=129

Let's Revise

edureka!

- ✓ Plan You Hadoop Cluster
- ✓ Security

Recover using Secondary NameNode

NameNode recovery in Hadoop 1.0

Hadoop Core Components

edureka!

Hadoop is a system for large scale data processing.

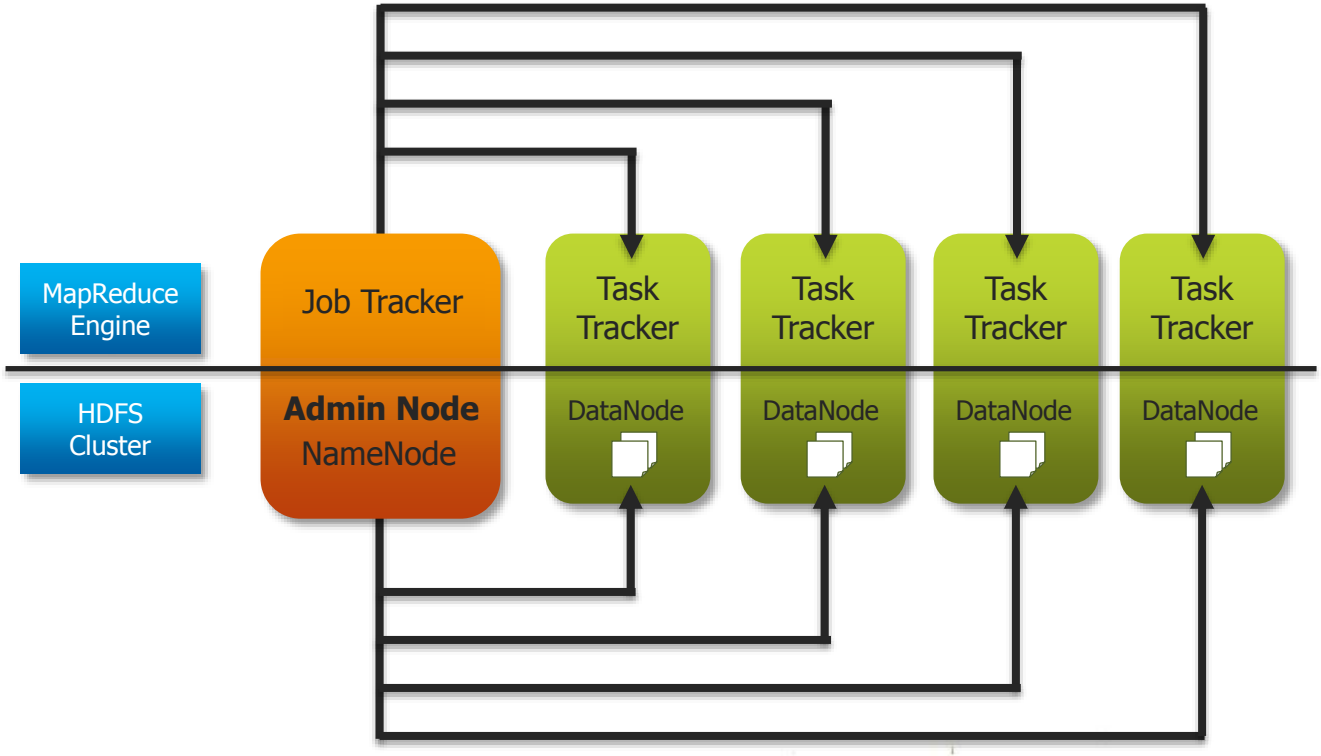
It has two main components:

- ✓ **HDFS – Hadoop Distributed File System (Storage)**
  - ✓ Distributed across "nodes"
  - ✓ Natively redundant
  - ✓ NameNode tracks locations.

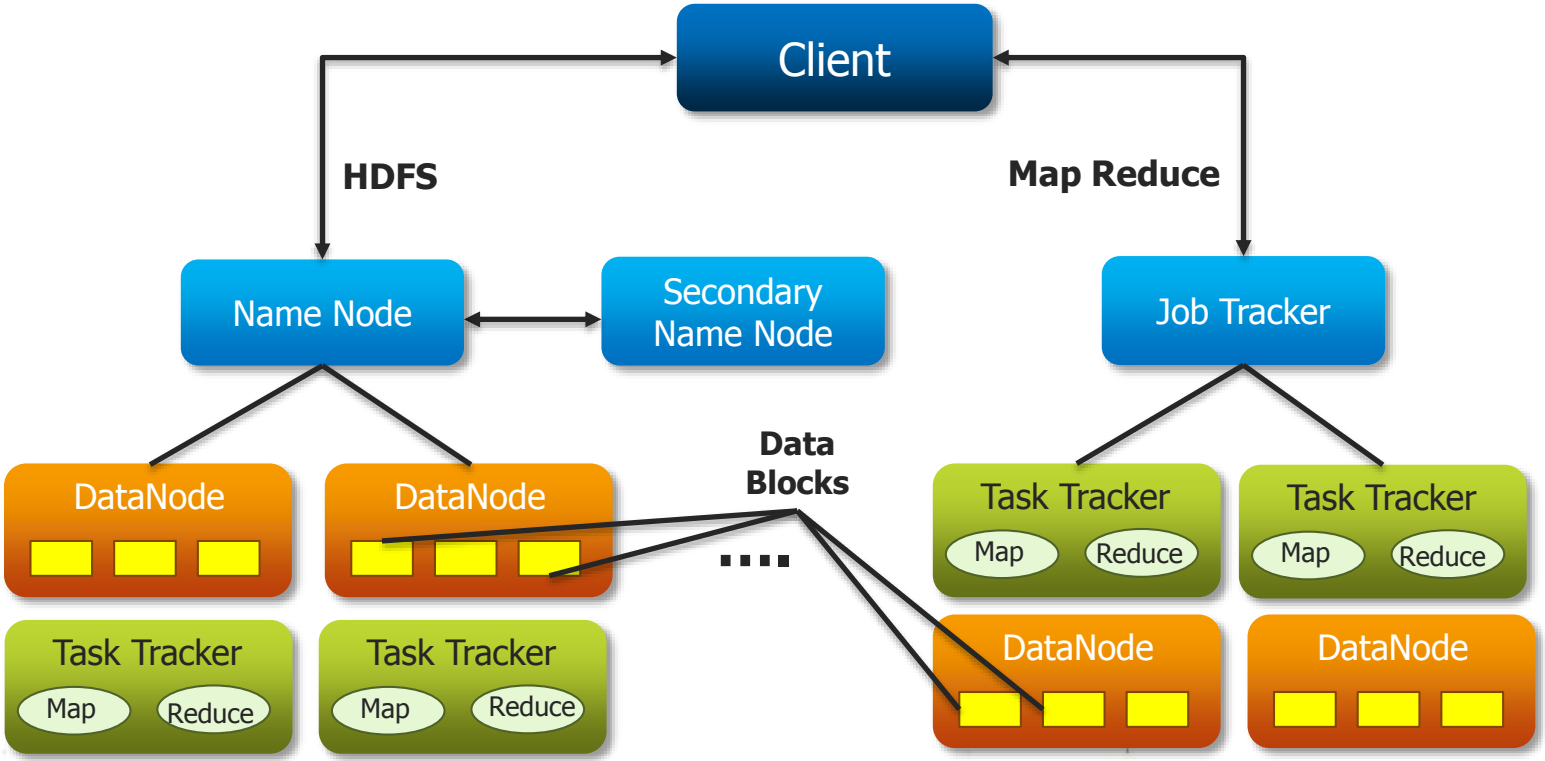
- ✓ **MapReduce (Processing)**
  - ✓ Splits a task across processors
  - ✓ "near" the data & assembles results
  - ✓ Self-Healing, High Bandwidth
  - ✓ Clustered storage
  - ✓ Job Tracker manages the Task Trackers

Hadoop Core Components (Contd.)

edureka!



Hadoop 1.0 – In Summary

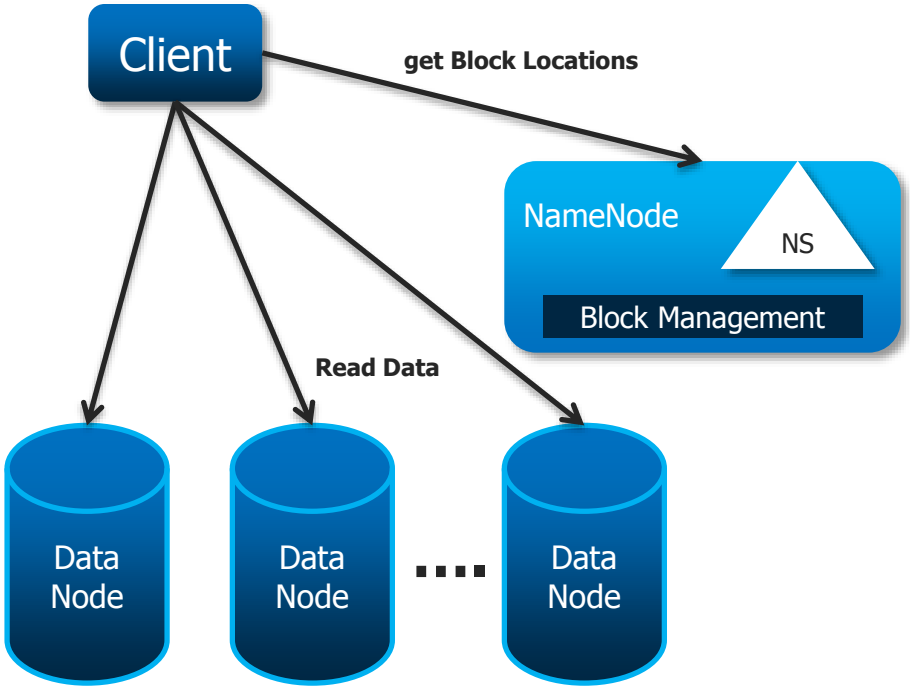


Hadoop 1.0 – Challenges

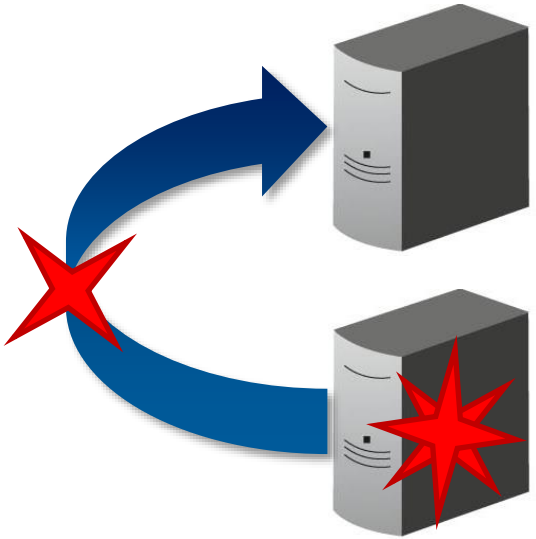
edureka!

Problem	Description
NameNode – No Horizontal Scalability	Single NameNode and Single Namespaces, limited by NameNode RAM
NameNode – No High Availability (HA)	NameNode is Single Point of Failure, Need manual recovery using Secondary NameNode in case of failure
Job Tracker – Overburdened	Spends significant portion of time and effort managing the life cycle of applications
MRv1 – Only Map and Reduce tasks	Humongous Data stored in HDFS remains unutilized and cannot be used for other workloads such as Graph processing etc.

NameNode – HA and Scale



NameNode - No Horizontal Scale

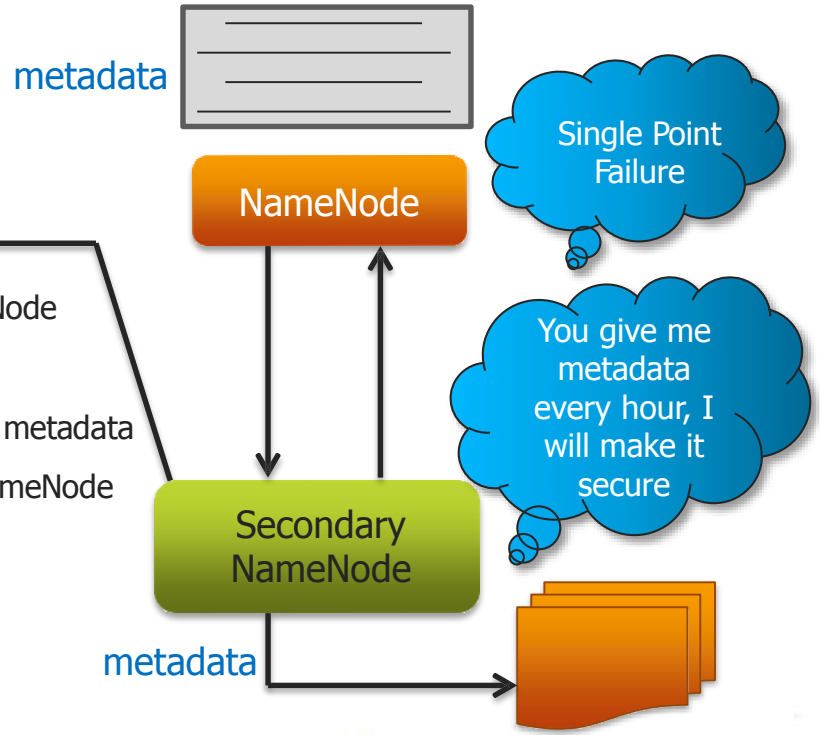


NameNode - No High Availability

NameNode – Single Point of Failure

edureka!

- ✓ **Secondary NameNode:**
  - ✓ **"Not a hot standby"** for the NameNode
  - ✓ Connects to NameNode every hour*
  - ✓ Housekeeping, backup of NemeNode metadata
  - ✓ Saved metadata can build a failed NameNode



Job Tracker – Overburdened

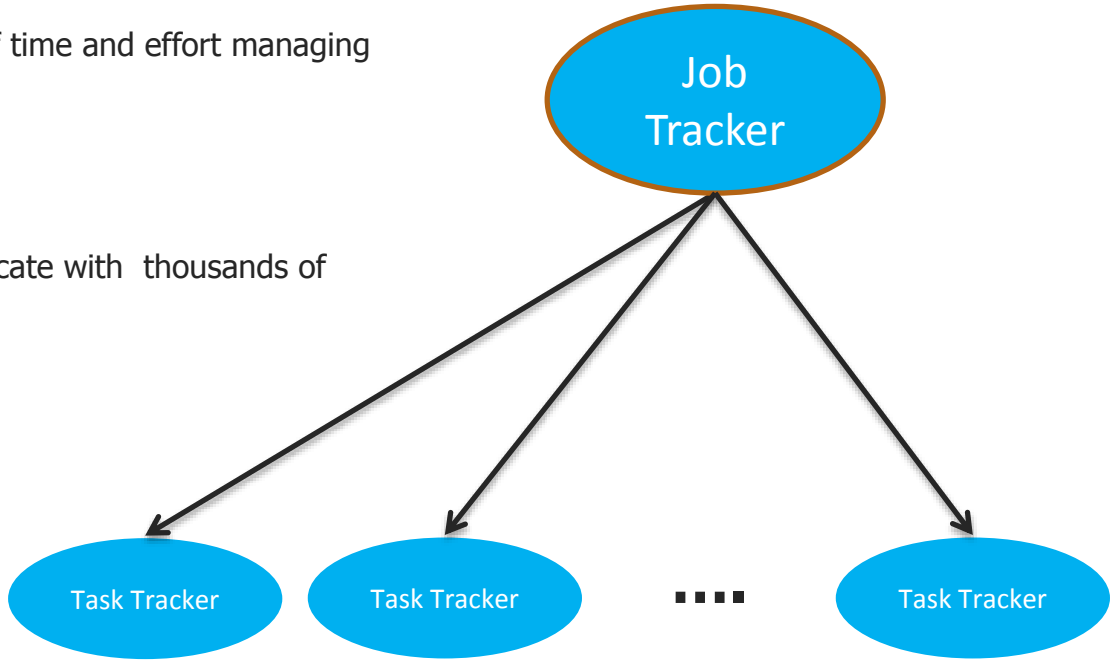
edureka!

CPU

- ✓ Spends a very significant portion of time and effort managing the life cycle of applications

Network

- ✓ Single Listener Thread to communicate with thousands of Map and Reduce Jobs



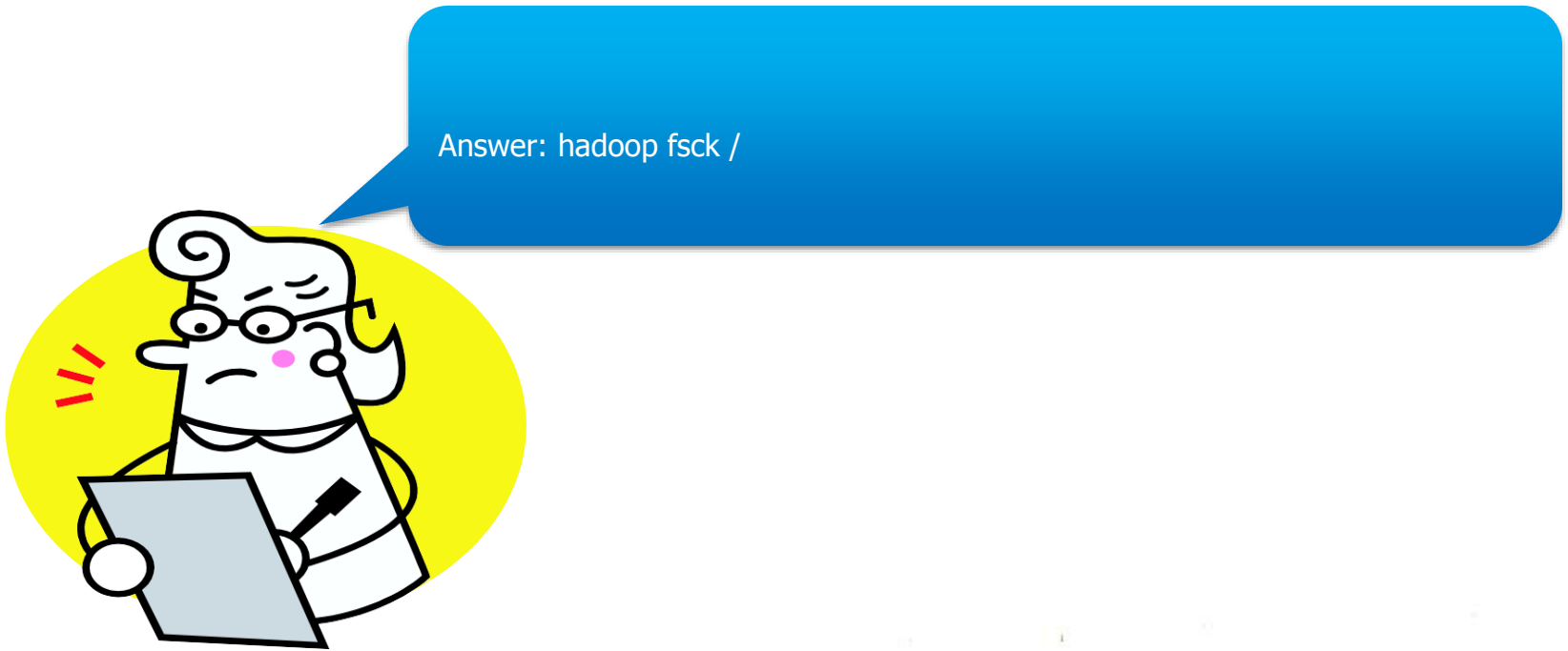
Annie's Question

edureka!

The following command will display the status for your entire file system namespace.

a) hadoop fsck /
b) hadoop fs ls

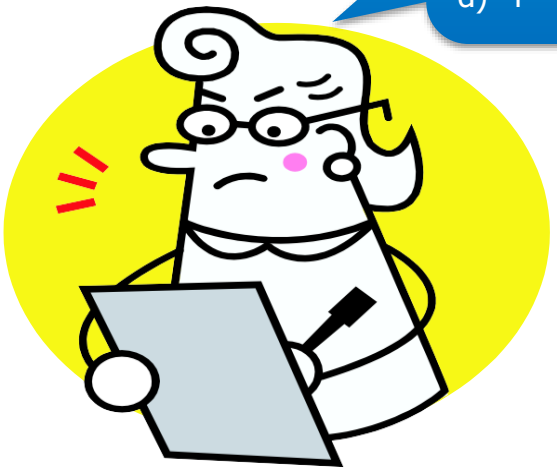
Annie's Answer



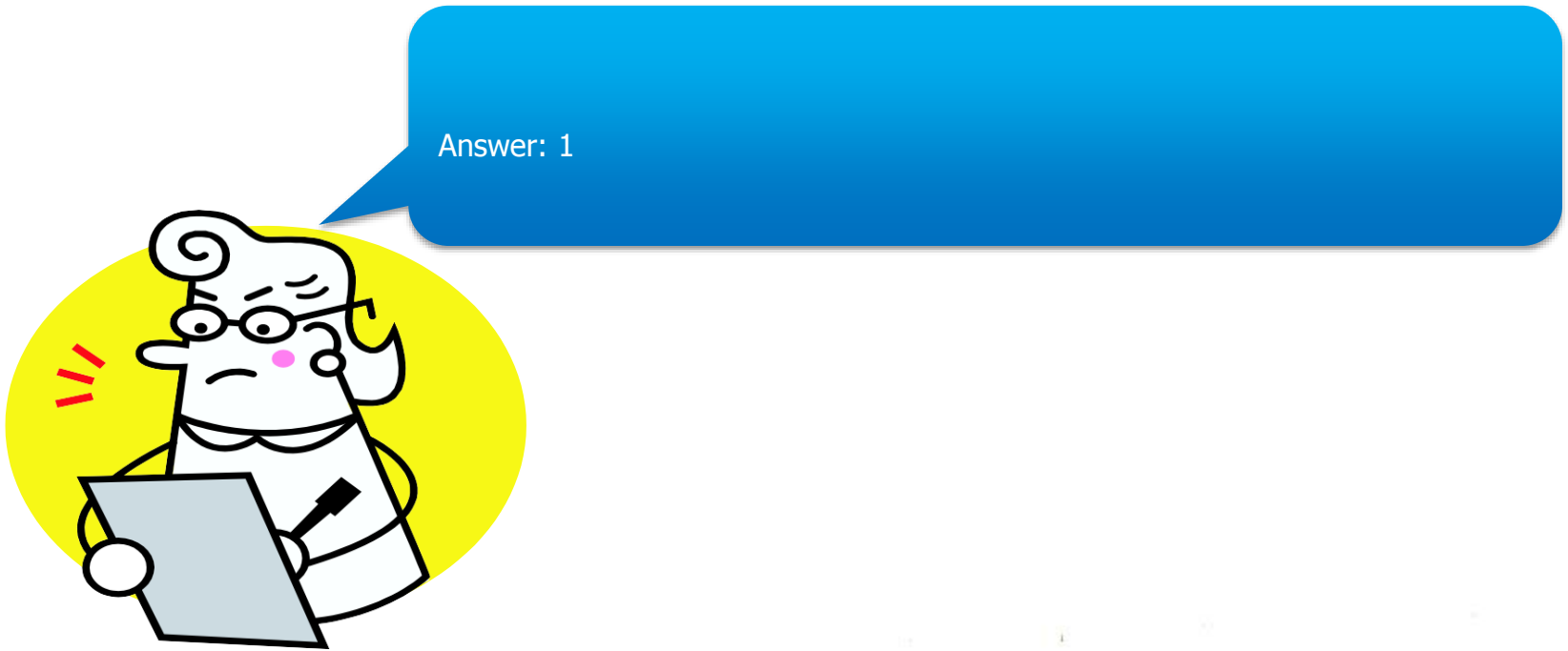
Annie's Question

edureka!

The Number of Job Trackers in a Hadoop Cluster are:
a) 1
b) 2
c) 3
d) 4



Annie’s Answer

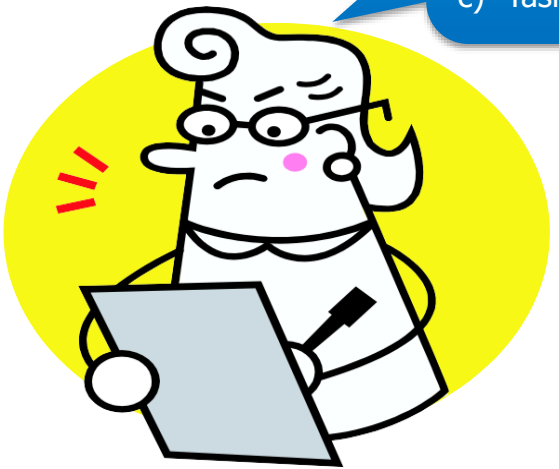


Annie's Question

edureka!

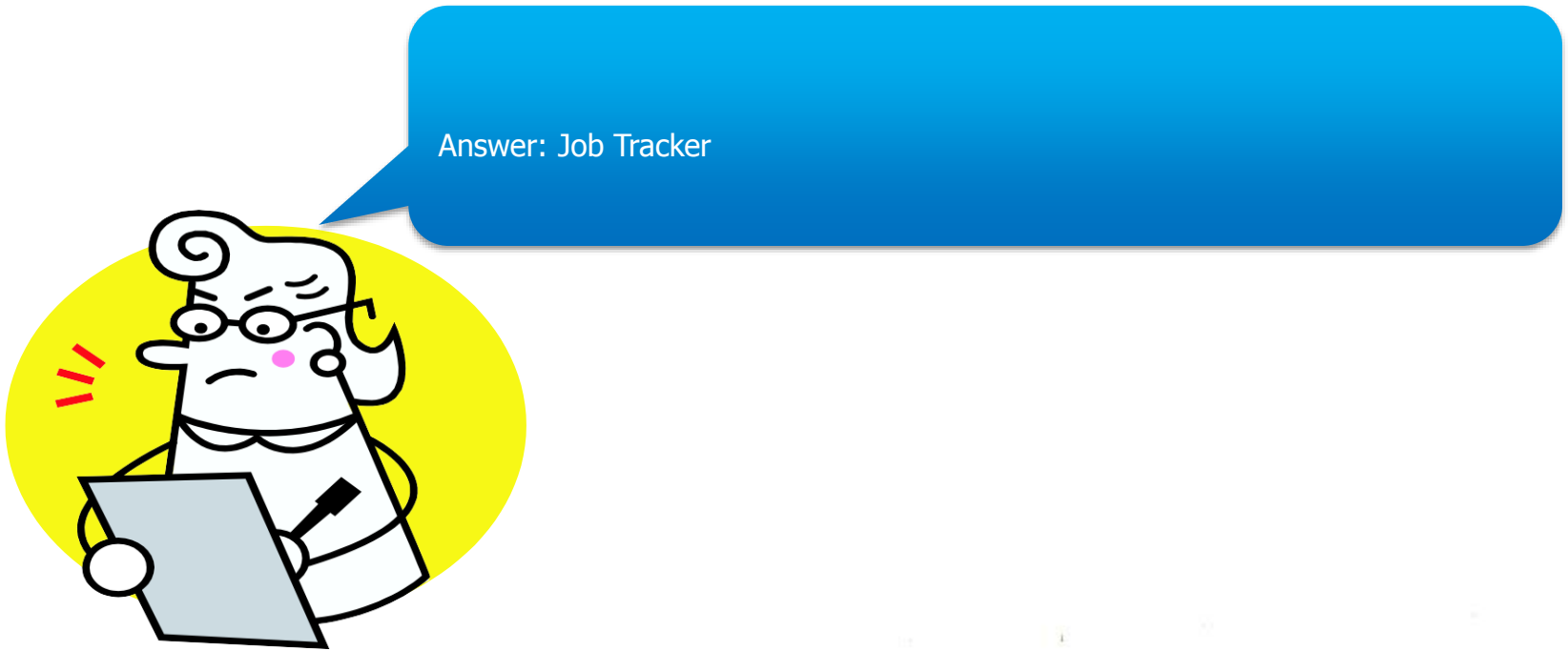
If a task fails, which of the following automatically resubmit the task (possibly on a different node)?

a) NameNode
b) Job Tracker
c) Task Tracker



Annie's Answer

edureka!



MRv1 – Unpredictability in Large Clusters

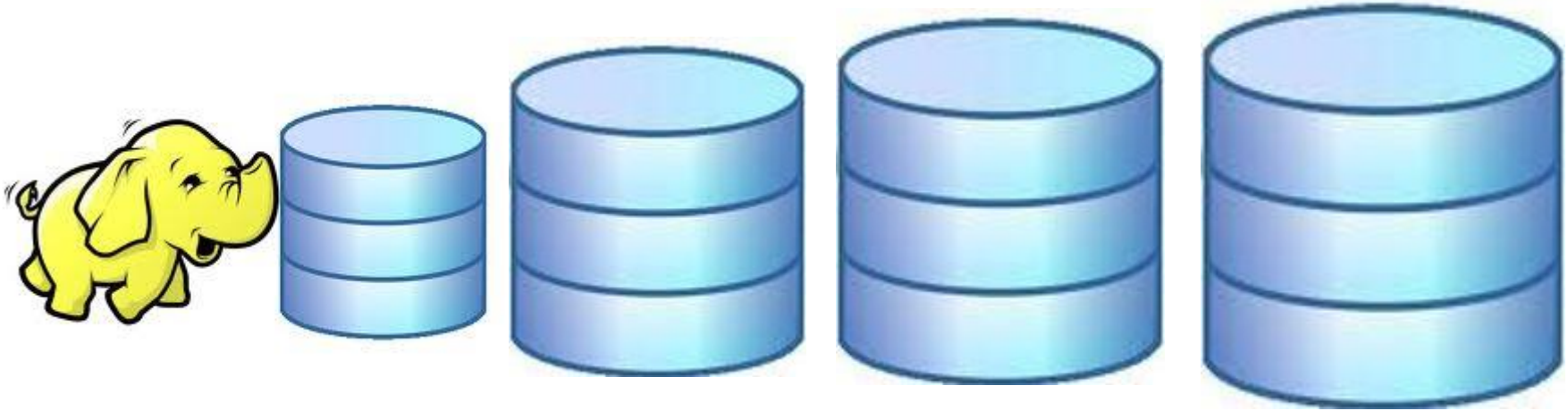
edureka!

As the cluster size grow and reaches to 4000 Nodes

- ✓ **Cascading Failures**
  - ✓ The DataNode failures results in a serious deterioration of the overall cluster performance because of attempts to replicate data and overload live nodes, through network flooding.
- ✓ **Multi-tenancy**
  - ✓ As clusters increase in size, you may want to employ these clusters for a variety of models. MRv1 dedicates its nodes to Hadoop and cannot be re-purposed for other applications and workloads in an Organization. With the growing popularity and adoption of cloud computing among enterprises, this becomes more important.

Unutilized Data in HDFS

edureka!



- ✓ Terabytes and Petabytes of data in HDFS can be used only for MapReduce processing

Hadoop 2.0 New Features

edureka!

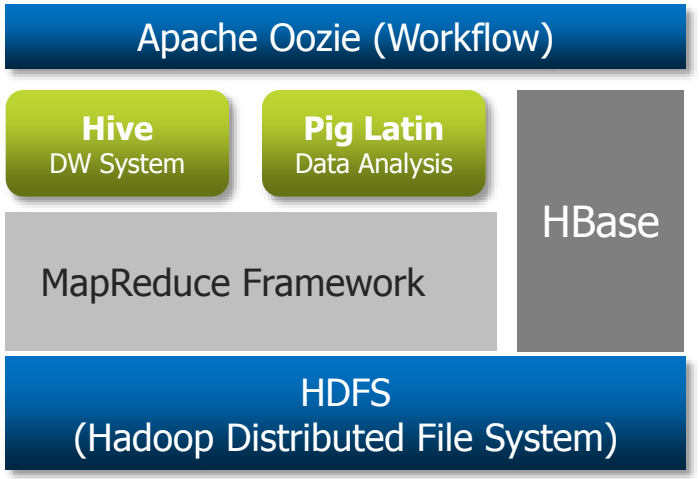
Property	Hadoop 1.0	Hadoop 2.0
Federation	One NameNode and Namespaces	Multiple NameNode and Namespaces
High Availability	Not present	Highly Available
YARN - Processing Control and Multi-tenancy	Job Tracker, Task Tracker	Resource Manager, Node Manager, App Master, Capacity Scheduler

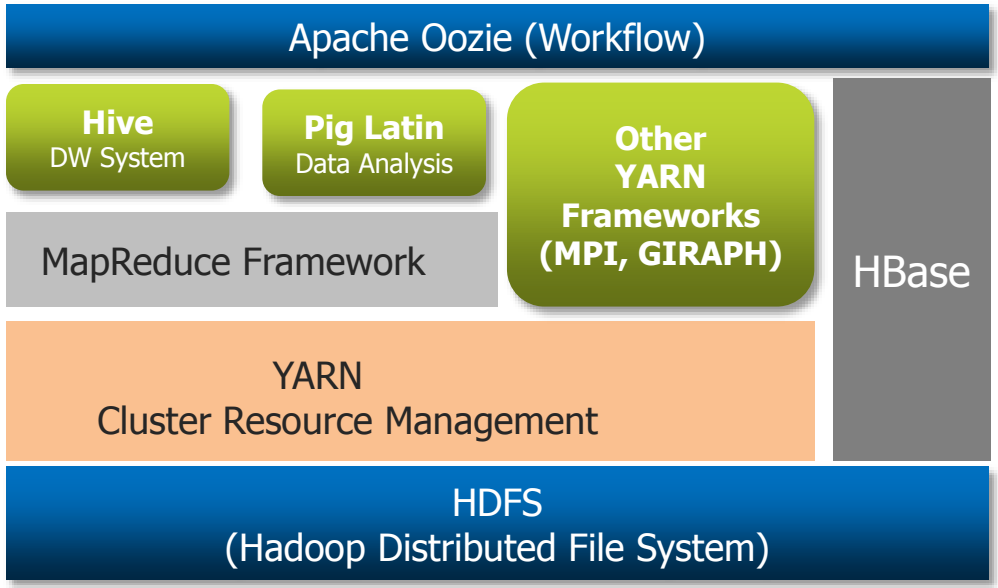
Other important Hadoop 2.0 features

- ✓ HDFS Snapshots
- ✓ NFSv3 access to data in HDFS
- ✓ Support for running Hadoop on MS Windows
- ✓ Binary Compatibility for MapReduce applications built on Hadoop 1.0
- ✓ Substantial amount of Integration testing with rest of the projects (such as PIG, HIVE) in Hadoop ecosystem

YARN and Hadoop Ecosystem

edureka!



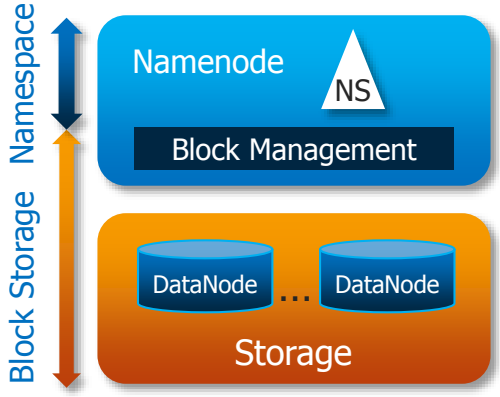


YARN adds a more general interface to run non-MapReduce jobs (such as Graph Processing) within the Hadoop framework

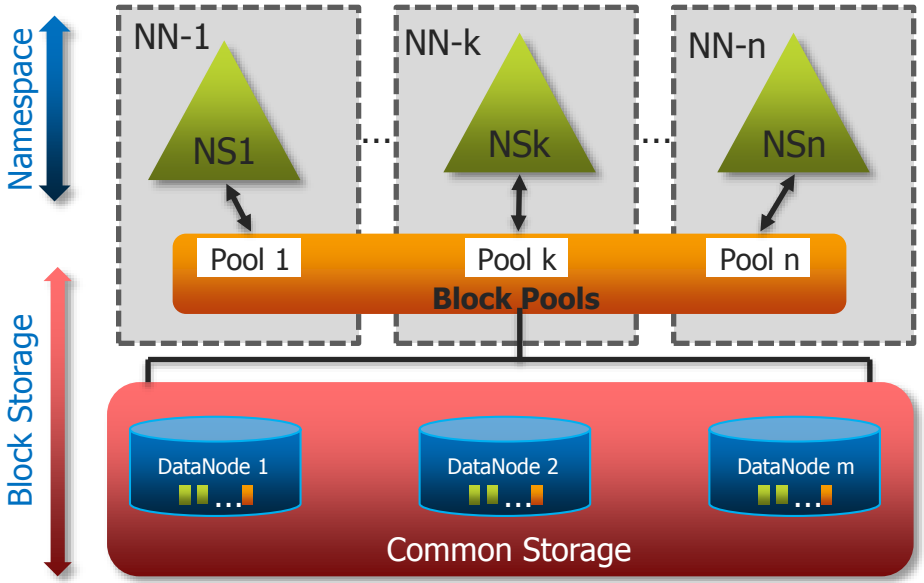
Hadoop 2.0 Cluster Architecture - Federation

edureka!

Hadoop 1.0



Hadoop 2.0



http://hadoop.apache.org/docs/stable2/hadoop-project-dist/hadoop-hdfs/Federation.html

Annie's Question

edureka!

How does HDFS Federation help HDFS Scale horizontally?

a) Reduces the load on any single NameNode by using the multiple, independent NameNode to manage individual parts of the file system namespace.

b) Provides cross-data centre (non-local) support for HDFS, allowing a cluster administrator to split the Block Storage outside the local cluster.

Annie's Answer

Answer: In order to scale the name service horizontally, HDFS federation uses multiple independent NameNode. The NameNode are federated, that is, the NameNode are independent and don't require coordination with each other.

Annie's Question

edureka!

You have configured two name nodes to manage /marketing and /finance respectively. What will happen if you try to put a file in /accounting directory?

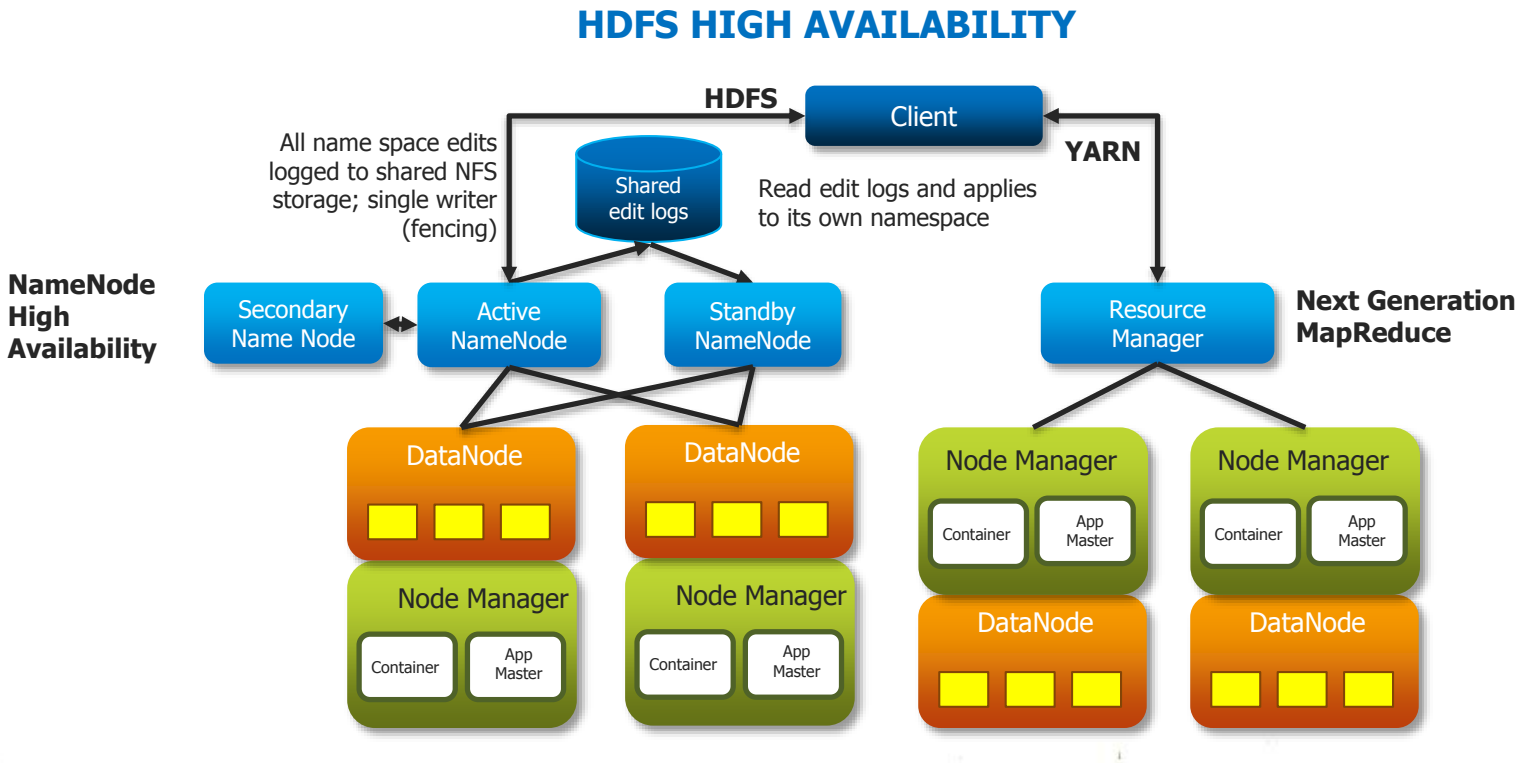
Annie's Answer

edureka!

Answer: The put will fail. None of the namespace will manage the file and you will get an IOException with a No such file or directory error.

Hadoop 2.0 Cluster Architecture - HA

edureka!



http://hadoop.apache.org/docs/stable2/hadoop-yarn/hadoop-yarn-site/HDFSHighAvailabilityWithNFS.html

Annie's Question

edureka!

HDFS HA was developed to overcome the following disadvantage in Hadoop 1.0?

a) Single Point Of Failure Of NameNode
b) Only one version can be run in classic MapReduce
c) Too much burden on Job Tracker

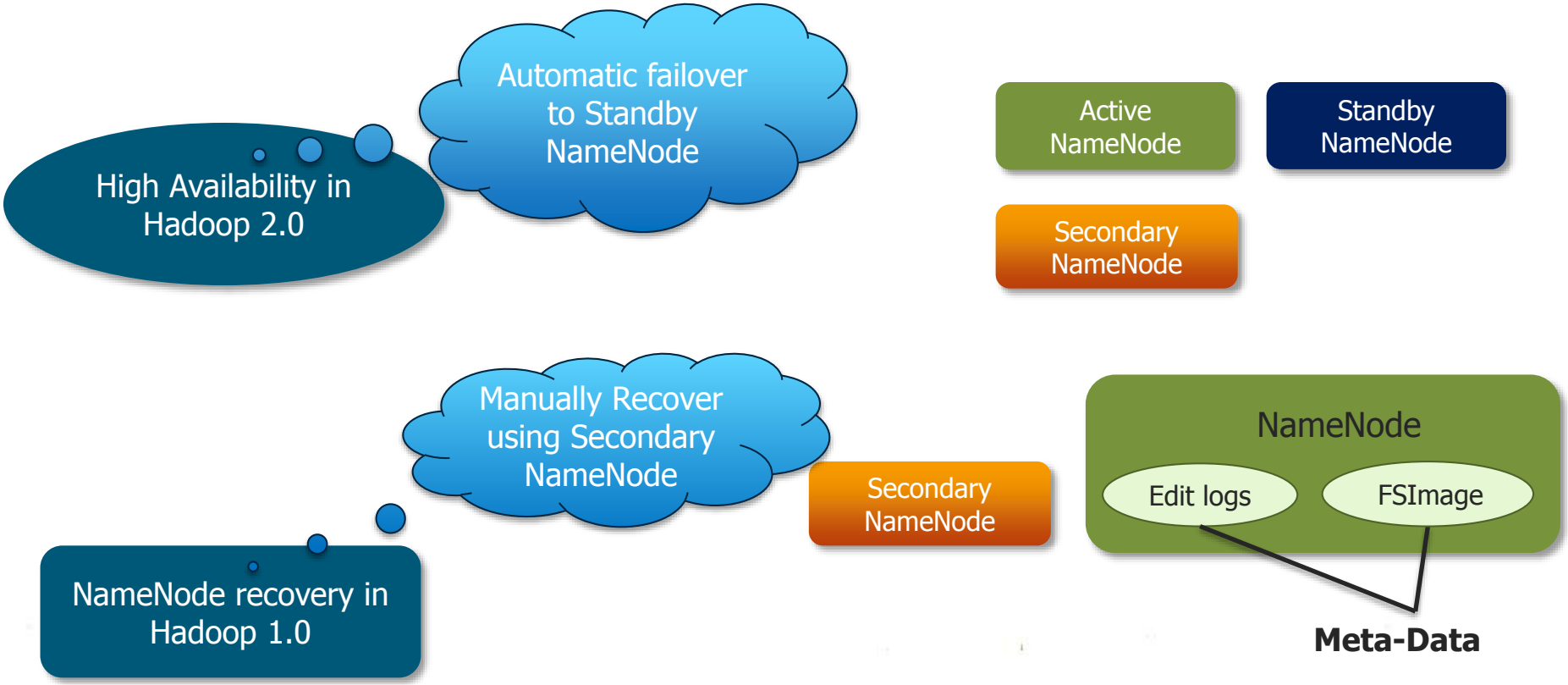
Annie's Answer

edureka!

Answer: Single Point of Failure of NameNode.

NameNode Recovery Vs. Failover

edureka!



Hadoop 1.0 Configuration Files

edureka!

Configuration Filenames	Description of Log Files
hadoop-env.sh	Environment variables that are used in the scripts to run Hadoop.
core-site.xml	Configuration settings for Hadoop Core such as I/O settings that are common to HDFS and MapReduce.
hdfs-site.xml	Configuration settings for HDFS daemons, the namenode, the secondary namenode and the data nodes.
mapred-site.xml	Configuration settings for MapReduce daemons : the job-tracker and the task-trackers.
masters	A list of machines (one per line) that each run a secondary NameNode.
slaves	A list of machines (one per line) that each run a datanode and a task-tracker.

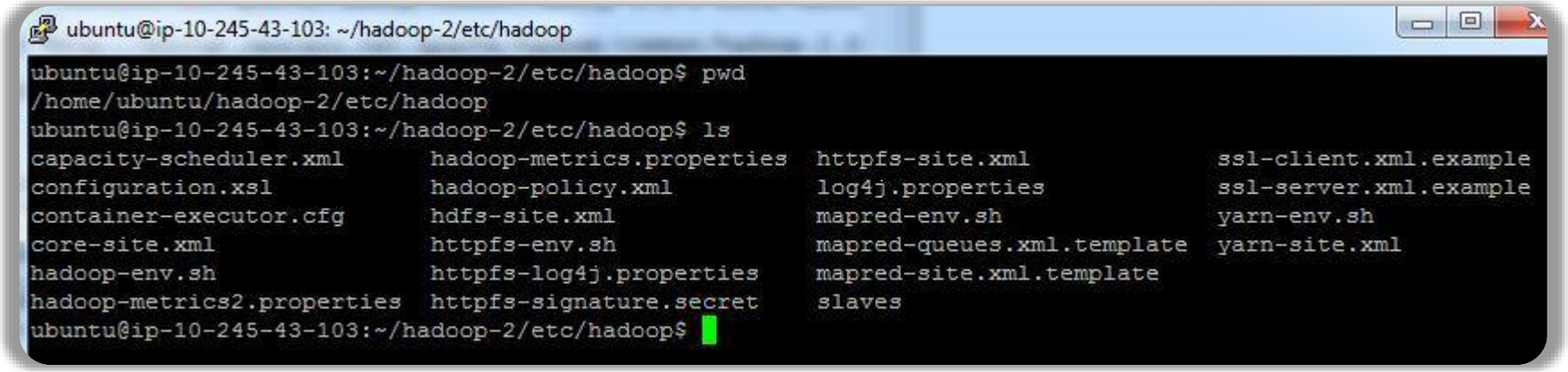
Hadoop 2.0 Configuration Files

edureka!

Configuration Filenames	Description of Log Files
hadoop-env.sh yarn-env.sh	Settings for Hadoop Daemon's process environment.
core-site.xml	Configuration settings for Hadoop Core such as I/O settings that common to both HDFS and YARN.
hdfs-site.xml	Configuration settings for HDFS Daemons, the Name Node and the Data Nodes.
yarn-site.xml	Configuration setting for Resource Manager and Node Manager.
mapred-site.xml	Configuration settings for MapReduce Applications.
slaves	A list of machines (one per line) that each run DataNode and Node Manager.

Hadoop 2.0 Configuration Files

edureka!



Deprecated Properties

edureka!

The core functionality and usage of these core configuration files are same in Hadoop 2.0 and 1.0 but many new properties have been added and many have been deprecated.

For example:

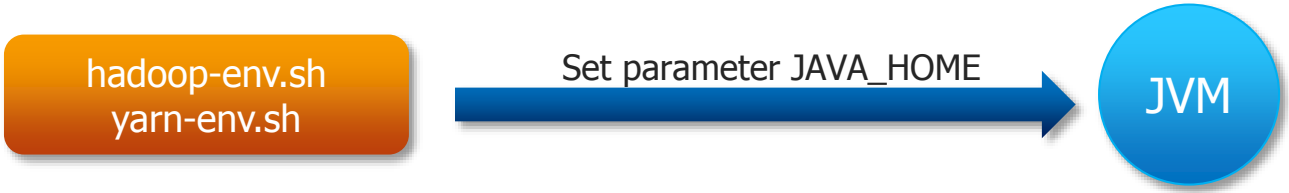
- ✓ 'fs.default.name' has been deprecated and replaced with 'fs.defaultFS' for YARN in core-site.xml
- ✓ 'dfs.nameservices' has been added to enable NameNode High Availability in hdfs-site.xml

Deprecated Property Name	New Property Name
dfs.data.dir	dfs.datanode.data.dir
dfs.http.address	dfs.namenode.http-address
fs.default.name	fs.defaultFS

- ✓ In Hadoop 2.2.0 release, you can use either the old or the new properties.
- ✓ The old property names are now deprecated, but still work!

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/DeprecatedProperties.html

Runtime Environment

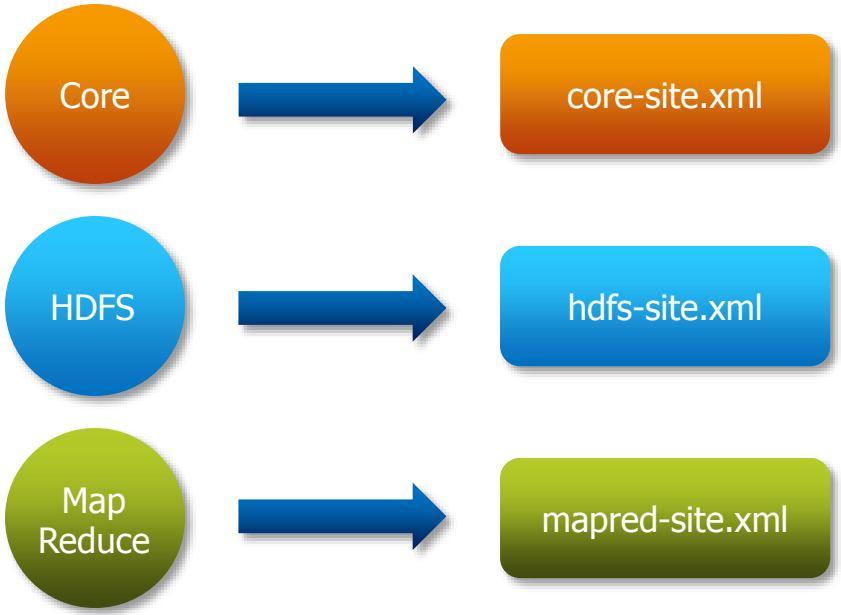


- ✓ Offers a way to provide custom parameters for each of the servers.
- ✓ Sourced by the Hadoop Daemons start/stop scripts.
- ✓ **Examples of environment variables that you can specify:**

HADOOP_DATANODE_HEAPSIZE
YARN_HEAPSIZE

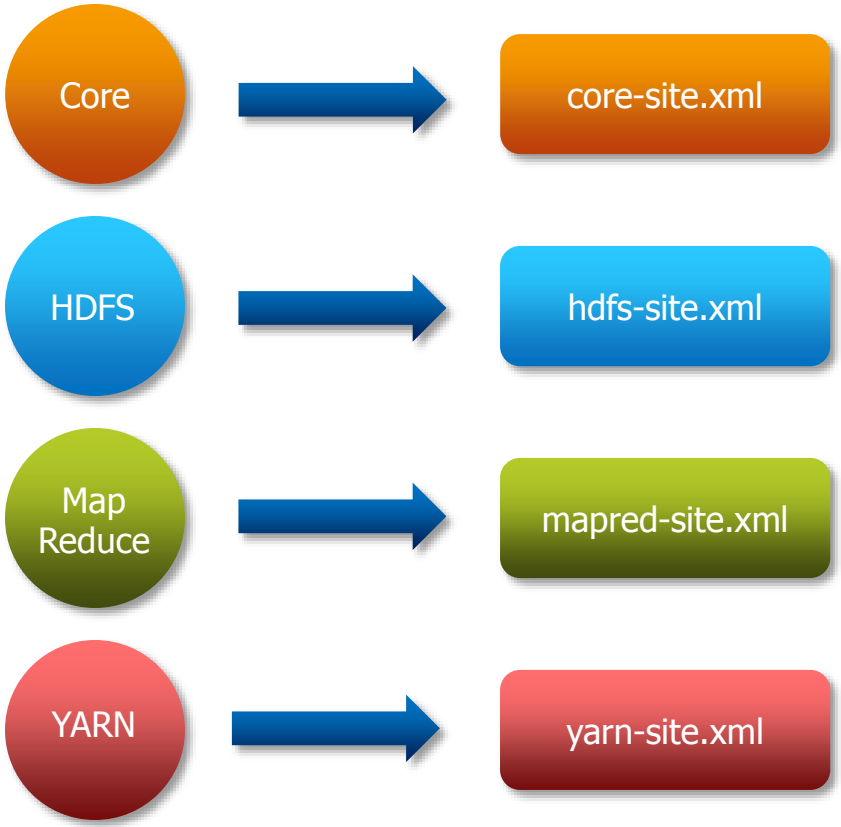
Hadoop 1.0: Core Configuration Files

edureka!



Hadoop 2.0: Core Configuration Files

edureka!



Hadoop 2.0: core-site.xml and hdfs-site.xml

edureka!

hdfs-site.xml	core-site.xml
<?xml version - "1.0"?>	<?xml version ="1.0"?>
<!--hdfs-site.xml-->	<!--core-site.xml-->
<configuration>	<configuration>
<property>	<property>
<name>**dfs.replication**</name>	<name>**fs.defaultFS**</name>
<value>**1**</value>	<value>**hdfs://test.abc.in:8020/** </value>
</property>	</property>
</configuration>	</configuration>

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/core-default.xml

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml

Hadoop 2.0: mapred-site.xml

edureka!

mapred-site.xml
<?xml version="1.0"?>
<configuration>
<property>
<name>**mapreduce.jobhistory.address**</name>
<value>**test.abc.in:10020**</value>
<property>
</configuration>

http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml

Notice difference in URL for current and stable release

http://hadoop.apache.org/docs/stable/mapred_tutorial.html

Hadoop 2.0: yarn-site.xml

edureka!

yarn-site.xml
<?xml version="1.0"?>
<configuration>
<property>
<name>**yarn.resourcemanager.address**</name>
<value>**test.abc.in:8021**</value>
<property>
</configuration>

http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-common/yarn-default.xml

Hadoop 2.0: Slaves

edureka!

Slaves

- ✓ Contains a list of slave hosts, one per line, that are to host **DataNode** and **Node Manager** servers.

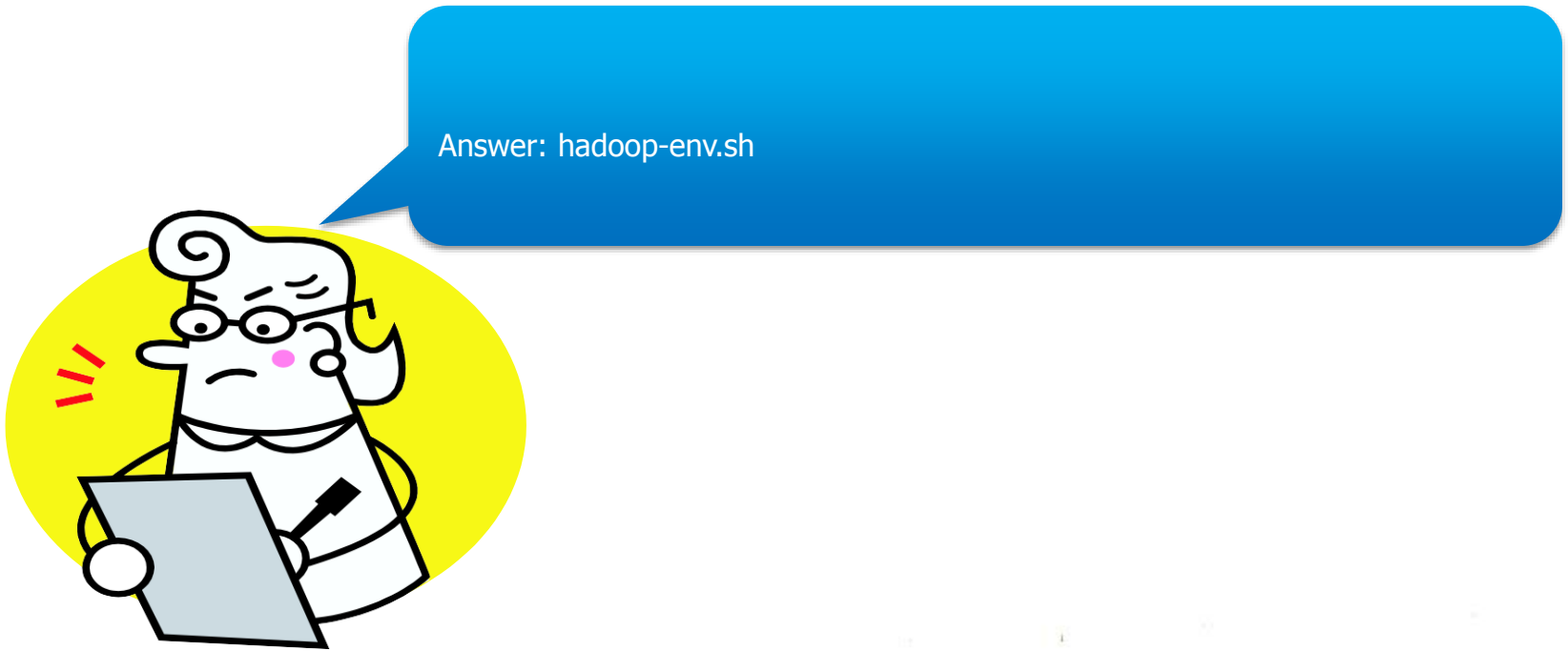
Annie's Question

edureka!

Environment variables that are used in the Hadoop start-up scripts can be configured in:

a) core-site.xml
b) hadoop-env.sh
c) hdfs-site.xml

Annie's Answer



Annie's Question

edureka!

We can configure setting for Resource Manager and Node Manager in:

a) HDFS
b) yarn-site.xml
c) core-site.xml
d) yarn.env-site.xml

Annie’s Answer

edureka!

Answer: yarn-site.xml

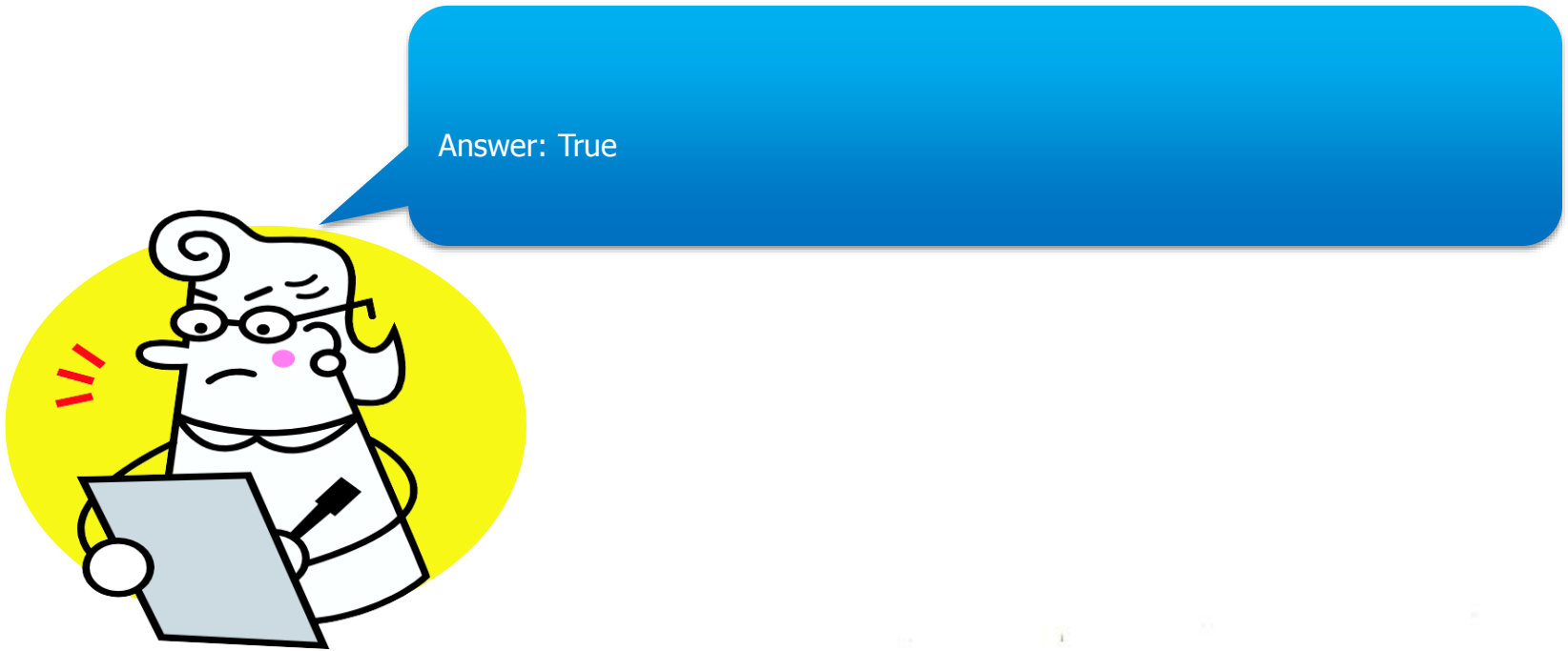
Annie's Question

edureka!

In Hadoop 2.2.0 release, we can use either the old or the new properties.

a) True
b) False

Annie’s Answer

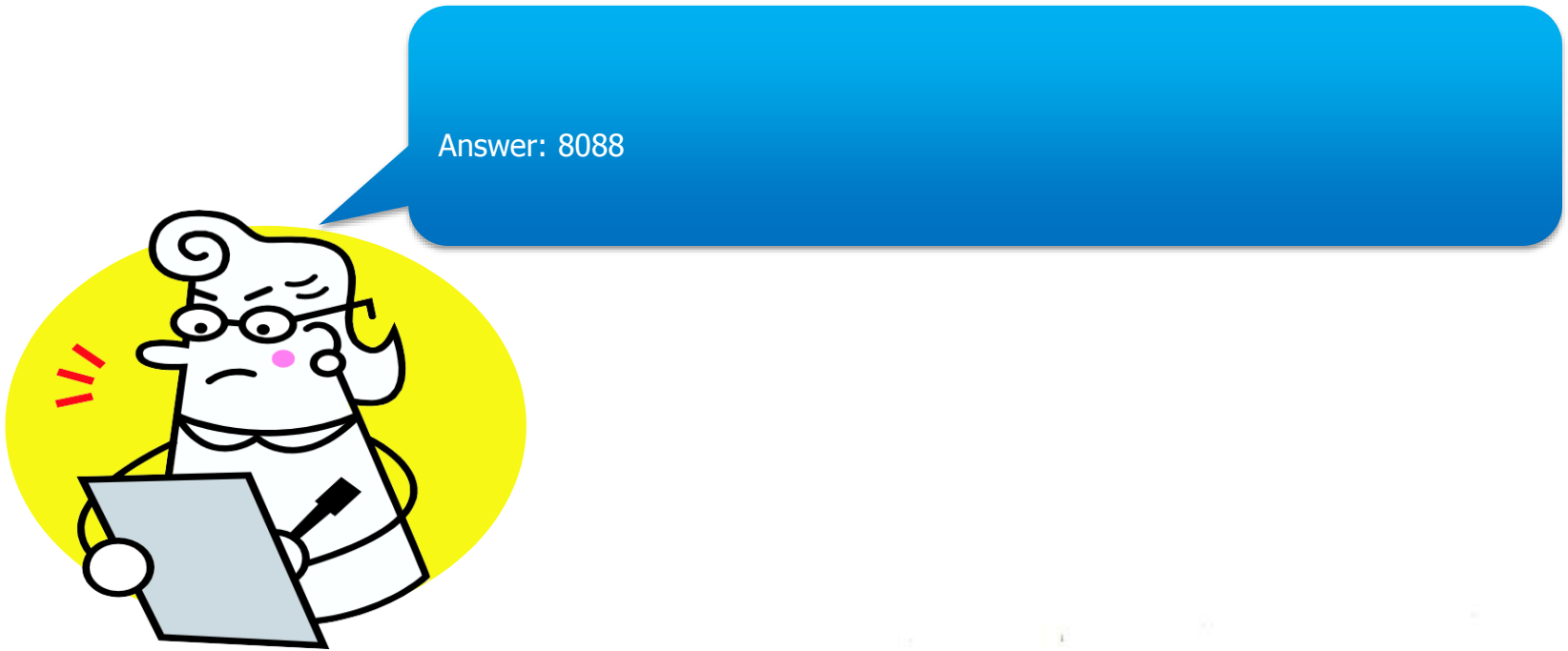


Annie's Question

edureka!

Default HTTP port for Resource Manager is:
a) 8088
b) 8080
c) 19088

Annie’s Answer



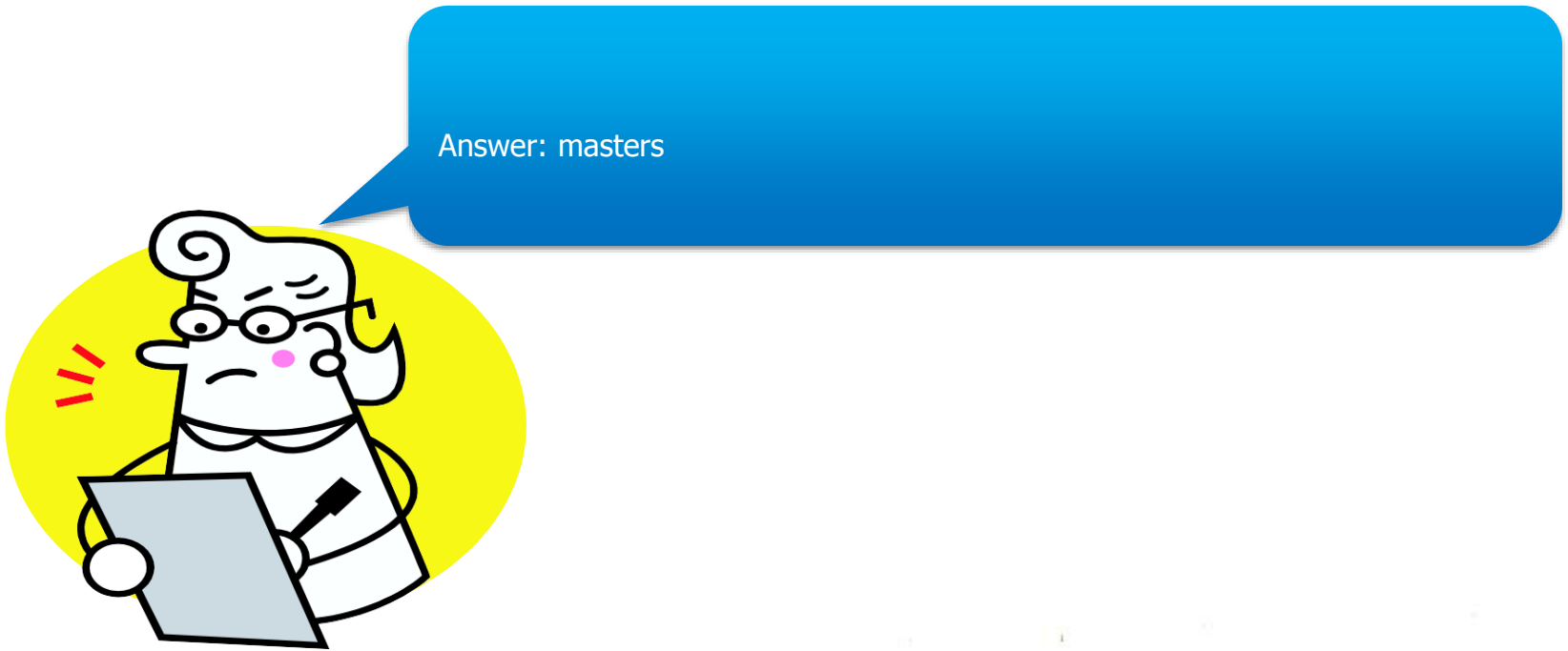
Annie's Question

edureka!

Which of the following file contains a list of machines (one per line) that each run a secondary Name Node?

a) masters
b) slaves

Annie’s Answer



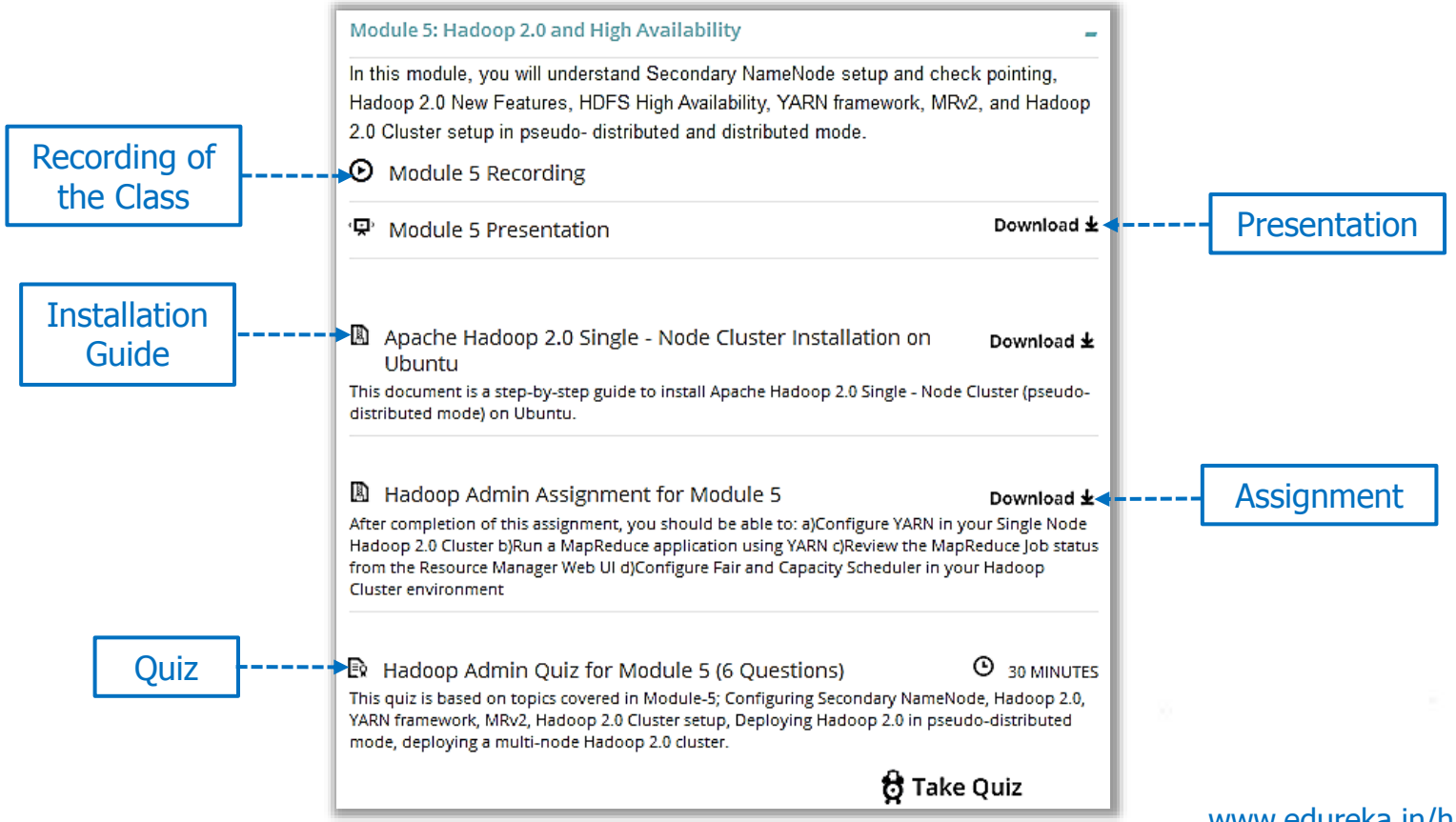
Assignments

edureka!

Tasks for you

- **Attempt the following Assignments using the documents present in the LMS:**
  - Install single-node Apache Hadoop 2.0 using a Virtual Machine in VMPlayer or VirtualBox.
  - Configure YARN in your Single Node Hadoop 2.0 Cluster.
  - Run a MapReduce application using YARN.
  - Review the Job status form the resource manager Web UI.
  - Configure Fair and Capacity Scheduler in your Virtual Cluster Environment.

What's Within the LMS?



edureka!

Thank You

See You in Class Next Week