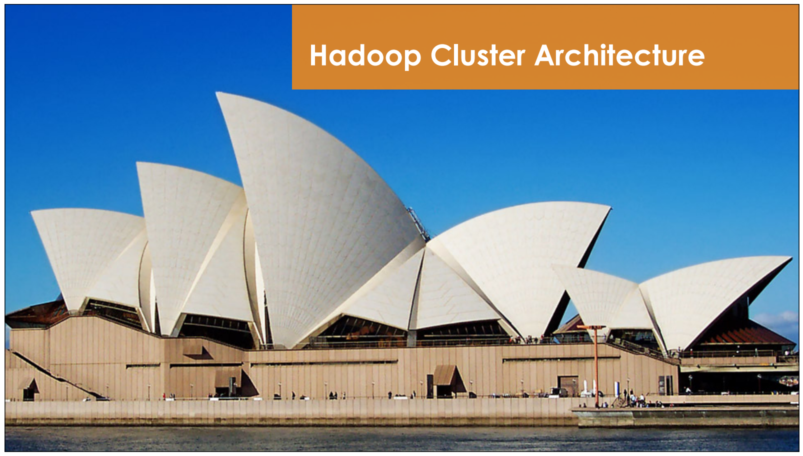
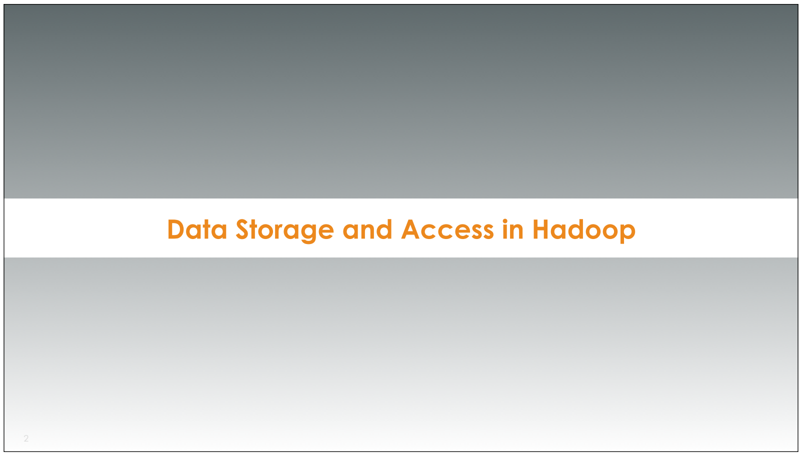


Let's dive into some basic Hadoop architecture



We're going to start with Data Storage and Access in Hadoop. We'll begin by asking a question: What is a cluster?

Hadoop Definitions

A *cluster* is a
- collection of independent servers
- connected by a network
- managed by software as one system

We define a cluster as a collection of independent servers <<click>
connected by a network <<click>>
managed by software as one system <<click>

Now it turns out we can drill down on this definition to define the components of a cluster.

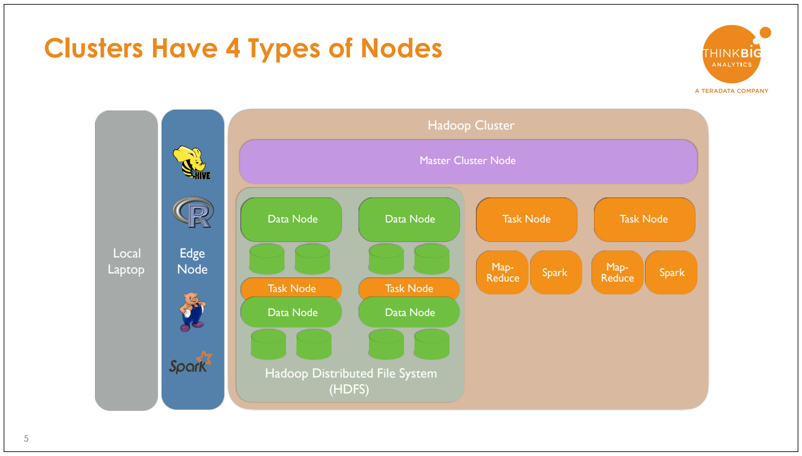
Hadoop Definitions: Nodes

Term	Definition
node	A generic server with processors, memory, and optionally disks. A collection of these make up a cluster.
master node	The node that directs the rest of the cluster nodes. Typically this node runs programs such as YARN Application Manager and the Name Server.
edge node	A node that connects to both the cluster and data center networks. Typically, this node will run applications such as Hive and Pig. The master node may play the role of an edge node.
worker or core node	All non-master nodes are usually worker nodes that execute Hadoop tasks.
data node	A core node that has disks and participates in HDFS
task node	A core node that runs parallelized application tasks

Clusters are made up of nodes, as we just said. So what's a node? It turns out there are many types. Let's walk through each one.

....

A picture will help illustrate this on the next slide.



We start over on the left with whatever local computer you are using. Here's we've called it your local laptop.

That usually connects to an Edge node, which is a node connected to both the cluster and the outer network.

The cluster itself has one or more master nodes, where most of the cluster services run.

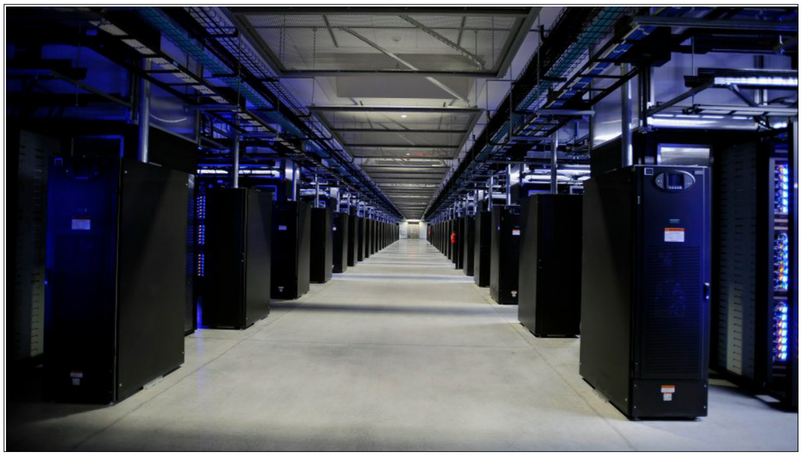
The actual work is done by worker nodes of various types, depending on their hardware and the service modules configured on them.

Data nodes have disks attached, while task nodes run executors such as MapReduce and Spark.

Nodes can be both Data and Task nodes at once.

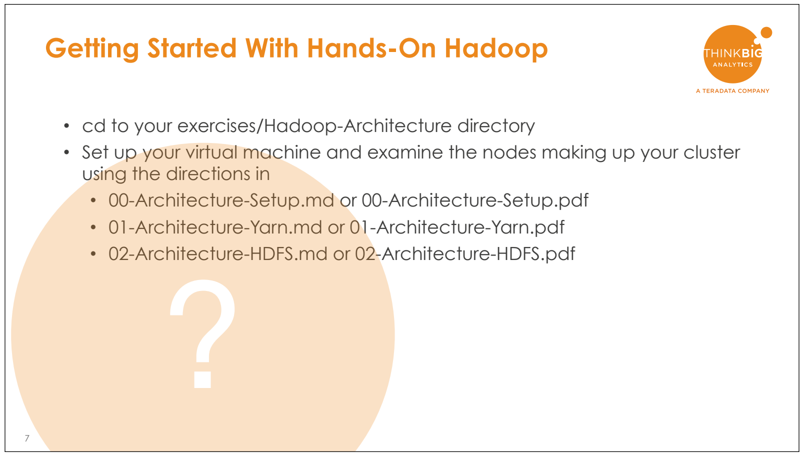
All the data nodes participate in the cluster file system, referred to as the Hadoop Distributed File System.

That's a logical diagram of a cluster. What does a real one look like?



Well here's a good sized cluster. This one resides at Facebook. Facebook has some very large clusters, although Google, Yahoo, LinkedIn and others continually compete for who is running the largest cluster currently.

So let's get you set up with your own clusters.



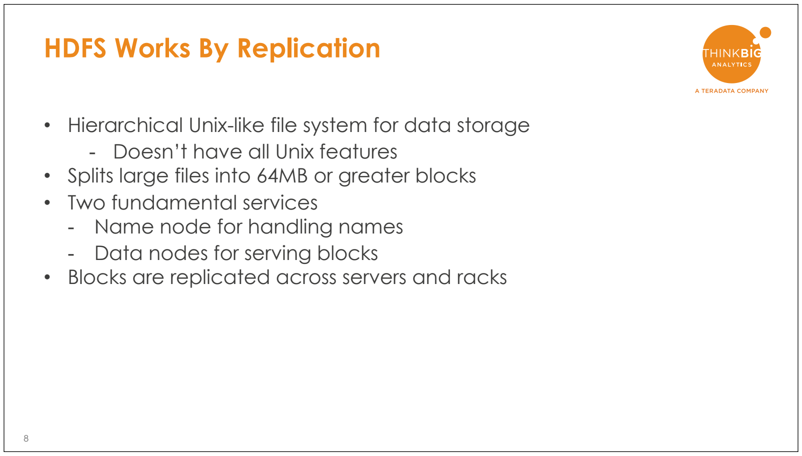
We now going to perform three labs which will get you used to connecting to your clusters. You should have a directory with all the course materials called sparkclass. Underneath that directory are several other directories including

data
exercises
handouts
images
slides

You'll find these labs in your exercises/Hadoop-Architecture directory.

You can read these labs in either pdf or Markdown form. The PDF form is prettier, but the text in the Markdown version will be easier to copy and paste when you get to complicated commands.

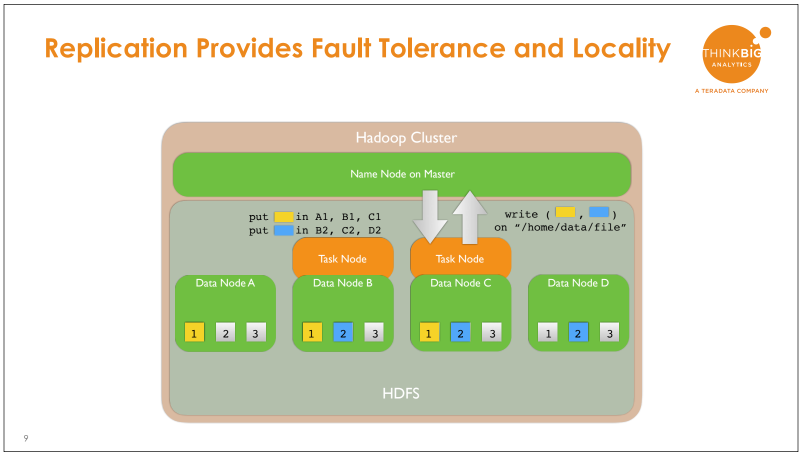
Go ahead and run those three labs. We'll allot about 15 minutes for you to do this.



At this point, you'll now have some experience with Yarn and HDFS. Let's tackle what HDFS is about.

On this last bullet, this is an illustration of spending resources to save resources. We replicate blocks not only for reliability, but also for speed of access. It's going to be faster for a node to access a data block the node requesting it is close to the node with the data. As a result, you can gain performance if you replicate your data more. Said another way, you can spend resources to save resources and time.

Let's look at how this works.



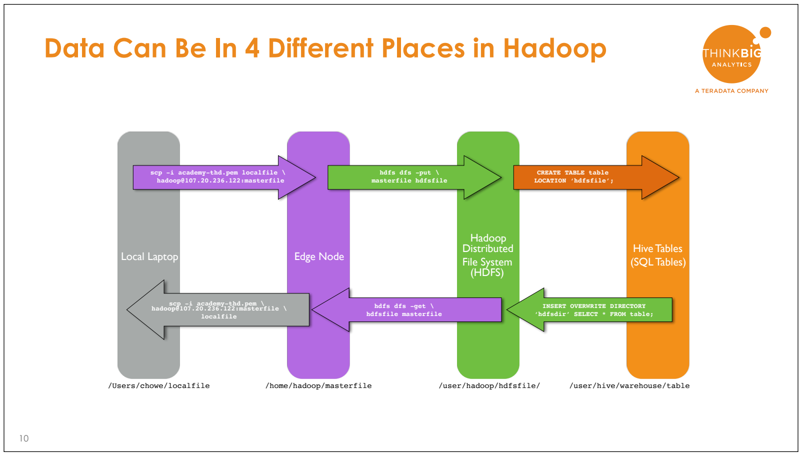
Here we have a five node cluster with one master node <click> and four data nodes.

Over on the right, an application on one of the task nodes asks to write two HDFS blocks, a yellow one and a blue one. That request gets sent to the Name Node. <click>

Unlike in most operating systems, the name node doesn't write that block. Instead, it provides block addresses that specify where the original application should write the blocks. In this case, it says that the yellow block should be written on Data Node A, block 1, Data Node B block 1 and Data Node C block 1. <click>

Meanwhile, the blue block is to be written on Data Node B block 2, Data Node C block 2, and Data Node D block 2.

Once those blocks have been written, we can lose any of the data nodes and still have access to both yellow and blue blocks. In fact, we can lose two nodes and still retain all of the data we wrote.



One of the challenges of working with Hadoop is that data can live in 4 different places, all of which look rather similar because they are all Unix-like file systems and have Unix-like names

Data can live on

- your laptop, such as in /Users/chowe/localfile
- an Edge node, in a location like /home/hadoop/masterfile
- on HDFS in a location like /user/hadoop/hdfsfile/
- or in the Hive warehouse in a location like /user/hive/warehouse/table

What's difficult is we use different commands to move files among those locations.

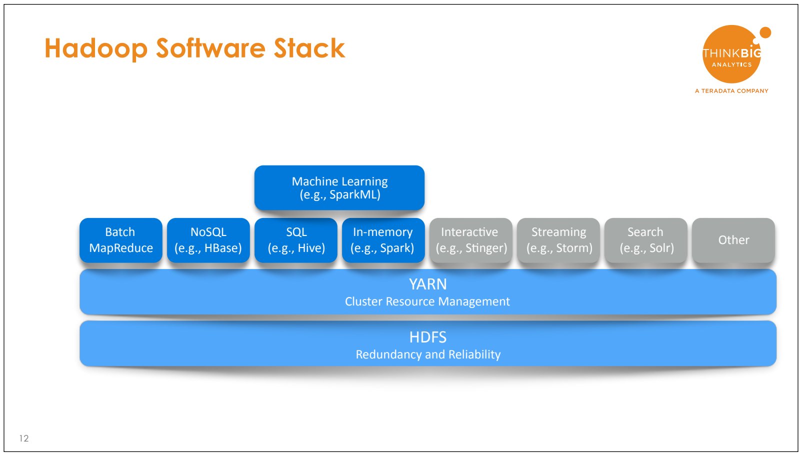
1.To move a file from my local laptop to the edge node, I'd type something like
scp -i academy-thd.pem localfile hadoop@107.20.236.122:masterfile <click>
2. To move a file from the edge node to HDFS, you'd use a command like
hdfs dfs -put masterfile hdfsfile <click>
3. and finally to move a file from HDFS into the Hive warehouse, we'd use a SQL command like
CREATE TABLE table LOCATION 'hdfsfile'; <click>

We use similar but different commands to bring the data back.
4. To move data from the Hive warehouse to HDFS, we'd do something like
INSERT OVERWRITE DIRECTORY 'hdfsdir' SELECT * FROM table; <click>

Hadoop Terminal Commands

Shell Command	Meaning
hdfs dfs -ls dest	List files in directory dest
hdfs dfs -put localfile hdfsfile	Copy file from master node local file system to HDFS
hdfs dfs -get hdfsfile localfile	Copy file from HDFS to local file system
hadoop distcp srcURL destURL	Recursively copy the directory specified by srcURL to the directory indicated by destURL using MapReduce
yarn jar jarfile args	Run Java program jarfile with arguments args

The first three commands are fairly self-explanatory for listing and moving files. Hadoop distcp is a distributed copy command -- it actually runs a MapReduce job to do the copy, thereby speeding up the movement of large amounts of data by employing the entire cluster to do the moving. Finally, we can have the cluster run Java code by specifying a JAR file to the YARN JAR command.

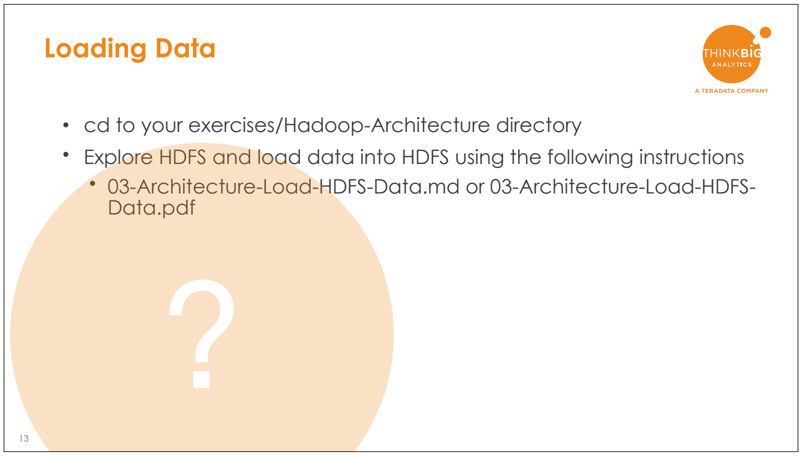


These two components we've discussed in this module—HDFS and YARN—make up the heart of every Hadoop cluster.

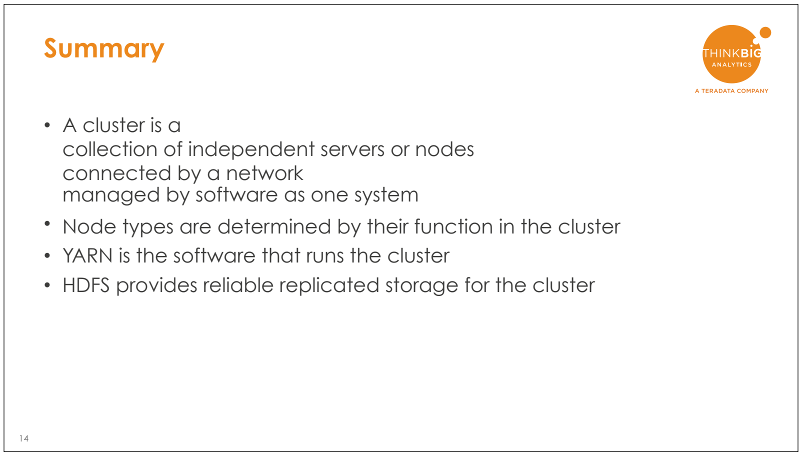
HDFS gives us a redundant and reliable data store for big data through replication. <click>
Yarn provide us with overall cluster resource management, just as any operating system would. <click>

All the other projects in Hadoop—things like NoSQL databases such as HBase, SQL query engines like Hive, In-memory execution systems like Spark, and even Machine Learning packages like SparkML—run as applications on top of this framework. <click>

Hadoop has more than 50 different application frameworks (and more pop up all the time), so we can't discuss all of them. However, we will discuss the ones in dark blue as part of this course.



Before we start getting into Hadoop applications though, we will need to import our datasets into HDFS. So let's return to our the exercises/Hadoop-Architecture directory and run Lab 03 to load our data into HDFS.



In summary,

A cluster is a
collection of independent servers or nodes
connected by a network
managed by software as one system

Node types are determined by their function in the cluster

YARN is the software that runs the cluster

HDFS provides reliable replicated storage for the cluster