MITS6005

Big Data

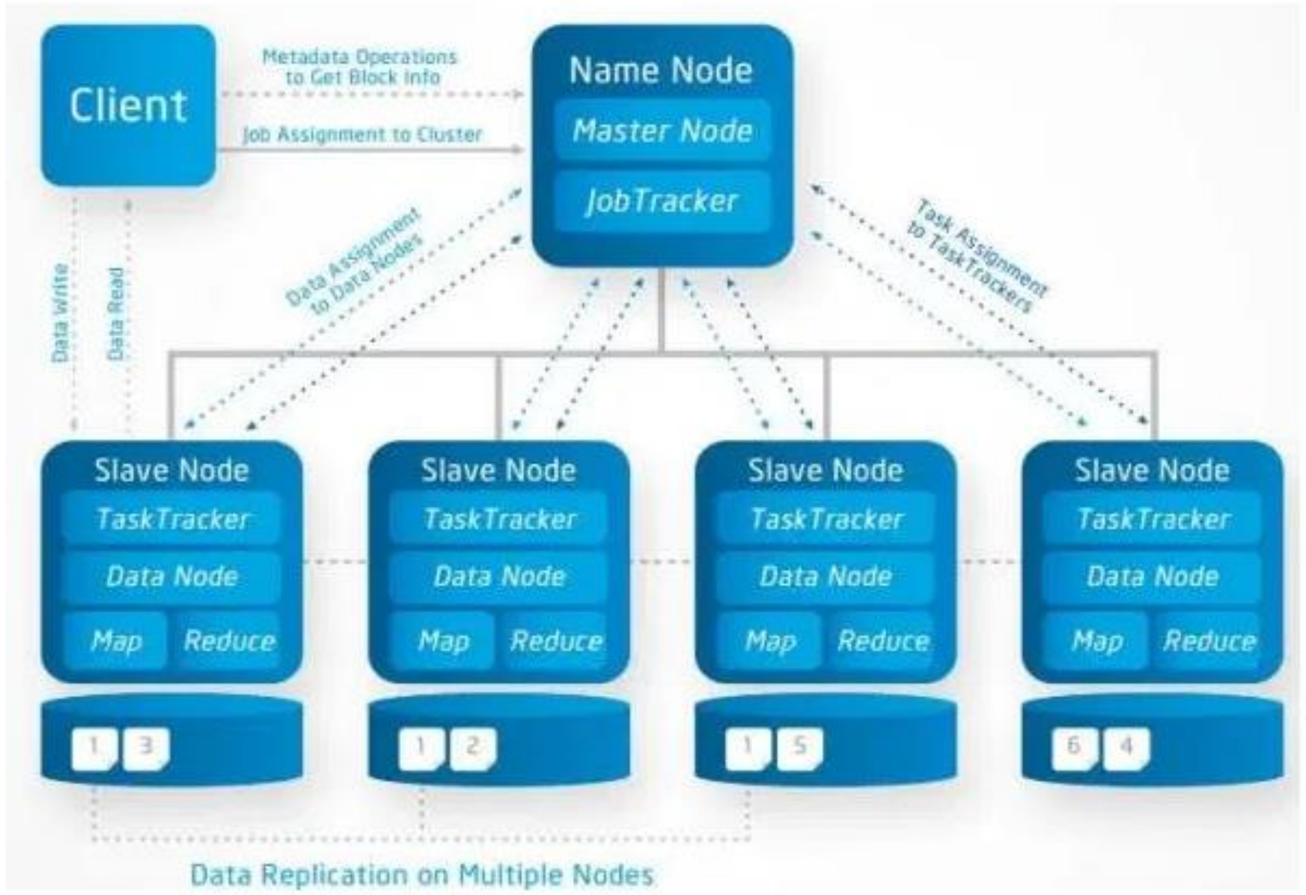
Session 5a

Hadoop Cluster

What is Hadoop Cluster

What is Hadoop Cluster?

- A Hadoop cluster is a special type of computational cluster designed specifically for storing and analyzing huge amounts of unstructured data in a distributed computing environment.
- A group of distributed computers working together
- It basically has two **Master** (Name Node and JobTracker) and numerous number of **Slaves** (Data Node and TaskTracker)**.**
- Master nodes supervise and manage the work; assigns the tasks to the Slaves
- Slave nodes (data nodes) do the actual work



Client
Metadata Operations to Get Block Info
Job Assignment to Cluster
Name Node
Master Node
JobTracker
Data Write
Data Read
Data Assignment to Data Nodes
Task Assignment to TaskTrackers
Slave Node
TaskTracker
Data Node
Map
Reduce
1
3
2
5
6
4
Data Replication on Multiple Nodes

Hadoop Cluster - Advantages

Some of the major **Advantages** are as follows:

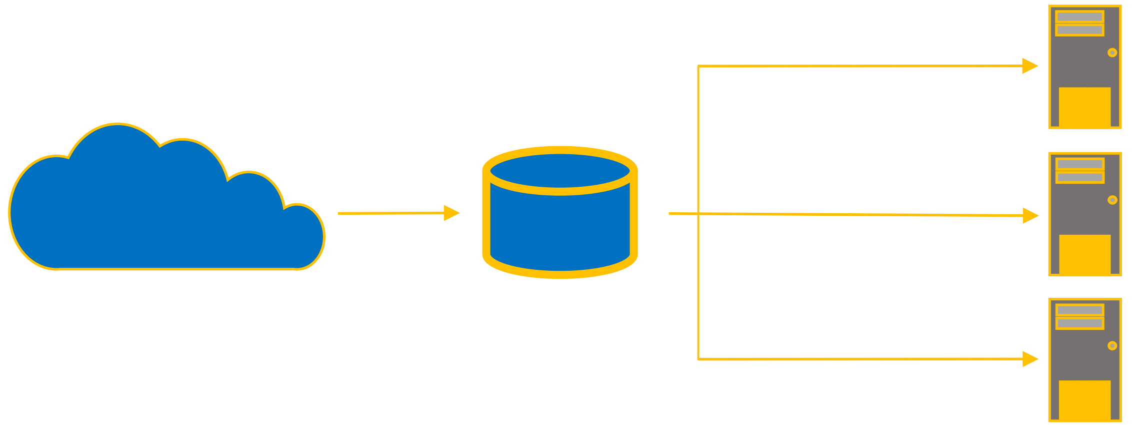
- **Scalable:** One can scale a Hadoop cluster by adding new servers to the cluster if needed.
- **Cost-effective:** It is inexpensive.
- **Flexible:** Deal with data from many sources and formats in a very quick, easy manner.
- **Fast:** The cluster helps in increasing the speed of the analysis process.
- **Resilient to failure:** clusters are failure resilient.

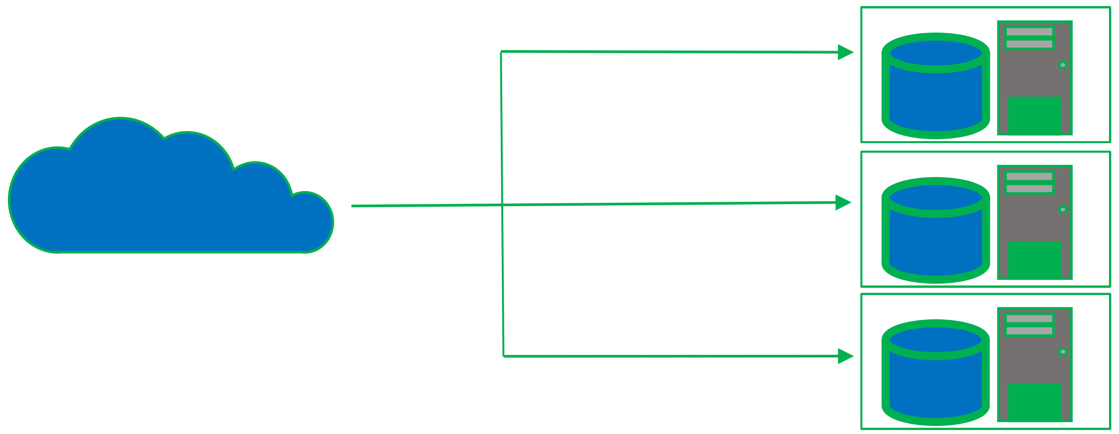
It is possible to deploy Hadoop using a single-node installation, for evaluation purposes.

Distributed File System (DFS)

- Traditional DFS store data in a central location
- Data is copied to processors at run time.
- Data copy and network traffic becomes bottleneck.

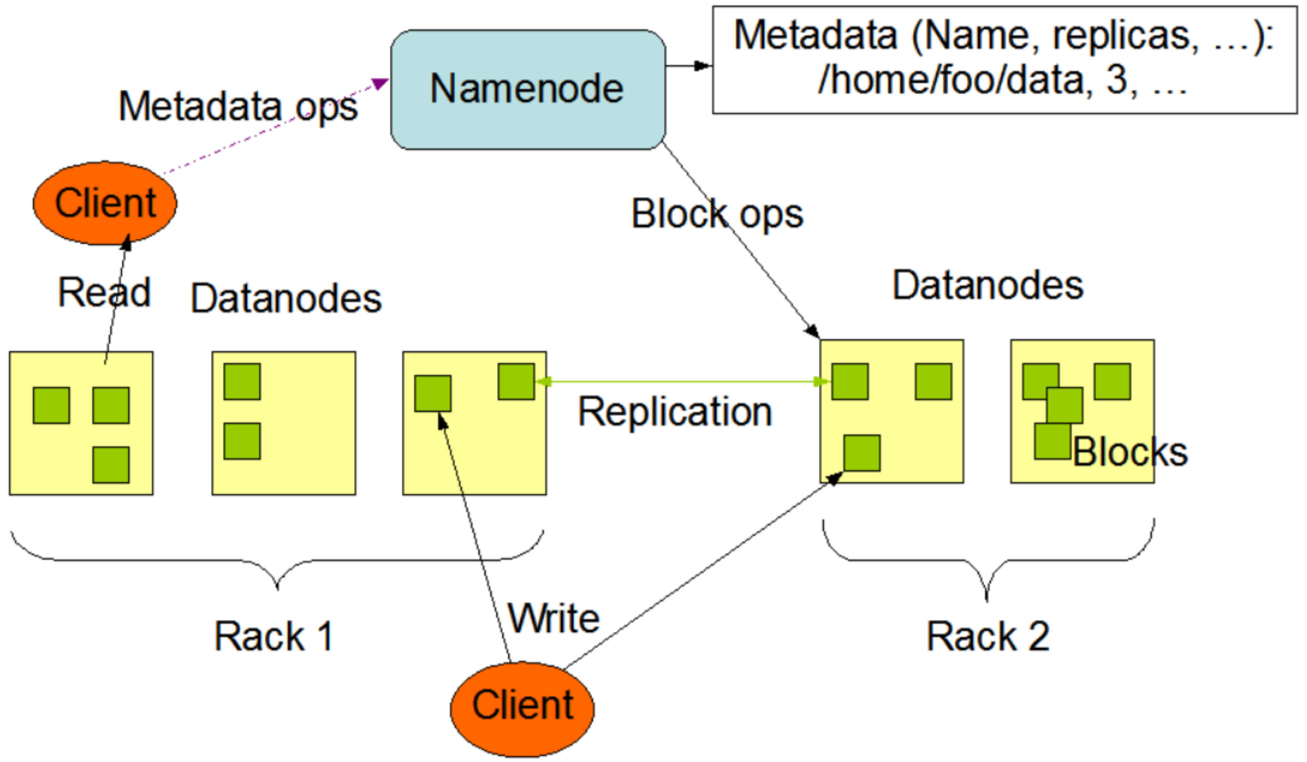
- Hadoop distributes data when data is stored.
- Replicates the data on multiple nodes
- Runs computation where the data is located





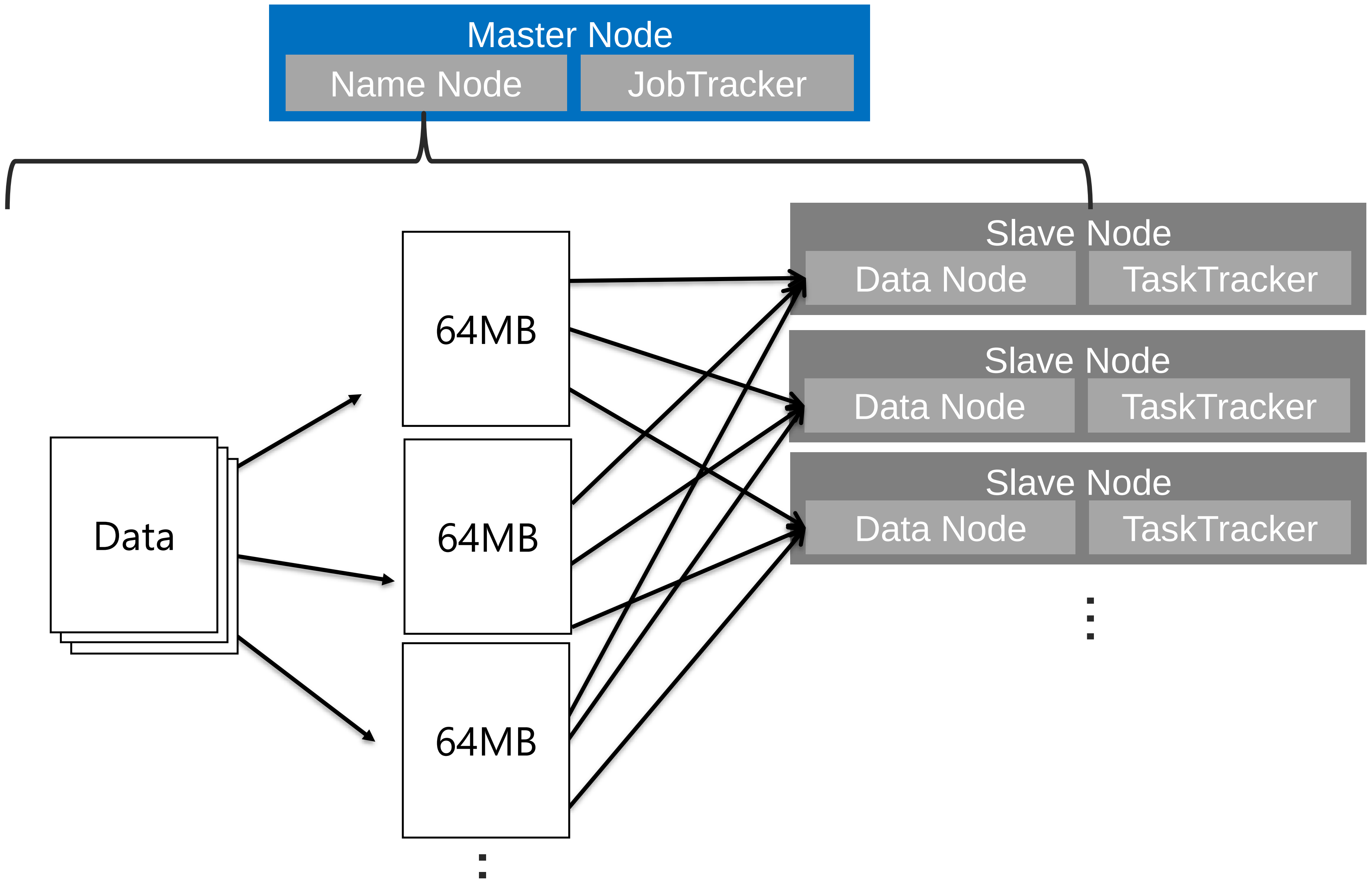
HDFS: Hadoop Distributed File System

- HDFS is the storage system in a Hadoop platform.
- Provides reliable and inexpensive storage.
- Built on top of native file systems.
- Performs best with modest number of large files. rather than many small files.
- Files are ‘write once’.
- Highly fault-tolerant and designed to be deployed on low-cost hardware.
- Provides high throughput access to application data.



https://hadoop.apache.org/docs/r3.2.0/hadoop-yarn/hadoop-yarn-site/YARN.htr

HDFS Diagram



HDFS shell commands

Hadoop file system shell commands are used to perform various Hadoop HDFS operations and in order to manage the files present on HDFS clusters

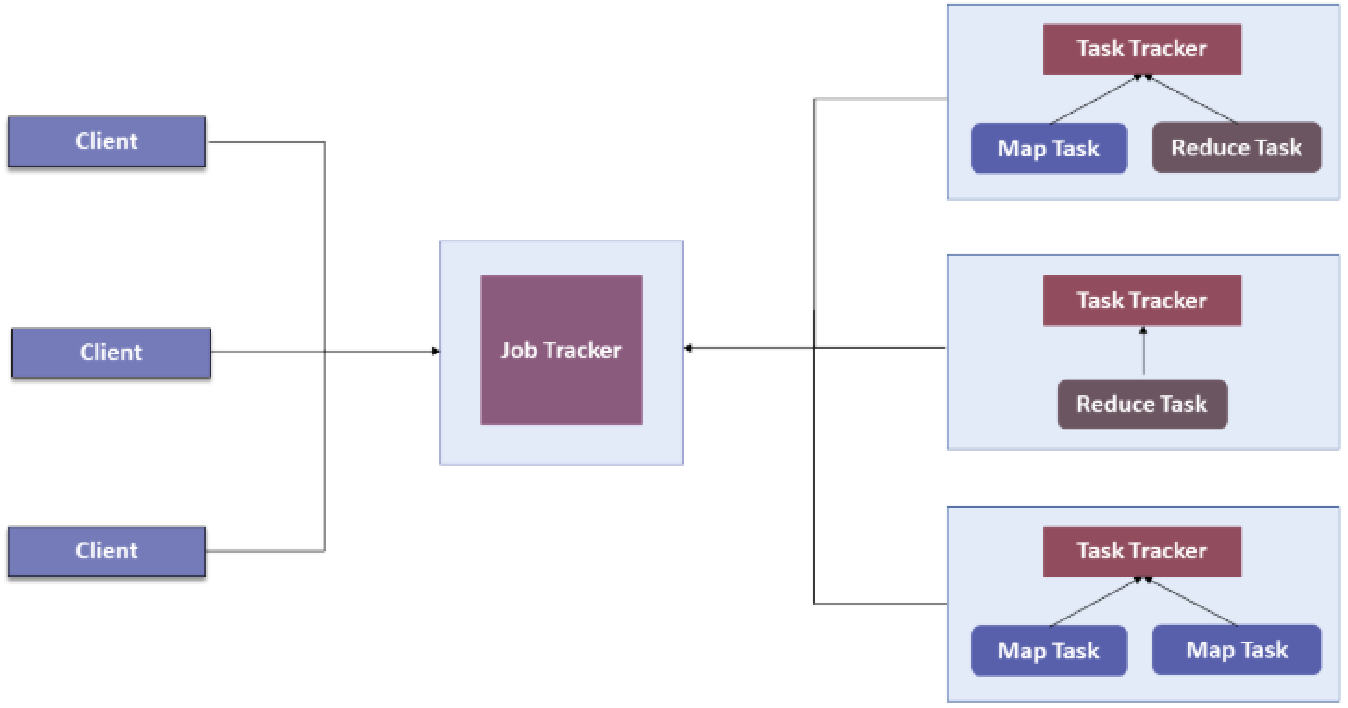
Command options:

- Version: e.g. hdfs dfs version
- ls (list): e.g. hdfs dfs -ls /user/dataflair/dir1
- cp and mv (copy and move): e.g. hadoop fs -cp /user/dataflair/dir2/purchases.txt /user/dataflair/dir1
- rm (remove)
- mkdir (make directory): e.g. hdfs dfs -mkdir /user/dataflair/dir1
- put and get (transfer files between local file system and HDFS)

  e.g. hdfs dfs -put /home/dataflair/Desktop/sample /user/dataflair/dir1
  e.g. hdfs dfs -get /user/dataflair/dir2/sample /home/dataflair/Desktop
- copyFromLocal: e.g. hdfs dfs -copyFromLocal /home/dataflair/Desktop/sample /user/dataflair/dir1
- copyToLocal: e.g hdfs dfs -copyToLocal /user/dataflair/dir1/sample /home/dataflair/Desktop

MapReduce (MRV1) - Architecture

- In Hadoop version 1.0 (referred as MRV1), MapReduce performed both processing and resource management functions.
- Job Tracker:
  - Single master
  - Manages cluster resources and job scheduling.
  - Assigns map and reduce tasks on a number of Task Trackers
- Task Tracker:
  - Manages tasks status on slave node. ask Trackers
  - periodically reported their progress to the Job Tracker.

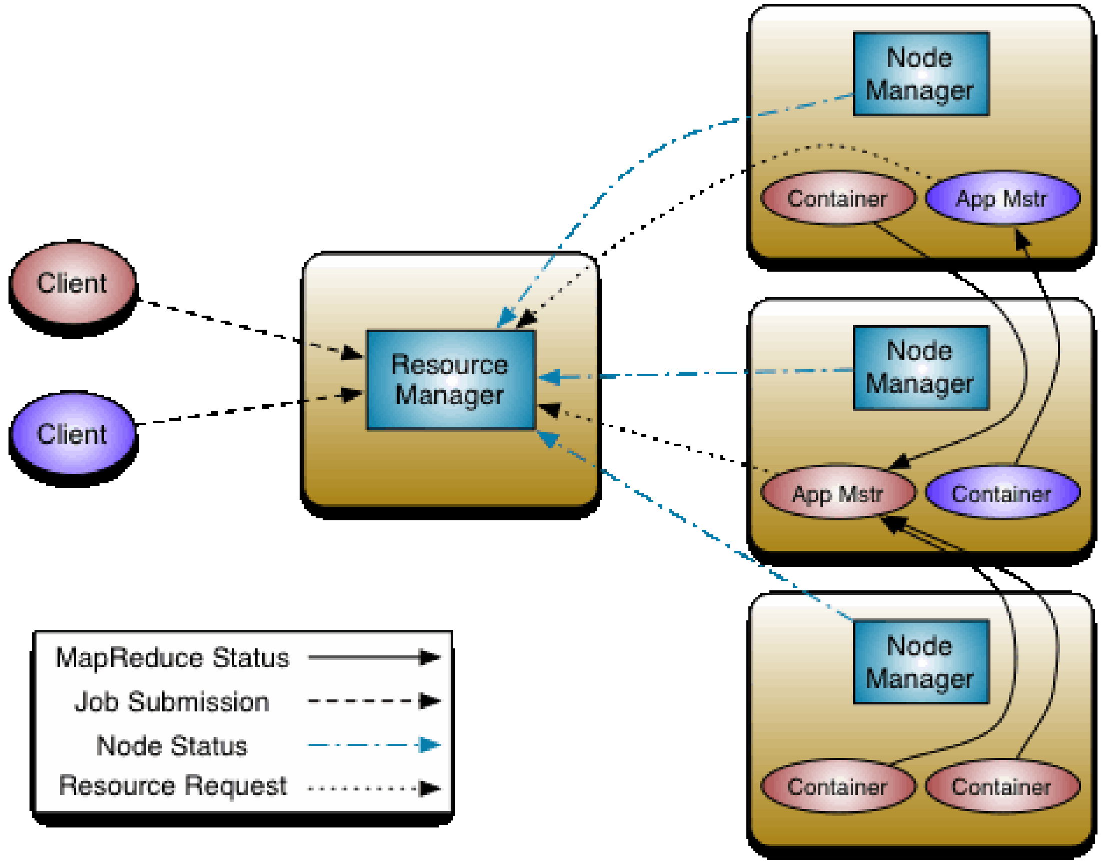


https://www.edureka.co/blog/hadoop-yarn-tutorial/

YARN : Yet Another Resource Negotiator

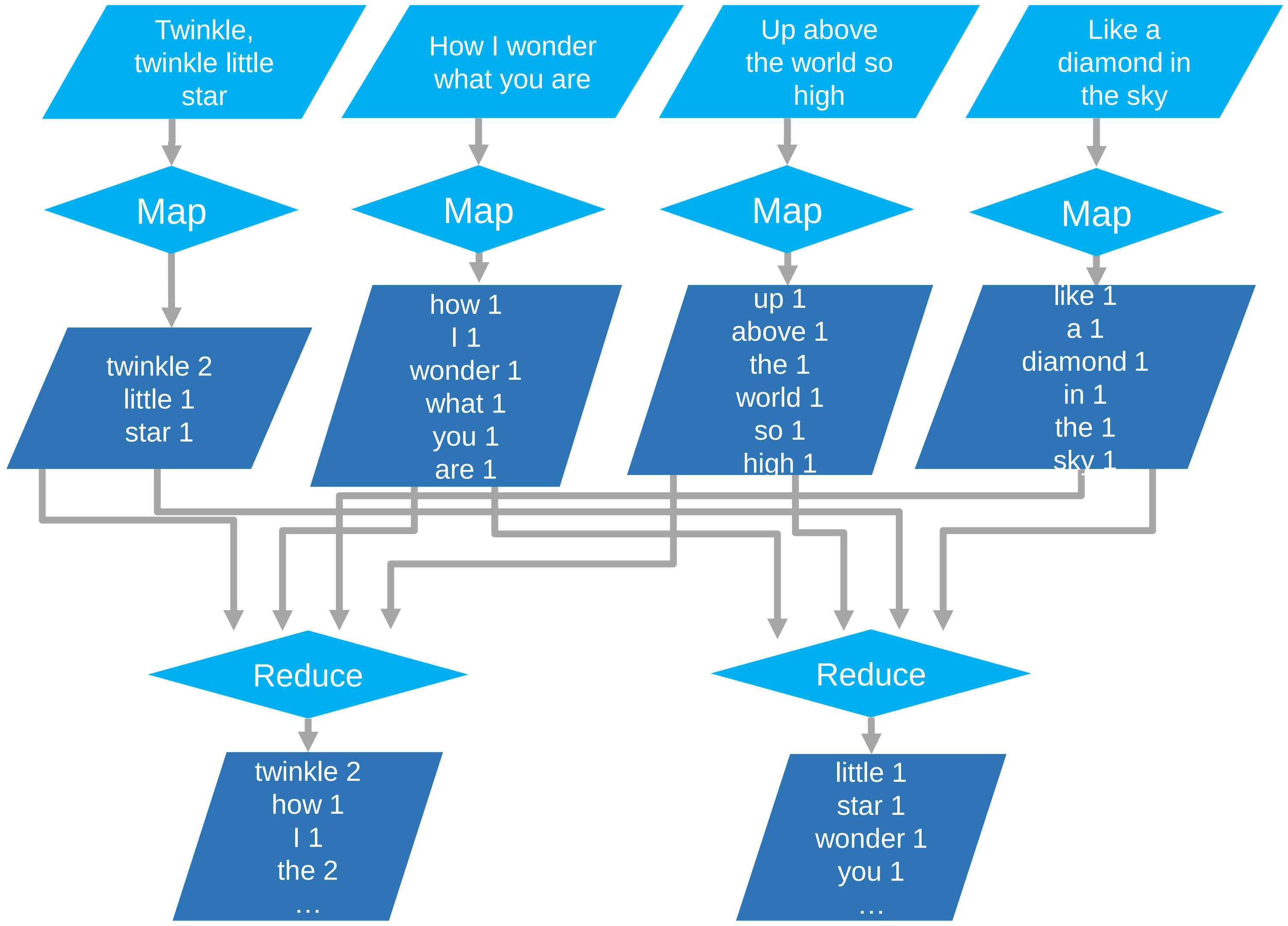
- Fundamental idea of YARN: to split up the functionalities of resource management and job scheduling/monitoring into separate daemons.
- Hadoop YARN offers a central platform that brings **security**, and **data governance tools**, as well as **resource management over Hadoop clusters**.

YARN components:
- **Resource Manager** (the master)
  - Scheduler
  - Application Manager
- **Node Manager** (the slave)
  - manages user jobs and workflow on the given node.
- **Application Master**
  - a single job submitted to the framework
- **Container**
  - collection of physical resources such as RAM, CPU cores, and disks on a single node



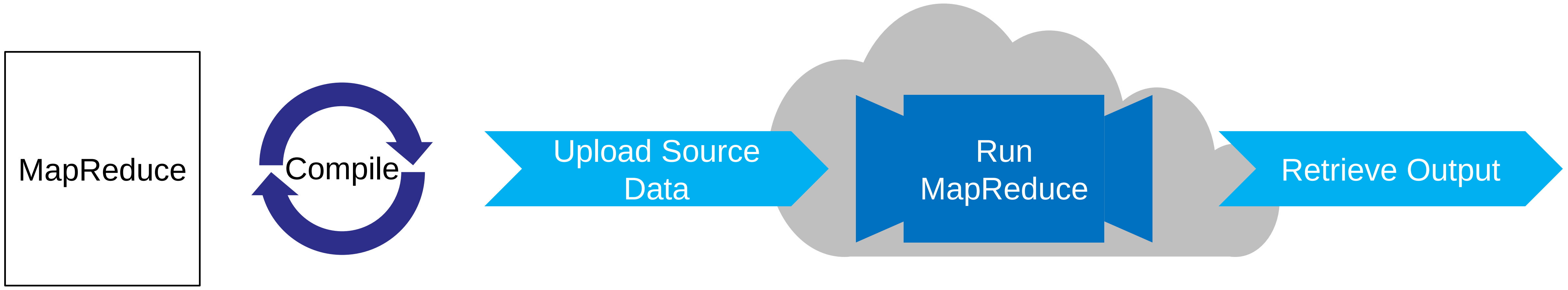
https://hadoop.apache.org/docs/r3.2.0/hadoop-yarn/hadoop-yarn-site/YARN.html

MapReduce: Basic Concept



Running a MapReduce Job

1. Compile executable MapReduce code
2. Upload Source data
3. Run MapReduce executable on cluster
4. Retrieve job output

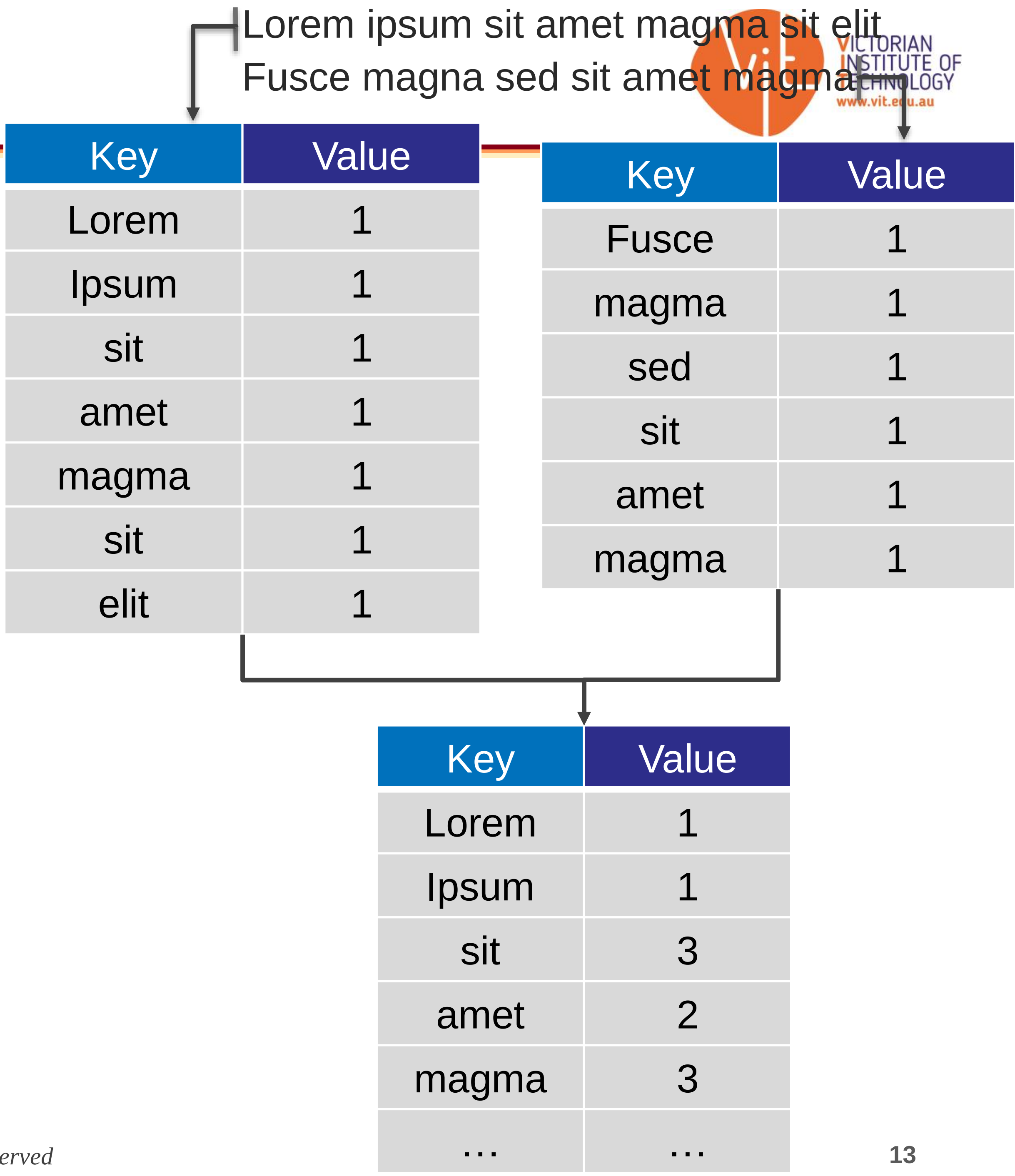


```
hadoop jar my.jar myclass /data/src /data/out
```

Microsoft Azure

Word Count

- Map Function
  - Generates input data to tuples (key-value) pairs
  - Map task runs in parallel among data nodes
    - creates key/value pairs with words as keys and placeholder values of 1



Lorem ipsum sit amet magma sit elit

Fusce magna sed sit amet magma

Key	Value
Lorem	1
Ipsum	1
sit	1
amet	1
magma	1
sit	1
elit	1

Key	Value
Fusce	1
magma	1
sed	1
sit	1
amet	1
magma	1

Key	Value
Lorem	1
Ipsum	1
sit	3
amet	2
magma	3
…	…

Word Count

Reduce Function

- reduce() combines those intermediate values into one or more final values for that same key
- Reduce phase aggregates values for each key by adding the values for each word
- reduce() functions also run in parallel, but can’t start until map phase is completely finished

Word Count (Java)

```
public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
       private final static IntWritable one = new IntWritable(1);
       private Text word = new Text();
       public void map(LongWritable key, Text value, Context context) {
           String line = value.toString();
           StringTokenizer tokenizer = new StringTokenizer(line);
           while (tokenizer.hasMoreTokens()) {
               word.set(tokenizer.nextToken());
               context.write(word, one);
           }
       }
}
public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {
       public void reduce(Text key, Iterable<IntWritable> values, Context context)
{
           int sum = 0;
           for (IntWritable val : values) {
               sum += val.get();
           }
       }
       context.write(key, new IntWritable(sum));
}
```

Word Count (compile and run)

```
                         //Compile:
C:\WINDOWS\system32\wordCountSample> Javac *.java
C:\WINDOWS\system32\wordCountSample> jar –cvf wordcount.jar *.class



   //Submit and Run the application:
C:\WINDOWS\system32\wordCountSample> hadoop jar wordcount.jar WordCount input-file  output
```

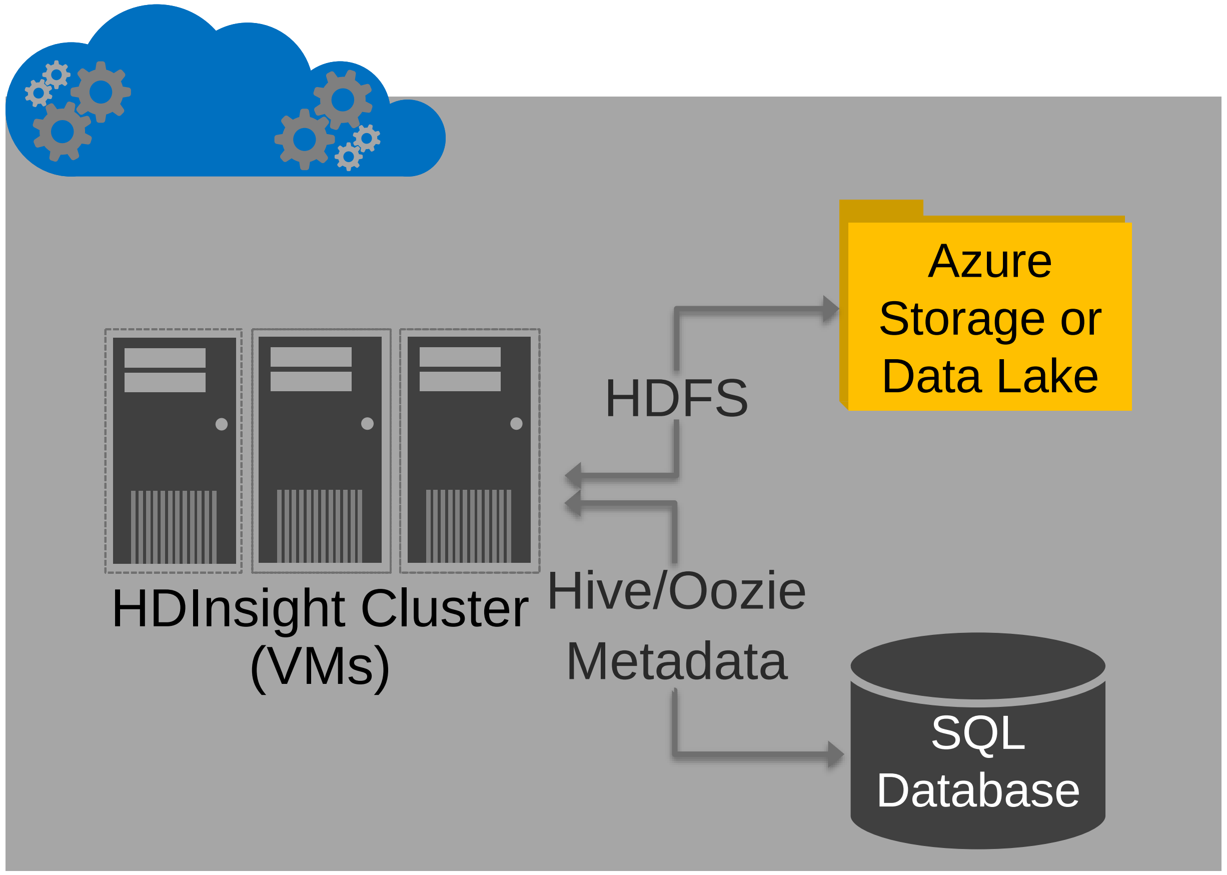
What is HDInsight?

HDInsight is the Microsoft implementation of Hadoop ecosystem components in the cloud.

Azure HDInsight brings the power of Hadoop to Azure to process Big Data.

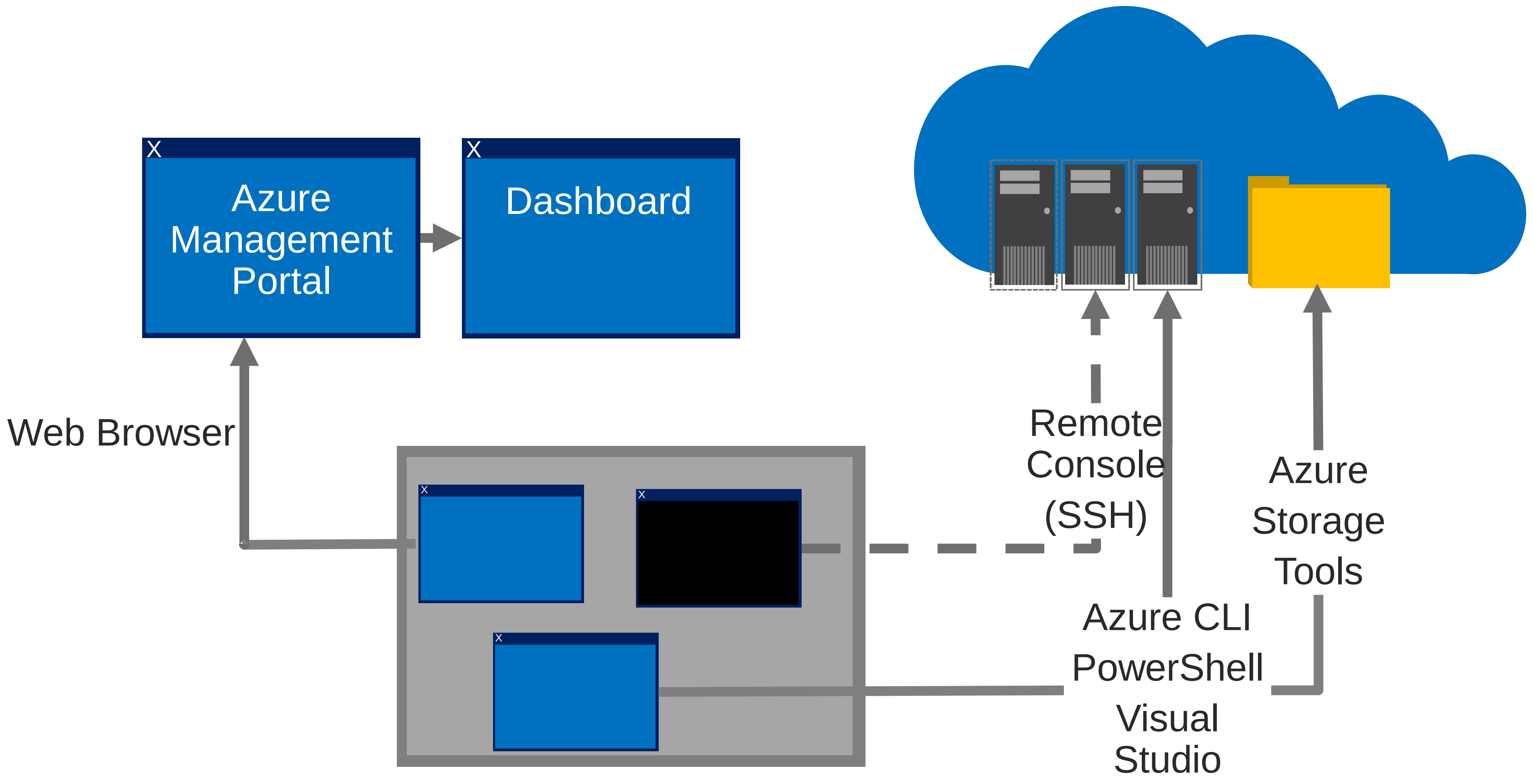
What comes with HDInsight?

Apache Hadoop	Apache Spark	Apache Kafka	
Apache HBase	Apache Hive LLAP	Apache Storm	Machine Learning



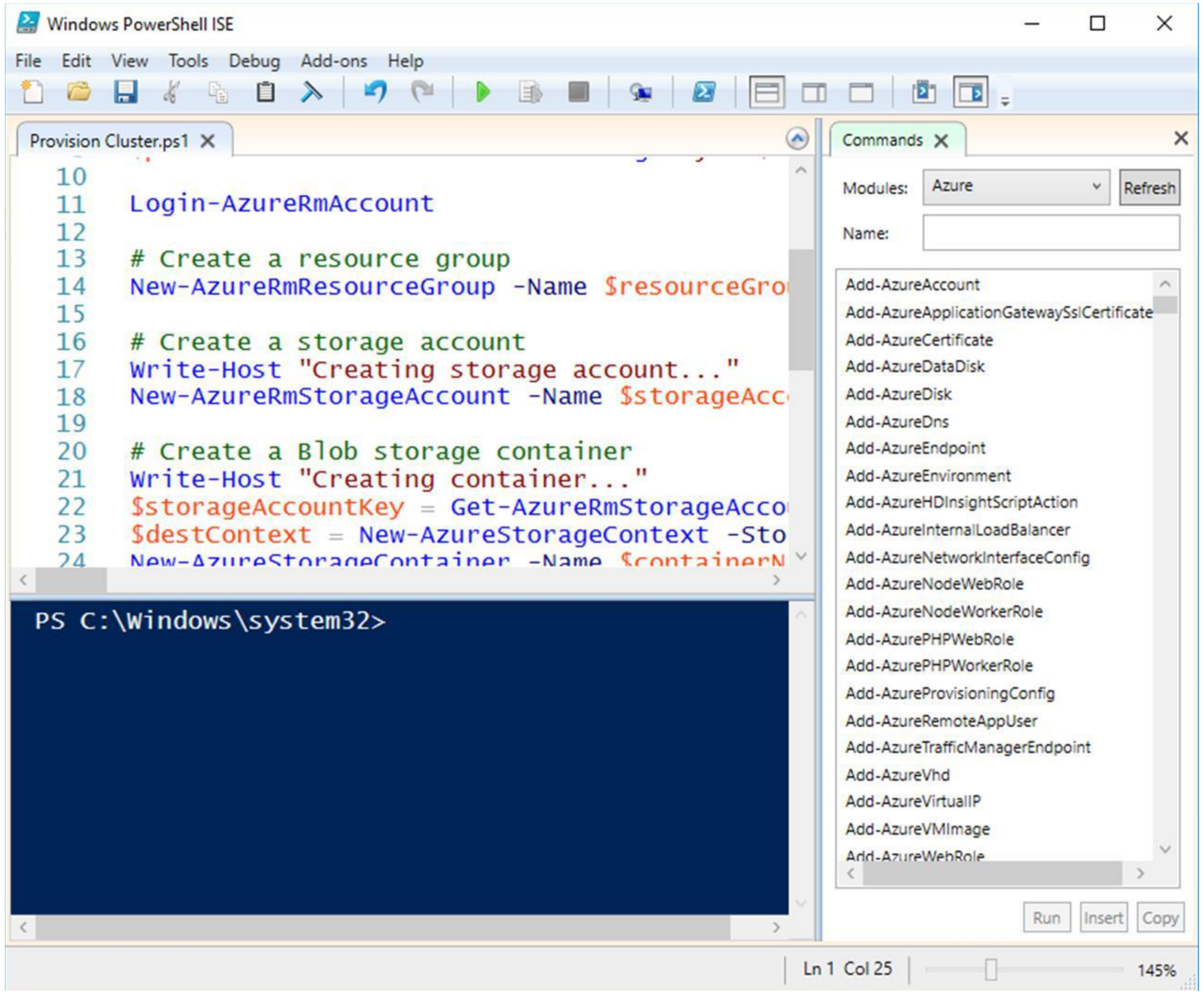
https://azure.microsoft.com/en-au/pricing/details/hdinsight/

Client Tools for HDInsight



PowerShell with HDInsight

- Use PowerShell to:
  - Provision HDInsight clusters
  - Upload/download files
  - Submit jobs
  - Manage cluster resources



```
Login-AzureRmAccount

# Create a resource group
New-AzureRmResourceGroup -Name $resourceGro

# Create a storage account
Write-Host "Creating storage account..."
New-AzureRmStorageAccount -Name $storageAcc

# Create a Blob storage container
Write-Host "Creating container..."
$storageAccountKey = Get-AzureRmStorageAcco
$destContext = New-AzureStorageContext -Sto
New-AzureStorageContainer -Name $containerN
```

```
PS C:\Windows\system32>
```

```
Add-AzureAccount
Add-AzureApplicationGatewaySslCertificate
Add-AzureCertificate
Add-AzureDataDisk
Add-AzureDisk
Add-AzureDns
Add-AzureEndpoint
Add-AzureEnvironment
Add-AzureHDInsightScriptAction
Add-AzureInternalLoadBalancer
Add-AzureNetworkInterfaceConfig
Add-AzureNodeWebRole
Add-AzureNodeWorkerRole
Add-AzurePHPWebRole
Add-AzurePHPWorkerRole
Add-AzureProvisioningConfig
Add-AzureRemoteAppUser
Add-AzureTrafficManagerEndpoint
Add-AzureVhd
Add-AzureVirtualIP
Add-AzureVMImage
Add-AzureWebRole
```

The End