House Rules

- Welcome! Thank you for joining us for **Big Data Processing on AWS Elastic MapReduce (EMR) using Hadoop and Spark.**
- Please feel free to pop your questions in the Q&A and our Engineers will answer them as we go along.
- Our Presenter will also answer some of the questions at different points.
- Please check your emails from Nikki for your AWS Account (Hash Link). Please check your Junk-Mail or Spam.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

About AWS Premium Support

AWS Premium Support is one-on-one, fast-response support from experienced technical support engineers. The service helps customers use AWS products and features. The **Big Data Cloud Support Engineering** team is one of our teams within Premium Support. Our Engineers support our customers around the globe, along with becoming Subject Matter Experts in their respective services, provide and receive training from global partners and have the opportunity to contribute to the development of our services.

Cloud Support Engineer - Emmanuel

The purpose of this event is to give you a chance to learn from our Big Data Cloud Support Engineers, learn more about one of our big data services and stand a chance to participate in our recruitment process, should you be successful in the completion of technical challenge on Day 3.

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

What are Cloud Support Engineers?

- The front line of AWS technical support
- Multiple teams, each with specific focus areas
- What do we do?
    - Interact with AWS customer through a variety of contact channels
    - Solving critical and complex customer problems that may span multiple AWS services
    - Training
    - Learning **fast**
    - Check out a day in the life of an AWS engineer blog post

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Characteristics of a Cloud Engineer……..

- Has great communication skills
- Is technical, knows the fundamentals of databases, storage, networks and operating systems
- Is thorough and diligent
- Is passionate about learning
- Has great problem solving skills
- Can articulate themselves without being too technical
- Can read in between the lines

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Agenda

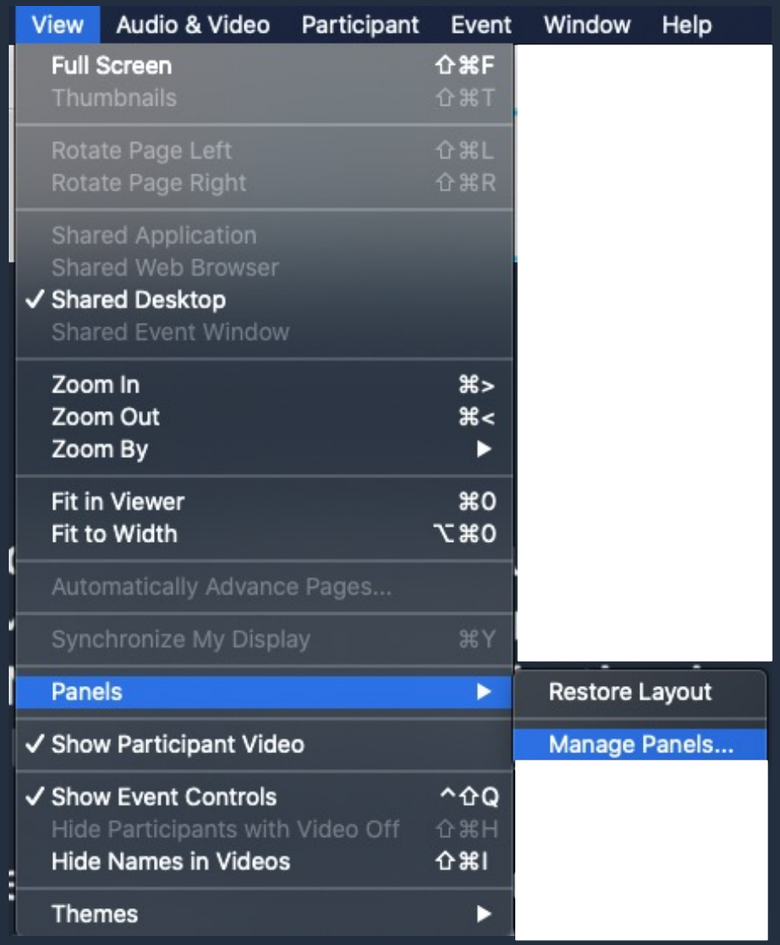
Day 1
- Introduction to Hadoop Framework
- HDFS

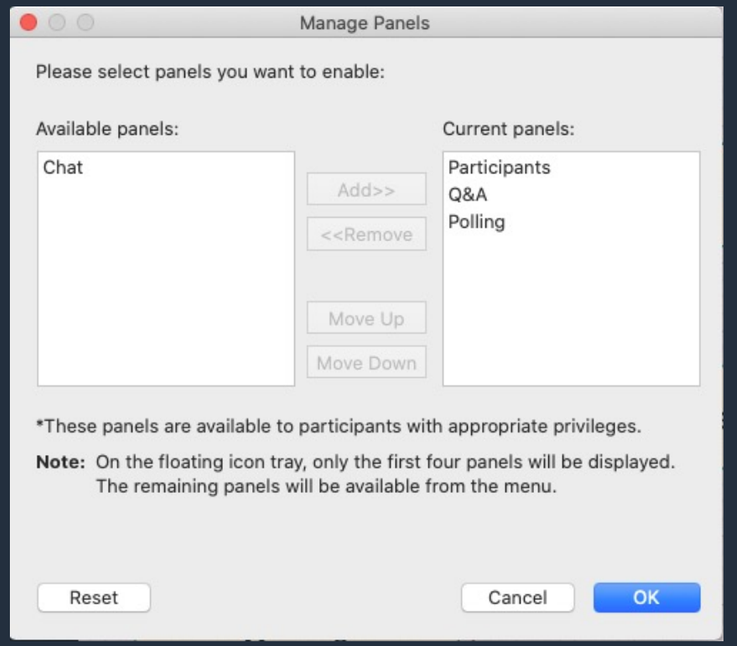
Day 2
- Spark
- Consolidated Lab

Day 3
- Technical Challenge

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Cisco Webex

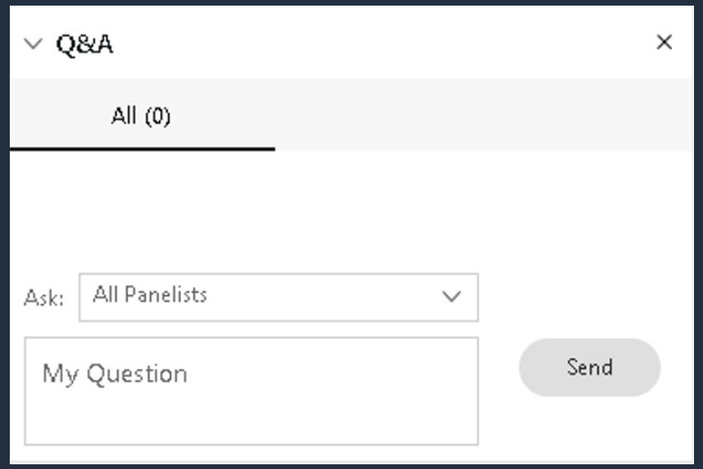




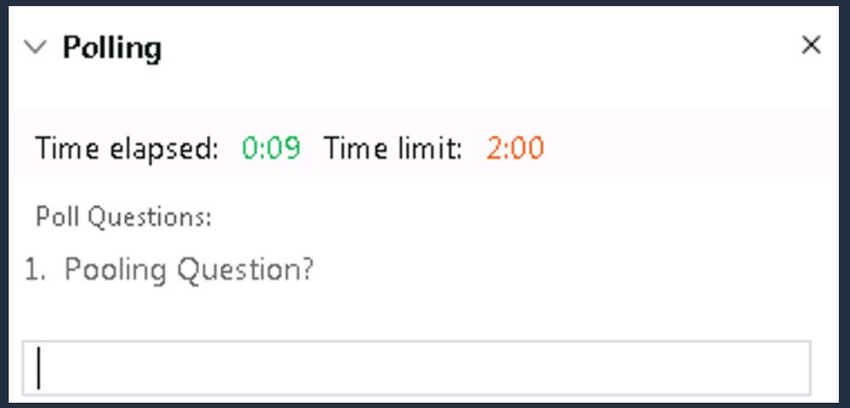
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Cisco Webex

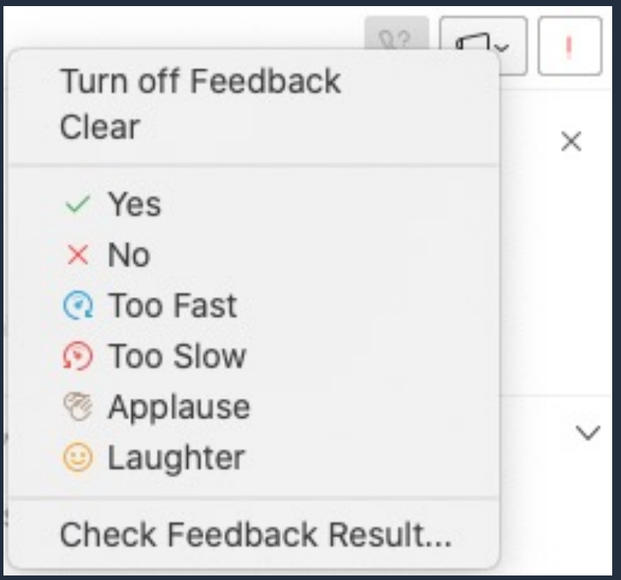
Q&A



Pooling Questions



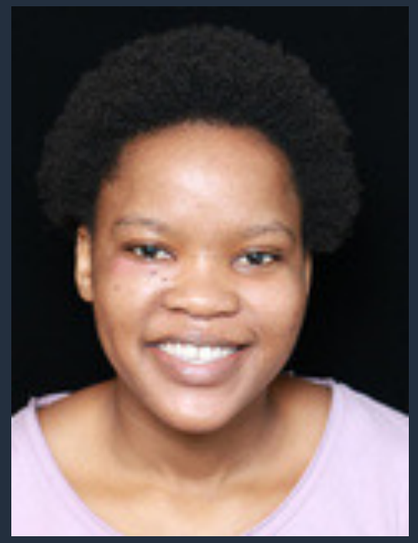
Interaction



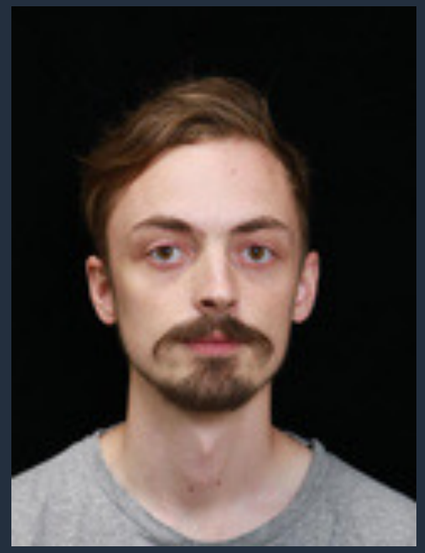
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Introduction

- Day 1 Instructors



Mandisa



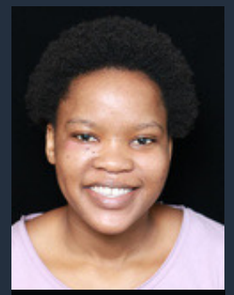
Stephen

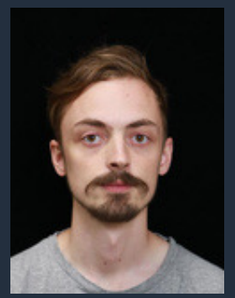
Emmanuel

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

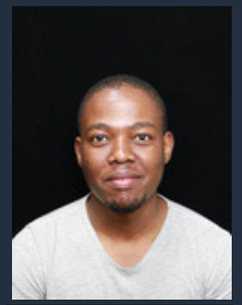
Introduction

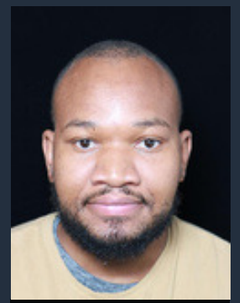
Big Data VLS Team

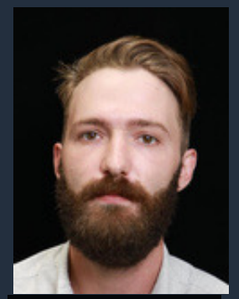

Mandisa

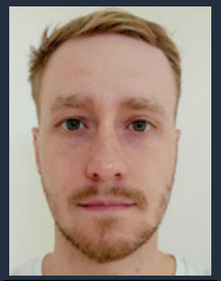

Stephen

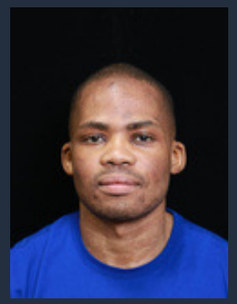

Emmanuel


Mahlatse

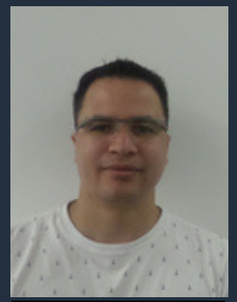

Odwa


Reinhardt


David


Samkelo


Ridick


Patrick


Emmanuel


Lavesh


Eddy

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Introduction to Hadoop Framework

Presented by
Mandisa

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Day 1 - Table of Contents

- Introduction to Hadoop
- Hadoop Components
- Comparison of Hadoop versions
- Hadoop use-cases
- Questions/ Break
- Lab 1 demonstration

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Introduction to Hadoop

- Big data as a concept is a vast amount of data (structured and unstructured) collected as means to extract meaningful insights .

- What are the requirements of big data?
    - High storage space
    - Processing power
    - Scalability

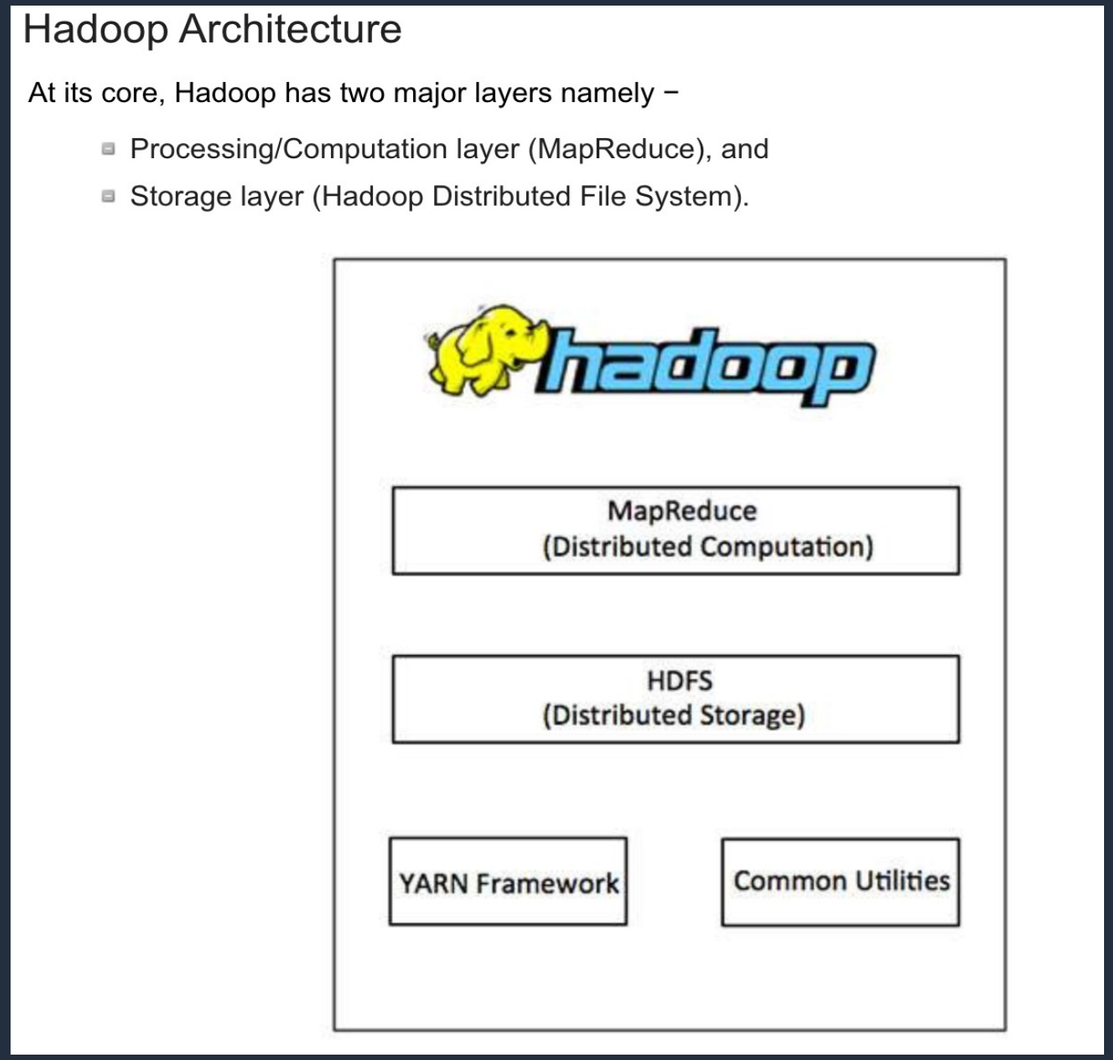
There is a need to handle the big data Vs (volume, velocity, variety, veracity)!

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Introduction to Hadoop

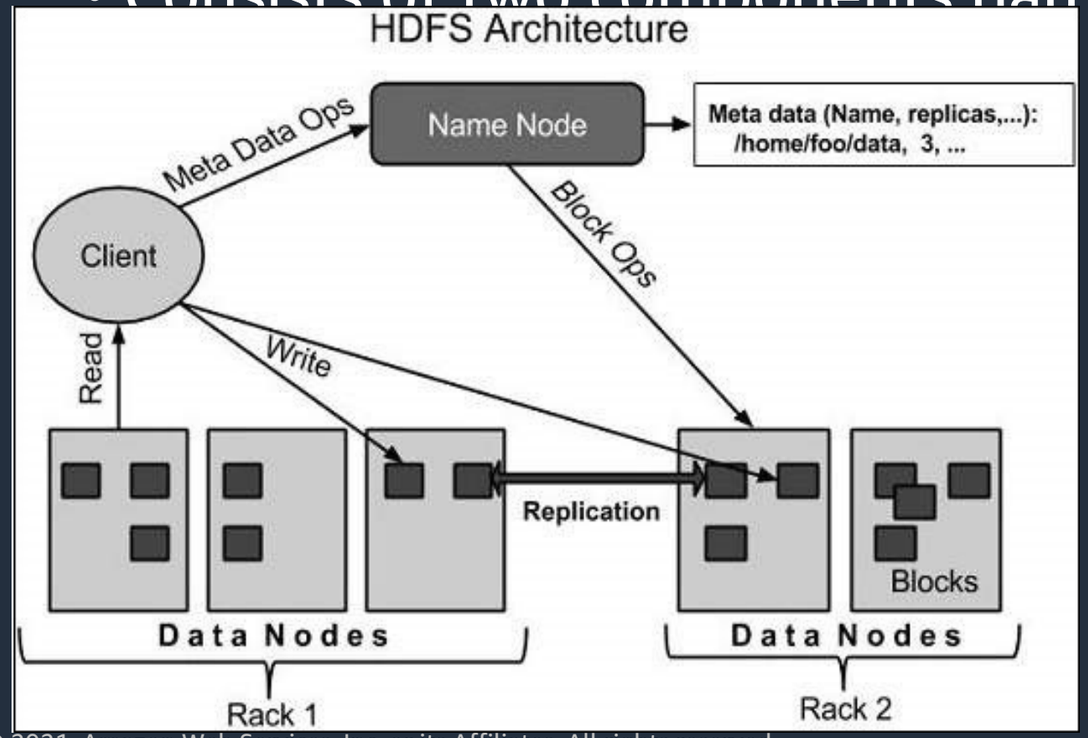
- Hadoop is defined as an open source framework that is used to store, process, and analyse chunks of data.
    - HDFS (Hadoop Distributed File System) for data storage
    - MapReduce/ Yarn (Yet another resource negotiator) for data processing

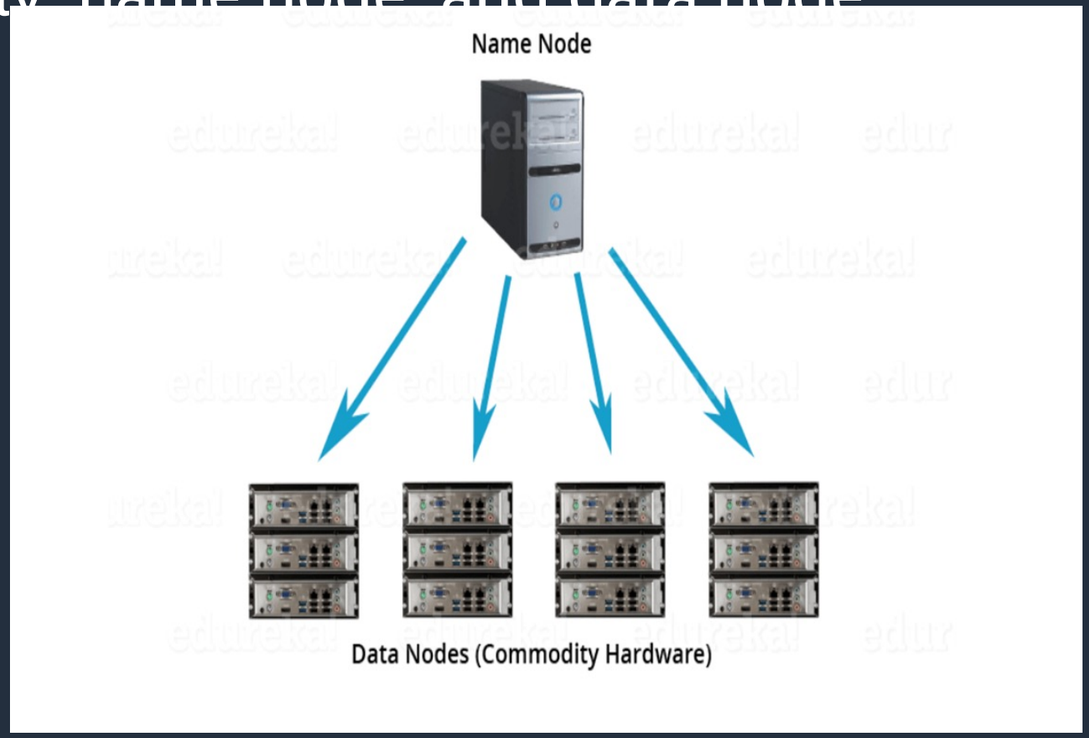


© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Hadoop core components: HDFS

- A distributed file system that runs on commodity hardware used for managing and storing data in a form of blocks across the cluster. The configuration is maintained on the cluster using hdfs-site.xml and core-site.xml files.
- Consists of two components namely, name node, and data node





© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Hadoop core components: HDFS

Name node

- Runs on a master daemon
- Manages and maintain data nodes
- Records the data metadata in memory such as size, permission, storage location, etc.
- Receives heartbeat and block report from all the data nodes.
    - ‘hdfs dfsadmin –report’, ‘hdfs fsck /’

Data node

- Runs on a slave daemon
- Stores the actual data in disk space.
- Responsible for serving read and write requests from the clients.
- When a data node is down, it does not affect the availability of data or the cluster.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Hadoop core components: MapReduce

- MapReduce is a software framework and programming model that allows easy writing of applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner.
- There are two phases involved in a MapReduce job:
    - Map – splits and maps data
    - Reduce – shuffles and reduce the data
- The input to each phase is **key-value** pairs
- Limitation:
    - Only supports batch processing not real-time data processing
    - Not suitable for small data files < 128 MB (Block size)

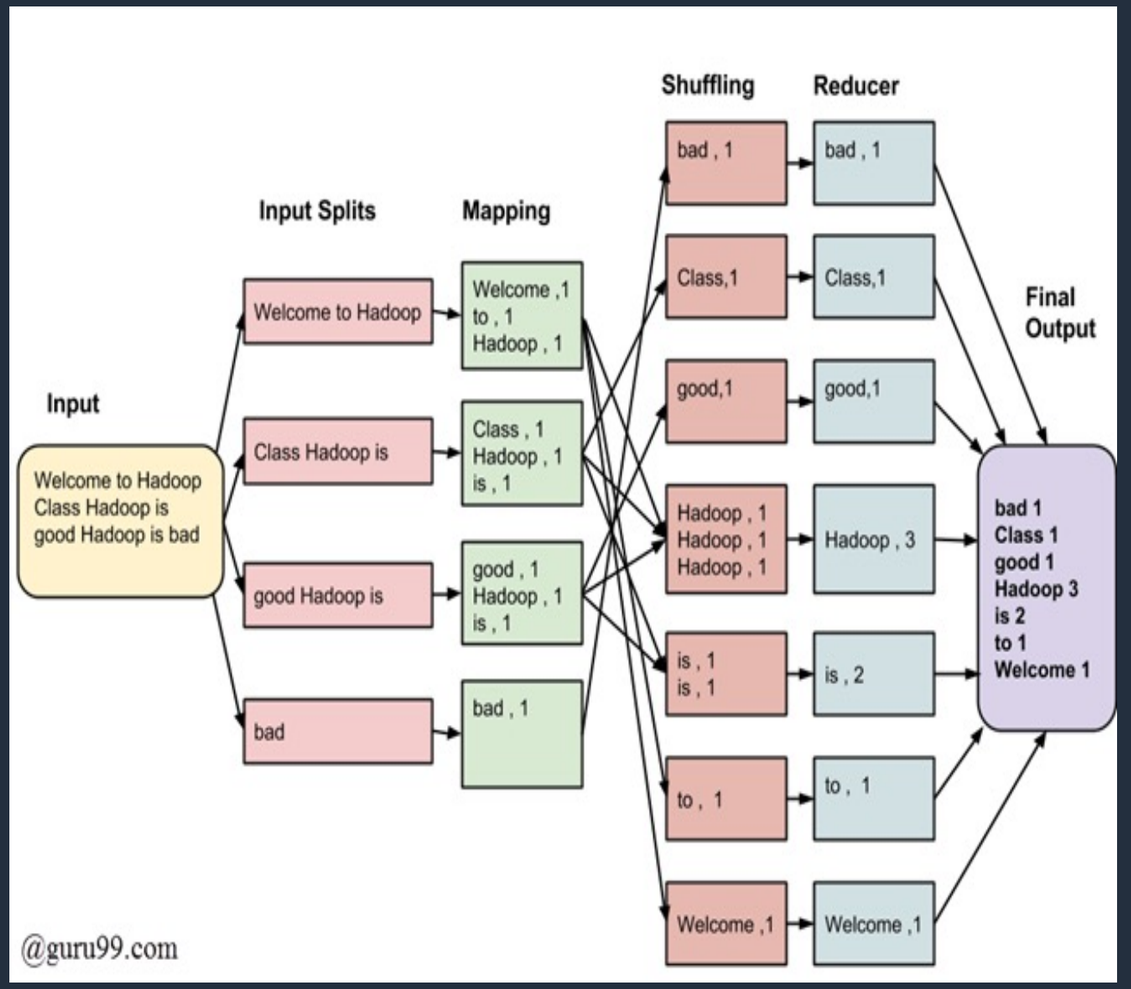
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Hadoop core components: MapReduce

For an example MapReduce word count job processes:

- Input splits – the input file is divided into a fixed size of records.
- Mapping - Count a number of occurrences of each word from input splits.
- Shuffling - Same words are grouped together along with their respective frequency.
- Reducing – Summarizes the complete data set using results from the shuffling phase.



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Hadoop core components: YARN

- Apache YARN (Yet Another Resource Negotiator) is a resource management system introduced in Hadoop v2 to improve the MapReduce implementation, but it is generally enough to support other distributed computing paradigms such as Spark, hive, Hbase and others.

- YARN makes use of the components namely, resource manager, node manager, application master and containers for job scheduling, processing of activities by allocating resources and scheduling tasks.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Hadoop core components: YARN

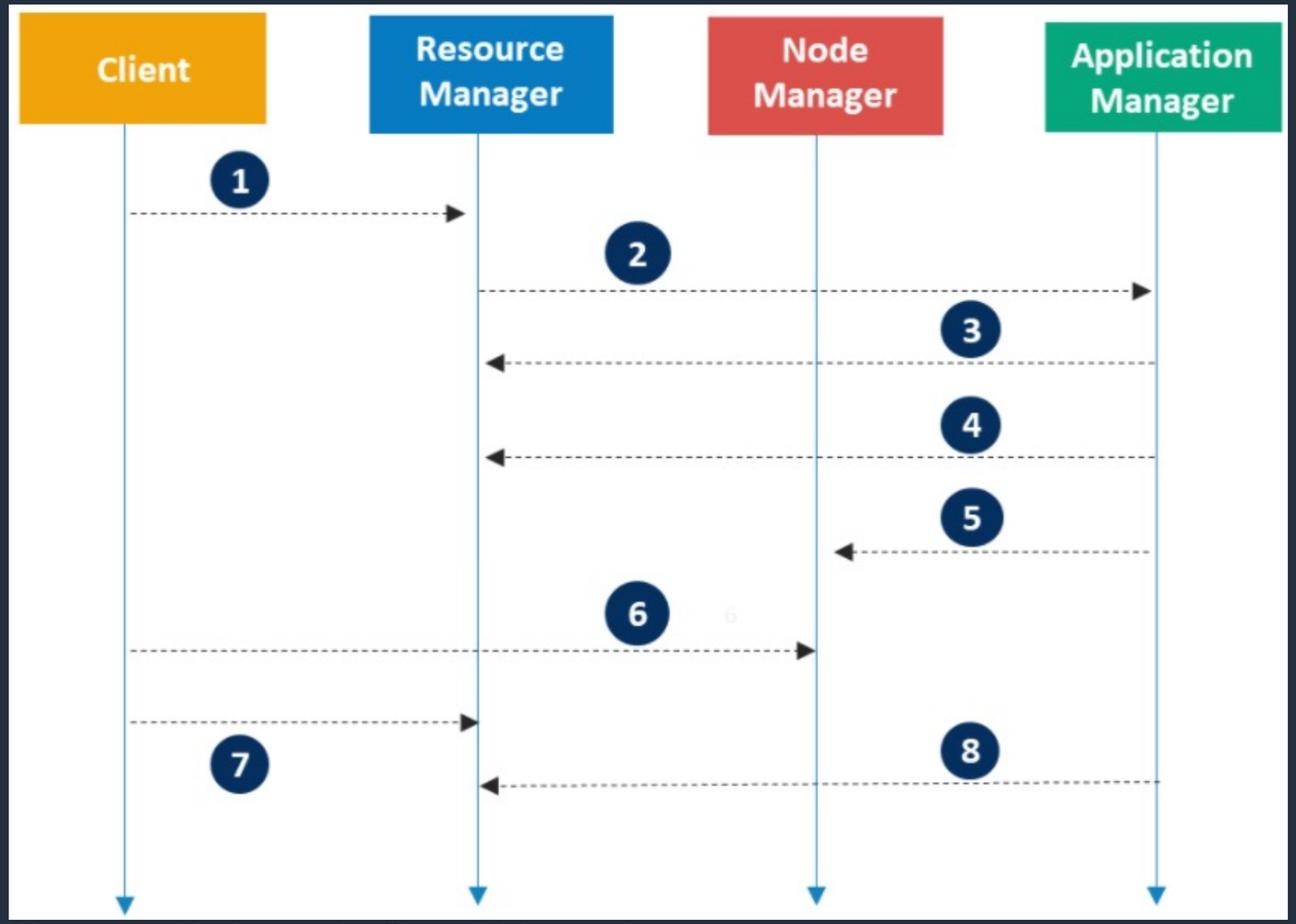
Components:

- Resource manager runs on a master daemon and used to manage resource allocation in the cluster.
- Application manager is responsible for managing user job lifecycle and resource requirements for individual applications.
    - Node Manager and monitors the execution of tasks.
- Node manager runs on a slave daemon and is used for managing the execution of tasks from the data node.
- Containers are packages of resources (RAM, CPU, Network, HDD and others) on a single node used to run jobs.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Hadoop core components: YARN



How application workflow is carried in YARN

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Hadoop core components: YARN

Useful commands and tools to monitor YARN applications:

- Yarn rmadmin (https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YarnCommands.html)
    - Reload queue properties
    - Refresh host information at the resource manager

- View/ monitor running yarn applications from the command line
    - Yarn top
    - Yarn application list \
    - Yarn application –kill <application id>

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Key differences between Hadoop versions

- You can confirm Hadoop version used by the cluster – Hadoop version command.

- There are currently 3 Hadoop versions
    - Hadoop 1,
    - Hadoop 2,
    - and Hadoop 3.

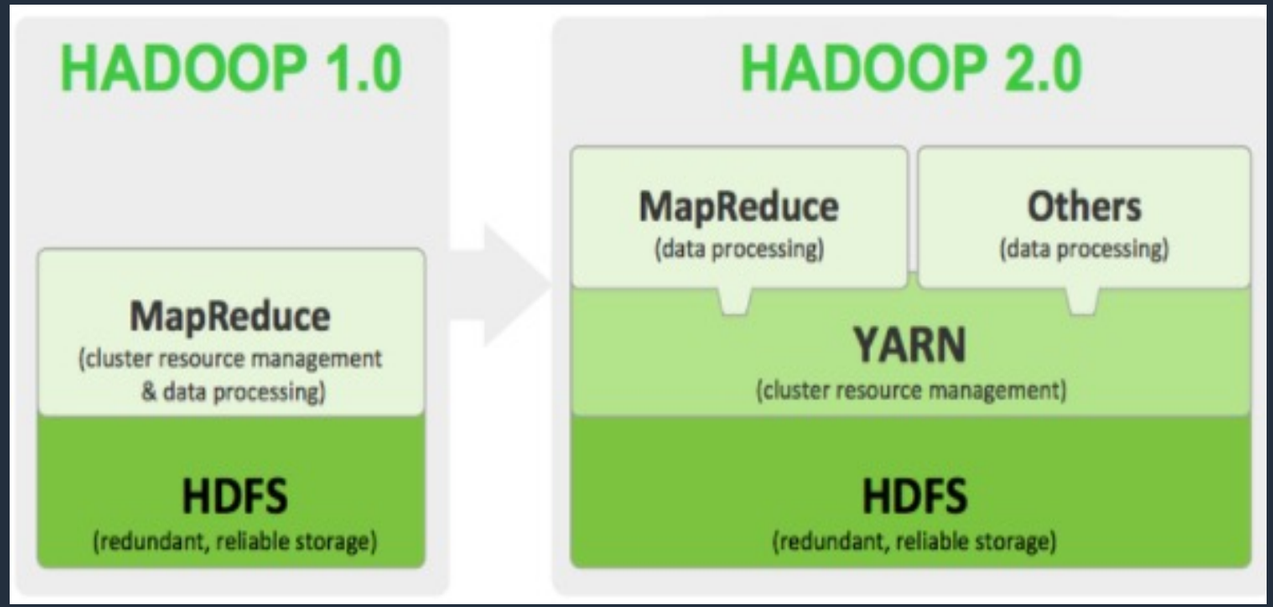
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Key differences between Hadoop versions: V1 and V2

Hadoop v2 was introduced to mitigate v1 limitations such as:

- No suitable for real-time and data streaming.
- Runs only MapReduce jobs
- Job tracker is a single point of failure
- No multi-tenancy support
- Scales up to 4000 nodes



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

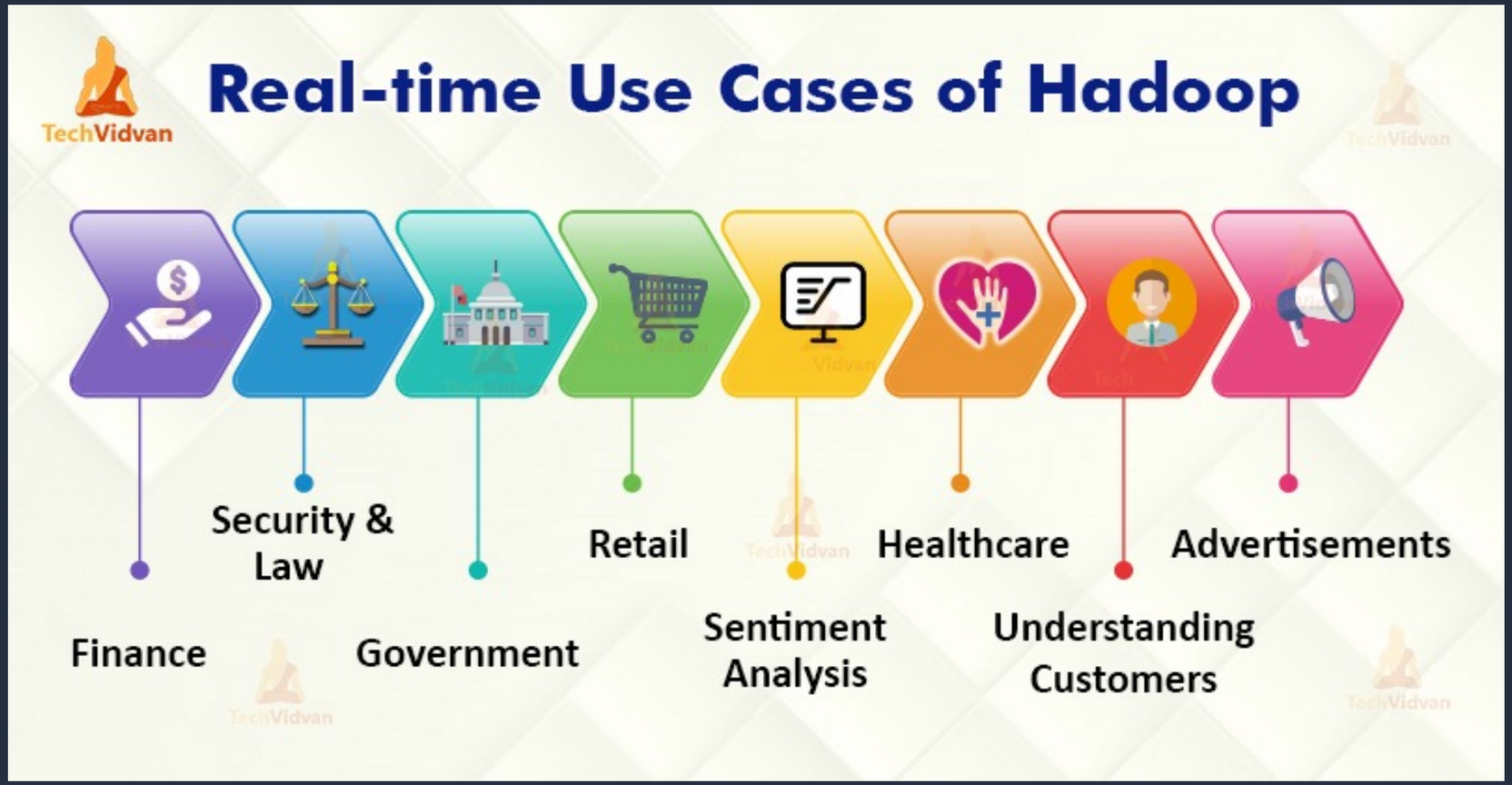
Key differences between Hadoop versions: V2 and V3

Feature	Hadoop v2	Hadoop v3
Min java version required	Java 7	Java 8
Fault-tolerance	Replication	Erasure coding
Storage scheme	3x replication factor for data reliability with 200% overhead	Erasure coding for data reliability with 50% overhead
Yarn Timeline service	Scalability issues	High scalable and reliable
Standby name node	Support only 1	Supports 2 or more
Heap management	Manual tuning of hadoop heap size	Auto tunes heap

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Hadoop use-cases



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Questions

Will resume in 15 minutes

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

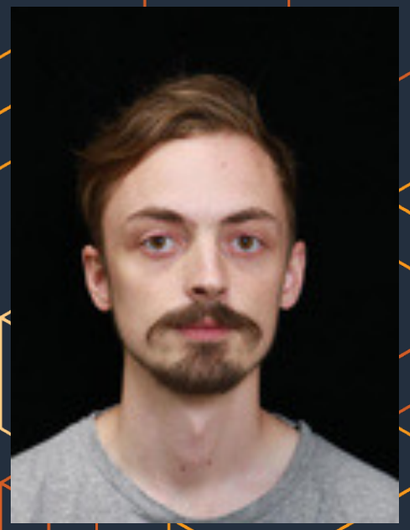
Lab 1: MapReduce job execution

https://github.com/big-data-vls/labs/tree/master/day-1/mapreduce-lab

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

VLS HDFS Content

A brief overview of the HDFS architecture and how it works.



By Stephen Hunt

June 2021

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Key Terms

- HDFS

The Hadoop Distributed Files System. A distributed file system designed to be highly fault tolerant and able to handle very large amounts of data.

- Name node

The core piece of the HDFS file system. The Name node stores the metadata, such as the directory tree, of all files stored in HDFS. The Name node is responsible for managing where in the cluster the file data is kept. The Name node does not store any of the file data itself.

- Data node

The component of HDFS that handles storing the file data. A functional HDFS system has more than one Data node, with the data replicated across them. Data in these nodes are stored as blocks, with a default block size of 128MB. The Data node periodically sends a list of blocks it is storing to the Name node.

- Block

A block is a file segment for the files stored on HDFS. When a file is added to HDFS, it is divided into 128MB blocks that are then stored in the Data nodes.

- Bastion Host

A host whose purpose is to provide access to a private network from an external network, such as the Internet. Because of its exposure to potential attack, a bastion host must minimize the chances of penetration.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Core Aspects of HDFS

- ## Streaming Data Access

Applications running on HDFS require streaming access to the data. HDFS is designed more for batch processing than interactive with an emphasis on high throughput rather than low latency data access.

- ## Large Data Sets

HDFS is designed to support large files. A typical file in HDFS is expected to be gigabytes to terabytes in size. HDFS is designed to provide high aggregate data bandwidth, scale to hundreds of nodes in a cluster and support tens of millions of files in a single instance.

- ## Simple Coherency Model

HDFS applications need a write-once-read-many access for files. Once a file is created, written and closed it is not expected to be changed aside from appends and truncates. Appending content to the end of files is supported, but files cannot be updated at arbitrary points. This assumption simplifies data coherency issues and enables high throughput data access.

- ## Portability Across Platforms

HDFS is designed to be easily portable from one platform to another to facilitate widespread adoption.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Poll – What is the default HDFS Block Size

1. 64MB
2. 128MB
3. 256MB
4. 512MB

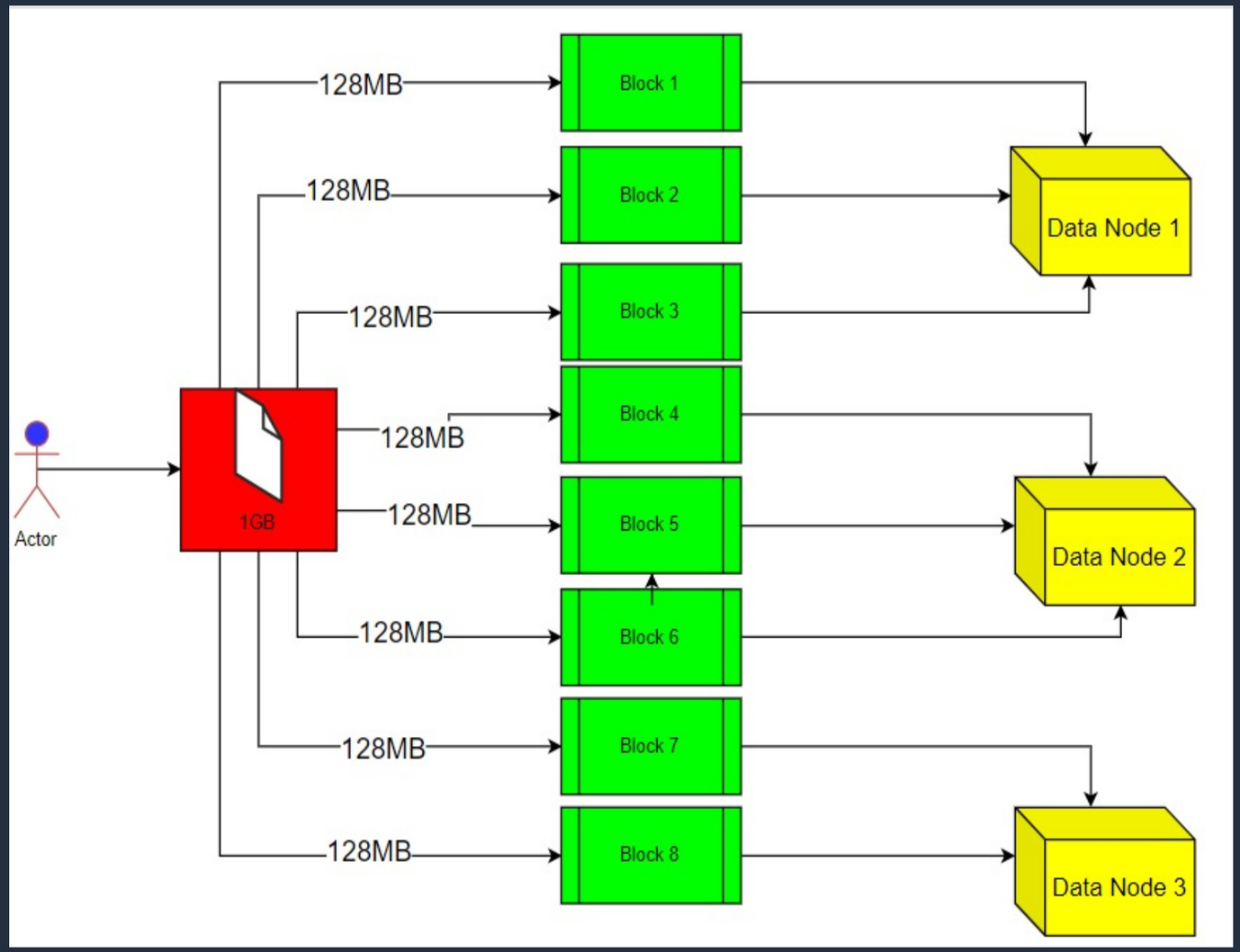
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

How HDFS Stores Files in Blocks

1. Actor writes a 1GB files to HDFS

2. The file is broken into blocks of 128MB

3. These blocks are distributed across all Data Nodes

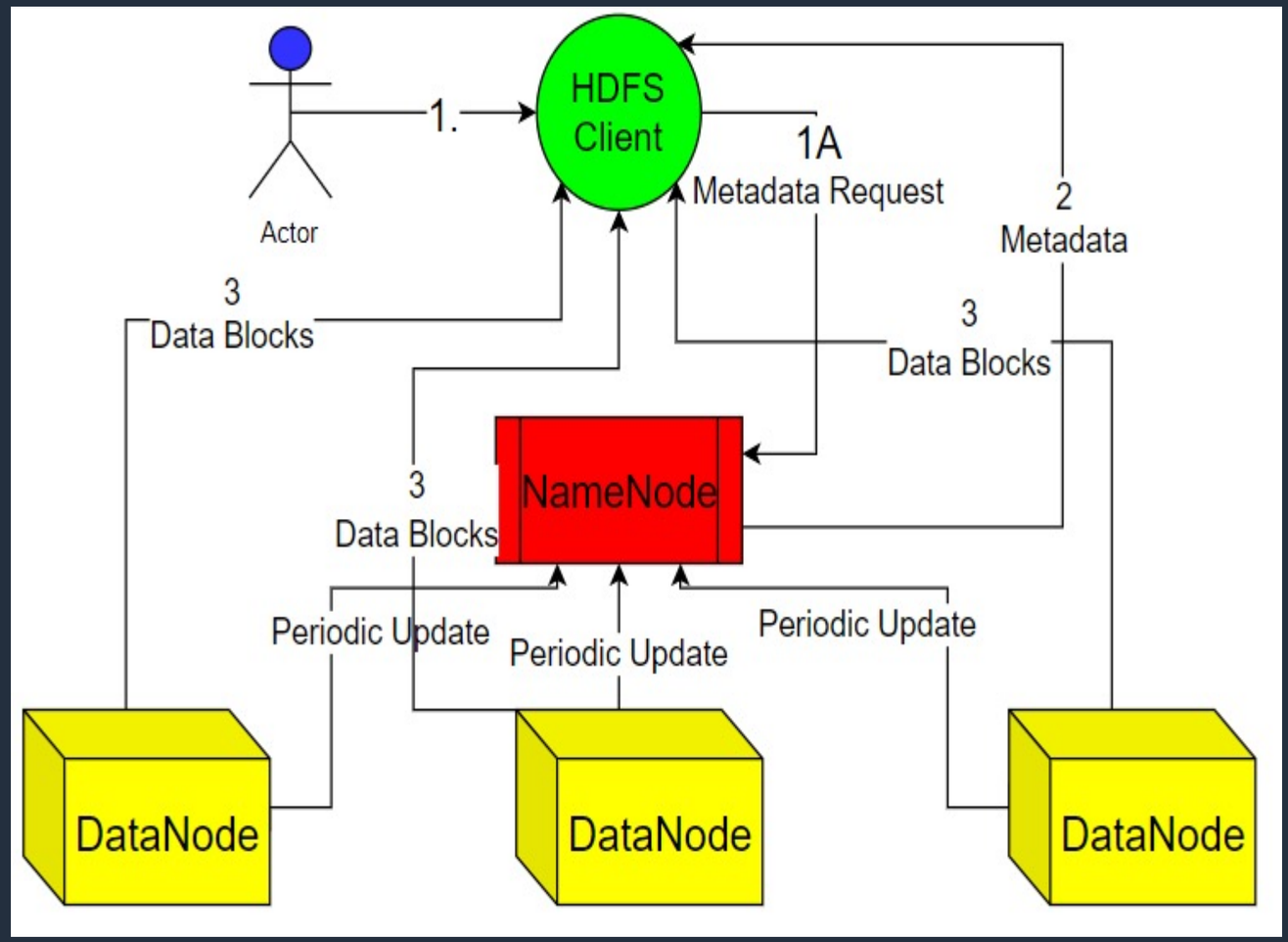


© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

How A Read/Write is Handled in HDFS

Client Application tries to read/write a file on HDFS

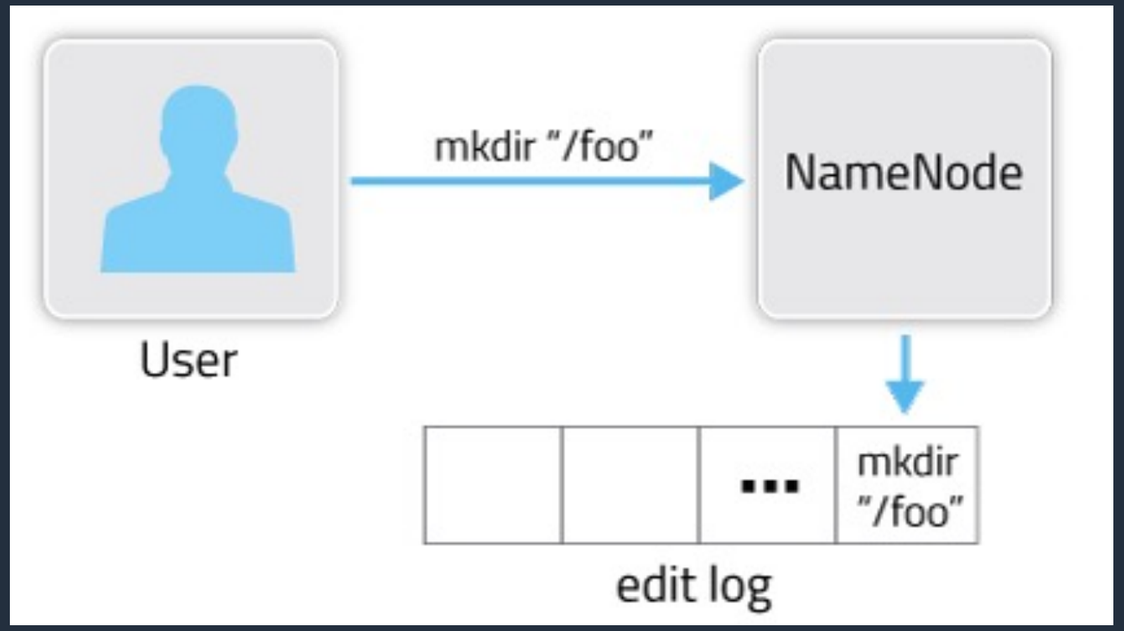
1. Client requests file system metadata.
2. NameNode sends Client the metadata, which includes a list of relevant Data nodes where the data lives.
3. Client application now communicates directly with the DataNodes to read/write data



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Persistence of File System Metadata

- HDFS namespace stored by the NameNode.
- NameNode uses a transaction log called the EditLog to persistently record every change that occurs to the file system metadata.
- Includes reads, writes, changes to replication factor etc. The entire namespace, including mapping of blocks to files and file system properties, is stored in a file called the FsImage.
- Both the EditLog and FsImage are stored on the NameNode's local file system.
- The NameNode keeps an image of the entire namespace and file Blockmap in memory.



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Poll – How does HDFS ensure that blocks are not prematurely replicated when loading a previous state?

1. Blocks all actions while starting
2. Blocks are not replicated during start-up
3. Only allows read-only access during start-up
4. Creates a duplicate of the metadata and uses that to restore state

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

HDFS Safemode

- Read-only mode.
- Used when loading file system state.
- Prevents premature replication of blocks when waiting for DataNodes to report their blocks.

- hdfs dfsadmin –safemode enter

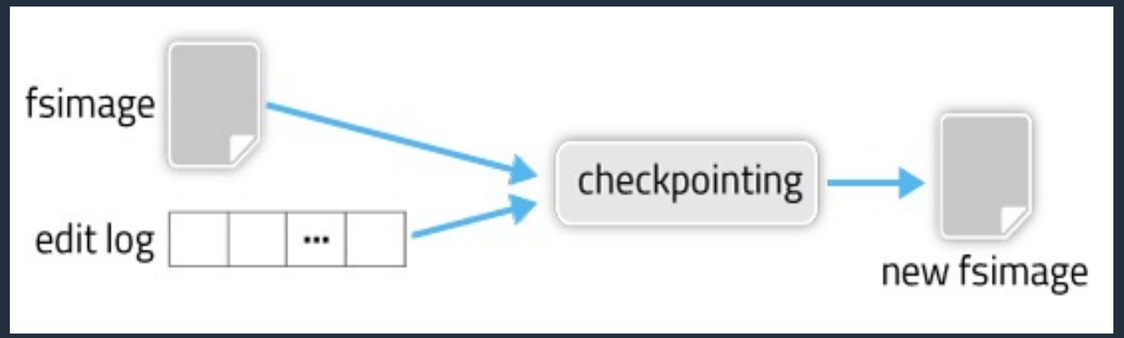
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

NameNode Start-up and Checkpointing

1. NameNode enters Safe Mode.
2. NameNode reads the FsImage and EditLog from disk and applies all transactions from the EditLog to the in-memory representation of the FsImage.
3. The in-memory representation of the FsImage is flushed into a new FsImage on disk.
4. The old EditLog is truncated as all transactions have been applies.
5. NameNode exits Safe Mode.

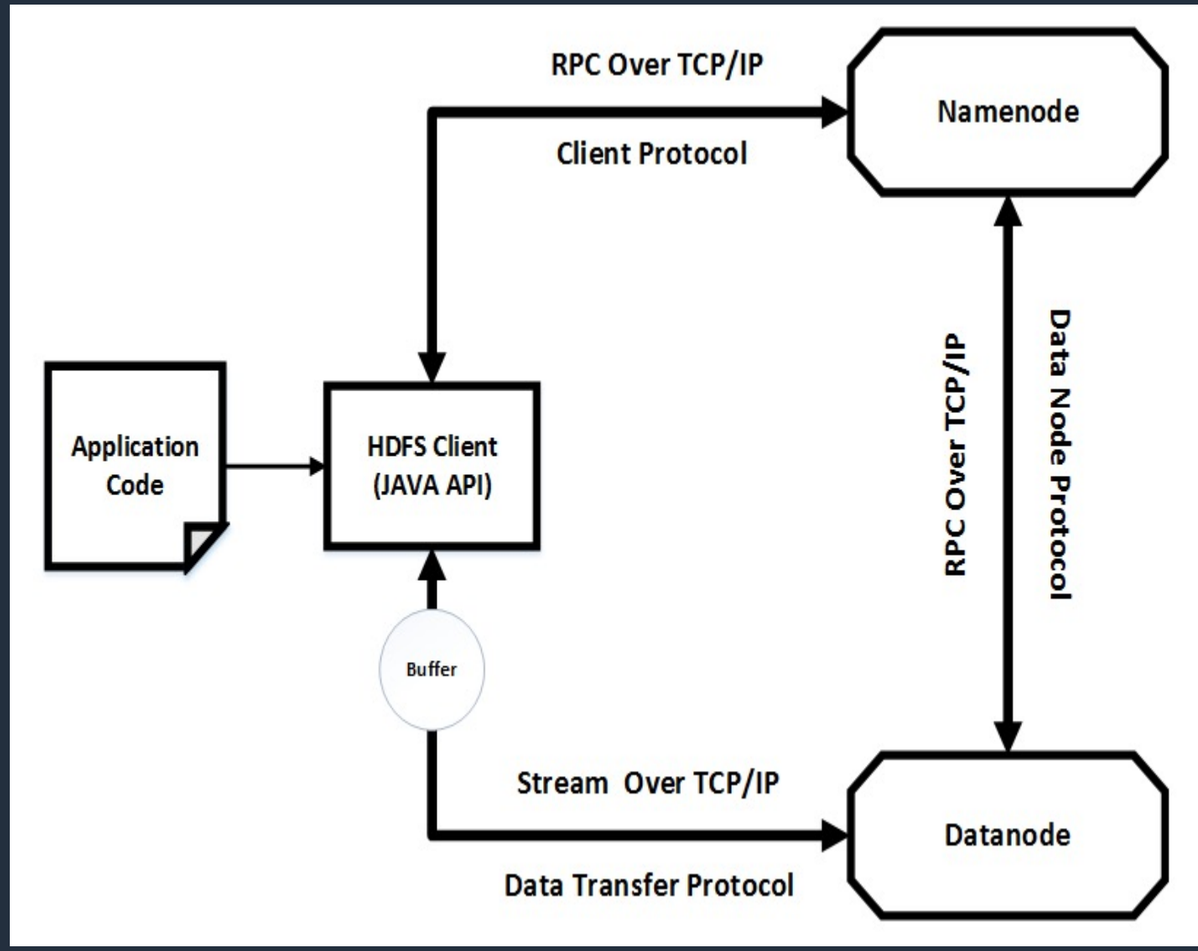
- This process ensures that HDFS maintains a consistent view of the files system metadata.



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

HDFS Communication Protocols

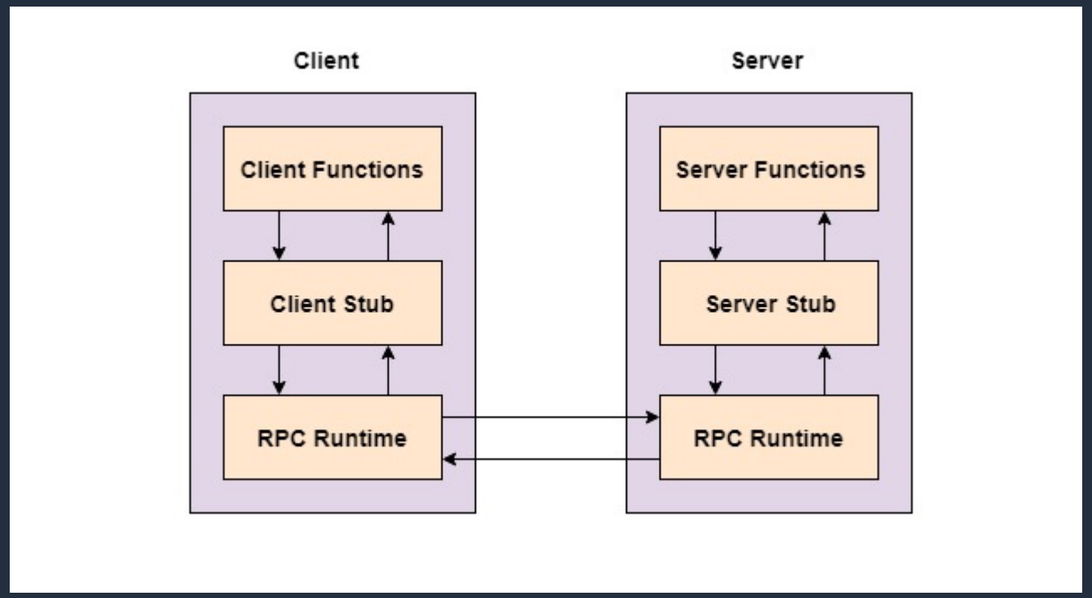
- TCP/IP
- Client Protocol
- Data Node Protocol
- Remote Procedure Call (RPC) wraps both.
- The NameNode never initiates, it only responds to DataNodes or clients.



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Remote Procedure Calls

- Inter-process communication technique.
- Abstracts message passing system away from user.
- Stalls execution on Client until a response is received.



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

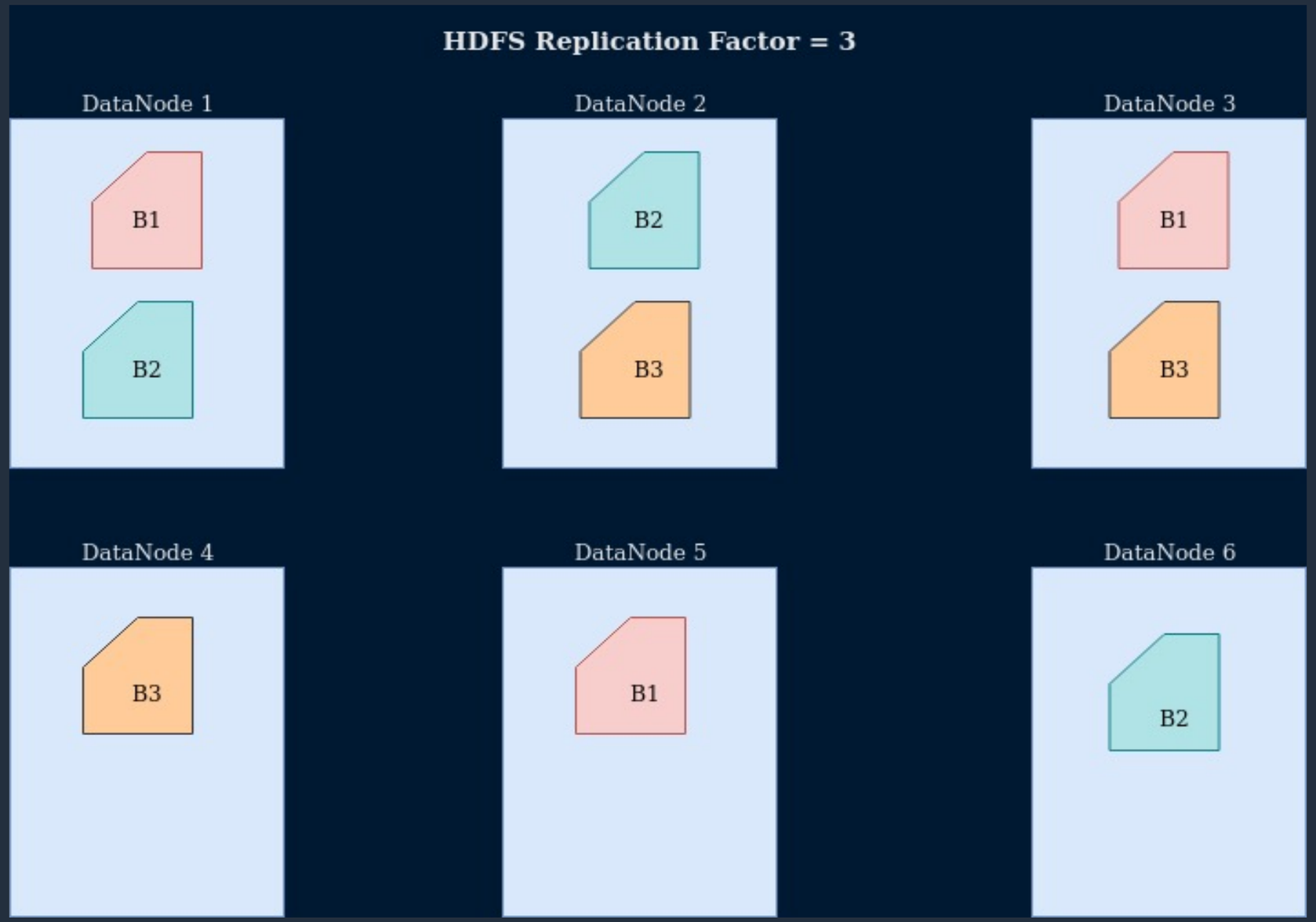
Poll – How does HDFS ensure data resiliency?

1. Blocks are replicated to a set factor
2. All blocks have a copy stored on the NameNode
3. Each DataNode holds every block
4. All of the above

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

HDFS Replication Factor

- To ensure that HDFS is fault tolerant, data is replicated to a set factor.
- 3 blocks in HDFS with a Replication factor of 3 equals a total of 9 blocks.
- Replication increases overall storage needs. Storing 1TB with a replication factor of 3 needs 3TB storage.



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Rack Awareness

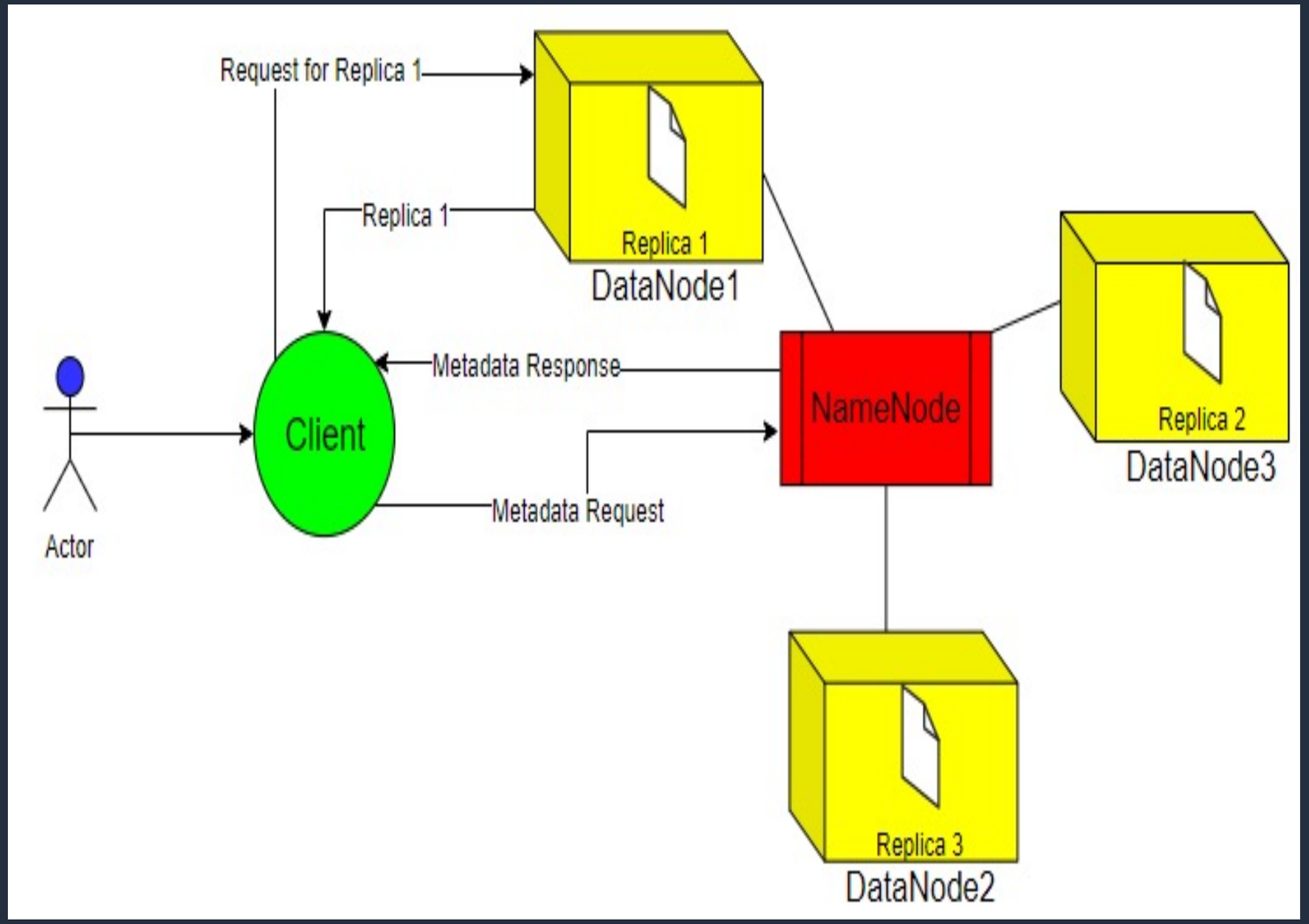
- HDFS block placement avoids placing replicas on the same rack to provide better data resiliency and availability.
- Rack ID found using either a specified external script or via a configured Java class.
- Interface expects a one-to-one correspondence to be maintained and topology information to be in the format ‘/rack/host’ where ‘/’ is the topology delimiter, ‘rack’ is the rack identifier and ‘host’ is the individual host.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Replica Selection on Read

- HDFS will try satisfy a read with the replica that is closest to the reader.
- Local replicas are favoured over remote replicas.
- Reduces global bandwidth consumption and read latency.



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Data Node Disk Usage and HDFS

- Files stored on HDFS are distributed amongst all Data Nodes.
- Overutilization of HDFS can cause high disk usage on Data Node instances and cause issues with frameworks such as YARN.
- Check total HDFS size and usage using ‘hdfs dfsadmin –report’
- Check Data Node instance disk utilisation using ‘df –h’

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

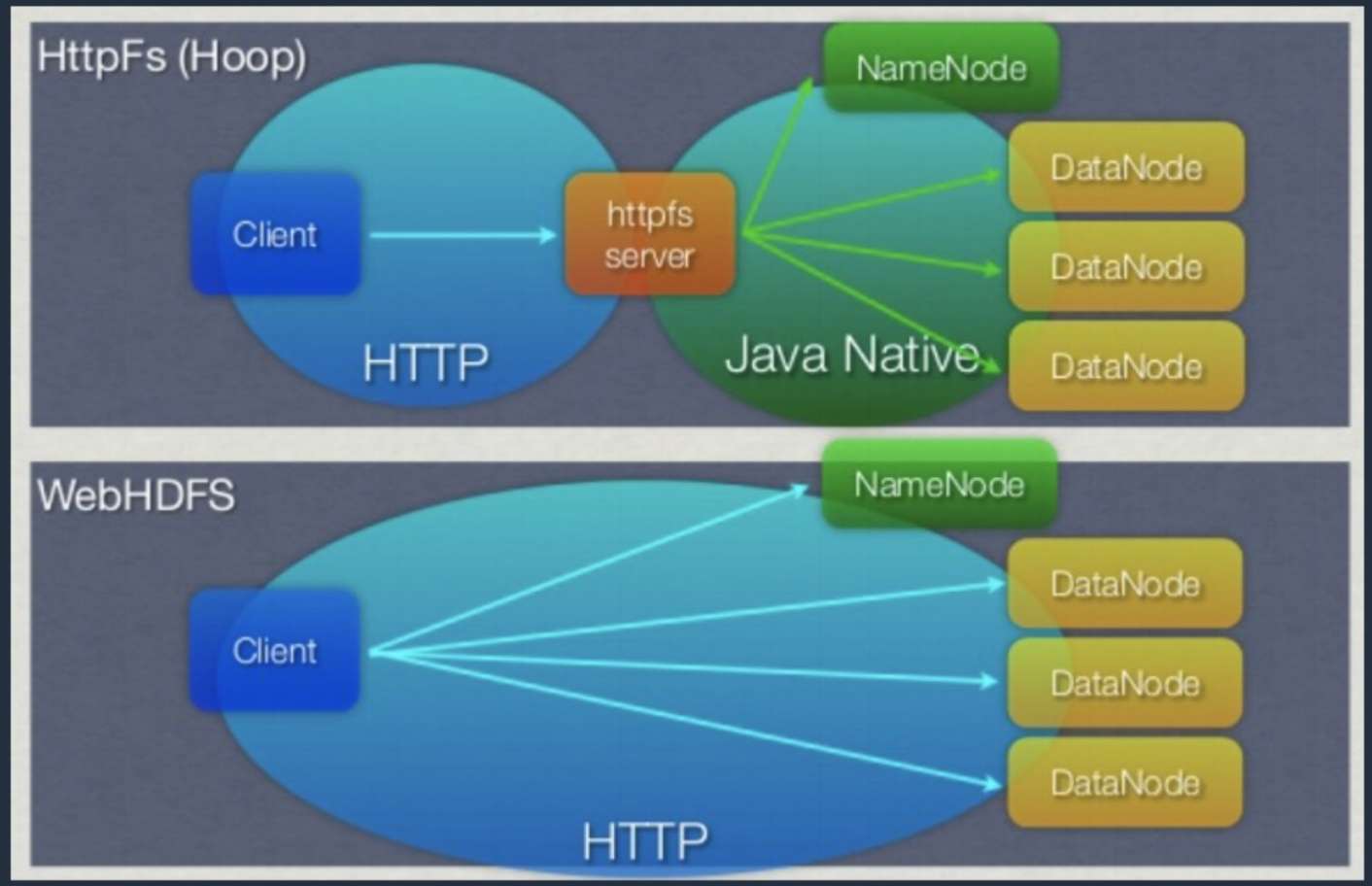
HttpFS vs WebHDFS

HttpFS

- A single node acts as a ‘gateway’
- Potential chokepoint during file transfer.
- Help minimize footprint required to access HDFS.

WebHDFS

- Needs access to all nodes.
- Data is transmitted from node directly when read



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Using HttpFS - Setup

1. Ensure that the Security Group on the Master allows traffic on port 14000
2. If the cluster is in a private subnet:
    1. Configure a Bastion host in a public subnet of the same VPC
    2. Ensure the Bastion host Security Group allows traffic from your IP on port 22
    3. Ensure Master Security Group allows traffic from the Bastion Security Group
    4. SSH into the Bastion to access HttpFS

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Poll – What Command Line tool can be used to make HTTP requests?

1. ping
2. netstat
3. traceroute
4. curl

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Using HttpFS

- Create a directory inside the tmp directory called mydir

```
curl -i -X PUT "http://<master-ip>:14000/webhdfs/v1/tmp/mydir?op=MKDIRS&user.name=hadoop"
```

- Push a local files inside mydir

```
curl -i -X PUT -L --header "Content-Type: application/octet-stream"
"http://<masterip>:14000/webhdfs/v1/tmp/mydir/file1.txt?op=CREATE&user.name=hadoop" -T file1.txt
```

- List all files inside the mydir directory

```
curl -i "http://<master-node-ip>:14000/webhdfs/v1/tmp/mydir?op=LISTSTATUS&user.name=hadoop"
```

- Get the file content of a file

```
curl -i -X GET -L "http://<master-nodeip>:14000/webhdfs/v1/tmp/mydir/file1.txt?op=OPEN&user.name=hadoop"
```

- Delete a file.

```
curl -i -X DELETE "http://<masterip>:14000/webhdfs/v1/tmp/mydir/file1.txt?op=DELETE&user.name=hadoop"
```

- Delete the mydir folder

```
curl -i -X DELETE "<master-ip>:14000/webhdfs/v1/tmp/mydir/?op=DELETE&user.name=hadoop&recursive=true"
```

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

HDFS and User Home Directories

- Only users that exist on the NameNode exist to HDFS.
- Requests for HDFS files all have to go through the NameNode.
- User and groups data is retrieved from the NameNode requests.
- If a user or group exists on a DataNode instance, but not on the NameNode instance, it does not exist in HDFS.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

The Small Files Problem

- Each file is stored in its own block.
- The NameNode needs to store information about each block.
- MapReduce processes each block one at a time.
- Lots of small files means potentially hitting the OS open file limits.

- 10 000 1MB files vs 10 1GB files?
- ➢ 10 000 1MB files = 10 000 blocks in HDFS
- ➢ 10 1GB files = 80 ((1GB/128) * 10) blocks in HDFS

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

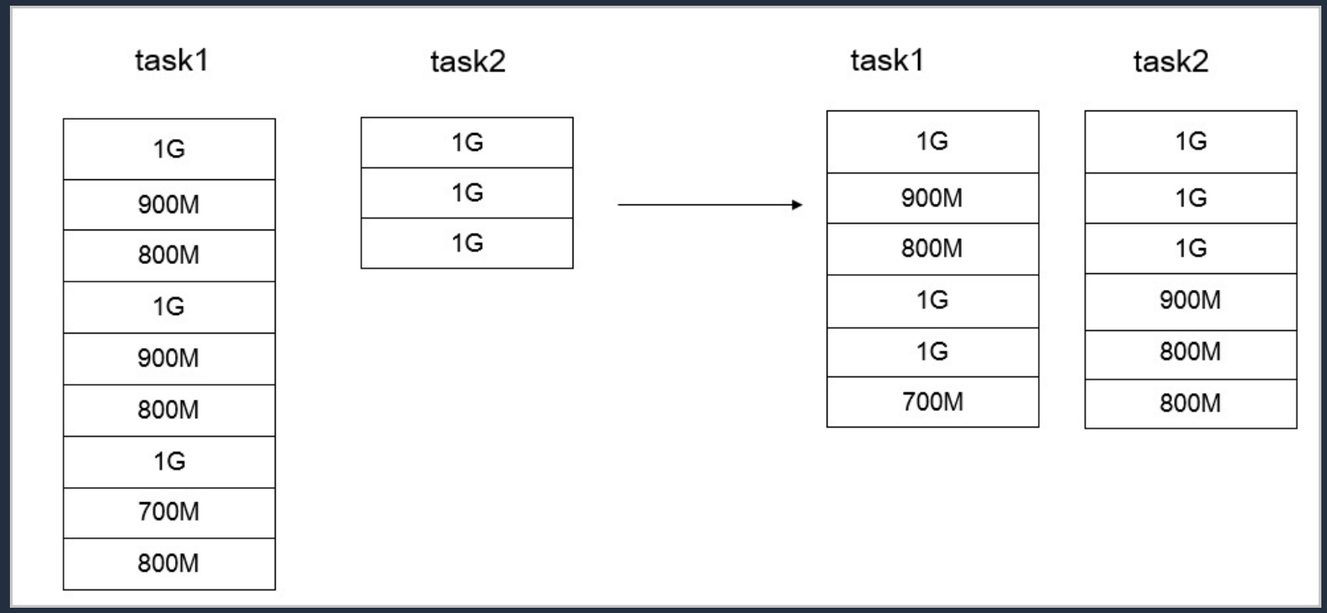
The Small Files Problem - Solutions

- Group files together into larger files.

This method effectively avoid the small files problem, but does not work for all situations (e.g. thousands of image files)

- Use HDFS's sync() method.

This is an HDFS method of writing large files, but also does not work for all situations.

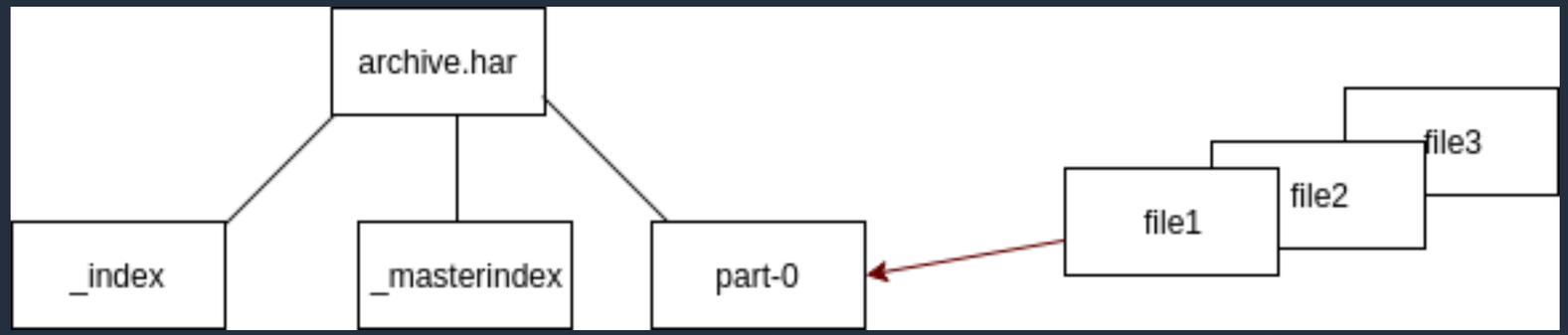


© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

The Small Files Problem - Solutions

- Use HAR (Hadoop Archives) to build a layered file system on top of HDFS.

HAR files aim to reduce the number of files by grouping them in archives. However, this comes at the cost of file access being slowed due to two index file reads on top of the data file read.



- Use Sequence Files to allow for streaming of the small files.

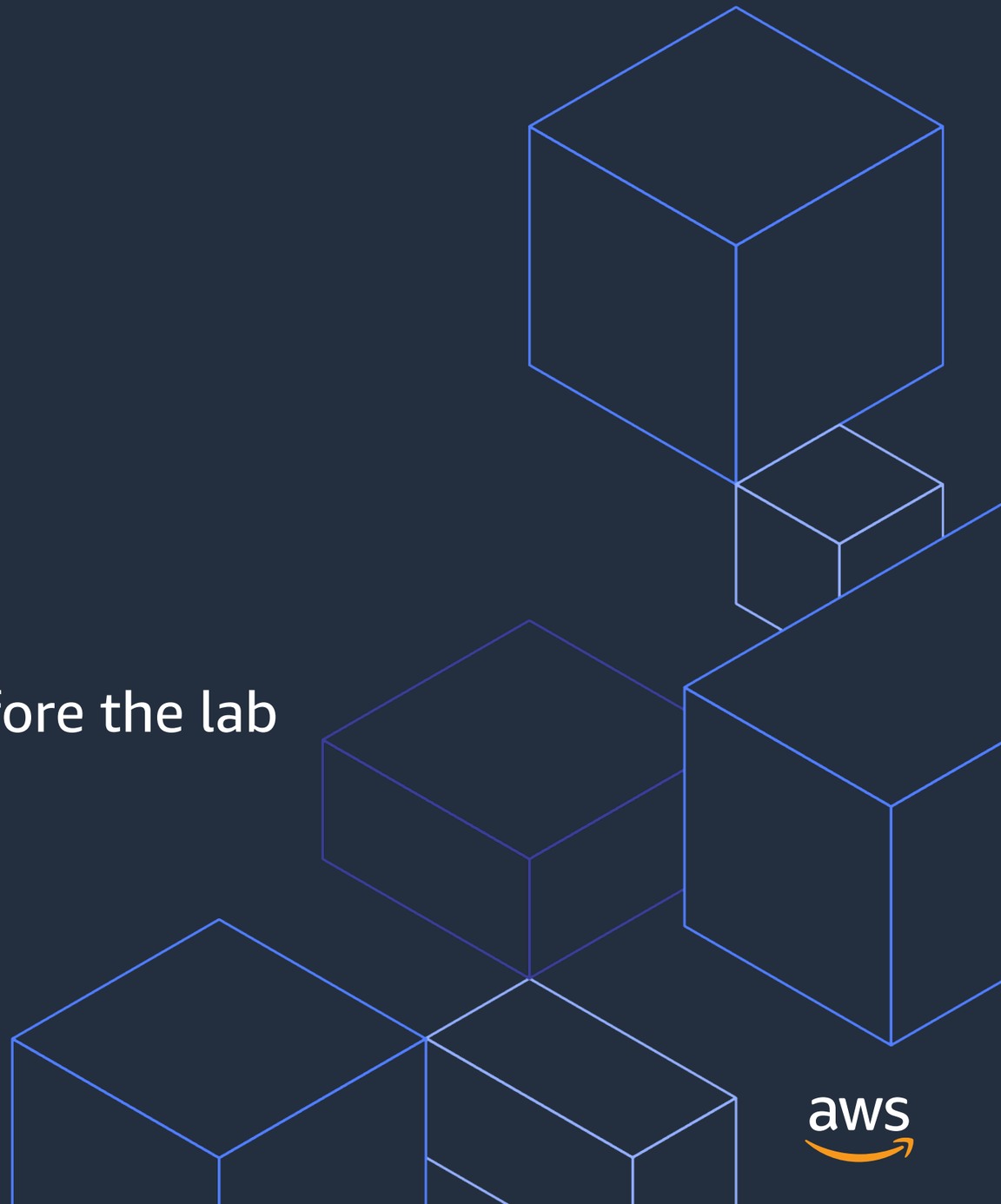
Sequence files work on the concept of using the filename as a key and the contents as a value. These files can then be processed in a streaming fashion and split into chunks for better parallelism.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Break

15 minute break before the lab



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Hands-on Lab for HDFS

- Log in to the EMR Master Node
- Run HDFS DFS and Admin commands
- Increase the replication factor
- Use Safe Mode

- Find the lab at: https://github.com/aws-support-bigdata-cpt-vls/2021/tree/main/Day%201/HDFS%20Lab

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws