**Assignment 2**

Q1: Explain NameNode startup & NameNode Checkpoint in detail.

Answer:

- NameNode Startup:

- When the NameNode starts, it enters a safe mode. Safe mode is a read-only mode for the HDFS cluster, where no new blocks are allowed to be written.

- During startup, the NameNode loads the namespace image (fsimage) from the disk into memory.

- It then reads the edit log (edits) to apply any recent changes to the filesystem since the last checkpoint.

- After applying the changes, the updated namespace image is written back to disk, and the edit log is truncated.

- The NameNode then starts to process block reports from DataNodes to ensure the block replication factor is met before exiting safe mode and allowing normal operations.

- NameNode Checkpoint:

- Checkpointing is the process of merging the edit log with the fsimage to create a new fsimage.

- The Secondary NameNode or a Backup Node performs this checkpointing process.

- It periodically downloads the edit log and fsimage from the NameNode, merges them, and uploads the new fsimage back to the NameNode.

- This reduces the size of the edit log and ensures faster NameNode restarts.

Q2: How HDFS handles DataNode failure?

Answer:

- HDFS is designed to handle DataNode failures automatically to ensure data reliability and availability.

- When a DataNode fails, the NameNode detects the failure through missed heartbeat signals.

- The NameNode re-replicates the blocks stored on the failed DataNode to other active DataNodes to maintain the desired replication factor.

- Clients are redirected to other DataNodes to access the data without interruption.

- The NameNode maintains an updated list of DataNodes and their blocks, ensuring that all data blocks have sufficient replicas distributed across the cluster.

Q3: Explain HDFS File Permissions.

Answer:

- HDFS file permissions are similar to Unix file permissions and include the following components:

- Owner: The user who owns the file or directory.

- Group: The group associated with the file or directory.

- Permissions: Read (r), write (w), and execute (x) permissions for the owner, group, and others.

- Permissions are represented as a string of 10 characters (e.g., `-rwxr-xr--`):

- The first character indicates the type (e.g., `-` for a file, `d` for a directory).

- The next three characters represent the owner's permissions.

- The following three characters represent the group's permissions.

- The last three characters represent others' permissions.

- Commands like `hdfs dfs -chmod`, `hdfs dfs -chown`, and `hdfs dfs -chgrp` are used to change permissions, ownership, and group associations.

Q4: Explain HDFS fsck command and its options in detail with an example.

Answer:

- The `hdfs fsck` command is used to check the health of the HDFS filesystem and to find any inconsistencies.

- Options:

- `-move`: Move corrupted files to lost+found directory.

- `-delete`: Delete corrupted files.

- `-files`: Print out files being checked.

- `-blocks`: Print out block report.

- `-locations`: Print out locations for every block.

- `-racks`: Print out network topology for data-node locations.

- Example:

```sh

hdfs fsck / -files -blocks -locations

```

This command checks the entire HDFS filesystem, printing details about the files, blocks, and their locations.

Q5: Explain curl command and its options.

Answer:

- The `curl` command is a tool to transfer data from or to a server using various protocols like HTTP, HTTPS, FTP, etc.

- Options:

- `-o [file]`: Write output to the specified file.

- `-O`: Write output to a file named as the remote file.

- `-d [data]`: Send the specified data in a POST request.

- `-H [header]`: Pass custom header(s) to the server.

- `-X [method]`: Specify the request method to use (e.g., GET, POST, PUT, DELETE).

- `-I`: Fetch the headers only.

- Example:

```sh

curl -X GET https://example.com -H "Accept: application/json"

```

This command sends a GET request to `https://example.com` with a custom header `Accept: application/json`.

Q6: Which property is used to specify the block size of a file stored in HDFS?

Answer:

- The property `dfs.blocksize` is used to specify the block size of a file stored in HDFS.

- This property can be set in the HDFS configuration files (`hdfs-site.xml`) and determines the size of each block in bytes.

Q7: The NameNode maintains the namespace of the filesystem using which two sets of files?

Answer:

- The NameNode maintains the namespace of the filesystem using:

- fsimage: A snapshot of the filesystem metadata at a point in time.

- edits: A log of changes made to the filesystem metadata after the latest fsimage.

Q8: What does the following command do? `hdfs dfs -ls -R /user/thomas/`

Answer:

- The command `hdfs dfs -ls -R /user/thomas/` lists all files and directories under the `/user/thomas/` directory recursively.

- The `-R` option stands for recursive listing.

Q9: What does the following command do? `hdfs dfs -ls /user/thomas/`

Answer:

- The command `hdfs dfs -ls /user/thomas/` lists all files and directories directly under the `/user/thomas/` directory.

- It does not list contents recursively.