Hadoop Storage File System (HDFS)

Create a directory structure in HDFS to organize and store your data effectively.

A **by Anirudh Reddy**

Create a Directory Structure in HDFS

1

Step 1: Access HDFS

Access the Hadoop Distributed File System using the appropriate command line interface.

2

Step 2: Use Command

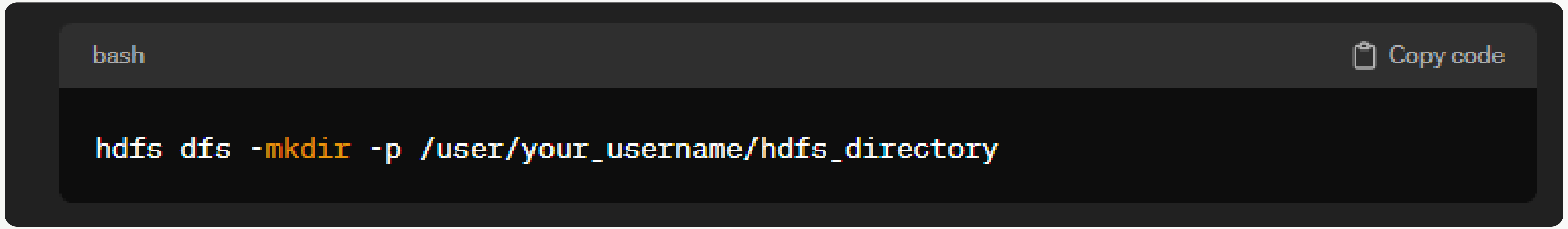
Execute the command "hdfs dfs -mkdir -p /user/your_username/hdfs_dir" to create the directory.

3

Step 3: Verify

Verify the successful creation of the directory within HDFS.

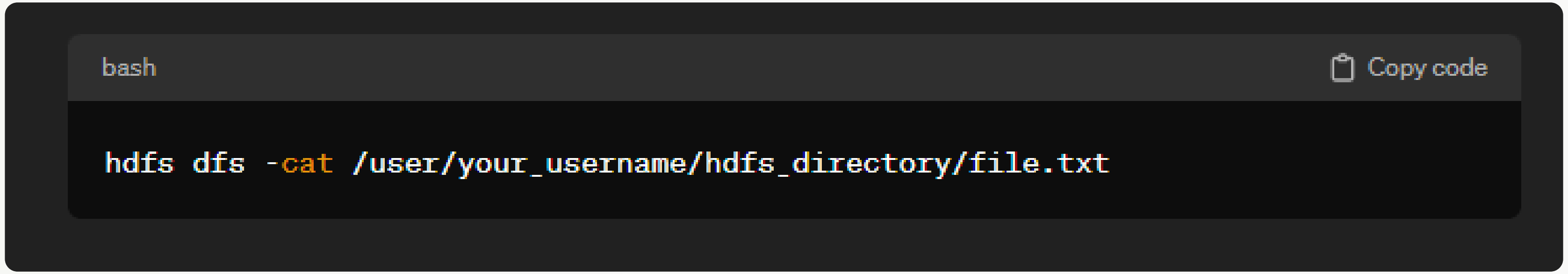
1.Create a directory structure in HDFS:



```bash
hdfs dfs -mkdir -p /user/your_username/hdfs_directory
```

- **hdfs dfs**: This is the command to interact with HDFS.
- **-mkdir -p**: This set of options instructs HDFS to create a directory, and the -p flag ensures that parent directories are created if they do not exist.
- **/user/your_username/hdfs_directory**: This is the path where the new directory will be created. Replace your_username with your actual username and hdfs_directory with the desired directory name.

2.Write HDFS command to see contents of files which are present in HDFS:



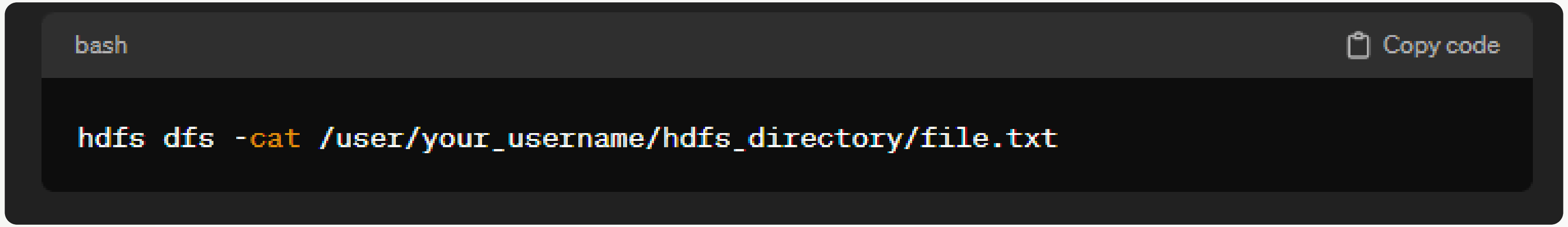
```bash
hdfs dfs -cat /user/your_username/hdfs_directory/file.txt
```

- **hdfs dfs**: This is the command to interact with HDFS.
- **-cat**: This option is used to concatenate and display the contents of files.
- **/user/your_username/hdfs_directory/file.txt**: This is the path to the file whose contents you want to view. Replace your_username, hdfs_directory, and file.txt with the actual values.

Viewing Data Contents, Files and Directory

In Hadoop Distributed File System (HDFS), efficient data exploration is crucial for effective data management. Understanding the contents of files and directories is key to unlocking insights. Utilizing HDFS commands, users can seamlessly navigate through data, view file contents, and examine directory structures. This capability facilitates a comprehensive analysis of data stored in HDFS, laying the foundation for informed decision-making and data-driven insights.

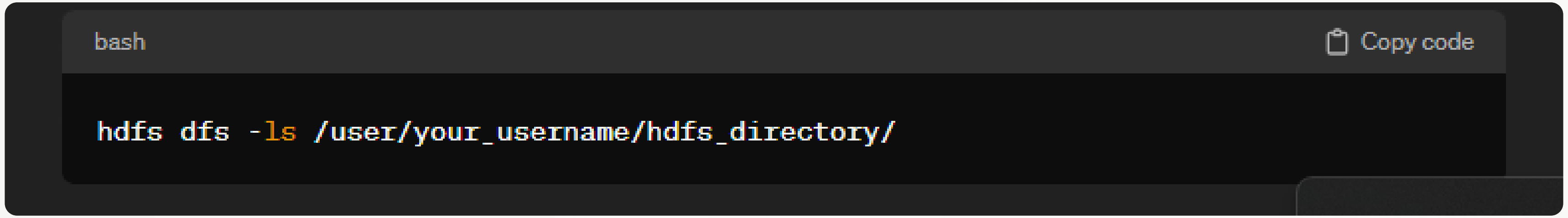
1.See the contents of files in HDFS:



```bash
hdfs dfs -cat /user/your_username/hdfs_directory/file.txt
```

Replace your_username with your actual username, hdfs_directory with the HDFS directory, and file.txt with the specific file you want to view.

2.See the contents of files in a directory in HDFS:



```bash
hdfs dfs -ls /user/your_username/hdfs_directory/
```

Replace your_username and hdfs_directory with the actual username and HDFS directory.

Understanding the Purpose of HDFS

Data Storage

HDFS is designed for reliable and efficient storage of large datasets.

Data Retrieval

It allows for the retrieval of data through distributed processing for fast access.

Fault Tolerance

HDFS ensures data reliability and fault tolerance through data replication.

HDFS Architecture

NameNode

Responsible for managing metadata and coordinating access to files.

DataNode

Stores actual data and performs read/write operations as directed by the NameNode.

Client Interactions

Clients communicate with NameNode and DataNode to perform file system operations.

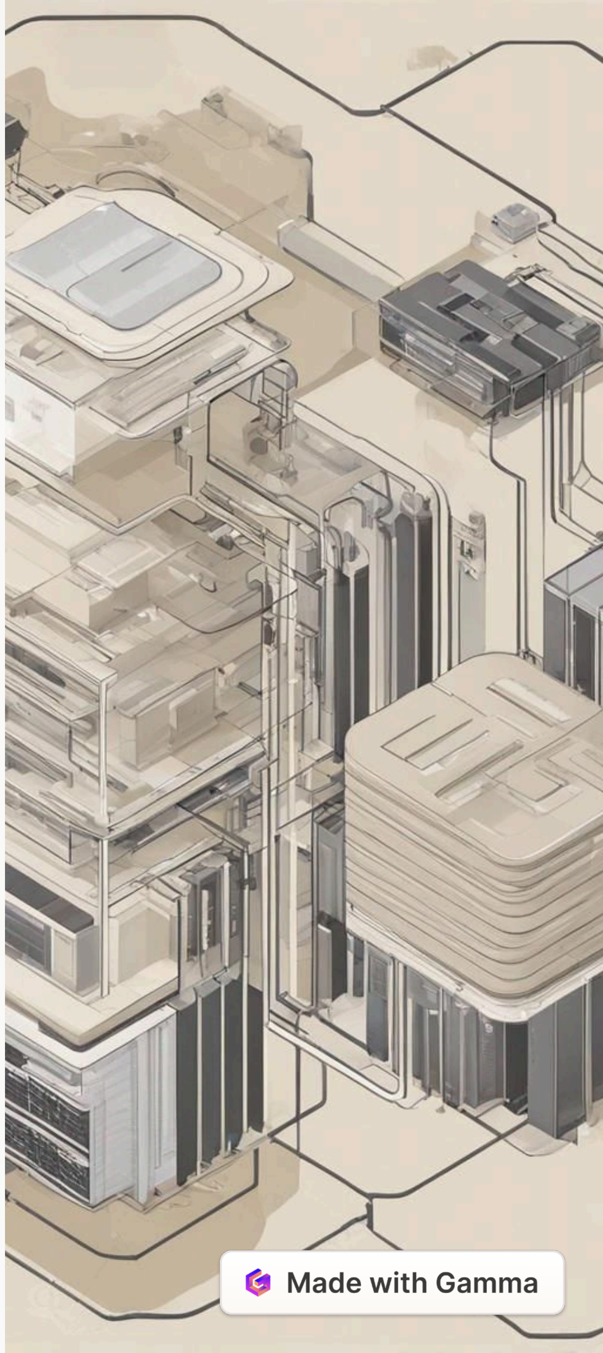
HDFS Components: NameNode and DataNode

1 NameNode

Manages metadata and oversees DataNode operations.

2 DataNode

Stores and retrieves data based on NameNode instructions.



HDFS Data Replication

1

2

Step 1: Block Splitting

Original file is broken into blocks for replication across DataNodes.

Step 2: Replication

Data is replicated to multiple DataNodes to ensure fault tolerance and data availability.

HDFS Fault Tolerance

3

Data Recovery

Automated recovery process ensures data integrity against failures.

99%

Data Reliability

High percentage of data reliability through replication and fault tolerance mechanisms.

HDFS Use Cases

- Big Data Analytics
- Cloud Computing
- Distributed Storage