

VIẾT CHƯƠNG TRÌNH ĐỌC VÀ GHI FILE TRONG HDFS

Mục tiêu

- Giúp sinh viên biết cách áp dụng các Java API, Hadoop API để tiến hành thao tác đọc và ghi file trong HDFS.

Nội dung

- Sử dụng FileSystem API, Hadoop API để ghi file đến HDFS và đọc file từ HDFS về local.

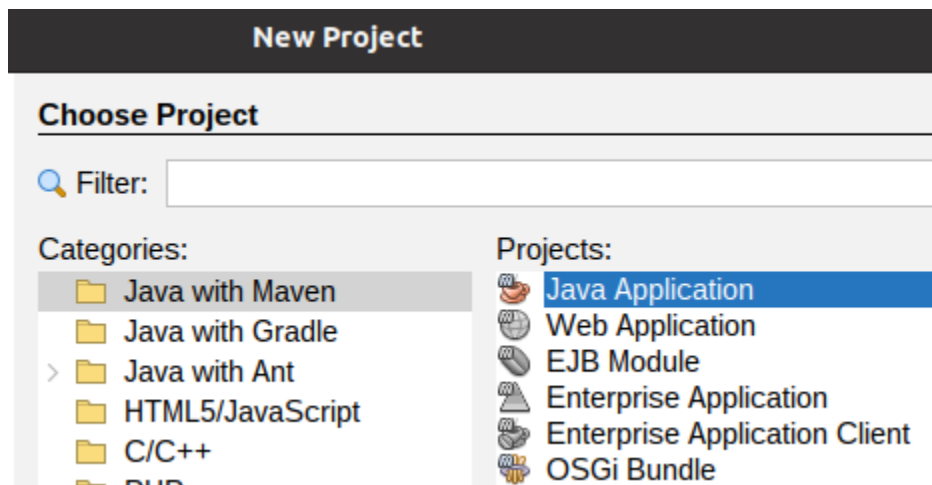
Tài liệu tham khảo

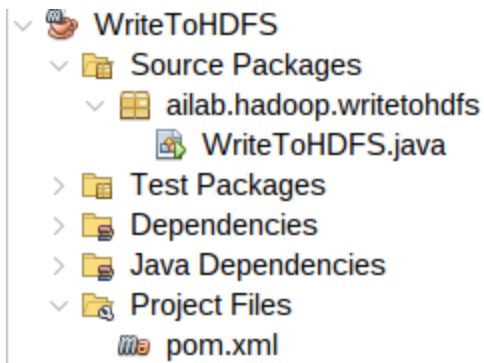
- <https://princetonits.com/blog/technology/using-file-system-api-to-read-and-write-data-to-hdfs/>

HƯỚNG DẪN THỰC HIỆN:

- Đọc dữ liệu từ HDFS và ghi dữ liệu vào HDFS có thể được thực hiện theo nhiều cách. Bây giờ chúng ta hãy bắt đầu bằng cách sử dụng FileSystem API để tạo và ghi vào một tập tin trong HDFS, tiếp theo là một ứng dụng để đọc một tập tin từ HDFS và ghi nó lại vào hệ thống tệp cục bộ.

Bước 1: Tạo một chương trình Java từ Netbean, đặt tên là “**WriteToHDFS**” như sau:





Bước 2: Cập nhật file cấu hình khai báo các dependencies, pom.xml, thêm vào các mô tả bên dưới:

```
<dependencies>
  <!-- Hadoop -->
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-hdfs</artifactId>
    <version>3.2.4</version>
    <exclusions>
      <exclusion>
        <groupId>javax.servlet</groupId>
        <artifactId>*</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-client</artifactId>
    <version>3.2.4</version>
    <exclusions>
      <exclusion>
        <groupId>javax.servlet</groupId>
        <artifactId>*</artifactId>
      </exclusion>
    </exclusions>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

Bước 3: *Viết mã cho việc đọc một tập tin từ hệ thống tệp cục bộ và ghi nội dung vào HDFS.*

```
package ailab.hadoop.writetohdfs;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.util.Tool;
import java.io.BufferedInputStream;
import java.io.FileInputStream;
import java.io.InputStream;
import java.io.OutputStream;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IOUtils;
import org.apache.hadoop.util.ToolRunner;

public class WriteToHDFS extends Configured implements Tool {
    public static final String FS_PARAM_NAME = "fs.defaultFS";
    public int run(String[] args) throws Exception {

        if (args.length < 2) {
            System.err.println("WriteToHDFS [local input path] [hdfs output path]");
            return 1;
        }

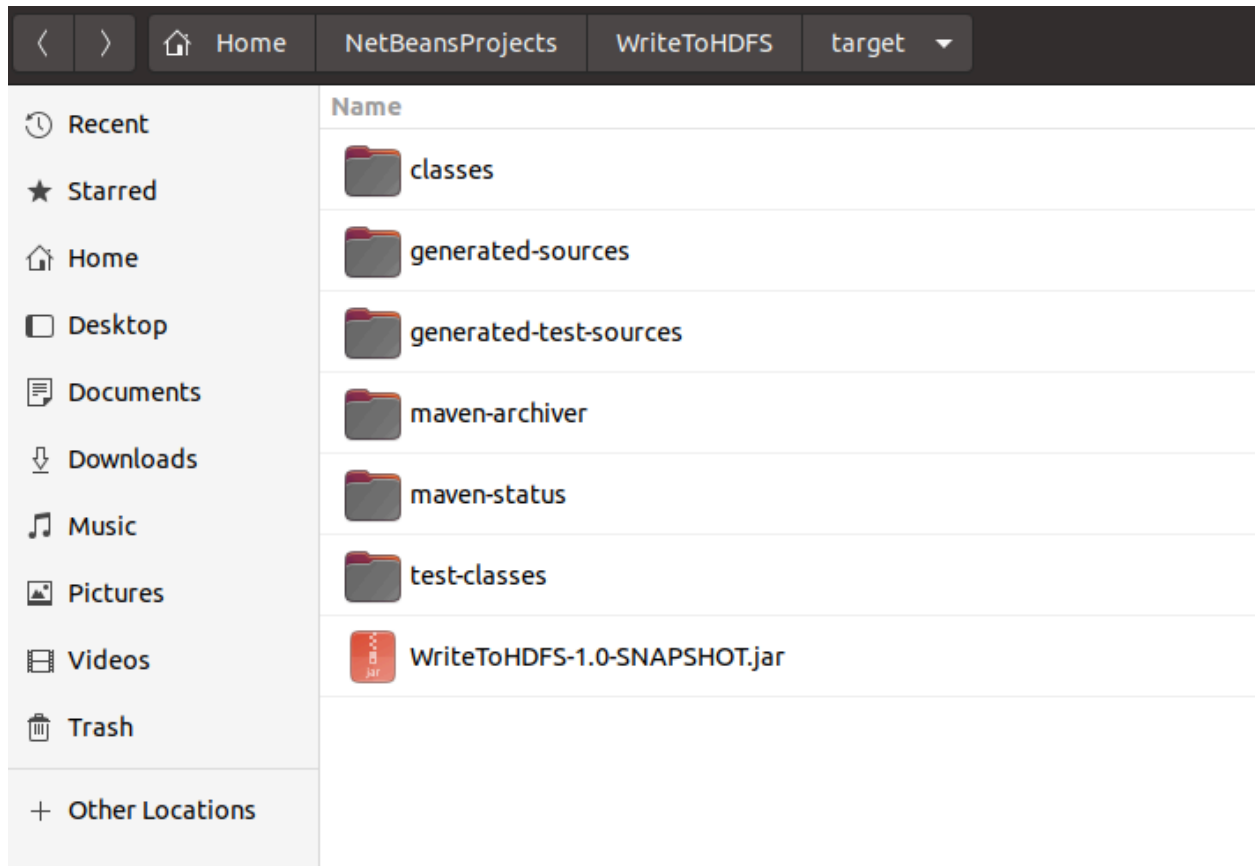
        String localInputPath = args[0];
        Path outputPath = new Path(args[1]);
        Configuration conf = getConf();
        System.out.println("configured filesystem = " + conf.get(FS_PARAM_NAME));
        FileSystem fs = FileSystem.get(conf);
        if (fs.exists(outputPath)) {
            System.err.println("output path exists");
            return 1;
        }
        OutputStream os = fs.create(outputPath);
        InputStream is = new BufferedInputStream(new FileInputStream(localInputPath));
```

```
        IOUtils.copyBytes(is, os, conf);
        return 0;
    }

    public static void main(String[] args) throws Exception {
        int returnCode = ToolRunner.run(new WriteToHDFS(), args);
        System.exit(returnCode);
    }
}
```

Bước 4: Build chương trình để tạo file .jar và chạy mã từ terminal để ghi một tệp mẫu vào HDFS.

- Sau khi build chương trình chúng ta có file jar nằm ở thư mục trên máy tính như sau:
[/home/tinhhuynh/NetBeansProjects/WriteToHDFS/target/WriteToHDFS-1.0-SNAPSHOT.jar](#) (xem hình chụp bên dưới)
- Đường dẫn của lớp chứa hàm main nằm trong package sau:
[ailab.hadoop.writetohdfs.WriteToHDFS](#)



- Chạy chương trình từ terminal như sau để ghi nội dung từ tập tin `/home/hadoop/tmp/data.txt` trên máy cục bộ đến một file mới trên HDFS nằm trong thư mục `"/data"` trên HDFS tên là `/data/Write2HDFS.txt` như sau:

```
hadoop@tinhuynh-PC:~$ hadoop jar
/home/tinhuynh/NetBeansProjects/WriteToHDFS/target/WriteToHDFS-1.0-SNAPSHOT.
jar aila.hadoop.writetohdfs.WriteToHDFS /home/hadoop/tmp/data.txt
/data/Write2HDFS.txt
```

Bước 5: Xác minh liệu tệp đã được ghi vào HDFS và kiểm tra nội dung của tệp.

```
hadoop@tinhuynh-PC:~$ hadoop fs -cat /data/Write2HDFS.txt
```

Show entries

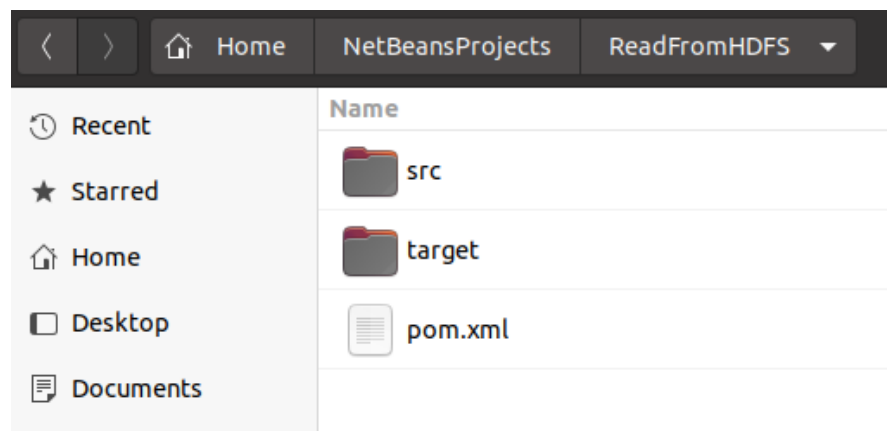
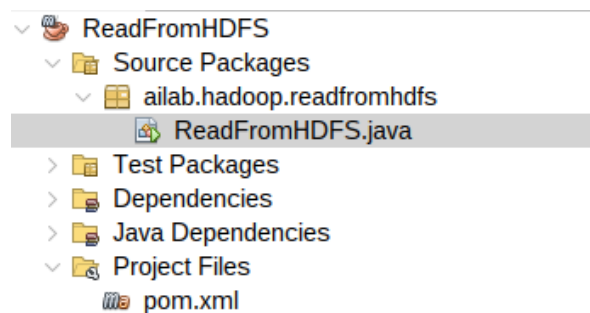
Search:

<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	-rw-r--r--	hadoop	supergroup	120 B	May 22 09:24	1	128 MB	Write2HDFS.txt	
<input type="checkbox"/>	drwxr-xr-x	hadoop	supergroup	0 B	May 20 15:02	0	0 B	testing-data	
<input type="checkbox"/>	drwxr-xr-x	hadoop	supergroup	0 B	May 20 14:23	0	0 B	training-data	

Showing 1 to 3 of 3 entries

```
hadoop@tinhuynh-PC:~$ hadoop fs -cat /data/Write2HDFS.txt
Hom nay troi nang chang chang
Meo con di hoc chang mang thu gi
Chi mang mot mau banh mi
Va mang mot cay but chi con con
```

Bước 6: Tiếp theo, tương tự như trên chúng ta viết một java project, ứng dụng để đọc tệp mà chúng ta vừa tạo trong HDFS và ghi nội dung của nó trở lại hệ thống tệp cục bộ như sau:



```

package ailab.hadoop.readfromhdfs;
import java.io.BufferedOutputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IOUtils;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
public class ReadFromHDFS extends Configured implements Tool {
    public static final String FS_PARAM_NAME = "fs.defaultFS";
    public int run(String[] args) throws Exception {
        if (args.length < 2) {
            System.err.println("ReadFromHDFS [hdfs input path] [local output
path]");
            return 1;
        }

        Path inputPath = new Path(args[0]);
        String localOutputPath = args[1];
        Configuration conf = getConf();
        System.out.println("configured filesystem = " +
conf.get(FS_PARAM_NAME));
        FileSystem fs = FileSystem.get(conf);
        InputStream is = fs.open(inputPath);
        OutputStream os = new BufferedOutputStream(new
FileOutputStream(localOutputPath));
        IOUtils.copyBytes(is, os, conf);
        return 0;
    }

    public static void main( String[] args ) throws Exception {
        int returnCode = ToolRunner.run(new ReadFromHDFS(), args);
        System.exit(returnCode);
    }
}

```

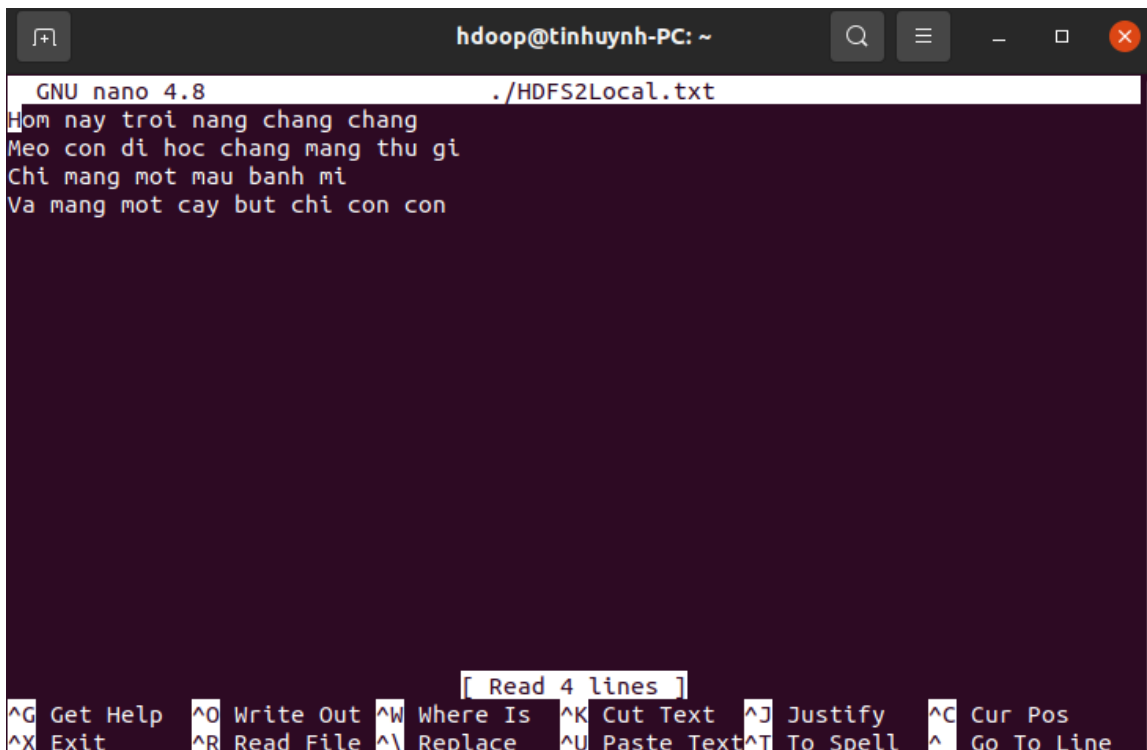
Bước 5: Xuất tệp Jar và chạy mã từ terminal để ghi một tệp mẫu vào HDFS.

```
hadoop@tinhuynh-PC:~$ hadoop jar
/home/tinhuynh/NetBeansProjects/ReadFromHDFS/target/ReadFromHDFS-1.0-SNAPSHOT.
jar ailab.hadoop.readfromhdfs.ReadFromHDFS /data/Write2HDFS.txt
/home/hadoop/HDFS2Local.txt
```

Bước 6: Xác minh xem tệp tin đã được ghi lại vào hệ thống tệp cục bộ chưa.

```
hadoop@tinhuynh-PC:~$ nano ./HDFS2Local.txt
```

```
hadoop@tinhuynh-PC:~$ ls
dfsdata      hadoop-3.2.4      hadoop-3.2.4.tar.gz.1  tmp
downloads    hadoop-3.2.4.tar.gz  HDFS2Local.txt         tmpdata
```



```
GNU nano 4.8                               ./HDFS2Local.txt
Hôm nay trời nắng chang chang
Mèo con đi học chàng mang thu gì
Chỉ mang một màu bánh mì
Và mang một cây bút chì con con

[ Read 4 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

- Đọc, tìm hiểu, phân tích mã nguồn chương trình
-