# Comprehensive Notes on HBase Management and Operations

## 1. Managing HBase with Cloudera Manager

- Role Requirements: To manage HBase effectively, roles like Operator, Configurator, Cluster Administrator, or Full Administrator are required.

- Functions:  
 - Configuration: Cloudera Manager configures and manages HBase settings, ensuring services can be properly handled.  
 - Graceful Shutdown: Ensures that the regions hosted by a RegionServer are moved to other RegionServers before shutting down. This is done to prevent data loss.  
 - Region Mover Threads: To accelerate rolling restarts, adjust this property (recommended: 5 or fewer threads).

## 2. Creating the HBase Root Directory

- Automatic Creation: The root directory is created when adding the HBase service.

- Manual Creation:  
 - Go to \*HBase > Status tab\*, select \*Create Root Directory\*, and confirm.

## 3. Data Insertion Using HBase Shell

- Commands:

- `put` command: Used for inserting data.  
 - Syntax:  
 ```  
 put '<table\_name>', '<row>', '<column\_family:column\_name>', '<value>'  
 ```  
 - Example:  
 ```  
 put 'Emp', '1', 'Personal Data:EName', 'Arshad'  
 put 'Emp', '1', 'Personal Data:EState', 'UP'  
 ```

## 4. Data Insertion Using Java API

- Classes Used:

- Configuration: Loads HBase configurations.  
 ```  
 Configuration conf = HbaseConfiguration.create();  
 ```  
 - HTable: Interface to communicate with HBase.  
 ```  
 HTable hTable = new HTable(conf, tableName);  
 ```  
 - Put Class:  
 ```  
 Put p = new Put(Bytes.toBytes("row1"));  
 p.add(Bytes.toBytes("column\_family"), Bytes.toBytes("column\_name"), Bytes.toBytes("value"));  
 ```  
 - Saving Data:  
 ```  
 hTable.put(p);  
 hTable.close();  
 ```

## 5. Data Update Using HBase Shell

- Update Command:  
 - Use the `put` command to update existing cell values.  
 - Example:  
 ```  
 put 'Emp', 'row1', 'Personal:EState', 'NewState'  
 ```

## 6. Updating Data Using Java API

- Steps:  
 1. Instantiate Configuration.  
 2. Create HTable instance.  
 3. Instantiate Put Class with the new value.  
 4. Use `put()` method of HTable to save.

## 7. Reading Data from HBase Shell

- Get Command:  
 - Used to read specific rows.  
 ```  
 get '<table\_name>', '<row>'  
 ```  
 - Reading Specific Columns:  
 ```  
 get '<table\_name>', '<row>', {COLUMN => 'column\_family:column\_name'}  
 ```  
 - Example:  
 ```  
 get 'Emp', '1', {COLUMN => 'Personal:EName'}  
 ```

## 8. Reading Data Using Java API

- Classes:  
 - Get Class:  
 ```  
 Get get = new Get(Bytes.toBytes("row1"));  
 ```  
 - HTable Method:  
 ```  
 Result result = hTable.get(get);  
 byte[] value = result.getValue(Bytes.toBytes("Personal"), Bytes.toBytes("EName"));  
 ```