Command Line vs Jvm System Properties

Command-line properties and JVM system properties are related concepts but serve different purposes in the context of a Java application, including a Spring Boot application.

1. **Command-Line Properties**:
   - Command-line properties are passed directly to the application when it is started via the command line.
   - They are specified after the `java -jar` command and before the name of the JAR file.
   - Command-line properties are typically used to configure application-specific settings or to override default settings defined in the application configuration files.
   - Example: `java -jar my-application.jar --spring.profiles.active=dev`

2. **JVM System Properties**:
   - JVM system properties are set using the `-D` flag when starting the Java Virtual Machine (JVM).
   - They are specified before the `-jar` option when running a Java application.
   - JVM system properties are used to configure aspects of the JVM itself, such as memory settings (`-Xmx`, `-Xms`), system properties (`java.version`, `java.home`), or other JVM-related configurations.
   - Example: `java -Djava.security.debug=all -jar my-application.jar`

In summary, command-line properties are specific to the application being run and are typically used for application-level configuration, while JVM system properties are global settings that affect the behavior of the JVM itself. Both types of properties can be useful for configuring and tuning a Spring Boot application, but they serve different purposes and are set using different syntax when starting the application.