When using Java in AWS Lambda, the cold start time can be longer than when using other languages such as Node.js. This is because the Java Virtual Machine (JVM) takes longer to start up than the Node.js runtime. However, there are several ways to reduce the cold start time when using Java in AWS Lambda.

1. you can package your code and dependencies into a single .jar file. This will reduce the amount of time it takes for AWS Lambda to download and unpack your code, which can help to reduce the cold start time.
2. you can use smaller container sizes. AWS Lambda uses containers to run your code, and the smaller the container, the faster it will start up. You can use the smallest container that can run your code to reduce the cold start time.
3. you can use the AWS Lambda Provisioned Concurrency feature. This feature allows you to keep a certain number of instances of your function “warm” and ready to handle requests at all times, which can help to reduce the cold start time.
4. you can consider using a custom runtime. This allows you to use any other language besides the ones provided by AWS, so you can use the one that you feel more comfortable with and have better cold start times.
5. you can use a framework such as Spring Boot, which has a feature called “spring-cloud-function-context” that allows you to cache the context of your function, reducing the cold start time significantly.

By following these tips, you can reduce the cold start time when using Java in AWS Lambda and ensure that your function is able to respond quickly to requests.