

Deploy Application on cloud

ALGORITHM:

1. `pom.xml`: This is the Maven configuration file that defines the project's dependencies, plugins, and other settings. Here are the key elements:
 - The project's parent is set to `spring-boot-starter-parent` with version `3.1.2`.
 - The project has a group ID `com`, artifact ID `AmazonEC2App`, and version `0.0.1-SNAPSHOT`.
 - The project uses Java version 20.
 - The project has two dependencies: `spring-boot-starter-web` and `spring-boot-starter-test`.
 - The build section defines the `spring-boot-maven-plugin`, which is required for packaging and running Spring Boot applications.
2. `AmazonEc2AppApplication.java`: This is the main class that serves as the entry point for the Spring Boot application. It's annotated with `@SpringBootApplication`, which combines `@Configuration`, `@EnableAutoConfiguration`, and `@ComponentScan`.
 - The `@ComponentScan` annotation is set to scan the package `com.example.demo`.
 - The `main` method starts the Spring Boot application by calling `SpringApplication.run()`.
3. `MainController.java`: This class is a Spring `@RestController`, which means it handles incoming HTTP requests and returns responses.
 - The `hello()` method is mapped to handle the root URL (`/`) and simply returns the string "Hello! This is the Amazon AWS EC2 App...".
4. Overall, this application sets up a basic Spring Boot web application with a single RESTful endpoint that returns a greeting message.
5. When the application is run, it will start a web server and listen on a port (default is usually 8080).
6. Accessing the root URL will trigger the `hello()` method in the `MainController`.