01 - Spring To SpringBoot

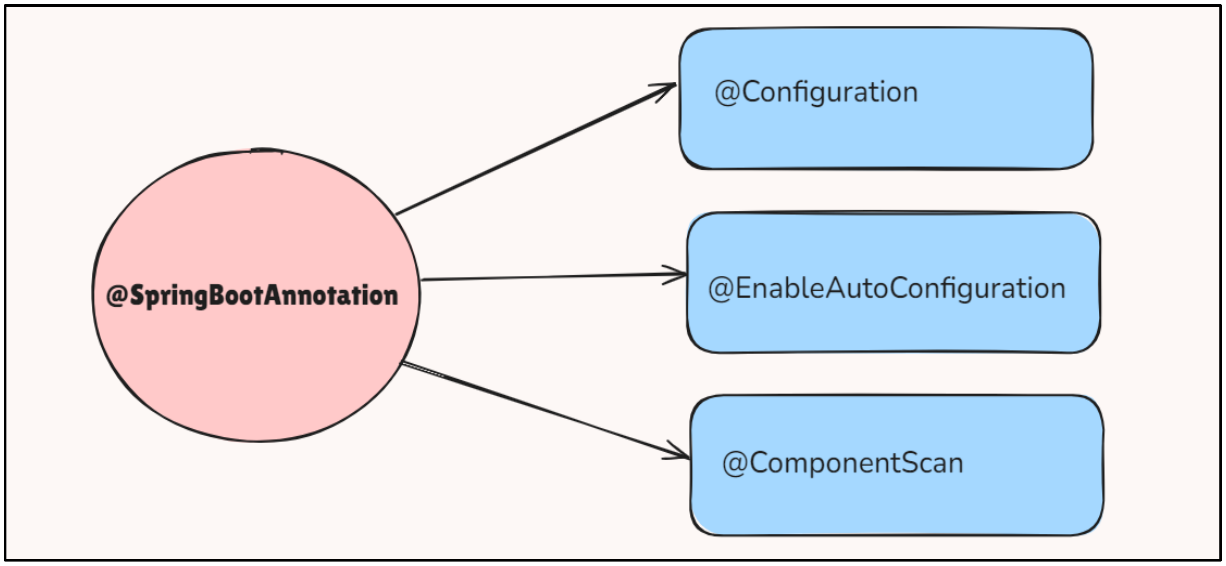
Spring Boot is an opinionated framework that simplifies the development with Spring by providing defaults and reducing configuration efforts.

Key Features of Spring Boot:

- **Simplified Configuration:** It auto-configures Spring applications based on the dependencies present on the classpath.
- **Embedded Servers:** It includes embedded servers like Tomcat, making it easy to run applications without needing to deploy to an external server.
- **Production-Ready Features:** It provides production-ready features such as metrics, health checks and externalize configuration.
- **Spring Boot Starter Projects:** It creates stand-alone Spring Application that can be started using Java-jar.

👉 @SpringBootApplication Annotation:

- The **@SpringBootApplication** annotation is the key to starting a Spring Boot project.
- By using **@SpringBootApplication**, we can effectively perform:
  - Automatic configuration
  - Component scanning
  - Declaring configuration files
- It encapsulates three important annotations and serves as an entry point for bootstrapping the Spring Boot application.



1. **@Configuration**:

- Indicates that the class is a source of bean definitions.
- It allows Java-based configuration of Spring Beans.

2. **@EnableAutoConfiguration**:

- It enables Spring Boot to automatically configure the application based on the dependencies we have added.

  For example, if we have spring-boot-starter-web dependency, Spring Boot will configure everything needed for a web application (like a DispatcherServlet, embedded server, etc.).

3. **@ComponentScan**:

- It helps Spring to scan the current package and its sub-packages for components (such as beans, controllers, services, etc.).
- It helps in auto-detecting Spring beans without needing to manually define them.

Example:

```
import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;



@SpringBootApplication

public class MySpringBootApplication {

    public static void main(String[] args) {

        SpringApplication.run(MySpringBootApplication.class, args);

    }

}
```

- **@SpringBootApplication** tells Spring Boot to enable auto-configuration and component scanning.
- The main() method starts the application by calling **SpringApplication.run()** which boots up the entire Spring context.