---

1. @Component

What it does:

- Marks a **Java class as a Spring-managed bean**.
- Tells Spring: "Create an object of this class automatically and put it in the Spring container."

Example:

```
@Component
public class Address {


	private String addressLine1, city, state, country;
```

Meaning:

- You don't need to write XML to declare this bean.
- Spring will **detect it automatically** when you use component scanning.

2. @Configuration

What it does:

- Marks a class as a **configuration class** (like a replacement for XML configuration).
- Inside this class, you can define beans using **@Bean** methods.

Example:

```
@Configuration
@ComponentScan("com.hcl.autoconfig")
public class AppConfig {

	@Bean
	public Address address() {

		Address address= new Address();
		address.setAddressLine1("CP");
		address.setCity("New Delhi");
		address.setState("Delhi");
		address.setCountry("India");
		return address;


	}
```

Meaning:

- This class defines beans manually in Java code.
- Instead of beans.xml, you use this class to tell Spring what to create.

---

3. @ComponentScan("com.hcl.autoconfig")

What it does:

- Tells Spring **where to look for classes annotated with @Component, @Service, @Repository, @Controller.**
- This is called **component scanning**.

Example:

```
@Configuration
@ComponentScan("com.hcl.autoconfig")
public class AppConfig {
```

Meaning:

- Spring will look in the com.hcl.autoconfig package and automatically create beans for any class annotated with @Component (or related annotations).

---

4. @Bean

What it does:

- Used inside a @Configuration class.
- Defines a method that **returns an object to be managed as a Spring bean**.

Example:

```
@Bean
	public Address address() {

		Address address= new Address();
		address.setAddressLine1("CP");
```

```
        address.setCity("New Delhi");
        address.setState("Delhi");
        address.setCountry("India");
        return address;


    }
```

Meaning:

- Spring will call this method and put the returned object (MyService) into the container.
- This is similar to declaring a bean in XML.

Quick Analogy

Annotation	What it Means
@Component	"This class is a bean. Manage it automatically."
@Configuration	"This class provides bean definitions (like XML config)."
@ComponentScan	"Look in this package for @Component-annotated classes."
@Bean	"Call this method to create a bean."
