

# Spring Framework Deep Dive: Days 31-40

Master Advanced Concepts in Spring AOP, JDBC, Transactions, REST APIs, and Spring Boot Configuration



# **What Is a Pointcut in Spring AOP?**

## **Day 31**

DURGASOFT

Page 2



# Understanding Pointcuts

DURGASOFT

Page 3



## Where to Apply

Defines which methods, packages, or classes should be intercepted by aspects



## Pattern Matching

Uses Pointcut Expressions to match method signatures accurately



## Automatic Execution

Without a pointcut, your advice won't know where to run

A **Pointcut** in Spring AOP defines *where* an aspect should apply. It tells Spring **which methods, which packages, or which classes** should be intercepted. It uses patterns called **Pointcut Expressions** to match method signatures.

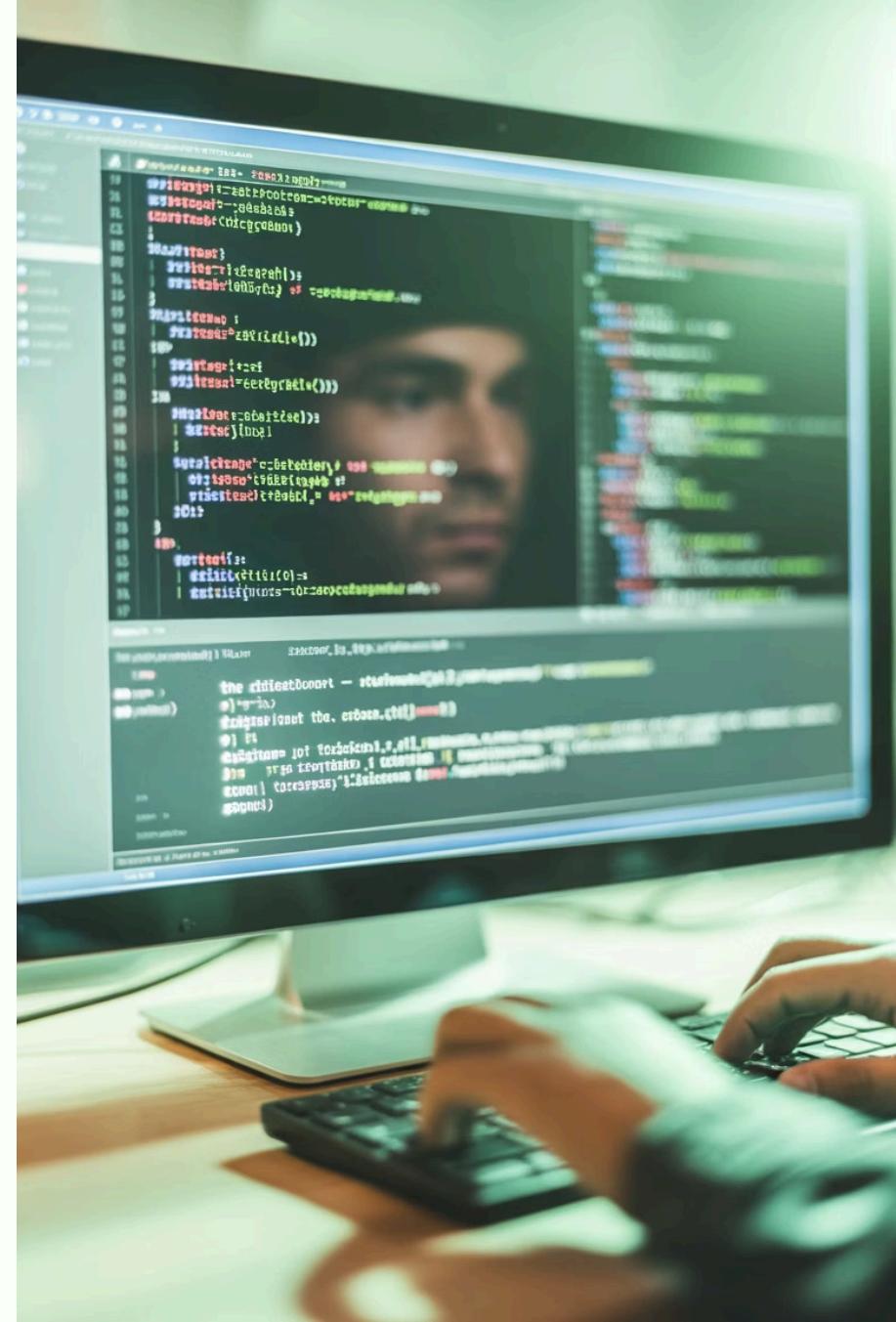
# Pointcut Code Example

DURGASOFT

Page 4

```
@Aspect  
@Component  
public class SecurityAspect {  
  
    @Pointcut("execution(* com.durgasoft.service.*.*(..))")  
    public void serviceLayer() {}  
  
    @Before("serviceLayer()")  
    public void checkAccess() {  
        System.out.println("Security check...");  
    }  
}
```

Here, the pointcut matches **all methods** inside the service package. Spring runs `checkAccess()` *before* every method in that layer. This powerful pattern matching allows you to apply cross-cutting concerns like security, logging, and performance monitoring across your entire application with minimal code.





**"Focus  
your  
effort  
where it  
matters —  
let  
pointcuts  
handle the  
rest."**

## **Join DURGASOFT for Expert Spring Training**

 *If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.*

 **Contact:** 9246212143, 8885252627

 [www.durgasoftonline.com](http://www.durgasoftonline.com)

 [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)

```
29 |  
25 |     .cianstycol{{}},lataba();  
21 |     preresctprol{ls}[ocegco--;Ja|z{})"peensauieectaonetion()  
18 |  
12 |     yad|--catoionriccne{{}} gätref"Jde[el]t)ie->  
15 |  
25 |     roeaillér-{ ||;  
22 |     Stenn"coetetibnesto{stactos|):,();  
25 |     praqaotucrtiasg)  
35 |     enve-afl'llicathtte">,Cätdehéliona{{})  
55 |     enve-afl'llicathtte">,Cätdehéliona{{})
```

# What Is Spring JdbcTemplate?

# Day 32

DURGASOFT

Page 7

# Simplifying Database Operations

DURGASOFT

Page 8



## Before JdbcTemplate

Manual connection creation, statement handling, result set closing, exception management

## With JdbcTemplate

Spring handles all boilerplate automatically

## Result

Focus only on SQL and business logic

**JdbcTemplate** is Spring's helper class that makes database operations easy. It removes the need to write long JDBC code like creating connections, handling statements, closing result sets, and managing exceptions. Spring does all this automatically so you can focus only on the actual SQL and business logic.

# JdbcTemplate in Action

DURGASOFT

Page 9

```
@Autowired  
private JdbcTemplate jdbcTemplate;  
  
public int getUserCount() {  
    return jdbcTemplate.queryForObject(  
        "SELECT COUNT(*) FROM users",  
        Integer.class  
    );  
}
```

## What JdbcTemplate Handles

- Connection creation and management
- Statement preparation
- Result set processing
- Exception handling and translation
- Resource cleanup

No connection creation, no manual closing, no exception handling. JdbcTemplate simplifies everything and is widely used in real projects across industries.

**"Write  
less code,  
avoid  
boilerplat  
e – focus  
on real  
logic."**

## **Join DURGASOFT for Expert Spring Training**

 *If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.*

 **Contact:** 9246212143, 8885252627

 [www.durgasoftonline.com](http://www.durgasoftonline.com)

 [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)



# What Is Spring Transaction Management?

## Day 33

DURGASOFT

Page 12

# Ensuring Data Integrity

DURGASOFT

Page 13

## Safe Operations

Protects your database operations from partial failures and data corruption

## Consistency

Ensures all operations complete successfully or none at all

## Automatic Rollback

If something fails, Spring can rollback the entire transaction

Spring Transaction Management ensures your database operations are **safe, consistent, and error-free**. If something fails in the middle of a process (like money transfer or order placement), Spring can **rollback** the entire transaction so your data stays correct. This is critical for enterprise and cloud applications where data integrity cannot be compromised.



# Transaction Management Example

DURGASOFT

Page 14

```
@Service  
@Transactional  
public class PaymentService {  
  
    public void makePayment() {  
        debitAmount();  
        creditAmount();  
        // If any step fails, Spring rolls back automatically  
    }  
}
```

**@Transactional** is all you need – Spring handles connection management, rollback, and commit behind the scenes. This single annotation provides enterprise-grade transaction management with ACID properties, ensuring your business operations are reliable and consistent.



**"Reliable  
transactions  
create reliable  
systems — and  
reliable  
developers."**

DURGASOFT

Page 15

## **Join DURGASOFT for Expert Spring Training**

 *If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.*

 **Contact:** 9246212143, 8885252627

 [www.durgasoftonline.com](http://www.durgasoftonline.com)

 [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)

# Spring REST Controller Mapping

## Day 34

GET, POST, PUT, DELETE

DURGASOFT

Page 17



# Building REST APIs Made Easy

DURGASOFT

Page 18



## @GetMapping

Retrieves data from the server. Used for reading information without modifying it.



## @PostMapping

Sends data to create new resources on the server. Used for creating new records.



## @PutMapping

Updates existing resources completely. Replaces the entire resource with new data.



## @DeleteMapping

Removes resources from the server. Used for deletion operations.

Spring Boot makes REST API development very easy using simple annotations. Each annotation maps an HTTP method to a Java method. This is the foundation of building cloud-ready microservices and backend APIs.

# REST Controller Code Example

DURGASOFT

Page 19

```
@RestController  
public class UserController {  
  
    @GetMapping("/users")  
    public String getUsers() {  
        return "All Users";  
    }  
  
    @PostMapping("/users")  
    public String createUser() {  
        return "User Created";  
    }  
}
```

## Perfect For:

- Mobile app backends
- Web application APIs
- Microservices architecture
- Cloud-native applications
- Third-party integrations

With these small annotations, Spring automatically handles the routing, request processing, and response formatting. This simplicity accelerates development and makes maintaining APIs straightforward.

**"APIs are  
the new  
electricity  
— learn  
them, and  
every door  
opens."**

## **Join DURGASOFT for Expert Spring Training**

 *If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.*

 **Contact:** 9246212143, 8885252627

 [www.durgasoftonline.com](http://www.durgasoftonline.com)

 [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)



# What Is Spring Boot Auto-Configuration?

## Day 35

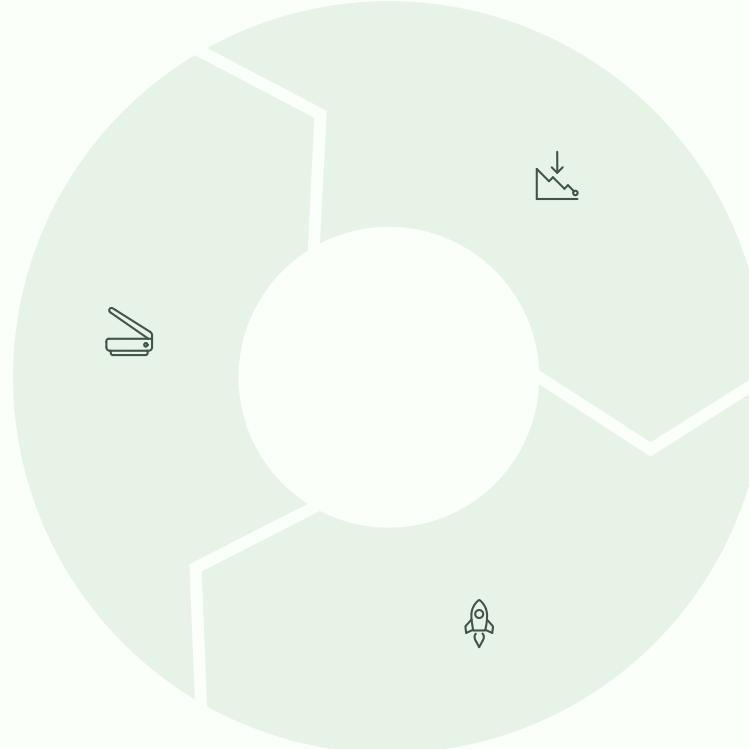
DURGASOFT

Page 22

# Smart Configuration Without Effort

DURGASOFT

Page 23



## Detect Dependencies

Spring Boot scans your classpath

## Configure Automatically

Sets up beans and configurations

## Ready to Run

Your app starts immediately

Spring Boot's **Auto-Configuration** automatically configures your application based on the libraries present in your classpath. For example, if Spring sees a DB driver, it configures a DataSource. If it sees Spring Web, it configures a DispatcherServlet. This reduces setup time and lets you focus on business logic instead of boilerplate configuration.

# Auto-Configuration in Action

DURGASOFT

Page 24

```
@SpringBootApplication  
public class AppStarter {  
    public static void main(String[] args) {  
        SpringApplication.run(AppStarter.class, args);  
    }  
}
```



## **@EnableAutoConfiguration**

Activates automatic bean configuration based on classpath



## **@ComponentScan**

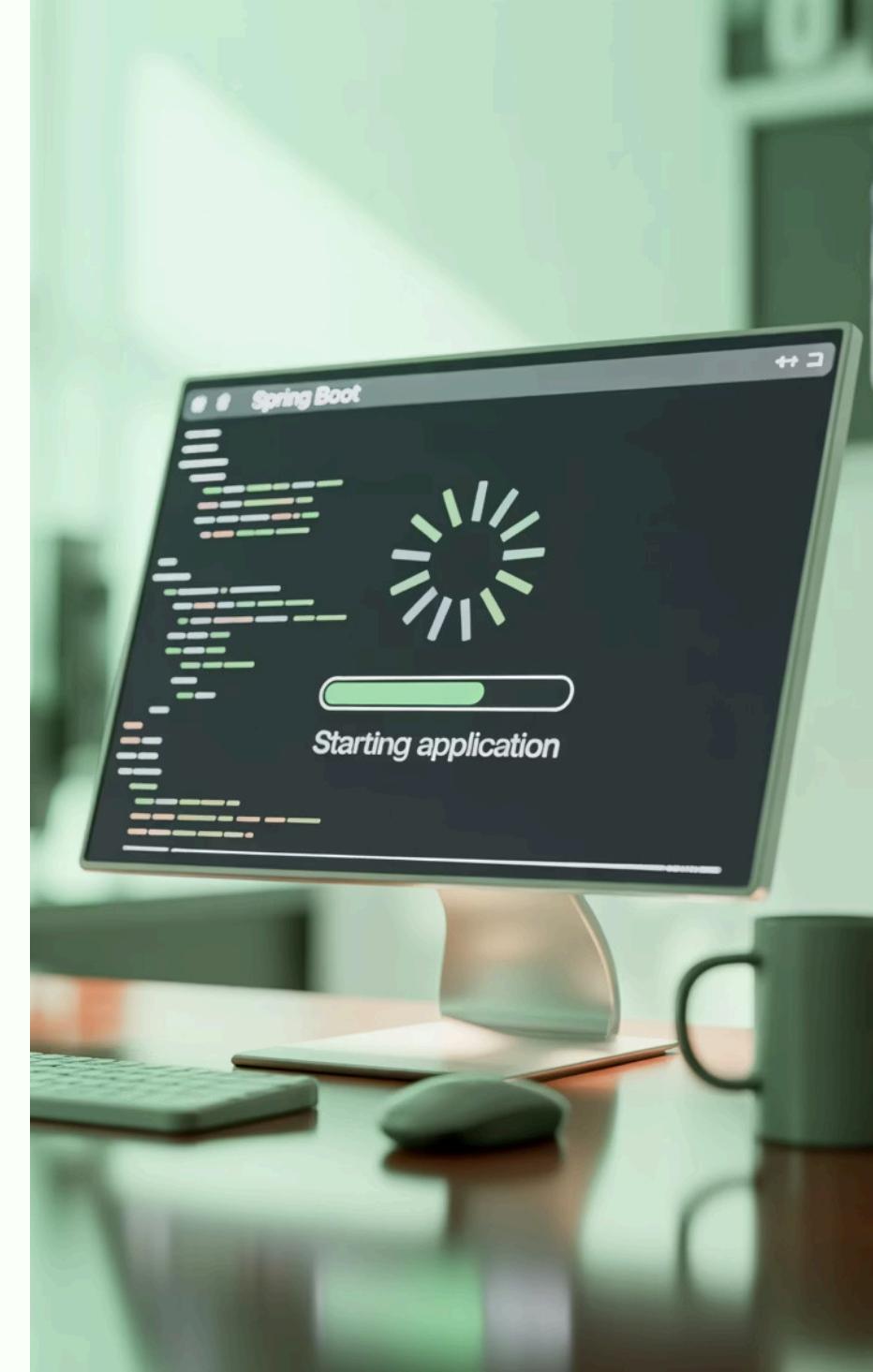
Scans for Spring components, services, and controllers



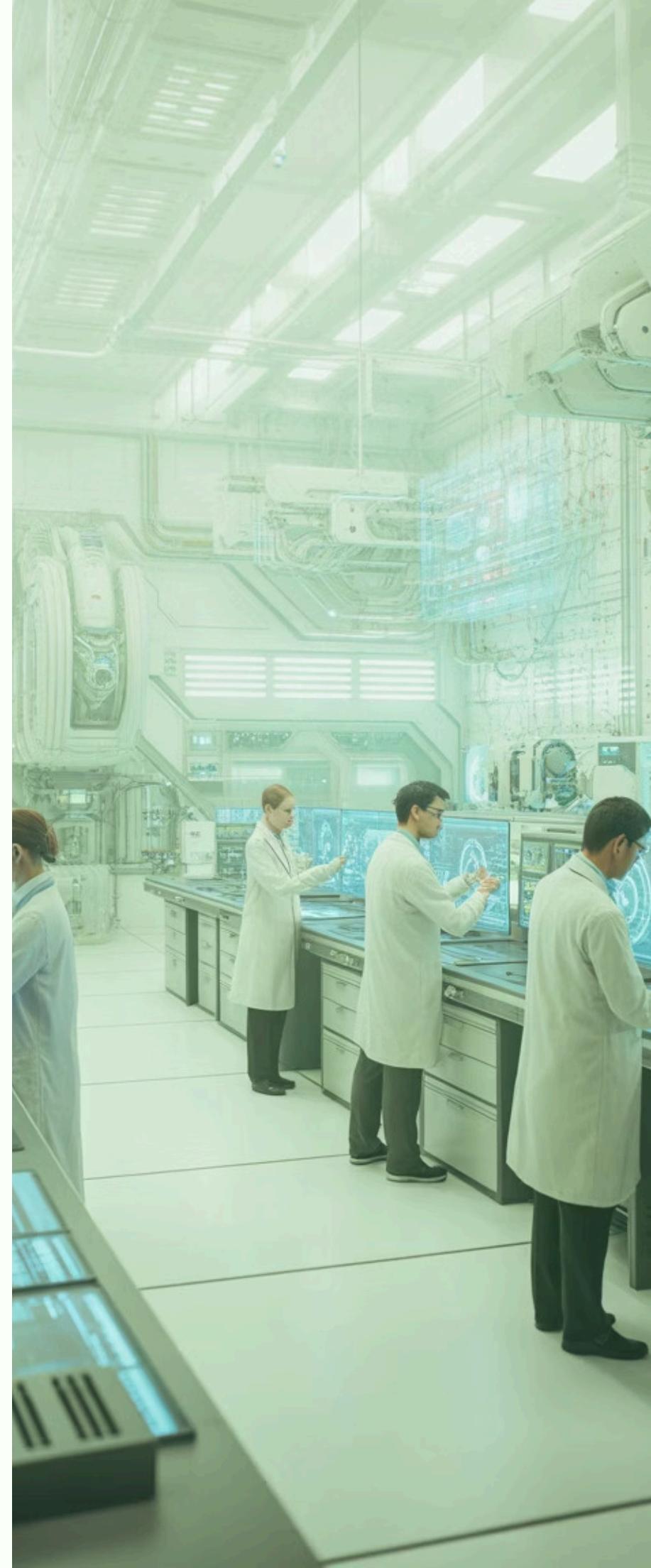
## **@Configuration**

Marks the class as a source of bean definitions

This single annotation triggers auto-configuration and sets up the entire Spring Boot project automatically. Perfect for rapid development and cloud deployment.



"Let  
Spring  
Boot  
handle the  
setup —  
you focus  
on  
building  
the  
future."



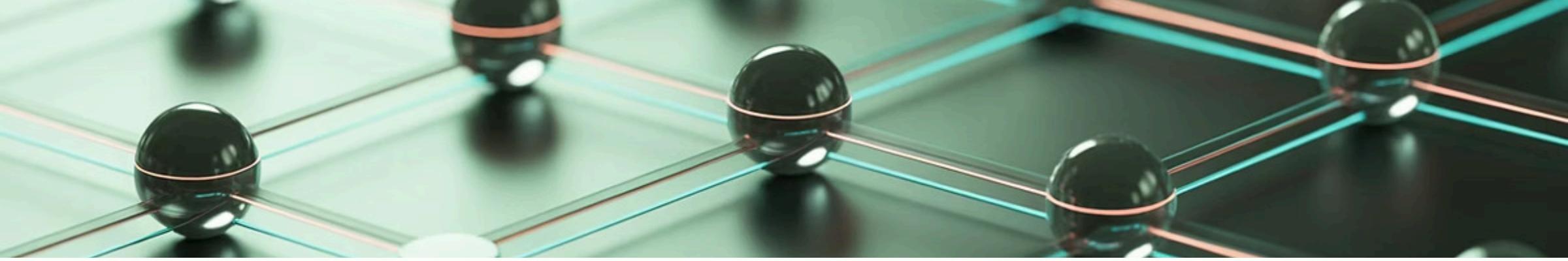
## **Join DURGASOFT for Expert Spring Training**

 *If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.*

 **Contact:** 9246212143, 8885252627

 [www.durgasoftonline.com](http://www.durgasoftonline.com)

 [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)



# **What Is Spring Boot Starter Dependency?**

## **Day 36**

**DURGASOFT**

Page 27

# One Starter, Everything Included

DURGASOFT

Page 28

## Without Starters

Add 10-12 individual dependencies manually. Risk version conflicts and compatibility issues.

## With Starters

Add ONE starter dependency. Spring pulls everything needed automatically with compatible versions.

Spring Boot **Starters** are ready-made dependency bundles that simplify project setup. Instead of adding multiple individual dependencies, you add **one starter**, and Spring pulls everything needed. Example: `spring-boot-starter-web` gives you Tomcat, Spring MVC, JSON support – all in one line.

# Popular Spring Boot Starters

DURGASOFT

Page 29

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```



## Spring MVC

Web application framework for building REST APIs and web pages



## Embedded Tomcat

Built-in web server, no external server installation needed



## Jackson JSON

Automatic JSON conversion for REST API responses



## Validation

Bean validation libraries for input validation

Starters save time and avoid version conflicts – perfect for cloud and microservices apps.

**"Use  
smart  
shortcuts  
— they  
make you  
faster and  
more  
productiv  
e."**

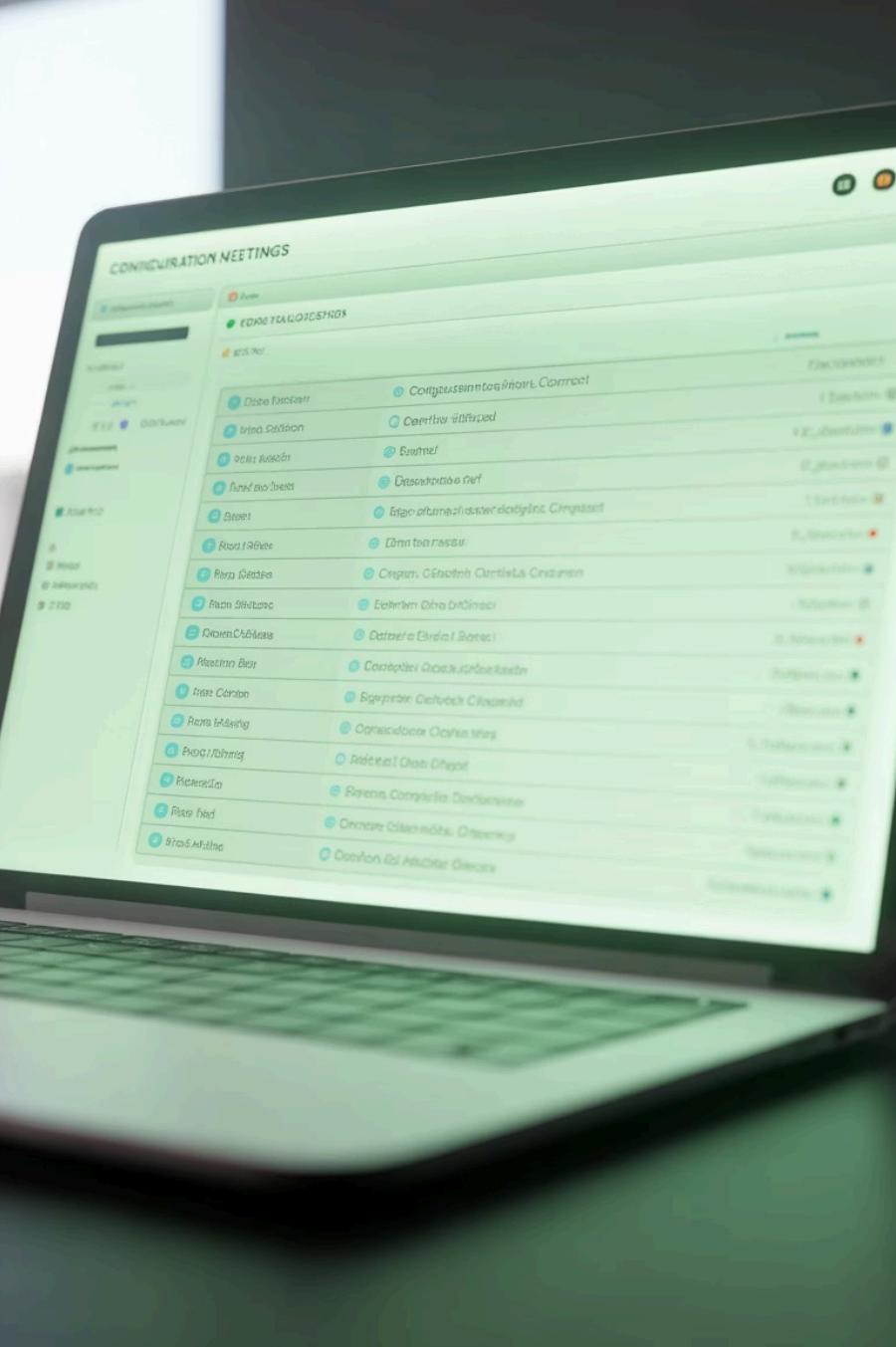
## **Join DURGASOFT for Expert Spring Training**

 *If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.*

 **Contact:** 9246212143, 8885252627

 [www.durgasoftonline.com](http://www.durgasoftonline.com)

 [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)



# What Is application.properties?

## Day 37

DURGASOFT

Page 32

# Central Configuration Management

DURGASOFT

Page 33

## Server Settings

Configure port numbers, context paths, and server properties

## Database Details

Define connection URLs, usernames, passwords, and driver classes

## Logging Levels

Set logging verbosity for different packages and classes

## Custom Values

Store application-specific configuration values

**application.properties** is the main configuration file in Spring Boot where you define settings like server port, database details, logging levels, and custom values. Instead of writing configurations inside code, you keep them here – making your application flexible and easy to manage in different environments (dev, test, prod).

# Sample Configuration

DURGASOFT

Page 34

```
server.port=8081

spring.datasource.url=jdbc:mysql://localhost:3306/durgadb
spring.datasource.username=root
spring.datasource.password=root

spring.jpa.show-sql=true

app.welcome.message=Welcome to DURGASOFT
```

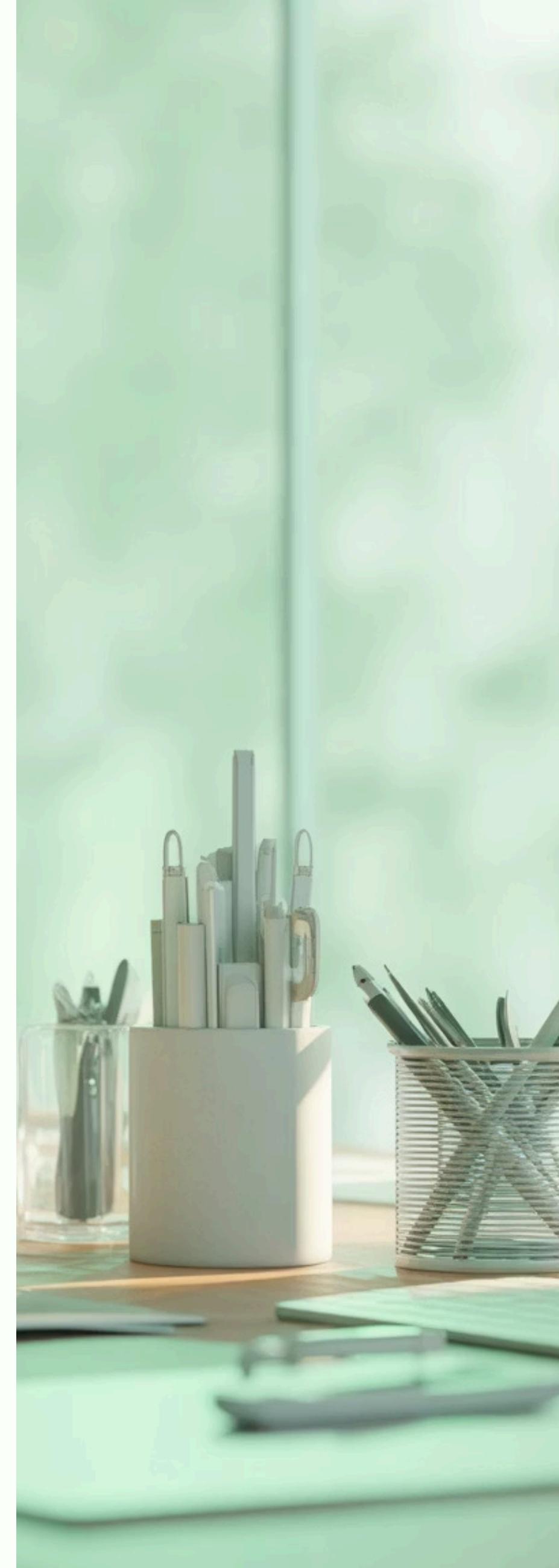
## Reading Properties in Code

```
@Value("${app.welcome.message}")
private String welcomeMsg;

@ConfigurationProperties(prefix="app")
public class AppConfig {
    private String welcomeMessage;
    // getters and setters
}
```

Spring Boot automatically reads these values at startup. You can also inject them into your Beans using **@Value** or **@ConfigurationProperties**.

"Good configuration brings good clarity — and clarity builds powerful applications."



## **Join DURGASOFT for Expert Spring Training**

 *If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.*

 **Contact:** 9246212143, 8885252627

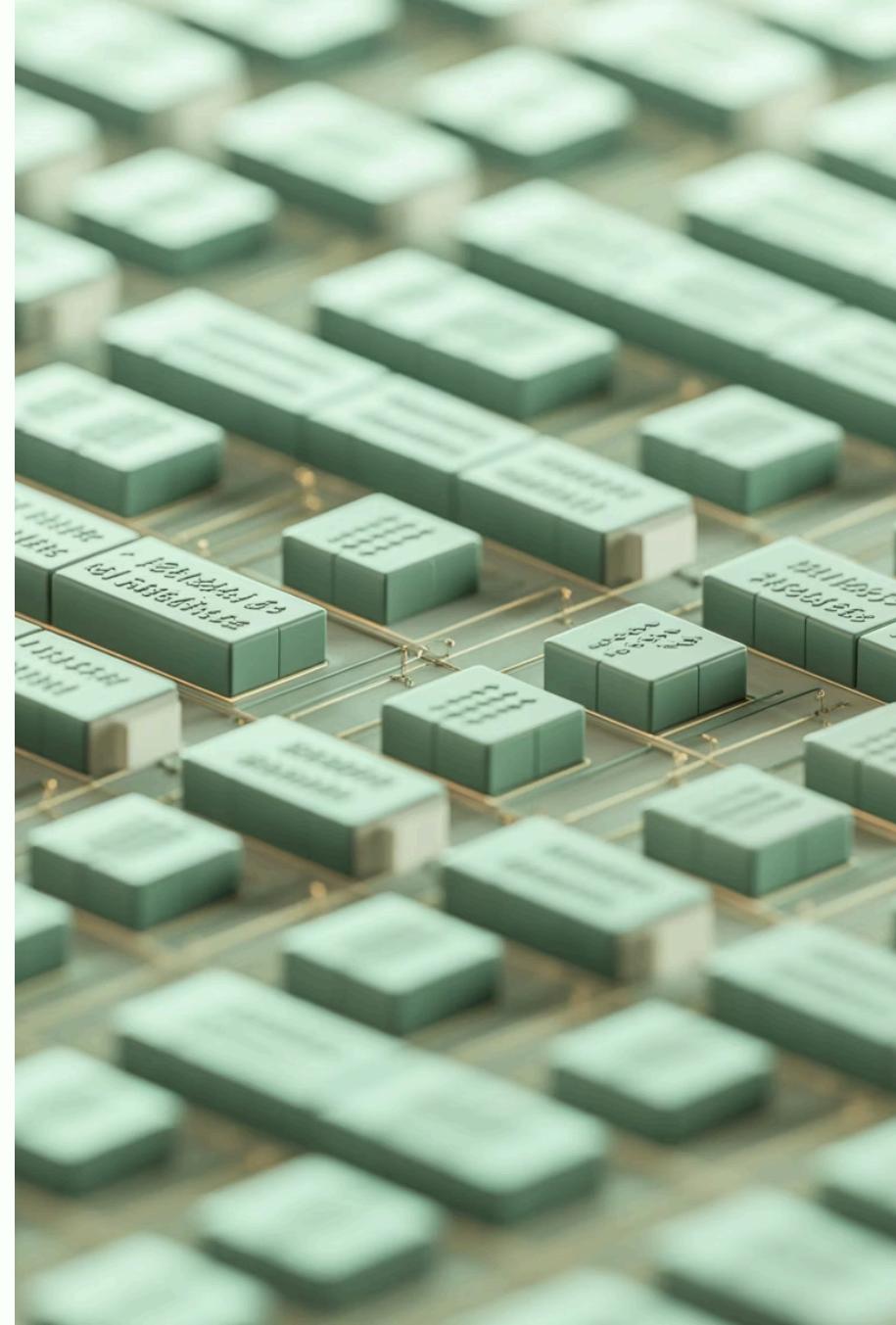
 [www.durgasoftonline.com](http://www.durgasoftonline.com)

 [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)

# What Is application.yml? Day 38

DURGASOFT

Page 37



# YAML: Cleaner, Structured Configuration

DURGASOFT

Page 38



## Hierarchical Structure

Uses tree-like format with indentation for nested settings



## Better Readability

Cleaner syntax with less repetition makes it easier to read



## Natural Grouping

Related properties are visually grouped together

**application.yml** is an alternative to application.properties but offers **cleaner, structured, and more readable** configuration. It uses a tree-like format, making it easy to configure nested settings like DB, logging, security, and microservice-level options. Most modern Spring Boot projects prefer YAML for clarity.

# YAML Configuration Example

DURGASOFT

Page 39

## application.yml

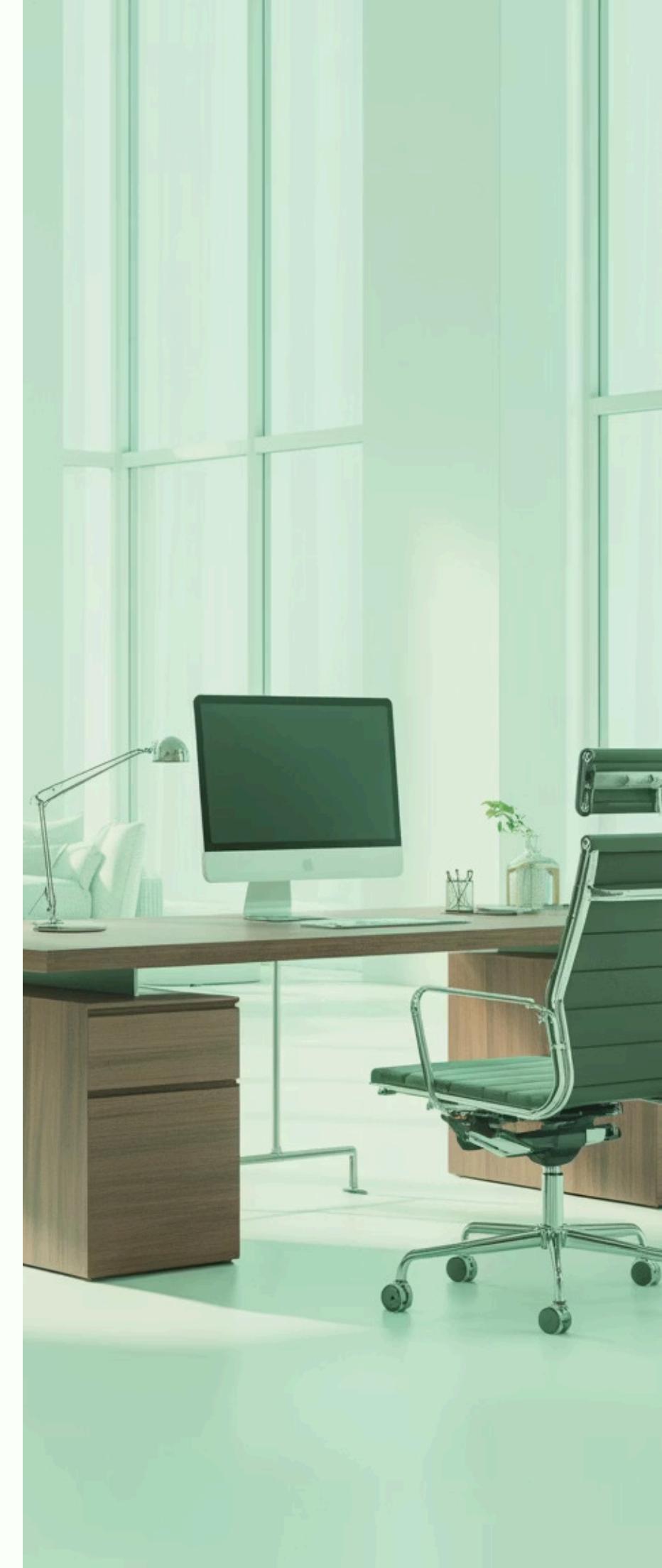
```
server:  
  port: 9090  
  
spring:  
  datasource:  
    url: jdbc:mysql://localhost:3306/durgadb  
    username: root  
    password: root  
  
  jpa:  
    show-sql: true  
  
app:  
  welcome: "Welcome to DURGASOFT"  
  features:  
    - Spring Boot  
    - Microservices  
    - Cloud
```

## Equivalent .properties

```
server.port=9090  
spring.datasource.url=jdbc:mysql://localhost:3306/durgadb  
spring.datasource.username=root  
spring.datasource.password=root  
spring.jpa.show-sql=true  
app.welcome=Welcome to DURGASOFT  
app.features[0]=Spring Boot  
app.features[1]=Microservices  
app.features[2]=Cloud
```

YAML supports nesting, indentation, and grouping – making large cloud projects easier to manage and maintain.

"Clear configuration leads to clean development — YAML keeps everything simple."



## **Join DURGASOFT for Expert Spring Training**

 If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.

 **Contact:** 9246212143, 8885252627

 [www.durgasoftonline.com](http://www.durgasoftonline.com)

 [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)



# What Are Spring Profiles?

## Day 39

**dev, test, prod**

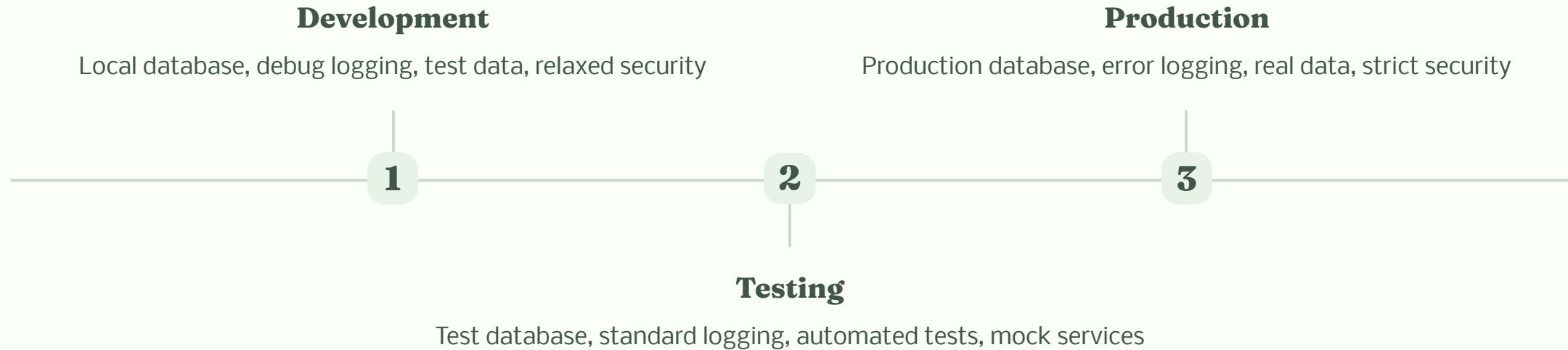
**DURGASOFT**

Page 42

# Environment-Specific Configuration

DURGASOFT

Page 43



Spring Profiles allow you to run your application with **different configurations** for different environments like **development**, **testing**, and **production**. Instead of changing files manually every time, Spring automatically loads the right settings based on the active profile. This prevents configuration errors and makes deployments safer.

# Using Spring Profiles

DURGASOFT

Page 44

## application-dev.yml

```
server:  
  port: 8081  
  
spring:  
  datasource:  
    url:  
      jdbc:mysql://localhost:3306/devdb  
    username: devuser  
    password: devpass
```

## application-test.yml

```
server:  
  port: 8082  
  
spring:  
  datasource:  
    url:  
      jdbc:mysql://testserver/testdb  
    username: testuser  
    password: testpass
```

## application-prod.yml

```
server:  
  port: 80  
  
spring:  
  datasource:  
    url:  
      jdbc:mysql://prod-server/db  
    username: produser  
    password: prodpass
```

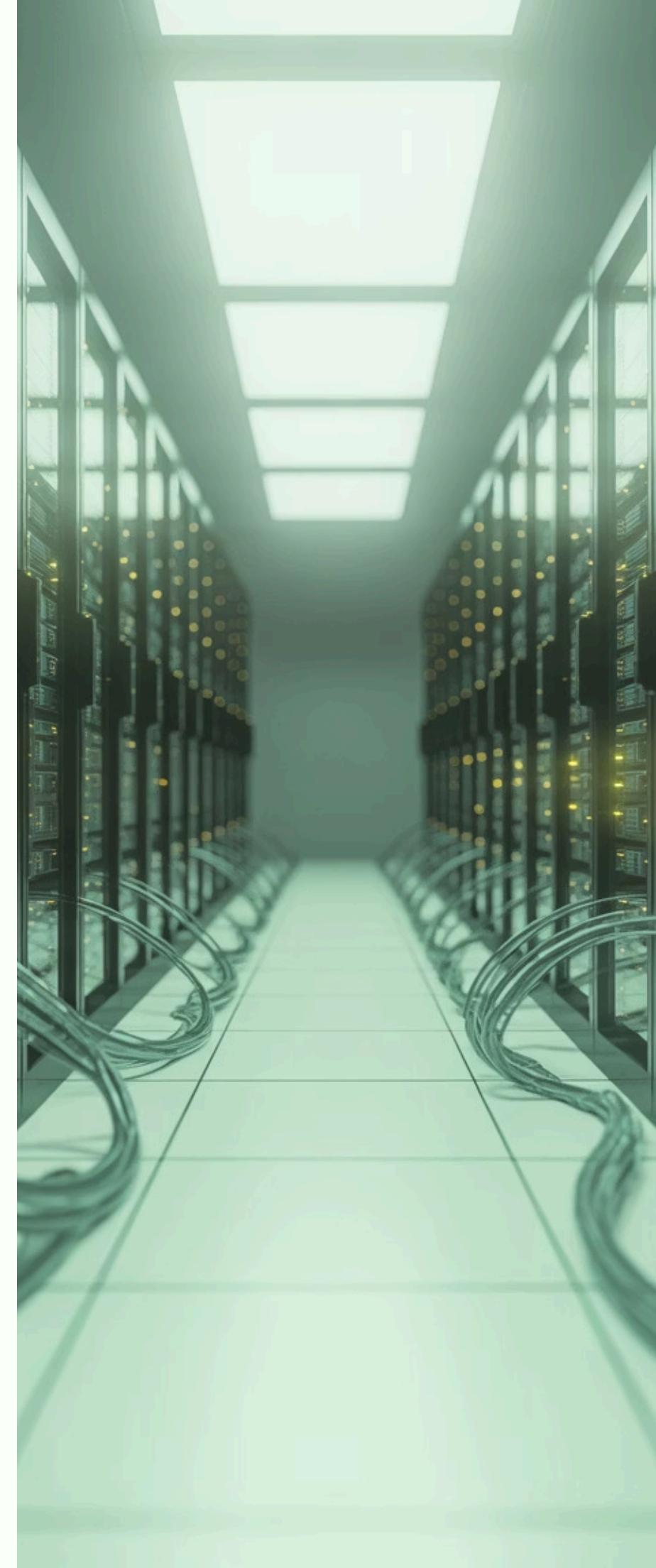
## Activating a Profile

```
spring.profiles.active=dev
```

Profiles help avoid mistakes and make cloud deployment smooth and safe. Each environment gets its own configuration automatically.



**"Right  
configura  
tion at the  
right time  
– this is  
how  
professio  
nals build  
reliable  
systems."**



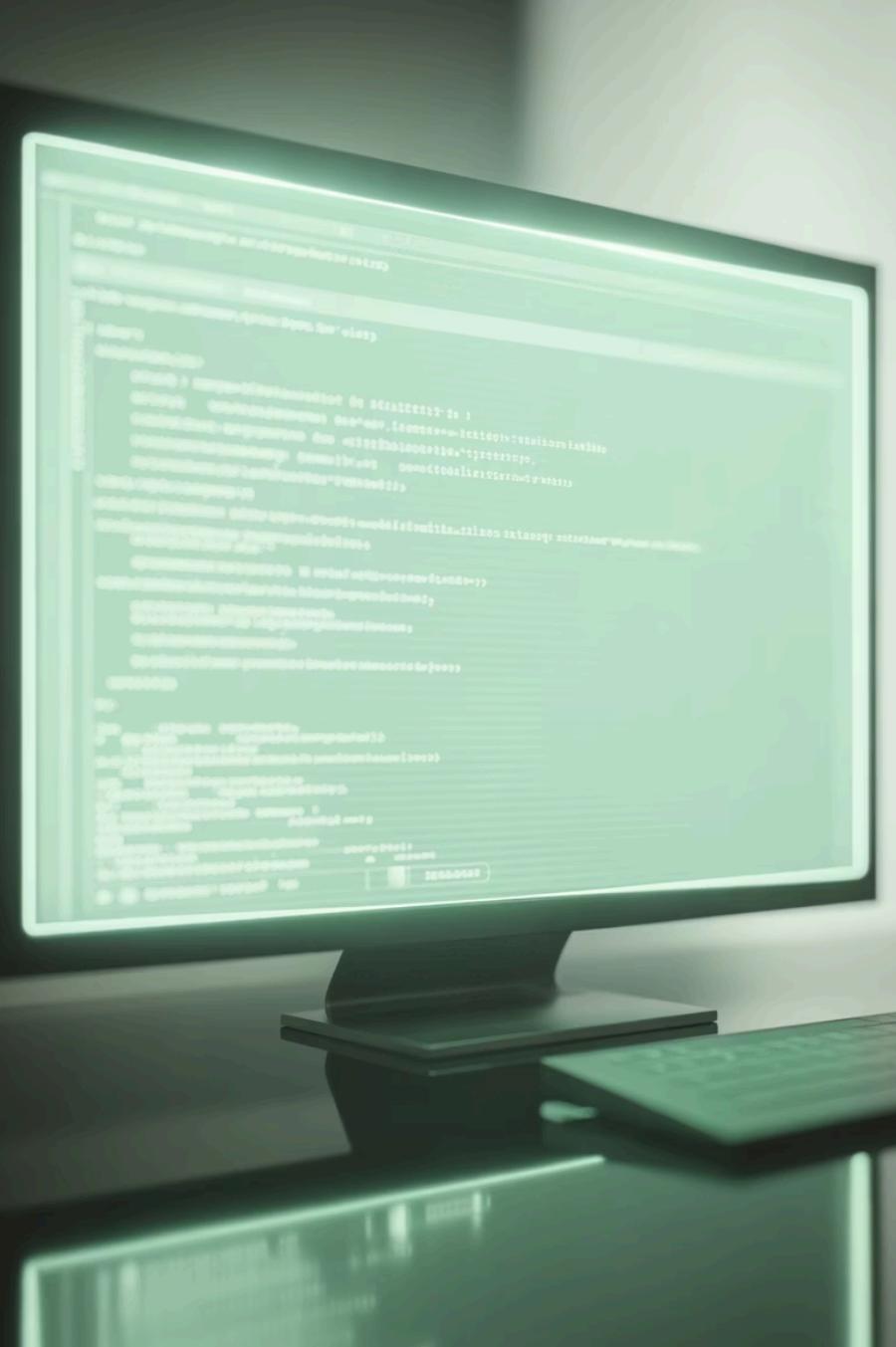
## **Join DURGASOFT for Expert Spring Training**

 *If you want online training in SPRING Framework with SPRING BOOT Cloud, join DURGASOFT - India's trusted training institute.*

 **Contact:** 9246212143, 8885252627

 [www.durgasoftonline.com](http://www.durgasoftonline.com)

 [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)



# **CommandLineRunn er & ApplicationRunner Day 40**

**DURGASOFT**

Page 47

# Running Code at Startup

DURGASOFT

Page 48



## Load Initial Data

Populate database with default records, master data, or seed information when app starts



## Trigger Background Jobs

Start scheduled tasks, queue processors, or asynchronous operations



## Check Configurations

Validate settings, verify connections, or test external services



## Print Startup Logs

Display application info, version details, or configuration summary

**CommandLineRunner** and **ApplicationRunner** are special interfaces in Spring Boot that allow you to run code **immediately after the application starts**. They are perfect for initialization tasks, data loading, and startup validation.

# Implementation Example

DURGASOFT

Page 49

## CommandLineRunner

```
@Component  
public class StartupTask  
    implements CommandLineRunner {  
  
    @Override  
    public void run(String... args) {  
        System.out.println("App started!");  
        System.out.println("Running startup logic...");  
        // Load data  
        // Initialize cache  
        // Check connections  
    }  
}
```

## ApplicationRunner

```
@Component  
public class AppInitializer  
    implements ApplicationRunner {  
  
    @Override  
    public void run(ApplicationArguments args) {  
        System.out.println("App initialized!");  
        // Access named arguments  
        if (args.containsOption("mode")) {  
            String mode = args.getOptionValues("mode").get(0);  
        }  
    }  
}
```

Both runners support argument handling and execute only once at startup. Useful for microservices, batch apps, and cloud initialization tasks.

**ApplicationRunner** provides better argument parsing with named options.

**"Start  
strong –  
run  
important  
tasks the  
moment  
your app  
comes  
alive."**

---

**Thank You for Learning  
with DURGASOFT**

**Continue Your Spring  
Journey**

★ Master SPRING Framework with SPRING BOOT Cloud through expert-led online training at DURGASOFT - India's trusted training institute.

📞 **Contact:** 9246212143, 8885252627

🌐 [www.durgasoftonline.com](http://www.durgasoftonline.com)

✉️ [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com)

