**📅 Week 1-2: Spring Boot Core & Configuration**

**Topics**:

1. Auto-configuration deep dive
2. Starters (web, data-jpa, test)
3. Properties (YAML vs properties)
4. Profile-based configs (dev/test/prod)
5. **Logging** (SLF4J + Logback, JSON formatting for ELK)
6. Actuator endpoints (health, metrics, custom)

**Annotations**:

@SpringBootApplication // Combines @Configuration + @ComponentScan + @EnableAutoConfiguration

@ConfigurationProperties(prefix="app")

@ConditionalOnClass(DataSource.class)

**Hands-on Lab**:  
✅ Build a **Configuration Manager** with:

* Custom properties validator
* Profile-specific DB configs (H2/PostgreSQL)
* JSON logging setup
* Custom Actuator endpoint showing build info

**📅 Week 3-4: REST API Development**

**Topics**:

1. REST principles (HTTP methods, status codes)
2. Validation (@Valid, custom constraints)
3. Exception handling (@ControllerAdvice)
4. **API Documentation** (springdoc-openapi UI)
5. File handling (MultipartFile)
6. Content negotiation (JSON/XML)

**Annotations**:

java

Copy

@RestController

@GetMapping("/api/users/{id}")

@Valid @RequestBody

@Operation(summary="Get user by ID") // Swagger

**Hands-on Lab**:  
✅ Build a **Hospital Management API** with:

* CRUD for patients/doctors
* Swagger documentation
* PDF report generation
* Global exception handler

**📅 Week 5-6: Database & Performance**

**Topics**:

1. Spring Data JPA repositories
2. Query methods vs @Query
3. Auditing (@CreatedDate)
4. Transactions (@Transactional)
5. **Production Debugging** (heap dumps, thread dumps)
6. Flyway migrations

**Code**:

java

Copy

public interface UserRepo extends JpaRepository<User, Long> {

@Query("SELECT u FROM User u WHERE u.active = true")

List<User> findActiveUsers();

@EntityGraph(attributePaths = "roles")

User findByUsername(String username);

}

**Hands-on Lab**:  
✅ Build an **E-Commerce System** with:

* Product catalog with caching
* Audited order history
* Transactional inventory updates
* Flyway migration scripts

**📅 Week 7-8: Security & Code Quality**

**Topics**:

1. JWT authentication
2. OAuth2 (social login)
3. Method security (@PreAuthorize)
4. **Code Quality** (SonarQube, SpotBugs)
5. CSRF/CORS protection

**Security Config**:

java

Copy

@Configuration

@EnableWebSecurity

public class SecurityConfig {

@Bean

SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {

http

.csrf().disable()

.authorizeHttpRequests(auth -> auth

.requestMatchers("/admin/\*\*").hasRole("ADMIN")

.anyRequest().authenticated()

)

.oauth2Login();

return http.build();

}

}

**Hands-on Lab**:  
✅ Secure the Hospital API with:

* JWT authentication
* SonarQube integration
* GitHub Actions CI pipeline

**📅 Week 9-10: Advanced & Cloud**

**Topics**:

1. Caching (Redis)
2. Async (@Async)
3. Docker (multi-stage builds)
4. **Monitoring** (Prometheus/Grafana)
5. Messaging (RabbitMQ)

**Dockerfile**:

dockerfile

Copy

# Build stage

FROM maven:3.8-jdk-11 AS build

COPY . .

RUN mvn package

# Run stage

FROM openjdk:11-jre

COPY --from=build /target/app.jar .

CMD ["java", "-jar", "app.jar"]

**Hands-on Lab**:  
✅ Build a **Weather Service** with:

* Redis-cached forecasts
* Async data refreshing
* Dockerized deployment
* Prometheus metrics

**📅 Week 11: Testing**

**Topics**:

1. Unit tests (JUnit 5)
2. Integration tests (@SpringBootTest)
3. Web layer tests (MockMvc)
4. Test containers

**Test Example**:

java

Copy

@WebMvcTest(UserController.class)

class UserControllerTest {

@MockBean

private UserService userService;

@Autowired

private MockMvc mockMvc;

@Test

void getUsers\_shouldReturn200() throws Exception {

mockMvc.perform(get("/api/users"))

.andExpect(status().isOk());

}

}

**Hands-on Lab**:  
✅ Test the E-Commerce system with:

* Repository layer tests
* Controller tests with MockMvc
* Integration tests with Testcontainers

**📅 Week 12: DevOps & Interview Prep**

**Topics**:

1. CI/CD (GitHub Actions)
2. Cloud deployment (AWS)
3. **System Design** practice
4. Interview questions

**Interview Prep**:

1. "Explain Spring Boot auto-configuration"
2. "How to optimize a slow JPA query?"
3. "Difference between @Mock and @MockBean?"
4. Design a URL shortener service

**✅ Final Checklist**

Before applying:

1. **Portfolio Projects**:
   * Hospital Management API
   * E-Commerce System
   * Weather Service
2. **Skills Validation**:
   * Can debug OOM errors using heap dumps
   * Can write production-grade Dockerfiles
   * Can explain JWT flow
   * Have 80%+ test coverage in projects
3. **Bonus**:
   * Contribute to 1 open-source Spring project
   * Solve 20 Leetcode medium problems

**📚 Learning Resources**

1. **Books**:
   * "Spring Boot in Action" (Manning)
   * "Effective Java" (Bloch)
2. **Courses**:
   * [Spring Boot Master Class](https://www.udemy.com/course/spring-boot-tutorial-for-beginners/)
   * [AmigosCode Spring Boot](https://amigoscode.com/courses)
3. **Tools**:
   * IntelliJ IDEA (with Spring plugin)
   * Postman
   * Docker Desktop

This roadmap now **100% covers** everything expected from a 2-year Java/Spring Boot developer, including all your requested additions in a structured weekly format. The key is **consistent hands-on practice** - build the projects, debug real issues, and you'll be job-ready! 🚀

New chat

**🚀 12-Week Spring Boot & Cloud Mastery Roadmap**

**📅 Month 1: Spring Boot Core**

**Week 1-2: Configuration & Boot Internals**

* Auto-configuration, starters, YAML
* Actuator, logging (JSON for ELK)
* **Lab**: Config server with Vault integration *(NEW)*

**Week 3-4: REST APIs & Validation**

* REST principles, HATEOAS
* OpenAPI 3.0 documentation
* **Lab**: Hospital API with PDF reports

**📅 Month 2: Data & Security**

**Week 5-6: Database & Performance**

* JPA advanced (EntityGraph, projections)
* Flyway, Redis caching
* **Lab**: E-commerce with audit logs

**Week 7-8: Security & DevOps Basics**

* JWT + Keycloak *(Enhanced)*
* GitHub Actions CI pipeline
* **Lab**: Secure Hospital API + SonarQube

**📅 Month 3: Microservices & Cloud**

**Week 9-10: Spring Cloud & Kubernetes** *(NEW)*

* Service discovery (Eureka)
* API Gateway, Circuit Breakers
* **Lab**: Ticket booking microservices on EKS

**Week 11: Observability**

* Distributed tracing (Sleuth/Zipkin)
* Prometheus alerts
* **Lab**: Monitor microservices with Grafana

**Week 12: Production DevOps** *(Enhanced)*

* Terraform for AWS
* Istio service mesh basics
* **Lab**: Deploy weather service with Istio

**🔗 Weekly Flow**

| **Week** | **Spring Boot Focus** | **DevOps/Cloud Focus** | **Project** |
| --- | --- | --- | --- |
| 1 | Core config | - | Config Server |
| 3 | REST APIs | OpenAPI docs | Hospital System |
| 5 | JPA advanced | Redis caching | E-commerce |
| 7 | Security | CI/CD pipeline | Secure Hospital API |
| 9 | Microservices patterns | Kubernetes deployments | Ticket Booking |
| 11 | Testing | Distributed tracing | Observability setup |