

## Lab Setup Overview

This setup includes:

- Java Development Kit (JDK)
  - Spring Boot CLI & IDE
  - Maven Build Tool
  - Git & GitHub
  - Docker Desktop for Windows
  - Minikube (Local Kubernetes Cluster)
  - Tools: Postman, VisualVM, Zipkin, Prometheus, Grafana
  - Spring Cloud Components (e.g., Eureka, Config Server, Gateway)
- 

## Prerequisites

Ensure your system meets the following requirements:

Requirement	Specification
OS	Windows 11 (64-bit)
RAM	Minimum 8 GB (16 GB recommended)
Disk Space	At least 20 GB free space
Processor	Dual Core or better
Internet Connection	Required for downloads

---

## Step-by-Step Setup Instructions

### 1. Install Java Development Kit (JDK)

Why?

Spring Boot requires JDK to run and compile applications.

Steps:

1. Download **OpenJDK 17** from <https://adoptium.net>
2. Choose the **Windows x64 Installer**
3. Run the installer and accept defaults.
4. Set `JAVA_HOME`:

- Open **Environment Variables**
- Under **System variables**, add:  
JAVA\_HOME = C:\Program Files\Java\jdk-17.x.x
- Update Path to include:  
%JAVA\_HOME%\bin

Verify:

```
java -version  
javac -version
```

---

## 2. Install Apache Maven

Why?

Maven is used to build and manage dependencies for Spring Boot projects.

Steps:

1. Download latest Maven from <https://maven.apache.org/download.cgi>
2. Extract it to C:\tools\apache-maven-<version>
3. Set environment variables:

```
MAVEN_HOME = C:\tools\apache-maven-<version>  
Path += %MAVEN_HOME%\bin
```

Verify:

```
mvn -v
```

---

## 3. Install an IDE

Recommended Options:

- IntelliJ IDEA (Ultimate Edition preferred)
- Eclipse with Spring Tools Suite plugin
- VS Code with Java extensions

*IntelliJ IDEA Installation:*

1. Download from <https://www.jetbrains.com/idea/download/>
2. Install and launch

3. Install **Spring Boot Plugin** if not already included

---

## 4. Install Git

Why?

For version control and cloning sample projects.

Steps:

1. Download Git from <https://git-scm.com/download/win>
2. During installation:
  - o Choose “Use Git from Windows Command Prompt”
  - o Use default settings otherwise

Verify:

```
git --version
```

---

## 5. Install Docker Desktop for Windows

Why?

To containerize and deploy microservices locally.

Steps:

1. Download Docker Desktop from <https://www.docker.com/products/docker-desktop>
2. Install and enable **WSL2** if prompted during install
3. Launch Docker Desktop and sign in (optional)

Verify:

```
docker --version  
docker info
```

---

## 6. Install Minikube (Local Kubernetes Cluster)

Why?

To simulate Kubernetes deployment for microservices.

## Steps:

1. Download `minikube-windows-amd64.exe` from  
<https://minikube.sigs.k8s.io/docs/start/>
2. Rename file to `minikube.exe` and move to `C:\tools`
3. Add to Path:

```
Path += C:\tools
```

## Start Minikube:

```
minikube start --driver=docker
```

## Verify:

```
kubectl get nodes
```

---

## 7. Install kubectl (Kubernetes CLI)

### Steps:

1. Download `kubectl.exe` from <https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/>
2. Place in `C:\tools` and add to Path

## Verify:

```
kubectl version --client
```

---

## 8. Install Postman (for API Testing)

Download from <https://learning.postman.com/download/>  
Install and create a free account.

---

## 9. Install Monitoring Tools

### Optional but Recommended:

- **Prometheus:** For metrics
- **Grafana:** For dashboards
- **Zipkin:** For distributed tracing
- **ELK Stack:** For centralized logging

These can be installed via Docker Compose later as needed.

Example:

```
docker run -d -p 9411:9411 openzipkin/zipkin
```

Access Zipkin at: <http://localhost:9411>

---

## 10. Clone Course Project Repository

Example:

Assuming you have a GitHub repo with starter code for labs:

```
git clone https://github.com/your-org/microservices-course.git  
cd microservices-course
```

Each chapter has its own folder with starter code.

---

## 11. Optional: Install Spring Boot CLI

Why?

Quickly prototype Spring Boot apps from command line.

Steps:

1. Download from <https://spring.io/projects/spring-boot#cli>
2. Unzip and add `spring.bat` to your path

Verify:

```
spring --version
```

---

## 12. Sample App Test

Run a simple Spring Boot app to test the setup:

1. Navigate to a sample project:

```
cd microservices-course/chapter1-sample-app
```

2. Build and run:

```
mvn spring-boot:run
```

3. Access in browser: <http://localhost:8080>

---

## Tools Summary Table

Tool	Purpose	Installed?
JDK 17	Java runtime and compilation	✓
Maven	Dependency management/build tool	✓
Git	Version control	✓
Docker Desktop	Containerization	✓
Minikube	Local Kubernetes cluster	✓
kubectl	Kubernetes CLI	✓
Postman	REST API testing	✓
Spring Boot CLI	Quick app prototyping	✓ (Optional)
Zipkin/Prometheus	Tracing and monitoring	✓ (Optional)

---