

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
<div>Spring Boot<ul style="list-style-type: none"><li><input checked="" type="radio"/> Introduction to Spring Boot</li><li><input checked="" type="radio"/> Multi-Tier Architecture</li><li><input type="radio"/> Beans</li><li><input type="radio"/> Bean Scopes</li><li><input type="radio"/> Bean Validation</li><li><input type="radio"/> Dependency Injection</li><li><input type="radio"/> Components</li><li><input type="radio"/> Configuration</li><li><input type="radio"/> Connecting to a Database</li><li><input type="radio"/> Entities</li><li><input type="radio"/> Postman</li><li><input type="radio"/> Controllers</li><li><input type="radio"/> Services</li><li><input type="radio"/> Repositories</li><li><input type="radio"/> Custom Queries</li><li><input type="radio"/> Data Transfer Objects</li><li><input type="radio"/> Lombok</li><li><input type="radio"/> Custom Exceptions</li><li><input type="radio"/> Swagger</li><li><input type="radio"/> Profiles</li><li><input type="radio"/> Pre-Populating Databases for Testing</li><li><input type="radio"/> Unit testing with Mockito</li></ul></div>

# Multi-Tier Architecture

## Contents

- [Overview](#)
- [Benefits](#)
- [Three-tier Architecture](#)
- [Tutorial](#)
- [Exercises](#)

## Overview

Multi-tier architecture (also called *N*-tier architecture) describes the process of splitting up the major components of an application onto different environments. For example, a web-app might be spread onto three different environments, like so:

- Database running on a dedicated Cloud DB such as AWS RDS.
- Back-end running on cloud-based VM's such as AWS EC2 instances or on a serverless architecture such as AWS Lambda.
- Front-end being deployed to a static content store such as AWS S3.

## Benefits

Like a lot of practices in software development, multi-tier architecture is designed to reduce *coupling*. By splitting the full app onto different environments it is *far* easier to swap out one section when required or to rapidly scale up one particular section, something *far* harder and more costly to do with a traditional single-tier application where every single component could be running on the same hardware.

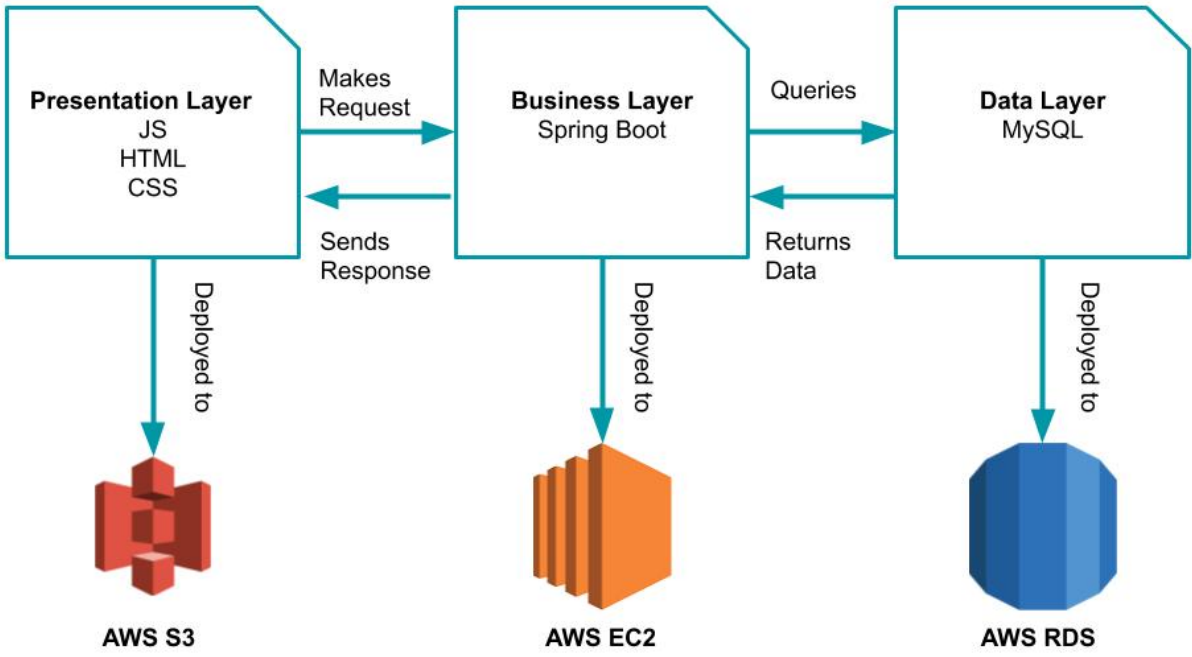
## Three-tier Architecture

The most common example of multi-tier architecture is the **three-tier architecture**, which consists of:

- Presentation Layer** - The 'front-end', the part of the application seen by the user.
- Business Layer** - The 'back-end', this part of the application deals with applying *business rules* to any data received from or sent to the user.
- Persistence/Data Layer** - The 'database', persists data in order for the app to function.

The below image shows an example three-tier architecture using a Spring-based web application:

<div><div></div><div>Testing</div></div>
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet



## Tutorial

There is no tutorial for this module.

## Exercises

There are no exercises for this module.