COURSEWARE

Multi-Tier Architecture

Contents

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:

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

Benefits

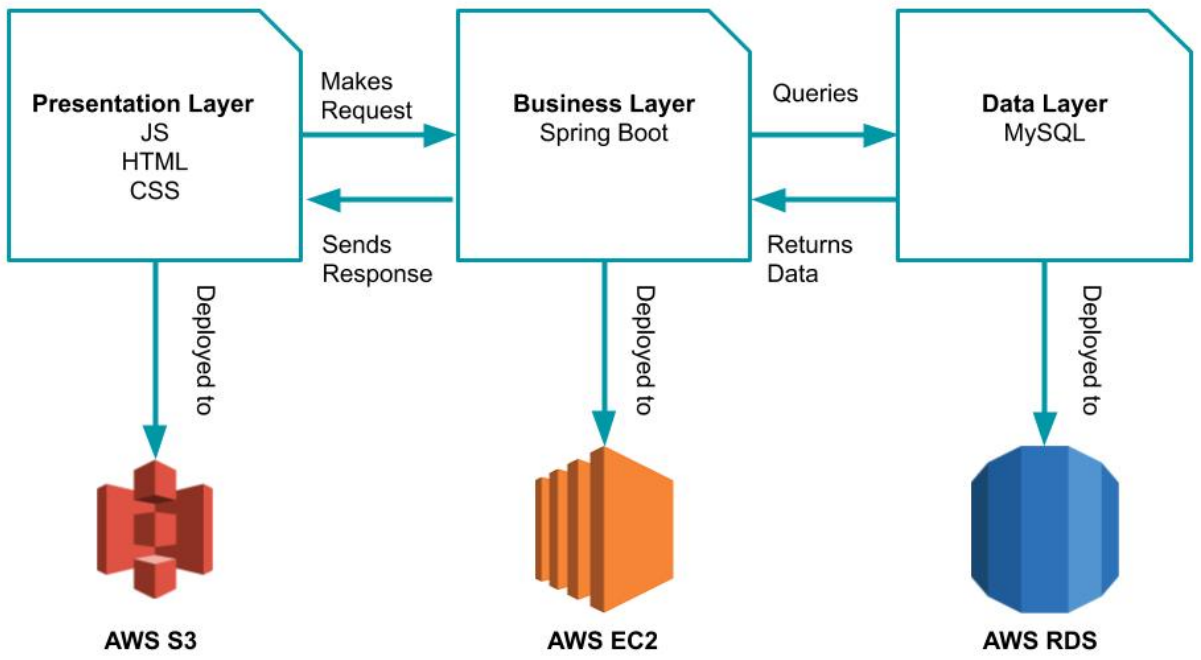
Like a lot of practices in software development, multi-tier architecture is designed to reduce *coupling*.
By splitting the full app onto different environments it is *far* easier to swap out one section when required or to rapidly scale up one particular section, something *far* harder and more costly 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:

1. **Presentation Layer** - The 'front-end', the part of the application seen by the user.
2. **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.
3. **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:



Tutorial

There is no tutorial for this module.

Exercises

There are no exercises for this module.