Tuesday Week 7

Testing Concepts, SonarCloud analysis, Code Coverage

Objectives

- After today, you should be able to
  - Explain concepts conceptually with regards to testing
  - Understand the idea of performing static code analysis with SonarCloud
  - Generate code coverage reports

Testing Concepts

- Markdown notes
- Important to understand the high level concepts
  - Then dive into the specific details
  - Then understand testing from the perspective of the demos we have already covered
- For your P2, include unit AND integration tests into your backend Maven project
  - Unit and integration tests contribute to code coverage
  - Create another project for the E2E and utilize Selenium
    - I would highly recommend using Cucumber and Selenium together for the E2E tests (also try using TestNG)

SonarCloud

- Static code analysis “in the cloud”
  - http://sonarcloud.io
  - Produce information such as “code smells” and any security vulnerabilities that our code could potentially have
  - “Code Smell”: code that could be written more cleanly
- We will also be able to display code coverage reports
  - This requires a Maven plugin known as Jacoco (separate from SonarCloud)
  - This plugin will scan through the code that is executed during the running of our tests and provide a report file
  - We can configure SonarCloud to load this report file and display it as a code coverage metric

Wednesday Week 7

Standalone Tomcat server, AWS cloud computing notes, deployment onto EC2, and creation of ship manager Angular app w/ login and viewing ships

Objectives

- After today, you should be able to
  - Understand the ecosystem surrounding a standalone Tomcat server
    - Compare and contrast this with the IDE managed Tomcat server
    - Deploy applications to the Tomcat server on our local machines
  - Understand the conceptual ideas behind cloud computing
  - Setup Tomcat server on EC2 and deploy app on EC2
  - Have an Angular app connected with the backend with a full understanding of how to handle login and CORS related settings
    - Deploy the Angular app on EC2 as well

IDE Managed Tomcat v. Standalone Tomcat

- IDE Managed Tomcat:
  - Enables developers to quickly make changes and see those changes in action
  - Provides features such as hot reloading, so that when you do make a change, it will automatically restart the application
- Standalone Tomcat:
  - Used to actually host an application deployed to production
  - We would use Maven to package a .war file (web archive file) and then deploy this to the standalone Tomcat server
- Development servers v. Deployment servers
  - IDE managed tomcat (developer) v. Deployment to production (Standalone Tomcat)
  - Angular: ng serve / npm run start (http://localhost:4200 development) v. ng build -> index.html -> various .js files -> placing these files onto a dedicated web server

Thursday Week 7

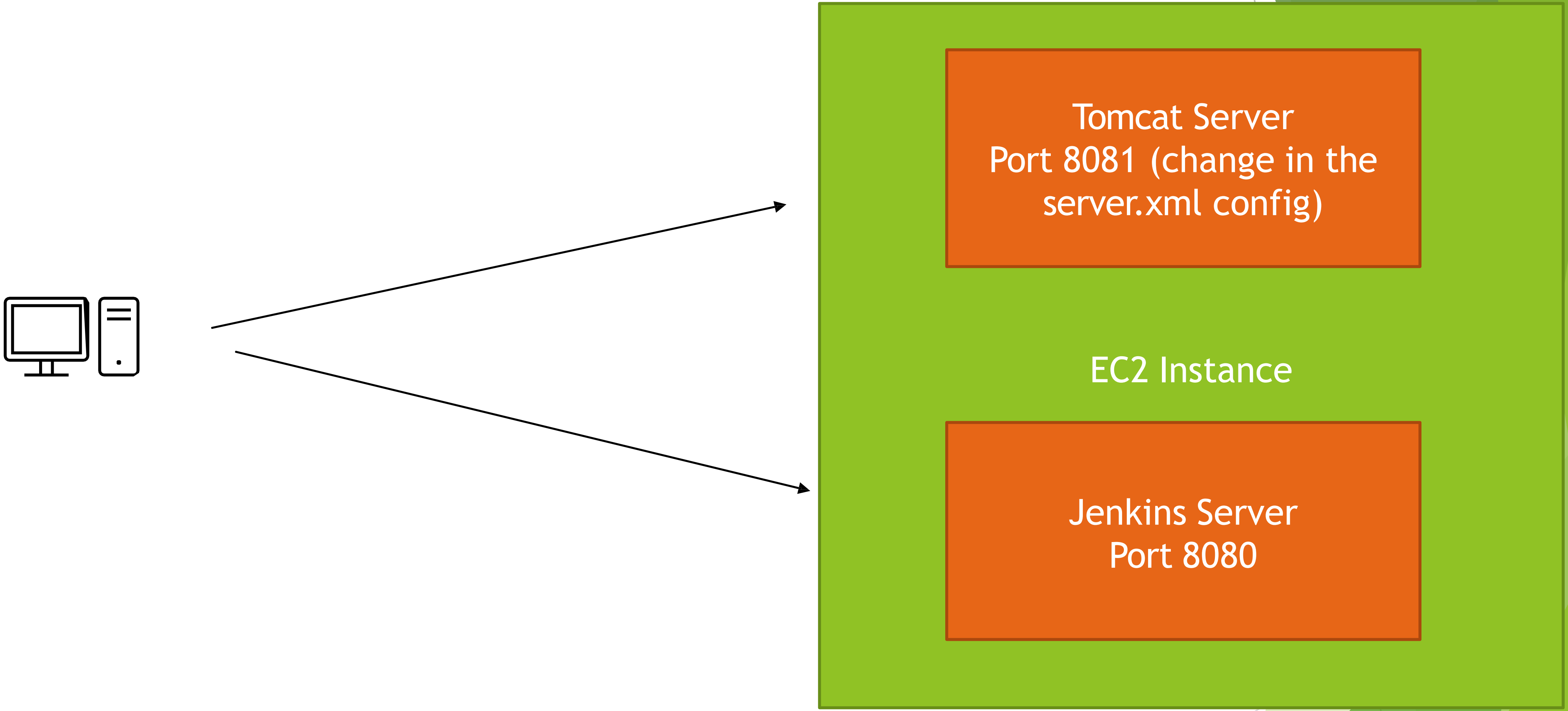
DevOps (with Jenkins)

Interviewing Information (Typical Java w/ Automation information)

- So far, looks like the client will be Infosys (for interviews possibly starting next week)
- Based on past data (20-25 associates):
  - Java (85% of the interviews)
  - Test Automation (70% of the interviews)
    - Selenium
    - Cucumber (BDD)
  - JUnit/Mockito (20% of the interviews)
  - SQL (40% of the interviews)
  - Agile (35% of the interviews)
  - Talking about your projects (25% of the interviews)
  - REST (25% of the interviews)
  - Angular (20% of the interviews)
  - JDBC (20% of the interviews)
  - DevOps (20% of the interviews)
  - SDLC (20% of the interviews)
- Coding Challenge: 20% of the interviews
- The past doesn’t indicate the present/future

Interviewing Information (Typical Java MSA Information)

- Java (95% of the interviews)
- Project (70% of the interviews)
- Spring (70% of the interviews)
- Angular (50% of the interviews)
- SQL (50% of the interviews)
- Hibernate (40% of the interviews)
- REST (40% of the interviews)
- Microservices (35% of the interviews)
- JDBC (30% of the interviews)
- JavaScript (30% of the interviews)
- Servlets (20% of the interviews)



Tomcat Server
Port 8081 (change in the server.xml config)

EC2 Instance

Jenkins Server
Port 8080

Friday Week 7

S3 bucket deployment, Docker