

Final Project

EPAM DevOps Online Winter 2022

IHOR LEONTIEV

JULY 2022

Self-Presentation



MY BACKGROUND

- International trading agent (grains, fertilizers)
- 15+ years of experience in Swiss and Singapore companies
- Full employee lifecycle in the commodities trading domain (from logistics manager to the country manager)
- 3 Higher educations (Master's Degrees in Engineering, Economics, Law)
- Open-minded and passionate to self-developing and mastering new technologies
- IT-switcher since 2021 passing Coursera trainings and studying in EPAM DevOps external online program



Project Tasks

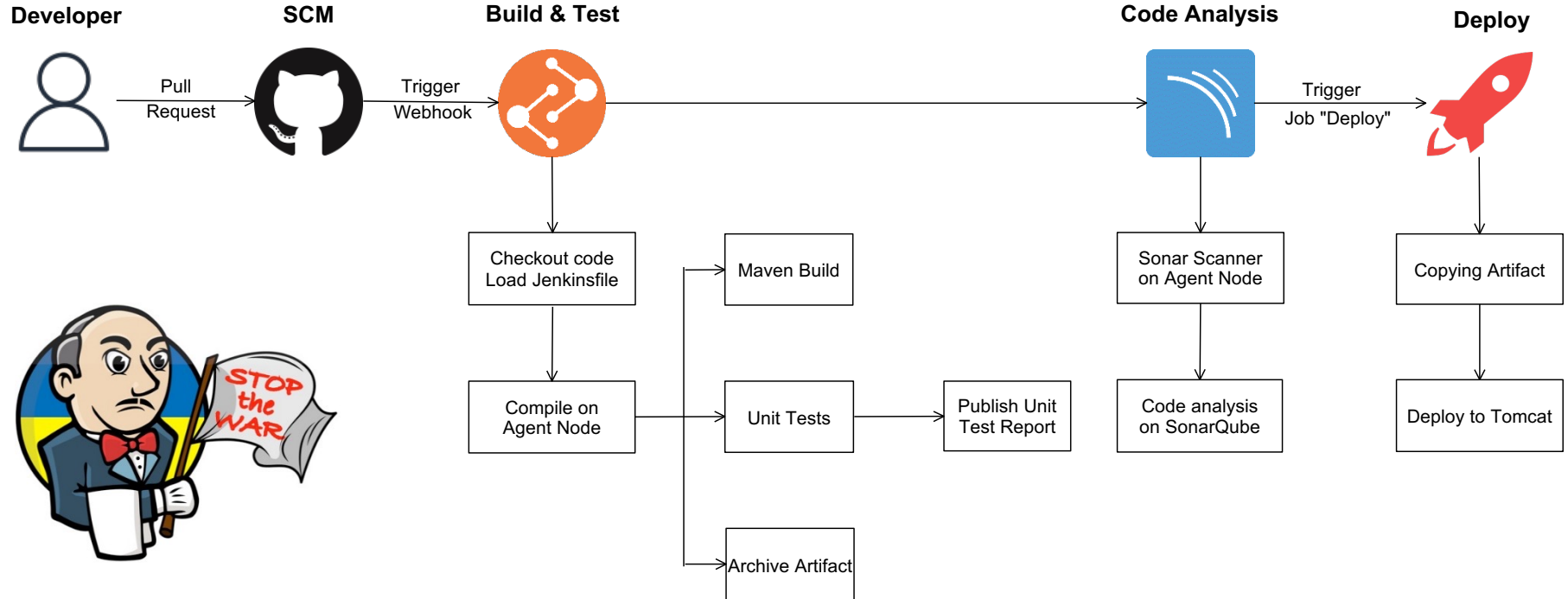
-
- Create CI/CD pipeline for Java App
 - Integrate code analysis
 - Implement deployment to Tomcat
 - Verify deployment of changes

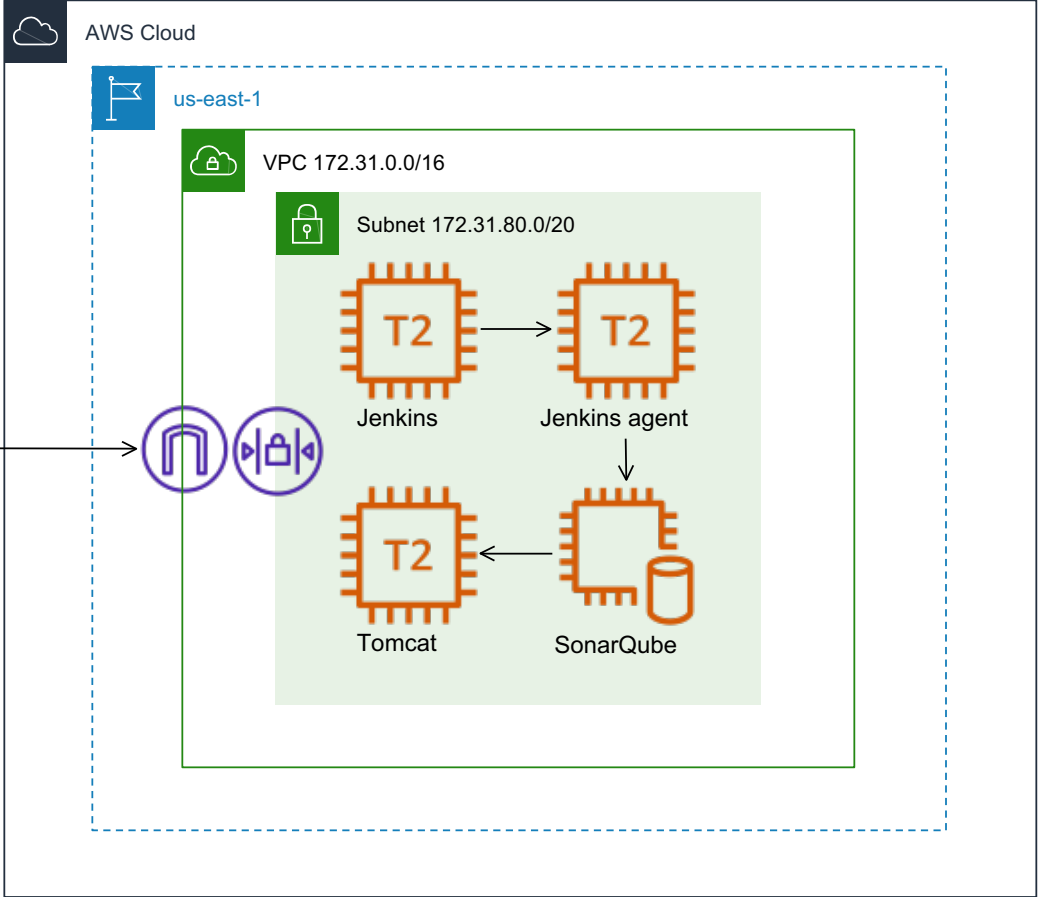


Project Results

Jenkins CI/CD pipeline for Spring PetClinic App with code analysis

Tech Stack and Project Implementation Steps






Pipeline Flow

 **Jenkins**

Q Search 

 1  1  Ihor Leontiev

Dashboard > Petclinic >

 Back to Dashboard

 Status

 Changes

 Build Now

 Configure

 Delete Pipeline


 Full Stage View

 SonarQube

 Rename

 Pipeline Syntax

 GitHub Hook Log

 Build History trend

 Atom feed for all  Atom feed for failures

Pipeline Petclinic

 Last Successful Artifacts

 petclinic.war 39.98 MB  view

 Recent Changes



Stage View

	Preparation	Build	SonarQube analysis	Archive Artifact	Deploy
	2s	31s	2min 31s	5s	22s
#67 Jul 11 22:48 1 commit	2s	31s	2min 35s	5s	29s
#66 Jul 11 22:36 1 commit	3s	32s	2min 28s	5s	14s

SonarQube Quality Gate

petclinic **Passed**

server-side processing: **Success**

SonarQube Quality Gate Status

The screenshot displays the SonarQube interface for the 'petclinic' project. The main section, 'QUALITY GATE STATUS', shows a large green 'Passed' box with the text 'All conditions passed.' Below this, the 'MEASURES' section provides a detailed breakdown of the project's quality metrics. A sidebar on the left offers navigation options and filters, while a bottom summary bar provides a quick overview of key metrics.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

petclinic ☆ master + Last analysis had 1 warning July 11, 2022 at 10:48 PM Version 4.2.5-SNAPSHOT

Overview Issues Security Hotspots Measures Code Activity Project Information

QUALITY GATE STATUS

Passed
All conditions passed.

MEASURES

New Code
Since June 30, 2022
Started 11 days ago

Overall Code

174 🐛 Bugs Reliability **E**

9 🔒 Vulnerabilities Security **D**

6 🔒 Security Hotspots 0.0% Reviewed Security Review **E**

73d Debt 1.1k 🪞 Code Smells Maintainability **A**

0.0% Coverage on 21k Lines to cover Unit Tests 59

65.4% Duplications on 55k Lines 2.3k Duplicated Blocks

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

My Favorites All Search by project name 1 projects

Filters

Quality Gate

Passed 1
Failed 0

Reliability (🐛 Bugs) **A** 0

petclinic **Passed**

🐛 Bugs **E** 🔒 Vulnerabilities **D** 🔒 Hotspots Reviewed **E** 🪞 Code Smells **A**

Coverage 0.0% Duplications 65.4% Lines 55k JavaScript...

Console Output Excerpt 1/3



Console Output

```
Started by GitHub push by Ihor-2022
Obtained jenkinsfile.groovy from git https://github.com/Ihor-2022/petclinic
[Pipeline] Start of Pipeline
[Pipeline] node
Running on node_1 in /home/ubuntu/workspace/Petclinic
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Preparation)
[Pipeline] git
The recommended git tool is: git
using credential clfaldd8-c316-446e-aede-cc85f5d67a7b
Fetching changes from the remote Git repository
> git rev-parse --resolve-git-dir /home/ubuntu/workspace/Petclinic/.git # timeout=10
> git config remote.origin.url https://github.com/Ihor-2022/petclinic # timeout=10
Fetching upstream changes from https://github.com/Ihor-2022/petclinic
> git --version # timeout=10
> git --version # 'git version 2.25.1'
using GIT_ASKPASS to set credentials github token
> git fetch --tags --force --progress -- https://github.com/Ihor-2022/petclinic
+refs/heads/*:refs/remotes/origin/* # timeout=10
Checking out Revision 2bfb84bcb535117b578b4d90e1dcbf2976e45c51 (refs/remotes/origin/master)
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
```

Console Output Excerpt 2/3

```
INFO LocalContainerEntityManagerFactoryBean - Closing JPA EntityManagerFactory for persistence unit 'petclinic'

Results :

Tests run: 59, Failures: 0, Errors: 0, Skipped: 0

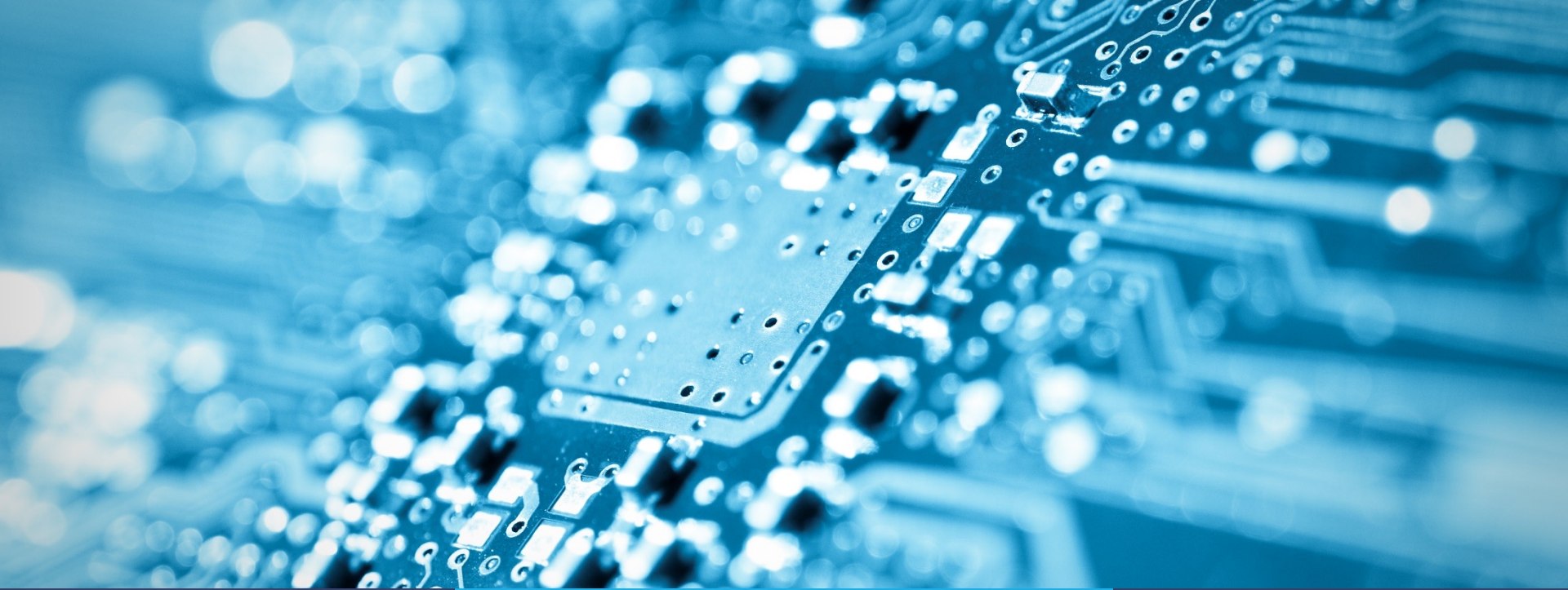
[INFO]
[INFO] --- maven-war-plugin:2.3:war (default-war) @ spring-petclinic ---
[INFO] Packaging webapp
[INFO] Assembling webapp [spring-petclinic] in [/home/ubuntu/workspace/Petclinic/target/spring-petclin.
4.2.5-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/ubuntu/workspace/Petclinic/src/main/webapp]
[INFO] Webapp assembled in [1016 msecs]
[INFO] Building war: /home/ubuntu/workspace/Petclinic/target/petclinic.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 27.219 s
[INFO] Finished at: 2022-07-11T20:48:43Z
[INFO] -----
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube analysis)
[Pipeline] withSonarQubeEnv
Injecting SonarQube environment variables using the configuration: my-sonarqube
```

Console Output Excerpt 3/3

```
[INFO] Analysis report uploaded in 207ms
[INFO] ANALYSIS SUCCESSFUL, you can browse http://172.31.87.217:9000/dashboard?id=org.springframework.samples%3Aspring-petclinic
[INFO] Note that you will be able to access the updated dashboard once the server has processed the sul
analysis report
[INFO] More about the report processing at http://172.31.87.217:9000/api/ce/task?id=AYHvB5TLJJC4G\_TIwS
[INFO] Analysis total time: 2:23.079 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 02:28 min
[INFO] Finished at: 2022-07-11T20:51:14Z
[INFO] -----
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Archive Artifact)
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] build (Building Petclinic deploy)
Scheduling project: Petclinic deploy
Starting building: Petclinic deploy #42
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Conclusions

- The implemented pipeline builds Spring PetClinic App on Jenkins node
- Pipeline passes successfully in 03:34 minutes
- Code analysis allows to determine bugs and vulnerabilities
- Load testing can be added to ensure application stability and performance



Thank you!