

Oleg Mironov

DevOps Pipeline with Docker

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

15 April 2018

Author(s) Title	Oleg Mironov DevOps Pipeline with Docker
Number of Pages Date	55 pages + 11 appendices 2 December 2017
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Marko Klemetti, CTO Erik Pätynen, Senior Lecturer
<p>Software complexity and size are growing at the ever-increasing rate. This study attempts to use modern DevOps practices and technologies to create a reusable pipeline template capable of meeting the demands of modern software and strengthen up the development practices. Such pipeline should largely be reusable and flexible as well as being able to be scaled and maintained with high reliability.</p> <p>Theoretical research on the topic of how DevOps came to be and what it means is an important part of this project. Then a set of technologies that reflect the state of DevOps today was carefully studied and analysed. Docker and its fast-growing ecosystem is the core technology of this project. With Docker and some other technologies, a set of configuration files could be used to run, develop, test and deploy an application. Such approach allows maximum automation while ensuring transparency of those automation processes.</p> <p>The result of this project is a fully working pipeline setup that is fully automated and is able to support a fast-paced software development. The pipeline is built against a reference project. Most of the pipeline is configured with a set of different configuration files meaning that from a fresh start it could be brought up with minimal human interaction. It covers all parts of a reference application lifespan from a development environment to a potential production deployment. There is a set of technologies used in the pipeline such as Jenkins, Docker and container orchestration with Rancher.</p>	
Keywords	Docker, Jenkins, Rancher, DevOps, Pipeline, Automation, Continuous Integration, Continuous Delivery

Contents

1	Introduction	1
2	Theoretical Background	3
2.1	Continuous Integration	3
2.2	Continuous Delivery and Continuous Deployment	5
2.3	DevOps	7
2.4	Virtualization	10
2.5	Docker	12
2.6	Container Orchestration	15
2.7	Container Management Software	17
3	Project Methods and Materials	19
3.1	Project Workflow and Methods	19
3.2	Requirements	19
3.3	Tools and Technologies	20
4	Pipeline Design and Implementation	22
4.1	Reference Project Description	22
4.2	Pipeline Flow	23
4.3	Implementation Plan	26
4.4	Dockerizing Voting App	27
4.5	Local Container Orchestration	30
4.6	Virtual Machines with Ansible and Vagrant	34
4.7	Configuring Rancher Environment	36
5	Results and Discussion	49
5.1	Summary of Results	49
5.2	Evaluation of Results	50
5.3	Project Challenges	52
6	Conclusions	55
	References	56
	Appendix 1. GitHub Web Links	60

Appendix 2. Voting App Interface Screenshots	61
Appendix 3. Rancher docker-compose.dev.yml.	63
Appendix 4. Rancher rancher-compose.dev.yml	65
Appendix 5. Rancher load balancer UI.	67
Appendix 6. Add Backend Webhook.	68
Appendix 7. Unlock Jenkins.	69
Appendix 8. Select Jenkins Plugins.	70
Appendix 9. Selecting Jenkins job type.	71
Appendix 10. Adding DockerHub credentials to Jenkins.	72
Appendix 11. Final Jenkinsfile.	73

Abbreviations

CD	Continuous Delivery
CDt	Continuous Deployment
CI	Continuous Integration
DevOps	Development Operations
OS	Operating System
QA	Quality Assurance
VM	Virtual Machine
UI	User Interface

[Click here to download full PDF material](#)