FINAL 3.27.2018 (JP)

COURSE NAME	Advanced Application Development with Red Hat OpenShift
DEVELOPER	Wolfgang Kulhanek
DURATION	32 hours
MODALITY	ILT (instructor-led training)
PROFICIENCY LEVEL	Advanced
URL	https://learning.redhat.com/course/view.php?id=823
PRODUCT & VERSION	• Red Hat OpenShift Container Platform 3.7
SPECIALIZATION	• Middleware Solutions
TRACK	• Platform-as-a-Service (PaaS) Development
ROLE	• Delivery
OUTCOME	• 12 CE credits • Red Hat Delivery Specialist - Advanced Platform-as-a-Service (PaaS) Development
COURSE DESCRIPTION	*Advanced Application Development with Red Hat OpenShift* teaches students advanced application development, configuration, and management skills using Red Hat OpenShift Container Platform. It is NOT a programming course; it focuses on making all kinds of applications run on Red Hat OpenShift Container Platform. The course also discusses CI/CD with Red Hat OpenShift Container Platform. After completing this course, students will be able to: • Discuss microservices architecture using the 12-Factor Application • Create applications using the Kubernetes Core Workloads APIs • Set up a full CI/CD environment using source code repositories, artifact managers, code analysis tools, and Jenkins as the CI/CD orchestrator • Build applications for OpenShift

	• Configure a complex CI/CD pipeline in Jenkins
SKILLS PREREQUISITES	• Experience with Red Hat OpenShift Container Platform • Ability to read source code • Ability to create scripting source code (e.g., Groovy for the Jenkins pipeline labs) • Knowledge of software configuration management, Git in particular • JEE development experience with Maven builds is helpful but not required
TRAINING PREREQUISITES	• *Red Hat OpenShift Foundations (required)* • *Application Development with OpenShift (required)* • *Red Hat OpenShift Container Platform Implementation* (optional, highly recommended) • *Advanced Deployment with Red Hat OpenShift* (optional, highly recommended)
COURSE OUTLINE	**Introduction to Course and Learning Environment** • Learn about the *OpenStack Implementation with Red Hat OpenStack Platform 10 director* course. • Understand the prerequisites, training environment, and system designations used during the lab procedures. • Learn tips for successfully completing the labs. • Understand course resources. **Twelve-Factor Applications** • Learn about the twelve-factor methodology: 1. Codebase 2. Dependencies 3. Configuration 4. Backing services 5. Build, release, run 6. Processes 7. Port binding 8. Concurrency

	9. Disposability 10. Development/production parity 11. Logs 12. Administrative processes ● Learn about health checks. **Controllers** ● Learn about the Kubernetes Core Workload API and: ○ Deployments ○ Replica sets ○ Stateful sets ○ Daemon sets ● Learn about: ○ Blue-Green deployments ○ Jobs and cronjobs ○ Health checks ○ Sidecar containers ● Deploy a stateful set. ● Execute a Blue-Green deployment. ● Understand and set up OpenShift health checks (readiness and liveness probes). ● Set up OpenShift Health checks. ● Set up jobs and cronjobs. ● Review the use cases for sidecar containers (logging/OAuth-proxy) and set up a sidecar container (logging or OAuth-proxy). **CI/CD Tools** ● Learn about CI/CD with OpenShift: ○ Gogs ○ Jenkins ○ Customizing Jenkins ○ Jenkins slave pods ○ Customizing Jenkins slave pods ○ Nexus

	- SonarQube - skopeo - Understand the concepts and benefits of DevOps, continuous Integration, and continuous deployments. - Review open source tools used for continuous integration and continuous delivery. - Understand the OpenShift Jenkins Image. - Customize a Jenkins slave pod to include skopeo. - Understand escribe the need for a private Git repository (Gogs) and set up a private Git repository. - Set up an artifact manager and proxy (Nexus). - Set up a code coverage and analysis tool (Sonarqube). - Use skopeo to move an image from one registry to another **Building Applications** - Learn about: - Build definition - Build configuration - Build input - Build secrets - Build strategy - Advanced build operations - Set requests and limits for a build configuration. - Set up a chained build. - Use build source secrets to access private source code repositories and private artifact repositories. - Use external artifacts in a build. - Set up an incremental build. - Execute a binary build. **Pipelines** - Learn about: - Jenkins pipelines - Image tagging and promotion

	○ Pipeline build configuration ● Create a pipeline-based Jenkins job. ● Create a complex pipeline using a private repository, artifact manager, and code analysis. ● Create a complex pipeline using binary builds and blue-green deployments. **Assignment** ● Complete and submit final assessment.