

Advanced Deployment with OpenShift – Homework

This is a document to provide the necessary information about running the script which is used to deploy the OpenShift cluster and complete the CICD pipeline which is needed to successfully complete the homework assignment.

Git repository with the homework scripts:

https://github.com/jelenas316/ocp_advanced_deployment_homework

Step	Description	Command
1.	Go to root	<code>sudo -i</code>
2.	Clone the git repository	<code>git clone https://github.com/jelenas316/ocp_advanced_deployment_homework.git</code>
3.	Run the ansible playbook	<code>ansible-playbook ./ ocp_advanced_deployment_homework /homework.yaml</code>
4.	Uninstall the cluster (optional)	<code>sh ./ ocp_advanced_deployment_homework /scripts/uninstall.sh</code>

The homework.yaml script automatically deploys the OpenShift cluster, creates PVs with different sizes (5G and 10G) and creates the different users requested in the assignment. The script also deploys the NodeJS-Mongo-Persistent app as a smoke test to see the ability to deploy a simple app. The CICD pipeline is created in the task-dev project and it is promoted to the task-prod project automatically through the pipeline. In the end the scripts provide two groups with the requested users and creates the limit ranges.

The following table represents the projects, their routes and login credentials:

Service name	Route	Login credentials
gogs	gogs-tasks-dev.apps.7a77.example.opentlc.com	gogs/gogs
jenkins	jenkins-tasks-dev.apps.7a77.example.opentlc.com	andrew/r3dh4t1!
nexus	nexus-tasks-dev.apps.7a77.example.opentlc.com	admin/admin123
sonarqube	sonarqube-tasks-dev.apps.7a77.example.opentlc.com	admin/admin
tasks	tasks-tasks-prod.apps.7a77.example.opentlc.com	
node-js-app	nodejs-mongo-persistent-smoke-test.apps.7a77.example.opentlc.com	

Release version – 3.11.43

Instructor – Jindrich Kana

Venue – Garni Hotel Centar, Novi Sad Serbia

Participant – Jelena Saveljic (jelena.saveljic@devoteam.com)

Jenkins

Pipeline tasks-dev/tasks-pipeline

Full project name: tasks-dev/tasks-dev-tasks-pipeline

[add description](#)

Disable Project



[Recent Changes](#)

Stage View

Average stage times:
(Average full run time: ~5min 2s)

#1
Dec 12
10:07
No Changes

Build App	Test	Code Analysis	Archive App	Create Image Builder	Build Image	Create DEV	Deploy DEV	Promote to STAGE?	Deploy STAGE
1min 50s	9s	20s	7s	6s	10s	57s	484ms	527ms	2s
1min 50s	9s	20s	7s	6s	10s	57s	484ms	527ms (rounded for 30s)	2s

Nexus

Not secure | nexus-tasks-dev.apps.7a77.example.opentlc.com

Click to go back, hold to see history

Search components

OS 3.12.1-01

Sign in

Welcome Learn about Sonatype Nexus Repository Manager

⚠ Nexus Repository 3.14 is now available with new cleanup policies, performance improvements and updates to search. [Release notes](#) [Download it now](#)

Get Started

- 🔄 **Upgrading**
[Upgrade to the latest version](#)
- 🔧 **Configuration**
[Set things up properly](#)
- 📖 **Documentation**
[Visit our help site](#)
- 👤 **Community**
[Ask and answer questions](#)

Repository Formats

[APT](#) [Composer](#) [Conan](#) [CPAN](#) [Docker](#) [ELPA](#) [Git LFS](#) [Helm](#) [Maven](#) [npm](#)

[NuGet](#) [P2](#) [PyPI](#) [R](#) [Raw](#) [RubyGems](#) [YUM](#) [Yum](#)

* Community supported

DepShield Survey

Open Source projects work diligently to fix disclosed vulnerabilities. Why shouldn't your project be powered with the knowledge of when and where these vulnerabilities exist and how to eradicate them? Checkout Sonatype DepShield, a GitHub App used by developers to identify and remediate vulnerabilities in their open source dependencies.

Provide Feedback in <10 seconds

Review the below messaging and icon design for DepShield in the GitHub Marketplace. Answer all 7 questions [here!](#)

Sonatype DepShield

Monitor your open source components for security vulnerabilities - goodbye 🐛.

All Day DevOps 2018

[Register now](#)

Nexus Firewall

Block high risk OSS components before they are cached in your repository manager.

[Learn more](#)

Work At Sonatype

[See openings](#)

Sonarqube

The screenshot shows the Sonarqube web interface. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, and Quality Gates. A search bar is also present. The main content area is divided into several sections:

- Continuous Code Quality:** Includes a 'Log in' button and a 'Read documentation' link. It also shows statistics: 1 Projects Analyzed, 2 Bugs, 1 Vulnerabilities, and 17 Code Smells.
- Multi-Language:** Lists 25+ programming languages supported, including Java, C/C++, C#, COBOL, ABAP, HTML, PHP, JavaScript, TypeScript, Objective C, and XML.
- Quality Model:** Explains the quality model with three pillars: Bug track code that is demonstrably wrong, Vulnerabilities are raised on code that is potentially vulnerable to exploitation, and Code Smells will confuse maintainers or give them pause.
- Write Clean Code:** Discusses the importance of maintaining a clean code base.
- Enforce Quality Gate:** Explains how the Quality Gate ensures the next version's quality is better than the last.
- Run Analysis With A SonarQube Scanner:** Provides information on how to run the scanner in different environments.
- Footer:** Includes logos for various integrations like SonarQube Scanner CLI, Maven, Gradle, Jenkins, and others.

Task-prod

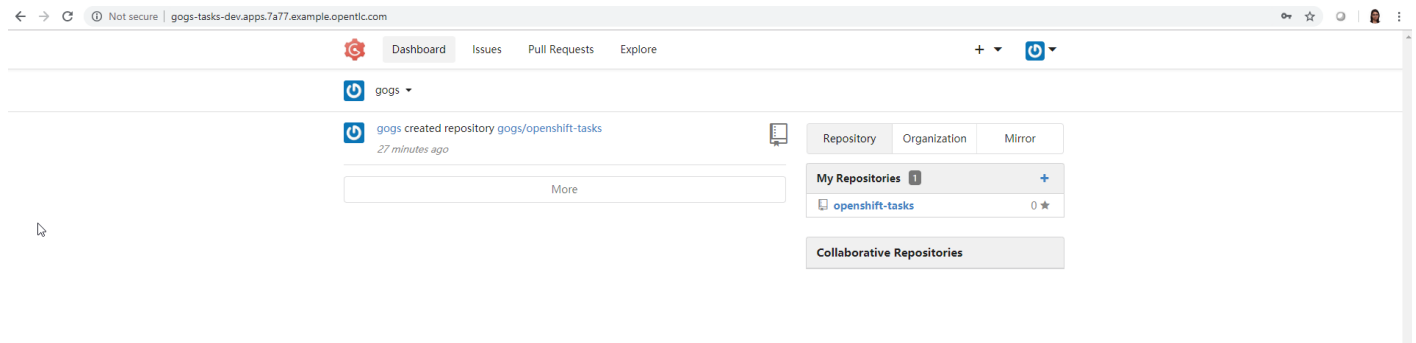
The screenshot shows the OpenShift Tasks Demo web interface. It features a dashboard with several interactive sections:

- Logger:** Contains buttons for 'Log Info', 'Log Warning', and 'Log Error'.
- Load Generator:** Includes a 'Seconds' input field and a 'Load!' button.
- Danger Zone:** Contains buttons for 'HEALTHY', 'Toggle Health', and 'Kill Instance'.
- Info:** Displays a table with system information:

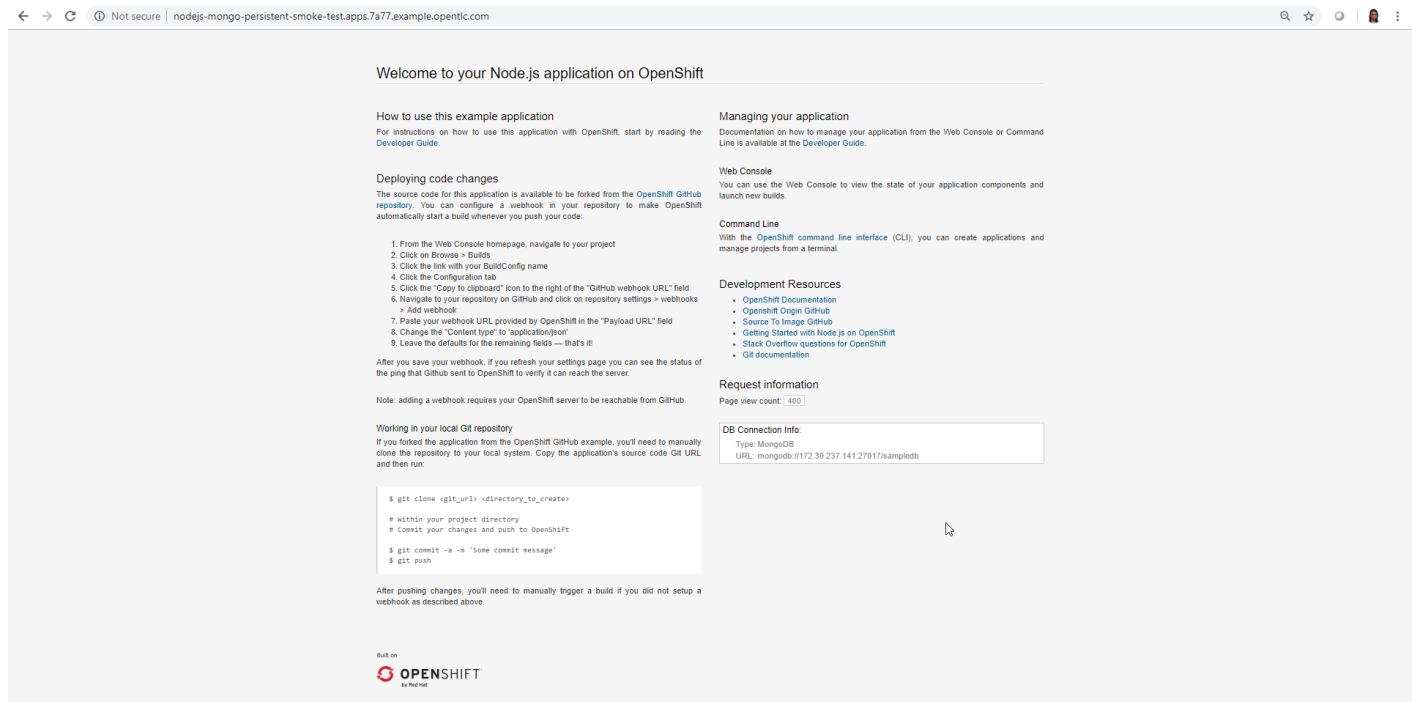
Pod Hostname	tasks-3-f522q
Pod IP	null
Used Memory	1250 MB
Session ID	dcs44hQvCiW1E2d1ohBlacwqRZgSiUL92IWMN

- Messages:** A section for displaying messages, currently showing 'Nothing to report.'

Gogs



Nodejs-mongo-persistent



Putty

```
PLAY RECAP *****
infranode1.7a77.internal : ok=177 changed=47 unreachable=0 failed=0
infranode2.7a77.internal : ok=177 changed=47 unreachable=0 failed=0
loadbalancer.7a77.internal : ok=61 changed=13 unreachable=0 failed=0
localhost : ok=66 changed=23 unreachable=0 failed=0
master1.7a77.internal : ok=1194 changed=432 unreachable=0 failed=0
master2.7a77.internal : ok=363 changed=128 unreachable=0 failed=0
master3.7a77.internal : ok=363 changed=128 unreachable=0 failed=0
node1.7a77.internal : ok=176 changed=46 unreachable=0 failed=0
node2.7a77.internal : ok=176 changed=46 unreachable=0 failed=0
node3.7a77.internal : ok=176 changed=46 unreachable=0 failed=0
support1.7a77.internal : ok=95 changed=8 unreachable=0 failed=0

INSTALLER STATUS *****
Initialization : Complete (0:01:00)
Health Check : Complete (0:00:13)
Node Bootstrap Preparation : Complete (0:03:54)
etcd Install : Complete (0:01:12)
NFS Install : Complete (0:00:13)
Load Balancer Install : Complete (0:00:20)
Master Install : Complete (0:06:00)
Master Additional Install : Complete (0:01:22)
Node Join : Complete (0:00:53)
Hosted Install : Complete (0:01:04)

    The use of NFS for the core OpenShift Container Platform components is not recommended, as NFS (and the NFS Protocol) does not provide the proper consistency needed for the applications that make up the OpenShift Container Platform infrastructure.

Cluster Monitoring Operator : Complete (0:00:49)
Web Console Install : Complete (0:00:24)
Console Install : Complete (0:00:24)
Metrics Install : Complete (0:02:24)
metrics-server Install : Complete (0:00:51)
Logging Install : Complete (0:03:28)
Service Catalog Install : Complete (0:02:23)
[root@bastion ~]#
```