

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/smitric/advc_dep_homework.git

Step	Description	Command
1.	be root	sudo -i
2.	clone the git repo	git clone https://github.com/smitric/advc_dep_homework.git
3.	run the ansible playbook	ansible-playbook ./advc_dep_homework/homework.yaml
4.	uninstall the cluster	sh ./advc_dep_homework/scripts/uninstall.sh

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 provides 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.cdb7.example.opentlc.com	gogs/gogs
Jenkins	jenkins-tasks-dev.apps.cdb7.example.opentlc.com	andrew/r3dh4t1!
Nexus	nexus-tasks-dev.apps.cdb7.example.opentlc.com	admin/admin123
Sonarqube	sonarqube-tasks-dev.apps.cdb7.example.opentlc.com	admin/admin
Tasks	tasks-tasks-prod.apps.cdb7.example.opentlc.com	
Node-js-app	nodejs-mongo-persistent-smoke-test.apps.cdb7.example.opentlc.com	

Release version – 3.10.34

Instructor – Jindrich Kana

Venue – Garni Hotel Centar, Novi Sad Serbia

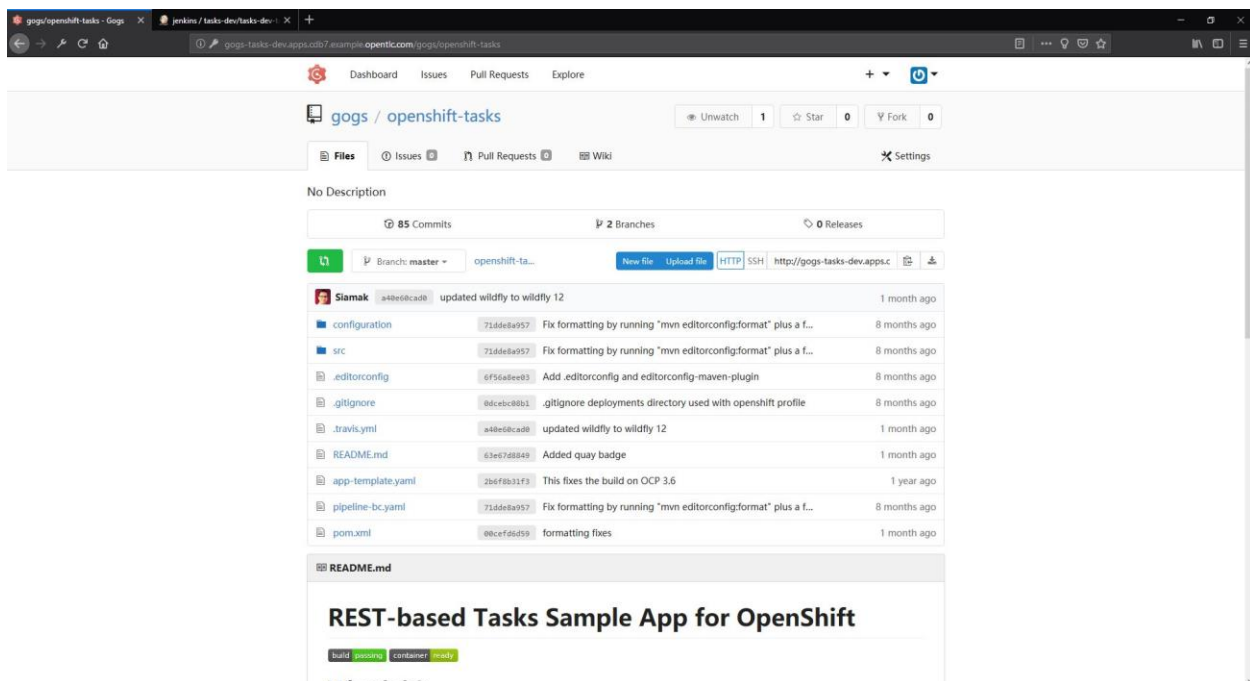
Participant – Saša Mitrić (sasa.mitric@devoteam.com)

Playbook recap

```
PLAY RECAP *****[infranodel1.cdb7.]
infranodel1.cdb7.internal : ok=188 changed=75 unreachable=0 failed=0
loadbalancer1.cdb7.internal : ok=186 changed=18 unreachable=0 failed=0
localhost : ok=67 changed=24 unreachable=0 failed=0
master1.cdb7.internal : ok=1135 changed=447 unreachable=0 failed=0
master2.cdb7.internal : ok=374 changed=156 unreachable=0 failed=0
master3.cdb7.internal : ok=374 changed=156 unreachable=0 failed=0
node1.cdb7.internal : ok=188 changed=75 unreachable=0 failed=0
node2.cdb7.internal : ok=188 changed=75 unreachable=0 failed=0
node3.cdb7.internal : ok=188 changed=75 unreachable=0 failed=0
support1.cdb7.internal : ok=75 changed=17 unreachable=0 failed=0

INSTALLER STATUS *****Initialization
: Complete (0:00:34)
Health Check : Complete (0:00:21)
Node Bootstrap Preparation : Complete (0:07:00)
etcd Install : Complete (0:00:52)
NFS Install : Complete (0:00:12)
Load Balancer Install : Complete (0:00:10)
Master Install : Complete (0:04:32)
Master Additional Install : Complete (0:00:57)
Node Join : Complete (0:00:22)
Hosted Install : Complete (0:00:40)
Web Console Install : Complete (0:00:22)
Metrics Install : Complete (0:02:00)
Logging Install : Complete (0:02:50)
Prometheus Install : Complete (0:00:36)
Service Catalog Install : Complete (0:01:28)
[root@bastion ~]#
```

Gogs



Jenkins

The screenshot shows the Jenkins Pipeline interface for a pipeline named 'tasks-dev/tasks-pipeline-2'. The pipeline is currently in a 'Deploy STAGE' state, which is 7s into its execution. The pipeline consists of several stages: Start, Build App, Test, Code Analysis, Archive App, Create Image Builder, Build Image, Create DEV, Deploy DEV, Promote to STAGE?, and Deploy STAGE. The 'Deploy STAGE' stage is currently active and shows a list of tasks being executed, including internal utility functions for OpenShift DSL and reading files from the workspace. The pipeline was last committed by user 'andrew' a minute ago.

tasks-dev / tasks-dev/tasks-pipeline-2

Branch: — 2m 12s No changes
Commit: — a minute ago Started by user andrew

Description OpenShift Build tasks-dev/tasks-pipeline-2

Start Build App Test Code Analysis Archive App Create Image Builder Build Image Create DEV Deploy DEV Promote to STAGE? Deploy STAGE End

Deploy STAGE - 7s

- > Internal utility function for OpenShift DSL <1s
- > Internal utility function for OpenShift DSL <1s
- > /var/run/secrets/kubernetes.io/serviceaccount/token — Read file from workspace <1s
- > Internal utility function for OpenShift DSL <1s
- > /var/run/secrets/kubernetes.io/serviceaccount/token — Read file from workspace <1s
- > Internal utility function for OpenShift DSL 1s
- > /var/run/secrets/kubernetes.io/serviceaccount/token — Read file from workspace <1s
- > Internal utility function for OpenShift DSL <1s
- > /var/run/secrets/kubernetes.io/serviceaccount/token — Read file from workspace <1s
- > Internal utility function for OpenShift DSL <1s
- > /var/run/secrets/kubernetes.io/serviceaccount/token — Read file from workspace <1s
- > Internal utility function for OpenShift DSL 2s
- > /var/run/secrets/kubernetes.io/serviceaccount/token — Read file from workspace <1s
- > Internal utility function for OpenShift DSL <1s

Nexus

The screenshot shows the Sonatype Nexus Repository Manager interface. The main header displays 'Welcome - Nexus Repository Manager' and 'Learn about Sonatype Nexus Repository Manager'. Below the header, there are sections for 'Get Started' (Upgrading, Configuration, Documentation, Community), 'Repository Formats' (listing various supported formats like APT, Composer, Conan, etc.), and a 'DepShield Survey' section. The survey asks if the user is interested in using DepShield, what format they would like to see supported next, and if they are interested in providing UX feedback. A 'Done' button is at the bottom of the survey.

gogs/openshift-tasks - Gogs jenkins / tasks-dev/tasks-dev Welcome - Nexus Repository Manager

Search components

Search

Upload

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

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

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.

Are you interested in using DepShield?

What format would you like to see supported next for DepShield?

Are you interested in providing UX feedback on DepShield? Qualified participants will receive a \$100 Amazon gift card for a 1-hour call. Enter your email below:

Done

Sonarqube

The screenshot shows the SonarQube web interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. The main content area displays the project 'JBoss EAP - Tasks JAX-RS App' with a 'Passed' status. The project metrics are: 2 Bugs, 1 Vulnerabilities, 17 Code Smells, 5.9% Coverage, and 0.0% Duplications. The left sidebar contains filters for Quality Gate, Reliability, Security, Maintainability, and Coverage. A warning message at the bottom states: 'Embedded database should be used for evaluation purpose only'.

Task-prod

The screenshot shows the OpenShift Tasks Demo web interface. The top navigation bar includes links for Home, OpenShift Demo Tasks, and Projects. The main content area displays the 'OpenShift Tasks Demo' page. The page includes a 'Logger' section with buttons for Log Info, Log Warning, and Log Error. A 'Load Generator' section with a 'Seconds' input and a 'Load!' button. A 'Danger Zone' section with buttons for HEALTHY, Toggle Health, and Kill Instance. An 'Info' section displaying pod details (Pod Hostname, Pod IP, Used Memory, Session ID). A 'Messages' section with the text 'Nothing to report'.

Nodejs-mongo-persistent

The screenshot shows a web browser window with multiple tabs. The active tab is titled 'Welcome to OpenShift' and displays the URL 'nodes-mongo-persistent-smoke-test.apps.cib7.example.opentlc.com'. The page content is a 'Welcome to your Node.js application on OpenShift' message. It includes sections for 'How to use this example application', 'Managing your application', 'Deploying code changes', 'Web Console', 'Command Line', 'Development Resources', 'Request information', and 'DB Connection Info'. A terminal window at the bottom shows the commands to clone the repository, commit changes, and push them to OpenShift.

Welcome to your Node.js application on OpenShift

How to use this example application

For instructions on how to use this application with OpenShift, start by reading the [Developer Guide](#).

Managing your application

Documentation on how to manage your application from the Web Console or Command Line is available at the [Developer Guide](#).

Deploying code changes

The source code for this application is available to be forked from the [OpenShift GitHub repository](#). You can configure a webhook in your repository to make OpenShift automatically start a build whenever you push your code:

1. From the Web Console homepage, navigate to your project
2. Click on Browse > Builds
3. Click the link with your BuildConfig name
4. Click the Configuration tab
5. Click the "Copy to clipboard" icon to the right of the "GitHub webhook URL" field
6. Navigate to your repository on GitHub and click on repository settings > webhooks
7. Add webhook
8. Paste your webhook URL provided by OpenShift in the "Payload URL" field
9. Change the "Content type" to "application/json"
10. Leave the defaults for the remaining fields — that's it!

After you save your webhook, if you refresh your settings page you can see the status of the ping that GitHub sent to OpenShift to verify it can reach the server.

Note: adding a webhook requires your OpenShift server to be reachable from GitHub.

Working in your local Git repository

If you forked the application from the OpenShift GitHub example, you'll need to manually clone the repository to your local system. Copy the application's source code Git URL and then run:

```
$ git clone <git_url> <directory_to_create>

# Within your project directory
# Commit your changes and push to OpenShift

$ git commit -a -m 'Some commit message'
$ git push
```

Web Console

You can use the Web Console to view the state of your application components and launch new builds.

Command Line

With the OpenShift [command line interface \(CLI\)](#), you can create applications and manage projects from a terminal.

Development Resources

- [OpenShift Documentation](#)
- [OpenShift Origin GitHub](#)
- [Source to Image GitHub](#)
- [Getting Started with Node.js on OpenShift](#)
- [Stack Overflow questions for OpenShift](#)
- [Git documentation](#)

Request information

Page view count: [536](#)

DB Connection Info:

Type: MongoDB
URL: mongodb://172.30.74.26:27017/sampledb