

Red Hat

Prepared For:

MitziCom

Date delivered : 30 March 2018

TABLE OF CONTENTS

[Table of Contents](#)

[Document Information](#)

[Originator](#)

[Owner](#)

[Copyright](#)

[Distribution](#)

[Confidentiality](#)

[Additional Copies](#)

[Purpose](#)

[Executive Summary](#)

[Red Hat Consulting Contact Information](#)

[REFERENCE](#)

[12 Factor Applications and Controllers](#)

[CI/CD Tool](#)

[Building Applications](#)

[Pipelines:](#)

[PoC Description and requirements:](#)

[CI/CD Infrastructure Setup](#)

[GIT REPOSITORY](#)

[NEXUS](#)

[HTTP Console](#)

[SAVED TEMPLATES & Description](#)

[SonarQube](#)

[SAVED TEMPLATES & Description](#)

[HTTP link :](#)

[Jenkins](#)

[SAVED TEMPLATES & Description](#)

[HTTP LINK](#)

[Custom Jenkins Slave Pod](#)

[Unit tests](#)

[1\) create and run a test pipeline on Jenkins](#)

[2\) Test Local Workstation Build](#)

[OpenShift Setup](#)

[Create two OpenShift projects "MitziCom_parkmap Development" for development \(including test\) and "MitziCom parkmap Production" for production.](#)

[Set up the correct permissions for Jenkins to manipulate objects in the development and production projects.](#)

[Set up the correct permissions for the production project to deploy images from the development project.](#)

[Set up a MongoDB database in the development project.](#)

[Set up a replicated MongoDB database \(StatefulSet\) with at least three replicas in the production project.](#)

[Create build configurations in the development project for each application](#)

[MLBPARKS APP DEFINITION](#)

[NATIONALPARKS APP DEFINITION](#)

[PARKSMAPS APP DEFINITIONS](#)

[Create deployment configurations in both the development and production projects.](#)

[Configure the applications using ConfigMaps.](#)

[WARNING BEST PRACTICE : ADDING HEALTH CHECK on the Application](#)

[SAVED TEMPLATES & DescriptionMitziCom](#)

[Development Pipeline](#)

[Create a Jenkins pipeline for each of the three services and store it in the source code repositories.](#)

[Make sure that the development pipelines:](#)

[Deployment Pipeline](#)

[Continue your pipeline development with the following:](#)

[Challenges, Resolutions, and Recommendations](#)

[Relevant Red Hat Documentation](#)

[Relevant Red Hat Training](#)

[APPENDIX 1](#)

[Scrapbook](#)

[APPLICATION TESTING](#)

[Main app test](#)

[APPENDIX 2](#)

[Configure MongoDB in Dev](#)

[Configure MongoDB in Prod](#)

[0\) Create Secret \(as an example provided, not used by PoC\)](#)

[1\) Create the internal headless service that the pods in the StatefulSet use to communicate with each other.](#)

[2\) Create the regular service that database clients use to connect to the database.](#)

[3\) Create StatefulSet for MongoDB Database](#)

[4\) Scale the StatefulSet \(in case needed\)](#)

[APPENDIX 4](#)

[PIPELINE IN OPENSIFT FOR APPLICATION MLBPARKS](#)

[PIPELINE IN OPENSIFT FOR APPLICATION NATIONALPARKS](#)

[PIPELINE IN OPENSIFT FOR APPLICATION PARKSMAP](#)

[APPENDIX 5](#)

[Appendix A: Engaging Red Hat Global Support Services](#)

DOCUMENT INFORMATION

Originator

Red Hat Consulting

Owner

Red Hat Consulting - Confidential / Restricted Distribution

Copyright

This document contains proprietary information which is for exclusive use of Red Hat, Inc. and is not to be shared with personnel other than Red Hat, Inc. This document, and any portion thereof, may not be copied, reproduced, photocopied, stored electronically on a retrieval system, or transmitted without the express written consent of the owner.

Red Hat Consulting does not warrant this document to be free of errors or omissions. Red Hat Consulting reserves the right to make corrections, updates, revisions, or changes to the information contained herein. Red Hat Consulting does not warrant the material described herein to be free of patent infringement.

Unless provided otherwise in writing BY RED HAT Consulting, the information and programs described herein are provided "as is" without warranty of any kind, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. In no event will RED HAT Consulting, its officers, directors, or employees or affiliates of RED HAT Consulting, their respective officers, directors, or employees be liable to any entity for any special, collateral, incidental, or consequential damages, including without any limitation, for any lost profits or lost savings, related or arising in any way from or out of the use or inability to use the information or programs set forth herein, even if it has been notified of the possibility of such damage by the purchaser or any third party.

Distribution

Do not forward or copy without written permission from Red Hat Consulting.

Copies of this document are restricted to the following names:

Red Hat, Inc.
MitziCom

Confidentiality

All information supplied to Intel for the purpose of this engagement is to be considered Red Hat confidential.

Additional Copies

Additional copies of this document can be obtained from the Service Delivery Manager listed in the [Red Hat Consulting Contact Information](#) section.

PURPOSE

This report documents the overview, contact information, and technical details of the activities performed during the consulting engagement. Its purpose is to serve as a reference after the engagement is complete.

EXECUTIVE SUMMARY

The Red hat Team and I want to thank you for the opportunity to work with your teams on this MitziCom project, driven as a Proof of Concept (PoC), with some elements that will be valuable for your company to leverage for your next "real" projects.

As a telecommunications' company, MitziCom provides hosting and cloud services to a variety of clients, from medium-sized companies to enterprise giants.

This proof-of-concept is using Red Hat OpenShift Container Platform. The purpose of the PoC is to determine the feasibility of using Red Hat OpenShift as a target for an existing Java-based microservices workload. We have proposed fully integrated CI/CD pipelines using a source code control system (Github in that case), an artifact repository (Nexus located and hosted directly on OpenShift as a project), code analysis (SonarQube technology also located and hosted on OpenShift as a project), and deployed to production in a blue-green strategy orchestrated by Jenkins (located and hosted on OpenShift as a project).

For that project, you asked us to use the existing MitziCom application called ParksMap (referenced link : <https://github.com/wkulhanek/ParksMap.git>). We have cloned the application on a separate instance of github and manage to update that clone(no touch on the existing repository). We may also remind you that there are some improvement of the application that Red Hat consultant would be happy to engage with MitziCom in order to make it even more valuable for your customers.

This application consists into three microservices—two back-end services and one front-end service calling the back-end services.

The below document and the repository where it's located (<https://github.com/lucpierson/ParksMap.git>) is a description of the work done, sort of a tracing document, and is to be considered as a MitziCom asset. We may delete this repository as soon as it's required. Sharing the app on GitHub went through a MitziCom approval before the PoC started.

Please provide some feedback in our next meeting as a follow-up and we will be happy to answer any question that could remain.

With our best regards,

Luc Pierson

Technical Director @ Red Hat EMEA and OpenSource enthusiast.

luc.pierson@redhat.com

<http://www.linkedin.com/in/lucpierson>

RED HAT CONSULTING CONTACT INFORMATION

The table below contains the contact information for the Red Hat Consulting personnel that supported the delivery of this consulting engagement.

Role	Name	Email
Solution Architect Lead	Luc Pierson	lpierson@redhat.com
Solution Reviewed	Wolfgang Kulhanek	wkulhanek@redhat.com

REFERENCE

This Proof of Concept will find many answers and explanations through our Partner training team GPTE and the advanced training you might want to have a look at.

Here are some links of the main chapters

please send an email to Wolfgang Kulhanek <wkulhane@redhat.com> with me in CC Luc Pierson <lperson@redhat.com> in case of problem accessing those links

12 Factor Applications and Controllers

Intro :

http://appdev-course-wk-course.apps.fra.example.opentlc.com/01_Course_Intro/AllSlides.html

http://appdev-course-wk-course.apps.fra.example.opentlc.com/02_12_Factor_Applications/AllSlides.html

http://appdev-course-wk-course.apps.fra.example.opentlc.com/03_Controllers/AllSlides.html

Lab Solution:

https://www.opentlc.com/labs/ocp_advanced_development/03_1_Controllers_Solution_Lab.html

CI/CD Tool

Intro :

http://appdev-course-wk-course.apps.fra.example.opentlc.com/04_CICD_Tools/AllSlides.html

Lab Solution:

https://www.opentlc.com/labs/ocp_advanced_development/04_1_CICD_Tools_Solution_Lab.html

Building Applications

Intro:

http://appdev-course-wk-course.apps.fra.example.opentlc.com/05_Building_Applications/AllSlides.html

Lab Solution:

https://www.opentlc.com/labs/ocp_advanced_development/05_1_Building_Applications_Solution_Lab.html

Pipelines:

Intro:

http://appdev-course-wk-course.apps.fra.example.opentlc.com/06_Pipelines/AllSlides.html

Lab Solution:

https://www.opentlc.com/labs/ocp_advanced_development/06_1_Pipelines_Solution_Lab.html

PoC Description and requirements:

https://www.opentlc.com/labs/ocp_advanced_development/07_1_Assignment_Lab.html

At any times, please refer to the scrapbook in APPENDIX 1 to get some easy printed commands that might be useful to follow up

CI/CD INFRASTRUCTURE SETUP

GIT REPOSITORY

Would you have a private gitlab, gogs or other git for private, installed in openshift or not is up to MitziCom.

for this PoC purpose based on sharing with external people, we decided to clone the original project on github and share from there (was approved before the PoC started)

We considered using GitHub for this PoC <https://github.com/lucpierson/ParksMap>

should you need to connect with use/password, please contact lperson@redhat.com to get the password.

URL : <https://github.com/lucpierson/ParksMap.git>

WARNING :

the 3 services asked to use by MitziCom (mlbparks, nationalparks, parksmapi) are located on same Git project (PoC).

As microservices, we recommend that they should be located on 3 different projects to reflect their μ S independence and refer to nexus settings outside the projects

NEXUS

We consider that for an enterprise project, you might consider hosting a global nexus instances outside OpenShift.

Choosing to host Nexus in OpenShift for this PoC is a way to prove to MitziCom the extended capabilities of the solution.

Nexus is located on the OpenShift project "lperson-nexus", exposed on 2 ports, for regular console and update management (8081) and Docker registry (5000)

Nexus3 being the application is set up with memory limits 2gb and persistent storage 4gb.

Sonatype provides a Nexus 3 image labeled sonatype/nexus3:latest in DockerHub.

We use the "Recreate" deployment strategy rather than Rolling to set up Nexus.

Nexus requires a large amount of memory. We set the memory request to 1Gi and the memory limit to 2Gi .

The Nexus 3 image defines a VOLUME at /nexus-data .

Nexus is initialized and configured using an existing configuration script that you might want to look at :

- duplicated in the project :
https://github.com/lucpierson/ParksMap/tree/master/Nexus/setup_nexus3.sh
- original :
https://raw.githubusercontent.com/wkulhanek/ocp_advanced_development_resources/master/nexus/setup_nexus3.sh

Nexus app default user ID (admin) and password (admin123)

Because located on OpenShift cluster, Nexus may be targeted 2 ways

- External and Internal traffic, ex <http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com>
- Internal traffic (when called from another app in the same openShift Cluster), ex :

nexus3.lpierson-nexus.svc.cluster.local:5000

created also : an OpenShift route called nexus-registry that uses edge termination for the TLS encryption and exposes port 5000.

```
oc create route edge nexus-registry --service=nexus-registry --port=5000
```

HTTP Console

<http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/admin/admin123>

SAVED TEMPLATES & Description

please see templates and according commands @ <https://github.com/lucpierson/ParksMap/tree/master/Nexus>

SonarQube

Continuous Inspection : SonarQube provides the capability to not only show health of an application but also to highlight issues newly introduced. With a Quality Gate in place, you can fix the leak and therefore improve code quality systematically.

SonarQube is installed on OpenShift (lpierson-sonarqube) project for demo purpose of code quality verification in the pipeline. for PoC acceleration purposes, we choose to use an existing SonarQube configuration ready to be used and opened as a docker image located on docker-hub (image : wkulhanek/sonarqube:6.7.2)

SonarQube app default user ID (admin) and password (admin). the application relies on a postgresql instance installed on openshift also (separated Pod in same project, user=sonar/pw=sonar)

SAVED TEMPLATES & Description

please see templates and according commands @ <https://github.com/lucpierson/ParksMap/tree/master/SonarQube>

HTTP link :

<http://sonarqube-lpierson-sonarqube.apps.na37.openshift.opentlc.com/projects/admin/admin>

Jenkins

Jenkins is an open source automation server written in Java. Jenkins helps to automate the non-human part of

the software development process, with continuous integration and facilitating technical aspects of continuous delivery.

Jenkins is installed on OpenShift (lpierson-jenkins) project for demo purpose of CI/CD integration and pipeline holding. it uses the OpenShift proposed standard template, "Jenkins service, with persistent storage". We have set up a persistent Jenkins instance with 2 GB of memory and a persistent volume claim of 4 GB.

We consider that after the PoC, the review of Jenkins configuration into the global MitziCom architecture is to be reviewed as

- installed globally on a dedicated server instance for all project and with a global administration
- installed on local project instances (as in this PoC) but limited to transient pipelines that are linked to the project.

SAVED TEMPLATES & Description

please see templates and according commands @ <https://github.com/lucpierson/ParksMap/tree/master/Jenkins>

HTTP LINK

<https://jenkins-lpierson-jenkins.apps.na37.openshift.opentlc.com/>

user and passwords are identical to OpenShift (SSO Configuration, PoC was using lpierson-redhat.com)

Custom Jenkins Slave Pod

The Jenkins Maven slave pod does not have skopeo installed. However, skopeo is needed to be available in order to move your built container images into another registry.

This means that we build a custom slave pod. we extend the existing slave pod and install skopeo into that pod.

Then we make this container image available to OpenShift by pushing it into the OpenShift Container Registry.

Because we are building this image ourselves we use your current Jenkins project (lpierson-jenkins) as the home for the container image.

Unit tests

1) create and run a test pipeline on Jenkins

```
Started by user lpierson-redhat.com
[Pipeline] node
Still waiting to schedule task
maven-appdev-0t8v6 is offline
Running on maven-appdev-0t8v6 in /tmp/workspace/testCustomSlavePod
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Test skopeo)
[Pipeline] sh
[testCustomSlavePod] Running shell script
+ skopeo --version
skopeo version 0.1.28
```

```
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] }  
[Pipeline] // node  
[Pipeline] End of Pipeline  
Finished: SUCCESS
```

2) Test Local Workstation Build

```
cd ~/ParksMap/mlbparks  
mvn clean install -DskipTests=true -s nexus_settings_openshift.xml  
mvn test -s nexus_settings_openshift.xml  
cd ~/ParksMap/nationalparks  
mvn clean package -DskipTests=true -s nexus_settings_openshift.xml  
mvn test -s .nexus_settings_openshift.xml  
cd ~/ParksMap/parksmap  
mvn clean package -DskipTests=true -s nexus_settings_openshift.xml  
spring-boot:repackage -Dcom.redhat.xpaas.repo.redhatga  
mvn test -s nexus_settings_openshift.xml  
mvn sonar:sonar -s nexus_settings_openshift.xml -Dsonar.host.url=http://$(oc get  
route sonarqube -n lpierson-sonarqube --template='{ .spec.host }')
```

==> OK, see logs of sonarQube for the parksmap app in APPENDIX 3 (as an example)

OPENSHIFT SETUP

For the PoC, we considered the following tasks, for each of 3 applications and the accordingly commands

Create two OpenShift projects "MitziCom parkmap Development" for development (including test) and "MitziCom parkmap Production" for production.

```
oc new-project lpierson-parkmap-dev --display-name "MitziCom parkmap Development"
oc new-project lpierson-parkmap-prod --display-name "MitziCom parkmap Production"
```

Set up the correct permissions for Jenkins to manipulate objects in the development and production projects.

```
oc policy add-role-to-user edit system:serviceaccount:lpierson-jenkins:jenkins -n lpierson-parkmap-dev
oc policy add-role-to-user edit system:serviceaccount:lpierson-jenkins:default -n lpierson-parkmap-dev
oc policy add-role-to-user view --serviceaccount=default -n lpierson-parkmap-dev
oc policy add-role-to-user edit system:serviceaccount:lpierson-jenkins:jenkins -n lpierson-parkmap-prod
oc policy add-role-to-user edit system:serviceaccount:lpierson-jenkins:default -n lpierson-parkmap-prod
oc policy add-role-to-user view --serviceaccount=default -n lpierson-parkmap-prod
```

Set up the correct permissions for the production project to deploy images from the development project.

```
oc policy add-role-to-group system:image-puller
system:serviceaccounts:lpierson-parkmap-prod -n lpierson-parkmap-dev
```

Set up a MongoDB database in the development project.

for associated commands, See APPENDIX 2 named "Configure MongoDB in Dev"

Set up a replicated MongoDB database (StatefulSet) with at least three replicas in the production project.

for associated commands, See APPENDIX 2 named "Configure MongoDB in Prod"

Create build configurations in the development project for each application

MLBPARKS APP DEFINITION

```
oc new-build --binary=true --name="mlbparks" jboss-eap70-openshift:1.6 -n
lpierson-parkmap-dev
oc new-app lpierson-parkmap-dev/mlbparks:0.0-0 --name=mlbparks
--allow-missing-imagestream-tags=true -n lpierson-parkmap-dev
oc set triggers dc/mlbparks --remove-all -n lpierson-parkmap-dev
# Create Blue Application
oc new-app lpierson-parkmap-dev/mlbparks:0.0 --name=mlbparks-blue
--allow-missing-imagestream-tags=true -n lpierson-parkmap-prod
oc set triggers dc/mlbparks-blue --remove-all -n lpierson-parkmap-prod
# Create Green Application
oc new-app lpierson-parkmap-dev/mlbparks:0.0 --name=mlbparks-green
--allow-missing-imagestream-tags=true -n lpierson-parkmap-prod
oc set triggers dc/mlbparks-green --remove-all -n lpierson-parkmap-prod
```

NATIONALPARKS APP DEFINITION

```
oc new-build --binary=true --name="nationalparks" redhat-openjdk18-openshift:1.2 -n
lpierson-parkmap-dev
oc new-app lpierson-parkmap-dev/nationalparks:0.0-0 --name=nationalparks
--allow-missing-imagestream-tags=true -n lpierson-parkmap-dev
oc set triggers dc/nationalparks --remove-all -n lpierson-parkmap-dev
# Create Blue Application
oc new-app lpierson-parkmap-dev/nationalparks:0.0 --name=nationalparks-blue
--allow-missing-imagestream-tags=true -n lpierson-parkmap-prod
oc set triggers dc/nationalparks-blue --remove-all -n lpierson-parkmap-prod
# Create Green Application
oc new-app lpierson-parkmap-dev/nationalparks:0.0 --name=nationalparks-green
--allow-missing-imagestream-tags=true -n lpierson-parkmap-prod
oc set triggers dc/nationalparks-green --remove-all -n lpierson-parkmap-prod
```

PARKSMAPS APP DEFINITIONS

```
oc new-build --binary=true --name="parksmapi" redhat-openjdk18-openshift:1.2 -n
lpierson-parkmap-dev
oc new-app lpierson-parkmap-dev/parksmapi:0.0-0 --name=parksmapi
--allow-missing-imagestream-tags=true -n lpierson-parkmap-dev
oc set triggers dc/parksmapi --remove-all -n lpierson-parkmap-dev
# Create Blue Application
oc new-app lpierson-parkmap-dev/parksmapi:0.0 --name=parksmapi-blue
--allow-missing-imagestream-tags=true -n lpierson-parkmap-prod
oc set triggers dc/parksmapi-blue --remove-all -n lpierson-parkmap-prod
# Create Green Application
oc new-app lpierson-parkmap-dev/parksmapi:0.0 --name=parksmapi-green
--allow-missing-imagestream-tags=true -n lpierson-parkmap-prod
oc set triggers dc/parksmapi-green --remove-all -n lpierson-parkmap-prod
```

Create deployment configurations in both the development and production projects.

```
oc expose dc mlbparks --port 8080 -n lpierson-parkmap-dev
oc expose svc mlbparks -n lpierson-parkmap-dev
oc label route mlbparks type=parksmap-backend -n lpierson-parkmap-dev
oc expose dc mlbparks-green --port 8080 -n lpierson-parkmap-prod
oc expose dc mlbparks-blue --port 8080 -n lpierson-parkmap-prod
# Expose Blue service as route to make blue application active
oc expose svc/mlbparks-blue --name mlbparks -n lpierson-parkmap-prod

oc expose dc nationalparks --port 8080 -n lpierson-parkmap-dev
oc expose svc nationalparks -n lpierson-parkmap-dev
oc label route nationalparks type=parksmap-backend -n lpierson-parkmap-dev
oc expose dc nationalparks-green --port 8080 -n lpierson-parkmap-prod
oc expose dc nationalparks-blue --port 8080 -n lpierson-parkmap-prod
# Expose Blue service as route to make blue application active
oc expose svc/nationalparks-blue --name nationalparks -n lpierson-parkmap-prod

oc expose dc parksmap --port 8080 -n lpierson-parkmap-dev
oc expose svc parksmap -n lpierson-parkmap-dev
oc expose dc parksmap-green --port 8080 -n lpierson-parkmap-prod
oc expose dc parksmap-blue --port 8080 -n lpierson-parkmap-prod
# Expose Blue service as route to make blue application active
oc expose svc/parksmap-blue --name parksmap -n lpierson-parkmap-prod
```

Manual update the DeploymentDescriptors environment variables of concerned applications with MongoDB connection information

- DB_HOST=mongodb (Name of the MongoDB Service)
- DB_PORT=27017 (Port the MongoDB Service is running under)
- DB_USERNAME=mongodb (Username for the MongoDB database)
- DB_PASSWORD=mongodb (Password for the user)
- DB_NAME=parks (Database Name)
- DB_REPLICASET=rs0 (only in production where use of stateful set)

Configure the applications using ConfigMaps.

```
oc create configmap mlbparks-config
--from-literal="application-users.properties=Placeholder"
--from-literal="application-roles.properties=Placeholder" -n lpierson-parkmap-dev
oc set volume dc/mlbparks --add --name=jboss-config
--mount-path=/opt/eap/standalone/configuration/application-users.properties
--sub-path=application-users.properties --configmap-name=mlbparks-config -n
lpierson-parkmap-dev
oc set volume dc/mlbparks --add --name=jboss-config1
--mount-path=/opt/eap/standalone/configuration/application-roles.properties
```



```
--sub-path=application-roles.properties --configmap-name=mlbparks-config -n  
lpierson-parkmap-dev  
# Create Blue Application  
oc create configmap mlbparks-blue-config  
--from-literal="application-users.properties=Placeholder"  
--from-literal="application-roles.properties=Placeholder" -n lpierson-parkmap-prod  
oc set volume dc/mlbparks-blue --add --name=jboss-config  
--mount-path=/opt/eap/standalone/configuration/application-users.properties  
--sub-path=application-users.properties --configmap-name=mlbparks-blue-config -n  
lpierson-parkmap-prod  
oc set volume dc/mlbparks-blue --add --name=jboss-config1  
--mount-path=/opt/eap/standalone/configuration/application-roles.properties  
--sub-path=application-roles.properties --configmap-name=mlbparks-blue-config -n  
lpierson-parkmap-prod  
# Create Green Application  
oc create configmap mlbparks-green-config  
--from-literal="application-users.properties=Placeholder"  
--from-literal="application-roles.properties=Placeholder" -n lpierson-parkmap-prod  
oc set volume dc/mlbparks-green --add --name=jboss-config  
--mount-path=/opt/eap/standalone/configuration/application-users.properties  
--sub-path=application-users.properties --configmap-name=mlbparks-green-config -n  
lpierson-parkmap-prod  
oc set volume dc/mlbparks-green --add --name=jboss-config1  
--mount-path=/opt/eap/standalone/configuration/application-roles.properties  
--sub-path=application-roles.properties --configmap-name=mlbparks-green-config -n  
lpierson-parkmap-prod  
oc expose dc/mlbparks-blue --port 8080 -n lpierson-parkmap-prod
```

WARNING BEST PRACTICE : ADDING HEALTH CHECK on the Application

in case the app don't answer, the OpenShift cluster will take the decision to re-start the pod. This should be done using

- Deployment, Actions, add healthCheck
completing "Readiness Probe" and "Liveness Probe"

This was out of scope for the PoC exercise, but we will be happy to put in place the solution along with you

SAVED TEMPLATES & DescriptionMitziCom

please see templates and according commands

@ <https://github.com/lucpierson/ParksMap/tree/master/lpierson-parkmap-dev>

@ <https://github.com/lucpierson/ParksMap/tree/master/lpierson-parkmap-prod>

DEVELOPMENT PIPELINE

Create a Jenkins pipeline for each of the three services and store it in the source code repositories.

the Jenkinsfiles for the 3 applications can be viewed into the following directories

<https://github.com/lucpierson/ParksMap/tree/master/mlbparks>
<https://github.com/lucpierson/ParksMap/tree/master/nationalparks>
<https://github.com/lucpierson/ParksMap/tree/master/parksmap>

some of the steps in Jenkinsfiles were bypassed (comment using `"/"` before the actions) to make some memory economy (tigh resources)

the Pipeline is built from the 3 jenkinsfile located on Github.

In the configuration for MitziCom PoC, the pipelines can be triggered directly on the Jenkins console or through the pipeline menu in OpenShift console, project lpierson-jenkins. See APPENDIX 4 for the OpenShift pipeline configuration that were used during PoC.

Make sure that the development pipelines:

Build the source code, using the artifact repository as a Maven proxy cache.

```
sh "${mvnCmd} clean package -DskipTests"
```

Execute the following tests in parallel and present the results in the build log:

```
Unit tests : sh "${mvnCmd} test"
```

```
Code coverage tests : sh "${mvnCmd} sonar:sonar
```

```
-Dsonar.host.url=http://sonarqube-lpierson-sonarqube.apps.na37.openshift.opentlc.com/
```

```
-Dsonar.projectName=${JOB_BASE_NAME}-${devTag} "
```

Tag the image with the version and build number : `sh "${mvnCmd} versions:set`

```
-DnewVersion=${devTag} "
```

Upload the generated artifact to the artifact repository :

```
sh "${mvnCmd} deploy -DskipTests=true
```

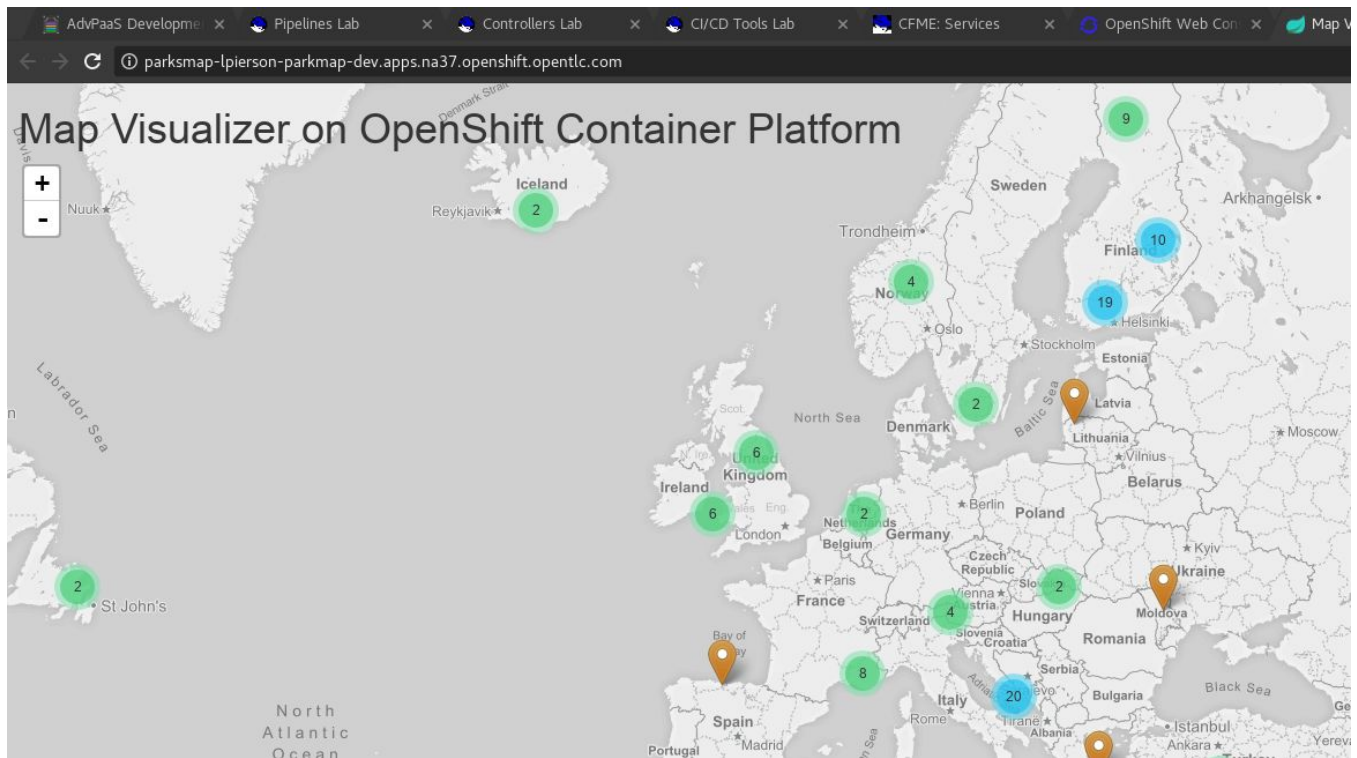
```
-DaltDeploymentRepository=nexus::default::http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/releases/"
```

Run an integration test if the service built was a back-end service

```
sh "curl -i -u 'mlbparks:redhat1' -H 'Content-Length: 0' -X POST  
http://mlbparks.lpierson-parkmap-dev.svc.cluster.local:8080/ws/mlbparks/integration_test_1"
```

Upload the tested container image to another (Nexus) Docker registry, example of command

```
sh "skopeo copy --src-tls-verify=false --dest-tls-verify=false --src-creds
openshift:\$(oc whoami -t) --dest-creds admin:admin123
docker://docker-registry.default.svc.cluster.local:5000/lpierson-parkmap-dev/mlbparks:
${devTag} docker://nexus-registry.lpi-nexus.svc.cluster.local:5000/mlbparks:${devTag}"
```



DEPLOYMENT PIPELINE

Continue your pipeline development with the following:

We tag the image as version for production deployment, eg

```
openshiftTag alias: 'false', destStream: 'mlbparks', destTag: prodTag,  
destinationNamespace: 'lpierson-parkmap-dev', namespace: 'lpierson-parkmap-dev',  
srcStream: 'mlbparks', srcTag: devTag, verbose: 'false'
```

We deploy the newly built microservice using a blue-green strategy, eg

```
// Update the Image on the Production Deployment Config  
sh "oc set image dc/${destApp}  
${destApp}=docker-registry.default.svc:5000/lpierson-parkmap-dev/mlbparks:${prodTag}  
-n lpierson-parkmap-prod"  
// Update the Config Map which contains the users for the mlbparks application  
sh "oc delete configmap ${destApp}-config -n lpierson-parkmap-prod"  
sh "oc create configmap ${destApp}-config  
--from-file=./configuration/application-users.properties  
--from-file=./configuration/application-roles.properties -n lpierson-parkmap-prod"  
  
// Deploy the inactive application.  
openshiftDeploy depCfg: destApp, namespace: 'lpierson-parkmap-prod', verbose:  
'false', waitTime: '', waitUnit: 'sec'  
openshiftVerifyDeployment depCfg: destApp, namespace: 'lpierson-parkmap-prod',  
replicaCount: '1', verbose: 'false', verifyReplicaCount: 'true', waitTime: '',  
waitUnit: 'sec'  
openshiftVerifyService namespace: 'lpierson-parkmap-prod', svcName: destApp,  
verbose: 'false'
```

We wait for approval to execute the final go-live switch (patching the route to use the new in production service)

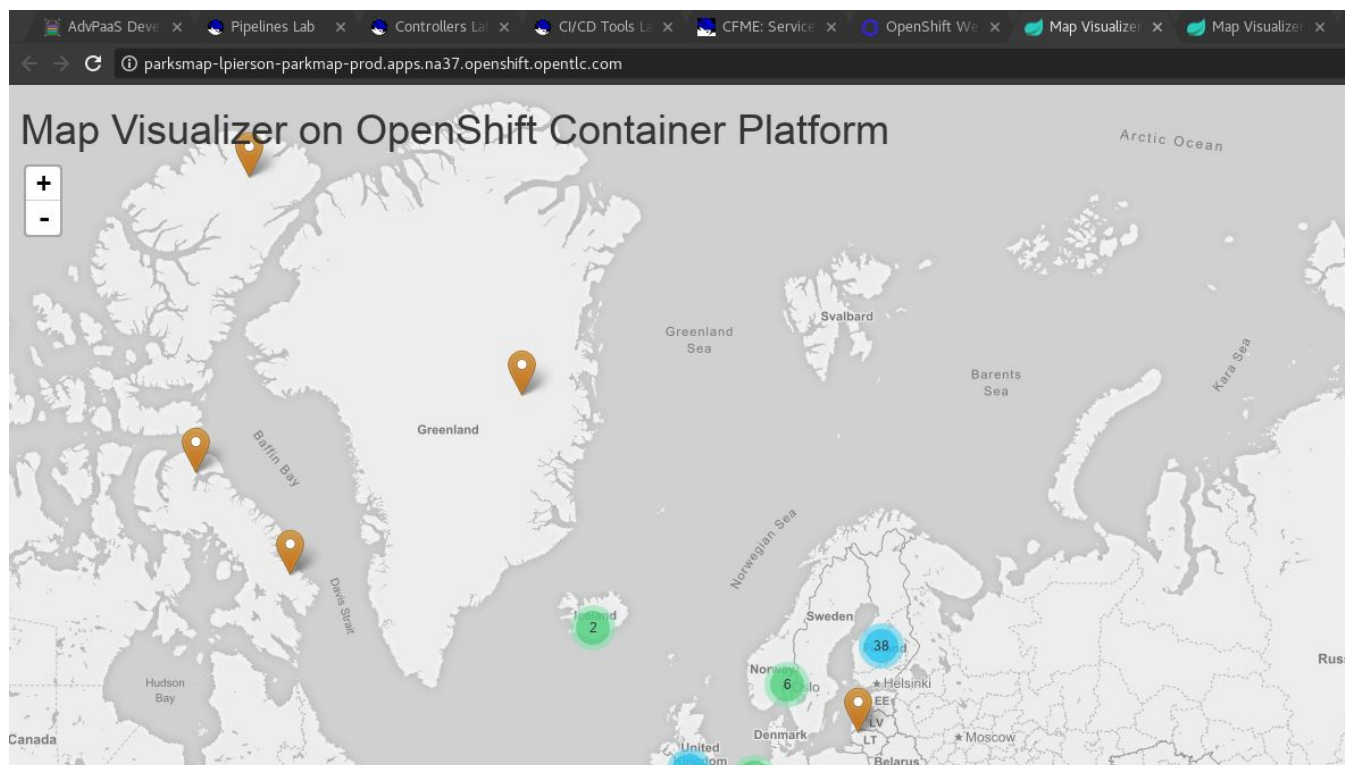
```
input message: "Approve ${destApp} to GO LIVE ?"  
sh 'oc patch route mlbparks -n lpierson-parkmap-prod -p \{"spec":{"to":{"name":"' +  
destApp + '"}}}\'
```

As a Best practice for further usage by MitziCom

we suggest to propose triggers for automatic deployment on image change, instead of creating a deploymentConfig in the pipeline.

APPENDIX 4

For review by MitziCom, all the descriptions of the projects are exported and placed into the directory "homeworks" (<https://github.com/lucpierson/ParksMap/tree/master/homeworks>),
see APPENDIX 5 for detailed export command used



CHALLENGES, RESOLUTIONS, AND RECOMMENDATIONS

Various best practices and alternative configuration (no use of direct github for such project, etc...) can be underlined and will be covered during PoC presentation

Some alternative were also proposed in this document

RELEVANT RED HAT DOCUMENTATION

This section contains links to online Red Hat documentation that is relevant to this engagement.

{Include relevant online documentation that may be helpful to the client

Example:

- Red Hat OpenShift Container Platform Documentation:
https://access.redhat.com/documentation/en-us/openshift_container_platform/?version=3.7

RELEVANT RED HAT TRAINING

This section contains links to Red Hat trainings that are relevant to this engagement.

{Include relevant Red Hat training that may be helpful to the client

Example:

- Red Hat OpenShift Administration (DO280):
- <https://www.redhat.com/en/services/training/do280-red-hat-openshift-administration-i>

APPENDIX 1

Scrapbook

oc login -u lpierson-redhat.com --server https://master.na37.openshift.opentlc.com

OpenTlc Management : <https://labs.opentlc.com/>
OpenShift shared cluster : <https://master.na37.openshift.opentlc.com>
Client machine : [ssh lpierson-redhat.com@ocplab-a13c.oslab.opentlc.com](ssh:lpierson-redhat.com@ocplab-a13c.oslab.opentlc.com)
Connect to OpenShift from client : [oc login -u lpierson-redhat.com --server https://master.na37.openshift.opentlc.com](https://master.na37.openshift.opentlc.com)

Original project from MitziCom : <https://github.com/wkulhanek/ParksMap>
Replicated project for PoC : <https://github.com/lucpierson/ParksMap>

APPLICATION TESTING

Testing the presence

DEV

<http://mlbparks-lpierson-parkmap-dev.apps.na37.openshift.opentlc.com/ws/healthz>
<http://nationalparks-lpierson-parkmap-dev.apps.na37.openshift.opentlc.com/ws/healthz/>

PROD

<http://mlbparks-lpierson-parkmap-prod.apps.na37.openshift.opentlc.com/ws/info>
<http://nationalparks-lpierson-parkmap-prod.apps.na37.openshift.opentlc.com/ws/info/>

Feed with data in Development ==> MongoDB simple DB

Load the records of mlbparks :

<http://mlbparks-lpierson-parkmap-dev.apps.na37.openshift.opentlc.com/ws/data/load>

Load the records of National Parks :

<http://nationalparks-lpierson-parkmap-dev.apps.na37.openshift.opentlc.com/ws/data/load>

Feed with data in Production ==> MongoDBCluster

Load the records of mlbparks :

<http://mlbparks-lpierson-parkmap-prod.apps.na37.openshift.opentlc.com/ws/data/load>

Load the records of National Parks :

<http://nationalparks-lpierson-parkmap-prod.apps.na37.openshift.opentlc.com/ws/data/load>

Main app test

DEV :

<http://parksmap-lpierson-parkmap-dev.apps.na37.openshift.opentlc.com/>
<http://parksmap-lpierson-parkmap-dev.apps.na37.openshift.opentlc.com/ws/backends/list>

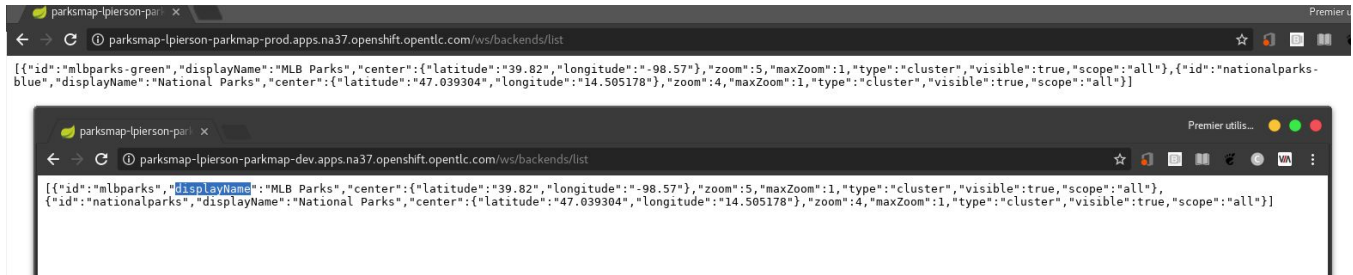
PROD :

<http://parksmap-lpierson-parkmap-prod.apps.na37.openshift.opentlc.com/>

<http://parksmap-lpierson-parkmap-prod.apps.na37.openshift.opentlc.com/ws/backends/list>

```
curl -i
```

```
http://parksmap-lpierson-parkmap-prod.apps.na37.openshift.opentlc.com/ws/backends/list
```



END APPENDIX 1

APPENDIX 2

Configure MongoDB in Dev

```
oc new-app -e \
MONGODB_USER="mongodb",MONGODB_PASSWORD="mongodb",MONGODB_DATABASE="parks",MONGODB_ADMIN_PASSWORD="mongodb_admin_password" \
registry.access.redhat.com/rhsc1/mongodb-32-rhel7:3.2
oc expose svc/mongodb-32-rhel7
```

Configure MongoDB in Prod

If MitziCom decide to follow up on this project with Red Hat, a good practice would be to create an OpenShift Secret that would contains the values for the following keys

- name: MONGODB_DATABASE
- name: MONGODB_USER
- name: MONGODB_PASSWORD
- name: MONGODB_ADMIN_PASSWORD

Names of the components (app, user, passwords) are set accordingly to the application need (<https://github.com/lucpierson/ParksMap/tree/master/mlbparks>)

0) Create Secret (as an example provided, not used by PoC)

```
oc create secret generic mongodb-secret --from-literal=mongodb-username=mongodb
--from-literal=mongodb-password=mongodb--from-literal=mongodb-database=parks
--from-literal=mongodb-admin-password=mongodb_admin_password
```

1) Create the internal headless service that the pods in the StatefulSet use to communicate with each other.

this is done using the container image from Red Hat collections :
registry.access.redhat.com/rhsc1/mongodb-32-rhel7:3.2

```
oc project lpierson-parkmap-prod
echo 'kind: Service
apiVersion: v1
metadata:
  name: "mongodb-internal"
  labels:
    name: "mongodb"
  annotations:
    service.alpha.kubernetes.io/tolerate-unready-endpoints: "true"
spec:
  clusterIP: None
  ports:
```

```
- name: mongodb
port: 27017
selector:
  name: "mongodb" | oc create -f -
```

2) Create the regular service that database clients use to connect to the database.

```
echo 'kind: Service
apiVersion: v1
metadata:
  name: "mongodb"
  labels:
    name: "mongodb"
spec:
  ports:
    - name: mongodb
      port: 27017
  selector:
    name: "mongodb" | oc create -f -
```

3) Create StatefulSet for MongoDB Database

```
echo 'kind: StatefulSet
apiVersion: apps/v1beta1
metadata:
  name: "mongodb"
spec:
  serviceName: "mongodb-internal"
  replicas: 3
  template:
    metadata:
      labels:
        name: "mongodb"
    spec:
      containers:
        - name: mongo-container
          image: "registry.access.redhat.com/rhsc1/mongodb-32-rhel7:3.2"
          ports:
            - containerPort: 27017
          args:
            - "run-mongod-replication"
          volumeMounts:
            - name: mongo-data
              mountPath: "/var/lib/mongodb/data"
          env:
            - name: MONGODB_DATABASE
              value: "parks"
            - name: MONGODB_USER
              value: "mongodb"
            - name: MONGODB_PASSWORD
              value: "mongodb"
```

```
- name: MONGODB_ADMIN_PASSWORD
value: "mongodb_admin_password"
- name: MONGODB_REPLICA_NAME
value: "rs0"
- name: MONGODB_KEYFILE_VALUE
value: "12345678901234567890"
- name: MONGODB_SERVICE_NAME
value: "mongodb"
readinessProbe:
  exec:
  command:
  - stat
  - /tmp/initialized
volumeClaimTemplates:
  - metadata:
    name: mongo-data
    labels:
    name: "mongodb"
    spec:
    accessModes: [ ReadWriteOnce ]
    resources:
    requests:
      storage: "4Gi" | oc create -f -
```

4) Scale the StatefulSet (in case needed)

```
oc get pvc
oc scale statefulset mongodb --replicas=XXX
```

NOTE : for applications in dev to use this MongoDB app, use the syntax -e
MONGO_URL="mongodb://mongodb:mongodb@mongodb:27017/parks?replicaSet=rs0"

END APPENDIX 2

APPENDIX 3

results of sonarQube for Parkmap app

```
$mvn sonar:sonar -s nexus_settings_openshift.xml -Dsonar.host.url=http://$(oc get
route sonarqube -n lpierson-sonarqube --template='{{ .spec.host }}')
[INFO] Scanning for projects...
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/xml-maven-plugin/1.0/xml-maven-plugin-1.0.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/xml-maven-plugin/1.0/xml-maven-plugin-1.0.pom (8 KB at 10.7 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/mojo-parent/28/mojo-parent-28.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/mojo-parent/28/mojo-parent-28.pom (26 KB at 54.9 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/xml-maven-plugin/1.0/xml-maven-plugin-1.0.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/xml-maven-plugin/1.0/xml-maven-plugin-1.0.jar (34 KB at 104.4 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/build-helper-maven-plugin/1.10/build-helper-maven-plugin-1.10.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/build-helper-maven-plugin/1.10/build-helper-maven-plugin-1.10.pom (6 KB at 17.7 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/build-helper-maven-plugin/1.10/build-helper-maven-plugin-1.10.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/build-helper-maven-plugin/1.10/build-helper-maven-plugin-1.10.jar (50 KB at 145.4 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/m
aven/plugins/maven-site-plugin/3.5.1/maven-site-plugin-3.5.1.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/m
aven/plugins/maven-site-plugin/3.5.1/maven-site-plugin-3.5.1.pom (18 KB at 56.2 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/m
aven/plugins/maven-site-plugin/3.5.1/maven-site-plugin-3.5.1.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/m
aven/plugins/maven-site-plugin/3.5.1/maven-site-plugin-3.5.1.jar (129 KB at 280.0 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/jooq/joo
q-codegen-maven/3.8.4/jooq-codegen-maven-3.8.4.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/jooq/joo
q-codegen-maven/3.8.4/jooq-codegen-maven-3.8.4.pom (3 KB at 9.5 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/jooq/joo
q-parent/3.8.4/jooq-parent-3.8.4.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/jooq/joo
q-parent/3.8.4/jooq-parent-3.8.4.pom (10 KB at 30.4 KB/sec)
Downloading:
```

```
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/jooq/jooq-codegen-maven/3.8.4/jooq-codegen-maven-3.8.4.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/jooq/jooq-codegen-maven/3.8.4/jooq-codegen-maven-3.8.4.jar (15 KB at 45.9 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-assembly-plugin/2.6/maven-assembly-plugin-2.6.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-assembly-plugin/2.6/maven-assembly-plugin-2.6.pom (17 KB at 50.6 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-assembly-plugin/2.6/maven-assembly-plugin-2.6.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-assembly-plugin/2.6/maven-assembly-plugin-2.6.jar (240 KB at 517.1 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-dependency-plugin/2.10/maven-dependency-plugin-2.10.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-dependency-plugin/2.10/maven-dependency-plugin-2.10.pom (12 KB at 36.6 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-dependency-plugin/2.10/maven-dependency-plugin-2.10.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-dependency-plugin/2.10/maven-dependency-plugin-2.10.jar (157 KB at 477.6 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-failsafe-plugin/2.18.1/maven-failsafe-plugin-2.18.1.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-failsafe-plugin/2.18.1/maven-failsafe-plugin-2.18.1.pom (9 KB at 27.8 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-failsafe-plugin/2.18.1/maven-failsafe-plugin-2.18.1.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-failsafe-plugin/2.18.1/maven-failsafe-plugin-2.18.1.jar (81 KB at 246.8 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-javadoc-plugin/2.10.4/maven-javadoc-plugin-2.10.4.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-javadoc-plugin/2.10.4/maven-javadoc-plugin-2.10.4.pom (16 KB at 49.8 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-javadoc-plugin/2.10.4/maven-javadoc-plugin-2.10.4.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-javadoc-plugin/2.10.4/maven-javadoc-plugin-2.10.4.jar (406 KB at 884.7 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-source-plugin/2.4/maven-source-plugin-2.4.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-source-plugin/2.4/maven-source-plugin-2.4.pom (7 KB at 20.6 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/maven/plugins/maven-source-plugin/2.4/maven-source-plugin-2.4.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/apache/m
```



```
aven/plugins/maven-source-plugin/2.4/maven-source-plugin-2.4.jar (31 KB at 98.4 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/versions-maven-plugin/2.2/versions-maven-plugin-2.2.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/versions-maven-plugin/2.2/versions-maven-plugin-2.2.pom (16 KB at 49.3 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/mojo-parent/34/mojo-parent-34.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/mojo-parent/34/mojo-parent-34.pom (24 KB at 77.9 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/codehaus-parent/4/codehaus-parent-4.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/codehaus-parent/4/codehaus-parent-4.pom (5 KB at 15.3 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/versions-maven-plugin/2.2/versions-maven-plugin-2.2.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/codehaus
/mojo/versions-maven-plugin/2.2/versions-maven-plugin-2.2.jar (251 KB at 709.0 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/pl/project13
/maven/git-commit-id-plugin/2.1.11/git-commit-id-plugin-2.1.11.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/pl/project13
/maven/git-commit-id-plugin/2.1.11/git-commit-id-plugin-2.1.11.pom (10 KB at 8.8 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/sonatype
/oss/oss-parent/6/oss-parent-6.pom
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/org/sonatype
/oss/oss-parent/6/oss-parent-6.pom (5 KB at 4.7 KB/sec)
Downloading:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/pl/project13
/maven/git-commit-id-plugin/2.1.11/git-commit-id-plugin-2.1.11.jar
Downloaded:
http://nexus3-lpierson-nexus.apps.na37.openshift.opentlc.com/repository/maven-all-public/pl/project13
/maven/git-commit-id-plugin/2.1.11/git-commit-id-plugin-2.1.11.jar (61 KB at 89.1 KB/sec)
[INFO]
[INFO] -----
[INFO] Building parksmap 1.0
[INFO] -----
[INFO]
[INFO] --- sonar-maven-plugin:3.4.0.905:sonar (default-cli) @ parksmap ---
[INFO] User cache: /home/lpierson-redhat.com/.sonar/cache
[INFO] SonarQube version: 6.7.2
[INFO] Default locale: "en_US", source code encoding: "UTF-8"
[INFO] Publish mode
[INFO] Load global settings
[INFO] Load global settings (done) | time=523ms
[INFO] Server id: AWI9VdALmYtTbDkqpq3Y
[INFO] User cache: /home/lpierson-redhat.com/.sonar/cache
[INFO] Load plugins index
[INFO] Load plugins index (done) | time=267ms
[INFO] Process project properties
[INFO] Load project repositories
[INFO] Load project repositories (done) | time=464ms
[INFO] Load quality profiles
[INFO] Load quality profiles (done) | time=211ms
```

```
[INFO] Load active rules
[INFO] Load active rules (done) | time=2637ms
[INFO] Load metrics repository
[INFO] Load metrics repository (done) | time=187ms
[INFO] Project key: com.openshift.evg.roadshow:parksmap
[INFO] ----- Scan parksmap
[INFO] Load server rules
[INFO] Load server rules (done) | time=467ms
[INFO] Base dir: /home/lpierson-redhat.com/ParksMap/parksmap
[INFO] Working dir: /home/lpierson-redhat.com/ParksMap/parksmap/target/sonar
[INFO] Source paths: pom.xml, src/main/java
[INFO] Source encoding: UTF-8, default locale: en_US
[INFO] Index files
[INFO] 18 files indexed
[INFO] Quality profile for java: Sonar way
[INFO] Quality profile for xml: Sonar way
[INFO] Sensor JavaSquidSensor [java]
[INFO] Configured Java source version (sonar.java.source): 6
[INFO] JavaClasspath initialization
[INFO] JavaClasspath initialization (done) | time=27ms
[INFO] JavaTestClasspath initialization
[INFO] JavaTestClasspath initialization (done) | time=18ms
[INFO] Java Main Files AST scan
[INFO] 17 source files to be analyzed
[INFO] 17/17 source files have been analyzed
[INFO] Java Main Files AST scan (done) | time=5930ms
[INFO] Java Test Files AST scan
[INFO] 0 source files to be analyzed
[INFO] Java Test Files AST scan (done) | time=3ms
[INFO] Sensor JavaSquidSensor [java] (done) | time=7644ms
[INFO] Sensor SurefireSensor [java]
[INFO] 0/0 source files have been analyzed
[INFO] parsing [/home/lpierson-redhat.com/ParksMap/parksmap/target/surefire-reports]
[INFO] Sensor SurefireSensor [java] (done) | time=17ms
[INFO] Sensor JaCoCoSensor [java]
[INFO] Sensor JaCoCoSensor [java] (done) | time=1ms
[INFO] Sensor SonarJavaXmlFileSensor [java]
[INFO] 1 source files to be analyzed
[INFO] Sensor SonarJavaXmlFileSensor [java] (done) | time=2390ms
[INFO] Sensor XML Sensor [xml]
[INFO] 1/1 source files have been analyzed
[INFO] Sensor XML Sensor [xml] (done) | time=358ms
[INFO] Sensor Analyzer for "php.ini" files [php]
[INFO] Sensor Analyzer for "php.ini" files [php] (done) | time=5ms
[INFO] Sensor Zero Coverage Sensor
[INFO] Sensor Zero Coverage Sensor (done) | time=134ms
[INFO] Sensor CPD Block Indexer
[INFO] Sensor CPD Block Indexer (done) | time=184ms
[INFO] SCM provider for this project is: git
[INFO] 1 files to be analyzed
[INFO] 1/1 files analyzed
[INFO] 5 files had no CPD blocks
[INFO] Calculating CPD for 12 files
[INFO] CPD calculation finished
[INFO] Analysis report generated in 388ms, dir size=129 KB
[INFO] Analysis reports compressed in 146ms, zip size=60 KB
[INFO] Analysis report uploaded in 483ms
[INFO] ANALYSIS SUCCESSFUL, you can browse
http://sonarqube-lpierson-sonarqube.apps.na37.openshift.opentlc.com/dashboard/index/com.openshift.evg.roadshow:parksmap
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at
http://sonarqube-lpierson-sonarqube.apps.na37.openshift.opentlc.com/api/ce/task?id=AWJhM2yGU9sJ-rQrVt
```

7H

```
[INFO] Task total time: 22.736 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 46.656s
[INFO] Finished at: Mon Mar 26 03:26:22 EDT 2018
[INFO] Final Memory: 31M/145M
[INFO] -----
```

END APPENDIX 3

APPENDIX 4

OpenShift build configuration with a Pipeline build strategy.

PIPELINE IN OPENSIFT FOR APPLICATION MLB PARKS

```
echo "apiVersion: v1
items:
- kind: "BuildConfig"
  apiVersion: "v1"
  metadata:
    name: "mlbparks-pipeline"
  spec:
    source:
      type: "Git"
      git:
        uri: "https://github.com/lucpierson/ParksMap"
        strategy:
          type: "JenkinsPipeline"
          jenkinsPipelineStrategy:
            jenkinsfilePath: "mlbparks/Jenkinsfile"
kind: List
metadata: []" | oc create -f - -n lpierson-jenkins
```

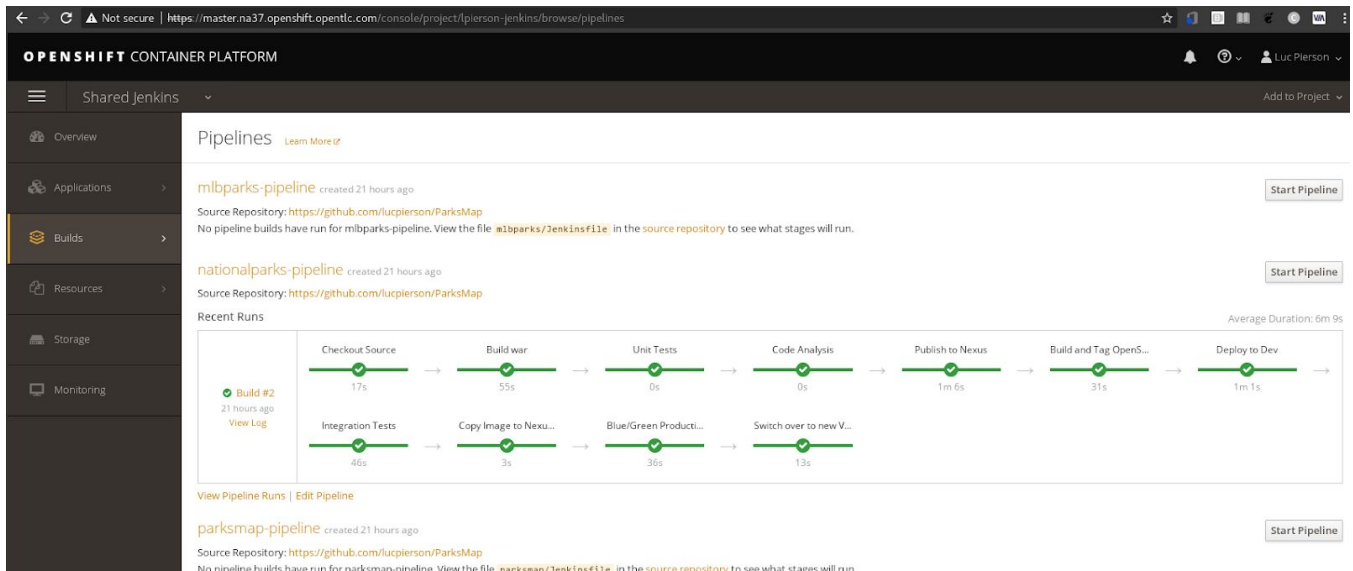
PIPELINE IN OPENSIFT FOR APPLICATION NATIONAL PARKS

```
echo "apiVersion: v1
items:
- kind: "BuildConfig"
  apiVersion: "v1"
  metadata:
    name: "nationalparks-pipeline"
  spec:
    source:
      type: "Git"
      git:
        uri: "https://github.com/lucpierson/ParksMap"
        strategy:
          type: "JenkinsPipeline"
          jenkinsPipelineStrategy:
            jenkinsfilePath: "nationalparks/Jenkinsfile"
kind: List
metadata: []" | oc create -f - -n lpierson-jenkins
```

PIPELINE IN OPENSIFT FOR APPLICATION PARKSMAP

```
echo "apiVersion: v1
```

```
items:
- kind: "BuildConfig"
  apiVersion: "v1"
  metadata:
    name: "parksmap-pipeline"
  spec:
    source:
      type: "Git"
      git:
        uri: "https://github.com/lucpierson/ParksMap"
        strategy:
          type: "JenkinsPipeline"
          jenkinsPipelineStrategy:
            jenkinsfilePath: "parksmap/Jenkinsfile"
kind: List
metadata: []" | oc create -f - -n lpierson-jenkins
```



OPENSHIFT CONTAINER PLATFORM

Shared Jenkins

Pipelines [Learn More](#)

mlbparks-pipeline created 21 hours ago [Start Pipeline](#)

Source Repository: <https://github.com/lucpierson/ParksMap>

No pipeline builds have run for mlbparks-pipeline. View the file [mlbparks/Jenkinsfile](#) in the source repository to see what stages will run.

nationalparks-pipeline created 21 hours ago [Start Pipeline](#)

Source Repository: <https://github.com/lucpierson/ParksMap>

Recent Runs

Build #2 21 hours ago [View Log](#)

Average Duration: 6m 9s

Stage	Duration
Checkout Source	17s
Build war	55s
Unit Tests	0s
Code Analysis	0s
Publish to Nexus	1m 6s
Build and Tag OpenS...	31s
Deploy to Dev	1m 1s
Integration Tests	46s
Copy Image to Nexu...	3s
Blue/Green Producti...	36s
Switch over to new V...	13s

[View Pipeline Runs](#) | [Edit Pipeline](#)

parksmap-pipeline created 21 hours ago [Start Pipeline](#)

Source Repository: <https://github.com/lucpierson/ParksMap>

No pipeline builds have run for parksmap-pipeline. View the file [parksmap/Jenkinsfile](#) in the source repository to see what stages will run.

END APPENDIX 4

APPENDIX 5

Configuration Saved in project/homeworks Directory, in
<https://github.com/lucpierson/ParksMap>

Export command used

```
mkdir homeworks  
cd homeworks
```

#Export all OpenShift Artefacts for the projects of PoC

```
oc get all -n lpierson-jenkins > lpierson-jenkins.getAll.txt  
oc get all -n lpierson-nexus > lpierson-.getAll.txt  
oc get all -n lpierson-sonarqube > lpierson-sonarqube.getAll.txt  
oc get all -n lpierson-parkmap-dev > lpierson-parkmap-dev.getAll.txt  
oc get all -n lpierson-parkmap-prod > lpierson-parkmap-prod.getAll.txt
```

#Export all OpenShift Artefacts as Templates for the projects of PoC

```
oc export all --as-template jenkins -o yaml -n lpierson-jenkins  
>lpierson-jenkins.template.txt  
oc export all --as-template nexus -o yaml -n lpierson-nexus  
>lpierson-nexus.template.txt  
oc export all --as-template sonarqube -o yaml -n  
lpierson-sonarqube>lpierson-sonarqube.template.txt  
oc export all --as-template parkmap-dev -o yaml -n  
lpierson-parkmap-dev>lpierson-parkmap-dev.template.txt  
oc export all --as-template parkmap-prod -o yaml -n  
lpierson-parkmap-prod>lpierson-parkmap-prod.template.txt
```

END APPENDIX 5

APPENDIX A: ENGAGING RED HAT GLOBAL SUPPORT SERVICES

More information on how to engage support can be found at:

<http://www.redhat.com/support>

In some cases, it has been found that your questions or challenges may have been seen already. In this scenario, either Red Hat or a Client teammate has documented the challenge and resolution. To expedite your search on areas specific to Red Hat based solutions, please visit Red Hat's Knowledgebase:

<http://kbase.redhat.com/faq/en>

If you are unable to find what you are looking for, please feel free to engage Red Hat's Global Support Services (GSS).

Valid subscriptions include access to GSS. To understand the level of service your subscription includes, please visit:

<https://www.redhat.com/support/policy/sla/production/>

To expedite your support request, it would be helpful to have the following prior to engaging GSS:

- Define the Problem. Please include use case/scenario. For example, "The server is used for batch processing and run at desired rate up through # of jobs. After that, server performance reduces by X."
- Gather Background Information. For example, the application stack, drivers, of the server or workstation itself along with servers or peripherals that interact with the server in question.
- Gather Relevant diagnostic information.
- Determine the severity level (1-4). For severity level definitions, please visit https://www.redhat.com/support/policy/GSS_severity.html
- Ensure you have an RHN Login or your account number.

When you are ready to contact GSS, there are a few ways for you to do so – based on your severity.

Support requests may be initiated and updated at anytime. For support that is included with your valid subscription, our Production Support team is available via web (<https://www.redhat.com/wapps/support/>) or phone (<https://www.redhat.com/wapps/support/>).

If you have a Technical Account Manager (TAM), please initiate/update support via the dedicated Issue Tracker queue or call your TAM directly.

For Severity 1, it is highly recommended to call us immediately after you have initiated a ticket.

A GSS Quick Guide (pdf) may be found at:

<https://www.redhat.com/support/process/production/#howto>