



# Customer Talk

Regain control of your  
delivery process with  
OpenShift

**stpg**

#

04.02.2021

Charles Gauchon – Pascal Romanens

# Who we are ?



- Charles Gauchon
  - Head of IT Infrastructure & Operations @TPG
  - Focused on SLA and IT process optimization
- 
- Pascal Romanens - IT Architect @TPG
  - Main contributor for the implementation of OpenShift @TPG

 [www.linkedin.com/in/pascal-romanens](https://www.linkedin.com/in/pascal-romanens)



# What we do ?

- Transports Publics Genevois
- Our mission is to transport people
- 600'000 passengers / day
- ~50% electric vehicles
- Trams, buses, trolley buses
- 1 line with autonomous vehicle



- 13 IT engineers – 5 network – 6 systems – 2 workstations
- 700 VM on vSphere classic Blade/SAN configuration
- 580 Windows 120 Linux
- RFP constraints (public tenders) lead to heterogeneous IT landscape
- A lot of projects
- A lot of IT Services
- A lot of vendors with different technologies / procedures

- Linux footprint grew up from 20 to 120 VM during the last 3 years
- The trend is a fast paced growing «Linux market share» vs Microsoft
- Several Linux distributions : Debian, Red Hat, Suse
- TPG Linux environment much less mature than Microsoft's
- 1 Linux engineer from august 2020

# 2 converging points of vue

- IT Operations & Infrastructure
- Delivery process

- Security patching (more and more audits)
- OS lifecycle management
- Lack of skills for Linux OS provisionning and configuration
- Mainly Microsoft Systems skills so how to improve Linux skills in an effective manner ?

- Reduce Linux OS footprint and improve standardization by replacing the VM with containers
- Simplify patching and lifecycle management
- Improve security
- Reduce maintenance windows impact
- Reduce human/skills dependancies in IT operations
- Improve knowledge sharing
- Invest in Automation technologies
- Focus our training efforts on 1 main hosting technology after VMware



- The dream of every IT manager is

«One Click Upgrade»

# Delivery process context

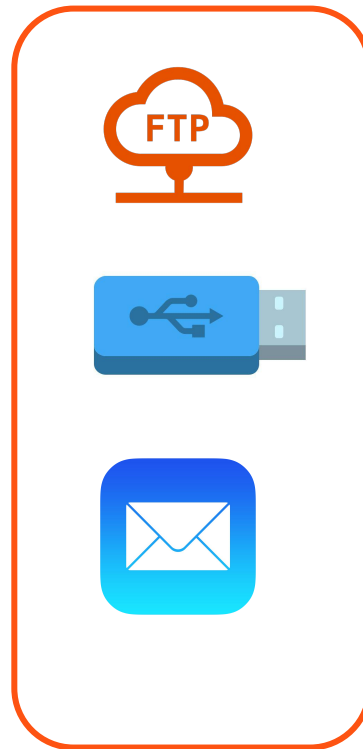
- What we had



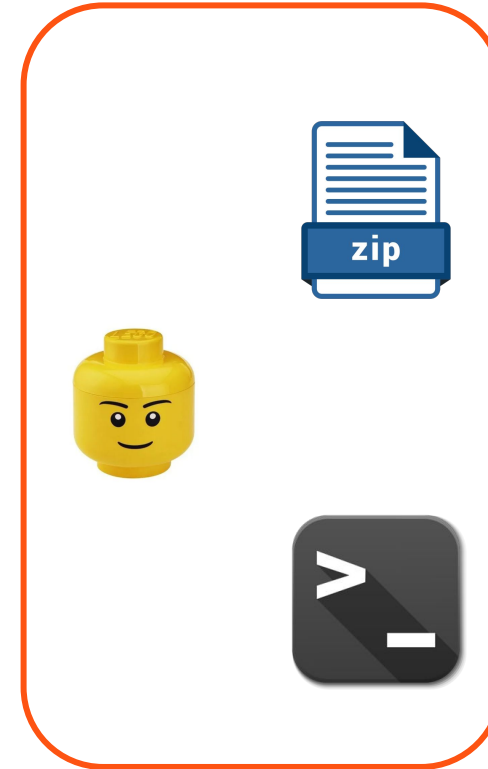
Our stacks



Our runtimes



Our delivery  
process



Our deployment  
process



Our fallback  
process

# Delivery process context



- No version control, no easy rollback
- Same processes repeated again and again, **manually**
  - Error prone, time consuming, costly (lots of overtime)
- Too many stacks, no integration teams @TPG, lack of skills had us delegate deployment activities to our suppliers with **privileged accounts**
- This led to a loss of control of our delivery process



KEEP  
CALM  
AND REGAIN  
CONTROL

# 1 - Regain control of your source code



- Implement Git as the main entry point of your delivery process
  - « *If it's not in Git it doesn't get deployed, period* »
- Use every features of Git to support your delivery lifecycle
  - Define a branching model
  - Use version tagging
  - Use pull requests as code moves toward production
  - Implement merge requests reviews



# 2 - Regain control of your runtimes

- It is **your responsibility** to provide runtimes to your supplier not the other way around.
- Create your own base container images :
  - OS
  - Runtimes
  - Builders
  - Tools
- Follow the editor's versioning (ie we only provide LTS versions)
- Tune them according to your needs and infrastructure
- Patch them and control their overall lifecycle
- **Forbid usage of public registries like Docker Hub**

# 2 - Regain control of your runtimes



- Implement a registry that will support your use case.
- At TPG we use RedHat Quay
  - Our Quay registry is publicly available so our suppliers have access to all our base images
  - It implements RBAC with appropriate permissions on private/public images
  - It scans all of our images for known CVEs
  - It integrates with Openshift and let us know which vulnerabilities made it to our environments

## T tpg\_images-factory

+ Create New Repository



## Repositories



1 - 33 of 33



Filter Repositories...

REPOSITORY NAME

LAST MODIFIED

ACTIVITY ↓

STAR

T tpg\_images-factory / [tpg-node](#)

Today at 1:15 AM

T tpg\_images-factory / [tpg-alpine](#)

Yesterday at 11:00 AM

T tpg\_images-factory / [tpg-tomcat](#)

Yesterday at 10:37 PM

T tpg\_images-factory / [tpg-openjdk](#)

Yesterday at 10:08 PM

T tpg\_images-factory / [tpg-debian](#)

01/21/2021

T tpg\_images-factory / [tpg-openshift-tools](#)

Yesterday at 9:12 PM

T tpg\_images-factory / [tpg-angular-tester](#)

Today at 2:43 AM

T tpg\_images-factory / [tpg-php-fpm](#)

Today at 3:30 PM

T tpg\_images-factory / [tpg-nginx](#)

Yesterday at 10:00 PM

T tpg\_images-factory / [tpg-python](#)

Today at 12:13 AM



← Repositories tpg\_images-factory / tpg-python ☆

Repository Tags

1 - 2 of 2

TAG	LAST MODIFIED ↓	SECURITY SCAN	SIZE	EXPIRES
<input type="checkbox"/> 3-debian	15 hours ago	11 Low	72.3 MB	Never
<input type="checkbox"/> 3-alpine	15 hours ago	Passed	29.3 MB	Never

CVE scans are automatically performed by Clair within Quay

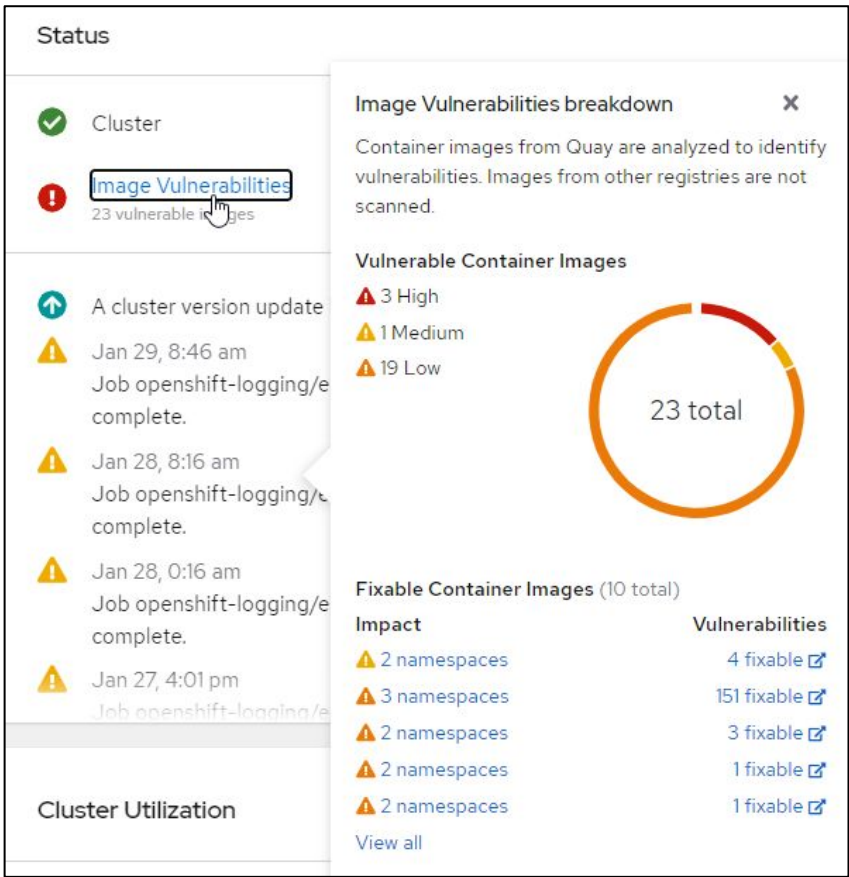
84ac34bb547d

Quay Security Scanner has detected **101** vulnerabilities.

11 Low-level vulnerabilities.  
 56 Negligible-level vulnerabilities.  
 34 Unknown-level vulnerabilities.

**Vulnerabilities** Filter Vulnerabilities... ☐ Only show fixable

CVE	SEVERITY ↓	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN LAYER
▶ CVE-2019-17498 <a href="#">🔗</a>	Low	libssh2	1.8.0-2.1	(None)	<b>RUN</b> set -eux && groupadd -g 10002 -r user && use...
▶ CVE-2019-15847 <a href="#">🔗</a>	Low	gcc-8	8.3.0-6	(None)	<b>ADD</b> file:d2abb0e4e7ac1773741f51f57d3a0b8ffc79073...
▶ CVE-2019-17543 <a href="#">🔗</a>	Low	lz4	1.8.3-1	(None)	<b>ADD</b> file:d2abb0e4e7ac1773741f51f57d3a0b8ffc79073...
▶ CVE-2019-1551 <a href="#">🔗</a>	Low	openssl	1.1.1d-0+deb10u4	(None)	<b>RUN</b> set -eux && groupadd -g 10002 -r user && use...
▶ CVE-2019-14855 <a href="#">🔗</a>	Low	gnupg2	2.2.12-1+deb10u1	(None)	<b>ADD</b> file:d2abb0e4e7ac1773741f51f57d3a0b8ffc79073...
▶ CVE-2020-6096 <a href="#">🔗</a>	Low	glibc	2.28-10	(None)	<b>ADD</b> file:d2abb0e4e7ac1773741f51f57d3a0b8ffc79073...



Quay integrates natively with Openshift and displays which vulnerabilities hit running pods

Project: sds-prism-prod

### Image Manifest Vulnerabilities

Image Name	Namespace	Highest Severity	Affected Pods	Fixable
<b>IMV</b> tpg_images-factory/tpg-mariadb	<b>NS</b> sds-prism-prod	<b>Low</b>	1	1
<b>IMV</b> tpg_sds-prism-prod/tpg-prism-superset	<b>NS</b> sds-prism-prod	<b>Low</b>	2	0
<b>IMV</b> tpg_images-factory/tpg-mariadb	<b>NS</b> sds-prism-prod	<b>Low</b>	1	0
<b>IMV</b> tpg_images-factory/tpg-mariadb	<b>NS</b> sds-prism-prod	<b>Low</b>	1	1
<b>IMV</b> tpg_images-factory/tpg-mariadb	<b>NS</b> sds-prism-prod	<b>Low</b>	1	1
<b>IMV</b> tpg_images-factory/tpg-mariadb	<b>NS</b> sds-prism-prod	<b>Low</b>	1	3
<b>IMV</b> tpg_images-factory/tpg-mariadb	<b>NS</b> sds-prism-prod	<b>Low</b>	1	0
<b>IMV</b> tpg_sds-prism-prod/tpg-prism-airflow	<b>NS</b> sds-prism-prod	<b>Low</b>	3	0



# 3 - Regain control of your environments

- Define a governance that will grant your suppliers with the privileges they need
- Openshift implements a huge number of privileges that you can use to fine tune the roles you grant
- Our approach is simple:
  - Projects can only be created or destroyed by cluster admins (TPG staff)
  - ResourceQuotas and EgressIP are implemented at the project level
  - We create one project per app/env
  - We only grant «**edit**» or «**view**» role on a «per group per environment» basis
  - RBAC is synchronized with our AD where appropriate groups and OU are maintained

## EXPLORER

## &gt; OPEN EDITORS

## v OPENSIFT

## v monitoring

## &gt; network-policy

## &gt; nfs

## &gt; openshift

## &gt; production-install

## v projects

## v providers

## &gt; elazur

## v link

## v b2b

! dev.yaml

! prod.yaml

! qual.yaml

## v portail-communes

! dev.yaml

! prod.yaml

! qual.yaml

## v prism

00-service-account.sh

! dev.yaml

! prod.yaml

! qual.yaml

## v tpg-pay

! dev.yaml

! prod.yaml

! qual.yaml

## &gt; webshop

## &gt; moviplus

## &gt; tpg

! dev.yaml x

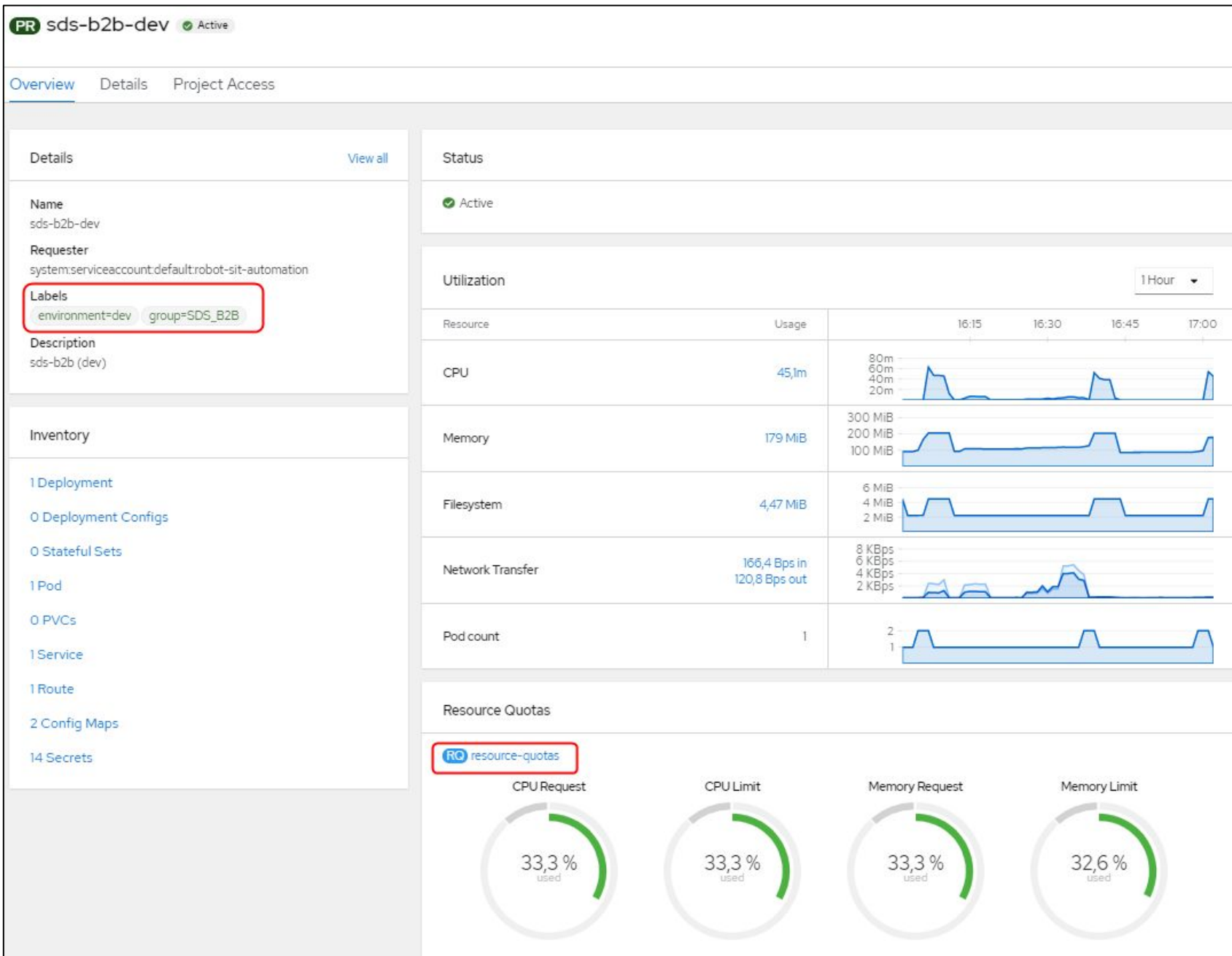
projects &gt; providers &gt; link &gt; b2b &gt; ! dev.yaml

```
1 APP_NAME: sds-b2b
2 ENV: dev
3 AD_GROUP: SDS_B2B
4 # the ip address from where B2B will request external resources (for Firewall ACL)
5 EGRESS_IP: 172.27.140.26
6 # Resource quotas
7 CPU_REQUEST: 150m
8 CPU_LIMIT: 900m
9 MEMORY_REQUEST: 450Mi
10 MEMORY_LIMIT: 1.2Gi
11 PODS_LIMIT: 3
12 PERSISTENT_VOLUME_CLAIMS_LIMIT: 0
```

Openshift projects are automatically created by our infra-as-code Git repository

Theses scripts are executed with the cluster-admin privilege

This repo is only accessible to appropriate TPG staff



NetNamespaces > NetNamespace Details

**NN sds-b2b-dev**

Details YAML

```
1 | apiVersion: network.openshift.io/v1
2 | egressIPs:
3 |   - 172.27.140.26
4 | kind: NetNamespace
5 | metadata:
```

Active Directory group memberships are synced with  
Openshift RBAC

Red Hat

OpenShift

Container Platform

Administrator

Home

Overview

Projects

Search

Explore

Events

Operators

Workloads

PR sds-b2b-dev

Active

Overview

Details

YAML

Workloads

Role Bindings

Create Binding

Filter

Name

Search by name...

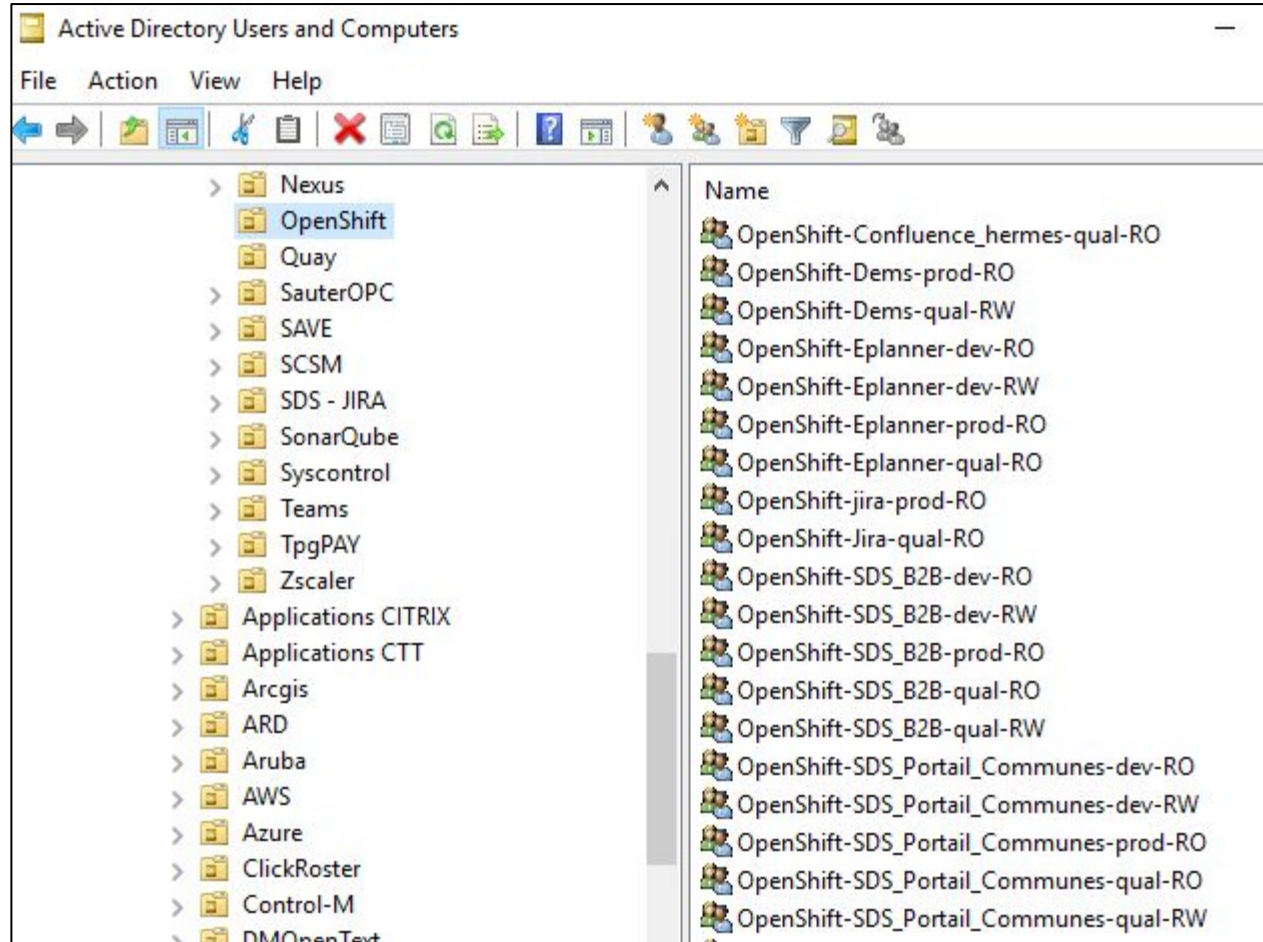
Kind

Namespace Role Bind...

Clear all filters

Name	Role Ref	Subject Kind	Subject Name	Namespace
RB edit	CR edit	Group	OpenShift-SDS_B2B-dev-RW	NS sds-b2b-dev
RB view	CR view	Group	OpenShift-SDS_B2B-dev-RO	NS sds-b2b-dev

## Our Active Directory supports Openshift RBAC



- RO = view mode
  - Access to the project in readonly
  - Topology
  - logs
  - Monitoring metrics etc
- RW = edit mode
  - All the above
  - Can RSH to pods
  - Can edit project's objects like Deployment



# 4 - Regain control of your deployment

- Implement appropriate CI/CD pipelines to automate delivery upon code changes
- Industrialise your deployment pipelines so anyone can deploy by «pushing a button»

The screenshot shows the Bamboo web interface for the 'Openshift - Deploy Webshop' project. The top navigation bar includes links for 'My Bamboo', 'Projects', 'Build', 'Deploy', 'Specs', 'Reports', and a 'Create' button. A search bar is also present. The main content area shows the 'Deployment project summary' for the 'Openshift - Deploy Webshop' project. A 'Deploy' dropdown menu is open, showing options for 'Dev', 'Qual', and 'Prod'. Below this, a warning message states: 'No **shared** artifacts found for related plan. Configure the build plan to publish shared artifacts.' The 'Environments' section is visible at the bottom, showing a table of deployment environments.

Name	Release	Result	Completed
Dev	develop-341	Logs	28 Jan 2021 12:00 PM
Qual	release-143	Logs	28 Jan 2021 12:09 PM
Prod	production-21	Logs	21 Jan 2021 09:58 AM

- The adoption of OpenShift has set the first step of TPG DevOps era with an IT transformation process
- Legacy SysAdmins are evolving to DevOps/Automation Engineer
- Git has become the key point of our «sharing» mindset
- Now we think «automation first» for all our IT operations
- We have reduced our time to market with transparent operations ...

The background of the bottom section is a repeating pattern of large, 3D-style yellow smiley face emojis with black eyes and curved mouths, creating a cheerful and positive atmosphere.

**We have improved the  
pleasure to work**