Customer Talk

Regain control of your delivery process with OpenShift

tpg

#

04.02.2021

Charles Gauchon – Pascal Romanens

Who we are ?

tpg

- 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

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

TPG Context

- 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

TPG Context

- 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

tpg

- IT Operations & Infrastructure
- Delivery process

IT Ops & Infrastructure point of vue

- 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 ?

IT Ops objectives

- 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

IT Ops objectives

- The dream of every IT manager is

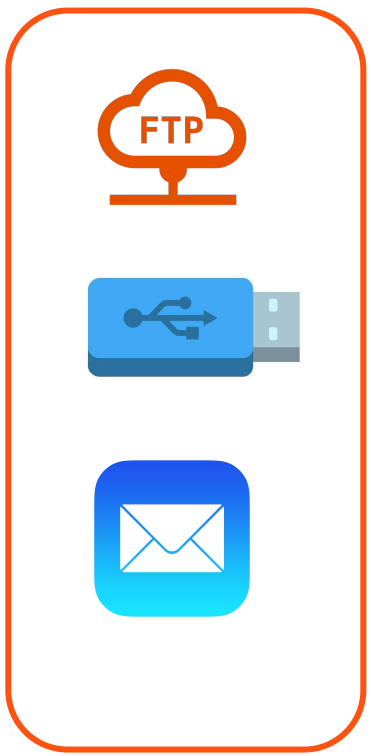
«One Click Upgrade»

Delivery process context

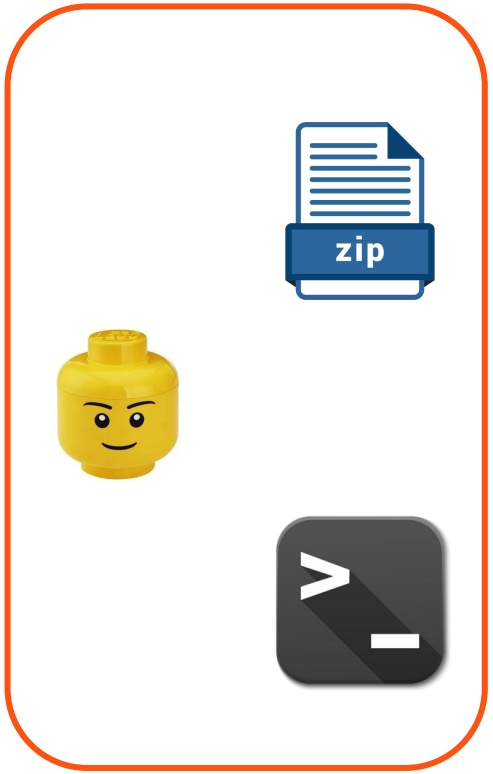
- What we had

Our stacks

Our runtimes



Our delivery process



Our deployment process

Our fallback process

Delivery process context

tpg

- 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

tpg

- 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

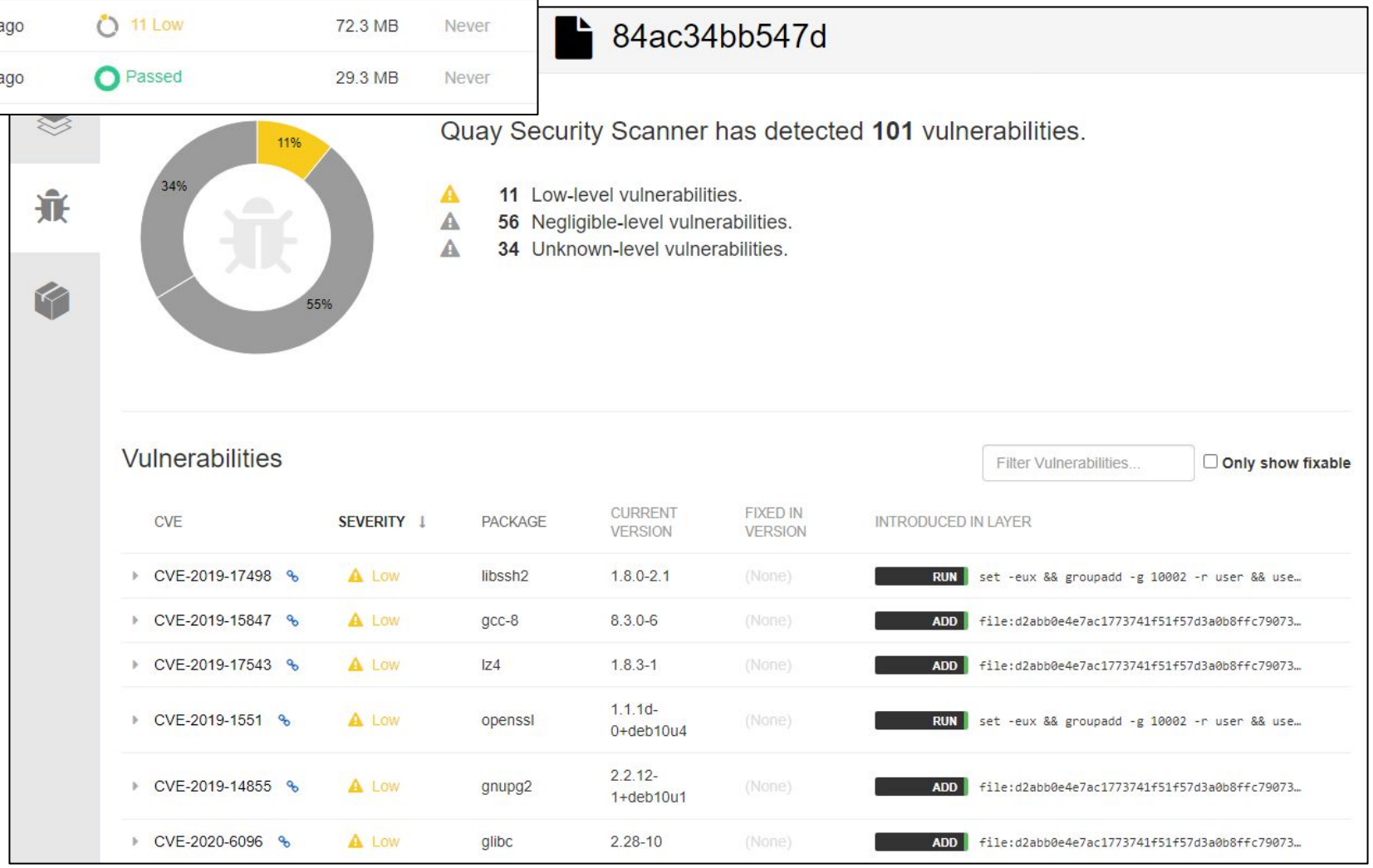
+ Create New Repository

Repositories

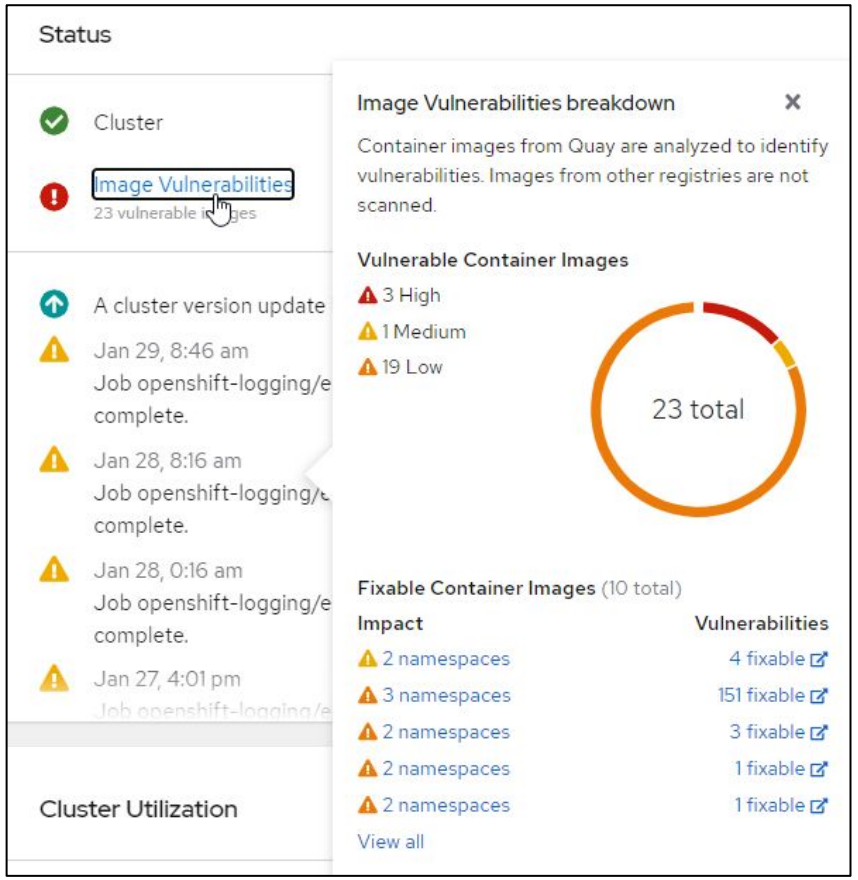
1 - 33 of 33 Filter Repositories...

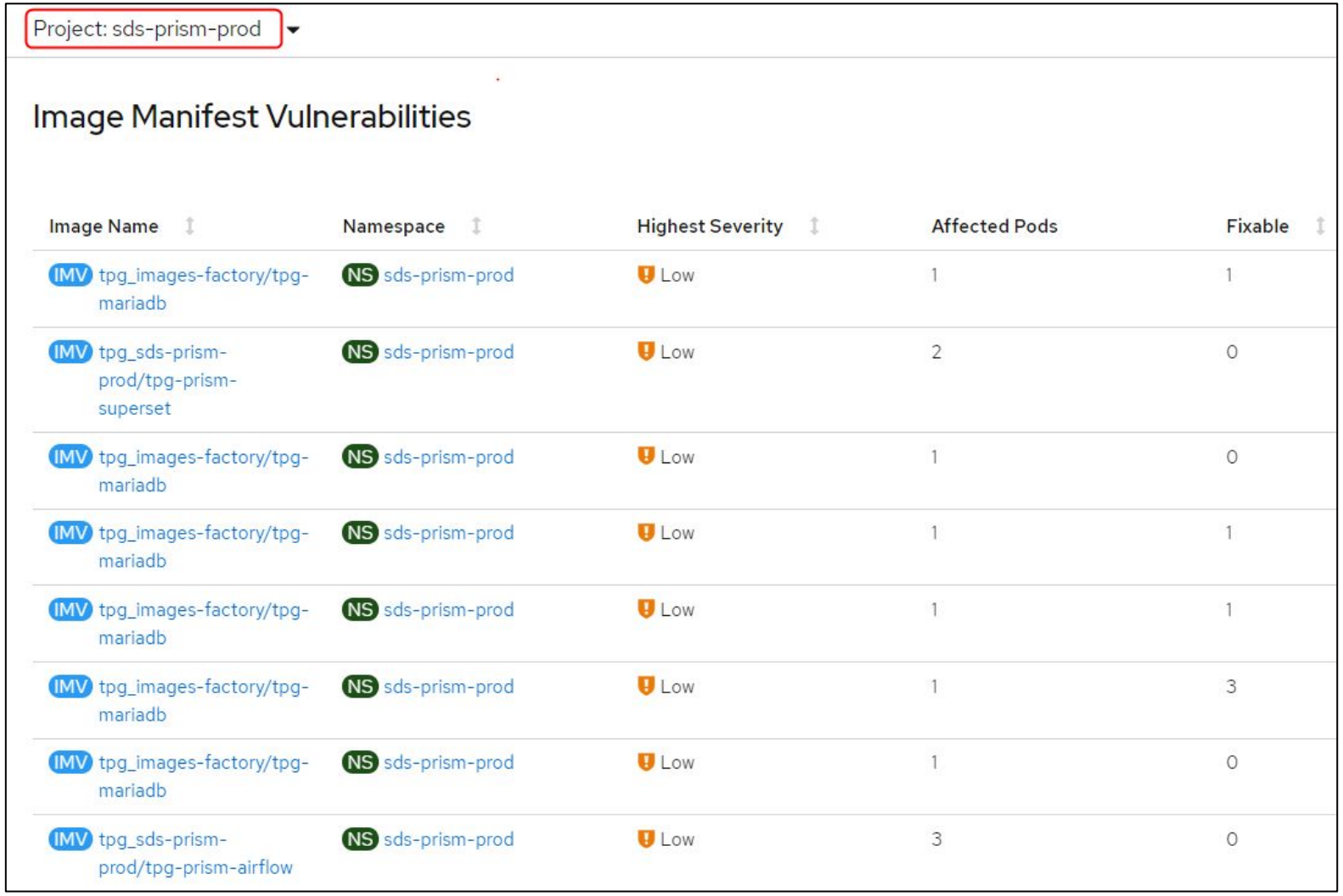
REPOSITORY NAME	LAST MODIFIED	ACTIVITY	STAR
tpg_images-factory / tpg-node	Today at 1:15 AM		
tpg_images-factory / tpg-alpine	Yesterday at 11:00 AM		
tpg_images-factory / tpg-tomcat	Yesterday at 10:37 PM		
tpg_images-factory / tpg-openjdk	Yesterday at 10:08 PM		
tpg_images-factory / tpg-debian	01/21/2021		
tpg_images-factory / tpg-openshift-tools	Yesterday at 9:12 PM		
tpg_images-factory / tpg-angular-tester	Today at 2:43 AM		
tpg_images-factory / tpg-php-fpm	Today at 3:30 PM		
tpg_images-factory / tpg-nginx	Yesterday at 10:00 PM		
tpg_images-factory / tpg-python	Today at 12:13 AM		

CVE scans are automatically performed by Clair within Quay



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





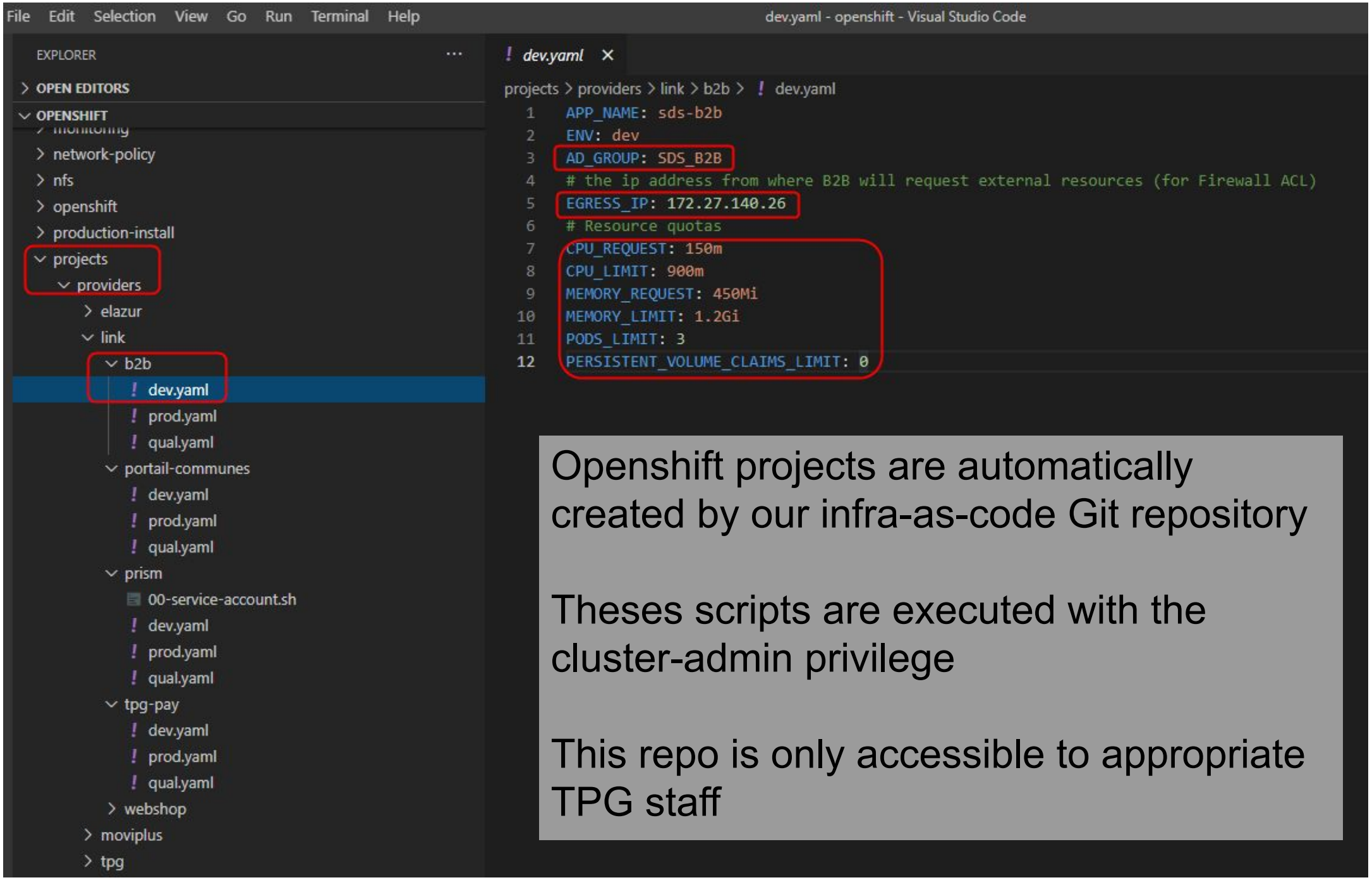
3 - Regain control of your environments

- 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



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

PR sds-b2b-dev Active

Overview Details Project Access

Details

View all

Name
sds-b2b-dev

Requester
system:serviceaccount:default:robot-sit-automation

Labels
environment=dev group=SDS_B2B

Description
sds-b2b (dev)

Inventory

1 Deployment
0 Deployment Configs
0 Stateful Sets
1 Pod
0 PVCs
1 Service
1 Route
2 Config Maps
14 Secrets

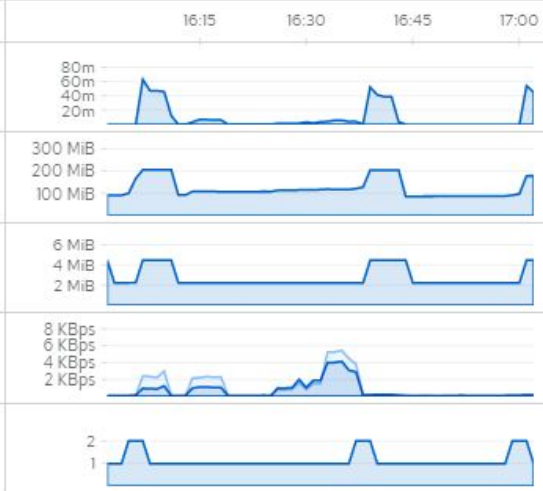
Status

Active

Utilization

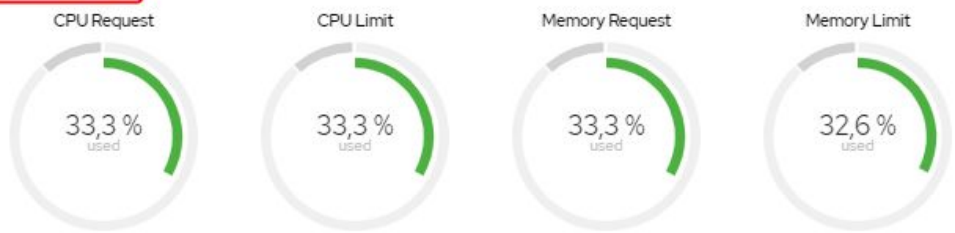
1 Hour

Resource	Usage
CPU	45,1m
Memory	179 MiB
Filesystem	4,47 MiB
Network Transfer	166,4 Bps in 120,8 Bps out
Pod count	1



Resource Quotas

RQ resource-quotas



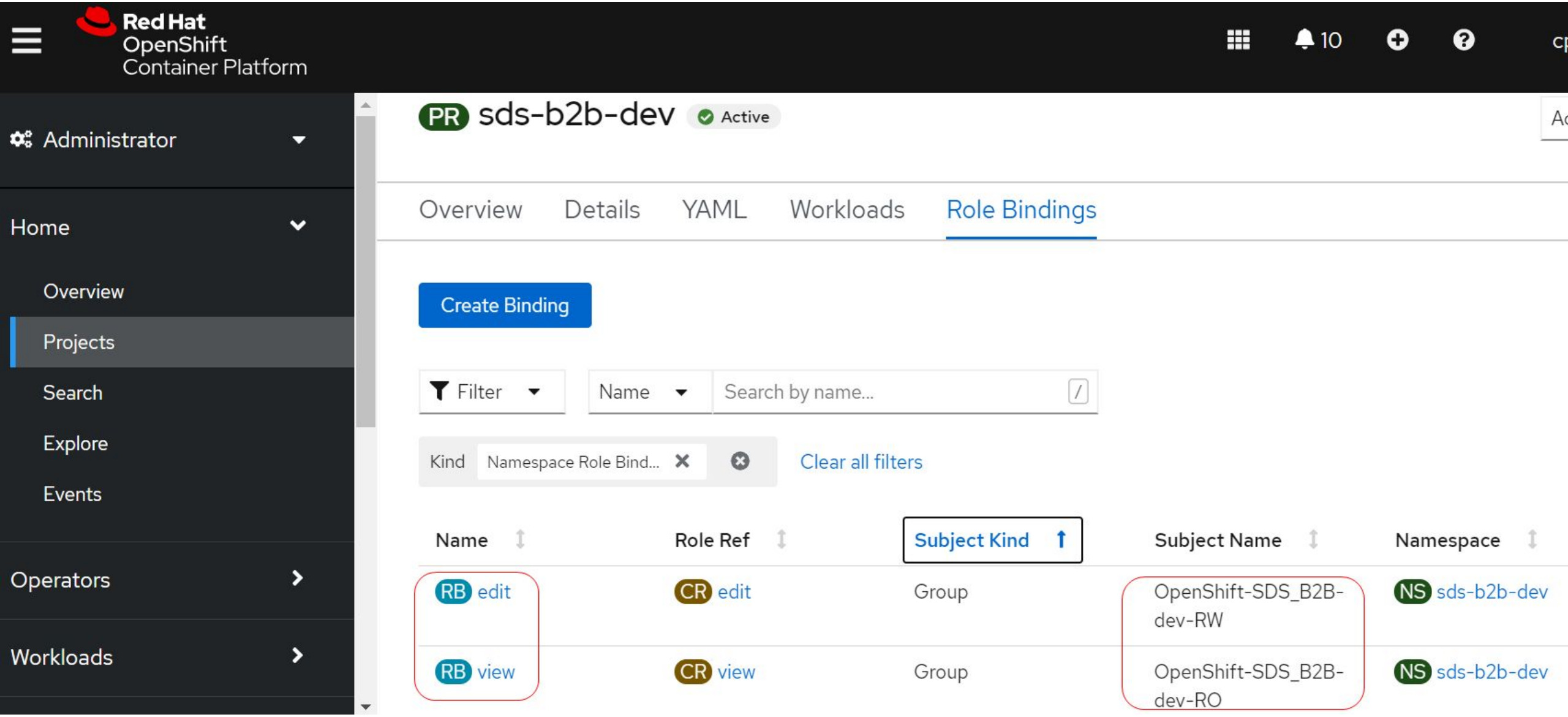
NetNamespaces > NetNamespace Details

NN sds-b2b-dev

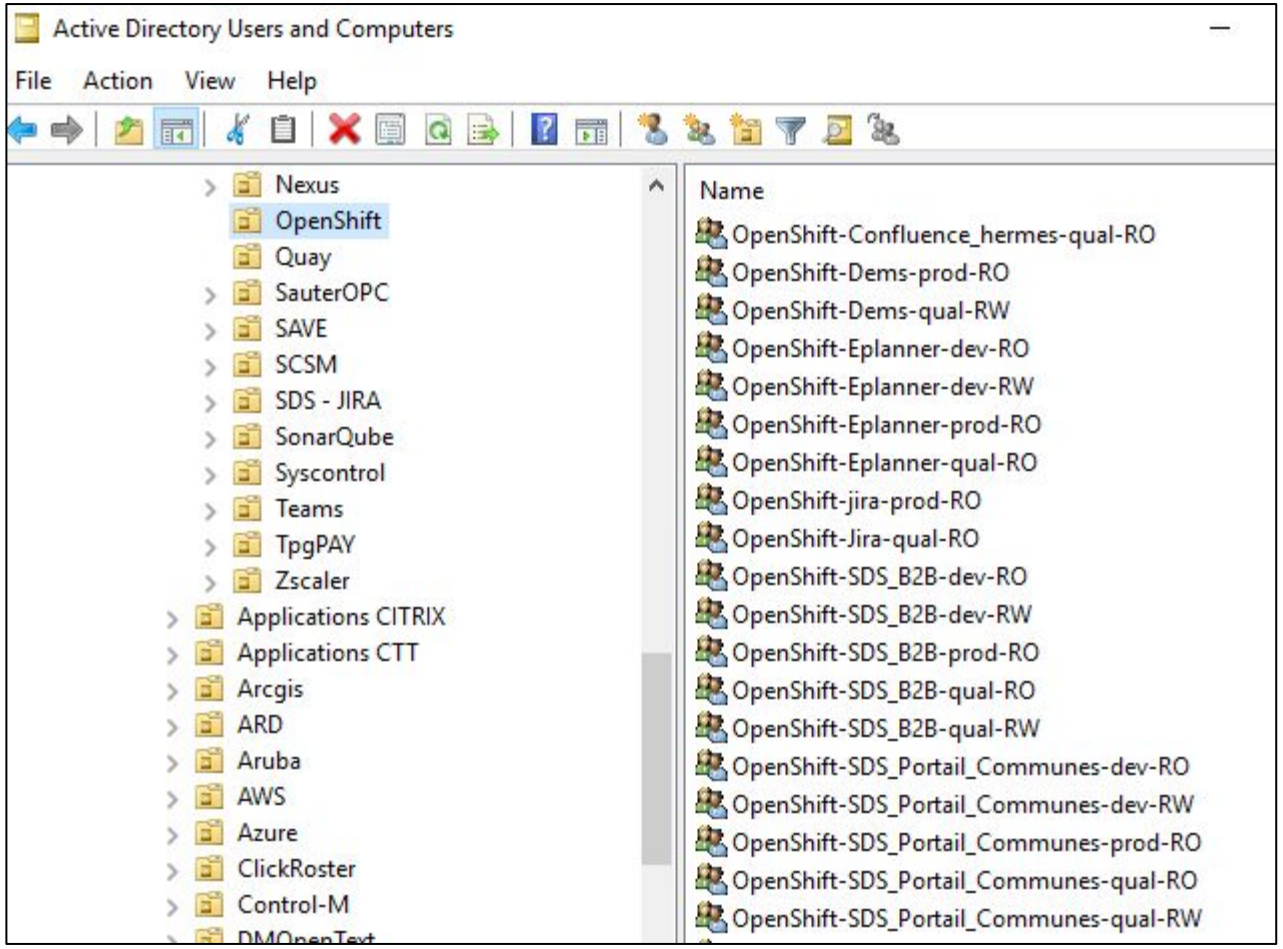
Details YAML

```
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



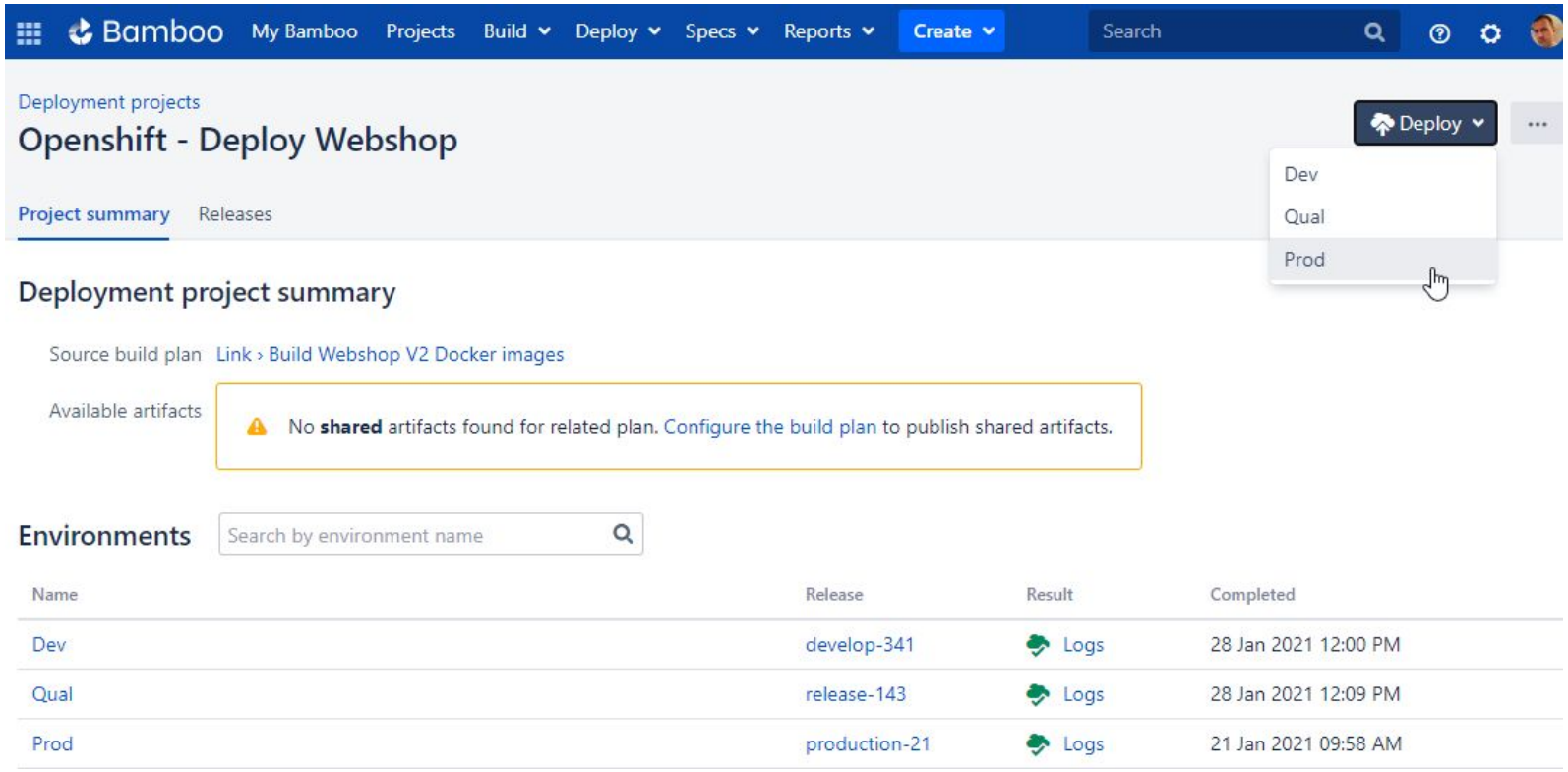
Our Active Directory supports Openshift RBAC



- 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»



Conclusion

- 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 …