Automated project provisioning using the operator software developer kit

Marc Brugger
Senior Software Engineer
SIX

Beat Stutz
Project Manager
SIX

Sylvain Chen
Middleware Consultant
Red Hat

May 8th, 2019
Agenda

• About SIX
• OpenShift @ SIX
• Project provisioning operator and demo
• Challenges & Takeaways
Our Assignment

SIX develops and operates infrastructure services for the Swiss financial center and Swiss banks. This includes services in the securities business, for payment transactions, for financial information, and for cyber security.
Stock Exchange

2000 times faster than the blink of an eye
the stronghold for securities
Financial Information

135,252 price updates per second
Payment & Banking Services

99.998%
application up-time

the IT that never sleeps
OpenShift Project Overview: Four Main Objectives

Operationalize the container platform

Build up know-how and a dedicated team

Position the platform as a standard for new containerized applications

Enable cloud-readiness for our IT Infrastructure

- Operationalize the container platform
- Build up know-how and a dedicated team
- Position the platform as a standard for new containerized applications
- Enable cloud-readiness for our IT Infrastructure
Oliver Guggenbühl
Tech Lead PaaS
Marc Brugger
Software Engineer
Dominik Hahn
Cloud Engineer
Beat Stutz
Project Manager
Ben Schwarz
Head Linux CC
Dave Brupbacher
Head IT Infrastructure & Operations
Sylvain Chen
Red Hat Consultant
Benefits for Business & IT of the container platform

- Ability to deploy new features fast
- Concentrate on **core business** instead of tooling and environment configuration
- **Operational** improvements, stability and transparency (integrated logging, monitoring)
- **Flexibility**: up-to-date technology that will work anywhere, including in the cloud
- High availability **included**
- **Scale** on demand, no **fixed cost**
The Expo

Paas
the future belongs to containers

SIX Containers
OpenShift environments

- On premise infrastructure
- 3 clusters:
  - Test cluster, few nodes
  - NonProd, 100 nodes, 200 projects
  - Prod, 70 nodes, 50 projects
- Two datacenters
- 30+ network zones
Project provisioning operator
Project requirements

- Naming conventions
- Namespace access rights
- Resources
- Chargeback
- Network zone
- Firewall information
- Alerting target
- Contact information
- Logserver credentials
The first steps without an operator

- Disable self provisioning of projects
- Provisioning of projects with OpenShift Templates and property files
  - Manual work by Ops team for each project
  - Version control is with Git but real state is in OpenShift
  - No proper lifecycle for projects
  - Global changes would impact all property files
Operator benefits

- Kubernetes manages Kubernetes
- Automation of project provisioning
- The unique source of truth is the CR (CustomResource)
- The operator controls the project lifecycle
- Developers can create projects
- Ops and developer teams get notifications
Project provisioning operator timeline

- **v0.0.6**
  - Aug: NonProd
  - Oct: Generate Rolebindings
  - Nov: Validation generic Rolebindings
  - Dev: Resource Limits

- **v0.1.0**
  - Sep: Production
  - Sep: Setup email for customer
  - Oct: Send email to Support

- **v0.2.1**
  - Sep: CLI

- **v0.3.0**
  - Oct: Tiller per project

- **v0.5.0**
  - Nov: Automatically update node selector

- **v0.7.0**
  - Dec: Version hashes
How to order OpenShift projects

- IaaS Portal calls the OpenShift APIs to provision the Project Provisioning CR
- Usage of Service Account with limited RBAC on the CR
- The Operator takes care of everything else by watching the CRs on the six-project-spaces project
Managed Objects

```
kind: ProjectProvisioning
metadata:
  name: rh-summit-dev
spec:
  project:
    name: rh-summit-dev
displayName: rh-summit
description: Demo Project for Red Hat Summit
requesterEmail: john.doe@six-group.com
location: 'node-group=app,country=ch'
group: demo-users
egress: true
jenkins: true
resources:
  project / pod / container:
    min / max:
      cpu: 4000m
      memory: 4Gi
splunkToken: my-splunk-hec-token
additionalRoleBindings:
  'system:image-pullers':
    roleBindings:
      - kind: ServiceAccount
        name: default
        namespace: rh-summit-qa
annotations:
labels:
```
Reconciliation Flow

CR watch event received

Is CR Deleted?
- Yes: Delete namespace
- No: Has Changes?

Has Changes?
- Yes: Is valid?
  - Yes: Create/Update namespace with all components
  - No: Save result to status
- No: Ignore

Is valid?
- Yes: Create/Update namespace with all components
- No: Save result to status
Validation

- Content validation with govalidator
- Messages in status
- Label on error
- Prometheus metric

```go
Name      string `valid:"required,length(2|56),matches([a-zA-Z0-9]*(?:[-a-zA-Z0-9]*[a-zA-Z0-9])?)"`
ContactEmail string `valid:"optional,email"`
```

```
status:
  configHash: 884fd34ce6edf1d518bbe9b85d08f5ef
  operatorVersion: v2.0.20190403174537
  processingError: []
  specHash: 457229db1f91aece3596a26cd82ebd9c5
  status: VALIDATION_ERROR
  validationError: 'location: non zero value required'
```
Demo
Quota

Resource Requests
CPU
- Used: 4
- Limit: 4

Memory
- Used: 4 GiB
- Limit: 4 GiB

Limit Range

Resource Requests
CPU
- Used: 0
- Limit: 0

Memory
- Used: 0
- Limit: 0
Challenges with operator SDK

- Periodically full reload of all CR’s
  - Hashing CR to check if changed or just reloaded
- Periodically logged errors from operator SDK or controller framework
- Logs are saved to file in /tmp when logged as error
  - Permission denied when running in scratch container
    → emptyDir for /tmp
- Getting the correct go module versions of k8s and OpenShift when building with go mod
Key Takeaways

- Managing 200+ projects
- Easy to apply changes
- Additional operators in use
  - route ACL enforcement
  - egress configuration
Outlook

● Implement new features as needed

● Setting up tiller per namespace

● Use validating admission webhook

● Concurrent processing
Links

- https://github.com/kubernetes-sigs/controller-runtime
- https://github.com/asaskevich/govalidator
- https://upx.github.io/
- https://learn.openshift.com/operatorframework/go-operator-podset
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/redhat