RED HAT SUMMIT

Avaloq’s journey to a standardized, scalable banking reference architecture

Christoph Zehnder, Avaloq
Daniel Bejinaru, Avaloq
Daniel Schaefer, Red Hat

May 2019

Speakers

Christoph Zehnder
Avaloq

Hands on software architect

Daniel Bejinaru
Avaloq

From the "as a service" part of the company

Daniel Schaefer
Red Hat

Sr. Solutions Architect working with ISVs across EMEA

# avaloq

A software

- Core banking software, digital banking and digital wealth management used by 150 banks
- Offered "as a service" and "on premise"

BARCLAYS deutsche apotheker- und ärztebank BT Financial Group Coutts

DBS Deutsche Bank Edmond de Rothschild HSBC

LGT Maybank RAIFFEISEN SOCIETE GENERALE

A company

- 2200 employees (including 700 developers)
- Service centers in Switzerland, Germany and Singapore
- Development in Switzerland, UK, Philippines

Why are we here?

Our story

- OpenShift is a great product
- But challenging to implement in an enterprise / financial industries environment
- Present the challenges we faced and some decisions we took

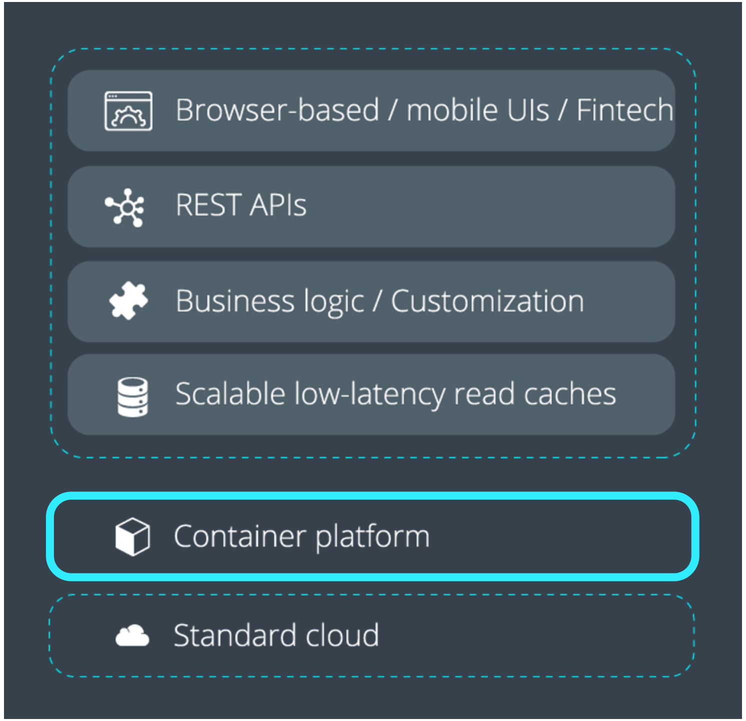
And: Don’t expect us to sell our products or services!

But: We are always looking for good developers :)

Adapt to changed requirements

organizational scalability
mobile
innovation
Future?
fintech
consistency
SaaS
risk
open banking
STP
modular adoption
Security
low cost
access to talents
regulatory
investment protection
UX
data quality
short time to market
access to

Avaloq's open banking architecture



Browser-based / mobile UIs / Fintech
REST APIs
Business logic / Customization
Scalable low-latency read caches
Container platform
Standard cloud

The platform to efficiently and effectively develop and operate great banking functionality.

Container platform

kubernetes

RED HAT®
OPENSHIFT

Transformation in ecosystems

Why we think this case is interesting

It’s all about the application!

Boosting customer success with modernized applications and by enabling new DevOps methodology.

Standardize and open up!

Because solving complex business problems with complex software on a complex IT stack won’t scale well…

Ecosystem Collaboration!

ISV, Service Provider, Technology Partner and System Integrator collaborating for joined customers.

Who are you going to call?

Embedded Model allows Avaloq ownership of customer support experience with Red Hat in the back.

Challenges

OpenShift is a technical requirement

- Investment in implementing and supporting the Avaloq Container Platform
- Client uncertainty poses open questions (from design, security, to high effort estimation and budget)

Financial Regulations / Security

- Protecting Client Identifying Data
- Swiss banking rules are strict - there is no easy way to do "DevOps"
- Service provider and software builder for banks (not a bank)

Shift of responsibilities

“Dev” and “Ops” in two separate companies

Build trust

Focus on security, enable transparency and communication

Collaboration

Involvement and contribution

Designed for operations

New responsibilities for all parties

Reuse

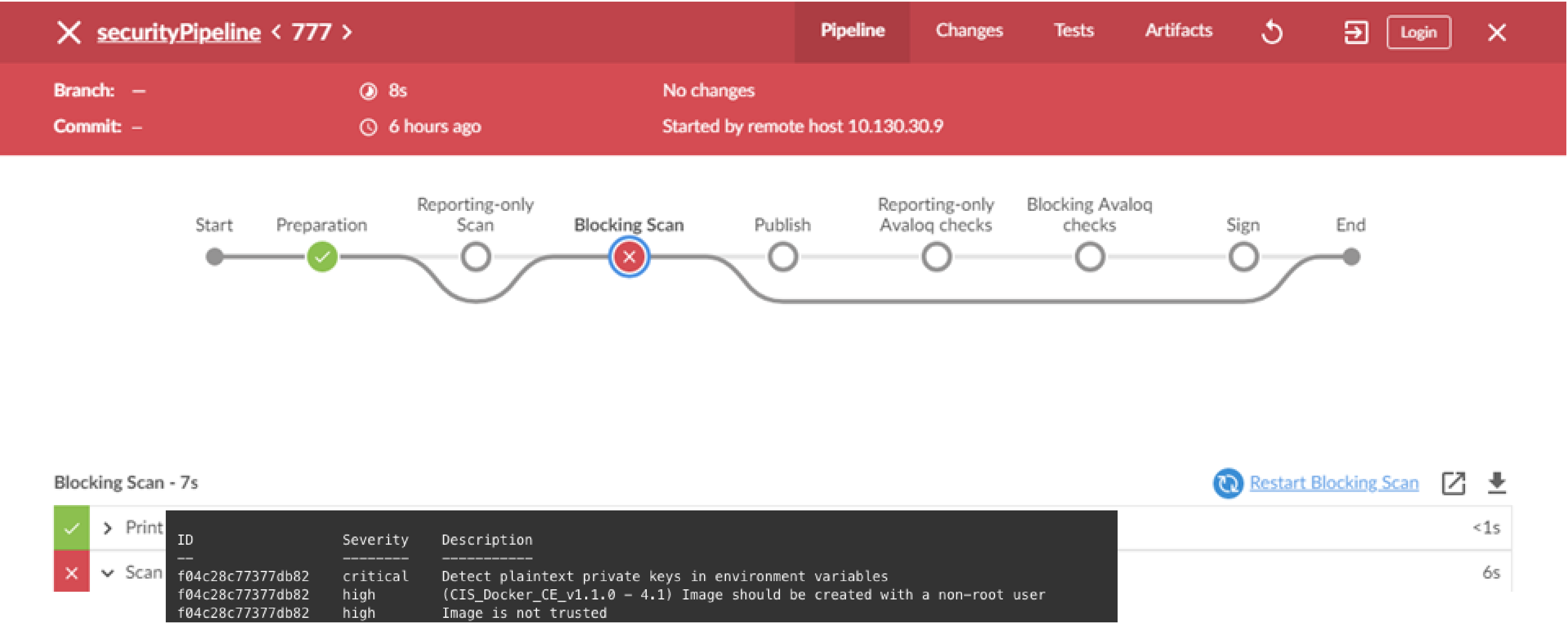
And avoid re-inventing the wheel

Build Trust: Security

Secure container images

```
FROM docker.io/centos
ENV privatekey myprivatekey.pem
CMD while true; do sleep 1; done
```

Security pipeline



Branch: —

Commit: –

8s

6 hours ago

No changes

Started by remote host 10.130.30.9

Blocking Scan - 7s

```
ID                  Severity    Description
--                  --------    -----------
f04c28c77377db82    critical    Detect plaintext private keys in environment variables
f04c28c77377db82    high        (CIS_Docker_CE_v1.1.0 - 4.1) Image should be created with a non-root user
f04c28c77377db82    high        Image is not trusted
```

Secure container images

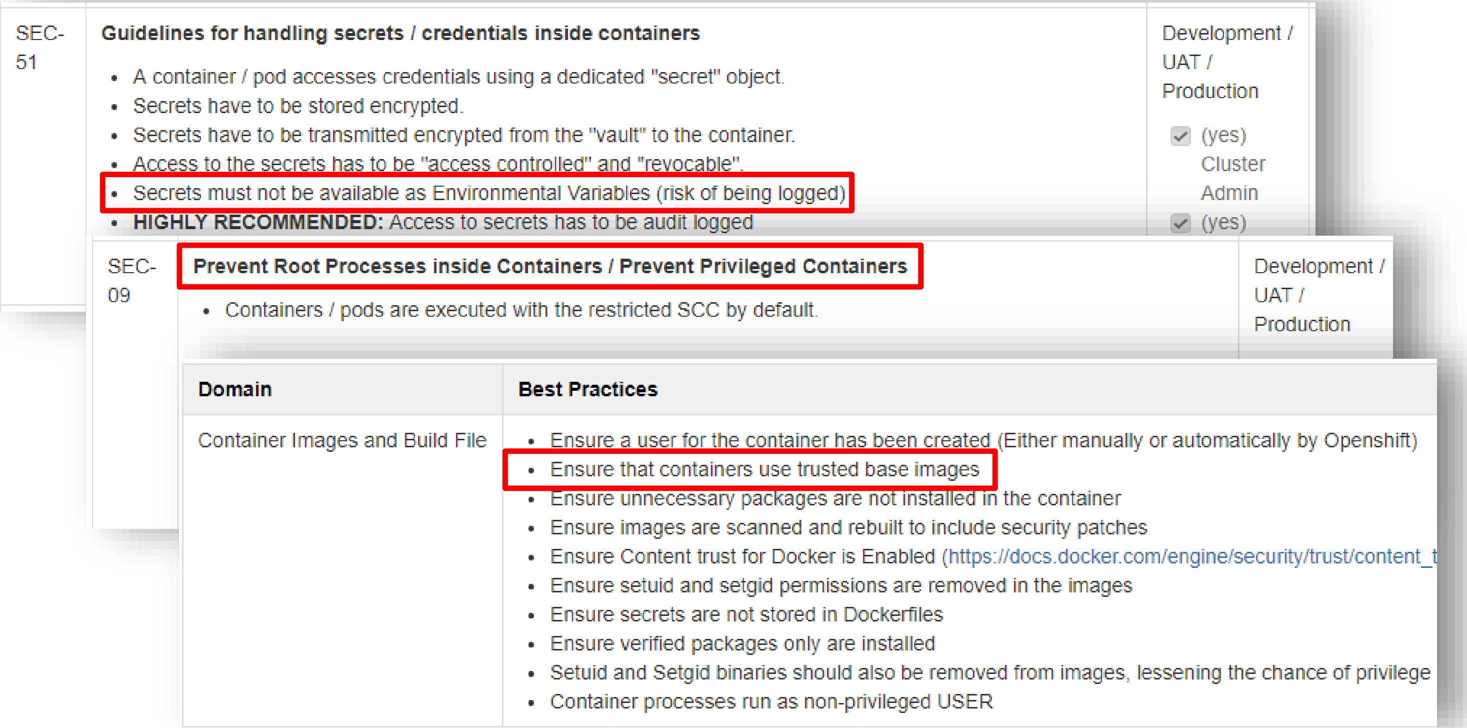
```
FROM docker.io/centos
ENV privatekey myprivatekey.pem
CMD while true; do sleep 1; done
```

No trusted Avaloq Base Image

Credentials in environment variables

No unprivileged user specified

Security guidelines and best practices

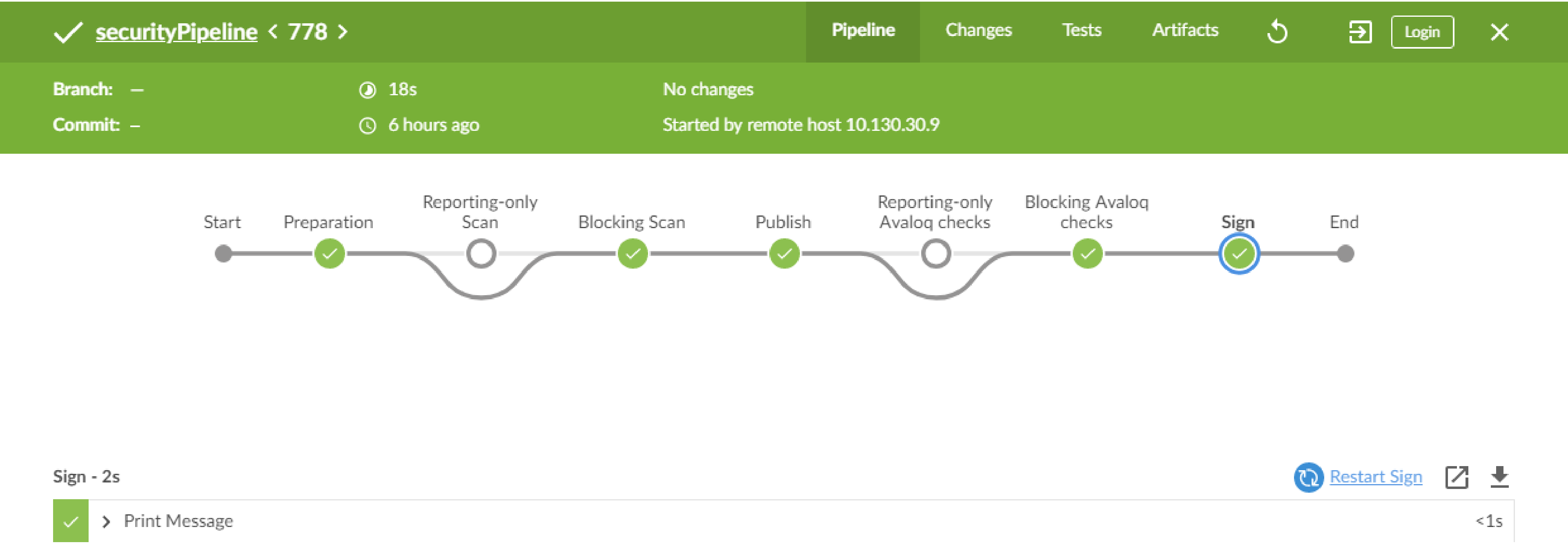


SEC-51	**Guidelines for handling secrets / credentials inside containers** • A container / pod accesses credentials using a dedicated "secret" object. • Secrets have to be stored encrypted. • Secrets have to be transmitted encrypted from the "vault" to the container. • Access to the secrets has to be "access controlled" and "revocable". • Secrets must not be available as Environmental Variables (risk of being logged) • **HIGHLY RECOMMENDED:** Access to secrets has to be audit logged	Development / UAT / Production ✓ (yes) Cluster Admin ✓ (yes)

SEC-09	**Prevent Root Processes inside Containers / Prevent Privileged Containers** • Containers / pods are executed with the restricted SCC by default.	Development / UAT / Production

Domain	Best Practices
Container Images and Build File	• Ensure a user for the container has been created (Either manually or automatically by Openshift) • Ensure that containers use trusted base images • Ensure unnecessary packages are not installed in the container • Ensure images are scanned and rebuilt to include security patches • Ensure Content trust for Docker is Enabled (https://docs.docker.com/engine/security/trust/content_t • Ensure setuid and setgid permissions are removed in the images • Ensure secrets are not stored in Dockerfiles • Ensure verified packages only are installed • Setuid and Setgid binaries should also be removed from images, lessening the chance of privilege • Container processes run as non-privileged USER

Security pipeline



Branch: —

Commit: –

18s

6 hours ago

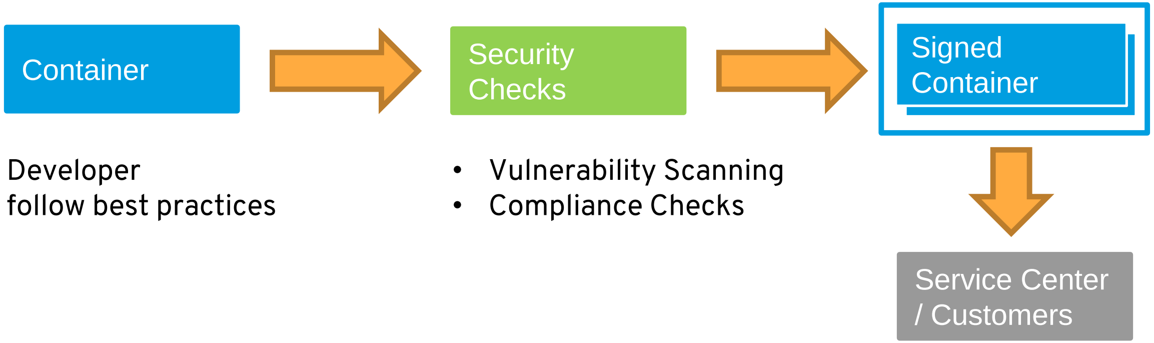
No changes

Started by remote host 10.130.30.9

Sign - 2s

✓ › Print Message <1s

Approved container images



Container

Developer
follow best practices

Security Checks

- Vulnerability Scanning
- Compliance Checks

Periodic rescanning of
all images

Signed Container

Service Center / Customers

Signature validation on
OpenShift Nodes

What cluster setup do I need?

Does this fit into a secure 3-tiered network?

"It depends ..."

Reference Architectures

PCI-DSS Reference Architecture

Ten Layers of Container Security

OpenShift Docs

Physical vs. Logical Segregation

“Is SDN secure enough?”

Physical node per tenant?

Can Dev and Prod run in the same cluster?

Node Placement

Where to put Masters, Routers and Workers?

One cluster per network zone is more secure – but way more expensive…

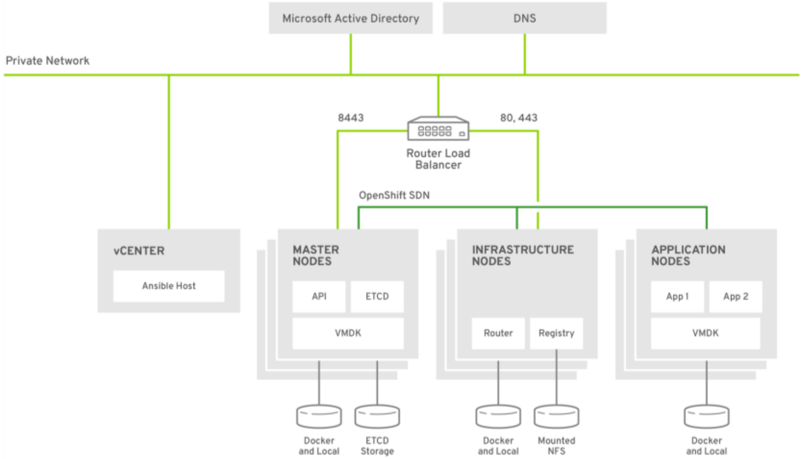
Everything in one zone to reduce complexity?

Lock down

Will additional firewalls lead to more security?

Blog Post on Security Zone Coexistence Approach

OpenShift reference architecture



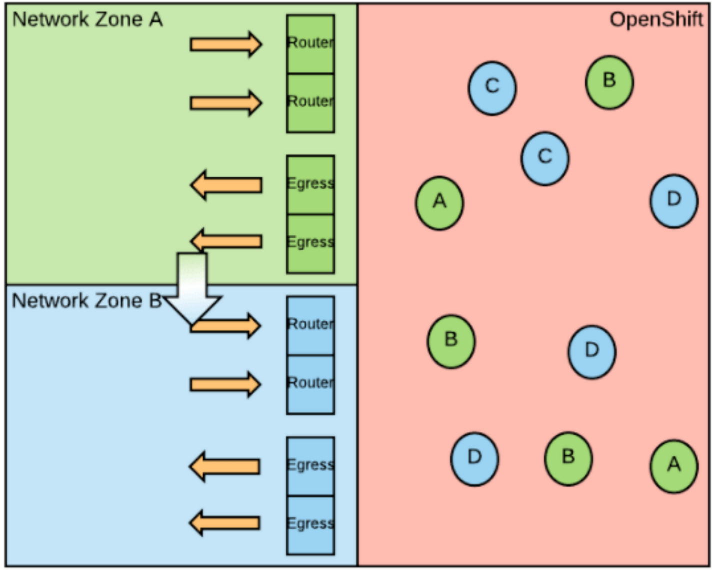
Considered the reference for the implementation

"It is not enough" to fulfill security requirements

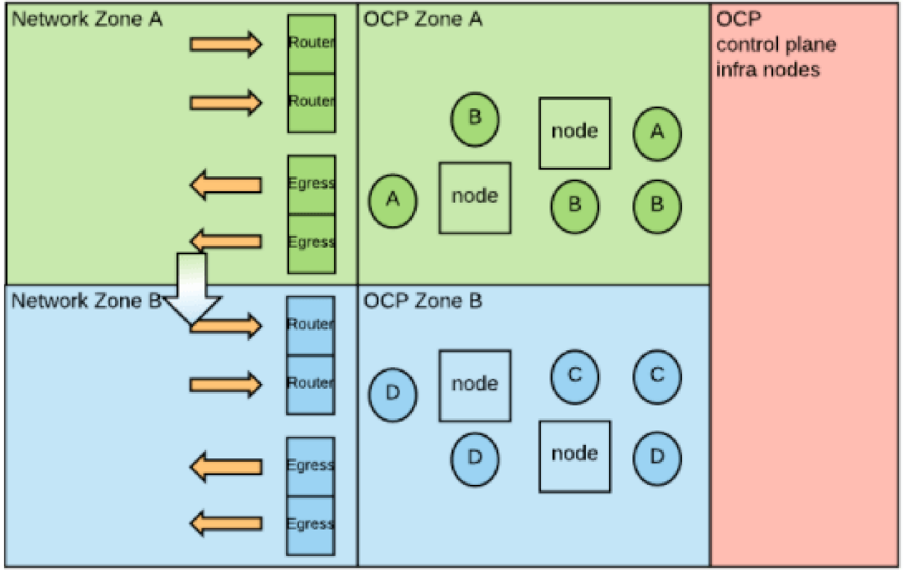
Keep compatibility, avoid major customizations

Different placement options

OpenShift allows flexible placement of nodes in different zones



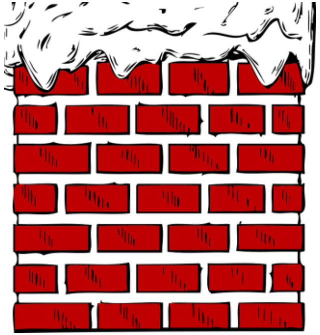
Shared Master and Worker nodes
Dedicated Router per zone



Dedicated Router and Worker nodes per zone
Shared control plane
Node selectors to place pods on specific worker nodes

Source: Blog Post on Security Zone Coexistence Approach

How can this run in my datacenter?



Security

- Network segregation
- Reduced attack surface and impact mitigation

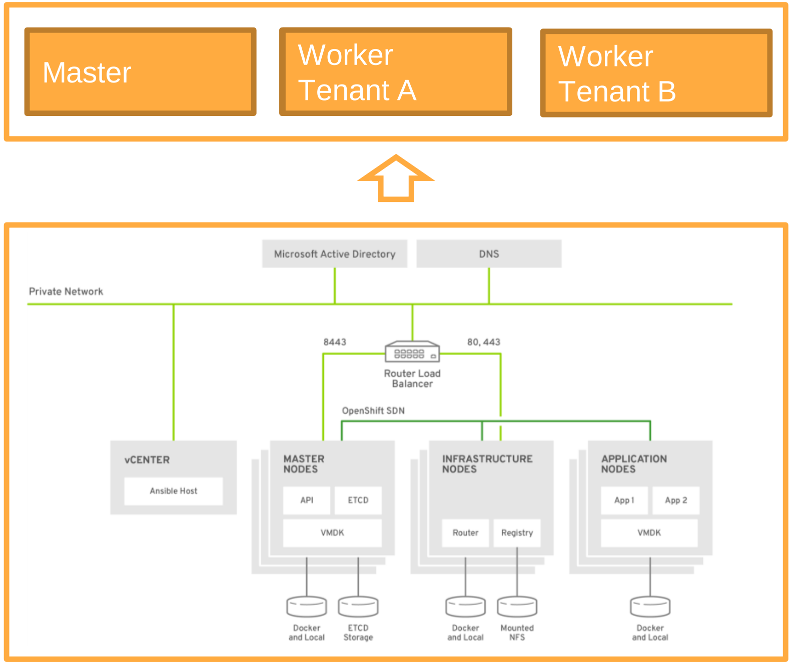
Operations

- Established Operating Model and tools with layered accountability
- Strictly defined processes, rules and tools from DEV to OPS

Control

- Strictly controlled configuration
- Traditional change and release management process

OpenShift reference architecture

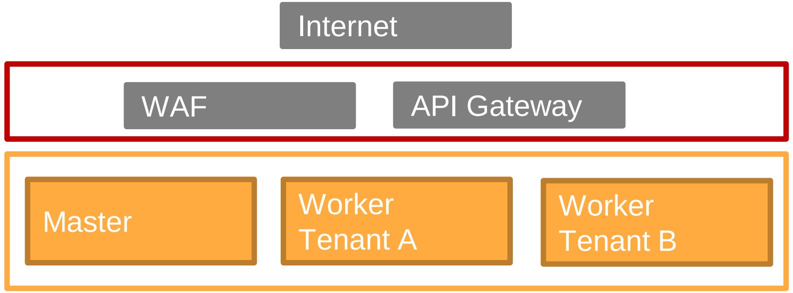


Master

Worker Tenant A

Worker Tenant B

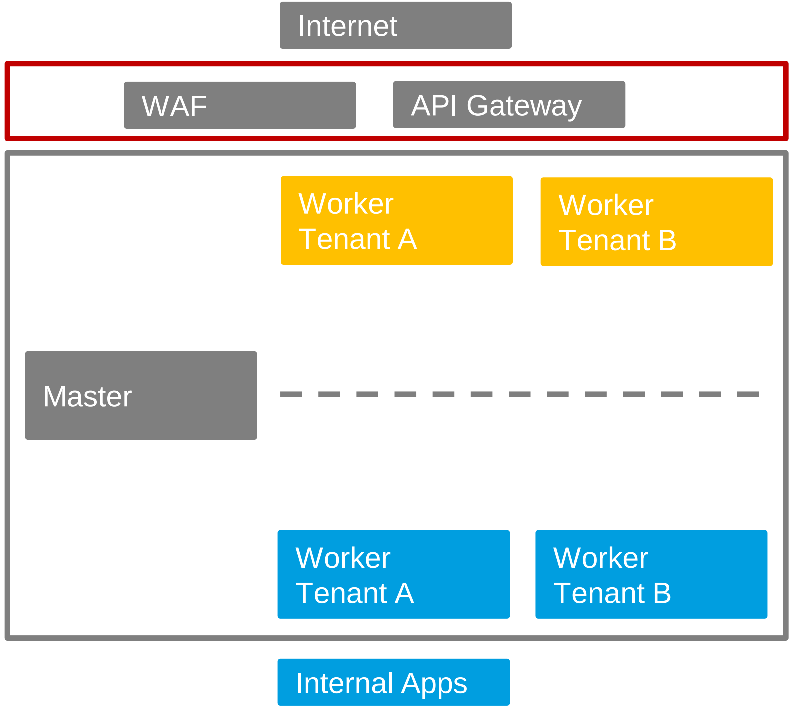
Internet publishing



Untrusted Zone (DMZ)

- Internet facing components

Security zones



Untrusted Zone (DMZ)

- Internet facing components

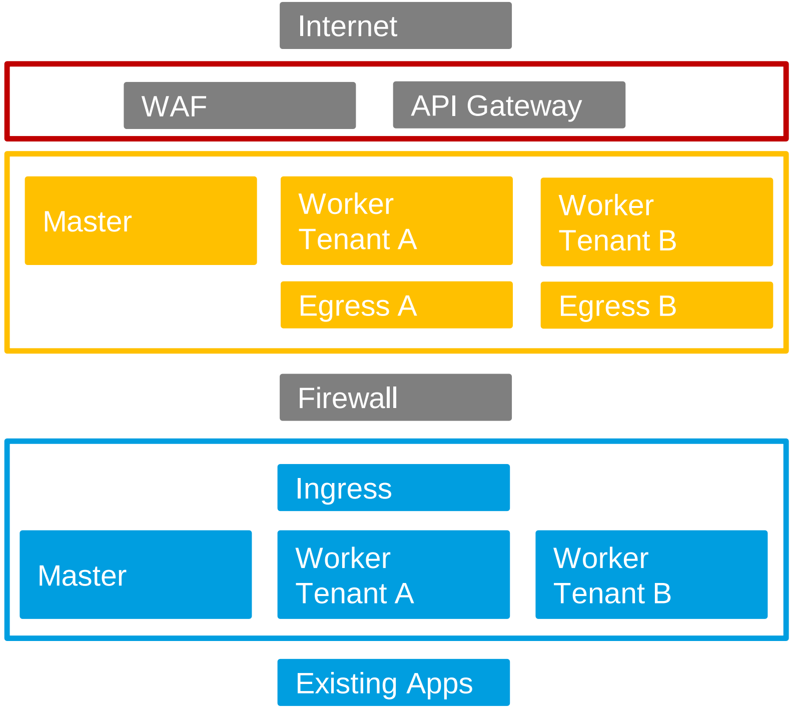
Semi-trusted Zone

- Internet Exposed Services

Trusted Zone

- Customer Identifying Data (CID)
- Business critical processes

Security zones: two clusters



Untrusted Zone (DMZ)

- Internet facing components

Semi-trusted Zone

- Internet Exposed Services

Trusted Zone

- Customer Identifying Data (CID)
- Business critical processes

Impact on Customers and Service Centers

Additional Costs

Service Availability

Managing 800 Clusters

Hybrid Environments

The answer is standardization, re-use and automation!

How can I efficiently deploy applications?

Different options to deploy containers

Deployments can be easily automated, choose your way:

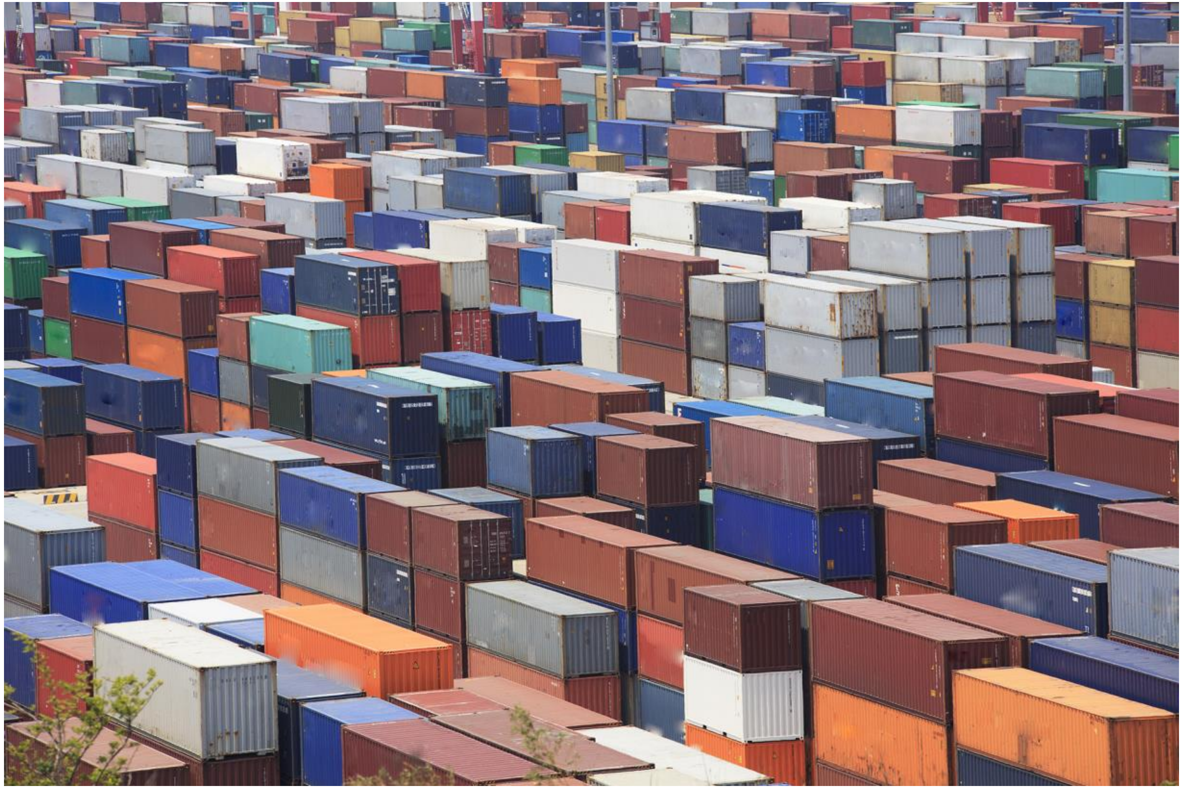
OpenShift Deployments & Templates

Ansible Playbook Bundles

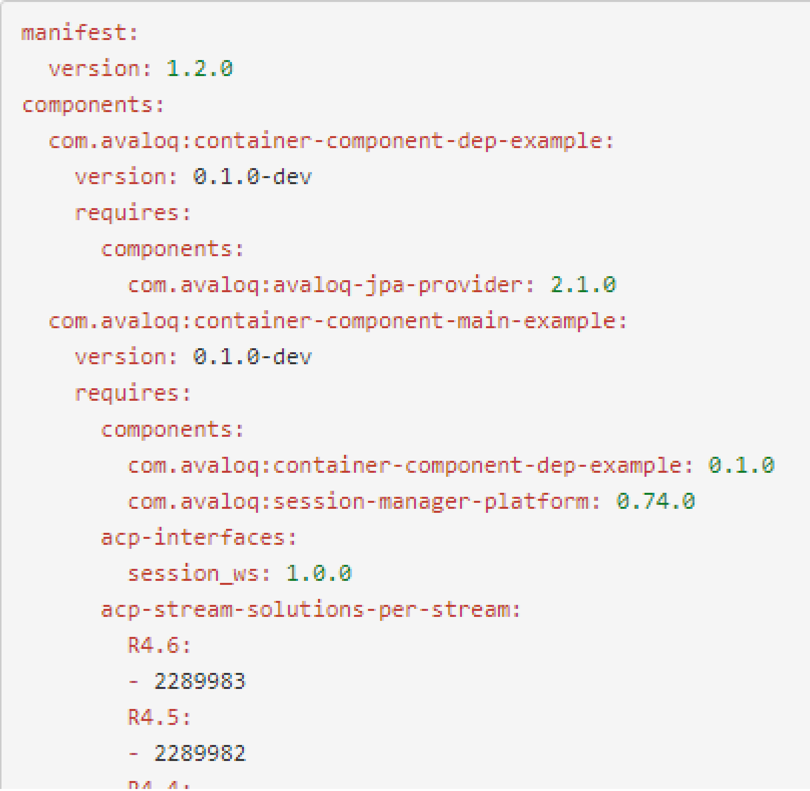
Operator Framework

- Native Kubernetes commands (kubectl)
- Helm Charts
- Many open source tools, active and deprecated (ksonnet, …)
- Anything can call an API (CI/CD tools, automation tools, scripts, …)

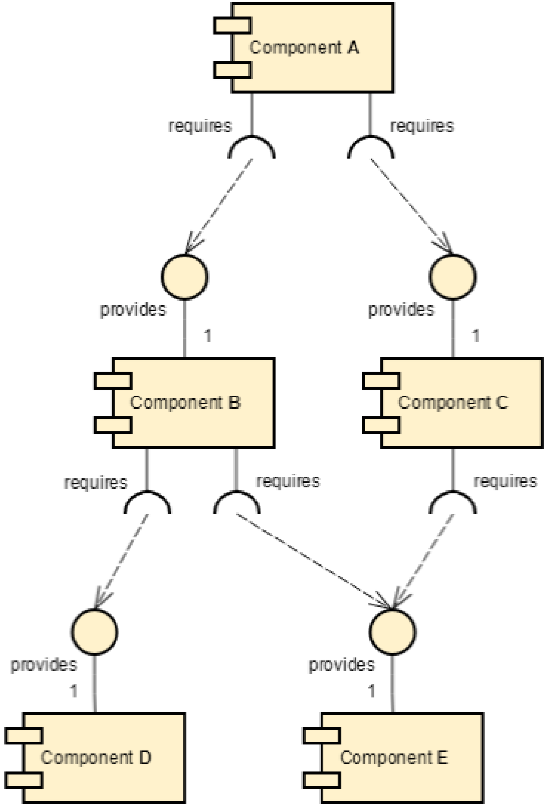
Which containers do I have to deploy?



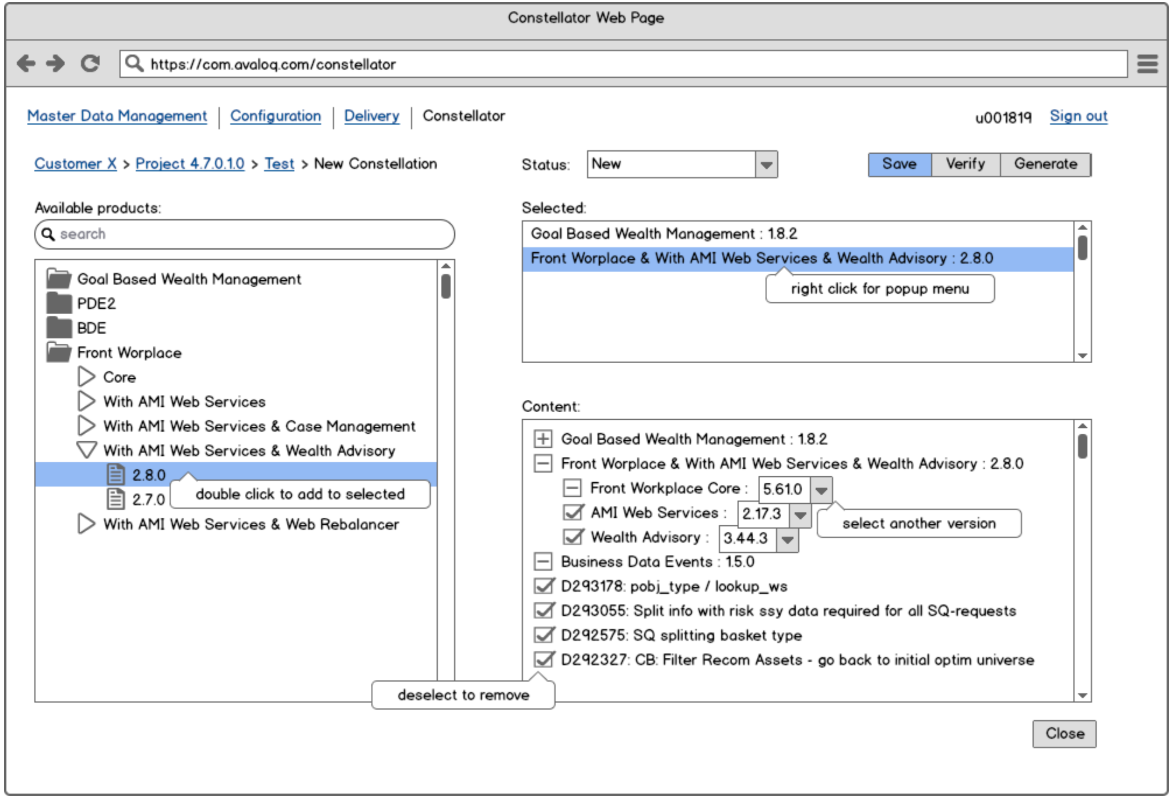
Define dependencies



```
manifest:
  version: 1.2.0
components:
  com.avaloq:container-component-dep-example:
    version: 0.1.0-dev
    requires:
      components:
        com.avaloq:avaloq-jpa-provider: 2.1.0
  com.avaloq:container-component-main-example:
    version: 0.1.0-dev
    requires:
      components:
        com.avaloq:container-component-dep-example: 0.1.0
        com.avaloq:session-manager-platform: 0.74.0
      acp-interfaces:
        session_ws: 1.0.0
      acp-stream-solutions-per-stream:
        R4.6:
        - 2289983
        R4.5:
        - 2289982
        R4.4:
```



Customer selects components



Constellator Web Page

https://com.avaloq.com/constellator

Master Data Management | Configuration | Delivery | Constellator u001819 Sign out

Customer X > Project 4.7.0.1.0 > Test > New Constellation

Status: New

Available products:

- Goal Based Wealth Management
- PDE2
- BDE
- Front Worplace
  - ▷ Core
  - ▷ With AMI Web Services
  - ▷ With AMI Web Services & Case Management
  - ▽ With AMI Web Services & Wealth Advisory
    - 2.8.0
    - 2.7.0
  - ▷ With AMI Web Services & Web Rebalancer

double click to add to selected

Selected:

Goal Based Wealth Management : 1.8.2

Front Worplace & With AMI Web Services & Wealth Advisory : 2.8.0

right click for popup menu

Content:

- ⊞ Goal Based Wealth Management : 1.8.2
- ⊟ Front Worplace & With AMI Web Services & Wealth Advisory : 2.8.0
  - ⊟ Front Workplace Core : 5.61.0
  - ☑ AMI Web Services : 2.17.3
  - ☑ Wealth Advisory : 3.44.3
- ⊟ Business Data Events : 15.0
- ☑ D293178: pobj_type / lookup_ws
- ☑ D293055: Split info with risk ssy data required for all SQ-requests
- ☑ D292575: SQ splitting basket type
- ☑ D292327: CB: Filter Recom Assets - go back to initial optim universe

select another version

deselect to remove

Constellation

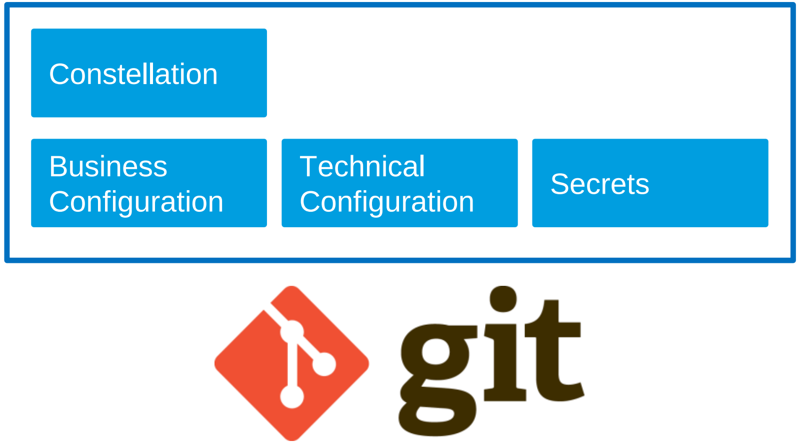
```
products:
- components:
  - group: com.avaloq
    name: avaloq-aws
    version: 2.24.0
  group: com.avaloq
  name: aws
  requiredProducts: null
  version: 1.0.0
- components:
  - group: com.avaloq
    name: avaloq-zookeeper
    version: 3.0.0
  - group: com.avaloq
    name: avaloq-kafka
    version: 5.3.0
  - group: com.avaloq
    name: avaloq-meteor
```

Store everything in a repository

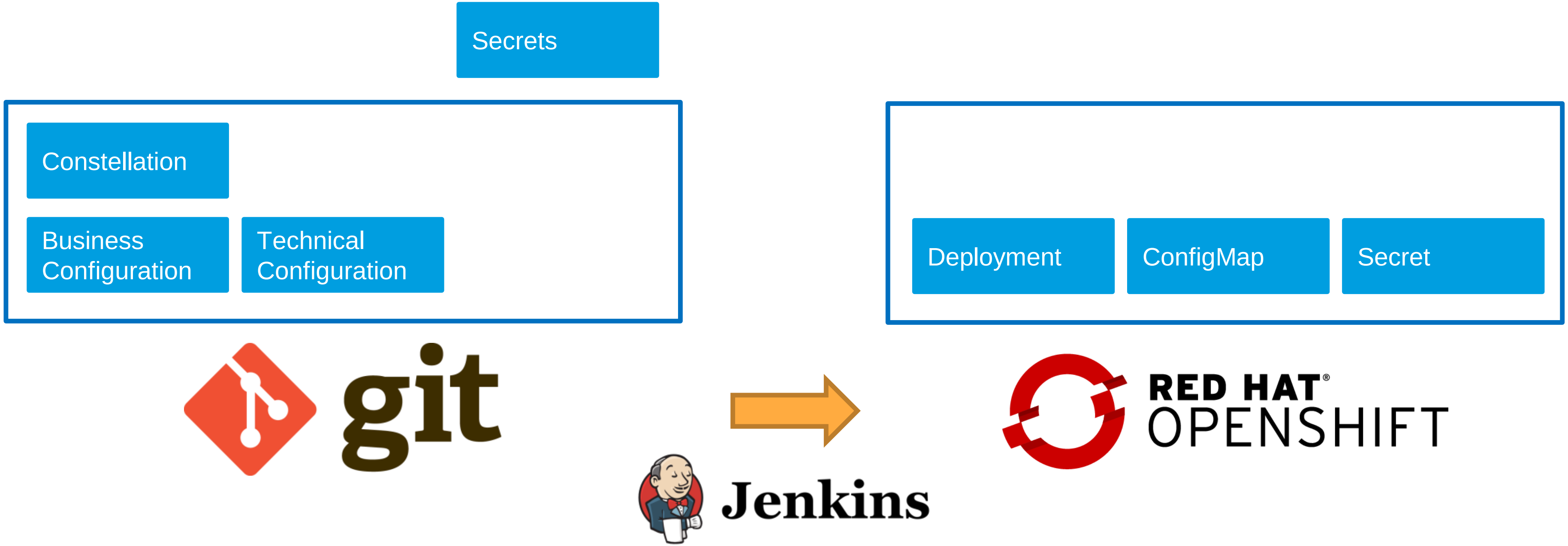
Constellation

git

Add configuration



Use a job to deploy to OpenShift



Secrets

Constellation

Business Configuration

Technical Configuration

Deployment

ConfigMap

Secret

OpenShift configuration in source control

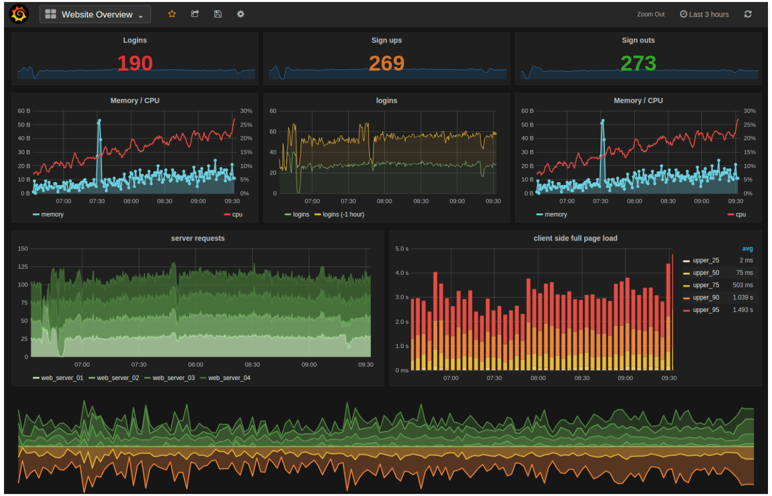
Why "do it yourself"?

How can I operate this?

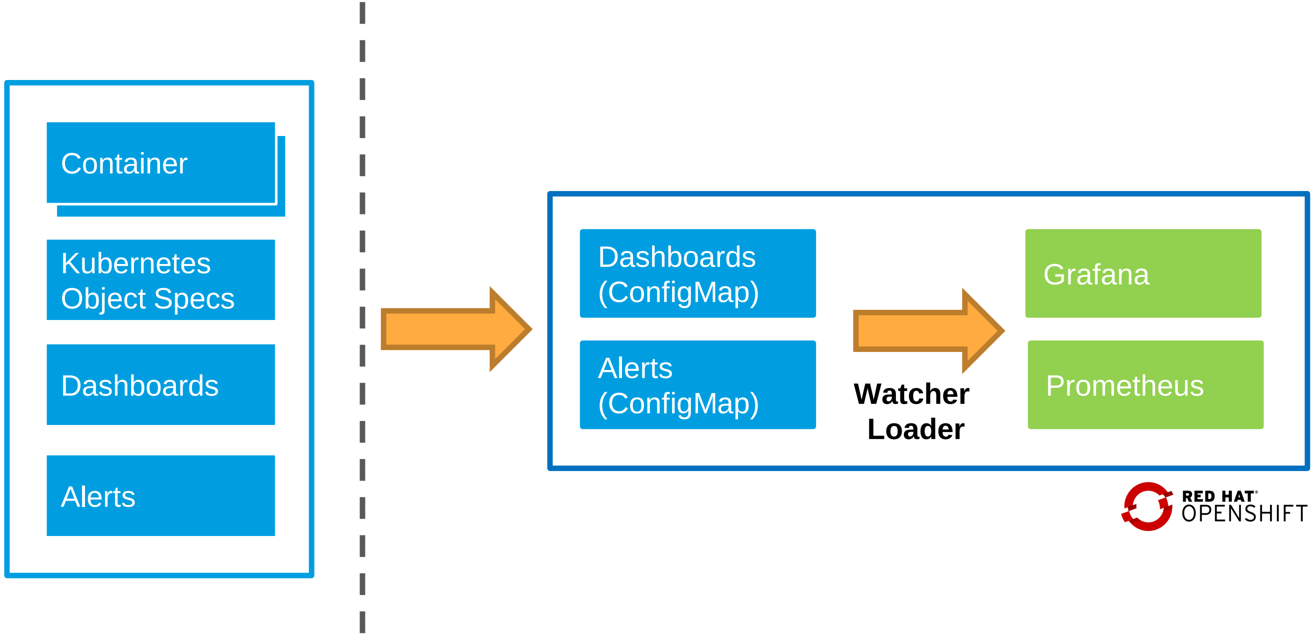
What do I need additionally?

→ Dashboards

→ Alerts



Dashboards and alerts



Producer
(Developer / Vendor)

Consumer
(Service Center / Bank)

Conclusion

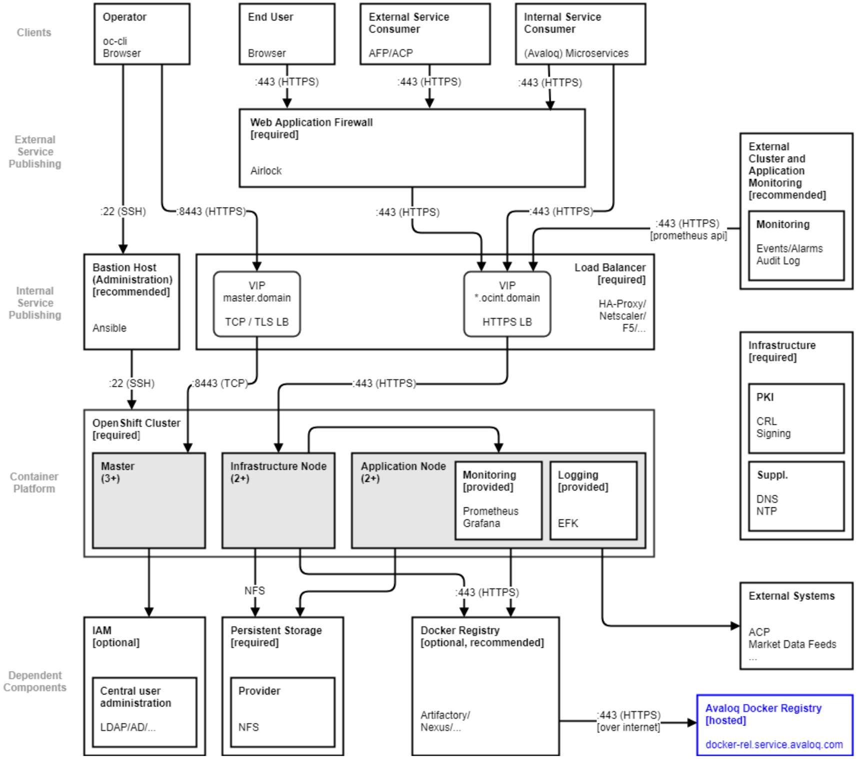
- There are **many solutions with different options** from Red Hat and the cloud native community.
- But Avaloq customers **can’t lose time evaluating options.**
- They just need **a good default!** (for the banking industry)
- **Avaloq needs standardization** to support our customers! (we are an ISV)

Avaloq Container Platform

Share within the Avaloq community

Concepts

Reference architecture



Clients

Operator
oc-cli
Browser

End User
Browser

External Service Consumer
AFP/ACP

Internal Service Consumer
(Avaloq) Microservices

:443 (HTTPS)

:443 (HTTPS)

:443 (HTTPS)

External Service Publishing

Web Application Firewall
[required]
Airlock

External Cluster and Application Monitoring
[recommended]

Monitoring
Events/Alarms
Audit Log

:22 (SSH)

:8443 (HTTPS)

:443 (HTTPS)

:443 (HTTPS)

:443 (HTTPS)
[prometheus api]

Internal Service Publishing

Bastion Host (Administration)
[recommended]
Ansible

VIP
master.domain
TCP / TLS LB

VIP
*.ocint.domain
HTTPS LB

Load Balancer
[required]
HA-Proxy/
Netscaler/
F5/...

Infrastructure
[required]

PKI
CRL
Signing

Suppl.
DNS
NTP

:22 (SSH)

:8443 (TCP)

:443 (HTTPS)

Container Platform

OpenShift Cluster
[required]

Master
(3+)

Infrastructure Node
(2+)

Application Node
(2+)

Monitoring
[provided]
Prometheus
Grafana

Logging
[provided]
EFK

NFS

:443 (HTTPS)

External Systems
ACP
Market Data Feeds
...

Dependent Components

IAM
[optional]

Central user administration
LDAP/AD/...

Persistent Storage
[required]

Provider
NFS

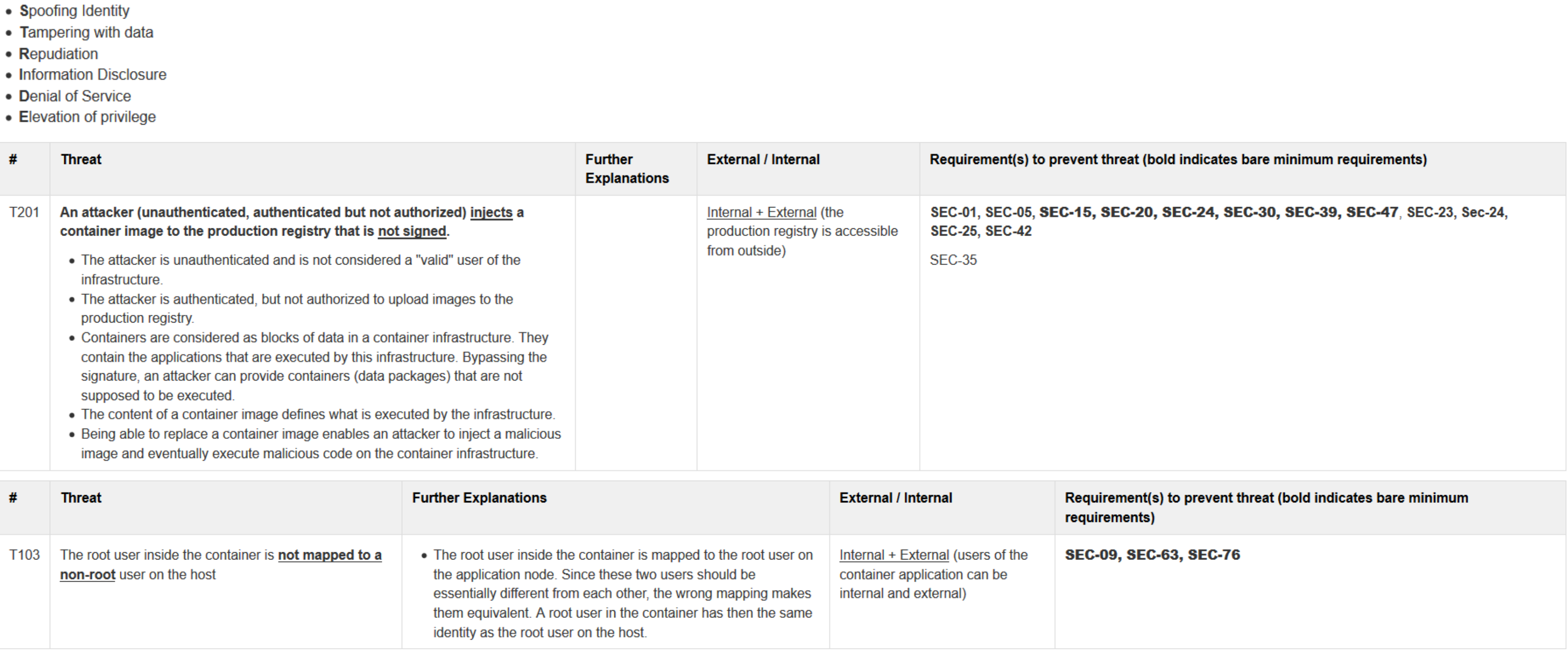
Docker Registry
[optional, recommended]
Artifactory/
Nexus/...

:443 (HTTPS)
[over internet]

Avaloq Docker Registry
[hosted]
docker-rel.service.avaloq.com

Documentation

Security: threat model

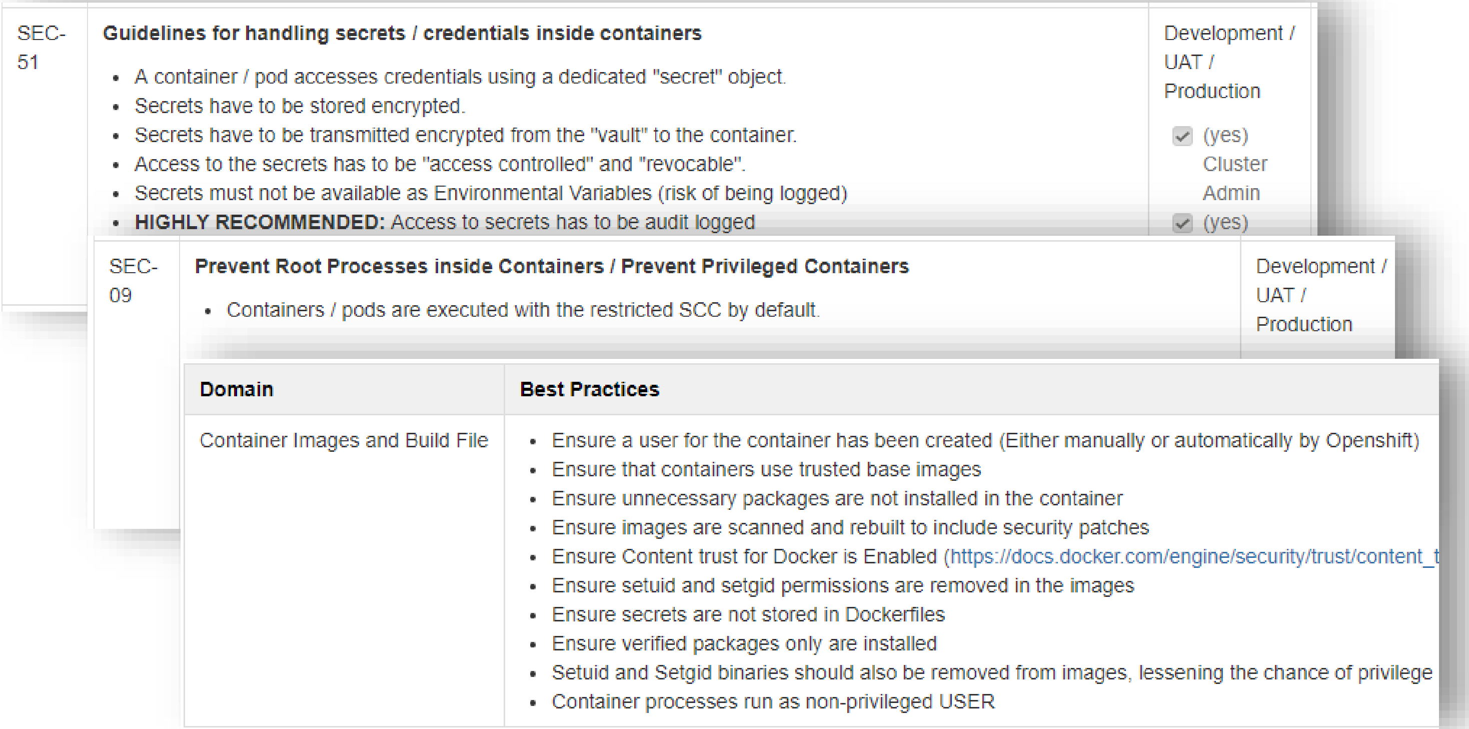


- **S**poofing Identity
- **T**ampering with data
- **R**epudiation
- **I**nformation Disclosure
- **D**enial of Service
- **E**levation of privilege

#	Threat	Further Explanations	External / Internal	Requirement(s) to prevent threat (bold indicates bare minimum requirements)
T201	**An attacker (unauthenticated, authenticated but not authorized) injects a container image to the production registry that is not signed.** • The attacker is unauthenticated and is not considered a "valid" user of the infrastructure. • The attacker is authenticated, but not authorized to upload images to the production registry. • Containers are considered as blocks of data in a container infrastructure. They contain the applications that are executed by this infrastructure. Bypassing the signature, an attacker can provide containers (data packages) that are not supposed to be executed. • The content of a container image defines what is executed by the infrastructure. • Being able to replace a container image enables an attacker to inject a malicious image and eventually execute malicious code on the container infrastructure.		Internal + External (the production registry is accessible from outside)	SEC-01, SEC-05, **SEC-15, SEC-20, SEC-24, SEC-30, SEC-39, SEC-47**, SEC-23, Sec-24, SEC-25, SEC-42 SEC-35

#	Threat	Further Explanations	External / Internal	Requirement(s) to prevent threat (bold indicates bare minimum requirements)
T103	The root user inside the container is **not mapped to a non-root** user on the host	• The root user inside the container is mapped to the root user on the application node. Since these two users should be essentially different from each other, the wrong mapping makes them equivalent. A root user in the container has then the same identity as the root user on the host.	Internal + External (users of the container application can be internal and external)	**SEC-09, SEC-63, SEC-76**

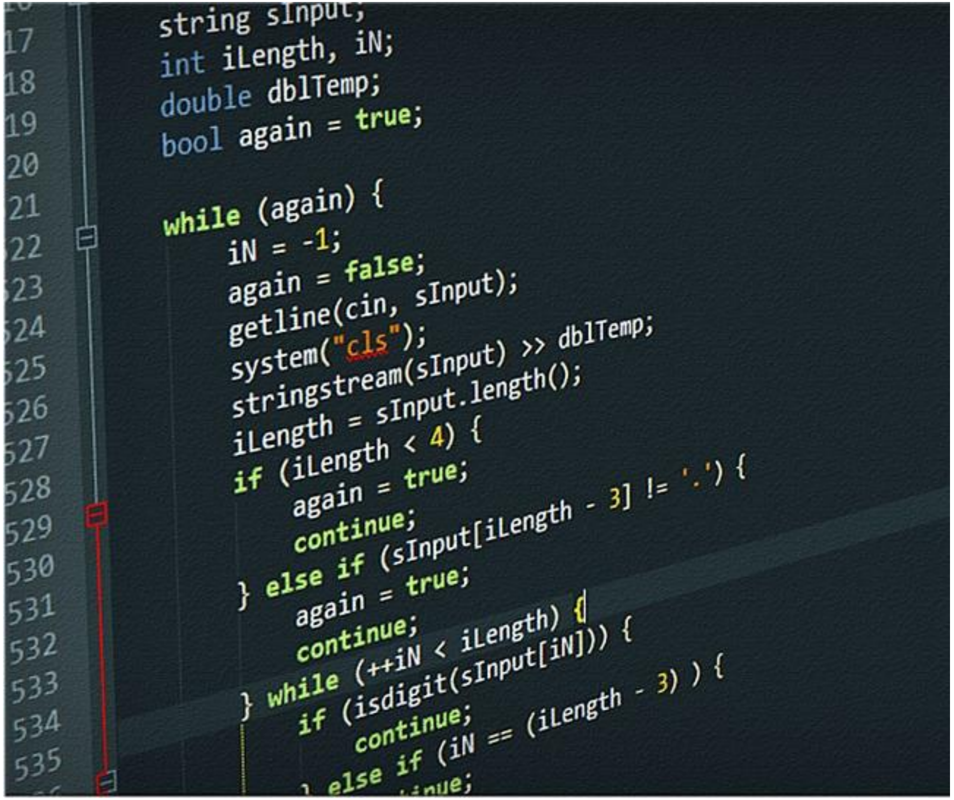
Security guidelines and best practices



SEC-51	**Guidelines for handling secrets / credentials inside containers** • A container / pod accesses credentials using a dedicated "secret" object. • Secrets have to be stored encrypted. • Secrets have to be transmitted encrypted from the "vault" to the container. • Access to the secrets has to be "access controlled" and "revocable". • Secrets must not be available as Environmental Variables (risk of being logged) • **HIGHLY RECOMMENDED:** Access to secrets has to be audit logged	Development / UAT / Production ☑ (yes) Cluster Admin ☑ (yes)

SEC-09	**Prevent Root Processes inside Containers / Prevent Privileged Containers** • Containers / pods are executed with the restricted SCC by default.	Development / UAT / Production

Domain	Best Practices
Container Images and Build File	• Ensure a user for the container has been created (Either manually or automatically by Openshift) • Ensure that containers use trusted base images • Ensure unnecessary packages are not installed in the container • Ensure images are scanned and rebuilt to include security patches • Ensure Content trust for Docker is Enabled (https://docs.docker.com/engine/security/trust/content_t • Ensure setuid and setgid permissions are removed in the images • Ensure secrets are not stored in Dockerfiles • Ensure verified packages only are installed • Setuid and Setgid binaries should also be removed from images, lessening the chance of privilege • Container processes run as non-privileged USER

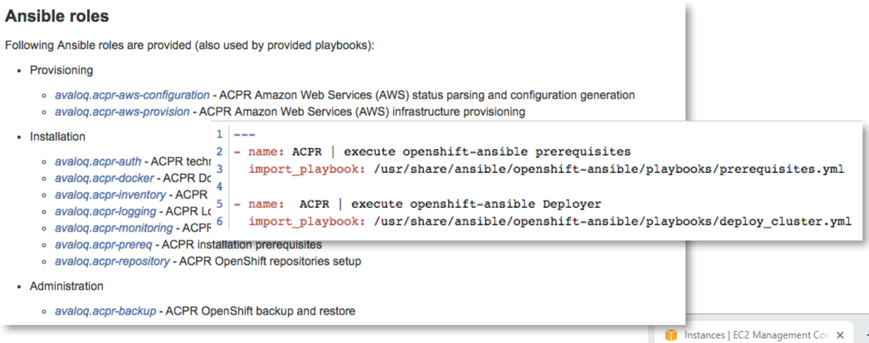


```
string sInput;
int iLength, iN;
double dblTemp;
bool again = true;

while (again) {
    iN = -1;
    again = false;
    getline(cin, sInput);
    system("cls");
    stringstream(sInput) >> dblTemp;
    iLength = sInput.length();
    if (iLength < 4) {
        again = true;
        continue;
    } else if (sInput[iLength - 3] != '.') {
        again = true;
        continue;
    } while (++iN < iLength) {
        if (isdigit(sInput[iN])) {
            continue;
        } else if (iN == (iLength - 3) ) {
            inue;
```

Code

Example deployment



Ansible roles

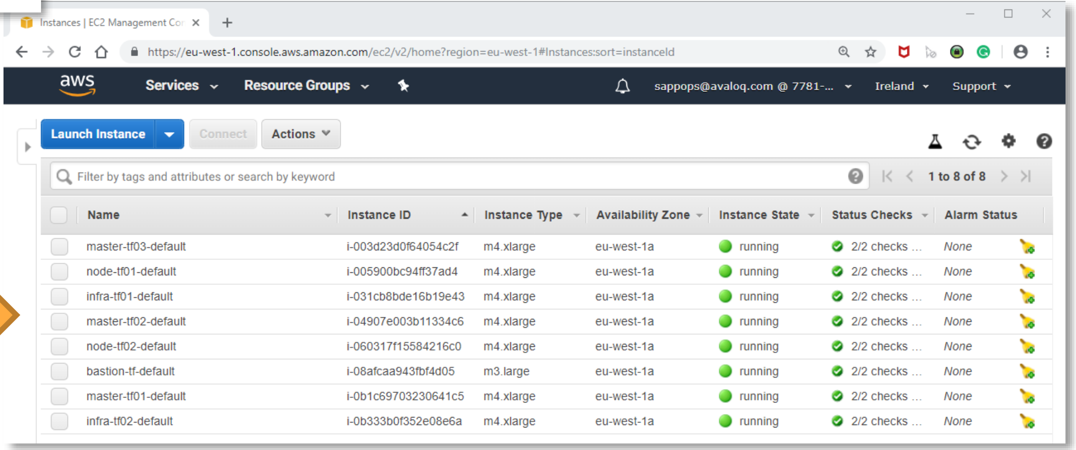
Following Ansible roles are provided (also used by provided playbooks):

- Provisioning
  - *avaloq.acpr-aws-configuration* - ACPR Amazon Web Services (AWS) status parsing and configuration generation
  - *avaloq.acpr-aws-provision* - ACPR Amazon Web Services (AWS) infrastructure provisioning
- Installation
  - *avaloq.acpr-auth* - ACPR techn
  - *avaloq.acpr-docker* - ACPR Dc
  - *avaloq.acpr-inventory* - ACPR
  - *avaloq.acpr-logging* - ACPR Lc
  - *avaloq.acpr-monitoring* - ACPF
  - *avaloq.acpr-prereq* - ACPR installation prerequisites
  - *avaloq.acpr-repository* - ACPR OpenShift repositories setup
- Administration
  - *avaloq.acpr-backup* - ACPR OpenShift backup and restore

```yaml
---
- name: ACPR | execute openshift-ansible prerequisites
  import_playbook: /usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml

- name:  ACPR | execute openshift-ansible Deployer
  import_playbook: /usr/share/ansible/openshift-ansible/playbooks/deploy_cluster.yml
```

RED HAT ANSIBLE



Instances | EC2 Management Co
https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#Instances:sort=instanceId
aws
Services
Resource Groups
sappops@avaloq.com @ 7781-...
Ireland
Support
Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 8 of 8

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
□	master-tf03-default	i-003d23d0f64054c2f	m4.xlarge	eu-west-1a	● running	✓ 2/2 checks ...	*None*
□	node-tf01-default	i-005900bc94ff37ad4	m4.xlarge	eu-west-1a	● running	✓ 2/2 checks ...	*None*
□	infra-tf01-default	i-031cb8bde16b19e43	m4.xlarge	eu-west-1a	● running	✓ 2/2 checks ...	*None*
□	master-tf02-default	i-04907e003b11334c6	m4.xlarge	eu-west-1a	● running	✓ 2/2 checks ...	*None*
□	node-tf02-default	i-060317f15584216c0	m4.xlarge	eu-west-1a	● running	✓ 2/2 checks ...	*None*
□	bastion-tf-default	i-08afcaa943fbf4d05	m3.large	eu-west-1a	● running	✓ 2/2 checks ...	*None*
□	master-tf01-default	i-0b1c69703230641c5	m4.xlarge	eu-west-1a	● running	✓ 2/2 checks ...	*None*
□	infra-tf02-default	i-0b333b0f352e08e6a	m4.xlarge	eu-west-1a	● running	✓ 2/2 checks ...	*None*

Recap

Call to action

- Q&A
- We are looking for feedback and others interested in similar challenges
- Open sourcing is not yet an option for Avaloq
- Who can recommend a format for collaboration?

RED HAT SUMMIT

THANK YOU

avaloq

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