

Rucio deployment

Radu Carpa

17 October 2023

About myself

- **Rucio core developer**
 - Mostly working on transfer and deletion workflows
- **Also, in charge of the ATLAS Rucio Kubernetes installation**

Ways to deploy Rucio

- **Directly via `pip install`**
- **Containers (provided by the Rucio core team)**
- **Helm charts on Kubernetes (recommended way for production deployments)**

ATLAS Rucio installation

- **Running in Kubernetes: 1 (small) integration + 3 production clusters (50 nodes in total)**
 - Required capacity: ~ ½ of that. The rest is for comfortable rolling re-installs of clusters
- **Configuration management via Flux2**
- **Self-managed load-balancer (haproxy 2.7.10) on puppet-managed VMs**
 - Still hoping to get rid of them

Why multiple clusters?

To increase our agility and reduce impact of risky changes:

- **Upgrading Kubernetes clusters**
- **Testing new versions of dependencies**
- **Less likely to be impacted by issues on CERN IT side (load balancer problems; removal of clusters)**

GitOps for ATLAS Rucio Operation

- **Many layers of templating engines***
 - Flux2
 - Sops (to store encrypted secrets in git)
 - Kustomize (required to use sops in flux2, but also used for rucio hot-patching)
 - Helm (managing the Rucio installation and containers)

* we are afraid Leonardo DiCaprio will have to come and save us from the **TemplateInception**

kubectl kustomize via flux

Better multi-cluster handling

Store encrypted secrets in the git repository

Reduce repetition

```
Project
├── releases
│   ├── atlas-rucio-int-01
│   ├── atlas-rucio-int-02
│   └── atlas-rucio-prod-01
│       └── kustomization.yaml
│   ├── atlas-rucio-prod-02
│   └── atlas-rucio-prod-03
├── base
│   ├── grafana-dashboards
│   ├── secrets
│   ├── configuration_rucio_helm_release.yaml
│   ├── daemonset-memcached.yaml
│   ├── kustomization.yaml
│   ├── prometheusrule-monit-forward.yaml
│   ├── rucio-charts.yaml
│   └── rucio-namespace.yaml
├── integration
└── production
    ├── common-includes
    ├── daemonprod.yaml
    ├── kustomization.yaml
    ├── serverprod_common_rucio.cfg
    ├── serverprodauth.yaml
    ├── serverprodpandawriter.yaml
    ├── serverprodtracer.yaml
    └── serverprodwriter.yaml
.gitlab-ci.yml
README.md
```

```
atlas-rucio-prod-01/kustomization.yaml
1 apiVersion: kustomize.config.k8s.io/v1alpha1
2 kind: Kustomization
3 resources:
4   - ../production
5
6 # Patch helm releases to use CERN LBAA
7 patches:
8   - target:
9       name: serverprodauth
10      kind: HelmRelease
11    patch: |-
12      apiVersion: helm.toolkit.fluxcd.io/v1
13      kind: HelmRelease
14      metadata:
15        name: serverprodauth
16      spec:
17        values:
18          service:
19            protocol: TCP
20            allocateLoadBalancerNodePorts: true
21            loadBalancerClass: null
22          annotations:
23            service.beta.kubernetes.io/aws-load-balancer-ssl-cert: arn:aws:iam::123456789012:server-certificate/arn:aws:iam::123456789012:server-certificate
24            loadbalancer.openstack.org/ssl-cert: arn:aws:iam::123456789012:server-certificate/arn:aws:iam::123456789012:server-certificate
25
26   - target:
27       name: serverprodpandawriter
```

```
production/kustomization.yaml
1 apiVersion: kustomize.config.k8s.io/v1alpha1
2 kind: Kustomization
3 resources:
4   - ../base
5   - common-includes
6   - daemonprod.yaml
7   - serverprodpandawriter.yaml
8   - serverprodwriter.yaml
9   - serverprodtracer.yaml
10  - serverprodauth.yaml
```

```
base/kustomization.yaml
1 apiVersion: kustomize.config.k8s.io/v1alpha1
2 kind: Kustomization
3
4 configurations:
5   - configuration_rucio_helm_release.yaml
6
7 resources:
8   - ../grafana-dashboards
9   - ../secrets
10  - daemonset-memcached.yaml
11  - rucio-namespace.yaml
```

Helm via flux

Allows to manage any helm-based component (most of k8s world).

```
daemonprod.yaml x      rucio-charts.yaml x
1  apiVersion: helm.toolkit.fluxcd.io/v2beta2  ✓ 45 ^ v 1  apiVersion: source.toolkit.fluxcd.io/v1beta1
2  kind: HelmRelease 2  kind: HelmRepository
3  metadata: 3  metadata:
4    name: daemonprod 4    name: rucio-charts
5    namespace: rucio 5    namespace: rucio
6  spec: 6  spec:
7    releaseName: daemonprod 7    interval: 10m
8    interval: 5m 8    url: https://rucio.github.io/helm-charts/
9    chart: 9
10   spec:
11     sourceRef:
12       kind: HelmRepository
13       name: rucio-charts
14       chart: rucio-daemons
15       version: 32.0.0
16   valuesFrom:
17     - kind: Secret
18       name: db-secret-daemons
19   values:
20     automatixCount: 1
21     conveyorTransferSubmitterCount: 1
22     conveyorPollerCount: 1
23
24     automatix:
25       sleepTime: 180
26       threads: 1
```


GitOps examples

1. **managing the Rucio installation**
2. **hot-patching Rucio** <https://rucio.cern.ch/documentation/operator/administration/>
3. **storing encrypted secrets in Git**
4. **applying a change only on one cluster**

Observability

Internal cluster metrics using kube-prometheus-stack

Cross-cluster aggregation via thanos

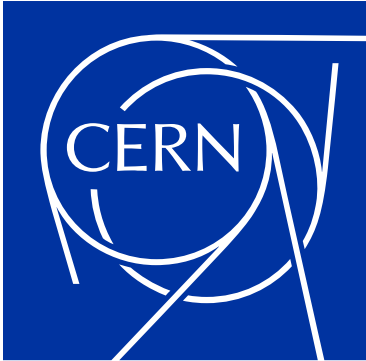
Alerts via alertmanager

Logs collected by filebeat and sent to CERN monit

Did you know about the following configuration option?

```
1 [common]
2 logjson = True
3
```

- ▼ infrastructure
 - > atlas-rucio-int-01
 - > atlas-rucio-int-02
 - > atlas-rucio-prod-01
 - > atlas-rucio-prod-02
 - > atlas-rucio-prod-03
- ▼ base
 - > deploy-igtf-ca
 - > inject-ssh-keys
 - ⊗ bgp-config.yaml
 - ⊗ helmrelease-filebeat.yaml
 - ⊗ helmrelease-ingress-nginx.yaml
 - ⊗ helmrelease-kube-prometheus-stack.yaml
 - ⊗ helmrelease-oauth2-proxy.yaml
 - ⊗ helmrelease-prometheus-adapter.yaml
 - ⊗ helmrelease-pushgateway.yaml
 - ⊗ helmrelease-thanos.yaml
 - ⊗ helmrelease-x509-certificate-exporter.yaml
 - ⊗ helmrepository.yaml
 - ⊗ kustomization.yaml
 - ⊗ pvc-thanos.yaml
 - ⊗ secret-alertmanager-receivers.yaml
 - ⊗ secret-cern-monit-credentials.yaml
 - ⊗ secret-grafana-credentials.yaml
 - ⊗ secret-thanos-objstore.yaml



home.cern