

GitOps for Web Frameworks

ARGOCD, GITLAB CI, CUSTOM OPERATORS

JACK HENSCHEL

KONSTANTINOS SAMARAS-TSAKIRIS

IT - CDA - WF

WHAT IS GITOPS?

At its core, GitOps maintains that a system must have its desired state expressed **declaratively**, and that the desired state is **versioned** and **immutable**, **pulled automatically**, and **continuously reconciled**.

- <https://www.redhat.com/en/blog/argocd-and-gitops-whats-next>

WHY GITOPS?

- **Infrastructure-as-Code**
 - Industry best-practice
 - Useful for managing complex infrastructure across multiple environments
- **Automated deployments**
 - Can be done by any privileged team member
 - Less room for human error
- **Safety & stability** thanks to continuous reconciliation
 - ArgoCD ensures all resources are in the desired state, if not: **alerts**

ANECDOTE: MISSING SSH KEYS FOR CLUSTER ACCESS

webservices > webframeworks-planning > Issues > #901

Closed Created 1 month ago by **Jack Henschel** Maintainer

Reopen issue

PaaS Cluster SSH_PRIVATE_KEY is incorrect

The `SSH_PRIVATE_KEY` stored in the Gitlab CI/CD variables does not work for the PaaS cluster.

Unknown cause and we need to figure out whether someone still has a working key.



Jack Henschel @jhensche added **Area: operations** **Known issues** **P2** **Project: PaaS** labels 1 month ago



Jack Henschel @jhensche · 1 month ago

Author

Maintainer



There are online resources that say it is possible to extract the key from `machineconfig/99-master-ssh`, but we don't have that resource in our clusters.

<https://access.redhat.com/solutions/3868301>

okd

Administrator

Project: default

Home

Overview

Projects

Search

API Explorer

Events

Operators

Workloads

Networking

Storage

Builds

Observe

Compute

MC 01-worker-kubelet

MC 99-master-afterburn-hostname

MC 99-master-generated-registries

MC 99-master-okd-extensions

MC 99-master-ssh

MC 99-master-ssh

MC 99-okd-master-disable-mitigations

MC 99-okd-worker-disable-mitigations

MC 99-worker-afterburn-hostname

MC 99-worker-generated-kubelet

MC 99-worker-generated-registries

MC 99-worker-okd-extensions

MC 99-worker-ssh

ARGOCD @ CERN OPENSIFT

STATUS QUO FOR OKD4

ArgoCD is used to

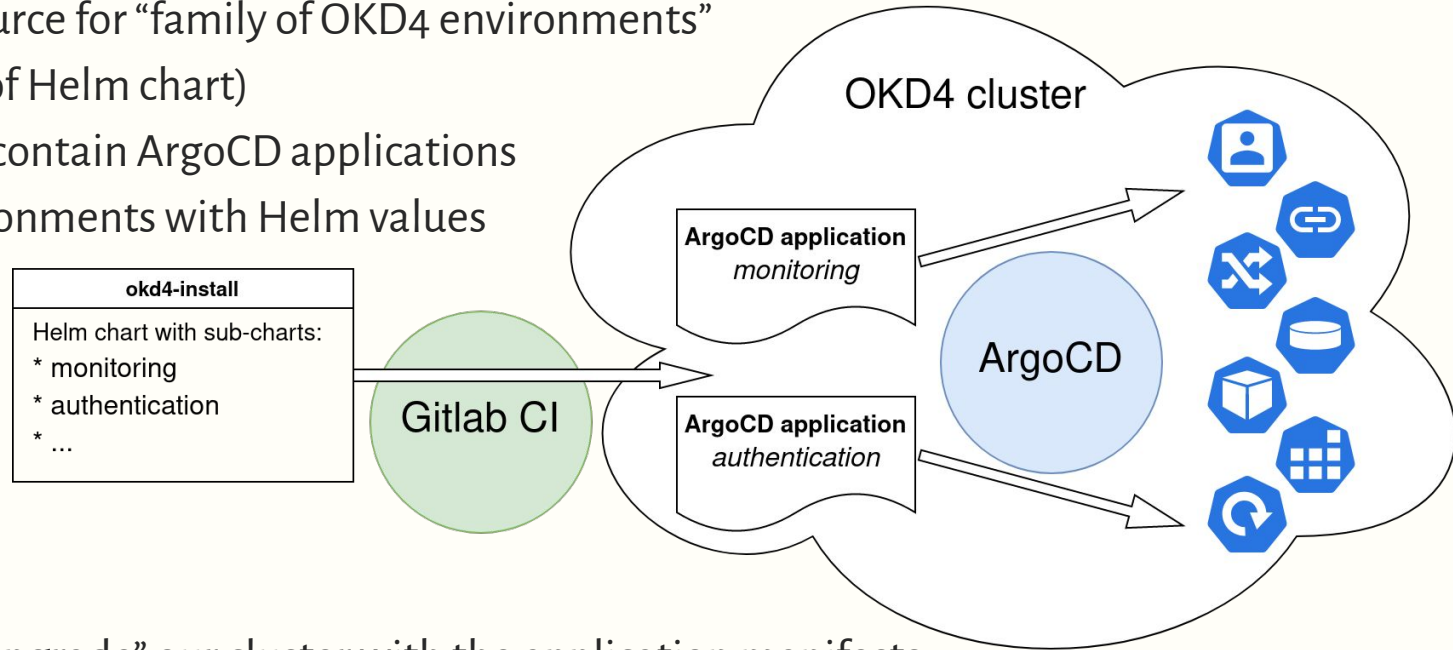
- Inject configuration for Openshift operators
- Install base components shared across all clusters (e.g. cloud-controller, logging, CERN authentication)
- Install specific components according to cluster “flavor”:
 - Drupal: DrupalSite, TektonCD and DBOD operators
 - PaaS: namespace permissions for users, EOS and CVMFS mounts etc.
 - App-Catalogue: Wordpress, Grafana and Nexus operators
 - WebEOS: WebEOS and Gitlab Pages operators

ArgoCD is **not** used to manage user applications!

OKD4 DEPLOYMENT

“Okd4-install” is our central configuration repository:

- Canonical source for “family of OKD4 environments”
(in the form of Helm chart)
- Helm charts contain ArgoCD applications
- Specify environments with Helm values



We “helm install/upgrade” our cluster with the application manifests

Then ArgoCD takes over and ensures the cluster is in the described state

DEPLOYING CHANGES TO CLUSTER COMPONENTS

1. MR on okd4-install: change a dependency X, update ArgoCD Application
 2. MR gets merged and commit lands on master:
 - No continuous deployment
 3. Run CI pipeline to deploy: `helm upgrade` our okd4-install chart
 4. Changes to applications are picked up by ArgoCD and synchronized
 5. Sometimes, *manual pre/post deployment* steps are required
- => “mostly” continuous delivery

Pre-deployment action:

- ☐ add `cern-accounts-integration.authzApiCredentials.XXX` to all Helm values files in Gitlab CI variables
 - ☒ HELM_SECRETS_FILE
 - ☒ HELM_VALUES_FILE app-catalogue
 - ☒ HELM_VALUES_FILE paas
 - ☒ HELM_VALUES_FILE drupal-stg
 - ☒ HELM_VALUES_FILE paas-stg
 - ☒ HELM_VALUES_FILE drupal
 - ☒ HELM_VALUES_FILE app-cat-stg
 - ☒ HELM_VALUES_FILE webeos-stg
 - ☒ HELM_VALUES_FILE webeos
 - ☐ HELM_VALUES_FILE webeos-proto (uses HELM_SECRETS_FILE)

Post-deployment action:

remove ldap sync labels from group and add our custom label:

- ☒ webeos-proto
- ☐ webeos-stg
- ☐ webeos
- ☐ paas-stg
- ☐ paas
- ☐ app-cat-stg
- ☐ app-catalogue
- ☐ drupal-stg
- ☐ drupal

```
# make backup of all groups
oc get groups -o yaml > groups-{CLUSTER_NAME}.yaml

# for each group synchronized from ldap
for group in $(oc get group -l openshift.io/ldap.host -o name); do
  # remove labels and annotations from ldap sync
  oc annotate "$group" openshift.io/ldap.sync-time- openshift.io/ldap.uid- openshift.io/ldap.url-
  oc label "$group" openshift.io/ldap.host-

  # add our annotation
  oc label "$group" okd.cern.ch/sync.host=cern-authz-api
done
```

“MOSTLY” CONTINUOUS DELIVERY

Instead of ArgoCD “pulling” changes from master, we “helm upgrade” weekly:

- Mainly: controlled releases (with manual pre-/post-deployment steps)
- Also: currently not solution for storing secrets in Git
- But: updates are deployable at any time (Gitlab CI)

SECRETS MANAGEMENT

- For each environment, Helm values are split into two files:
 - “Public” configuration: stored as YAML in okd4-install repository
 - “Sensitive” configuration: stored as Gitlab CI variables
- This is undesirable, because we cannot manage secrets with version control!
 - No change history
 - Potential for human error, *hard to recover*
 - Weak protection, just “hidden”

CUSTOM GITOPS

DRUPAL NEEDS CD

CERN Drupal Distribution

- Support multiple versions of Drupal concurrently
- Constant small updates
 - Versions → **release streams**

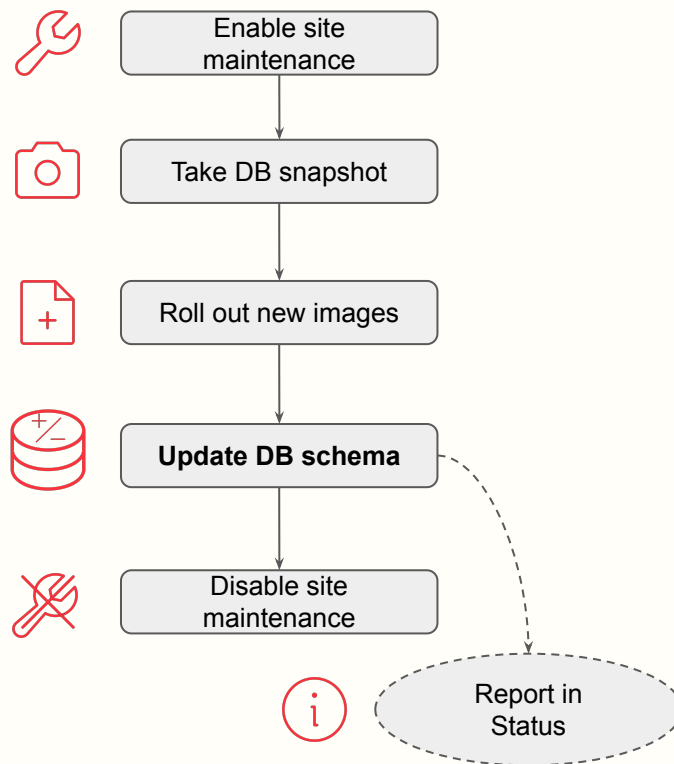
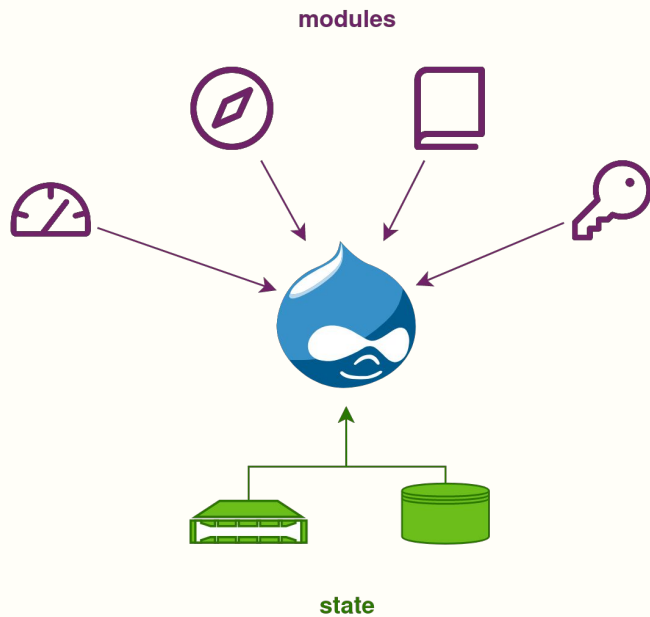
versions →

v9.2-1	v9.3-1
RELEASE-2022.02 .22T10-10-20Z	RELEASE-2022.03 .29T15-12-20Z
RELEASE-2022.03 .22T18-10-20Z	RELEASE-2022.04 .20T10-12-20Z
RELEASE-2022.04 .22T11-10-20Z	RELEASE-2022.04 .22T12-12-20Z

We want to:

- Advertise latest release for each stream in-cluster
- Automatically update every Drupal site to its latest release

UPDATING DRUPAL

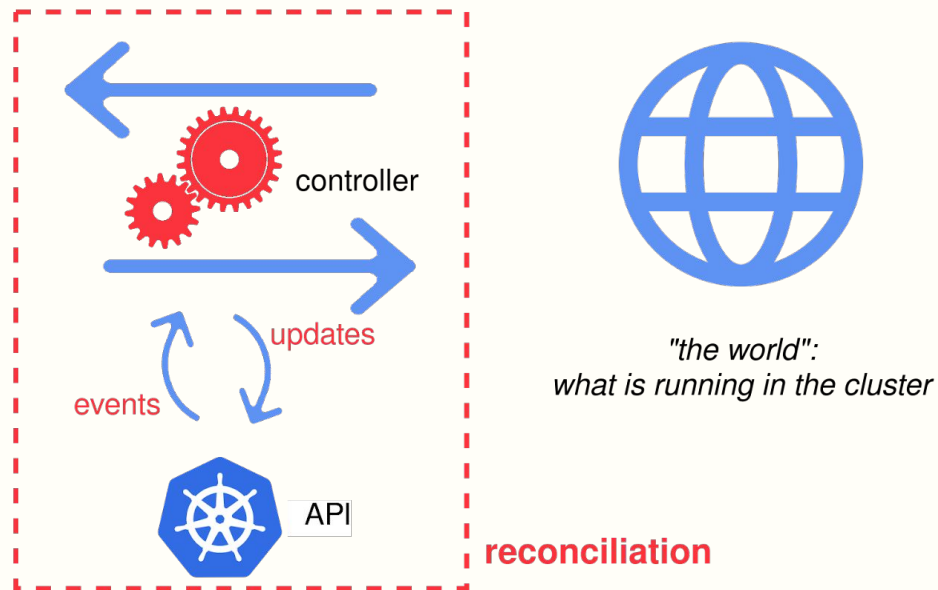
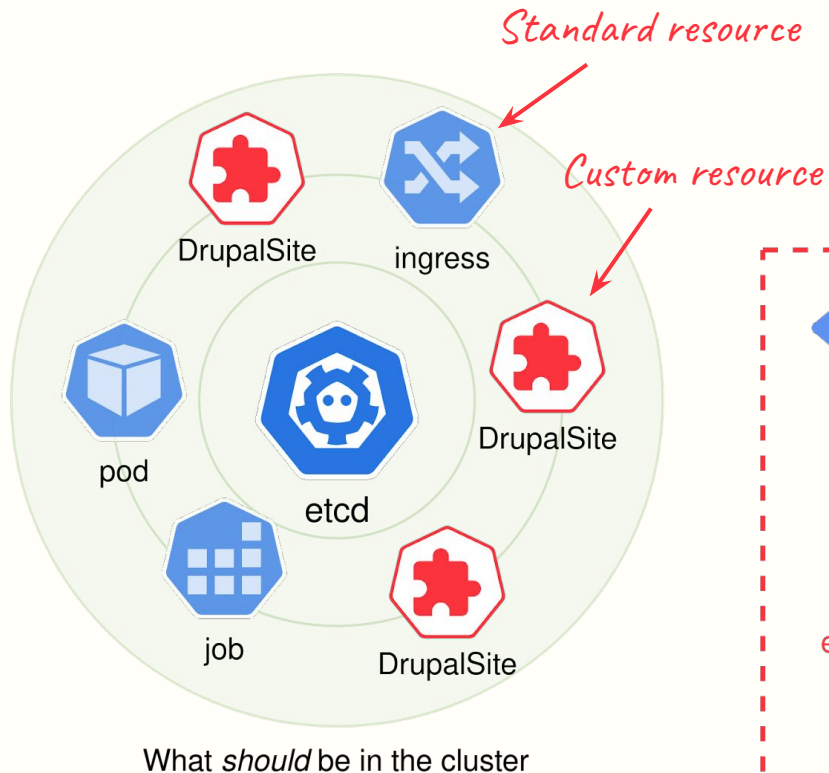


DRUPAL REPOSITORY

CERN Drupal Distribution supports a few release streams concurrently

- Each release stream is a Git branch
- Change by Merge Request
- CI: test and push release image

v9.2-1	v9.3-1
RELEASE-2022.02 .22T10-10-20Z	RELEASE-2022.03 .29T15-12-20Z
RELEASE-2022.03 .22T18-10-20Z	RELEASE-2022.04 .20T10-12-20Z
RELEASE-2022.04 .22T11-10-20Z	RELEASE-2022.04 .22T12-12-20Z




Operator

a *custom resource and controller*, conceptually similar to an OOP **class** with data & methods


CONTINUOUS DELIVERY WITH OPERATOR

Custom resources:

- DrupalSite CRD
 - Subscribe to release stream
- SupportedDrupalVersions CRD
 - List latest release for each stream
 - Poll image registry



```
apiVersion: drupal.webservices.cern.ch/v1alpha1
kind: DrupalSite
metadata:
  name: drupalsite-sample
spec:
  version:
    name: v9.3-1
```



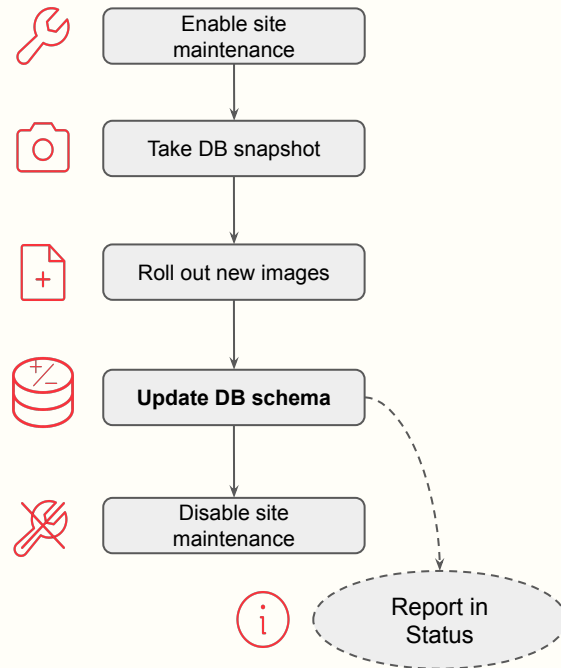
```
apiVersion: drupal.webservices.cern.ch/v1alpha1
kind: SupportedDrupalVersions
metadata:
  name: supporteddrupalversions
spec:
  blacklist:
    - v9.3-1
  defaultVersion: v9.3-2
status:
  versions:
    - name: v8.9-2
      releaseSpec:
        latest: RELEASE-2022.04.05T09-31-46Z
    - name: v9.3-2
      releaseSpec:
        latest: RELEASE-2022.04.22T12-12-20Z
```

CONTINUOUS DELIVERY WITH OPERATOR

DrupalSite controller

- Pick latest release
- Automatically update DrupalSite to latest release

```
apiVersion: drupal.webservices.cern.ch/v1alpha1
kind: DrupalSite
metadata:
  name: drupalsite-sample
spec:
  version:
    name: v9.3-1
status:
  releaseSpec: RELEASE-2022.04.11T16-47-53Z
  releaseID:
    current: v9.3-1-RELEASE-2022.04.11T16-47-53Z
    failsafe: v9.3-1-RELEASE-2022.04.11T16-47-53Z
```



WHAT WE'D LIKE TO SEE

- Native and well-supported approach for storing secrets *AND* accessing them from ArgoCD
 - Promising solution: SOPS + Vault => still missing a centrally-hosted, HA Vault setup
- Use ArgoCD ApplicationSet for deploying multiple related components
- For PaaS: evaluate if/how we can expose GitOps workflows for users
 - Currently, PaaS supports CD with kubectl/oc/Helm charts, but not fully-fledged GitOps with continuous reconciliation (ArgoCD/Flux)
- For Drupal: continuous deployment in production

Thanks for your attention!

Any questions?