
SciLifeLab

EXPLORANDO LA SEGURIDAD Y ORQUESTACIÓN EN LA NUBE CON KUBERNETES

ALVARO REVUELTA

SciLifeLab

WHO AM I?

Alvaro Revuelta.

- Alumni from this faculty - ETSIINF-UPM.
- working and living in Sweden.
- Working as a developer (Backend) in SciLifeLab.

SciLifeLab

SCILIFELAB

Laboratory for Life Sciences.

1. Data Centre - Focus on the development of IT solutions on behalf of research and innovation.
2. DDS - Data Delivery System - System for secure delivering of data (incl. sensitive) to researchers
3. All the projects are Open Source.

SciLifeLab

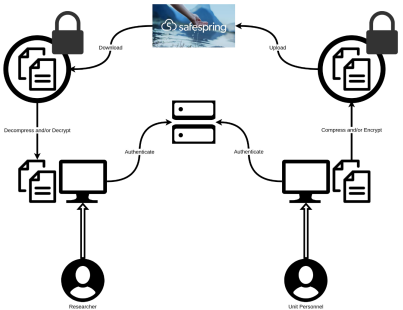
ESTRUCTURA

1. Introduction.
2. Orchestration in Kubernetes.
3. DevOps & GitOps.
4. Security in DDS.
5. Other projects in SciLifeLab.
6. Final - Questions.

SciLifeLab

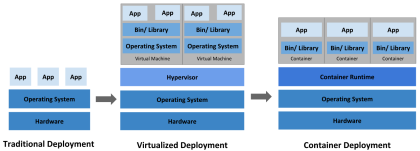
1. INTRODUCTION - DATA DELIVERY SYSTEM

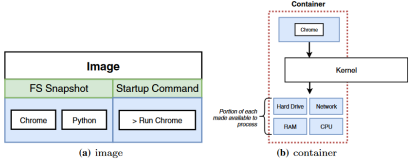
- The data is always kept on Sweden.
- The system has built-in encryption and key management.
- NOT a storage solution.



SciLifeLab

1. INTRODUCTION - CONTAINERS

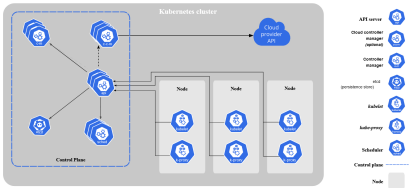




SciLifeLab

1. INTRODUCTION - KUBERNETES

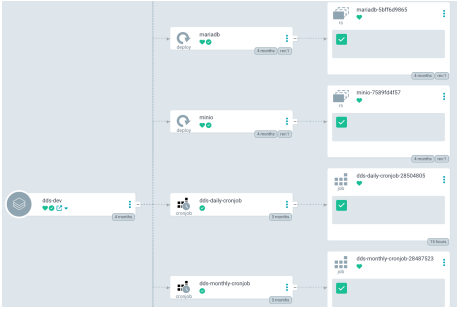
- Orchestration and management of containers.
- Declarative language (YAML files).



SciLifeLab

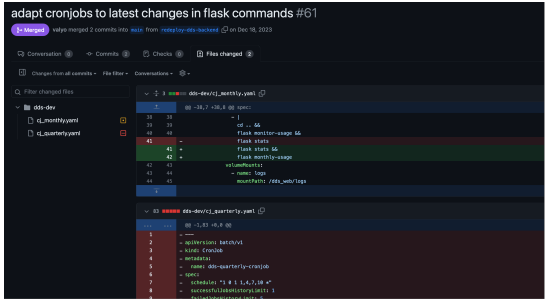
2. ORCHESTRATION IN KUBERNETES

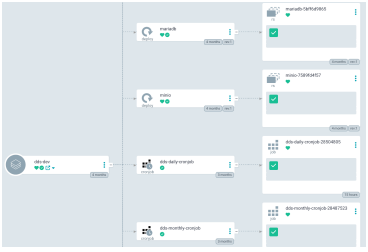
- **ArgoCD** is used as a GitOps tool for CD of K8s applications.
- It is connected to a repository (GitHub) and for each change, all the required resources are updated **automatically**.
- Motorization of alerts and errors with **OpenSearch** and resources with **Grafana**.



SciLifeLab

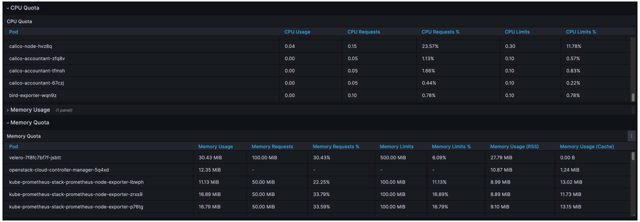
2. - ORCHESTRATION IN KUBERNETES





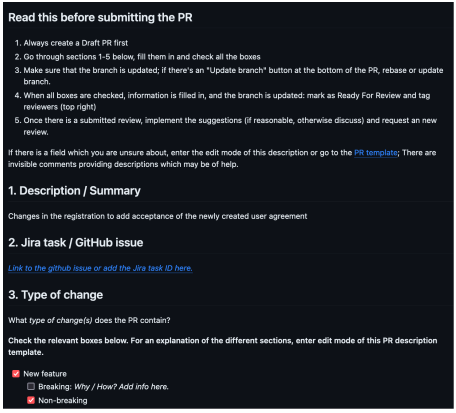
SciLifeLab

2. - ORCHESTRATION IN KUBERNETES



3. DEVOPS & GITOPS

- Branching Strategy: For every change, new **Branch** and **Pull Request** (PR)
- GitHub Actions: Automatic testing.
- You work on this new branch, which will be **deleted** after merge.



Read this before submitting the PR

1. Always create a Draft PR first
2. Go through sections 1-5 below, fill them in and check all the boxes
3. Make sure that the branch is updated; if there's an "Update branch" button at the bottom of the PR, rebase or update branch.
4. When all boxes are checked, information is filled in, and the branch is updated: mark as Ready For Review and tag reviewers (top right)
5. Once there is a submitted review, implement the suggestions (if reasonable, otherwise discuss) and request an new review.

If there is a field which you are unsure about, enter the edit mode of this description or go to the PR template; There are invisible comments providing descriptions which may be of help.

1. Description / Summary

Changes in the registration to add acceptance of the newly created user agreement

2. Jira task / GitHub issue

Link to the github issue or add the Jira task ID here.

3. Type of change

What *type of change(s)* does the PR contain?

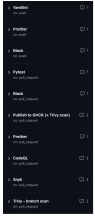
Check the relevant boxes below. For an explanation of the different sections, enter edit mode of this PR description template.

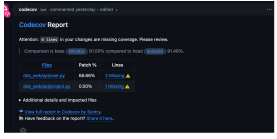
- [x] New feature
  - [ ] Breaking: *Why / How? Add info here.*
  - [x] Non-breaking

SciLifeLab

3. DEVOPS & GITOPS

- For every PR, a new suite of tests will be executed.
- If any of them fails, the PR will be **blocked**.

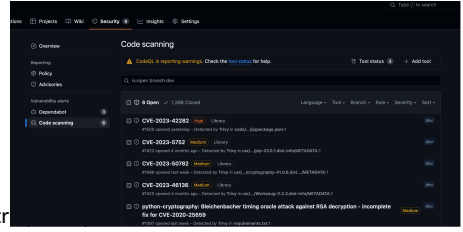




SciLifeLab

3. DEVOPS & GITOPS

- Some actions will only be executed **after** merging. Some of them will continouly scan for **security vulnerabilities**.
- At the last stage, a new image with the specified tag will be publish on **Docker Hub**.
  See dds_web/blob/dev/.github/workflows/publish_and_tr



SUMMARY SO FAR

Two repositories:

1. The main one, where the app code lives, it publishes the images with the dockerized application.
2. A second one connected with ArgoCD with contains the YAML for Kubernetes.

SciLifeLab

4. SECURITY AND CIPHERS

- Public-private key for each user, the private key is further ciphered with the password.
- When a new project is created, another key pair is generated. The project private key is then ciphered with the user public key.
- Therefore, the project private key can only be deciphered with the user private key, which is obtained thought the user password.

SciLifeLab

4. SECURITY AND CIPHERS.

- For the project keys, the algorithm used is **X25519** Diffie Helman Elliptic Curve - "Crypthopgraphy" Python package.
  Link
- The subsequent cipher of the private key is done with RSA.
  Link.

```python
def generate_project_key_pair(user, project):
    private_key = asymmetric.x25519.X25519PrivateKey.generate()
    private_key_bytes = private_key.private_bytes(
        encoding=serialization.Encoding.Raw,
        format=serialization.PrivateFormat.Raw,
        encryption_algorithm=serialization.NoEncryption(),
    )
    public_key_bytes = private_key.public_key().public_bytes(
        encoding=serialization.Encoding.Raw,
        format=serialization.PublicFormat.Raw,
    )
    project.public_key = public_key_bytes
    for unit_user in user.unit.users:
        __init_and_append_project_user_key(unit_user, project, private_key_bytes)
    del private_key_bytes
    del private_key
    gc.collect()
```

SciLifeLab

4. SECURITY AND CIPHERS.

- To access the application a **temporal token** has to be generated with **2FA**.
- First, the user sends their username and password. Then a code is sent to the email or authenticator app. Finally, a new token with a limited duration is created.
- In this case it is a **JWT** (JSON Web Token).

- The data is ciphered with the algorithm **ChaCha20-Poly1305** https://datatracker.ietf.org/doc/html/rfc7539

SciLifeLab

5. OTHER PROJECTS

IN SCILIFELAB

SciLifeLab

6. FINAL

CONTACT:
LINKEDIN - ALVARO REVUELTA MARTINEZ
ALVARO.REVUELTA@SCILIFELAB.UU.SE

SciLifeLab

6. FINAL

QUESTIONS

SciLifeLab