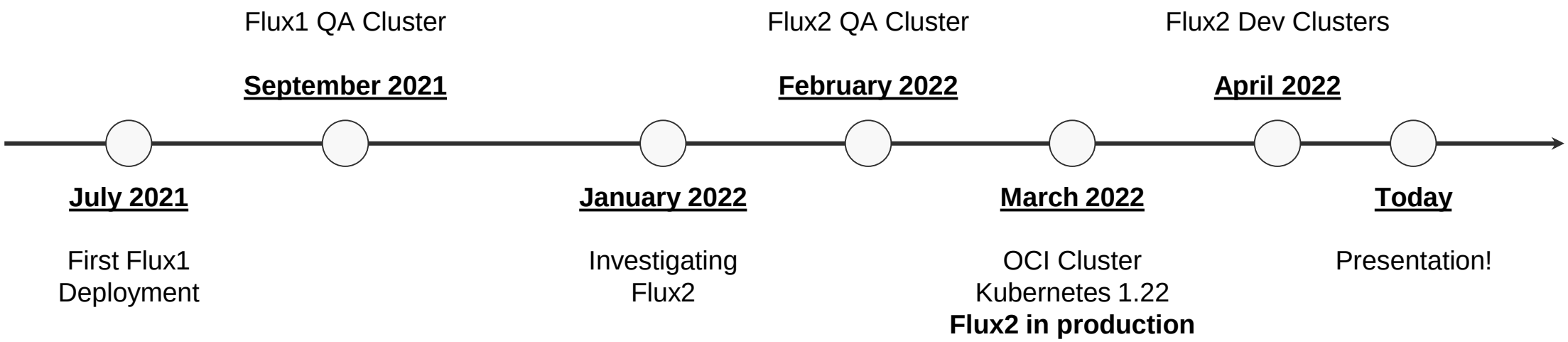
GitOps in MONIT

Luca Bello

27.04.2022

Our history with GitOps



- Over time, we started managing more and more services in Kubernetes
- The cluster growth made us face Flux1 limitations: poor multi-environment support, monolithic release configurations, no internal monitoring, weird interactions with Helm3, and more
- Flux2 solved all those issues, allowing for smooth and simple Kubernetes operations

Tools and resources

Flux2

- Repository as single source of truth
- Extremely flexible

Helm3

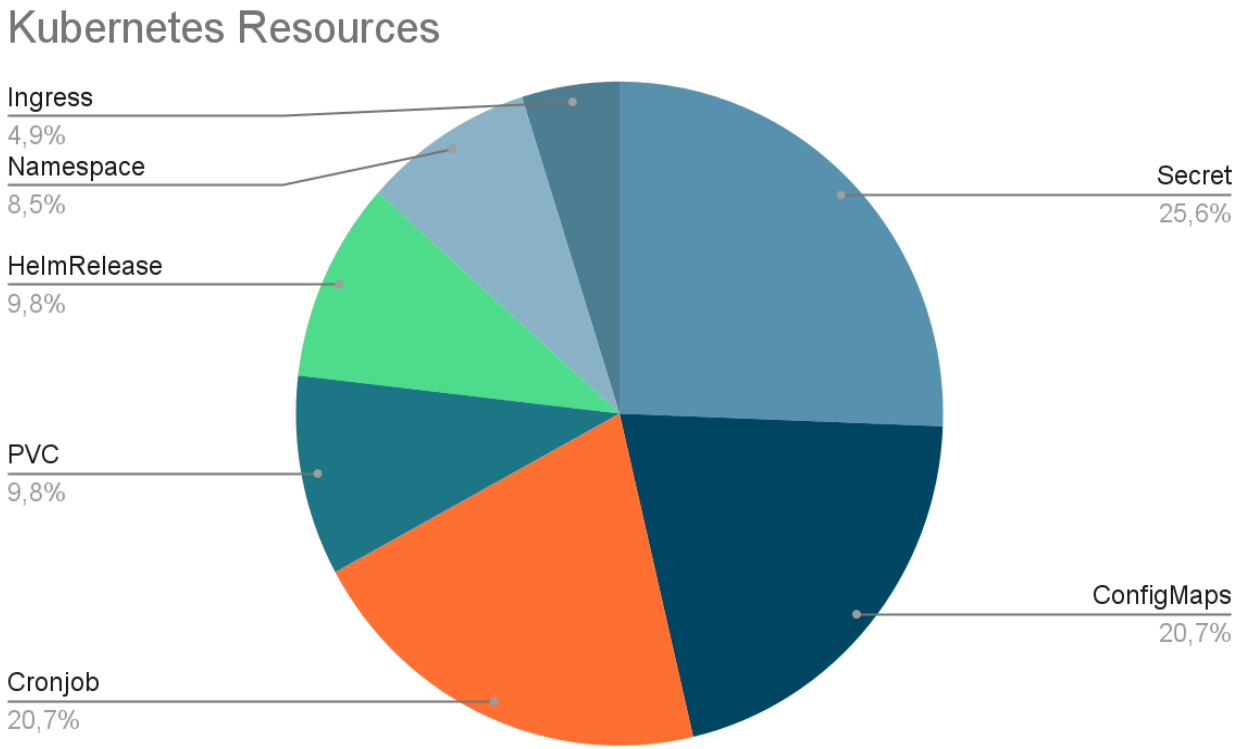
- Simple configurable deployments

GitLab CI

- YAML linting and validation
- Automatic QA rebase for remote-probes

SOPS

- Secrets encryption with AGE keys



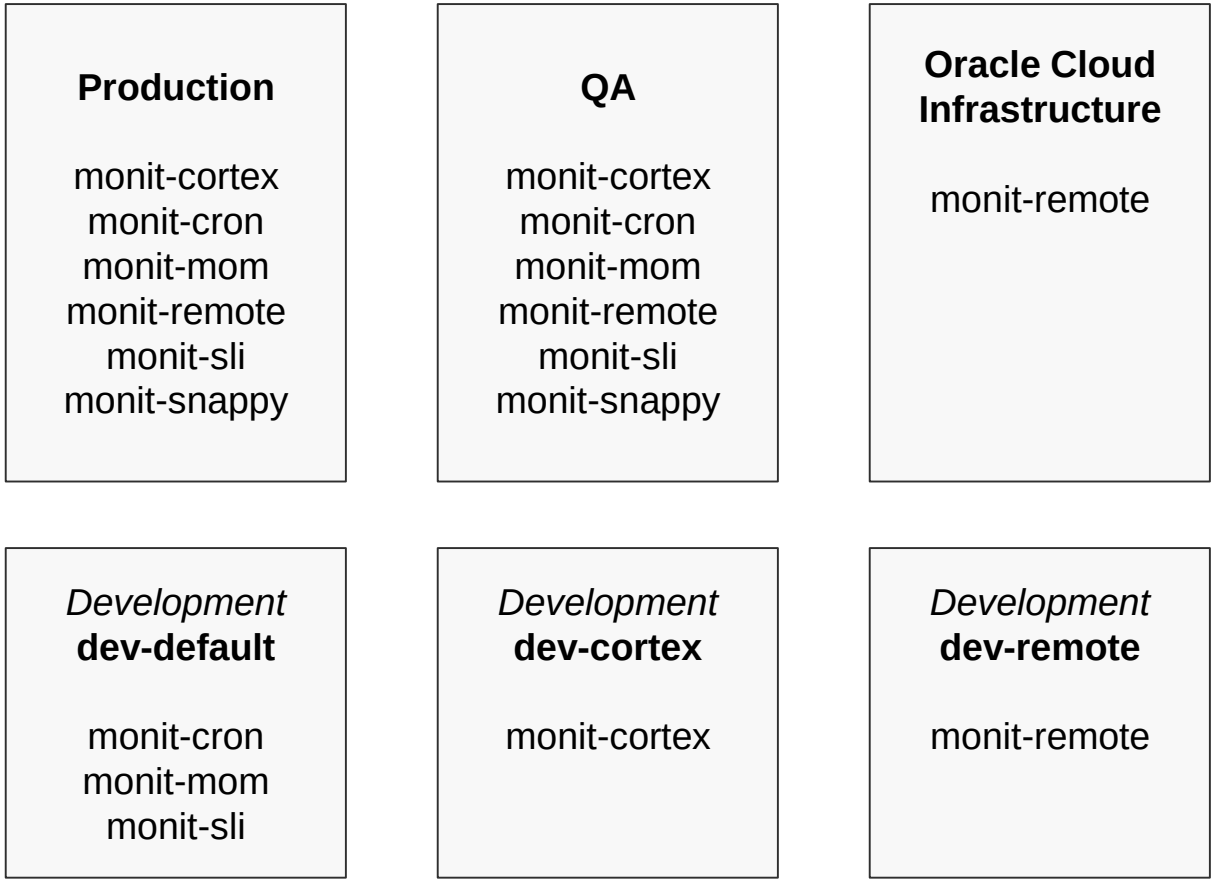
Clusters

Multiple clusters and environments

- Production, QA, Development, OCI
- Extremely flexible

Development environment in 3 clusters

- Dedicated cluster for bigger namespaces
- Better isolation for component changes
- Allows for cleaner release process



Flux2 Folder Structure

Things to deploy on Kubernetes

- HelmReleases
- Cronjobs
- Secrets
- Ingress
- PVCs
- PVs

Flux configuration for each cluster, and more cluster-specific resources

- Kustomizations
- Ingress

Common infrastructure tools that are necessary for apps

- Namespaces
- Helm Sources

apps/

- apps/base/
- apps/production/
- apps/qa/
- apps/dev/
- apps/oci/

clusters/

- clusters/base/
- clusters/production/
- cluster/qa/
- clusters/dev-default/
- clusters/dev-cortex/
- clusters/dev-remote/

infrastructure/

Flux2 Files

- **Flexible inner file structure**
  - Files are explicitly included in Kustomization resources
- **apps and infrastructure split by namespace**
  - Quick identification and localization of errors
  - Errors in a namespace don't stop the reconciliation for the others
  - Reconciliation can be suspended with finer granularity
- **Clear separation between default configurations and patches**
  - Higher degree of modularity which makes for a better development process
  - Makes our multi-environment setup very easy to manage

Releases

- **Helm based configurations are shared across environments**
  - Environment-specific values can be specified with patches
  - Our Cortex deployment shrank from 1500+ lines to only 60
- **Cronjobs patched by ConfigMaps**
  - ConfigMaps can be mounted to export environment variables

```yaml
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: environment-service-costing
  namespace: monit-cron
data:
  ENDPOINT_SERVICE_COSTING: ""
  PRODUCER_CHARGEGROUP_LOCATION: ""
```

```yaml
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: environment-service-costing
  namespace: monit-cron
data:
  ENDPOINT_SERVICE_COSTING: "accounting-receiver-dev.cern.ch"
  PRODUCER_CHARGEGROUP_LOCATION: "https://monit-docs.web.cern.c
```

Secrets

- **Secrets are managed via Mozilla's SOPS**
  - Allows to encrypt parts of a file so it can be safely uploaded to the repository
  - Identify secret fields with regular expressions
  - Flux2 automatically decrypts the files in the reconciliation process
  - Encryption using simple AGE keys, but lots of options are supported

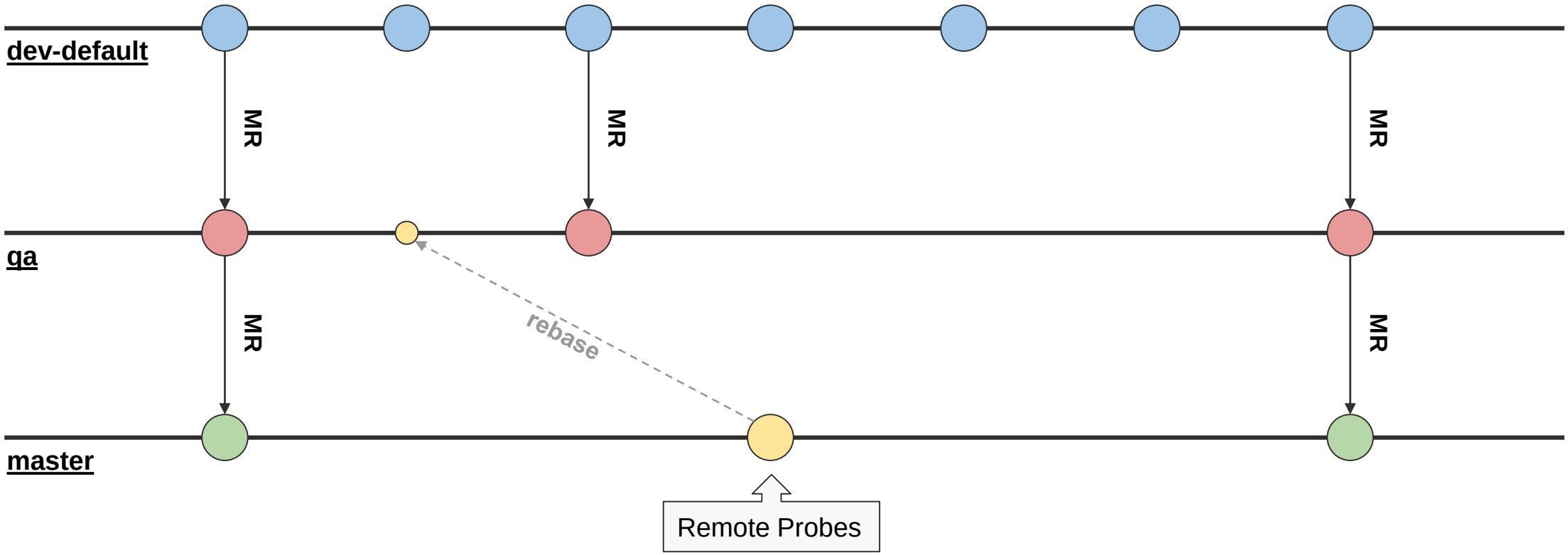
```
# created: 2022-04-26T17:22:56+02:00
# public key: age10xfr7yqsd8dpcw9kk73304sh3xlqs4xxhgymj26am0dxyw2t53msq3t078
AGE-SECRET-KEY-1LSL5W54Y3FE7H2R54QZAFM8WHM6CPXK3RZQALUM4Z9EVPDV6WQ0Q75TM9L
```

- **Higher modularity means we can encrypt only what's necessary**

.sops.yaml 1.16 KB

```
1  creation_rules:
2    - path_regex: monit-cortex/.*\.yaml
3      encrypted_regex: 'tls.key|tls.crt|access_
4      age: 'age158dwqkakd04p64c3j2mu2hxx5dsy8hx
5    - path_regex: monit-cron/.*\.yaml
6      encrypted_regex: 'ES_|GRF_|monitops\.keyt
7      age: 'age158dwqkakd04p64c3j2mu2hxx5dsy8hx
8    - path_regex: monit-mom/.*\.yaml
9      encrypted_regex: 'password|kafka_auth.key
10     age: 'age158dwqkakd04p64c3j2mu2hxx5dsy8hx
11   - path_regex: monit-remote/.*\.yaml
12     encrypted_regex: '\.dockerconfigjson|.*pa
13     age: 'age158dwqkakd04p64c3j2mu2hxx5dsy8hx
14   - path_regex: monit-sli/.*\.yaml
15     encrypted_regex: 'user|password|\.dockerc
16     age: 'age158dwqkakd04p64c3j2mu2hxx5dsy8hx
17   - path_regex: monit-snappy/.*\.yaml
18     encrypted_regex: '\.dockerconfigjson|tls.
19     age: 'age158dwqkakd04p64c3j2mu2hxx5dsy8hx
```

Release Process

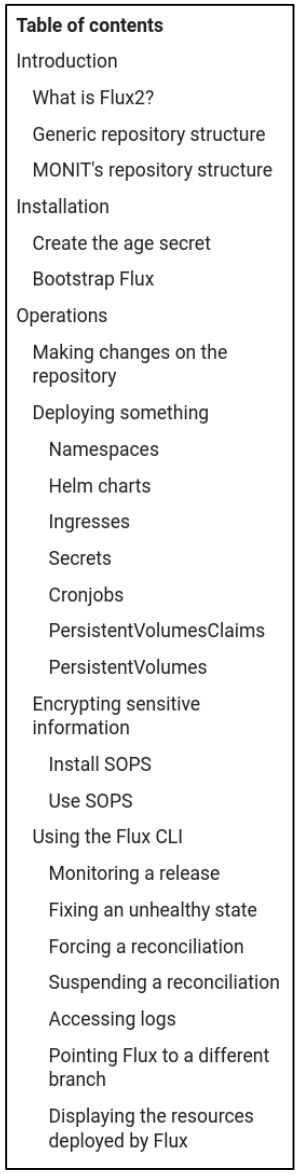


Continuous Integration

- **Automatic rebase of QA on master**
  - monit-remote-probes commits directly to master
  - Helps to maintain QA and master in sync
- **YAML linting and validation**
  - Helps to spot early mistakes and ensures style consistency
  - Possibility to perform validation against specific Helm chart schemas in the future

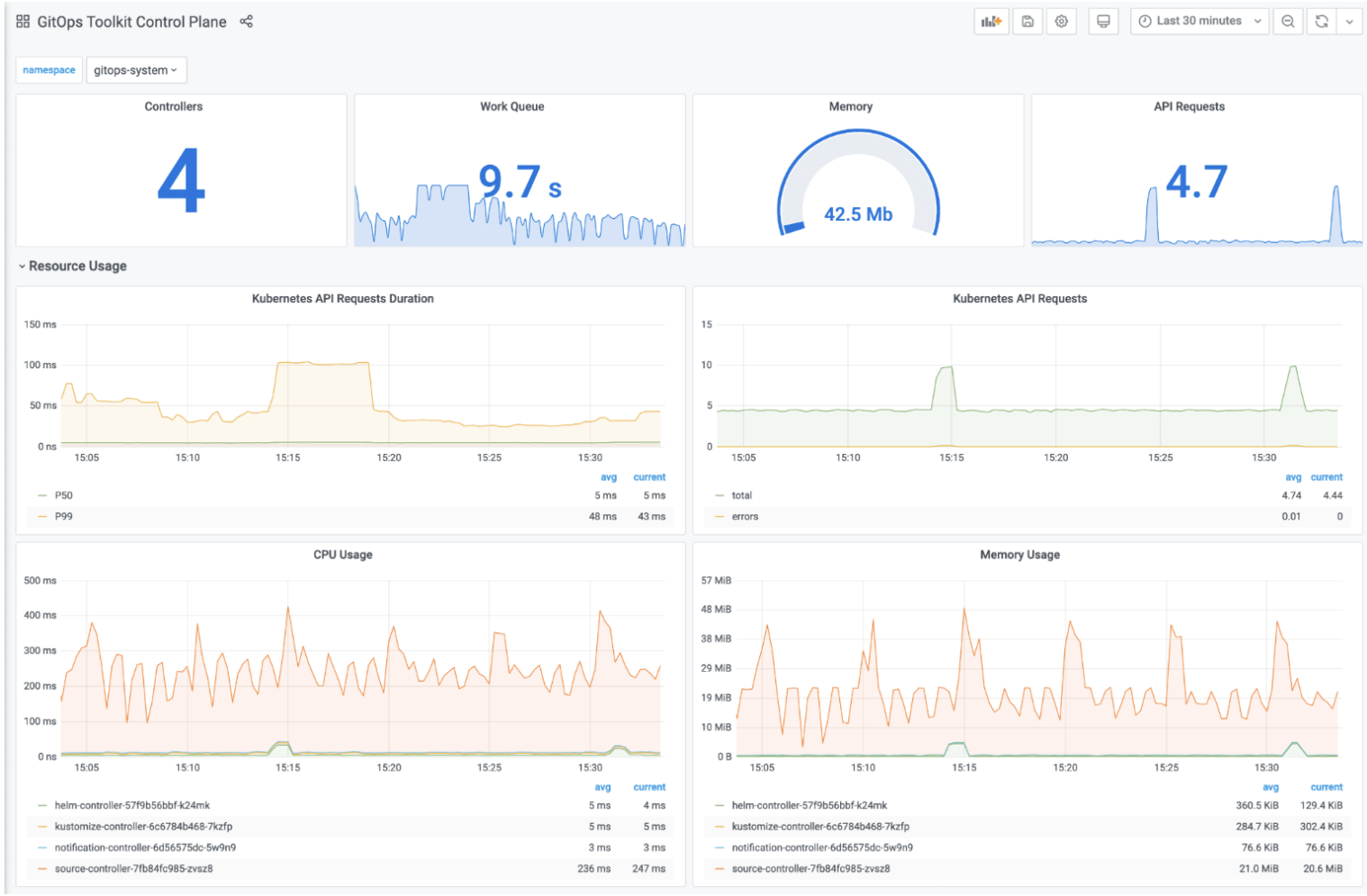
Documentation

- **Over time we wrote an extensive Flux2 documentation (3500+ words)**
  - Step-by-step guide and examples
  - Includes a troubleshooting section for common issues
- **Currently internal as it includes MONIT-specific details**
  - We would be happy to generalize it and make it available to everyone!



Future Plans

- **Flux2 internal monitoring**
  - Custom Grafana dashboards
  - Supported out-of-the-box
  - Useful to make sure Flux2 is always in a healthy state



What we are happy with

- **Easy and powerful multi-environment support**
- **Familiar development process with one single source of truth**
- **Explicit and declarative inclusion of resources**
- **Actual separation between namespaces when reconciling**
- **Modular resource configurations**
  - The files to maintain more frequently are very slim
- **Transparent integration with Helm3 and SOPS**
- **Internal monitoring with Grafana**

Q & A


CERN