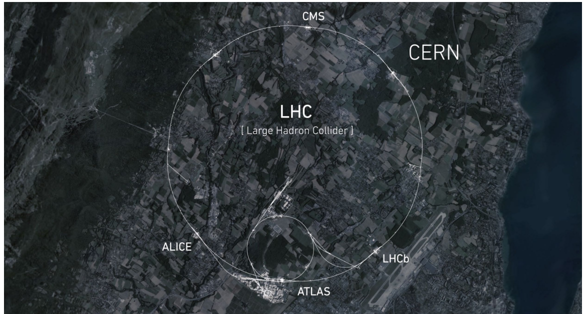
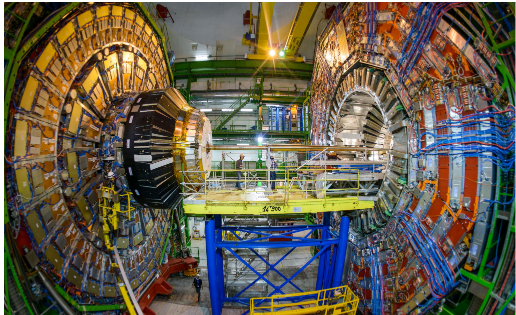
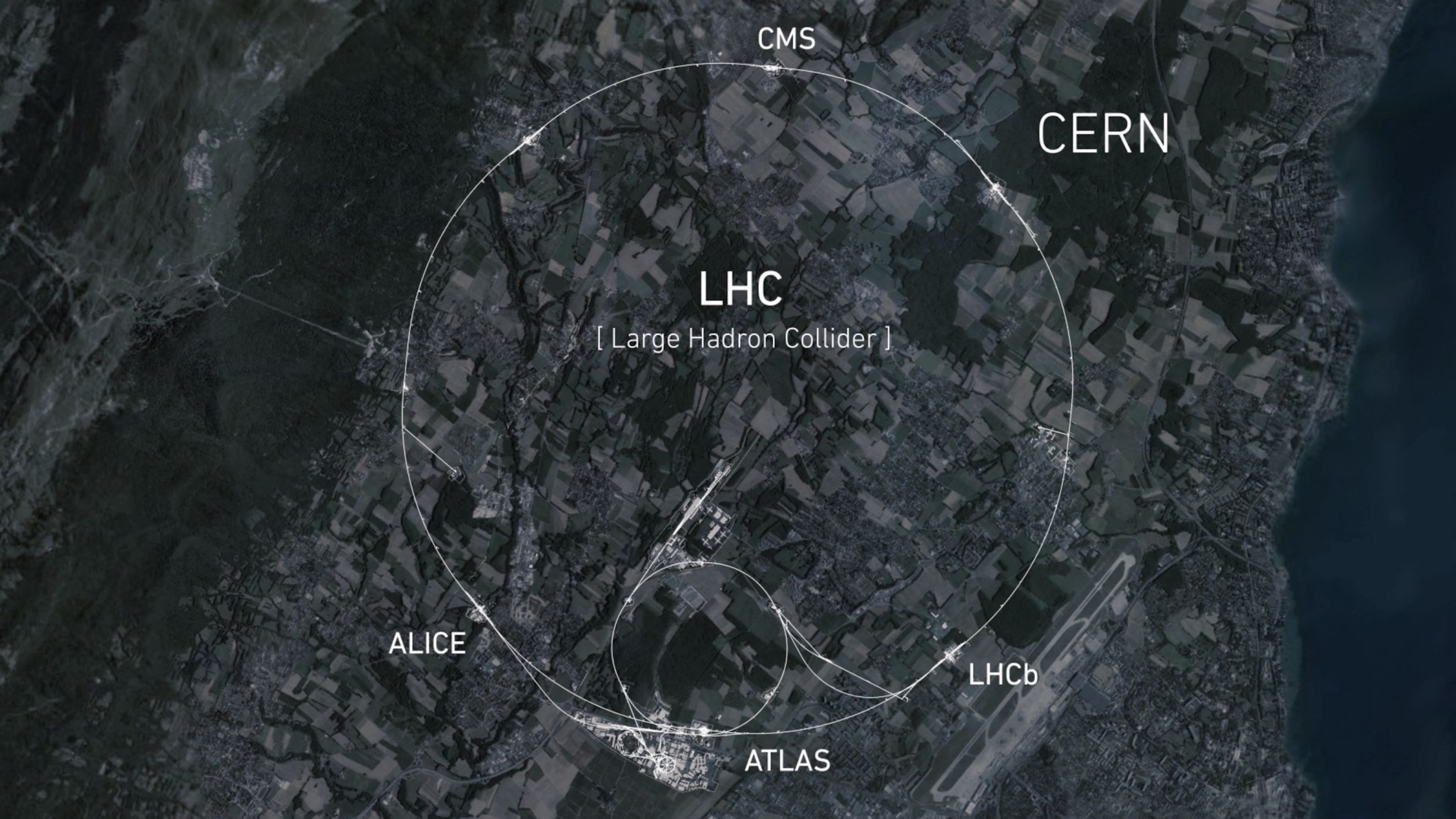
Managing Helm Deployments with GitOPS at CERN

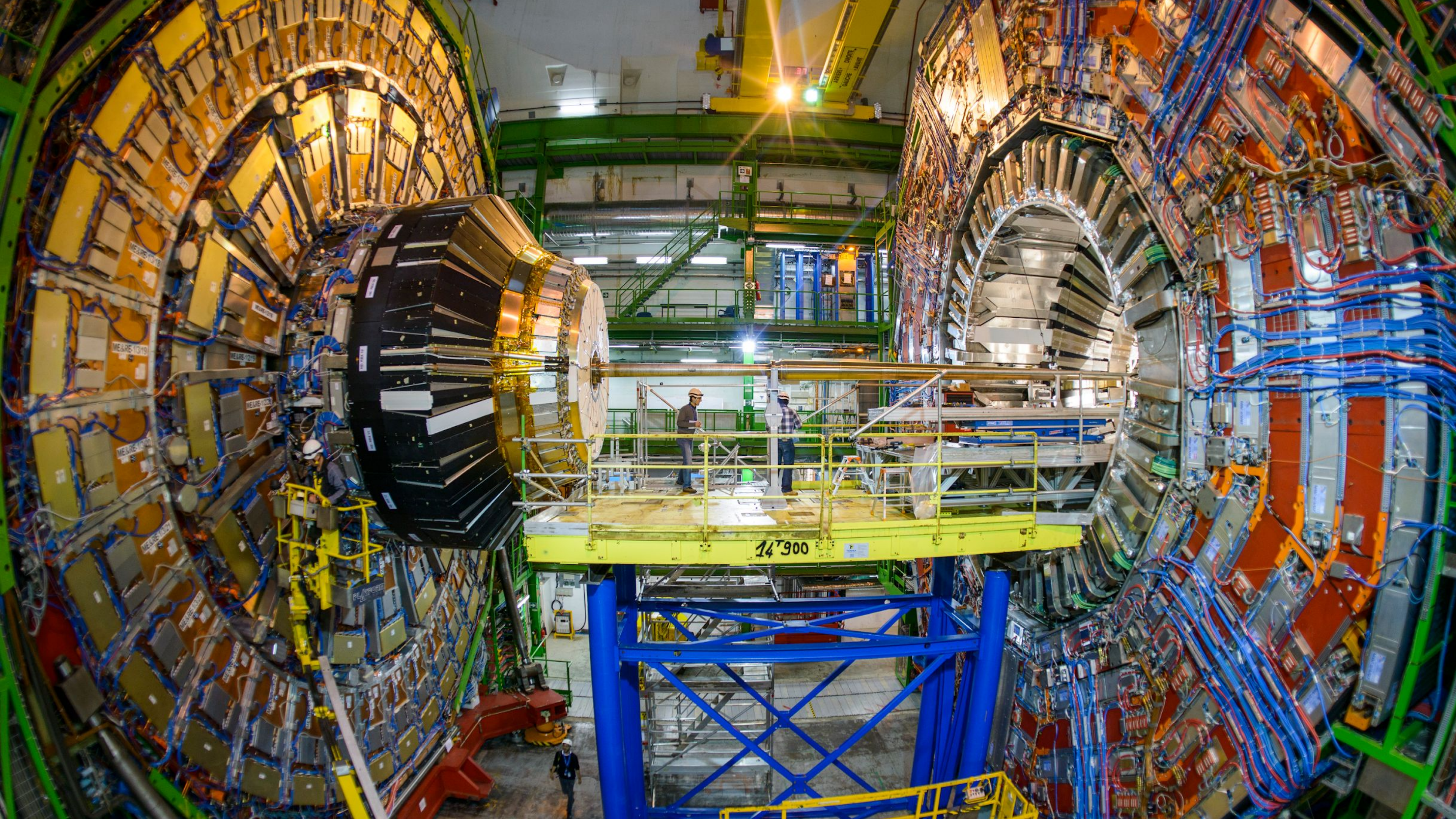
Ricardo Rocha

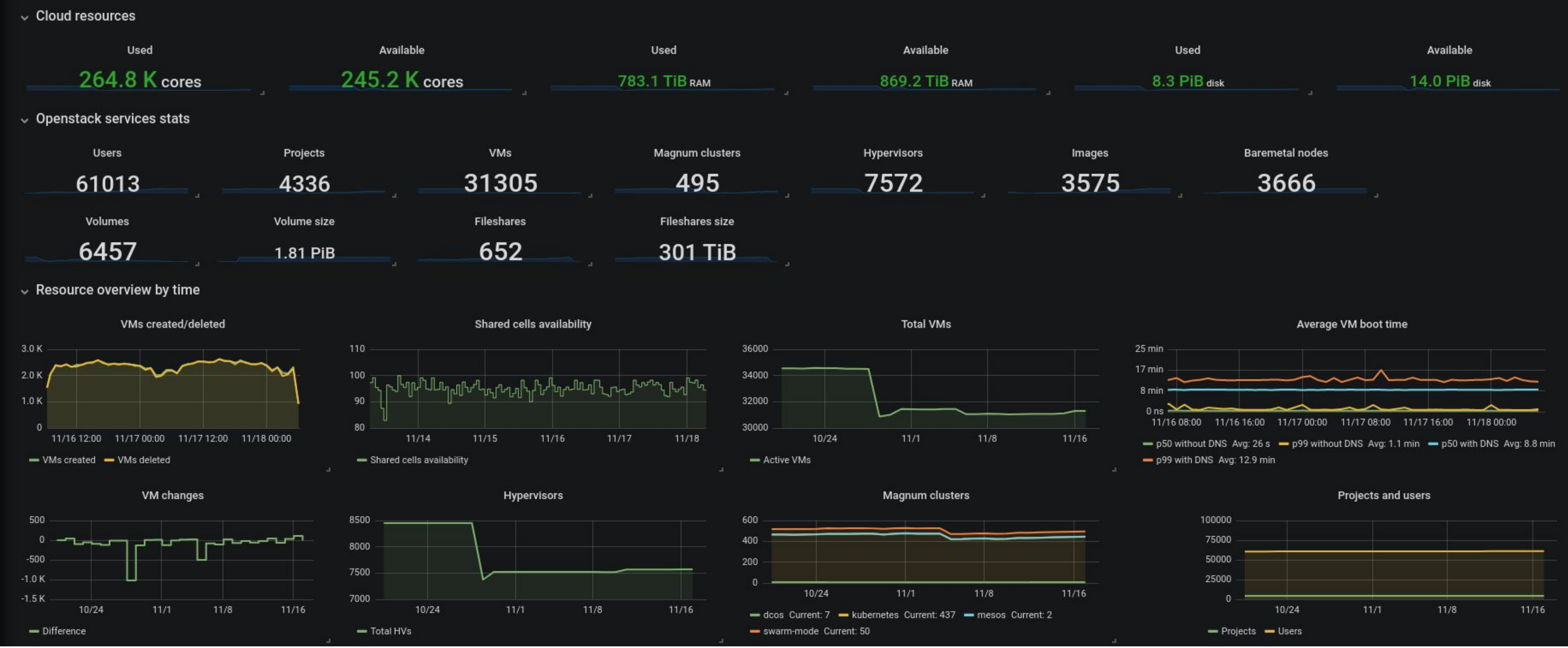
@ahcorporto
ricardo.rocha@cern.ch






CMS
CERN
LHC
[Large Hadron Collider]
ALICE
LHCb
ATLAS


14T900



⌄ Cloud resources

Used	Available	Used	Available	Used	Available
264.8 K cores	245.2 K cores	783.1 TiB RAM	869.2 TiB RAM	8.3 PiB disk	14.0 PiB disk

⌄ Openstack services stats

Users	Projects	VMs	Magnum clusters	Hypervisors	Images	Baremetal nodes
61013	4336	31305	495	7572	3575	3666

Volumes	Volume size	Fileshares	Fileshares size
6457	1.81 PiB	652	301 TiB

⌄ Resource overview by time

VMs created/deleted

🟩 VMs created 🟨 VMs deleted

Shared cells availability

🟩 Shared cells availability

Total VMs

🟩 Active VMs

Average VM boot time

🟩 p50 without DNS Avg: 26 s 🟨 p99 without DNS Avg: 1.1 min 🟦 p50 with DNS Avg: 8.8 min
🟧 p99 with DNS Avg: 12.9 min

VM changes

🟩 Difference

Hypervisors

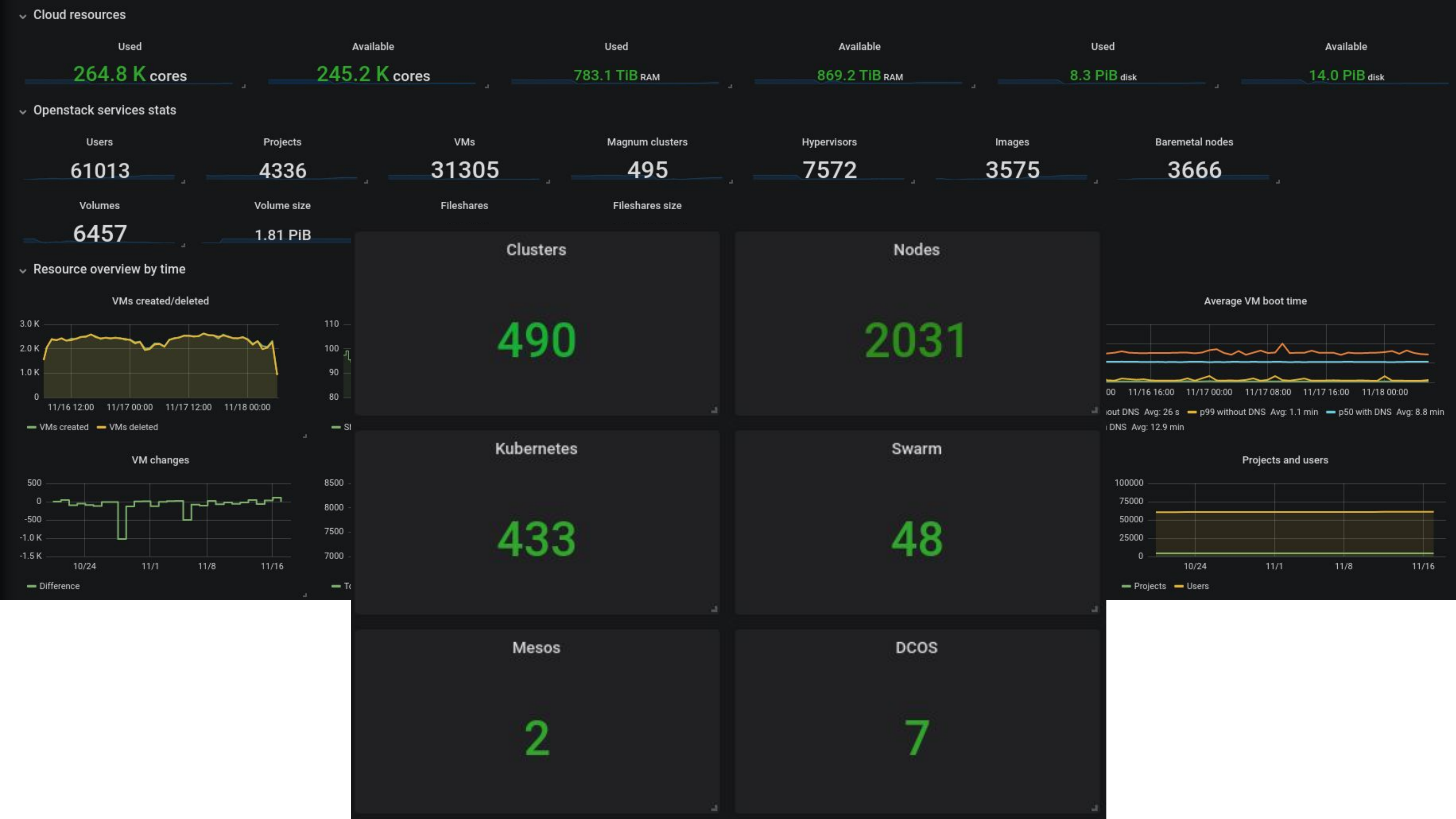
🟩 Total HVs

Magnum clusters

🟩 dcos Current: 7 🟨 kubernetes Current: 437 🟦 mesos Current: 2
🟧 swarm-mode Current: 50

Projects and users

🟩 Projects 🟨 Users



Cloud resources

Used	Available	Used	Available	Used	Available
264.8 K cores	245.2 K cores	783.1 TiB RAM	869.2 TiB RAM	8.3 PiB disk	14.0 PiB disk

Openstack services stats

Users	Projects	VMs	Magnum clusters	Hypervisors	Images	Baremetal nodes
61013	4336	31305	495	7572	3575	3666

Volumes	Volume size	Fileshares	Fileshares size
6457	1.81 PiB		

Resource overview by time

Clusters	Nodes
490	2031

Kubernetes	Swarm
433	48

Mesos	DCOS
2	7

Computing at CERN

Increased numbers, increased automation

1970s

2007

Computing at CERN

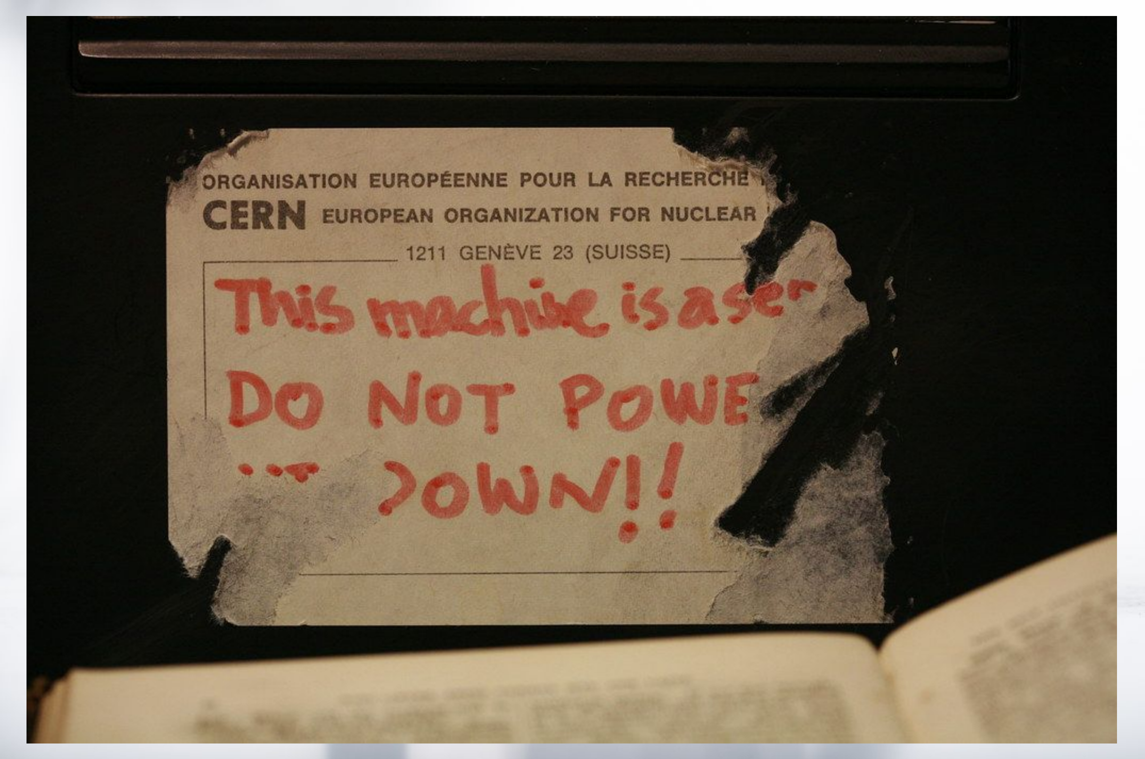
Increased numbers, increased automation

Computing at CERN

Increased numbers, increased automation

Computing at CERN

Increased numbers, increased automation



ORGANISATION EUROPÉENNE POUR LA RECHERCHE

CERN EUROPEAN ORGANIZATION FOR NUCLEAR

1211 GENÈVE 23 (SUISSE)

This machine is a ser

DO NOT POWE

DOWN!!

Automation and Efficiency

	Provisioning	Maintenance	Deployment	Update	Utilization
Physical Infrastructure	Days or Weeks	Highly Intrusive	Minutes or Hours	Minutes or Hours	Poor

	Provisioning	Maintenance	Deployment	Update	Utilization
Physical Infrastructure	Days or Weeks	Highly Intrusive	Minutes or Hours	Minutes or Hours	Poor
Cloud API Virtualization	Minutes	Potentially Less Intrusive	Minutes or Hours	Minutes or Hours	Good

	Provisioning	Maintenance	Deployment	Update	Utilization
Physical Infrastructure	Days or Weeks	Highly Intrusive	Minutes or Hours	Minutes or Hours	Poor
Cloud API Virtualization	Minutes	Potentially Less Intrusive	Minutes or Hours	Minutes or Hours	Good
Containers	Seconds	Less Intrusive	Seconds	Seconds	Very Good

Physical to Virtualization and Cloud

“ Where is my machine hosted? “

“ What is the state of the hypervisor? “

“ Could you check for noisy neighbors? “

But similar automation tools, ssh, systemd, syslog, etc

And then to containers ...

“ How do i retrieve my application’s logs? And how to log rotate? “

“ How do i access the node running container X ? “

“ How do i install package X on the nodes? “

“ Seems like one of the cluster node’s filesystem went read-only... “

“ Docker, Kubernetes, Ingress … now Helm … this is a lot of new stuff! “

Significant change in mindset and a steeper learning curve

Container Use Cases

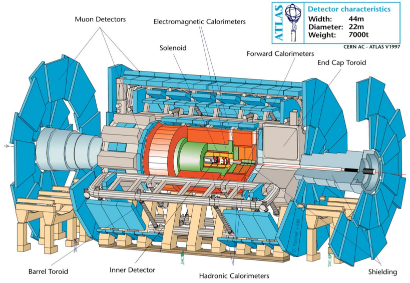
Experiment Trigger farms

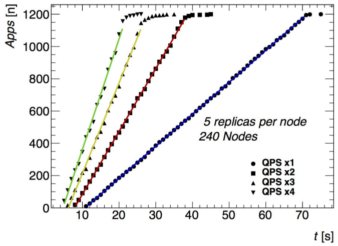
Spark as a Service, on demand Spark clusters on Kubernetes

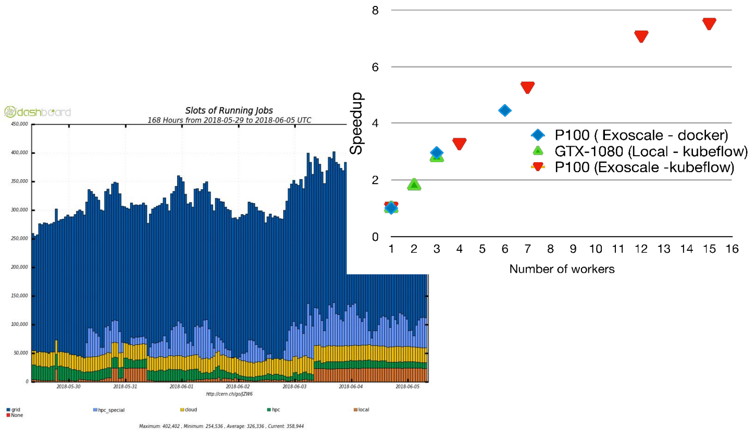
KubeFlow and distributed ML training

Batch on Kubernetes, Native and HTCondor

WebLogic and other internal services







Making it easier...

Container Trainings, Workshops, Office Hours

One thing is similar … what is now called GitOps

- We’ve used git for years to store and manage configuration
- Maybe that can help onboarding more service managers

Puppet to Helm

- Manifests vs Golang, YAML config for both
- Much faster turn-around

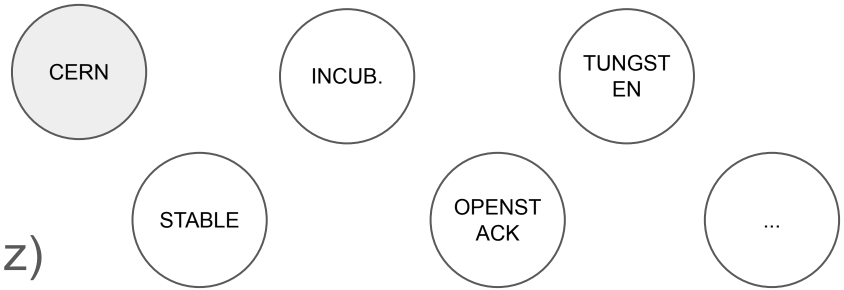
HELM

Charts Repository

Initially package charts stored in plain S3

Moved to chartmuseum to have a management API, with S3 as backend

Mirrored and home grown chart repositories



CERN

INCUB.

TUNGST
EN

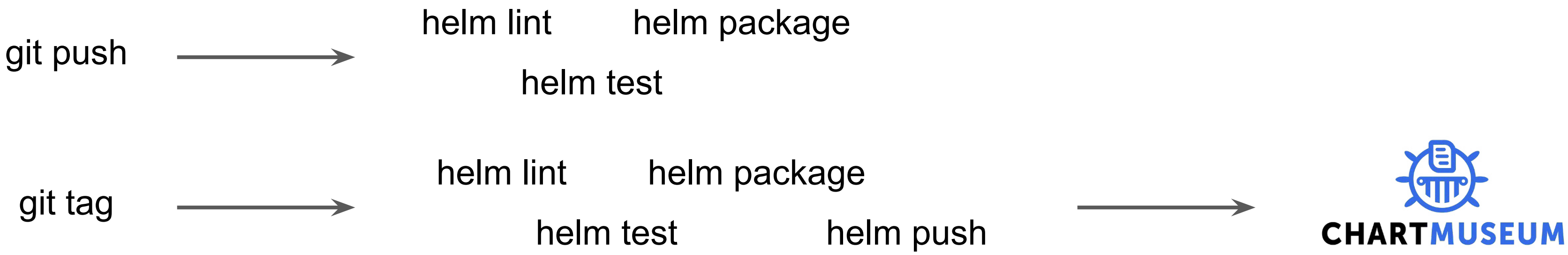
STABLE

OPENST
ACK

...

All triggered by GitLab CI

Versions include commit hash (x.y.z-cern-x.y.z)



git push → helm lint helm package helm test

git tag → helm lint helm package helm test helm push → CHARTMUSEUM

Umbrella Charts

Meta charts wrapping the different charts required per application

Units of deployment with all dependencies and any additional manifests

Stored separately as they manage cluster state (permissions and visibility)

First go relied on branches for environments and a custom structure

```
$ ls
Chart.yaml requirements.yaml secrets.yaml templates/ values.yaml
```

```
$ cat requirements.yaml
dependencies:
  - name: binderhub
    version: 0.2.0-575fb2a
    repository: https://charts.cern.ch/jupyterhub
```

```
$ ls templates
ds-gpu.yaml psp.yaml
```

Managing Secrets

Option 1: Building on Kubernetes Secrets or similar CRDs

- No easy or obvious way to plug external secrets
- Bitnami SealedSecrets: works well, but hard with existing charts
- Vault an option to fully delegate secret management

Option 2: Take (part of) the helm values as secret data, not the resources

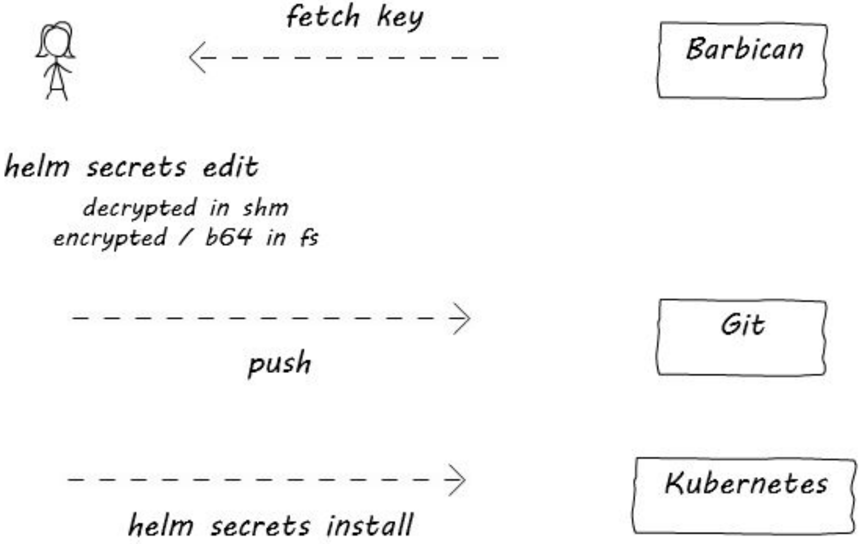
- Versioning of secrets along the rest of the configuration
- Futuresimple helm-secrets (existing plugin) with sops

A Barbican Secret Plugin for Helm

Similar interface to futuresimple helm-secrets

Builds on existing identity scheme to access and manage encryption keys

```
$ helm --name <release> secrets
     view secrets.yaml
     edit secrets.yaml
     install stable/nginx --values secrets.yaml
     upgrade stable/nginx --values secrets.yaml
     lint --values secrets.yaml
```



Similar wrapper for kubectl

https://github.com/cernops/helm-barbican

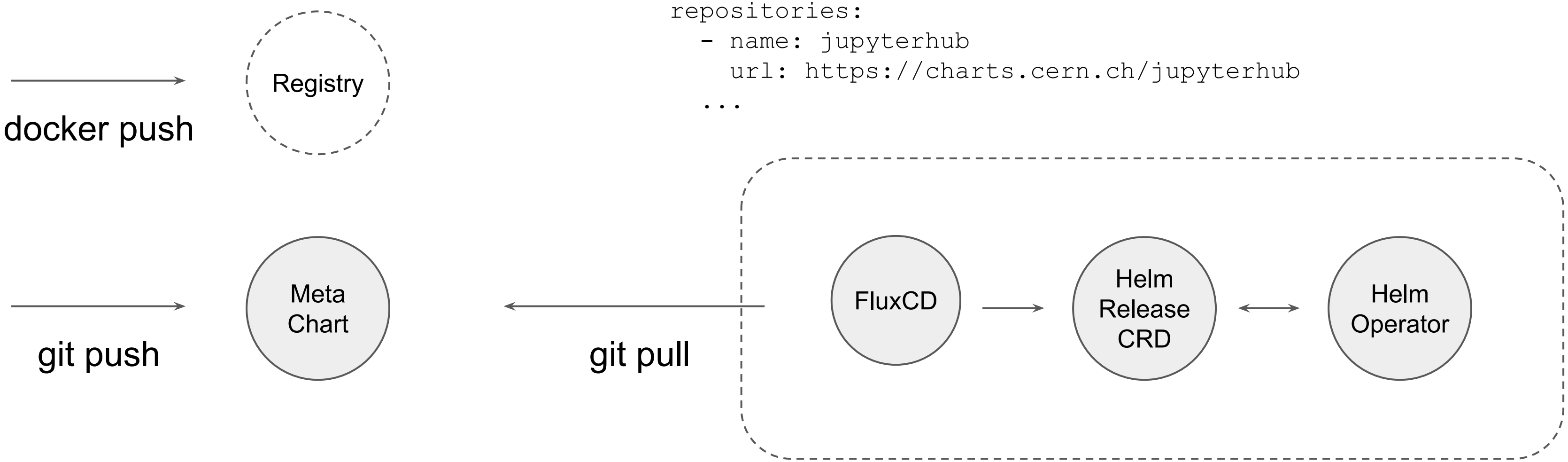
Flux and GitOps

Our end goal from the start

Relying on chart updates only

```
$ helm install fluxcd/flux \
      --namespace flux --name flux --values flux-values.yaml
      --set git.pollInterval=1m
      --set git.url=https://gitlab.cern.ch/.../hub

$ cat flux-values.yaml
rbac:
  create: true
helmOperator:
  create: true
  chartsSyncInterval: 5m
  configureRepositories:
    enable: true
    repositories:
      - name: jupyterhub
        url: https://charts.cern.ch/jupyterhub
      ...
```



Flux and GitOps

What’s in a Helm Release?

```
|-- charts
    |-- hub
        Chart.yaml requirements.yaml values.yaml
        |-- templates
            custom-manifest.yaml
|-- namespaces
    prod.yaml stg.yaml
|-- releases
    |-- prod
        hub.yaml
    |-- stg
        hub.yaml
|-- secrets
    |-- prod
        secrets.yaml
    |-- stg
        secrets.yaml
```

```
apiVersion: flux.weave.works/v1beta1
kind: HelmRelease
metadata:
  name: hub
  namespace: prod
spec:
  releaseName: hub
  chart:
    git: https://gitlab.cern.ch/.../hub.git
    path: charts/hub
    ref: master
  valuesFrom:
  - secretKeyRef:
      name: hub-secrets
      key: values.yaml
  values:
    binderhub:
      ...
```

This is how we plug our encrypted values data

Use Case: JupyterHub + BinderHub

Demo time

Ongoing: GitOps for Cluster Lifecycle

Currently validating this solution to centrally manage upgrades

Reduce the scope of the cluster orchestration tool to base components

Let a single Flux HelmRelease manage all add-ons (staging, prod)

```
dependencies:
  - name: eosxd
    version: 0.3.1-cern-0.1.0-7+ba5e81
    repository: http://charts.cern.ch/cern
  - name: fluentd
    version: 2.2.1-cern-0.1.0-3+1c551a1
    repository: http://charts.cern.ch/stable
  - name: prometheus
    version: 9.3.1-cern-0.1.0-3+1c551a1
    repository: http://charts.cern.ch/stable
  - name: traefik
    version: 1.79.0-cern-0.1.0-3+1c551a1
    repository: http://charts.cern.ch/stable
  ...
```

Conclusion & Next Steps

Helm and (Argo) Flux give us a familiar toolset for containerized applications

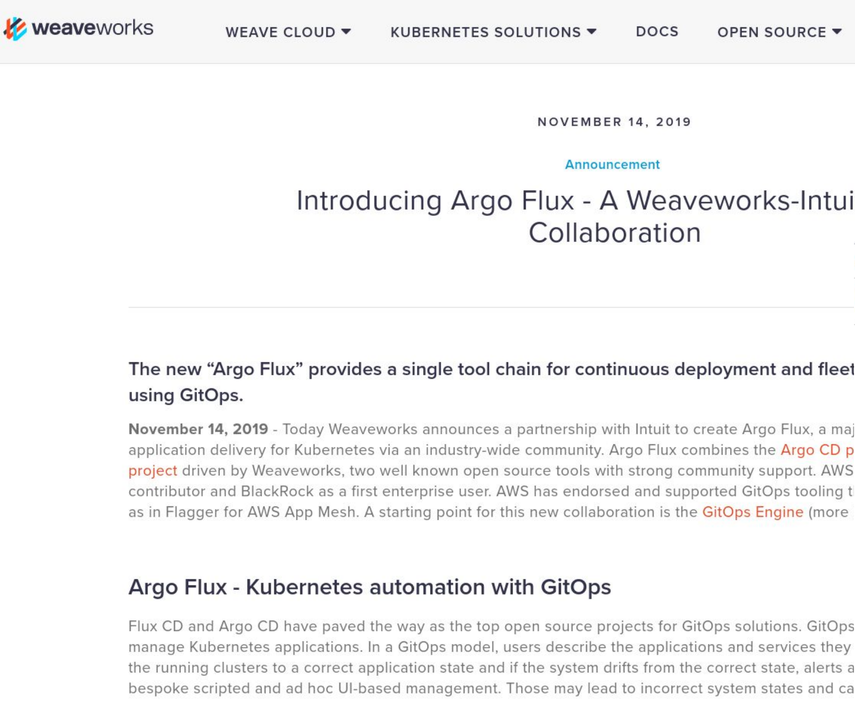
Git as the source of truth

Helm v3 and goodbye Tiller

Helm Hub, Signed Helm Charts

(re) Consider automation of charts and container image updates

Cattle clusters, Blue / Green, Canary with Service Mesh



NOVEMBER 14, 2019

Announcement

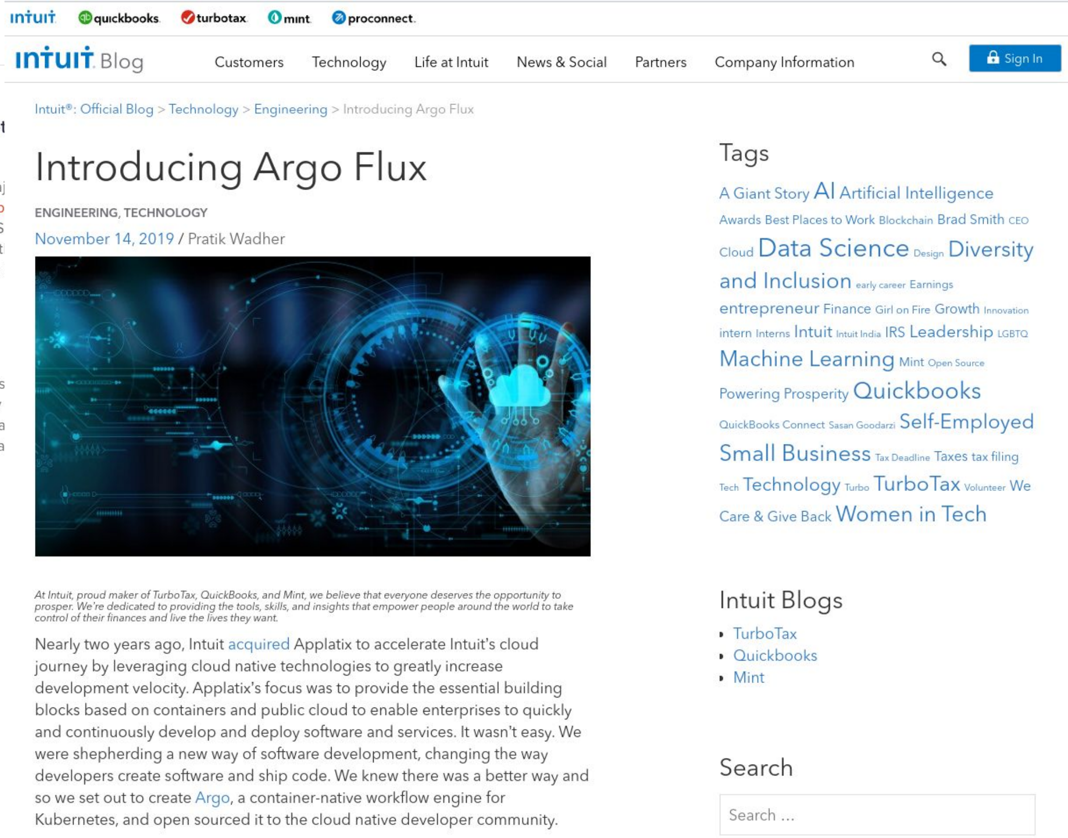
Introducing Argo Flux - A Weaveworks-Intuit-AWS Collaboration

The new "Argo Flux" provides a single tool chain for continuous deployment and fleet using GitOps.

November 14, 2019 - Today Weaveworks announces a partnership with Intuit to create Argo Flux, a maj application delivery for Kubernetes via an industry-wide community. Argo Flux combines the Argo CD p project driven by Weaveworks, two well known open source tools with strong community support. AWS contributor and BlackRock as a first enterprise user. AWS has endorsed and supported GitOps tooling t as in Flagger for AWS App Mesh. A starting point for this new collaboration is the GitOps Engine (more

Argo Flux - Kubernetes automation with GitOps

Flux CD and Argo CD have paved the way as the top open source projects for GitOps solutions. GitOps manage Kubernetes applications. In a GitOps model, users describe the applications and services they the running clusters to a correct application state and if the system drifts from the correct state, alerts a bespoke scripted and ad hoc UI-based management. Those may lead to incorrect system states and ca



Intuit®: Official Blog > Technology > Engineering > Introducing Argo Flux

Introducing Argo Flux

ENGINEERING, TECHNOLOGY

November 14, 2019 / Pratik Wadher

At Intuit, proud maker of TurboTax, QuickBooks, and Mint, we believe that everyone deserves the opportunity to prosper. We're dedicated to providing the tools, skills, and insights that empower people around the world to take control of their finances and live the lives they want.

Nearly two years ago, Intuit acquired Applatix to accelerate Intuit's cloud journey by leveraging cloud native technologies to greatly increase development velocity. Applatix's focus was to provide the essential building blocks based on containers and public cloud to enable enterprises to quickly and continuously develop and deploy software and services. It wasn't easy. We were shepherding a new way of software development, changing the way developers create software and ship code. We knew there was a better way and so we set out to create Argo, a container-native workflow engine for Kubernetes, and open sourced it to the cloud native developer community.

Tags
A Giant Story AI Artificial Intelligence Awards Best Places to Work Blockchain Brad Smith CEO Cloud Data Science Design Diversity and Inclusion early career Earnings entrepreneur Finance Girl on Fire Growth Innovation intern Interns Intuit Intuit India IRS Leadership LGBTQ Machine Learning Mint Open Source Powering Prosperity Quickbooks QuickBooks Connect Sasan Goodarzi Self-Employed Small Business Tax Deadline Taxes tax filing Tech Technology Turbo TurboTax Volunteer We Care & Give Back Women in Tech
Intuit Blogs
TurboTax
Quickbooks
Mint
Search
Search ...

Questions?

LHC is in a long shutdown for the next year, underground visits possible

https://visit.cern

Follow our tech blog https://techblog.web.cern.ch

@ahcorporto , ricardo.rocha@cern.ch