# Helm and GitOps at CERN

Ricardo Rocha

## 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 |

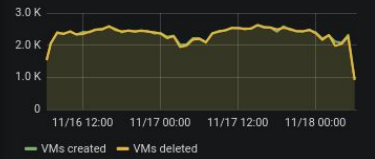
## Openstack services stats

| 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

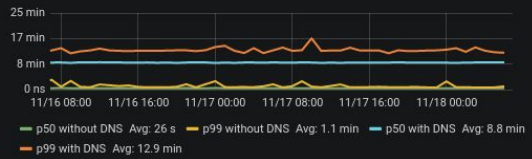
### VMs created/deleted




### Shared cells availability


### Total VMs

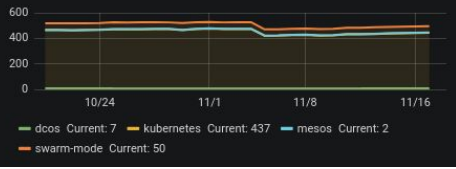

### Average VM boot time



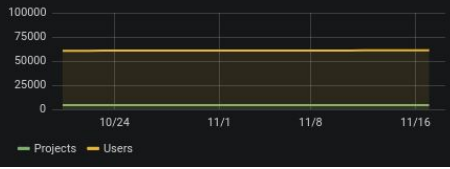

### VM changes


### Hypervisors


### Magnum clusters




### Projects and 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

VMs created/deleted


VM changes


### Clusters

490

### Nodes

2031

### Kubernetes

433

### Swarm

48

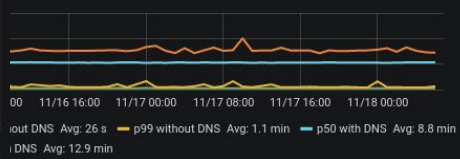
### Mesos

2

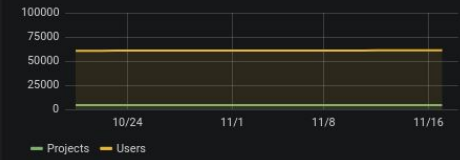
### DCOS

7

Average VM boot time




Projects and users



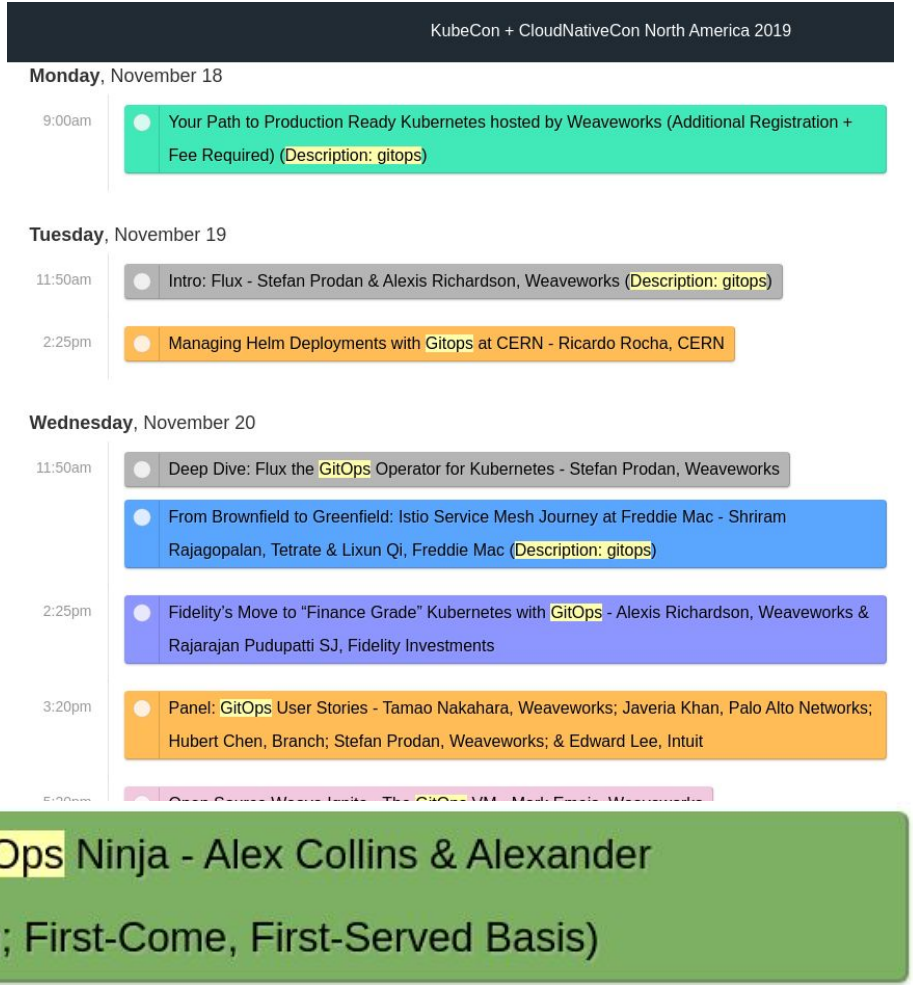
# Why

Single source of truth

Reusability

Automation of deployment and upgrades

Multi cluster configuration



And also ...

# Transition to kubernetes is not trivial ...

*“ 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 steep learning curve**

# What is Helm

The Kubernetes package manager

A **Chart** manages the deployment and configuration of an application

- Reusable, shareable units
- Includes all required manifests, plus any required libraries for lifecycle

Separate **values** definitions for instance configuration

## Template

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: eosxd-config
  labels:
    app: {{ template "eosxd.name" . }}
    chart: {{ template "eosxd.chart" . }}
    release: {{ .Release.Name }}
    heritage: {{ .Release.Service }}
...
  {{- range $area, $mountpoints := .Values.mounts }}
    {{- range $mountpoint, $letters := $mountpoints }}

  fuse.{{ $mountpoint }}.conf: |+
    {"name":"{{ $mountpoint }}","hostport":"eos{{
$mountpoint }}.cern.ch","localmountdir":"/eos/{{
$mountpoint }}/", "remotemountdir":"/eos/{{ $area }}/",
"bind":"{{ $letters }}"}

    {{- end }}
  {{- end }}
```

## Template

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: eosxd-config
  labels:
    app: {{ template "eosxd.name" . }}
    chart: {{ template "eosxd.chart" . }}
    release: {{ .Release.Name }}
    heritage: {{ .Release.Service }}
...
  {{- range $area, $mountpoints := .Values.mounts }}
    {{- range $mountpoint, $letters := $mountpoints }}

  fuse.{{ $mountpoint }}.conf: |+
    {"name":"{{ $mountpoint }}","hostport":"eos{{
$mountpoint }}.cern.ch","localmountdir":"/eos/{{
$mountpoint }}/", "remotemountdir":"/eos/{{ $area }}/",
"bind":"{{ $letters }}"}

    {{- end }}
  {{- end }}
```

## Values

```
image:
  repository: gitlab-registry.../eosd
  tag: 0.4.0
  pullPolicy: IfNotPresent
mounts:
  ams:
  atlas:
  cms:
  experiment:
  lhcb:
  project:
    project-i00: "a e j g v k q y"
    project-i01: "l h b p s f w n o"
    project-i02: "d c i r m t u x z"
  theory:
  user:
    home-i00: "l n t z"
    home-i01: "a g j k w"
    home-i02: "h o r s y"
    home-i03: "b e m v x"
    home-i04: "c f i p q"
  workspace:
```

# Charts Repository

CERN instance: https://charts.cern.ch


A central catalog: the Helm Hub

Quite recent but already points to most popular charts

Does not host the charts, acts like a catalog

https://hub.helm.sh

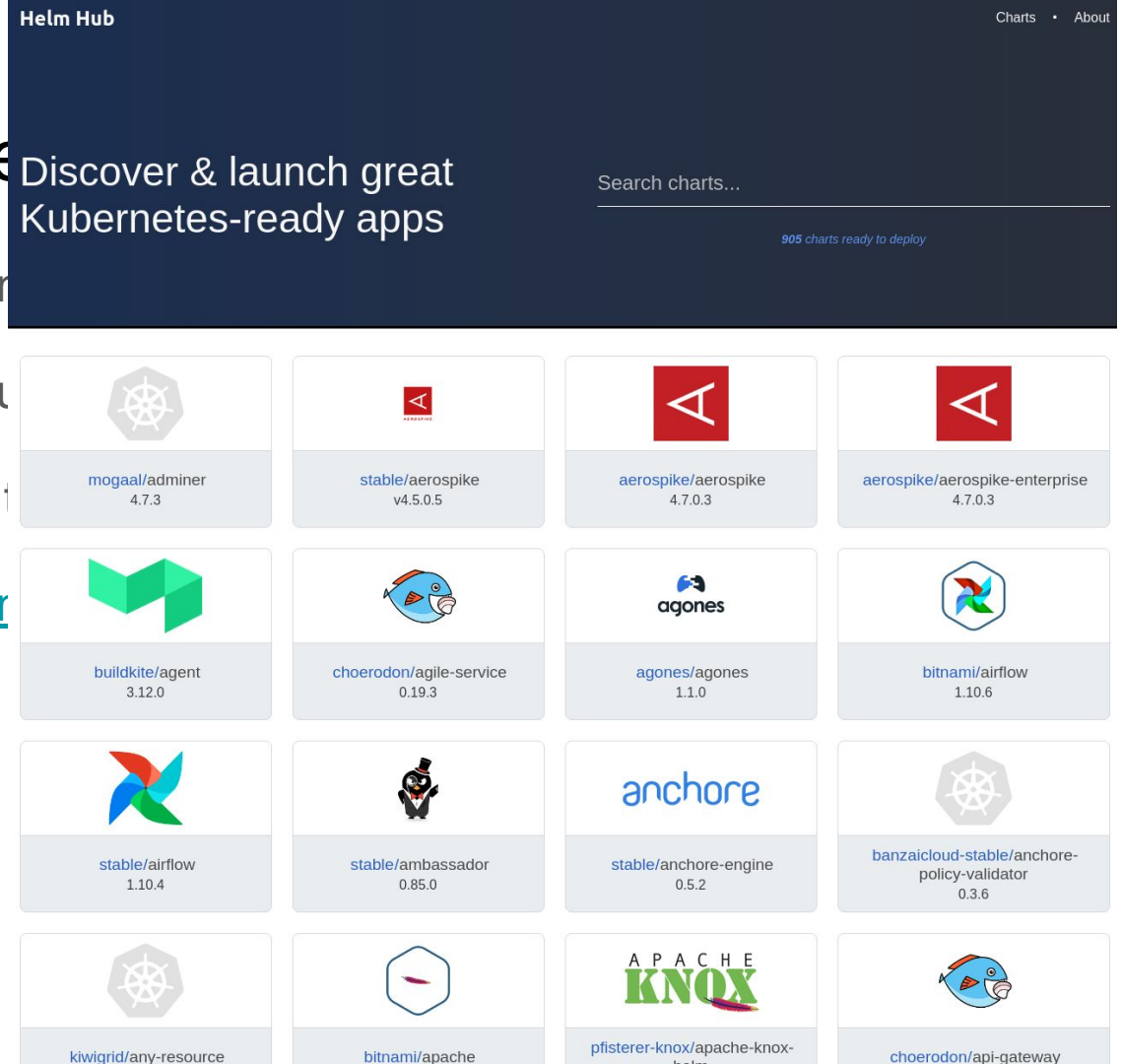
# Charts Re

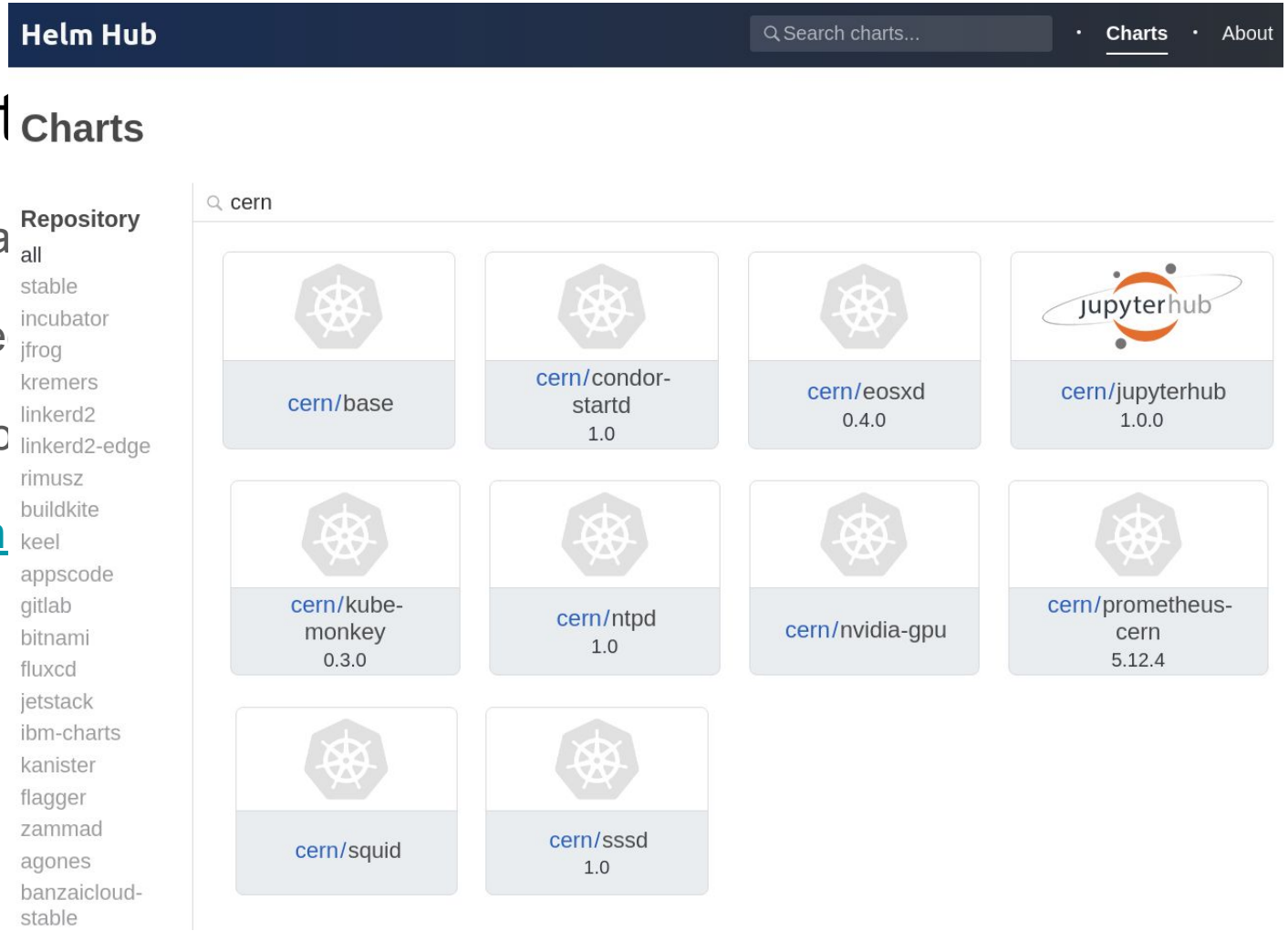
A central instar

Quite recent bu

Does not host

https://hub.helr






Chart

A centra

Quite re

Does nc

https://h

# Umbrella Charts

Charts are reusable deployments units

Most applications have multiple dependencies

Umbrella charts wrap all the required charts into a single deployment unit

With any additional manifests required

```
$ dependencies:
  - name: mysql
    version: 5.3
    repository: https://kubernetes-charts.storage.googleapis.com/
  - name: nginx
    version: 1.16.1
    repository: https://kubernetes-charts.storage.googleapis.com/
```

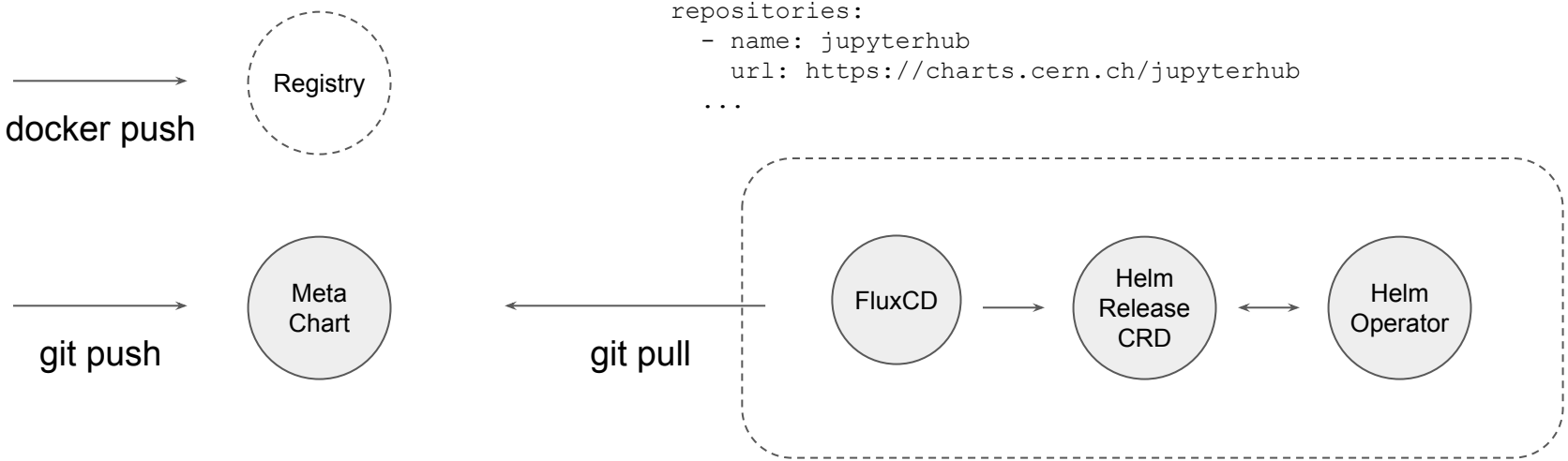
# Flux and GitOps

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?

```
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:
      ...
```



```
|-- 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
```



*This is how we plug our encrypted values data*

# A Barbican Secret Plugin for Helm

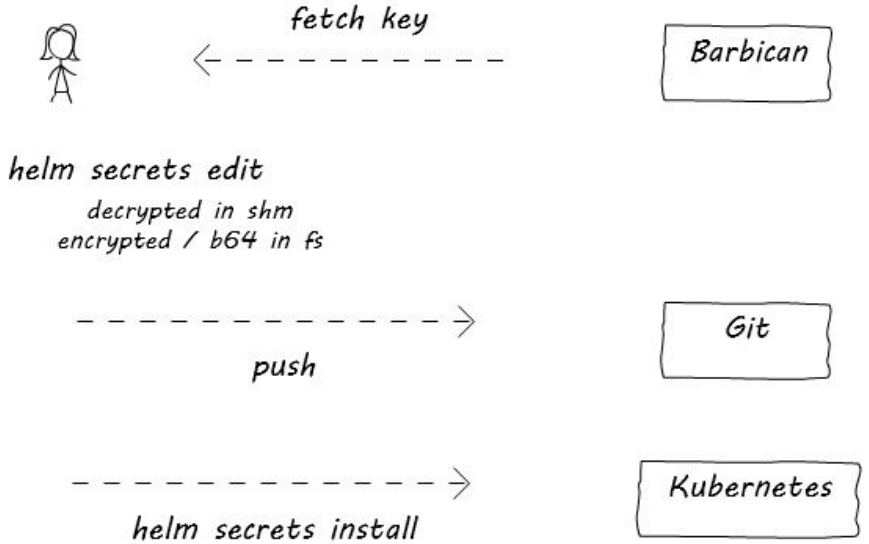
Similar interface to futuresimple helm-secrets

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



# Deployment Model

**1** ⇢ **1**: This is currently our most common model

Kubernetes clusters live in the end user's project

Any service aggregation and consolidation is done at that level

**1** ⇢ ***: Replicate the same application in multiple clusters

HA, Blast Radius, Blue / Green style

* ⇢ *: Workloads also share the underlying resources

Nice separation between service managers and infrastructure

Stronger requirements on multi-tenancy, quotas at cluster level

# Conclusion & Next Steps

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

Git as the source of truth

After our own tools, working on dissemination (see Rucio's talk)

Experimenting with the best model to distribute workloads

- Likely a mix in the end

Cattle clusters, Blue / Green, Canary with Service Mesh

? Should we keep a WLCG repository for common reusable charts