The background of the slide features a dramatic photograph of two climbers on a snow-covered mountain ridge. One climber is in the foreground, seen from behind, wearing a blue jacket and a red helmet. The other climber is partially visible behind them. They are looking out over a vast, layered mountain range under a clear blue sky.

RUD: Rapid Unscheduled Deletion of Critical Infrastructure Workloads

Kubernetes Community Days Zürich - 13th of June 2024

Clément Nussbaumer



RUD - Rapid Unscheduled ...

Disassembly 🚀 💥



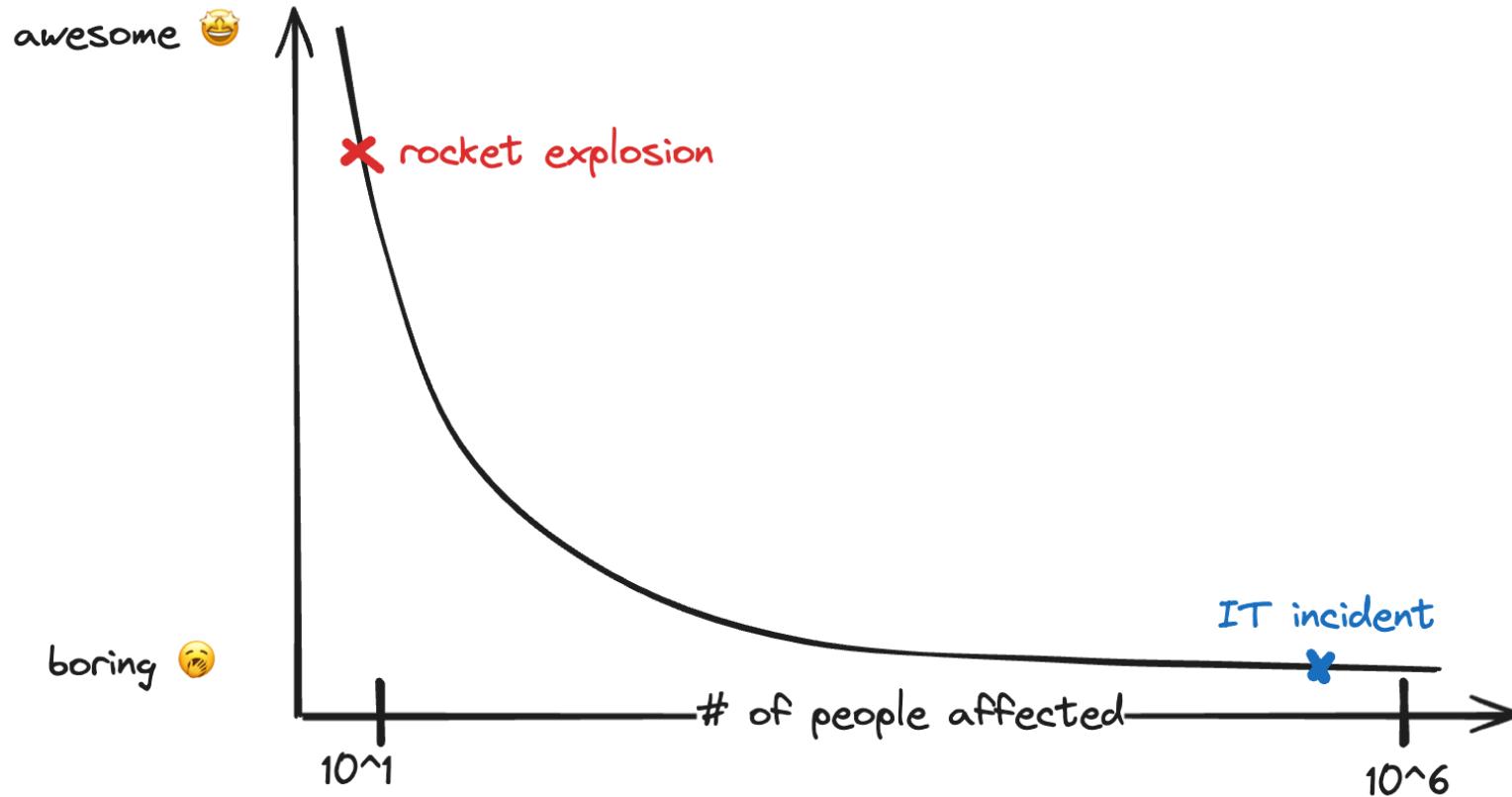
SPACE.com Footage courtesy: SPadre.com

Deletion of Critical Infrastructure Workload 😭 ⏳

The screenshot shows a web-based interface for managing Kubernetes applications. On the left, a sidebar displays icons for Applications, Services, Deployments, and Events. The main area shows a card for an application named 'kube-metrics-ns' in the 'kube-argocd-master' namespace. The card indicates the application is pending deletion. Below the card, a 'LIVE MANIFEST' section shows the YAML configuration for the application:

```
1 apiVersion: argoproj.io/v1alpha1
2 kind: Application
3 metadata:
4   creationTimestamp: '2024-06-10T02:21:42Z'
5   deletionGracePeriodSeconds: 0
6   deletionTimestamp: '2024-06-10T17:44:56Z'
```

Awesomeness scale



Agenda:

1. Kubernetes @ PostFinance
2. Infrastructure Workloads
3. ArgoCD ApplicationsSet
4. The RUD Incident

Kubernetes @ PostFinance

Some figures:

- **1406 namespaces** across **30** Kubernetes clusters
- **18'000 pods, 4200** ingresses
- **450 nodes** (24 CPU / 64 GiB RAM)
 - 14 GPU nodes for machine learning workloads
- **4000 RPS** peak load across all ingresses
- **400** card transaction per second

Some facts:

- **Vanilla v1.28 Kubernetes**
 - Oldest cluster will soon turn **5 years old**
- **Self-service repository + controller** to manage all namespaces
- **ArgoCD** and GitOps for almost all deployments
- **Chaos Monkey** on all clusters 

Most critical workload: **card payment services**  

Infrastructure Workloads

A little overview

```
> kubectl get namespace kube-metrics
```

NAME	STATUS	AGE
kube-metrics	Active	215d

```
> kubectl --namespace=kube-metrics get deployments.apps
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
kube-state-metrics	1/1	1	1	215d
metrics-server	1/1	1	1	215d

```
> kubectl --namespace=kube-metrics get daemonsets.apps
```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
node-exporter	103	103	103	103	103	kubernetes.io/os=linux	215d

```
> kubectl get apiservice v1beta1.metrics.k8s.io
```

NAME	SERVICE	AVAILABLE	AGE
v1beta1.metrics.k8s.io	kube-metrics/metrics-server	True	216d

Infrastructure Workloads

How to deploy them on all clusters ?

Up until September 2022: combination of `ansible`, `helm`, and GitLab pipelines → 🐢

After September 2022: migration to ArgoCD `ApplicationSets` → 🐰

The screenshot shows the ArgoCD application details interface for the application `kube-argocd-master`. The sync status is **Synced** to main (09ff55f). The application set `kube-log` contains several applications, each with a sync status of **Synced** and a timestamp indicating the last sync. The applications are:

- `e1-k8s-pfnet-a-kube-log` (2 months ago)
- `e1-k8s-pfnet-lab-a-kube-log` (a day ago)
- `e1-k8s-pfnet-lab-b-kube-log` (2 months ago)
- `e1-k8s-pfnet-lab-e-kube-log` (10 hours ago)
- `e1-k8s-pfnet-lab-f-kube-log` (a day ago)
- `e1-k8s-pfnet-lab-g-kube-log` (an hour ago)
- `e1-k8s-pfnet-lab-h-kube-log` (4 days ago)
- `e1-k8s-pfnet-lab-v-kube-log` (a day ago)



ArgoCD Application

What is it ?

```
> kubectl explain application.spec
GROUP:      argoproj.io
KIND:       Application
VERSION:    v1alpha1

FIELDS:
destination  <Object> -required-
  Destination is a reference to the target Kubernetes server and namespace

source       <Object>
  Source is a reference to the location of the application's manifests or
  chart

syncPolicy   <Object>
  SyncPolicy controls when and how a sync will be performed
```

ArgoCD ApplicationSet

What is it ?

```
> kubectl explain applicationset.spec
GROUP:      argoproj.io
KIND:       ApplicationSet
VERSION:    v1alpha1
```

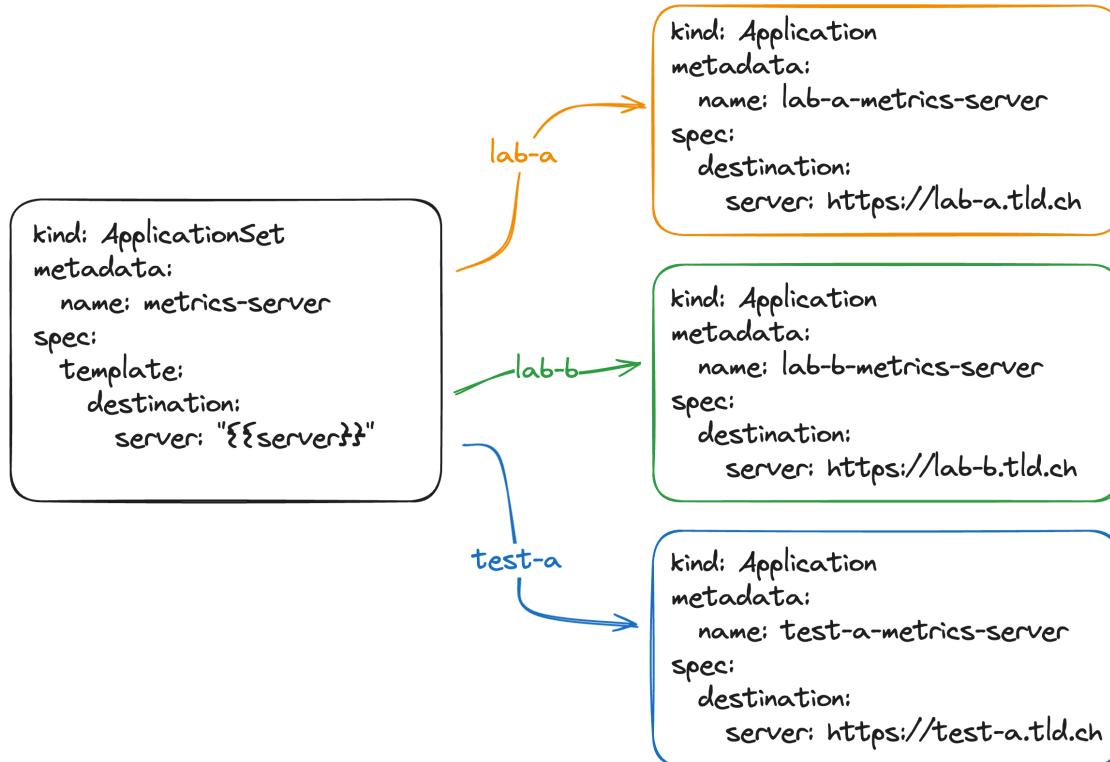
FIELDS:

```
generators    <[]Object> -required-
<no description>
```

```
syncPolicy    <Object>
<no description>
```

```
template      <Object> -required-
<no description>
```

ArgoCD ApplicationSet



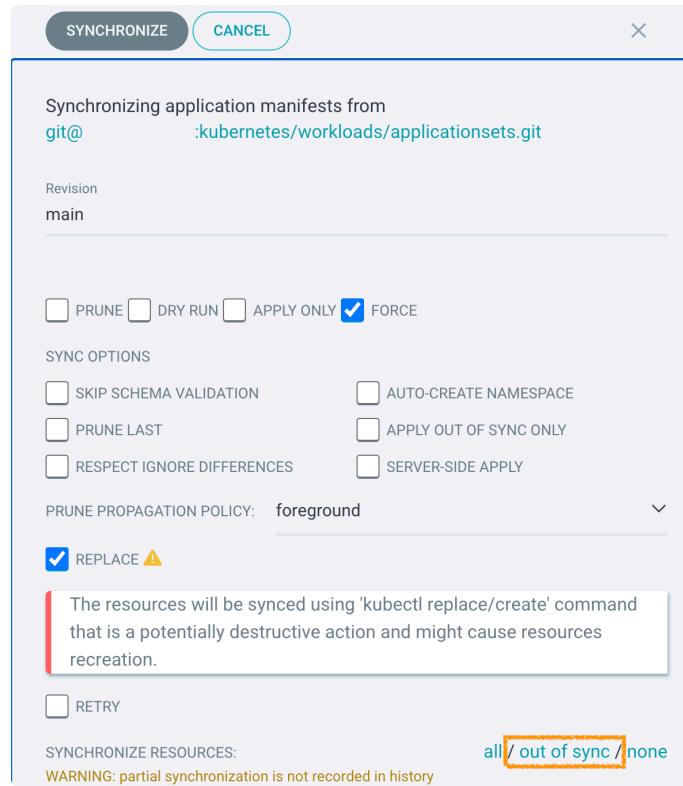
ArgoCD ApplicationSet

metrics-server example

```
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: kube-metrics-metrics-server
spec:
  generators:
  - clusters:
    selector:
      matchLabels:
        stage: lab
    values:
      revision: HEAD
  - clusters:
    selector:
      matchLabels:
        stage: test
    values:
      revision: v0.1.4
```

Rapid Unscheduled Deletion Incident

November 8th 2023 - 16:35



out of sync ApplicationSets:

- kube-contour
- kube-log
- kube-metrics-ns
- kube-metrics-server
- kube-otel

→ all got replaced 💣

Unable to delete namespaces

APP HEALTH <small>?</small>	SYNC STATUS <small>?</small>	LAST SYNC <small>?</small>
 Progressing	 OutOfSync from HEAD (aa5950b) Auto sync is enabled. Author: Comment: chore: update changelog with 0.1.6 release	 Deleting Running a few seconds ago (Wed Jun 12 2024 16:28:10 GMT+0200)

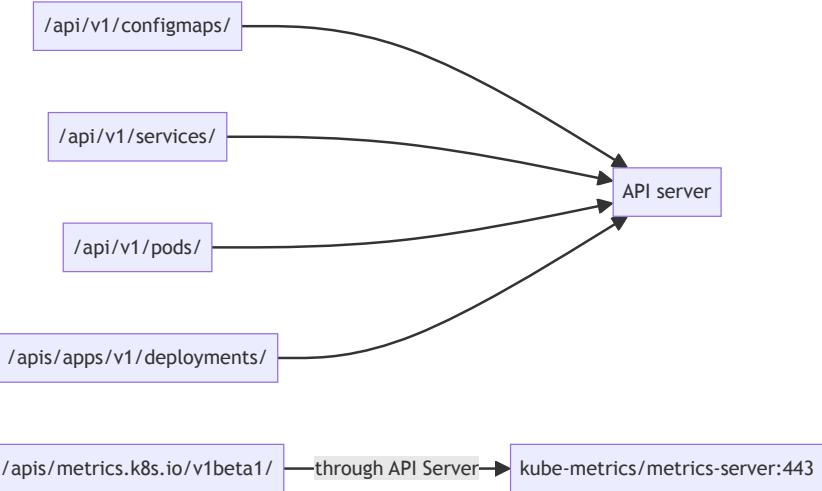
```
> kubectl delete namespace kube-metrics --timeout=10s
namespace "kube-metrics" deleted
error: timed out waiting for the condition on namespaces/kube-metrics

> kubectl get namespace kube-metrics -o=yaml | grep -e "^.status" -A8
status:
  conditions:
  - lastTransitionTime: "2024-06-11T17:20:17Z"
    message: 'Discovery failed for some groups, 1 failing: unable to retrieve the
      complete list of server APIs: metrics.k8s.io/v1beta1: stale GroupVersion discovery:
      metrics.k8s.io/v1beta1'
    reason: DiscoveryFailed
    status: "True"
    type: NamespaceDeletionDiscoveryFailure
```

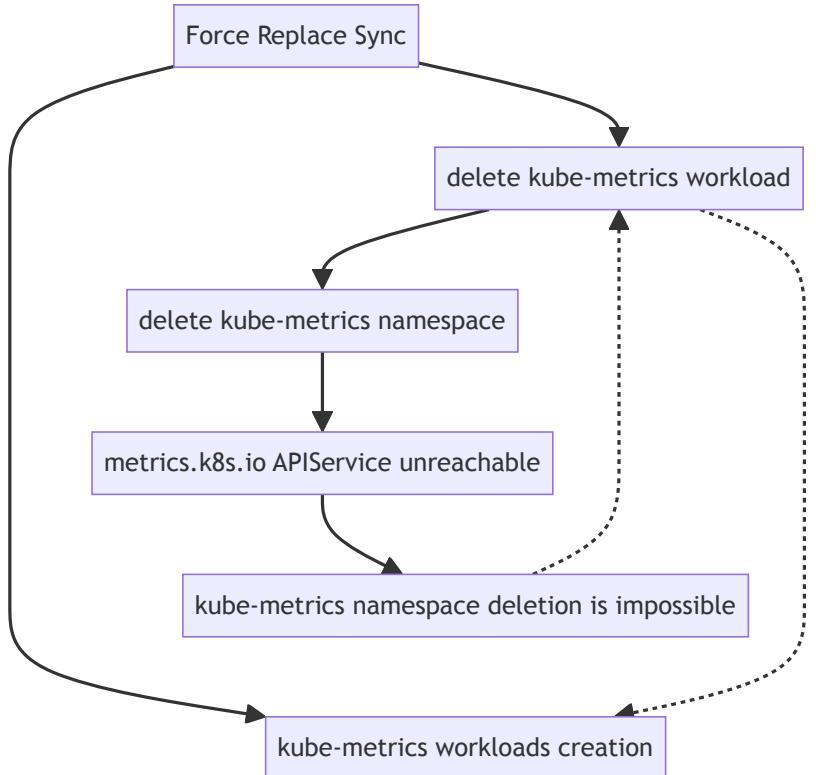
metrics.k8s.io API Service

```
apiVersion: apiregistration.k8s.io/v1
kind: APIService
metadata:
  name: v1beta1.metrics.k8s.io
spec:
  group: metrics.k8s.io
  service:
    name: metrics-server
    namespace: kube-metrics
    port: 443
  version: v1beta1
```

```
> kubectl get --raw /apis/metrics.k8s.io/v1beta1/nodes/
{
  "kind": "NodeMetricsList",
  "apiVersion": "metrics.k8s.io/v1beta1",
  "metadata": {},
  "items": [
    ...
  ]
}
```



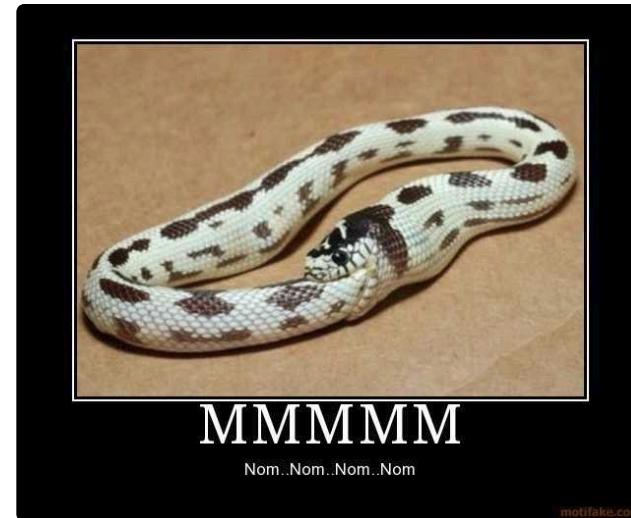
metrics.k8s.io circular dependency



Consequences

all deleted namespaces stay in `Terminating` state indefinitely

`kube-contour` , `kube-log` , etc. are stuck



*snake
biting its
tail*

metrics.k8s.io circular dependency

How to break it ?

```
> kubectl foreach '/^p1-k8s-.*/' -- delete apiservice v1beta1.metrics.k8s.io
Will run command in context(s):
- p1-k8s-cluster-a
- p1-k8s-cluster-b
- p1-k8s-cluster-c
- p1-k8s-cluster-d
- p1-k8s-cluster-e
Continue? [Y/n]: Y
p1-k8s-cluster-a | apiservice.apiregistration.k8s.io "v1beta1.metrics.k8s.io" deleted
p1-k8s-cluster-b | apiservice.apiregistration.k8s.io "v1beta1.metrics.k8s.io" deleted
p1-k8s-cluster-c | apiservice.apiregistration.k8s.io "v1beta1.metrics.k8s.io" deleted
p1-k8s-cluster-d | apiservice.apiregistration.k8s.io "v1beta1.metrics.k8s.io" deleted
p1-k8s-cluster-e | apiservice.apiregistration.k8s.io "v1beta1.metrics.k8s.io" deleted
```

Safeguards

How do we prevent this from happening again ?

1. Split ArgoCD instances (nonprod / prod)
2. `syncPolicy` change on critical `ApplicationSet`

```
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: kube-metrics-metrics-server
spec:
  syncPolicy:
    preserveResourcesOnDeletion: true
  template:
    metadata:
      name: "{{server}}-metrics-server"
      namespace: kube-metrics
  spec:
    ...

```

Considerations ⚠

Which other circular dependencies ?

- Any other API Service extension
- Kubernetes resources with a finalizer
 - PersistentVolumes
 - CustomResourceDefinition
 - ...
- ValidatingWebhook
- MutatingWebhook

GitOps saved the day

Once the circular dependencies were broken

all workloads were back in one minute

Demo

A wide-angle photograph of a mountain range during sunset. The sky is filled with large, billowing clouds that are illuminated from below by the setting sun, giving them a bright orange and yellow glow. In the foreground, dark, rugged mountain peaks rise, some partially covered in snow. A prominent glacier is visible on the left side of the frame, its white surface contrasting with the dark rock. The overall atmosphere is serene and majestic.

Questions ?