CERN

Improving Resource availability in CERN Cloud

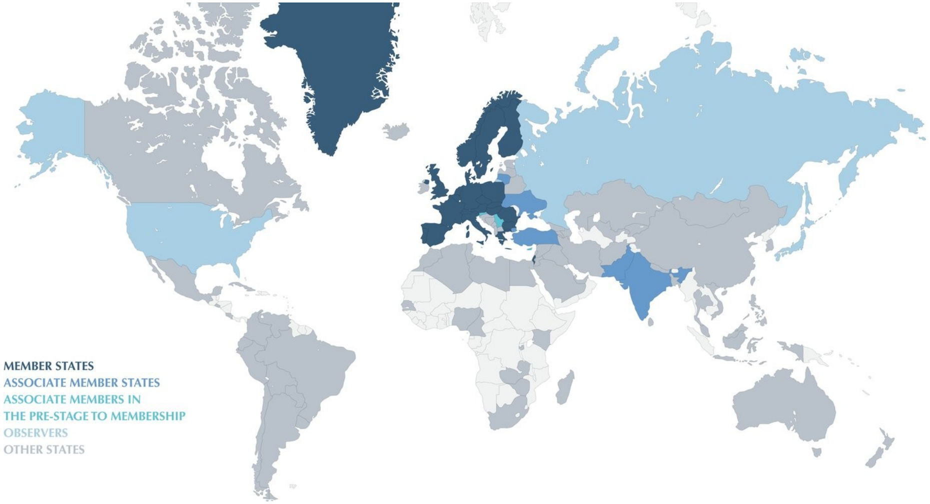
José Castro León & Spyros Trigazis
CERN Cloud Infrastructure

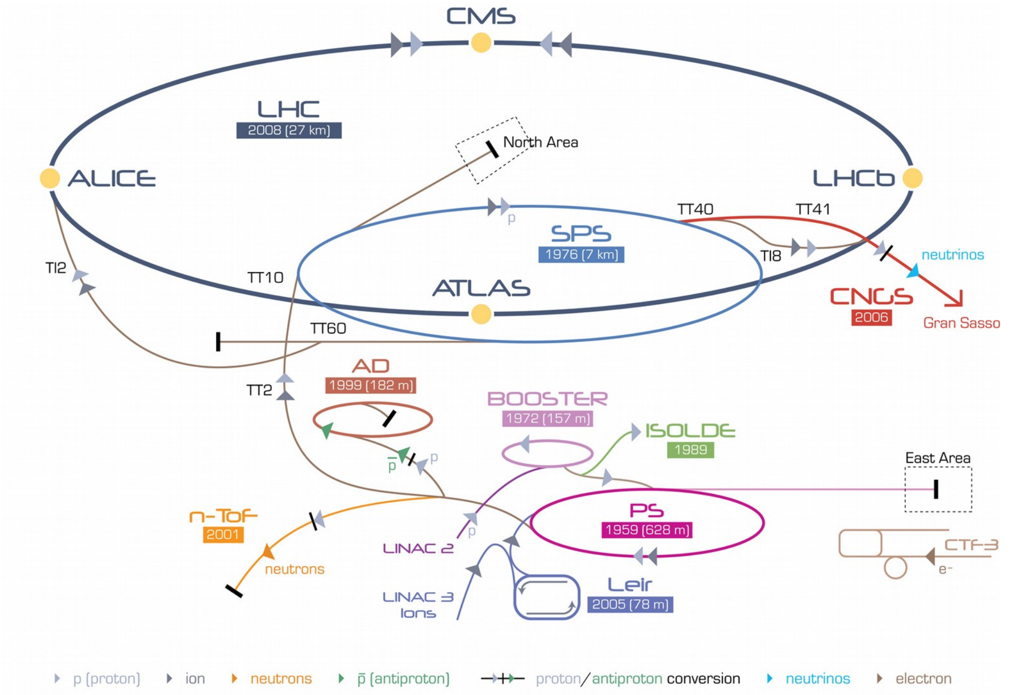
Outlines

- Introduction
- CERN Cloud service
- Get the most of cloud resources
  - Automation
  - Optimization
  - Preemptibles
  - Containers on Baremetal

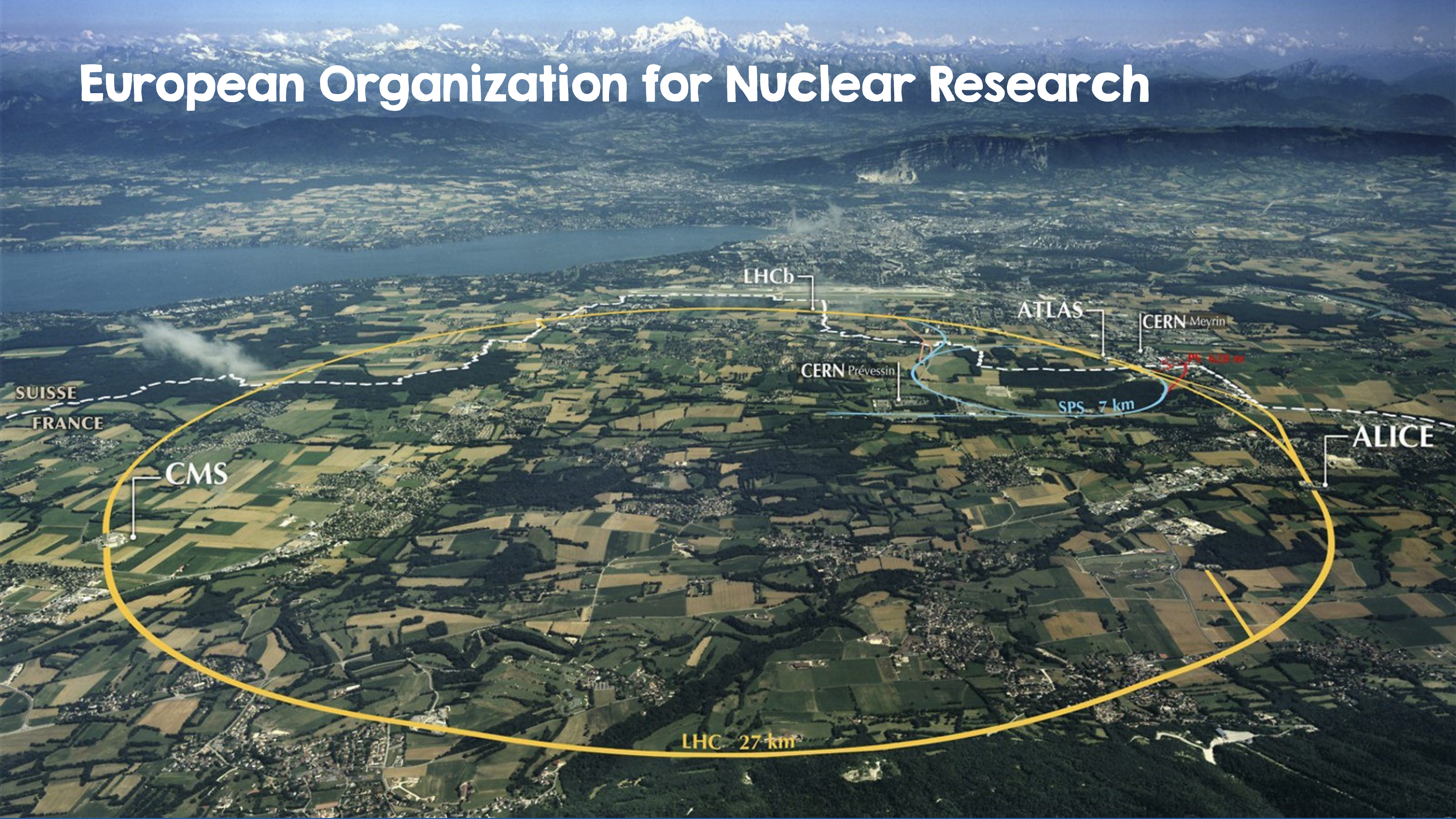
European Organization for Nuclear Research

- **World largest particle physics laboratory**
- **Founded in 1954**
- **23 member states**
- **Fundamental research in physics**



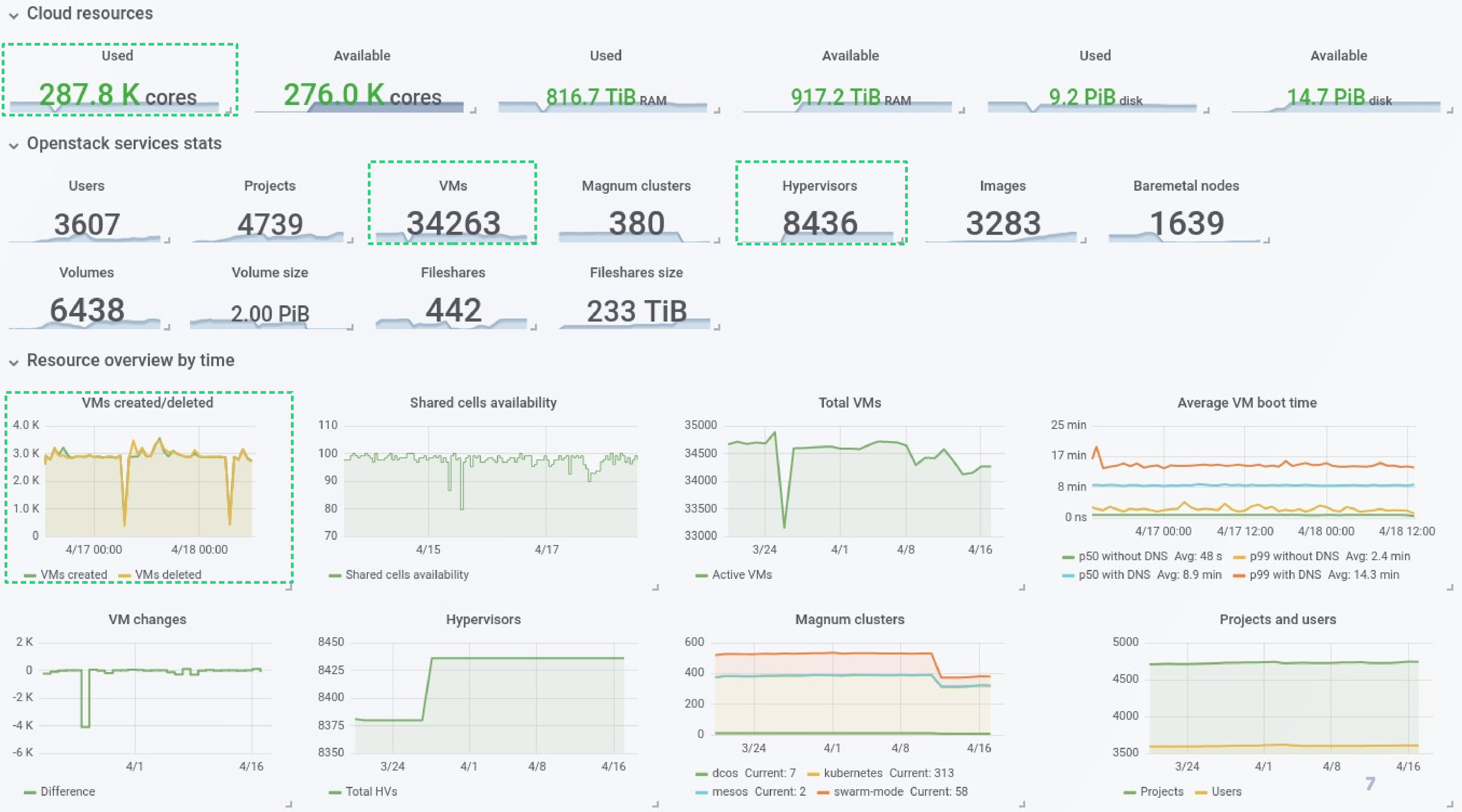


European Organization for Nuclear Research



CERN Cloud Service

- Infrastructure as a Service
- Production since July 2013
- CentOS 7 based
- Geneva and Wigner Computer centres
- Highly scalable architecture > 70 nova cells
  - 2 regions new
- Currently running Rocky release



Cloud resources

Used	Available	Used	Available	Used	Available
287.8 K cores	276.0 K cores	816.7 TiB RAM	917.2 TiB RAM	9.2 PiB disk	14.7 PiB disk

Openstack services stats

Users	Projects	VMs	Magnum clusters	Hypervisors	Images	Baremetal nodes
3607	4739	34263	380	8436	3283	1639

Volumes	Volume size	Fileshares	Fileshares size
6438	2.00 PiB	442	233 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: 48 s
p99 without DNS Avg: 2.4 min
p50 with DNS Avg: 8.9 min
p99 with DNS Avg: 14.3 min

VM changes

Difference

Hypervisors

Total HVs

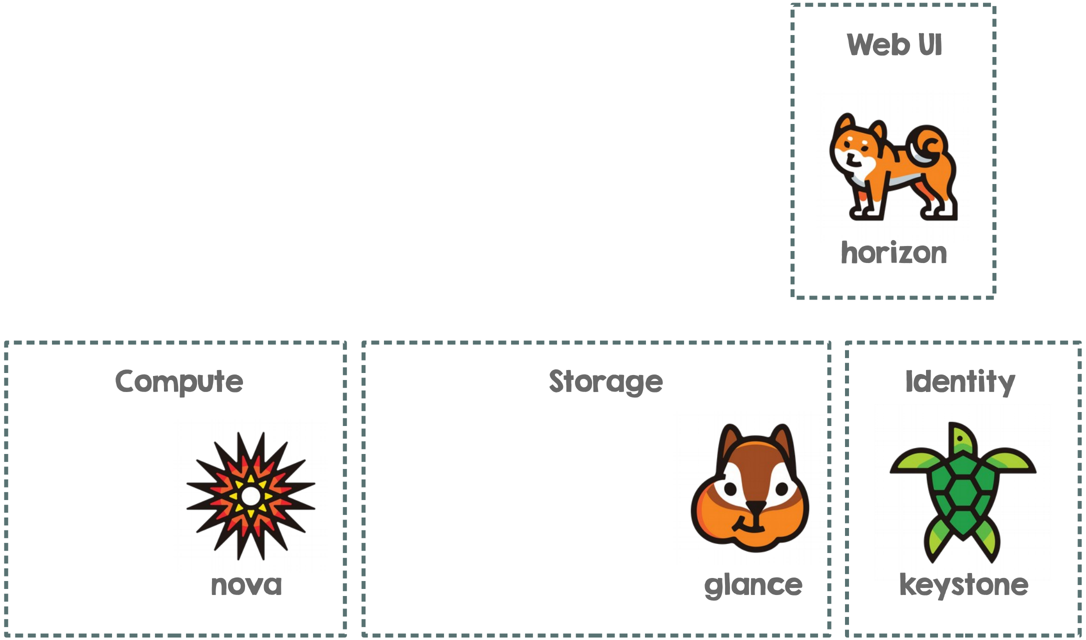
Magnum clusters

dcos Current: 7
kubernetes Current: 313
mesos Current: 2
swarm-mode Current: 58

Projects and users

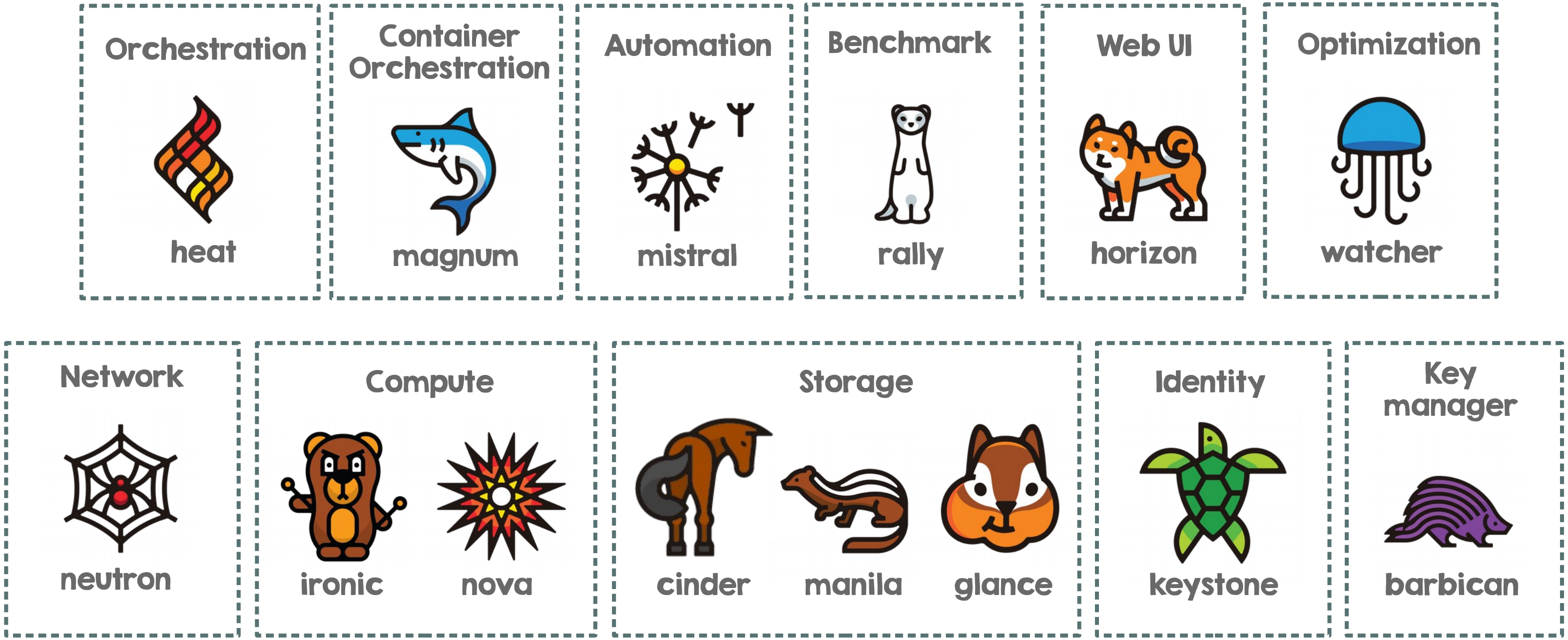
Projects Users

CERN Cloud Infrastructure - initial offering



IaaS

CERN Cloud Infrastructure



IaaS+

Orchestration
heat

Container Orchestration
magnum

Automation
mistral

Benchmark
rally

Web UI
horizon

Optimization
watcher

IaaS

Network
neutron

Compute
ironic
nova

Storage
cinder
manila
glance

Identity
keystone

Key manager
barbican

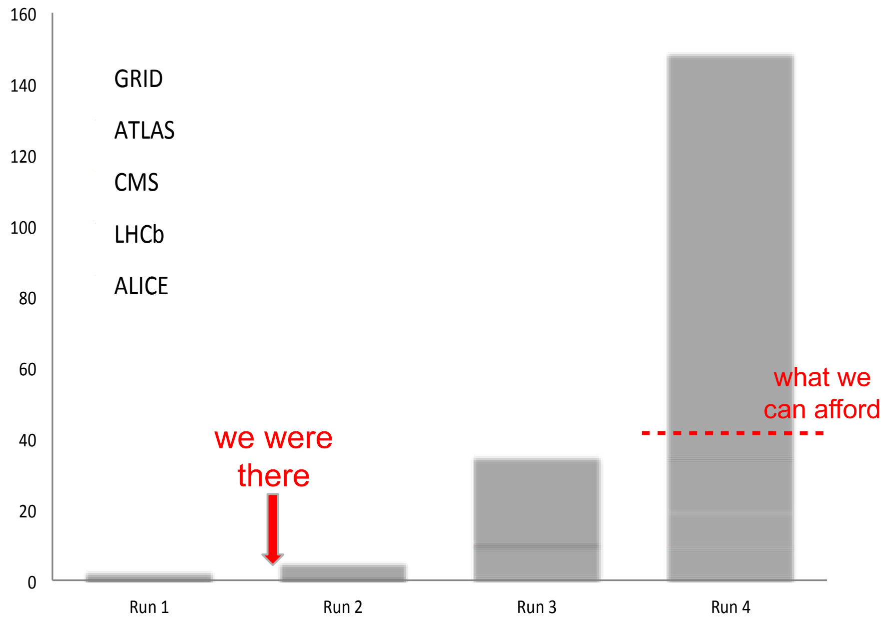
Back in 2012

- LHC Computing and Data requirements where increasing
- Constant team size
- LS one ahead next window on 2019
- Other deployments have surpassed CERN

3 core areas:

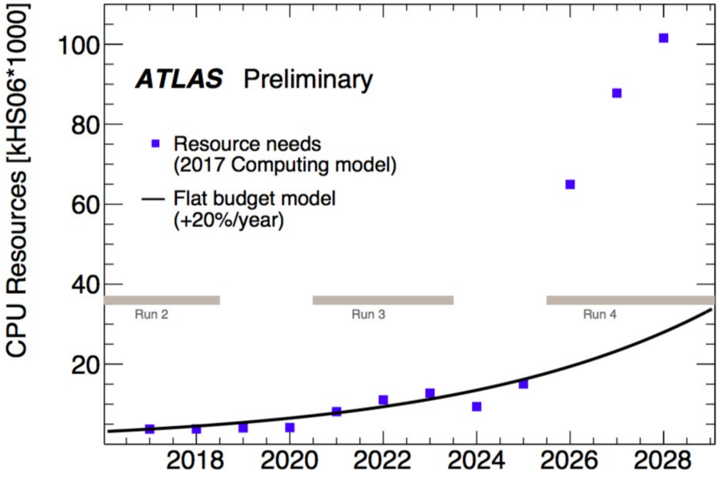
- Centralized Monitoring
- Configuration management
- IaaS based on OpenStack

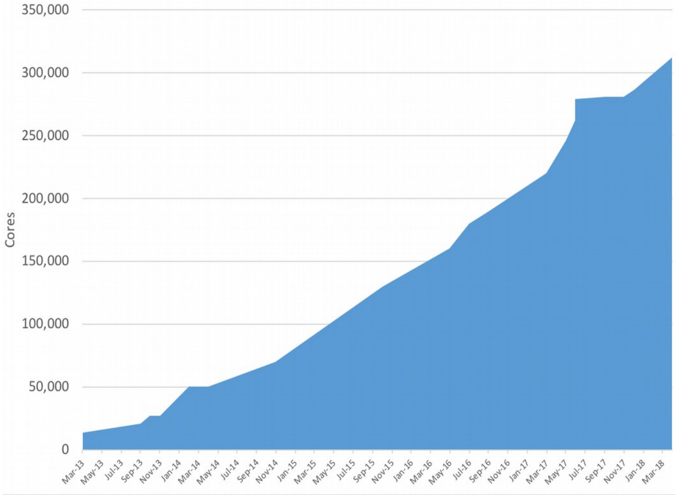
“All servers shall be virtual!”



Situation now

- ~300k core cloud and increasing
  - Addition of new services
  - Continuous improvements on existing ones
- No change in number of staff
- Improvement areas
  - Code efficiency
  - Improve algorithms with Machine learning
  - Use of Compute accelerators GPUs / FPGAs
  - Resource availability

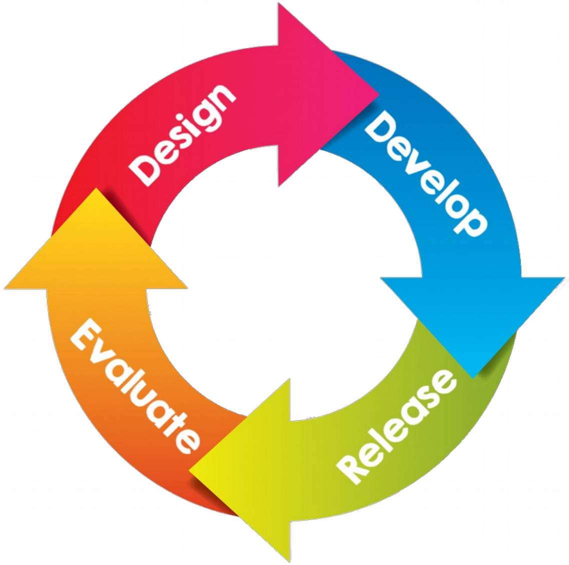




Improve resource availability

- Continuous improvement process
  - Evaluate current cloud status
  - Find room for improvement
  - Develop new solutions and services
  - Make those services available to our users

- Get the most of cloud resources
  - Performance
  - Availability



Automation

mistral

Optimization

watcher

Preemptibles

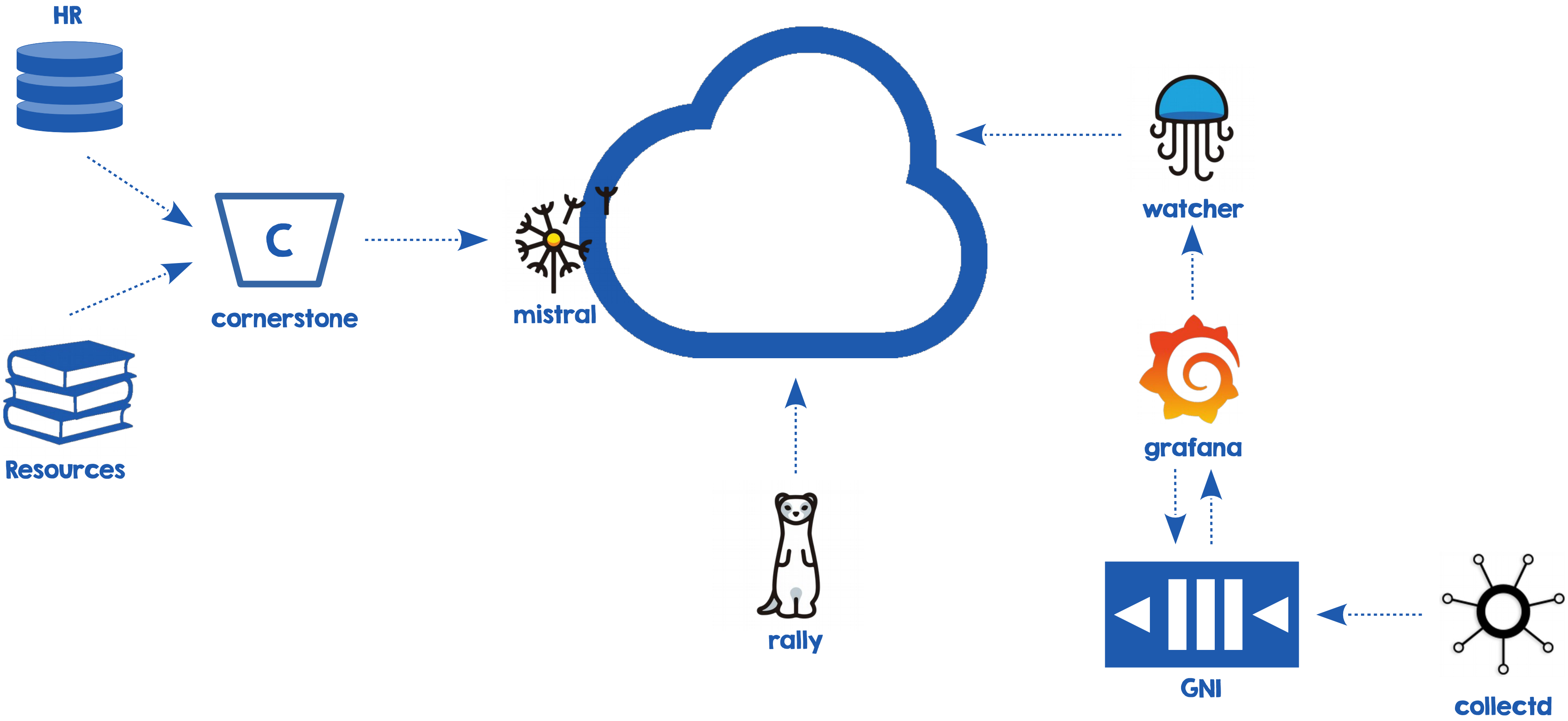
404
Image not found

aardvark

Baremetal Containers

Ironic + magnum

CERN Cloud Automation



Main objectives of automation

- Simplify resource management
  - Focus on getting the last bit of performance
- Optimize user experience
- Maximize resources available
  - Cleanup of orphaned resources
  - Expire unused resources

Resource Lifecycle Management

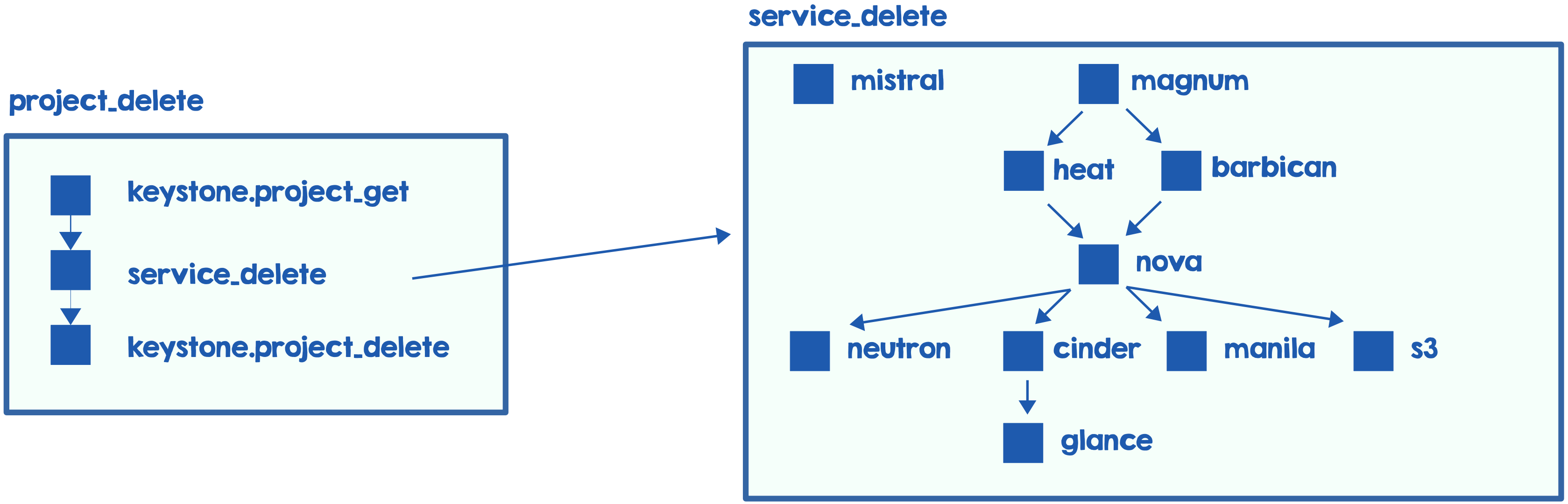
- Types of projects

	Affiliation Expired	User Disabled	User Deletion
Shared	Promote	-	-
Personal	-	Stop	Delete

- Provisioning and cleanup in Mistral workflows
  - Service inter-dependencies
  - Multi-region support new

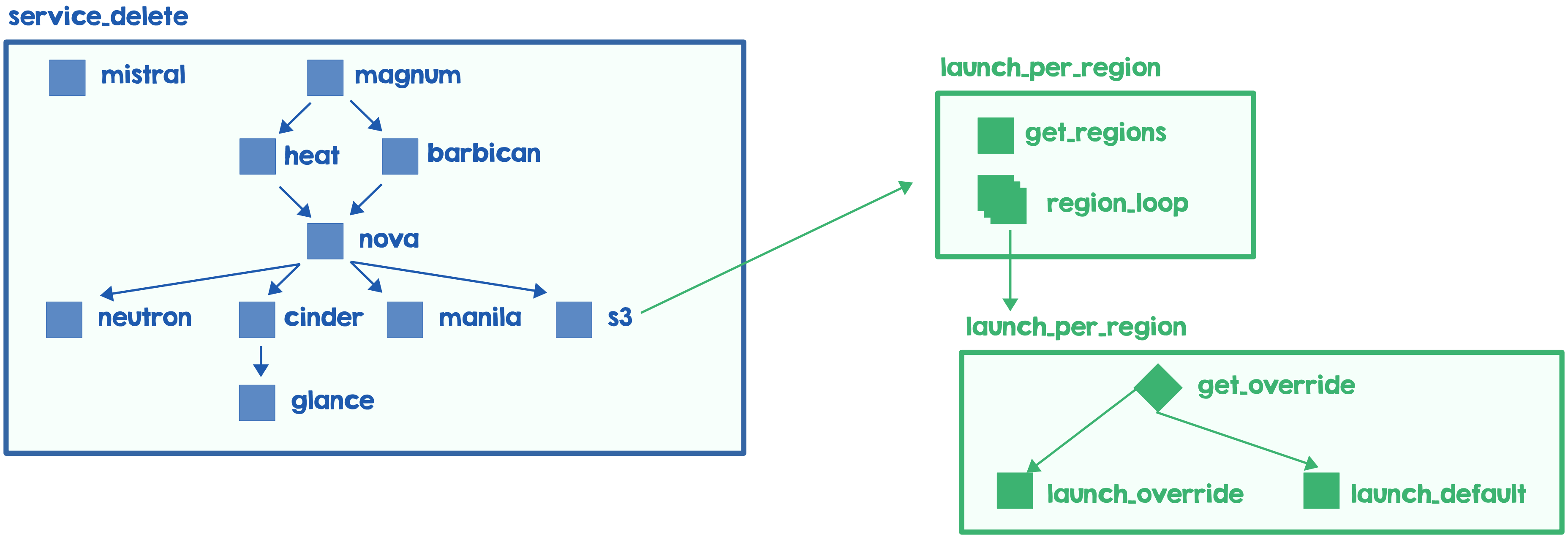
Resource Lifecycle Management in detail

- Set of workbooks interconnected to manage
  - Projects
  - Services

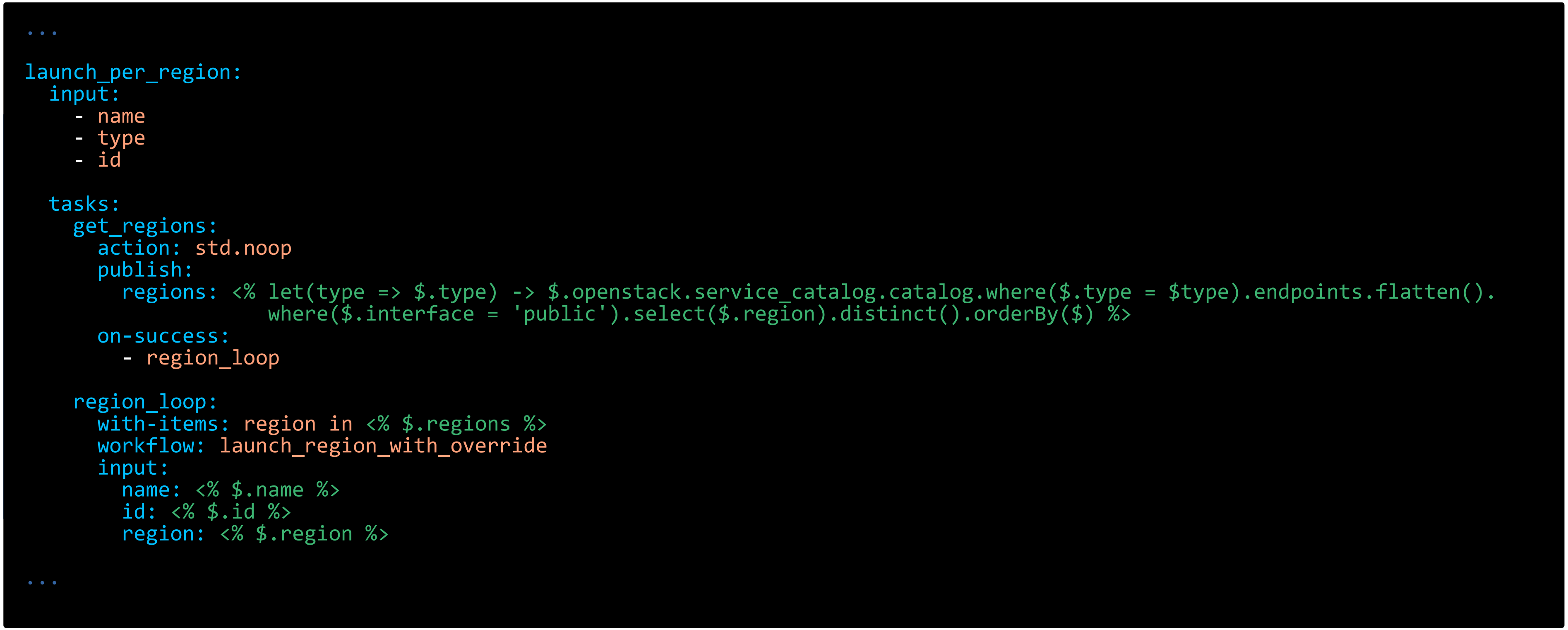


Multi region support

- We've just added a 2nd region



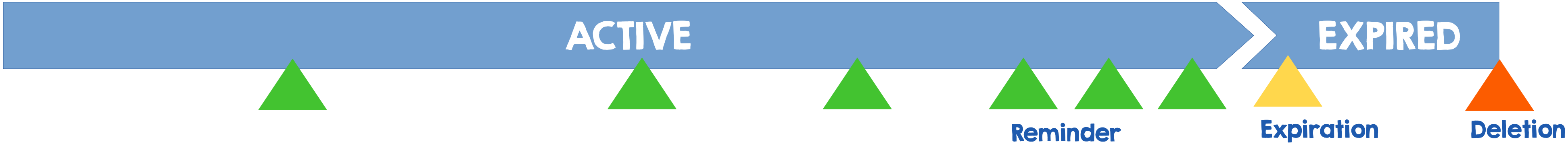
Multi region support (code)



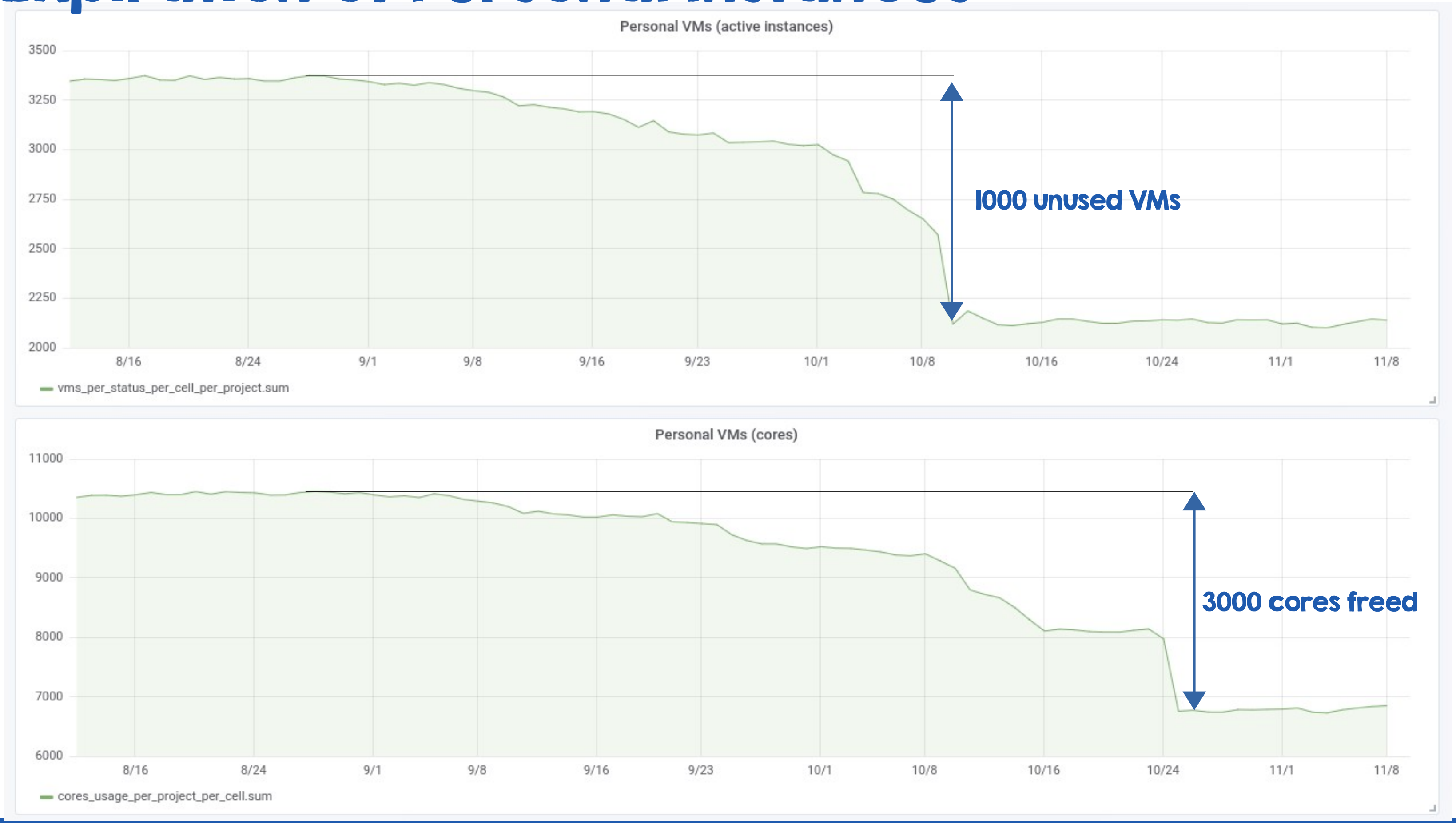
```
...

launch_per_region:
  input:
    - name
    - type
    - id

  tasks:
    get_regions:
      action: std.noop
      publish:
        regions: <% let(type => $.type) -> $.openstack.service_catalog.catalog.where($.type = $type).endpoints.flatten().
                    where($.interface = 'public').select($.region).distinct().orderBy($) %>
      on-success:
        - region_loop

    region_loop:
      with-items: region in <% $.regions %>
      workflow: launch_region_with_override
      input:
        name: <% $.name %>
        id: <% $.id %>
        region: <% $.region %>

...
```

Optimize resource availability - Expiration

- Each VM in a personal project has an expiration date
- Set shortly after creation and evaluated daily
- Configured to 180 days and renewable
- Reminder mails starting 30 days before expiration
- Implemented as a Workbook in Mistral



Expiration of Personal Instances



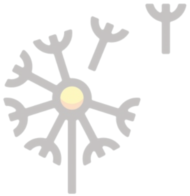
Personal VMs (active instances)

1000 unused VMs

Personal VMs (cores)

3000 cores freed

Automation



mistral

Optimization

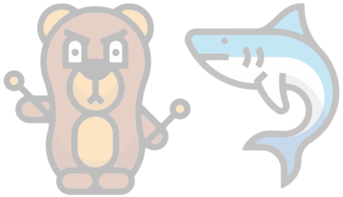
watcher

Preemptibles

404
Image not found

aardvark

Baremetal Containers



Ironic + magnum

Towards Optimization service at CERN

- Successful evaluation of Watcher service

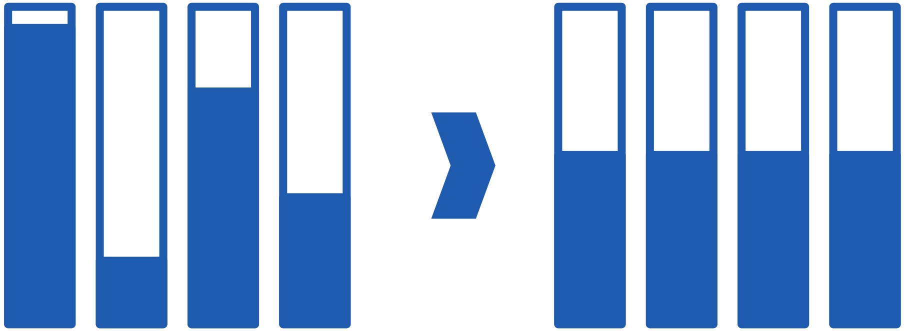
- Recently involved with upstream community
  - Corne Lukken @D4ntali0n

- Room for improvement
  - Execution at scale
  - Additional datasources
  - Strategy improvements

Get the most of the infrastructure

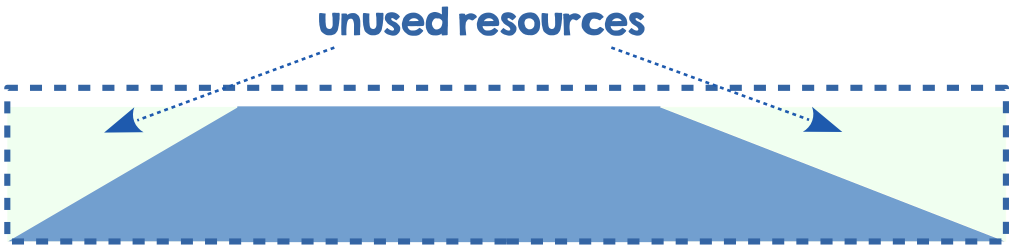
- Per-cell audit on the Cloud
  - Improve Cloud service user perception (fair share)
  - Early discovery of performance issues

- Dynamically adjust workloads in hyperconverged environments
  - Keeping free resources for IO
  - Avoid impact on compute
  - Automatic live-migration



Watcher strategy as preemptible scheduler?

- Use case:
  - Hardware procurement 2 times per year
  - Once provisioned, the users will start to use them
  - On decommission, they are slowly being drained
- Issue:

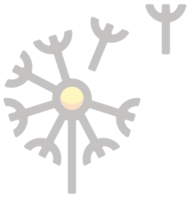


- Watcher automatic audit could create preemptible instances with BOINC workloads

Optimization service status

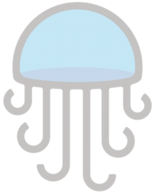
- Execution at scale
  - ✓ Audit Scope
- Datasources
  - ✓ Grafana-proxy
- Strategies
  - ✓ Per-cell workload balancer
  - 🛠 Hyperconverged balancer
  - 🛠 Preemptible scheduler

Automation



mistral

Optimization



watcher

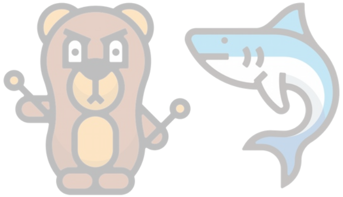
Preemptibles

404

Image not found

aardvark

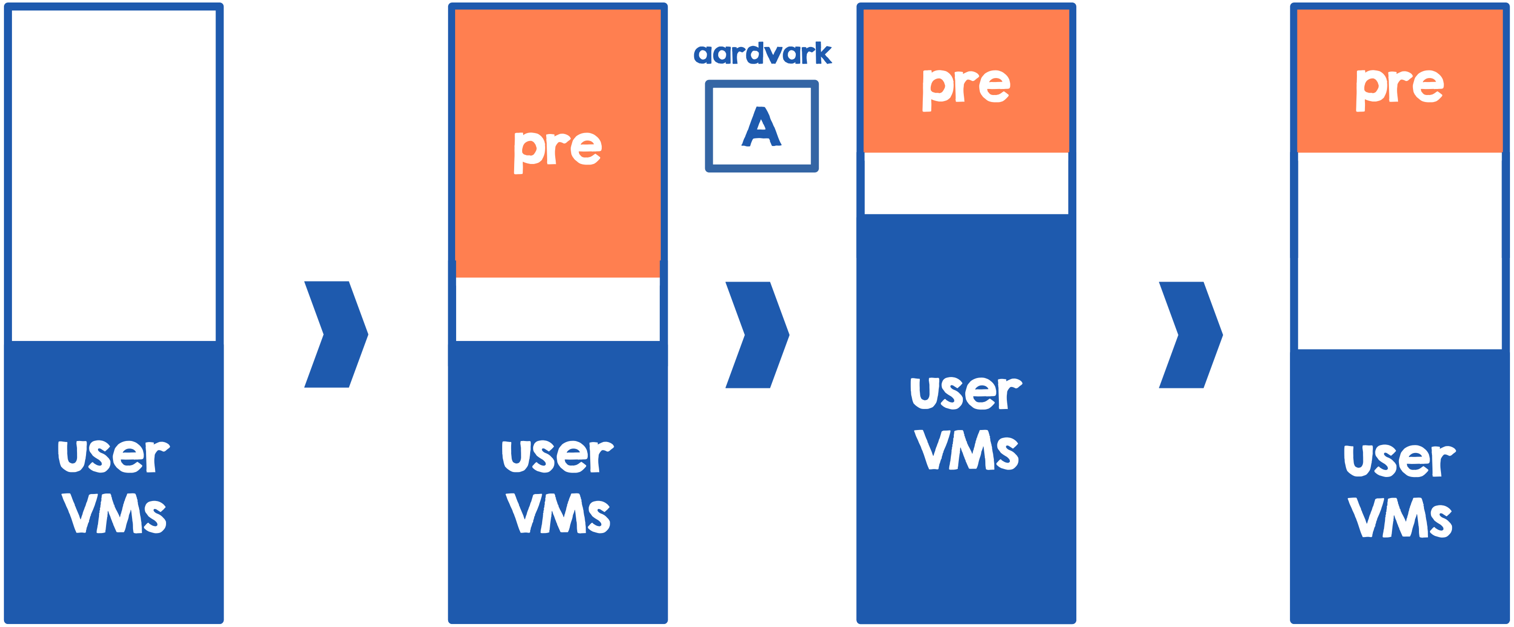
Baremetal Containers

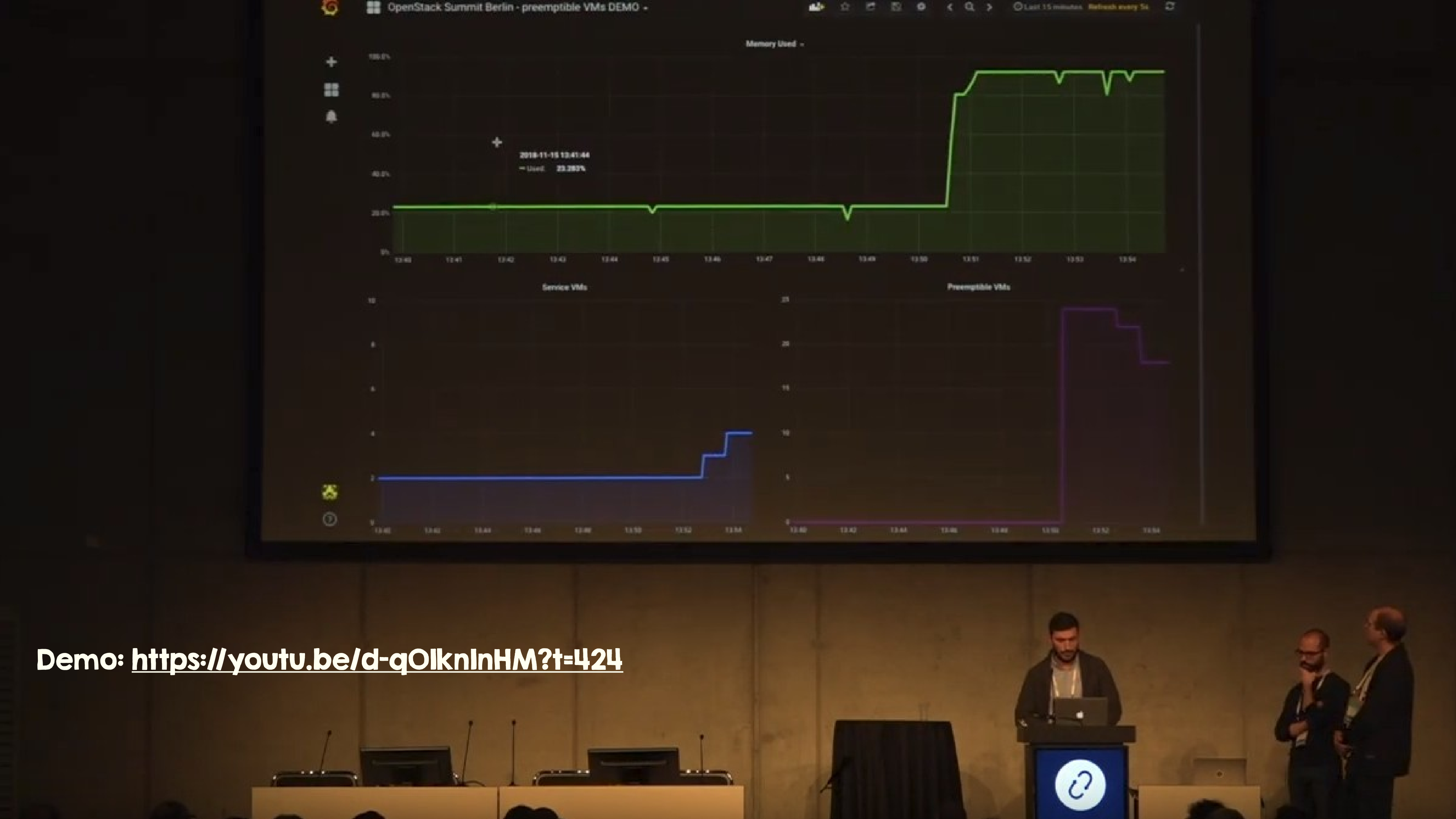


Ironic + magnum

Preemptibles

404
Image not found





Demo: https://youtu.be/d-qOIknInHM?t=424

Preemptible Service Status

404
Image not found

- Upstream work
  - Add instance state PENDING

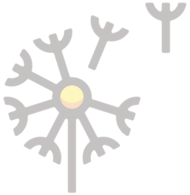
    ✓ spec 🛠 code
  - Allow rebuild instances in cell0

    🛠 spec - code
- Users

  🛠 LHC@home

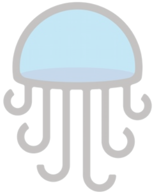
  🛠 Opportunistic Batch

Automation



mistral

Optimization



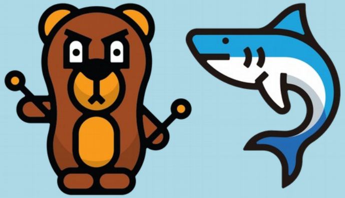
watcher

Preemptibles

404
Image not found

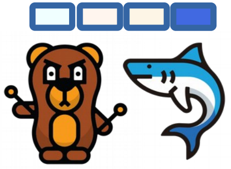
aardvark

Baremetal Containers



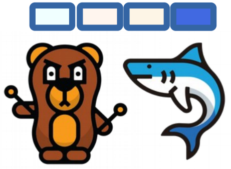
Ironic + magnum

Containers on Baremetal



- Get the last bit of performance
  - Put together OpenStack managed containers and baremetal
- Batch farm runs in VMs as well
  - 3% performance overhead, 0% with containers
- Federated kubernetes for cluster integration

Containers on Baremetal Status



- Typical deployment
  - Masters in VMs
  - Minions in Physical nodes
- Users
  - Batch farm
    - ✓ Clusters available
    - 🛠 Adapting own Terraform templates
    - HTCondor queues
    - Job submission

One more thing...

Tech Blog

- Backfilling Kubernetes Clusters by Ricardo Rocha
  - https://techblog.web.cern.ch/techblog/post/priority-preemption-boinc-backfill/

- Splitting the CERN OpenStack Cloud into Two Regions by Belmiro Moreira
  - https://techblog.web.cern.ch/techblog/post/region-split/

- Expiry of VMs in the CERN cloud by José Castro León
  - https://techblog.web.cern.ch/techblog/post/expiry-of-vms-in-cern-cloud/

- Maximizing resource utilization with Preemptible Instances by Theodoros Tsioutsias
  - https://techblog.web.cern.ch/techblog/post/maximizing-resource-utilization-with/

Thank you

gitlab.cern.ch/cloud-infrastructure

cern.ch/techblog

jose.castro.leon@cern.ch
@josecastroleon

spyridon.trigazis@cern.ch
@strigazi

CERN