CERN

Openstack @ CERN

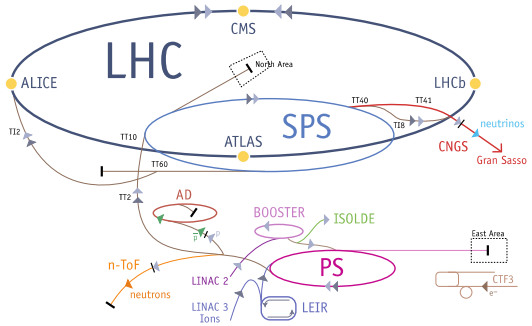
José Castro León
CERN Cloud Infrastructure

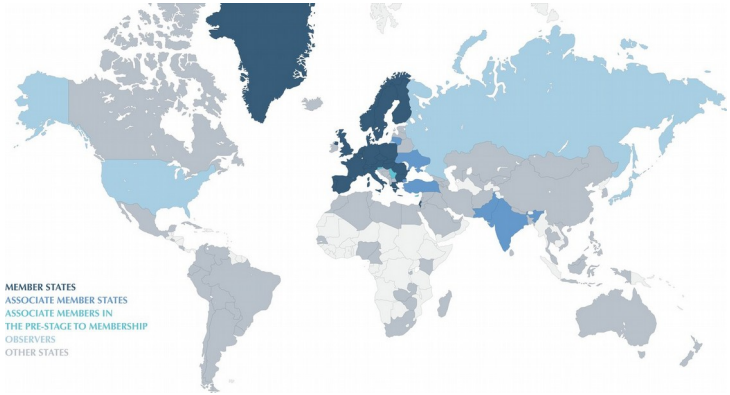
Outlines

- Introduction
- CERN Cloud service
  - Service operations
  - Service automation
  - Baremetal provisioning
  - Storage Services
- Upcoming work
- Q & A

European Organization for Nuclear Research

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





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

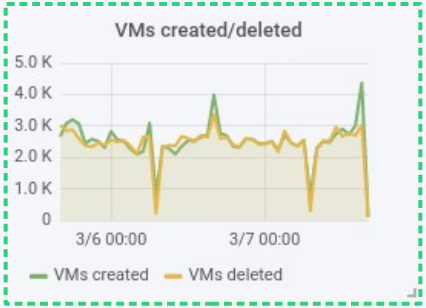
Used	Available	Used	Available	Used	Available
295.7 K cores	285.0 K cores	840.2 TiB RAM	927.9 TiB RAM	9.6 PiB disk	15.2 PiB disk

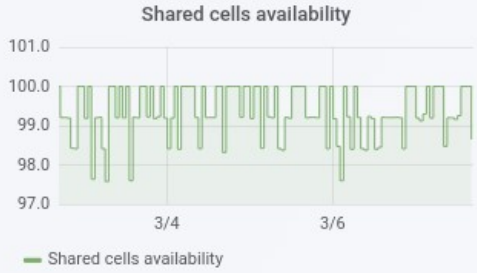
Openstack services stats

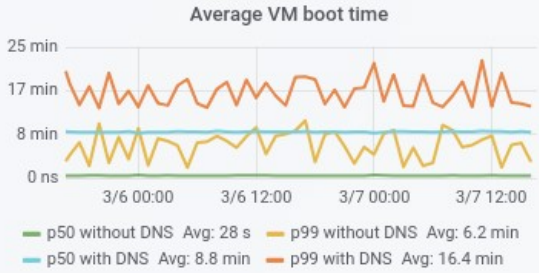
Users	Projects	VMs	Magnum clusters	Hypervisors	Images	Baremetal nodes
3572	4718	35617	543	8848	3349	1602

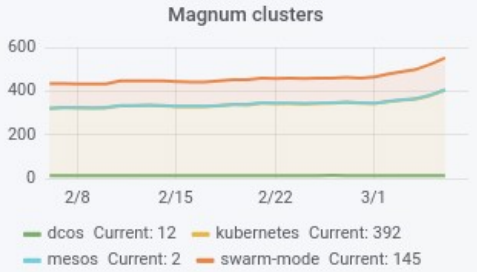
Volumes	Volume size	Fileshares	Fileshares size
6386	2.02 PiB	442	232 TiB

Resource overview by time









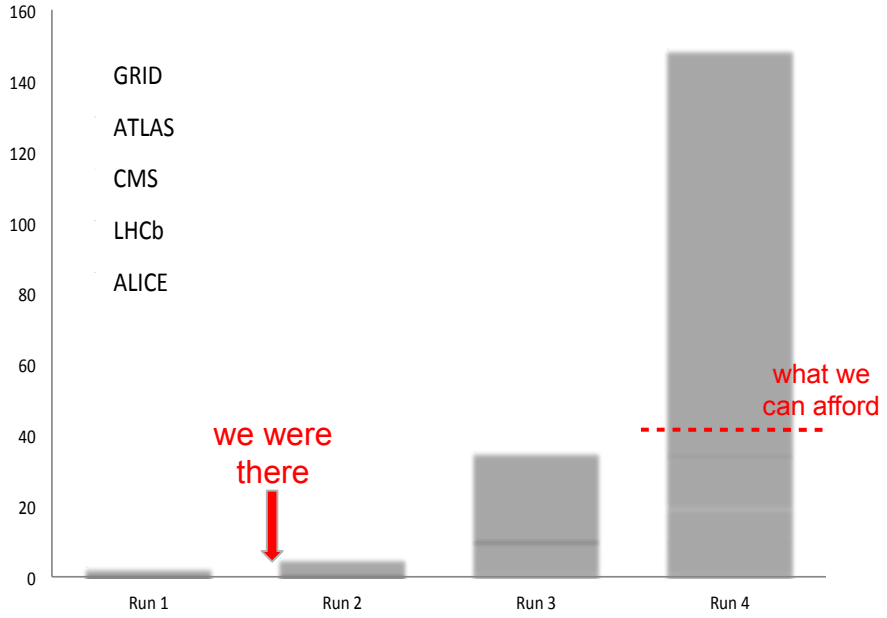
Back in 2012

- 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:

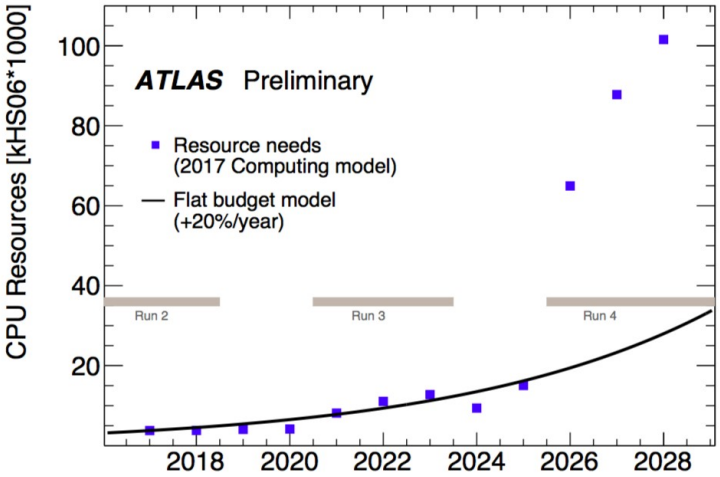
- Centralized Monitoring
- Configuration management
- IaaS based on OpenStack

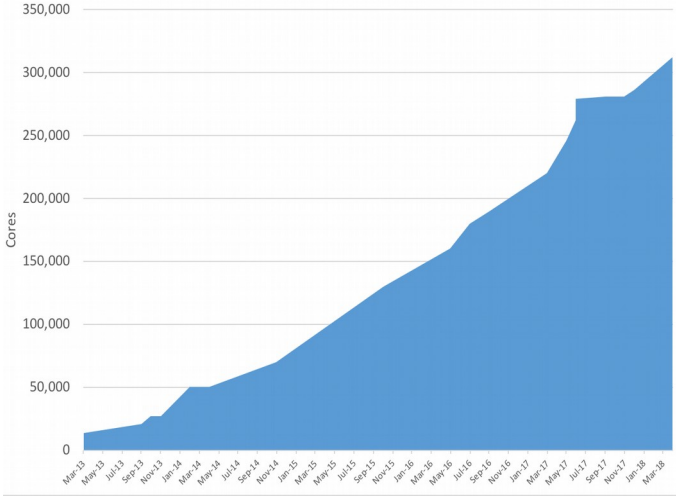
"All servers shall be virtual!"



Situation now

- 300k core cloud and increasing
  - Addition of new services
  - Continuous improvements on existing ones
- No change in number of staff
- Follow technological trends
  - Incorporate new use cases
  - Integrate them into ecosystem
  - Improve current infrastructure

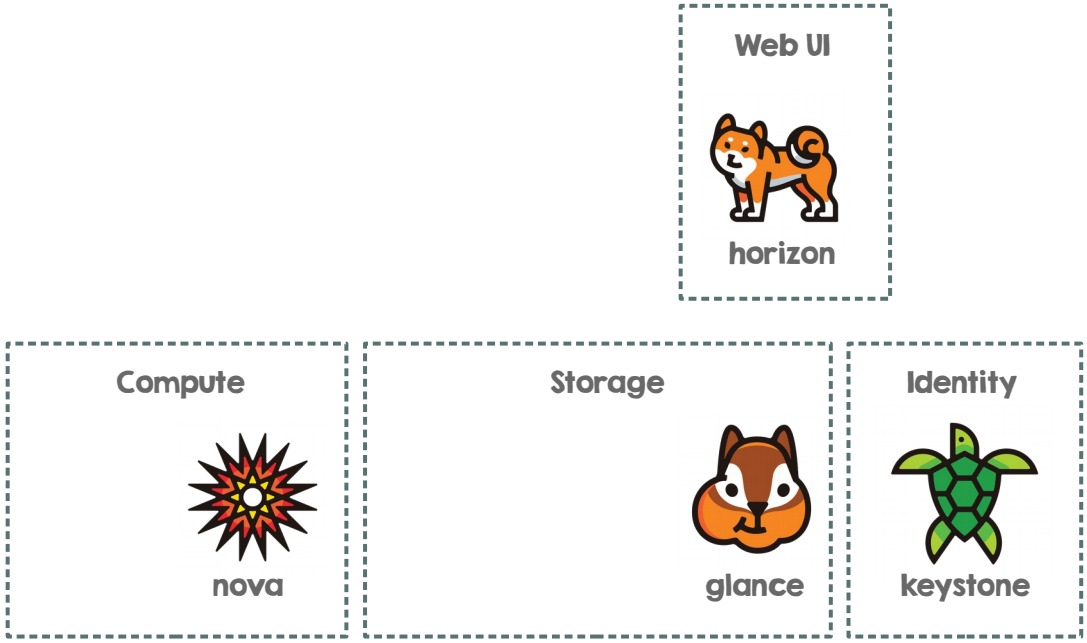




CERN Cloud Infrastructure - initial offering

IaaS+

IaaS



CERN Cloud Infrastructure - now

IaaS+

- Orchestration: heat
- Container Orchestration: magnum
- Automation: mistral
- Test deploy: rally
- Web UI: horizon
- Optimization: watcher
- HA: masakari

IaaS

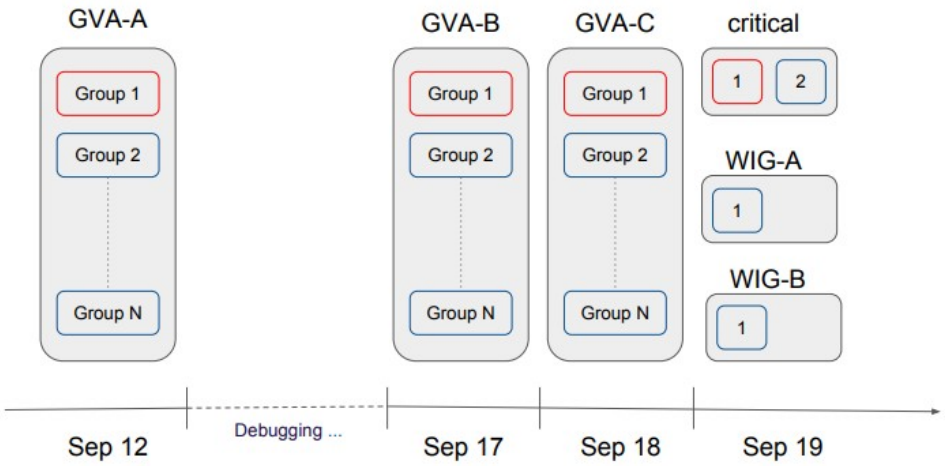
- Network: neutron
- Compute: ironic, nova
- Storage: cinder, manila, glance
- Identity: keystone
- Key manager: barbican

Service operations

- Availability techniques for users
  - 3 availability zones in Meyrin, 2 in Wigner + critical area
- Eat our own dogfood (use same tools as the rest of IT)
- Automation "likes"
  - Delegate some administrative tasks
  - Detect and fix known issues
  - Communicate with end users
- Quite some global campaigns:
  - Consolidation to KVM, Spectre/Meltdown and LITF

Patch the entire cloud

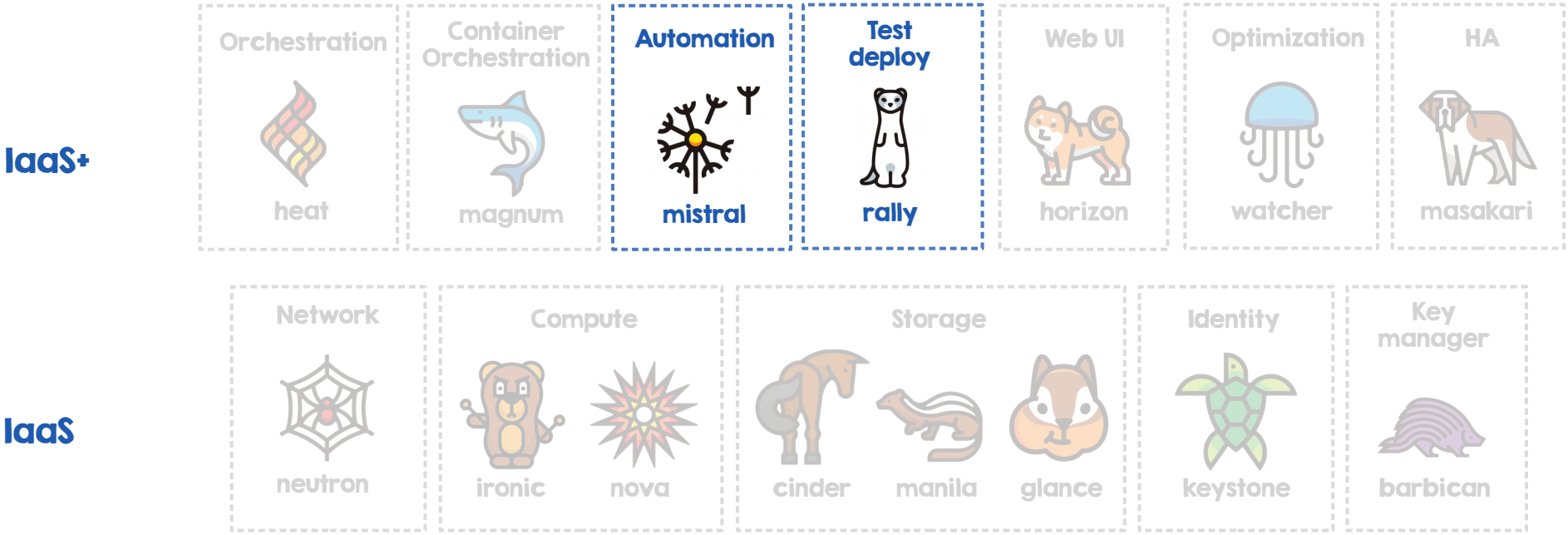
- Patching the cloud after Spectre/Meltdown/LITF
  - LITF ~1100 servers rebooted (~11.5k VMs)
- Validated on QA environment
- Review steps
  - Install latest kernel, make sure is default
  - Configure litf_full kernel boot option
  - Reboot and wait
  - Check hypervisors and VMs



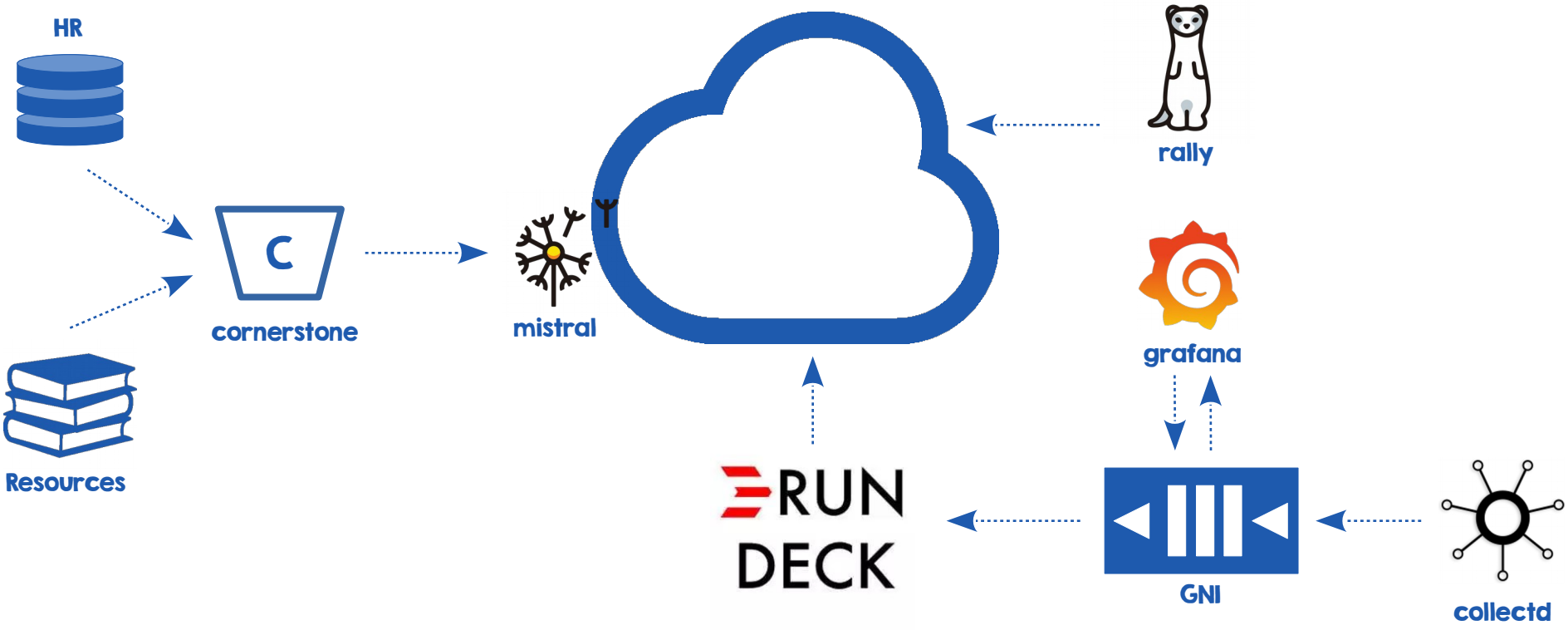
Patch the entire cloud feedback

- LITF announcement upfront to all user community
- ASDF announcements and updates
- Updates via SSB (and Town Square in mattermost)
- Reachable by tickets
  - Only 5 service teams tickets on LITF campaign
- No serious issues found during campaign
  - Performance impact after disabling SMT

Automation in the CERN Cloud



Automation in the CERN Cloud - architecture



Automation in the CERN Cloud - topics

Host and Service monitoring

Resource Lifecycle management

Optimize resource availability

Improve VM availability and Performance

Host and Service Monitoring

- Monitor HW events with Collectd
- Collect service logs through Flume
- General Notification Infrastructure
  - Support tickets for repairs
- Service alarms in Grafana
- Rundeck jobs
  - Time-scheduled jobs to fix common issues
  - Offload ticket handling
  - Schedule interventions

RunDeck: Task delegation

- Rely on Rundeck for offloading tasks to different teams
  - Procurement
  - Repair Team
  - Resource Coordinator
  - Cloud Service operations
- Example: disk replacement

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

Resource Lifecycle Management for end user

REQUEST NEW PROJECT

Request new project

General

Experiment or department

Name of the project

Description

Owner (primary account)

E-group(s) of project members

Compute

Number of instances	25
Number of cores	25
RAM	50

Volumes

Volume type description

Name:	standard
Usage:	default
Max IOPS:	100
Max Throughput:	80 MB/s

	Number	Space	
standard	0	0	GB
cp1	0	0	GB
cpio1	0	0	GB
crypt	0	0	GB
io1	0	0	GB
wig-cp1	0	0	GB

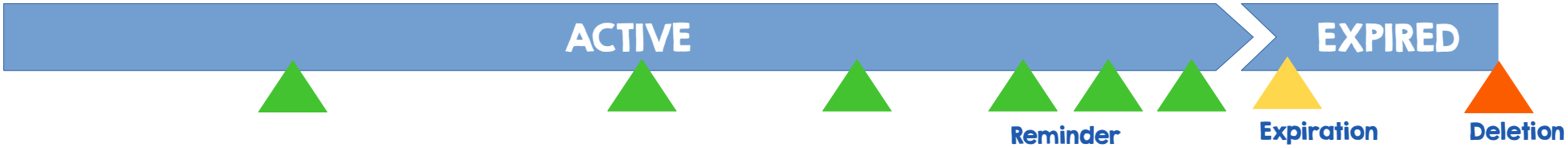
servicenow

RUN DECK

mistral

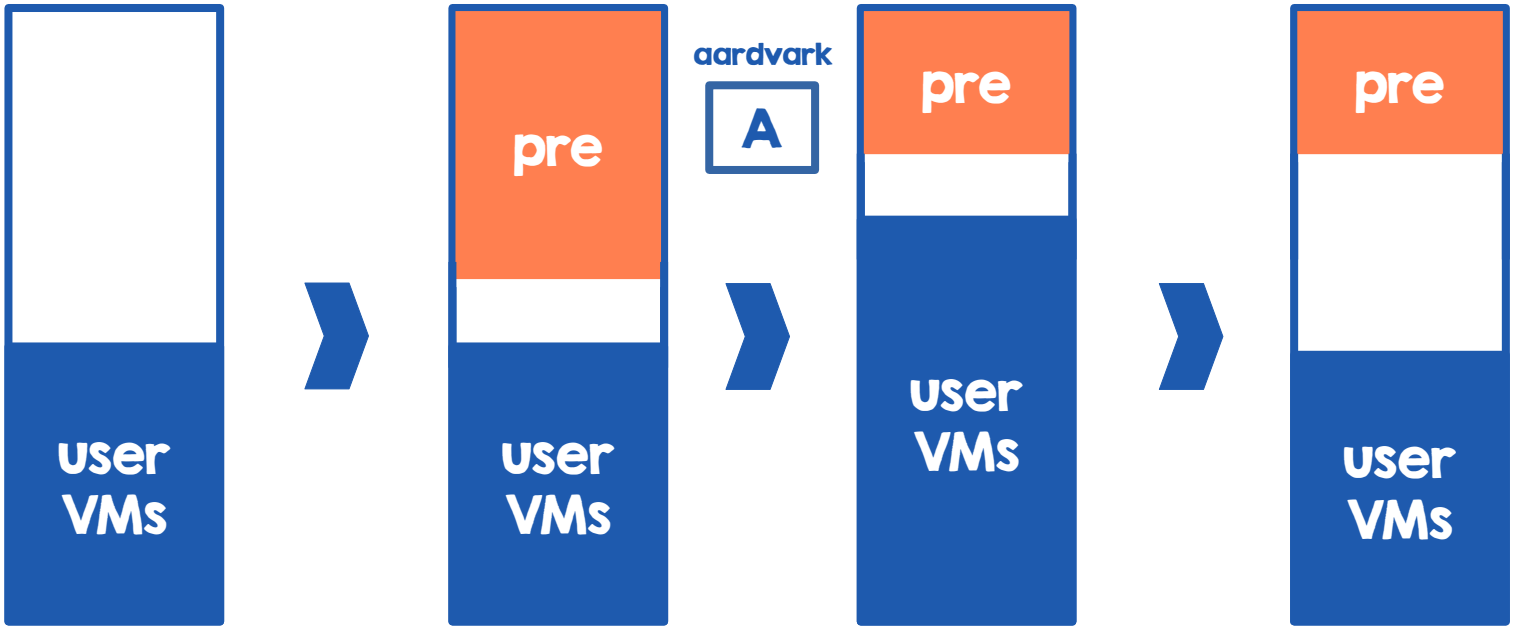
Optimize resource availability - Expiration

- 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 on a Workbook in Mistral

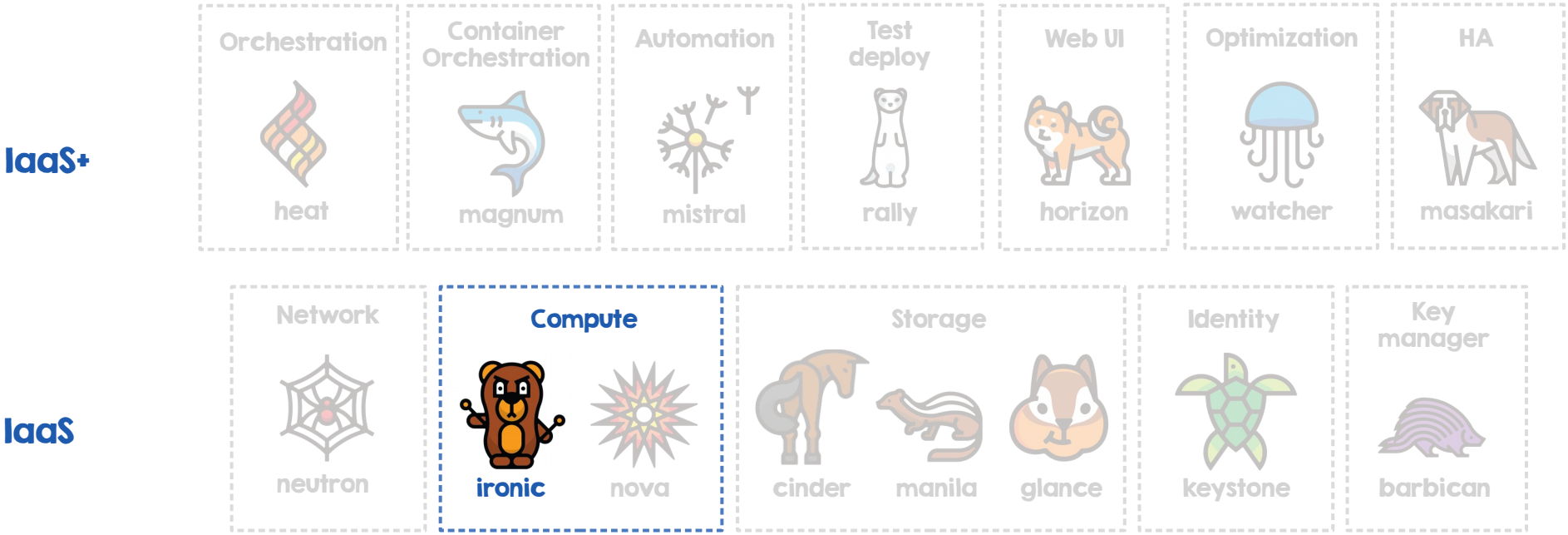


Expiration of Personal Instances

Improve Cloud utilization



Baremetal provisioning



Why baremetal provisioning?

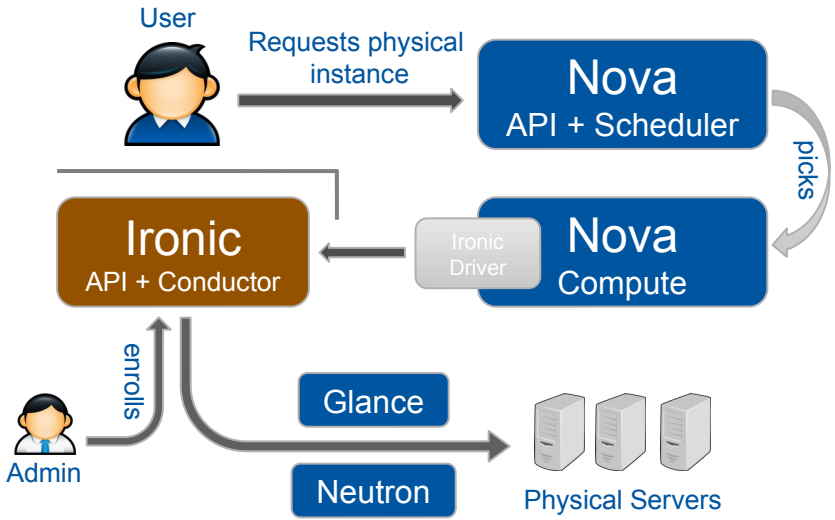
ironic

- Vms not sensible/suitable for all of our use cases
  - Storage nodes, HPC clusters,
- Complete our service offering
  - Physical nodes (in addition to VMs and containers)
  - OpenStack as single pane of glass
- Simplify hardware provisioning workflows
- Consolidate accounting & bookkeeping
  - Machine re-assignments will be easier to track

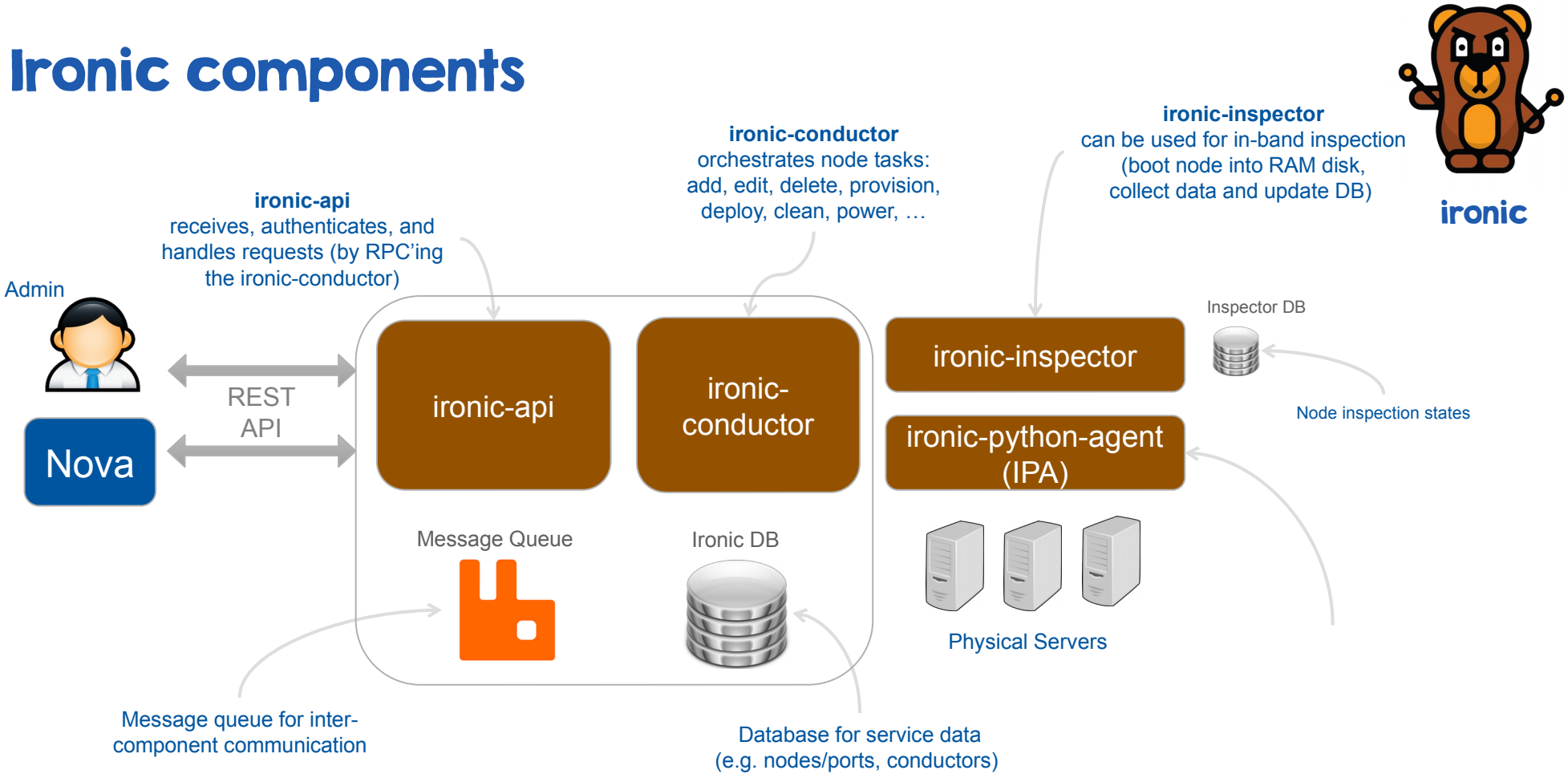
Baremetal as a Service

ironic

- Provision 'physical' instances
- Compute service manages physical servers as if they were virtual machines
- Users interfaces with Nova
  - Quotas, scheduling, …
- HW management via common interfaces
  - PXE, IPMI
  - Allows for unified interface to manage the whole park

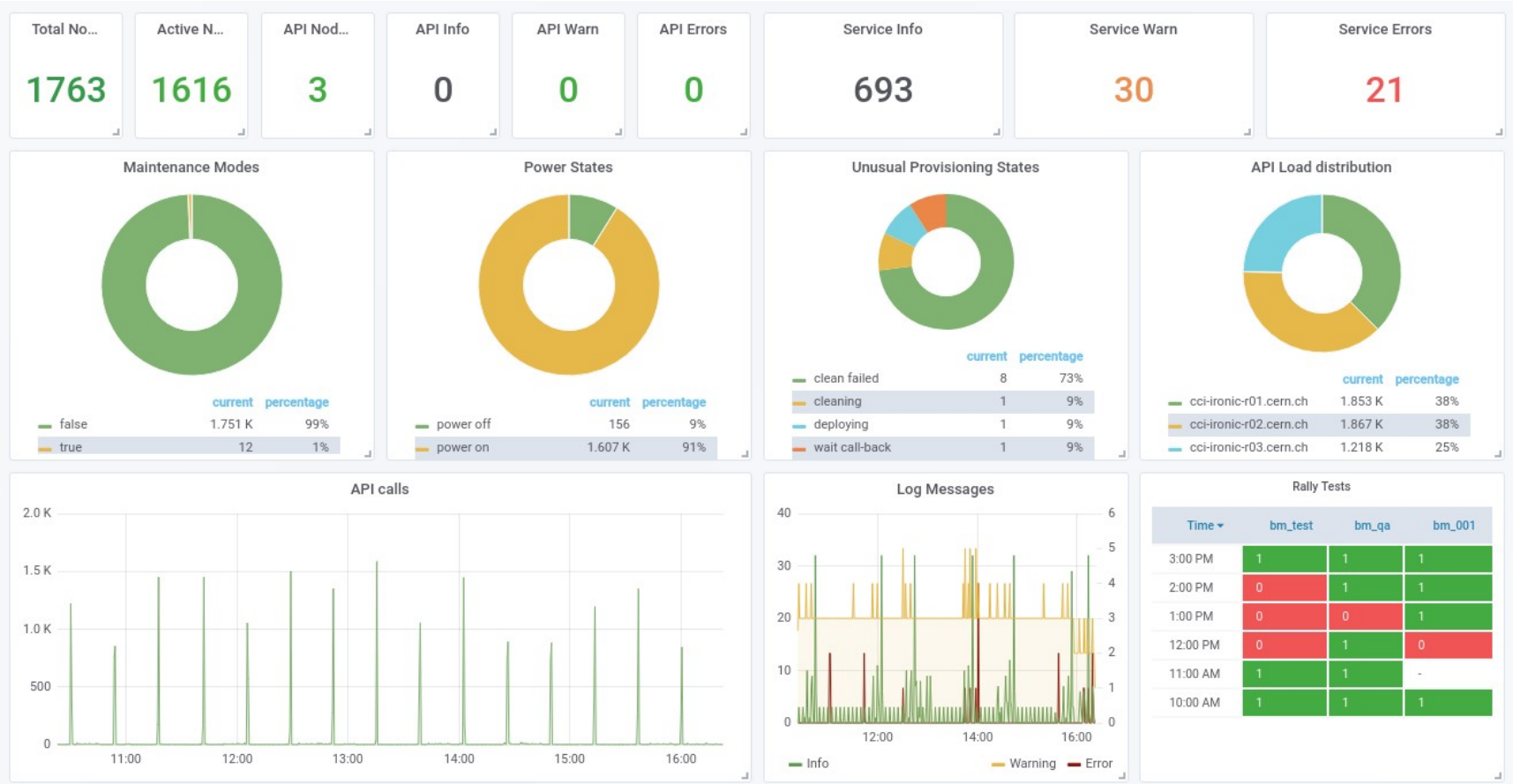


Ironic components



Ironic Service setup and status

ironic



Users:

- Cloud
- HPC
- Windows
- DB
- ...

A new use case: Containers on Baremetal

- Put together OpenStack managed containers and baremetal
- General service offer: managed clusters
  - Users get only K8s credentials
  - Cloud team manages the cluster and the underlying infra
- Batch farm runs in VMs as well
  - 3% performance overhead, 0% with containers
  - Federated kubernetes for hybrid cloud integration

Storage Services

IaaS+

- Orchestration: heat
- Container Orchestration: magnum
- Automation: mistral
- Test deploy: rally
- Web UI: horizon
- Optimization: watcher
- HA: masakari

IaaS

- Network: neutron
- Compute: nova, ironic
- **Storage**: **cinder**, **manila**, glance
- Identity: keystone
- Key manager: barbican

Block Storage as a Service

cinder

- Allows to add additional block devices to instances
- Connects to several Ceph clusters
- Volume types define QoS client capabilities and/or location
  - standard, iol, cpl, cpiol, wig-cpl, wig-cpiol, hyperc
- Volume type mapped to single cluster => no availability zones

- Last upstream contributions
  - Deferred deletion
  - Extension of RBD in-use volumes

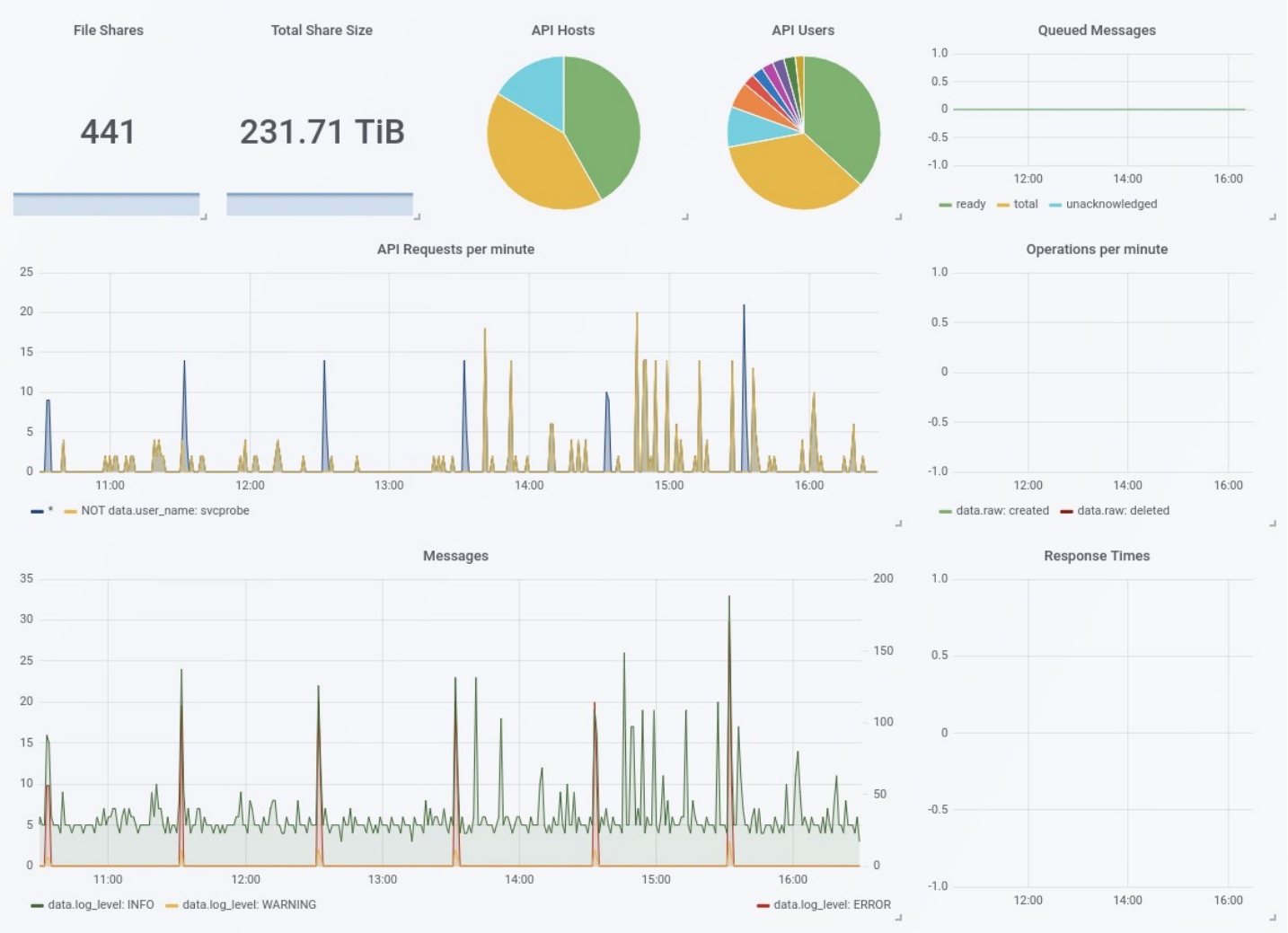

Volumes
12740
Volume Quota
4.04 PiB
Queued Messages
1.0
0.5
0
-0.5
-1.0
12:00
14:00
16:00
ready
total
unacknowledged
Unusual Volume States
API Requests
300
250
200
150
100
50
0
11:00
12:00
13:00
14:00
15:00
16:00
*
NOT data.user_name: svcprobe
Volume Operations
25
20
15
10
5
0
11:00
12:00
13:00
14:00
15:00
16:00
volume.create.end
volume.delete.end
volume.detach.end
volume.attach.end
Messages
300
200
100
0
11:00
12:00
13:00
14:00
15:00
16:00
data.log_level: ERROR
data.log_level: INFO
data.log_level: WARNING
API Users
API Hosts

cinder

File shares as a Service

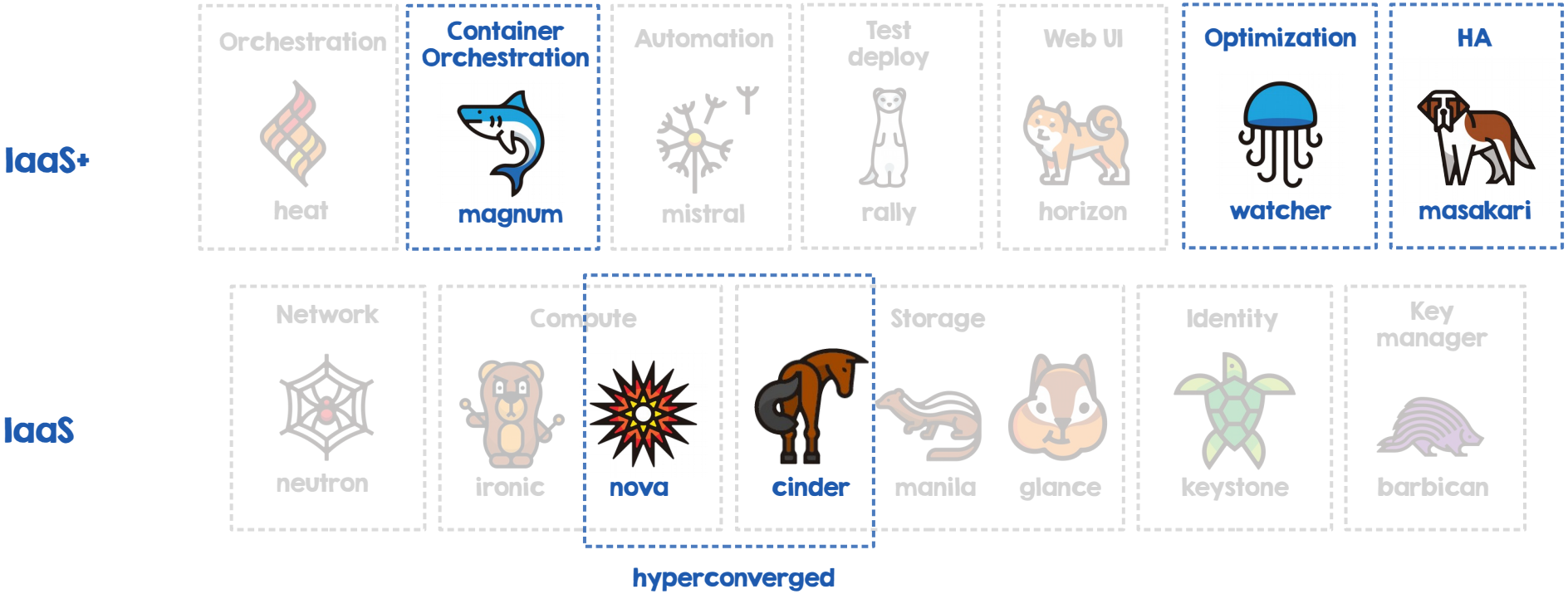
manila

- #1 user request
  - Block devices <> File Shares
- Share protocols
  - CephFS
- Use cases
  - High-Performance Computing
  - Replacement of NFS Filers
- Ongoing work
  - Enable NFS access through Ganesha


File Shares
441
Total Share Size
231.71 TiB
API Hosts
API Users
Queued Messages
1.0
0.5
0
-0.5
-1.0
12:00
14:00
16:00
ready
total
unacknowledged
API Requests per minute
25
20
15
10
5
0
11:00
12:00
13:00
14:00
15:00
16:00
*
NOT data.user_name: svcprobe
Operations per minute
1.0
0.5
0
-0.5
-1.0
12:00
14:00
16:00
data.raw: created
data.raw: deleted
Messages
35
30
25
20
15
10
5
0
200
150
100
50
0
11:00
12:00
13:00
14:00
15:00
16:00
data.log_level: INFO
data.log_level: WARNING
data.log_level: ERROR
Response Times
1.0
0.5
0
-0.5
-1.0
12:00
14:00
16:00

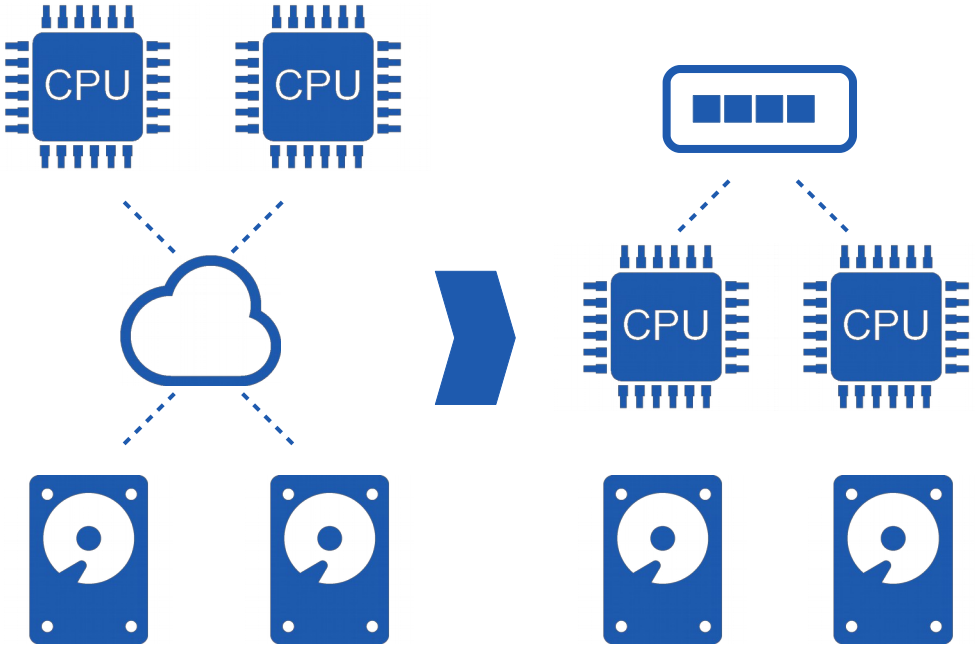
manila

Upcoming work



Hyperconverged Servers

- Compute + Storage Nodes
- Local Ceph pool
  - Instances
  - Volumes
- Ease management
- Small IO latency
- Increased Disk capacity
- Use cases:
  - DB and Storage services

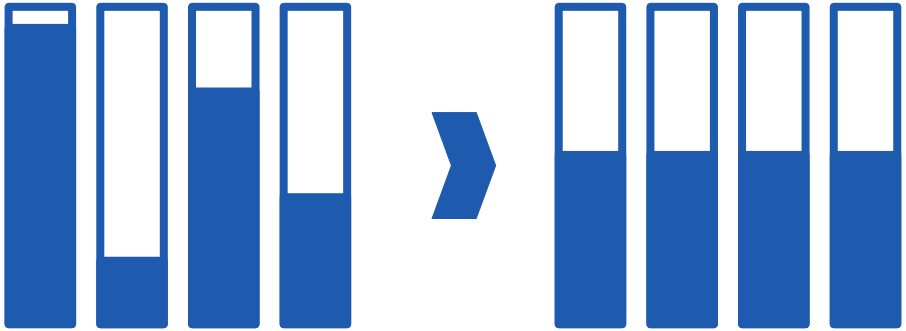


Get even more performance

watcher

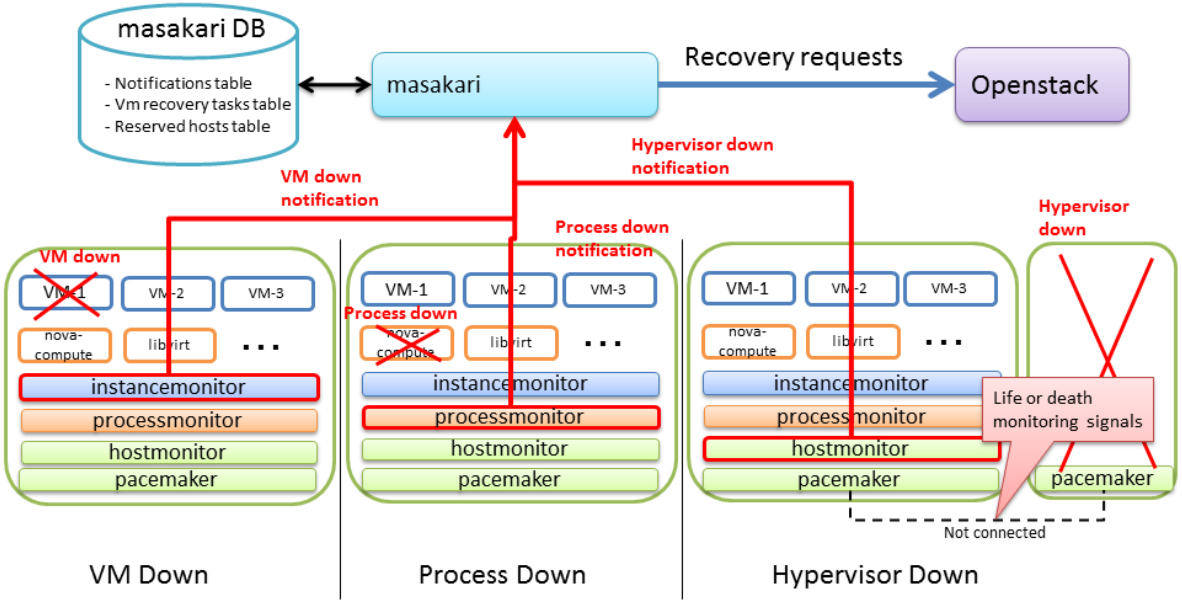
- Hyperconverged servers
  - Fixed CPU allocation for protecting IO operations

- Dynamically adjust CPU usage in the setup
  - Keeping free resources for IO
  - Avoid impact on compute
  - Automatic live-migration



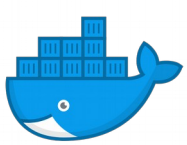
Improve resource availability

- **Automatic recovery requests**



Container orchestration Engines

- Creates clusters for container deployment
- Template based
  - Kubernetes, docker-swarm, DCOS
- Integration into ecosystem
  - CVMFS, Kerberos, CSI (CephFS)
- Ongoing work:
  - Automation (upgrades and healing)
  - Availability (Kubernetes multi master)
  - Central logging
  - Multitenancy


Here are the links

- https://gitlab.cern.ch/cloud-infrastructure/
  - cinder, horizon, ironic, keystone, mistral, neutron and nova
  - mistral-workflows
  - mistral-radosgw-actions (python-radosgw-admin)
  - hzrequestspanel
  - cci-scripts
  - cci-tools

Thank you

gitlab.cern.ch/cloud-infrastructure

techblog.web.cern.ch

jose.castro.leon@cern.ch

@josecastroleon

CERN

BACKUP SLIDES