
CERN

Hepix 2022

CERN Cloud Infrastructure - operations and service update

Jayaditya Gupta

CERN, 28th April 2022

Since the last HEPiX

- Brief overview
- Ironic upgrades
- Anomaly detection upgrades
- Glance upgrade
- Migration cycle tool (VM migration tool)

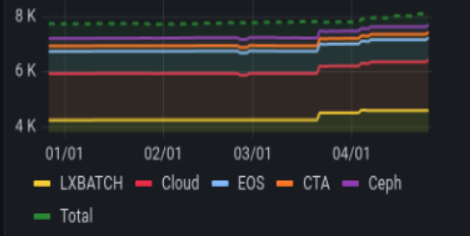
Openstack services statistics

Users	Projects	Kubernetes clusters	Images	Volumes	Volumes size	File Shares	File Shares ...	Object Stor...	Object Stor...
3330	4498	343	3533	7378	3.82 PB	5314	896 TB	447	47.2 TB

Servers				Cores			RAM			Batch		
Physical	Physical in use	Hypervisors	Virtual	Physical	Hypervisors	Virtual	Physical	Hypervisors	Virtual	Servers	Cores	RAM
9102	8124	1843	13375	453 K	52.8 K	87.2 K	1.87 PB	332 TB	207 TB	4902	261684	989 TB

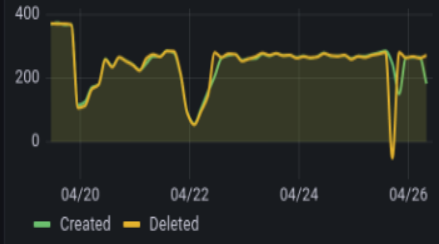
Time series

Physical servers — Last 4 months



Virtual Machines

Virtual Machines created/deleted — Last 7 days



Shared cells availability — Last 30 days

Kubernetes clusters

Volumes

File Shares

Object Store size

Openstack baremetal (Ironic) service upgrad

IRONIC

an OpenStack Community Project

What is ironic ?

- Ironic is an openstack program that aims to provide bare metal instead of virtual machines.

Use of ironic?

- Ironic allows physical servers to be managed as though they were virtual machines.
- API driven Installation of physical machines to support cloud infrastructure or tenant workloads.

Openstack baremetal (Ironic) service upgrades

Physical machines uses Ironic for whole onboarding process.

- Auto Registration
- Inventory
- Verification
- Burn-in
- Benchmarking

Provisioning of the machines was already done with Ironic before.

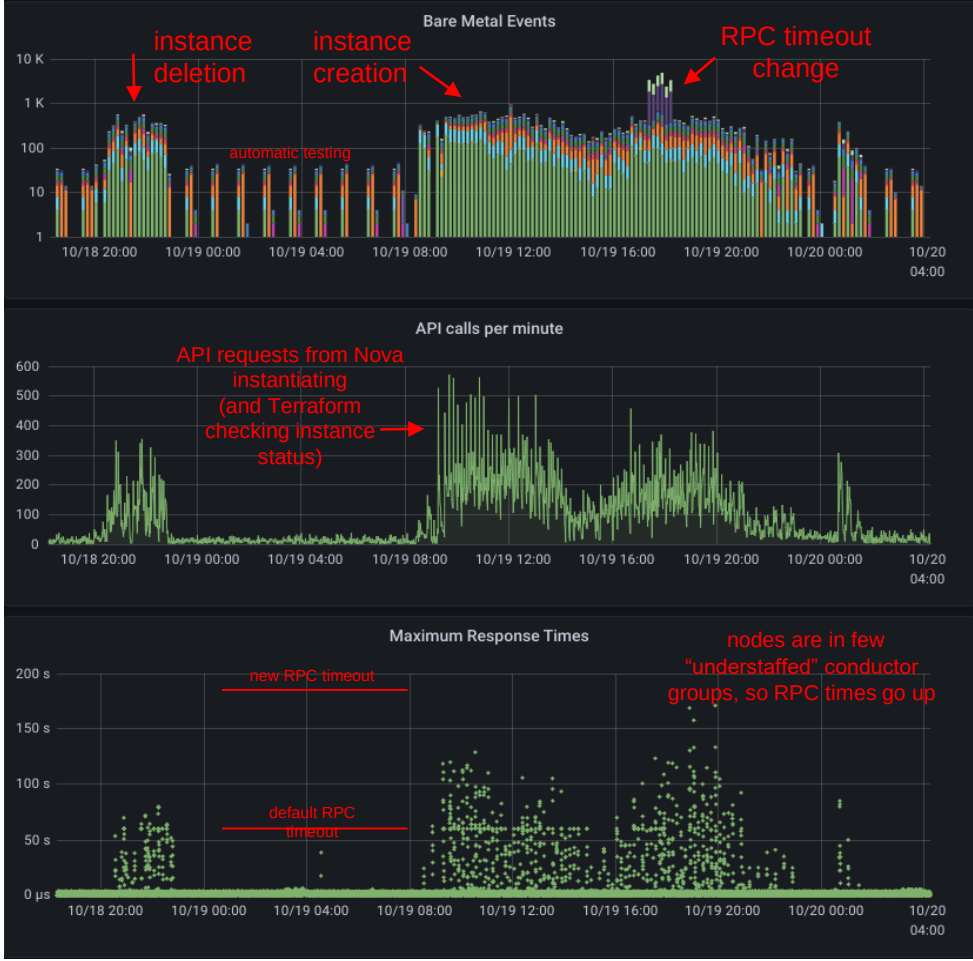
- To Learn more : https://indico.cern.ch/event/1096851/

Physical Batch

- **Conversion of virtual to physical batch**
  - with the availability of a bare metal API, we revisited the virtualisation tax
- **3’775 hypervisors recreated as physical batch instances**
  - done in multiple chunks over several months
- **Terraform as the ‘Infrastructure-as-Code’ tool to interface with OpenStack/Ironic**

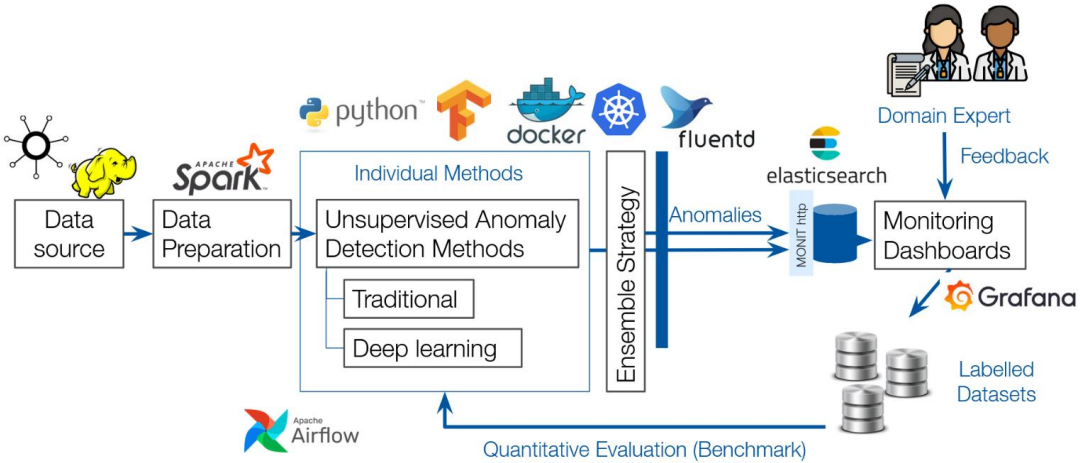
HashiCorp Terraform

Bonus: 16’000 VMs less than one year ago … 10k+ IPv4 addresses free’d up.



Anomaly Detection

- Fully automated Anomaly Detection Pipeline extending the previous work
- Implementation of new Unsupervised Machine Learning algorithms
- Combination of the algorithms in an Ensemble
- Work in progress to include in Daily Operations
- https://indico.cern.ch/event/1123214/contributions/4809938/



Openstack image “glance” service upgrade

What is glance?

- The Image service (glance) project provides a service where users can upload and discover data assets that are meant to be used with other services. This currently includes images and metadata definitions.

Use of glance?

- Allows users to discover, register, and retrieve virtual machine images.
- Allow VMs to be created in batches, reducing deployment time.

Openstack image “glance” service upgrade

GLANCE
an OpenStack Community Project

- Upgraded to Xena version.
- Support for glance quotas
- Why quotas are important
- [50 TB of Cloud Images blog]

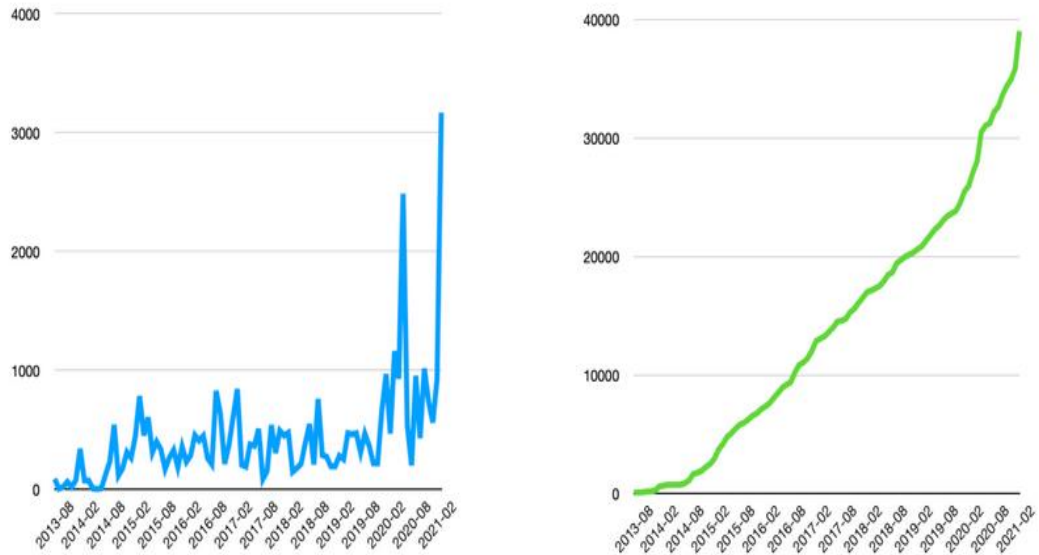


Fig. 1 - Amount of storage used by the available images (per month and cumulative in GB)

Virtual machines migration 101

What is virtual machine migrations?

- In simplest terms migration is the task of moving a virtual machine from one physical hardware environment to another.

Virtual machines migration 101

What is virtual machine migrations?

- In simplest terms migration is the task of moving a virtual machine from one physical hardware environment to another.

Types of virtual machine migrations

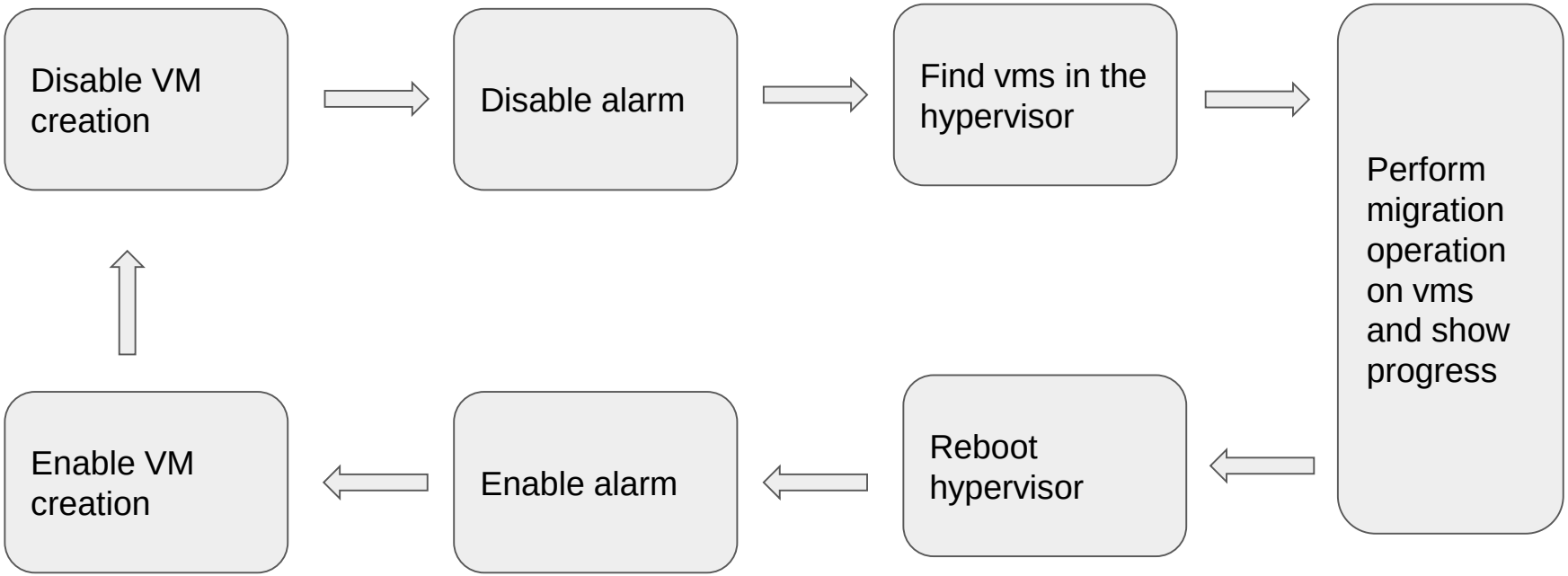
1. **Cold migrations** : cold migration is a virtual machine that is powered off in the entire duration of migration.
2. **Live migrations** : live migration means that the workload and application will remain available during the migration.

Migration cycle tool

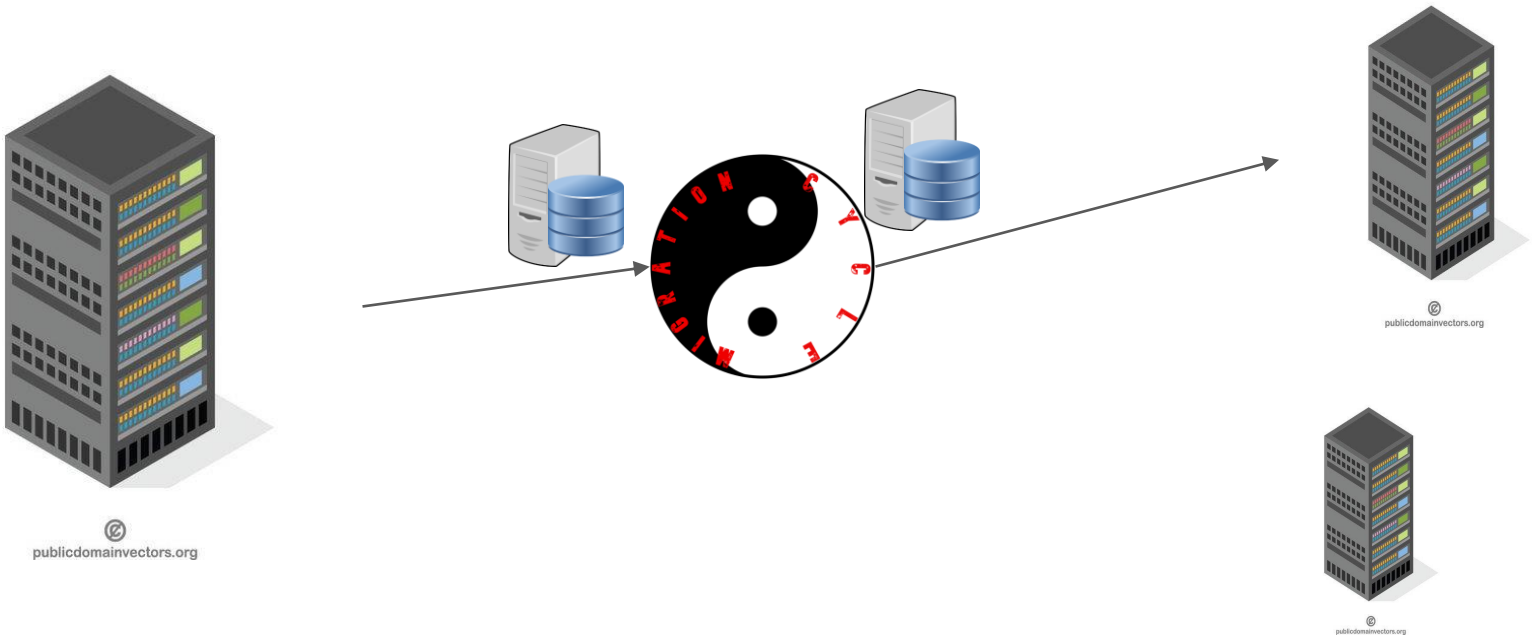
- **Why it was created? What's the use case?**
  - To increase user's VM availability
  - Eliminate/Reduce VM downtime due to hardware interventions
  - Kernel security upgrades
  - Trigger and monitoring live migration execution

- **Where is it deployed?**
  - Automation tool(rundeck) jobs
  - Dedicated node to run and manage vm migrations

How migration cycle works ?



How migration cycle works

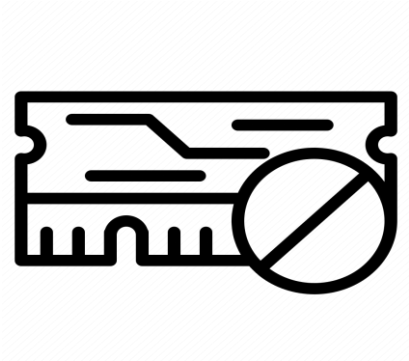


Scenario 1

Eliminate/Reduce VM downtime due to hardware interventions

- Hardware is failing or need repairs/upgrades.

- Example: Memory replacement on a compute node.

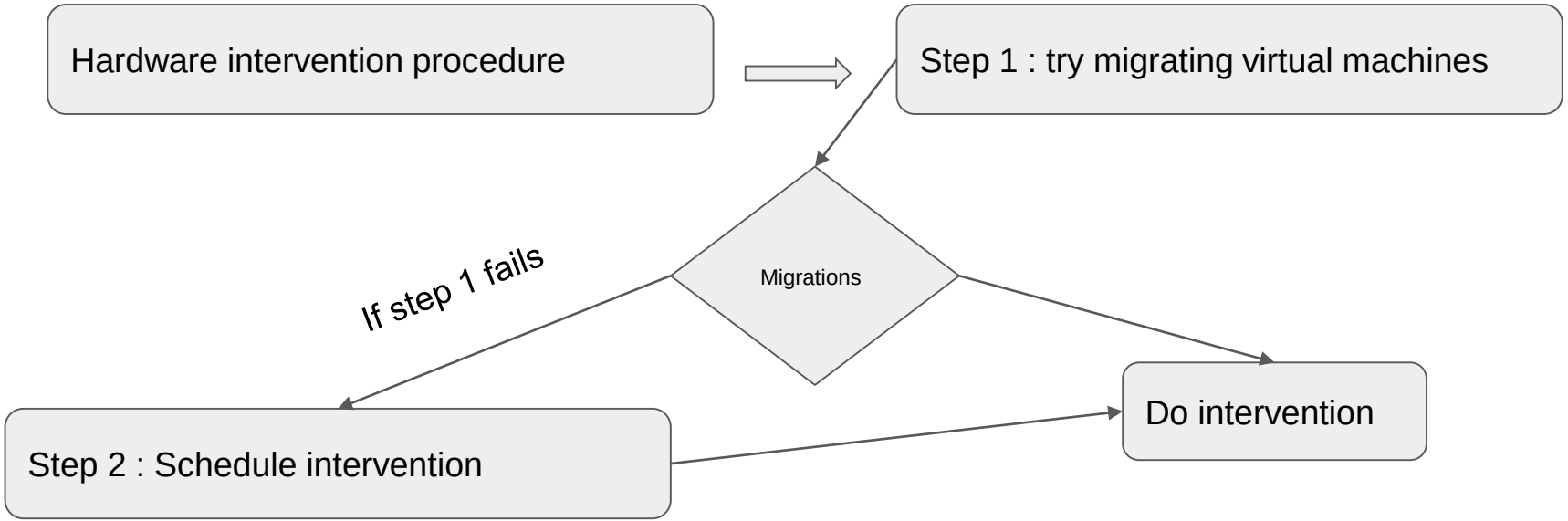


Problem with Previous Solution

- Current procedures causes interruption to users as the maintenance procedure has to be scheduled.
- The hosted virtual machines will be unavailable for the time of the operation.
- The current procedure send email to users stating that “your vm will be down from this time to that time”.
- This procedure disrupts the service and users might not always comply.
- Repair team needs to wait for the schedule date to intervene in the compute node.

New/Improved solution

- Combining previous solution with live and cold migration operations.
- Scheduling server intervention job + migration cycle tool.



How new solution tackles the problem

- New procedures avoid causing downtime by first trying to migrate the instances from the host.

- Essentially the job becomes 2 step process.

- **Step 1** : first the job will perform instances migrations on the hosts that require the intervention. If successful the intervention can be performed immediately. Thus not impacting the user.

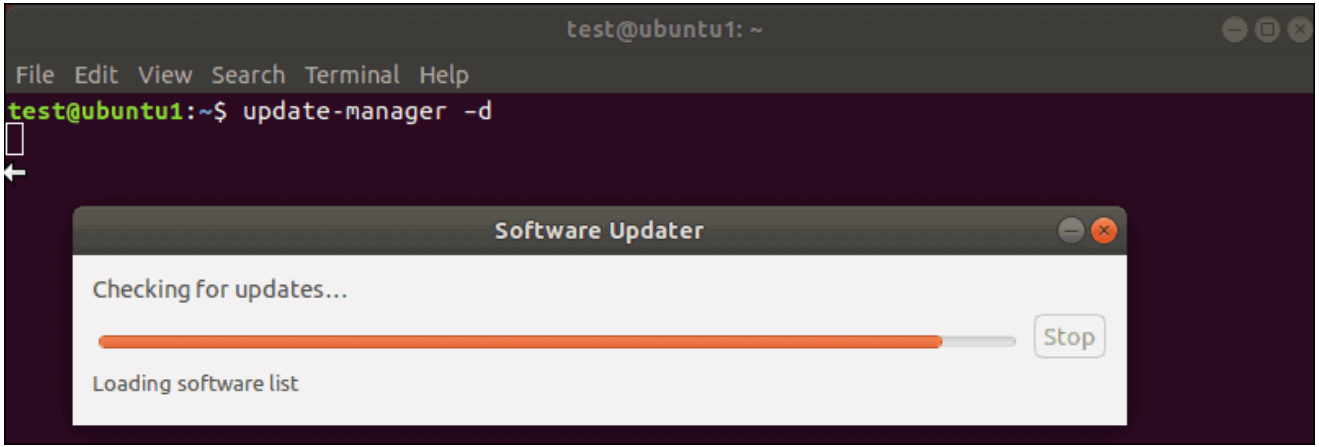
- **Step 2** : if step1 fails the existing procedure will be executed i.e. scheduling intervention.

Scenario 2

Kernel security upgrades

- All compute nodes need to be rebooted for kernel upgrades.

- Example: Security issues, new kernel features...



Scenario 3

Trigger and monitor live migrations ....

- Decommission a hardware/ end of life.

- Need to free up a set of compute nodes.

- https://techblog.web.cern.ch/techblog/post/beyond-live-migrating-virtual-machines/

Migration cycle CLI

```
usage: migration_cycle [-h] --hosts HOSTS [--cloud CLOUD] [--reboot REBOOT]
                       [--compute-enable COMPUTE_ENABLE]
                       [--roger-enable ROGER_ENABLE]
                       [--disable-reason DISABLE_REASON]
                       [--skip-shutdown-vms SKIP_SHUTDOWN_VMS]
                       [--skip-disabled-compute-nodes SKIP_DISABLED_COMPUTE_NODES]
                       [--max-threads MAX_THREADS] [--no-logfile]

Migration cycle interface

optional arguments:
  -h, --help            show this help message and exit
  --hosts HOSTS         select the hosts to empty (default: None)
  --cloud CLOUD         cloud in clouds.yaml for the compute nodes (default:
                        cern)
  --reboot REBOOT       reboot host true/false when host is empty. (default:
                        True)
  --compute-enable COMPUTE_ENABLE
                        enable/disable the compute service after reboot
                        (default: True)
  --roger-enable ROGER_ENABLE
                        enable/disable roger after reboot (default: True)
  --disable-reason DISABLE_REASON
                        disable reason to use in the service (default: None)
  --skip-shutdown-vms SKIP_SHUTDOWN_VMS
                        do not cold migrate instances if they are inshutdown
                        state (default: False)
  --skip-disabled-compute-nodes SKIP_DISABLED_COMPUTE_NODES
                        perform migration on disabled node true/false
                        (default: True)
  --max-threads MAX_THREADS
                        max number of compute nodes to work on at time
                        (default: 1)
  --no-logfile          do not write to log file. just output logs. (default:
```

Future use cases/plans

- Targeting specific virtual machines: currently we can only specify cell or hosts
- Operating system upgrades (major releases)
- Applying security patches at large scale.
- Deploy it on K8s

Conclusion

- In last years we have invested in support for improving service level of virtual machines by investing in virtual machine migrations.
- **Upgraded hardware**
  - Decommissioned 6 year old hardware with better machines
  - They had 128GB of RAM, an Intel Xeon CPU E5-2630 v3 @ 2.40GHz and 2 SSDs of 900GB each that we configured in RAID 1. They are now replaced by 120 compute nodes with 192GB of RAM, an Intel Xeon Silver 4216 CPU @ 2.10GHz and 2 SSDs of 1.8TB each, again configured in RAID 1.
  - Hardware refresh campaign
- **Migration cycle tool**
  - Execution via automated tool (rundeck) : 175+
  - Migrated virtual machines : ~1600
  - OSS release : https://gitlab.cern.ch/cloud-infrastructure/migration_cycle/