Towards a self automated CERN Cloud
Outlines

- Introduction
- CERN Cloud service
- Automation status
- Upcoming challenges
- Improvement plan
- Source code
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
- Currently running Rocky release
CERN Cloud Infrastructure - initial offering

IaaS

Compute
- nova

Storage
- glance

Identity
- keystone

Web UI
- horizon
CERN Cloud Infrastructure - now

**IaaS+**
- Orchestration: heat
- Container Orchestration: magnum
- Automation: mistral
- Web UI: horizon

**IaaS**
- Network: neutron
- Compute: nova, ironic
- Storage: cinder, manila, glance
- Identity: keystone
- Key manager: barbican
Automation in the CERN Cloud

HR

cornerstone

mistral

Resources

grafana

RUN DECK

GNI

collectd
Back in 2012

- LHC Computing and Data requirements where increasing
- Constant team size
- Improve manageability and efficiency
- Automation
  - Considered early on
  - Exercise it as much as possible
Situation now

- 300k core cloud and increasing
  - Addition of new services
  - Continuous improvements on existing ones
- No change in number of staff
- Automation is key
  - Keep service knowledge
  - Offload common tasks
  - Simplify management
Automation in the CERN Cloud @today

- 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

<table>
<thead>
<tr>
<th>Affiliation</th>
<th>User Disabled</th>
<th>User Deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared</td>
<td>Promote</td>
<td>-</td>
</tr>
<tr>
<td>Personal</td>
<td>-</td>
<td>Stop</td>
</tr>
</tbody>
</table>

- Provisioning and cleanup in Mistral workflows
  - Service inter-dependencies
Resource Lifecycle Management in detail

- Set of workbooks interconnected to manage
  - Projects
  - Services

```
project.delete
  → keystone.project.get
  → service.delete
  ← keystone.project.delete

service.delete
  ← mistral
  ← magnum
  → heat
  → barbican
  ← nova
  ← neutron
  ← cinder
  → manila
  ← s3
  ← glance
```
# Resource Lifecycle Management for end user

## 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

<table>
<thead>
<tr>
<th>Volume type description</th>
<th>Number</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard</td>
<td>0</td>
<td>0 GB</td>
</tr>
<tr>
<td>cp1</td>
<td>0</td>
<td>0 GB</td>
</tr>
<tr>
<td>cpio1</td>
<td>0</td>
<td>0 GB</td>
</tr>
<tr>
<td>crypt</td>
<td>0</td>
<td>0 GB</td>
</tr>
<tr>
<td>io1</td>
<td>0</td>
<td>0 GB</td>
</tr>
<tr>
<td>wig-cp1</td>
<td>0</td>
<td>0 GB</td>
</tr>
</tbody>
</table>
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

[Graph showing the expiration of personal instances]

[Graph showing the expiration of personal cores]
Expiration workbook in detail

- Based on project expiration tag and expire_at instance attribute

```
retrieve_projects

check_status
checkExpiration
fixExpiration
processExpiration
reminder
expire
delete
```
Improve VM availability and performance

- Hyperconverged servers
  - Compute + Storage Nodes
  - Local Ceph pool
    - Instances
    - Volumes
  - Ease management
  - Small IO latency
  - Increased Disk capacity
- Use cases:
  - DB and Storage services
Automation in the CERN Cloud @next

- Add new services
- Root Cause Analysis
- Kubernetes Jobs
- Improve further more availability and performance
Continuous addition of new services

- Project management workbooks are prepared to be extended
- Latest addition is the S3 service through RadosGW
- Uses AdminOps API for quota operations
  - python-radosgw-admin
  - python-mistral-radosgw-actions
- Modify workflows accordingly
Root Cause Analysis

- Find root cause of issues
  - Degradation of response of an application
    - CPU issue? kernel degradation?
- Improve alarms with scope
  - Automatically list impacted services
- Find hidden service dependencies
- Trigger automatic resolutions
  - Run healing workflows
Kubernetes jobs

- Moving towards running control plane in Kubernetes
  - Based on Helm charts
  - Healing operations added as jobs
- All automated tasks in Rundeck can be “dockerized”
- Rundeck now interfaces with Kubernetes
- Start moving tasks into jobs
Get even more performance

- 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 Cloud utilization

- Interested in preemptibles:  Preemptible Instances at CERN on Thursday Nov 15th 1:40pm Hall A3
Improve Cloud utilization

- Dynamic allocation of preemptible instances
#talk is cheap
show me the code
Here are the links

- [https://gitlab.cern.ch/cloud-infrastructure/](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
- openstack-in-production.blogspot.ch
- jose.castro.leon@cern.ch
- @josecastroleon
BACKUP SLIDES