Lifecycle Management, Business Continuity and Disaster Recovery Planning for the LHCb Experiment Control System Infrastructure

Pierfrancesco Cifra for the LHCb Online team, CERN

26th International Conference on Computing in High Energy and Nuclear Physics
May 8-12 2023
The LHCb Experiment
The LHCb Experiment Control System (ECS)

• The experiment Control System (ECS) handles the configuration, monitoring and operation of all experimental equipment in all areas of the Online System.

• The Control Framework is based on a SCADA system called WinCC-OA.

• 125 WinCC-OA projects running on Linux Virtual Machines.
The LHCb ECS Virtualization and Storage Infrastructure

**Virtualisation**

- VMs run on a Red Hat Virtualisation (RHV) 4.4 cluster, 8 hypervisors
- 1024 total cores
- 1 TB total memory
- 2x100G Network connectivity per node
- Redundancy, Live Migration, High Availability

**Storage**

- VMs Disks and WinCC-OA project are stored on a remote NetApp, full flash SSD
- VM Disk: NFS mounted on hypervisor, low I/O
- WinCC-OA projects: NFS mounted on the VM, high I/O
- Deduplication, Snapshot, TIVOLI Backup
- 100 TB usable disk space
The LHCb ECS VMs Lifecycle Management

- VM deployment, OS provisioning, ECS configuration
- Need to be able to quickly deploy VM with standard configuration for control system
- Automatic procedure to easily and quickly deploy the VMs
The LHCb ECS VMs Lifecycle Management: Deployment

- Our setup for the deployment:
  - Red Hat Virtualization (Ovirt) and Foreman for VM deployment and OS provisioning
    - Foreman DNS and DHCP registration using a custom foreman Smart Proxy plugin (No official integration between KEA DHCP and Foreman)
    - DHCP used only for OS provisioning with network boot
  - Puppet for the configuration of the VM
The LHCb ECS VMs Lifecycle Management: Configuration

- Puppet code in a git repository
- ECS VMs are organised per sub detector (mapped to Foreman Hostgroup)
- Hierarchical structure:
  - General LHCb machine configuration (Authentication, logs...)
  - ECS VM configuration (WinCC-OA, packages, NFS mounts...)
  - Per sub-detector specific configuration (WinCC-OA Projects, permissions...)
The LHCb ECS VMs Lifecycle Management: WinCC OA Project

• Each project is managed by a Systemd Service
  • Can be automatically started at boot time minimising downtime
  • Backup of the project before is started (better to avoid “live backup” because of database corruption)

• WinCC-OA licenses in the puppet git repository
  • Licenses requested only once and linked to the MAC address of the VM
The LHCb ECS VMs Lifecycle Management: Monitoring

- Node Exporter + Prometheus + Grafana + ELK

- Mainly monitoring: Memory, CPU, project status, VM status

- Custom exporter for monitoring the status of the WinCC-OA projects
The LHCb ECS Disaster Recovery

- What if we have a major failure (real disaster, human error...) on the virtualisation cluster?
- Goal: have a procedure to recreate the cluster from scratch
  - All the VMs and their WinCC-OA projects need to be up and running
- Improve general reliability of the system

ASSUMPTIONS

- This only take into account the recreation of VMS
- We are assuming that the virtualization cluster has been already recovered and a foreman instance is running
- Basic services (Network, Storage, DNS, DHCP...) are fully operational
The LHCb ECS Disaster Recovery: what do we need?

• External Database to keep track of VMs resources (CPU, Memory...)

• MAC address of the VM for the linked WinCC-OA license
  • We don’t want to request again a new license for the VM

• Script that interact with foreman and Red Hat Virtualization API to deploy the VMs in the right order and with the correct resources:

  1. Deploy the VM in the Virtualization cluster with the defined MAC

  2. Register the VM in foreman

  3. Start VM, OS provisioning and configuration
The LHCb ECS Disaster Recovery: Testing the procedure

• We had the possibility to test the procedure:
  • Bring down the cluster and make a backup of the VMs on a different storage infrastructure
  • Start the automatic procedure to re-install all the ECS VMs

• Results:
  • 3 hours to have back the cluster with the projects running from scratch
  • Good opportunity to spot modifications made by hand and not added to the puppet configuration
  • We intend to repeat this test on a regular basis
Conclusion

• Foreman + Puppet works very well (open source, well integrated and documented, large community, customisation)

• Deployment very easy and smooth

• Good to have a disaster recovery procedure

• 125 production VMs have been commissioned to support the LHCb Data Acquisition for RUN 3
THANK YOU FOR YOUR ATTENTION
BACKUP
<table>
<thead>
<tr>
<th>Hypervisor Specification</th>
<th>NetApp Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server</strong></td>
<td><strong>Head Model</strong></td>
</tr>
<tr>
<td>Gigabyte MZ92-FS0-00</td>
<td>NS224NSM100</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td><strong>Software</strong></td>
</tr>
<tr>
<td>2 x AMD EPYC 7502 32-Core Processor</td>
<td>NetApp ONTAP 9.11</td>
</tr>
<tr>
<td><strong>OS</strong></td>
<td><strong>Shelf</strong></td>
</tr>
<tr>
<td>RHEL 8.6, Linux 4.18</td>
<td>2 x AFF-A400</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td><strong>Disks</strong></td>
</tr>
<tr>
<td>Mellanox MT28800 ConnectX-5 Ex</td>
<td>28 x 3.90 TB SSD</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td></td>
</tr>
<tr>
<td>16 x 64 GB DDR4 3200</td>
<td></td>
</tr>
</tbody>
</table>
ControlVM LDAP Down (0 active)
ControlVM Down (0 active)
ControlVM CPU HIGH (0 active)
ControlVM High MEMORY (0 active)
ControlVM High system load (0 active)
ControlVM network error in (0 active)
ControlVM network drop out (0 active)
ControlVM DISK USAGE HIGH (0 active)

WinCC project Systemd unit in ERROR (2 active)
name: WinCC project Systemd unit in ERROR
expr: wincc_project_status = 2
for: 15s
labels:
  notification_class: control_vm
  resolve: false
  severity: critical
  annotations:
    summary: On {{ $labels.host }} the WinCC Systemd unit for project {{ $labels.project }} is in ERROR

Labels

<table>
<thead>
<tr>
<th>State</th>
<th>Active Since</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRING</td>
<td>2023-02-09T12:33:44.78504415Z</td>
<td>2</td>
</tr>
<tr>
<td>FIRING</td>
<td>2023-02-13T06:01:44.78504415Z</td>
<td>2</td>
</tr>
<tr>
<td>host</td>
<td>project</td>
<td>Status</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>utsurface01</td>
<td>UTSURFACETEST</td>
<td>ERROR</td>
</tr>
<tr>
<td>vadaoprbs01</td>
<td>VADAQPRBS</td>
<td>ERROR</td>
</tr>
<tr>
<td>mudaqtest01</td>
<td>MUDAQFEE</td>
<td>STOPPED</td>
</tr>
<tr>
<td>cadcs01</td>
<td>CADCSLV</td>
<td>RUNNING</td>
</tr>
<tr>
<td>cadcs01</td>
<td>CADCSMV</td>
<td>RUNNING</td>
</tr>
<tr>
<td>caecs01</td>
<td>CAECS</td>
<td>RUNNING</td>
</tr>
<tr>
<td>catfcaio101</td>
<td>CASOL40</td>
<td>RUNNING</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>host</th>
<th>project</th>
</tr>
</thead>
<tbody>
<tr>
<td>onldirac01</td>
<td>ONLDIRAC</td>
</tr>
<tr>
<td>pldcs01</td>
<td>PLDCSMV</td>
</tr>
</tbody>
</table>