Notifications workflows using the CERN IT central messaging infrastructure

ZHECHKA TOTEVA (CERN/IT), DARKO LUKIC (UNIV. OF NOVI SAD), LIONEL CONS (CERN/IT)

CHEP2018 – 10/07/2018
Motivation for the CERNMegabus project
- Roger and different scenarios to propagate Roger state change
- Need for improvements
- CERNMegabus architecture and project goals

CERNMegabus use cases and project evolution
- EOS and CASTOR
- DNS Load Balancing
- CERN Computer Centre power cut management

Future plans
Background

Roger

43 000 Puppet managed machines

DNS Load Balancing

CASTOR

EOS

ActiveMQ

FOREMAN

HAProxy

puppet

CASTOR

RabbitMQ

43 000 Puppet managed machines
-bash-4.2$ ai-dump `hostname`

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>aiadm72.cern.ch</td>
</tr>
<tr>
<td>Hardware</td>
<td>virtual, 2 cores, 3.37 GiB memory, swap, 2 disks</td>
</tr>
<tr>
<td>Hostgroup</td>
<td>aiadm/nodes/login</td>
</tr>
<tr>
<td>Comment</td>
<td>aiadm machines are for administrating nodes in the computer centre</td>
</tr>
<tr>
<td>Environment</td>
<td>qa</td>
</tr>
<tr>
<td>Responsible</td>
<td><a href="mailto:ai-config-team@cern.ch">ai-config-team@cern.ch</a></td>
</tr>
<tr>
<td>FE Responsible</td>
<td>Configuration Management</td>
</tr>
<tr>
<td>OS</td>
<td>CentOS 7.5.1804 x86_64 (3.10.0-862.2.3.el7.x86_64)</td>
</tr>
<tr>
<td>VM Project</td>
<td>IT Configuration Management Services</td>
</tr>
<tr>
<td>VM Flavour</td>
<td>m2.medium</td>
</tr>
<tr>
<td>Avail zone</td>
<td>cern-geneva-c</td>
</tr>
<tr>
<td>LANDBsets</td>
<td>it_cc_loads_with_ssh</td>
</tr>
<tr>
<td>LB aliases</td>
<td>aiadm7.cern.ch, aiadm7-testing.cern.ch, aiadm.cern.ch, ...</td>
</tr>
<tr>
<td>CNAME aliases</td>
<td>-</td>
</tr>
<tr>
<td>IPv4</td>
<td>137.xxx.xxx.xxx (ITS) (SS13-C-VM32)</td>
</tr>
<tr>
<td>IPv6</td>
<td>2001:1xxx:xxx::xxx (ITS) (SS13-C-VM32)</td>
</tr>
<tr>
<td>App state</td>
<td>production</td>
</tr>
<tr>
<td>Alarm mask</td>
<td>Hardware(N) OS(N) App(N) NoContact(N)</td>
</tr>
<tr>
<td>Last report</td>
<td>4 minutes ago</td>
</tr>
</tbody>
</table>

- Is a tool that stores and manages the overall application state of a machine
- Example states: production, disabled, intervention, draining
Roger state and Puppet

Machine owner

Change roger state

Roger

Have to wait for a Puppet run? 😞

Puppet agent

cached roger state

Service X

1. Puppet run
Roger state and manual synchronisation

2. Manual synchronisation

Manually? 😊

Machine owner

Change roger state

Roger

Puppet agent

cached roger state

Service X
Roger state and querying Roger server

Change roger state

Machine owner

3. Query regularly

So frequently for something that changes so rare? 😃
Change roger state

4. Push over locally running RabbitMQ message broker

Propagate synchroniously with a roger state change 😊😊😊

Security ❓ Support ❓ Scalability 😞
Need for improvements

- **Speed up** the propagation of a *roger state change* on the machine
- **Without** unnecessarily **querying** the roger server
- **Without** extra lines of **private code**
- By using **supported, large-scale proven, messaging infrastructure** with a flexible **authentication and authorisation schema**
  - **Central IT ActiveMQ message brokers**
Solution: CERNMegabus architecture

- Unify the code in python libraries for the publishers
- Provided python libraries for the subscribers as well
- Offer Puppet configuration both for publishers and subscribers
CERNMegabus use cases and project evolution

- Read/write vs ReadOnly mode of CASTOR tapes and EOS disks
- Presence in a DNS Load Balancing (LB) alias
- CERN Computer centre (CC) power cut management
- Alarms handled by the CERN IT monitoring infrastructure
Start small: EOS and CASTOR use cases

**Replaced** local RabbitMQ with central IT
ActiveMQ message brokers

- Organise **ActiveMQ topic**
- **Local user name**-based authentication schema
- Changed the python library – from PIKA to STOMP.PY
- Use Puppet to configure both consumer and publisher

EOS and CASTOR using `::cemmegabus::client::consumer` (Dec 2017)

```python
::cemmegabus::client::consumer {'eos':
    server  => 'agileinf-mb.cern.ch',
    port    => 61233,
    user    => 'rogercon',
    pass_key => 'roger_key',
    destination => 'roger.notification.hostgroup',
    destination_type => 'topic',
    top_hostgroup  => 'eos',
    hostgroup_selector => "eos/$(instance_without_eos)/storage",
    host_selector   => '',
    owner           => 'root',
}
```
DNS LB: Before CERNMegabus

Change roger state

Roger

Query regularly (~5 mins)

Alias Members

SNMP get

LBD Master

Load metric

Fallback

cached roger state

CERNMegabus - CHEP2018 10-Jul-18
DNS LB: After CERNMegabus

- Change roger state
- Publish roger state change
- ActiveMQ broker(s)
  - Receive message & Run roger_actions
  - Query directly
- cached roger state
  - Publish roger state change
- Roger
  - Change roger state
-Alias Members
  - Query directly
  - SNMP get
- LBD Master
  - Load metric
More challenges: DNS LB use case

- **Orchestration issue**: more listeners than in the EOS/CASTOR use cases (2000 vs 20)
  - “Publish to one and Listens to all” vs “Publish to all and Listens to one”
  - Offer both orchestration models to publishers and consumers

- Use `stompclt` to configure the consumer
  - listen to ActiveMQ message broker for a roger state change
  - update the **cached roger state**

- Use **Puppet** to configure the `stompclt` configuration file

- Use the existing `roger_action` script when a message is received
  - A positive side effect: trigger **the custom defined roger actions**

- Use **certificates** for authentication
CERN CC Power cut management

Challenges

- **20 minutes** on UPS
- Specific format of the **UPS monitoring data**

Requirements

- Formalise complex algorithms for decision making if **there is a power cut**
- **Propagate** the power cut event to all machines in the CC
- Handle the event depending on a **predefined recipe**

![Diagram of UPS monitoring data and decision making algorithm]
During mid-annual power cut test on the 2\textsuperscript{nd} of July, 2018

- Detected power cut
- Notified the subscribed machines
- Shutdown the machines, which had been predefined to be shutdown
- Detected the power back
- Notified the machines, which had been predefined to wait
Future plans

- Install CERNMegabus client on all machines in the CERN CC
- Release CERN CC Power Cut management in production
- Use DNS LB client with roger state criterion
- Configure stompclt configuration files with CERNMegabus Stompclt module
- Assist colleagues to define and realise their CERNMegabus use cases
Thank you!

Questions?
See you at the poster sessions at 16:30

Securing and sharing Elasticsearch resources with ReadonlyREST
Concurrent Adaptative Load Balancing at (@CERN)
THANKS our customers: CASTOR, EOS, DNS LB,
THANKS our collaborators: CERN/IT-CF, CERN/EN-EL
THANKS all my colleagues from IT-CM
New Puppet resource ::cernmegabus::plugins::roger
- CASTOR and EOS did not need hundreds of lines of private code
- On-boarded the Puppet master HAproxy configuration with CERNMegabus

New predefined plugin to update the cached roger state
- we satisfy the needs for alarms handling by the monitoring infrastructure
- The later will be included in base
- Provide easy way to re-write a stompcp configuration file with Puppet

::cernmegabus::plugins::roger { 'protect tapes':
  on_change_param_name => 'appstate',
  on_change_command =>
    'modifydiskserver -s $(echo ${NEW_APPSTATE} | sed -e "s/quiesce/readonly/g") ${HOSTNAME}',
}

::cernmegabus::plugins::roger{'disable-aips-via-roger':
  on_change_param_name => 'appstate',
  on_change_param_from => 'production',
  filters =>
  {'hostgroup' =>
    "punch/puppet/ps/v4/%/::{hostgroup_3}"},
  on_change_command => '/usr/bin/haproxyctl disable all ${HOSTNAME}',
}
CERNMegabus Puppet module – computer center power cut management

- Already implemented the predefined Puppet standard client action

  ```
  include ::cernmegabus::plugins::ccpco
  ```

- Decided to be “send an email” (and/not sink to disk) during the test phase

- Possibility to use a predefined action

  ```
  class{ 'cernmegabus::plugins::ccpco':
    standard_action  => 'shutdown',
  }
  ```

- Or even custom action (both on power loss and on power restore events)

  ```
  class{ 'cernmegabus::plugins::ccpco':
    custom_action => '/bin/backupdata.sh',
    power_back_action => '/bin/restoredata.sh',
  }
  ```
Roger notification and the CC power cut management use messages with **public content**

- BUT the **publishers must be verified**
- Discussing if the CC power cut management needs also to exchange **signed messages** for extra validation
- Unknown need of the future use cases