Post-mortem

OTG0149207 - PuppetDB overload causing Puppet runs to fail

OTG0149214 - Puppet and PuppetDB servers are down

ASDF 2024-04-18

G. Tenaglia, S. Traylen (Config Team), L. Fernández Álvarez (Cloud Team), B. Aparicio Cotarelo (Hadoop Team), A. Cabezas Alonso (DBOD Team)

Executive summary

"PuppetDB, one of the databases used for configuring computing services, was serving incorrect data on Tuesday, March 19th 2024 between 11:55 and 12:30, due to a human error. The outage caused the unavailability of some IT services including the Single Sign-On, the OpenStack API and the NXCALS and Analytix clusters. The root cause of the problem was an incorrect attempt to solve an overload problem that was occurring since March 14th, 2024 16:30. As a workaround PuppetDB was stopped, preventing it from returning wrong results to other services, and restored from backups, which solved the problem. The initial overload problem was permanently fixed on March 20th, 2024 16:00."

Link to C5 post-mortem

What is PuppetDB

"PuppetDB collects data generated by Puppet®. It enables advanced Puppet features like exported resources, and can be the foundation for other applications that use Puppet's data."

What PuppetDB is used for

Puppet functions `query_facts`, `query_nodes` and wrappers (`cernlib` etc)

```
  function cernlib::hostgroups2ips (
    Variant[String[1],Array[String[1],1]] $hostgroups,
[...]
      $_query_components = $_hostgroups.map | $_hg | {"hostgroup=\"${_hg}\""}
[...]
    $_query_result = query_facts($_query_components.join(' or '),['networking']
```

```
    # Allow the load balancers to reach 8140 (PS listening port)
    $_hap_ips = cernlib::hostgroups2ips('punch/puppet/hap',true)
    nftables::set{ 'hap4':
      type     => 'ipv4_addr',
      elements => $_hap_ips['ipv4'],
    }
    nftables::set{ 'hap6':
      type     => 'ipv6_addr',
      elements => $_hap_ips['ipv6'],
    }
```

What PuppetDB is used for

`ai-pdb(1)` official tool

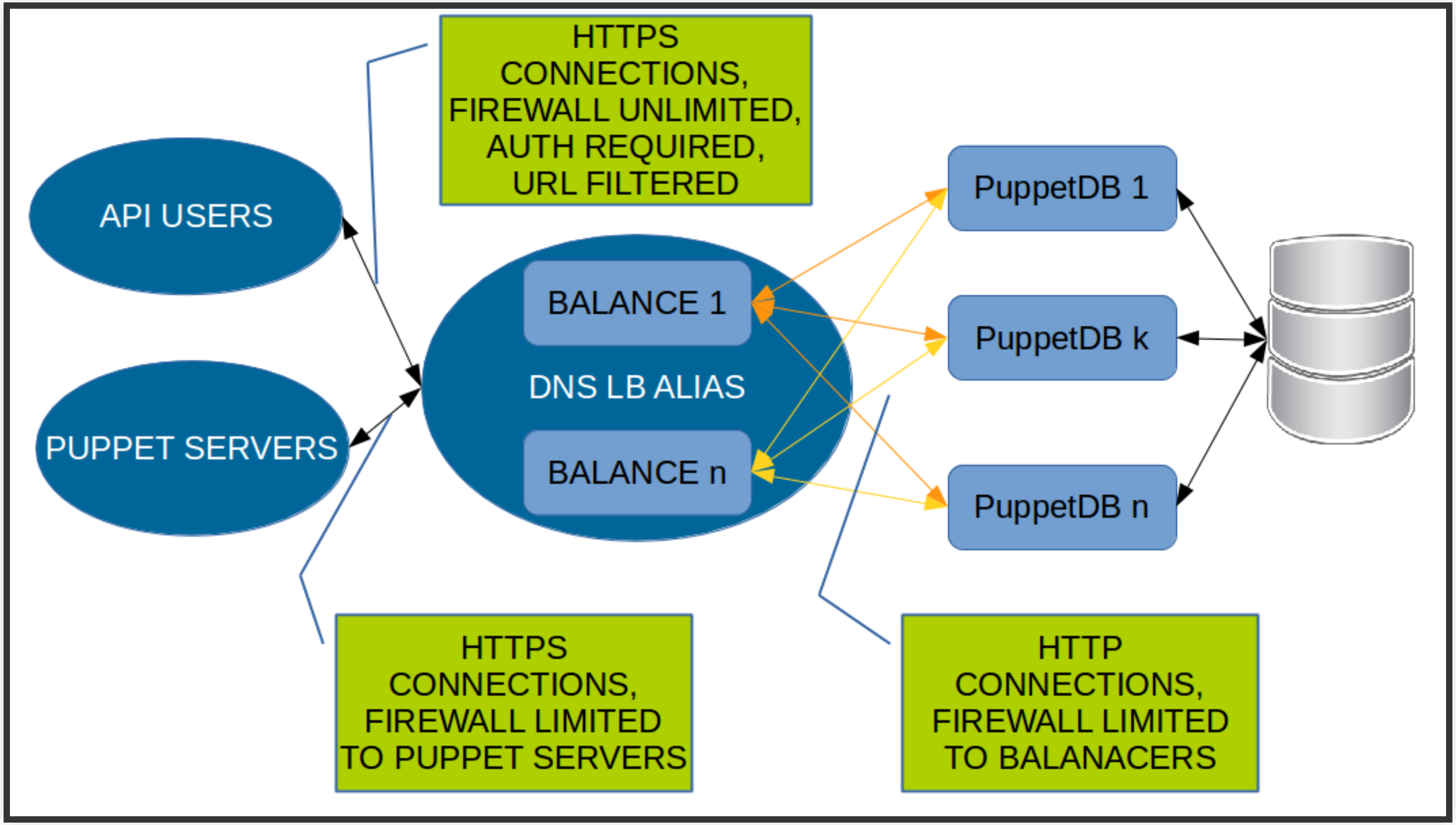
```
$ ai-pdb hostgroup_fact --subgroups 'punch/pdb' os | jq -r '.[] | "\(.value.rel
9.3 aipdbbal91.cern.ch
9.3 aipdbbal90.cern.ch
9.3 aipdb93.cern.ch
9.3 aipdb91.cern.ch
9.3 aipdb90.cern.ch
9.3 aipdb83.cern.ch
9.3 aipdb81.cern.ch
9.3 aipdb80.cern.ch
9.3 aipdb66.cern.ch
9.3 aipdb00.cern.ch
9.3 aipbdbal93.cern.ch
```

What PuppetDB is used for

Direct queries to the HTTP API

```
sub PDBGetAllDevices ($@) {
    my ($set, @hostgroupsofset) = @_;
    my @devices = ();
    my $pdb_address = $config->get('puppetdb_address');
    my $pdb_port = $config->get('puppetdb_port');
    my $curl = "/usr/bin/curl";
[...]
    my $cmd = "$curl -s --negotiate -u :
        -H 'Accept: application/json' -k
        https://$pdb_address:$pdb_port/pdb/query/v4/resources/Cernfw::Landbset/
    my $resp = `$cmd`;
[...]
    my $json = JSON->new();
    my $dec = eval { $json->decode($resp); };
[...]
    for my $entry (@{$dec}) {
        next unless exists $entry->{'certname'};
        my ($basememberhostgroup, @rest) = split(/\//, $entry->{'parameters'}->
        if (grep {$_ eq $basememberhostgroup} @hostgroupsofset) {
```

PuppetDB Architecture


HTTPS CONNECTIONS, FIREWALL UNLIMITED, AUTH REQUIRED, URL FILTERED
API USERS
PUPPET SERVERS
BALANCE 1
DNS LB ALIAS
BALANCE n
PuppetDB 1
PuppetDB k
PuppetDB n
HTTPS CONNECTIONS, FIREWALL LIMITED TO PUPPET SERVERS
HTTP CONNECTIONS, FIREWALL LIMITED TO BALANACERS

Impact

- OTG0149221 - ProxySQL keycloak instance not reachable due to firewall update
    - OTG0149216 - CERN Single Sign-On partially unavailable
        - OTG0149217 - LanDB Web applications not accessible without SSO session
- OTG0149218 - lxplus alias not updating
- OTG0149215 - [Cloud Infrastructure] OpenStack APIs unavailable
- OTG0149222 - [IT-DA] Analytix and NXCALS Hadoop clusters degraded & OTG0149224 - Connectivity issues to NXCALS and Analytix clusters from SWAN

Root cause

- Large facts slowing down PuppetDB: mistake done by service managers.
- Deletion of 15k nodes from PuppetDB: mistake in the garbage collector configuration (configured a 1m purge timeout *and* a 1m expire timeout rather than just the purge one).

PuppetDB GC configuration

```
# How often (in minutes) to compact the database
# gc-interval = 60
gc-interval = 0
[...]
node-ttl = 30d
node-purge-ttl = 10d
```

- Outside working hours: switch the above comment.
- `node-ttl`: expires nodes not reporting for 30d.
- `node-purge-ttl`: deletes nodes expired for 10d (= not reporting for 40d)

The trigger

```
# This doesn't work as anything 0 (0d, 0m, ...) means "disabled"
node-ttl = 0d
node-purge-ttl = 0d

# This works and deletes all nodes not having run Puppet for the last 2 minutes
node-ttl = 1m
node-purge-ttl = 1m
```

→ Resulted in 15k nodes removed from PuppetDB.

What went well

- Service managers seem to have nice procedures in place to put in place workarounds/fix issues.
- DBOD backups were available and restore was fast and effective.
- Great expertise in the team (ZT/ST) to help the newcomer.
- Great on-boarding exercise for the newcomer.

What went wrong

- We misunderstood the garbage collector configuration parameters.
- We underestimate the reliance on *live and user-populated data* on PuppetDB by critical services.
- Council Week ™️
- We understand DB recovery was complicated as it required a Puppet run 🐔/🥚.
- The deletion of 15k nodes from PuppetDB did not solve the overload problem.
- We run an ancient version of PuppetDB so the reaction from the Puppet community (chat) was “upgrade to a supported version”.
- It was lunch time so few service managers around.

Where we got lucky

- It was lunch time so few end users around.
- The outage was worked around in 20-35 minutes effectively meaning two thirds to half of the servers on average did not experience wrong data being served but just failed Puppet runs / PuppetDB queries.

Timeline

Initial problem and fix development

- 2024-03-14 16:30 - **START PuppetDB Overload**
  - Large fact mistakenly injected on 3 nodes.
  - Unable to replace them due to SQL stmt too large.
- 2024-03-15 09:54 - Issue understood
  - Problematic configuration rolled back.
  - Fix development (fact block-listing), no success.

Timeline

Second fix and second problem

- 2024-03-19 11:30 - Solution identified
  - Expire the problematic hosts, run the PDB GC.
  - “Less risky and error-prone option.”
- 2024-03-19 11:55 - **START PuppetDB serving wrong/empty data**
  - See before, expired a total of 15516 nodes.
- 2024-03-19 12:10 - **START Puppet runs failing everywhere**
  - All Puppet Servers stopped.
- 2024-03-19 12:30 - **END PuppetDB serving wrong/empty data** - **START failed PuppetDB queries**
  - All PuppetDB Servers stopped

Timeline

Second fix and second problem (cont^d)

- 2024-03-19 13:49 - PuppetDB primary restored.
- 2024-03-19 14:48 - **END Puppet runs failing everywhere** - **END failed PuppetDB queries**
  - All Puppet and PuppetDB Servers started.

Timeline

The ultimate fix

- 2024-03-20 09:00 - Back to work on the initial problem
    - Development of API-based solution, without success.
    - Development of a manual database clean-up solution.
- 2024-03-20 16:00 - **END PuppetDB Overload**
    - Manual clean-up of the PuppetDB database that fixed the original problem.

Resolution

After triple checks of DB schema and taken a backup

```
DELETE FROM factsets
  WHERE certname=host{1,2,3}.cern.ch`;
DELETE FROM fact_paths
  WHERE name = 'offending_fact'
  AND paths LIKE 'offending_fact%';
```

Follow up action items

- Mitigate the impact of PuppetDB returning empty/wrong results:
    - Always validate results from PuppetDB
    - LBD service - module-lbclient #70
    - Added min number of hosts to `cernlib::hostgroups2ip` module-cernlib #127.
- Operational improvements
    - Possibly check PuppetDB validity from haproxy as to the number of hosts it contains when it checks state of back end node.

Follow up action items (cont^d)

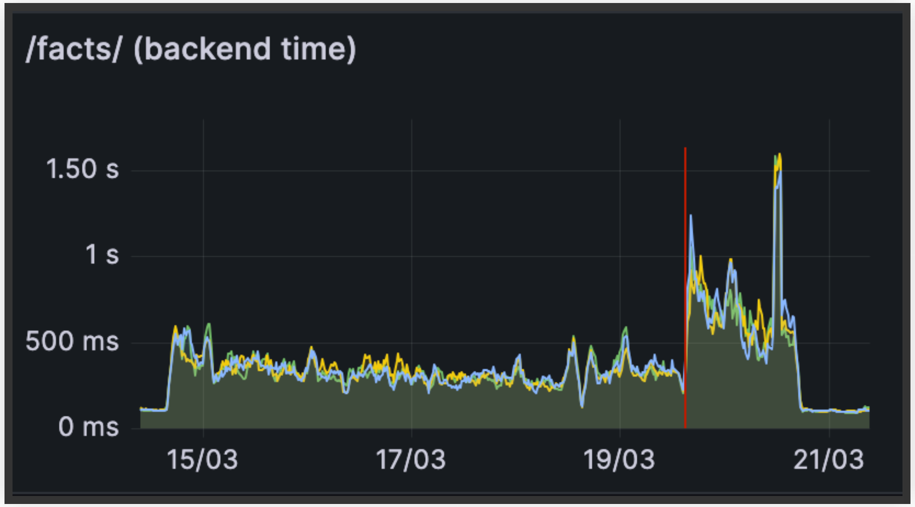
- Large fact injection prevention/detection:
  - Large facts injection supposedly fixed on newer PuppetDB.
  - In the past we could not upgrade due to performance reasons (see PDB-4830). Maybe now puppetlabs/puppetdb#3592 makes it feasible. Tracked at AI-6278.

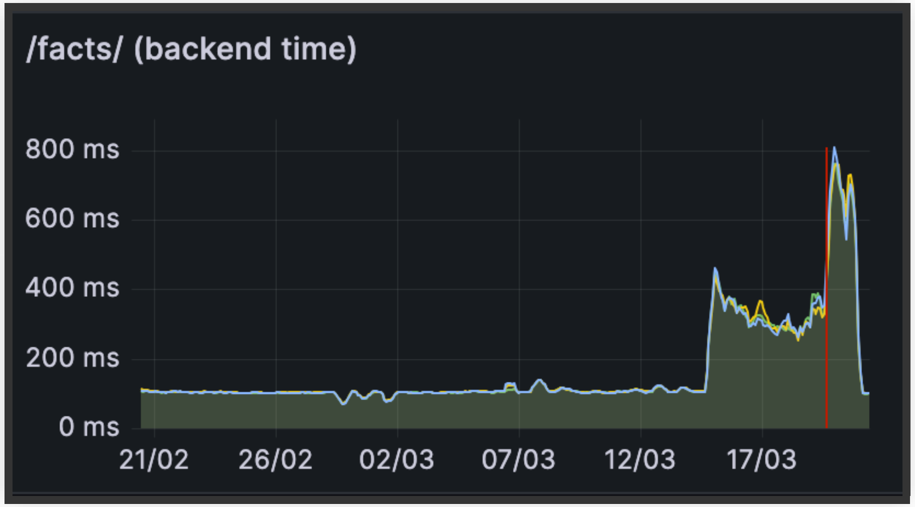
Supporting information

- PuppetDB Grafana dashboard
- Internal Jira with useful details: AI-6277
- Communication:
  - ~Puppet MM channel
  - ~DownForEveryoneOrJustMe MM channel

Supporting information

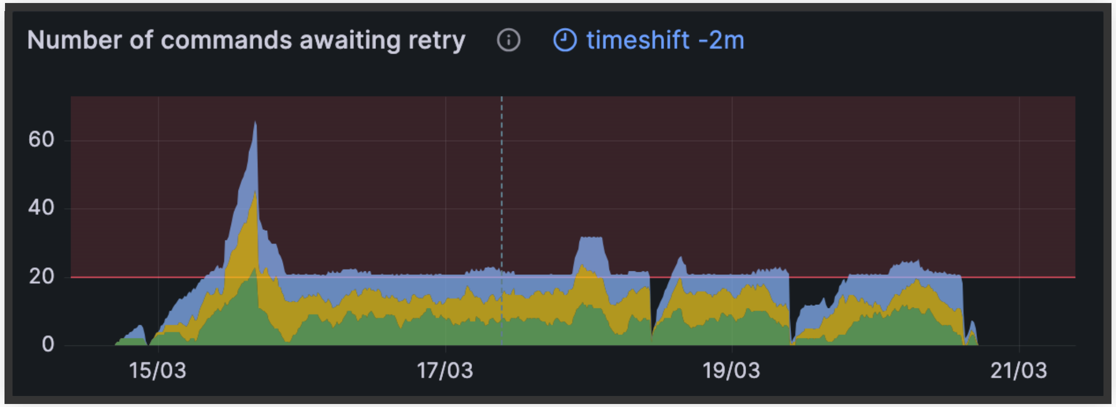
`/facts/` endpoint latency

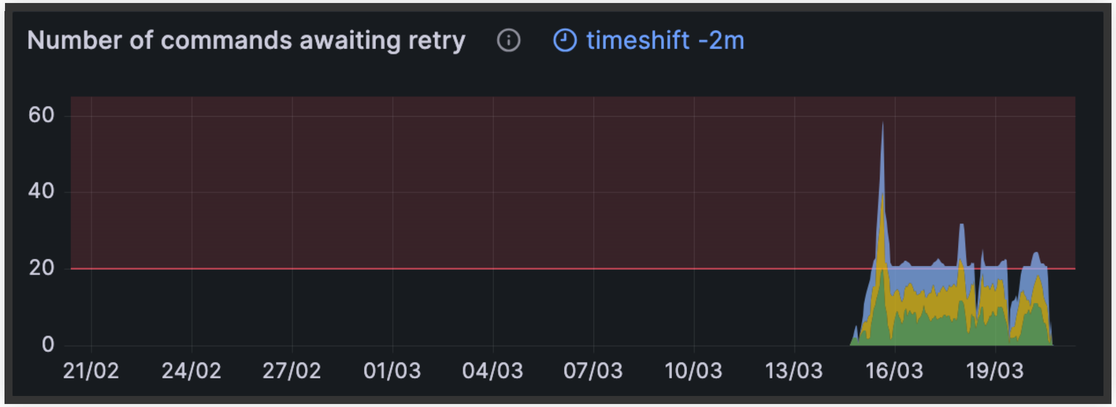




Supporting information

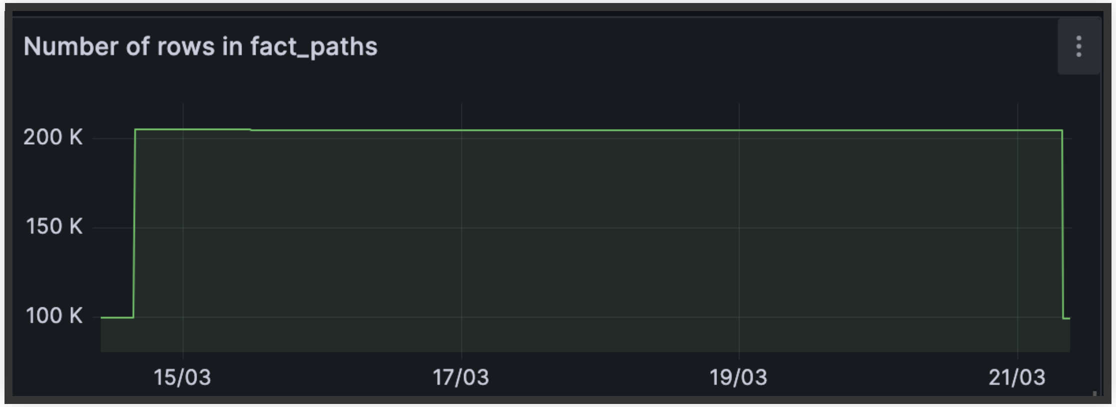
Commands in the queue

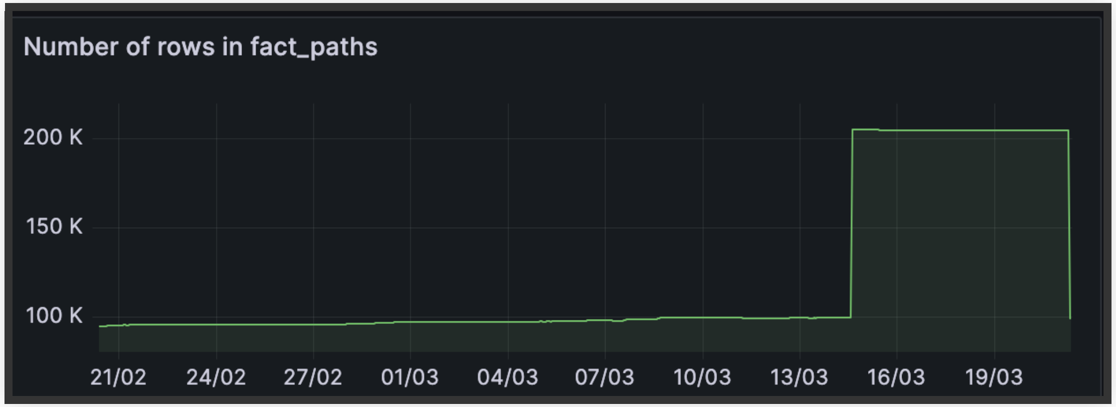




Supporting information

`fact_paths` database table size





Summary of impacted services

Cloud Infrastructure (I)

Access to OpenStack APIs degraded for **1 hour**. Running machines not affected.

Postmortem: OS-17798

Triggered by: firewall rules lost, proxies missing nodes, incomplete list of nodes in config,…

- 12:04 - **OUTAGE BEGINS**: Errors start in APIs.
- 12:06 - Problem is detected.
- 12:15 - Puppet disabled and manual fix start.
- 13:06 - **OUTAGE MITIGATED**: service is operational.
- 14:53 - Puppet is re-enabled.

Cloud Infrastructure (II)

- **Where we got lucky**
  - Puppet was stopped fast.
  - Spotted quickly, it helped reducing degradation period.
- **What went well/wrong**
  - Existing monitoring helped pointing to right places.
  - …but manual recovery is sub-optimal (expected).
- **Follow-up actions**
  - Apply Config team mitigation recommendations.
  - Extend functional probing alarms with proxy alarms.

Hadoop Service (I)

HBase cluster for NXCALS (Next Generation CERN Accerator Logging system), rejecting new connections

Postmortem: ITHADOOP-1944

Triggered by: empty ipsets used for firewall rules

- 14h22: **OUTAGE BEGINS** First alarms reporting delays in job processing
- 15h19: NXCALS team reports issues getting new connections to HBase
- 15h20: Identified root cause: lack of connectivity between master HBase nodes and one region server
- 16h20: **OUTAGE MITIGATED** Concluded that config is correct. Services in affected nodes restarted to re-initialize connections broken

Hadoop Service (II)

- **Where we got lucky**
  - Only access to some data partitions in HBase NXCALS cluster was affected
  - (Partitions hosted in the node where Puppet run while PuppetDB returned no data)
- **What went well/wrong**
  - Monitoring of delays in job processing worked correctly
  - HBase services had to be restarted to restablish connections to master nodes
- **Follow-up actions**
  - Apply Config team mitigation (check for empty data returned by PuppetDB queries)
  - Review delay seen on collectd alarms

DBOD Service

ProxySQL keycloak instance not reachable due to firewall update

OTG0149221

Impact:

- New keycloak DB client sessions failing to resolve the DB proxy alias (existing DB sessions where unaffected)
- OTG0149216- CERN Single Sign-On partially unavailable.
    - OTG0149217 - LanDB Web applications not accessible without SSO session

DBOD MySQL HA architecture:

- 14 Nomad nodes configured with the same lbalias
- One database proxy (ProxySQL) per MySQL DB cluster that can run on any node of the Nomad cluster
- Only one node (the one where the proxy runs) serves the DB clients requests

Root cause: it-puppet-module-lbclient

- Lbclient (alias member) needs to allow snmp traffic from lbd4 and ldb6 servers
- List of ldb4 and ldb6 were retrieved through a `query_facts` function:

```
#Let's allow with ipset
    $_lbdnw = query_facts('hostgroup_0=ailbd',['networking'])
    $_lbdips4 = sort(delete_undef_values($_lbdnw.map |$_host,$_nw| { $_nw['netw
    # Avoid occassional scope6=link (link-local) IPv6 addresses in ipset
    $_lbdips6 = sort(delete_undef_values($_lbdnw.map |$_host,$_nw| { if $_nw['n
```

- Two ipsets were created from these lists
- Finally the firewall rules were created to allow snmp traffic only from the ips in the ipset
- Since the ipsets were empty, the lb servers could not query the node due to firewall blocking -> nodes where puppet run were expelled from the alias

Timeline

- 2024-03-19 at ~12:00 puppet runs and returns an empty list for `$_lbdnw` (list of aildb ips)
- 2024-03-19 at 12:04 The node is expelled from the lbalias since the lbservers cannot ping it, snmp firewall rules were updated based on the new (empty) `_lbdips4` and `_lbdips6` ipsets.
- 2024-03-19 at 12:36 the db proxy instance is relocated to a host that is still member of the lbalias
- 2024-03-19 at 12:37 DB clients connections are re-established

Resolution

- Migrated the keycloak ProxySQL instance to a node of the cluster that was still part of the lbalias
- Marked as uneligible the three Nomad nodes affected by this incident

What went wrong

- We were unlucky since only 3 nodes out of the 14 nodes comprising the cluster were affected, the keycloak proxy was running on one of these 3.

What went well

- Only new DB client sessions where affected

Future planned actions

- lbclient puppet module: Fail if the ailbd contains no members done by Steve Traylen ✓
- DBOD: Migrate MySQL HA clusters to MySQL InnoDB cluster in order to avoid a SPOF at the proxy layer