

A hybrid system for monitoring & automated recovery @ Glasgow Tier-2 cluster



Motivation

We have been looking at how an increased level of automation can allow significantly more CPU-power to be delivered with the existing staff-effort.

Most sites already use all sort of automation tools, from system provisioning, remote management, monitoring and issue tracking. This allows WLCG sites to run many instances with relatively little manpower.

For this project we have developed an automated recovery system that can resolve the most common issues, such as a disk clean-up or a system reboot, creating a Tier-2 cluster that is more resilient, more efficient and requires less human intervention.

Outline

- Glasgow Tier2 site description
- Cluster Monitoring & Logging
- Alerts & Automation
- Visualization
- Conclusions & Outlook



ScotGrid Glasgow:

Emanuele Simili, Gareth Roy (†), Gordon Stewart, Samuel Skipsey, David Britton

ScotGrid Glasgow Site

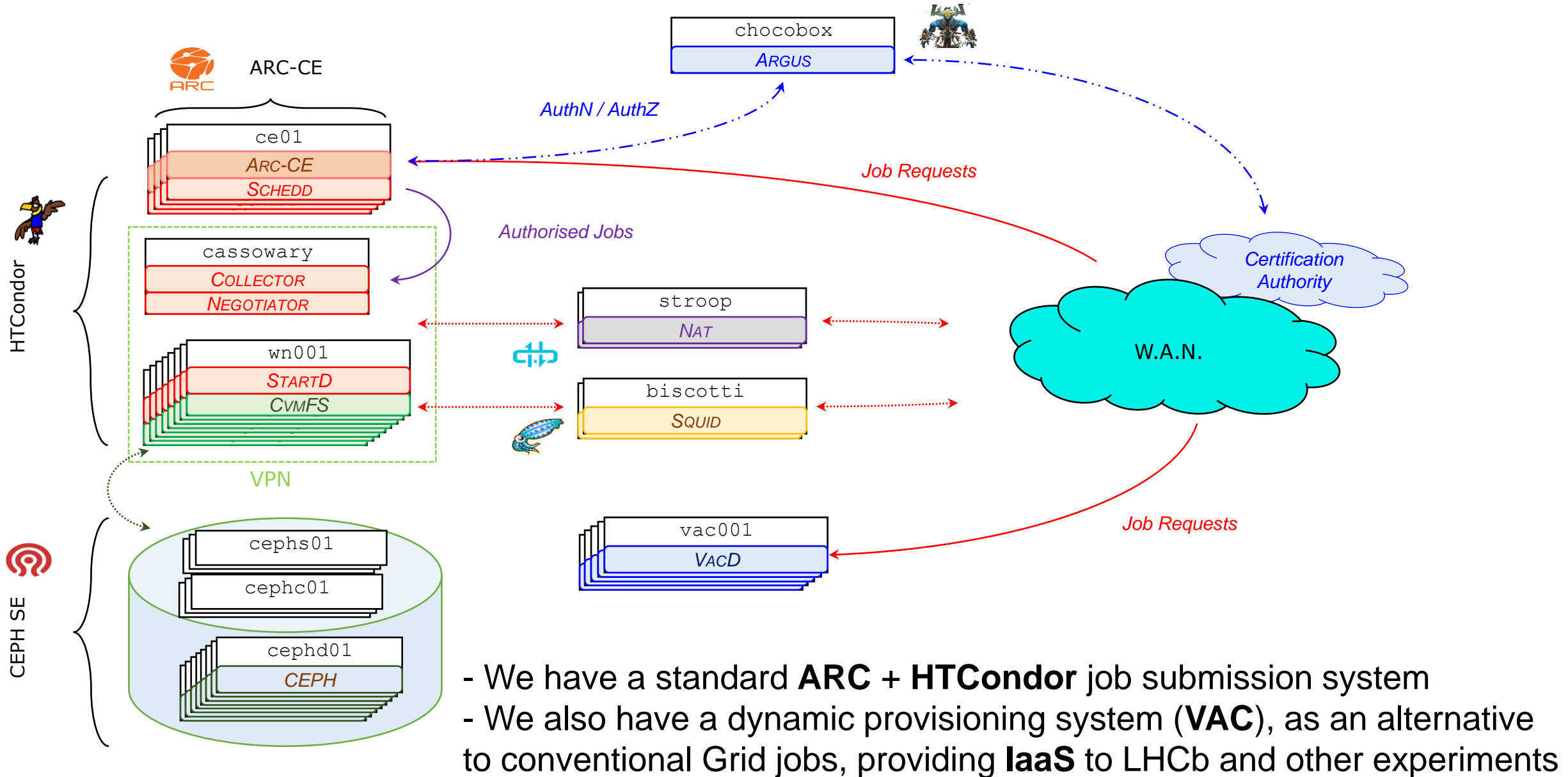


At present, ScotGrid Glasgow consists of:

- ~6000 CPU cores
- 4.9 PB storage (3.2 PB CEPH + 1.7 PB DPM) *
- 160 Gb/s internal network bandwidth

For a rough total of about 61 Khs06

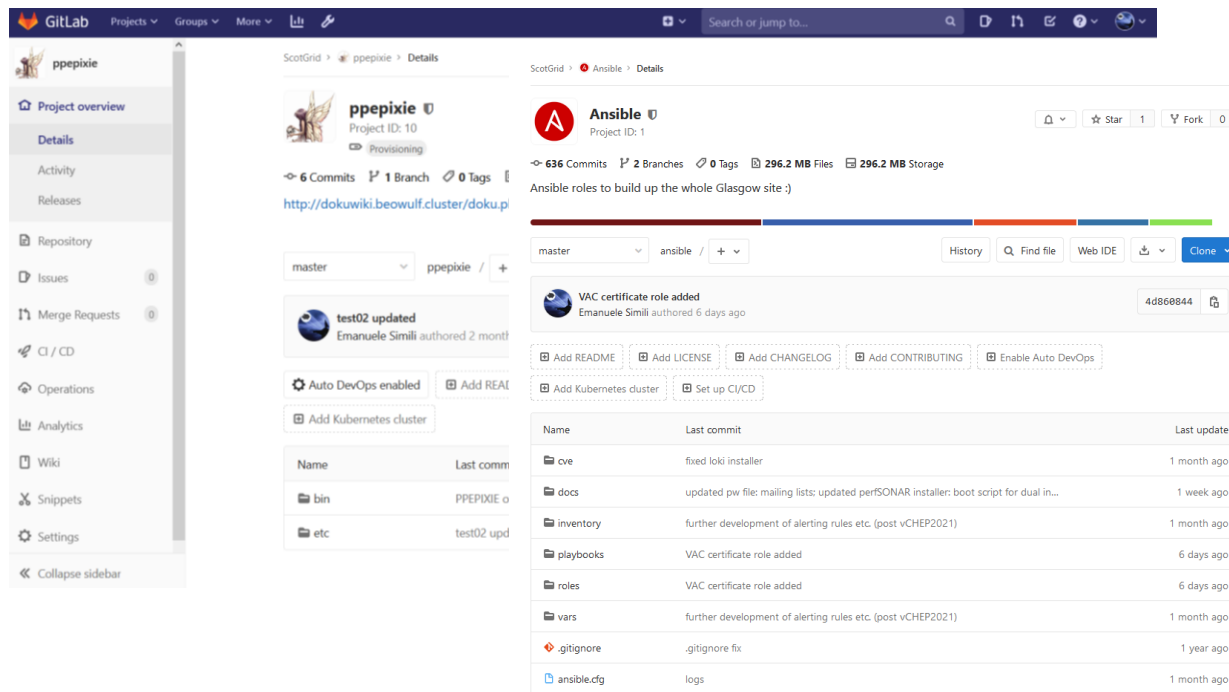
Cluster Map



Automation Tools

As a site of significant size, we use many sys-admin empowering tools:

- A network provisioning system (**PPE PiXiE**)
- A remote management tool (**Ansible**)
- A central monitoring & logging system (**PLG**)



And we have invested quite some time in preparing such environments...

Example Procedures

Create a new workernodes (assuming node **00x** is cabled and plugged in):

- Fill in MAC, IP and HostName (in Ansible), then run the play-book to update the network
nedit roles/dhcp/files/hosts.conf roles/dnsmasq/files/10.0.0.0_8
ansible-playbook update_networking.yml
- Create a PXE system profile, then boot the machine to install OS
nedit systems/nodes/wn00x.yml
systemctl restart pxecli
- Finally run ansible to install the software and configure the workernode
ansible-playbook -I wn00x workernodes.yml

Add a new VO to the standard job submission system:

- Fill in the VO coordinates in a proper Ansible role, then run it
cp -r roles/vo-template roles/vo-xxx
nedit roles/vo-xxx/files/xxx.grid-mapfile
ansible-playbook add_vo.yml -extra-vars "vo=xxx"

Clean CVMFS cache and reboot node **00x**:

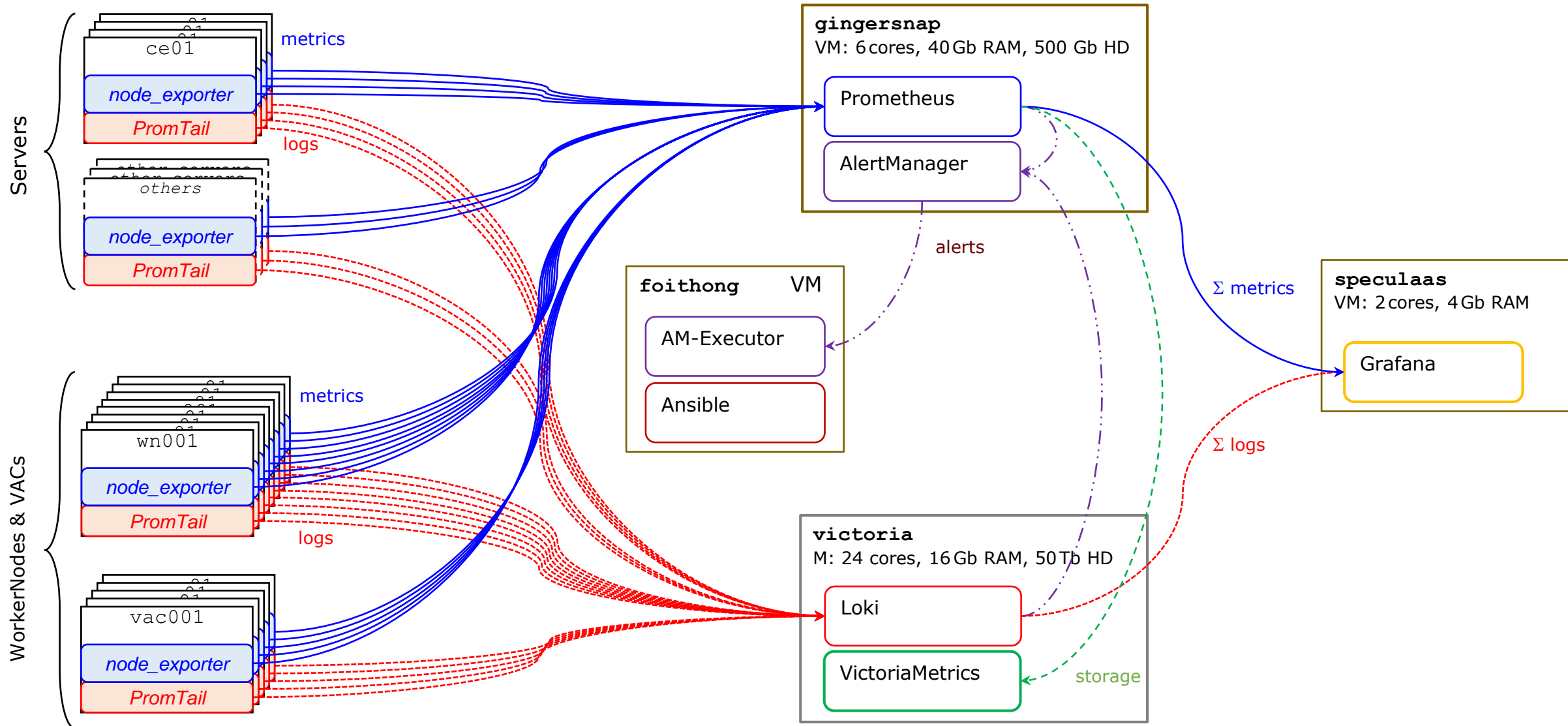
- Just run the ansible play-book
ansible-playbook -I wn00x cleanCVMFS.yml

Monitoring & Logging (1)

Our system is built on **Prometheus**, **Loki** and **Grafana (PLG)**

- Metrics are exported by **node_exporter** (installed on nodes) and collected by a central **Prometheus** instance
- Logs are exported by **PromTail** (nodes) and collected by **Loki** (central)
- Data visualization and querying is done by **Grafana**, which pulls data from both **Prometheus** and **Loki**
- Alerts are sent out by **AlertManager**, installed on the Prometheus server, and received by **AM-Executor**, installed on the configuration management server that also runs **Ansible**
- Long term storage of metrics is done by **VictoriaMetrics**, installed on the same server as **Loki** to take full advantage of the large storage

Monitoring & Logging (2)



Exported Metrics

node_exporter produces by default a large set of metrics (e.g., CPU load, network activity, disk I/O, ...)

And it can be further customised with ad-hoc metrics:

Metric	Purpose	Tool
Temperature HD	check for overheat	S.M.A.R.T. tools
Temperature CPU	check for overheat	IPMI tools
HTCondor jobs stats	check the status and ownership of running jobs	condor_status
VAC jobs stats	check the number and status of virtual machines	check-vacd
Running processes	runtime and resource usage	ps + grep
Number of Reboots	static counter to keep track of unexpected restarts	cron job
Static info	check for unexpected changes in hardware (e.g., memory bank fail)	lscpu, vmstat , lsblk

Exported Logs

PromTail tails the content of local logs and sends it to a central **Loki**.

We have chosen this tool (instead than the widely spread **ElasticStearch**) because:

- it well integrates with Grafana
- it is very easy to set-up ...

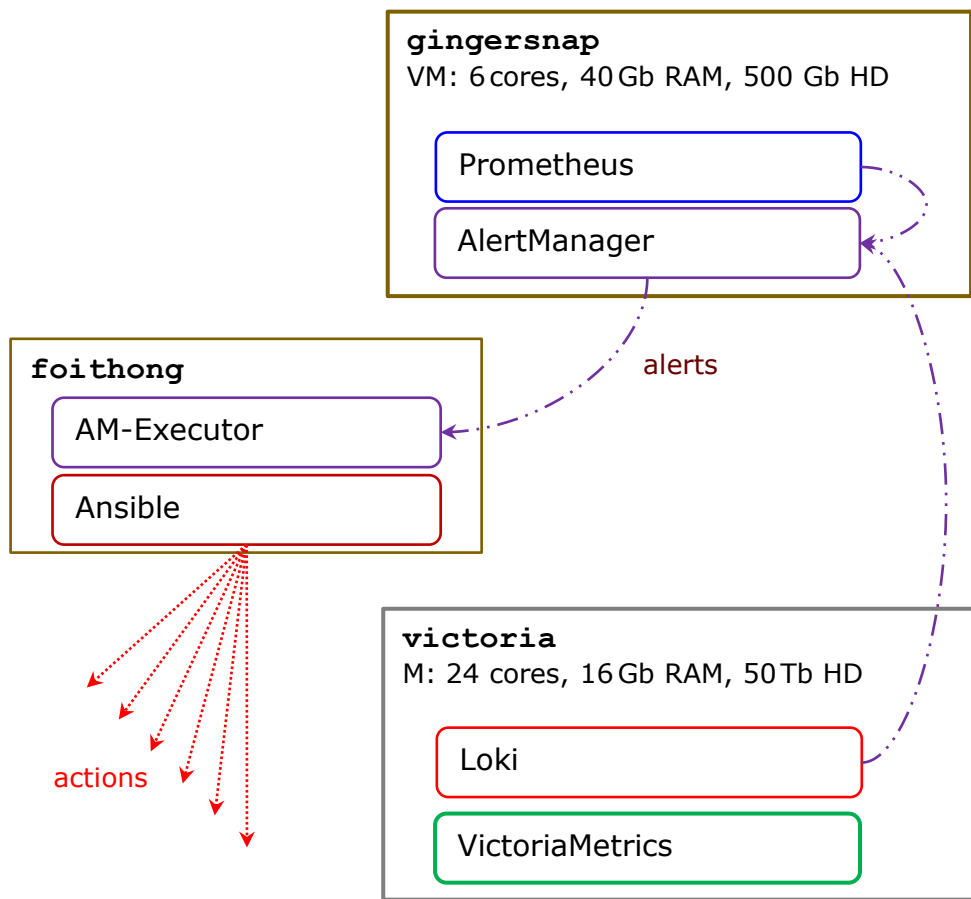
Logs to scrapes are specified in the configuration (we scrape default system logs* and specific service logs)

** we do not export the journal because it clogs the system*

Service	Exported Logs
all	/var/log/messages /var/log/secure /var/log/cron /var/log/yum.log
HTCondor	/var/log/condor/
ARC-CE	/var/log/arc/ /var/log/arc/bdii/
ARGUS	/var/log/argus/pap/ /var/log/argus/pdp/ /var/log/argus/pepd/
Squid	/var/log/squid/
Apache	/var/log/httpd/
Ansible	/var/log/ansible/ /var/log/alerts/

Alerts & Automation (1)

Alerts are generated according to the outcome of conditional tests involving metrics or queries against certain criteria. Both **Prometheus** and **Loki*** can be configured to trigger alerts when a custom set of conditions is met.

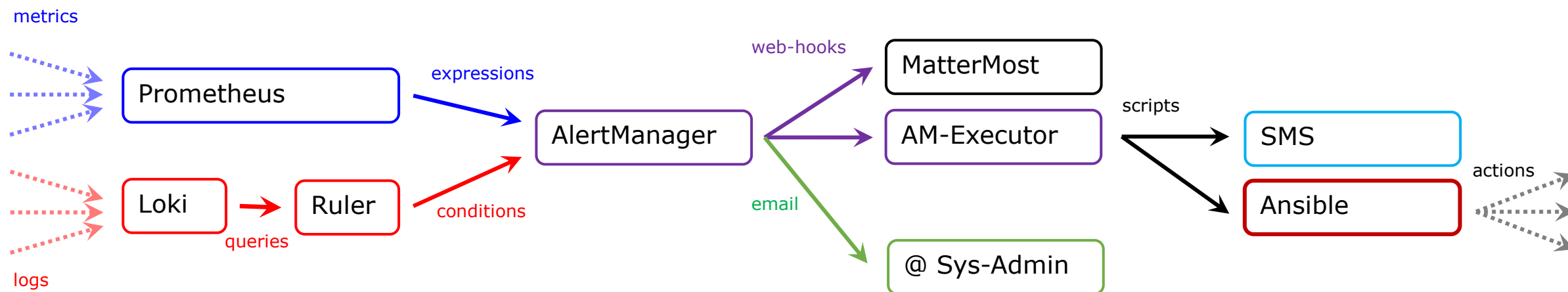


Alerts are handled and sent out by the **AlertManager** ...

... and are received by the **AM-Executor**, which performs some pre-encoded actions.

* Loki's alert system is new in v2.0 and uses a specific plug-in (`ruler`), which in turns calls `AlertManager` ...

Alerts & Automation (2)



In summary, these are the stages of an alert :

- 0) **Prometheus** or **Loki** verifies a test condition and triggers **AlertManager**
- 1) An alert messages is generated by **AlertManager** and sent as a web-hook
- 2) **AM-Executor** is an HTTP server that listens for alerts, parses their content, and triggers custom commands matching the alert type and source
- 3) By running **AM-Executor** on the same management server that runs **Ansible**, we can harness the full power of Ansible remote management, with a variety of play-books prepared beforehand

AM-Executor

AM-Executor is configured with a custom script that defines the action to be taken based on the alert:

- Actions implemented in a case-switch
- Multiple alerts handled in a for loop
- The script can call an Ansible playbook or trigger another script (SMS)
- Privileges for Ansible remote management are set by a password-less SSH key *

* This can raise security concerns, and we'll use a better solution in the future

```
# case-switch to assign an action
case $AM_ALERTNAME in
    InstanceFull) ansible-playbook -l $AM_NODE /etc/ansible/playbooks/do_cvmfsclean.yml -b
        echo "> ansible-playbook -l $AM_NODE do_cvmfsclean.yml -b" >> $AM_LOGFILE
        ;;

    CpuHot)      ansible-playbook -l $AM_NODE /etc/ansible/playbooks/do_shutdown.yml -b
        echo "> ansible-playbook -l $AM_NODE do_shutdown.yml -b" >> $AM_LOGFILE
        ;;

    HdHot)       ansible-playbook -l $AM_NODE /etc/ansible/playbooks/do_shutdown.yml -b
        echo "> ansible-playbook -l $AM_NODE do_shutdown.yml -b" >> $AM_LOGFILE
        ;;

    NodeLazy)    ansible-playbook -l $AM_NODE /etc/ansible/playbooks/do_clean-enable.yml -b
        echo "> ansible-playbook -l $AM_NODE do_clean-enable.yml -b" >> $AM_LOGFILE
        ;;

    VacLazy)     ansible $AM_NODE -m reboot -b
        echo "> ansible -l $AM_NODE -m reboot -b" >> $AM_LOGFILE
        ;;

    WayTooHot)   ansible-playbook /etc/ansible/playbooks/do_shutdown.yml -b
        echo "> ansible-playbook do_shutdown.yml -b" >> $AM_LOGFILE
        ;;

    TooHot)      . /etc/am-executor/send_sms_alert.sh
        echo "> . /etc/am-executor/send_sms_alert.sh" >> $AM_LOGFILE
        ;;

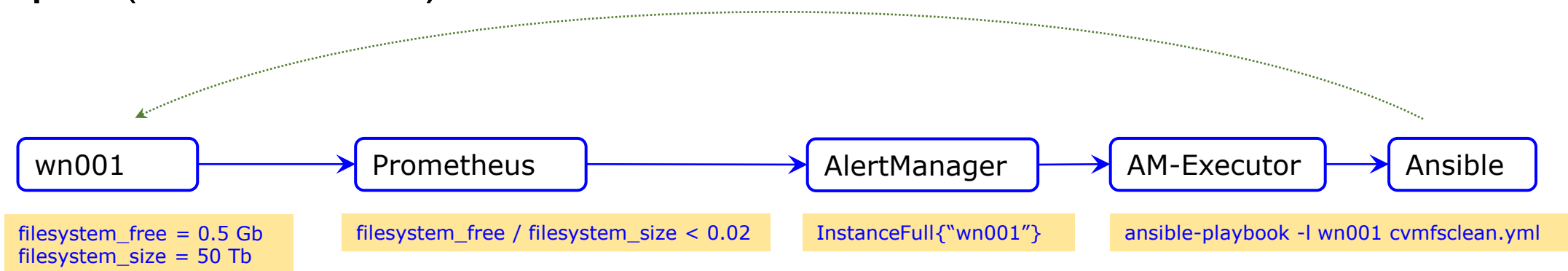
    *) echo "Alert not recognised: $AM_ALERTNAME !" >> $AM_LOGFILE
        ;;
esac
```


Alert Conditions

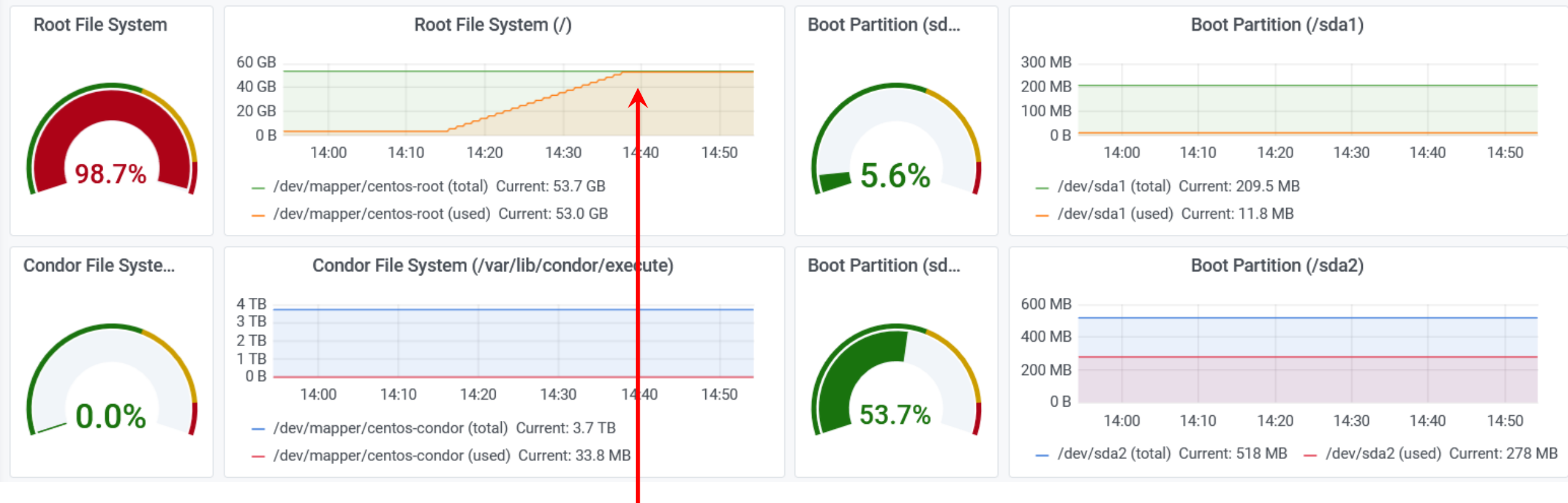
So far, only a limited number of alert-cases have been implemented:

Alert condition	Alert Name	Action	Mail Chat	SMS
$\text{avg}(\text{node_sensors_ambient}\{all\}) > 25$	TooHot	/	✓	✓
$\text{avg}(\text{node_sensors_ambient}\{all\}) \geq 30$	WayTooHot	shutdown{all}	✓	
$\text{node_hwmon_temp}\{node\} > 80$ $\vee \text{node_sensors_hddtemp}\{node\} > 35$	CpuHot HdHot	shutdown{node}	✓	
$\frac{\text{node_filesystem_free}\{node\}}{\text{node_filesystem_size}\{node\}} < 0.02$	InstanceFull	clean{node}	✓	
$\text{avg}(\text{node_condor_cpu}\{node\})[1h] \leq 1$ $\wedge \text{avg}(\text{node_condor_cpu})[1h] > 10$	NodeLazy	reboot{node} + enable{node}	✓	
$\text{rate}(\{node\} \mid \sim "[Ee]rror"[1h]) > 0.1$	ManyErrors	disable{node}	✓	


Example (instance full):



Example Alert



One of the disk has filled-up and an 'Alert' message is generated (after 10 min):

 AlertManager BOT 2:53 PM

[FIRING:1] InstanceFull local nodes (/dev/mapper/centos-root xfs new_dell wn-d20-001:9100 workernodes /)

Alert Pending

Prometheus Alerts Graph Status ▾ Help Classic UI

✓ Inactive (6)

✓ Pending (2)

✓ Firing (1)

Show annotations

/etc/prometheus/alertrules.yml > toohot

> TooHot_DC (0 active)

/etc/prometheus/alertrules.yml > vcchep

> WayTooHot (0 active)

> NoJobs (0 active)

> NoVACjobs (0 active)

> CpuHot (1 active)

> HdHot (0 active)

> InstanceFull (1 active)

> NodeLazy (1 active)

> VacLazy (0 active)

▼ InstanceFull (1 active)

name: InstanceFull

expr: node_filesystem_free_bytes{device=~"/dev/mapper/.*",job="workernodes"} / node_filesystem_size_bytes{device=~"/dev/mapper/.*",job="workernodes"} < 0.02

for: 10m

labels:

severity: local

type: nodes

annotations:

description: {{ \$labels.instance }} has the {{ \$labels.device }} partition full (mountpoint = {{ \$labels.mountpoint }}). Clean it up!

summary: Instance {{ \$labels.instance }} is full

Labels

	State	Active Since	Value
alertname=InstanceFull device=/dev/mapper/centos-root fstype=xfs group=new_dell instance=wn-d20-001:9100 job=workernodes mountpoint=/ severity=local type=nodes	PENDING	2021-05-18T13:38:16.967886192Z	0.013029280654616513

Annotations

description

wn-d20-001:9100 has the /dev/mapper/centos-root partition full (mountpoint = /). Clean it up!

summary

Instance wn-d20-001:9100 is full

Alert Firing

✓ Inactive (6)
✓ Pending (1)
✓ Firing (2)

☒ Show annotations

```
/etc/prometheus/alertrules.yml > toohot
```

➤ TooHot_DC (0 active)

```
/etc/prometheus/alertrules.yml > vcchep
```

➤ **WayTooHot** (0 active)

➤ **NoJobs** (0 active)

➤ **NoVACjobs** (0 active)

➤ **CpuHot** (1 active)

> HdHot (0 active)

➤ **InstanceFull** (1 active)

➤ **NodeLazy** (1 active)

➤ **VacLazy** (0 active)

▼ InstanceFull (1 active)

```
name: InstanceFull
```

```
expr: node_filesystem_free_bytes{device=~"/dev/mapper/.*",job="workernodes"} / node_filesystem_size_bytes{device=~"/dev/mapper/.*",job="workernodes"} < 0.02
```

for: 10m

```
labels:
```

sever

```
type: nodes
```

```

annotations:

```

descripti

```
summary: Instance {{ $labels.instance }} is full
```

Labels

alertname=InstanceFull

severity=local type=nodes

Annotations

description

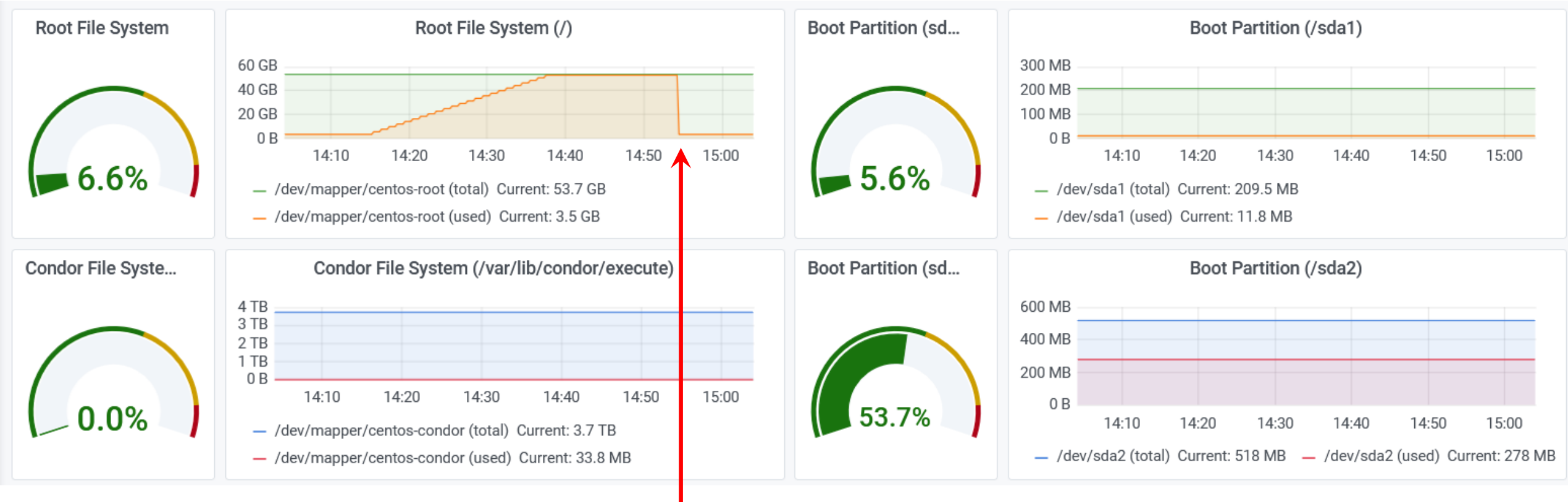
wn-d20-001:9100 has the /dev/mapper/centos-root partition full (mountpoint = /). Clean it up!

summary

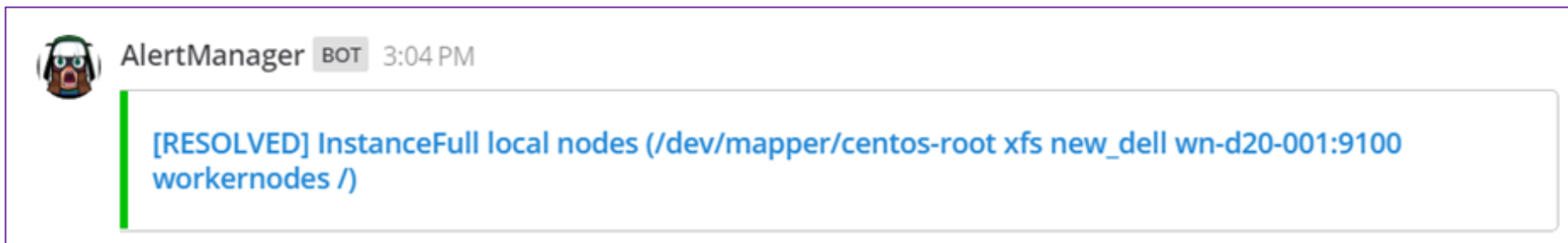
Instance wn-d20-001:9100 is full

Labels	State	Active Since	Value
<code>alertname=InstanceFull</code> <code>device=/dev/mapper/centos-root</code> <code>fstype=xfs</code> <code>group=new_dell</code> <code>instance=wn-d20-001:9100</code> <code>job=workernodes</code> <code>mountpoint=/</code> <code>severity=local</code> <code>type=nodes</code>	FIRING	2021-05-18T13:38:16.967886192Z	0.013029662310698584

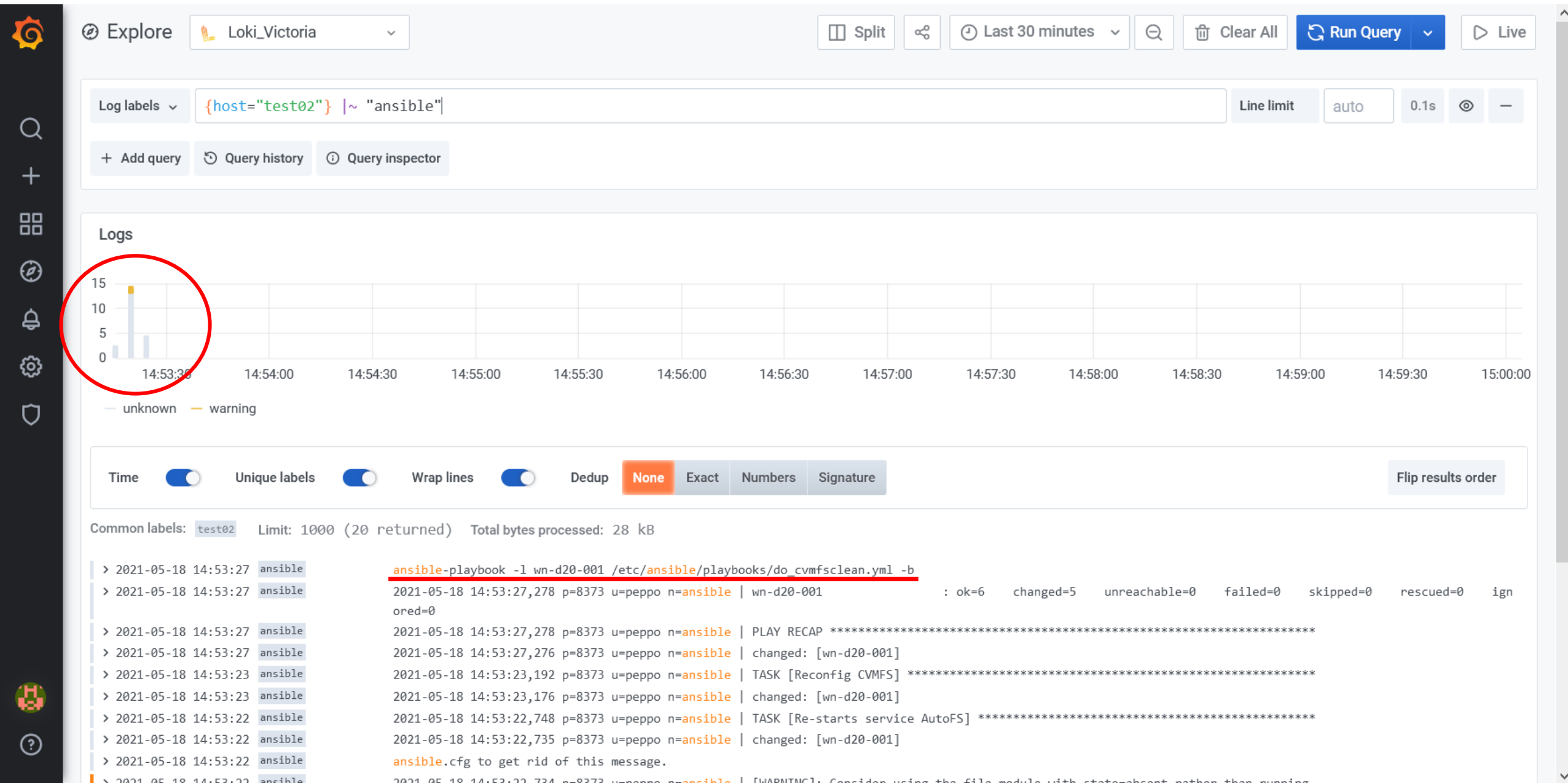
Alert Solution



The disk has been cleaned and a 'Resolved Alert' message is sent (after 10 min):



Ansible Logs



Conclusions & Outlook

This system is relatively new (first working prototype in January 2021) and it is still under critical evaluation*:

- We have an alerting system that can send emails/SMS in case of emergency, and few simple recovery actions that run automatically (clean HD, reboot, ...)
- Next, we want to identify more complex alerts based on combinations of metrics and define multi-step recovery actions, such as re-provision a node
- Also, it would be nice to extend the monitoring actions to network stats
- Eventually, we would like to experiment with ML Anomaly Detection systems
- We plan to engage other GridPP sites to come up with the best practices for monitoring and automated fault recovery, as well as get involved with the CERN Operational Intelligence group

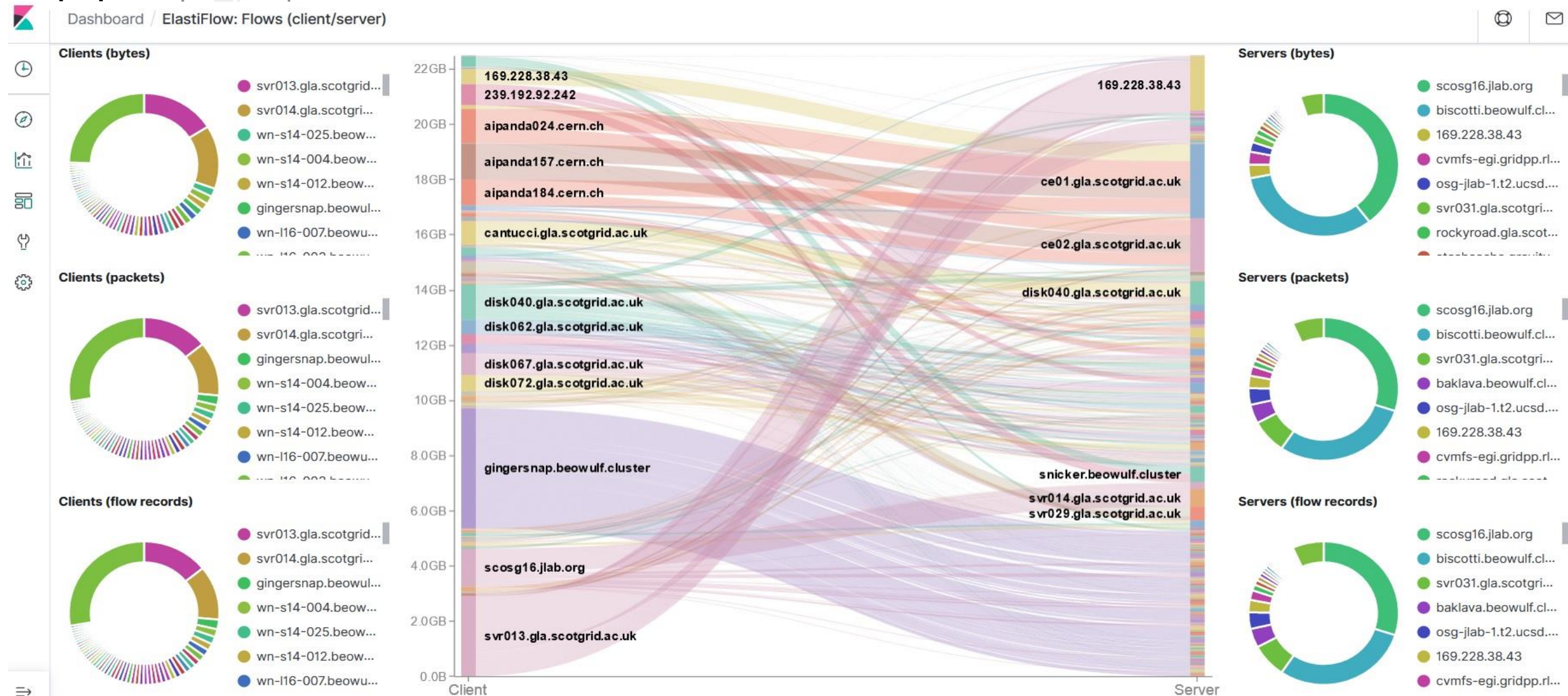
* It is well known that giving full control to the machines without proper evaluation will eventually lead to *Skynet*



Thanks for listening.

ElasticFlow

We did some experimenting with **ElasticSearch** modules to analyze network flows from



... but this is another story.

Example Logging

As an example, these are the bits in the **PromTail** configuration (`promtail.yml`) that export general and HTCondor logs:

```
1  # PromTail Configuration defines the rules for scraping local logs
2
3  server:
4    http_listen_port: 9080
5    grpc_listen_port: 0
6
7  # Positions
8  positions:
9    filename: /tmp/positions.yaml
10
11 # Loki Server URL
12 clients:
13   - url: http://{{ central_log_server }}:3100/loki/api/v1/push
14
15 scrape_configs:
16   ## Common Logs
17   - job_name: messages      # log messages (/var/log/messages)
18     static_configs:
19       - targets:
20         - "{{ inventory_hostname }}:9080"
21         labels:
22           job: messages
23           host: "{{ inventory_hostname }}"
24           __path__: /var/log/messages
25   # ...
```

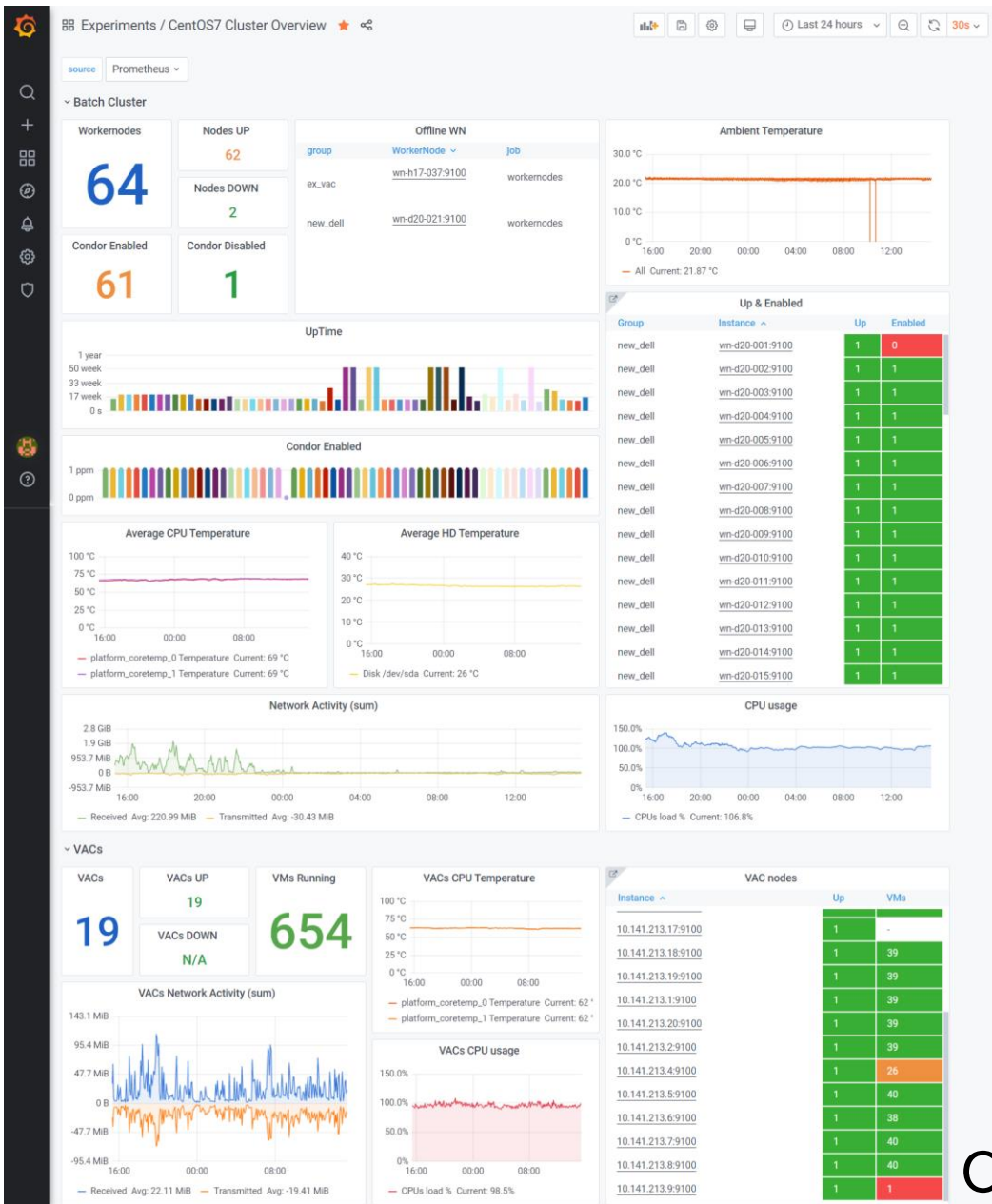
General Logs (system messages)

```
27  ## Service Logs
28  - job_name: condor
29    static_configs:
30      - targets:
31        - "{{ inventory_hostname }}:9080"
32        labels:
33          job: condor
34          host: "{{ inventory_hostname }}"
35          __path__: /var/log/condor/*og
```

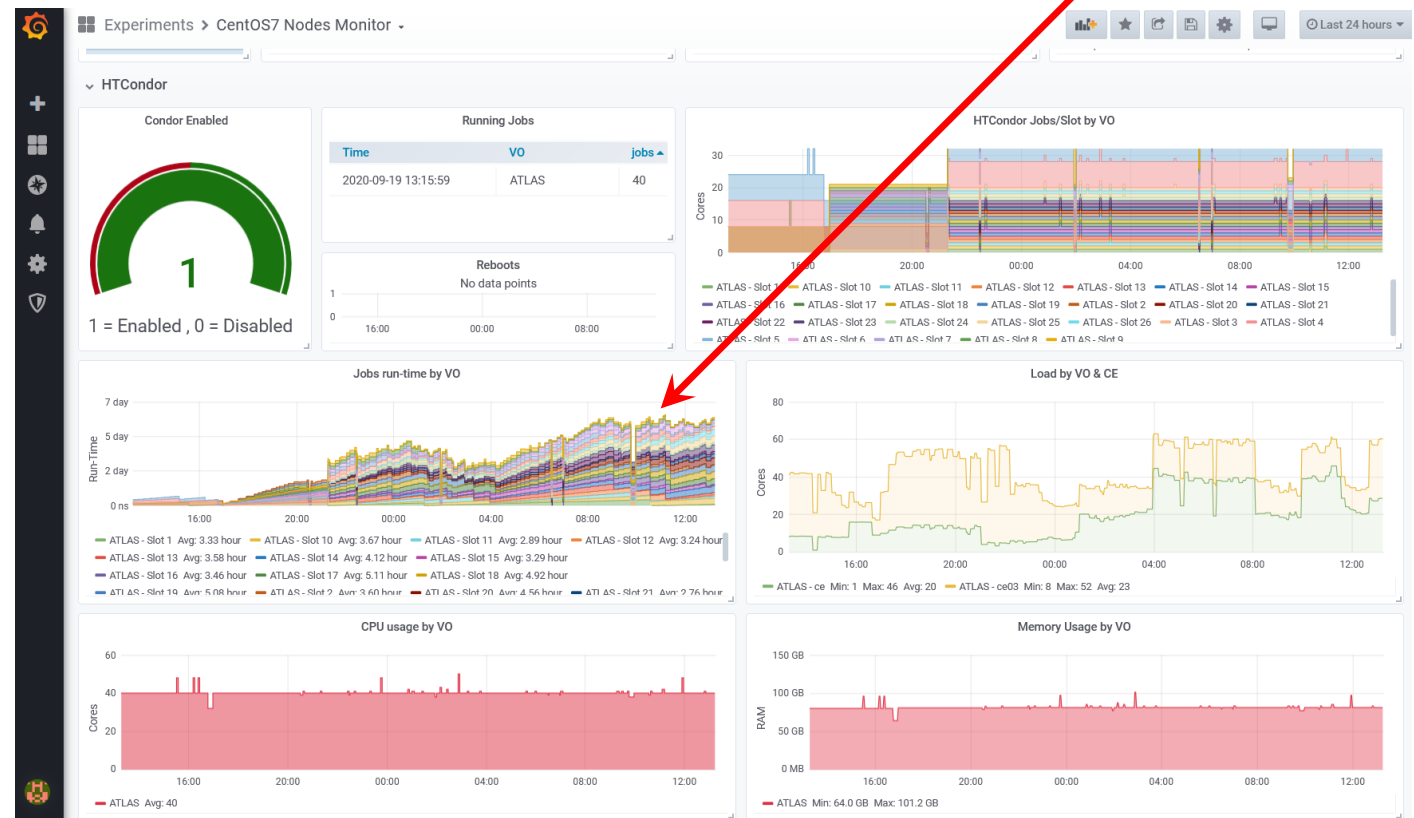
HTCondor logs:

/var/log/condor/



ClusterView Dashboards





Running jobs per VO

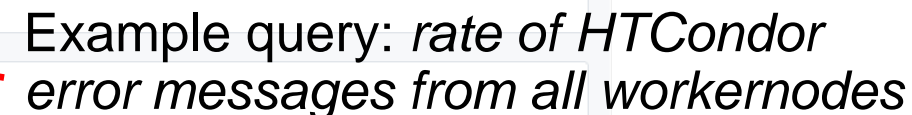


Over 50 graphs about CPU, Memory, Disks, Networking ...

Log labels Line limit auto 12.3s  

+ Add query  Query history  Query inspector

Example query: rate of



Explore

Loki_Victoria

Split

Refresh

Search

Clear All

Run Query

Save

Log labels

{job="condor"} | - "[Ee]rror"

Line limit

auto

4.2s

⌵

+ Add query

Query history

Query inspector

Logs

40

30

20

10

0

18.34

18.36

18.38

18.40

18.42

18.44

18.46

18.48

18.50

18.52

18.54

18.56

18.58

19.00

19.02

19.04

error

Time

Unique labels

Wrap lines

Dedup

None

Exact

Numbers

Signature

Flip results order

Common labels: condor

Limit: 1000 (1000 returned)

Total bytes processed: 105 MB

>

2021-04-29 15:05:18

wn-h29-831

04/29/21 15:05:17

error: problem finding resource for 484 (DEACTIVATE_CLAD_FORCELY)

>

2021-04-29 15:05:18

wn-h29-831

04/29/21 15:05:17

error: can't find resource with ClaimId (c1b.1.68.31.9618?addr=38.1.68.31-9618?sock=2045_8307_5.9161651136028989#...) -- perhaps th

>

2021-04-29 15:05:17

wn-h29-831

04/29/21 15:05:16

groupTracker (pid = 459126): fopen error: Failed to open file '/proc/65536/group'. Error: No such file or directory (2)

>

2021-04-29 15:05:00

wn-h29-841

04/29/21 15:05:00

groupTracker (pid = 24544): fopen error: Failed to open file '/proc/24544/group'. Error: No such file or directory (2)

>

2021-04-29 15:05:00

wn-h29-835

04/29/21 15:05:00

groupTracker (pid = 49590): fopen error: Failed to open file '/proc/49590/group'. Error: No such file or directory (2)

>

2021-04-29 15:05:05

wn-h29-840

04/29/21 15:05:05

groupTracker (pid = 87089): fopen error: Failed to open file '/proc/87089/group'. Error: No such file or directory (2)

>

2021-04-29 15:05:04

wn-h29-835

04/29/21 15:05:04

groupTracker (pid = 18017): fopen error: Failed to open file '/proc/18017/group'. Error: No such file or directory (2)

Log Labels:

filename

/var/log/condor/ProcLog

host

wn-h29-808

job

condor

Parsed Fields:

ts

2021-04-29T14:05:04.234Z

trace

1619785584234949114

>

2021-04-29 15:05:02

wn-h29-809

04/29/21 15:05:02

groupTracker (pid = 18003): fopen error: Failed to open file '/proc/18003/group'. Error: No such file or directory (2)

>

2021-04-29 15:05:02

wn-h29-809

04/29/21 15:05:02

groupTracker (pid = 18011): fopen error: Failed to open file '/proc/18011/group'. Error: No such file or directory (2)

>

2021-04-29 15:05:00

wn-h29-832

04/29/21 15:05:00

groupTracker (pid = 4241): fopen error: Failed to open file '/proc/4241/group'. Error: No such file or directory (2)

>

2021-04-29 15:04:49

wn-h29-838

04/29/21 15:04:49

groupTracker (pid = 30955): fopen error: Failed to open file '/proc/30955/group'. Error: No such file or directory (2)

>

2021-04-29 15:04:41

wn-h29-831

04/29/21 15:04:41

groupTracker (pid = 63866): fopen error: Failed to open file '/proc/63866/group'. Error: No such file or directory (2)

>

2021-04-29 15:04:41

wn-h29-835

04/29/21 15:04:41

groupTracker (pid = 45663): fopen error: Failed to open file '/proc/45663/group'. Error: No such file or directory (2)

>

2021-04-29 15:04:37

wn-h29-831

04/29/21 15:04:36

groupTracker (pid = 43673): fopen error: Failed to open file '/proc/43673/group'. Error: No such file or directory (2)

>

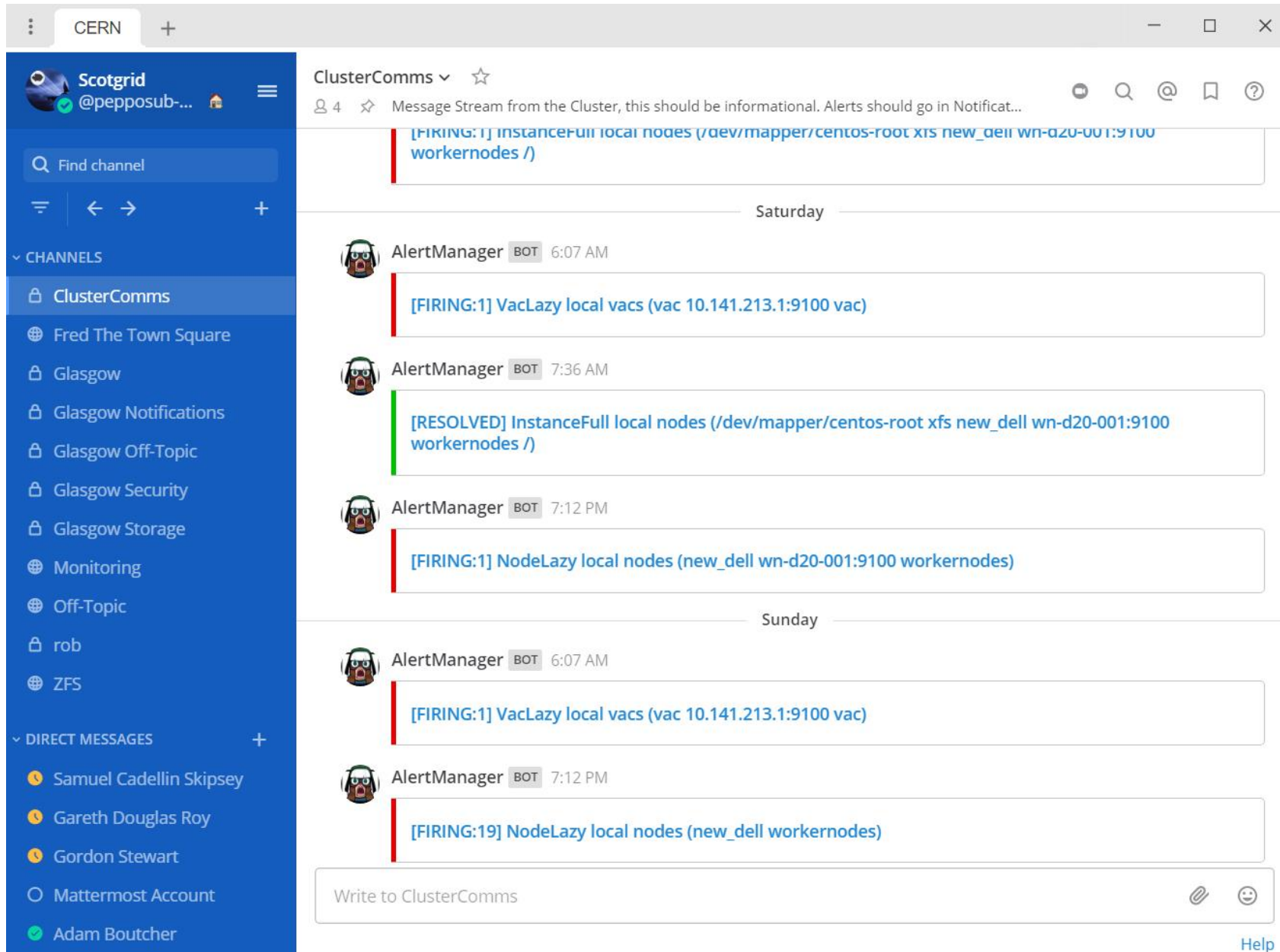
2021-04-29 15:04:34

wn-h29-841

04/29/21 15:04:34

groupTracker (pid = 19700): fopen error: Failed to open file '/proc/19700/socks'. Error: No such file or directory (2)

MatterMost Channel



The screenshot displays the Mattermost web interface. On the left sidebar, the user profile is 'Scotgrid @pepposub-...'. Below the search bar, the 'CHANNELS' section is expanded, and 'ClusterComms' is highlighted with a red arrow. Other channels listed include 'Fred The Town Square', 'Glasgow', 'Glasgow Notifications', 'Glasgow Off-Topic', 'Glasgow Security', 'Glasgow Storage', 'Monitoring', 'Off-Topic', 'rob', and 'ZFS'. The 'DIRECT MESSAGES' section lists several contacts.

The main channel view for 'ClusterComms' shows a message stream. The channel header includes a dropdown menu, a star icon, and a description: 'Message Stream from the Cluster, this should be informational. Alerts should go in Notificat...'. The message history shows several alerts from the 'AlertManager' bot, including:

- [FIRING:1] InstanceFull local nodes (/dev/mapper/centos-root xfs new_dell wn-d20-001:9100 workernodes /)
- [FIRING:1] VacLazy local vacs (vac 10.141.213.1:9100 vac)
- [RESOLVED] InstanceFull local nodes (/dev/mapper/centos-root xfs new_dell wn-d20-001:9100 workernodes /)
- [FIRING:1] NodeLazy local nodes (new_dell wn-d20-001:9100 workernodes)
- [FIRING:1] VacLazy local vacs (vac 10.141.213.1:9100 vac)
- [FIRING:19] NodeLazy local nodes (new_dell workernodes)


The interface includes a search bar, a 'Find channel' input, and a 'Write to ClusterComms' text box at the bottom. A 'Help' link is visible in the bottom right corner.

Ansible repository on our internal Git:

- ~70 Ansible roles
- ~50 Ansible play-books (incl. maintenance)

~70 Ansible roles

~50 Ansible play-books (incl. maintenance)



Ansible

Project ID: 1

🔔

★ Star

1

🍴 Fork

0

🔗 636 Commits

🌿 2 Branches

🏷️ 0 Tags

📁 296.2 MB Files

💾 296.2 MB Storage

Ansible roles to build up the whole Glasgow site :)

master

ansible /

+


History

🔍 Find file

Web IDE

📄

Clone



VAC certificate role added

Emanuele Simili authored 6 days ago

4d860844

📄

Add README

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING


Enable Auto DevOps

Add Kubernetes cluster

Set up CI/CD

Name	Last commit	Last update
📁 cve	fixed loki installer	1 month ago
📁 docs	updated pw file; mailing lists; updated perfSONAR installer; boot script for dual in...	1 week ago
📁 inventory	further development of alerting rules etc. (post vCHEP2021)	1 month ago
📁 playbooks	VAC certificate role added	6 days ago
📁 roles	VAC certificate role added	6 days ago
📁 vars	further development of alerting rules etc. (post vCHEP2021)	1 month ago
🔥 .gitignore	.gitignore fix	1 year ago
📄 ansible.cfg	logs	1 month ago

Internal wiki with
detailed procedures
and code snippets
for cluster built:
~50 pages


 ScotGrid

You are here: <<

Logged in as: Sharnice Smith (page) | Update Profile | Admin | Log Out
[Recent Changes](#) | [Media Manager](#) | [Help](#)

ScotGrid Glasgow

These Wiki pages contain information and practical procedures for the administration of the Glasgow computing cluster as part of the ScotGrid Scotland project at the Universities of Durham, Edinburgh and Glasgow.



Getting Started

- Logging in via SSH and PuTTY (ssh)
- Log into the VICEG User(s)
- Log into the ScotGrid Cluster (admin)
- Managing ScotGrid Billing Lists
- Cluster legal notices and inventory of hardware installed within Data Center
- Cluster/Facet automated inventory (as public link)

Documentation & Tools

- U-GuideWiki pages & admin setup
- U-Guide Repository & user setup
- Unraid ecosystem reference page
- Host Certificate request manual and command the code
- GOC-QS service registration

Regular Services

- ProtonMail WebAppLinker
 - Quickstart (usage)
 - FAQ (faq)
- PROXMO Provisioning System (Glasgow & repo)
 - PROXMO basic-to-vmware
 - The PROXMO Process
 - PROXMO Configuration and adding a New OS Drive
- Ramote Configuration Management with Jinja2 (Glasgow & repo)
 - Roles and play-books for Service setup - Jinja2
 - Roles and play-books for Network management - Jinja2
- Central Backup Server - BCD

Operational Intelligence

- Monitoring & Alerting and Centralised Logging (resource map)
 - CanNot cluster for dashboard
 - Centralised Logging for query
 - CanNot cluster for log files (QueryURL to Mailbox)
- Network Flow and SNMP Monitoring (resource map)
 - ElasticFlow Dashboard on HBase (to publish temporary down)
- perftestWARR monitoring measurement tools?
 - Local perftestWARR server for perftest analysis
 - Full test workflow view
- Other monitoring resources (external links)
- CSI Accounting Summary In a Flash table (Flash server page)

Hardware & Network

- Network Servers: DNS, DHCP, NUT
- Baremetal Requests
- Hardware Setup
 - Dashley PUE and Intel i70-GSS cards
 - ORILL Chassis the Dell Remote Access Controller
 - HP LTO the HP Integrated Lights-Out
 - IP Switch manager network commands for re-provisioning
- Host Security page (usage)
- Switches configuration
 - Inter-Lan switch configuration

Grid Services

- Workload scheduler configuration (CanNot?)
- VDC nodes configuration (...)
- CVMATP CERN Virtual Machine File-system
- Squid Cache Services
- Grid Mapping (VDC, VOUGES Grid Map Files and Post-accounts)
 - Add new VO with our online automated procedure
- JRCUS Authentication Service
 - Authenticate a Policy on JRCUS
 - JRCUS/Guest Check the Authentication Chain
- JRC-CSE the JRC Compute Element
 - Job execution submission on JRC-CSE
 - Integration test JRC-CSE + HTC under JRCUS

Table of Contents

[Getting Started](#)
[Documentation & Tools](#)
[Regular Services](#)
[Operational Intelligence](#)
[Hardware & Network](#)
[Grid Services](#)
[Under Construction / Missing](#)
[Old](#)
[External Links](#)

Glasgow Tier-2 Cluster

Submission system consists of a few hundred (*) nodes managed by **NorduGrid ARC**, **ARGUS** and **HTCondor**. It provides computing power to LHC experiments (ATLAS, CMS) and non-LHC experiments (LIGO, CLAS12, ...).

- We have 4 ARC-CE server for job submission: **ce01**, **ce02**, ...
- We have ~300 * physical workernodes (~80 connected): **wn00x**
- We have 1 ARGUS for user authentication and authorization
- We have 1 HTCondor manager which collects scheduled jobs
- We have 2 Squids acting as a cache for CVMFS
- We have 2 NAT servers for communicating with the outside world
- We have ~20 VAC nodes *: **vac00x**
- We have a local Storage Element (SE) **
- And, of course, everything is within our private VPN (WireGuard)

* Beside the conventional Grid submission system, the Glasgow site also runs a smaller number of VAC machines to provide an Infrastructure-as-a-Service (IaaS) cloud for various other experiments, including LHCb.

** The Storage Element is composed of three sets of physical machines running Ceph, including about 20 dedicated storage machines or disks (**cephd0x**), six physical caches (**cephc0x**) and three entry points or servers (**cephs0x**).

UKI-SCOTGRID-GLASGOW

- Part of the GridPP collaboration providing resources to the Worldwide LHC Compute Grid (WLCG).
- One of 19 institutions comprising 4 distributed Tier-2 sites (SCOTGRID, NORTHGRID, SOUTHGRID and LT2).
- Part of the SCOTGRID Distributed Tier-2 including Glasgow, Edinburgh and Durham Universities.

