SUSE Cluster Monitoring Solution

Diego Akechi
Engineering Manager
dakechi@suse.com

Dario Maiocchi
Software Engineer
dmaiocchi@suse.com
We are hiring!

Linux Solutions Developer (Core Cluster components development)

Solutions Developer - SAP Basis infrastructure solutions

And many more at:
Introduction
Transforming your Datacenter

**Mode 1**
Classic deployment methods and infrastructure
Reproduce and install manually using best practices

**Mode 1.5**
Part of the transformation to a more modern data center
Support for public cloud technologies
Modern deployment methods:
Reduce effort and time to deploy applications
Repeatable deployment
Infrastructure as code (version, change, and control)

Enhance 'Day 2’ – Monitoring capabilities
Monitoring HA Landscapes

Monitoring is important, but there are challenges:

Landscapes are getting more and more complex
SUSE HA can be viewed with HAWK for a single cluster
Comprehensive monitoring can be complex to setup

We need a simpler way to deal with this
Who is behind HA Cluster Monitoring:

*Thanks to people* who has contributed/developed the project:

https://github.com/kermat
https://github.com/ashleyprimo
https://github.com/MalloZup,
https://github.com/stefanotorresi
https://github.com/arbulu89
https://github.com/Simranpal
https://github.com/gao-yan .. and many others !!!
Cluster Monitoring Solution
Technology Architecture:

Prometheus and Grafana:
Flexible and easy to use
Easy to Integrate
Prometheus provides a time series database
Grafana renders and visualize data, with easy to modify and adapt dashboards

Prometheus Exporters:
Node_exporter - node metrics (CPU, disk, NFS ....)
Ha_cluster_exporter – cluster stack (pacemaker, drbd, corosync ....)
Application specific exporters: SAP HANA, Oracle, Netweaver, etc.
Architecture Overview (1)
Monitoring deployment:

1) Manually

2) Automatic deployment
1) Manually:

1) Install on each node the prometheus exporters (install package and start services, etc.)

2) Configure grafana and prometheus servers, that get data from exporters
2) Automated deployment (cloud or libvirt)

SAP HANA

SLES and Prometheus

SAP HANA

..
Prometheus and Grafana
Why Prometheus and Grafana?

Prometheus is an open-source systems monitoring and alerting toolkit.

Prometheus joined the Cloud Native Computing Foundation in 2016 as the second hosted project, after Kubernetes.
Why Prometheus and Grafana?

Prometheus has exporters for many different applications. (most every major opensource project), and big opensource community behind it

It is the main monitoring solution for containers and combining it with Grafana makes possible to create a single panel of glass for not only HA clusters, but mixed applications environments...
Why Prometheus and Grafana?
Prometheus Querying:

Enable query history

- ha_cluster_pacemaker_resources_total

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ha_cluster_pacemaker_resources_total{instance=&quot;192.168.110.19:9002&quot;,job=&quot;hana-cluster&quot;}</td>
<td>6</td>
</tr>
<tr>
<td>ha_cluster_pacemaker_resources_total{instance=&quot;192.168.110.20:9002&quot;,job=&quot;hana-cluster&quot;}</td>
<td>6</td>
</tr>
</tbody>
</table>
Grafana Graph creation:
Grafana dashboards:

**Comprehensive Dashboards**, allowing insight into your landscape

Get notified of issues with **active alerts and reporting**

Provides **real-time and historic views** to ease troubleshooting

Based on flexible, powerful open-source technology

Protects value of existing monitoring solution by Integrating with existing monitoring tools.

All dashboard can be configured, changed adapted fitting special needs.

New dashboards can be created in a simple fashion
Grafana dashboards:
HA Cluster Dashboard
Multi-cluster dashboard

Dashboard manage important tasks of different clusters.

- Show Failed resources, Stopped/Failed Services on the Clusters

- Show statistics, like summary of number of Failures, number of Warnings, number of healthy clusters.
ha_cluster_exporter
What is an exporter?

An exporter exposes metrics which prometheus will get via http requests.
http://NODE_IP:PORT_NUMBER/metrics

```yaml
# HELP ha_cluster_corosync_quorate Whether or not the cluster is quorate
# TYPE ha_cluster_corosync_quorate gauge
ha_cluster_corosync_quorate 1 1580314752621

# HELP ha_cluster_corosync_quorum_votes Cluster quorum votes; one line per type
# TYPE ha_cluster_corosync_quorum_votes gauge
ha_cluster_corosync_quorum_votes{type="expected_votes"} 2 1580314752621
ha_cluster_corosync_quorum_votes{type="highest_expected"} 2 1580314752621
ha_cluster_corosync_quorum_votes{type="quorum"} 1 1580314752621

ha_cluster_corosync_quorum_votes{type="total_votes"} 2 1580314752621

# HELP ha_cluster_corosync_ring_errors_total Total number of corosync ring errors
# TYPE ha_cluster_corosync_ring_errors_total gauge
ha_cluster_corosync_ring_errors_total 0 1580314752605

# HELP ha_cluster_pacemaker_config_last_change The timestamp of the last change of the cluster configuration
# TYPE ha_cluster_pacemaker_config_last_change counter
ha_cluster_pacemaker_config_last_change 1.580318305e+09 1580314752639

# HELP ha_cluster_pacemaker_fail_count The Fail count number per node and resource id
# TYPE ha_cluster_pacemaker_fail_count gauge
ha_cluster_pacemaker_fail_count{node="h-nhana01",resource="rsc_SAPHanaTopology_PRD_HDB00"} 0 1580314752621
ha_cluster_pacemaker_fail_count{node="h-nhana01",resource="rsc_SAPHana_PRD_HDB00"} +Inf 1580314752621
ha_cluster_pacemaker_fail_count{node="h-nhana02",resource="rsc_SAPHanaTopology_PRD_HDB00"} 0 1580314752621
ha_cluster_pacemaker_fail_count{node="h-nhana02",resource="rsc_SAPHana_PRD_HDB00"} 0 1580314752621
ha_cluster_pacemaker_fail_count{node="h-nhana02",resource="rsc_ip_PRD_HDB00"} 0 1580314752621
ha_cluster_pacemaker_fail_count{node="h-nhana02",resource="stonith-sbd"} 0 1580314752621

# HELP ha_cluster_pacemaker_migration_threshold The migration_threshold number per node and resource id
```

SUSE. SUSEcon 20
HA_CLUSTER exporter

The exporter exposes drbd, corosync, sbd and pacemaker metrics. (all cluster components)
HA_CLUSTER exporter (installation)

Manual installation:

1) Copy the binary on the node or install the RPM package.

2) Run the service or just run the binary in daemon mode. (ha_cluster_exporter --help.)
By design of exporter, a metric will not visible only if your node has the needed component.

Example: if your node has only drbd but not pacemaker, the exporter will only export valid drbd metrics. (it detect in smart ways the presence of binaries)
HA_CLUSTERS exporter (metrics)

Metrics naming:

ha_cluster_COMPONENT_NAME_metric.

Examples:

ha_cluster_sbd_devices_total
ha_cluster_drbd_resources
ha_cluster_pacemaker_fail_count
Monitoring Alerts
Alerts

1) Prometheus alerts

2) Grafana alerts
Prometheus alerts

They are triggered when an event/alert occurs, many different channels are configurable.
Prometheus alerts

groups:

# sap alerts
- name: sap-hana-resource-monitoring
  rules:
  - alert: sap-hana-master-resource-down
    expr: absent(ha_cluster_pacemaker_resources{resource="rsc_SAPHana_.*",role="master",status="active"})
    labels:
      severity: page
    annotations:
      summary: Primary SAP-HANA resource down

  - alert: sap-hana-secondary-resource-absent
    expr: absent(ha_cluster_pacemaker_resources{resource="rsc_SAPHana_.*",role="slave",status="active"})
    labels:
      severity: page
    annotations:
      summary: Slave SAP-HANA resource absent
Grafana vs Prometheus alerts

Grafana alerts intent is to be helpful for reading the dashboard and require an operator/human observing the dashboards.

Prometheus alerts are events sent via mails channels and doesn’t require someone observing the dashboards.
Grafana alerts

![Node CPU Load Graph](image)
Further information:

Deployment project:
https://github.com/SUSE/ha-sap-terraform-deployments

Prometheus Exporter projects
https://github.com/SUSE/hanadb_exporter
https://github.com/ClusterLabs/ha_cluster_exporter
Thank you!
Unpublished Work of SUSE LLC. All Rights Reserved.
This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer
This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.