Orchestrator - Feature #45410

cephadm: Support upgrading alertmanager, grafana, prometheus and node_exporter

05/06/2020 03:53 PM - Sebastian Wagner

Status:	New	**% Done:**	40%
Priority:	Normal		
Assignee:			
Category:	cephadm/monitoring		
Target version:			
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:		**Pull request ID:**	

Description

Right now, we're simply downloading :latest, which might even differ between daemons on different hosts.

Subtasks:

Feature # 45463: cephadm: allow custom images for grafana, prometheus, alertmanager and...	**Resolved**
Feature # 45859: cephadm: use fixed versions	**Resolved**
Documentation # 45860: cephadm: document upgrades of monitoring components	**New**
Feature # 45864: cephadm: include monitoring components in usual upgrade process	**New**
Feature # 46499: Requesting a "ceph orch redeploy monitoring" command, as an option, so...	**New**

Related issues:

Related to Orchestrator - Documentation #45411: cephadm: add section about co...	**Resolved**
Related to Ceph - Bug #45908: monitoring: Status Panel breaks with Grafana 6....	**New**

History

#1 - 05/06/2020 03:53 PM - Sebastian Wagner

- Category changed from cephadm to cephadm/monitoring

#2 - 05/06/2020 04:04 PM - Juan Miguel Olmo Martínez

It would be nice to have this two things:
1. Use by default fixed versions images of the different components of the monitoring stack
2. Provide an easy way to use other images.

#3 - 05/06/2020 04:18 PM - Patrick Seidensal

This might not be an issue for minor version upgrades in Grafana and Prometheus, although it would be hard to guarantee that (if minor versions can be upgraded without us being able to verify that they work as expected). But I think it is necessary to be able to upgrade minor versions for security reasons. I'm not even sure if we can use some kind of tag or label to achieve that. We should prevent upgrades to new and (on our side) untested major versions. If there's no mechanism to achieve that, we might need to stick to fixed versions. But then it is our responsibility to upgrade promptly if security issues have been fixed and those versions have been released.

I'm a little bit concerned about the Node exporter, though. Minor version upgrades have broken metric names in the past. On the other hand, there's a pre-release of v1.0.0 available which might indicate that things as metric names might stay stable within minor version upgrades in the future. Of course this won't be an issue if we decide to use fixed versions and test all (including minor version) upgrades beforehand.

#4 - 05/07/2020 07:22 AM - Sebastian Wagner

- Related to Documentation #45411: cephadm: add section about container images added

#5 - 05/07/2020 10:03 AM - Alfonso Martínez

These are the monitoring stack versions that we use in our nautilus-based releases:
grafana: 5.4.3
prometheus: v2.7.2
alertmanager: 0.16.2
node_exporter: 0.17.0

grafana plugins:
https://docs.ceph.com/docs/master/mgr/dashboard/#enabling-the-embedding-of-grafana-dashboards
Not sure, but we can assume that today latest versions of mentioned plugins here are a good start:
grafana-piechart-panel: 1.4.0
vonage-status-panel: 1.0.9

#6 - 05/07/2020 10:24 AM - Patrick Seidensal

This are our current versions

Grafana 5.3.3
Alertmanager 0.16.2
Prometheus 2.11.1
Node exporter 0.17.0

grafana-piechart-panel 1.3.6
grafana-status-panel 1.0.9

#7 - 05/08/2020 12:00 PM - Patrick Seidensal

It currently seems that using fixed versions for monitoring stack containers are the only way to be ensure that major versions of those applications aren't automatically updated. Being responsible to publish updates, that won't break functionality, includes a responsibility to upgrade those applications when security issues arise.

To be at least notified about upcoming security vulnerabilities, Clair can be used to have those images checked, even automatically.

Clair is a tool that checks for security vulnerabilities in container images. It uses a a PostgresQL database, which is by default populated with CVEs from various different sources like the Debian Security Bug Tracker, Ubuntu CVE Tracker, Red Hat Security Data, SUSE OVAL Descriptions and others.

It is used in products like quay.io from Red Hat.

Clair can be integrated into Container registries for automatic security checks, like the aforementioned product.

Through tools like klar, clair can be used without an integration to registries and is capable of checking local and remote container images for security vulnerabilities.

This is how it such a check might look like:

```
user@home ~ » CLAIR_ADDR=localhost klar grafana/grafana:latest


clair timeout 1m0s
docker timeout: 1m0s
no whitelist file
Analysing 8 layers
```

```
Got results from Clair API v1
Found 0 vulnerabilities
```

#8 - 06/05/2020 01:53 PM - Patrick Seidensal

- Related to Bug #45908: monitoring: Status Panel breaks with Grafana 6.7.0 (maybe 7.x too) added