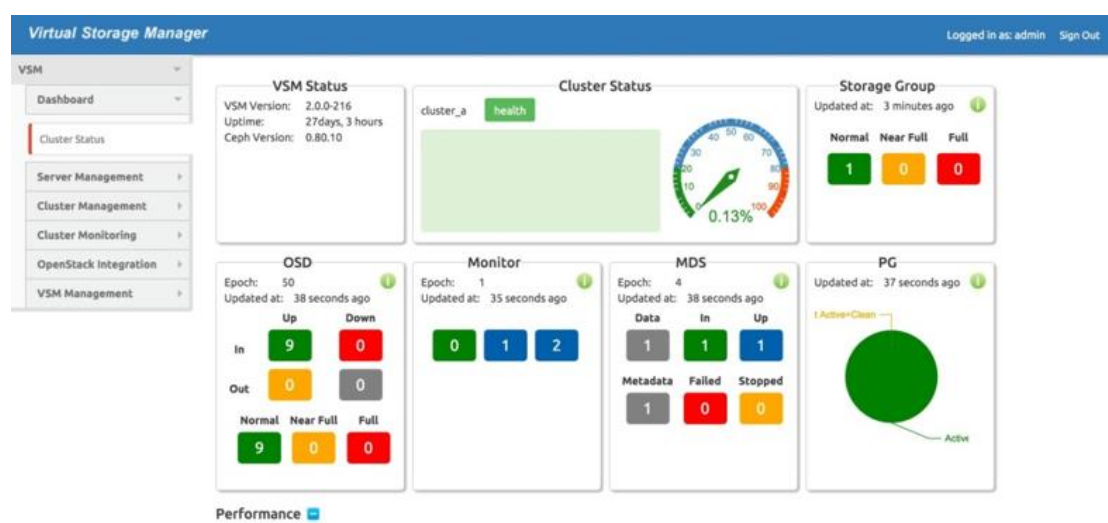


Ceph 开源管理监控平台分析

现状

Ceph 的开源管理监控平台有如 VSM（三年前最后更新，read-only），InkScope，Calamari, Suse-enterprise-storage（SUSE 收费服务）、CEPH DASHBOARD, prometheus + grafana 等；

VSM（Virtual Storage Manager）



概览图

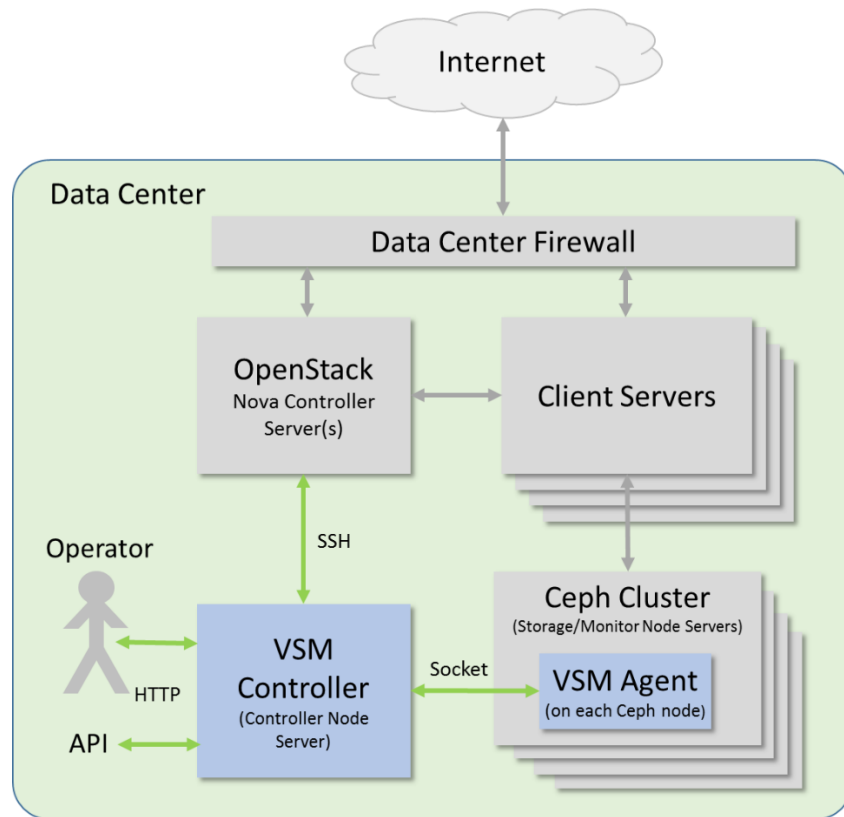
GitHub 主页:

<https://github.com/intel/virtual-storage-manager>

VSM 架构

VSM 的设计是完全按照 OpenStack 的架构设计。遵循一定的原则来设计这个架构:

- 容易与 OpenStack 的模块集成
- 完全与其它的 OpenStack 模块解耦合
- 易于合并到其他云平台
- 具有高可靠性，大规模数据中心的高可用性
- 易于使用，部署和管理



VSM 架构图

VSM 分为核心的两个部分，VSM 控制节点与 VSM 存储节点。

VSM 控制节点

- WebUI – 通过访问 VSM REST API 用于集群的管理、监控
- REST API – 供 vsm client 访问
- mariadb, rabbitmq

VSM 存储节点

- 使用 diamond 收集 ceph 节点的监控信息
- vsm-agent 工具对 ceph 节点进行管理

组件

- Dashboard (vsm-dashboard): VSM 的 webUI 界面，用户通过 Dashboard 来管理与监控 ceph 集群
- vsmclient (python-vsmclient): VSM restapi 调用的 client
- API (vsm): VSM 的 restapi
- scheduler (vsm): VSM 的调度组件
- conductor (vsm): VSM 的数据库操作组件，即所有的数据库操作都是通过 conductor

- 来调用 mysql
- RabbitMQ: 消息中间件, VSM 的各个组件相对独立, 都是通过发送消息, 通过 RPC 的方式来相互调用
- agent (vsm): VSM 代理服务

组件特点

- 分布式: 分开独立部署
- 无状态: 各个请求独立, 可扩展性强
- RESTFUL
- RPC
- plugin: 插件式设计, 松耦合

组件代码

- VSM 目前最新的发行版本为 2.1, VSM 的代码组件分为四个:
- vsm-dashboard: VSM 的管理与监控 web 界面
- python-vsmclient: 调用 restapi 的 client
- vsm: VSM 的核心组件 (包括 api、scheduler、conductor、agent 等)
- vsm-deploy: ceph 部署工具

代码基于 Python 语言, 使用了 wsgi、django 等技术框架

支持功能

- 仪表盘: 查看 vsm、cluster、storage group、OSD、MON、MDS、PG 的状态统计信息
 - 🚦 可以判断 OSD 是否正常运行, 空间是否满
 - 🚦 查看 IOPS、latency、bandwidth、CPU 实时监控信息(通过 diamond 实现数据的收集)
 - 🚦 可以用来发现 ntp 延迟的问题
- 所有的宿主节点都需要在安装 vsm 的时候写在配置文件中
- 添加删除 MON/OSD 守护进程
- OSD 增删、重启、恢复 (N/A)
- osd pool 的管理 – 支持 cache tier 的增删、replicated/EC pool 的创建
- StorageGroup 的管理 – 添加新的 SG, 存储资源将以 SG 为单位进行统计
- 支持 Ceph 系统的升级功能, 通过 github 下载源码实现
- 将通过 ssh 配置 openstack 的控制节点把 rbd pool present 给 cinder
- 管理系统的临界值, 将在 dashboard 中得到体现
- vsm 账户管理 (keystone 管理账户)

优缺点

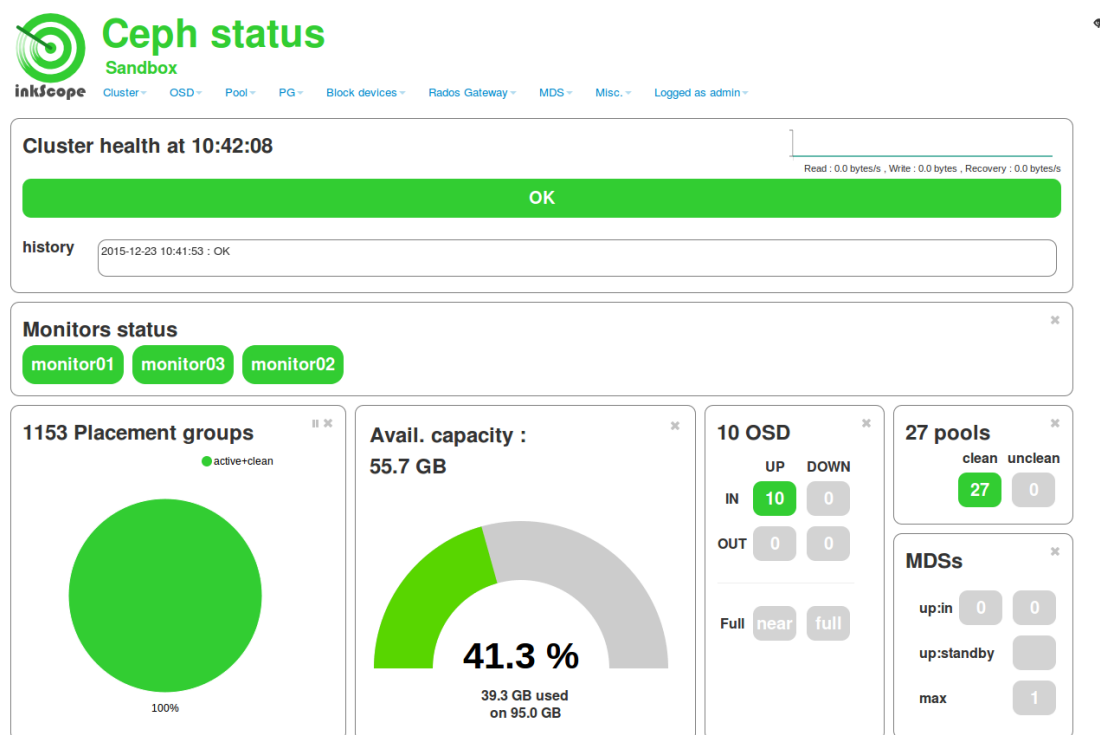
优点:

1. 管理功能完善、充足
2. 界面友好
3. 可以部署 Ceph 和监控 Ceph
4. 与 OpenStack 一脉传承, 设计风格类似 (详见架构部分说明)

缺点:

1. 非官方, 社区维护, 且目前已处于归档状态 (read-only)
2. 依赖 OpenStack 某些包和组件
3. 封装一套自己的 rest-api, 代码复杂度较高

Inkscope

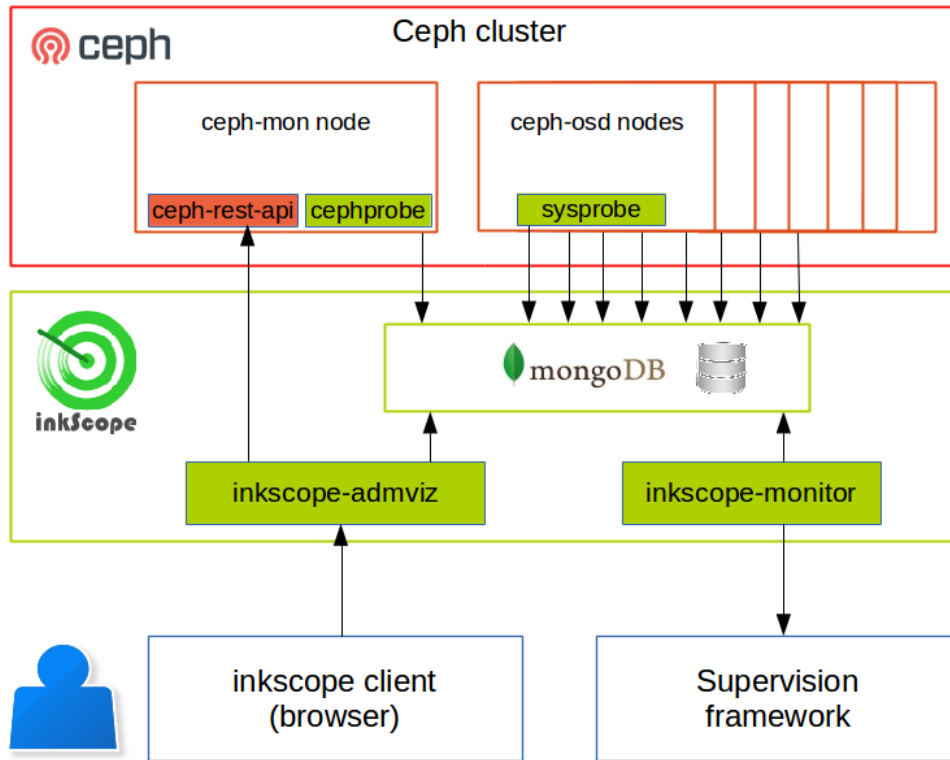


概览图

GitHub 主页:

<https://github.com/inkscope/inkscope>

Inkscope 架构



Inkscope 架构图

组件

- inkscope-common: 包含 inkscope 的默认配置文件以及其他进程(cephprobe,sysprobe)启动所需的依赖文件, 所有相关节点都需要安装。
- inkscope-admviz: 包含 inkscope 的 web 控制台文件, 含接口和界面, 仅需要安装一个, 该节点 (管理节点) 上同时需要按安装 flask 和 mongodb
- inkscope-cephrestapi: 用于安装启动 ceph rest api 的脚本, 仅需要安装在提供 api 接口的节点上, 即 mon 节点。
- inkscope-cephprobe: 用于安装启动 cephprobe 的脚本(整个集群只需一个), 安装在 mon 节点, 脚本主要实现: 获取 Ceph 集群的一些信息, 并使用端口 (5000) 提供服务, 将数据存入 mongodb 数据库中。
- inkscope-sysprobe: 安装用于所有 mon 和 osd 的 sysprobe 所需要脚本, 即所有节点均安装, 实现获取节点设备资源信息如: CPU、内存、磁盘等等。

组件代码

- inkscopeViz: Web 客户端
- inkscopeCtrl: inkscope 的服务器端, 提供了 REST API
- inkscopeProbe: 收集 ceph 节点的系统信息, 收集到的数据将传输到 MongoDB
 - ✚ cephprobe: 用来或者集群的相关信息和操作的
 - ✚ sysprobe: 获取节点的磁盘分区等相关信息的
- inkscopeMonitor: 对接第三方监控框架

代码基于 python 语言, 使用了 wsgi、flask、MongoDB 等技术框架

支持功能

- 仪表盘: 查看 cluster、OSD、MON、MDS、PG 的状态统计信息可以判断 OSD 是否正常运作, 空间是否满
- 分模块管理 OSD、Pool、PG、RBD、RadosGW、MDS
- inkscope 账户管理

优缺点

优点:

1. 易部署
2. 轻量级
3. 灵活 (可以自定义开发功能)

缺点:

1. 监控选项少
2. 缺乏 Ceph 管理功能

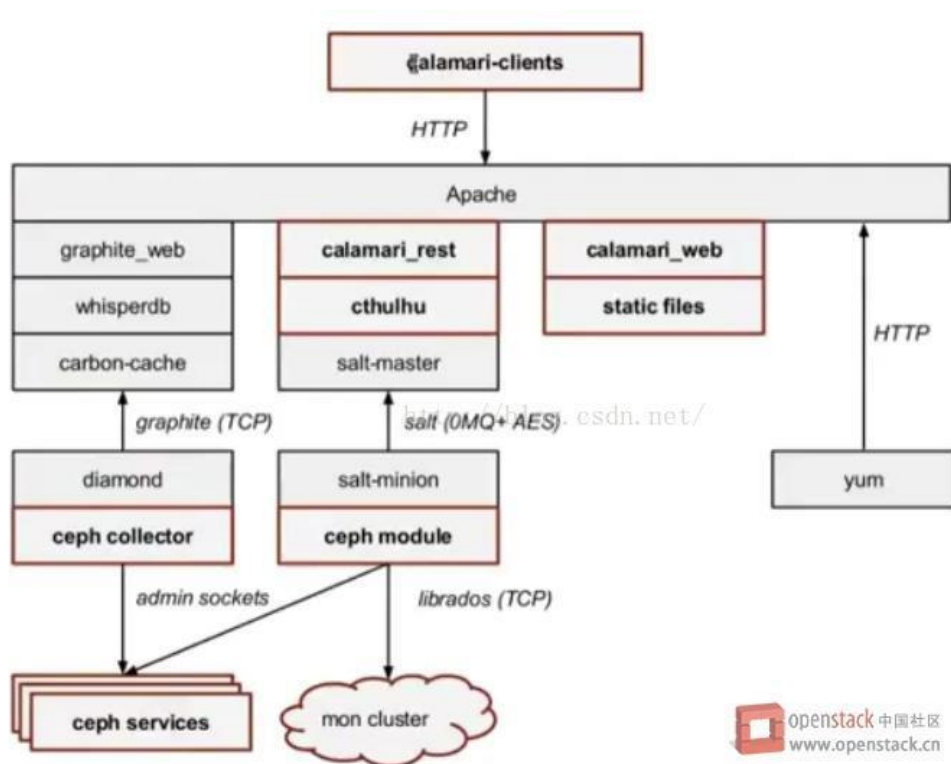
Calamari



概览图

Github 地址: <https://github.com/ceph/calamari>

架构



- Calamari 监控平台, 使用 Apache 做服务器
 - ✚ Calamari-server: 服务端
 - ✚ Calamari-client: 客户端, 包括了 web 端的 UI, 可以自己定制

- ✚ graphite: 收集数据的存储与展现, 提供了接口, 可以获取指定间隔的统计量。可以通过服务器地址单独访问 {calamari}/graphite/dashboard。包含在 Calamari 中
- Salt 服务器基础架构管理平台, 具备配置管理、远程执行、监控等功能。
 - ✚ Salt-master Salt 的服务管理端
 - ✚ Salt-minion Salt 的节点端
- diamond 各个节点的数据收集器, 收集节点信息发送到服务端。Calamari 定制了一个版本。

组件

Calamari 包含的组件主要有 calamari-server; romana; salt-minion; salt-master; diamond。这些模块各自的作用:

calamari-server: 这个是提供一个与集群进行交互, 并且自己封装了一个自己的 API, 做集中管理的平台, 这个只需要在集群当中的某一台机器上安装, 也可以独立安装;

romana: 就是原来的 calamari-client, 是一个 web 的界面, 现在已经更名为 romana, 这个也是只需要在集群当中的某一台机器上安装, 也可以独立安装, 需要跟 calamari-server 安装在一台机器上;

salt-master: 是一个管理的工具, 可以批量的管理其他的机器, 可以对安装了 salt-minion 的机器进行管理, 在集群当中, 这个也是跟 calamari-server 安装在一起的;

salt-minion: 是安装在集群的所有节点上的, 这个是接收 salt-master 的指令对集群的机器进行操作, 并且反馈一些信息到 salt-master 上;

diamond: 这个是系统的监控信息的收集控件, 提供集群的硬件信息的监控和集群的信息的监控, 数据是发送到 romana 的机器上的, 是由 romana 上的 carbon 来收取数据并存储到机器当中的数据库中。

Graphite 不仅是一个企业级的监控工具, 还可以实时绘图。Graphite 后端运行一个名为 carbon-cache.py 的 python 程序, 是高度可扩展的事件驱动的 I/O 架构的后端进程, 负责处理客户端节点上的业务数据, 它可以有效地跟大量的客户端通信并且以较低的开销处理大量的业务量。配置文件位于 /etc/graphite/carbon.conf

Calamari 使用了 Saltstack 让 Calamari Server 和 Ceph server node 通信。Saltstack 是一个开源的自动化运维管理工具, 与 Chef 和 Puppet 功能类似。Salt-master 发送指令给指定的 Salt-minion 来完成对 Ceph Cluster 的管理工作; Salt-minion 在 Ceph server node 安装后都会从 master 同步并安装一个 ceph.py 文件, 里面包含 Ceph 操作的 API, 它会调用 librados 或命令行来最终和 Ceph Cluster 通信。

cthulhu 可以理解是 Calamari Server 的 Service 层, 对上为 API 提供接口, 对下调用 Salt-master。但是代码美中不足的是 calamari_rest 有些功能直接调用了 Salt-master 而没有调用 cthulhu。calamari_rest 提供 Calamari REST API, 详细的接口请大家参照官方文档。Ceph 的 REST API 是一种低层次的接口, 其中每个 URL 直接映射到等效的 CEPH CLI; Calamari REST API 提供了一个更高层次的接口, API 的使用者可以习惯的使用 GET/POST/PATCH 方法来操作对象, 而无需知道底层的 Ceph 的命令; 它们之间的主要区别在于, Ceph 的 REST API 的使用者需要非常了解 Ceph 本身, 而 Calamari 的 REST API 更贴近对 Ceph 资源的描述, 所以更加适合给上层的应用程序调用。

supervisord 是一个允许用户监控和控制进程数量的系统程序。它可以指定一个服务如何运行。

romana (calamari_clients) 是一个提供 web UI 的模块，主要为客户端使用 Calamari API 提供服务，由 salt-minion 和 diamond 组成。

Diamond 负责收集监控数据，它支持非常多的数据类型和 metrics，通过查看源代码，它支持 90 多种类型的数据；每一个类型的数据都是上图中的一个 collector，它除了收集 Ceph 本身的状态信息，它还可以收集关键的资源使用情况和性能数据，包括 CPU，内存，网络，I/O 负载和磁盘指标，而且还能收集很多流行软件的性能指标，包括 Hadoop, Mongo, Kafka, MySQL, NetApp, RabbitMQ, Redis, and AWS S3 等。Collector 都是使用本地的命令行来收集数据，然后报告给 Graphite。

romana 包括 dashboard、login、admin、manage 四大模块，构建 rpm 软件包时，这些模块缺一不可

dashboard 是一个 javascript 的客户端，直接与 ceph restful api 交互来管理 ceph。dashboard 包含 3 个逻辑部分，分别为 dashboard、workbench、graphs。

- dashboard 是一个只读的视图，负责展现 ceph 集群的健康状态

- workbench 是后台 OSD 和 host 的虚拟展现，最多限制展现 256 个 OSD

- graphs 是有负责展示图形的 graphite 和负责在每个节点收集数据的 diamond 共同展示各种度量数据的视图

login 模块用于登录 web 界面

admin 模块用来管理用户和 calamari 信息的管理工具

manage 模块用于管理 ceph 集群中的各种应用，如 OSD 管理、pool 管理、集群设置和集群日志展现等功能

calamari_clients 是一套用户界面，Calamari Server 在安装的过程中会首先创建 opt/calamari/webapp 目录，并且把 webapp/calamari 下的 manager.py(django 配置)文件考进去，calamari_web 的所有内容都要放到 opt/calamari/webapp 下面来提供 UI 的访问页面。

calamari-web 包下面的文件提供所有 web 相关的配置，calamari_rest 和 calamari_clients 都要用到。

优缺点

优点

1. 轻量级
2. 官方化
3. 界面友好

缺点

1. 不易安装
2. 管理功能滞后
3. 提供的管理功能太少

Calamari 为 Ceph 的运维和管理提供了一个统一的平台，而且用户还可以基于这个平台扩展自己的存储管理产品，但同时也存在着不足和需要改进的地方。

首先，Calamari 还不能完成 Ceph deploy 所实现的部署功能，这是它最大一个不足；Fuel 可以完成部署功能，并且可以选择 Ceph server 的数据盘和日志盘以及定制默认的备份数等，所以 Calamari + Fuel 可以来实现一个完成的基于 Ceph 的部署和管理工具。

其次，Calamari 提供的管理功能太少，用户无法只使用它来运维一个 Ceph 环境。

最后，用户可以基于 Calamari 开发自己的 Ceph 管理软件，UI 部分可以修改 calamari_clients 的页面，也可单独实现一套自己的 UI 基于 calamari_rest 和 Graphite_web，后端的功能的监控部分可以扩展 diamond 的 collector 实现，管理 Ceph 的功能可以扩展 rest api, cthulhu, salt 等来实现。

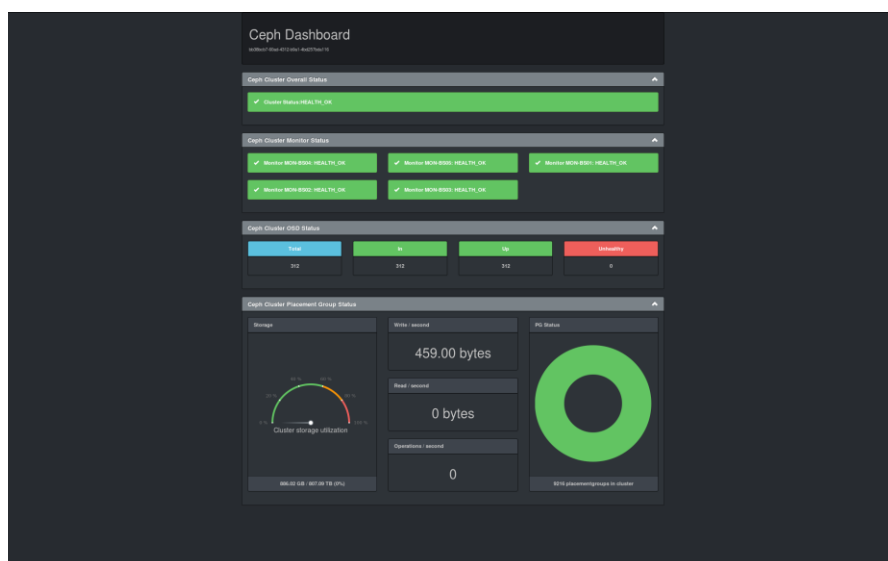
安装参考

<https://www.gitbook.com/book/zphj1987/calamaribook>

<https://wiki.shileizcc.com/confluence/display/CEPH/Ceph%20Calamari>

Ceph-Dash

Ceph-Dash 是用 Python 开发的一个 Ceph 的监控面板，用来监控 Ceph 的运行状态。同时提供 REST API 来访问状态数据。



概览图

Github 地址: <https://github.com/Crapworks/ceph-dash>

优缺点

优点


1. 易部署
2. 轻量级
3. 灵活（可以自定义开发功能）

缺点


1. 只有监控功能，无管理功能（参考横向对比）

横向对比

背景

	Calamari	VSM	InkScope	ceph-dash
	Red Hat	Intel	Orange Labs	Christian Eichelmann
Backing From	Red Hat	Intel	Orange Labs	Christian Eichelmann
Lastest Version	1.2.3	2014.12-09.1	1.1	1.0
Release Date	Sep 2014	Dec 2014	Jan 2015	Feb 2015
Capabilities	Monitor + Light Config	Monitor + Config	Monitor + Light Config	Monitor Only
Compatability	Wide	Limited	Wide	Wide


管理



MANAGEMENT

	Calamari	VSM	InkScope	ceph-dash
Deploy a Cluster	N	Y	N	N
Deploy Hosts (add/remove)	N	Y	N	N
Deploy Storage Groups	N	Y	N	N
Cluster Services (daemons)	OSD only	Y	N(?)	N
Cluster Settings (ops flags)	Y	N	Y	N
Cluster Settings (parameters)	Y	N	View	N
Cluster Settings (CRUSH)	N	Partial	View	N
Cluster Settings (EC Profiles)	N	Y	Y	N
OSD (start/stop/in/out)	Partial	Y	Y	N
Pools (Replicated)	Y (limited)	Y	Y	N
Pools (EC & Tiering)	N	Y	Partial	N
RBDs	N	Partial	N	N
S3/Swift Users/Buckets	N	N	Y	N
Link to OpenStack Nova	N	Y	N	N

监控



MONITORING

	Calamari	VSM	InkScope	ceph-dash
Mon Status	Y	Y	Y	Y
OSD Status	Y	Y	Y	Y
OSD-Host Mapping	Y	Y	Y	Y
PG Status	Y	Y	Y	Y
PG-OSD Mapping	N	N	Y	N
MDS Status	N	Y	Y	N
Host Status	Y	Y	Y	Y
Capacity Utilization	Y	via Groups	Y	Y
Throughput (Cluster)	N	Y	Y	Y
IOPS (Cluster)	Y	Y	Y	Y
Errors/Warnings	Y	Y	Y	Y
View Logs	Y	N	N	N
Send Alerts (email)	N	N	N	via nagios plug-in
Charts/Graphs	Y	N	N	via nagios plug-in

一种声音

从 ceph 社区 qq 群看过去，总会有一些运维或者开发询问哪种 ceph 管理平台方便好用，然后就开始对比 inkscope、vsm、calamari。其实

这些都不是重点，重点是看看 github 上的这些项目已经 long long ago 不更新代码了，也就是说软件的生命周期走到了尽头，没有更新和扩展。想想群里的兄弟在生产环境上用这些软件，最后是什么结果……况且大部分公司都是一两个码农在搬砖，投入到开发这三个监控平台上也不现实。

大部分生产环境都是用 cli 对 ceph 进行管理，所以生产环境对 ceph 的管理需求不大。在监控上，ustack 之前的文档提过了一套监控方案。建议关注一下 <https://prometheus.io/> 项目。前台集成 grafana，运维人员根据自己实际需求，DIY 监控面板，配合后端 exporters 很小的开发量，实现监控任意指标。报警方面 prometheus 也有自己的解决方案。联邦机制的实现可以使监控平台横向扩展。目前很多公司的生产环境都在用此方案。考虑到大规模运营，后续还需要 ELK 等工具的帮助。

<https://my.oschina.net/yangfanlinux/blog/783756>

参考资料

- ✧ http://de.slideshare.net/lnktank_Ceph/07-ceph-days-sf2015-paul-evans-static
- ✧ <http://www.hl10502.com/2017/03/30/ceph-web-manage/>
- ✧ <https://www.cnblogs.com/luxiaodai/p/10043183.html>