

Red Hat OpenStack Platform Metrics & Monitoring Service Assurance Framework

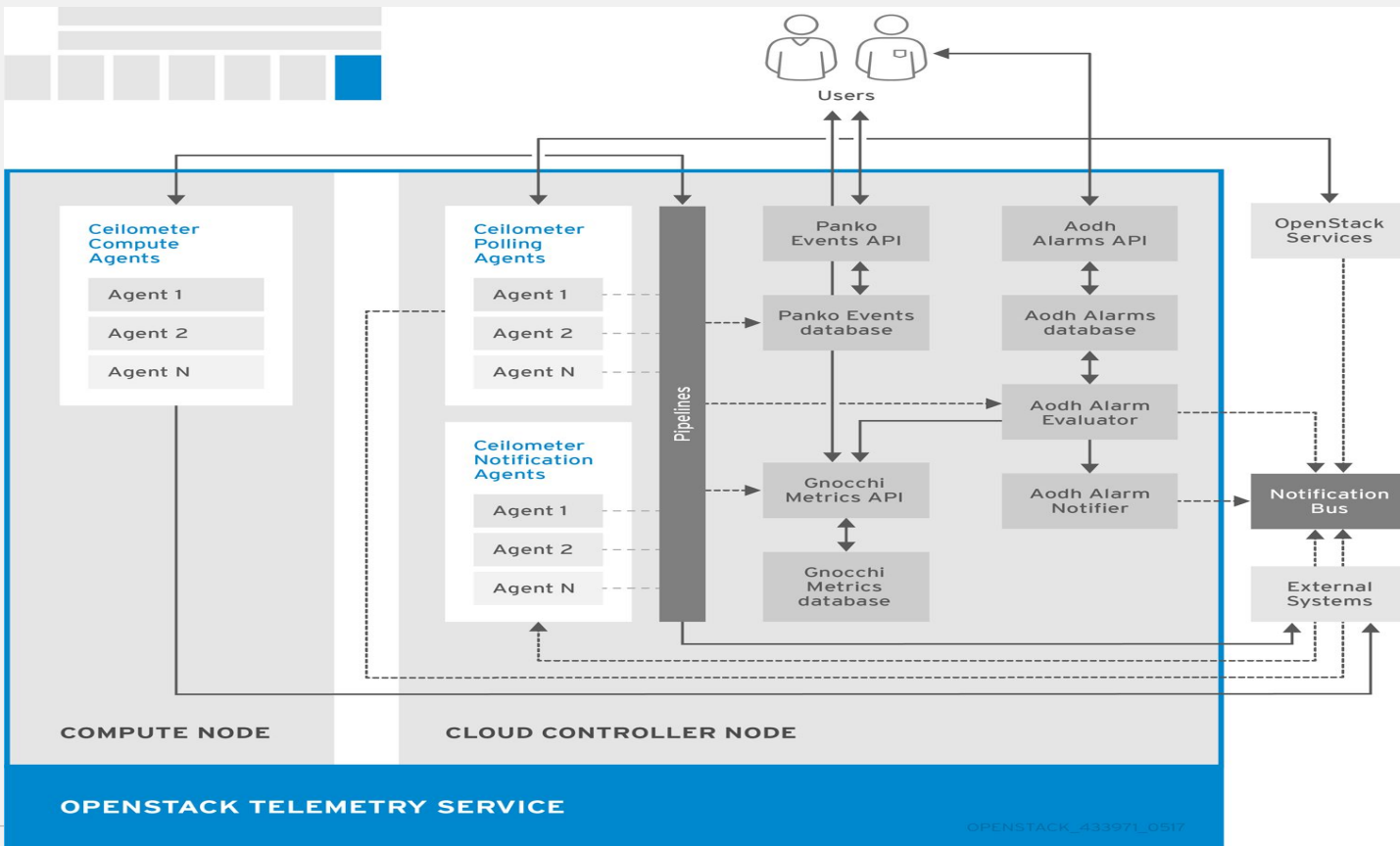
20 August 2018

Anandeep Pannu, Senior Principal Product Manager, RHOSP

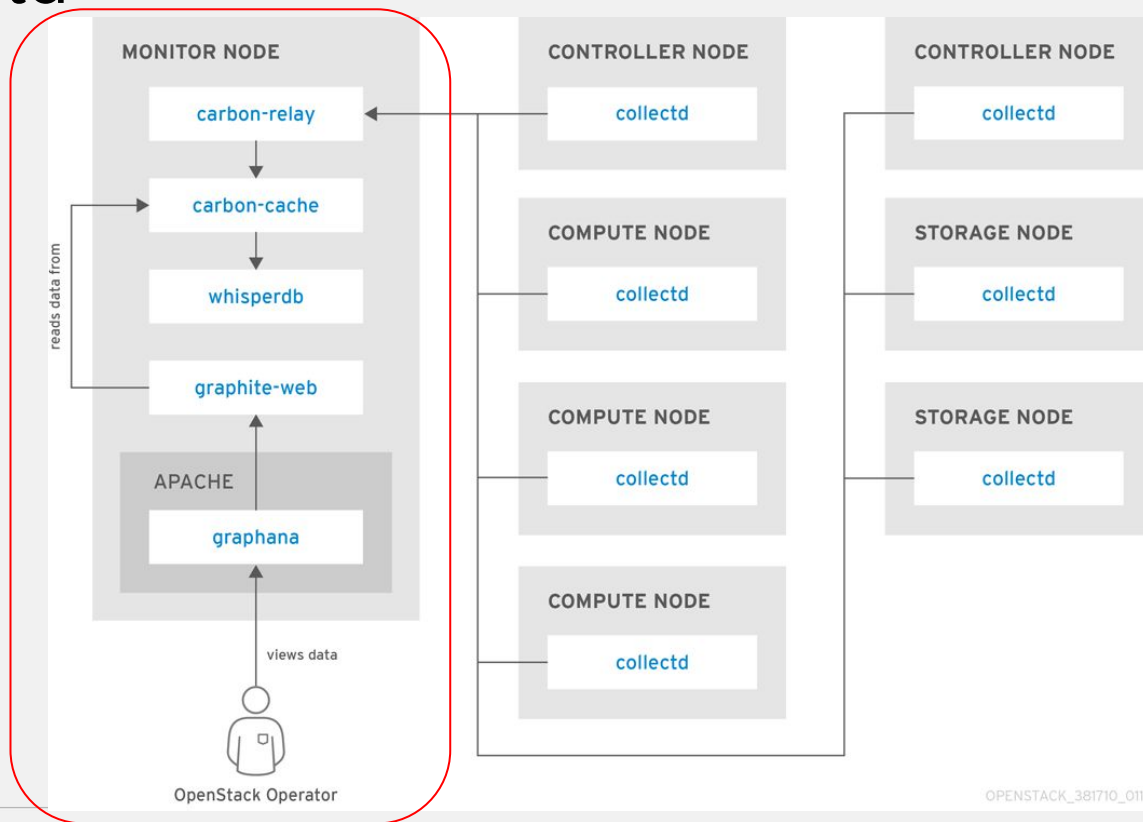
Agenda

- Current (OSP 13) Metrics & Monitoring
- RHOSP Metrics & Monitoring gaps
- Platform based approach to Metrics & Monitoring
- Architecture for metrics & events
- Proposed Metrics & Monitoring roadmap
- Q & A

Current (RHOSP 13) Metrics & Monitoring



Current (RHOSP 10+) Metrics & Monitoring Collectd



OPENSTACK_381710_0116

RHOSP 13 Collectd plug-ins

Pre-configured plug-ins:

1. Apache
 2. Ceph
 3. Cpu
 4. Df (disk file system info)
 5. Disk (disk statistics)
 6. Memory
 7. Load
 8. Interface
 9. Processes
 10. TCPConns
 11. Virt
-

NFV specific plug-ins *(to be added)*

1. OVS-events
 2. OVS-stats
 3. Hugepages
 4. Ping
 5. Connectivity
 6. Procevent
-

Current (RHOSP 10+) Metrics & Monitoring

Performance monitoring & collectd

- Collectd agents with plug-ins installed on every node
 - Infrastructure and OpenStack services can be monitored for performance metrics
 - Default series of plug-ins
 - Other plug-ins can be installed
 - Events can be captured and transmitted by collectd
- Additional plug-ins being tested and added for each release
- Collectd currently uses the HTTPWrite collector plug-in to write to http for access to metrics
- As of OSP 13, collectd metrics data is saved to Gnocchi

RHOSP Metrics & Monitoring Gaps

Ceilometer

1. Scale (number of samples per unit time X number of nodes)
 - Transport mechanism to time series database overloaded (e.g. http)
 - Ceilometer centralized polling mechanism imposes excessive loads
 - Default time series database (Gnocchi) grows in
 - Size
 - Latency
2. Platform issues for Ceilometer metrics & events
 - Ceilometer API deprecated
 - Panko API being deprecated
 - Separate Aodh alert API
 - No notification bus API
 - Gnocchi API available for time series data on metrics
 - Infrastructure monitoring is minimal
 - Monitors only OpenStack services and limited Nova information

RHOSP Metrics & Monitoring Gaps Collected

1. Scale (number of samples per unit time X number of nodes)
 - Transport mechanism to time series database overloaded (i.e. http)
 - Default time series database (Gnocchi) grows in
 - Size
 - Latency
2. Platform issues for collectd metrics & events
 - Only http access available to collectd
 - Gnocchi API available for collectd time series data on metrics
 - Server side components not packaged with RHOSP

Metrics & Events Platform based approach

1. Provide end-to-end metrics & events solution at SCALE
2. APIs for 3 levels
 - At “sensor” (collectd agent) level
 - Provide plug-ins (Kafka, AMQP1) to allow connect to collectd via message bus of choice
 - At message bus level
 - Integrated, highly available AMQ Interconnect message bus with collectd
 - Message bus client for multiple languages
 - Time series database / management cluster level
 - Prometheus Operator



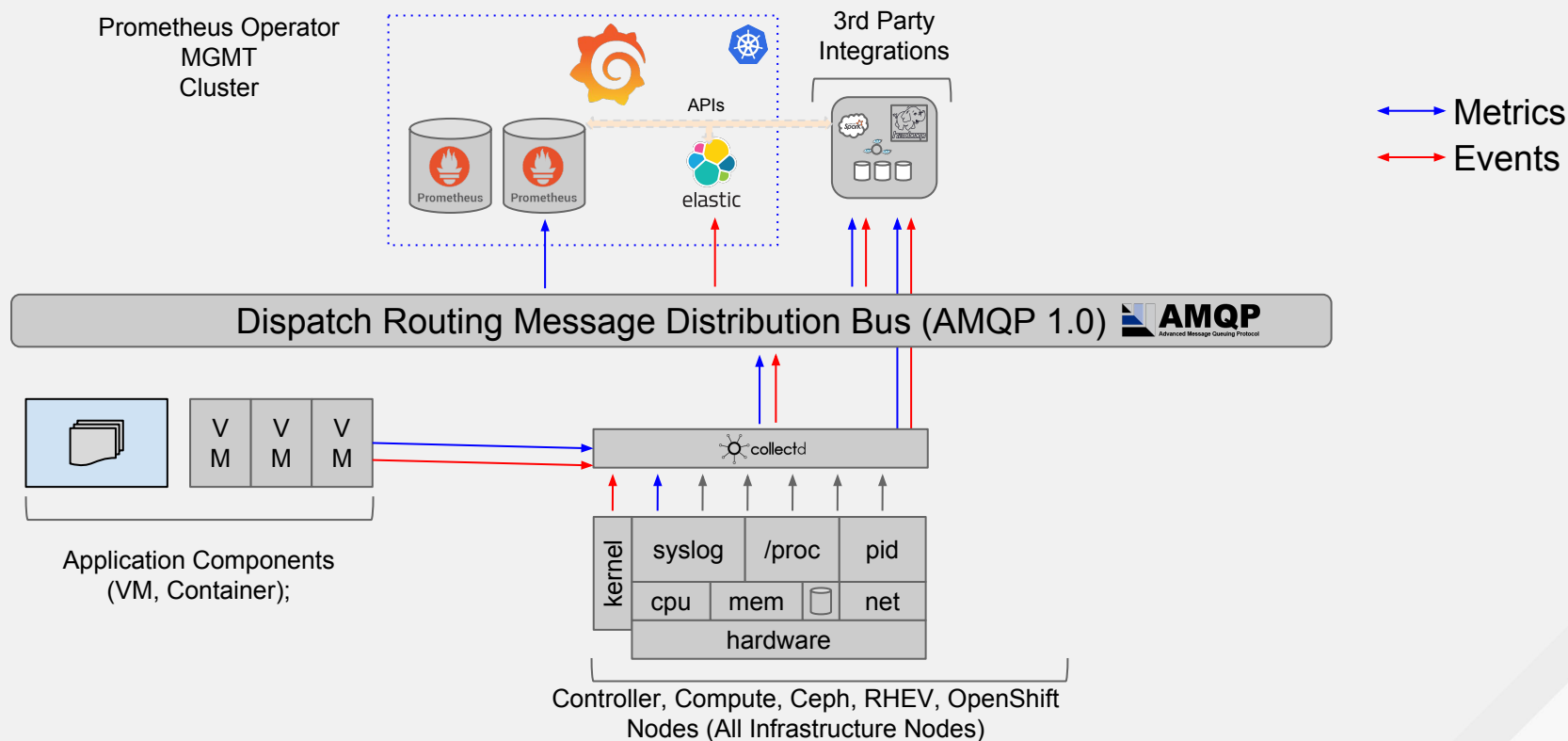
Architecture for infrastructure metrics & events

Architecture for infrastructure metrics & events

Based on the following elements

1. Collectd plug-ins for infrastructure & OpenStack services monitoring
2. AMQ Interconnect direct routing (QDR) message bus
3. Prometheus Operator database/management cluster
4. Ceilometer for chargeback/capacity planning

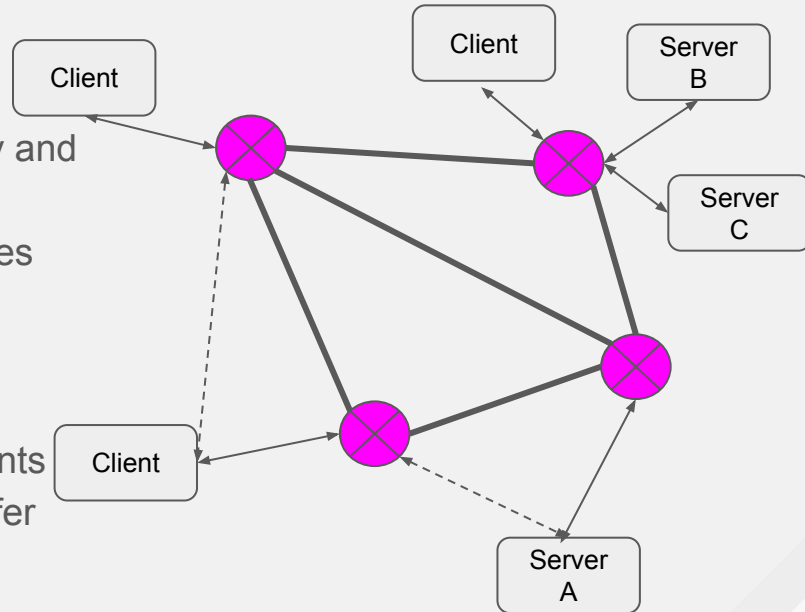
Architecture for infrastructure metrics & events



Architecture for infrastructure metrics & events

AMQ 7 Interconnect - Native AMQP 1.0 Message Router

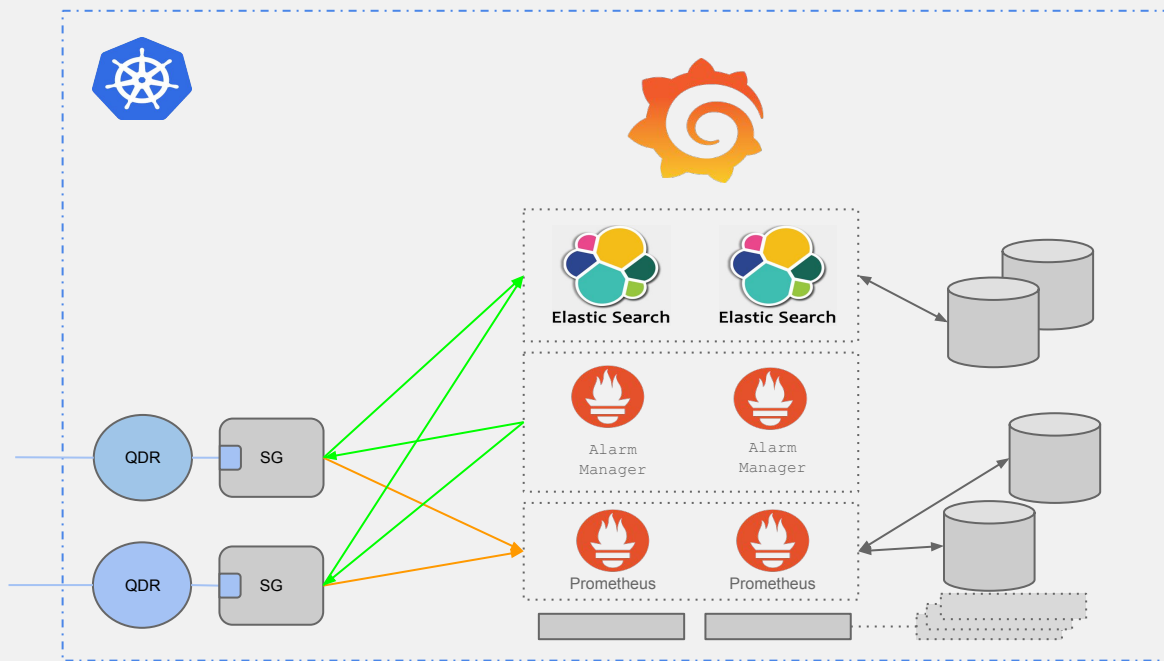
- Large Scale Message Networks
 - Offers shortest path (least cost) message routing
 - Used without broker
 - **High Availability** through redundant path topology and re-route (not clustering)
 - Automatic recovery from network partitioning failures
 - **Reliable delivery without requiring storage**
- QDR Router Functionality
 - Apache Qpid Dispatch Router QDR
 - Dynamically learn addresses of messaging endpoints
 - Stateless - no message queuing, end-to-end transfer



High Throughput, Low Latency
Low Operational Costs

Architecture for infrastructure metrics & events

Prometheus Operator & AMQ QDR clustered



Architecture for infrastructure metrics & events

Additional OpenStack components

1. AMQP1 collectd plug-in : Proton (“send” side) client for AMQ integrated
2. Proton send/receive client for connecting to AMQ for consumers
3. Collectd plug-ins from Barometer project integrated
4. Prometheus Management cluster running on OpenShift
 - At least two servers (for HA)
 - Each server has one QDR (Qpid Dispatch Router)
 - Prometheus Operator which consists of
 - i. Prometheus
 - ii. Prometheus Config
 1. Multiple Prometheus for HA (one per server)
 - iii. Prometheus Alert Manager (one per server)
 - Elasticsearch for logs
5. Director installation & configuration of all additional OpenStack components
6. Perf & Scale testing for integrated component

Proposed Metrics & Monitoring roadmap

1. Early access (limited availability) based on 13 (TBD)
2. OSP 14 TP (End 2018)
3. OSP 13 TP backport (Post OSP 14 TP)
4. OSP 15 GA (Mid 2019)

Q & A?



plus.google.com/+RedHat



linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHatNews