



Transaction Traceability Across ONAP Components

James MacNider

ONAP Beijing Release Developer Forum
December 11, 2017

Logging Enhancements Project Value

Problem Space:

- ONAP is a complex, distributed solution with many components that cooperate to execute workflows
- All these components produce a LOT of log data in several different formats
- When a use case isn't working, it's not always obvious which components were involved, or where to look first when an error happens.
 - This will become more of an issue in Beijing when many components will have more robust redundancy models
- Very difficult to observe and report on without manual effort

Solution:

The Logging Enhancements Project seeks to address these problems in two ways:

- Driving the institution and adoption of guidelines to support consistency, traceability, and machine-readability of logging across ONAP
- Integration with an industry-proven framework for log aggregation, persistence, analytics, and search (the Elastic stack)

The Elastic Stack



Beats

Ships data from the applications to Logstash and ElasticSearch



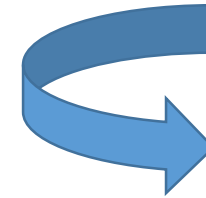
Logstash

Dynamic data collection pipeline. Consumes / parses data and inserts it into ElasticSearch



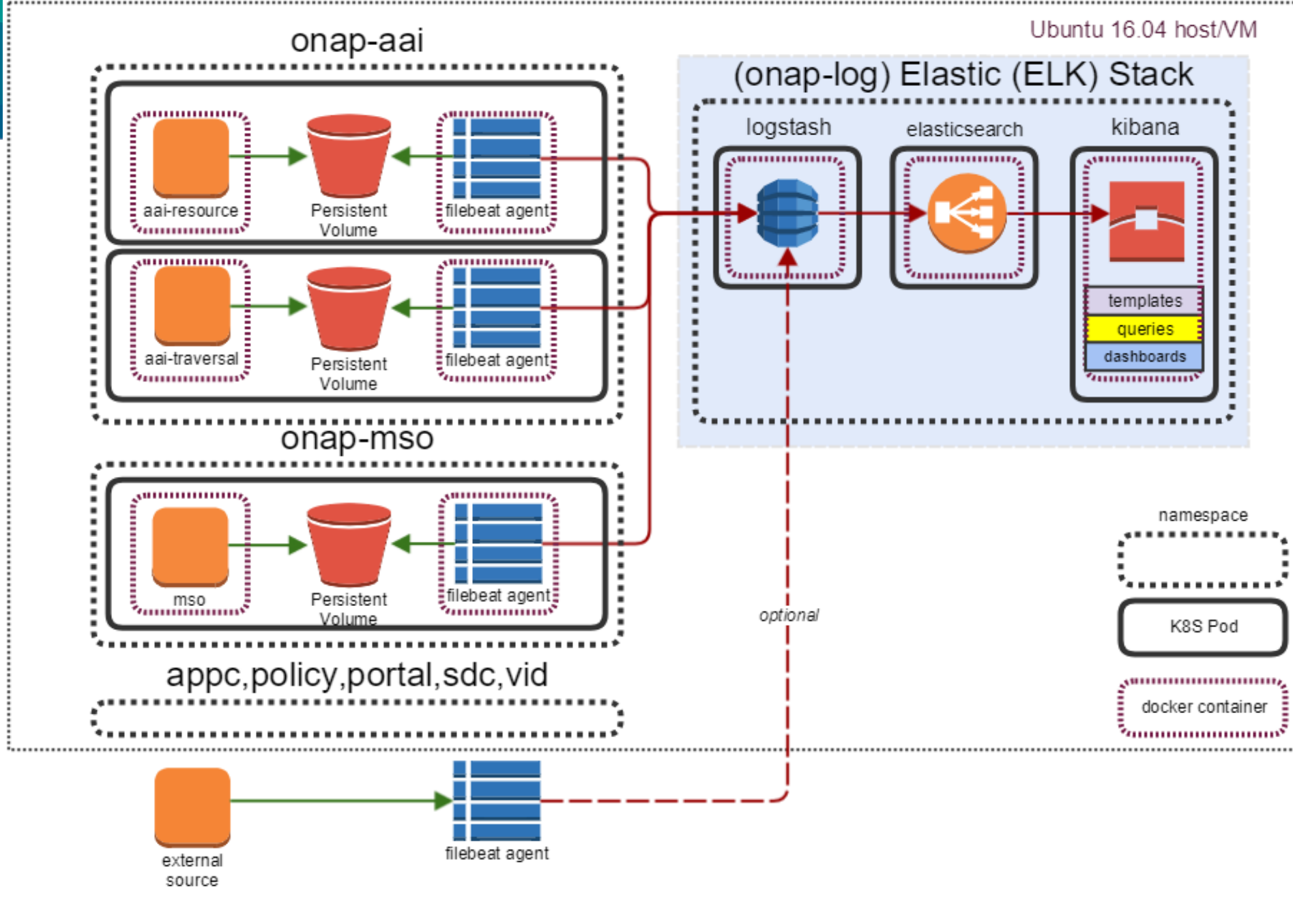
Elastic Search

Distributed JSON-based search and Analytics engine



Kibana

Data visualization front-end driven by data in ElasticSearch



Transaction Traceability Across ONAP Components

What's a "Transaction" in this context?

- Any set of operations set in motion by an event that ONAP reacts to.
For example:
 - Ops personnel initiating distribution of a Service
 - DCAE reacting to an event reported by a VNF

What makes a transaction traceable?

- As per the logging guidelines, when a transaction is initiated, a new Request ID (also called transaction ID) is generated as a UUID
- Components pass the Request ID in the "X-TransactionID" REST header for all interactions that are part of completing the transaction
- Downstream components that receive the Request ID are in turn obligated to re-use it and propagate it

Let's see it in action!

ONAP Logging Guidelines

- Components following a shared set of logging guidelines are what make it possible for tools like the Elastic Stack to work well.
- Initial draft of the version 1.2 Logging Guidelines are on the ONAP wiki [now](#).
 - They're still a work in progress and the team wants **you** be an active contributor and reviewer
 - No components conform to this version of the guidelines *yet*. Some work will be required to align.
 - Config changes or minor code changes

Current Gaps and Challenges

- Guideline needs to be completed, reviewed, and accepted by the community. Early adoption in Beijing will make things easier for us all
- Not all components are shipping logs to Elastic Stack yet
- Requests to components not using REST, such as over DMaaP, do not propagate RequestIDs
- Logging config is duplicated currently in OOM, often out of sync with components
 - Consolidated configuration initiative in OOM should address this
- Guideline is Java-centric
- Scalability and security for the Elastic Stack
- Devel-time debugging workflow

Logging Resources

- Project Homepage:
<https://wiki.onap.org/display/DW/Logging+Enhancements+Project>
- Logging Guidelines (1.2 Draft):
<https://wiki.onap.org/pages/viewpage.action?pageId=20087036>
- Logging Weekly call: Wednesdays 11AM Eastern (8AM Pacific)
<https://zoom.us/j/519971638>



ONAP

OPEN NETWORK AUTOMATION PLATFORM

Thank You