Alarm Correlation

OUCE 2018

David Smith – Software Developer @ OpenNMS

openNMS®

A walk through of the new Alarm Correlation feature

- What is Correlation?
- What is a Situation?
  - Could be a New Alarm or an Existing Alarm
  - Our engines create new Alarms
- Benefits of Alarm Correlation:
  - Reduces noise for Operators monitoring the network
  - Prioritizes the workflow

How is it done?

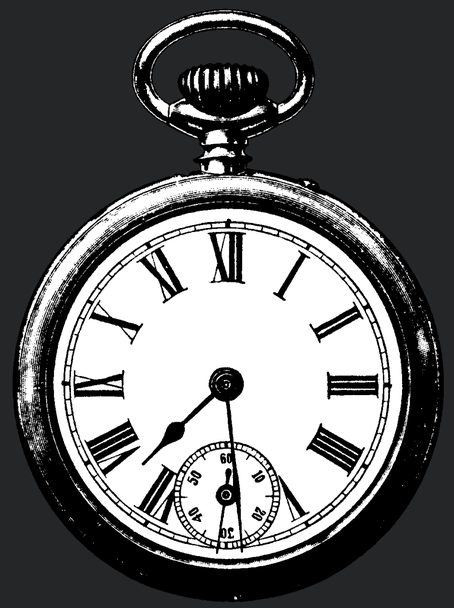
- Correlation Engine processing Alarms
  - Input: Stream of Alarms
  - Output: New Alarms (Situations) with Alarms in Buckets

Correlation Engine is an Interface

- Multiple implementations
- Simple interface:
  - init(Alarms, ExistingSituations, Inventory)
  - tick(timestamp)
  - registerSituationHandler(SituationHandler)
- Design your own if you wish....
- Or use existing...

Temporal Engine

- Simple Sliding Window
- Shows how Correlation can be done (the “Hello World” of correlation)
- Surprisingly accurate



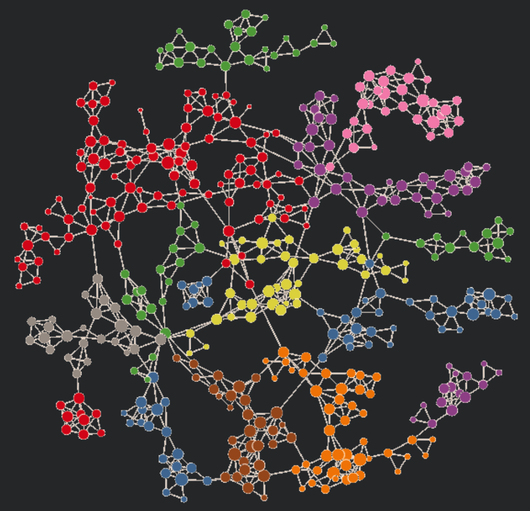
Topology (Rules Based) Engine

- Uses Network Topology
- Uses JBoss Drools Rules Engine
- Requires topology to be very accurate and constantly updated
- Requires very specific rules to cover Situations – can become unwieldy

```
// RULE #2
rule "CardDown"
  when
    $group : Group(owner.type == "Card", numberServiceAffecting == numberMembers, serviceAffectingTrend == CountTrend.INCREASING)
    not ModelObject(type == "Card", id == $group.getOwner().getId(), operationalState == OperationalState.SA)
  then
    actionMgr.log("RULE #2");
    actionMgr.synthesizeAlarm($group.getOwner(), OperationalState.SA, Severity.MAJOR, $group.getId());
end

```

Cluster (Graph Based) Engine



- Currently used engine
- It's graph based – each Alarm is attached to a Vertex
- Uses unsupervised ML: DBScan algo
- Correlation is calculated based on the distance on the graph
  - Measured in both space and time
- Space is the distance between network topology objects
- Network topology is calculated as alarms are received and the Topology Elements are extracted from the alarm

Walking through Correlation

- What happens...?
- Fire an Alarm
- Fire another Alarm
- Correlate the two
  - Engine applies logic to determine if any of the alarms are related

Viewing Situations

- HELM display of Situations

Alarms

Last 6 months

NOT in Situation

UEI	Log Message	Node Label	Count	Last Event Time
uei.opennms.org/provisioner/provisioningAdapterFailed	A provisioning adapter failed for host.	192.168.72.155	1	2018-09-17 11:22:53
uei.opennms.org/provisioner/provisioningAdapterFailed	A provisioning adapter failed for host.	192.168.72.155	1	2018-09-17 11:22:53
uei.opennms.org/nodes/interfaceDown	Interface 172.20.50.108 is down.	localhost	1	2018-09-17 20:28:02

Situations

Log Message	Situation Alarm Count	Affected Node Count
A problem has been triggered on localhost/0.0.0.0/FEEDBACK_F.	3	1
A problem has been triggered on localhost/0.0.0.0/ALARM_F.	3	1

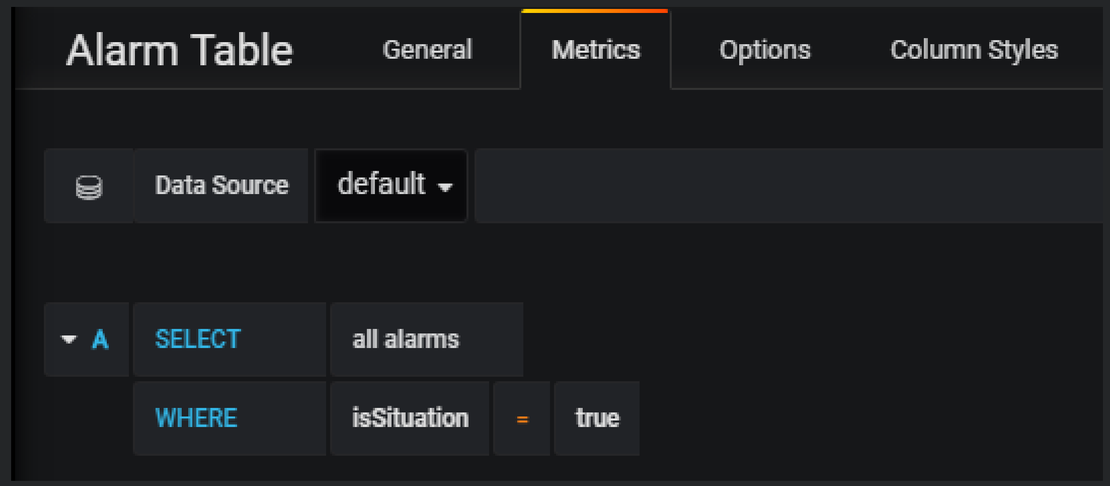
Alarms and Situations

UEI	Log Message	Node Label	Is Situation
uei.opennms.org/provisioner/provisioningAdapterFailed	A provisioning adapter failed for host.	192.168.72.155	N
uei.opennms.org/provisioner/provisioningAdapterFailed	A provisioning adapter failed for host.	192.168.72.155	N
uei.opennms.org/nodes/interfaceDown	Interface 172.20.50.108 is down.	localhost	N
uei.opennms.org/alarms/trigger	A problem has been triggered on localhost/0.0.0.0/FEEDBACK_B.	locally	N
uei.opennms.org/alarms/trigger	A problem has been triggered on localhost/0.0.0.0/FEEDBACK_F.	locally	Y

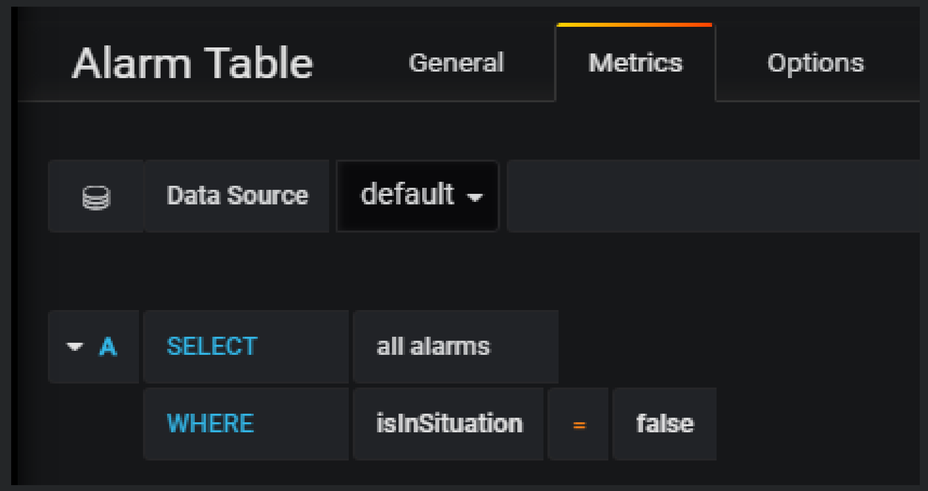
1 2 3

Viewing Situations – HELM Filtering

- Alarm tables can filter Situations
  - IsSituation – true or false

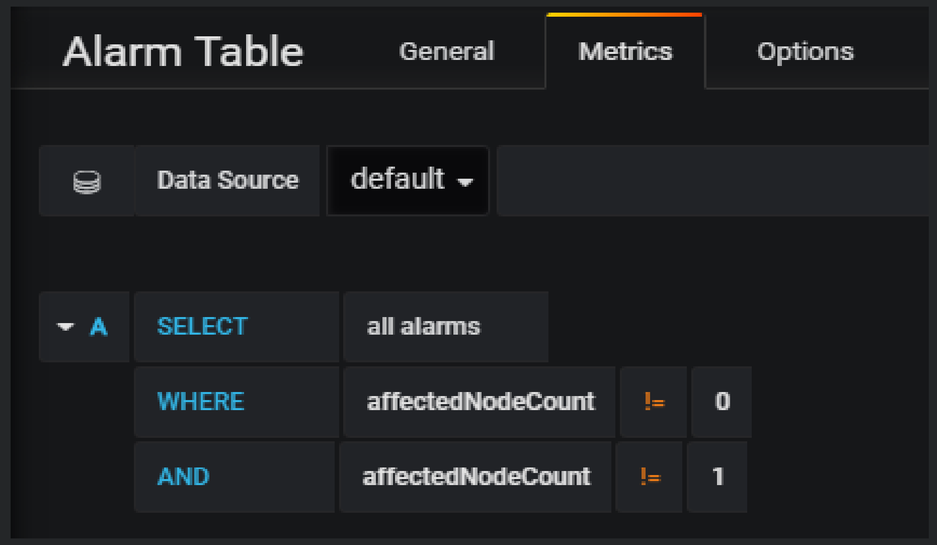


- Filter on IsInSituation == false
  - All alarms not correlated to a Situation

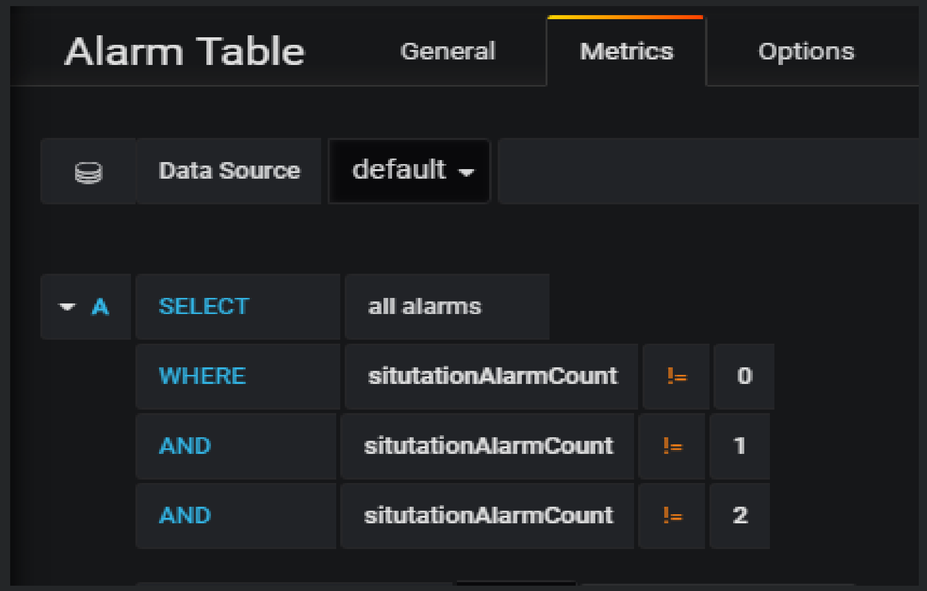


Viewing Situations – HELM Filtering (cont'd)

- Filter Situation attributes. E.G.
  - AffectedNodeCount > 1



  - AlarmCount > 2



Viewing Situations – HELM Filtering (cont'd)

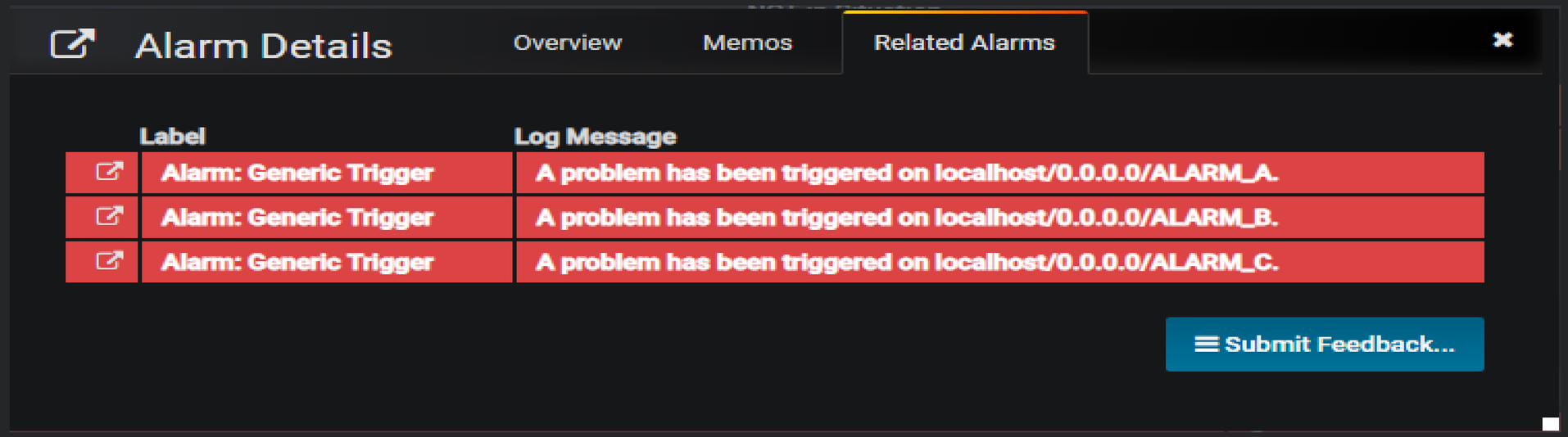
- Situation Table
  - Columns defined to show **AlarmCount** and **AffectedNodeCount**

Situations

Log Message	Situation Alarm Count	Affected Node Count
A problem has been triggered on localhost/0.0.0.0/FEEDBACK_F.	3	1
A problem has been triggered on localhost/0.0.0.0/ALARM_F.	3	1

Viewing Situations – Alarm Details

- Display a list of Alarms that the Situation Correlates
- New Tab for Related Alarms
- Links to Alarm details
- Situation Feedback



Situation Feedback

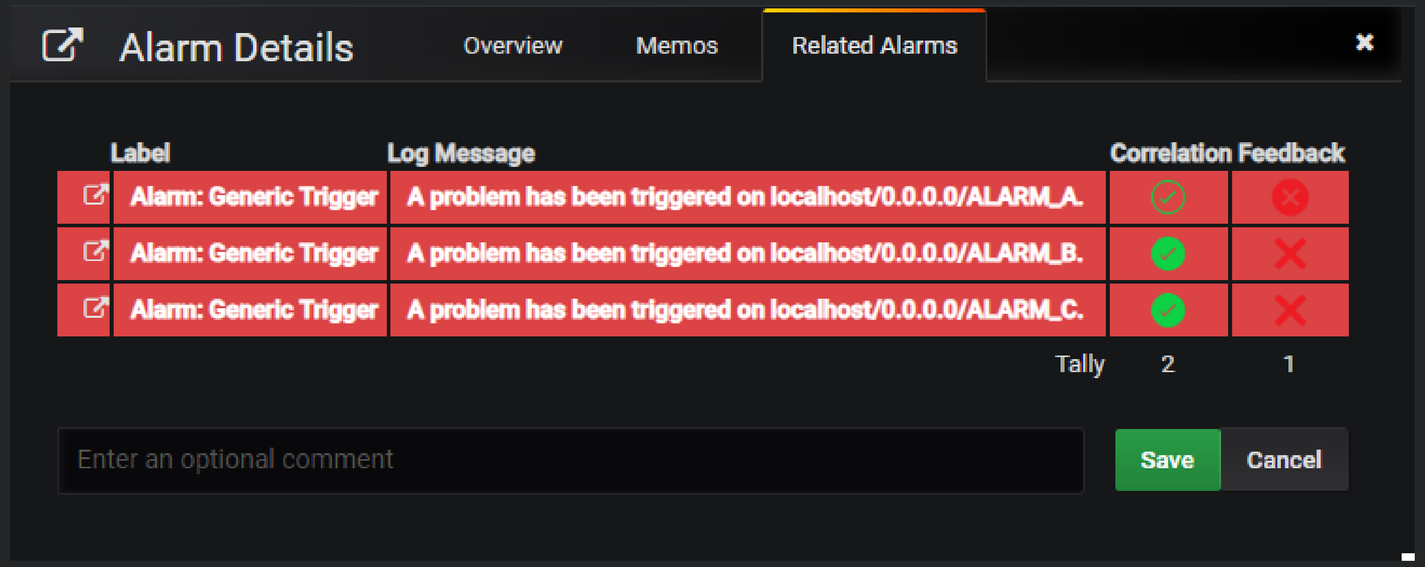
- What it is and how does it works
- Allows training (we'll come back to this)
- Allows for removing Alarms from a Situation (via HELM and ReST)
- Allows for adding Alarms to a Situation (via ReST)

Viewing Situations – Situation Feedback

- Indicates if Feedback has ever previously been submitted for this Situation

- Can denote False Positives
  - They will be removed from the correlation

- Feedback is then persisted in ElasticSearch

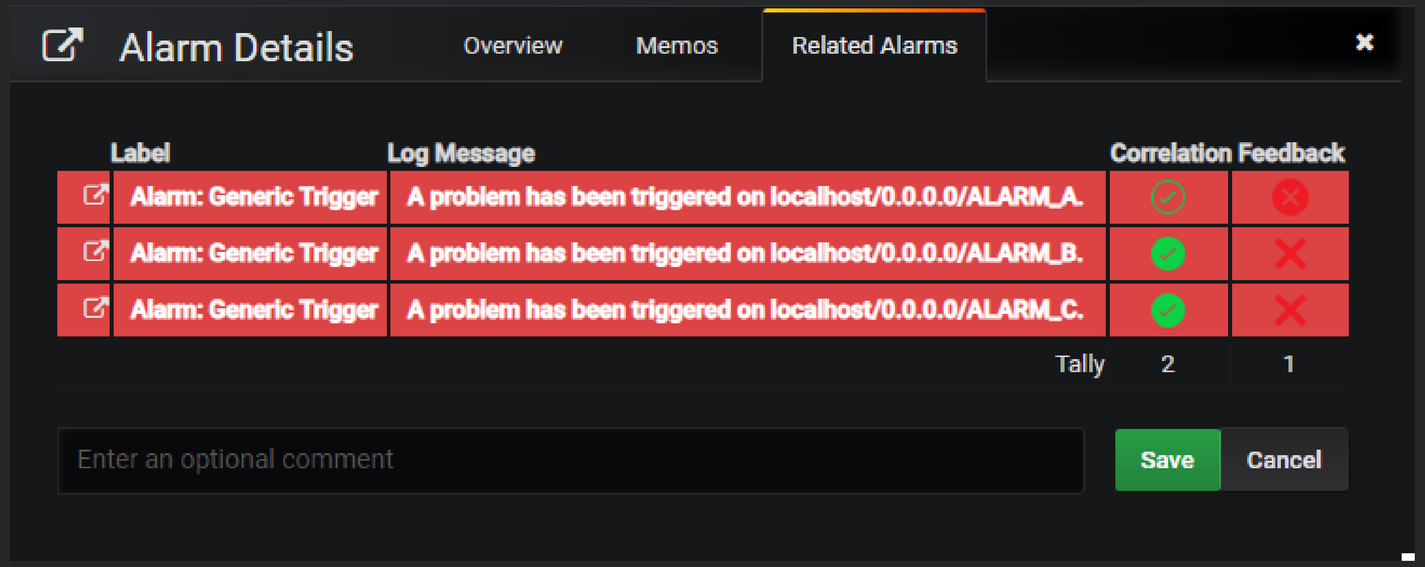


Viewing Situations – Situation Feedback

- Indicates if Feedback has ever previously been submitted for this Situation

- Can denote False Positives
  - They will be removed from the correlation

- Feedback is then persisted in ElasticSearch



Deployment

- Components
  - Driver
  - Engine
  - Datasource
    - Datasource requires using Karaf as a bus for Alarms and Situations
    - Download Kafka Docker image: spotify/kafka
    - Use Kafka Producer feature
      - Enable
      - configure
    - Use Sink Api
      - Enable Listening to Events API

Installation

- Clone the OpenNMS/oce Git repo
- Build source: mvn install
- Install the features: datasource, engine, and driver
- Karaf Shell – using the one with OpenNMS…
  - Access via SSH admin@localhost -p 8101
  - Use your OpenNMS 'admin' credentials
  - Datasource (TODO – does Datasource work with local deploy???)
  - *feature:install oce-datasource-opennms oce-engine-cluster oce-processor-standalone oce-driver-main*

Further Reading:

- Coming up! (WIP) DOCS:
  - Admin Guide
  - Developer Guide (Situation Feedback ReST API)
  - Available: https://docs.opennms.org/opennms/branches/develop/index.html

  - Helm Guide
  - Available: https://docs.opennms.org/helm/branches/master/helm/latest/welcome/index.html
- GitHub:
  - https://github.com/OpenNMS/oce
- Wiki:
  - https://wiki.opennms.org/wiki/DevProjects/Sextant

Thank you.

Questions?

smith@opennms.com