HP Helion rackspace HOSTING IBM

Paris OpenStack® Summit

This is your cloud

Monasca Deep Dive

Monitoring-as-a-Service (at Scale)

Roland Hochmuth, Sandy Walsh, Tong Li/ November 5, 2014

Agenda

Problem Statement

What is Monasca?

- Architecture
- Metrics
- Events/StackTach.v3
- Anomaly Detection

Current Status

Performance

Next Steps

Demo

Q&A

Problem Statement

Monitoring-as-a-Service: Lacking multi-tenant model

Performance, scalability and data retention

Multiple uses of the data: SLA calculations, business analytics, RCA, …

Elasticity and dynamic run-time configurability

- Metrics and Alarm management
- Spammy alerts and alert fatigue.

Real-time event stream processing

Extensibility: Integrate with other systems via API or internally

Multiple Systems: Internal/operational monitoring and external/customer-facing monitoring are separate systems. Health/Status different from metrics.

Cryptic data: Force fit metric/event names results in an impedance mismatch

What is Monasca?

Monitoring-as-a-Service solution based on a first-class REST API

- Multi-tenancy based on Keystone authentication. Supports self-service.

Highly-performant, scalable, fault-tolerant and capable of big data retention

Metrics storage/retrieval/statistics and alarm/thresholding engine

Notification system

Real-time event stream processing

Open-source and built-on open-source technologies such as:

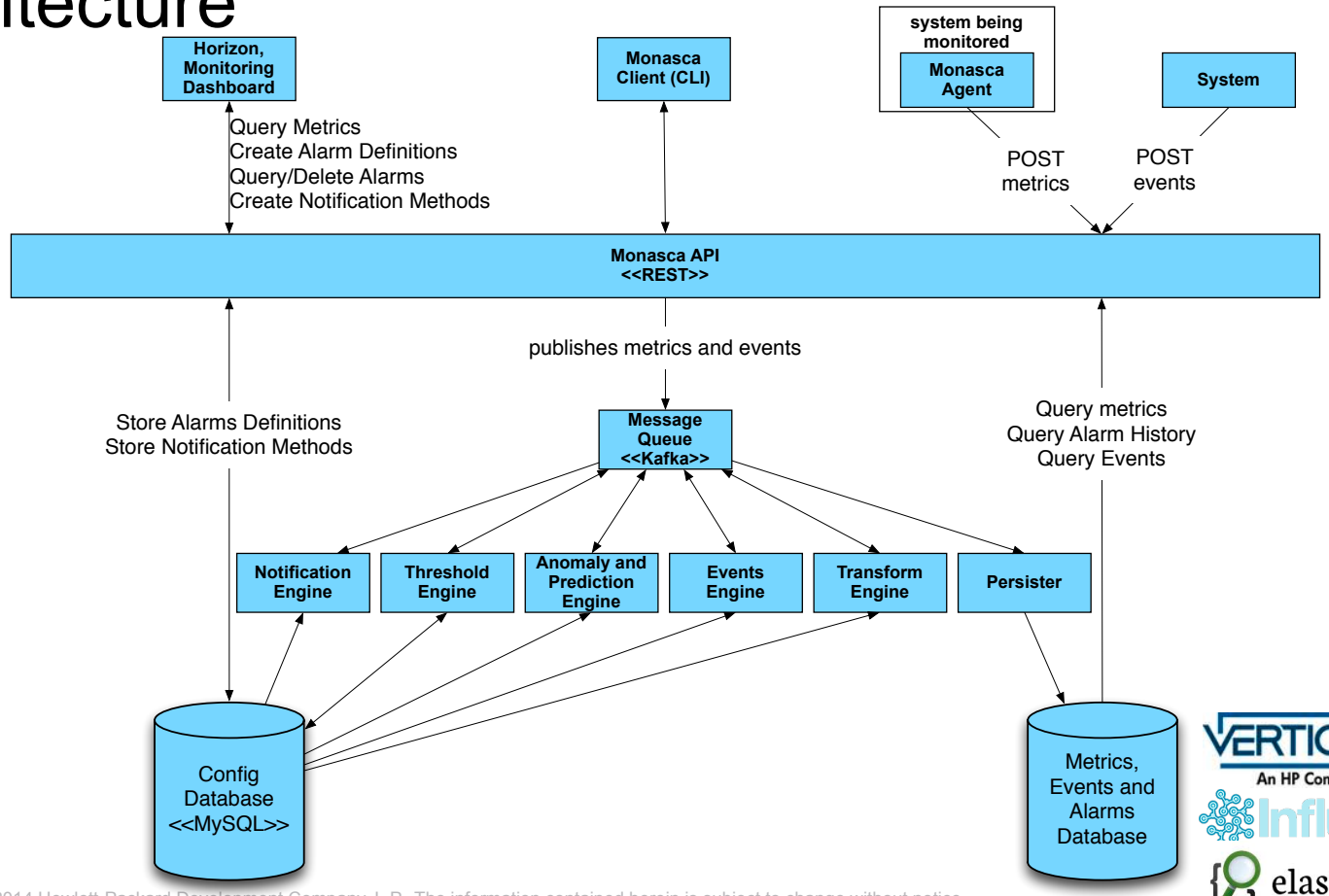
- Kafka: Performant, scalable, fault-tolerant, durable message queue. Used by LinkedIn, Twitter, …
- Apache Storm:
- Time-series databases: InfluxDB supported today. Elastic-search in progress.

Consolidates multiple monitoring systems into a single solution

- Used for both operational and customer facing monitoring.

Extensible based on micro-services message bus architecture

Architecture



Metrics

REST API

Metrics: Create, query and get statistics for metrics

```
{
        name: cpu.user_perc,
        dimensions: {
                hostname: hostname.domain.com,
                region: uswest,
                zone: 1,
                service: compute
        }
}
```

Simple, concise beautiful flexible description
Name (string)
Dimensions: Dictionary of arbitrary (key, value) pairs

Alarm Definitions

- Alarm definitions are templates that are used to automatically create alarms based on matching metric names and dimensions
- Simple compound expression grammar: avg(cpu.user_perc{}) > 85 or avg(disk_read_ops{device=vda}, 120) > 1000
- Actions associated with alarms for state transitions to ALARM, OK and UNDETERMINED
- Severity (LOW, MEDIUM, HIGH, CRITICAL).

Alarms: Query and Delete alarms and query alarm state history

Notification Methods: e.g. Email address. Associated with alarm definitions

Monasca Agent

Python monitoring agent

System metrics (cpu, memory, …)

Service metrics

- RabbitMQ, MySQL, Kafka, and many others

Application metrics

- Built-in statsd daemon
- Python Monasca Statsd library

VM metrics

Active checks

- HTTP status checks and response times
- System up/down checks (ping and ssh)

Runs any Nagios plugin

Extensible/Pluggable: Additional services can be easily added

UI

Horizon Dashboard

- Overview/Top-level drill-down
- Create/Read/Update/Delete alarm definitions using an expression builder
- Read/Delete alarms and alarm history
- Create/Read/Update/Delete notification methods

Grafana Dashboard (http://grafana.org/)

- Provides visualization of metrics

Events

Anomaly Detection

Monasca Anomaly Engine implements real-time streaming anomaly detection

Two algorithms:

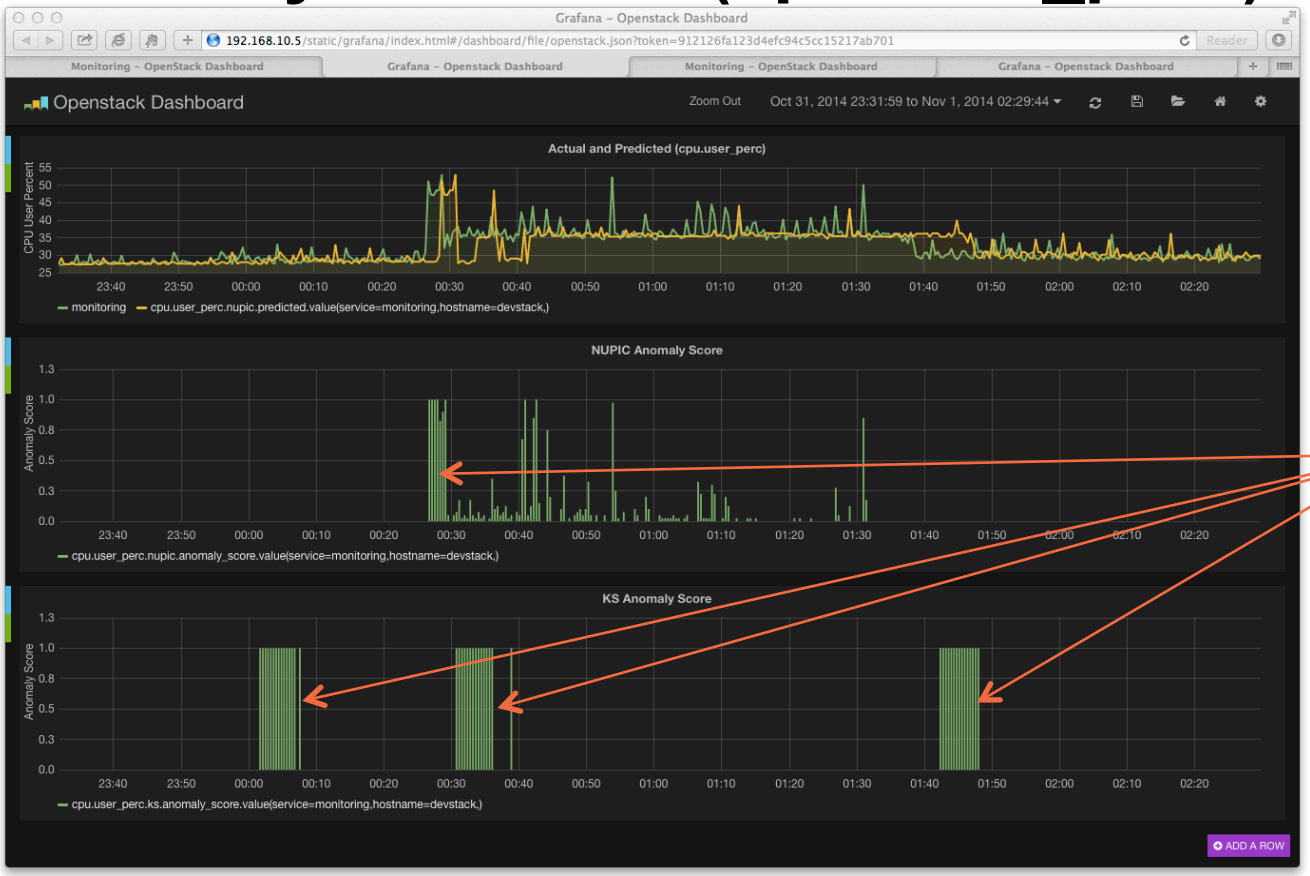
- Numenta Platform for Intelligent Computing (NuPIC) used by GROK
  - An open-source Python/C++ implementation of Hierarchical Temporal Memory
- Kolmogorov-Smirnov (K-S) Two Sample Test

Anomaly Engine

- Consumes metrics from the Kafka metrics topic
- Calculates predicted value and anomaly score (probability of an anomaly)
- Publishes calculated values as metrics to the Kafka metrics topic

Alarms can be created for Anomaly scores

Anomaly Detection (cpu.user_perc)



Anomalies detected

Current Status

Monasca and Stacktach.v3 is open-sourced in StackForge

Not an OpenStack incubated project, but we are targeting incubation

Metrics, Alarm Definitions, Alarms and Notification Methods completely supported/functional and ready for production deployment

Who is working on it?

- HP
- RackSpace
- IBM

Who is deploying it?

- HP: Public Cloud and Helion distribution
- Time Warner Cable (TWC)
- Workday

Performance (Metrics inserts/sec)

Test Deployment (HP R&D Cluster):

- Three HP Proliant SL390s G7 servers
- InfluxDB cluster

Performance:

- Total end-to-end performance including storage in InfluxDB: Approximately 25K to 30K metrics/second.
- monasca-api: > 50K metrics/sec per single API server.
- monasca-api > 150K metrics/sec for a three node cluster with a load-balancing VIP.

If you need more database performance?

- VERTICA An HP Company is supported. Scales to hundreds of thousands of metrics per second.

Next Steps

Events/StackTach.v3 integration is in progress

Anomaly detection is in progress

Formalize micro-services architecture

- Define message formats
- Define how services are published and registered

Python port is in progress:

- All components Python except for API and Threshold Engine
  - API is 75% ported to Python. Note, Java API is 100% functional
  - Threshold Engine is the only remaining Java component

Call to Action

Looking for contributors

- Monasca Service, StackTach.v3, Events, Anomaly Detection
- Monasca Agent: Help extend with additional services. E.g. Sensor data
- Help Integrate, Deploy, Test and Performance benchmarking

More info:

- Launchpad: https://launchpad.net/monasca
- Wiki: https://wiki.openstack.org/wiki/Monasca
- IRC: #openstack-monasca

Monasca development environment:

- monasca-vagrant (https://github.com/stackforge/monasca-vagrant): A turn-key development environment that installs Monasca and a Devstack VM
- Newly upgraded to use Ansible

Demo

HP Helion
rackspace HOSTING
IBM

hp

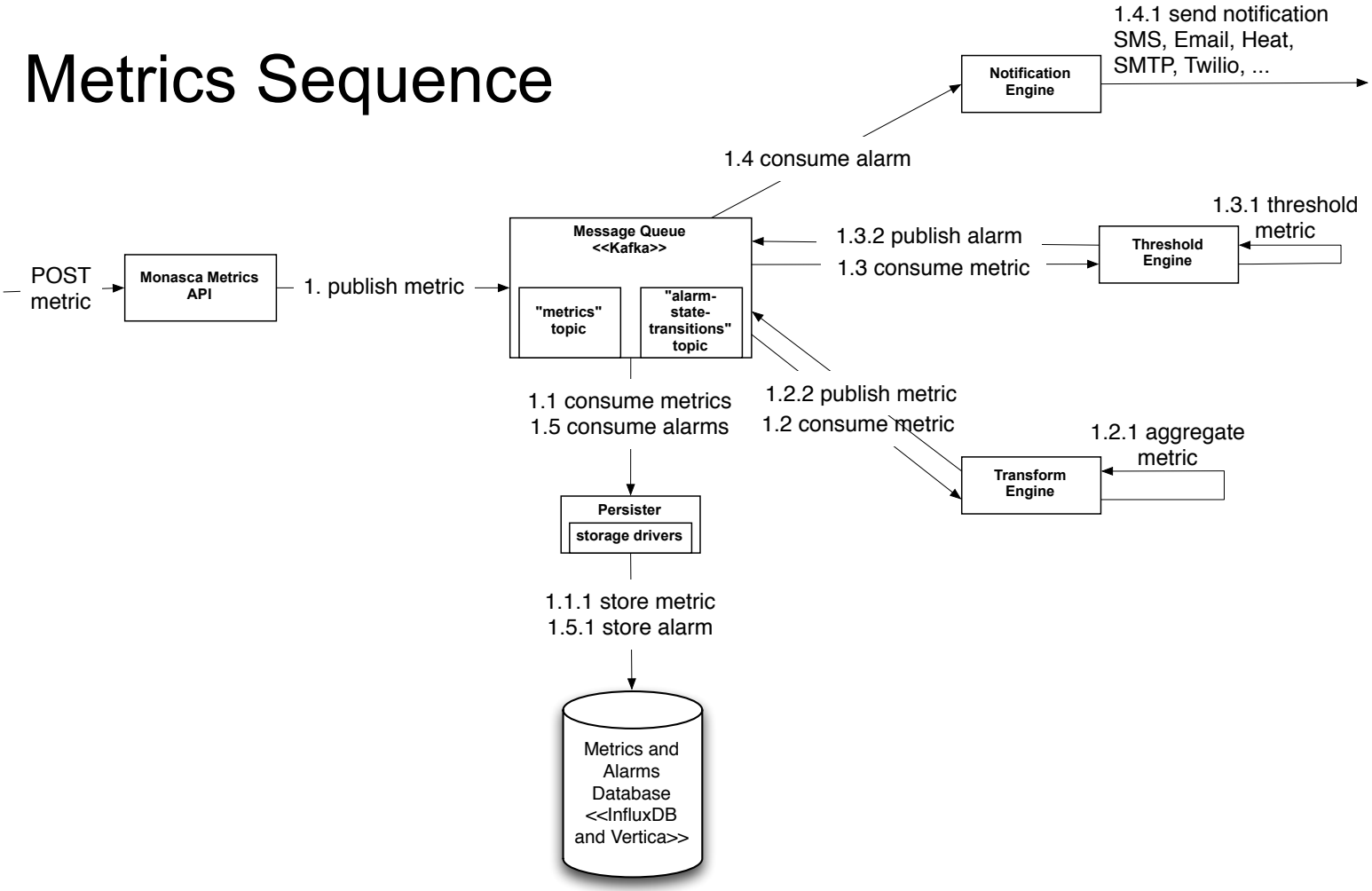
Demo Recap

What did we just show?

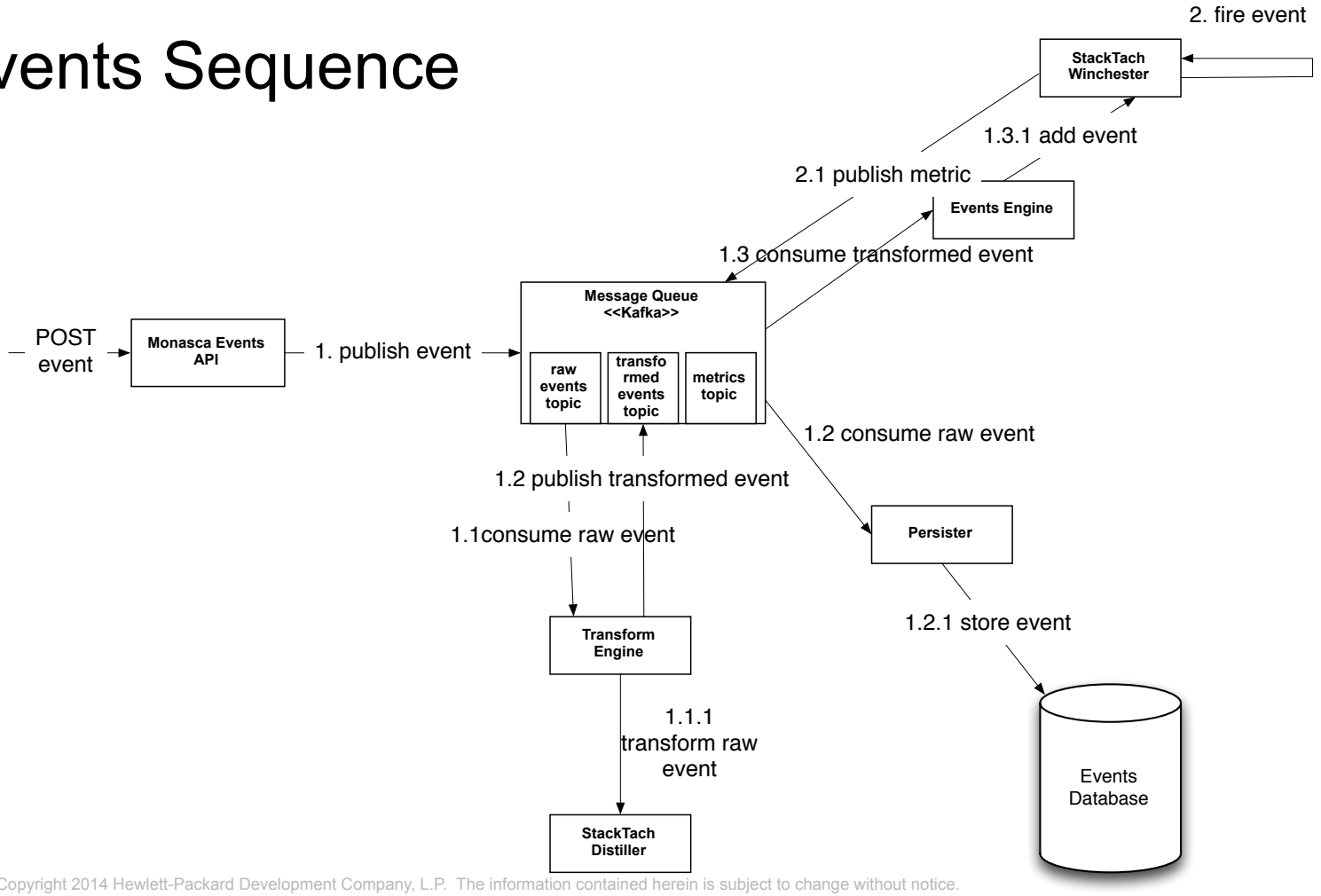
1. **An OpenStack Notification is sent to the Monasca Events API**
2. **The API publishes the notification to the Kafka raw events topic**
3. **The Transform Engine consumes, transforms (using StackTach Distiller) and publishes the event to the transformed event topic**
4. **The Events Engine consumes, adds to the StackTach Winchester Pipeline.**
5. **If the notification is a “compute.instance.create.end” event the Winchester pipeline handler fires, calculates the delta, and publishes to the Kafka metrics topic**
6. **The metric can then be alarmed on in the metrics pipeline or visualized**

Thank you

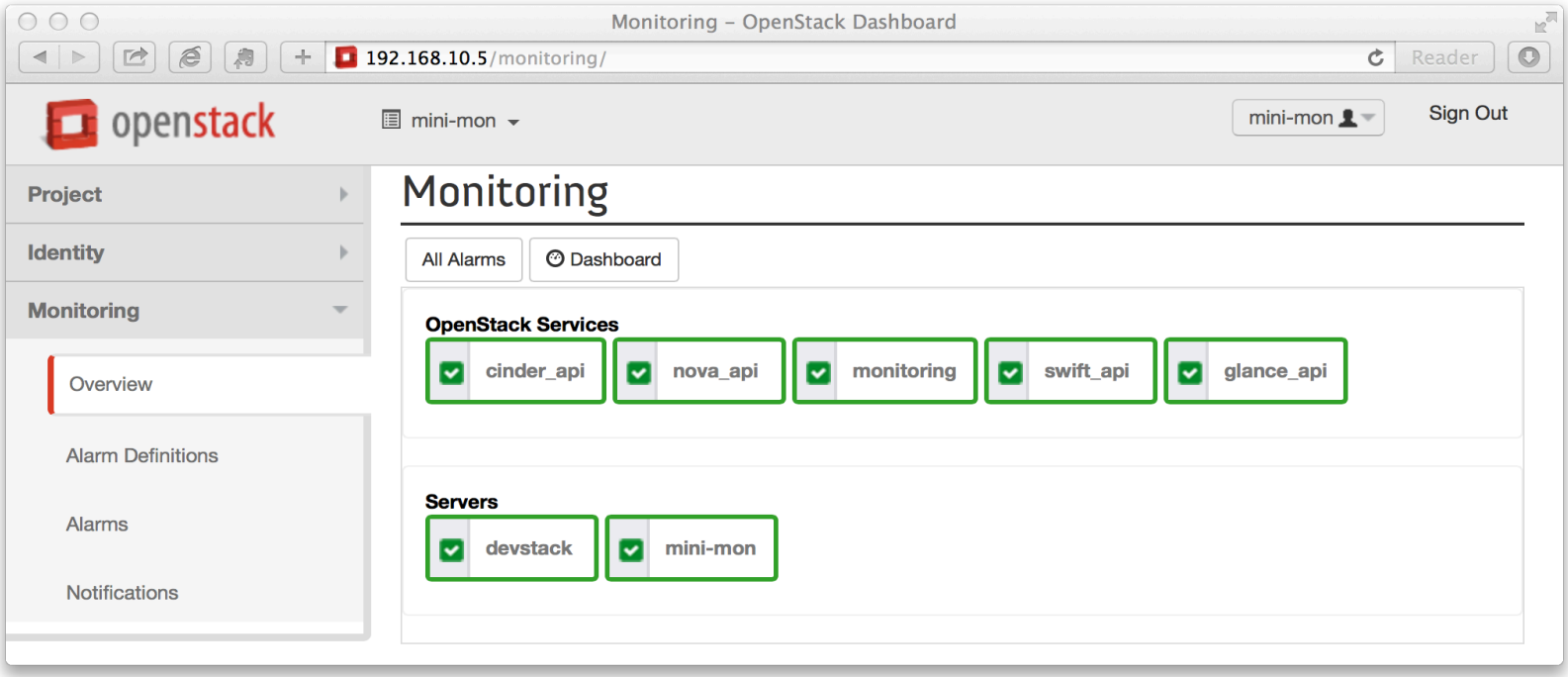
Metrics Sequence



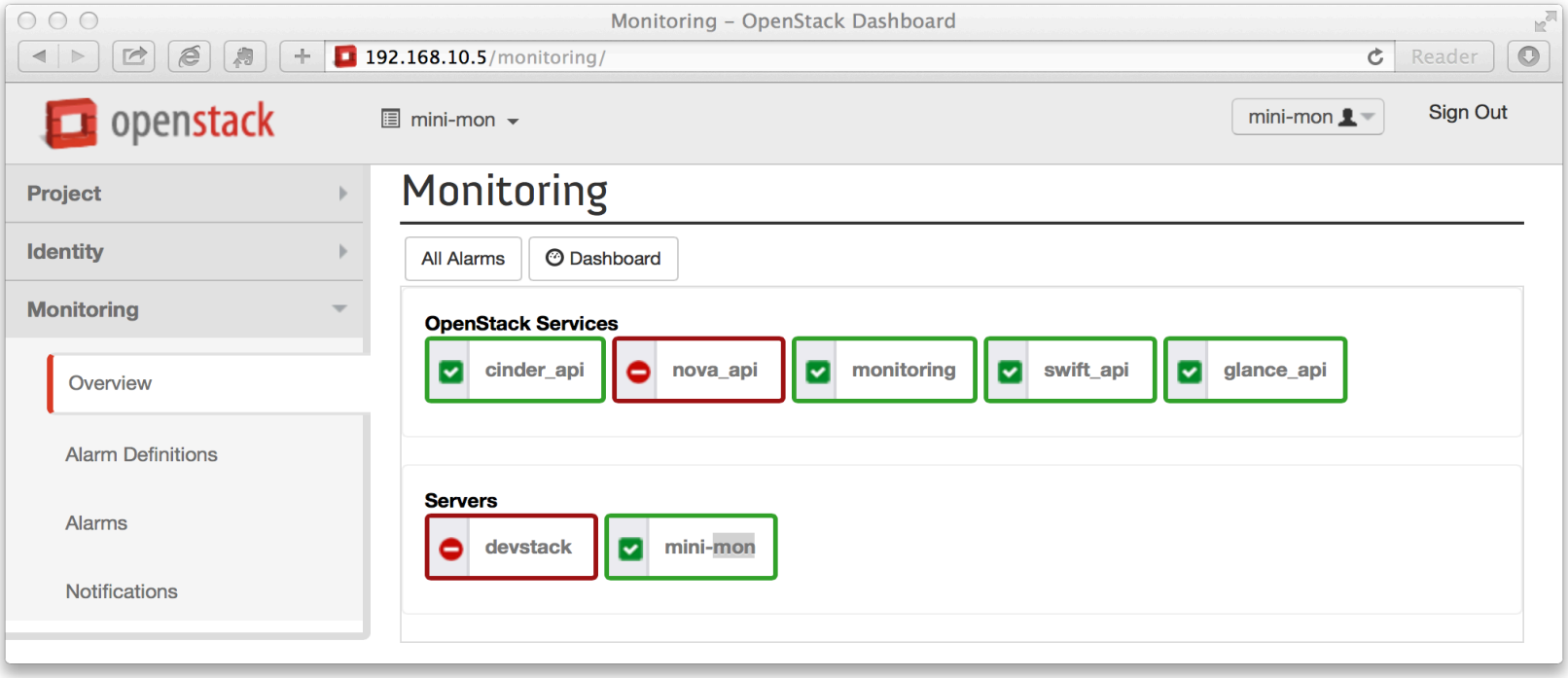
Events Sequence



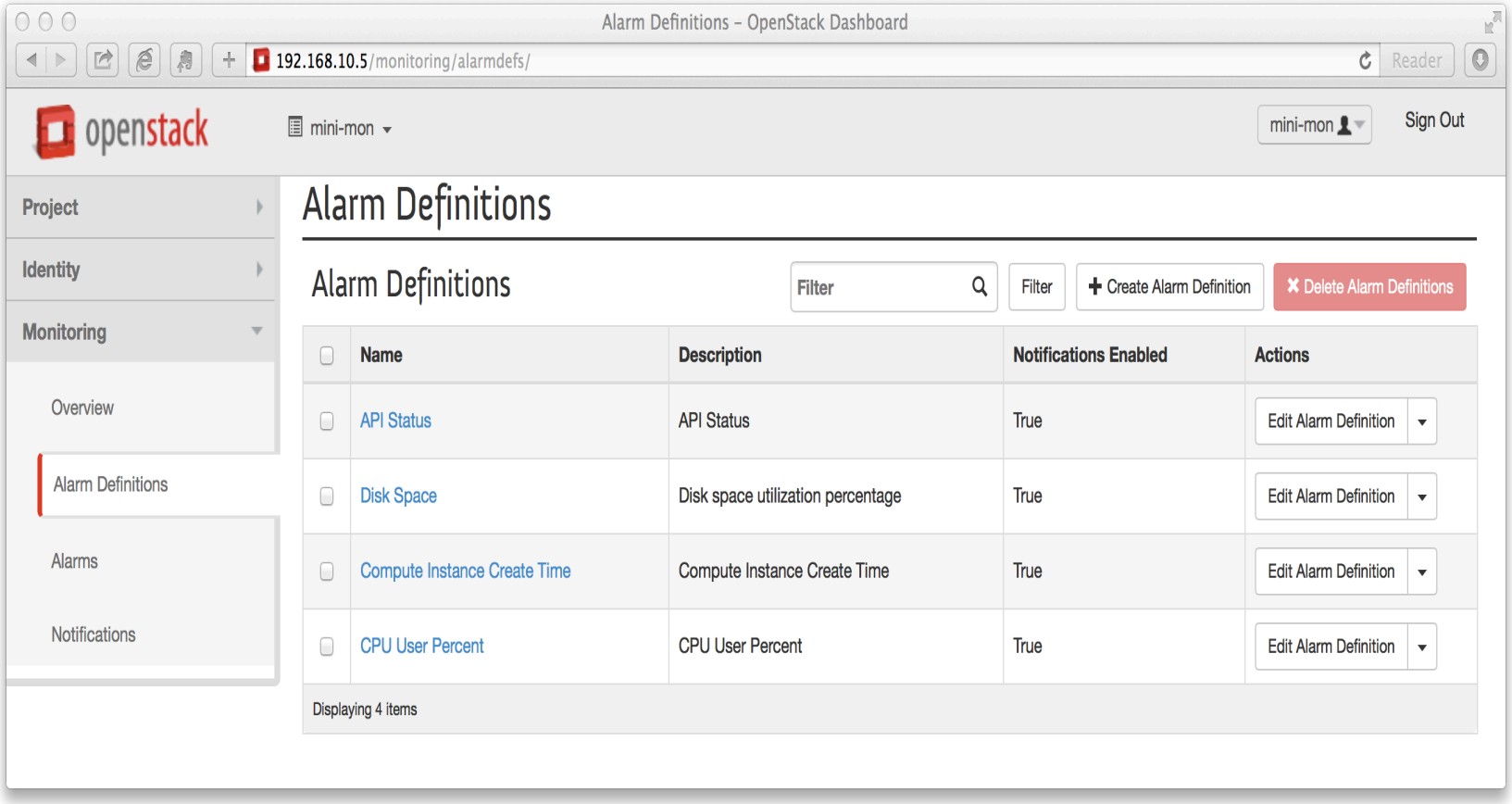
Monitoring Overview (All Services Healthy)



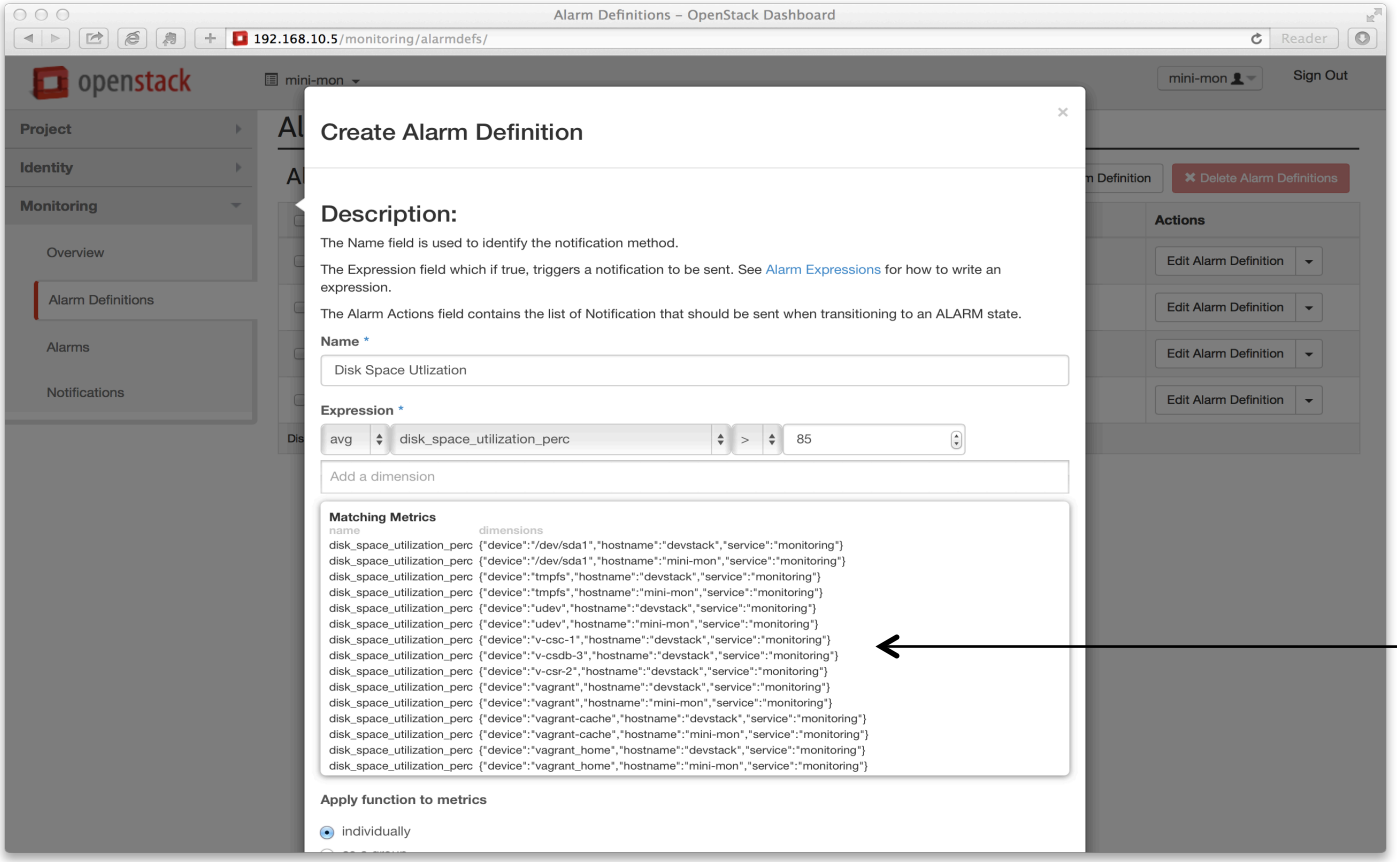
Monitoring Overview (nova-api down)



List Alarm Definitions

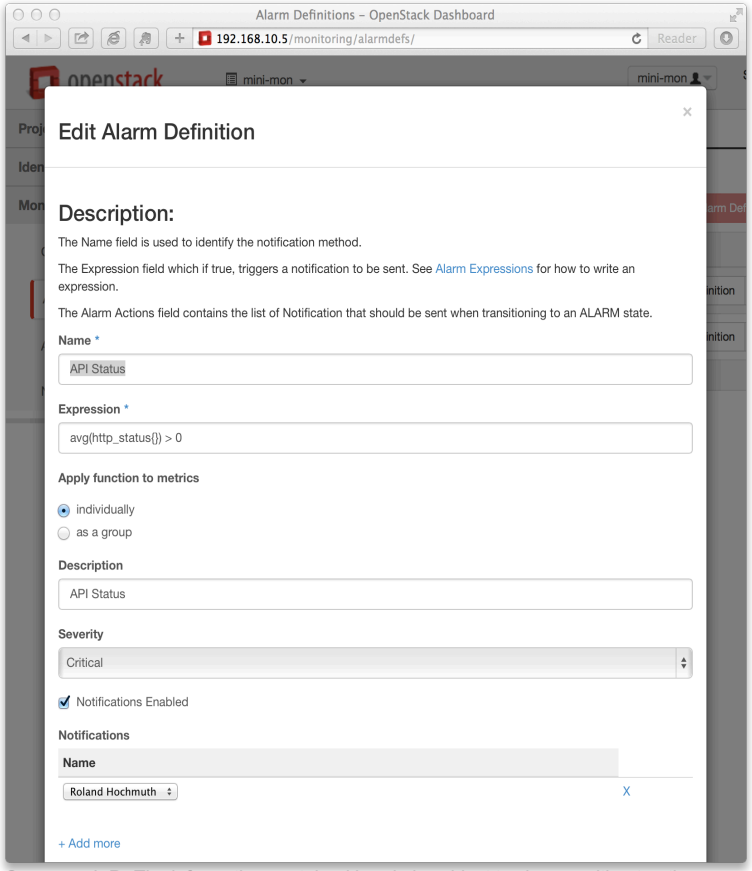


Create Alarm Definition (Disk Space)

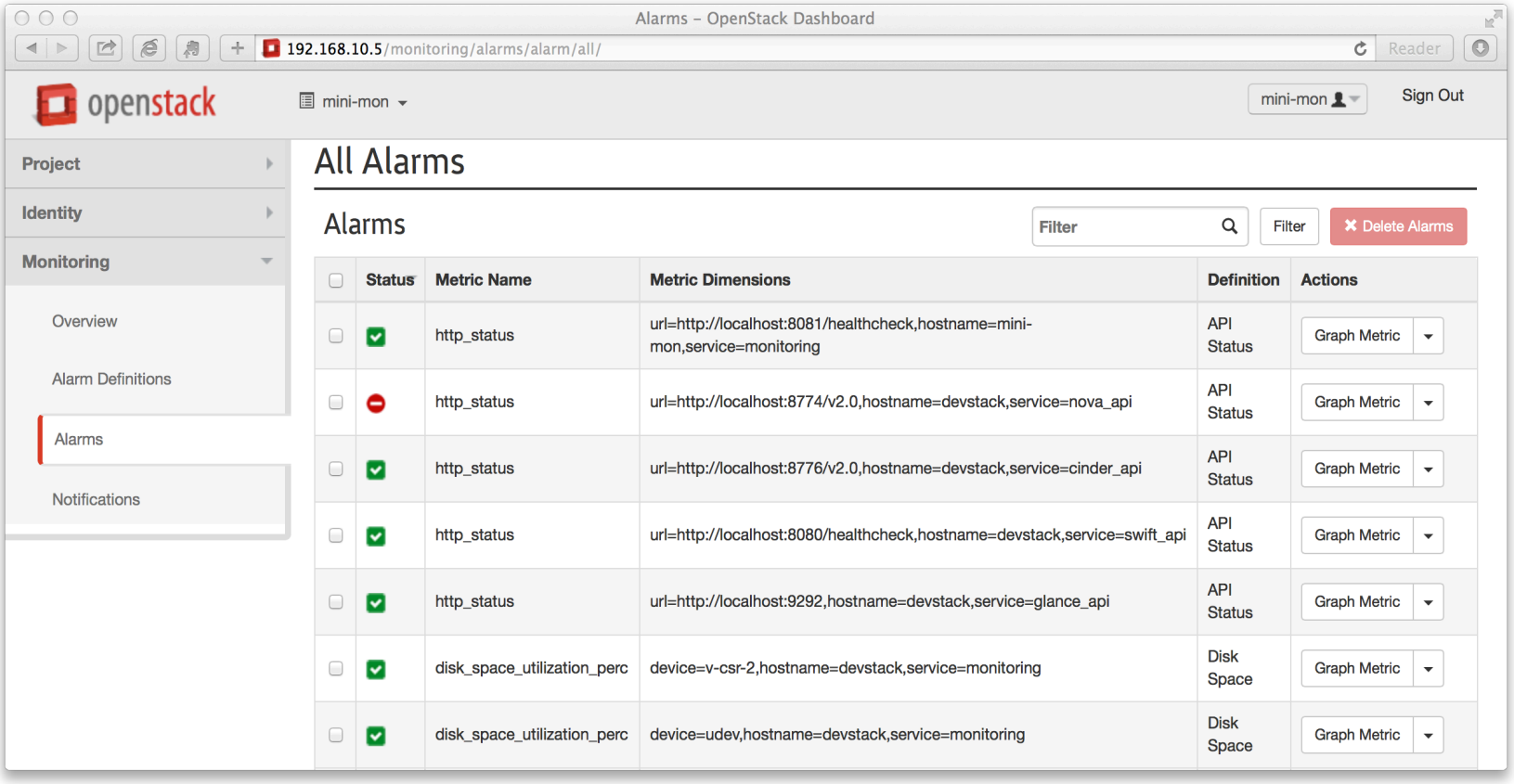


Matching metrics

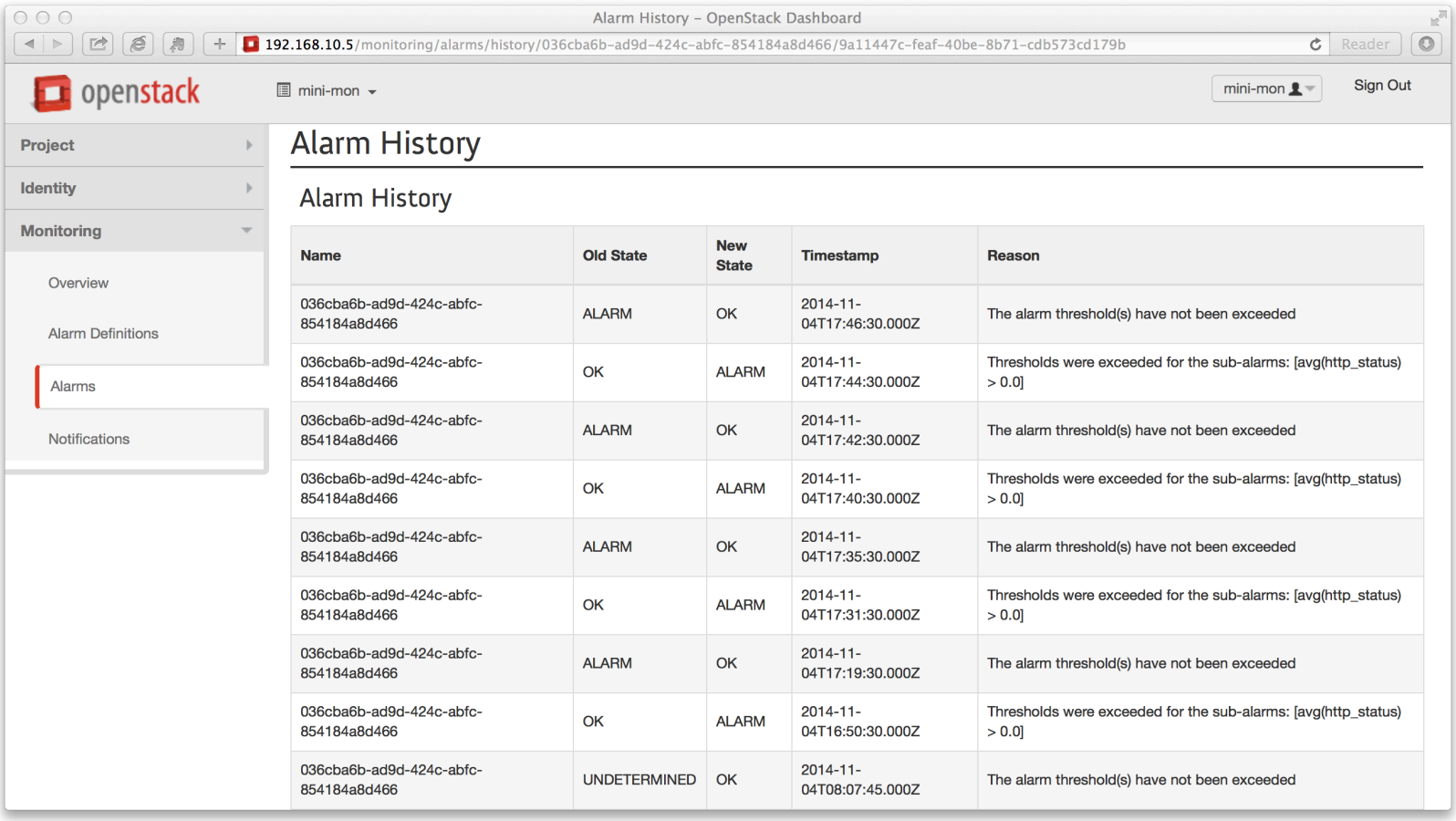
Edit Alarm Definition (API Status)



List All Alarms

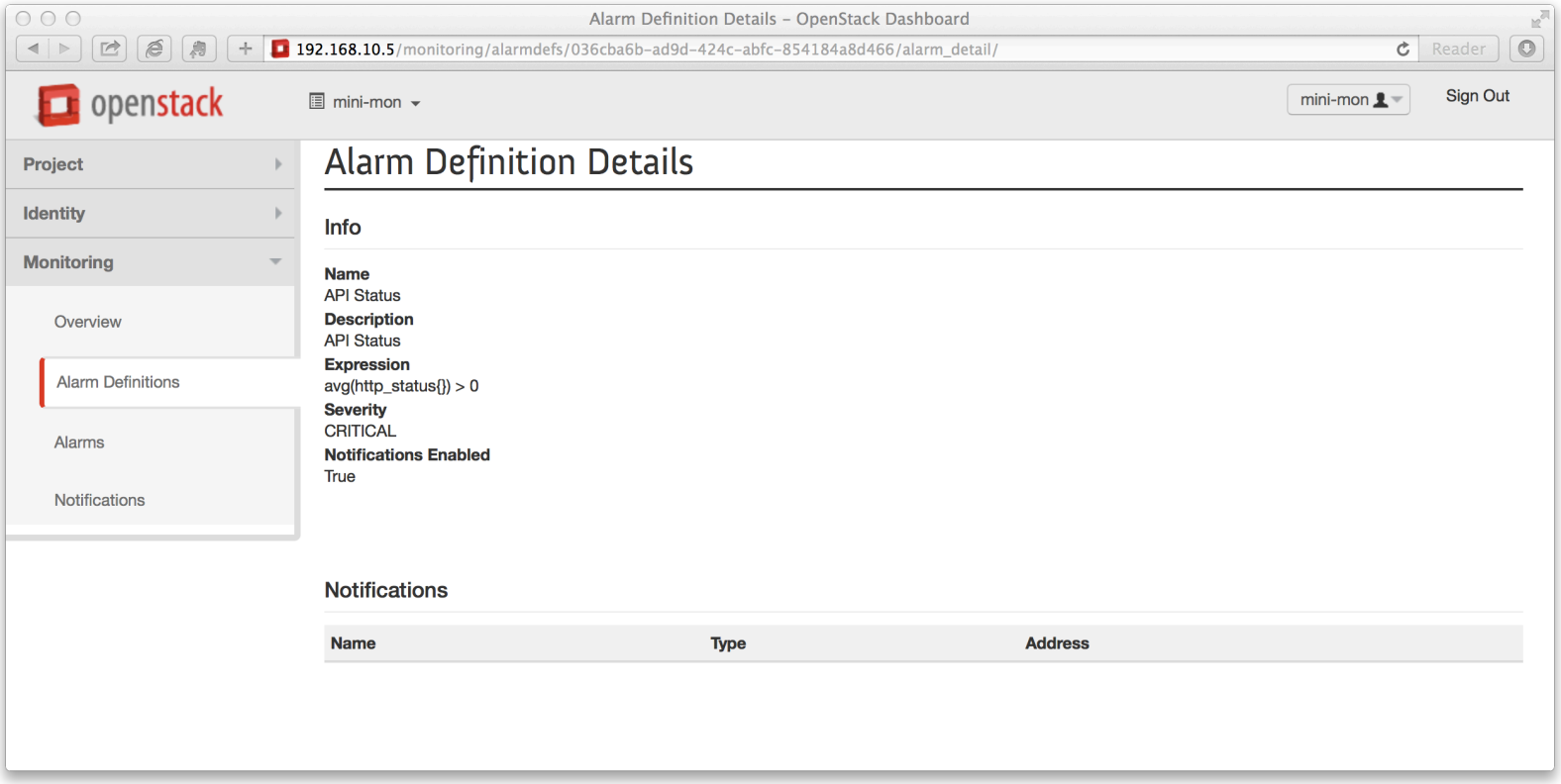


List Alarm History

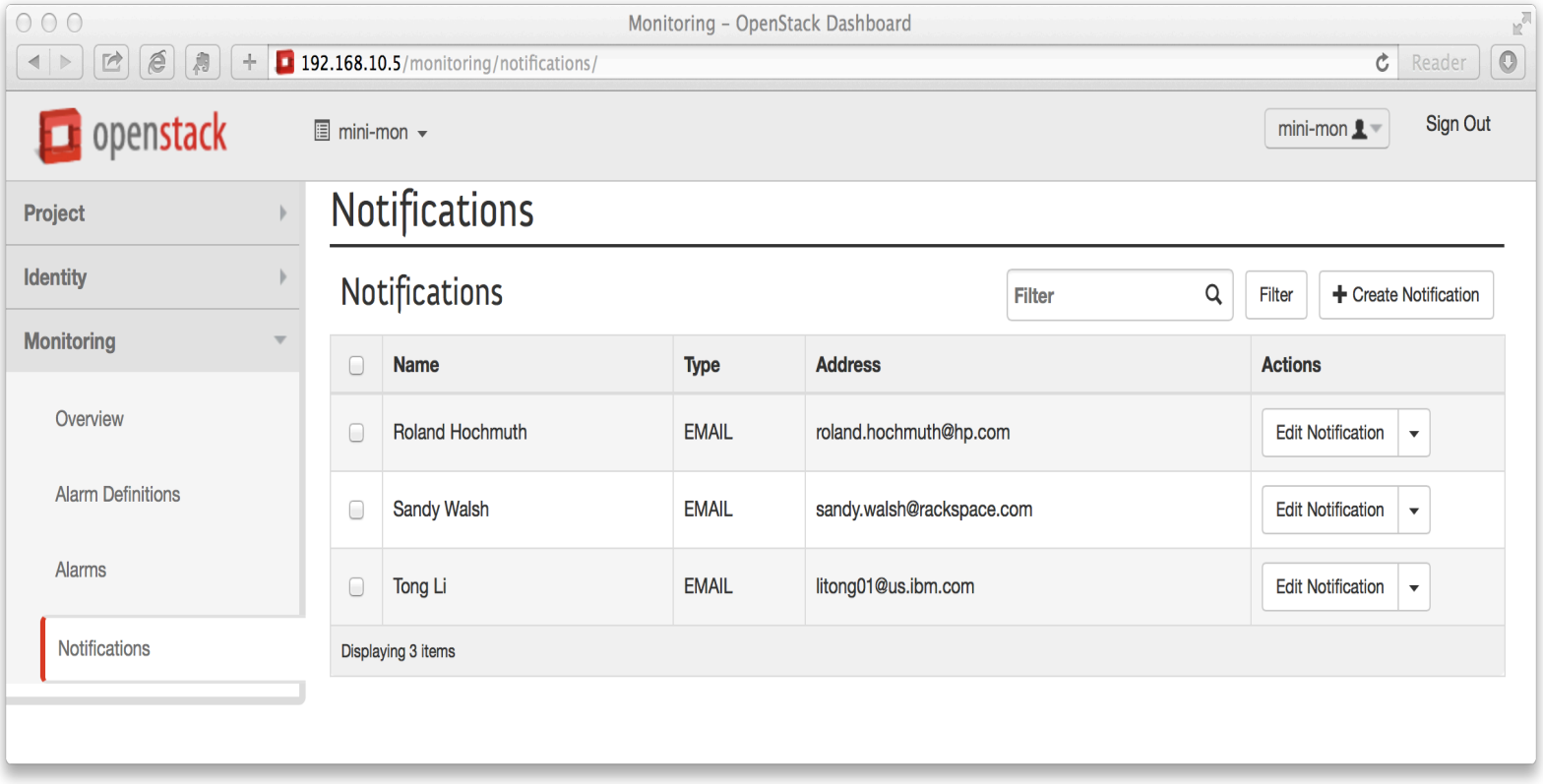


Name	Old State	New State	Timestamp	Reason
036cba6b-ad9d-424c-abfc-854184a8d466	ALARM	OK	2014-11-04T17:46:30.000Z	The alarm threshold(s) have not been exceeded
036cba6b-ad9d-424c-abfc-854184a8d466	OK	ALARM	2014-11-04T17:44:30.000Z	Thresholds were exceeded for the sub-alarms: [avg(http_status) > 0.0]
036cba6b-ad9d-424c-abfc-854184a8d466	ALARM	OK	2014-11-04T17:42:30.000Z	The alarm threshold(s) have not been exceeded
036cba6b-ad9d-424c-abfc-854184a8d466	OK	ALARM	2014-11-04T17:40:30.000Z	Thresholds were exceeded for the sub-alarms: [avg(http_status) > 0.0]
036cba6b-ad9d-424c-abfc-854184a8d466	ALARM	OK	2014-11-04T17:35:30.000Z	The alarm threshold(s) have not been exceeded
036cba6b-ad9d-424c-abfc-854184a8d466	OK	ALARM	2014-11-04T17:31:30.000Z	Thresholds were exceeded for the sub-alarms: [avg(http_status) > 0.0]
036cba6b-ad9d-424c-abfc-854184a8d466	ALARM	OK	2014-11-04T17:19:30.000Z	The alarm threshold(s) have not been exceeded
036cba6b-ad9d-424c-abfc-854184a8d466	OK	ALARM	2014-11-04T16:50:30.000Z	Thresholds were exceeded for the sub-alarms: [avg(http_status) > 0.0]
036cba6b-ad9d-424c-abfc-854184a8d466	UNDETERMINED	OK	2014-11-04T08:07:45.000Z	The alarm threshold(s) have not been exceeded

Show Alarm Definition



List Notifications



Grafana


Grafana – Openstack Dashboard
192.168.10.5/static/grafana/index.html#/dashboard/file/openstack.json?token=36ea96afcaa249ef9bf2b2ac711384df
Reader
Monitoring – OpenStack Dashboard
Grafana – Openstack Dashboard
Openstack Dashboard
Zoom Out
6 hours ago to a few seconds ago
Service API Health
1.0
0.5
0.0
-0.5
-1.0
13:00
13:30
14:00
14:30
15:00
15:30
16:00
16:30
17:00
17:30
18:00
18:30
swift_api
monitoring
nova_api
cinder_api
glance_api
CPU
80.0
60.0
40.0
20.0
0.0
13:00
14:00
15:00
16:00
17:00
18:00
devstack
mini-mon
Disk
9.0
8.0
7.0
6.0
5.0
4.0
13:00
14:00
15:00
16:00
17:00
18:00
devstack
mini-mon
Database
4.0
3.0
2.0
1.0
0.0
13:00
14:00
15:00
16:00
17:00
18:00
devstack
mini-mon
Network
1.3 K
1.0 K
750.0
500.0
250.0
0.0
13:00
14:00
15:00
16:00
17:00
18:00
devstack
mini-mon

Grafana (compute_instance_create_time)