Monasca: Project Onboarding

Witold Bedyk (Fujitsu)
witold.bedyk@est.fujitsu.com
IRC: witek

Johannes Graßler (SUSE)
johannes.grassler@suse.com
IRC: jgrassler
Preliminaries
Slides and Transcript

• Compiled slides (recommended)
  - http://btw23.de/johannes/talks/monasca-onboarding.tar.bz2

• odpdown sources
  - https://github.com/jgrassler/talks/tree/master/monasca-onboarding
This Session

• What it is:
  - Primer on Monasca
  - Overview of Monasca repositories and architecture
  - Introduction to the specifics of Monasca development
  - How can you contribute?

• What it is not
  - General introduction to OpenStack development
  - Refer to Code & Documentation Contributor Guide for that.
What is Monasca?

• Monitoring and Logging as a Service
  - Highly scalable
  - Fault tolerant
  - High Performance
  - Multi-tenant
What is Monasca? (cont.)

• Features:
  - Metrics with dimensions (key/value pairs) as metadata
  - Real-time alerting
  - Pluggable notification engine
  - Flexible aggregation engine
Sources of Documentation

• https://docs.openstack.org/monasca-api
• https://wiki.openstack.org/wiki/Monasca
• http://monasca.io/
Main Contributors

- Fujitsu
- HPE
- OP5
- StackHPC
- SUSE
Architecture and Development
Metrics API (monasca-api)
Metrics API (monasca-api)
Metrics API *(monasca-api)*

- **Repository**
  - [https://github.com/openstack/monasca-api](https://github.com/openstack/monasca-api)

- **Purpose**
  - Receives metrics from agents
  - Makes metrics available for visualization/processing
  - Interface for modifying configuration database (alarms, notifications, ...)

- **Development Information**
  - Most important documentation repository for Monasca: source for [https://docs.openstack.org/monasca-api](https://docs.openstack.org/monasca-api)
  - Contains data model for configuration database *(monasca_api/common/repositories)*
  - Contains database migrations for configuration database (being added in OpenStack Rocky)
  - Deprecated Java implementation: ignore when contributing
Creating Database Migrations

• Generate skeleton revision

   $ cd /opt/stack/monasca-api/monasca_api/db/
   alembic revision
   Generating
   /opt/stack/monasca-api/monasca_api/db/alembic/versions/8ae1c3750508_.py ... done

• Edit revision

   $EDITOR alembic/versions/8ae1c3750508_.py
Metrics API (monasca-api)
Monasca Agent (monasca-agent)
Monasca Agent (**monasca-agent**)

- **Repository**
  - [https://github.com/openstack/monasca-agent](https://github.com/openstack/monasca-agent)

- **Purpose**
  - Collect metrics on monitored systems and forward them to **monasca-api**
  - Easily extensible by adding custom plugins

- **Development Information**
  - Check plugins (for collecting metrics) in `monasca_agent/collector/checks_d`
  - Detection plugins (for detecting/configuring checks with **monasca-setup**) in `monasca_setup/detection/plugins`
  - Please create both if you add a new check.
  - Detailed documentation available in README
Monasca Agent (monasca-agent)
Monasca Client (**python-monascaclient**)
Monasca Client *(python-monascaclient)*

• Repository
  - https://github.com/openstack/python-monascaclient

• Purpose
  - Python client library and CLI client for the Monasca Metrics API
  - Used by users to retrieve metrics/manipulate alarms and by all components that communicate with the Metrics API

• Development Information
  - If you extend the Metrics API, you will have to implement the client side of that extension in *python-monascaclient*.  

Monasca Client (python-monascaclient)
Horizon Plugin (monasca-ui)
Horizon Plugin (monasca-ui)

- Repository
  - https://github.com/openstack/monasca-ui

- Purpose
  - Configuration of alarms/thresholds
  - Visualizing alarms
  - Provide links to metrics and log dashboards
Horizon Plugin (monasca-ui)
Message Queue

Horizon, Monitoring Dashboard

Monasca Client (CLI)

System being monitored
Monasca Agent

Monasca API

Message Queue
(Kafka)

Configuration Database
Message Queue: Interconnects Monasca Components

• Repository
  - N/A (third party component; Apache Kafka)

• Purpose
  - Shuttle metrics, notifications and log entries back and forth between components
Notification Engine (**monasca-notification**)

![Diagram showing the architecture of the Notification Engine](image.png)
Notification Engine (**monasca-notification**)

- **Repository**
  - [https://github.com/openstack/monasca-notification](https://github.com/openstack/monasca-notification)

- **Purpose**
  - Sends notifications if triggered by alarm
  - Supports E-Mail, Webhooks and various chat protocols

- **Development Information**
  - Plugin based
  - Plugins in `monasca_notification/plugins/`
  - Plugins must inherit from
    `monasca_notification.plugins.abstract_notifier.AbstractNotifier`
  - Plugins must be registered in configuration file
Notification Engine (**monasca-notification**)

```
Horizon, Monitoring Dashboard

Monasca Client (CLI)

System being monitored
Monasca Agent

Monasca API

Message Queue (Kafka)

Notification Engine

Configuration Database
```
Threshold Engine (**monasca-thresh**)

![Diagram of Threshold Engine]

- Monasca API
- Monasca Client (CLI)
- Monasca Agent
- System being monitored
- Horizon, Monitoring Dashboard
- Message Queue (Kafka)
- Notification Engine
- Threshold Engine
- Configuration Database
Threshold Engine (**monasca-thresh**)  

• Repository  
  - [https://github.com/openstack/monasca-thresh](https://github.com/openstack/monasca-thresh)

• Purpose  
  - Listen in on metrics and check them against alarm thresholds  
  - Produces messages for **monasca-notification** if thresholds exceeded

• Development Information  
  - Implemented in Java  
  - Contributions may entail changes to **monasca-common**  
  - Uses Apache Storm for processing metrics
Threshold Engine (monasca-thresh)
Transform Engine (monasca-transform)
Transform Engine (monasca-transform)

- Repository
  - https://github.com/openstack/monasca-transform

- Purpose
  - Republish transformed (usually aggregated) metrics as synthetic new metrics
Transform Engine \texttt{(monasca-transform)}
Persister (monasca-persister)
Persister (**monasca-persister**)

- Repository
  - https://github.com/openstack/monasca-persister

- Purpose
  - Consumes metrics from message queue
  - Stores metrics in time series database

- Development Information
  - Two implementations: Java and Python
  - Contributions may entail changes to **monasca-common**
Persister (**monasca-persister**)
Time Series Database for Measurements
Time Series Database for Measurements

• Repository
  - N/A (third party component; can be Cassandra, InfluxDB or Vertica)

• Purpose
  - Store metrics

• Development Information
  - To support a new type of time series database, you will need to add code to monasca-common, monasca-api and monasca-persister.
Monasca Logging Architecture
Monasca Log API (monasca-log-api)
Monasca Log API (monasca-log-api)

• Repository
  - https://github.com/openstack/monasca-log-api

• Purpose
  - Receives log messages from agents

• Development Information
  - Repository contains logging specific parts of documentation
  - Contributions may entail changes to monasca-common
Monasca Log API (monasca-log-api)
Log Agents
Log Agents

- Repository: N/A
- Purpose: Send logs
- Not part of Monasca: logstash, beaver or fluentd with Monasca output plugin.
Log Agents
Monasca Logging

Log Agents
Logstash / Beaver
Monasca output plugin

POST logs

Log API

publish log messages

Message Queue
Kafka
Log Metrics
Log Transformer

Log Agents
  - Logstash / Beaver
  - Monasca output plugin

POST logs

Log API

publish log messages

Message Queue
  - Kafka

parse logs

Log Transformer
  - Logstash
  - IN: Kafka
  - OUT: Kafka

Log Metrics
  - Logstash
  - IN: Kafka
  - OUT: Kafka

filter logs
publish metrics
Log Persister

Log Agents
Logstash / Beaver
Monasca output plugin

POST logs

Log API

publish log messages

Message Queue
Kafka

consume logs

parse logs

filter logs
publish metrics

Log Persister
Logstash
IN: Kafka
OUT: Elasticsearch

Log Transformer
Logstash
IN: Kafka
OUT: Kafka

Log Metrics
Logstash
IN: Kafka
OUT: Kafka
monasca-kibana-plugin

• Repository
  - https://github.com/openstack/monasca-kibana-plugin
Monasca Logging
Tutorial

- Interactive Jupyter notebook
- Demonstrates main Monasca functionalities
- https://github.com/witekest/monasca-bootcamp/
Development Environment
Devstack Setup for Monasca

• `local.conf` for default (Python based) Monasca stack

  ```
  enable_plugin monasca-api \
  git://git.openstack.org/openstack/monasca-api
  ```

• `local.conf` setting for Java based persister

  ```
  MONASCA_PERSISTER_IMPLEMENTATION_LANG=java
  ```
Devstack Setup with Vagrant

# cd monasca-api/devstack
# vagrant up
monasca-docker

- Containerized Monasca deployed with Docker Compose

  # git clone https://github.com/monasca/monasca-docker
  # cd monasca-docker
  # docker-compose up
Running unit tests

# cd $REPO
# tox
# tox -e py27,py35
# tox -e pep8
Running Integration (Tempest) Tests in Devstack

• Add `monasca-tempest-plugin` to `local.conf`

```bash
enable_plugin monasca-tempest-plugin \ 
https://git.openstack.org/openstack/monasca-tempest-plugin
```

• Run tests

```bash
# cd /opt/stack/tempest
# tempest run -r monasca_tempest_tests.tests.api
# tempest run -r monasca_tempest_tests.tests.log_api
```
Running Tempest Tests with `monasca-docker`

- Add section to `docker-compose.yaml`:

  ```yaml
  tempest-tests:
    image: monasca/tempest-tests:latest
  environment:
    KEYSSTONE_SERVER: "keystone"
    STAY_ALIVE_ON_FAILURE: "true"
    MONASCA_WAIT_FOR_API: "true"
  ```

- Run tests

  ```bash
  # docker-compose up -d tempest-tests
  ```
Become part of our community
Why Contribute?

• Pluggable
• Modular
• Customizable
• Small and friendly community
How to contribute?

• Contributor Guide
• We use StoryBoard!
  - Bugs
  - Feature requests
• Specifications repository
  - openstack/monasca-specs
Work to do

• Project priorities

• Important Tasks and Reviews
  - https://storyboard.openstack.org/#!board/60
Where can you help?

• Reviews
• Bugfixes and Bug Reports
  https://storyboard.openstack.org/#!/worklist/213
• Community wide goals
• Installers
• Documentation
Questions?
Thank You!
Thank you!

Slides and Transcript: https://bit.ly/2IfT7c7