Monitoring with Apache Karaf
Decanter
JB Onofré <jbonofre@apache.org>
ASF Member
@jbonofre
PMC Chair for Karaf
PMC member for ACE, Archiva, Aries, Camel, Falcon, Incubator, Lens, ServiceMix, Syncope
PPMC member for Beam, Unomi, …
Software Architect at Talend
Apache Karaf ?
Apache Karaf

- Polymorphic middleware
- Work as container or bootstrapper
- Enterprise ready (provisioning, dynamic, cluster, ...)
- Runtime for lot of other projects (Camel, CXF, ...)

[Image of a sunset over mountains]
Monitoring

• Karaf provides a complete JMX stack
• Without Decanter, monitoring performed by external tools
  • Require additional middleware
  • Not embedded
  • Focused on one feature

=> Decanter is a data collection solution embedded in Karaf providing monitoring and auditing for Karaf itself but also external applications
Decanter

- Karaf subproject and features
- Flexible and extendible
- Can be used for
  - Activity data collection (metric, log, ...)
  - BAM
  - Auditing
  - Alerting (SLA)
Architecture

- Collector
- Collector
- Collector

Dispatcher

- Appender
- Appender
- Appender

- Alerter
- Alerter
Collectors

- Collect any kind of data and send to the dispatcher
- Can be scheduled (poll) or event driven
- Decoupled from the appenders or alerters
- Dynamic (you can add and remove collectors on the fly, no need to restart)
- Bunch of collectors can be active
- Extensible (create your own collectors if needed)
Collectors

- Log & File
- Log4j Socket
- JMX
- REST
- System
- OSGi events (auditing)
- Camel Tracer
@Component(
    name="my.decanter.collector"
    immediate=true)
public class TimeCollector implements Runnable {
    private EventAdmin dispatcher;

    @Override
    public void run() {
        dispatcher.postEvent(new Event("decanter/collect/time", "It's " +
            System.currentTimeMillis()));
    }

    @Reference
    public void setDispatcher(EventAdmin dispatcher) {
        this.dispatcher = dispatcher;
    }
}
Dispatcher

- Powered by OSGi EventAdmin topics
- Very lightweight event broadcasting
- Receive collected data (from the collectors) and provide to the appenders and alerters
Appenders

- Receive the collected data from the dispatcher
- Responsible of pushing & storing the data to a backend
- Decoupled from the collectors & alerters
- Dynamic (you can add and remove appenders on the fly, no need to restart)
- Bunch of appenders can be active, sending the collected data to different backends
- Extensible (create your own appenders if needed)
Appenders

- Elasticsearch (1.x, 2.x, REST) and convenient Kibana feature
- Camel
- Cassandra
- JDBC
- JMS
- Kafka
- Log
- MQTT
- Redis
@Component(
    name=”my.decanter.appender”
    immediate=true,
    property = EventConstants.EVENT_TOPIC + “=decanter/collect/*”)
public class SystemOutAppender implements EventHandler {

    @Override
    public void handleEvent(Event event) {
        StringBuilder builder = new StringBuilder();
        for (String innerKey : event.getPropertyNames()) {
            builder.append(innerKey)
            .append(":")
            .append(event.getProperty(innerKey).toString()).append(" | ");
        }
        System.out.println(builder.toString());
    }
}
Alerters

- Special kind of appenders
- Allows to implement SLA and raise alerts
- Two steps: first check and then forward to alerters
- Dynamic (you can add and remove alerters on the fly, no need to restart)
- Bunch of alerters can be active, sending the alerts to different backends
- Extensible (create your own alerters if needed)
Checker

- Syntax: `attribute.level=expression`
- Attribute: name of the attribute in the collected data
- Level: alert level (warn or error)
- Expression:
  - Range for numeric attribute
  - Equal/Notequal for numeric attribute
  - Match/Notmatch for String attribute
- Examples:
  ThreadCount.error=range:[0,600]
  ThreadCount.warn=range:[0,300]
  loggerLevel.error=match:ERROR
Alerters

• E-mail
• Camel
• Log
Custom Alerter

• Exactly like an appender but listening on “decanter/alert/*” topics
Demo

- Monitoring Karaf (log messages and JMX metrics)
- Store into Elasticsearch
- Dashboards using Kibana
- SLA on the thread count and exceptions with alerts in log
Q&A
Thank You!

- http://karaf.apache.org
- jbonofre@apache.org
- @jbonofre
- http://blog.nanthrax.net