MQTT as a Unified Message Bus for Infrastructure Services

Matthew Treinish
Open Source Developer Advocate - IBM
mtreinish@kortar.org
mtreinish on Freenode

Jeremy Stanley
fungi@yuggoth.org
fungi on Freenode

January 23, 2018

https://github.com/mtreinish/firehose/tree/lca2018
The problem

- The OpenStack community infrastructure operates >40 services on >250 servers
- All the community infrastructure runs on donated public cloud resources
- Some services depend on automation from others
- Several user facing services expose event buses
- A real mess when you try to consume infrastructure events for any purpose
OpenStack Community Infrastructure Firehose

- An MQTT broker for the OpenStack community infrastructure
- Has anonymous, read-only access via MQTT on 1883/tcp
- SSL/TLS MQTT also available on 8883/tcp
- Websockets supported (but temporarily disabled)
MQTT

- Pub/sub messaging protocol
- Formerly MQ Telemetry Transport
- ISO/IEC 20922
- Protocol dates back to 1999
- Standard Maintained by OASIS
- Lightweight design, low bandwidth, and designed to handle unreliable networking
- Popular in IoT and sensor network applications
- Large application ecosystem
MQTT Clients

- Bindings available for most languages
- Eclipse Paho project provides similar interfaces across multiple languages

MQTT Brokers

- Relies on a central broker
- Many different options: https://github.com/mqtt/mqtt.github.io/wiki/servers
MQTT Topics and Subscriptions

- Topics are generated dynamically
- Topics are heirarchical
- Supports wildcarding

Examples:

```
sensors/HOSTNAME/temperature/HDD_NAME
```

- sensors/sinanju/temperature/nvme0n1p1
- sensors/+/temperature/+  
- sensors/sinanju/temperature/+      
- sensors/sinanju/#

3 QoS Levels:

- **0**: The broker/client will deliver the message once, with no confirmation.
- **1**: The broker/client will deliver the message at least once, with confirmation required.
- **2**: The broker/client will deliver the message exactly once by using a four step handshake.

QoS of a message sent with each PUBLISH

Client sends QoS with SUBSCRIBE
The Firehose

- Runs Mosquitto MQTT broker
- Single broker instance
- **Hardware Specs:**

  | vCPUs | 2 |
  | CPU Frequency | 2.6 GHz |
  | RAM | 2 GB |
  | swap | 0 B |
  | Disk | 40 GB |
  | Bandwidth | 200 Mbps |
Mosquitto

- MQTT broker implemented in C
- An Eclipse IoT project
- Support for MQTT v3.1 and v3.1.1
## Services Using the Firehose

<table>
<thead>
<tr>
<th>Service</th>
<th>Base Topic</th>
<th>Source of Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansible</td>
<td>ansible</td>
<td>Ansible MQTT Callback Plugin</td>
</tr>
<tr>
<td>Gerrit</td>
<td>gerrit</td>
<td>germqtt</td>
</tr>
<tr>
<td>Launchpad</td>
<td>launchpad</td>
<td>lpmqtt</td>
</tr>
<tr>
<td>Subunit Gearman Worker</td>
<td>gearman-subunit</td>
<td>subunit-gearman-worker</td>
</tr>
<tr>
<td>Logstash Workers</td>
<td>gearman-logstash</td>
<td>logstash-gearman-worker</td>
</tr>
</tbody>
</table>
Typical Firehose Load

Message Rates (messages / minute)

- Messages Sent
- Messages Received
Manually Load Testing

Message Rates (messages / minute)

- **Messages Sent**
- **Messages Received**
CPU Usage:  

Memory Usage:
Use cases for Firehose

- 3rd Party CI Operators
- Desktop Notifications:
  - mqttwarn: https://github.com/jpmens/mqttwarn
- Inter Service communication:
  - gerritbot: https://git.openstack.org/cgit/openstack-infra/gerritbot/
- Graphing metrics:
  - mqtt_statsd: https://git.openstack.org/cgit/openstack-infra/mqtt_statsd
  - grafana: http://grafana.openstack.org/dashboard/db/mosquitto-status
Where to get more information

**MQTT:**
- http://mqtt.org/
- https://mosquitto.org/
- https://www.eclipse.org/paho/
- #mqtt on Freenode
- https://github.com/mtreinish/pymqttbench

**Firehose:**
- openstack-infra ML openstack-infra@lists.openstack.org
- #openstack-infra on Freenode
- http://docs.openstack.org/infra/system-config/firehose.html
- https://docs.openstack.org/infra/system-config/firehose_schema.html