Nightwatch

Because the night is dark and full of errors…

Daniel Opitz, SuUB Bremen

Mannheim, 02.04.2019

Inhalte

1. Was ist Nightwatch und welche Probleme soll es lösen
2. Rückblick auf die bisherige Entwicklung (a.k.a. Warum hat es so lange gedauert?)
3. Aktuelle Funktionalität
4. Geplante Features
5. Zukunftwünsche

Nightwatch

- Web-basierter Service + Client-Bibliothek(en)
- Stack:
  - Node.js
  - PostgreSQL
  - Redis
  - (RabbitMQ)
  - (Python)

Nightwatch

Abbildung + Dokumentation der Workflows

Kontrolle / Überwachung der Prozesse

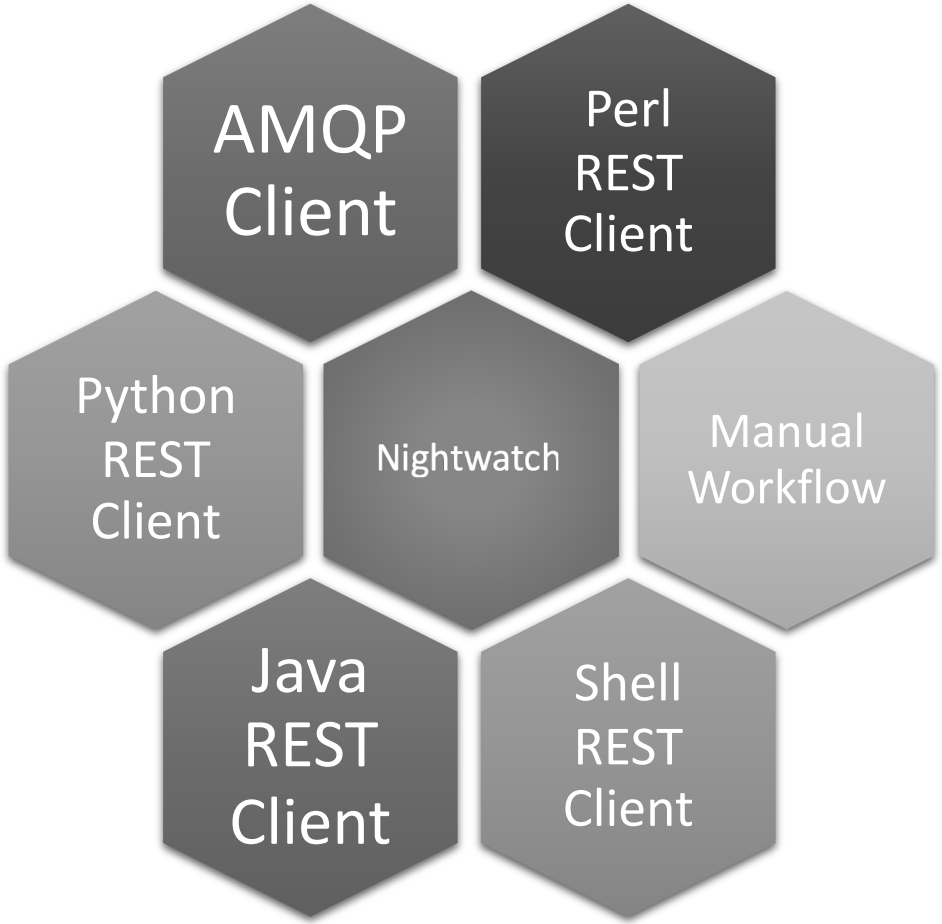
Metriken

Transparenz

Nightwatch

- Metadaten Manager / EntwicklerInnen
  - Welche Datensätze waren Fehlerhaft? Warum?
  - Wie ist der Workflow für die Einspielung von Daten aus XYZ?
- MitarbeiterInnen an der Auskunft
  - Warum sind die aktuellen Daten nicht in der Suchmaschine? Ist irgendwas schief gelaufen?
  - Wer ist der Ansprechpartner für Daten aus XYZ?
- Projektpartner
  - Wen muss ich fragen um die Parameter für ABC anzupassen? Kann ich das selbst machen?
  - Wie komme ich an die Daten?

Nightwatch


AMQP Client
Perl REST Client
Python REST Client
Nightwatch
Manual Workflow
Java REST Client
Shell REST Client

Was bisher geschah…

12.09.2016

Entwicklung des ersten Prototypen gestartet

2016-11-28 16:45:06

Erste Pipeline-Ausführung / Job (bisher 3950 Jobs ausgeführt)

Februar 2017

Entwicklung des 2. Prototypen gestartet
(nie im Einsatz gewesen)

Was bisher geschah…

Juni 2017 – Februar 2019

Kleine Fehlerbehebungen und Sammlung von Use Cases und Erfahrungen

Februar 2019

Überarbeitung des Konzepts + 3. Prototyp

Seit März 2019

Alpha-Version, Intensive Entwicklungsarbeiten, erste (nicht missionskritische) Pipelines laufen bereits

Fazit

Workflowmanagementsysteme sind ziemlich komplex.

Fazit 2

Metadatenverarbeitung ist ziemlich komplex.

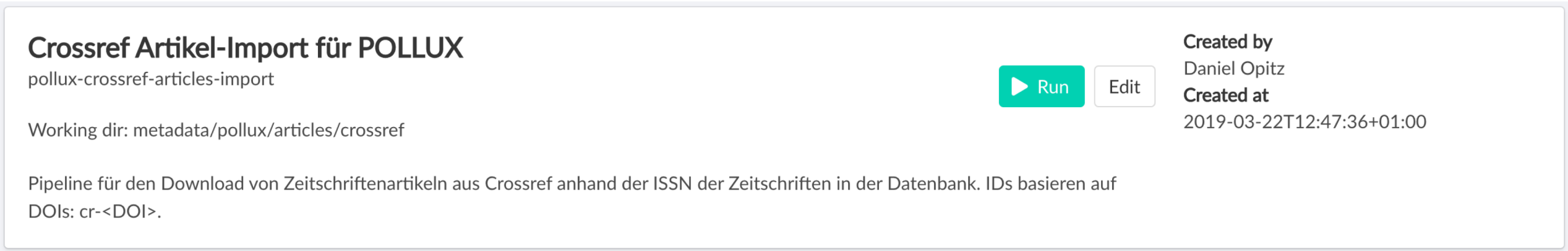
Fazit 3

Workflowmanagementsysteme für die Metadatenverarbeitung sind mindestens genauso komplex.

Funktionalität

- Definition von Pipelines (Workflows)
- Ausführung von Pipelines (Jobs)
- Variablen für die Ausführung von Pipelines
- Speicherung von Logs und Metriken zu den Jobs
- AMQP-Schnittstelle für die Kommunikation mit Clients, die AMQP verstehen
- Einfacher AMQP-Client (Worker) in Python + Protokoll für die Kommunikation mit Nightwatch
- Verteilte Verarbeitung der Pipelines mit AMQP

Definition von Pipelines / Workflows



Definition von Pipelines / Workflows

1

Get list of directories to import
steps.utils.import_checker

Sentinel

dir="<IMPORT_DIR>"

2

Get list of files to import
steps.utils.filelist_getter

ext=".json"

Read file contents
steps.utils.json_reader

Convert to internal representation
steps.converter.crossref_converter

Normalize records
steps.normalizer.default_normalizer

Validate records
steps.validator.default_validator

Export data to the DB

Blueprint

```
{
  "phases": [
    [
      {
        "name": "Get list of directories to import",
        "step": "steps.utils.import_checker",
        "params": {
          "dir": "<IMPORT_DIR>"
        },
        "sentinel": true
      }
    ],
    [
      {
        "name": "Get list of files to import",
        "step": "steps.utils.filelist_getter",
        "params": {
          "ext": ".json"
        },
        "passSentinel": true
      },
      {
        "name": "Read file contents",
        "step": "steps.utils.json_reader"
      },
      {
        "name": "Convert to internal representation",
        "step": "steps.converter.crossref_converter"
      },
      {
        "name": "Normalize records",
        "step": "steps.normalizer.default_normalizer"
      },
      {
        "name": "Validate records",
```

Variablen für die Pipelines

Variables

Crossref Artikel-Download für POLLUX

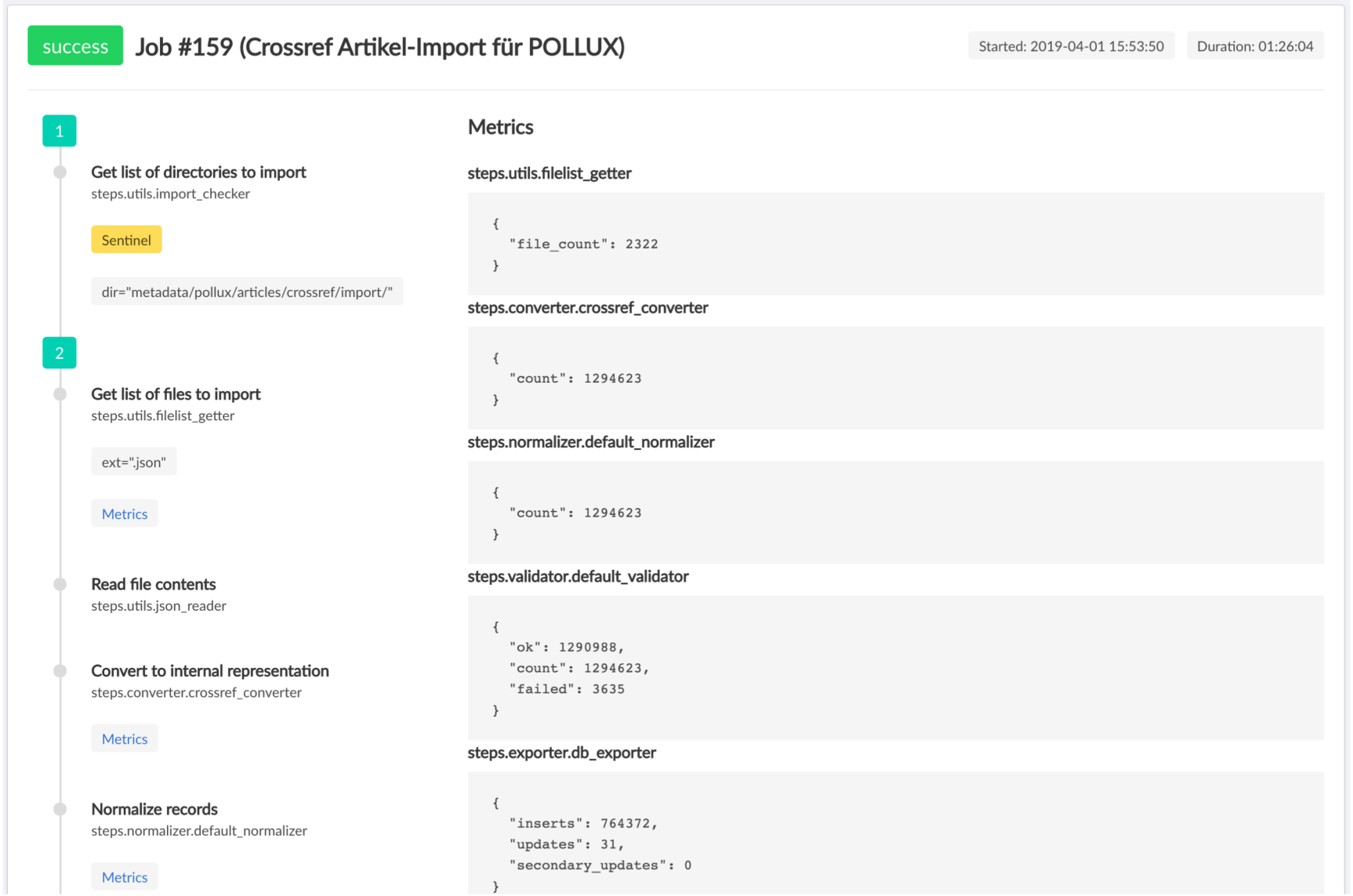
```
{
  "IMPORT_DIR": "$WORKING_DIR/import/$JOB_START_DATE",
  "DOWNLOAD_DIR": "$WORKING_DIR/download/$JOB_START_DATE"
}
```

Ausführung von Jobs

Jobs

159	pollux-crossref-articles-import	(metadata/pollux/articles/crossref/import/2019-04-01)	Started: 2019-04-01 15:53:50	Duration: 01:26:04	success 3
158	pollux-crossref-articles-import	(metadata/pollux/articles/crossref/import/2019-03-30)	Started: 2019-04-01 15:52:01	Duration: 00:00:19	success 3
157	pollux-crossref-articles-import	(metadata/pollux/articles/crossref/import/2019-03-30)	Started: 2019-04-01 15:49:05	Duration: 13:50:02	failed 3
156	pollux-crossref-articles-import	(metadata/pollux/articles/crossref/import/2019-03-30)	Started: 2019-04-01 15:48:02	Duration: 13:51:04	failed 3
155	pollux-crossref-articles-import	(metadata/pollux/articles/crossref/import/2019-03-30)	Started: 2019-04-01 15:47:07	Duration: 13:52:00	failed 3
154	pollux-crossref-articles-import	(metadata/pollux/articles/crossref/import/2019-04-01)	Started: 2019-04-01 15:37:20	Duration: 14:01:47	failed 3

Ausführung von Jobs



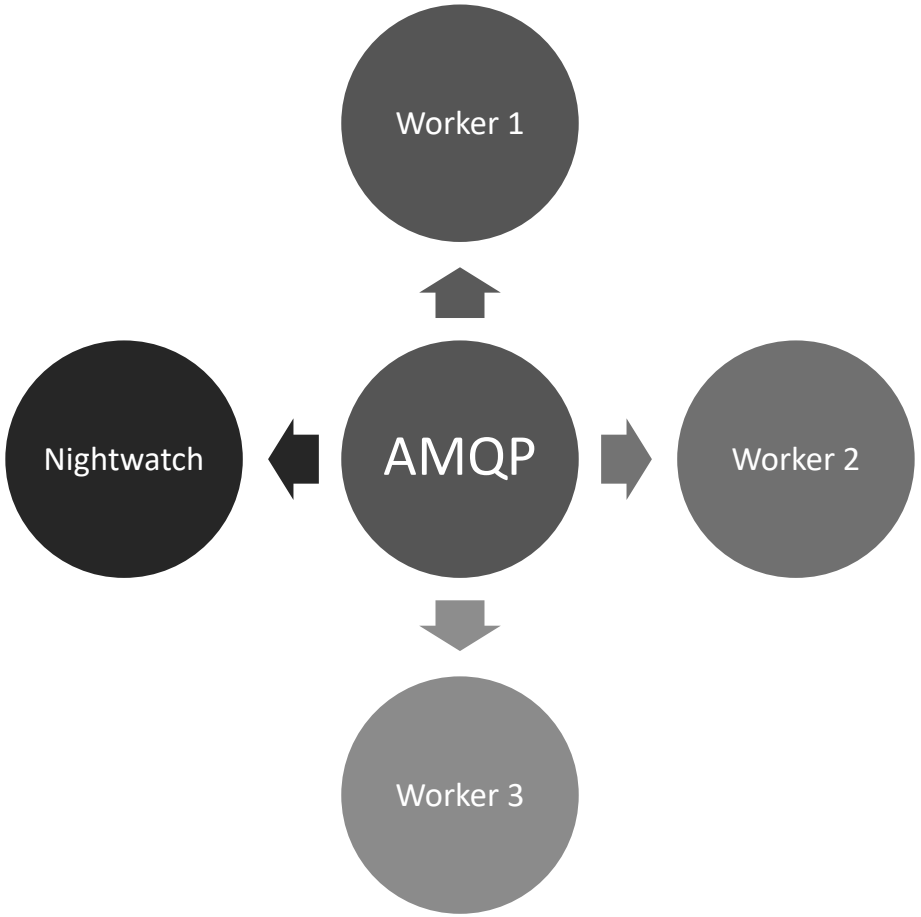
Sammlung von Logs

Logs

steps.normalizer.default_normalizer

```
[
  "Exception running steps.normalizer.default_normalizer",
  "Traceback (most recent call last):\n  File \"/Users/dx/Projects/nightwatch-
scripts/nightwatch/worker.py\", line 58, in process_task\n    result = mod.run(opts)\n  File
\"/Users/dx/Projects/nightwatch-scripts/steps/normalizer/default_normalizer.py\", line 37, in
run\n    normalized = [normalize(reocrd) for record in records]\n  File
\"/Users/dx/Projects/nightwatch-scripts/steps/normalizer/default_normalizer.py\", line 37, in
<listcomp>\n    normalized = [normalize(reocrd) for record in records]\nNameError: name 'reocrd'
is not defined\n"
]
```

AMQP

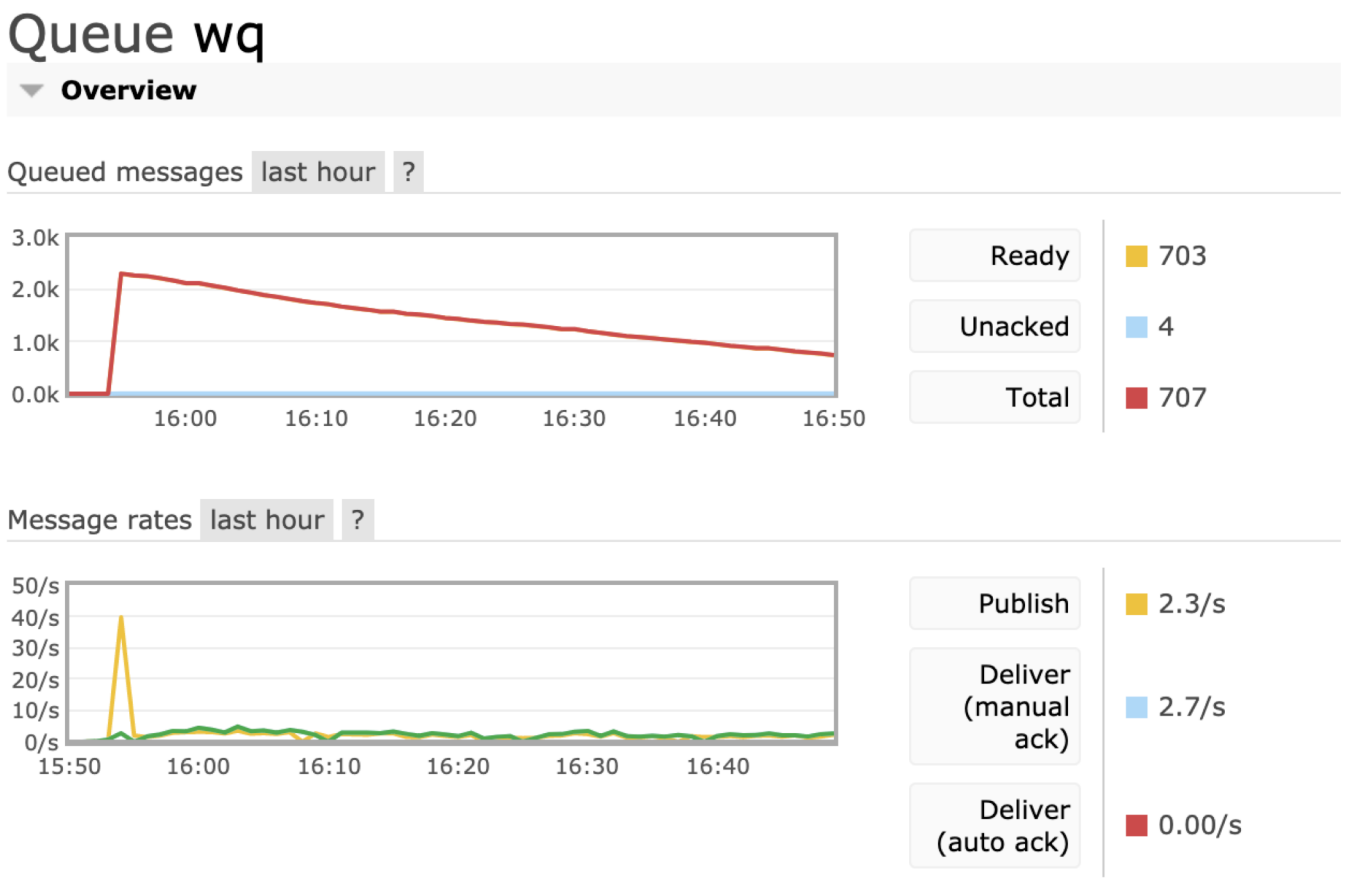


AMQP

Consumers

Channel	Consumer tag	Ack required	Exclusive	Prefetch count	Arguments
172.17.0.1:41082 (1)	ctag1.f09f8d60d91146cd91693608c3e8d16f	●	○	1	
172.17.0.1:41084 (1)	ctag1.73f5a2bf924a4395ab662cec0bcd88f0	●	○	1	
172.17.0.1:41078 (1)	ctag1.89a1ac0f87af41b4aa9ff2a3152feab3	●	○	1	
172.17.0.1:41080 (1)	ctag1.e46dc6d0e6df4cbd84d0abe9c24b8fea	●	○	1	

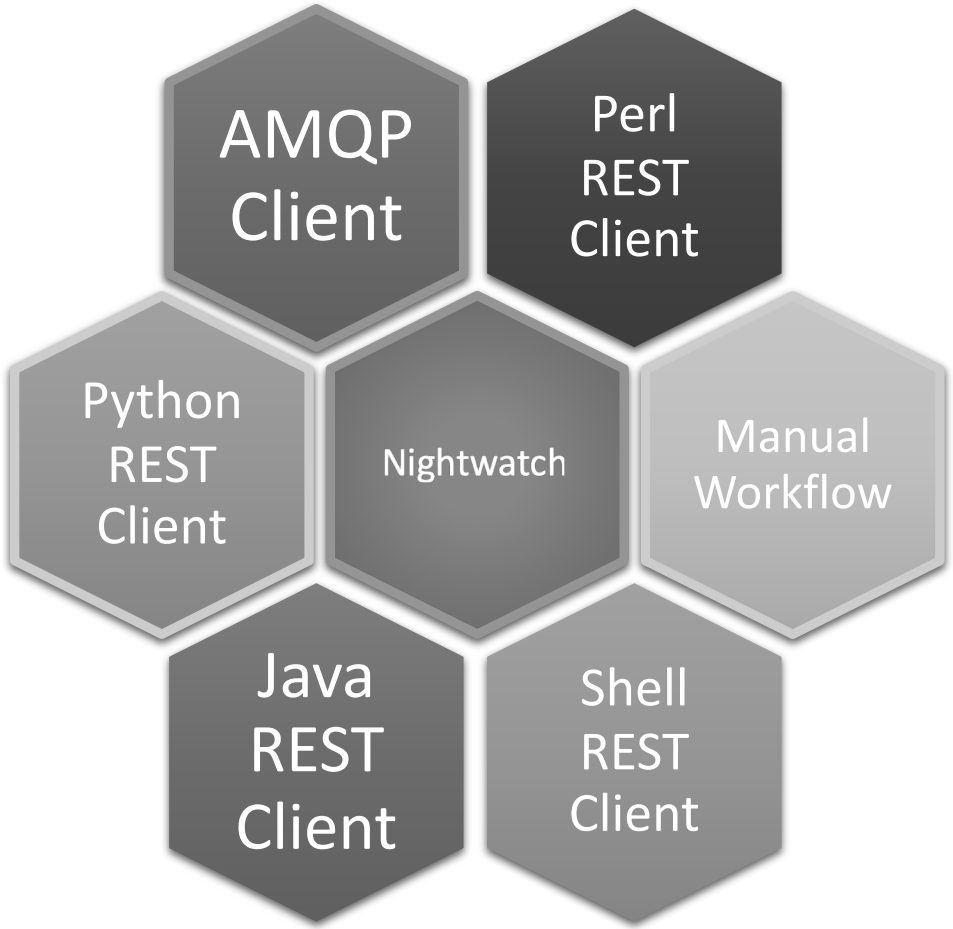
AMQP


Queue wq
Overview
Queued messages last hour ?
3.0k
2.0k
1.0k
0.0k
16:00
16:10
16:20
16:30
16:40
16:50
Ready 703
Unacked 4
Total 707
Message rates last hour ?
50/s
40/s
30/s
20/s
10/s
0/s
15:50
16:00
16:10
16:20
16:30
16:40
Publish 2.3/s
Deliver (manual ack) 2.7/s
Deliver (auto ack) 0.00/s

Nightwatch AMQP Worker (Python)

- Führt dynamisch Module aus
- Jedes Modul hat die gleiche API, kann von dem Worker gestartet und mit Daten / Parametern versorgt werden
- Worker führt eine Aufgabe aus, erzeugt neue Aufgaben und benachrichtigt Nightwatch mit dem Status, den Metriken und den Logs
- Ab einer gewissen Anzahl von programmierten Modulen kann man Pipelines „zusammenklicken“ – hohe Wiederverwendbarkeit
- Nightwatch kann die Pipeline anstoßen indem eine Initialaufgabe erzeugt wird
- Etwas Gewöhnung notwendig, verteilte / parallele Verarbeitung hat gewisse Besonderheiten

Geplante Features



Geplante Features – Kurzfristig (bis Mitte 2019)

- Manuelle Workflows
- Python REST Client + REST Protokoll
- Dokumentation (vor allem Nutzerdokumentation)
- Ausführliche Tests
- Stabilisierung der Code-Basis, der Protokolle und eine Beta

Geplante Features – Mittelfristig (bis Ende 2019)

- Mehr Dokumentation und mehr Tests
- Verbesserung der Benutzeroberfläche und der Benutzbarkeit
- Mehr Möglichkeiten Workflows / Pipelines zu Dokumentieren
- Ticketing System für Probleme
- Grafischer Editor für die Workflows / Pipelines
- Anbindung von Vault[1] für sichere Haltung von Zugangsdaten (Passwörter für die Datenbanken, FTP, API-Keys etc.)
- Benachrichtigungen
- Erste komplette Version und Open-Source-Veröffentlichung
- Download-Möglichkeiten für erzeugte Metadaten / andere Artefakte

1) https://vaultproject.io

Open Source - Warum braucht man so lange?

- Vernünftige Open-Source-Veröffentlichung braucht Zeit
- Software + Code müssen gewisse Qualität haben
- Benutzerdokumentation und Entwicklerdokumentation sind sehr wichtig
- Projekt muss ordentlich Präsentiert werden
- Ein Prozess zur Weiterentwicklung als OSS notwendig

Zukunftswünsche

- Anbindungen an andere Systeme (z.B. Issue Management, Slack / Mattermost, ∅MQ)
- Auswertungen der Metriken, grafische Darstellung
- Client-Bibliotheken in weiteren Programmiersprachen
- Versionierung der Workflows / Pipelines
- Besserer Worker / weitere Worker-Typen
- Nutzer und Mitarbeit der bibliothekarischen Community :-)

Vielen Dank, Fragen?

daniel.opitz@suub.uni-bremen.de