Navigating a Data Warehouse via CLI

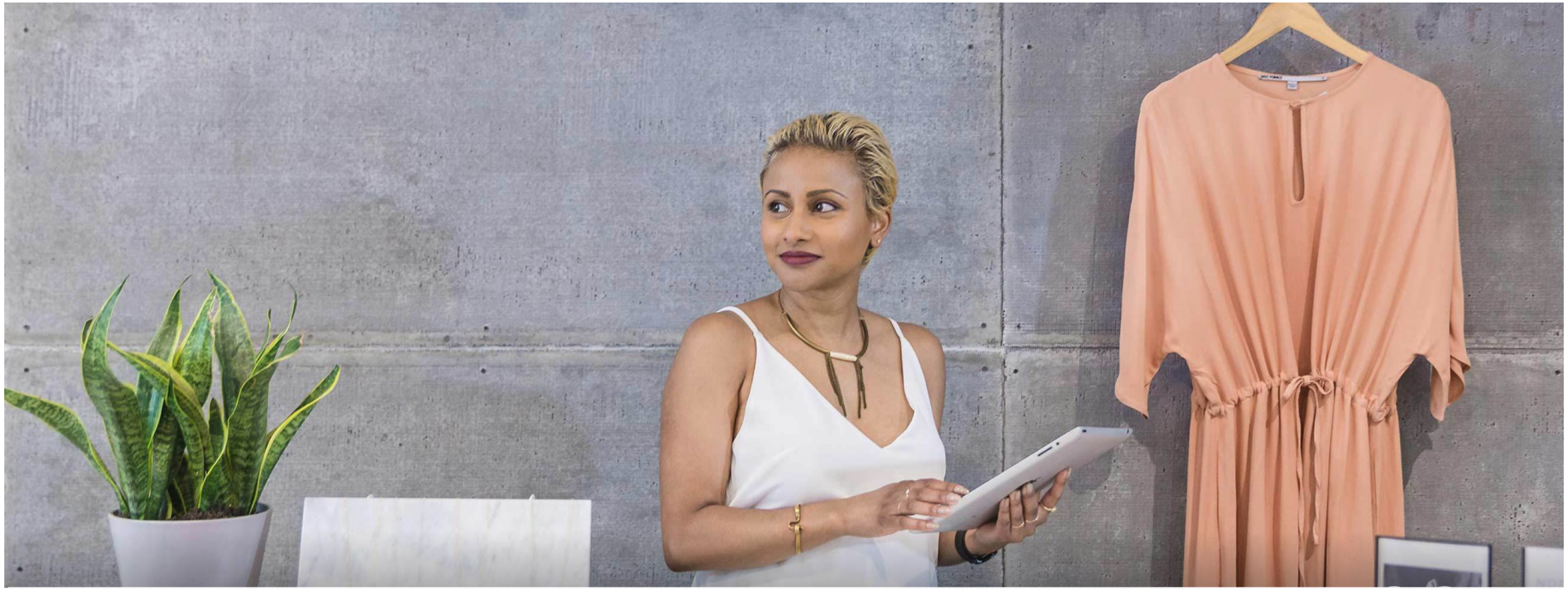
Chris Fournier (@cfournie)

The ecommerce platform made for you

Whether you sell online, on social media, in store, or out of the trunk of your car, Shopify has you covered.

Enter your email address Get started

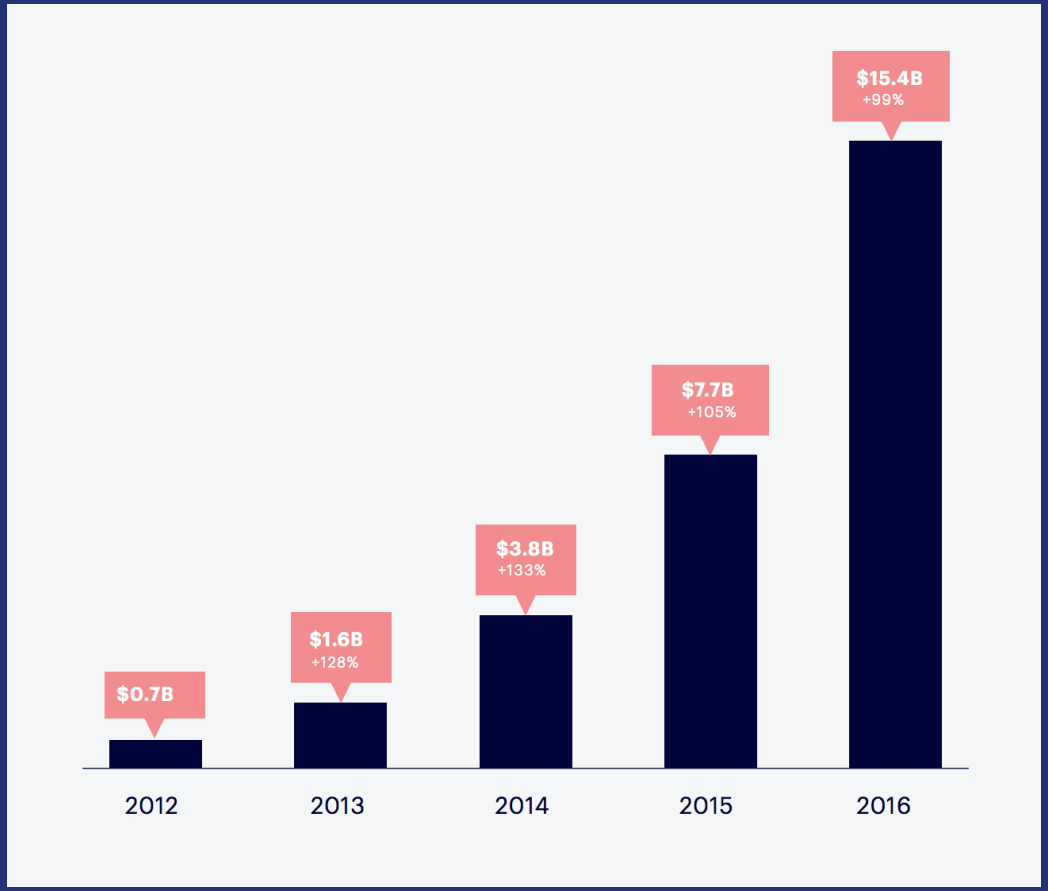
Try Shopify free for 14 days. No risk, and no credit card required.



Data Warehouse

Platform to collect, store, analyze, and report on data

Gross Merchandise Volume



Batch jobs

(E)xtract (T)ransform (L)oad

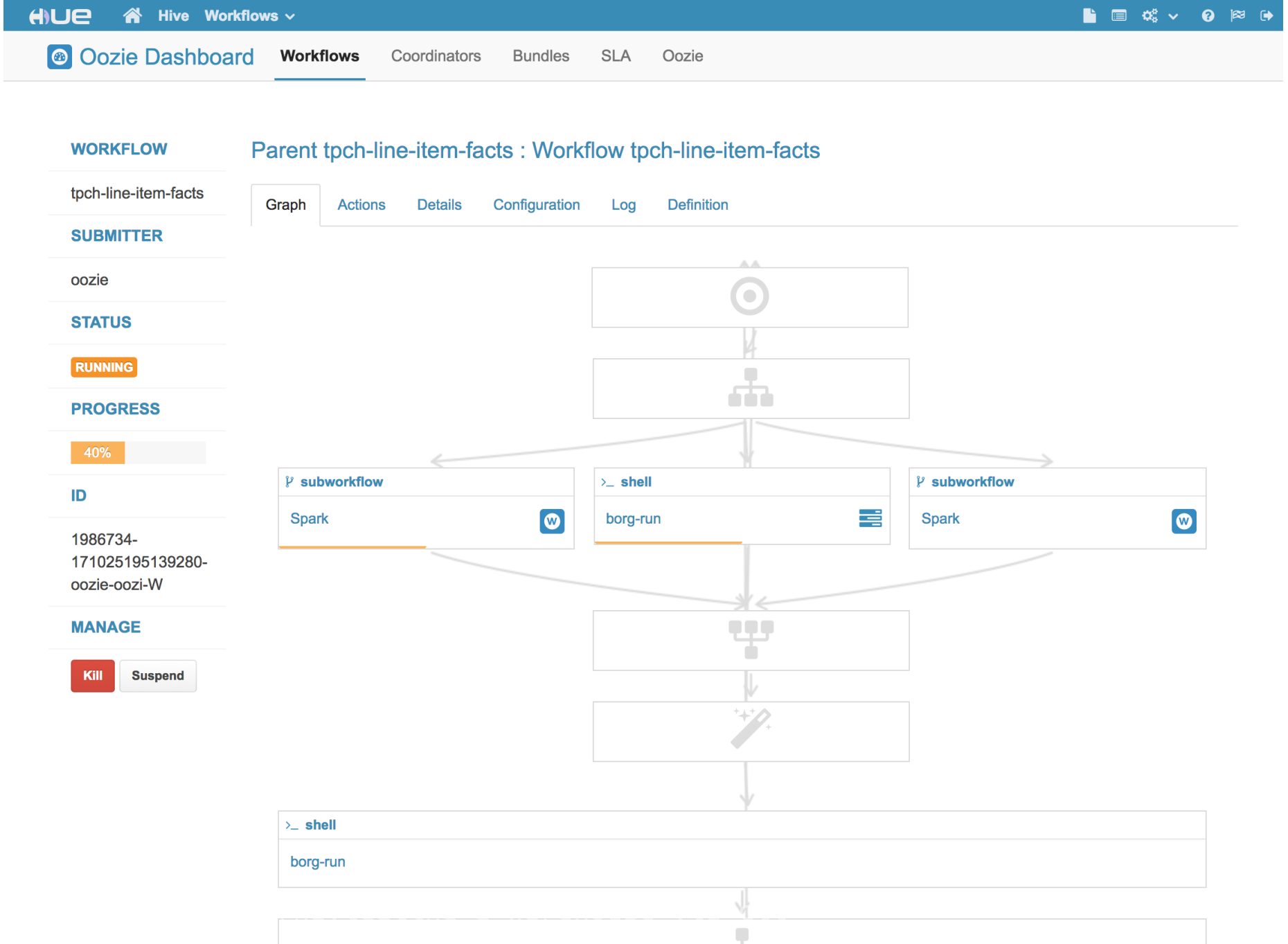
Development speed

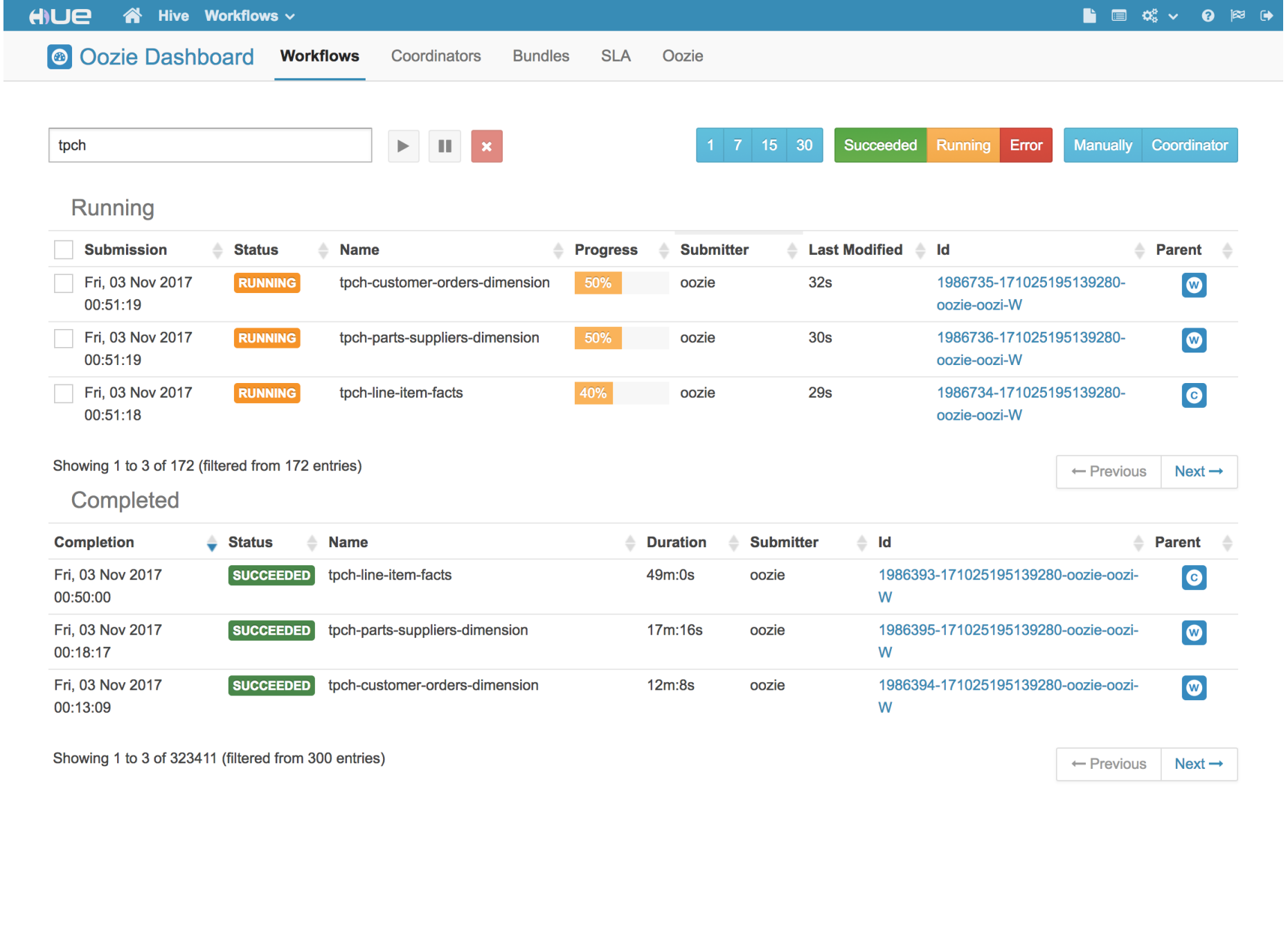
3000+

ETL jobs
(as of Nov 1017)

15.2

Deploys per day
on avg (±8.6 std dev)


Hue
Hive
Workflows
Oozie Dashboard
Workflows
Coordinators
Bundles
SLA
Oozie
WORKFLOW
tpch-line-item-facts
SUBMITTER
oozie
STATUS
RUNNING
PROGRESS
40%
ID
1986734-171025195139280-oozie-oozi-W
MANAGE
Kill
Suspend
Parent tpch-line-item-facts : Workflow tpch-line-item-facts
Graph
Actions
Details
Configuration
Log
Definition
subworkflow
Spark
shell
borg-run
subworkflow
Spark
shell
borg-run


Hive
Workflows
Oozie Dashboard
Workflows
Coordinators
Bundles
SLA
Oozie
tpch
1
7
15
30
Succeeded
Running
Error
Manually
Coordinator
Running
Submission
Status
Name
Progress
Submitter
Last Modified
Id
Parent
Fri, 03 Nov 2017 00:51:19
RUNNING
tpch-customer-orders-dimension
50%
oozie
32s
1986735-171025195139280-oozie-oozi-W
Fri, 03 Nov 2017 00:51:19
RUNNING
tpch-parts-suppliers-dimension
50%
oozie
30s
1986736-171025195139280-oozie-oozi-W
Fri, 03 Nov 2017 00:51:18
RUNNING
tpch-line-item-facts
40%
oozie
29s
1986734-171025195139280-oozie-oozi-W
Showing 1 to 3 of 172 (filtered from 172 entries)
← Previous
Next →
Completed
Completion
Status
Name
Duration
Submitter
Id
Parent
Fri, 03 Nov 2017 00:50:00
SUCCEEDED
tpch-line-item-facts
49m:0s
oozie
1986393-171025195139280-oozie-oozi-W
Fri, 03 Nov 2017 00:18:17
SUCCEEDED
tpch-parts-suppliers-dimension
17m:16s
oozie
1986395-171025195139280-oozie-oozi-W
Fri, 03 Nov 2017 00:13:09
SUCCEEDED
tpch-customer-orders-dimension
12m:8s
oozie
1986394-171025195139280-oozie-oozi-W
Showing 1 to 3 of 323411 (filtered from 300 entries)
← Previous
Next →

Search YAML schedule definitions

```yaml
gross-merchandise-volume-flow:
  frequency: 2h
  owner: team@example.com

  compute-gmv-job:
    resource_class: xxlarge
    executable: jobs/compute_gmv.py
    inputs:
      - /data/raw/orders/
      - /data/raw/transactions/
    output: /data/frontroom/gmv-facts/

  extract-orders-job:
    ...
```

Tables of metadata

Workflow (group)	Freq. (h)	Job	Resources	Etc.
abiding-poetry	6	comp	small	...
abiding-poetry	6	defeated	medium	
abiding-poetry	6	kind-big	xxlarge	
abiding-poetry	6	load-com	small	
abiding-poetry	6	load-kind-big	xlarge	
abiding-poetry	6	swift-energy	medium	
confused-indy	7	eight	small	
confused-indy	7	load-eight	small	
confused-indy	7	load-two-orig	small	
confused-indy	7	luxury-med	xlarge	

Parse and print metadata

```
for flow in flows:
    for job in flow.jobs:
        print('\t'.join((
            flow.name,
            str(flow.frequency),
            job.name,
            job.resource_class.value,
            job.output
        )))
```

What is piping?

```
python jobs2.py | column -t
```

https://brandonwamboldt.ca/how-linux-pipes-work-under-the-hood-1518/

Piping to head

```
python jobs3.py | head
```

```
Traceback (most recent call last):
  File "jobs3.py", line 13, in <module>
    job.output
BrokenPipeError: [Errno 32] Broken pipe
```

Handling SIGPIPE

```
import signal

def handle_sigpipe(signum, frame):
    sys.stdout.close()
    sys.exit(0)

signal.signal(signal.SIGPIPE, handle_sigpipe)
```

Dataset dependencies & paths

Start	End
/data/orders	/data/gmv_per_shop
/data/orders	gmv_per_shop@db
/data/transactions	/data/gmv_per_shop
/data/transactions	gmv_per_shop@db
/data/gmv_facts	gmv_per_shop@db

What's downstream of a bad dataset?

Start	End
/data/raw/orders	/data/gmv_per_shop
/data/raw/orders	gmv_per_shop@database
/data/raw/transactions	/data/gmv_per_shop
/data/raw/transactions	gmv_per_shop@database
/data/gmv_facts	gmv_per_shop@database

```
python downstream.py | awk '$1 == "bad"'
```

What is downstream of this list of bad datasets?

```
python jobs.py | awk '$2 < 7 && $4 == "small"' > bad

python downstream.py > downstream
```

`join` two files

```
a
b
c
```

```
a    1
b    2
b    3
d    4
```

```
join -1 1 -2 1 file_a file_b
```

```
a 1
b 2
b 3
```

How to trace bad data

1. Find bad jobs/outputs
   - Find properties describing a bug
   - Filter metadata tables for affected jobs

2. Find affected downstream outputs
   - Join affected job outputs to tables of paths to get downstream

UNIX tools for tables

- `head` to truncate
- `column -t` to pretty-print
- `grep` and `awk` to filter
- `wc -l` to count
- `sort` to sort
- `uniq` to dedupe / count freqs. (`-c`)
- `join` to join sorted files

Testing

- Move script code into modules
- Scripts produce parseable output

Performance

- `cProfile`
- Use generators (speeds up `head`)
- Use `pandas` (and `numpy`)
- Make expensive columns flags
  - `--expensive-col`
- Cache output (e.g. `joblib.memory`)

Thanks for listening!