AARHUS UNIVERSITET

Microservices and DevOps

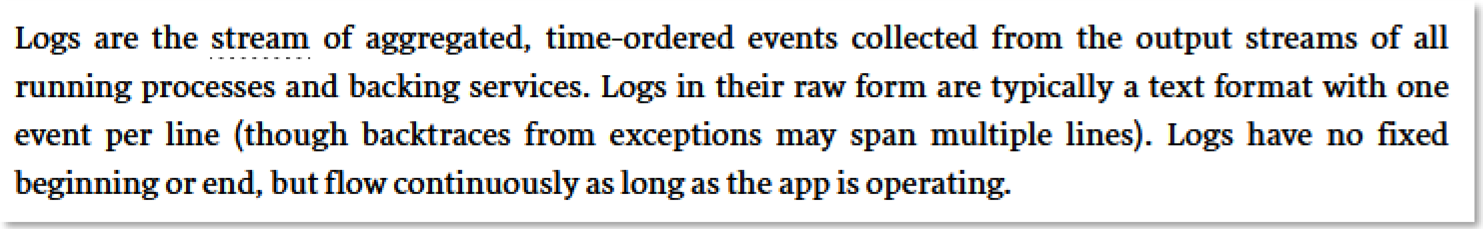
Scalable Microservices

Logging

Henrik Bærbak Christensen

Logging

- Definition



Logs are the stream of aggregated, time-ordered events collected from the output streams of all running processes and backing services. Logs in their raw form are typically a text format with one event per line (though backtraces from exceptions may span multiple lines). Logs have no fixed beginning or end, but flow continuously as long as the app is operating.

- I.e. the source is the *application itself*
  - Providing *architectural* and *domain* metrics
    - If we do our *log instrumentation of the application* properly
- 12Factor recommendation
  - *Logs go to 'stdout', never attempt to manage log output yourself*
  - Docker's recommendation is the same…

Monitoring Requirements

- Which boils down to three things
  - That *we have data* – live and historic
    - That is, our services must *provide data*
  - That data provides the information that allows corrective action
    - That is, our services must provide the *right and relevant data*
  - And – that we can actually find/overview that required data
    - That is, our services' data is available, searchable, and meaningful

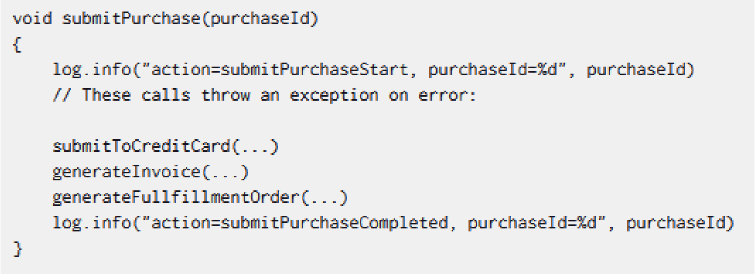
Supporting Requirements

- That *we have data*
  - Logging framework Like SLF4J + Log4J
- That logs *have the information allowing corrective action*
  - Architectural/Design/Coding challenge
    - log.error(“Bang!”); in 2.877 places in code is probably not good
    - (And you only know what data you *should have added* in the log message, once you have your first failure☺)
- That logs are searchable/provides meaningful overview
  - Consistent choice of logging format is essential

AARHUS UNIVERSITET

Logging Tactics

- Best practices (Splunk)
  - Use key-value pairs
  - Human readable formats
    - Binary is hell..
  - Timestamp every event
    - ISO 8601 ALWAYS ☺
  - Use the unique IDs in the domain
    - User id, transaction id, correlation id
  - Log more than just ‘debugging’ events
    - Application and Domain metrics
  - Identify the source (traceability to source code)
    - The method, class, thread, …



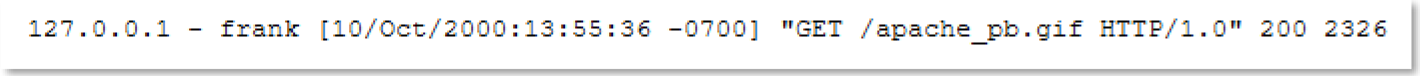
```
void submitPurchase(purchaseId)
{
    log.info("action=submitPurchaseStart, purchaseId=%d", purchaseId)
    // These calls throw an exception on error:

    submitToCreditCard(...)
    generateInvoice(...)
    generateFullfillmentOrder(...)
    log.info("action=submitPurchaseCompleted, purchaseId=%d", purchaseId)
}
```

Logging Formats

- Central tenet:
  - *Consistent format for log messages that can be easily parsed*
- Logs are for humans to read
  - Or a log aggregation system to parse and visualize!
- Serve two purposes
  - Incident handling – find cause to allow corrective measures
  - Overview of operations
- The latter must be facilitated by queries in the log output
  - *Thus the format must be easy to parse…*

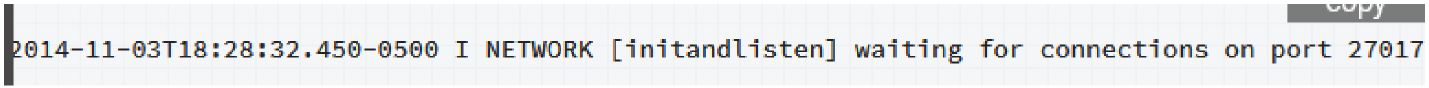
Logging Formats

- Common Log Format (CLF) – used by web servers



```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
```

- MongoDB



```
2014-11-03T18:28:32.450-0500 I NETWORK [initandlisten] waiting for connections on port 27017
```

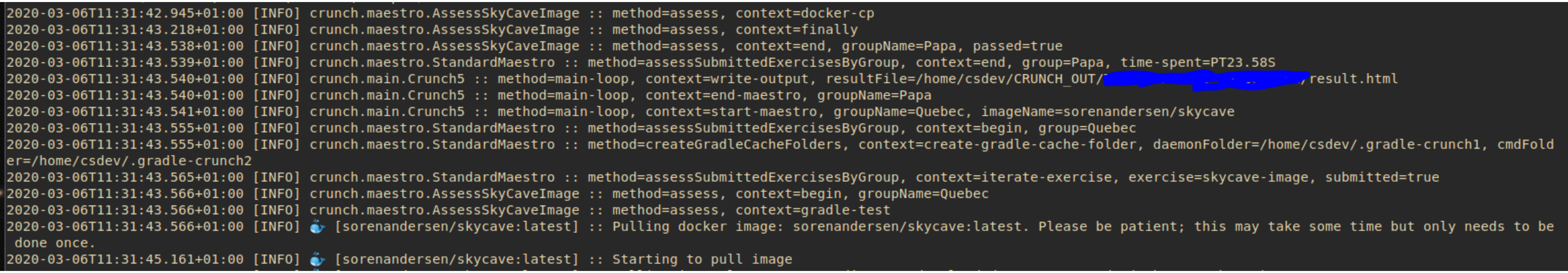
- Windows Event Log, CEF, SysLog, …

- Sigh…

https://www.graylog.org/post/log-formats-a-complete-guide

Key-value format

- I like the idea of key-value, though a bit cumbersome to keep adhering to in code instrumentation…
  - Parse friendly, but still relatively readable by me



```
2020-03-06T11:31:42.945+01:00 [INFO] crunch.maestro.AssessSkyCaveImage :: method=assess, context=docker-cp
2020-03-06T11:31:43.218+01:00 [INFO] crunch.maestro.AssessSkyCaveImage :: method=assess, context=finally
2020-03-06T11:31:43.538+01:00 [INFO] crunch.maestro.AssessSkyCaveImage :: method=assess, context=end, groupName=Papa, passed=true
2020-03-06T11:31:43.539+01:00 [INFO] crunch.maestro.StandardMaestro :: method=assessSubmittedExercisesByGroup, context=end, group=Papa, time-spent=PT23.58S
2020-03-06T11:31:43.540+01:00 [INFO] crunch.main.Crunch5 :: method=main-loop, context=write-output, resultFile=/home/csdev/CRUNCH_OUT/[redacted]/result.html
2020-03-06T11:31:43.540+01:00 [INFO] crunch.main.Crunch5 :: method=main-loop, context=end-maestro, groupName=Papa
2020-03-06T11:31:43.541+01:00 [INFO] crunch.main.Crunch5 :: method=main-loop, context=start-maestro, groupName=Quebec, imageName=sorenandersen/skycave
2020-03-06T11:31:43.555+01:00 [INFO] crunch.maestro.StandardMaestro :: method=assessSubmittedExercisesByGroup, context=begin, group=Quebec
2020-03-06T11:31:43.555+01:00 [INFO] crunch.maestro.StandardMaestro :: method=createGradleCacheFolders, context=create-gradle-cache-folder, daemonFolder=/home/csdev/.gradle-crunch1, cmdFold
er=/home/csdev/.gradle-crunch2
2020-03-06T11:31:43.565+01:00 [INFO] crunch.maestro.StandardMaestro :: method=assessSubmittedExercisesByGroup, context=iterate-exercise, exercise=skycave-image, submitted=true
2020-03-06T11:31:43.566+01:00 [INFO] crunch.maestro.AssessSkyCaveImage :: method=assess, context=begin, groupName=Quebec
2020-03-06T11:31:43.566+01:00 [INFO] crunch.maestro.AssessSkyCaveImage :: method=assess, context=gradle-test
2020-03-06T11:31:43.566+01:00 [INFO] 🐳 [sorenandersen/skycave:latest] :: Pulling docker image: sorenandersen/skycave:latest. Please be patient; this may take some time but only needs to be
 done once.
2020-03-06T11:31:45.161+01:00 [INFO] 🐳 [sorenandersen/skycave:latest] :: Starting to pull image
```

- Issue: 3rd party software drivers seldom agree on the logging format, and that’s how MS systems are ☹