

Logging Made Simple With Quarkus, MDC, and Google Cloud

Jure Repe
Software Engineer



OUTFIT7

Hello 🙌

- Software Engineer @ Outfit7 Group
- Backend department
- I like Java (most of the time)



Java the Cat

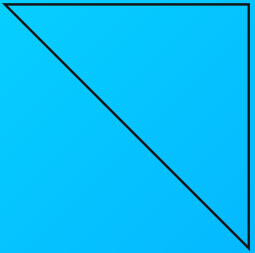
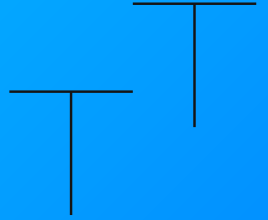
Agenda

Why even
bother with
logging?

How we
implemented
logging at
Outfit7 using
Quarkus and
Google Cloud

A short and
sweet code
example

Logging in Cloud Applications



Cloud Applications and Observability

- Observability
 - Application metrics
 - Distributed tracing
 - **Logging**
- An insight into our application service
 - Troubleshooting
 - Alerting
 - Audit logging

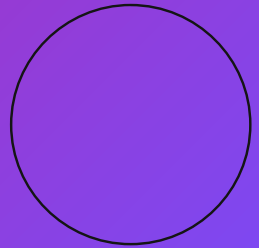
Logging Considerations

- Distributed systems produce distributed logs
 - Spread across different services
 - Hard to track request from start to finish
- Performance implications
- Cost

- Possible solutions ->
 - Log aggregation in a centralized system
 - Distributed tracing



Logging at Outfit7



Quarkus

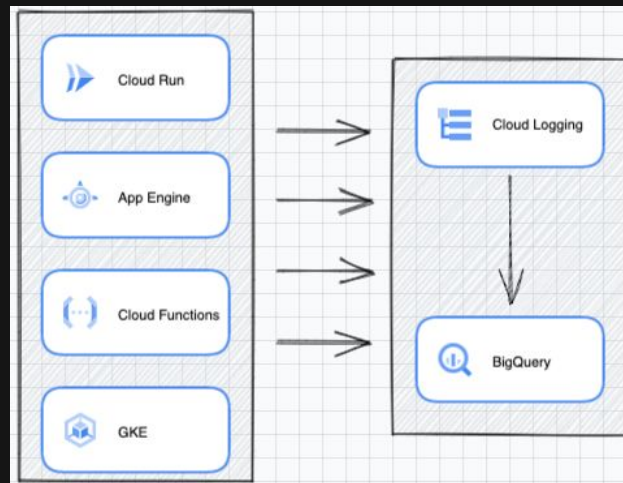
- Container-first, Supersonic, Subatomic Java framework
- Quarkus functionality can be extended with
 - Official extensions
 - Community extensions

- Quarkus Json Logging extension
 - Community extension
 - Produces a JSON structured log


```
{"timestamp":"2022-05-31T21:05:18.372+02:00","sequence":1762,"loggerClassName":"org.slf4j.impl.Slf4jLogger","loggerName":"com.example.rest.QuotesResource","level":"INFO","message":"Fetching Quote of the Day from external service","threadName":"executor-thread-0","threadId":87,"mdc":{"p":"/quotes/today"},"hostName":"jures-macbook-pro-2.local","processName":"makeit-quarkus-demo-dev.jar","processId":41861,"p":"/quotes/today","severity":"INFO","logger":"com.example.rest.QuotesResource"}
```

Google Cloud Platform

- Our applications are hosted on GCP
- Centralized Logging: Log Viewer
 - Log Exclusions
 - Log Sinks (BigQuery)



>	i	2022-05-31 14:15:24.697 CEST	Fetching Quote of the Day from external service	
>	i	2022-05-31 14:15:25.932 CEST	GET 200 761 B 7.245 s HTTPie 3.2.1 https://makeit-quarkus-demo-e3n2jpiutq-ew.a.run.app/quotes/today	
>	i	2022-05-31 14:15:50.093 CEST	run.googleapis.com _le.cloud.run.v1.Services.GetService _-playground/services/makeit-quarkus-demo jure.repe@outfit7.com audit_log, method: "google.cloud.run.v1.Services.GetService", ..	
>	i	2022-05-31 14:16:10.602 CEST	Fetching Quote of the Day from external service	 
>	i	2022-05-31 14:16:10.895 CEST	GET 200 702 B 297 ms HTTPie 3.2.1 https://makeit-quarkus-demo-e3n2jpiutq-ew.a.run.app/quotes/today	

Mapped Diagnostic Context (MDC)

- Problem ->
 - Requests flow through different parts of the service
 - Certain information isn't available when you produce the log
- Solution -> Mapped Diagnostic Context
 - Built into logging frameworks (logback, log4j,...)
 - Attached to the executing thread, available within that context
- With Mapped Diagnostic Context you can propagate information through the system

```
▼ jsonPayload: {
  hostName: "localhost"
  level: "INFO"
  logger: "com.example.rest.QuotesResource"
  loggerClassName: "org.slf4j.impl.Slf4jLogger"
  loggerName: "com.example.rest.QuotesResource"
  ▼ mdc: {
    X-B3-TraceId: "4c63f5817517b2254987de567ad879e1"
    p: "/quotes/today"
  }
  message: "Fetching Quote of the Day from external service"
  p: "/quotes/today"
  processId: 1
  processName: "app.jar"
  sequence: 124
  threadId: 15
  threadName: "executor-thread-0"
  timestamp: "2022-05-31T12:16:10.601Z"
}
```

Bringing It All Together

- Quarkus JSON logging extension
 - Logging to stdout
- Applications are hosted on GCP
 - Log Viewer for browsing logs
 - Special JSON fields (message, severity, trace, ...)
- At the start of every request we enrich the MDC with
 - Trace id, query parameters, timestamps, ...
 - And other custom fields

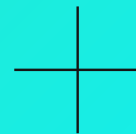
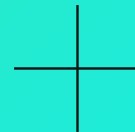
```
@Override
public void onStart() {
    setupRootLogger();
}

/**
 * Setup Stackdriver format, this can be removed once Stackdriver will support coloring of top logs automatically
 */
private void setupRootLogger() {
    Arrays.stream(Logger.getLogger("").getHandlers())
        .filter(it -> it instanceof ExtHandler)
        .flatMap(it -> Arrays.stream(((ExtHandler) it).getHandlers()))
        .filter(it -> it instanceof FileHandler)
        .forEach(it -> {
            it.setFormatter(applicationLogJsonFormatter);
            Writer writer = getFileWriterWithIncreasedSize();
            ((FileHandler) it).setWriter(writer);
        });
    Arrays.stream(Logger.getLogger(MdcLoggingFilter.class.getName()).getHandlers())
        .filter(it -> it instanceof FileHandler)
        // No need to increase buffer size for request log, since we log just empty message
        .forEach(it -> it.setFormatter(requestLogJsonFormatter));
}
```



**Stop with the talking,
show us the code**

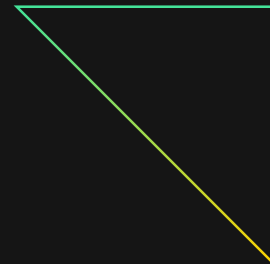
`(^o□o)^ ~ _ _ _`



Keep in Touch

- Jure Repe @ LinkedIn
- jure.repe@outfit7.com
- Example code available on Github





DISCLAIMER: The content of this presentation is proprietary and confidential information of Outfit7 Limited and related entities. It is prohibited to duplicate or distribute this presentation or its content to any third party without the prior written permission of Outfit7 Limited.

© 2010–2022 Outfit7 Limited and related entities. All rights reserved.

