Microservices

Monitoring & Observability

Hönnun og smíði hugbúnaðar

Haust 2022

Microservice erfiðleikar

- Microservices koma með marga góða hluti en einnig slæma
- Flækja eykst sérstaklega fyrir monitoring og observability
- Mörg service og servers til að monitor-a
- Við þurfum að geta tengt verkflæði saman
- *“We replaced our monolith with micro services so that every outage could be more like a murder mystery”*

Multiple Services, Multiple Servers

Þurfum að aggregate-a og tengja metrtics og logs fyrir service-in

Þurfum að monitor-a CPU / Memory fyrir container-a / servers / nodes

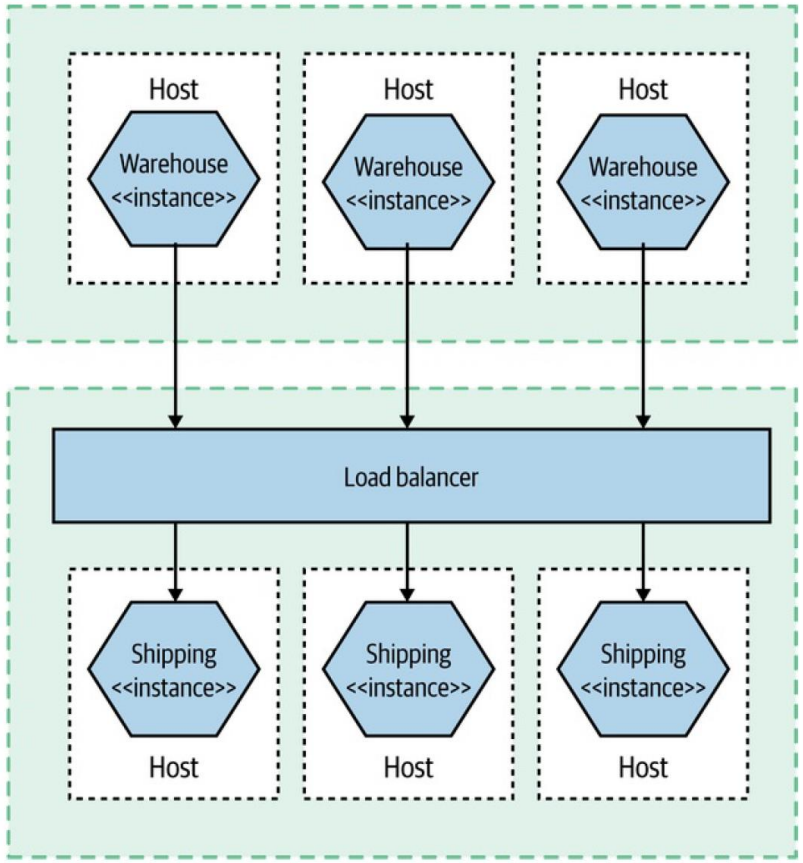


Figure 10-3. Multiple collaborating services distributed across multiple hosts

Observability vs Monitoring

- Tengd hugtök en ekki þau sömu
- Observability er eiginleiki kerfisins til að geta skilið og fengið innsýn inn í kerfið, verkflæði og stöðu þess
- Monitoring er eitthvað sem við gerum, *the activity of monitoring*
  - T.d. alerting, charts, graphs
- Observability gerir gott monitoring mögulegt
  - T.d. logging, tracing, metrics

Building Blocks for Observability

- Log Aggregation
- Metrics Aggregation
- Distributed Tracing
- Are we doing OK?
- Alerting
- Semantic monitoring
- Testing in production

Log Aggregation

- Í distributed kerfi þurfum við leið til að skoða log-a saman
- Gegnur ekki að hafa logs dreifð á milli servers / containers / DBs
- Við þurfum að safna logs saman á miðlægan stað
- Þ.e.a.s þurfum að aggregate-a logs
- Þetta eykur observability til muna
- Auðveldar að query-a og tengja logs á milli service-a
  - sérstaklega með Correlation Ids

Log Aggregation

BEFORE ANYTHING ELSE

Before you do anything else to build out your microservice architecture, get a log aggregation tool up and running. Consider it a prerequisite for building a microservice architecture. You'll thank me later.

Log Aggregation
Implementation

- Datadog
- Humio
- NewRelic
- Elastic

Kosturinn við þessa mynd er að application-ið getur verið óvart um undirliggjandi aggregation tól

Flest af þessum tólum virka svipað

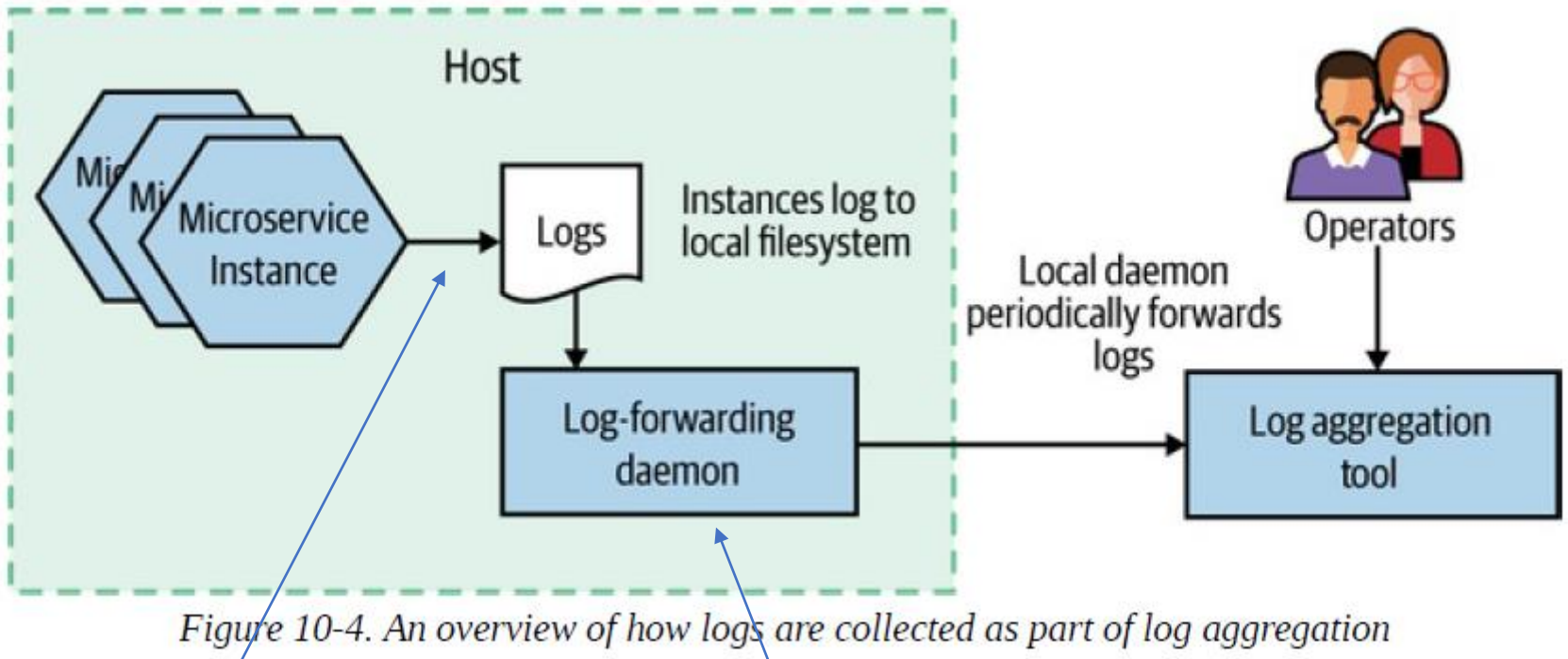


Figure 10-4. An overview of how logs are collected as part of log aggregation

Hvert service log-ar í sína eigin skrá

Einhvers skonar keyrlsa monitor-ar skránna og sendir gögn yfir í aggregation tól, t.d. Datadog Agent / Elastic Beats

Log Aggregation
Log format

- Json algengasta formatið
- Viljum hafa consistency á format-i á milli service-a
- Viljum hafa consitency á format-i fyrir sömu log *tegundir*
  - T.d. Gætum viljað query-að eftir order id fyrir ákveðnar request-ur
- Application-ið ætti að sjá um þetta consistency ekki aggregation tólið

Log Aggregation
Correlation Ids

- Þurfum að geta tengt log-a fyrir verkflæði saman
- Getum gert þetta með Correlation Ids
  - Líka þekkt sem trace ids, request ids
- Fyrir hvert request / event þá log-um við sameiginlegt Id
- Með log aggregation getum við þá query-að alla log-a fyrir ákveðið id
- Id-ið er generatað í byrjun á verkflæðinu (t.d. Með Api Gateway)
- Yfirleitt sent með sem header

Log Aggregation
Correlation Ids

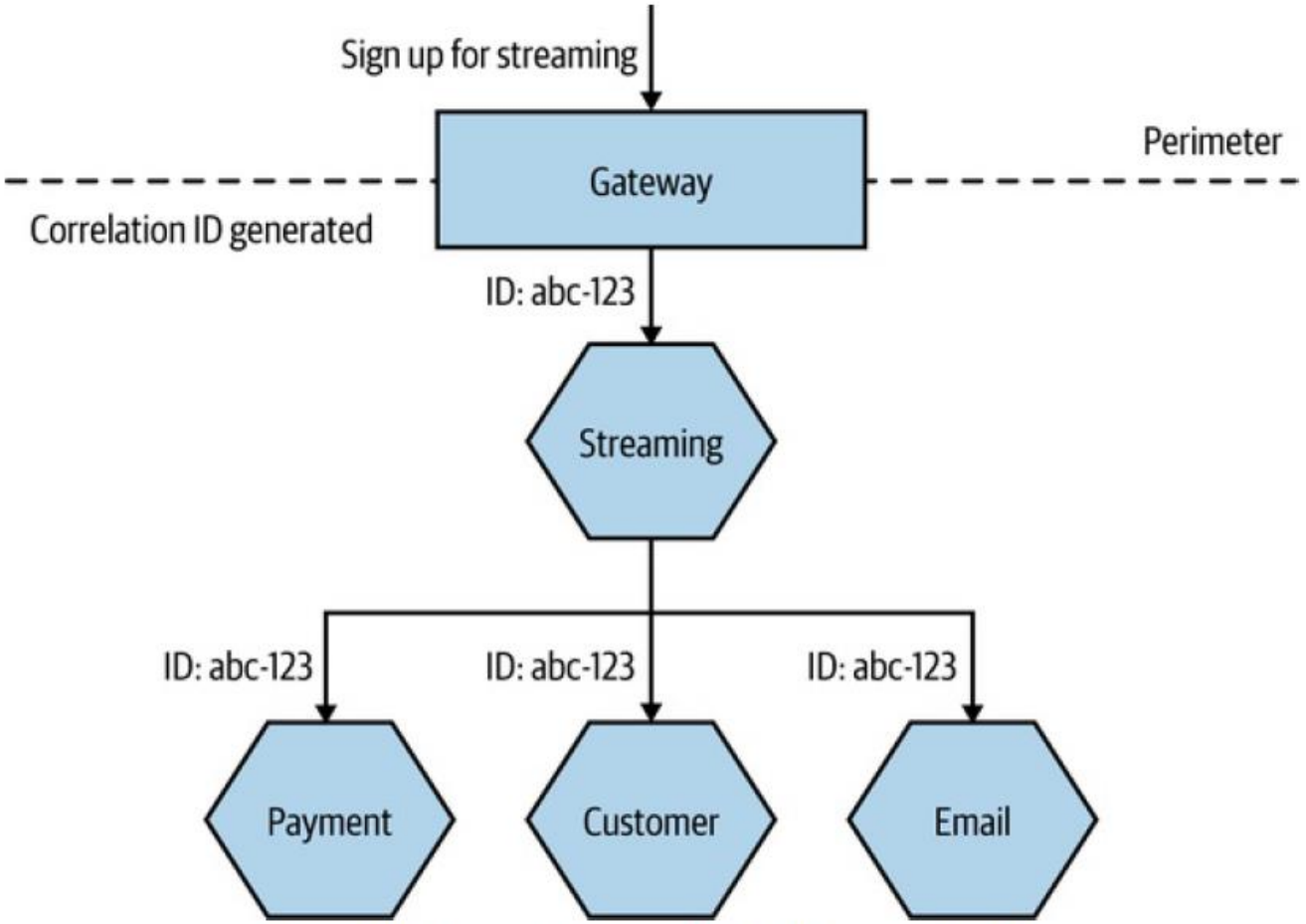


Figure 10-6. Generating a correlation ID for a set of calls

Log Aggregation
Distributed logging time pitfall

- Getum ekki endilega treyst log tímanum
- Klukkann á mismunandi server-um gæti verið *mismunandi*
  - Ekki in-sync
- Oft hægt að treysta á það en eitthvað til að hafa í huga
- Logs geta þannig verið sýnd í rangri röð
- Tracing tól geta hjálpað við þetta

Dæmi um hvað ætti að log-a

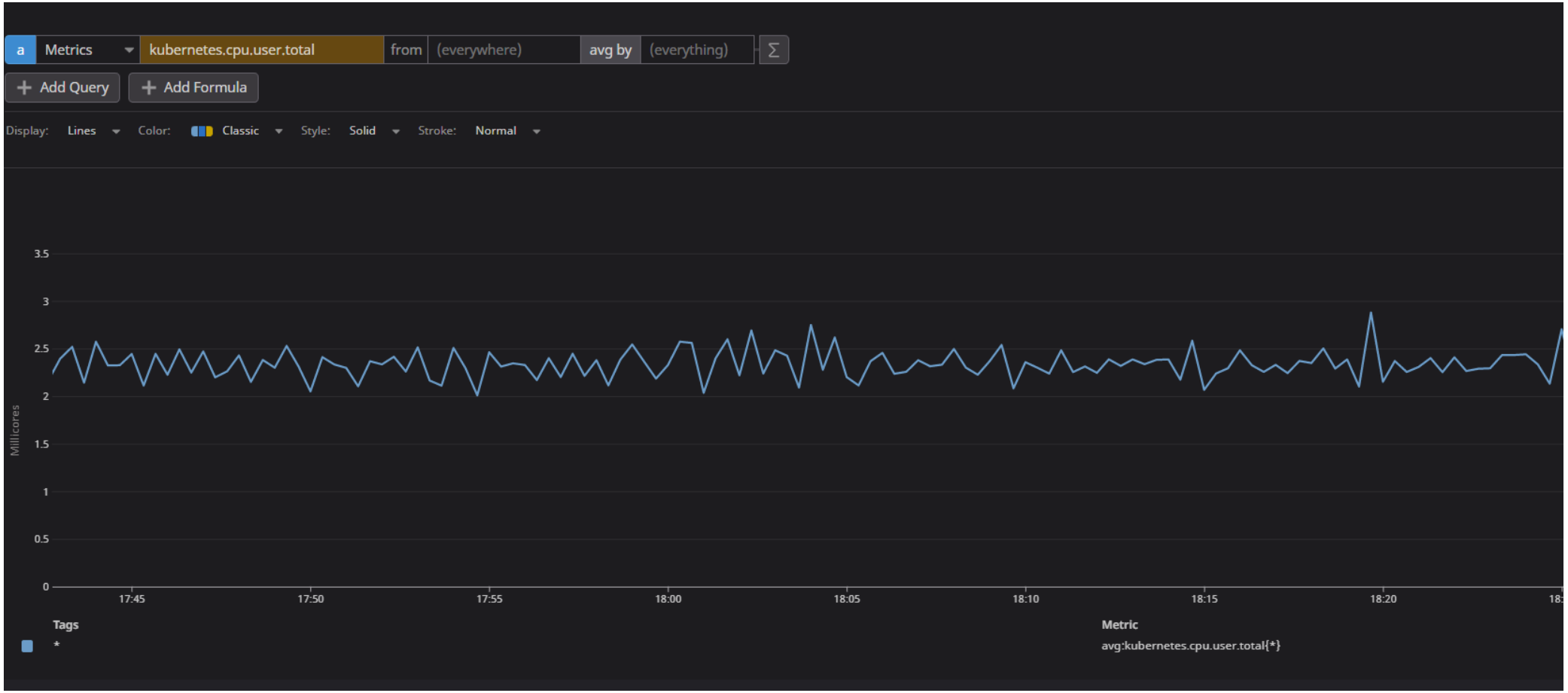
- Correlation Id
- Application
- Application instance
- Duration
- Start Date & End Date
- Environment
  - T.d. Dev, Prod, Staging
- Log Level
  - T.d. Error, Warning, Info
- Annað
  - Headers, http method, url path, request, response, status code, stack trace, calling method etc.

Metric Aggregation

- Metrics eru yfirleitt töluleg gildi um eitthvað í kerfinu
- T.d. CPU load, avg load time, avg response time, requests / minute, errors / minute, orders / minute, percentile of successful requests etc.
- Mikilvægt fyrir að fylgjast með heilsu kerfisins
- Oft hægt að gera gagnleg gröf og alerting út frá metrics

Metric Aggregation

Dæmi



Metric Aggregation
Implementation

- Prometheus
- Honeycomb
- Lightstep
- Datadog
- Elastic

Distributed Tracing

- Þurfum að geta séð sambönd milli service-a
- Þurfum að geta trace-að verkflæði frá einu service í annað
- Einföld útfærsla er að nota Correlation Ids
- Til meira fágaðar / sérhæfðar tracing lausnir er correlation ids

Distributed Tracing

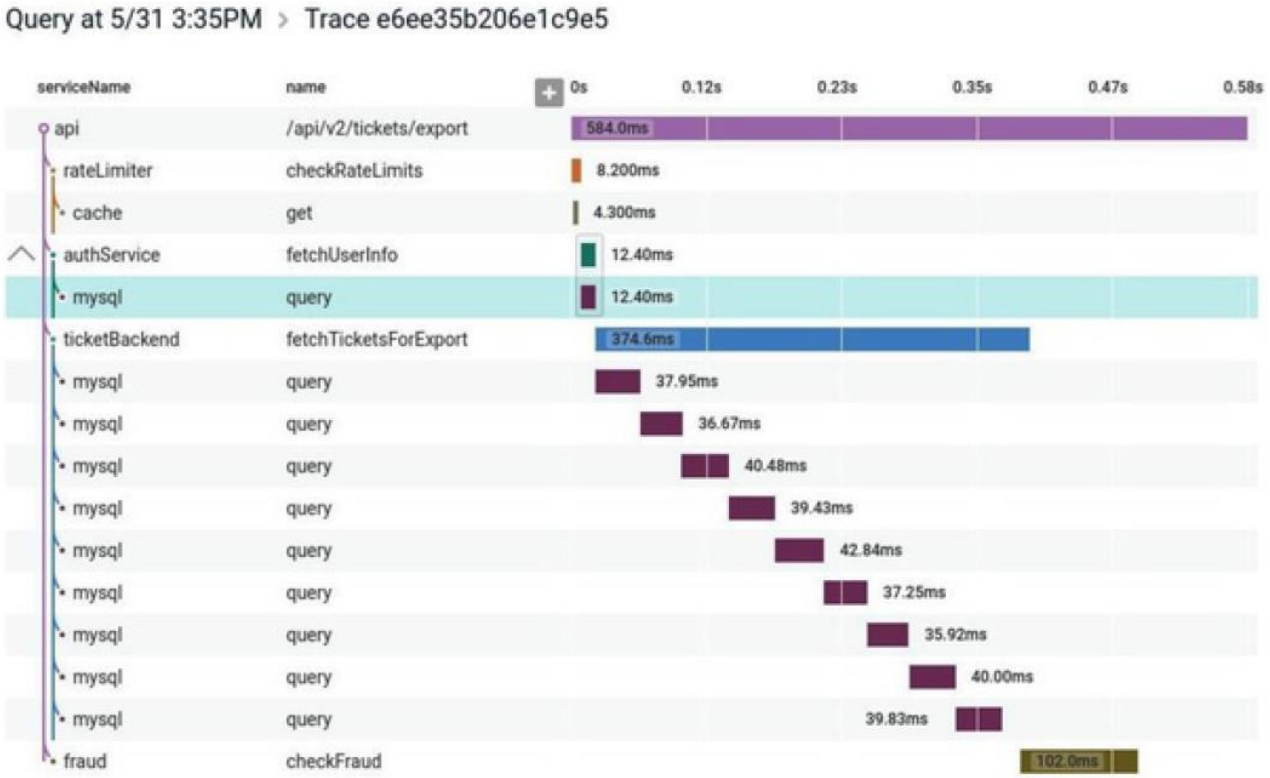


Figure 10-7. A distributed trace shown in Honeycomb, allowing you to identify where time is being spent for operations that can span multiple microservices

Distrubted Tracing

Implementations

- Jaeger
- Honeycomb
- Lightstep
- Datadog
- Elastic

Alerting

- Við þurfum að vita þegar villur eiga sér stað
- Með logging og metric aggregation getum við sett upp alerting kerfi
- Skilgreinum hvenær eitthvað er villa og látum einhvern vita
- T.d. á Slack, Teams, Email, Jira, Webhook
- Hægt að stilla eftir priority

Alerting

The boy who cried wolf

- Sumar villur eru mikilvægari en aðrar
- Ættum að reyna að vanda okkur hvenær við alert-um
- Viljum ekki kæfa þá sem fylgjast með kerfinu með villuskilaboðum
- Alvarlegar villur geta þannig týnst
- Sama vandamál og með flaky test -> The normalization of deviance
- Viljum hafa eitthvað error prioratization

Semantic Monitoring

- Snýst um að fylgjast með hvort kerfið er að hegða sér rétt
- Ekki bara hvort einhver villa var eða hvort performance er nógu gott
- Snýr meira að business hliðinni
- High-level kröfur til kerfisins
- T.d.
  - Nær notanda að skrá sig
  - Erum við að selja x mikið af vörum á klukkutíma

Semantic Monitoring & Testing in Production

Real User Monitoring (RUM)

- Skoðum hvað gerist í raun kerfinu og berum saman við *semantic model*
- Til þess þurfum við t.d. að log-a business information
- Ákveðið form af *testing in production*
- Vandamál við þetta er að við grípum villur ekki fyrr en þær gerast

Semantic Monitoring & Testing in Production

Synthetic Transactions

- Líka þekkt sem *injecting fake data*
- Testum kerfið okkar í production með *fake* gögnum
- Getum keyrt þetta við deployment / release
- Getum keyrt þetta á einhverri dagskrá (t.d. einu sinni á mínútu)
- Þurfum að passa okkur að valda ekki hliðarverkunum
- Líkt end-to-end test og oft geta þau verið notuð

Semantic Monitoring & Testing in Production

Synthetic Transactions - cautionary tale

- *“A friend told me a story about an ecommerce company that accidentally ran its tests against its production ordering systems. It didn’t realize its mistake until a large number of washing machines arrived at the head office.”*

Semantic Monitoring & Testing in Production

A/B Testing

- Með A/B Testing deploy-um við tveimur útfærslum af sama service
- Sjáum síðan hvor service-ið stendur sig betur
- T.d. Tvær útfærslur af *suggestion engine*