



Observable

Spring Boot apps

@ProchazkaFilip 

Q/A: sli.do/shipmonk

From nothing to basic monitoring

You start with absolutely nothing, and then you start adding:

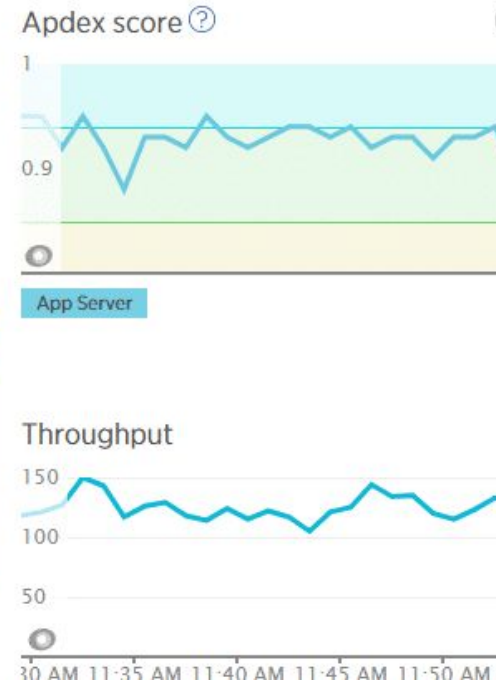
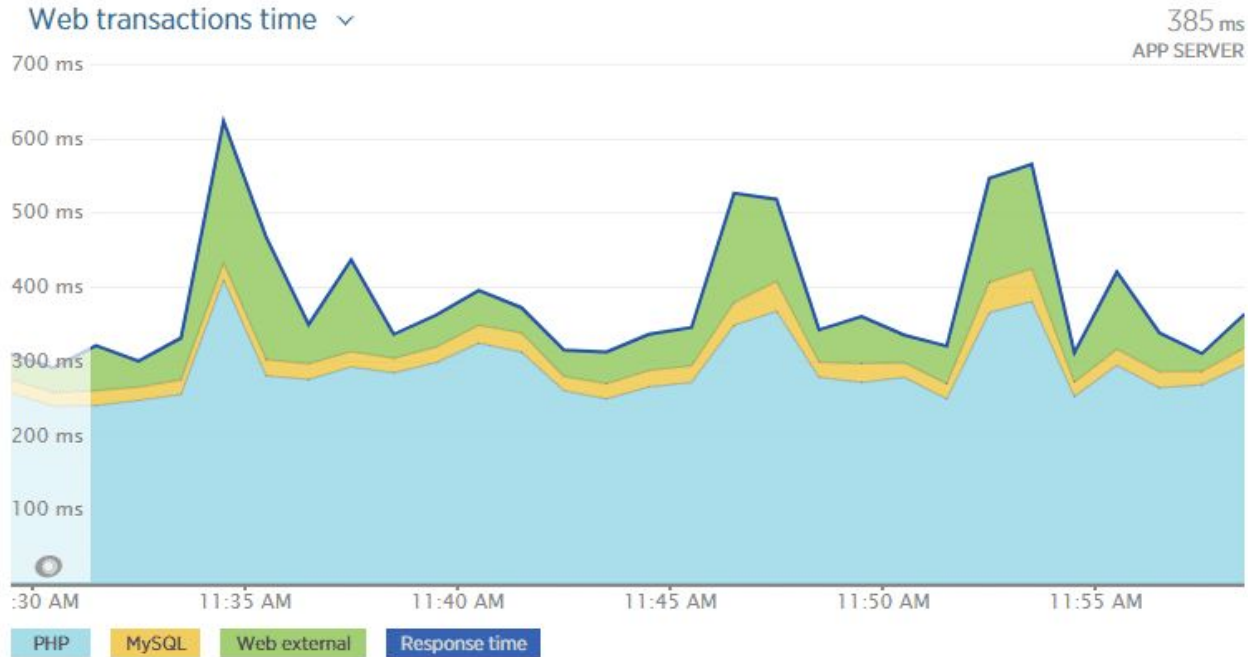
- Error logging (to local filesystem)
- Sending emails on error
- Infrastructure metrics (CPU, RAM, ...)
- Health-checks and availability monitoring
- Sentry
- Centralized logging

Application Performance Monitoring (APM)

- Pick a vendor
- Install their APM library
 - PHP => extension
 - Java => agent
- Start collecting data
- Profit

Application Performance Monitoring (APM)

shipmonk



Application Performance Monitoring (APM)

Slowest average response time ▾

/checkout/onepage/saveBilling	13.8 sec
/catalog/product/view	1,800 ms
/catalog/category/view	1,780 ms
/cms/index/index	1,440 ms
/checkout/cart/add	1,030 ms
/checkout/onepage/index	757 ms
/checkout/cart/index	726 ms
/cms/index/noRoute	313 ms
/checkout/onepage/saveMethod	254 ms
/index.php	247 ms
/core/index/noCookies	210 ms
/checkout/onepage/progress	198 ms
/cms/page/view	176 ms
/checkout/onepage/getAdditional	143 ms

Show all transactions table...

/checkout/onepage/saveBilling

Key transaction > (X)

App performance

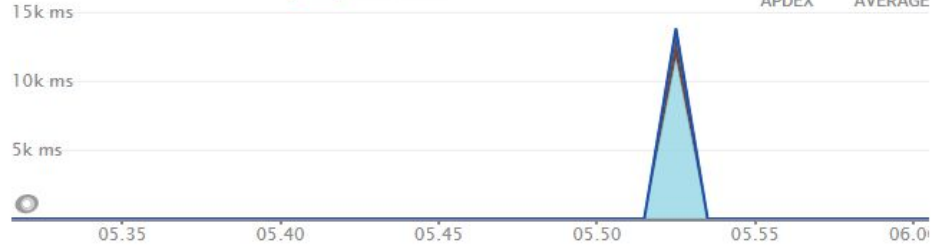
Historical performance

Map ^{Beta}

App server breakdown



0.0 13.8 sec
APDEX AVERAGE



/checkout/onepage/saveBilling

wsbeta.fedex.com

xmipi-ea.dhl.com

MySQL customer_form_attribute select

MySQL other

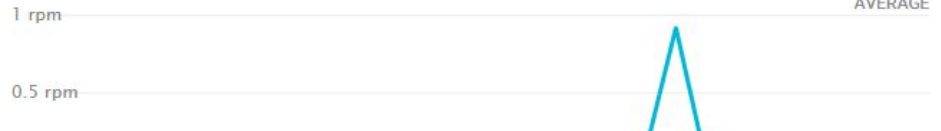
MySQL sales_flat_quote_address update

Other

Response time

Throughput

0.0333 rpm
AVERAGE

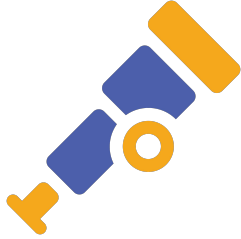


Application Performance Monitoring (APM)

- `newrelic_start_transaction()`
- `newrelic_add_custom_parameter()`
- `newrelic_background_job()`
- ...

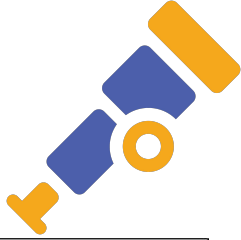
Vendor-lock





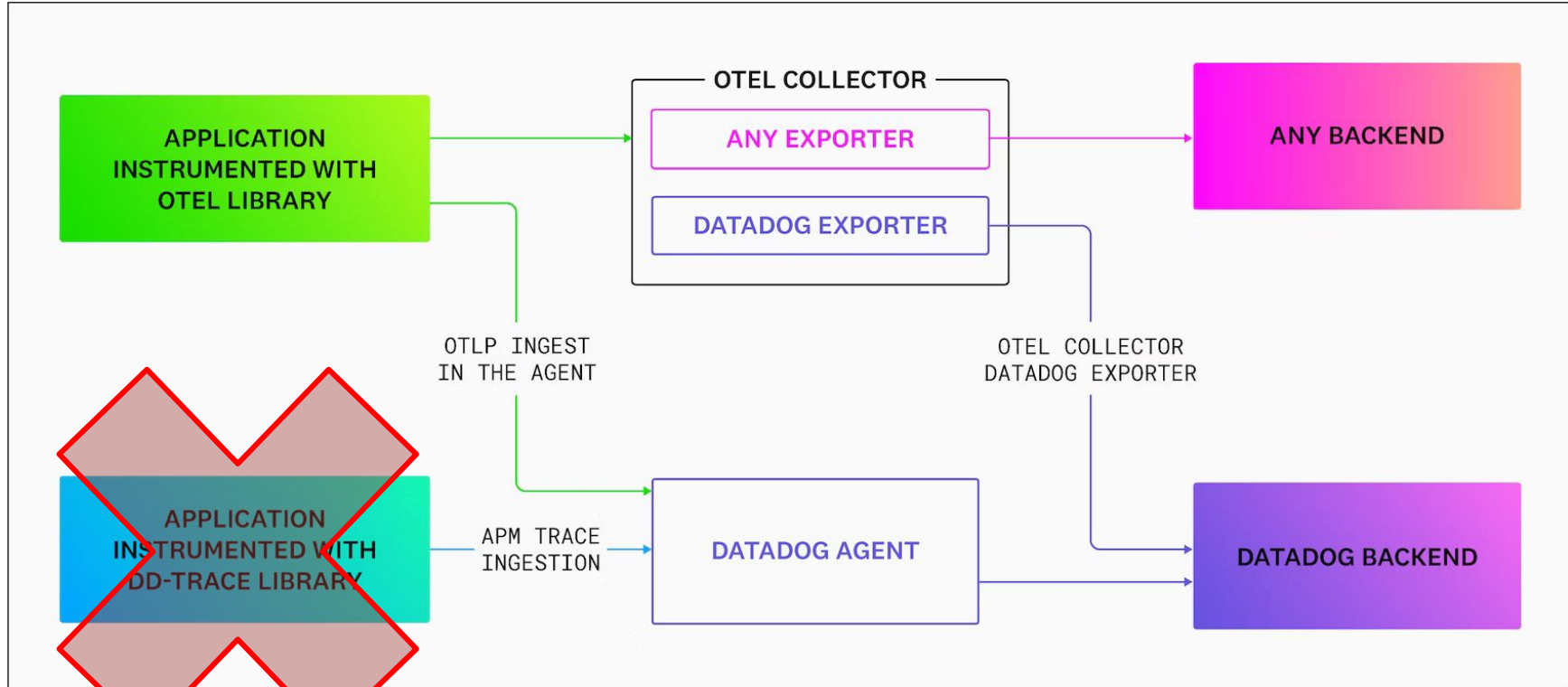
OpenTelemetry to the rescue

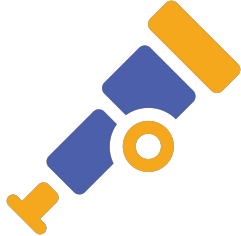
- OpenTracing + OpenCensus
- Vendor-agnostic protocol and libraries
- Instrumentation library (automatic and manual)
- Vendor-neutral collector (and exporter)
- Support for a variety of open source and commercial protocols



DataDog with OpenTelemetry

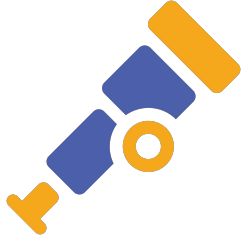
shipmonk





OpenTelemetry with Spring Boot

- Install Java Agent
- Configure
 - Exporter protocol
 - Exporter endpoint
- Automatic instrumentation
 - HTTP client
 - Database
 - Redis
 - Spring Webserver
 - ...



OpenTelemetry

- Signals
 - Traces - detailed information about individual requests
 - Metrics - combine individual measurements into aggregations
 - Logs

Structured logging

2021-07-29 14:54:55.1623 INFO: New report created by user 4253

```
{  
  "TimeStamp": "2021-07-29 14:52:55.1623",  
  "Level": "Info",  
  "Message": "New report created by user 4253"  
}
```

Mapped Diagnostic Context (MDC)

```
log.info("Crawler {} has finished", crawlerId);
```

```
// vs
```

```
try (org.slf4j.MDC.putCloseable("crawlerId", crawlerId)) {  
    log.info("Crawler has finished");  
}
```

Group into Fields Patterns Transactions

Visualize as List Timeseries Top List Table Tree Map Pie Chart



Search facets

Hide Controls 7,958 logs found

Download as CSV More... Options

Showing 147 of 147 + Add

CORE

Index

Source

Host

Service

Status

- Error 570
- Warn 40
- Info 1.28k
- Debug 6.06k

Cluster

Environment

KUBERNETES

Availability zone

Kubernetes Job

Kubernetes Namespace

Pod Name

WAREHOUSEPROCESS

MONGO

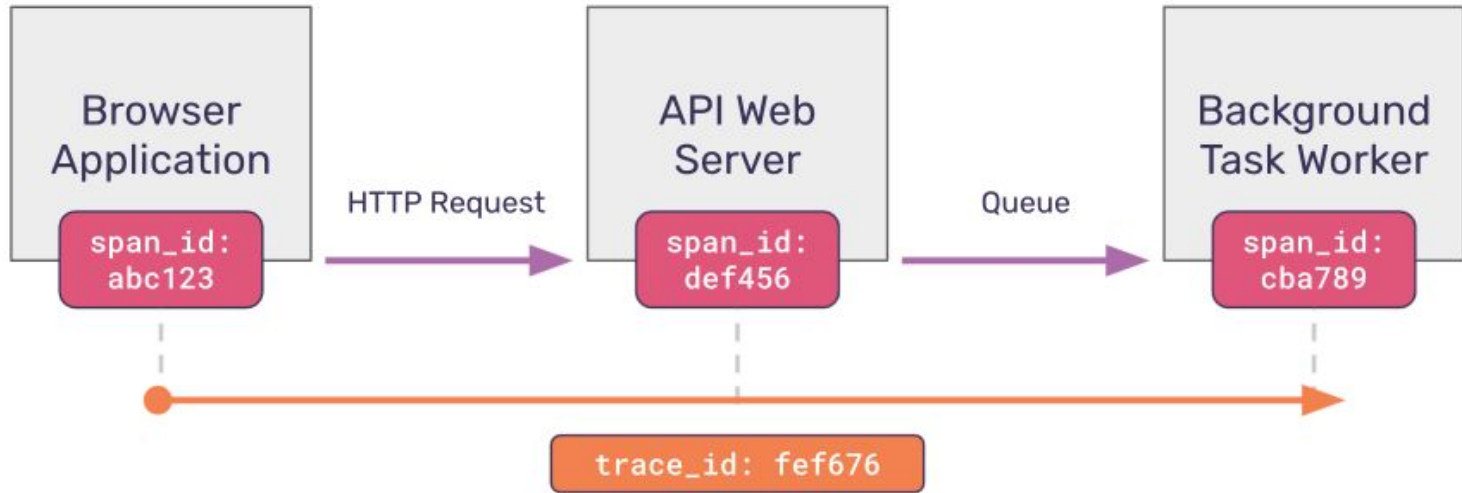
SOURCE CODE

Watchdog Insights

DATE	ENVIRONMENT	@THREAD	@INTEGRATION_TASK_OPERATION	CONTENT
Mar 13 18:30:51.569	dev27761	scheduling-1		Checking for unsent batches of integration api notifications
Mar 13 18:30:51.464	dev27761	scheduling-1		Checking for unsent backend api notifications
Mar 13 18:30:29.201	dev29067	scheduling-1		Checking for unsent batches of integration api notifications
Mar 13 18:30:29.193	dev29067	scheduling-1		Checking for unsent backend api notifications
Mar 13 18:30:25.590	devtest8	scheduling-1		Checking for unsent batches of integration api notifications
Mar 13 18:30:25.580	devtest8	scheduling-1		Checking for unsent backend api notifications
Mar 13 18:30:24.435	dev27761	integration-task-operation-267	MARKETPLACE_PULL_LISTING_PRODUCTS	[e079e33e-e945-4f44-b2f7-1613090b4087][MARKETPLACE_PULL_LISTING_PRODUCTS] task e_
Mar 13 18:30:24.435	dev27761	integration-task-operation-267	MARKETPLACE_PULL_LISTING_PRODUCTS	Skipping product 77 as it was fetched last time already
Mar 13 18:30:24.079	dev27761	integration-task-operation-267	MARKETPLACE_PULL_LISTING_PRODUCTS	Querying listing products updated since 2023-03-13T17:18:51Z
Mar 13 18:30:24.072	dev27761	integration-task-267	MARKETPLACE_PULL_LISTING_PRODUCTS	[e079e33e-e945-4f44-b2f7-1613090b4087][MARKETPLACE_PULL_LISTING_PRODUCTS]: execu_
Mar 13 18:30:24.071	dev27761	scheduling-1		Starting MARKETPLACE_PULL_LISTING_PRODUCTS for e079e33e-e945-4f44-b2f7-1613090b4_
Mar 13 18:30:17.097	dev29154	scheduling-1		Checking for unsent batches of integration api notifications
Mar 13 18:30:17.090	dev29154	scheduling-1		Checking for unsent backend api notifications
Mar 13 18:30:06.385	dev28763	scheduling-1		Checking for unsent batches of integration api notifications
Mar 13 18:30:06.378	dev28763	scheduling-1		Checking for unsent backend api notifications
Mar 13 18:30:04.917	dev29141	scheduling-1		Checking for unsent batches of integration api notifications
Mar 13 18:30:04.600	dev29141	scheduling-1		Checking for unsent backend api notifications
Mar 13 18:30:04.525	dev27761	integration-task-operation-266	MARKETPLACE_PULL_ORDERS	[e079e33e-e945-4f44-b2f7-1613090b4087][MARKETPLACE_PULL_ORDERS] task ended
Mar 13 18:30:04.525	dev27761	integration-task-operation-266	MARKETPLACE_PULL_ORDERS	Skipping order 219 as it was fetched last time already
Mar 13 18:30:04.072	dev27761	integration-task-operation-266	MARKETPLACE_PULL_ORDERS	Querying orders updated since 2023-03-13T14:00:51Z
Mar 13 18:30:04.065	dev27761	integration-task-266	MARKETPLACE_PULL_ORDERS	[e079e33e-e945-4f44-b2f7-1613090b4087][MARKETPLACE_PULL_ORDERS]: executing integ_
Mar 13 18:30:04.064	dev27761	scheduling-1		Starting MARKETPLACE_PULL_ORDERS for e079e33e-e945-4f44-b2f7-1613090b4087
Mar 13 18:30:03.565	dev29044	scheduling-1		Checking for unsent batches of integration api notifications
Mar 13 18:30:03.495	dev29044	scheduling-1		Checking for unsent backend api notifications

Distributed tracing

- Span - a unit of work or operation
- Trace - path of a request, through spans

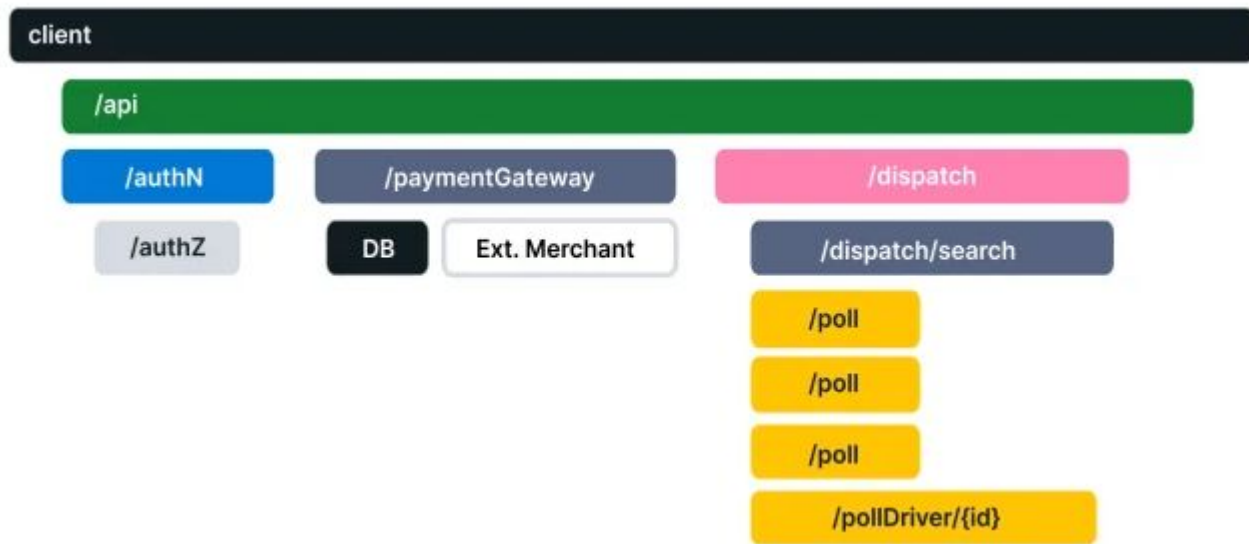


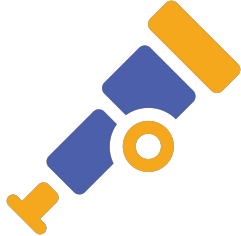
Distributed tracing

```
{
  "name": "Hello",
  "context": {
    "trace_id": "0x5b8aa5a2d2c872e8321cf37308d69df2",
    "span_id": "0x93564f51e1abe1c2"
  },
  "parent_id": "0x051581bf3cb55c13",
  "start_time": "2022-04-29T18:52:58.114492Z",
  "end_time": "2022-04-29T18:52:58.114631Z",
  "attributes": {
    "http.route": "some_route2"
  }
}
```

Distributed tracing

- Trace - path of a request, through spans
- Distributed trace - trace across multiple services





Manual instrumentation - spans

```
var tracer = openTelemetry
    .getTracer(BusinessOperation.class.getSimpleName());
```

```
Span span = tracer.spanBuilder("name")
    .setAttribute("user", userId)
    .startSpan();
```

```
try (Scope ss = span.makeCurrent()) {
    // In this scope, the span is the current/active span
} finally {
    span.end();
}
```

↓ DATE	SERVICE	RESOURCE	DURATION	METHOD	STATUS CODE	LATENCY BREAKDOWN
Mar 14 01:15:10.150	shipmonk-account-integrations	Notification BACKEND_API	93.6 ms			
Mar 14 01:15:10.056	shipmonk-account-integrations	Notification BACKEND_API	91.1 ms			
Mar 14 01:15:09.965	shipmonk-account-integrations	Notification BACKEND_API	96.0 ms			
Mar 14 01:15:09.869	shipmonk-account-integrations	Notification BACKEND_API	106 ms			
Mar 14 01:15:09.763	shipmonk-account-integrations	Notification BACKEND_API	119 ms			
Mar 14 01:15:09.643	shipmonk-account-integrations	Notification BACKEND_API	108 ms			
Mar 14 01:15:09.536	shipmonk-account-integrations	Notification BACKEND_API	108 ms			
Mar 14 01:15:09.427	shipmonk-account-integrations	Notification BACKEND_API	140 ms			
Mar 14 01:14:42.990	shipmonk-account-integrations	Notification INTEGRATION_API	5.86 s			
Mar 14 01:14:17.275	shipmonk-account-integrations	Notification BACKEND_API	102 ms			
Mar 14 01:14:17.172	shipmonk-account-integrations	Notification BACKEND_API	117 ms			
Mar 14 01:14:17.055	shipmonk-account-integrations	Notification BACKEND_API	97.2 ms			
Mar 14 01:13:36.911	shipmonk-account-integrations	Notification INTEGRATION_API	1.03 s			
Mar 14 01:13:08.962	shipmonk-account-integrations	Notification BACKEND_API	163 ms			
Mar 14 01:12:22.637	shipmonk-account-integrations	Scheduled MARKETPLACE_PULL_ORDERS	1.05 s			
Mar 14 01:11:29.087	shipmonk-account-integrations	Notification INTEGRATION_API	567 ms			
Mar 14 01:10:54.559	shipmonk-account-integrations	Scheduled MARKETPLACE_PULL_ORDERS	1.21 s			
Mar 14 01:10:43.714	shipmonk-account-integrations	Scheduled MARKETPLACE_PULL_ORDERS	2.49 s			
Mar 14 01:10:27.247	shipmonk-account-integrations	Notification INTEGRATION_API	1.07 s			
Mar 14 01:09:56.068	shipmonk-account-integrations	IntegrationTaskScheduler.garbageCollect..	2.09 ms			
Mar 14 01:09:48.104	shipmonk-account-integrations	IntegrationTaskScheduler.acquireLockAnd..	15.1 s			
Mar 14 01:09:18.014	shipmonk-account-integrations	IntegrationTaskScheduler.garbageCollect..	1.39 ms			
Mar 14 01:08:43.035	shipmonk-account-integrations	Scheduled MARKETPLACE_PULL_LISTING_PROD..	2.22 s			
Mar 14 01:08:42.050	shipmonk-account-integrations	Scheduled MARKETPLACE_PULL_LISTING_PROD..	1.24 s			
Mar 14 01:08:25.898	shipmonk-account-integrations	UnsentIntegrationApiNotificationEnqueue..	10.2 s			
Mar 14 01:08:22.876	shipmonk-account-integrations	Scheduled BACKEND_PULL_LISTING_PRODUCT_..	37.5 ms			

Span: Info Infrastructure Metrics Logs 0 Network Processes Code Hotspots

```

+ container {...}
  duration 1219140709
  env prod
+ host {...}
  integration_api_notification_ids ["67c0722d-444a-405b-b2a3-3cc7ac37cfb0"]
  integration_api_operation LISTING_PRODUCT_QUANTITY_UPDATED
  integration_id
  language
  notification_type
+ os {...}
+ otel {...}
+ process {...}
  release
+ telemetry {
  + auto {
    version 1.22.1
  }
  + sdk {
    language java
  }
}
    
```

- Filter by @integration_id:70bf16c6-e7db-4e...73f8
- Exclude @integration_id:70bf16c6-e7db-4e...73f8
- Replace filter with @integration_id:70bf16c6-e7db-4e...73f8
- Add column for @integration_id
- Create facet for @integration_id

Span/log correlation in DataDog

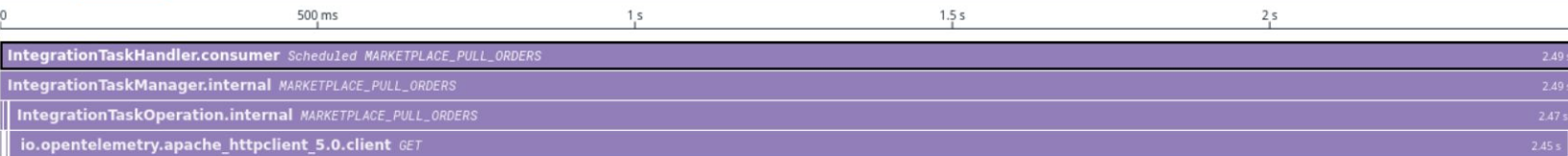
- OTel logs
 - For Java is experimental
 - For PHP is non-existent
- We're collecting logs using Docker/K8S native DD integration
 - It scrapes docker container stdout and forwards it to DD
- [Connect OpenTelemetry Traces and Logs](#)

🕒 2.49 s total duration p92

on Mar 14 01:10:41.226 (9m ago)

Trace: Flame Graph Span List 23 Map

[Hide Legend](#)



Service	% Exec Time
shipmonk-acco...	100%



🕒 2.49 s p92 (100% of total duration)

Span: Info Infrastructure Metrics Logs 3 Network Processes Code Hotspots

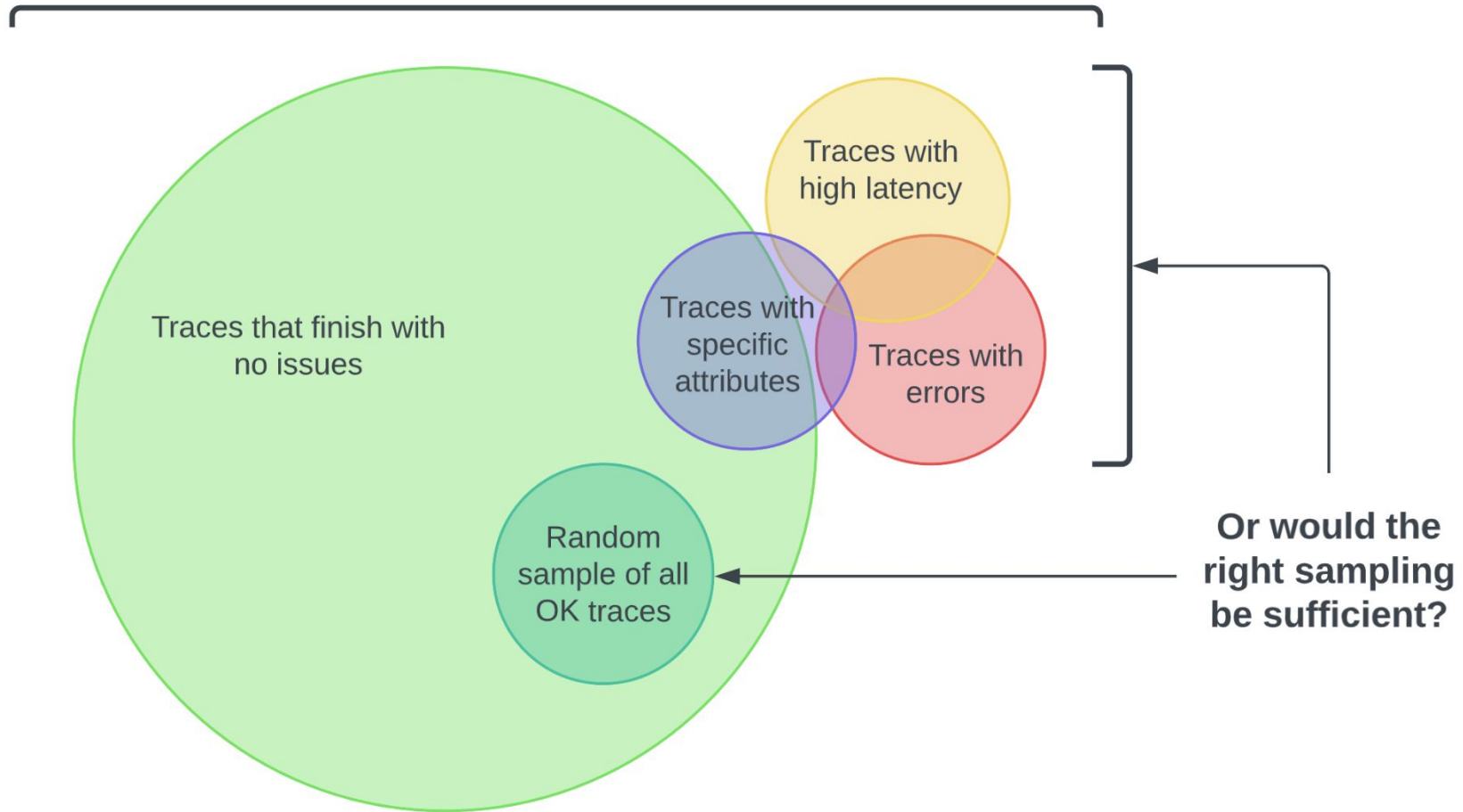
🔍 Trace ID trace_id:1698805

↑ DATE	SERVICE	HOST
Mar 14 01:10:41.226	account-integrations	1-03358b59
[855d6cf4-]	[MARKETPLACE_PULL_ORDERS]: executing integration task operation
Mar 14 01:10:41.246	account-integrations	1-03358b59
Querying orders updated since 2023-03-13T20:03:01Z		
Mar 14 01:10:43.710	account-integrations	1-03358b59
[855d6cf4-]	[MARKETPLACE_PULL_ORDERS] task ended

Tail Sampling in DataDog

- Storing all Traces is a lot of data
- DataDog is *very* expensive

Do you really need all of this data?

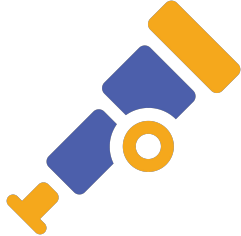


Tail Sampling in DataDog

- The cost can be reduced by *not* saving/indexing everything
- How do you know what **not** to drop?
 - Configurable, e.g.:
 - Send everything to DD
 - Index everything matching a filter (spans with error)
 - All other traces - keep only few %

If APM is sampled, how do we graph real numbers?

- **RED metrics** are always accurate (even with sampling)
 - requests, errors, duration
- Everything else with **custom metrics!**
- Be careful: attribute cardinality increases **cost** :’(ul>- OrderId => high cardinality
- Small enum => low cardinality



Manual instrumentation - metrics

```
// Gets or creates a named meter instance  
var meter = openTelemetry  
    .getMeter(BusinessOperation.class.getSimpleName());  
  
// Build counter e.g. LongCounter  
LongCounter counter = meter  
    .counterBuilder("processed_jobs")  
    .setDescription("Processed jobs")  
    .setUnit("1")  
    .build();
```

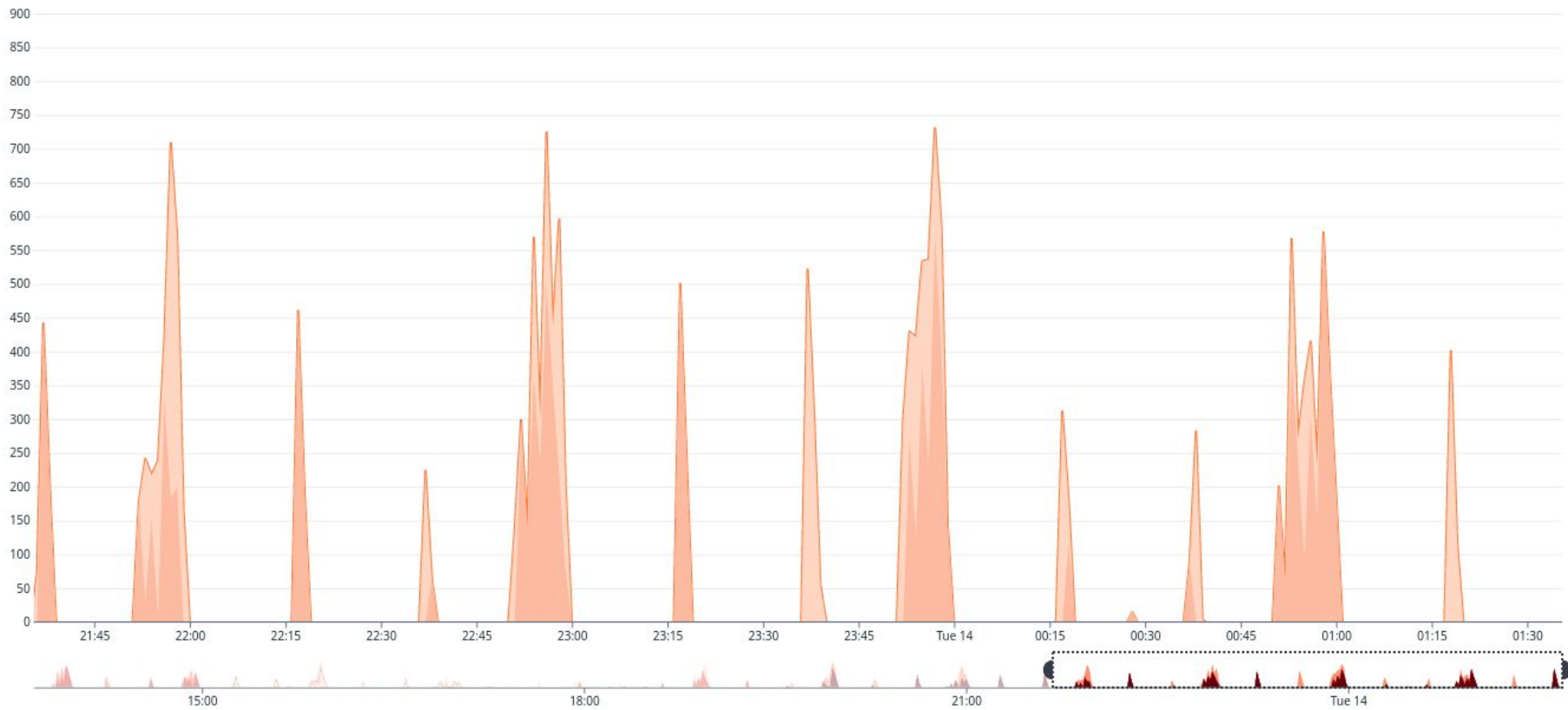


Manual instrumentation - metrics

```
Attributes attributes = Attributes.of(  
    stringKey("Key"),  
    "SomeWork"  
);  
  
// Record data  
counter.add(123, attributes);
```

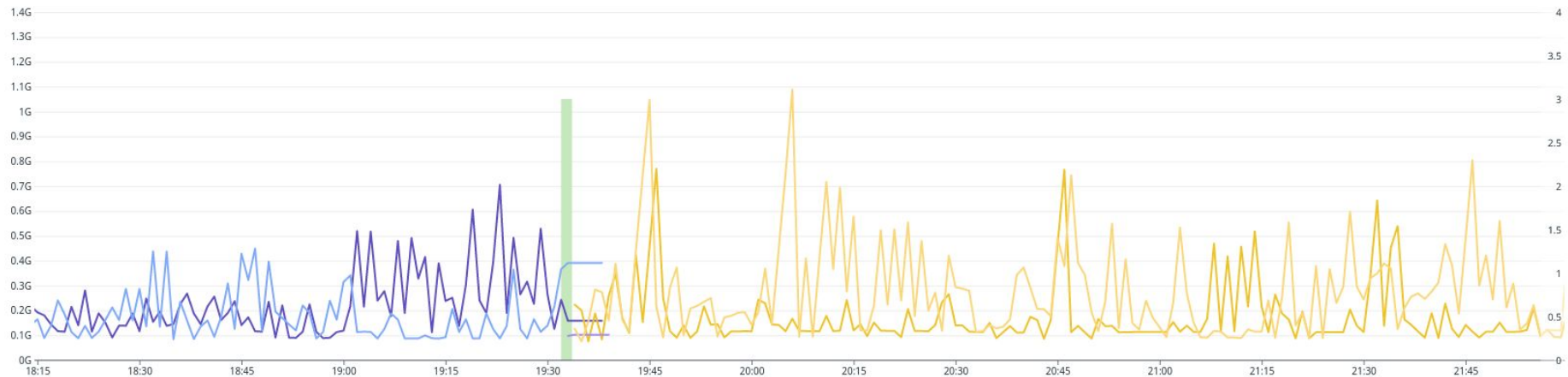
Integration API notifications queue size

[Save to Dashboard](#) [More...](#)



Filter series

container_id in integration_api.queue_size	Avg	Min	Max	Sum	Value
3f8a1f7bd067f9367da330d2bb06d8b6381082f002dcb0b26fc8f2875d4df07	31.4	0	524	7.54k	181.0
c3cb3f2fa4bd7fdc15bb995b63d9bb39612485892f013f811006428d9c9d8c2	43.1	0	578	10.34k	406.0



1 Select your visualization

- Timeseries**
- Query Value
- Top List
- Table
- Tree Map
- Pie Chart
- Distribution
- Heatmap
- Geomap
- Scatter Plot
- Change
- Host Map

2 Graph your data

Edit JSON Share

[Graphing help](#)

a Logs
Count * group by (everything) roll up every 2m (auto) Σ

+ Add Query + Add Formula

Display: Bars Color: Classic Y-axis: Right

a Metrics from
max by area * container_id * Σ

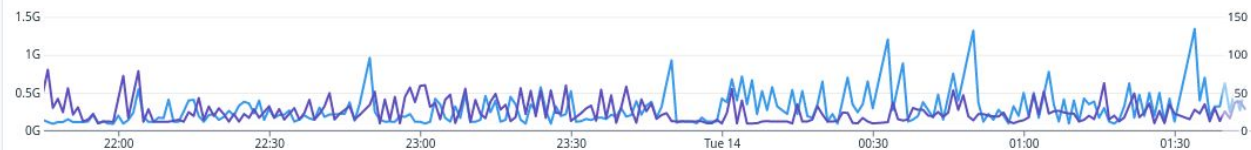
+ Add Query + Add Formula

Display: Lines Color: Classic Style: Solid Stroke: Normal Y-axis: Left

Platform container memory limit/usage



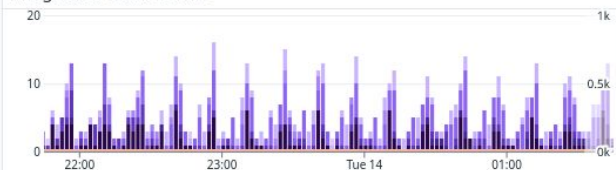
Max used heap memory per container + app starts



JVM threads count



Integration tasks counts



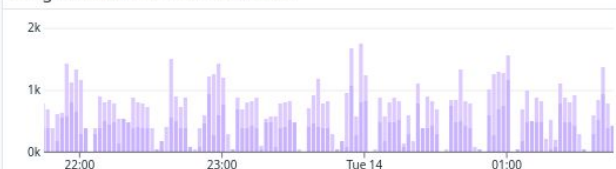
Integration task: items processing counts



Task executor queue size (other)



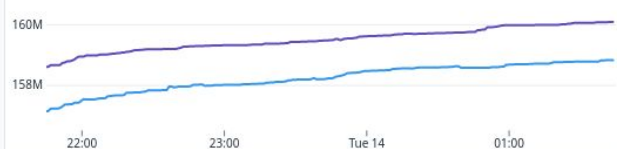
Integration API notification counts



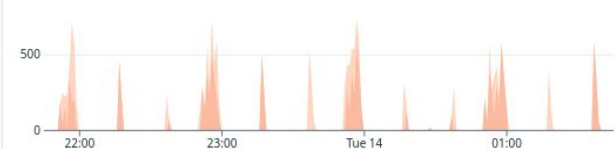
Backend API notification counts



Max used non-heap memory per container



Integration API notifications queue size



Backend API notifications queue size



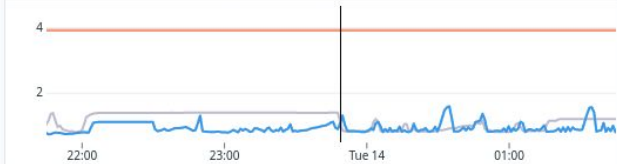
JVM GC pauses



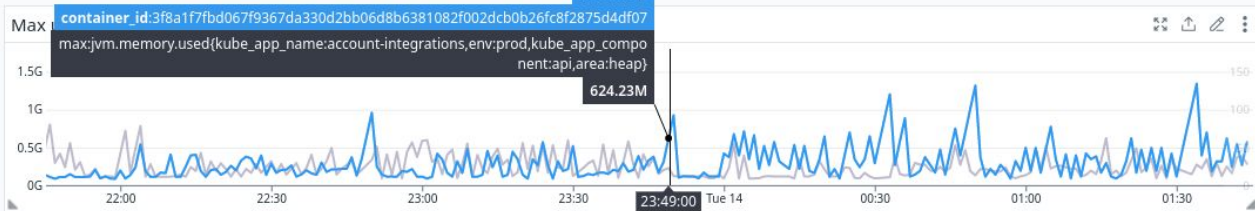
HTTP IN: requests by APM



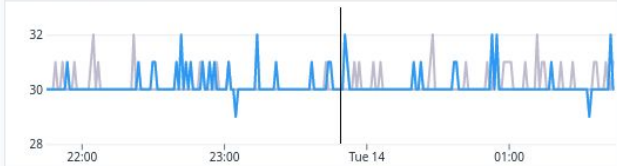
Platform container memory limit/usage



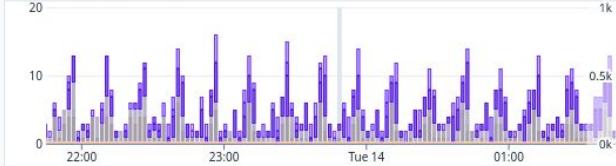
area:heap



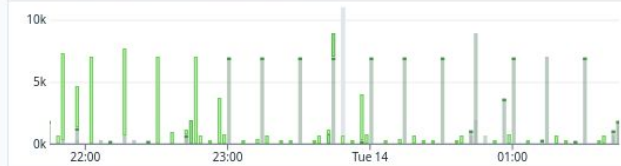
JVM threads count



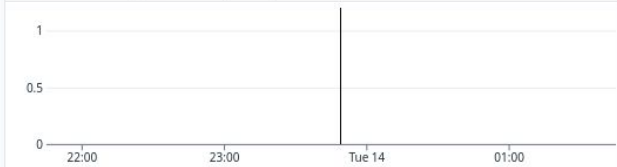
Integration tasks counts



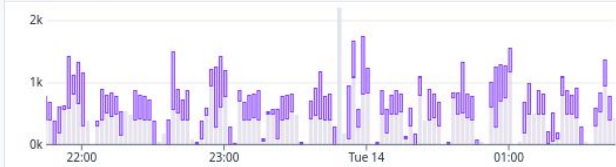
Integration task: items processing counts



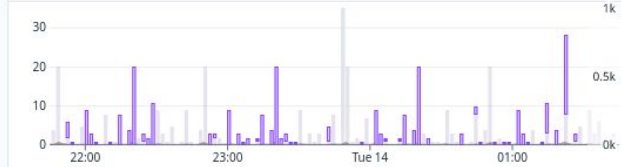
Task executor queue size (other)



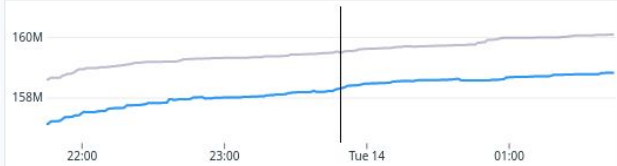
Integration API notification counts



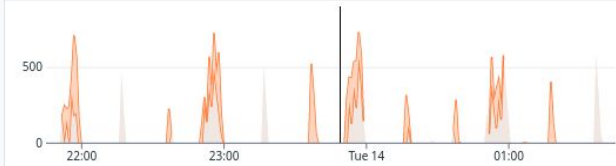
Backend API notification counts



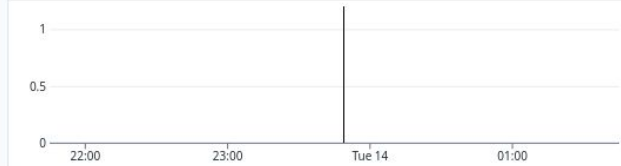
Max used non-heap memory per container



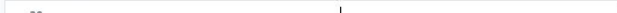
Integration API notifications queue size



Backend API notifications queue size



JVM GC pauses



HTTP IN: requests by APM





1 Choose the detection method

Threshold Alert Change Alert Anomaly Detection Outliers Alert Forecast Alert

An alert is triggered whenever a metric crosses a threshold. ?

2 Define the metric

Source Edit

a aws.rds.cpuutilization from dbinstanceidentifier × IN (prod-aurora-postgresql ×) avg by dbinstanceidentifier × Σ

</>

+ Add Query + Add Formula

Multi Alert ▼ Trigger a separate alert for each dbinstanceidentifier × ▼ reporting your metric ?

Evaluate the average ▼ of the query over the last 30 minutes ▼

Note: The selected metric provides data at low frequencies. We strongly recommend alerting on at least a 1 hour time frame.

3 Set alert conditions

Trigger when the evaluated value is above ▼ the threshold for any dbinstanceidentifier

Alert threshold: > 90 (90 %)

Warning threshold: > Optional



OpenTelemetry

- App is auto-instrumented using OTel
- We're able to create custom spans/metrics
- We're able to enhance spans and metrics with attributes
- We're able to enhance logs with MDC
- We have solved span/log correlation

Do we finally have **Observability**?

Observability: the power to ask new questions of your system, without having to ship new code or gather new data in order to ask those new questions.

Monitoring is about known-unknowns and actionable alerts, observability is about unknown-unknowns and empowering you to ask arbitrary new questions and explore where the cookie crumbs take you.

Observability means you can understand how your systems are working on the inside just by asking questions from outside.

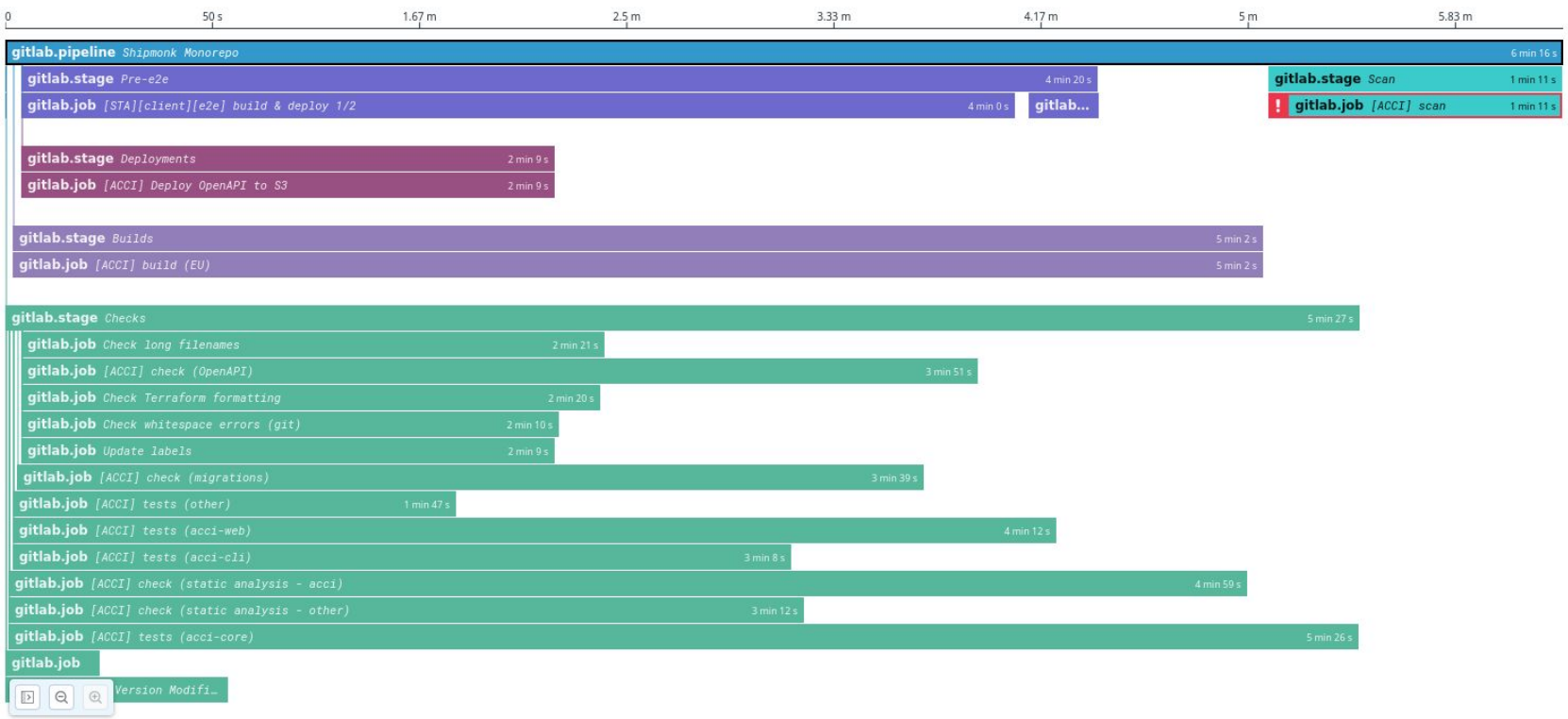
... so maybe?

Bonus tips:

- Switching OTel exporters is easy
 - On localhost, you can develop with e.g. Jaeger
 - Awesome for highly async apps
- Monitor jobs in your CI pipeline using APM
 - The same way you monitor production code

Show as: Flame Graph Span List

Hide Legend



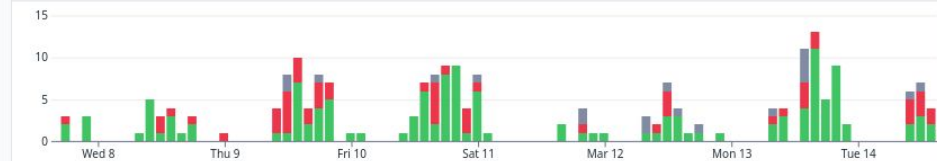
Stage % Exec Time

- Checks 63.4%
- Scan 16.0%
- Builds 11.1%
- Pre-e2e 7.32%
- Deployments 2.24%
- PostDeployments

Filter Spans

Errors

Pipelines by status



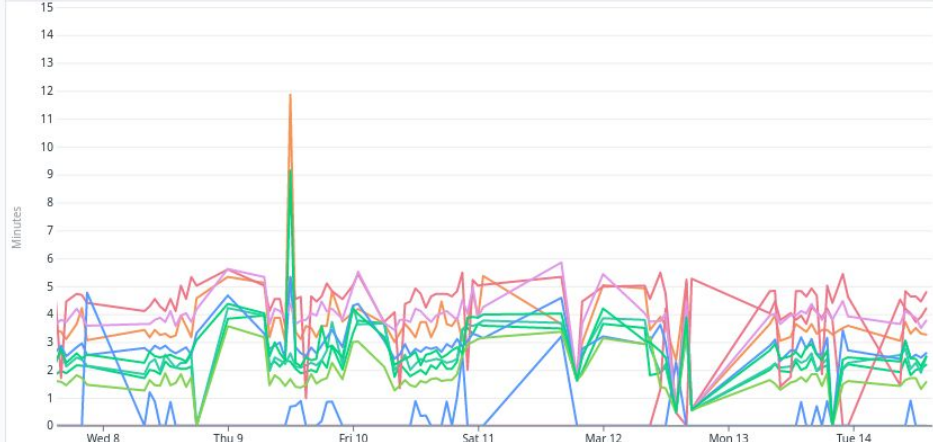
Job duration by status (master)



Job duration by status (MRs)



Job duration by name

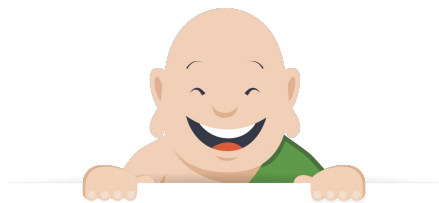


@ci.job.name in pc95:@duration@ci_level:job @ci.pipeline.name:"shipmonk-company/shipmonk-fulfilment/s...

	↓ Avg	Min	Max
[ACCI] check (static analysis - acci)	4.28 min	31.0 s	5.60 min
[ACCI] tests (acci-core)	3.94 min	30.2 s	5.85 min
[ACCI] build (EU)	3.68 min	35.0 s	11.86 min
[ACCI] check (static analysis - other)	2.84 min	0 s	5.33 min
[ACCI] tests (acci-web)	2.74 min	0 s	9.15 min
[ACCI] check (OpenAPI)	2.44 min	0 s	4.28 min
[ACCI] check (migrations)	2.24 min	0 s	3.95 min
[ACCI] tests (acci-cli)	1.79 min	0 s	3.57 min
[ACCI] check (OpenRewrite)	1.00 min	0 s	5.27 min
[ACCI] scan latest	0.32 min	0 s	4.77 min




Add Widgets or Powerpacks



Questions?

sli.do/shipmonk

A man with long brown hair, a full beard, and glasses is sitting at a desk in an office. He is wearing a dark jacket and has a lanyard around his neck. He is looking upwards and to the right with a thoughtful expression. The background shows office shelves filled with various items, including boxes and papers.

how do i get to the...

@ProchazkaFilip