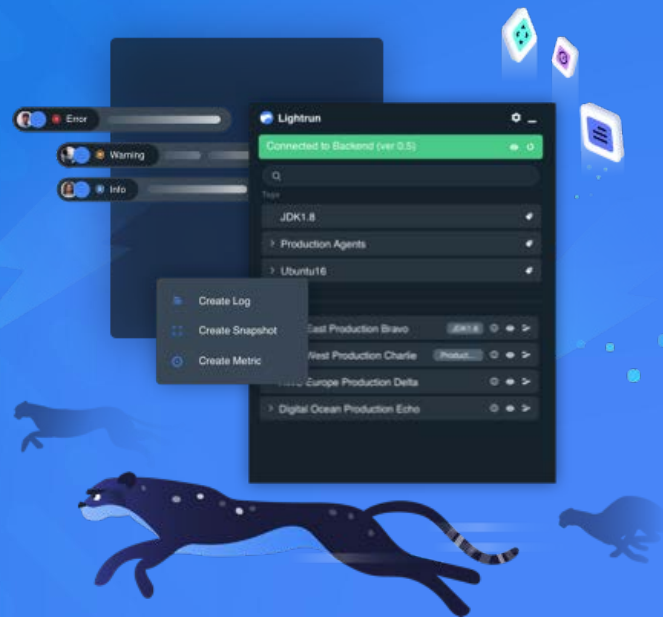




Debug NodeJS Applications in Production with Lightrun

Shai Almog
October 2021



```
Speaker ShaiAlmog = Speaker.builder()
    .withRoles(
        createDeveloperAdvocateAt("Lightrun"),
        createCoFounderAt("Codename One"))
    .withProfessionalExperience(30, TimeUnit.YEARS)
    .withTopCompanies("Sun", "Oracle", "Codename One", "Lightrun")
    .withTwitter("twitter.com/debugagent")
    .withBlog("talktotheduck.dev")
    .withEmail("shaia@lightrun.com")
    .withGitHub("github.com/shai-almog")
    .build();
```



I APMs

Real life

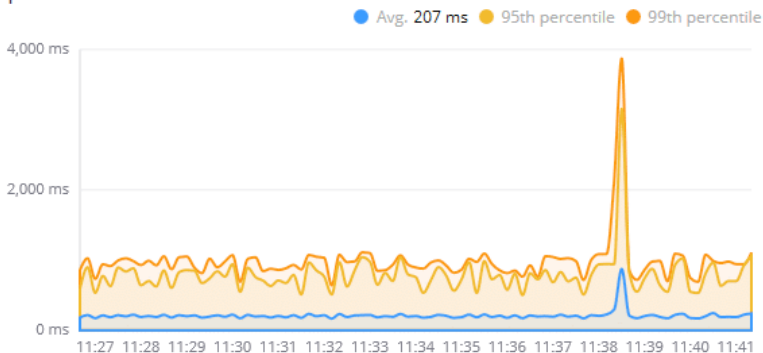


TicketMonster

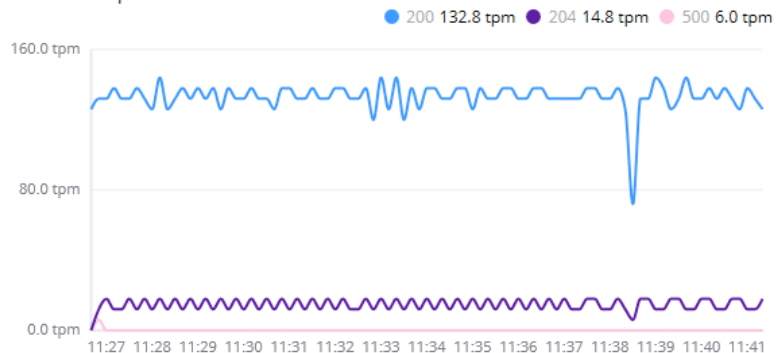
Request

Errors

Response times



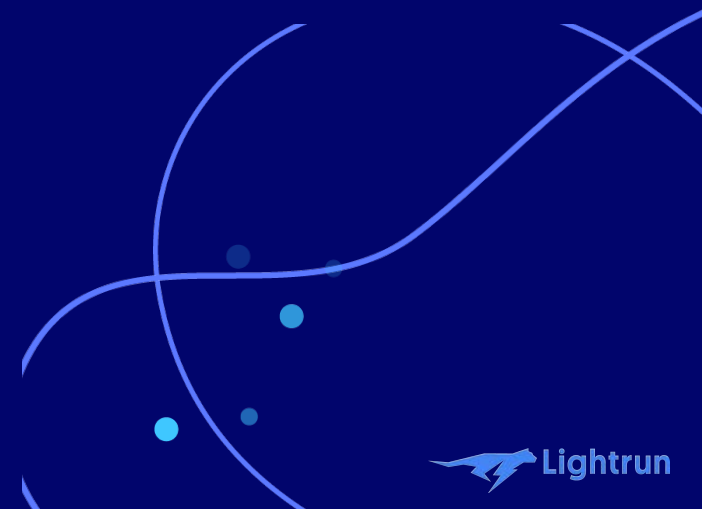
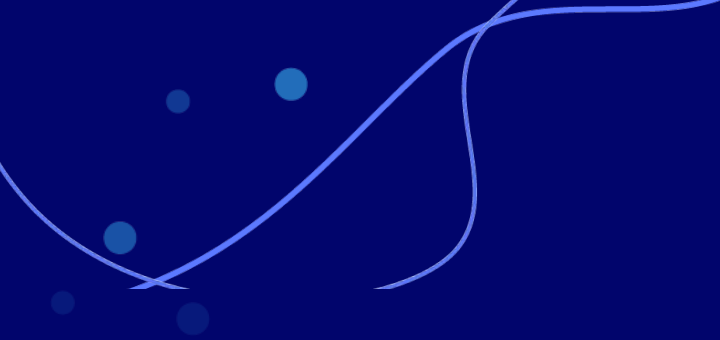
Transactions per minute

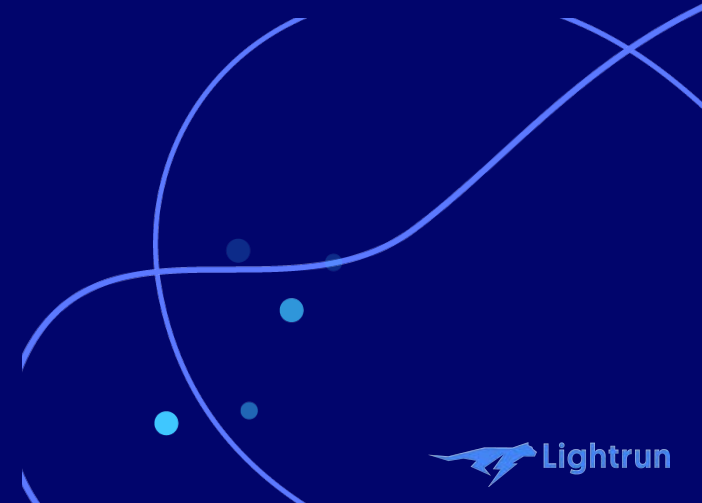
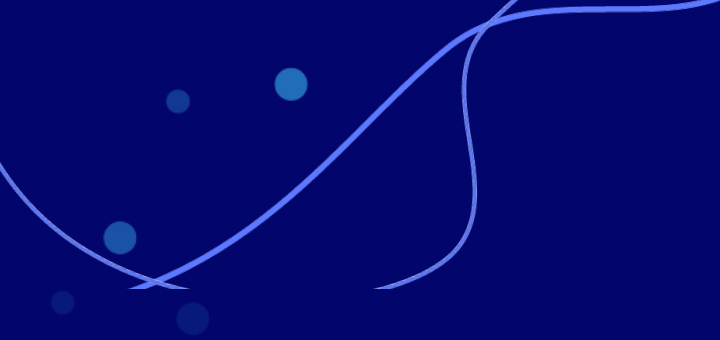


Request

Filter...

Name	Avg. resp. time	95th percentile	TPM	Impact ⓘ ↓
GET /ticket-monster/rest/bookings	879 ms	1,102 ms	14.8 tpm	<div style="width: 100%; height: 10px; background-color: #0070c0;"></div>
GET /ticket-monster/rest/shows	126 ms	127 ms	29.6 tpm	<div style="width: 75%; height: 10px; background-color: #0070c0;"></div>
GET /ticket-monster/rest/venues	155 ms	130 ms	14.8 tpm	<div style="width: 50%; height: 10px; background-color: #0070c0;"></div>
GET /ticket-monster/rest/venues/:venueId	140 ms	130 ms	14.8 tpm	<div style="width: 50%; height: 10px; background-color: #0070c0;"></div>







Developers



Developers



Devops





Developers



Devops





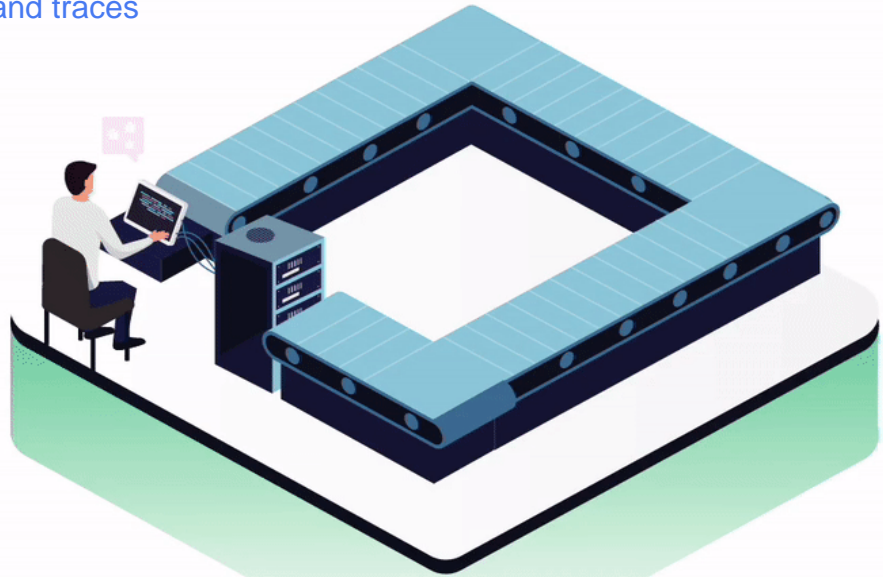


The Needle isn't
Necessarily Here

Production issues in 2021

1

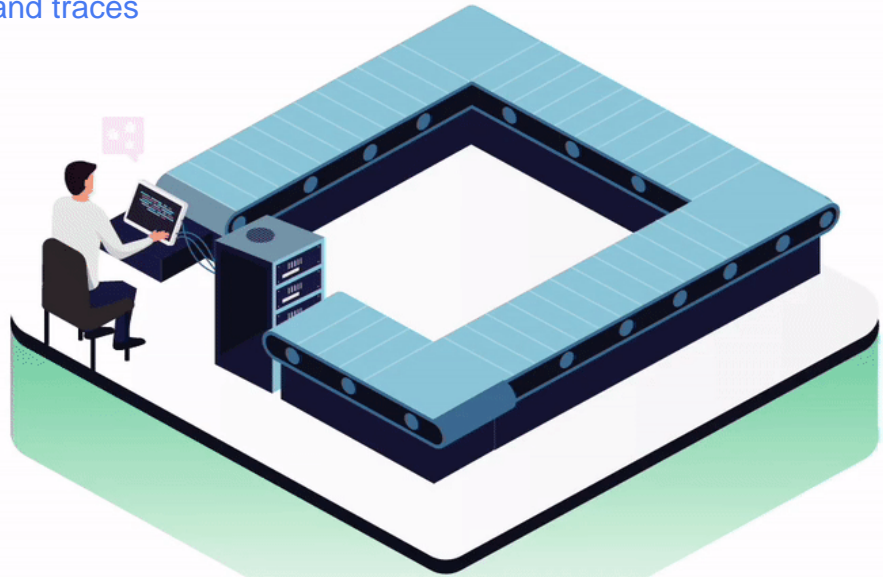
Add new logs,
metrics and traces



Production issues in 2021

1

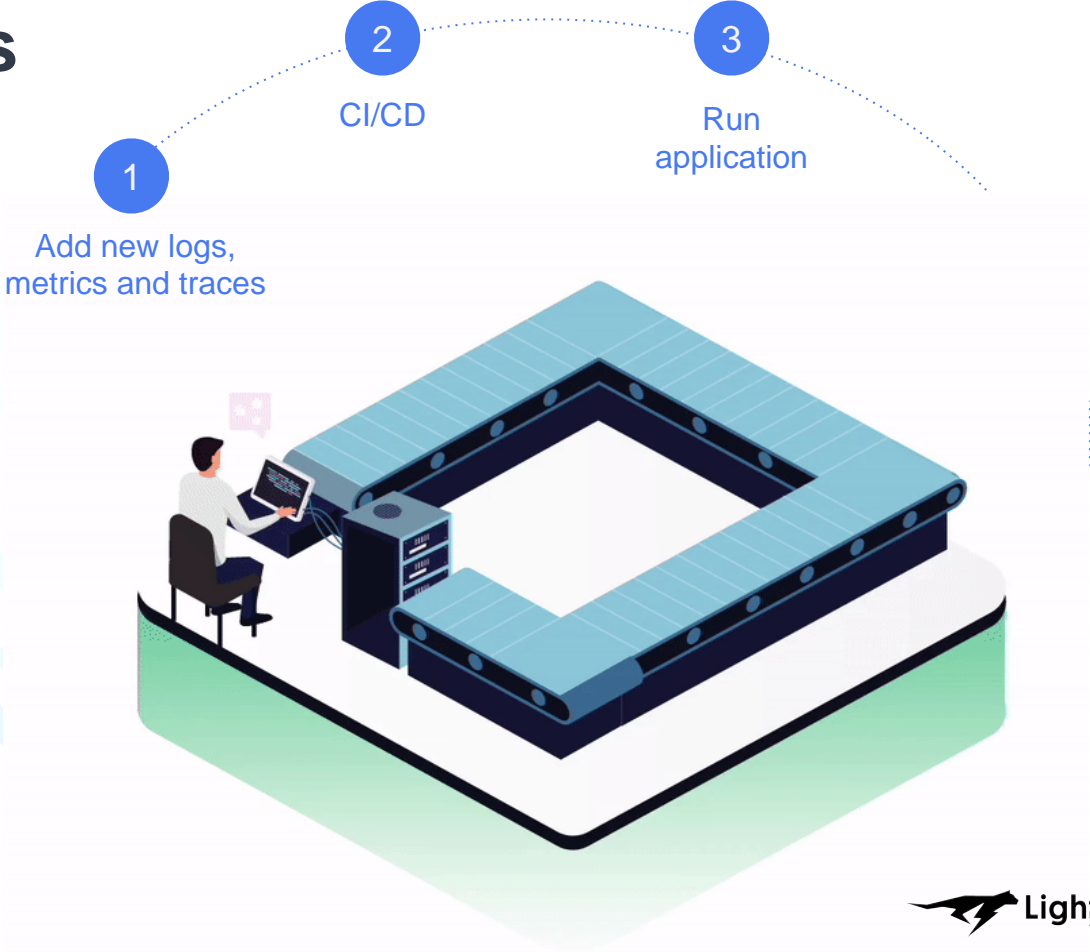
Add new logs,
metrics and traces



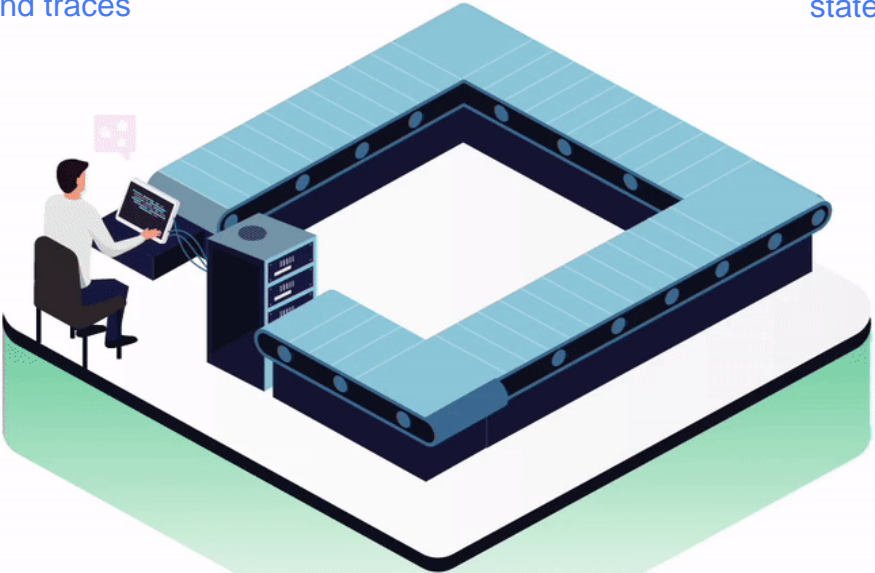
Production issues in 2021



Production issues in 2021



Production issues in 2021





Why local development for serverless is an anti-pattern



Gareth McCumskey • Jun 2





Vlad Mihalcea @vlad_mihalcea · Sep 20



Replying to [@debugagent](#) [@nicolas_frankel](#) and 15 others

Cloud providers can only love microservice architectures.



Vlad Mihalcea @vlad_mihalcea · May 16

The more microservices you need to deploy, the happier the cloud provider.



There has to be a Better
Way!

We don't know what
we'll run into

If Only we Could
Debug...

But Debuggers aren't
the Right Tool

They Can't Cross Server Boundaries

They can't handle
Different Languages





Developers



Devops



Developers



Devops

Production Debugging



Continuous Observability
Toolbox

How it Works

1

The developer uses Lightrun's IDE plugin to add an action in `example.java` line 100



Management Server



Service Running with Lightrun's Agent



per

How it Works

1

The developer uses Lightrun's IDE plugin to add an action in `example.java` line 100



Management Server



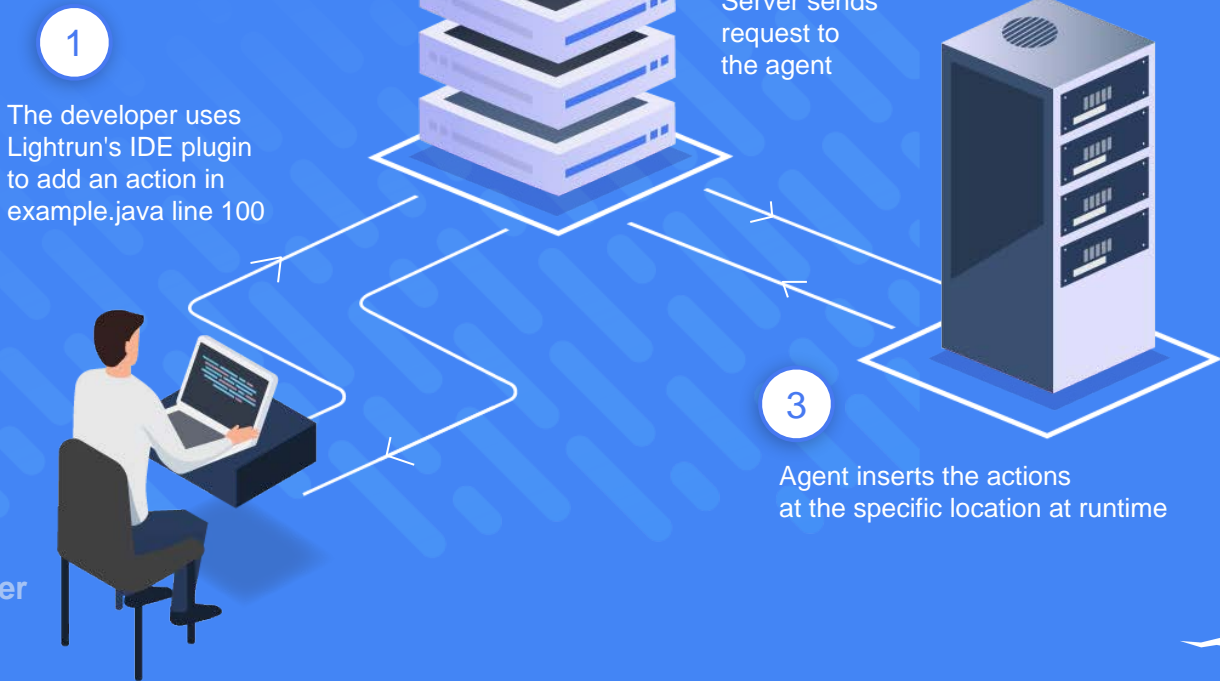
2

Management Server sends request to the agent

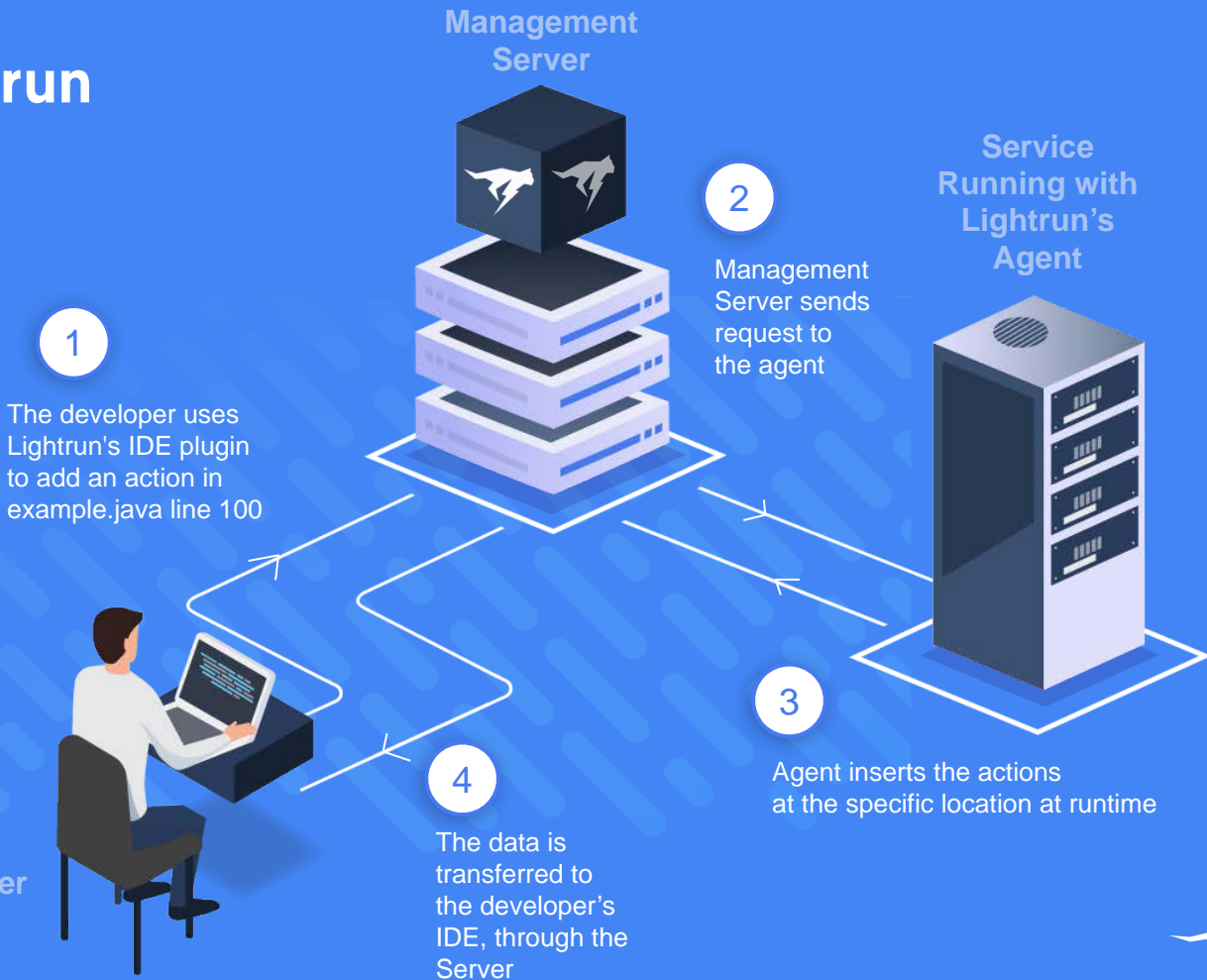
Service Running with Lightrun's Agent



How it Works



How it Works



```
1  import express from "express";
2
3  const app = express();
4
5  app.get("/", (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
6      res.send( body: "Hello World!");
7  });
8
9  app.get("/Hello", (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
10     res.send( body: "World!");
11 });
12
13 /*
14  PORT could be set via environment variable (e.g. by Google Cloud
15  or some other cloud service). Fallback to your desired PORT.
16 */
17 const PORT = process.env.PORT ?? 3000;
18
19 app.listen(PORT, callback: () => {
20     // only log this information in development.
21     if (process.env?.NODE_ENV !== "production")
22         console.log(`server listening at http://localhost:${PORT}`);
23 });
24
```

Getting Started With Lightrun

Lightrun allows you to add logs, snapshots and metrics to live applications in real time — no hotfixes, redeployments or restarts necessary.

Start using Lightrun in 3 easy steps:

1. Install the Plugin in one of the supported IDEs.
2. Install the Agent and run it with your application.
3. Add actions to your application code at runtime.

Alternatively, you can use the Lightrun CLI to add actions programmatically.



Install the Plugin

Use the plugin to add real-time Lightrun actions to your application, directly from your IDE



Install the Agent

The Lightrun agent runs alongside your application to enable Lightrun real-time instrumentation



Command Line Tool (Optional)

Use the Lightrun command-line interface to programmatically add real-time logs, snapshots and metrics to your application





```
npm install lightrun
```

```
npm run build
```

```
npm run start
```



```
require('lightrun').start({  
  company: 'defaultcompany',  
  lightrunSecret: 'xxxxxxx-946e-4a24-889d-000000000000',  
});
```

```
1 import express from "express";
2
3 const app = express();
4
5 app.get("/", (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
6     res.send( body: "Hello World!");
7 });
8
9 app.get("/Hello", (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
10     res.send( body: "World!");
11 });
12
13 /*
14 PORT could be set via environment variable (e.g. by Google Cloud
15 or some other cloud service). Fallback to your desired PORT.
16 */
17 const PORT = process.env.PORT ?? 3000;
18
19 app.listen(PORT, callback: () => {
20     // only log this information in development.
21     if (process.env?.NODE_ENV !== "production")
22         console.log(`server listening at http://localhost:${PORT}`);
23 });
24
```


Let's pull out the Todo Cliché

Front End JavaScript with NodeJS

The Backend is Spring Boot

todos

▼ *What needs to be done?*

Bla

Bla Bla

2 Tasks

All

Active

Completed

Clear completed

Double-click to edit a todo

Project: todo-app ~/dev/projects/todo-app

- idea
- node_modules library root
- src
 - .eslintignore
 - .eslintrc.json
 - .gitignore
 - index.js
 - LICENSE
 - package.json
 - package-lock.json
- External Libraries
- Scratches and Consoles

2 results

```
156 const clearCompleted = async (req) => {
157   await global.Store.ClearCompleted()
158   .then(null, (err) => {
159     if (err) {
160       logger.error(err);
161     }
162   });
163   return actionResult(ok: true, message: 'Cleared all complet
164 );
165
166
167
```

clearCompleted()

Connected to Backend (ver 1.1)

Tags: Production

Agents: Shais-MacBook-Air.local +1

Agents Exceptions

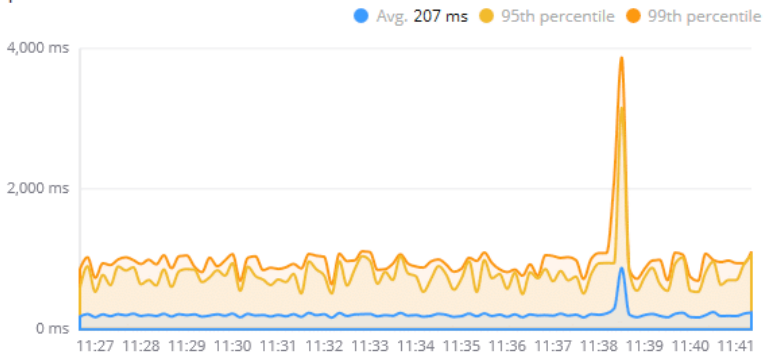
un Snapshots

TicketMonster

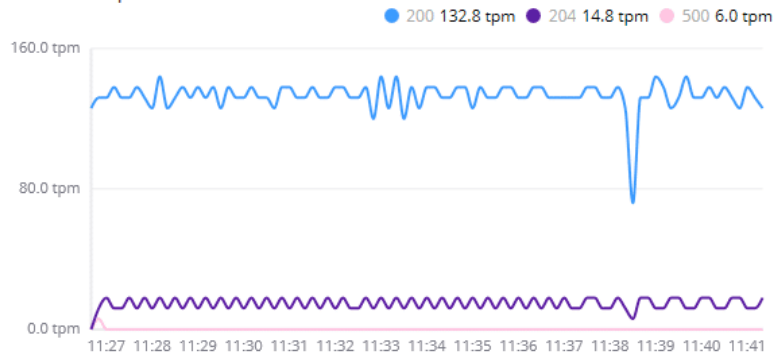
Request

Errors

Response times



Transactions per minute

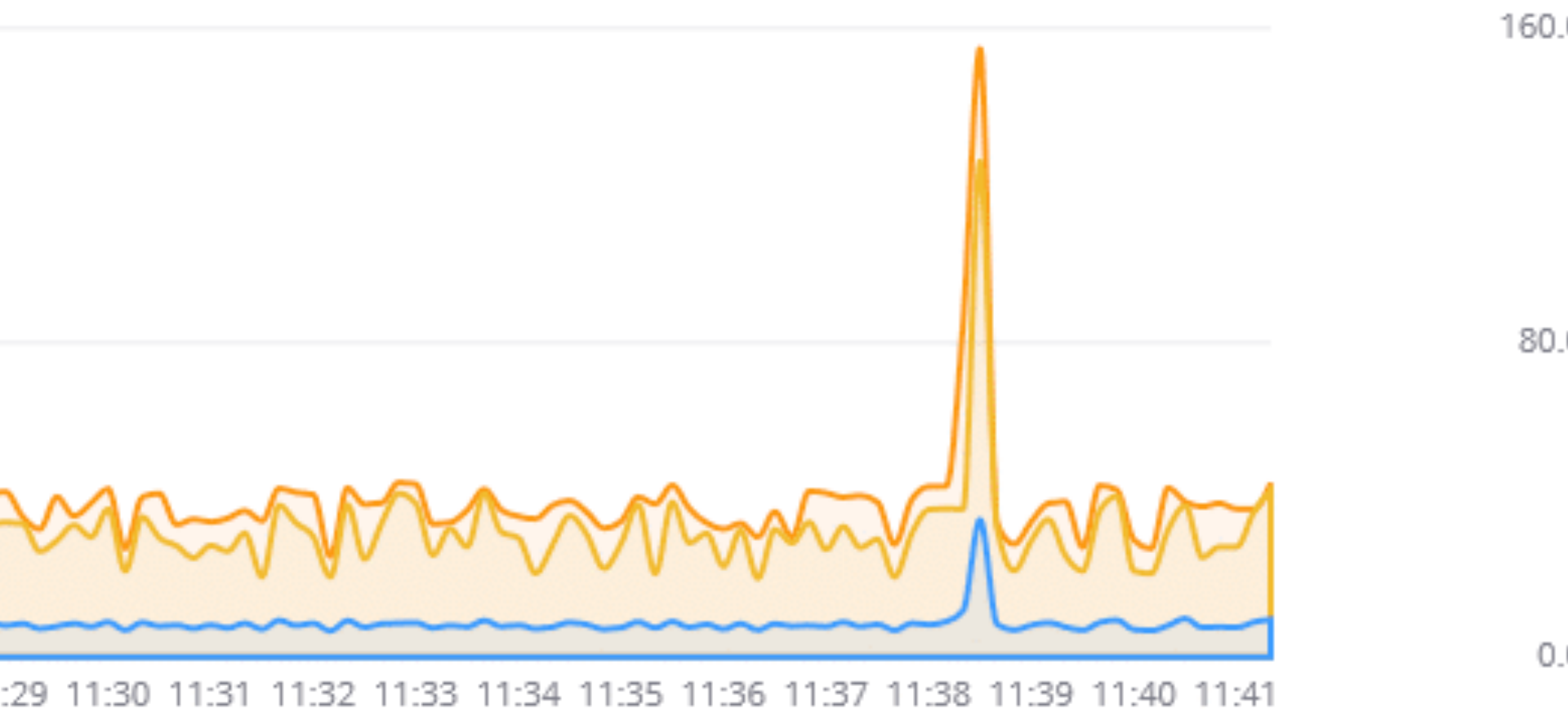


Request

Filter...

Name	Avg. resp. time	95th percentile	TPM	Impact ⓘ ↓
GET /ticket-monster/rest/bookings	879 ms	1,102 ms	14.8 tpm	<div style="width: 100%; height: 10px; background-color: #0070c0;"></div>
GET /ticket-monster/rest/shows	126 ms	127 ms	29.6 tpm	<div style="width: 25%; height: 10px; background-color: #0070c0;"></div>
GET /ticket-monster/rest/venues	155 ms	130 ms	14.8 tpm	<div style="width: 15%; height: 10px; background-color: #0070c0;"></div>
GET /ticket-monster/rest/venues/:venueId	140 ms	130 ms	14.8 tpm	<div style="width: 15%; height: 10px; background-color: #0070c0;"></div>

● Avg. 207 ms ● 95th percentile ● 99th percentile



Summary

Summary



Summary



Summary



Summary



Stability

Inserted Actions are emulated in a dedicated Sandbox to validate there are no side effects of the original flow and state of the process

Summary



Stability

Inserted Actions are emulated in a dedicated Sandbox to validate there are no side effects of the original flow and state of the process



Security

Authorization and authentication, integration with common IDPs



Summary



Stability

Inserted Actions are emulated in a dedicated Sandbox to validate there are no side effects of the original flow and state of the process



Security

Authorization and authentication, integration with common IDPs



Privacy

PII redaction and blacklisting of files / methods / members

Summary



Stability

Inserted Actions are emulated in a dedicated Sandbox to validate there are no side effects of the original flow and state of the process



Security

Authorization and authentication, integration with common IDPs



Privacy

PII redaction and blacklisting of files / methods / members



Footprint

CPU footprint is negligible.
Memory and network footprints are capped and configurable

Summary



Stability

Inserted Actions are emulated in a dedicated Sandbox to validate there are no side effects of the original flow and state of the process



Security

Authorization and authentication, integration with common IDPs



Privacy

PII redaction and blacklisting of files / methods / members



Footprint

CPU footprint is negligible.
Memory and network footprints are capped and configurable



Environment agnostic

Operates on-prem / cloud, microservices, serverless





Questions?

Shai Almog

Developer Advocate @ Lightrun



 twitter.com/debugagent

 talktotheduck.dev

 github.com/shai-almog

 shaia@lightrun.com

 codenameone@apache.org