



# [web-store] Shopist checkouts have a high number of fai...

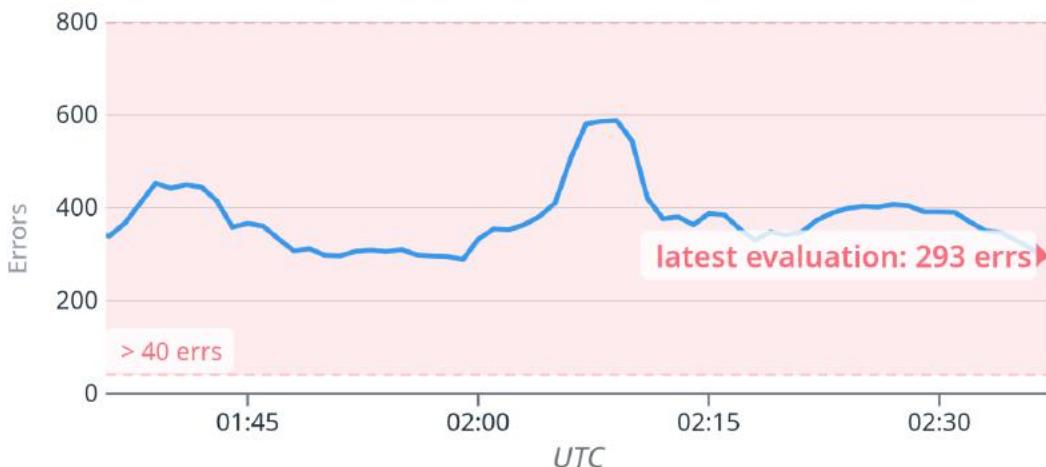
Status	Urgency	Triggered	Responder	Service	Tags
TRIGGERED	HIGH	May 15, 4:38 am	 Scott Gerring	 web-store	5 tags

## NEXT STEPS

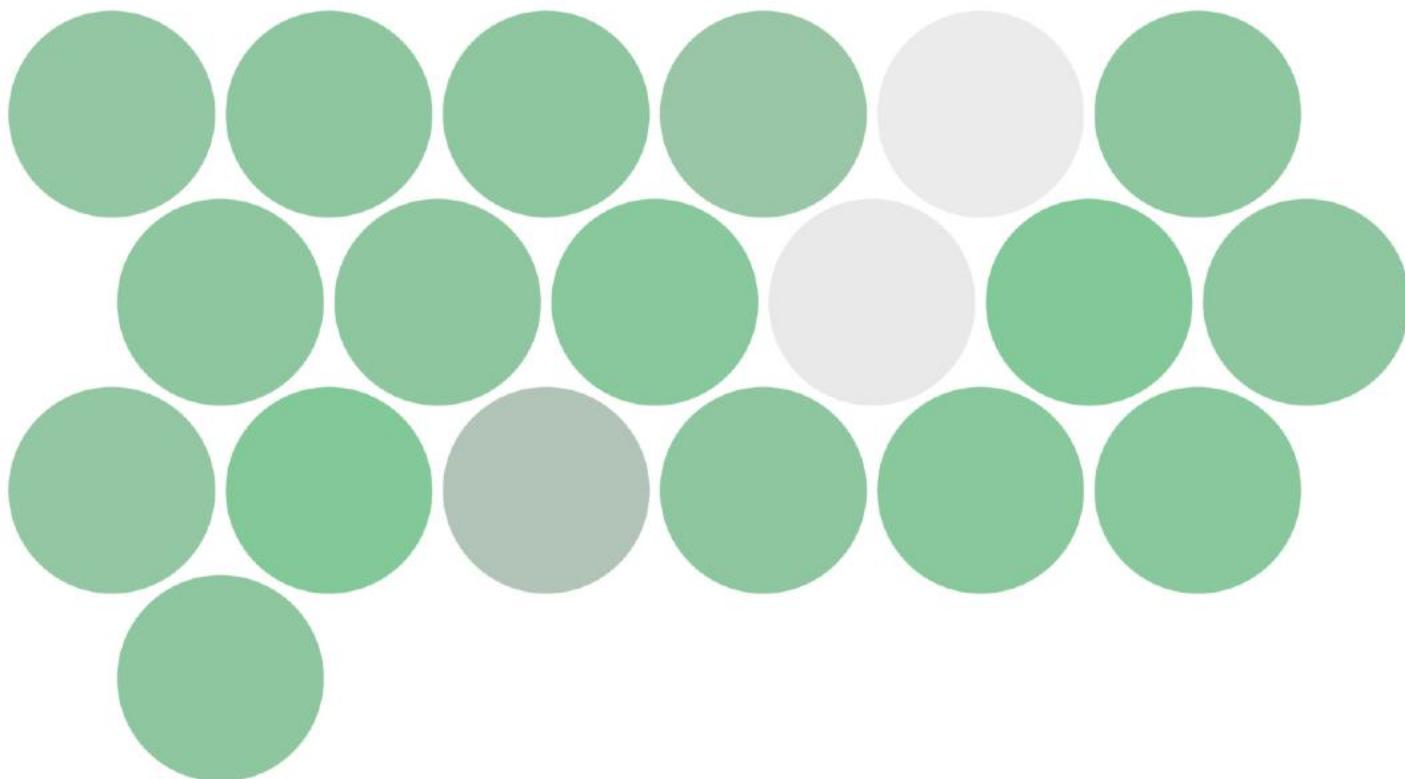
[Acknowledge](#)[✓ Resolve](#)[Declare Incident](#)

## Description

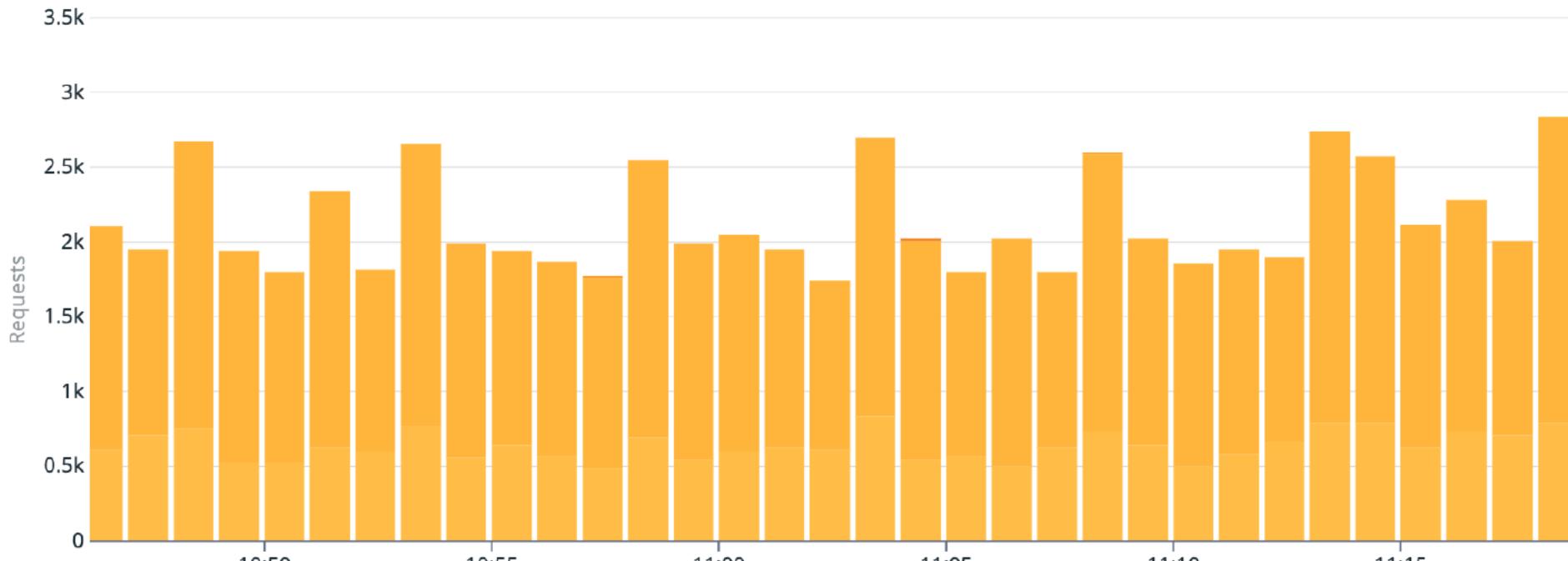
@slack-Datadog\_Incidents-shopist-ops @oncall-gs-oncall [@scott.gerring@datadoghq.com](#)

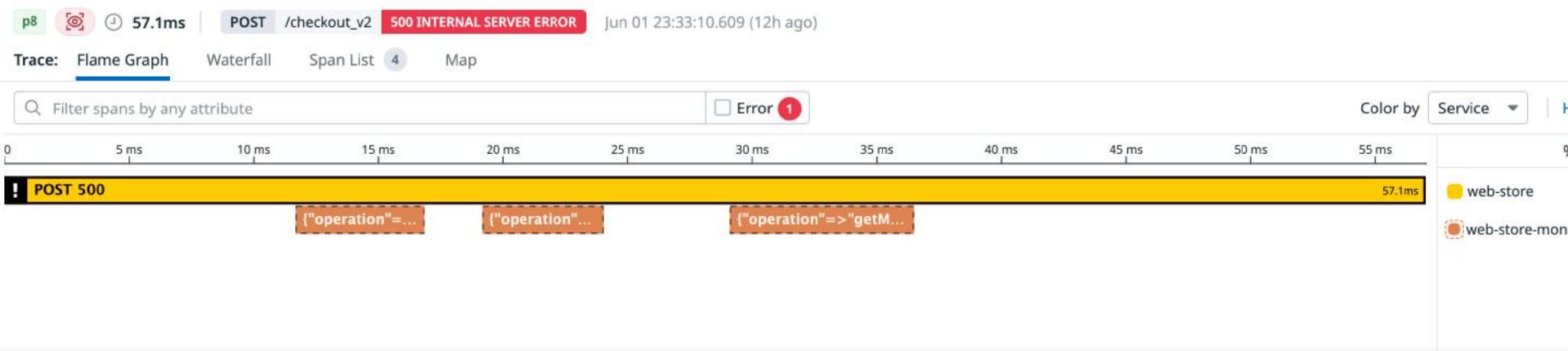


# Your infrastructure 19



## ELBs with Most Requests





Span: Overview Errors 1 Infrastructure Metrics Logs 1 Network Processes Profiles Dev Agent

! web-store > POST 500

This issue is 16 days old (vqaf0-958893) - Last seen 10 hours ago (vcd13006gh-56f3d8) [View in Errors](#)

[Pretty](#) [Raw](#)

ArgumentError: mday out of range

```
> lib/version_helper.rb:51:in `utc': mday out of range (ArgumentError)
>   from lib/version_helper.rb:51:in `get_twice_daily_version'
>   from controllers/application_controller.rb:115:in `add_version_tag'
Show 78 third-party frames
```

37 Open ✓ 1,986 Closed

Author ▾  Label ▾  Projects ▾  Milestones ▾  Reviews ▾  Assignee ▾  Sort ▾

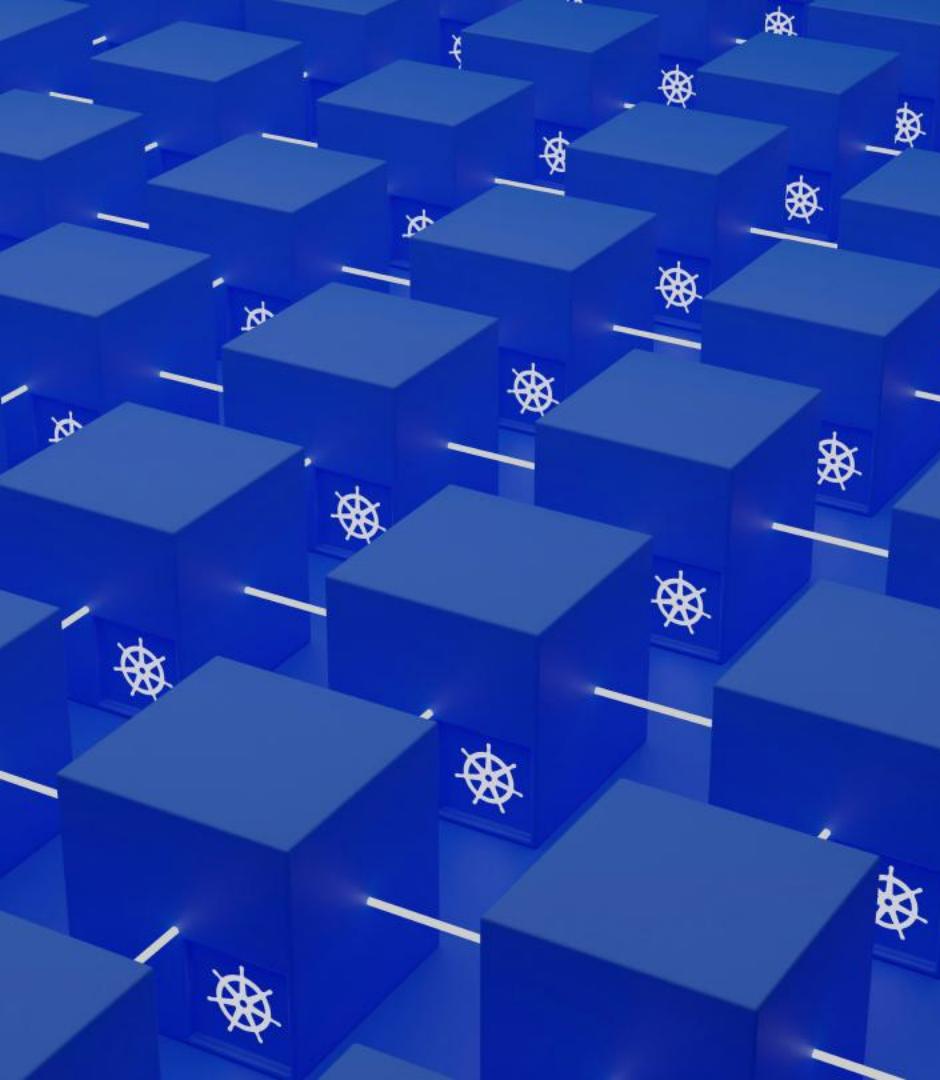
 Fix that pesky runtime issue ✓ 104

#1965 by Fred Developer was merged on Aug 23, 2024 • Approved  2 of 4 tasks

Back to Basics:

**Crafting Quality Software  
in the Age Of Complexity**

Scott Gerring, Datadog



**Microservices!**  
**The Cloud!**  
**GenAI!**

# GenAI

```
/// A simple function to generate a random axis (0, 1, or 2)
/// We'd normally use a proper random number generator,
/// but for simplicity we'll just use a deterministic pattern
fn rand_axis() -> usize {
    static mut COUNTER: usize = 0;

    unsafe {
        COUNTER = (COUNTER + 1) % 3;
        COUNTER
    }
}
```

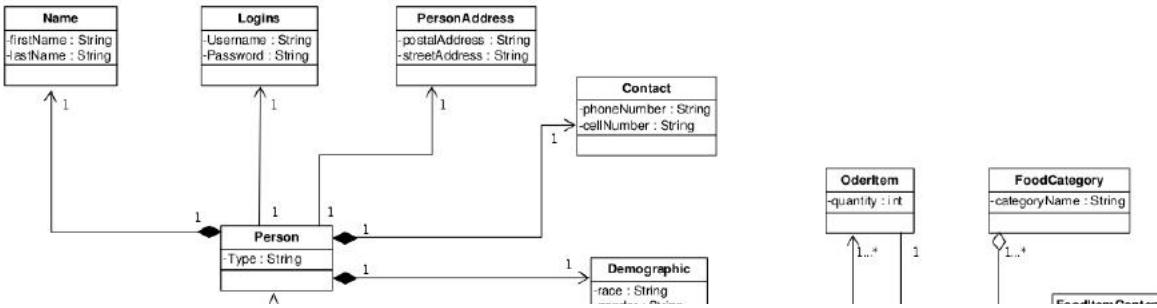
**Quality ... you know what it is, yet you don't know what it is.**

- Robert Pirsig

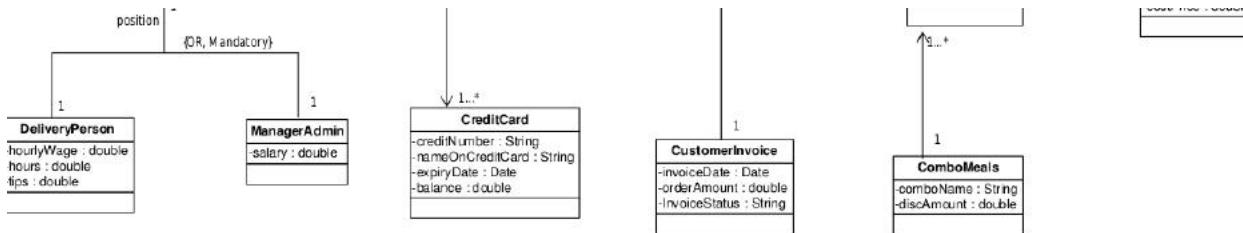


**“The earlier you catch  
defects, the cheaper they are  
to fix”**

David Farley, on not shooting yourself in  
the foot in production



# Thoughtful Design





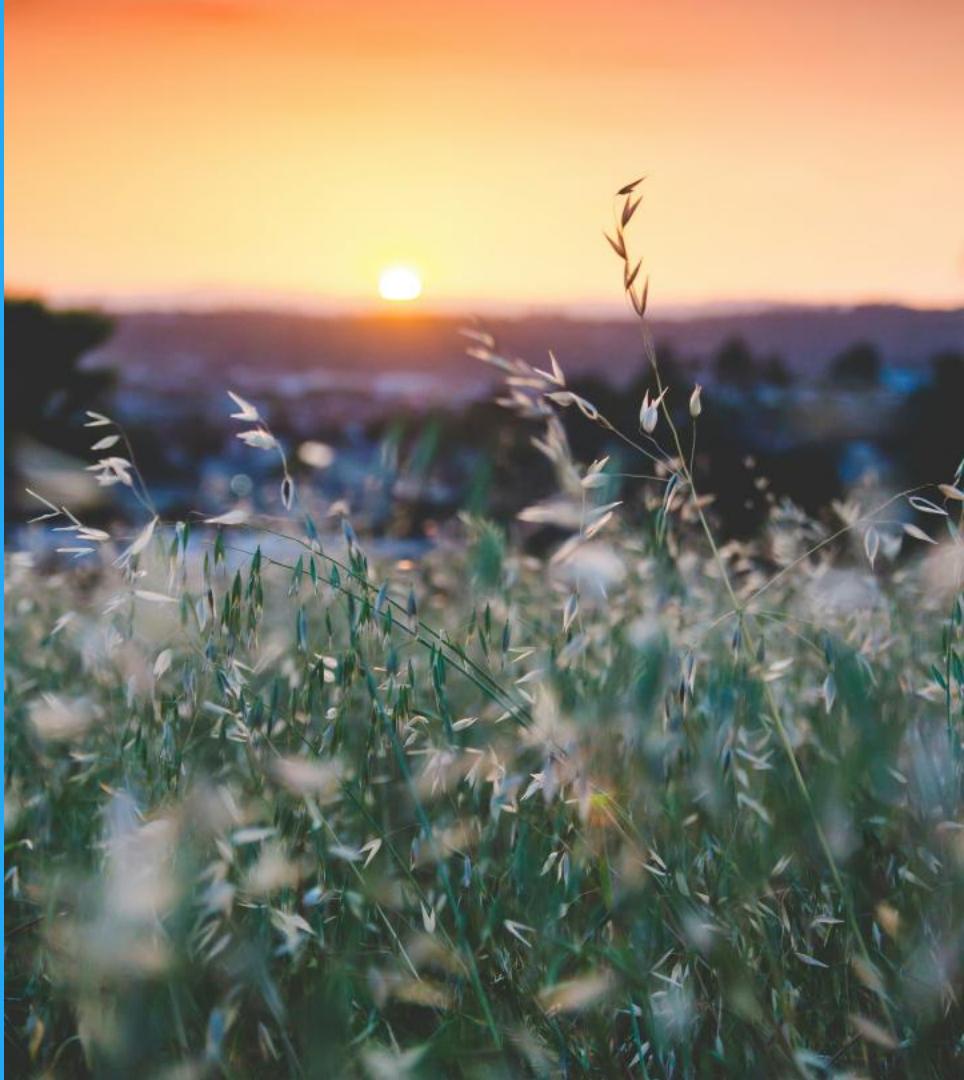
# Static Analysis

10. **Rule:** All code must be compiled, from the first day of development, with *all* compiler warnings enabled at the compiler's most pedantic setting. All code must compile with these setting without any warnings. All code must be checked daily with at least one, but preferably more than one, state-of-the-art static source code analyzer and should pass the analyses with zero warnings.

**Rationale:** There are several very effective static source code analyzers on the market today, and quite a few freeware tools as well.<sup>2</sup> There simply is no excuse for any software development effort not to make use of this readily available technology. It should be considered routine practice, even for non-critical code development.

# Linting & Style Checks

Easy, Fast, Cheap



“Gofmt’s style is no one’s favorite, yet gofmt is everyone’s favorite.”

# Common Tools

- **Java** - Spotbugs, Checkstyle, Spotless, Error Prone, PMD, ...
- **.net** - Roslyn Analyzers, StyleCop
- **Python** - pylint / flake8
- **Rust** - clippy / rustfmt
- **Agnostic**: Megalinter, Sonar

# Linting Recommendations

## Do:

- Use a **modern** linter!
- Embrace **Opinionated Formatting Tools** like **Spotless**
- Everything checked in CI and trivial to run locally

Demo:  
**Project Setup & Core Linting**

“Jane in security keeps telling me to ‘fix the log4shells’. We’ve not touched the codebase in years, and I have no idea what she’s talking about anyway”.

# Software Composition Analysis

We put an exploit in your supply chain so you can mine bitcoin while you deliver business value



Filters

is:open is:pr author:app/dependabot

Labels 11

Milestones 0

New pull request

X Clear current search query, filters, and sorts

 20 Open ✓ 18 Closed

Author ▾

Label ▾

Projects ▾

Milestones ▾

Reviews ▾

Assignee ▾

Sort ▾

  Bump typescript from 5.6.3 to 5.7.3 ✘ dependencies no-pr-activity

1

#38 opened on Feb 1 by dependabot bot  Bump @iconify-json mdi from 1.2.1 to 1.2.3 ✘ dependencies no-pr-activity

1

#37 opened on Feb 1 by dependabot bot  Bump astro-expressive-code from 0.37.1 to 0.40.1 ✘ dependencies no-pr-activity

1

#36 opened on Feb 1 by dependabot bot  Bump @astrojs/tailwind from 5.1.1 to 6.0.0 ✘ dependencies no-pr-activity

1

#35 opened on Feb 1 by dependabot bot  Bump prettier-plugin-tailwindcss from 0.6.8 to 0.6.11 ✘ dependencies no-pr-activity

1

#34 opened on Feb 1 by dependabot bot  Bump @astrojs/rss from 4.0.7 to 4.0.11 ✘ dependencies no-pr-activity

1

#33 opened on Feb 1 by dependabot bot  Bump @tailwindcss/typography from 0.5.15 to 0.5.16 ✘ dependencies no-pr-activity

1

#32 opened on Feb 1 by dependabot bot  Bump satori from 0.11.2 to 0.12.1 ✘ dependencies no-pr-activity

1

#31 opened on Feb 1 by dependabot bot  Bump react-util-directive from 2.0.0 to 2.1.0 ✘ dependencies no-pr-activity

1

# Dependabot alerts

[Give feedback](#)

Dependency files checked 2 minutes ago

is:open

14 Open  0 Closed

Package ▾

Ecosystem ▾

Manifest ▾

Severity ▾

Sort ▾

Misuse of ServerConfig.PublicKeyCallback may cause authorization bypass in golang.org/x/crypto Critical

#1 opened 2 minutes ago • Detected in golang.org/x/crypto (Go) • apps/pass-api/go.mod

Borsh serialization of HashMap is non-canonical High

#10 opened 2 minutes ago • Detected in hashbrown (Rust) • apps/pass-image-api/Cargo.lock

golang.org/x/crypto Vulnerable to Denial of Service (DoS) via Slow or Incomplete Key Exchange High

#3 opened 2 minutes ago • Detected in golang.org/x/crypto (Go) • apps/pass-api/go.mod

ring has some AES functions that may panic when overflow checking is enabled in Moderate

#14 opened 2 minutes ago • Detected in ring (Rust) • apps/pass-image-api/Cargo.lock

Some AES functions may panic when overflow checking is enabled in ring Moderate

#12 opened 2 minutes ago • Detected in ring (Rust) • apps/pass-image-api/Cargo.lock

`idna` accepts Punycode labels that do not produce any non-ASCII when decoded Moderate



 Checkmarx



**So many  
tools!**



# Software Composition Analysis Recommendations

- You're going to want to do this ...
- ... and GitHub's default tooling is a great low effort way to start

“I want to make sure that Joe the Intern is automatically prevented from passing unescaped user input to raw SQL queries ”

# Getting Weird(er)

Semgrep, CodeQL,  
and Static  
Application Security  
Testing



# CodeQL – Query your codebase

# Choose a workflow

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and [set up a workflow yourself →](#)

## Categories

Deployment

Security



Continuous integration

Automation

Pages

code scanning X

Found 78 workflows

### CodeQL Analysis

By GitHub



Security analysis from GitHub for C, C++, C#, Go, Java, JavaScript, TypeScript, Python, Ruby, Kotlin and Swift developers.

[Configure](#)

Code scanning ●

### Fortify Scan

By OpenText



Integrate Fortify's comprehensive static code analysis (SAST) for 33+ languages into your DevSecOps workflows.

[Configure](#)

Code scanning ●

### Snyk Security

By Snyk



Detect vulnerabilities across your applications and infrastructure with the Snyk platform.

[Configure](#)

Code scanning ●

### Codacy Security Scan

By Codacy



Free, out-of-the-box, security analysis provided by multiple open source static analysis tools.

[Configure](#)

Code scanning ●

### APIsec Scan

By APIsec



APIsec provides the industry's only automated and continuous API testing platform that uncovers security vulnerabilities and logic flaws in APIs.

[Configure](#)

Code scanning ●

### DevSkim

By Microsoft CST-E



DevSkim is a security linter that highlights common security issues in source code.

[Configure](#)

Code scanning ●

### EthicalCheck



### Mayhem for API



### NeuraLegion



# CodeQL & Java – things you can find

- Query built from user-controlled sources
- Externally controlled string lands in a command line exec
- Information disclosure through error messages
- Hardcoded passwords
- ... and much more!

# CodeQL

```
module LiteralToURLConfig implements DataFlow::ConfigSig {
    predicate isSource(DataFlow::
Node source) {
    source.asExpr() instanceof StringLiteral
}

    predicate isSink(DataFlow::
Node sink) {
    /* check if this is used to construct a URL */
}
}

module LiteralToURLFlow = DataFlow::Global
<LiteralToURLConfig>;

from DataFlow::
Node src, DataFlow::Node sink
where LiteralToURLFlow::flow(src, sink)
select src,
"This string constructs a URL $@.", sink, "here"
```

# Semgrep

```
> semgrep --lang java --pattern 'public static void main(...)'
```

## 1 Code Finding

```
.mvn/wrapper/MavenWrapperDownloader.java  
37| public static void main(String[] args) {
```

# Semgrep

```
> semgrep --lang java --pattern '@Path($A) public class $C { ... } '
```

2 Code Findings

```
src/main/java/com/datadoghq/stickerlandia/common/health/HealthResource.java
 13| @Path( "/health" )
 14| public class HealthResource {
...

```

```
src/main/java/com/datadoghq/stickerlandia/stickercatalogue/StickerResource.java
 32| @Path( "/api/stickers/v1" )
 33| public class StickerResource {
...

```

# Semgrep

```
rules:
  - id: entity-import-in-resource
    pattern: import $PACKAGE.entity.$CLASS;
    languages: [java]
    severity: WARNING
    message: Entity import detected in Resource class
    paths:
      include:
        - ".*Resource\\.java$"
```

# SAST, CodeQL, and friends

## Recommendations

- **SAST**
  - You're going to want to do this, too
  - Lean into what you already have available
  - Lots of overlap with SAST tools (Sonar, Snyk, etc.)
- **Code Querying**
  - Is a fun past-time and lets you start to build a metal model of your code

“I want to formalise my model of the architecture of my service to keep it on the rails over time”

# Architecture Testing

Automatically  
enforcing your  
mental model



The logo for ArchUnit, featuring a stylized icon composed of three overlapping blue semi-circles of increasing size from left to right, followed by the word "ArchUnit" in a bold, blue, sans-serif font.

ArchUnit

# Arch Rules like ...

- **Layering** - Things named `.*Resource` shouldn't import things from `javax.persistence`
- **Containment** - Classes named `.*DTO` should reside in packages named `".dto$"`
- **Consistency** - Classes implementing `IThingDoer` should be named `.*ThingDoer`
- **Cycle Checks** - Slices in my application should be free of cycles

```
@Test
void sticker_domain_should_not_import_award_domain() {
    ArchRule rule = noClasses()
        .that()
            .resideInAPackage(..sticker..)
        .should()
            .dependOnClassesThat()
                .resideInAPackage(..award..)
            .because("domains should be isolated");

    rule.check(classes);
}
```

# Architecture Testing: Recommendations

- Check it out!

**Demo:**  
**Architecture & Tech Enforcement**

# Dynamic Analysis

# Observability



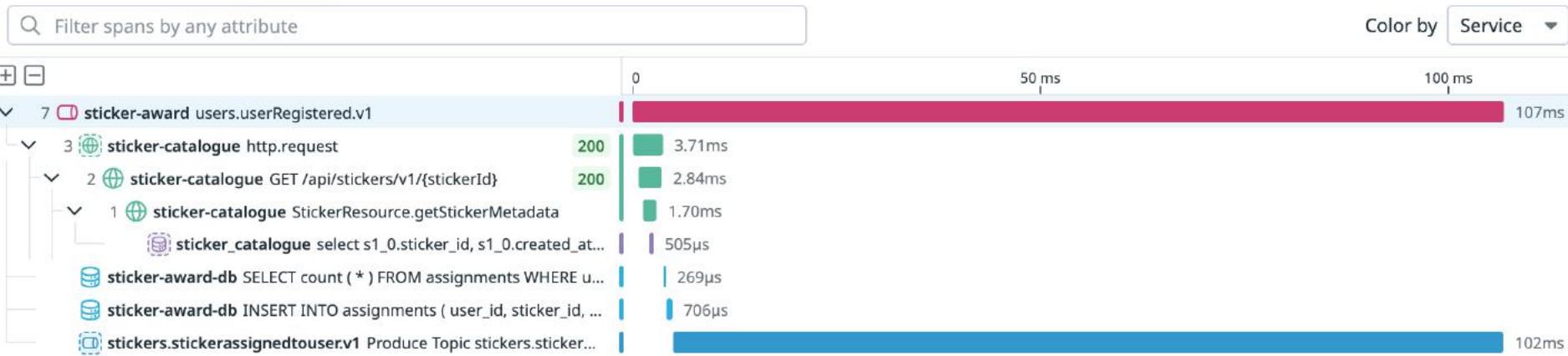
# Observability Is:

- Logs
- Traces
- Metrics



# Tracing

⌚ 107ms | Aug 26 14:23:48.277 (20h ago)

Trace: Flame Graph Waterfall Span List 8 Map

⌚ 107ms 100% total exec time



Span: Overview Infrastructure Metrics Logs 6 Network Processes SQL Queries 3 Profiles

SORT BY Duration

 Show only queries with errors
>  sticker-award-db | ⌚ 706µs | Aug 26, 2025 at 2:23:48.281 pm

View Query in APM →

INSERT INTO assignments (user\_id, sticker\_id, assigned\_at, removed\_at, reason, created\_at, updated\_at) VALUES ( ? ) RETURNING id





p1 1.61ms

GET http://pass-summary-api:80... 500 INTERNAL SERVER ERROR

Aug 25 07:56:38.552 (2d ago)

Trace: Flame Graph

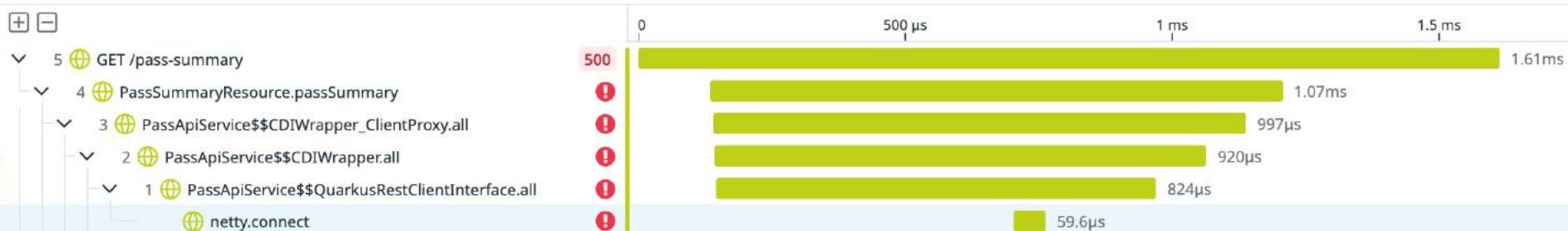
Waterfall

Span List 6 Map

Filter spans by any attribute

 Errors 6

Color by Service



pass-summary-api netty.connect netty.connect

59.6μs 3.69% total exec time



Span: Overview

Errors 6

Infrastructure

Metrics

Logs 0

Network

Processes

Profiles

Error Message

io.netty.channel.AbstractChannel\$AnnotatedConnectException: Connection refused: pass-api/172.20.83.30:8080

View stack trace

&gt; Pinned Span Attributes No pinned tags found

# Tracing - Getting Started

- Auto- vs. manual- instrumentation
- Explicit vs implicit collection

# Auto Instrumentation

```
import io.opentelemetry.api.GlobalOpenTelemetry;
import io.opentelemetry.api.trace.Span;
import io.opentelemetry.api.trace.Tracer;
import io.opentelemetry.extension.annotations.SpanAttribute;
import io.opentelemetry.extension.annotations.WithSpan;

public class AwardService {

    private static final Tracer tracer;

    @WithSpan
    public void grantAward(@SpanAttribute("user.id") String userId) {
        Span.current()
            .setAttribute("award.type", "certification")
            .addEvent("award_grant_started");

        // ...
    }
}
```

# Manual Instrumentation



# Observability Is:

- Logs
- Traces
- Metrics
- Continuous Profiling?!

# Manual Profiling

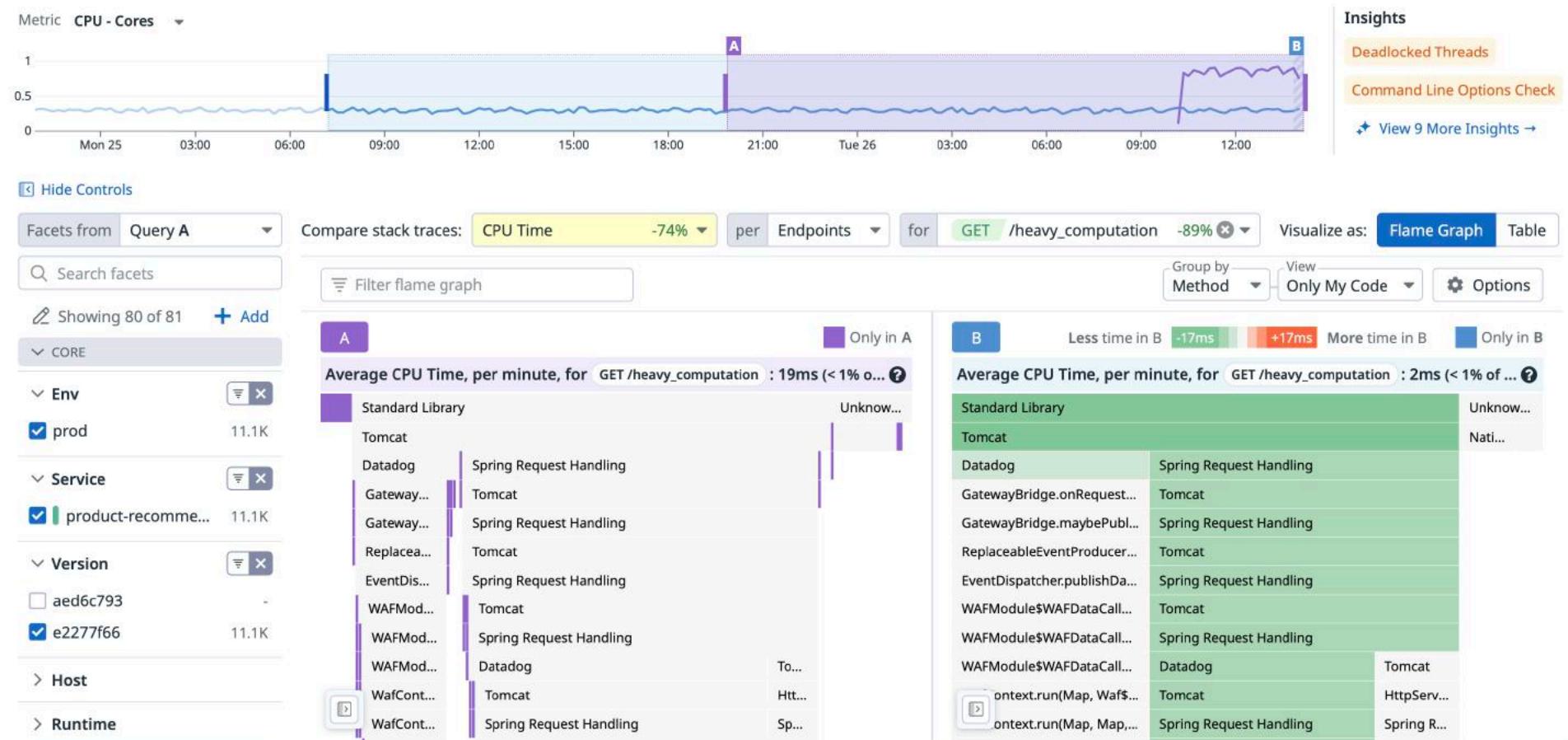
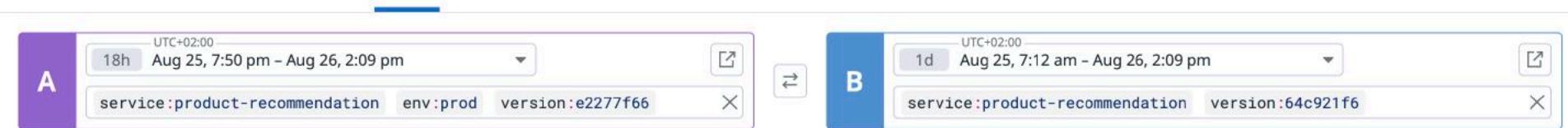
- Manual runs & analysis under test
- Custom builds (gprof)
- Binary rewriting (valgrind)
- Dynamic Instrumentation / eBPF (dtrace)
- Some statistical sampling (prof, JFR)





# Continuous Profiling

- Statistical sampling, load overhead
- Always on in prod
- Correlated with traces (and request metadata!)



# Continuous Profiling - Pragmatic Advice

- Keep an eye on it! It'll be table stakes soon enough.

# Crafting Quality Software

- Quality has to be tended to over time
- Good design is good ...
  - ... automated guard-rails are essential!
- Static|Dynamic analysis is here to help
- Encode structural checks for what matters

# That's it!



<https://scottgerring.com>

# Danke für Euer Feedback!



<https://cloud.ijug.eu/apps/forms/s/SAPcFoApG4rifPKewxoFPtkc>