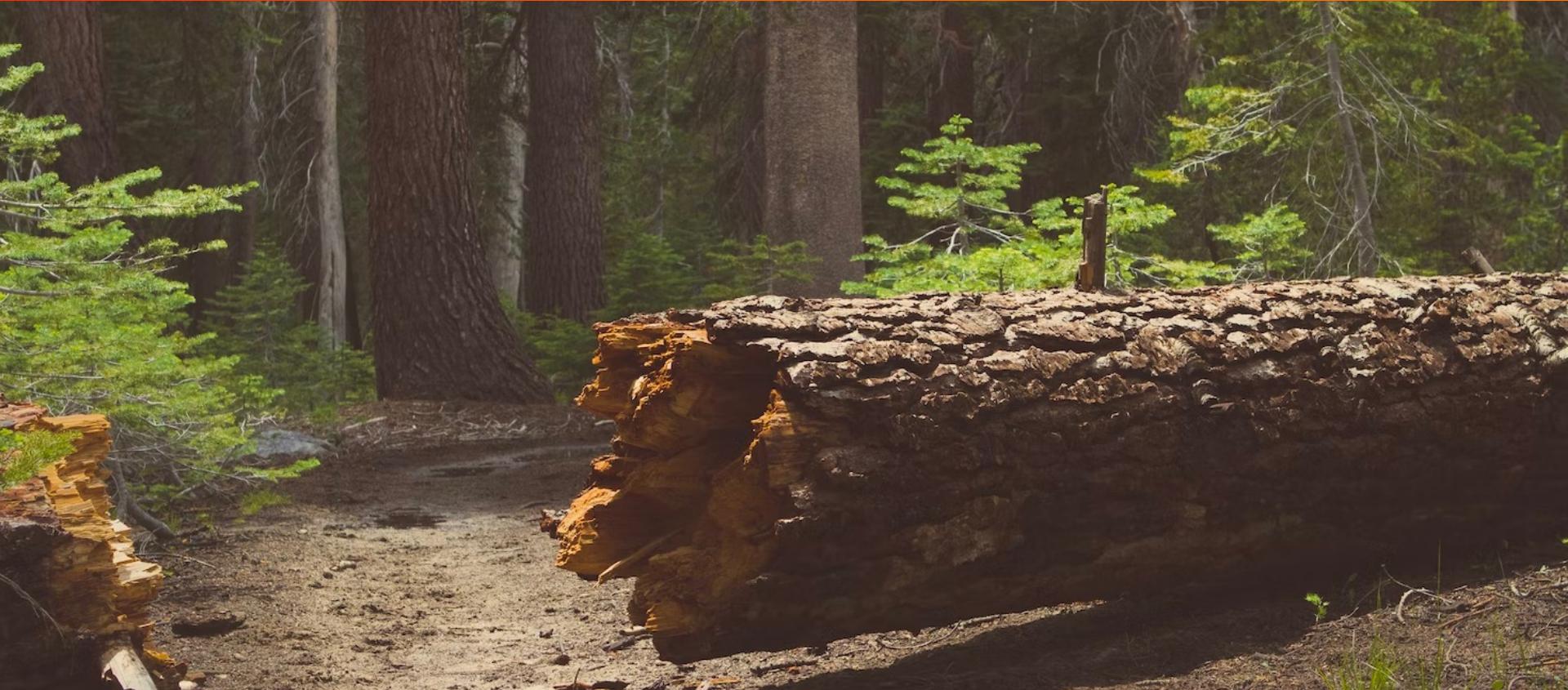




Debugging Schrödinger's App



How do fallen trees check for errors?



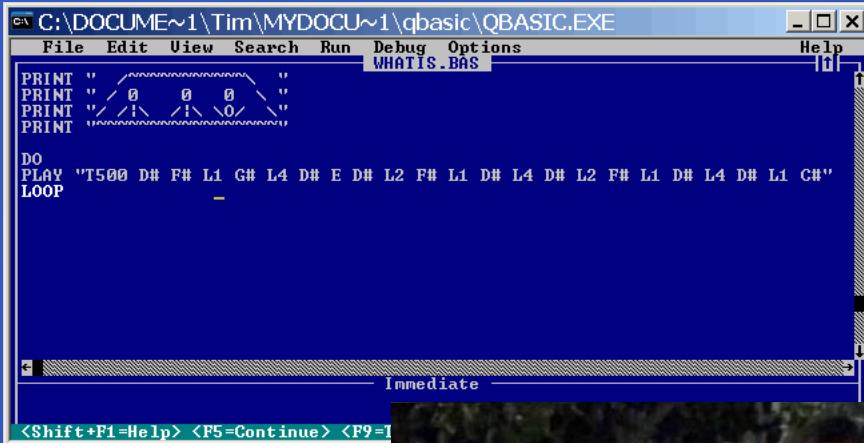


DeveloperSteve Coochin Lumigo Senior Developer Advocate

Dev/Maker/Geek/Speaker/streamer/ruralite



Building and Deploying Apps for Many Years



C:\DOCUMENTS\Tim\MYDOCUMENTS\QBASIC.EXE

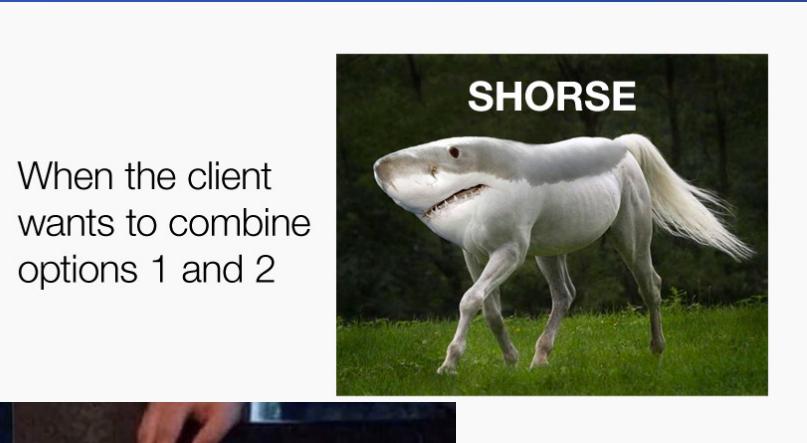
File Edit View Search Run Debug Options Help

WHATIS.BAS

```
PRINT "~~~~~"
PRINT "/ \   \   \ "
PRINT "/ \ \ / \ \ \ "
PRINT "DO
PLAY "T500 D# F# L1 G# L4 D# E D# L2 F# L1 D# L4 D# L2 F# L1 D# L4 D# L1 C#"
LOOP
```

Immediate

<Shift+F1=Help> <F5=Continue> <F9=



For Scale, Build you Must

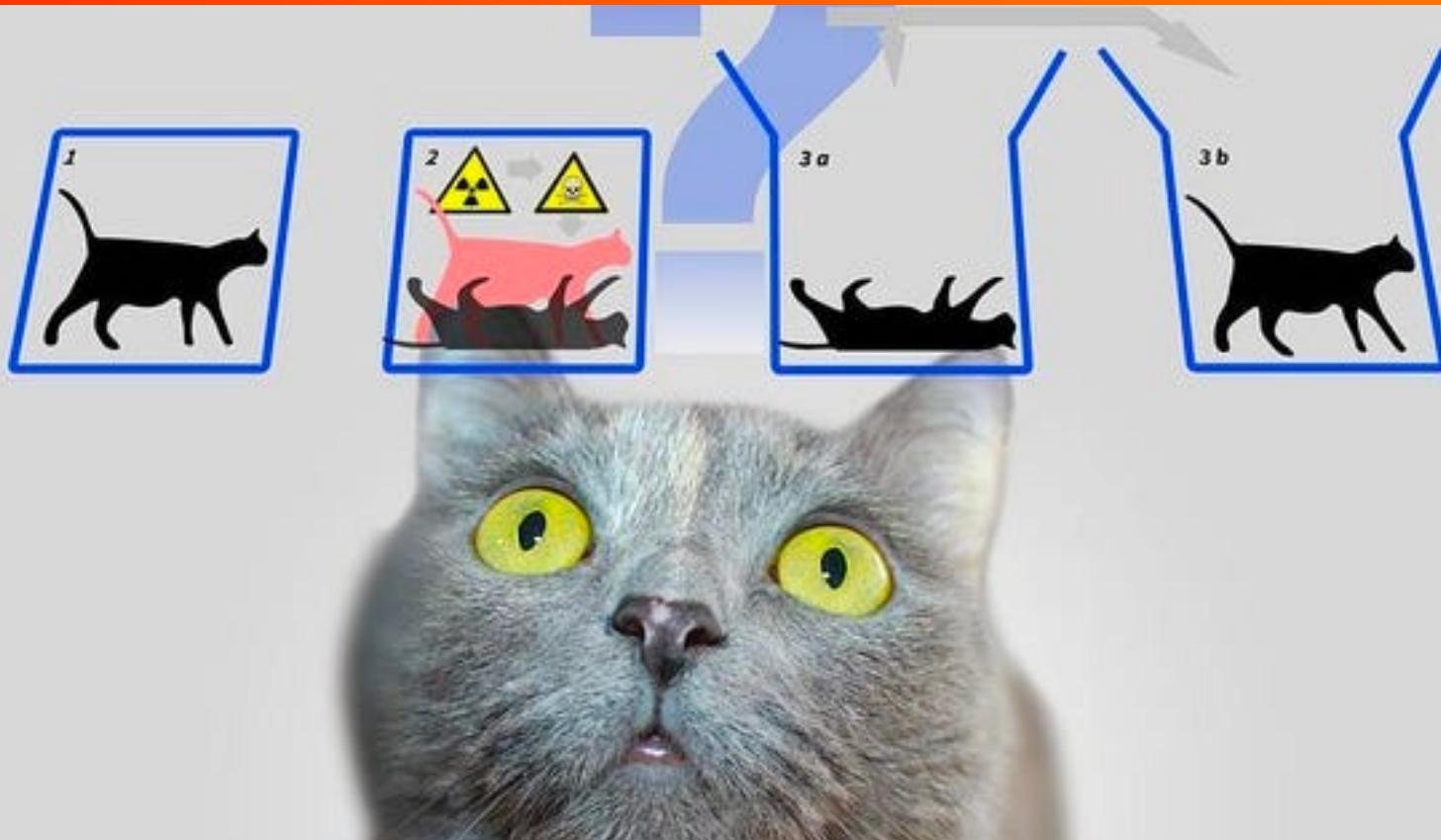


Python Versatility

- Installable game/app dev
- API's
- Data Science
- Web (Django, Flask)
- IOT

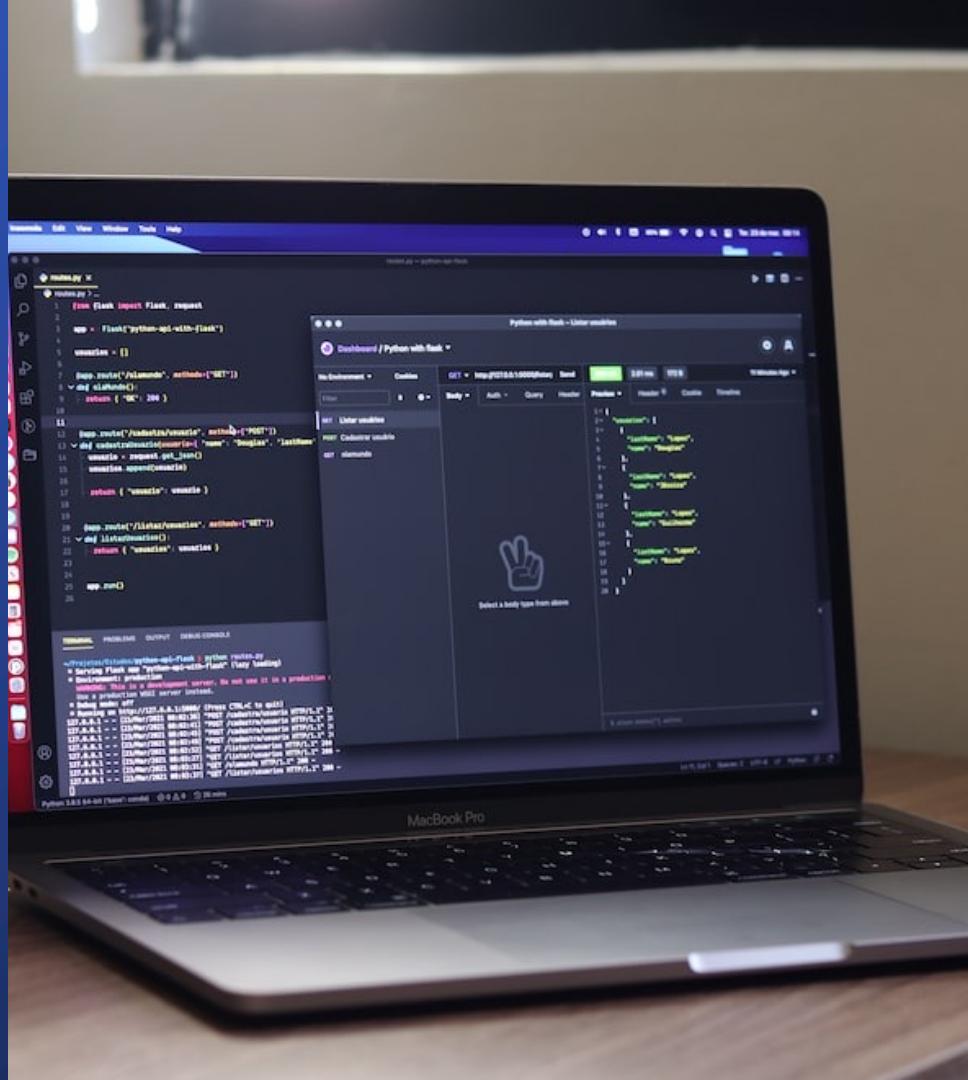


Schrodinger's Rules of Observability



Python local Debugging

- Core debugging
 - *print()*
 - *logging.warning()*
 - *Pdb*
- Library debugging IE
 - *pprint()*
- IDE (and plugins)
 - VSCode
 - Eclipse



Schrödinger's App

Schrödinger's App

"If a tree falls in the forest and no one hears it, then it is both standing and not standing."

Task Title

Enter Todo...

Add

Task #1 | What do you call 8 bits in Python?

Not Complete

Finished

Delete

A SNAKE BYTE

Schrödinger's App Routes



[get] todo

[post] todo

[update] todo

[delete] todo



Demo time

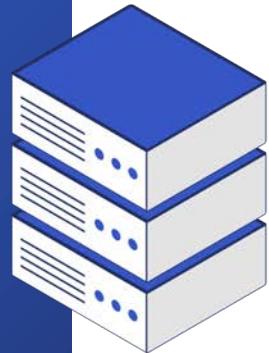


Cloud Deployed Debugging

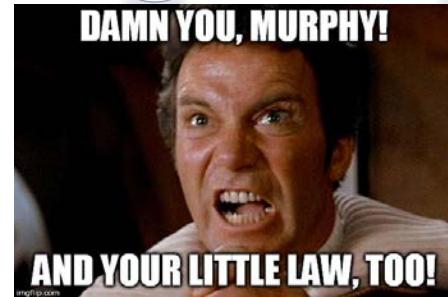


Cloud Native Applications

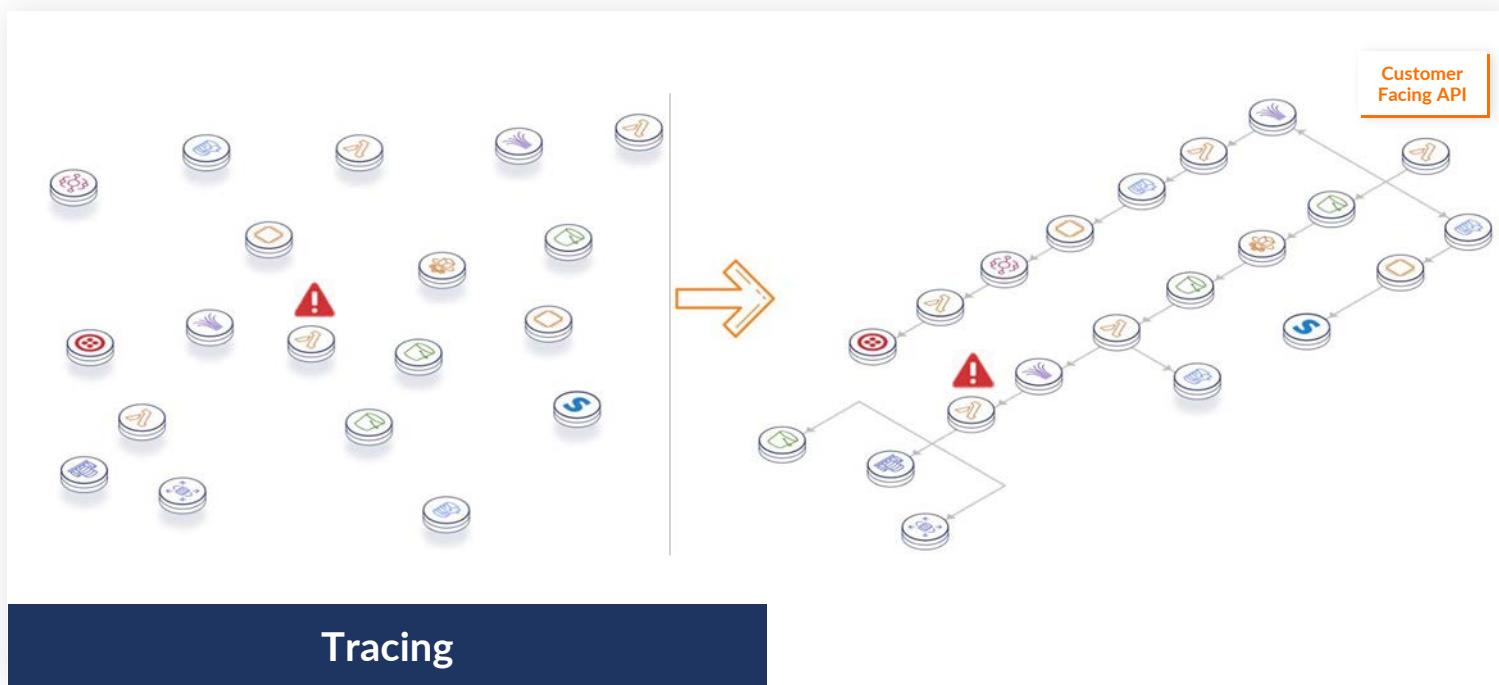
are highly distributed with hundreds of ever-changing components



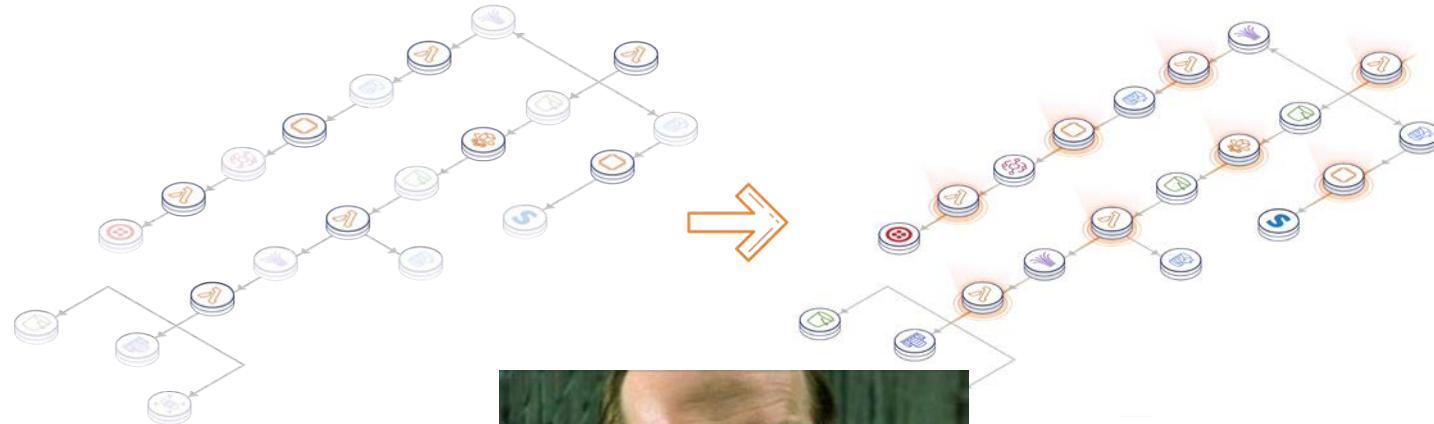
Hard to zoom-out and understand overall application health without understanding the context and connections between them



Distributed Tracing



Agentless Observability

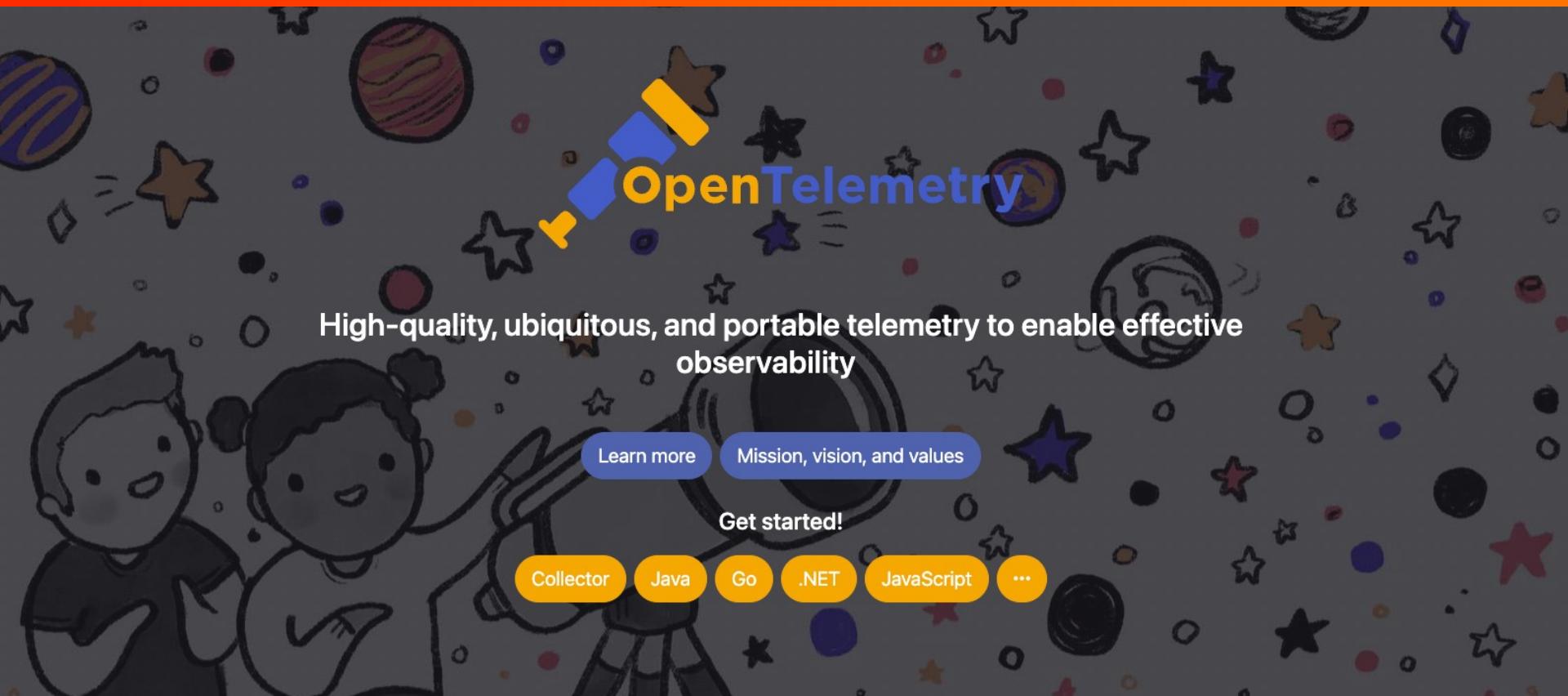


Tracing

Agentless Managed Services



Open Telemetry (OTEL)



Microservices

App Code

OTel Auto. Inst.

OTel API

OTel SDK

OTLP

3rd party
service

OTel
Collector

Kubernetes

OTLP

L7 Proxy

OTLP



OTLP

Shared
Infra



Managed DBs

APIs

Time Series
Databases

Trace
Databases

Column
Stores

Observability
Frontends
& APIs

Client Instrumentation

opentelemetry.io/docs/instrumentation

Docs

▶ Concepts

Instrumentation

▶ C++

▶ .NET

▶ Erlang/Elixir

▶ Go

▶ Java

▶ JavaScript

▶ PHP

▶ Python

▶ Ruby

Rust

Swift

▶ Collector

▶ Migration

▶ Reference

Acknowledgements

C++



A language-specific implementation of OpenTelemetry in C++.

.NET



A language-specific implementation of OpenTelemetry in .NET.

Erlang/Elixir



A language-specific implementation of OpenTelemetry in Erlang/Elixir.

Go



A language-specific implementation of OpenTelemetry in Go.

Java



A language-specific implementation of OpenTelemetry in Java.

JavaScript



A language-specific implementation of OpenTelemetry in JavaScript (for Node.js & the browser).

PHP



A language-specific implementation of OpenTelemetry in PHP.

Python



A language-specific implementation of OpenTelemetry in Python.

Ruby



A language-specific implementation of OpenTelemetry in Ruby.

[View page source](#)

[Edit this page](#)

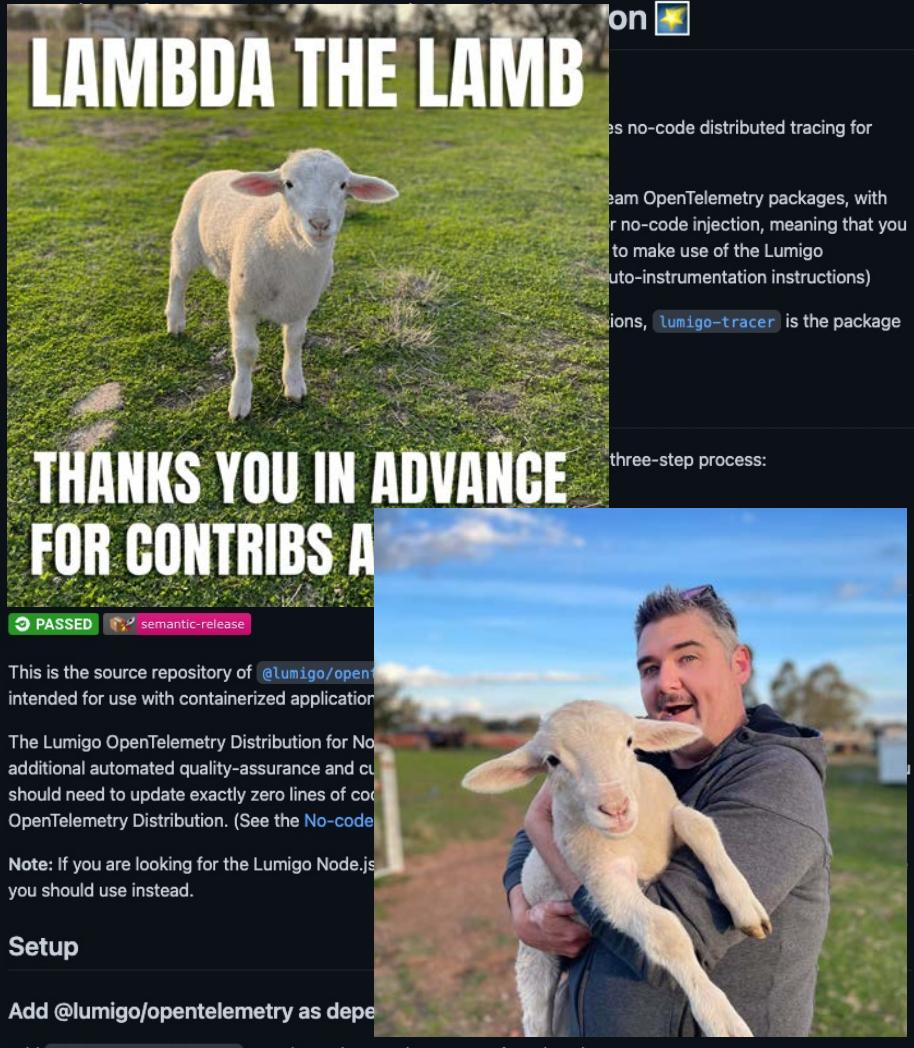
[Create documentation issue](#)



memegenerator.net

Lumigo Open Telemetry Tracer

- Python
github.com/lumigo-io/opentelemetry-python-distro
- NodeJS
github.com/lumigo-io/opentelemetry-js-distro



LAMBDA THE LAMB

This is the source repository of [@lumigo/opentelemetry](#) intended for use with containerized applications.

The Lumigo OpenTelemetry Distribution for Node.js includes additional automated quality-assurance and continuous integration. You should need to update exactly zero lines of code to use the Lumigo OpenTelemetry Distribution. (See the [No-code](#) section.)

Note: If you are looking for the Lumigo Node.js tracer, you should use instead.

Setup

Add [@lumigo/opentelemetry](#) as dependency

Add [@lumigo/opentelemetry](#) as a dependency using your preferred package manager:

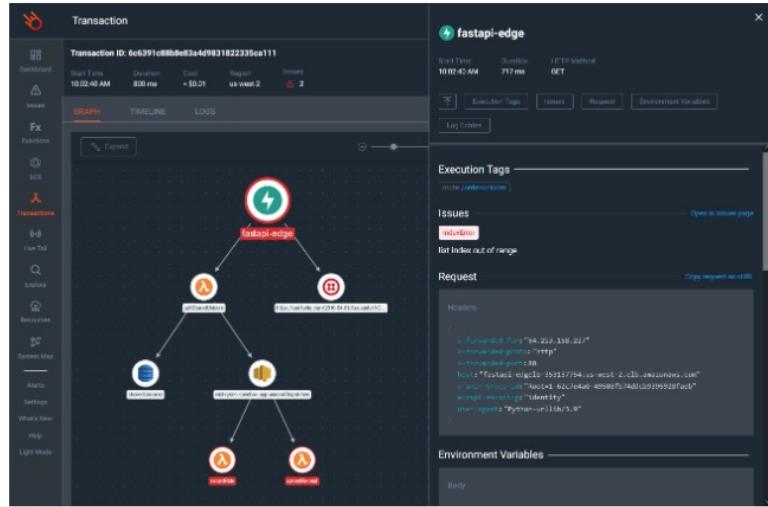
Install and configure the Lumigo tracers

- Install tracers
- Reference in the code
- Configure env vars
 - OTEL_SERVICE_NAME
 - LUMIGO_TRACER_TOKEN

Containerized applications

Documentation to get you started monitoring containerized applications with Lumigo and OpenTelemetry.

Containers are integrated in transactions alongside Lambda functions, DynamoDB, SQS, third-party services and everything else Lumigo supports.



A transaction shown in Lumigo that spans an ECS task running a FastAPI application, several Lambda functions, DynamoDB, SNS and Twilio



You can try it today, for free!

This screenshot is taken from [Lumigo's Sandbox](#), feel free to go try it for yourself!

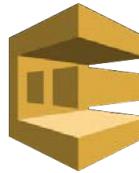
Deploying Schrödinger's App



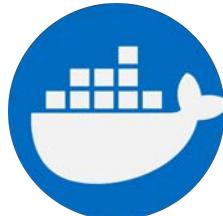
Flask



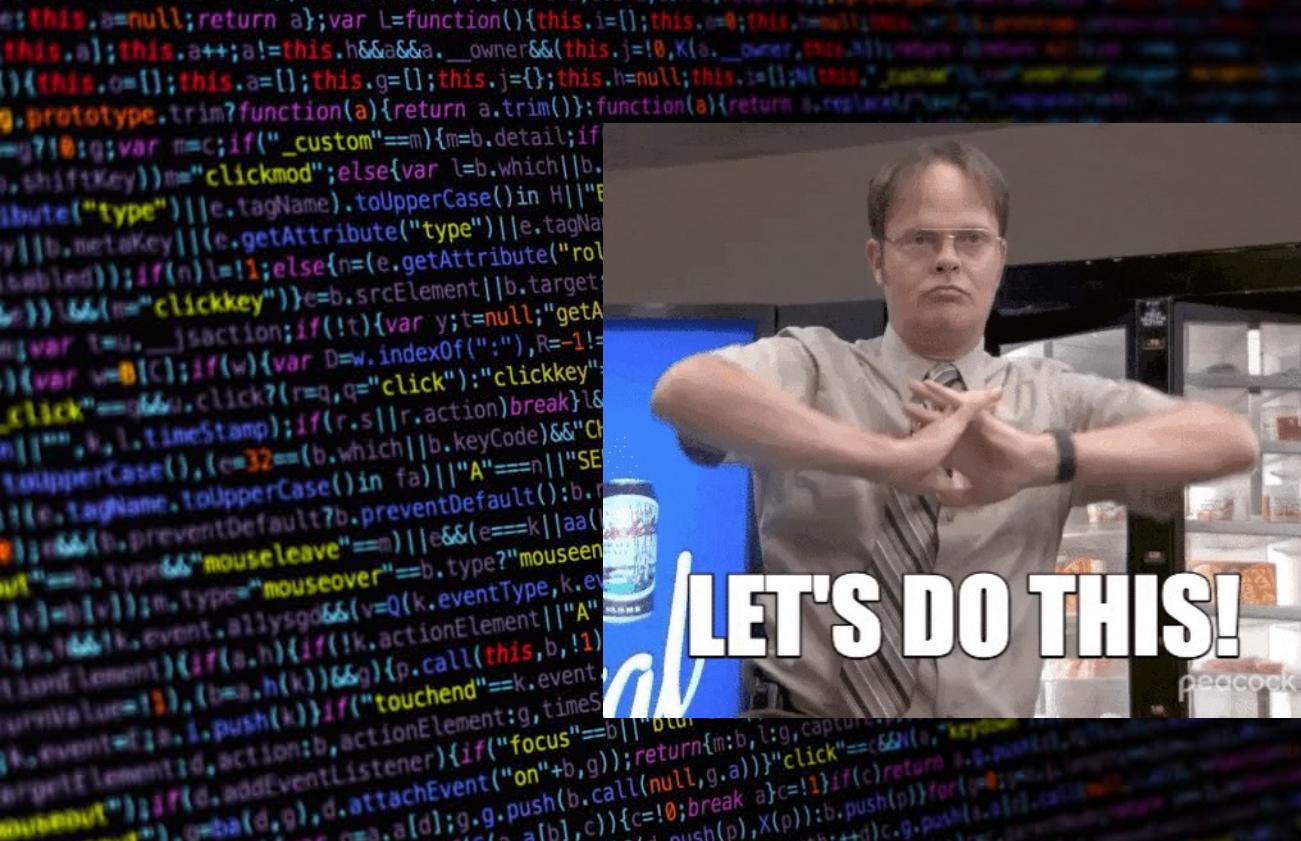
AWS ECS



Amazon
SQS



Demo time v2.0



Just one more thing.... Execution Tags

Issues & Alerts (5)					Alerts Configuration
	Issue	Occurrences	Last Occurrence ↑	Assignee	Muted
<input type="checkbox"/>	Programmatic Error wild-rydes-app-purchaseNewUnicorn us-west-2	8	11 minutes ago	-	
<input type="checkbox"/>	Timed out wild-rydes-app-purchaseNewUnicorn us-west-2 Task timed out after 6.01 seconds	3	5 minutes ago	-	
<input type="checkbox"/>	TypeError wild-rydes-app-postToSocial us-west-2 details.rideID cannot be an empty string	8	3 minutes ago	-	
<input type="checkbox"/>	400 Client error response wild-rydes-app-sumRides us-west-2 {"Error":"Cannot read property 'M' of undefined"}	17	2 minutes ago	-	
<input type="checkbox"/>	TypeError wild-rydes-app-uploadReceipt us-west-2 Cannot read property 'Name' of undefined	33	2 minutes ago	-	



Take Aways

- Always Build for Scale
- Rinse, Repeat and Refine
- Monitor and Trace Everything
 - APIs
 - SDKs
 - Especially Deployed Apps



Take it for a spin

github.com/developersteve/Schrodingers-Python-App

main	1 branch	0 tags	Go to file	Add file	Code
DeveloperSteve Coochin and DeveloperSteve Coochin added Lumigo token env vars in... 8221df1 on Oct 17 8 commits					
📁 db	added httpstat command				last month
📁 images	added screenshot				2 months ago
📁 templates	initial				2 months ago
📄 Dockerfile	initial				2 months ago
📄 LICENSE	initial				2 months ago
📄 README.md	added Lumigo token env vars in readme				last month
📄 app.py	added httpstat command				last month
📄 requirements.txt	initial				2 months ago



USE YOUR
TECH SUPER
POWERS
FOR GOOD



"Be excellent to each other"



Thank You