

Lightrun

FEB 2022

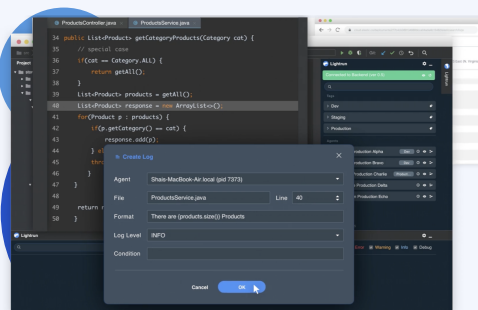
# Lightrun - Product Overview

# Lightrun - A Developer-Native Observability Platform

Lightrun builds an **IDE-Native** observability platform that enables developers to securely add logs, metrics and traces to live applications **in real time, on demand**. No hotfixes, redeployments or restarts are required.

Developers use Lightrun for multiple code-level observability needs, including:

- ✓ Code-level alerts
- ✓ Feature verification as part of progressive delivery (blue/green and feature flags)
- ✓ Testing / Debugging of live applications, without redeployment
- ✓ Code flow investigation
- ✓ Troubleshooting Microservices/Serverless/Big Data workers
- ✓ Pinpointed performance analysis



Lightrun eliminates the need to reproduce bugs locally or issue a new software version in order to add new logs or metrics to troubleshoot production issues. As a result, Lightrun's customers consistently **reduce their MTTR by up to 50-60%** and significantly improve development productivity. Issues that used to take 1-2 weeks to mitigate now take our customers on average less than an hour to solve.

Lightrun empowers our customers' developers by **eliminating the need** for costly developer lifecycle operations like reproducing locally, or issuing a new software version just for adding new logs or metrics.

With customers varying from Fortune 500 (including F10 companies) to SMEs and SMBs across multiple verticals, Lightrun is continuously shifting-left the observability landscape and empowering developers through strategic partnerships such as with IBM, HashiCorp and Dynatrace. Lightrun was named a 2021 Gartner Cool Vendor in Observability.



**Lightrun is backed by Insight Partners & Gliot Capital Partners.**



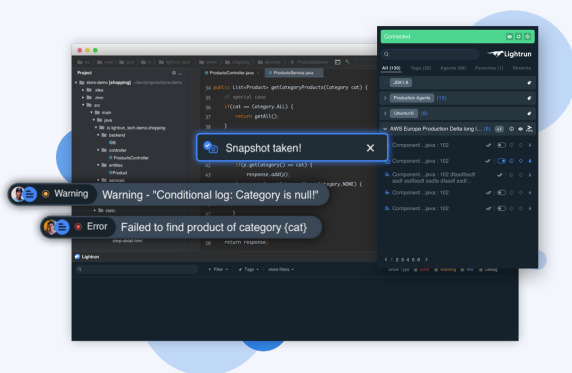
## Lightrun can be used for the following use cases:

- ✓ Debugging live applications both for production and pre-production environments
- ✓ Developers code level alerts
- ✓ Verifying new features
- ✓ Testing in progressive delivery workflows
- ✓ Performance analysis with metrics on demand
- ✓ Onboarding new developers with legacy applications
- ✓ Code flow analysis
- ✓ Debugging 3rd party or OS code
- ✓ Data-centers which allow cloud/microservices migration
- ✓ Debugging ML pipelines and Big Data infrastructure



# Full Observability

Lightrun is an observability platform which empowers developers to observe and troubleshoot live applications, without redeployment. It allows developers to easily and securely add logs, metrics and traces to development, staging and production environments in real-time and on-demand.



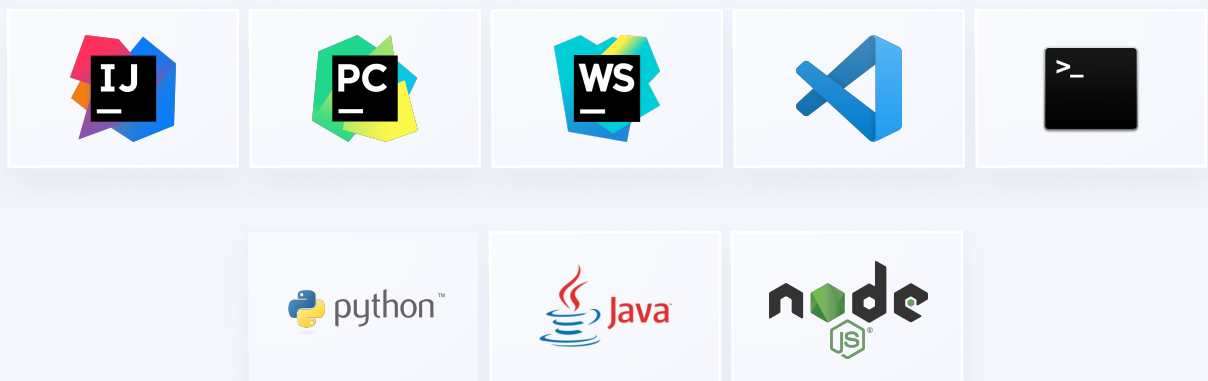
Lightrun can be used for numerous needs, including:

- ✓ **DEBUGGING**
- ✓ **CODE-LEVEL ALERTING**
- ✓ **FEATURE VERIFICATION**
- ✓ **PINPOINT PROFILING**

## SUPPORTED TECHNOLOGIES

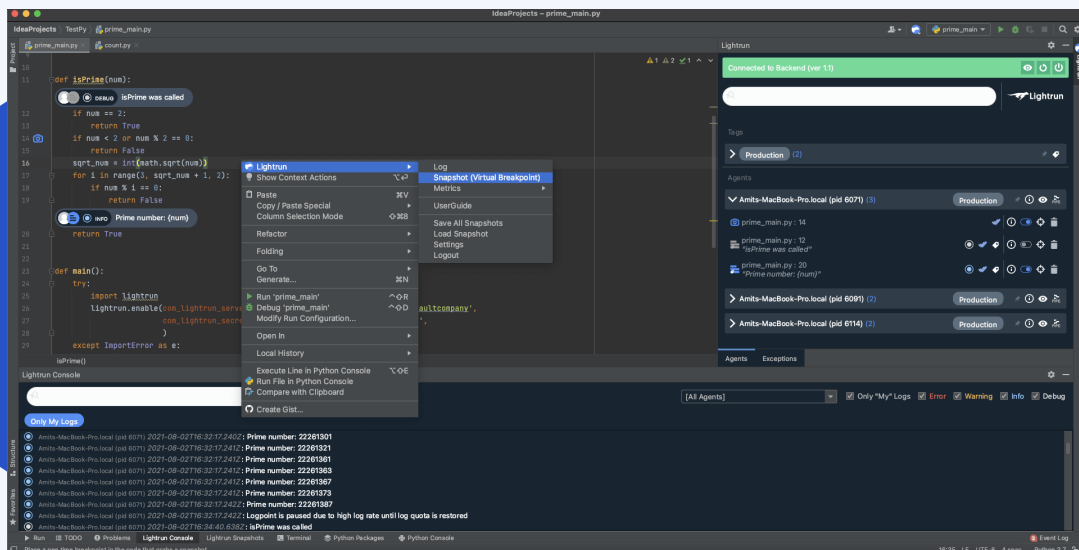
# Lightrun Supports Your Runtime

Lightrun offers rich plugins for popular JetBrains IDEs and VS Code for all supported languages, as well as a CLI for programmatic Lightrun usage. In addition, Lightrun is deployment-agnostic - we work wherever our supported runtimes do.



# Developer-Native

Lightrun enables developers to troubleshoot, test and debug their applications by investigating issues **directly in the environment** in which they occur, including development, staging and production environments.



# Handling Unknown Unknowns

Lightrun enables developers to observe and tackle issues **they did not anticipate and account for during development** - so-called unknown-unknowns - on the fly, and offers more observability without redeploying or restarting the live application.

# Lightrun Introduction

The Lightrun plugin's interface is intuitive and fits right into the developer's workflow - **Lightrun Actions** are created from inside the editor's window, and information is presented inside rich, interactive sidebars and consoles.

1

Choose a line of code you would like to get more runtime information about

2

Add logs, metrics and traces to running service from IDE (without stopping or slowing it down)

The screenshot illustrates the Lightrun IDE interface. On the left, a project tree shows the file structure. The main editor displays Java code for `ProductsController.java` and `ProductsService.java`. A blue dashed line points from the code to the Lightrun sidebar on the right. The sidebar shows the Lightrun status (Connected to Backend (ver 0.5)) and a list of agents (AWS East Production Alpha, AWS East Production Bravo, AWS West Production Charlie, AWS Europe Production Delta, Digital Ocean Production Echo). Below the sidebar, the Lightrun Console shows a log entry: "192.168.0.33 (pid 5089): There are 10 Products". A blue dashed line points from the console to the Lightrun Logs window on the right. The Logs window shows a table of logs with columns: Timestamp, Level, Instance / Zone, and Message.

Timestamp	Level	Instance / Zone	Message
Mar 19, 2020, 12:00:00 PM UTC	INFO	iD@us-east...	There are 10 Products
Mar 19, 2020, 12:30:03 PM UTC	INFO	iD@us-east...	[instance-0000000000] snapshot [found-acm8mmgar7mqhg1bpqa8eg/JWeRdEY3F
Mar 19, 2020, 12:00:00 PM UTC	WARN	iD@us-east...	Category is SHIRTS
Mar 19, 2020, 12:30:00 PM UTC	INFO	iD@us-east...	[instance-0000000000] snapshot lifecycle creation for [cloud-snapshot-2020.03.19-
Mar 19, 2020, 12:00:00 PM UTC	ERRO	iD@us-east...	Failed to find product of category SHOES
Mar 19, 2020, 12:00:03 PM UTC	INFO	iD@us-east...	[instance-0000000000] snapshot [found-1rx2atsqfmlfvmcawhqwWweNf4wRL

3

See info immediately at runtime, in your IDE or preferred integration

# Introducing: Logs

We've all felt the frustration and irritation of missing a log in the exact place we needed one. Lightrun enables you to add logs in **real-time** - while the service is still running. No need to release a new version to see the added log data.

**Create Log**

Add a line of log before the current code line

Agent: Toms-MBP (pid 3914)

File: ProductService.java 77

Format: Category: {c}

Condition:

Log will appear in your application logs

Buttons: Cancel, Advanced, OK

**Logs**

Timestamp	Level	Instance / Zone	Message
Mar 19, 2020, 12:00:00 PM UTC	INFO	id@us-east...	Category: SHIRTS
Mar 19, 2020, 12:30:03 PM UTC	INFO	id@us-east...	[instance-0000000000] snapshot [found-acm8mmgar7mnghtopqs8egj/WWeRDEY3P
Mar 19, 2020, 12:00:00 PM UTC	WARN	id@us-east...	Category is SHIRTS
Mar 19, 2020, 12:30:00 PM UTC	INFO	id@us-east...	[instance-0000000000] snapshot lifecycle creation for [cloud-snapshot-2020.03.19-
Mar 19, 2020, 12:00:00 PM UTC	ERROR	id@us-east...	Failed to find product of category SHIRTS
Mar 19, 2020, 12:00:03 PM UTC	INFO	id@us-east...	[instance-0000000000] snapshot [found-1rxz8atsqfmlavmscawh9j/WWeRDEY3P

**Agent**

A Lightrun agent is attached to your running production service at all times.

**File**

Current source code file.

**Expressions**

Add expressions to be evaluated at runtime alongside the rest of the trace, if you need to see something specific.

**Conditions**

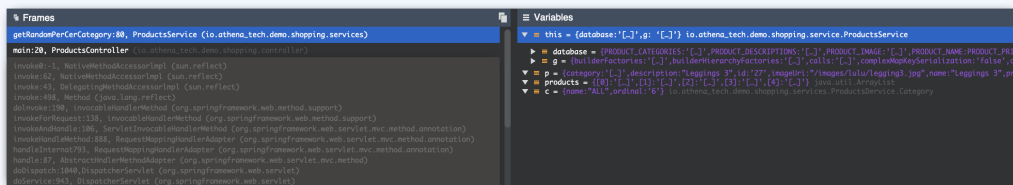
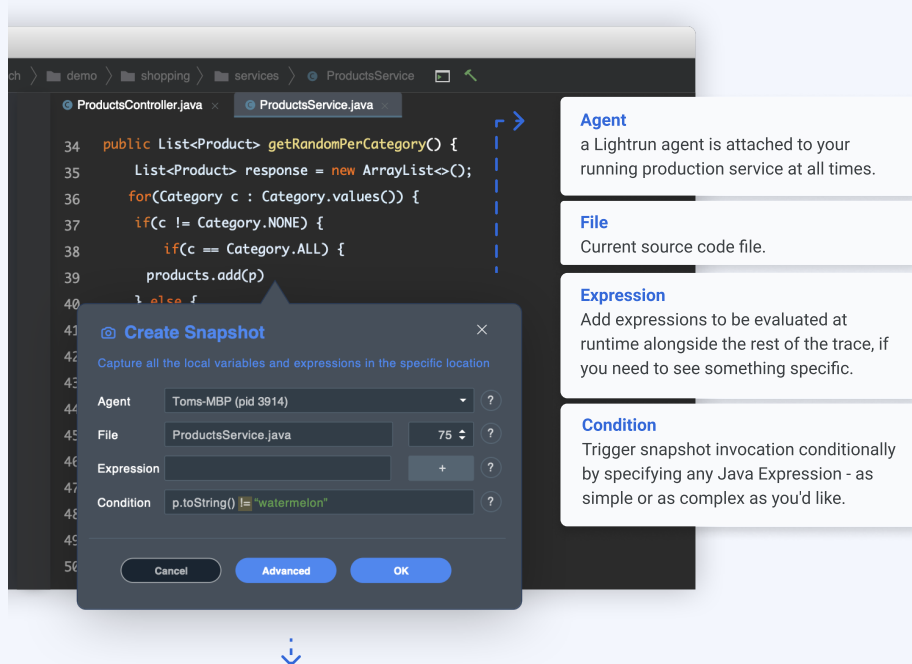
Trigger snapshot invocation conditionally by specifying any expression - as simple or as complex as you'd like.

## You Can:

- Troubleshoot live applications easily by dynamically adding log lines
- Add as many logs as you need until you identify the problem
- All logs are added to runtime immediately
- Add logs with the click of a button
- Add logs to multiple instances of a running production service with our tagin mechanism

# Introducing: Snapshots

When you need to explore your stack trace and variables thoroughly, add a Lightrun snapshot in **real-time and on-demand**. This virtual breakpoint extracts all the data you need for investigating **without pausing the process**.

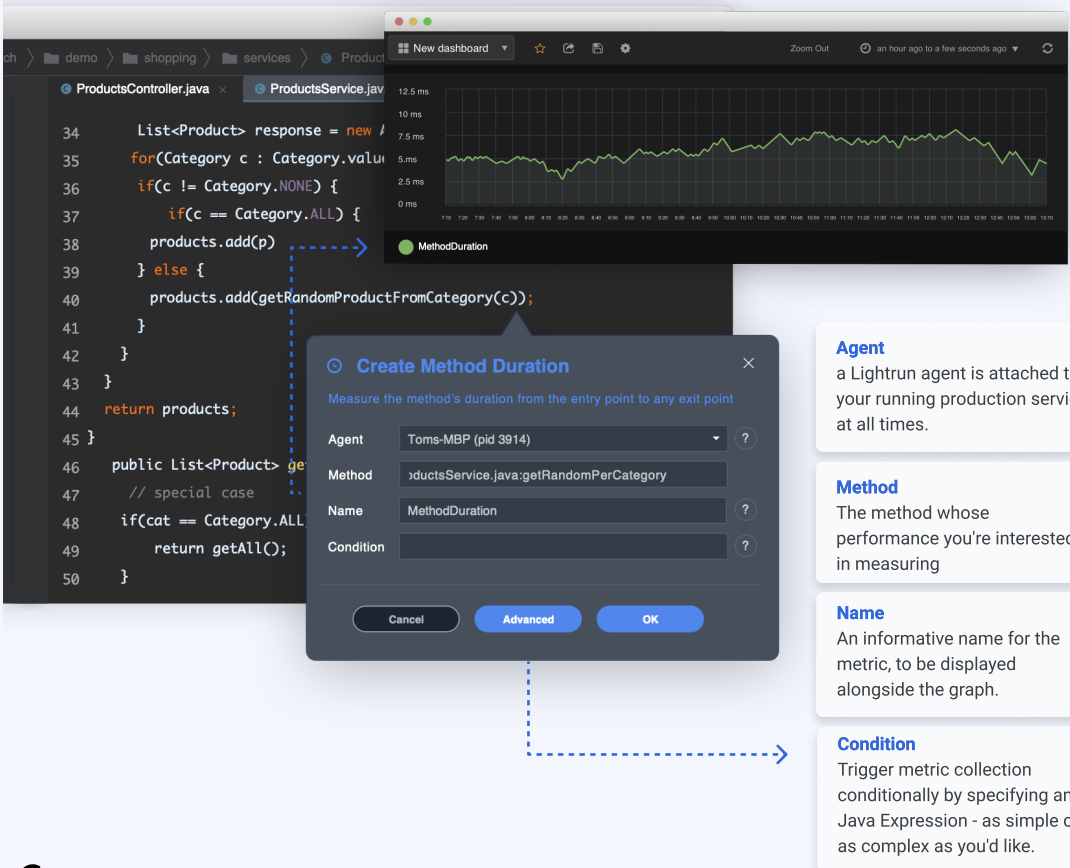


## You Can:

- Explore variable values and arguments at runtime, for every line of code in your application
- Explore the stack trace and the variables at **every frame**
- Add as many snapshots as you need without pausing the process
- Add **Watch Expressions**
- Tagging mechanism enables **multi-instance support** (microservices, big data workers)

# Introducing: Metrics

Measure any performance, synchronization or business logic metric in **real-time**. Get answers immediately to find bottlenecks without affecting system performance.



**Agent**  
a Lightrun agent is attached to your running production service at all times.

**Method**  
The method whose performance you're interested in measuring

**Name**  
An informative name for the metric, to be displayed alongside the graph.

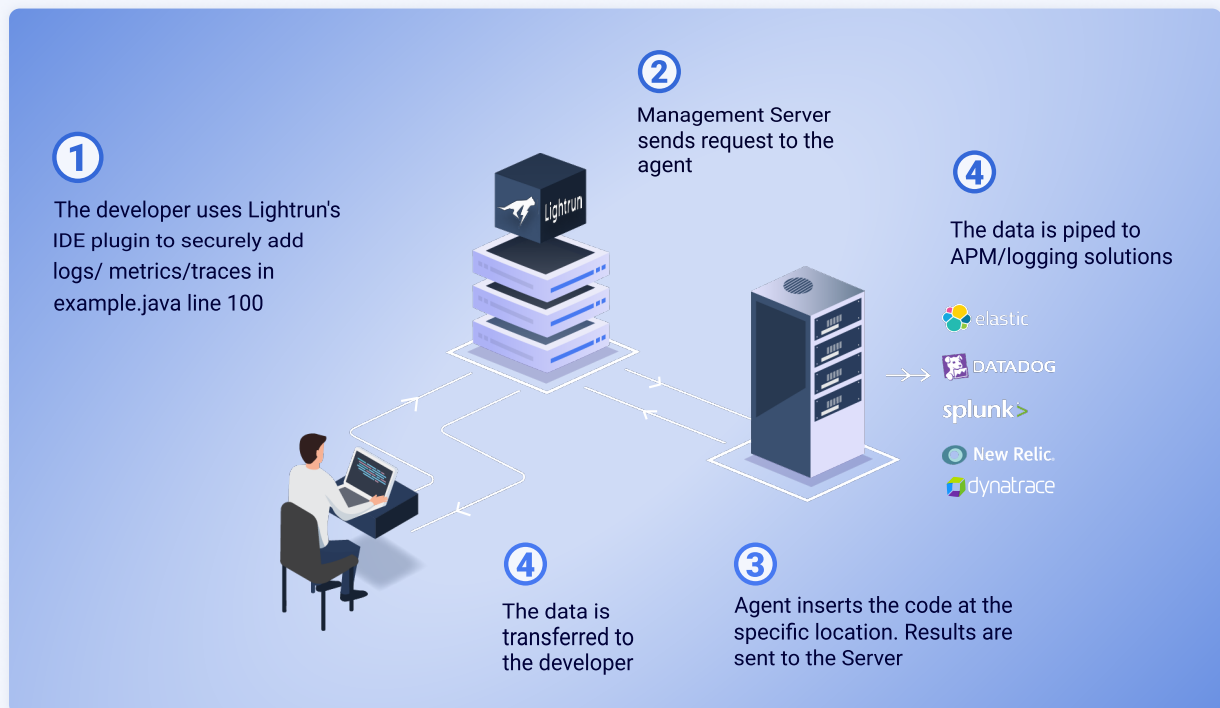
**Condition**  
Trigger metric collection conditionally by specifying any Java Expression - as simple or as complex as you'd like.

## You Can:

- Add metrics on-demand until you identify the issue while the app is running
- **Performance metrics** - use counters, timers, function durations and more on-demand
- **Custom metrics** - define any numerical Java expression you want to visualise
- Count line execution occurrences
- Collect system statistics like latency and throughput through counters and timers
- Find performance and synchronization issues easily in the actual version you released, not a local simulation

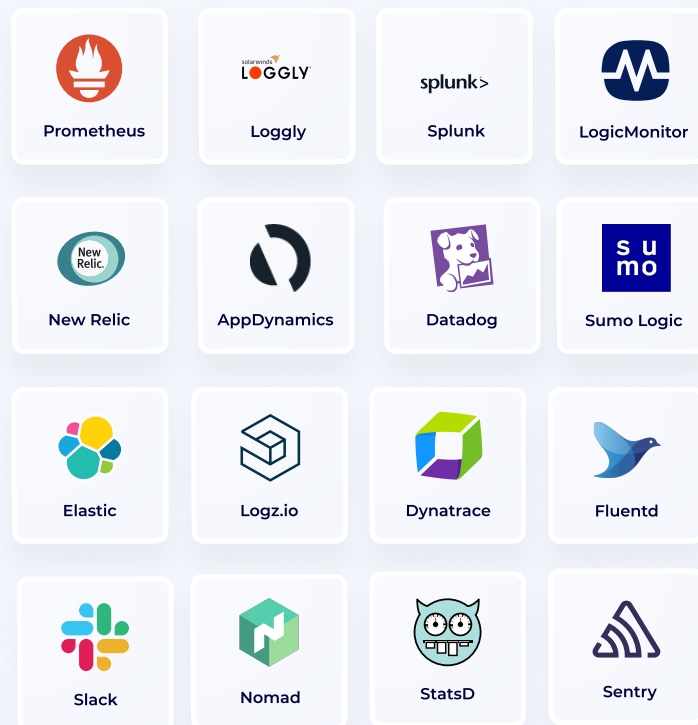
# System Topology & Infrastructure

Lightrun is a 3-part system, comprised of Lightrun Plugins (installed on developer computers), Lightrun Agents (installed on live applications) and the Lightrun Server. Lightrun Server is offered in 2 flavors: **Lightrun Cloud** (our hosted solution) and **Lightrun On-Prem** (for on-prem installations and airgapped networks).



## Lightrun Integrates With Logging & APM Solutions

Lightrun is integrated into the developer workflow, from the IDE to the logging tools and APM. We support multiple integrations to ensure the development and debugging process is frictionless.



## Lightrun Runs Where You Do

Lightrun operates everywhere and anywhere: on-premise, in the cloud [AWS, GCP, Azure], for microservices, for serverless, K8s, and more. Debug in any environment across any infrastructure.

