

# The Developer Observability Stack

--

Observability isn't just for production. Software engineers need the capabilities of observability in order to optimize all phases of the software development lifecycle – from local development environments to hyper-scale shift-right production ones.

## Observability and Performance Management

Observability has become a cornerstone technology for software engineers to continuously improve and maintain their code stacks. Observability provides indicators of the health and status of applications and infrastructure inside cloud-native, hybrid, and on-premise applications. Observability includes metrics and end-to-end traces, with context for elements like containers, microservices, orchestration tools like Kubernetes, and serverless.



### Instana

*Metrics, Tracing, Alerting, Dashboards*

The Open Source tools listed provide a great starting point for fresh services without real load. As we move our services into integration with our existing legacy applications and the full breadth of our application topology, more robust automation becomes indispensable. Instana's AutoTrace and unsampled trace collection give us complete visibility into the network calls between our services, allowing us to easily debug and troubleshoot distributed applications.



### Lightrun

*Production Profiling & Debugging*

Debugging using editor tools is one of the most important skills a new software engineer will develop. But now we work on dozens of services and some of them may never even run on our local machines. At some organizations like twitter, there isn't even a staging environment. With so much testing "shifted right", Lightrun's ability to instrument, profile, and troubleshoot production applications on-the-fly becomes an invaluable asset.

# Server & Application Software

To truly leverage observability, we need the ability to monitor our services from inception to release. At each stage, our stack requires instrumentation, an agent or collector, a backend and an accompanying UI. Full observability platforms are complex and cumbersome to run in small local environments. Instead, we can leverage OpenTelemetry instrumentation to ensure observability in all stages of our application life cycle, regardless of the backend.

	Local	Pre-Production + Production
Service Instrumentation	Open Telemetry Libraries	OTel Libraries + Instana Sensors
Host Agent	Prometheus	Instana Agent
Host Collector	OpenTelemetry	Instana Agent
Tracing Backend	Jaeger	Instana
Logs Backend	ELK	Instana
Metrics Backend	Prometheus	Instana

## Open Source Observability Toolchain

Shift-left observability reminds us to consider observability from the earliest stages of the application development lifecycle. Open source tools allow us to ensure we are gathering valuable telemetry from any service, even before it has ever been deployed outside of a local development environment.

OpenTelemetry provides instrumentation libraries for most popular languages. Jaeger and Prometheus can easily be run locally inside docker containers in order to validate trace and metrics data respectively from services being actively developed.



### OpenTelemetry Collector & SDKs

*Instrumentation & ETL Pipeline for Telemetry Signals*

The OpenTelemetry Collector sits at the heart of your observability pipeline. Sitting in-between your application telemetry and your backends, the collector receives metrics, traces, and logs before processing and forwarding them to your choice of backends.



### Jaeger

*Open backend for querying and viewing traces*

Jaeger provides capabilities to store, query, and view trace data. Jaeger can be used with in-memory storage or with a storage backend like Elasticsearch or Cassandra.

Using end-to-end distributed tracing in development can improve developer understanding of inter-service communication and can dramatically reduce troubleshooting and debugging time.

While in-memory storage works great for local and development debugging, for production workloads using Elasticsearch as a persistence layer is recommended; Alternatively, you could turn to an enterprise-ready tool like Instana.



## Prometheus

*Open backend for metrics*

Metrics provide vital insight into business, application, and infrastructure health – but storing them can be difficult without the right tools. Prometheus has emerged as the de-facto standard for metrics scraping and storage. When you're ready to move to production, you can keep your Prometheus instance alongside a full observability platform like Instana, or you can rely on Instana's ability to scrape and ingest Prometheus metrics directly from your monitored services.



## Grafana

*Dashboards and alerts*

If you're a fan of battle-stations full of large dashboards, then you're probably already using Grafana from Grafana Labs. Grafana provides a mature and open destination for viewing your metrics and logs (and with plugins, also traces). For custom dashboards or "single pane of glass" applications it is an excellent tool. For automatically created dashboards without requiring configuration, you can use Instana's Application Perspectives and entity dashboards.



## Logging

Instana can provide the core logging functionality that most organizations will require. For more complex requirements, an additional solution can be added. ELK and Loki are both excellent open source choices for logging, especially in local and pre-production environments. However, self-hosted data storage can get extremely expensive quickly, and for large volumes of logs a vendor solution such as Instana, Falcon LogScale, or Splunk may offer significant cost savings in addition to all the other benefits.

[Try Instana](#)