INSTANA

Instana Release Notes - Build #115

29. September 2016

Features

Cross-Language Tracing

Calls are now traced end-to-end despite the language processing them (e.g. Node.js->PHP->Java).

Node.js Tracing

Today we are extending our tracing support to Node.js!
Tracing is achieved by embedding the Instana Node.js sensor within your application for low impact and powerful insights into your applications' behavior and performance.

Supported Node.js Versions

- 4.5+
- 5.10+
- 6.0+

Usage

Usage of the Node.js tracing feature is easy. Add the `instana-nodejs-sensor` dependency to your Node.js application and activate it as the very first line within your application.

Installation:

```
npm install --save instana-nodejs-sensor
```

Activation:

```
require('instana-nodejs-sensor')();
```

Documentation

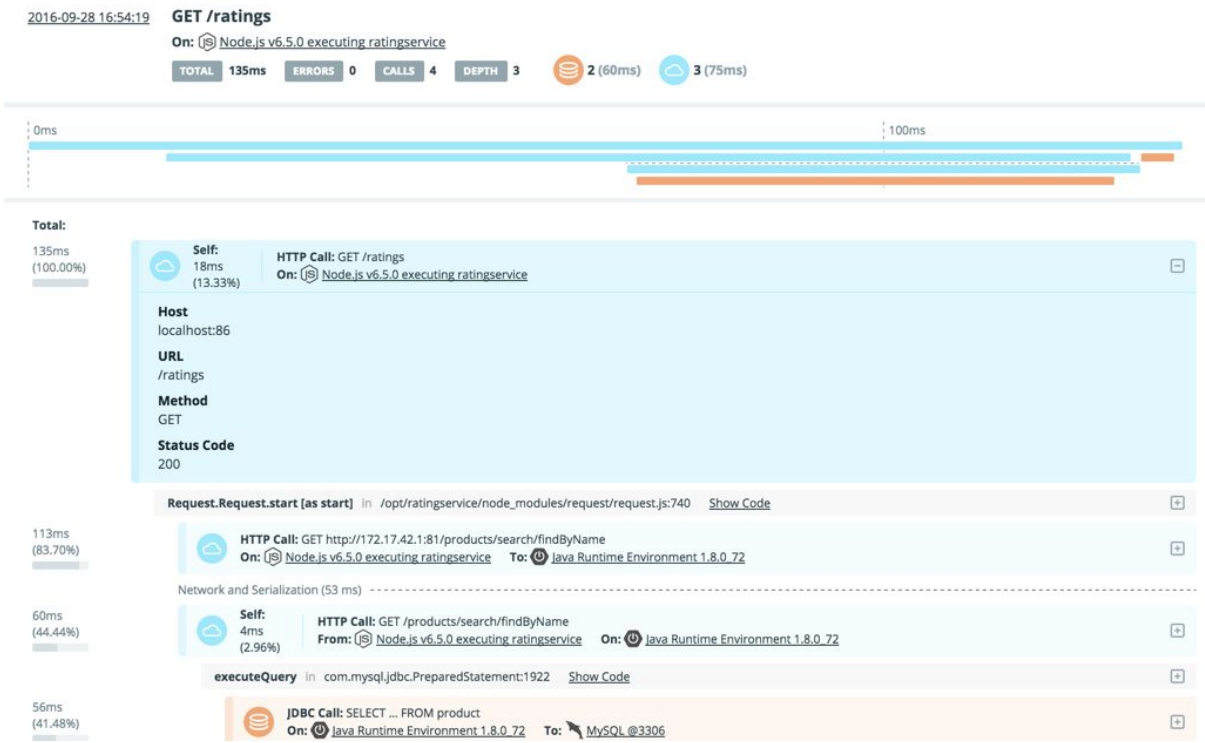
For full documentation on the Node.js sensor, please refer to:
https://instana.atlassian.net/wiki/display/DOCS/Node.js

Supported Frameworks and Technologies

Category	Supports	Modules
HTTP	Entry and Exit	http https express hapi
Elasticsearch	Exit	elasticsearch
MongoDB	Exit	mongodb mongoose

We are aware of issues arising through the usage of various libraries to handle asynchronous control flow. At the moment, we track calls across timers and native Promises.

Node.js traces are shown in the Trace View as other supported languages, and work cross language.

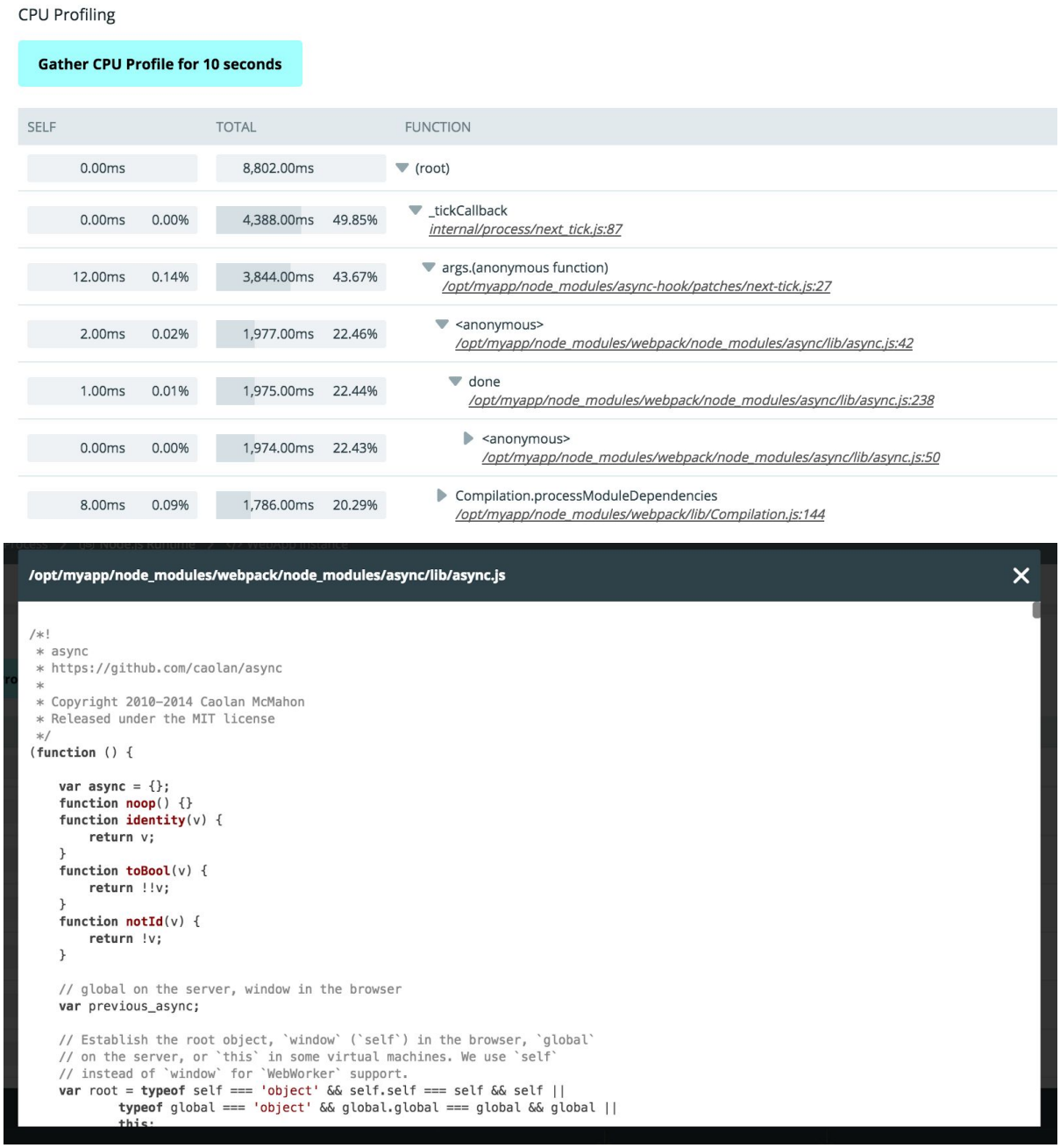


Stack traces and the drill down into the application code are available.

If you click on “show code” Instana will instantaneously fetch and inspect the code:

Node.js On Demand Profiling

Long running tasks and slow functions are highly problematic in Node.js applications as these block execution of other tasks. This is due to Node.js' single-threaded application architecture. Identifying slow functions is therefore important and typically requires you to attach a profiler to running applications. This is often problematic, especially in production environment. Now, you can temporarily switch any Instana-monitored Node.js application into profiling mode with the click of one button. At the moment, we are limiting this profiling period to 10 seconds. Once these 10 seconds are over, you will be presented with information about slow functions and the ability to directly inspect the code of these slow functions!



PHP Tracing Enabled by Default

This is now enabled by default. Should you want to deactivate it, turn it off in configuration.yaml:

```
com.instana.plugin.php:
  tracing:
    enabled: false
```

Agent Self Monitoring Exposed via UI

Not always everything is running smoothly. That is the reason why we built in the UI the ability to monitor the Instana agent itself. Simply open up any Host Dashboard, at the bottom you can activate the ‘self monitoring’ of the Agent. It will show you key metrics and will tail the log preventing you from logging into the server to retrieve the same information.

Monitor Any Process

Until now Instana did process monitoring only for processes running on a supported technology. Now it is possible to configure additional process names and Instana will then monitor them. You can configure the processes you want to monitor in the agent configuration.yaml.

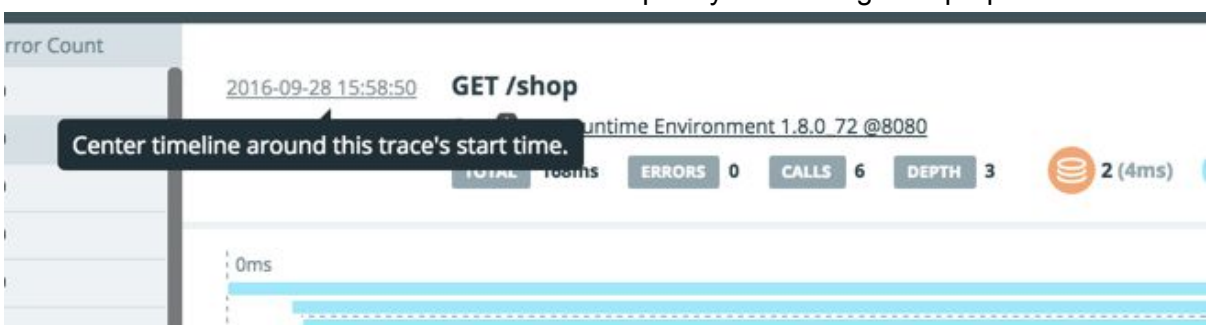
```
# Custom processes
# To enable low level metric and availability monitoring for not automatically
# discovered processes, uncomment the com.instana.plugin.process section and
# list all process names of processes that should be monitored.
#com.instana.plugin.process:
#  processes:
#    - 'sshd'
#    - 'slapd'
```

Switch to UTC Timestamps

In the Settings of your account (top right corner of your screen) you can now switch to UTC mode. All timestamps in Instana will be formatted according to UTC. By default, Instana formats the time according to the browser’s default timezone.

Jump to Moment of Trace

Just click on the timestamp of each trace and set the timepicker to that exact moment: the entire environment will be shifted to that timestamp for your investigation purposes.



Newly supported technologies

- **Varnish Sensor**
- **Glassfish Sensor**
- PHP Tracing: IBM-DB2
- Java Tracing: Java Mail, Nanohttpd, Mule, Vaadin, Camel, (S)Ftp, Wicket

Improvements

- You now can jump from the physical component directly to the implemented logical services and service instances.
- Improved compatibility with Docker for PHP tracing, PHP-FPM, HTTPd, MySQL, MariaDB sensors.
- PHP sensor now supports Plesk Installations.
- Tracing now excludes commands for database drivers which are not actually executed (like “set names utf-8;”).
- It is now possible to add specific queue names to RabbitMQ sensor, which will then be monitored.
- PHP-FPM sensor now supports multiple master processes.
- MySQL metric collection is now much more efficient.
- Cassandra sensor now captures replication factor.
- Updated the Apache Karaf Framework shipped as part of the agent installation for reduced memory consumption. To apply this fix, please re-download the agent package. This framework is not updated automatically.
- Support Kubernetes Installations with flannel networking.
- Services will only be discovered after a successful external call.
- Presence detection improved to better support ephemeral components.
- Faster event fetching and processing.
- Opening timepicker now shows the previously selected time.
- Improved RabbitMQ and Oracle DB service detection.

INSTANA

Fixed Issues

- Fixed typo in Elasticsearch dashboard.
- Prevent hanging Docker exe processes started by discovery.
- JBoss Data Grid sensor did warn frequently about missing MBean attributes on old versions.
- PostgreSQL sensor only warns once when stats tracking is off.
- MySQL and MariaDB sensors did not respect wildcard bind addresses.
- Httpd dashboard did visualize load incorrectly (off by factor 100, 916% instead of 9.16%).
- Google Compute Engine will be discovered correctly.
- High tracing load could cause the agent to go out of memory. This has been improved.
- Windows agent crashed after a while in rare scenarios.
- RabbitMQ detection did not work in rare scenarios.
- Fixed frequent disconnects of the agent.
- Long span labels prevented click on span in icicle graph.
- Long labels in entity breadcrumb prevented click on entity.
- Async root span timing was inaccurate.
- Metric gaps when reopening dashboards are fixed.

Known Issues

- We are still working on reducing the memory consumption of the agent process when receiving many traces under high load. As a temporary workaround, customers can give the agent more memory by increasing JAVA_MAX_MEM in bin/setenv.
- Tracing can fail to instrument in certain versions of IBM J9. This can be worked around by starting the JVM with an empty javaagent. See https://github.com/instana/instana-javaagent/blob/master/README.md for more information.

Please send feedback via Slack or email to michael.krumm@instana.com about the new features, the improvements and especially your ideas and other points you have. We have a lot cooking so stay tuned - more to come soon!