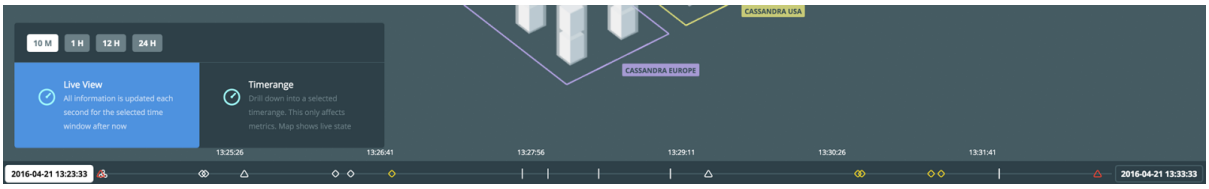
Instana Release Notes - Build #106

09. May 2016

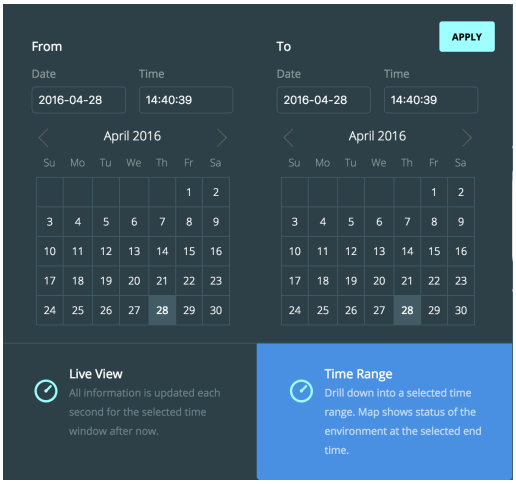
Features

Timeshift - first step

Until now Instana was limited to investigate timeranges up to 24h in the past. We put a lot of work to change this and are proud to release the capability to go back in time and select time ranges. This is a first step as we are working to improve our capabilities and the usability beyond this first step to provide you with a more powerful and interactive way to investigate the past and move through time.
In this first step we introduce a new timepicker:



You open it by clicking on the left or right date/time of the timeline.
You can select the already known live view with the range of time you want investigate from now into the past.
If you select Time Range you will be presented with a selector



At the moment we limit the range you can investigate a week. This will be extended over the next releases.
When you select a time range the map will show the state (Hosts and their entities, their health status) of your environment at the moment that is selected as the right border/closest to present of the time range shown in the timeline.

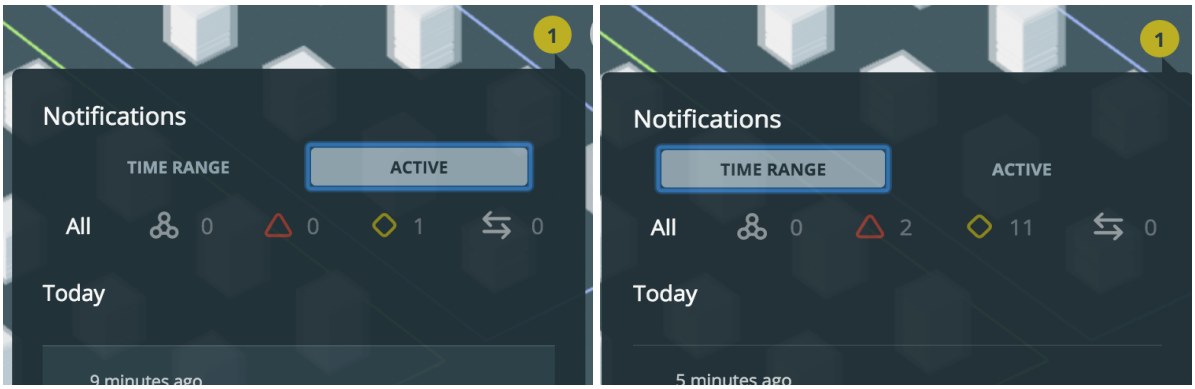
Dashboards will show the metrics of the whole time range.

We store the data in the following roll ups:
- 1 second for data not older than 10 minutes
- 5 seconds for data not older than 24 hours
- 60 seconds for data not older than 1 month
- 300 seconds/5 minutes for data not older than 3 months
- 3600 seconds/1 hour for data older than 3 months

The Dashboards will show data in sensible roll ups that are adjusted to your window size. If you want to drill into a time range in more detail: please narrow down the selected time range (we are working on improving this interaction).

As we just start to collect the data historically you will not be able to drill down into the past more than 3 days of the release date. From now on we collect the data over time in roll ups as a first shot forever. This may change over time as we understand your requirements and the technical impact better.

We believe that it is important that you are always informed about what is going on at the moment we also added information to the Status information and Notification overview:



Clicking on Active shows all active issues. Clicking on Time Range shows all Issues (active or not) in the selected time range.

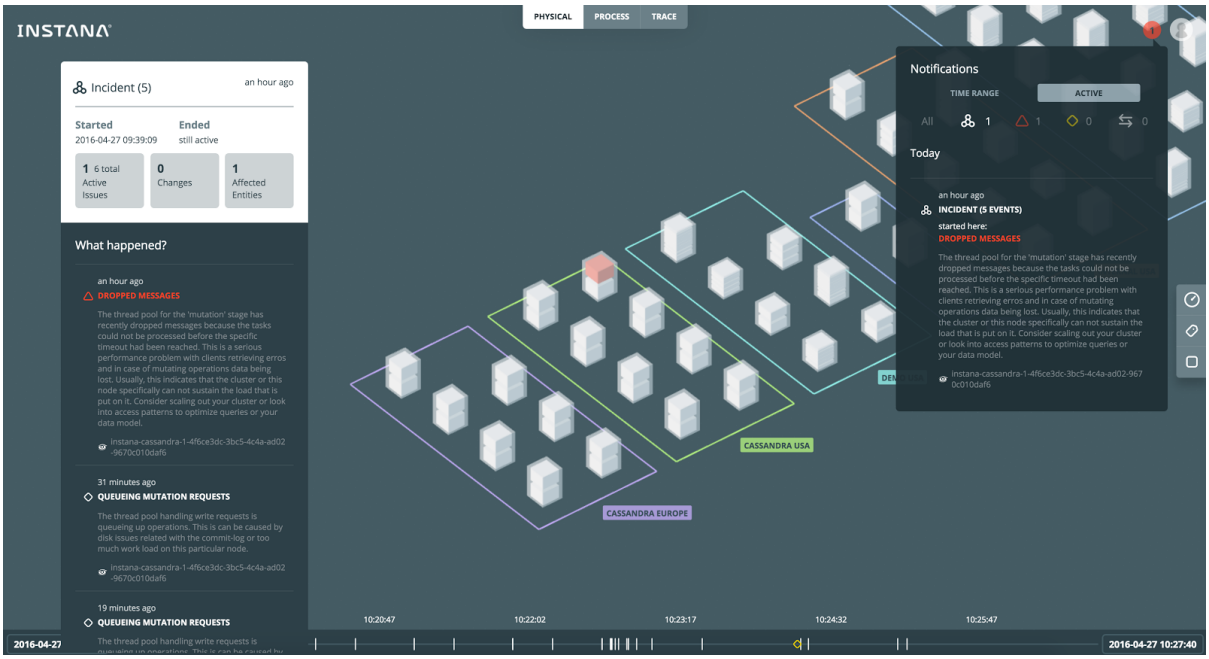
Incidents

Incidents intelligently group issues and changes by taking the dependencies and relations of the affected entities and the appearance in time of the issues into account. This helps to understand the impact and evolution of problems on a new level.

An Incident is detected, when at least two Events (Issue, Change, Online, Offline), where the older event has to be an Issue, that are related via dependencies in the graph, are active at the same time. The moment an Incident is detected we continue to group new issues and changes of the affected entities into it. The Incident is closed the moment there are no more active issues in the Incident.

By this approach we are able to understand dependencies of issues and their evolution. For example if a JVM experiences high Garbage Collection time and this affects the Cassandra Node running in the JVM we will group this into an Incident including changes on the mentioned entities in order to provide insight into this problematic evolution.

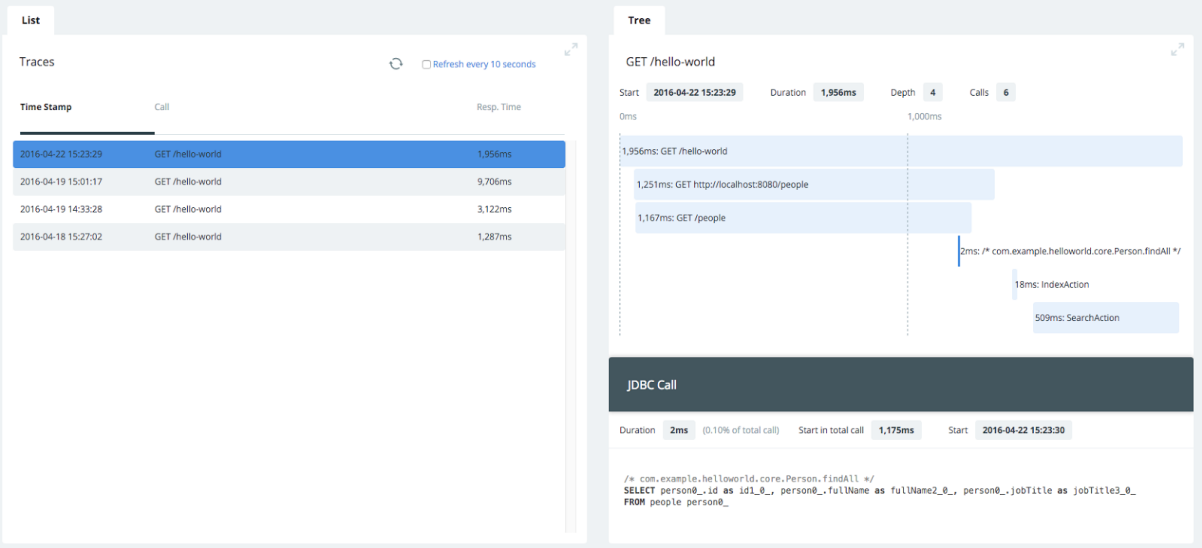
Incidents are shown separately in the UI as a new type of event:

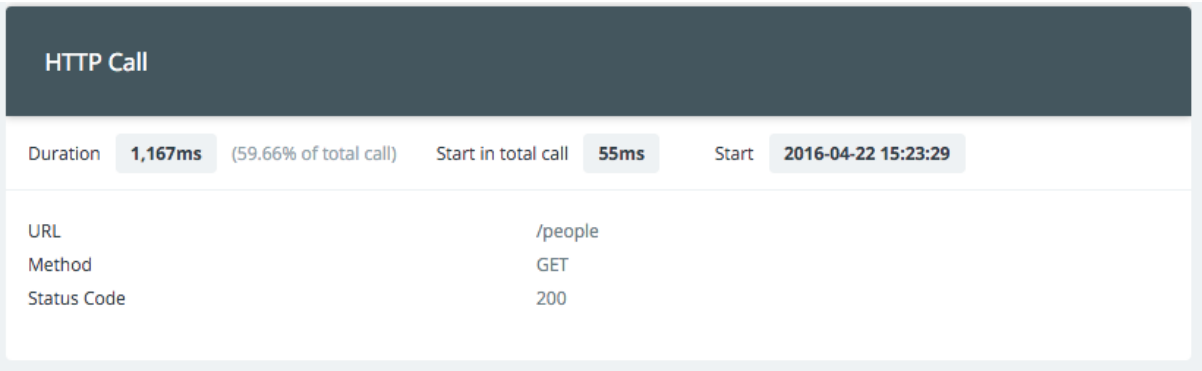


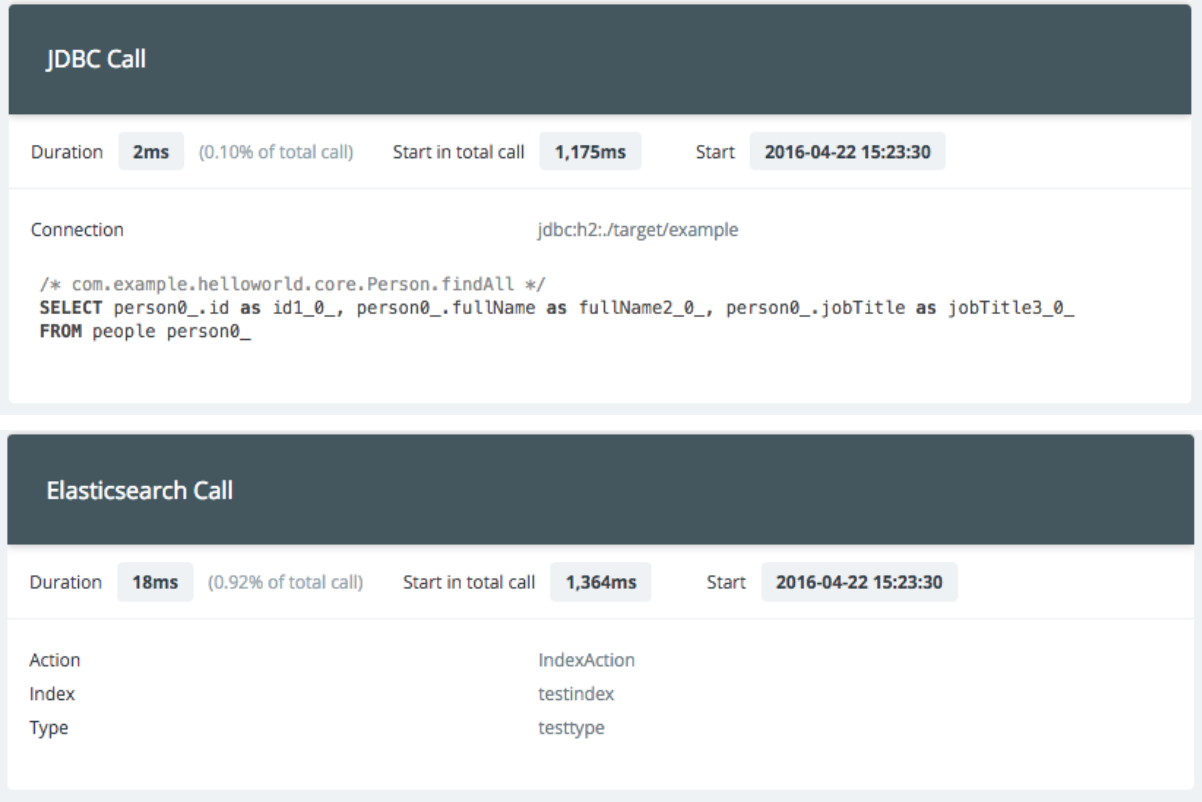
We are very excited to release this first version of Incidents and we will continue to work hard on improving them. Please feedback us how they work for you and what to improve!

Trace View

We also introduce our **Java** Tracing capabilities with this release in a first version.
Our Agent automatically attaches the tracing sensor to JVMs and traces calls using a very lightweight bytecode instrumentation approach. A trace is collected if an entry point (currently supports only HTTP) is hit. We time the execution of that call and also trace subsequent calls into other entities via exit points that we instrument (at the moment HTTP, JDBC, MongoDB, Cassandra and Elasticsearch). If an exit point is hit that enables distributed tracing (at the moment only HTTP) we tag it with an ID and correlate it and all subsequent calls into the whole call if the downstream JVM is also instrumented with Instana.
More information of calls (e.g. SQL statements for JDBC calls, HTTP Calls for HTTP) are collected and displayed.
Traces are persisted at the moment for 5 days. They are deleted afterwards.
The Trace View displays the following information:







The Trace View will constantly evolve and we will add more and more technologies and information as well as troubleshooting capabilities to it. This enables deep insights into your application behaviour helping you troubleshoot and better understand your Java applications.

Note: Tracing is an important feature to understand application behaviour. Due to the nature of the technology, compatibility to countless frameworks needs to be tested. We do our best to do compatibility testing, but it is possible that certain combinations prevent tracing from working. Please contact us if you are not seeing traces for calls of Type HTTP or JDBC. A detailed support matrix will be part of later documentation.

Note: Tracing can be turned off if desired in the `configuration.yaml` like this:

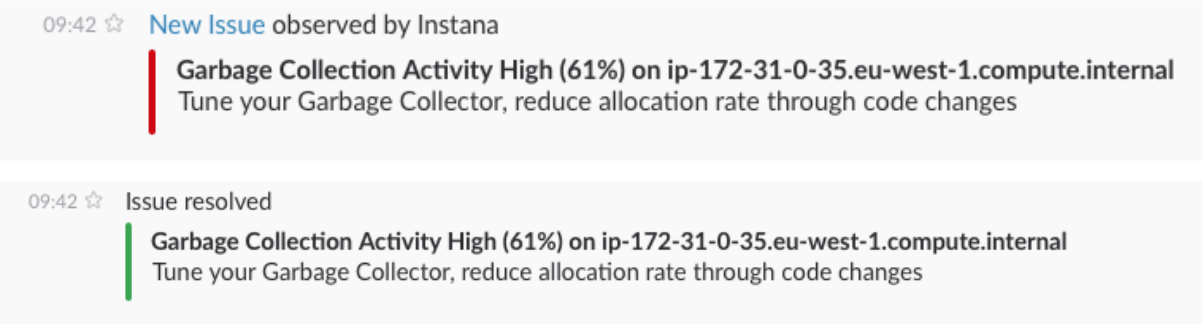
```
# Java Tracing
com.instana.plugin.javatrace:
  instrumentation:
    # Lightweight Bytecode Instrumentation
    enabled: false
```


Webhook Integration

You can now send alerts via a generic Webhook integration.
Configuration is done in the User Management Portal (click on the profile in the top right corner -> click on "Welcome to Instana, ...") then on Integrations.
You can add multiple Webhook URLs. When clicking on Webhook you can configure the URL the events shall be sent to.
We will send:
onOpen:
`{"issue":{"id":"53650436-8e35-49a3-a610-56b442ae7620","state":"OPEN","start":1460537793322,"severity":5,"text":"Garbage Collection Activity High (11%)","suggestion":"Tune your Garbage Collector, reduce allocation rate through code changes","link":"https://XXXXXXX/#/?snapshotId=rjhkZXdNzegliVVEswMScGNn0YY","fqdn":"host1.demo.com"}}`
onClose:
`{"issue":{"id":"6596e1c9-d6e4-4a8e-85fd-432432eddac3","state":"CLOSED","end":1460537777478}}`
coming as http POST to the configured URL. We support http and https.

Slack Notification Integration

You can now send alerts to your Slack instance.
Configuration is done in the User Management Portal (click on the profile in the top right corner -> click on "Welcome to Instana, ...") then on Integrations.
Add the slack webhook url, which you can create in Slack by navigating to “Browse apps > Custom Integrations > Incoming WebHooks”
If not specified the Slack integration will post to the channel configured on the slack side, you can override the channel and user icon by configuring it on the Instana side.



FQDN and Entity Type is sent via integrations

The FQDN and Entity Type is now sent via integrations in addition to the issue information.

Usage/Sensor Instance Minutes Overview

When you click on the profile in Instana and click on your username to go into the User Management Portal you will now see the new Usage Tab. It shows you which Sensors run

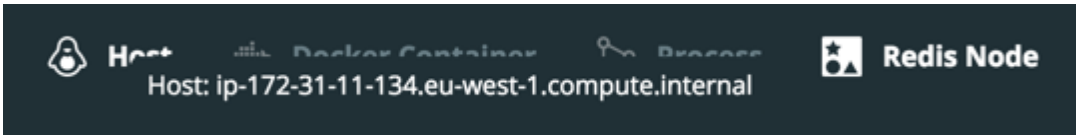
and how many so called Sensor Instance Minutes (SIMs) are used in total per month and per Sensor.
A SIM is used when a Sensor reported data at least once a minute.

Newly supported technologies

- **RabbitMQ**
- **JBoss Application Server**

Improvements

- Massive Elasticsearch Sensor Improvements
  We are now detecting Elasticsearch Clusters, monitor Nodes, Indices and Shards in far more detail. We also added some powerful health signatures that help you improve and troubleshoot your Elasticsearch installation better.
- Massive Redis Sensor improvements
  It now shows more metrics, the slow log and the length of custom keys. Also added health signatures.
- Removed TCP outgoing resets rule because of too many false positives
- Agent now supports HTTP/2 as standard protocol instead of SPDY
- Agent supports pinning the SSL certificate to prevent Man-In-The-Middle attacks.
- Support monitoring a redis server requiring authentication
- On hovering over the Component Hierarchy in the Sidebar and in the Dashboards the label of the Entities are shown (e.g. FQDN for a Host).



Fixed Issues

- Reduced CPU load of agent on linux.
- Fixed Java version detection when Java is version 6
- Tomcat Change Detection detected irrelevant changes
- Windows Sensor Disc Utilization graph fixed
- Compatibility with IIS having unmapped ports defined. Now monitored correctly.
- Graphs show data gaps in Dashboards
- The agent will no longer try to attach to “JVMTop”
- The agent will no longer try to attach to a JVM which has just died. This will reduce unnecessary error log messages.
- Fixed parsing of broken Docker 1.11 memory metrics, will be unavailable until fixed in 1.11.2 by Docker.

Known Issues

- Selecting a Tag in a time range will result in an empty map

- Performance especially with larger time ranges selected with a high count of events needs to be improved - We are already on it and have a solution cooking and in progress.
- Overview of Issue Count in past time ranges does not reflect the active state.

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