USER GUIDE

You'll see right away that Zebrium is different than typical observability tools. Traditional log managers rely on you to know what to search for and build alerts for.

But when you are troubleshooting a new problem, the root cause is most often found in a small number of new/rare events and related errors. In these cases you typically don't know exactly what to search for. Zebrium uses machine learning (ML) to automatically identify these correlated clusters of anomalies and errors, and puts them in RCA (root cause) reports.

All you need to do is connect a stream of logs (and optionally metrics) to Zebrium's ML engine. You can then consume Zebrium's ML generated RCA reports in one of two ways:

1. Connect it to your incident management tool like OpsGenie, PagerDuty or Slack, so that an RCA report is automatically created and sent back to the incident management tool.
2. Or, evaluate the feed of auto-detect incident RCA reports, particularly around times where you know things went wrong. You can also force the ML engine to do a deep scan and create a report on demand by clicking the "Scan for Root Cause" button.

In most cases, the default settings will work well, however, Zebrium provides several controls to adjust the sensitivity of its machine learning engine so you can tune signal / noise to best suit your environment. Zebrium also provides ways to group related log feeds so that the ML only correlates anomalies across related components of an application instance (the components that together define a failure domain. For details on changing the default sensitivity settings and managing deployments, please contact Zebrium or refer to the API section of the online documentation.

If you have any questions, please reach out to us via Slack or email hello@zebrium.com.

Key Concepts

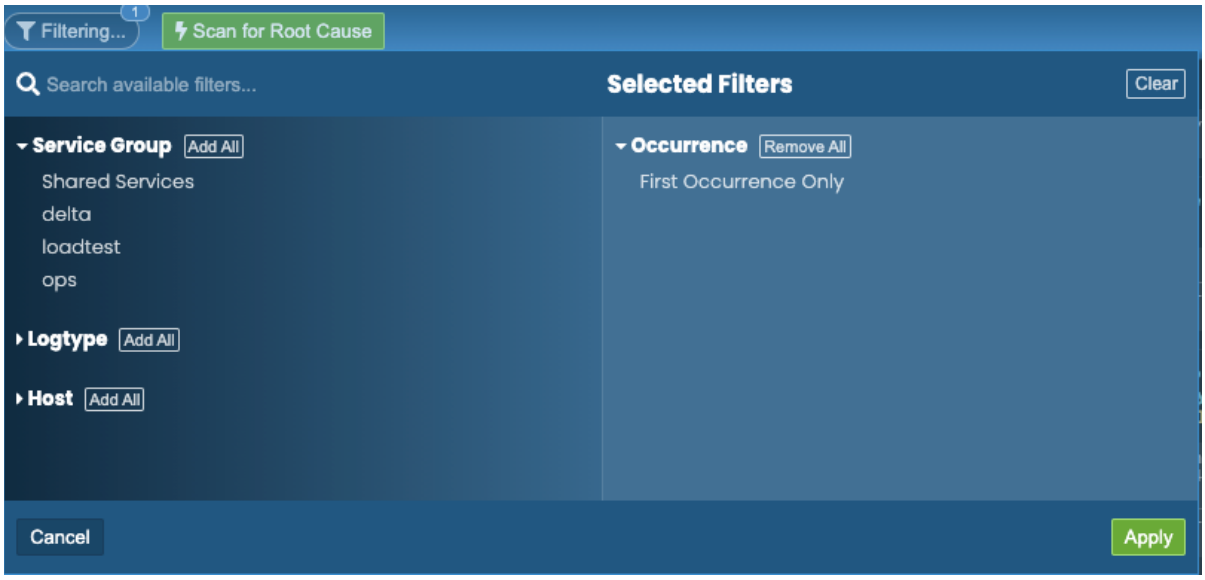
- **RCA Reports** (also called Incident RCAs) – each RCA report consists of a group of log events that our ML identified as being part of a problem.
- **Hallmark Events:** These events are shown in the summary view and often provide clues as to the nature of the problem. However, in some cases they may not contain anything obvious, and you will have to drill-down into the actual report to find relevant details. Hallmark events are also used as a "signature" for a particular incident type. There are typically two hallmark events:

    - The first event in the sequence, which is usually a rare event or anomaly and often relates the root cause.
    - A high severity event – either as determined by log severity, or other indicators such as certain words or phrases indicating a problem (e.g. exception, died, failed, could not restart, etc.)
- **Service Group:** is the collection of log types, pods, hosts etc that are all part of a "failure domain". In other words, logs from the micro-services and processes that could all interact with each other to contribute to an incident should be part of a Service Group. Zebrium's ML will only attempt to correlate anomalies and errors across logs that fall within a Service Group. For more complex applications, it is possible to have multiple Service Groups if there is more than one failure domain.

Navigation

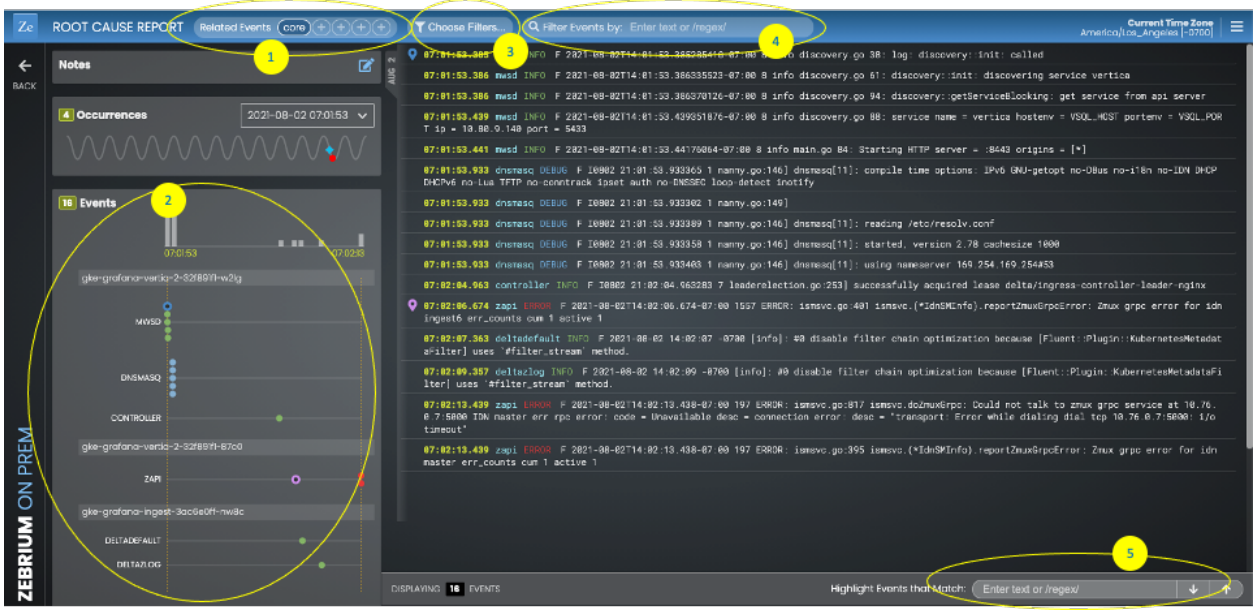
- The simplest way to navigate the Zebrium UI is to go to the main page showing a list of all known RCA reports

- The list is sorted by time (most recent first), and can be further filtered by log types, hosts, or service groups (using filter #1 above). If any log events in the RCA report match either of these attributes, they will be shown in the filtered view.

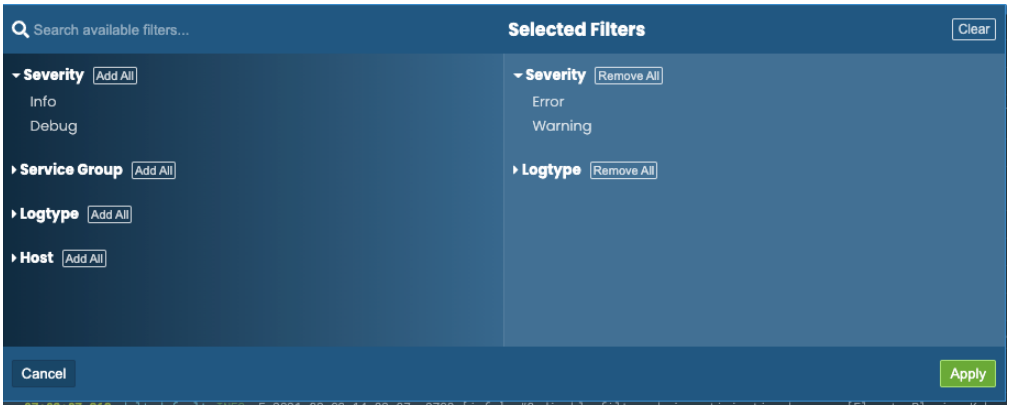


- The list shows the host, log types (which typically match container names), and the hallmark events that describe a given RCA report
- **Important:** By default, the list only shows the first occurrence of a new type of incident RCA. If the same problem has occurred previously, you can check for repeat occurrences by un-filtering the “first occurrence only” field
- If you don’t see a report a time of interest where you believe a problem occurred, it’s likely because it was suppressed by the existing sensitivity settings. You can force a report to be created via a deeper scan by clicking the “scan for root cause” button (#2 above), and specifying a time of interest.

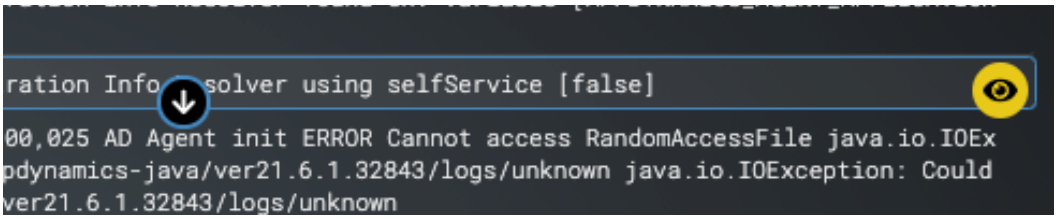
Clicking on any of the reports will take you to a “RCA details” view. In this view, you’ll be shown a more complete list of log events compiled by our ML to describe this particular problem.



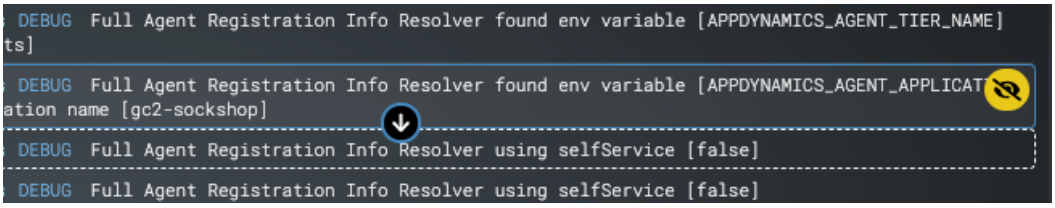
- You will also see a visualization of the log events (#2 above), partitioned by micro-service and host name, with the horizontal location of the dots based on chronological sequence.
- Typically, the “core” list in an RCA report will contain somewhere between 5-25 log events. However, you can then drill down by clicking on the “+” views (#1 in figure above) to see more anomalies, warnings and errors surrounding this core list of events. Each “+” will expand the list of events show (e.g. if the core list is 5 events, clicking “+” once may show you 205 events, and clicking the second one may show you 405 events.
- You can filter events by severity (e.g. error, warning etc), log type, host etc – refer to #3 above and the filter view below.



- You can further filter the log events at any zoom level, by matching against a text string or a (PCRE2 compliant) regex. *Note: regex filters should use the syntax "/regex/".*
- You can also highlight any desired alphanumeric strings within the visible log events using the "find" field at the bottom of the screen (#5 in the figure above).
- Alternatively, you can drill down on logs from a particular host or pod by clicking on the "peek" button. This is similar to looking at the log file for a single log generator.



- To exit the peek mode, click the "clear" button.



- Finally, you can add notes, a title and an issue tracking URL (e.g. a pointer to Jira) by clicking on the "edit" button.

GPT-3 Plain Language Summaries

- We also support a new feature that uses the GPT-3 NLP model to summarize root cause reports into plain language. These summaries will be automatically generated for some root cause reports. The summaries are designated "EXPERIMENTAL" to denote that this is a new feature that is constantly being

tuned/modified by Zebrium. In testing across many different stacks, we have found them to be useful approximately 40% of the time.