# Major enhancements from 3\_0 - 3\_45

## GCeasy

1. You can invoke GCeasy REST API by passing [remote http(s) URL location](https://blog.gceasy.io/2019/12/30/gceasy-api-remote-location/) of your GC log file. You don't need to send the GC log file contents in the API request payload. API will automatically download the file from the remote location and do the processing. This feature will be useful, if you are storing GC Logs in S3 buckets. You can just pass the http(s) URL of the file to GCeasy REST API.
2. Besides GC logs, now GCeasy can [parse 'jstat' output and generate analysis reports](https://blog.gceasy.io/2019/11/18/jstat-analysis/). 'jstat' is a handy tool that is available in JDK. In your application, if GC log isn’t enabled, you can upload jstat output to GCeasy, and study your application's GC behavior.
3. Heap usage graphs, GC Duration graphs, and few more graphs are sent as images in the REST API response. GCeasy API provides [powerful micrometrics](https://www.youtube.com/watch?v=hsqsD5c4nP0), which can predict & forecast performance characteristics of your application. Several Enterprises are embedding GCeasy micrometrics into their CI/CD reports. These graphs give them more meaningful visualizations.
4. Tool now parses modern GC algorithms such as Z GC, Shenandoah GC, Azul GC and gives more detailed reports.
5. Based on the JVM arguments, sometimes timestamps in GC logs are printed in relative format i.e. '0.209s'. However, engineers prefers to read graphs & metrics in absolute timestamp format i.e. 20:43:12. Using [this feature](https://blog.gceasy.io/2019/02/07/painting-graphs-in-absolute-timestamp-format/), you can make GCeasy to print graphs in absolute timestamp, even though timestamps are present in relative format in the GC log file.
6. GC log format varies based on the Java version, Java vendor, JVM arguments, GCV algorithm. If you are looking to normalize GC logs and generate a standardized GC log, you can use our [GC standardization API](https://blog.gceasy.io/2017/08/15/gc-log-standardization-api/).
7. In addition to selecting & uploading GC log file, you can also copy & paste GC logs in to the GCeasy web application i.e. equivalent to the “RAW” upload feature that you see in postman.
8. New problem detection patterns, new JVM argument recommendations have been added.

## FastThread

1. Along with uploading “top -H”, you can start uploading “top -H” output to FastThread. Tool will now effectively marry these two datasets to provide compelling visualizations. This visualization will facilitate to identify the exact line of code that is causing CPU & memory problems. More details on this approach [can be found here](https://blog.fastthread.io/2020/03/28/powerful-troubleshooting-marrying-top-thread-dumps/).
2. You can invoke FastThread REST API by passing [remote http(s) URL location](https://blog.fastthread.io/2019/12/30/fastthread-api-remote-location/) of your thread dump file. You don't need to send the thread dump file contents in the API request payload. API will automatically download the file from the remote location and do the processing. This feature will be useful, if you are storing thread dumps in S3 buckets. You can just pass the http(s) URL of the file to FastThread REST API.
3. Enhanced the tool support for more thread dump formats like:
   1. Spring boot generated thread dumps
   2. Jolokia thread dumps
   3. Enhanced support for IBM thread dumps
4. Certain tools generate thread dumps in UTF-16 character encoding format. FastThread is enhanced to parse these sort of UTF-16 thread dumps formats as well.
5. In addition to selecting & uploading GC log file, you can also copy & paste thread dumps in to FastThread web application i.e. application i.e. equivalent to the “RAW” upload feature that you see in postman.
6. Added more error detection patterns to detect and report slowdowns, CPU spikes, memory degradation. Also added more solution recommendations.

## HeapHero

1. You can invoke HeapHero REST API by passing [remote http(s) URL location](https://blog.heaphero.io/2019/12/30/heaphero-api-remote-location/) of your heap dump file. You don't need to send the heap dump file contents in the API request payload. API will automatically download the file from the remote location and do the processing. This feature will be useful, if you are storing heap dumps in S3 buckets. You can just pass the http(s) URL of the file to Heap Hero REST API.
2. In HeapHero enterprise edition, we have introduced a [new feature called Remote Location](https://blog.heaphero.io/2019/12/30/heaphero-remote-location/). Using this feature, you can specify the http(s) URL location of your heap dump in the request. Say suppose, you have stored your heap dump files in a remote location like AWS S3 bucket. In those cases, if you want to analyze your heap dumps then you will have to download files to your local machine and then upload heap dump to HeapHero tool. It can be a time-consuming task. To make it simple we have introduced a new option in heaphero home page. If heap dumps are present in remote locations, you can specify the http(s) URL in the remote location input box. HeapHero will download the heap dump file from this remote location, do the analysis and display heap report.
3. Suppose the SRE/DevOps team in your organization has uploaded the heap dump to HeapHero and shared the HeapHero report hyperlink with you. Now, if you want to download the copy of this heap dump to your local machine for doing more detailed analysis, it can be achieved by clicking on the ‘Download Heapdump’ button on the top right corner of the screen.
4. HeapHero report has a ‘System Properties’ section. In this section application’s system properties are printed. Sometimes system properties tend to contain sensitive information like JDBC URL, passwords,… For security reasons, if you decide not to publish them in clear text, we have exposed [this new feature](https://blog.tier1app.com/2016/10/30/enterprise-edition-admin-manual/) that can mask the data that is published in the report.
5. Some Heap dumps tend to contain thread dumps with in it. In those circumstances, HeapHero can invoke FastThread (thread dump analysis tool) to do the thread dump analysis and publish detailed thread dump analysis report. Details about this feature can be [found here](https://blog.tier1app.com/2016/10/30/enterprise-edition-admin-manual/).
6. We have built integration with Eclipse MAT. When you upload the heap dump file to HeapHero, it can internally invoke Eclipse MAT for parsing. Results of Eclipse MAT can be rendered on HeapHero.
7. HeapHero performance is enhanced to process large size heap dumps. Now it can process very large size heap dump files.

## General Enhancements

1. We have strengthened S3 Integrations. You can now store your dumps in AWS s3 buckets instead of local disk.
2. In Enterprise Edition APIs are invoked with one global api key. If you are planning to roll-out our service enterprise-wide, you might want to give unique api key for each organization in the enterprise. This approach will provide you with more granular control. If you decide to grant access or revoke an organization from accessing the service, then [this granular control will be of help](https://blog.tier1app.com/2016/10/30/enterprise-edition-admin-manual/).