Empowering the CI/CD process with

OPTIMIZED LOGGING

<by> **ARIEL ASSARAF**
CEO of Coralogix

Coralogix

7 LEVELS OF LOGGING MATURITY

Simple, file based logging

Access to logs for application provides insight into what the application is doing in real time.

Log centralization

Avoid jumping in and out of multiple servers when you have more than one running instance of your application.

Logs become Structured Events

Consistency in logs makes it easier to sift through what your logs are doing, as well as query on specific values.

Logging Graphs and Metrics

Visualising the logs in dashboards, graphs and counts means less time spent digging through actual log lines and provides a higher level view of the system.

Logging tools built into the applications and servers

Stop inconsistency between which apps are publishing logs and which aren't. Ensure that every component of the platform behaves consistently.

Alerts attached to logs

Alarms that are based on logs provide a powerful early warning system and shorten the time to recovery.

Machine learning analysis of logs

Fill in the gaps missed by your alerting rules, learn new and interesting patterns in your system. Categorise and optimise data for human consumption.

SUMMARY

Customers expect a great digital experience. This fact is pushing software development to the limits of efficiency. It has driven engineers to discover improved and sophisticated development processes that have offered a way to streamline the lifecycle from development to production.

Modern architectures need an optimized CI/CD pipeline to unlock faster development timescales and improved release cycles.
The result is speed to market, improved commercial models, and better customer experiences.

Your log data is the key

Many organizations have an invaluable repository of information, containing everything they need to optimize their workflow and speed up their delivery. Their log data. To take full advantage of this data requires a change in behaviour - the introduction of logging best practices.

Best practices, coupled with an intelligent approach to logging, based on machine learning (ML), provides the methodology to make a CI/CD pipeline optimal. This unlocks:

- ➜ **A fluid and seamless, CI/CD pipeline, removing backlogs and using benchmarks to keep delivery running smoothly;**
- ➜ **Focused alerts, removing the avalanche effect that provides little meaning. Engineers are enabled through knowledge; therefore,**
- ➜ **Faster and more accurate feedback between end users and Engineers.**

The result is a system that not only delivers efficiently into production but also takes full advantage of an intelligent data-driven approach to better software and services. This paper looks at the best practices and tools needed to achieve this goal in CI/CD pipeline optimization.

Why logging is important

In the world of modern software development, data is the perfect lubricant for optimized and effective delivery. But modern architectures need actionable data that provides deep and insightful details throughout the CI/ CD pipeline.

Best practices in logging and alerts lead to optimized software delivery. Behind these best practices are a series of goals needed for an organization to create amazing software in a competitive world.

ENSURE AN APPLICATION HONORS ITS SERVICE LEVEL AGREEMENT (SLA)

An SLA is intrinsically linked, via log data, to service level objectives (SLOs) and service-level indicators (SLIs). These log data give an organization sight of the ongoing quality levels of the service.

ENSURE AND PROVE AN APPLICATION COMPLIES WITH RULES AND REGULATIONS

Never before have there been so many rules and regulations on the use of data. Many data protection and privacy laws require that audits and checks are carried out regularly. Logs provide the necessary information on how data is being processed within a system. As such, logging compliments and augments regulatory compliance.

CREATE A 'SKETCH' OF THE OPERATING CHARACTERISTICS OF AN APPLICATION

Machine learning-enabled logging provides the deep analysis needed to sketch the operating characteristics of an application.

ASSESS AN APPLICATION'S HEALTH USING APPLICATION LOGS

Logging provides a snapshot of the health of an application. It does this by comparing current operating characteristics with a recent 'healthy' operating characteristic. This provides the intelligence needed to restore the health of a system.

DETECT FAILURES DUE TO EXTERNAL ATTACKS AND INTERNAL ERRORS

Malicious attacks against web endpoints are common. And, these attacks usually have a distinct fingerprint that logging detects. Using techniques such as pattern mining and outlier analysis the signals of cyber-attacks can be identified.

PROACTIVELY MAINTAIN AND EVOLVE AN APPLICATION

Continuous monitoring and logging of the operating characteristics of an application and its environment provides a pro-active picture of a system.

ASSESS THE EFFECTIVENESS OF CHANGES TO AN APPLICATION

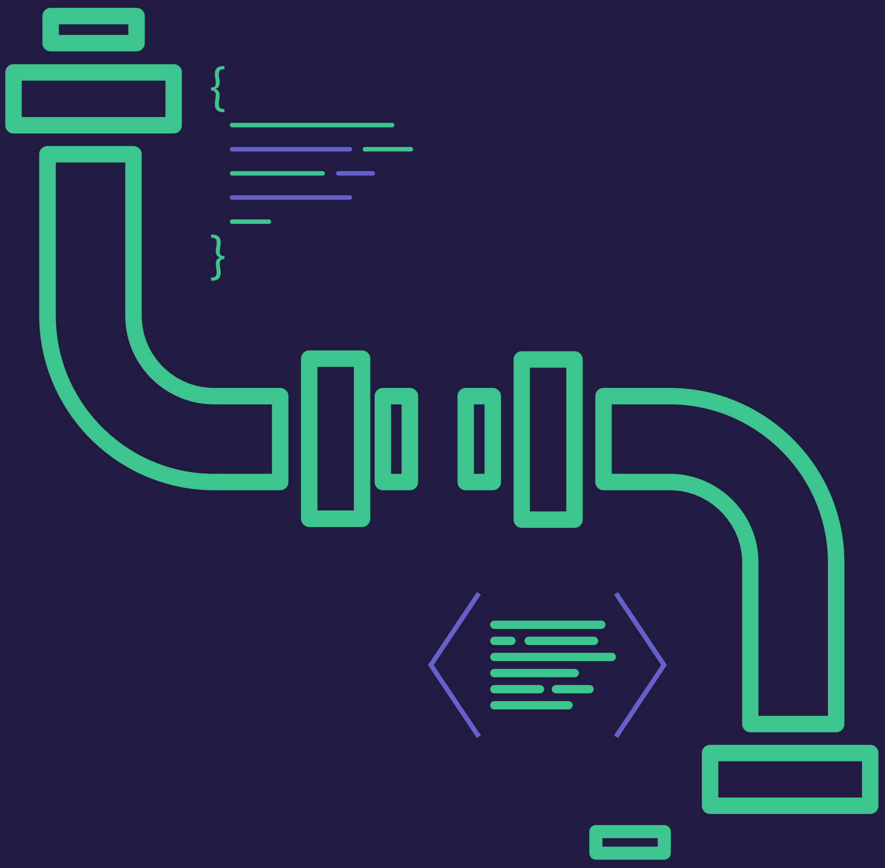
Logs can be used to measure the effectiveness of different versions.

DETECT SECOND-ORDER EFFECTS OF FEATURES OF AND CHANGES TO AN APPLICATION

Application logs provide a wealth of information. Logs that are used optimally, provide a rich seam of data that records the evolving nature of an application.

<SECTION ONE>

EMPOWERING YOUR CI/CD PIPELINE USING LOGGING 4



<01>

EMPOWERING YOUR CI/CD PIPELINE USING LOGGING

Engineers have to use key foundation stones to build successful outcomes. Monitoring and alerting provide one of these foundational practices in the delivery of services and software. Logging provides the basis for monitoring and alerting, and logging empowers decisions. Efficient, actionable logging does not come for free. One must follow best practices to achieve effective development that gives your business a competitive edge.

We are all familiar with the risks of software development and delivery. Delivery is a collaboration between multiple teams who often only receive critical data at the last minute. Actionable, informed decisions need to be made in moments. Optimized logs give engineering teams the data to make these decisions with the best available information to hand, resulting in fewer mistakes or opportunities missed.

Coralogix delivers actionable logging, by taking a new holistic approach to logging that empowers Continuous Integration/ Continuous Deployment (CI/CD) pipelines for modern architectures. By monitoring the entire pipeline, Coralogix provides the intelligence needed by engineers to ensure every stage of the pipeline is efficient and effective.

ASPECTS OF LOGGING:

“Adding life to log data”

Logging that helps you make key business decisions has to follow best practices, but what are these best practices? Each section below details a pivotal aspect of logging, ensuring your organization is set up to optimize the logging that you depend on.

Unstructured vs. JSON

JavaScript Object Notation (JSON) is a widely used data exchange language. Its ubiquity in software is unrivalled, and is supported by most modern programming languages. It is human- readable and can be easily generated by machines.

These features make JSON a better choice than unstructured formats in supporting other data formats (including legacy formats) as they can be transformed into JSON. This provides a good starting framework to build universal logging services.

> “JSON adds life to log data.
>
> **ARIEL ASSARAF**
>
> CEO
> CORALOGIX

Other ways that json surmounts unstructured data formats

JSON makes log files easier to visualize: JSON is now a standard format for logging. One of the reasons for JSON becoming a chosen format is that the universal data structure and other features of JSON support the creation of a structured database for logs. This lends itself to visualizing even very complex logging data that contains hundreds of thousands of events.

JSON IS EASY TO READ

Logs need to be visualized using dashboards and other interfaces. However, logs also need to be human-readable just in case an operator needs to dive in and check something.

JSON IS EASIER TO FILTER AND GROUP

JSON transforms logs from raw log text to database fields. Once transformed, the database can be searched, and filters used to drill down into the logs. These data are labeled, making it easy to combine a query or filter based on two fields. You can also set a query based on ranges of numeric parameters, for example. The result is a deep focus that builds a very accurate picture of an event.

```
{
        “api”: “authcode”,
        ”projectid: ““,
        ”message”: “login_success”,
        ”codetype: “secondfactor”,
        ”method”: “email”,
        ”refscope”: “authn”,
        ”state”: “98675637041634936789”,
        ”uid”:”6f4ff9f4ed250c1a000029”,
        ”datetime”: “2020-07-19 08:00:00.633475 UTC”,
        ”ip”: “xxx.xxx.xxx.xxx”,
        ”referrer”: “https://www.example.com”,
}
```

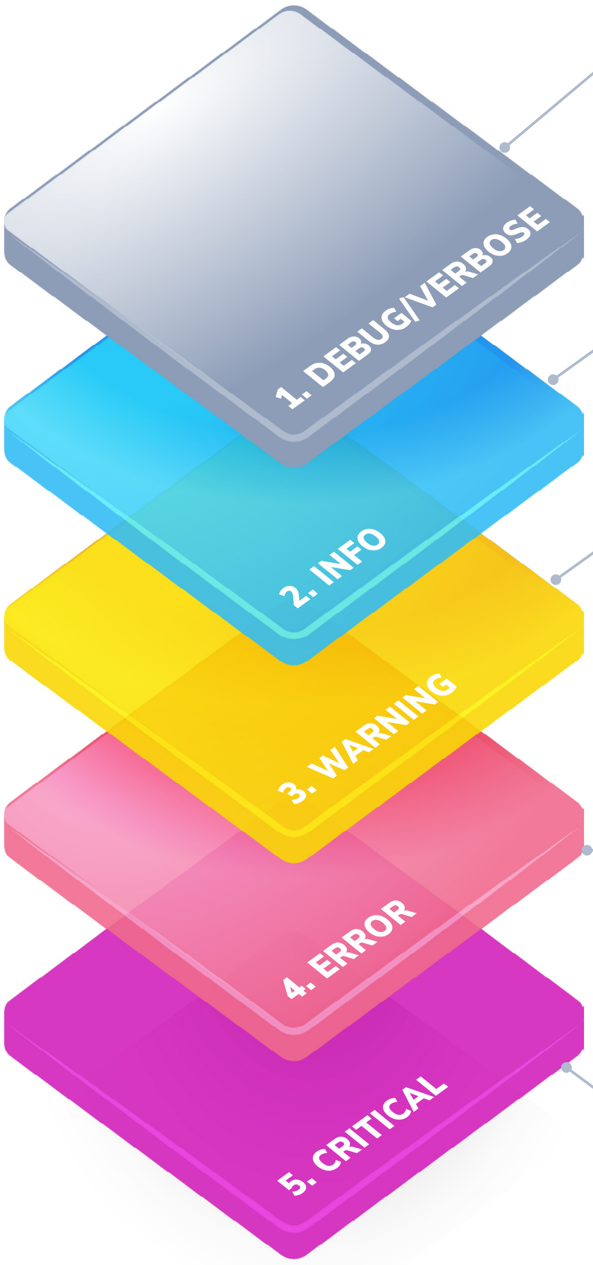
Example JSON log

Overall, JSON Is a powerful format that provides a definitive view for each user. JSON offers the flexibility and versatility needed for smart modern logging.

Log severity definition

Not all logs are created equal. This simple statement is the difference between log fatigue and missing a critical event. Log definition is a musthave for clarity, giving an instant understanding of how important a log is.

A classification system for logging is important as it provides a basis for effective action based on a clear view of severity level. The examples of log severity classes, shown below, are guidelines. Your company should create a classification system that best suits your requirements:



1. DEBUG/VERBOSE
Describes a background event. These logs are not typically needed except when as a method of context and analysis during a crisis. These logs are useful in tuning performance or debugging.

2. INFO
Logs that represent transaction data, e.g. a transaction is completed successfully.

3. WARNING
These logs represent unplanned events, for example, they may show that an illegal character for a username was received but is being ignored by the service. These are useful to locate possible flaws in a system over time and allow changes to improve system usability, etc.

4. ERROR
Represent failures in processes that may not materially affect the entire service. These logs are useful when aggregated and monitored to look for trends to help improve service issues.

5. CRITICAL
These logs trigger an immediate alert to take action. A fatal log implies a serious system failure that needs immediate attention. Typically, in a day, 0.01% of logs would be defined as high severity.

Optimizing the CI/CD Pipeline

Many CI/CD tools facilitate smooth, fast deployments but a build still needs to be optimized. Logs allow you to understand various elements of the build, such as timing, load issues, and scaling problems.

This data often turns up in the middle of a crisis. If you need to get a fix out fast, optimized logs give you the data to make informed decisions.

Benchmarking the Versions

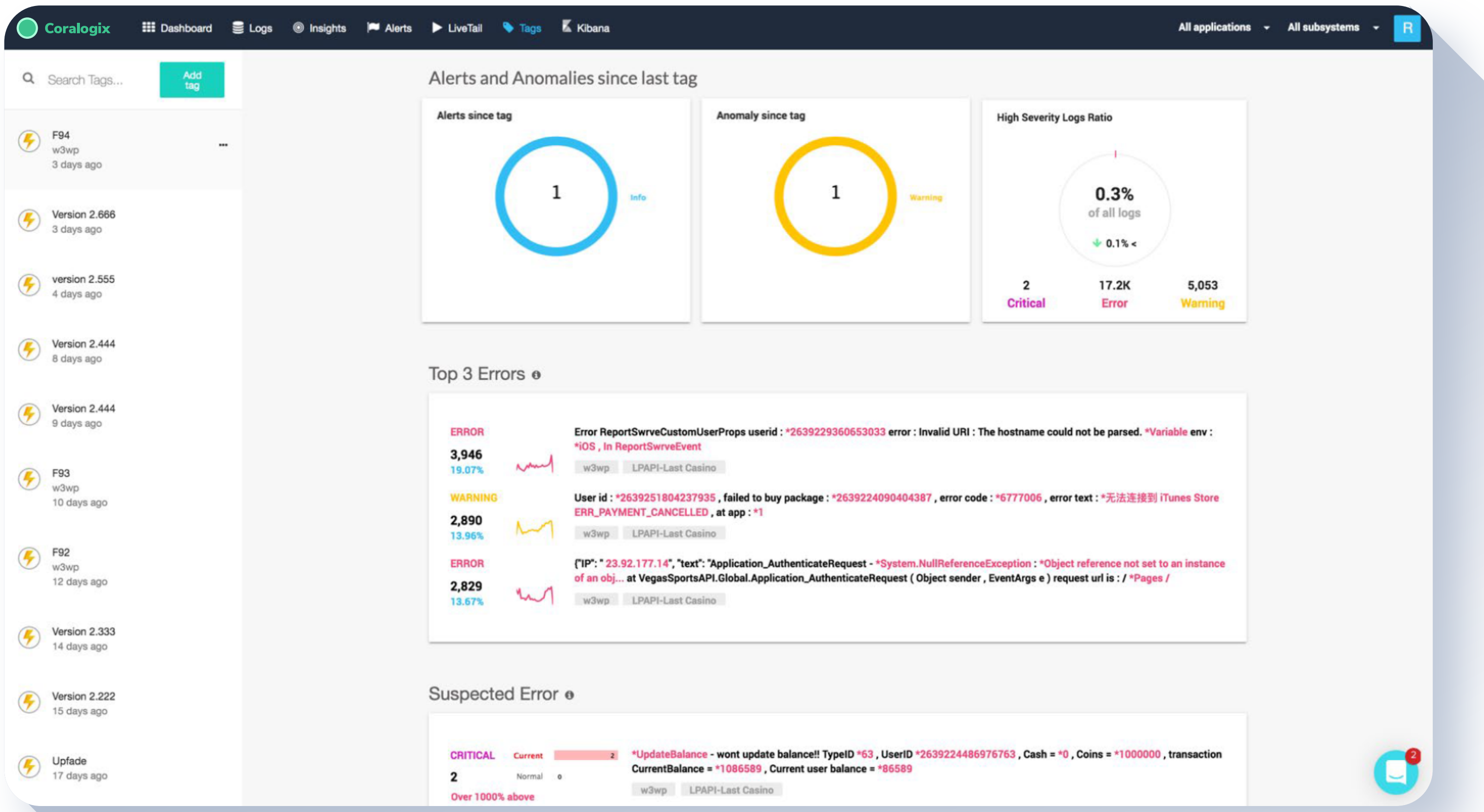
Coralogix integrates with any CI/CD tool allowing you to tag every version you upload to production. This tag is then represented in Coralogix every time you deploy. Machine learning (ML) for data analysis is used to benchmark this tag and maps it to the version.

ML lets you drill down to locate anomalies based on the benchmark. By leveraging the power of ML, this process allows you to detect brand new errors that have appeared in this new version, by comparing it with previous versions.

THIS PRACTICE PROVIDES

1. FASTER DELIVERY
2. CONTROL OVER A BUILD AND SCALING OF CI/CD PROCESS
3. BENCHMARKS OF ANY VERSION

Tags are presented whenever a log search is performed and data queried by tag. This facilitates easier search, better team collaboration, and ultimately a more efficient and effective CI/CD process.



F94
w3wp
3 days ago

Version 2.666
3 days ago

version 2.555
4 days ago

Version 2.444
8 days ago

Version 2.444
9 days ago

F93
w3wp
10 days ago

F92
w3wp
12 days ago

Version 2.333
14 days ago

Version 2.222
15 days ago

Upfade
17 days ago

Alerts and Anomalies since last tag

Alerts since tag: 1

Anomaly since tag: 1

High Severity Logs Ratio: 0.3% of all logs, 0.1% <

2	17.2K	5,053
Critical	Error	Warning

Top 3 Errors

```
ERROR    3,946  19.07%   Error ReportSwrveCustomUserProps userid : *2639229360653033 error : Invalid URI : The hostname could not be parsed. *Variable env :
                         *iOS , In ReportSwrveEvent
                         w3wp  LPAPI-Last Casino
WARNING  2,890  13.96%   User id : *2639251804237935 , failed to buy package : *2639224090404387 , error code : *6777006 , error text : *无法连接到 iTunes Store
                         ERR_PAYMENT_CANCELLED , at app : *1
                         w3wp  LPAPI-Last Casino
ERROR    2,829  13.67%   {"IP": " 23.92.177.14", "text": "Application_AuthenticateRequest - *System.NullReferenceException : *Object reference not set to an instance
                         of an obj... at VegasSportsAPI.Global.Application_AuthenticateRequest ( Object sender , EventArgs e ) request url is : / *Pages /
                         w3wp  LPAPI-Last Casino
```

Suspected Error

```
CRITICAL  Current 2   *UpdateBalance - wont update balance!! TypeID *63 , UserID *2639224486976763 , Cash = *0 , Coins = *1000000 , transaction
2         Normal  0   CurrentBalance = *1086589 , Current user balance = *86589
Over 1000% above      w3wp  LPAPI-Last Casino
```

Coralogix Version Benchmarks

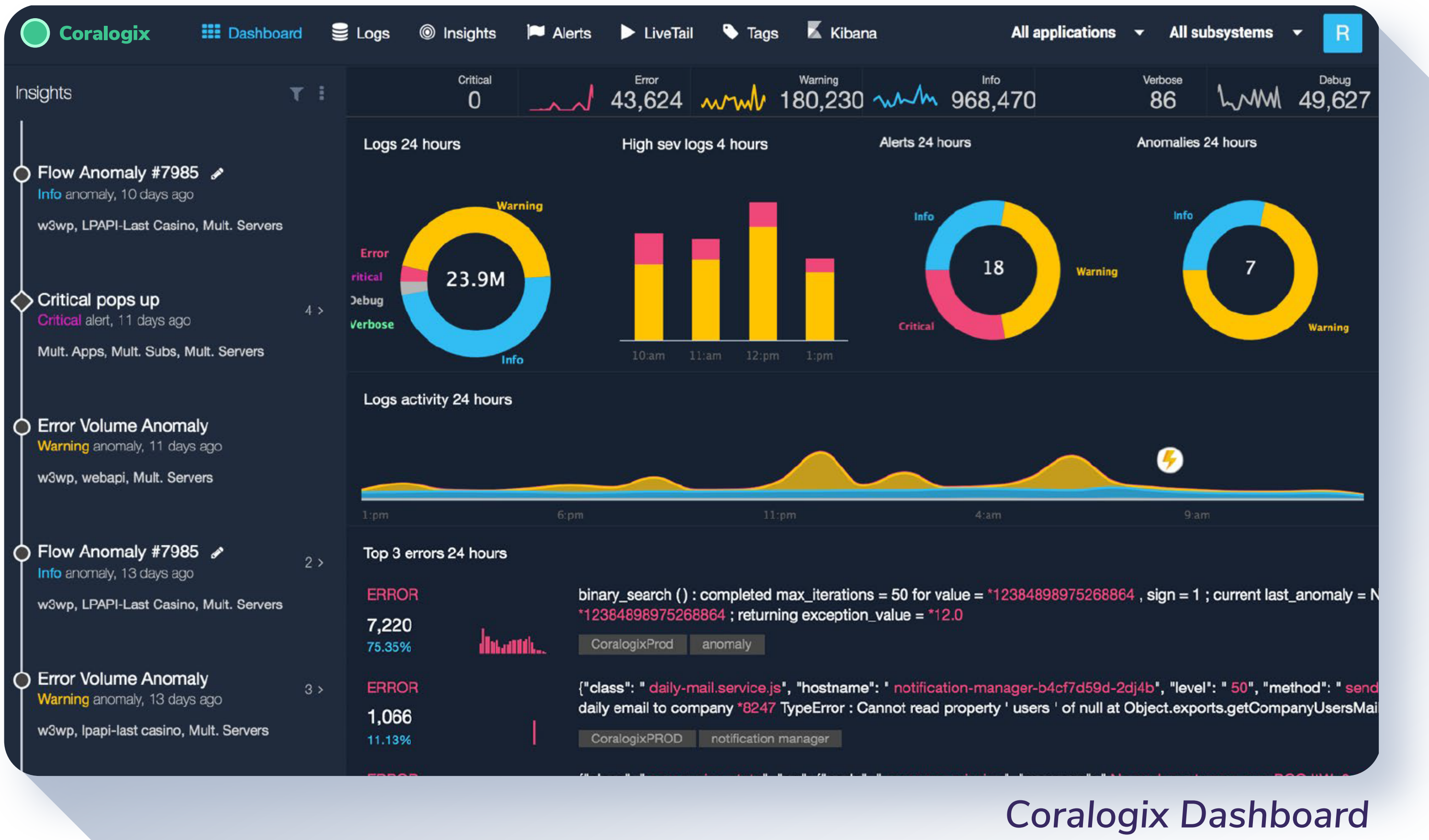
Smart Business Metric

Optimally using logs and associating metadata with logs, allows you to develop business logic metrics. These metrics offer business users specific actionable data. Examples of this include number of errors per purchase or number of active users and much more. Using this data, one can also discover correlations between events, such as the relationship between latency and drop off rate on the website.

LOGGREGATION

When you're searching for an error, you often need to filter through thousands of logs to find the relevant information. "Loggregation" is a process that analyzes logs automatically and clusters them into templates. This gives an at-a-glance view of anomalies and parameters that stand out. Users can visualize these simply by clicking on a parameter of interest.

Both operations and business data can be visualized using optimized logs. Coralogix offers a cost-effective “Logs to Metrics” component. This provides visualization of core business operations. Users can create a query, define aggregation, define labeling, and store for a year. This component lets a business visualize long term trends, providing deep insight into how business operations are working across time. This can help improve software development and optimize production and delivery.



Flow Anomaly #7985
Info anomaly, 10 days ago
w3wp, LPAPI-Last Casino, Mult. Servers

Critical pops up 4 >
Critical alert, 11 days ago
Mult. Apps, Mult. Subs, Mult. Servers

Error Volume Anomaly
Warning anomaly, 11 days ago
w3wp, webapi, Mult. Servers

Flow Anomaly #7985 2 >
Info anomaly, 13 days ago
w3wp, LPAPI-Last Casino, Mult. Servers

Error Volume Anomaly 3 >
Warning anomaly, 13 days ago
w3wp, lpapi-last casino, Mult. Servers

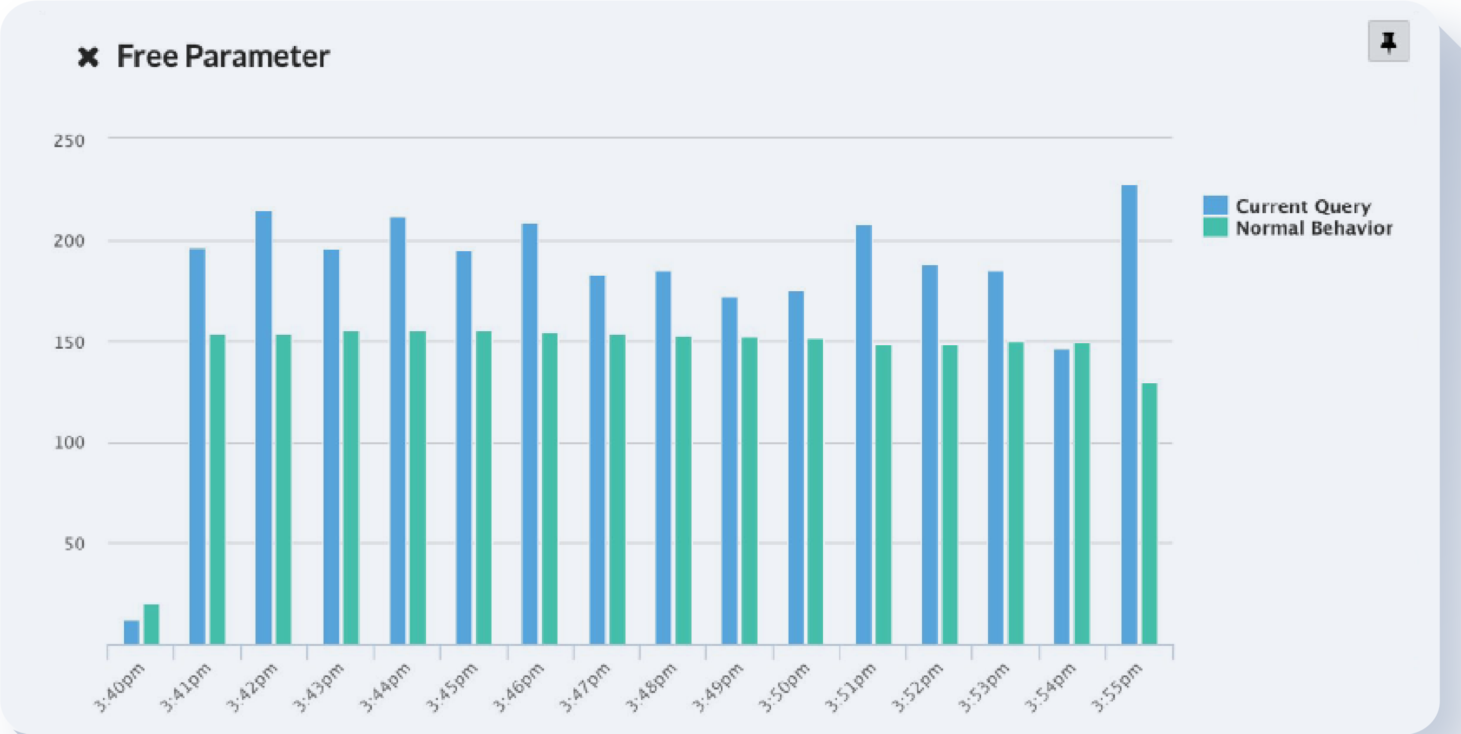
```
ERROR
7,220
75.35%
binary_search () : completed max_iterations = 50 for value = *12384898975268864 , sign = 1 ; current last_anomaly = N
*12384898975268864 ; returning exception_value = *12.0
CoralogixProd  anomaly

ERROR
1,066
11.13%
{"class": " daily-mail.service.js", "hostname": " notification-manager-b4cf7d59d-2dj4b", "level": " 50", "method": " send
daily email to company *8247 TypeError : Cannot read property ' users ' of null at Object.exports.getCompanyUsersMai
CoralogixPROD  notification manager
```

Coralogix Dashboard

FIVE LOGGING BEST PRACTICES

Severity definition provides a baseline to create a system based on actionable logs, but logging has a wider remit. One that requires best practices to optimize logging on an organizational basis. The five best practices shown below, facilitate the optimization of a CI/CD pipeline, ensuring that engineers and other stakeholders can use logging data in a clear and effective way.



1. Log communication between components

Services are ecosystems of interconnected components. All events in the system, including those that happen across the barrier between components, should be logged. The logs should detail what happens at each component, as well as the communication between components. This gives us a view of the lifecycle of an event.

2. Log communications with external APIS

The API economy has facilitated the extended service ecosystem, but API events often happen outside an organization. Optimized logging records what is communicated across the API layer. For example, if a service uses SendGrid to push out communication emails to users, you need to know if critical alert emails are being sent. If not, this would need to be addressed. Comprehensive logging of external services provides the visibility necessary to know the moment there is a service interruption.

3. Add valuable metadata to your log

In modern service ecosystems, many stakeholders need access to logs This includes Business Intelligence teams, application engineers, Support engineers, etc. Logs should include rich metadata, e.g. location, service name, version, environment name, and so on.

4. Log accessibility

You may not be the only one reading the logs. As companies scale, often access is needed by other stakeholders to change code, etc.

5. Combine textual and metric fields

Improve snapshot understanding of logs by having explanatory text (e.g., “failed to do xxx”) combined with metadata fields to provide more actionable insights. This offers a way to look at logs to see an at-a-glance view of the issue before drilling into performance data.

An occurrence within a chosen visualized parameter can then be expanded to see detail. Alerts can be associated with any chosen parameter.

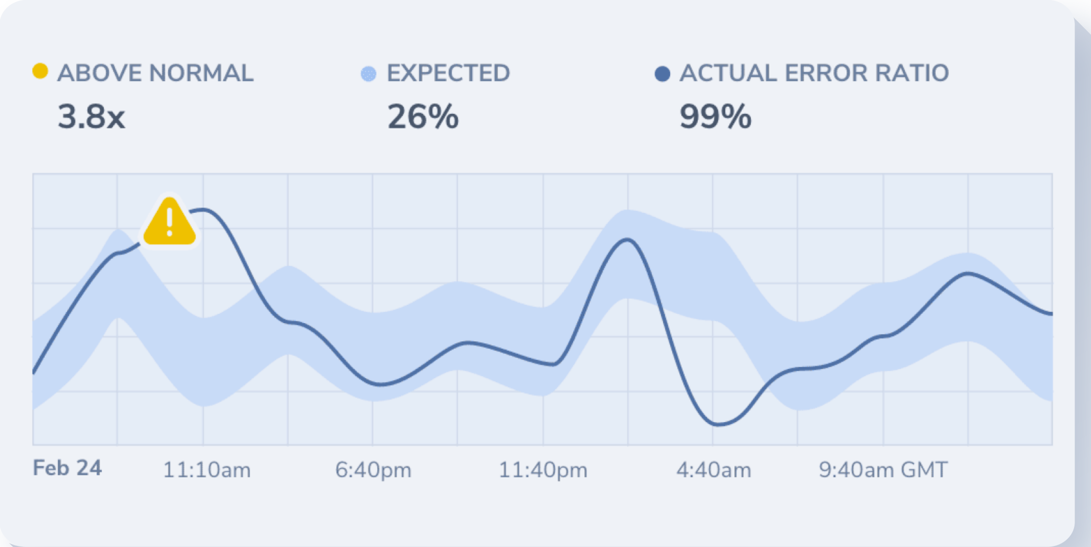
Loggregation is where machine learning collaborates with human insight and intelligence. Loggregation enables Coralogix to spot two types of anomalies:

FLOW ANOMALY

Coralogix discovers sequences of logs; logs that arrive together, for example. Coralogix then notifies on missing logs, understanding sequences across microservices, to spot issues.

ERROR VOLUME ANOMALY

Coralogix detects spikes, error spikes, bad API response spikes, and so on. These are often missed by users as the issue will quickly rectify back to normal. Coralogix will point out suspected errors during this spike. This can be tied to the benchmarks created during releases, and then used to identify errors.



Coralogix Error Anomalies

<SECTION TWO>

ACTIONABLE ALERTS

<02>

ACTIONABLE ALERTS

Logs are a leading indicator of issues and can be used for more than just a post-mortem analysis. Whilst metrics for infrastructure only present an outcome of problematic code/performance, logs are the first encounter with code in use. As such, logs offer an easy way to spot things before they are experienced at the user end. This is key to CI/CD enhancement and service/software optimization.

“Why are alerts needed?

Logs need to be accurate and have context (unlike metrics). Being able to make alerts specific and contextual will make that alert actionable. Classification of alerts will ensure that your organization gets the most from alerts without overwhelming engineering teams.

Defining Alerts

For alerts to be actionable they need definition. Below are three classes of alert:

IMMEDIATE ALERT

These alerts point to a critical event and are generated from critical and fatal logs. They require immediate attention to fix a serious issue.

“MORE THAN” ALERT

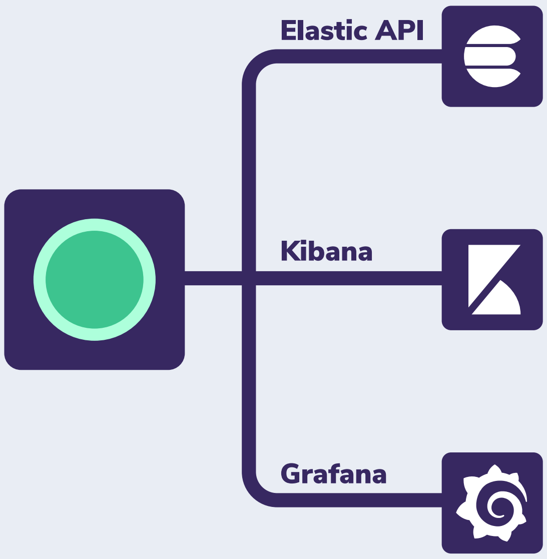
Sent out if something happens more than a predefined number of times.” “For example, an alert may trigger if more than 10 users are failing to pay. If these types of alerts are properly defined and sent to the right channel, they can be acted upon and be highly effective.”

"LESS THAN" ALERT

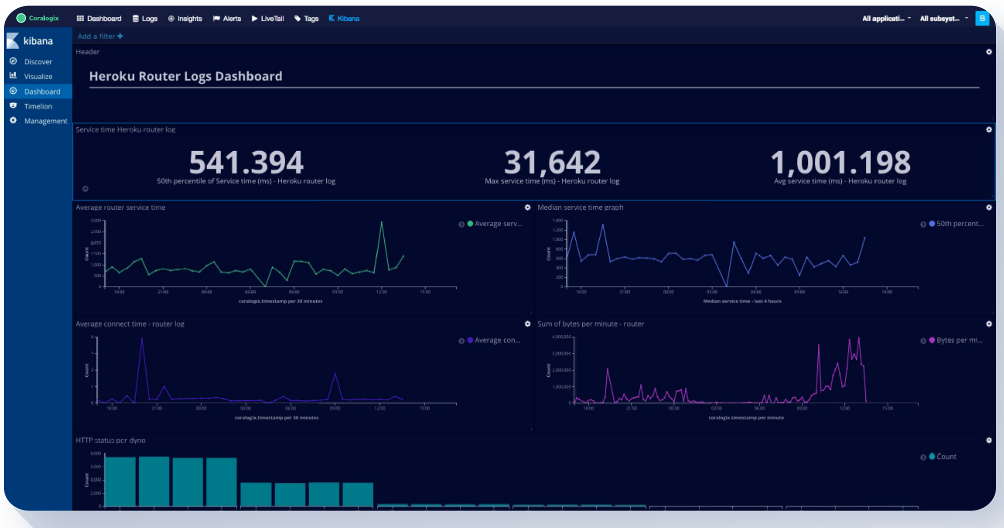
These are sent when something is NOT happening, e.g. a database clean-up did not run. Rather than waiting until the critical issue surfaces, your organization can be informed of the root cause as it happens and fix it before it cascades.

If alerts are defined and channeled correctly, have context, and can be interpreted easily, they will be actionable, add context, and therefore offer greater value."

A LOGGING AND ALERT ECOSYSTEM



Coralogix uses tools that are familiar to the industry; this includes Elastic API, Kibana, and Grafana. Logs can be used to generate metrics, then enriched using the Coralogix smart, machine learning, and granular classification system.


Coralogix
Dashboard
Logs
Insights
Alerts
LiveTail
Tags
Kibana
All applicati...
All subsyst...
kibana
Discover
Visualize
Dashboard
Timelion
Management
Add a filter +
Header
Heroku Router Logs Dashboard
Service time Heroku router log
541.394
50th percentile of Service time (ms) - Heroku router log
31,642
Max service time (ms) - Heroku router log
1,001.198
Avg service time (ms) - Heroku router log
Average router service time
Average serv...
Median service time graph
50th percent...
Average connect time - router log
Average con...
Sum of bytes per minute - router
Bytes per mi...
HTTP status per dyno
Count

Occurrence within visualized parameter expanded

Dynamic Alerts and Ratio Alerts

Coralogix goes further and adds granularity to alerts. This provides greater control over the creation of alerts making them even more actionable. Dynamic and Ratio alerts offer a deep level of management and control to alerts, making alerts smarter:

DYNAMIC ALERT

Rather than a fixed point that will trigger an alert, a dynamic alert can be resolved. This will create different thresholds based on the time and date.

RATIO ALERT

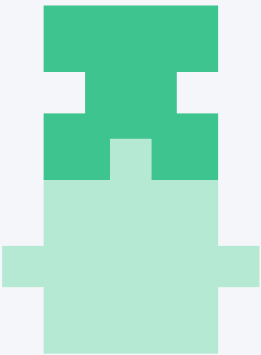
Create an alert based on a ratio between queries. This helps to ensure that alerts are only fired for consistent issues, and not minor fluctuations in site traffic. When alerts are generated, they carry more weight and meaning.

FOR EXAMPLE: *A 1% ratio alert for **"user failed to purchase / user purchased successfully".** If ratio goes over 1% then an alert is sent.*

Alert Structure

Classification of alerts is one criterion; another is the alert structure. This ensures that alerts go to the right people. For example, a dedicated area "can be set up in Slack for logging and metrics. Using channels, the alerts can be directed to specific teams. This technique is key when avoiding alert fatigue.

HOW IS CORALOGIX DIFFERENT TO GRAFANA



Grafana is mostly built for metrics data and does not allow data slicing. There are no version benchmarks, anomaly detection, ratio alerts, dynamic threshold alerts and much more. However, Coralogix is interoperable with Grafana. The two are symbiotic. You can generate metrics using Coralogix and present these in Grafana to enrich and empower your metrics collation.

To Push or Not to Push an Alert

The decision to push or not, an alert, is an important aspect of creating an effective 'alert culture'. Ask yourself these questions:

TEXT FOR FLOW

How would you feel if you received a specific type of alert at 2 am? What would you do? If you would be angered by getting this alert at that time, don't push it.

If, instead, your reaction is to say, "I must look at this tomorrow", define the alert on your dashboard, but **DO NOT** *push it.*

If the reaction is to stop what you are doing and respond – push the alert.

This simple logic can go a long way to making alerts actionable and workable. This method also helps to reduce alert fatigue to keep the workforce happier."

Empowering Alerts Using Visualization Tools

When you use logs in the right way, i.e., configured using optimized parameters such as metadata, contextual data, and defined correctly, logs can be used by more than just the engineering department. Great visualization takes logs to next level usability.

Visualization of logs empowers the use of logs as it creates an at-a-glance view of areas of interest. Great visualization of logs needs to be:

- **Contextual**
- **Easily read and understood by specific key stakeholders**
- **Presented using great visuals and graphs that engage**
- **Simple and easy to understand**

The result is an intelligent logging and alert ecosystem that generates actionable information and visuals. These enriched logs not only optimize the CI/CD process but engage stakeholders across the organization.

<SECTION THREE>

HOW CORALOGIX OPTIMIZES LOGS AND CREATES ACTIONABLE ALERTS

<03>

HOW CORALOGIX OPTIMIZES LOGS AND CREATES ACTIONABLE ALERTS

Coralogix brings the big picture of logging into sharp view, machine learning (ML) being used to find intrinsic value in logs.

Modern logging covers a vast array of devices, apps, and servers. The result can be millions, even billions, of log events each day. Finding meaning can be almost impossible. It can also be overwhelming for development teams, affecting morale and causing log fatigue.

Machine learning cuts through the mass of log data, creating cohesive, correlated categories. The logs can be grouped by user actions, log origin, system trends, time periods, or any number of other shared characteristics. New logs are then automatically deposited into existing groups that they correspond to.

Used in combination with log and alert best practices, then augmented with solutions such as Codefresh, Grafana, Heroku, and ELK, Coralogix provides an organization with actionable insights. These insights offer a company a powerful way to improve business processes, maintain a great user experience, and ultimately retain a competitive edge.

USE CASE

OPTIMIZING TO REDUCE ALERT CASCADES AND IMPROVE ACCURACY OF ALERTS

The Problem

This real-life use case shows how Coralogix deals with a cascade of alertsthat can overwhelm an organization. The organization in question had set up comprehensive logging across several levels: platform, services, and applications, with alerts enabled. The company saw a sudden 20-times spike in traffic within a 60 second period. The result was a cascade of alerts that left their DevOps team overwhelmed.

The team wanted to know if Coralogix could have prevented this problem, yet still ensure that real crises were dealt with.

Notably, the organization generates approximately 1 billion logs per day.

How Coralogix Optimizes Against Sudden Anomaly Spikes

Coralogix dynamic or ratio-based alerts solve sudden spike alert overload by using several mechanisms:

- ➜ **Whenever the traffic starts to spike, an organization gets a 'volume anomaly alert' allowing an organization to switch off the alerts while dealing with the situation.**
- ➜ **Log rules can define a "block rule" to block traffic you don't want to pay for. You can block on severity, application and much more.**
- ➜ **Dynamic and Ratio alerts offer a more granular and smart way to deal with sudden anomalies. Based on ratios rather than single events, a sudden surge in traffic does not initiate an alert; only a specified ratio would generate an alert.**
- ➜ **An alternative is to use 'Flow Anomalies' that look at sequences of logs Alerts are then only sent out if an expected sequence is broken.**
- ➜ **The level of granularity enabled by Coralogix smart logging facilitates a higher degree of control and introspection.**

> *"Honestly, the #1 feature is their amazing customer support! They reply online in less than a minute, so you can solve your production issues in real time"*

DAVID VIRTSER

HEAD OF INFRASTRUCTURE

monday

CONCLUSION

Logging acts like the eyes of an organization, but the insight offered by logging is only effective if it is optimized.

This is achievable by following logging best practices to build more actionable logs. Putting these practices into effect requires the right tools for the job. These tools need to facilitate the deep granularity necessary to query logs, by teasing out the key data needed to empower alerts and build visualizations that speak to stakeholders across the business.

The ecosystem behind the tools is made smarter through machine learning (ML). ML can interpret and analyze the vast amount of data generated across the array of devices in modern technology architectures. These analyses are the backbone of smart CI/CD pipelines that drive smoother SDLC and create better software.