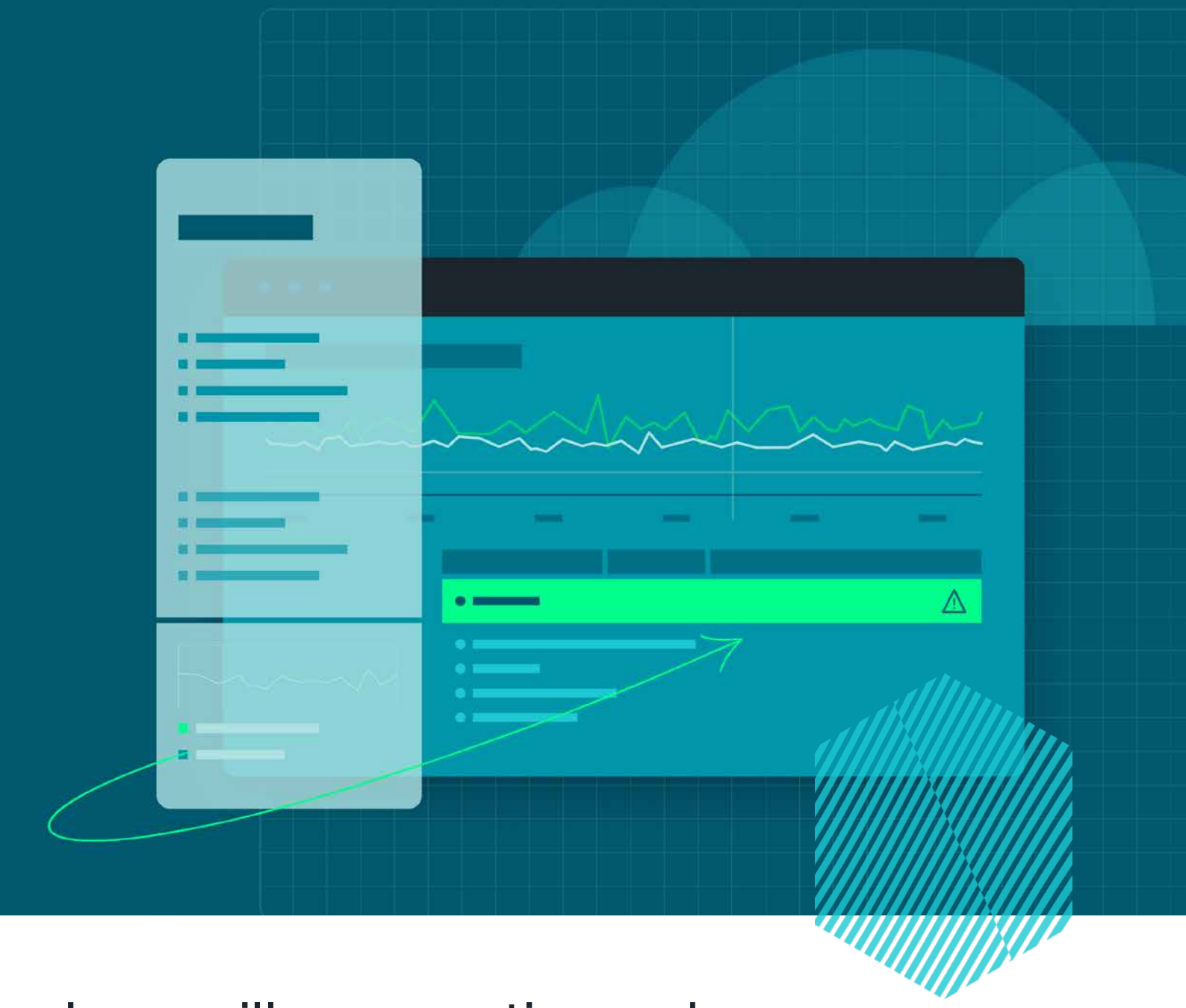
new relic



Les meilleures pratiques de gestion des logs

Des logs efficaces pour l'observabilité full-stack

Livre blanc / Août 2022

Table des matières

Introduction

La gestion des logs a évolué. Pour les organisations, l'examen minutieux des dumps de données brutes des logs d'applications et d'infrastructure dès qu'il y a une panne quelque part appartient désormais au passé. Aujourd'hui, la gestion des logs (ou logging) joue un rôle essentiel dans les opérations, l'intelligence commerciale et le marketing d'une organisation. Les logs sont le moteur de l'observabilité. S'ils sont bien structurés, ils sont le turbo qui permet aux organisations de rapidement et facilement comprendre comment fonctionne tout leur système et même d'empêcher les problèmes de se produire.

L'utilisation de logs pour l'observabilité exige plus que le simple déversement d'énormes quantités de logs piètrement formatés dans une base de données ou dans un fichier. Comment les organisations peuvent-elles changer intelligemment leurs pratiques de logging afin que les logs détaillés améliorent leur capacité à corréler les incidents sur l'infrastructure et toutes les applications, en temps réel, sans avoir à basculer entre différents outils ? Comment peuvent-elles mieux obtenir une observabilité de bout en bout ? Comment peuvent-elles se rapprocher encore plus de l'observabilité full-stack afin qu'elle soit utile à toute l'entreprise ?

Pour améliorer l'observabilité full-stack, la modification des pratiques de gestion des logs est simple. Dans ce livre blanc, nous abordons quelques-unes des bonnes pratiques pour les organisations modernes.

Gestion classique des logs

La gestion classique des logs se déroule dans un silo de données qui est stocké séparément des autres systèmes. Auparavant, l'observabilité s'appuyait sur le monitoring des performances des applications (APM) et le monitoring de l'infrastructure. Mais si le monitoring est important, il ne révèle pas tout ce qui passe dans les différents logs d'applications et périphériques de l'infrastructure. En effet, de nombreux outils de monitoring et de gestion des logs autonomes et compartimentés en silos se focalisent essentiellement sur les applications en ne tenant compte que d'une partie du stack et ne peuvent pas fournir d'informations complètes sur ce qui se passe et pourquoi.

Il est essentiel que les équipes aient les informations dont elles ont besoin pour accélérer les délais de commercialisation, obtenir des renseignements complets sur le comportement des clients, et réduire le temps de réponse aux incidents.

De nombreuses organisations qui souhaitent une observabilité full-stack doivent soit choisir de ne pas disposer des détails granulaires de leurs logs et se démener pour déterminer la cause profonde des problèmes, soit utiliser différents outils autonomes et essayer de relier les détails provenant des logs aux erreurs et aux traces. Lorsque les logs détaillés sont maintenus dans des silos distincts, il n'est pas possible pour les équipes d'avoir une vue complète sur tout. En conséquence, les coûts augmentent, le temps de commercialisation des produits est plus long, la visibilité sur l'expérience client s'en trouve réduite et le temps moyen de résolution des problèmes (MTTR) s'allonge.

Observabilité full-stack

La capacité à voir dans le stack technologique tout ce qui pourrait affecter l'expérience du client est appelée « observabilité full-stack » ou « observabilité de bout en bout ». Elle est basée sur une vue complète de toutes les données de télémétrie (métriques, événements, logs et traces).

L'observabilité full-stack fournit une visibilité complète sur les performances des applications et systèmes complexes (à partir d'une solution intégrée unique, de préférence) pour assurer le dépannage des incidents, la réduction du temps moyen de résolution (MTTR) et l'analyse de l'expérience client.

Avec l'observabilité full-stack, les ingénieurs et les développeurs ne sont plus obligés d'échantillonner les données, de compromettre la visibilité qu'ils ont sur le stack technologique, ni de perdre de temps à rassembler les données en silo. Au lieu de cela, ils peuvent se concentrer sur ce qui les intéresse : la programmation créative de haut niveau qui a un impact sur l'activité de l'entreprise.

La gestion des logs pour l'observabilité full-stack

La génération de logs pour tout le stack peut sembler être une tâche colossale. Les développeurs et les ingénieurs peuvent se poser des questions sur ce qui doit se trouver dans les logs, la somme de détails à inclure, et le coût entraîné par une quantité trop importante de données. De nombreuses entreprises paient le prix fort pour centraliser la gestion de leurs logs sur une plateforme différente et doivent finalement limiter les données de log envoyées en fonction des performances et du coût, ce qui limite aussi la visibilité et l'utilité pour l'entreprise. Sachant cela, nous avons examiné certaines bonnes pratiques de gestion des logs pour l'observabilité full-stack.

Savoir ce qu'il faut intégrer dans les logs

Les logs sont générés en écrivant du texte vers une sortie ou un fichier standard. La décision la plus importante consiste à choisir ce qui sera inclus dans les logs. Ceux-ci doivent contenir toutes les métadonnées nécessaires pour aider à identifier les événements et les causes profondes recherchés. Il peut s'agir d'éléments tels que des messages d'erreur ou des traces de stack et les valeurs, métriques ou événements connexes.
Tout ce qui doit être consigné dans les logs doit avoir un but. Qu'il s'agisse des données d'utilisation, des événements d'utilisateur ou des erreurs et exceptions d'application, tout doit présenter un intérêt pour l'équipe. Les informations sur les données de logs devraient :

- Être immédiatement utiles
- Fournir les détails nécessaires pour comprendre les causes sous-jacentes et prendre des décisions

Anticiper les scénarios courants

Les logs ne servent pas uniquement à répondre aux incidents. Ils peuvent aider d'autres aspects de l'activité, comme le profilage de la performance ou la collecte de statistiques.

Si l'on gère les logs en gardant à l'esprit quelques scénarios courants, on peut s'assurer de la valeur directe qu'apportent les logs à l'organisation. Par exemple, les logs sur les interactions des utilisateurs peuvent fournir des informations cruciales sur l'expérience des clients. Les logs système peuvent monitorer les problèmes ou les pannes matérielles. Les logs détaillés sur l'application peuvent aider à mieux comprendre les performances et les problèmes potentiels tels que les fuites de mémoire. Tout cela peut s'avérer très important lors des prises de décision.

Inclure des messages utiles dans les logs

Les messages des logs sont aussi importants que les informations et le contexte qu'ils fournissent. En ajoutant suffisamment de détails et en les rendant compréhensibles, les équipes peuvent utiliser les logs efficacement. Une infrastructure tierce tend déjà à capturer les détails granulaires nécessaires, mais pour les applications programmées en interne, les équipes doivent toujours obtenir les détails de log qui leur permettront de diagnostiquer un événement ou une erreur et d'en déterminer les raisons pour pouvoir prendre les mesures nécessaires qui auront un impact sur l'activité de l'entreprise.

Pour les erreurs d'application, le message doit communiquer ce qui se passe sur la ligne de code. Par exemple, un message d'erreur qui dit `Échec de la transaction` n'est pas aussi utile qu'un message d'erreur avec une description de type : `Échec de la transaction : impossible de créer l'utilisateur ${path/to/file:line-number}`. Et si le log inclut des données sur la transaction, cela aide le développeur à voir les raisons de l'échec.

En général, les codes d'erreur ou les codes d'état dans les programmes peuvent également indiquer le type de problèmes de l'application. Toutefois, si au lieu de simplement sortir le texte ou le numéro du code d'erreur on ajoute une courte description dans le log, cela permettra peut-être à un autre développeur ou ingénieur de ne pas perdre de temps à faire des recherches lors du dépannage.

Les logs doivent fournir des informations critiques à l'organisation. Les développeurs et les ingénieurs doivent éviter les messages cryptiques ou non descriptifs que seuls certains membres de l'équipe peuvent comprendre.

S'assurer que les logs sont simples et concis

Bien qu'il faille incorporer suffisamment d'informations dans le message de log, il est également important de ne pas en mettre trop. En effet, trop de données inutiles dans le message peuvent faire gonfler la taille du stockage et les coûts, mais aussi ralentir les logs de recherche et distraire du problème principal, ce qui complique son débogage.

Les équipes doivent s'assurer que les logs sont concis afin de capturer les informations les plus importantes. Les logs doivent contenir la raison de l'erreur tout en évitant tout élément inutile.

Ils doivent fournir des informations sur la cause profonde d'une erreur sans toutefois inclure le moindre détail sur l'environnement. Par exemple, si une application ne réussit pas à se connecter et à récupérer les données d'une API interne, il peut être bon de consigner tout message d'erreur provenant de l'API ou des informations sur l'état du réseau de cet environnement. Mais il n'est probablement pas nécessaire d'inclure la quantité de mémoire utilisée par l'application, ni le nombre d'applications en cours d'exécution.

Ne pas oublier l'horodatage

L'horodatage est très important pour les logs. C'est peut-être une évidence, mais si les développeurs et les ingénieurs ont l'habitude d'enregistrer les logs sur une base de données qui inclut la date et l'heure, il est très possible qu'ils oublient d'ajouter l'horodatage dans les messages de log. Ils doivent sélectionner le niveau granulaire le plus logique et le mettre dans les logs. Les tâches très fréquentes peuvent nécessiter de faire le suivi de l'heure à la milliseconde près, alors que pour d'autres tâches plus rares un suivi à la minute (voire au jour) près est préférable. Ce qui est important n'est pas simplement la granularité, mais l'application d'un standard cohérent dans toute l'organisation.

Autre point peut-être évident et important : il est essentiel de s'assurer que tous les systèmes sont synchronisés sur la même heure. Cela permet à la plateforme d'observabilité d'utiliser l'horodatage pour corréler les événements avec d'autres données télémétriques.

Utiliser un format analysable

Une plateforme d'observabilité ne peut pas extraire les données inutiles des logs. Les équipes doivent utiliser un format de log que les développeurs et les ingénieurs peuvent analyser et conserver une structure de logs homogènes afin de faciliter la collecte et l'agrégation. Par exemple, New Relic Log Management simplifie la façon dont les règles personnalisées d'analyse des logs sont définies,[1] mais la formule magique des règles d'analyse ne fonctionne pas si les données de log sont inintelligibles.

Un bon exemple de format de log non analysé est un log d'accès NGINX par défaut contenant du texte non structuré. Avec ce type de format, il est possible de faire des recherches, mais quasiment rien d'autre. Avec un format non analysé, les équipes doivent effectuer des recherches sur le texte intégral pour répondre à la plupart des questions. Voici un exemple d'une ligne type :

```
127.180.71.3 - - [10/May/2022:08:05:32 +0000]
"GET /downloads/product_1 HTTP/1.1" 304 0 "-"
"Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.21)"
```

[1] (New Relic, Inc., n.d.)

Après l'analyse, le log est organisé en attributs, tels que `response code` et `request URL`. Voici un exemple des mêmes informations de log au format de log analysable :

```
{
  "remote_addr":"93.180.71.3",
  "time":"1586514731",
  "method":"GET",
  "path":"/downloads/product_1",
  "version":"HTTP/1.1",
  "response":"304",
  "bytesSent": 0,
  "user_agent": "Debian APT-HTTP/1.3
  (0.8.16~exp12ubuntu10.21)"
}
```

Si le format est entièrement personnalisable, le paramétrage du type de log déclenche les règles d'analyse définies par le client.

Si une organisation a plusieurs applications qui servent un objectif commun, les équipes devraient se concentrer sur la standardisation d'un format de log pour toutes les applications. Cela leur permettra d'incorporer plus facilement les données à leur plateforme d'observabilité, même lorsque les équipes associées à chaque application souhaitent disposer de la visibilité sur des attributs différents.

Les formats de logs dans le détail

Une fois que l'outil d'agrégation des logs collecte les données, il y a trois catégories de format cohérent pour la structuration du texte avec des implications sur l'exploitabilité. Les trois catégories de format sont les suivantes :

- **Structuré** — JSON est l'un des formats structurés les plus courants pour les logs. De nombreux outils peuvent rapidement l'analyser. Il est très flexible et léger. Dans l'idéal, tous les logs sont générés dans un format structuré. Mais si JSON permet d'organiser hiérarchiquement les données, d'autres exemples de données de log structurées comprennent des formats courants tels que CSV et TSV.
- **Commun** — Un format commun n'est pas structuré, mais est bien connu, défini et cohérent. Le format de logs commun Apache pour les logs d'accès en est un exemple. L'avantage d'un format commun est que de nombreux outils peuvent analyser les données immédiatement.
- **Personnalisé** — Si une application ne compile pas ses logs dans un format structuré ou commun, elle les écrit alors dans un format personnalisé. Pour reconnaître le début et la fin d'une ligne de log lors du transfert, il peut s'avérer nécessaire d'effectuer une analyse. La création de règles définies par le client permettra de rendre les données plus utiles.

Catégoriser et grouper les logs

Le fait de spécifier un modèle de données pour les logs permet aux équipes d'effectuer des recherches plus efficacement. Elles devraient donc définir et inclure des attributs dès que possible pour catégoriser et grouper les logs en conséquence.

Les normes d'OpenTelemetry pour les logs ont été créées par une coalition de leaders du secteur, dont New Relic, et couvrent de nombreux éléments tels que les conventions d'attribution de noms et les définitions des valeurs de champs.[2] Les frameworks ne sont pas tous compatibles nativement avec les logs formatés exactement selon ces normes, mais ils peuvent servir de guide pratique.

Les attributs communs pouvant s'avérer utiles dans un modèle de données de log comprennent les ressources, les logs en contexte et les niveaux de log.

Ressources
Les ressources déterminent l'horodatage et la provenance des logs, par exemple :

- La date et l'heure
- Le nom d'hôte ou l'identificateur de la machine
- L'application ou le nom du service

Le nom de l'hôte peut être important dans les logs d'applications basées sur l'hôte classique dont les environnements sont nommés. Un identifiant de pod ou de conteneur organise mieux les logs d'environnement conteneurisés ou orchestrés.

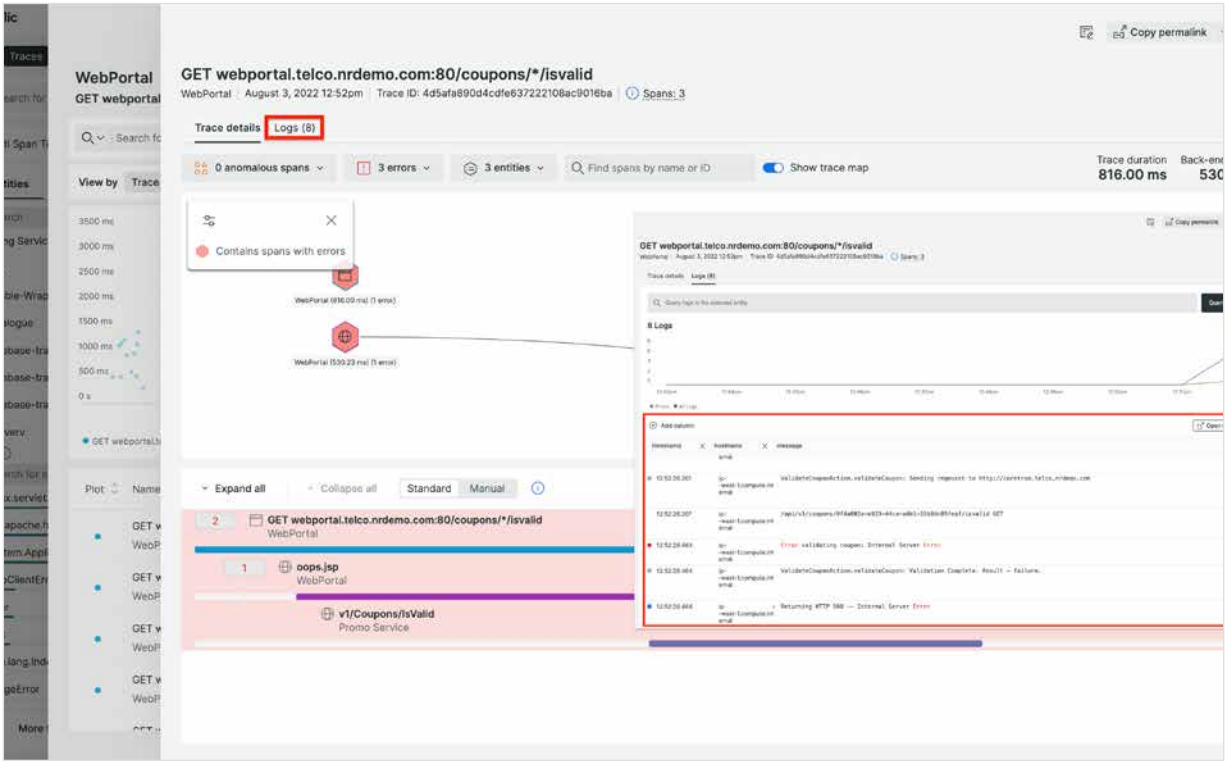
Les environnements orchestrés ou PaaS renseignent souvent automatiquement les logs avec un grand nombre de métadonnées, ce qui est parfait pour l'organisation, toutefois, il est également important d'annoter les logs avec des qualificateurs qu'un système ne peut pas connaître. Par exemple, les numéros de version des produits, les environnements de préproduction et de production, les branches de test, les versions de test A/B sont tous utiles. L'agrégation des logs signifie que tous les logs de plusieurs sources sont collectés dans le même système. Sans les bonnes métadonnées, les équipes ne peuvent pas distinguer un vrai log d'erreur en production d'une transaction qui a échoué dans le cadre d'un test.

[2] (OpenTelemetry, n.d.)

Les informations de transfert d'un log sont une autre ressource pouvant aider à identifier l'origine d'un log. Par exemple, la plupart des solutions de transfert de log fournies par New Relic annotent automatiquement les données avec le type et la version de l'outil utilisé pour envoyer les données.

Logs in Context

Il est utile que les équipes voient les logs dans le contexte des problèmes qui se produisent dans leurs applications et sur les hôtes. Par exemple, la fonctionnalité New Relic Logs in Context peut ajouter automatiquement les informations d'une application aux logs. L'agent New Relic APM fournit les données de gestion des performances des applications au framework de logging et les inclut dans les logs des applications. Résultat : Logs in Context corrèle automatiquement les données de logs avec les événements et traces des applications associées. Les erreurs et les traces distribuées d'APM sont directement liées aux logs créés pendant la même transaction que l'erreur ou la trace. Logs in Context crée cette corrélation en insérant un identifiant de span, un identifiant de trace et le nom de l'application dans les messages de log. Ainsi, les équipes peuvent regrouper les données de l'application et des logs et effectuer un dépannage beaucoup plus rapidement.



Logs filtrés pour montrer les erreurs dans le contexte de la trace sur la plateforme d'observabilité New Relic

Niveaux de logs
Les développeurs, les professionnels DevOps et les managers appellent parfois les niveaux de logs des « niveaux de sévérité ». Ces niveaux décrivent l'importance relative de l'événement (avec des termes tels que « debug », « info », « warning », « error », et « fatal ») et le niveau de densité des informations du framework de gestion des logs. L'attribut de sévérité aide à filtrer ou rejeter les informations moins critiques afin que les équipes puissent rechercher uniquement les erreurs critiques.

L'utilisation efficace des niveaux de logs peut limiter la quantité de données, réduire les coûts d'utilisation d'un outil de gestion des logs centralisé et assurer la rapidité des recherches. Dans certains cas, il peut être impossible de contrôler la façon dont les applications génèrent les logs, toutefois dans l'idéal, le système de gestion des logs peut aussi rejeter les données indésirables. Par exemple, dans New Relic, les équipes peuvent faire remonter à la surface les valeurs hors normes en utilisant des schémas guidés par l'apprentissage machine en fonction du niveau de log. Les niveaux de log codés par couleur fournissent également un indicateur visuel qui permet de porter son attention sur les zones les plus importantes.

Les équipes doivent utiliser les niveaux de logs avec précaution, et en particulier le niveau de débogage (debug). Ce niveau aide à capturer des messages très longs associés à un comportement particulier, mais un débogage inutile peut créer un volume de logs bien plus important et ralentir les fonctions d'ingestion et de recherche sans pour autant apporter plus d'éléments. Quand les équipes et les projets sont plus imposants, il peut être avantageux d'établir des normes pour les niveaux de log afin que les méthodes de regroupement, de catégorisation et de gestion soient cohérentes.

Utiliser des outils et frameworks de gestion des logs

Au lieu de passer du temps et de gaspiller des ressources à implémenter une solution de gestion des logs à partir de rien, un outil de logging éprouvé et un framework reconnu permettent de gagner du temps et d'éviter les ennuis. Par exemple, les agents de langage New Relic APM enrichissent les logs avec les métadonnées nécessaires pour donner accès à la fonctionnalité automatique des logs en contexte et de transfert des logs sans qu'il n'y ait besoin d'installer ou de maintenir des logiciels tiers, le tout dans un seul déploiement.

L'utilisation d'un framework cohérent simplifie l'adoption par les équipes d'ingénierie, normalise la sortie des logs et garantit que les équipes peuvent activer les logs en contexte de manière uniforme. De même qu'avec tout nouveau code, les équipes doivent faire preuve de prudence lors des premières utilisations des frameworks de logging et tester leur impact sur la performance.

Faire référence aux valeurs importantes, sans les inclure

Dans certains cas, les équipes auront peut-être besoin de volumes de données plus importants pour apporter un contexte plus détaillé (vidage mémoire ou jeu de fichiers ou d'images, par exemple). Il est généralement recommandé de conserver ces données séparément voire de les transférer sur un serveur désigné et de référencer leur emplacement dans le log au lieu de tout enregistrer dans celui-ci. Les équipes devraient conserver les logs les plus légers possible et accéder aux données séparément.

Partager des vues, requêtes et alertes utiles

Enfin, les équipes devraient créer et partager des vues, requêtes et alertes standard pour leurs logs. Elles pourront ainsi obtenir des informations plus globales sur l'état actuel de l'organisation et augmenter la visibilité et la communication entre les équipes. Profitez ainsi de toute la puissance de l'observabilité full-stack.

Que ne faut-il pas inclure dans les logs

Même s'il est tentant de consigner tout ce qui pourrait être utile, il existe quelques exceptions et pièges que les équipes doivent essayer d'éviter.

Informations sensibles

Les équipes doivent traiter les informations sensibles avec précaution. Il est essentiel de protéger les données réglementées, telles que les renseignements personnels et les numéros de cartes bancaires, conformément à la législation, comme le règlement général sur la protection des données (RGPD) de l'Union européenne[3] et le Health Insurance Portability and Accountability Act (HIPAA) des États-Unis.[4]

Le guide de logging de l'Open Web Application Security Project (OWASP) précise ce qui ne doit pas se trouver dans les logs, comme les jetons d'accès, les mots de passe, les informations sensibles et les renseignements que les personnes souhaitent garder privés.[5]

Pour les logs stockés sur un serveur ou une base de données privé, il est facile d'inclure accidentellement dans le log des renseignements personnels, tels que le nom ou l'adresse e-mail. Pour faire le suivi des actions ou événements d'un utilisateur particulier, il est préférable que les équipes utilisent des identifiants anonymes. Bien que les données de log soient en sécurité sur une plateforme d'observabilité comme celle de New Relic, il est important de faire très attention à ne pas transmettre de renseignements personnels en dehors de l'organisation.

Code source et données exclusives

Outre les informations réglementaires et de conformité, il est possible que les équipes ne veuillent pas inclure d'autres informations dans leurs logs, comme le code source des applications ou les données protégées au sein de l'organisation.

Outre le stockage sécurisé des logs, il est important de sécuriser aussi leur accès. Des informations qui peuvent révéler des secrets commerciaux ou des projets et fonctionnalités en cours de conception ou non annoncés n'ont pas leur place dans les logs. Les équipes ne devraient donc pas les inclure dans les logs, surtout si elles les stockent en dehors de l'entreprise avec un service tiers.

Informations en double

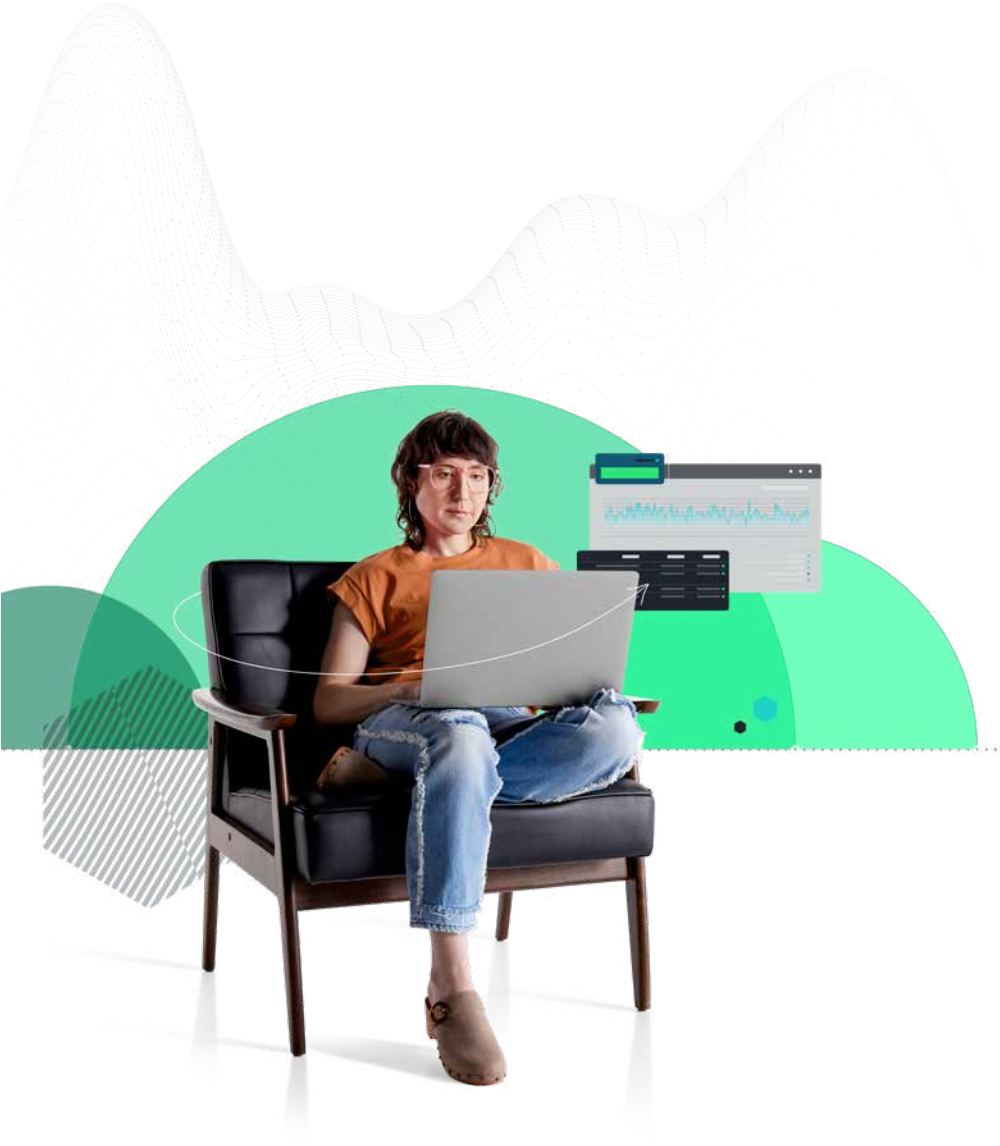
Si l'ajout d'informations en double ne cause pas de problème et s'il vaut également mieux avoir trop d'informations que pas assez, l'inclusion de beaucoup d'informations identiques peut créer des logs inutiles et entraîner des coûts plus élevés sans apporter aucun avantage.

[3] (European Commission, n.d.)
[4] (U.S. Department of Health and Human Services (HHS), n.d.)
[5] (Open Web Application Security Project (OWASP), n.d.)

Conclusion

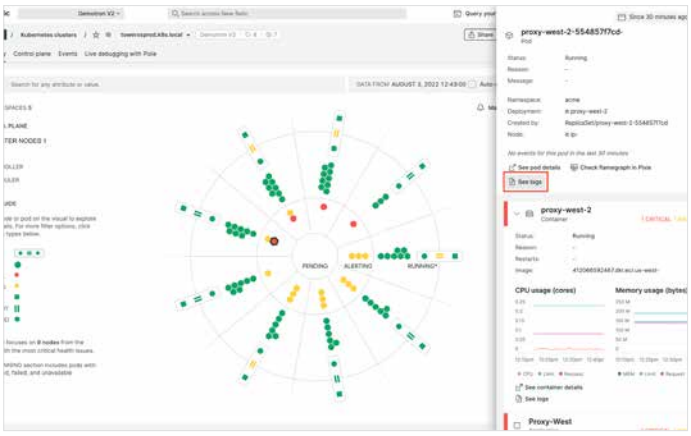
Des logs plus efficaces pour améliorer l'observabilité full-stack permettent une prise de décisions en temps réel qui a un impact sur l'activité, mais aussi un débogage plus rapide pour les développeurs et les ingénieurs qui passent ainsi moins de temps à répondre aux incidents et plus à se concentrer sur l'innovation.

Une fois ces pratiques en place, les logs peuvent apporter les détails nécessaires pour une exécution sans problème pour les clients, et une visibilité encore plus détaillée de tout le stack afin de résoudre les problèmes plus rapidement et d'accélérer le développement.



La plateforme d'observabilité New Relic

New Relic fournit une plateforme unique unifiée et uniformisée pour toutes les données télémétriques, y compris les logs détaillés. La plateforme d'observabilité New Relic incorpore la gestion des logs, l'APM, le monitoring (infrastructure, serverless, mobile, navigateur, synthétique, et Kubernetes) et le tracing distribué. Ces fonctionnalités permettent aux organisations de visualiser, analyser et dépanner tout le stack de logiciels. Dans ce cadre, New Relic Log Management permet de combiner les données de logs avec les données de monitoring des applications et de l'infrastructure, ce qui crée une plateforme d'observabilité puissante et complète.



L'APM, l'infrastructure, les événements et l'accès aux logs combinés en une seule et même vue

New Relic relie les métriques, les événements, les logs et les traces à partir de tout le stack de logiciels intégré avec AIOps (l'intelligence artificielle pour les opérations IT), ce qui permet aux organisations de rechercher les logs plus rapidement et à moindre coût par rapport aux solutions legacy disparates. Au lieu d'utiliser des outils distincts dans différentes sections du stack, les développeurs et les ingénieurs peuvent facilement visualiser tous les logs détaillés qui ont un rapport avec une erreur spécifique dans une vue unifiée et uniformisée.

Les problèmes de rapidité et de scalabilité dans les solutions de logs legacy rendent difficile l'interrogation des logs détaillés, car l'exécution avec des données différées peut prendre plusieurs minutes voire des heures. Par contre, une recherche avec la gestion des logs de New Relic prend seulement quelques secondes, ce qui permet une analyse des incidents et une réponse sur tout le stack de logiciels extrêmement rapides.

La plateforme d'observabilité New Relic comprend la gestion des logs. Celle-ci inclut un accès gratuit (Free Tier) pour les clients ayant un faible volume de données, et un prix bas par gigaoctet permettant d'ingérer tous les logs détaillés dont ils ont besoin.

Commencez à utiliser la gestion des logs New Relic, en vous inscrivant pour obtenir un compte gratuit dès aujourd'hui. Ils comprennent 100 Go/mois d'ingestion des données, un utilisateur Full Platform et un nombre illimité d'utilisateurs Basic.

S'inscrire

Références

European Commission. n.d. “EU data protection rules.” European Commission. Accessed July 19, 2022. https://ec.europa.eu/info/law/law-topic/data-protection/eu-data-protection-rules_en.

New Relic, Inc. n.d. “Parsing log data.” New Relic Documentation. Accessed July 28, 2022. https://docs.newrelic.com/docs/logs/ui-data/parsing/#custom-parsing.

OpenTelemetry. n.d. “OpenTelemetry Logging Overview.” OpenTelemetry. Accessed July 18, 2022. https://opentelemetry.io/docs/reference/specification/logs/overview/.

Open Web Application Security Project (OWASP). n.d. “OWASP Logging Guide.” https://owasp.org/www-pdf-archive/OWASP_Logging_Guide.pdf.

U.S. Department of Health and Human Services (HHS). n.d. “Summary of the HIPAA Security Rule.” HHS.gov. Accessed July 19, 2022. https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html.

new relic.