

LOG680

Introduction à l'approche DevOps

Création de télémétrie pour permettre de voir et de résoudre des problèmes

The DevOps Handbook

Part IV, Chap 14



Francis Bordeleau, 2021

Objectifs d'apprentissage

- Expliquer ce qu'est la télémétrie. Pourquoi est-elle essentielle?
- Donner des exemples concrets d'utilisation de télémétrie
- Expliquer ce qui différencie les entreprises les plus performantes ("high performers"), en comparaison aux entreprises les moins performantes ("low performers"), par rapport à l'utilisation de la télémétrie
- Expliquer en quoi consiste l'architecture de surveillance décrit par James Turnbull dans "The Art of Monitoring". Vous devez fournir un diagramme de l'architecture et décrire les principaux composants
- Expliquer en quoi consiste la centralisation de la télémétrie. Quels sont ses avantages
- Expliquer en quoi consiste les métriques au niveau des applications. Donner des exemples concrets de métriques d'application
- Expliquer en quoi consiste les métriques au niveau des affaires. Donner des exemples concrets de métriques d'affaires

Sujets

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- Conclusion

- **Introduction: Partie IV**
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- Conclusion

Introduction: Partie IV

- Comment mettre en œuvre les pratiques techniques de la deuxième voie (Second Way) pour créer un retour rapide et continu des opérations au développement
- **Raccourcir et amplifier les boucles de rétroaction** afin de pouvoir **voir les problèmes au fur et à mesure qu'ils se produisent** et **transmettre cette information à tous les utilisateurs du flux de valeur**
 - Permet de **rechercher et de résoudre rapidement les problèmes plus tôt dans le cycle de développement du logiciel**, idéalement bien avant qu'ils ne provoquent une défaillance catastrophique
- **Créer un système de travail** dans lequel **les connaissances acquises en aval dans les opérations sont intégrées aux travaux en amont du développement et de la gestion des produits**
 - Permet de **créer rapidement des améliorations et des apprentissages**
 - **Pour tous les aspects:** production, déploiement, ou modèles d'utilisation de nos clients
 - Permet à chacun d'obtenir un retour d'information sur son travail
 - Permet de tester rapidement les hypothèses de produits, ce qui nous aidera à déterminer si les fonctionnalités que nous développons permettent d'atteindre nos objectifs organisationnels

Introduction: Partie IV

- **Créer une télémétrie** à partir de nos **processus de construction**, de **test** et de **déploiement**, ainsi que du **comportement des utilisateurs**, des **problèmes de production** et des **pannes**, des **problèmes d'audit** et des **brèches de sécurité**
 - Permet de **voir et de résoudre les problèmes lorsqu'ils se présentent**
 - Permet de **développer des systèmes de travail sûrs**
 - Permet d'**apporter des modifications et de faire des expériences sur les produits**, en sachant que nous pouvons rapidement détecter et réparer les défaillances
- **Dans la mesure du possible**, nous voulons **établir un lien de cause à effet**
 - **Distinction entre corrélation et causalité**
- Renforcer les objectifs communs de gestion de produit, développement, assurance qualité, opérations et infosec
- Encourager les membres des équipes à partager la responsabilité de garantir le bon fonctionnement des services et à collaborer à l'amélioration du système dans son ensemble

Introduction: Partie IV

- Sujets principaux de la Partie IV:
 - Création de la télémétrie pour permettre de voir et de résoudre les problèmes
 - Utilisation de la télémétrie pour mieux anticiper les problèmes et atteindre les objectifs
 - Intégration de la recherche sur les utilisateurs et des commentaires dans le travail des équipes de produits
 - Activation des commentaires afin que les développeurs et les opérateurs puissent effectuer des déploiements en toute sécurité
 - Permettre une rétroaction pour augmenter la qualité de notre travail grâce à des examens par des pairs et à la programmation en binôme

- Introduction: Partie IV
- **Introduction :Chap. 14**
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- Conclusion

Introduction: Chap 14

- **Une réalité des opérations, c'est que les choses tournent mal**
 - **De petits changements peuvent entraîner de nombreux résultats inattendus**, notamment des pannes et des défaillances globales qui affectent tous nos clients
 - **Réalité de l'exploitation de systèmes complexes**, aucune personne ne peut voir l'ensemble du système et comprendre comment toutes les pièces s'emboîtent
- Lorsque des interruptions de production et d'autres problèmes surviennent dans notre travail quotidien, **nous ne disposons souvent pas des informations nécessaires pour résoudre le problème**
 - Par exemple, lors d'une panne, il se peut que nous ne puissions pas déterminer si le problème est dû à une défaillance de notre application (e.g. code défectueux), à notre environnement (e.g. problème de réseau, problème de configuration de serveur) ou quelque chose de complètement externe (e.g. une attaque par déni de service massif)

Introduction: Chap 14

- **Règle empirique pour traiter ce problème au niveau des Opérations**
 - En cas de problème en production, nous devons simplement redémarrer le serveur
 - Si cela ne fonctionne pas, redémarrez le serveur à côté
 - Si cela ne fonctionne pas, redémarrez tous les serveurs
 - Si cela ne fonctionne pas, blâmez les développeurs, ils causent toujours des pannes
- "Culture de causalité" – The Visible Ops Handbook (K. Behr, G. Kim et G. Spafford)
 - **Les entreprises les plus performantes utilise une approche disciplinée pour résoudre les problèmes**
 - Recourant à la **télémetrie de production** pour comprendre les facteurs susceptibles de contribuer à la résolution de leurs problèmes
 - Par opposition aux entreprises moins performantes qui redémarrent aveuglément des serveurs

Introduction: Chap 14

- Pour permettre un comportement discipliné de résolution de problèmes, nous devons **concevoir nos systèmes** de manière à **créer continuellement la télémétrie**
- **Télémétrie -- définition :**
« **Processus de communication automatisé par lequel des mesures et d'autres données sont collectées à des points distants et ensuite transmises à un monitoring** »
- **Objectif :**
Créer une télémétrie au sein de nos **applications** et de nos **environnements**, aussi bien dans nos **environnements de production** et de **pré-production** que dans notre **pipeline de déploiement**

Introduction: Chap 14

- L'une des conclusions du rapport "*2015 State of DevOps Report*" était que
 - Les entreprises les plus performantes ("high performers") pouvaient résoudre les incidents de production 168 fois plus rapidement que leurs pairs.
 - La valeur médiane du MTTR des "high performers" était mesuré en minutes, tandis que la celle des "low performers" était mesuré en jours
 - Les **deux meilleures pratiques techniques permettant un MTTR rapide** :
 - Utilisation du contrôle de versions par les opérations
 - **Télémetrie et surveillance proactive dans l'environnement de production**

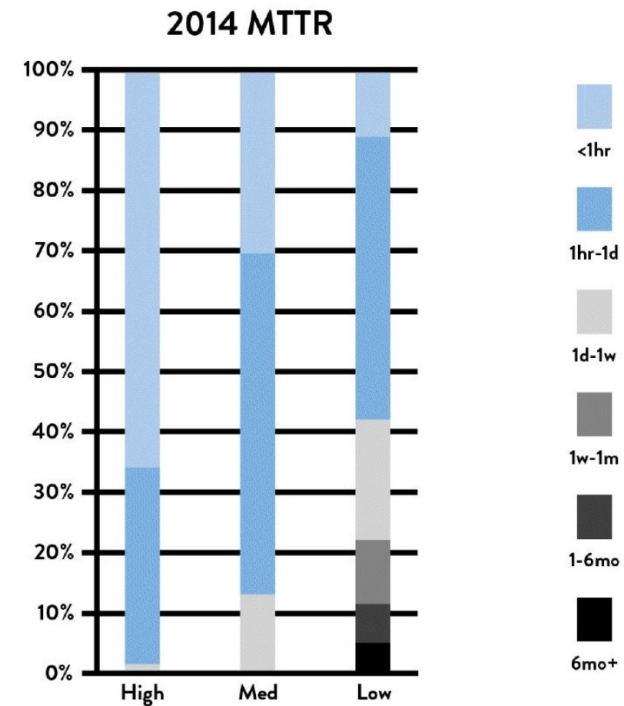


Figure 25: Incident resolution time for high, medium, and low performers
(Source: Puppet Labs, 2014 State of DevOps Report.)

Introduction: Chap 14

- L'**objectif dans ce chapitre** est de **s'assurer de toujours disposer de suffisamment de télémétrie pour pouvoir confirmer que nos services fonctionnent correctement en production**
- Lorsque des problèmes surviennent, il est possible de **déterminer rapidement ce qui ne va pas** et de **décider en connaissance de cause** de la meilleure solution, idéalement longtemps avant que les clients ne soient touchés
- De plus, la télémétrie est ce qui nous permet de **rassembler notre meilleure compréhension de la réalité** et de **détecter les cas où notre compréhension de la réalité est incorrecte**

- Introduction: Partie IV
- Introduction :Chap. 14
- **Infrastructure de télémétrie centralisée**
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- Conclusion

Surveillance opérationnelle et journalisation

- **La surveillance opérationnelle et la journalisation ne sont pas nouvelles**
 - Plusieurs générations d'ingénieurs Opérations ont utilisé et mis en place des infrastructures de surveillance personnalisées pour garantir l'intégrité des systèmes de production
 - Par exemple, HP OpenView, IBM Tivoli et BMC Patrol / BladeLogic
 - Les données étaient généralement collectées via des agents exécutés sur des serveurs ou via une surveillance sans agent
 - Par exemple, des interruptions SNMP ou des analyses basées sur des interrogations.
 - Souvent associés à une interface utilisateur graphique (GUI) et à un back-end de génération de rapports
- **Les pratiques de développement d'applications avec une journalisation efficace et de la gestion de la télémétrie résultante ne sont pas nouvelles**
 - Il existe une variété de bibliothèques de journalisation matures pour presque tous les langages de programmation

Infrastructure de télémétrie centralisée

- **Contexte:** pendant des décennies, nous avons créé des **silos d'informations**, dans lesquels Dev ne crée des événements de journalisation que pour les développeurs, et Ops contrôle uniquement si les environnements sont actifs ou non
- **Résultat:** lorsque des événements inopportuns se produisent, **personne ne peut déterminer pourquoi le système ne fonctionne pas comme prévu ou quel composant spécifique échoue**, ce qui empêche notre système de fonctionner à nouveau
- **Solution:** Pour que nous puissions voir tous les problèmes au fur et à mesure qu'ils surviennent, nous devons **concevoir et développer nos applications et nos environnements de manière à ce qu'ils génèrent une télémétrie suffisante**
 - Permet de comprendre le comportement de notre système dans son ensemble
 - Lorsque tous les niveaux de notre pile d'applications sont surveillés et enregistrés, nous activons d'autres fonctionnalités importantes, telles que la représentation graphique et la visualisation de nos métriques, la détection des anomalies, l'alerte et l'escalade proactives, etc.

Architecture de surveillance

- James Turnbull , "The Art of Monitoring"
Architecture de surveillance ("Monitoring Architecture") développée et utilisée par les ingénieurs d'exploitation d'entreprises Web (e.g. Google, Amazon, Facebook)
- Consiste souvent en un ensemble d'outils open source, tels que Nagios et Zenoss, qui ont été personnalisés et déployés à une échelle difficile à réaliser avec un logiciel commercial sous licence à l'époque
- Cette architecture comprend les composants suivants:
 - **Collecte de données**
 - **Routeur d'événements**

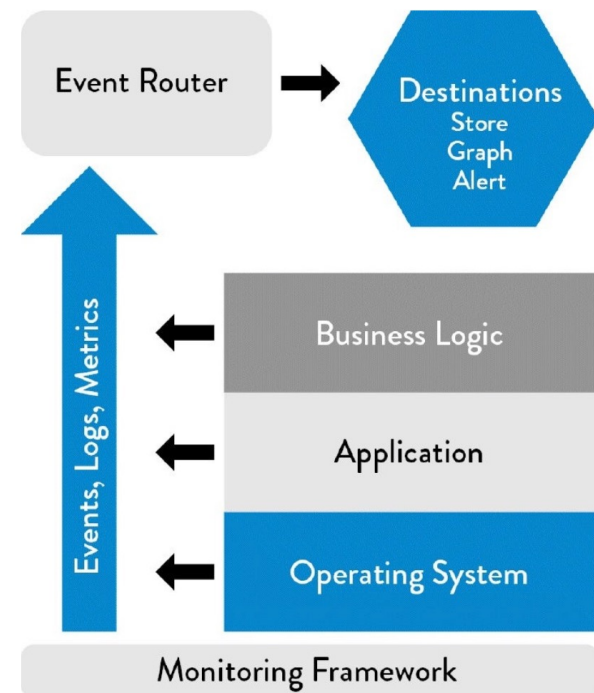


Figure 26: Monitoring framework (Source: Turnbull, The Art of Monitoring, Kindle edition, chap. 2.)

Collecte de données

- **Collecte de données** au niveau des **couches logique, application et environnement**
 - **Création de télémétrie** sous forme d'événements, de **journaux** et de **métriques** à **chacune des couches**
 - **Les journaux** peuvent être **stockés dans des fichiers spécifiques** à **chaque application** sur **chaque serveur** (par exemple, /var/log/httpd-error.log)
 - **Envoi de tous les journaux à un service commun** permettant une centralisation, une rotation et une suppression faciles
 - Aussi, nous **recueillons des métriques** sur **toutes les couches de la pile d'applications** pour **mieux comprendre le comportement de notre système**
 - Au niveau du système d'exploitation, nous pouvons collecter des métriques telles que l'utilisation du processeur, de la mémoire, du disque ou du réseau à l'aide d'outils tels que collectd, Ganglia, etc.
 - D'autres outils collectant des informations sur les performances incluent AppDynamics, New Relic et Pingdom

Routeur d'événements

- **Routeur d'événements** responsable du **stockage des événements et des métriques**
 - Permet la **visualisation**, l'**établissement de tendances**, la **création d'alerte**, la **détection d'anomalies**, etc.
 - En collectant, en stockant et en agrégeant toute notre télémétrie, nous permettons de **mieux analyser** et **vérifier l'état de santé**
 - Sert également à stocker les configurations liées à nos services (et à leurs applications et environnements de support), et probablement l'endroit où nous effectuons des vérifications de l'état de fonctionnement du système et des alertes basées sur l'atteinte de seuil

Infrastructure de télémétrie centralisée

- Une fois que nous avons centralisé nos journaux, nous pouvons **les transformer en métriques**
 - Par exemple, un événement de journal tel que "child pid 14024 exit signal Segmentation fault" peut être considéré et résumé en tant que métrique de segfault unique sur l'ensemble de l'infrastructure de production
- En transformant les événements de journaux en métriques, nous pouvons désormais y **effectuer des opérations statistiques**
 - E.g. détection d'anomalies, afin de détecter les valeurs aberrantes et les écarts le plus tôt possible
 - Permet de configurer une alerte pour nous avertir si nous passons de «dix segfaults la semaine dernière» à «des milliers de segfaults au cours de la dernière heure
- De plus, nous devons nous **assurer qu'il est facile d'entrer et de récupérer des informations depuis notre infrastructure de télémétrie**
 - De préférence, tout devrait être fait via des API en libre-service, au lieu d'obliger les gens à ouvrir des tickets et à attendre d'avoir des rapports

Infrastructure de télémétrie centralisée

- Idéalement, nous devons créer une télémétrie qui nous indique exactement **quand** quelque chose d'intérêt se produit, ainsi que **où** et **comment**
- Notre télémétrie devrait également convenir aux **analyses manuelles** et **automatisées** et devrait **pouvoir être analysée sans disposer de l'application qui a produit les journaux**
 - «La surveillance est tellement importante que nos systèmes de surveillance doivent être plus disponibles et plus évolutifs que les systèmes surveillés.» -- Adrian Cockcroft
- Télémétrie vs métrique
 - Les termes *télémétrie* et *métrique* sont utilisés de manière interchangeable
 - Comprend toutes les journalisations d'événements et les métriques créées par nos services à tous les niveaux de notre pile d'applications et générées à partir de tous nos environnements de production et de pré-production, ainsi que de notre pipeline de déploiement

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- **Création de télémétrie de journalisation d'application**
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- Conclusion

Télémétrie de journalisation d'application

- Maintenant que nous avons une infrastructure de télémétrie centralisée, nous devons nous assurer que les applications que nous construisons et exploitons créent suffisamment de télémétrie
 - Nous devons faire en sorte que les ingénieurs de **Dev et Ops créent la télémétrie de production dans le cadre de leur travail quotidien**, tant pour les services nouveaux que ceux déjà existants
- Scott Prugh (Architecte en chef et VP développement chez CSG)

«Chaque fois que la NASA lance une fusée, des millions de capteurs automatisés signalent l'état de chaque composant de cet actif précieux.

Et pourtant, nous ne prenons souvent pas le même soin avec les logiciels - nous avons constaté que **la création de télémétrie** applicative et d'infrastructure était **l'un des investissements les plus rentables** que nous ayons faits.

En 2014, nous avons créé plus d'un milliard d'événements de télémétrie par jour, avec plus de cent mille localisations de code instrumentées. ”

Télémétrie de journalisation d'application

- Dans les applications que nous créons et exploitons, **chaque fonction doit être instrumentée**
 - Si elle est suffisamment importante pour être mise en œuvre, elle est certainement assez importante pour générer suffisamment de télémétrie de production afin de pouvoir confirmer qu'elle fonctionne comme prévu et que les résultats sont atteints
- **Chaque membre de notre chaîne de valeur utilisera la télémétrie de différentes manières**
 - Les développeurs peuvent créer temporairement plus de télémétrie dans leur application pour mieux diagnostiquer les problèmes sur leur station de travail
 - Les ingénieurs d'Ops peuvent utiliser la télémétrie pour diagnostiquer un problème de production
 - Infosec et les auditeurs peuvent examiner la télémétrie pour confirmer l'efficacité d'un contrôle requis
 - Un responsable de produits peut les utiliser pour suivre les résultats commerciaux, l'utilisation des fonctionnalités ou les taux de conversion

Télémétrie de journalisation d'application

- Utilisation de **différents niveaux de journalisation**, dont certains peuvent également déclencher des alertes
 - **Niveau DEBUG**
 - Concernent tout ce qui se passe dans le programme, le plus souvent utilisé lors du débogage
 - Les journaux de débogage sont souvent désactivés en production mais temporairement activés lors du dépannage
 - **Niveau INFO**
 - Actions dirigées par l'utilisateur ou spécifiques à un système (e.g. "début de transaction par carte de crédit")
 - **Niveau WARN**
 - Conditions pouvant potentiellement devenir une erreur (e.g. un appel à la base de données prenant plus de temps qu'une certaine durée prédéfinie)
 - Déclencheront probablement une alerte et un dépannage, tandis que d'autres messages de journalisation pourraient nous aider à mieux comprendre ce qui a conduit à cette condition
 - **Niveau ERROR:**
 - Conditions d'erreur (e.g. échecs d'appels d'API, conditions d'erreur internes)
 - **Niveau FATAL**
 - Indiquent quand nous devons terminer (e.g. un démon réseau ne peut pas lier une socket réseau)

Téléométrie de journalisation d'application

- **Choisir le bon niveau de journalisation est important.**

«Lorsque vous décidez si un message doit être ERREUR ou WARN, imaginez être réveillé à 4 heures du matin. Une faible quantité de toner dans l'imprimante ne devrait pas être une ERREUR. »

-- Dan North (ancien consultant de ThoughtWorks qui a participé à plusieurs projets dans lesquels les concepts clés de livraison continue ont pris forme)

- Pour nous assurer que nous disposons des informations pertinentes pour le fonctionnement fiable et sécurisé de nos services, nous devons nous assurer que **tous les événements d'application potentiellement importants génèrent des entrées de journalisation**, y compris celles comprises dans la liste dressée par Anton A. Chuvakin – voir page suivante
- **Pour faciliter l'interprétation** de toutes les entrées de journal, nous devons (idéalement) **créer des catégories hiérarchiques de journalisation**, telles que les attributs non fonctionnels (performance, sécurité, par exemple) et les attributs associés à des fonctionnalités (recherche, classement, etc.)

Téléométrie de journalisation d'application

- Liste dressée par Anton A. Chuvakin (VP recherche, Gartner's GTP Security and Risk Management group) :
 - Décisions d'authentification / autorisation (y compris fermeture de session)
 - Accès au système et aux données
 - Modifications apportées au système et à l'application (modifications privilégiées en particulier)
 - Modifications de données, telles que l'ajout, la modification ou la suppression de données
 - Entrée invalide (injection malveillante possible, menaces, etc.)
 - Ressources (RAM, disque, CPU, bande passante ou toute autre ressource ayant des limites strictes ou modérées)
 - Santé et disponibilité
 - Les démarrages et les arrêts
 - Défauts et erreurs
 - Disjoncteurs
 - Retards
 - Sauvegarde réussie / échec

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- **Utilisation de la télémétrie**
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- Conclusion

Absence de télémétrie

- **Les organisations les plus performantes utilisent une approche disciplinée pour résoudre les problèmes**
- **Contraste avec la pratique plus courante basée sur la rumeur et les oui-dire**
 - L'absence de télémétrie publique entraîne une atmosphère politique très chargée, la nécessité de détourner les accusations et, pire encore, l'incapacité de créer une connaissance institutionnelle sur la manière dont les incidents se sont produits et les enseignements à tirer pour éviter que de telles erreurs ne se reproduisent à nouveau dans le futur

Utilisation de la télémétrie

- **La télémétrie nous permet d'utiliser la méthode scientifique pour formuler des hypothèses sur la cause d'un problème et ce qui est nécessaire pour le résoudre**
- Exemples de questions auxquelles nous pouvons répondre lors de la résolution d'un problème:
 - Quelle preuve avons-nous de notre surveillance (monitoring) qu'un problème se produit réellement?
 - Quels sont les événements pertinents, et les changements dans nos applications et environnements qui pourraient avoir contribué au problème?
 - Quelles hypothèses pouvons-nous formuler pour confirmer le lien entre les causes et les effets proposés?
 - Comment pouvons-nous prouver laquelle de ces hypothèses est correcte et réussir à résoudre le problème?
- La résolution de problèmes basée sur des faits résulte non seulement dans un MTTR nettement plus rapide (et de meilleurs résultats clients), mais également dans le renforcement de la perception d'une relation gagnant / gagnant entre développement et opérations

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- **Création de métriques de production**
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- Conclusion

Création de métriques de production

- Nous devons **activer la création de métriques de production dans le cadre du travail quotidien**
 - Pour **permettre à tous de trouver et de résoudre des problèmes dans leur travail quotidien**, nous devons permettre à chacun de créer des métriques dans leur travail quotidien pouvant être facilement créées, affichées et analysées
 - Nous devons **créer l'infrastructure et les bibliothèques nécessaires** pour **permettre à quiconque** dans le domaine du développement ou des opérations **de créer la télémétrie pour toutes les fonctionnalités qu'ils développent aussi facilement que possible**
 - Dans l'idéal, il **devrait être aussi simple que d'écrire une ligne de code pour créer une nouvelle métrique** qui apparaît dans un tableau de bord commun et visible par tous les utilisateurs du flux de valeur
- C'est la philosophie qui a guidé le développement de l'une des bibliothèques de mesures les plus utilisées, StatsD, qui a été créée et mises en open-source par Etsy

StatsD

- **Faciliter l'instrumentation du code**

« Ils [les développeurs] peuvent désormais le faire avec une seule ligne de code. Il était important pour nous que pour un développeur, l'ajout de la télémétrie de production ne soit pas aussi difficile que de modifier un schéma de base de données. » -- John Allspaw

- StatsD peut **générer des "timers" et des compteurs avec une seule ligne de code**

- Ruby, Perl, Python, Java et d'autres langages

- **Souvent utilisé avec Graphite ou Grafana**

- Convertit les événements et métriques en graphiques et en tableaux de bord

- Lorsque nous générons des graphiques de notre télémétrie, nous les **superposons également lorsque des changements de production se produisent**

- La grande majorité des problèmes de production sont causés par des changements de production, notamment des déploiements de code

- Des **bibliothèques alternatives** à StatsD permettant aux développeurs de générer une télémétrie de production **peuvent être facilement agrégées et analysées**, notamment les métriques JMX et codahale

Exemple StatsD

- Exemple de la manière dont une seule ligne de code crée un événement de connexion utilisateur
 - Dans ce cas, une ligne de code PHP: «StatsD :: increment (« login.successes »)»
 - Le graphique obtenu indique le nombre de connexions réussies et ayant échoué par minute
 - Les lignes verticales superposées sur le graphique représentent un déploiement en production

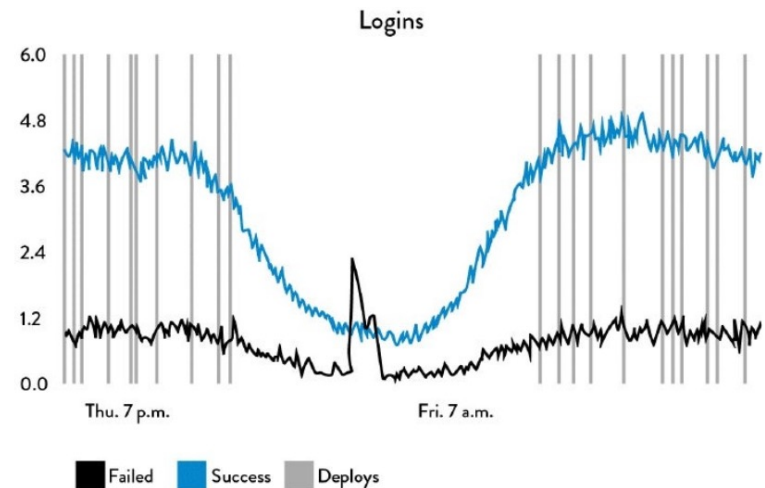


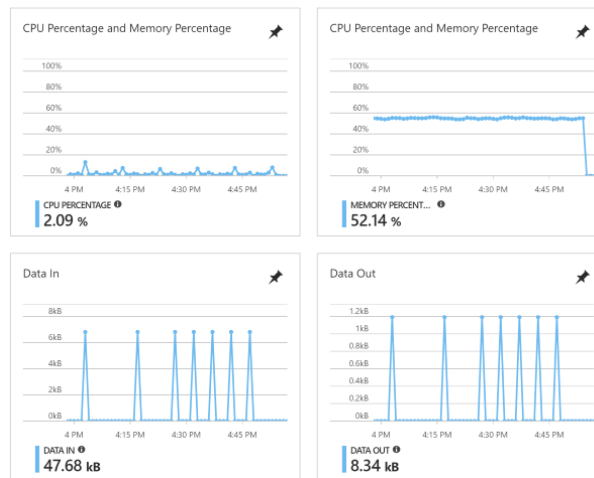
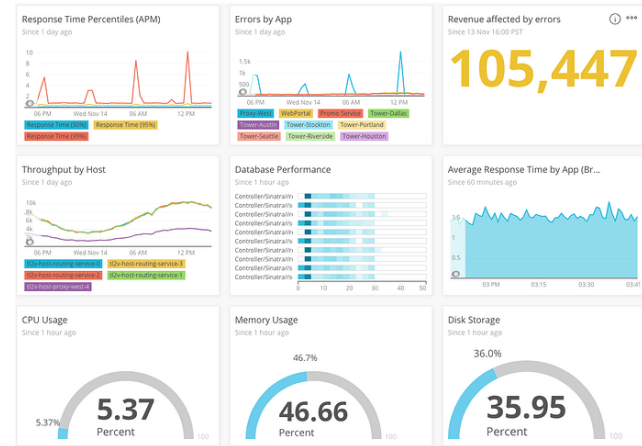
Figure 27: One line of code to generate telemetry using StatsD and Graphite at Etsy (Source: Ian Malpass, "Measure Anything, Measure Everything.")

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- **Accès libre service aux distributeurs de télémétrie et d'information**
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- Conclusion

Distributeurs de télémétrie et d'information

- Dans les étapes précédentes, nous avons permis à Dev et Ops de créer et d'améliorer la télémétrie de production dans le cadre de leur travail quotidien.
- Dans cette étape, notre **objectif** est de **transmettre cette information au reste de l'organisation**
 - Veiller à ce que toute personne souhaitant obtenir des informations sur l'un des services que nous exploitons puisse l'obtenir sans avoir besoin d'un accès au système de production ou de comptes privilégiés, ni d'ouvrir un ticket et d'attendre pendant des jours que quelqu'un configure le graphique pour eux.

Distributeurs de télémétrie et d'information



Distributeurs de télémétrie et d'information

- **En rendant la télémétrie rapide, facile à obtenir et suffisamment centralisée, tous les acteurs de la chaîne de valeur peuvent partager une vision commune de la réalité**
 - En règle générale, cela signifie que les mesures de production seront diffusées sur des pages Web générées par un serveur centralisé, tel que Graphite ou autre technologie précédemment décrite
- **Nous voulons que notre télémétrie de production soit très visible**
 - Nécessite de la placer dans des zones centrales où travaillent Dev et Ops
 - Permet à tous ceux qui sont intéressés de voir comment les services fonctionnent
 - Inclut tous les membres de notre flux de valeur, tels que Dev, Ops, la gestion de produits et Infosec
- **Distributeur d'information ("information radiator")**
 - Terme générique désignant l'un des nombreux affichages manuscrits, graphiques, imprimés ou électroniques qu'une équipe place dans un endroit très visible, afin que tous les membres de l'équipe ainsi que les passants puissent voir les dernières informations d'un coup d'œil: nombre de tests automatisés, vitesse, rapports d'incidents, état d'intégration continue, etc. – Agile Alliance
 - Cette idée est née dans le cadre du système de production Toyota

Distributeurs de télémétrie et d'information

- En plaçant les distributeurs d'information dans des endroits très visibles, nous favorisons la responsabilité parmi les membres de l'équipe, en mettant en évidence les valeurs suivantes:
 - L'équipe n'a **rien à cacher à ses visiteurs** (clients, parties prenantes, etc.)
 - L'équipe n'a **rien à se cacher**: elle reconnaît et fait face aux problèmes
- Maintenant que nous possédons l'infrastructure pour créer et diffuser la télémétrie de production à l'ensemble de l'organisation, **nous pouvons également choisir de diffuser ces informations à nos clients internes et même à nos clients externes**
 - Par exemple, nous pourrions le faire en créant des pages d'état de service accessibles au public afin que les clients puissent savoir comment les services dont ils dépendent fonctionnent
 - Cela démontre que nous attachons de la valeur à la transparence, contribuant ainsi à créer et à gagner la confiance des clients. ** Voir l'annexe 10

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- **Trouver et combler les lacunes de télémétrie**
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- Conclusion

Trouver et combler les lacunes de télémétrie

- Nous avons maintenant créé l'infrastructure nécessaire pour créer rapidement une télémétrie de production dans l'ensemble de notre pile d'applications et la diffuser dans l'ensemble de notre organisation
- Au cours de cette étape, nous **identifierons les lacunes de notre télémétrie** qui pourraient entraver notre capacité à détecter et à résoudre rapidement les incidents
 - Cela est particulièrement pertinent si Dev et Ops ont présentement peu (ou pas) de télémétrie
 - Nous utiliserons ces données ultérieurement pour mieux anticiper les problèmes et permettre à chacun de rassembler les informations nécessaires pour prendre de meilleures décisions afin d'atteindre les objectifs de l'organisation
- Pour ce faire, nous **devons créer suffisamment de télémétrie à tous les niveaux de la pile d'applications pour tous nos environnements**, ainsi que pour **les pipelines de déploiement qui les prennent en charge**

Trouver et combler les lacunes de télémétrie

- Nous avons besoin de métriques aux niveaux suivants:
 - **Niveau commercial**
 - E.g. nombre de transactions de vente, le revenu des ventes, les inscriptions des utilisateurs, le taux de résiliation, les résultats des tests A/B, etc.
 - **Niveau d'application**
 - E.g. temps de transaction, temps de réponse de l'utilisateur, erreurs d'application, etc.
 - **Niveau d'infrastructure** (e.g. BD, système d'exploitation, réseau, stockage)
 - E.g. trafic sur le serveur Web, charge sur le CPU, utilisation de la mémoire, etc.
 - **Niveau de logiciel client** (e.g. JavaScript sur le navigateur client ou l'application mobile)
 - E.g. erreurs d'application, crash, temps de transaction mesurés par l'utilisateur, etc.
 - **Niveau du pipeline de déploiement**
 - E.g. état du pipeline de construction (rouge ou vert pour nos diverses suites de tests automatisés), délais de déploiement, fréquences de déploiement, promotions de l'environnement de test et le statut de l'environnement
- **En disposant d'une couverture télémétrique dans tous ces domaines, nous pourrions voir la santé de tout ce sur quoi notre service s'appuie, en utilisant des données et des faits** au lieu de rumeurs, de pointer du doigt, de blâmer, etc.

Trouver et combler les lacunes de télémétrie

- Nous permettons de mieux **détecter les événements liés à la sécurité** en surveillant les défauts d'application et d'infrastructure
 - E.g. des terminaisons de programme anormales, des erreurs et exceptions d'application, ainsi que des erreurs de serveur et de stockage)
 - Non seulement cette télémétrie informe-t-elle mieux le développement et les opérations lorsque nos services tombent en panne, mais ces erreurs indiquent souvent qu'une vulnérabilité en matière de sécurité est activement exploitée
- En **détectant et en corrigeant les problèmes plus tôt, nous pouvons les résoudre lorsqu'ils sont petits et faciles à résoudre, avec moins de clients touchés**
 - **Après chaque incident de production**, nous devrions **identifier toute télémétrie manquante** qui aurait pu permettre une détection et une récupération plus rapides
 - Nous **pouvons aussi identifier ces lacunes lors du développement des fonctionnalités de notre processus d'évaluation par les pairs**

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- **Métriques d'application et d'affaires**
- Métriques d'infrastructure
- Autres informations
- Conclusion

Métriques d'application et d'affaires

- **Objectif:** nous assurer que nous générons la télémétrie non seulement autour de la **santé des applications**, mais également dans quelle mesure nous atteignons nos **objectifs organisationnels**
 - **Santé des applications:** utilisation de la mémoire, nombre de transactions, etc.
 - **Objectifs organisationnels :** nombre de nouveaux utilisateurs, événements de connexion, durée d'une session, pourcentage d'utilisateurs actifs, fréquence d'utilisation de certaines fonctionnalités, etc.
 - Par exemple, si nous avons un service qui prend en charge le commerce électronique, nous voulons nous assurer que la télémétrie concerne tous les événements utilisateur menant à une transaction réussie générant des revenus
 - Nous pouvons ensuite instrumenter toutes les actions de l'utilisateur requises pour obtenir les résultats souhaités pour le client

Métriques d'application et d'affaires

- Ces **mesures varient selon les différents domaines et objectifs de l'organisation**
 - Pour les sites de commerce électronique, nous souhaitons peut-être optimiser le temps passé sur le site
 - Toutefois, pour les moteurs de recherche, nous souhaitons peut-être réduire le temps passé sur le site, car de longues sessions peuvent indiquer que les utilisateurs ont du mal à trouver ce qu'ils recherchent
- En général, les **métriques d'affaires** feront partie d'un **entonnoir d'acquisition de clients** ("customer acquisition funnel"), qui correspond aux étapes théoriques qu'un client potentiel doit suivre pour effectuer un achat
 - Par exemple, sur un site de commerce électronique, les événements de parcours mesurables incluent le temps total passé sur le site, les clics sur les liens vers les produits, les annonces dans le panier et les commandes terminées

Métriques d'application et d'affaires

- **Objectif:** faire en sorte que **chaque métrique d'affaire soit exploitable** –
 - Ces métriques de premier plan devraient aider à indiquer comment changer notre produit et permettre l'expérimentation et les tests A/B
 - Lorsque les métriques ne sont pas exploitables, il s'agit probablement de mesures simples qui fournissent peu d'informations utiles – ce sont des informations que nous voulons stocker, mais pas afficher, et encore moins créer des alertes
- Idéalement, toute personne consultant nos distributeurs d'information sera en mesure de donner un sens aux informations que nous montrons dans le contexte des résultats organisationnels souhaités, tels que les **objectifs en termes de revenus**, **d'engagement d'utilisateurs**, de **taux de conversion**, etc.
- Nous devons définir et **relier chaque métrique à une mesure des résultats de l'entreprise** dès les premières étapes de la définition et du développement des fonctionnalités, et mesurer les résultats après leur déploiement en production
 - De plus, cela aide les propriétaires de produits à décrire le contexte commercial de chaque fonctionnalité pour tous les utilisateurs du flux de valeur

Métriques d'application et d'affaires

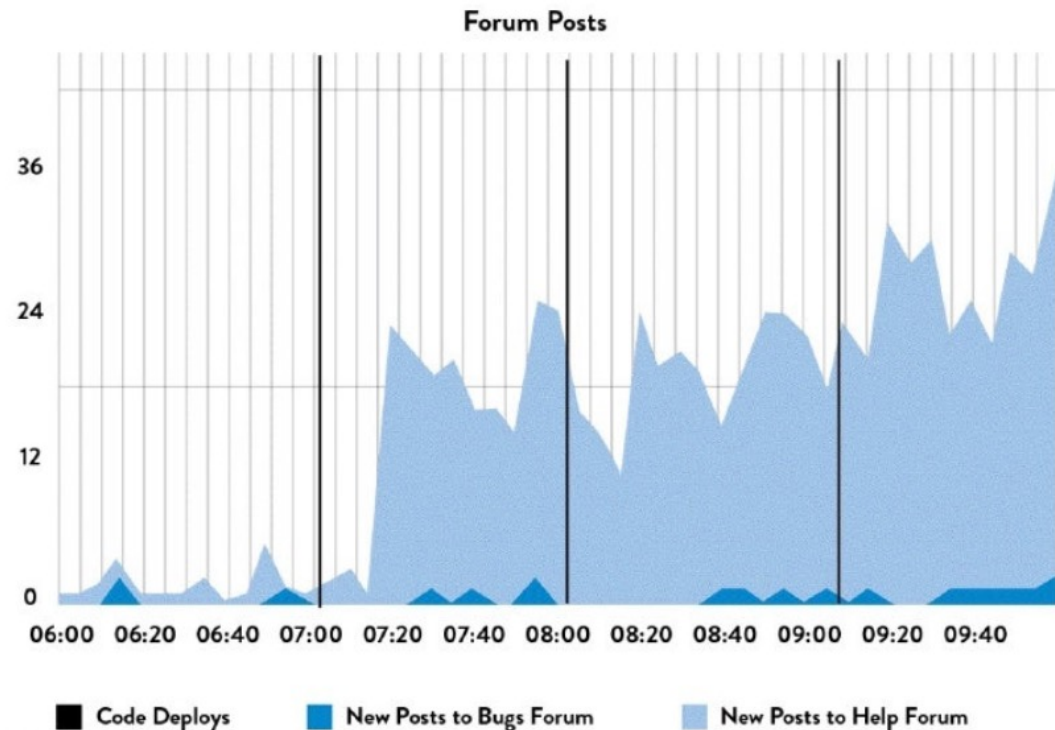


Figure 28: Amount of user excitement of new features in user forum posts after deployments (Source: Mike Brittain, "Tracking Every Release," CodeasCraft.com, December 8, 2010, <https://codeascraft.com/2010/12/08/track-every-release/>.)

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- **Métriques d'infrastructure**
- Autres informations
- Conclusion

Métriques d'infrastructure

- Objectif: **générer suffisamment de télémétrie afin de pouvoir déterminer rapidement si un problème survient dans n'importe quel environnement, si l'infrastructure est une des causes du problème**
 - Nous devons être en mesure de déterminer exactement ce qui dans l'infrastructure contribue au problème (base de données, système d'exploitation, stockage, mise en réseau, etc.)
- Nous voulons rendre visible le plus possible la télémétrie d'infrastructure, à travers tous les acteurs de la technologie, idéalement organisée par service ou application
 - En d'autres termes, lorsque quelque chose ne va pas dans notre environnement, nous devons savoir exactement quels applications et services pourraient être ou sont affectés
- En plus de surveiller nos services de production, nous avons également besoin de télémétrie pour ces services dans nos environnements de pré-production par exemple, développement, test, transfert, etc.)
 - Permet de résoudre les problèmes avant leur mise en production

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- **Autres informations**
- Conclusion

Autres informations

- Même après que nous ayons créé notre pipeline de déploiement qui nous permet d'apporter des modifications de production petites et fréquentes, les modifications créent toujours un risque inhérent
- Les effets secondaires opérationnels ne sont pas simplement des pannes, mais également des perturbations et des écarts importants par rapport aux opérations standard
- Pour rendre les modifications visibles, nous rendons le travail visible en superposant toutes nos activités de déploiement et de production sur nos graphiques
 - Par exemple, pour un service qui traite un grand nombre de transactions entrantes, les modifications de la production peuvent entraîner une période de stabilisation importante, au cours de laquelle les performances se dégradent considérablement lorsque toutes les recherches dans la mémoire cache sont manquantes
- Pour mieux comprendre et préserver la qualité de service, nous souhaitons comprendre à quelle vitesse les performances reviennent à la normale et, si nécessaire, prendre des mesures pour améliorer les performances
- De même, nous souhaitons superposer d'autres activités opérationnelles utiles, telles que la maintenance ou la sauvegarde du service, à des emplacements où il est possible d'afficher ou de supprimer les alertes

- Introduction: Partie IV
- Introduction :Chap. 14
- Infrastructure de télémétrie centralisée
- Création de télémétrie de journalisation d'application
- Utilisation de la télémétrie
- Création de métriques de production
- Accès libre service aux distributeurs de télémétrie et d'information
- Trouver et combler les lacunes de télémétrie
- Métriques d'application et d'affaires
- Métriques d'infrastructure
- Autres informations
- **Conclusion**

Conclusion

- Les améliorations apportées par la télémétrie de production d'Etsy et de LinkedIn nous montrent à quel point il est crucial de voir les problèmes au fur et à mesure de leur apparition, ce qui nous permet de rechercher la cause et de remédier rapidement à la situation
- **En analysant tous les éléments de notre service émettant de la télémétrie**, que ce soit dans notre application, notre base de données ou dans notre environnement, et **en rendant cette télémétrie largement disponible, nous pouvons détecter et résoudre les problèmes bien avant qu'ils ne causent une catastrophe**
 - Idéalement longtemps avant même qu'un client remarque que quelque chose ne va pas
- **Le résultat est non seulement des clients plus satisfaits, mais en réduisant le nombre de pompiers et de crises en cas de problème, nous avons un lieu de travail plus heureux et plus productif, avec moins de stress et moins d'épuisement professionnel**