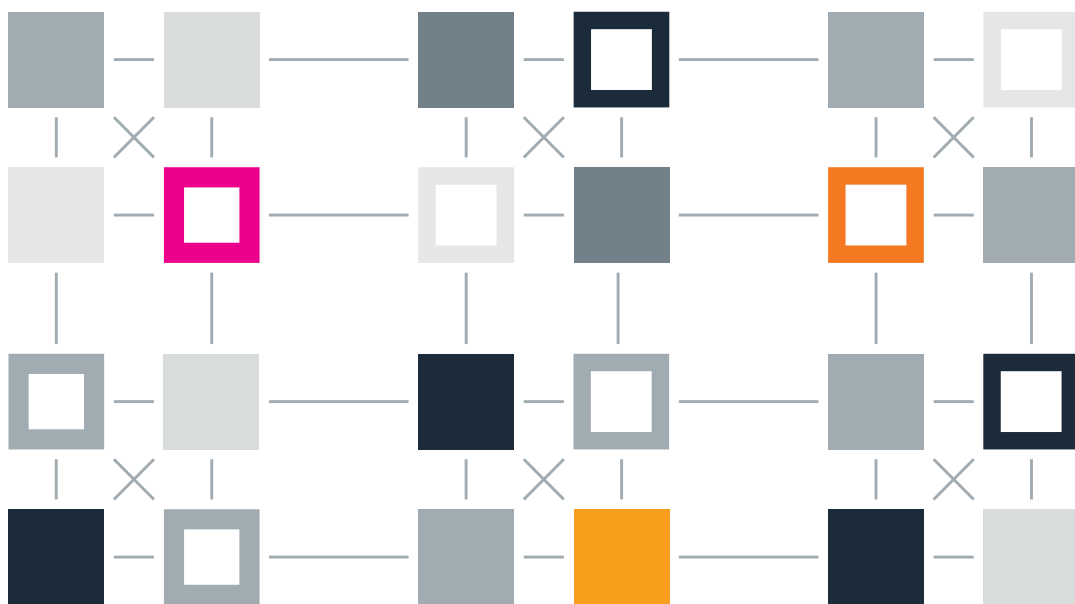


Superviser les microservices dans Kubernetes



Selon la dernière étude bisannuelle de la CNCF, la supervision est l'un des plus grands défis des entreprises qui adoptent Kubernetes.

Ce livre blanc apporte des éclairages sur les meilleures approches de la supervision des clusters Kubernetes et des microservices déployés sur Kubernetes, dans le but de localiser les problèmes de performance et minimiser le temps moyen de résolution (MTTR).

La gestion des problèmes de performance en temps réel constitue un moyen contrôlable d'assurer une expérience utilisateur irréprochable. Les entreprises doivent établir des bonnes pratiques pour administrer les clusters Kubernetes et pour comprendre les performances sur l'intégralité de la pile, en incluant les traces distribuées sur tous les microservices.

Le cerveau derrière l'orchestration des applications en conteneurs

Kubernetes est une solution d'orchestration de conteneurs de pointe pour le déploiement d'applications en conteneurs. Des entreprises de tous les secteurs adoptent les microservices en conteneurs pour leurs nombreux avantages :

- **Vitesse** : accélérer le rythme de l'innovation ;
- **Échelle** : délivrer des expériences d'application optimisées sur différentes régions géographiques et fournisseurs de cloud ;
- **Efficacité** : apporter plus de valeur à un coût plus bas en optimisant la consommation des ressources.

Kubernetes inclut plusieurs couches logiques qui appuient différents scénarios d'utilisation et gèrent le cycle de vie des microservices en conteneurs.

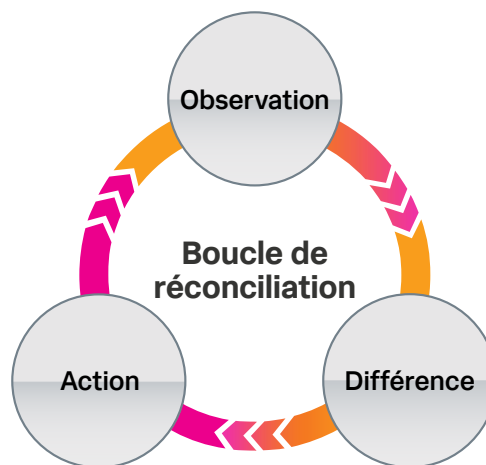
La philosophie de déploiement de Kubernetes repose sur trois piliers fondamentaux

Abstraction de l'infrastructure	Configuration déclarative	Immuabilité
Orchestration des applications à partir des ressources de l'infrastructure (serveurs de calcul, volumes de stockage et réseaux)	Kubernetes supervise constamment l'état des applications déployées et des objets Kubernetes. Il implémente également l'état souhaité du système selon les instructions des opérateurs, par exemple en gérant le nombre de répliques d'un service en cours d'exécution	Conformément à l'exigence d'immuabilité imposée par Docker, où les images des conteneurs sont immuables, les objets Kubernetes sont également immuables : des versions différentes des services Kubernetes sont entièrement séparées et ne sont pas interchangeables

Pourquoi utiliser Kubernetes ?

Outre les avantages de la conteneurisation pour les développeurs d'applications, Kubernetes facilite les opérations à de nombreux égards :

Gestion automatique du cycle de vie des applications	Interopérabilité du cloud	Système capable de se réparer lui-même
<p>Kubernetes automatise la gestion du cycle de vie des applications : cette solution de bout en bout inclut le provisionnement des applications, leur dimensionnement automatique et la gestion des répliques, ainsi que l'intégration continue et la livraison continue (CI/CD)</p>	<p>Kubernetes ajoute une couche d'abstraction sur l'infrastructure pour permettre aux applications de devenir vraiment portables d'un cloud à l'autre</p>	<p>Kubernetes protège les applications contre les interruptions de service et les comportements imprévisibles qui déstabilisent le système. Il agit constamment pour garantir la conformité de l'état du système à l'état désiré décrit dans la configuration déclarative</p>

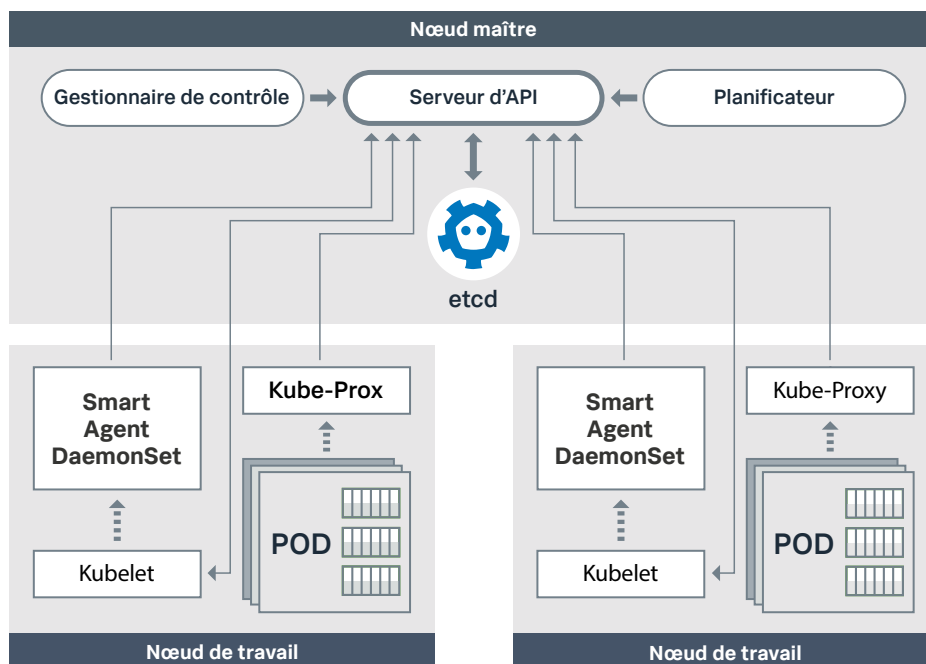


Le gestionnaire de contrôle Kubernetes compare continuellement l'état souhaité à l'état réel, évalue les différences entre les deux et les résout en prenant les mesures appropriées.

Mieux comprendre les composants Kubernetes

Kubernetes est un système qui assemble un groupe de machines en une unité unique assimilable via une API. Kubernetes segmente ensuite les ressources de calcul en deux groupes :

nœuds de travail et nœuds maîtres



Voici les principaux composants exécutés sur les nœuds maîtres et les nœuds de travail :

Nœuds maîtres	Nœuds de travail
<p style="text-align: center;">Serveur d'API</p> <ul style="list-style-type: none"> • Passerelle vers le cluster Kubernetes • Transmet toutes les requêtes émanant des clients et des objets d'API stockés dans etcd • Assure l'authentification et effectue un contrôle des accès basé sur le rôle (RBAC) • Demande les validations et le contrôle des admissions • Pour en savoir plus : http://bit.ly/k8s-apiserver 	<p style="text-align: center;">Kubelet – Agent sur chaque nœud de travail</p> <ul style="list-style-type: none"> • Initie les pods (groupes d'un ou plusieurs conteneurs) à l'aide de PodSpec et assure l'exécution et le bon état de tous les pods • Interagit avec les conteneurs tels que Docker • Pour en savoir plus : http://bit.ly/k8s-kubelet
<p style="text-align: center;">Gestionnaire de contrôle</p> <ul style="list-style-type: none"> • Processus daemon qui implémente les boucles de contrôle intégrées dans Kubernetes : déploiements roulants, groupes de répliques (ReplicaSets), nombre de nœuds de travail, etc. • Pour en savoir plus : http://bit.ly/k8s-ctrl-mgr 	<p style="text-align: center;">Kube Proxy – Agent sur chaque nœud de travail</p> <ul style="list-style-type: none"> • Proxy réseau et dispositif d'équilibrage des charges pour les services Kubernetes • Pour en savoir plus : http://bit.ly/k8s-proxy
<p style="text-align: center;">Planificateur</p> <ul style="list-style-type: none"> • Décide où les pods doivent être exécutés en fonction de plusieurs facteurs : affinité, ressources disponibles, étiquettes QoS, etc. • Pour en savoir plus : http://bit.ly/k8s-scheduler 	
<p style="text-align: center;">etcd</p> <ul style="list-style-type: none"> • Le cœur d'un cluster Kubernetes ; maintient de façon persistante des informations clé-valeur sur tous les objets Kubernetes • Pour en savoir plus : http://bit.ly/2COKMx9 	

Voici les extensions Kubernetes requises pour la plupart des applications :

kube-dns	kubectl
<ul style="list-style-type: none"> • Provisionnées sous la forme d'un pod et d'un service sur Kubernetes • Chaque service reçoit une entrée DNS dans Kubernetes • kube-dns résout le DNS de tous les services du cluster 	<ul style="list-style-type: none"> • Ligne de commande officielle de Kubernetes • Utilise en coulisses des appels API REST à destination du serveur d'API Kubernetes

Constructions et objets Kubernetes essentiels :

Espaces de nommage	Pods
Segmentation virtuelle des différents clusters	Groupe logique d'un ou plusieurs conteneurs gérés par Kubernetes
Nœuds	Ensemble de répliques
Tissu d'infrastructure de Kubernetes (hôte de composants de travail et de composants maîtres)	Boucle continue garantissant l'exécution d'un nombre donné de pods
Rôles	Ingresses
Contrôle des accès basé sur le rôle du cluster Kubernetes	Gère le trafic HTTP externe vers le service hébergé
Déploiements	Services
Gère un ensemble de répliques, des définitions de pods, les mises à jour et d'autres concepts	Couche logique qui fournit la persistance des IP, DNS, etc. aux pods dynamiques

Défis de supervision dans les environnements Kubernetes

Selon la dernière étude de la CNCF, la complexité, la supervision et la sécurité font partie des grands défis des entreprises qui adoptent Kubernetes. Les stratégies de supervision d'antan ne fonctionnent pas dans l'ère du Cloud Native, principalement pour les raisons suivantes :

- Il y a trop de composants à superviser ;
- Les conteneurs sont éphémères et dynamiques ;
- Kubernetes planifie automatiquement les pods pour une utilisation optimale des ressources. S'il accroît sensiblement l'efficacité, il ajoute également de l'imprévisibilité quant au lieu de déploiement et d'exécution des pods, sauf si des intentions spécifiques ont été exprimées à l'aide d'affinités ou de taints.

Cela produit l'ensemble de conditions suivantes pour les environnements Kubernetes :

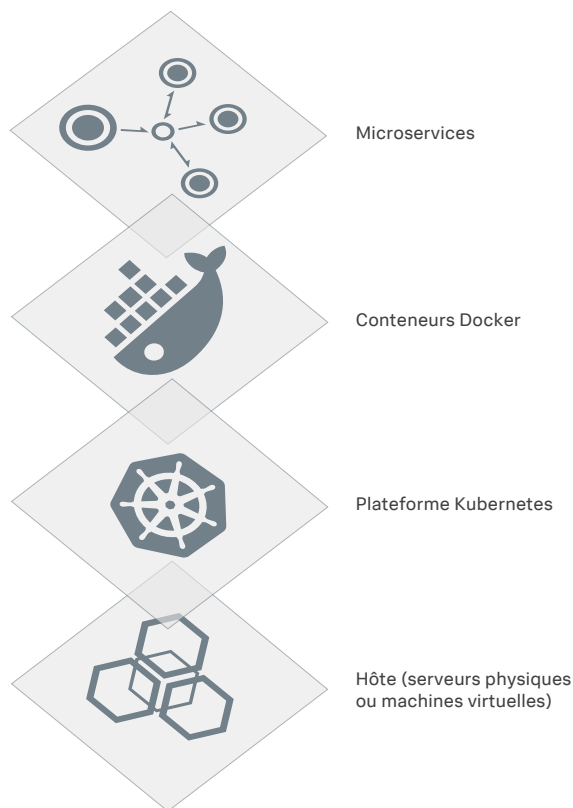
- Les composants à superviser sont beaucoup plus nombreux ;
- Les conteneurs sont éphémères et la localisation des pods est imprévisible ;
- Les microservices en conteneurs sont plus complexes à dépanner.

Un nombre accru de composants à superviser

Dans un monde monolithique, il n'y a que deux composants à superviser : les applications et les hôtes sur lesquels elles ont été déployées.

À l'ère du Cloud Native, les applications en conteneurs orchestrées par Kubernetes possèdent plusieurs composants qui doivent tous être supervisés :

- Les hôtes
- La plateforme Kubernetes elle-même
- Les conteneurs Docker
- Les microservices en conteneurs



Nature éphémère et localisation imprévisible des conteneurs

Contrairement au modèle traditionnel des hôtes aux exécutions longues, les applications modernes basées sur des microservices sont typiquement déployées sur des conteneurs dynamiques et éphémères. Kubernetes veille à ce que le nombre souhaité de répliques des applications soit exécuté. Kubernetes place des pods sur les nœuds qu'il considère appropriés, sauf instruction contraire donnée via une affinité ou une taint. En effet, laisser Kubernetes planifier les pods constitue l'objectif de conception clé de ce système qui s'ajuste automatiquement.

Les approches traditionnelles de supervision ne fonctionnent pas dans ces environnements hautement dynamiques car ils tendent à tracer les hôtes par leur nom ou leur adresse IP. Dans les environnements de conteneurs, les outils de supervision doivent effectuer une découverte immédiate des services et détecter automatiquement les événements de cycle de vie des conteneurs, tout en ajustant la collecte des métriques au démarrage et à l'arrêt des conteneurs, ce qui se produit en quelques secondes.

Superviser les performances des microservices

La localisation des problèmes dans un environnement de microservices est plus difficile que dans les environnements monolithiques, car les requêtes traversent à la fois plusieurs couches de la pile et plusieurs services. Les outils modernes de supervision doivent superviser ces couches interconnectées tout en corrélant efficacement le comportement des applications et de l'infrastructure pour harmoniser la résolution des problèmes.

Indicateurs clés de performance à superviser

Métriques des pods

Il est indispensable de superviser tous les objets Kubernetes pour garantir le bon état de santé de chaque cluster et l'optimisation de l'utilisation des ressources. La supervision des métriques des pods Kubernetes, des déploiements et des services permettra de déterminer si Kubernetes fonctionne comme prévu dans votre environnement.

PODS	Nombre de pods souhaités
	Nombre de pods disponibles
	Pods par phase (échec, en attente, en cours d'exécution)
	Pods souhaités par déploiement
	Pods souhaités par service
	Pods disponibles par déploiement
	Pods disponibles par service
	Pods souhaités par ReplicaSet
	Pods en cours d'exécution par nœud

Métriques d'utilisation des ressources

Il est également important de suivre l'utilisation des ressources pour veiller au bon état de santé de vos applications et clusters Kubernetes.

- Docker Socket fournit des métriques sur les conteneurs et sur l'utilisation des ressources au niveau de chaque nœud, comme l'utilisation de CPU et de mémoire. Kubernetes fournit des métriques collectées ainsi que des métriques émises par Kubernetes.
- Corréler les métriques de CPU, de mémoire, d'I/O disque et de performance du réseau avec les performances des applications et les événements Kubernetes permet de parvenir plus rapidement à la cause profonde d'un problème de performance.
- Superviser les événements Docker et Kubernetes, et notamment les événements de conteneurs ou de cycle de vie des pods, permet de localiser les problèmes de configuration et les manques de ressources.

Supervision de Kubernetes avec Splunk

Kubernetes fournit des informations détaillées sur ses composants et sur l'utilisation des ressources par les applications à l'échelle du cluster, du pod et du service. Ces informations sont faciles à recueillir à l'aide d'un large éventail de solutions de supervision, mais c'est aux utilisateurs finaux de rendre ces données exploitables.

Splunk vous permet de superviser Kubernetes à l'aide d'une instrumentation flexible et ouverte et de tableaux de bord prédéfinis qui donnent une visibilité immédiate sur les différentes couches de l'environnement. En combinant l'analyse en flux à la traçabilité distribuée, les utilisateurs de Splunk peuvent dépasser la simple collecte des données, pour exploiter des alertes en temps réel et des analyses de performances afin de localiser et résoudre les problèmes en quelques secondes.

Collecte des métriques d'un fournisseur cloud

Pour la supervision de base d'un service Kubernetes géré comme AWS Elastic Container Service for Kubernetes (EKS), Google Container Engine (GKE) ou Azure Kubernetes Service, le moyen le plus simple de collecter des métriques consiste à intégrer Splunk à des services comme AWS CloudWatch, Google Stackdriver, et Azure Monitor.

Cette approche est la plus simple, et elle permet à Splunk de recueillir des métriques Kubernetes sans avoir à installer un agent ni à modifier le code des applications. Toutefois, ces services sont configurés par défaut pour agréger et rapporter des métriques à des intervalles relativement espacés (AWS CloudWatch se met à jour toutes les 5 minutes par défaut) et ne fournissent aucun renseignement sur les services spécifiques déployés sur vos clusters Kubernetes.

Supervision de Kubernetes entièrement automatisée

Pour plus d'informations sur les services et une supervision plus fine des métriques des conteneurs, nous recommandons d'installer le Smart Agent sur tous vos clusters Kubernetes.

Agent de collecte de métriques open source reposant sur collectd, le Smart Agent effectue une découverte automatique des services et se configure pour superviser le contenu. L'un des avantages de cette approche réside dans la capacité du Smart Agent à envoyer des métriques à Splunk avec une résolution d'une seconde, ce qui la rend idéale étant donné la nature éphémère des pods Kubernetes.

Le Smart Agent s'exécute comme un DaemonSet dans Kubernetes pour assurer sa présence sur tous les nœuds du cluster. Il collecte les données de Kubernetes et utilise cAdvisor pour obtenir les caractéristiques de ressources et de performance des conteneurs Docker en cours d'exécution. Cette configuration sans contact avec découverte automatique des composants Kubernetes et des services en conteneurs supervise instantanément toute la pile. Smart Agent s'installe en une étape simple :

```
helm install splunk --set splunkAccessToken=<
YOUR_ACCESS_TOKEN> --set
clusterName=<YOUR_CLUSTER_NAME> splunk/
splunk-agent
```

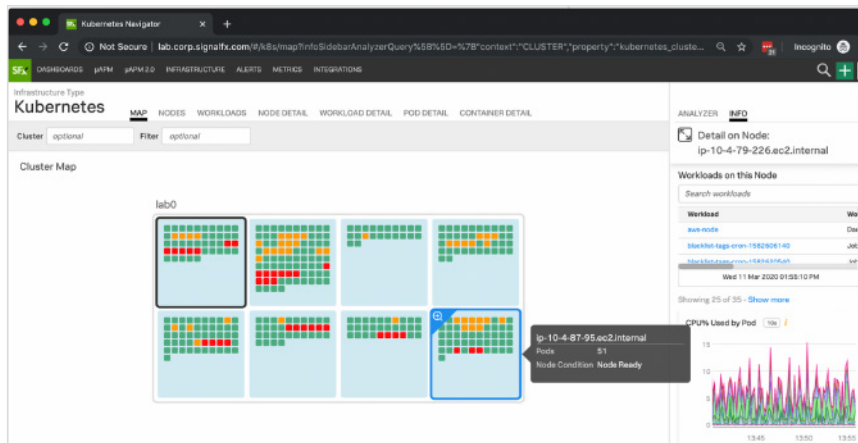
Non seulement le Smart Agent supervise Kubernetes en amont, mais il est également capable de collecter des métriques auprès des services Kubernetes gérés comme AWS Elastic Container Service for Kubernetes (EKS) et Google Container Engine (GKE), ainsi que des plateformes telles qu'OpenShift et Pivotal Kubernetes Service (PKS).

Le Smart Agent fournit également des moniteurs qui collectent les données d'autres protocoles de métriques comme Prometheus et StatsD, permettant ainsi aux équipes qui exploitent des pipelines de métriques existantes de profiter de Splunk.

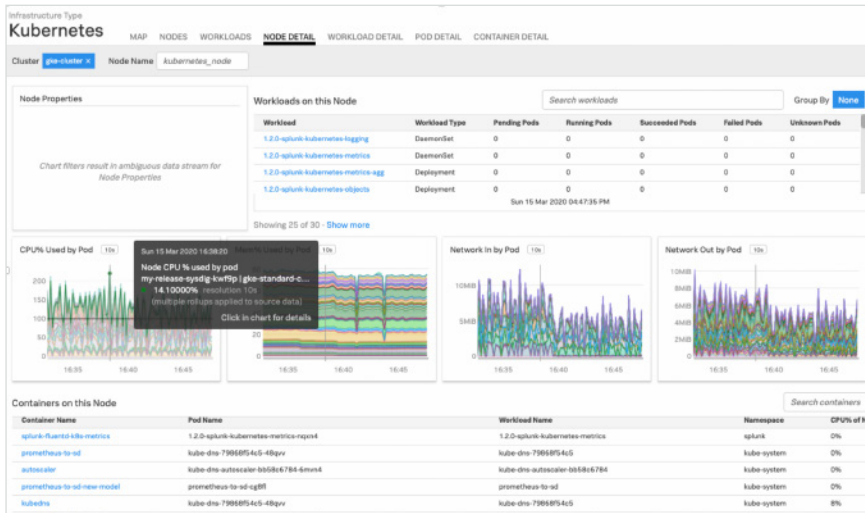
Naviguer dans votre environnement Kubernetes

Tableaux de bord Kubernetes prédéfinis

Offrant comme point de départ une vue d'ensemble, Kubernetes Navigator permet aux équipes de comprendre les performances de l'ensemble de l'environnement Kubernetes avec une navigation intuitive et hiérarchique. Sélectionnez, filtrez ou recherchez n'importe quelle entité Kubernetes (nœud, pod, conteneur) et explorez-la pour procéder à une analyse détaillée en quelques secondes. Comprenez les relations entre les composants Kubernetes dynamiques et corrigez rapidement les problèmes de performance interdépendants provenant de voisins bruyants.



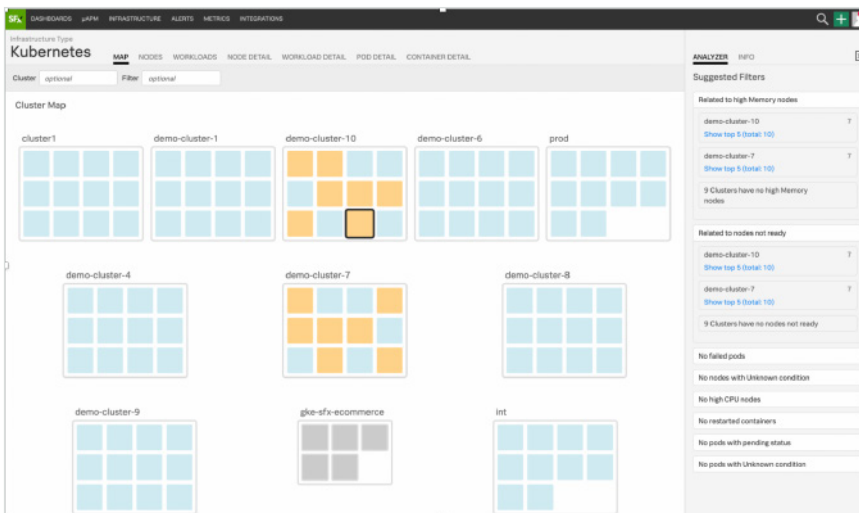
L'exploration d'un nœud spécifique fournit des métriques système pour ce nœud, mais aussi des tableaux de bord pour les services qui sont exécutés sur ce nœud.



Vous pouvez également voir tous les pods en cours d'exécution dans votre cluster Kubernetes et suivre l'activité d'un pod particulier ou de tous les pods de votre cluster. À partir de là, vous pouvez ensuite explorer les conteneurs Docker de chaque pod.

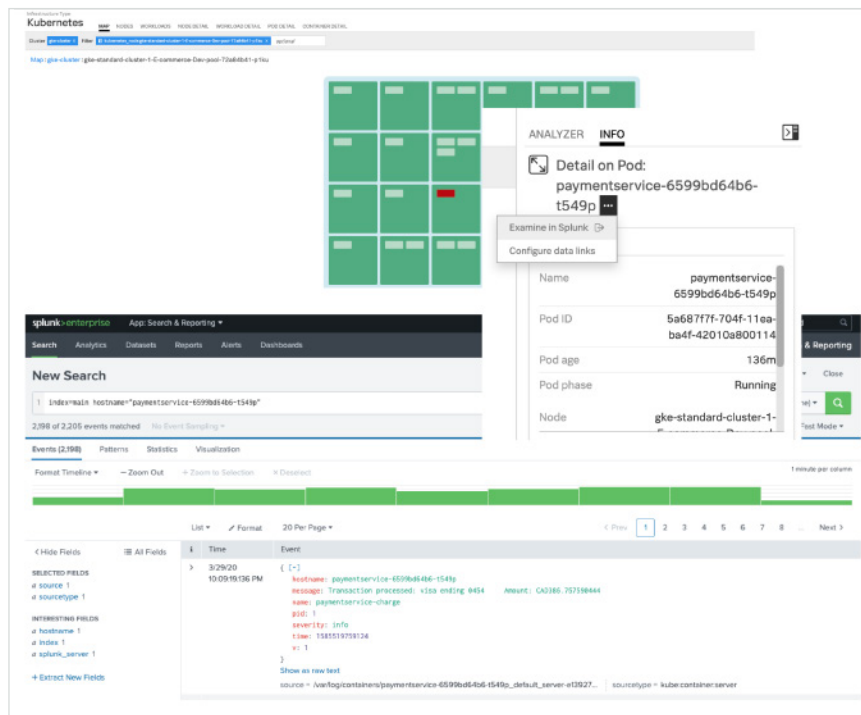
Analyseur Kubernetes

Cette capacité d'analyse basée sur une IA met automatiquement au jour des renseignements et des recommandations afin de révéler en temps réel ce qui cause les anomalies sur l'ensemble du cluster Kubernetes : nœuds, conteneurs et applicatifs. Des algorithmes sophistiqués, comme la référence des performances historiques et le changement soudain, détectent les problèmes de niveau système tels qu'une augmentation subite des routines ou des redémarrages des clusters, et produisent une alerte en quelques secondes.



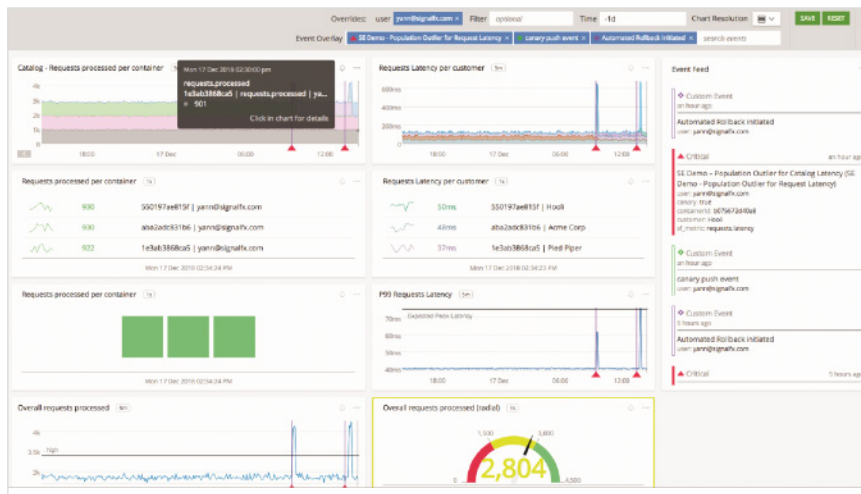
Logs contextualisés

Accédez en toute simplicité aux logs et obtenez une visibilité granulaire sur les logs des applications, de Kubernetes et des conteneurs pour corrélérer les performances de toute la pile, sans changer de contexte. La visibilité sur les événements du cycle de vie de Kubernetes et les logs d'API Server Audit vous aide à comprendre et maintenir vos postures de sécurité et de conformité.



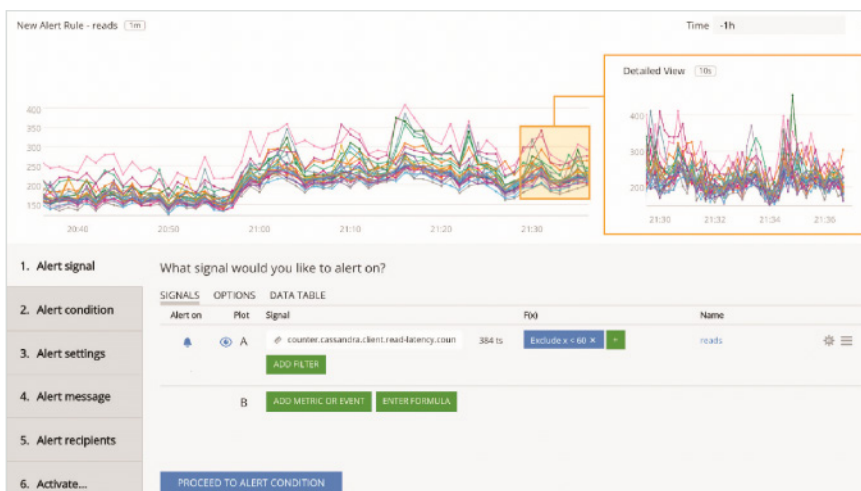
Créer de tableaux de bord de services personnalisés

Splunkx vous permet d'afficher les métriques des conteneurs aux côtés d'autres indicateurs de performance pour obtenir de la visibilité sur toutes les couches de votre environnement. Par exemple, un responsable de service qui souhaite superviser des déploiements canaris peut créer le tableau de bord suivant, en affichant les métriques des conteneurs à côté de graphiques mesurant la latence des requêtes et d'un flux d'événements traçant les poussées de code, les alertes et les mesures de correction prises.



Détecter des problèmes en temps réel

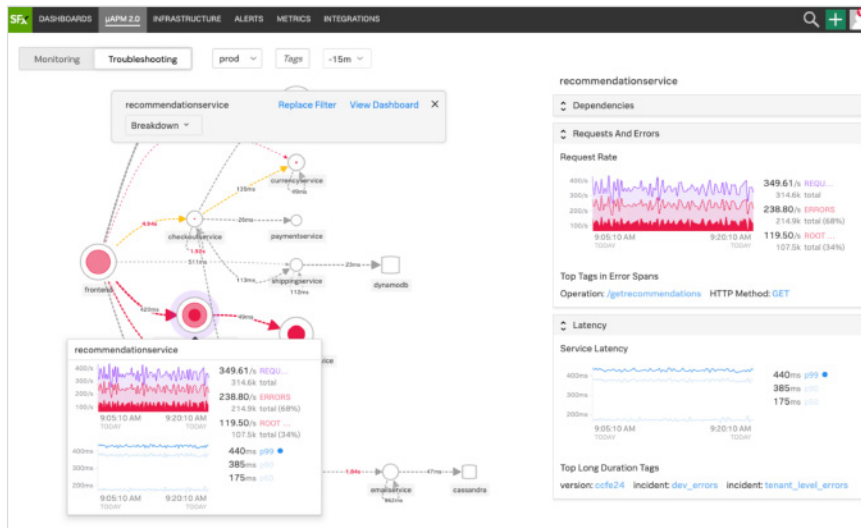
Avec Splunk, la mise en place d'alertes n'a pas besoin d'être un compromis entre précision et pertinence. Générez des alertes proactives en combinant des métriques haute résolution provenant de vos clusters Kubernetes avec une bibliothèque de fonctions statistiques et d'algorithmes dans Splunk. Prévisualisez les alertes sur des données historiques afin de les affiner avant le déploiement, et appliquez les méthodes de la science des données aux alertes pour éviter les faux positifs et les déluges d'alertes, courants dans les environnements hautement éphémères.



Exploiter la traçabilité distribuée pour guider la résolution des problèmes

Splunk APM fournit aux utilisateurs des capacités de traçabilité distribuée permettant une exploration approfondie des événements impactant les performances. Quel que soit l'endroit de votre environnement Kubernetes où le problème surgit, vous pouvez passer directement de l'alerte temps réel aux traces d'applications et corrélérer les tendances de performance entre les infrastructures, Kubernetes et vos microservices. La solution Splunk APM est bâtie sur une **architecture NoSample™** Full-Fidelity qui importe et analyse toutes les traces, de façon à ce que les valeurs anormales et les anomalies ne soient jamais ignorées.

Outlier Analyzer™ guide la résolution des problèmes en mettant au jour les tendances les plus courantes dans les traces anormales. Il vous permet ainsi de circonscrire rapidement les nœuds, le cluster, la région de cloud ou les étiquettes et balises propres à une application qui présentent une anomalie.



Des DevOps plus rapides

Une fois que les données de Kubernetes sont importées dans Splunk, les équipes DevOps peuvent plus facilement appliquer les bonnes pratiques et solutions de supervision aux problèmes courants de toute l'organisation. Splunk offre une visibilité instantanée sur les hôtes, les conteneurs, les pods et les objets Kubernetes grâce à des tableaux de bord prédéfinis et prêts à l'emploi, tout en offrant aux équipes la possibilité de personnaliser des tableaux de bord, des alertes et des notifications selon leurs propres besoins.

Les équipes centrales peuvent implémenter la supervision dans le code en utilisant l'API Splunk pour créer, de façon programmatique, du contenu de supervision et définir des analyses, puis exploiter le Splunk Service Bureau pour contrôler l'utilisation, l'accès et les permissions des utilisateurs et des équipes.

```
curl \ --request post \ --header "x-sf-token: your_access_token" \ --header "content-type: application/json" \ --data '{"name": "cpu load", "programtext": "data(`cpu.load`).publish()"}' \ https://api.splunk.com/v2/chart
```

Les plus grandes entreprises font confiance à Splunk

Des entreprises de tous les secteurs et du monde entier utilisent Splunk pour accélérer en toute confiance leur passage à une approche Cloud Native et l'adoption de Kubernetes dans leurs organisations. Voici une petite sélection de ces clients.

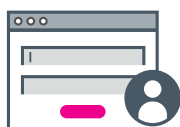
Pour en savoir plus : https://www.splunk.com/en_us/customers.html



Commencez dès aujourd'hui avec Splunk Infrastructure Monitoring

Prêt à découvrir comment Splunk peut accélérer l'adoption de Kubernetes dans votre entreprise ?

Nos responsables de la réussite client et nos ingénieurs solutions travailleront avec vos équipes pour accélérer l'adoption de Kubernetes et mettre en place la supervision de Kubernetes en temps réel. Votre essai gratuit de 14 jours ici : <https://www.observability.splunk.com>.



Inscrivez-vous pour effectuer un essai gratuit

[Commencez ici](#)



Déployez Smart Agent

[En savoir plus](#)



Lancez la supervision des environnements Kubernetes

[En savoir plus](#)

Pour en savoir plus sur le Kubernetes Navigator pour la supervision en temps réel et le dépannage des environnements Kubernetes, quelles que soient leurs dimensions, téléchargez la [Fiche technique de Kubernetes Navigator](#).