Plate-forme de conteneurs OpenShift 4.12

Enregistrement

Notes d'installation, d'utilisation et de mise à jour d'OpenShift Logging

Last Updated: 2023-04-27

Plate-forme de conteneurs OpenShift 4.12 Enregistrement

Notes d'installation, d'utilisation et de mise à jour d'OpenShift Logging

Notice légale

Résumé

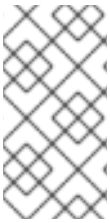
Ce document fournit des instructions pour l'installation, la configuration et l'utilisation d'OpenShift Logging, qui regroupe les logs pour une gamme de services OpenShift Container Platform.

Table des matières

CHAPITRE 1. NOTES DE MISE À JOUR POUR LA JOURNALISATION

NOTE

Le sous-système de journalisation pour Red Hat OpenShift est fourni en tant que composant installable, avec un cycle de publication distinct de celui de la plateforme principale OpenShift Container Platform. La politique de cycle de vie de Red Hat OpenShift Container Platform décrit la compatibilité des versions.



NOTE

Le canal **stable** ne fournit des mises à jour que pour la version la plus récente du logiciel d'exploitation. Pour continuer à recevoir les mises à jour des versions antérieures, vous devez changer votre canal d'abonnement pour **stable-X**, où **X** est la version de l'exploitation que vous avez installée.

1.1. JOURNALISATION 5.6.4

Cette version inclut la version 5.6.4 de la correction des bugs de journalisation d'OpenShift .

1.1.1. Bug fixes

- Avant cette mise à jour, lorsque LokiStack était déployé comme magasin de logs, les logs générés par les pods Loki étaient collectés et envoyés à LokiStack. Avec cette mise à jour, les logs générés par Loki sont exclus de la collecte et ne seront pas stockés.(LOG-3280)

- Avant cette mise à jour, lorsque l'éditeur de requêtes de la page Logs de l'OpenShift Web Console était vide, les menus déroulants ne s'affichaient pas. Avec cette mise à jour, si une requête vide est tentée, un message d'erreur s'affiche et les menus déroulants se remplissent maintenant comme prévu.(LOG-3454)

- Avant cette mise à jour, lorsque l'option **tls.insecureSkipVerify** était définie sur **true**, l'opérateur de journalisation de cluster générait une configuration incorrecte. En conséquence, l'opérateur n'envoyait pas de données à Elasticsearch lorsqu'il tentait d'ignorer la validation du certificat. Avec cette mise à jour, le Cluster Logging Operator génère une configuration TLS correcte même lorsque **tls.insecureSkipVerify** est activé. Par conséquent, les données peuvent être envoyées avec succès à Elasticsearch même lorsque l'on tente d'ignorer la validation du certificat.(LOG-3475)

- Avant cette mise à jour, lorsque l'analyse structurée était activée et que les messages étaient transmis à plusieurs destinations, ils n'étaient pas copiés en profondeur. Par conséquent, certains des journaux reçus incluaient le message structuré, tandis que d'autres ne le faisaient pas. Avec cette mise à jour, la génération de configuration a été modifiée pour copier en profondeur les messages avant l'analyse JSON. Par conséquent, tous les messages reçus contiennent désormais des messages structurés, même lorsqu'ils sont transmis à plusieurs destinations.(LOG-3640)

- Avant cette mise à jour, si le champ **collection** contenait **{}**, l'opérateur pouvait se bloquer. Avec cette mise à jour, l'opérateur ignorera cette valeur, ce qui lui permettra de continuer à fonctionner sans interruption.(LOG-3733)

- Avant cette mise à jour, l'attribut **nodeSelector** pour le composant Gateway de LokiStack n'avait aucun effet. Avec cette mise à jour, l'attribut **nodeSelector** fonctionne comme prévu.(LOG-3783)

- Avant cette mise à jour, la configuration statique de la liste des membres de LokiStack reposait uniquement sur des réseaux IP privés. Par conséquent, lorsque le réseau de pods du cluster OpenShift Container Platform était configuré avec une plage d'IP publique, les pods LokiStack se bloquaient. Avec cette mise à jour, l'administrateur de LokiStack a maintenant la possibilité d'utiliser le réseau de pods pour la configuration de la liste des membres. Cela résout le problème et empêche les pods LokiStack d'entrer dans un état de crashloop lorsque le réseau de pods du cluster OpenShift Container Platform est configuré avec une plage d'IP publique.(LOG-3814)

- Avant cette mise à jour, si le champ **tls.insecureSkipVerify** était défini sur **true**, l'opérateur de journalisation de cluster générait une configuration incorrecte. Par conséquent, l'opérateur n'envoyait pas de données à Elasticsearch lorsqu'il tentait d'ignorer la validation du certificat. Avec cette mise à jour, l'opérateur génère une configuration TLS correcte même lorsque **tls.insecureSkipVerify** est activé. Par conséquent, les données peuvent être envoyées avec succès à Elasticsearch même lorsque l'on tente d'ignorer la validation du certificat.(LOG-3838)

- Avant cette mise à jour, si le Cluster Logging Operator (CLO) était installé sans l'Elasticsearch Operator, le pod CLO affichait en permanence un message d'erreur lié à la suppression d'Elasticsearch. Avec cette mise à jour, le CLO effectue désormais des vérifications supplémentaires avant d'afficher des messages d'erreur. Par conséquent, les messages d'erreur liés à la suppression d'Elasticsearch ne sont plus affichés en l'absence de l'opérateur Elasticsearch.(LOG-3763)

1.1.2. CVE

- CVE-2022-4304
- CVE-2022-4450
- CVE-2023-0215
- CVE-2023-0286
- CVE-2023-0767
- CVE-2023-23916

1.2. JOURNALISATION 5.6.3

Cette version inclut la version 5.6.3 de la correction des bugs de journalisation d'OpenShift .

1.2.1. Bug fixes

- Avant cette mise à jour, l'opérateur stockait les informations relatives au secret du locataire de la passerelle dans une carte de configuration. Avec cette mise à jour, l'opérateur stocke ces informations dans un secret.(LOG-3717)

- Avant cette mise à jour, le collecteur Fluentd ne capturait pas les événements de connexion OAuth stockés dans **/var/log/auth-server/audit.log**. Avec cette mise à jour, Fluentd capture ces événements de connexion OAuth, ce qui résout le problème.(LOG-3729)

1.2.2. CVE

- CVE-2020-10735
- CVE-2021-28861
- CVE-2022-2873
- CVE-2022-4415
- CVE-2022-40897
- CVE-2022-41222
- CVE-2022-43945
- CVE-2022-45061
- CVE-2022-48303

1.3. JOURNALISATION 5.6.2

Cette version inclut la version 5.6.2 de la correction des bugs de journalisation d'OpenShift .

1.3.1. Bug fixes

- Avant cette mise à jour, le collecteur ne définissait pas correctement les champs **level** en fonction de la priorité des journaux systemd. Avec cette mise à jour, les champs **level** sont définis correctement.(LOG-3429)
- Avant cette mise à jour, l'Opérateur générait incorrectement des avertissements d'incompatibilité sur OpenShift Container Platform 4.12 ou plus récent. Avec cette mise à jour, la valeur de la version max OpenShift Container Platform de l'Opérateur a été corrigée, ce qui résout le problème.(LOG-3584)
- Avant cette mise à jour, la création d'une ressource personnalisée (CR) **ClusterLogForwarder** avec une valeur de sortie de **default** ne générait aucune erreur. Avec cette mise à jour, un avertissement d'erreur indiquant que cette valeur n'est pas valide est généré de manière appropriée.(LOG-3437)
- Avant cette mise à jour, lorsque la ressource personnalisée (CR) **ClusterLogForwarder** avait plusieurs pipelines configurés avec une sortie définie comme **default**, les pods collecteurs redémarraient. Avec cette mise à jour, la logique de validation des sorties a été corrigée, ce qui résout le problème.(LOG-3559)
- Avant cette mise à jour, les pods collecteurs redémarraient après avoir été créés. Avec cette mise à jour, le collecteur déployé ne redémarre pas de lui-même.(LOG-3608)
- Avant cette mise à jour, les versions des correctifs supprimaient les versions précédentes des opérateurs du catalogue. Cela rendait l'installation des anciennes versions impossible. Cette mise à jour modifie les configurations des paquets de sorte que les versions précédentes de la même version mineure restent dans le catalogue.(LOG-3635)

1.3.2. CVE

- CVE-2022-23521

- CVE-2022-40303
- CVE-2022-40304
- CVE-2022-41903
- CVE-2022-47629
- CVE-2023-21835
- CVE-2023-21843

1.4. JOURNALISATION 5.6.1

Cette version inclut la version 5.6.1 de la correction des bugs de journalisation d'OpenShift .

1.4.1. Bug fixes

- Avant cette mise à jour, le compacteur signalait des erreurs de certificat TLS lors des communications avec l'interrogateur lorsque la rétention était active. Avec cette mise à jour, le compacteur et l'interrogateur ne communiquent plus de manière erronée via HTTP.(LOG-3494)
- Avant cette mise à jour, l'opérateur Loki ne réessayait pas de définir l'état de **LokiStack** CR, ce qui entraînait des informations d'état périmées. Avec cette mise à jour, l'opérateur réessaie de mettre à jour les informations d'état en cas de conflit.(LOG-3496)
- Avant cette mise à jour, le serveur Webhook de l'opérateur Loki provoquait des erreurs TLS lorsque l'opérateur **kube-apiserver-operator** vérifiait la validité du webhook. Avec cette mise à jour, l'ICP du webhook de l'opérateur Loki est gérée par le gestionnaire du cycle de vie de l'opérateur (OLM), ce qui résout le problème.(LOG-3510)
- Avant cette mise à jour, le LokiStack Gateway Labels Enforcer générait des erreurs d'analyse pour les requêtes LogQL valides lors de l'utilisation de filtres d'étiquettes combinés avec des expressions booléennes. Avec cette mise à jour, l'implémentation LogQL de LokiStack prend en charge les filtres d'étiquettes avec des expressions booléennes et résout le problème.(LOG-3441),(LOG-3397)
- Avant cette mise à jour, les enregistrements écrits dans Elasticsearch échouaient si plusieurs clés d'étiquettes avaient le même préfixe et si certaines clés comportaient des points. Avec cette mise à jour, les traits de soulignement remplacent les points dans les clés d'étiquettes, ce qui résout le problème.(LOG-3463)
- Avant cette mise à jour, l'opérateur **Red Hat OpenShift Logging** n'était pas disponible pour les clusters OpenShift Container Platform 4.10 en raison d'une incompatibilité entre la console OpenShift Container Platform et le plugin logging-view. Avec cette mise à jour, le plugin est correctement intégré à la console d'administration d'OpenShift Container Platform 4.10.(LOG-3447)
- Avant cette mise à jour, la réconciliation de la ressource personnalisée **ClusterLogForwarder** signalait de manière incorrecte un état dégradé des pipelines qui font référence au logstore par défaut. Avec cette mise à jour, le pipeline est validé correctement.(LOG-3477)

1.4.2. CVE

- CVE-2021-46848

- CVE-2022-3821
- CVE-2022-35737
- CVE-2022-42010
- CVE-2022-42011
- CVE-2022-42012
- CVE-2022-42898
- CVE-2022-43680
- CVE-2021-35065
- CVE-2022-46175

1.5. ENREGISTREMENT 5.6

Cette version inclut la version 5.6 d'OpenShift Logging.

1.5.1. Avis de dépréciation

Dans Logging 5.6, Fluentd est obsolète et il est prévu de le supprimer dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Comme alternative à fluentd, vous pouvez utiliser Vector.

1.5.2. Améliorations

- Avec cette mise à jour, la journalisation est conforme aux politiques cryptographiques à l'échelle du cluster d'OpenShift Container Platform.(LOG-895)
- Avec cette mise à jour, vous pouvez déclarer des politiques de rétention par locataire, par flux et globales via la ressource personnalisée LokiStack, classées par ordre de priorité.(LOG-2695)
- Avec cette mise à jour, Splunk est une option de sortie disponible pour le transfert de logs.(LOG-2913)
- Avec cette mise à jour, Vector remplace Fluentd comme collecteur par défaut.(LOG-2222)
- Avec cette mise à jour, le rôle **Developer** peut accéder aux journaux de charge de travail par projet auxquels ils sont affectés dans le plugin Log Console sur les clusters exécutant OpenShift Container Platform 4.11 et plus.(LOG-3388)
- Avec cette mise à jour, les logs de n'importe quelle source contiennent un champ **openshift.cluster_id**, l'identifiant unique du cluster dans lequel l'Opérateur est déployé. Vous pouvez visualiser la valeur de **clusterID** à l'aide de la commande ci-dessous. (LOG-2715)

```
$ oc get clusterversion/version -o jsonpath='{.spec.clusterID}{"\n"}'
```

1.5.3. Problèmes connus

- Avant cette mise à jour, Elasticsearch rejetait les journaux si plusieurs clés de label avaient le

même préfixe et si certaines clés incluaient le caractère **.**. Cette mise à jour corrige la limitation d'Elasticsearch en remplaçant **.** dans les clés d'étiquettes par **_**. Pour contourner ce problème, supprimez les étiquettes qui provoquent des erreurs ou ajoutez un espace de noms à l'étiquette.(LOG-3463)

1.5.4. Bug fixes

- Avant cette mise à jour, si vous supprimiez la ressource personnalisée Kibana, la console web de OpenShift Container Platform continuait à afficher un lien vers Kibana. Avec cette mise à jour, la suppression de la ressource personnalisée Kibana supprime également ce lien.(LOG-2993)

- Avant cette mise à jour, un utilisateur n'était pas en mesure de voir les journaux d'application des espaces de noms auxquels il avait accès. Avec cette mise à jour, l'opérateur Loki crée automatiquement un rôle de cluster et un lien de rôle de cluster permettant aux utilisateurs de lire les journaux d'application.(LOG-3072)

- Avant cette mise à jour, l'opérateur supprimait toutes les sorties personnalisées définies dans la ressource personnalisée **ClusterLogForwarder** lorsqu'il utilisait LokiStack comme stockage de logs par défaut. Avec cette mise à jour, l'opérateur fusionne les sorties personnalisées avec les sorties par défaut lors du traitement de la ressource personnalisée **ClusterLogForwarder**.(LOG-3090)

- Avant cette mise à jour, la clé de l'autorité de certification était utilisée comme nom de volume pour le montage de l'autorité de certification dans Loki, ce qui provoquait des états d'erreur lorsque la clé de l'autorité de certification comprenait des caractères non conformes, tels que des points. Avec cette mise à jour, le nom de volume est normalisé à une chaîne interne, ce qui résout le problème.(LOG-3331)

- Avant cette mise à jour, une valeur par défaut définie dans la définition des ressources personnalisées de LokiStack entraînait l'impossibilité de créer une instance de LokiStack sans **ReplicationFactor** de **1**. Avec cette mise à jour, l'opérateur définit la valeur réelle de la taille utilisée.(LOG-3296)

- Avant cette mise à jour, Vector analysait le champ message lorsque l'analyse JSON était activée sans définir les valeurs **structuredTypeKey** ou **structuredTypeName**. Avec cette mise à jour, une valeur est requise pour **structuredTypeKey** ou **structuredTypeName** lors de l'écriture de journaux structurés dans Elasticsearch.(LOG-3195)

- Avant cette mise à jour, le composant de création de secret de l'Elasticsearch Operator modifiait constamment les secrets internes. Avec cette mise à jour, le secret existant est correctement géré.(LOG-3161)

- Avant cette mise à jour, l'opérateur pouvait entrer dans une boucle de suppression et de recréation du daemonset du collecteur pendant que les déploiements Elasticsearch ou Kibana changeaient d'état. Avec cette mise à jour, une correction dans la gestion de l'état de l'opérateur résout le problème.(LOG-3157)

- Avant cette mise à jour, Kibana avait un délai d'expiration du cookie OAuth fixe **24h**, ce qui entraînait des erreurs 401 dans Kibana chaque fois que le champ **accessTokenInactivityTimeout** était défini sur une valeur inférieure à **24h**. Avec cette mise à jour, le délai d'expiration du cookie OAuth de Kibana se synchronise sur le champ **accessTokenInactivityTimeout**, avec une valeur par défaut de **24h**.(LOG-3129)

- Avant cette mise à jour, le modèle général des opérateurs pour le rapprochement des ressources consistait à essayer de créer un objet avant d'essayer de l'obtenir ou de le mettre à jour, ce qui entraînait des réponses HTTP 409 constantes après la création. Avec cette mise à

jour, les opérateurs tentent d'abord de récupérer un objet et ne le créent ou ne le mettent à jour que s'il est manquant ou différent de ce qui a été spécifié.(LOG-2919)

- Avant cette mise à jour, les champs **.level** et `.structure.level` de Fluentd pouvaient contenir des valeurs différentes. Avec cette mise à jour, les valeurs sont les mêmes pour chaque champ.(LOG-2819)

- Avant cette mise à jour, l'opérateur n'attendait pas que le groupe d'autorités de certification de confiance soit peuplé et déployait le collecteur une deuxième fois une fois le groupe mis à jour. Avec cette mise à jour, l'opérateur attend brièvement de voir si le groupe a été peuplé avant de poursuivre le déploiement du collecteur.(LOG-2789)

- Avant cette mise à jour, les informations de télémétrie d'enregistrement apparaissaient deux fois lors de l'examen des métriques. Avec cette mise à jour, les informations de télémétrie s'affichent comme prévu.(LOG-2315)

- Avant cette mise à jour, les logs de Fluentd pod contenaient un message d'avertissement après avoir activé l'ajout de l'analyse JSON. Avec cette mise à jour, ce message d'avertissement n'apparaît plus.(LOG-1806)

- Avant cette mise à jour, le script **must-gather** ne s'exécutait pas car **oc** a besoin d'un dossier avec des droits d'écriture pour construire son cache. Avec cette mise à jour, **oc** a les droits d'écriture sur un dossier et le script **must-gather** s'exécute correctement. (LOG-3446)

- Avant cette mise à jour, le SCC du collecteur de journaux pouvait être remplacé par d'autres SCC sur le cluster, rendant le collecteur inutilisable. Cette mise à jour définit la priorité du SCC du collecteur de journaux de manière à ce qu'il soit prioritaire sur les autres.(LOG-3235)

- Avant cette mise à jour, il manquait à Vector le champ **sequence**, qui a été ajouté à fluentd pour pallier le manque de précision des nanosecondes. Avec cette mise à jour, le champ **openshift.sequence** a été ajouté aux journaux d'événements. (LOG-3106)

1.5.5. CVE

- CVE-2020-36518
- CVE-2021-46848
- CVE-2022-2879
- CVE-2022-2880
- CVE-2022-27664
- CVE-2022-32190
- CVE-2022-35737
- CVE-2022-37601
- CVE-2022-41715
- CVE-2022-42003
- CVE-2022-42004
- CVE-2022-42010

- CVE-2022-42011
- CVE-2022-42012
- CVE-2022-42898
- CVE-2022-43680

1.6. JOURNALISATION 5.5.10

Cette version inclut la version 5.5.10 de la correction des bugs de journalisation d'OpenShift .

1.6.1. Bug fixes

- Avant cette mise à jour, le plugin logging view de l'OpenShift Web Console n'affichait qu'un texte d'erreur lorsque la LokiStack n'était pas joignable. Après cette mise à jour, le plugin affiche un message d'erreur approprié avec des détails sur la façon de réparer la LokiStack inaccessible.(LOG-2874)

1.6.2. CVE

- CVE-2022-4304
- CVE-2022-4450
- CVE-2023-0215
- CVE-2023-0286
- CVE-2023-0361
- CVE-2023-23916

1.7. JOURNALISATION 5.5.9

Cette version inclut la version 5.5.9 de la correction des bugs de journalisation d'OpenShift .

1.7.1. Bug fixes

- Avant cette mise à jour, un problème avec le collecteur Fluentd faisait qu'il ne capturait pas les événements de connexion OAuth stockés dans **/var/log/auth-server/audit.log**. Cela conduisait à une collecte incomplète des événements de connexion du service OAuth. Avec cette mise à jour, le collecteur Fluentd résout maintenant ce problème en capturant tous les événements de connexion du service OAuth, y compris ceux stockés dans **/var/log/auth-server/audit.log**, comme prévu.(LOG-3730)

- Avant cette mise à jour, lorsque l'analyse structurée était activée et que les messages étaient transmis à plusieurs destinations, ils n'étaient pas copiés en profondeur. Par conséquent, certains des journaux reçus incluaient le message structuré, tandis que d'autres ne le faisaient pas. Avec cette mise à jour, la génération de configuration a été modifiée pour copier en profondeur les messages avant l'analyse JSON. Par conséquent, tous les journaux reçus contiennent désormais des messages structurés, même lorsqu'ils sont transmis à plusieurs destinations.(LOG-3767)

1.7.2. CVE

- CVE-2022-4304
- CVE-2022-4450
- CVE-2022-41717
- CVE-2023-0215
- CVE-2023-0286
- CVE-2023-0767
- CVE-2023-23916

1.8. JOURNALISATION 5.5.8

Cette version inclut la version 5.5.8 de la correction des bugs de journalisation d'OpenShift .

1.8.1. Bug fixes

- Avant cette mise à jour, le champ **priority** était absent des journaux **systemd** en raison d'une erreur dans la manière dont le collecteur définissait les champs **level**. Avec cette mise à jour, ces champs sont définis correctement, ce qui résout le problème.(LOG-3630)

1.8.2. CVE

- CVE-2020-10735
- CVE-2021-28861
- CVE-2022-2873
- CVE-2022-4415
- CVE-2022-24999
- CVE-2022-40897
- CVE-2022-41222
- CVE-2022-41717
- CVE-2022-43945
- CVE-2022-45061
- CVE-2022-48303

1.9. JOURNALISATION 5.5.7

Cette version inclut la version 5.5.7 de la correction des bugs de journalisation d'OpenShift .

1.9.1. Bug fixes

- Avant cette mise à jour, le LokiStack Gateway Labels Enforcer générait des erreurs d'analyse pour les requêtes LogQL valides lors de l'utilisation de filtres d'étiquettes combinés avec des expressions booléennes. Avec cette mise à jour, l'implémentation LogQL de LokiStack prend en charge les filtres d'étiquettes avec des expressions booléennes et résout le problème.(LOG-3534)

- Avant cette mise à jour, la ressource personnalisée (CR) **ClusterLogForwarder** ne transmettait pas les informations d'identification TLS pour la sortie syslog à Fluentd, ce qui entraînait des erreurs lors de la transmission. Avec cette mise à jour, les informations d'identification sont correctement transmises à Fluentd, ce qui résout le problème.(LOG-3533)

1.9.2. CVE

CVE-2021-46848CVE-2022-3821CVE-2022-35737CVE-2022-42010CVE-2022-42011CVE-2022-42012CVE-2022-42898CVE-2022-43680

1.10. JOURNALISATION 5.5.6

Cette version inclut la version 5.5.6 de la correction des bugs de journalisation d'OpenShift .

1.10.1. Bug fixes

- Avant cette mise à jour, le contrôleur d'admission Pod Security a ajouté le label **podSecurityLabelSync = true** à l'espace de noms **openshift-logging**. Les étiquettes de sécurité que nous avions spécifiées étaient donc écrasées et les pods Collector ne démarraient pas. Avec cette mise à jour, l'étiquette **podSecurityLabelSync = false** préserve les étiquettes de sécurité. Les pods du collecteur se déploient comme prévu.(LOG-3340)

- Avant cette mise à jour, l'opérateur installait le plugin d'affichage de la console, même s'il n'était pas activé sur le cluster. Cela provoquait le plantage de l'opérateur. Avec cette mise à jour, si un compte pour un cluster n'a pas la vue console activée, l'Opérateur fonctionne normalement et n'installe pas la vue console.(LOG-3407)

- Avant cette mise à jour, une correction antérieure visant à prendre en charge une régression dans laquelle le statut du déploiement d'Elasticsearch n'était pas mis à jour entraînait un plantage de l'opérateur à moins que le site **Red Hat Elasticsearch Operator** ne soit déployé. Avec cette mise à jour, cette correction a été annulée de sorte que l'opérateur est maintenant stable mais réintroduit le problème précédent lié à l'état rapporté.(LOG-3428)

- Avant cette mise à jour, l'Opérateur Loki ne déployait qu'une seule réplique de la passerelle LokiStack quelle que soit la taille de la pile choisie. Avec cette mise à jour, le nombre de répliques est correctement configuré en fonction de la taille choisie.(LOG-3478)

- Avant cette mise à jour, les enregistrements écrits dans Elasticsearch échouaient si plusieurs clés d'étiquettes avaient le même préfixe et si certaines clés comportaient des points. Avec cette mise à jour, les traits de soulignement remplacent les points dans les clés d'étiquettes, ce qui résout le problème.(LOG-3341)

- Avant cette mise à jour, le plugin logging view contenait une fonctionnalité incompatible avec certaines versions d'OpenShift Container Platform. Avec cette mise à jour, la version correcte du plugin résout le problème.(LOG-3467)

- Avant cette mise à jour, la réconciliation de la ressource personnalisée **ClusterLogForwarder** signalait de manière incorrecte un état dégradé d'un ou de plusieurs pipelines, ce qui entraînait le redémarrage des pods collecteurs toutes les 8 à 10 secondes. Avec cette mise à jour, la

réconciliation de la ressource personnalisée **ClusterLogForwarder** se déroule correctement, ce qui résout le problème.(LOG-3469)

- Avant cette modification, la spécification du champ **outputDefaults** de la ressource personnalisée ClusterLogForwarder appliquait les paramètres à chaque type de sortie Elasticsearch déclaré. Ce changement corrige le comportement pour correspondre à la spécification d'amélioration où le paramètre s'applique spécifiquement au magasin Elasticsearch géré par défaut.(LOG-3342)

- Avant cette mise à jour, le script **must-gather** de l'OpenShift CLI (oc) ne se terminait pas car l'OpenShift CLI (oc) a besoin d'un dossier avec des droits d'écriture pour construire son cache. Avec cette mise à jour, l'OpenShift CLI (oc) a des droits d'écriture sur un dossier, et le script **must-gather** se termine avec succès. (LOG-3472)

- Avant cette mise à jour, le serveur webhook de Loki Operator provoquait des erreurs TLS. Avec cette mise à jour, l'ICP du webhook de Loki Operator est gérée par la gestion dynamique du webhook de Operator Lifecycle Manager, ce qui résout le problème.(LOG-3511)

1.10.2. CVE

- CVE-2021-46848
- CVE-2022-2056
- CVE-2022-2057
- CVE-2022-2058
- CVE-2022-2519
- CVE-2022-2520
- CVE-2022-2521
- CVE-2022-2867
- CVE-2022-2868
- CVE-2022-2869
- CVE-2022-2953
- CVE-2022-2964
- CVE-2022-4139
- CVE-2022-35737
- CVE-2022-42010
- CVE-2022-42011
- CVE-2022-42012
- CVE-2022-42898
- CVE-2022-43680

1.11. JOURNALISATION 5.5.5

Cette version inclut la version 5.5.5 de la correction des bugs de journalisation d'OpenShift .

1.11.1. Bug fixes

- Avant cette mise à jour, Kibana avait un délai d'expiration du cookie OAuth fixe **24h**, ce qui entraînait des erreurs 401 dans Kibana chaque fois que le champ **accessTokenInactivityTimeout** était défini sur une valeur inférieure à **24h**. Avec cette mise à jour, le délai d'expiration du cookie OAuth de Kibana se synchronise sur le champ **accessTokenInactivityTimeout**, avec une valeur par défaut de **24h**.(LOG-3305)

- Avant cette mise à jour, Vector analysait le champ message lorsque l'analyse JSON était activée sans définir les valeurs **structuredTypeKey** ou **structuredTypeName**. Avec cette mise à jour, une valeur est requise pour **structuredTypeKey** ou **structuredTypeName** lors de l'écriture de journaux structurés dans Elasticsearch.(LOG-3284)

- Avant cette mise à jour, l'alerte **FluentdQueueLengthIncreasing** pouvait ne pas se déclencher en cas de problème de cardinalité avec l'ensemble des étiquettes renvoyées par cette expression d'alerte. Cette mise à jour réduit les étiquettes pour n'inclure que celles nécessaires à l'alerte.(LOG-3226)

- Avant cette mise à jour, Loki n'avait pas de support pour atteindre un stockage externe dans un cluster déconnecté. Avec cette mise à jour, les variables d'environnement proxy et les bundles d'autorité de certification proxy sont inclus dans l'image du conteneur pour prendre en charge ces connexions.(LOG-2860)

- Avant cette mise à jour, les utilisateurs de la console web d'OpenShift Container Platform ne pouvaient pas choisir l'objet **ConfigMap** qui inclut le certificat CA pour Loki, ce qui faisait que les pods fonctionnaient sans le CA. Avec cette mise à jour, les utilisateurs de la console web peuvent sélectionner la carte de configuration, ce qui résout le problème.(LOG-3310)

- Avant cette mise à jour, la clé de l'autorité de certification était utilisée comme nom de volume pour le montage de l'autorité de certification dans Loki, ce qui provoquait des erreurs lorsque la clé de l'autorité de certification comportait des caractères non conformes (tels que des points). Avec cette mise à jour, le nom de volume est normalisé à une chaîne interne, ce qui résout le problème.(LOG-3332)

1.11.2. CVE

- CVE-2016-3709
- CVE-2020-35525
- CVE-2020-35527
- CVE-2020-36516
- CVE-2020-36558
- CVE-2021-3640
- CVE-2021-30002
- CVE-2022-0168

- CVE-2022-0561
- CVE-2022-0562
- CVE-2022-0617
- CVE-2022-0854
- CVE-2022-0865
- CVE-2022-0891
- CVE-2022-0908
- CVE-2022-0909
- CVE-2022-0924
- CVE-2022-1016
- CVE-2022-1048
- CVE-2022-1055
- CVE-2022-1184
- CVE-2022-1292
- CVE-2022-1304
- CVE-2022-1355
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1852
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2078
- CVE-2022-2097
- CVE-2022-2509
- CVE-2022-2586
- CVE-2022-2639
- CVE-2022-2938
- CVE-2022-3515

- CVE-2022-20368
- CVE-2022-21499
- CVE-2022-21618
- CVE-2022-21619
- CVE-2022-21624
- CVE-2022-21626
- CVE-2022-21628
- CVE-2022-22624
- CVE-2022-22628
- CVE-2022-22629
- CVE-2022-22662
- CVE-2022-22844
- CVE-2022-23960
- CVE-2022-24448
- CVE-2022-25255
- CVE-2022-26373
- CVE-2022-26700
- CVE-2022-26709
- CVE-2022-26710
- CVE-2022-26716
- CVE-2022-26717
- CVE-2022-26719
- CVE-2022-27404
- CVE-2022-27405
- CVE-2022-27406
- CVE-2022-27950
- CVE-2022-28390
- CVE-2022-28893
- CVE-2022-29581

- CVE-2022-30293
- CVE-2022-34903
- CVE-2022-36946
- CVE-2022-37434
- CVE-2022-39399

1.12. JOURNALISATION 5.5.4

Cette version inclut la version 5.5.4 de la correction des bogues de journalisation d'OpenShift (RHSA-2022:7434).

1.12.1. Bug fixes

- Avant cette mise à jour, une erreur dans l'analyseur de requêtes du plugin logging view entraînait la disparition de certaines parties de la requête de logs si celle-ci contenait des parenthèses curly **{}**. Cela rendait les requêtes invalides, ce qui entraînait le renvoi d'erreurs pour des requêtes valides. Avec cette mise à jour, l'analyseur traite correctement ces requêtes.(LOG-3042)
- Avant cette mise à jour, l'opérateur pouvait entrer dans une boucle de suppression et de recréation du daemonset du collecteur pendant que les déploiements Elasticsearch ou Kibana changeaient d'état. Avec cette mise à jour, une correction dans la gestion du statut de l'opérateur résout le problème.(LOG-3049)
- Avant cette mise à jour, aucune alerte n'était mise en œuvre pour prendre en charge l'implémentation du collecteur Vector. Cette modification ajoute des alertes Vector et déploie des alertes distinctes, en fonction de l'implémentation du collecteur choisie.(LOG-3127)
- Avant cette mise à jour, le composant de création de secret de l'Elasticsearch Operator modifiait constamment les secrets internes. Avec cette mise à jour, le secret existant est correctement géré.(LOG-3138)
- Avant cette mise à jour, une refonte des scripts de journalisation **must-gather** a supprimé l'emplacement prévu pour les artefacts. Cette mise à jour annule ce changement pour écrire les artefacts dans le dossier /**must-gather**.(LOG-3213)
- Avant cette mise à jour, sur certains clusters, l'exportateur Prometheus se liait à IPv4 au lieu d'IPv6. Après cette mise à jour, Fluentd détecte la version IP et se lie à **0.0.0.0** pour IPv4 ou **[::]** pour IPv6.(LOG-3162)

1.12.2. CVE

- CVE-2020-35525
- CVE-2020-35527
- CVE-2022-0494
- CVE-2022-1353
- CVE-2022-2509
- CVE-2022-2588

- CVE-2022-3515
- CVE-2022-21618
- CVE-2022-21619
- CVE-2022-21624
- CVE-2022-21626
- CVE-2022-21628
- CVE-2022-23816
- CVE-2022-23825
- CVE-2022-29900
- CVE-2022-29901
- CVE-2022-32149
- CVE-2022-37434
- CVE-2022-40674

1.13. JOURNALISATION 5.5.3

Cette version inclut la version 5.5.3 de la correction des bugs de journalisation d'OpenShift .

1.13.1. Bug fixes

- Avant cette mise à jour, les entrées de journal comportant des messages structurés incluaient le champ du message original, ce qui augmentait la taille de l'entrée. Cette mise à jour supprime le champ de message pour les journaux structurés afin de réduire la taille de l'entrée.(LOG-2759)

- Avant cette mise à jour, la configuration du collecteur excluait les journaux des pods **collector**, **default-log-store**, et **visualization**, mais n'était pas en mesure d'exclure les journaux archivés dans un fichier **.gz**. Avec cette mise à jour, les journaux archivés stockés dans les fichiers **.gz** des pods **collector**, **default-log-store** et **visualization** sont également exclus. (LOG-2844)

- Avant cette mise à jour, lorsque des requêtes vers un pod indisponible étaient envoyées via la passerelle, aucune alerte ne prévenait de l'interruption. Avec cette mise à jour, des alertes individuelles seront générées si la passerelle a des problèmes pour terminer une requête d'écriture ou de lecture.(LOG-2884)

- Avant cette mise à jour, les métadonnées de pods pouvaient être modifiées par des plugins fluents car les valeurs passaient par le pipeline par référence. Cette mise à jour assure que chaque message de log reçoit une copie des métadonnées du pod afin que chaque message soit traité indépendamment.(LOG-3046)

- Avant cette mise à jour, la sélection de la gravité **unknown** dans la vue des journaux de la console OpenShift excluait les journaux avec une valeur **level=unknown**. Avec cette mise à jour, les journaux sans niveau et avec des valeurs **level=unknown** sont visibles lors du filtrage par gravité **unknown**.(LOG-3062)

- Avant cette mise à jour, les enregistrements de logs envoyés à Elasticsearch avaient un champ supplémentaire nommé **write-index** qui contenait le nom de l'index vers lequel les logs devaient être envoyés. Ce champ ne fait pas partie du modèle de données. Après cette mise à jour, ce champ n'est plus envoyé.(LOG-3075)

- Avec l'introduction du nouveau contrôleur d'admission à la sécurité des pods intégré, les pods qui ne sont pas configurés conformément aux normes de sécurité définies globalement ou au niveau de l'espace de noms ne peuvent pas être exécutés. Avec cette mise à jour, l'opérateur et les collecteurs permettent une exécution privilégiée et s'exécutent sans avertissement ni erreur d'audit de sécurité.(LOG-3077)

- Avant cette mise à jour, l'opérateur supprimait toutes les sorties personnalisées définies dans la ressource personnalisée **ClusterLogForwarder** lorsqu'il utilisait LokiStack comme stockage de logs par défaut. Avec cette mise à jour, l'opérateur fusionne les sorties personnalisées avec les sorties par défaut lors du traitement de la ressource personnalisée **ClusterLogForwarder**.(LOG-3095)

1.13.2. CVE

- CVE-2015-20107
- CVE-2022-0391
- CVE-2022-2526
- CVE-2022-21123
- CVE-2022-21125
- CVE-2022-21166
- CVE-2022-29154
- CVE-2022-32206
- CVE-2022-32208
- CVE-2022-34903

1.14. JOURNALISATION 5.5.2

Cette version inclut la version 5.5.2 de la correction des bugs de journalisation d'OpenShift .

1.14.1. Bug fixes

- Avant cette mise à jour, les règles d'alerte pour le collecteur Fluentd n'adhéraient pas aux directives de style de surveillance de OpenShift Container Platform. Cette mise à jour modifie ces alertes pour inclure l'étiquette de l'espace de noms, ce qui résout le problème.(LOG-1823)

- Avant cette mise à jour, le script de basculement de la gestion des index ne parvenait pas à générer un nouveau nom d'index lorsque le nom de l'index comportait plus d'un trait d'union. Avec cette mise à jour, les noms d'index sont générés correctement.(LOG-2644)

- Avant cette mise à jour, la route Kibana définissait une valeur **caCertificate** sans qu'un certificat soit présent. Avec cette mise à jour, aucune valeur **caCertificate** n'est définie. (LOG-2661)

- Avant cette mise à jour, un changement dans les dépendances du collecteur provoquait l'émission d'un message d'avertissement pour les paramètres non utilisés. Avec cette mise à jour, la suppression des paramètres de configuration inutilisés résout le problème.(LOG-2859)

- Avant cette mise à jour, les pods créés pour les déploiements créés par l'opérateur Loki étaient planifiés par erreur sur des nœuds avec des systèmes d'exploitation non-Linux, si de tels nœuds étaient disponibles dans le cluster dans lequel l'opérateur s'exécutait. Avec cette mise à jour, l'opérateur attache un sélecteur de nœud supplémentaire aux définitions de pods qui permet uniquement de planifier les pods sur des nœuds basés sur Linux.(LOG-2895)

- Avant cette mise à jour, la vue Logs de la console OpenShift ne filtrait pas les logs par gravité en raison d'un problème d'analyseur LogQL dans la passerelle LokiStack. Avec cette mise à jour, un correctif d'analyseur résout le problème et la vue Logs de la console OpenShift peut filtrer par gravité.(LOG-2908)

- Avant cette mise à jour, une refonte des plugins du collecteur Fluentd a supprimé le champ timestamp pour les événements. Cette mise à jour rétablit le champ timestamp, qui provient de l'heure de réception de l'événement.(LOG-2923)

- Avant cette mise à jour, l'absence d'un champ **level** dans les journaux d'audit provoquait une erreur dans les journaux vectoriels. Avec cette mise à jour, l'ajout d'un champ **level** dans l'enregistrement du journal d'audit résout le problème.(LOG-2961)

- Avant cette mise à jour, si vous supprimiez la ressource personnalisée Kibana, la console web de OpenShift Container Platform continuait à afficher un lien vers Kibana. Avec cette mise à jour, la suppression de la ressource personnalisée Kibana supprime également ce lien.(LOG-3053)

- Avant cette mise à jour, chaque travail de basculement créait des index vides lorsque la ressource personnalisée **ClusterLogForwarder** avait une analyse JSON définie. Avec cette mise à jour, les nouveaux index ne sont pas vides(LOG-3063)

- Avant cette mise à jour, lorsque l'utilisateur supprimait la LokiStack après une mise à jour vers Loki Operator 5.5, les ressources créées à l'origine par Loki Operator 5.4 étaient conservées. Avec cette mise à jour, les références propriétaires des ressources pointent vers la LokiStack 5.5.(LOG-2945)

- Avant cette mise à jour, un utilisateur n'était pas en mesure de voir les journaux d'application des espaces de noms auxquels il avait accès. Avec cette mise à jour, l'opérateur Loki crée automatiquement un rôle de cluster et un lien de rôle de cluster permettant aux utilisateurs de lire les journaux d'application.(LOG-2918)

- Avant cette mise à jour, les utilisateurs ayant des privilèges d'administrateur de cluster n'étaient pas en mesure de visualiser correctement les journaux d'infrastructure et d'audit à l'aide de la console de journalisation. Avec cette mise à jour, le contrôle des autorisations a été étendu pour reconnaître également les utilisateurs des groupes cluster-admin et dedicated-admin en tant qu'administrateurs.(LOG-2970)

1.14.2. CVE

- CVE-2015-20107
- CVE-2022-0391
- CVE-2022-21123
- CVE-2022-21125

- CVE-2022-21166
- CVE-2022-29154
- CVE-2022-32206
- CVE-2022-32208
- CVE-2022-34903

1.15. JOURNALISATION 5.5.1

Cette version inclut la version 5.5.1 de la correction des bugs de journalisation d'OpenShift .

1.15.1. Améliorations

- Cette amélioration ajoute un onglet **Aggregated Logs** à la page **Pod Details** de la console web d'OpenShift Container Platform lorsque le plugin Logging Console est utilisé. Cette amélioration n'est disponible que sur OpenShift Container Platform 4.10 et plus.(LOG-2647)
- Cette amélioration ajoute Google Cloud Logging comme option de sortie pour la redirection des journaux.(LOG-1482)

1.15.2. Bug fixes

- Avant cette mise à jour, l'opérateur ne s'assurait pas que le module était prêt, ce qui entraînait un état inopérant du cluster lors d'un redémarrage. Avec cette mise à jour, l'opérateur marque les nouveaux pods comme étant prêts avant de passer à un nouveau pod lors d'un redémarrage, ce qui résout le problème.(LOG-2745)
- Avant cette mise à jour, Fluentd ne reconnaissait parfois pas que la plateforme Kubernetes effectuait une rotation du fichier de log et ne lisait plus les messages de log. Cette mise à jour corrige cela en définissant le paramètre de configuration suggéré par l'équipe de développement en amont.(LOG-2995)
- Avant cette mise à jour, l'ajout de la détection des erreurs multilignes entraînait une modification du routage interne et l'acheminement des enregistrements vers la mauvaise destination. Avec cette mise à jour, le routage interne est correct.(LOG-2801)
- Avant cette mise à jour, la modification de l'intervalle de rafraîchissement de la console web d'OpenShift Container Platform créait une erreur lorsque le champ **Query** était vide. Avec cette mise à jour, la modification de l'intervalle n'est pas une option disponible lorsque le champ **Query** est vide.(LOG-2917)

1.15.3. CVE

- CVE-2022-1705
- CVE-2022-2526
- CVE-2022-29154
- CVE-2022-30631
- CVE-2022-32148

- CVE-2022-32206
- CVE-2022-32208

1.16. ENREGISTREMENT 5.5

Les avis suivants sont disponibles pour Logging 5.5:Release 5.5

1.16.1. Améliorations

- Avec cette mise à jour, vous pouvez transférer des logs structurés provenant de différents conteneurs au sein d'un même pod vers différents index. Pour utiliser cette fonctionnalité, vous devez configurer le pipeline avec le support multi-conteneurs et annoter les pods.(LOG-1296)

IMPORTANT

Le formatage JSON des journaux varie selon les applications. La création d'un trop grand nombre d'index ayant un impact sur les performances, limitez l'utilisation de cette fonctionnalité à la création d'index pour les journaux dont les formats JSON sont incompatibles. Utilisez des requêtes pour séparer les journaux provenant de différents espaces de noms ou d'applications dont les formats JSON sont compatibles.

- Avec cette mise à jour, vous pouvez filtrer les journaux avec des sorties Elasticsearch en utilisant les étiquettes communes Kubernetes, **app.kubernetes.io/component**, **app.kubernetes.io/managed-by**, **app.kubernetes.io/part-of**, et **app.kubernetes.io/version**. Les types de sorties non Elasticsearch peuvent utiliser toutes les étiquettes incluses dans **kubernetes.labels**.(LOG-2388)

- Avec cette mise à jour, les clusters avec AWS Security Token Service (STS) activé peuvent utiliser l'authentification STS pour transmettre les journaux à Amazon CloudWatch.(LOG-1976)

- Avec cette mise à jour, l'opérateur "Loki Operator" et le collecteur vectoriel passent de l'aperçu technique à la disponibilité générale. La parité complète des fonctionnalités avec les versions précédentes est en attente, et certaines API restent en avant-première technique. Voir la section **Logging with the LokiStack** pour plus de détails.

1.16.2. Bug fixes

- Avant cette mise à jour, les clusters configurés pour transmettre les journaux à Amazon CloudWatch écrivaient les fichiers journaux rejetés dans le stockage temporaire, ce qui entraînait une instabilité du cluster au fil du temps. Avec cette mise à jour, la sauvegarde de morceaux pour toutes les options de stockage a été désactivée, ce qui résout le problème.(LOG-2746)

- Avant cette mise à jour, l'Opérateur utilisait des versions de certaines API qui sont obsolètes et dont la suppression est prévue dans les prochaines versions d'OpenShift Container Platform. Cette mise à jour déplace les dépendances vers les versions d'API prises en charge.(LOG-2656)

Avant cette mise à jour, l'Opérateur utilisait des versions de certaines API qui sont obsolètes et dont la suppression est prévue dans les prochaines versions d'OpenShift Container Platform. Cette mise à jour déplace les dépendances vers les versions d'API prises en charge.(LOG-2656)

- Avant cette mise à jour, plusieurs pipelines **ClusterLogForwarder** configurés pour la détection d'erreurs multilignes provoquaient l'entrée du collecteur dans l'état d'erreur **crashloopbackoff**.

Cette mise à jour corrige le problème où plusieurs sections de configuration avaient le même identifiant unique.(LOG-2241)

- Avant cette mise à jour, le collecteur ne pouvait pas enregistrer les symboles non UTF-8 dans les journaux de stockage Elasticsearch. Avec cette mise à jour, le collecteur encode les symboles non UTF-8, ce qui résout le problème.(LOG-2203)

- Avant cette mise à jour, les caractères non latins s'affichaient de manière incorrecte dans Kibana. Avec cette mise à jour, Kibana affiche correctement tous les symboles UTF-8 valides.(LOG-2784)

1.16.3. CVE

- CVE-2021-38561
- CVE-2022-1012
- CVE-2022-1292
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2097
- CVE-2022-21698
- CVE-2022-30631
- CVE-2022-32250

1.17. JOURNALISATION 5.4.11

Cette version inclut la version 5.4.11 de la correction des bugs de journalisation d'OpenShift .

1.17.1. Bug fixes

- BZ 2099524
- BZ 2161274

1.17.2. CVE

- CVE-2021-46848
- CVE-2022-3821
- CVE-2022-35737
- CVE-2022-42010

- CVE-2022-42011
- CVE-2022-42012
- CVE-2022-42898
- CVE-2022-43680

1.18. JOURNALISATION 5.4.10

Cette version inclut la version 5.4.10 de la correction des bugs de journalisation d'OpenShift .

1.18.1. Bug fixes

Aucun.

1.18.2. CVE

- CVE-2021-46848
- CVE-2022-2056
- CVE-2022-2057
- CVE-2022-2058
- CVE-2022-2519
- CVE-2022-2520
- CVE-2022-2521
- CVE-2022-2867
- CVE-2022-2868
- CVE-2022-2869
- CVE-2022-2953
- CVE-2022-2964
- CVE-2022-4139
- CVE-2022-35737
- CVE-2022-42010
- CVE-2022-42011
- CVE-2022-42012
- CVE-2022-42898
- CVE-2022-43680

1.19. JOURNALISATION 5.4.9

Cette version inclut la version 5.4.9 de la correction des bugs de journalisation d'OpenShift .

1.19.1. Bug fixes

- Avant cette mise à jour, le collecteur Fluentd avertissait des paramètres de configuration non utilisés. Cette mise à jour supprime ces paramètres de configuration et leurs messages d'avertissement.(LOG-3074)

- Avant cette mise à jour, Kibana avait un délai d'expiration du cookie OAuth fixe **24h**, ce qui entraînait des erreurs 401 dans Kibana chaque fois que le champ **accessTokenInactivityTimeout** était défini sur une valeur inférieure à **24h**. Avec cette mise à jour, le délai d'expiration du cookie OAuth de Kibana se synchronise sur le champ **accessTokenInactivityTimeout**, avec une valeur par défaut de **24h**.(LOG-3306)

1.19.2. CVE

- CVE-2016-3709
- CVE-2020-35525
- CVE-2020-35527
- CVE-2020-36516
- CVE-2020-36558
- CVE-2021-3640
- CVE-2021-30002
- CVE-2022-0168
- CVE-2022-0561
- CVE-2022-0562
- CVE-2022-0617
- CVE-2022-0854
- CVE-2022-0865
- CVE-2022-0891
- CVE-2022-0908
- CVE-2022-0909
- CVE-2022-0924
- CVE-2022-1016
- CVE-2022-1048
- CVE-2022-1055

- CVE-2022-1184
- CVE-2022-1292
- CVE-2022-1304
- CVE-2022-1355
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1852
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2078
- CVE-2022-2097
- CVE-2022-2509
- CVE-2022-2586
- CVE-2022-2639
- CVE-2022-2938
- CVE-2022-3515
- CVE-2022-20368
- CVE-2022-21499
- CVE-2022-21618
- CVE-2022-21619
- CVE-2022-21624
- CVE-2022-21626
- CVE-2022-21628
- CVE-2022-22624
- CVE-2022-22628
- CVE-2022-22629
- CVE-2022-22662
- CVE-2022-22844

- CVE-2022-23960
- CVE-2022-24448
- CVE-2022-25255
- CVE-2022-26373
- CVE-2022-26700
- CVE-2022-26709
- CVE-2022-26710
- CVE-2022-26716
- CVE-2022-26717
- CVE-2022-26719
- CVE-2022-27404
- CVE-2022-27405
- CVE-2022-27406
- CVE-2022-27950
- CVE-2022-28390
- CVE-2022-28893
- CVE-2022-29581
- CVE-2022-30293
- CVE-2022-34903
- CVE-2022-36946
- CVE-2022-37434
- CVE-2022-39399

1.20. JOURNALISATION 5.4.8

Cette version inclut la version 5.4.8 de la correction des bogues de journalisation d'OpenShift (RHSA-2022:7435).

1.20.1. Bug fixes

Aucun.

1.20.2. CVE

- CVE-2016-3709

- CVE-2020-35525
- CVE-2020-35527
- CVE-2020-36518
- CVE-2022-1304
- CVE-2022-2509
- CVE-2022-3515
- CVE-2022-22624
- CVE-2022-22628
- CVE-2022-22629
- CVE-2022-22662
- CVE-2022-26700
- CVE-2022-26709
- CVE-2022-26710
- CVE-2022-26716
- CVE-2022-26717
- CVE-2022-26719
- CVE-2022-30293
- CVE-2022-32149
- CVE-2022-37434
- CVE-2022-40674
- CVE-2022-42003
- CVE-2022-42004

1.21. JOURNALISATION 5.4.6

Cette version inclut la version 5.4.6 de la correction des bugs de journalisation d'OpenShift .

1.21.1. Bug fixes

- Avant cette mise à jour, Fluentd ne reconnaissait parfois pas que la plateforme Kubernetes effectuait une rotation du fichier de log et ne lisait plus les messages de log. Cette mise à jour corrige cela en définissant le paramètre de configuration suggéré par l'équipe de développement en amont.(LOG-2792)

- Avant cette mise à jour, chaque travail de basculement créait des index vides lorsque la ressource personnalisée **ClusterLogForwarder** avait une définition de l'analyse JSON. Avec cette mise à jour, les nouveaux index ne sont pas vides(LOG-2823)

- Avant cette mise à jour, si vous supprimiez la ressource personnalisée Kibana, la console web de OpenShift Container Platform continuait à afficher un lien vers Kibana. Avec cette mise à jour, la suppression de la ressource personnalisée Kibana supprime également ce lien.(LOG-3054)

1.21.2. CVE

- CVE-2015-20107
- CVE-2022-0391
- CVE-2022-21123
- CVE-2022-21125
- CVE-2022-21166
- CVE-2022-29154
- CVE-2022-32206
- CVE-2022-32208
- CVE-2022-34903

1.22. JOURNALISATION 5.4.5

Cette version inclut la version 5.4.5 de la correction des bogues de journalisation d'OpenShift, RHSA-2022:6183.

1.22.1. Bug fixes

- Avant cette mise à jour, l'opérateur ne s'assurait pas que le module était prêt, ce qui entraînait un état inopérant du cluster lors d'un redémarrage. Avec cette mise à jour, l'opérateur marque les nouveaux pods comme étant prêts avant de passer à un nouveau pod lors d'un redémarrage, ce qui résout le problème.(LOG-2881)

- Avant cette mise à jour, l'ajout de la détection des erreurs multilignes entraînait une modification du routage interne et l'acheminement des enregistrements vers la mauvaise destination. Avec cette mise à jour, le routage interne est correct.(LOG-2946)

- Avant cette mise à jour, l'opérateur ne pouvait pas décoder les réponses JSON relatives aux paramètres d'index avec une valeur booléenne citée, ce qui entraînait une erreur. Avec cette mise à jour, l'opérateur peut décoder correctement cette réponse JSON.(LOG-3009)

- Avant cette mise à jour, les modèles d'index Elasticsearch définissaient les champs des étiquettes avec les mauvais types. Cette modification met à jour ces modèles pour qu'ils correspondent aux types attendus transmis par le collecteur de logs.(LOG-2972)

1.22.2. CVE

- CVE-2022-1292

- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2097
- CVE-2022-30631

1.23. JOURNALISATION 5.4.4

Cette version inclut la version 5.4.4 de RHBA-2022:5907-OpenShift Logging Bug Fix .

1.23.1. Bug fixes

- Avant cette mise à jour, les caractères non latins s'affichaient de manière incorrecte dans Elasticsearch. Avec cette mise à jour, Elasticsearch affiche correctement tous les symboles UTF-8 valides.(LOG-2794)
- Avant cette mise à jour, les caractères non-latins s'affichaient incorrectement dans Fluentd. Avec cette mise à jour, Fluentd affiche correctement tous les symboles UTF-8 valides.(LOG-2657)
- Avant cette mise à jour, le serveur de métriques pour le collecteur tentait de se lier à l'adresse en utilisant une valeur exposée par une valeur d'environnement. Ce changement modifie la configuration pour se lier à n'importe quelle interface disponible.(LOG-2821)
- Avant cette mise à jour, l'opérateur **cluster-logging** s'appuyait sur le cluster pour créer un secret. Ce comportement du cluster a changé dans OpenShift Container Platform 4.11, ce qui a entraîné l'échec des déploiements de journalisation. Avec cette mise à jour, l'opérateur **cluster-logging** résout le problème en créant le secret si nécessaire. (LOG-2840)

1.23.2. CVE

- CVE-2022-21540
- CVE-2022-21541
- CVE-2022-34169

1.24. JOURNALISATION 5.4.3

Cette version inclut la version 5.4.3 de la correction des bogues de journalisation d'OpenShift (RHSA-2022:5556).

1.24.1. Avis de dépréciation d'Elasticsearch Operator

Dans le sous-système de journalisation 5.4.3, l'opérateur Elasticsearch est obsolète et il est prévu de le supprimer dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne

recevra plus d'améliorations et sera supprimée. Au lieu d'utiliser l'opérateur Elasticsearch pour gérer le stockage des journaux par défaut, vous pouvez utiliser l'opérateur Loki.

1.24.2. Bug fixes

- Avant cette mise à jour, le tableau de bord OpenShift Logging Dashboard affichait le nombre de shards primaires actifs au lieu de tous les shards actifs. Avec cette mise à jour, le tableau de bord affiche tous les shards actifs.(LOG-2781)

- Avant cette mise à jour, un bogue dans une bibliothèque utilisée par **elasticsearch-operator** contenait une vulnérabilité d'attaque par déni de service. Avec cette mise à jour, la bibliothèque a été mise à jour vers une version qui ne contient pas cette vulnérabilité.(LOG-2816)

- Avant cette mise à jour, lors de la configuration de Vector pour transmettre les journaux à Loki, il n'était pas possible de définir un jeton de support personnalisé ou d'utiliser le jeton par défaut si Loki avait activé TLS. Avec cette mise à jour, Vector peut transmettre les journaux à Loki en utilisant des jetons avec TLS activé.(LOG-2786

- Avant cette mise à jour, l'opérateur ElasticSearch omettait la propriété **referencePolicy** de la ressource personnalisée **ImageStream** lors de la sélection d'une image **oauth-proxy**. Cette omission entraînait l'échec du déploiement de Kibana dans certains environnements. Avec cette mise à jour, l'utilisation de **referencePolicy** résout le problème et l'opérateur peut déployer Kibana avec succès.(LOG-2791)

- Avant cette mise à jour, les règles d'alerte pour la ressource personnalisée **ClusterLogForwarder** ne prenaient pas en compte les sorties multiples. Cette mise à jour résout le problème.(LOG-2640)

- Avant cette mise à jour, les clusters configurés pour transmettre les journaux à Amazon CloudWatch écrivaient les fichiers journaux rejetés dans le stockage temporaire, ce qui entraînait une instabilité du cluster au fil du temps. Avec cette mise à jour, la sauvegarde de chunk pour CloudWatch a été désactivée, ce qui résout le problème.(LOG-2768)

1.24.3. CVE

Exemple 1.1. Cliquez pour agrandir CVEs

- CVE-2020-28915
- CVE-2021-40528
- CVE-2022-1271
- CVE-2022-1621
- CVE-2022-1629
- CVE-2022-22576
- CVE-2022-25313
- CVE-2022-25314
- CVE-2022-26691
- CVE-2022-27666

- CVE-2022-27774
- CVE-2022-27776
- CVE-2022-27782
- CVE-2022-29824

1.25. JOURNALISATION 5.4.2

Cette version inclut RHBA-2022:4874-OpenShift Logging Bug Fix Release 5.4.2

1.25.1. Bug fixes

- Avant cette mise à jour, l'édition de la configuration du collecteur à l'aide de **oc edit** était difficile en raison de l'utilisation incohérente des espaces blancs. Cette modification introduit une logique de normalisation et de formatage de la configuration avant toute mise à jour par l'opérateur afin qu'elle soit facile à éditer à l'aide de **oc edit**.(LOG-2319)
- Avant cette mise à jour, l'alerte **FluentdNodeDown** ne pouvait pas fournir les étiquettes d'instance dans la section du message de manière appropriée. Cette mise à jour résout le problème en corrigeant la règle d'alerte pour fournir des étiquettes d'instance dans les cas d'échecs partiels de l'instance.(LOG-2607)
- Avant cette mise à jour, plusieurs niveaux d'enregistrement, tels que "critique", qui étaient documentés comme étant pris en charge par le produit ne l'étaient pas. Cette mise à jour corrige l'anomalie de sorte que les niveaux de journalisation documentés sont maintenant pris en charge par le produit.(LOG-2033)

1.25.2. CVE

Exemple 1.2. Cliquez pour agrandir CVEs

- CVE-2018-25032
- CVE-2020-0404
- CVE-2020-4788
- CVE-2020-13974
- CVE-2020-19131
- CVE-2020-27820
- CVE-2021-0941
- CVE-2021-3612
- CVE-2021-3634
- CVE-2021-3669
- CVE-2021-3737

- CVE-2021-3743
- CVE-2021-3744
- CVE-2021-3752
- CVE-2021-3759
- CVE-2021-3764
- CVE-2021-3772
- CVE-2021-3773
- CVE-2021-4002
- CVE-2021-4037
- CVE-2021-4083
- CVE-2021-4157
- CVE-2021-4189
- CVE-2021-4197
- CVE-2021-4203
- CVE-2021-20322
- CVE-2021-21781
- CVE-2021-23222
- CVE-2021-26401
- CVE-2021-29154
- CVE-2021-37159
- CVE-2021-41617
- CVE-2021-41864
- CVE-2021-42739
- CVE-2021-43056
- CVE-2021-43389
- CVE-2021-43976
- CVE-2021-44733
- CVE-2021-45485
- CVE-2021-45486

- CVE-2022-0001
- CVE-2022-0002
- CVE-2022-0286
- CVE-2022-0322
- CVE-2022-1011
- CVE-2022-1271

1.26. JOURNALISATION 5.4.1

Cette version inclut la version 5.4.1 de la correction des bogues de journalisation d'OpenShift RHSA-2022:2216.

1.26.1. Bug fixes

- Avant cette mise à jour, l'exportateur de métriques de fichiers journaux ne signalait que les journaux créés pendant que l'exportateur était en cours d'exécution, ce qui entraînait des données inexactes sur la croissance des journaux. Cette mise à jour résout ce problème en surveillant **/var/log/pods**.(LOG-2442)

- Avant cette mise à jour, le collecteur était bloqué parce qu'il essayait continuellement d'utiliser une connexion périmée lorsqu'il transmettait les journaux aux récepteurs de transfert fluentd. Avec cette version, la valeur de **keepalive_timeout** a été fixée à 30 secondes (**30s**) afin que le collecteur recycle la connexion et réessaie d'envoyer les messages échoués dans un délai raisonnable.(LOG-2534)

- Avant cette mise à jour, une erreur dans le composant de la passerelle appliquant la tenance pour la lecture des journaux limitait l'accès aux journaux avec un espace de noms Kubernetes, ce qui rendait illisibles les journaux \N "audit" et certains journaux \N "infrastructure". Avec cette mise à jour, le proxy détecte correctement les utilisateurs ayant un accès administrateur et autorise l'accès aux journaux sans espace de noms.(LOG-2448)

- Avant cette mise à jour, le compte de service **system:serviceaccount:openshift-monitoring:prometheus-k8s** avait des privilèges au niveau du cluster en tant que **clusterrole** et **clusterrolebinding**. Cette mise à jour restreint le compte de service ` à l'espace de noms **openshift-logging** avec un rôle et une liaison de rôle. (LOG-2437)

- Avant cette mise à jour, l'analyse de l'heure du journal d'audit Linux reposait sur la position ordinale d'une paire clé/valeur. Cette mise à jour modifie l'analyse pour utiliser une expression régulière afin de trouver l'entrée temporelle.(LOG-2321)

1.26.2. CVE

Exemple 1.3. Cliquez pour agrandir CVEs

- CVE-2018-25032
- CVE-2021-4028
- CVE-2021-37136

- CVE-2021-37137
- CVE-2021-43797
- CVE-2022-0778
- CVE-2022-1154
- CVE-2022-1271
- CVE-2022-21426
- CVE-2022-21434
- CVE-2022-21443
- CVE-2022-21476
- CVE-2022-21496
- CVE-2022-21698
- CVE-2022-25636

1.27. ENREGISTREMENT 5.4

Les avis suivants sont disponibles pour la journalisation 5.4 : Sous-système de journalisation pour Red Hat OpenShift version 5.4

1.27.1. Aperçus technologiques

IMPORTANT

Vector est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

1.27.2. À propos de Vector

Vector est un collecteur de journaux proposé en tant qu'alternative technique au collecteur par défaut du sous-système de journalisation.

Les sorties suivantes sont prises en charge :

- **elasticsearch**. Une instance Elasticsearch externe. La sortie **elasticsearch** peut utiliser une connexion TLS.

- **kafka**. Un courtier Kafka. La sortie **kafka** peut utiliser une connexion non sécurisée ou TLS.
- **loki**. Loki, un système d'agrégation de logs horizontalement extensible, hautement disponible et multi-tenant.

1.27.2.1. Vecteur d'habilitation

Vector n'est pas activé par défaut. Suivez les étapes suivantes pour activer Vector sur votre cluster OpenShift Container Platform.

IMPORTANT

Vector ne prend pas en charge les clusters compatibles FIPS.

Conditions préalables

- OpenShift Container Platform : 4.12
- Sous-système de journalisation pour Red Hat OpenShift : 5.4
- FIPS désactivé

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc -n openshift-logging edit ClusterLogging instance
```

2. Ajouter une annotation **logging.openshift.io/preview-vector-collector: enabled** à la ressource personnalisée (CR) **ClusterLogging**.
3. Ajouter **vector** comme type de collection à la ressource personnalisée (CR) **ClusterLogging**.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
  annotations:
    logging.openshift.io/preview-vector-collector: enabled
spec:
  collection:
    logs:
      type: "vector"
      vector: {}
```

Ressources complémentaires

- Documentation sur les vecteurs

IMPORTANT

Loki Operator est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes sur le plan fonctionnel. Red Hat ne recommande pas de les utiliser en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, permettant aux clients de tester les fonctionnalités et de fournir un retour d'information au cours du processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

1.27.3. À propos de Loki

Loki est un système d'agrégation de journaux horizontalement extensible, hautement disponible et multi-tenant, actuellement proposé comme alternative à Elasticsearch en tant que magasin de journaux pour le sous-système de journalisation.

Ressources complémentaires

- Loki Documentation

1.27.3.1. Déploiement du Lokistack

Vous pouvez utiliser la console web d'OpenShift Container Platform pour installer l'opérateur Loki.

Conditions préalables

- OpenShift Container Platform : 4.12
- Sous-système de journalisation pour Red Hat OpenShift : 5.4

Pour installer l'opérateur Loki à l'aide de la console web d'OpenShift Container Platform :

1. Installer l'opérateur Loki :
   a. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
   b. Choisissez **Loki Operator** dans la liste des opérateurs disponibles et cliquez sur **Install**.
   c. Sous **Installation Mode**, sélectionnez **All namespaces on the cluster**.
   d. Sous **Installed Namespace**, sélectionnez **openshift-operators-redhat**.
   Vous devez spécifier l'espace de noms **openshift-operators-redhat**. L'espace de noms **openshift-operators** peut contenir des Community Operators, qui ne sont pas fiables et qui pourraient publier une métrique avec le même nom qu'une métrique OpenShift Container Platform, ce qui causerait des conflits.
   e. Sélectionnez **Enable operator recommended cluster monitoring on this namespace**
   Cette option définit l'étiquette **openshift.io/cluster-monitoring: "true"** dans l'objet Namespace. Vous devez sélectionner cette option pour vous assurer que la surveillance des clusters récupère l'espace de noms **openshift-operators-redhat**.

f. Sélectionnez un site **Approval Strategy**.

- La stratégie **Automatic** permet à Operator Lifecycle Manager (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible.
- La stratégie **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.

g. Cliquez sur **Install**.

h. Vérifiez que vous avez installé Loki Operator. Visitez la page **Operators → Installed Operators** et cherchez \N "Loki Operator.\N"

i. Assurez-vous que **Loki Operator** est listé dans tous les projets dont **Status** est **Succeeded**.

1.27.4. Bug fixes

- Avant cette mise à jour, le site **cluster-logging-operator** utilisait des rôles et des liaisons à l'échelle du cluster pour établir des autorisations permettant au compte de service Prometheus d'analyser les mesures. Ces autorisations étaient créées lors du déploiement de l'opérateur à l'aide de l'interface de la console, mais n'existaient pas lors du déploiement à partir de la ligne de commande. Cette mise à jour corrige le problème en faisant en sorte que les rôles et les liaisons soient définis par l'espace de noms.(LOG-2286)

- Avant cette mise à jour, une modification antérieure visant à corriger la réconciliation du tableau de bord a introduit un champ **ownerReferences** dans la ressource à travers les espaces de noms. En conséquence, la carte de configuration et le tableau de bord n'ont pas été créés dans l'espace de noms. Avec cette mise à jour, la suppression du champ **ownerReferences** résout le problème et le tableau de bord OpenShift Logging est disponible dans la console.(LOG-2163)

- Avant cette mise à jour, les modifications apportées aux tableaux de bord de mesure n'étaient pas déployées car **cluster-logging-operator** ne comparait pas correctement les cartes de configuration existantes et modifiées contenant le tableau de bord. Avec cette mise à jour, l'ajout d'une valeur de hachage unique aux étiquettes d'objets résout le problème.(LOG-2071)

- Avant cette mise à jour, le tableau de bord OpenShift Logging n'affichait pas correctement les pods et namespaces dans le tableau qui affiche les conteneurs les plus productifs collectés au cours des dernières 24 heures. Avec cette mise à jour, les pods et namespaces sont affichés correctement.(LOG-2069)

- Avant cette mise à jour, lorsque le site **ClusterLogForwarder** était configuré avec **Elasticsearch OutputDefault** et que les sorties Elasticsearch n'avaient pas de clés structurées, la configuration générée contenait des valeurs incorrectes pour l'authentification. Cette mise à jour corrige le secret et les certificats utilisés.(LOG-2056)

- Avant cette mise à jour, le tableau de bord OpenShift Logging affichait un graphique CPU vide en raison d'une référence à une métrique invalide. Avec cette mise à jour, le point de données correct a été sélectionné, ce qui résout le problème.(LOG-2026)

- Avant cette mise à jour, l'image du conteneur Fluentd incluait des outils de construction qui n'étaient pas nécessaires à l'exécution. Cette mise à jour supprime ces outils de l'image.(LOG-1927)

- Avant cette mise à jour, un changement de nom du collecteur déployé dans la version 5.3 entraînait la génération de l'alerte **FluentdNodeDown** par le collecteur de journalisation. Cette mise à jour résout le problème en corrigeant le nom du travail pour l'alerte Prometheus.(LOG-1918)

- Avant cette mise à jour, le collecteur de logs collectait ses propres logs en raison d'une refonte du changement de nom du composant. Cela conduisait à une boucle de rétroaction potentielle du collecteur traitant ses propres journaux, ce qui pouvait entraîner des problèmes de mémoire et de taille des messages de journaux. Cette mise à jour résout le problème en excluant les journaux du collecteur de la collecte.(LOG-1774)

- Avant cette mise à jour, Elasticsearch générait l'erreur **Unable to create PersistentVolumeClaim due to forbidden: exceeded quota: infra-storage-quota.** si le PVC existait déjà. Avec cette mise à jour, Elasticsearch vérifie les PVC existants, ce qui résout le problème.(LOG-2131)

- Avant cette mise à jour, Elasticsearch ne pouvait pas revenir à l'état prêt lorsque le secret **elasticsearch-signing** était supprimé. Avec cette mise à jour, Elasticsearch est en mesure de revenir à l'état prêt après la suppression de ce secret.(LOG-2171)

- Avant cette mise à jour, la modification du chemin à partir duquel le collecteur lit les journaux de conteneurs entraînait la transmission de certains enregistrements aux mauvais index. Avec cette mise à jour, le collecteur utilise maintenant la configuration correcte pour résoudre le problème.(LOG-2160)

- Avant cette mise à jour, les clusters comportant un grand nombre d'espaces de noms empêchaient Elasticsearch de servir les requêtes car la liste des espaces de noms atteignait la limite de taille maximale de l'en-tête. Avec cette mise à jour, les en-têtes n'incluent qu'une liste de noms d'espaces de noms, ce qui résout le problème.(LOG-1899)

- Avant cette mise à jour, le tableau de bord **OpenShift Container Platform Logging** affichait un nombre de shards 'x' fois supérieur à la valeur réelle lorsque Elasticsearch avait 'x' nœuds. Ce problème était dû au fait que le tableau de bord imprimait tous les shards primaires pour chaque pod Elasticsearch et calculait une somme, alors que la sortie était toujours pour l'ensemble du cluster Elasticsearch. Avec cette mise à jour, le nombre de shards est maintenant correctement calculé.(LOG-2156)

- Avant cette mise à jour, les secrets **kibana** et **kibana-proxy** n'étaient pas recréés s'ils étaient supprimés manuellement. Avec cette mise à jour, **elasticsearch-operator** surveillera les ressources et les recréera automatiquement si elles sont supprimées.(LOG-2250)

- Avant cette mise à jour, le réglage de la taille du bloc de la mémoire tampon pouvait entraîner la génération d'un avertissement par le collecteur concernant la taille du bloc dépassant la limite d'octets pour le flux d'événements. Avec cette mise à jour, vous pouvez également régler la limite de ligne de lecture, ce qui résout le problème.(LOG-2379)

- Avant cette mise à jour, le lien de la console de journalisation dans la console web d'OpenShift n'était pas supprimé avec le CR ClusterLogging. Avec cette mise à jour, la suppression de la CR ou la désinstallation de Cluster Logging Operator supprime le lien.(LOG-2373)

- Avant cette mise à jour, une modification du chemin d'accès aux journaux des conteneurs faisait que la métrique de collecte était toujours égale à zéro avec les anciennes versions configurées avec le chemin d'accès d'origine. Avec cette mise à jour, le plugin qui expose les métriques sur les journaux collectés prend en charge la lecture à partir de l'un ou l'autre chemin pour résoudre le problème.(LOG-2462)

1.27.5. CVE

- CVE-2022-0759
    - BZ-2058404

- CVE-2022-21698
  - BZ-2045880

1.28. JOURNALISATION 5.3.14

Cette version inclut la version 5.3.14 de la correction des bugs de journalisation d'OpenShift .

1.28.1. Bug fixes

- Avant cette mise à jour, la carte de taille des fichiers journaux générée par le composant **log-file-metrics-exporter** ne supprimait pas les entrées des fichiers supprimés, ce qui augmentait la taille des fichiers et la mémoire du processus. Avec cette mise à jour, le plan de taille des fichiers journaux ne contient pas d'entrées pour les fichiers supprimés.(LOG-3293)

1.28.2. CVE

- CVE-2016-3709
- CVE-2020-35525
- CVE-2020-35527
- CVE-2020-36516
- CVE-2020-36558
- CVE-2021-3640
- CVE-2021-30002
- CVE-2022-0168
- CVE-2022-0561
- CVE-2022-0562
- CVE-2022-0617
- CVE-2022-0854
- CVE-2022-0865
- CVE-2022-0891
- CVE-2022-0908
- CVE-2022-0909
- CVE-2022-0924
- CVE-2022-1016
- CVE-2022-1048
- CVE-2022-1055

- CVE-2022-1184
- CVE-2022-1292
- CVE-2022-1304
- CVE-2022-1355
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1852
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2078
- CVE-2022-2097
- CVE-2022-2509
- CVE-2022-2586
- CVE-2022-2639
- CVE-2022-2938
- CVE-2022-3515
- CVE-2022-20368
- CVE-2022-21499
- CVE-2022-21618
- CVE-2022-21619
- CVE-2022-21624
- CVE-2022-21626
- CVE-2022-21628
- CVE-2022-22624
- CVE-2022-22628
- CVE-2022-22629
- CVE-2022-22662
- CVE-2022-22844

- CVE-2022-23960
- CVE-2022-24448
- CVE-2022-25255
- CVE-2022-26373
- CVE-2022-26700
- CVE-2022-26709
- CVE-2022-26710
- CVE-2022-26716
- CVE-2022-26717
- CVE-2022-26719
- CVE-2022-27404
- CVE-2022-27405
- CVE-2022-27406
- CVE-2022-27950
- CVE-2022-28390
- CVE-2022-28893
- CVE-2022-29581
- CVE-2022-30293
- CVE-2022-34903
- CVE-2022-36946
- CVE-2022-37434
- CVE-2022-39399
- CVE-2022-42898

1.29. JOURNALISATION 5.3.13

Cette version inclut la version 5.3.13 de RHSA-2022:68828-OpenShift Logging Bug Fix .

1.29.1. Bug fixes

Aucun.

1.29.2. CVE

Exemple 1.4. Cliquez pour agrandir CVEs

- CVE-2020-35525
- CVE-2020-35527
- CVE-2022-0494
- CVE-2022-1353
- CVE-2022-2509
- CVE-2022-2588
- CVE-2022-3515
- CVE-2022-21618
- CVE-2022-21619
- CVE-2022-21624
- CVE-2022-21626
- CVE-2022-21628
- CVE-2022-23816
- CVE-2022-23825
- CVE-2022-29900
- CVE-2022-29901
- CVE-2022-32149
- CVE-2022-37434
- CVE-2022-39399
- CVE-2022-40674

1.30. JOURNALISATION 5.3.12

Cette version inclut la version 5.3.12 de la correction des bugs de journalisation d'OpenShift .

1.30.1. Bug fixes

Aucun.

1.30.2. CVE

- CVE-2015-20107
- CVE-2022-0391

- CVE-2022-21123
- CVE-2022-21125
- CVE-2022-21166
- CVE-2022-29154
- CVE-2022-32206
- CVE-2022-32208
- CVE-2022-34903

1.31. JOURNALISATION 5.3.11

Cette version inclut la version 5.3.11 de la correction des bugs de journalisation d'OpenShift .

1.31.1. Bug fixes

- Avant cette mise à jour, l'opérateur ne s'assurait pas que le module était prêt, ce qui entraînait un état inopérant du cluster lors d'un redémarrage. Avec cette mise à jour, l'opérateur marque les nouveaux pods comme étant prêts avant de passer à un nouveau pod lors d'un redémarrage, ce qui résout le problème.(LOG-2871)

1.31.2. CVE

- CVE-2022-1292
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2097
- CVE-2022-30631

1.32. JOURNALISATION 5.3.10

Cette version inclut la version 5.3.10 de RHSA-2022:5908-OpenShift Logging Bug Fix .

1.32.1. Bug fixes

- BZ-2100495

1.32.2. CVE

Exemple 1.5. Cliquez pour agrandir CVEs

- CVE-2021-38561
- CVE-2021-40528
- CVE-2022-1271
- CVE-2022-1621
- CVE-2022-1629
- CVE-2022-21540
- CVE-2022-21541
- CVE-2022-22576
- CVE-2022-25313
- CVE-2022-25314
- CVE-2022-27774
- CVE-2022-27776
- CVE-2022-27782
- CVE-2022-29824
- CVE-2022-34169

1.33. JOURNALISATION 5.3.9

Cette version inclut la correction de bogue RHBA-2022:5557-OpenShift Logging Release 5.3.9 .

1.33.1. Bug fixes

- Avant cette mise à jour, le collecteur de journalisation incluait un chemin d'accès comme étiquette pour les mesures qu'il produisait. Ce chemin changeait fréquemment et contribuait à des changements de stockage importants pour le serveur Prometheus. Avec cette mise à jour, l'étiquette a été supprimée pour résoudre le problème et réduire la consommation de stockage.(LOG-2682)

1.33.2. CVE

Exemple 1.6. Cliquez pour agrandir CVEs

- CVE-2020-28915
- CVE-2021-40528
- CVE-2022-1271
- CVE-2022-1621
- CVE-2022-1629

- CVE-2022-22576
- CVE-2022-25313
- CVE-2022-25314
- CVE-2022-26691
- CVE-2022-27666
- CVE-2022-27774
- CVE-2022-27776
- CVE-2022-27782
- CVE-2022-29824

1.34. JOURNALISATION 5.3.8

Cette version inclut la version 5.3.8 de RHBA-2022:5010-OpenShift Logging Bug Fix

1.34.1. Bug fixes

(Aucun.)

1.34.2. CVE

Exemple 1.7. Cliquez pour agrandir CVEs

- CVE-2018-25032
- CVE-2020-0404
- CVE-2020-4788
- CVE-2020-13974
- CVE-2020-19131
- CVE-2020-27820
- CVE-2021-0941
- CVE-2021-3612
- CVE-2021-3634
- CVE-2021-3669
- CVE-2021-3737
- CVE-2021-3743
- CVE-2021-3744

- CVE-2021-3752
- CVE-2021-3759
- CVE-2021-3764
- CVE-2021-3772
- CVE-2021-3773
- CVE-2021-4002
- CVE-2021-4037
- CVE-2021-4083
- CVE-2021-4157
- CVE-2021-4189
- CVE-2021-4197
- CVE-2021-4203
- CVE-2021-20322
- CVE-2021-21781
- CVE-2021-23222
- CVE-2021-26401
- CVE-2021-29154
- CVE-2021-37159
- CVE-2021-41617
- CVE-2021-41864
- CVE-2021-42739
- CVE-2021-43056
- CVE-2021-43389
- CVE-2021-43976
- CVE-2021-44733
- CVE-2021-45485
- CVE-2021-45486
- CVE-2022-0001
- CVE-2022-0002

- CVE-2022-0286
- CVE-2022-0322
- CVE-2022-1011
- CVE-2022-1271

1.35. OPENSHIFT LOGGING 5.3.7

Cette version inclut RHSA-2022:2217 OpenShift Logging Bug Fix Release 5.3.7

1.35.1. Bug fixes

- Avant cette mise à jour, l'analyse de l'heure du journal d'audit Linux reposait sur la position ordinale d'une paire clé/valeur. Cette mise à jour modifie l'analyse en utilisant une expression rationnelle pour trouver l'entrée temporelle.(LOG-2322)
- Avant cette mise à jour, certaines sorties de transfert de journaux pouvaient réorganiser les journaux ayant le même horodatage. Avec cette mise à jour, un numéro de séquence a été ajouté à l'enregistrement du journal pour ordonner les entrées dont les horodatages correspondent.(LOG-2334)
- Avant cette mise à jour, les clusters comportant un grand nombre d'espaces de noms empêchaient Elasticsearch de servir les requêtes car la liste des espaces de noms atteignait la limite de taille maximale de l'en-tête. Avec cette mise à jour, les en-têtes n'incluent qu'une liste de noms d'espaces de noms, ce qui résout le problème.(LOG-2450)
- Avant cette mise à jour, **system:serviceaccount:openshift-monitoring:prometheus-k8s** avait des privilèges de niveau cluster en tant que **clusterrole** et **clusterrolebinding**. Cette mise à jour restreint **serviceaccount** à l'espace de noms **openshift-logging** avec un rôle et un lien de rôle.(LOG-2481))

1.35.2. CVE

Exemple 1.8. Cliquez pour agrandir CVEs

- CVE-2018-25032
- CVE-2021-4028
- CVE-2021-37136
- CVE-2021-37137
- CVE-2021-43797
- CVE-2022-0759
- CVE-2022-0778
- CVE-2022-1154
- CVE-2022-1271

- CVE-2022-21426
- CVE-2022-21434
- CVE-2022-21443
- CVE-2022-21476
- CVE-2022-21496
- CVE-2022-21698
- CVE-2022-25636

1.36. OPENSHIFT LOGGING 5.3.6

Cette version inclut RHBA-2022:1377 OpenShift Logging Bug Fix Release 5.3.6

1.36.1. Bug fixes

- Avant cette mise à jour, la définition d'une tolérance sans clé avec l'opérateur existant entraînait l'impossibilité pour l'opérateur d'effectuer une mise à niveau. Avec cette mise à jour, cette tolérance ne bloque plus l'achèvement de la mise à niveau.(LOG-2126)
- Avant cette modification, il était possible que le collecteur génère un avertissement lorsque la limite de l'octet du bloc dépassait un événement émis. Avec ce changement, vous pouvez ajuster la limite de lecture pour résoudre le problème comme le conseille la documentation en amont.(LOG-2380)

1.37. OPENSHIFT LOGGING 5.3.5

Cette version inclut RHSA-2022:0721 OpenShift Logging Bug Fix Release 5.3.5

1.37.1. Bug fixes

- Avant cette mise à jour, si vous supprimez OpenShift Logging de OpenShift Container Platform, la console web continuait d'afficher un lien vers la page **Logging**. Avec cette mise à jour, la suppression ou la désinstallation d'OpenShift Logging supprime également ce lien.(LOG-2182)

1.37.2. CVE

Exemple 1.9. Cliquez pour agrandir CVEs

- CVE-2020-28491
- CVE-2021-3521
- CVE-2021-3872
- CVE-2021-3984
- CVE-2021-4019

- CVE-2021-4122
- CVE-2021-4192
- CVE-2021-4193
- CVE-2022-0552

1.38. OPENSHIFT LOGGING 5.3.4

Cette version inclut RHBA-2022:0411 OpenShift Logging Bug Fix Release 5.3.4

1.38.1. Bug fixes

- Avant cette mise à jour, les modifications apportées aux tableaux de bord de mesure n'avaient pas encore été déployées parce que le site **cluster-logging-operator** ne comparait pas correctement les cartes de configuration existantes et souhaitées contenant le tableau de bord. Cette mise à jour corrige la logique en ajoutant une valeur de hachage unique aux étiquettes des objets.(LOG-2066)
- Avant cette mise à jour, les pods Elasticsearch ne démarraient pas après une mise à jour avec FIPS activé. Avec cette mise à jour, les pods Elasticsearch démarrent avec succès.(LOG-1974)
- Avant cette mise à jour, elasticsearch générait l'erreur "Unable to create PersistentVolumeClaim due to forbidden : exceeded quota : infra-storage-quota." si le PVC existait déjà. Avec cette mise à jour, elasticsearch vérifie les PVC existants, ce qui résout le problème.(LOG-2127)

1.38.2. CVE

Exemple 1.10. Cliquez pour agrandir CVEs

- CVE-2021-3521
- CVE-2021-3872
- CVE-2021-3984
- CVE-2021-4019
- CVE-2021-4122
- CVE-2021-4155
- CVE-2021-4192
- CVE-2021-4193
- CVE-2022-0185
- CVE-2022-21248
- CVE-2022-21277
- CVE-2022-21282

- CVE-2022-21283
- CVE-2022-21291
- CVE-2022-21293
- CVE-2022-21294
- CVE-2022-21296
- CVE-2022-21299
- CVE-2022-21305
- CVE-2022-21340
- CVE-2022-21341
- CVE-2022-21360
- CVE-2022-21365
- CVE-2022-21366

1.39. OPENSHIFT LOGGING 5.3.3

Cette version inclut RHSA-2022:0227 OpenShift Logging Bug Fix Release 5.3.3

1.39.1. Bug fixes

- Avant cette mise à jour, les modifications apportées aux tableaux de bord n'avaient pas encore été déployées car l'opérateur cluster-logging-operator ne comparait pas correctement les cartes de configuration existantes et souhaitées contenant le tableau de bord. Cette mise à jour corrige la logique en ajoutant une valeur de hachage unique du tableau de bord aux étiquettes des objets.(LOG-2066)
- Cette mise à jour modifie la dépendance de log4j en 2.17.1 pour résoudre la CVE-2021-44832.(LOG-2102)

1.39.2. CVE

Exemple 1.11. Cliquez pour agrandir CVEs

- CVE-2021-27292
  - BZ-1940613
- CVE-2021-44832
  - BZ-2035951

1.40. OPENSHIFT LOGGING 5.3.2

Cette version inclut RHSA-2022:0044 OpenShift Logging Bug Fix Release 5.3.2

1.40.1. Bug fixes

- Avant cette mise à jour, Elasticsearch rejetait les journaux provenant du routeur d'événements en raison d'une erreur d'analyse. Cette mise à jour modifie le modèle de données pour résoudre l'erreur d'analyse. Cependant, en conséquence, les indices précédents peuvent provoquer des avertissements ou des erreurs dans Kibana. Le champ **kubernetes.event.metadata.resourceVersion** provoque des erreurs jusqu'à ce que les index existants soient supprimés ou réindexés. Si ce champ n'est pas utilisé dans Kibana, vous pouvez ignorer les messages d'erreur. Si vous avez une politique de rétention qui supprime les anciens index, la politique finit par supprimer les anciens index et arrête les messages d'erreur. Sinon, réindexez manuellement pour arrêter les messages d'erreur.(LOG-2087)

- Avant cette mise à jour, le tableau de bord OpenShift Logging Dashboard affichait le mauvais pod namespace dans le tableau qui affiche les conteneurs les plus produits et collectés au cours des dernières 24 heures. Avec cette mise à jour, le tableau de bord OpenShift Logging Dashboard affiche le bon espace de noms de pods.(LOG-2051)

- Avant cette mise à jour, si **outputDefaults.elasticsearch.structuredTypeKey** dans l'instance de ressource personnalisée (CR) **ClusterLogForwarder** n'avait pas de clé structurée, la CR remplaçait le secret de sortie par le secret par défaut utilisé pour communiquer avec le magasin de journaux par défaut. Avec cette mise à jour, le secret de sortie défini est correctement utilisé.(LOG-2046)

1.40.2. CVE

Exemple 1.12. Cliquez pour agrandir CVEs

- CVE-2020-36327
  - BZ-1958999
- CVE-2021-45105
  - BZ-2034067
- CVE-2021-3712
- CVE-2021-20321
- CVE-2021-42574

1.41. OPENSHIFT LOGGING 5.3.1

Cette version inclut RHSA-2021:5129 OpenShift Logging Bug Fix Release 5.3.1

1.41.1. Bug fixes

- Avant cette mise à jour, l'image du conteneur Fluentd incluait des outils de construction qui n'étaient pas nécessaires au moment de l'exécution. Cette mise à jour supprime ces outils de l'image.(LOG-1998)

- Avant cette mise à jour, le tableau de bord Logging affichait un graphique CPU vide en raison d'une référence à une métrique non valide. Avec cette mise à jour, le tableau de bord d'enregistrement affiche correctement les graphiques de CPU.(LOG-1925)

- Avant cette mise à jour, le plugin d'exportation Elasticsearch Prometheus compilait les métriques au niveau de l'index à l'aide d'une requête coûteuse qui affectait les performances du nœud Elasticsearch. Cette mise à jour implémente une requête moins coûteuse qui améliore les performances.(LOG-1897)

1.41.2. CVE

Exemple 1.13. Cliquez pour agrandir CVEs

- CVE-2021-21409
  - BZ-1944888
- CVE-2021-37136
  - BZ-2004133
- CVE-2021-37137
  - BZ-2004135
- CVE-2021-44228
  - BZ-2030932
- CVE-2018-25009
- CVE-2018-25010
- CVE-2018-25012
- CVE-2018-25013
- CVE-2018-25014
- CVE-2019-5827
- CVE-2019-13750
- CVE-2019-13751
- CVE-2019-17594
- CVE-2019-17595
- CVE-2019-18218
- CVE-2019-19603
- CVE-2019-20838
- CVE-2020-12762
- CVE-2020-13435

- CVE-2020-14145
- CVE-2020-14155
- CVE-2020-16135
- CVE-2020-17541
- CVE-2020-24370
- CVE-2020-35521
- CVE-2020-35522
- CVE-2020-35523
- CVE-2020-35524
- CVE-2020-36330
- CVE-2020-36331
- CVE-2020-36332
- CVE-2021-3200
- CVE-2021-3426
- CVE-2021-3445
- CVE-2021-3481
- CVE-2021-3572
- CVE-2021-3580
- CVE-2021-3712
- CVE-2021-3800
- CVE-2021-20231
- CVE-2021-20232
- CVE-2021-20266
- CVE-2021-20317
- CVE-2021-22876
- CVE-2021-22898
- CVE-2021-22925
- CVE-2021-27645
- CVE-2021-28153

- CVE-2021-31535
- CVE-2021-33560
- CVE-2021-33574
- CVE-2021-35942
- CVE-2021-36084
- CVE-2021-36085
- CVE-2021-36086
- CVE-2021-36087
- CVE-2021-42574
- CVE-2021-43267
- CVE-2021-43527
- CVE-2021-45046

1.42. OPENSHIFT LOGGING 5.3.0

Cette version inclut RHSA-2021:4627 OpenShift Logging Bug Fix Release 5.3.0

1.42.1. Nouvelles fonctionnalités et améliorations

- Avec cette mise à jour, les options d'autorisation pour le Log Forwarding ont été étendues. Les sorties peuvent désormais être configurées avec SASL, nom d'utilisateur/mot de passe ou TLS.

1.42.2. Bug fixes

- Avant cette mise à jour, si vous transmettiez des logs en utilisant le protocole syslog, la sérialisation d'un hash ruby encodait les paires clé/valeur pour qu'elles contiennent un caractère '⇒' et remplaçait les tabulations par "#11". Cette mise à jour corrige le problème afin que les messages de log soient correctement sérialisés en tant que JSON valide.(LOG-1494)

- Avant cette mise à jour, les journaux d'application n'étaient pas correctement configurés pour être transmis au flux Cloudwatch approprié lorsque la détection d'erreurs multilignes était activée.(LOG-1939)

- Avant cette mise à jour, un changement de nom du collecteur déployé dans la version 5.3 provoquait la génération de l'alerte 'fluentnodedown'.(LOG-1918)

- Avant cette mise à jour, une régression introduite dans une configuration de version antérieure faisait que le collecteur vidait ses messages en mémoire tampon avant l'arrêt, ce qui retardait l'arrêt et le redémarrage des Pods du collecteur. Avec cette mise à jour, fluentd ne vide plus les tampons à l'arrêt, ce qui résout le problème.(LOG-1735)

- Avant cette mise à jour, une régression introduite dans une version antérieure désactivait intentionnellement l'analyse des messages JSON. Cette mise à jour réactive l'analyse JSON. Elle définit également l'entrée de journal "niveau" en fonction du champ "niveau" dans le message

JSON analysé ou en utilisant une expression rationnelle pour extraire une correspondance d'un champ de message.(LOG-1199)

- Avant cette mise à jour, la ressource personnalisée (CR) **ClusterLogging** appliquait la valeur du champ **totalLimitSize** au champ Fluentd **total_limit_size**, même si l'espace tampon requis n'était pas disponible. Avec cette mise à jour, la CR applique la plus petite des deux valeurs **totalLimitSize** ou 'default' au champ Fluentd **total_limit_size**, ce qui résout le problème. (LOG-1776)

1.42.3. Problèmes connus

- Si vous transférez des logs vers un serveur Elasticsearch externe et que vous modifiez ensuite une valeur configurée dans le secret du pipeline, comme le nom d'utilisateur et le mot de passe, le forwarder Fluentd charge le nouveau secret mais utilise l'ancienne valeur pour se connecter à un serveur Elasticsearch externe. Ce problème se produit parce que l'opérateur de journalisation de Red Hat OpenShift ne surveille pas actuellement les secrets pour les changements de contenu.(LOG-1652)
  Comme solution de contournement, si vous changez le secret, vous pouvez forcer les pods Fluentd à se redéployer en entrant :

```
$ oc delete pod -l component=collector
```

1.42.4. Fonctionnalités obsolètes et supprimées

Certaines fonctionnalités disponibles dans les versions précédentes sont devenues obsolètes ou ont été supprimées.

Les fonctionnalités dépréciées sont toujours incluses dans OpenShift Logging et continuent d'être prises en charge ; cependant, elles seront supprimées dans une prochaine version de ce produit et ne sont pas recommandées pour les nouveaux déploiements.

1.42.4.1. Le transfert des journaux utilisant les anciennes méthodes Fluentd et syslog a été supprimé

Dans OpenShift Logging 5.3, les anciennes méthodes de transmission des logs vers Syslog et Fluentd sont supprimées. Les corrections de bugs et le support sont fournis jusqu'à la fin du cycle de vie d'OpenShift Logging 5.2. Après cela, aucune nouvelle amélioration de fonctionnalité n'est apportée.

Utilisez plutôt les méthodes non traditionnelles suivantes :

- Transférer les journaux en utilisant le protocole de transfert Fluentd
- Transmission des journaux à l'aide du protocole syslog

1.42.4.2. Les mécanismes de configuration des anciennes méthodes de transfert ont été supprimés

Dans OpenShift Logging 5.3, le mécanisme de configuration pour la transmission des logs est supprimé : Vous ne pouvez pas transférer les journaux en utilisant la méthode Fluentd et la méthode Syslog. Utilisez les méthodes de transfert de logs standard à la place.

1.42.5. CVE

Exemple 1.14. Cliquez pour agrandir CVEs

- CVE-2018-20673
- CVE-2018-25009
- CVE-2018-25010
- CVE-2018-25012
- CVE-2018-25013
- CVE-2018-25014
- CVE-2019-5827
- CVE-2019-13750
- CVE-2019-13751
- CVE-2019-14615
- CVE-2019-17594
- CVE-2019-17595
- CVE-2019-18218
- CVE-2019-19603
- CVE-2019-20838
- CVE-2020-0427
- CVE-2020-10001
- CVE-2020-12762
- CVE-2020-13435
- CVE-2020-14145
- CVE-2020-14155
- CVE-2020-16135
- CVE-2020-17541
- CVE-2020-24370
- CVE-2020-24502
- CVE-2020-24503
- CVE-2020-24504
- CVE-2020-24586
- CVE-2020-24587

- CVE-2020-24588
- CVE-2020-26139
- CVE-2020-26140
- CVE-2020-26141
- CVE-2020-26143
- CVE-2020-26144
- CVE-2020-26145
- CVE-2020-26146
- CVE-2020-26147
- CVE-2020-27777
- CVE-2020-29368
- CVE-2020-29660
- CVE-2020-35448
- CVE-2020-35521
- CVE-2020-35522
- CVE-2020-35523
- CVE-2020-35524
- CVE-2020-36158
- CVE-2020-36312
- CVE-2020-36330
- CVE-2020-36331
- CVE-2020-36332
- CVE-2020-36386
- CVE-2021-0129
- CVE-2021-3200
- CVE-2021-3348
- CVE-2021-3426
- CVE-2021-3445
- CVE-2021-3481

- CVE-2021-3487
- CVE-2021-3489
- CVE-2021-3564
- CVE-2021-3572
- CVE-2021-3573
- CVE-2021-3580
- CVE-2021-3600
- CVE-2021-3635
- CVE-2021-3659
- CVE-2021-3679
- CVE-2021-3732
- CVE-2021-3778
- CVE-2021-3796
- CVE-2021-3800
- CVE-2021-20194
- CVE-2021-20197
- CVE-2021-20231
- CVE-2021-20232
- CVE-2021-20239
- CVE-2021-20266
- CVE-2021-20284
- CVE-2021-22876
- CVE-2021-22898
- CVE-2021-22925
- CVE-2021-23133
- CVE-2021-23840
- CVE-2021-23841
- CVE-2021-27645
- CVE-2021-28153

- CVE-2021-28950
- CVE-2021-28971
- CVE-2021-29155
- lCVE-2021-29646
- CVE-2021-29650
- CVE-2021-31440
- CVE-2021-31535
- CVE-2021-31829
- CVE-2021-31916
- CVE-2021-33033
- CVE-2021-33194
- CVE-2021-33200
- CVE-2021-33560
- CVE-2021-33574
- CVE-2021-35942
- CVE-2021-36084
- CVE-2021-36085
- CVE-2021-36086
- CVE-2021-36087
- CVE-2021-42574

1.43. JOURNALISATION 5.2.13

Cette version inclut la version 5.2.13 de RHSA-2022:5909-OpenShift Logging Bug Fix .

1.43.1. Bug fixes

- BZ-2100495

1.43.2. CVE

Exemple 1.15. Cliquez pour agrandir CVEs

- CVE-2021-38561
- CVE-2021-40528

- CVE-2022-1271
- CVE-2022-1621
- CVE-2022-1629
- CVE-2022-21540
- CVE-2022-21541
- CVE-2022-22576
- CVE-2022-25313
- CVE-2022-25314
- CVE-2022-27774
- CVE-2022-27776
- CVE-2022-27782
- CVE-2022-29824
- CVE-2022-34169

1.44. JOURNALISATION 5.2.12

Cette version inclut la correction de bogue RHBA-2022:5558-OpenShift Logging Release 5.2.12 .

1.44.1. Bug fixes

Aucun.

1.44.2. CVE

Exemple 1.16. Cliquez pour agrandir CVEs

- CVE-2020-28915
- CVE-2021-40528
- CVE-2022-1271
- CVE-2022-1621
- CVE-2022-1629
- CVE-2022-22576
- CVE-2022-25313
- CVE-2022-25314
- CVE-2022-26691

- CVE-2022-27666
- CVE-2022-27774
- CVE-2022-27776
- CVE-2022-27782
- CVE-2022-29824

1.45. JOURNALISATION 5.2.11

Cette version inclut la version 5.2.11 de RHBA-2022:5012-OpenShift Logging Bug Fix

1.45.1. Bug fixes

- Avant cette mise à jour, les clusters configurés pour effectuer le transfert CloudWatch écrivaient les fichiers journaux rejetés dans le stockage temporaire, ce qui entraînait une instabilité du cluster au fil du temps. Avec cette mise à jour, la sauvegarde de chunk pour CloudWatch a été désactivée, ce qui résout le problème.(LOG-2635)

1.45.2. CVE

Exemple 1.17. Cliquez pour agrandir CVEs

- CVE-2018-25032
- CVE-2020-0404
- CVE-2020-4788
- CVE-2020-13974
- CVE-2020-19131
- CVE-2020-27820
- CVE-2021-0941
- CVE-2021-3612
- CVE-2021-3634
- CVE-2021-3669
- CVE-2021-3737
- CVE-2021-3743
- CVE-2021-3744
- CVE-2021-3752
- CVE-2021-3759

- CVE-2021-3764
- CVE-2021-3772
- CVE-2021-3773
- CVE-2021-4002
- CVE-2021-4037
- CVE-2021-4083
- CVE-2021-4157
- CVE-2021-4189
- CVE-2021-4197
- CVE-2021-4203
- CVE-2021-20322
- CVE-2021-21781
- CVE-2021-23222
- CVE-2021-26401
- CVE-2021-29154
- CVE-2021-37159
- CVE-2021-41617
- CVE-2021-41864
- CVE-2021-42739
- CVE-2021-43056
- CVE-2021-43389
- CVE-2021-43976
- CVE-2021-44733
- CVE-2021-45485
- CVE-2021-45486
- CVE-2022-0001
- CVE-2022-0002
- CVE-2022-0286
- CVE-2022-0322

- CVE-2022-1011
- CVE-2022-1271

1.46. OPENSHIFT LOGGING 5.2.10

Cette version inclut la version 5.2.10 de la correction des bugs de journalisation d'OpenShift]

1.46.1. Bug fixes

- Avant cette mise à jour, certaines sorties de transfert de journaux pouvaient réorganiser les journaux ayant le même horodatage. Avec cette mise à jour, un numéro de séquence a été ajouté à l'enregistrement du journal afin d'ordonner les entrées dont les horodatages correspondent.(LOG-2335)

- Avant cette mise à jour, les clusters comportant un grand nombre d'espaces de noms empêchaient Elasticsearch de servir les requêtes car la liste des espaces de noms atteignait la limite de taille maximale de l'en-tête. Avec cette mise à jour, les en-têtes n'incluent qu'une liste de noms d'espaces de noms, ce qui résout le problème.(LOG-2475)

- Avant cette mise à jour, **system:serviceaccount:openshift-monitoring:prometheus-k8s** avait des privilèges de niveau cluster en tant que **clusterrole** et **clusterrolebinding**. Cette mise à jour restreint **serviceaccount** à l'espace de noms **openshift-logging** avec un rôle et un lien de rôle.(LOG-2480)

- Avant cette mise à jour, le site **cluster-logging-operator** utilisait des rôles et des liaisons à l'échelle du cluster pour établir des autorisations permettant au compte de service Prometheus d'analyser les mesures. Ces autorisations n'étaient créées que lors du déploiement de l'opérateur à l'aide de l'interface de la console et n'existaient pas lorsque l'opérateur était déployé à partir de la ligne de commande. Ceci corrige le problème en rendant l'espace de noms de ce rôle et de cette liaison étendu.(LOG-1972)

1.46.2. CVE

Exemple 1.18. Cliquez pour agrandir CVEs

- CVE-2018-25032
- CVE-2021-4028
- CVE-2021-37136
- CVE-2021-37137
- CVE-2021-43797
- CVE-2022-0778
- CVE-2022-1154
- CVE-2022-1271
- CVE-2022-21426

- CVE-2022-21434
- CVE-2022-21443
- CVE-2022-21476
- CVE-2022-21496
- CVE-2022-21698
- CVE-2022-25636

1.47. OPENSHIFT LOGGING 5.2.9

Cette version inclut RHBA-2022:1375 OpenShift Logging Bug Fix Release 5.2.9]

1.47.1. Bug fixes

- Avant cette mise à jour, la définition d'une tolérance sans clé avec l'opérateur existant entraînait l'impossibilité pour l'opérateur d'effectuer une mise à niveau. Avec cette mise à jour, cette tolérance ne bloque plus l'achèvement de la mise à niveau.(LOG-2304)

1.48. OPENSHIFT LOGGING 5.2.8

Cette version inclut RHSA-2022:0728 OpenShift Logging Bug Fix Release 5.2.8

1.48.1. Bug fixes

- Avant cette mise à jour, si vous supprimez OpenShift Logging de OpenShift Container Platform, la console web continuait d'afficher un lien vers la page **Logging**. Avec cette mise à jour, la suppression ou la désinstallation d'OpenShift Logging supprime également ce lien.(LOG-2180)

1.48.2. CVE

Exemple 1.19. Cliquez pour agrandir CVEs

- CVE-2020-28491
  - BZ-1930423
- CVE-2022-0552
  - BG-2052539

1.49. OPENSHIFT LOGGING 5.2.7

Cette version inclut RHBA-2022:0478 OpenShift Logging Bug Fix Release 5.2.7

1.49.1. Bug fixes

- Avant cette mise à jour, les pods Elasticsearch avec FIPS activé ne démarraient pas après la mise à jour. Avec cette mise à jour, les pods Elasticsearch démarrent avec succès.(LOG-2000)
- Avant cette mise à jour, si une demande de volume persistant (PVC) existait déjà, Elasticsearch générait une erreur, "Unable to create PersistentVolumeClaim due to forbidden : exceeded quota : infra-storage-quota." Avec cette mise à jour, Elasticsearch vérifie les PVC existants, ce qui résout le problème.(LOG-2118)

1.49.2. CVE

Exemple 1.20. Cliquez pour agrandir CVEs

- CVE-2021-3521
- CVE-2021-3872
- CVE-2021-3984
- CVE-2021-4019
- CVE-2021-4122
- CVE-2021-4155
- CVE-2021-4192
- CVE-2021-4193
- CVE-2022-0185

1.50. OPENSHIFT LOGGING 5.2.6

Cette version inclut RHSA-2022:0230 OpenShift Logging Bug Fix Release 5.2.6

1.50.1. Bug fixes

- Avant cette mise à jour, la version n'incluait pas un changement de filtre qui provoquait le plantage de Fluentd. Avec cette mise à jour, le filtre manquant a été corrigé.(LOG-2104)
- Cette mise à jour modifie la dépendance de log4j en 2.17.1 pour résoudre la CVE-2021-44832.(LOG-2101)

1.50.2. CVE

Exemple 1.21. Cliquez pour agrandir CVEs

- CVE-2021-27292
  - BZ-1940613
- CVE-2021-44832
  - BZ-2035951

1.51. OPENSHIFT LOGGING 5.2.5

Cette version inclut RHSA-2022:0043 OpenShift Logging Bug Fix Release 5.2.5

1.51.1. Bug fixes

- Avant cette mise à jour, Elasticsearch rejetait les journaux provenant du routeur d'événements en raison d'une erreur d'analyse. Cette mise à jour modifie le modèle de données pour résoudre l'erreur d'analyse. Cependant, en conséquence, les indices précédents peuvent provoquer des avertissements ou des erreurs dans Kibana. Le champ **kubernetes.event.metadata.resourceVersion** provoque des erreurs jusqu'à ce que les index existants soient supprimés ou réindexés. Si ce champ n'est pas utilisé dans Kibana, vous pouvez ignorer les messages d'erreur. Si vous avez une politique de rétention qui supprime les anciens index, la politique finit par supprimer les anciens index et arrête les messages d'erreur. Sinon, réindexez manuellement pour arrêter les messages d'erreur. LOG-2087)

1.51.2. CVE

Exemple 1.22. Cliquez pour agrandir CVEs

- CVE-2021-3712
- CVE-2021-20321
- CVE-2021-42574
- CVE-2021-45105

1.52. OPENSHIFT LOGGING 5.2.4

Cette version inclut RHSA-2021:5127 OpenShift Logging Bug Fix Release 5.2.4

1.52.1. Bug fixes

- Avant cette mise à jour, les enregistrements envoyés via syslog sérialisaient un hash ruby encodant les paires clé/valeur pour qu'elles contiennent un caractère '⇒', et remplaçaient les tabulations par des "#11". Cette mise à jour sérialise le message correctement en JSON.(LOG-1775)

- Avant cette mise à jour, le plugin d'exportation Elasticsearch Prometheus compilait les métriques au niveau de l'index à l'aide d'une requête coûteuse qui avait un impact sur les performances du nœud Elasticsearch. Cette mise à jour implémente une requête moins coûteuse qui améliore les performances.(LOG-1970)

- Avant cette mise à jour, Elasticsearch rejetait parfois des messages lorsque Log Forwarding était configuré avec plusieurs sorties. Cela se produisait parce que la configuration de l'une des sorties modifiait le contenu du message pour en faire un message unique. Avec cette mise à jour, Log Forwarding duplique les messages pour chaque sortie afin que le traitement spécifique à la sortie n'affecte pas les autres sorties.(LOG-1824)

1.52.2. CVE

Exemple 1.23. Cliquez pour agrandir CVEs

- CVE-2018-25009
- CVE-2018-25010
- CVE-2018-25012
- CVE-2018-25013
- CVE-2018-25014
- CVE-2019-5827
- CVE-2019-13750
- CVE-2019-13751
- CVE-2019-17594
- CVE-2019-17595
- CVE-2019-18218
- CVE-2019-19603
- CVE-2019-20838
- CVE-2020-12762
- CVE-2020-13435
- CVE-2020-14145
- CVE-2020-14155
- CVE-2020-16135
- CVE-2020-17541
- CVE-2020-24370
- CVE-2020-35521
- CVE-2020-35522
- CVE-2020-35523
- CVE-2020-35524
- CVE-2020-36330
- CVE-2020-36331
- CVE-2020-36332
- CVE-2021-3200
- CVE-2021-3426

- CVE-2021-3445
- CVE-2021-3481
- CVE-2021-3572
- CVE-2021-3580
- CVE-2021-3712
- CVE-2021-3800
- CVE-2021-20231
- CVE-2021-20232
- CVE-2021-20266
- CVE-2021-20317
- CVE-2021-21409
- CVE-2021-22876
- CVE-2021-22898
- CVE-2021-22925
- CVE-2021-27645
- CVE-2021-28153
- CVE-2021-31535
- CVE-2021-33560
- CVE-2021-33574
- CVE-2021-35942
- CVE-2021-36084
- CVE-2021-36085
- CVE-2021-36086
- CVE-2021-36087
- CVE-2021-37136
- CVE-2021-37137
- CVE-2021-42574
- CVE-2021-43267
- CVE-2021-43527

- CVE-2021-44228
- CVE-2021-45046

1.53. OPENSHIFT LOGGING 5.2.3

Cette version inclut RHSA-2021:4032 OpenShift Logging Bug Fix Release 5.2.3

1.53.1. Bug fixes

- Avant cette mise à jour, certaines alertes n'incluaient pas d'étiquette d'espace de noms. Cette omission n'est pas conforme aux directives de l'équipe de surveillance d'OpenShift pour l'écriture de règles d'alerte dans OpenShift Container Platform. Avec cette mise à jour, toutes les alertes dans Elasticsearch Operator incluent une étiquette d'espace de noms et suivent toutes les directives pour l'écriture de règles d'alerte dans OpenShift Container Platform.(LOG-1857)

- Avant cette mise à jour, une régression introduite dans une version antérieure désactivait intentionnellement l'analyse des messages JSON. Cette mise à jour réactive l'analyse JSON. Elle définit également l'entrée de journal **level** en fonction du champ **level** dans le message JSON analysé ou en utilisant une expression rationnelle pour extraire une correspondance d'un champ de message.(LOG-1759)

1.53.2. CVE

Exemple 1.24. Cliquez pour agrandir CVEs

- CVE-2021-23369
  - BZ-1948761
- CVE-2021-23383
  - BZ-1956688
- CVE-2018-20673
- CVE-2019-5827
- CVE-2019-13750
- CVE-2019-13751
- CVE-2019-17594
- CVE-2019-17595
- CVE-2019-18218
- CVE-2019-19603
- CVE-2019-20838
- CVE-2020-12762

- CVE-2020-13435
- CVE-2020-14155
- CVE-2020-16135
- CVE-2020-24370
- CVE-2021-3200
- CVE-2021-3426
- CVE-2021-3445
- CVE-2021-3572
- CVE-2021-3580
- CVE-2021-3778
- CVE-2021-3796
- CVE-2021-3800
- CVE-2021-20231
- CVE-2021-20232
- CVE-2021-20266
- CVE-2021-22876
- CVE-2021-22898
- CVE-2021-22925
- CVE-2021-23840
- CVE-2021-23841
- CVE-2021-27645
- CVE-2021-28153
- CVE-2021-33560
- CVE-2021-33574
- CVE-2021-35942
- CVE-2021-36084
- CVE-2021-36085
- CVE-2021-36086
- CVE-2021-36087

1.54. OPENSHIFT LOGGING 5.2.2

Cette version inclut RHBA-2021:3747 OpenShift Logging Bug Fix Release 5.2.2

1.54.1. Bug fixes

- Avant cette mise à jour, la ressource personnalisée (CR) **ClusterLogging** appliquait la valeur du champ **totalLimitSize** au champ Fluentd **total_limit_size**, même si l'espace tampon requis n'était pas disponible. Avec cette mise à jour, la CR applique la plus petite des deux valeurs **totalLimitSize** ou 'default' au champ Fluentd **total_limit_size**, ce qui résout le problème. (LOG-1738)

- Avant cette mise à jour, une régression introduite dans une configuration de version antérieure faisait que le collecteur vidait ses messages en mémoire tampon avant l'arrêt, ce qui créait un retard dans l'arrêt et le redémarrage des pods collecteurs. Avec cette mise à jour, Fluentd ne vide plus les tampons à l'arrêt, ce qui résout le problème.(LOG-1739)

- Avant cette mise à jour, un problème dans les bundle manifests empêchait l'installation de l'Elasticsearch Operator via OLM sur OpenShift Container Platform 4.9. Avec cette mise à jour, une correction des bundle manifests permet à nouveau l'installation et la mise à jour en 4.9.(LOG-1780)

1.54.2. CVE

Exemple 1.25. Cliquez pour agrandir CVEs

- CVE-2020-25648
- CVE-2021-22922
- CVE-2021-22923
- CVE-2021-22924
- CVE-2021-36222
- CVE-2021-37576
- CVE-2021-37750
- CVE-2021-38201

1.55. OPENSHIFT LOGGING 5.2.1

Cette version inclut RHBA-2021:3550 OpenShift Logging Bug Fix Release 5.2.1

1.55.1. Bug fixes

- Avant cette mise à jour, en raison d'un problème dans les scripts du pipeline de diffusion, la valeur du champ **olm.skipRange** restait inchangée à **5.2.0** au lieu de refléter le numéro de diffusion actuel. Cette mise à jour corrige les scripts du pipeline pour mettre à jour la valeur de ce champ lorsque les numéros de version changent.(LOG-1743)

1.55.2. CVE

(Aucun)

1.56. OPENSHIFT LOGGING 5.2.0

Cette version inclut RHBA-2021:3393 OpenShift Logging Bug Fix Release 5.2.0

1.56.1. Nouvelles fonctionnalités et améliorations

- Avec cette mise à jour, vous pouvez transférer les données des journaux vers Amazon CloudWatch, qui assure la surveillance des applications et de l'infrastructure. Pour plus d'informations, voir Transférer les journaux vers Amazon CloudWatch .(LOG-1173)

- Avec cette mise à jour, vous pouvez transférer les données des journaux vers Loki, un système d'agrégation de journaux multitenant, évolutif horizontalement et hautement disponible. Pour plus d'informations, voir Transférer les journaux vers Loki .(LOG-684)

- Avec cette mise à jour, si vous utilisez le protocole de transfert Fluentd pour transférer des données de journal sur une connexion cryptée TLS, vous pouvez désormais utiliser un fichier de clé privée cryptée par mot de passe et spécifier la phrase de passe dans la configuration du Cluster Log Forwarder. Pour plus d'informations, voir Transférer des logs en utilisant le protocole Fluentd forward.(LOG-1525)

- Cette amélioration vous permet d'utiliser un nom d'utilisateur et un mot de passe pour authentifier une connexion de transfert de protocole vers une instance Elasticsearch externe. Par exemple, si vous ne pouvez pas utiliser TLS mutuel (mTLS) parce qu'un tiers exploite l'instance Elasticsearch, vous pouvez utiliser HTTP ou HTTPS et définir un secret contenant le nom d'utilisateur et le mot de passe. Pour plus d'informations, voir Transférer les logs vers une instance Elasticsearch externe.(LOG-1022)

- Avec cette mise à jour, vous pouvez collecter les logs d'audit de la politique réseau OVN pour les transmettre à un serveur de journalisation.(LOG-1526)

- Par défaut, le modèle de données introduit dans OpenShift Container Platform 4.5 donnait aux logs de différents espaces de noms un index unique en commun. Ce changement rendait plus difficile de voir quels espaces de noms produisaient le plus de logs.
  La version actuelle ajoute des métriques d'espace de noms au tableau de bord **Logging** dans la console OpenShift Container Platform. Avec ces métriques, vous pouvez voir quels espaces de noms produisent des logs et combien de logs chaque espace de noms produit pour un timestamp donné.

  Pour voir ces métriques, ouvrez la perspective **Administrator** dans la console web d'OpenShift Container Platform, et naviguez vers **Observe → Dashboards → Logging/Elasticsearch**...(LOG-1680)

- La version actuelle, OpenShift Logging 5.2, permet deux nouvelles mesures : Pour un timestamp ou une durée donnée, vous pouvez voir le total des logs produits ou enregistrés par des conteneurs individuels, et le total des logs collectés par le collecteur. Ces mesures sont étiquetées par espace de noms, pod et nom de conteneur afin que vous puissiez voir combien de journaux chaque espace de noms et pod collecte et produit.(LOG-1213)

1.56.2. Bug fixes

- Avant cette mise à jour, lorsque l'OpenShift Elasticsearch Operator créait des cronjobs de

gestion d'index, il ajoutait la variable d'environnement **POLICY_MAPPING** deux fois, ce qui amenait l'apiserver à signaler la duplication. Cette mise à jour corrige le problème de sorte que la variable d'environnement **POLICY_MAPPING** n'est définie qu'une seule fois par cronjob, et qu'il n'y a pas de duplication signalée par l'apiserver.(LOG-1130)

- Avant cette mise à jour, la suspension d'un cluster Elasticsearch à zéro nœud ne suspendait pas les cronjobs de gestion d'index, ce qui mettait ces cronjobs en backoff maximum. Ensuite, après la suspension du cluster Elasticsearch, ces cronjobs restaient interrompus en raison de l'atteinte du délai maximal. Cette mise à jour résout le problème en suspendant les cronjobs et le cluster.(LOG-1268)

- Avant cette mise à jour, dans le tableau de bord **Logging** de la console OpenShift Container Platform, la liste des 10 premiers conteneurs produisant des logs ne comportait pas l'étiquette "chart namespace" et fournissait un nom de métrique incorrect, **fluentd_input_status_total_bytes_logged**. Avec cette mise à jour, le graphique affiche l'étiquette de l'espace de noms et le nom correct de la métrique, **log_logged_bytes_total**.(LOG-1271)

- Avant cette mise à jour, si un cronjob de gestion d'index se terminait par une erreur, il ne signalait pas le code de sortie de l'erreur : à la place, son statut était "complete." Cette mise à jour résout le problème en signalant les codes de sortie d'erreur des cronjobs de gestion d'index qui se terminent par des erreurs.(LOG-1273)

- Le site **priorityclasses.v1beta1.scheduling.k8s.io** a été supprimé dans la version 1.22 et remplacé par **priorityclasses.v1.scheduling.k8s.io** (**v1beta1** a été remplacé par **v1**). Avant cette mise à jour, des alertes **APIRemovedInNextReleaseInUse** étaient générées pour **priorityclasses** car **v1beta1** était toujours présent. Cette mise à jour résout le problème en remplaçant **v1beta1** par **v1**. L'alerte n'est plus générée. (LOG-1385)

- Auparavant, l'OpenShift Elasticsearch Operator et le Red Hat OpenShift Logging Operator ne disposaient pas de l'annotation requise pour apparaître dans la liste des opérateurs pouvant s'exécuter dans un environnement déconnecté de la console web d'OpenShift Container Platform. Cette mise à jour ajoute l'annotation **operators.openshift.io/infrastructure-features: '["Disconnected"]'** à ces deux opérateurs afin qu'ils apparaissent dans la liste des opérateurs qui s'exécutent dans des environnements déconnectés.(LOG-1420)

- Avant cette mise à jour, les pods Red Hat OpenShift Logging Operator étaient planifiés sur des cœurs de CPU qui étaient réservés aux charges de travail des clients sur des clusters à nœud unique aux performances optimisées. Avec cette mise à jour, les pods de l'opérateur de journalisation de cluster sont planifiés sur les cœurs de CPU corrects.(LOG-1440)

- Avant cette mise à jour, certaines entrées de journal contenaient des octets UTF-8 non reconnus, ce qui amenait Elasticsearch à rejeter les messages et à bloquer l'ensemble de la charge utile mise en mémoire tampon. Avec cette mise à jour, les charges utiles rejetées abandonnent les entrées de journal invalides et soumettent à nouveau les entrées restantes pour résoudre le problème.(LOG-1499)

- Avant cette mise à jour, le pod **kibana-proxy** entrait parfois dans l'état **CrashLoopBackoff** et enregistrait le message suivant : **Invalid configuration: cookie_secret must be 16, 24, or 32 bytes to create an AES cipher when pass_access_token == true or cookie_refresh != 0, but is 29 bytes.** Le nombre exact d'octets peut varier. Avec cette mise à jour, la génération du secret de session Kibana a été corrigée, et le pod kibana-proxy n'entre plus dans l'état **CrashLoopBackoff** à cause de cette erreur. (LOG-1446)

- Avant cette mise à jour, le plugin AWS CloudWatch Fluentd consignait ses appels API AWS dans le journal Fluentd à tous les niveaux de journal, ce qui consommait des ressources de nœuds OpenShift Container Platform supplémentaires. Avec cette mise à jour, le plugin AWS

CloudWatch Fluentd enregistre les appels API AWS uniquement aux niveaux de journalisation "debug" et "trace". Ainsi, au niveau de journalisation par défaut "warn", Fluentd ne consomme pas de ressources de nœuds supplémentaires.(LOG-1071)

- Avant cette mise à jour, le plugin de sécurité Elasticsearch OpenDistro provoquait l'échec des migrations d'index utilisateur. Cette mise à jour résout le problème en fournissant une version plus récente du plugin. Désormais, les migrations d'index se déroulent sans erreur.(LOG-1276)

- Avant cette mise à jour, dans le tableau de bord **Logging** de la console OpenShift Container Platform, la liste des 10 premiers conteneurs produisant des logs manquait de points de données. Cette mise à jour résout le problème et le tableau de bord affiche tous les points de données.(LOG-1353)

- Avant cette mise à jour, si vous ajustiez les performances du log forwarder Fluentd en ajustant les valeurs **chunkLimitSize** et **totalLimitSize**, le message **Setting queued_chunks_limit_size for each buffer to** indiquait des valeurs trop faibles. La mise à jour actuelle corrige ce problème de sorte que ce message signale les valeurs correctes.(LOG-1411)

- Avant cette mise à jour, le plugin de sécurité Kibana OpenDistro provoquait l'échec des migrations d'index d'utilisateurs. Cette mise à jour résout le problème en fournissant une version plus récente du plugin. Désormais, les migrations d'index se déroulent sans erreur.(LOG-1558)

- Avant cette mise à jour, l'utilisation d'un filtre d'entrée d'espace de noms empêchait les journaux de cet espace de noms d'apparaître dans d'autres entrées. Avec cette mise à jour, les journaux sont envoyés à toutes les entrées qui peuvent les accepter.(LOG-1570)

- Avant cette mise à jour, un fichier de licence manquant pour la dépendance **viaq/logerr** provoquait l'abandon des scanners de licence sans succès. Avec cette mise à jour, la dépendance **viaq/logerr** est sous licence Apache 2.0 et les scanners de licence s'exécutent avec succès.(LOG-1590)

- Avant cette mise à jour, une étiquette de brassage incorrecte pour **curator5** dans le pipeline de construction de **elasticsearch-operator-bundle** provoquait l'extraction d'une image épinglée à un SHA1 factice. Avec cette mise à jour, le pipeline de construction utilise la référence **logging-curator5-rhel8** pour **curator5**, ce qui permet aux cronjobs de gestion d'index d'extraire la bonne image de **registry.redhat.io**.(LOG-1624)

- Avant cette mise à jour, un problème avec les permissions de **ServiceAccount** provoquait des erreurs telles que **no permissions for [indices:admin/aliases/get]**. Avec cette mise à jour, une correction des permissions résout le problème.(LOG-1657)

- Avant cette mise à jour, la définition de ressource personnalisée (CRD) pour l'opérateur de journalisation Red Hat OpenShift manquait le type de sortie Loki, ce qui entraînait le rejet de l'objet de ressource personnalisée **ClusterLogForwarder** par le contrôleur d'admission. Avec cette mise à jour, le CRD inclut Loki comme type de sortie afin que les administrateurs puissent configurer **ClusterLogForwarder** pour envoyer les journaux à un serveur Loki. (LOG-1683)

- Avant cette mise à jour, OpenShift Elasticsearch Operator reconciliation of the **ServiceAccounts** écrasait les champs appartenant à des tiers et contenant des secrets. Ce problème provoquait des pics de mémoire et de CPU en raison de la recréation fréquente des secrets. Cette mise à jour résout le problème. Désormais, l'OpenShift Elasticsearch Operator n'écrase plus les champs appartenant à des tiers.(LOG-1714)

- Avant cette mise à jour, dans la définition de la ressource personnalisée (CR) **ClusterLogging**, si vous avez spécifié une valeur **flush_interval** mais n'avez pas défini **flush_mode** à **interval**, l'opérateur de journalisation Red Hat OpenShift a généré une configuration Fluentd.

Cependant, le collecteur Fluentd a généré une erreur lors de l'exécution. Avec cette mise à jour, Red Hat OpenShift Logging Operator valide la définition **ClusterLogging** CR et génère la configuration Fluentd uniquement si les deux champs sont spécifiés.(LOG-1723)

1.56.3. Problèmes connus

- Si vous transférez des logs vers un serveur Elasticsearch externe et que vous modifiez ensuite une valeur configurée dans le secret du pipeline, comme le nom d'utilisateur et le mot de passe, le forwarder Fluentd charge le nouveau secret mais utilise l'ancienne valeur pour se connecter à un serveur Elasticsearch externe. Ce problème se produit parce que l'opérateur de journalisation de Red Hat OpenShift ne surveille pas actuellement les secrets pour les changements de contenu.(LOG-1652)
  Comme solution de contournement, si vous changez le secret, vous pouvez forcer les pods Fluentd à se redéployer en entrant :

  ```
  $ oc delete pod -l component=collector
  ```

1.56.4. Fonctionnalités obsolètes et supprimées

Certaines fonctionnalités disponibles dans les versions précédentes sont devenues obsolètes ou ont été supprimées.

Les fonctionnalités dépréciées sont toujours incluses dans OpenShift Logging et continuent d'être prises en charge ; cependant, elles seront supprimées dans une prochaine version de ce produit et ne sont pas recommandées pour les nouveaux déploiements.

1.56.5. Le transfert des journaux à l'aide des anciennes méthodes Fluentd et syslog a été supprimé

Depuis OpenShift Container Platform 4.6 jusqu'à aujourd'hui, l'envoi de logs en utilisant les méthodes suivantes a été déprécié et sera supprimé dans une prochaine version :

- Transférer les journaux à l'aide de l'ancienne méthode Fluentd
- Transférer les journaux à l'aide de la méthode syslog traditionnelle

Utilisez plutôt les méthodes non traditionnelles suivantes :

- Transférer les journaux en utilisant le protocole de transfert Fluentd
- Transmission des journaux à l'aide du protocole syslog

1.56.6. CVE

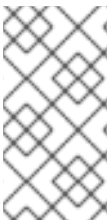
Exemple 1.26. Cliquez pour agrandir CVEs

- CVE-2021-22922
- CVE-2021-22923
- CVE-2021-22924
- CVE-2021-32740

- CVE-2021-36222
- CVE-2021-37750

CHAPITRE 2. ENREGISTREMENT 5.6

2.1. NOTES DE VERSION SUR LA JOURNALISATION 5.6



NOTE

Le sous-système de journalisation pour Red Hat OpenShift est fourni en tant que composant installable, avec un cycle de publication distinct de celui de la plateforme principale OpenShift Container Platform. La politique de cycle de vie de Red Hat OpenShift Container Platform décrit la compatibilité des versions.

NOTE

Le canal **stable** ne fournit des mises à jour que pour la version la plus récente du logiciel d'exploitation. Pour continuer à recevoir les mises à jour des versions antérieures, vous devez changer votre canal d'abonnement pour **stable-X**, où **X** est la version de l'exploitation que vous avez installée.

2.1.1. Journalisation 5.6.4

Cette version inclut la version 5.6.4 de la correction des bugs de journalisation d'OpenShift .

2.1.1.1. Bug fixes

- Avant cette mise à jour, lorsque LokiStack était déployé comme magasin de logs, les logs générés par les pods Loki étaient collectés et envoyés à LokiStack. Avec cette mise à jour, les logs générés par Loki sont exclus de la collecte et ne seront pas stockés.(LOG-3280)

- Avant cette mise à jour, lorsque l'éditeur de requêtes de la page Logs de l'OpenShift Web Console était vide, les menus déroulants ne s'affichaient pas. Avec cette mise à jour, si une requête vide est tentée, un message d'erreur s'affiche et les menus déroulants se remplissent maintenant comme prévu.(LOG-3454)

- Avant cette mise à jour, lorsque l'option **tls.insecureSkipVerify** était définie sur **true**, l'opérateur de journalisation de cluster générait une configuration incorrecte. En conséquence, l'opérateur n'envoyait pas de données à Elasticsearch lorsqu'il tentait d'ignorer la validation du certificat. Avec cette mise à jour, le Cluster Logging Operator génère une configuration TLS correcte même lorsque **tls.insecureSkipVerify** est activé. Par conséquent, les données peuvent être envoyées avec succès à Elasticsearch même lorsque l'on tente d'ignorer la validation du certificat.(LOG-3475)

- Avant cette mise à jour, lorsque l'analyse structurée était activée et que les messages étaient transmis à plusieurs destinations, ils n'étaient pas copiés en profondeur. Par conséquent, certains des journaux reçus incluaient le message structuré, tandis que d'autres ne le faisaient pas. Avec cette mise à jour, la génération de configuration a été modifiée pour copier en profondeur les messages avant l'analyse JSON. Par conséquent, tous les messages reçus contiennent désormais des messages structurés, même lorsqu'ils sont transmis à plusieurs destinations.(LOG-3640)

- Avant cette mise à jour, si le champ **collection** contenait **{}**, l'opérateur pouvait se bloquer. Avec cette mise à jour, l'opérateur ignorera cette valeur, ce qui lui permettra de continuer à fonctionner sans interruption.(LOG-3733)

- Avant cette mise à jour, l'attribut **nodeSelector** pour le composant Gateway de LokiStack n'avait aucun effet. Avec cette mise à jour, l'attribut **nodeSelector** fonctionne comme prévu.(LOG-3783)

- Avant cette mise à jour, la configuration statique de la liste des membres de LokiStack reposait uniquement sur des réseaux IP privés. Par conséquent, lorsque le réseau de pods du cluster OpenShift Container Platform était configuré avec une plage d'IP publique, les pods LokiStack se bloquaient. Avec cette mise à jour, l'administrateur de LokiStack a maintenant la possibilité d'utiliser le réseau de pods pour la configuration de la liste des membres. Cela résout le problème et empêche les pods LokiStack d'entrer dans un état de crashloop lorsque le réseau de pods du cluster OpenShift Container Platform est configuré avec une plage d'IP publique.(LOG-3814)

- Avant cette mise à jour, si le champ **tls.insecureSkipVerify** était défini sur **true**, l'opérateur de journalisation de cluster générait une configuration incorrecte. Par conséquent, l'opérateur n'envoyait pas de données à Elasticsearch lorsqu'il tentait d'ignorer la validation du certificat. Avec cette mise à jour, l'opérateur génère une configuration TLS correcte même lorsque **tls.insecureSkipVerify** est activé. Par conséquent, les données peuvent être envoyées avec succès à Elasticsearch même lorsque l'on tente d'ignorer la validation du certificat.(LOG-3838)

- Avant cette mise à jour, si le Cluster Logging Operator (CLO) était installé sans l'Elasticsearch Operator, le pod CLO affichait en permanence un message d'erreur lié à la suppression d'Elasticsearch. Avec cette mise à jour, le CLO effectue désormais des vérifications supplémentaires avant d'afficher des messages d'erreur. Par conséquent, les messages d'erreur liés à la suppression d'Elasticsearch ne sont plus affichés en l'absence de l'opérateur Elasticsearch.(LOG-3763)

2.1.1.2. CVE

- CVE-2022-4304
- CVE-2022-4450
- CVE-2023-0215
- CVE-2023-0286
- CVE-2023-0767
- CVE-2023-23916

2.1.2. Journalisation 5.6.3

Cette version inclut la version 5.6.3 de la correction des bugs de journalisation d'OpenShift .

2.1.2.1. Bug fixes

- Avant cette mise à jour, l'opérateur stockait les informations relatives au secret du locataire de la passerelle dans une carte de configuration. Avec cette mise à jour, l'opérateur stocke ces informations dans un secret.(LOG-3717)

- Avant cette mise à jour, le collecteur Fluentd ne capturait pas les événements de connexion OAuth stockés dans **/var/log/auth-server/audit.log**. Avec cette mise à jour, Fluentd capture ces événements de connexion OAuth, ce qui résout le problème.(LOG-3729)

2.1.2.2. CVE

- CVE-2020-10735
- CVE-2021-28861
- CVE-2022-2873
- CVE-2022-4415
- CVE-2022-40897
- CVE-2022-41222
- CVE-2022-43945
- CVE-2022-45061
- CVE-2022-48303

2.1.3. Journalisation 5.6.2

Cette version inclut la version 5.6.2 de la correction des bugs de journalisation d'OpenShift .

2.1.3.1. Bug fixes

- Avant cette mise à jour, le collecteur ne définissait pas correctement les champs **level** en fonction de la priorité des journaux systemd. Avec cette mise à jour, les champs **level** sont définis correctement.(LOG-3429)
- Avant cette mise à jour, l'Opérateur générait incorrectement des avertissements d'incompatibilité sur OpenShift Container Platform 4.12 ou plus récent. Avec cette mise à jour, la valeur de la version max OpenShift Container Platform de l'Opérateur a été corrigée, ce qui résout le problème.(LOG-3584)
- Avant cette mise à jour, la création d'une ressource personnalisée (CR) **ClusterLogForwarder** avec une valeur de sortie de **default** ne générait aucune erreur. Avec cette mise à jour, un avertissement d'erreur indiquant que cette valeur n'est pas valide est généré de manière appropriée.(LOG-3437)
- Avant cette mise à jour, lorsque la ressource personnalisée (CR) **ClusterLogForwarder** avait plusieurs pipelines configurés avec une sortie définie comme **default**, les pods collecteurs redémarraient. Avec cette mise à jour, la logique de validation des sorties a été corrigée, ce qui résout le problème.(LOG-3559)
- Avant cette mise à jour, les pods collecteurs redémarraient après avoir été créés. Avec cette mise à jour, le collecteur déployé ne redémarre pas de lui-même.(LOG-3608)
- Avant cette mise à jour, les versions des correctifs supprimaient les versions précédentes des opérateurs du catalogue. Cela rendait l'installation des anciennes versions impossible. Cette mise à jour modifie les configurations des paquets de sorte que les versions précédentes de la même version mineure restent dans le catalogue.(LOG-3635)

2.1.3.2. CVE

- CVE-2022-23521
- CVE-2022-40303

- CVE-2022-40304
- CVE-2022-41903
- CVE-2022-47629
- CVE-2023-21835
- CVE-2023-21843

2.1.4. Journalisation 5.6.1

Cette version inclut la version 5.6.1 de la correction des bugs de journalisation d'OpenShift .

2.1.4.1. Bug fixes

- Avant cette mise à jour, le compacteur signalait des erreurs de certificat TLS lors des communications avec l'interrogateur lorsque la rétention était active. Avec cette mise à jour, le compacteur et l'interrogateur ne communiquent plus de manière erronée via HTTP.(LOG-3494)
- Avant cette mise à jour, l'opérateur Loki ne réessayait pas de définir l'état de **LokiStack** CR, ce qui entraînait des informations d'état périmées. Avec cette mise à jour, l'opérateur réessaie de mettre à jour les informations d'état en cas de conflit.(LOG-3496)
- Avant cette mise à jour, le serveur Webhook de l'opérateur Loki provoquait des erreurs TLS lorsque l'opérateur **kube-apiserver-operator** vérifiait la validité du webhook. Avec cette mise à jour, l'ICP du webhook de l'opérateur Loki est gérée par le gestionnaire du cycle de vie de l'opérateur (OLM), ce qui résout le problème.(LOG-3510)
- Avant cette mise à jour, le LokiStack Gateway Labels Enforcer générait des erreurs d'analyse pour les requêtes LogQL valides lors de l'utilisation de filtres d'étiquettes combinés avec des expressions booléennes. Avec cette mise à jour, l'implémentation LogQL de LokiStack prend en charge les filtres d'étiquettes avec des expressions booléennes et résout le problème.(LOG-3441),(LOG-3397)
- Avant cette mise à jour, les enregistrements écrits dans Elasticsearch échouaient si plusieurs clés d'étiquettes avaient le même préfixe et si certaines clés comportaient des points. Avec cette mise à jour, les traits de soulignement remplacent les points dans les clés d'étiquettes, ce qui résout le problème.(LOG-3463)
- Avant cette mise à jour, l'opérateur **Red Hat OpenShift Logging** n'était pas disponible pour les clusters OpenShift Container Platform 4.10 en raison d'une incompatibilité entre la console OpenShift Container Platform et le plugin logging-view. Avec cette mise à jour, le plugin est correctement intégré à la console d'administration d'OpenShift Container Platform 4.10.(LOG-3447)
- Avant cette mise à jour, la réconciliation de la ressource personnalisée **ClusterLogForwarder** signalait de manière incorrecte un état dégradé des pipelines qui font référence au logstore par défaut. Avec cette mise à jour, le pipeline est validé correctement.(LOG-3477)

2.1.4.2. CVE

- CVE-2021-46848
- CVE-2022-3821

- CVE-2022-35737
- CVE-2022-42010
- CVE-2022-42011
- CVE-2022-42012
- CVE-2022-42898
- CVE-2022-43680
- CVE-2021-35065
- CVE-2022-46175

2.1.5. Journalisation 5.6.0

Cette version inclut la version 5.6 d'OpenShift Logging.

2.1.5.1. Avis de dépréciation

Dans la version 5.6 de Logging, Fluentd est obsolète et il est prévu de le supprimer dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Comme alternative à Fluentd, vous pouvez utiliser Vector.

2.1.5.2. Améliorations

- Avec cette mise à jour, la journalisation est conforme aux politiques cryptographiques à l'échelle du cluster d'OpenShift Container Platform.(LOG-895)
- Avec cette mise à jour, vous pouvez déclarer des politiques de rétention par locataire, par flux et globales via la ressource personnalisée LokiStack, classées par ordre de priorité.(LOG-2695)
- Avec cette mise à jour, Splunk est une option de sortie disponible pour le transfert de logs.(LOG-2913)
- Avec cette mise à jour, Vector remplace Fluentd comme collecteur par défaut.(LOG-2222)
- Avec cette mise à jour, le rôle **Developer** peut accéder aux journaux de charge de travail par projet auxquels ils sont affectés dans le plugin Log Console sur les clusters exécutant OpenShift Container Platform 4.11 et plus.(LOG-3388)
- Avec cette mise à jour, les logs de n'importe quelle source contiennent un champ **openshift.cluster_id**, l'identifiant unique du cluster dans lequel l'Opérateur est déployé. Vous pouvez visualiser la valeur de **clusterID** à l'aide de la commande ci-dessous. (LOG-2715)

```
$ oc get clusterversion/version -o jsonpath='{.spec.clusterID}{"\n"}'
```

2.1.5.3. Problèmes connus

- Avant cette mise à jour, Elasticsearch rejetait les journaux si plusieurs clés de label avaient le même préfixe et si certaines clés incluaient le caractère **.**. Cette mise à jour corrige la limitation d'Elasticsearch en remplaçant **.** dans les clés d'étiquettes par _. Pour contourner ce problème,

supprimez les étiquettes qui provoquent des erreurs ou ajoutez un espace de noms à l'étiquette.(LOG-3463)

2.1.5.4. Bug fixes

- Avant cette mise à jour, si vous supprimiez la ressource personnalisée Kibana, la console web de OpenShift Container Platform continuait à afficher un lien vers Kibana. Avec cette mise à jour, la suppression de la ressource personnalisée Kibana supprime également ce lien.(LOG-2993)

- Avant cette mise à jour, un utilisateur n'était pas en mesure de voir les journaux d'application des espaces de noms auxquels il avait accès. Avec cette mise à jour, l'opérateur Loki crée automatiquement un rôle de cluster et un lien de rôle de cluster permettant aux utilisateurs de lire les journaux d'application.(LOG-3072)

- Avant cette mise à jour, l'opérateur supprimait toutes les sorties personnalisées définies dans la ressource personnalisée **ClusterLogForwarder** lorsqu'il utilisait LokiStack comme stockage de logs par défaut. Avec cette mise à jour, l'opérateur fusionne les sorties personnalisées avec les sorties par défaut lors du traitement de la ressource personnalisée **ClusterLogForwarder**.(LOG-3090)

- Avant cette mise à jour, la clé de l'autorité de certification était utilisée comme nom de volume pour le montage de l'autorité de certification dans Loki, ce qui provoquait des états d'erreur lorsque la clé de l'autorité de certification comprenait des caractères non conformes, tels que des points. Avec cette mise à jour, le nom de volume est normalisé à une chaîne interne, ce qui résout le problème.(LOG-3331)

- Avant cette mise à jour, une valeur par défaut définie dans la définition des ressources personnalisées de LokiStack entraînait l'impossibilité de créer une instance de LokiStack sans **ReplicationFactor** de **1**. Avec cette mise à jour, l'opérateur définit la valeur réelle de la taille utilisée.(LOG-3296)

- Avant cette mise à jour, Vector analysait le champ message lorsque l'analyse JSON était activée sans définir les valeurs **structuredTypeKey** ou **structuredTypeName**. Avec cette mise à jour, une valeur est requise pour **structuredTypeKey** ou **structuredTypeName** lors de l'écriture de journaux structurés dans Elasticsearch.(LOG-3195)

- Avant cette mise à jour, le composant de création de secret de l'Elasticsearch Operator modifiait constamment les secrets internes. Avec cette mise à jour, le secret existant est correctement géré.(LOG-3161)

- Avant cette mise à jour, l'opérateur pouvait entrer dans une boucle de suppression et de recréation du daemonset du collecteur pendant que les déploiements Elasticsearch ou Kibana changeaient d'état. Avec cette mise à jour, une correction dans la gestion de l'état de l'opérateur résout le problème.(LOG-3157)

- Avant cette mise à jour, Kibana avait un délai d'expiration du cookie OAuth fixe **24h**, ce qui entraînait des erreurs 401 dans Kibana chaque fois que le champ **accessTokenInactivityTimeout** était défini sur une valeur inférieure à **24h**. Avec cette mise à jour, le délai d'expiration du cookie OAuth de Kibana se synchronise sur le champ **accessTokenInactivityTimeout**, avec une valeur par défaut de **24h**.(LOG-3129)

- Avant cette mise à jour, le modèle général des opérateurs pour le rapprochement des ressources consistait à essayer de créer un objet avant d'essayer de l'obtenir ou de le mettre à jour, ce qui entraînait des réponses HTTP 409 constantes après la création. Avec cette mise à jour, les opérateurs tentent d'abord de récupérer un objet et ne le créent ou ne le mettent à jour que s'il est manquant ou différent de ce qui a été spécifié.(LOG-2919)

- Avant cette mise à jour, les champs **.level** et `.structure.level` de Fluentd pouvaient contenir des valeurs différentes. Avec cette mise à jour, les valeurs sont les mêmes pour chaque champ.(LOG-2819)

- Avant cette mise à jour, l'opérateur n'attendait pas que le groupe d'autorités de certification de confiance soit peuplé et déployait le collecteur une deuxième fois une fois le groupe mis à jour. Avec cette mise à jour, l'opérateur attend brièvement de voir si le groupe a été peuplé avant de poursuivre le déploiement du collecteur.(LOG-2789)

- Avant cette mise à jour, les informations de télémétrie d'enregistrement apparaissaient deux fois lors de l'examen des métriques. Avec cette mise à jour, les informations de télémétrie s'affichent comme prévu.(LOG-2315)

- Avant cette mise à jour, les logs de Fluentd pod contenaient un message d'avertissement après avoir activé l'ajout de l'analyse JSON. Avec cette mise à jour, ce message d'avertissement n'apparaît plus.(LOG-1806)

- Avant cette mise à jour, le script **must-gather** ne s'exécutait pas car **oc** a besoin d'un dossier avec des droits d'écriture pour construire son cache. Avec cette mise à jour, **oc** a les droits d'écriture sur un dossier et le script **must-gather** s'exécute correctement. (LOG-3446)

- Avant cette mise à jour, le SCC du collecteur de journaux pouvait être remplacé par d'autres SCC sur le cluster, rendant le collecteur inutilisable. Cette mise à jour définit la priorité du SCC du collecteur de journaux de manière à ce qu'il soit prioritaire sur les autres.(LOG-3235)

- Avant cette mise à jour, il manquait à Vector le champ **sequence**, qui a été ajouté à fluentd pour pallier le manque de précision des nanosecondes. Avec cette mise à jour, le champ **openshift.sequence** a été ajouté aux journaux d'événements. (LOG-3106)

2.1.5.5. CVE

- CVE-2020-36518
- CVE-2021-46848
- CVE-2022-2879
- CVE-2022-2880
- CVE-2022-27664
- CVE-2022-32190
- CVE-2022-35737
- CVE-2022-37601
- CVE-2022-41715
- CVE-2022-42003
- CVE-2022-42004
- CVE-2022-42010
- CVE-2022-42011

- CVE-2022-42012
- CVE-2022-42898
- CVE-2022-43680

2.2. DÉMARRER AVEC LA JOURNALISATION 5.6

Cette vue d'ensemble du processus de déploiement de la journalisation est fournie à titre de référence. Elle ne remplace pas la documentation complète. Pour les nouvelles installations, les sites **Vector** et **LokiStack** sont recommandés.

NOTE

À partir de la version 5.5, vous pouvez choisir entre les implémentations de collecteurs **Fluentd** ou **Vector** et les magasins de journaux **Elasticsearch** ou **LokiStack**. La documentation relative à la journalisation est en cours de mise à jour afin de refléter ces changements de composants sous-jacents.

NOTE

Le sous-système de journalisation pour Red Hat OpenShift est fourni en tant que composant installable, avec un cycle de publication distinct de celui de la plateforme principale OpenShift Container Platform. La politique de cycle de vie de Red Hat OpenShift Container Platform décrit la compatibilité des versions.

Conditions préalables

- Préférence LogStore : **Elasticsearch** ou **LokiStack**
- Préférence de mise en œuvre du collecteur : **Fluentd** ou **Vector**
- Informations d'identification pour vos sorties de transfert de logs

NOTE

À partir de la version 5.4.3 de Logging, l'Elasticsearch Operator est obsolète et il est prévu de le supprimer dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Au lieu d'utiliser l'opérateur Elasticsearch pour gérer le stockage des journaux par défaut, vous pouvez utiliser l'opérateur Loki.

1. Installez l'opérateur pour le logstore que vous souhaitez utiliser.
   - Pour **Elasticsearch**, installez le site **OpenShift Elasticsearch Operator**.
   - Pour **LokiStack**, installez le site **Loki Operator**.
     - Créer une instance de ressource personnalisée (CR) à l'adresse **LokiStack**.
2. Installer le site **Red Hat OpenShift Logging Operator**.
3. Créer une instance de ressource personnalisée (CR) à l'adresse **ClusterLogging**.
   a. Sélectionnez la mise en œuvre de votre collecteur

a. Sélectionnez la mise en œuvre de votre collecteur.

NOTE

À partir de la version 5.6 de l'exploitation forestière, Fluentd est obsolète et il est prévu qu'il soit supprimé dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Comme alternative à Fluentd, vous pouvez utiliser Vector.

4. Créer une instance de ressource personnalisée (CR) à l'adresse **ClusterLogForwarder**.

5. Créer un secret pour le pipeline de sortie sélectionné.

2.3. ADMINISTRATION DU DÉPLOIEMENT DE LA JOURNALISATION

Le sous-système de journalisation se compose des éléments logiques suivants :

- **Collector** - Lit les données d'enregistrement des conteneurs sur chaque nœud et les transmet aux sorties configurées.
- **Store** - Stocke les données du journal en vue de leur analyse ; c'est la sortie par défaut du transitaire.
- **Visualization** - Interface graphique pour la recherche, l'interrogation et la visualisation des journaux stockés.

Ces composants sont gérés par des opérateurs et des fichiers YAML de ressources personnalisées (CR).

Le sous-système de journalisation de Red Hat OpenShift collecte les journaux des conteneurs et des nœuds. Ceux-ci sont classés par type :

- **application** - Journaux de conteneurs générés par des conteneurs ne faisant pas partie de l'infrastructure.
- **infrastructure** - Les journaux des conteneurs des espaces de noms **kube-*** et **openshift-***, et les journaux des nœuds de **journald**.
- **audit** - Journaux provenant de **auditd**, **kube-apiserver**, **openshift-apiserver**, et **ovn** si l'option est activée.

Le collecteur de logs est un daemonset qui déploie des pods sur chaque nœud d'OpenShift Container Platform. Les journaux du système et de l'infrastructure sont générés par les messages de journald du système d'exploitation, de l'exécution du conteneur et d'OpenShift Container Platform.

Les journaux de conteneurs sont générés par les conteneurs qui s'exécutent dans des pods sur le cluster. Chaque conteneur génère un flux de journaux distinct. Le collecteur recueille les journaux de ces sources et les transmet en interne ou en externe, comme configuré dans la ressource personnalisée **ClusterLogForwarder**.

2.4. ADMINISTRATION DU DÉPLOIEMENT DE LA JOURNALISATION

2.4.1. Déployer Red Hat OpenShift Logging Operator à l'aide de la console web

Vous pouvez utiliser la console web de OpenShift Container Platform pour déployer Red Hat OpenShift Logging Operator.

CONDITIONS PRÉALABLES

Le sous-système de journalisation pour Red Hat OpenShift est fourni en tant que composant installable, avec un cycle de publication distinct de celui de la plateforme principale OpenShift Container Platform. La politique de cycle de vie de Red Hat OpenShift Container Platform décrit la compatibilité des versions.

Procédure

Pour déployer Red Hat OpenShift Logging Operator à l'aide de la console web OpenShift Container Platform :

1. Installez l'opérateur de journalisation Red Hat OpenShift :
   a. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
   b. Tapez **Logging** dans le champ **Filter by keyword**.
   c. Choisissez **Red Hat OpenShift Logging** dans la liste des opérateurs disponibles et cliquez sur **Install**.
   d. Sélectionnez **stable** ou **stable-5.y** comme **Update Channel**.

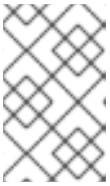
NOTE

Le canal **stable** ne fournit des mises à jour que pour la version la plus récente du logiciel d'exploitation. Pour continuer à recevoir les mises à jour des versions antérieures, vous devez changer votre canal d'abonnement pour **stable-X**, où **X** est la version de l'exploitation que vous avez installée.

   e. Assurez-vous que **A specific namespace on the cluster**est sélectionné sous **Installation Mode**.
   f. Assurez-vous que **Operator recommended namespace** est **openshift-logging** sous **Installed Namespace**.
   g. Sélectionnez **Enable Operator recommended cluster monitoring on this Namespace**
   h. Sélectionnez une option pour **Update approval**.
      - L'option **Automatic** permet à Operator Lifecycle Manager (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible.
      - L'option **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.
   i. Sélectionnez **Enable** ou **Disable** pour le plugin Console.
   j. Cliquez sur **Install**.
2. Vérifiez que le site **Red Hat OpenShift Logging Operator** est installé en passant à la page **Operators → Installed Operators**.

a. Assurez-vous que **Red Hat OpenShift Logging** est listé dans le projet **openshift-logging** avec un **Status** de **Succeeded**.

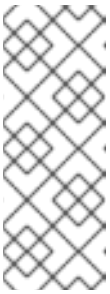
3. Créer une instance **ClusterLogging**.



NOTE

La vue du formulaire de la console web ne comprend pas toutes les options disponibles. Il est recommandé d'utiliser le site **YAML view** pour compléter votre installation.

a. Dans la section **collection**, sélectionnez une implémentation de collecteur.



NOTE

À partir de la version 5.6 de l'exploitation forestière, Fluentd est obsolète et il est prévu qu'il soit supprimé dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Comme alternative à Fluentd, vous pouvez utiliser Vector.

b. Dans la section **logStore**, sélectionnez un type.

NOTE

À partir de la version 5.4.3 de Logging, l'Elasticsearch Operator est obsolète et il est prévu de le supprimer dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Au lieu d'utiliser l'opérateur Elasticsearch pour gérer le stockage des journaux par défaut, vous pouvez utiliser l'opérateur Loki.

c. Cliquez sur **Create**.

2.4.2. Déploiement de l'opérateur Loki à l'aide de la console web

Vous pouvez utiliser la console web d'OpenShift Container Platform pour installer l'opérateur Loki.

Conditions préalables

- Log Store pris en charge (AWS S3, Google Cloud Storage, Azure, Swift, Minio, OpenShift Data Foundation)

Procédure

Pour installer l'opérateur Loki à l'aide de la console web d'OpenShift Container Platform :

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
2. Tapez **Loki** dans le champ **Filter by keyword**.

a. Choisissez **Loki Operator** dans la liste des opérateurs disponibles et cliquez sur **Install**.

3. Sélectionnez **stable** ou **stable-5.y** comme **Update Channel**.

NOTE

Le canal **stable** ne fournit des mises à jour que pour la version la plus récente du logiciel d'exploitation. Pour continuer à recevoir les mises à jour des versions antérieures, vous devez changer votre canal d'abonnement pour **stable-X**, où **X** est la version de l'exploitation que vous avez installée.

4. Assurez-vous que **All namespaces on the cluster**est sélectionné sous **Installation Mode**.

5. Assurez-vous que **openshift-operators-redhat** est sélectionné sous **Installed Namespace**.

6. Sélectionnez **Enable Operator recommended cluster monitoring on this Namespace** Cette option définit l'étiquette **openshift.io/cluster-monitoring: "true"** dans l'objet Namespace. Vous devez sélectionner cette option pour vous assurer que la surveillance des clusters récupère l'espace de noms **openshift-operators-redhat**.

7. Sélectionnez une option pour **Update approval**.

   - L'option **Automatic** permet à Operator Lifecycle Manager (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible.

   - L'option **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.

8. Cliquez sur **Install**.

9. Vérifiez que le site **LokiOperator** est installé en passant à la page **Operators → Installed Operators**.

   a. Veillez à ce que **LokiOperator** soit listé avec **Status** et **Succeeded** dans tous les projets.

10. Créez un fichier YAML **Secret** qui utilise les champs **access_key_id** et **access_key_secret** pour spécifier vos informations d'identification et **bucketnames**, **endpoint**, et **region** pour définir l'emplacement de stockage de l'objet. AWS est utilisé dans l'exemple suivant :

```
apiVersion: v1
kind: Secret
metadata:
  name: logging-loki-s3
  namespace: openshift-logging
stringData:
  access_key_id: AKIAIOSFODNN7EXAMPLE
  access_key_secret: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
  bucketnames: s3-bucket-name
  endpoint: https://s3.eu-central-1.amazonaws.com
  region: eu-central-1
```

11. Sélectionnez **Create instance** sous LokiStack dans l'onglet **Details**. Sélectionnez ensuite **YAML view**. Collez le modèle suivant, en remplaçant les valeurs le cas échéant.

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki (1)
```

```
  namespace: openshift-logging
spec:
  size: 1x.small 2
  storage:
    schemas:
    - version: v12
      effectiveDate: '2022-06-01'
    secret:
      name: logging-loki-s3 3
      type: s3 4
  storageClassName: <storage_class_name> 5
  tenants:
    mode: openshift-logging
```

1. Le nom doit être **logging-loki**.
2. Sélectionnez la taille de déploiement de votre Loki.
3. Définissez le secret utilisé pour le stockage des journaux.
4. Définir le type de stockage correspondant.
5. Saisissez le nom d'une classe de stockage existante pour le stockage temporaire. Pour de meilleures performances, spécifiez une classe de stockage qui alloue des blocs de stockage. Les classes de stockage disponibles pour votre cluster peuvent être répertoriées à l'aide de **oc get storageclasses**.

    a. Appliquer la configuration :

    ```
    oc apply -f logging-loki.yaml
    ```

12. Créer ou modifier un CR **ClusterLogging**:

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  managementState: Managed
  logStore:
    type: lokistack
    lokistack:
      name: logging-loki
    collection:
      type: vector
```

a. Appliquer la configuration :

```
oc apply -f cr-lokistack.yaml
```

2.4.3. Installation à partir d'OperatorHub en utilisant le CLI

Au lieu d'utiliser la console web de OpenShift Container Platform, vous pouvez installer un Operator depuis OperatorHub en utilisant le CLI. Utilisez la commande **oc** pour créer ou mettre à jour un objet **Subscription**.

Conditions préalables

- Accès à un cluster OpenShift Container Platform à l'aide d'un compte disposant des autorisations **cluster-admin**.
- Installez la commande **oc** sur votre système local.

Procédure

1. Voir la liste des opérateurs disponibles pour la grappe à partir d'OperatorHub :

```
$ oc get packagemanifests -n openshift-marketplace
```

Exemple de sortie

```
NAME                              CATALOG               AGE
3scale-operator                   Red Hat Operators     91m
advanced-cluster-management       Red Hat Operators     91m
amq7-cert-manager                 Red Hat Operators     91m
...
couchbase-enterprise-certified    Certified Operators   91m
crunchy-postgres-operator         Certified Operators   91m
mongodb-enterprise                Certified Operators   91m
...
etcd                              Community Operators   91m
jaeger                            Community Operators   91m
kubefed                           Community Operators   91m
...
```

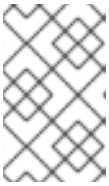
Notez le catalogue de l'opérateur souhaité.

2. Inspectez l'opérateur de votre choix pour vérifier les modes d'installation pris en charge et les canaux disponibles :

```
oc describe packagemanifests <operator_name> -n openshift-marketplace
```

3. Un groupe d'opérateurs, défini par un objet **OperatorGroup**, sélectionne des espaces de noms cibles dans lesquels générer l'accès RBAC requis pour tous les opérateurs dans le même espace de noms que le groupe d'opérateurs.
L'espace de noms auquel vous abonnez l'opérateur doit avoir un groupe d'opérateurs qui correspond au mode d'installation de l'opérateur, soit le mode **AllNamespaces** ou **SingleNamespace**. Si l'opérateur que vous avez l'intention d'installer utilise le mode **AllNamespaces**, l'espace de noms **openshift-operators** dispose déjà d'un groupe d'opérateurs approprié.

   Cependant, si l'opérateur utilise le mode **SingleNamespace** et que vous n'avez pas déjà un groupe d'opérateurs approprié en place, vous devez en créer un.



NOTE

La version console web de cette procédure gère la création des objets **OperatorGroup** et **Subscription** automatiquement dans les coulisses lorsque vous choisissez le mode **SingleNamespace**.

a. Créez un fichier YAML de l'objet **OperatorGroup**, par exemple **operatorgroup.yaml**:

Exemple d'objet OperatorGroup

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <operatorgroup_name>
  namespace: <namespace>
spec:
  targetNamespaces:
  - <namespace>
```

b. Créer l'objet **OperatorGroup**:

```
$ oc apply -f operatorgroup.yaml
```

4. Créez un fichier YAML de l'objet **Subscription** pour abonner un espace de noms à un opérateur, par exemple **sub.yaml**:

Exemple d'objet Subscription

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: <subscription_name>
  namespace: openshift-operators (1)
spec:
  channel: <channel_name> (2)
  name: <operator_name> (3)
  source: redhat-operators (4)
  sourceNamespace: openshift-marketplace (5)
  config:
    env: (6)
    - name: ARGS
      value: "-v=10"
    envFrom: (7)
    - secretRef:
        name: license-secret
    volumes: (8)
    - name: <volume_name>
      configMap:
        name: <configmap_name>
    volumeMounts: (9)
    - mountPath: <directory_name>
      name: <volume_name>
    tolerations: (10)
```

```
 - operator: "Exists"
 resources: 11
   requests:
     memory: "64Mi"
     cpu: "250m"
   limits:
     memory: "128Mi"
     cpu: "500m"
 nodeSelector: 12
   foo: bar
```

1. Pour l'utilisation du mode d'installation **AllNamespaces**, indiquez l'espace de noms **openshift-operators**. Sinon, indiquez l'espace de noms unique correspondant à l'utilisation du mode d'installation **SingleNamespace**.
2. Nom du canal auquel s'abonner.
3. Nom de l'opérateur auquel s'abonner.
4. Nom de la source du catalogue qui fournit l'opérateur.
5. Espace de noms de la source de catalogue. Utilisez **openshift-marketplace** pour les sources de catalogue par défaut d'OperatorHub.
6. Le paramètre **env** définit une liste de variables d'environnement qui doivent exister dans tous les conteneurs du module créé par OLM.
7. Le paramètre **envFrom** définit une liste de sources pour alimenter les variables d'environnement dans le conteneur.
8. Le paramètre **volumes** définit une liste de volumes qui doivent exister sur le pod créé par OLM.
9. Le paramètre **volumeMounts** définit une liste de VolumeMounts qui doivent exister dans tous les conteneurs du pod créé par OLM. Si un **volumeMount** fait référence à un **volume** qui n'existe pas, OLM ne parvient pas à déployer l'opérateur.
10. Le paramètre **tolerations** définit une liste de tolérances pour le module créé par OLM.
11. Le paramètre **resources** définit les contraintes de ressources pour tous les conteneurs du module créé par OLM.
12. Le paramètre **nodeSelector** définit un **NodeSelector** pour le module créé par OLM.

5. Créer l'objet **Subscription**:

```
$ oc apply -f sub.yaml
```

A ce stade, OLM connaît l'opérateur sélectionné. Une version de service de cluster (CSV) pour l'opérateur devrait apparaître dans l'espace de noms cible, et les API fournies par l'opérateur devraient être disponibles pour la création.

2.4.4. Suppression d'opérateurs d'une grappe à l'aide de la console web

Les administrateurs de cluster peuvent supprimer les opérateurs installés dans un espace de noms sélectionné à l'aide de la console web.

Conditions préalables

- Vous avez accès à la console web d'un cluster OpenShift Container Platform en utilisant un compte avec les permissions **cluster-admin**.

Procédure

1. Naviguez jusqu'à la page **Operators → Installed Operators**.
2. Faites défiler ou saisissez un mot-clé dans le champ **Filter by name** pour trouver l'opérateur que vous souhaitez supprimer. Cliquez ensuite dessus.
3. Sur le côté droit de la page **Operator Details**, sélectionnez **Uninstall Operator** dans la liste **Actions**.
   Une boîte de dialogue **Uninstall Operator?** s'affiche.
4. Sélectionnez **Uninstall** pour supprimer l'opérateur, les déploiements de l'opérateur et les pods. Suite à cette action, l'opérateur cesse de fonctionner et ne reçoit plus de mises à jour.

NOTE

Cette action ne supprime pas les ressources gérées par l'opérateur, y compris les définitions de ressources personnalisées (CRD) et les ressources personnalisées (CR). Les tableaux de bord et les éléments de navigation activés par la console Web et les ressources hors cluster qui continuent de fonctionner peuvent nécessiter un nettoyage manuel. Pour les supprimer après la désinstallation de l'opérateur, vous devrez peut-être supprimer manuellement les CRD de l'opérateur.

2.4.5. Suppression d'opérateurs d'une grappe à l'aide de la CLI

Les administrateurs de clusters peuvent supprimer les opérateurs installés dans un espace de noms sélectionné à l'aide de l'interface de ligne de commande.

Conditions préalables

- Accès à un cluster OpenShift Container Platform à l'aide d'un compte disposant des autorisations **cluster-admin**.
- **oc** installée sur le poste de travail.

Procédure

1. Vérifiez la version actuelle de l'opérateur souscrit (par exemple, **jaeger**) dans le champ **currentCSV**:

```
$ oc get subscription jaeger -n openshift-operators -o yaml | grep currentCSV
```

Exemple de sortie

```
currentCSV: jaeger-operator.v1.8.2
```

2. Supprimer l'abonnement (par exemple, **jaeger**) :

```
$ oc delete subscription jaeger -n openshift-operators
```

Exemple de sortie

```
subscription.operators.coreos.com "jaeger" deleted
```

3. Supprimez le CSV de l'opérateur dans l'espace de noms cible en utilisant la valeur **currentCSV** de l'étape précédente :

```
$ oc delete clusterserviceversion jaeger-operator.v1.8.2 -n openshift-operators
```

Exemple de sortie

```
clusterserviceversion.operators.coreos.com "jaeger-operator.v1.8.2" deleted
```

2.5. RÉFÉRENCES EN MATIÈRE D'ENREGISTREMENT

2.5.1. Caractéristiques du collectionneur

Sortie	Protocol	Testé avec	Fluentd	Vecteur
Cloudwatch	REST sur HTTP(S)		✓	✓
Elasticsearch v6		v6.8.1	✓	✓
Elasticsearch v7		v7.12.2, 7.17.7	✓	✓
Elasticsearch v8		v8.4.3		✓
Fluent Forward (en français dans le texte)	Fluentd forward v1	Fluentd 1.14.6, Logstash 7.10.1	✓	
Journalisation de Google Cloud				✓
HTTP	HTTP 1.1	Fluentd 1.14.6, Vector 0.21		
Kafka	Kafka 0.11	Kafka 2.4.1, 2.7.0, 3.3.1	✓	✓
Loki	REST sur HTTP(S)	Loki 2.3.0, 2.7	✓	✓
Splunk	HEC	v8.2.9, 9.0.0		✓

Sortie	Protocol	Testé avec	Fluentd	Vecteur
Syslog	RFC3164, RFC5424	Rsyslog 8.37.0-9.el7	✓	

Tableau 2.1. Sources des journaux

Fonctionnalité	Fluentd	Vecteur
Journaux des conteneurs d'applications	✓	✓
Routage spécifique à l'application	✓	✓
Routage spécifique à l'application par espace de noms	✓	✓
Registres des conteneurs Infra	✓	✓
Journal de bord de l'infra	✓	✓
Journaux d'audit de l'API Kube	✓	✓
Journaux d'audit de l'API OpenShift	✓	✓
Journaux d'audit de l'Open Virtual Network (OVN)	✓	✓

Tableau 2.2. Autorisation et authentification

Fonctionnalité	Fluentd	Vecteur
Certificats Elasticsearch	✓	✓
Nom d'utilisateur / mot de passe Elasticsearch	✓	✓
Clés Cloudwatch	✓	✓
Cloudwatch STS	✓	✓
Certificats Kafka	✓	✓

Fonctionnalité	Fluentd	Vecteur
Nom d'utilisateur / mot de passe Kafka	✓	✓
Kafka SASL	✓	✓
Jeton du porteur de Loki	✓	✓

Tableau 2.3. Normalisations et transformations

Fonctionnalité	Fluentd	Vecteur
Modèle de données Viaq - app	✓	✓
Modèle de données Viaq - infra	✓	✓
Modèle de données Viaq - infra(journal)	✓	✓
Modèle de données Viaq - Audit Linux	✓	✓
Modèle de données Viaq - audit kube-apiserver	✓	✓
Modèle de données Viaq - Audit API OpenShift	✓	✓
Modèle de données Viaq - OVN	✓	✓
Normalisation des niveaux de journalisation	✓	✓
Analyse JSON	✓	✓
Indice structuré	✓	✓
Détection des erreurs multilignes	✓	
Indices multiconteneurs / fractionnés	✓	✓
Aplatir les étiquettes	✓	✓
Étiquettes statiques de la NSI	✓	✓

Tableau 2.4. Accorder

Fonctionnalité	Fluentd	Vecteur
Limite de lecture de Fluentd	✓	
Tampon Fluentd	✓	
- taille limite du chunk	✓	
- taille totale	✓	
- débordementaction	✓	
- flushthreadcount	✓	
- mode flush	✓	
- intervalle de rinçage	✓	
- retrywait	✓	
- type de tentative	✓	
- retrymaxinterval	✓	
- délai de réessai	✓	

Tableau 2.5. Visibilité

Fonctionnalité	Fluentd	Vecteur
Metrics	✓	✓
Tableau de bord	✓	✓
Alertes	✓	

Tableau 2.6. Divers

Fonctionnalité	Fluentd	Vecteur
Prise en charge globale du proxy	✓	✓
support x86	✓	✓
Support ARM	✓	✓

Fonctionnalité	Fluentd	Vecteur
Support IBM Power	✓	✓
Support IBM zSystems	✓	✓
Prise en charge de l'IPv6	✓	✓
Mise en mémoire tampon des événements du journal	✓	
Groupe déconnecté	✓	✓

Ressources complémentaires

- Documentation sur les vecteurs

2.5.2. Journalisation 5.6 Référence API

2.5.2.1. ClusterLogForwarder

ClusterLogForwarder est une API permettant de configurer le transfert des journaux.

Vous configurez le transfert en spécifiant une liste de **pipelines**, qui transfèrent un ensemble d'entrées nommées vers un ensemble de sorties nommées.

Il existe des noms d'entrée intégrés pour les catégories de journaux les plus courantes, et vous pouvez définir des entrées personnalisées pour effectuer des filtrages supplémentaires.

Il existe un nom de sortie intégré pour le magasin de logs openshift par défaut, mais vous pouvez définir vos propres sorties avec une URL et d'autres informations de connexion pour transmettre les logs à d'autres magasins ou processeurs, à l'intérieur ou à l'extérieur du cluster.

Pour plus de détails, voir la documentation sur les champs de l'API.

Propriété	Type	Description
spécimen	objet	Spécification du comportement souhaité du ClusterLogForwarder
statut	objet	État du ClusterLogForwarder

2.5.2.1.1. .spec

2.5.2.1.1.1. Description

ClusterLogForwarderSpec définit la manière dont les journaux doivent être transmis aux cibles distantes.

2.5.2.1.1.1.1. Type

- objet

Propriété	Type	Description
entrées	réseau	**(optional)** Les entrées sont des filtres nommés pour les messages de journalisation à transmettre.
outputDefaults	objet	**(optional)** DEPRECATED OutputDefaults spécifie explicitement la configuration du transitaire pour le magasin par défaut.
sorties	réseau	**(optional)** Les sorties sont des destinations nommées pour les messages de journalisation.
pipelines	réseau	Les pipelines transmettent les messages sélectionnés par un ensemble d'entrées à un ensemble de sorties.

2.5.2.1.2. .spec.inputs[]

2.5.2.1.2.1. Description

InputSpec définit un sélecteur de messages de journalisation.

2.5.2.1.2.1.1. Type

- réseau

Propriété	Type	Description
application	objet	**(optional)** L'application, si elle est présente, active un ensemble nommé de journaux **application** qui
nom	chaîne de caractères	Nom utilisé pour désigner l'entrée d'un site **pipeline**.

2.5.2.1.3. .spec.inputs[].application

2.5.2.1.3.1. Description

Sélecteur de journaux d'application. Toutes les conditions du sélecteur doivent être remplies (ET logique) pour sélectionner les journaux.

2.5.2.1.3.1.1. Type

- objet

Propriété	Type	Description
espaces nominatifs	réseau	**(optional)** Espaces de noms à partir desquels collecter les journaux d'application.
sélecteur	objet	**(optional)** Sélecteur de billes de bois provenant d'une cosse dont l'étiquette correspond à celle de la cosse.

2.5.2.1.4. .spec.inputs[].application.namespaces[]

2.5.2.1.4.1. Description

2.5.2.1.4.1.1. Type

- réseau

2.5.2.1.5. .spec.inputs[].application.selector

2.5.2.1.5.1. Description

Un sélecteur d'étiquettes est une requête d'étiquettes sur un ensemble de ressources.

2.5.2.1.5.1.1. Type

- objet

Propriété	Type	Description
matchLabels	objet	**(optional)** matchLabels est une carte de paires {key,value}. Un seul {key,value} dans la carte matchLabels

2.5.2.1.6. .spec.inputs[].application.selector.matchLabels

2.5.2.1.6.1. Description

2.5.2.1.6.1.1. Type

- objet

2.5.2.1.7. .spec.outputDefaults

2.5.2.1.7.1. Description

2.5.2.1.7.1.1. Type

- objet

Propriété	Type	Description
elasticsearch	objet	**(optional)** Valeurs par défaut d'Elasticsearch OutputSpec

2.5.2.1.8. .spec.outputDefaults.elasticsearch

2.5.2.1.8.1. Description

ElasticsearchStructuredSpec est une spécification liée aux modifications du journal structuré pour déterminer l'index Elasticsearch

2.5.2.1.8.1.1. Type

- objet

Propriété	Type	Description
enableStructuredContainerLogs (activer les journaux structurés des conteneurs)	bool	**(optional)** EnableStructuredContainerLogs permet d'activer les journaux structurés multi-conteneurs afin de permettre
structuredTypeKey	chaîne de caractères	**(optional)** StructuredTypeKey spécifie la clé de métadonnées à utiliser comme nom de l'index elasticsearch
structuredTypeName	chaîne de caractères	**(optional)** StructuredTypeName spécifie le nom du schéma elasticsearch

2.5.2.1.9. .spec.outputs[]

2.5.2.1.9.1. Description

La sortie définit une destination pour les messages du journal.

2.5.2.1.9.1.1. Type

- réseau

Propriété	Type	Description
syslog	objet	**(optional)**
fluentdForward	objet	**(optional)**
elasticsearch	objet	**(optional)**
kafka	objet	**(optional)**
cloudwatch	objet	**(optional)**
loki	objet	**(optional)**
googleCloudLogging	objet	**(optional)**
splunk	objet	**(optional)**
nom	chaîne de caractères	Nom utilisé pour désigner la sortie d'un site **pipeline**.
secret	objet	**(optional)** Secret d'authentification.
tls	objet	TLS contient des paramètres permettant de contrôler les options des connexions client TLS.
type	chaîne de caractères	Type de plugin de sortie.
url	chaîne de caractères	**(optional)** URL à laquelle envoyer les enregistrements.

2.5.2.1.10. .spec.outputs[].secret

2.5.2.1.10.1. Description

OutputSecretSpec est une référence secrète contenant uniquement le nom, sans espace de noms.

2.5.2.1.10.1.1. Type

- objet

Propriété	Type	Description
nom	chaîne de caractères	Nom d'un secret dans l'espace de noms configuré pour les secrets de transfert de journaux.

2.5.2.1.11. .spec.outputs[].tls

2.5.2.1.11.1. Description

OutputTLSSpec contient des options pour les connexions TLS qui ne dépendent pas du type de sortie.

2.5.2.1.11.1.1. Type

- objet

Propriété	Type	Description
insecureSkipVerify	bool	Si InsecureSkipVerify est vrai, le client TLS sera configuré pour ignorer les erreurs de certificats.

2.5.2.1.12. .spec.pipelines[]

2.5.2.1.12.1. Description

Les PipelinesSpec relient un ensemble d'entrées à un ensemble de sorties.

2.5.2.1.12.1.1. Type

- réseau

Propriété	Type	Description
détecter les erreurs multilignes	bool	**(optional)** DetectMultilineErrors active la détection des erreurs multilignes dans les journaux des conteneurs
inputRefs	réseau	InputRefs liste les noms (**input.name**) des entrées de ce pipeline.
étiquettes	objet	**(optional)** Étiquettes appliquées aux enregistrements qui passent par ce pipeline.

Propriété	Type	Description
nom	chaîne de caractères	**(optional)** Le nom est facultatif, mais il doit être unique dans la liste **pipelines** s'il est fourni.
outputRefs	réseau	OutputRefs liste les noms (**output.name**) des sorties de ce pipeline.
analyser	chaîne de caractères	**(optional)** Parse permet d'analyser les entrées du journal pour en faire des journaux structurés

2.5.2.1.13. .spec.pipelines[].inputRefs[]

2.5.2.1.13.1. Description

2.5.2.1.13.1.1. Type

- réseau

2.5.2.1.14. .spec.pipelines[].labels

2.5.2.1.14.1. Description

2.5.2.1.14.1.1. Type

- objet

2.5.2.1.15. .spec.pipelines[].outputRefs[]

2.5.2.1.15.1. Description

2.5.2.1.15.1.1. Type

- réseau

2.5.2.1.16. .statut

2.5.2.1.16.1. Description

ClusterLogForwarderStatus définit l'état observé du ClusterLogForwarder

2.5.2.1.16.1.1. Type

- objet

Propriété	Type	Description
conditions	objet	Conditions de l'expéditeur de journaux.
entrées	Conditions	Les entrées associent le nom de l'entrée à la condition de l'entrée.
sorties	Conditions	Les sorties associent le nom de la sortie à l'état de la sortie.
pipelines	Conditions	Pipelines associe le nom du pipeline à son état.

2.5.2.1.17. .status.conditions

2.5.2.1.17.1. Description

2.5.2.1.17.1.1. Type

- objet

2.5.2.1.18. .status.inputs

2.5.2.1.18.1. Description

2.5.2.1.18.1.1. Type

- Conditions

2.5.2.1.19. .status.outputs

2.5.2.1.19.1. Description

2.5.2.1.19.1.1. Type

- Conditions

2.5.2.1.20. .status.pipelines

2.5.2.1.20.1. Description

2.5.2.1.20.1.1. Type

- Conditions== ClusterLogging Une instance de journalisation Red Hat OpenShift. ClusterLogging est le schéma de l'API clusterloggings

Propriété	Type	Description
spécimen	objet	Spécification du comportement souhaité de ClusterLogging
statut	objet	Le statut définit l'état observé de ClusterLogging

2.5.2.1.21. .spec

2.5.2.1.21.1. Description

ClusterLoggingSpec définit l'état souhaité de ClusterLogging

2.5.2.1.21.1.1. Type

- objet

Propriété	Type	Description
collection	objet	Spécification du composant de collecte pour le cluster
curation	objet	**(DEPRECATED) (optional)** Obsolète. Spécification du composant de curation pour le cluster
transitaire	objet	**(DEPRECATED) (optional)** Déclassé. Spécification du composant Forwarder pour le cluster
logStore	objet	**(optional)** Spécification du composant Log Storage pour le cluster
état de la gestion	chaîne de caractères	**(optional)** Indicateur indiquant si la ressource est "gérée" ou "non gérée" par l'opérateur
visualisation	objet	**(optional)** Spécification du composant de visualisation pour le cluster

2.5.2.1.22. .spec.collection

2.5.2.1.22.1. Description

Il s'agit de la structure qui contiendra les informations relatives à la collecte des journaux et des événements

2.5.2.1.22.1.1. Type

- objet

Propriété	Type	Description
ressources	objet	**(optional)** Les ressources nécessaires pour le collecteur
nodeSelector	objet	**(optional)** Définir les nœuds sur lesquels les pods sont planifiés.
tolérances	réseau	**(optional)** Définir les tolérances acceptées par les pods
fluentd	objet	**(optional)** Fluentd représente la configuration des transitaires de type fluentd.
bûches	objet	**(DEPRECATED) (optional)** Déclassé. Spécification de la collecte de logs pour le cluster
type	chaîne de caractères	**(optional)** Le type de collecte de journaux à configurer

2.5.2.1.23. .spec.collection.fluentd

2.5.2.1.23.1. Description

FluentdForwarderSpec représente la configuration des transitaires de type fluentd.

2.5.2.1.23.1.1. Type

- objet

Propriété	Type	Description
tampon	objet	
inFile	objet	

2.5.2.1.24. .spec.collection.fluentd.buffer

2.5.2.1.24.1. Description

FluentdBufferSpec représente un sous-ensemble de paramètres de tampon fluentd permettant d'ajuster la configuration du tampon pour toutes les sorties fluentd. Il prend en charge un sous-ensemble de paramètres pour configurer la taille des tampons et des files d'attente, les opérations de vidage et les tentatives de vidage.

Pour les paramètres généraux, voir : https://docs.fluentd.org/configuration/buffer-section#buffering-parameters

Pour les paramètres de rinçage, voir : https://docs.fluentd.org/configuration/buffer-section#flushing-parameters

Pour les paramètres de réessai, voir : https://docs.fluentd.org/configuration/buffer-section#retries-parameters

2.5.2.1.24.1.1. Type

- objet

Propriété	Type	Description
chunkLimitSize	chaîne de caractères	**(optional)** ChunkLimitSize représente la taille maximale de chaque bloc. Les événements seront
flushInterval	chaîne de caractères	**(optional)** FlushInterval représente le temps d'attente entre deux vidanges consécutives
flushMode	chaîne de caractères	**(optional)** FlushMode représente le mode d'écriture des blocs par le thread de vidange. Le mode
flushThreadCount	int	**(optional)** FlushThreadCount représente le nombre de threads utilisés par le tampon fluentd
action de débordement	chaîne de caractères	**(optional)** OverflowAction représente l'action du plugin fluentd buffer pour
retryMaxInterval	chaîne de caractères	**(optional)** RetryMaxInterval représente l'intervalle de temps maximum pour le backoff exponentiel
délai de réessai	chaîne de caractères	**(optional)** RetryTimeout représente l'intervalle de temps maximum pour effectuer des tentatives avant d'abandonner

Propriété	Type	Description
retryType	chaîne de caractères	**(optional)** RetryType représente le type de répétition des opérations de purge. Les opérations de purge peuvent
retryWait	chaîne de caractères	**(optional)** RetryWait représente la durée entre deux tentatives consécutives de rinçage
totalLimitSize	chaîne de caractères	**(optional)** TotalLimitSize représente le seuil d'espace de nœud autorisé par fluentd

2.5.2.1.25. .spec.collection.fluentd.inFile

2.5.2.1.25.1. Description

FluentdInFileSpec représente un sous-ensemble de paramètres du plugin fluentd in-tail permettant d'ajuster la configuration pour toutes les entrées fluentd in-tail.

Pour les paramètres généraux, voir : https://docs.fluentd.org/input/tail#parameters

2.5.2.1.25.1.1. Type

- objet

Propriété	Type	Description
readLinesLimit	int	**(optional)** ReadLinesLimit représente le nombre de lignes à lire à chaque opération d'E/S

2.5.2.1.26. .spec.collection.logs

2.5.2.1.26.1. Description

2.5.2.1.26.1.1. Type

- objet

Propriété	Type	Description
fluentd	objet	Spécification du composant Fluentd Log Collection

Propriété	Type	Description
type	chaîne de caractères	Le type de collecte de journaux à configurer

2.5.2.1.27. .spec.collection.logs.fluentd

2.5.2.1.27.1. Description

CollectorSpec est une spécification permettant de définir l'ordonnancement et les ressources d'un collecteur

2.5.2.1.27.1.1. Type

- objet

Propriété	Type	Description
nodeSelector	objet	**(optional)** Définir les nœuds sur lesquels les pods sont planifiés.
ressources	objet	**(optional)** Les ressources nécessaires pour le collecteur
tolérances	réseau	**(optional)** Définir les tolérances acceptées par les pods

2.5.2.1.28. .spec.collection.logs.fluentd.nodeSelector

2.5.2.1.28.1. Description

2.5.2.1.28.1.1. Type

- objet

2.5.2.1.29. .spec.collection.logs.fluentd.resources

2.5.2.1.29.1. Description

2.5.2.1.29.1.1. Type

- objet

Propriété	Type	Description

Propriété	Type	Description
limites	objet	**(optional)** Limites décrit la quantité maximale de ressources de calcul autorisée.
demandes	objet	**(optional)** Les demandes décrivent la quantité minimale de ressources informatiques requises.

2.5.2.1.30. .spec.collection.logs.fluentd.resources.limits

2.5.2.1.30.1. Description

2.5.2.1.30.1.1. Type

- objet

2.5.2.1.31. .spec.collection.logs.fluentd.resources.requests

2.5.2.1.31.1. Description

2.5.2.1.31.1.1. Type

- objet

2.5.2.1.32. .spec.collection.logs.fluentd.tolerations[]

2.5.2.1.32.1. Description

2.5.2.1.32.1.1. Type

- réseau

Propriété	Type	Description
effet	chaîne de caractères	**(optional)** Effect indique l'effet d'altération à prendre en compte. Vide signifie que tous les effets d'altération doivent être pris en compte.
clé	chaîne de caractères	**(optional)** Key est la clé d'altération à laquelle s'applique la tolérance. Vide signifie que la tolérance s'applique à toutes les clés d'altération.

Propriété	Type	Description
opérateur	chaîne de caractères	**(optional)** L'opérateur représente la relation entre la clé et la valeur.
secondes de tolérance	int	**(optional)** TolerationSeconds représente la période de temps pendant laquelle la tolérance (qui doit être
valeur	chaîne de caractères	**(optional)** La valeur est la valeur d'altération à laquelle correspond la tolérance.

2.5.2.1.33. .spec.collection.logs.fluentd.tolerations[].tolerationSeconds

2.5.2.1.33.1. Description

2.5.2.1.33.1.1. Type

- int

2.5.2.1.34. .spec.curation

2.5.2.1.34.1. Description

Il s'agit de la structure qui contiendra les informations relatives à la curation du journal (Curator)

2.5.2.1.34.1.1. Type

- objet

Propriété	Type	Description
conservateur	objet	La spécification de la curation à configurer
type	chaîne de caractères	Le type de curation à configurer

2.5.2.1.35. .spec.curation.curator

2.5.2.1.35.1. Description

2.5.2.1.35.1.1. Type

- objet

Propriété	Type	Description
nodeSelector	objet	Définir les nœuds sur lesquels les pods sont planifiés.
ressources	objet	**(optional)** Les ressources nécessaires pour le conservateur
calendrier	chaîne de caractères	The cron schedule that the Curator job is run. Defaults to "30 3 * * *"
tolérances	réseau	

2.5.2.1.36. .spec.curation.curator.nodeSelector

2.5.2.1.36.1. Description

2.5.2.1.36.1.1. Type

- objet

2.5.2.1.37. .spec.curation.curator.resources

2.5.2.1.37.1. Description

2.5.2.1.37.1.1. Type

- objet

Propriété	Type	Description
limites	objet	**(optional)** Limites décrit la quantité maximale de ressources de calcul autorisée.
demandes	objet	**(optional)** Les demandes décrivent la quantité minimale de ressources informatiques requises.

2.5.2.1.38. .spec.curation.curator.resources.limits

2.5.2.1.38.1. Description

2.5.2.1.38.1.1. Type

- objet

2.5.2.1.39. .spec.curation.curator.resources.requests

2.5.2.1.39.1. Description

2.5.2.1.39.1.1. Type

- objet

2.5.2.1.40. .spec.curation.curator.tolerations[]

2.5.2.1.40.1. Description

2.5.2.1.40.1.1. Type

- réseau

Propriété	Type	Description
effet	chaîne de caractères	**(optional)** Effect indique l'effet d'altération à prendre en compte. Vide signifie que tous les effets d'altération doivent être pris en compte.
clé	chaîne de caractères	**(optional)** Key est la clé d'altération à laquelle s'applique la tolérance. Vide signifie que la tolérance s'applique à toutes les clés d'altération.
opérateur	chaîne de caractères	**(optional)** L'opérateur représente la relation entre la clé et la valeur.
secondes de tolérance	int	**(optional)** TolerationSeconds représente la période de temps pendant laquelle la tolérance (qui doit être
valeur	chaîne de caractères	**(optional)** La valeur est la valeur d'altération à laquelle correspond la tolérance.

2.5.2.1.41. .spec.curation.curator.tolerations[].tolerationSeconds

2.5.2.1.41.1. Description

2.5.2.1.41.1.1. Type

- int

2.5.2.1.42. .spec.forwarder

2.5.2.1.42.1. Description

ForwarderSpec contient des paramètres de réglage globaux pour des implémentations spécifiques de transitaires. Ce champ n'est pas nécessaire pour une utilisation générale, mais il permet aux utilisateurs connaissant la technologie sous-jacente du transitaire de régler les performances. Actuellement pris en charge : **fluentd**.

2.5.2.1.42.1.1. Type

- objet

Propriété	Type	Description
fluentd	objet	

2.5.2.1.43. .spec.forwarder.fluentd

2.5.2.1.43.1. Description

FluentdForwarderSpec représente la configuration des transitaires de type fluentd.

2.5.2.1.43.1.1. Type

- objet

Propriété	Type	Description
tampon	objet	
inFile	objet	

2.5.2.1.44. .spec.forwarder.fluentd.buffer

2.5.2.1.44.1. Description

FluentdBufferSpec représente un sous-ensemble de paramètres de tampon fluentd permettant d'ajuster la configuration du tampon pour toutes les sorties fluentd. Il prend en charge un sous-ensemble de paramètres pour configurer la taille des tampons et des files d'attente, les opérations de vidage et les tentatives de vidage.

Pour les paramètres généraux, voir : https://docs.fluentd.org/configuration/buffer-section#buffering-parameters

Pour les paramètres de rinçage, voir : https://docs.fluentd.org/configuration/buffer-section#flushing-parameters

Pour les paramètres de réessai, voir : https://docs.fluentd.org/configuration/buffer-section#retries-parameters

2.5.2.1.44.1.1. Type

- objet

Propriété	Type	Description
chunkLimitSize	chaîne de caractères	**(optional)** ChunkLimitSize représente la taille maximale de chaque bloc. Les événements seront
flushInterval	chaîne de caractères	**(optional)** FlushInterval représente le temps d'attente entre deux vidanges consécutives
flushMode	chaîne de caractères	**(optional)** FlushMode représente le mode d'écriture des blocs par le thread de vidange. Le mode
flushThreadCount	int	**(optional)** FlushThreadCount représente le nombre de threads utilisés par le tampon fluentd
action de débordement	chaîne de caractères	**(optional)** OverflowAction représente l'action du plugin fluentd buffer pour
retryMaxInterval	chaîne de caractères	**(optional)** RetryMaxInterval représente l'intervalle de temps maximum pour le backoff exponentiel
délai de réessai	chaîne de caractères	**(optional)** RetryTimeout représente l'intervalle de temps maximum pour effectuer des tentatives avant d'abandonner
retryType	chaîne de caractères	**(optional)** RetryType représente le type de répétition des opérations de purge. Les opérations de purge peuvent
retryWait	chaîne de caractères	**(optional)** RetryWait représente la durée entre deux tentatives consécutives de rinçage

Propriété	Type	Description
totalLimitSize	chaîne de caractères	**(optional)** TotalLimitSize représente le seuil d'espace de nœud autorisé par fluentd

2.5.2.1.45. .spec.forwarder.fluentd.inFile

2.5.2.1.45.1. Description

FluentdInFileSpec représente un sous-ensemble de paramètres du plugin fluentd in-tail permettant d'ajuster la configuration pour toutes les entrées fluentd in-tail.

Pour les paramètres généraux, voir : https://docs.fluentd.org/input/tail#parameters

2.5.2.1.45.1.1. Type

- objet

Propriété	Type	Description
readLinesLimit	int	**(optional)** ReadLinesLimit représente le nombre de lignes à lire à chaque opération d'E/S

2.5.2.1.46. .spec.logStore

2.5.2.1.46.1. Description

La spécification LogStoreSpec contient des informations sur la manière dont les journaux sont stockés.

2.5.2.1.46.1.1. Type

- objet

Propriété	Type	Description
elasticsearch	objet	Spécification du composant Elasticsearch Log Store
lokistack	objet	LokiStack contient des informations sur la LokiStack à utiliser pour le stockage des journaux si Type est défini sur LogStoreTypeLokiStack.

Propriété	Type	Description
politique de rétention	objet	**(optional)** La politique de conservation définit l'âge maximum d'un index après lequel il doit être supprimé
type	chaîne de caractères	Le type de stockage de logs à configurer. L'opérateur prend actuellement en charge l'utilisation d'ElasticSearch

2.5.2.1.47. .spec.logStore.elasticsearch

2.5.2.1.47.1. Description

2.5.2.1.47.1.1. Type

- objet

Propriété	Type	Description
nodeCount	int	Nombre de nœuds à déployer pour Elasticsearch
nodeSelector	objet	Définir les nœuds sur lesquels les pods sont planifiés.
mandataire	objet	Spécification du composant Elasticsearch Proxy
politique de redondance	chaîne de caractères	**(optional)**
ressources	objet	**(optional)** Les ressources nécessaires pour Elasticsearch
stockage	objet	**(optional)** Spécification de stockage pour les nœuds de données Elasticsearch
tolérances	réseau	

2.5.2.1.48. .spec.logStore.elasticsearch.nodeSelector

2.5.2.1.48.1. Description

2.5.2.1.48.1.1. Type

- objet

2.5.2.1.49. .spec.logStore.elasticsearch.proxy

2.5.2.1.49.1. Description

2.5.2.1.49.1.1. Type

- objet

Propriété	Type	Description
ressources	objet	

2.5.2.1.50. .spec.logStore.elasticsearch.proxy.resources

2.5.2.1.50.1. Description

2.5.2.1.50.1.1. Type

- objet

Propriété	Type	Description
limites	objet	**(optional)** Limites décrit la quantité maximale de ressources de calcul autorisée.
demandes	objet	**(optional)** Les demandes décrivent la quantité minimale de ressources informatiques requises.

2.5.2.1.51. .spec.logStore.elasticsearch.proxy.resources.limits

2.5.2.1.51.1. Description

2.5.2.1.51.1.1. Type

- objet

2.5.2.1.52. .spec.logStore.elasticsearch.proxy.resources.requests

2.5.2.1.52.1. Description

2.5.2.1.52.1.1. Type

- objet

2.5.2.1.53. .spec.logStore.elasticsearch.resources

2.5.2.1.53.1. Description

2.5.2.1.53.1.1. Type

- objet

Propriété	Type	Description
limites	objet	**(optional)** Limites décrit la quantité maximale de ressources de calcul autorisée.
demandes	objet	**(optional)** Les demandes décrivent la quantité minimale de ressources informatiques requises.

2.5.2.1.54. .spec.logStore.elasticsearch.resources.limits

2.5.2.1.54.1. Description

2.5.2.1.54.1.1. Type

- objet

2.5.2.1.55. .spec.logStore.elasticsearch.resources.requests

2.5.2.1.55.1. Description

2.5.2.1.55.1.1. Type

- objet

2.5.2.1.56. .spec.logStore.elasticsearch.storage

2.5.2.1.56.1. Description

2.5.2.1.56.1.1. Type

- objet

Propriété	Type	Description
taille	objet	Capacité de stockage maximale pour le nœud à provisionner.

Propriété	Type	Description
nom de la classe de stockage	chaîne de caractères	**(optional)** Le nom de la classe de stockage à utiliser pour créer le PVC du nœud.

2.5.2.1.57. .spec.logStore.elasticsearch.storage.size

2.5.2.1.57.1. Description

2.5.2.1.57.1.1. Type

- objet

Propriété	Type	Description
Format	chaîne de caractères	Changer de format à volonté. Voir le commentaire pour Canonicalize pour
d	objet	d est la quantité sous forme inf.Dec si d.Dec != nil
i	int	i est la quantité sous forme d'échelle int64, si d.Dec == nil
s	chaîne de caractères	s est la valeur générée de cette quantité pour éviter un nouveau calcul

2.5.2.1.58. .spec.logStore.elasticsearch.storage.size.d

2.5.2.1.58.1. Description

2.5.2.1.58.1.1. Type

- objet

Propriété	Type	Description
Déc	objet	

2.5.2.1.59. .spec.logStore.elasticsearch.storage.size.d.Dec

2.5.2.1.59.1. Description

2.5.2.1.59.1.1. Type

- objet

Propriété	Type	Description
échelle	int	
sans échelle	objet	

2.5.2.1.60. .spec.logStore.elasticsearch.storage.size.d.Dec.unscaled

2.5.2.1.60.1. Description

2.5.2.1.60.1.1. Type

- objet

Propriété	Type	Description
abs	Mot	signe
négliger	bool	

2.5.2.1.61. .spec.logStore.elasticsearch.storage.size.d.Dec.unscaled.abs

2.5.2.1.61.1. Description

2.5.2.1.61.1.1. Type

- Mot

2.5.2.1.62. .spec.logStore.elasticsearch.storage.size.i

2.5.2.1.62.1. Description

2.5.2.1.62.1.1. Type

- int

Propriété	Type	Description
échelle	int	
valeur	int	

2.5.2.1.63. .spec.logStore.elasticsearch.tolerations[]

2.5.2.1.63.1. Description

2.5.2.1.63.1.1. Type

- réseau

Propriété	Type	Description
effet	chaîne de caractères	**(optional)** Effect indique l'effet d'altération à prendre en compte. Vide signifie que tous les effets d'altération doivent être pris en compte.
clé	chaîne de caractères	**(optional)** Key est la clé d'altération à laquelle s'applique la tolérance. Vide signifie que la tolérance s'applique à toutes les clés d'altération.
opérateur	chaîne de caractères	**(optional)** L'opérateur représente la relation entre la clé et la valeur.
secondes de tolérance	int	**(optional)** TolerationSeconds représente la période de temps pendant laquelle la tolérance (qui doit être
valeur	chaîne de caractères	**(optional)** La valeur est la valeur d'altération à laquelle correspond la tolérance.

2.5.2.1.64. .spec.logStore.elasticsearch.tolerations[].tolerationSeconds

2.5.2.1.64.1. Description

2.5.2.1.64.1.1. Type

- int

2.5.2.1.65. .spec.logStore.lokistack

2.5.2.1.65.1. Description

LokiStackStoreSpec est utilisé pour configurer le cluster-logging afin d'utiliser une LokiStack comme stockage de logs. Il pointe vers une LokiStack existante dans le même espace de noms.

2.5.2.1.65.1.1. Type

- objet

Propriété	Type	Description
nom	chaîne de caractères	Nom de la ressource LokiStack.

2.5.2.1.66. .spec.logStore.retentionPolicy

2.5.2.1.66.1. Description

2.5.2.1.66.1.1. Type

- objet

Propriété	Type	Description
application	objet	
audit	objet	
infra	objet	

2.5.2.1.67. .spec.logStore.retentionPolicy.application

2.5.2.1.67.1. Description

2.5.2.1.67.1.1. Type

- objet

Propriété	Type	Description
seuil de disquePourcentage	int	**(optional)** Le pourcentage seuil de l'utilisation du disque ES qui, lorsqu'il est atteint, entraîne la suppression des anciens index (par exemple 75)
maxAge	chaîne de caractères	**(optional)**
namespaceSpec	réseau	**(optional)** La spécification par espace de noms pour supprimer les documents plus anciens qu'un âge minimum donné

Propriété	Type	Description
pruneNamespacesIntervalle	chaîne de caractères	**(optional)** Quelle est la fréquence d'exécution d'une nouvelle tâche "prune-namespaces" (élaguer les espaces de noms) ?

2.5.2.1.68. .spec.logStore.retentionPolicy.application.namespaceSpec[]

2.5.2.1.68.1. Description

2.5.2.1.68.1.1. Type

- réseau

Propriété	Type	Description
âge min	chaîne de caractères	**(optional)** Supprimer les enregistrements correspondant aux espaces de noms qui sont plus anciens que cet âge minimum (par exemple 1d)
espace de noms	chaîne de caractères	Espace de noms cible pour supprimer les journaux plus anciens que MinAge (valeur par défaut : 7d)

2.5.2.1.69. .spec.logStore.retentionPolicy.audit

2.5.2.1.69.1. Description

2.5.2.1.69.1.1. Type

- objet

Propriété	Type	Description
seuil de disquePourcentage	int	**(optional)** Le pourcentage seuil de l'utilisation du disque ES qui, lorsqu'il est atteint, entraîne la suppression des anciens index (par exemple 75)
maxAge	chaîne de caractères	**(optional)**

Propriété	Type	Description
namespaceSpec	réseau	**(optional)** La spécification par espace de noms pour supprimer les documents plus anciens qu'un âge minimum donné
pruneNamespacesIntervalle	chaîne de caractères	**(optional)** Quelle est la fréquence d'exécution d'une nouvelle tâche "prune-namespaces" (élaguer les espaces de noms) ?

2.5.2.1.70. .spec.logStore.retentionPolicy.audit.namespaceSpec[]

2.5.2.1.70.1. Description

2.5.2.1.70.1.1. Type

- réseau

Propriété	Type	Description
âge min	chaîne de caractères	**(optional)** Supprimer les enregistrements correspondant aux espaces de noms qui sont plus anciens que cet âge minimum (par exemple 1d)
espace de noms	chaîne de caractères	Espace de noms cible pour supprimer les journaux plus anciens que MinAge (valeur par défaut : 7d)

2.5.2.1.71. .spec.logStore.retentionPolicy.infra

2.5.2.1.71.1. Description

2.5.2.1.71.1.1. Type

- objet

Propriété	Type	Description
seuil de disquePourcentage	int	**(optional)** Le pourcentage seuil de l'utilisation du disque ES qui, lorsqu'il est atteint, entraîne la suppression des anciens index (par exemple 75)

Propriété	Type	Description
maxAge	chaîne de caractères	**(optional)**
namespaceSpec	réseau	**(optional)** La spécification par espace de noms pour supprimer les documents plus anciens qu'un âge minimum donné
pruneNamespacesIntervalle	chaîne de caractères	**(optional)** Quelle est la fréquence d'exécution d'une nouvelle tâche "prune-namespaces" (élaguer les espaces de noms) ?

2.5.2.1.72. .spec.logStore.retentionPolicy.infra.namespaceSpec[]

2.5.2.1.72.1. Description

2.5.2.1.72.1.1. Type

- réseau

Propriété	Type	Description
âge min	chaîne de caractères	**(optional)** Supprimer les enregistrements correspondant aux espaces de noms qui sont plus anciens que cet âge minimum (par exemple 1d)
espace de noms	chaîne de caractères	Espace de noms cible pour supprimer les journaux plus anciens que MinAge (valeur par défaut : 7d)

2.5.2.1.73. .spec.visualisation

2.5.2.1.73.1. Description

Il s'agit de la structure qui contiendra les informations pertinentes pour la visualisation des logs (Kibana)

2.5.2.1.73.1.1. Type

- objet

Propriété	Type	Description

Propriété	Type	Description
kibana	objet	Spécification du composant de visualisation Kibana
type	chaîne de caractères	Le type de visualisation à configurer

2.5.2.1.74. .spec.visualization.kibana

2.5.2.1.74.1. Description

2.5.2.1.74.1.1. Type

- objet

Propriété	Type	Description
nodeSelector	objet	Définir les nœuds sur lesquels les pods sont planifiés.
mandataire	objet	Spécification du composant Kibana Proxy
répliques	int	Nombre d'instances à déployer pour un déploiement de Kibana
ressources	objet	**(optional)** Ressources nécessaires pour Kibana
tolérances	réseau	

2.5.2.1.75. .spec.visualization.kibana.nodeSelector

2.5.2.1.75.1. Description

2.5.2.1.75.1.1. Type

- objet

2.5.2.1.76. .spec.visualization.kibana.proxy

2.5.2.1.76.1. Description

2.5.2.1.76.1.1. Type

- objet

Propriété	Type	Description
ressources	objet	

2.5.2.1.77. .spec.visualization.kibana.proxy.resources

2.5.2.1.77.1. Description

2.5.2.1.77.1.1. Type

- objet

Propriété	Type	Description
limites	objet	**(optional)** Limites décrit la quantité maximale de ressources de calcul autorisée.
demandes	objet	**(optional)** Les demandes décrivent la quantité minimale de ressources informatiques requises.

2.5.2.1.78. .spec.visualization.kibana.proxy.resources.limits

2.5.2.1.78.1. Description

2.5.2.1.78.1.1. Type

- objet

2.5.2.1.79. .spec.visualization.kibana.proxy.resources.requests

2.5.2.1.79.1. Description

2.5.2.1.79.1.1. Type

- objet

2.5.2.1.80. .spec.visualization.kibana.replicas

2.5.2.1.80.1. Description

2.5.2.1.80.1.1. Type

- int

2.5.2.1.81. .spec.visualization.kibana.resources

2.5.2.1.81.1. Description

2.5.2.1.81.1.1. Type

- objet

Propriété	Type	Description
limites	objet	**(optional)** Limites décrit la quantité maximale de ressources de calcul autorisée.
demandes	objet	**(optional)** Les demandes décrivent la quantité minimale de ressources informatiques requises.

2.5.2.1.82. .spec.visualization.kibana.resources.limits

2.5.2.1.82.1. Description

2.5.2.1.82.1.1. Type

- objet

2.5.2.1.83. .spec.visualization.kibana.resources.requests

2.5.2.1.83.1. Description

2.5.2.1.83.1.1. Type

- objet

2.5.2.1.84. .spec.visualization.kibana.tolerations[]

2.5.2.1.84.1. Description

2.5.2.1.84.1.1. Type

- réseau

Propriété	Type	Description
effet	chaîne de caractères	**(optional)** Effect indique l'effet d'altération à prendre en compte. Vide signifie que tous les effets d'altération doivent être pris en compte.

Propriété	Type	Description
clé	chaîne de caractères	**(optional)** Key est la clé d'altération à laquelle s'applique la tolérance. Vide signifie que la tolérance s'applique à toutes les clés d'altération.
opérateur	chaîne de caractères	**(optional)** L'opérateur représente la relation entre la clé et la valeur.
secondes de tolérance	int	**(optional)** TolerationSeconds représente la période de temps pendant laquelle la tolérance (qui doit être
valeur	chaîne de caractères	**(optional)** La valeur est la valeur d'altération à laquelle correspond la tolérance.

2.5.2.1.85. .spec.visualization.kibana.tolerations[].tolerationSeconds

2.5.2.1.85.1. Description

2.5.2.1.85.1.1. Type

- int

2.5.2.1.86. .statut

2.5.2.1.86.1. Description

ClusterLoggingStatus définit l'état observé de ClusterLogging

2.5.2.1.86.1.1. Type

- objet

Propriété	Type	Description
collection	objet	**(optional)**
conditions	objet	**(optional)**
curation	objet	**(optional)**

Propriété	Type	Description
logStore	objet	**(optional)**
visualisation	objet	**(optional)**

2.5.2.1.87. .status.collection

2.5.2.1.87.1. Description

2.5.2.1.87.1.1. Type

- objet

Propriété	Type	Description
bûches	objet	**(optional)**

2.5.2.1.88. .status.collection.logs

2.5.2.1.88.1. Description

2.5.2.1.88.1.1. Type

- objet

Propriété	Type	Description
fluentdStatus	objet	**(optional)**

2.5.2.1.89. .status.collection.logs.fluentdStatus

2.5.2.1.89.1. Description

2.5.2.1.89.1.1. Type

- objet

Propriété	Type	Description
clusterCondition	objet	**(optional)**
daemonSet	chaîne de caractères	**(optional)**
nœuds	objet	**(optional)**

Propriété	Type	Description
gousses	chaîne de caractères	**(optional)**

2.5.2.1.90. .status.collection.logs.fluentdStatus.clusterCondition

2.5.2.1.90.1. Description

operator-sdk generate crds n'autorise pas le map-of-slice, il doit utiliser un type nommé.

2.5.2.1.90.1.1. Type

- objet

2.5.2.1.91. .status.collection.logs.fluentdStatus.nodes

2.5.2.1.91.1. Description

2.5.2.1.91.1.1. Type

- objet

2.5.2.1.92. .status.conditions

2.5.2.1.92.1. Description

2.5.2.1.92.1.1. Type

- objet

2.5.2.1.93. .status.curation

2.5.2.1.93.1. Description

2.5.2.1.93.1.1. Type

- objet

Propriété	Type	Description
statut du curateur	réseau	**(optional)**

2.5.2.1.94. .status.curation.curatorStatus[]

2.5.2.1.94.1. Description

2.5.2.1.94.1.1. Type

- réseau

Propriété	Type	Description
clusterCondition	objet	**(optional)**
cronJobs	chaîne de caractères	**(optional)**
horaires	chaîne de caractères	**(optional)**
suspendu	bool	**(optional)**

2.5.2.1.95. .status.curation.curatorStatus[].clusterCondition

2.5.2.1.95.1. Description

operator-sdk generate crds n'autorise pas le map-of-slice, il doit utiliser un type nommé.

2.5.2.1.95.1.1. Type

- objet

2.5.2.1.96. .status.logStore

2.5.2.1.96.1. Description

2.5.2.1.96.1.1. Type

- objet

Propriété	Type	Description
elasticsearchStatus	réseau	**(optional)**

2.5.2.1.97. .status.logStore.elasticsearchStatus[]

2.5.2.1.97.1. Description

2.5.2.1.97.1.1. Type

- réseau

Propriété	Type	Description
groupe	objet	**(optional)**
clusterConditions	objet	**(optional)**

Propriété	Type	Description
clusterHealth	chaîne de caractères	**(optional)**
nom du groupe	chaîne de caractères	**(optional)**
déploiements	réseau	**(optional)**
nodeConditions	objet	**(optional)**
nodeCount	int	**(optional)**
gousses	objet	**(optional)**
ensembles de répliques	réseau	**(optional)**
shardAllocationEnabled	chaîne de caractères	**(optional)**
statefulSets	réseau	**(optional)**

2.5.2.1.98. .status.logStore.elasticsearchStatus[].cluster

2.5.2.1.98.1. Description

2.5.2.1.98.1.1. Type

- objet

Propriété	Type	Description
tessons primaires actifs	int	Le nombre de Shards primaires actifs pour le cluster Elasticsearch
activeShards	int	Le nombre de Shards actifs pour le cluster Elasticsearch
initialisation des tessons	int	Le nombre de Shards d'initialisation pour le cluster Elasticsearch
numDataNodes	int	Nombre de nœuds de données pour le cluster Elasticsearch
numNodes	int	Le nombre de nœuds pour le cluster Elasticsearch

Propriété	Type	Description
tâches en cours	int	
relocalisationShards	int	Le nombre de Shards de relocalisation pour le cluster Elasticsearch
statut	chaîne de caractères	L'état actuel du cluster Elasticsearch
tessons non attribués	int	Le nombre de Shards non assignés pour le cluster Elasticsearch

2.5.2.1.99. .status.logStore.elasticsearchStatus[].clusterConditions

2.5.2.1.99.1. Description

2.5.2.1.99.1.1. Type

- objet

2.5.2.1.100. .status.logStore.elasticsearchStatus[].deployments[]

2.5.2.1.100.1. Description

2.5.2.1.100.1.1. Type

- réseau

2.5.2.1.101. .status.logStore.elasticsearchStatus[].nodeConditions

2.5.2.1.101.1. Description

2.5.2.1.101.1.1. Type

- objet

2.5.2.1.102. .status.logStore.elasticsearchStatus[].pods

2.5.2.1.102.1. Description

2.5.2.1.102.1.1. Type

- objet

2.5.2.1.103. .status.logStore.elasticsearchStatus[].replicaSets[]

2.5.2.1.103.1. Description

2.5.2.1.103.1.1. Type

- réseau

2.5.2.1.104. .status.logStore.elasticsearchStatus[].statefulSets[]

2.5.2.1.104.1. Description

2.5.2.1.104.1.1. Type

- réseau

2.5.2.1.105. .status.visualization

2.5.2.1.105.1. Description

2.5.2.1.105.1.1. Type

- objet

Propriété	Type	Description
kibanaStatus	réseau	**(optional)**

2.5.2.1.106. .status.visualization.kibanaStatus[]

2.5.2.1.106.1. Description

2.5.2.1.106.1.1. Type

- réseau

Propriété	Type	Description
clusterCondition	objet	**(optional)**
déploiement	chaîne de caractères	**(optional)**
gousses	chaîne de caractères	**(optional)** Le statut de chacun des pods Kibana pour le composant Visualisation
ensembles de répliques	réseau	**(optional)**
répliques	int	**(optional)**

2.5.2.1.107. .status.visualization.kibanaStatus[].clusterCondition

2.5.2.1.107.1. Description

2.5.2.1.107.1.1. Type

- objet

2.5.2.1.108. .status.visualization.kibanaStatus[].replicaSets[]

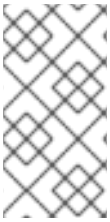
2.5.2.1.108.1. Description

2.5.2.1.108.1.1. Type

- réseau

CHAPITRE 3. ENREGISTREMENT 5.5

3.1. NOTES DE VERSION SUR LA JOURNALISATION 5.5



NOTE

Le sous-système de journalisation pour Red Hat OpenShift est fourni en tant que composant installable, avec un cycle de publication distinct de celui de la plateforme principale OpenShift Container Platform. La politique de cycle de vie de Red Hat OpenShift Container Platform décrit la compatibilité des versions.

3.1.1. Journalisation 5.5.10

Cette version inclut la version 5.5.10 de la correction des bugs de journalisation d'OpenShift .

3.1.1.1. Bug fixes

- Avant cette mise à jour, le plugin logging view de l'OpenShift Web Console n'affichait qu'un texte d'erreur lorsque la LokiStack n'était pas joignable. Après cette mise à jour, le plugin affiche un message d'erreur approprié avec des détails sur la façon de réparer la LokiStack inaccessible.(LOG-2874)

3.1.1.2. CVE

- CVE-2022-4304
- CVE-2022-4450
- CVE-2023-0215
- CVE-2023-0286
- CVE-2023-0361
- CVE-2023-23916

3.1.2. Journalisation 5.5.9

Cette version inclut la version 5.5.9 de la correction des bugs de journalisation d'OpenShift .

3.1.2.1. Bug fixes

- Avant cette mise à jour, un problème avec le collecteur Fluentd faisait qu'il ne capturait pas les événements de connexion OAuth stockés dans **/var/log/auth-server/audit.log**. Cela conduisait à une collecte incomplète des événements de connexion du service OAuth. Avec cette mise à jour, le collecteur Fluentd résout maintenant ce problème en capturant tous les événements de connexion du service OAuth, y compris ceux stockés dans **/var/log/auth-server/audit.log**, comme prévu.(LOG-3730)

- Avant cette mise à jour, lorsque l'analyse structurée était activée et que les messages étaient transmis à plusieurs destinations, ils n'étaient pas copiés en profondeur. Par conséquent, certains des journaux reçus incluaient le message structuré, tandis que d'autres ne le faisaient pas. Avec cette mise à jour, la génération de configuration a été modifiée pour copier en

profondeur les messages avant l'analyse JSON. Par conséquent, tous les journaux reçus contiennent désormais des messages structurés, même lorsqu'ils sont transmis à plusieurs destinations.(LOG-3767)

3.1.2.2. CVE

- CVE-2022-4304
- CVE-2022-4450
- CVE-2022-41717
- CVE-2023-0215
- CVE-2023-0286
- CVE-2023-0767
- CVE-2023-23916

3.1.3. Journalisation 5.5.8

Cette version inclut la version 5.5.8 de la correction des bugs de journalisation d'OpenShift .

3.1.3.1. Bug fixes

- Avant cette mise à jour, le champ **priority** était absent des journaux **systemd** en raison d'une erreur dans la manière dont le collecteur définissait les champs **level**. Avec cette mise à jour, ces champs sont définis correctement, ce qui résout le problème.(LOG-3630)

3.1.3.2. CVE

- CVE-2020-10735
- CVE-2021-28861
- CVE-2022-2873
- CVE-2022-4415
- CVE-2022-24999
- CVE-2022-40897
- CVE-2022-41222
- CVE-2022-41717
- CVE-2022-43945
- CVE-2022-45061
- CVE-2022-48303

3.1.4. Journalisation 5.5.7

Cette version inclut la version 5.5.7 de la correction des bugs de journalisation d'OpenShift .

3.1.4.1. Bug fixes

- Avant cette mise à jour, le LokiStack Gateway Labels Enforcer générait des erreurs d'analyse pour les requêtes LogQL valides lors de l'utilisation de filtres d'étiquettes combinés avec des expressions booléennes. Avec cette mise à jour, l'implémentation LogQL de LokiStack prend en charge les filtres d'étiquettes avec des expressions booléennes et résout le problème.(LOG-3534)

- Avant cette mise à jour, la ressource personnalisée (CR) **ClusterLogForwarder** ne transmettait pas les informations d'identification TLS pour la sortie syslog à Fluentd, ce qui entraînait des erreurs lors de la transmission. Avec cette mise à jour, les informations d'identification sont correctement transmises à Fluentd, ce qui résout le problème.(LOG-3533)

3.1.4.2. CVE

CVE-2021-46848CVE-2022-3821CVE-2022-35737CVE-2022-42010CVE-2022-42011CVE-2022-42012CVE-2022-42898CVE-2022-43680

3.1.5. Journalisation 5.5.6

Cette version inclut la version 5.5.6 de la correction des bugs de journalisation d'OpenShift .

3.1.5.1. Bug fixes

- Avant cette mise à jour, le contrôleur d'admission Pod Security a ajouté le label **podSecurityLabelSync = true** à l'espace de noms **openshift-logging**. Les étiquettes de sécurité que nous avions spécifiées étaient donc écrasées et les pods Collector ne démarraient pas. Avec cette mise à jour, l'étiquette **podSecurityLabelSync = false** préserve les étiquettes de sécurité. Les pods du collecteur se déploient comme prévu.(LOG-3340)

- Avant cette mise à jour, l'opérateur installait le plugin d'affichage de la console, même s'il n'était pas activé sur le cluster. Cela provoquait le plantage de l'opérateur. Avec cette mise à jour, si un compte pour un cluster n'a pas la vue console activée, l'Opérateur fonctionne normalement et n'installe pas la vue console.(LOG-3407)

- Avant cette mise à jour, une correction antérieure visant à prendre en charge une régression dans laquelle le statut du déploiement d'Elasticsearch n'était pas mis à jour entraînait un plantage de l'opérateur à moins que le site **Red Hat Elasticsearch Operator** ne soit déployé. Avec cette mise à jour, cette correction a été annulée de sorte que l'opérateur est maintenant stable mais réintroduit le problème précédent lié à l'état rapporté.(LOG-3428)

- Avant cette mise à jour, l'Opérateur Loki ne déployait qu'une seule réplique de la passerelle LokiStack quelle que soit la taille de la pile choisie. Avec cette mise à jour, le nombre de répliques est correctement configuré en fonction de la taille choisie.(LOG-3478)

- Avant cette mise à jour, les enregistrements écrits dans Elasticsearch échouaient si plusieurs clés d'étiquettes avaient le même préfixe et si certaines clés comportaient des points. Avec cette mise à jour, les traits de soulignement remplacent les points dans les clés d'étiquettes, ce qui résout le problème.(LOG-3341)

- Avant cette mise à jour, le plugin logging view contenait une fonctionnalité incompatible avec certaines versions d'OpenShift Container Platform. Avec cette mise à jour, la version correcte du plugin résout le problème.(LOG-3467)

- Avant cette mise à jour, la réconciliation de la ressource personnalisée **ClusterLogForwarder** signalait de manière incorrecte un état dégradé d'un ou de plusieurs pipelines, ce qui entraînait le redémarrage des pods collecteurs toutes les 8 à 10 secondes. Avec cette mise à jour, la réconciliation de la ressource personnalisée **ClusterLogForwarder** se déroule correctement, ce qui résout le problème.(LOG-3469)

- Avant cette modification, la spécification du champ **outputDefaults** de la ressource personnalisée ClusterLogForwarder appliquait les paramètres à chaque type de sortie Elasticsearch déclaré. Ce changement corrige le comportement pour correspondre à la spécification d'amélioration où le paramètre s'applique spécifiquement au magasin Elasticsearch géré par défaut.(LOG-3342)

- Avant cette mise à jour, le script **must-gather** de l'OpenShift CLI (oc) ne se terminait pas car l'OpenShift CLI (oc) a besoin d'un dossier avec des droits d'écriture pour construire son cache. Avec cette mise à jour, l'OpenShift CLI (oc) a des droits d'écriture sur un dossier, et le script **must-gather** se termine avec succès. (LOG-3472)

- Avant cette mise à jour, le serveur webhook de Loki Operator provoquait des erreurs TLS. Avec cette mise à jour, l'ICP du webhook de Loki Operator est gérée par la gestion dynamique du webhook de Operator Lifecycle Manager, ce qui résout le problème.(LOG-3511)

3.1.5.2. CVE

- CVE-2021-46848
- CVE-2022-2056
- CVE-2022-2057
- CVE-2022-2058
- CVE-2022-2519
- CVE-2022-2520
- CVE-2022-2521
- CVE-2022-2867
- CVE-2022-2868
- CVE-2022-2869
- CVE-2022-2953
- CVE-2022-2964
- CVE-2022-4139
- CVE-2022-35737
- CVE-2022-42010
- CVE-2022-42011
- CVE-2022-42012

- CVE-2022-42898
- CVE-2022-43680

3.1.6. Journalisation 5.5.5

Cette version inclut la version 5.5.5 de la correction des bugs de journalisation d'OpenShift .

3.1.6.1. Bug fixes

- Avant cette mise à jour, Kibana avait un délai d'expiration du cookie OAuth fixe **24h**, ce qui entraînait des erreurs 401 dans Kibana chaque fois que le champ **accessTokenInactivityTimeout** était défini sur une valeur inférieure à **24h**. Avec cette mise à jour, le délai d'expiration du cookie OAuth de Kibana se synchronise sur le champ **accessTokenInactivityTimeout**, avec une valeur par défaut de **24h**.(LOG-3305)

- Avant cette mise à jour, Vector analysait le champ message lorsque l'analyse JSON était activée sans définir les valeurs **structuredTypeKey** ou **structuredTypeName**. Avec cette mise à jour, une valeur est requise pour **structuredTypeKey** ou **structuredTypeName** lors de l'écriture de journaux structurés dans Elasticsearch.(LOG-3284)

- Avant cette mise à jour, l'alerte **FluentdQueueLengthIncreasing** pouvait ne pas se déclencher en cas de problème de cardinalité avec l'ensemble des étiquettes renvoyées par cette expression d'alerte. Cette mise à jour réduit les étiquettes pour n'inclure que celles nécessaires à l'alerte.(LOG-3226)

- Avant cette mise à jour, Loki n'avait pas de support pour atteindre un stockage externe dans un cluster déconnecté. Avec cette mise à jour, les variables d'environnement proxy et les bundles d'autorité de certification proxy sont inclus dans l'image du conteneur pour prendre en charge ces connexions.(LOG-2860)

- Avant cette mise à jour, les utilisateurs de la console web d'OpenShift Container Platform ne pouvaient pas choisir l'objet **ConfigMap** qui inclut le certificat CA pour Loki, ce qui faisait que les pods fonctionnaient sans le CA. Avec cette mise à jour, les utilisateurs de la console web peuvent sélectionner la carte de configuration, ce qui résout le problème.(LOG-3310)

- Avant cette mise à jour, la clé de l'autorité de certification était utilisée comme nom de volume pour le montage de l'autorité de certification dans Loki, ce qui provoquait des erreurs lorsque la clé de l'autorité de certification comportait des caractères non conformes (tels que des points). Avec cette mise à jour, le nom de volume est normalisé à une chaîne interne, ce qui résout le problème.(LOG-3332)

3.1.6.2. CVE

- CVE-2016-3709
- CVE-2020-35525
- CVE-2020-35527
- CVE-2020-36516
- CVE-2020-36558
- CVE-2021-3640

- CVE-2021-30002
- CVE-2022-0168
- CVE-2022-0561
- CVE-2022-0562
- CVE-2022-0617
- CVE-2022-0854
- CVE-2022-0865
- CVE-2022-0891
- CVE-2022-0908
- CVE-2022-0909
- CVE-2022-0924
- CVE-2022-1016
- CVE-2022-1048
- CVE-2022-1055
- CVE-2022-1184
- CVE-2022-1292
- CVE-2022-1304
- CVE-2022-1355
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1852
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2078
- CVE-2022-2097
- CVE-2022-2509
- CVE-2022-2586
- CVE-2022-2639

- CVE-2022-2938
- CVE-2022-3515
- CVE-2022-20368
- CVE-2022-21499
- CVE-2022-21618
- CVE-2022-21619
- CVE-2022-21624
- CVE-2022-21626
- CVE-2022-21628
- CVE-2022-22624
- CVE-2022-22628
- CVE-2022-22629
- CVE-2022-22662
- CVE-2022-22844
- CVE-2022-23960
- CVE-2022-24448
- CVE-2022-25255
- CVE-2022-26373
- CVE-2022-26700
- CVE-2022-26709
- CVE-2022-26710
- CVE-2022-26716
- CVE-2022-26717
- CVE-2022-26719
- CVE-2022-27404
- CVE-2022-27405
- CVE-2022-27406
- CVE-2022-27950
- CVE-2022-28390

- CVE-2022-28893
- CVE-2022-29581
- CVE-2022-30293
- CVE-2022-34903
- CVE-2022-36946
- CVE-2022-37434
- CVE-2022-39399

3.1.7. Journalisation 5.5.4

Cette version inclut la version 5.5.4 de la correction des bugs de journalisation d'OpenShift .

3.1.7.1. Bug fixes

- Avant cette mise à jour, une erreur dans l'analyseur de requêtes du plugin logging view entraînait la disparition de certaines parties de la requête de logs si celle-ci contenait des parenthèses curly **{}**. Cela rendait les requêtes invalides, ce qui entraînait le renvoi d'erreurs pour des requêtes valides. Avec cette mise à jour, l'analyseur traite correctement ces requêtes.(LOG-3042)
- Avant cette mise à jour, l'opérateur pouvait entrer dans une boucle de suppression et de recréation du daemonset du collecteur pendant que les déploiements Elasticsearch ou Kibana changeaient d'état. Avec cette mise à jour, une correction dans la gestion du statut de l'opérateur résout le problème.(LOG-3049)
- Avant cette mise à jour, aucune alerte n'était mise en œuvre pour prendre en charge l'implémentation du collecteur Vector. Cette modification ajoute des alertes Vector et déploie des alertes distinctes, en fonction de l'implémentation du collecteur choisie.(LOG-3127)
- Avant cette mise à jour, le composant de création de secret de l'Elasticsearch Operator modifiait constamment les secrets internes. Avec cette mise à jour, le secret existant est correctement géré.(LOG-3138)
- Avant cette mise à jour, une refonte des scripts de journalisation **must-gather** a supprimé l'emplacement prévu pour les artefacts. Cette mise à jour annule ce changement pour écrire les artefacts dans le dossier /**must-gather**.(LOG-3213)
- Avant cette mise à jour, sur certains clusters, l'exportateur Prometheus se liait à IPv4 au lieu d'IPv6. Après cette mise à jour, Fluentd détecte la version IP et se lie à **0.0.0.0** pour IPv4 ou **[::]** pour IPv6.(LOG-3162)

3.1.7.2. CVE

- CVE-2020-35525
- CVE-2020-35527
- CVE-2022-0494
- CVE-2022-1353

- CVE-2022-2509
- CVE-2022-2588
- CVE-2022-3515
- CVE-2022-21618
- CVE-2022-21619
- CVE-2022-21624
- CVE-2022-21626
- CVE-2022-21628
- CVE-2022-23816
- CVE-2022-23825
- CVE-2022-29900
- CVE-2022-29901
- CVE-2022-32149
- CVE-2022-37434
- CVE-2022-40674

3.1.8. Journalisation 5.5.3

Cette version inclut la version 5.5.3 de la correction des bugs de journalisation d'OpenShift .

3.1.8.1. Bug fixes

- Avant cette mise à jour, les entrées de journal comportant des messages structurés incluaient le champ du message original, ce qui augmentait la taille de l'entrée. Cette mise à jour supprime le champ de message pour les journaux structurés afin de réduire la taille de l'entrée.(LOG-2759)
- Avant cette mise à jour, la configuration du collecteur excluait les journaux des pods **collector**, **default-log-store**, et **visualization**, mais n'était pas en mesure d'exclure les journaux archivés dans un fichier **.gz**. Avec cette mise à jour, les journaux archivés stockés dans les fichiers **.gz** des pods **collector**, **default-log-store** et **visualization** sont également exclus. (LOG-2844)
- Avant cette mise à jour, lorsque des requêtes vers un pod indisponible étaient envoyées via la passerelle, aucune alerte ne prévenait de l'interruption. Avec cette mise à jour, des alertes individuelles seront générées si la passerelle a des problèmes pour terminer une requête d'écriture ou de lecture.(LOG-2884)
- Avant cette mise à jour, les métadonnées de pods pouvaient être modifiées par des plugins fluents car les valeurs passaient par le pipeline par référence. Cette mise à jour assure que chaque message de log reçoit une copie des métadonnées du pod afin que chaque message soit traité indépendamment.(LOG-3046)
- Avant cette mise à jour, la sélection de la gravité **unknown** dans la vue des journaux de la console OpenShift excluait les journaux avec une valeur **level=unknown**. Avec cette mise à

jour, les journaux sans niveau et avec des valeurs **level=unknown** sont visibles lors du filtrage par gravité **unknown**.(LOG-3062)

- Avant cette mise à jour, les enregistrements de logs envoyés à Elasticsearch avaient un champ supplémentaire nommé **write-index** qui contenait le nom de l'index vers lequel les logs devaient être envoyés. Ce champ ne fait pas partie du modèle de données. Après cette mise à jour, ce champ n'est plus envoyé.(LOG-3075)

- Avec l'introduction du nouveau contrôleur d'admission à la sécurité des pods intégré, les pods qui ne sont pas configurés conformément aux normes de sécurité définies globalement ou au niveau de l'espace de noms ne peuvent pas être exécutés. Avec cette mise à jour, l'opérateur et les collecteurs permettent une exécution privilégiée et s'exécutent sans avertissement ni erreur d'audit de sécurité.(LOG-3077)

- Avant cette mise à jour, l'opérateur supprimait toutes les sorties personnalisées définies dans la ressource personnalisée **ClusterLogForwarder** lorsqu'il utilisait LokiStack comme stockage de logs par défaut. Avec cette mise à jour, l'opérateur fusionne les sorties personnalisées avec les sorties par défaut lors du traitement de la ressource personnalisée **ClusterLogForwarder**.(LOG-3095)

3.1.8.2. CVE

- CVE-2015-20107
- CVE-2022-0391
- CVE-2022-2526
- CVE-2022-21123
- CVE-2022-21125
- CVE-2022-21166
- CVE-2022-29154
- CVE-2022-32206
- CVE-2022-32208
- CVE-2022-34903

3.1.9. Journalisation 5.5.2

Cette version inclut la version 5.5.2 de la correction des bugs de journalisation d'OpenShift .

3.1.9.1. Bug fixes

- Avant cette mise à jour, les règles d'alerte pour le collecteur Fluentd n'adhéraient pas aux directives de style de surveillance de OpenShift Container Platform. Cette mise à jour modifie ces alertes pour inclure l'étiquette de l'espace de noms, ce qui résout le problème.(LOG-1823)

- Avant cette mise à jour, le script de basculement de la gestion des index ne parvenait pas à générer un nouveau nom d'index lorsque le nom de l'index comportait plus d'un trait d'union. Avec cette mise à jour, les noms d'index sont générés correctement.(LOG-2644)

- Avant cette mise à jour, la route Kibana définissait une valeur **caCertificate** sans qu'un certificat soit présent. Avec cette mise à jour, aucune valeur **caCertificate** n'est définie. (LOG-2661)

- Avant cette mise à jour, un changement dans les dépendances du collecteur provoquait l'émission d'un message d'avertissement pour les paramètres non utilisés. Avec cette mise à jour, la suppression des paramètres de configuration inutilisés résout le problème.(LOG-2859)

- Avant cette mise à jour, les pods créés pour les déploiements créés par l'opérateur Loki étaient planifiés par erreur sur des nœuds avec des systèmes d'exploitation non-Linux, si de tels nœuds étaient disponibles dans le cluster dans lequel l'opérateur s'exécutait. Avec cette mise à jour, l'opérateur attache un sélecteur de nœud supplémentaire aux définitions de pods qui permet uniquement de planifier les pods sur des nœuds basés sur Linux.(LOG-2895)

- Avant cette mise à jour, la vue Logs de la console OpenShift ne filtrait pas les logs par gravité en raison d'un problème d'analyseur LogQL dans la passerelle LokiStack. Avec cette mise à jour, un correctif d'analyseur résout le problème et la vue Logs de la console OpenShift peut filtrer par gravité.(LOG-2908)

- Avant cette mise à jour, une refonte des plugins du collecteur Fluentd a supprimé le champ timestamp pour les événements. Cette mise à jour rétablit le champ timestamp, qui provient de l'heure de réception de l'événement.(LOG-2923)

- Avant cette mise à jour, l'absence d'un champ **level** dans les journaux d'audit provoquait une erreur dans les journaux vectoriels. Avec cette mise à jour, l'ajout d'un champ **level** dans l'enregistrement du journal d'audit résout le problème.(LOG-2961)

- Avant cette mise à jour, si vous supprimiez la ressource personnalisée Kibana, la console web de OpenShift Container Platform continuait à afficher un lien vers Kibana. Avec cette mise à jour, la suppression de la ressource personnalisée Kibana supprime également ce lien.(LOG-3053)

- Avant cette mise à jour, chaque travail de basculement créait des index vides lorsque la ressource personnalisée **ClusterLogForwarder** avait une analyse JSON définie. Avec cette mise à jour, les nouveaux index ne sont pas vides(LOG-3063)

- Avant cette mise à jour, lorsque l'utilisateur supprimait la LokiStack après une mise à jour vers Loki Operator 5.5, les ressources créées à l'origine par Loki Operator 5.4 étaient conservées. Avec cette mise à jour, les références propriétaires des ressources pointent vers la LokiStack 5.5.(LOG-2945)

- Avant cette mise à jour, un utilisateur n'était pas en mesure de voir les journaux d'application des espaces de noms auxquels il avait accès. Avec cette mise à jour, l'opérateur Loki crée automatiquement un rôle de cluster et un lien de rôle de cluster permettant aux utilisateurs de lire les journaux d'application.(LOG-2918)

- Avant cette mise à jour, les utilisateurs ayant des privilèges d'administrateur de cluster n'étaient pas en mesure de visualiser correctement les journaux d'infrastructure et d'audit à l'aide de la console de journalisation. Avec cette mise à jour, le contrôle des autorisations a été étendu pour reconnaître également les utilisateurs des groupes cluster-admin et dedicated-admin en tant qu'administrateurs.(LOG-2970)

3.1.9.2. CVE

- CVE-2015-20107

- CVE-2022-0391

- CVE-2022-21123

- CVE-2022-21125
- CVE-2022-21166
- CVE-2022-29154
- CVE-2022-32206
- CVE-2022-32208
- CVE-2022-34903

3.1.10. Journalisation 5.5.1

Cette version inclut la version 5.5.1 de la correction des bugs de journalisation d'OpenShift .

3.1.10.1. Améliorations

- Cette amélioration ajoute un onglet **Aggregated Logs** à la page **Pod Details** de la console web OpenShift Container Platform lorsque le plug-in Logging Console est utilisé. Cette amélioration n'est disponible que sur OpenShift Container Platform 4.10 et plus.(LOG-2647)
- Cette amélioration ajoute Google Cloud Logging comme option de sortie pour la redirection des journaux.(LOG-1482)

3.1.10.2. Bug fixes

- Avant cette mise à jour, l'opérateur ne s'assurait pas que le module était prêt, ce qui entraînait un état inopérant du cluster lors d'un redémarrage. Avec cette mise à jour, l'opérateur marque les nouveaux pods comme étant prêts avant de passer à un nouveau pod lors d'un redémarrage, ce qui résout le problème.(LOG-2745)
- Avant cette mise à jour, Fluentd ne reconnaissait parfois pas que la plateforme Kubernetes effectuait une rotation du fichier de log et ne lisait plus les messages de log. Cette mise à jour corrige cela en définissant le paramètre de configuration suggéré par l'équipe de développement en amont.(LOG-2995)
- Avant cette mise à jour, l'ajout de la détection des erreurs multilignes entraînait une modification du routage interne et l'acheminement des enregistrements vers la mauvaise destination. Avec cette mise à jour, le routage interne est correct.(LOG-2801)
- Avant cette mise à jour, la modification de l'intervalle de rafraîchissement de la console web d'OpenShift Container Platform créait une erreur lorsque le champ **Query** était vide. Avec cette mise à jour, la modification de l'intervalle n'est pas une option disponible lorsque le champ **Query** est vide.(LOG-2917)

3.1.10.3. CVE

- CVE-2022-1705
- CVE-2022-2526
- CVE-2022-29154
- CVE-2022-30631

- CVE-2022-32148
- CVE-2022-32206
- CVE-2022-32208

3.1.11. Journalisation 5.5.0

Cette version comprend:OpenShift Logging Bug Fix Release 5.5.0 .

3.1.11.1. Améliorations

- Avec cette mise à jour, vous pouvez transférer des logs structurés provenant de différents conteneurs au sein d'un même pod vers différents index. Pour utiliser cette fonctionnalité, vous devez configurer le pipeline avec le support multi-conteneurs et annoter les pods.(LOG-1296)

IMPORTANT

Le formatage JSON des journaux varie selon les applications. La création d'un trop grand nombre d'index ayant un impact sur les performances, limitez l'utilisation de cette fonctionnalité à la création d'index pour les journaux dont les formats JSON sont incompatibles. Utilisez des requêtes pour séparer les journaux provenant de différents espaces de noms ou d'applications dont les formats JSON sont compatibles.

- Avec cette mise à jour, vous pouvez filtrer les journaux avec des sorties Elasticsearch en utilisant les étiquettes communes Kubernetes, **app.kubernetes.io/component**, **app.kubernetes.io/managed-by**, **app.kubernetes.io/part-of**, et **app.kubernetes.io/version**. Les types de sorties non Elasticsearch peuvent utiliser toutes les étiquettes incluses dans **kubernetes.labels**.(LOG-2388)

- Avec cette mise à jour, les clusters avec AWS Security Token Service (STS) activé peuvent utiliser l'authentification STS pour transmettre les journaux à Amazon CloudWatch.(LOG-1976)

- Avec cette mise à jour, l'opérateur 'LokiOperator' et le collecteur vectoriel passent de l'aperçu technique à la disponibilité générale. La parité complète des fonctionnalités avec les versions antérieures est en attente, et certaines API restent en avant-première technique. Voir la section **Logging with the LokiStack** pour plus de détails.

3.1.11.2. Bug fixes

- Avant cette mise à jour, les clusters configurés pour transmettre les journaux à Amazon CloudWatch écrivaient les fichiers journaux rejetés dans le stockage temporaire, ce qui entraînait une instabilité du cluster au fil du temps. Avec cette mise à jour, la sauvegarde de morceaux pour toutes les options de stockage a été désactivée, ce qui résout le problème.(LOG-2746)

- Avant cette mise à jour, l'Opérateur utilisait des versions de certaines API qui sont obsolètes et dont la suppression est prévue dans les prochaines versions d'OpenShift Container Platform. Cette mise à jour déplace les dépendances vers les versions d'API prises en charge.(LOG-2656)

- Avant cette mise à jour, plusieurs pipelines **ClusterLogForwarder** configurés pour la détection d'erreurs multilignes provoquaient l'entrée du collecteur dans l'état d'erreur **crashloopbackoff**. Cette mise à jour corrige le problème où plusieurs sections de configuration avaient le même identifiant unique.(LOG-2241)

- Avant cette mise à jour, le collecteur ne pouvait pas enregistrer les symboles non UTF-8 dans les journaux de stockage Elasticsearch. Avec cette mise à jour, le collecteur encode les symboles non UTF-8, ce qui résout le problème.(LOG-2203)

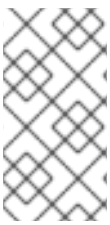
- Avant cette mise à jour, les caractères non latins s'affichaient de manière incorrecte dans Kibana. Avec cette mise à jour, Kibana affiche correctement tous les symboles UTF-8 valides.(LOG-2784)

3.1.11.3. CVE

- CVE-2021-38561
- CVE-2022-1012
- CVE-2022-1292
- CVE-2022-1586
- CVE-2022-1785
- CVE-2022-1897
- CVE-2022-1927
- CVE-2022-2068
- CVE-2022-2097
- CVE-2022-21698
- CVE-2022-30631
- CVE-2022-32250

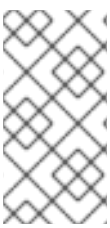
3.2. DÉMARRER AVEC LA JOURNALISATION 5.5

Cette vue d'ensemble du processus de déploiement de la journalisation est fournie à titre de référence. Elle ne remplace pas la documentation complète. Pour les nouvelles installations, les sites **Vector** et **LokiStack** sont recommandés.



NOTE

À partir de la version 5.5, vous pouvez choisir entre les implémentations de collecteurs **Fluentd** ou **Vector** et les magasins de journaux **Elasticsearch** ou **LokiStack**. La documentation relative à la journalisation est en cours de mise à jour afin de refléter ces changements de composants sous-jacents.

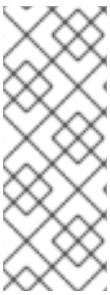


NOTE

Le sous-système de journalisation pour Red Hat OpenShift est fourni en tant que composant installable, avec un cycle de publication distinct de celui de la plateforme principale OpenShift Container Platform. La politique de cycle de vie de Red Hat OpenShift Container Platform décrit la compatibilité des versions.

Conditions préalables

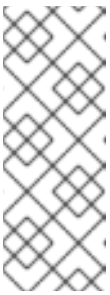
- Préférence LogStore : **Elasticsearch** ou **LokiStack**
- Préférence de mise en œuvre du collecteur : **Fluentd** ou **Vector**
- Informations d'identification pour vos sorties de transfert de logs



NOTE

À partir de la version 5.4.3 de Logging, l'Elasticsearch Operator est obsolète et il est prévu de le supprimer dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Au lieu d'utiliser l'opérateur Elasticsearch pour gérer le stockage des journaux par défaut, vous pouvez utiliser l'opérateur Loki.

1. Installez l'opérateur pour le logstore que vous souhaitez utiliser.
   - Pour **Elasticsearch**, installez le site **OpenShift Elasticsearch Operator**.
   - Pour **LokiStack**, installez le site **Loki Operator**.
     - Créer une instance de ressource personnalisée (CR) à l'adresse **LokiStack**.
2. Installer le site **Red Hat OpenShift Logging Operator**.
3. Créer une instance de ressource personnalisée (CR) à l'adresse **ClusterLogging**.
   a. Sélectionnez la mise en œuvre de votre collecteur.



NOTE

À partir de la version 5.6 de l'exploitation forestière, Fluentd est obsolète et il est prévu qu'il soit supprimé dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Comme alternative à Fluentd, vous pouvez utiliser Vector.

4. Créer une instance de ressource personnalisée (CR) à l'adresse **ClusterLogForwarder**.
5. Créer un secret pour le pipeline de sortie sélectionné.

3.3. COMPRENDRE L'ARCHITECTURE DE LA JOURNALISATION

Le sous-système de journalisation se compose des éléments logiques suivants :

- **Collector** - Lit les données d'enregistrement des conteneurs sur chaque nœud et les transmet aux sorties configurées.
- **Store** - Stocke les données du journal en vue de leur analyse ; c'est la sortie par défaut du transitaire.
- **Visualization** - Interface graphique pour la recherche, l'interrogation et la visualisation des journaux stockés.

Ces composants sont gérés par des opérateurs et des fichiers YAML de ressources personnalisées (CR).

Le sous-système de journalisation de Red Hat OpenShift collecte les journaux des conteneurs et des nœuds. Ceux-ci sont classés par type :

- **application** - Journaux de conteneurs générés par des conteneurs ne faisant pas partie de l'infrastructure.
- **infrastructure** - Les journaux des conteneurs des espaces de noms **kube-*** et **openshift-***, et les journaux des nœuds de **journald**.
- **audit** - Journaux provenant de **auditd**, **kube-apiserver**, **openshift-apiserver**, et **ovn** si l'option est activée.

Le collecteur de logs est un daemonset qui déploie des pods sur chaque nœud d'OpenShift Container Platform. Les journaux du système et de l'infrastructure sont générés par les messages de journald du système d'exploitation, de l'exécution du conteneur et d'OpenShift Container Platform.

Les journaux de conteneurs sont générés par les conteneurs qui s'exécutent dans des pods sur le cluster. Chaque conteneur génère un flux de journaux distinct. Le collecteur recueille les journaux de ces sources et les transmet en interne ou en externe, comme configuré dans la ressource personnalisée **ClusterLogForwarder**.

3.4. ADMINISTRATION DU DÉPLOIEMENT DE LA JOURNALISATION

3.4.1. Déployer Red Hat OpenShift Logging Operator à l'aide de la console web

Vous pouvez utiliser la console web de OpenShift Container Platform pour déployer Red Hat OpenShift Logging Operator.

CONDITIONS PRÉALABLES

Le sous-système de journalisation pour Red Hat OpenShift est fourni en tant que composant installable, avec un cycle de publication distinct de celui de la plateforme principale OpenShift Container Platform. La politique de cycle de vie de Red Hat OpenShift Container Platform décrit la compatibilité des versions.

Procédure

Pour déployer Red Hat OpenShift Logging Operator à l'aide de la console web OpenShift Container Platform :

1. Installez l'opérateur de journalisation Red Hat OpenShift :
   a. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
   b. Tapez **Logging** dans le champ **Filter by keyword**.
   c. Choisissez **Red Hat OpenShift Logging** dans la liste des opérateurs disponibles et cliquez sur **Install**.
   d. Sélectionnez **stable** ou **stable-5.y** comme **Update Channel**.

NOTE

Le canal **stable** ne fournit des mises à jour que pour la version la plus récente du logiciel d'exploitation. Pour continuer à recevoir les mises à jour des versions antérieures, vous devez changer votre canal d'abonnement pour **stable-X**, où **X** est la version de l'exploitation que vous avez installée.

e. Assurez-vous que **A specific namespace on the cluster**est sélectionné sous **Installation Mode**.

f. Assurez-vous que **Operator recommended namespace** est **openshift-logging** sous **Installed Namespace**.

g. Sélectionnez **Enable Operator recommended cluster monitoring on this Namespace**

h. Sélectionnez une option pour **Update approval**.

- L'option **Automatic** permet à Operator Lifecycle Manager (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible.
- L'option **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.

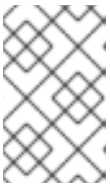
i. Sélectionnez **Enable** ou **Disable** pour le plugin Console.

j. Cliquez sur **Install**.

2. Vérifiez que le site **Red Hat OpenShift Logging Operator** est installé en passant à la page **Operators → Installed Operators**.

a. Assurez-vous que **Red Hat OpenShift Logging** est listé dans le projet **openshift-logging** avec un **Status** de **Succeeded**.

3. Créer une instance **ClusterLogging**.



NOTE

La vue du formulaire de la console web ne comprend pas toutes les options disponibles. Il est recommandé d'utiliser le site **YAML view** pour compléter votre installation.

a. Dans la section **collection**, sélectionnez une implémentation de collecteur.

NOTE

À partir de la version 5.6 de l'exploitation forestière, Fluentd est obsolète et il est prévu qu'il soit supprimé dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Comme alternative à Fluentd, vous pouvez utiliser Vector.

b. Dans la section **logStore**, sélectionnez un type.

NOTE

À partir de la version 5.4.3 de Logging, l'Elasticsearch Operator est obsolète et il est prévu de le supprimer dans une prochaine version. Red Hat fournira des corrections de bogues et une assistance pour cette fonctionnalité pendant le cycle de vie de la version actuelle, mais cette fonctionnalité ne recevra plus d'améliorations et sera supprimée. Au lieu d'utiliser l'opérateur Elasticsearch pour gérer le stockage des journaux par défaut, vous pouvez utiliser l'opérateur Loki.

c. Cliquez sur **Create**.

3.4.2. Déploiement de l'opérateur Loki à l'aide de la console web

Vous pouvez utiliser la console web d'OpenShift Container Platform pour installer l'opérateur Loki.

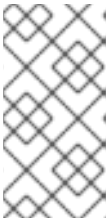
Conditions préalables

- Log Store pris en charge (AWS S3, Google Cloud Storage, Azure, Swift, Minio, OpenShift Data Foundation)

Procédure

Pour installer l'opérateur Loki à l'aide de la console web d'OpenShift Container Platform :

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
2. Tapez **Loki** dans le champ **Filter by keyword**.
   a. Choisissez **Loki Operator** dans la liste des opérateurs disponibles et cliquez sur **Install**.
3. Sélectionnez **stable** ou **stable-5.y** comme **Update Channel**.



NOTE

Le canal **stable** ne fournit des mises à jour que pour la version la plus récente du logiciel d'exploitation. Pour continuer à recevoir les mises à jour des versions antérieures, vous devez changer votre canal d'abonnement pour **stable-X**, où **X** est la version de l'exploitation que vous avez installée.

4. Assurez-vous que **All namespaces on the cluster**est sélectionné sous **Installation Mode**.
5. Assurez-vous que **openshift-operators-redhat** est sélectionné sous **Installed Namespace**.
6. Sélectionnez **Enable Operator recommended cluster monitoring on this Namespace** Cette option définit l'étiquette **openshift.io/cluster-monitoring: "true"** dans l'objet Namespace. Vous devez sélectionner cette option pour vous assurer que la surveillance des clusters récupère l'espace de noms **openshift-operators-redhat**.
7. Sélectionnez une option pour **Update approval**.
   - L'option **Automatic** permet à Operator Lifecycle Manager (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible.
   - L'option **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.

8. Cliquez sur **Install**.

9. Vérifiez que le site **LokiOperator** est installé en passant à la page **Operators → Installed Operators**.

   a. Veillez à ce que **LokiOperator** soit listé avec **Status** et **Succeeded** dans tous les projets.

10. Créez un fichier YAML **Secret** qui utilise les champs **access_key_id** et **access_key_secret** pour spécifier vos informations d'identification et **bucketnames**, **endpoint**, et **region** pour définir l'emplacement de stockage de l'objet. AWS est utilisé dans l'exemple suivant :

```
apiVersion: v1
kind: Secret
metadata:
  name: logging-loki-s3
  namespace: openshift-logging
stringData:
  access_key_id: AKIAIOSFODNN7EXAMPLE
  access_key_secret: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
  bucketnames: s3-bucket-name
  endpoint: https://s3.eu-central-1.amazonaws.com
  region: eu-central-1
```

11. Sélectionnez **Create instance** sous LokiStack dans l'onglet **Details**. Sélectionnez ensuite **YAML view**. Collez le modèle suivant, en remplaçant les valeurs le cas échéant.

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki (1)
  namespace: openshift-logging
spec:
  size: 1x.small (2)
  storage:
    schemas:
    - version: v12
      effectiveDate: '2022-06-01'
    secret:
      name: logging-loki-s3 (3)
      type: s3 (4)
  storageClassName: <storage_class_name> (5)
  tenants:
    mode: openshift-logging
```

(1) Le nom doit être **logging-loki**.

(2) Sélectionnez la taille de déploiement de votre Loki.

(3) Définissez le secret utilisé pour le stockage des journaux.

(4) Définir le type de stockage correspondant.

(5) Saisissez le nom d'une classe de stockage existante pour le stockage temporaire. Pour de meilleures performances, spécifiez une classe de stockage qui alloue des blocs de stockage. Les classes de stockage disponibles pour votre cluster peuvent être répertoriées

à l'aide de **oc get storageclasses**.

a. Appliquer la configuration :

```
oc apply -f logging-loki.yaml
```

12. Créer ou modifier un CR **ClusterLogging**:

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  managementState: Managed
  logStore:
    type: lokistack
    lokistack:
      name: logging-loki
    collection:
      type: vector
```

a. Appliquer la configuration :

```
oc apply -f cr-lokistack.yaml
```

3.4.3. Installation à partir d'OperatorHub en utilisant le CLI

Au lieu d'utiliser la console web de OpenShift Container Platform, vous pouvez installer un Operator depuis OperatorHub en utilisant le CLI. Utilisez la commande **oc** pour créer ou mettre à jour un objet **Subscription**.

Conditions préalables

- Accès à un cluster OpenShift Container Platform à l'aide d'un compte disposant des autorisations **cluster-admin**.
- Installez la commande **oc** sur votre système local.

Procédure

1. Voir la liste des opérateurs disponibles pour la grappe à partir d'OperatorHub :

```
$ oc get packagemanifests -n openshift-marketplace
```

Exemple de sortie

```
NAME                              CATALOG               AGE
3scale-operator                   Red Hat Operators     91m
advanced-cluster-management       Red Hat Operators     91m
amq7-cert-manager                 Red Hat Operators     91m
...
couchbase-enterprise-certified    Certified Operators   91m
```

```
crunchy-postgres-operator        Certified Operators   91m
mongodb-enterprise               Certified Operators   91m
...
etcd                          Community Operators   91m
jaeger                        Community Operators   91m
kubefed                       Community Operators   91m
...
```

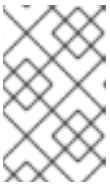
Notez le catalogue de l'opérateur souhaité.

2. Inspectez l'opérateur de votre choix pour vérifier les modes d'installation pris en charge et les canaux disponibles :

```
oc describe packagemanifests <operator_name> -n openshift-marketplace
```

3. Un groupe d'opérateurs, défini par un objet **OperatorGroup**, sélectionne des espaces de noms cibles dans lesquels générer l'accès RBAC requis pour tous les opérateurs dans le même espace de noms que le groupe d'opérateurs.
L'espace de noms auquel vous abonnez l'opérateur doit avoir un groupe d'opérateurs qui correspond au mode d'installation de l'opérateur, soit le mode **AllNamespaces** ou **SingleNamespace**. Si l'opérateur que vous avez l'intention d'installer utilise le mode **AllNamespaces**, l'espace de noms **openshift-operators** dispose déjà d'un groupe d'opérateurs approprié.

   Cependant, si l'opérateur utilise le mode **SingleNamespace** et que vous n'avez pas déjà un groupe d'opérateurs approprié en place, vous devez en créer un.



NOTE

La version console web de cette procédure gère la création des objets **OperatorGroup** et **Subscription** automatiquement dans les coulisses lorsque vous choisissez le mode **SingleNamespace**.

   a. Créez un fichier YAML de l'objet **OperatorGroup**, par exemple **operatorgroup.yaml**:

Exemple d'objet OperatorGroup

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <operatorgroup_name>
  namespace: <namespace>
spec:
  targetNamespaces:
  - <namespace>
```

   b. Créer l'objet **OperatorGroup**:

```
$ oc apply -f operatorgroup.yaml
```

4. Créez un fichier YAML de l'objet **Subscription** pour abonner un espace de noms à un opérateur, par exemple **sub.yaml**:

Exemple d'objet Subscription

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: <subscription_name>
  namespace: openshift-operators 1
spec:
  channel: <channel_name> 2
  name: <operator_name> 3
  source: redhat-operators 4
  sourceNamespace: openshift-marketplace 5
  config:
    env: 6
    - name: ARGS
      value: "-v=10"
    envFrom: 7
    - secretRef:
        name: license-secret
    volumes: 8
    - name: <volume_name>
      configMap:
        name: <configmap_name>
    volumeMounts: 9
    - mountPath: <directory_name>
      name: <volume_name>
    tolerations: 10
    - operator: "Exists"
    resources: 11
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
    nodeSelector: 12
      foo: bar
```

1. Pour l'utilisation du mode d'installation **AllNamespaces**, indiquez l'espace de noms **openshift-operators**. Sinon, indiquez l'espace de noms unique correspondant à l'utilisation du mode d'installation **SingleNamespace**.
2. Nom du canal auquel s'abonner.
3. Nom de l'opérateur auquel s'abonner.
4. Nom de la source du catalogue qui fournit l'opérateur.
5. Espace de noms de la source de catalogue. Utilisez **openshift-marketplace** pour les sources de catalogue par défaut d'OperatorHub.
6. Le paramètre **env** définit une liste de variables d'environnement qui doivent exister dans tous les conteneurs du module créé par OLM.
7. Le paramètre **envFrom** définit une liste de sources pour alimenter les variables d'environnement dans le conteneur.

8. Le paramètre **volumes** définit une liste de volumes qui doivent exister sur le pod créé par OLM.

9. Le paramètre **volumeMounts** définit une liste de VolumeMounts qui doivent exister dans tous les conteneurs du pod créé par OLM. Si un **volumeMount** fait référence à un **volume** qui n'existe pas, OLM ne parvient pas à déployer l'opérateur.

10. Le paramètre **tolerations** définit une liste de tolérances pour le module créé par OLM.

11. Le paramètre **resources** définit les contraintes de ressources pour tous les conteneurs du module créé par OLM.

12. Le paramètre **nodeSelector** définit un **NodeSelector** pour le module créé par OLM.

5. Créer l'objet **Subscription**:

```
$ oc apply -f sub.yaml
```

A ce stade, OLM connaît l'opérateur sélectionné. Une version de service de cluster (CSV) pour l'opérateur devrait apparaître dans l'espace de noms cible, et les API fournies par l'opérateur devraient être disponibles pour la création.

3.4.4. Suppression d'opérateurs d'une grappe à l'aide de la console web

Les administrateurs de cluster peuvent supprimer les opérateurs installés dans un espace de noms sélectionné à l'aide de la console web.

Conditions préalables

- Vous avez accès à la console web d'un cluster OpenShift Container Platform en utilisant un compte avec les permissions **cluster-admin**.

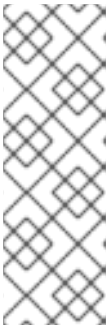
Procédure

1. Naviguez jusqu'à la page **Operators → Installed Operators**.

2. Faites défiler ou saisissez un mot-clé dans le champ **Filter by name** pour trouver l'opérateur que vous souhaitez supprimer. Cliquez ensuite dessus.

3. Sur le côté droit de la page **Operator Details**, sélectionnez **Uninstall Operator** dans la liste **Actions**.
   Une boîte de dialogue **Uninstall Operator?** s'affiche.

4. Sélectionnez **Uninstall** pour supprimer l'opérateur, les déploiements de l'opérateur et les pods. Suite à cette action, l'opérateur cesse de fonctionner et ne reçoit plus de mises à jour.



NOTE

Cette action ne supprime pas les ressources gérées par l'opérateur, y compris les définitions de ressources personnalisées (CRD) et les ressources personnalisées (CR). Les tableaux de bord et les éléments de navigation activés par la console Web et les ressources hors cluster qui continuent de fonctionner peuvent nécessiter un nettoyage manuel. Pour les supprimer après la désinstallation de l'opérateur, vous devrez peut-être supprimer manuellement les CRD de l'opérateur.

3.4.5. Suppression d'opérateurs d'une grappe à l'aide de la CLI

Les administrateurs de clusters peuvent supprimer les opérateurs installés dans un espace de noms sélectionné à l'aide de l'interface de ligne de commande.

Conditions préalables

- Accès à un cluster OpenShift Container Platform à l'aide d'un compte disposant des autorisations **cluster-admin**.
- **oc** installée sur le poste de travail.

Procédure

1. Vérifiez la version actuelle de l'opérateur souscrit (par exemple, **jaeger**) dans le champ **currentCSV**:

```
$ oc get subscription jaeger -n openshift-operators -o yaml | grep currentCSV
```

Exemple de sortie

```
  currentCSV: jaeger-operator.v1.8.2
```

2. Supprimer l'abonnement (par exemple, **jaeger**) :

```
$ oc delete subscription jaeger -n openshift-operators
```

Exemple de sortie

```
subscription.operators.coreos.com "jaeger" deleted
```

3. Supprimez le CSV de l'opérateur dans l'espace de noms cible en utilisant la valeur **currentCSV** de l'étape précédente :

```
$ oc delete clusterserviceversion jaeger-operator.v1.8.2 -n openshift-operators
```

Exemple de sortie

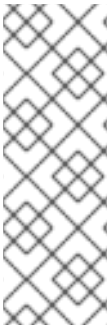
```
clusterserviceversion.operators.coreos.com "jaeger-operator.v1.8.2" deleted
```

CHAPITRE 4. COMPRENDRE LE SOUS-SYSTÈME DE JOURNALISATION POUR RED HAT OPENSHIFT

En tant qu'administrateur de cluster, vous pouvez déployer le sous-système de journalisation pour agréger tous les journaux de votre cluster OpenShift Container Platform, tels que les journaux d'audit du système de nœuds, les journaux de conteneurs d'applications et les journaux d'infrastructure. Le sous-système de journalisation regroupe ces journaux provenant de l'ensemble de votre cluster et les stocke dans un magasin de journaux par défaut. Vous pouvez utiliser la console web Kibana pour visualiser les données des journaux.

Le sous-système de journalisation regroupe les types de journaux suivants :

- **application** - Journaux de conteneurs générés par les applications utilisateur exécutées dans le cluster, à l'exception des applications de conteneurs d'infrastructure.
- **infrastructure** - Les journaux générés par les composants d'infrastructure fonctionnant dans le cluster et les nœuds OpenShift Container Platform, tels que les journaux. Les composants d'infrastructure sont des pods qui s'exécutent dans les projets **openshift***, **kube***, ou **default**.
- **audit** - Journaux générés par auditd, le système d'audit des nœuds, qui sont stockés dans le fichier **/var/log/audit/audit.log**, et les journaux d'audit de l'apiserver Kubernetes et de l'apiserver OpenShift.



NOTE

Étant donné que le magasin de journaux Elasticsearch interne d'OpenShift Container Platform ne fournit pas de stockage sécurisé pour les journaux d'audit, les journaux d'audit ne sont pas stockés dans l'instance Elasticsearch interne par défaut. Si vous souhaitez envoyer les journaux d'audit au magasin de journaux Elasticsearch interne par défaut, par exemple pour afficher les journaux d'audit dans Kibana, vous devez utiliser l'API Log Forwarding comme décrit dans Transférer les journaux d'audit vers le magasin de journaux.

4.1. GLOSSAIRE DES TERMES COURANTS POUR LA JOURNALISATION DE LA PLATEFORME OPENSHIFT CONTAINER PLATFORM

Ce glossaire définit les termes communs utilisés dans le contenu de l'OpenShift Container Platform Logging.

annotation

Vous pouvez utiliser des annotations pour attacher des métadonnées aux objets.

Opérateur de journalisation de cluster (CLO)

L'opérateur de journalisation du cluster fournit un ensemble d'API permettant de contrôler la collecte et la transmission des journaux d'application, d'infrastructure et d'audit.

Ressource personnalisée (CR)

Un CR est une extension de l'API Kubernetes. Pour configurer OpenShift Container Platform Logging et log forwarding, vous pouvez personnaliser les ressources personnalisées **ClusterLogging** et **ClusterLogForwarder**.

routeur d'événements

Le routeur d'événements est un pod qui surveille les événements de OpenShift Container Platform. Il collecte les logs en utilisant OpenShift Container Platform Logging.

Fluentd

Fluentd est un collecteur de logs qui réside sur chaque nœud d'OpenShift Container Platform. Il rassemble les logs d'application, d'infrastructure et d'audit et les transmet à différentes sorties.

collecte des ordures

La collecte de déchets est le processus de nettoyage des ressources du cluster, telles que les conteneurs et les images terminés qui ne sont pas référencés par les pods en cours d'exécution.

Elasticsearch

Elasticsearch est un moteur de recherche et d'analyse distribué. OpenShift Container Platform utilise ELasticsearch comme magasin de logs par défaut pour OpenShift Container Platform Logging.

Opérateur Elasticsearch

L'opérateur Elasticsearch est utilisé pour faire fonctionner un cluster Elasticsearch au-dessus d'OpenShift Container Platform. L'opérateur Elasticsearch fournit un libre-service pour les opérations du cluster Elasticsearch et est utilisé par OpenShift Container Platform Logging.

indexation

L'indexation est une technique de structure de données utilisée pour localiser et accéder rapidement aux données. L'indexation optimise les performances en minimisant le nombre d'accès au disque requis lors du traitement d'une requête.

Journalisation JSON

L'API Log Forwarding d'OpenShift Container Platform permet d'analyser les logs JSON en un objet structuré et de les transmettre à Elasticsearch, géré par OpenShift Container Platform Logging, ou à tout autre système tiers pris en charge par l'API Log Forwarding.

Kibana

Kibana est une interface de console basée sur un navigateur pour interroger, découvrir et visualiser vos données Elasticsearch par le biais d'histogrammes, de graphiques linéaires et de graphiques circulaires.

Serveur API Kubernetes

Le serveur API Kubernetes valide et configure les données pour les objets API.

Étiquettes

Les étiquettes sont des paires clé-valeur que vous pouvez utiliser pour organiser et sélectionner des sous-ensembles d'objets, tels qu'un pod.

Enregistrement

Avec OpenShift Container Platform Logging, vous pouvez agréger les logs d'application, d'infrastructure et d'audit à travers votre cluster. Vous pouvez également les stocker dans un magasin de logs par défaut, les transmettre à des systèmes tiers, et interroger et visualiser les logs stockés dans le magasin de logs par défaut.

collecteur de données

Un collecteur de journaux recueille les journaux du cluster, les formate et les transmet au magasin de journaux ou à des systèmes tiers.

magasin de journaux

Un magasin de logs est utilisé pour stocker les logs agrégés. Vous pouvez utiliser le magasin de logs Elasticsearch par défaut ou transférer les logs vers des magasins de logs externes. Le magasin de journaux par défaut est optimisé et testé pour le stockage à court terme.

visualiseur de logs

Le visualiseur de logs est le composant de l'interface utilisateur (IU) que vous pouvez utiliser pour afficher des informations telles que des logs, des graphiques, des diagrammes et d'autres métriques. L'implémentation actuelle est Kibana.

nœud

Un nœud est une machine de travail dans le cluster OpenShift Container Platform. Un nœud est soit une machine virtuelle (VM), soit une machine physique.

Opérateurs

Les opérateurs sont la méthode privilégiée pour conditionner, déployer et gérer une application Kubernetes dans un cluster OpenShift Container Platform. Un opérateur prend les connaissances opérationnelles humaines et les encode dans un logiciel qui est emballé et partagé avec les clients.

nacelle

Un pod est la plus petite unité logique de Kubernetes. Il se compose d'un ou plusieurs conteneurs et s'exécute sur un nœud...

Contrôle d'accès basé sur les rôles (RBAC)

Le RBAC est un contrôle de sécurité essentiel pour garantir que les utilisateurs et les charges de travail des clusters n'ont accès qu'aux ressources nécessaires à l'exécution de leur rôle.

éclats

Elasticsearch organise les données de log de Fluentd en datastores, ou index, puis subdivise chaque index en plusieurs morceaux appelés shards.

souillure

Les taints garantissent que les pods sont planifiés sur les nœuds appropriés. Vous pouvez appliquer un ou plusieurs taints à un nœud.

tolérance

Vous pouvez appliquer des tolérances aux modules. Les tolérances permettent à l'ordonnanceur de programmer des pods dont les taches correspondent.

console web

Une interface utilisateur (UI) pour gérer OpenShift Container Platform.

4.2. À PROPOS DU DÉPLOIEMENT DU SOUS-SYSTÈME DE JOURNALISATION POUR RED HAT OPENSHIFT

Les administrateurs de cluster d'OpenShift Container Platform peuvent déployer le sous-système de journalisation en utilisant la console web d'OpenShift Container Platform ou le CLI pour installer l'OpenShift Elasticsearch Operator et le Red Hat OpenShift Logging Operator. Lorsque les opérateurs sont installés, vous créez une ressource personnalisée (CR) **ClusterLogging** pour planifier les pods du sous-système de journalisation et les autres ressources nécessaires à la prise en charge du sous-système de journalisation. Les opérateurs sont responsables du déploiement, de la mise à niveau et de la maintenance du sous-système de journalisation.

Le CR **ClusterLogging** définit un environnement de sous-système de journalisation complet qui inclut tous les composants de la pile de journalisation pour collecter, stocker et visualiser les journaux. L'opérateur de journalisation de Red Hat OpenShift surveille le CR du sous-système de journalisation et ajuste le déploiement de la journalisation en conséquence.

Les administrateurs et les développeurs d'applications peuvent consulter les journaux des projets pour lesquels ils ont un accès de visualisation.

Pour plus d'informations, voir Installer le sous-système de journalisation pour Red Hat OpenShift .

4.2.1. A propos de JSON Logging OpenShift Container Platform

Vous pouvez utiliser la journalisation JSON pour configurer l'API Log Forwarding afin qu'elle analyse les chaînes JSON dans un objet structuré. Vous pouvez effectuer les tâches suivantes :

- Analyse des journaux JSON
- Configurer les données de journalisation JSON pour Elasticsearch
- Transférer les journaux JSON vers le magasin de journaux Elasticsearch

4.2.2. À propos de la collecte et du stockage des événements Kubernetes

L'OpenShift Container Platform Event Router est un pod qui observe les événements Kubernetes et les enregistre pour qu'ils soient collectés par OpenShift Container Platform Logging. Vous devez déployer manuellement l'Event Router.

Pour plus d'informations, voir À propos de la collecte et du stockage des événements Kubernetes .

4.2.3. A propos de la mise à jour de la journalisation de la plateforme OpenShift Container

OpenShift Container Platform vous permet de mettre à jour la journalisation d'OpenShift Container Platform. Vous devez mettre à jour les opérateurs suivants lors de la mise à jour de la journalisation d'OpenShift Container Platform :

- Opérateur Elasticsearch
- Opérateur de journalisation des clusters

Pour plus d'informations, voir Mise à jour de la journalisation OpenShift .

4.2.4. À propos de l'affichage du tableau de bord de la grappe

Le tableau de bord de journalisation d'OpenShift Container Platform contient des graphiques qui montrent les détails de votre instance Elasticsearch au niveau du cluster. Ces graphiques vous aident à diagnostiquer et à anticiper les problèmes.

Pour plus d'informations, voir À propos de l'affichage du tableau de bord de la grappe .

4.2.5. À propos du dépannage de la plateforme OpenShift Container Platform Logging

Vous pouvez résoudre les problèmes de journalisation en effectuant les tâches suivantes :

- Visualisation de l'état de la journalisation
- Visualisation de l'état du magasin de journaux
- Comprendre les alertes de journalisation
- Collecte de données de journalisation pour Red Hat Support
- Dépannage pour les alertes critiques

4.2.6. A propos de la désinstallation de OpenShift Container Platform Logging

Vous pouvez arrêter l'agrégation des journaux en supprimant la ressource personnalisée ClusterLogging (CR). Après la suppression de la CR, il reste d'autres composants de journalisation de cluster, que vous pouvez éventuellement supprimer.

Pour plus d'informations, voir Désinstallation d'OpenShift Logging.

4.2.7. A propos de l'exportation de champs

Le système de journalisation exporte des champs. Les champs exportés sont présents dans les enregistrements de logs et sont disponibles pour des recherches à partir d'Elasticsearch et de Kibana.

Pour plus d'informations, voir À propos de l'exportation de champs .

4.2.8. À propos des composants du sous-système de journalisation

Les composants du sous-système de journalisation comprennent un collecteur déployé sur chaque nœud du cluster OpenShift Container Platform qui collecte tous les journaux des nœuds et des conteneurs et les écrit dans un magasin de journaux. Vous pouvez utiliser une interface web centralisée pour créer des visualisations et des tableaux de bord riches avec les données agrégées.

Les principaux composants du sous-système de journalisation sont les suivants :

- collection - C'est le composant qui collecte les logs du cluster, les formate et les transmet au magasin de logs. L'implémentation actuelle est Fluentd.
- log store - C'est l'endroit où sont stockés les journaux. L'implémentation par défaut est Elasticsearch. Vous pouvez utiliser le magasin de journaux Elasticsearch par défaut ou transférer les journaux vers des magasins de journaux externes. Le magasin de journaux par défaut est optimisé et testé pour le stockage à court terme.
- visualisation - Il s'agit du composant de l'interface utilisateur que vous pouvez utiliser pour afficher les journaux, les graphiques, les diagrammes, etc. L'implémentation actuelle est Kibana.

Ce document peut faire référence au log store ou à Elasticsearch, à la visualisation ou à Kibana, à la collecte ou à Fluentd, de manière interchangeable, sauf indication contraire.

4.2.9. À propos du collecteur de journalisation

Le sous-système de journalisation de Red Hat OpenShift collecte les journaux des conteneurs et des nœuds.

Par défaut, le collecteur de journaux utilise les sources suivantes :

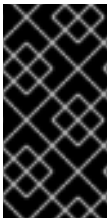
- journald pour tous les journaux du système
- **/var/log/containers/*.log** pour tous les journaux de conteneurs

Si vous configurez le collecteur de journaux pour qu'il recueille les journaux d'audit, il les obtient à partir de **/var/log/audit/audit.log**.

Le collecteur de logs est un ensemble de démons qui déploie des pods sur chaque nœud d'OpenShift Container Platform. Les journaux du système et de l'infrastructure sont générés par les messages de journald du système d'exploitation, du runtime du conteneur et d'OpenShift Container Platform. Les logs d'application sont générés par le moteur de conteneur CRI-O. Fluentd collecte les logs de ces sources et les transmet en interne ou en externe comme vous le configurez dans OpenShift Container Platform.

Les moteurs d'exécution des conteneurs fournissent des informations minimales pour identifier la source des messages de journalisation : le projet, le nom du pod et l'ID du conteneur. Ces informations ne sont pas suffisantes pour identifier de manière unique la source des journaux. Si un module portant

un nom et un projet donnés est supprimé avant que le collecteur de journaux ne commence à traiter ses journaux, les informations du serveur API, telles que les étiquettes et les annotations, risquent de ne pas être disponibles. Il se peut qu'il n'y ait aucun moyen de distinguer les messages de journal d'un module et d'un projet portant un nom similaire ou de remonter à la source des journaux. Cette limitation signifie que la collecte et la normalisation des journaux sont considérées comme **best effort**.



IMPORTANT

Les moteurs d'exécution des conteneurs disponibles fournissent des informations minimales pour identifier la source des messages de journalisation et ne garantissent pas l'unicité des messages de journalisation ou la possibilité de remonter à la source de ces messages.

Pour plus d'informations, voir Configuration du collecteur de journalisation.

4.2.10. À propos de l'entrepôt de données

Par défaut, OpenShift Container Platform utilise Elasticsearch (ES) pour stocker les données de log. En option, vous pouvez utiliser l'API Log Forwarder pour transférer les journaux vers un magasin externe. Plusieurs types de magasins sont pris en charge, notamment fluentd, rsyslog, kafka et d'autres.

L'instance Elasticsearch du sous-système de journalisation est optimisée et testée pour un stockage à court terme, environ sept jours. Si vous souhaitez conserver vos journaux à plus long terme, il est recommandé de déplacer les données vers un système de stockage tiers.

Elasticsearch organise les données de log de Fluentd dans des datastores, ou *indices*, puis subdivise chaque index en plusieurs morceaux appelés *shards*, qu'il répartit sur un ensemble de nœuds Elasticsearch dans un cluster Elasticsearch. Vous pouvez configurer Elasticsearch pour qu'il fasse des copies des morceaux, appelées *replicas*, qu'il répartit également sur les nœuds Elasticsearch. La ressource personnalisée (CR) **ClusterLogging** vous permet de spécifier la manière dont les fichiers sont répliqués afin d'assurer la redondance des données et la résistance aux pannes. Vous pouvez également spécifier la durée de conservation des différents types de journaux à l'aide d'une politique de conservation dans la CR **ClusterLogging**.

NOTE

Le nombre d'unités primaires pour les modèles d'index est égal au nombre de nœuds de données Elasticsearch.

Red Hat OpenShift Logging Operator et son compagnon OpenShift Elasticsearch Operator garantissent que chaque nœud Elasticsearch est déployé à l'aide d'un déploiement unique qui inclut son propre volume de stockage. Vous pouvez utiliser une ressource personnalisée (CR) **ClusterLogging** pour augmenter le nombre de nœuds Elasticsearch, si nécessaire. Voir la documentation Elasticsearch pour les considérations liées à la configuration du stockage.

NOTE

Un environnement Elasticsearch hautement disponible nécessite au moins trois nœuds Elasticsearch, chacun sur un hôte différent.

Le contrôle d'accès basé sur les rôles (RBAC) appliqué aux indices Elasticsearch permet de contrôler l'accès des développeurs aux journaux. Les administrateurs peuvent accéder à tous les journaux et les développeurs ne peuvent accéder qu'aux journaux de leurs projets.

Pour plus d'informations, voir Configuration du magasin de journaux .

4.2.11. À propos de la visualisation de l'enregistrement

OpenShift Container Platform utilise Kibana pour afficher les données de logs collectées par Fluentd et indexées par Elasticsearch.

Kibana est une interface de console basée sur un navigateur pour interroger, découvrir et visualiser vos données Elasticsearch à l'aide d'histogrammes, de graphiques linéaires, de diagrammes circulaires et d'autres visualisations.

Pour plus d'informations, voir Configuration du visualiseur de journaux .

4.2.12. À propos de l'acheminement des événements

Event Router est un pod qui surveille les événements de OpenShift Container Platform afin qu'ils puissent être collectés par le sous-système de journalisation de Red Hat OpenShift. L'Event Router collecte les événements de tous les projets et les écrit sur **STDOUT**. Fluentd collecte ces événements et les transmet à l'instance Elasticsearch de OpenShift Container Platform. Elasticsearch indexe les événements dans l'index **infra**.

Vous devez déployer manuellement le routeur d'événements.

Pour plus d'informations, voir Collecte et stockage des événements Kubernetes .

4.2.13. À propos de la redirection des journaux

Par défaut, le sous-système de journalisation pour Red Hat OpenShift envoie les journaux au magasin de journaux Elasticsearch interne par défaut, défini dans la ressource personnalisée (CR) **ClusterLogging**. Si vous souhaitez transférer les journaux vers d'autres agrégateurs de journaux, vous pouvez utiliser les fonctionnalités de transfert de journaux pour envoyer les journaux à des points d'extrémité spécifiques à l'intérieur ou à l'extérieur de votre cluster.

Pour plus d'informations, voir Transférer les journaux vers des systèmes tiers .

4.3. À PROPOS DE VECTOR

Vector est un collecteur de journaux proposé comme alternative à Fluentd pour le sous-système de journalisation.

Les sorties suivantes sont prises en charge :

- **elasticsearch**. Une instance Elasticsearch externe. La sortie **elasticsearch** peut utiliser une connexion TLS.
- **kafka**. Un courtier Kafka. La sortie **kafka** peut utiliser une connexion non sécurisée ou TLS.
- **loki**. Loki, un système d'agrégation de logs horizontalement extensible, hautement disponible et multitenant.

4.3.1. Vecteur d'habilitation

Vector n'est pas activé par défaut. Suivez les étapes suivantes pour activer Vector sur votre cluster OpenShift Container Platform.

IMPORTANT

Vector ne prend pas en charge les clusters compatibles FIPS.

Conditions préalables

- OpenShift Container Platform : 4.12
- Sous-système de journalisation pour Red Hat OpenShift : 5.4
- FIPS désactivé

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc -n openshift-logging edit ClusterLogging instance
```

2. Ajouter une annotation **logging.openshift.io/preview-vector-collector: enabled** à la ressource personnalisée (CR) **ClusterLogging**.
3. Ajouter **vector** comme type de collection à la ressource personnalisée (CR) **ClusterLogging**.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
  annotations:
    logging.openshift.io/preview-vector-collector: enabled
spec:
  collection:
  logs:
    type: "vector"
    vector: {}
```

Ressources complémentaires

- Documentation sur les vecteurs

4.3.2. Caractéristiques du collectionneur

Tableau 4.1. Sources des journaux

Fonctionnalité	Fluentd	Vecteur
Journaux des conteneurs d'applications	✓	✓
Routage spécifique à l'application	✓	✓

Fonctionnalité	Fluentd	Vecteur
Routage spécifique à l'application par espace de noms	✓	✓
Registres des conteneurs Infra	✓	✓
Journal de bord de l'infra	✓	✓
Journaux d'audit de l'API Kube	✓	✓
Journaux d'audit de l'API OpenShift	✓	✓
Journaux d'audit de l'Open Virtual Network (OVN)	✓	✓

Tableau 4.2. Sorties

Fonctionnalité	Fluentd	Vecteur
Elasticsearch v5-v7	✓	✓
En avant toute	✓	
Syslog RFC3164	✓	
Syslog RFC5424	✓	
Kafka	✓	✓
Cloudwatch	✓	✓
Loki	✓	✓

Tableau 4.3. Autorisation et authentification

Fonctionnalité	Fluentd	Vecteur
Certificats Elasticsearch	✓	✓
Nom d'utilisateur / mot de passe Elasticsearch	✓	✓
Clés Cloudwatch	✓	✓
Cloudwatch STS	✓	

Fonctionnalité	Fluentd	Vecteur
Certificats Kafka	✓	✓
Nom d'utilisateur / mot de passe Kafka	✓	✓
Kafka SASL	✓	✓
Jeton du porteur de Loki	✓	✓

Tableau 4.4. Normalisations et transformations

Fonctionnalité	Fluentd	Vecteur
Modèle de données Viaq - app	✓	✓
Modèle de données Viaq - infra	✓	✓
Modèle de données Viaq - infra(journal)	✓	✓
Modèle de données Viaq - Audit Linux	✓	✓
Modèle de données Viaq - audit kube-apiserver	✓	✓
Modèle de données Viaq - Audit API OpenShift	✓	✓
Modèle de données Viaq - OVN	✓	✓
Normalisation des niveaux de journalisation	✓	✓
Analyse JSON	✓	✓
Indice structuré	✓	✓
Détection des erreurs multilignes	✓	
Indices multiconteneurs / fractionnés	✓	✓

Fonctionnalité	Fluentd	Vecteur
Aplatir les étiquettes	✓	✓
Étiquettes statiques de la NSI	✓	✓

Tableau 4.5. Accorder

Fonctionnalité	Fluentd	Vecteur
Limite de lecture de Fluentd	✓	
Tampon Fluentd	✓	
- taille limite du chunk	✓	
- taille totale	✓	
- débordementaction	✓	
- flushthreadcount	✓	
- mode flush	✓	
- intervalle de rinçage	✓	
- retrywait	✓	
- type de tentative	✓	
- retrymaxinterval	✓	
- délai de réessai	✓	

Tableau 4.6. Visibilité

Fonctionnalité	Fluentd	Vecteur
Metrics	✓	✓
Tableau de bord	✓	✓
Alertes	✓	

Tableau 4.7. Divers

Fonctionnalité	Fluentd	Vecteur
Prise en charge globale du proxy	✓	✓
support x86	✓	✓
Support ARM	✓	✓
Support IBM Power	✓	✓
Support IBM zSystems	✓	✓
Prise en charge de l'IPv6	✓	✓
Mise en mémoire tampon des événements du journal	✓	
Groupe déconnecté	✓	✓

CHAPITRE 5. INSTALLATION DU SOUS-SYSTÈME DE JOURNALISATION POUR RED HAT OPENSHIFT

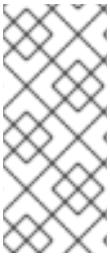
Vous pouvez installer le sous-système de journalisation pour Red Hat OpenShift en déployant les opérateurs OpenShift Elasticsearch et Red Hat OpenShift Logging. L'opérateur OpenShift Elasticsearch crée et gère le cluster Elasticsearch utilisé par OpenShift Logging. L'opérateur du sous-système de journalisation crée et gère les composants de la pile de journalisation.

Le processus de déploiement du sous-système de journalisation sur OpenShift Container Platform est le suivant :

- Examen des considérations relatives au stockage du sous-système de journalisation .
- Installation du sous-système de journalisation pour OpenShift Container Platform à l'aide de la console web ou du CLI.

5.1. INSTALLATION DU SOUS-SYSTÈME DE JOURNALISATION POUR RED HAT OPENSHIFT À L'AIDE DE LA CONSOLE WEB

Vous pouvez utiliser la console web d'OpenShift Container Platform pour installer les opérateurs OpenShift Elasticsearch et Red Hat OpenShift Logging.



NOTE

Si vous ne souhaitez pas utiliser le magasin de logs Elasticsearch par défaut, vous pouvez supprimer les composants internes Elasticsearch **logStore** et Kibana **visualization** de la ressource personnalisée (CR) **ClusterLogging**. La suppression de ces composants est facultative mais permet d'économiser des ressources. Pour plus d'informations, voir les ressources supplémentaires de cette section.

Conditions préalables

- Assurez-vous que vous disposez du stockage persistant nécessaire pour Elasticsearch. Notez que chaque nœud Elasticsearch nécessite son propre volume de stockage.

NOTE

Si vous utilisez un volume local pour le stockage persistant, n'utilisez pas de volume de blocs bruts, qui est décrit avec **volumeMode: block** dans l'objet **LocalVolume**. Elasticsearch ne peut pas utiliser de volumes de blocs bruts.

Elasticsearch est une application gourmande en mémoire. Par défaut, OpenShift Container Platform installe trois nœuds Elasticsearch avec des demandes de mémoire et des limites de 16 Go. Cet ensemble initial de trois nœuds OpenShift Container Platform peut ne pas avoir assez de mémoire pour faire fonctionner Elasticsearch dans votre cluster. Si vous rencontrez des problèmes de mémoire liés à Elasticsearch, ajoutez des nœuds Elasticsearch supplémentaires à votre cluster plutôt que d'augmenter la mémoire des nœuds existants.

Procédure

Pour installer OpenShift Elasticsearch Operator et Red Hat OpenShift Logging Operator à l'aide de la console web OpenShift Container Platform :

1. Installer l'opérateur OpenShift Elasticsearch :

a. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.

b. Choisissez **OpenShift Elasticsearch Operator** dans la liste des opérateurs disponibles et cliquez sur **Install**.

c. Assurez-vous que le site **All namespaces on the cluster**est sélectionné sous **Installation Mode**.

d. Assurez-vous que **openshift-operators-redhat** est sélectionné sous **Installed Namespace**. Vous devez spécifier l'espace de noms **openshift-operators-redhat**. L'espace de noms **openshift-operators** peut contenir des opérateurs communautaires, qui ne sont pas fiables et qui pourraient publier une métrique portant le même nom que la métrique, ce qui provoquerait des conflits.

e. Sélectionnez **Enable operator recommended cluster monitoring on this namespace** Cette option définit l'étiquette **openshift.io/cluster-monitoring: "true"** dans l'objet Namespace. Vous devez sélectionner cette option pour vous assurer que la surveillance des clusters récupère l'espace de noms **openshift-operators-redhat**.

f. Sélectionnez **stable-5.x** comme **Update Channel**.

g. Sélectionnez un site **Approval Strategy**.

- La stratégie **Automatic** permet à Operator Lifecycle Manager (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible.

- La stratégie **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.

h. Cliquez sur **Install**.

i. Vérifiez que l'OpenShift Elasticsearch Operator est installé en passant à la page **Operators → Installed Operators**.

j. Veillez à ce que **OpenShift Elasticsearch Operator** figure dans tous les projets dont l'adresse **Status** est **Succeeded**.

2. Installez l'opérateur de journalisation Red Hat OpenShift :

a. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.

b. Choisissez **Red Hat OpenShift Logging** dans la liste des opérateurs disponibles et cliquez sur **Install**.

c. Assurez-vous que le site **A specific namespace on the cluster**est sélectionné sous **Installation Mode**.

d. Assurez-vous que **Operator recommended namespace** est **openshift-logging** sous **Installed Namespace**.

e. Sélectionnez **Enable operator recommended cluster monitoring on this namespace** Cette option définit l'étiquette **openshift.io/cluster-monitoring: "true"** dans l'objet Namespace. Vous devez sélectionner cette option pour vous assurer que la surveillance des clusters récupère l'espace de noms **openshift-logging**.

f. Sélectionnez **stable-5.x** comme **Update Channel**.

g. Sélectionnez un site **Approval Strategy**.

- La stratégie **Automatic** permet à Operator Lifecycle Manager (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible.
- La stratégie **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.

h. Cliquez sur **Install**.

i. Vérifiez que Red Hat OpenShift Logging Operator est installé en passant à la page **Operators → Installed Operators**.

j. Assurez-vous que **Red Hat OpenShift Logging** est listé dans le projet **openshift-logging** avec un **Status** de **Succeeded**.
Si l'opérateur n'apparaît pas tel qu'il a été installé, il convient de poursuivre le dépannage :

- Passez à la page **Operators → Installed Operators** et vérifiez que la colonne **Status** ne comporte pas d'erreurs ou de défaillances.
- Passez à la page **Workloads → Pods** et vérifiez les journaux de tous les pods du projet **openshift-logging** qui signalent des problèmes.

3. Créer une instance OpenShift Logging :

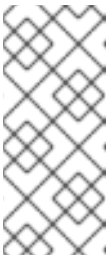
a. Passez à la page **Administration → Custom Resource Definitions**.

b. Sur la page **Custom Resource Definitions**, cliquez sur **ClusterLogging**.

c. Sur la page **Custom Resource Definition details**, sélectionnez **View Instances** dans le menu **Actions**.

d. Sur la page **ClusterLoggings**, cliquez sur **Create ClusterLogging**.
Il se peut que vous deviez rafraîchir la page pour charger les données.

e. Dans le champ YAML, remplacez le code par le suivant :



NOTE

Cette configuration par défaut d'OpenShift Logging devrait prendre en charge un large éventail d'environnements. Consultez les rubriques sur le réglage et la configuration des composants du sous-système de journalisation pour obtenir des informations sur les modifications que vous pouvez apporter à votre cluster OpenShift Logging.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" 1
  namespace: "openshift-logging"
spec:
  managementState: "Managed" 2
  logStore:
    type: "elasticsearch" 3
```

```
    retentionPolicy: 4
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3 5
      storage:
        storageClassName: "<storage_class_name>" 6
        size: 200G
      resources: 7
          limits:
            memory: "16Gi"
          requests:
            memory: "16Gi"
      proxy: 8
        resources:
          limits:
            memory: 256Mi
          requests:
            memory: 256Mi
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana" 9
    kibana:
      replicas: 1
  collection:
    logs:
      type: "fluentd" 10
      fluentd: {}
```

1. Le nom doit être **instance**.
2. L'état de gestion d'OpenShift Logging. Dans certains cas, si vous modifiez les paramètres par défaut d'OpenShift Logging, vous devez définir ce paramètre à **Unmanaged**. Cependant, un déploiement non géré ne reçoit pas de mises à jour jusqu'à ce qu'OpenShift Logging soit replacé dans un état géré.
3. Paramètres de configuration d'Elasticsearch. En utilisant le CR, vous pouvez configurer la politique de réplication et le stockage persistant.
4. Spécifiez la durée pendant laquelle Elasticsearch doit conserver chaque source de journal. Saisissez un nombre entier et une désignation de temps : semaines(w), heures(h/H), minutes(m) et secondes(s). Par exemple, **7d** pour sept jours. Les journaux antérieurs à **maxAge** sont supprimés. Vous devez spécifier une politique de rétention pour chaque source de logs, sinon les index Elasticsearch ne seront pas créés pour cette source.
5. Spécifiez le nombre de nœuds Elasticsearch. Voir la note qui suit cette liste.
6. Saisissez le nom d'une classe de stockage existante pour le stockage Elasticsearch. Pour de meilleures performances, spécifiez une classe de stockage qui alloue un stockage en bloc. Si vous ne spécifiez pas de classe de stockage, OpenShift Logging

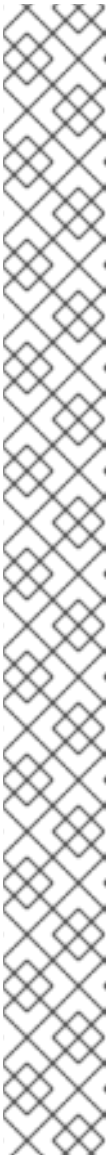
utilise un stockage éphémère.

7 Spécifiez les demandes de CPU et de mémoire pour Elasticsearch si nécessaire. Si vous laissez ces valeurs vides, OpenShift Elasticsearch Operator définit des valeurs par défaut qui devraient être suffisantes pour la plupart des déploiements. Les valeurs par défaut sont **16Gi** pour la demande de mémoire et **1** pour la demande de CPU.

8 Spécifiez les demandes de CPU et de mémoire pour le proxy Elasticsearch si nécessaire. Si vous laissez ces valeurs vides, OpenShift Elasticsearch Operator définit des valeurs par défaut qui devraient être suffisantes pour la plupart des déploiements. Les valeurs par défaut sont **256Mi** pour la demande de mémoire et **100m** pour la demande de CPU.

9 Paramètres de configuration de Kibana. En utilisant le CR, vous pouvez mettre à l'échelle Kibana pour la redondance et configurer le CPU et la mémoire pour vos nœuds Kibana. Pour plus d'informations, voir **Configuring the log visualizer**.

10 Paramètres pour la configuration de Fluentd. En utilisant le CR, vous pouvez configurer les limites de CPU et de mémoire de Fluentd. Pour plus d'informations, voir **Configuring Fluentd**.



NOTE

Le nombre maximum de nœuds de plan de contrôle Elasticsearch est de trois. Si vous spécifiez une adresse **nodeCount** supérieure à **3**, OpenShift Container Platform crée trois nœuds Elasticsearch qui sont des nœuds éligibles au rôle de maître, avec les rôles de maître, de client et de données. Les nœuds Elasticsearch supplémentaires sont créés en tant que nœuds de données uniquement, avec les rôles de client et de données. Les nœuds du plan de contrôle effectuent des actions à l'échelle du cluster, telles que la création ou la suppression d'un index, l'allocation de shards et le suivi des nœuds. Les nœuds de données détiennent les nuages et effectuent des opérations liées aux données telles que CRUD, la recherche et les agrégations. Les opérations liées aux données sont gourmandes en E/S, en mémoire et en CPU. Il est important de surveiller ces ressources et d'ajouter des nœuds de données supplémentaires si les nœuds actuels sont surchargés.

Par exemple, si **nodeCount=4**, les nœuds suivants sont créés :

```
$ oc get deployment
```

Exemple de sortie

```
cluster-logging-operator       1/1     1          1          18h
elasticsearch-cd-x6kdekli-1    0/1     1          0          6m54s
elasticsearch-cdm-x6kdekli-1   1/1     1          1          18h
elasticsearch-cdm-x6kdekli-2   0/1     1          0          6m49s
elasticsearch-cdm-x6kdekli-3   0/1     1          0          6m44s
```

Le nombre d'unités primaires pour les modèles d'index est égal au nombre de nœuds de données Elasticsearch.

f. Cliquez sur **Create**. Cela crée les composants du sous-système de journalisation, la ressource personnalisée **Elasticsearch** et ses composants, ainsi que l'interface Kibana.

4. Vérifier l'installation :

a. Passez à la page **Workloads → Pods**.

b. Sélectionnez le projet **openshift-logging**.
Vous devriez voir plusieurs pods pour OpenShift Logging, Elasticsearch, Fluentd, et Kibana similaires à la liste suivante :

- cluster-logging-operator-cb795f8dc-xkckc
- collecteur-pb2f8
- elasticsearch-cdm-b3nqzchd-1-5c6797-67kfz
- elasticsearch-cdm-b3nqzchd-2-6657f4-wtprv
- elasticsearch-cdm-b3nqzchd-3-588c65-clg7g
- fluentd-2c7dg
- fluentd-9z7kk
- fluentd-br7r2
- fluentd-fn2sb
- fluentd-zqgqx
- kibana-7fb4fd4cc9-bvt4p

Ressources complémentaires

- Installer des opérateurs à partir de l'OperatorHub
- Suppression des composants inutilisés si vous n'utilisez pas le magasin de logs Elasticsearch par défaut

5.2. POST-INSTALLATION TASKS

Si vous prévoyez d'utiliser Kibana, vous devez créer manuellement vos modèles d'index et vos visualisations Kibana pour explorer et visualiser les données dans Kibana.

Si votre module d'extension réseau assure l'isolation du réseau, autorisez le trafic réseau entre les projets qui contiennent le sous-système de journalisation Operators.

5.3. INSTALLATION DU SOUS-SYSTÈME DE JOURNALISATION POUR RED HAT OPENSHIFT À L'AIDE DU CLI

Vous pouvez utiliser le CLI de OpenShift Container Platform pour installer les opérateurs OpenShift Elasticsearch et Red Hat OpenShift Logging.

Conditions préalables

- Assurez-vous que vous disposez du stockage persistant nécessaire pour Elasticsearch. Notez que chaque nœud Elasticsearch nécessite son propre volume de stockage.

NOTE

Si vous utilisez un volume local pour le stockage persistant, n'utilisez pas de volume de blocs bruts, qui est décrit avec **volumeMode: block** dans l'objet **LocalVolume**. Elasticsearch ne peut pas utiliser de volumes de blocs bruts.

Elasticsearch est une application gourmande en mémoire. Par défaut, OpenShift Container Platform installe trois nœuds Elasticsearch avec des demandes de mémoire et des limites de 16 Go. Cet ensemble initial de trois nœuds OpenShift Container Platform peut ne pas avoir assez de mémoire pour faire fonctionner Elasticsearch dans votre cluster. Si vous rencontrez des problèmes de mémoire liés à Elasticsearch, ajoutez des nœuds Elasticsearch supplémentaires à votre cluster plutôt que d'augmenter la mémoire des nœuds existants.

Procédure

Pour installer OpenShift Elasticsearch Operator et Red Hat OpenShift Logging Operator à l'aide de la CLI :

1. Créer un espace de noms pour l'OpenShift Elasticsearch Operator.

   a. Créer un fichier YAML d'objet d'espace de noms (par exemple, **eo-namespace.yaml**) pour l'opérateur OpenShift Elasticsearch :

```
apiVersion: v1
kind: Namespace
metadata:
  name: openshift-operators-redhat 1
  annotations:
    openshift.io/node-selector: ""
  labels:
    openshift.io/cluster-monitoring: "true" 2
```

   1 Vous devez spécifier l'espace de noms **openshift-operators-redhat**. Pour éviter d'éventuels conflits avec les métriques, vous devez configurer la pile Prometheus Cluster Monitoring pour qu'elle récupère les métriques de l'espace de noms **openshift-operators-redhat** et non de l'espace de noms **openshift-operators**. L'espace de noms **openshift-operators** peut contenir des opérateurs communautaires, qui ne sont pas fiables et qui pourraient publier une mesure portant le même nom que la mesure, ce qui entraînerait des conflits.

   2 Chaîne. Vous devez spécifier cette étiquette comme indiqué pour vous assurer que la surveillance des clusters balaie l'espace de noms **openshift-operators-redhat**.

   b. Créer l'espace de noms :

```
oc create -f <nom-de-fichier>.yaml
```

   Par exemple :

```
$ oc create -f eo-namespace.yaml
```

2. Créez un espace de noms pour Red Hat OpenShift Logging Operator :

   a. Créez un fichier YAML d'objet d'espace de noms (par exemple, **olo-namespace.yaml**) pour l'opérateur de journalisation de Red Hat OpenShift :

```
apiVersion: v1
kind: Namespace
metadata:
  name: openshift-logging
  annotations:
    openshift.io/node-selector: ""
  labels:
    openshift.io/cluster-monitoring: "true"
```

   b. Créer l'espace de noms :

```
oc create -f <nom-de-fichier>.yaml
```

   Par exemple :

```
$ oc create -f olo-namespace.yaml
```

3. Installez l'OpenShift Elasticsearch Operator en créant les objets suivants :

   a. Créer un fichier YAML d'objet Operator Group (par exemple, **eo-og.yaml**) pour l'opérateur OpenShift Elasticsearch :

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: openshift-operators-redhat
  namespace: openshift-operators-redhat (1)
spec: {}
```

   (1) Vous devez spécifier l'espace de noms **openshift-operators-redhat**.

   b. Créer un objet Groupe d'opérateurs :

```
oc create -f <nom-de-fichier>.yaml
```

   Par exemple :

```
$ oc create -f eo-og.yaml
```

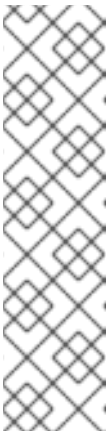
   c. Créer un fichier YAML d'objet d'abonnement (par exemple, **eo-sub.yaml**) pour abonner un espace de noms à l'opérateur Elasticsearch d'OpenShift.

   Exemple d'abonnement

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: "elasticsearch-operator"
```

```
  namespace: "openshift-operators-redhat" (1)
spec:
  channel: "stable-5.5" (2)
  installPlanApproval: "Automatic" (3)
  source: "redhat-operators" (4)
  sourceNamespace: "openshift-marketplace"
  name: "elasticsearch-operator"
```

1. Vous devez spécifier l'espace de noms **openshift-operators-redhat**.
2. Spécifiez **stable** ou **stable-5.<x>** comme canal. Voir la note suivante.
3. **Automatic** permet au gestionnaire du cycle de vie de l'opérateur (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible. **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.
4. Spécifiez **redhat-operators**. Si votre cluster OpenShift Container Platform est installé sur un réseau restreint, également connu sous le nom de cluster déconnecté, indiquez le nom de l'objet CatalogSource créé lors de la configuration de l'Operator Lifecycle Manager (OLM).



NOTE

En spécifiant **stable**, vous installez la version actuelle de la dernière version stable. L'utilisation de **stable** avec **installPlanApproval: "Automatic"**, mettra automatiquement à niveau vos opérateurs vers la dernière version stable majeure et mineure.

En spécifiant **stable-5.<x>**, vous installez la version mineure actuelle d'une version majeure spécifique. L'utilisation de **stable-5.<x>** avec **installPlanApproval: "Automatic"**, mettra automatiquement à niveau vos opérateurs vers la dernière version mineure stable de la version majeure que vous spécifiez avec **x**.

d. Créer l'objet Abonnement :

```
oc create -f <nom-de-fichier>.yaml
```

Par exemple :

```
$ oc create -f eo-sub.yaml
```

L'OpenShift Elasticsearch Operator est installé dans l'espace de noms **openshift-operators-redhat** et copié dans chaque projet du cluster.

e. Vérifier l'installation de l'opérateur :

```
$ oc get csv --all-namespaces
```

Exemple de sortie

```
NAMESPACE                                 NAME                                         DISPLAY
VERSION            REPLACES   PHASE
default                                   elasticsearch-operator.5.1.0-202007012112.p0
OpenShift Elasticsearch Operator   5.5.0-202007012112.p0            Succeeded
kube-node-lease                           elasticsearch-operator.5.5.0-202007012112.p0
OpenShift Elasticsearch Operator   5.5.0-202007012112.p0            Succeeded
kube-public                               elasticsearch-operator.5.5.0-202007012112.p0
OpenShift Elasticsearch Operator   5.5.0-202007012112.p0            Succeeded
kube-system                               elasticsearch-operator.5.5.0-202007012112.p0
OpenShift Elasticsearch Operator   5.5.0-202007012112.p0            Succeeded
openshift-apiserver-operator              elasticsearch-operator.5.5.0-
202007012112.p0    OpenShift Elasticsearch Operator   5.5.0-202007012112.p0
Succeeded
openshift-apiserver                       elasticsearch-operator.5.5.0-202007012112.p0
OpenShift Elasticsearch Operator   5.5.0-202007012112.p0            Succeeded
openshift-authentication-operator         elasticsearch-operator.5.5.0-
202007012112.p0    OpenShift Elasticsearch Operator   5.5.0-202007012112.p0
Succeeded
openshift-authentication                  elasticsearch-operator.5.5.0-
202007012112.p0    OpenShift Elasticsearch Operator   5.5.0-202007012112.p0
Succeeded
...
```

Il devrait y avoir un OpenShift Elasticsearch Operator dans chaque espace de noms. Le numéro de version peut être différent de celui indiqué.

4. Installez l'opérateur de journalisation Red Hat OpenShift en créant les objets suivants :

   a. Créez un fichier YAML d'objet de groupe d'opérateurs (par exemple, **olo-og.yaml**) pour l'opérateur de journalisation Red Hat OpenShift :

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cluster-logging
  namespace: openshift-logging (1)
spec:
  targetNamespaces:
  - openshift-logging (2)
```

(1) (2) Vous devez spécifier l'espace de noms **openshift-logging**.

   b. Créer l'objet OperatorGroup :

```
oc create -f <nom-de-fichier>.yaml
```

Par exemple :

```
$ oc create -f olo-og.yaml
```

   c. Créez un fichier YAML d'objet d'abonnement (par exemple, **olo-sub.yaml**) pour abonner un espace de noms à l'opérateur de journalisation de Red Hat OpenShift.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: cluster-logging
  namespace: openshift-logging 1
spec:
  channel: "stable" 2
  name: cluster-logging
  source: redhat-operators 3
  sourceNamespace: openshift-marketplace
```

1. Vous devez spécifier l'espace de noms **openshift-logging**.
2. Spécifiez **stable** ou **stable-5.<x>** comme canal.
3. Spécifiez **redhat-operators**. Si votre cluster OpenShift Container Platform est installé sur un réseau restreint, également connu sous le nom de cluster déconnecté, indiquez le nom de l'objet CatalogSource que vous avez créé lors de la configuration de l'Operator Lifecycle Manager (OLM).

```
oc create -f <nom-de-fichier>.yaml
```

Par exemple :

```
$ oc create -f olo-sub.yaml
```

Red Hat OpenShift Logging Operator est installé dans l'espace de noms **openshift-logging**.

d. Vérifier l'installation de l'opérateur.
Il devrait y avoir un Red Hat OpenShift Logging Operator dans l'espace de noms **openshift-logging**. Le numéro de version peut être différent de celui indiqué.

```
$ oc get csv -n openshift-logging
```

Exemple de sortie

```
NAMESPACE                                       NAME                                         DISPLAY
VERSION            REPLACES   PHASE
...
openshift-logging                               clusterlogging.5.1.0-202007012112.p0
OpenShift Logging        5.1.0-202007012112.p0            Succeeded
...
```

5. Créer une instance OpenShift Logging :

a. Créez un fichier YAML d'objet d'instance (par exemple, **olo-instance.yaml**) pour l'opérateur de journalisation de Red Hat OpenShift :

NOTE

Cette configuration par défaut d'OpenShift Logging devrait prendre en charge un large éventail d'environnements. Consultez les rubriques sur le réglage et la configuration des composants du sous-système de journalisation pour obtenir des informations sur les modifications que vous pouvez apporter à votre cluster OpenShift Logging.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" 1
  namespace: "openshift-logging"
spec:
  managementState: "Managed" 2
  logStore:
    type: "elasticsearch" 3
    retentionPolicy: 4
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3 5
      storage:
        storageClassName: "<storage-class-name>" 6
        size: 200G
      resources: 7
        limits:
          memory: "16Gi"
        requests:
          memory: "16Gi"
      proxy: 8
        resources:
          limits:
            memory: 256Mi
          requests:
            memory: 256Mi
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana" 9
    kibana:
      replicas: 1
  collection:
    logs:
      type: "fluentd" 10
      fluentd: {}
```

1. Le nom doit être **instance**.
2. L'état de gestion d'OpenShift Logging. Dans certains cas, si vous modifiez les paramètres par défaut d'OpenShift Logging, vous devez définir ce paramètre à **Unmanaged**. Cependant, un déploiement non géré ne reçoit pas de mise à jour

Unmanaged. Cependant, un déploiement non géré ne reçoit pas de mises à jour jusqu'à ce qu'OpenShift Logging soit replacé dans un état géré. Le fait de remettre un déploiement dans un état géré peut annuler toutes les modifications que vous avez apportées.

3 Paramètres de configuration d'Elasticsearch. En utilisant la ressource personnalisée (CR), vous pouvez configurer la politique de réplication et le stockage persistant.

4 Spécifiez la durée pendant laquelle Elasticsearch doit conserver chaque source de journal. Saisissez un nombre entier et une désignation de temps : semaines(w), heures(h/H), minutes(m) et secondes(s). Par exemple, **7d** pour sept jours. Les journaux antérieurs à **maxAge** sont supprimés. Vous devez spécifier une politique de rétention pour chaque source de logs, sinon les index Elasticsearch ne seront pas créés pour cette source.

5 Spécifiez le nombre de nœuds Elasticsearch. Voir la note qui suit cette liste.

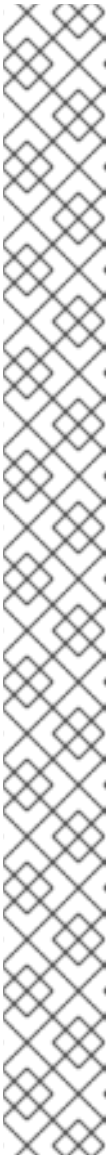
6 Saisissez le nom d'une classe de stockage existante pour le stockage Elasticsearch. Pour de meilleures performances, spécifiez une classe de stockage qui alloue des blocs de stockage. Si vous ne spécifiez pas de classe de stockage, OpenShift Container Platform déploie OpenShift Logging avec un stockage éphémère uniquement.

7 Spécifiez les demandes de CPU et de mémoire pour Elasticsearch si nécessaire. Si vous laissez ces valeurs vides, OpenShift Elasticsearch Operator définit des valeurs par défaut qui sont suffisantes pour la plupart des déploiements. Les valeurs par défaut sont **16Gi** pour la demande de mémoire et **1** pour la demande de CPU.

8 Spécifiez les demandes de CPU et de mémoire pour le proxy Elasticsearch si nécessaire. Si vous laissez ces valeurs vides, OpenShift Elasticsearch Operator définit des valeurs par défaut qui devraient être suffisantes pour la plupart des déploiements. Les valeurs par défaut sont **256Mi** pour la demande de mémoire et **100m** pour la demande de CPU.

9 Paramètres de configuration de Kibana. En utilisant le CR, vous pouvez mettre à l'échelle Kibana pour la redondance et configurer le CPU et la mémoire pour vos pods Kibana. Pour plus d'informations, voir **Configuring the log visualizer**.

10 Paramètres pour la configuration de Fluentd. En utilisant le CR, vous pouvez configurer les limites de CPU et de mémoire de Fluentd. Pour plus d'informations, voir **Configuring Fluentd**.



NOTE

Le nombre maximum de nœuds de plan de contrôle Elasticsearch est de trois. Si vous spécifiez une adresse **nodeCount** supérieure à **3**, OpenShift Container Platform crée trois nœuds Elasticsearch qui sont des nœuds éligibles au rôle de maître, avec les rôles de maître, de client et de données. Les nœuds Elasticsearch supplémentaires sont créés en tant que nœuds de données uniquement, avec les rôles de client et de données. Les nœuds du plan de contrôle effectuent des actions à l'échelle du cluster, telles que la création ou la suppression d'un index, l'allocation de shards et le suivi des nœuds. Les nœuds de données détiennent les nuages et effectuent des opérations liées aux données telles que CRUD, la recherche et les agrégations. Les opérations liées aux données sont gourmandes en E/S, en mémoire et en CPU. Il est important de surveiller ces ressources et d'ajouter des nœuds de données supplémentaires si les nœuds actuels sont surchargés.

Par exemple, si **nodeCount=4**, les nœuds suivants sont créés :

```
$ oc get deployment
```

Exemple de sortie

```
cluster-logging-operator       1/1     1          1          18h
elasticsearch-cd-x6kdekli-1    1/1     1          0          6m54s
elasticsearch-cdm-x6kdekli-1   1/1     1          1          18h
elasticsearch-cdm-x6kdekli-2   1/1     1          0          6m49s
elasticsearch-cdm-x6kdekli-3   1/1     1          0          6m44s
```

Le nombre d'unités primaires pour les modèles d'index est égal au nombre de nœuds de données Elasticsearch.

b. Create the instance:

```
oc create -f <nom-de-fichier>.yaml
```

Par exemple :

```
$ oc create -f olo-instance.yaml
```

Cela crée les composants du sous-système de journalisation, la ressource personnalisée **Elasticsearch** et ses composants, ainsi que l'interface Kibana.

6. Vérifier l'installation en listant les pods dans le projet **openshift-logging**.
Vous devriez voir plusieurs pods pour les composants du sous-système de journalisation, comme dans la liste suivante :

```
$ oc get pods -n openshift-logging
```

Exemple de sortie

```
NAME                                        READY   STATUS    RESTARTS   AGE
cluster-logging-operator-66f77ffccb-ppzbg   1/1     Running   0          7m
```

```
elasticsearch-cdm-ftuhduuw-1-ffc4b9566-q6bhp   2/2   Running   0   2m40s
elasticsearch-cdm-ftuhduuw-2-7b4994dbfc-rd2gc   2/2   Running   0   2m36s
elasticsearch-cdm-ftuhduuw-3-84b5ff7ff8-gqnm2   2/2   Running   0   2m4s
collector-587vb                                 1/1   Running   0   2m26s
collector-7mpb9                                 1/1   Running   0   2m30s
collector-flm6j                                 1/1   Running   0   2m33s
collector-gn4rn                                 1/1   Running   0   2m26s
collector-nlgb6                                 1/1   Running   0   2m30s
collector-snpkt                                 1/1   Running   0   2m28s
kibana-d6d5668c5-rppqm                          2/2   Running   0   2m39s
```

5.4. POST-INSTALLATION TASKS

Si vous prévoyez d'utiliser Kibana, vous devez créer manuellement vos modèles d'index et vos visualisations Kibana pour explorer et visualiser les données dans Kibana.

Si votre module d'extension réseau assure l'isolation du réseau, autorisez le trafic réseau entre les projets qui contiennent le sous-système de journalisation Operators.

5.4.1. Définir les modèles d'index Kibana

Un modèle d'index définit les index Elasticsearch que vous souhaitez visualiser. Pour explorer et visualiser des données dans Kibana, vous devez créer un modèle d'index.

Conditions préalables

- Un utilisateur doit avoir le rôle **cluster-admin**, le rôle **cluster-reader** ou les deux rôles pour voir les index **infra** et **audit** dans Kibana. L'utilisateur par défaut **kubeadmin** dispose des autorisations nécessaires pour afficher ces index.
  Si vous pouvez voir les pods et les journaux dans les projets **default**, **kube-** et **openshift-**, vous devriez pouvoir accéder à ces index. Vous pouvez utiliser la commande suivante pour vérifier si l'utilisateur actuel dispose des autorisations appropriées :

  ```
  $ oc auth can-i get pods/log -n <projet>
  ```

  Exemple de sortie

  ```
  yes
  ```

  NOTE

  Les journaux d'audit ne sont pas stockés dans l'instance interne d'OpenShift Container Platform Elasticsearch par défaut. Pour afficher les journaux d'audit dans Kibana, vous devez utiliser l'API Log Forwarding pour configurer un pipeline qui utilise la sortie **default** pour les journaux d'audit.

- Les documents Elasticsearch doivent être indexés avant de pouvoir créer des modèles d'index. Cette opération est effectuée automatiquement, mais elle peut prendre quelques minutes dans un cluster nouveau ou mis à jour.

Procédure

Pour définir des modèles d'index et créer des visualisations dans Kibana :

1. Dans la console OpenShift Container Platform, cliquez sur le lanceur d'applications  et sélectionnez **Logging**.

2. Créez vos modèles d'index Kibana en cliquant sur **Management → Index Patterns → Create index pattern**:

   - Chaque utilisateur doit créer manuellement des modèles d'index lors de sa première connexion à Kibana pour voir les journaux de ses projets. Les utilisateurs doivent créer un modèle d'index nommé **app** et utiliser le champ **@timestamp** time pour afficher les journaux de leurs conteneurs.

   - Chaque utilisateur administrateur doit créer des modèles d'index lors de sa première connexion à Kibana pour les index **app**, **infra**, et **audit** en utilisant le champ **@timestamp** time.

3. Créer des visualisations Kibana à partir des nouveaux modèles d'index.

5.4.2. Permettre le trafic entre les projets lorsque l'isolation du réseau est activée

Le plugin réseau de votre cluster peut imposer l'isolation du réseau. Si c'est le cas, vous devez autoriser le trafic réseau entre les projets qui contiennent les opérateurs déployés par OpenShift Logging.

L'isolation du réseau bloque le trafic réseau entre les pods ou les services qui se trouvent dans des projets différents. Le sous-système de journalisation installe *OpenShift Elasticsearch Operator* dans le projet **openshift-operators-redhat** et *Red Hat OpenShift Logging Operator* dans le projet **openshift-logging**. Vous devez donc autoriser le trafic entre ces deux projets.

OpenShift Container Platform propose deux choix pour le plugin réseau, OpenShift SDN et OVN-Kubernetes. Ces deux fournisseurs mettent en œuvre diverses politiques d'isolation du réseau.

OpenShift SDN dispose de trois modes :

politique de réseau

Il s'agit du mode par défaut. Si aucune politique n'est définie, il autorise tout le trafic. Toutefois, si un utilisateur définit une politique, il commence généralement par refuser tout le trafic, puis ajoute des exceptions. Ce processus risque d'interrompre les applications exécutées dans différents projets. Par conséquent, il convient de configurer explicitement la stratégie afin d'autoriser le trafic à sortir d'un projet lié à la journalisation vers l'autre.

multitenant

Ce mode permet d'isoler le réseau. Vous devez joindre les deux projets liés à la journalisation pour permettre le trafic entre eux.

sous-réseau

Ce mode autorise tout le trafic. Il n'applique pas l'isolation du réseau. Aucune action n'est nécessaire.

OVN-Kubernetes utilise toujours un **network policy**. Par conséquent, comme avec OpenShift SDN, vous devez configurer la politique pour permettre au trafic de sortir d'un projet lié à la journalisation vers l'autre.

Procédure

- Si vous utilisez OpenShift SDN en mode **multitenant**, joignez les deux projets. Par exemple :

```
$ oc adm pod-network join-projects --to=openshift-operators-redhat openshift-logging
```

- Sinon, pour OpenShift SDN en mode **network policy** et OVN-Kubernetes, effectuez les actions suivantes :

  a. Définir une étiquette sur l'espace de noms **openshift-operators-redhat**. Par exemple :

  ```
  $ oc label namespace openshift-operators-redhat project=openshift-operators-redhat
  ```

  b. Créez un objet de politique de réseau dans l'espace de noms **openshift-logging** qui autorise l'entrée des projets **openshift-operators-redhat**, **openshift-monitoring** et **openshift-ingress** dans le projet openshift-logging. Par exemple :

  ```
  apiVersion: networking.k8s.io/v1
  kind: NetworkPolicy
  metadata:
    name: allow-from-openshift-monitoring-ingress-operators-redhat
  spec:
    ingress:
    - from:
      - podSelector: {}
    - from:
      - namespaceSelector:
          matchLabels:
            project: "openshift-operators-redhat"
    - from:
      - namespaceSelector:
          matchLabels:
            name: "openshift-monitoring"
    - from:
      - namespaceSelector:
          matchLabels:
            network.openshift.io/policy-group: ingress
    podSelector: {}
    policyTypes:
    - Ingress
  ```

Ressources complémentaires

- A propos de la politique de réseau
- À propos du fournisseur de réseau CNI par défaut d'OpenShift SDN
- À propos du fournisseur de réseau par défaut OVN-Kubernetes Container Network Interface (CNI)

CHAPITRE 6. CONFIGURATION DU DÉPLOIEMENT DE LA JOURNALISATION

6.1. À PROPOS DE LA RESSOURCE PERSONNALISÉE CLUSTER LOGGING

Pour configurer le sous-système de journalisation pour Red Hat OpenShift, vous devez personnaliser la ressource personnalisée (CR) **ClusterLogging**.

6.1.1. À propos de la ressource personnalisée ClusterLogging

Pour modifier l'environnement de votre sous-système de journalisation, créez et modifiez la ressource personnalisée (CR) **ClusterLogging**.

Les instructions relatives à la création ou à la modification d'une CR sont fournies dans cette documentation, le cas échéant.

L'exemple suivant montre une ressource personnalisée typique pour le sous-système de journalisation.

Exemple ClusterLogging ressource personnalisée (CR)

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance" 1
  namespace: "openshift-logging" 2
spec:
  managementState: "Managed" 3
  logStore:
    type: "elasticsearch" 4
    retentionPolicy:
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3
      resources:
        limits:
          memory: 16Gi
        requests:
          cpu: 500m
          memory: 16Gi
      storage:
        storageClassName: "gp2"
        size: "200G"
      redundancyPolicy: "SingleRedundancy"
  visualization: 5
    type: "kibana"
    kibana:
      resources:
```

```
      limits:
        memory: 736Mi
      requests:
        cpu: 100m
        memory: 736Mi
    replicas: 1
collection: 6
  logs:
    type: "fluentd"
    fluentd:
      resources:
        limits:
          memory: 736Mi
        requests:
          cpu: 100m
          memory: 736Mi
```

1. Le nom du CR doit être **instance**.
2. Le CR doit être installé dans l'espace de noms **openshift-logging**.
3. L'état de gestion de l'opérateur de journalisation de Red Hat OpenShift. Lorsqu'il est défini sur **unmanaged**, l'opérateur est dans un état non pris en charge et ne recevra pas de mises à jour.
4. Paramètres du magasin de journaux, y compris la politique de conservation, le nombre de nœuds, les demandes et limites de ressources et la classe de stockage.
5. Paramètres du visualiseur, y compris les demandes de ressources et les limites, ainsi que le nombre de répliques de pods.
6. Paramètres du collecteur de journaux, y compris les demandes de ressources et les limites.

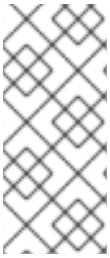
6.2. CONFIGURATION DU COLLECTEUR DE JOURNALISATION

Le sous-système de journalisation pour Red Hat OpenShift collecte les opérations et les journaux d'application de votre cluster et enrichit les données avec les métadonnées des pods et des projets Kubernetes.

Vous pouvez configurer les limites de CPU et de mémoire pour le collecteur de logs et déplacer les pods du collecteur de logs vers des nœuds spécifiques. Toutes les modifications prises en charge pour le collecteur de journaux peuvent être effectuées via la strophe **spec.collection.log.fluentd** dans la ressource personnalisée (CR) **ClusterLogging**.

6.2.1. À propos des configurations non prises en charge

La manière supportée de configurer le sous-système de journalisation pour Red Hat OpenShift est de le configurer en utilisant les options décrites dans cette documentation. N'utilisez pas d'autres configurations, car elles ne sont pas prises en charge. Les paradigmes de configuration peuvent changer à travers les versions d'OpenShift Container Platform, et de tels cas ne peuvent être gérés avec élégance que si toutes les possibilités de configuration sont contrôlées. Si vous utilisez des configurations autres que celles décrites dans cette documentation, vos changements disparaîtront car l'OpenShift Elasticsearch Operator et le Red Hat OpenShift Logging Operator réconcilient toutes les différences. Les opérateurs inversent tout à l'état défini par défaut et par conception.



NOTE

Si vous *must* effectuez des configurations non décrites dans la documentation d'OpenShift Container Platform, vous *must* configurez votre Red Hat OpenShift Logging Operator ou OpenShift Elasticsearch Operator sur **Unmanaged**. Un environnement OpenShift Logging non géré est *not supported* et ne reçoit pas de mises à jour jusqu'à ce que vous remettiez OpenShift Logging sur **Managed**.

6.2.2. Visualisation des pods du collecteur de journalisation

Vous pouvez visualiser les pods du collecteur de logs Fluentd et les nœuds correspondants sur lesquels ils s'exécutent. Les pods Fluentd logging collector s'exécutent uniquement dans le projet **openshift-logging**.

Procédure

- Exécutez la commande suivante dans le projet **openshift-logging** pour afficher les pods du collecteur de logs Fluentd et leurs détails :

```
$ oc get pods --selector component=collector -o wide -n openshift-logging
```

Exemple de sortie

```
NAME          READY  STATUS    RESTARTS   AGE     IP            NODE                NOMINATED
NODE   READINESS GATES
fluentd-8d69v  1/1    Running   0          134m    10.130.2.30   master1.example.com   <none>
<none>
fluentd-bd225  1/1    Running   0          134m    10.131.1.11   master2.example.com   <none>
<none>
fluentd-cvrzs  1/1    Running   0          134m    10.130.0.21   master3.example.com   <none>
<none>
fluentd-gpqg2  1/1    Running   0          134m    10.128.2.27   worker1.example.com   <none>
<none>
fluentd-l9j7j  1/1    Running   0          134m    10.129.2.31   worker2.example.com   <none>
<none>
```

6.2.3. Configurer les limites du processeur et de la mémoire du collecteur de journaux

Le collecteur de journaux permet d'ajuster les limites de l'unité centrale et de la mémoire.

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc -n openshift-logging edit ClusterLogging instance
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: openshift-logging
```

```
...

spec:
  collection:
    logs:
      fluentd:
        resources:
          limits: 1
            memory: 736Mi
          requests:
            cpu: 100m
            memory: 736Mi
```

1 Spécifiez les limites et les demandes de CPU et de mémoire si nécessaire. Les valeurs indiquées sont les valeurs par défaut.

6.2.4. Configuration avancée du redirecteur de logs

Le sous-système de journalisation pour Red Hat OpenShift inclut plusieurs paramètres Fluentd que vous pouvez utiliser pour régler la performance du transitaire de journaux Fluentd. Avec ces paramètres, vous pouvez changer les comportements suivants de Fluentd :

- Taille des blocs et des tampons de blocs
- Comportement d'évacuation des morceaux
- Comportement des tentatives de réacheminement de morceaux

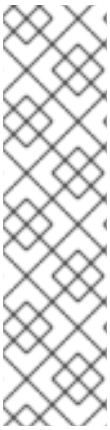
Fluentd collecte les données de log dans un blob unique appelé *chunk*. Quand Fluentd crée un chunk, le chunk est considéré comme étant dans le *stage*, où le chunk est rempli de données. Lorsque le bloc est plein, Fluentd le déplace vers *queue*, où les blocs sont conservés avant d'être vidés, ou écrits vers leur destination. Fluentd peut échouer à vider un chunk pour un certain nombre de raisons, comme des problèmes de réseau ou de capacité à la destination. Si un chunk ne peut pas être flushé, Fluentd retente le flushing comme configuré.

Par défaut dans OpenShift Container Platform, Fluentd utilise la méthode *exponential backoff* pour réessayer le flushing, où Fluentd double le temps qu'il attend entre les tentatives pour réessayer le flushing, ce qui aide à réduire les demandes de connexion à la destination. Vous pouvez désactiver le backoff exponentiel et utiliser la méthode *periodic* retry à la place, qui réessaie de vider les chunks à un intervalle spécifié.

Ces paramètres peuvent vous aider à déterminer les compromis entre la latence et le débit.

- Pour optimiser le débit de Fluentd, vous pouvez utiliser ces paramètres pour réduire le nombre de paquets réseau en configurant des tampons et des files d'attente plus grands, en retardant les vidages et en fixant des délais plus longs entre les tentatives. Sachez que des tampons plus grands nécessitent plus d'espace sur le système de fichiers du nœud.
- Pour optimiser la latence, vous pouvez utiliser les paramètres pour envoyer les données dès que possible, éviter l'accumulation de lots, avoir des files d'attente et des tampons plus courts, et utiliser plus fréquemment des tentatives d'effacement et de réessai.

Vous pouvez configurer le comportement du chunking et du flushing en utilisant les paramètres suivants dans la ressource personnalisée (CR) **ClusterLogging**. Les paramètres sont alors automatiquement ajoutés à la carte de configuration de Fluentd pour être utilisés par Fluentd.



NOTE

Ces paramètres sont les suivants

- Non pertinent pour la plupart des utilisateurs. Les paramètres par défaut devraient permettre d'obtenir de bonnes performances générales.
- Uniquement pour les utilisateurs avancés ayant une connaissance détaillée de la configuration et des performances de Fluentd.
- Uniquement pour l'optimisation des performances. Elles n'ont aucun effet sur les aspects fonctionnels de la journalisation.

Tableau 6.1. Paramètres de configuration avancée de Fluentd

Paramètres	Description	Défaut
chunkLimitSize	La taille maximale de chaque bloc. Fluentd arrête d'écrire des données dans un chunk lorsqu'il atteint cette taille. Ensuite, Fluentd envoie le chunk dans la file d'attente et ouvre un nouveau chunk.	**8m**
totalLimitSize	La taille maximale du tampon, qui est la taille totale de l'étape et de la file d'attente. Si la taille du tampon dépasse cette valeur, Fluentd arrête d'ajouter des données aux chunks et échoue avec une erreur. Toutes les données qui ne sont pas dans des chunks sont perdues.	**8G**
flushInterval	Intervalle entre les vidages de blocs. Vous pouvez utiliser **s** (secondes), **m** (minutes), **h** (heures) ou **d** (jours).	**1s**

Paramètres	Description	Défaut
flushMode	La méthode pour effectuer les rinçages : • **lazy**: Vider les chunks en fonction du paramètre **timekey**. Vous ne pouvez pas modifier le paramètre **timekey**. • **interval**: Purge des blocs en fonction du paramètre **flushInterval**. • **immediate**: Vider les blocs immédiatement après que des données ont été ajoutées à un bloc.	**interval**
flushThreadCount	Le nombre de threads qui effectuent la vidange des blocs. L'augmentation du nombre de threads améliore le débit de vidage, ce qui masque la latence du réseau.	**2**
overflowAction	Le comportement de découpage lorsque la file d'attente est pleine : • **throw_exception**: Lève une exception et l'affiche dans le journal. • **block**: Arrêter le découpage des données jusqu'à ce que le problème de la mémoire tampon complète soit résolu. • **drop_oldest_chunk**: Abandonner le morceau le plus ancien pour accepter les nouveaux morceaux entrants. Les morceaux plus anciens ont moins de valeur que les morceaux plus récents.	**block**

Paramètres	Description	Défaut
retryMaxInterval	Délai maximum en secondes pour la méthode de réessai **exponential_backoff**.	**300s**
retryType	La méthode de réessai en cas d'échec de la vidange : • **exponential_backoff**: Augmenter le temps entre les tentatives de rinçage. Fluentd double le temps d'attente avant la prochaine tentative jusqu'à ce que le paramètre **retry_max_interval** soit atteint. • **periodic**: Réitère les tentatives d'effacement périodiquement, en fonction du paramètre **retryWait**.	**exponential_backoff**
retryTimeOut	Intervalle de temps maximum entre les tentatives avant que l'enregistrement ne soit rejeté.	**60m**
retryWait	Temps en secondes avant le prochain vidage de la mémoire.	**1s**

Pour plus d'informations sur le cycle de vie des chunk de Fluentd, voir Buffer Plugins dans la documentation de Fluentd.

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc edit ClusterLogging instance
```

2. Ajoutez ou modifiez l'un des paramètres suivants :

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  forwarder:
    fluentd:
      buffer:
        chunkLimitSize: 8m (1)
```

```
      flushInterval: 5s (2)
      flushMode: interval (3)
      flushThreadCount: 3 (4)
      overflowAction: throw_exception (5)
      retryMaxInterval: "300s" (6)
      retryType: periodic (7)
      retryWait: 1s (8)
      totalLimitSize: 32m (9)
...
```

1. Spécifiez la taille maximale de chaque bloc avant qu'il ne soit mis en file d'attente pour la vidange.
2. Spécifiez l'intervalle entre les vidanges de blocs.
3. Spécifiez la méthode pour effectuer les vidanges de blocs : **lazy**, **interval**, ou **immediate**.
4. Spécifier le nombre de threads à utiliser pour les vidanges de blocs.
5. Spécifiez le comportement de découpage lorsque la file d'attente est pleine : **throw_exception**, **block**, ou **drop_oldest_chunk**.
6. Spécifiez l'intervalle maximal en secondes pour la méthode de vidange des blocs **exponential_backoff**.
7. Spécifiez le type de réessai en cas d'échec de la vidange des blocs : **exponential_backoff** ou **periodic**.
8. Indiquer le délai en secondes avant le prochain vidage de morceaux.
9. Spécifier la taille maximale de la mémoire tampon.

3. Vérifier que les pods Fluentd sont redéployés :

```
$ oc get pods -l component=collector -n openshift-logging
```

4. Vérifiez que les nouvelles valeurs se trouvent dans la carte de configuration **fluentd**:

```
$ oc extract configmap/fluentd --confirm
```

Exemple fluentd.conf

```
<buffer>
 @type file
 path '/var/lib/fluentd/default'
 flush_mode interval
 flush_interval 5s
 flush_thread_count 3
 retry_type periodic
 retry_wait 1s
 retry_max_interval 300s
 retry_timeout 60m
 queued_chunks_limit_size "#{ENV['BUFFER_QUEUE_LIMIT'] || '32'}"
```

```
  total_limit_size 32m
  chunk_limit_size 8m
  overflow_action throw_exception
</buffer>
```

6.2.5. Suppression des composants inutilisés si vous n'utilisez pas le magasin de logs Elasticsearch par défaut

En tant qu'administrateur, dans le cas rare où vous transmettez les journaux à un magasin de journaux tiers et n'utilisez pas le magasin de journaux Elasticsearch par défaut, vous pouvez supprimer plusieurs composants inutilisés de votre cluster de journalisation.

En d'autres termes, si vous n'utilisez pas le magasin de logs Elasticsearch par défaut, vous pouvez supprimer les composants internes Elasticsearch **logStore** et Kibana **visualization** de la ressource personnalisée (CR) **ClusterLogging**. La suppression de ces composants est facultative mais permet d'économiser des ressources.

Conditions préalables

- Vérifiez que votre redirecteur de logs n'envoie pas les données de logs au cluster Elasticsearch interne par défaut. Inspectez le fichier YAML **ClusterLogForwarder** CR que vous avez utilisé pour configurer le transfert de journaux. Vérifiez qu'il *does not* comporte un élément **outputRefs** qui spécifie **default**. Par exemple :

```
outputRefs:
- default
```

AVERTISSEMENT

Supposons que la CR **ClusterLogForwarder** transmette les données de journal au cluster Elasticsearch interne et que vous supprimiez le composant **logStore** de la CR **ClusterLogging**. Dans ce cas, le cluster Elasticsearch interne ne sera pas présent pour stocker les données du journal. Cette absence peut entraîner une perte de données.

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc edit ClusterLogging instance
```

2. S'ils sont présents, supprimez les strophes **logStore** et **visualization** du CR **ClusterLogging**.

3. Conservez la strophe **collection** de la CR **ClusterLogging**. Le résultat devrait ressembler à l'exemple suivant :

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
```

```
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  collection:
    logs:
      type: "fluentd"
      fluentd: {}
```

4. Vérifier que les pods collecteurs sont redéployés :

```
$ oc get pods -l component=collector -n openshift-logging
```

Ressources complémentaires

- Transmission des journaux à des systèmes tiers

6.3. CONFIGURATION DE L'ENTREPÔT DE DONNÉES

Le sous-système de journalisation de Red Hat OpenShift utilise Elasticsearch 6 (ES) pour stocker et organiser les données de journalisation.

Vous pouvez apporter des modifications à votre magasin de journaux, notamment :

- stockage pour votre cluster Elasticsearch
- la réplication de la pile de données entre les nœuds de données du cluster, de la réplication complète à l'absence de réplication
- accès externe aux données Elasticsearch

Elasticsearch est une application gourmande en mémoire. Chaque nœud Elasticsearch a besoin d'au moins 16G de mémoire pour les requêtes et les limites de mémoire, sauf si vous spécifiez autre chose dans la ressource personnalisée **ClusterLogging**. L'ensemble initial de nœuds OpenShift Container Platform peut ne pas être assez grand pour supporter le cluster Elasticsearch. Vous devez ajouter des nœuds supplémentaires au cluster OpenShift Container Platform pour qu'il fonctionne avec la mémoire recommandée ou supérieure, jusqu'à un maximum de 64G pour chaque nœud Elasticsearch.

Chaque nœud Elasticsearch peut fonctionner avec un paramètre de mémoire inférieur, bien que cela ne soit pas recommandé pour les environnements de production.

6.3.1. Transmission des journaux d'audit à la base de données des journaux

Par défaut, OpenShift Logging ne stocke pas les logs d'audit dans le log store interne d'OpenShift Container Platform Elasticsearch. Vous pouvez envoyer les logs d'audit vers ce log store afin, par exemple, de les visualiser dans Kibana.

Pour envoyer les journaux d'audit au magasin de journaux Elasticsearch interne par défaut, par exemple pour afficher les journaux d'audit dans Kibana, vous devez utiliser l'API Log Forwarding.

IMPORTANT

Le magasin interne de journaux Elasticsearch de OpenShift Container Platform ne fournit pas de stockage sécurisé pour les journaux d'audit. Vérifiez que le système vers lequel vous transmettez les journaux d'audit est conforme aux réglementations de votre organisation et du gouvernement et qu'il est correctement sécurisé. Le sous-système de journalisation de Red Hat OpenShift n'est pas conforme à ces réglementations.

Procédure

Pour utiliser l'API Log Forward afin de transmettre les journaux d'audit à l'instance Elasticsearch interne :

1. Créez ou modifiez un fichier YAML qui définit l'objet **ClusterLogForwarder** CR :

   - Créez un CR pour envoyer tous les types de journaux à l'instance Elasticsearch interne. Vous pouvez utiliser l'exemple suivant sans y apporter de modifications :

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  pipelines: 1
  - name: all-to-default
    inputRefs:
    - infrastructure
    - application
    - audit
    outputRefs:
    - default
```

1 Un pipeline définit le type de journaux à transmettre à l'aide de la sortie spécifiée. La sortie par défaut transmet les journaux à l'instance Elasticsearch interne.

NOTE

Vous devez spécifier les trois types de journaux dans le pipeline : application, infrastructure et audit. Si vous ne spécifiez pas un type de journal, ces journaux ne sont pas stockés et seront perdus.

   - Si vous disposez d'un CR **ClusterLogForwarder** existant, ajoutez un pipeline à la sortie par défaut des journaux d'audit. Il n'est pas nécessaire de définir la sortie par défaut. Par exemple, il n'est pas nécessaire de définir la sortie par défaut :

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  outputs:
   - name: elasticsearch-insecure
     type: "elasticsearch"
```

```
    url: http://elasticsearch-insecure.messaging.svc.cluster.local
    insecure: true
  - name: elasticsearch-secure
    type: "elasticsearch"
    url: https://elasticsearch-secure.messaging.svc.cluster.local
    secret:
      name: es-audit
  - name: secureforward-offcluster
    type: "fluentdForward"
    url: https://secureforward.offcluster.com:24224
    secret:
      name: secureforward
pipelines:
  - name: container-logs
    inputRefs:
    - application
    outputRefs:
    - secureforward-offcluster
  - name: infra-logs
    inputRefs:
    - infrastructure
    outputRefs:
    - elasticsearch-insecure
  - name: audit-logs
    inputRefs:
    - audit
    outputRefs:
    - elasticsearch-secure
    - default (1)
```

(1) Ce pipeline envoie les journaux d'audit à l'instance Elasticsearch interne en plus d'une instance externe.

Ressources complémentaires

- Pour plus d'informations sur l'API de transfert de journaux, voir Transfert de journaux à l'aide de l'API de transfert de journaux.

6.3.2. Configuration de la durée de conservation des journaux

Vous pouvez configurer une adresse *retention policy* qui spécifie la durée pendant laquelle le magasin de logs Elasticsearch par défaut conserve des index pour chacune des trois sources de logs : logs d'infrastructure, logs d'application et logs d'audit.

Pour configurer la politique de rétention, vous définissez un paramètre **maxAge** pour chaque source de journal dans la ressource personnalisée (CR) **ClusterLogging**. La CR applique ces valeurs au calendrier de reconduction d'Elasticsearch, qui détermine quand Elasticsearch supprime les index reconduits.

Elasticsearch effectue le roulement d'un index, en déplaçant l'index actuel et en créant un nouvel index, lorsqu'un index répond à l'une des conditions suivantes :

- L'index est plus ancien que la valeur **rollover.maxAge** dans le CR **Elasticsearch**.
- La taille de l'index est supérieure à 40 Go × le nombre de disques primaires.

- Le nombre de documents de l'index est supérieur à 40960 Ko × le nombre de disques primaires.

Elasticsearch supprime les index prorogés en fonction de la politique de rétention que vous configurez. Si vous ne créez pas de politique de rétention pour les sources de logs, les logs sont supprimés par défaut au bout de sept jours.

Conditions préalables

- Le sous-système de journalisation pour Red Hat OpenShift et l'opérateur OpenShift Elasticsearch doivent être installés.

Procédure

Pour configurer la durée de conservation des journaux :

1. Editer le CR **ClusterLogging** pour ajouter ou modifier le paramètre **retentionPolicy**:

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
...
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    retentionPolicy: 1
      application:
        maxAge: 1d
      infra:
        maxAge: 7d
      audit:
        maxAge: 7d
    elasticsearch:
      nodeCount: 3
...
```

1 Spécifiez la durée pendant laquelle Elasticsearch doit conserver chaque source de journal. Saisissez un nombre entier et une désignation de temps : semaines(w), heures(h/H), minutes(m) et secondes(s). Par exemple, **1d** pour un jour. Les journaux antérieurs à **maxAge** sont supprimés. Par défaut, les journaux sont conservés pendant sept jours.

2. Vous pouvez vérifier les paramètres dans la ressource personnalisée (CR) **Elasticsearch**. Par exemple, l'opérateur de journalisation Red Hat OpenShift a mis à jour le document suivant **Elasticsearch** CR pour configurer une politique de rétention qui inclut des paramètres permettant de renouveler les index actifs pour les journaux d'infrastructure toutes les huit heures et les index renouvelés sont supprimés sept jours après le renouvellement. OpenShift Container Platform vérifie toutes les 15 minutes si les index doivent être reconduits.

```
apiVersion: "logging.openshift.io/v1"
kind: "Elasticsearch"
metadata:
  name: "elasticsearch"
spec:
...
  indexManagement:
    policies: 1
```

```
    - name: infra-policy
      phases:
        delete:
          minAge: 7d (2)
        hot:
          actions:
            rollover:
              maxAge: 8h (3)
      pollInterval: 15m (4)
...
```

(1) Pour chaque source de logs, la politique de conservation indique quand supprimer et reconduire les logs de cette source.

(2) Lorsque OpenShift Container Platform supprime les index reportés. Ce paramètre est le **maxAge** que vous avez défini dans le CR **ClusterLogging**.

(3) L'âge de l'index pour OpenShift Container Platform à prendre en compte lors de la reconduction des index. Cette valeur est déterminée par la valeur **maxAge** que vous avez définie dans le CR **ClusterLogging**.

(4) Quand OpenShift Container Platform vérifie si les indices doivent être reconduits. Ce paramètre est par défaut et ne peut pas être modifié.

NOTE

La modification de la CR **Elasticsearch** n'est pas prise en charge. Toutes les modifications apportées aux politiques de conservation doivent être effectuées dans le CR **ClusterLogging**.

L'OpenShift Elasticsearch Operator déploie un job cron pour rouler les index pour chaque mapping en utilisant la politique définie, planifiée en utilisant le **pollInterval**.

```
$ oc get cronjob
```

Exemple de sortie

```
NAME                     SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
elasticsearch-im-app     */15 * * * *   False     0        <none>          4s
elasticsearch-im-audit   */15 * * * *   False     0        <none>          4s
elasticsearch-im-infra   */15 * * * *   False     0        <none>          4s
```

6.3.3. Configuration des demandes de CPU et de mémoire pour le log store

Chaque spécification de composant permet d'ajuster les demandes de CPU et de mémoire. Vous ne devriez pas avoir à ajuster manuellement ces valeurs car l'OpenShift Elasticsearch Operator définit des valeurs suffisantes pour votre environnement.

NOTE

Dans les clusters à grande échelle, la limite de mémoire par défaut pour le conteneur de proxy Elasticsearch peut ne pas être suffisante, ce qui entraîne l'annulation du conteneur de proxy (OOMKilled). Si vous rencontrez ce problème, augmentez les demandes et les limites de mémoire pour le proxy Elasticsearch.

Chaque nœud Elasticsearch peut fonctionner avec un paramètre de mémoire inférieur, bien que cela soit recommandé pour les déploiements en production (**not**). Pour une utilisation en production, vous ne devriez pas avoir moins que les 16Gi par défaut alloués à chaque pod. Il est préférable d'allouer autant de mémoire que possible, jusqu'à 64Gi par pod.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc edit ClusterLogging instance
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
....
spec:
    logStore:
      type: "elasticsearch"
      elasticsearch: 1
        resources:
          limits: 2
            memory: "32Gi"
          requests: 3
            cpu: "1"
            memory: "16Gi"
        proxy: 4
          resources:
            limits:
              memory: 100Mi
            requests:
              memory: 100Mi
```

1. Spécifiez les demandes de CPU et de mémoire pour Elasticsearch si nécessaire. Si vous laissez ces valeurs vides, OpenShift Elasticsearch Operator définit des valeurs par défaut qui devraient être suffisantes pour la plupart des déploiements. Les valeurs par défaut sont **16Gi** pour la demande de mémoire et **1** pour la demande de CPU.

2. Quantité maximale de ressources qu'un module peut utiliser.

3. Les ressources minimales requises pour planifier un pod.

4. Spécifiez les demandes de CPU et de mémoire pour le proxy Elasticsearch si nécessaire. Si

Lors de l'ajustement de la quantité de mémoire Elasticsearch, la même valeur doit être utilisée pour **requests** et **limits**.

Par exemple :

```
resources:
  limits: 1
    memory: "32Gi"
  requests: 2
    cpu: "8"
    memory: "32Gi"
```

1 La quantité maximale de la ressource.

2 Le montant minimum requis.

Kubernetes adhère généralement à la configuration du nœud et ne permet pas à Elasticsearch d'utiliser les limites spécifiées. En définissant la même valeur pour **requests** et **limits**, vous vous assurez qu'Elasticsearch peut utiliser la mémoire que vous souhaitez, en supposant que le nœud dispose de la mémoire nécessaire.

6.3.4. Configuration de la politique de réplication pour le magasin de journaux

Vous pouvez définir comment les shards Elasticsearch sont répliqués sur les nœuds de données du cluster.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc edit clusterlogging instance
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"

....

spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      redundancyPolicy: "SingleRedundancy" 1
```

1 Spécifiez une politique de redondance pour les shards. La modification est appliquée lors de l'enregistrement des changements.

   - **FullRedundancy**. Elasticsearch réplique entièrement les shards primaires de chaque

index sur chaque nœud de données. Cette méthode offre la plus grande sécurité, mais au prix de la plus grande quantité de disque nécessaire et de la plus faible performance.

- **MultipleRedundancy**. Elasticsearch réplique entièrement les shards primaires de chaque index sur la moitié des nœuds de données. Cela permet d'obtenir un bon compromis entre sécurité et performance.

- **SingleRedundancy**. Elasticsearch fait une copie des shards primaires pour chaque index. Les logs sont toujours disponibles et récupérables tant qu'il existe au moins deux nœuds de données. Meilleure performance que MultipleRedundancy, lors de l'utilisation de 5 nœuds ou plus. Vous ne pouvez pas appliquer cette politique aux déploiements d'un seul nœud Elasticsearch.

- **ZeroRedundancy**. Elasticsearch n'effectue pas de copies des shards primaires. Les journaux peuvent être indisponibles ou perdus en cas d'arrêt ou de défaillance d'un nœud. Utilisez ce mode si vous êtes plus préoccupé par les performances que par la sécurité, ou si vous avez mis en œuvre votre propre stratégie de sauvegarde/restauration sur disque/PVC.

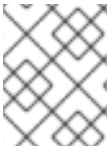
NOTE

Le nombre d'unités primaires pour les modèles d'index est égal au nombre de nœuds de données Elasticsearch.

6.3.5. Réduire la taille des pods Elasticsearch

La réduction du nombre de pods Elasticsearch dans votre cluster peut entraîner une perte de données ou une dégradation des performances d'Elasticsearch.

Si vous réduisez l'échelle, vous devez réduire l'échelle d'un pod à la fois et permettre au cluster de rééquilibrer les shards et les réplicas. Une fois que l'état de santé d'Elasticsearch est revenu à **green**, vous pouvez procéder à une réduction d'échelle pour un autre module.



NOTE

Si votre cluster Elasticsearch est défini sur **ZeroRedundancy**, vous ne devez pas réduire vos pods Elasticsearch.

6.3.6. Configuration du stockage persistant pour le magasin de journaux

Elasticsearch nécessite un stockage persistant. Plus le stockage est rapide, plus les performances d'Elasticsearch sont élevées.

AVERTISSEMENT

L'utilisation du stockage NFS en tant que volume ou volume persistant (ou via un NAS tel que Gluster) n'est pas prise en charge pour le stockage Elasticsearch, car Lucene s'appuie sur un comportement du système de fichiers que NFS ne fournit pas. Une corruption des données et d'autres problèmes peuvent survenir.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Modifiez la CR **ClusterLogging** pour spécifier que chaque nœud de données dans le cluster est lié à une revendication de volume persistant.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
# ...
spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: "gp2"
        size: "200G"
```

Cet exemple précise que chaque nœud de données du cluster est lié à une réclamation de volume persistant qui demande "200G" de stockage AWS General Purpose SSD (gp2).

NOTE

Si vous utilisez un volume local pour le stockage persistant, n'utilisez pas de volume de blocs bruts, qui est décrit avec **volumeMode: block** dans l'objet **LocalVolume**. Elasticsearch ne peut pas utiliser de volumes de blocs bruts.

6.3.7. Configuration du magasin de journaux pour le stockage de emptyDir

Vous pouvez utiliser emptyDir avec votre log store, ce qui crée un déploiement éphémère dans lequel toutes les données d'un pod sont perdues au redémarrage.

NOTE

Lorsque vous utilisez emptyDir, si le stockage des journaux est redémarré ou redéployé, vous perdrez des données.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Modifiez le CR **ClusterLogging** pour spécifier emptyDir :

```
spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage: {}
```

6.3.8. Redémarrage d'un cluster Elasticsearch en continu

Effectuez un redémarrage progressif lorsque vous modifiez la carte de configuration **elasticsearch** ou l'une des configurations de déploiement **elasticsearch-***.

En outre, il est recommandé de procéder à un redémarrage progressif si les nœuds sur lesquels fonctionne un pod Elasticsearch doivent être redémarrés.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

Pour effectuer un redémarrage progressif de la grappe :

1. Modification du projet **openshift-logging**:

```
$ oc project openshift-logging
```

2. Obtenir les noms des pods Elasticsearch :

```
$ oc get pods -l component=elasticsearch-
```

3. Réduire les pods collecteurs afin qu'ils cessent d'envoyer de nouveaux journaux à Elasticsearch :

```
$ oc -n openshift-logging patch daemonset/collector -p '{"spec":{"template":{"spec":
{"nodeSelector":{"logging-infra-collector": "false"}}}}}'
```

4. Effectuer un shard synced flush à l'aide de l'outil OpenShift Container Platform **es_util** pour s'assurer qu'il n'y a pas d'opérations en attente d'écriture sur le disque avant l'arrêt :

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- es_util --query="_flush/synced" -
XPOST
```

Par exemple :

```
$ oc exec -c elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6  -c elasticsearch -- es_util --
query="_flush/synced" -XPOST
```

–

Exemple de sortie

```
{"_shards":{"total":4,"successful":4,"failed":0},".security":
{"total":2,"successful":2,"failed":0},".kibana_1":{"total":2,"successful":2,"failed":0}}
```

5. Empêcher l'équilibrage des shards lors de l'arrêt volontaire des nœuds à l'aide de l'outil es_util d'OpenShift Container Platform :

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- es_util --
query="_cluster/settings" -XPUT -d '{ "persistent" : { "cluster.routing.allocation.enable" : \N-
"primaries\N" } }'
```

Par exemple :

```
$ oc exec elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6 -c elasticsearch -- es_util --
query="_cluster/settings" -XPUT -d '{ "persistent": { "cluster.routing.allocation.enable" :
"primaries" } }'
```

Exemple de sortie

```
{"acknowledged":true,"persistent":{"cluster":{"routing":{"allocation":
{"enable":"primaries"}}}},"transient":
```

6. Une fois la commande terminée, pour chaque déploiement que vous avez pour un cluster ES :

   a. Par défaut, le cluster Elasticsearch d'OpenShift Container Platform bloque les déploiements sur ses nœuds. Utilisez la commande suivante pour autoriser les déploiements et permettre au pod de récupérer les modifications :

   ```
   oc rollout resume deployment/<deployment-name> $ oc rollout resume
   deployment/<deployment-name>
   ```

   Par exemple :

   ```
   $ oc rollout resume deployment/elasticsearch-cdm-0-1
   ```

   Exemple de sortie

   ```
   deployment.extensions/elasticsearch-cdm-0-1 resumed
   ```

   Un nouveau pod est déployé. Une fois que le pod a un conteneur prêt, vous pouvez passer au déploiement suivant.

   ```
   $ oc get pods -l component=elasticsearch-
   ```

   Exemple de sortie

   ```
   NAME                                            READY   STATUS    RESTARTS   AGE
   elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6k    2/2     Running   0          22h
   elasticsearch-cdm-5ceex6ts-2-f799564cb-l9mj7    2/2     Running   0          22h
   elasticsearch-cdm-5ceex6ts-3-585968dc68-k7kjr   2/2     Running   0          22h
   ```

b. Une fois les déploiements terminés, réinitialisez le pod pour interdire les déploiements :

```
oc rollout pause deployment/<deployment-name>
```

Par exemple :

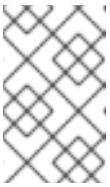
```
$ oc rollout pause deployment/elasticsearch-cdm-0-1
```

Exemple de sortie

```
deployment.extensions/elasticsearch-cdm-0-1 paused
```

c. Vérifiez que le cluster Elasticsearch est dans un état **green** ou **yellow**:

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- es_util --
query=_cluster/health?pretty=true
```



NOTE

Si vous avez effectué un déploiement sur le pod Elasticsearch que vous avez utilisé dans les commandes précédentes, le pod n'existe plus et vous avez besoin d'un nouveau nom de pod ici.

Par exemple :

```
$ oc exec elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6 -c elasticsearch -- es_util --
query=_cluster/health?pretty=true
```

```
{
  "cluster_name" : "elasticsearch",
  "status" : "yellow", (1)
  "timed_out" : false,
  "number_of_nodes" : 3,
  "number_of_data_nodes" : 3,
  "active_primary_shards" : 8,
  "active_shards" : 16,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 1,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
```

(1) Assurez-vous que la valeur de ce paramètre est **green** ou **yellow** avant de poursuivre.

7. Si vous avez modifié la carte de configuration d'Elasticsearch, répétez ces étapes pour chaque module d'Elasticsearch.

8. Une fois que tous les déploiements de la grappe ont été effectués, réactivez l'équilibrage de la grappe :

```
$ oc exec <any_es_pod_in_the_cluster> -c elasticsearch -- es_util --
query="_cluster/settings" -XPUT -d '{ "persistent" : { "cluster.routing.allocation.enable" : "tous"
} }'
```

Par exemple :

```
$ oc exec elasticsearch-cdm-5ceex6ts-1-dcd6c4c7c-jpw6 -c elasticsearch -- es_util --
query="_cluster/settings" -XPUT -d '{ "persistent": { "cluster.routing.allocation.enable" : "all" }
}'
```

Exemple de sortie

```
{
  "acknowledged" : true,
  "persistent" : { },
  "transient" : {
    "cluster" : {
      "routing" : {
        "allocation" : {
          "enable" : "all"
        }
      }
    }
  }
}
```

9. Augmenter les pods collecteurs pour qu'ils envoient de nouveaux journaux à Elasticsearch.

```
$ oc -n openshift-logging patch daemonset/collector -p '{"spec":{"template":{"spec":
{"nodeSelector":{"logging-infra-collector": "true"}}}}}'
```

6.3.9. Exposer le service de stockage de logs en tant que route

Par défaut, le magasin de journaux qui est déployé avec le sous-système de journalisation pour Red Hat OpenShift n'est pas accessible depuis l'extérieur du cluster de journalisation. Vous pouvez activer un itinéraire avec terminaison de rechiffrement pour l'accès externe au service de stockage de journaux pour les outils qui accèdent à ses données.

En externe, vous pouvez accéder au log store en créant une route reencrypt, votre token OpenShift Container Platform et le certificat CA du log store installé. Accédez ensuite à un nœud hébergeant le service de stockage de journaux à l'aide d'une requête cURL contenant :

- Les **Authorization: Bearer ${token}**
- La route Elasticsearch reencrypt et une demande d'API Elasticsearch.

En interne, vous pouvez accéder au service de stockage de logs en utilisant l'IP du cluster de stockage de logs, que vous pouvez obtenir à l'aide de l'une des commandes suivantes :

```
$ oc get service elasticsearch -o jsonpath={.spec.clusterIP} -n openshift-logging
```

Exemple de sortie

```
172.30.183.229
```

```
$ oc get service elasticsearch -n openshift-logging
```

Exemple de sortie

```
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
elasticsearch   ClusterIP   172.30.183.229   <none>        9200/TCP   22h
```

Vous pouvez vérifier l'adresse IP du cluster à l'aide d'une commande similaire à la suivante :

```
$ oc exec elasticsearch-cdm-oplnhinv-1-5746475887-fj2f8 -n openshift-logging -- curl -tlsv1.2 --
insecure -H "Authorization: Bearer ${token}" "https://172.30.183.229:9200/_cat/health"
```

Exemple de sortie

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    29  100    29    0     0    108      0 --:--:-- --:--:-- --:--:--   108
```

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.
- Vous devez avoir accès au projet pour pouvoir accéder aux journaux.

Procédure

Pour exposer le magasin de journaux à l'extérieur :

1. Modification du projet **openshift-logging**:

   ```
   $ oc project openshift-logging
   ```

2. Extraire le certificat de l'autorité de certification de la base de données et l'écrire dans le fichier ***admin-ca*** dans le fichier

   ```
   $ oc extract secret/elasticsearch --to=. --keys=admin-ca
   ```

   #### Exemple de sortie

   ```
   admin-ca
   ```

3. Créez la route pour le service de stockage de journaux sous la forme d'un fichier YAML :

   a. Créez un fichier YAML avec ce qui suit :

   ```
   apiVersion: route.openshift.io/v1
   kind: Route
   metadata:
   ```

```
  name: elasticsearch
  namespace: openshift-logging
spec:
  host:
  to:
    kind: Service
    name: elasticsearch
  tls:
    termination: reencrypt
    destinationCACertificate: | 1
```

1 Ajoutez le certificat de l'autorité de certification du magasin de journaux ou utilisez la commande de l'étape suivante. Il n'est pas nécessaire de définir les paramètres **spec.tls.key**, **spec.tls.certificate** et **spec.tls.caCertificate** requis par certains itinéraires reencrypt.

b. Exécutez la commande suivante pour ajouter le certificat de l'autorité de certification du magasin de journaux à l'itinéraire YAML créé à l'étape précédente :

```
$ cat ./admin-ca | sed -e "s/^/      /" >> <file-name>.yaml
```

c. Créer l'itinéraire :

```
oc create -f <nom-de-fichier>.yaml
```

Exemple de sortie

```
route.route.openshift.io/elasticsearch created
```

4. Vérifier que le service Elasticsearch est exposé :

a. Obtenir le jeton de ce compte de service à utiliser dans la demande :

```
$ token=$(oc whoami -t)
```

b. Définissez la route **elasticsearch** que vous avez créée comme variable d'environnement.

```
$ routeES=`oc get route elasticsearch -o jsonpath={.spec.host}`
```

c. Pour vérifier que la route a été créée avec succès, exécutez la commande suivante qui accède à Elasticsearch via la route exposée :

```
curl -tlsv1.2 --insecure -H "Authorization: Bearer ${token}" "https://${routeES}"
```

La réponse ressemble à ce qui suit :

Exemple de sortie

```
{
  "name" : "elasticsearch-cdm-i40ktba0-1",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "0eY-tJzcR3KOdpgeMJo-MQ",
```

```
  "version" : {
  "number" : "6.8.1",
  "build_flavor" : "oss",
  "build_type" : "zip",
  "build_hash" : "Unknown",
  "build_date" : "Unknown",
  "build_snapshot" : true,
  "lucene_version" : "7.7.0",
  "minimum_wire_compatibility_version" : "5.6.0",
  "minimum_index_compatibility_version" : "5.0.0"
},
  "<tagline>" : "<for search>"
}
```

6.4. CONFIGURATION DU VISUALISATEUR DE JOURNAUX

OpenShift Container Platform utilise Kibana pour afficher les données de journalisation collectées par le sous-système de journalisation.

Vous pouvez dimensionner Kibana pour la redondance et configurer le CPU et la mémoire de vos nœuds Kibana.

6.4.1. Configuration des limites de l'unité centrale et de la mémoire

Les composants du sous-système de journalisation permettent d'ajuster les limites de l'unité centrale et de la mémoire.

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc -n openshift-logging edit ClusterLogging instance
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: openshift-logging

...

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      resources: 1
        limits:
          memory: 16Gi
        requests:
          cpu: 200m
          memory: 16Gi
      storage:
```

```
      storageClassName: "gp2"
      size: "200G"
    redundancyPolicy: "SingleRedundancy"
visualization:
  type: "kibana"
  kibana:
    resources: 2
      limits:
        memory: 1Gi
      requests:
        cpu: 500m
        memory: 1Gi
    proxy:
      resources: 3
        limits:
          memory: 100Mi
        requests:
          cpu: 100m
          memory: 100Mi
    replicas: 2
collection:
  logs:
    type: "fluentd"
    fluentd:
      resources: 4
        limits:
          memory: 736Mi
        requests:
          cpu: 200m
          memory: 736Mi
```

1. Spécifiez les limites de CPU et de mémoire, ainsi que les requêtes pour le magasin de journaux, si nécessaire. Pour Elasticsearch, vous devez ajuster à la fois la valeur de la demande et la valeur de la limite.

2 3. Spécifiez les limites de CPU et de mémoire et les requêtes pour le visualiseur de journaux si nécessaire.

4. Spécifiez les limites de CPU et de mémoire et les requêtes pour le collecteur de journaux si nécessaire.

6.4.2. Redondance d'échelle pour les nœuds du visualisateur de journaux

Vous pouvez mettre à l'échelle le module qui héberge le visualiseur de journaux pour assurer la redondance.

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc edit ClusterLogging instance
```


```
$ oc edit ClusterLogging instance
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"

....

spec:
    visualization:
      type: "kibana"
      kibana:
        replicas: 1 (1)
```

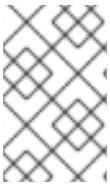
(1) Indiquez le nombre de nœuds Kibana.

6.5. CONFIGURATION DU STOCKAGE DU SOUS-SYSTÈME DE JOURNALISATION

Elasticsearch est une application gourmande en mémoire. L'installation par défaut du sous-système de journalisation déploie 16 Go de mémoire pour les demandes et les limites de mémoire. L'ensemble initial de nœuds OpenShift Container Platform peut ne pas être assez grand pour supporter le cluster Elasticsearch. Vous devez ajouter des nœuds supplémentaires au cluster OpenShift Container Platform pour fonctionner avec la mémoire recommandée ou supérieure. Chaque nœud Elasticsearch peut fonctionner avec un paramètre de mémoire inférieur, bien que cela ne soit pas recommandé pour les environnements de production.

6.5.1. Considérations sur le stockage du sous-système de journalisation pour Red Hat OpenShift

Un volume persistant est nécessaire pour chaque configuration de déploiement d'Elasticsearch. Sur OpenShift Container Platform, cela se fait à l'aide de réclamations de volumes persistants.



NOTE

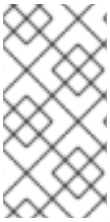
Si vous utilisez un volume local pour le stockage persistant, n'utilisez pas de volume de blocs bruts, qui est décrit avec **volumeMode: block** dans l'objet **LocalVolume**. Elasticsearch ne peut pas utiliser de volumes de blocs bruts.

L'opérateur OpenShift Elasticsearch nomme les PVC en utilisant le nom de la ressource Elasticsearch.

Fluentd envoie tous les journaux de **systemd journal** et **/var/log/containers/** à Elasticsearch.

Elasticsearch a besoin de suffisamment de mémoire pour effectuer des opérations de fusion importantes. S'il n'a pas assez de mémoire, il ne répond plus. Pour éviter ce problème, évaluez la quantité de données de journal d'application dont vous avez besoin et allouez environ le double de cette capacité de stockage libre.

Par défaut, lorsque la capacité de stockage atteint 85ull, Elasticsearch cesse d'allouer de nouvelles données au nœud. À 90%, Elasticsearch tente de relocaliser les shards existants de ce nœud vers d'autres nœuds si possible. Mais si aucun nœud n'a une capacité libre inférieure à 85%, Elasticsearch rejette effectivement la création de nouveaux index et devient RED.



NOTE

Ces valeurs de filigrane basses et hautes sont les valeurs par défaut d'Elasticsearch dans la version actuelle. Vous pouvez modifier ces valeurs par défaut. Bien que les alertes utilisent les mêmes valeurs par défaut, vous ne pouvez pas modifier ces valeurs dans les alertes.

6.5.2. Ressources complémentaires

- Configuration du stockage persistant pour le magasin de journaux

6.6. CONFIGURATION DES LIMITES DE CPU ET DE MÉMOIRE POUR LES COMPOSANTS DU SOUS-SYSTÈME DE JOURNALISATION

Vous pouvez configurer les limites de l'unité centrale et de la mémoire pour chacun des composants du sous-système de journalisation en fonction des besoins.

6.6.1. Configuration des limites de l'unité centrale et de la mémoire

Les composants du sous-système de journalisation permettent d'ajuster les limites de l'unité centrale et de la mémoire.

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc -n openshift-logging edit ClusterLogging instance
```

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: openshift-logging

...

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      resources: 1
        limits:
          memory: 16Gi
        requests:
          cpu: 200m
          memory: 16Gi
      storage:
        storageClassName: "gp2"
        size: "200G"
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana"
```

```
    kibana:
      resources: 2
        limits:
          memory: 1Gi
        requests:
          cpu: 500m
          memory: 1Gi
      proxy:
        resources: 3
          limits:
            memory: 100Mi
          requests:
            cpu: 100m
            memory: 100Mi
      replicas: 2
  collection:
    logs:
      type: "fluentd"
      fluentd:
        resources: 4
          limits:
            memory: 736Mi
          requests:
            cpu: 200m
            memory: 736Mi
```

1 Spécifiez les limites de CPU et de mémoire, ainsi que les requêtes pour le magasin de journaux, si nécessaire. Pour Elasticsearch, vous devez ajuster à la fois la valeur de la demande et la valeur de la limite.

2 3 Spécifiez les limites de CPU et de mémoire et les requêtes pour le visualiseur de journaux si nécessaire.

4 Spécifiez les limites de CPU et de mémoire et les requêtes pour le collecteur de journaux si nécessaire.

6.7. UTILISER LES TOLÉRANCES POUR CONTRÔLER LE PLACEMENT DES PODS OPENSHIFT LOGGING

Vous pouvez utiliser les taints et les tolérances pour vous assurer que les pods du sous-système de journalisation s'exécutent sur des nœuds spécifiques et qu'aucune autre charge de travail ne peut s'exécuter sur ces nœuds.

Les taches et les tolérances sont des paires simples **key:value**. Une tare sur un nœud indique au nœud de repousser tous les pods qui ne tolèrent pas la tare.

key est une chaîne de caractères quelconque, d'une longueur maximale de 253 caractères, et **value** est une chaîne de caractères quelconque, d'une longueur maximale de 63 caractères. La chaîne doit commencer par une lettre ou un chiffre et peut contenir des lettres, des chiffres, des traits d'union, des points et des traits de soulignement.

Sous-système d'enregistrement des échantillons CR avec tolérances

```
apiVersion: "logging.openshift.io/v1"
```

```
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: openshift-logging

...

spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      tolerations: 1
      - key: "logging"
        operator: "Exists"
        effect: "NoExecute"
        tolerationSeconds: 6000
      resources:
        limits:
          memory: 16Gi
        requests:
          cpu: 200m
          memory: 16Gi
      storage: {}
      redundancyPolicy: "ZeroRedundancy"
  visualization:
    type: "kibana"
    kibana:
      tolerations: 2
      - key: "logging"
        operator: "Exists"
        effect: "NoExecute"
        tolerationSeconds: 6000
      resources:
        limits:
          memory: 2Gi
        requests:
          cpu: 100m
          memory: 1Gi
      replicas: 1
  collection:
    logs:
      type: "fluentd"
      fluentd:
        tolerations: 3
        - key: "logging"
          operator: "Exists"
          effect: "NoExecute"
          tolerationSeconds: 6000
        resources:
          limits:
            memory: 2Gi
          requests:
            cpu: 100m
            memory: 1Gi
```

–

1. Cette tolérance est ajoutée aux pods Elasticsearch.
2. Cette tolérance est ajoutée au pod Kibana.
3. Cette tolérance est ajoutée aux pods du collecteur de journalisation.

6.7.1. Utilisation des tolérances pour contrôler le placement des pods du magasin de logs

Vous pouvez contrôler les nœuds sur lesquels les pods de stockage de logs s'exécutent et empêcher d'autres charges de travail d'utiliser ces nœuds en utilisant des tolérances sur les pods.

Vous appliquez des tolérances aux pods de stockage de journaux par le biais de la ressource personnalisée (CR) **ClusterLogging** et vous appliquez des altérations à un nœud par le biais de la spécification de nœud. Une tare sur un nœud est une **key:value pair** qui ordonne au nœud de repousser tous les modules qui ne tolèrent pas la tare. L'utilisation d'une paire spécifique de **key:value** qui n'est pas sur d'autres pods garantit que seuls les pods de stockage de logs peuvent fonctionner sur ce nœud.

Par défaut, les pods de stockage de logs ont la tolérance suivante :

```
tolerations:
- effect: "NoExecute"
  key: "node.kubernetes.io/disk-pressure"
  operator: "Exists"
```

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Utilisez la commande suivante pour ajouter un taint à un nœud où vous souhaitez planifier les pods OpenShift Logging :

   ```
   $ oc adm taint nodes <node-name> <key>=<value>:<effect>
   ```

   Par exemple :

   ```
   $ oc adm taint nodes node1 elasticsearch=node:NoExecute
   ```

   Cet exemple place une taint sur **node1** qui a la clé **elasticsearch**, la valeur **node**, et l'effet de taint **NoExecute**. Les nœuds avec l'effet **NoExecute** programment uniquement les pods qui correspondent à l'altération et suppriment les pods existants qui ne correspondent pas à l'altération.

2. Modifiez la section **logstore** du CR **ClusterLogging** pour configurer une tolérance pour les pods Elasticsearch :

   ```
   logStore:
     type: "elasticsearch"
     elasticsearch:
       nodeCount: 1
   ```

```
tolerations:
- key: "elasticsearch" 1
  operator: "Exists" 2
  effect: "NoExecute" 3
  tolerationSeconds: 6000 4
```

1. Spécifiez la clé que vous avez ajoutée au nœud.
2. Spécifiez l'opérateur **Exists** pour exiger qu'une altération avec la clé **elasticsearch** soit présente sur le nœud.
3. Spécifiez l'effet **NoExecute**.
4. En option, le paramètre **tolerationSeconds** permet de définir la durée pendant laquelle un pod peut rester lié à un nœud avant d'être expulsé.

Cette tolérance correspond à l'altération créée par la commande **oc adm taint**. Un pod avec cette tolérance pourrait être programmé sur **node1**.

6.7.2. Utilisation des tolérances pour contrôler le placement du pod du visualiseur de logs

Vous pouvez contrôler le nœud sur lequel le module de visualisation du journal s'exécute et empêcher d'autres charges de travail d'utiliser ces nœuds en utilisant des tolérances sur les modules.

Vous appliquez des tolérances au pod de visualisation du journal par le biais de la ressource personnalisée (CR) **ClusterLogging** et vous appliquez des taches à un nœud par le biais de la spécification du nœud. Une tare sur un nœud est une **key:value pair** qui demande au nœud de repousser tous les pods qui ne tolèrent pas la tare. L'utilisation d'une paire spécifique de **key:value** qui n'est pas sur d'autres pods garantit que seul le pod Kibana peut fonctionner sur ce nœud.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Utilisez la commande suivante pour ajouter une taint à un nœud où vous souhaitez planifier le pod de visualisation de logs :

   ```
   $ oc adm taint nodes <node-name> <key>=<value>:<effect>
   ```

   Par exemple :

   ```
   $ oc adm taint nodes node1 kibana=node:NoExecute
   ```

   Cet exemple place une taint sur **node1** qui a la clé **kibana**, la valeur **node**, et l'effet de taint **NoExecute**. Vous devez utiliser l'effet de taint **NoExecute**. **NoExecute** planifie uniquement les pods qui correspondent au taint et supprime les pods existants qui ne correspondent pas.

2. Modifiez la section **visualization** de la CR **ClusterLogging** pour configurer une tolérance pour le module Kibana :

```
visualization:
  type: "kibana"
  kibana:
    tolerations:
    - key: "kibana" 1
      operator: "Exists" 2
      effect: "NoExecute" 3
      tolerationSeconds: 6000 4
```

1. Spécifiez la clé que vous avez ajoutée au nœud.
2. Spécifiez l'opérateur **Exists** pour que les paramètres **key/value/effect** correspondent.
3. Spécifiez l'effet **NoExecute**.
4. En option, le paramètre **tolerationSeconds** permet de définir la durée pendant laquelle un pod peut rester lié à un nœud avant d'être expulsé.

Cette tolérance correspond à la taint créée par la commande **oc adm taint**. Un pod avec cette tolérance pourrait être programmé sur **node1**.

6.7.3. Utilisation des tolérances pour contrôler l'emplacement du collecteur de logs

Vous pouvez vous assurer des nœuds sur lesquels les pods du collecteur de journalisation s'exécutent et empêcher d'autres charges de travail d'utiliser ces nœuds en utilisant des tolérances sur les pods.

Vous appliquez des tolérances aux pods collecteurs de journalisation via la ressource personnalisée (CR) **ClusterLogging** et vous appliquez des taints à un nœud via la spécification de nœud. Vous pouvez utiliser les taints et les tolérances pour vous assurer que le pod n'est pas expulsé pour des raisons telles que des problèmes de mémoire ou de CPU.

Par défaut, les pods du collecteur de logs ont la tolérance suivante :

```
tolerations:
- key: "node-role.kubernetes.io/master"
  operator: "Exists"
  effect: "NoExecute"
```

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Utilisez la commande suivante pour ajouter une taint à un nœud où vous souhaitez que les pods collecteurs de journalisation planifient les pods collecteurs de journalisation :

   ```
   $ oc adm taint nodes <node-name> <key>=<value>:<effect>
   ```

   Par exemple :

   ```
   $ oc adm taint nodes node1 collector=node:NoExecute
   ```

Cet exemple place une taint sur **node1** qui a la clé **collector**, la valeur **node**, et l'effet de taint **NoExecute**. Vous devez utiliser l'effet de taint **NoExecute**. **NoExecute** planifie uniquement les pods qui correspondent au taint et supprime les pods existants qui ne correspondent pas.

2. Modifiez la strophe **collection** de la ressource personnalisée (CR) **ClusterLogging** pour configurer une tolérance pour les pods du collecteur de journalisation :

```
collection:
  logs:
    type: "fluentd"
    fluentd:
      tolerations:
      - key: "collector" (1)
        operator: "Exists" (2)
        effect: "NoExecute" (3)
        tolerationSeconds: 6000 (4)
```

(1) Spécifiez la clé que vous avez ajoutée au nœud.

(2) Spécifiez l'opérateur **Exists** pour que les paramètres **key/value/effect** correspondent.

(3) Spécifiez l'effet **NoExecute**.

(4) En option, le paramètre **tolerationSeconds** permet de définir la durée pendant laquelle un pod peut rester lié à un nœud avant d'être expulsé.

Cette tolérance correspond à la taint créée par la commande **oc adm taint**. Un pod avec cette tolérance pourrait être programmé sur **node1**.

6.7.4. Ressources complémentaires

- Contrôle du placement de pods à l'aide de taches de nœuds .

6.8. DÉPLACEMENT DES RESSOURCES DU SOUS-SYSTÈME DE JOURNALISATION À L'AIDE DE SÉLECTEURS DE NŒUDS

Vous pouvez utiliser des sélecteurs de nœuds pour déployer les pods Elasticsearch et Kibana sur différents nœuds.

6.8.1. Déplacer les ressources de journalisation d'OpenShift

Vous pouvez configurer le Cluster Logging Operator pour déployer les pods des composants du sous-système de journalisation, tels qu'Elasticsearch et Kibana, sur différents nœuds. Vous ne pouvez pas déplacer le pod Cluster Logging Operator de son emplacement d'installation.

Par exemple, vous pouvez déplacer les pods Elasticsearch vers un nœud distinct en raison des exigences élevées en matière de CPU, de mémoire et de disque.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés. Ces fonctionnalités ne sont pas installées par défaut.

Procédure

1. Modifiez la ressource personnalisée (CR) **ClusterLogging** dans le projet **openshift-logging**:

```
$ oc edit ClusterLogging instance
```

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging

...

spec:
  collection:
    logs:
      fluentd:
        resources: null
      type: fluentd
  logStore:
    elasticsearch:
      nodeCount: 3
      nodeSelector: 1
        node-role.kubernetes.io/infra: ''
      tolerations:
      - effect: NoSchedule
        key: node-role.kubernetes.io/infra
        value: reserved
      - effect: NoExecute
        key: node-role.kubernetes.io/infra
        value: reserved
      redundancyPolicy: SingleRedundancy
      resources:
        limits:
          cpu: 500m
          memory: 16Gi
        requests:
          cpu: 500m
          memory: 16Gi
      storage: {}
    type: elasticsearch
  managementState: Managed
  visualization:
    kibana:
      nodeSelector: 2
        node-role.kubernetes.io/infra: ''
      tolerations:
      - effect: NoSchedule
        key: node-role.kubernetes.io/infra
        value: reserved
      - effect: NoExecute
        key: node-role.kubernetes.io/infra
        value: reserved
      proxy:
        resources: null
      replicas: 1
      resources: null
```

```
    type: kibana

...
```

1 2 Ajoutez un paramètre **nodeSelector** avec la valeur appropriée au composant que vous souhaitez déplacer. Vous pouvez utiliser un **nodeSelector** dans le format indiqué ou utiliser des paires **<key>: <value>**, en fonction de la valeur spécifiée pour le nœud. Si vous avez ajouté une tare au nœud de l'infrasructure, ajoutez également une tolérance correspondante.

Vérification

Pour vérifier qu'un composant a été déplacé, vous pouvez utiliser la commande **oc get pod -o wide**.

Par exemple :

- Vous souhaitez déplacer le pod Kibana du nœud **ip-10-0-147-79.us-east-2.compute.internal**:

```
$ oc get pod kibana-5b8bdf44f9-ccpq9 -o wide
```

Exemple de sortie

```
NAME                     READY   STATUS    RESTARTS   AGE   IP            NODE
NOMINATED NODE   READINESS GATES
kibana-5b8bdf44f9-ccpq9  2/2     Running   0          27s   10.129.2.18   ip-10-0-147-79.us-
east-2.compute.internal   <none>           <none>
```

- Vous souhaitez déplacer le pod Kibana vers le nœud **ip-10-0-139-48.us-east-2.compute.internal**, un nœud d'infrastructure dédié :

```
$ oc get nodes
```

Exemple de sortie

```
NAME                                         STATUS   ROLES          AGE   VERSION
ip-10-0-133-216.us-east-2.compute.internal   Ready    master         60m   v1.25.0
ip-10-0-139-146.us-east-2.compute.internal   Ready    master         60m   v1.25.0
ip-10-0-139-192.us-east-2.compute.internal   Ready    worker         51m   v1.25.0
ip-10-0-139-241.us-east-2.compute.internal   Ready    worker         51m   v1.25.0
ip-10-0-147-79.us-east-2.compute.internal    Ready    worker         51m   v1.25.0
ip-10-0-152-241.us-east-2.compute.internal   Ready    master         60m   v1.25.0
ip-10-0-139-48.us-east-2.compute.internal    Ready    infra          51m   v1.25.0
```

Notez que le nœud a une étiquette **node-role.kubernetes.io/infra: ''**:

```
$ oc get node ip-10-0-139-48.us-east-2.compute.internal -o yaml
```

Exemple de sortie

```
kind: Node
apiVersion: v1
metadata:
  name: ip-10-0-139-48.us-east-2.compute.internal
```

```
  selfLink: /api/v1/nodes/ip-10-0-139-48.us-east-2.compute.internal
  uid: 62038aa9-661f-41d7-ba93-b5f1b6ef8751
  resourceVersion: '39083'
  creationTimestamp: '2020-04-13T19:07:55Z'
  labels:
    node-role.kubernetes.io/infra: ''
...
```

- Pour déplacer le pod Kibana, modifiez le CR **ClusterLogging** pour ajouter un sélecteur de nœud :

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging

...

spec:

...

  visualization:
    kibana:
      nodeSelector: 1
        node-role.kubernetes.io/infra: ''
      proxy:
        resources: null
      replicas: 1
      resources: null
    type: kibana
```

1 Ajouter un sélecteur de nœud correspondant à l'étiquette de la spécification du nœud.

- Après avoir sauvegardé le CR, le pod Kibana actuel est terminé et le nouveau pod est déployé :

```
$ oc get pods
```

Exemple de sortie

```
NAME                                            READY   STATUS        RESTARTS   AGE
cluster-logging-operator-84d98649c4-zb9g7       1/1     Running       0          29m
elasticsearch-cdm-hwv01pf7-1-56588f554f-kpmlg   2/2     Running       0          28m
elasticsearch-cdm-hwv01pf7-2-84c877d75d-75wqj   2/2     Running       0          28m
elasticsearch-cdm-hwv01pf7-3-f5d95b87b-4nx78    2/2     Running       0          28m
fluentd-42dzz                                   1/1     Running       0          28m
fluentd-d74rq                                   1/1     Running       0          28m
fluentd-m5vr9                                   1/1     Running       0          28m
fluentd-nkxl7                                   1/1     Running       0          28m
fluentd-pdvqb                                   1/1     Running       0          28m
fluentd-tflh6                                   1/1     Running       0          28m
kibana-5b8bdf44f9-ccpq9                         2/2     Terminating   0          4m11s
kibana-7d85dcffc8-bfpfp                         2/2     Running       0          33s
```

- Le nouveau pod se trouve sur le nœud **ip-10-0-139-48.us-east-2.compute.internal**:

```
$ oc get pod kibana-7d85dcffc8-bfpfp -o wide
```

Exemple de sortie

```
NAME                     READY   STATUS       RESTARTS   AGE   IP          NODE
NOMINATED NODE   READINESS GATES
kibana-7d85dcffc8-bfpfp  2/2     Running      0          43s   10.131.0.22   ip-10-0-139-48.us-
east-2.compute.internal   <none>           <none>
```

- Après quelques instants, le pod Kibana original est retiré.

```
$ oc get pods
```

Exemple de sortie

```
NAME                                READY   STATUS    RESTARTS   AGE
cluster-logging-operator-84d98649c4-zb9g7       1/1     Running   0          30m
elasticsearch-cdm-hwv01pf7-1-56588f554f-kpmlg   2/2     Running   0          29m
elasticsearch-cdm-hwv01pf7-2-84c877d75d-75wqj   2/2     Running   0          29m
elasticsearch-cdm-hwv01pf7-3-f5d95b87b-4nx78    2/2     Running   0          29m
fluentd-42dzz                                   1/1     Running   0          29m
fluentd-d74rq                                   1/1     Running   0          29m
fluentd-m5vr9                                   1/1     Running   0          29m
fluentd-nkxl7                                   1/1     Running   0          29m
fluentd-pdvqb                                   1/1     Running   0          29m
fluentd-tflh6                                   1/1     Running   0          29m
kibana-7d85dcffc8-bfpfp                         2/2     Running   0          62s
```

6.9. CONFIGURATION DE SYSTEMD-JOURNALD ET FLUENTD

Parce que Fluentd lit à partir du journal, et que les paramètres par défaut du journal sont très bas, les entrées du journal peuvent être perdues parce que le journal ne peut pas suivre le taux d'enregistrement des services du système.

Nous recommandons de définir **RateLimitIntervalSec=30s** et **RateLimitBurst=10000** (ou plus si nécessaire) pour éviter que le journal ne perde des entrées.

6.9.1. Configurer systemd-journald pour OpenShift Logging

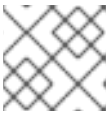
Au fur et à mesure que vous développez votre projet, l'environnement de journalisation par défaut peut nécessiter quelques ajustements.

Par exemple, si vous manquez de logs, vous devrez peut-être augmenter les limites de taux pour journald. Vous pouvez ajuster le nombre de messages à conserver pendant une période de temps spécifiée pour s'assurer que OpenShift Logging n'utilise pas de ressources excessives sans laisser tomber les journaux.

Vous pouvez également déterminer si vous souhaitez que les journaux soient compressés, combien de temps ils doivent être conservés, comment ou si les journaux sont stockés, ainsi que d'autres paramètres.

Procédure

1. Créez un fichier de configuration Butane, **40-worker-custom-journald.bu**, qui inclut un fichier **/etc/systemd/journald.conf** avec les paramètres requis.



NOTE

See "Creating machine configs with Butane" for information about Butane.

```
variant: openshift
version: 4.12.0
metadata:
  name: 40-worker-custom-journald
  labels:
    machineconfiguration.openshift.io/role: "worker"
storage:
  files:
  - path: /etc/systemd/journald.conf
    mode: 0644 (1)
    overwrite: true
    contents:
      inline: |
        Compress=yes (2)
        ForwardToConsole=no (3)
        ForwardToSyslog=no
        MaxRetentionSec=1month (4)
        RateLimitBurst=10000 (5)
        RateLimitIntervalSec=30s
        Storage=persistent (6)
        SyncIntervalSec=1s (7)
        SystemMaxUse=8G (8)
        SystemKeepFree=20% (9)
        SystemMaxFileSize=10M (10)
```

(1) Définissez les autorisations pour le fichier **journal.conf**. Il est recommandé de définir les autorisations pour **0644**.

(2) Indiquez si vous souhaitez que les journaux soient compressés avant d'être écrits sur le système de fichiers. Indiquez **yes** pour compresser le message ou **no** pour ne pas le compresser. La valeur par défaut est **yes**.

(3) Permet de déterminer s'il convient de transmettre les messages du journal. La valeur par défaut est **no** pour chaque message. Spécifier :

- **ForwardToConsole** pour transmettre les journaux à la console du système.
- **ForwardToKsmg** pour acheminer les journaux vers le tampon de journaux du noyau.
- **ForwardToSyslog** à transmettre à un démon syslog.
- **ForwardToWall** pour transférer les messages en tant que messages muraux à tous les utilisateurs connectés.

(4) Indiquez la durée maximale de stockage des écritures de journal. Entrez un nombre pour spécifier les secondes. Ou indiquez une unité : \N- "year\N", \N- "month\N", \N- "week\N", \N- "day\N", \N- "h\N" ou \N- "m\N". Entrez **0** pour désactiver. La valeur par défaut est

1month.

5 Configurer la limitation du débit. Si le nombre de journaux reçus est supérieur à celui spécifié dans **RateLimitBurst** pendant l'intervalle de temps défini par **RateLimitIntervalSec**, tous les messages supplémentaires dans l'intervalle sont abandonnés jusqu'à ce que l'intervalle soit écoulé. Il est recommandé de définir **RateLimitIntervalSec=30s** et **RateLimitBurst=10000**, qui sont les valeurs par défaut.

6 Spécifiez comment les journaux sont stockés. La valeur par défaut est **persistent**:

- **volatile** pour stocker les journaux en mémoire à l'adresse **/var/log/journal/**.
- **persistent** pour stocker les journaux sur le disque à l'adresse **/var/log/journal/**. systemd crée le répertoire s'il n'existe pas.
- **auto** pour stocker les journaux dans **/var/log/journal/** si le répertoire existe. S'il n'existe pas, systemd stocke temporairement les journaux dans **/run/systemd/journal**.
- **none** pour ne pas stocker les journaux. systemd supprime tous les journaux.

7 Spécifiez le délai d'attente avant de synchroniser les fichiers journaux sur le disque pour les journaux **ERR**, **WARNING**, **NOTICE**, **INFO**, et **DEBUG**. systemd synchronise immédiatement après avoir reçu un journal **CRIT**, **ALERT**, ou **EMERG**. La valeur par défaut est **1s**.

8 Spécifiez la taille maximale que le journal peut utiliser. La valeur par défaut est **8G**.

9 Spécifiez l'espace disque que systemd doit laisser libre. La valeur par défaut est **20%**.

10 Spécifiez la taille maximale des fichiers journaux individuels stockés de manière persistante sur **/var/log/journal**. La valeur par défaut est **10M**.

NOTE

Si vous supprimez la limite de débit, il se peut que vous constatiez une augmentation de l'utilisation de l'unité centrale sur les démons de journalisation du système, car ils traitent les messages qui auraient été précédemment limités.

Pour plus d'informations sur les paramètres de systemd, voir https://www.freedesktop.org/software/systemd/man/journald.conf.html. Les paramètres par défaut énumérés sur cette page peuvent ne pas s'appliquer à OpenShift Container Platform.

2. Utilisez Butane pour générer un fichier objet **MachineConfig**, **40-worker-custom-journald.yaml**, contenant la configuration à fournir aux nœuds :

```
$ butane 40-worker-custom-journald.bu -o 40-worker-custom-journald.yaml
```

3. Appliquer la configuration de la machine. Par exemple :

```
$ oc apply -f 40-worker-custom-journald.yaml
```

Le contrôleur détecte le nouvel objet **MachineConfig** et génère une nouvelle version **rendered-worker-<hash>**.

4. Contrôler l'état du déploiement de la nouvelle configuration rendue à chaque nœud :

```
$ oc describe machineconfigpool/worker
```

Exemple de sortie

```
Name:         worker
Namespace:
Labels:       machineconfiguration.openshift.io/mco-built-in=
Annotations:  <none>
API Version:  machineconfiguration.openshift.io/v1
Kind:         MachineConfigPool

...

Conditions:
  Message:
  Reason:               All nodes are updating to rendered-worker-
913514517bcea7c93bd446f4830bc64e
```

6.10. MAINTENANCE ET ASSISTANCE

6.10.1. À propos des configurations non prises en charge

La manière supportée de configurer le sous-système de journalisation pour Red Hat OpenShift est de le configurer en utilisant les options décrites dans cette documentation. N'utilisez pas d'autres configurations, car elles ne sont pas prises en charge. Les paradigmes de configuration peuvent changer à travers les versions d'OpenShift Container Platform, et de tels cas ne peuvent être gérés avec élégance que si toutes les possibilités de configuration sont contrôlées. Si vous utilisez des configurations autres que celles décrites dans cette documentation, vos changements disparaîtront car l'OpenShift Elasticsearch Operator et le Red Hat OpenShift Logging Operator réconcilient toutes les différences. Les opérateurs inversent tout à l'état défini par défaut et par conception.

NOTE

Si vous *must* effectuez des configurations non décrites dans la documentation d'OpenShift Container Platform, vous *must* configurez votre Red Hat OpenShift Logging Operator ou OpenShift Elasticsearch Operator sur **Unmanaged**. Un environnement OpenShift Logging non géré est *not supported* et ne reçoit pas de mises à jour jusqu'à ce que vous remettiez OpenShift Logging sur **Managed**.

6.10.2. Configurations non prises en charge

Vous devez configurer l'Opérateur de journalisation de Red Hat OpenShift à l'état non géré pour modifier les composants suivants :

- Le CR **Elasticsearch**
- Le déploiement de Kibana

- Le fichier **fluent.conf**
- Le jeu de démons Fluentd

Vous devez mettre l'OpenShift Elasticsearch Operator à l'état non géré pour modifier le composant suivant :

- les fichiers de déploiement d'Elasticsearch.

Les cas explicitement non pris en charge sont les suivants :

- **Configuring default log rotation**. Vous ne pouvez pas modifier la configuration par défaut de la rotation des journaux.
- **Configuring the collected log location** Vous ne pouvez pas modifier l'emplacement du fichier de sortie du collecteur de journaux, qui est par défaut **/var/log/fluentd/fluentd.log**.
- **Throttling log collection**. Vous ne pouvez pas réduire la vitesse à laquelle les journaux sont lus par le collecteur de journaux.
- **Configuring the logging collector using environment variables** Vous ne pouvez pas utiliser de variables d'environnement pour modifier le collecteur de journaux.
- **Configuring how the log collector normalizes logs**Vous ne pouvez pas modifier la normalisation des journaux par défaut.

6.10.3. Politique de soutien aux opérateurs non gérés

Le site *management state* d'un opérateur détermine si celui-ci gère activement les ressources de son composant dans le cluster comme prévu. Si un opérateur est dans l'état *unmanaged*, il ne réagit pas aux changements de configuration et ne reçoit pas de mises à jour.

Bien que cela puisse être utile dans les clusters de non-production ou lors du débogage, les opérateurs dans un état non géré ne sont pas pris en charge et l'administrateur du cluster assume le contrôle total des configurations et des mises à niveau des composants individuels.

Un opérateur peut être placé dans un état non géré à l'aide des méthodes suivantes :

- Individual Operator configuration
  Chaque opérateur dispose d'un paramètre **managementState** dans sa configuration. Il est possible d'y accéder de différentes manières, en fonction de l'opérateur. Par exemple, l'opérateur de journalisation de Red Hat OpenShift accomplit ceci en modifiant une ressource personnalisée (CR) qu'il gère, tandis que l'opérateur d'échantillons de clusters utilise une ressource de configuration à l'échelle du cluster.

  La modification du paramètre **managementState** en **Unmanaged** signifie que l'opérateur ne gère pas activement ses ressources et qu'il ne prendra aucune mesure concernant le composant en question. Certains opérateurs peuvent ne pas supporter cet état de gestion car il pourrait endommager le cluster et nécessiter une récupération manuelle.

AVERTISSEMENT

Le passage d'opérateurs individuels à l'état **Unmanaged** rend ce composant et cette fonctionnalité particuliers non pris en charge. Les problèmes signalés doivent être reproduits dans l'état **Managed** pour que l'assistance soit assurée.

- **Cluster Version Operator (CVO) overrides**
  Le paramètre **spec.overrides** peut être ajouté à la configuration de l'OVC pour permettre aux administrateurs de fournir une liste de dérogations au comportement de l'OVC pour un composant. La définition du paramètre **spec.overrides[].unmanaged** à **true** pour un composant bloque les mises à niveau du cluster et alerte l'administrateur lorsqu'une dérogation de l'OVC a été définie :

> Disabling ownership via cluster version overrides prevents upgrades. Please remove overrides before continuing.

AVERTISSEMENT

La mise en place d'une surcharge CVO place l'ensemble du cluster dans un état non supporté. Les problèmes signalés doivent être reproduits après la suppression de toute surcharge pour que l'assistance soit assurée.

CHAPITRE 7. LOKI

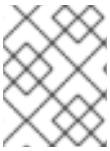
7.1. À PROPOS DE LOKISTACK

Dans la documentation du sous-système de journalisation, **LokiStack** fait référence à la combinaison du sous-système de journalisation pris en charge par Loki et le proxy web avec l'intégration de l'authentification de la plate-forme OpenShift Container. Le proxy de LokiStack utilise l'authentification OpenShift Container Platform pour renforcer la multi-location. **Loki** fait référence au magasin de logs en tant que composant individuel ou magasin externe.

Loki est un système d'agrégation de logs horizontalement extensible, hautement disponible et multi-tenant, actuellement proposé comme alternative à Elasticsearch en tant que magasin de logs pour le sous-système de logs. Elasticsearch indexe complètement les enregistrements de logs entrants lors de l'ingestion. Loki n'indexe que quelques étiquettes fixes lors de l'ingestion, et reporte l'analyse plus complexe jusqu'à ce que les journaux aient été stockés. Cela signifie que Loki peut collecter les journaux plus rapidement. Comme pour Elasticsearch, vous pouvez interroger Loki à l'aide de chemins JSON ou d'expressions régulières.

7.1.1. Dimensionnement du déploiement

Le dimensionnement de Loki suit le format suivant **N<x>.*<size>*** où la valeur **<N>** correspond au nombre d'instances et **<size>** spécifie les capacités de performance.



NOTE

1x.extra-small est utilisé à des fins de démonstration uniquement et n'est pas pris en charge.

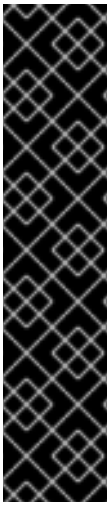
Tableau 7.1. Taille Loki

	1x.extra-petit	1x.petit	1x.moyen
Data transfer	Utilisation à des fins de démonstration uniquement.	500GB/jour	2TB/jour
Queries per second (QPS)	Utilisation à des fins de démonstration uniquement.	25-50 QPS à 200ms	25-75 QPS à 200ms
Replication factor	Aucun	2	3
Total CPU requests	5 vCPU	36 vCPUs	54 vCPUs
Total Memory requests	7.5Gi	63Gi	139Gi
Total Disk requests	150Gi	300Gi	450Gi

7.1.2. Définitions de ressources personnalisées API prises en charge

LokiStack est en cours de développement, toutes les API ne sont pas encore prises en charge.

CustomResourceDefinition (CRD)	ApiVersion	État d'appui
LokiStack	lokistack.loki.grafana.com/v1	Pris en charge dans la version 5.5
RulerConfig	rulerconfig.loki.grafana/v1beta1	Avant-première technologique
Règle d'alerte	alertingrule.loki.grafana/v1beta1	Avant-première technologique
Règle d'enregistrement	recordingrule.loki.grafana/v1beta1	Avant-première technologique



IMPORTANT

L'utilisation de **RulerConfig**, **AlertingRule** et **RecordingRule** custom resource definitions (CRDs) est une fonctionnalité d'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat et peuvent ne pas être complètes d'un point de vue fonctionnel. Red Hat ne recommande pas leur utilisation en production. Ces fonctionnalités offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Pour plus d'informations sur la portée de l'assistance des fonctionnalités de l'aperçu technologique de Red Hat, voir Portée de l'assistance des fonctionnalités de l'aperçu technologique.

7.2. DÉPLOIEMENT DE LA LOKISTACK

Vous pouvez utiliser la console web d'OpenShift Container Platform pour déployer LokiStack.

Conditions préalables

- Sous-système de journalisation pour Red Hat OpenShift Operator 5.5 et versions ultérieures
- Log Store pris en charge (AWS S3, Google Cloud Storage, Azure, Swift, Minio, OpenShift Data Foundation)

Procédure

1. Installer l'opérateur **Loki Operator**:
   a. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → OperatorHub**.
   b. Choisissez **Loki Operator** dans la liste des opérateurs disponibles et cliquez sur **Install**.
   c. Sous **Installation Mode**, sélectionnez **All namespaces on the cluster**.
   d. Sous **Installed Namespace**, sélectionnez **openshift-operators-redhat**.
   Vous devez spécifier l'espace de noms **openshift-operators-redhat**. L'espace de noms

openshift-operators peut contenir des Community Operators, qui ne sont pas fiables et qui pourraient publier une métrique avec le même nom qu'une métrique OpenShift Container Platform, ce qui causerait des conflits.

e. Sélectionnez **Enable operator recommended cluster monitoring on this namespace** Cette option définit l'étiquette **openshift.io/cluster-monitoring: "true"** dans l'objet Namespace. Vous devez sélectionner cette option pour vous assurer que la surveillance des clusters récupère l'espace de noms **openshift-operators-redhat**.

f. Sélectionnez un site **Approval Strategy**.

- La stratégie **Automatic** permet à Operator Lifecycle Manager (OLM) de mettre automatiquement à jour l'opérateur lorsqu'une nouvelle version est disponible.
- La stratégie **Manual** exige qu'un utilisateur disposant des informations d'identification appropriées approuve la mise à jour de l'opérateur.

g. Cliquez sur **Install**.

h. Vérifiez que vous avez installé l'opérateur Loki. Visitez la page **Operators → Installed Operators** et cherchez **Loki Operator**.

i. Veillez à ce que **Loki Operator** soit listé avec **Status** et **Succeeded** dans tous les projets.

2. Créez un fichier YAML **Secret** qui utilise les champs **access_key_id** et **access_key_secret** pour spécifier vos informations d'identification AWS et **bucketnames**, **endpoint** et **region** pour définir l'emplacement de stockage de l'objet. Par exemple :

```
apiVersion: v1
kind: Secret
metadata:
  name: logging-loki-s3
  namespace: openshift-logging
stringData:
  access_key_id: AKIAIOSFODNN7EXAMPLE
  access_key_secret: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
  bucketnames: s3-bucket-name
  endpoint: https://s3.eu-central-1.amazonaws.com
  region: eu-central-1
```

3. Créer la ressource personnalisée **LokiStack**:

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  size: 1x.small
  storage:
    schemas:
    - version: v12
      effectiveDate: "2022-06-01"
    secret:
      name: logging-loki-s3
      type: s3
```

```
    storageClassName: gp3-csi (1)
    tenants:
      mode: openshift-logging
```

(1) Ou **gp2-csi**.

a. Appliquer la configuration :

```
oc apply -f logging-loki.yaml
```

4. Créer ou modifier un CR **ClusterLogging**:

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  managementState: Managed
  logStore:
    type: lokistack
    lokistack:
      name: logging-loki
  collection:
    type: vector
```

a. Appliquer la configuration :

```
oc apply -f cr-lokistack.yaml
```

5. Activez le plugin RedHat OpenShift Logging Console :

a. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → Installed Operators**.

b. Sélectionnez l'opérateur **RedHat OpenShift Logging**.

c. Sous le plugin Console, cliquez sur **Disabled**.

d. Sélectionnez **Enable** puis **Save**. Ce changement va redémarrer les pods 'openshift-console'.

e. Après le redémarrage des pods, vous recevrez une notification indiquant qu'une mise à jour de la console web est disponible, vous invitant à l'actualiser.

f. Après avoir actualisé la console web, cliquez sur **Observe** dans le menu principal de gauche. Une nouvelle option pour **Logs** sera disponible.

NOTE

Ce plugin n'est disponible que sur OpenShift Container Platform 4.10 et plus.

7.3. ACTIVATION DE LA RÉTENTION BASÉE SUR LES FLUX AVEC LOKI

À partir de la version 5.6 de Logging, vous pouvez configurer des politiques de rétention basées sur les flux de logs. Les règles peuvent être définies globalement, par locataire ou les deux. Si vous configurez les deux, les règles par locataire s'appliquent avant les règles globales.

1. Pour activer la rétention basée sur les flux, créez ou modifiez la ressource personnalisée (CR) **LokiStack**:

```
oc create -f <nom-du-fichier>.yaml
```

1. Vous pouvez vous référer aux exemples ci-dessous pour configurer votre LokiStack CR.

Exemple de rétention globale basée sur les flux

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  limits:
    global: 1
      retention: 2
        days: 20
        streams:
        - days: 4
          priority: 1
          selector: '{kubernetes_namespace_name=~"test.+"}' 3
        - days: 1
          priority: 1
          selector: '{log_type="infrastructure"}'
  managementState: Managed
  replicationFactor: 1
  size: 1x.small
  storage:
    schemas:
    - effectiveDate: "2020-10-11"
      version: v11
    secret:
      name: logging-loki-s3
      type: aws
  storageClassName: standard
  tenants:
    mode: openshift-logging
```

1. Définit la politique de rétention pour tous les flux de données. **Note: This field does not impact the retention period for stored logs in object storage.**
2. La rétention est activée dans le cluster lorsque ce bloc est ajouté au CR.
3. Contient la requête LogQL utilisée pour définir le flux de données.

Exemple de rétention par locataire basée sur les flux

```
apiVersion: loki.grafana.com/v1
```

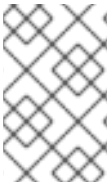
```
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  limits:
    global:
      retention:
        days: 20
    tenants: 1
      application:
        retention:
          days: 1
          streams:
            - days: 4
              selector: '{kubernetes_namespace_name=~"test.+"}' 2
      infrastructure:
        retention:
          days: 5
          streams:
            - days: 1
              selector: '{kubernetes_namespace_name=~"openshift-cluster.+"}'
  managementState: Managed
  replicationFactor: 1
  size: 1x.small
  storage:
    schemas:
    - effectiveDate: "2020-10-11"
      version: v11
    secret:
      name: logging-loki-s3
      type: aws
  storageClassName: standard
  tenants:
    mode: openshift-logging
```

1 Définit la politique de rétention par locataire. Les types de locataires valides sont **application**, **audit**, et **infrastructure**.

2 Contient la requête LogQL utilisée pour définir le flux de données.

1. Appliquez ensuite votre configuration :

```
oc apply -f <nom-du-fichier>.yaml
```



NOTE

Il ne s'agit pas de gérer la rétention des journaux stockés. Les périodes de rétention globales pour les journaux stockés, jusqu'à un maximum de 30 jours, sont configurées avec votre stockage d'objets.

7.4. TRANSFÉRER LES JOURNAUX À LOKISTACK

Pour configurer la transmission des journaux à la passerelle LokiStack, vous devez créer une ressource personnalisée (CR) ClusterLogging.

Conditions préalables

- Sous-système de journalisation pour Red Hat OpenShift : 5.5 et versions ultérieures
- **Loki Operator** Opérateur

Procédure

1. Créer ou modifier un fichier YAML qui définit la ressource personnalisée (CR) **ClusterLogging**:

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging
metadata:
  name: instance
  namespace: openshift-logging
spec:
  managementState: Managed
  logStore:
    type: lokistack
    lokistack:
      name: logging-loki
  collection:
    type: vector
```

7.4.1. Dépannage des erreurs de Loki "entry out of order" (entrée en dehors de l'ordre)

Si votre Fluentd transmet un grand bloc de messages à un système de journalisation Loki qui dépasse la limite de débit, Loki génère des erreurs "entry out of order". Pour résoudre ce problème, vous devez mettre à jour certaines valeurs dans le fichier de configuration du serveur Loki, **loki.yaml**.

NOTE

loki.yaml n'est pas disponible sur les serveurs Loki hébergés par Grafana. Cette rubrique ne s'applique pas aux serveurs Loki hébergés par Grafana.

Conditions

- La ressource personnalisée **ClusterLogForwarder** est configurée pour transmettre les journaux à Loki.
- Votre système envoie à Loki un bloc de messages d'une taille supérieure à 2 Mo, par exemple :

```
"values":[["1630410392689800468","{\"kind\":\"Event\",\"apiVersion\":\
.......
......
......
......
\"received_at\":\"2021-08-31T11:46:32.800278+00:00\",\"version\":\"1.7.4
1.6.0\"}},\"@timestamp\":\"2021-08-
```

> 31T11:46:32.799692+00:00\",\"viaq_index_name\":\"audit-write\",\"viaq_msg_id\":\"MzFjYjJkZjItNjY0MC00YWU4LWIwMTEtNGNmM2E5ZmViMGU4\",\"log_type\":\"audit\"}"]]}]}

- Lorsque vous entrez **oc logs -c fluentd**, les journaux Fluentd dans votre cluster OpenShift Logging affichent les messages suivants :

> 429 Too Many Requests Ingestion rate limit exceeded (limit: 8388608 bytes/sec) while attempting to ingest '2140' lines totaling '3285284' bytes
>
> 429 Too Many Requests Ingestion rate limit exceeded' or '500 Internal Server Error rpc error: code = ResourceExhausted desc = grpc: received message larger than max (5277702 vs. 4194304)'

- Lorsque vous ouvrez les journaux sur le serveur Loki, ils affichent des messages **entry out of order** comme ceux-ci :

> ,\nentry with timestamp 2021-08-18 05:58:55.061936 +0000 UTC ignored, reason: 'entry out of order' for stream:
>
> {fluentd_thread=\"flush_thread_0\", log_type=\"audit\"},\nentry with timestamp 2021-08-18 06:01:18.290229 +0000 UTC ignored, reason: 'entry out of order' for stream: {fluentd_thread="flush_thread_0", log_type="audit"}

Procédure

1. Mettez à jour les champs suivants dans le fichier de configuration **loki.yaml** sur le serveur Loki avec les valeurs indiquées ici :
   - **grpc_server_max_recv_msg_size: 8388608**
   - **chunk_target_size: 8388608**
   - **ingestion_rate_mb: 8**
   - **ingestion_burst_size_mb: 16**
2. Appliquez les changements dans **loki.yaml** au serveur Loki.

Exemple de fichier loki.yaml

```
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096
  grpc_server_max_recv_msg_size: 8388608

ingester:
  wal:
    enabled: true
    dir: /tmp/wal
  lifecycler:
    address: 127.0.0.1
    ring:
```

```
      kvstore:
        store: inmemory
      replication_factor: 1
    final_sleep: 0s
  chunk_idle_period: 1h       # Any chunk not receiving new logs in this time will be flushed
  chunk_target_size: 8388608
  max_chunk_age: 1h           # All chunks will be flushed when they hit this age, default is 1h
  chunk_retain_period: 30s    # Must be greater than index read cache TTL if using an index cache
(Default index read cache TTL is 5m)
  max_transfer_retries: 0     # Chunk transfers disabled

schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

storage_config:
  boltdb_shipper:
    active_index_directory: /tmp/loki/boltdb-shipper-active
    cache_location: /tmp/loki/boltdb-shipper-cache
    cache_ttl: 24h         # Can be increased for faster performance over longer query periods, uses
more disk space
    shared_store: filesystem
  filesystem:
    directory: /tmp/loki/chunks

compactor:
  working_directory: /tmp/loki/boltdb-shipper-compactor
  shared_store: filesystem

limits_config:
  reject_old_samples: true
  reject_old_samples_max_age: 12h
  ingestion_rate_mb: 8
  ingestion_burst_size_mb: 16

chunk_store_config:
  max_look_back_period: 0s

table_manager:
  retention_deletes_enabled: false
  retention_period: 0s

ruler:
  storage:
    type: local
    local:
      directory: /tmp/loki/rules
  rule_path: /tmp/loki/rules-temp
  alertmanager_url: http://localhost:9093
  ring:
```

```
  kvstore:
    store: inmemory
enable_api: true
```

Ressources complémentaires

- Configuration de Loki

7.5. RESSOURCES COMPLÉMENTAIRES

- Documentation sur le langage de requête Loki (LogQL)
- Documentation du tableau de bord Grafana
- Documentation sur le stockage d'objets Loki
- Documentation sur le schéma de stockage de Loki

CHAPITRE 8. CONSULTATION DES JOURNAUX D'UNE RESSOURCE

Vous pouvez consulter les journaux de diverses ressources, telles que les builds, les déploiements et les pods, en utilisant la CLI d'OpenShift (oc) et la console web.

NOTE

Les journaux de ressources sont une fonctionnalité par défaut qui offre une capacité limitée de visualisation des journaux. Pour améliorer l'expérience de récupération et de visualisation des journaux, il est recommandé d'installer OpenShift Logging. Le sous-système de journalisation regroupe tous les journaux de votre cluster OpenShift Container Platform, tels que les journaux d'audit du système de nœuds, les journaux de conteneurs d'application et les journaux d'infrastructure, dans un magasin de journaux dédié. Vous pouvez ensuite interroger, découvrir et visualiser vos données de journalisation via l'interface Kibana. Les journaux de ressources n'accèdent pas au magasin de journaux du sous-système de journalisation.

8.1. VISUALISATION DES JOURNAUX DE RESSOURCES

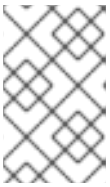
Vous pouvez consulter le journal de diverses ressources dans l'interface de commande OpenShift (oc) et dans la console Web. Les journaux se lisent à partir de la queue, ou de la fin, du journal.

Conditions préalables

- Accès à la CLI d'OpenShift (oc).

Procédure (UI)

1. Dans la console OpenShift Container Platform, naviguez vers **Workloads → Pods** ou naviguez vers le pod via la ressource que vous souhaitez étudier.



NOTE

Certaines ressources, telles que les constructions, n'ont pas de pods à interroger directement. Dans ce cas, vous pouvez trouver le lien **Logs** sur la page **Details** de la ressource.

2. Sélectionnez un projet dans le menu déroulant.
3. Cliquez sur le nom du module que vous souhaitez examiner.
4. Cliquez sur **Logs**.

Procédure (CLI)

- Visualiser le journal d'un pod spécifique :

```
oc logs -f <nom_du_pod> -c <nom_du_conteneur>
```

où :

-f

Facultatif : Spécifie que la sortie suit ce qui est écrit dans les journaux.

<pod_name>

Spécifie le nom du module.

<container_name>

Facultatif : Spécifie le nom d'un conteneur. Lorsqu'un module a plus d'un conteneur, vous devez spécifier le nom du conteneur.

Par exemple :

```
$ oc logs ruby-58cd97df55-mww7r
```

```
$ oc logs -f ruby-57f7f4855b-znl92 -c ruby
```

Le contenu des fichiers journaux est imprimé.

- Consulter le journal d'une ressource spécifique :

```
oc logs <object_type>/<resource_name> 1
```

1 Spécifie le type et le nom de la ressource.

Par exemple :

```
$ oc logs deployment/ruby
```

Le contenu des fichiers journaux est imprimé.

CHAPITRE 9. VISUALISER LES LOGS DES CLUSTERS À L'AIDE DE KIBANA

Le sous-système de journalisation comprend une console web pour visualiser les données de journalisation collectées. Actuellement, OpenShift Container Platform déploie la console Kibana pour la visualisation.

Le visualiseur de journal vous permet d'effectuer les opérations suivantes avec vos données :

- rechercher et parcourir les données à l'aide de l'onglet **Discover**.
- et cartographier les données à l'aide de l'onglet **Visualize**.
- créer et afficher des tableaux de bord personnalisés à l'aide de l'onglet **Dashboard**.

L'utilisation et la configuration de l'interface Kibana sortent du cadre de cette documentation. Pour plus d'informations sur l'utilisation de l'interface, consultez la documentation Kibana.

NOTE

Les journaux d'audit ne sont pas stockés dans l'instance interne d'OpenShift Container Platform Elasticsearch par défaut. Pour afficher les journaux d'audit dans Kibana, vous devez utiliser l'API Log Forwarding pour configurer un pipeline qui utilise la sortie **default** pour les journaux d'audit.

9.1. DÉFINIR LES MODÈLES D'INDEX KIBANA

Un modèle d'index définit les index Elasticsearch que vous souhaitez visualiser. Pour explorer et visualiser des données dans Kibana, vous devez créer un modèle d'index.

Conditions préalables

- Un utilisateur doit avoir le rôle **cluster-admin**, le rôle **cluster-reader** ou les deux rôles pour voir les index **infra** et **audit** dans Kibana. L'utilisateur par défaut **kubeadmin** dispose des autorisations nécessaires pour afficher ces index.
  Si vous pouvez voir les pods et les journaux dans les projets **default**, **kube-** et **openshift-**, vous devriez pouvoir accéder à ces index. Vous pouvez utiliser la commande suivante pour vérifier si l'utilisateur actuel dispose des autorisations appropriées :

```
$ oc auth can-i get pods/log -n <projet>
```

Exemple de sortie

```
yes
```

NOTE

Les journaux d'audit ne sont pas stockés dans l'instance interne d'OpenShift Container Platform Elasticsearch par défaut. Pour afficher les journaux d'audit dans Kibana, vous devez utiliser l'API Log Forwarding pour configurer un pipeline qui utilise la sortie **default** pour les journaux d'audit.

- Les documents Elasticsearch doivent être indexés avant de pouvoir créer des modèles d'index. Cette opération est effectuée automatiquement, mais elle peut prendre quelques minutes dans un cluster nouveau ou mis à jour.

Procédure

Pour définir des modèles d'index et créer des visualisations dans Kibana :

1. Dans la console OpenShift Container Platform, cliquez sur le lanceur d'applications  et sélectionnez **Logging**.

2. Créez vos modèles d'index Kibana en cliquant sur **Management → Index Patterns → Create index pattern**:

   - Chaque utilisateur doit créer manuellement des modèles d'index lors de sa première connexion à Kibana pour voir les journaux de ses projets. Les utilisateurs doivent créer un modèle d'index nommé **app** et utiliser le champ **@timestamp** time pour afficher les journaux de leurs conteneurs.

   - Chaque utilisateur administrateur doit créer des modèles d'index lors de sa première connexion à Kibana pour les index **app**, **infra**, et **audit** en utilisant le champ **@timestamp** time.

3. Créer des visualisations Kibana à partir des nouveaux modèles d'index.

9.2. VISUALISATION DES JOURNAUX DES CLUSTERS DANS KIBANA

Les journaux des clusters sont affichés dans la console web Kibana. Les méthodes d'affichage et de visualisation de vos données dans Kibana dépassent le cadre de cette documentation. Pour plus d'informations, consultez la documentation Kibana.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

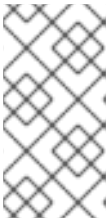
- Les modèles d'index Kibana doivent exister.

- Un utilisateur doit avoir le rôle **cluster-admin**, le rôle **cluster-reader** ou les deux rôles pour voir les index **infra** et **audit** dans Kibana. L'utilisateur par défaut **kubeadmin** dispose des autorisations nécessaires pour afficher ces index.
  Si vous pouvez voir les pods et les journaux dans les projets **default**, **kube-** et **openshift-**, vous devriez pouvoir accéder à ces index. Vous pouvez utiliser la commande suivante pour vérifier si l'utilisateur actuel dispose des autorisations appropriées :

  ```
  $ oc auth can-i get pods/log -n <projet>
  ```

  Exemple de sortie

  ```
  yes
  ```



NOTE

Les journaux d'audit ne sont pas stockés dans l'instance interne d'OpenShift Container Platform Elasticsearch par défaut. Pour afficher les journaux d'audit dans Kibana, vous devez utiliser l'API Log Forwarding pour configurer un pipeline qui utilise la sortie **default** pour les journaux d'audit.

Procédure

Pour afficher les journaux dans Kibana :

1. Dans la console OpenShift Container Platform, cliquez sur le lanceur d'applications  et sélectionnez **Logging**.

2. Connectez-vous en utilisant les mêmes informations d'identification que celles utilisées pour vous connecter à la console OpenShift Container Platform.
   L'interface Kibana est lancée.

3. Dans Kibana, cliquez sur **Discover**.

4. Sélectionnez le modèle d'index que vous avez créé dans le menu déroulant situé dans le coin supérieur gauche : **app**, **audit**, ou **infra**.
   Les données du journal s'affichent sous forme de documents horodatés.

5. Développez l'un des documents horodatés.

6. Cliquez sur l'onglet **JSON** pour afficher l'entrée du journal pour ce document.

Exemple 9.1. Exemple d'entrée de journal d'infrastructure dans Kibana

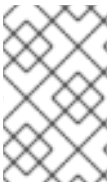

```
{
  "_index": "infra-000001",
  "_type": "_doc",
  "_id": "YmJmYTBlNDkZTRmLTliMGQtMjE3NmFiOGUyOWM3",
  "_version": 1,
  "_score": null,
  "_source": {
    "docker": {
      "container_id": "f85fa55bbef7bb783f041066be1e7c267a6b88c4603dfce213e32c1"
    },
    "kubernetes": {
      "container_name": "registry-server",
      "namespace_name": "openshift-marketplace",
      "pod_name": "redhat-marketplace-n64gc",
      "container_image": "registry.redhat.io/redhat/redhat-marketplace-index:v4.7",
      "container_image_id": "registry.redhat.io/redhat/redhat-marketplace-
index@sha256:65fc0c45aabb95809e376feb065771ecda9e5e59cc8b3024c4545c168f",
      "pod_id": "8f594ea2-c866-4b5c-a1c8-a50756704b2a",
      "host": "ip-10-0-182-28.us-east-2.compute.internal",
      "master_url": "https://kubernetes.default.svc",
      "namespace_id": "3abab127-7669-4eb3-b9ef-44c04ad68d38",
      "namespace_labels": {
        "openshift_io/cluster-monitoring": "true"
      },
      "flat_labels": [
        "catalogsource_operators_coreos_com/update=redhat-marketplace"
```

```
      ]
    },
    "message": "time=\"2020-09-23T20:47:03Z\" level=info msg=\"serving registry\"
database=/database/index.db port=50051",
    "level": "unknown",
    "hostname": "ip-10-0-182-28.internal",
    "pipeline_metadata": {
      "collector": {
        "ipaddr4": "10.0.182.28",
        "inputname": "fluent-plugin-systemd",
        "name": "fluentd",
        "received_at": "2020-09-23T20:47:15.007583+00:00",
        "version": "1.7.4 1.6.0"
      }
    },
    "@timestamp": "2020-09-23T20:47:03.422465+00:00",
    "viaq_msg_id": "YmJmYTBlNDktMDMGQtMjE3NmFiOGUyOWM3",
    "openshift": {
      "labels": {
        "logging": "infra"
      }
    }
  },
  "fields": {
    "@timestamp": [
      "2020-09-23T20:47:03.422Z"
    ],
    "pipeline_metadata.collector.received_at": [
      "2020-09-23T20:47:15.007Z"
    ]
  },
  "sort": [
    1600894023422
  ]
}
```

CHAPITRE 10. TRANSFÉRER LES JOURNAUX VERS DES SYSTÈMES DE JOURNALISATION TIERS EXTERNES

Par défaut, le sous-système de journalisation envoie les journaux des conteneurs et de l'infrastructure au magasin de journaux interne par défaut défini dans la ressource personnalisée **ClusterLogging**. Cependant, il n'envoie pas les journaux d'audit au magasin interne car il ne fournit pas de stockage sécurisé. Si cette configuration par défaut répond à vos besoins, vous n'avez pas besoin de configurer le Cluster Log Forwarder.

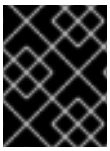
Pour envoyer des logs à d'autres agrégateurs de logs, vous utilisez l'API OpenShift Container Platform Cluster Log Forwarder. Cette API vous permet d'envoyer des journaux de conteneurs, d'infrastructure et d'audit à des points d'extrémité spécifiques à l'intérieur ou à l'extérieur de votre cluster. En outre, vous pouvez envoyer différents types de journaux à divers systèmes afin que différentes personnes puissent accéder à chaque type. Vous pouvez également activer la prise en charge de Transport Layer Security (TLS) pour envoyer les journaux en toute sécurité, selon les besoins de votre organisation.



NOTE

Pour envoyer les journaux d'audit au magasin de journaux Elasticsearch interne par défaut, utilisez le Cluster Log Forwarder comme décrit dans Transférer les journaux d'audit vers le magasin de journaux.

Lorsque vous transmettez des journaux en externe, le sous-système de journalisation crée ou modifie une carte de configuration Fluentd pour envoyer des journaux en utilisant les protocoles que vous souhaitez. Vous êtes responsable de la configuration du protocole sur l'agrégateur de logs externe.



IMPORTANT

Vous ne pouvez pas utiliser les méthodes de carte de configuration et le Cluster Log Forwarder dans le même cluster.

10.1. À PROPOS DE LA TRANSMISSION DES JOURNAUX À DES SYSTÈMES TIERS

Pour envoyer des journaux à des points d'extrémité spécifiques à l'intérieur et à l'extérieur de votre cluster OpenShift Container Platform, vous spécifiez une combinaison de *outputs* et *pipelines* dans une ressource personnalisée (CR) **ClusterLogForwarder**. Vous pouvez également utiliser *inputs* pour transmettre les journaux d'application associés à un projet spécifique à un point de terminaison. L'authentification est fournie par un objet Kubernetes *Secret*.

output

La destination des données d'enregistrement que vous définissez, ou l'endroit où vous souhaitez que les enregistrements soient envoyés. Une sortie peut être de l'un des types suivants :

- **elasticsearch**. Une instance Elasticsearch externe. La sortie **elasticsearch** peut utiliser une connexion TLS.
- **fluentdForward**. Une solution externe d'agrégation de logs qui supporte Fluentd. Cette option utilise les protocoles Fluentd **forward**. La sortie **fluentForward** peut utiliser une connexion TCP ou TLS et prend en charge l'authentification par clé partagée en fournissant un champ **shared_key** dans un secret. L'authentification par clé partagée peut être utilisée avec ou sans TLS.

- **syslog**. Une solution externe d'agrégation de journaux qui prend en charge les protocoles syslog RFC3164 ou RFC5424. La sortie **syslog** peut utiliser une connexion UDP, TCP ou TLS.
- **cloudwatch**. Amazon CloudWatch, un service de surveillance et de stockage de journaux hébergé par Amazon Web Services (AWS).
- **loki**. Loki, un système d'agrégation de logs horizontalement extensible, hautement disponible et multi-tenant.
- **kafka**. Un courtier Kafka. La sortie **kafka** peut utiliser une connexion TCP ou TLS.
- **default**. L'instance interne d'Elasticsearch de OpenShift Container Platform. Vous n'êtes pas obligé de configurer la sortie par défaut. Si vous configurez une sortie **default**, vous recevez un message d'erreur car la sortie **default** est réservée à Red Hat OpenShift Logging Operator.

pipeline

Définit le routage simple d'un type de journal vers une ou plusieurs sorties, ou les journaux que vous souhaitez envoyer. Les types de journaux sont les suivants

- **application**. Journaux des conteneurs générés par les applications utilisateur exécutées dans le cluster, à l'exception des applications de conteneurs d'infrastructure.
- **infrastructure**. Journaux de conteneurs provenant de pods qui s'exécutent dans les projets **openshift***, **kube***, ou **default** et journaux provenant du système de fichiers du nœud.
- **audit**. Journaux d'audit générés par le système d'audit des nœuds, **auditd**, le serveur API Kubernetes, le serveur API OpenShift et le réseau OVN.

Vous pouvez ajouter des étiquettes aux messages de journaux sortants en utilisant les paires **key:value** dans le pipeline. Par exemple, vous pouvez ajouter une étiquette aux messages qui sont transmis à d'autres centres de données ou étiqueter les journaux par type. Les étiquettes ajoutées aux objets sont également transmises avec le message de journal.

input

Transfère les journaux d'application associés à un projet spécifique vers un pipeline.
Dans le pipeline, vous définissez les types de journaux à transférer à l'aide d'un paramètre **inputRef** et l'endroit où transférer les journaux à l'aide d'un paramètre **outputRef**.

Secret

Un site **key:value map** qui contient des données confidentielles telles que des informations d'identification de l'utilisateur.

Il convient de noter ce qui suit :

- Si un objet CR **ClusterLogForwarder** existe, les journaux ne sont pas transmis à l'instance Elasticsearch par défaut, sauf s'il existe un pipeline avec la sortie **default**.
- Par défaut, le sous-système de journalisation envoie les journaux des conteneurs et de l'infrastructure au magasin de journaux Elasticsearch interne par défaut, défini dans la ressource personnalisée **ClusterLogging**. Cependant, il n'envoie pas les journaux d'audit au magasin interne car il ne fournit pas de stockage sécurisé. Si cette configuration par défaut répond à vos besoins, ne configurez pas l'API Log Forwarding.

- Si vous ne définissez pas de pipeline pour un type de journal, les journaux des types non définis sont abandonnés. Par exemple, si vous spécifiez un pipeline pour les types **application** et **audit**, mais que vous ne spécifiez pas de pipeline pour le type **infrastructure**, les journaux **infrastructure** sont abandonnés.

- Vous pouvez utiliser plusieurs types de sorties dans la ressource personnalisée (CR) **ClusterLogForwarder** pour envoyer des journaux à des serveurs qui prennent en charge différents protocoles.

- L'instance interne d'OpenShift Container Platform Elasticsearch ne fournit pas de stockage sécurisé pour les journaux d'audit. Nous vous recommandons de vous assurer que le système vers lequel vous transmettez les journaux d'audit est conforme aux réglementations de votre organisation et de votre gouvernement et qu'il est correctement sécurisé. Le sous-système de journalisation n'est pas conforme à ces réglementations.

L'exemple suivant transfère les journaux d'audit vers une instance Elasticsearch externe sécurisée, les journaux d'infrastructure vers une instance Elasticsearch externe non sécurisée, les journaux d'application vers un courtier Kafka et les journaux d'application du projet **my-apps-logs** vers l'instance Elasticsearch interne.

Exemples de sorties et de pipelines de transfert de logs

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  outputs:
   - name: elasticsearch-secure 3
     type: "elasticsearch"
     url: https://elasticsearch.secure.com:9200
     secret:
        name: elasticsearch
   - name: elasticsearch-insecure 4
     type: "elasticsearch"
     url: http://elasticsearch.insecure.com:9200
   - name: kafka-app 5
     type: "kafka"
     url: tls://kafka.secure.com:9093/app-topic
  inputs: 6
   - name: my-app-logs
     application:
        namespaces:
        - my-project
  pipelines:
   - name: audit-logs 7
     inputRefs:
      - audit
     outputRefs:
      - elasticsearch-secure
      - default
     parse: json 8
     labels:
       secure: "true" 9
```

```
    datacenter: "east"
- name: infrastructure-logs 10
  inputRefs:
   - infrastructure
  outputRefs:
   - elasticsearch-insecure
  labels:
    datacenter: "west"
- name: my-app 11
  inputRefs:
   - my-app-logs
  outputRefs:
   - default
- inputRefs: 12
   - application
  outputRefs:
   - kafka-app
  labels:
    datacenter: "south"
```

1. Le nom du CR **ClusterLogForwarder** doit être **instance**.
2. L'espace de noms pour le CR **ClusterLogForwarder** doit être **openshift-logging**.
3. Configuration pour une sortie sécurisée d'Elasticsearch utilisant un secret avec une URL sécurisée.
   - Un nom pour décrire la sortie.
   - Le type de sortie : **elasticsearch**.
   - L'URL et le port sécurisés de l'instance Elasticsearch sous la forme d'une URL absolue valide, y compris le préfixe.
   - Le secret requis par le point final pour la communication TLS. Le secret doit exister dans le projet **openshift-logging**.
4. Configuration pour une sortie Elasticsearch non sécurisée :
   - Un nom pour décrire la sortie.
   - Le type de sortie : **elasticsearch**.
   - L'URL et le port non sécurisés de l'instance Elasticsearch sous la forme d'une URL absolue valide, y compris le préfixe.
5. Configuration d'une sortie Kafka utilisant une communication TLS authentifiée par le client via une URL sécurisée
   - Un nom pour décrire la sortie.
   - Le type de sortie : **kafka**.
   - Spécifiez l'URL et le port du courtier Kafka sous la forme d'une URL absolue valide, y compris le préfixe.
6. Configuration d'une entrée pour filtrer les journaux d'application de l'espace de noms **my-project**.

7 Configuration d'un pipeline pour l'envoi des journaux d'audit à l'instance Elasticsearch externe sécurisée :

- Un nom pour décrire le pipeline.
- L'adresse **inputRefs** est le type de journal, dans cet exemple **audit**.
- **outputRefs** est le nom de la sortie à utiliser, dans cet exemple **elasticsearch-secure** pour transmettre à l'instance Elasticsearch sécurisée et **default** pour transmettre à l'instance Elasticsearch interne.
- Facultatif : Étiquettes à ajouter aux journaux.

8 Facultatif : Indiquez si les entrées de log JSON structurées doivent être transmises en tant qu'objets JSON dans le champ **structured**. L'entrée de log doit contenir du JSON structuré valide ; sinon, OpenShift Logging supprime le champ **structured** et envoie l'entrée de log à l'index par défaut, **app-00000x**.

9 Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux. Citez des valeurs comme "true" pour qu'elles soient reconnues comme des chaînes de caractères et non comme des booléens.

10 Configuration d'un pipeline pour l'envoi des journaux d'infrastructure à l'instance Elasticsearch externe non sécurisée.

11 Configuration d'un pipeline pour l'envoi des logs du projet **my-project** vers l'instance Elasticsearch interne.

- Un nom pour décrire le pipeline.
- Le site **inputRefs** est une entrée spécifique : **my-app-logs**.
- Le site **outputRefs** est **default**.
- Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux.

12 Configuration d'un pipeline pour envoyer des logs au courtier Kafka, sans nom de pipeline :

- L'adresse **inputRefs** est le type de journal, dans cet exemple **application**.
- **outputRefs** est le nom de la sortie à utiliser.
- Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux.

Gestion des journaux Fluentd lorsque l'agrégateur de journaux externe n'est pas disponible

Si votre agrégateur de logs externe devient indisponible et ne peut pas recevoir les logs, Fluentd continue à collecter les logs et les stocke dans un tampon. Lorsque l'agrégateur de logs devient disponible, la transmission des logs reprend, y compris les logs mis en mémoire tampon. Si le tampon se remplit complètement, Fluentd arrête de collecter les logs. OpenShift Container Platform fait tourner les logs et les supprime. Vous ne pouvez pas ajuster la taille du tampon ou ajouter une revendication de volume persistant (PVC) à l'ensemble de démon Fluentd ou aux pods.

Clés d'autorisation prises en charge

Les types de clés les plus courants sont fournis ici. Certains types de sortie prennent en charge des clés spécialisées supplémentaires, documentées par le champ de configuration spécifique à la sortie. Toutes les clés secrètes sont facultatives. Activez les fonctions de sécurité que vous souhaitez en définissant

les clés correspondantes. Vous êtes responsable de la création et de la maintenance de toutes les configurations supplémentaires que les destinations externes pourraient exiger, telles que les clés et les secrets, les comptes de service, les ouvertures de port ou la configuration globale du proxy. Open Shift Logging n'essaiera pas de vérifier une incompatibilité entre les combinaisons d'autorisation.

Sécurité de la couche transport (TLS)

L'utilisation d'une URL TLS ('http://...' ou 'ssl://...') sans secret permet une authentification TLS de base côté serveur. Des fonctions TLS supplémentaires sont activées en incluant un secret et en définissant les champs facultatifs suivants :

- **tls.crt**: (chaîne) Nom de fichier contenant un certificat client. Permet l'authentification mutuelle. Nécessite **tls.key**.
- **tls.key**(chaîne) Nom de fichier contenant la clé privée permettant de déverrouiller le certificat du client. Nécessite **tls.crt**.
- **passphrase**: (chaîne) Phrase de passe pour décoder une clé privée TLS encodée. Nécessite **tls.key**.
- **ca-bundle.crt**(chaîne) Nom de fichier d'une autorité de certification cliente pour l'authentification du serveur.

Nom d'utilisateur et mot de passe

- **username**: (chaîne) Nom d'utilisateur de l'authentification. Requiert **password**.
- **password**: (chaîne) Mot de passe d'authentification. Requiert **username**.

Couche de sécurité d'authentification simple (SASL)

- **sasl.enable** (booléen) Active ou désactive explicitement SASL. S'il est absent, le SASL est automatiquement activé lorsque l'une des autres clés **sasl.** est activée.
- **sasl.mechanisms**(tableau) Liste des noms de mécanismes SASL autorisés. S'ils sont absents ou vides, les valeurs par défaut du système sont utilisées.
- **sasl.allow-insecure**: (booléen) Autorise les mécanismes qui envoient des mots de passe en texte clair. La valeur par défaut est false.

10.1.1. Création d'un secret

Vous pouvez créer un secret dans le répertoire qui contient vos fichiers de certificats et de clés à l'aide de la commande suivante :

```
$ oc create secret generic -n openshift-logging <my-secret> \
 --from-file=tls.key=<your_key_file>
 --from-file=tls.crt=<your_crt_file>
 --from-file=ca-bundle.crt=<your_bundle_file>
 --from-literal=username=<your_username>
 --from-literal=password=<your_password>
```

NOTE

Les secrets génériques ou opaques sont recommandés pour de meilleurs résultats.

10.2. TRANSFÉRER LES LOGS JSON DES CONTENEURS D'UN MÊME POD VERS DES INDEX DISTINCTS

Vous pouvez transmettre des journaux structurés provenant de différents conteneurs au sein d'un même module à différents index. Pour utiliser cette fonctionnalité, vous devez configurer le pipeline avec la prise en charge de plusieurs conteneurs et annoter les modules. Les journaux sont écrits dans les index avec un préfixe de **app-**. Il est recommandé de configurer Elasticsearch avec des alias pour s'adapter à cela.

IMPORTANT

Le formatage JSON des journaux varie selon les applications. La création d'un trop grand nombre d'index ayant un impact sur les performances, limitez l'utilisation de cette fonctionnalité à la création d'index pour les journaux dont les formats JSON sont incompatibles. Utilisez des requêtes pour séparer les journaux provenant de différents espaces de noms ou d'applications dont les formats JSON sont compatibles.

Conditions préalables

- Sous-système de journalisation pour Red Hat OpenShift : 5.5

Procédure

1. Créez ou modifiez un fichier YAML qui définit l'objet **ClusterLogForwarder** CR :

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance
  namespace: openshift-logging
spec:
  outputDefaults:
    elasticsearch:
      enableStructuredContainerLogs: true (1)
  pipelines:
  - inputRefs:
    - application
    name: application-logs
    outputRefs:
    - default
    parse: json
```

(1) Active les sorties multi-conteneurs.

2. Créez ou modifiez un fichier YAML qui définit l'objet **Pod** CR :

```
  apiVersion: v1
  kind: Pod
  metadata:
    annotations:
      containerType.logging.openshift.io/heavy: heavy (1)
      containerType.logging.openshift.io/low: low
  spec:
```

```
containers:
- name: heavy 2
  image: heavyimage
- name: low
  image: lowimage
```

1 Format : **containerType.logging.openshift.io/<container-name>: <index>**

2 Les noms des annotations doivent correspondre aux noms des conteneurs

AVERTISSEMENT

Cette configuration peut augmenter de manière significative le nombre de shards sur le cluster.

Ressources complémentaires

- Annotations Kubernetes

10.3. TYPES DE SORTIE DE DONNÉES DE LOGS PRIS EN CHARGE DANS OPENSHIFT LOGGING 5.1

Red Hat OpenShift Logging 5.1 fournit les types de sortie et les protocoles suivants pour envoyer des données de journal aux collecteurs de journaux cibles.

Red Hat teste chacune des combinaisons présentées dans le tableau suivant. Cependant, vous devriez être en mesure d'envoyer des données de log à un plus grand nombre de collecteurs de logs cibles qui ingèrent ces protocoles.

Types de sorties	Protocoles	Testé avec
elasticsearch	elasticsearch	Elasticsearch 6.8.1 Elasticsearch 6.8.4 Elasticsearch 7.12.2
fluentdForward	fluentd forward v1	fluentd 1.7.4 logstash 7.10.1
kafka	kafka 0.11	kafka 2.4.1 kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

NOTE

Auparavant, la sortie syslog ne prenait en charge que la norme RFC-3164. La sortie syslog actuelle ajoute la prise en charge de RFC-5424.

10.4. TYPES DE SORTIE DE DONNÉES DE LOGS PRIS EN CHARGE DANS OPENSHIFT LOGGING 5.2

Red Hat OpenShift Logging 5.2 fournit les types de sortie et les protocoles suivants pour envoyer des données de journal aux collecteurs de journaux cibles.

Red Hat teste chacune des combinaisons présentées dans le tableau suivant. Cependant, vous devriez être en mesure d'envoyer des données de log à un plus grand nombre de collecteurs de logs cibles qui ingèrent ces protocoles.

Types de sorties	Protocoles	Testé avec
Amazon CloudWatch	REST sur HTTPS	La version actuelle d'Amazon CloudWatch
elasticsearch	elasticsearch	Elasticsearch 6.8.1 Elasticsearch 6.8.4 Elasticsearch 7.12.2
fluentdForward	fluentd forward v1	fluentd 1.7.4 logstash 7.10.1
Loki	REST sur HTTP et HTTPS	Loki 2.3.0 déployé sur les laboratoires OCP et Grafana
kafka	kafka 0.11	kafka 2.4.1 kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

10.5. TYPES DE SORTIE DE DONNÉES DE LOGS PRIS EN CHARGE DANS OPENSHIFT LOGGING 5.3

Red Hat OpenShift Logging 5.3 fournit les types de sortie et les protocoles suivants pour envoyer des données de journal aux collecteurs de journaux cibles.

Red Hat teste chacune des combinaisons présentées dans le tableau suivant. Cependant, vous devriez être en mesure d'envoyer des données de log à un plus grand nombre de collecteurs de logs cibles qui ingèrent ces protocoles.

Types de sorties	Protocoles	Testé avec
Amazon CloudWatch	REST sur HTTPS	La version actuelle d'Amazon CloudWatch
elasticsearch	elasticsearch	Elasticsearch 7.10.1
fluentdForward	fluentd forward v1	fluentd 1.7.4 logstash 7.10.1
Loki	REST sur HTTP et HTTPS	Loki 2.2.1 déployé sur l'OCP
kafka	kafka 0.11	kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

10.6. TYPES DE SORTIE DE DONNÉES DE LOGS PRIS EN CHARGE DANS OPENSHIFT LOGGING 5.4

Red Hat OpenShift Logging 5.4 fournit les types de sortie et les protocoles suivants pour envoyer des données de journal aux collecteurs de journaux cibles.

Red Hat teste chacune des combinaisons présentées dans le tableau suivant. Cependant, vous devriez être en mesure d'envoyer des données de log à un plus grand nombre de collecteurs de logs cibles qui ingèrent ces protocoles.

Types de sorties	Protocoles	Testé avec
Amazon CloudWatch	REST sur HTTPS	La version actuelle d'Amazon CloudWatch
elasticsearch	elasticsearch	Elasticsearch 7.10.1
fluentdForward	fluentd forward v1	fluentd 1.14.5 logstash 7.10.1
Loki	REST sur HTTP et HTTPS	Loki 2.2.1 déployé sur l'OCP
kafka	kafka 0.11	kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

10.7. TYPES DE SORTIE DE DONNÉES DE LOGS PRIS EN CHARGE DANS OPENSHIFT LOGGING 5.5

Red Hat OpenShift Logging 5.5 fournit les types de sortie et les protocoles suivants pour envoyer des données de journal aux collecteurs de journaux cibles.

Red Hat teste chacune des combinaisons présentées dans le tableau suivant. Cependant, vous devriez être en mesure d'envoyer des données de log à un plus grand nombre de collecteurs de logs cibles qui ingèrent ces protocoles.

Types de sorties	Protocoles	Testé avec
Amazon CloudWatch	REST sur HTTPS	La version actuelle d'Amazon CloudWatch
elasticsearch	elasticsearch	Elasticsearch 7.10.1
fluentdForward	fluentd forward v1	fluentd 1.14.6 logstash 7.10.1
Loki	REST sur HTTP et HTTPS	Loki 2.5.0 déployé sur l'OCP
kafka	kafka 0.11	kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

10.8. TYPES DE SORTIE DE DONNÉES DE LOGS PRIS EN CHARGE DANS OPENSHIFT LOGGING 5.6

Red Hat OpenShift Logging 5.6 fournit les types de sortie et les protocoles suivants pour envoyer des données de journal aux collecteurs de journaux cibles.

Red Hat teste chacune des combinaisons présentées dans le tableau suivant. Cependant, vous devriez être en mesure d'envoyer des données de log à un plus grand nombre de collecteurs de logs cibles qui ingèrent ces protocoles.

Types de sorties	Protocoles	Testé avec
Amazon CloudWatch	REST sur HTTPS	La version actuelle d'Amazon CloudWatch
elasticsearch	elasticsearch	Elasticsearch 6.8.23 Elasticsearch 7.10.1 Elasticsearch 8.6.1
fluentdForward	fluentd forward v1	fluentd 1.14.6 logstash 7.10.1
Loki	REST sur HTTP et HTTPS	Loki 2.5.0 déployé sur l'OCP

Types de sorties	Protocoles	Testé avec
kafka	kafka 0.11	kafka 2.7.0
syslog	RFC-3164, RFC-5424	rsyslog-8.39.0

IMPORTANT

Fluentd ne supporte pas Elasticsearch 8 à partir de la version 5.6.2. Vector ne supporte pas fluentd/logstash/rsyslog avant la version 5.7.0.

10.9. TRANSFÉRER LES JOURNAUX VERS UNE INSTANCE ELASTICSEARCH EXTERNE

Vous pouvez optionnellement transmettre les logs à une instance Elasticsearch externe en plus ou à la place de l'instance Elasticsearch interne d'OpenShift Container Platform. Vous êtes responsable de la configuration de l'agrégateur de logs externe pour recevoir les données de logs d'OpenShift Container Platform.

Pour configurer la redirection des journaux vers une instance Elasticsearch externe, vous devez créer une ressource personnalisée (CR) **ClusterLogForwarder** avec une sortie vers cette instance et un pipeline qui utilise la sortie. La sortie Elasticsearch externe peut utiliser la connexion HTTP (non sécurisée) ou HTTPS (HTTP sécurisée).

Pour transmettre les journaux à une instance Elasticsearch externe et interne, créez des sorties et des pipelines vers l'instance externe et un pipeline qui utilise la sortie **default** pour transmettre les journaux à l'instance interne. Il n'est pas nécessaire de créer une sortie **default**. Si vous configurez une sortie **default**, vous recevrez un message d'erreur car la sortie **default** est réservée à Red Hat OpenShift Logging Operator.

NOTE

Si vous souhaitez transmettre les logs à **only** l'instance Elasticsearch interne d'OpenShift Container Platform, vous n'avez pas besoin de créer un CR **ClusterLogForwarder**.

Conditions préalables

- Vous devez disposer d'un serveur de journalisation configuré pour recevoir les données de journalisation à l'aide du protocole ou du format spécifié.

Procédure

1. Créez ou modifiez un fichier YAML qui définit l'objet **ClusterLogForwarder** CR :

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  outputs:
   - name: elasticsearch-insecure 3
```

```
    type: "elasticsearch" 4
    url: http://elasticsearch.insecure.com:9200 5
  - name: elasticsearch-secure
    type: "elasticsearch"
    url: https://elasticsearch.secure.com:9200 6
    secret:
      name: es-secret 7
pipelines:
 - name: application-logs 8
   inputRefs: 9
   - application
   - audit
   outputRefs:
   - elasticsearch-secure 10
   - default 11
   parse: json 12
   labels:
     myLabel: "myValue" 13
 - name: infrastructure-audit-logs 14
   inputRefs:
   - infrastructure
   outputRefs:
   - elasticsearch-insecure
   labels:
     logs: "audit-infra"
```

1. Le nom du CR **ClusterLogForwarder** doit être **instance**.
2. L'espace de noms pour le CR **ClusterLogForwarder** doit être **openshift-logging**.
3. Spécifiez un nom pour la sortie.
4. Spécifiez le type de **elasticsearch**.
5. Spécifiez l'URL et le port de l'instance Elasticsearch externe sous la forme d'une URL absolue valide. Vous pouvez utiliser le protocole **http** (non sécurisé) ou **https** (HTTP sécurisé). Si le proxy à l'échelle du cluster utilisant l'annotation CIDR est activé, la sortie doit être un nom de serveur ou un FQDN, et non une adresse IP.
6. Pour une connexion sécurisée, vous pouvez spécifier une URL **https** ou **http** que vous authentifiez en spécifiant un **secret**.
7. Pour un préfixe **https**, indiquez le nom du secret requis par le point d'extrémité pour la communication TLS. Le secret doit exister dans le projet **openshift-logging** et doit avoir des clés de : **tls.crt**, **tls.key**, et **ca-bundle.crt** qui pointent vers les certificats respectifs qu'elles représentent. Sinon, pour les préfixes **http** et **https**, vous pouvez spécifier un secret contenant un nom d'utilisateur et un mot de passe. Pour plus d'informations, voir l'exemple suivant : "Exemple : Définition d'un secret contenant un nom d'utilisateur et un mot de passe.\N-"
8. Facultatif : Spécifiez un nom pour le pipeline.
9. Spécifiez les types de journaux à transférer en utilisant le pipeline : **application,** **infrastructure** ou **audit**.

10 Spécifiez le nom de la sortie à utiliser lors du transfert des journaux avec ce pipeline.

11 Facultatif : Spécifiez la sortie **default** pour envoyer les journaux à l'instance Elasticsearch interne.

12 Facultatif : Indiquez si les entrées de log JSON structurées doivent être transmises en tant qu'objets JSON dans le champ **structured**. L'entrée de log doit contenir du JSON structuré valide ; sinon, OpenShift Logging supprime le champ **structured** et envoie l'entrée de log à l'index par défaut, **app-00000x**.

13 Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux.

14 Facultatif : Configurez plusieurs sorties pour transmettre les journaux à d'autres agrégateurs de journaux externes, quel que soit le type pris en charge :

- Un nom pour décrire le pipeline.
- L'adresse **inputRefs** est le type de journal à transmettre en utilisant le pipeline : **application, infrastructure**, ou **audit**.
- **outputRefs** est le nom de la sortie à utiliser.
- Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux.

2. Créer l'objet CR :

```
oc create -f <nom-de-fichier>.yaml
```

Exemple : Définition d'un secret contenant un nom d'utilisateur et un mot de passe

Vous pouvez utiliser un secret contenant un nom d'utilisateur et un mot de passe pour authentifier une connexion sécurisée à une instance Elasticsearch externe.

Par exemple, si vous ne pouvez pas utiliser de clés TLS mutuelles (mTLS) parce qu'un tiers exploite l'instance Elasticsearch, vous pouvez utiliser HTTP ou HTTPS et définir un secret contenant le nom d'utilisateur et le mot de passe.

1. Créez un fichier YAML **Secret** similaire à l'exemple suivant. Utilisez des valeurs encodées en base64 pour les champs **username** et **password**. Le type de secret est opaque par défaut.

```
apiVersion: v1
kind: Secret
metadata:
  name: openshift-test-secret
data:
  username: dGVzdHVzZXJuYW1lCg==
  password: dGVzdHBhc3N3b3JkCg==
```

2. Créer le secret :

```
$ oc create secret -n openshift-logging openshift-test-secret.yaml
```

3. Spécifiez le nom du secret dans le CR **ClusterLogForwarder**:

```
kind: ClusterLogForwarder
```

```
metadata:
  name: instance
  namespace: openshift-logging
spec:
  outputs:
   - name: elasticsearch
     type: "elasticsearch"
     url: https://elasticsearch.secure.com:9200
     secret:
       name: openshift-test-secret
```

NOTE

Dans la valeur du champ **url**, le préfixe peut être **http** ou **https**.

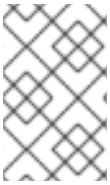
4. Créer l'objet CR :

```
oc create -f <nom-de-fichier>.yaml
```

10.10. TRANSFÉRER LES JOURNAUX EN UTILISANT LE PROTOCOLE DE TRANSFERT FLUENTD

Vous pouvez utiliser le protocole Fluentd **forward** pour envoyer une copie de vos logs à un agrégateur de logs externe qui est configuré pour accepter le protocole à la place ou en plus du magasin de logs Elasticsearch par défaut. Vous êtes responsable de la configuration de l'agrégateur de logs externe pour recevoir les logs d'OpenShift Container Platform.

Pour configurer la transmission des journaux à l'aide du protocole **forward**, vous devez créer une ressource personnalisée (CR) **ClusterLogForwarder** avec une ou plusieurs sorties vers les serveurs Fluentd, et des pipelines qui utilisent ces sorties. La sortie Fluentd peut utiliser une connexion TCP (non sécurisée) ou TLS (TCP sécurisée).



NOTE

Vous pouvez également utiliser une carte de configuration pour transmettre les journaux à l'aide des protocoles **forward**. Cependant, cette méthode est obsolète dans OpenShift Container Platform et sera supprimée dans une prochaine version.

Conditions préalables

- Vous devez disposer d'un serveur de journalisation configuré pour recevoir les données de journalisation à l'aide du protocole ou du format spécifié.

Procédure

1. Créez ou modifiez un fichier YAML qui définit l'objet **ClusterLogForwarder** CR :

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance (1)
  namespace: openshift-logging (2)
```

```
spec:
  outputs:
   - name: fluentd-server-secure 3
     type: fluentdForward 4
     url: 'tls://fluentdserver.security.example.com:24224' 5
     secret: 6
        name: fluentd-secret
   - name: fluentd-server-insecure
     type: fluentdForward
     url: 'tcp://fluentdserver.home.example.com:24224'
  pipelines:
   - name: forward-to-fluentd-secure 7
     inputRefs: 8
     - application
     - audit
     outputRefs:
     - fluentd-server-secure 9
     - default 10
     parse: json 11
     labels:
       clusterId: "C1234" 12
   - name: forward-to-fluentd-insecure 13
     inputRefs:
     - infrastructure
     outputRefs:
     - fluentd-server-insecure
     labels:
       clusterId: "C1234"
```

1. Le nom du CR **ClusterLogForwarder** doit être **instance**.
2. L'espace de noms pour le CR **ClusterLogForwarder** doit être **openshift-logging**.
3. Spécifiez un nom pour la sortie.
4. Spécifiez le type de **fluentdForward**.
5. Spécifiez l'URL et le port de l'instance externe de Fluentd en tant qu'URL absolue valide. Vous pouvez utiliser le protocole **tcp** (non sécurisé) ou **tls** (TCP sécurisé). Si le proxy à l'échelle du cluster utilisant l'annotation CIDR est activé, la sortie doit être un nom de serveur ou un FQDN, et non une adresse IP.
6. Si vous utilisez un préfixe **tls**, vous devez spécifier le nom du secret requis par le point final pour la communication TLS. Le secret doit exister dans le projet **openshift-logging** et doit avoir des clés de : **tls.crt**, **tls.key**, et **ca-bundle.crt** qui pointent vers les certificats respectifs qu'elles représentent. Sinon, pour les préfixes http et https, vous pouvez spécifier un secret contenant un nom d'utilisateur et un mot de passe. Pour plus d'informations, voir l'exemple suivant : "Exemple : Définition d'un secret contenant un nom d'utilisateur et un mot de passe.\N-"
7. Facultatif : Spécifiez un nom pour le pipeline.
8. Spécifiez les types de journaux à transférer en utilisant le pipeline : **application, infrastructure** ou **audit**.

9 Spécifiez le nom de la sortie à utiliser lors du transfert des journaux avec ce pipeline.

10 Facultatif : Spécifiez la sortie **default** pour transmettre les journaux à l'instance Elasticsearch interne.

11 Facultatif : Indiquez si les entrées de log JSON structurées doivent être transmises en tant qu'objets JSON dans le champ **structured**. L'entrée de log doit contenir du JSON structuré valide ; sinon, OpenShift Logging supprime le champ **structured** et envoie l'entrée de log à l'index par défaut, **app-00000x**.

12 Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux.

13 Facultatif : Configurez plusieurs sorties pour transmettre les journaux à d'autres agrégateurs de journaux externes, quel que soit le type pris en charge :

- Un nom pour décrire le pipeline.
- L'adresse **inputRefs** est le type de journal à transmettre en utilisant le pipeline : **application, infrastructure**, ou **audit**.
- **outputRefs** est le nom de la sortie à utiliser.
- Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux.

2. Créer l'objet CR :

```
oc create -f <nom-de-fichier>.yaml
```

10.10.1. Permettre une précision de l'ordre de la nanoseconde pour Logstash afin d'ingérer les données de fluentd

Pour que Logstash puisse ingérer les données de fluentd, vous devez activer la précision de la nanoseconde dans le fichier de configuration de Logstash.

Procédure

- Dans le fichier de configuration de Logstash, définissez **nanosecond_precision** comme **true**.

Exemple de fichier de configuration Logstash

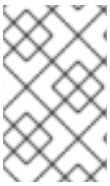
```
input { tcp { codec => fluent { nanosecond_precision => true } port => 24114 } }
filter { }
output { stdout { codec => rubydebug } }
```

10.11. TRANSMISSION DES JOURNAUX À L'AIDE DU PROTOCOLE SYSLOG

Vous pouvez utiliser le protocole **syslog** RFC3164 ou RFC5424 pour envoyer une copie de vos journaux à un agrégateur de journaux externe configuré pour accepter le protocole à la place ou en plus du magasin de journaux Elasticsearch par défaut. Vous êtes responsable de la configuration de l'agrégateur de journaux externe, tel qu'un serveur syslog, pour recevoir les journaux d'OpenShift Container Platform.

Pour configurer la transmission des journaux à l'aide du protocole **syslog**, vous devez créer une

ressource personnalisée (CR) **ClusterLogForwarder** avec une ou plusieurs sorties vers les serveurs syslog, et des pipelines qui utilisent ces sorties. La sortie syslog peut utiliser une connexion UDP, TCP ou TLS.



NOTE

Vous pouvez également utiliser une carte de configuration pour transmettre les journaux en utilisant les protocoles **syslog** RFC3164. Cependant, cette méthode est obsolète dans OpenShift Container Platform et sera supprimée dans une prochaine version.

Conditions préalables

- Vous devez disposer d'un serveur de journalisation configuré pour recevoir les données de journalisation à l'aide du protocole ou du format spécifié.

Procédure

1. Créez ou modifiez un fichier YAML qui définit l'objet **ClusterLogForwarder** CR :

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  outputs:
   - name: rsyslog-east 3
     type: syslog 4
     syslog: 5
       facility: local0
       rfc: RFC3164
       payloadKey: message
       severity: informational
     url: 'tls://rsyslogserver.east.example.com:514' 6
     secret: 7
       name: syslog-secret
   - name: rsyslog-west
     type: syslog
     syslog:
      appName: myapp
      facility: user
      msgID: mymsg
      procID: myproc
      rfc: RFC5424
      severity: debug
     url: 'udp://rsyslogserver.west.example.com:514'
  pipelines:
   - name: syslog-east 8
     inputRefs: 9
     - audit
     - application
     outputRefs: 10
     - rsyslog-east
     - default 11
```

```
    parse: json 12
    labels:
      secure: "true" 13
      syslog: "east"
  - name: syslog-west 14
    inputRefs:
    - infrastructure
    outputRefs:
    - rsyslog-west
    - default
    labels:
      syslog: "west"
```

1. Le nom du CR **ClusterLogForwarder** doit être **instance**.
2. L'espace de noms pour le CR **ClusterLogForwarder** doit être **openshift-logging**.
3. Spécifiez un nom pour la sortie.
4. Spécifiez le type de **syslog**.
5. En option : Spécifiez les paramètres syslog, énumérés ci-dessous.
6. Spécifiez l'URL et le port de l'instance syslog externe. Vous pouvez utiliser le protocole **udp** (non sécurisé), **tcp** (non sécurisé) ou **tls** (TCP sécurisé). Si le proxy à l'échelle du cluster utilisant l'annotation CIDR est activé, la sortie doit être un nom de serveur ou un FQDN, et non une adresse IP.
7. Si vous utilisez un préfixe **tls**, vous devez spécifier le nom du secret requis par le point de terminaison pour la communication TLS. Le secret doit exister dans le projet **openshift-logging** et doit avoir des clés de : **tls.crt**, **tls.key**, et **ca-bundle.crt** qui pointent vers les certificats respectifs qu'elles représentent.
8. Facultatif : Spécifiez un nom pour le pipeline.
9. Spécifiez les types de journaux à transférer en utilisant le pipeline : **application, infrastructure** ou **audit**.
10. Spécifiez le nom de la sortie à utiliser lors du transfert des journaux avec ce pipeline.
11. Facultatif : Spécifiez la sortie **default** pour transmettre les journaux à l'instance Elasticsearch interne.
12. Facultatif : Indiquez si les entrées de log JSON structurées doivent être transmises en tant qu'objets JSON dans le champ **structured**. L'entrée de log doit contenir du JSON structuré valide ; sinon, OpenShift Logging supprime le champ **structured** et envoie l'entrée de log à l'index par défaut, **app-00000x**.
13. Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux. Citez des valeurs comme "true" pour qu'elles soient reconnues comme des chaînes de caractères et non comme des booléens.
14. Facultatif : Configurez plusieurs sorties pour transmettre les journaux à d'autres agrégateurs de journaux externes, quel que soit le type pris en charge :

    - Un nom pour décrire le pipeline.

- L'adresse **inputRefs** est le type de journal à transmettre en utilisant le pipeline : **application, infrastructure**, ou **audit**.
- **outputRefs** est le nom de la sortie à utiliser.
- Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux.

2. Créer l'objet CR :

```
oc create -f <nom-de-fichier>.yaml
```

10.11.1. Ajout d'informations sur la source du journal à la sortie du message

Vous pouvez ajouter les éléments **namespace_name**, **pod_name** et **container_name** au champ **message** de l'enregistrement en ajoutant le champ **AddLogSource** à votre ressource personnalisée (CR) **ClusterLogForwarder**.

```
spec:
  outputs:
  - name: syslogout
    syslog:
      addLogSource: true
      facility: user
      payloadKey: message
      rfc: RFC3164
      severity: debug
      tag: mytag
    type: syslog
    url: tls://syslog-receiver.openshift-logging.svc:24224
  pipelines:
  - inputRefs:
    - application
    name: test-app
    outputRefs:
    - syslogout
```

NOTE

Cette configuration est compatible avec les normes RFC3164 et RFC5424.

Exemple de message syslog sans AddLogSource

```
<15>1 2020-11-15T17:06:14+00:00 fluentd-9hkb4 mytag - - - {"msgcontent"=>"Message Contents",
"timestamp"=>"2020-11-15 17:06:09", "tag_key"=>"rec_tag", "index"=>56}
```

Exemple de sortie de message syslog avec AddLogSource

```
<15>1 2020-11-16T10:49:37+00:00 crc-j55b9-master-0 mytag - - - namespace_name=clo-test-
6327,pod_name=log-generator-ff9746c49-qxm7l,container_name=log-generator,message=
{"msgcontent":"My life is my message", "timestamp":"2020-11-16 10:49:36", "tag_key":"rec_tag",
"index":76}
```

10.11.2. Paramètres Syslog

Vous pouvez configurer les éléments suivants pour les sorties de **syslog**. Pour plus d'informations, voir le RFC syslog RFC3164 ou RFC5424.

- facility : L'installation syslog. La valeur peut être un entier décimal ou un mot-clé insensible à la casse :
  - **0** ou **kern** pour les messages du noyau
  - **1** ou **user** pour les messages de niveau utilisateur, la valeur par défaut.
  - **2** ou **mail** pour le système de messagerie
  - **3** ou **daemon** pour les démons du système
  - **4** ou **auth** pour les messages de sécurité/authentification
  - **5** ou **syslog** pour les messages générés en interne par syslogd
  - **6** ou **lpr** pour le sous-système d'impression de ligne
  - **7** ou **news** pour le sous-système d'information en réseau
  - **8** ou **uucp** pour le sous-système UUCP
  - **9** ou **cron** pour le démon de l'horloge
  - **10** ou **authpriv** pour les messages d'authentification de sécurité
  - **11** ou **ftp** pour le démon FTP
  - **12** ou **ntp** pour le sous-système NTP
  - **13** ou **security** pour le journal d'audit syslog
  - **14** ou **console** pour le journal d'alerte syslog
  - **15** ou **solaris-cron** pour le démon de planification
  - **16**-**23** ou **local0** - **local7** pour les installations utilisées localement
- Facultatif : **payloadKey**: Le champ d'enregistrement à utiliser comme charge utile pour le message syslog.

NOTE

La configuration du paramètre **payloadKey** empêche les autres paramètres d'être transmis au syslog.

- rfc : Le RFC à utiliser pour l'envoi de journaux via syslog. La valeur par défaut est RFC5424.
- severity (gravité) : La gravité syslog à définir sur les enregistrements syslog sortants. La valeur peut être un entier décimal ou un mot-clé insensible à la casse :
  - **0** ou **Emergency** pour les messages indiquant que le système est inutilisable

- **1** ou **Alert** pour les messages indiquant qu'une action doit être entreprise immédiatement
- **2** ou **Critical** pour les messages indiquant des conditions critiques
- **3** ou **Error** pour les messages indiquant des conditions d'erreur
- **4** ou **Warning** pour les messages indiquant des conditions d'alerte
- **5** ou **Notice** pour les messages indiquant des conditions normales mais significatives
- **6** ou **Informational** pour les messages d'information
- **7** ou **Debug** pour les messages indiquant des messages de niveau débogage, la valeur par défaut est

- tag : Tag spécifie un champ d'enregistrement à utiliser comme tag sur le message syslog.
- trimPrefix : Supprime le préfixe spécifié de la balise.

10.11.3. Paramètres syslog supplémentaires RFC5424

Les paramètres suivants s'appliquent à RFC5424 :

- appName : APP-NAME est une chaîne de texte libre qui identifie l'application qui a envoyé le journal. Doit être spécifié pour **RFC5424**.
- msgID : Le MSGID est une chaîne de texte libre qui identifie le type de message. Doit être spécifié pour **RFC5424**.
- procID : Le PROCID est une chaîne de texte libre. Un changement de valeur indique une discontinuité dans les rapports syslog. Doit être spécifié pour **RFC5424**.

10.12. TRANSFÉRER LES JOURNAUX VERS AMAZON CLOUDWATCH

Vous pouvez transmettre les journaux à Amazon CloudWatch, un service de surveillance et de stockage de journaux hébergé par Amazon Web Services (AWS). Vous pouvez transmettre les journaux à CloudWatch en plus ou à la place du magasin de journaux par défaut.

Pour configurer la transmission des journaux à CloudWatch, vous devez créer une ressource personnalisée (CR) **ClusterLogForwarder** avec une sortie pour CloudWatch et un pipeline qui utilise la sortie.

Procédure

1. Créez un fichier YAML **Secret** qui utilise les champs **aws_access_key_id** et **aws_secret_access_key** pour spécifier vos informations d'identification AWS encodées en base64. Par exemple :

```
apiVersion: v1
kind: Secret
metadata:
  name: cw-secret
  namespace: openshift-logging
data:
```

```
  aws_access_key_id: QUtJQUlPU0ZPRE5ON0VYQU1QTEUK
  aws_secret_access_key:
d0phbHJYVXRuRkVNSS9LN01ERU5HL2JQeFJmaUNZRVhBTVBMRUtFWQo=
```

2. Créez le secret. Par exemple :

```
$ oc apply -f cw-secret.yaml
```

3. Créez ou modifiez un fichier YAML qui définit l'objet **ClusterLogForwarder** CR. Dans le fichier, indiquez le nom du secret. Par exemple :

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  outputs:
   - name: cw 3
     type: cloudwatch 4
     cloudwatch:
       groupBy: logType 5
       groupPrefix: <group prefix> 6
       region: us-east-2 7
     secret:
        name: cw-secret 8
  pipelines:
    - name: infra-logs 9
      inputRefs: 10
        - infrastructure
        - audit
        - application
      outputRefs:
        - cw 11
```

1. Le nom du CR **ClusterLogForwarder** doit être **instance**.
2. L'espace de noms pour le CR **ClusterLogForwarder** doit être **openshift-logging**.
3. Spécifiez un nom pour la sortie.
4. Spécifiez le type de **cloudwatch**.
5. Facultatif : Indiquez comment regrouper les journaux :
   - **logType** crée des groupes de journaux pour chaque type de journal
   - **namespaceName** crée un groupe de journaux pour chaque espace de noms d'applications. Il crée également des groupes de journaux distincts pour les journaux d'infrastructure et d'audit.
   - **namespaceUUID** crée un nouveau groupe de journaux pour chaque UUID d'espace de noms d'applications. Il crée également des groupes de journaux distincts pour les journaux d'infrastructure et d'audit.

6. Facultatif : Spécifiez une chaîne de caractères pour remplacer le préfixe par défaut **infrastructureName** dans les noms des groupes de journaux.
7. Spécifiez la région AWS.
8. Indiquez le nom du secret qui contient vos informations d'identification AWS.
9. Facultatif : Spécifiez un nom pour le pipeline.
10. Spécifiez les types de journaux à transférer en utilisant le pipeline : **application, infrastructure** ou **audit**.
11. Spécifiez le nom de la sortie à utiliser lors du transfert des journaux avec ce pipeline.

4. Créer l'objet CR :

```
oc create -f <nom-de-fichier>.yaml
```

Exemple : Utilisation de ClusterLogForwarder avec Amazon CloudWatch

Vous voyez ici un exemple de ressource personnalisée (CR) **ClusterLogForwarder** et les données de journal qu'elle envoie à Amazon CloudWatch.

Supposons que vous exécutiez un cluster OpenShift Container Platform nommé **mycluster**. La commande suivante renvoie l'adresse **infrastructureName** du cluster, que vous utiliserez pour composer des commandes **aws** par la suite :

```
$ oc get Infrastructure/cluster -ojson | jq .status.infrastructureName
"mycluster-7977k"
```

Pour générer les données de journalisation de cet exemple, vous exécutez un module **busybox** dans un espace de noms appelé **app**. Le module **busybox** écrit un message sur la sortie standard (stdout) toutes les trois secondes :

```
$ oc run busybox --image=busybox -- sh -c 'while true; do echo "My life is my message"; sleep 3;
done'
$ oc logs -f busybox
My life is my message
My life is my message
My life is my message
...
```

Vous pouvez consulter l'UUID de l'espace de noms **app** dans lequel s'exécute le pod **busybox**:

```
$ oc get ns/app -ojson | jq .metadata.uid
"794e1e1a-b9f5-4958-a190-e76a9b53d7bf"
```

Dans votre ressource personnalisée (CR) **ClusterLogForwarder**, vous configurez les types de journaux **infrastructure**, **audit** et **application** en tant qu'entrées du pipeline **all-logs**. Vous connectez également ce pipeline à la sortie **cw**, qui transmet les journaux à une instance CloudWatch dans la région **us-east-2**:

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
```

```
  name: instance
  namespace: openshift-logging
spec:
  outputs:
   - name: cw
     type: cloudwatch
     cloudwatch:
       groupBy: logType
       region: us-east-2
     secret:
        name: cw-secret
  pipelines:
    - name: all-logs
      inputRefs:
        - infrastructure
        - audit
        - application
      outputRefs:
        - cw
```

Chaque région dans CloudWatch contient trois niveaux d'objets :

- groupe de logs
  - flux de données
    - événement de journal

Avec **groupBy: logType** dans le CR **ClusterLogForwarding**, les trois types de journaux dans **inputRefs** produisent trois groupes de journaux dans Amazon Cloudwatch :

```
$ aws --output json logs describe-log-groups | jq .logGroups[].logGroupName
"mycluster-7977k.application"
"mycluster-7977k.audit"
"mycluster-7977k.infrastructure"
```

Chaque groupe de journaux contient des flux de journaux :

```
$ aws --output json logs describe-log-streams --log-group-name mycluster-7977k.application | jq
.logStreams[].logStreamName
"kubernetes.var.log.containers.busybox_app_busybox-
da085893053e20beddd6747acdbaf98e77c37718f85a7f6a4facf09ca195ad76.log"
```

```
$ aws --output json logs describe-log-streams --log-group-name mycluster-7977k.audit | jq
.logStreams[].logStreamName
"ip-10-0-131-228.us-east-2.compute.internal.k8s-audit.log"
"ip-10-0-131-228.us-east-2.compute.internal.linux-audit.log"
"ip-10-0-131-228.us-east-2.compute.internal.openshift-audit.log"
...
```

```
$ aws --output json logs describe-log-streams --log-group-name mycluster-7977k.infrastructure | jq
.logStreams[].logStreamName
"ip-10-0-131-228.us-east-2.compute.internal.kubernetes.var.log.containers.apiserver-69f9fd9b58-
zqzw5_openshift-oauth-apiserver_oauth-apiserver-
453c5c4ee026fe20a6139ba6b1cdd1bed25989c905bf5ac5ca211b7cbb5c3d7b.log"
```

```
"ip-10-0-131-228.us-east-2.compute.internal.kubernetes.var.log.containers.apiserver-797774f7c5-lftrx_openshift-apiserver_openshift-apiserver-ce51532df7d4e4d5f21c4f4be05f6575b93196336be0027067fd7d93d70f66a4.log"
"ip-10-0-131-228.us-east-2.compute.internal.kubernetes.var.log.containers.apiserver-797774f7c5-lftrx_openshift-apiserver_openshift-apiserver-check-endpoints-82a9096b5931b5c3b1d6dc4b66113252da4a6472c9fff48623baee761911a9ef.log"
...
```

Chaque flux de journaux contient des événements de journaux. Pour voir un événement du Pod **busybox**, vous devez spécifier son flux de journaux dans le groupe de journaux **application**:

```
$ aws logs get-log-events --log-group-name mycluster-7977k.application --log-stream-name kubernetes.var.log.containers.busybox_app_busybox-da085893053e20beddd6747acdbaf98e77c37718f85a7f6a4facf09ca195ad76.log
{
    "events": [
        {
            "timestamp": 1629422704178,
            "message": "{\"docker\":
{\"container_id\":\"da085893053e20beddd6747acdbaf98e77c37718f85a7f6a4facf09ca195ad76\"},\"kubernetes\":
{\"container_name\":\"busybox\",\"namespace_name\":\"app\",\"pod_name\":\"busybox\",\"container_image\":\"docker.io/library/busybox:latest\",\"container_image_id\":\"docker.io/library/busybox@sha256:0f354ec1728d9ff32edcd7d1b8bbdfc798277ad36120dc3dc683be44524c8b60\",\"pod_id\":\"870be234-90a3-4258-b73f-4f4d6e2777c7\",\"host\":\"ip-10-0-216-3.us-east-2.compute.internal\",\"labels\":
{\"run\":\"busybox\"},\"master_url\":\"https://kubernetes.default.svc\",\"namespace_id\":\"794e1e1a-b9f5-4958-a190-e76a9b53d7bf\",\"namespace_labels\":
{\"kubernetes_io/metadata_name\":\"app\"}},\"message\":\"My life is my message\",\"level\":\"unknown\",\"hostname\":\"ip-10-0-216-3.us-east-2.compute.internal\",\"pipeline_metadata\":{\"collector\":
{\"ipaddr4\":\"10.0.216.3\",\"inputname\":\"fluent-plugin-systemd\",\"name\":\"fluentd\",\"received_at\":\"2021-08-20T01:25:08.085760+00:00\",\"version\":\"1.7.4 1.6.0\"}},\"@timestamp\":\"2021-08-20T01:25:04.178986+00:00\",\"viaq_index_name\":\"app-write\",\"viaq_msg_id\":\"NWRjZmUyMWQtZjgzNC00MjI4LTk3MjMtNTk3NmY3ZjU4NDk1\",\"log_type\":\"application\",\"time\":\"2021-08-20T01:25:04+00:00\"}",
            "ingestionTime": 1629422744016
        },
...
```

Exemple : Personnalisation du préfixe dans les noms de groupes de journaux

Dans les noms de groupes de journaux, vous pouvez remplacer le préfixe par défaut **infrastructureName**, **mycluster-7977k**, par une chaîne arbitraire telle que **demo-group-prefix**. Pour ce faire, vous devez mettre à jour le champ **groupPrefix** dans le CR **ClusterLogForwarding**:

```
cloudwatch:
  groupBy: logType
  groupPrefix: demo-group-prefix
  region: us-east-2
```

La valeur de **groupPrefix** remplace le préfixe par défaut **infrastructureName**:

```
$ aws --output json logs describe-log-groups | jq .logGroups[].logGroupName
"demo-group-prefix.application"
```

```
"demo-group-prefix.audit"
"demo-group-prefix.infrastructure"
```

Exemple : Nommer les groupes de journaux d'après les noms des espaces de noms des applications

Pour chaque espace de noms d'applications dans votre cluster, vous pouvez créer un groupe de logs dans CloudWatch dont le nom est basé sur le nom de l'espace de noms d'applications.

Si vous supprimez un objet d'espace de noms d'application et en créez un nouveau qui porte le même nom, CloudWatch continue d'utiliser le même groupe de journaux qu'auparavant.

Si vous considérez que des objets successifs de l'espace de noms d'applications qui portent le même nom sont équivalents, utilisez l'approche décrite dans cet exemple. Dans le cas contraire, si vous devez distinguer les groupes de journaux résultants les uns des autres, reportez-vous plutôt à la section suivante "Nommer les groupes de journaux pour les UUID de l'espace de noms d'application".

Pour créer des groupes de journaux d'application dont les noms sont basés sur les noms des espaces de noms d'application, vous définissez la valeur du champ **groupBy** sur **namespaceName** dans le CR **ClusterLogForwarder**:

```
cloudwatch:
  groupBy: namespaceName
  region: us-east-2
```

Le réglage de **groupBy** à **namespaceName** n'affecte que le groupe d'enregistrement de l'application. Il n'affecte pas les groupes de journaux **audit** et **infrastructure**.

Dans Amazon Cloudwatch, le nom de l'espace de noms apparaît à la fin de chaque nom de groupe de journaux. Comme il n'existe qu'un seul espace de noms d'application, \N "app", la sortie suivante montre un nouveau groupe de journaux **mycluster-7977k.app** au lieu de **mycluster-7977k.application**:

```
$ aws --output json logs describe-log-groups | jq .logGroups[].logGroupName
"mycluster-7977k.app"
"mycluster-7977k.audit"
"mycluster-7977k.infrastructure"
```

Si le cluster de cet exemple avait contenu plusieurs espaces de noms d'applications, la sortie montrerait plusieurs groupes de journaux, un pour chaque espace de noms.

Le champ **groupBy** n'affecte que le groupe de journaux d'application. Il n'affecte pas les groupes de journaux **audit** et **infrastructure**.

Exemple : Nommer les groupes de journaux d'après les UUID de l'espace de noms de l'application

Pour chaque espace de noms d'applications dans votre cluster, vous pouvez créer un groupe de logs dans CloudWatch dont le nom est basé sur l'UUID de l'espace de noms d'applications.

Si vous supprimez un objet d'espace de noms d'application et en créez un nouveau, CloudWatch crée un nouveau groupe de journaux.

Si vous considérez que des objets successifs de l'espace de noms de l'application portant le même nom sont différents les uns des autres, utilisez l'approche décrite dans cet exemple. Sinon, consultez la section précédente "Exemple : Nommer les groupes de journaux pour les espaces de noms d'applications".

Pour nommer les groupes de journaux d'après les UUID de l'espace de noms de l'application, vous définissez la valeur du champ **groupBy** sur **namespaceUUID** dans le CR **ClusterLogForwarder**:

```
cloudwatch:
  groupBy: namespaceUUID
  region: us-east-2
```

Dans Amazon Cloudwatch, l'UUID de l'espace de noms apparaît à la fin de chaque nom de groupe de journaux. Étant donné qu'il n'existe qu'un seul espace de noms d'application, "app", la sortie suivante montre un nouveau groupe de journaux **mycluster-7977k.794e1e1a-b9f5-4958-a190-e76a9b53d7bf** au lieu de **mycluster-7977k.application**:

```
$ aws --output json logs describe-log-groups | jq .logGroups[].logGroupName
"mycluster-7977k.794e1e1a-b9f5-4958-a190-e76a9b53d7bf" // uid of the "app" namespace
"mycluster-7977k.audit"
"mycluster-7977k.infrastructure"
```

Le champ **groupBy** n'affecte que le groupe de journaux d'application. Il n'affecte pas les groupes de journaux **audit** et **infrastructure**.

10.12.1. Transférer les journaux vers Amazon CloudWatch à partir de clusters compatibles avec STS

Pour les clusters avec AWS Security Token Service (STS) activé, vous pouvez créer un compte de service AWS manuellement ou créer une demande d'informations d'identification à l'aide de l'utilitaire Cloud Credential Operator (CCO) **ccoctl**.

NOTE

Cette fonction n'est pas prise en charge par le collecteur de vecteurs.

Conditions préalables

- Sous-système de journalisation pour Red Hat OpenShift : 5.5 et versions ultérieures

Procédure

1. Créez une ressource personnalisée YAML pour **CredentialsRequest** en utilisant le modèle ci-dessous :

   Modèle de demande d'informations d'identification CloudWatch

   ```
   apiVersion: cloudcredential.openshift.io/v1
   kind: CredentialsRequest
   metadata:
     name: <your_role_name>-credrequest
     namespace: openshift-cloud-credential-operator
   spec:
     providerSpec:
       apiVersion: cloudcredential.openshift.io/v1
       kind: AWSProviderSpec
       statementEntries:
         - action:
             - logs:PutLogEvents
   ```

```
        - logs:CreateLogGroup
        - logs:PutRetentionPolicy
        - logs:CreateLogStream
        - logs:DescribeLogGroups
        - logs:DescribeLogStreams
      effect: Allow
      resource: arn:aws:logs:*:*:*
  secretRef:
    name: <your_role_name>
    namespace: openshift-logging
  serviceAccountNames:
    - logcollector
```

2. Utilisez la commande **ccoctl** pour créer un rôle pour AWS à l'aide de votre CR **CredentialsRequest**. Avec l'objet **CredentialsRequest**, cette commande **ccoctl** crée un rôle IAM avec une politique de confiance liée au fournisseur d'identité OIDC spécifié et une politique de permissions qui accorde des permissions pour effectuer des opérations sur les ressources CloudWatch. Cette commande crée également un fichier de configuration YAML dans **/<path_to_ccoctl_output_dir>/manifests/openshift-logging-<your_role_name>-credentials.yaml**. Ce fichier secret contient la clé/valeur **role_arn** utilisée lors de l'authentification avec le fournisseur d'identité AWS IAM.

```
$ ccoctl aws create-iam-roles \
--name=<name> \
--region=<aws_region> \
--credentials-requests-dir=
<path_to_directory_with_list_of_credentials_requests>/credrequests \
--identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.
<aws_region>.amazonaws.com (1)
```

(1) <name> est le nom utilisé pour marquer vos ressources cloud et doit correspondre au nom utilisé lors de l'installation de votre cluster STS

3. Appliquer le secret créé :

```
oc apply -f output/manifests/openshift-logging-<votre_nom_de_rôle>-credentials.yaml
```

4. Créer ou modifier une ressource personnalisée **ClusterLogForwarder**:

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance (1)
  namespace: openshift-logging (2)
spec:
  outputs:
   - name: cw (3)
     type: cloudwatch (4)
     cloudwatch:
       groupBy: logType (5)
       groupPrefix: <group prefix> (6)
       region: us-east-2 (7)
     secret:
```

```
      name: <your_role_name> 8
pipelines:
  - name: to-cloudwatch 9
    inputRefs: 10
      - infrastructure
      - audit
      - application
    outputRefs:
      - cw 11
```

1. Le nom du CR **ClusterLogForwarder** doit être **instance**.
2. L'espace de noms pour le CR **ClusterLogForwarder** doit être **openshift-logging**.
3. Spécifiez un nom pour la sortie.
4. Spécifiez le type de **cloudwatch**.
5. Facultatif : Indiquez comment regrouper les journaux :
   - **logType** crée des groupes de journaux pour chaque type de journal
   - **namespaceName** crée un groupe de journaux pour chaque espace de noms d'applications. Les journaux d'infrastructure et d'audit ne sont pas affectés et restent regroupés par **logType**.
   - **namespaceUUID** crée un nouveau groupe de journaux pour chaque UUID d'espace de noms d'applications. Il crée également des groupes de journaux distincts pour les journaux d'infrastructure et d'audit.
6. Facultatif : Spécifiez une chaîne de caractères pour remplacer le préfixe par défaut **infrastructureName** dans les noms des groupes de journaux.
7. Spécifiez la région AWS.
8. Indiquez le nom du secret qui contient vos informations d'identification AWS.
9. Facultatif : Spécifiez un nom pour le pipeline.
10. Spécifiez les types de journaux à transférer en utilisant le pipeline : **application, infrastructure** ou **audit**.
11. Spécifiez le nom de la sortie à utiliser lors du transfert des journaux avec ce pipeline.

Ressources complémentaires

- Référence de l'API AWS STS

10.12.1.1. Création d'un secret pour AWS CloudWatch avec un rôle AWS existant

Si vous avez un rôle existant pour AWS, vous pouvez créer un secret pour AWS avec STS à l'aide de la commande **oc create secret --from-literal**.

Procédure

- Dans l'interface de commande, entrez la commande suivante pour générer un secret pour AWS :

```
$ oc create secret generic cw-sts-secret -n openshift-logging --from-
literal=role_arn=arn:aws:iam::123456789012:role/my-role_with-permissions
```

Exemple Secret

```
apiVersion: v1
kind: Secret
metadata:
  namespace: openshift-logging
  name: my-secret-name
stringData:
  role_arn: arn:aws:iam::123456789012:role/my-role_with-permissions
```

10.13. TRANSFÉRER LES JOURNAUX À LOKI

Vous pouvez transmettre les journaux à un système de journalisation externe Loki en plus ou à la place de l'instance Elasticsearch interne par défaut d'OpenShift Container Platform.

Pour configurer la transmission des journaux à Loki, vous devez créer une ressource personnalisée (CR) **ClusterLogForwarder** avec une sortie vers Loki et un pipeline qui utilise la sortie. La sortie vers Loki peut utiliser la connexion HTTP (non sécurisée) ou HTTPS (HTTP sécurisée).

Conditions préalables

- Un système de journalisation Loki doit fonctionner à l'URL spécifiée dans le champ **url** du CR.

Procédure

1. Créez ou modifiez un fichier YAML qui définit l'objet **ClusterLogForwarder** CR :

```
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  outputs:
   - name: loki-insecure 3
     type: "loki" 4
     url: http://loki.insecure.com:3100 5
     loki:
       tenantKey: kubernetes.namespace_name
       labelKeys: kubernetes.labels.foo
   - name: loki-secure 6
     type: "loki"
     url: https://loki.secure.com:3100
     secret:
       name: loki-secret 7
     loki:
       tenantKey: kubernetes.namespace_name 8
       labelKeys: kubernetes.labels.foo 9
```

```
pipelines:
 - name: application-logs 10
   inputRefs: 11
   - application
   - audit
   outputRefs: 12
   - loki-secure
```

1. Le nom du CR **ClusterLogForwarder** doit être **instance**.
2. L'espace de noms pour le CR **ClusterLogForwarder** doit être **openshift-logging**.
3. Spécifiez un nom pour la sortie.
4. Spécifiez le type comme **"loki"**.
5. Spécifiez l'URL et le port du système Loki sous la forme d'une URL absolue valide. Vous pouvez utiliser le protocole **http** (non sécurisé) ou **https** (HTTP sécurisé). Si le proxy à l'échelle du cluster utilisant l'annotation CIDR est activé, la sortie doit être un nom de serveur ou un FQDN, et non une adresse IP. Le port par défaut de Loki pour la communication HTTP(S) est 3100.
6. Pour une connexion sécurisée, vous pouvez spécifier une URL **https** ou **http** que vous authentifiez en spécifiant un **secret**.
7. Pour un préfixe **https**, indiquez le nom du secret requis par le point d'extrémité pour la communication TLS. Le secret doit exister dans le projet **openshift-logging** et doit avoir des clés de : **tls.crt**, **tls.key**, et **ca-bundle.crt** qui pointent vers les certificats respectifs qu'elles représentent. Sinon, pour les préfixes **http** et **https**, vous pouvez spécifier un secret contenant un nom d'utilisateur et un mot de passe. Pour plus d'informations, voir l'exemple suivant : "Exemple : Définition d'un secret contenant un nom d'utilisateur et un mot de passe.\N-"
8. Facultatif : Spécifiez un champ clé de métadonnées pour générer des valeurs pour le champ **TenantID** dans Loki. Par exemple, le paramètre **tenantKey: kubernetes.namespace_name** utilise les noms des espaces de noms Kubernetes comme valeurs pour les identifiants de locataire dans Loki. Pour connaître les autres champs d'enregistrement que vous pouvez spécifier, voir le lien "Champs d'enregistrement" dans la section suivante "Ressources supplémentaires".
9. Facultatif : Spécifiez une liste de clés de champs de métadonnées pour remplacer les étiquettes Loki par défaut. Les noms des étiquettes Loki doivent correspondre à l'expression régulière **[a-zA-Z_:][a-zA-Z0-9_:]***. Les caractères illégaux dans les clés de métadonnées sont remplacés par _ pour former le nom de l'étiquette. Par exemple, la clé de métadonnées **kubernetes.labels.foo** devient l'étiquette Loki **kubernetes_labels_foo**. Si vous ne définissez pas **labelKeys**, la valeur par défaut est : **[log_type, kubernetes.namespace_name, kubernetes.pod_name, kubernetes_host]**. Veillez à ce que l'ensemble des étiquettes soit restreint, car Loki limite la taille et le nombre d'étiquettes autorisées. Voir Configuration de Loki, limits_config . Vous pouvez toujours effectuer des requêtes basées sur n'importe quel champ de l'enregistrement à l'aide de filtres de requête.
10. Facultatif : Spécifiez un nom pour le pipeline.
11. Spécifiez les types de journaux à transférer en utilisant le pipeline : **application, infrastructure** ou **audit**.

12 Spécifiez le nom de la sortie à utiliser lors du transfert des journaux avec ce pipeline.

NOTE

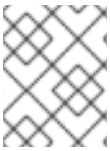
Comme Loki exige que les flux de journaux soient correctement classés par date, **labelKeys** inclut toujours le jeu d'étiquettes **kubernetes_host**, même si vous ne le spécifiez pas. Cette inclusion garantit que chaque flux provient d'un seul hôte, ce qui empêche les horodatages d'être désordonnés en raison des différences d'horloge sur les différents hôtes.

2. Créer l'objet CR :

```
oc create -f <nom-de-fichier>.yaml
```

10.13.1. Dépannage des erreurs de Loki "entry out of order" (entrée en dehors de l'ordre)

Si votre Fluentd transmet un grand bloc de messages à un système de journalisation Loki qui dépasse la limite de débit, Loki génère des erreurs "entry out of order". Pour résoudre ce problème, vous devez mettre à jour certaines valeurs dans le fichier de configuration du serveur Loki, **loki.yaml**.



NOTE

loki.yaml n'est pas disponible sur les serveurs Loki hébergés par Grafana. Cette rubrique ne s'applique pas aux serveurs Loki hébergés par Grafana.

Conditions

- La ressource personnalisée **ClusterLogForwarder** est configurée pour transmettre les journaux à Loki.
- Votre système envoie à Loki un bloc de messages d'une taille supérieure à 2 Mo, par exemple :

```
"values":[["1630410392689800468","{\"kind\":\"Event\",\"apiVersion\":\
.......
......
......
......
\"received_at\":\"2021-08-31T11:46:32.800278+00:00\",\"version\":\"1.7.4
1.6.0\"}},\"@timestamp\":\"2021-08-
31T11:46:32.799692+00:00\",\"viaq_index_name\":\"audit-
write\",\"viaq_msg_id\":\"MzFjYjJkZjItNjY0MC00YWU4LWIwMTEtNGNmM2E5ZmViMGU4\",\"lo
g_type\":\"audit\"}"]]}]}
```

- Lorsque vous entrez **oc logs -c fluentd**, les journaux Fluentd dans votre cluster OpenShift Logging affichent les messages suivants :

```
429 Too Many Requests Ingestion rate limit exceeded (limit: 8388608 bytes/sec) while
attempting to ingest '2140' lines totaling '3285284' bytes
```

> 429 Too Many Requests Ingestion rate limit exceeded' or '500 Internal Server Error rpc error: code = ResourceExhausted desc = grpc: received message larger than max (5277702 vs. 4194304)'

- Lorsque vous ouvrez les journaux sur le serveur Loki, ils affichent des messages **entry out of order** comme ceux-ci :

> ,\nentry with timestamp 2021-08-18 05:58:55.061936 +0000 UTC ignored, reason: 'entry out of order' for stream:
>
> {fluentd_thread=\"flush_thread_0\", log_type=\"audit\"},\nentry with timestamp 2021-08-18 06:01:18.290229 +0000 UTC ignored, reason: 'entry out of order' for stream: {fluentd_thread="flush_thread_0", log_type="audit"}

Procédure

1. Mettez à jour les champs suivants dans le fichier de configuration **loki.yaml** sur le serveur Loki avec les valeurs indiquées ici :
   - **grpc_server_max_recv_msg_size: 8388608**
   - **chunk_target_size: 8388608**
   - **ingestion_rate_mb: 8**
   - **ingestion_burst_size_mb: 16**
2. Appliquez les changements dans **loki.yaml** au serveur Loki.

Exemple de fichier loki.yaml

```
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096
  grpc_server_max_recv_msg_size: 8388608

ingester:
  wal:
    enabled: true
    dir: /tmp/wal
  lifecycler:
    address: 127.0.0.1
    ring:
      kvstore:
        store: inmemory
      replication_factor: 1
    final_sleep: 0s
  chunk_idle_period: 1h       # Any chunk not receiving new logs in this time will be flushed
  chunk_target_size: 8388608
  max_chunk_age: 1h           # All chunks will be flushed when they hit this age, default is 1h
  chunk_retain_period: 30s    # Must be greater than index read cache TTL if using an index cache
(Default index read cache TTL is 5m)
  max_transfer_retries: 0     # Chunk transfers disabled
```

```
schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

storage_config:
  boltdb_shipper:
    active_index_directory: /tmp/loki/boltdb-shipper-active
    cache_location: /tmp/loki/boltdb-shipper-cache
    cache_ttl: 24h         # Can be increased for faster performance over longer query periods, uses
more disk space
    shared_store: filesystem
  filesystem:
    directory: /tmp/loki/chunks

compactor:
  working_directory: /tmp/loki/boltdb-shipper-compactor
  shared_store: filesystem

limits_config:
  reject_old_samples: true
  reject_old_samples_max_age: 12h
  ingestion_rate_mb: 8
  ingestion_burst_size_mb: 16

chunk_store_config:
  max_look_back_period: 0s

table_manager:
  retention_deletes_enabled: false
  retention_period: 0s

ruler:
  storage:
    type: local
    local:
      directory: /tmp/loki/rules
  rule_path: /tmp/loki/rules-temp
  alertmanager_url: http://localhost:9093
  ring:
    kvstore:
      store: inmemory
  enable_api: true
```

Ressources complémentaires

- Configuration de Loki

Ressources complémentaires

- Champs de l'enregistrement du journal .
- Configuration du serveur Loki

10.14. TRANSFÉRER LES JOURNAUX VERS GOOGLE CLOUD PLATFORM (GCP)

Vous pouvez transférer les logs vers Google Cloud Logging en plus ou à la place du magasin de logs interne par défaut d'OpenShift Container Platform.

NOTE

L'utilisation de cette fonction avec Fluentd n'est pas prise en charge.

Conditions préalables

- Sous-système de journalisation pour Red Hat OpenShift Operator 5.5.1 et versions ultérieures

Procédure

1. Créez un secret en utilisant la clé de votre compte de service Google .

```
oc -n openshift-logging create secret generic gcp-secret --from-file google-application-credentials.json= $ oc -n openshift-logging create secret generic gcp-secret --from-file google-application-credentials.json=<your_service_account_key_file.json>
```

2. Créez une ressource personnalisée YAML pour **ClusterLogForwarder** en utilisant le modèle ci-dessous :

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogForwarder"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  outputs:
    - name: gcp-1
      type: googleCloudLogging
      secret:
        name: gcp-secret
      googleCloudLogging:
        projectId : "openshift-gce-devel" 1
        logId : "app-gcp" 2
  pipelines:
    - name: test-app
      inputRefs: 3
        - application
      outputRefs:
        - gcp-1
```

1 Définissez un champ **projectId**, **folderId**, **organizationId**, ou **billingAccountId** et sa valeur correspondante, en fonction de l'endroit où vous souhaitez stocker vos journaux dans la hiérarchie des ressources GCP.

2 Définissez la valeur à ajouter au champ **logName** de l'entrée de journal.

3 Spécifiez les types de journaux à transférer en utilisant le pipeline : **application**, **infrastructure**, ou **audit**.

Ressources complémentaires

- Documentation sur la facturation de Google Cloud
- Documentation de Google Cloud Logging Query Language

10.15. TRANSFÉRER LES LOGS VERS SPLUNK

Vous pouvez transmettre les journaux au collecteur d'événements HTTP Splunk (HEC) en plus ou à la place du magasin de journaux interne par défaut d'OpenShift Container Platform.

NOTE

L'utilisation de cette fonction avec Fluentd n'est pas prise en charge.

Conditions préalables

- Red Hat OpenShift Logging Operator 5.6 et supérieur
- Instance de ClusterLogging avec un vecteur spécifié comme collecteur
- Jeton Splunk HEC encodé en Base64

Procédure

1. Créez un secret en utilisant votre jeton Splunk HEC encodé en Base64.

```
oc -n openshift-logging create secret generic vector-splunk-secret --from-literal hecToken=
<HEC_Token>
```

2. Créez ou modifiez la ressource personnalisée (CR) **ClusterLogForwarder** à l'aide du modèle ci-dessous :

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogForwarder"
metadata:
  name: "instance" 1
  namespace: "openshift-logging" 2
spec:
  outputs:
    - name: splunk-receiver 3
      secret:
        name: vector-splunk-secret 4
      type: splunk 5
      url: <http://your.splunk.hec.url:8088> 6
  pipelines: 7
    - inputRefs:
```

```
    - application
    - infrastructure
  name: 8
  outputRefs:
    - splunk-receiver 9
```

1. Le nom du CR ClusterLogForwarder doit être **instance**.
2. L'espace de noms du CR ClusterLogForwarder doit être **openshift-logging**.
3. Spécifiez un nom pour la sortie.
4. Indiquez le nom du secret qui contient votre jeton HEC.
5. Spécifiez le type de sortie comme **splunk**.
6. Spécifiez l'URL (y compris le port) de votre Splunk HEC.
7. Spécifiez les types de journaux à transférer en utilisant le pipeline : **application**, **infrastructure**, ou **audit**.
8. Facultatif : Spécifiez un nom pour le pipeline.
9. Spécifiez le nom de la sortie à utiliser lors du transfert des journaux avec ce pipeline.

10.16. TRANSMISSION DES JOURNAUX D'APPLICATION DE PROJETS SPÉCIFIQUES

Vous pouvez utiliser le Cluster Log Forwarder pour envoyer une copie des journaux d'application de projets spécifiques à un agrégateur de journaux externe. Vous pouvez le faire en plus ou à la place de l'utilisation du magasin de logs Elasticsearch par défaut. Vous devez également configurer l'agrégateur de logs externe pour qu'il reçoive les données de logs d'OpenShift Container Platform.

Pour configurer le transfert des journaux d'application à partir d'un projet, vous devez créer une ressource personnalisée (CR) **ClusterLogForwarder** avec au moins une entrée provenant d'un projet, des sorties facultatives pour d'autres agrégateurs de journaux et des pipelines qui utilisent ces entrées et ces sorties.

Conditions préalables

- Vous devez disposer d'un serveur de journalisation configuré pour recevoir les données de journalisation à l'aide du protocole ou du format spécifié.

Procédure

1. Créez ou modifiez un fichier YAML qui définit l'objet **ClusterLogForwarder** CR :

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  outputs:
```

```
  - name: fluentd-server-secure 3
    type: fluentdForward 4
    url: 'tls://fluentdserver.security.example.com:24224' 5
    secret: 6
      name: fluentd-secret
  - name: fluentd-server-insecure
    type: fluentdForward
    url: 'tcp://fluentdserver.home.example.com:24224'
inputs: 7
  - name: my-app-logs
    application:
      namespaces:
      - my-project
pipelines:
  - name: forward-to-fluentd-insecure 8
    inputRefs: 9
    - my-app-logs
    outputRefs: 10
    - fluentd-server-insecure
    parse: json 11
    labels:
      project: "my-project" 12
  - name: forward-to-fluentd-secure 13
    inputRefs:
    - application
    - audit
    - infrastructure
    outputRefs:
    - fluentd-server-secure
    - default
    labels:
      clusterId: "C1234"
```

1. Le nom du CR **ClusterLogForwarder** doit être **instance**.
2. L'espace de noms pour le CR **ClusterLogForwarder** doit être **openshift-logging**.
3. Spécifiez un nom pour la sortie.
4. Spécifiez le type de sortie : **elasticsearch**, **fluentdForward**, **syslog**, ou **kafka**.
5. Spécifiez l'URL et le port de l'agrégateur de journaux externe sous la forme d'une URL absolue valide. Si le proxy à l'échelle du cluster utilisant l'annotation CIDR est activé, la sortie doit être un nom de serveur ou un FQDN, et non une adresse IP.
6. Si vous utilisez un préfixe **tls**, vous devez spécifier le nom du secret requis par le point final pour la communication TLS. Le secret doit exister dans le projet **openshift-logging** et avoir des clés **tls.crt**, **tls.key** et **ca-bundle.crt** qui pointent chacune vers les certificats qu'elles représentent.
7. Configuration d'une entrée pour filtrer les journaux d'application des projets spécifiés.
8. Configuration d'un pipeline qui utilise l'entrée pour envoyer les logs de l'application du projet à une instance Fluentd externe.

9 L'entrée **my-app-logs**.

10 Le nom de la sortie à utiliser.

11 Facultatif : Indiquez si les entrées de log JSON structurées doivent être transmises en tant qu'objets JSON dans le champ **structured**. L'entrée de log doit contenir du JSON structuré valide ; sinon, OpenShift Logging supprime le champ **structured** et envoie l'entrée de log à l'index par défaut, **app-00000x**.

12 Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux.

13 Configuration d'un pipeline pour l'envoi de journaux à d'autres agrégateurs de journaux.

- Facultatif : Spécifiez un nom pour le pipeline.
- Spécifiez les types de journaux à transférer en utilisant le pipeline : **application, infrastructure** ou **audit**.
- Spécifiez le nom de la sortie à utiliser lors du transfert des journaux avec ce pipeline.
- Facultatif : Spécifiez la sortie **default** pour transmettre les journaux à l'instance Elasticsearch interne.
- Facultatif : Chaîne. Une ou plusieurs étiquettes à ajouter aux journaux.

2. Créer l'objet CR :

```
oc create -f <nom-de-fichier>.yaml
```

10.17. TRANSFÉRER LES JOURNAUX D'APPLICATION DE PODS SPÉCIFIQUES

En tant qu'administrateur de cluster, vous pouvez utiliser les étiquettes de pods Kubernetes pour collecter des données de journal à partir de pods spécifiques et les transmettre à un collecteur de journaux.

Supposons que vous ayez une application composée de pods fonctionnant avec d'autres pods dans différents espaces de noms. Si ces pods ont des étiquettes qui identifient l'application, vous pouvez collecter et envoyer leurs données de journalisation vers un collecteur de journaux spécifique.

Pour spécifier les étiquettes de pods, vous utilisez une ou plusieurs paires clé-valeur **matchLabels**. Si vous spécifiez plusieurs paires clé-valeur, les pods doivent tous correspondre à ces paires pour être sélectionnés.

Procédure

1. Créez ou modifiez un fichier YAML qui définit l'objet **ClusterLogForwarder** CR. Dans le fichier, spécifiez les étiquettes de pods à l'aide de sélecteurs simples basés sur l'égalité sous **inputs[].name.application.selector.matchLabels**, comme le montre l'exemple suivant.

Exemple ClusterLogForwarder Fichier YAML de CR

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogForwarder
```

```
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  pipelines:
    - inputRefs: [ myAppLogData ] 3
      outputRefs: [ default ] 4
      parse: json 5
  inputs: 6
    - name: myAppLogData
      application:
        selector:
          matchLabels: 7
            environment: production
            app: nginx
        namespaces: 8
        - app1
        - app2
  outputs: 9
    - default
    ...
```

1. Le nom du CR **ClusterLogForwarder** doit être **instance**.
2. L'espace de noms pour le CR **ClusterLogForwarder** doit être **openshift-logging**.
3. Spécifiez une ou plusieurs valeurs séparées par des virgules à partir de **inputs[].name**.
4. Spécifiez une ou plusieurs valeurs séparées par des virgules à partir de **outputs[]**.
5. Facultatif : Indiquez si les entrées de log JSON structurées doivent être transmises en tant qu'objets JSON dans le champ **structured**. L'entrée de log doit contenir du JSON structuré valide ; sinon, OpenShift Logging supprime le champ **structured** et envoie l'entrée de log à l'index par défaut, **app-00000x**.
6. Définir un site **inputs[].name** unique pour chaque application qui possède un ensemble unique d'étiquettes de pods.
7. Indiquez les paires clé-valeur des étiquettes de pods dont vous souhaitez collecter les données de journalisation. Vous devez spécifier à la fois une clé et une valeur, et pas seulement une clé. Pour être sélectionnés, les modules doivent correspondre à toutes les paires clé-valeur.
8. Facultatif : Indiquez un ou plusieurs espaces de noms.
9. Spécifiez une ou plusieurs sorties pour transmettre vos données de journalisation. La sortie optionnelle **default** montrée ici envoie les données de log à l'instance Elasticsearch interne.

2. Facultatif : Pour limiter la collecte de données de journalisation à des espaces de noms spécifiques, utilisez **inputs[].name.application.namespaces**, comme indiqué dans l'exemple précédent.

3. Facultatif : vous pouvez envoyer des données de journal provenant d'applications supplémentaires ayant des étiquettes de pods différentes vers le même pipeline.

   a. Pour chaque combinaison unique d'étiquettes de gousses, créez une section supplémentaire **inputs[].name** semblable à celle illustrée.

   b. Mettre à jour le site **selectors** pour qu'il corresponde aux étiquettes de la capsule de cette demande.

   c. Ajoutez la nouvelle valeur **inputs[].name** à **inputRefs**. Par exemple :

   ```
   - inputRefs: [ myAppLogData, myOtherAppLogData ]
   ```

4. Créer l'objet CR :

   ```
   oc create -f <nom-de-fichier>.yaml
   ```

Ressources complémentaires

- Pour plus d'informations sur **matchLabels** dans Kubernetes, voir Ressources qui prennent en charge les exigences basées sur des ensembles.

Ressources complémentaires

- Journalisation des règles de pare-feu et de stratégie de réseau de sortie

10.18. DÉPANNAGE DE LA REDIRECTION DES JOURNAUX

Lorsque vous créez une ressource personnalisée (CR) **ClusterLogForwarder**, si Red Hat OpenShift Logging Operator ne redéploie pas automatiquement les pods Fluentd, vous pouvez supprimer les pods Fluentd pour les forcer à se redéployer.

Conditions préalables

- Vous avez créé un objet ressource personnalisé (CR) **ClusterLogForwarder**.

Procédure

- Supprimer les pods Fluentd pour les forcer à se redéployer.

  ```
  $ oc delete pod --selector logging-infra=collector
  ```

CHAPITRE 11. ACTIVATION DE LA JOURNALISATION JSON

Vous pouvez configurer l'API Log Forwarding pour qu'elle analyse les chaînes JSON en un objet structuré.

11.1. ANALYSE DES JOURNAUX JSON

Les journaux, y compris les journaux JSON, sont généralement représentés sous la forme d'une chaîne de caractères dans le champ **message**. Il est donc difficile pour les utilisateurs d'interroger des champs spécifiques à l'intérieur d'un document JSON. L'API Log Forwarding d'OpenShift Logging vous permet d'analyser les logs JSON en un objet structuré et de les transmettre à Elasticsearch, géré par OpenShift Logging, ou à tout autre système tiers pris en charge par l'API Log Forwarding.

Pour illustrer ce fonctionnement, supposons que vous ayez l'entrée de journal JSON structurée suivante.

Exemple d'entrée de journal JSON structurée

```
{"level":"info","name":"fred","home":"bedrock"}
```

Normalement, la ressource personnalisée (CR) **ClusterLogForwarder** transmet cette entrée de journal dans le champ **message**. Le champ **message** contient l'équivalent de la chaîne de caractères JSON de l'entrée de journal JSON, comme le montre l'exemple suivant.

Exemple **message** champ

```
{"message":"{\"level\":\"info\",\"name\":\"fred\",\"home\":\"bedrock\"",
 "more fields..."}
```

Pour activer l'analyse du journal JSON, vous ajoutez **parse: json** à un pipeline dans le CR **ClusterLogForwarder**, comme le montre l'exemple suivant.

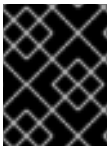
Exemple d'extrait montrant **parse: json**

```
pipelines:
- inputRefs: [ application ]
  outputRefs: myFluentd
  parse: json
```

Lorsque vous activez l'analyse des journaux JSON à l'aide de **parse: json**, le CR copie l'entrée de journal structurée en JSON dans un champ **structured**, comme le montre l'exemple suivant. Cela ne modifie pas le champ original **message**.

Exemple **structured** contenant l'entrée de journal JSON structurée

```
{"structured": { "level": "info", "name": "fred", "home": "bedrock" },
 "more fields..."}
```



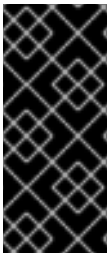
IMPORTANT

Si l'entrée du journal ne contient pas de JSON structuré valide, le champ **structured** sera absent.

Pour activer l'analyse des journaux JSON pour des plates-formes de journalisation spécifiques, voir Transférer les journaux vers des systèmes tiers .

11.2. CONFIGURATION DES DONNÉES DE JOURNALISATION JSON POUR ELASTICSEARCH

Si vos journaux JSON suivent plusieurs schémas, leur stockage dans un index unique peut entraîner des conflits de type et des problèmes de cardinalité. Pour éviter cela, vous devez configurer la ressource personnalisée (CR) **ClusterLogForwarder** pour regrouper chaque schéma dans une définition de sortie unique. De cette manière, chaque schéma est transmis à un index distinct.



IMPORTANT

Si vous transmettez des logs JSON à l'instance Elasticsearch par défaut gérée par OpenShift Logging, celle-ci génère de nouveaux index en fonction de votre configuration. Pour éviter les problèmes de performance associés à un trop grand nombre d'index, envisagez de maintenir le nombre de schémas possibles à un niveau bas en standardisant les schémas communs.

Types de structures

Vous pouvez utiliser les types de structure suivants dans le CR **ClusterLogForwarder** pour construire des noms d'index pour le magasin de journaux Elasticsearch :

- **structuredTypeKey** (chaîne, facultatif) est le nom d'un champ de message. La valeur de ce champ, si elle est présente, est utilisée pour construire le nom de l'index.
    - **kubernetes.labels.<key>** est l'étiquette du pod Kubernetes dont la valeur est utilisée pour construire le nom de l'index.
    - **openshift.labels.<key>** est l'élément **pipeline.label.<key>** du CR **ClusterLogForwarder** dont la valeur est utilisée pour construire le nom de l'index.
    - **kubernetes.container_name** utilise le nom du conteneur pour construire le nom de l'index.
- **structuredTypeName**: (string, optional) Si **structuredTypeKey** n'est pas défini ou si sa clé n'est pas présente, OpenShift Logging utilise la valeur de **structuredTypeName** comme type structuré. Lorsque vous utilisez à la fois **structuredTypeKey** et **structuredTypeName**, **structuredTypeName** fournit un nom d'index de repli si la clé de **structuredTypeKey** est absente des données du journal JSON.

NOTE

Bien que vous puissiez attribuer à **structuredTypeKey** la valeur de n'importe quel champ présenté dans la rubrique "Champs d'enregistrement des journaux", les champs les plus utiles sont présentés dans la liste précédente des types de structure.

Une clé structuréeTypeKey : kubernetes.labels.<key> example

Supposons ce qui suit :

- Votre cluster exécute des pods d'application qui produisent des journaux JSON dans deux formats différents, "apache" et "google".

- L'utilisateur étiquette ces modules d'application avec **logFormat=apache** et **logFormat=google**.
- Vous utilisez l'extrait suivant dans votre fichier YAML **ClusterLogForwarder** CR.

```
outputDefaults:
 elasticsearch:
    structuredTypeKey: kubernetes.labels.logFormat 1
    structuredTypeName: nologformat
pipelines:
- inputRefs: <application>
  outputRefs: default
  parse: json 2
```

1 Utilise la valeur de la paire clé-valeur formée par l'étiquette Kubernetes **logFormat**.

2 Permet d'analyser les journaux JSON.

Dans ce cas, l'enregistrement structuré suivant est placé dans l'index **app-apache-write**:

```
{
  "structured":{"name":"fred","home":"bedrock"},
  "kubernetes":{"labels":{"logFormat": "apache", ...}}
}
```

L'enregistrement structuré suivant est placé dans l'index **app-google-write**:

```
{
  "structured":{"name":"wilma","home":"bedrock"},
  "kubernetes":{"labels":{"logFormat": "google", ...}}
}
```

Une clé structuréeTypeKey : openshift.labels.<key> exemple

Supposons que vous utilisiez l'extrait suivant dans votre fichier YAML **ClusterLogForwarder** CR.

```
outputDefaults:
 elasticsearch:
    structuredTypeKey: openshift.labels.myLabel 1
    structuredTypeName: nologformat
pipelines:
 - name: application-logs
   inputRefs:
   - application
   - audit
   outputRefs:
   - elasticsearch-secure
   - default
   parse: json
   labels:
     myLabel: myValue 2
```

1 Utilise la valeur de la paire clé-valeur formée par le label OpenShift **myLabel**.

2 L'élément **myLabel** donne sa valeur de chaîne, **myValue**, à l'enregistrement structuré.

Dans ce cas, l'enregistrement structuré suivant est placé dans l'index **app-myValue-write**:

```
{
  "structured":{"name":"fred","home":"bedrock"},
  "openshift":{"labels":{"myLabel": "myValue", ...}}
}
```

Autres considérations

- Le site Elasticsearch *index* pour les enregistrements structurés est formé en ajoutant "app-" au type structuré et en ajoutant "-write".
- Les enregistrements non structurés ne sont pas envoyés à l'index structuré. Ils sont indexés comme d'habitude dans les index d'application, d'infrastructure ou d'audit.
- S'il n'y a pas de type structuré non vide, transmettre un enregistrement *unstructured* sans champ **structured**.

Il est important de ne pas surcharger Elasticsearch avec un trop grand nombre d'indices. N'utilisez que des types structurés distincts pour des journaux distincts *formats*, **not** pour chaque application ou espace de noms. Par exemple, la plupart des applications Apache utilisent le même format de journal JSON et le même type structuré, comme **LogApache**.

11.3. TRANSFÉRER LES JOURNAUX JSON VERS LE MAGASIN DE JOURNAUX ELASTICSEARCH

Pour un magasin de journaux Elasticsearch, si vos entrées de journaux JSON *follow different schemas*, configurez la ressource personnalisée (CR) **ClusterLogForwarder** pour regrouper chaque schéma JSON dans une seule définition de sortie. De cette façon, Elasticsearch utilise un index distinct pour chaque schéma.

IMPORTANT

Étant donné que le transfert de différents schémas vers le même index peut entraîner des conflits de type et des problèmes de cardinalité, vous devez effectuer cette configuration avant de transférer des données vers le magasin Elasticsearch.

Pour éviter les problèmes de performance liés à un trop grand nombre d'indices, il convient de limiter le nombre de schémas possibles en adoptant des schémas communs.

Procédure

1. Ajoutez l'extrait suivant à votre fichier YAML **ClusterLogForwarder** CR.

```
outputDefaults:
 elasticsearch:
   structuredTypeKey: <log record field>
   structuredTypeName: <name>
pipelines:
- inputRefs:
```

```
    - application
    outputRefs: default
    parse: json
```

2. Facultatif : Utilisez **structuredTypeKey** pour spécifier l'un des champs de l'enregistrement du journal, comme décrit dans la rubrique précédente, Configuration des données de journal JSON pour Elasticsearch. Sinon, supprimez cette ligne.

3. Facultatif : Utilisez **structuredTypeName** pour spécifier un **<name>**, comme décrit dans la rubrique précédente, Configuration des données de journalisation JSON pour Elasticsearch . Sinon, supprimez cette ligne.

IMPORTANT

Pour analyser les journaux JSON, vous devez définir soit **structuredTypeKey** ou **structuredTypeName**, soit **structuredTypeKey** et **structuredTypeName**.

4. Pour **inputRefs**, spécifiez les types de journaux à transférer en utilisant ce pipeline, par exemple **application, infrastructure** , ou **audit**.

5. Ajouter l'élément **parse: json** aux pipelines.

6. Créer l'objet CR :

```
oc create -f <nom-de-fichier>.yaml
```

Le Red Hat OpenShift Logging Operator redéploie les pods Fluentd. Cependant, s'ils ne se redéploient pas, supprimez les pods Fluentd pour les forcer à se redéployer.

```
$ oc delete pod --selector logging-infra=collector
```

Ressources complémentaires

- Transmission des journaux à des systèmes tiers

CHAPITRE 12. COLLECTE ET STOCKAGE DES ÉVÉNEMENTS KUBERNETES

L'OpenShift Container Platform Event Router est un pod qui observe les événements Kubernetes et les enregistre pour qu'ils soient collectés par le sous-système de journalisation. Vous devez déployer manuellement l'Event Router.

Le routeur d'événements collecte les événements de tous les projets et les écrit sur **STDOUT**. Le collecteur transmet ensuite ces événements au magasin défini dans la ressource personnalisée (CR) **ClusterLogForwarder**.

IMPORTANT

Le routeur d'événements ajoute une charge supplémentaire à Fluentd et peut avoir un impact sur le nombre d'autres messages de journaux qui peuvent être traités.

12.1. DÉPLOIEMENT ET CONFIGURATION DE L'EVENT ROUTER

Suivez les étapes suivantes pour déployer l'Event Router dans votre cluster. Vous devez toujours déployer le routeur d'événements dans le projet **openshift-logging** afin de vous assurer qu'il collecte les événements de l'ensemble du cluster.

L'objet Template suivant crée le compte de service, le rôle de cluster et la liaison de rôle de cluster requis pour le routeur d'événements. Le modèle configure et déploie également le pod Event Router. Vous pouvez utiliser ce modèle sans y apporter de modifications ou modifier les demandes de CPU et de mémoire de l'objet de déploiement.

Conditions préalables

- Vous devez disposer des autorisations appropriées pour créer des comptes de service et mettre à jour les liaisons de rôles de cluster. Par exemple, vous pouvez exécuter le modèle suivant avec un utilisateur ayant le rôle **cluster-admin**.
- Le sous-système de journalisation pour Red Hat OpenShift doit être installé.

Procédure

1. Créer un modèle pour le routeur d'événements :

```
kind: Template
apiVersion: template.openshift.io/v1
metadata:
  name: eventrouter-template
  annotations:
    description: "A pod forwarding kubernetes events to OpenShift Logging stack."
    tags: "events,EFK,logging,cluster-logging"
objects:
  - kind: ServiceAccount 1
    apiVersion: v1
    metadata:
      name: eventrouter
      namespace: ${NAMESPACE}
  - kind: ClusterRole 2
    apiVersion: rbac.authorization.k8s.io/v1
```

```
  metadata:
    name: event-reader
  rules:
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["get", "watch", "list"]
- kind: ClusterRoleBinding (3)
  apiVersion: rbac.authorization.k8s.io/v1
  metadata:
    name: event-reader-binding
  subjects:
  - kind: ServiceAccount
    name: eventrouter
    namespace: ${NAMESPACE}
  roleRef:
    kind: ClusterRole
    name: event-reader
- kind: ConfigMap (4)
  apiVersion: v1
  metadata:
    name: eventrouter
    namespace: ${NAMESPACE}
  data:
    config.json: |-
      {
        "sink": "stdout"
      }
- kind: Deployment (5)
  apiVersion: apps/v1
  metadata:
    name: eventrouter
    namespace: ${NAMESPACE}
    labels:
      component: "eventrouter"
      logging-infra: "eventrouter"
      provider: "openshift"
  spec:
    selector:
      matchLabels:
        component: "eventrouter"
        logging-infra: "eventrouter"
        provider: "openshift"
    replicas: 1
    template:
      metadata:
        labels:
          component: "eventrouter"
          logging-infra: "eventrouter"
          provider: "openshift"
        name: eventrouter
      spec:
        serviceAccount: eventrouter
        containers:
          - name: kube-eventrouter
            image: ${IMAGE}
            imagePullPolicy: IfNotPresent
```

```
            resources:
              requests:
                cpu: ${CPU}
                memory: ${MEMORY}
            volumeMounts:
            - name: config-volume
              mountPath: /etc/eventrouter
          volumes:
            - name: config-volume
              configMap:
                name: eventrouter
parameters:
  - name: IMAGE 6
    displayName: Image
    value: "registry.redhat.io/openshift-logging/eventrouter-rhel8:v0.4"
  - name: CPU 7
    displayName: CPU
    value: "100m"
  - name: MEMORY 8
    displayName: Memory
    value: "128Mi"
  - name: NAMESPACE
    displayName: Namespace
    value: "openshift-logging" 9
```

1. Crée un compte de service dans le projet **openshift-logging** pour le routeur d'événements.
2. Crée un ClusterRole pour surveiller les événements dans le cluster.
3. Crée un ClusterRoleBinding pour lier le ClusterRole au compte de service.
4. Crée une carte de configuration dans le projet **openshift-logging** pour générer le fichier **config.json** requis.
5. Crée un déploiement dans le projet **openshift-logging** pour générer et configurer le pod Event Router.
6. Spécifie l'image, identifiée par une balise telle que **v0.4**.
7. Spécifie la quantité minimale de CPU à allouer au pod Event Router. La valeur par défaut est **100m**.
8. Spécifie la quantité minimale de mémoire à allouer au pod Event Router. La valeur par défaut est **128Mi**.
9. Spécifie le projet **openshift-logging** dans lequel les objets doivent être installés.

2. Utilisez la commande suivante pour traiter et appliquer le modèle :

```
oc process -f <templatefile> | oc apply -n openshift-logging -f -
```

Par exemple :

```
$ oc process -f eventrouter.yaml | oc apply -n openshift-logging -f -
```

-

Exemple de sortie

```
serviceaccount/eventrouter created
clusterrole.authorization.openshift.io/event-reader created
clusterrolebinding.authorization.openshift.io/event-reader-binding created
configmap/eventrouter created
deployment.apps/eventrouter created
```

3. Validez que l'Event Router est installé dans le projet **openshift-logging**:

   a. Voir le nouveau pod Event Router :

   ```
   $ oc get pods --selector  component=eventrouter -o name -n openshift-logging
   ```

   Exemple de sortie

   ```
   pod/cluster-logging-eventrouter-d649f97c8-qvv8r
   ```

   b. Affichez les événements collectés par le routeur d'événements :

   ```
   oc logs <cluster_logging_eventrouter_pod> -n openshift-logging
   ```

   Par exemple :

   ```
   $ oc logs cluster-logging-eventrouter-d649f97c8-qvv8r -n openshift-logging
   ```

   Exemple de sortie

   ```
   {"verb":"ADDED","event":{"metadata":{"name":"openshift-service-catalog-controller-
   manager-remover.1632d931e88fcd8f","namespace":"openshift-service-catalog-
   removed","selfLink":"/api/v1/namespaces/openshift-service-catalog-
   removed/events/openshift-service-catalog-controller-manager-
   remover.1632d931e88fcd8f","uid":"787d7b26-3d2f-4017-b0b0-
   420db4ae62c0","resourceVersion":"21399","creationTimestamp":"2020-09-
   08T15:40:26Z"},"involvedObject":{"kind":"Job","namespace":"openshift-service-catalog-
   removed","name":"openshift-service-catalog-controller-manager-
   remover","uid":"fac9f479-4ad5-4a57-8adc-
   cb25d3d9cf8f","apiVersion":"batch/v1","resourceVersion":"21280"},"reason":"Completed","
   message":"Job completed","source":{"component":"job-
   controller"},"firstTimestamp":"2020-09-08T15:40:26Z","lastTimestamp":"2020-09-
   08T15:40:26Z","count":1,"type":"Normal"}}
   ```

   Vous pouvez également utiliser Kibana pour afficher les événements en créant un modèle d'index à l'aide de l'index Elasticsearch **infra**.

CHAPITRE 13. MISE À JOUR DE LA JOURNALISATION D'OPENSHIFT

13.1. VERSIONS PRISES EN CHARGE

Pour des informations sur la compatibilité des versions et l'assistance, voir Red Hat OpenShift Container Platform Life Cycle Policy (Politique de cycle de vie de Red Hat OpenShift Container Platform)

Pour passer de la journalisation de cluster dans OpenShift Container Platform version 4.6 et antérieures à OpenShift Logging 5.x, vous mettez à jour le cluster OpenShift Container Platform à la version 4.7 ou 4.8. Ensuite, vous mettez à jour les opérateurs suivants :

- De Elasticsearch Operator 4.x à OpenShift Elasticsearch Operator 5.x
- De Cluster Logging Operator 4.x à Red Hat OpenShift Logging Operator 5.x

Pour passer d'une version précédente d'OpenShift Logging à la version actuelle, vous mettez à jour OpenShift Elasticsearch Operator et Red Hat OpenShift Logging Operator vers leurs versions actuelles.

13.2. MISE À JOUR ENREGISTREMENT DE LA VERSION ACTUELLE

Pour mettre à jour Logging à la version actuelle, vous changez les abonnements pour OpenShift Elasticsearch Operator et Red Hat OpenShift Logging Operator.

IMPORTANT

Vous devez mettre à jour l'OpenShift Elasticsearch Operator *before* vous mettez à jour le Red Hat OpenShift Logging Operator. Vous devez également mettre à jour les opérateurs *both* à la même version.

Si vous mettez à jour les opérateurs dans le mauvais ordre, Kibana n'est pas mis à jour et la ressource personnalisée (CR) Kibana n'est pas créée. Pour contourner ce problème, vous supprimez le pod Red Hat OpenShift Logging Operator. Lorsque le pod Red Hat OpenShift Logging Operator se redéploie, il crée la CR Kibana et Kibana redevient disponible.

Conditions préalables

- La version d'OpenShift Container Platform est 4.7 ou ultérieure.
- L'état de la journalisation est sain :
  - Toutes les cosses sont **ready**.
  - Le cluster Elasticsearch est sain.
- Vos données Elasticsearch et Kibana sont sauvegardées .

Procédure

1. Mettre à jour l'opérateur OpenShift Elasticsearch :
   a. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → Installed Operators**.

b. Sélectionnez le projet **openshift-Operators-redhat**.

c. Cliquez sur le site **OpenShift Elasticsearch Operator**.

d. Cliquez sur **Subscription → Channel**.

e. Dans la fenêtre **Change Subscription Update Channel**, sélectionnez **stable-5.x** et cliquez sur **Save**.

f. Attendez quelques secondes, puis cliquez sur **Operators → Installed Operators**.

g. Vérifiez que la version d'OpenShift Elasticsearch Operator est 5.x.x.

h. Attendez que le champ **Status** indique **Succeeded**.

2. Mettez à jour l'opérateur de journalisation de Red Hat OpenShift :

a. Dans la console web d'OpenShift Container Platform, cliquez sur **Operators → Installed Operators**.

b. Sélectionnez le projet **openshift-logging**.

c. Cliquez sur le site **Red Hat OpenShift Logging Operator**.

d. Cliquez sur **Subscription → Channel**.

e. Dans la fenêtre **Change Subscription Update Channel**, sélectionnez **stable-5.x** et cliquez sur **Save**.

f. Attendez quelques secondes, puis cliquez sur **Operators → Installed Operators**.

g. Vérifiez que la version de Red Hat OpenShift Logging Operator est 5.y.z

h. Attendez que le champ **Status** indique **Succeeded**.

3. Vérifier les composants de journalisation :

a. Assurez-vous que tous les pods Elasticsearch sont dans l'état **Ready**:

```
$ oc get pod -n openshift-logging --selector component=elasticsearch
```

Exemple de sortie

```
NAME                                            READY   STATUS    RESTARTS   AGE
elasticsearch-cdm-1pbrl44l-1-55b7546f4c-mshhk   2/2     Running   0          31m
elasticsearch-cdm-1pbrl44l-2-5c6d87589f-gx5hk   2/2     Running   0          30m
elasticsearch-cdm-1pbrl44l-3-88df5d47-m45jc     2/2     Running   0          29m
```

b. Assurez-vous que le cluster Elasticsearch est sain :

```
$ oc exec -n openshift-logging -c elasticsearch elasticsearch-cdm-1pbrl44l-1-
55b7546f4c-mshhk -- health
```

```
{
  "cluster_name" : "elasticsearch",
  "status" : "green",
```

```
}
```

c. Assurez-vous que les tâches cron d'Elasticsearch sont créées :

```
$ oc project openshift-logging
```

```
$ oc get cronjob
```

```
NAME                     SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
elasticsearch-im-app     */15 * * * *   False     0        <none>          56s
elasticsearch-im-audit   */15 * * * *   False     0        <none>          56s
elasticsearch-im-infra   */15 * * * *   False     0        <none>          56s
```

d. Vérifiez que le magasin de logs est mis à jour à la version 5.x et que les index sont **green**:

```
oc exec -c elasticsearch <any_es_pod_in_the_cluster> -- indices
```

e. Vérifiez que la sortie comprend les indices **app-00000x**, **infra-00000x**, **audit-00000x**, **.security**.

Exemple 13.1. Exemple de sortie avec les indices dans un état vert

```
Tue Jun 30 14:30:54 UTC 2020
health status index                                                    uuid                   pri rep
docs.count docs.deleted store.size pri.store.size
green  open   infra-000008
bnBvUFEXTWi92z3zWAzieQ   3 1      222195            0       289          144
green  open   infra-000004
rtDSzoqsSl6saisSK7Au1Q   3 1      226717            0       297          148
green  open   infra-000012
RSf_kUwDSR2xEuKRZMPqZQ   3 1      227623            0       295          147
green  open   .kibana_7
1SJdCqlZTPWlIAaOUd78yg   1 1          4            0         0            0
green  open   infra-000010
iXwL3bnqTuGEABbUDa6OVw   3 1      248368            0       317          158
green  open   infra-000009
YN9EsULWSNaxWeeNvOs0RA   3 1      258799            0       337          168
green  open   infra-000014
YP0U6R7FQ_GVQVQZ6Yh9Ig   3 1      223788            0       292          146
green  open   infra-000015
JRBbAbEmSMqK5X40df9HbQ   3 1      224371            0       291          145
green  open   .orphaned.2020.06.30
n_xQC2dWQzConkvQqei3YA   3 1          9            0         0            0
green  open   infra-000007
llkkAVSzSOmosWTSAJM_hg   3 1      228584            0       296          148
green  open   infra-000005
d9BoGQdiQASsS3BBFm2iRA   3 1      227987            0       297          148
green  open   infra-000003                                              1-
goREK1QUKlQPAIVkWVaQ   3 1      226719            0       295          147
green  open   .security
zeT65uOuRTKZMjg_bbUc1g   1 1          5            0         0            0
green  open   .kibana-377444158_kubeadmin                               wvMhDwJkR-
mRZQO84K0gUQ   3 1          1            0         0            0
green  open   infra-000006                                              5H-
```

```
KBSXGQKiO7hdapDE23g 3 1 226676 0 295 147
green open infra-000001 eH53BQ-
bSxSWR5xYZB6lVg 3 1 341800 0 443 220
green open .kibana-6
RVp7TemSSemGJcsSUmuf3A 1 1 4 0 0 0
green open infra-000011
J7XWBauWSTe0jnzX02fU6A 3 1 226100 0 293 146
green open app-000001
axSAFfONQDmKwatkjPXdtw 3 1 103186 0 126 57
green open infra-000016
m9c1iRLtStWSF1GopaRyCg 3 1 13685 0 19 9
green open infra-000002 Hz6WvINtTvKcQzw-
ewmbYg 3 1 228994 0 296 148
green open infra-000013 KR9mMFUpQl-
jraYtanyIGw 3 1 228166 0 298 148
green open audit-000001
eERqLdLmQOiQDFES1LBATQ 3 1 0 0 0 0
```

f. Vérifiez que le collecteur de journaux est mis à jour :

```
$ oc get ds collector -o json | grep collector
```

g. Vérifiez que la sortie comprend un conteneur **collectort**:

```
"containerName": "collector"
```

h. Vérifiez que le visualiseur de logs est mis à jour vers la version 5.x en utilisant le CRD Kibana :

```
$ oc get kibana kibana -o json
```

i. Vérifiez que la sortie inclut un pod Kibana avec le statut **ready**:

Exemple 13.2. Exemple de sortie avec un pod Kibana prêt

```
[
{
"clusterCondition": {
"kibana-5fdd766ffd-nb2jj": [
{
"lastTransitionTime": "2020-06-30T14:11:07Z",
"reason": "ContainerCreating",
"status": "True",
"type": ""
},
{
"lastTransitionTime": "2020-06-30T14:11:07Z",
"reason": "ContainerCreating",
"status": "True",
"type": ""
}
]
},
"deployment": "kibana",
"pods": {
```

```
"failed": [],
"notReady": []
"ready": []
},
"replicaSets": [
"kibana-5fdd766ffd"
],
"replicas": 1
}
]
```

CHAPITRE 14. VISUALISATION DES TABLEAUX DE BORD DES CLUSTERS

Les tableaux de bord **Logging/Elasticsearch Nodes** et **Openshift Logging** de la console web OpenShift Container Platform contiennent des détails détaillés sur votre instance Elasticsearch et les nœuds Elasticsearch individuels que vous pouvez utiliser pour prévenir et diagnostiquer les problèmes.

Le tableau de bord **OpenShift Logging** contient des graphiques qui montrent les détails de votre instance Elasticsearch au niveau du cluster, y compris les ressources du cluster, le ramassage des ordures, les shards dans le cluster et les statistiques Fluentd.

Le tableau de bord **Logging/Elasticsearch Nodes** contient des graphiques qui montrent des détails sur votre instance Elasticsearch, souvent au niveau du nœud, y compris des détails sur l'indexation, les shards, les ressources, etc.

14.1. ACCÈS AUX TABLEAUX DE BORD ELASTICSEARCH ET OPENSHIFT LOGGING

Vous pouvez consulter les tableaux de bord **Logging/Elasticsearch Nodes** et **OpenShift Logging** dans la console web de OpenShift Container Platform.

Procédure

Pour lancer les tableaux de bord :

1. Dans la console web d'OpenShift Container Platform, cliquez sur **Observe → Dashboards**.
2. Sur la page **Dashboards**, sélectionnez **Logging/Elasticsearch Nodes** ou **OpenShift Logging** dans le menu **Dashboard**.
   Pour le tableau de bord **Logging/Elasticsearch Nodes**, vous pouvez sélectionner le nœud Elasticsearch que vous souhaitez visualiser et définir la résolution des données.

   Le tableau de bord approprié s'affiche et présente plusieurs graphiques de données.
3. En option : Sélectionnez une plage de temps différente à afficher ou un taux de rafraîchissement des données dans les menus **Time Range** et **Refresh Interval**.

Pour plus d'informations sur les tableaux de bord, voir À propos du tableau de bord OpenShift Logging et À propos du tableau de bord Logging/Elastisearch Nodes .

14.2. A PROPOS DU TABLEAU DE BORD OPENSHIFT LOGGING

Le tableau de bord **OpenShift Logging** contient des graphiques qui présentent des détails sur votre instance Elasticsearch au niveau du cluster, que vous pouvez utiliser pour diagnostiquer et anticiper les problèmes.

Tableau 14.1. Graphiques de journalisation OpenShift

Métrique	Description

Métrique	Description
Statut de la grappe élastique	L'état actuel d'Elasticsearch : • ONLINE - Indique que l'instance Elasticsearch est en ligne. • OFFLINE - Indique que l'instance Elasticsearch est hors ligne.
Nœuds élastiques	Le nombre total de nœuds Elasticsearch dans l'instance Elasticsearch.
Éclats élastiques	Le nombre total de shards Elasticsearch dans l'instance Elasticsearch.
Documents élastiques	Le nombre total de documents Elasticsearch dans l'instance Elasticsearch.
Taille totale de l'index sur le disque	L'espace disque total utilisé pour les index Elasticsearch.
Tâches en attente élastiques	Le nombre total de modifications Elasticsearch qui n'ont pas été achevées, telles que la création d'index, le mappage d'index, l'allocation de shard ou la défaillance de shard.
Temps de GC de la JVM élastique	Temps passé par la JVM à exécuter les opérations de collecte des déchets d'Elasticsearch dans le cluster.
Taux de GC de la JVM élastique	Nombre total de fois où la JVM a exécuté des activités de nettoyage par seconde.
Somme des temps de latence Elastic Query/Fetch	• Temps de latence de la requête : Le temps moyen d'exécution de chaque requête de recherche Elasticsearch. • Latence d'extraction : Le temps moyen que chaque requête de recherche Elasticsearch passe à récupérer des données. Le temps de latence de la recherche est généralement plus court que le temps de latence de la requête. Si la latence d'extraction augmente constamment, cela peut indiquer des disques lents, un enrichissement des données ou des requêtes volumineuses avec trop de résultats.

Métrique	Description
Taux d'interrogation élastique	Le nombre total de requêtes exécutées contre l'instance Elasticsearch par seconde pour chaque nœud Elasticsearch.
UNITÉ CENTRALE	La quantité de CPU utilisée par Elasticsearch, Fluentd et Kibana, indiquée pour chaque composant.
Heap de la JVM élastique utilisé	La quantité de mémoire JVM utilisée. Dans un cluster en bonne santé, le graphique montre des baisses régulières au fur et à mesure que la mémoire est libérée par le garbage collection de la JVM.
Utilisation du disque Elasticsearch	L'espace disque total utilisé par l'instance Elasticsearch pour chaque nœud Elasticsearch.
Descripteurs de fichiers utilisés	Le nombre total de descripteurs de fichiers utilisés par Elasticsearch, Fluentd et Kibana.
Compteur d'émission FluentD	Le nombre total de messages Fluentd par seconde pour la sortie par défaut de Fluentd, et le nombre de tentatives pour la sortie par défaut.
Disponibilité des tampons FluentD	Le pourcentage de la mémoire tampon de Fluentd qui est disponible pour les chunks. Un tampon plein peut indiquer que Fluentd n'est pas capable de traiter le nombre de logs reçus.
Octets rx élastiques	Le nombre total d'octets qu'Elasticsearch a reçu de FluentD, des nœuds Elasticsearch et d'autres sources.
Taux de défaillance de l'indice élastique	Le nombre total de fois par seconde qu'un index Elasticsearch échoue. Un taux élevé peut indiquer un problème d'indexation.
Taux d'erreur en sortie de FluentD	Le nombre total de fois par seconde où FluentD n'est pas capable de sortir des logs.

14.3. GRAPHIQUES DANS LE TABLEAU DE BORD LOGGING/ELASTICSEARCH NODES

Le tableau de bord **Logging/Elasticsearch Nodes** contient des graphiques qui montrent des détails sur votre instance Elasticsearch, souvent au niveau du nœud, pour des diagnostics plus poussés.

État d'Elasticsearch

Le tableau de bord **Logging/Elasticsearch Nodes** contient les graphiques suivants sur l'état de votre instance Elasticsearch.

Tableau 14.2. Champs d'état Elasticsearch

Métrique	Description
Statut de la grappe	L'état de santé du cluster pendant la période sélectionnée, en utilisant les états vert, jaune et rouge d'Elasticsearch : • 0 - Indique que l'instance Elasticsearch est en statut vert, ce qui signifie que tous les shards sont alloués. • 1 - Indique que l'instance Elasticsearch est en état jaune, ce qui signifie que les répliques d'au moins un nuage ne sont pas allouées. • 2 - Indique que l'instance Elasticsearch est en état rouge, ce qui signifie qu'au moins un shard primaire et ses répliques ne sont pas alloués.
Nœuds de la grappe	Le nombre total de nœuds Elasticsearch dans le cluster.
Nœuds de données en grappe	Nombre de nœuds de données Elasticsearch dans le cluster.
Tâches en suspens du cluster	Nombre de modifications de l'état de la grappe qui ne sont pas terminées et qui sont en attente dans une file d'attente de la grappe, par exemple, la création d'index, la suppression d'index ou l'allocation de la grappe. Une tendance à la hausse indique que le cluster n'est pas en mesure de suivre les changements.

État de l'index du cluster Elasticsearch

Chaque index Elasticsearch est un groupe logique d'un ou plusieurs shards, qui sont des unités de base de données persistantes. Il existe deux types d'index : les shards primaires et les shards répliqués. Lorsqu'un document est indexé dans un index, il est stocké dans l'un de ses disques primaires et copié dans chaque réplique de ce disque. Le nombre de disques primaires est spécifié lors de la création de l'index et ne peut pas être modifié pendant la durée de vie de l'index. Vous pouvez modifier le nombre de répliques à tout moment.

Le groupe de stockage d'index peut se trouver dans plusieurs états en fonction de la phase de son cycle de vie ou des événements qui se produisent dans le cluster. Lorsque le groupe de stockage est en mesure d'exécuter des requêtes de recherche et d'indexation, il est actif. S'il ne peut pas exécuter ces requêtes, il est inactif. Un groupe de stockage peut être inactif s'il est en cours d'initialisation, de réaffectation, de désaffectation, etc.

Les nuages d'index sont constitués d'un certain nombre de blocs internes plus petits, appelés segments d'index, qui sont des représentations physiques des données. Un segment d'index est un index Lucene relativement petit et immuable qui est créé lorsque Lucene enregistre des données nouvellement indexées. Lucene, une bibliothèque de recherche utilisée par Elasticsearch, fusionne les segments

d'index en segments plus grands en arrière-plan pour maintenir le nombre total de segments à un niveau bas. Si le processus de fusion des segments est plus lent que la vitesse de création de nouveaux segments, cela peut indiquer un problème.

Lorsque Lucene effectue des opérations de données, telles qu'une opération de recherche, il effectue l'opération sur les segments d'index dans l'index concerné. À cette fin, chaque segment contient des structures de données spécifiques qui sont chargées dans la mémoire et mappées. Le mappage de l'index peut avoir un impact significatif sur la mémoire utilisée par les structures de données des segments.

Le tableau de bord **Logging/Elasticsearch Nodes** contient les diagrammes suivants sur les cartes d'index Elasticsearch.

Tableau 14.3. Graphiques de l'état de la grappe Elasticsearch

Métrique	Description
Clusters actifs	Le nombre d'unités primaires actives et le nombre total d'unités, y compris les répliques, dans le cluster. Si le nombre d'unités augmente, les performances du cluster peuvent commencer à se dégrader.
Initialisation du cluster	Le nombre de disques non actifs dans le cluster. Un disque non actif est un disque en cours d'initialisation, en cours de réaffectation à un autre nœud ou non affecté. En règle générale, un cluster a des cartes non actives pendant de courtes périodes. Un nombre croissant de cartes non actives sur de longues périodes peut être le signe d'un problème.
Cluster relocalisant des shards	Le nombre de shards qu'Elasticsearch est en train de relocaliser sur un nouveau nœud. Elasticsearch relocalise les nœuds pour de multiples raisons, comme une utilisation élevée de la mémoire sur un nœud ou après l'ajout d'un nouveau nœud au cluster.
Cluster d'ombres non assignées	Le nombre de shards non assignés. Les cartes Elasticsearch peuvent être non attribuées pour des raisons telles que l'ajout d'un nouvel index ou la défaillance d'un nœud.

Métriques des nœuds Elasticsearch

Chaque nœud d'Elasticsearch dispose d'une quantité finie de ressources qui peuvent être utilisées pour traiter les tâches. Lorsque toutes les ressources sont utilisées et qu'Elasticsearch tente d'exécuter une nouvelle tâche, Elasticsearch place les tâches dans une file d'attente jusqu'à ce que des ressources soient disponibles.

Le tableau de bord **Logging/Elasticsearch Nodes** contient les graphiques suivants sur l'utilisation des ressources pour un nœud sélectionné et le nombre de tâches en attente dans la file d'attente Elasticsearch.

Tableau 14.4. Graphiques métriques des nœuds Elasticsearch

Métrique	Description
Tâches du ThreadPool	Nombre de tâches en attente dans les différentes files d'attente, indiqué par type de tâche. Une accumulation prolongée de tâches dans une file d'attente peut indiquer une pénurie de ressources du nœud ou un autre problème.
Utilisation de l'unité centrale	La quantité de CPU utilisée par le nœud Elasticsearch sélectionné en pourcentage de la CPU totale allouée au conteneur hôte.
Utilisation de la mémoire	La quantité de mémoire utilisée par le nœud Elasticsearch sélectionné.
Utilisation du disque	L'espace disque total utilisé pour les données d'index et les métadonnées sur le nœud Elasticsearch sélectionné.
Taux d'indexation des documents	Le taux d'indexation des documents sur le nœud Elasticsearch sélectionné.
Temps de latence de l'indexation	Le temps nécessaire pour indexer les documents sur le nœud Elasticsearch sélectionné. La latence d'indexation peut être affectée par de nombreux facteurs, tels que la mémoire Heap de la JVM et la charge globale. Une latence croissante indique un manque de capacité de ressources dans l'instance.
Taux de recherche	Le nombre de requêtes de recherche exécutées sur le nœud Elasticsearch sélectionné.
Temps de latence de la recherche	Le temps nécessaire pour compléter les requêtes de recherche sur le nœud Elasticsearch sélectionné. La latence de la recherche peut être affectée par de nombreux facteurs. Une latence croissante indique un manque de capacité des ressources dans l'instance.
Nombre de documents (avec répliques)	Le nombre de documents Elasticsearch stockés sur le nœud Elasticsearch sélectionné, y compris les documents stockés dans les shards primaires et les shards répliqués qui sont alloués sur le nœud.
Taux de suppression des documents	Nombre de documents Elasticsearch supprimés de l'une des zones d'index attribuées au nœud Elasticsearch sélectionné.
Taux de fusion des documents	Le nombre de documents Elasticsearch en cours de fusion dans l'une des zones d'indexation allouées au nœud Elasticsearch sélectionné.

Nœud Elasticsearch fielddata

Fielddata est une structure de données Elasticsearch qui contient des listes de termes dans un index et qui est conservée dans le tas de la JVM. La construction des données de champ étant une opération coûteuse, Elasticsearch met en cache les structures de données de champ. Elasticsearch peut expulser un cache de données de terrain lorsque le segment d'index sous-jacent est supprimé ou fusionné, ou s'il n'y a pas assez de mémoire JVM HEAP pour tous les caches de données de terrain.

Le tableau de bord **Logging/Elasticsearch Nodes** contient les graphiques suivants sur les données de terrain Elasticsearch.

Tableau 14.5. Graphiques des données de terrain du nœud Elasticsearch

Métrique	Description
Taille de la mémoire des données de terrain	La quantité de JVM Heap utilisée pour le cache des données de terrain sur le nœud Elasticsearch sélectionné.
Evictions des données de terrain	Nombre de structures de données de terrain supprimées du nœud Elasticsearch sélectionné.

Cache de requête du nœud Elasticsearch

Si les données stockées dans l'index ne changent pas, les résultats de la recherche sont mis en cache dans un cache de requête au niveau du nœud pour être réutilisés par Elasticsearch.

Le tableau de bord **Logging/Elasticsearch Nodes** contient les graphiques suivants sur le cache de requête du nœud Elasticsearch.

Tableau 14.6. Graphiques de requêtes de nœuds Elasticsearch

Métrique	Description
Taille du cache des requêtes	La quantité totale de mémoire utilisée pour le cache de requête pour tous les shards alloués au nœud Elasticsearch sélectionné.
Evictions du cache des requêtes	Le nombre d'évictions du cache des requêtes sur le nœud Elasticsearch sélectionné.
Nombre d'occurrences dans le cache des requêtes	Le nombre d'occurrences du cache de requête sur le nœud Elasticsearch sélectionné.
Manquements au cache des requêtes	Le nombre d'échecs du cache de requête sur le nœud Elasticsearch sélectionné.

Gestion de l'index Elasticsearch

Lors de l'indexation des documents, Elasticsearch stocke les documents dans des segments d'index, qui sont des représentations physiques des données. Dans le même temps, Elasticsearch fusionne périodiquement des segments plus petits en un segment plus grand afin d'optimiser l'utilisation des ressources. Si l'indexation est plus rapide que la capacité à fusionner les segments, le processus de

fusion ne s'achève pas assez rapidement, ce qui peut entraîner des problèmes de recherche et de performance. Pour éviter cette situation, Elasticsearch limite l'indexation, généralement en réduisant le nombre de threads alloués à l'indexation à un seul thread.

Le tableau de bord **Logging/Elasticsearch Nodes** contient les graphiques suivants sur l'étranglement des index Elasticsearch.

Tableau 14.7. Graphiques d'étranglement de l'indice

Métrique	Description
Indexation de l'étranglement	Durée pendant laquelle Elasticsearch a limité les opérations d'indexation sur le nœud Elasticsearch sélectionné.
Fusionner l'étranglement	Durée pendant laquelle Elasticsearch a limité les opérations de fusion de segments sur le nœud Elasticsearch sélectionné.

Statistiques sur le tas de la JVM Node

Le tableau de bord **Logging/Elasticsearch Nodes** contient les graphiques suivants sur les opérations de la JVM Heap.

Tableau 14.8. Graphiques statistiques de la JVM Heap

Métrique	Description
Tas utilisé	Le montant de l'espace total alloué à la JVM Heap qui est utilisé sur le nœud Elasticsearch sélectionné.
Nombre de GC	Le nombre d'opérations de collecte de déchets qui ont été exécutées sur le nœud Elasticsearch sélectionné, par ancienne et jeune collecte de déchets.
Temps GC	Le temps que la JVM a passé à exécuter des opérations de collecte de déchets sur le nœud Elasticsearch sélectionné, par ancienne et jeune collecte de déchets.

CHAPITRE 15. DÉPANNAGE JOURNALISATION

15.1. VISUALISATION DE L'ÉTAT DE LA JOURNALISATION D'OPENSHIFT

Vous pouvez afficher l'état de l'opérateur de journalisation de Red Hat OpenShift et d'un certain nombre de composants du sous-système de journalisation.

15.1.1. Afficher l'état de l'opérateur de journalisation de Red Hat OpenShift

Vous pouvez afficher l'état de votre opérateur de journalisation Red Hat OpenShift.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Modification du projet **openshift-logging**.

```
$ oc project openshift-logging
```

2. Pour afficher l'état de la journalisation d'OpenShift :

   a. Obtenir l'état de la journalisation d'OpenShift :

```
$ oc get clusterlogging instance -o yaml
```

Exemple de sortie

```
apiVersion: logging.openshift.io/v1
kind: ClusterLogging

....

status: 1
  collection:
    logs:
      fluentdStatus:
        daemonSet: fluentd 2
        nodes:
          fluentd-2rhqp: ip-10-0-169-13.ec2.internal
          fluentd-6fgjh: ip-10-0-165-244.ec2.internal
          fluentd-6l2ff: ip-10-0-128-218.ec2.internal
          fluentd-54nx5: ip-10-0-139-30.ec2.internal
          fluentd-flpnn: ip-10-0-147-228.ec2.internal
          fluentd-n2frh: ip-10-0-157-45.ec2.internal
        pods:
          failed: []
          notReady: []
          ready:
          - fluentd-2rhqp
          - fluentd-54nx5
```

```
        - fluentd-6fgjh
        - fluentd-6l2ff
        - fluentd-flpnn
        - fluentd-n2frh
    logstore: 3
      elasticsearchStatus:
      - ShardAllocationEnabled:  all
        cluster:
          activePrimaryShards:    5
          activeShards:           5
          initializingShards:     0
          numDataNodes:           1
          numNodes:               1
          pendingTasks:           0
          relocatingShards:       0
          status:                 green
          unassignedShards:       0
        clusterName:              elasticsearch
        nodeConditions:
          elasticsearch-cdm-mkkdys93-1:
        nodeCount:  1
        pods:
          client:
            failed:
            notReady:
            ready:
            - elasticsearch-cdm-mkkdys93-1-7f7c6-mjm7c
          data:
            failed:
            notReady:
            ready:
            - elasticsearch-cdm-mkkdys93-1-7f7c6-mjm7c
          master:
            failed:
            notReady:
            ready:
            - elasticsearch-cdm-mkkdys93-1-7f7c6-mjm7c
    visualization:  4
      kibanaStatus:
      - deployment: kibana
        pods:
          failed: []
          notReady: []
          ready:
          - kibana-7fb4fd4cc9-f2nls
        replicaSets:
        - kibana-7fb4fd4cc9
        replicas: 1
```

1. Dans la sortie, les champs d'état de la grappe apparaissent dans la strophe **status**.
2. Informations sur les cosses Fluentd.
3. Informations sur les pods Elasticsearch, y compris la santé du cluster Elasticsearch, **green**, **yellow**, ou **red**.

4 Informations sur les modules Kibana.

15.1.1.1. Exemples de messages de condition

Voici des exemples de messages de condition provenant de la section **Status.Nodes** de l'instance OpenShift Logging.

Un message d'état similaire au suivant indique qu'un nœud a dépassé le filigrane bas configuré et qu'aucun billon ne lui sera attribué :

Exemple de sortie

```
nodes:
- conditions:
  - lastTransitionTime: 2019-03-15T15:57:22Z
    message: Disk storage usage for node is 27.5gb (36.74%). Shards will be not
      be allocated on this node.
    reason: Disk Watermark Low
    status: "True"
    type: NodeStorage
  deploymentName: example-elasticsearch-clientdatamaster-0-1
  upgradeStatus: {}
```

Un message d'état similaire au suivant indique qu'un nœud a dépassé le filigrane élevé configuré et que les fragments seront déplacés vers d'autres nœuds :

Exemple de sortie

```
nodes:
- conditions:
  - lastTransitionTime: 2019-03-15T16:04:45Z
    message: Disk storage usage for node is 27.5gb (36.74%). Shards will be relocated
      from this node.
    reason: Disk Watermark High
    status: "True"
    type: NodeStorage
  deploymentName: cluster-logging-operator
  upgradeStatus: {}
```

Un message d'état similaire au suivant indique que le sélecteur de nœuds Elasticsearch dans le CR ne correspond à aucun nœud du cluster :

Exemple de sortie

```
  Elasticsearch Status:
    Shard Allocation Enabled:  shard allocation unknown
    Cluster:
      Active Primary Shards:  0
      Active Shards:          0
      Initializing Shards:    0
      Num Data Nodes:         0
      Num Nodes:              0
      Pending Tasks:          0
      Relocating Shards:      0
```

```
    Status:                 cluster health unknown
    Unassigned Shards:      0
  Cluster Name:             elasticsearch
  Node Conditions:
    elasticsearch-cdm-mkkdys93-1:
      Last Transition Time:  2019-06-26T03:37:32Z
      Message:               0/5 nodes are available: 5 node(s) didn't match node selector.
      Reason:                Unschedulable
      Status:                True
      Type:                  Unschedulable
    elasticsearch-cdm-mkkdys93-2:
  Node Count:  2
  Pods:
    Client:
      Failed:
      Not Ready:
        elasticsearch-cdm-mkkdys93-1-75dd69dccd-f7f49
        elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
      Ready:
    Data:
      Failed:
      Not Ready:
        elasticsearch-cdm-mkkdys93-1-75dd69dccd-f7f49
        elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
      Ready:
    Master:
      Failed:
      Not Ready:
        elasticsearch-cdm-mkkdys93-1-75dd69dccd-f7f49
        elasticsearch-cdm-mkkdys93-2-67c64f5f4c-n58vl
      Ready:
```

Un message d'état similaire au suivant indique que le PVC demandé n'a pas pu se lier à PV :

Exemple de sortie

```
  Node Conditions:
    elasticsearch-cdm-mkkdys93-1:
      Last Transition Time:  2019-06-26T03:37:32Z
      Message:               pod has unbound immediate PersistentVolumeClaims (repeated 5 times)
      Reason:                Unschedulable
      Status:                True
      Type:                  Unschedulable
```

Un message d'état similaire au suivant indique que les pods Fluentd ne peuvent pas être planifiés car le sélecteur de nœuds ne correspond à aucun nœud :

Exemple de sortie

```
Status:
  Collection:
    Logs:
      Fluentd Status:
        Daemon Set:  fluentd
        Nodes:
```

```
Pods:
  Failed:
  Not Ready:
  Ready:
```

15.1.2. Visualisation de l'état des composants du sous-système de journalisation

Vous pouvez consulter l'état d'un certain nombre de composants du sous-système de journalisation.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Modification du projet **openshift-logging**.

```
$ oc project openshift-logging
```

2. Affichez l'état du sous-système de journalisation pour l'environnement Red Hat OpenShift :

```
$ oc describe deployment cluster-logging-operator
```

Exemple de sortie

```
Name:                   cluster-logging-operator

....

Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable

....

Events:
  Type    Reason             Age   From                    Message
  ----    ------             ----  ----                    -------
  Normal  ScalingReplicaSet  62m   deployment-controller  Scaled up replica set cluster-
logging-operator-574b8987df to 1----
```

3. Affichez l'état de l'ensemble de répliques du sous-système de journalisation :

   a. Obtenir le nom d'un ensemble de répliques :

Exemple de sortie

```
$ oc get replicaset
```

Exemple de sortie

```
NAME                                DESIRED   CURRENT   READY   AGE
cluster-logging-operator-574b8987df       1         1         1       159m
elasticsearch-cdm-uhr537yu-1-6869694fb    1         1         1       157m
elasticsearch-cdm-uhr537yu-2-857b6d676f   1         1         1       156m
elasticsearch-cdm-uhr537yu-3-5b6fdd8cfd   1         1         1       155m
kibana-5bd5544f87                         1         1         1       157m
```

b. Obtenir l'état de l'ensemble de répliques :

```
$ oc describe replicaset cluster-logging-operator-574b8987df
```

Exemple de sortie

```
Name:           cluster-logging-operator-574b8987df

....

Replicas:       1 current / 1 desired
Pods Status:    1 Running / 0 Waiting / 0 Succeeded / 0 Failed

....

Events:
  Type    Reason            Age   From                   Message
  ----    ------            ----  ----                   -------
  Normal  SuccessfulCreate  66m   replicaset-controller  Created pod: cluster-logging-
operator-574b8987df-qjhqv----
```

15.2. VISUALISATION DE L'ÉTAT DU MAGASIN DE LOGS ELASTICSEARCH

Vous pouvez consulter l'état de l'OpenShift Elasticsearch Operator et d'un certain nombre de composants Elasticsearch.

15.2.1. Visualisation de l'état du magasin de journaux

Vous pouvez consulter l'état de votre magasin de journaux.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

1. Modification du projet **openshift-logging**.

```
$ oc project openshift-logging
```

2. Pour visualiser l'état :

   a. Obtenir le nom de l'instance du magasin de journaux :

```
$ oc get Elasticsearch
```

Exemple de sortie

```
NAME          AGE
elasticsearch   5h9m
```

b. Obtenir l'état de l'entrepôt de données :

```
$ oc get Elasticsearch <Elasticsearch-instance> -o yaml
```

Par exemple :

```
$ oc get Elasticsearch elasticsearch -n openshift-logging -o yaml
```

La sortie comprend des informations similaires à celles qui suivent :

Exemple de sortie

```
status: 1
  cluster: 2
    activePrimaryShards: 30
    activeShards: 60
    initializingShards: 0
    numDataNodes: 3
    numNodes: 3
    pendingTasks: 0
    relocatingShards: 0
    status: green
    unassignedShards: 0
  clusterHealth: ""
  conditions: [] 3
  nodes: 4
  - deploymentName: elasticsearch-cdm-zjf34ved-1
    upgradeStatus: {}
  - deploymentName: elasticsearch-cdm-zjf34ved-2
    upgradeStatus: {}
  - deploymentName: elasticsearch-cdm-zjf34ved-3
    upgradeStatus: {}
  pods: 5
    client:
      failed: []
      notReady: []
      ready:
      - elasticsearch-cdm-zjf34ved-1-6d7fbf844f-sn422
      - elasticsearch-cdm-zjf34ved-2-dfbd988bc-qkzjz
      - elasticsearch-cdm-zjf34ved-3-c8f566f7c-t7zkt
    data:
      failed: []
      notReady: []
      ready:
      - elasticsearch-cdm-zjf34ved-1-6d7fbf844f-sn422
      - elasticsearch-cdm-zjf34ved-2-dfbd988bc-qkzjz
```

```
    - elasticsearch-cdm-zjf34ved-3-c8f566f7c-t7zkt
  master:
    failed: []
    notReady: []
    ready:
    - elasticsearch-cdm-zjf34ved-1-6d7fbf844f-sn422
    - elasticsearch-cdm-zjf34ved-2-dfbd988bc-qkzjz
    - elasticsearch-cdm-zjf34ved-3-c8f566f7c-t7zkt
shardAllocationEnabled: all
```

1. Dans la sortie, les champs d'état de la grappe apparaissent dans la strophe **status**.
2. L'état de l'entrepôt de données :
   - Nombre d'unités primaires actives.
   - Nombre d'unités actives.
   - Nombre d'unités en cours d'initialisation.
   - Nombre de nœuds de données du magasin de journaux.
   - Nombre total de nœuds de stockage de journaux.
   - Le nombre de tâches en attente.
   - L'état du magasin de journaux : **green**, **red**, **yellow**.
   - Le nombre de tessons non attribués.
3. Toute condition d'état, le cas échéant. L'état du magasin de stockage indique les raisons pour lesquelles l'ordonnanceur n'a pas pu placer un module. Tous les événements liés aux conditions suivantes sont affichés :
   - Conteneur en attente pour les conteneurs de stockage de logs et de proxy.
   - Conteneur Terminé pour les conteneurs de stockage de logs et de proxy.
   - Pod inschedulable. Une condition est également indiquée pour un certain nombre de questions ; voir **Example condition messages**.
4. Les nœuds de stockage de logs dans le cluster, avec **upgradeStatus**.
5. Les pods client, données et maître du log store dans le cluster, répertoriés sous l'état "failed", **notReady** ou **ready**.

15.2.1.1. Exemples de messages de condition

Voici des exemples de messages de condition provenant de la section **Status** de l'instance Elasticsearch.

Le message d'état suivant indique qu'un nœud a dépassé le filigrane bas configuré et qu'aucun billon ne lui sera attribué.

```
status:
  nodes:
```

```
  - conditions:
    - lastTransitionTime: 2019-03-15T15:57:22Z
      message: Disk storage usage for node is 27.5gb (36.74%). Shards will be not
        be allocated on this node.
      reason: Disk Watermark Low
      status: "True"
      type: NodeStorage
    deploymentName: example-elasticsearch-cdm-0-1
    upgradeStatus: {}
```

Le message d'état suivant indique qu'un nœud a dépassé le filigrane élevé configuré et que les fragments seront déplacés vers d'autres nœuds.

```
status:
  nodes:
  - conditions:
    - lastTransitionTime: 2019-03-15T16:04:45Z
      message: Disk storage usage for node is 27.5gb (36.74%). Shards will be relocated
        from this node.
      reason: Disk Watermark High
      status: "True"
      type: NodeStorage
    deploymentName: example-elasticsearch-cdm-0-1
    upgradeStatus: {}
```

Le message d'état suivant indique que le sélecteur de nœud du magasin de journaux dans le CR ne correspond à aucun nœud du cluster :

```
status:
    nodes:
    - conditions:
      - lastTransitionTime: 2019-04-10T02:26:24Z
        message: '0/8 nodes are available: 8 node(s) didn''t match node selector.'
        reason: Unschedulable
        status: "True"
        type: Unschedulable
```

Le message d'état suivant indique que le magasin de journaux CR utilise une revendication de volume persistant (PVC) inexistante.

```
status:
    nodes:
    - conditions:
      - last Transition Time:  2019-04-10T05:55:51Z
        message:               pod has unbound immediate PersistentVolumeClaims (repeated 5 times)
        reason:              Unschedulable
        status:              True
        type:                Unschedulable
```

Le message d'état suivant indique que votre cluster de stockage de journaux n'a pas suffisamment de nœuds pour prendre en charge la stratégie de redondance.

```
status:
  clusterHealth: ""
```

```
  conditions:
  - lastTransitionTime: 2019-04-17T20:01:31Z
    message: Wrong RedundancyPolicy selected. Choose different RedundancyPolicy or
      add more nodes with data roles
    reason: Invalid Settings
    status: "True"
    type: InvalidRedundancy
```

Ce message d'état indique que votre cluster a trop de nœuds de plan de contrôle :

```
status:
  clusterHealth: green
  conditions:
    - lastTransitionTime: '2019-04-17T20:12:34Z'
      message: >-
        Invalid master nodes count. Please ensure there are no more than 3 total
        nodes with master roles
      reason: Invalid Settings
      status: 'True'
      type: InvalidMasters
```

Le message d'état suivant indique que le stockage Elasticsearch ne prend pas en charge la modification que vous avez essayé d'apporter.

Par exemple :

```
status:
  clusterHealth: green
  conditions:
    - lastTransitionTime: "2021-05-07T01:05:13Z"
      message: Changing the storage structure for a custom resource is not supported
      reason: StorageStructureChangeIgnored
      status: 'True'
      type: StorageStructureChangeIgnored
```

Les champs **reason** et **type** indiquent le type de changement non pris en charge :

StorageClassNameChangeIgnored

Modification non prise en charge du nom de la classe de stockage.

StorageSizeChangeIgnored

Non pris en charge modifier la taille de la mémoire.

StorageStructureChangeIgnored

Changement non pris en charge entre les structures de stockage éphémères et persistantes.

IMPORTANT

Si vous essayez de configurer la ressource personnalisée (CR) **ClusterLogging** pour passer d'un stockage éphémère à un stockage persistant, l'OpenShift Elasticsearch Operator crée une réclamation de volume persistant (PVC) mais ne crée pas de volume persistant (PV). Pour effacer l'état **StorageStructureChangeIgnored**, vous devez annuler la modification apportée à la CR **ClusterLogging** et supprimer le PVC.

15.2.2. Visualisation de l'état des composants du magasin de journaux

Vous pouvez consulter l'état d'un certain nombre de composants du magasin de journaux.

Indices Elasticsearch

Vous pouvez consulter l'état des index Elasticsearch.

1. Obtenir le nom d'un module Elasticsearch :

```
$ oc get pods --selector component=elasticsearch -o name
```

Exemple de sortie

```
pod/elasticsearch-cdm-1godmszn-1-6f8495-vp4lw
pod/elasticsearch-cdm-1godmszn-2-5769cf-9ms2n
pod/elasticsearch-cdm-1godmszn-3-f66f7d-zqkz7
```

2. Obtenir l'état des indices :

```
$ oc exec elasticsearch-cdm-4vjor49p-2-6d4d7db474-q2w7z -- indices
```

Exemple de sortie

```
Defaulting container name to elasticsearch.
Use 'oc describe pod/elasticsearch-cdm-4vjor49p-2-6d4d7db474-q2w7z -n openshift-
logging' to see all of the containers in this pod.

green  open   infra-000002                                S4QANnf1QP6NgCegfnrnbQ
3   1    119926            0       157           78
green  open   audit-000001                                8_EQx77iQCSTzFOXtxRqFw
3   1          0          0         0            0
green  open   .security                                 iDjscH7aSUGhIdq0LheLBQ   1
1          5          0         0            0
green  open   .kibana_-377444158_kubeadmin
yBywZ9GfSrKebz5gWBZbjw   3   1          1          0         0            0
green  open   infra-000001                                z6Dpe__ORgiopEpW6Yl44A
3   1    871000            0       874           436
green  open   app-000001                                  hIrazQCeSISewG3c2VIvsQ
3   1       2453           0         3            1
green  open   .kibana_1                                 JCitcBMSQxKOvIq6iQW6wg
1   1          0          0         0            0
green  open   .kibana_-1595131456_user1                        gIYFIEGRRe-
ka0W3okS-mQ   3   1          1          0         0            0
```

Nodules de stockage de grumes

Vous pouvez consulter l'état des pods qui hébergent le magasin de journaux.

1. Obtenir le nom d'un pod :

```
$ oc get pods --selector component=elasticsearch -o name
```

Exemple de sortie

```
pod/elasticsearch-cdm-1godmszn-1-6f8495-vp4lw
pod/elasticsearch-cdm-1godmszn-2-5769cf-9ms2n
pod/elasticsearch-cdm-1godmszn-3-f66f7d-zqkz7
```

2. Obtenir l'état d'un pod :

```
$ oc describe pod elasticsearch-cdm-1godmszn-1-6f8495-vp4lw
```

La sortie comprend les informations d'état suivantes :

Exemple de sortie

```
....
Status:             Running

....

Containers:
  elasticsearch:
    Container ID:   cri-o://b7d44e0a9ea486e27f47763f5bb4c39dfd2
    State:          Running
      Started:      Mon, 08 Jun 2020 10:17:56 -0400
    Ready:          True
    Restart Count:  0
    Readiness:  exec [/usr/share/elasticsearch/probe/readiness.sh] delay=10s timeout=30s
period=5s #success=1 #failure=3

....

  proxy:
    Container ID:  cri-
o://3f77032abaddbb1652c116278652908dc01860320b8a4e741d06894b2f8f9aa1
    State:          Running
      Started:      Mon, 08 Jun 2020 10:18:38 -0400
    Ready:          True
    Restart Count:  0

....

Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True

....

Events:         <none>
```

Configuration du déploiement des pods de stockage de logs

Vous pouvez consulter l'état de la configuration du déploiement de la base de données de journaux.

1. Obtenir le nom d'une configuration de déploiement :

```
$ oc get deployment --selector component=elasticsearch -o name
```

Exemple de sortie

```
deployment.extensions/elasticsearch-cdm-1gon-1
deployment.extensions/elasticsearch-cdm-1gon-2
deployment.extensions/elasticsearch-cdm-1gon-3
```

2. Obtenir l'état de la configuration du déploiement :

```
$ oc describe deployment elasticsearch-cdm-1gon-1
```

La sortie comprend les informations d'état suivantes :

Exemple de sortie

```
....
  Containers:
   elasticsearch:
    Image:      registry.redhat.io/openshift-logging/elasticsearch6-rhel8
    Readiness:  exec [/usr/share/elasticsearch/probe/readiness.sh] delay=10s timeout=30s
period=5s #success=1 #failure=3

....

Conditions:
  Type           Status   Reason
  ----           ------   ------
  Progressing    Unknown  DeploymentPaused
  Available      True     MinimumReplicasAvailable

....

Events:          <none>
```

Ensemble de répliques du magasin de journaux

Vous pouvez consulter l'état de l'ensemble de répliques du magasin de journaux.

1. Obtenir le nom d'un ensemble de répliques :

```
$ oc get replicaSet --selector component=elasticsearch -o name

replicaset.extensions/elasticsearch-cdm-1gon-1-6f8495
replicaset.extensions/elasticsearch-cdm-1gon-2-5769cf
replicaset.extensions/elasticsearch-cdm-1gon-3-f66f7d
```

2. Obtenir l'état de l'ensemble de répliques :

```
$ oc describe replicaSet elasticsearch-cdm-1gon-1-6f8495
```

La sortie comprend les informations d'état suivantes :

Exemple de sortie

```
....
  Containers:
   elasticsearch:
    Image:      registry.redhat.io/openshift-logging/elasticsearch6-
rhel8@sha256:4265742c7cdd85359140e2d7d703e4311b6497eec7676957f455d6908e7b1
c25
    Readiness:  exec [/usr/share/elasticsearch/probe/readiness.sh] delay=10s timeout=30s
period=5s #success=1 #failure=3

....

Events:          <none>
```

15.2.3. État du cluster Elasticsearch

Un tableau de bord dans la section **Observe** de la console web OpenShift Container Platform affiche l'état du cluster Elasticsearch.

Pour obtenir l'état du cluster OpenShift Elasticsearch, visitez le tableau de bord dans la section **Observe** de la console web OpenShift Container Platform à l'adresse **<cluster_url>/monitoring/dashboards/grafana-dashboard-cluster-logging**.

Champs d'état Elasticsearch

eo_elasticsearch_cr_cluster_management_state

Indique si le cluster Elasticsearch est dans un état géré ou non géré. Par exemple :

```
eo_elasticsearch_cr_cluster_management_state{state="managed"} 1
eo_elasticsearch_cr_cluster_management_state{state="unmanaged"} 0
```

eo_elasticsearch_cr_restart_total

Indique le nombre de fois où les nœuds Elasticsearch ont redémarré pour des redémarrages par certificat, des redémarrages par roulement ou des redémarrages planifiés. Par exemple :

```
eo_elasticsearch_cr_restart_total{reason="cert_restart"} 1
eo_elasticsearch_cr_restart_total{reason="rolling_restart"} 1
eo_elasticsearch_cr_restart_total{reason="scheduled_restart"} 3
```

es_index_namespaces_total

Affiche le nombre total d'espaces de noms d'index Elasticsearch. Par exemple :

```
Total number of Namespaces.
es_index_namespaces_total 5
```

es_index_document_count

Indique le nombre d'enregistrements pour chaque espace de noms. Par exemple :

```
es_index_document_count{namespace="namespace_1"} 25
es_index_document_count{namespace="namespace_2"} 10
es_index_document_count{namespace="namespace_3"} 5
```

Le message "Les champs secrets d'Elasticsearch sont soit manquants, soit vides"

S'il manque à Elasticsearch les fichiers **admin-cert**, **admin-key**, **logging-es.crt**, ou **logging-es.key**, le tableau de bord affiche un message d'état similaire à l'exemple suivant :

```
message": "Secret \"elasticsearch\" fields are either missing or empty: [admin-cert, admin-key,
logging-es.crt, logging-es.key]",
"reason": "Missing Required Secrets",
```

15.3. COMPRENDRE LES ALERTES DU SOUS-SYSTÈME DE JOURNALISATION

Toutes les alertes du collecteur de logs sont listées dans l'interface utilisateur Alerting de la console web OpenShift Container Platform.

15.3.1. Visualisation des alertes du collecteur de journalisation

Les alertes sont affichées dans la console web de OpenShift Container Platform, dans l'onglet **Alerts** de l'interface utilisateur Alerting. Les alertes sont dans l'un des états suivants :

- **Firing**. La condition d'alerte est vraie pendant la durée du délai d'attente. Cliquez sur le menu **Options** à la fin de l'alerte pour obtenir plus d'informations ou faire taire l'alerte.
- **Pending** La condition d'alerte est actuellement vraie, mais le délai d'attente n'a pas été atteint.
- **Not Firing**. L'alerte n'est pas encore déclenchée.

Procédure

Pour afficher le sous-système de journalisation et d'autres alertes OpenShift Container Platform :

1. Dans la console OpenShift Container Platform, cliquez sur **Observe → Alerting**.
2. Cliquez sur l'onglet **Alerts**. Les alertes sont répertoriées en fonction des filtres sélectionnés.

Ressources complémentaires

- Pour plus d'informations sur l'interface utilisateur des alertes, voir Gestion des alertes.

15.3.2. À propos de l'enregistrement des alertes du collecteur

Les alertes suivantes sont générées par le collecteur de logs. Vous pouvez voir ces alertes dans la console web de OpenShift Container Platform sur la page **Alerts** de l'interface utilisateur Alerting.

Tableau 15.1. Alertes Fluentd Prometheus

Alerte	Message	Description	Sévérité
FluentDHighErrorRate	**<value> of records have resulted in an error by fluentd <instance>.**	Le nombre d'erreurs de sortie de FluentD est élevé, par défaut plus de 10 dans les 15 minutes précédentes.	Avertissement

Alerte	Message	Description	Sévérité
FluentdNodeDown	**Prometheus could not scrape fluentd <instance> for more than 10m.**	Fluentd rapporte que Prometheus n'a pas pu scraper une instance spécifique de Fluentd.	Critique
FluentdQueueLengthIncreasing	**In the last 12h, fluentd <instance> buffer queue length constantly increased more than 1. Current value is <value>.**	Fluentd signale que la taille de la file d'attente augmente.	Critique
FluentDVeryHighErrorRate	**<value> of records have resulted in an error by fluentd <instance>.**	Le nombre d'erreurs de sortie de FluentD est très élevé, par défaut plus de 25 dans les 15 minutes précédentes.	Critique

15.3.3. À propos des règles d'alerte Elasticsearch

Vous pouvez consulter ces règles d'alerte dans Prometheus.

Tableau 15.2. Règles d'alerte

Alerte	Description	Sévérité
ElasticsearchClusterNotHealthy	L'état de santé de la grappe est ROUGE depuis au moins 2 minutes. Le cluster n'accepte pas les écritures, des shards peuvent être manquants ou le nœud maître n'a pas encore été élu.	Critique
ElasticsearchClusterNotHealthy	L'état de santé du cluster est JAUNE depuis au moins 20 minutes. Certaines répliques de la grappe ne sont pas allouées.	Avertissement
ElasticsearchDiskSpaceRunningLow	Le cluster devrait être à court d'espace disque dans les 6 prochaines heures.	Critique
ElasticsearchHighFileDescriptorUsage	Il est prévu que la grappe soit à court de descripteurs de fichiers dans l'heure qui suit.	Avertissement
ElasticsearchJVMHeapUseHigh	L'utilisation de la mémoire vive de la JVM sur le nœud spécifié est élevée.	Alerte

Alerte	Description	Sévérité
ElasticsearchNodeDiskWatermarkReached	Le nœud spécifié a atteint le filigrane bas en raison d'un faible espace disque disponible. Il n'est plus possible d'allouer des barrettes à ce nœud. Vous devriez envisager d'ajouter de l'espace disque à ce nœud.	Info
ElasticsearchNodeDiskWatermarkReached	Le nœud spécifié a atteint le filigrane élevé en raison d'un faible espace disque disponible. Certains fichiers seront réattribués à d'autres nœuds si possible. Assurez-vous que plus d'espace disque est ajouté au nœud ou supprimez les anciens index alloués à ce nœud.	Avertissement
ElasticsearchNodeDiskWatermarkReached	Le nœud spécifié a atteint le filigrane d'inondation en raison d'un faible espace disque disponible. Chaque index qui a un bloc alloué sur ce nœud est imposé comme un bloc en lecture seule. Le bloc d'index doit être libéré manuellement lorsque l'utilisation du disque passe en dessous du seuil élevé.	Critique
ElasticsearchJVMHeapUseHigh	L'utilisation de la mémoire vive de la JVM sur le nœud spécifié est trop élevée.	Alerte
ElasticsearchWriteRequestsRejectionJumps	Elasticsearch connaît une augmentation des rejets d'écriture sur le nœud spécifié. Il se peut que ce nœud n'arrive pas à suivre la vitesse d'indexation.	Avertissement
AggregatedLoggingSystemCPUHigh	L'unité centrale utilisée par le système sur le nœud spécifié est trop élevée.	Alerte
ElasticsearchProcessCPUHigh	L'unité centrale utilisée par Elasticsearch sur le nœud spécifié est trop élevée.	Alerte

15.4. COLLECTE DE DONNÉES DE JOURNALISATION POUR RED HAT SUPPORT

Lorsque vous ouvrez un dossier d'assistance, il est utile de fournir des informations de débogage sur votre cluster à l'équipe d'assistance de Red Hat.

L'outil**must-gather** vous permet de collecter des informations de diagnostic pour les ressources au niveau du projet, les ressources au niveau du cluster et chacun des composants du sous-système de journalisation.

Pour une assistance rapide, fournissez des informations de diagnostic pour OpenShift Container Platform et OpenShift Logging.

NOTE

N'utilisez pas le script **hack/logging-dump.sh**. Ce script n'est plus pris en charge et ne collecte pas de données.

15.4.1. À propos de l'outil de collecte obligatoire

La commande CLI **oc adm must-gather** recueille les informations de votre cluster les plus susceptibles d'être nécessaires au débogage des problèmes.

Pour votre sous-système de journalisation, **must-gather** collecte les informations suivantes :

- Ressources au niveau du projet, y compris les pods, les cartes de configuration, les comptes de service, les rôles, les liaisons de rôles et les événements au niveau du projet
- Ressources au niveau du cluster, y compris les nœuds, les rôles et les liaisons de rôles au niveau du cluster
- Ressources OpenShift Logging dans les espaces de noms **openshift-logging** et **openshift-operators-redhat**, y compris l'état de santé du collecteur de logs, du magasin de logs et du visualiseur de logs

Lorsque vous exécutez **oc adm must-gather**, un nouveau module est créé sur le cluster. Les données sont collectées sur ce module et enregistrées dans un nouveau répertoire commençant par **must-gather.local**. Ce répertoire est créé dans le répertoire de travail actuel.

15.4.2. Conditions préalables

- Le sous-système de journalisation et Elasticsearch doivent être installés.

15.4.3. Collecte des données de journalisation d'OpenShift

Vous pouvez utiliser la commande CLI **oc adm must-gather** pour collecter des informations sur votre sous-système de journalisation.

Procédure

Pour collecter des informations sur le sous-système de journalisation à l'aide de **must-gather**:

1. Naviguez jusqu'au répertoire dans lequel vous souhaitez stocker les informations de **must-gather**.
2. Exécutez la commande **oc adm must-gather** contre l'image OpenShift Logging :

   ```
   $ oc adm must-gather --image=$(oc -n openshift-logging get deployment.apps/cluster-logging-operator -o jsonpath='{.spec.template.spec.containers[?(@.name == "cluster-logging-operator")].image}')
   ```

   L'outil **must-gather** crée un nouveau répertoire commençant par **must-gather.local** dans le répertoire actuel. Par exemple : **must-gather.local.4157245944708210408**.
3. Créez un fichier compressé à partir du répertoire **must-gather** qui vient d'être créé. Par exemple, sur un ordinateur utilisant un système d'exploitation Linux, exécutez la commande suivante :

   ```
   $ tar -cvaf must-gather.tar.gz must-gather.local.4157245944708210408
   ```
4. Joignez le fichier compressé à votre demande d'assistance sur le portail client de Red Hat .

15.5. DÉPANNAGE POUR LES ALERTES CRITIQUES

15.5.1. La santé du cluster Elasticsearch est rouge

Au moins un nuage primaire et ses répliques ne sont pas attribués à un nœud.

Dépannage

1. Vérifiez l'état de santé du cluster Elasticsearch et vérifiez que le cluster **status** est rouge.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- health
```

2. Liste des nœuds qui ont rejoint le cluster.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_cat/nodes?v
```

3. Listez les pods Elasticsearch et comparez-les aux nœuds dans la sortie de la commande de l'étape précédente.

```
oc -n openshift-logging get pods -l component=elasticsearch
```

4. Si certains nœuds Elasticsearch n'ont pas rejoint le cluster, procédez comme suit.

   a. Confirmer qu'Elasticsearch a un nœud de plan de contrôle élu.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_cat/master?v
```

   b. Examinez les journaux de pods du nœud de plan de contrôle élu pour détecter les problèmes.

```
oc logs <elasticsearch_master_pod_name> -c elasticsearch -n openshift-logging
```

   c. Examinez les journaux des nœuds qui n'ont pas rejoint le cluster.

```
oc logs <elasticsearch_node_name> -c elasticsearch -n openshift-logging
```

5. Si tous les nœuds ont rejoint la grappe, effectuez les étapes suivantes pour vérifier si la grappe est en cours de récupération.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_cat/recovery?active_only=true
```

   S'il n'y a pas de sortie de commande, le processus de récupération peut être retardé ou bloqué par des tâches en attente.

6. Vérifier s'il y a des tâches en attente.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- health |grep
number_of_pending_tasks
```

7. S'il y a des tâches en suspens, surveillez leur état.
   Si leur état change et indique que la grappe se rétablit, continuez à attendre. Le délai de récupération varie en fonction de la taille de la grappe et d'autres facteurs.

Dans le cas contraire, si l'état des tâches en attente ne change pas, cela indique que la récupération est bloquée.

8. S'il semble que la récupération soit bloquée, vérifiez si **cluster.routing.allocation.enable** est réglé sur **none**.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_cluster/settings?pretty
```

9. Si **cluster.routing.allocation.enable** est défini sur **none**, définissez-le sur **all**.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_cluster/settings?pretty -X PUT -d '{"persistent" :
{\i1}"cluster.routing.allocation.enable":\N "all"}}'
```

10. Vérifier quels indices sont encore rouges.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_cat/indices?v
```

11. Si certains indices sont encore rouges, essayez de les effacer en suivant les étapes suivantes.

    a. Vider le cache.

    ```
    oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
    query=<elasticsearch_index_name>/_cache/clear?pretty
    ```

    b. Augmenter le nombre maximum de tentatives d'allocation.

    ```
    oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
    query=<elasticsearch_index_name>/_settings?pretty -X PUT -d
    '{"index.allocation.max_retries":10}'
    ```

    c. Supprimer tous les éléments du défilement.

    ```
    oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
    query=_search/scroll/_all -X DELETE
    ```

    d. Augmenter le délai d'attente.

    ```
    oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
    query=<elasticsearch_index_name>/_settings?pretty -X PUT -d
    '{"index.unassigned.node_left.delayed_timeout":\N "10m"}'
    ```

12. Si les étapes précédentes ne permettent pas d'effacer les indices rouges, supprimez les indices individuellement.

    a. Identifier le nom de l'index rouge.

    ```
    oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
    query=_cat/indices?v
    ```

    b. Supprimer l'index rouge.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=<elasticsearch_red_index_name> -X DELETE
```

13. S'il n'y a pas d'indices rouges et que l'état de la grappe est rouge, vérifiez qu'un nœud de données n'est pas soumis à une charge de traitement élevée et continue.

    a. Vérifiez si l'utilisation de la mémoire vive de la JVM Elasticsearch est élevée.

    ```
    oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
    query=_nodes/stats?pretty
    ```

    Dans la sortie de la commande, examinez le champ **node_name.jvm.mem.heap_used_percent** pour déterminer l'utilisation de la mémoire vive de la JVM.

    b. Vérifier si l'utilisation de l'unité centrale est élevée.

Ressources complémentaires

- Rechercher "Free up or increase disk space" dans la rubrique Elasticsearch, Correction d'un état de cluster rouge ou jaune.

15.5.2. La santé du cluster Elasticsearch est jaune

Les nuages de répliques d'au moins un nuage primaire ne sont pas attribués à des nœuds.

Dépannage

1. Augmenter le nombre de nœuds en ajustant **nodeCount** dans le CR **ClusterLogging**.

Ressources complémentaires

- À propos de la ressource personnalisée Cluster Logging
- Configuration du stockage persistant pour le magasin de journaux
- Rechercher "Free up or increase disk space" dans la rubrique Elasticsearch, Correction d'un état de cluster rouge ou jaune.

15.5.3. Nœud de recherche Elastic atteint le seuil de faible utilisation du disque

Elasticsearch n'attribue pas de parts aux nœuds qui atteignent le filigrane le plus bas .

Dépannage

1. Identifiez le nœud sur lequel Elasticsearch est déployé.

```
oc -n openshift-logging get po -o wide
```

2. Vérifier s'il y a **unassigned shards**.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_cluster/health?pretty | grep unassigned_shards
```

3. S'il y a des unités de stockage non attribuées, vérifiez l'espace disque sur chaque nœud.

```
for pod in `oc -n openshift-logging get po -l component=elasticsearch -o
jsonpath='{.items[*].metadata.name}'`; do echo $pod; oc -n openshift-logging exec -c
elasticsearch $pod -- df -h /elasticsearch/persistent; done
```

4. Vérifiez le champ **nodes.node_name.fs** pour déterminer l'espace disque libre sur ce nœud. Si le pourcentage de disque utilisé est supérieur à 85 %, le nœud a dépassé le filigrane bas et les barrettes ne peuvent plus être allouées à ce nœud.

5. Essayez d'augmenter l'espace disque sur tous les nœuds.

6. S'il n'est pas possible d'augmenter l'espace disque, essayez d'ajouter un nouveau nœud de données au cluster.

7. Si l'ajout d'un nouveau nœud de données pose problème, diminuez la politique de redondance totale de la grappe.

   a. Vérifier le courant **redundancyPolicy**.

```
oc -n openshift-logging get es elasticsearch -o jsonpath='{.spec.redundancyPolicy}'
```

NOTE

Si vous utilisez un CR **ClusterLogging**, entrez :

```
oc -n openshift-logging get cl -o
jsonpath='{.items[*].spec.logStore.elasticsearch.redundancyPolicy}'
```

   b. Si le cluster **redundancyPolicy** est plus élevé que **SingleRedundancy**, réglez-le sur **SingleRedundancy** et enregistrez cette modification.

8. Si les étapes précédentes ne permettent pas de résoudre le problème, supprimez les anciens indices.

   a. Vérifier l'état de tous les index sur Elasticsearch.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- indices
```

   b. Identifier un ancien index qui peut être supprimé.

   c. Supprimer l'index.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=<elasticsearch_index_name> -X DELETE
```

Ressources complémentaires

- Recherche de "redundancyPolicy" dans la ressource personnalisée "Sample **ClusterLogging** custom resource (CR)" dans À propos de la ressource personnalisée Cluster Logging

15.5.4. Elasticsearch Node Disk High Watermark Reached (en anglais)

Elasticsearch tente de relocaliser les shards loin d'un nœud qui a atteint le filigrane le plus élevé .

Dépannage

1. Identifiez le nœud sur lequel Elasticsearch est déployé.

```
oc -n openshift-logging get po -o wide
```

2. Vérifiez l'espace disque sur chaque nœud.

```
for pod in `oc -n openshift-logging get po -l component=elasticsearch -o
jsonpath='{.items[*].metadata.name}'`; do echo $pod; oc -n openshift-logging exec -c
elasticsearch $pod -- df -h /elasticsearch/persistent; done
```

3. Vérifier si le cluster est en cours de rééquilibrage.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=_cluster/health?pretty | grep relocating_shards
```

   Si la sortie de la commande indique que des shards ont été déplacés, cela signifie que le High Watermark a été dépassé. La valeur par défaut du filigrane élevé est de 90 %.

   Les ensembles de données sont déplacés vers un nœud où l'utilisation du disque est faible et qui n'a pas franchi de seuil de filigrane.

4. Pour allouer des fragments à un nœud particulier, libérez de l'espace.

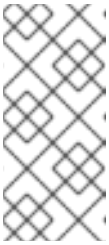
5. Essayez d'augmenter l'espace disque sur tous les nœuds.

6. S'il n'est pas possible d'augmenter l'espace disque, essayez d'ajouter un nouveau nœud de données au cluster.

7. Si l'ajout d'un nouveau nœud de données pose problème, diminuez la politique de redondance totale de la grappe.

   a. Vérifier le courant **redundancyPolicy**.

```
oc -n openshift-logging get es elasticsearch -o jsonpath='{.spec.redundancyPolicy}'
```



NOTE

Si vous utilisez un CR **ClusterLogging**, entrez :

```
oc -n openshift-logging get cl -o
jsonpath='{.items[*].spec.logStore.elasticsearch.redundancyPolicy}'
```

   b. Si le cluster **redundancyPolicy** est plus élevé que **SingleRedundancy**, réglez-le sur **SingleRedundancy** et enregistrez cette modification.

8. Si les étapes précédentes ne permettent pas de résoudre le problème, supprimez les anciens indices.

   a. Vérifier l'état de tous les index sur Elasticsearch.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- indices
```

b. Identifier un ancien index qui peut être supprimé.

c. Supprimer l'index.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=<elasticsearch_index_name> -X DELETE
```

Ressources complémentaires

- Recherche de "redundancyPolicy" dans la ressource personnalisée "Sample **ClusterLogging** custom resource (CR)" dans À propos de la ressource personnalisée Cluster Logging

15.5.5. Le filigrane de l'inondation des disques des nœuds Elasticsearch est atteint

Elasticsearch impose un bloc d'index en lecture seule à chaque index présentant ces deux conditions :

- Un ou plusieurs dépôts sont attribués au nœud.
- Un ou plusieurs disques dépassent le niveau d'inondation.

Dépannage

1. Vérifiez l'espace disque du nœud Elasticsearch.

```
for pod in `oc -n openshift-logging get po -l component=elasticsearch -o
jsonpath='{.items[*].metadata.name}'`; do echo $pod; oc -n openshift-logging exec -c
elasticsearch $pod -- df -h /elasticsearch/persistent; done
```

   Vérifiez le champ **nodes.node_name.fs** pour déterminer l'espace disque libre sur ce nœud.

2. Si le pourcentage de disques utilisés est supérieur à 95 %, cela signifie que le nœud a franchi le seuil d'inondation. L'écriture est bloquée pour les disques alloués à ce nœud particulier.

3. Essayez d'augmenter l'espace disque sur tous les nœuds.

4. S'il n'est pas possible d'augmenter l'espace disque, essayez d'ajouter un nouveau nœud de données au cluster.

5. Si l'ajout d'un nouveau nœud de données pose problème, diminuez la politique de redondance totale de la grappe.

   a. Vérifier le courant **redundancyPolicy**.

```
oc -n openshift-logging get es elasticsearch -o jsonpath='{.spec.redundancyPolicy}'
```

NOTE

Si vous utilisez un CR **ClusterLogging**, entrez :

```
oc -n openshift-logging get cl -o
jsonpath='{.items[*].spec.logStore.elasticsearch.redundancyPolicy}'
```

b. Si le cluster **redundancyPolicy** est plus élevé que **SingleRedundancy**, réglez-le sur **SingleRedundancy** et enregistrez cette modification.

6. Si les étapes précédentes ne permettent pas de résoudre le problème, supprimez les anciens indices.

   a. Vérifier l'état de tous les index sur Elasticsearch.

   ```
   oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- indices
   ```

   b. Identifier un ancien index qui peut être supprimé.

   c. Supprimer l'index.

   ```
   oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
   query=<elasticsearch_index_name> -X DELETE
   ```

7. Continuez à libérer et à surveiller l'espace disque jusqu'à ce que l'espace disque utilisé soit inférieur à 90 %. Débloquez ensuite l'écriture sur ce nœud particulier.

   ```
   oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
   query=_all/_settings?pretty -X PUT -d '{"index.blocks.read_only_allow_delete" : null}'
   ```

Ressources complémentaires

- Recherche de "redundancyPolicy" dans la ressource personnalisée "Sample **ClusterLogging** custom resource (CR)" dans À propos de la ressource personnalisée Cluster Logging

15.5.6. L'utilisation de la mémoire vive de la JVM Elasticsearch est élevée

La mémoire Heap de la JVM du nœud Elasticsearch est supérieure à 75 %.

Dépannage

Envisager d'augmenter la taille du tas.

15.5.7. L'unité centrale du système de journalisation agrégé est élevée

L'utilisation de l'unité centrale du système sur le nœud est élevée.

Dépannage

Vérifiez l'unité centrale du nœud de cluster. Envisagez d'allouer davantage de ressources CPU au nœud.

15.5.8. L'unité centrale du processus Elasticsearch est élevée

L'utilisation de l'unité centrale du processus Elasticsearch sur le nœud est élevée.

Dépannage

Vérifiez l'unité centrale du nœud de cluster. Envisagez d'allouer davantage de ressources CPU au nœud.

15.5.9. L'espace disque d'Elasticsearch est faible

Le cluster Elasticsearch devrait être à court d'espace disque dans les 6 prochaines heures, d'après l'utilisation actuelle du disque.

Dépannage

1. Obtenir l'espace disque du nœud Elasticsearch.

```
for pod in `oc -n openshift-logging get po -l component=elasticsearch -o
jsonpath='{.items[*].metadata.name}'`; do echo $pod; oc -n openshift-logging exec -c
elasticsearch $pod -- df -h /elasticsearch/persistent; done
```

2. Dans la sortie de la commande, vérifiez le champ **nodes.node_name.fs** pour déterminer l'espace disque libre sur ce nœud.

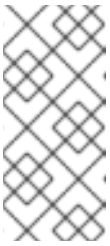
3. Essayez d'augmenter l'espace disque sur tous les nœuds.

4. S'il n'est pas possible d'augmenter l'espace disque, essayez d'ajouter un nouveau nœud de données au cluster.

5. Si l'ajout d'un nouveau nœud de données pose problème, diminuez la politique de redondance totale de la grappe.

   a. Vérifier le courant **redundancyPolicy**.

```
oc -n openshift-logging get es elasticsearch -o jsonpath='{.spec.redundancyPolicy}'
```



NOTE

Si vous utilisez un CR **ClusterLogging**, entrez :

```
oc -n openshift-logging get cl -o
jsonpath='{.items[*].spec.logStore.elasticsearch.redundancyPolicy}'
```

   b. Si le cluster **redundancyPolicy** est plus élevé que **SingleRedundancy**, réglez-le sur **SingleRedundancy** et enregistrez cette modification.

6. Si les étapes précédentes ne permettent pas de résoudre le problème, supprimez les anciens indices.

   a. Vérifier l'état de tous les index sur Elasticsearch.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- indices
```

   b. Identifier un ancien index qui peut être supprimé.

   c. Supprimer l'index.

```
oc exec -n openshift-logging -c elasticsearch <elasticsearch_pod_name> -- es_util --
query=<elasticsearch_index_name> -X DELETE
```

Ressources complémentaires

- Recherche de "redundancyPolicy" dans la ressource personnalisée "Sample **ClusterLogging** custom resource (CR)" dans À propos de la ressource personnalisée Cluster Logging
- Recherchez "ElasticsearchDiskSpaceRunningLow" dans " À propos des règles d'alerte Elasticsearch".
- Rechercher "Free up or increase disk space" dans la rubrique Elasticsearch, Correction d'un état de cluster rouge ou jaune.

15.5.10. L'utilisation du descripteur de fichiers Elasticsearch est élevée

Sur la base des tendances actuelles d'utilisation, le nombre prévu de descripteurs de fichiers sur le nœud est insuffisant.

Dépannage

Vérifiez et, si nécessaire, configurez la valeur de **max_file_descriptors** pour chaque nœud, comme décrit dans la rubrique Elasticsearch File descriptors.

Ressources complémentaires

- Recherchez "ElasticsearchHighFileDescriptorUsage" dans " À propos des règles d'alerte Elasticsearch".
- Search for "File Descriptors In Use" in OpenShift Logging dashboards.

CHAPITRE 16. DÉSINSTALLER OPENSHIFT LOGGING

Vous pouvez supprimer le sous-système de journalisation de votre cluster OpenShift Container Platform.

16.1. DÉSINSTALLATION DU SOUS-SYSTÈME DE JOURNALISATION POUR RED HAT OPENSHIFT

Vous pouvez arrêter l'agrégation des journaux en supprimant la ressource personnalisée (CR) **ClusterLogging**. Après la suppression de la CR, il reste d'autres composants du sous-système de journalisation, que vous pouvez éventuellement supprimer.

La suppression du CR **ClusterLogging** ne supprime pas les réclamations de volumes persistants (PVC). Pour préserver ou supprimer les PVC restants, les volumes persistants (PV) et les données associées, vous devez prendre d'autres mesures.

Conditions préalables

- Les opérateurs Red Hat OpenShift Logging et Elasticsearch doivent être installés.

Procédure

Pour supprimer OpenShift Logging :

1. Utilisez la console web d'OpenShift Container Platform pour supprimer le CR **ClusterLogging**:
   a. Passez à la page **Administration → Custom Resource Definitions**.
   b. Sur la page **Custom Resource Definitions**, cliquez sur **ClusterLogging**.
   c. Sur la page **Custom Resource Definition Details**, cliquez sur **Instances**.
   d. Cliquez sur le menu Options  à côté de l'instance et sélectionnez **Delete ClusterLogging**.
2. Facultatif : Supprimer les définitions de ressources personnalisées (CRD) :
   a. Passez à la page **Administration → Custom Resource Definitions**.
   b. Cliquez sur le menu Options  à côté de **ClusterLogForwarder** et sélectionnez **Delete Custom Resource Definition**.
   c. Cliquez sur le menu Options  à côté de **ClusterLogging** et sélectionnez **Delete Custom Resource Definition**.
   d. Cliquez sur le menu Options  à côté de **Elasticsearch** et sélectionnez **Delete Custom Resource Definition**.

3. Optionnel : Supprimez Red Hat OpenShift Logging Operator et OpenShift Elasticsearch Operator :

   a. Passez à la page **Operators → Installed Operators**.

   b. Cliquez sur le menu Options à côté de Red Hat OpenShift Logging Operator et sélectionnez **Uninstall Operator**.

   c. Cliquez sur le menu Options à côté de OpenShift Elasticsearch Operator et sélectionnez **Uninstall Operator**.

4. Optionnel : Supprimez les projets OpenShift Logging et Elasticsearch.

   a. Passez à la page **Home → Projects**.

   b. Cliquez sur le menu Options à côté du projet **openshift-logging** et sélectionnez **Delete Project**.

   c. Confirmez la suppression en tapant **openshift-logging** dans la boîte de dialogue et cliquez sur **Delete**.

   d. Cliquez sur le menu Options à côté du projet **openshift-operators-redhat** et sélectionnez **Delete Project**.

   IMPORTANT

   Ne supprimez pas le projet **openshift-operators-redhat** si d'autres opérateurs globaux sont installés dans cet espace de noms.

   e. Confirmez la suppression en tapant **openshift-operators-redhat** dans la boîte de dialogue et cliquez sur **Delete**.

5. Pour conserver les PVC afin de les réutiliser avec d'autres pods, conservez les étiquettes ou les noms de PVC dont vous avez besoin pour récupérer les PVC.

6. Facultatif : si vous ne souhaitez pas conserver les PVC, vous pouvez les supprimer.

AVERTISSEMENT

La libération ou la suppression de PVC peut supprimer des PV et entraîner une perte de données.

   a. Passez à la page **Storage → Persistent Volume Claims**

b. Cliquez sur le menu Options  à côté de chaque PVC et sélectionnez **Delete Persistent Volume Claim**.

c. Si vous souhaitez récupérer de l'espace de stockage, vous pouvez supprimer les PV.

Ressources complémentaires

- Récupération manuelle d'un volume persistant

CHAPITRE 17. CHAMPS DE L'ENREGISTREMENT DU JOURNAL

Les champs suivants peuvent être présents dans les enregistrements exportés par le sous-système de journalisation. Bien que les enregistrements soient généralement formatés en tant qu'objets JSON, le même modèle de données peut être appliqué à d'autres encodages.

Pour rechercher ces champs à partir d'Elasticsearch et de Kibana, utilisez le nom complet du champ en pointillés lors de la recherche. Par exemple, avec une recherche Elasticsearch **/_search URL**, pour rechercher un nom de pod Kubernetes, utilisez /**_search/q=kubernetes.pod_name:name-of-my-pod**.

Les champs de premier niveau peuvent être présents dans chaque enregistrement.

CHAPITRE 18. MESSAGE

Le texte original de l'entrée du journal, encodé en UTF-8. Ce champ peut être absent ou vide si un champ **structured** non vide est présent. Voir la description de **structured** pour plus d'informations.

Type de données	texte
Exemple de valeur	**HAPPY**

CHAPITRE 19. STRUCTURÉ

Entrée originale du journal en tant qu'objet structuré. Ce champ peut être présent si le transitaire a été configuré pour analyser les journaux JSON structurés. Si l'entrée originale du journal était un journal structuré valide, ce champ contiendra une structure JSON équivalente. Dans le cas contraire, ce champ sera vide ou absent, et le champ **message** contiendra le message d'origine. Le champ **structured** peut contenir n'importe quel sous-champ inclus dans le message d'enregistrement, il n'y a pas de restrictions définies ici.

Type de données	groupe
Exemple de valeur	map[message:starting fluentd worker pid=21631 ppid=21618 worker=0 pid:21631 ppid:21618 worker:0]

CHAPITRE 20. @TIMESTAMP

Valeur UTC indiquant l'heure de création du fichier journal ou, si l'heure de création n'est pas connue, l'heure à laquelle le fichier journal a été collecté pour la première fois. Le préfixe "@" indique un champ réservé à un usage particulier. Par défaut, la plupart des outils recherchent "@timestamp" avec ElasticSearch.

Type de données	date
Exemple de valeur	**2015-01-24 14:06:05.071000000 Z**

CHAPITRE 21. NOM D'HÔTE

Le nom de l'hôte d'où provient ce message de journal. Dans un cluster Kubernetes, il s'agit du même nom que **kubernetes.host**.

Type de données	mot-clé

CHAPITRE 22. IPADDR4

L'adresse IPv4 du serveur source. Peut être un tableau.

Type de données	ip

CHAPITRE 23. IPADDR6

L'adresse IPv6 du serveur source, si elle est disponible. Peut être un tableau.

Type de données	ip

CHAPITRE 24. NIVEAU

Le niveau de journalisation provenant de diverses sources, notamment **rsyslog(severitytext property)**, un module de journalisation Python, et d'autres.

Les valeurs suivantes proviennent de **syslog.h**et sont précédées de leurs équivalents numériques:

- **0** = **emerg**, le système est inutilisable.
- **1** = **alert**, des mesures doivent être prises immédiatement.
- **2** = **crit**, conditions critiques.
- **3** = **err**, conditions d'erreur.
- **4** = **warn**, conditions d'alerte.
- **5** = **notice**, condition normale mais significative.
- **6** = **info**, informationnel.
- **7** = **debug**, messages de niveau débogage.

Les deux valeurs suivantes ne font pas partie de **syslog.h** mais sont largement utilisées :

- **8** = **trace**, messages de niveau "trace", qui sont plus verbeux que les messages de niveau " **debug** ".
- **9** = **unknown**, lorsque le système d'enregistrement reçoit une valeur qu'il ne reconnaît pas.

Associez les niveaux de journalisation ou les priorités d'autres systèmes de journalisation à leur correspondance la plus proche dans la liste précédente. Par exemple, à partir de la journalisation python, vous pouvez faire correspondre **CRITICAL** avec **crit**, **ERROR** avec **err**, et ainsi de suite.

Type de données	mot-clé
Exemple de valeur	**info**

CHAPITRE 25. PID

L'identifiant du processus de l'entité d'enregistrement, s'il est disponible.

Type de données	mot-clé

CHAPITRE 26. SERVICE

Le nom du service associé à l'entité de journalisation, s'il est disponible. Par exemple, les propriétés **APP-NAME** de syslog et **programname** de rsyslog sont associées au champ service.

Type de données	mot-clé

CHAPITRE 27. ÉTIQUETTES

Facultatif. Une liste de balises définie par l'opérateur et placée sur chaque journal par le collecteur ou le normalisateur. La charge utile peut être une chaîne avec des jetons de chaîne délimités par des espaces blancs ou une liste JSON de jetons de chaîne.

Type de données	texte

CHAPITRE 28. FICHIER

Chemin d'accès au fichier journal à partir duquel le collecteur lit cette entrée de journal. Normalement, il s'agit d'un chemin dans le système de fichiers /**var**/**log** d'un nœud de cluster.

Type de données	texte

CHAPITRE 29. COMPENSATION

La valeur du décalage. Peut représenter des octets jusqu'au début de la ligne de journal dans le fichier (zéro ou un), ou des numéros de ligne de journal (zéro ou un), tant que les valeurs augmentent de façon strictement monotone dans le contexte d'un seul fichier de journal. Les valeurs peuvent être enveloppées, ce qui représente une nouvelle version du fichier journal (rotation).

Type de données	long

CHAPITRE 30. KUBERNETES

L'espace de noms pour les métadonnées spécifiques à Kubernetes

Type de données	groupe

30.1. KUBERNETES.POD_NAME

Le nom du pod

Type de données	mot-clé

30.2. KUBERNETES.POD_ID

L'identifiant Kubernetes du pod

Type de données	mot-clé

30.3. KUBERNETES.NAMESPACE_NAME

Le nom de l'espace de noms dans Kubernetes

Type de données	mot-clé

30.4. KUBERNETES.NAMESPACE_ID

L'ID de l'espace de noms dans Kubernetes

Type de données	mot-clé

30.5. KUBERNETES.HOST

Le nom du nœud Kubernetes

Type de données	mot-clé

30.6. KUBERNETES.CONTAINER_NAME

Le nom du conteneur dans Kubernetes

Type de données	mot-clé

30.7. KUBERNETES.ANNOTATIONS

Annotations associées à l'objet Kubernetes

Type de données	groupe

30.8. KUBERNETES.LABELS

Étiquettes présentes sur le pod Kubernetes original

Type de données	groupe

30.9. KUBERNETES.EVENT

L'événement Kubernetes obtenu à partir de l'API principale de Kubernetes. Cette description d'événement suit vaguement **type Event** dans Event v1 core.

Type de données	groupe

30.9.1. kubernetes.event.verb

Le type d'événement, **ADDED**, **MODIFIED**, ou **DELETED**

Type de données	mot-clé
Exemple de valeur	**ADDED**

30.9.2. kubernetes.event.metadata

Informations relatives au lieu et à l'heure de la création de l'événement

Type de données	groupe

30.9.2.1. kubernetes.event.metadata.name

Le nom de l'objet qui a déclenché la création de l'événement

Type de données	mot-clé
Exemple de valeur	**java-mainclass-1.14d888a4cfc24890**

30.9.2.2. kubernetes.event.metadata.namespace

Le nom de l'espace de noms où l'événement s'est produit à l'origine. Notez qu'il diffère de **kubernetes.namespace_name**, qui est l'espace de noms où l'application **eventrouter** est déployée.

Type de données	mot-clé
Exemple de valeur	**default**

30.9.2.3. kubernetes.event.metadata.selfLink

Un lien vers l'événement

Type de données	mot-clé
Exemple de valeur	**/api/v1/namespaces/javaj/events/java-mainclass-1.14d888a4cfc24890**

30.9.2.4. kubernetes.event.metadata.uid

L'identifiant unique de l'événement

Type de données	mot-clé
Exemple de valeur	**d828ac69-7b58-11e7-9cf5-5254002f560c**

30.9.2.5. kubernetes.event.metadata.resourceVersion

Chaîne de caractères identifiant la version interne du serveur de l'événement. Les clients peuvent utiliser cette chaîne pour déterminer quand les objets ont changé.

Type de données	entier

Exemple de valeur	**311987**

30.9.3. kubernetes.event.involvedObject

L'objet sur lequel porte l'événement.

Type de données	groupe

30.9.3.1. kubernetes.event.involvedObject.kind

Le type d'objet

Type de données	mot-clé
Exemple de valeur	**ReplicationController**

30.9.3.2. kubernetes.event.involvedObject.namespace

Le nom de l'espace de noms de l'objet concerné. Notez qu'il peut être différent de **kubernetes.namespace_name**, qui est l'espace de noms dans lequel l'application **eventrouter** est déployée.

Type de données	mot-clé
Exemple de valeur	**default**

30.9.3.3. kubernetes.event.involvedObject.name

Le nom de l'objet qui a déclenché l'événement

Type de données	mot-clé
Exemple de valeur	**java-mainclass-1**

30.9.3.4. kubernetes.event.involvedObject.uid

L'identifiant unique de l'objet

Type de données	mot-clé
Exemple de valeur	**e6bff941-76a8-11e7-8193-5254002f560c**

30.9.3.5. kubernetes.event.involvedObject.apiVersion

La version de l'API maître de kubernetes

Type de données	mot-clé
Exemple de valeur	**v1**

30.9.3.6. kubernetes.event.involvedObject.resourceVersion

Une chaîne qui identifie la version interne du serveur du pod qui a déclenché l'événement. Les clients peuvent utiliser cette chaîne pour déterminer si des objets ont été modifiés.

Type de données	mot-clé
Exemple de valeur	**308882**

30.9.4. kubernetes.event.reason

Une courte chaîne de caractères compréhensible par la machine qui donne la raison pour laquelle cet événement a été généré

Type de données	mot-clé
Exemple de valeur	**SuccessfulCreate**

30.9.5. kubernetes.event.source_component

Le composant qui a signalé cet événement

Type de données	mot-clé

Exemple de valeur	**replication-controller**

30.9.6. kubernetes.event.firstTimestamp

Heure à laquelle l'événement a été enregistré pour la première fois

Type de données	date
Exemple de valeur	**2017-08-07 10:11:57.000000000 Z**

30.9.7. kubernetes.event.count

Le nombre de fois que cet événement s'est produit

Type de données	entier
Exemple de valeur	**1**

30.9.8. kubernetes.event.type

Le type d'événement, **Normal** ou **Warning**. De nouveaux types pourraient être ajoutés à l'avenir.

Type de données	mot-clé
Exemple de valeur	**Normal**

CHAPITRE 31. OPENSHIFT

L'espace de noms pour les métadonnées spécifiques à openshift-logging

Type de données	groupe

31.1. OPENSHIFT.LABELS

Étiquettes ajoutées par la configuration du Cluster Log Forwarder

Type de données	groupe