SUSE

Livre blanc

Les meilleures pratiques pour sécuriser les environnements Kubernetes

Index

Introduction

Kubernetes s'est imposé comme un puissant outil d'automatisation pour orchestrer les services, les applications et les workloads en conteneurs. Jouant un rôle clé dans le développement de logiciels modernes, Kubernetes fournit une plate-forme flexible et évolutive pour le déploiement et la gestion d'applications en conteneurs dans des environnements cloud. Alors que sa popularité ne cesse de croître, de plus en plus d'organisations se tournent vers Kubernetes pour soutenir leurs initiatives de développement logiciel. Selon une enquête menée en 2021 par la Cloud Native Computing Foundation (CNCF), 96 % des organisations interrogées utilisaient ou envisageaient d'utiliser Kubernetes et « 90 % des utilisateurs de Kubernetes tiraient parti des services gérés dans le cloud, contre près de 70 % en 2020 ».

Cette augmentation de l'adoption nécessite de connaître les meilleures pratiques de sécurité liées à Kubernetes afin d'éviter d'introduire involontairement des paramètres de sécurité mal configurés, qui rendent vos déploiements Kubernetes vulnérables aux attaques pouvant compromettre les données sensibles, perturber les opérations et nuire à la réputation de votre organisation.

En mettant en oeuvre des pratiques de sécurité efficaces, vous pouvez protéger vos déploiements Kubernetes contre ces menaces et garantir l'intégrité, la disponibilité et la confidentialité de vos applications et workloads.

Considérations générales relatives à la sécurisation de Kubernetes

Il est essentiel d'avoir une stratégie pour prévenir les incidents de sécurité dans les environnements Kubernetes et y répondre. Adopter une approche qui consiste à agir « quand cela se produit » plutôt que « si cela se produit », avec une stratégie bien conçue et efficace, peut permettre à votre organisation de réduire l'impact des incidents et de protéger les applications et les données contre les dommages.

La mise en oeuvre d'une stratégie Zero Trust pour la gestion de Kubernetes aide les organisations à réagir plus rapidement en cas d'incident, ce qui permet de réduire les risques et les dommages potentiels d'une cyberattaque.

L'élément clé de ces efforts consiste à adopter un modèle de sécurité Zero Trust, qui part du principe que tous les utilisateurs, réseaux, serveurs, services, ainsi que toutes les applications et API, internes ou externes, sont non approuvés, et ce, jusqu'à ce qu'il soit prouvé que ces éléments sont fiables. Bien que la sécurité traditionnelle repose principalement sur le renforcement du périmètre, dans les environnements de développement et de déploiement actuels, où le périmètre est incertain ou inexistant, le Zero Trust consiste à adopter une approche reposant sur le principe « ne jamais faire confiance, toujours vérifier ». Cela signifie que toutes les tentatives d'accès, qu'elles soient internes ou externes au réseau, doivent être soumises au même niveau d'examen.

L'un des principaux composants d'une stratégie Zero Trust dans Kubernetes est l'utilisation d'un modèle de sécurité déclarative, où l'état souhaité est déclaré via des fichiers YAML, des graphiques Helm et d'autres outils. En utilisant un modèle de sécurité déclaratif, Kubernetes peut être configuré pour appliquer automatiquement des stratégies de sécurité, telles que les contrôles d'accès et la segmentation du réseau, en fonction de l'état souhaité.

La mise en oeuvre d'une stratégie Zero Trust pour la gestion de Kubernetes aide également une organisation à réagir plus rapidement face aux incidents, ce qui permet de réduire les risques et les dommages potentiels d'une cyberattaque. Afin de permettre la résolution la plus rapide possible des incidents de type Zero Day ou autres, votre stratégie de gestion de Kubernetes doit préciser les rôles et les responsabilités des différentes équipes et personnes, et doit décrire les étapes permettant d'identifier, d'étudier et de réduire les incidents de sécurité.

Ces incidents comprennent les éléments suivants :

Fuites de données : causées par de mauvaises pratiques en matière de sécurité des données, les fuites de données désignent des cas où les clés (secrets) utilisées pour identifier et autoriser la communication entre les services et les API sont gérées ou stockées de manière non sécurisée, ce qui donne aux acteurs malveillants un point d'entrée vers les données sensibles.

Violations de données : les violations de données peuvent se produire lorsque les pirates obtiennent un accès non autorisé à des données sensibles telles que les informations d'identification des utilisateurs, les informations financières ou les données relatives à la propriété intellectuelle. Cela peut entraîner une perte de données, une usurpation d'identité et d'autres conséquences graves.

Déni de service : les attaques par déni de service (DoS) sont des cyberattaques dans lesquelles le pirate rend une ressource indisponible pour ses utilisateurs prévus, en perturbant les services d'un hôte connecté à un réseau. Cela peut perturber ou désactiver les services Kubernetes, tels que le serveur d'API Kubernetes, ce qui entraîne des interruptions de service, une perte de productivité et d'autres effets négatifs. Les services API du serveur d'API Kubernetes et de l'application en conteneurs sont des services back-end stratégiques. L'attaque réseau sur les services de l'API REST interrompt le service et entraîne des risques pour l'environnement exécutant l'application.

Logiciels malveillants : les logiciels malveillants, tels que les virus, les vers ou les ransomware, peuvent infecter

les environnements Kubernetes et compromettre la sécurité et la disponibilité des applications et des données.

Cryptomining : les applications de cryptomining sont packagées sous forme de conteneurs ou injectées dans des applications de conteneurs populaires. Une fois ces conteneurs déployés et mis à l'échelle, les mineurs intégrés commencent à épuiser les ressources informatiques des utilisateurs et à présenter un risque pour l'intégrité normale des applications et des services.

Les composants d'un plan de réponse comprennent généralement les étapes suivantes :

Endiguement : la première étape pour répondre à un incident de sécurité consiste à endiguer cet incident, afin de l'empêcher de se propager ou de causer d'autres dommages. Les mesures d'endiguement peuvent inclure la déconnexion des systèmes affectés du réseau, l'isolation des ressources compromises ou le blocage de l'accès aux données sensibles. Les stratégies Zero Trust reposant sur les comportements qui détectent automatiquement les anomalies et y répondent en temps réel permettent de limiter la propagation d'un incident, ce qui peut contribuer à réduire les interruptions de service et l'impact global de l'incident, ainsi que potentiellement à permettre au service de continuer à fonctionner normalement.

Éradication : une fois l'incident endigué, l'étape suivante consiste à éliminer la cause de l'incident afin d'éviter qu'il ne se reproduise. Cela peut impliquer l'identification et la suppression des logiciels malveillants, l'application de correctifs aux vulnérabilités ou la mise en oeuvre de contrôles de sécurité supplémentaires.

Récupération : la dernière étape pour répondre à un incident de sécurité consiste à récupérer après cet incident, à restaurer les opérations normales et à réduire au maximum l'impact sur les applications et les données. Les activités de récupération peuvent inclure la restauration des sauvegardes, la reconstruction des systèmes ou la fourniture de support et d'assistance aux utilisateurs touchés.

Création de rapports : facultatif pour de nombreuses organisations, à l'exception de celles soumises à des exigences de création de rapports obligatoire (par exemple NIST SSDF ou PCI), cette étape implique d'informer les acteurs de toute amélioration supplémentaire requise, ainsi que les autorités, afin d'aider à interpeller les criminels à l'origine de l'incident.

En développant et en mettant en oeuvre un plan complet de préparation et de réponse aux incidents, votre organisation peut améliorer sa capacité à réagir efficacement aux incidents de sécurité urgents, tels que les attaques de ransomware.

Un cycle de vie sécurisé

Assurer la sécurité d'un cluster Kubernetes implique bien plus que le cluster lui-même. L'ensemble du cycle de vie du système, de la chaîne d'approvisionnement de l'infrastructure et des logiciels sous-jacents au déploiement et à la gestion continue des applications, nécessite une planification et une gestion rigoureuses pour réduire le risque de vulnérabilités de sécurité.

Une chaîne d'approvisionnement des conteneurs sécurisée

L'une des considérations les plus importantes est la sécurisation de votre chaîne d'approvisionnement des conteneurs. Il est essentiel de s'assurer que les composants de votre cluster Kubernetes, y compris le logiciel Kubernetes lui-même, sont obtenus à partir de sources fiables, afin de vous protéger contre les logiciels malveillants et d'autres menaces de sécurité. Cela implique de faire appel à des fournisseurs officiels et vérifiés pour votre logiciel Kubernetes et d'autres composants, et de mettre ces derniers à jour et d'appliquer des correctifs régulièrement, afin de corriger les vulnérabilités connues.

La compréhension des composants d'un logiciel, l'identification des vulnérabilités potentielles en matière de sécurité et la garantie d'une licence et d'une compatibilité correctes sont des opérations qui demandent des efforts considérables, mais elles peuvent être simplifiées par l'acquisition d'une nomenclature logicielle. Une nomenclature logicielle (SBOM) est définie par la National Telecommunications and Information Administration (NTIA) comme « une liste formellement structurée de composants, de bibliothèques, et de modules nécessaires pour construire (compiler et relier) un logiciel donné et les relations de la chaîne d'approvisionnement. Ces composants peuvent être Open Source ou propriétaires, gratuits ou payants, et faire l'objet d'un accès largement disponible ou restreint. » Une nomenclature logicielle supprime la découverte manuelle nécessaire pour garantir que les composants d'une solution logicielle sont correctement identifiés et contrôlés. Le Software Package Data Exchange (SPDX) est devenu une norme pour la communication SBOM entre plusieurs fournisseurs et plates-formes.

Stockage et chiffrement réseau

Le chiffrement des données sensibles constitue un autre aspect important de la sécurité de Kubernetes, à la fois en stockage et en transit. Il s'effectue en utilisant des protocoles sécurisés tels que SSL/TLS, qui chiffrent les données lorsqu'elles sont envoyées sur un réseau. Ceci est particulièrement important pour empêcher les données sensibles, telles que les informations d'identification des utilisateurs et les données financières, d'être interceptées et accessibles par des parties non autorisées.

Les organisations adoptent les meilleures pratiques en matière de sécurité

Plusieurs organisations de normes de sécurité promeuvent des méthodologies efficaces pour sécuriser les déploiements Kubernetes. Il s'agit notamment des directives d'organisations telles que le Center for Internet Security (CIS), le National Institute of Standards and Technology (NIST), l'Open Web Application Security Project (OWASP), le Supply-chain Levels for Software Artifacts (SLSA) de la Fondation Linux, l'Open Source Security Foundation (OpenSSF), le MITRE et la National Security Agency (NSA).

Ces organisations fournissent des recommandations spécifiques pour sécuriser les clusters Kubernetes et la chaîne d'approvisionnement de logiciels, y compris les configurations recommandées et les mesures de renforcement. Tout effort visant à développer une stratégie de sécurité Kubernetes doit se référer aux ressources de ces organisations et respecter les directives recommandées.

Solutions de sécurité

Plusieurs solutions de sécurité sont disponibles pour Kubernetes, notamment des outils de monitoring et de sécurisation du réseau, d'authentification et d'autorisation, ainsi que d'analyse et de sécurisation des images. Par exemple, SUSE NeuVector est une solution de sécurité de SUSE qui assure une sécurité continue des conteneurs, y compris la sécurité du réseau et de l'exécution, ainsi que la conformité.

Automatisation de la sécurité

L'automatisation de la sécurité est essentielle lorsqu'il s'agit de gérer et de déployer des applications sur Kubernetes. En automatisant les tâches de sécurité, les équipes peuvent réduire le risque d'erreur humaine et améliorer la sécurité globale de leurs déploiements. L'une des approches pour automatiser la sécurité dans Kubernetes consiste à recourir à GitOps, une méthode d'utilisation de Git comme source de référence unique

pour les configurations déclaratives d'applications et d'infrastructures. Cela permet aux équipes d'automatiser et de sécuriser le déploiement, l'évolutivité et la gestion des applications et de l'infrastructure, en utilisant Git comme point central de collaboration et de contrôle. Comme indiqué dans l'article de blog The GitOps Kubernetes Connection, « GitOps améliore vos processus, augmente la maturité et aide les équipes à livrer (et à exploiter) des applications en exploitant des processus et des outils déclaratifs. »

Une autre approche pour automatiser la sécurité dans Kubernetes consiste à utiliser des moteurs de déploiement et de gestion de conteneurs comme Fleet, un outil permettant de déployer, de gérer et de faire évoluer les clusters Kubernetes de manière cohérente et sécurisée. À mesure que la demande de déploiement de clusters Kubernetes à la périphérie à grande échelle augmente, la gestion de clusters passe d'un niveau individuel à une gestion d'ensembles de clusters. Fleet fournit un mécanisme intégré pour personnaliser les ensembles par cluster cible, à l'aide d'outils standard tels que Helm et Kustomize.

Un service de création sécurisée

Le concept de création sécurisée met l'accent sur la sécurité intégrée. Il s'agit de « créer des logiciels de manière standardisée et reproductible, en utilisant des composants sécurisés, y compris des dépendances logicielles tierces », selon OWASP. Dans cette optique, l'objectif final d'un service de création sécurisée est de définir le processus de création ainsi que les audits de sécurité obligatoires, et d'atteindre une automatisation maximale ou complète, dans laquelle les images de conteneur ne sont pas créées de manière individuelle sur le poste de travail d'un développeur ou ouvertes à la falsification par des utilisateurs non autorisés. Au lieu de cela, un service de création sécurisée crée et déploie une image dans une enclave isolée accessible via une API spécifique, réduisant ainsi la surface d'attaque et minimisant les risques.

L'OWASP a mis en place un modèle SAMM (Software Assurance Maturity Model) pour « fournir à tous les types d'organisations un moyen efficace et mesurable d'analyser et d'améliorer leur stratégie de sécurité logicielle ». Le tableau du flux de maturité de la création sécurisée du SAMM, illustré ci-dessous, est utile pour le processus de création.

Niveau de maturité	Définition	Flux A Processus de création	Flux B Dépendances logicielles
1	Le processus de création est reproductible et cohérent.	Créez une définition formelle du processus de création afin qu'il devienne cohérent et reproductible.	Créez des enregistrements avec la nomenclature de vos applications et analysez-les lorsque cela est pertinent.
2	Le processus de création est optimisé et entièrement intégré au workflow.	Automatisez votre pipeline de création et sécurisez les outils utilisés. Ajoutez des contrôles de sécurité dans le pipeline de création.	Évaluez les dépendances utilisées et assurez-vous de réagir rapidement en cas de situation présentant des risques pour vos applications.
3	Le processus de création permet d'éviter que des défauts connus ne pénètrent dans l'environnement de production.	Définissez des contrôles de sécurité obligatoires dans le processus de création et assurez-vous que la création d'artefacts non conformes soit impossible.	Analysez les dépendances utilisées pour les problèmes de sécurité de manière comparable à votre propre code.

Source : OWASP

Définitions de ressources personnalisées

Les définitions de ressources personnalisées (CRD) dans Kubernetes offrent un moyen puissant d'automatiser la sécurité en permettant aux développeurs et aux équipes DevOps de déclarer le comportement autorisé pour les environnements système de leurs applications. Ce comportement peut ensuite être surveillé et appliqué dans un environnement de production, afin de garantir la sécurité et la conformité des environnements système des applications. Il est possible de faire cela en créant des ressources personnalisées qui représentent l'état souhaité de l'application, y compris les stratégies de sécurité qui doivent être appliquées.

En outre, les définitions de ressources personnalisées peuvent être utilisées pour automatiser le processus de création et de mise à jour des objets Kubernetes, tels que les pods, les services et les déploiements, ce qui peut aider à réduire les erreurs humaines et à améliorer la sécurité globale des environnements système des applications.

Ces outils peuvent vous aider à automatiser le processus de sécurité et de conformité, ce qui vous permet de déployer des applications sécurisées en toute simplicité.

L'impact de l'évolutivité et de la sécurité sur les performances

L'évolutivité et la sécurité sont souvent étroitement liées. Lorsque la capacité à faire évoluer rapidement et facilement une application livre des performances optimales, la plate-forme de sécurité doit être suffisamment robuste pour ne pas entraver l'application et affecter sa sécurité globale.

Dans Kubernetes, lorsque les environnements système sont dynamiques et que le système réagit automatiquement aux modifications, si la plate-forme de sécurité n'est pas adaptée, l'application ne peut pas évoluer (en raison de l'impact de la plate-forme de sécurité sur les performances), ou la plate-forme de sécurité ne parvient pas à protéger l'application. Une plate-forme de sécurité doit pouvoir évoluer sans nuire aux performances des applications, afin de répondre à la demande en matière de workload.

Dans Kubernetes, lorsque les environnements système sont dynamiques et que le système réagit automatiquement aux modifications, si la plate-forme de sécurité n'est pas adaptée, l'application ne peut pas évoluer (en raison de l'impact de la plate-forme de sécurité sur les performances), ou la plate-forme de sécurité ne parvient pas à protéger l'application.

Infrastructure et système d'exploitation

Assurer la sécurité de l'infrastructure et du système d'exploitation constitue un aspect essentiel des meilleures pratiques de sécurité Kubernetes. Examinez le modèle des 4 « C » de la sécurité cloud native de Kubernetes.io : les couches du modèle (cloud, cluster, conteneur et code) bénéficient de chacune des autres couches. L'infrastructure et le système d'exploitation sous-jacents constituent la base d'un déploiement Kubernetes sécurisé, et toute vulnérabilité ou faiblesse de ces composants peut compromettre la sécurité de l'ensemble du système.

L'importance d'une base solide

Il est essentiel de disposer d'une base solide pour les déploiements Kubernetes, afin de garantir la sécurité. Cela passe par l'utilisation d'une chaîne d'approvisionnement sécurisée pour obtenir l'infrastructure et les composants du système d'exploitation, comme l'utilisation d'une distribution Linux fiable telle que SUSE Linux Enterprise Server (SLES). SUSE Linux Enterprise Server fournit une base sécurisée et stable pour les déploiements Kubernetes, avec des fonctions avancées de sécurité et de conformité, la prise en charge des conteneurs et des microservices, et la prise en charge des environnements hybrides et multicloud. Selon le modèle des 4 C, « on ne peut pas se protéger contre les mauvaises normes de sécurité dans les couches de base en traitant la sécurité au niveau du code. »

Une chaîne d'approvisionnement et des certifications sécurisées

En plus d'utiliser une chaîne d'approvisionnement sécurisée, il est important de vous assurer que votre infrastructure et vos composants de système d'exploitation disposent des certifications appropriées. Les certifications telles que Common Criteria/EAL4+ offrent aux organisations la garantie d'une évaluation de sécurité du niveau le plus élevé possible pour un système d'exploitation Open Source. Ces certifications garantissent également que la chaîne d'approvisionnement de logiciels est sécurisée dans son étendue actuelle.

La norme FIPS 140-2/3 (Federal information Processing Standard), publiée par le NIST, est une norme largement reconnue pour les modules cryptographiques utilisés afin de protéger les informations sensibles dans les systèmes informatiques. Ces certifications assurent aux organisations que les composants qu'elles utilisent sont sécurisés et conformes. Cela vous garantit ainsi que votre distribution Kubernetes respecte les normes acceptées.

L'analyse du registre à la recherche de vulnérabilités dans les images est une étape clé du processus de sécurité de la chaîne d'approvisionnement. Elle permet d'identifier les faiblesses ou vulnérabilités connues pouvant être exploitées par des pirates ou des acteurs malveillants, ce qui vous donne la possibilité de corriger et de garantir la sécurité de vos déploiements Kubernetes.

Cependant, il ne suffit pas d'analyser le registre pour garantir la sécurité d'un déploiement Kubernetes. Il est également important de mettre en place des protocoles de sécurité robustes tout au long de la chaîne d'approvisionnement, notamment des pratiques de programmation, de développement et de déploiement sécurisées, y compris en ce qui concerne les logiciels utilisés pour déployer et créer des changements.

En outre, comme nous l'avons mentionné précédemment, chaque organisation doit mettre en place un plan de réponse aux incidents pour diriger les opérations en cas de violation de la sécurité et, lorsque cela est possible, bloquer les attaques et les exploits en cours avant qu'ils ne puissent se propager à d'autres systèmes et environnements système. Cela réduit l'impact sur votre société et vos clients.

Conformité réglementaire

Votre organisation peut être très réglementée et exiger la conformité aux normes réglementaires telles que la norme PCI DSS (Payment Card Industry Standard), le RGPD (Règlement général sur la protection des données), la loi HIPAA (Health Insurance Portability and Accountability Act) et le cadre de cybersécurité NIST. Ces normes fournissent des directives et de meilleures pratiques pour la protection des données sensibles et le maintien de la sécurité des systèmes.

Par exemple, la norme PCI DSS exige que les organisations qui gèrent les transactions par carte de crédit mettent en oeuvre certaines mesures de sécurité, telles que le chiffrement des données sensibles et la mise en oeuvre de stratégies de mots de passe solides. Le RGPD définit les exigences relatives à la protection des données personnelles et vise à vérifier que les individus ont le contrôle de leurs propres données. La loi fédérale HIPAA (Health Insurance Portability and Accountability Act) établit des normes de protection des informations médicales sensibles.

Pour être en conformité avec ces normes, vous devez mettre en oeuvre des mesures de sécurité appropriées et effectuer des audits réguliers de votre environnement Kubernetes. Cela inclut la mise en oeuvre de contrôles d'accès, de la sécurité du réseau et de l'infrastructure, ainsi que le monitoring des failles de sécurité potentielles. En suivant ces directives, les organisations peuvent contribuer à réduire le risque de violation de données et s'assurer qu'elles protègent leurs systèmes et leurs données sensibles de la manière la plus efficace possible.

Conformité avec Rancher Prime

L'un des moyens d'assurer la conformité réglementaire lors de l'utilisation de Kubernetes est d'utiliser Rancher Prime by SUSE, qui fournit des fonctions telles que des contrôles d'accès basés sur les rôles et des stratégies réseau pour optimiser la sécurité sur les clusters et dans les environnements système. Rancher Prime est intégré à SUSE NeuVector afin de fournir une sécurité continue pour les environnements Kubernetes, y compris en ce qui concerne la gestion des vulnérabilités, la protection de l'exécution et la création de rapports de conformité, ce qui vous aidera à mettre en oeuvre et à maintenir plus facilement les normes réglementaires comme celles mentionnées dans ce document.

Contexte de sécurité et SELinux (Security-Enhanced Linux)/ AppArmor

Le contexte de sécurité est un concept dans Kubernetes qui fait référence aux paramètres et stratégies de sécurité qui s'appliquent à un workload ou à une ressource donnée. Ces paramètres peuvent inclure des éléments tels que les autorisations, les rôles et les capacités, et permettent de contrôler l'accès aux ressources et l'utilisation de ces dernières. Le contexte de sécurité est utilisé pour appliquer des contrôles d'accès et empêcher tout accès non autorisé aux données et ressources sensibles.

SELinux et AppArmor sont des infrastructures de sécurité qui appliquent des stratégies de sécurité dans les déploiements Kubernetes en limitant les opérations que les processus peuvent effectuer. Cela peut aider à empêcher tout accès non autorisé aux ressources. Cependant, ces infrastructures ne protègent pas les applications de tout écart par rapport à leur comportement standard, sauf si des stratégies spécifiques sont créées pour cela. Cette tâche peut s'avérer complexe, car elle nécessite une compréhension approfondie du comportement exact

des applications à un niveau bas. SELinux et AppArmor permettent de contrôler de manière précise l'accès aux ressources, et peuvent être utilisés pour se protéger contre les menaces de sécurité potentielles. SELinux est un module de sécurité du kernel Linux, tandis qu'AppArmor est utilisé dans Linux et d'autres systèmes d'exploitation.

Cluster

Un cluster Kubernetes est un groupe de noeuds qui fonctionnent ensemble pour exécuter et gérer des applications. Ces noeuds peuvent être des machines virtuelles ou des serveurs physiques qui communiquent entre eux pour garantir le bon fonctionnement et l'efficacité des applications. Un cluster Kubernetes se compose d'un plan de contrôle, d'au moins un noeud principal et de plusieurs noeuds de travail. Le noeud maître est chargé de maintenir l'intégrité globale du cluster et de s'assurer que tous les noeuds de travail fonctionnent correctement. Pour cela, le noeud maître planifie et déploie des applications, surveille l'état du cluster et fournit une API permettant aux utilisateurs d'interagir avec le cluster.

Les noeuds de travail sont responsables de l'exécution des applications déployées sur le cluster. Pour cela, ils exécutent des conteneurs, qui sont des packages légers et portables contenant tout le code et les dépendances nécessaires à l'exécution d'une application. Chaque noeud de travail procède à une exécution de conteneur, telle que Docker, qui est responsable du lancement et de la gestion des conteneurs. Les noeuds de travail communiquent également avec le noeud maître pour générer des rapports sur leur état et recevoir des instructions pour le déploiement et la gestion des applications. En utilisant un système distribué comme Kubernetes, les organisations peuvent facilement faire évoluer leurs applications et maintenir une haute disponibilité.

La sécurisation de vos clusters Kubernetes est d'une importance capitale, car les erreurs de configuration peuvent compromettre le cluster et entraîner des fuites de données ou d'autres problèmes. Le monitoring de ces éléments à la recherche de défauts connus et de configurations dangereuses peut réduire considérablement le risque d'avoir un cluster de production vulnérable.

Noeuds

Les noeuds sont les machines individuelles, virtuelles ou physiques, qui composent le cluster. Pour garantir la sécurité des noeuds, les meilleures pratiques incluent l'analyse régulière des bancs d'essai CIS (Center for Internet Security) pour identifier les vulnérabilités potentielles et générer des rapports de vulnérabilité afin de suivre et de résoudre les problèmes. En outre, l'utilisation d'un système d'exploitation certifié avec les certifications CC (Common Criteria) permet de vérifier que la chaîne d'approvisionnement est sécurisée jusqu'à ce point. Les mesures d'application, telles que la restriction de l'accès aux noeuds et la mise en oeuvre de stratégies de sécurité réseau, peuvent également aider à sécuriser les noeuds.

SDN - Plug-ins réseau

Kubernetes utilise le Software-defined Networking (SDN) pour gérer et mettre en oeuvre l'infrastructure réseau dans un environnement virtualisé. Les plug-ins réseau dans Kubernetes activent la communication entre les différents composants du cluster et fournissent des services réseau aux applications. Bien que les stratégies réseau puissent être utilisées pour contrôler et restreindre l'accès au réseau, la mise en oeuvre du chiffrement pour les communications réseau et l'utilisation d'outils de sécurité complets tels que SUSE NeuVector peuvent fournir une solution de sécurité plus robuste pour un cluster Kubernetes.

API Cluster

L'API Cluster est un projet Kubernetes qui fournit une API déclarative pour la gestion des clusters. Pour sécuriser l'API Cluster, les meilleures pratiques incluent l'utilisation de mesures d'authentification et d'autorisation rigoureuses, telles que l'utilisation d'informations d'identification chiffrées et la mise en oeuvre de contrôles d'accès basés sur des rôles. L'utilisation d'une plate-forme de gestion Kubernetes telle que Rancher fournit une fonction de proxy d'authentification capable de sécuriser l'accès à l'API Cluster.

Informations d'identification

Les informations d'identification, telles que les combinaisons d'un nom d'utilisateur et d'un mot de passe ou les jetons d'authentification, déterminent qui ou quoi peut accéder au cluster Kubernetes et à ses ressources. Pour sécuriser les informations d'identification dans Kubernetes, les meilleures pratiques incluent la mise en oeuvre de stratégies de mots de passe difficiles, la limitation des droits assignés aux utilisateurs à ce qu'ils ont besoin de faire uniquement, la suppression des informations d'identification qui ne sont plus nécessaires, l'établissement d'un processus de mise hors service des utilisateurs/outils et l'utilisation du chiffrement pour protéger les informations sensibles.

En outre, la rotation et la suppression régulières des informations d'identification sont un aspect clé pour garantir que seuls les personnes ou les outils appropriés ont accès au cluster Kubernetes et à ses ressources. Cela permet d'éviter les accès non autorisés et les violations de données en réduisant le risque de vol ou de compromission des informations d'identification.

Mises à jour

Selon ZDNet, « une violation sur trois est causée par des vulnérabilités non corrigées ». Étant donné que ces événements auraient pu être évités par une mise à jour logicielle disponible, la mise en place de mesures pour maintenir à jour les outils, les dépendances et les autres composants réduit considérablement la probabilité d'une violation réussie.

L'exécution de la dernière version de Kubernetes est le moyen le plus simple d'améliorer la sécurité des clusters. C'est aussi simple que de télécharger la dernière version de Kubernetes. Pour vérifier quelle est la version la plus à jour de Kubernetes, exécutez la commande suivante :

```
kubectl version
```

Étant donné que les développeurs individuels sont responsables de la maintenance de leur code, l'intégration des contrôles de sécurité dans les workflows des développeurs est un moyen efficace de s'assurer que les dépendances restent à jour.

Journaux d'audit/Audit de l'API Kubernetes

Les journaux d'audit et l'audit de l'API Kubernetes sont importants pour le suivi et le monitoring de l'activité au sein d'un cluster Kubernetes. Les journaux d'audit fournissent un enregistrement de toutes les requêtes d'API envoyées au cluster, y compris des informations sur la demande, l'utilisateur qui l'a effectuée et la réponse à cette demande. Cela peut être utile pour détecter et examiner les problèmes de sécurité potentiels, ainsi que pour suivre l'intégrité et les performances globales du cluster. En outre, la configuration d'alertes et de protocoles d'action peut constituer un bon moyen d'éviter les interruptions de service et détecter les incidents en temps voulu. Ainsi, si une activité inhabituelle ou suspecte est détectée, l'équipe responsable peut être immédiatement informée et effectuer les opérations nécessaires pour atténuer les risques.

Segmentation des noeuds

La segmentation des applications est une mesure de sécurité qui consiste à diviser les applications d'un cluster Kubernetes en différents groupes ou segments, en fonction de leur rôle ou de leur fonction. Cela permet d'empêcher toute communication ou tout accès non autorisé entre les applications et d'appliquer différentes stratégies ou restrictions de sécurité à différents groupes d'applications. Ainsi, il est garanti que seules les communications autorisées entre les applications sont permises, et que les communications non autorisées sont restreintes.

RBAC

Dans Kubernetes, l'accès aux ressources au sein d'un cluster est géré et contrôlé par une combinaison d'éléments : Roles, ClusterRoles, RoleBindings, ResourceQuotas et Role-Based Access Control (RBAC). Les éléments « Roles » et « ClusterRoles » définissent les autorisations et les droits spécifiques d'un utilisateur ou d'un groupe d'utilisateurs au sein du cluster. Les éléments « RoleBindings » et « ResourceQuotas » sont utilisés pour limiter les ressources auxquelles les utilisateurs peuvent accéder ou qu'ils peuvent utiliser. Le RBAC est le système qui gère et applique ces contrôles d'accès pour s'assurer que les utilisateurs peuvent uniquement effectuer les opérations et accéder aux ressources pour lesquelles ils disposent d'une autorisation. En résumé, les éléments « Roles » et « ClusterRoles » définissent les autorisations, les éléments « RoleBindings » et « ResourceQuotas » limitent l'accès aux ressources, et le RBAC est le système qui applique les contrôles d'accès pour garantir la sécurité et la conformité.

Pour confirmer qu'une identité dispose des autorisations adéquates pour créer des pods, exécutez la commande suivante :

```
kubectl auth can-i create pods
-as=<identity>
```

Cela renvoie une valeur booléenne « vrai » si l'utilisateur donné dispose des autorisations adéquates. L'audit des autorisations peut sembler fastidieux, mais il est beaucoup moins difficile que le traitement d'une violation de données ou d'un cluster compromis en raison d'un manque de contrôle d'accès basé sur les rôles.

Stratégies et traitement automatique

Les stratégies et la correction automatique sont des outils permettant de gérer et d'appliquer la sécurité et la conformité dans un cluster Kubernetes. Les stratégies définissent les règles et les normes dans le cluster, tandis que la correction automatique désigne l'application automatique de ces règles et normes pour garantir la conformité. Par exemple, une stratégie peut spécifier que tous vos conteneurs s'exécutent avec un certain niveau de sécurité, et la correction automatique applique automatiquement cette règle en s'assurant que tous les conteneurs qui ne répondent pas aux exigences de sécurité ne sont pas autorisés à s'exécuter.

Analyse des vulnérabilités

Il est important d'analyser régulièrement votre plate-forme Kubernetes pour détecter les vulnérabilités connues. Cela peut aider à identifier les problèmes de sécurité potentiels, tels que les vulnérabilités connues dans les versions de Kubernetes ou les dépendances utilisées. En analysant régulièrement les vulnérabilités et en appliquant des correctifs ou des mises à niveau selon les besoins, vous pouvez vous assurer que la plate-forme Kubernetes elle-même est aussi sécurisée que possible.

Réseautique

Dans Kubernetes, la réseautique fait référence à la façon dont les conteneurs et autres ressources communiquent entre elles et avec les éléments extérieurs. Cela inclut le réseau virtuel qui connecte les conteneurs au sein d'un cluster Kubernetes, ainsi que les connexions entre le cluster et d'autres réseaux externes. Une configuration réseautique appropriée est essentielle pour garantir que les applications peuvent communiquer entre elles et accéder aux ressources dont elles ont besoin, tout en les protégeant contre les accès non autorisés et autres menaces de sécurité.

Stratégies réseau

Les stratégies réseau sont une fonction Kubernetes qui permet aux administrateurs de définir des règles de circulation du trafic réseau au sein d'un cluster. Ces règles permettent d'isoler les différents composants ou applications, limitant ainsi la communication entre ces éléments aux seules connexions nécessaires. Cela peut contribuer à empêcher toute communication ou tout accès non autorisé entre les composants, réduisant ainsi le risque de vulnérabilités de sécurité.

Ports exposés sur les hôtes Kubernetes

Dans Kubernetes, les conteneurs sont généralement accessibles et gérés via des ports réseau. Par défaut, tous les conteneurs d'un cluster se voient attribuer une adresse IP unique et une plage de ports réseau qu'ils peuvent utiliser pour la communication. Cependant, dans certains cas, il peut être nécessaire d'exposer des ports spécifiques sur la machine hôte exécutant le cluster Kubernetes, ce qui permet à des clients externes d'accéder directement aux conteneurs. Cela peut s'avérer utile à des fins de débogage ou autres, mais cela augmente également la surface d'attaque du cluster et nécessite des considérations de sécurité minutieuses.

SUSE NeuVector : une solution de sécurité Zero Trust

SUSE NeuVector va au-delà des stratégies réseau pour fournir des fonctions avancées telles que la segmentation Layer7 (couche applicative) sophistiquée, le contrôle de sortie, la prévention des pertes de données (DLP) et les protections WAF (Web Application Firewall). Cela permet de s'assurer que le réseau est sécurisé et protégé contre divers types d'attaques et de vulnérabilités.

Sortie et entrée

La sortie, dans le contexte de Kubernetes, désigne le flux du trafic réseau qui provient d'un cluster Kubernetes. Il est important de contrôler et de surveiller les sorties à des fins de sécurité, car cela permet d'empêcher tout accès non autorisé à des ressources en dehors du cluster. Par exemple, si un pirate pouvait accéder à votre cluster et exfiltrer des données sensibles via le trafic de sortie, il pourrait compromettre la sécurité de l'ensemble du système. Utilisez les stratégies réseau pour spécifier les ressources pouvant communiquer avec des ressources externes, et utilisez des outils tels que le monitoring réseau et les systèmes de détection d'intrus pour détecter et prévenir toute activité suspecte.

L'entrée désigne le flux du trafic réseau vers un cluster Kubernetes. Tout comme pour le trafic de sortie, il est important de contrôler et de surveiller correctement le trafic d'entrée pour empêcher tout accès non autorisé au cluster et à ses ressources. Par exemple, un attaquant pourrait potentiellement accéder à votre cluster en exploitant les vulnérabilités du trafic entrant.

Les meilleures pratiques exigent l'utilisation de stratégies réseau pour spécifier les ressources externes pouvant communiquer avec le cluster et le recours à des outils tels que les pare-feu et les pare-feu d'applications Web afin de détecter et d'empêcher toute activité suspecte. En outre, il est important de mettre à jour et de corriger régulièrement tous les contrôleurs d'entrée ou autres composants qui gèrent le trafic d'entrée pour garantir leur sécurité.

Maillage de sécurité, maillage de services

Une autre bonne pratique pour le trafic Kubernetes consiste à utiliser un maillage de services, tel qu'Istio ou Linkerd, qui fournit une couche supplémentaire d'abstraction réseau au sein d'un cluster Kubernetes. Cette couche d'infrastructure dédiée vous permet de définir et d'appliquer des stratégies de trafic granulaires pour vos microservices, y compris des règles de routage, de nouvelles tentatives et de rupture de circuit.

SUSE NeuVector : visibilité et sécurité de bout en bout pour les microservices d'entreprise

SUSE NeuVector fournit une sécurité d'exécution complète pour les applications en conteneurs, intégrée de manière transparente aux maillages de services tels qu'Istio, afin de fournir une visibilité et une sécurité de bout en bout aux microservices de votre organisation.

SUSE NeuVector permet également de gérer les vulnérabilités en suivant les dépendances utilisées pour les déploiements Kubernetes, et fonctionne avec votre système CI/CD existant pour sécuriser le pipeline. En outre, SUSE NeuVector audite la sécurité des hôtes et des conteneurs avec Docker Bench et Kubernetes CIS Benchmark pour les tests de sécurité, et fournit ainsi une attestation de conformité.

Applications

Kubernetes est utilisé pour déployer et gérer des applications dans un environnement en conteneurs. Ces applications peuvent être n'importe quel type de logiciel, y compris des applications Web, des microservices et des bases de données. Pour garantir la sécurité de ces applications, il est important de suivre les meilleures pratiques en matière de gestion des privilèges d'application, de communications de pods et de secrets.

Privilèges des pods/des applications

Les pods sont les composants de base de Kubernetes, représentant un groupe d'un ou plusieurs conteneurs déployés ensemble. Chaque pod s'exécute avec un ensemble spécifique de privilèges, déterminé par son contexte de sécurité. Pour sécuriser Kubernetes, il est important d'exécuter des pods avec le moins de privilèges possible. Cela signifie donner à chaque pod uniquement les autorisations dont il a besoin pour exécuter la fonction prévue pour lui, et pas plus.

Pour gérer les privilèges des pods, Kubernetes propose plusieurs options. L'une des options consiste à utiliser le RBAC, qui vous permet de spécifier quels utilisateurs ou groupes ont accès à des ressources spécifiques dans le cluster. Cela garantit que seuls les utilisateurs autorisés peuvent accéder à des pods ou des applications spécifiques.

Une autre option consiste à utiliser des stratégies de sécurité de pods (PSP), qui vous permettent de spécifier un ensemble de contraintes qui doivent être respectées par tous les pods du cluster. Ces contraintes peuvent inclure les privilèges minimaux autorisés pour un pod, ou les ressources minimales de processeur et de mémoire autorisées.

Secrets

L'une des meilleures pratiques de sécurité les plus importantes pour Kubernetes est la gestion des secrets. Les secrets sont des informations sensibles, telles que les mots de passe, les jetons et les certificats, utilisées par les applications pour accéder à d'autres ressources. Dans Kubernetes, les secrets sont stockés dans le cluster sous forme chiffrée et sont accessibles par les applications via l'API Kubernetes. Il est important de s'assurer que les secrets sont correctement gérés pour empêcher tout accès non autorisé et protéger les données sensibles.

Pour gérer efficacement les secrets, vous devez suivre les meilleures pratiques suivantes :

- Utilisez un gestionnaire de mots de passe pour stocker et gérer les secrets en toute sécurité.
- Utilisez le chiffrement pour protéger les secrets au repos et en transit.
- Utilisez le RBAC de Kubernetes pour limiter l'accès aux secrets uniquement aux applications et aux utilisateurs qui en ont besoin.
- Utilisez les stratégies de sécurité des pods Kubernetes pour appliquer des stratégies de sécurité sur les secrets, comme empêcher l'utilisation de mots de passe faibles ou limiter l'accès aux secrets de certains utilisateurs ou applications.

Gestion des ressources

L'un des principaux moyens d'assurer une gestion appropriée des ressources dans Kubernetes consiste à définir des quotas et des limites de ressources adaptés pour chaque application. Ces quotas et limites correspondent aux besoins en ressources attendus de l'application, en prenant en considération les pics potentiels d'utilisation des ressources. En définissant ces quotas et ces limites, vous pouvez

vous assurer qu'une application ne peut pas consommer plus de ressources qu'elle ne le devrait, ce qui peut empêcher l'épuisement des ressources et les attaques par déni de service potentielles.

Une autre bonne pratique pour la gestion des ressources dans Kubernetes consiste à surveiller étroitement l'utilisation des ressources à l'aide d'outils tels que les tableaux de bord Kubernetes ou les outils de monitoring tels que Prometheus. En surveillant l'utilisation des ressources, vous pouvez rapidement identifier les problèmes ou les attaques potentielles et prendre les mesures correctives nécessaires.

Enfin, il est important d'examiner et d'ajuster régulièrement les quotas et les limites de ressources selon les besoins. Au fur et à mesure que les besoins d'une application évoluent, les équipes peuvent trouver nécessaire d'augmenter ou de réduire la quantité de ressources allouées. En examinant et en ajustant régulièrement ces quotas et limites, vous pouvez garantir une utilisation efficace des ressources et que les applications peuvent fonctionner efficacement, sans contraintes de ressources.

Images vérifiées

Dans le contexte de Kubernetes, une image est un package préconstruit contenant l'ensemble du code, des bibliothèques et des dépendances nécessaires à l'exécution d'une application en conteneurs. Ces images sont acquises à partir de diverses sources, y compris les registres de conteneurs publics tels que Docker Hub et les registres privés créés par des organisations. Il est important que les organisations réfléchissent attentivement à la manière dont elles acquièrent leurs images, car l'utilisation d'images provenant de sources non fiables peut augmenter le risque d'introduire des vulnérabilités ou du code malveillant dans leur cluster Kubernetes.

Les images vérifiées sont des images qui ont été examinées par une source fiable pour s'assurer qu'elles sont sûres et exemptes de vulnérabilités. Ce processus de vérification peut varier, mais implique généralement un examen approfondi du code et des dépendances de l'image, afin d'identifier tout problème de sécurité potentiel. En utilisant des images vérifiées, vous pouvez être sûr que les conteneurs qui s'exécutent sur votre cluster Kubernetes ne sont pas compromis par un code malveillant ou d'autres vulnérabilités. Il est essentiel de vérifier les images que vous utilisez, car toute vulnérabilité ou tout code malveillant présent dans l'image peut compromettre la sécurité de l'ensemble du cluster.

Balisage et positionnement des pods

Le balisage est le processus d'assignation de libellés à des objets Kubernetes tels que des pods et des noeuds. Ces libellés organisent et regroupent les objets au sein du cluster, et spécifient des contraintes et des règles pour le placement des pods. Par exemple, une balise peut indiquer qu'un pod spécifique doit être placé sur un noeud avec une certaine quantité de mémoire ou de processeur disponible. Cela permet de s'assurer que les pods sont situés aux emplacements les plus appropriés au sein du cluster, et d'éviter les conflits d'accès aux ressources, ainsi que d'autres problèmes.

Analyse des vulnérabilités

L'analyse des vulnérabilités Kubernetes est un aspect essentiel de la sécurisation d'un déploiement Kubernetes. Elle implique l'utilisation d'outils et de processus spécialisés pour identifier les vulnérabilités de sécurité potentielles dans un système, telles que des logiciels obsolètes ou des paramètres mal configurés. En identifiant et en gérant ces vulnérabilités, vous

pouvez protéger vos systèmes contre les attaques potentielles et garantir l'intégrité et la sécurité de vos données.

Il existe différents outils et processus disponibles pour l'analyse des vulnérabilités Kubernetes. Il s'agit notamment de détecteurs automatisés capables d'identifier les vulnérabilités en temps réel, ainsi que de processus qui impliquent une vérification manuelle des vulnérabilités et des exploits potentiels. Les détecteurs automatisés sont particulièrement utiles, car ils identifient rapidement et efficacement les vulnérabilités potentielles, ce qui vous permet de garder une longueur d'avance sur les menaces potentielles. D'autre part, les processus manuels nécessitent plus de temps et d'efforts, mais peuvent fournir une analyse plus approfondie du système. En fin de compte, le choix de l'outil ou du processus dépendra des besoins et des exigences spécifiques de votre organisation.

Contrôle d'admission

Le contrôle d'admission désigne le processus de contrôle des applications et des autres ressources déployées sur un cluster Kubernetes. Les contrôleurs d'admission sont des plug-ins qui appliquent des règles et des stratégies pour ces déploiements. Kubernetes et SUSE NeuVector fournissent tous deux des règles et des plug-ins de contrôle d'admission qui appliquent des stratégies de sécurité et empêchent les déploiements non autorisés ou malveillants.

Le contrôle d'admission désigne le processus de contrôle des applications et des autres ressources déployées sur un cluster Kubernetes. Il s'agit d'un moyen de s'assurer que seuls les déploiements autorisés et conformes sont permis. Les contrôleurs d'admission sont des plug-ins qui appliquent des règles et des stratégies pour ces déploiements. Kubernetes et d'autres outils Open Source tels que SUSE NeuVector, Rancher et KubeWarden

L'analyse des vulnérabilités connues dans les pods est essentielle pour identifier les vulnérabilités potentielles qui peuvent être présentes dans les applications et les autres ressources contenues dans les pods. Ce processus implique d'examiner le code, la configuration et d'autres aspects du pod pour identifier les faiblesses potentielles qui pourraient être exploitées par les pirates.

fournissent des règles et des plug-ins de contrôle d'admission qui appliquent des stratégies de sécurité et empêchent les déploiements non autorisés ou malveillants. En utilisant les contrôles d'admission, les organisations peuvent s'assurer que leurs clusters Kubernetes sont sécurisés et conformes à leurs stratégies de sécurité.

Recherche de vulnérabilités connues dans les pods

L'analyse des vulnérabilités connues dans les pods est essentielle pour identifier les vulnérabilités potentielles qui peuvent être présentes dans les applications et les autres ressources contenues dans les pods. Ce processus implique d'examiner le code, la configuration et d'autres aspects du pod pour identifier les faiblesses potentielles qui pourraient être exploitées par les pirates.

Il existe de nombreux outils et techniques pour rechercher des vulnérabilités connues dans les pods. Certains de ces outils sont spécialement conçus à cet effet, tandis que d'autres sont des outils plus généraux utilisés pour identifier les vulnérabilités dans différents

contextes. Quels que soient les outils et les techniques utilisés, l'objectif est toujours le même : identifier et corriger les failles de sécurité avant qu'elles ne soient exploitées par les pirates. En analysant régulièrement les pods pour détecter les vulnérabilités connues, vous pouvez prendre des mesures proactives pour protéger vos systèmes et prévenir les failles de sécurité.

Protection des applications

La protection des applications dans un cluster Kubernetes implique à la fois des mesures proactives, telles que la mise en oeuvre de contrôles et de stratégies de sécurité, et des mesures réactives, telles que la réponse aux incidents de sécurité et l'atténuation de leur impact. L'un des aspects clés de la protection des applications est la mise en oeuvre d'une approche Zero Trust basée sur le comportement, qui implique de surveiller et de vérifier en permanence la fiabilité des applications et des autres ressources du cluster, et d'autoriser uniquement l'accès à celles qui sont fiables.

Approche Zero Trust basée sur le comportement

L'approche Zero Trust est un modèle de sécurité qui suppose que tout le trafic réseau est potentiellement malveillant, indépendamment de son origine ou de sa destination. Dans ce modèle, l'accès aux ressources réseau est accordé en fonction du comportement de la requête, plutôt que de l'identité du requêteur ou de l'emplacement de la requête. Cette approche peut aider à empêcher un accès non autorisé aux ressources Kubernetes, car elle nécessite que toutes les requêtes soient étudiées avant qu'un accès ne soit accordé.

WAF

Un pare-feu des applications Web (WAF) est un outil de sécurité qui surveille et protège les applications Web contre le trafic malveillant. Dans le contexte de Kubernetes, un WAF est utilisé pour protéger le panneau de configuration Web (le Dashboard Kubernetes) et d'autres interfaces Web contre les attaques telles que l'injection de SQL et les scripts inter-sites (XSS). Un WAF peut également être utilisé pour bloquer le trafic provenant d'adresses IP malveillantes connues, ainsi que pour surveiller et alerter sur les modèles de trafic suspects.

Prévention des pertes de données

La prévention des pertes de données (DLP) est un aspect important de la sécurité Kubernetes qui aide les organisations à protéger leurs données sensibles contre les fuites accidentelles ou malveillantes. Les solutions de DLP pour Kubernetes, incluses dans SUSE NeuVector 5.0, reposent généralement sur le monitoring et le contrôle du flux de données sensibles dans le cluster, y compris au niveau du réseau, du stockage et des applications. Cela peut inclure l'identification des données sensibles et leur balisage en tant que telles, la mise en oeuvre du chiffrement et des contrôles d'accès, ainsi que le monitoring du trafic réseau pour détecter toute activité suspecte. En mettant en oeuvre la prévention des pertes de données, les organisations peuvent mieux protéger leurs données sensibles et se conformer aux réglementations telles que la loi fédérale HIPAA (Health Insurance Portability and Accountability Act) et la norme PCI-DSS.

Accélération de la sécurité Kubernetes

Pour appliquer les meilleures pratiques de sécurité aux environnements de production Kubernetes, vous devez aspirer à suivre les conseils de ce livre blanc et des bancs d'essai CIS. Les solutions et services de sécurité jouent également un rôle essentiel dans le renforcement de la sécurité des environnements Kubernetes, ce qui permet à votre organisation d'améliorer sa capacité à analyser, identifier et répondre aux menaces et vulnérabilités de sécurité, tout en protégeant les applications et les données contre les dommages.

Il existe de nombreux outils conçus pour améliorer la sécurité de vos environnements Kubernetes, notamment SUSE NeuVector, une plate-forme de sécurité native Kubernetes qui fournit des fonctions de sécurité avancées, telles que la segmentation du réseau, la détection d'intrus et la sécurité de l'exécution. Sans oublier Rancher Prime, la version enrichie de la célèbre plate-forme de gestion Kubernetes Rancher, qui inclut également des services et un support packagés qui s'intègrent à la sécurité des conteneurs tout au long du cycle de vie de SUSE NeuVector. Avec Rancher Prime et SUSE NeuVector, vous bénéficiez d'une visibilité et d'une sécurité Kubernetes maximales sur l'ensemble de vos opérations de cluster. SUSE NeuVector peut aider votre organisation à sécuriser son environnement Kubernetes et à fournir visibilité, contrôle et protection pour les applications et les données, tandis que Rancher Prime livre des opérations et une gestion multicluster, une gestion intuitive des workloads et un support d'entreprise 24 heures sur 24 et 7 jours sur 7.

Rancher Prime et SUSE NeuVector offrent une transparence inégalée des bases de code pour les entreprises de secteurs très réglementés tels que les services financiers, la santé et l'administration.

SUSE propose également une gamme de solutions relatives à la sécurité des produits, telles que des correctifs, des alertes et des évaluations. Ces offres peuvent aider les organisations à s'informer sur les menaces et vulnérabilités de sécurité, et peuvent fournir des conseils et une assistance pour sécuriser les environnements Kubernetes des organisations qui les utilisent.

En savoir plus

Demandez une démo à l'équipe Rancher by SUSE dès aujourd'hui.

SUSE Software Solutions
Germany GmbH

Frankenstraße 146
90461 Nuremberg
Allemagne

www.suse.com

Pour en savoir plus, contactez
SUSE aux numéros suivants :

+1 800 796 3700 (États-Unis/Canada)

+49 (0)911-740 53-0 (International)

Merci