Sécuriser l'intégralité de l'environnement Kubernetes : le guide complet

Avec Prisma Cloud

paloalto® NETWORKS | PRISMA™ CLOUD BY PALO ALTO NETWORKS

Sommaire

Introduction

La plupart des discussions autour de la sécurité de Kubernetes® portent sur le défi que représente la protection des clusters. De ce que l'on sait, Kubernetes n'offre qu'une poignée de fonctionnalités de sécurité natives, ce qui explique l'extrême difficulté à protéger chaque couche de cet environnement.

En effet, Kubernetes intègre un nombre restreint d'outils de sécurité avec lesquels il faut pourtant gérer différents types de vulnérabilités sur plusieurs couches d'infrastructure. Ceci dit, sécuriser un environnement Kubernetes n'a rien d'une mission impossible.

Au contraire, de par son étendue et sa multitude d'intégrations, cette plateforme facilite la création de processus automatisés et systématiques qui placent la sécurité au cœur des développements et déploiements sur Kubernetes. Elle permet ainsi de mettre en œuvre une stratégie de sécurité étroitement intégrée pour neutraliser les menaces sur toutes les couches et à tous les niveaux de votre environnement.

Cet eBook explique comment concevoir une stratégie de sécurité qui non seulement n'entrave nullement vos processus Kubernetes, mais en plus les renforce. Il revient notamment sur les problématiques de sécurité depuis les nœuds jusqu'au reste de l'environnement, avant d'aborder les solutions pour y remédier. Il met pour cela l'accent sur des approches automatisées et évolutives, garantes de la sécurité des workloads basés sur Kubernetes – quelle que soit la taille du cluster ou son infrastructure d'hébergement (sur site, cloud public, service managé, etc.).

Chapitre 1 – Principes de base de la sécurité Kubernetes

Avant de nous pencher sur les problématiques de sécurité dans Kubernetes ou sur les stratégies permettant d'y remédier, passons brièvement en revue les facteurs essentiels à prendre en compte.

Kubernetes, un environnement multicouches

Avant toute chose, il est important de bien comprendre que Kubernetes est une plateforme complexe regroupant plus d'une demi-douzaine de composants différents. Elle intègre un serveur d'API pour faire communiquer les différentes parties d'un cluster, un planificateur pour assurer la répartition des workloads, et des contrôleurs pour gérer son propre état. Elle comprend également un agent qui s'exécute sur chaque nœud, ou serveur, au sein d'un cluster, et une base de données clé-valeur sur laquelle sont hébergées les données de configuration du cluster.

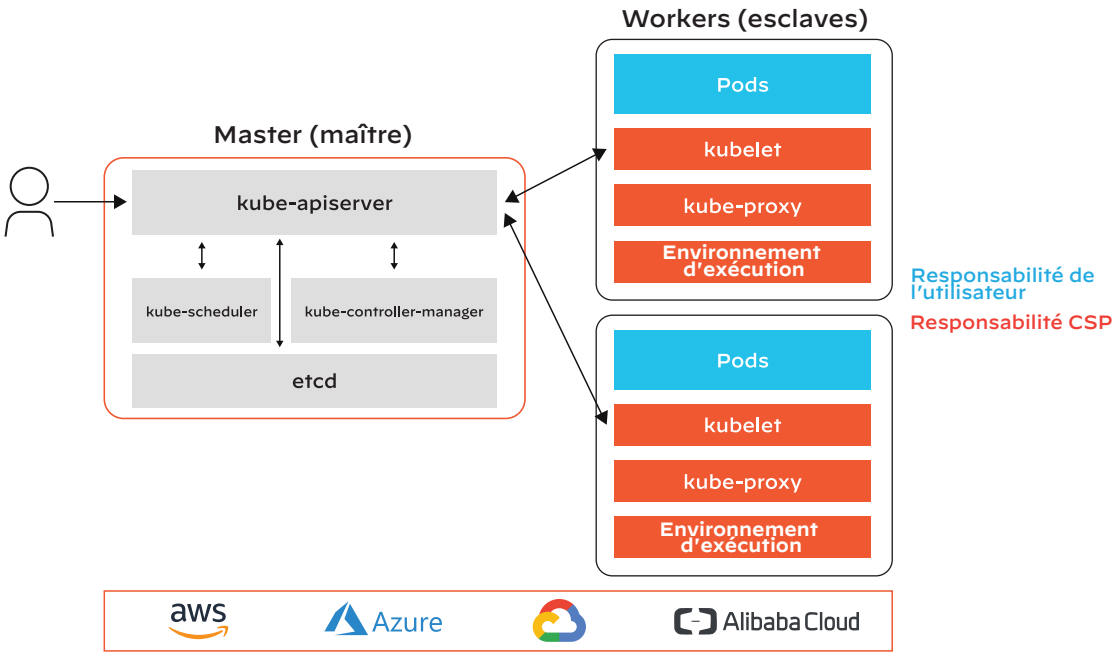


Figure 1 : Architecture des services Kubernetes managés

Il ne s'agit là que des principaux composants de Kubernetes. Pour fonctionner, un cluster Kubernetes s'appuie sur beaucoup d'autres éléments : un environnement d'exécution des containers, une solution de stockage persistant, un outil de journalisation, des systèmes d'exploitation pour chaque nœud, etc.

Or, chacune des pièces de ce puzzle apporte avec elle son lot de vulnérabilités potentielles. Une faille dans le code des environnements d'exécution peut par exemple laisser le champ libre à une escalade de privilèges dans un container. Une mauvaise configuration du serveur d'API Kubernetes peut, quant à elle, donner à des attaquants l'accès à des ressources censées être protégées. Enfin, les vulnérabilités d'une application containerisée, ou de systèmes d'exploitation exécutés sur des nœuds Kubernetes, peuvent favoriser des attaques par escalade de privilèges ou l'accès à des données sensibles. Et ce ne sont là que quelques exemples.

Bref, protéger une plateforme Kubernetes revient à sécuriser un large éventail de composants, dont chacun a des besoins qui lui sont propres. Cela étant, il n'existe aucun outil ou processus standard capable de protéger facilement tous les aspects d'un cluster Kubernetes contre tous les types de vulnérabilités. Vous devez adopter une défense à plusieurs volets.

Protections Kubernetes natives

Bien que Kubernetes intègre certaines fonctionnalités de sécurité, sa protection nécessite généralement l'intervention d'outils externes, ce qui complique nécessairement la donne.

Kubernetes permet aux administrateurs de définir des politiques de contrôle d'accès basé sur les rôles (RBAC) pour éviter tout accès non autorisé aux ressources du cluster. Ces derniers peuvent également configurer des politiques de sécurité pour empêcher certaines utilisations abusives des pods et du réseau qui les connecte. Ils ont en outre la possibilité d'imposer des quotas de ressources pour réduire les perturbations causées par la compromission d'une partie du cluster. Ces quotas empêchent notamment les attaques par déni de service (DoS) visant à priver le reste du cluster des ressources nécessaires à son fonctionnement (à condition que la compromission ne se propage pas en-dehors de la partie du cluster d'où elle provient).

De la même manière, les fonctionnalités natives de Kubernetes pallient certaines failles de sécurité au sein d'un cluster Kubernetes. Mais elles ne sont que de peu, voire d'aucune utilité face à beaucoup d'autres risques comme les exploits affectant les systèmes d'exploitation des nœuds ou les environnements d'exécution des containers.

Kubernetes permet aux administrateurs de définir des politiques de contrôle d'accès basé sur les rôles (RBAC) pour empêcher les accès non autorisés aux ressources du cluster.

L'élaboration d'une stratégie de sécurité holistique pour Kubernetes nécessite donc de se projeter au-delà des quelques fonctionnalités de sécurité intégrées à la plateforme. Car si ces dernières peuvent servir à réduire les risques de sécurité, elles sont très loin de répondre à elles seules à tous les besoins de sécurité d'un cluster.

C'est pourquoi nous vous invitons à découvrir comment combler ces lacunes dans les chapitres suivants.

Chapitre 2 – Sécurisation de l'infrastructure Kubernetes

La première couche à sécuriser dans Kubernetes est la couche de développement (ou build), à savoir l'ensemble des outils utilisés par les développeurs pour créer le code qui s'exécutera dans cet environnement.

Ces outils ne font pas partie de Kubernetes. Cependant, étant donné que la sécurité d'un cluster Kubernetes ne vaut que par celle du code qu'il exécute, sécuriser le code avant son déploiement sur un cluster est une condition préalable à la sécurisation de tous les aspects de Kubernetes.

Ce chapitre explique comment sécuriser les builds Kubernetes, avec un accent particulier sur les trois principaux points d'intégration de la sécurité à votre pipeline de développement automatisé : environnements de développement intégrés (IDE), gestion des configurations et intégration continue.

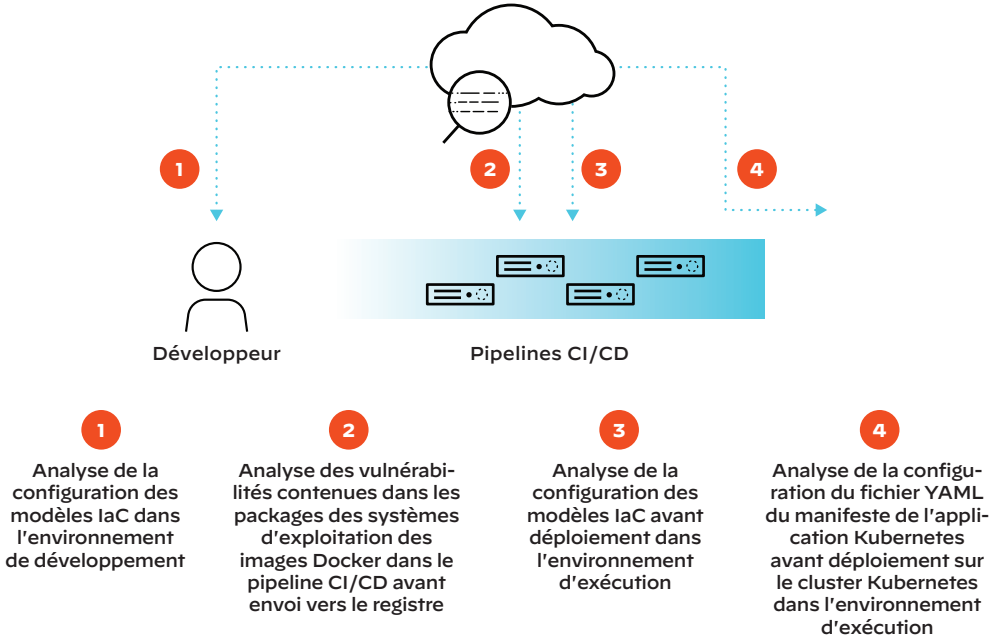


Figure 2 : Sécurité dans le pipeline de développement

IDE

Les IDE sont les outils que les développeurs utilisent généralement pour écrire le code source des applications. En tant que point de départ du pipeline de déploiement d'applications, un IDE doit également être le premier élément à faire l'objet d'une analyse de vulnérabilités. La plupart des IDE s'intègrent à divers outils tiers d'analyse des vulnérabilités chargés de détecter d'éventuelles failles de sécurité dans le code applicatif.

Intégration continue

Les outils d'intégration continue (CI) hébergent le code source et le transforment en binaires pouvant être déployés dans Kubernetes. Ils représentent une autre étape de recherche des vulnérabilités dans le code source. Tout comme les IDE, les serveurs CI sont compatibles avec divers outils d'analyse des vulnérabilités.

Gestion des configurations

Aujourd'hui, la plupart des pipelines de développement et de déploiement d'applications Kubernetes reposent sur une gestion des configurations automatisée et basée sur des politiques, sous la forme d'infrastructure IaC (Infrastructure as Code) et de fichiers YAML.

Ces approches permettent aux administrateurs Kubernetes d'écrire du code qui définit la manière dont un cluster (et l'infrastructure qui l'héberge) doit être configuré, puis d'appliquer ce code automatiquement.

En plus de rationaliser le processus de provisionnement de Kubernetes, les outils de gestion des configurations permettent de détecter les problèmes de sécurité dans les fichiers de configuration avant qu'ils ne soient appliqués. À l'image de Prisma™ Cloud, certains outils le font automatiquement en comparant vos fichiers IaC et YAML à des états de sécurité de référence. Prisma Cloud s'intègre directement à votre système de gestion du code source (GitHub®, GitLab®, etc.), ce qui facilite la création d'un processus de sécurisation des fichiers de configuration Kubernetes entièrement automatisé et compatible avec les pipelines de développement existants.

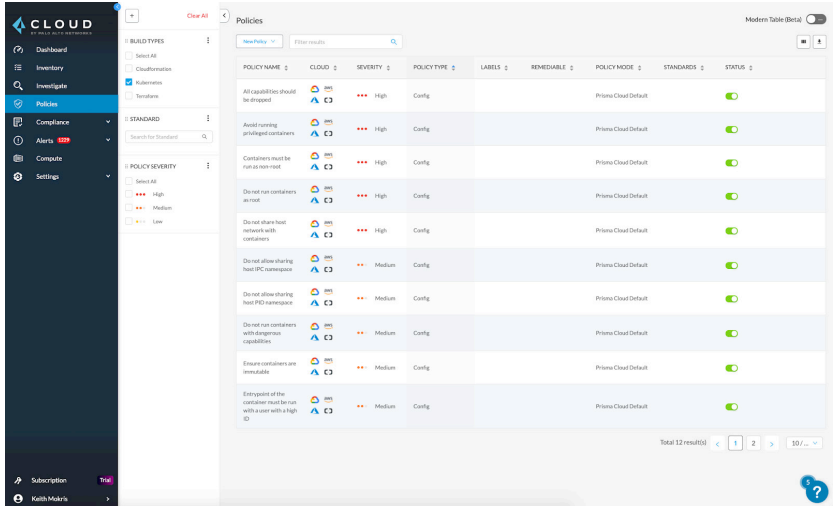


Figure 3 : Politiques Kubernetes dans Prisma Cloud

Chapitre 3 – Sécurisation des images de containers à exécuter sur Kubernetes

En général, les applications sont déployées sur Kubernetes sous forme d'images de containers. (Il est possible de gérer d'autres types d'objets de déploiement sur Kubernetes, y compris des machines virtuelles, mais cela est moins courant.) Des analyses sont effectuées sur les images de containers pour détecter les vulnérabilités dans le code du container et dans toutes les dépendances en amont de l'image.

Poste développeur

Il existe deux manières de détecter les problèmes de sécurité sur les images de containers. La première consiste à analyser les images manuellement et individuellement à l'aide d'un outil de type twistcli. Cette méthode se révèle particulièrement utile lorsque vous devez vérifier ponctuellement la sécurité d'une image.

Lorsque des images de containers sont créées à l'aide de workflows automatisés, les équipes pourront être amenées à intégrer des analyses de vulnérabilité et de conformité à ces workflows.

Intégration continue

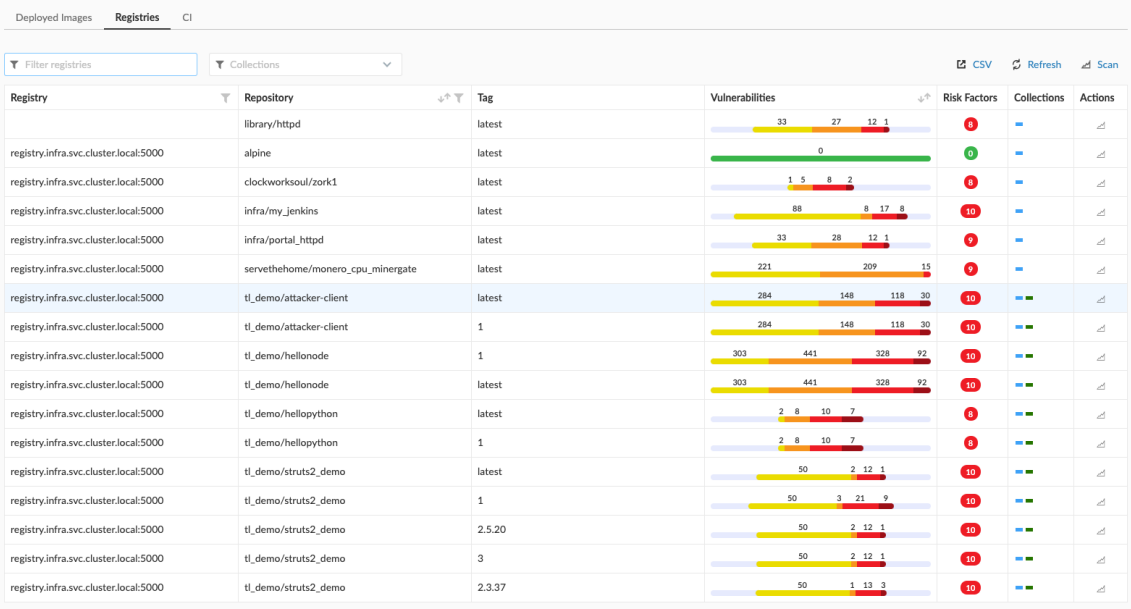
Lorsque des images de containers sont créées à l'aide de workflows automatisés sur des plate-formes comme Jenkins®, CircleCI® ou Azure® DevOps, les développeurs et les équipes DevOps pourront être amenés à intégrer des analyses de vulnérabilité et de conformité à ces workflows. À l'instar de Prisma Cloud, certaines plateformes de sécurité analysent ces images de containers pour identifier les problèmes par rapport à des frameworks (CIS Benchmark pour Docker, par exemple) et appliquer certains standards en réponse aux exigences de l'organisation ou de l'application.

Registres de containers

Une vérification automatisée et évolutive des images de containers passe par l'analyse régulière de toutes les images dans un registre de containers (référentiel stockant des images de containers). En intégrant une analyse de vulnérabilités à votre registre, vous bénéficiez d'une visibilité complète sur toutes les menaces pouvant exister dans les images stockées dans ce registre.

L'un des défis réside toutefois dans la variété des registres de containers disponibles. En effet, certaines distributions Kubernetes (ex. : Red Hat® OpenShift® et services Kubernetes managés hébergés dans des clouds publics) intègrent leurs propres registres, tandis que d'autres permettent aux administrateurs de choisir parmi toute une sélection de registres tiers.

Cette diversité d'options et de configurations souligne l'importance d'opter pour un outil d'analyse d'images de containers capable de s'intégrer à tout type de registre. Prisma Cloud offre cette flexibilité au travers d'une solution d'analyse d'images universelle et totalement indépendante de la configuration du cluster Kubernetes.



Registry	Repository	Tag	Vulnerabilities	Risk Factors	Collections	Actions
	library/httpd	latest	33 27 12 1	8		
registry.infra.svc.cluster.local:5000	alpine	latest	0	0		
registry.infra.svc.cluster.local:5000	clockworksoul/zork1	latest	1 5 8 2	8		
registry.infra.svc.cluster.local:5000	infra/my_jenkins	latest	88 8 17 8	10		
registry.infra.svc.cluster.local:5000	infra/portal_httpd	latest	33 28 12 1	9		
registry.infra.svc.cluster.local:5000	servethehome/monero_cpu_minergate	latest	221 209 15	9		
registry.infra.svc.cluster.local:5000	tl_demo/attacker-client	latest	284 148 118 30	10		
registry.infra.svc.cluster.local:5000	tl_demo/attacker-client	1	284 148 118 30	10		
registry.infra.svc.cluster.local:5000	tl_demo/hellonode	1	303 441 328 92	10		
registry.infra.svc.cluster.local:5000	tl_demo/hellonode	latest	303 441 328 92	10		
registry.infra.svc.cluster.local:5000	tl_demo/hellopython	latest	2 8 10 7	8		
registry.infra.svc.cluster.local:5000	tl_demo/hellopython	1	2 8 10 7	8		
registry.infra.svc.cluster.local:5000	tl_demo/struts2_demo	latest	50 2 12 1	10		
registry.infra.svc.cluster.local:5000	tl_demo/struts2_demo	1	50 3 21 9	10		
registry.infra.svc.cluster.local:5000	tl_demo/struts2_demo	2.5.20	50 2 12 1	10		
registry.infra.svc.cluster.local:5000	tl_demo/struts2_demo	3	50 2 12 1	10		
registry.infra.svc.cluster.local:5000	tl_demo/struts2_demo	2.3.37	50 1 13 3	10		

Figure 4 : Résultats d'une analyse de registres d'images de containers dans Prisma Cloud

Chapitre 4 – Sécurité de l'environnement d'exécution Kubernetes

L'aspect le plus complexe de la sécurité Kubernetes consiste à sécuriser une application déjà déployée dans un cluster. En effet, une application en cours d'exécution est sujette à de nombreuses vulnérabilités susceptibles d'être exploitées aussi bien via l'application que via Kubernetes.

Hormis les mesures décrites dans les chapitres précédents pour sécuriser des applications avant leur déploiement, il existe différents moyens de réduire les risques de sécurité post-déploiement.

Visibilité

Avant toute chose, il est essentiel de maintenir une visibilité continue sur vos services et ressources Kubernetes. Sachant que les compromissions peuvent venir de partout, plus vous collectez de données sur l'environnement applicatif, plus vous améliorez vos chances de détecter une anomalie symptomatique d'une compromission.

Les équipes de sécurité ne savent pas toujours où s'exécutent tous leurs clusters et peuvent, par conséquent, manquer d'informations cohérentes pour déterminer l'état global de leur sécurité. Avec Prisma Cloud, elles ont accès aux données d'API des fournisseurs de services cloud publics (CSP), ce qui leur donne une visibilité continue sur les emplacements des clusters, leur configuration et leur état de conformité.

Protection de l'environnement d'exécution

Collecter des données d'environnement à partir de Kubernetes est relativement simple. Toutefois, l'utilisation de ces données pour surveiller les problèmes de sécurité se heurte à un problème de taille : les clusters Kubernetes ont tendance à changer à mesure que les nœuds se déconnectent ou s'arrêtent, que les applications montent ou baissent en charge en fonction de la demande, etc. Résultat : il est impossible d'établir une base de référence d'une activité « normale » et d'évaluer tout écart par rapport à celle-ci.

D'où l'intérêt de protéger votre environnement d'exécution avec Prisma Cloud, une solution capable d'apprendre automatiquement la manière dont les applications déployées dans Kubernetes se comportent en fonction des conditions. Les utilisateurs peuvent ainsi faire clairement la distinction entre des changements normaux dans le comportement des applications et ceux reflétant un problème de sécurité.

Protection du réseau

Côté réseau, les menaces de sécurité peuvent impacter un environnement Kubernetes de deux manières : d'une part via les réseaux publics qui connectent les applications à Internet, et d'autre part via les réseaux internes que les containers Kubernetes utilisent pour échanger des données entre eux.

Une sécurisation efficace des ressources réseau de Kubernetes passe donc par une détection fiable des signes d'activité malveillante sur ces deux types de réseaux. Étant donné que l'activité réseau, comme le reste de l'environnement Kubernetes et les adresses IP des containers, est en état de fluctuation permanente, il vous faut un outil d'analyse capable de comprendre les nuances du trafic réseau dans un cluster Kubernetes. Il vous faut également un pare-feu qui vous permette de définir des règles de protection contre les menaces réseau, et vous alerte ou bloque automatiquement les menaces en cas de violation de ces règles. Entre une analyse du réseau orientée containers et des fonctionnalités de pare-feu sur la couche L4, Prisma Cloud agit sur tous ces fronts.

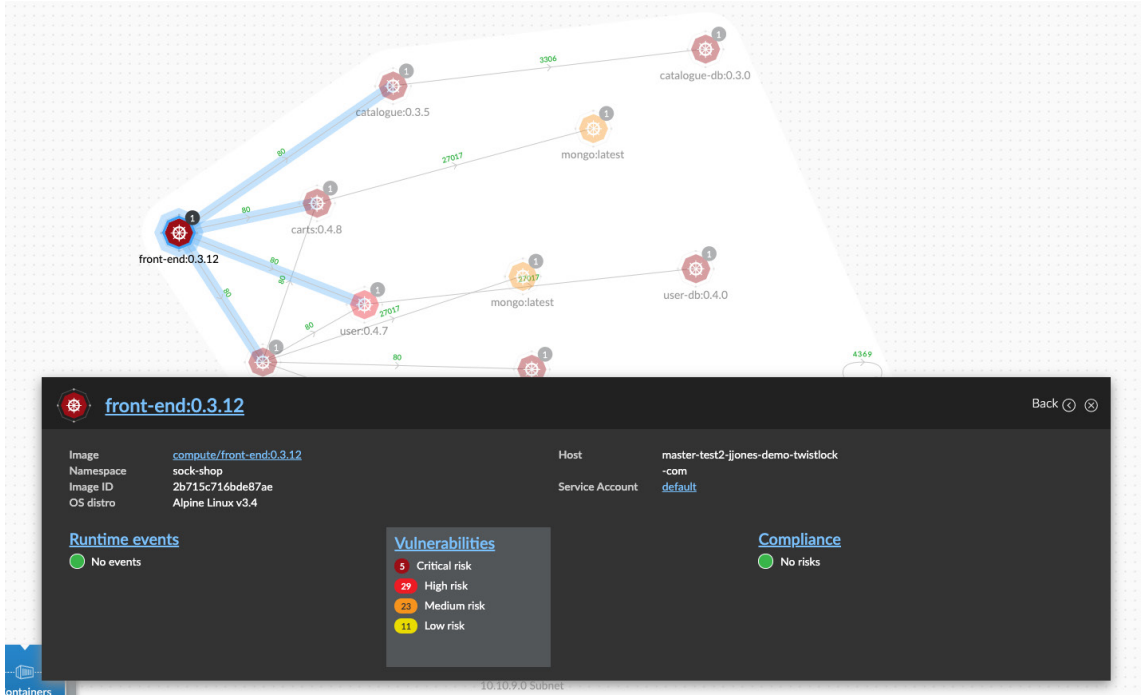


Figure 5 : Topologie du réseau et visualisation de la sécurité des containers dans Prisma Cloud

Surveillance des systèmes d'exploitation des nœuds

Un attaquant qui contrôle le système d'exploitation d'un nœud de votre cluster Kubernetes peut faire des ravages de toutes sortes. C'est pourquoi il est essentiel de surveiller non seulement les composants internes de Kubernetes, mais également les systèmes d'exploitation qui sous-tendent chacun de vos nœuds.

Dans l'idéal, vous pourrez faire tout ceci avec la solution chargée de surveiller vos applications Kubernetes. Ainsi, pas besoin de jongler entre différents outils et tableaux de bord pour détecter les menaces provenant de sources internes et externes. Prisma Cloud intègre une fonctionnalité de surveillance holistique qui vous permet de surveiller tout type de système d'exploitation, d'infrastructure cloud et d'environnement Kubernetes.

Kubernetes : sécurité, audit et conformité

Pour finir, mettez en place un processus d'audit régulier qui analyse toutes les couches de votre cluster et toutes les configurations Kubernetes pour vérifier leur conformité aux standards et bonnes pratiques en vigueur. Les audits ne permettent pas nécessairement de détecter les menaces en temps réel. Toutefois, ils vous aident à détecter proactivement les problèmes de sécurité ou les erreurs de configuration susceptibles de fournir aux attaquants un point d'entrée à votre cluster ou vos applications.

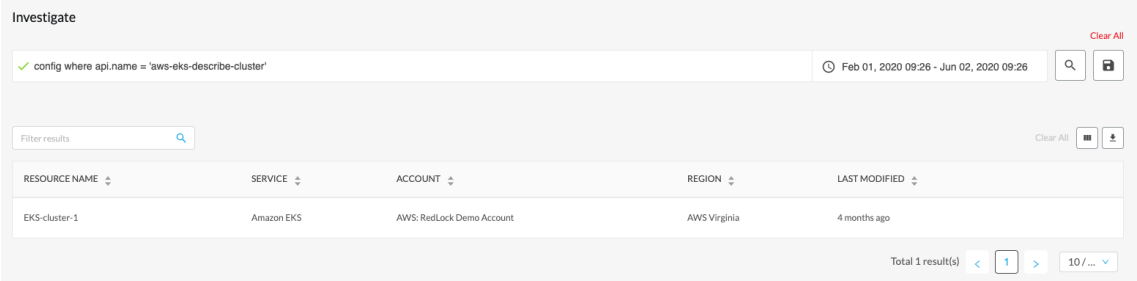


Figure 6 : Détails de l'investigation du cluster Kubernetes dans Prisma Cloud

Prisma Cloud permet d'effectuer des audits de sécurité Kubernetes complets qui passent au crible tous les composants du cluster à la recherche d'éventuels écarts par rapport aux benchmarks établis (CIS Benchmark pour Kubernetes, par exemple) et aux bonnes pratiques en vigueur. Au total, Prisma Cloud intègre plus de 100 points de contrôle personnalisables pour les configurations, les communications, etc., définis pour chaque application ou environnement.

Ces contrôles peuvent ensuite être mappés à des modèles préintégrés de conformité aux cadres réglementaires courants (PCI DSS, HIPAA, RGPD, etc.) et aux spécifications NIST SP 800-190. Vous pouvez également rédiger vos propres contrôles de conformité couvrant des technologies cloud-native comme Open Policy Agent ou Kubernetes AuditSink. Une telle approche s'avère utile si vous êtes soumis à des réglementations sectorielles spécifiques ou si vous déployez une application métier personnalisée dont les besoins de sécurité ne sont pas couverts par les politiques d'audit génériques.

Conclusion

La sécurité d'un environnement Kubernetes peut sembler une tâche insurmontable, notamment au vu de la multitude de variables concernées. Il existe pourtant un ensemble d'outils centralisés à même de traiter les différents aspects de cette sécurité. Les fonctionnalités de sécurité natives de Kubernetes vous aideront à faire une partie du chemin. Mais pour aller jusqu'au bout, vous devrez également utiliser des outils externes capables de remédier aux vulnérabilités (code malveillant dans les images de containers, par exemple), d'auditer les configurations Kubernetes pour les risques de sécurité, et de détecter les menaces en temps réel.

Prisma Cloud fournit toutes les fonctionnalités de sécurité dont les équipes ont besoin pour être présentes sur tous les fronts de la sécurité Kubernetes. Cette solution s'intègre nativement à Kubernetes et à une variété d'outils associés, ce qui facilite l'intégration de la sécurité aux processus de développement, de déploiement et de gestion de Kubernetes déjà en place.

Pour en savoir plus sur les avantages de Prisma Cloud pour la sécurité Kubernetes, rendez-vous sur notre site web.

Prisma par Palo Alto Networks

Prisma™ Cloud est la plateforme de sécurité cloud-native la plus complète du marché. Sa mission : assurer la protection et la mise en conformité de vos applications, données et technologies cloud-native tout au long du cycle de développement sur vos environnements cloud hybrides et multi-cloud.

Son approche intégrée permet aux équipes DevOps et de sécurité opérationnelle de collaborer efficacement et d'accélérer le développement d'applications cloud-native en toute sécurité.

Prisma Cloud protège et s'intègre aux architectures et outils cloud-native pour assurer une couverture complète de l'environnement, brisant au passage les silos opérationnels d'un bout à l'autre du cycle de vie des applications. Elle favorise ainsi l'adoption des processus DevSecOps et améliore la réactivité des équipes face à l'évolution des besoins de sécurité des architectures cloud-native.

Pour plus d'informations, rendez-vous sur paloaltonetworks.com/prisma/cloud.

Oval Tower, De Entrée 99 – 197
1101HE Amsterdam
Pays-Bas

+31 20 888 1883

www.paloaltonetworks.fr

prisma-cloud-complete-guide-kubernetes-ebook-093020-fr