Proposé par :

veeam

Sauvegarde et restauration Kubernetes

pour les nuls®
A Wiley Brand

Découvrez pourquoi la sauvegarde native Kubernetes est importante

Découvrez comment protéger vos applications Kubernetes

Créez des stratégies de sauvegarde automatisées pour Kubernetes

Steve Kaelble

3e édition spéciale de Veeam®

À propos de Veeam

Veeam est le leader incontesté dans le domaine de la sauvegarde et de la récupération d'urgence (DR) pour Kubernetes. Grâce à Veeam Kasten, les entreprises peuvent facilement gérer les sauvegardes en production et déployer leurs applications Kubernetes en toute sérénité. Pour en savoir plus, consultez notre page dédiée sur veeam.com/products/cloud/kubernetes-data-protection.html?ad=menu-solutions ou découvrez les dernières nouveautés de Veeam Kasten sur http:/docs.kasten.io. Retrouvez également toute notre actualité sur X à l'adresse https://twitter.com/veeam.

Sauvegarde et restauration Kubernetes

3e édition spéciale de Veeam®

par Steve Kaelble

pour les nuls®
A Wiley Brand

Sauvegarde et restauration Kubernetes pour les Nuls®, 3e édition spéciale de Veeam®

Publié par
John Wiley & Sons, Inc.
111 River St., Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-394-35368-2 (pbk); ISBN 978-1-394-35369-9 (ebk); ISBN 978-1-394-35370-5 (ePub)

Pour obtenir des renseignements généraux sur nos autres produits et services, ou sur la publication d'un livre *pour les Nuls* destiné à votre entreprise ou organisation, veuillez contacter notre service de développement commercial aux États-Unis, par téléphone au 877-409-4177, par e-mail à info@dummies.biz, ou consulter notre site www.wiley.com/go/custompub. Pour obtenir des informations sur les licences de la marque *pour les Nuls* pour des produits ou services, veuillez contacter BrandedRights&Licenses@Wiley.com.

Remerciements de l'éditeur

Cet ouvrage a été réalisé avec la participation des personnes suivantes :

Éditeur de développement : Rebecca Senninger

Rédacteur chargé des acquisitions : Traci Martin

Responsable éditorial : Rev Mengle

Agent de développement commercial : Matt Cox

Éditeur de la production : Saikarthick Kumarasamy

Table des matières

Introduction

Le cloud est devenu incontournable de nos jours ; et l'adoption des technologies cloud connaît de toute évidence une croissance fulgurante. Kubernetes s'impose désormais comme la plateforme de référence pour les entreprises. Cette plateforme sert aujourd'hui de socle à de nombreuses applications, et ce succès n'est pas le fruit du hasard. Sa portabilité, son agilité, son évolutivité et sa fiabilité exceptionnelle en font un outil idéal pour les développeurs.

Toutefois, cette plateforme n'est *pas* la panacée pour la protection de vos données. L'architecture qui fait sa force apporte aussi son lot de problèmes en termes de gestion et de sécurité. Les protections intégrées aux applications ne s'appliquent pas automatiquement aux données, ce qui complique leur sécurisation. Il ne suffit donc pas de modifier les architectures de sauvegarde traditionnelle pour les adapter à un environnement cloud.

Vous avez besoin d'une solution de sauvegarde réellement native cloud, qui ne fait pas que s'intégrer à Kubernetes, mais qui s'inscrit pleinement dans cet écosystème novateur. De plus, l'adoption des meilleures pratiques est essentielle pour assurer une transition fluide vers une utilisation optimale des technologies cloud.

À propos du livre

Cette édition spéciale de Veeam du livre *Sauvegarde et restauration Kubernetes pour les Nuls* est un guide complet qui vous accompagne dans la découverte de cette solution. Il présente les fondamentaux en retraçant l'histoire et l'évolution de Kubernetes depuis ses débuts. Vous comprendrez ainsi pourquoi les méthodes traditionnelles de sauvegarde ne sont plus adaptées aux besoins actuels et pourquoi il faut repenser nos approches en matière de protection des données.

Ce guide vous permettra d'améliorer vos compétences en matière de sauvegarde et de restauration de vos applications Kubernetes. Vous découvrirez des techniques axées sur les applications pour garantir leur capacité de restauration et leur sécurité. Vous verrez comment gérer efficacement une architecture mutualisée, comment optimiser vos processus de restauration, et comment exploiter pleinement la flexibilité qu'offrent les applications cloud.

Ce livre vous propose des solutions pratiques pour sécuriser vos données dans un environnement numérique en constante mutation, tout en simplifiant le travail des développeurs et des opérateurs. Vous y trouverez aussi des conseils pour mener une transformation en douceur, sans trop perturber l'environnement existant.

Quelques suppositions idiotes

Lors de la préparation de cet ouvrage, nous avons fait quelques suppositions concernant le lecteur que vous pouvez être.

- Vous êtes probablement quelqu'un qui a un profil technique, spécialisé dans le DevOps d'applications cloud.
- Vous êtes peut-être un commercial au sein d'une entreprise, mais vous êtes attentif aux possibilités des solutions cloud, tout en restant conscient des risques que cette technologie peut poser.
- Quoi qu'il en soit, vous êtes déterminé à faire en sorte que votre organisation réussisse dans ce domaine en constante évolution.

Icônes employées dans ce livre

Nous avons ajouté des icônes dans les marges de ce livre pour faciliter votre lecture. Ces repères visuels, qui font office d'alertes, mettent en évidence les termes importants à retenir.

RAPPEL

N'hésitez pas à parcourir le texte à votre guise, mais concentrez-vous surtout sur ces passages qui renferment des éléments essentiels.

CONSEIL

Nous vous avons promis des informations utiles. Vous les trouverez donc ici, à côté de cette icône.

ATTENTION

La sauvegarde de vos données n'est pas toujours simple. Cette icône vous indique les points sensibles qui méritent une attention particulière afin d'éviter les problèmes.

JARGON TECHNIQUE

Ces paragraphes approfondissent les questions techniques. Si vous êtes passionné par la technologie et aimez entrer dans les détails, cette section est faite pour vous !

Au-delà de ce livre

Cet ouvrage se veut avant tout un support pour stimuler votre réflexion et vous initier aux concepts clés à l'aide d'exemples concrets. Si après l'avoir lu vous souhaitez approfondir le sujet, nous vous invitons à consulter le site `https://www.veeam.com/products/cloud/kubernetes-data-protection.html` où vous trouverez des ressources complémentaires, des éléments de contexte et des livres blancs.

DANS CE CHAPITRE

- Choisir des applications cloud et conteneurisées
- Adopter Kubernetes
- Préciser les avantages
- Démystifier Kubernetes

Chapitre 1
Comprendre Kubernetes et les applications cloud

Apple a marqué les esprits avec son slogan publicitaire pour l'iPhone « Il y a une application pour ça », devenu par la suite un mème populaire. La marque avait vu juste : les applications gèrent désormais un nombre croissant de tâches quotidiennes, tant pour les particuliers que pour les entreprises. De nos jours, les développeurs d'applications adoptent massivement les architectures basées sur les conteneurs, avec des applications déployées dans l'environnement Kubernetes.

Ce chapitre traite de l'utilisation croissante des applications conteneurisées et cloud. Il aborde ensuite l'essor de Kubernetes. Les avantages d'une architecture native Kubernetes y sont détaillés en profondeur. La dernière partie démystifie certaines idées reçues sur cette plateforme, désormais reconnue comme la plus efficace pour déployer des applications métiers modernes.

La montée en puissance des applications cloud

Quelle est la place des applications conteneurs dans votre organisation ? Que nous réserve l'avenir ? Les études suggèrent que nous connaissons bien un boom des plateformes cloud, basées sur les conteneurs.

Kubernetes a été largement adopté pour sa capacité à automatiser le déploiement, l'évolutivité et la gestion des applications conteneurisées. D'après une enquête ESG d'avril 2023, Kubernetes est arrivé à maturité puisque 66 % des personnes interrogées ont affirmé qu'elles l'utilisaient déjà pour gérer et orchestrer leurs conteneurs (https://go.veeam.com/wp-ar-enterprise-kubernetes-protection).

L'enquête annuelle 2023 de la Cloud Native Computing Foundation (https://www.cncf.io/reports/cncf-annual-survey-2023/) confirme ces résultats. Kubernetes a renforcé sa position comme technologie incontournable. En effet, 84 % des répondants l'utilisent ou l'évaluent à ce jour. De plus, les implémentations de cloud hybride gagnent en popularité.

Autre résultat important, cette enquête révèle que le plus grand défi est le manque de formation. En effet, 46 % des personnes n'ayant pas encore déployé de conteneurs en production mentionnent ce problème, tout comme 28 % de celles qui les utilisent de manière limitée. Lorsque les conteneurs sont utilisés pour presque toutes les applications, la sécurité devient alors le principal défi.

Selon le rapport « Data on Kubernetes 2021 » (https://dok.community/dokc-2021-report/#:~:text=Key%20Findings,2x%20or%20greater%20productivity%20gains), le déploiement d'applications avec état, par rapport à celles sans état, est un processus bien ancré avec Kubernetes. Déjà 90 % des répondants estiment que Kubernetes est prêt pour les workloads avec état, et une grande majorité (70 %) les utilise en production, avec les bases de données en tête de liste. Les entreprises ont rapporté des avantages significatifs en termes de standardisation, de cohérence et de gestion.

Exécuter Kubernetes

RAPPEL

Kubernetes, lancé en open source en 2015, est vite devenue la plateforme de référence pour la planification et l'orchestration des conteneurs. Elle s'impose progressivement comme le socle de presque toutes les applications, peu importe l'endroit où elles sont déployées. Elle est sur le point de rejoindre Linux et vSphere en tant que plateforme privilégiée par les entreprises.

Kubernetes assure le déploiement, la gestion et l'évolutivité des applications en fonction de différentes métriques, comme l'utilisation de la CPU et de la mémoire. Son architecture repose sur des éléments fondamentaux appelés *primitives*. Ce système représente les ressources de calcul et de stockage sous forme d'objets. Les principaux objets de l'API Kubernetes sont les suivants :

- Clusters
- Nœuds
- Étiquettes et sélecteurs
- Jeu de réplication
- Déploiement

La solidité de Kubernetes donne aux utilisateurs les moyens de déployer, de mettre à l'échelle et de gérer des applications conteneurisées. Son extensibilité ainsi que sa portabilité ont grandement contribué à sa popularité dans le secteur du cloud computing. De plus, Kubernetes permet aux utilisateurs de choisir librement le langage de programmation ou l'infrastructure qui leur convient. Il offre également des outils pour surveiller les performances et consigner les erreurs.

BREF HISTORIQUE DE KUBERNETES

Revenons sur les étapes qui ont conduit Kubernetes à son succès actuel. Le terme « Kubernetes » se traduit par *timonier* ou *pilote* en grec et partage ses origines avec le mot « cybernétique ». Ce projet a vu le jour en 2014, grâce à une équipe d'ingénieurs de Google, et portait initialement le nom de Project 7.

Pour les passionnés de science-fiction, un détail mérite d'être souligné : le nom de code initial faisait référence au personnage Seven of Nine de *Star Trek*, qui appartenait autrefois au redoutable collectif Borg. Ce clin d'œil met en évidence l'influence directe d'un ancien système de Google, appelé Borg, sur la conception de Kubernetes.

RAPPEL

En définitive, cette approche contribue à améliorer la productivité, à accélérer la mise en production des applications et à réduire les coûts. Kubernetes simplifie considérablement les tâches en automatisant de nombreux processus liés au déploiement, à la gestion et à l'évolutivité. Grâce à ses fonctionnalités avancées, il devient possible de créer des clusters d'hôtes pour exécuter des conteneurs, tout en assurant leur gestion sur des environnements de cloud publics, privés ou hybrides. Vous pouvez également vous attendre à une performance exceptionnelle, avec un minimum de temps d'arrêt des applications.

Il n'est guère surprenant que Kubernetes ait connu une adoption aussi rapide. La section suivante offre un aperçu de quelques-uns de ses principaux avantages. Cependant, comme c'est souvent le cas avec les technologies qui gagnent rapidement en popularité, certains utilisateurs s'y aventurent sans en saisir pleinement les implications.

Avec Kubernetes, cette situation a engendré ce que l'on pourrait qualifier de défis de production du « Jour 2 », notamment en matière de gestion des données, de sécurité et d'observabilité. Si une grande partie des complexités liées à la haute disponibilité et à l'évolutivité des services applicatifs a été éliminée, ces avantages ne s'appliquent pas automatiquement aux données. Par conséquent, il devient essentiel de donner la priorité à la gestion des données dans vos applications Kubernetes.

Les avantages

RAPPEL

Les équipes de développement apprécient grandement l'agilité, la portabilité et la fiabilité accrues offertes par Kubernetes. Cette popularité croissante explique l'arrivée massive d'applications sur la plateforme. On y trouve non seulement des applications sans état, mais également des applications avec état, y compris des applications qui utilisent des bases de données NoSQL ou des bases de données relationnelles pour leur backend.

Voici quelques avantages de l'utilisation d'une infrastructure cloud et de Kubernetes :

- Vous pouvez facilement accéder aux ressources informatiques, au stockage et aux capacités réseau nécessaires pour accompagner une croissance rapide.
- Le stockage est facile à utiliser et le libre-service est un jeu d'enfant. Kubernetes permet d'intégrer facilement et sans difficulté aussi bien les bases de données relationnelles que NoSQL.

- Les modifications apportées aux applications conteneurisées peuvent être facilement déployées. Les applications, même complexes, peuvent être mises à jour et améliorées en un temps record.
- La plateforme est capable de s'adapter quasiment instantanément aux besoins d'évolutivité.

La figure 1-1 ci-dessous donne un aperçu de l'écosystème Kubernetes.

FIGURE 1-1 : L'écosystème Kubernetes.

Les mythes

On entend souvent que les applications déployées sur la plateforme Kubernetes doivent être conçues sans état. Au début de Kubernetes, cette idée était en partie justifiée. Cependant, les choses évoluent rapidement dans le domaine de la technologie, et ce qui était vrai autrefois relève maintenant du mythe.

Aujourd'hui, la gestion du stockage et des applications avec état est chose acquise. Kubernetes s'impose désormais comme une plateforme idéale, qu'il s'agisse d'applications avec ou sans état. De plus, pour assurer la protection de vos applications Kubernetes avec état, vous devez conserver une copie des données dans un emplacement entièrement distinct et totalement autonome.

ATTENTION

Et c'est là que se trouve le véritable enjeu. Les développeurs d'applications peuvent facilement obtenir du stockage dynamique en libre-service, ce qui est en soi une excellente chose. Cependant, on observe souvent une surutilisation du stockage dans les clusters Kubernetes, alors que ceux-ci n'ont pas été initialement pensés pour héberger des applications avec état.

L'adoption massive des bases de données relationnelles et NoSQL sur Kubernetes peut s'avérer problématique. En effet, sans une gouvernance des données appropriée, vos informations risquent d'être vulnérables, ce qui représente un véritable danger pour l'entreprise.

Ce danger n'est pas forcément visible aux yeux de tous, et c'est compréhensible. En effet, l'un des principaux atouts de Kubernetes est sa capacité à maintenir les applications disponibles même en cas de défaillances logicielles, de problèmes de serveur ou de pannes localisées. Pour les applications avec état, cette plateforme simplifie grandement l'utilisation de services de données répliqués sur différentes zones et offre ainsi une meilleure protection contre les pannes.

RAPPEL

En fin de compte, vous devez comprendre que la haute disponibilité diffère fondamentalement de la sauvegarde. Une infrastructure hautement disponible ne garantit pas l'existence d'une copie de sauvegarde de votre workload à un autre endroit. La réplication renforce certes la disponibilité des données et offre une protection contre les défaillances partielles de l'infrastructure. Toutefois, les données restent vulnérables aux pertes ou aux corruptions, qu'elles soient malveillantes ou accidentelles. Cet ouvrage propose des solutions pour assurer cette protection essentielle.

DANS CE CHAPITRE

- **Comprendre l'importance de la sauvegarde native**
- **Prendre en compte les nouveaux modèles de déploiement**
- **S'adapter au monde DevOps**
- **Soulager le stress des opérateurs**
- **Combler les lacunes en matière de protection des données**
- **Se connecter à plusieurs services de données**

Chapitre 2
Créer une protection native des données pour Kubernetes

La réussite de votre organisation dépend de la disponibilité de ses applications, et Kubernetes est formidable pour atteindre cet objectif. Cependant, que feriez-vous si vous perdiez toutes les données de votre entreprise ?

Ce chapitre explique précisément pourquoi les solutions de sauvegarde natives Kubernetes sont indispensables. De nombreux facteurs les rendent absolument essentielles, notamment les divers modèles de déploiement et les pratiques DevOps. Par la suite, vous verrez comment la sauvegarde native cloud peut simplifier le travail des opérateurs et leur donner plus de liberté pour innover. Cette solution permet aussi de renforcer la protection des données et d'assurer une meilleure préparation, même lorsque vos applications reposent sur différents services de données.

Reconnaître l'existence d'un besoin

Vous savez déjà à quel point la sauvegarde est importante. Quel que soit votre rôle dans le secteur technologique, vous vous souvenez probablement d'avoir fait au moins un mauvais rêve dans lequel vous étiez confronté à une perte de données catastrophique.


ATTENTION

De nombreux scénarios catastrophes peuvent transformer ce mauvais rêve en un véritable cauchemar. Parmi ceux-ci, on trouve la suppression accidentelle des données, une mauvaise compréhension générale de la plateforme ainsi que les ransomwares et autres activités malveillantes. Vos données sont donc confrontées à de nombreuses menaces, mais votre entreprise pourrait-elle survivre sans ces données ?


RAPPEL

Nous ne voulons pas vous faire paniquer encore plus, mais force est de constater que Kubernetes ouvre un nouveau chapitre dans le monde de l'informatique. Le problème, c'est qu'il peut aggraver les risques de perte de données déjà présents dans le cloud. Sa complexité pose un vrai défi, d'autant plus que beaucoup n'en maîtrisent pas encore tous les rouages. Sans compter que la gestion est plus décentralisée, ce qui multiplie les risques d'erreurs.


ATTENTION

D'excellents outils sont disponibles pour les environnements virtualisés, mais les systèmes non virtualisés nécessitent une approche différente. Confier la gestion des sauvegardes aux équipes applicatives paraît séduisant, mais cela crée en réalité une dispersion qui fragilise le système et prolonge le temps de restauration en cas de problème. Et on sait que le temps, c'est de l'argent. D'autant plus qu'une panne d'application critique coûte en moyenne 500 000 dollars par heure.

Vous avez donc tout intérêt à mettre en place une solution de sauvegarde native cloud. Pour sauvegarder et protéger les applications hébergées sur Kubernetes, vous aurez besoin d'une solution de sauvegarde native Kubernetes.

Gérer les différents modèles de déploiement

Kubernetes change indéniablement la donne dans le monde informatique. Ce système révolutionne les pratiques grâce à une architecture novatrice qui déplace la couche d'abstraction et offre une flexibilité accrue pour l'exécution des workloads. La plateforme Kubernetes se démarque fondamentalement de toutes les infrastructures de calcul précédentes.

Ce qui n'a pas changé, en revanche, ce sont les exigences en matière de sauvegarde. Tout administrateur se doit d'établir un plan de sauvegarde, que ce soit pour Kubernetes ou pour n'importe quelle autre plateforme.

La règle de sauvegarde des 3-2-1 est un principe incontournable pour faire face efficacement à tous les scénarios de défaillance. Cette méthode répond précisément à deux questions essentielles concernant le nombre de fichiers de sauvegarde nécessaires et leurs lieux de stockage. Les recommandations en la matière sont les suivantes :

- **3 :** vous devez conserver au minimum trois copies de vos données.
- **2 :** vous devez sauvegarder vos copies sur deux supports de stockage distincts.
- **1 :** vous devez toujours garder une copie de sauvegarde à l'extérieur de vos locaux.

RAPPEL

Commençons par le fait que vous ne mappez pas les applications conteneurisées sur des serveurs ou des machines virtuelles (VM). Les conteneurs se distinguent des machines virtuelles par leurs besoins minimaux : une portion adaptée du système d'exploitation, des programmes et des bibliothèques compatibles, ainsi que des ressources système essentielles à l'exécution d'un programme spécifique.

Par conséquent, vous pouvez mettre deux ou trois fois plus d'applications sur un seul serveur avec des conteneurs qu'avec une machine virtuelle. Les conteneurs permettent aussi de créer un environnement d'exploitation portable et cohérent pour le développement, les tests et le déploiement. De plus, Kubernetes répartit les composants d'application sur l'ensemble des serveurs grâce à sa propre stratégie de placement. Cette approche optimise donc les performances et améliore la tolérance aux pannes.

Un système traditionnel de gestion des données vous ferait probablement échouer dans ce genre de situation. Vous pourriez éventuellement réaliser une sauvegarde, mais si vous utilisez des outils qui ne sont pas conçus de manière native pour le cloud, la restauration des données deviendra compliquée.

Or, les applications cloud tirent pleinement parti de la nature dynamique de leur environnement. L'équilibrage des charges peut être optimisé en reprogrammant les conteneurs à la volée ou en les redimensionnant sur différents nœuds. Les déploiements s'enchaînent sans interruption, tandis que des composants sont constamment ajoutés ou retirés.

RAPPEL

En d'autres termes, les applications évoluent constamment. Une solution de sauvegarde adaptée aux modèles architecturaux du cloud devient alors indispensable. Cette solution doit prendre en compte la variabilité des adresses IP. Elle doit être aussi flexible que votre application cloud Kubernetes. Elle doit enfin adopter le changement avec la même agilité.

Les solutions de sauvegarde traditionnelles, parfaitement adaptées aux environnements de serveurs et de machines virtuelles, risquent de perdre leur efficacité lorsqu'elles sont déployées dans un environnement Kubernetes. En effet, pour répondre à des besoins spécifiques tels que la découverte dynamique des applications, la sauvegarde instantanée, la restauration intégrée aux plateformes et la prise en compte de l'ensemble du contexte applicatif, une sauvegarde native Kubernetes devient indispensable.

JARGON TECHNIQUE

Pour mieux appréhender cette évolution, il faut savoir que les systèmes physiques reposaient sur une approche basée sur des agents pour assurer la protection des données et du système d'exploitation. Avec l'avènement de la virtualisation, de nombreux fournisseurs de solutions de sauvegarde ont simplement adapté leurs agents pour les intégrer dans les environnements virtuels. Cette transition a engendré une surcharge des machines virtuelles, car elles continuaient d'être traitées comme des machines physiques dans ce nouvel écosystème.

La meilleure méthode pour protéger ces workloads virtuels s'est rapidement imposée au niveau de la couche de virtualisation grâce à l'utilisation des API. Cette approche permet de générer rapidement et efficacement des sauvegardes cohérentes au niveau des applications. De plus, elle préserve les performances de la machine virtuelle.

En résumé, le même scénario se répète dans l'univers de Kubernetes. Théoriquement, il est possible d'adapter vos processus de sauvegarde physique ou de virtualisation pour protéger une partie de votre environnement et de vos données Kubernetes. Toutefois, seule une fraction de ces éléments peut être protégée, ce qui complique considérablement les opérations de restauration.

Adopter le décalage à gauche (Shift Left)

La philosophie DevOps est souvent représentée par le symbole de l'infini : un « 8 » couché qui illustre une boucle continue de mouvements allant de la gauche vers la droite, vers la gauche, et ainsi de suite. On peut également la comparer à une piste de course, une analogie parfaitement adaptée, car DevOps vise à accélérer les cycles de développement des applications tout en maintenant un rythme soutenu et constant.

RAPPEL

Lorsqu'on associe DevOps à l'image de l'infini, on imagine souvent le développement positionné à gauche et les opérations à droite. Dans le contexte de Kubernetes, la philosophie DevOps établit un lien plus étroit entre les besoins des développeurs et ceux des opérations, une convergence qui n'existait pas traditionnellement. Par ailleurs, une activité accrue se concentre sur le côté gauche de cette boucle, ce qui a donné naissance au concept de décalage à gauche (Shift Left).

Kubernetes est principalement conçu pour répondre aux besoins des développeurs et de leurs applications, tout en maintenant le rythme soutenu des cycles de développement. En raison de l'architecture même de la plateforme, les solutions de sauvegarde doivent impérativement se focaliser sur les applications plutôt que sur l'infrastructure sous-jacente.

Dans cet environnement, les développeurs définissent sous forme de code non seulement les composants de l'application, mais aussi les besoins en infrastructure, comme les systèmes de stockage et les équilibreurs de charge. Les exigences liées à la protection des données doivent s'intégrer directement dans ce processus et adopter le même langage que leur code. Cette approche permet aux développeurs de configurer eux-mêmes les plannings de protection, en remplaçant ainsi la méthode traditionnelle qui confiait cette responsabilité à l'administrateur de l'infrastructure ou des sauvegardes.

CONSEIL

Vous devrez probablement intégrer directement les tâches de gestion des données dans le processus quotidien de développement. Cette responsabilité ne repose donc plus uniquement sur l'équipe des opérations. Une solution efficace pour y parvenir consiste à adopter une approche basée sur les API. La plateforme de sauvegarde doit être conçue avec une priorité donnée aux API et utiliser une API cloud.

Il s'agit ici d'API spécifiques de Kubernetes, et non d'anciennes API REST ou SOAP. Cette approche permet une authentification et une autorisation fluides, tout en offrant une intégration simplifiée des applications et des flux de travail. De plus, les développeurs et les opérateurs peuvent s'appuyer sur des outils familiers, comme kubectl, pour gérer efficacement leurs tâches.

Résoudre les problèmes des opérateurs

La virtualisation vise à abstraire le matériel physique afin de l'exploiter pour créer plusieurs machines virtuelles, principalement utilisées pour héberger des workloads monolithiques. À l'inverse, les workloads cloud sont entièrement centrés sur l'application, plutôt que sur la machine virtuelle. Cette approche simplifie considérablement la gestion en prenant l'application comme unité opérationnelle principale. En parallèle, elle s'affranchit de l'infrastructure sous-jacente et des datastores.

CONSEIL

Un outil de sauvegarde se doit d'être à la fois flexible et polyvalent dans sa façon de fonctionner. Toutes les organisations n'ont pas forcément besoin d'une API pour gérer leurs flux de sauvegarde. Certaines préféreront utiliser un tableau de bord intuitif pour les accompagner dans leur transition vers Kubernetes.

Une intégration avancée de Kubernetes peut masquer la complexité de la plateforme sous-jacente. La modernisation des flux de travail de sauvegarde pour les applications cloud permet d'optimiser l'expérience utilisateur. Cette approche élimine ou réduit alors considérablement les tâches manuelles et les efforts d'intégration.

Autrefois, une application se limitait à quelques machines virtuelles. En revanche, les applications conteneurisées actuelles comprennent généralement des centaines de ressources Kubernetes distinctes, telles que des configurations, des disques et des objets Secret.

Et cela ne représente qu'une seule application. La gestion d'un cluster implique la protection de millions de composants répartis sur l'ensemble des applications. Cette complexité devient ingérable, sauf si l'application elle-même devient l'unité opérationnelle de sauvegarde.

ATTENTION

Un système de sauvegarde traditionnelle a tendance à se concentrer sur l'infrastructure, comme les disques et les volumes, et à négliger les ressources Kubernetes. Cette approche augmente considérablement le risque d'erreurs dans les playbooks de restauration, notamment en raison de relations manquantes, ce qui entraîne des temps de restauration extrêmement longs.

Un scénario de ce type exige un processus manuel pour identifier les sauvegardes nécessaires à la restauration. La reconnexion des volumes restaurés aux applications Kubernetes demande ensuite une autre intervention manuelle plus complexe. Cette approche entraîne une charge opérationnelle considérable. La situation reste problématique même si les objets Kubernetes demeurent identiques entre le moment de la sauvegarde et celui de la restauration.

Combler les lacunes

Kubernetes excelle lorsqu'il s'agit de maintenir l'exécution des applications, même face à des pannes partielles de l'infrastructure. Bien que la tolérance aux pannes soit un atout majeur, il ne faut pas se faire de fausses idées sur le niveau de sécurité qu'elle offre. Les besoins en matière de sauvegarde et de restauration, de récupération d'urgence (DR), de mobilité des applications et de protection contre les ransomwares ne doivent en aucun cas être négligés.

RAPPEL

Encore une fois, la haute disponibilité et la réplication ne doivent pas être confondues avec la sauvegarde. Le risque de corruption ou de suppression de données demeure, qu'il s'agisse d'une erreur accidentelle ou d'un acte malveillant. Dans un tel cas, la corruption ou la suppression peut se propager à tous les réplicas et entraîner une perte de données aux conséquences potentiellement catastrophiques.

Le fait que Kubernetes fonctionne fréquemment sur des clouds publics garantit-il la sécurité du stockage face aux défaillances ? Absolument pas, c'est un mythe. Les solutions de stockage cloud les plus fiables offrent un taux de disponibilité impressionnant, mais vous restez entièrement responsable de la protection des données.

Qu'en est-il des fournisseurs de stockage local ? Ils offrent effectivement la possibilité de créer des snapshots de volume. Ceux-ci présentent néanmoins des risques en raison de leur vulnérabilité aux défaillances matérielles. De plus, en cas de suppression d'un volume, les snapshots associés sont généralement supprimés automatiquement.

C'est à ce moment-là que la règle de sauvegarde des 3-2-1, mentionnée précédemment dans la section « Gérer les différents modèles de déploiement », prend tout son sens. En suivant cette règle, vous pouvez vous prémunir contre la majorité, voire la totalité, des scénarios de défaillance. L'efficacité de cette approche est davantage renforcée lorsque vous ajoutez une protection contre l'immutabilité à vos copies hors site.

JARGON TECHNIQUE

Sans privilèges de sécurité élevés dans Kubernetes, vous ne pourrez généralement pas suspendre les activités du système de fichiers. Cependant, un système de sauvegarde Kubernetes ayant des autorisations bien définies et un contrôle d'accès en fonction du rôle permet d'accéder à des hooks. Ceux-ci sont nécessaires pour suspendre les bases de données et les workloads Kubernetes. Cette approche offre ainsi des résultats équivalents sans compromettre la sécurité.

L'important, c'est d'avoir une meilleure synergie entre les équipes de développement et d'exploitation, avec une collaboration bien plus étroite qu'auparavant. Leur objectif commun est clair : prendre les meilleures décisions pour garantir la sécurité et l'intégrité des données.

La DR est un autre aspect essentiel à ne pas négliger. Si les sauvegardes sont indispensables pour gérer des scénarios de défaillance récupérables au sein du même cluster, la DR va plus loin. Elle permet le redéploiement de vos workloads dans un environnement totalement différent.

Intégrer les écosystèmes

JARGON TECHNIQUE

La *persistance polyglotte* désigne l'utilisation de plusieurs technologies de stockage pour répondre aux différents besoins en gestion des données au sein d'une application ou de ses composants individuels. Elle ne se limite pas aux bases de données relationnelles ou non relationnelles, mais inclut également des zones de stockage comme les flux de données par lots et les files d'attente de messages.

Ces besoins variés en matière de stockage peuvent apparaître dans une entreprise qui gère plusieurs applications, ou au sein même d'une application unique où différents composants nécessitent des approches distinctes pour stocker leurs données. Avec l'expansion de Kubernetes, la persistance polyglotte devient de plus en plus courante.

Malgré cette complexité accrue, des sauvegardes plus efficaces et mieux adaptées aux workloads deviennent possibles grâce à l'intégration avec Kubernetes. La découverte automatisée des workloads permet à une solution de sauvegarde de tenir compte des exigences spécifiques de l'application et de sélectionner la primitive de capture la plus appropriée, qu'il s'agisse de snapshots de volume, de sauvegardes cohérentes au niveau des applications ou de vidages logiques.

Alors qu'un service de données unique tend à disparaître, les métadonnées Kubernetes jouent un rôle clé. Elles permettent à la solution de sauvegarde d'identifier automatiquement les relations entre plusieurs services de données indépendants.

RAPPEL

La compréhension précise de la topologie des applications permet à la solution de sauvegarde Kubernetes de capturer une copie cohérente de toute la pile applicative, tant au sein des services qu'entre eux. Les fonctionnalités comprennent notamment l'identification et la consolidation des données des réplicas, ce qui réduit l'impact sur l'application. En parallèle, l'exploitation du parallélisme natif de Kubernetes améliore les performances des restaurations pour optimiser l'efficacité globale.

Cette capacité devient d'autant plus importante à mesure que de nombreuses organisations adoptent des configurations multiclusters Kubernetes dans différents environnements. Dans ce contexte, une plateforme de sauvegarde doit impérativement s'intégrer harmonieusement à l'écosystème cloud.

RAPPEL

Cette approche ne se limite pas à une meilleure protection des données. Elle optimise l'expérience utilisateur, renforce l'efficacité des équipes sur le terrain et permet de réaliser des économies importantes. Les développeurs et les opérateurs peuvent donc continuer à utiliser des outils cloud familiers, tels que Prometheus pour la supervision et les alertes, ou les API Kubernetes pour les accès basés sur les rôles (RBAC), la journalisation et l'audit nécessaires à l'analyse approfondie des problèmes.

DANS CE CHAPITRE

- Se concentrer sur l'application dans son ensemble
- Explorer et faire évoluer l'architecture
- Garantir la capacité de restauration
- Faciliter les opérations
- Maintenir un niveau de sécurité élevé dans un environnement mutualisé
- Réussir la restauration tout en conservant sa portabilité

Chapitre 3
Respecter les meilleures pratiques pour la sauvegarde Kubernetes

Comme mentionné dans les chapitres 1 et 2, l'environnement Kubernetes constitue un écosystème à part entière. Les méthodes de sauvegarde traditionnelles ne sont plus adaptées et risquent même de ne plus fonctionner dans ce contexte.

Ce chapitre explore les meilleures pratiques pour gérer la sauvegarde des applications Kubernetes. Il met en lumière les spécificités architecturales qui imposent une approche axée sur les applications. Il décrit en détail les stratégies pour assurer une capacité de restauration fiable. Par ailleurs, il explique comment un système de sauvegarde natif simplifie les opérations, notamment lors des variations de charge des applications. La sécurité représente également un aspect crucial, bien que la gestion mutualisée puisse introduire des complexités supplémentaires. Enfin, ce chapitre aborde les défis liés à l'évolution constante des objets et des API, tout en soulignant l'importance de la portabilité des sauvegardes.

Capturer l'application

L'application reste au cœur des préoccupations, tandis que l'infrastructure agit en coulisse pour garantir son bon fonctionnement et son

adaptabilité. Cette réalité, bien que constante, revêt une importance accrue dans le contexte de Kubernetes, une plateforme qui place les développeurs et leurs applications au premier plan et qui permet d'optimiser la rapidité de développement et de mise à niveau des solutions.

Une plateforme orientée développeurs et applications exige une solution de sauvegarde qui respecte cette philosophie. Comme souligné dans les chapitres 1 et 2, une sauvegarde efficace doit être basée sur Kubernetes pour répondre aux besoins spécifiques des applications conteneurisées dans cet écosystème.


CONSEIL

L'adoption d'une approche centrée sur l'application implique une compréhension approfondie des concepts propres à Kubernetes, et non une focalisation exclusive sur l'infrastructure sous-jacente. Pour cela, il est essentiel de garantir une capture exhaustive de l'application. La capture doit assurer une protection complète de tous ses composants. Elle ne doit laisser de côté ni ressource, ni filtre, ni étiquette.

Tirer parti de l'architecture

Ce livre s'attache exclusivement à la sauvegarde Kubernetes, un élément clé pour garantir le succès sur cette plateforme. Nul besoin d'explorer en détail toute l'architecture de Kubernetes pour atteindre cet objectif. Cependant, vous devez prendre le temps d'étudier les bases de cette architecture pour mieux concevoir et déployer une stratégie de sauvegarde efficace.

Connecter les composants

La figure 3-1 décrit les éléments essentiels qui composent une application Kubernetes, notamment les pods, les services, les certificats, les objets Secret ainsi que les volumes persistants.

Dans un environnement de production, une application peut compter des centaines de composants, ce qui soulève deux questions majeures : comment assurer une protection et une restauration optimales des données et de l'ensemble des éléments internes ? Et plus important encore, quelle approche adopter pour déployer efficacement une telle stratégie de sauvegarde et de restauration à grande échelle ?


CONSEIL

Ne vous inquiétez pas, vous n'avez pas à tout gérer vous-même. La solution de sauvegarde s'intègre naturellement à Kubernetes grâce à son serveur API, ce qui lui permet de détecter automatiquement toutes les applications Kubernetes qui tournent sur le cluster. Elle se charge aussi de protéger l'ensemble de l'infrastructure (serveurs, réseau et stockage) pour garantir une sauvegarde complète.

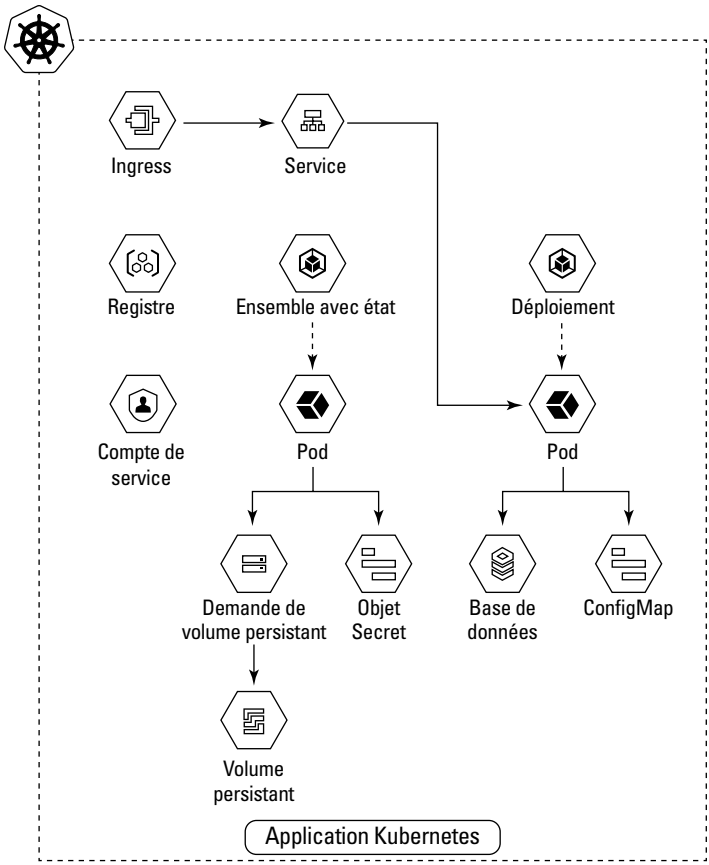


FIGURE 3-1 : Une application Kubernetes de base se compose d'un ensemble de composants interdépendants. Chacun de ces composants doit être intégré de manière cohérente au système de sauvegarde et de restauration.

La première étape consiste à bien comprendre comment les applications interagissent avec leur espace de stockage. Une fois ce lien établi, nous pourrons choisir la meilleure approche pour extraire les données des applications hébergées sur les volumes persistants, sans oublier de prendre en compte toutes les ressources liées à ces applications. La réussite de l'opération repose sur une gestion rigoureuse et harmonieuse de chaque élément du processus.

La prochaine étape consiste à choisir où stocker vos sauvegardes. Vous pouvez opter pour un système de stockage optimisé pour une restauration rapide, ou pour des snapshots durables si vous utilisez l'un des principaux fournisseurs de services cloud. Toutefois, nous conseillons généralement de garder vos sauvegardes dans un stockage en mode

objet situé dans une zone distincte. Cette solution permet aussi la géo-réplication pour mieux gérer la récupération d'urgence (DR) en cas d'incident. Mieux vaut être prudent et anticiper que d'avoir à réparer les dégâts plus tard.

CONSEIL

La règle de sauvegarde des 3-2-1, expliquée au chapitre 2, reste une référence incontournable pour planifier efficacement vos sauvegardes. Cette règle préconise de conserver au moins trois copies de vos données, de les répartir sur deux supports différents, et de stocker l'une de ces copies hors site. Cette approche est essentielle pour ne pas exposer vos données à des risques considérablement plus élevés.

Deux aspects supplémentaires méritent réflexion lorsqu'il s'agit d'intégrer le stockage dans Kubernetes :

- Le stockage s'effectue sur des volumes persistants accessibles aux conteneurs. Vous devez protéger ces données, car elles représentent un actif stratégique pour l'entreprise.
- Comment allons-nous stocker ces données ? Le stockage local en mode bloc peut être une solution. Sinon, nous pouvons opter pour une solution de stockage en mode objet comme Amazon S3 ou en mode objet blob Azure Microsoft, que notre infrastructure Kubernetes soit hébergée localement ou dans le cloud. Pour choisir le système de stockage secondaire idéal pour nos sauvegardes, il s'agit de privilégier la flexibilité, la variété des options disponibles et la simplicité d'utilisation.

La liste des tâches à accomplir dans ce domaine comprend donc les éléments suivants :

- Comprendre la relation entre les applications et leur stockage.
- Déterminer comment capturer les données d'application sur les volumes persistants.
- Décider de l'endroit où conserver la copie de sauvegarde pour veiller à respecter la règle des 3-2-1.

RAPPEL

Peu importe la solution adoptée pour sécuriser vos applications Kubernetes, celle-ci doit pouvoir détecter automatiquement tous les éléments en cours d'exécution sur le cluster. Elle doit également traiter l'application comme un ensemble indivisible. Pour une protection complète, l'application doit inclure l'état réparti sur tous les volumes de stockage et toutes les bases de données, mais aussi toutes les données de configuration stockées dans les ressources Kubernetes, notamment les objets ConfigMap et Secret.

Monter en charge

Une application qui s'exécute sur Kubernetes dissimule une complexité impressionnante. Grâce à son architecture en microservices et aux nombreuses fonctionnalités de Kubernetes, les applications comportent des centaines d'éléments qui fonctionnent de manière autonome. La plateforme prend en charge de nombreux aspects, notamment la gestion des configurations et des objets Secret. Toutefois, cette complexité reste largement masquée aux yeux des utilisateurs. En effet, seul Kubernetes en a une vision complète.

Une solution de sauvegarde native cloud est indispensable pour gérer les millions de composants présents dans les grands clusters Kubernetes. Elle doit être capable de comprendre les relations entre les applications, les données qu'elles utilisent et l'état Kubernetes associé. Seule une solution véritablement native cloud peut capturer et gérer ces éléments de manière cohérente et efficace à grande échelle.

RAPPEL

À l'instar de Kubernetes et des applications natives cloud qui s'adaptent naturellement aux fluctuations de charge, les systèmes de sauvegarde doivent faire preuve de la même souplesse. Voici ce que vous devez impérativement exiger de votre solution de sauvegarde :

- Elle doit adopter le même modèle architectural cloud afin de pouvoir évoluer en fonction des changements dans les applications et les clusters.
- Sa charge doit pouvoir être inexistante, si elle n'est pas utilisée.
- Elle doit effectuer toutes ces opérations automatiquement, sans intervention manuelle de l'opérateur.

RAPPEL

Une plateforme de sauvegarde capable de s'adapter en temps réel à la croissance du cluster offre des performances optimales. Elle permet également de réaliser des économies, car son empreinte en termes de ressources est ajustée aux besoins actuels, qui peuvent varier instantanément, au lieu d'être dimensionnée pour les pics de charge. De plus, cette évolutivité linéaire, alignée sur la croissance de l'application et du cluster, apporte une flexibilité bien supérieure. Contrairement à un modèle basé sur une appliance, elle évite les sauts brusques de capacités ou de fonctionnalités.

Les problèmes de montée en charge deviennent encore plus complexes avec l'utilisation grandissante de Kubernetes dans les environnements multiclusters. Un cluster unique peut héberger des milliers d'espaces de noms, qui contiennent chacun des centaines de ressources Kubernetes.

Vous avez des clusters répartis sur différents environnements de développement, de test et de production. Mais vous avez aussi des clivages entre les applications, la sécurité et les équipes. Cette infrastructure

s'étend sur plusieurs *availability zones* (zones de disponibilité), régions géographiques, clouds et datacenters locaux, ce qui rend leur administration particulièrement complexe.

La gestion manuelle de tous ces clusters représente une tâche colossale et, soyons francs, cette tâche est quasiment impossible sans une plateforme de sauvegarde cloud native : une solution capable de gérer les environnements multiclusters, d'offrir une vue d'ensemble complète et de garantir une administration centralisée et performante.

Voici les étapes essentielles pour assurer l'évolutivité de la solution :

- Assurez-vous que votre solution de sauvegarde peut être dimensionnée en fonction des applications qu'elle protège.
- Identifiez comment votre solution de sauvegarde peut relever les défis liés au multiclustering dans Kubernetes.

Planifier la capacité de restauration

La capacité de restauration est le véritable sujet traité dans ce livre. Elle exige une planification rigoureuse et une exécution précise, ce qui vous oblige à aller au-delà de la simple recréation d'objets Kubernetes et de volumes de stockage.

CONSEIL

Votre application conteneurisée est, par nature, complexe et constituée de nombreux composants Kubernetes. La première étape consiste à élaborer un plan d'exécution capable de gérer les aspects suivants :

- Vérifier les dépendances du cluster.
- Créer de nouvelles vues Kubernetes des données à restaurer.
- Identifier l'infrastructure de calcul et le cluster Kubernetes où la restauration sera lancée, par exemple dans le cadre d'une récupération entre *availability zones*.

Une fois le plan défini, vous devez identifier les sources des données de sauvegarde, comme le stockage en mode objet, les snapshots et les sauvegardes existantes. Vous devez également préparer le stockage de destination, ce qui peut nécessiter une redéfinition des classes de stockage et divers réglages si vous changez de système de stockage.

CONSEIL

Évaluez si le plan doit être ajusté pour répondre aux exigences de la restauration. Intégrez des éléments comme la régénération des certificats TLS, les modifications des configurations DNS et la mise à jour des objets Secret obsolètes. Une fois ces ajustements effectués, mettez à jour les composants Kubernetes de l'application afin de les aligner avec les nouvelles ressources de stockage générées par le processus de restauration.

Une fois la planification terminée, la plateforme de sauvegarde doit convertir le plan en appels d'API Kubernetes adaptés pour générer les ressources nécessaires. Par exemple, elle peut envoyer des appels pour créer un équilibreur de charge ou recréer un objet Secret. Tous les microservices et toutes les ressources qui composent une application cloud doivent être redéployés avec les configurations appropriées. La figure 3-2 illustre ce processus de manière visuelle pour mieux comprendre chaque étape.

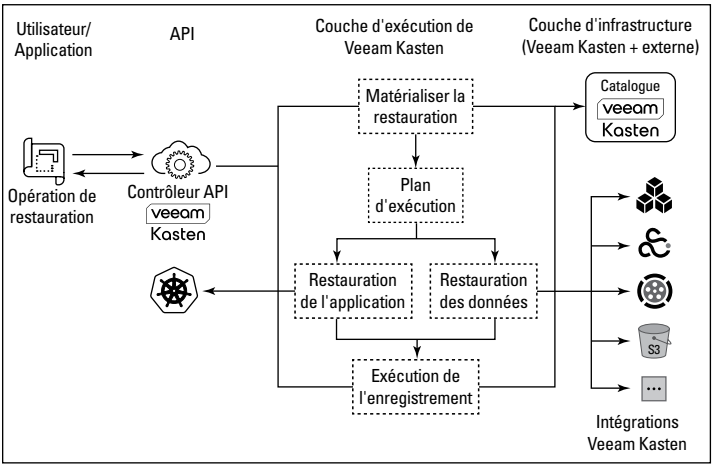


FIGURE 3-2 : Aperçu du processus de restauration, y compris les appels API qui créent toutes les ressources nécessaires.

L'objectif principal est de restaurer tous les composants de l'application à l'emplacement de votre choix. La granularité offerte devrait vous permettre de ne récupérer qu'un sous-ensemble d'applications si nécessaire, comme un volume de données spécifique. Votre solution de sauvegarde doit également vous permettre de sélectionner, à tout moment, la copie la plus adaptée de l'application. Est-ce trop demander de simplifier ce processus au maximum ? Pas du tout.

En résumé, la DR repose sur une planification rigoureuse combinée à l'utilisation d'un outil adapté pour assurer une exécution efficace :

- Créez votre plan d'exécution en vous concentrant sur les dépendances des clusters, les vues des données et l'endroit où la restauration sera initiée.
- Identifiez les sources de données et le stockage de destination.
- Déterminez si une transformation est nécessaire et comment les composants seront mis à jour.
- Transformez tout cela en appels API Kubernetes.

Mettre l'accent sur les opérations

Rappelez-vous ce qui rend Kubernetes si populaire : sa rapidité et sa simplicité offrent aux développeurs la capacité de déployer des applications. Ils peuvent ensuite les mettre à niveau et les faire évoluer facilement. La dernière chose à faire serait d'adopter une plateforme de sauvegarde qui entrave cette efficacité. Une telle situation serait frustrante et pourrait pousser les développeurs à contourner les processus et les meilleures pratiques.

CONSEIL

S'assurer que chacun respecte bien les procédures demeure un vrai défi, surtout lors du déploiement de nouveaux outils, services ou fonctionnalités dans une infrastructure qui évolue constamment. Votre plateforme de sauvegarde native Kubernetes doit :

- pouvoir être utilisée à grande échelle ;
- donner aux équipes opérationnelles les fonctionnalités de flux de travail dont elles ont besoin ;
- répondre aux différentes exigences en matière de conformité et de supervision ; et
- être conçue pour fonctionner de manière transparente, sans créer de complications pour les développeurs.

Pour les équipes de développement, la plateforme doit s'intégrer naturellement dans leurs habitudes de travail, sans nécessiter de changements dans leur code, leurs packages, leurs outils ou leurs méthodes de déploiement. En même temps, les équipes d'opérateurs doivent pouvoir mettre à disposition des développeurs des fonctionnalités en libre-service.

RAPPEL

Par exemple, les développeurs doivent non seulement pouvoir restaurer les applications à leur guise, mais aussi adapter les processus de sauvegarde selon les spécificités de leurs services. La gestion des dépendances entre services et leur suspension temporaire doit être sous leur contrôle, qu'ils choisissent d'utiliser leurs propres outils ou ceux proposés par des fournisseurs de bases de données. L'ensemble de ces opérations doit être accessible via des API pour garantir une intégration simple et une grande souplesse d'utilisation.

Pour les opérateurs, la meilleure approche en matière de sauvegarde est celle qui les libère de la gestion minutieuse des centaines de composants Kubernetes que comportent une application. Ils privilégient des solutions de sauvegarde automatisées qui leur offrent une vision globale de l'application, sans avoir à se soucier des ressources individuelles ou des spécificités techniques du stockage.

La stratégie détaillée et l'identification des éléments à sécuriser dans l'application doivent se faire automatiquement pendant l'exécution. Une fois mise en place, la solution doit s'adapter d'elle-même aux évolutions de l'application, sans nécessiter d'interventions manuelles pour prendre en compte les nouveaux composants ou leurs modifications, qui sont inévitables dans un environnement en constante évolution.

LES TÂCHES DE SAUVEGARDE ET LES STRATÉGIES INTELLIGENTES

Quels sont les avantages des stratégies de sauvegarde intelligentes par rapport à la configuration de tâches de sauvegarde spécifiques ? Comparons-les pour y voir plus clair.

Les tâches de sauvegarde exigent une configuration précise pour décrire chaque étape de la pile logicielle, y compris l'heure d'exécution. Elles ne prennent pas beaucoup en compte l'architecture globale du datacenter et nécessitent une surveillance continue pour éviter qu'elles n'affectent les workloads ou ne saturent les ressources réseau.

Les stratégies, en revanche, adoptent une approche axée sur les résultats. Plutôt que de planifier des actions spécifiques, vous définissez un délai optimal de reprise d'activité ou un résultat à atteindre. Une fois configurée, la stratégie s'exécute de manière autonome, surveille en permanence les besoins et les problèmes éventuels, et s'assure que l'objectif souhaité est toujours respecté.

Vous pouvez ainsi distinguer clairement quelle solution offre plus d'automatisation et laquelle nécessite une plus grande implication manuelle. Cette intervention humaine n'est pas seulement chronophage, elle augmente aussi les risques d'erreurs, surtout dans les cas complexes où de nombreux réglages et suivis sont nécessaires.

CONSEIL

Une autre bonne pratique pour maximiser la simplicité consiste à créer des stratégies de sauvegarde globales basées sur des étiquettes. Celles-ci peuvent automatiquement détecter et prendre en charge de nouvelles applications dès leur déploiement.

Par exemple, vous pouvez configurer une stratégie qui s'applique automatiquement à l'ensemble des applications MongoDB ou à celles déployées avec Helm. Cette approche évite aux équipes opérationnelles de devoir créer des processus manuels pour gérer les modifications. De plus, elle garantit que toutes les applications respectent les contrats de niveau de service de sauvegarde, qu'elles soient nouvellement créées ou supprimées.

En résumé, optez pour une solution de sauvegarde capable de répondre aux attentes et aux besoins spécifiques de chaque utilisateur. Il est tout à fait possible de déployer une plateforme qui satisfait à la fois les exigences des équipes chargées des opérations conteneurs et celles des développeurs. Insistez sur ce point.

En ce qui concerne les opérations, voici les principales tâches à accomplir :

- Veillez à ce que votre solution de sauvegarde soit réellement une solution, non un problème. Elle doit stimuler l'efficacité, répondre aux besoins en matière de flux de travail et répondre aux exigences.
- Identifiez où des stratégies intelligentes peuvent résoudre les problèmes et évitez de devoir créer des tâches de sauvegarde spécifiques.
- Créez des stratégies capables de découvrir par elles-mêmes de nouvelles applications alignées sur leur objectif.

Garantir la sécurité

Que vous déployiez des solutions dans un cloud public, une infrastructure locale ou un environnement hybride, la sécurité doit être au cœur de vos préoccupations. Les gros titres des journaux rappellent chaque jour à quel point elle est essentielle. De plus, l'essor de la gestion mutualisée renforce l'importance d'une vigilance accrue en matière de sécurité.

Appliquer plusieurs couches de protection

Heureusement, Kubernetes propose de nombreuses fonctionnalités de sécurité, comme les stratégies réseau qui protègent les composants internes des applications ainsi que les services de données associés. Non seulement Kubernetes empêche l'accès à ces composants depuis l'extérieur du cluster, mais il complique également la tâche des applications non fiables exécutées au sein du même cluster.

RAPPEL

Cette approche présente un avantage certain, mais elle implique également une contrainte, car vous ne pourrez pas exécuter des solutions de sauvegarde en dehors des clusters Kubernetes. Ces solutions ne pourront ni découvrir les applications ni interagir avec elles, comme effectuer une suspension, sans compromettre les stratégies d'isolation. La solution idéale repose sur une architecture native Kubernetes bien conçue, capable de s'intégrer directement au plan de contrôle.

Un autre défi auquel vous pourriez être confronté est la nécessité d'offrir des fonctionnalités en libre-service, car les développeurs prennent de plus en plus en charge la gestion de l'infrastructure. Le déploiement de

votre système de sauvegarde doit inclure des contrôles stricts sur la gestion des identités et des accès (Identity and Access Management). Le contrôle d'accès en fonction du rôle (RBAC) est indispensable. Il définit de manière précise les droits d'accès, les restrictions et les privilèges accordés aux différents utilisateurs et groupes sur la plateforme de sauvegarde.

En fait, votre accès limité doit s'appuyer sur les rôles et outils déjà en place dans Kubernetes. C'est préférable à l'ajout de nouveaux systèmes de gestion des droits que vos équipes devraient apprendre à maîtriser.

Pour l'équipe des opérations, une approche basée sur le principe du moindre privilège est recommandée pour les tâches quotidiennes, comme la supervision des sauvegardes, la vérification de leur succès et de leur intégrité, ainsi que la gestion des restaurations demandées. Si nécessaire, vous pouvez définir des cas d'usage spécifiques. Par exemple, vous pouvez permettre aux développeurs de réaliser des restaurations rapides et des clones à partir de snapshots, tout en réservant l'accès aux sauvegardes stockées hors site à un groupe restreint d'utilisateurs.

De plus, votre plateforme de sauvegarde native cloud doit s'intégrer parfaitement aux systèmes de gestion des identités et des accès, ainsi, qu'aux systèmes de gestion des clés et des certificats hébergés dans le cloud. Par ailleurs, un système de gestion des données véritablement basé sur Kubernetes doit pouvoir se connecter directement à la solution d'authentification du fournisseur de services cloud, sans nécessiter de gestion supplémentaire des utilisateurs ou des groupes, ni l'ajout de nouveaux outils ou de nouvelles API pour la gestion des stratégies d'accès basés sur les rôles (RBAC).

CONSEIL

Gardez à l'esprit que Kubernetes délègue le chiffrement des données au système de stockage sous-jacent ainsi qu'à la plateforme de sauvegarde. Vous devez vous assurer que les données de votre application ne sont jamais stockées ni transférées en texte clair. À cette fin, la plateforme de sauvegarde doit :

- Comprendre la gestion des certificats Kubernetes
- Collaborer avec des systèmes de gestion des clés intégrés au stockage
- Prendre en charge les clés de chiffrement gérées par le client via l'interface des secrets Kubernetes

Prenons l'exemple du stockage en mode objet : si une application Kubernetes déployée localement envoie ses sauvegardes vers AWS S3, les données transiteront via une connexion Internet externe. La plateforme de sauvegarde doit donc garantir leur chiffrement à l'aide d'un protocole comme TLS.

Le chiffrement ne se limite pas au transfert des données, il doit aussi s'appliquer à leur stockage. Si elles ne sont pas protégées au repos, elles restent vulnérables, quel que soit l'endroit où elles se trouvent. Mettre en place un système d'accès basés sur les rôles (RBAC) et des stratégies associées ne suffit pas ; vous devez absolument utiliser des algorithmes de chiffrement éprouvés, comme l'AES-256-GCM, avec des clés dédiées à chaque application. Sans ces précautions, le risque de fuite accidentelle ou de copie malveillante est bien réel.

ATTENTION

Face aux menaces malveillantes, votre système doit en permanence surveiller les logiciels malveillants et les ransomwares qui visent Kubernetes. Le risque s'intensifie, notamment parce que de nombreuses applications Kubernetes interagissent directement avec des clients externes.

Dans cette logique, assurez-vous d'utiliser la fonctionnalité d'*immutabilité*. Elle empêche toute modification ou suppression de vos sauvegardes stockées en mode objet, y compris en cas de tentative de chiffrement ou de suppression malveillante. Cette mesure est essentielle pour contrer efficacement les ransomwares.

En réunissant tous ces éléments, il devient évident qu'une solution de sauvegarde doit être à la fois native de Kubernetes et capable de créer des sauvegardes fiables, indépendantes de Kubernetes et du système de stockage. Elle doit également offrir des intégrations avancées pour garantir des restaurations rapides et entièrement automatisées.

La sécurité est une tâche essentielle, et voici quelques éléments clés à prendre en compte :

- Intégrez une solution de sauvegarde native de Kubernetes dans le plan de contrôle.
- Créez des fonctionnalités en libre-service qui ne compromettent pas la sécurité.
- Veillez à ce que votre solution offre toujours un chiffrement et d'autres couches de protection.

Vivre avec la gestion mutualisée

Les solutions de sauvegarde traditionnelles sont souvent conçues pour les administrateurs, ce qui limite leur accessibilité. Dans la plupart des cas, seules quelques personnes maîtrisent l'ensemble du système. Certains utilisateurs peuvent uniquement consulter les sauvegardes, tandis qu'un nombre restreint possède des autorisations complètes. Pour restaurer des données, vous devez alors souvent passer par une demande officielle ou solliciter l'intervention d'un tiers.

Les clusters Kubernetes fonctionnent généralement en mode mutualisé, avec des développeurs et des équipes qui rejoignent ou quittent

constamment le système. Chacun dispose de ses propres espaces et applications et doit respecter des limites claires pour éviter toute interférence avec les autres.


CONSEIL

Dans cet environnement, une solution de sauvegarde doit être en libre-service pour permettre aux développeurs de gérer le réseau, le pare-feu, le provisionnement du stockage et les restaurations, mais uniquement pour leurs propres applications. Le système doit être strictement délimité. Chaque développeur ne doit avoir accès et ne doit pouvoir voir que les applications dont il est responsable.

Par exemple, un développeur responsable de l'application X ne doit avoir accès qu'à cette application, sans aucune visibilité sur l'application Y, qui appartient à un autre développeur. Inversement, le développeur de l'application Y ne doit ni voir ni accéder à l'application X.

Comment y parvenir ? L'un des éléments clés est le contrôle d'accès basé sur les rôles, comme évoqué plus haut dans ce chapitre. Certains développeurs doivent avoir les autorisations nécessaires pour effectuer des sauvegardes et des restaurations, tandis que d'autres ne peuvent que restaurer des données sans pouvoir les sauvegarder. Certains n'ont qu'un accès en lecture seule. L'objectif est de définir plusieurs groupes avec des niveaux d'accès adaptés à leurs responsabilités.

Comment automatiser ce processus ? Des outils comme l'Open Policy Agent (OPA) permettent de définir et d'appliquer des stratégies sous forme de code.


CONSEIL

L'essentiel est d'adopter une approche native pour Kubernetes. Il ne s'agit pas d'ajouter un système externe d'accès basés sur les rôles (RBAC) pour gérer les utilisateurs, mais plutôt d'intégrer ces contrôles directement dans la plateforme. Le système de sauvegarde doit tirer parti des API existantes afin d'éviter toute gestion supplémentaire ou l'ajout d'outils tiers. Toute autre approche risque d'alourdir inutilement l'infrastructure.

Voici les principales préoccupations que vous devez avoir en matière de gestion mutualisée :

- Assurez-vous que les options en libre-service restreignent l'accès des développeurs uniquement à leurs applications.
- Privilégiez la cohérence afin de ne pas avoir à gérer des systèmes distincts pour la gestion des utilisateurs.

Transformer le support de restauration

Kubernetes évolue à un rythme soutenu, avec une nouvelle version publiée environ tous les trois mois. Les objets peuvent donc changer rapidement. De plus, les organisations ne mettent pas toujours à jour

leur infrastructure à chaque version et peuvent sauter plusieurs mises à niveau d'un coup. À cela s'ajoute la nécessité de garantir la portabilité propre aux environnements basés sur les conteneurs Kubernetes. Comment concilier ces enjeux et assurer un fonctionnement fluide ?

Évoluer au rythme des changements

ATTENTION

Dans un contexte où les mises à jour se succèdent à un rythme effréné, il n'est pas rare de constater que nos sauvegardes, vieilles de quelques mois voire d'une année, ne parviennent plus à identifier certains objets qui ont évolué entretemps. Ce problème ne concerne pas seulement les API Kubernetes, mais aussi les ressources personnalisées et d'autres composants essentiels. Si vous essayez de restaurer avec une ancienne version d'API qui n'est plus prise en charge, la restauration échouera et vous ne pourrez pas récupérer vos données.

CONSEIL

C'est pourquoi le système doit être capable d'adapter et de convertir les descripteurs et caractéristiques des composants issus d'anciennes versions vers des versions mises à jour. Il doit, par exemple, pouvoir faire évoluer l'API de la version précédente vers la version plus récente. Il doit pouvoir gérer un certificat à durée de vie limitée ou encore actualiser des objets Secret nécessitant une mise à jour.

La transformation joue également un rôle clé lorsque les clusters Kubernetes sont déplacés d'un environnement à un autre, comme nous le verrons dans la section suivante sur la portabilité. Que vous passiez d'une infrastructure locale à un cloud public, cette capacité d'adaptation garantit une restauration fluide et portable.

Dans un environnement aussi rapide et dynamique, il est impossible d'ignorer l'obsolescence des composants, l'évolution des versions d'API ou les mises à jour constantes. De plus, la portabilité de Kubernetes ajoute une couche de complexité supplémentaire. Lors d'une restauration, il ne suffit pas de simplement récupérer les données ; il faut aussi être en mesure de les adapter aux nouvelles versions pour garantir un fonctionnement optimal.

Pour réussir votre transformation, voici les points essentiels à retenir :

- Assurez-vous que votre solution s'adapte aux changements pour suivre le rythme accéléré des évolutions.
- Utilisez la transformation pour faciliter la portabilité.

Faciliter la portabilité

La portabilité est l'un des atouts majeurs de Kubernetes, et une plateforme de sauvegarde peut exploiter cette fonctionnalité pour répondre à de nombreux cas d'usage. La Figure 3-3 illustre quelques-unes des opportunités offertes par cette fonctionnalité.

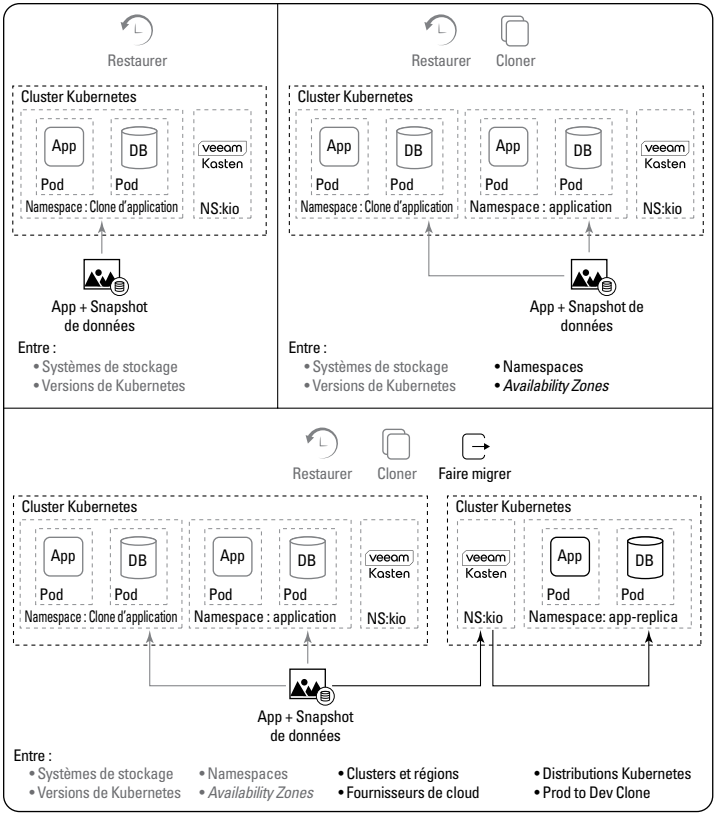


FIGURE 3-3 : La puissance de la portabilité dans la sauvegarde native de Kubernetes.

Voici quelques-uns des cas d'usage potentiels de la portabilité :

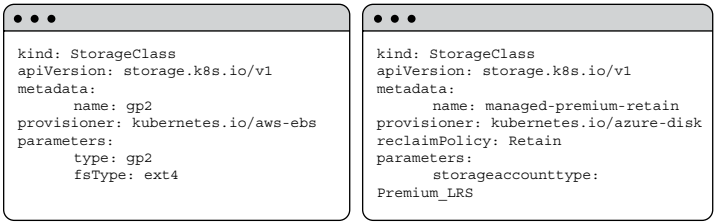
- Entre les namespaces d'un même cluster
- Entre les différents systèmes de stockage
- Entre les clusters, les distributions et les versions de Kubernetes
- Entre les *availability zones* d'une même région
- Entre les régions d'un même cloud
- Entre les environnements cloud ou hybrides
- Entre les environnements de développement et de test

RAPPEL

Comme vous le savez, l'écosystème Kubernetes est varié, avec des déploiements aussi bien en local que dans le cloud. De plus, de nombreuses entreprises adoptent un modèle de cloud hybride pour exécuter leurs applications conteneurisées. Dans ce contexte, une plateforme de

sauvegarde et de gestion des données doit offrir la flexibilité nécessaire pour faire migrer les applications entre des clusters source et destination qui s'exécutent sur des infrastructures différentes.

Reportez-vous à la figure 3-4 pour visualiser les défis liés à la migration d'un workload à partir d'Amazon Elastic Kubernetes Service vers Microsoft Azure Kubernetes Service.



```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
      name: gp2
provisioner: kubernetes.io/aws-ebs
parameters:
      type: gp2
      fsType: ext4
```

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
      name: managed-premium-retain
provisioner: kubernetes.io/azure-disk
reclaimPolicy: Retain
parameters:
      storageaccounttype:
Premium_LRS
```

FIGURE 3-4 : Migration à partir d'EKS vers AKS et comment la terminologie doit être transformée.

Les différences entre les classes de stockage ne sont qu'un premier défi parmi d'autres lorsqu'il s'agit de gérer diverses distributions, même si elles reposent sur la même version de Kubernetes. Il est donc essentiel que votre plateforme de sauvegarde puisse restaurer des applications sur ces différentes distributions et infrastructures. Pour garantir une transition fluide, elle doit être capable de transformer automatiquement la sauvegarde afin de l'adapter à l'environnement de restauration.

RAPPEL

Cette tâche est complexe, mais essentielle. La plateforme de sauvegarde doit être capable d'analyser toutes les dépendances des applications et de les adapter avec précision pour assurer une migration réussie d'un environnement à un autre.

Pour réussir la migration, il est essentiel de mettre en place un plan détaillé. Ce dernier doit s'assurer que toutes les ressources nécessaires, qu'il s'agisse de l'infrastructure, du cluster ou des dépendances applicatives, sont soit accessibles, soit remplacées par des alternatives adaptées. Gardez à l'esprit que la migration ne se limite pas aux conteneurs et aux volumes de stockage ; elle implique également l'adaptation en cours de transfert d'éléments tels que les noms de domaine complets (FQDN), les objets Secret et les adresses DNS.

Pour résumer, la portabilité repose sur plusieurs facteurs :

- Assurez-vous que votre solution de sauvegarde et de gestion des données peut faire migrer des applications entre des clusters s'exécutant sur des infrastructures différentes.
- Ayez un plan de migration pour assurer que toutes les dépendances sont disponibles ou transformées en une ressource équivalente.

DANS CE CHAPITRE

- Se familiariser avec l'environnement cloud
- Unifier la sauvegarde et la récupération d'urgence
- Prospérer dans un environnement multicluster

Chapitre 4
Assurer la mobilité des applications cloud

Après avoir lu les trois derniers chapitres, on pourrait penser que la sauvegarde et la récupération d'urgence (DR) sont particulièrement compliquées dans un environnement conteneurisé. Même si ces processus sont effectivement sophistiqués, ils deviennent tout à fait gérables avec les bons outils. Le plus important, c'est d'avoir une plateforme unique qui permet d'orchestrer et de gérer toutes ces opérations de manière efficace.

Ce chapitre se penche sur un enjeu fondamental, à savoir comment permettre aux applications de rester mobiles tout en assurant leur protection et la DR. Il examine la relation complexe entre les processus de sauvegarde et de restauration ; et il s'intéresse aux difficultés posées par la multiplication rapide des clusters et des ressources.

Se lancer dans le cloud

CONSEIL

Pour ceux qui découvrent Kubernetes, le moyen le plus simple de se familiariser avec son fonctionnement est de l'utiliser dans un cloud public. Les géants du cloud tels que Google, Microsoft et AWS mettent à disposition Kubernetes sous forme de service, ce qui simplifie grandement son installation et son administration.

Les fournisseurs de cloud public prennent en charge la maintenance du service et gèrent des aspects clés comme la rotation des certificats, les mises à jour de version et l'application des correctifs. Ils offrent également des solutions de stockage et s'occupent de l'ensemble de l'infrastructure sous-jacente du cluster.

Approfondissez vos connaissances dans les domaines de la structure et de l'architecture en déployant votre propre cluster Kubernetes. Bien que vous puissiez tout créer à partir de rien, n'oubliez pas qu'en administrant votre propre cluster, vous devrez assumer toutes les tâches généralement prises en charge par les fournisseurs de services des clouds publics.

Des frontières de plus en plus floues

Si vous avez ouvert ce livre, c'est que vous comprenez déjà l'importance cruciale de la sauvegarde et de la DR pour la réussite, voire la survie, de votre entreprise. Cela n'a rien de surprenant.

Vous savez sans doute que la sauvegarde et la restauration n'ont pas toujours été des tâches simples. Par le passé, et surtout avant l'arrivée de Kubernetes en 2015, ces processus reposaient souvent sur des plateformes distinctes pour différentes fonctions, avec plusieurs copies des données dispersées dans divers systèmes.

RAPPEL

Dans l'univers Kubernetes, les utilisateurs recherchent une approche simplifiée où les données ne sont capturées qu'une seule fois. Un même jeu de données doit pouvoir servir à plusieurs usages, comme la sauvegarde et la restauration, la DR, la mobilité des applications ou encore la migration entre différents clusters. Les frontières entre la sauvegarde et la DR sont de plus en plus floues, mais les outils disponibles permettent d'en simplifier la gestion.

Ces concepts mettent en avant la nécessité d'offrir des solutions en libre-service aux développeurs et aux équipes opérationnelles qui ont de multiples responsabilités. Ceci est fondamental étant donné les multiples applications concernées ; et il ne faut surtout pas le voir comme une simple retouche de dernière minute. La portabilité et la mobilité des applications ne doivent pas devenir des processus complexes. C'est pourquoi la prise en charge de la transformation, abordée au chapitre 3, joue un rôle clé dans la réussite de ces opérations.

Lorsqu'il s'agit de migrer d'un cloud à un autre ou de changer de région, la transformation des composants devient essentielle. Mais comment adapter ces éléments pour garantir une mobilité fluide des applications ? C'est précisément l'une des fonctionnalités fondamentales qu'une plateforme de gestion des données doit offrir.

Ajouter des clusters

De nombreuses organisations exécutent un grand nombre de clusters, et il n'est pas rare d'en voir plus de 50 en production. Souvent, ces organisations ont démarré à petite échelle avant d'étendre leur infrastructure, que ce soit en augmentant la taille des clusters existants ou en multipliant les environnements dédiés aux tests, au développement, à la préproduction et à la production.

Les clusters peuvent être configurés pour exécuter des workloads en fonction de différents critères, comme des applications spécifiques, des exigences de sécurité particulières ou le niveau de préparation au déploiement. Ils peuvent également être segmentés par unité commerciale ou même par zone géographique, selon les besoins de l'organisation.

L'approche multicluster offre de nombreux avantages. Les clusters s'adaptent aux workloads spécifiques grâce à un ajustement précis de la taille des nœuds. De plus, le rayon d'action limité réduit l'impact potentiel des incidents. Les environnements multiclusters représentent désormais la norme, même pour les petits déploiements qui comportent généralement trois, cinq ou dix nœuds.

RAPPEL

La complexité de la gestion atteint rapidement des proportions vertigineuses après quelques opérations simples. Le nombre de clusters, d'applications et de ressources ne cesse d'augmenter. Les volumes associés à chaque application s'accumulent aussi. Une gestion indépendante de chaque élément devient donc totalement irréaliste.

CONSEIL

Pour découvrir des études de cas concernant l'implémentation de la sauvegarde et de la restauration dans Kubernetes, consultez le site `https://www.veeam.com/resources/customer-stories.html?product=product%3A68`.

En effet, la gestion des sauvegardes s'accompagne de la gestion des autorisations d'accès basés sur les rôles (RBAC) évoquées dans les chapitres précédents. La gestion mutualisée des ressources et les exigences de sécurité compliquent davantage les choses.

Votre système de sauvegarde et de DR doit offrir un point de gestion centralisé pour simplifier cette complexité et fournir une vue d'ensemble claire de l'ensemble des opérations. Il doit permettre d'analyser les différents contextes et de faciliter la mobilité des applications entre ces environnements.

Il est essentiel de fournir aux opérateurs un point de gestion unique pour centraliser et simplifier l'administration des clusters. En même

temps, les développeurs doivent bénéficier d'un environnement intuitif qui facilite leur travail. L'objectif n'est pas seulement d'ajouter des clusters, mais aussi d'assurer une mobilité fluide des applications dans plusieurs environnements.

Voici quelques exigences clés pour assurer une gestion efficace des données et des opérations dans un environnement multicluster :

- **Sécurité :** elle doit s'appliquer aux données et aux opérations. Cette approche permet d'offrir une visibilité appropriée aux utilisateurs et aux systèmes concernés.
- **Configuration simplifiée :** vous devez pouvoir effectuer des opérations multiclusters sans installations complexes. De plus, la configuration et la gestion des stratégies et des ressources doivent se faire en toute simplicité.
- **Découverte automatisée :** ce processus inclut la découverte des applications Kubernetes, des stratégies de sauvegarde et de tout changement sur l'ensemble des clusters.
- **Vue d'ensemble :** insistez sur l'utilisation d'un panneau de contrôle unique pour avoir un aperçu en temps réel de la situation globale.
- **Stratégies et ressources globales :** les équipes doivent pouvoir définir des stratégies globales comme la fréquence des sauvegardes. Elles ont aussi besoin de spécifier les ressources nécessaires, notamment l'emplacement du stockage cible.
- **Regroupement flexible des clusters :** permettez aux utilisateurs de définir facilement des regroupements logiques flexibles et arbitraires de clusters pour la distribution des stratégies et des ressources globales.
- **Analyse détaillée des clusters :** vous devez pouvoir analyser chaque cluster pour maintenir une vision globale du système.

RAPPEL

La leçon à retenir ici est claire : vous devez avoir une solution native de sauvegarde et de gestion des données pour Kubernetes, qui puisse réellement simplifier la complexité de cet environnement.

DANS CE CHAPITRE

- Apprendre à connaître l'écosystème
- S'adapter au mode de travail des opérateurs
- Activer l'audit et les stratégies de sécurité
- Instaurer la journalisation et accroître la visibilité
- Garder le système de sauvegarde à jour

Chapitre 5
Intégrer l'écosystème cloud

Un système de sauvegarde efficace doit concilier de nombreux éléments pour répondre aux besoins de chacun tout en garantissant la sécurité des données. Il doit s'intégrer harmonieusement aux flux de travail, aux préférences et aux stratégies de l'organisation. De plus, il doit évoluer en permanence pour rester à la pointe de la technologie dans un écosystème en constante mutation.

Ce chapitre traite de l'importance d'intégrer le système aux outils déjà adoptés par les opérateurs, ainsi qu'aux stratégies de sécurité et aux exigences d'audit en vigueur. Il met en avant la nécessité d'une journalisation efficace pour donner aux opérateurs une meilleure visibilité et une observabilité accrue sur le fonctionnement du système. Enfin, il explique pourquoi des mises à jour fréquentes sont essentielles pour suivre le rythme rapide des évolutions technologiques.

Comprendre l'écosystème de la gestion des données

Comme le montre la figure 5-1, l'écosystème de gestion des données Kubernetes comprend quatre sous-composants importants qui jouent chacun un rôle central dans une solution globale :

- **Applications :** pour une solution complète, les applications doivent avoir des intégrations préqualifiées avec les principaux services de données, y compris les systèmes relationnels et NoSQL.
- **Distributions Kubernetes :** les distributions doivent prendre en charge toutes les principales offres Kubernetes gérées dans le cloud et toutes les principales distributions sur site.
- **Prise en charge de l'infrastructure de stockage :** pour une efficacité optimale, vous devez prévoir des interfaces de stockage de conteneurs ainsi que des intégrations de partage direct des ressources.
- **Services de sécurité :** la sécurité de Kubernetes doit couvrir plusieurs aspects essentiels, comme la protection de l'infrastructure du cluster, la gestion du contrôle d'accès, la sécurisation des données sensibles et la supervision proactive des vulnérabilités et des attaques par ransomware.

FIGURE 5-1 : Les quatre principaux composants de l'écosystème de gestion des données Kubernetes.

Une solution robuste doit offrir une cohérence globale, intégrer les bases de données, permettre la découverte automatique des applications et assurer une mobilité fluide entre différents environnements cloud. Une interface utilisateur web puissante et intuitive doit également venir compléter ces fonctionnalités essentielles.

Intégrer Prometheus et Grafana

Les anciens systèmes de sauvegarde ont apporté des fonctionnalités précieuses, comme la journalisation, le reporting, les alertes et l'audit, mais ils n'ont pas toujours simplifié le travail des opérateurs. Aujourd'hui, un bon système de sauvegarde ne doit pas seulement être efficace dans la protection des données, il doit aussi minimiser toute friction pour les équipes. L'enjeu est de savoir comment il peut s'intégrer naturellement dans leurs flux de travail et coexister harmonieusement avec l'infrastructure qu'ils utilisent déjà.

RAPPEL

Dans l'écosystème Kubernetes, Prometheus est la solution de référence pour la supervision et le stockage des métriques issues des applications, des logiciels et du matériel. Elle permet de suivre l'état du système, d'identifier les problèmes et de les résoudre. Grafana, un outil de visualisation open source, est souvent utilisé en complément pour afficher ces données de manière claire et exploitable. Une plateforme de sauvegarde native Kubernetes doit impérativement s'intégrer à ces deux outils. Les opérateurs pourront ainsi avoir les informations nécessaires pour créer des tableaux de bord, configurer des alertes, définir des déclencheurs et obtenir une vue détaillée de l'état du système.

Obtenir des informations essentielles grâce à l'audit

En matière de données, vous devez pouvoir retracer chaque action, son auteur, le lieu et le moment où elle a été effectuée. Un bon système de sauvegarde doit permettre de suivre toutes ces informations avec précision. Il doit aussi offrir une transparence totale qui facilite les contrôles et renforce la sécurité.

RAPPEL

Plutôt que d'imposer un système d'audit indépendant, une solution de sauvegarde Kubernetes doit s'intégrer automatiquement à l'audit natif de Kubernetes. Cette approche est particulièrement efficace dans un environnement mutualisé, car elle permet aux autorisations et aux identités des utilisateurs de s'appliquer directement à l'infrastructure.

Un système de sauvegarde efficace doit être aligné sur les besoins de l'utilisateur pour générer des rapports d'audit précis et exhaustifs. Il ne suffit pas de savoir que « le système de sauvegarde a exécuté X » ; vous devez comprendre que « le système de sauvegarde a exécuté X pour le compte de l'utilisateur Bob ». Cette rigueur est indispensable pour assurer un suivi précis et pertinent des activités.

S'intégrer aux stratégies et à la sécurité du réseau

Il est essentiel de veiller constamment à la sécurité de votre application. Prenons l'exemple d'une base de données SQL dans un conteneur : celle-ci ne doit jamais être directement exposée à Internet, même si c'est juste pour faire une sauvegarde.

CONSEIL

Même si la base de données est interne à l'application, un système de sauvegarde intégré au plan de contrôle Kubernetes peut y accéder sans l'exposer au monde extérieur. Cette approche renforce la sécurité en limitant les surfaces d'attaque. Elle diminue également les risques d'attaques par ransomware et via d'autres logiciels malveillants.

Améliorer la gestion des logs

Chaque utilisateur doit avoir son propre système de gestion des logs. Vous devez pouvoir configurer le système de sauvegarde au niveau des logs afin qu'il s'adapte facilement aux besoins spécifiques de votre entreprise.

Vous devez pouvoir extraire simplement les logs tout en garantissant leur sécurité afin d'éviter la moindre perte d'information. Le système de sauvegarde doit également offrir une interface bien établie avec les solutions tierces de gestion des logs déjà utilisées dans votre environnement.

Améliorer l'observabilité

L'observabilité repose sur une supervision approfondie du système de sauvegarde et sur l'accès aux données pour corréler les événements. Votre organisation doit avoir une visibilité complète sur les applications qui ne se comportent pas correctement, ainsi que sur celles dont l'activité intense a entraîné un allongement des temps de sauvegarde.

L'observabilité du backend des opérations de sauvegarde est tout aussi essentielle, notamment lors du transfert des données vers un stockage en mode objet. Vous devez pouvoir superviser ces processus en temps réel et suivre précisément le flux de données.

Suivre les cycles de versions de Kubernetes

Face au rythme trimestriel des mises à jour de Kubernetes, les changements qui en découlent affectent particulièrement les fonctions essentielles comme la sauvegarde et la restauration des données. Le système de sauvegarde doit donc suivre ce rythme pour que l'organisation soit toujours protégée.

Les distributions commerciales ajoutent une couche de complexité supplémentaire, car elles ont leur propre cycle de publication. Les entreprises mettent régulièrement à jour leurs systèmes, conscientes que leur version actuelle deviendra obsolète en seulement trois à quatre cycles de publication, voire plus tôt.

RAPPEL

La sauvegarde doit donc évoluer en permanence pour suivre ces changements et garantir une protection efficace. Dans un environnement où tout évolue rapidement, des mises à jour semestrielles ne suffisent pas.

Prévoyez plusieurs mises à jour par mois. Veeam, par exemple, publie généralement des mises à jour toutes les deux semaines, voire plus fréquemment selon les besoins. Ce rythme soutenu est essentiel pour suivre l'évolution rapide de l'écosystème.

Chapitre 6
Dix points essentiels à retenir sur la sauvegarde dans Kubernetes

La sauvegarde dans Kubernetes fait appel à de multiples concepts, dont la plupart ont été examinés en détail dans les 40 pages qui précèdent. Pour aller à l'essentiel, ce chapitre met en avant les points clés à retenir. Lisez la suite pour découvrir les principaux points à garder en mémoire.

Comprendre l'architecture

- **Suivre l'évolution de Kubernetes :** dans l'univers de Kubernetes, les évolutions sont si rapides qu'un simple instant d'inattention peut vous faire manquer des avancées majeures. Lors de son lancement en 2015, la plupart des applications déployées sur Kubernetes étaient conçues pour être « sans état ». Or, ce n'est plus le cas aujourd'hui. Le plan de contrôle s'est imposé comme un support efficace aussi bien pour les applications avec état que pour celles sans état. Cette évolution offre d'immenses possibilités, mais elle impose également une vigilance accrue quant à la gestion des données.

- **Renforcer les capacités des développeurs :** Kubernetes est conçu pour garantir le bon fonctionnement des applications, un objectif que les développeurs partagent pleinement. Cependant, bien qu'ils soient aux commandes, ils évoluent en territoire inconnu et font face à des obstacles inédits. En plus de créer des applications, les développeurs doivent désormais définir les besoins en infrastructure, une tâche qui ne faisait généralement pas partie de leurs attributions auparavant. Cette transformation apporte de nouveaux défis, d'où la nécessité absolue de mettre en place un système fiable de sauvegarde et de restauration pour protéger cet environnement qui ne cesse d'évoluer.

Mettre l'accent sur les opérations

- **Séparer le déploiement :** l'infrastructure Kubernetes se distingue radicalement des approches traditionnelles. Elle permet de concevoir des applications conteneurisées, dont les composants sont répartis sur plusieurs nœuds afin d'optimiser les performances et de renforcer la tolérance aux pannes. Contrairement aux modèles classiques, ces applications conteneurisées ne sont pas directement mappées sur des serveurs ou des machines virtuelles. Une solution de sauvegarde efficace doit donc être conçue pour comprendre ces modèles architecturaux cloud et s'adapter à leur fonctionnement afin d'assurer une protection optimale des données.
- **Augmenter et réduire la charge :** dans un cluster Kubernetes, les applications conteneurisées peuvent automatiquement augmenter ou réduire leur charge en fonction des besoins. La solution de sauvegarde doit suivre cette même logique. En adoptant la même architecture cloud, elle peut s'adapter aux variations des applications et des clusters. Cette approche optimise non seulement les performances, mais représente aussi la solution la plus rentable.

Optimiser les sauvegardes

- **Choisir un système natif :** la sauvegarde et la récupération d'urgence (DR) sont essentielles, mais il est illusoire de vouloir connecter des applications Kubernetes avec état flambant neuves à un outil de sauvegarde traditionnelle conçu pour une infrastructure virtualisée. Kubernetes repose sur une approche fondamentalement différente, incompatible avec les solutions existantes du monde de la virtualisation. La seule option fiable consiste à utiliser un système de sauvegarde natif Kubernetes, conçu pour s'intégrer parfaitement à cet environnement et répondre à ses exigences spécifiques.

- **Ne pas confondre disponibilité et sauvegarde :** les temps d'arrêt peuvent entraîner des coûts considérables, ce qui explique en partie pourquoi tant d'organisations adoptent les applications conteneurisées sur Kubernetes. Cependant, ces fonctionnalités de réplication, aussi puissantes soient-elles, ne remplacent pas une sauvegarde efficace. Sans une stratégie de protection des données clairement définie, ces dernières restent vulnérables aux incidents et menaces.
- **Protéger la capacité de restauration :** la sauvegarde et la DR natives de Kubernetes prennent en compte les relations et les dépendances entre les nombreux composants d'un conteneur. Cet orchestrateur réalloue constamment des ressources afin d'assurer la continuité du service et de garantir l'évolutivité. Par exemple, une application de vente au détail doit pouvoir gérer des pics de charge importants lors du Black Friday. Pour être efficace, la sauvegarde doit être capable de reconstruire l'ensemble de l'environnement correctement. Un plan de reprise détaillé est indispensable. Il doit inclure une vue complète des données à restaurer et préciser les sources de sauvegarde ainsi que les destinations de stockage. Enfin, vous devrez identifier les éléments nécessitant une transformation au cours du processus de restauration.

Garantir une sécurité constante

- **Profiter de mesures de sécurité natives :** Kubernetes intègre des mesures de sécurité avancées qui garantissent que seuls les utilisateurs et composants autorisés peuvent accéder au système, tandis que les intrusions sont bloquées. Cette protection ne concerne pas uniquement les personnes, mais aussi les différents composants des applications. Si la solution de sauvegarde et de récupération fonctionne en dehors des clusters Kubernetes, elle perd en efficacité. Elle ne peut pas assurer correctement la découverte des applications ni accéder aux ressources nécessaires.
- **Respecter vos voisins :** avez-vous déjà vécu dans un immeuble où vous partagez des murs avec des voisins ? Dans une telle situation, vous apprenez à trouver un équilibre entre la confiance mutuelle et la protection de votre espace personnel. Les clusters Kubernetes fonctionnent de la même manière, avec des applications qui cohabitent grâce à la gestion mutualisée. Les développeurs doivent pouvoir accéder à leurs propres applications sans interférer avec celles des autres. La solution de sauvegarde doit respecter ces mêmes principes. Elle doit appliquer des contrôles d'accès stricts afin de garantir que chaque élément reste à sa place et que la sécurité de l'ensemble du système soit préservée.

Modifier la vitesse d'amélioration

- **Rester à jour :** certains automobilistes gardent leur voiture jusqu'à atteindre 150 000 voire 200 000 km au compteur, tandis que d'autres préfèrent en changer tous les deux ans. Dans l'univers de Kubernetes, le code n'a jamais le temps d'accumuler des kilomètres. Les équipes se tiennent toujours au fait des dernières innovations et avancées technologiques, qu'elles intègrent volontiers. Votre solution de sauvegarde doit suivre cette même dynamique. De nouvelles versions seront fréquemment déployées, car c'est le seul moyen d'assurer une protection efficace dans un environnement en perpétuelle évolution, où les mises à jour s'enchaînent à un rythme soutenu.

Sauvegarde et restauration native Kubernetes, récupération d'urgence (DR), mobilité des applications et protection contre les ransomwares

Kubernetes connaît une expansion fulgurante dans le monde des infrastructures logicielles et s'est imposée comme la plateforme de référence pour les applications de niveau entreprise. Si Kubernetes excelle dans la haute disponibilité et l'évolutivité des services d'applications, la protection des données reste un défi majeur. C'est pourquoi la protection des données est devenue cruciale pour les workloads Kubernetes. Cet ouvrage vous fournit toutes les ressources et solutions nécessaires pour garantir une protection optimale de vos applications Kubernetes.

À l'intérieur…

- Comprendre Kubernetes et les applications cloud natives
- Créer une protection native des données pour Kubernetes
- Meilleures pratiques pour la sauvegarde et restauration Kubernetes
- Assurer la mobilité des applications cloud natives
- L'écosystème cloud natif

veeam

Steve Kaelble est l'auteur de nombreux ouvrages dans la série *pour les Nuls*, et a publié des articles dans divers magazines, journaux et rapports annuels d'entreprise. Quand il n'est pas plongé dans ses recherches *pour les Nuls* ou en train de rédiger ses articles, il évolue dans le domaine de la communication santé.

Rendez-vous sur Dummies.com®
pour voir des vidéos, des tutoriels en photos, des articles pratiques, ou pour faire des achats !

ISBN: 978-1-394-35368-2

pour les nuls®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.