



# Tendances technologiques

## Kubernetes

Architecture d'entreprise, Direction générale du dirigeant principal  
de la technologie

Version 0.1

Date : 2019-05-09



Shared Services  
Canada

Services partagés  
Canada

Canada

## Table des matières

<b>Dossier d'affaires</b> .....	<b>3</b>
<b>Mémoire technique</b> .....	<b>4</b>
<b>Utilisation par l'industrie</b> .....	<b>4</b>
<b>Utilisation par le gouvernement du Canada</b> .....	<b>5</b>
<b>Répercussions pour Services partagés Canada (SPC)</b> .....	<b>6</b>
Proposition de valeur .....	6
Défis .....	8
Facteurs à prendre en considération.....	10
<b>Références</b> .....	<b>13</b>

## Dossier d'affaires

Les conteneurs d'applications sont de plus en plus utilisés dans l'infrastructure pour les applications infonuagiques natives et les applications de microservices. Plus précisément, Docker est le moteur le plus couramment utilisé pour créer des conteneurs. De nombreux développeurs d'applications ont noté la valeur des conteneurs dans la gestion des dépendances, car ceux-ci regroupent l'application et ses dépendances, les bibliothèques et les autres éléments binaires dans un conteneur pour éliminer les différences dans la distribution du système d'exploitation et l'infrastructure sous-jacente. Cela résout de nombreux problèmes causés par les différences dans les environnements d'exécution lorsqu'un logiciel passe d'un environnement à un autre.

Les conteneurs améliorent la portabilité et l'évolutivité des applications, ce qui permet à ces dernières d'être publiées et mises à jour de manière simple et rapide, sans temps d'arrêt. Cependant, il existe encore une demande liée à la gestion des conteneurs lorsque les services fournis doivent être déployés dans une grappe de serveurs hôtes pour obtenir une haute disponibilité et une reprise après sinistre. C'est là que le déploiement des conteneurs en grappes et les outils de gestion comme Kubernetes offrent une valeur ajoutée. Les développeurs peuvent maintenant commencer à déployer et à orchestrer les services comme une collection de conteneurs dans une grappe de serveurs. Les besoins en ressources pour les conteneurs peuvent être explicitement déclarés, ce qui permet aux développeurs de regrouper le code de l'application avec des configurations d'environnement. De plus, en augmentant la densité des conteneurs, les ressources peuvent être utilisées plus efficacement, ce qui améliore l'utilisation du matériel [1]. Les conteneurs permettent d'isoler les applications, de sorte qu'une équipe de développement peut être désignée responsable de conteneurs spécifiques.

Kubernetes, également connu sous le nom de K8s, est une plateforme de source ouverte portable et extensible qui sert à gérer les charges de travail et les services conteneurisés, ce qui facilite la configuration déclarative et l'automatisation. Cette plateforme fournit un environnement de gestion axé sur le conteneur. Elle orchestre les ressources machines, la mise en réseau et l'infrastructure de stockage pour les charges de travail des utilisateurs. De plus, Kubernetes offre des fonctions d'autoguérison, de déploiement automatisé et de retours arrière, qui améliorent considérablement la haute disponibilité et la souplesse des opérations. L'un des plus grands avantages de Kubernetes est sa souplesse. De nombreuses intégrations PaaS dictent des cadres spécifiques, sont adaptées à des charges de travail spécifiques ou imposent des restrictions quant aux durées d'exécution de langages qui peuvent être utilisées[1]. Ces problèmes sont tous éliminés avec Kubernetes. Par conséquent, si l'application d'une organisation peut être exécutée dans un conteneur, Kubernetes est une option viable pour l'orchestration du conteneur.

## Mémoire technique

Le déploiement ou la grappe Kubernetes peut être décomposé en plusieurs composantes. Le « maître » Kubernetes est la machine chargée de gérer d'autres machines de travail appelées « nœuds », lesquelles sont chargées d'exécuter les tâches qui lui sont transmises par l'utilisateur ou le maître. Le maître et les nœuds peuvent être des machines physiques ou virtuelles. Dans chaque grappe Kubernetes, il y a une machine maître et plusieurs nœuds. L'objectif principal de Kubernetes est de parvenir à la « gestion de l'état souhaité ». Le maître reçoit une configuration spécifique par le biais de son interface (API) RESTful et l'expose à l'utilisateur; le maître est alors responsable de l'exécution de cette configuration sur son ensemble de « nœuds ». En fait, les nœuds peuvent être considérés comme une multitude de conteneurs. Ils communiquent avec le maître par l'intermédiaire de l'agent dans chaque nœud – il s'agit du processus « Kubelet ». Par ailleurs, pour établir une configuration spécifique, le maître reçoit un fichier de déploiement comportant l'extension « .yaml », lequel contient divers renseignements sur la configuration. Parmi ces renseignements, on retrouve les « Pods » et les « répliques ». Dans Kubernetes, il existe un concept de « Pod », qui peut être décrit comme une collection logique de conteneurs qui sont gérés comme une seule application. Les ressources peuvent être partagées au sein d'un Pod et comprennent le stockage partagé (volumes), une grappe d'adresses IP uniques, et des renseignements sur la façon d'exécuter chaque conteneur. Un Pod peut être considéré comme l'unité de base du modèle objet Kubernetes; il représente le déploiement d'une seule instance d'une application dans Kubernetes [8]. Il peut contenir un ou plusieurs conteneurs d'applications. De plus, il existe deux modèles de déploiement des Pods au sein d'une grappe. Le terme « un Pod par conteneur » signifie qu'un seul Pod sera associé à un seul conteneur. Il peut également y avoir plusieurs conteneurs qui fonctionnent dans un même Pod, où ces conteneurs peuvent devoir communiquer entre eux, car ils partagent des ressources. Dans l'un ou l'autre modèle, le Pod peut être considéré comme une enveloppe autour des conteneurs d'applications. Kubernetes gère les instances Pod plutôt que de gérer directement les conteneurs. Les Pods sont exécutés sur les nœuds pour effectuer des tâches. Quant aux répliques, il s'agit simplement d'instances des Pods. Dans le fichier de déploiement « .yaml », les spécifications indiquent à la machine « maître » combien d'instances ou de répliques de chaque Pod doivent être exécutées – une tâche gérée par un contrôleur de réplication [8]. Lorsqu'un nœud meurt ou qu'un Pod en cours d'exécution subit une terminaison inattendue, le contrôleur de réplication en prendra note et s'en occupera en créant le nombre approprié de Pods [8].

## Utilisation par l'industrie

Kubernetes est un système de source ouverte, et de nombreuses entreprises ont commencé à l'adopter dans leur architecture existante et à l'adapter à leurs besoins spécifiques. Il a été élaboré à l'origine par Google et est devenu un projet de source ouverte en 2014. La Cloud Native Computing Foundation est un projet de la Fondation

Linux qui forme une communauté pour diverses entreprises qui cherchent à développer Kubernetes et à élaborer d'autres projets d'orchestration de conteneurs. Plusieurs plateformes et fournisseurs de services infonuagiques de grande envergure, dont Google Cloud Compute, HP Helion Cloud, RedHat Openshift, VMware Cloud et Windows Azure prennent en charge l'utilisation de Kubernetes [7]. Une enquête, réalisée par iDatalabs en 2017, a révélé que 2 867 entreprises utilisent actuellement Kubernetes. Ces entreprises sont généralement situées aux États-Unis et travaillent aussi en grande majorité dans l'industrie des logiciels informatiques. De plus, les entreprises figurant sur la liste emploient de 50 à 200 employés, et enregistrent entre 1 et 100 millions de dollars de revenus par année. Parmi les principales entreprises figurant sur cette liste, mentionnons GoDaddy Inc., Pivotal Software Inc., Globant SA et Splunk Inc. Enfin, Kubernetes détient environ 8,6 % de part de marché dans la catégorie des logiciels de gestion de la virtualisation [9].

## Utilisation par le gouvernement du Canada

On retrouve peu d'initiatives et de programmes documentés du gouvernement du Canada (GC) qui font la promotion de l'utilisation actuelle et future de la technologie Kubernetes. En tant qu'élément stratégique en TI du GC, Kubernetes est absent du Plan stratégique des opérations numériques du GC : 2018-2022, et du Plan stratégique du GC pour la gestion de l'information et la technologie de l'information de 2017 à 2021. La situation peut s'expliquer par le fait que le gouvernement doit actuellement composer avec la mise en œuvre de services infonuagiques et que la majorité des ressources sont ainsi occupées, ainsi que par les questions de sécurité entourant la protection des renseignements des Canadiens.

Cependant, l'arrivée des conteneurs sur le marché a démontré que les grandes organisations qui participent au développement d'applications infonuagiques natives ainsi qu'à la mise en réseau peuvent grandement bénéficier de l'utilisation des conteneurs [7]. Bien que les applications d'infrastructure fournissant des services infonuagiques puissent être basées uniquement sur des machines virtuelles (MV), les coûts de maintenance associés à l'utilisation de différents systèmes d'exploitation sur des MV individuelles dépassent les avantages qu'elles présentent [6]. Les conteneurs et la conteneurisation constituent une architecture complémentaire ou de remplacement aux MV. Au fur et à mesure que le CG ira de l'avant avec les services infonuagiques et le développement d'applications infonuagiques natives, l'utilisation de conteneurs et leur orchestration avec Kubernetes pourraient devenir une partie intégrante de l'architecture de TI du CG.

# Répercussions pour Services partagés Canada (SPC)

## *Proposition de valeur*

La principale valeur opérationnelle de Kubernetes est la portabilité de la technologie et sa mobilité, indépendamment de l'environnement. Sa capacité à gérer et à orchestrer les conteneurs d'applications d'une organisation constitue un avantage notable. En outre, la valeur commerciale secondaire de Kubernetes est qu'elle permet à l'entreprise d'agir rapidement, ce qui signifie que chaque équipe de produits peut envoyer, de façon sécuritaire, des mises à jour plusieurs fois par jour; effectuer un déploiement instantané; observer les résultats en temps réel; et utiliser cette rétroaction pour revenir à une version antérieure ou passer à une version subséquente des conteneurs, afin d'améliorer rapidement l'expérience client.

À l'ère des services Web modernes, les utilisateurs s'attendent à ce que leurs applications soient disponibles 24 heures sur 24, 7 jours sur 7, et les développeurs s'attendent à pouvoir déployer de nouvelles versions de ces applications plusieurs fois par jour, avec un temps d'arrêt minimal. Les conteneurs sont devenus l'un des principaux moyens de gérer les applications au sein de l'infrastructure de TI de l'entreprise et aussi l'un des domaines les plus difficiles à gérer efficacement.

Kubernetes, en tant que système de source ouverte, est une technologie qui permet d'administrer et de gérer un grand nombre d'applications conteneurisées qui sont réparties sur des grappes de serveurs, tout en fournissant des mécanismes de base pour le déploiement, la maintenance et la mise à l'échelle des applications.<sup>ii</sup> Par ailleurs, un conteneur d'applications est une unité standard de logiciel qui regroupe le code et toutes ses dépendances afin que l'application s'exécute rapidement et de façon fiable d'un environnement informatique à un autre<sup>iii</sup>. Kubernetes automatise plus efficacement la distribution et la planification des conteneurs d'applications dans une grappe<sup>iv</sup>.

Les conteneurs offrent un mécanisme de regroupement logique qui permet de faire abstraction des applications par rapport à l'environnement dans lequel elles s'exécutent réellement. Ainsi, les applications basées sur des conteneurs peuvent être déployées facilement et uniformément dans n'importe quel environnement, qu'il s'agisse d'un centre de données privé, du nuage public ou encore de l'ordinateur personnel d'un développeur<sup>v</sup>. Un autre avantage de la conteneurisation est que le système d'exploitation est moins sollicité.

Kubernetes étant un système de source ouverte, il offre à l'entreprise la liberté de tirer profit de l'infrastructure infonuagique publique, hybride et sur place, ainsi que la capacité de déplacer facilement les charges de travail<sup>vi</sup>. Les applications conteneurisées sont plus souples et disponibles que dans les modèles de déploiement antérieurs, où les applications étaient installées directement sur des machines spécifiques sous forme de progiciels profondément intégrés dans l'hôte. De plus, Kubernetes regroupe les conteneurs qui constituent une application en unités logiques pour en faciliter la gestion et la découverte. Les abstractions de Kubernetes vous permettent de déployer des applications conteneurisées dans une grappe sans les lier spécifiquement à des machines individuelles (c'est-à-dire des machines virtuelles). Les applications peuvent être situées sur les mêmes machines, sans avoir une incidence les unes sur les autres. Cela signifie que les tâches de plusieurs utilisateurs peuvent être regroupées sur un nombre réduit de machines, ce qui permet une plus grande efficacité et réduit le coût du matériel puisqu'on utilise moins de machines.

Kubernetes contient des outils pour l'orchestration, la gestion des secrets, la découverte des services, la mise à l'échelle et l'équilibrage de charge, et comprend l'emballage (« bin packing ») automatique pour optimiser le placement des conteneurs, et il applique les configurations par l'entremise des fonctionnalités de gestion de configuration<sup>vii</sup>. Il protège la charge de travail des conteneurs en déployant ou en annulant les changements, et offre des contrôles de disponibilité et de qualité pour les conteneurs en remplaçant ou en redémarrant les conteneurs ayant échoué. Au fur et à mesure que les exigences évoluent, un utilisateur peut déplacer les charges de travail des conteneurs dans Kubernetes d'un fournisseur de services infonuagiques à un autre, ou d'une infrastructure d'hébergement à une autre, sans modifier le code<sup>viii</sup>. Cela offre une excellente valeur aux développeurs, car leur travail est protégé, et il y a une piste de vérification des changements.

Les concepts de base de Kubernetes qui permettent une grande vitesse sont l'immuabilité, la configuration déclarative et les systèmes d'autoguérison<sup>ix</sup>.

Les conteneurs et Kubernetes encouragent les développeurs à construire des systèmes distribués qui adhèrent aux principes d'une infrastructure immuable. Dans une infrastructure immuable, un artefact créé ne sera pas modifié lorsque l'utilisateur apporte des modifications. Pour mettre à jour les applications dans une infrastructure immuable, une nouvelle image de conteneur est générée avec une nouvelle étiquette et est ensuite déployée, mettant ainsi fin à l'ancien conteneur avec l'ancienne version de l'image. De cette façon, l'entreprise a toujours un enregistrement d'artefacts indiquant ce qui a été fait et s'il y a eu une erreur dans la nouvelle image. Si une erreur est détectée, le conteneur est ramené à l'image précédente<sup>x</sup>. Tout ce qui entre dans un conteneur comporte un fichier texte, lequel peut être traité comme un code de source d'application qui assure l'immuabilité.

La configuration déclarative permet à l'utilisateur de décrire exactement l'état dans lequel le système devrait se trouver. Les outils traditionnels de développement, tels que le contrôle des sources, les tests unitaires, etc. peuvent être utilisés avec des configurations déclaratives d'une manière qui est impossible avec des configurations impératives. Les systèmes impératifs décrivent comment se rendre d'un point A à un point B, mais comprennent rarement les instructions pour faire l'opération en sens inverse. La configuration déclarative de Kubernetes rend les retours en arrière assez faciles, ce qui est impossible avec des configurations impératives.<sup>xi</sup>

Enfin, Kubernetes dispose d'une fonctionnalité d'autoguérison. Lorsque Kubernetes reçoit une configuration d'état désiré, il ne se contente pas de prendre des mesures pour que l'état actuel corresponde à l'état souhaité à un moment donné; il prend continuellement des mesures pour s'assurer que l'état actuel corresponde en permanence à l'état désiré.<sup>xii</sup>

## Défis

Le plus grand défi lié à Kubernetes est sa complexité. Cependant, la sécurité, le stockage et la mise en réseau, la maturité et les priorités concurrentes en matière de transformation d'entreprise sont également des défis auxquels la technologie de Kubernetes est confrontée.

### Complexité de Kubernetes et expérience des analystes

Il existe un défi lié au manque d'expérience de l'organisation et des analystes en ce qui concerne la gestion des conteneurs et l'utilisation de Kubernetes. La gestion, la mise à jour et la modification d'une grappe Kubernetes peuvent être complexes sur le plan opérationnel, et encore plus si les analystes manquent d'expérience. Le système lui-même fournit une solide base d'infrastructure pour une plateforme-service (PaaS), qui est un cadre de travail pouvant réduire la complexité pour les développeurs. Cependant, la mise à l'essai dans un environnement Kubernetes demeure une tâche complexe. Bien que ses cas d'utilisation en matière de mise à l'essai soient bien notés, la mise à l'essai de plusieurs parties mobiles d'une infrastructure pour déterminer la bonne fonctionnalité d'application est une tâche difficile [1]. Cela signifie que les équipes des opérations qui développent et gèrent l'infrastructure de Kubernetes auront besoin de beaucoup de nouvelles connaissances. Plus l'entreprise est grande, plus l'utilisateur de Kubernetes est susceptible d'être confronté à des défis liés aux conteneurs<sup>xiii</sup>.

### Sécurité

Dans un environnement distribué et hautement évolutif, les modèles de sécurité traditionnels et typiques ne couvrent pas toutes les menaces. La sécurité doit être harmonisée pour les conteneurs et dans le contexte de Kubernetes. De plus, il est



essentiel que les équipes des opérations comprennent la sécurité de Kubernetes en termes de conteneurs, de déploiement et de sécurité des réseaux. Les périmètres de sécurité sont de nature poreuse; les conteneurs doivent être sécurisés au niveau du nœud, mais également au niveau de l'image et du registre. Ainsi, les pratiques de sécurité dans le contexte de divers modèles de déploiement constitueront un défi permanent<sup>xiv</sup>.

### **Stockage et mise en réseau**

Les technologies de stockage et de mise en réseau sont des piliers de l'infrastructure des centres de données, mais elles ont été conçues à l'origine pour les environnements client/serveur et virtualisés. Les technologies des conteneurs amènent les entreprises à repenser le fonctionnement et l'exploitation des technologies de stockage et de mise en réseau<sup>xv</sup>. Par ailleurs, les architectures sont de plus en plus axées sur les applications, et le stockage n'est pas nécessairement disponible sur la même machine que l'application ou ses services. Les grandes entreprises ont tendance à exécuter davantage de conteneurs; pour ce faire dans des environnements de production évolutifs, il faut adopter de nouvelles approches en matière d'infrastructure<sup>xvi</sup>.

Certains systèmes existants peuvent exécuter des conteneurs et, parfois seulement, les machines virtuelles peuvent être remplacées par des conteneurs. Il pourrait y avoir d'importantes conséquences techniques pour les systèmes existants si la conteneurisation et Kubernetes sont mis en œuvre dans un système existant qui n'est pas conçu pour gérer ce changement. De plus, il peut s'avérer nécessaire d'effectuer un réusinage des systèmes existants et de les adapter à la conteneurisation. Certains éléments d'un système peuvent être divisés et conteneurisés. En général, tout ce qui est exposé à Internet devrait être exécuté dans des conteneurs.

### **Maturité**

On vérifie encore la maturité de Kubernetes. Pour l'instant, Kubernetes est le chef de file du marché ainsi que le moyen normalisé d'orchestrer les conteneurs et de déployer des applications distribuées. Google est la principale organisation commerciale derrière Kubernetes, mais elle ne prend pas en charge Kubernetes en tant que produit logiciel. Elle offre un service commercial Kubernetes géré, connu sous le nom de Google Kubernetes Engine (GKE), mais pas sous la forme d'un logiciel. Cela peut être considéré à la fois comme une force et une faiblesse. En l'absence de commercialisation, l'utilisateur bénéficie d'une plus grande flexibilité quant à la manière dont Kubernetes peut être mis en œuvre dans son infrastructure. Cependant, sans un ensemble concret de normes pour les services que Kubernetes peut offrir, il existe un risque que le soutien continu de Google ne soit pas garanti. Ses dons du code de Kubernetes et de la propriété intellectuelle à la Cloud Native Computing Foundation minimisent ce risque puisqu'il y a une organisation qui appliquera les normes appropriées et vérifiera les services que Kubernetes pourra offrir à l'avenir [1]. Il est également important de noter que les défis organisationnels auxquels les utilisateurs

de Kubernetes sont confrontés dépendent maintenant davantage de la taille de l'organisation qui l'utilise.

Kubernetes fait face à d'autres technologies d'ordonnanceur et d'orchestrateur, telles que Docker Swarm et Mesosphere DC/OS. Bien que Kubernetes soit parfois utilisé pour gérer les conteneurs Docker, il est également en concurrence avec les fonctionnalités propres de Docker Swarm en termes de mise en grappe<sup>xvii</sup>. Cependant, Kubernetes peut être exécuté dans le cadre d'un service d'infonuagique public ou sur place; est hautement modulaire et de source ouverte; et possède une communauté dynamique. Enfin, des entreprises de toutes tailles investissent dans ce système, et de nombreux fournisseurs de services infonuagiques offrent Kubernetes comme service<sup>xviii</sup>.

### **Priorités concurrentes en matière de transformation d'entreprise**

L'élaboration et la mise en œuvre de l'initiative de Kubernetes sont confrontées à un dernier défi, c'est-à-dire la place que l'initiative occupe dans la liste des priorités d'une organisation en ce qui concerne la transformation de la TI. Souvent, il y a beaucoup d'initiatives qui peuvent avoir préséance sur les projets de Kubernetes.

## ***Facteurs à prendre en considération***

### **Ressourcement stratégique et planification du réseau**

Une approche stratégique en matière d'investissements liés à Kubernetes devra être élaborée afin de s'assurer de bien tirer profit des possibilités. Le GC investit une part importante de son budget annuel dans la TI et l'infrastructure associée. Sans une orientation stratégique relative à Kubernetes, les approches fragmentées se rapportant aux investissements en matière de TI jumelées à l'évolution rapide de la technologie et aux pratiques opérationnelles disparates pourraient nuire à la prestation efficace et efficiente des programmes et des services du GC<sup>xix</sup>. L'établissement d'une vision et d'un mandat clairs en ce qui concerne la façon dont Kubernetes transformera les services, ainsi que la détermination de ce à quoi devrait ressembler l'état final de l'initiative Kubernetes, constituent des éléments importants à prendre en considération.

SPC devrait envisager de définir une stratégie réseau pour l'adoption de Kubernetes. Plusieurs facteurs doivent être pris en compte, y compris les ressources, le financement et l'expertise qui seront nécessaires pour l'élaboration et la mise à l'essai des technologies Kubernetes. De plus, il est impératif d'effectuer le calcul des besoins en ressources, y compris l'unité centrale, la mémoire, le stockage, etc., au début des projets Kubernetes. Les éléments à prendre en compte comprennent la question de savoir s'il faut élaborer une solution Kubernetes interne, ou si une solution peut être acquise. Parmi les autres considérations stratégiques, mentionnons l'analyse de différentes approches d'orchestration pour différents cas d'utilisation d'applications.

## **Complexité et le manque de compétences**

Kubernetes est une bonne technologie et représente la norme de facto pour l'orchestration des conteneurs, et les conteneurs sont l'avenir de la livraison de logiciels modernes. Cependant, le produit est notoirement complexe à gérer pour les charges de travail d'entreprise, où les accords de niveau de service sont essentiels. Les difficultés opérationnelles liées à la gestion de l'environnement Kubernetes de production sont accentuées par la pénurie de talents et le manque de compétences à l'échelle de l'industrie. Aujourd'hui, la plupart des organisations ont de la difficulté à recruter des experts Kubernetes, et même ces « experts » n'ont pas acquis une expérience avancée de Kubernetes pour assurer son bon fonctionnement à grande échelle. SPC devra faire preuve de prudence dans la mise en œuvre de Kubernetes et avoir un personnel expérimenté et à l'aise dans son utilisation.

## **La personnalisation et l'intégration sont encore nécessaires**

La technologie et l'écosystème de Kubernetes évoluent rapidement. En raison de son état relativement nouveau, il est difficile de trouver des solutions regroupées comportant un soutien complet et prêtes à l'emploi pour des scénarios d'entreprise complexes et à grande échelle. En tant que grande organisation d'entreprise sophistiquée, SPC devra consacrer d'importantes ressources à la personnalisation et à la formation. Les professionnels de l'architecture d'entreprise devront se concentrer sur l'ensemble de l'architecture des applications infonuagiques natives, et suivre de près l'évolution technologique et l'industrie.

La mise en œuvre prend généralement plus de temps que prévu, mais le consensus qui s'est dégagé lors de l'enquête sur l'expérience utilisateur de Kubernetes réalisée par New Stack est que Kubernetes réduit le temps de déploiement du code et augmente la fréquence de ces déploiements<sup>xx</sup>. Cependant, à court terme, la phase de mise en œuvre utilise davantage de ressources humaines.

## **Réussite du projet pilote et à l'échelle**

SPC pourrait évaluer le catalogue de services actuel afin de déterminer les situations pour lesquelles Kubernetes peut d'abord être mis à profit afin d'améliorer l'efficacité, de diminuer les coûts et de réduire le fardeau administratif en ce qui concerne les services existants et de déterminer de quelle façon offrir les nouveaux services Kubernetes de manière uniforme. Tous les nouveaux appareils ou plateformes acquis doivent avoir une valeur élevée sur le marché et doivent pouvoir être facilement intégrés au réseau du GC. Par ailleurs, SPC devrait éviter d'utiliser une solution Kubernetes élaborée à l'interne pour des applications de production essentielles à la mission. Le taux d'échec des déploiements internes est élevé et, par conséquent, ces derniers devraient être évités. Enfin, SPC devrait établir et mettre à l'essai une grappe Kubernetes. La réalisation d'un projet pilote est préférable pour n'importe quelle nouvelle technologie infonuagique. L'accent doit d'abord être mis sur un ensemble

restreint d'objectifs et un scénario d'application unique pour former une grappe d'essai.

### **Mettre en œuvre de pratiques et d'outils robustes de surveillance, de journalisation et de vérification**

La surveillance fournit une visibilité et des mesures détaillées de l'infrastructure Kubernetes. Cela comprend des mesures granulaires sur l'utilisation et le rendement pour tous les fournisseurs de services infonuagiques ou les centres de données privés, les régions, les serveurs, les réseaux, le stockage, ainsi que les machines virtuelles ou les conteneurs individuels. L'objectif est d'améliorer l'efficacité et l'utilisation des centres de données, tant sur place que sur les ressources infonuagiques publiques. De plus, la journalisation est une fonction complémentaire, et le développement de la capacité requise pour assurer une surveillance efficace constitue également un objectif. La journalisation garantit que les journaux de chaque couche de l'architecture seront tous saisis aux fins d'analyse, de dépannage et de diagnostic. Par ailleurs, la visualisation et la gestion des journaux centralisés et distribués sont des capacités clés<sup>xxi</sup>. Enfin, peu importe les mesures de contrôle mises en place, la vérification régulière portera sur des sujets qui ne font pas l'objet d'une surveillance normale. Traditionnellement, la vérification est un processus manuel, mais les outils automatisés dans l'environnement Kubernetes s'améliorent rapidement.

### **Sécurité**

La sécurité est un élément essentiel des applications infonuagiques natives, et Kubernetes ne fait pas exception. Elle est une préoccupation constante tout au long du cycle de vie des conteneurs et est nécessaire tout au long de la conception, du développement, du processus de développement opérationnel (DevOps), et des choix d'infrastructure pour les applications basées sur des conteneurs. Une gamme de choix technologiques est disponible pour couvrir divers domaines, tels que la sécurité à l'échelle des applications ainsi que la sécurité du conteneur et de l'infrastructure elle-même. Différents outils qui assurent la certification et la sécurité du contenu du conteneur lui-même (comme le registre d'image, la signature d'image, l'intégration), les analyses Common Vulnerability Exposures/Enumeration (CVE), et plus sont aussi requis<sup>xxii</sup>. Enfin, SPC devra s'assurer que des mesures de sécurité appropriées sont utilisées avec toute nouvelle initiative de Kubernetes, y compris le contenu des conteneurs faisant l'objet d'une orchestration.

## **Références**

<sup>i</sup> Jayanandana, Nilesh. (May 2<sup>nd</sup>, 2018). *Benefits of Kubernetes*. Medium Newspaper. Retrieved 16-May-2019 from: <https://medium.com/platformer-blog/benefits-of-kubernetes-e6d5de39bc48>

<sup>ii</sup> GitHub. (2019). *Production-Grade Container Scheduling and Management*. GitHub. 2019. Retrieved 16-May-2019 from: <https://github.com/kubernetes/kubernetes>

<sup>iii</sup> Docker. (2019). *What is a Container? A Standardized Unit of Software*. Docker Inc. 2019. Retrieved 16-May-2019 from: <https://www.docker.com/resources/what-container>

<sup>iv</sup> Kubernetes. (2019). *Using Minikube to Create a Cluster*. Kubernetes. 2019. ICP license: 京ICP备17074266号-3. Retrieved 16-May-2019 from: <https://kubernetes.io/docs/tutorials/kubernetes-basics/create-cluster/cluster-intro/>

<sup>v</sup> <https://cloud.google.com/containers/>

<sup>vi</sup> Kubernetes. (2019). *Production-Grade Container Orchestration*. Kubernetes. 2019. ICP license: 京ICP备17074266号-3. Retrieved 16-May-2019 from: <https://kubernetes.io/>

<sup>vii</sup> Rouse, Margaret, et al. (August 2017). *Kubernetes*. TechTarget Inc. 2019. Retrieved 16-May-2019 from: <https://searchitoperations.techtarget.com/definition/Google-Kubernetes>

<sup>viii</sup> Rouse, Margaret, et al. (August 2017). *Kubernetes*. TechTarget Inc. 2019. Retrieved 16-May-2019 from: <https://searchitoperations.techtarget.com/definition/Google-Kubernetes>

<sup>ix</sup> Jayanandana, Nilesh. (May 2<sup>nd</sup>, 2018). *Benefits of Kubernetes*. Medium Newspaper. Retrieved 16-May-2019 from: <https://medium.com/platformer-blog/benefits-of-kubernetes-e6d5de39bc48>

<sup>x</sup> Jayanandana, Nilesh. (May 2<sup>nd</sup>, 2018). *Benefits of Kubernetes*. Medium Newspaper. Retrieved 16-May-2019 from: <https://medium.com/platformer-blog/benefits-of-kubernetes-e6d5de39bc48>

<sup>xi</sup> Jayanandana, Nilesh. (May 2<sup>nd</sup>, 2018). *Benefits of Kubernetes*. Medium Newspaper. Retrieved 16-May-2019 from: <https://medium.com/platformer-blog/benefits-of-kubernetes-e6d5de39bc48>

<sup>xii</sup> Jayanandana, Nilesh. (May 2<sup>nd</sup>, 2018). *Benefits of Kubernetes*. Medium Newspaper. Retrieved 16-May-2019 from: <https://medium.com/platformer-blog/benefits-of-kubernetes-e6d5de39bc48>

<sup>xiii</sup> Williams, Alex, et al. *Kubernetes Deployment & Security Patterns*. The New Stack. 2019. 20180622. thenewstack.io. Retrieved 15-May-2019 from: <https://thenewstack.io/ebooks/kubernetes/kubernetes-deployment-and-security-patterns/>

<sup>xiv</sup> Williams, Alex, et al. *Kubernetes Deployment & Security Patterns*. The New Stack. 2019. 20180622. thenewstack.io. Retrieved 15-May-2019 from: <https://thenewstack.io/ebooks/kubernetes/kubernetes-deployment-and-security-patterns/>

<sup>xv</sup> Williams, Alex, et al. *Kubernetes Deployment & Security Patterns*. The New Stack. 2019. 20180622. thenewstack.io. Retrieved 15-May-2019 from: <https://thenewstack.io/ebooks/kubernetes/kubernetes-deployment-and-security-patterns/>

<sup>xvi</sup> Williams, Alex, et al. *Kubernetes Deployment & Security Patterns*. The New Stack. 2019. 20180622. thenewstack.io. Retrieved 15-May-2019 from: <https://thenewstack.io/ebooks/kubernetes/kubernetes-deployment-and-security-patterns/>

<sup>xvii</sup> Rouse, Margaret, et al. (August 2017). *Kubernetes*. TechTarget Inc. 2019. Retrieved 16-May-2019 from: <https://searchitoperations.techtarget.com/definition/Google-Kubernetes>

<sup>xviii</sup> Tsang, Daisy. (February 12<sup>th</sup>, 2018). *Kubernetes vs. Docker: What Does It Really Mean?* Sumo Logic. 2019. Retrieved 16-May-2019 from: <https://www.sumologic.com/blog/kubernetes-vs-docker/>

<sup>xix</sup> Treasury Board of Canada Secretariat. December 3, 2018. *Directive on Management of Information Technology*. Treasury Board of Canada Secretariat. Government of Canada. Retrieved 27-Dec-2018 from: <https://www.tbs-sct.gc.ca/pol/doc-eng.aspx?id=15249>

<sup>xx</sup> Williams, Alex, et al. *The State of the Kubernetes Ecosystem*. The New Stack. thenewstack.io. Retrieved 15-May-2019 from: <https://thenewstack.io/ebooks/kubernetes/state-of-kubernetes-ecosystem/>

<sup>xxi</sup> Chemitiganti, Vamsi, and Fray, Peter. (February 20<sup>th</sup>, 2019). *7 Key Considerations for Kubernetes in Production*. The New Stack. 2019. Retrieved 16-May-2019 from: <https://thenewstack.io/7-key-considerations-for-kubernetes-in-production/>

<sup>xxii</sup> Chemitiganti, Vamsi, and Fray, Peter. (February 20<sup>th</sup>, 2019). *7 Key Considerations for Kubernetes in Production*. The New Stack. 2019. Retrieved 16-May-2019 from: <https://thenewstack.io/7-key-considerations-for-kubernetes-in-production/>

1. Clayton, T. and Watson, R. (2018). Using Kubernetes to Orchestrate Container-Based Cloud and Microservices Applications. [online] Gartner.com. Available at: <https://www.gartner.com/doc/3873073/using-kubernetes-orchestrate-containerbased-cloud> [Accessed 6 Jul. 2018].

2. Anon, (2018). [online] Available at: <https://www.infoworld.com/article/3173266/containers/4-reasons-you-should-use-kubernetes.htm> [Accessed 6 Jul. 2018].

3. Contino. (2018). Who's Using Docker?. [online] Available at: <https://www.contino.io/insights/whos-using-docker> [Accessed 6 Jul. 2018].

4. Team, S. and Team, S. (2018). 3 Open Source Container Management Tools (Comparison & Review). [online] Shadow-Soft. Available at: <https://shadow-soft.com/open-source-container-management-tools/> [Accessed 6 Jul. 2018].

5. MSV, J. (2018). Kubernetes: An Overview – The New Stack. [online] The New Stack. Available at: <https://thenewstack.io/kubernetes-an-overview/> [Accessed 6 Jul. 2018].

~~6. Heron, P. (2018). Experimenting with containerised infrastructure for GOV.UK – Inside GOV.UK. [online] Insidegovuk.blog.gov.uk. Available at:~~

<https://insidegovuk.blog.gov.uk/2017/09/15/experimenting-with-containerised-infrastructure-for-gov-uk/> [Accessed 6 Jul. 2018].

7. CENGN. (2018). CENGN and CloudOps Collaborate to Train Industry on Docker and Kubernetes. [online] Available at: <https://www.cengn.ca/docker-kubernetes-training-jan18/> [Accessed 13 Jul. 2018].

8. Kubernetes.io. (2018). Kubernetes Basics – Kubernetes. [online] Available at: <https://kubernetes.io/docs/tutorials/kubernetes-basics/> [Accessed 13 Jul. 2018].

9. Idatalabs.com. (2018). Kubernetes commands 8.62% market share in Virtualization Management Software. [online] Available at: <https://idatalabs.com/tech/products/kubernetes> [Accessed 13 Jul. 2018].