

Introduction au génie logiciel pour le Cloud



Année académique :2021-2022

Présentation du formateur



M. OUEDRAOGO W. A. Marc Christian

Ingénieur des travaux en réseaux et maintenance
informatique

Ingénieur de conception en analyse et programmation

Doctorant en Intelligence Artificielle

PLAN GENERAL DU COURS

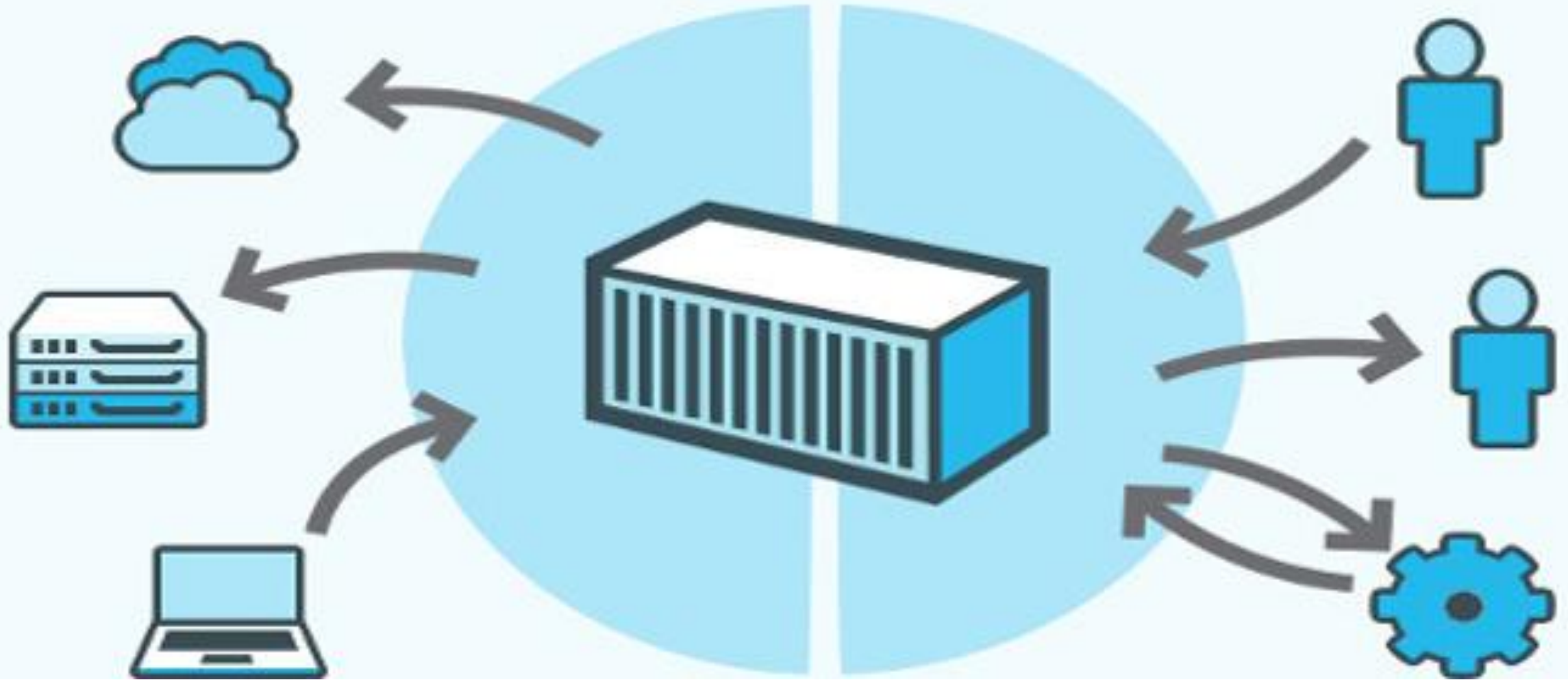
Séquence 1 : Les notions fondamentales de la virtualisation

Séquence 2 : Les notions fondamentales du cloud computing

Séquence 3 : La containerisation

Séquence 4 : La conception d'applications cloud





Séquence 3 : La conteneurisation

OG 3 : Comprendre la conteneurisation

- Décrire la conteneurisation
- Distinguer les conteneurs des machines virtuelles
- Distinguer les conteneurs des micro services
- Expliquer Orchestration des conteneurs

PLAN

Chapitre I : La conteneurisation

Chapitre 2 : Les conteneurs et les machines virtuelles

Chapitre 3 : Les conteneurs et les micro services

Chapitre 4 : Orchestration des conteneurs

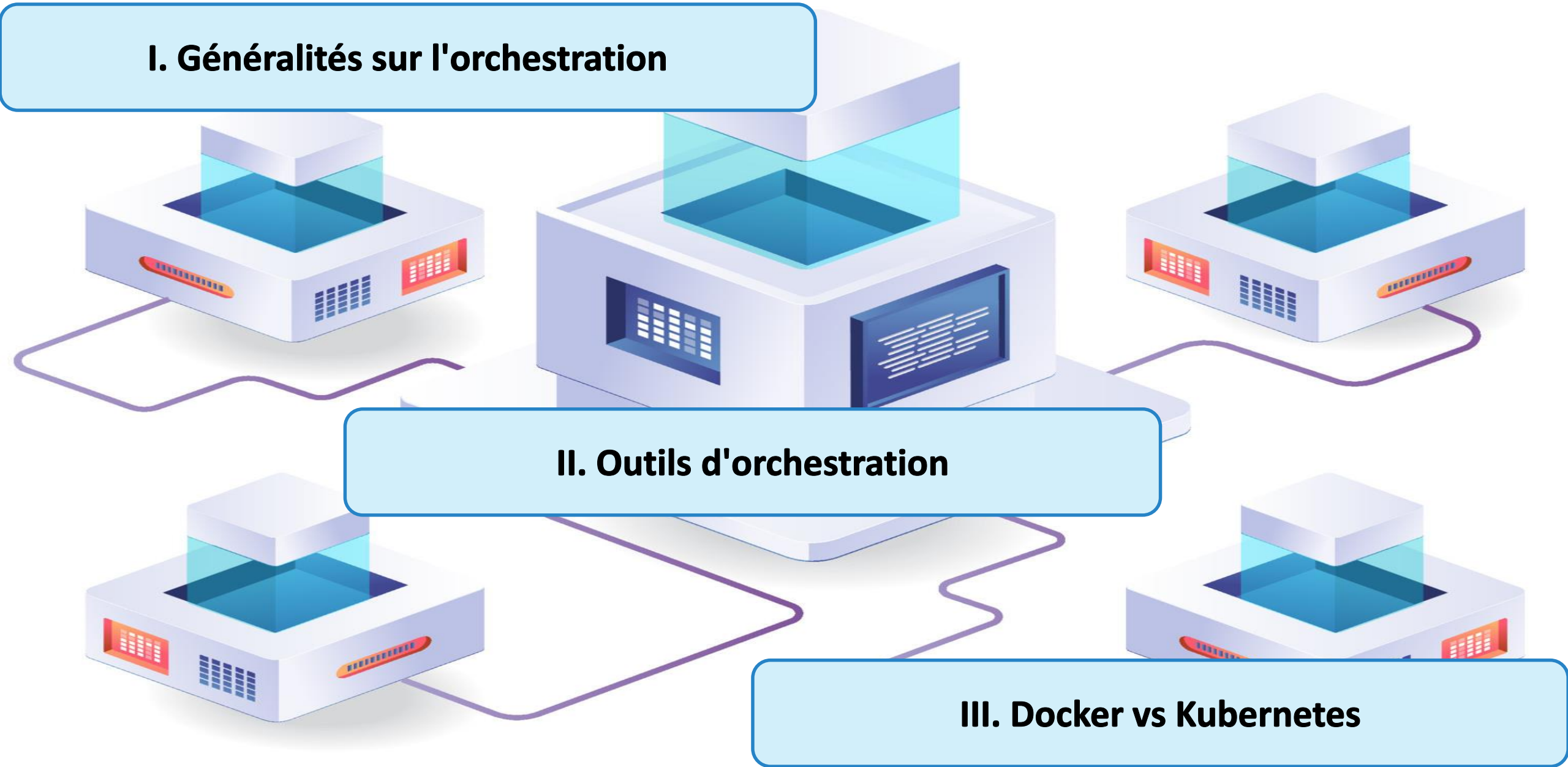


Chapitre 4 : Orchestration des conteneurs

I. Généralités sur l'orchestration

II. Outils d'orchestration

III. Docker vs Kubernetes



I. Généralités sur l'orchestration

L'orchestration de conteneurs automatise le déploiement, la gestion, la mise à l'échelle et la mise en réseau des conteneurs.

Les entreprises qui doivent déployer et gérer des centaines ou des milliers de conteneurs et d'hôtes Linux® peuvent bénéficier de l'orchestration de conteneurs.

L'orchestration de conteneurs peut être utilisée dans tout environnement où vous utilisez des conteneurs.

Elle peut vous aider à déployer la même application dans différents environnements sans avoir à la reconcevoir.

Et les microservices dans les conteneurs facilitent l'orchestration des services, notamment le stockage, la mise en réseau et la sécurité.

I. Généralités sur l'orchestration

Les conteneurs offrent à vos applications basées sur des microservices une unité de déploiement d'applications idéale et un environnement d'exécution autonome.

Ils permettent d'exécuter plusieurs parties d'une application de manière indépendante dans des microservices, sur le même matériel, avec un contrôle beaucoup plus important sur les pièces individuelles et les cycles de vie.

Avec les interfaces de programmation d'applications (API) et les équipes DevOps, les microservices conteneurisés constituent la base des applications "cloud-native".

I. Généralités sur l'orchestration

L'orchestration de conteneurs est utiliser pour automatiser et gérer des tâches telles que :

- ✓ L'approvisionnement et le déploiement
- ✓ la configuration et l'ordonnancement
- ✓ l'allocation des ressources
- ✓ la disponibilité des conteneurs
- ✓ Mise à l'échelle ou suppression des conteneurs en fonction de l'équilibrage des charges de travail dans votre infrastructure
- ✓ Équilibrage des charges et routage du trafic
- ✓ Surveillance de l'état des conteneurs
- ✓ Configuration des applications en fonction du conteneur dans lequel elles seront exécutées Sécurisation des interactions entre les conteneurs

I. Généralités sur l'orchestration

L'orchestration des conteneurs est essentielle pour travailler avec des conteneurs, et elle permet aux organisations d'en tirer tous les avantages.

Elle offre également ses propres avantages pour un environnement conteneurisé, notamment :

- ❑ **Des opérations simplifiées** : C'est l'avantage le plus important de l'orchestration de conteneurs et la principale raison de son adoption. Les conteneurs introduisent une grande quantité de complexité qui peut rapidement devenir hors de contrôle sans orchestration de conteneurs pour la gérer.

I. Généralités sur l'orchestration

L'orchestration des conteneurs est essentielle pour travailler avec des conteneurs, et elle permet aux organisations d'en tirer tous les avantages.

Elle offre également ses propres avantages pour un environnement conteneurisé, notamment :

- ❑ **Résilience** : Les outils d'orchestration de conteneurs peuvent redémarrer ou faire évoluer automatiquement un conteneur ou un cluster, ce qui renforce la résilience.
- ❑ **Sécurité accrue** : L'approche automatisée de l'orchestration de conteneurs permet de sécuriser les applications conteneurisées en réduisant ou en éliminant le risque d'erreur humaine.

I. Généralités sur l'orchestration

L'orchestration de conteneurs est l'automatisation d'une grande partie de l'effort opérationnel nécessaire pour exécuter des charges de travail et des services conteneurisés.

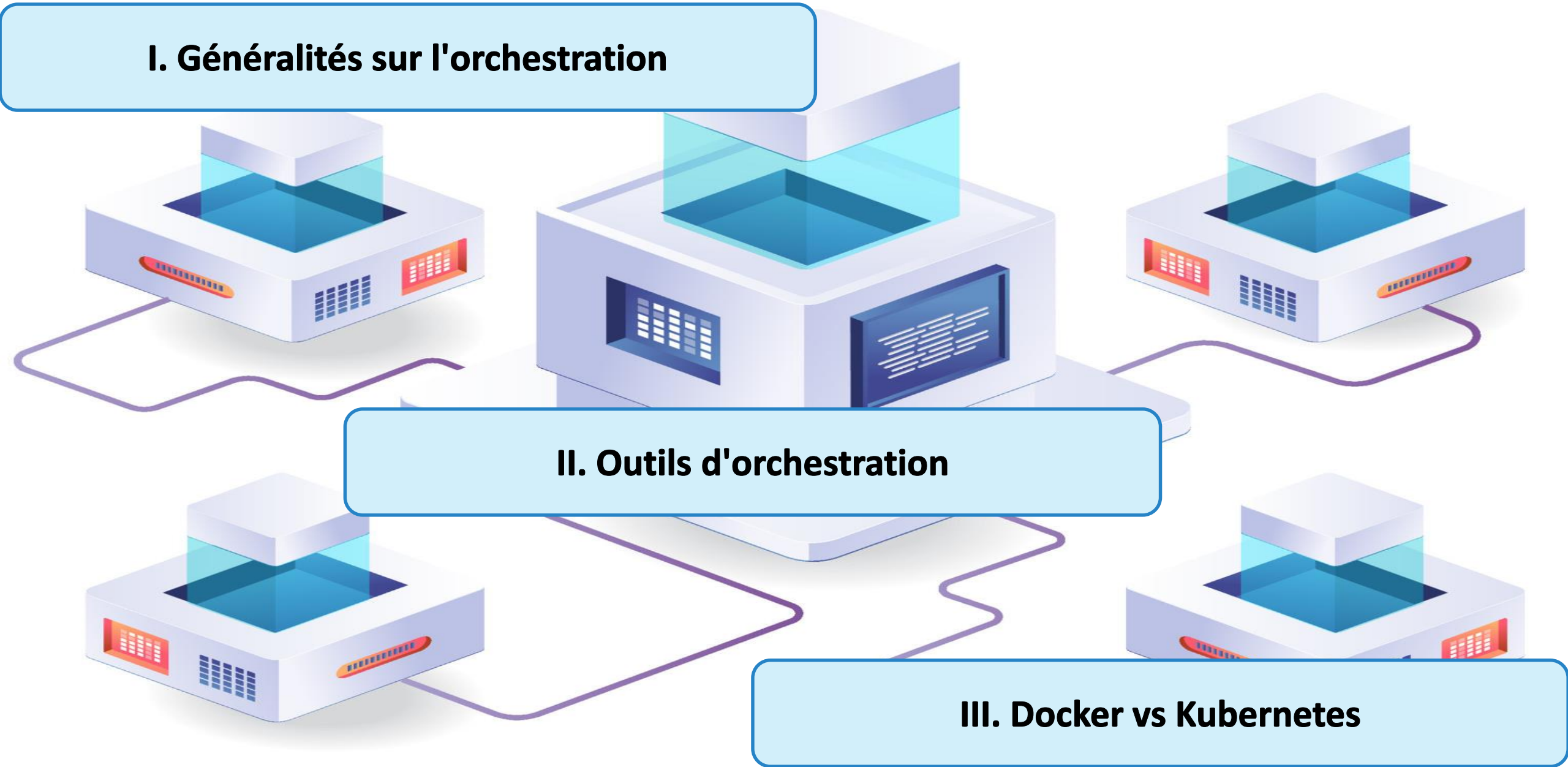
Cela inclut un large éventail d'éléments dont les équipes logicielles ont besoin pour gérer le cycle de vie d'un conteneur, notamment l'approvisionnement, le déploiement, la mise à l'échelle (ascendante et descendante), la mise en réseau, l'équilibrage des charges, etc.

Chapitre 4 : Orchestration des conteneurs

I. Généralités sur l'orchestration

II. Outils d'orchestration

III. Docker vs Kubernetes



II. Outils d'orchestration

Les outils d'orchestration de conteneurs fournissent un cadre pour gérer les conteneurs et l'architecture de microservices à l'échelle.

Il existe de nombreux outils d'orchestration de conteneurs qui peuvent être utilisés pour la gestion du cycle de vie des conteneurs.

Parmi les outils populaires, on peut citer :

- ✓ **Kubernetes,**
- ✓ **OpenShift,**
- ✓ **Docker Swarm,**
- ✓ **Docker Compose,**
- ✓ **Cloudify,**
- ✓ **MiniKube**
- ✓ **et Apache Mesos.**

II. Outils d'orchestration



kubernetes

II. Outils d'orchestration

Kubernetes est un outil d'orchestration de conteneurs open source qui a été développé et conçu à l'origine par des ingénieurs de Google.

En 2015, Google a fait don du projet Kubernetes à la toute nouvelle Cloud Native Computing Foundation.

II. Outils d'orchestration

L'orchestration Kubernetes vous permet de créer des services d'application qui englobent plusieurs conteneurs, de planifier les conteneurs dans un cluster, de mettre à l'échelle ces conteneurs et de gérer leur santé dans le temps.

Kubernetes élimine une grande partie des processus manuels impliqués dans le déploiement et la mise à l'échelle des applications conteneurisées.

Vous pouvez regrouper des groupes d'hôtes, qu'il s'agisse de machines physiques ou virtuelles, qui exécutent des conteneurs Linux, et Kubernetes vous fournit la plate-forme pour gérer facilement et efficacement ces clusters.

II. Outils d'orchestration

Plus largement, il vous aide à mettre en œuvre et à utiliser pleinement une infrastructure basée sur des conteneurs dans des environnements de production.

Ces clusters peuvent s'étendre sur des hôtes à travers des clouds publics, privés ou hybrides.

C'est pourquoi Kubernetes est une plateforme idéale pour l'hébergement d'applications natives du cloud qui nécessitent une mise à l'échelle rapide.

Kubernetes contribue également à la portabilité de la charge de travail et à l'équilibrage de la charge en vous permettant de déplacer les applications sans les redessiner.

II. Outils d'orchestration

Principaux composants de Kubernetes :

- ❑ **Cluster** : Un plan de contrôle et une ou plusieurs machines de calcul, ou nœuds.
- ❑ **Plan de contrôle** : L'ensemble des processus qui contrôlent les nœuds Kubernetes. C'est de là que proviennent toutes les affectations de tâches.
- ❑ **Kubelet** : Ce service s'exécute sur les nœuds et lit les manifestes des conteneurs et s'assure que les conteneurs définis sont démarrés et fonctionnent.
- ❑ **Pod** : Un groupe d'un ou plusieurs conteneurs déployés sur un seul nœud. Tous les conteneurs d'un pod partagent une adresse IP, un IPC, un nom d'hôte et d'autres ressources.

II. Outils d'orchestration

Kubernetes, souvent décrit comme le "Linux du cloud", est la plateforme d'orchestration de conteneurs la plus populaire pour une raison bien précise.

Voici quelques-unes de ces raisons :

❑ Opérations automatisées

Kubernetes est doté d'une puissante API et d'un outil en ligne de commande, appelé **kubectl**, qui prend en charge une grande partie de la gestion des conteneurs en vous permettant d'automatiser vos opérations. Le modèle de contrôleur de Kubernetes garantit que les applications/conteneurs s'exécutent exactement comme spécifié.

II. Outils d'orchestration

Kubernetes, souvent décrit comme le "Linux du cloud", est la plateforme d'orchestration de conteneurs la plus populaire pour une raison bien précise.

Voici quelques-unes de ces raisons :

☐ **Abstraction de l'infrastructure**

Kubernetes gère les ressources mises à sa disposition en votre nom. Les développeurs peuvent ainsi se concentrer sur l'écriture du code de l'application et non sur l'infrastructure sous-jacente de calcul, de réseau ou de stockage.

II. Outils d'orchestration

Kubernetes, souvent décrit comme le "Linux du cloud", est la plateforme d'orchestration de conteneurs la plus populaire pour une raison bien précise.

Voici quelques-unes de ces raisons :

☐ Surveillance de la santé des services

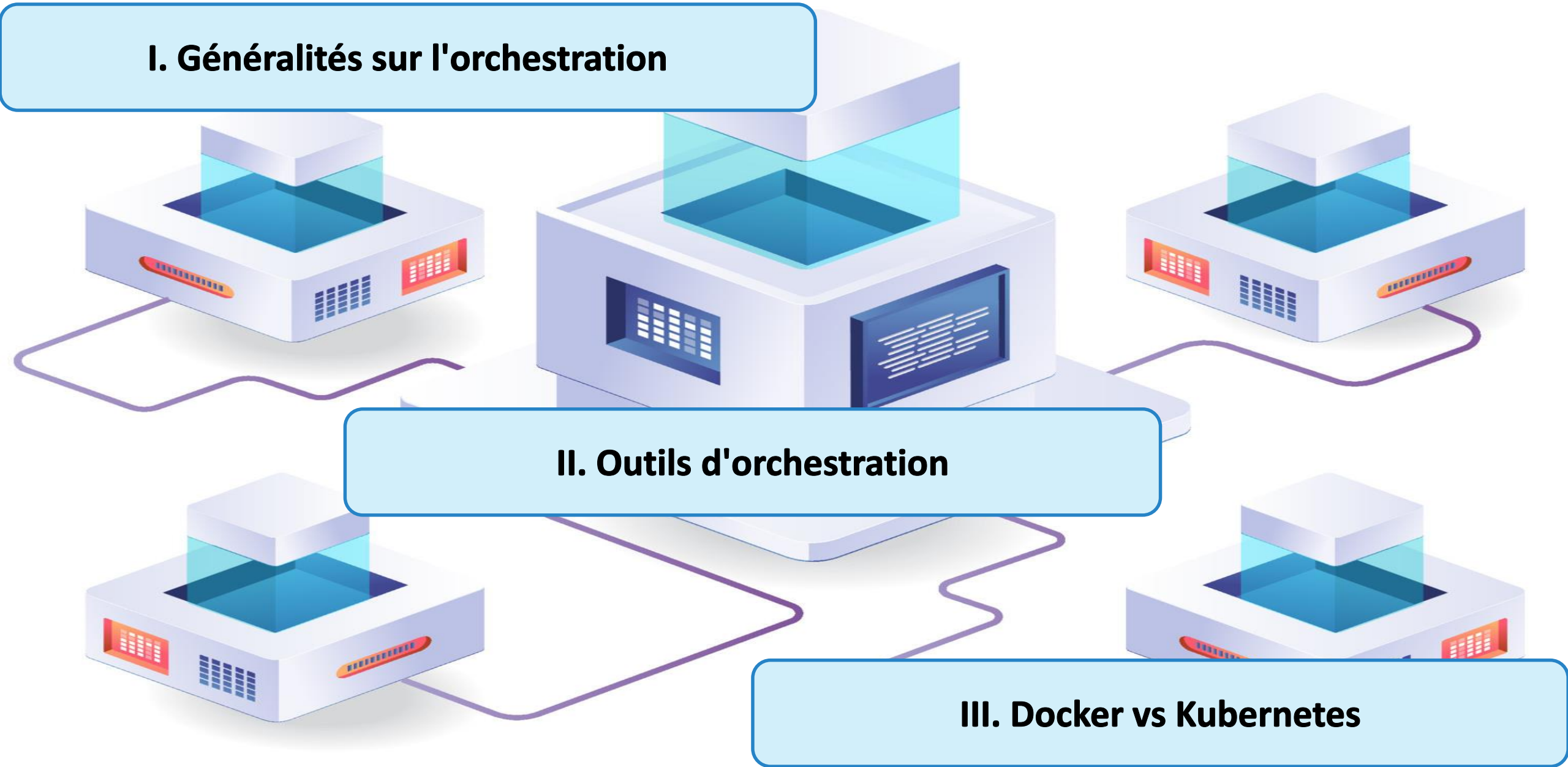
Kubernetes surveille l'environnement en cours d'exécution et le compare à l'état souhaité. Il effectue des contrôles de santé automatisés sur les services et redémarre les conteneurs qui ont échoué ou se sont arrêtés. Kubernetes ne rend les services disponibles que lorsqu'ils sont en cours d'exécution et prêts.

Chapitre 4 : Orchestration des conteneurs

I. Généralités sur l'orchestration

II. Outils d'orchestration

III. Docker vs Kubernetes



III. Docker vs Kubernetes

Docker est une plateforme spécifique pour la création de conteneurs, y compris le moteur d'exécution de conteneurs Docker Engine, tandis que l'orchestration de conteneurs est un terme plus large faisant référence à l'automatisation du cycle de vie de tout conteneur.

Docker comprend également Docker Swarm, qui est l'outil d'orchestration de conteneurs propre à la plateforme et qui peut démarrer automatiquement les conteneurs Docker.

III. Docker vs Kubernetes

Docker est une plateforme et un moteur d'exécution de conteneurisation et Kubernetes est une plateforme permettant d'exécuter et de gérer des conteneurs à partir de nombreux moteurs d'exécution de conteneurs.

Kubernetes prend en charge de nombreux moteurs d'exécution de conteneurs, dont Docker.

III. Docker vs Kubernetes

Si Docker Swarm et Kubernetes sont tous deux des plateformes d'orchestration de conteneurs, lequel choisir ?

Docker Swarm nécessite généralement moins d'installation et de configuration que Kubernetes si vous construisez et exploitez votre propre infrastructure.

Il offre les mêmes avantages que Kubernetes, comme le déploiement de votre application au moyen de fichiers YAML déclaratifs, la mise à l'échelle automatique des services jusqu'à l'état souhaité, l'équilibrage de la charge entre les conteneurs au sein du cluster, ainsi que la sécurité et le contrôle de l'accès à vos services.

Si vous avez peu de charges de travail en cours d'exécution, que cela ne vous dérange pas de gérer votre propre infrastructure ou que vous n'avez pas besoin d'une fonctionnalité spécifique offerte par Kubernetes, alors Docker Swarm peut être un excellent choix.

III. Docker vs Kubernetes

Kubernetes est plus complexe à mettre en place au début, mais offre plus de flexibilité et de fonctionnalités. Il bénéficie également d'un large soutien de la part d'une communauté open source active.

Kubernetes prend en charge plusieurs stratégies de déploiement dès le départ, peut gérer l'entrée de votre réseau et fournit une observabilité dès le départ dans vos conteneurs.

Tous les grands fournisseurs de cloud computing proposent des services Kubernetes gérés qui facilitent considérablement la prise en main et permettent de tirer parti des fonctionnalités natives du cloud computing, comme l'auto-scaling.

Si vous exécutez de nombreuses charges de travail et avez besoin d'une interopérabilité **"cloud native"**, et si vous avez de nombreuses équipes dans votre organisation, ce qui crée le besoin d'une plus grande isolation des services, alors Kubernetes est probablement la plate-forme que vous devriez envisager.

CONCLUSION

**MERCI DE VOTRE
AIMABLE ATTENTION!!!**