Maîtrisez Docker La Révolution des Conteneurs

Présenté par :

Mr Bonitah RAMBELOSON

Ingénieur Consultant DevOps et Cloud

Public Concerné

- Développeurs et ingénieurs logiciels
- Administrateurs système et DevOps
- Architectes solutions et cloud
- Étudiants en informatique et technologies de l'information
- Toute personne intéressée par l'optimisation du cycle de développement et de déploiement des applications

Prérequis :

➢ Compétences en Informatique :

- Connaissances de base en ligne de commande (CLI)
- Familiarité avec les concepts de réseaux et de systèmes d'exploitation
- Expérience avec au moins un langage de programmation (facultatif mais recommandé)

Prérequis

➢ Ressources Matérielles :

- Un ordinateur avec accès administrateur pour installer Docker
- Connexion Internet pour télécharger Docker et accéder aux ressources en ligne
- Environnement de développement intégré (IDE) ou éditeur de texte pour écrire des Dockerfiles

Objectifs de la Présentation

COMPRENDRE LES FONDAMENTAUX DE DOCKER ET SON IMPORTANCE DANS LE DÉVELOPPEMENT MODERNE

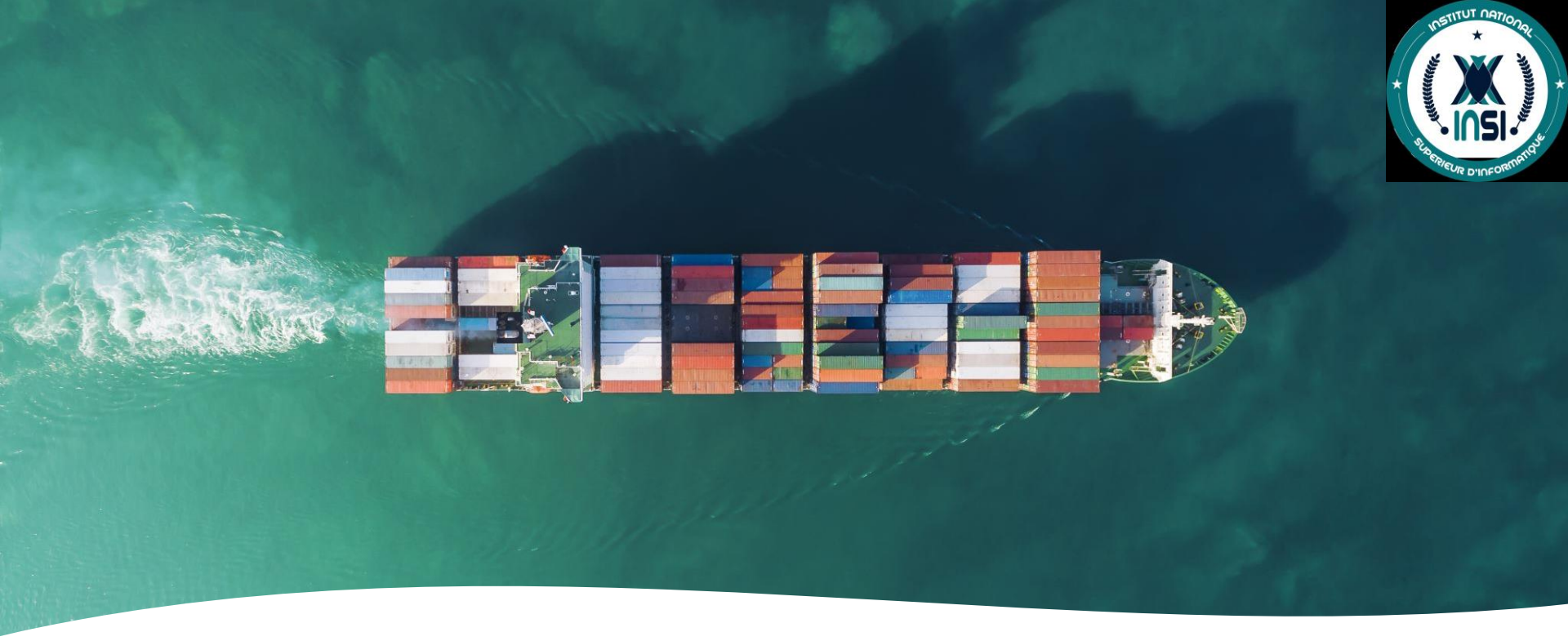
EXPLORER LES CONCEPTS CLÉS ET LES AVANTAGES DE L'UTILISATION DE DOCKER

APPRENDRE À INSTALLER ET CONFIGURER DOCKER SUR DIFFÉRENTES PLATEFORMES

DÉCOUVRIR LES MEILLEURES PRATIQUES POUR CRÉER ET GÉRER DES CONTENEURS DOCKER

ÉTUDIER DES CAS D'UTILISATION RÉELS ET DES EXEMPLES D'IMPLÉMENTATION RÉUSSIE

Formateur Consultant | Ingénieur DevOps & Cloud



Introduction à Docker

Docker est une plateforme open-source qui automatise le déploiement, la mise à l'échelle et l'exécution des applications dans des conteneurs. Les conteneurs permettent aux développeurs de regrouper une application et toutes ses dépendances dans un environnement standardisé, garantissant que l'application fonctionne de manière cohérente sur n'importe quelle infrastructure.

Formateur Consultant | Ingénieur DevOps & Cloud

Pourquoi utiliser à Docker ?

Avec Docker vous pouvez :

- Isoler les applications : Chaque conteneur fonctionne comme un environnement léger et isolé, ce qui évite les conflits entre les applications.
- Accélérer le développement : Les conteneurs se lancent rapidement et permettent de tester les applications dans des environnements identiques à la production.
- Simplifier le déploiement : Docker facilite le déploiement continu et l'intégration continue (CI/CD), permettant de livrer des applications plus rapidement et plus efficacement.
- Docker est devenu un outil essentiel pour les développeurs et les équipes DevOps, transformant la manière dont les applications sont construites, livrées et exécutées.

Formateur Consultant | Ingénieur DevOps & Cloud

Plan du Cours

PARTIE I

- Concepts de Base (Historique et évolution de Docker)
- Conteneurs vs Machines Virtuelles
- Architecture de Docker

PARTIE II

- Installation de Docker
- Images Docker
- Conteneurs docker
- Registre Docker (Docker Hub)

PARTIE III

- Création d'Images Docker avec Dockerfile
- Persistance des données sur Docker (Docker volumes)
- Réseaux Docker (Docker Network)

PARTIE IV

- Docker Compose (application multi-conteneurs)

PARTIE V

- Études de cas réussies (**TD** et **TP**)
- Conclusion et Ressources

PARTIE I

Concepts de Base : Historique et Évolution de Docker

- **Qu'est-ce que Docker ?**

• Docker est une plateforme open-source qui automatise le déploiement, la mise à l'échelle et l'exécution des applications dans des conteneurs.

• Les conteneurs permettent d'emballer une application et toutes ses dépendances dans un environnement standardisé, garantissant une exécution cohérente sur n'importe quelle infrastructure.

Concepts de Base : Historique et Évolution de Docker

- **Historique de Docker**

- **2010 :** Lancement de la technologie de conteneurs Linux (LXC) qui a inspiré Docker.
- **2013 :** Docker est lancé par Solomon Hykes lors de la conférence PyCon.
- **2014 :** DockerHub est lancé, offrant un registre public pour partager des images Docker.
- **2015 :** Introduction de Docker Compose pour gérer des applications multi-conteneurs.
- **2016 :** Docker Swarm est intégré pour l'orchestration native des conteneurs.
- **2017 :** Docker adopte Kubernetes comme solution d'orchestration, renforçant l'interopérabilité.

Concepts de Base : Historique et Évolution de Docker

- **Évolution de Docker**

- **Adoption Croissante :** Docker est rapidement adopté par les entreprises pour moderniser leurs infrastructures.
- **Écosystème Riche :** Développement d'outils et de services autour de Docker pour améliorer la gestion des conteneurs.
- **Communauté Active :** Une communauté open-source dynamique contribue à l'amélioration continue de Docker.
- **Intégration CI/CD :** Docker devient un élément clé des pipelines d'intégration et de déploiement continus.

Concepts de Base : Historique et Évolution de Docker

- **Pourquoi Docker ?**

- **Portabilité :** Les applications Docker fonctionnent de manière cohérente sur n'importe quel environnement.
- **Efficacité :** Les conteneurs sont légers et se lancent rapidement, optimisant l'utilisation des ressources.
- **Isolation :** Chaque conteneur est isolé, évitant les conflits entre les applications.

Conteneurs vs Machines Virtuelles

- Machines Virtuelles (VM)

- **Définition :** Une machine virtuelle est une émulation d'un système informatique, comprenant un système d'exploitation complet et des ressources matérielles virtuelles.
- **Isolation :** Chaque VM inclut un noyau d'OS complet, offrant une isolation totale entre les applications.
- **Ressources :** Les VM sont souvent lourdes et nécessitent plus de ressources (CPU, mémoire) pour fonctionner.
- **Démarrage :** Le démarrage d'une VM peut prendre plusieurs minutes en raison de l'initialisation du système d'exploitation.
- **Utilisation :** Idéales pour exécuter plusieurs systèmes d'exploitation sur un même matériel ou pour des environnements nécessitant une isolation complète.

Conteneurs vs Machines Virtuelles

- Conteneurs

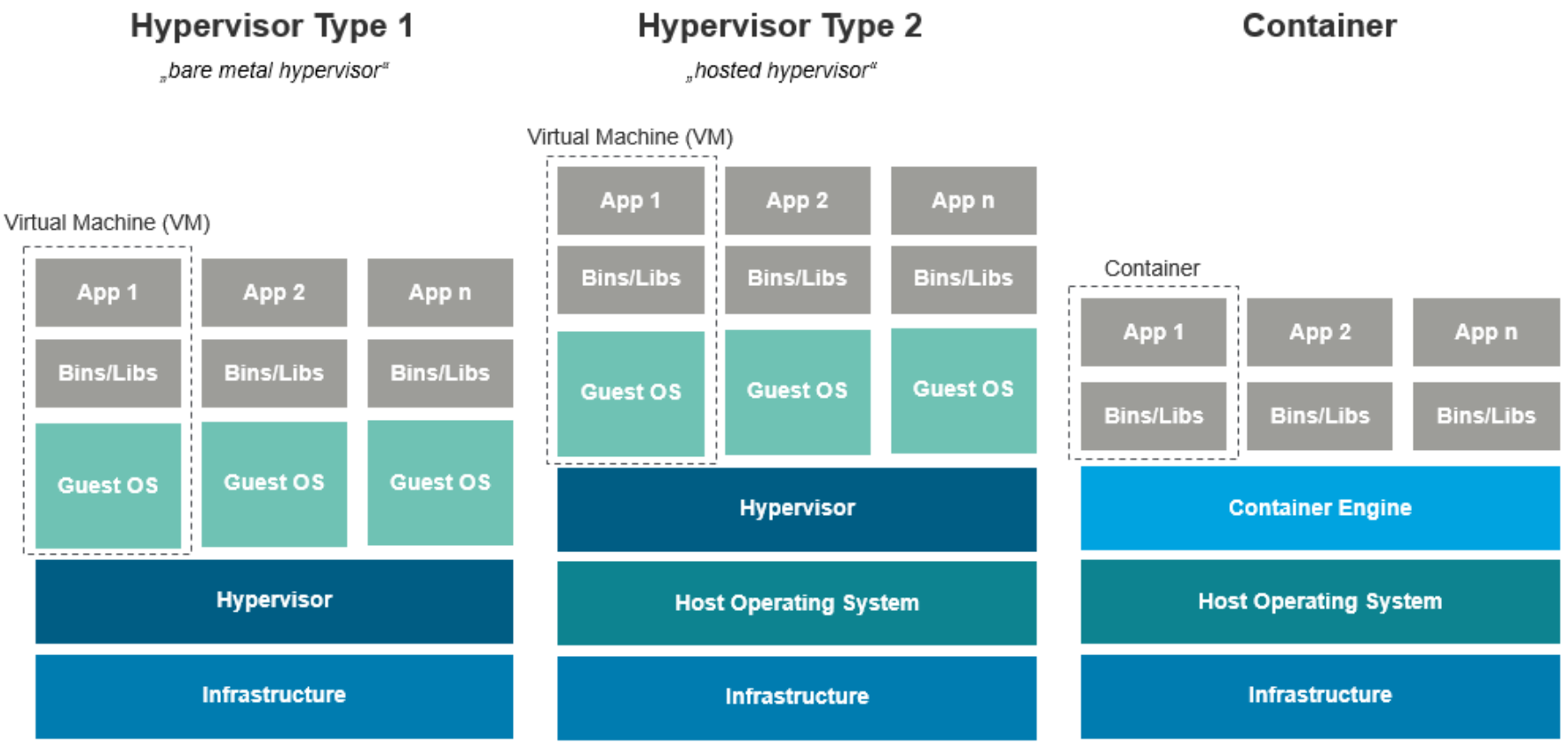
- **Définition :** Un conteneur est un environnement léger et portable qui regroupe une application et toutes ses dépendances.
- **Isolation :** Les conteneurs partagent le noyau du système d'exploitation hôte, offrant une isolation au niveau des processus.
- **Ressources :** Les conteneurs sont légers et utilisent moins de ressources que les VM, car ils ne nécessitent pas un OS complet.
- **Démarrage :** Les conteneurs se lancent en quelques secondes, ce qui accélère le développement et le déploiement.
- **Utilisation :** Parfaits pour le développement et le déploiement d'applications modernes, les microservices, et les environnements CI/CD.

Conteneurs vs Machines Virtuelles

- Comparaison

- **Efficacité :** Les conteneurs sont plus efficaces en termes de ressources et de rapidité de démarrage.
- **Portabilité :** Les conteneurs offrent une portabilité accrue, permettant de déployer des applications de manière cohérente sur différentes infrastructures.
- **Gestion :** Les conteneurs sont plus faciles à gérer et à orchestrer, surtout dans des environnements à grande échelle.

Architecture Conteneurs vs Machines Virtuelles



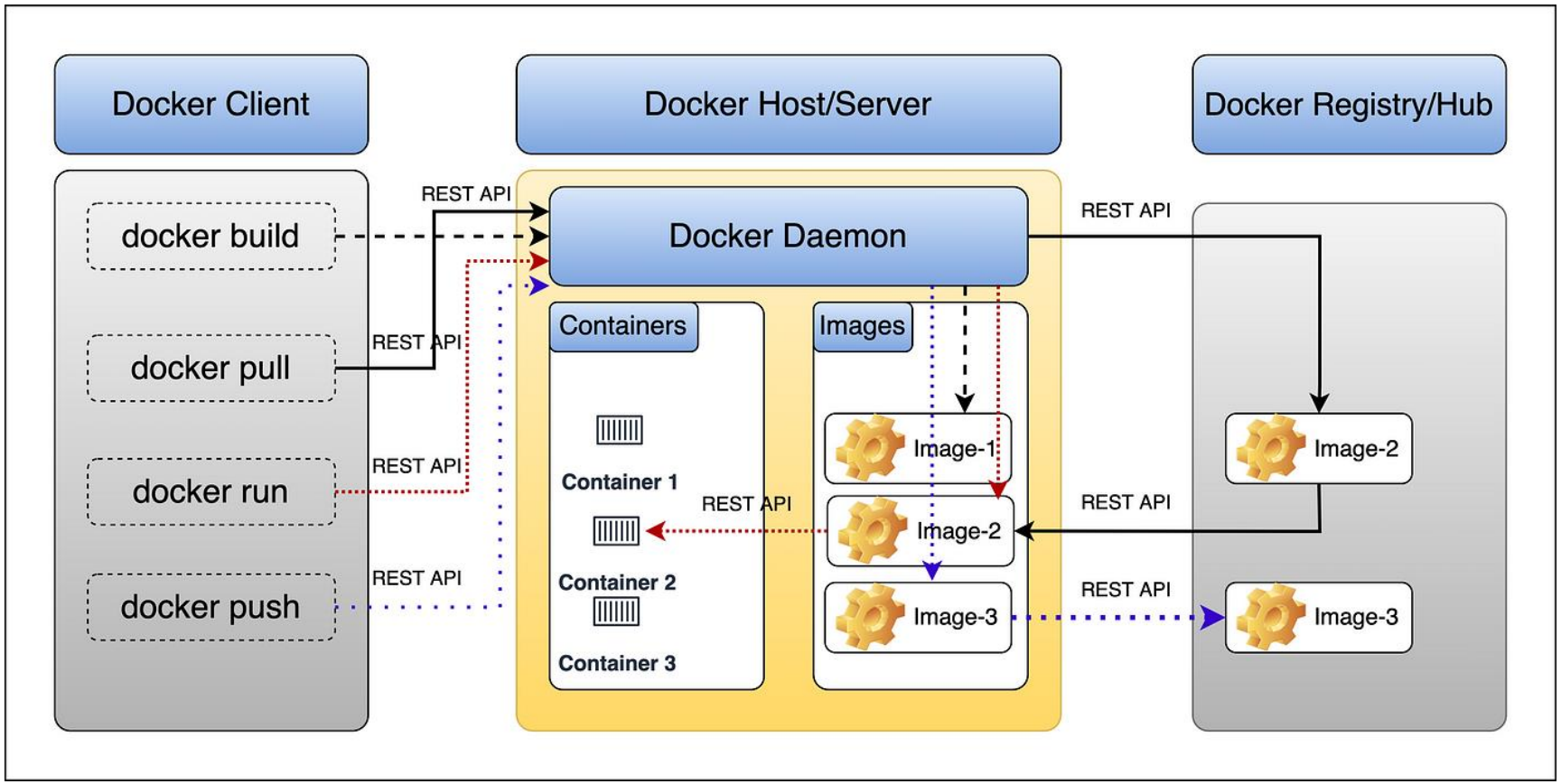
Formateur Consultant | Ingénieur DevOps & Cloud

Conclusion

Conteneurs vs Machines Virtuelles

Les conteneurs Les conteneurs, comme ceux utilisés par Docker, offrent une solution légère et efficace pour le déploiement d'applications, tout en maintenant un niveau d'isolation suffisant pour la plupart des cas d'utilisation.

Architecture de Docker



Composants Clés

- Docker Engine

  - **Docker Daemon (dockerd) :** Le cœur de Docker, qui gère les conteneurs, les images, les réseaux et les volumes.

  - **Docker CLI :** Interface en ligne de commande permettant aux utilisateurs d'interagir avec le Docker Daemon.

  - **Docker API :** Interface permettant aux applications d'interagir avec le Docker Daemon.

Composants Clés

- Images Docker

- **Définition :** Une image Docker est un modèle en lecture seule utilisé pour créer des conteneurs.

- **Couches :** Les images sont constituées de couches, chacune représentant une modification (ajout, suppression, modification de fichiers).

- **Dockerfile :** Fichier texte contenant les instructions pour construire une image Docker.

Composants Clés

- Conteneurs

  - **Instance d'une Image :** Un conteneur est une instance exécutable d'une image Docker.

  - **Isolation :** Chaque conteneur fonctionne dans un environnement isolé, partageant le noyau de l'hôte mais avec ses propres ressources.

  - **Portabilité :** Les conteneurs peuvent être déployés sur n'importe quelle infrastructure compatible avec Docker.

Composants Clés

- Registres Docker

  - **Docker Hub :** Registre public où les utilisateurs peuvent stocker et partager des images Docker.

  - **Registres Privés :** Solutions pour stocker des images Docker de manière privée, souvent utilisées dans les entreprises.

Composants Clés

- Réseaux et Volumes

Réseaux : Docker permet de créer des réseaux virtuels pour connecter les conteneurs entre eux.

Volumes : Stockage persistant pour les données générées par les conteneurs, permettant de conserver les données même après l'arrêt du conteneur.

Fonctionnement

- **Création d'un Conteneur**

- Un utilisateur exécute une commande Docker via le CLI.
- Le Docker Daemon reçoit la commande via l'API.
- Le Daemon crée un conteneur à partir d'une image Docker.
- Le conteneur s'exécute en utilisant les ressources de l'hôte

PARTIE II

Installation de docker

Installation docker sur linux

Formateur Consultant | Ingénieur DevOps & Cloud

Installer et configurer Docker sur Ubuntu

Distribution DEBIAN

- **Mettre à jour votre système d'exploitation**

Il faut mettre à jour le système pour le rendre plus sécurisé et fiable pour l'installation du Docker.

- **Exécutez les deux commandes suivantes :**

sudo apt update

sudo apt upgrade

Installer les paquets pré-requis

Distribution DEBIAN

Une fois qu'on a mis à jour le système, on doit installer quelques paquets nécessaires avant d'installer Docker. On pourra le faire avec une seule commande :

sudo apt-get install curl apt-transport-https ca-certificates software-properties-common

Distribution DEBIAN

Ajouter les dépôts Docker

Maintenant, nous devons ajouter les dépôts Docker.

D’abord, nous ajoutons la clé GPG avec la commande suivante dans la ligne de commande :

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable“

sudo apt update

Installer Docker

Distribution DEBIAN

Le processus d'installation sera ainsi beaucoup plus facile. Cela nous permet d'utiliser la méthode d'installation officiellement supportée.

sudo apt install docker-ce

sudo systemctl status docker

sudo usermod -aG docker ${USER}

Installer et configurer Docker sur Fedora

Distribution REDHAT

- **Mettre à jour votre système d'exploitation**

Il faut mettre à jour le système pour le rendre plus sécurisé et fiable pour l'installation du Docker.

- **Exécutez les deux commandes suivantes :**

dnf update -y

Installer les paquets pré-requis

Distribution REDHAT

Une fois qu'on a mis à jour le système, on doit installer quelques paquets nécessaires avant d'installer Docker. On pourra le faire avec une seule commande :

dnf -y install dnf-plugins-core

Distribution REDHAT

Ajouter les dépôts Docker

Maintenant, nous devons ajouter les dépôts Docker.

D’abord, nous ajoutons la clé GPG avec la commande suivante dans la ligne de commande :

dnf config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo

Installer Docker

Distribution REDHAT

Le processus d'installation sera ainsi beaucoup plus facile. Cela nous permet d'utiliser la méthode d'installation officiellement supportée.

dnf install docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-compose –y

systemctl start docker

systemctl enable docker

systemctl status docker

Installation de docker

Installation docker sur Windows préférable windows 10 et plus.

Installer et configurer Docker sur Windows 10

Docker Desktop pour Windows 10 :

- Initialement conçu pour Linux, Docker a étendu sa portée avec Docker Desktop, désormais disponible sur Windows. Explorons ensemble les prérequis et les étapes nécessaires pour installer Docker sur Windows 10, afin de développer et tester vos applications dans des environnements conteneurisés isolés.

Installer et configurer Docker sur Windows 10

Spécifications techniques

- Docker Desktop offre une suite complète d'outils pour exécuter des applications Docker sur votre poste de travail. Il inclut le moteur Docker, les outils en ligne de commande (CLI) et la fonctionnalité Docker Compose.
- De plus, il propose des fonctionnalités avancées telles que le téléchargement automatique d'images, le partage de conteneurs via le cloud, ainsi que l'intégration avec des environnements de développement intégrés (IDE) et des outils de construction.

Installer et configurer Docker sur Windows 10

Spécifications techniques

Votre système Windows 10 doit satisfaire aux prérequis suivants pour garantir la bonne installation de Docker

- **Processeur (CPU) :** 64 bits avec traduction d'adresse de deuxième niveau

(« Second level address translation » ou SLAT)

- **Mémoire vive (RAM) :** 4 Go
- **Système d'exploitation :** Windows 10 Famille, Professionnel ou Entreprise
- **Virtualisation :** il convient d'activer la virtualisation du matériel dans le système d'entrée/sortie de base (« Basic input/output system » ou BIOS) de l'ordinateur
- **Hyper-V :** facultatif pour Windows Professionnel ou Entreprise
- **Sous-système Windows pour Linux 2 (WSL 2) :** il convient de l'activer pour Windows Famille
- **Espace en disque dur :** 20 Go minimum
- **Connexion Internet :** suffisante pour télécharger des paquets

Installer et configurer Docker sur Windows 10

Instructions étape par étape

Deux méthodes sont disponibles pour installer Docker sur Windows 10 : l'utilisation de l'assistant d'installation graphique ou l'installation via la ligne de commande. Explorons ensemble ces deux approches, compatibles avec toutes les versions de Windows 10.

Installer et configurer Docker sur Windows 10

Étape 1 : télécharger et exécuter Docker Desktop

- Rendez-vous sur Docker Hub pour télécharger la version officielle de Docker Desktop.
- Une fois le téléchargement effectué, lancez le fichier Docker Desktop Installer.exe.
- Si votre système supporte à la fois Hyper-V et WSL 2, vous devrez choisir l'une de ces deux options.
- Si Hyper-V ou WSL 2 est déjà activé, seule l'option correspondante sera disponible. Vous pouvez également créer un raccourci sur votre bureau pour un accès rapide.
- L'assistant d'installation vous permet de choisir Hyper-V ou WSL 2 pour configurer Docker

Installer et configurer Docker sur Windows 10

Étape 2 : Utiliser la ligne de commande pour installer Docker sous Windows

Vous avez également la possibilité d'installer Docker sur Windows 10 via la ligne de commande. Pour ce faire, il vous suffit d'exécuter la commande suivante dans un terminal :

```
$ "Docker Desktop Installer.exe" install
```

Il est également possible de passer par PowerShell pour l'installation :

```
Start-Process 'Docker Desktop Installer.exe' -Wait install
```

Vous pouvez utiliser la ligne de commande Windows (cmd) de façon similaire pour l'installation :

```
start /w "" "Docker Desktop Installer.exe" install
```

Installer et configurer Docker sur Windows 10

Étape 3 : Redémarrer Windows

Après avoir terminé l'installation, il est recommandé de redémarrer votre système.

Si le compte administrateur diffère du compte utilisateur, vous devrez ajouter ce dernier au groupe docker-users.

Pour cela, accédez à « Gestion de l'ordinateur » en tant qu'administrateur, puis naviguez vers Utilisateurs et groupes locaux/Groupes/docker-users. Ajoutez l'utilisateur au groupe en effectuant un clic droit. Une nouvelle connexion sera nécessaire pour appliquer les modifications.

Vous pouvez également réaliser cette opération via le terminal. Remplacez <user> par votre nom d'utilisateur :

```
$  net localgroup docker-users <user> /add
```

Installer et configurer Docker sur Windows 10

Étape 4 : Démarrer Docker Desktop

Après le redémarrage de votre système, ouvrez Docker Desktop et lancez votre premier conteneur. Une fois l'installation complétée, vous pourrez également utiliser les commandes Docker directement depuis la ligne de commande.

Installation de docker

Installer Docker sur MacOS

Formateur Consultant | Ingénieur DevOps & Cloud

Installer et configurer Docker sur Mac

Prérequis à l'installation

Afin d'installer Docker sur votre mac, il va falloir avoir la configuration suivante:

- macOS 10.14 ou plus (Mojave, Catalina ou Big Sur)
- Ne pas avoir VirtualBox dans une version inférieure à 4.3.30 d'installé sur son Mac

Installer et configurer Docker sur Mac

Installation

- Accédez au site de Docker Hub et téléchargez l'installateur adapté à votre Mac. Veillez à sélectionner la version correspondant à votre type de processeur, qu'il soit Intel ou Apple.

- Pour identifier le processeur de votre Mac, cliquez sur le menu Pomme situé dans le coin supérieur gauche de votre écran, puis sélectionnez "**À propos de ce Mac**". La deuxième ligne vous indiquera si votre processeur est un Intel ou un Apple.

Installer et configurer Docker sur Mac

Installation

- Après avoir téléchargé le fichier Docker.dmg, double-cliquez pour l'ouvrir, puis faites glisser l'icône Docker vers le dossier Applications.
- Pour installer Docker, il suffit de déplacer l'icône Docker dans le dossier Applications.
- Vous pouvez désormais lancer Docker depuis le Finder, dans l'onglet "Applications", ou en utilisant la recherche Spotlight (Cmd + Espace) en tapant "Docker".
- Vous verrez l'icône Docker apparaître dans la barre de menu en haut à droite de votre écran. Une fois que les conteneurs cessent de clignoter, Docker est prêt à l'emploi.
- L'icône Docker sera visible dans la barre de menu en haut à droite de votre Mac.

IMAGES DOCKER

C’est quoi ?

Les images Docker sont des fichiers légers, autonomes et exécutables qui contiennent tout ce dont une application a besoin pour s'exécuter : le code, la runtime, les bibliothèques, les variables d'environnement et les paramètres de configuration.

IMAGES DOCKER

Couches d'une Image

- Les images Docker sont constituées de couches. Chaque instruction dans un Dockerfile crée une nouvelle couche.
- Les couches permettent de réutiliser des parties communes entre différentes images, ce qui économise de l'espace et accélère les constructions.

IMAGES DOCKER

Commandes de Base

Construire une Image :
docker build -t nom_image .

Lister les Images : docker images

Exécuter un Conteneur à partir d'une Image : docker run nom_image

Formateur Consultant | Ingénieur DevOps & Cloud

IMAGES DOCKER

Les images Docker sont des fichiers binaires qui n'ont pas d'extension spécifique.

Elles sont généralement stockées dans un format propriétaire de Docker et sont manipulées à l'aide des commandes Docker CLI.

Voici quelques points clés concernant les images Docker :

Format de Fichier :

- Les images Docker sont des archives compressées qui contiennent plusieurs couches. Chaque couche représente une modification ou une instruction dans le Dockerfile.
- Les images sont souvent stockées dans des registres Docker (comme Docker Hub) et téléchargées au besoin.

IMAGES DOCKER

Les images Docker sont des fichiers binaires qui n'ont pas d'extension spécifique.

Elles sont généralement stockées dans un format propriétaire de Docker et sont manipulées à l'aide des commandes Docker CLI.

Voici quelques points clés concernant les images Docker :

Extensions :

- Les images Docker n'ont pas d'extension de fichier spécifique. Lorsque vous les manipulez avec Docker, elles sont généralement référencées par leur nom et leur tag (par exemple, ubuntu:latest).
- Cependant, lorsque vous exportez une image Docker en utilisant la commande docker save, vous pouvez lui donner une extension .tar pour indiquer qu'il s'agit d'une archive. Par exemple :

```
docker save -o mon_image.tar nom_image
```

IMAGES DOCKER

Les images Docker sont des fichiers binaires qui n'ont pas d'extension spécifique.

Elles sont généralement stockées dans un format propriétaire de Docker et sont manipulées à l'aide des commandes Docker CLI.

Voici quelques points clés concernant les images Docker :

Lecture/Écriture :

- Les images Docker sont en lecture seule. Une fois qu'une image est construite, elle ne peut pas être modifiée directement. Si vous devez apporter des modifications, vous devez créer une nouvelle image basée sur l'image existante.
- Les conteneurs, en revanche, sont des instances exécutables des images et peuvent avoir des couches en lecture/écriture. Les modifications apportées à un conteneur en cours d'exécution ne sont pas persistées dans l'image originale.

IMAGES DOCKER

Les images Docker sont des fichiers binaires qui n'ont pas d'extension spécifique.

Elles sont généralement stockées dans un format propriétaire de Docker et sont manipulées à l'aide des commandes Docker CLI.

Voici quelques points clés concernant les images Docker :

Stockage :

Les images Docker sont stockées dans le système de fichiers local de Docker sur la machine hôte. Vous pouvez voir où elles sont stockées en utilisant la commande docker info et en regardant le champ Docker Root Dir.

IMAGES DOCKER

Les images Docker sont des fichiers binaires qui n'ont pas d'extension spécifique.

Elles sont généralement stockées dans un format propriétaire de Docker et sont manipulées à l'aide des commandes Docker CLI.

Voici quelques points clés concernant les images Docker :

Manipulation :

Pour manipuler les images Docker, vous utilisez les commandes Docker CLI comme docker build, docker pull, docker push, docker save, et docker load.

IMAGES DOCKER

Les images Docker sont des fichiers binaires qui n'ont pas d'extension spécifique.

Elles sont généralement stockées dans un format propriétaire de Docker et sont manipulées à l'aide des commandes Docker CLI.

Voici quelques points clés concernant les images Docker :

Bonnes Pratiques :

- Utiliser des images de base légères (comme alpine).
- Minimiser le nombre de couches en combinant les instructions RUN.
- Utiliser des instructions .dockerignore pour exclure des fichiers inutiles.

CONTENEUR DOCKER

- Qu'est-ce qu'un conteneur informatique ?

- Un conteneur regroupe le code d'une application ainsi que ses fichiers associés.
- Plus légers et portables que les machines virtuelles, les conteneurs réduisent les coûts et permettent aux développeurs et aux équipes techniques de déployer des applications dans divers environnements.
- Ils facilitent ainsi les transitions entre les phases de développement, de test et de production.

CONTENEUR DOCKER

- **Utilisations des conteneurs**

- « Lift and shift » des applications vers des architectures cloud modernes
- Optimisation de la prise en charge pour les architectures de microservices
- Prise en charge DevOps pour l'intégration et le déploiement continus (CI/CD)

CONTENEUR DOCKER

- **« Lift and shift » des applications vers des architectures cloud modernes**

- Certaines entreprises adoptent les conteneurs pour transférer leurs applications vers des environnements plus modernes.
- Bien que cette approche offre certains avantages de la virtualisation, elle ne permet pas d'exploiter pleinement le potentiel d'une architecture applicative modulaire basée sur des conteneurs.

CONTENEUR DOCKER

- **Optimisation de la prise en charge pour les architectures de microservices**

- Les applications et microservices distribués peuvent être isolés, déployés et mis à l'échelle plus efficacement grâce aux composants de base des conteneurs individuels.

CONTENEUR DOCKER

- **Prise en charge DevOps pour l'intégration et le déploiement continus (CI/CD)**

- La technologie des conteneurs simplifie la création, les tests et le déploiement à partir des images de conteneurs.

- Les conteneurs sont souvent utilisés pour exécuter un ou plusieurs processus similaires en arrière-plan, tels que les fonctions ETL ou les tâches planifiées.

PARTIE III