

Filière : 1^{ème} année cyber-security

Semestre : 6

Module : INFONUAGIQUE

Thème :

RAPPORT: LA VIRTUALISATION

Encadré par : **Pr A.AMAMOU**

Préparé par l'étudiante: **Hajar
baba ahmed**

SOMMAIRE

- 01** Introduction général
- 02** Généralité sur la virtualisation
- 03** Configuration de VMware vCenter Server
- 04** La sécurisation de vCenter
- 05** déploiement de Kubeflow en utilisant Docker et Kubernetes
- 06** Conclusion

01 INTRODUCTION

La virtualisation est devenue une technologie incontournable pour les entreprises souhaitant optimiser l'utilisation de leurs ressources informatiques et s'adapter à une demande croissante en puissance de calcul et en espace de stockage.

n effet, la virtualisation permet de diviser les ressources matérielles d'un seul serveur physique en plusieurs machines virtuelles indépendantes, chacune pouvant exécuter son propre système d'exploitation et ses propres applications. Cela évite d'avoir à déployer de multiples serveurs physiques sous-utilisés, ce qui serait coûteux et inefficace.

- La virtualisation apporte de nombreux avantages aux entreprises :
 - :Flexibilité : les ressources peuvent être allouées et réallouées dynamiquement en fonction des besoins
- Économies : meilleure utilisation du matériel existant, réduction des coûts d'infrastructure et d'exploitation
- Évolutivité : possibilité de faire évoluer rapidement les capacités en fonction de la charge de travail
- Disponibilité : les machines virtuelles peuvent être facilement déplacées, copiées ou sauvegardées
- Sécurité : isolation des applications et des systèmes d'exploitation, renforcement de la sécurité

La virtualisation est également la technologie clé qui permet le modèle du cloud computing. Elle permet aux fournisseurs de cloud d'exploiter efficacement leur infrastructure matérielle existante, et aux utilisateurs de n'acheter que les ressources dont ils ont besoin à un instant donné, en les faisant évoluer de manière rentable au fil du temps.

02 GÉNÉRALITÉ SUR LA VIRTUALISATION

N° 01 - Définition:

La virtualisation consiste à attribuer les caractéristiques d'une machine physique à une ou plusieurs machines virtuelles, de manière à faire fonctionner différents systèmes d'exploitation sur un seul et unique serveur. Cette technologie permet de créer des environnements isolés pour les systèmes d'exploitation et les applications, ce qui améliore l'efficacité, la flexibilité et la sécurité des ressources informatiques.

N° 02 - Types de Virtualisation :

- Virtualisation des postes de travail : permet d'exécuter plusieurs bureaux virtuels sur des machines virtuelles disponibles.
- Virtualisation des applications : abstraite la couche d'application du système d'exploitation, permettant des applications indépendantes du système d'exploitation.
- Virtualisation des serveurs : permet de diviser les ressources d'un seul serveur physique en plusieurs machines virtuelles.
- Virtualisation du stockage : permet de consolider les données de plusieurs sources dans une seule couche pour accéder facilement aux données sans détails sur la source ou l'emplacement.
- Virtualisation des réseaux : permet de créer des réseaux virtuels pour isoler les communications et améliorer la sécurité.
- Virtualisation des données : permet de consolider les données de plusieurs sources dans une seule couche pour accéder facilement aux données sans détails sur la source ou l'emplacement.

N° 03 - Plateformes de Virtualisation:

- VMware : une plateforme populaire pour la virtualisation des serveurs et des applications.
- KVM : un hyperviseur open-source pour la virtualisation des serveurs.
- Citrix XenServer : un hyperviseur pour la virtualisation des serveurs et des applications.
- Oracle VM VirtualBox : un hyperviseur pour la virtualisation des serveurs et des applications.
- Microsoft Hyper-V : un hyperviseur pour la virtualisation des serveurs et des applications.

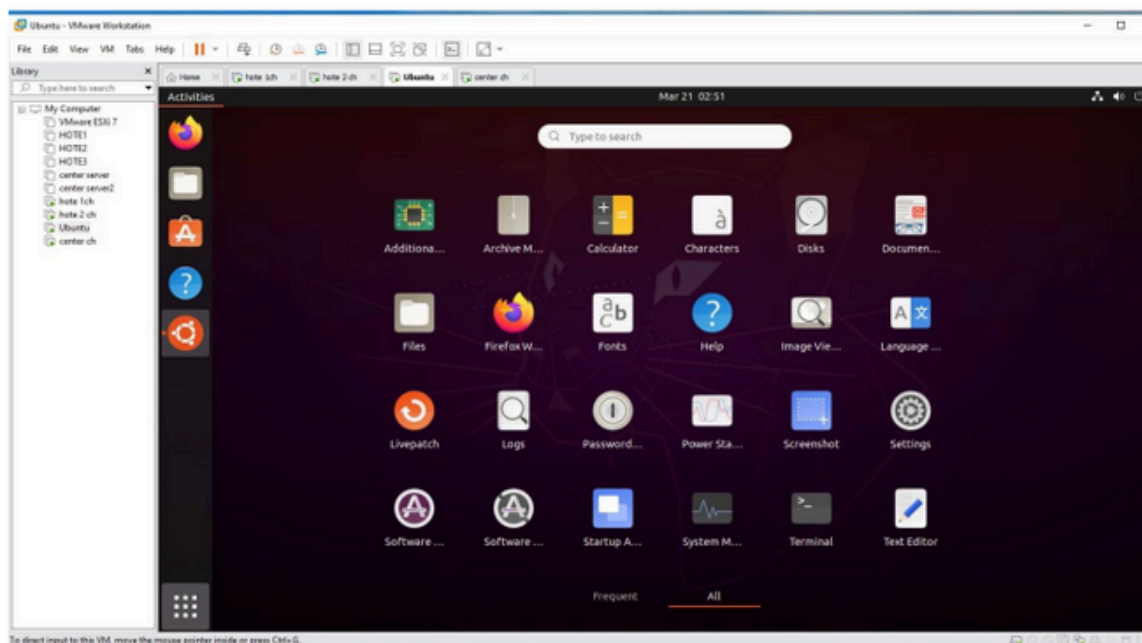
03 CONFIGURATION DE VMWARE VCENTER SERVER

D'abord, on a installé le logiciel VMware workstation Pro qui est une solution logicielle professionnelle, puissante et complète qui vous permettra de gérer l'ensemble de vos machines virtuelles locales ou sur le réseau. La solution ultime de virtualisation pour émuler et gérer plusieurs systèmes d'exploitation. :



1.-Configuration des serveurs DHCP et DNS:

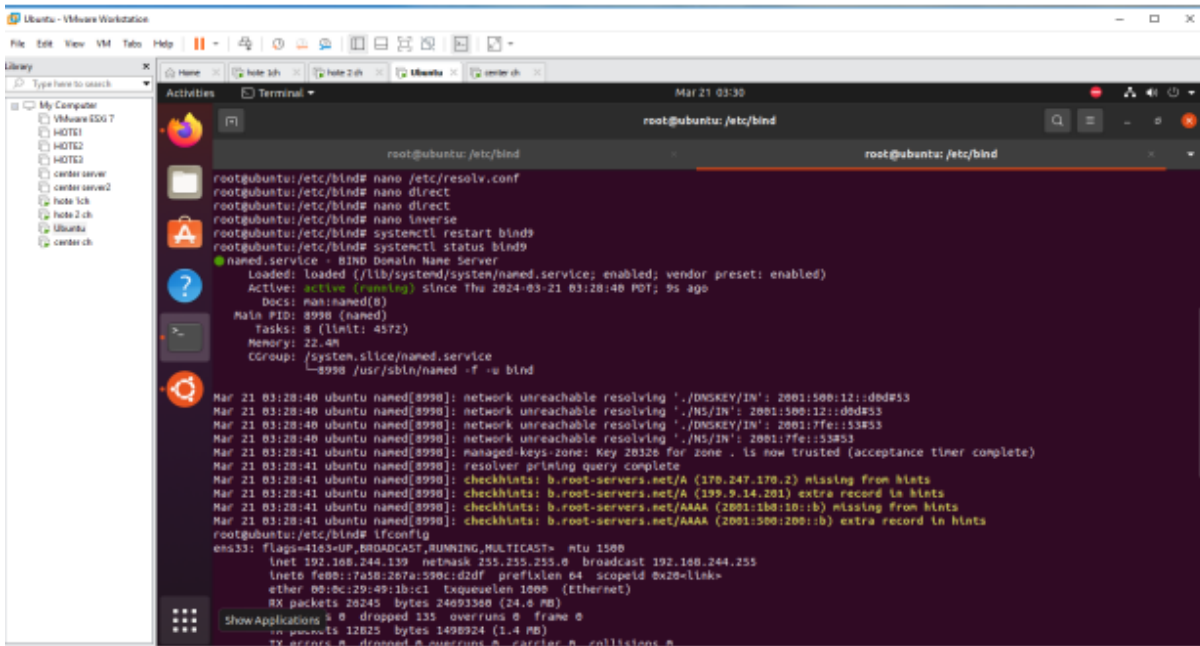
Pour configurer les serveurs DHCP et DNS, on utilise créer machine virtuelle Ubuntu en suivant les étapes nécessaires pour l'installation.



N° 01 - Serveur DNS :

Le DNS, ou Domain Name System (Système de Noms de Domaine), est un système hiérarchique et distribué qui permet de traduire les noms de domaine, tels que `www.example.com`, en adresses IP compréhensibles par les ordinateurs sur Internet et vice versa.

Pour notre cas on constate que le service est activé :

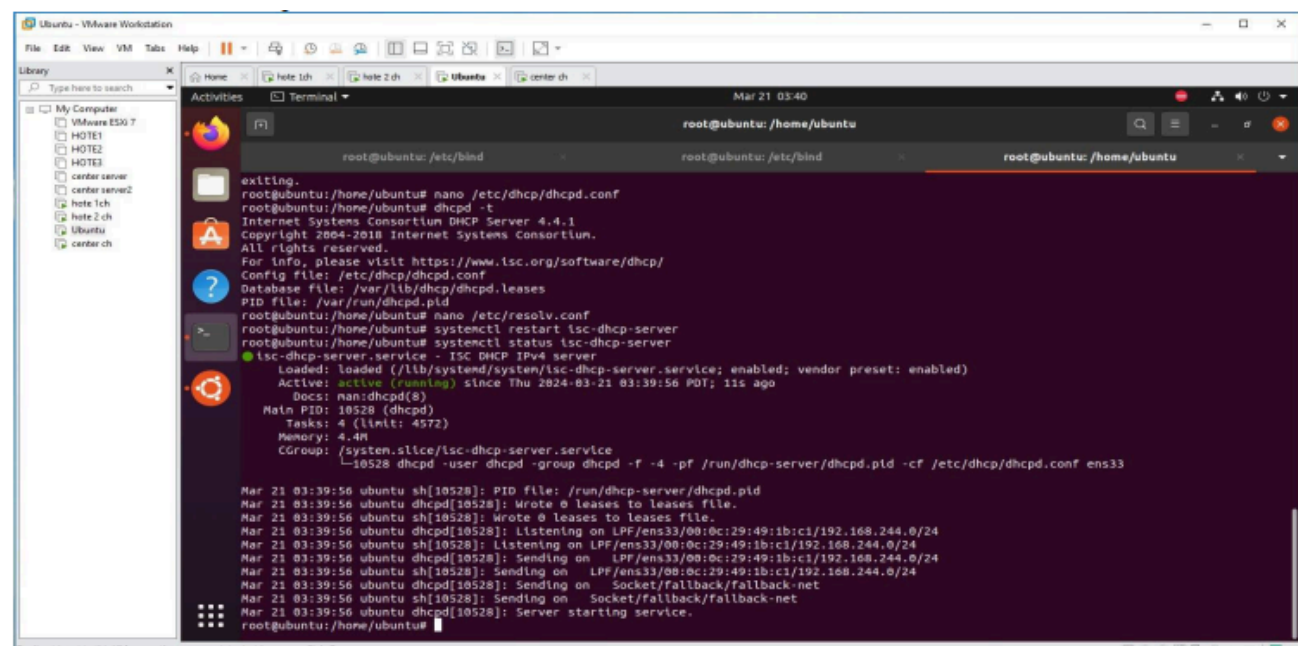


```
root@ubuntu: /etc/bind
root@ubuntu: /etc/bind# nano /etc/resolv.conf
root@ubuntu: /etc/bind# nano direct
root@ubuntu: /etc/bind# nano direct
root@ubuntu: /etc/bind# nano inverse
root@ubuntu: /etc/bind# systemctl restart bind9
root@ubuntu: /etc/bind# systemctl status bind9
● named.service - BIND Domain Name Server
Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2024-03-21 03:28:40 PDT; 9s ago
Docs: man:named(8)
Main PID: 8998 (named)
Tasks: 8 (limit: 4572)
Memory: 22.4M
CGroup: /system.slice/named.service
└─8998 /usr/sbin/named -f -u bind

Mar 21 03:28:40 ubuntu named[8998]: network unreachable resolving './DNSKEY/IN': 2001:500:12::d8d853
Mar 21 03:28:40 ubuntu named[8998]: network unreachable resolving './NS/IN': 2001:500:12::d8d853
Mar 21 03:28:40 ubuntu named[8998]: network unreachable resolving './DNSKEY/IN': 2001:7fe::53853
Mar 21 03:28:41 ubuntu named[8998]: managed-keys-zone: Key 20326 for zone . is now trusted (acceptance timer complete)
Mar 21 03:28:41 ubuntu named[8998]: resolver printing query complete
Mar 21 03:28:41 ubuntu named[8998]: checkhints: b.root-servers.net/A (170.247.170.2) missing from hints
Mar 21 03:28:41 ubuntu named[8998]: checkhints: b.root-servers.net/A (199.9.14.201) extra record in hints
Mar 21 03:28:41 ubuntu named[8998]: checkhints: b.root-servers.net/AAAA (2001:1b0:10::b) missing from hints
Mar 21 03:28:41 ubuntu named[8998]: checkhints: b.root-servers.net/AAAA (2001:500:200::b) extra record in hints
root@ubuntu: /etc/bind# ifconfig
ens33: flags=160<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.244.139 netmask 255.255.255.0 broadcast 192.168.244.255
    inet6 fe80::7a5b:267a:590c:d2df prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:49:1b:c1 txqueuelen 1000 (Ethernet)
    RX packets 20245 bytes 24693360 (24.6 MB)
    rx errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

N° 02 - Serveur DHCP:

: Le protocole DHCP (pour Dynamic Host Configuration Protocol) est un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres réseau d'une station, notamment en lui assignant automatiquement une adresse IP et un masque de sous-réseau...



```
root@ubuntu: /home/ubuntu
root@ubuntu: /etc/bind
root@ubuntu: /etc/bind
root@ubuntu: /home/ubuntu# nano /etc/dhcp/dhcpd.conf
root@ubuntu: /home/ubuntu# dhcpd -t
Internet Systems Consortium DHCP Server 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcp/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
root@ubuntu: /home/ubuntu# nano /etc/resolv.conf
root@ubuntu: /home/ubuntu# systemctl restart isc-dhcp-server
root@ubuntu: /home/ubuntu# systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2024-03-21 03:39:56 PDT; 11s ago
Docs: man:dhcpd(8)
Main PID: 10528 (dhcpd)
Tasks: 4 (limit: 4572)
Memory: 4.4M
CGroup: /system.slice/isc-dhcp-server.service
└─10528 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf ens33

Mar 21 03:39:56 ubuntu sh[10528]: PID file: /run/dhcp-server/dhcpd.pid
Mar 21 03:39:56 ubuntu dhcpd[10528]: Wrote 0 leases to leases file.
Mar 21 03:39:56 ubuntu sh[10528]: Wrote 0 leases to leases file.
Mar 21 03:39:56 ubuntu dhcpd[10528]: Listening on LPF/ens33/00:0c:29:49:1b:c1/192.168.244.0/24
Mar 21 03:39:56 ubuntu dhcpd[10528]: Listening on LPF/ens33/00:0c:29:49:1b:c1/192.168.244.0/24
Mar 21 03:39:56 ubuntu sh[10528]: Sending on LPF/ens33/00:0c:29:49:1b:c1/192.168.244.0/24
Mar 21 03:39:56 ubuntu sh[10528]: Sending on LPF/ens33/00:0c:29:49:1b:c1/192.168.244.0/24
Mar 21 03:39:56 ubuntu dhcpd[10528]: Sending on Socket/fallback/fallback-net
Mar 21 03:39:56 ubuntu dhcpd[10528]: Sending on Socket/fallback/fallback-net
Mar 21 03:39:56 ubuntu dhcpd[10528]: Server starting service.
root@ubuntu: /home/ubuntu#
```

2.Installation de vShper :

Un hyperviseur de type 1 est un système qui s'installe directement sur le matériel physique du serveur, sans passer par un système d'exploitation intermédiaire. Il devient le système d'exploitation principal de la machine et gère directement les ressources matérielles pour faire fonctionner les machines virtuelles. Les hyperviseurs de type 1 sont les plus performants et sécurisés, et sont généralement utilisés dans les environnements d'entreprise nécessitant des performances élevées et une gestion optimale des ressources. Quelques exemples d'hyperviseurs de type 1 sont VMware ESXi, Microsoft Hyper-V et Citrix XenServer.

Un hyperviseur de type 2 fonctionne quant à lui comme une application installée sur un système d'exploitation hôte préexistant. Il dépend donc du système d'exploitation sous-jacent et n'a pas un accès direct au matériel. Les hyperviseurs de type 2 sont généralement moins performants que ceux de type 1, mais sont plus faciles à installer et peuvent convenir à des usages spécifiques comme sur des postes de travail individuels. Des exemples d'hyperviseurs de type 2 sont VirtualBox, VMware Workstation et Parallels Desktop.

vSphere est une suite de logiciels de virtualisation développée par VMware. Elle est principalement utilisée pour la gestion et l'optimisation des ressources informatiques au sein d'un environnement virtualisé.

Principaux Composants de vSphere:

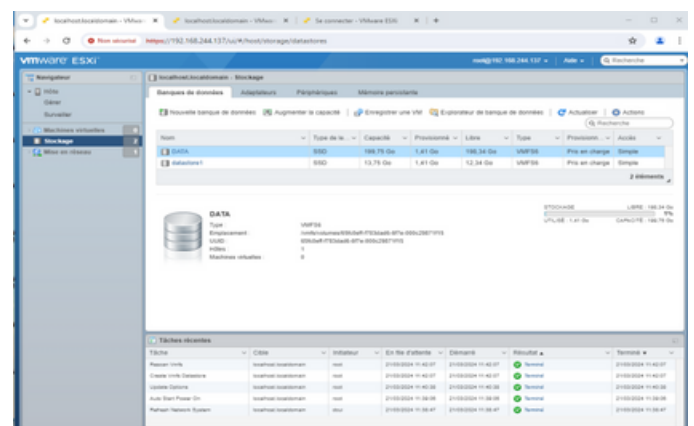
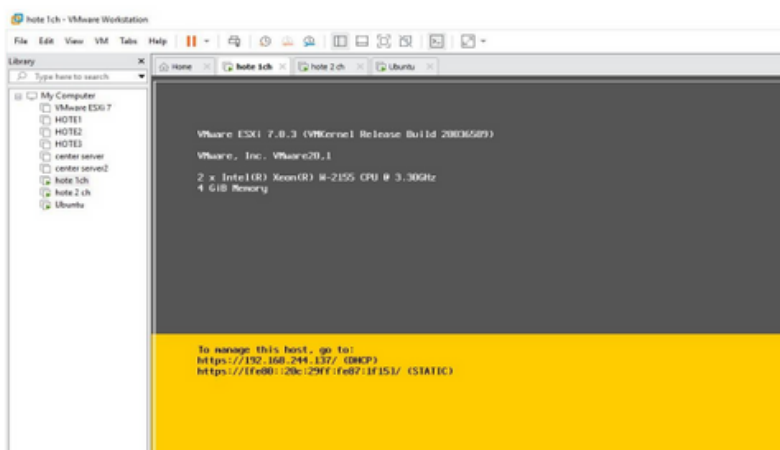
1. ESXi Hypervisor :

- ESXi est un hyperviseur bare-metal, ce qui signifie qu'il s'installe directement sur le matériel physique (serveurs) sans nécessiter un système d'exploitation sous-jacent. Il permet de créer et gérer des machines virtuelles (VM) sur le serveur physique.

2. vCenter Server :

- vCenter Server est une plateforme de gestion centralisée pour les environnements vSphere. Elle permet de gérer plusieurs hôtes ESXi et les machines virtuelles qu'ils contiennent, facilitant ainsi des tâches comme la configuration des clusters, la gestion des ressources et la planification des sauvegardes et des restaurations.

Pour créer une machine virtuelle ESXi, on utilise un environnement de virtualisation comme VMware Workstation. Après avoir créé la machine VM ESXI, on arrive aux configurations des spécifications matérielles où on doit allouer au moins 2 processeurs, 4 GB de RAM et un disque virtuel d'au moins 40 GB. Après la terminaison des autres processus de création, on obtient une @IP avec laquelle on se connecte à notre VMware ESXI via le navigateur. Dans le VM ESXI on peut créer des machines virtuelles, ajouter des banques de donnée...



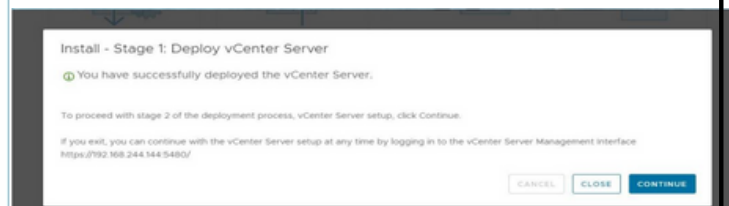
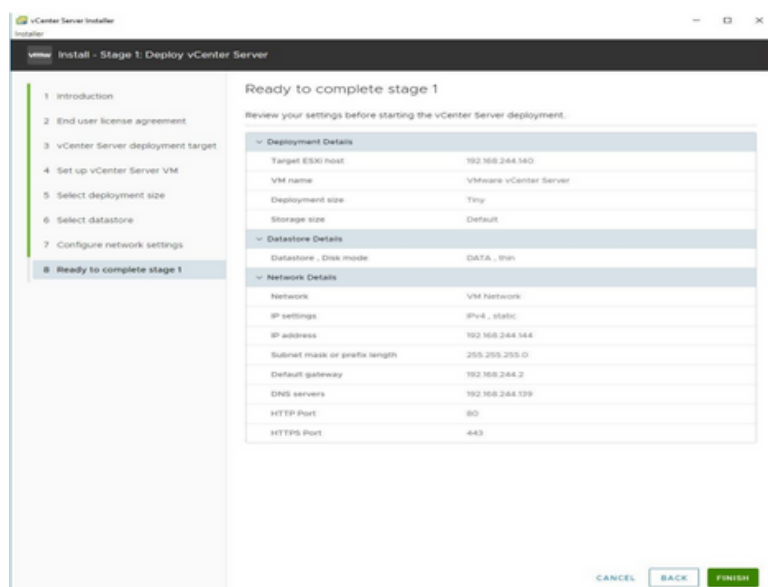
3.Installation de vCenter :

vCenter Server: est l'outil central de gestion et d'orchestration de VMware vSphere. Il permet d'administrer de manière unifiée et centralisée les différents hôtes ESXi et les machines virtuelles qui y sont hébergées.

Grâce à vCenter Server, les administrateurs peuvent effectuer de nombreuses tâches avancées qui ne sont pas disponibles directement sur les hôtes ESXi, comme :

- Créer et gérer des clusters de haute disponibilité et de tolérance de pannes
- Configurer l'équilibrage de charge dynamique des ressources (DRS)
- Déployer et provisionner des machines virtuelles de manière automatisée
- Appliquer des stratégies de sécurité et de conformité à grande échelle
- Surveiller les performances et générer des rapports détaillés sur l'utilisation des ressources

Configurer VMware vCenter Server implique plusieurs étapes, allant de l'installation à la configuration des hôtes et des machines virtuelles. Voici un guide détaillé pour configurer vCenter Server, en utilisant principalement vCenter Server Appliance (VCSA) pour son déploiement. Pour commencer, téléchargez l'image ISO de vCenter Server Appliance depuis le site de VMware et montez-la sur une machine locale. Ensuite, exécutez le programme d'installation et sélectionnez "Install" pour commencer l'installation. Configurez les paramètres de déploiement, tels que la connexion à l'hôte ESXi, le certificat SSL, le nom de la VM et le mot de passe root. Sélectionnez la taille de déploiement appropriée en fonction du nombre d'hôtes et de machines virtuelles que vCenter Server générera, ainsi que la taille de stockage et les paramètres réseau. Après le déploiement, accédez à l'interface de configuration via un navigateur web à l'adresse [https://\[IP_vCenter\]:5480](https://[IP_vCenter]:5480). Configurez le SSO (Single Sign-On) en définissant le domaine SSO et en créant un compte administrateur SSO. Configurez également les paramètres de base, tels que les paramètres de temps (NTP) et les paramètres réseau supplémentaires si nécessaire. Enfin, vérifiez les configurations et cliquez sur "Finish" pour terminer la configuration. Pour ajouter des hôtes ESXi au cluster, cliquez avec le bouton droit sur le cluster et sélectionnez "Ajouter un hôte". Entrez l'IP ou le FQDN de l'hôte, puis le nom d'utilisateur root et le mot de passe. Activez la reprise sur incident et vSphere HA en cochant les cases. Configurez également le coredump sur les hôtes ESXi en démarrant le service VMware vSphere ESXi Dump Collector et en exécutant des commandes SSH pour configurer le réseau de coredump

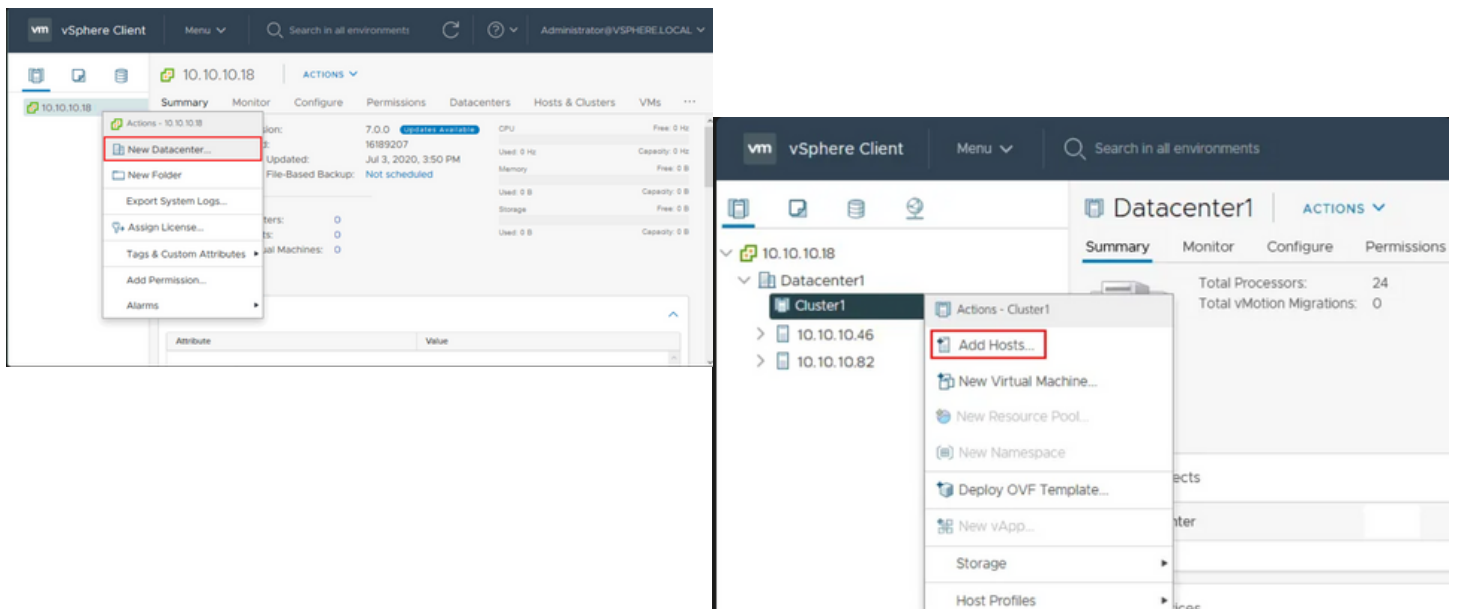


4. Configuration de Cluster:

Configurer un cluster dans VMware vCenter Server permet de regrouper plusieurs hôtes ESXi pour bénéficier de fonctionnalités avancées telles que la haute disponibilité et la répartition des ressources .

1. Connectez-vous à l'interface web de vCenter Server à l'adresse [https://\[IP_vCenter\]/ui](https://[IP_vCenter]/ui) en utilisant les identifiants administratifs.
2. Créez un nouveau centre de données dans lequel vous placerez le cluster. Faites un clic droit sur le centre de données et sélectionnez "Nouveau Cluster". Donnez un nom au cluster, activez DRS (Distributed Resource Scheduler) et HA (High Availability), et configurez vSAN si nécessaire. Cliquez sur "OK" pour créer le cluster.
3. Ajoutez les hôtes ESXi au cluster nouvellement créé. Sélectionnez le cluster, cliquez sur "Ajouter Hôte", entrez l'adresse IP ou le FQDN de l'hôte ainsi que les identifiants root, puis suivez l'assistant pour ajouter chaque hôte au cluster. Répétez cette procédure pour tous les hôtes à ajouter.

Une fois les hôtes ajoutés, leurs ressources deviennent disponibles pour le cluster qui les gère. Vous pouvez alors configurer davantage le cluster en activant les fonctionnalités HA et DRS pour améliorer la tolérance aux pannes et la répartition des ressources entre les hôtes ESXi



04 SÉCURISATION DE VCENTER SERVER

N° 01 - Importance de la sécurité de vmware vcenter :

La sécurité de VMware vCenter est d'une importance capitale pour la protection de votre infrastructure virtuelle. VMware publie des guides et des recommandations pour assurer la sécurité de vCenter Server et des ESXi hosts. Ces guides incluent des pratiques de sécurité pour les différents composants de votre infrastructure vSphere, tels que la sécurisation d'ESXi Hosts, la sécurisation de vCenter Server Systems, la sécurité des machines virtuelles, et la sécurité du réseau vSphere.

N° 03 - Sécurisation de Vcenter server

Le système vCenter Server et les services associés sont protégés par l'authentification via vCenter Single Sign-On, ainsi que par l'autorisation via le modèle d'autorisations vCenter Server.

- Renforcer toutes les machines hôtes vCenter : Pour protéger votre environnement vCenter, vous devez commencer par renforcer chaque machine qui exécute vCenter Server ou un service associé. Ceci s'applique aussi bien à une machine physique qu'à une machine virtuelle. Installez toujours les derniers correctifs de sécurité pour votre système d'exploitation et mettez en œuvre les meilleures pratiques standard de l'industrie pour protéger la machine hôte.

- Configurer vCenter Single Sign-On: vCenter Server et les services associés sont protégés par la structure d'authentification vCenter Single Sign-On. Lors de la première installation des logiciels, vous devez spécifier un mot de passe pour l'administrateur du domaine vCenter Single Sign On (par défaut, administrator@vsphere.local). Seul ce domaine est disponible initialement comme source d'identité. Vous pouvez ajouter d'autres sources d'identité (Active Directory ou LDAP) et définir une source d'identité par défaut. Dorénavant, les utilisateurs qui peuvent s'authentifier auprès d'une de ces sources d'identité ont la possibilité d'afficher des objets et d'effectuer des tâches, dans la mesure où ils y ont été autorisés.

- Attribuer des rôles à des utilisateurs ou groupes nommés: Pour optimiser la journalisation, chaque autorisation octroyée pour un objet peut être associée à un utilisateur ou groupe nommé, ainsi qu'à un rôle prédéfini ou personnalisé. Le modèle d'autorisations vSphere 6.0 procure une grande flexibilité en offrant la possibilité d'autoriser les utilisateurs et les groupes de diverses façons.

- Configurer NTP: Configurez NTP pour chaque nœud de votre environnement. L'infrastructure de certificats exige un horodatage précis et ne fonctionne correctement que si les nœuds sont synchronisés.

05 DÉPLOIEMENT DE KUBEFLOW EN UTILISANT DOCKER ET KUBERNETES

1.Docker:

Docker est une plateforme de conteneurisation d'applications informatiques qui permet de créer, déployer et gérer des applications virtualisées sur un système d'exploitation. Les principaux éléments de Docker sont :

- Le Docker Engine : l'application à installer sur la machine hôte pour créer, exécuter et gérer les conteneurs Docker. C'est le moteur du système Docker.
- Le Docker Daemon : traite les requêtes API pour gérer les différents aspects de l'installation comme les images, les conteneurs ou les volumes.
- Le Docker Client : l'interface principale pour communiquer avec le système Docker et transmettre les commandes au Docker Daemon.
- Docker Compose : un outil en ligne de commande qui permet d'assembler des applications à partir de conteneurs multiples.

Le rôle principal de Docker est de permettre aux développeurs de créer, tester et déployer rapidement des applications sous forme de conteneurs. Les conteneurs regroupent tout le code, les dépendances et les bibliothèques nécessaires pour exécuter une application, ce qui la rend portable et reproductible dans différents environnements.

Docker facilite ainsi le développement, l'intégration continue et le déploiement des applications, en les isolant dans des conteneurs standardisés. C'est devenu une technologie très populaire et largement adoptée dans l'industrie du développement logiciel.

2.Kubernetes:

Kubernetes est un système open source développé par Google pour automatiser le déploiement, la mise à l'échelle et la gestion des applications conteneurisées. Il fournit une plateforme pour orchestrer et coordonner les conteneurs sur un cluster de machines virtuelles ou physiques.

Kubernetes offre de nombreuses fonctionnalités avancées, notamment :

- Orchestration du stockage : Kubernetes monte automatiquement le système de stockage choisi, local, cloud ou réseau.
- Auto-guérison : Kubernetes redémarre les conteneurs défectueux, remplace et reprogramme les conteneurs en cas de panne de nœud.
- Gestion des secrets et de la configuration : Déploie et met à jour les secrets et la configuration sans reconstruire l'image.
- Placement automatique : Place les conteneurs en fonction de leurs ressources et contraintes, sans sacrifier la disponibilité.
- Mise à l'échelle horizontale : Monte ou descend en charge simplement, manuellement ou automatiquement.
- Extensibilité : Ajoute des fonctionnalités sans changer le code amont.

3.Kubeflow:

Kubeflow est une plateforme open source conçue pour simplifier le déploiement, la gestion et la mise à l'échelle de workflows ML sur des clusters Kubernetes. Il fournit un ensemble de composants pour construire, entraîner, déployer et gérer des modèles ML de manière portable et scalable.

Les principaux composants de Kubeflow sont:

- Kubeflow Pipelines pour créer et déployer des workflows ML portables et scalables
- Notebooks pour exécuter des environnements de développement web sur le cluster
- Un tableau de bord central
- Katib pour l'AutoML et le réglage de hyperparamètres
- Des opérateurs d'entraînement pour les frameworks ML populaires
- KServe pour servir les modèles en production

Kubeflow permet aux équipes de data science et MLOps de travailler de manière collaborative et reproductible, en automatisant les tâches courantes comme la gestion des données, l'entraînement et le déploiement.

Ses avantages clés sont la rapidité de déploiement, la portabilité sur tout Kubernetes, la gestion des ressources par Kubernetes en arrière-plan, et un espace collaboratif pour partager pipelines et modèles.

Cependant, Kubeflow comporte de nombreuses briques logicielles interdépendantes avec des configurations complexes à maîtriser dans leur ensemble. Il ne faut pas sous-estimer cette complexité sous-jacente.

En résumé, Kubeflow est une boîte à outils très complète pour faire du machine learning sur Kubernetes de manière productive et collaborative, malgré une certaine complexité technique. Son adoption dépendra des besoins spécifiques du projet et de l'équipe.

4.nœud Master:

Le nœud master, également appelé nœud de contrôle, est un élément crucial dans l'architecture de Kubernetes. Il est responsable de la gestion et de la coordination du cluster, en orchestrant les différentes tâches et en s'assurant que l'état souhaité du cluster est maintenu.

les composants principaux du nœud master :

1. API Server : Le serveur d'API est l'interface principale du cluster Kubernetes. Il expose l'API Kubernetes et traite les requêtes provenant des utilisateurs, des outils CLI (kubectl) et des autres composants du cluster.
2. etcd : etcd est un magasin de clés-valeurs distribué et fiable qui stocke toutes les données de configuration du cluster. Cela inclut les informations sur l'état du cluster et les définitions des objets Kubernetes.
3. Controller Manager : Ce composant exécute des contrôleurs qui gèrent les routines de contrôle de l'état des ressources Kubernetes. Par exemple, il s'assure que le nombre de répliques d'un déploiement correspond au nombre souhaité.
4. Scheduler : Le planificateur décide sur quel nœud worker chaque pod doit être exécuté en fonction des ressources disponibles et des contraintes spécifiées.

Un cluster Kubernetes peut avoir un ou plusieurs nœuds master pour assurer la haute disponibilité et la redondance. Cela permet de maintenir la disponibilité du cluster même en cas de panne d'un nœud master individuel.

5.nœud Worker :

Les nœuds worker, ou nœuds de travail, sont responsables de l'exécution des charges de travail, c'est-à-dire des pods. Chaque nœud worker contient les composants suivants :

- Kubelet : Le kubelet est un agent qui tourne sur chaque nœud worker. Il reçoit les spécifications des pods depuis l'API Server et s'assure que les conteneurs sont correctement lancés et exécutés sur le nœud.
- Kube-proxy : Kube-proxy est un réseau proxy qui gère le routage du trafic réseau vers les pods. Il assure le service de réseau et le proxying des requêtes réseau pour les services Kubernetes.
- Container Runtime : Le runtime de conteneur est le logiciel qui exécute les conteneurs. Docker est l'un des runtimes de conteneur les plus utilisés, mais Kubernetes supporte également d'autres runtimes comme containerd et CRI-O.

les nœuds worker sont les machines qui exécutent réellement les applications conteneurisées dans des pods. Le kubelet et kube-proxy permettent de communiquer avec le master et de gérer le cycle de vie des conteneurs, tandis que le runtime de conteneur (Docker, containerd, etc.) est responsable de l'exécution effective des conteneurs.

5. Interaction entre les Nœuds Master et Worker :

L'interaction entre les nœuds master et worker dans Kubernetes est essentielle pour le déploiement, la planification et l'exécution des applications.

Déploiement

- **Traitement de la demande :** Lorsqu'un utilisateur soumet une demande de déploiement d'une application via `kubectl`, cette demande est traitée par l'API Server sur le nœud master.

Planification

- **Assignation des nœuds :** Le Scheduler du nœud master décide sur quel nœud worker les pods de l'application seront déployés en fonction des ressources disponibles et des contraintes.

Exécution

- **Lancement des conteneurs :** Le kubelet sur chaque nœud worker reçoit les spécifications des pods et utilise le container runtime (par exemple, Docker) pour lancer les conteneurs.

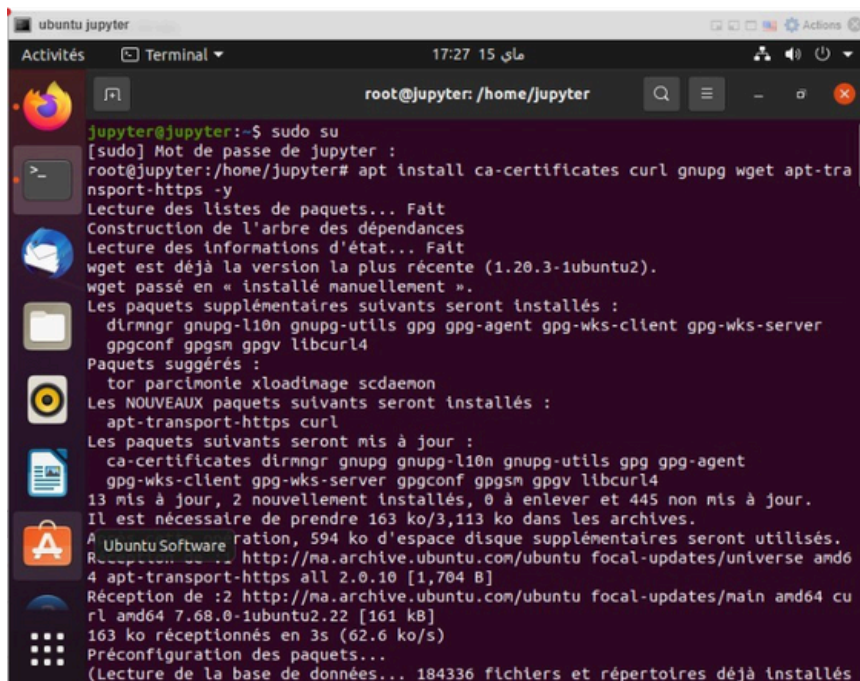
Réseau et Service

- **Routage du trafic :** Kube-proxy sur chaque nœud worker assure le routage du trafic réseau vers les pods, permettant ainsi la communication entre les pods et les entités extérieures au cluster.

l'interaction entre les nœuds master et worker implique une coordination étroite pour déployer, planifier et exécuter les applications dans un cluster Kubernetes.

6. configuration de Docker, Kubernetes, et déploiement de Kubeflow :

6.1. Installation et Configuration de Docker :



```
ubuntu jupyter
Activités Terminal 17:27 15 ماي
root@jupyter: /home/jupyter
jupyter@jupyter:~$ sudo su
[sudo] Mot de passe de jupyter :
root@jupyter:/home/jupyter# apt install ca-certificates curl gnupg wget apt-tr
nsport-https -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
wget est déjà la version la plus récente (1.20.3-1ubuntu2).
wget passé en « installé manuellement ».
Les paquets supplémentaires suivants seront installés :
  dirnmgr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server
  gpgconf gpgsm gpgv libcurl4
Paquets suggérés :
  tor parcimonie xloadimage sdcdaemon
Les NOUVEAUX paquets suivants seront installés :
  apt-transport-https curl
Les paquets suivants seront mis à jour :
  ca-certificates dirnmgr gnupg gnupg-l10n gnupg-utils gpg gpg-agent
  gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv libcurl4
13 mis à jour, 2 nouvellement installés, 0 à enlever et 445 non mis à jour.
Il est nécessaire de prendre 163 ko/3,113 ko dans les archives.
A UbuntuSoftware ration, 594 ko d'espace disque supplémentaires seront utilisés.
Récupération de : http://ma.archive.ubuntu.com/ubuntu focal-updates/universe amd6
4 apt-transport-https all 2.0.10 [1,704 B]
Réception de :2 http://ma.archive.ubuntu.com/ubuntu focal-updates/main amd64 cu
rl amd64 7.68.0-1ubuntu2.22 [161 kB]
163 ko réceptionnés en 3s (62.6 ko/s)
Préconfiguration des paquets...
(Lecture de la base de données... 184336 fichiers et répertoires déjà installés)
```

La commande "`apt install ca-certificates curl gnupg wget apt-transport-https -y`" pour installer plusieurs paquets importants avant l'installation de Docker et Kubernetes:

- ca-certificates : Gère les certificats de sécurité utilisés par les sites web et autres services.
- curl : Outil en ligne de commande pour transférer des données avec des protocoles de réseau.
- gnupg : Outil de cryptographie pour signer et vérifier des signatures ainsi que pour chiffrer et déchiffrer des textes.
- wget : Outil en ligne de commande pour télécharger des fichiers depuis le web.
- apt-transport-https : Permet à apt de récupérer des paquets via le protocole HTTPS.

les commandes exécutées dans cette étape pour ajouter le dépôt Docker, installer Docker et mettre à jour les paquets.

Après avoir installé Docker, on peut vérifier que Docker est correctement installé et fonctionne

```
ubuntu jupyter
17:27 15 مای
root@jupyter: /home/jupyter
root@jupyter: /home/jupyter# install -m 0755 -d /etc/apt/keyrings
root@jupyter: /home/jupyter# curl -fsSL https://download.docker.com/linux/ubuntu
/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
root@jupyter: /home/jupyter# chmod a+r /etc/apt/keyrings/docker.gpg
root@jupyter: /home/jupyter# echo \
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docke
r.gpg] https://download.docker.com/linux/ubuntu \
> "$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
root@jupyter: /home/jupyter# apt update
Réception de :1 https://download.docker.com/linux/ubuntu focal InRelease [57.7
kB]
Atteint :2 http://ma.archive.ubuntu.com/ubuntu focal InRelease
Atteint :3 http://security.ubuntu.com/ubuntu focal-security InRelease
Atteint :4 http://ma.archive.ubuntu.com/ubuntu focal-updates InRelease
Atteint :5 http://ma.archive.ubuntu.com/ubuntu focal-backports InRelease
Réception de :6 https://download.docker.com/linux/ubuntu focal/stable amd64 Pac
kages [43.0 kB]
101 ko réceptionnés en 3s (33.0 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
445 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour le
s voir.
root@jupyter: /home/jupyter# apt install -y docker-ce docker-ce-cli containerd
io docker-buildx-plugin docker-compose-plugin
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
```

```
machine2
Modules complémentaires chargés : fastestmirror, product-id, search-disabled-repos, subscription-
manager
This system is not registered with an entitlement server. You can use subscription-manager to regist
er.
Loading mirror speeds from cached hostfile
* base: mirror.marwan.ma
* extras: mirror.marwan.ma
* updates: mirror.marwan.ma
Le paquet 2:docker-1.13.1-210.git7d71120.el7.centos.x86_64 est déjà installé dans sa dernière versio
n
Rien à faire
[machine@localhost ~]$ sudo systemctl start docker
[machine@localhost ~]$ sudo systemctl enable docker
[machine@localhost ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since ven. 2024-05-03 18:59:39 CEST; 7min ago
     Docs: http://docs.docker.com
   Main PID: 1199 (dockerd-current)
   CGroup: /system.slice/docker.service
           └─1199 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/libexec/docker/docker-r...
             └─1346 /usr/bin/docker-containerd-current -l unix:///var/run/docker/libcontainerd/dock...
mai 03 18:59:37 localhost.localdomain dockerd-current[1199]: time="2024-05-03T18:59:37.903662164..."
mai 03 18:59:39 localhost.localdomain dockerd-current[1199]: time="2024-05-03T18:59:39.806907912..."
mai 03 18:59:39 localhost.localdomain dockerd-current[1199]: time="2024-05-03T18:59:39.891335426..."
mai 03 18:59:39 localhost.localdomain dockerd-current[1199]: time="2024-05-03T18:59:39.126834281..."
mai 03 18:59:39 localhost.localdomain dockerd-current[1199]: time="2024-05-03T18:59:39.536847758..."
mai 03 18:59:39 localhost.localdomain dockerd-current[1199]: time="2024-05-03T18:59:39.646118654..."
mai 03 18:59:39 localhost.localdomain dockerd-current[1199]: time="2024-05-03T18:59:39.890626307..."
mai 03 18:59:39 localhost.localdomain dockerd-current[1199]: time="2024-05-03T18:59:39.899655381..."
mai 03 18:59:39 localhost.localdomain dockerd-current[1199]: time="2024-05-03T18:59:39.913978744..."
mai 03 18:59:39 localhost.localdomain systemd[1]: Started Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.
[machine@localhost ~]$
```

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```


6.2.Installation et Configuration de Kubectl :

```
sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
sudo swapoff -a
```

```
curl -LO https://dl.k8s.io/release/v1.21.7/bin/linux/amd64/kubectl
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
kubectl version --client
```

Après avoir terminé la configuration et l'installation de kubectl, on doit vérifier la version pour assurer qu'il a été installé avec succès.

```
seokii@seokii:~$ kubectl version --client
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.7", GitCommit:"1f86e54e8bfc86bc90b61c98f84d4", GitTreeState:"clean", BuildDate:"2021-11-17T14:41:19Z", GoVersion:"go1.16.10", Compiler:"gc", Platform:"linux/amd64"}
```

6.3.Installation et Configuration de Kubernetes:

```
curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.14.0/kind-linux-amd64
chmod +x ./kind
sudo mv ./kind /usr/local/bin/kind
```

Avant de configurer le cluster kubernetes, on doit installer kind, kind est un outil qui permet d'exécuter un cluster kubernetes, on l'installe via les commandes ci-dessus.

Après l'installation de kind, on peut créer un cluster on utilise kind:

```
seokii@seokii:~$ kind create cluster
Creating cluster "kind" ...
 ✓ Ensuring node image (kindest/node:v1.24.0)
 ✓ Preparing nodes
 ✓ Writing configuration
 ✓ Starting control-plane
 ✓ Installing CNI
 ✓ Installing StorageClass
Set kubectl context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Not sure what to do next? Check out https://kind.sigs.k8s.io/docs/user/quick-start/
```

```
seokii@seokii:~$ kubectl get nodes
NAME                 STATUS    ROLES    AGE   VERSION
kind-control-plane   Ready    control-plane   44s   v1.24.0
```

6.4.Installation des modules de base:

Avant de configurer et installer kubeflow, on doit avoir des modules de base tels que Helm et Kustomize et pour les installer, on suit les commandes suivantes:

```
wget https://get.helm.sh/helm-v3.7.1-linux-amd64.tar.gz
tar -zxvf helm-v3.7.1-linux-amd64.tar.gz
sudo mv linux-amd64/helm /usr/local/bin/helm

helm help

wget https://github.com/kubernetes-sigs/kustomize/releases/download/kustomize%2Fv3.10.0/kustomize_v3.10.0_linux_amd64.tar.gz
tar -zxvf kustomize_v3.10.0_linux_amd64.tar.gz
sudo mv kustomize /usr/local/bin/kustomize

kustomize help
```

6.5.Installation de Kubeflow:

On peut installer kubeflow via kustomize. On doit télécharger les fichiers kubeflow nécessaires dans git et les déplacer vers le dossier

```
git clone -b v1.4.0 https://github.com/kubeflow/manifests.git
cd manifests

while ! kustomize build example | kubectl apply -f -; do echo "Retrying to apply resources";
```

6.6.Gestionnaire de certificats:

```
kustomize build common/cert-manager/cert-manager/base | kubectl apply -f -
```

```
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-orders created
clusterrole.rbac.authorization.k8s.io/cert-manager-edit created
clusterrole.rbac.authorization.k8s.io/cert-manager-view created
clusterrole.rbac.authorization.k8s.io/cert-manager-webhook:subjectaccessreviews created
rolebinding.rbac.authorization.k8s.io/cert-manager-webhook:dynamic-serving created
rolebinding.rbac.authorization.k8s.io/cert-manager-cainjector:leaderelection created
rolebinding.rbac.authorization.k8s.io/cert-manager:leaderelection created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-cainjector created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-approve:cert-manager-io created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-certificates created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-challenges created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-clusterissuers created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-ingress-shim created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-issuers created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-orders created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-webhook:subjectaccessreviews created
service/cert-manager created
service/cert-manager-webhook created
deployment.apps/cert-manager created
deployment.apps/cert-manager-cainjector created
deployment.apps/cert-manager-webhook created
mutatingwebhookconfiguration.admissionregistration.k8s.io/cert-manager-webhook created
validatingwebhookconfiguration.admissionregistration.k8s.io/cert-manager-webhook created
seokii@seokii:~/manifests$
```

```
kubectl get pod -n cert-manager
```

```
seokii@seokii:~/manifests$ kubectl get pod -n cert-manager
```

NAME	READY	STATUS	RESTARTS	AGE
cert-manager-7dd5854bb4-q2qbm	1/1	Running	0	3m
cert-manager-cainjector-64c949654c-pqxsr	1/1	Running	0	3m
cert-manager-webhook-6b57b9b886-8sc54	1/1	Running	0	3m

```
seokii@seokii:~/manifests$ kustomize build common/cert-manager/kubeflow-issuer/base | kubectl apply -f -
clusterissuer.cert-manager.io/kubeflow-self-signing-issuer created
seokii@seokii:~/manifests$
```

6.7. Installation de Istio:

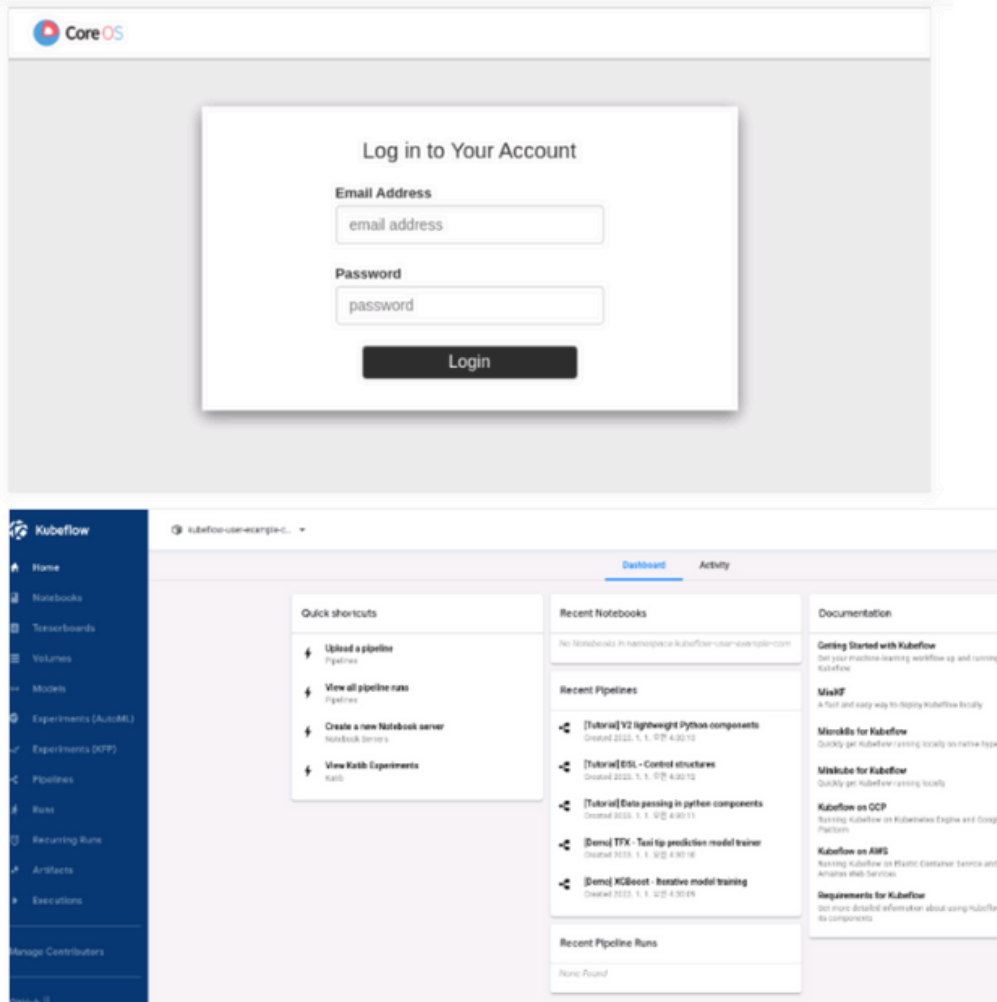
```
kustomize build common/istio-1-9/istio-crds/base | kubectl apply -f -
kustomize build common/istio-1-9/istio-namespace/base | kubectl apply -f -

seokii@seokii:~/manifests$ kustomize build common/cert-manager/kubeflow-issuer/base | kubectl apply -f -
clusterissuer.cert-manager.io/kubeflow-self-signing-issuer created
seokii@seokii:~/manifests$ kustomize build common/istio-1-9/istio-crds/base | kubectl apply -f -
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/authorizationpolicies.security.istio.io created
customresourcedefinition.apiextensions.k8s.io/destinationrules.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/envoyfilters.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/gateways.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/istiooperators.install.istio.io created
customresourcedefinition.apiextensions.k8s.io/peerauthentications.security.istio.io created
customresourcedefinition.apiextensions.k8s.io/requestauthentications.security.istio.io created
customresourcedefinition.apiextensions.k8s.io/serviceentries.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/sidecars.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/virtualservices.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/workloadentries.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/workloadgroups.networking.istio.io created
seokii@seokii:~/manifests$ kustomize build common/istio-1-9/istio-namespace/base | kubectl apply -f -
namespace/istio-system created

validatingwebhookconfiguration.admissionregistration.k8s.io/istiod-istio-system created
seokii@seokii:~/manifests$ kubectl get po -n istio-system
NAME                                READY   STATUS             RESTARTS   AGE
istio-ingressgateway-79b665c95-9cxs9 0/1     ContainerCreating   0           23s
istiod-86457659bb-g4z57              1/1     Running            0           23s
seokii@seokii:~/manifests$ kubectl get po -n istio-system
NAME                                READY   STATUS    RESTARTS   AGE
istiod-86457659bb-g4z57            1/1     Running   0           66s
istio-ingressgateway-79b665c95-9cxs9 1/1     Running   0           66s
seokii@seokii:~/manifests$
```

6.7. Confirmation finale:

```
kubectl port-forward svc/istio-ingressgateway -n istio-system 8080:80
```



06 CONCLUSION

La virtualisation est une technologie essentielle qui permet de créer des représentations logicielles de ressources informatiques physiques comme des serveurs, du stockage ou des réseaux. Elle dématérialise le matériel sous-jacent en machines virtuelles qui fonctionnent de manière isolée les unes des autres.

La virtualisation apporte de nombreux avantages aux entreprises et aux organisations. Tout d'abord, elle permet une allocation plus claire et une meilleure utilisation des ressources informatiques. Les machines virtuelles étant cloisonnées, cela offre un environnement plus sécurisé et robuste, avec un isolement des charges de travail. La virtualisation facilite également la gestion et la maintenance des infrastructures, en simplifiant les déploiements, les mises à jour et les sauvegardes.

Sur le plan opérationnel, la virtualisation se traduit par une consolidation des serveurs physiques, ce qui réduit les coûts en termes d'espace, d'alimentation et de refroidissement dans les datacenters. Elle permet aussi de tester de nouvelles applications dans des environnements isolés avant leur déploiement en production.

La virtualisation peut également servir à émuler du matériel obsolète ou plus ancien, offrant ainsi une approche stratégique de la gestion des ressources. Elle s'oriente de plus en plus vers une abstraction et une spécialisation accrue des machines virtuelles, avec l'émergence de technologies comme les conteneurs. Bien que la virtualisation soit souvent associée au cloud computing, les deux concepts restent distincts. Le cloud computing est une infrastructure complète qui fournit des services informatiques en ligne, tandis que la virtualisation est une technologie qui peut être utilisée dans un environnement cloud mais aussi en dehors.

En résumé, la virtualisation est une innovation technologique majeure qui transforme profondément la gestion des infrastructures informatiques. Elle apporte de nombreux bénéfices en termes d'efficacité, de flexibilité et de sécurité, et continue d'évoluer pour s'adapter aux besoins toujours plus complexes des entreprises et des organisations.