Voici les étapes pour créer et pousser les images taguées de votre service Web et de votre service DB sur votre compte Docker Hub :

1. **Connectez-vous à Docker Hub en utilisant la commande docker login. Entrez votre nom d'utilisateur et votre mot de passe lorsque vous êtes invité.**

docker login

1. **Allez dans le répertoire contenant votre fichier docker-compose.yml.**
2. **Construisez l'image taguée pour le service Web en utilisant la commande** docker build**. Assurez-vous de donner un nom et un tag approprié à l'image (dans cet exemple, j'utilise "kefifaiez/webapp:latest" comme nom et tag d'image).**

docker build -t kefifaiez/webapp:latest .

1. **Construisez l'image taguée pour le service DB en utilisant la commande** docker build**. Assurez-vous de donner un nom et un tag approprié à l'image (dans cet exemple, j'utilise "kefifaiez/dbimage:latest" comme nom et tag d'image).**

docker build -t kefifaiez/dbimage:latest -f Dockerfile.db .

1. **Vérifiez que les images ont été créées avec succès en utilisant la commande** docker images**.**

docker images

1. **Poussez les images taguées vers votre compte Docker Hub en utilisant la commande** docker push**.**

docker push kefifaiez/webapp:latest

docker push kefifaiez/dbimage:latest

* Cette commande pousse les images taguées "kefifaiez/webapp:latest" et "kefifaiez/dbimage:latest" vers votre compte Docker Hub.
* Vérifiez que les images ont été poussées avec succès en vous connectant à votre compte Docker Hub et en vérifiant que les images sont présentes dans la section "Repositories".

1. **Vous pouvez également vérifier que les images sont disponibles en utilisant la commande** docker search **:**

docker search kefifaiez/webapp

docker search kefifaiez/dbimage

1. **Ces commandes recherchent les images correspondantes dans le registre public de Docker Hub.**

Et voilà ! Vous avez maintenant créé et poussé les images taguées de vos services Web et DB sur votre compte Docker Hub. Vous pouvez maintenant les utiliser pour exécuter des conteneurs sur n'importe quelle machine avec Docker installé

1. **voici le code complet du playbook Ansible en supprimant directement les images Docker et en ignorant les erreurs si les images n'existent pas :**

---

- name: Automate Docker image creation and push

hosts: localhost

gather\_facts: false

tasks:

- name: Remove Docker images

command: docker rmi -f "{{ item }}"

ignore\_errors: true

with\_items:

- kefifaiez/webapp

- kefifaiez/dbimage

- name: Build Docker image for web service

command: docker build -t kefifaiez/webapp:latest .

changed\_when: "'Successfully built' in build\_result.stdout"

register: build\_result

- name: Build Docker image for DB service

command: docker build -t kefifaiez/dbimage:latest -f Dockerfile.db .

changed\_when: "'Successfully built' in build\_result.stdout"

register: build\_result

- name: Push Docker images to Docker Hub

command: docker push "{{ item }}"

with\_items:

- kefifaiez/webapp:latest

- kefifaiez/dbimage:latest

when: build\_result.changed

1. **Vous pouvez enregistrer ce code dans un fichier YAML, par exemple** docker\_automation.yml**, puis exécuter le playbook Ansible avec la commande suivante :**

ansible-playbook docker\_automation.yml