



Kubeflow

NetApp Solutions

NetApp
May 10, 2024

Sommaire

- Kubeflow 1
- Déploiement Kubeflow 1
- Exemple d'opérations et de tâches Kubeflow 2

Kubeflow

Déploiement Kubeflow

Cette section décrit les tâches à effectuer pour déployer Kubeflow dans votre cluster Kubernetes.

Prérequis

Avant d'effectuer l'exercice de déploiement décrit dans cette section, nous supposons que vous avez déjà effectué les tâches suivantes :

1. Vous disposez déjà d'un cluster Kubernetes en fonctionnement et vous exécutez une version de Kubernetes prise en charge par la version Kubeflow que vous envisagez de déployer. Pour obtenir la liste des versions Kubernetes prises en charge, reportez-vous aux dépendances de votre version Kubeflow dans le "[Documentation officielle Kubeflow](#)".
2. Vous avez déjà installé et configuré NetApp Astra Trident dans votre cluster Kubernetes. Pour plus d'informations sur Astra Trident, consultez le "[Documentation Astra Trident](#)".

Définissez la classe de stockage Kubernetes par défaut

Avant de déployer Kubeflow, nous vous recommandons de désigner une classe de stockage par défaut dans votre cluster Kubernetes. Le processus de déploiement Kubeflow peut tenter de provisionner de nouveaux volumes persistants à l'aide de la classe de stockage par défaut. Si aucune classe de stockage n'est désignée comme classe de stockage par défaut, le déploiement risque d'échouer. Pour désigner une classe de stockage par défaut dans votre cluster, effectuez la tâche suivante à partir de l'hôte de démarrage du déploiement. Si vous avez déjà désigné une classe de stockage par défaut dans votre cluster, vous pouvez ignorer cette étape.

1. Désignez une des classes de stockage existantes comme classe de stockage par défaut. Les exemples de commandes ci-dessous montrent la désignation d'une classe de stockage nommée `ontap-ai-flexvols-retain` Comme classe de stockage par défaut.



Le `ontap-nas-flexgroup` Le type de volume interne Trident a une taille de volume persistant minimale et est relativement élevée. Par défaut, Kubeflow tente de provisionner des demandes de volume qui n'ont que quelques Go. Par conséquent, vous ne devez pas désigner une classe de stockage utilisant le `ontap-nas-flexgroup` Taper back-end comme classe de stockage par défaut pour le déploiement Kubeflow.

```

$ kubectl get sc
NAME                                PROVISIONER                        AGE
ontap-ai-flexgroups-retain          csi.trident.netapp.io             25h
ontap-ai-flexgroups-retain-iface1   csi.trident.netapp.io             25h
ontap-ai-flexgroups-retain-iface2   csi.trident.netapp.io             25h
ontap-ai-flexvols-retain            csi.trident.netapp.io             3s
$ kubectl patch storageclass ontap-ai-flexvols-retain -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
storageclass.storage.k8s.io/ontap-ai-flexvols-retain patched
$ kubectl get sc
NAME                                PROVISIONER                        AGE
ontap-ai-flexgroups-retain          csi.trident.netapp.io             25h
ontap-ai-flexgroups-retain-iface1   csi.trident.netapp.io             25h
ontap-ai-flexgroups-retain-iface2   csi.trident.netapp.io             25h
ontap-ai-flexvols-retain (default)   csi.trident.netapp.io             54s

```

Options de déploiement Kubeflow

Il existe de nombreuses options de déploiement Kubeflow. Reportez-vous à la ["Documentation officielle Kubeflow"](#) pour obtenir une liste d'options de déploiement, choisissez l'option qui correspond le mieux à vos besoins.

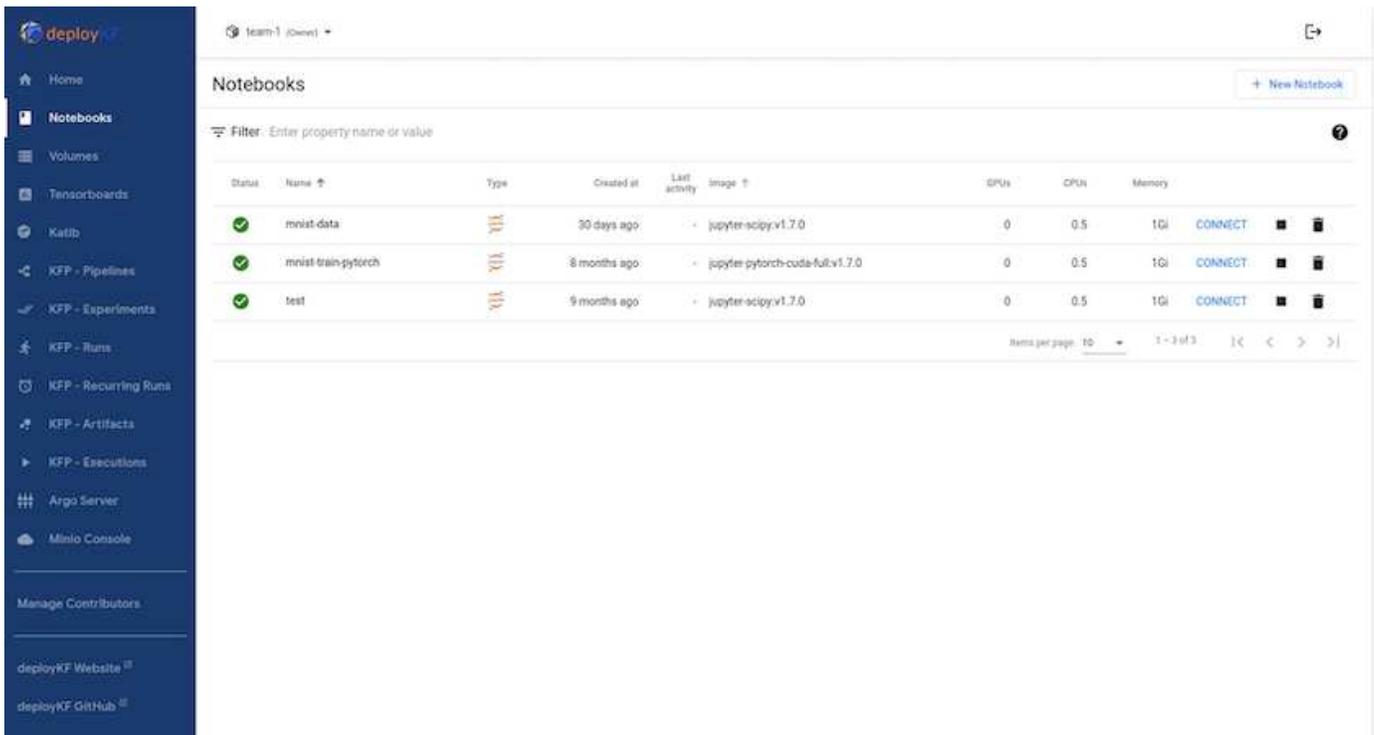


À des fins de validation, nous avons déployé Kubeflow 1.7 avec ["DeployKF"](#) 0.1.1.

Exemple d'opérations et de tâches Kubeflow

Provisionner un espace de travail Jupyter Notebook pour un usage Data Scientist ou Developer

Kubeflow est capable de provisionner rapidement de nouveaux serveurs Jupyter Notebook pour agir en tant qu'espaces de travail de data Scientist. Pour plus d'informations sur les ordinateurs portables Jupyter dans le contexte Kubeflow, reportez-vous au ["Documentation officielle Kubeflow"](#).



Utilisez le kit NetApp DataOps avec Kubeflow

Le "[Kit NetApp Data Science Toolkit pour Kubernetes](#)" Peut être utilisé avec Kubeflow. Grâce au kit NetApp Data Science Toolkit avec Kubeflow, il offre les avantages suivants :

- Les data Scientists peuvent effectuer des opérations avancées de gestion des données NetApp, telles que la création de snapshots et de clones, directement depuis un ordinateur portable Jupyter.
- Les opérations avancées de gestion des données NetApp, telles que la création de snapshots et de clones, peuvent être intégrées dans des workflows automatisés à l'aide de la structure Kubeflow pipelines.

Reportez-vous à la "[Exemples Kubeflow](#)" Section du référentiel GitHub pour le kit NetApp Data Science pour plus d'informations sur l'utilisation du kit avec Kubeflow.

Exemple de flux de travail - entraînez un modèle de reconnaissance d'image à l'aide de Kubeflow et du kit d'outils NetApp DataOps

Cette section décrit les étapes de la formation et du déploiement d'un réseau neuronal pour la reconnaissance des images à l'aide de Kubeflow et du kit NetApp DataOps. L'objectif est de fournir un exemple de formation intégrant le stockage NetApp.

Prérequis

Créez un fichier Dockerfile avec les configurations requises pour les étapes de train et de test dans le pipeline Kubeflow.

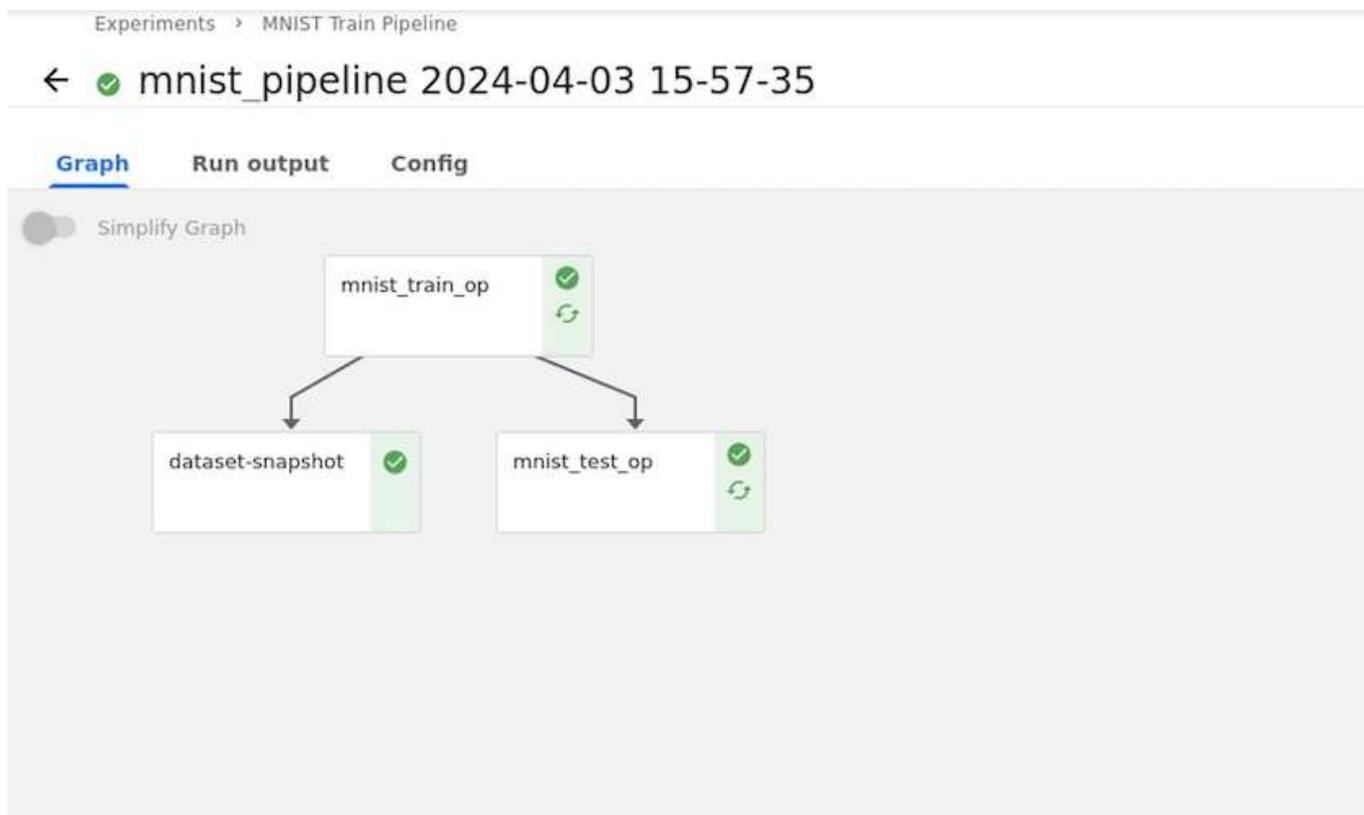
Voici un exemple de fichier Dockerfile -

```
FROM pytorch/pytorch:latest
RUN pip install torchvision numpy scikit-learn matplotlib tensorboard
WORKDIR /app
COPY . /app
COPY train_mnist.py /app/train_mnist.py
CMD ["python", "train_mnist.py"]
```

En fonction de vos besoins, installez toutes les bibliothèques et tous les packages requis pour exécuter le programme. Avant d'entraîner le modèle de machine learning, il est supposé que vous disposez déjà d'un déploiement Kubeflow fonctionnel.

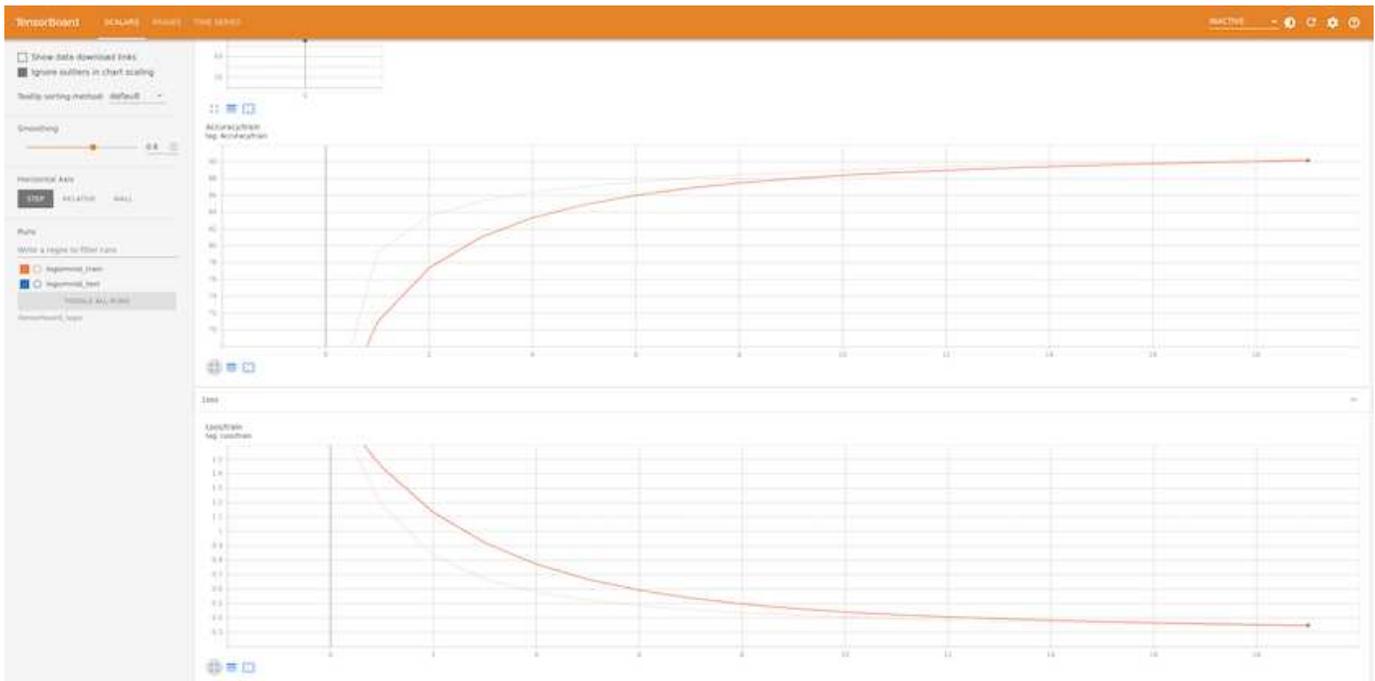
Entraîner un petit NN sur des données MNIST à l'aide de pipelines PyTorch et Kubeflow

Nous utilisons l'exemple d'un petit réseau neuronal formé sur les données MNIST. Le jeu de données MNIST se compose d'images manuscrites de chiffres compris entre 0 et 9. La taille des images est de 28 x 28 pixels. Le dataset est divisé en 60,000 images d'entraînement et 10,000 images de validation. Le réseau neuronal utilisé pour cette expérience est un réseau d'avance à 2 couches. L'entraînement est exécuté à l'aide de Kubeflow pipelines. Reportez-vous à la documentation "[ici](#)" pour en savoir plus. Notre pipeline Kubeflow intègre l'image docker de la section Prerequisites.



Visualisez les résultats à l'aide de Tensorboard

Une fois le modèle entraîné, nous pouvons visualiser les résultats à l'aide de Tensorboard. "[Tensorboard](#)" Elle est disponible en tant que fonctionnalité dans le tableau de bord Kubeflow. Vous pouvez créer un tableau de tension personnalisé pour votre travail. Un exemple ci-dessous montre le tracé de la précision de l'entraînement par rapport à nombre de séries de tests et de pertes d'entraînement par rapport à nombre de séries de tests.



Testez les hyperparamètres à l'aide de Katib

"Katib" Est un outil de Kubeflow qui peut être utilisé pour tester les hyperparamètres du modèle. Pour créer une expérience, définissez d'abord une mesure/un objectif souhaité. Il s'agit généralement de la précision du test. Une fois la mesure définie, choisissez les hyperparamètres que vous souhaitez utiliser (optimiseur/apprentissage_rate/nombre de couches). Katib effectue un balayage hyperparamètre avec les valeurs définies par l'utilisateur pour trouver la meilleure combinaison de paramètres qui répondent à la mesure souhaitée. Vous pouvez définir ces paramètres dans chaque section de l'interface utilisateur. Vous pouvez également définir un fichier **YAML** avec les spécifications nécessaires. Vous trouverez ci-dessous une illustration d'une expérience Katib -

Objective	
Name	Validation-accuracy
Type	maximize
Goal	0.9
Additional metrics	Train-accuracy

Trials	
Max failed trials	3
Max trials	12
Parallel trials	3

Parameters	
lr	Parameter type: double Min: 0.01 Max: 0.03
num-layers	Parameter type: int Min: 1 Max: 64
optimizer	Parameter type: categorical sgd, adam, ttr

Algorithm	
Name	grid

Metrics collector	
Collector type	File

The screenshot shows the Kubeflow Katib interface. On the left is a navigation sidebar with options like Home, Notebooks, Volumes, Tensorboards, and Katib. The main panel displays 'Experiment details' for 'mnist-pytorch'. A message at the top states 'Couldn't find any successful Trial'. Below this is a table with columns: OVERVIEW, TRIALS, DETAILS, and YAML. The table lists various metrics such as Name, Status (Experiment is running), Best trial, Best trial's params, Best trial performance, User defined goal (Validation-accuracy > 0.9), Running trials (3), Failed trials (0), and Succeeded trials (0). At the bottom, there is a 'Filter' input field.

Utilisez les instantanés NetApp pour enregistrer les données pour la traçabilité

Lors de l'entraînement du modèle, nous pouvons enregistrer un instantané du dataset d'entraînement à des fins de traçabilité. Pour ce faire, nous pouvons ajouter une étape snapshot au pipeline, comme illustré ci-dessous. Pour créer le snapshot, nous pouvons utiliser le "[Kit NetApp DataOps pour Kubernetes](#)".

```
@dsl.pipeline(
    name = 'MNIST Classification Pipeline',
    description = 'Train a simple NN for classification'
)
def mnist_pipeline():
    mnist_train_task = mnist_train_op()
    mnist_train_task.apply(
        kfp.onprem.mount_pvc('mnist-data', 'mnist-data-vol', '/mnt/data/')
    )

    mnist_test_task = mnist_test_op()
    mnist_test_task.apply(
        kfp.onprem.mount_pvc('mnist-data', 'mnist-data-vol', '/mnt/data/')
    )

    volume_snapshot_name = "mnist-pytorch-snapshot"
    dataset_snapshot = dsl.ContainerOp(
        name="dataset-snapshot",
        image="python:3.9",
        command=["/bin/bash", "-c"],
        arguments=["\
            python3 -m pip install netapp-dataops-k8s && \
            echo "" + volume_snapshot_name + "" > /volume_snapshot_name.txt && \
            netapp_dataops_k8s_cli.py create volume-snapshot --pvc-name="" + "mnist-data" + " --snapshot-name="" + str(volume_snapshot_name) + " --namespace={workflow.namespace}"],
        file_outputs={'volume_snapshot_name': '/volume_snapshot_name.txt'}
    )
    mnist_test_task.after(mnist_train_task)
    dataset_snapshot.after(mnist_train_task)
```

Reportez-vous à la "[Exemple de kit NetApp DataOps pour Kubeflow](#)" pour en savoir plus.

Informations sur le copyright

Copyright © 2024 NetApp, Inc. Tous droits réservés. Imprimé aux États-Unis. Aucune partie de ce document protégé par copyright ne peut être reproduite sous quelque forme que ce soit ou selon quelque méthode que ce soit (graphique, électronique ou mécanique, notamment par photocopie, enregistrement ou stockage dans un système de récupération électronique) sans l'autorisation écrite préalable du détenteur du droit de copyright.

Les logiciels dérivés des éléments NetApp protégés par copyright sont soumis à la licence et à l'avis de non-responsabilité suivants :

CE LOGICIEL EST FOURNI PAR NETAPP « EN L'ÉTAT » ET SANS GARANTIES EXPRESSES OU TACITES, Y COMPRIS LES GARANTIES TACITES DE QUALITÉ MARCHANDE ET D'ADÉQUATION À UN USAGE PARTICULIER, QUI SONT EXCLUES PAR LES PRÉSENTES. EN AUCUN CAS NETAPP NE SERA TENU POUR RESPONSABLE DE DOMMAGES DIRECTS, INDIRECTS, ACCESSOIRES, PARTICULIERS OU EXEMPLAIRES (Y COMPRIS L'ACHAT DE BIENS ET DE SERVICES DE SUBSTITUTION, LA PERTE DE JOUISSANCE, DE DONNÉES OU DE PROFITS, OU L'INTERRUPTION D'ACTIVITÉ), QUELLES QU'EN SOIENT LA CAUSE ET LA DOCTRINE DE RESPONSABILITÉ, QU'IL S'AGISSE DE RESPONSABILITÉ CONTRACTUELLE, STRICTE OU DÉLICTELLE (Y COMPRIS LA NÉGLIGENCE OU AUTRE) DÉCOULANT DE L'UTILISATION DE CE LOGICIEL, MÊME SI LA SOCIÉTÉ A ÉTÉ INFORMÉE DE LA POSSIBILITÉ DE TELS DOMMAGES.

NetApp se réserve le droit de modifier les produits décrits dans le présent document à tout moment et sans préavis. NetApp décline toute responsabilité découlant de l'utilisation des produits décrits dans le présent document, sauf accord explicite écrit de NetApp. L'utilisation ou l'achat de ce produit ne concède pas de licence dans le cadre de droits de brevet, de droits de marque commerciale ou de tout autre droit de propriété intellectuelle de NetApp.

Le produit décrit dans ce manuel peut être protégé par un ou plusieurs brevets américains, étrangers ou par une demande en attente.

LÉGENDE DE RESTRICTION DES DROITS : L'utilisation, la duplication ou la divulgation par le gouvernement sont sujettes aux restrictions énoncées dans le sous-paragraphe (b)(3) de la clause Rights in Technical Data-Noncommercial Items du DFARS 252.227-7013 (février 2014) et du FAR 52.227-19 (décembre 2007).

Les données contenues dans les présentes se rapportent à un produit et/ou service commercial (tel que défini par la clause FAR 2.101). Il s'agit de données propriétaires de NetApp, Inc. Toutes les données techniques et tous les logiciels fournis par NetApp en vertu du présent Accord sont à caractère commercial et ont été exclusivement développés à l'aide de fonds privés. Le gouvernement des États-Unis dispose d'une licence limitée irrévocable, non exclusive, non cessible, non transférable et mondiale. Cette licence lui permet d'utiliser uniquement les données relatives au contrat du gouvernement des États-Unis d'après lequel les données lui ont été fournies ou celles qui sont nécessaires à son exécution. Sauf dispositions contraires énoncées dans les présentes, l'utilisation, la divulgation, la reproduction, la modification, l'exécution, l'affichage des données sont interdits sans avoir obtenu le consentement écrit préalable de NetApp, Inc. Les droits de licences du Département de la Défense du gouvernement des États-Unis se limitent aux droits identifiés par la clause 252.227-7015(b) du DFARS (février 2014).

Informations sur les marques commerciales

NETAPP, le logo NETAPP et les marques citées sur le site <http://www.netapp.com/TM> sont des marques déposées ou des marques commerciales de NetApp, Inc. Les autres noms de marques et de produits sont des marques commerciales de leurs propriétaires respectifs.