

TensorFlow sur Azure (Brouillon)

Antoine Gréa | Aptitek

2024-04-25

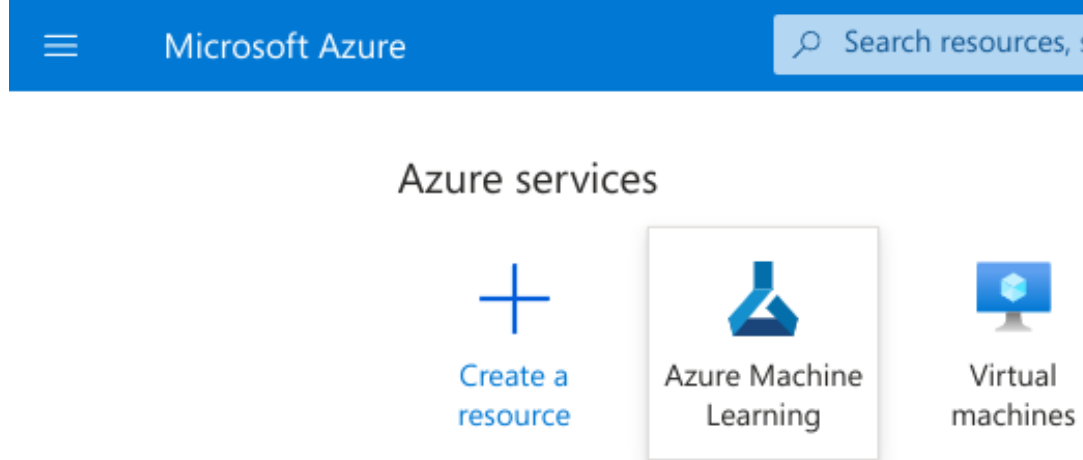
Compute sur Microsoft Azure

Ce projet nécessite un abonnement Microsoft Azure

Afin de ne pas gaspiller des fonds, pensez à supprimer toutes ressources ou groupes inutilisés.

Pour créer un compute sur Azure, vous devez aller sur <https://portal.azure.com>.

On commence par aller dans le service **Azure Machine Learning** pour créer une ressource. Si vous ne le trouvez pas allez



dans la recherche et taper son nom.



Figure 1: Vibecheck all texts !

[Home](#) >

Azure Ma

ESTIAM

[+ Create](#) ▾



New wor
For ML pr



New regi
For sharin

Vous vous retrouverez devant une liste vide (normalement). Supprimez tout espace de travail inutilisé.

Créez ensuite un nouvel espace de travail avec le bouton [**+ Create**] en haut à gauche. Vous serez redirigé vers un formulaire que vous complétez comme suit :

Vous devez créer le groupe de ressource et vérifiez bien l'abonnement ainsi que la région. On clique sur [**Review + create**]

Azure Machine Learning ...

Create a machine learning workspace

[Basics](#) [Networking](#) [Encryption](#) [Identity](#) [Tags](#) [Review + create](#)

Resource details

Every workspace must be assigned to an Azure subscription, which is where billing happens. You use resource groups like folders to organize and manage resources, including the workspace you're about to create.

[Learn more about Azure resource groups](#)

Subscription * ⓘ

Subscription ▼

Resource group * ⓘ

tensorvibe ▼

[Create new](#)

Workspace details

Configure your basic workspace settings like its storage connection, authentication, container, and more. [Learn more](#)

Name * ⓘ

tensorvibe ✓

Region * ⓘ

France Central ▼

Storage account * ⓘ

(new) tensorvibe0850852966 ▼

[Create new](#)

Key vault * ⓘ

(new) tensorvibe3837163291 ▼

[Create new](#)

Application insights * ⓘ

(new) tensorvibe7049632100 ▼

[Create new](#)

Container registry ⓘ

None ▼

[Create new](#)

[Review + create](#)

[< Previous](#)

[Next : Networking](#)

Figure 2: Formulaire pour créer l'espace de travail

Azure Machine Learning ...

Create a machine learning workspace

✓ Validation passed

Basics Networking Encryption Identity Tags Review + create

Basics

Subscription	Subscription
Resource group	tensorvibe
Region	France Central
Name	tensorvibe
Storage account	(new) tensorvibe8895431384
Key vault	(new) tensorvibe5452292706
Application insights	(new) tensorvibe8146038473
Container registry	None

Networking

Connectivity method	Enable public access from all networks
Network isolation	Public

Encryption

Encryption type	Microsoft-managed keys
-----------------	------------------------

Identity

Identity type	System assigned
Enable HBI Flag	Disabled

Create

< Previous

Next >

[Download a template for automation](#)

puis on vérifie

avant de cliquer sur [Create].

Après quelques minutes, vous devrez voir ceci :

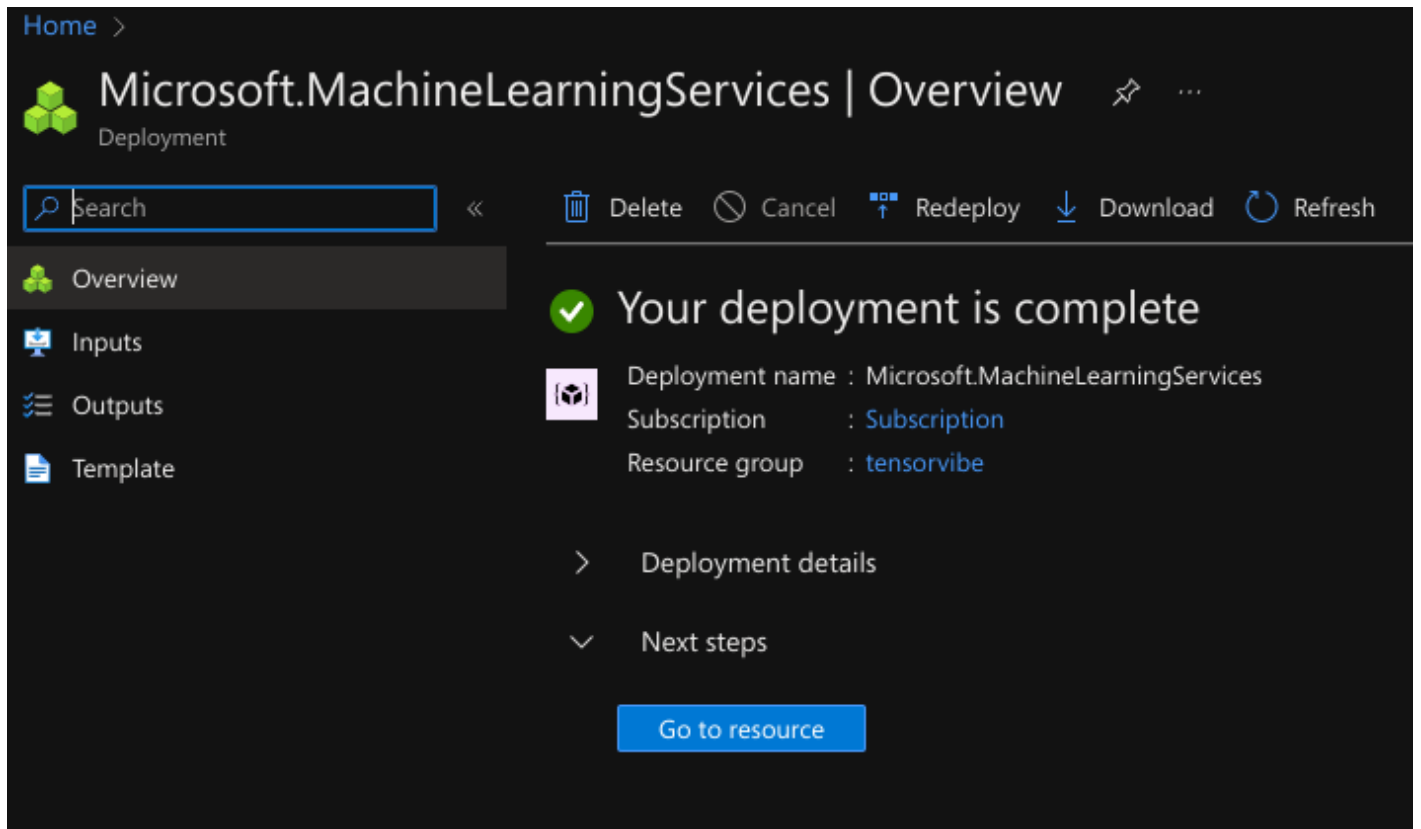


Figure 3: Espace de travail créé avec succès

Cliquez [Go to resource] puis [Launch studio]. Si vous ne le trouvez pas le studio est sur <https://ml.azure.com/>

Une fois sur le portail machine learning, ouvrez votre espace de travail :

Vous créer un compute pour exécuter votre code tensorflow. On a besoin de ceci afin de faire des calculs sur une machine plus puissante (Normalement avec une bonne carte graphique). Pour se faire allez dans l'onglet **Compute** > [+ New].

Malheureusement l'abonnement par défaut d'Azure ne permet pas d'utiliser de compute GPU... On crée donc un compute CPU comme suit :

Sélectionnez bien la machine **Standard_E4ds_v4** Afin d'avoir les meilleurs performances pour ce projet.

Cette fois-ci, on clique sur suivant jusqu'à l'écran **Security**. Vous devrez mettre en place un accès SSH. Activez l'option (o) **Enable SSH access** puis sélectionnez l'utilisation d'une clé publique existante. Entrez la clé dans le champ approprié



Work with your models in Azure Machine Learning Studio

The Azure Machine Learning Studio is a web app where you can build, train, test, and deploy ML models. Launch it now to start exploring, or [learn more about the Azure Machine Learning studio](#)

Launch studio

Figure 4: Espace de travail Azure ML

Create compute instance

✓ Required settings

✓ Scheduling
optional

3 Security
optional

✓ Applications
optional

✓ Tags
optional

6 Review

Security

Configure security settings such as SSH, virtual network, root access, and managed identity for your compute instance.

User assignment

☐ Assign to another user ⓘ

Assigned identity

☐ Assign a managed identity ⓘ

SSH

☒ Enable SSH access ⓘ

Admin username * ⓘ

azureuser

SSH public key source *

Use existing public key

SSH public key * ⓘ

ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAQGC6VNnDF6i3gzlG2YXEVLM2hHuVC3JGMZrLLrjXpCQzULY6JQcRUH8BbmAl7zeCkDihDzNe3GFX8lrKds8ohVrO7yUCuCHkHoHTt9ZfwDUo0PpvSjNyoVHgvlXR69g/HONa1/bPME14F8n69gf7vAWQvp+mR

Virtual network

☐ Enable virtual network ⓘ

Azure AI | Machine Learning Studio

Home

Model catalog

Authoring

Notebooks

Automated ML

Designer

Prompt flow

Assets

Data

Jobs

Components

Pipelines

Environments

Models

Endpoints

Manage

Compute

Monitoring

Data Labeling

ESTIAM > tensorvibe > Compute

Compute

The "Kubernetes clusters" tab is now where you can access previous versions of "inference clusters" (also known as "AKS clusters") and "attach" with any previously created compute targets using those types. [Learn more](#) about Kubernetes clusters.


Compute instances

Compute clusters

Kubernetes clusters

Attached computes

Choose from a selection of CPU or GPU instances preconfigured with popular tools such as VS Code, JupyterLab, Jupyter, and frameworks, and GPU drivers. [Learn more about compute instances](#)



Get started with Azure Machine Learning notebooks and R scripts by creating a compute instance

Choose from a selection of CPU or GPU instances preconfigured with popular tools such as VS Code, JupyterLab, Jupyter, and RStudio, ML packages, deep learning frameworks, and GPU drivers. [Learn more](#)

+ New

Figure 5: Création du compute

8

Create compute instance



- 1 Required settings
- 2 Scheduling
optional
- 3 Security
optional
- 4 Applications
optional
- 5 Tags
optional
- 6 Review

Configure required settings

Select the name and virtual machine size you would like to use for your compute instance

① Note that a compute instance can not be shared. It can only be used by a single assigned user. By default, it will be assigned to the creator and you can change this to a different user in the Security step.

Compute name * ①



tensorvibe

Virtual machine type ①

☒ CPU ☐ GPU

Virtual machine size ①

☒ Select from recommended options ☐ Select from all options

Name ↑	Category	Workload types	Available quota ①	Cost ①
<input type="radio"/> Standard_DS11_v2 2 cores, 14GB RAM, 28GB storage	Memory optimized	Development on Notebooks (or other IDE) and light weight testing	6 cores	\$0.23/hr
<input type="radio"/> Standard_DS3_v2 4 cores, 14GB RAM, 28GB storage	General purpose	Classical ML model training on small datasets	6 cores	\$0.35/hr
<input checked="" type="radio"/> Standard_E4ds_v4 4 cores, 32GB RAM, 150GB storage	Memory optimized	Data manipulation and training on medium-sized datasets (1-10GB)	10 cores	\$0.34/hr
<input type="radio"/> Standard_F4s_v2 4 cores, 8GB RAM, 32GB storage	Compute optimiz...	Data manipulation and training on large datasets (>10 GB)	16 cores	\$0.20/hr

Review + Create

Back

Next

Cancel

Figure 6: Formulaire de création du compute

Client et clé SSH

SSH (Secure SHell) est un protocole réseau de connexion à une console distante. Par défaut, le client et serveur SSH sont désactivé sur Windows 11. Vous devez donc soit l'activer soit en installer un.

On utilise souvent Chocolatey pour installer des packages sur Windows. Pour l'installer, utilisez la combinaison de touche [Win] + [X] > Powershell/Console Admin puis collez ceci dans le terminal et appuyez sur [Entrer].

```
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [Sy
```

Sur d'autre OS, veuillez utiliser votre gestionnaire de package habituel.

Méthode 1 : SSHFS + Sirikali

La méthode la plus simple, selon moi, est d'utiliser Chocolatey pour installer sshfs :

```
choco install sshfs -y
```

Puis installez Sirikali manuellement

Méthode 2 : Cyberduck et autres

Vous pouvez utiliser les alternatives suivantes :

- Cyberduck `choco install cyberduck -y`. Pour se connecter ouvrez Cyberduck et suivez la documentation afin de générer une clé privée et publique.
- Bitvise `choco install bitvise-ssh-client -y`

Méthode 3 : Openssh via Chocolatey

Utilisez la combinaison de touche [Win] + [X] > Powershell/Console Admin puis collez ceci dans le terminal et appuyez sur [Enter].

```
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [Sy  
choco install mls-software-openssh -y
```

Méthode 4 : Tutorial Officiel Microsoft

Vous pouvez aussi suivre les tutoriaux Microsoft sur l'installation d'openssh et la création de clé SSH

Python et Tensorflow

Tensor-quoi ?

TensorFlow est une bibliothèque open-source développée par Google pour le calcul numérique et l'apprentissage automatique. Elle est largement utilisée pour construire, entraîner et déployer des modèles d'apprentissage automatique dans une variété de domaines, notamment la vision par ordinateur, le traitement du langage naturel, la reconnaissance vocale, etc.

Principes de Base

1. **Graphes de Calcul** : TensorFlow est basé sur un modèle de programmation par graphe de calcul. Les opérations mathématiques sont représentées sous forme de nœuds dans un graphe, et les données sont représentées sous forme de tenseurs (tableaux multidimensionnels) qui circulent entre les nœuds.
2. **Tenseurs** : Les tenseurs sont la structure de données fondamentale de TensorFlow. Ils peuvent être de différentes dimensions : des scalaires (0D), des vecteurs (1D), des matrices (2D) ou des tenseurs de rang supérieur.
3. **Sessions** : Pour exécuter des opérations dans un graphe TensorFlow, vous devez créer une session TensorFlow. Une session encapsule l'environnement dans lequel les opérations de calcul sont exécutées et les tenseurs sont évalués.
4. **Variables** : Les variables sont des tenseurs dont la valeur peut être modifiée pendant l'exécution du graphe. Elles sont souvent utilisées pour stocker les paramètres des modèles d'apprentissage automatique, comme les poids et les biais.

5. **Optimiseurs** : TensorFlow propose une gamme d'optimiseurs pour ajuster les paramètres des modèles pendant l'entraînement. Ces optimiseurs utilisent des algorithmes d'optimisation tels que la descente de gradient stochastique (SGD), Adam, RMSProp, etc.

Architecture de modèle Réseaux de Neurones Convolutifs (CNN) :

Les CNN sont particulièrement efficaces pour la vision par ordinateur, mais ils peuvent également être utilisés pour le traitement du langage naturel. Ils sont composés de couches convolutionnelles qui extraient des caractéristiques des données en effectuant des opérations de convolution, suivies de couches de pooling pour réduire la dimensionnalité des caractéristiques extraites.

Réseaux de Neurones Récurrents (RNN) :

Les RNN sont adaptés au traitement de données séquentielles, comme le langage naturel. Ils ont une boucle récurrente qui leur permet de conserver une mémoire à court terme des séquences précédentes. Cela les rend efficaces pour les tâches de prédiction séquentielle, telles que la traduction automatique et la génération de texte.

Réseaux de Neurones Récurrents à Longue Mémoire (LSTM) et Réseaux de Neurones Récurrents Gated Recurrent Unit (GRU) :

Les LSTM et les GRU sont des variantes des RNN qui ont été conçues pour mieux gérer le problème du gradient qui disparaît ou explose pendant l'entraînement des RNN. Ils utilisent des mécanismes de porte pour contrôler le flux d'informations à travers les cellules récurrentes, ce qui leur permet de capturer des dépendances à plus long terme dans les séquences.

Transformers :

Les Transformers sont une architecture de modèle récente qui a révolutionné de nombreux domaines du traitement du langage naturel. Ils se composent de couches d'attention multi-têtes qui permettent au modèle de focaliser son attention sur différentes parties de la séquence en même temps, capturant ainsi des dépendances à longue distance de manière efficace.

Vérifier l'installation de Python et Tensorflow

Créez le fichier Python `app.py` :

```
import sys
import tensorflow as tf

def main() -> int:
    print("Hello")
    return 0

if __name__ == '__main__':
    sys.exit(main())
```

Vous pouvez utiliser la commande `nano` afin de créer des fichier texte en ligne de commande. On enregistre avec `[Ctrl] + [O] > [Enter]`.

Puis tapez la commande : `python app.py`

Si besoin : Installation via docker

Via SSH sur la machine virtuelle, exécutez les commandes suivantes :

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh ./get-docker.sh
sudo usermod -aG docker $USER
sudo apt install docker-compose -y
docker -v
docker-compose -v
```

Créez le fichier `Dockerfile` suivant :

```
FROM tensorflow/tensorflow
```

```
COPY ${PWD} /app
```

```
WORKDIR /app
```

```
CMD python /app/app.py
```

Puis vous créez l'image docker avec la commande `docker build . -t tensorflow` puis vous lancez le conteneur via `docker run tensorflow`. Vous pouvez voir les logs avec `docker logs tensorflow` et accéder à la console du conteneur avec `docker exec -it tensorflow bash`

Git et Github

Pour votre projet, utilisez Git. Pour l'installer en local vous pouvez utiliser `choco install git -y` dans une console administrateur ou bien utilisez VScode `choco install vscode`

Pour utiliser Git sur la machine virtuelle :

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install git
```

```
git --version
```