

# Projet Cloud : Automatiser le déploiement des VM

---

**École : Ecole Nationale Supérieure Polytechnique de Yaoundé**

**Encadreur : Jean-Marc Menaud - Professor - IMT-Atlantique**

---

## Participants :

1. MENRA WEDWANG ROMIAL
2. NCHOUWET NGANE LOIC
3. KAMGA DJIDJOU WILFRIED JUNIOR
4. AMOU'OU AMOU'OU GEORGES JUNIOR
5. FOFU NOBOSSE

## Projet Cloud : Automatiser le déploiement des VM

École : Ecole Nationale Supérieure Polytechnique de Yaoundé

Encadreur : Jean-Marc Menaud - Professor - IMT-Atlantique

Participants :

## Introduction

## Fonctionnalités Implémentées

## Documentation

Installation et Configuration

Utilisation

1. Déploiement d'une VM à partir d'un fichier OVA :
2. Création d'une VM à partir d'un fichier ISO
3. Déploiement de VM sur de multiples ESXi

## Conclusion

# Introduction

---

L'application que nous avons développée en Python est une solution pour gérer et déployer des machines virtuelles sur VMware ESXi. Elle offre une gamme de fonctionnalités qui permettent aux utilisateurs de créer, déployer, gérer et contrôler des machines virtuelles à partir de différentes sources, telles que des fichiers OVA ou des images ISO.

## Fonctionnalités Implémentées

---

1. **Déploiement de VM à partir du fichier OVA** : Permet de déployer une machine virtuelle à partir d'un fichier OVA préconfiguré.
2. **Création d'une VM à partir d'un fichier ISO** : Permet la création et le déploiement d'une machine virtuelle à partir d'une image ISO.
3. **Récupération de toutes les VM** : Offre une liste complète de toutes les machines virtuelles actuellement présentes sur le serveur ESXi.
4. **Renommage de la VM** : Permet de changer le nom d'une machine virtuelle existante.
5. **Liste des banques de données** : Fournit une liste des banques de données disponibles sur le serveur ESXi.
6. **Téléversement de fichiers** : Permet de transférer des fichiers vers le serveur ESXi pour une utilisation ultérieure.
7. **Suppression de la VM** : Offre la possibilité de supprimer une machine virtuelle spécifique.
8. **Détails du centre V** : Donne des informations détaillées sur le centre V ou l'hôte ESXi.
9. **État de la VM** : Indique si une machine virtuelle est sous tension ou non.
10. **Déploiement de VM sur de multiples ESXi** : Permet le déploiement simultané de machines virtuelles sur plusieurs hôtes ESXi.

## Documentation

---

### Installation et Configuration

Pour utiliser cette application, assurez-vous d'avoir Python installé. Ensuite, clonez le référentiel depuis

GitHub [https://github.com/menraromial/cloud\\_esxi.git](https://github.com/menraromial/cloud_esxi.git)

et suivez les étapes de configuration suivantes :

Créer un environnement virtuel Python est une excellente pratique pour isoler les dépendances et les bibliothèques spécifiques à un projet, évitant ainsi les conflits entre différentes versions de packages. Voici un guide pas à pas pour créer et utiliser un environnement virtuel Python :

1. **Assurez-vous d'avoir Python installé** :

Vérifiez si Python est installé sur votre système en ouvrant un terminal ou une invite de commande et en tapant :

```
python --version
```

Si Python est installé, cela affichera la version installée.

## 2. Installation de `virtualenv` (si ce n'est pas déjà fait) :

```
pip install virtualenv
```

## 3. Créez un répertoire pour votre projet (si vous n'en avez pas déjà un) :

```
mkdir mon_projet  
cd mon_projet
```

## 4. Créez un environnement virtuel :

Utilisez `virtualenv` pour créer un environnement virtuel spécifique à Python dans votre répertoire de projet :

**Sur Windows :**

```
python -m virtualenv venv
```

**Sur macOS et Linux :**

```
python3 -m venv venv
```

## 5. Activez l'environnement virtuel :

**Sur Windows :**

```
venv\Scripts\activate
```

**Sur macOS et Linux :**

```
source venv/bin/activate
```

Vous saurez que l'environnement virtuel est activé car le nom de votre environnement virtuel apparaîtra dans votre invite de commande (par exemple, `(venv)` ).

Pour installer les dépendances d'un projet Python à partir d'un fichier `requirements.txt` , vous pouvez utiliser la commande `pip` suivante :

```
pip install -r requirements.txt
```

Assurez-vous de vous placer dans le répertoire racine de votre projet où se trouve le fichier `requirements.txt` . Cette commande va parcourir le fichier `requirements.txt` et installer toutes les dépendances spécifiées, ce qui est pratique pour configurer rapidement un environnement de développement avec les bibliothèques nécessaires.

Pour quitter l'environnement virtuel, utilisez la commande :

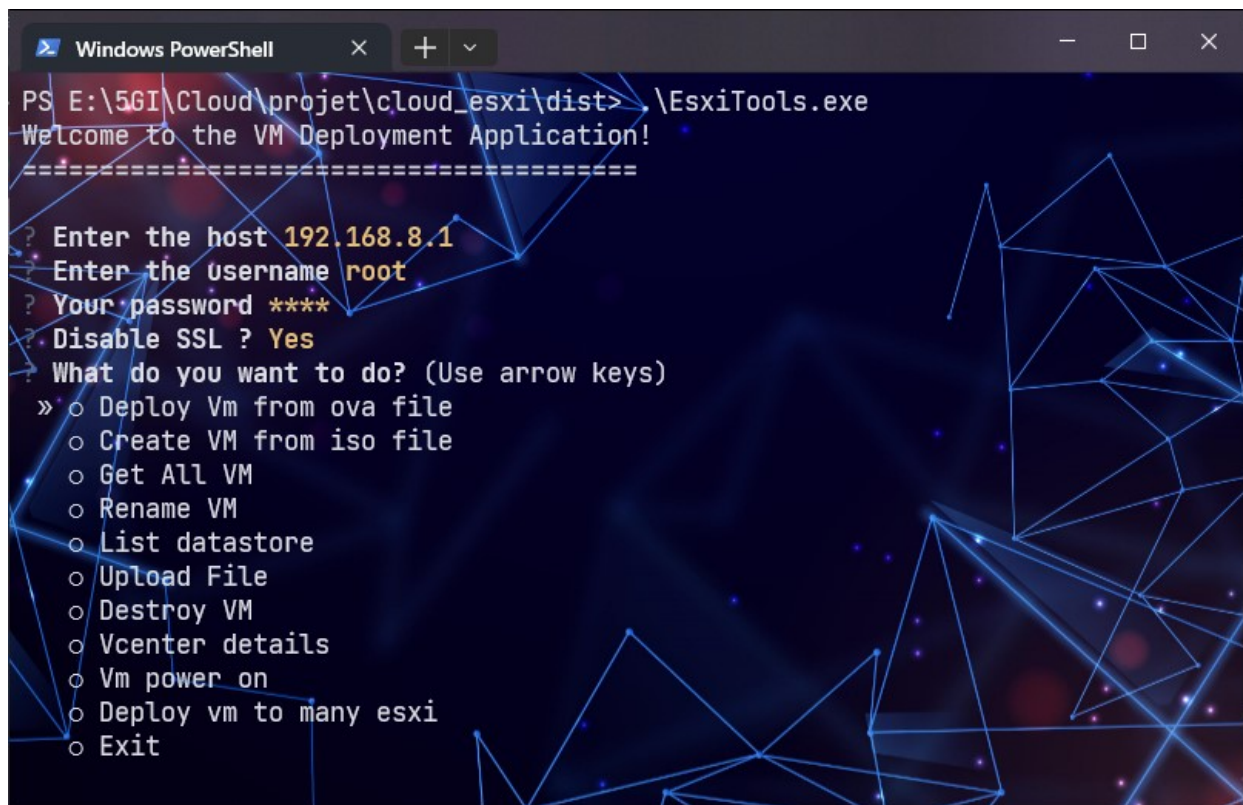
```
deactivate
```

**Les utilisateurs de Windows peuvent se passer de toutes ces étapes en lançant directement le fichier binaire (.exe) en console. Vous pouvez télécharger le fichier .exe avec le code source y compris ici:**

[https://github.com/menraromial/cloud\\_esxi/releases/tag/v1.0.2](https://github.com/menraromial/cloud_esxi/releases/tag/v1.0.2)

## Utilisation

Pour utiliser les fonctionnalités, il est nécessaire pour l'utilisateur de saisir les paramètres de connexion à l'hôte ESXI comme sur l'image si dessous:



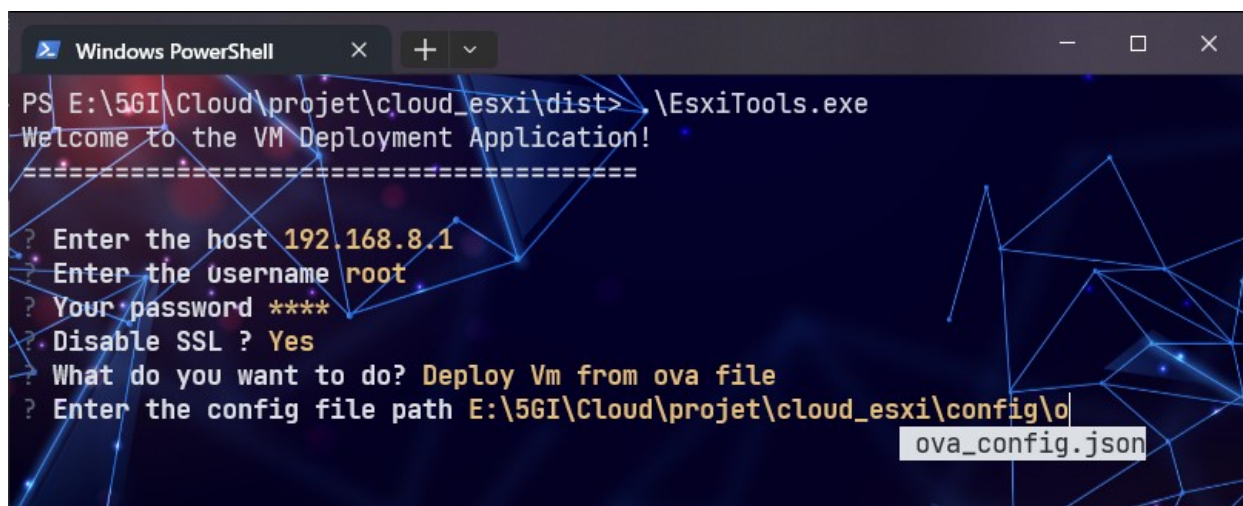
```
PS E:\5GI\Cloud\projet\cloud_esxi\dist> .\EsxiTools.exe
Welcome to the VM Deployment Application!
=====
? Enter the host 192.168.8.1
? Enter the username root
? Your password ****
? Disable SSL ? Yes
? What do you want to do? (Use arrow keys)
  » ○ Deploy Vm from ova file
    ○ Create VM from iso file
    ○ Get All VM
    ○ Rename VM
    ○ List datastore
    ○ Upload File
    ○ Destroy VM
    ○ Vcenter details
    ○ Vm power on
    ○ Deploy vm to many esxi
    ○ Exit
```

Vous devez maintenant choisir une fonctionnalité parmi la liste

## 1. Déploiement d'une VM à partir d'un fichier OVA :

Une fois que vous avez choisi **Deploy Vm from ova file**, vous devez fournir le chemin vers le fichier de configuration qui doit être en json.

Vous pouvez voir sur l'image si dessous que notre application gère autocomplétions des chemins



```
PS E:\5GI\Cloud\projet\cloud_esxi\dist> .\EsxiTools.exe
Welcome to the VM Deployment Application!
=====
? Enter the host 192.168.8.1
? Enter the username root
? Your password ****
? Disable SSL ? Yes
? What do you want to do? Deploy Vm from ova file
? Enter the config file path E:\5GI\Cloud\projet\cloud_esxi\config\o
ova_config.json
```

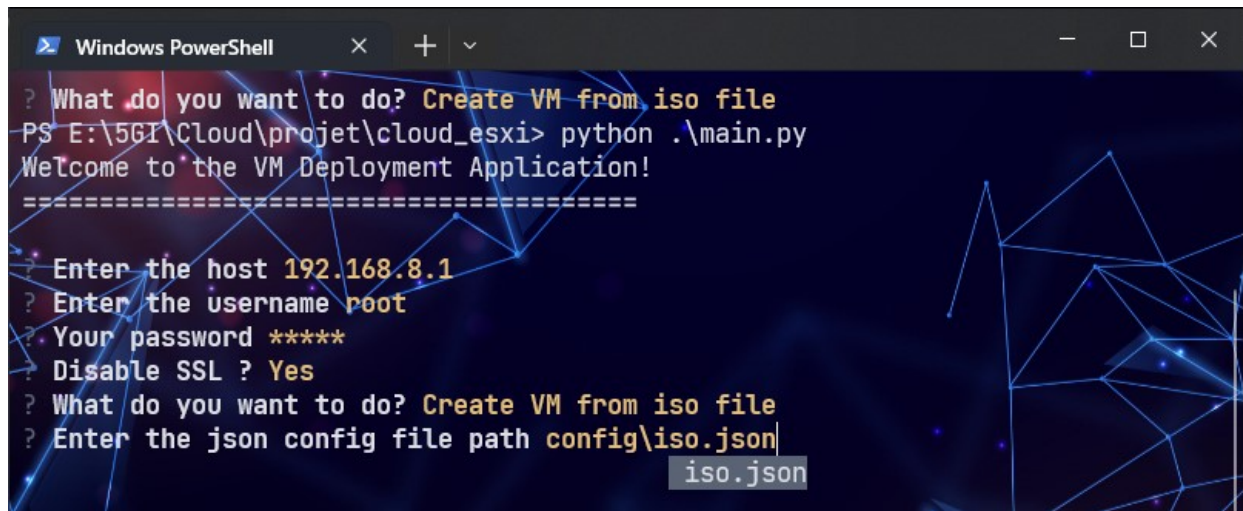
Voici la structure du fichier **json**, il est important de respecter cela pour que l'application marche:

```
{
  "ova_path": "E:\\OVA-Linux\\yVM.ova",
  "datacenter_name": "",
  "datastore_name": "",
  "resource_pool": "",
  "num_instance": 2
}
```

You, 2 hours ago • prod

Nous avons des paramètres obligatoires : **ova\_path** qui est le chemin du fichier **ova** et **num\_instance** le nombre d'instance de machine virtuelle à déployer

## 2. Création d'une VM à partir d'un fichier ISO



```
Windows PowerShell
? What do you want to do? Create VM from iso file
PS E:\5GI\Cloud\projet\cloud_esxi> python .\main.py
Welcome to the VM Deployment Application!
=====
? Enter the host 192.168.8.1
? Enter the username root
? Your password *****
? Disable SSL ? Yes
? What do you want to do? Create VM from iso file
? Enter the json config file path config\iso.json
iso.json
```

Le fichier de configuration est structuré comme suit:

```
{
  "iso_path": "E:\\OVA-Linux\\Core-5.4.iso",
  "num_cpus": 1,
  "vm_name": "test_iso",
  "memory": 1024,
  "unit_number": 0,
  "capacity": 20971520,
  "disk_mode": "persistent",
  "log_directory": null,
  "snapshot_directory": null,
  "suspend_directory": null
}
```

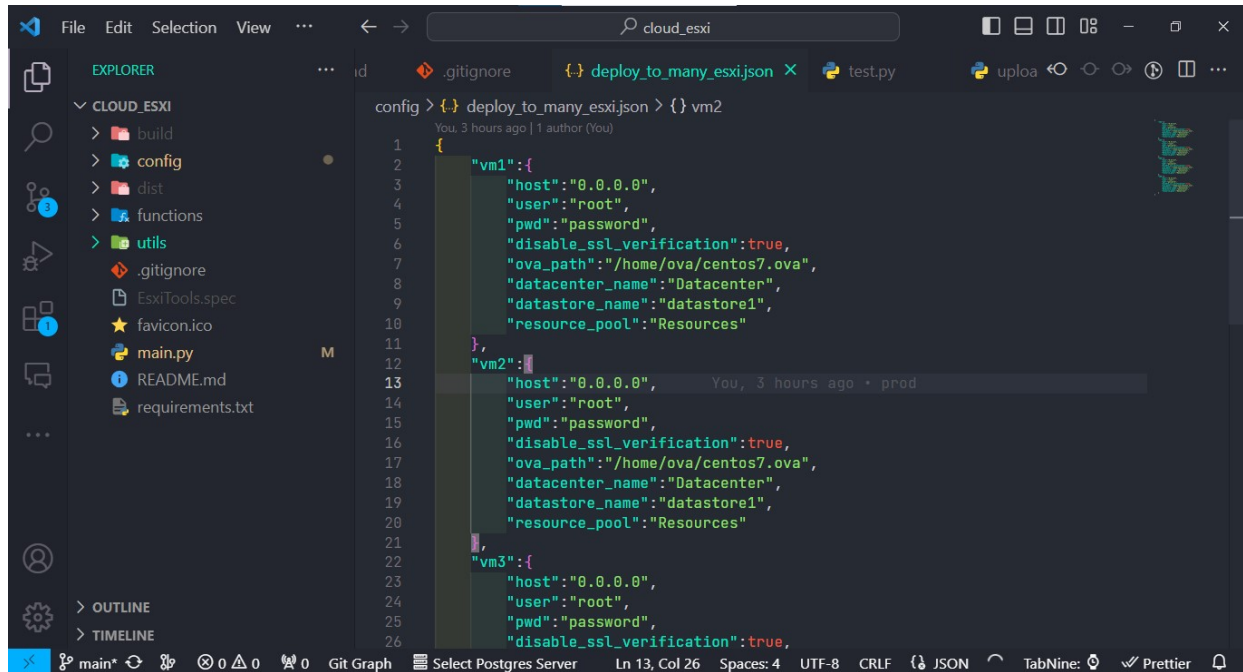
You, 3 hours ago • prod



Il faut préciser entre autre; le chemin de l'iso, le nombre de CPU, le nom de la machine virtuelle, la mémoire (RAM), la capacité du disque dur, etc.

### 3. Déploiement de VM sur de multiples ESXi

La démarche est identique aux deux derniers, c'est juste la structure du fichier de configuration qui diffère:



```
config > {} deploy_to_many_esxi.json > {} vm2
You, 3 hours ago | 1 author (You)
1  {
2      "vm1": {
3          "host": "0.0.0.0",
4          "user": "root",
5          "pwd": "password",
6          "disable_ssl_verification": true,
7          "ova_path": "/home/ova/centos7.ova",
8          "datacenter_name": "Datacenter",
9          "datastore_name": "datastore1",
10         "resource_pool": "Resources"
11     },
12     "vm2": {
13         "host": "0.0.0.0",
14         "user": "root",
15         "pwd": "password",
16         "disable_ssl_verification": true,
17         "ova_path": "/home/ova/centos7.ova",
18         "datacenter_name": "Datacenter",
19         "datastore_name": "datastore1",
20         "resource_pool": "Resources"
21     },
22     "vm3": {
23         "host": "0.0.0.0",
24         "user": "root",
25         "pwd": "password",
26         "disable_ssl_verification": true,
```

*L'utilisation des autres modules est simple car ne nécessite pas de fichier de configuration.*

## Conclusion

Cette application Python offre une suite de fonctionnalités pour la gestion des machines virtuelles sur VMware ESXi. En combinant la puissance de Python avec l'API VMware, elle simplifie et automatise le déploiement et la gestion des machines virtuelles, offrant ainsi une solution flexible et efficace pour les administrateurs de systèmes et les professionnels de la virtualisation.