
Guide Utilisateur

Déploiement Automatisé de Machines Virtuelles Dans Vcenter



Documentation Technique

Réalisé par : Hasna Daoui & Nana Diawara

8 juin 2025

Table des matières

1	Vue d'ensemble	2
2	Prérequis	2
3	Installation	2
3.1	Structure du Projet	2
3.2	Installation Automatique	3
3.2.1	Étape 1 : Téléchargement	3
3.2.2	Étape 2 : Exécution du Script d'Installation	3
4	Configuration	3
4.1	Fichier de Configuration Principal	3
4.2	Paramètres de Configuration	4
4.2.1	Section vCenter	4
4.2.2	Section VM	4
5	Utilisation	5
5.1	Déploiement d'une Machine Virtuelle	5
5.2	Processus de Déploiement	5
5.3	Suivi et Journalisation	5
6	Dépannage	5
6.1	Erreurs Courantes	5
6.1.1	Erreur de Connexion vCenter	5
6.1.2	Template Non Trouvé	6
6.1.3	Ressources Insuffisantes	6
6.2	Commandes de Diagnostic	6
7	Personnalisation Avancée	6
7.1	Déploiement Multiple	6
7.2	Scripts Personnalisés	7
8	Bonnes Pratiques	7
8.1	Sécurité	7
8.2	Nommage	7
8.3	Maintenance	7
9	Support et Ressources	7
9.1	Fichiers de Log	7
9.2	Documentation VMware	8

1 Vue d'ensemble

Ce guide vous aidera à installer et utiliser le système de déploiement automatisé de machines virtuelles dans votre environnement VMware vSphere. Le système utilise des scripts Python pour cloner automatiquement des VMs à partir de templates, configurer les ressources et les déployer avec un minimum d'intervention manuelle.

Objectif

Automatiser le processus de création de machines virtuelles pour réduire les erreurs humaines et standardiser les déploiements.

2 Prérequis

Avant de commencer, assurez-vous d'avoir :

- **Python 3.6 ou supérieur** installé sur votre système
- **Accès au serveur VMware vCenter** avec les permissions appropriées
- **Identifiants administrateur** pour vCenter
- **Templates de VM** déjà créés dans votre environnement vSphere
- **Accès réseau** à votre serveur vCenter

Attention

Vérifiez que vous disposez des droits suffisants pour créer et configurer des machines virtuelles dans vCenter.

3 Installation

3.1 Structure du Projet

Le projet s'organise selon l'arborescence suivante :

Listing 1 – Structure des fichiers

```
1 vm-deployment/  
2     config/  
3         vm-config.json           # Configuration principale  
4     scripts/  
5         deploy-vm.py             # Script principal de  
6     d p l o i e m e n t  
7         functions.py             # Fonctions utilitaires  
8     templates/  
9         vm-template-config.json  # Param tres du template  
10    logs/  
11        deployment.log           # Logs de d p l o i e m e n t  
12    requirements.txt             # D pendances Python  
13    install.py                   # Script d'installation  
14    README.md                    # Documentation
```

3.2 Installation Automatique

3.2.1 Étape 1 : Téléchargement

1. Téléchargez les fichiers du projet vm-deployment
2. Extrayez-les dans le répertoire de votre choix
3. Ouvrez un terminal/invoke de commande dans le répertoire du projet

3.2.2 Étape 2 : Exécution du Script d'Installation

Listing 2 – Commande d'installation

```
1 python install.py
```

Ce script va automatiquement :

- Vérifier votre version de Python
- Créer les répertoires nécessaires
- Installer les packages Python requis
- Vérifier la présence des fichiers de configuration

Résultat Attendu

Si l'installation réussit, vous verrez le message : "Installation terminée avec succès!"

```
PS C:\Users\cyber\Desktop\vm-deployment\scripts> cd ..
PS C:\Users\cyber\Desktop\vm-deployment> python install.py
=====
INSTALLATION DU SYSTEME DE DÉPLOIEMENT VM
=====

Vérification de Python...
✓ Python 3.13.0 détecté

Création des dossiers...
Création de la structure de dossiers...
✓ Dossier créé: config/
✓ Dossier créé: scripts/
✓ Dossier créé: templates/
✓ Dossier créé: logs/

Installation des dépendances...
Installation des dépendances...
Requirement already satisfied: pyvmon>=7.0.3 in c:\program files\python313\lib\site-packages (from -r requirements.txt (line 2)) (8.0.3.0.1)
Requirement already satisfied: requests>=2.28.0 in c:\program files\python313\lib\site-packages (from -r requirements.txt (line 3)) (2.32.3)
Requirement already satisfied: aio>=17.3 in c:\program files\python313\lib\site-packages (from pyvmon>=7.0.3->-r requirements.txt (line 2)) (1.17.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\program files\python313\lib\site-packages (from requests>=2.28.0->-r requirements.txt (line 3)) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in c:\program files\python313\lib\site-packages (from requests>=2.28.0->-r requirements.txt (line 3)) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\program files\python313\lib\site-packages (from requests>=2.28.0->-r requirements.txt (line 3)) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\program files\python313\lib\site-packages (from requests>=2.28.0->-r requirements.txt (line 3)) (2025.4.26)

[notice] A new release of pip is available: 24.2 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
✓ Dépendances installées avec succès

Configuration des exemples...
Configuration des fichiers d'exemple...
✓ Fichier de configuration principal trouvé

Test des modules...
Test des imports...
✓ Tous les modules requis sont disponibles

=====
✓ INSTALLATION TERMINÉE AVEC SUCCÈS
=====

Étapes suivantes:
1. Éditez config/vm-config.json avec vos paramètres
2. Exécutez: python deploy-vm.py
=====
```

4 Configuration

4.1 Fichier de Configuration Principal

Le fichier config/vm-config.json contient tous les paramètres nécessaires au déploiement. Voici un exemple de configuration :

Listing 3 – Exemple de vm-config.json

```

1 {
2   "vcenter": {
3     "host": "10.11.5.83",
4     "username": "administrator@vsphere.local",
5     "password": "Cyber@2025",
6     "port": 443
7   },
8   "vm": {
9     "name": "ServeurWeb-01",
10    "template": "vm1",
11    "datacenter": "Datacenter",
12    "datastore": "datastore1",
13    "cpu": 2,
14    "memory_mb": 4096,
15    "disk_gb": 20,
16    "network": "vlan_100",
17    "power_on": true
18  }
19 }
```

4.2 Paramètres de Configuration

4.2.1 Section vCenter

Paramètre	Description	Exemple
host	IP ou nom d'hôte du serveur vCenter	10.11.5.83
username	Nom d'utilisateur vCenter	administrator@vsphere.local
password	Mot de passe vCenter	Cyber@2025
port	Port de connexion (généralement 443)	443

4.2.2 Section VM

Paramètre	Description	Exemple
name	Nom de la nouvelle VM	ServeurWeb-01
template	Template existant à cloner	vm1
datacenter	Nom du datacenter cible	Datacenter
datastore	Emplacement de stockage	datastore1
cpu	Nombre de cœurs CPU	2
memory_mb	RAM en mégaoctets	4096
disk_gb	Taille du disque en gigaoctets	20
network	Nom du réseau/VLAN	vlan_100
power_on	Démarrage automatique	true

5 Utilisation

5.1 Déploiement d'une Machine Virtuelle

Une fois la configuration terminée, vous pouvez déployer une VM avec la commande suivante :

Listing 4 – Commande de déploiement

```
1 python scripts/deploy-vm.py
```

5.2 Processus de Déploiement

Le script effectue automatiquement les étapes suivantes :

1. **Connexion à vCenter** : Établissement de la connexion sécurisée
2. **Validation des ressources** : Vérification de l'existence du template et des ressources
3. **Clonage de la VM** : Création de la nouvelle VM à partir du template
4. **Configuration des ressources** : Attribution des CPU, mémoire et stockage
5. **Configuration réseau** : Connexion au réseau spécifié
6. **Démarrage** : Mise sous tension de la VM (si configuré)

5.3 Suivi et Journalisation

Tous les événements sont enregistrés dans le dossier `logs/` avec un nom de fichier horodaté :

Listing 5 – Exemple de fichier de log

```
1 logs/deploy_20250606_143022.log
```

6 Dépannage

6.1 Erreurs Courantes

6.1.1 Erreur de Connexion vCenter

Problème

"Impossible de se connecter au serveur vCenter"

Solutions :

- Vérifiez l'adresse IP/nom d'hôte du serveur
- Contrôlez les identifiants de connexion
- Assurez-vous que le port 443 est accessible

6.1.2 Template Non Trouvé

Problème

"Template 'nom-template' introuvable"

Solutions :

- Vérifiez que le template existe dans vCenter
- Contrôlez l'orthographe du nom du template
- Assurez-vous d'avoir les droits d'accès au template

6.1.3 Ressources Insuffisantes

Problème

"Ressources insuffisantes pour le déploiement"

Solutions :

- Vérifiez l'espace disponible sur le datastore
- Contrôlez les ressources CPU et mémoire du cluster
- Réduisez les spécifications de la VM si nécessaire

6.2 Commandes de Diagnostic

Pour vérifier l'installation :

Listing 6 – Vérification Python

```
1 python --version
```

Pour tester les dépendances :

Listing 7 – Test des modules

```
1 python -c "import ssl, json; from pyVmomi import vim; print('
  Modules OK')"
```

7 Personnalisation Avancée

7.1 Déploiement Multiple

Pour déployer plusieurs VMs, vous pouvez créer plusieurs fichiers de configuration :

Listing 8 – Configuration multiple

```
1 config/
2     vm-web-config.json
3     vm-db-config.json
4     vm-app-config.json
```

7.2 Scripts Personnalisés

Le fichier `functions.py` peut être étendu pour ajouter des fonctionnalités :

Listing 9 – Exemple d’extension

```
1 def configure_advanced_settings(vm, settings):
2     """Configure des param tres avanc s de la VM"""
3     spec = vim.vm.ConfigSpec()
4     for key, value in settings.items():
5         opt = vim.option.OptionValue()
6         opt.key = key
7         opt.value = value
8         spec.extraConfig.append(opt)
9
10    task = vm.ReconfigVM_Task(spec=spec)
11    return task
```

8 Bonnes Pratiques

8.1 Sécurité

- Ne stockez jamais les mots de passe en clair dans les fichiers de configuration
- Utilisez des variables d’environnement pour les informations sensibles
- Limitez les permissions des fichiers de configuration

8.2 Nommage

- Adoptez une convention de nommage cohérente pour vos VMs
- Incluez des informations sur l’environnement (dev, test, prod)
- Utilisez des préfixes/suffixes descriptifs

8.3 Maintenance

- Sauvegardez régulièrement vos fichiers de configuration
- Documentez les modifications apportées aux scripts
- Testez les déploiements dans un environnement de test

9 Support et Ressources

9.1 Fichiers de Log

En cas de problème, consultez toujours les fichiers de log dans le dossier `logs/`. Ils contiennent des informations détaillées sur chaque étape du processus.

9.2 Documentation VMware

Pour des informations avancées sur l'API VMware, consultez :

- Documentation officielle pyVmomi
- Guide de l'API vSphere
- Forums de la communauté VMware

Félicitations

Vous êtes maintenant prêt à utiliser le système de déploiement automatisé de machines virtuelles !

```
PS C:\Users\cyber\Desktop\vm-deployment\scripts> python deploy-vm.py
2025-06-03 16:06:39,422 - INFO - Connexion à vCenter...
2025-06-03 16:06:39,460 - INFO - Recherche du datacenter...
2025-06-03 16:06:39,467 - INFO - Recherche du template...
2025-06-03 16:06:39,481 - INFO - Recherche du resource pool...
2025-06-03 16:06:39,488 - INFO - Création de la spécification de clonage...
2025-06-03 16:06:39,489 - INFO - Clonage de la VM...
2025-06-03 16:06:40,508 - INFO - VM clonée avec succès: New-VM-Test
2025-06-03 16:06:40,509 - INFO - Démarrage de la VM...
2025-06-03 16:06:41,517 - INFO - VM démarrée avec succès.
2025-06-03 16:06:41,517 - INFO - Déconnexion de vCenter.
PS C:\Users\cyber\Desktop\vm-deployment\scripts>
```

