Moon

- Moon Cheat Sheet

? Moon Cheat Sheet

Moon est le **Task Runner officiel** de la plateforme Homelab.

Il fournit une interface unique permettant d'exécuter les tâches d'administration, de maintenance et de déploiement sans avoir à mémoriser les commandes complexes.

? Objectif

Moon simplifie l'utilisation de la plateforme en standardisant l'exécution des scripts et des playbooks.

Il gère automatiquement :

- l'exécution depuis la racine du dépôt ;
- les variables d'environnement nécessaires ;
- la configuration Ansible ;
- la standardisation des commandes ;
- la mise en cache des tâches lorsque cela est possible.

? Commandes principales

Toutes les commandes doivent être exécutées depuis la racine du dépôt.

```
cd ~/workspace/homelab-infra-monorepo
```

? Maintenance de la plateforme

Commande	Description
`moon run :apply-system_update-config`	Mise à jour des systèmes Linux (APT/DNF), redémarrages automatiques et maintenance.
`moon run :apply-cluster-restart`	Rolling Restart complet du cluster Kubernetes (Drain → Reboot → Uncordon).

Commande	Description
`moon run :apply-docker-prune`	Suppression des images Docker inutilisées sur les hôtes Docker.
`moon run :apply-network-config`	Déploiement de la configuration réseau (AdGuard Home, DHCP, DNS).

? Gestion des applications

Commande	Description
`moon run :apply-docker-registry-config`	Déploiement et mise à jour du Docker Registry.
`moon run :apply-bookstack-config`	Mise à jour de BookStack.
`moon run :apply-plex-config`	Configuration du serveur Plex.

? Mise à niveau du cluster Kubernetes

Via Moon

```
moon run :upgrade-k8s-cluster -- \
-e "target_version=1.3x.x-1.1"
```

Commande Ansible équivalente

```
ansible-playbook \
-i infra/ansible/inventory.yml \
infra/ansible/playbooks/upgrade-cluster.yml \
-e "target_version=1.3x.x-1.1"
```

? Ajouter une nouvelle tâche

Les tâches Moon sont définies dans les fichiers de configuration du dépôt.

Exemple :

```
tasks:

  ma-nouvelle-tache:

    command: |
      export ANSIBLE_ROLES_PATH=$(pwd)/infra/ansible/roles
      export ANSIBLE_HOST_KEY_CHECKING=False

      ansible-playbook \
      -i infra/ansible/inventory.yml \
      infra/ansible/playbooks/mon_playbook.yml

    options:
      runFromWorkspaceRoot: true
      interactive: true

    inputs:
      - infra/ansible/playbooks/mon_playbook.yml
```

? Débogage

Simulation (Dry Run)

```
moon run :ma-tache --dryRun
```

Forcer l'exécution

```
moon run :ma-tache --force
```

Cette option ignore le cache de Moon.

Afficher les tâches disponibles

```
moon project :
```

? Dépannage

Problème	Solution
La tâche est indiquée comme "cached"	Utiliser `--force`.
Erreur Ansible	Vérifier l'inventaire et les rôles.
Variables d'environnement manquantes	Exécuter la commande depuis la racine du dépôt.
Commande inconnue	Vérifier la définition dans la configuration Moon.

? Bonnes pratiques

- Utiliser Moon comme point d'entrée pour toutes les opérations manuelles.
- Ne pas exécuter directement les playbooks Ansible lorsqu'une tâche Moon existe déjà.
- Conserver des noms de tâches cohérents.
- Une tâche Moon = une responsabilité.
- Versionner systématiquement la configuration Moon dans Git.
- Préférer Moon pour les opérations répétitives afin de garantir une exécution homogène.

? Philosophie de la plateforme

Composant	Responsabilité
Git	Source de vérité
Moon	Point d'entrée utilisateur
Terraform	Provisionnement de l'infrastructure
AWX / Ansible	Configuration des systèmes
ArgoCD	Déploiement GitOps Kubernetes

Moon ne remplace pas Terraform, AWX ou ArgoCD. Il fournit une interface unique permettant d'orchestrer les opérations manuelles de la plateforme de manière simple et reproductible.

? Voir également

- Ansible
- Terraform

- ArgoCD
- Kubernetes
- Monitoring

```
```