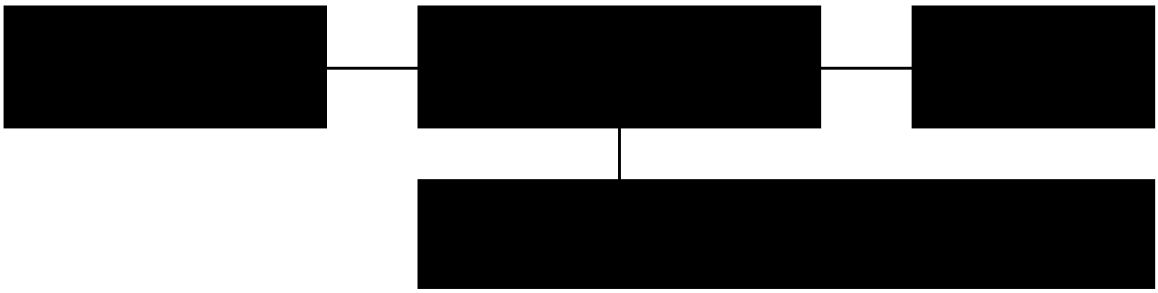
TD Avancé – Terraform & Ansible (2h30)

Niveau : 4e année Ingénieur
Durée : 2h30
Périmètre : Terraform (Docker provider) + Ansible (validation et hygiène minimale)

Contexte général

Vous intervenez comme ingénieur plateforme. L'objectif est de mettre en place un socle reproductible pour des environnements applicatifs conteneurisés. Terraform est utilisé pour provisionner des ressources Docker, tandis qu'Ansible intervient ensuite pour valider l'état du service et appliquer une hygiène minimale sans reconfigurer l'infrastructure.

Chaîne logique des outils



Principe fondamental : Terraform provisionne (image, réseau, container). Ansible valide et applique une hygiène minimale. Aucun outil ne déborde de son rôle.

Objectifs pédagogiques

• Concevoir un projet Terraform maintenable
• Provisionner un container Docker de manière reproductible
• Utiliser Ansible pour la validation et l'hygiène de base
• Comprendre la frontière Terraform / Ansible / runtime

Architecture cible

• 1 réseau Docker dédié
• 1 volume Docker (persistant)
• 1 container applicatif (nginx ou équivalent)
• Port exposé configurable (ex: 8080 → 80)
• Validation et hygiène via Ansible (local)

Découpage temporel (2h30)

• Terraform – structure & providers : 30 min
• Terraform – réseau, volume, container : 45 min
• Outputs & exploitation : 15 min
• Ansible – validation HTTP : 30 min
• Ansible – hygiène minimale (read-only checks) : 30 min

Structure du projet

terraform/
■ main.tf
■ variables.tf
■ outputs.tf
■ versions.tf
■ terraform.tfvars

ansible/
■ inventory.ini
■ validate.yml

Travail demandé – Terraform

Étape 1 – Base Terraform
1) Définir versions.tf (Terraform + provider Docker).
2) Déclarer les variables (image, container_name, external_port, environment).

Étape 2 – Provisionnement Docker
3) Créer un réseau Docker dédié.
4) Créer un volume Docker persistant.
5) Provisionner l'image et le container.
6) Exposer le port externe vers le port interne du service.

Étape 3 – Outputs
7) Exposer URL locale, ports, id et nom du container.

Travail demandé – Ansible

Étape 4 – Validation HTTP
8) Installer Ansible en local.
9) Créer un inventaire localhost.
10) Écrire un playbook validant que le service HTTP répond (module uri).

Étape 5 – Hygiène minimale
11) Vérifier l'état du container (running).
12) Vérifier l'exposition du port attendu.
13) Vérifier que le container redémarre automatiquement (policy).

■■ Ansible ne doit pas modifier la configuration du container.

Contraintes strictes

• Terraform ne contient aucun script de configuration applicative.
• Ansible ne modifie pas l'infrastructure Docker.
• Pas de local-exec / remote-exec pour configurer le service.
• Les checks Ansible doivent être idempotents et non intrusifs.

Questions (README)

1) Pourquoi séparer provisionnement et validation ?
2) En quoi les outputs Terraform facilitent l'automatisation ?
3) Quelle est la valeur d'Ansible dans un rôle non configurant ?
4) Comment ce socle évoluerait vers un environnement CI/CD ?

Livrables attendus

• Projet Terraform fonctionnel
• Playbook Ansible de validation/hygiène
• README expliquant les choix et limites
• Preuves de validation (outputs, curl, logs Ansible)