Projet de Fin de Module : Automatisation de l'Infrastructure avec Ansible pour KapsuleKorp

Date de soutenance : 01/12/2025

1. Contexte de l'Entreprise : KapsuleKorp

KapsuleKorp est une startup technologique en hyper-croissance spécialisée dans le développement de solutions logicielles innovantes. Face à l'expansion rapide de ses services, l'équipe technique est confrontée à des défis majeurs pour maintenir, déployer et mettre à l'échelle son infrastructure informatique. Les déploiements manuels actuels sont lents, sources d'erreurs et ne permettent pas de garantir la cohérence entre les différents environnements (développement, staging, production).

Pour répondre à ces enjeux, KapsuleKorp a décidé d'adopter une approche d'**Infrastructure as Code (IaC)**. L'objectif est d'automatiser entièrement le cycle de vie de son infrastructure, depuis le provisionnement des ressources jusqu'à la configuration des applications. Cette transition stratégique vise à améliorer l'agilité, la fiabilité et la sécurité de ses opérations.

2. Objectif du Projet

En tant que consultants DevOps juniors, votre mission est de concevoir et de mettre en œuvre une solution d'automatisation pour l'infrastructure de KapsuleKorp. Le projet se concentrera principalement sur l'utilisation d'**Ansible** pour la gestion de la configuration et le déploiement d'une application web standard.

Vous devrez structurer votre projet en suivant les meilleures pratiques, en créant des rôles réutilisables et en sécurisant les données sensibles. Une partie **bonus** portera sur l'utilisation de **Terraform** pour le provisionnement de l'infrastructure sous-jacente, illustrant ainsi un cycle IaC complet.

3. Architecture et Pile Technologique

L'infrastructure à déployer comprendra deux environnements distincts pour simuler un cycle de vie de développement professionnel :

- **Environnement Staging** : 2 serveurs web + 1 serveur de base de données
- **Environnement Production** : 3 serveurs web + 1 serveur de base de données

- A adapter si pas les ressources disponibles

La pile technologique à mettre en place est la suivante :

- **OS** : Ubuntu 22.04 LTS
- **Serveur Web** : Nginx
- **Base de Données** : MySQL 8.0
- **Langage** : PHP 8.1 avec PHP-FPM

4. Prérequis Techniques

Avant de commencer, assurez-vous que votre machine de contrôle dispose des outils suivants :

- **Ansible** : Version 2.9 ou supérieure.
- **Accès SSH** : Accès par clé SSH configuré vers les machines virtuelles cibles.
- **Python 3** : Doit être installé sur les machines cibles.

Pour la partie bonus, vous aurez également besoin de :

- **Terraform** : Version 1.0 ou supérieure.
- **VirtualBox** : Pour la virtualisation locale.
- **Plugin Terraform pour VirtualBox**.

5. Périmètre du Projet et Tâches Requises

5.1. Partie Principale : Configuration avec Ansible

L'objectif est de déployer une application web simple sur une pile LEMP (Linux, Nginx, MySQL, PHP) en utilisant Ansible.

Tâches Requises

1. **Initialisation du Projet**

- Créez une structure de répertoires pour votre projet Ansible en respectant les bonnes pratiques.
- Mettez en place un fichier d'inventaire (`inventory.ini`) pour définir vos environnements (`staging` et `production`) et les hôtes associés.

1. **Développement des Rôles Ansible** Vous devrez développer des rôles modulaires et réutilisables pour chaque composant de la pile applicative :

2. **Création du Playbook Principal**

- Rédigez un playbook principal (`site.yml`) qui orchestre l'application des différents rôles sur les hôtes cibles définis dans votre inventaire.

1. **Gestion des Données Sensibles**

- Utilisez **Ansible Vault** pour chiffrer toutes les informations sensibles, telles que les mots de passe de la base de données. Ces secrets seront stockés dans des fichiers de variables chiffrés et utilisés par vos playbooks.

5.2. Partie Bonus : Provisionnement avec Terraform

Cette partie est optionnelle et vise à démontrer votre compréhension du cycle IaC complet.

Objectif : Écrire un code Terraform pour provisionner automatiquement les machines virtuelles nécessaires pour les environnements `staging` et `production` en utilisant un provider local (ex: VirtualBox).

Tâches Requises

1. **Configuration Terraform**

- Rédigez un fichier `main.tf` pour définir les machines virtuelles à créer.
- Utilisez des variables (`variables.tf`) pour paramétrer le nombre d'instances, leur mémoire, etc.

1. **Génération d'Inventaire Dynamique**

- Configurez Terraform pour qu'il génère automatiquement le fichier d'inventaire Ansible (`inventory.ini`) une fois les machines provisionnées.

6. Livrables Attendus

- Un dépôt Git contenant l'intégralité de votre projet Ansible (et Terraform pour le bonus).
- Un fichier `README.md` ou word complet expliquant la structure, l'installation et l'utilisation de votre projet.
- Une démonstration fonctionnelle lors de la soutenance

7. Instructions d'Installation et d'Utilisation

Étape 1 : Cloner le Projet

```
Bash
```

```
git clone <url-du-depot>
cd projet-kapsulekorp-iac
```

Étape 2 : Configurer l'Inventaire

Modifiez le fichier `inventory.ini` pour correspondre aux adresses IP de vos machines virtuelles.

Étape 3 : Configurer les Variables Sensibles

Éditez le fichier `group_vars/all/vault.yml` avec vos secrets, puis chiffrez-le :

Bash

```
ansible-vault encrypt group_vars/all/vault.yml
```

Étape 4 : Déployer l'Infrastructure

Bash

```
ansible-playbook -i inventory.ini site.yml --ask-vault-pass
```

8. Déroulement de la Soutenance

- **Durée :** 15-20 minutes de présentation + 10 minutes de questions.
- **Contenu attendu :**
  1. Présentation du contexte et de l'architecture.
  2. Démonstration en direct du déploiement.
  3. Explication du code des rôles principaux.
  4. Démonstration de l'utilisation d'Ansible Vault.
  5. (Bonus) Démonstration du provisionnement avec Terraform.

10. Ressources

- Documentation Officielle d'Ansible
- Documentation Officielle de Terraform
- Bonnes Pratiques Ansible

Bon courage !