# Réponses aux Questions

## 4)

Pour configurer le mot de passe de l'administrateur de la base de données, la variable pwd=azerty a été définie dans le fichier ~/host\_vars/192.168.10.30. Proposez une méthode pour forcer le mot de passe à la valeur Admin1234.

### Réponse:

Pour définir le mot de passe de l'administrateur de la base de données à Admin1234, vous pouvez mettre à jour la variable Ansible dans le fichier host\_vars pour l'hôte 192.168.10.30.

Voici comment vous pouvez modifier le fichier ~/host\_vars/192.168.10.30 :

# ~/host\_vars/192.168.10.30  
pwd: Admin1234

Et assurez-vous que le playbook utilise cette variable pour configurer le mot de passe de la base de données. Voici un exemple de la tâche dans un playbook Ansible :

- name: Set database administrator password  
 hosts: database  
 become: yes  
 tasks:  
 - name: Configure database password  
 mysql\_user:  
 name: admin  
 password: "{{ pwd }}"  
 priv: '\*.\*:ALL,GRANT'  
 state: present

Dans cet exemple, le module mysql\_user est utilisé pour définir le mot de passe de l'administrateur de la base de données à la valeur spécifiée dans la variable pwd.

## 5)

Pour automatiser l'approvisionnement des machines, la DSI a opté pour l'utilisation de l'outil Terraform.  
a. Illustrez l'architecture et le fonctionnement de l'outil.  
b. Une fois les machines virtuelles déployées, on voudrait ajouter, en utilisant les mêmes fichiers .tf les volumes pour le stockage. Est-il possible de modifier une infrastructure déjà déployée par Terraform ? Justifiez.

### Réponse:

#### a) Architecture et Fonctionnement de Terraform

Terraform est un outil open-source d'infrastructure en tant que code qui permet de définir et de provisionner l'infrastructure de centre de données en utilisant un langage de configuration de haut niveau appelé HCL (HashiCorp Configuration Language).

##### Composants:

- Fichiers de configuration : Ils sont écrits en HCL et décrivent l'état souhaité de votre infrastructure.  
- Providers : Terraform interagit avec de nombreux fournisseurs de cloud comme AWS, Azure, et d'autres via des plugins appelés providers.  
- Fichier d'état : Terraform maintient l'état de votre infrastructure dans un fichier qui garde une trace des ressources gérées par Terraform.  
- CLI Terraform : L'interface en ligne de commande utilisée pour interagir avec Terraform et exécuter diverses commandes comme init, plan, apply et destroy.

##### Fonctionnement:

1. Écrire la configuration : Définir l'infrastructure dans des fichiers .tf.  
2. Initialiser : Exécuter terraform init pour initialiser le répertoire de travail contenant les fichiers de configuration.  
3. Planifier : Exécuter terraform plan pour créer un plan d'exécution, qui montre quelles actions seront entreprises sans apporter de modifications.  
4. Appliquer : Exécuter terraform apply pour exécuter les actions planifiées et créer ou modifier l'infrastructure.  
5. Gestion de l'état : Terraform maintient le fichier d'état pour suivre les ressources qu'il gère.

#### b) Modification de l’Infrastructure Déployée avec Terraform

Oui, il est possible de modifier une infrastructure déjà déployée avec Terraform. C'est l'une des fonctionnalités principales de Terraform. Vous pouvez mettre à jour les fichiers de configuration pour refléter les modifications souhaitées, puis exécuter terraform apply pour appliquer ces modifications.

Justification : Terraform utilise le fichier d'état pour suivre l'état actuel de votre infrastructure. Lorsque vous mettez à jour vos fichiers de configuration et exécutez terraform apply, Terraform compare l'état actuel avec l'état souhaité décrit dans les fichiers de configuration. Il génère ensuite un plan d'exécution pour aligner l'état actuel avec l'état souhaité. Cela vous permet d'ajouter, de modifier ou de supprimer des ressources dans votre infrastructure sans avoir besoin de gérer manuellement les modifications.