

Brief 4 : Terraform + Ansible + scaling = 🤯

Contexte du projet

Vous êtes de retour de vacances bien saluaires. Néanmoins, vos congés ont été ternis par le goût amer de l'imperfection du script de déploiement sur lequel vous avez travaillé avant de partir.

Alors dès votre reprise, vous creusez les tréfonds du Web à la recherche de la solution ultime. Au fur et à mesure de votre cheminement dans les méandres de l'Internet, vous remarquez des mots qui reviennent souvent : "Terraform", "Ansible", "Chef", "Puppet", peut-être même "Docker", mais pas "Kubernetes", non. Pas pour cette fois.

Ainsi, vous demandez-vous, serait-il plus efficace d'utiliser des outils conçus pour le déploiement automatisé plutôt que de réinventer la roue ?

Ces outils, utilisés par des milliers, des dizaines de milliers, peut-être même des centaines de milliers d'informaticiens de par le monde seraient-ils plus puissants que votre script ?

Un seul moyen de le savoir : essayer !

Vous vous sentez contrit, mais vous savez maintenant quoi faire pour expier vos péchés :

- Concevoir une infrastructure sécurisée, résiliente et scalable
- Utiliser Terraform et Ansible pour déployer
- Créer un Scale set pour configurer le scaling
- Monitorer l'application et ses ressources
- Sauvegarder les données de l'application
- Implémenter un script de test de charge
- Expérimenter le scaling
- Rédiger les documentations et présentations

Modalités pédagogiques

Tous les déploiements sont à faire en utilisant soit Terraform, soit Ansible.

Vous déploierez une des applications suivantes :

- Nextcloud,
- Jenkins,
- Gitea,
- Nagios,
- Wordpress,
- ou une application de votre choix (soumis à validation de Bryan ou d'Alfred).

Vous pourrez déployer des containers Docker si vous le souhaitez.

Toutes les données de l'application seront stockées en base de données ou dans un espace de stockage.

La méthode Scrum sera mise en place, avec un Scrum de Scrum tous les lundis.

Concernant la durée du brief, un point sera fait le 07/09 pour décider d'une semaine d'extension ou non.

Chapitre 0 : préparation

1. Préparer minutieusement votre plan projet comprenant :
 - une topologie de l'infrastructure,

- la liste des ressources Azure,
 - la liste des tâches à faire,
 - la stratégie de scaling,
 - les tests et métriques de monitoring,
 - le plan de test de charge,
 - la politique de backup.
2. Créer un projet Github
 3. Créer un dépôt Github associé au projet
 4. Reporter la liste des tâches dans le Kanban
 5. Restitution collective en groupe **jeudi matin**

Chapitre 1 : déploiement d'une infrastructure minimale

1. Déployer un resource group
2. Déployer un vnet
3. Déployer un subnet
4. Déployer une IP publique avec une étiquette DNS (DNS label)
5. Déployer une VM linux
6. Se connecter en SSH sur la VM pour exécuter une commande

Chapitre 2 : (si nécessaire) déploiement d'une base de données

1. déployer une instance Azure Database for MariaDB (ou PostgreSQL)
2. Créer une règle de firewall pour autoriser la VM application
3. installer sur la VM applicative le paquet mysql-client (ou postgresql-client-14) et utiliser la commande mysql pour tester la connection à la base de données

BONUS :

4. Créer un private endpoint connection pour la VM applicative au lieu d'une règle de firewall

Chapitre 3 : (si nécessaire) déploiement d'un espace de stockage

1. Créer un compte de stockage
2. Créer un espace de stockage SMB avec un quota de 5GO
3. Ajouter une règle ACL pour autoriser la VM applicative
4. installer sur la VM applicative le paquet cifs-utils et tester le montage de l'espace de stockage

BONUS :

5. Créer un private endpoint connection pour la VM applicative au lieu d'une règle ACL

Chapitre 4 : script cloud-init

1. Implémenter un script cloud-init qui installe et configure l'application choisie
2. Vérifier que l'application est bien disponible depuis le web en utilisant le FQDN de la VM

Chapitre 5 : déploiement d'une application gateway

1. Créer un subnet dédié
2. Déployer l'application gateway en utilisant l'adresse IP publique de la VM applicative. Le backend est la VM applicative.

Chapitre 6 : mise en place de TLS

1. Créer un Azure Key Vault
2. Créer un compte de stockage public
3. Créer un conteneur **de stockage**
4. Dans ce conteneur **de stockage**, créer un chemin `"/.well-known/acme-challenge"`
5. Créer un blob avec un contenu aléatoire dans ce chemin
6. Configurer une règle de chemin sur l'Application gateway qui redirige `"/.well-known/acme-challenge/*"` vers le conteneur **de stockage**
7. Tester l'URL depuis un navigateur web
8. Implémenter un script qui :
 1. crée un certificat via Certbot avec le challenge http-01
 2. qui crée un blob dans le conteneur **de stockage** avec le bon nom et le bon contenu pour résoudre le challenge.
 3. Ajoute le certificat dans le vault
9. Vérifier via votre navigateur que le site est bien considéré comme sûr

BONUS :

10. Utilisez le service Azure Automation pour vérifier la date d'expiration du certificat et le renouveler 10 jours avant expiration.

Chapitre 7 : Monitoring de l'application

1. Activer le monitoring de la VM dans Azure Monitor
2. Activer le cas échéant le monitoring de la base de données et/ou de l'espace de stockage dans Azure Monitor
3. Activer le monitoring de l'application dans Azure Monitor
4. Créer un workbook (classeur) avec les métriques pertinentes à suivre pour vos ressources et votre application
5. Créer une alerte en cas d'indisponibilité de l'application
6. Créer une alerte en cas d'usage CPU > 90% sur la VM applicative
7. Créer une alerte si la date d'expiration du certificat TLS est < 7 jours
8. Créer une alerte si l'espace disponible sur l'espace de stockage < 10%
9. Configurer les alertes pour envoyer des emails à votre groupe et aux formateurs

Chapitre 8 : script de test de montée en charge

1. Implémenter un script qui effectue des requêtes HTTP à l'application de manière massive et rapide.
2. Constater l'impact du script sur l'usage des ressources de la VM applicative.

BONUS :

1. Créer une ressource Azure Load Testing pour effectuer un test de charge
2. Créer un Quick test
3. Exécuter le test et constater l'impact du test sur l'usage des ressources de la VM applicative

Chapitre 9 : backup

1. Créer un backup de l'espace de stockage le cas échéant
2. Créer un backup de la base de données le cas échéant
3. Supprimer des données dans la base et/ou dans l'espace de stockage

4. Vérifier que l'application ne fonctionne plus
5. Attendre 10 minutes
6. Restaurer le backup
7. Vérifier que l'application fonctionne à nouveau
8. Vérifier que la défaillance apparaît dans Azure Monitor

Chapitre 10 : scale set



Échange des projets entre équipes sur base d'un transfert de dépôt Git et de documentation. Une réunion de passation d'une durée maximale de 30 minutes sera à planifier. Ceux qui ont choisi Terraform récupéreront un projet Ansible et vice-versa.

1. Créer un VM scale set pour la VM applicative
2. Exécuter un scale out de 8 replica
3. Exécuter le test de montée en charge
4. Constater l'impact du test sur les VMs
5. Exécuter un scale in de 2 replica
6. Exécuter le test de montée en charge
7. Constater l'impact du test sur les VMs

Chapitre 11 : auto scale

1. Configurer le profile d'autoscale du scale set
2. Créer une règle de scale out
3. Créer une règle de scale in
4. Exécuter le test de montée en charge
5. Constater l'impact du test sur les VMs

Critères de performance

- Le code est lisible (clair et facilement compréhensible)
- Les scripts s'exécutent correctement et déploient automatiquement l'application
- L'application est correctement monitorée et sauvegardée
- Le plan exécuté est proche du plan prévu
- la méthode Scrum a été suivie
- la passation de votre travail au chapitre 10 a donnée satisfaction

Modalités d'évaluation

Restitution en groupe.

Relecture commentée de vos livrables par les formateurs.

Livrables

- documentation des infrastructures (Terraform et Ansible)
- lien vers les dépôts des sources
- un executive summary de votre travail (technique et organisationnel) à présenter en groupe
- dans votre executive summary, vous présenterez le fonctionnement de Terraform ou d'Ansible
- votre avis sur la réception du travail de l'autre groupe

Objectifs

À l'issue de ce brief, vous aurez :

- mieux anticipé vos actions que dans le brief précédent
- déployé de nouvelles ressources dans Azure
- implémenté des scripts
- utilisé Terraform
- utilisé Ansible
- écrit du YAML
- documenté votre travail
- présenté votre travail
- pratiqué Scrum
- abordé la gestion de projet
- abordé la communication en équipe
- utilisé certbot pour déployer un certificat TLS
- sauvegardé les données applicatives
- monitoré l'application et la VM
- configuré de l'alerting
- scalé une application
- subi le travail des autres