

# Brief 3 : Déploiement scripté d'une application

---

## Contexte du projet

---

Félicitations ! Vous avez brillé auprès de votre chef lors du brief précédent.

Il ne jure plus que par vous pour traiter les demandes d'installation d'applications.

Ce matin, à votre arrivée au bureau, vous recevez cet email :

--- From: [le.chef@simplon.co](mailto:le.chef@simplon.co) ---

Bonjour,

Tu as bien travaillé pour le Nextcloud de Lifesense, merci !

J'ai regardé les demandes de déploiements d'applications et j'ai remarqué qu'il y a beaucoup de tickets qui concernent ces 3 applications : Nextcloud, Gitea et Jenkins.

On pourrait gagner beaucoup de temps à les automatiser !

Est-ce que tu peux en choisir une et préparer un script de déploiement complètement automatisé ?

Je compte sur toi.

Cordialement,

Le chef

Pour répondre à cette demande, qui pourrait bien déboucher sur une promotion, vous allez :

1. Concevoir une infrastructure en respectant le référentiel de conformité Azure de l'entreprise
2. Programmer des scripts de déploiement
3. Rédiger les documentations d'infrastructure et d'usage de vos scripts

Vous choisirez de faire vos scripts en PowerShell, en Bash ou en Python.

Vous partirez du principe que la CLI Azure est correctement installée et configurée sur la machine qui exécutera les scripts, mais sans extension.

**Aucun autre prérequis ne sera réputé acquis.**

## Modalités pédagogiques

---

:::danger

⚠ **NE RIEN DÉPLOYER AVANT LA RESTITUTION EN GROUPE (ÉTAPE 2)**

:::

:::warning

Vous utiliserez une stratégie de branching pour votre dépôt GIT

:::

En groupe de 3 personnes, vous allez :

1. Préparer minutieusement votre plan projet comprenant :
  - une topologie de l'infrastructure,
  - la liste des ressources Azure que vous prévoyez de déployer,
  - la liste des tâches à faire que vous avez prévu.

2. Restitution collective en groupe
3. Exécution du plan projet
4. Recette : déploiement de l'application sur une infrastructure de test en utilisant les scripts
5. Publication des scripts et documentations sur un dépôt Github

Pour ce brief, le travail sera organisé en suivant la méthode agile SCRUM. Le brief durera le temps d'un sprint de 2 semaines. Pour cela, chaque groupe :

- choisira un scrum master
- organisera un sprint planning
- organisera chaque jour un stand-up meeting
- organisera un sprint review et une rétrospective
- tiendra un document de suivi de projet

Il sera aussi organisé un scrum de scrum. Pour cela :

- un scrum de scrum master sera choisi
- il organisera deux scrum de scrum meetings par semaine

Les product owners seront Bryan et Alfred.

## Critères de performance

---

- Le code est lisible (clair et facilement compréhensible)
- Les scripts s'exécutent correctement et déploient automatiquement l'application
- Le référentiel de conformité Azure a été strictement respecté
- Le code est lisible
- Le plan exécuté est proche du plan prévu
- la méthode Scrum a été suivie
- Le code est lisible

### BONUS

- les paramètres suivants seront configurables via arguments aux scripts :
  - Le dimensionnement de la VM applicative
  - Le nom de la machine
  - La localisation des ressources Azure
  - Le dernier sous-domaine du FQDN à utiliser
- En cas d'erreur, le script sera capable de rollback automatiquement (il supprimera toutes les ressources qu'il a créées)
- L'application et sa base de données sont sauvegardées en utilisant le service Backup de Azure

## Modalités d'évaluation

---

Restitution intermédiaire en groupe.

Restitution finale individuelle.

Relecture commentée de vos livrables par le formateur.

## Livrables

---

- Documentation de l'infrastructure

- Documentation d'usage des scripts (intégrée aux scripts ou à part)
- Le ou Les scripts
- Un executive summary de votre travail (technique et organisationnel) à présenter en individuel devant le chef (slides)
- Les plans projets prévu et exécuté

## 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 un bastion
- mis en place un moyen de surveillance de la disponibilité de l'application
- configuré la rétention des logs
- parsé du JSON
- présenté votre travail
- pratiqué Scrum
- abordé la gestion de projet
- abordé la communication en équipe
- utilisé certbot pour déployer un certificat TLS

### BONUS

- créé une CLI
- géré le rollback en cas d'erreur
- utilisé les services Azure Backup vault et Backup pour sauvegarder une VM et une base de données
- défini une politique de sauvegarde

# Référentiel de conformité Azure

---

La DSI, en collaboration avec la RSSI, a établi les contraintes suivantes, à suivre **impérativement**, pour utiliser le cloud Azure :

- L'usage d'un bastion Azure est obligatoire. Aucune VM ne sera accessible en administration à distance sans passer par le bastion.
- L'authentification SSH sera par clés dès le déploiement de la VM (pas de passage par un mot de passe).
- Chaque VM devra avoir un disque dédié au système d'exploitation. Les données applicatives devront donc être stockées sur un disque différent.
- Le disque de données devra être chiffré. La clé sera gérée par Azure.
- Les accès publics seront réduits au strict minimum.
- Tous les logs devront être conservés pendant 1 an sur la machine.
- L'applicatif devra être monitoré en utilisant le service Application Insight.
- Si une base de données est requise, vous utiliserez un des services Azure pour l'implémenter.
- Pour des raisons de coûts, il est demandé d'ajuster les ressources au strict nécessaire.
- Les accès administrateurs devront être **nominatifs et individuels**.
- Les applications HTTP devront être configurées en TLS. Si aucun certificat n'est fourni par le client, un certificat *Let's encrypt* sera utilisé.
- Le certificat *Let's encrypt* devra être automatiquement renouvelé.

## BONUS

- Les VM et les bases de données seront sauvegardées quotidiennement dans une période creuse avec une rétention de 14 jours.