

# Cahier des charges - Projet Services Web / Wordpress



DataScientest . com

Etape	Description	Objectif	Modules / Masterclass / Templates	Conditions de validation de l'étape intermédiaire
1 : Cahier des charges	Construire le cahier des charges et prise en main de l'application	Comprendre les objectifs du projet et les exigences fonctionnelles. <ul style="list-style-type: none"><li>Définition des besoins métiers et de l'application utilisée.</li><li>Importation de l'application sur votre VCS</li><li>Schéma d'implémentation</li><li>Instancier un dépôt Git : l'organiser et le documenter</li><li>Organiser la gestion du projet : Choix des méthodes</li></ul>	<ul style="list-style-type: none"><li>Documentation externe</li><li>GitHub</li><li>Trello</li></ul>	Cahier des charges Format PDF

## Rappels sur les attentes du projet

### ➤ **Services Web**

L'idée de ce projet est de reprendre les **livrables de votre examen Linux** c'est à dire :

- **2 serveurs Web** (Apache ou NGINX)
- Avec leurs **CMS** respectifs (Wordpress, Prestashop, Magento...)
- Et son **Load Balancer**

Et de faire évoluer cette solution pour la déployer dans **Kubernetes**. On pourra se servir d'un provider Cloud comme AWS pour déployer l'application.

Cette application pourra évoluer au fur et à mesure de votre avancement lors de la formation :

- Construire les **Dockerfiles** pour chaque composant de l'application
- Déployer cette application dans un environnement de test et s'assurer qu'elle fonctionne. On pourra créer des simples **tests unitaires** pour s'en assurer
- Orchestrer ces conteneurs via **Kubernetes**
- **Automatiser** le déploiement et les changements via un **pipeline** (Codepipeline, GitHub Actions, GitLab CI, Jenkins...)
- Créer des **templates IaC** (Infrastructure As Code) pour déployer l'infrastructure de manière automatisée
- Déployer la solution en **production**

## **Cahier des charges pour le déploiement d'une infrastructure WordPress en Production**

### **1. Objectifs et contexte du projet :**

- Ce projet consiste à déployer une infrastructure de Services Web
- Le client souhaite une infrastructure qui devra garantir la haute disponibilité, la scalabilité, la performance (repartitions des la charge) et la sécurité de la plateforme.
- Il conviendra de choisir la solution permettant de porter un déploiement de la solution (As Code), donc automatisée.
- A ce stade le client dispose d'une infrastructure locale, constituée de

4 containers Docker, (1 pour la répartition avec NGINX, 1 pour la base MariaDB, et 2 pour Wordpress)

- Nous envisageons une solution cible portée sur AWS.
- Il disposera d'environnements distincts (DEV, Préprod et prod) et d'un pipeline CI/CD
- *L'architecture cible (sous AWS), n'est pas figée à ce stage : les stratégies et les services AWS possibles sont à débattre. Elle pourra évoluer selon l'avancement de la formation DevOps.*

## 2. Spécifications techniques :

- Usage de ECR (Elastic Container Registry)
- Usage de EC2 pour réaliser des namespaces par environnements.
- L'infrastructure devra être deployable as Code (IaC)
- Terraform portera le build pour définir et implémenter sous AWS :
  - - la sécurité des accès (IAM),
  - - les certificat SSL
  - - le provisionnement VPC, et des composants/services réseaux
  - - les zones,
  - - la répartition de charge (avec ALB ? ELB ? )
  - - les bases de données Wordpress (Postgres ? Maria DB ? ou Aurora)
  - - le stockage persistant pour les fichiers Wordpress (si EC2 /DB MySQL)
  - - les besoins spécifiques sur la sauvegarde des données (si Aurora)
  - - la stratégie de sauvegarde si (db <> Aurora)
  - - le provisionnement des Kubernetes via EKS permettant la scalabilité, et le cryptage des données
  - - le provisionnement des EC2 requis pour les pods qui seront groupés dans des namespaces par environnements
  - - les groupes de sécurité EC2 pour contrôler les accès au serveur web et à la base de données.

## 3. Sécurité :

### Les Accès

- Définition de comptes utilisateurs et de rôles IAM pour accéder aux services AWS
- Les ressources ECR seront accessibles selon les politiques/rôles IAM définis
- Mise en place d'un Vault intégré avec :
  - - IAM pour l'authentification (utilisateurs et applications) et pour limiter les privilèges d'accès aux ressources AWS
  - - ECR pour stocker les secrets permettant d'accéder aux images

Docker

- Configuration d'un certificat SSL/TLS pour garantir le chiffrement des données transitant entre les utilisateurs et le serveur web.
- Configuration d'alertes de sécurité pour être informé des activités suspectes.

### **Les données**

- Toutes les données DB devront être stockées de manière sécurisée et distribuée afin de couvrir la tolérance de panne et la haute disponibilités.
- Application des correctifs de sécurité réguliers pour le système d'exploitation, les logiciels serveurs et WordPress lui-même.
- Utilisation de pare-feu d'application web (WAF) pour filtrer le trafic HTTP/HTTPS et se protéger contre les attaques web.

### **4. Performances :**

- Dimensionnement adéquat des ressources pour répondre aux besoins de charge du site web.
- Utilisation d'un équilibreur de charge
- Optimisation des performances du serveur web, de la base de données et de la mise en cache pour garantir une expérience utilisateur fluide.
- CloudFront pour la mise en cache et distribuer la partie statique du site wordpress

### **5. Gestion des sauvegardes :**

- Mise en place d'une stratégie de sauvegarde régulière des données, avec une rétention appropriée. *(Selon le choix de stockage DB retenu)*
- Test régulier de la restauration des sauvegardes pour garantir leur efficacité en cas de besoin. *(Selon le stockage DB retenu et la stratégie de développement du client)*
- Configuration de sauvegardes régulières des fichiers WordPress. *(Selon le choix du stockage retenu)*

### **6. Documentation et Organisation :**

- Documentation détaillée de l'architecture de l'infrastructure, des configurations et des procédures de maintenance.
- Instructions pour le déploiement de mises à jour et la résolution des incidents courants.
- L'espace collaboratif Gitlab <https://gitlab.com/tp-datascientest/wordpress> portera les développements et toutes les documentations autour de ce projet
- La répartition des tâches et leur suivi, seront en place via trello ou Azure DevOps (à définir)

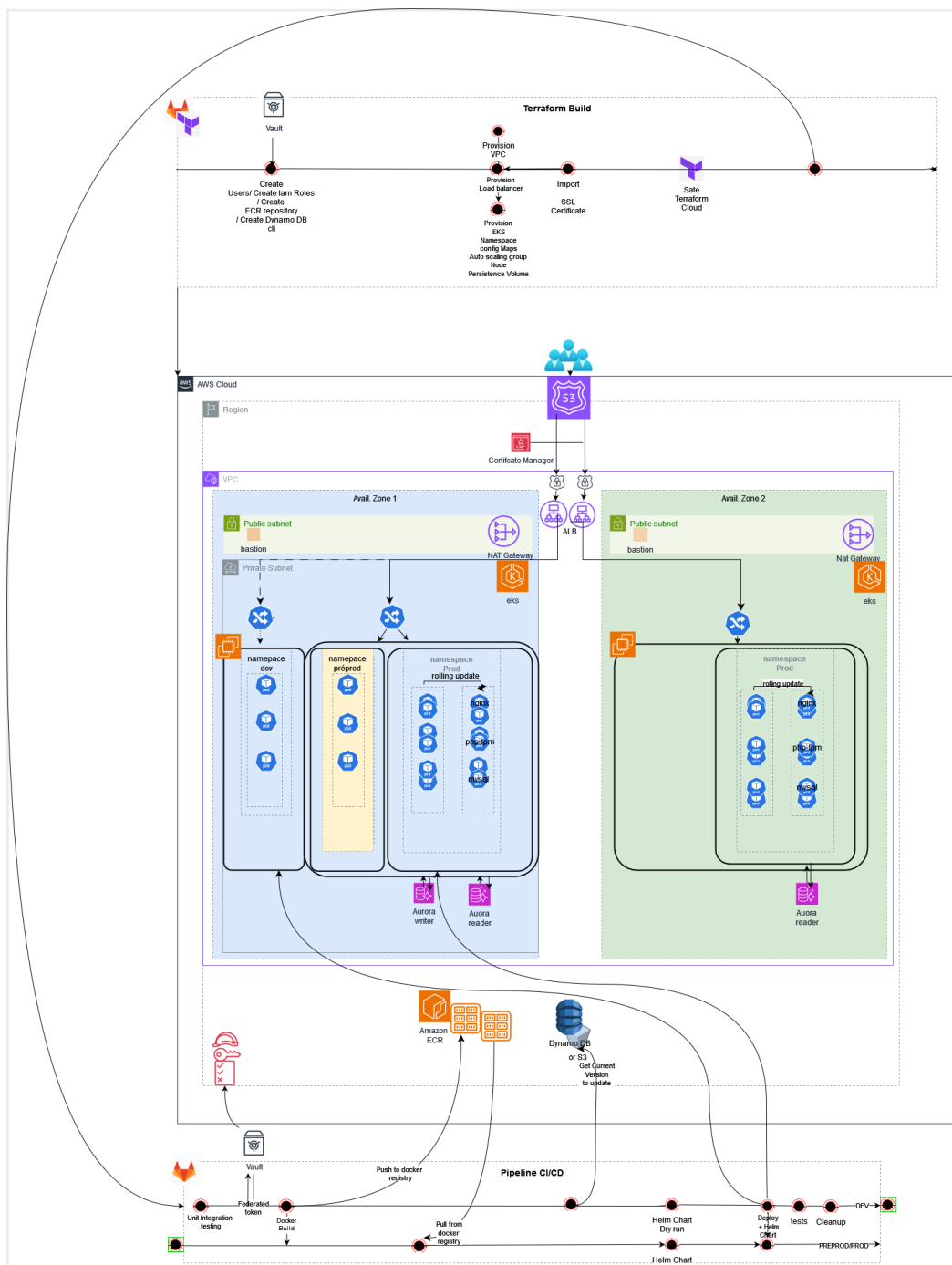
## **7. Contraintes de délai :**

- Le déploiement de l'infrastructure doit être réalisé dans un délai d'un mois.
- Une Sandbox AWS est requise pour démarrer les travaux. (*Demande à formuler à DST après validation du groupe*)
- En raison des facturations AWS, les services envisagés seront à considérer selon leur pertinence, mais surtout en fonction du budget fourni par le client.

## **8. Suivi et support :**

- Mise en place d'un suivi continu de la performance, de la sécurité et de la disponibilité de l'infrastructure.
- AWS CloudWatch pourra surveiller des métriques de performance
- Disponibilité d'un support technique pour répondre aux incidents et aux demandes d'assistance.

## **Architecture cible(à valider)**



S