

Retour d'expérience : héberger une application Vue.js via des buckets S3 configurés en mode « website »

mercredi 8 octobre 2025 08:45 (45 minutes)

Auteure : Pauline Gilg

Université de Strasbourg, Direction du numérique, Services métiers, Développeuse front end au sein du pôle Développement, intégration et paramétrage (DIP)

Pauline Gilg est développeuse front end, avec une formation en linguistique, spécialisée en études japonaises et en localisation. Elle a évolué dans le domaine de la communication numérique en tant que webmaster avant de se spécialiser dans le développement web. Ses intérêts portent principalement sur l'expérience utilisateur et l'accessibilité numérique, avec une attention particulière aux standards du web et à la conception d'interfaces inclusives. Elle travaille également sur des problématiques de déploiement, en mettant en place des pipelines d'intégration continue (CI) et des processus d'automatisation pensés pour être facilement utilisables, notamment dans des contextes multi-locataires ou institutionnels. Son parcours croise ainsi des compétences en sciences humaines et en technologies du numérique, qu'elle mobilise dans le cadre de projets orientés utilisateur.

Résumé

Campulse est une plateforme web développée par l'Université de Strasbourg à destination des associations étudiantes et des Services de la Vie Universitaire (SVU), centralisant annuaire, signature de chartes, demandes de subventions et suivi administratif. Financé dans le cadre du programme Services numériques aux étudiants (France Relance), le projet est en cours de déploiement dans 11 universités et suscite un intérêt national.

L'architecture repose sur un découplage strict entre front end et back end. Le front, développé en Vue.js, consomme une API REST exposée par un back end Django. Ce découplage a permis de faire un choix d'hébergement spécifique pour l'interface : le mode static website hosting de S3, optimisé pour la diffusion de fichiers HTML, CSS et JavaScript.

L'objectif principal : permettre à chaque université d'adapter l'interface à son identité (images, couleurs, textes, polices), de manière totalement autonome, tout en conservant une solution automatisée, scalable et maintenable. Cette personnalisation est injectée à la compilation via des scripts de parsing et des variables, puis buildée en une instance front end autonome.

Une CI (intégration continue) basée sur GitLab CI orchestre le processus :

- Build de l'application en fonction des personnalisations,
- Création à la volée du bucket S3 si besoin (un par environnement : préproduction et production),
- Déploiement automatisé des fichiers statiques.

Chaque bucket S3 est configuré en mode « website » pour servir directement les fichiers, avec :

- Définition explicite des MIME types pour assurer le bon rendu dans les navigateurs,
- Paramétrage des Access Control Lists (ACL) pour exposer uniquement les fichiers requis.

L'accès public aux interfaces est exposé via des objets VirtualService gérés par Istio et déployés sur l'infrastructure cible.

Ce choix technique présente plusieurs avantages :

- Scalabilité horizontale : chaque établissement dispose de son propre bucket déployé automatiquement, sans incidence sur les autres.
- Sécurité : aucun accès SSH nécessaire ; seules des clés d'accès S3 sont utilisées, réduisant la surface d'attaque.
- Simplicité d'administration : une console S3 permet un accès rapide à la liste des buckets et à leur contenu.
- Pertinence technique : les fichiers buildés sont statiques et non sensibles.

Ce retour d'expérience propose donc un modèle d'hébergement statique, découplé, sécurisé et multi-locataire, particulièrement adapté aux interfaces front end dans un contexte public ou institutionnel. L'approche allie automatisation, personnalisation et robustesse, tout en s'intégrant

dans des pipelines DevOps modernes.

Orateur: Mme GILG, Pauline (Université de Strasbourg)