Civiparoisse : projets d’évolutions lourdes

(SOLR pour les adresses postales, envoi de mails via AWS)

CivisolrAddr : Problématique

- Numéro, répéteur et nom de voie sont saisis sur la même ligne
- Erreurs de saisies, incohérences dans les données => à corriger et à normaliser => solution curative
- Idéalement : aussi éviter les erreurs de saisie => solution préventive
- Besoin de créer des listes de distribution physique (ex : Nouveau Messager)

Source de données : Base des Adresses Nationale

- données : https://adresse.data.gouv.fr/data/ban/adresses/latest/csv
- schéma de données : https://github.com/BaseAdresseNationale/adresse.data.gouv.fr/blob/master/public/schemas/adresses-csv.md
- Licence:
  - https://www.data.gouv.fr/fr/datasets/base-adresse-nationale/
  - https://www.etalab.gouv.fr/wp-content/uploads/2017/04/ETALAB-Licence-Ouverte-v2.0.pdf => indiquer source et date de récupération
- données mises à jour quotidiennement par département
- UEPAL : Alsace (67 Bas-Rhin, 68 Haut-Rhin), Lorraine (54 Meurthe-et-Moselle, 55 Meuse, 57 Moselle, 88 Vosges) => 6 départements
- Taille des données limitée : environ 300Mo

Contraintes d’exploitation des données

- Mise à disposition centralisée de données à jour
- exploration avancée des données : recherche sur des débuts de mots, des variantes de mots, voire des synonymes (ex : Pl : Place, St : Saint), correction orthographique
- exploration hiérarchique des données : par département, ville, nom de voie, numéro, répéteur
- mise à jour de données non bloquante

Moteur de recherche SOLR

- notion de schéma de données, avec traitement différenciable document/ requête
- en natif : intégration des données en CSV, spellcheck, et faceting
- mise à jour des données : core swapping, collection alias
- image Docker communautaire officielle
- Solr Operator : trop complexe pour le besoin courant
- Solr dispose d'un mécanisme d'authentification basé sur des certificats SSL (intégration via cert-manager et trust-manager sous K8S)
- gestion des rôles avec des droits précis et de les mapper sur des utilisateurs (ex : admin, indexeur, requêteur)
- Mise en oeuvre d'un proxy forward SSL
  pour les requêtes de recherche (ex : socat)

Utilisation curative

- si on n'a pas le département, on se base sur le code postal pour déduire le département, puis on vérifie la ville, et enfin on cherche le reste du contenu de l'adresse
- le résultat est stocké dans la BD locale, avec un état (unchecked, valid, invalid, stale), dans une table, avec une entité d'Api v4
- on propose ce résultat à l'utilisateur, qui le valide ou non ; lorsque l'adresse change, le résultat passe à l'état stale avant de passer à unchecked lors de la recherche suivante
- on profite de l'api v4 pour exploiter les résultats via des rapport searchkits
- on exploite le DisplayPaged pour la sortie papier

Utilisation préventive (à creuser)

- On fournit un widget qui pourra proposer une recherche hiérarchique sur l'adresse (ex : ouverture d'un modal)
- CiviCRM propose un type de données qui est le chainSelect, avec un ControlField : piste assez vraisemblable
- CiviCRM propose aussi l'utilisation du widget select2 (select2.org), qui peut utiliser des sources AJAX et fournit de l'autocomplétion sur les listes de données
- La solution reposera donc sur de l'AngularJS et de l'Afform.

Envoi de mails : problématiques

- Envoi de mails pour transmettre les notifications des services (AWS)
- Envoi de mails de masse (AWS)
  - idéalement, isoler les paroisses les unes des autres vis à vis des mesures antispams
  - automatisation de la configuration
  - traitement des retours de mails
  - adresse fournie simple et lisible
- Réception de mails sur l’adresse d’envoi (OVH)

Pourquoi Amazon Web Services (AWS) ?

- Un nombre restreint de fournisseurs
  - externalisation nécessaire de la solution
  - fonctionnalité d'envoi de mails en masse pas disponible chez tous les fournisseurs
  - gestion de comptes multiples => segmentation des risques
- Une contrainte budgétaire :
  - coût en fonction de l'utilisation/ réservation des ressources
  - les tarifs catalogue d'autres fournisseurs semblent moins compétitifs
  - Éventuelles réductions via techsoup
- Un service à construire soi-même sur mesure :
  - documentation disponible : guides de référence, Skill Builder…
  - intégration possible de la solution avec K8S
  - provisionning de type Infrastructure as Code

Utilisation préssentie des services AWS

- Catégorisation des services :
  - services structurels
  - services de monitoring
  - services de provisionning
  - services fonctionnels

Services structurels

- AWS Identity and Access Management (IAM) (utilisateurs, groupes, rôles…
- AWS IAM Roles Anywhere (Interfaçage X509)
  - https://github.com/aws/rolesanywhere-credential-helper
- AWS Organization (Compte comme container de sécurité)
- AWS Control Tower (Best practices)

Services de monitoring

- AWS Budgets : surveillance des coûts du cloud
- AWS Cost Explorer : analyse des coûts
- AWS CloudTrail : audit / centralisation de certains logs (dont Lambda et S3)
- AWS CloudWatch Logs : service de logs
- AWS CloudWatch : métriques et déclenchement d'actions sur les métriques

Services de provisionning

- AWS CloudFormation : service de déploiement de ressources via des déclarations (yaml, json,…)
- AWS Cloud Development Kit (CDK) : surcouche à CloudFormation pour simplifier l’écriture et la maintenance des configurations, avec d’éventuelles abstractions

Services « fonctionnels »

- AWS Simple Notification Service (SNS) : service de notification
- AWS Simple Email Service (SES) : envoi de mails de masse
- AWS EventBridge (anciennement CloudWatch Events): bus qui permet de router des évènements vers des cibles que l'on configure ; dispose également de fonctions de replay et d'archives.
- AWS Lambda : exécution de charges de travail (sous forme de fonctions avec une sous-couche docker)
- AWS Simple Queue Service (SQS) : service de file de messages : permettra de centraliser les informations à récupérer au niveau de chaque instance CiviCRM
- AWS Simple Storage Service (S3) : service de stockage de buckets

Cas d’utilisation : notifications

- Prometheus :
  - serveur de collecte de métriques
  - évaluation du déclenchement d'alertes selon des règles
  - alertes visibles depuis Grafana
- AlertManager :
  - distributeur de notifications
  - sorties multiples, dont SMTP et AWS SNS
  - fonctionnalités avancées : aggrégation d'alertes, priorisation d'alertes
- Notification d'alertes par mail : pourrait ressembler à du spam
- SNS permet d'éviter de gérer soi-même cette problématique

Cas d’utilisation : envoi de mails en masse

- domaine uepal.net :
  - SPF, DKIM, DMARC pour envoyer les mails depuis SES
  - MX pour faire transiter les mails entrants par OVH
- domaine bounce.uepal.net :
  - SPF, DMARC pour ne pas autoriser d'envoi de mail
  - MX pour faire transiter les mails entrants vers SES
- mise à disposition d'un accès limité AWS : Organizations (compte AWS), Délégation d'identité d'email, IAM
- authentification Civiparoisse sur AWS : IAM Roles Anywhere + K8S Cert-Manager
- envoi de mail depuis Civiparoisse : SDK : API SES (récupération message id)
- génération des stats : EventBridge -> Lambda -> SQS
- réception des données depuis Civiparoisse : EventBridge ->SQS
- mécanisme de fusible : Cloudwatch + Lambda

Envoi de mails en masse: objets à provisionner : compte gestion Uepal

- (global) identité de mail sur le domaine
- identité de mail sur l'email + configuration set
- le compte de paroisse

Envoi de mails en masse: objets à provisionner : compte paroisse

- groupes IAM : sendGroup consumerGroup
- file SQS qui sera relevée périodiquement par CiviCRM
- groupe de logs pour CloudWatch Logs, et son bucket S3 associé
- configuration EventBridge, avec la création d'une archive et la règle pour tout envoyer vers le group de log et la file SQS
- configuration set, qui contiendra la configuration à utiliser pour l'envoi de mails (dont la destination des events vers l'EventBridge)
- des rôles pour les Lambda, et les Lambdas : stats,fuse et rearm (fonction seulement)
- la règle d'eventBridge pour générer le cron pour les stats
- les alarmes sur bounceRate et complaintRate au niveau configurationSet et emailIdentity, avec le lancement des actions fuse
- role pour RolesAnywhere

Envoi de mails en masse: infrastructure as code

- Nécessaire pour gérer la maintenance et la scalabilité
- Plusieurs technologies disponibles : le SDK, CloudFormation et Cloud Development Kit
- CloudFormation : "briques de base" pour certains services
- Cloud Developement Kit :
  - Véritable code
  - Génère une sortie CloudFormation
  - Propose des abstractions de niveaux supérieurs à CloudFormation
- Localstack (https://www.localstack.cloud/) :
  - Emulation locale de certaines fonctionnalités
  - dispose d'une version gratuite
- Code déjà écrit, synthétise avec LocalStack, mais pas encore testé

Cas d’utilisation : réception de mails

- Nécessité de gérer les mails entrants
- Piste SES
- Piste AWS Workmail
- Piste IMAP/Webmail (ex : dovecot/roundcube)
- Piste aperçu de mail dans CiviCRM (Apache Tika, IMP Horde, Firefox, Marionette)
- Piste retenue : OVH : redirection ou alias de mail

Perspectives de développements complémentaires

- intégration du helper d'authentification (sidecar ? service ? networkPolicies ?)
- développements dans une extension CiviCRM :
  - utilisation du AWS SDK en PHP (intégré via composer et un fichier de configuration)
  - envoi des mails : via l'API AWS SES du SDK, puis récupération et intégration du messsageId dans le natif
  - file de messages SQS : récupérer, stocker et traiter les évènements (ex : hardbounce), qui peuvent éventuellements être des replays
  - rapports SearchKit pour exploiter els données
  - mise en évidence d'un fusible qui aurait fondu, et donc contrôle des bounceRates et complaintRates

Encore d’autres projets lourds… avec vous ?

- aléas de la maintenance
- intégration continue
- passage en Afform des formulaires spécifiques
- (r)évolution multiparoisse ?
- évolutions d'infrastructures ?
- évolution des services utilisateurs ?
- évolution de la documentation
- priorisation des évolutions
- équipe trop restreinte !

- **Venez mouiller le maillot !**

Union des Églises protestantes d'Alsace et de Lorraine