Dédicace

A ... merci ...

Remerciements

Au terme de ce travail, je tiens à exprimer ma profonde gratitude à

Résumé

Ce projet a pour objectif de

Table des matières

Liste des figures

Liste des tableaux

1. introduction generale

1.1 Introduction

Dans un contexte technologique en constante évolution, marqué par une forte concurrence et une accélération des cycles de développement, l'importance des concepts que nous avons mis en œuvre est devenue capitale. L'automatisation, qui était autrefois perçue comme un avantage optionnel, s'impose désormais comme une nécessité incontournable pour garantir la fiabilité, la sécurité et la rapidité des processus.

1.2 Présentation de l'organisme d'accueil

Oneex est une entreprise française basée à Clermont-Ferrand, avec un établissement secondaire à Paris. Elle est spécialisée dans la conception et le développement de solutions logicielles et matérielles dédiées à la vérification d'identité et à l'analyse documentaire. Grâce à des technologies avancées telles que l'intelligence artificielle et l'analyse d'image, Oneex propose des solutions innovantes pour lutter contre la fraude documentaire.

1.2.1 Historique de l'entreprise

- **Création de la société (2018)** Le concept Oneex est né de l'initiative de son fondateur, confronté à la problématique de l'analyse des documents d'identité. N'ayant trouvé aucune solution souveraine respectant les contraintes réglementaires sur les données personnelles, il a décidé de créer et de développer Oneex.
- **Recherche et développement (2018-2020)** Pendant deux années, l'entreprise s'est consacrée à la recherche et au développement. Le logiciel ScanApp a ainsi été créé pour reproduire la vision humaine grâce à une intelligence artificielle capable d'analyser avec précision le pays, le format et les spécificités techniques d'un document.
- **Déploiement de la solution Desktop (2020)** Forte de ce développement, Oneex a lancé une offre complète associant hardware et software, donnant naissance à la solution Desktop.
- **Reconnaissance et impact (2021-2023)** L'entreprise a rapidement acquis une reconnaissance dans son domaine, obtenant des distinctions et certifications de la part de leaders de l'industrie. Elle poursuit aujourd'hui sa croissance à l'international.
- **Percées technologiques et évolution (2023-2024)** Oneex a enrichi ses produits de nouvelles fonctionnalités, notamment le monitoring à distance, l'accès à des statistiques détaillées, la gestion autonome du parc matériel et un accompagnement expert en fraude documentaire. Après une levée de fonds importante en 2024, l'entreprise développe de nouveaux produits pour renforcer son positionnement en tant que leader de l'analyse

documentaire.

1.2.2 Domaine d'activité

Oneex propose des solutions transversales adaptées à de nombreux secteurs, notamment la santé, les banques, la sécurité et le contrôle d'accès, la location de véhicules, ainsi que les aéroports et compagnies aériennes. L'entreprise se distingue par son application *Oneex Cloud*, une plateforme qui offre un contrôle centralisé et un suivi avancé des vérifications d'identité.

Voici quelques fonctionnalités clés :

- **Suivi des documents scannés** : historique complet, résultats détaillés et traçabilité optimale.
- **Demande d'expertise** : sollicitation d'experts pour garantir des analyses approfondies et fiables.
- **Statistiques avancées** : exploitation des tendances de la fraude pour optimiser la gestion des incidents.

Ces solutions permettent aux entreprises de contrôler efficacement les accès, de réduire les fraudes et de fluidifier les processus d'intégration dans le respect des réglementations RGPD.

1.2.3 Organisation interne

La direction et les équipes de Oneex rassemblent des profils pluridisciplinaires aux parcours variés :

Alexandre Casagrande Fondateur et Président Directeur Général. Après 12 ans au sein du Ministère des Armées et plusieurs années dans la sûreté de grands groupes, il a fondé Oneex avec la volonté de développer une solution souveraine et innovante.

François-Xavier Hauet Directeur Général. Ancien haut fonctionnaire, il a piloté la transformation numérique du Centre Interministériel de Crise puis de la Présidence de la République avant de rejoindre Oneex en 2025.

Julien Otal CTO. Développeur spécialisé dans le multiplateforme, il possède une solide expérience dans la sécurisation de systèmes critiques et la traçabilité des flux.

Xavier Matton Directeur des Opérations. Ingénieur et ancien officier au Ministère de l'Intérieur, il est expert en contrôle des flux et lutte contre la fraude documentaire.

Sébastien Kowalczuk Directeur des Opérations Sud-Ouest. Ancien enquêteur en contre-ingérence économique, il apporte une expertise forte en sécurité des données sensibles.

1.2.4 Produits et services de l'entreprise

Oneex propose une gamme de solutions matérielles et logicielles, parmi lesquelles trois produits phares :

- **Oneex Desktop** : une station de vérification d'identité clé en main.
- **Oneex Suitcase** : une valise mobile permettant de réaliser des contrôles sur le terrain.
- **Oneex Kiosk** : un kiosque en libre-service pour l'accueil et le contrôle des visiteurs.

Ces produits sont complétés par la suite logicielle Oneex Cloud et ScanApp, garantissant un pilotage centralisé et une intégration fluide dans les environnements clients.

1.2.5 Services informatiques et outils internes

Afin de garantir un haut niveau de qualité, Oneex s'appuie sur une infrastructure robuste et des outils modernes facilitant la gestion des opérations.

Les services informatiques

Oneex ScanApp constitue le cœur opérationnel de l'entreprise. Il assure l'analyse documentaire et la vérification d'identité, disponible sur postes fixes comme sur mobiles, avec une interface ergonomique et des fonctionnalités avancées.

Oneex Cloud complète cet écosystème en permettant :

- la gestion des vérifications,
- le suivi historique et la traçabilité,
- la sollicitation d'experts en cas de doute,
- l'analyse statistique des fraudes détectées.

Les outils internes

L'entreprise utilise plusieurs outils collaboratifs et techniques, parmi lesquels : GitLab, Harbor, Nextcloud, Jitsi, Label Studio, YouTrack, Vault, SSO Keycloak. Les bases de données sont hébergées sur des serveurs dédiés et sécurisés, garantissant la confidentialité et la disponibilité des informations sensibles.

2. Objectifs du mémoire et besoins de l'entreprise

2.1 Introduction

Dans un environnement où les systèmes d'information deviennent de plus en plus complexes et interconnectés, la maîtrise de l'infrastructure et des processus de déploiement est un enjeu majeur. Ce mémoire s'inscrit dans cette dynamique, avec pour objectif principal de concevoir et mettre en place une solution automatisée et sécurisée permettant de déployer, superviser et maintenir l'infrastructure technique de l'entreprise Oneex.

Le projet vise à répondre aux besoins opérationnels croissants, à réduire les erreurs manuelles et à garantir un haut niveau de qualité de service, tout en respectant les contraintes de sécurité et de conformité réglementaire.

2.2 Cas et exemples

Plusieurs situations rencontrées chez Oneex ont mis en évidence la nécessité d'une solution d'automatisation et de gestion centralisée de l'infrastructure. Parmi les cas récurrents :

- **Problèmes de cohérence des environnements** : la configuration manuelle des serveurs engendrait des divergences entre les environnements de développement, de test et de production.
- **Difficultés de gestion des secrets** : le stockage et le partage des identifiants, clés d'API et certificats étaient réalisés de façon disparate, augmentant les risques de fuite.
- **Manque de visibilité** : l'absence d'outils de supervision unifiés complexifiait le suivi de l'état des services et la détection des incidents.
- **Temps de déploiement élevé** : chaque mise en place d'une infrastructure nécessitait plusieurs jours de préparation et de validation.

Ces constats ont motivé la définition d'un projet structuré et ambitieux, articulé autour de l'automatisation, de la sécurité et de l'observabilité.

2.3 Les besoins fonctionnels

Les besoins fonctionnels définissent les fonctionnalités attendues de la solution mise en œuvre. Ils se déclinent comme suit :

- **Automatisation du provisioning** : création et configuration des machines virtuelles via des outils d'Infrastructure as Code (Terraform).

- **Gestion centralisée des configurations** : mise en place d'Ansible pour orchestrer l'installation des paquets et le paramétrage des systèmes.
- **Déploiement applicatif automatisé** : utilisation d'un processus GitOps avec Argo CD pour garantir la cohérence des déploiements.
- **Supervision et alertes** : intégration de Prometheus et Grafana pour la collecte et l'affichage des métriques.
- **Gestion sécurisée des secrets** : déploiement de HashiCorp Vault afin de stocker et distribuer les informations sensibles.
- **Mise en place d'environnements distincts** : séparation claire entre les environnements de test, de pré-production et de production.

2.4 Les besoins non fonctionnels

Les besoins non fonctionnels définissent les critères de qualité que la solution doit respecter :

- **Haute disponibilité** : assurer une disponibilité supérieure à 99,9% des services critiques.
- **Sécurité renforcée** : garantir la protection des données sensibles et la conformité aux standards de sécurité.
- **Scalabilité** : permettre l'évolution de l'infrastructure en fonction de la croissance de l'activité et de l'augmentation du nombre de clients.
- **Maintenabilité** : faciliter les mises à jour, les correctifs et l'évolution des configurations.
- **Traçabilité et auditabilité** : conserver l'historique des changements et des déploiements.
- **Réduction du temps de mise en production** : diminuer les délais de déploiement des nouvelles fonctionnalités.

3. Étude théorique et analyse bibliographique

3.1 Concepts généraux

3.1.1 Gestion des secrets

La gestion des secrets consiste à centraliser, stocker et distribuer de manière sécurisée les informations sensibles telles que les mots de passe, les clés API, les certificats et les tokens d'authentification. Elle permet de réduire les risques de compromission liés aux erreurs humaines, aux partages non contrôlés et au stockage non chiffré de ces données. Des solutions comme HashiCorp Vault répondent à ces enjeux en proposant un stockage chiffré et des mécanismes de contrôle d'accès fin.

3.1.2 Gestion de configuration

La gestion de configuration vise à décrire, versionner et appliquer automatiquement les paramètres nécessaires au bon fonctionnement des systèmes et des applications. Elle favorise la cohérence entre les environnements (développement, test, production) et la traçabilité des modifications. Ansible est un outil de référence pour automatiser ces tâches via des playbooks déclaratifs.

3.1.3 Infrastructure as Code (IaC)

L'Infrastructure as Code est une approche qui consiste à définir l'infrastructure (réseaux, machines virtuelles, ressources cloud) sous forme de code versionné. Elle permet d'automatiser la création et la mise à jour des environnements, de gagner en rapidité et de réduire les erreurs. Deux approches principales existent : la déclarative, qui décrit l'état souhaité (ex. Terraform), et l'impérative, qui décrit les étapes nécessaires.

3.1.4 Continuous Integration et Continuous Deployment (CI/CD)

La CI/CD regroupe l'intégration continue (CI) et le déploiement continu (CD). La CI vise à automatiser la compilation et les tests à chaque changement de code, garantissant la qualité du logiciel. La CD va plus loin en automatisant la livraison jusqu'en production. Ces pratiques favorisent l'agilité et réduisent le temps entre le développement et la mise en production.

3.1.5 DevOps

DevOps est un mouvement culturel et organisationnel qui rapproche les équipes de développement et d'exploitation. Il encourage la collaboration, l'automatisation et la responsabilité partagée sur l'ensemble du cycle de vie applicatif. L'objectif est d'améliorer la qualité, la rapidité et la fiabilité des mises en production.

3.1.6 Conteneurisation

La conteneurisation consiste à empaqueter une application avec toutes ses dépendances dans un conteneur léger et portable. Docker est la technologie la plus répandue. Elle simplifie le déploiement et assure la reproductibilité des environnements, quel que soit l'hôte.

3.1.7 Orchestration

L'orchestration permet de gérer automatiquement le cycle de vie des conteneurs sur un cluster de machines. Kubernetes est devenu la référence en la matière. Il offre des fonctionnalités de déploiement, de montée en charge, de tolérance aux pannes et de mise à jour continue.

3.1.8 Monitoring et observabilité

Le monitoring désigne la collecte de métriques et de logs sur les systèmes et applications. L'observabilité élargit cette notion en permettant de comprendre le comportement interne d'un système à partir de ses sorties. Des outils comme Prometheus, Grafana, Loki et Tempo contribuent à assurer la visibilité et la traçabilité nécessaires pour détecter et corriger rapidement les incidents.

3.1.9 Stockage distribué

Le stockage distribué permet de rendre les données disponibles de manière redondante et tolérante aux pannes. Dans un environnement Kubernetes, des solutions comme Longhorn apportent un stockage persistant hautement disponible.

3.1.10 Gestion des logs

La gestion centralisée des logs facilite l'analyse des événements applicatifs et systèmes. Des solutions comme la stack Grafana Loki collectent, stockent et permettent de requêter les logs de manière unifiée.

3.1.11 Reverse proxy

Un reverse proxy est un serveur qui reçoit les requêtes des clients et les redirige vers les serveurs applicatifs. Il joue un rôle essentiel dans la répartition de charge, le chiffrement TLS et la sécurité.

NGINX et Traefik sont des reverse proxies couramment utilisés.

3.1.12 Gestion de pare-feu

La gestion des pare-feux consiste à contrôler les flux réseau entrants et sortants afin de protéger l'infrastructure contre les attaques. Elle implique la définition de règles de filtrage, de NAT et d'accès aux services.

3.1.13 Convention de commits

Les conventions de commits comme Conventional Commits définissent une structure standardisée des messages de commit. Elles facilitent la compréhension des changements, l'automatisation des changelogs et les processus de versioning sémantique.

3.2 Les outils et les solutions

3.2.1 Proxmox

Proxmox est une plateforme de virtualisation open source permettant de gérer des machines virtuelles et des conteneurs. Elle est utilisée comme socle d'infrastructure.

3.2.2 Terraform

Terraform est un outil d'Infrastructure as Code qui permet de décrire l'infrastructure dans un langage déclaratif et de l'appliquer automatiquement.

3.2.3 Cloud-init

Cloud-init est un utilitaire qui configure automatiquement les machines lors de leur démarrage, notamment le réseau, les utilisateurs et les clés SSH.

3.2.4 Vault

Vault est une solution de gestion des secrets qui offre un stockage chiffré et des mécanismes de contrôle d'accès aux informations sensibles.

3.2.5 Consul

Consul fournit des fonctionnalités de service discovery, de configuration distribuée et de supervision des services.

3.2.6 Ansible

Ansible est un outil de gestion de configuration et d'automatisation qui permet de déployer et configurer des applications via des playbooks.

3.2.7 Kubernetes

Kubernetes est une plateforme d'orchestration des conteneurs qui automatise le déploiement, la mise à l'échelle et la gestion des applications conteneurisées.

3.2.8 Kustomize

Kustomize est un outil natif Kubernetes permettant de gérer des configurations complexes par superposition de patches.

3.2.9 Helm

Helm est un gestionnaire de packages Kubernetes qui simplifie le déploiement d'applications à l'aide de charts préconfigurés.

3.2.10 Longhorn

Longhorn est une solution de stockage distribué qui fournit du stockage persistant pour les clusters Kubernetes.

3.2.11 Grafana

Grafana est une solution de visualisation de métriques qui permet de créer des tableaux de bord dynamiques.

3.2.12 Prometheus

Prometheus est un système de monitoring et de collecte de métriques utilisé pour superviser l'infrastructure et les applications.

3.2.13 Loki

Loki est un outil de centralisation des logs optimisé pour une intégration avec Grafana.

3.2.14 Tempo

Tempo est un système de traçabilité des requêtes distribué qui facilite l'analyse des performances applicatives.

3.2.15 Argo CD

Argo CD est un outil de GitOps qui synchronise les déploiements Kubernetes avec le contenu d'un dépôt Git.

3.2.16 MetalLB

MetalLB apporte la gestion des adresses IP flottantes et du Load Balancing dans des clusters Kubernetes en mode bare-metal.

3.2.17 GitLab CI

GitLab CI est un système d'intégration et de déploiement continus qui automatise les pipelines de build et de test.

3.2.18 NGINX

NGINX est un reverse proxy et serveur HTTP performant qui sert de point d'entrée aux applications.

3.2.19 Commitlint

Commitlint valide les messages de commit en s'assurant qu'ils respectent une convention donnée.

3.2.20 Husky

Husky permet d'exécuter des hooks Git (par exemple le lint) avant les commits ou les pushs.

3.2.21 Semantic Release

Semantic Release automatise la gestion des versions et la publication des packages en fonction des commits.

3.3 Les projets informatiques de la société

3.3.1 Introduction

L'entreprise Oneex développe plusieurs projets informatiques stratégiques qui répondent à différents besoins métiers et techniques.

3.3.2 Oneex Front

Application frontend permettant la gestion des opérations, l'affichage des données et l'interaction avec les utilisateurs finaux.

3.3.3 Oneex Back

Backend exposant les API et orchestrant les processus métiers critiques.

3.3.4 Oneex Scanner

Solution logicielle dédiée à l'acquisition et à l'analyse des documents d'identité.

3.3.5 Oneex ScanApp

Application mobile ou desktop facilitant le scan et la vérification en temps réel des documents.

3.3.6 Oneex CSharp

Projet spécifique développé en C# destiné à répondre à des besoins d'intégration ou d'outillage interne.

4. Réalisation du projet

4.1 Introduction

Le projet de mise en place d'une infrastructure automatisée s'est déroulé sur plusieurs mois, en parallèle des activités opérationnelles de l'entreprise. Il s'inscrivait dans une démarche progressive, visant à moderniser les outils et à fiabiliser les processus de déploiement et d'exploitation. Plusieurs contraintes ont dû être prises en compte, notamment la compatibilité avec l'existant, la disponibilité continue des services et la nécessité d'assurer un haut niveau de sécurité. Le planning général a été structuré en phases : conception, mise en œuvre, validation et transfert de compétences.

4.2 Objectifs du projet

Les objectifs principaux du projet étaient les suivants :

- Automatiser le provisioning de l'infrastructure afin de réduire les délais de déploiement.
- Centraliser et sécuriser la gestion des configurations et des secrets.
- Améliorer la supervision et l'observabilité des services.
- Mettre en place une approche GitOps pour les déploiements applicatifs.
- Renforcer la sécurité des accès et des flux réseau.
- Disposer d'environnements distincts (test, staging, production) alignés sur les bonnes pratiques DevOps.

Ces objectifs contribuent à la fiabilisation des opérations et à la création d'une base technique évolutive et maintenable.

4.3 Architecture du projet

L'architecture globale repose sur une infrastructure virtualisée sous Proxmox, pilotée par Terraform et configurée par Ansible. Kubernetes assure l'orchestration des conteneurs, tandis que des solutions complémentaires (Vault, Grafana, Prometheus, Loki, Tempo, Longhorn, Argo CD) viennent couvrir les besoins en sécurité, stockage et supervision.

4.3.1 Infrastructure as Code avec Terraform

Terraform a été utilisé pour décrire et provisionner toute l'infrastructure. Les ressources suivantes ont été créées de façon automatisée :

- Les machines virtuelles sous Proxmox (masters et workers Kubernetes, serveurs utilitaires).
- Les réseaux virtuels et les volumes de stockage.
- Les configurations initiales des systèmes (cloud-init).

Les modules Terraform ont été organisés de manière modulaire afin de pouvoir réutiliser et adapter la configuration facilement.

4.3.2 Configuration automatique avec Ansible

Ansible a permis de configurer les serveurs après leur création. Des rôles dédiés ont été développés pour :

- Installer Kubernetes (RKE2) sur les nœuds.
- Déployer les outils de monitoring.
- Configurer Vault et Consul.
- Appliquer les règles de sécurité et les configurations système.

Cette approche garantit une configuration homogène et reproductible.

4.3.3 GitOps avec Argo CD

Argo CD a été mis en place pour gérer les déploiements applicatifs de manière GitOps. Chaque modification des manifestes Kubernetes dans le dépôt Git est automatiquement synchronisée avec le cluster. Cette pratique permet de disposer d'un historique complet des changements et de simplifier les rollbacks en cas de problème.

4.3.4 Monitoring et observabilité avec Grafana, Prometheus, Loki et Tempo

Le monitoring s'appuie sur Prometheus pour la collecte des métriques et Grafana pour la visualisation. Loki centralise les logs applicatifs et systèmes, tandis que Tempo gère les traces distribuées. L'ensemble forme une stack cohérente permettant d'assurer la supervision et le diagnostic des incidents.

4.3.5 Stockage distribué avec Longhorn

Longhorn a été déployé comme solution de stockage distribué pour Kubernetes. Il fournit un stockage persistant, tolérant aux pannes, avec des fonctionnalités de snapshot et de restauration.

4.3.6 Gestion des secrets avec Vault

Vault est utilisé comme coffre-fort centralisé pour la gestion des secrets. Il permet de stocker et de distribuer les mots de passe, tokens et certificats de manière sécurisée. Des politiques de contrôle d'accès ont été configurées pour limiter l'exposition des informations sensibles.

4.3.7 Mise en place de services internes

Plusieurs services internes ont été installés pour répondre aux besoins quotidiens des équipes :

- GitLab pour la gestion des dépôts et de l'intégration continue.
- Harbor comme registre privé de conteneurs.
- YouTrack pour le suivi des tâches et incidents.
- Nextcloud pour le partage sécurisé de fichiers.

Ces outils sont hébergés dans l'infrastructure Kubernetes.

4.3.8 Sécurité et gestion de pare-feu avec pfSense

La sécurité périmétrique repose sur un pare-feu pfSense. Des règles de filtrage ont été définies pour contrôler les flux entrants et sortants, ainsi qu'un VPN pour les accès administratifs. Des mesures de surveillance des connexions ont également été mises en place.

4.3.9 Mise en place des services de test et de staging

Des environnements spécifiques ont été créés pour permettre la validation des développements avant leur mise en production. Ces environnements reprennent l'architecture de production, facilitant les tests de montée en charge et les scénarios de validation.

4.3.10 Mise en place de la CI/CD avec GitLab CI

GitLab CI a été configuré pour automatiser les étapes de build, test et déploiement des applications. Des pipelines ont été créés pour assurer la qualité du code et le déploiement vers les environnements Kubernetes.

4.3.11 Mise en place des services de production pour les clients

Enfin, les environnements de production ont été finalisés et validés en respectant les exigences de performance et de sécurité. La mise en service a inclus la configuration des sauvegardes, la documentation des procédures et la formation des utilisateurs.

Conclusion

4.4 Conclusion générale

4.5 Les limites du projet

4.6 Les améliorations possibles

4.7 Les enseignements personnels

Comme expliqué par [1], il est essentiel de sécuriser les accès.

References

[1] M. Akermi. “Why is Big Data not so big anymore”. In: *Conference proceedings*. Springer. 2022, pp. 1–13.