

Rapport de stage

DEVOPS : mise en production et automatisation d'application sur le cloud.

Du 19 avril 2021 au 27 août 2021

PYMMA-SOFTWARE



133 Rue Simon Vollant, 59130 Lambersart

Université de Lille



42 Rue Paul Duez, 59000 Lille

Tuteur dans l'entreprise : Nicolas Héron

Tuteur universitaire : Philippe Marquet

Remerciements

Tout d'abord, je tiens à remercier **Monsieur Nicolas Héron**, gérant de l'entreprise PYMMA-SOFTWARE pour m'avoir guidé et répondu à toutes mes interrogations concernant mon projet professionnel.

Je remercie également le personnel de l'entreprise et les autres stagiaires qui m'ont permis de passer des bons moments.

Je remercie **Monsieur Philippe Marquet**, tuteur universitaire, pour m'avoir accompagné durant mon parcours en entreprise, pour son écoute et son suivi hebdomadaire.

Je remercie mes professeurs de l'université pour la qualité des enseignements qu'ils m'ont dispensé.

Je profite de l'occasion pour remercier mes camarades de classe avec qui j'ai partagé des moments riches et agréables dans la mise en œuvre des projets collectifs.

Je remercie mes parents de m'avoir soutenu pendant cette période particulière de pandémie.

Résumé

Ma période de stage au sein de l'entreprise PYMMA-SOFTWARE a démarré du 19 avril 2021 et prendra fin le 27 août 2021. Cette entreprise est une Entreprise de Services du Numérique basée à Lambersart. Lors de ce stage, on m'a confié la mission d'automatiser la mise en production d'applications sur un cloud. Ce travail comporte différents projets qui portent sur le déploiement des applications selon l'infrastructure d'accueil. L'objectif de ce rapport est de vous montrer en détail les missions d'un DEVOPS au sein d'une entreprise. Ma fonction était de concevoir l'infrastructure puis de créer ou d'adapter des scripts d'installations pour qu'ils correspondent au résultat. Elle est conçue pour permettre de générer une architecture fiable et rapide à mettre en place. Les méthodes employées sont des scripts YAML couplés avec Ansible AWX, un logiciel d'automatisation des tâches avec une interface graphique. Ce dernier est relié en mode SSH aux machines de l'infrastructure pour déployer les logiciels dont a besoin la machine pour faire fonctionner l'application. Lorsqu'un bogue intervient, une prise en main de la machine est effectuée en SSH en vue de trouver les erreurs et de les corriger. Dans les premiers temps de mon stage, j'ai mis en place des infrastructures tests pour vérifier le bon fonctionnement des scripts. Il a également été question de me former à concevoir et/ou améliorer des scripts. Dans un second temps, une fois Ansible AWX et les scripts YAML pris en main, mon tuteur d'entreprise m'a remis une infrastructure virtualisée sur un cloud avec Proxmox. Cette formation m'a donnée les clés pour répondre aux attentes de l'entreprise, à savoir le déploiement rapide d'applications sur un cloud. Je suis heureux d'effectuer mon stage dans cette entreprise qui m'a accueilli chaleureusement. Après une année particulière liée à la crise sanitaire, c'est grand plaisir à me retrouver dans une ambiance de travail stimulante et constructive. Pour conclure, cette expérience m'a permis de me perfectionner dans l'analyse de situations concrètes, de m'y confronter et d'y apporter les réponses adéquates. Elle m'a permis aussi d'apprendre à utiliser de nouveaux langages et de nouveaux logiciels précieux pour mon insertion et mon parcours professionnel.

Abstract

My internship at the company PYMMA-SOFTWARE started on April 19th 2021 and will take an end on August 27th 2021. This company is an IT company based in Lambersart. During this traineeship, I was given the mission to automate the production of applications on a cloud. This work includes different projects related to applications deployment in accordance with the host infrastructure. The objective of the present report is to describe in detail the missions of a DEVOPS within a company. My function was to design the infrastructure and then create or adapt installation scripts to match the results desired. It is designed to generate a reliable and fast architecture to set up. The methods employed are YML scripts doubled with Ansible AWX, a software to automate tasks with a graphic interface. The latter is connected in SSH mode to the machines of the infrastructure in order to run the scripts. When a bug occurs, the machine is runned in SSH mode to find the errors and correct them. During my first days as a trainee, I set up test infrastructures to check the correct functioning of the scripts. This also trained me to design or improve these scripts. In a second step, once I had learned Ansible AWX and YAML scripts, my company tutor gave me a virtualized infrastructure on a cloud with Proxmox. This training gave me the keys to meet the company's expectations, namely the rapid deployment of applications on a cloud. I am delighted to do my traineeship in PYMMA-SOFTWARE where I immediately felt comfortable. After this particularly year due to the health crisis, it is a great pleasure to find myself in a stimulating and constructive work atmosphere. To conclude, this experience has allowed me to improve my analysis of concrete situations, to confront them and to provide the appropriate answers. It also allowed me to learn how to use new languages and software, which will be invaluable for my integration and my professional career.

Sommaire

Introduction	5
Contexte	6
I/ Présentation de l'entreprise	6
1) Présentation de l'entreprise.....	6
2) Son domaine d'activité	6
3) Sa clientèle	6
4) L'équipe	7
5) L'environnement de travail	8
II/ Présentation des projets	8
Contribution	9
I/ Introduction.....	9
1) Analyse du problème.....	9
2) Objectif des projets.....	11
3) Organisation du travail	12
4) Les principaux outils utilisés	12
a. Les outils pour déployer.....	12
b. Les outils à déployer.....	14
II/ Projet Onisep	15
1) Introduction	15
2) L'infrastructure	16
3) Les solutions retenues	16
4) Ma réalisation.....	17
a. Ajout du mode ReplicaSet sur les bases MongoDB	17
b. Intervention sur les scripts	18
5) Les difficultés rencontrées.....	19
III/ Projet Assurfleet.....	20
1) Introduction	20
2) L'infrastructure	20
3) Les solutions retenues	21
4) Ma réalisation.....	22
a. Les erreurs	22
b. Les corrections	22
c. Les tests	23
5) Les difficultés rencontrées.....	23
6) Test et validation du projet	24
Conclusion	25
Bilan	26
Bibliographie	27
Annexes	28

Introduction

Dans le cadre de ma troisième année de Licence Informatique à l'Université de Lille, je dois effectuer un stage de fin d'étude afin de valider mon année. Ce stage vise à mettre en pratique les connaissances théoriques acquises lors de mon cursus. Il me permet aussi d'être formé au sein d'une entreprise dans le but d'apprendre de nouvelles compétences.

Ce rapport présente mon environnement de travail ainsi que ma mission principale effectuée à l'entreprise PYMMA-SOFTWARE. Celle-ci est l'automatisation de la mise en production d'application sur un cloud. En effet, l'entreprise possède déjà des scripts d'automatisation cependant certains doivent être améliorés pour correspondre à l'architecture souhaitée.

Tout d'abord, je présenterai la société PYMMA-SOFTWARE ainsi que le projet élaboré durant mon stage. Puis, je détaillerai le contenu de ma mission avec les divers aspects techniques utilisés. Enfin, je dresserai un bilan des compétences acquises tant sur le plan technique que sur le plan humain.

Contexte

Je vais vous présenter dans un premier temps mon entreprise d'accueil, son domaine activité, sa clientèle, son équipe, l'environnement de travail et dans un second temps, je détaillerai les projets.

I / Présentation de l'entreprise

1) Présentation de l'entreprise

Créée en 2014, PYMMA-SOFTWARE est une ESN¹, spécialisée en ingénierie informatique et située à Lambersart dans la zone d'activité de la Cessoie. Monsieur Nicolas Héron en est le fondateur et gérant. Elle réalise des prestations pour des professionnels en combinant la démarche commerciale et leurs propres systèmes d'information.



2) Son domaine d'activité

L'univers de l'informatique est en perpétuelle évolution. Les logiciels open-sources en sont un exemple. Ces logiciels sont libres de droit et progressent grâce à la communauté. Au sein de l'entreprise, l'utilisation de logiciel open-source est privilégié pour concevoir des projets. L'autre domaine de compétence est l'expertise informatique auprès de divers clients.

3) Sa clientèle

PYMMA-SOFTWARE travaille pour de nombreuses entreprises qui ont des secteurs d'activités très variés. J'ai travaillé sur le déploiement de deux projets, le premier est le projet Onisep et le second est le projet Assurfleet.

¹ ESN : Entreprise de Services du Numérique



Le client du projet Onise est l'ONISEP², un organisme public français dépendant du ministère de l'éducation nationale.



Le client du projet Assurfleet est Altima, une filiale de la MAIF spécialisée dans l'assurance des grandes flottes automobiles.

Autres clients :

Almady, une jeune start-up basée en Arabie Saoudite. Son domaine d'activité est la logistique.



Adeo est un groupe propriétaire entre autres des magasins Leroy Merlin et Bricoman.

4) L'équipe

PYMMA-SOFTWARE fonctionne sous forme de SARL³, Société À Responsabilité Limitée et compte à ce jour 4 salariés, 1 alternant et 7 stagiaires.

Les salariés sont :

- Mr Nicolas Héron : Fondateur et gérant de PYMMA-SOFTWARE
- Mr Guillaume Hochart : Lead Développeur⁴
- Mr Pierre Delgrange : Développeur
- Mr Alois Drey : Développeur

L'alternant est Cheick Fofana et les stagiaires sont Inas Bekkouche, Hodari Bigwi, Mouctar Fofana, Hassan Elmiloudi, Louis Verdon, Fatima Rami et moi-même.

² ONISEP : Office National d'Information Sur les Enseignements et les Professions

³ SARL : Société À Responsabilité Limitée

⁴ Lead Développeur : Son rôle est de diriger une équipe de développeurs

Je vous présente ci-dessous l'organigramme de cette structure :

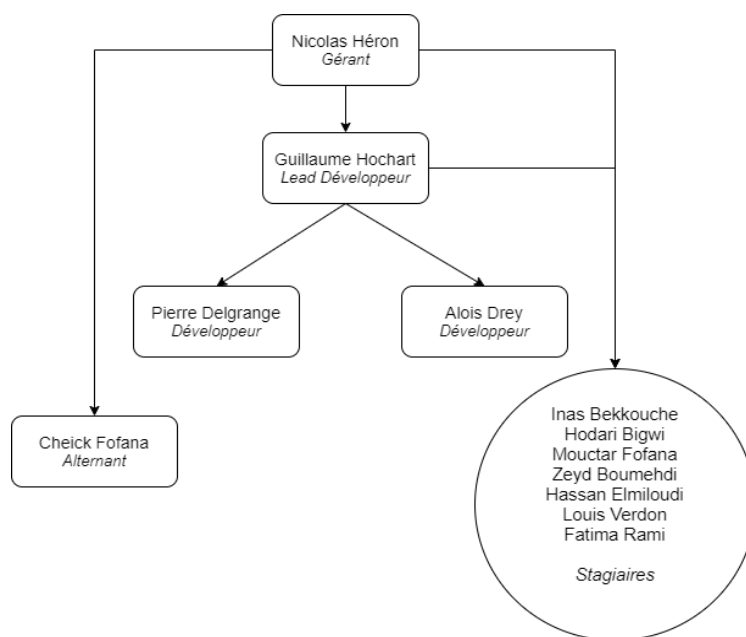


Illustration 1 – Organigramme de l'entreprise PYMMA-SOFTWARE

5) L'environnement de travail

L'équipe évolue dans un open-space⁵ où l'ambiance est agréable. C'est un environnement professionnel qui facilite une meilleure entraide entre collègues. De plus, pour pouvoir communiquer aisément et rapidement nous utilisons la messagerie Slack au sein de l'équipe. L'organisation du travail se fait autour de réunions régulières pour comprendre les projets à mener.

II / Présentation des projets

Durant mon stage, j'ai travaillé sur deux projets qui ont la même finalité : l'automatisation et le déploiement d'une application sur un cloud⁶. Cependant les infrastructures étaient différentes donc l'analyse, la recherche et la mise en œuvre des scripts⁷ diffèrent.

Depuis le démarrage de ma mission de 19 avril 2021, l'entreprise m'a confié le déploiement des 2 projets suivants :

- **Onisep**, une application d'aide à l'orientation des étudiants
- **Assurfleet**, une application de gestion de grande flotte automobile pour les assurances Altima, une filiale de la MAIF

⁵ Open-space : Espace de travail collectif

⁶ Cloud : Plusieurs serveurs accessibles depuis internet et opérationnels en permanence

⁷ Script : Langage de programmation utilisant les fonctionnalités d'un système informatique

Contribution

Après la présentation de l'entreprise et des projets, je vais vous développer mes réalisations effectuées lors de mon stage.

De plus, je vais approfondir sur ma contribution dans le projet Onisep et le projet Assurfleet.

I / Introduction

1) Analyse du problème

Lorsqu'une entreprise crée une application, celle-ci doit impérativement prévoir son hébergement et comment s'effectuera son déploiement.

Le déploiement d'une application est une mission difficile car cela peut prendre du temps, il peut surgir des erreurs et il faut entretenir les versions des applications.

Pour cela, il existe plusieurs solutions :

- Le déploiement non automatisé
- Le déploiement automatisé
- Le déploiement externalisé par un prestataire

Après avoir utilisé un cloud externe de grands prestataires informatiques, PYMMA-SOFTWARE a relevé le défi de créer son propre cloud pour plus d'autonomie, plus de facilité pour la gestion de sa clientèle et pour en réduire ses coûts.

Je vous présente ci-dessous un schéma reprenant l'architecture réseau de l'entreprise PYMMA-SOFTWARE.

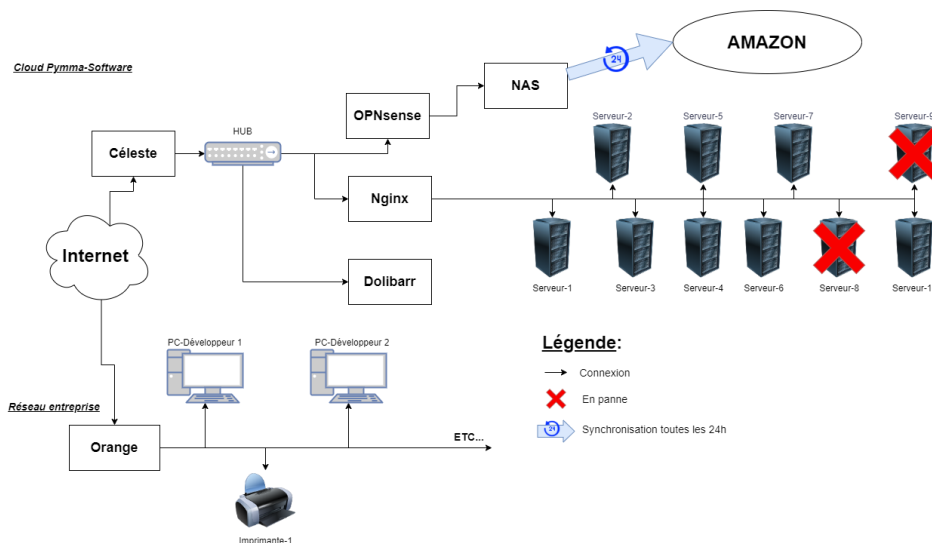


Illustration 2 – Architecture réseau de l'entreprise PYMMA-SOFTWARE

L'infrastructure est composée de deux réseaux, un pour le cloud et l'autre pour les ordinateurs des développeurs et l'imprimante. Le réseau du cloud reçoit la fibre Céleste, soit 1 Gbits/seconde. Celle-ci est reliée à un Hub⁸.

À partir de là trois voies s'offrent :

- Un pare-feu OPNsense qui permet de sécuriser le NAS⁹ où sont stockés toutes les données des applications installées sur le cloud (les base de données et les logs¹⁰). Le NAS est synchronisé chaque nuit aux serveurs d'Amazon.
- Un reverse-proxy NGINX pour augmenter la sécurité de l'infrastructure. De plus, dix serveurs reliés au NGINX sont utilisés pour la virtualisation avec Proxmox afin de créer le cloud. Sur ces dix serveurs, deux sont en pannes et Il y a environ soixante-dix machines virtuelles en fonctionnement sur les serveurs en activité et celles-ci comportent les outils et les logiciels nécessaires pour :
 - Le déploiement d'applications sur cloud (Ansible AWX)
 - La compilation de code des développeurs (Jenkins)
 - Le stock des fichiers exécutable (Nexus)
- Un serveur avec Dolibarr installé dessus. Dolibarr est un ERP¹¹, utilisé pour la gestion des clients et des devis.

Le second réseau a comme fournisseur d'accès internet Orange et sert aux développeurs. Il a été mis en place pour sécuriser d'avantage et éviter les erreurs humaines sur le cloud. Cependant, les développeurs peuvent accéder au cloud par Internet, si une route existe sur le reverse-proxy¹² NGINX.

Mr Nicolas Héron a décidé de ce choix pour mettre en avant le savoir-faire de l'entreprise, et également, pour qualifier le personnel et permettre sa montée en compétences.

En ce qui me concerne, le choix de cette méthode me paraît pertinent car il offre à l'entreprise une grande liberté dans la mise en œuvre de ces projets.

⁸ Hub : Appareil relié à plusieurs machines en réseau afin de permettre la transmission d'informations

⁹ NAS : Serveur de stockage sur un réseau

¹⁰ Log : C'est un historique des événements d'une application ou d'un système

¹¹ ERP : Progiciel de gestion intégré

¹² Reverse-proxy : Serveur qui gère le trafic externe pour le renvoyer sur des serveurs en interne

2) Objectif des projets

Nous avons porté notre choix sur une mise en production automatisée de l'application car plus rapide, plus fiable à mettre en œuvre, permet plus facilement l'entretien des versions des applications.

De plus, il existe déjà plusieurs ressources dans l'entreprise :

- Des scripts YAML
- Une architecture avec des outils d'automatisation sur un cloud (Ansible AWX / Proxmox)

Le but de l'automatisation est de permettre aux développeurs lorsqu'ils ont fini une version d'une application, de la déployer sur le cloud de l'entreprise afin de montrer au client l'avancée du projet.

Ce processus est mis en place de la manière suivante :

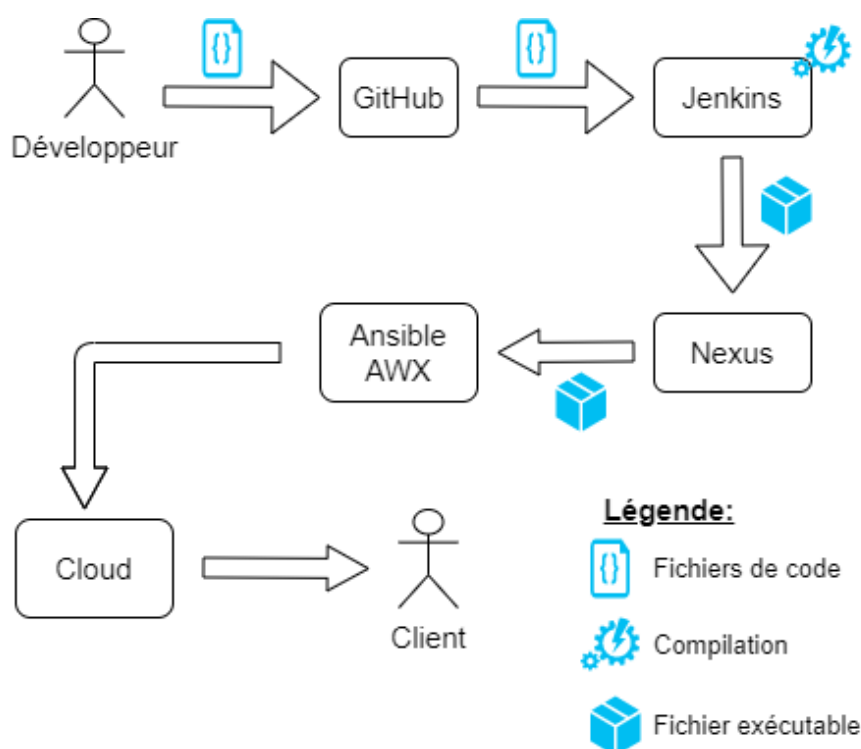


Illustration 3 – Schéma du processus d'automatisation du déploiement d'application

À chaque nouvelle version de l'application, les développeurs envoient leurs codes sur GitHub. Pour compiler ce code, nous utilisons Jenkins. L'exécutable ainsi créé est envoyé à Nexus et sera utilisé dans Ansible AWX pour déployer cette dernière version sur le cloud et la rendre accessible au client.

Ce travail est basé sur une recherche documentaire et en ligne, de procédures installations. Les forums de discussion constituent pour moi une ressource importante d'aide et de formation.

3) Organisation du travail

Avant de démarrer chaque projet, j'assistais à une réunion organisée par mon tuteur d'entreprise afin de m'expliquer, avec un schéma, l'infrastructure à mettre en place.

Ensuite, il me donnait l'accès aux scripts existant relatif à chaque projet depuis un dossier Git.

Lors des projets, nous étions deux, moi et mon tuteur d'entreprise. J'avais le rôle de DEVOPS¹³ et mon tuteur celui de leader du projet. Pour le projet Onisep, je n'ai pas eu de contrainte de temps pour sa réalisation. Cependant pour le projet Assurfleet, je devais finir le projet en deux semaines.

Chaque jour, mon tuteur d'entreprise prenait des nouvelles sur l'avancée du projet et abordions ensemble les difficultés que je rencontrais afin de m'apporter son aide précieuse.

Aussi, je pouvais demander de l'aide à Guillaume Hochart ou à Alois Drey.

4) Les principaux outils utilisés

a. Les outils pour déployer



Docker est logiciel de virtualisation qui utilise le principe des conteneurs, c'est-à-dire les machines virtualisées auront le même système d'exploitation et les mêmes ressources que la machine hôte.



Vagrant est un logiciel de virtualisation couplé avec Virtual Box qui permet de créer, configurer, détruire des machines virtuelles rapidement.



Proxmox est un logiciel de virtualisation fournit avec une interface web ce qui facilite l'administration de l'infrastructure et à une fonction de clustering ce qui assure une haute disponibilité.

¹³ DEVOPS : Technique qui cherche à allier le développement logiciel et l'administration des infrastructures



Ansible AWX est un logiciel d'automatisation des tâches avec une interface graphique, il utilise le moteur de tâche Ansible et il fonctionne grâce à une API REST. C'est une version open-source du logiciel Ansible Tower.



GitHub est une application web qui propose l'hébergement et la gestion des projets logiciels grâce à l'utilisation de son logiciel de gestion de version Git.



Linux est un groupe de systèmes d'exploitation open-sources. Lors de mes projets, j'ai utilisé les systèmes d'exploitation Ubuntu et CentOS.



YAML Ain't Markup Language est un langage utilisé pour mettre en série des données. L'extension de ces fichiers est le « .yaml » ou « .yml ». Ces fichiers sont connus pour être facilement compréhensible par l'homme.



Visual Studio Code est IDE¹⁴ développé par Microsoft. Il est rapide à prendre en main et permet l'ajout d'extensions pour améliorer l'environnement de développement.



Jenkins est un outil d'intégration en continue. Il est open-source.



Nexus repository est un logiciel de gestion de dépôt et a une forte dépendance envers Maven car il utilise un système d'artéfacts.

¹⁴ IDE : Environnement de développement

b. Les outils à déployer



Spring Boot est un framework¹⁵ utilisé pour construire une infrastructure Java facilement. Il permet donc d'aider au développement et aux tests d'une application.



Vaadin est une plateforme comportant un ensemble de composants Web, un framework Web Java et un ensemble d'outils.



Drools est un système de gestion de règles métier capable de définir des règles et de les appliquer à des données. C'est une solution utilisée pour déduire certaines informations d'un ensemble de données.



AngularJS est un framework JavaScript développé par Google afin de permettre la création de pages Web.



PostgreSQL est un SGBGRO, Système de Gestion de Base de Données Relationnelle et Objet, et il utilise le langage SQL pour interagir avec les bases de données.



mongoDB

MongoDB est un Système de Gestion de Base de Données Orienté Documents et stock ces données sous le format JSON. C'est une base de données NoSQL car elle n'utilise pas le langage SQL pour interagir avec sa base.



Apache Tomcat est un serveur d'application Web Java, il permet de fournir un environnement aux développeurs pour exécuter leurs applications Web Java.

¹⁵ Framework : Ensemble de bibliothèques de fonctionnalités utiles pour les développeurs



Nginx est un serveur Web et un reverse proxy, pratique afin de rediriger le trafic internet d'un serveur à un autre.

II / Projet Onisep

1) Introduction

La société PYMMA-SOFTWARE disposait d'une version plus ancienne de l'application Onisep. Les développeurs ont procédé à une mise à jour pour répondre aux attentes du client. Il s'agit de l'ajout du mode ReplicaSet sur les bases de données MongoDB.

Ce mode doit être composé de plusieurs serveurs, un primaire et les autres secondaires. Cette architecture est basée sur une relation « maître / esclave », c'est-à-dire que le serveur primaire reçoit toutes les requêtes et gère la cohérence des données et lors d'une mise à jour de la base primaire, les serveurs secondaires recevront une réplication.

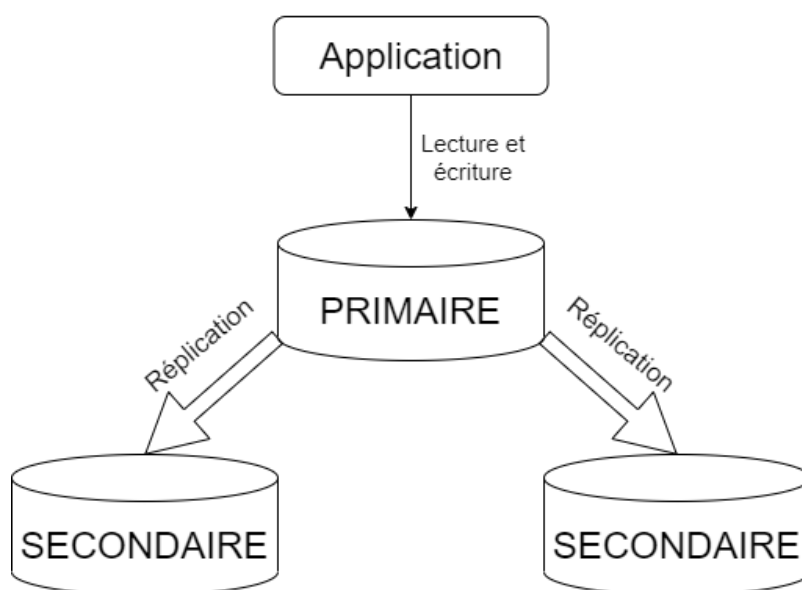


Illustration 4 – Schéma du fonctionnement d'un ReplicaSet

Ma mission est aussi de repérer les blocs qui ne fonctionnent pas dans scripts, de les corriger et de les tester.

2) Infrastructure

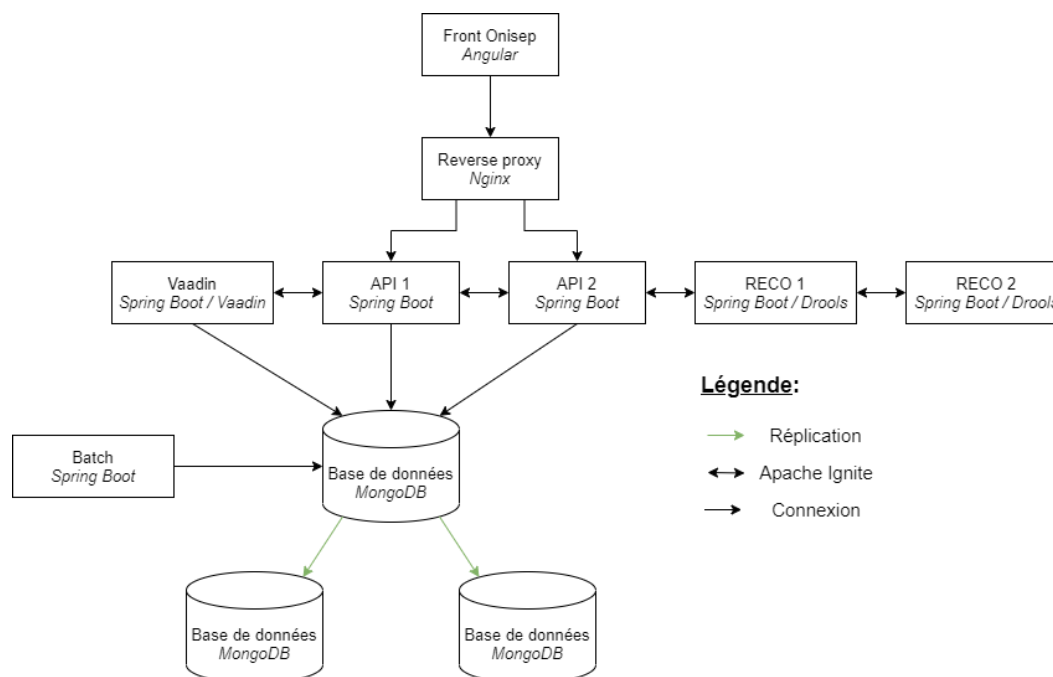


Illustration 5 – Infrastructure de l'application Onisep

L'architecture est composée de onze machines virtuelles, trois de sont utilisées pour les bases de données MongoDB reliées entre elles par un RépliaSet. Ensuite, deux APIs¹⁶, un logiciel Batch¹⁷ et une application Vaadin se connectent aux bases de données pour récupérer les informations. En parallèle, deux machines avec un moteur de règles Drools utilisent les données des APIs. Vaadin, les APIs et les moteurs de règles sont reliés par un calculateur Apache Ignite. Enfin, un front-end¹⁸ en AngularJS se connecte aux APIs en passant par un reverse-proxy pour récupérer les données. Le reverse-proxy a pour mission de basculer d'une API à l'autre afin de fluidifier le trafic.

3) Les solutions retenues

Tout d'abord, nous avons retenues l'utilisation du logiciel Ansible AWX car cet outil est parfait pour lancer les scripts d'automatisation et il est déjà présent sur l'architecture final, c'est-à-dire le cloud de PYMMA-SOFTWARE.

¹⁶ API : Interface de programmation applicative

¹⁷ Batch : Logiciel de traitement par lots des données

¹⁸ Front-end : Correspond à tout ce qui est visible pour l'utilisateur final sur une application web

Cependant pour mon premier projet, je devais faire mes preuves avant de pouvoir directement accéder au cloud. Pour cela, j'utilisais le logiciel Docker afin de créer une infrastructure virtualisée rapidement pour me permettre de bien prendre en main Ansible AWX et le système exploitation Ubuntu.

Lorsque ces outils ont été maîtrisés, nous avons décidé de changer de logiciel de virtualisation en passant sur le logiciel Vagrant pour créer une architecture plus fidèle à la réalité.

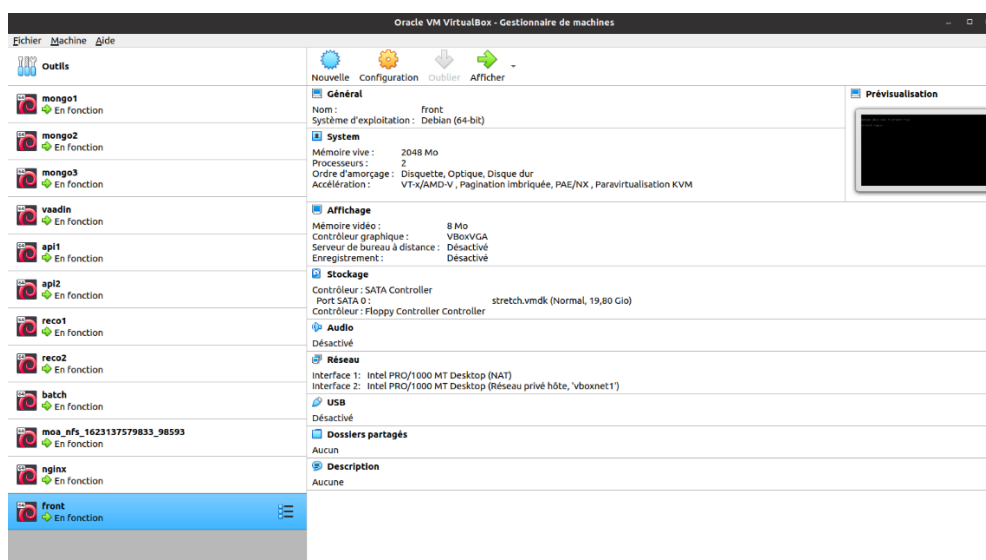


Illustration 5 – Infrastructure virtualisée de l'application Onisep avec Vagrant

4) Ma réalisation

a. Ajout du mode ReplicaSet sur les bases MongoDB

Tout d'abord, il n'existait aucun script pour l'installation des bases de données MongoDB donc j'ai dû le réaliser. Celui-ci effectue les tâches suivantes :

- Installation de MongoDB
- Lancement de MongoDB
- Rendre les bases de données aptes au RéplicaSet
- Créer un RéplicaSet sur la base de données primaire

Ensuite, une connexion à l'une des bases de données MongoDB est opérée pour accéder au terminal afin d'écrire la commande suivante : « rs.status() ; ». Cette commande permet de vérifier le bon fonctionnement du RéplicaSet.

Pour comprendre le fonctionnement d'un RéplicaSet et découvrir les différentes commandes que je devais mettre en place, j'ai entrepris des recherches et lu les documentations officielles de MongoDB sur le sujet.

b. Intervention sur les scripts

- Les erreurs

Les scripts existants dans l'entreprise devaient être remis à jour car ils faisaient appel à des logiciels non utilisés sur le cloud de PYMMA-SOFTWARE ou bien à des liens de téléchargements morts.

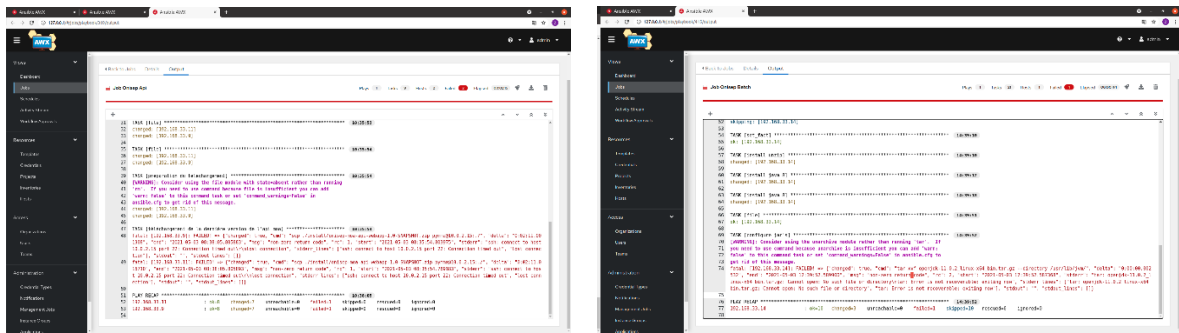


Illustration 6 – Erreur lien de téléchargement mort ou non valide

De plus, j'ai également rencontré des erreurs sur l'activation du RéplicaSet sur les bases de données MongoDB.

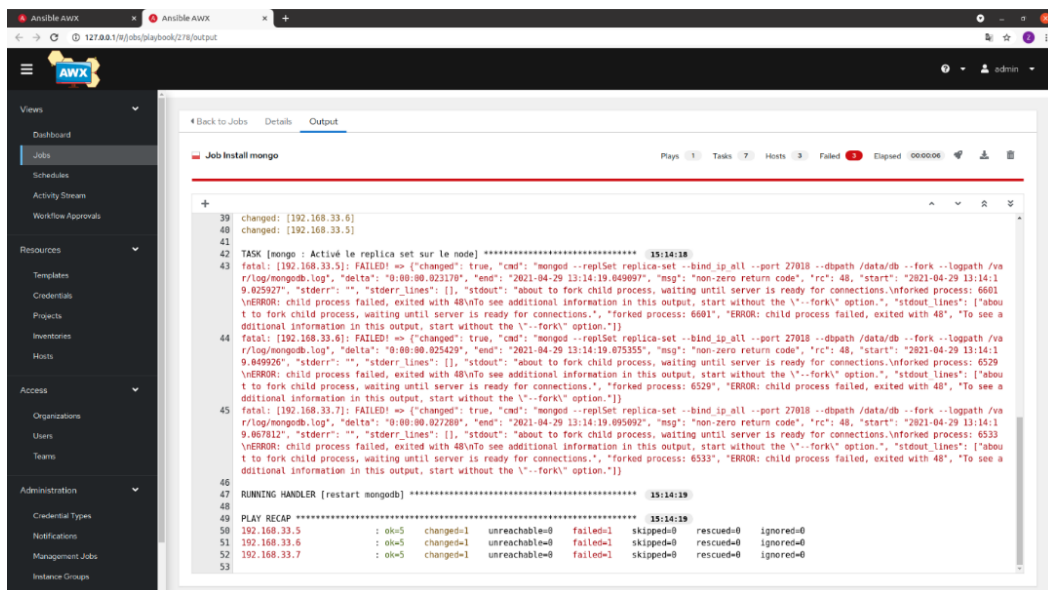


Illustration 7 – Erreur lors de l'activation du RéplicaSet sur les bases

- Les corrections

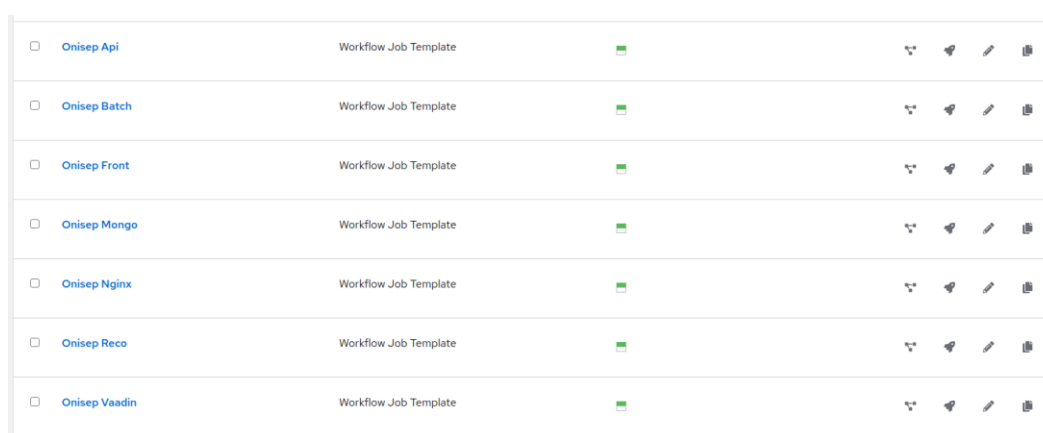
Pour pallier les erreurs des logiciels non utilisés, j'ai dû les supprimer du script.

Les liens de téléchargements morts ont été changées.

Pour finir, concernant l'erreur du RéplicaSet, j'ai procédé à un débogage en SSH¹⁹ sur la machine pour vérifier les logs et aussi s'il n'existait pas des erreurs lors de l'installation de MongoDB.

- Le résultat

Après toutes ces vérifications et tests, tous les scripts ont fonctionné sur Ansible AWX



<input type="checkbox"/>	Onisep Api	Workflow Job Template	■	⌵ ⌵ ✎ 🗑
<input type="checkbox"/>	Onisep Batch	Workflow Job Template	■	⌵ ⌵ ✎ 🗑
<input type="checkbox"/>	Onisep Front	Workflow Job Template	■	⌵ ⌵ ✎ 🗑
<input type="checkbox"/>	Onisep Mongo	Workflow Job Template	■	⌵ ⌵ ✎ 🗑
<input type="checkbox"/>	Onisep Nginx	Workflow Job Template	■	⌵ ⌵ ✎ 🗑
<input type="checkbox"/>	Onisep Reco	Workflow Job Template	■	⌵ ⌵ ✎ 🗑
<input type="checkbox"/>	Onisep Vaadin	Workflow Job Template	■	⌵ ⌵ ✎ 🗑

Illustration 8 – Capture d'écran du bon fonctionnement des scripts du projet Onisep sur Ansible AWX

Néanmoins, j'ai rencontré une grosse difficulté à partir de ce moment.

5) Les difficultés rencontrées

Elles sont de plusieurs ordres :

- Nouveaux logiciels que j'ai appris en une formation donc il me manquait beaucoup d'informations sur certains sujets. Heureusement qu'il existe des documentations et des forums en ligne pour m'aider.
- L'application Onisep fonctionne avec des APIs codées en Spring Boot. Cependant, Spring Boot pour s'exécuter a besoin d'un fichier de configuration appelé « properties », j'essayé en vain de les faire fonctionner mais sans succès. Cette erreur

¹⁹ SSH : Secure Shell permet de se connecter de manière sécurisée à un hôte distant

n'était pas prise en compte par Ansible AWX donc mes scripts fonctionnaient mais ne lançaient pas l'application sur les machines concernées.

Malgré des recherches sur le sujet, mon tuteur de stage a repris le projet et me donna une nouvelle mission le projet Assurfleet.

Plus tard, j'ai appris par mon tuteur que l'erreur des fichiers "properties" étaient difficilement solvables car il fallait avoir une grande connaissance en Spring Boot pour comprendre le problème.

III / Projet Assurfleet

1) Introduction

L'entreprise disposait déjà pour ce projet d'un moyen de déploiement automatique. Cependant, les différents scripts comportaient des erreurs.

Comme pour le précédent projet, j'ai procédé à un repérage des blocs défectueux, je les ai corrigés puis testés.

2) Infrastructure

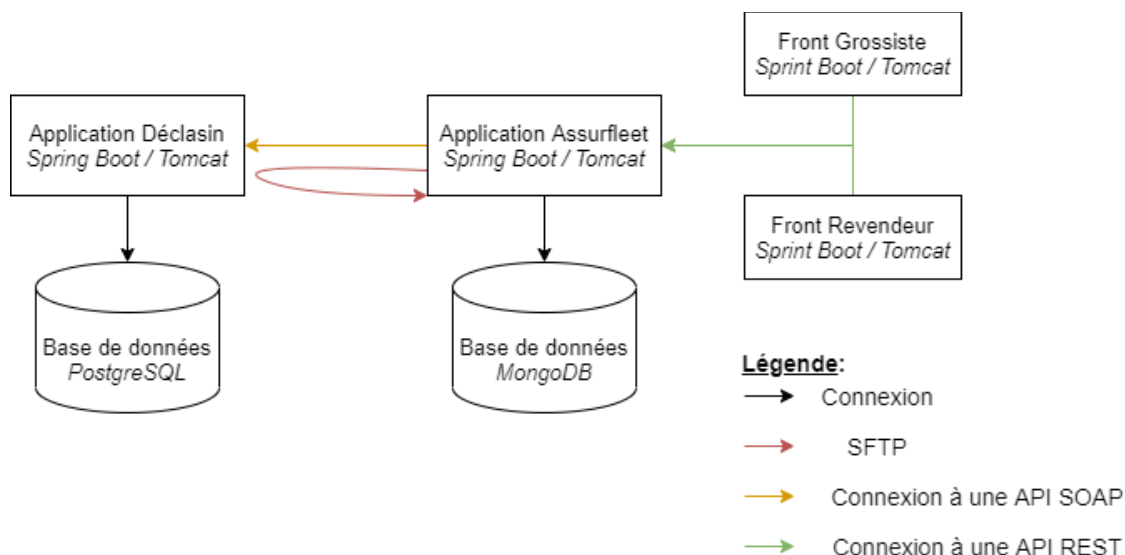


Illustration 9 – Infrastructure de l'application Assurfleet

L'architecture est composée de six machines virtuelles, deux d'entre elles sont affectées au front-end, Front Grossiste et Front Revendeur. Elles sont connectées à l'API Assurfleet grâce à une API REST²⁰ pour récupérer les données.

La troisième machine contient l'API Assurfleet. Elle est connectée à une base de données MongoDB (4ème machine virtuelle) et à l'application Déclasin (5ème machine virtuelle). Pour se connecter à Déclasin, il y a 2 moyens possibles, le protocole SFTP²¹ pour l'envoi sécurisé de fichiers entre les 2 applications et l'utilisation de l'API SOAP²² pour récupérer les données.

L'application Déclasin est connectée à une base de données PostgreSQL (6ème machine virtuelle).

Toutes ces machines sont connectées au NAS du cloud qui récupère ses informations importantes comme les fichiers de configuration, les logs et les données.

3) Les solutions retenues

Nous avons retenu l'utilisation du logiciel Proxmox afin de mettre en place notre architecture virtualisée.

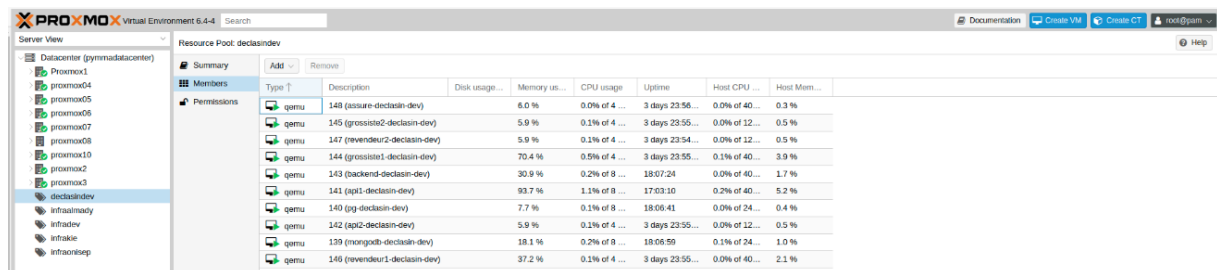


Illustration 10 – Infrastructure virtualisée de l'application Assurfleet sur le cloud Proxmox

Nous avons vu dans l'introduction que le cloud de PYMMA-SOFTWARE possède plusieurs outils, ils seront utiles pour l'élaboration de ce projet. On peut citer comme outil, Ansible AWX pour lancer les scripts, Nexus pour stocker les versions des applications et Jenkins pour compiler le code.

²⁰ API REST : Interface qui permet d'établir une communication entre votre ordinateur et un serveur avec des requêtes HTTP (GET, POST, PUT, DELETE)

²¹ SFTP : SSH File Transfert Protocol, utilise le mode de connexion SSH pour transférer des fichiers

²² API SOAP : Protocole d'échange d'information fondé sur XML qui autorise la communication via des requêtes HTTP.

Toutes les machines fonctionnent sous le système d'exploitation CentOS afin de reproduire l'infrastructure du client Altima.

4) Ma réalisation

a. Les erreurs

Après le déploiement de l'application, nous avons constaté que tous les scripts fonctionnaient normalement. Cependant, l'application ne se lançait pas. Avec le soutien de mon collègue Alois, nous nous sommes aperçus que le pare-feu de la base de données n'autorisait pas l'accès à l'application.

Par ailleurs, les scripts existants avaient des erreurs d'installation, c'est-à-dire qu'ils téléchargeaient une version antérieure à celui du projet actuel ou des versions obsolètes inutilisables.

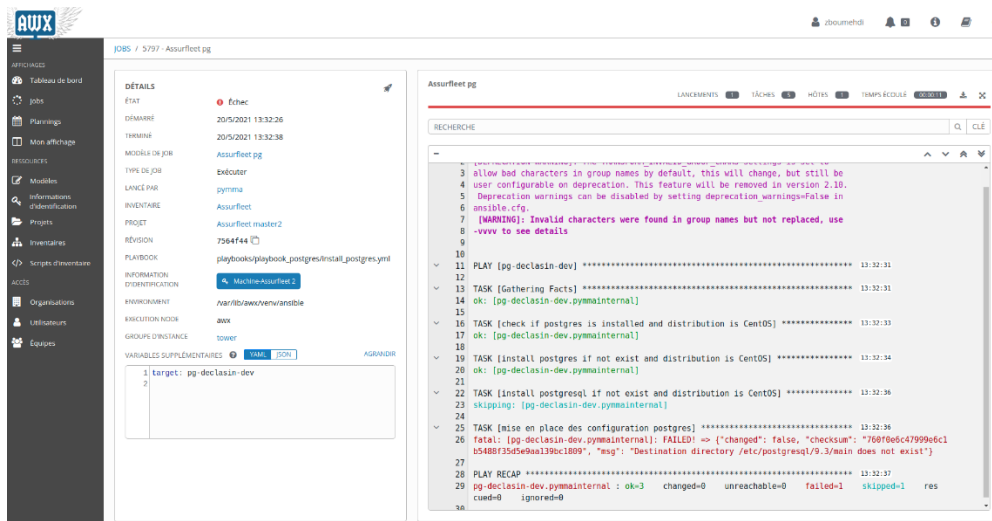


Illustration 11 – Erreur sur Ansible AWX lors de l'installation de PostgreSQL

b. Les corrections

Afin de corriger l'erreur du pare-feu qui bloquait la connexion à la base de données, j'ai dû ajouter un bloc dans le script PostgreSQL. Ce bloc a pour rôle d'ouvrir la connexion sur le port de la base de données.

Les erreurs d'installations sont faciles à corriger puisqu'il existe des documentations en ligne sur la procédure d'installation. Par exemple sur ce projet, le script existant installait la version de PostgreSQL 9.2 alors que la suite du script attendait la version 9.3 au minimum, pour

résoudre cela j'ai dû comprendre d'où vient l'erreur puis de changer la version de PostgreSQL lors de l'installation.

c. Le résultat

Les scripts fonctionnent parfaitement, comme le prouve l'illustration ci-dessous :









<input type="checkbox"/>	Assurfleet API	Workflow Job Template		   
<input type="checkbox"/>	Assurfleet Déclasin	Workflow Job Template		   
<input type="checkbox"/>	Assurfleet Front Revendeur	Workflow Job Template		   
<input type="checkbox"/>	Assurfleet Grossiste	Workflow Job Template		   
<input type="checkbox"/>	Assurfleet Mongo	Workflow Job Template		   
<input type="checkbox"/>	Assurfleet PostgreSQL	Workflow Job Template		   

Illustration 12 – Capture d'écran du bon fonctionnement des scripts du projet Assurfleet sur Ansible AWX

Et d'autres part, le déploiement de l'application a bien été effectué. Les front-end des applications Assurfleet et Déclasin apparaissent et réagissent.

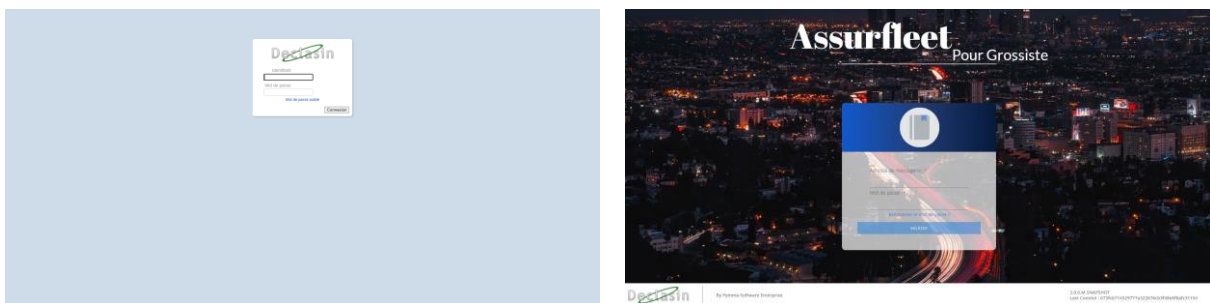


Illustration 13 – Capture d'écran du front-end Déclasin et Assurfleet Grossiste

5) Les difficultés rencontrées

Elles sont de plusieurs ordres :

- La compréhension des scripts existants nécessite un travail de recherche important.

- La méconnaissance de CentOS m'a fait perdre un temps précieux pour comprendre et travailler sur les scripts.

J'ai toutefois réussi à déboguer les problèmes rencontrés et à trouver les solutions adéquates.

6) Validation du projet

Cette phase a nécessité la mise en place de plusieurs tests.

Il s'agit de vérifier que :

- Les données sont dans la base de données
- Les données sont accessibles depuis l'application
- La bonne version de l'application
- Les logs

À l'issue de ce travail, Monsieur Hochart, Lead Développeur, a organisé une réunion avec les autres développeurs afin qu'ils puissent prendre en main le projet pour déployer leurs futures versions de l'application.

De mon côté, j'ai complété un manuel existant avec les informations importantes à savoir pour déployer une nouvelle version de l'application et les caractéristiques des machines utilisées.

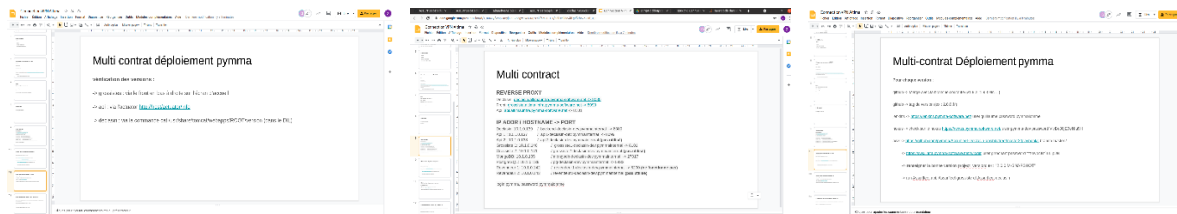


Illustration 14 – Documentation d'aide pour les développeurs sur « Comment déployer l'application Assurfleet ? »

Conclusion

Le stage est pour moi une expérience enrichissante et très constructive. Travailler dans de bonnes conditions, mettre en pratique mes acquis, entouré d'une équipe stimulante, accueillante et bienveillante sont autant de choses positives que j'en retire.

Grâce à ce stage, j'ai élargi mes connaissances en étant confronté avec des professionnels avec qui j'ai pu échanger. J'ai pu ainsi découvrir le fonctionnement d'une entreprise, des différents projets menés et des différentes techniques utilisées.

Cette expérience a été découpée en trois phases. Dans un premier temps la formation pour mieux appréhender les techniques de DEVOPS. Ensuite, d'intervenir sur 2 applications importantes de la société pour les déployer sur le cloud créé et géré par PYMMA-SOFTWARE. Et enfin, une validation des projets par le responsable.

De plus, il est nécessaire de rappeler que le stage n'est pas encore terminé puisqu'il se clôturera le 27 août 2021. D'autres projets viendront conforter mon expérience et enrichir mes acquis.

Ma présence auprès de professionnels au sein de l'entreprise m'a permis de mieux définir mon projet professionnel et d'envisager mon futur parcours de formation. En effet, les acquis du stage sont un complément utile, formateur pour me familiariser avec les processus de recherche, d'analyse et d'élaboration adéquate.

Bilan

Après une année compliquée à cause de la crise sanitaire, j'ai été très heureux de pouvoir effectuer mon stage en présentiel avec un accueil chaleureux et stimulant.

Je vais vous faire d'un état des lieux des compétences acquises tant sur le plan technique qu'humain mais aussi vous faire part des difficultés rencontrées.

Sur le plan des acquisitions, ce stage m'a permis d'être plus autonome dans la recherche de l'information et des solutions.

Sur le plan humain, la disponibilité de l'ensemble du personnel, les échanges avec d'autres stagiaires ont été source d'échange de pratiques, de soutien en cas de difficultés.

Cependant, j'ai rencontré quelques difficultés en ce qui concerne le temps consacré à la formation et la compréhension des scripts existants.

Pour conclure, cette immersion dense et constructive m'a donné l'envie de poursuivre mon cursus de formation par la voie de l'alternance. Les mises en situation de travail correspondent tout à fait à mon mode d'apprentissage.

Bibliographie

- <https://pymma-software.heron-software.com/>
- <https://fr.linkedin.com/in/nicolas-heron-pymma>
- <https://github.com/ansible/awx>
- <https://docs.docker.com/get-started/>
- <https://www.vagrantup.com/docs>
- https://pve.proxmox.com/wiki/Main_Page
- <https://www.jenkins.io/doc/>
- <https://content.sonatype.com/nexus-repository-pro>
- <https://docs.mongodb.com/manual/introduction/>
- <https://www.postgresql.org/docs/current/intro-what-is.html>
- <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#getting-started>

Annexes

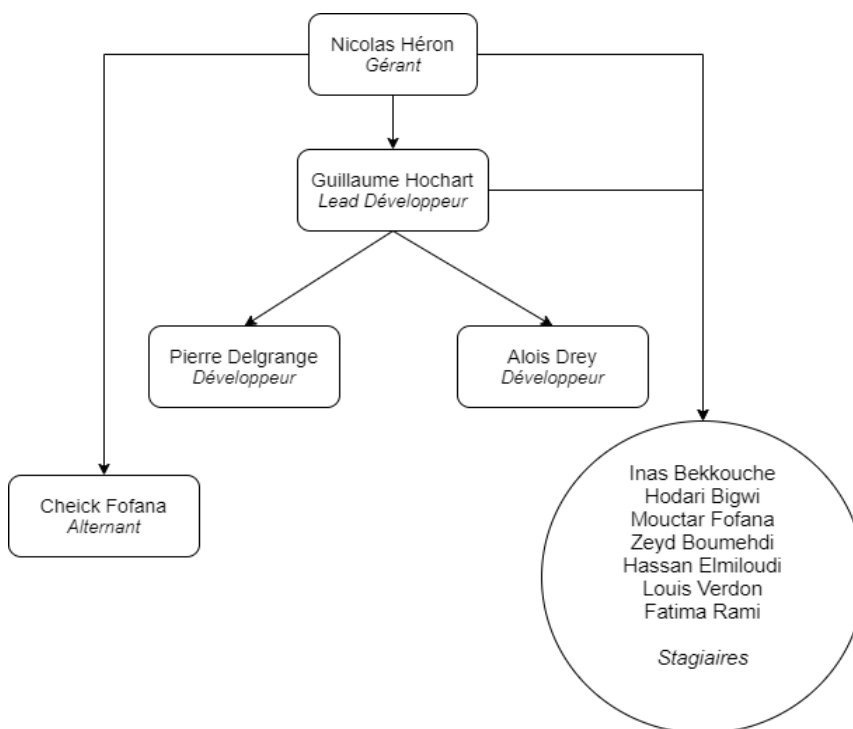


Illustration 1 – Organigramme de l'entreprise PYMMA-SOFTWARE

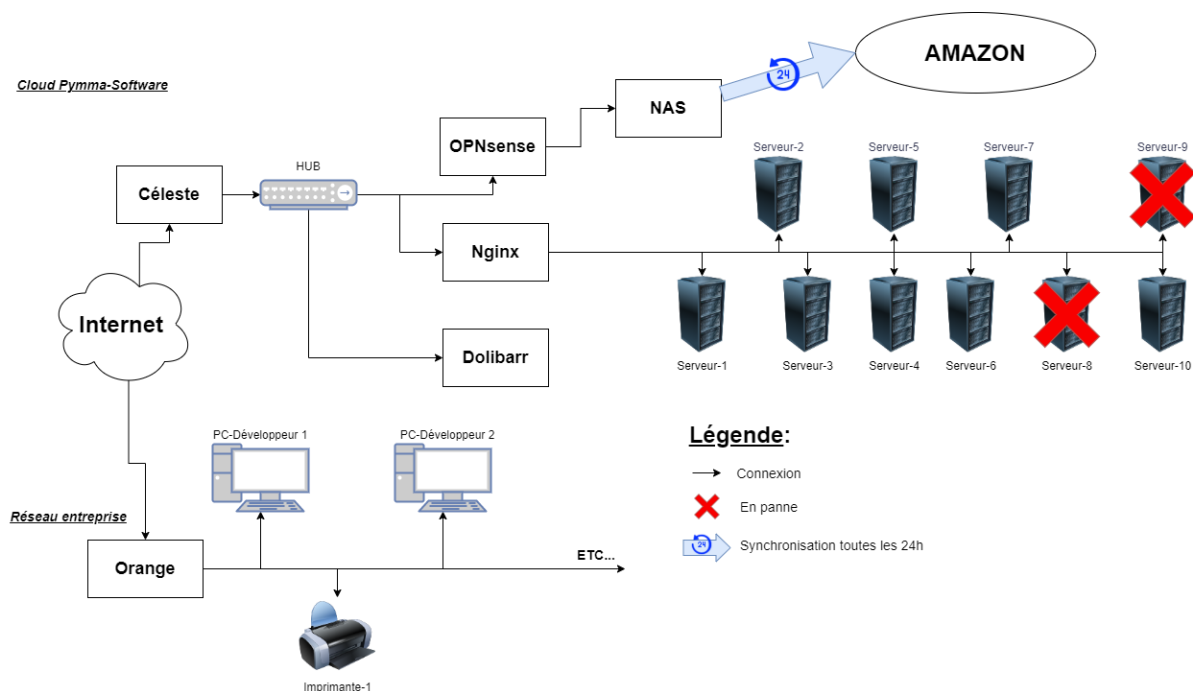


Illustration 2 – Architecture réseau de l'entreprise PYMMA-SOFTWARE

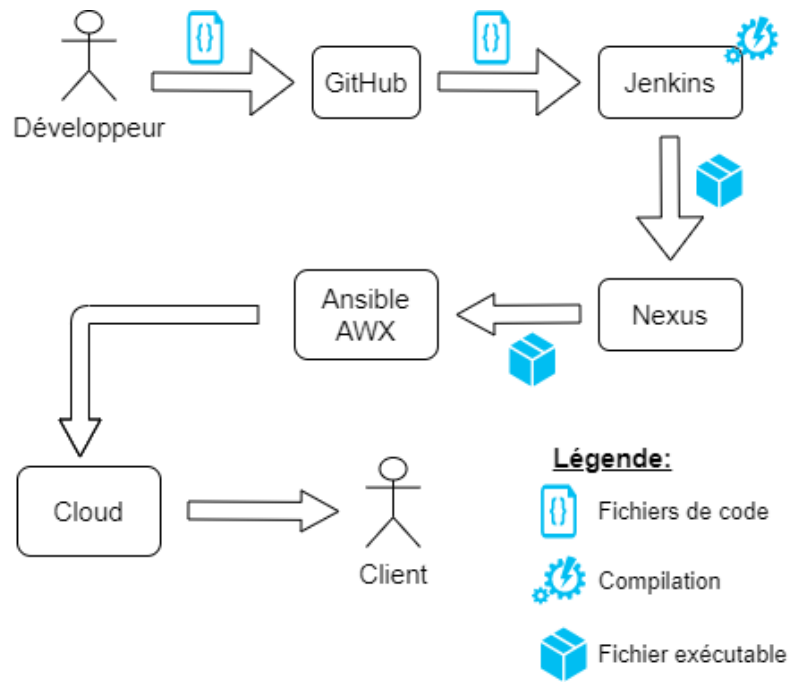


Illustration 3 – Schéma du processus d'automatisation du déploiement d'application

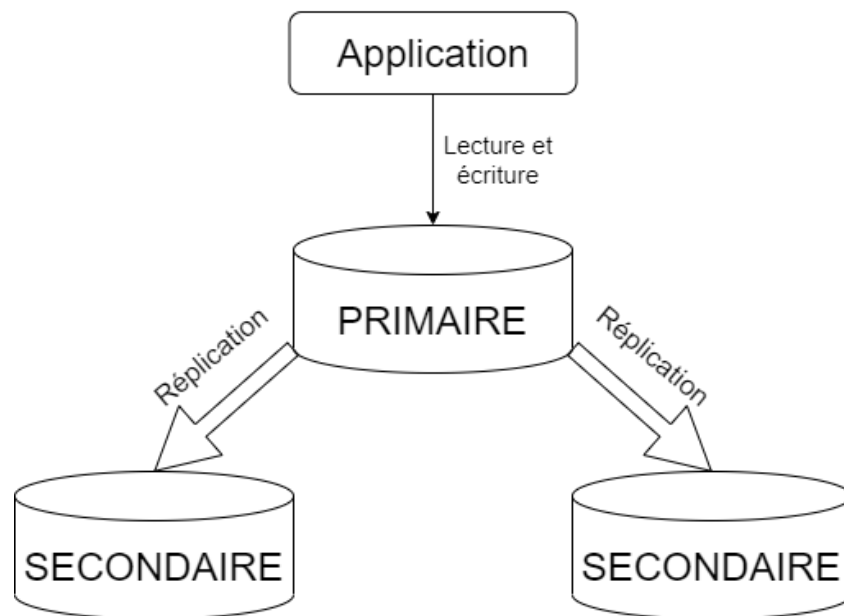


Illustration 4 – Schéma du fonctionnement d'un ReplicaSet

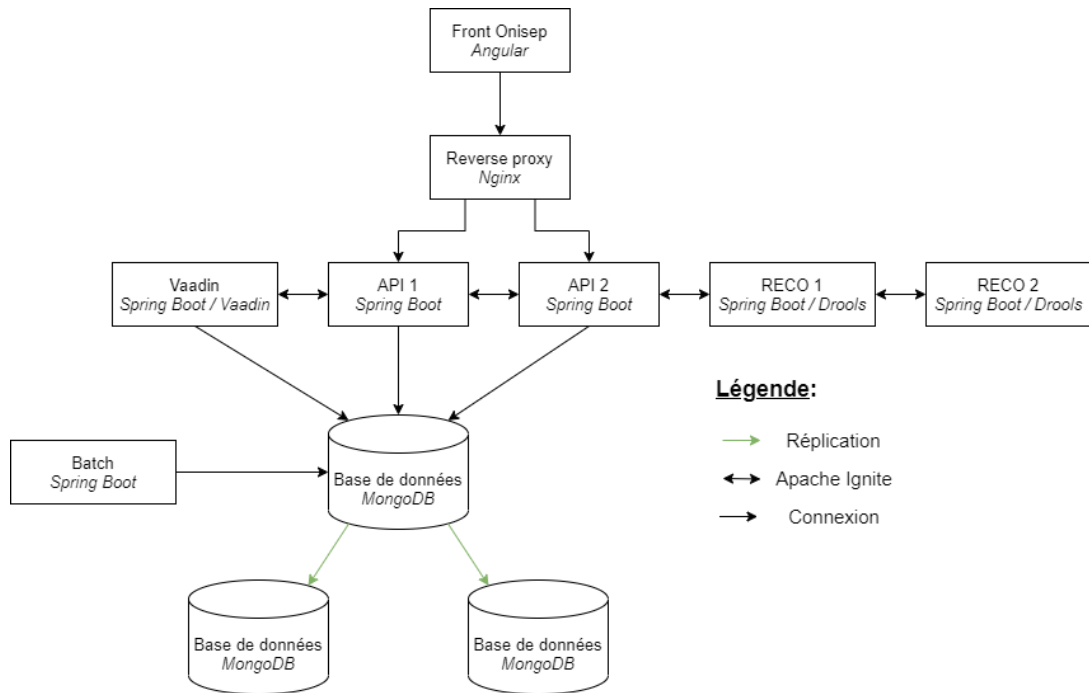


Illustration 5 – Infrastructure de l’application Onisep

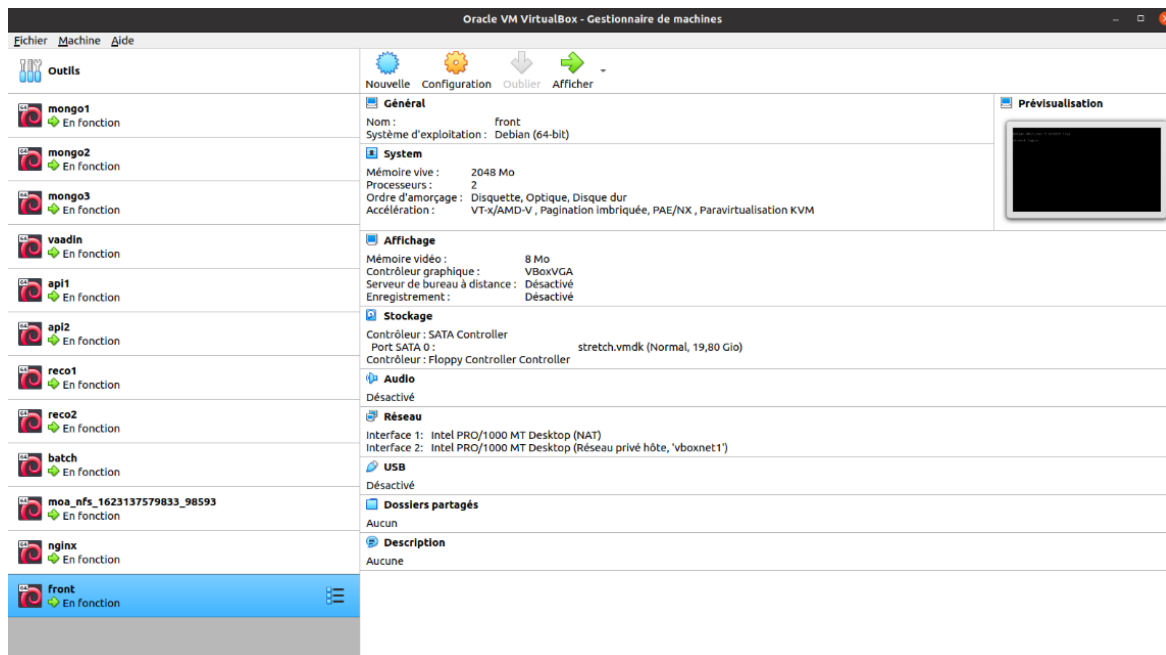


Illustration 5 – Infrastructure virtualisée de l’application Onisep avec Vagrant

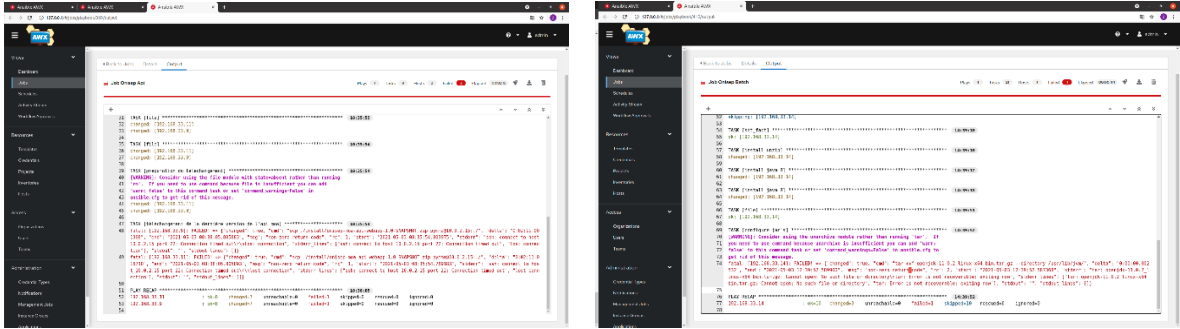


Illustration 6 – Erreur lien de téléchargement mort ou non valide

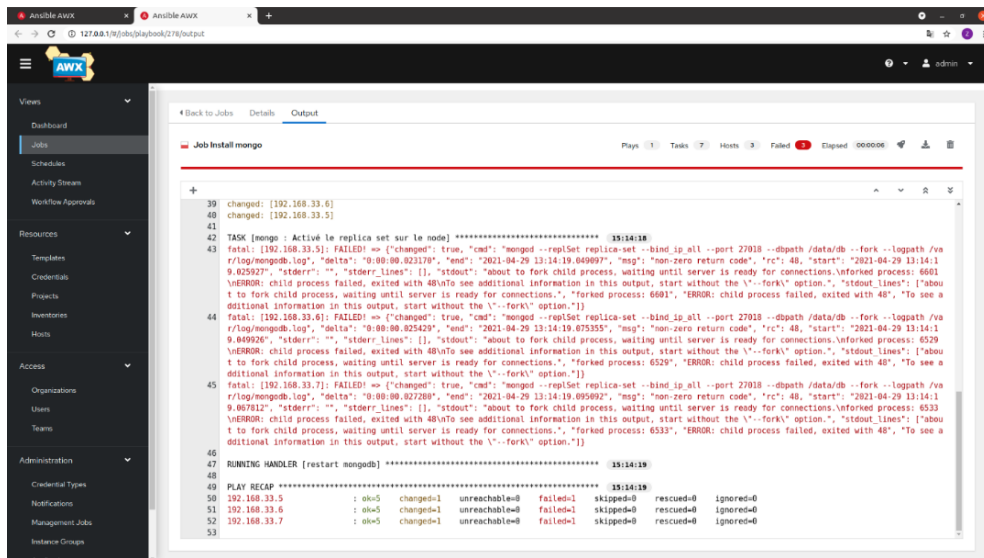


Illustration 7 – Erreur lors de l'activation du RéplicaSet sur les bases

<input type="checkbox"/>	Onisep Api	Workflow Job Template	✔
<input type="checkbox"/>	Onisep Batch	Workflow Job Template	✔
<input type="checkbox"/>	Onisep Front	Workflow Job Template	✔
<input type="checkbox"/>	Onisep Mongo	Workflow Job Template	✔
<input type="checkbox"/>	Onisep Nginx	Workflow Job Template	✔
<input type="checkbox"/>	Onisep Reco	Workflow Job Template	✔
<input type="checkbox"/>	Onisep Vaadin	Workflow Job Template	✔

Illustration 8 – Capture d'écran du bon fonctionnement des scripts du projet Onisep sur Ansible AWX

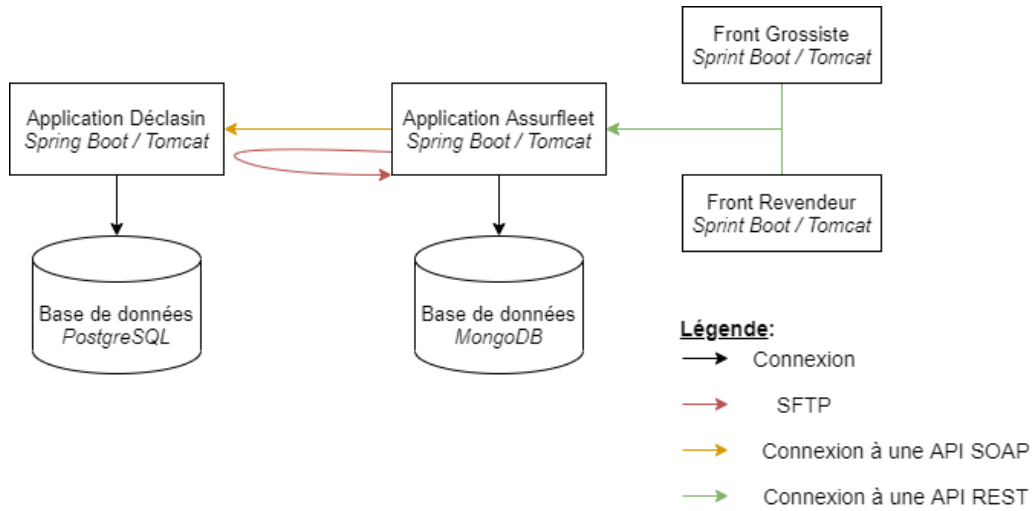


Illustration 9 – Infrastructure de l'application Assurfleet

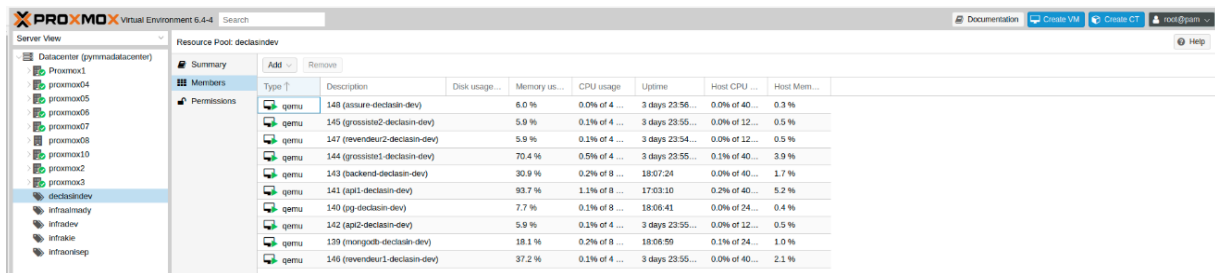


Illustration 10 – Infrastructure virtualisée de l'application Assurfleet sur le cloud Proxmox

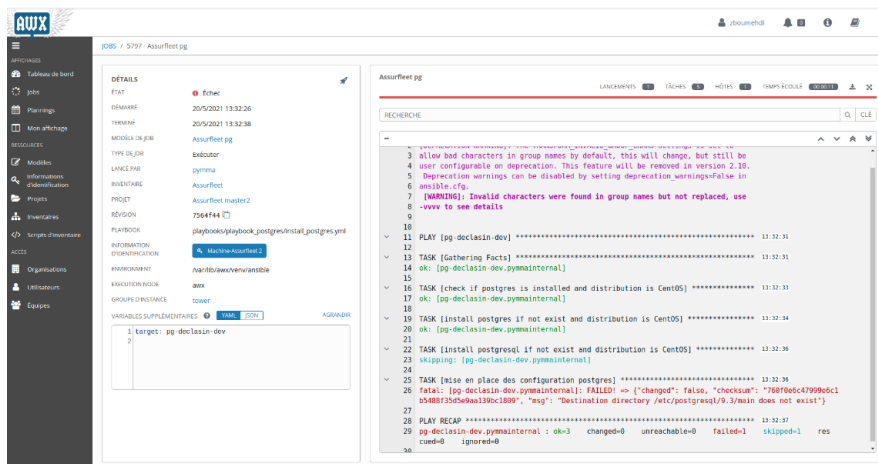


Illustration 11 – Erreur sur Ansible AWX lors de l'installation de PostgreSQL

<input type="checkbox"/>	Assurfleet API	Workflow Job Template					
<input type="checkbox"/>	Assurfleet Déclasin	Workflow Job Template					
<input type="checkbox"/>	Assurfleet Front Revendeur	Workflow Job Template					
<input type="checkbox"/>	Assurfleet Grossiste	Workflow Job Template					
<input type="checkbox"/>	Assurfleet Mongo	Workflow Job Template					
<input type="checkbox"/>	Assurfleet PostgreSQL	Workflow Job Template					

Illustration 12 – Capture d’écran du bon fonctionnement des scripts du projet Assurfleet sur Ansible AWX

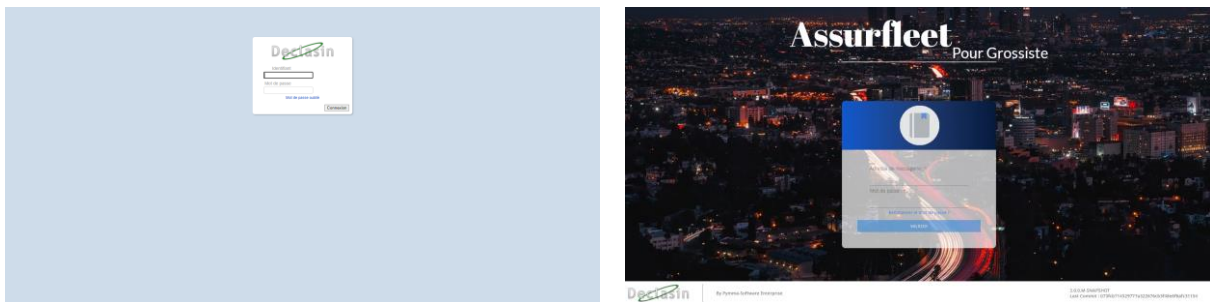


Illustration 13 – Capture d’écran du front-end Déclasin et Assurfleet Grossiste

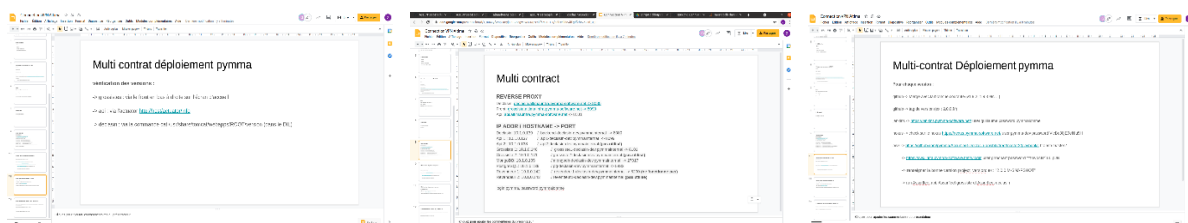


Illustration 14 – Documentation d’aide pour les développeurs sur « Comment déployer l’application Assurfleet ? »