



# Rapport de Projet de Fin d'Etudes

Présenté en vue de l'obtention du diplôme de

**LICENCE NATIONALE EN TECHNOLOGIES DE L'INFORMATIQUE**

**PARCOURS : Réseaux et Services Informatiques (RSI)**

Déploiement d'une Solution d'Allocation  
Dynamique de Ressources avec Proxmox et Odoo

Entreprise : MAG Security

Réalisé par :

Taboubi Firas et Ben Ghorbel Youssef

Soutenu le 18/06/2025 devant les membres du jury :

Mme. HELALI Saida : Président

M. ELMAHROUG Sofien : Rapporteur

Mme. HAJJI Raya: Encadrant ISET

M. GOMRI Mohamed Achref: Encadrant Entreprise

# Dédicaces

## À mes chers parents,

Aucune dédicace ne saurait exprimer la profondeur de ma gratitude pour vos sacrifices, votre soutien inconditionnel et vos encouragements. C'est grâce à votre amour et votre confiance que j'ai pu atteindre cette étape. Que ce travail soit le témoignage de ma reconnaissance éternelle.

## À mes chers frères et sœurs,

Votre soutien constant, vos conseils et votre précieuse amitié qui ont été une source de force tout au long de mon parcours. Merci d'être toujours là pour moi.

## À mes chers amis,

Pour tous les moments de rire, de soutien et de complicité qui ont rendu cette aventure plus légère et mémorable. Votre amitié est un trésor.

## À mon binôme Youssef,

Pour ton sérieux, tes compétences techniques et notre excellente collaboration. Ce projet n'aurait pas été le même sans ton engagement et notre synergie. Je suis ravi d'avoir partagé cette expérience intense avec toi. Merci pour tout.

Avec respect et considération,

Firas Taboubi

## À mes chers parents,

Je vous dédie ce travail en témoignage de mon amour et de ma profonde reconnaissance. Votre amour, votre patience et vos encouragements infaillibles ont été ma plus grande motivation. J'espère que ce travail vous rendra fiers.

## À ma chère famille,

Pour votre présence rassurante et votre soutien qui m'ont toujours poussé à me dépasser. Merci pour tout ce que vous m'avez apporté.

### **À mes chers amis,**

En souvenir de nos moments inoubliables, de votre soutien moral et de votre amitié sincère. Merci d'avoir rendu ces années d'études si spéciales.

### **À mon binôme Firas,**

Pour ta rigueur, ta vision et l'esprit d'équipe exceptionnel que nous avons formé. Ce fut un véritable plaisir de relever ce défi technique à tes côtés. Ta bonne humeur et ta persévérance ont été essentielles à notre succès.

Avec respect et considération,

Youssef Ben Ghorbel

# Remerciements

Au terme de ce Projet de Fin d'Études, nous tenons à exprimer notre profonde gratitude à toutes les personnes qui, de près ou de loin, ont contribué à sa réussite.

Nos remerciements s'adressent tout particulièrement à Mme Saida HELALI, pour nous avoir recommandé cette opportunité de stage enrichissante au sein de MAG Security, point de départ de cette belle aventure.

Nous exprimons notre vive reconnaissance à notre encadrant en entreprise, M. Mohamed Achref GOMRI, pour sa confiance, sa vision stratégique du projet, sa disponibilité constante et ses conseils techniques avisés. Il a été un guide exceptionnel. Son accompagnement précieux et sa pédagogie nous ont permis de surmonter les défis et de mener ce projet à bien.

Nous souhaitons également remercier chaleureusement notre encadrante universitaire, Mme Raya HAJJI, pour son suivi rigoureux, sa disponibilité, ses conseils pertinents et son soutien constant, qui nous ont aidés à structurer notre travail et à rédiger ce rapport.

Nos remerciements vont également à l'ensemble de l'équipe de MAG Security pour leur accueil chaleureux et pour l'environnement de travail stimulant qu'ils nous ont offert.

Enfin, nous remercions sincèrement les membres du jury d'avoir accepté d'évaluer notre travail et de nous honorer de leur présence.

# Table de matières

Table des figures .....	10
Liste des Tableaux .....	13
Liste d'abréviation .....	15
Introduction Générale .....	1
<b>Chapitre 1</b>	
<b>Cadre général du projet .....</b>	<b>3</b>
1.1.    Introduction .....	4
1.2.    Présentation de l'organisme d'accueil .....	4
1.2.1.    Présentation générale de MAG Security .....	4
1.2.2.    Localisation de MAG Security.....	4
1.2.3.    Organigramme de MAG Security .....	5
1.2.4.    Environnement de l'entreprise.....	5
1.2.5.    Services MAG Security .....	6
1.2.6.    Partenaires MAG Security.....	6
1.3.    Présentation du projet.....	7
1.3.1.    Contexte du projet et problématique .....	7
1.3.2.    Solution proposée.....	8
1.3.3.    Objectifs du projet.....	8
1.4.    Choix de la Méthodologie.....	9
1.4.1.    Justification du Choix.....	9
1.4.2.    Organisation de l'Équipe et Rôles .....	9
1.4.3.    Artefacts et Outils .....	10
1.5.    Conclusion .....	10
<b>Chapitre 2</b> Étude Théorique et choix des outils utilisés.....	11
2.1.    Introduction .....	12
2.2.    Paradigmes Technologiques et Concepts Clés.....	12
2.3.    La Virtualisation : Plusieurs appartements dans un seul immeuble .....	13
2.4.    L'Infrastructure en tant que Code (IaC) : Le plan d'architecte pour nos serveurs ....	13
2.5.    La Surveillance : Le tableau de bord de notre infrastructure .....	13
2.6.    La Sauvegarde : Notre assurance vie numérique .....	14

## Table de matières

---

2.7.	L'Intégration : Comment faire discuter Odoo et Proxmox ?.....	14
2.8.	La Méthode Agile Scrum : Construire le projet brique par brique.....	14
2.9.	Présentation et Justification des Outils et Technologies Clés .....	15
2.10.	Synthèse des Choix Stratégiques et Synergie des Outils .....	18
2.11.	Conclusion .....	18
	Chapitre 3 : Analyse de besoins et conduite du projet.....	19
3.1.	Introduction.....	20
3.2.	Identification des Acteurs.....	20
3.3.	Analyse des Exigences .....	21
3.3.1.	Besoins fonctionnels.....	21
3.3.2.	Besoins Non Fonctionnels .....	22
3.3.3.	Product Backlog Initial .....	23
3.3.4.	Répartition du Backlog en Sprints .....	26
3.4.	Vision d'ensemble des Fonctionnalités .....	27
3.4.1.	Diagramme des Cas d'Utilisation Global .....	27
3.4.2.	Diagramme d'activité global .....	28
3.5.	Conception de l'Architecture Technique Cible .....	30
3.5.1.	Architecture physique .....	30
3.5.2.	Architecture Logique .....	32
3.5.3.	Conception du Réseau Client et de la Sécurité.....	35
3.5.4.	Conception Initiale du diagramme de classes.....	37
3.6.	Planification des sprints .....	38
3.7.	Conclusion .....	38
	Chapitre 4 : Préparation de l'Environnement et des Outils .....	39
4.1.	Introduction.....	40
4.2.	Mise en Place de l'Infrastructure d'Hébergement .....	40
4.2.1.	Configuration de l'Infrastructure Physique .....	40
A.	Préparation Matérielle et Configuration Initiale d'iDRAC .....	40
B.	Activation et Validation des Accès de Gestion .....	40
4.2.2.	Déploiement des Plateformes de Virtualisation et de Sauvegarde.....	42
A.	Installation de Proxmox VE sur le Serveur de Production (SERV01) .....	42
B.	Déploiement du Proxmox Backup Server (SERV02) .....	43
4.3.	Mise en Place de la Plateforme de Gestion Odoo.....	45

4.3.1.	Installation et Configuration de Base d'Odoo.....	45
4.3.2.	Configuration des Dépendances Critiques.....	46
4.3.3.	Création du Catalogue de Produits et de sa Tarification .....	48
4.3.4.	Initialisation de l'interface Web .....	50
4.3.5.	Initialisation du Module Odoo .....	50
4.4.	Déploiement des Services de Support et de Sécurité .....	51
4.4.1.	Déploiement de la VM Reverse Proxy (NGINX) .....	51
4.4.2.	Déploiement de la Pile de Monitoring.....	51
4.5.	Préparation des Artefacts pour l'Automatisation.....	53
4.5.1.	Installation des Outils d'Infrastructure as Code.....	53
4.5.2.	Installation des Outils et Dépendances .....	54
4.5.3.	Mise en Place de l'Identité d'Automatisation pour Ansible .....	54
4.5.4.	Préparation des Images et Templates sur Proxmox.....	55
4.5.5.	Déploiement Initial de la VM Bastion .....	57
4.6.	Conclusion .....	58
	Chapitre 5 Sprint 1 : Fondation et Parcours Clients (MVP) .....	59
5.1.	Introduction .....	60
5.2.	Objectif attendu .....	60
5.3.	Analyse Fonctionnelle : Définition des Besoins du Sprint .....	60
5.3.1.	Identification des Acteurs et de leurs Rôles .....	60
5.3.2.	Modélisation des Cas d'Utilisation.....	61
5.3.3.	Spécifications Détaillées des Cas d'Utilisation.....	62
5.3.3.1.	Spécifications des Cas d'Utilisation de l'Administrateur .....	62
5.3.3.2.	Spécifications des Cas d'Utilisation du Client Final .....	68
5.4.	Planification du Développement (Sprint Planning) .....	71
5.4.1.	Périmètre du Sprint (Sprint Backlog) .....	72
5.4.2.	Plan d'Action .....	73
5.5.	Conception Technique et Architecture de la Solution .....	75
5.5.1.	Architecture du Modèle de Données (Diagramme de Classes) .....	75
5.5.2.	Architecture Comportementale (Diagrammes de Séquence).....	77
5.6.	Réalisation .....	80
5.6.1.	Étape 1 : Configuration par l'Administrateur (US-07, US-08) .....	80
5.6.2.	Étape 2 : Le Parcours d'Achat Client (US-01) .....	82

5.6.3.	Étape 3 : Le Déclenchement Automatisé (US-13).....	83
5.6.4.	Étape 4 : Provisionnement du Compte et Notification Client (US-18, US-03, US-02) .....	83
5.6.5.	Étape 5 : Déploiement de la Machine Virtuelle (US-24) .....	85
5.6.6.	Étape 6 : Interfaces de Supervision (US-11, US-12) .....	86
5.6.7.	Fonction Transverse : Fiabilisation par la Synchronisation.....	87
5.7.	Bilan et Rétrospective du Sprint (Sprint Review).....	88
5.7.1.	Atteinte de l'Objectif et Synthèse des Acquis.....	88
5.7.2.	Leçons Apprises et Décisions d'Adaptation .....	88
5.8.	Conclusion .....	89
	Chapitre 6 Sprint 2 : Isolation Client et Gestion Réseau .....	90
6.1.	Introduction : De la Preuve de Concept à la Sécurisation .....	91
6.2.	Objectif du Sprint (Sprint Goal).....	91
6.3.	Analyse Fonctionnelle.....	91
6.3.1.	Identification des Acteurs et de leurs Nouveaux Rôles .....	91
6.3.2.	La Stratégie d'Allocation Différenciée.....	92
6.3.3.	Évolution des Cas d'Utilisation.....	92
6.3.4.	Spécifications Détaillées des Nouveaux Cas d'Utilisation.....	93
6.4.	Planification du Sprint .....	94
6.4.1.	Périmètre du Sprint (Sprint Backlog) .....	94
6.4.2.	Plan d'Action.....	95
6.5.	Conception Technique : L'Architecture de la Connectivité Client .....	96
6.5.1.	Le Processus Manuel de l'Administrateur : .....	96
6.5.2.	Évolution du Modèle de Données .....	97
6.5.3.	Architecture Comportementale : Le Processus d'Allocation Logique.....	98
6.6.	Réalisation .....	99
6.6.1.	Étape 1 : Le Travail de l'Administrateur sur l'Infrastructure (Processus Manuel) (US-16 , US-17).....	99
6.6.2.	Étape 2 : Configuration de la Stratégie Réseau dans Odoo (US-10) .....	101
6.6.3.	Étape 3 : Allocation Automatique et Déploiement Isolé (US-19, US-15, US-04) .....	102
6.6.4.	Étape 4 : Supervision des Réseaux par l'Admin (US-11) .....	103
6.6.5.	Étape 5 : Gestion de l'Interaction Client (US-05) .....	104
6.7.	Bilan et Rétrospective du Sprint (Sprint Review).....	105

6.7.1.	Atteinte de l'Objectif et Synthèse des Acquis.....	105
6.7.2.	Leçon Apprise : La Limite du Processus Manuel.....	105
6.8.	Conclusion .....	105
Chapitre 7 Sprint 3 : Orchestration de l'Infrastructure (IaC).....		106
7.1.	Introduction.....	107
7.2.	Objectif du Sprint (Sprint Goal).....	107
7.3.	Analyse Fonctionnelle et Planification du Sprint .....	107
7.3.1.	Analyse des Nouveaux Besoins.....	107
7.3.2.	Périmètre du Sprint (Sprint Backlog) .....	108
7.3.3.	Plan d'Action.....	109
7.4.	Conception Technique : L'Architecture d'Orchestration IaC .....	110
7.4.1.	Vue d'Ensemble de l'Architecture d'Orchestration .....	110
7.4.2.	Le Flux d'Exécution Automatisé (Diagramme de Séquence) .....	111
7.4.3.	Évolution du Modèle de Données (Sprint 3) .....	112
7.5.	Réalisation .....	114
7.5.1.	Étape 1 : Configuration des Outils d'Orchestration (US-31).....	114
7.5.2.	Étape 2 : Automatisation de l'Infrastructure Réseau via Ansible (US-20) .....	115
7.5.3.	Étape 3 : Automatisation de l'Accès Sécurisé via Ansible (US-21) .....	115
7.5.4.	Étape 4 : Déploiement de la VM via Terraform et Cloud-Init (US-22, US-23)	116
7.6.	Bilan et Rétrospective du Sprint (Sprint Review).....	119
7.6.1.	Atteinte de l'Objectif et Synthèse des Acquis.....	119
7.6.2.	Leçons Apprises : Complexité de l'Orchestration .....	119
7.7.	Conclusion .....	120
Chapitre 8 Sprint 4 : Monitoring et Fiabilisation.....		121
8.1.	Introduction.....	122
8.2.	Objectif du Sprint (Sprint Goal).....	122
8.3.	Analyse Fonctionnelle.....	122
8.3.1.	Identification des Acteurs .....	122
8.3.2.	Spécifications Détaillées des Nouveaux Cas d'Utilisation.....	124
8.4.	Planification du Sprint .....	125
8.4.1.	Périmètre du Sprint (Sprint Backlog) .....	125
8.4.2.	Plan d'action du Sprint .....	126
8.5.	Conception Technique : .....	126

## Table de matières

---

8.6.	Réalisation .....	127
8.7.	Bilan et Rétrospective du Sprint (Sprint Review).....	129
8.8.	Conclusion .....	129
	<b>Chapitre 9 Sprint 5 : Sauvegarde et Reprise d'Activité .....</b>	<b>130</b>
9.1.	Introduction.....	131
9.2.	Objectif du Sprint (Sprint Goal).....	131
9.3.	Analyse Fonctionnelle.....	131
9.4.	Planification du Sprint .....	134
9.5.	Conception Technique : L'Architecture de la Résilience .....	135
9.6.	Réalisation .....	136
9.7.	Bilan et Rétrospective du Sprint (Sprint Review).....	137
9.8.	Conclusion .....	138
	<b>Conclusion Générale .....</b>	<b>139</b>
	<b>Webographie .....</b>	<b>141</b>

# Table des figures

Figure 1 : Organigramme de l'entreprise.....	5
Figure 2 : Diagramme des Cas d'utilisation Global .....	27
Figure 3 : Diagramme d'activités global Phase 1 & Phase 2.....	28
Figure 4 : Diagramme d'activités Phase 1 .....	29
Figure 5 : Diagramme d'activités Phase 2 .....	29
Figure 6 : Architecture physique de la solution .....	30
Figure 7 : Architecture Logique .....	32
Figure 8 : Diagramme de classes .....	37
Figure 9 : Planification des Sprints .....	38
Figure 10 : Accès à l'interface réussi.....	41
Figure 11 : Accès SSH Activé .....	41
Figure 12 : Écran de connexion de l'interface web de Proxmox VE.....	42
Figure 13 : Écran de connexion de l'interface web de Proxmox Backup Server.....	43
Figure 14 : Ajout d'un datastore.....	43
Figure 15 : Résumé du Datastore créé dans l'interface de Proxmox Backup Server.....	44
Figure 16 : Ajout de "Proxmox Backup Server" .....	44
Figure 17 : Stockage PBS visible dans la liste des stockages de Proxmox VE .....	44
Figure 18 : Configuration clé du fichier odoo.conf .....	45
Figure 19 : Tableau de bord des applications Odoo installés .....	46
Figure 20 : Écran de Configuration du Serveur de Messagerie dans Odoo .....	46
Figure 21 : Interface de supervision du module "Queue Job" .....	47
Figure 22 : Journaux et plans comptables chargés .....	47
Figure 23 : Liste des Attributs créés pour la personnalisation des VMs) .....	48
Figure 24 : Configuration des Prix Supplémentaires pour un Attribut.....	49
Figure 25 : Configuration de produits avec les attributs appropriés .....	49
Figure 26 : Initialisation de la Page d'Accueil du Site Web .....	50
Figure 27 : Arborescence de module .....	50
Figure 28 : Extrait de la configuration NGINX redirigeant vers Odoo .....	51
Figure 29 : Point de terminaison Prometheus actif .....	51
Figure 30 : Connectivité avec Prometheus réussie.....	52
Figure 31 : Alertmanager est opérationnel.....	52
Figure 32 : Node Exporter actif .....	52
Figure 33 : Pve Exporter actif.....	53

## Table de figures

---

Figure 34 : Terraform et Ansible Installés .....	53
Figure 35 : Clé SSH générée utilisée par Ansible.....	54
Figure 36 : Images ISO déployées sur le stockage de Proxmox .....	55
Figure 37 : Template « Cloud-Ready » visible dans l'interface Proxmox .....	56
Figure 38 : Template Ubuntu configuré .....	57
Figure 39 : VM Bastion dans la liste des VMs de Proxmox .....	57
Figure 40 : Diagramme des Cas d'Utilisation - Rôle Administrateur (Sprint 1) .....	61
Figure 41 : Diagramme des Cas d'Utilisation - Rôle Client (Sprint 1) .....	62
Figure 42 : Diagramme de classes dans le périmètre du sprint 1.....	76
Figure 43 : Séquence 1 - Déclenchement Automatisé par le Paiement .....	77
Figure 44 : Séquence 2 - Provisionnement d'un compte client sur Proxmox .....	78
Figure 45 : Séquence 3 - Déploiement de la Machine Virtuelle .....	79
Figure 46 : Section Intégration Proxmox .....	80
Figure 47 : Interface de Configuration de l'Intégration Proxmox .....	81
Figure 48 : Fiche Produit Configurée pour le Déploiement VM.....	82
Figure 49 : Page Produit avec Sélection des Variantes de la VM.....	82
Figure 50 : Suivi et Déclenchement Manuel depuis la Facture.....	83
Figure 51 : Preuve de la Création du Compte sur Proxmox .....	84
Figure 52 : Retour : Identifiants Proxmox enregistrés dans la fiche client Odoo.....	84
Figure 53 : Email de bienvenue envoyé au client avec ses identifiants Proxmox.....	85
Figure 54 : Portail d'Accès Proxmox Présenté au Client .....	85
Figure 55 : Preuve du Déploiement de la VM sur l'Interface Proxmox .....	86
Figure 56 : Vue Centralisée du Parc de Machines Virtuelles dans Odoo.....	86
Figure 57 : Tableau Récapitulatif des VMs sur la Fiche Client. ....	86
Figure 58 : Action Manuelle de Synchronisation du Statut des VMs.....	87
Figure 59 : Configuration de la Tâche Planifiée (CRON) pour la Synchronisation Automatique. ....	87
Figure 60 : Diagramme des Cas d'Utilisation - Rôle Administrateur (Sprint 2) .....	92
Figure 61 : Diagramme d'Activité - Configuration Manuelle du Réseau et de la Sécurité.....	96
Figure 62 : Évolution du Diagramme de Classes (Focus Réseau).....	97
Figure 63 : Séquence d'Allocation d'un Nouveau Réseau Client.....	98
Figure 64 : Configuration du Bridge Réseau sur Proxmox .....	99
Figure 65 : Configuration des Interfaces d'Écoute du Service DHCP. ....	100
Figure 66 : Configuration des Pools d'Adresses DHCP .....	100
Figure 67 : Configuration des Règles de NAT via iptables .....	100
Figure 68 : Configuration de l'Accès Client sur le Bastion SSH.....	101
Figure 69 : Interface de Configuration des Super-Réseaux.....	101
Figure 70 : Preuve de la Création d'un Réseau Client Logique dans Odoo .....	102
Figure 71 : Preuve du Déploiement de la VM sur le Bridge Isolé .....	102
Figure 72 : Email d'Information Réseau Envoyé au Client .....	103
Figure 73 : Vue Centralisée des Réseaux Clients Alloués .....	103
Figure 74 : Email de Choix de Réseau Envoyé au Client Existant .....	104
Figure 75 : Diagramme d'Architecture Globale (Odoo + IaC) .....	110

## Table de figures

---

Figure 76 : Séquence Complète de l'Orchestration IaC .....	111
Figure 77 : Évolution du Diagramme de Classes (Focus IaC) .....	112
Figure 78 : Nouveaux Paramètres de Configuration IaC dans Odoo.....	114
Figure 79 : Preuve de la Création du Bridge par Ansible sur un Nœud Proxmox.....	115
Figure 80 : Utilisateur jumper créé sur le bastion avec succès.....	115
Figure 81 : Preuve du Déploiement de la VM sur son Bridge Privé par Terraform .....	116
Figure 82 : Preuve de la Connexion à la Console via le Mot de Passe Généré .....	116
Figure 83 : Vérification de l'Installation du Service Additionnel (Prometheus Node Exporter) .....	117
Figure 84 : Vérification de l'Injection de la Clé Publique SSH du Client .....	117
Figure 85 : Email d'Accès Final à la VM) .....	118
Figure 86 : Diagramme des Cas d'Utilisation - Rôle Administrateur (Sprint 4) .....	123
Figure 87 : Vue du Tableau de Bord Global de l'Infrastructure sur Grafana .....	127
Figure 88 : Vue Détallée des Performances d'une VM Cliente sur Grafana .....	128
Figure 89 : Notification d'Alerte Reçue sur le Canal Slack .....	128
Figure 90 : Diagramme des Cas d'Utilisation - Rôle Administrateur ((Sprint 5)) .....	132
Figure 91 : Séquence - Sauvegarde Automatique d'une VM.....	135
Figure 92 : Diagramme de Séquence - Restauration d'une VM Cliente .....	136
Figure 93 : Configuration de la Tâche de Sauvegarde dans Proxmox VE.....	136
Figure 94 : Journal d'une Sauvegarde Réussie vers PBS .....	137

# Liste des Tableaux

Tableau 1 : Comparaison entre Proxmox VE et VMware vSphere.....	15
Tableau 2 : Comparaison entre Odoo et les Différents alternatives .....	16
Tableau 3 : Comparaison des outils d'automatisation .....	16
Tableau 4 : Comparaison entre les outils de surveillances .....	17
Tableau 5 : Comparaison entre les outils de Backups.....	17
Tableau 6 : Description des acteurs .....	20
Tableau 7 : Besoins Fonctionnels.....	22
Tableau 8 : Besoins non Fonctionnels .....	23
Tableau 9 : Backlog de produit .....	25
Tableau 10 : Tableau de Planification.....	26
Tableau 11 : Exemple d'Allocation de Sous-réseaux.....	36
Tableau 12 : Spécification du Cas d'Utilisation 'Configurer l'intégration' .....	63
Tableau 13 : Spécification du Cas d'Utilisation "Désigner produit VM" .....	64
Tableau 14 : Spécification du Cas d'Utilisation 'Visualiser le parc VMs' .....	64
Tableau 15 : Spécification du Cas d'Utilisation 'Consulter fiche client' .....	65
Tableau 16 : Spécification du Cas d'Utilisation 'Relancer déploiement' .....	66
Tableau 17 : Spécification du Cas d'Utilisation 'Synchroniser statut VMs' .....	67
Tableau 18 : Spécification du Cas d'Utilisation 'Parcourir le catalogue' .....	68
Tableau 19 : Spécification du Cas d'Utilisation 'Commander une VM' .....	69
Tableau 20 : Spécification du Cas d'Utilisation 'Recevoir ses accès (email)' .....	70
Tableau 21 : Spécification du Cas d'Utilisation 'Se connecter au portail Proxmox' .....	71
Tableau 22 : User Stories du Sprint 1 .....	72
Tableau 23 : Décomposition des Tâches Techniques du Sprint 1 .....	73
Tableau 24 : Spécification du Cas d'Utilisation 'Définir la stratégie réseau' .....	93
Tableau 25 : Spécification du Cas d'Utilisation 'Superviser les réseaux alloués' .....	94
Tableau 26 : User Stories du Sprint 2 .....	94
Tableau 27 : User Stories du Sprint 3 .....	108
Tableau 28 : Décomposition des Tâches Techniques du Sprint 3 .....	109
Tableau 29 : Spécification du Cas d'Utilisation 'Superviser les performances de la plateforme'	124
Tableau 30 : Spécification du Cas d'Utilisation 'Être notifié d'un incident critique' .....	124
Tableau 31 : User Stories du Sprint 4 .....	125
Tableau 32 : Plan d'action de sprint 4 .....	126
Tableau 33 : Spécification du Cas d'Utilisation 'Gérer la politique de sauvegarde' .....	133

## Liste des Tableaux

---

Tableau 34 : Spécification du Cas d'Utilisation 'Exécuter le Plan de Reprise d'Activité'.....	133
Tableau 35 : User Story de sprint 5 .....	134
Tableau 36 : Plan d'action de sprint 5 .....	134

# Liste d'abréviation

Abréviation	Signification
<b>API</b>	Application Programming Interface
<b>ACL</b>	Access Control List
<b>CIDR</b>	Classless Inter-Domain Routing
<b>CRON</b>	Command Run ON
<b>CRUD</b>	Create, Read, Update, Delete
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>DNS</b>	Domain Name System
<b>ERP</b>	Enterprise Resource Planning
<b>IaC</b>	Infrastructure as Code
<b>ID</b>	Identifier
<b>ISO</b>	International Organization for Standardization
<b>IaaS</b>	Infrastructure as a Service
<b>JSON</b>	JavaScript Object Notation
<b>KVM</b>	Kernel-based Virtual Machine
<b>MVP</b>	Minimum Viable Product
<b>NAT</b>	Network Address Translation
<b>PBS</b>	Proxmox Backup Server
<b>PVE</b>	Proxmox Virtual Environment
<b>PRA</b>	Plan de Reprise d'Activité
<b>RAM</b>	Random-Access Memory
<b>RPO</b>	Recovery Point Objective
<b>RTO</b>	Recovery Time Objective
<b>SSH</b>	Secure Shell
<b>SMTP</b>	Simple Mail Transfer Protocol
<b>UML</b>	Unified Modeling Language
<b>URL</b>	Uniform Resource Locator
<b>US</b>	User Story
<b>VM</b>	Virtual Machine
<b>XML</b>	eXtensible Markup Language

# Introduction Générale

Dans un monde où l'accès à des ressources informatiques flexibles et à la demande est primordial, la virtualisation s'est affirmée comme la virtualisation s'est affirmée comme pilier essentiel des infrastructures informatiques modernes, en permettant une utilisation optimale des ressources matérielles, une flexibilité accrue, et une meilleure gestion des services, que ce soit en environnement local ou dans le cloud. Cependant, pour les fournisseurs de services d'hébergement, la complexité inhérente au provisionnement, à la configuration et à la gestion continue de ces environnements virtualisés représente un défi constant.

Face à cette exigence croissante d'agilité et de maîtrise des coûts, l'**automatisation** des processus de gestion d'infrastructure n'est plus une option mais une nécessité. Du provisionnement initial des ressources à leur configuration, leur surveillance et leur maintenance, chaque étape du cycle de vie des services virtualisés offre des opportunités d'optimisation. L'enjeu réside dans la capacité à orchestrer ces opérations complexes de manière fluide et intégrée, tout en garantissant la sécurité, la performance et une expérience utilisateur de qualité. C'est dans cette quête d'efficience et de services à valeur ajoutée que s'inscrit la nécessité de solutions logicielles capables de combler le fossé entre les besoins métiers et la gestion technique des infrastructures sous-jacentes.

Le présent Projet de Fin d'Études réalisé au sein de MAG Security aborde à cette problématique en explorant l'intégration d'une plateforme de gestion d'entreprise avec un système de virtualisation avancé. Il vise à démontrer comment l'orchestration logicielle en s'appuyant sur les principes de l'Infrastructure en tant que Code (IaC), peut transformer une infrastructure de virtualisation robuste en une offre de services rationalisée et accessible.

L'objectif est de développer un module d'intégration fonctionnel qui, à la suite d'une commande client, déclenche un pipeline de déploiement et de configuration entièrement automatisé, depuis la requête sur le portail commercial jusqu'à la livraison d'un environnement virtualisé prêt à l'emploi.

Ce rapport documente la démarche suivie et les résultats obtenus au cours de notre stage, en adoptant une méthodologie Agile Scrum pour une progression itérative et adaptative.

Il s'articule comme suit :

- Le **premier chapitre** posera le cadre général du projet, en présentant le contexte, les objectifs et la méthodologie adoptée.
- Le **deuxième chapitre** établira les fondations théoriques et technologiques, en explorant les concepts clés qui sous-tendent notre solution.
- Le **troisième chapitre** sera dédié à l'analyse des besoins et à la conception de l'architecture cible, définissant ainsi le plan directeur de notre développement.
- Le **quatrième chapitre** décrira la préparation de l'environnement de travail et des outils, socle technique et méthodologique de notre projet.
- Le cœur du rapport, **des chapitres cinq à neuf**, retracera la réalisation incrémentale de la solution à travers cinq sprints successifs, de la preuve de concept initiale à l'automatisation complète, en passant par l'ajout des couches de monitoring et de sauvegarde.
- Enfin, une **Conclusion Générale** synthétisera les travaux réalisés, évaluera l'atteinte des objectifs et proposera des pistes d'évolution pour la plateforme développée.

# Chapitre 1

## Cadre général du projet

## 1.1. Introduction

Ce premier chapitre a pour objectif de présenter le cadre dans lequel s'est déroulé notre Projet de Fin d'Études. Nous commencerons par une présentation de l'organisme d'accueil, afin de comprendre son secteur d'activité et ses enjeux. Nous détaillerons ensuite la problématique qui a motivé ce projet, en définissant clairement ses objectifs, son périmètre et les livrables attendus. Enfin, nous exposerons la méthodologie de gestion de projet Agile Scrum qui a été adoptée pour structurer et piloter la réalisation de manière itérative et adaptative.

## 1.2. Présentation de l'organisme d'accueil

### 1.2.1. Présentation générale de MAG Security

MAG Security est une entreprise tunisienne spécialisée dans l'étude, l'intégration et l'installation de systèmes de sécurité électroniques. Sa mission consiste à rendre l'environnement de travail agréable en veillant aux conditions de travail et à la pénibilité, car ces éléments sont considérés comme des leviers importants pour motiver et mobiliser. Forte d'une expérience de plus de 9 ans, MAG Sécurité propose des produits et des solutions sur mesure destinés à la fois aux professionnels et aux particuliers.

### 1.2.2. Localisation de MAG Security

Le siège social et le principal showroom de MAG Security sont situés en banlieue sud de Tunis, à une adresse stratégique :

**Adresse :** Immeuble Marina Centre, Avenue Hedi Nouira, Route du Port, Rades, Tunisie.

Cette localisation permet une accessibilité aisée pour sa clientèle et une bonne desserte logistique. Le showroom est un espace dédié à la démonstration des produits et des solutions, où les clients peuvent bénéficier de conseils d'experts pour leurs projets de sécurité.

### 1.2.3. Organigramme de MAG Security

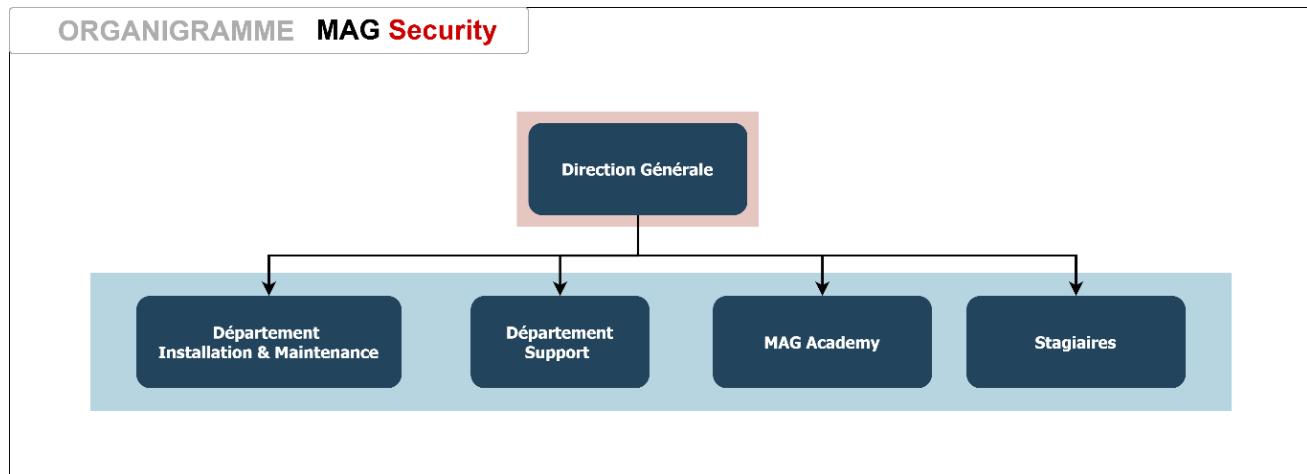


Figure 1 : Organigramme de l'entreprise

L'organigramme de MAG Security reflète une structure dynamique et collaborative, mettant en avant :

- **Direction Générale** : Supervision des opérations et stratégie globale.
- **Département Technique** : Installation et maintenance des systèmes de sécurité.
- **Département Support** : Relation client et support.
- **MAG Academy** : Formation pour les professionnels et utilisateurs de systèmes de sécurité.

### 1.2.4. Environnement de l'entreprise

#### a. Participants internes

- Équipe Technique : Techniciens spécialisés dans l'installation et la maintenance.
- Équipe Commerciale : Chargés de clientèle et experts en solutions de sécurité.
- Direction : Prise de décisions stratégiques et gestion des opérations.

#### b. Participants externes

- Clients : Particuliers et entreprises nécessitant des solutions de sécurité.
- Fournisseurs : Partenaires technologiques fournissant les équipements.
- Formateurs : Experts intervenant dans la formation continue.

### 1.2.5. Services MAG Security

MAG Sécurité se distingue par une offre de services complète et diversifiée, visant à couvrir l'ensemble des besoins en sécurité électronique et en solutions technologiques associées. Ses principales prestations incluent :

- **Systèmes de Sécurité Electroniques** : Installation et maintenance de systèmes de vidéosurveillance (CCTV analogique et IP), contrôle d'accès (biométrique, par badges, interphonie), systèmes d'alarme anti-intrusion, détection de mouvement, et portiques de sécurité.
- **Équipements de Sécurité Incendie** : Solutions de détection incendie (conventionnelle et adressable), systèmes d'extinction automatique, signalisation de sécurité, et plans d'évacuation.
- **Tenues et Habillements de Sécurité Professionnels** : Fourniture d'équipements de protection individuelle (EPI) et de tenues professionnelles adaptées aux métiers de la sécurité et à d'autres secteurs.
- **Standard Téléphonique et Solutions VoIP** : Installation et configuration de standards téléphoniques PABX/IPBX, solutions de téléphonie sur IP (VoIP) pour entreprises, et systèmes de visioconférence.
- **Solutions de Sonorisation et Télédistribution** : Systèmes de sonorisation de confort et de sécurité (Public Address), et solutions de télédistribution pour collectivités et entreprises.
- **Électricité Bâtiment** : Travaux d'installation et de maintenance électrique courant fort et courant faible pour bâtiments résidentiels, tertiaires et industriels.

Cette polyvalence permet à MAG Sécurité de se positionner comme un interlocuteur unique pour des projets d'équipement technologique complexes.

### 1.2.6. Partenaires MAG Security

Afin de garantir la qualité, la fiabilité et la pérennité des solutions proposées, MAG Security a tissé des liens de partenariat solides avec des fabricants et distributeurs de renommée internationale. Parmi ses partenaires technologiques majeurs, on peut citer (liste non exhaustive) :

- **Vidéosurveillance et Contrôle d'Accès** : Uniview, Dahua Technology, Hikvision, ZKTeco.
- **Systèmes d'Alarme et Détection Incendie** : Teletek Electronics, Inim Electronics, AMC Elettronica.

- **Solutions VoIP et Télécommunication** : Grandstream Networks, Yalink.
- **Autres Équipements Spécifiques** : Zavag (probablement pour des solutions spécifiques).

Ces partenariats permettent à MAG Security de bénéficier d'un accès privilégié aux dernières innovations, à un support technique avancé, et à des conditions commerciales avantageuses, se répercutant positivement sur les solutions offertes aux clients.

### 1.3. Présentation du projet

À la suite de la présentation de MAG Security, l'organisme d'accueil où nous avons mené notre projet de fin d'études, nous allons aborder le contexte de ce projet ainsi que la solution que nous avons proposée. Ce stage s'inscrit dans le cadre de l'obtention du diplôme national de licence en Technologies de l'Informatique à l'Institut Supérieur des Études Technologiques de Radès. La réalisation de ce projet s'est déroulée du 03 février 2024 au 05 juin 2024.

#### 1.3.1. Contexte du projet et problématique

Historiquement, le processus de mise à disposition de machines virtuelles (VMs) au sein de **MAG Security**, que ce soit pour des besoins internes ou pour les premiers clients pilotes, s'appuyait sur des opérations manuelles. Cette approche, bien que fonctionnelle à une échelle limitée, a révélé des freins significatifs à l'ambition de l'entreprise de développer une offre de services de virtualisation compétitive et scalable.

La problématique centrale peut se résumer en quatre points critiques :

- **Processus chronophage** : Chaque déploiement requiert l'intervention manuelle d'un technicien pour créer la VM, configurer le réseau, et gérer les accès, ce qui entraîne des délais de livraison non compétitifs.
- **Risque d'erreurs humaines** : La nature manuelle des configurations augmente la probabilité d'incohérences ou de failles de sécurité dans les environnements livrés.
- **Manque de scalabilité** : Le modèle opérationnel actuel est incapable de supporter une augmentation significative du volume de commandes.
- **Absence d'interface client intégrée** : Les clients ne disposent d'aucun portail pour commander, personnaliser ou suivre leurs services, ce qui limite leur autonomie et nécessite une communication constante avec les équipes techniques.

### 1.3.2. Solution proposée

Face à ces constats, MAG Security a identifié un besoin stratégique de passer d'un modèle artisanal à un modèle industriel. Ce projet de fin d'études s'inscrit donc directement dans cette démarche en visant à automatiser de bout en bout la chaîne de provisionnement. L'objectif est de répondre à cette problématique en intégrant la plateforme de gestion d'entreprise Odoo, utilisée comme interface commerciale et chef d'orchestre, avec notre hyperviseur Proxmox VE.

### 1.3.3. Objectifs du projet

L'objectif principal de ce projet est de concevoir et de développer une solution d'orchestration complète et automatisée pour la fourniture de services de machines virtuelles.

Pour atteindre cet objectif global, nous avons défini les quatre objectifs spécifiques suivants :

#### 1. Développer une plateforme de vente et de gestion intégrée :

- Mettre en place une interface e-commerce sur **Odoo** permettant aux clients de commander et de personnaliser des VMs.
- Fournir aux administrateurs une vue centralisée dans Odoo pour superviser les déploiements et configurer l'intégration.
- Automatiser les notifications par mails pour informer les clients de l'état de leur compte Proxmox, ainsi que des factures et des commandes

#### 2. Automatiser le provisionnement de l'infrastructure via les principes IaC :

- Standardiser la création des VMs en s'appuyant sur des templates préconfigurés ("Cloud-Ready") et des mécanismes de personnalisation comme Cloud-Init.
- Déclencher automatiquement le déploiement d'une VM complète à la suite d'une commande client validée.
- Utiliser des outils d'Infrastructure as Code (comme Terraform et Ansible) pour garantir des déploiements fiables, reproductibles et rapides.

### 3. Garantir la sécurité et l'isolation des environnements clients :

- Mettre en œuvre une architecture réseau qui isole chaque client dans un sous-réseau privé.
- Sécuriser tous les accès externes aux VMs via une machine bastion (Jump Host).
- Automatiser la gestion des identités et des clés d'accès SSH pour chaque client.

### 4. Assurer la fiabilité et la supervision du service :

- Déployer une pile de monitoring (basée sur Prometheus et Grafana) pour superviser en temps réel la santé de l'infrastructure et des VMs clientes.
- Mettre en place une stratégie de sauvegarde robuste (via Proxmox Backup Server) et définir des procédures de restauration pour protéger les données des clients.

## 1.4. Choix de la Méthodologie

### 1.4.1. Justification du Choix

Compte tenu de la nature innovante et de la complexité technique du projet, la méthodologie **Agile Scrum** a été retenue. Cette approche itérative et incrémentale était la plus adaptée pour gérer les incertitudes, permettre des ajustements rapides et livrer de la valeur de manière continue. Elle favorise la collaboration, la transparence et l'adaptation, des facteurs clés de succès pour ce type de projet.

### 1.4.2. Organisation de l'Équipe et Rôles

Notre équipe projet a fonctionné selon les rôles suivants :

- **Product Owner** : Ce rôle a été assuré par notre encadrant professionnel, M.Mohamed Achref Gomri, qui a défini la vision globale et priorisé les fonctionnalités à développer.
- **Équipe de Développement** : Composée de nous deux, Firas Taboubi et Youssef Ben Ghorbel. Nous étions collectivement responsables de l'ensemble du cycle de développement : de la conception technique à la réalisation et aux tests des fonctionnalités prévues pour chaque sprint

Pour assurer le bon déroulement du projet, nous avons **partagé les responsabilités** typiquement associées au rôle de Scrum Master. Concrètement, nous nous chargions ensemble de l'organisation des réunions, du suivi de notre progression, et de la résolution des obstacles techniques que nous rencontrions.

#### 1.4.3. Artefacts et Outils

Pour appliquer cette méthodologie, nous nous sommes appuyés sur plusieurs artefacts et outils :

- **Artefacts Scrum** : Un **Product Backlog** a été utilisé pour lister l'ensemble des fonctionnalités. Pour chaque itération, un **Sprint Backlog** était défini.
- **Outils de Gestion** : Le suivi des tâches a été réalisé à l'aide de l'outil en ligne **Jira**.

#### 1.5. Conclusion

Ce premier chapitre a établi le contexte et le cadre de notre projet. Nous avons présenté l'entreprise, défini la problématique à résoudre, les objectifs visés, ainsi que la méthodologie agile qui a guidé nos travaux. Le chapitre suivant s'attachera à présenter les fondements théoriques ainsi que les choix technologiques qui constituent ayant guidé la conception de la solution proposée.

# Chapitre 2 Étude Théorique et choix des outils utilisés

## 2.1. Introduction

Ce chapitre a pour objectif de présenter les fondements théoriques et conceptuels qui structurent ce projet de fin d'études.

Il vise à établir une compréhension solide des principes qui régissent les technologies mises en œuvre et à justifier de manière exhaustive les choix d'outils qui architectureront la solution pour MAG Security.

L'importance de ces concepts est capitale, car ils garantissent la cohérence, la maintenabilité et la pertinence de la solution développée.

Nous présenterons les paradigmes technologiques, la méthodologie de gestion de projet adoptée, et enfin, chaque composant de la pile technologique retenue.

## 2.2. Paradigmes Technologiques et Concepts Clés

Pour bien comprendre notre projet, il faut d'abord parler des grandes idées et des technologies qui le composent. Voyons-les une par une, de manière simple.

### Les différents niveaux de service : L'analogie de la Pizza

Imaginez que vous voulez manger une pizza. Vous avez plusieurs options, qui correspondent aux modèles de service informatique :

- **Fait maison (Infrastructure sur site)** : Vous achetez la farine, la sauce, le fromage, vous avez votre propre four, votre propre cuisine... Vous contrôlez tout, mais c'est beaucoup de travail.
- **Pizza à emporter avec kit (IaaS - Infrastructure as a Service)** : On vous fournit la cuisine, le four et les ingrédients de base. C'est à vous d'assembler et de cuire la pizza comme vous le voulez.

**C'est le cœur de notre projet.** MAG Security fournit l'infrastructure (le "four" et les "ingrédients" virtuels) avec Proxmox, et le client "cuit" ce qu'il veut dessus.

- **Pizza livrée (PaaS - Platform as a Service)** : Vous commandez une pizza qui arrive chaude et prête à être mangée. Vous ne gérez ni la cuisine, ni la cuisson.
- **Manger au restaurant (SaaS - Software as a Service)** : Vous allez au restaurant, vous vous asseyez et on vous sert une pizza. Vous profitez du service complet sans vous soucier de rien.

Le portail client sur **Odoo** est un peu comme ça : le client l'utilise comme un logiciel prêt à l'emploi.

Enfin, tout cela se passe dans notre propre "cuisine" (infrastructure privée), ce qui nous donne un contrôle total sur la qualité et la sécurité.

### 2.3. La Virtualisation : Plusieurs appartements dans un seul immeuble

La virtualisation, c'est un peu comme construire un grand immeuble sur un terrain, au lieu de construire une seule maison.

- **Le serveur physique** est l'immeuble.
- **Une machine virtuelle (VM)** est un appartement.

Chaque appartement est isolé, a sa propre porte, sa propre cuisine, etc. mais tous partagent la structure de l'immeuble (électricité, eau...). La virtualisation permet de faire la même chose : faire fonctionner plusieurs "ordinateurs" (VMs) isolés sur une seule machine physique puissante.

C'est exactement ce que nous faisons avec **Proxmox VE**, notre gestionnaire d'immeuble.

### 2.4. L'Infrastructure en tant que Code (IaC) : Le plan d'architecte pour nos serveurs

Plutôt que de construire un serveur en cliquant partout ("mets un mur ici, une fenêtre là"), ce qui est long et source d'erreurs, on utilise l'IaC. C'est comme dessiner un plan d'architecte détaillé.

Ce plan (un fichier de code) décrit exactement à quoi le serveur doit ressembler. Ensuite, les "ouvriers" (nos outils d'automatisation) lisent le plan et construisent le serveur parfaitement, à chaque fois. Si on veut un deuxième serveur identique, on réutilise le même plan !

Un principe clé est l'**idempotence** : c'est comme appuyer sur le bouton d'un ascenseur. Que vous appuyiez une fois ou dix fois, le résultat est le même : l'ascenseur est appelé. Nos scripts fonctionnent pareil pour éviter les erreurs.

### 2.5. La Surveillance : Le tableau de bord de notre infrastructure

Imaginez le tableau de bord d'une voiture. Il vous indique la vitesse (l'usage du CPU), le niveau d'essence (l'espace disque), la température du moteur (la RAM), et il a des voyants rouges qui s'allument en cas de problème (**les alertes**).

Notre système de surveillance fait exactement ça pour notre infrastructure. On **collecte** les données, on les **affiche** sur des graphiques (avec Grafana), et on envoie une **alerte** si quelque chose ne va pas.

## 2.6. La Sauvegarde : Notre assurance vie numérique

Les données sont précieuses. Les protéger est une priorité. Pour cela, on se pose deux questions simples :

1. **RPO** : Si tout plante, quelle est la quantité de travail maximale (ex: 1 heure) que nous pouvons nous permettre de perdre ?
2. **RTO** : Combien de temps au maximum (ex: 4 heures) peut-on être hors service avant que ce soit une catastrophe ?

Pour répondre à ça, on suit **la règle du 3-2-1** : avoir **3** copies des données, sur **2** supports différents, dont 1 à un autre endroit.

## 2.7. L'Intégration : Comment faire discuter Odoo et Proxmox ?

Odoo (le site web) et Proxmox (l'usine à VM) ne parlent pas la même langue. Il leur faut un traducteur. Ce traducteur, c'est l'**API** (Application Programming Interface).

Pensez à un restaurant :

- Le client (ici, notre module Odoo) veut commander un plat.
- Il ne va pas dans la cuisine (Proxmox).
- Il parle au serveur (l'**API**). Il lui donne sa commande. Le serveur la transmet à la cuisine, puis rapporte le plat.

L'**API** permet aux deux systèmes de communiquer de manière standardisée et sécurisée, sans avoir besoin de connaître les détails complexes de l'autre. C'est ce qu'on appelle un couplage lâche, un peu comme des briques de Lego : chaque brique est indépendante mais peut s'emboîter avec n'importe quelle autre.

## 2.8. La Méthode Agile Scrum : Construire le projet brique par brique

Plutôt que de tout construire d'un coup en espérant que ça fonctionne à la fin (ce qui est très risqué), on utilise la méthode Scrum. C'est comme construire une maison en procédant par étapes courtes et validées :

1. **Sprint 1** : On construit les fondations solides. On vérifie qu'elles sont bonnes.
2. **Sprint 2** : On monte les murs. On vérifie.
3. **Sprint 3** : On pose le toit... et ainsi de suite.

Chaque étape (Sprint) dure quelques semaines et a un objectif clair. Cela nous permet de nous adapter, de corriger les problèmes tôt et de nous assurer que ce que nous construisons correspond bien au besoin. On s'appuie sur un Product Owner (celui qui a la vision), une

Development Team (les constructeurs) et un Scrum Master (celui qui aide l'équipe à bien travailler ensemble).

## 2.9. Présentation et Justification des Outils et Technologies Clés

Le passage d'un processus manuel à une offre de service automatisée exige des outils non seulement performants, mais aussi parfaitement alignés avec les contraintes et les ambitions de MAG Security. Chaque choix technologique a été fait après une analyse comparative.

- Proxmox VE est la plateforme de virtualisation open-source qui hébergera l'ensemble des services clients et internes.

Tableau 1 : Comparaison entre Proxmox VE et VMware vSphere

Critère	Notre Choix : Proxmox VE	Alternative : VMware vSphere
Modèle de Licence & Coût	Open-Source (AGPLv3). Gratuit. Support optionnel abordable.	Propriétaire. Coûts de licence élevés et complexes (par CPU, par an).
Flexibilité & Accès	Basé sur Debian. Accès total au système d'exploitation sous-jacent. Personnalisation illimitée.	Hyperviseur fermé (ESXi). Personnalisation limitée à l'API et aux outils fournis.
API pour l'Automatisation	API RESTful complète, bien documentée et nativement intégrée.	API robuste mais souvent liée à des licences spécifiques (vCenter).

Pour une PME comme MAG Security qui lance une nouvelle offre de service, la maîtrise des coûts initiaux est une nécessité absolue. Proxmox VE élimine la barrière financière des licences, permettant de concentrer les investissements sur le matériel. Sa nature open-source et son API complète offrent une flexibilité maximale pour construire une solution d'automatisation sur mesure, sans être contraint par un écosystème propriétaire. C'est le choix de la souveraineté technique et de la viabilité économique.

- Odoo est l'ERP qui servira à la fois de vitrine commerciale pour les clients et de cerveau pour l'orchestration du déploiement.

Tableau 2 : Comparaison entre Odoo et les Différents alternatives

Critère	Notre Choix : Odoo Community	Alternative 1 : WHMCS	Alternative 2 : SAP/Dynamics
Modèle Économique	Gratuit et open-source.	Commercial (abonnement).	Coûts d'implémentation et de licence très élevés.
Flexibilité d'Intégration	Totale. Développement de modules personnalisés en Python.	Limitée aux modules existants et à l'API fournie.	Extensible mais via des technologies propriétaires et complexes.
Suite Fonctionnelle	ERP complet (Vente, CRM, Facturation, Site Web).	Spécifique à la facturation d'hébergement.	Suite ERP extrêmement complète, souvent surdimensionnée.

MAG Security a besoin d'une solution qui soit à la fois un portail e-commerce et un orchestrateur technique. WHMCS, bien que spécialisé, est trop rigide pour permettre l'intégration profonde requise. Les grands ERP sont financièrement et techniquement démesurés. Odoo Community est le compromis parfait : il est gratuit, offre une base e-commerce/CRM fonctionnelle "out-of-the-box" et, surtout, sa nature extensible en Python permet de développer précisément le module d'intégration avec Proxmox, qui est le cœur de ce projet.

- Cette chaîne d'outils est responsable de la transformation d'une commande client en une VM fonctionnelle.

Tableau 3 : Comparaison des outils d'automatisation

Approche	Notre Choix : Terraform + Ansible	Alternative : Scripts Shell/Python Purs
Philosophie	Déclarative (Terraform) et procédurale (Ansible).	Purement impérative.
Gestion de l'État	Robuste et automatique (Terraform suit les ressources créées).	Manuelle et source d'erreurs.
Maintenabilité	Élevée. Code structuré et modulaire.	Faible. Difficile à maintenir et à faire évoluer.

S'appuyer sur des scripts manuels est la recette pour une dette technique ingérable. La chaîne Terraform + Ansible applique le principe de "séparation des préoccupations" : Terraform est l'expert du provisionnement (créer la VM), Ansible l'expert de la configuration (installer des logiciels dedans), et Cloud-Init gère la personnalisation initiale. Cette approche structurée est une nécessité pour construire un service fiable, prévisible et maintenable sur le long terme.

- La pile Prometheus (Prometheus, Grafana, Alertmanager) est déployée pour superviser la santé de la plateforme.

Tableau 4 : Comparaison entre les outils de surveillances

Critère	Notre Choix : Pile Prometheus	Alternative : TICK Stack (InfluxDB)
Modèle de Collecte	Pull (Tirage). Prometheus contacte les cibles. Simplifie la découverte.	Push (Poussée). Les agents envoient les métriques.
Écosystème & Communauté	Immense. Standard de facto, vaste choix d'exportateurs et de tableaux de bord.	Solide mais plus petit et centré sur InfluxData.

La fiabilité est un argument de vente. La pile Prometheus est le standard de l'industrie, garantissant une surveillance de niveau professionnel sans coût de licence. Son modèle "pull" et son immense communauté (notamment pour Grafana) permettent de mettre en place rapidement une supervision complète et personnalisée, ce qui est indispensable pour garantir la qualité de service aux clients.

- PBS est la solution dédiée à la protection des données des machines virtuelles.

Tableau 5 : Comparaison entre les outils de Backups

Critère	Notre Choix : Proxmox Backup Server	Alternative : Veeam
Intégration Proxmox	Native et parfaite. Gérée depuis l'interface Proxmox VE.	Tierce. Nécessite une configuration et une gestion séparées.
Sécurité	Chiffrement côté client (la clé ne quitte jamais l'hyperviseur).	Chiffrement robuste, mais souvent géré côté serveur de sauvegarde.
Coût	Gratuit (open-source).	Commercial (licence par VM ou instance).

Utiliser une solution de sauvegarde parfaitement intégrée n'est pas un luxe, c'est une nécessité pour la simplicité opérationnelle. PBS est conçu par Proxmox pour Proxmox. Cette intégration native, combinée à une sécurité de premier ordre (chiffrement côté client) et à l'absence de coût, en fait un choix non négociable pour une offre de service IaaS basée sur Proxmox.

## 2.10. Synthèse des Choix Stratégiques et Synergie des Outils

L'architecture technologique présentée ici n'est pas un simple assemblage d'outils, mais une réponse stratégique délibérée aux défis de MAG Security.

La nécessité de cette approche repose sur quatre piliers :

1. **Viabilité Économique** : En s'appuyant quasi exclusivement sur des solutions open-source (Proxmox, Odoo, Prometheus), MAG Security élimine la barrière à l'entrée que représentent les coûts de licence, rendant son offre compétitive.
2. **Efficacité Opérationnelle** : L'automatisation de bout-en-bout via la chaîne Odoo → Terraform → Ansible est une nécessité pour passer d'un modèle artisanal à un modèle industriel, capable de gérer un volume de clients sans augmenter proportionnellement les effectifs.
3. **Flexibilité et Souveraineté** : Le choix d'outils open-source et extensibles (Odoo en Python, Proxmox) garantit à MAG Security un contrôle total sur sa plateforme, lui permettant de l'adapter et de la faire évoluer sans dépendre d'un fournisseur unique.
4. **Fiabilité et Confiance** : L'intégration de solutions de surveillance (Prometheus) et de sauvegarde (PBS) de niveau professionnel n'est pas une option. C'est une nécessité pour construire une relation de confiance avec les clients et garantir la pérennité de leur activité hébergée.

## 2.11. Conclusion

Ce chapitre a posé les fondements théoriques et, surtout, a justifié les choix technologiques comme des décisions stratégiques nécessaires à la réussite du projet. En démontrant la synergie et la pertinence de chaque outil, nous avons établi un socle robuste. Les chapitres suivants se concentreront sur l'implémentation pratique de cette architecture, en traduisant cette vision stratégique en une solution fonctionnelle.

# Chapitre 3 : Analyse de besoins et conduite du projet

### 3.1. Introduction

Ce chapitre pour objectif de traduire la problématique générale, définie au chapitre précédent, en un ensemble de spécifications claires et en une architecture technique robuste. Nous commencerons par identifier les différents acteurs qui interagiront avec notre solution, avant de formaliser les exigences fonctionnelles et non fonctionnelles sous la forme d'un Product Backlog. Nous modéliserons ensuite les fonctionnalités clés à l'aide de diagrammes de cas d'utilisation. Enfin, ce chapitre présentera en détail l'architecture technique cible — applicative, réseau et sécurité — qui servira de plan directeur pour toute la phase de développement.

### 3.2. Identification des Acteurs

L'identification précise des acteurs est une étape essentielle pour définir les interactions avec le système et comprendre les besoins de chacun. Les acteurs de notre solution sont répartis en deux catégories : les acteurs humains, qui interagissent directement avec l'interface, et les acteurs systèmes, qui sont les composants logiciels orchestrés.

Le tableau ci-dessous synthétise ces acteurs, leur rôle et leurs objectifs principaux.

Tableau 6 : Description des acteurs

Type	Acteur	Rôle et Description	Objectifs Principaux
Humain	Client Final	Utilisateur externe qui utilise <b>deux interfaces principales</b> : le portail Odoo pour l'aspect commercial et l'interface Proxmox VE pour la gestion technique de ses propres ressources.	<ul style="list-style-type: none"> <li>○ <b>Sur Odoo :</b> Consulter, personnaliser et commander des VMs, recevoir ses notifications et factures.</li> <li>○ <b>Sur Proxmox :</b> Se connecter avec son propre compte, visualiser ses VMs dans son pool dédié, et effectuer des opérations de base (démarrer, arrêter, console).</li> </ul>
	Administrateur	L'utilisateur technique unique, responsable de la gestion et de la configuration de l'ensemble de la plateforme, aussi bien sur <i>Odoo</i> que sur <i>Proxmox VE</i> .	<ul style="list-style-type: none"> <li>○ <b>Sur Odoo :</b> Créer les produits « VM », configurer les paramètres de l'intégration, et superviser les commandes.</li> <li>○ <b>Sur Proxmox :</b> Maintenir la santé de l'hyperviseur, gérer les templates de VM, superviser l'infrastructure globale et configurer/gérer la politique de sauvegarde des VMs.</li> </ul>

	Système Odoo (Orchestrateur)	Le module d'intégration développé, qui agit comme le cerveau de l'automatisation.	<ul style="list-style-type: none"> <li>○ Traduire une commande Odoo en un processus technique.</li> <li>○ Créer automatiquement le compte utilisateur et le pool de ressources du client dans Proxmox.</li> <li>○ Piloter les outils IaC.</li> </ul>
<b>Système</b>	Outils IaC (Terraform & Ansible)	Les exécutables appelés en ligne de commande par Odoo pour agir sur l'infrastructure.	<ul style="list-style-type: none"> <li>○ <b>Terraform</b> : Provisionner les VMs dans le pool du bon client.</li> <li>○ <b>Ansible</b> : Configurer le réseau, le bastion et les « VM » d'utilisateurs.</li> </ul>
	API Proxmox VE	L'interface de programmation de l'hyperviseur qui expose les fonctionnalités de virtualisation	<ul style="list-style-type: none"> <li>○ Exécuter les ordres de création (VMs, utilisateurs, pools, ACLs).</li> <li>○ Retourner l'état des ressources à Odoo.</li> </ul>

### 3.3. Analyse des Exigences

Cette section définit les exigences fonctionnelles et non fonctionnelles qui encadrent le développement de notre solution. Elles décrivent ce que le système doit faire et les qualités qu'il doit posséder pour être considéré comme un succès.

#### 3.3.1. Besoins fonctionnels

Les besoins fonctionnels décrivent les actions et les services que le système doit fournir aux utilisateurs. Ils sont synthétisés dans le tableau ci-dessous, regroupés par grande capacité métier.

Capacité Métier	ID	Description du Besoin Fonctionnel
<b>Expérience Client (Portail Odoo)</b>	BF-01	Permettre à un client de commander et personnaliser une VM (OS, CPU, etc.) via une interface e-commerce.
	BF-02	Fournir un espace client dans le portail Odoo où le client peut visualiser la liste de ses VMs déployées et leurs états.
	BF-03	Gérer le cycle de vie de la commande (devis, facture, paiement) via les fonctionnalités standards d'Odoo.
	BF-04	Envoyer des notifications par mail automatisées au client pour les étapes clés (confirmation de commande, accès, etc.).
<b>Provisionnement (IaC)</b>	BF-05	Déclencher automatiquement le déploiement d'un environnement complet à la suite d'une commande client validée.
	BF-06	Le déploiement doit être basé sur des templates standardisés (« Cloud-Ready ») et permettre la personnalisation post-déploiement.

<b>Gestion Client (Portail Proxmox)</b>	BF-07	Créer automatiquement un compte utilisateur dédié pour chaque client dans Proxmox avec des permissions restreintes.
	BF-08	Le client doit pouvoir se connecter à l'interface Proxmox pour réaliser des opérations de base sur ses VMs (démarrer, etc.).
<b>Administration et Supervision (Odoo)</b>	BF-09	Fournir une interface centralisée dans Odoo pour que l'administrateur puisse configurer les paramètres de l'intégration.
	BF-10	Permettre à l'administrateur de définir et gérer la stratégie d'allocation des réseaux clients (super-réseaux, etc.).
	BF-11	Fournir une vue centralisée listant toutes les VMs déployées avec leurs détails (client, état, IP).
	BF-12	Afficher les informations de virtualisation pertinentes (ID utilisateur, pool, VMs) directement sur la fiche client.
<b>Sécurité et Réseau</b>	BF-13	Allouer automatiquement un réseau virtuel privé et isolé à chaque client.
	BF-14	Sécuriser tout accès SSH externe via une machine bastion (Jump Host) obligatoire.
	BF-15	Générer et attribuer automatiquement des clés SSH pour garantir des connexions sécurisées.
<b>Exploitation et Fiabilité</b>	BF-15	Mettre en place une supervision de la performance de l'infrastructure globale et des VMs clientes.
	BF-16	Mettre en place une stratégie de sauvegarde automatisée des VMs et disposer de procédures de restauration.

Tableau 7 : Besoins Fonctionnels

### 3.3.2. Besoins Non Fonctionnels

Catégorie	ID	Description du Besoin Non Fonctionnel
<b>Sécurité</b>	BNF-01	L'isolation des environnements clients doit être stricte. L'accès des administrateurs à l'infrastructure doit être contrôlé et tracé.
	BNF-02	Toutes les communications doivent être chiffrées
<b>Performance</b>	BNF-03	Le temps de provisionnement d'une VM standard (de la commande à la disponibilité) ne doit pas excéder 10 minutes.
	BNF-04	L'interface web Odoo doit rester réactive même lors d'opérations de déploiement en arrière-plan.
<b>Fiabilité</b>	BNF-05	Le processus de déploiement doit être idempotent et robuste aux erreurs.
	BNF-06	La plateforme doit viser une haute disponibilité, en minimisant les points de défaillance uniques (SPOF).
<b>Scalabilité</b>	BNF-07	L'architecture doit pouvoir supporter une augmentation progressive du nombre de clients et de VMs sans nécessiter une refonte majeure.

Maintenabilité	BNF-08	Le code doit être modulaire et documenté pour faciliter les futures évolutions et la correction des anomalies.
Usabilité	BNF-09	Les interfaces (boutique pour le client, back-office pour l'admin) doivent être intuitives et ne pas requérir de formation approfondie.

Tableau 8 : Besoins non Fonctionnels

### 3.3.3. Product Backlog Initial

Les exigences fonctionnelles ont été décomposées en User Stories pour former notre Product Backlog. Ce dernier représente la liste de toutes les fonctionnalités à développer, priorisées en fonction de leur valeur métier. Le tableau ci-dessous présente les principales User Stories regroupées par grande thématique (Epic).

Epic	ID	User Story
EPIC 1 : Expérience Client	US-01	En tant que Client, je veux parcourir un catalogue et personnaliser ma VM afin de commander un service adapté à mes besoins.
	US-02	En tant que Client, je veux me connecter à un portail Proxmox afin d'effectuer des opérations de base (On/Off, Console) sur mes VMs.
	US-03	En tant que Client, je veux recevoir un mail de bienvenue avec mes identifiants Proxmox afin de pouvoir accéder à mon espace de gestion.
	US-04	En tant que Client, je veux recevoir un mail d'information sur mon réseau privé afin de connaître les détails de ma connectivité.
	US-05	En tant que Client existant, je veux être consulté par mail pour le choix du réseau afin de pouvoir regrouper ou isoler mes services.
	US-06	En tant que Client, je veux recevoir un mail d'accès final à ma VM afin de pouvoir me connecter en SSH via le bastion.
EPIC 2 : Administration Odoo	US-07	En tant qu'Admin, je veux une page de configuration centrale afin de gérer facilement les paramètres de l'intégration.
	US-08	En tant qu'Admin, je veux désigner un produit comme "VM Proxmox" afin de le rendre éligible à l'automatisation.

	US-9	En tant qu'Admin, je veux définir la stratégie d'allocation réseau afin de contrôler l'utilisation des plages d'IP.
	US-10	En tant qu'Admin, je veux une vue listant tous les réseaux clients alloués afin de superviser l'occupation des sous-réseaux.
	US-11	En tant qu'Admin, je veux une vue centralisée de toutes les VMs déployées afin de suivre rapidement l'état du parc.
	US-12	En tant qu'Admin, je veux un résumé des infos Proxmox sur la fiche client afin d'avoir une vue à 360° sans changer d'écran.
	US-13	En tant qu'Admin, je veux pouvoir relancer un déploiement depuis la facture afin de gérer manuellement un cas d'erreur.
	US-14	En tant qu'Admin, je veux configurer les chemins et les paramètres des outils d'orchestration (Terraform, Ansible) afin que le système puisse les piloter correctement.
<b>EPIC 3 : Réseau et Sécurité</b>	US-15	En tant qu'Admin, je veux que chaque client soit dans un réseau privé et isolé afin de garantir la confidentialité et la sécurité.
	US-16	En tant qu'administrateur réseau, je souhaite que les clients reçoivent automatiquement une adresse IP via le service DHCP, afin qu'ils puissent accéder à Internet de manière transparente.
	US-17	En tant qu'Admin, je veux que l'accès SSH externe passe par un bastion sécurisé afin de centraliser et contrôler les connexions.
<b>EPIC 4 : Automatisation</b>	US-18	En tant que Système, je veux créer automatiquement un compte et un pool Proxmox afin d'isoler administrativement les ressources de chaque client.
	US-19	En tant que Système, je veux allouer automatiquement un sous-réseau privé logique afin d'appliquer la stratégie réseau de l'admin.
	US-20	En tant que Système, je veux automatiser la configuration du réseau physique (Bridge, DHCP, NAT) afin d'éliminer les interventions manuelles.
	US-21	En tant que Système, je veux générer les clés SSH et configurer l'utilisateur sur le bastion afin d'automatiser la mise en place de l'accès sécurisé

<b>EPIC 5 : Exploitation</b>	US-22	En tant que Système, je veux déployer la VM via IaC en clonant un Template afin de garantir un provisionnement rapide et standardisé.
	US-23	En tant que Système, je veux installer des services additionnels sur la VM afin de livrer un environnement prêt à l'emploi.
	US- 24	En tant que Système, je veux déployer une Machine Virtuelle à partir d'une image ISO pour une commande client afin de fournir le service commandé
	US-25	En tant qu'Admin, je veux superviser les performances via des Dashboard afin de détecter proactivement les surcharges.
	US-26	En tant qu'Admin, je veux recevoir des alertes en cas de problème critique afin d'être informé immédiatement d'un incident.
	US-27	En tant qu'Admin, je veux configurer des sauvegardes automatiques et pouvoir lancer des sauvegardes manuelles afin de garantir une protection flexible des données.
	US-28	En tant qu'Admin, je veux disposer d'un plan de reprise d'activité (PRA) documenté afin de savoir comment réagir en cas de sinistre.
	US-29	En tant qu'Admin, je veux pouvoir synchroniser manuellement le statut d'une ou plusieurs VMs depuis Odoo afin d'obtenir leur état en temps réel.
	US-30	En tant que Système, je veux synchroniser périodiquement le statut de toutes les VMs gérées afin que les informations dans Odoo restent à jour.

Tableau 9 : Backlog de produit

### 3.3.4. Répartition du Backlog en Sprints

Le Product Backlog, présenté précédemment, a été séquencé en cinq sprints distincts. Chaque sprint a un objectif clair et s'appuie sur les fondations établies par les précédents, créant ainsi une narration logique de la construction du produit, du simple MVP à la solution complète et automatisée. Le tableau 5 ci-dessous détaille cette répartition.

Sprint	Objectif du Sprint	User Stories Clés Réalisées (par ID)
<b>Sprint 1 : Le Flux Client (MVP)</b>	Valider un parcours client de A à Z avec un déploiement simple (sur réseau public) et une supervision admin de base.	US-01, US-02, US-03, US-07, US-08, US-11, US-12, US-13, US-18, US-24, US-29, US-30,
<b>Sprint 2 : Isolation Client et Gestion Réseau</b>	Automatiser la gestion logique du réseau et l'interaction client dans Odoo, tout en conservant la création physique de l'infrastructure sous la responsabilité de l'administrateur.	US-04, US-05, US-9, US-10, US-15, US-16, US-17, US-19
<b>Sprint 3 : Orchestration de l'Infrastructure (IaC)</b>	Automatiser les tâches manuelles identifiées au Sprint 2 et finaliser le déploiement à l'aide de templates, afin de fournir un service entièrement fonctionnel.	US-06, US-14, US-20, US-21, US-22, US-23
<b>Sprint 4 : Monitoring et Fiabilisation</b>	Mettre en place une supervision active de l'infrastructure avec un système d'alerte pour garantir la fiabilité du service.	US-25, US-26
<b>Sprint 5 : Sauvegarde et Reprise d'Activité</b>	Définir et implémenter une stratégie complète de sauvegarde et de restauration des données pour assurer la continuité d'activité.	US-27, US-28

Tableau 10 : Tableau de Planification

### 3.4. Vision d'ensemble des Fonctionnalités

#### 3.4.1. Diagramme des Cas d'Utilisation Global

Pour illustrer les fonctionnalités clés de notre solution et les interactions entre les différents acteurs, nous avons élaboré le diagramme de cas d'utilisation global ci-dessous (Figure 2). Ce diagramme offre une vue d'ensemble du périmètre fonctionnel du système, en se concentrant sur les objectifs que chaque acteur cherche à atteindre. Il constitue une représentation visuelle des besoins fonctionnels identifiés précédemment.

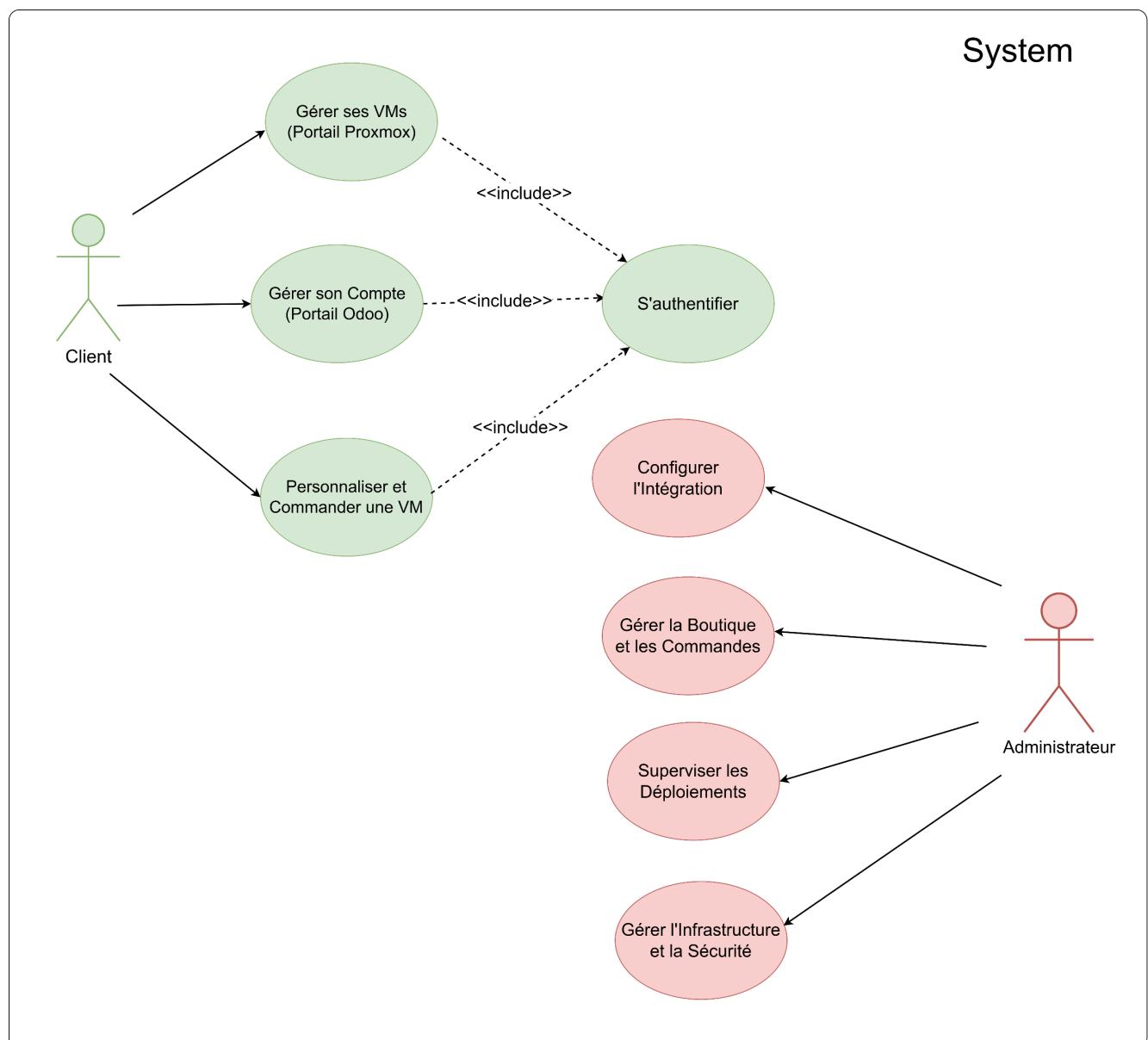


Figure 2 : Diagramme des Cas d'utilisation Global

### 3.4.2. Diagramme d'activité global

Pour illustrer le cycle de vie complet d'un service, de sa commande jusqu'à sa maintenance, le diagramme d'activité global (Figure 3) a été élaboré. Il représente le flux de travail métier idéal, en mettant en évidence les responsabilités de chaque acteur et l'enchaînement logique des actions. Pour en faciliter la lecture, nous décrirons ce flux en deux grandes phases : la phase de commande et de déploiement, puis la phase d'exploitation

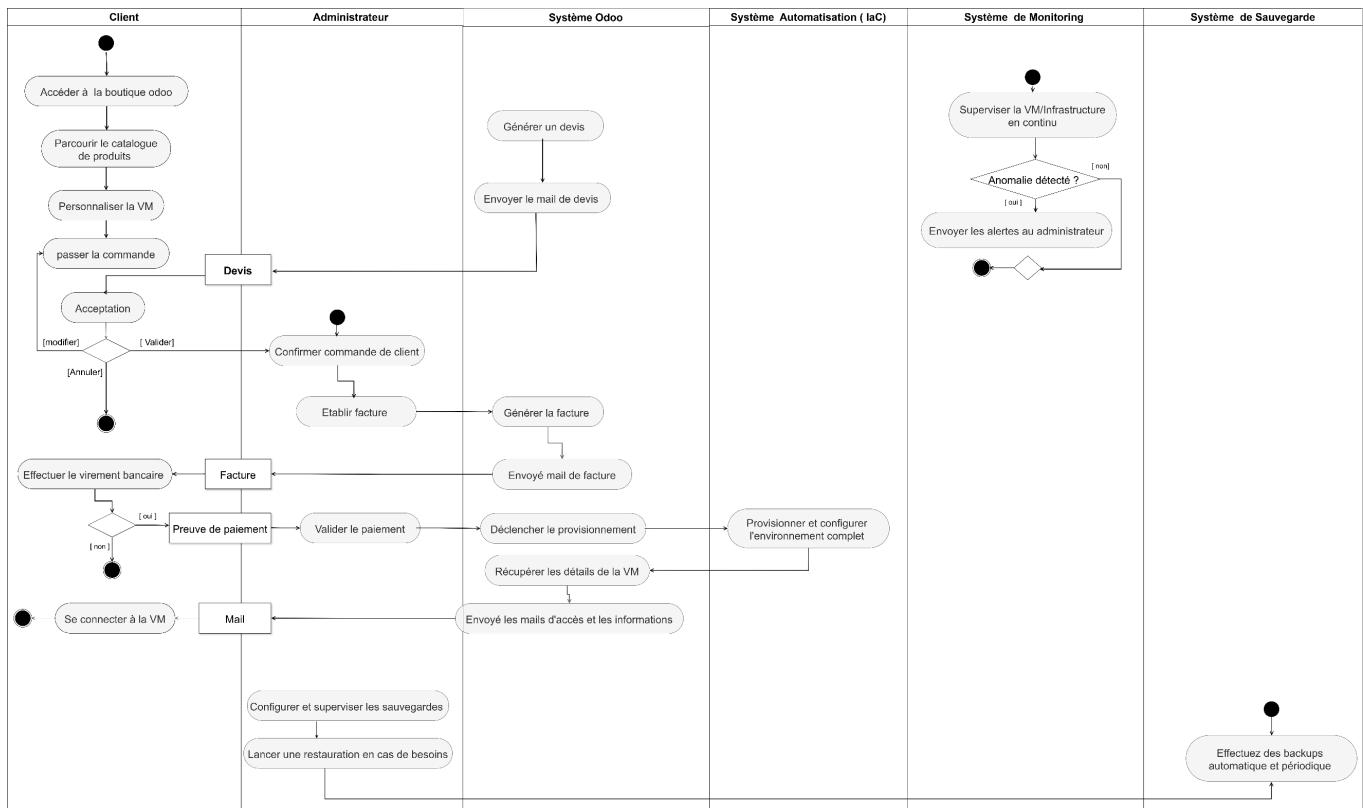


Figure 3 : Diagramme d'activités global Phase 1 & Phase 2

#### Phase 1 : De la Commande à la Livraison

Le processus débute avec le **Client** qui personnalise et commande sa VM sur le portail Odoo. Le **Système Odoo** génère alors automatiquement les documents commerciaux (devis, puis facture) après validation par l'**Administrateur**. L'étape clé de ce flux est la validation manuelle du paiement par l'administrateur, qui agit comme le déclencheur du processus technique. Une fois cette validation effectuée, le **Système d'Automatisation (IaC)** prend le relais pour provisionner et configurer l'ensemble de l'environnement sans aucune autre intervention humaine, jusqu'à l'envoi des informations d'accès finales au client.

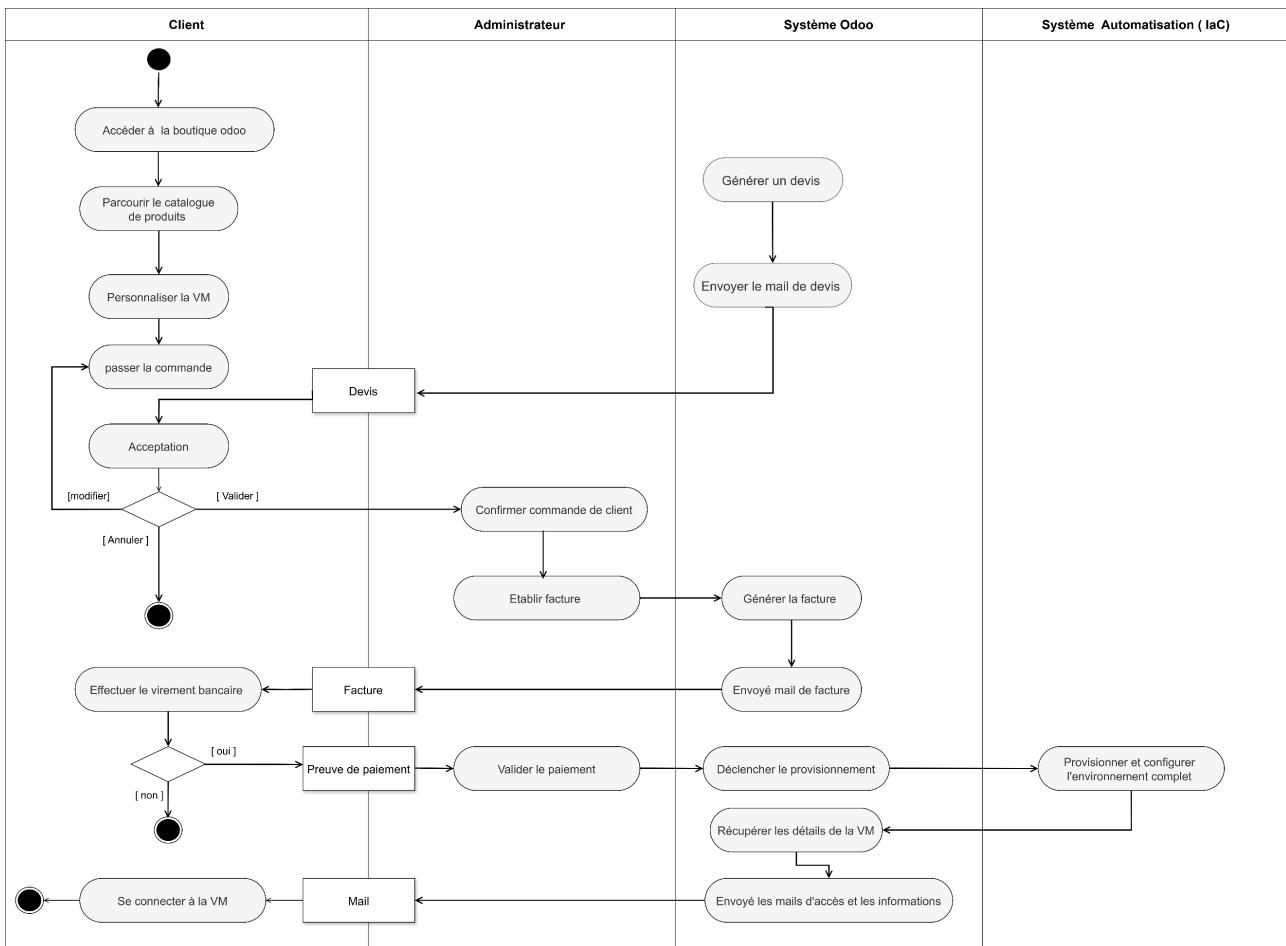


Figure 4 : Diagramme d'activités Phase 1

## Phase 2 : Exploitation et Maintenance

Une fois la VM livrée, le service entre dans sa phase d'exploitation. L'infrastructure est continuellement surveillée par le **Système de Monitoring**. En cas d'anomalie, une alerte est automatiquement envoyée à l'**Administrateur** pour qu'il puisse intervenir. En parallèle, le **Système de Sauvegarde** effectue des backups périodiques des données du client. En cas d'incident majeur, l'administrateur peut s'appuyer sur ces sauvegardes et sur le plan de reprise d'activité pour restaurer le service, bouclant ainsi le cycle de vie complet de la gestion du service.

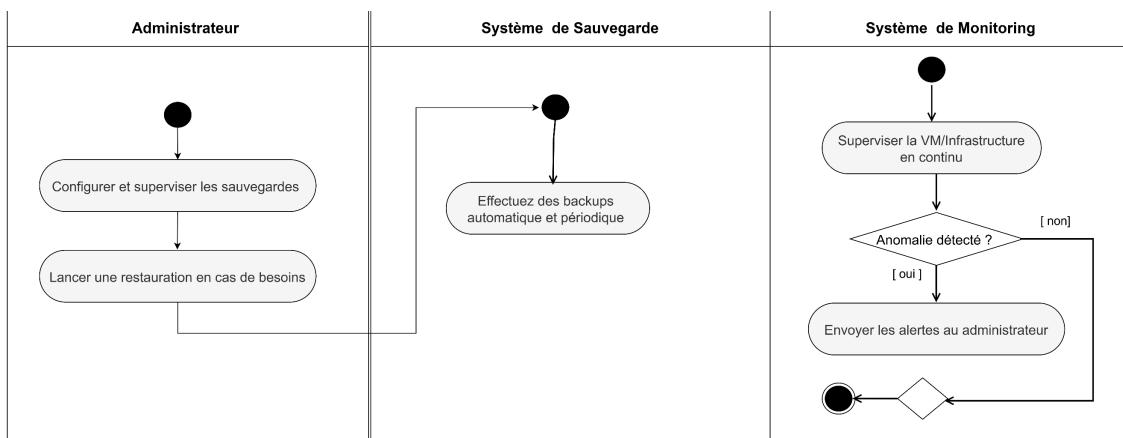


Figure 5 : Diagramme d'activités Phase 2

### 3.5. Conception de l'Architecture Technique Cible

#### 3.5.1. Architecture physique

L'architecture physique de notre solution a été conçue pour garantir la performance, la sécurité et la séparation des fonctions critiques. Elle repose sur deux serveurs physiques distincts et une topologie réseau qui différencie clairement l'accès public des clients des accès de gestion réservés à l'administrateur.

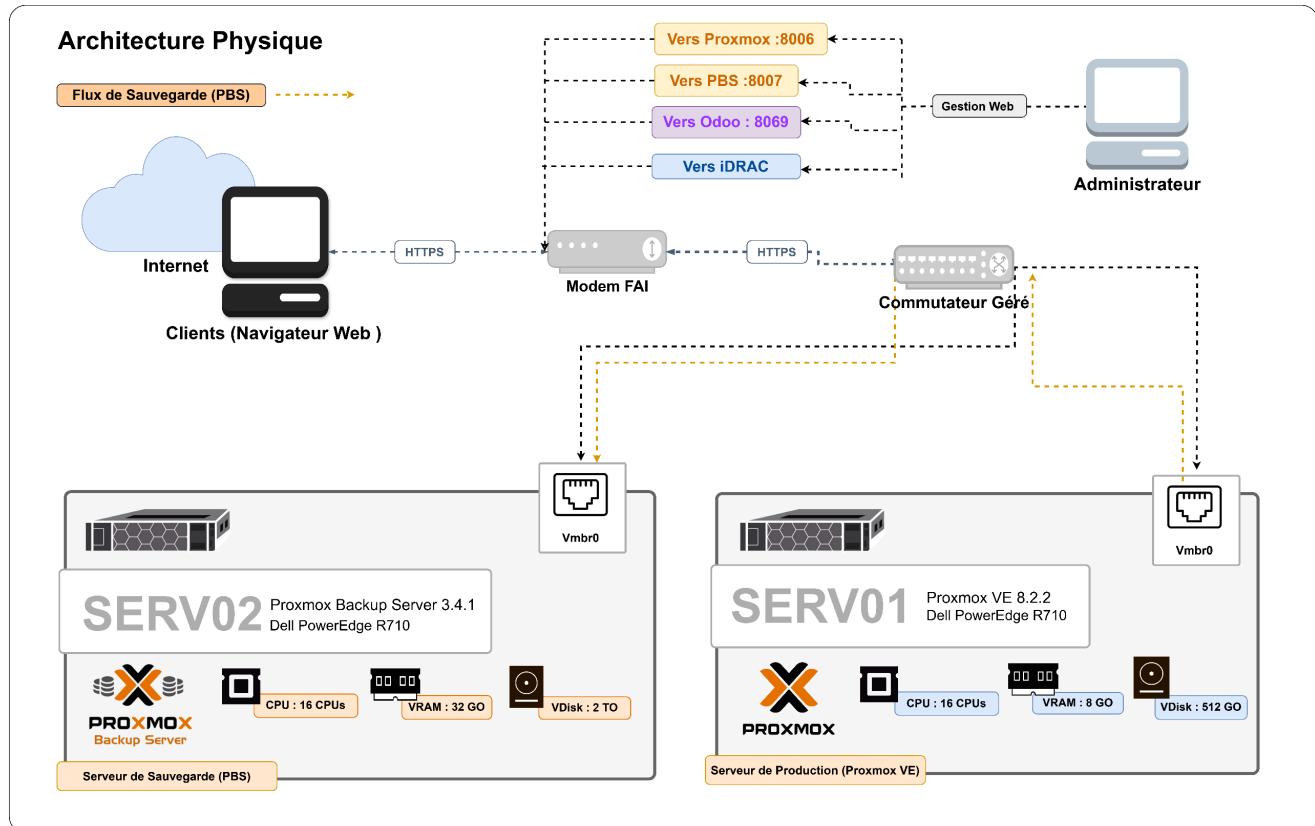


Figure 6 : Architecture physique de la solution

#### 1. Infrastructure Matérielle

Notre infrastructure matérielle se compose de deux serveurs Dell PowerEdge R710, chacun jouant un rôle bien défini :

- **Serveur de Production (SERV01) :**
  - **Spécifications :** 16 vCPUs, 8 Go de RAM, 512 Go de stockage.
  - **Rôle :** Ce serveur héberge l'hyperviseur Proxmox VE 8.2.2. Il est le cœur de la production, responsable de l'exécution de toutes les services (Odoo, NGINX, Monitoring) ainsi que des VMs des clients.

- **Serveur de Sauvegarde (SERV02) :**
  - **Spécifications** : 16 vCPUs, 32 Go de RAM, 2 To de stockage.
  - **Rôle** : Ce serveur est exclusivement dédié à la protection des données via Proxmox Backup Server (PBS) 3.4.1. Ses ressources, notamment 2 To de stockage, sont dimensionnées pour accueillir les sauvegardes de l'ensemble du parc de VMs.

## 2. Topologie du Réseau et des Flux de Données

Le Commutateur Géré est le cœur de notre réseau local. Il orchestre tous les flux de communication :

- **Flux d'Accès Client (Public)** : Les connexions des Clients depuis Internet arrivent via le Modem FAI. Le trafic HTTPS est dirigé par le commutateur vers l'interface réseau vmbr0 du serveur de production (SERV01).
- **Flux de Sauvegarde (Interne)** : Le flux de Sauvegarde (PBS) est purement local. Le serveur de production (SERV01) initie la communication avec le serveur de sauvegarde (SERV02) à travers le commutateur géré pour transférer les données.
- **Flux de Gestion (Accès Administrateur)** :

L'**Administrateur** dispose de plusieurs points d'accès sécurisés pour la gestion de l'infrastructure, distincts du flux client :

- **Accès aux Interfaces d'Infrastructure** : L'administrateur se connecte directement aux interfaces web de Proxmox VE (sur SERV01) et de Proxmox Backup Server (sur SERV02) pour gérer les hyperviseurs et les sauvegardes.
- **Accès aux Services Applicatifs** : Il a également accès aux interfaces de gestion des services hébergés sur SERV01, comme le back-office d'Odoo ou les tableaux de bord Grafana.
- **Accès Hors-Bande (iDRAC)** : Chaque serveur Dell est accessible via son interface iDRAC, permettant une maintenance de bas niveau (redémarrage matériel, etc.) indépendamment du système d'exploitation.

### 3.5.2. Architecture Logique

L'architecture logique de notre solution est structurée en plusieurs niveaux fonctionnels distincts, garantissant la modularité, la sécurité et la clarté des flux de données. Cette organisation en couches, présentée dans la Figure 7, permet à chaque composant de jouer un rôle bien défini, tout en assurant des interactions claires.

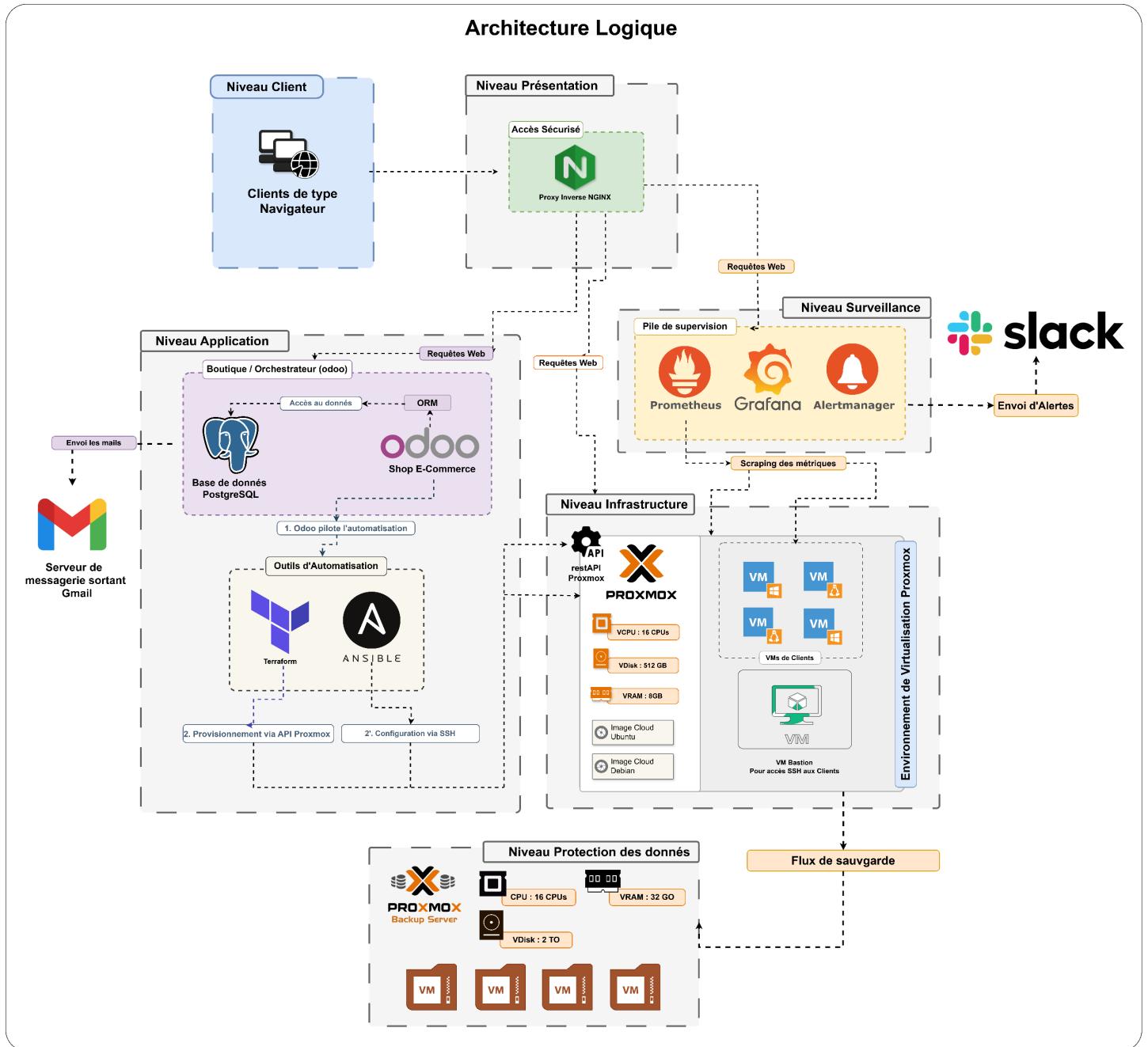


Figure 7 : Architecture Logique

## 1. Niveau Client

Ce niveau représente l'acteur externe, l'utilisateur final qui interagit avec le système via un navigateur web standard depuis Internet. C'est le point de départ de tout le processus commercial.

## 2. Niveau Présentation

C'est la façade publique et sécurisée de notre infrastructure. Ce niveau agit comme un point de contrôle unique pour tout le trafic entrant.

- **Proxy Inverse (NGINX)** : Toutes les requêtes des clients sont interceptées par NGINX. Ce composant est crucial pour la sécurité : il gère le chiffrement des communications (HTTPS), masque la topologie interne de notre réseau, et agit comme un répartiteur de charge en redirigeant les requêtes vers les services applicatifs appropriés.

## 3. Niveau Application : Le Cerveau de l'Orchestration

Ce niveau est le cœur de notre solution, où la logique métier, la gestion des données et l'automatisation sont exécutées.

- **Boutique / Orchestrateur (Odoo)** : Ce composant central, basé sur l'ERP Odoo, joue un double rôle. Son interface "Shop E-Commerce" constitue la vitrine visible par le client pour la commande de services. En arrière-plan, notre module personnalisé agit comme l'orchestrateur, traduisant les commandes en actions techniques. Il s'appuie sur son ORM (Object-Relational Mapping) pour interagir de manière structurée avec sa base de données PostgreSQL.
- **Outils d'Automatisation (Terraform & Ansible)** : Directement pilotés par Odoo via des appels système, ces outils sont les agents d'exécution de l'orchestration. Terraform est invoqué pour le provisionnement déclaratif de l'infrastructure via l'API Proxmox, tandis qu'Ansible est utilisé pour la configuration fine des systèmes via des connexions SSH sécurisées.
- **Serveur de Messagerie (Gmail)** : Intégré à ce niveau, un service SMTP externe comme Gmail est utilisé par Odoo pour gérer toutes les communications sortantes par mail, telles que les confirmations de commande, les notifications de réseau et l'envoi des informations d'accès.

#### 4. Niveau Infrastructure (Proxmox VE)

Ce niveau représente la plateforme de virtualisation où les ressources sont concrètement créées et gérées sur commande du niveau applicatif.

- **API Proxmox** : C'est l'interface de programmation RESTful exposée par Proxmox. Elle est la cible des appels de Terraform et permet de manipuler par programme toutes les ressources de l'hyperviseur.
- **Environnement de Virtualisation** : Il s'agit du parc de machines virtuelles géré par Proxmox. On y distingue les VMs des Clients, qui sont isolées les unes des autres, et la VM Bastion, une machine renforcée qui sert de point d'accès SSH unique et sécurisé pour les clients.

#### 5. Niveau Surveillance

Ce niveau, essentiel à l'exploitation, est dédié à l'observation et à la fiabilisation de la plateforme en temps réel.

- **Pile de Supervision (Prometheus, Grafana, Alertmanager)** : Cet ensemble d'outils open-source est déployé pour garantir la visibilité sur l'état de santé de l'infrastructure. Prometheus collecte activement les métriques de performance des VMs et des hôtes via un processus de scraping. Grafana interroge cette base de données de séries temporelles pour offrir des tableaux de bord visuels et intuitifs à l'administrateur. Enfin, Alertmanager traite les alertes définies dans Prometheus et les transmet au canal de communication de l'équipe technique.
- **Service de Notification d'Alertes (Slack)** : Intégré à Alertmanager, Slack est le récepteur final des alertes critiques, permettant une réaction rapide de l'administrateur en cas d'incident.

#### 6. Niveau Protection des Données

Logiquement et physiquement séparé de la production, ce niveau assure la pérennité et la sécurité des données clientes.

- **Proxmox Backup Server (PBS)** : Ce service spécialisé reçoit le flux de sauvegarde de l'environnement de production. Il stocke les données des VMs de manière incrémentale et dédupliquée, permettant des restaurations rapides et efficaces en cas de besoin.

### 3.5.3. Conception du Réseau Client et de la Sécurité

Pour garantir une isolation stricte entre les clients, une sécurité robuste et une gestion automatisable, nous avons conçu une architecture réseau multi-niveaux. Cette conception repose sur une segmentation par super-réseaux, un plan d'adressage IP structuré, et un modèle de sécurité basé sur un bastion d'accès.

#### 1. Segmentation par Super-Réseaux

L'espace d'adressage IP a été divisé en quatre super-réseaux principaux, chacun jouant un rôle bien défini :

- **Super-réseau de Management (192.168.1.0/24) :**
  - **Rôle :** Ce réseau est réservé à l'infrastructure de gestion. Il héberge les services critiques comme les serveurs Odoo, Proxmox VE, Proxmox Backup Server, et la pile de monitoring. Il n'est pas accessible depuis l'extérieur, sauf pour les accès de gestion de l'administrateur.
- **Super-réseau Clients « Standard » (10.10.0.0/16) :**
  - **Rôle :** C'est le pool d'adresses principal utilisé pour allouer des réseaux privés aux clients de type « particulier ».
- **Super-réseau Clients « Réserve » (10.11.0.0/16) :**
  - **Rôle :** Ce pool sert de réserve. Il sera utilisé pour les clients « particulier » uniquement si le super-réseau principal (10.10.0.0/16) est entièrement alloué, garantissant ainsi la continuité du service.
- **Super-réseau Clients « Entreprise » (172.16.0.0/16) :**
  - **Rôle :** Ce réseau est exclusivement dédié aux clients de type « entreprise », permettant une segmentation logique et potentiellement des politiques de sécurité ou de qualité de service (QoS) différenciées à l'avenir.

## 2. Plan d'Adressage et Allocation par Client (Sous-réseautage)

Pour isoler chaque client, nous adoptons une stratégie de « **un sous-réseau par client** ».

- **Principe du Sous-réseautage :** Chaque super-réseau (de taille /16) est découpé en 256 sous-réseaux de taille /24.
  - **Calcul :** Un masque /16 laisse 16 bits pour les hôtes. En utilisant un masque /24, nous empruntons 8 bits à la partie hôte ( $2^8 = 256$  sous-réseaux possibles).
- **Capacité :** Chaque sous-réseau /24 peut accueillir jusqu'à 254 machines, ce qui est largement suffisant pour les besoins d'un client unique.
- **Allocation Automatisée :** Le système Odoo est programmé pour scanner les super-réseaux dans l'ordre de priorité (Standard -> Réserve) et d'allouer le premier sous-réseau /24 disponible à chaque nouveau client.

Tableau 11 : Exemple d'Allocation de Sous-réseaux

Type de Client	Super-réseau Source	Sous-réseau Alloué au Client 1	Sous-réseau Alloué au Client 2
Particulier	10.10.0.0/16	10.10.1.0/24	10.10.2.0/24
Entreprise	172.16.0.0/16	172.16.1.0/24	172.16.2.0/24

## 3. Architecture de Sécurité d'Accès

L'accès client aux machines virtuelles est conçu selon un modèle "Zero Trust", où aucun accès direct depuis Internet n'est autorisé. Tout le trafic SSH doit transiter par une machine bastion.

- **La VM Bastion :**
  - C'est un serveur renforcé, unique point d'entrée SSH pour tous les clients.
  - Elle est déployée sur le réseau public (vmbr0).
- **Le Flux de Connexion Client :**
  1. Le client, depuis son poste, se connecte en SSH à la VM Bastion en utilisant un utilisateur dédié (ex : firas\_jumper) et sa clé privée.
  2. Une fois connecté au bastion, il peut "sauter" (SSH ProxyJump) vers sa VM finale, qui se trouve sur son réseau privé (ex : 10.10.1.5).

- Utilisateurs Techniques pour l'Automatisation :

Pour que ce modèle fonctionne de manière automatisée et sécurisée, plusieurs utilisateurs techniques sont nécessaires :

- **ansible\_admin** (sur les hôtes Proxmox) : Un utilisateur avec des permissions restreintes sur Proxmox, utilisé par Ansible pour créer et configurer les éléments d'infrastructure (bridges, NAT, DHCP). Sa clé SSH publique est déployée sur les hôtes Proxmox.
- **bastion\_setup\_user** (sur la VM Bastion) : Un utilisateur avec des droits sudo, utilisé par Ansible pour effectuer des tâches d'administration sur le bastion, notamment la création des comptes utilisateurs pour les clients finaux.

### 3.5.4. Conception Initiale du diagramme de classes

La conception de notre solution s'appuie sur le mécanisme d'héritage d'Odoo pour étendre le modèle de données natif. Avant de créer nos propres modèles, nous avons identifié les classes fondamentales qui serviront de base à notre intégration. La Figure 8 présente ce diagramme de classes initial, montrant les points d'héritage de notre module.

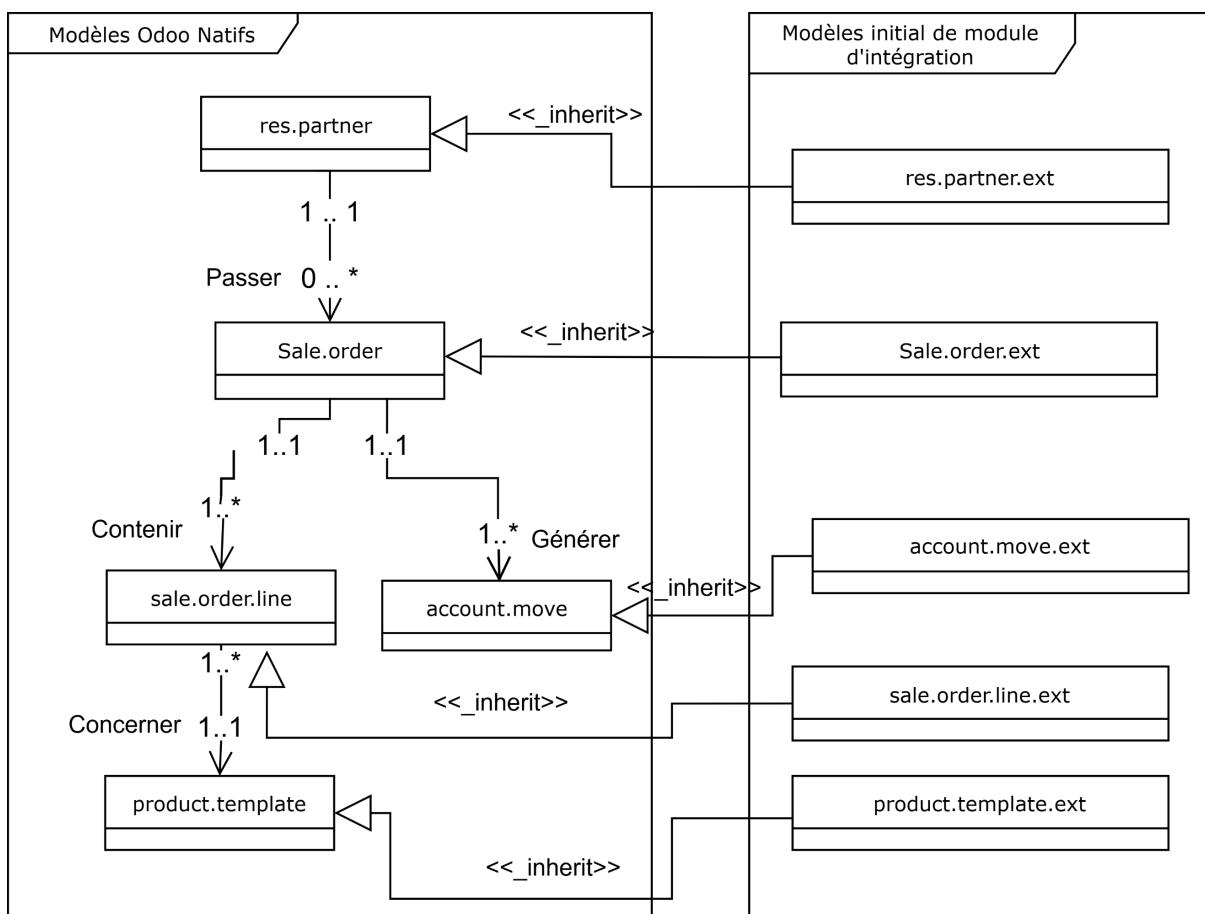


Figure 8 : Diagramme de classes

Ce diagramme illustre que notre module d'intégration hérite des classes standards d'Odoo pour y ajouter des champs et des logiques spécifiques à Proxmox. L'évolution de ce modèle, sera détaillée de manière incrémentale dans les chapitres dédiés à chaque sprint.

### 3.6. Planification des sprints

La figure 9 ci-dessous offre une représentation visuelle de haut niveau de notre feuille de route. Elle illustre l'enchaînement chronologique des sprints thématiques sur la durée du projet.

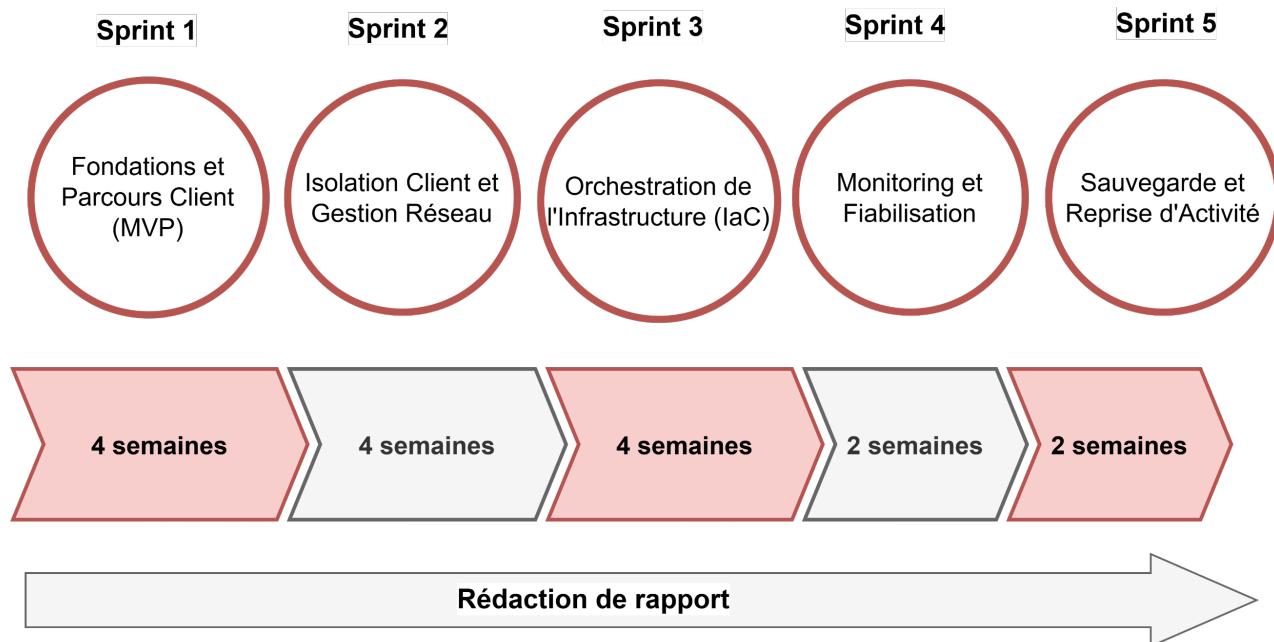


Figure 9 : Planification des Sprints

### 3.7. Conclusion

Ce chapitre a posé les fondements analytiques et conceptuels sur lesquels repose ce projet. Nous avons défini le périmètre fonctionnel à travers les exigences et le Product Backlog, modélisé les interactions et conçu une architecture technique cible répondant à la problématique.

Enfin, nous avons établi une feuille de route claire pour la réalisation du projet. Le chapitre suivant décrira la préparation concrète de notre environnement de travail, préambule à la phase de développement itératif qui constituera le noyau structurant de ce rapport.

# Chapitre 4 : Préparation de l'Environnement et des Outils

## 4.1. Introduction

Ce chapitre décrit la phase fondamentale de préparation de l'infrastructure et des outils, équivalente à un « Sprint 0 ». L'objectif est de construire un socle technique robuste et sécurisé sur lequel l'ensemble de notre solution d'orchestration sera déployé. Nous détaillerons le processus de configuration des serveurs physiques, l'installation des plateformes de virtualisation et de gestion, le déploiement de la pile de monitoring, et la préparation des artefacts essentiels à notre pipeline d'automatisation.

## 4.2. Mise en Place de l'Infrastructure d'Hébergement

### 4.2.1. Configuration de l'Infrastructure Physique

La base de notre solution repose sur deux serveurs physiques Dell PowerEdge R710. La première étape a consisté à les configurer pour une gestion à distance sécurisée.

#### A. Préparation Matérielle et Configuration Initiale d'iDRAC

Pour chaque serveur, une vérification matérielle initiale a été effectuée pour s'assurer de la présence des composants requis. Les serveurs ont ensuite été connectés physiquement (alimentation, réseau, périphériques d'entrée) pour permettre leur premier démarrage.

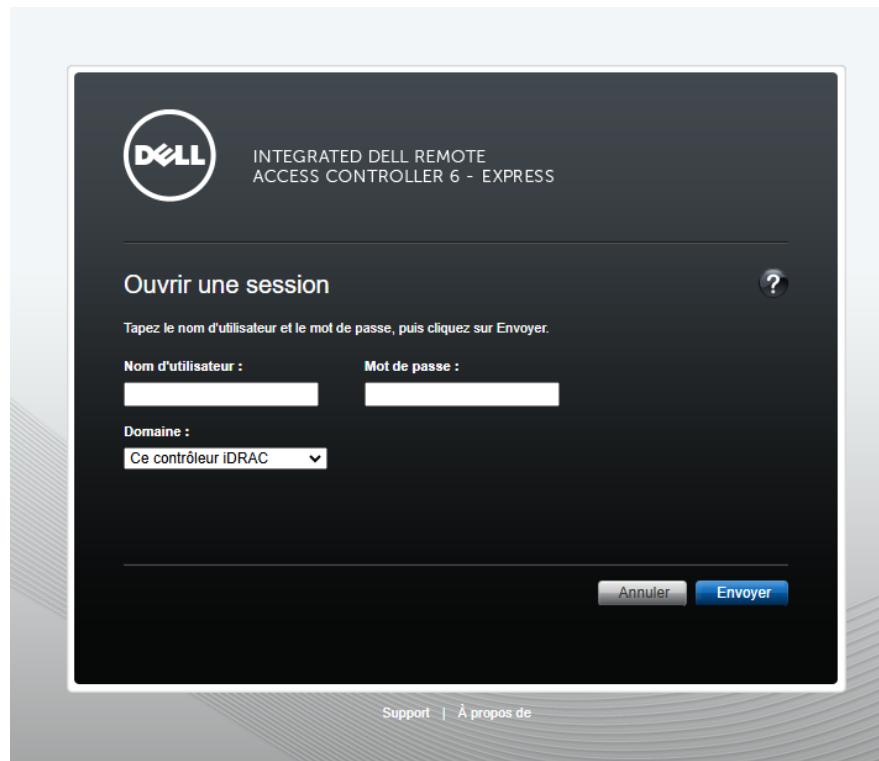
L'accès à l'interface de gestion à distance iDRAC (Integrated Dell Remote Access Controller) a été configuré au démarrage du BIOS (Ctrl + E). Cette étape est cruciale car elle permet une gestion hors-bande (out-of-band) complète du serveur. Une configuration réseau statique a été assignnée à chaque interface iDRAC, incluant une adresse IP dédiée, un masque de sous-réseau et une passerelle, afin de les rendre accessibles sur notre réseau de management.

#### B. Activation et Validation des Accès de Gestion

Depuis l'interface web d'iDRAC, nous avons activé les services d'accès à distance essentiels :

- **Accès Web (HTTPS)** : Pour permettre la gestion via l'interface graphique d'iDRAC, qui offre un contrôle complet sur le matériel, la consommation électrique et l'accès à la console virtuelle.
- **Accès SSH** : Pour permettre des connexions sécurisées en ligne de commande directement à l'interface iDRAC, utile pour l'automatisation de certaines tâches de maintenance.

La validation finale de cette étape a consisté à se connecter avec succès à l'interface web d'iDRAC via un navigateur et à établir une session SSH, confirmant ainsi que les serveurs étaient prêts pour l'installation des systèmes d'exploitation.



**Système**  
PowerEdge R710  
root, Admin

**Résumé du système**

Condition	Composant
<input checked="" type="checkbox"/>	Batteries
<input checked="" type="checkbox"/>	Ventilateurs
<input checked="" type="checkbox"/>	Intrusion
<input checked="" type="checkbox"/>	Blocs d'alimentation
<input checked="" type="checkbox"/>	Températures
<input checked="" type="checkbox"/>	Tensions

**Informations sur le serveur**

État de l'alimentation	SOUS TENSION
Modèle du système	PowerEdge R710
Révision du système	II
Nom d'hôte du système	W
Système d'exploitation	
Version du système d'exploitation	
Numéro de service	C6TL95J
Code de service express	26533885591
Version du BIOS	6.6.0
Version du micrologiciel	2.92 (Build 05)
Adresse(s) IP	192.168.1.100

**Aperçu de la console virtuelle**

iDRAC Express ne prend pas

**Tâches de lancement rapide**

- Mettre SOUS/HORS TENSION
- Exécuter un cycle d'alimentation sur le système (redémarrage à froid)
- Lancer la console virtuelle
- Consulter le journal des événements système
- Consulter le journal iDRAC
- Mettre à jour le micrologiciel
- Réinitialiser iDRAC

Figure 10 : Accès à l'interface réussi

**SSH**

[▲ Retour au début](#)

Attribut	Valeur
Activé	<input checked="" type="checkbox"/>
Nombre maximal de sessions	2
Sessions actives	0
Délai d'attente	1800 secondes
Numéro de port	22

Figure 11 : Accès SSH Activé

## 4.2.2. Déploiement des Plateformes de Virtualisation et de Sauvegarde

Une fois les serveurs physiques préparés et accessibles à distance, l'étape suivante a consisté à déployer les plateformes logicielles qui constituent le socle de notre infrastructure : l'hyperviseur Proxmox VE sur le serveur de production et le Proxmox Backup Server sur le serveur dédié.

### A. Installation de Proxmox VE sur le Serveur de Production (SERV01)

L'hyperviseur Proxmox VE 8.2.2 a été installé sur le serveur SERV01. Le processus a été réalisé à partir de l'image ISO officielle [1], en suivant un assistant d'installation graphique. Les étapes de configuration clés durant ce processus ont été :

- **Sélection du Disque Cible** : le serveur dispose d'un disque de 512 Go.
- **Paramètres Régionaux** : La localisation, le fuseau horaire et la disposition du clavier ont été configurés.
- **Création des Identifiants** : Un mot de passe administrateur robuste a été défini pour l'utilisateur root, et une adresse email a été renseignée pour la réception des notifications système critiques.
- **Configuration du Réseau de Gestion** : Une configuration réseau statique a été mise en place pour l'interface de gestion principale, incluant une adresse IP, un masque de sous-réseau, une passerelle et un serveur DNS

Après un résumé final de la configuration, l'installation a été lancée. Une fois terminée, le serveur a redémarré, et l'accès à l'interface de gestion web via l'URL <https://192.168.162.129:8006> a été validé.

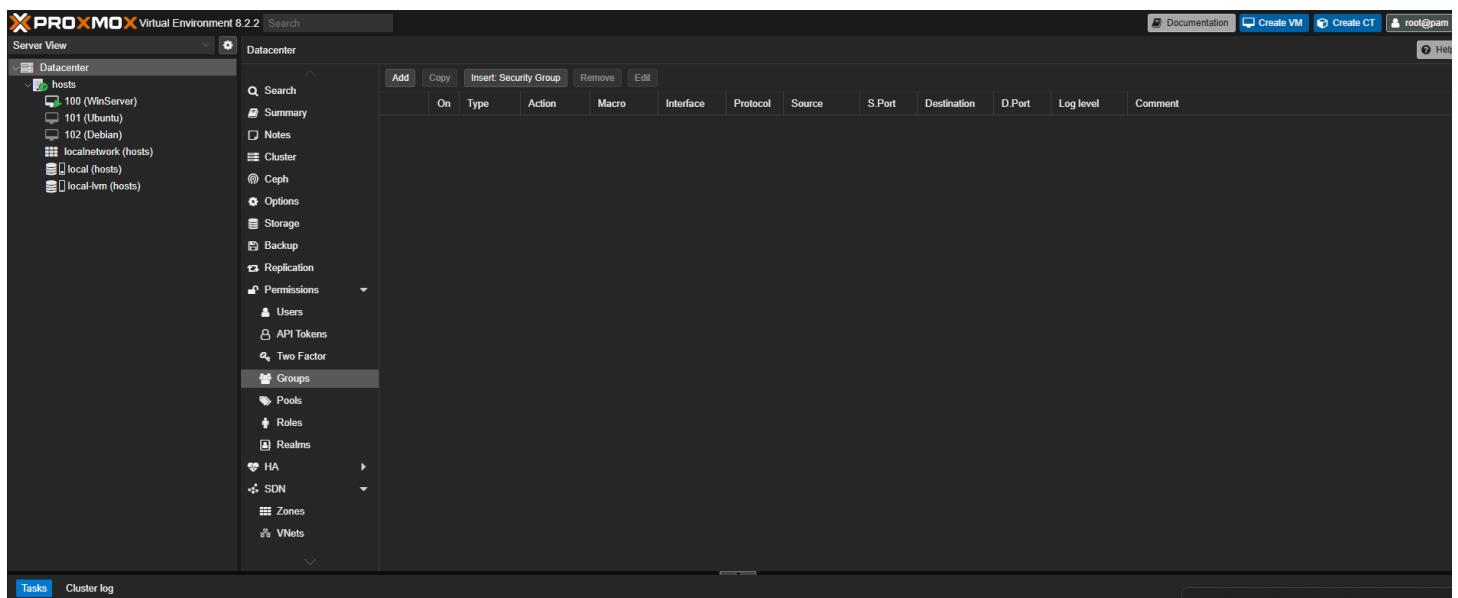


Figure 12 : Écran de connexion de l'interface web de Proxmox VE

## B. Déploiement du Proxmox Backup Server (SERV02)

Le processus de déploiement pour le serveur de sauvegarde est similaire à celui de Proxmox VE.

- Installation et Configuration Initiale :** Proxmox Backup Server 3.4.1 a été installé sur le serveur SERV02. Les mêmes étapes de configuration (disque cible, localisation, mot de passe, réseau) ont été suivies.

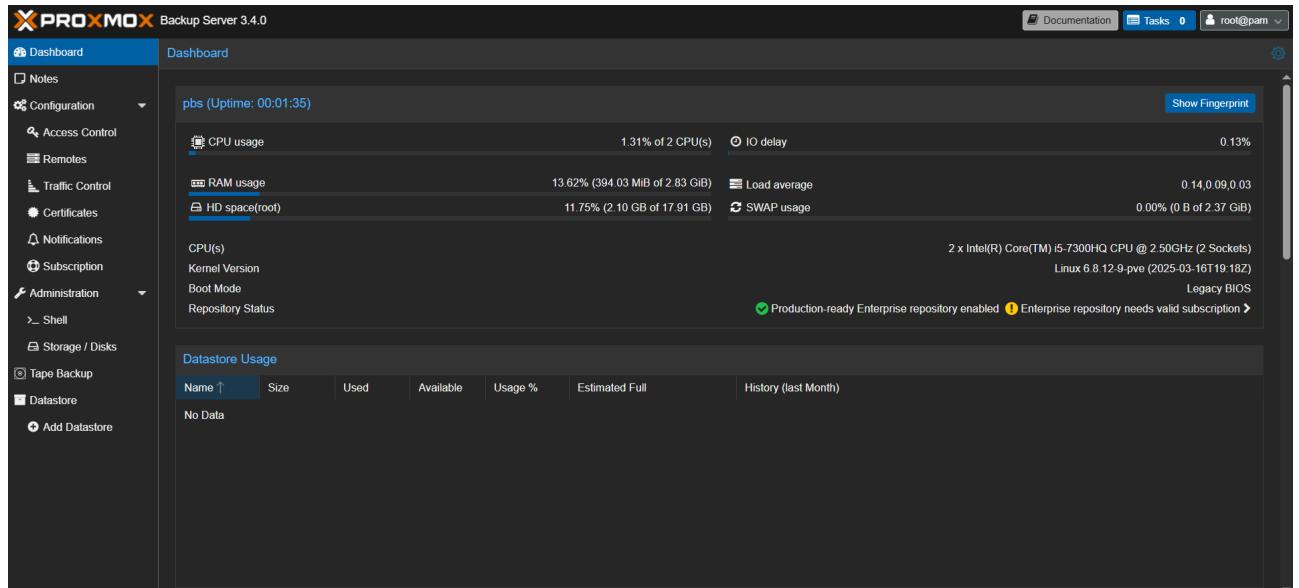


Figure 13 : Écran de connexion de l'interface web de Proxmox Backup Server

- Création du Datastore de Sauvegarde :** Après l'installation, la première action a été de configurer l'espace de stockage. Un disque de 2 To a été initialisé et formaté (par exemple, en ext4) pour créer un répertoire de stockage. Ce répertoire a ensuite été déclaré dans PBS en tant que datastore, qui est l'emplacement logique où les sauvegardes seront stockées.

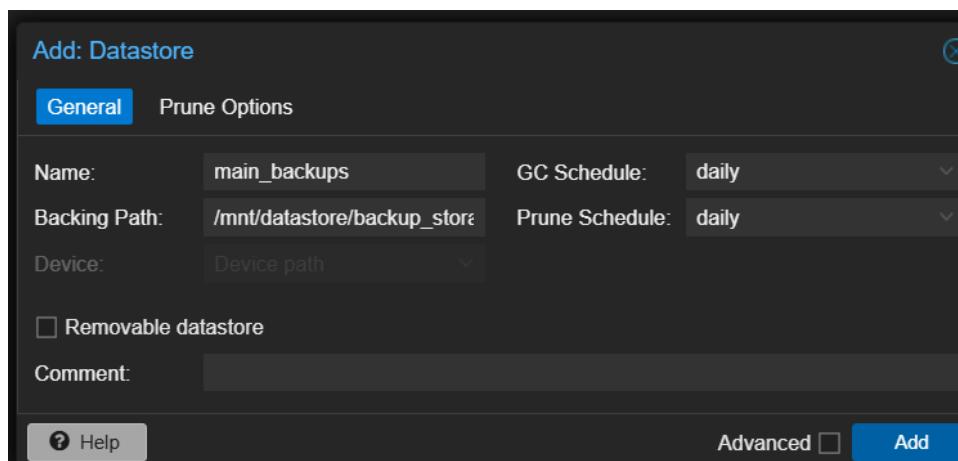


Figure 14 : Ajout d'un datastore

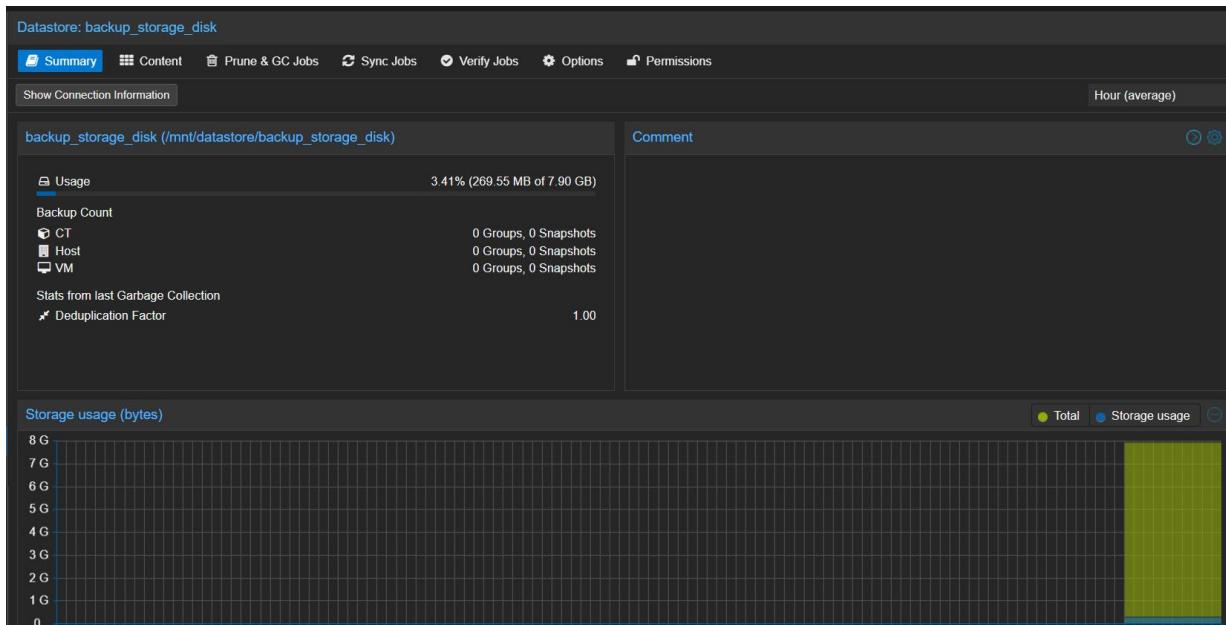


Figure 15 : Résumé du Datastore créé dans l'interface de Proxmox Backup Server

- Liaison avec l'Environnement de Production :** Pour finaliser la mise en place, la connexion entre les deux serveurs a été établie. Dans l'interface de Proxmox VE (SERV01), nous avons ajouté un nouveau « Stockage » de type « Proxmox Backup Server », en renseignant l'adresse IP de SERV02, les identifiants de connexion, et le nom du datastore créé précédemment.

ID:	pbs-test	Nodes:	pve
Server:	192.168.11.134	Enable:	<input checked="" type="checkbox"/>
Username:	root@pve	Content:	backup
Password:	*****	Datastore:	backup_storage_disk
Fingerprint:	62:10:96:ef:92:2f:f1:eb:9c:7c:3f:3d:b5:e4:2e:63:ed:96:89:1b:96:fa:f4:67:b7:30:c2		
<a href="#">Help</a>		<a href="#">Add</a>	

Figure 16 : Ajout de "Proxmox Backup Server"

ID ↑	Type	Content	Path/Target	Shared	Enabled	Bandwidth Limit
local	Directory	VZDump backup file, ISO image, Container ...	/var/lib/vz	No	Yes	
local-lvm	LVM-Thin	Disk image, Container		No	Yes	
pbs	Proxmox B...	VZDump backup file		Yes	Yes	
pbs-test	Proxmox B...	VZDump backup file		Yes	Yes	

Figure 17 : Stockage PBS visible dans la liste des stockages de Proxmox VE

## 4.3. Mise en Place de la Plateforme de Gestion Odoo

Une fois l'infrastructure de virtualisation prête, l'étape suivante a consisté à déployer la plateforme Odoo 18.0. Ce composant est le cœur de notre solution, agissant à la fois comme boutique e-commerce pour les clients et comme chef d'orchestre pour l'ensemble du processus d'automatisation.

### 4.3.1. Installation et Configuration de Base d'Odoo

Une machine virtuelle sous Ubuntu 24.02 a été dédiée à Odoo. Les dépendances requises, notamment Python 3 et PostgreSQL, ont été installées via le gestionnaire de paquets système et le serveur Odoo a ensuite été installé.

La configuration principale du serveur a été définie dans le fichier `odoo.conf`. Cet emplacement centralise les paramètres critiques, notamment la connexion à la base de données, le port d'écoute, et les chemins vers nos modules personnalisés. L'extrait ci-dessous met en évidence les directives les plus importantes pour notre déploiement.

```
[options]
admin_passwd = odoo123
db_host = False
db_port = False
db_user = odoo
db_password = False
logfile = /var/log/odoo18/odoo-server.log
addons_path = /opt/odoo/odoo/addons,/opt/odoo/custom-addons, /opt/
    odoo/custom-addons/queue
proxy_mode = True
xmlrpc_port = 8069
workers = 6
max_cron_threads = 3
server_wide_modules = web,queue_job
limit_time_cpu = 600
limit_time_real = 1200
```

Figure 18 : Configuration clé du fichier `odoo.conf`

Une fois le service Odoo démarré avec cette configuration, la base de données a été créée et les modules applicatifs natifs nécessaires au projet ont été installés : **Site Web**, **Vente**, **Facturation**, et **e-commerce**.

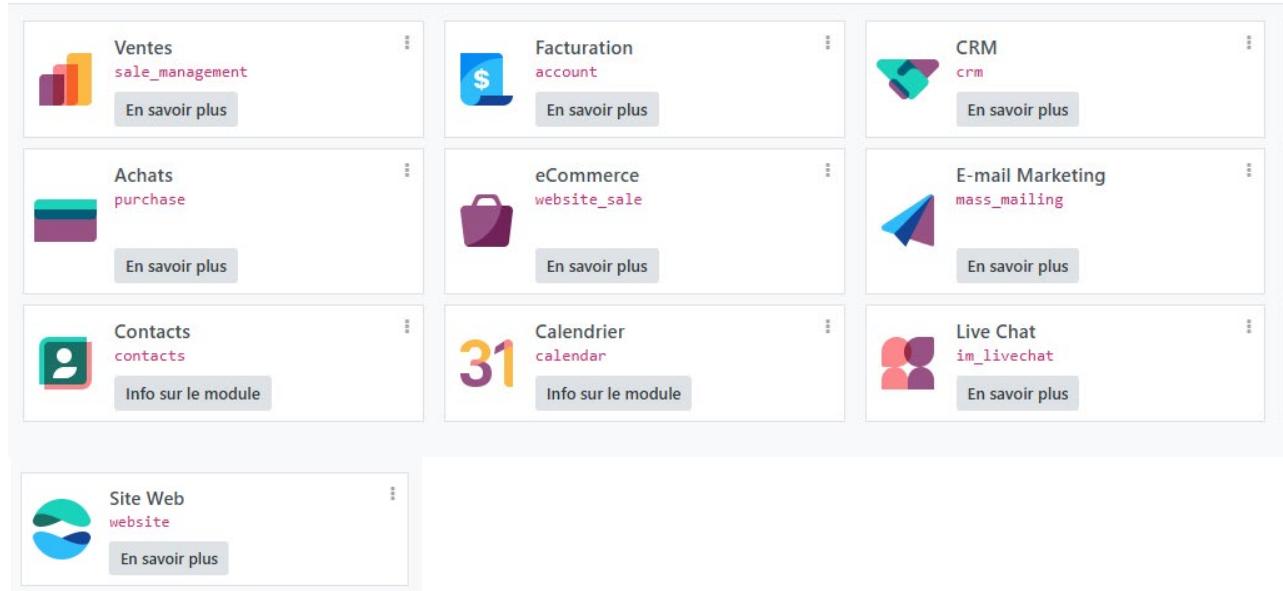


Figure 19 : Tableau de bord des applications Odoo installés

#### 4.3.2. Configuration des Dépendances Critiques

Pour garantir le bon fonctionnement de notre processus métier, plusieurs configurations spécifiques ont été mises en place :

- **Serveur de Messagerie Sortant** : La communication par email avec les clients est essentielle. Un serveur SMTP a été configuré dans les paramètres d'Odoo en utilisant un compte Gmail dédié. Pour des raisons de sécurité, l'authentification à deux facteurs (2FA) a été activée sur ce compte, et un « mot de passe d'application » a été généré pour permettre à Odoo de s'authentifier de manière sécurisée.

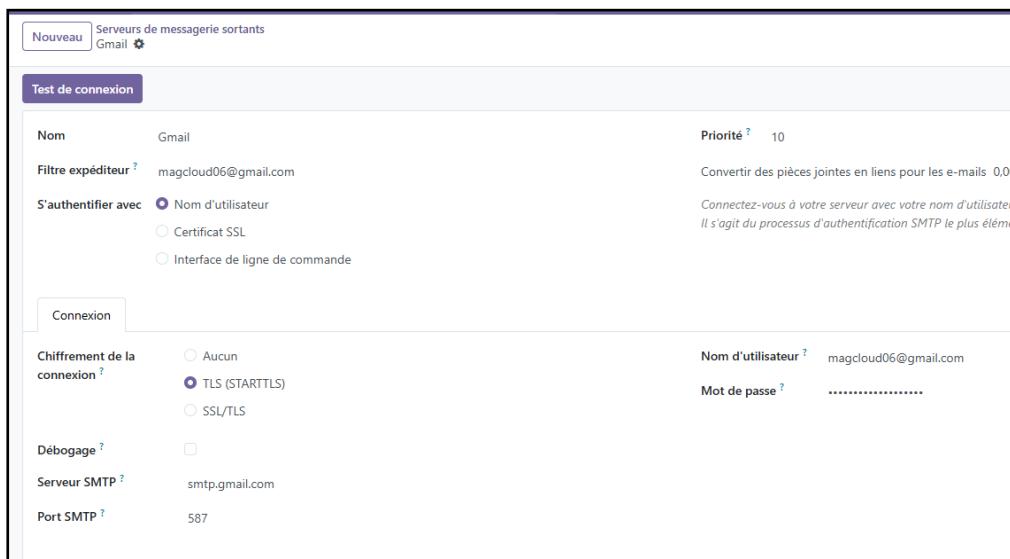


Figure 20 : Écran de Configuration du Serveur de Messagerie dans Odoo

- Module de Tâches Asynchrones (queue\_job)** : Le déploiement d'une VM est une opération qui peut prendre plusieurs minutes. Pour éviter de bloquer l'interface utilisateur d'Odoo pendant ce temps, le module communautaire queue\_job a été installé. Ce module est un prérequis technique majeur car il permet de déléguer les tâches longues à des processus d'arrière-plan, assurant une expérience fluide. (La référence du module est disponible en bibliographie [X]).

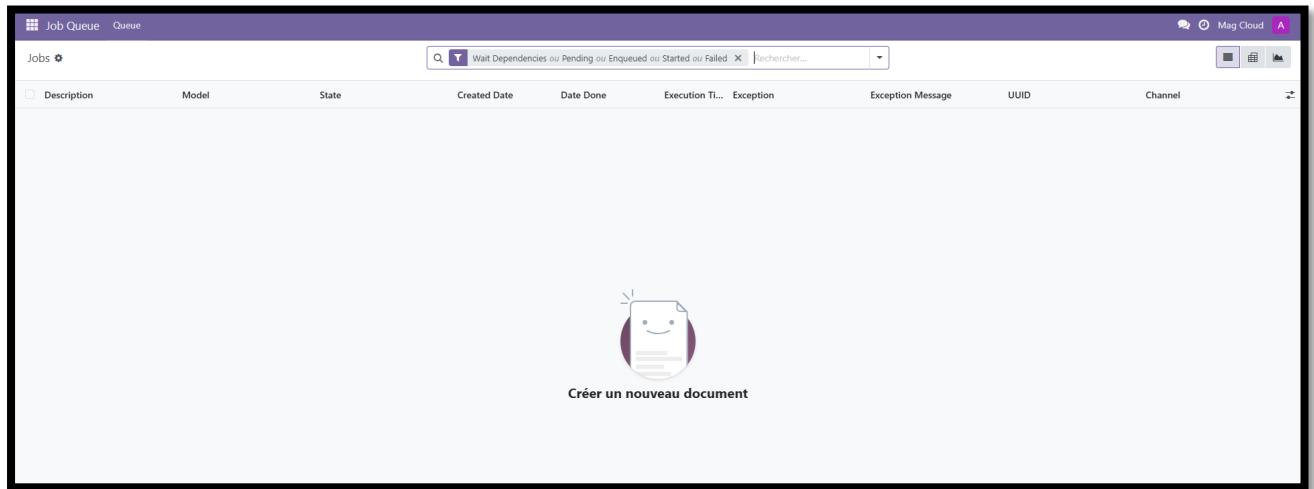


Figure 21 : Interface de supervision du module "Queue Job"

- Localisation Tunisienne (l10n\_tn)** : Pour adapter la facturation au contexte local, le module de localisation tunisien a été installé. Cette étape a permis de charger automatiquement le plan comptable, les journaux et les taxes conformes à la législation en vigueur.

The screenshot shows the 'Tunisie - Comptabilité' module. On the left, there's a sidebar with a 'TN' logo and a link to 'Info sur le module'. The main area has a 'Nouveau Plan comptable' button. Below it is a table with columns: 'Tous', 'Code', 'Nom du compte', and 'Type'. The table lists various account codes with their descriptions and types, such as 'Capital souscrit - non appelé' (Ventes), 'Capital souscrit - appelé, non versé' (Achats), etc.

Tous	Code	Nom du compte	Type
1	101100	Capital souscrit - non appelé	Ventes
2	101200	Capital souscrit - appelé, non versé	Achats
3	101310	Capital non amorti	Banque
4	101320	Capital amorti	Divers
5	101800	Capital souscrit soumis à une réglementation particulière	Divers
6	105000	Fonds de dotation	Espèces
7	108000	Compte de l'exploitant	Divers
	109000	Actionnaires, capital souscrit - non appelé	
	111000	Réserve légale	
	112000	Réserves statutaires	
	117100	Primes d'émission	

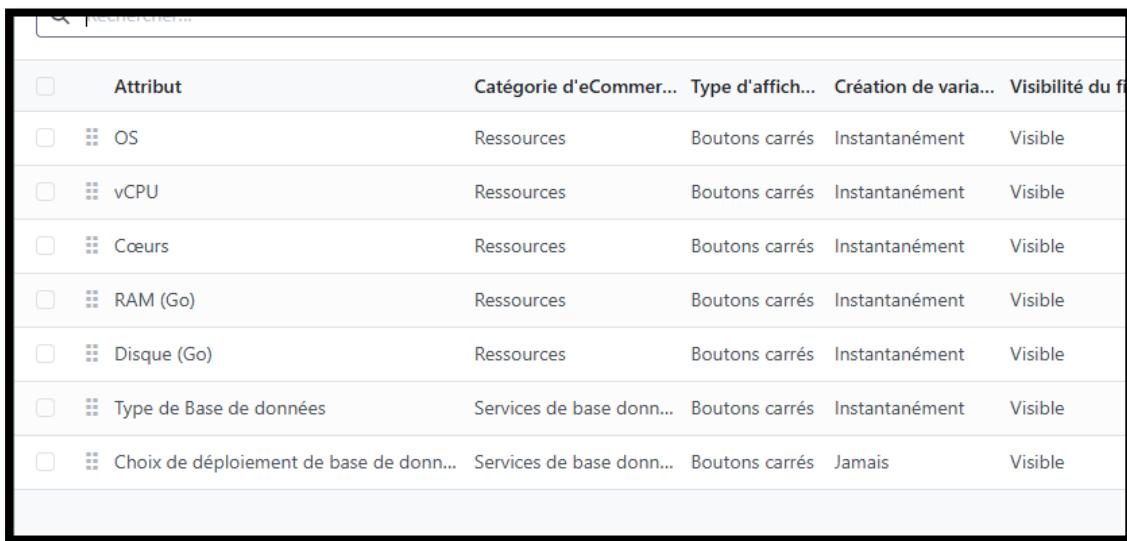
Figure 22 : Journaux et plans comptables chargés

### 4.3.3. Création du Catalogue de Produits et de sa Tarification

Une étape fondamentale de la préparation a été de configurer le catalogue pour la vente de machines virtuelles avec une tarification dynamique.

- **Définition des Attributs :**

- Plusieurs attributs de produit ont été créés pour permettre la personnalisation par le client : OS, vCPU, Cœurs, RAM (Go), et Disque (Go). Pour chaque attribut, différentes valeurs ont été définies.



The screenshot shows a table with the following data:

<input type="checkbox"/> Attribut	Catégorie d'eCommerce	Type d'affichage	Création de variation	Visibilité du filtre
<input type="checkbox"/> OS	Ressources	Boutons carrés	Instantanément	Visible
<input type="checkbox"/> vCPU	Ressources	Boutons carrés	Instantanément	Visible
<input type="checkbox"/> Cœurs	Ressources	Boutons carrés	Instantanément	Visible
<input type="checkbox"/> RAM (Go)	Ressources	Boutons carrés	Instantanément	Visible
<input type="checkbox"/> Disque (Go)	Ressources	Boutons carrés	Instantanément	Visible
<input type="checkbox"/> Type de Base de données	Services de base donn...	Boutons carrés	Instantanément	Visible
<input type="checkbox"/> Choix de déploiement de base de donn...	Services de base donn...	Boutons carrés	Jamais	Visible

Figure 23 : Liste des Attributs créés pour la personnalisation des VMs)

- **Configuration de la Tarification Variable :**

- Pour offrir une tarification transparente et flexible, nous avons adopté un modèle de coût par ressource additionnelle. Un prix de base a été défini pour la configuration minimale de la VM. Ensuite, pour chaque option de personnalisation (vCPU, RAM, Disque), un prix supplémentaire mensuel a été configuré directement dans les attributs du produit sur Odoo.

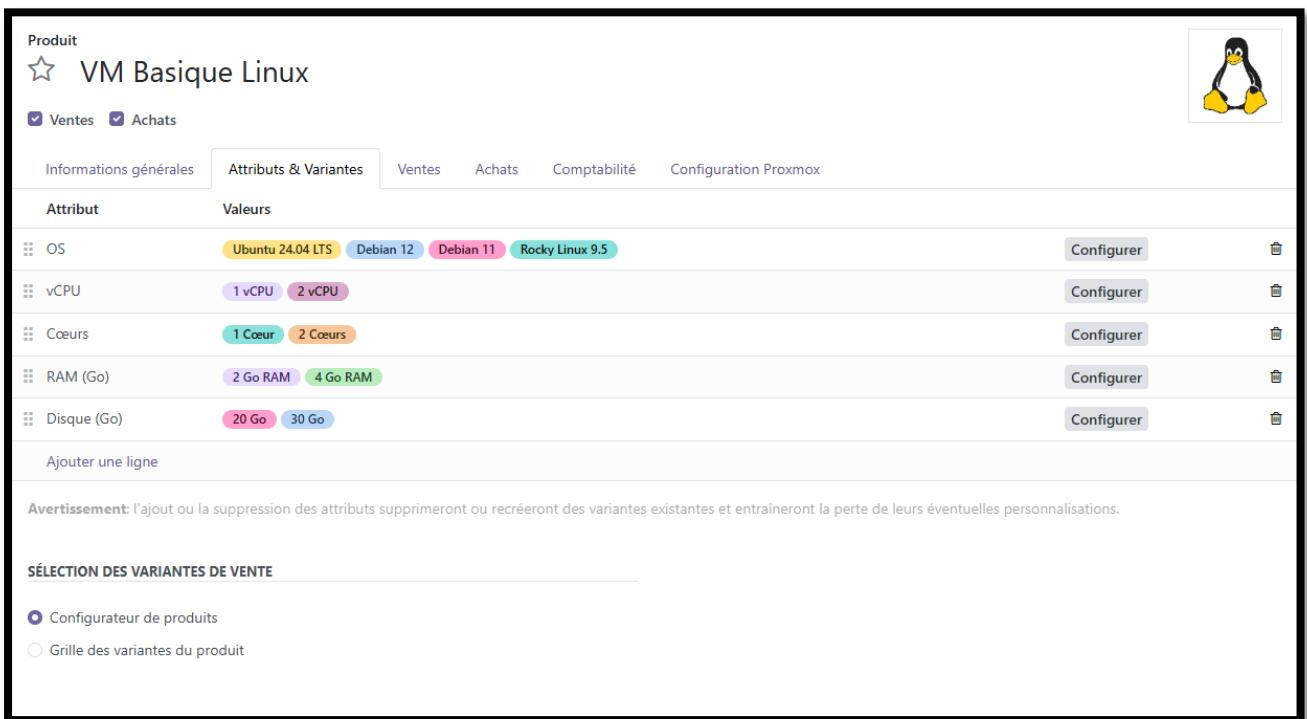
Ce mécanisme permet à Odoo de calculer et d'afficher dynamiquement le prix total en temps réel sur la page du produit, en additionnant le prix de base et les surcoûts des options sélectionnées par le client.

Valeurs d'attribut		
Valeur	Texte libre	Prix supplémentaire par défaut
1 vCPU	<input type="checkbox"/>	10,00
2 vCPU	<input type="checkbox"/>	20,00
4 vCPU	<input type="checkbox"/>	30,00
8 vCPU	<input type="checkbox"/>	40,00
16 vCPU	<input type="checkbox"/>	50,00 <a href="#">+ Ajouter</a>
<a href="#">Ajouter une ligne</a>		

Figure 24 : Configuration des Prix Supplémentaires pour un Attribut

- Association au Produit et Génération des Variantes :

- Ces attributs ont ensuite été associés aux produits. En se basant sur ces configurations, Odoo génère automatiquement toutes les combinaisons possibles (les variantes), chacune avec un prix calculé dynamiquement, prête à être présentée au client sur la boutique en ligne.



The screenshot shows the Odoo product configuration interface for a product named "VM Basique Linux". The top navigation bar includes tabs for "Produit", "Ventes", "Achats", and a logo of Tux. Below the navigation, there are tabs for "Informations générales", "Attributs & Variantes" (which is selected), "Ventes", "Achats", "Comptabilité", and "Configuration Proxmox".

The "Attributs & Variantes" tab displays a list of attributes with their possible values:

- Attribut: OS, Values: Ubuntu 24.04 LTS, Debian 12, Debian 11, Rocky Linux 9.5, with a "Configurer" button and a trash icon.
- Attribut: vCPU, Values: 1 vCPU, 2 vCPU, with a "Configurer" button and a trash icon.
- Attribut: Coeurs, Values: 1 Cœur, 2 Coeurs, with a "Configurer" button and a trash icon.
- Attribut: RAM (Go), Values: 2 Go RAM, 4 Go RAM, with a "Configurer" button and a trash icon.
- Attribut: Disque (Go), Values: 20 Go, 30 Go, with a "Configurer" button and a trash icon.

A note at the bottom states: "Avertissement: l'ajout ou la suppression des attributs supprimeraient ou recréeront des variantes existantes et entraîneront la perte de leurs éventuelles personnalisations."

Below the attributes, there is a section titled "SÉLECTION DES VARIANTES DE VENTE" with two radio buttons:

- Configurateur de produits
- Grille des variantes du produit

Figure 25 : Configuration de produits avec les attributs appropriés

#### 4.3.4. Initialisation de l'interface Web

En parallèle de la configuration du produit, l'aspect visuel de la boutique a été initialisé. Le thème par défaut a été personnalisé avec le logo de l'entreprise et une page d'accueil simple a été esquissée. La conception du design final et l'optimisation de l'expérience utilisateur seront abordées lors du premier sprint



Figure 26 : Initialisation de la Page d'Accueil du Site Web

#### 4.3.5. Initialisation du Module Odoo

Enfin la structure de base de notre module « custom\_proxmox\_integration » a été créée, et le modèle de données initial (décris au Chapitre 3) a été implémenté.

```

1 /opt/odoo/custom-addons/custom_proxmox_integration
2 .
3     controllers
4         __init__.py
5     __init__.py
6     __manifest__.py
7     models
8         account_move.py
9         __init__.py
10        product_template.py
11        res_partner.py
12        sale_order_line.py
13        sale_order.py
14     security
15         ir.model.access.csv
16     views

```

Figure 27 : Arborescence de module

## 4.4. Déploiement des Services de Support et de Sécurité

### 4.4.1. Déploiement de la VM Reverse Proxy (NGINX)

Une VM Debian minimale a été déployée pour jouer le rôle de reverse proxy. NGINX a été configuré pour être le point d'entrée unique de tout le trafic HTTPS et pour rediriger les requêtes vers les services internes appropriés sur leurs ports respectifs (Odoo, Proxmox, Grafana, etc.). Pour cet environnement, des certificats auto-signés ont été générés pour activer le chiffrement TLS.

Cette approche masque complètement notre architecture interne et centralise la gestion de la sécurité des accès web.

```

57      #      proxy_cache_valid 200 60m;
58      #      proxy_buffering on;
59      #      expires 864000; # Cache navigateur pour ~10 jours
60      #      proxy_pass http://odoo_server;
61      #
62
63      # ----- Optimisation : Compression Gzip (Optionnel mais

```

Figure 28 : Extrait de la configuration NGINX redirigeant vers Odoo

### 4.4.2. Déploiement de la Pile de Monitoring

Afin d'obtenir une visibilité complète sur l'état de santé de l'infrastructure, une pile de monitoring a été mise en place dans une machine virtuelle dédiée.

- Composants Installés :
  - Prometheus a été installé et configuré pour collecter les métriques des hôtes Proxmox via un processus de *scraping*.

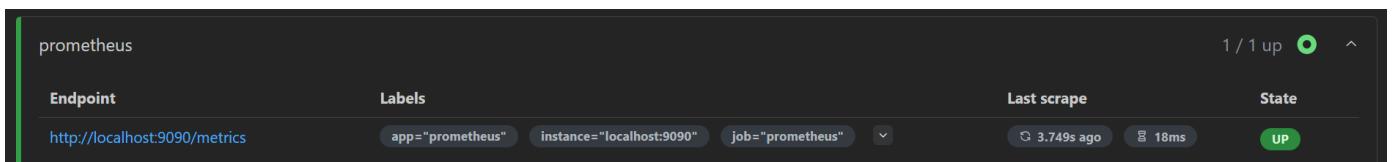
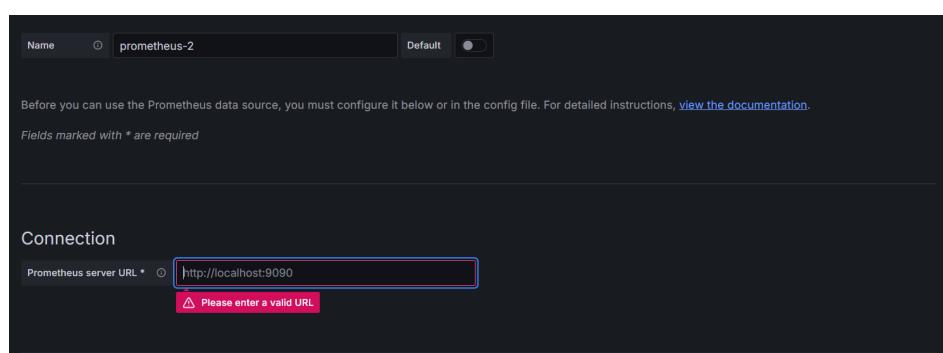


Figure 29 : Point de terminaison Prometheus actif

- Grafana a été déployé pour la visualisation, puis connecté à Prometheus comme source de données.



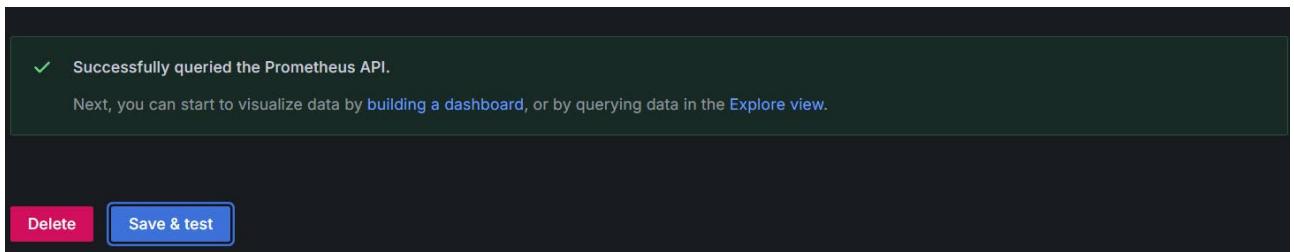


Figure 30 : Connectivité avec Prometheus réussie

- Alertmanager a été installé en préparation de la configuration des futures règles d'alerte.

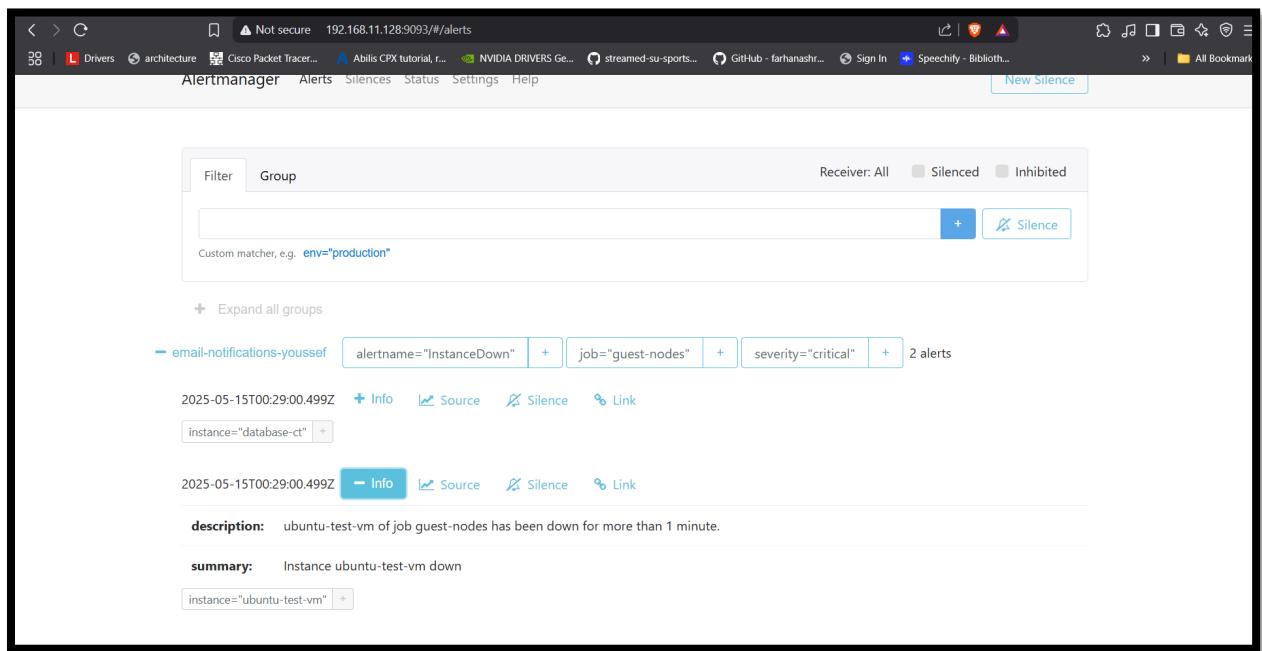


Figure 31 : Alertmanager est opérationnel

- **Collecte des Métriques** : Collecte des Métriques : Pour permettre à Prometheus de superviser l'hyperviseur, l'agent Node Exporter a été installé manuellement sur l'hôte Proxmox VE (SERV01) afin d'exposer les métriques système.

De plus, PVE Exporter a également été installé pour permettre la collecte spécifique des métriques liées à l'infrastructure Proxmox (machines virtuelles, nœuds, stockage, etc.)

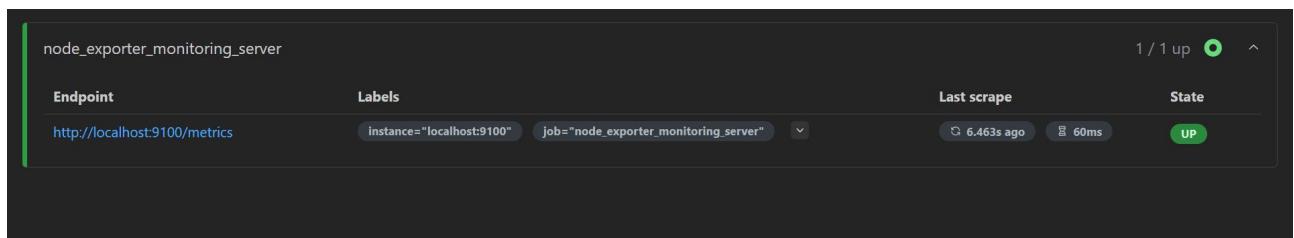


Figure 32 : Node Exporter actif

Endpoint	Labels	Last scrape	State
<a href="http://localhost:9221/metrics">http://localhost:9221/metrics</a>	instance="localhost:9221" job="pve_exporter"	12.443s ago	UP

Figure 33 : Pve Exporter actif

## 4.5. Préparation des Artefacts pour l'Automatisation

Avec l'infrastructure et les services de support en place, cette dernière phase de préparation se concentre sur la mise en place des outils et des artefacts qui seront au cœur de notre pipeline d'automatisation.

### 4.5.1. Installation des Outils d'Infrastructure as Code

Pour permettre à notre VM Odoo d'agir en tant que chef d'orchestre, les outils d'IaC ont été installés directement sur son système d'exploitation.

**Installation de Terraform et Ansible :** Les exécutables de Terraform et d'Ansible ont été installés via les méthodes recommandées pour un environnement Ubuntu. La présence et la version de chaque outil ont été vérifiées en ligne de commande, confirmant que notre orchestrateur était prêt à piloter l'infrastructure.

```
root@MyLaptop:~# terraform --version
Terraform v1.12.1
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.12.2. You can update by downloading from https://developer.hashicorp.com/terraform/install
root@MyLaptop:~#
```

```
root@MyLaptop:~# ansible --version
ansible [core 2.16.3]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.7 (main, Feb 4 2025, 14:46:03) [GCC 14.2.0] (/usr/bin/python3)
  jinja version = 3.1.3
  libyaml = True
root@MyLaptop:~#
```

Figure 34 : Terraform et Ansible Installés

#### 4.5.2. Installation des Outils et Dépendances

Pour permettre à notre module Odoo d'assurer ses fonctions d'orchestration et de sécurité, plusieurs dépendances logicielles ont été installées sur la machine virtuelle

- **Bibliothèques Python spécifiques** : Le module requiert plusieurs bibliothèques Python indispensables pour ses opérations clés, installées via pip :
  - **cryptography** : Cruciale pour la création sécurisée des paires de clés SSH destinées aux clients.
  - **passlib** : Employée pour un hachage robuste des mots de passe injectés via Cloud-Init.
  - **pyyaml** : Utilisée pour la génération dynamique des fichiers de configuration Cloud-Init au format YAML.

#### 4.5.3. Mise en Place de l'Identité d'Automatisation pour Ansible

Pour permettre à Ansible d'agir de manière sécurisée sur les serveurs distants, une identité dédiée a été créée.

1. **Génération de la Paire de Clés SSH** : Sur la VM Odoo, une paire de clés SSH (publique et privée) a été générée avec la commande `ssh-keygen`. Cette paire de clés sera l'identité unique de notre orchestrateur Ansible.

```
odoo@MyLaptop:~$ ssh-keygen -t rsa -b 4096 -f ~/.ssh/ansible_admin_key -C "odoo_ansible_admin_key_for_all_infra"
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /opt/odoo/.ssh/ansible_admin_key
Your public key has been saved in /opt/odoo/.ssh/ansible_admin_key.pub
```

```
odoo@MyLaptop:~$ ls -la ~/.ssh/
total 16
drwx----- 2 odoo odoo 4096 Jun  3 15:33 .
drwxr-xr-x 11 odoo odoo 4096 Jun  3 15:31 ..
-rw-----  1 odoo odoo 3414 Jun  3 15:33 ansible_admin_key
-rw-r--r--  1 odoo odoo   762 Jun  3 15:33 ansible_admin_key.pub
odoo@MyLaptop:~$ cd ~.
odoo@MyLaptop:~$
```

Figure 35 : Clé SSH générée utilisée par Ansible

**2. Déploiement de la Clé Publique :** La clé publique générée a été copiée manuellement dans le fichier `~/.ssh/authorized_keys` des utilisateurs de gestion sur les serveurs cibles :

- Sur l'utilisateur `ansible_admin` de l'hôte Proxmox.
- Sur l'utilisateur `bastion_setup_user` de la VM Bastion.

Cette étape est un prérequis fondamental qui "donne la permission" à notre serveur Odoo de se connecter et d'exécuter des playbooks de configuration sur l'infrastructure.

#### 4.5.4. Préparation des Images et Templates sur Proxmox

Une stratégie d'images à deux niveaux a été mise en place pour répondre aux besoins des différents sprints.

- **Images d'Installation (ISO) :** Pour le déploiement simple du Sprint 1, les images ISO officielles de plusieurs systèmes d'exploitation (notamment Ubuntu, Debian 12, Windows Server 2012) ont été téléchargées et stockées dans le répertoire dédié de Proxmox comme illustré dans la figure 36.

Storage 'local' on node 'hostexample'				
	Name	Date	Format	Size
ISO Images	debian-11.6.0-amd64.iso	2025-05-30 21:14:12	iso	406.85 MB
CT Templates	debian-12.11.0-amd64.iso	2025-05-30 21:15:05	iso	702.55 MB
Permissions	en_windows_server_2012_r2_x64.iso	2025-05-30 21:16:26	iso	4.27 GB
	en-us_windows_server_2019_x64.iso	2025-05-30 21:17:58	iso	5.65 GB
	noble-server-cloudimg-amd64.img	2025-06-04 19:38:37	iso	613.43 MB
	Pfsense.iso	2025-05-30 21:05:55	iso	989.73 MB
	Rocky-9.5-x86_64-minimal.iso	2025-05-04 00:15:15	iso	1.90 GB
	ubuntu-20.04-live-server-amd64.iso	2025-05-30 21:04:25	iso	952.11 MB

Figure 36 : Images ISO déployées sur le stockage de Proxmox

- **Templates « Cloud-Ready » :**

Dans le cadre de l'automatisation complète du Sprint 3, un soin particulier a été apporté à la création d'un « gold template ». Contrairement à une installation classique à partir d'une image ISO, nous avons adopté une méthode plus moderne et fiable basée sur les images Cloud officielles fournies par les distributions Linux (ex. : Ubuntu Cloud Images).

Le processus de création du Template « Cloud-Ready » a suivi les étapes suivantes :

1. **Téléchargement de l'image Cloud** : Une image minimaliste et optimisée au format .qcow2 a été récupérée depuis le dépôt officiel de la distribution.
2. **Création d'une VM initiale dans Proxmox** : Une machine virtuelle vide a été créée, sans disque dur principal ni lecteur CD/DVD.
3. **Importation du disque Cloud** : Le fichier .qcow2 a été importé dans Proxmox à l'aide de la commande `qm importdisk`, puis attaché à la VM en tant que disque principal (scsi0).
4. **Configuration Cloud-Init** : Un lecteur CD-ROM a été ajouté et configuré comme disque **Cloud-Init**, permettant l'injection automatique des données de personnalisation (user-data, meta-data) au démarrage.
5. **Préparation du système** :
  - o Définition du disque importé comme disque de démarrage.
  - o Démarrage unique de la VM pour effectuer des vérifications (langue/clavier en **Fr**, présence du **QEMU Guest Agent**, etc.).

6. **Conversion en Template** : Après arrêt de la VM, celle-ci a été convertie en **Template Proxmox**. Ce Template standardisé est désormais prêt à être cloné rapidement et intégré dans le pipeline d'automatisation. Chaque nouvelle VM bénéficie ainsi d'une base propre, sécurisée et personnalisable via Cloud-Init.

Ce Template est désormais prêt à être cloné rapidement pour créer des VMs pré-configurées comme illustré dans les figures 37 et 38.

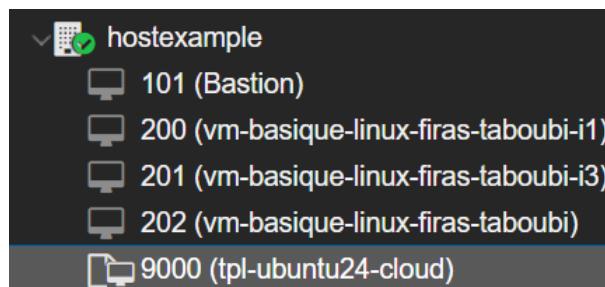


Figure 37 : Template « Cloud-Ready » visible dans l'interface Proxmox

Hour (average)		
HA State	none	hostexample
Node		
Processors	1	
Memory	2.00 GiB	
Bootdisk size	3.50 GiB	

Add	Remove	Edit	Disk Action	Revert
Memory	2.00 GiB			
Processors	1 (1 sockets, 1 cores) [x86-64-v2-AES]			
BIOS	Default (SeaBIOS)			
Display	Default			
Machine	q35			
SCSI Controller	VirtIO SCSI			
CloudInit Drive (ide2)	local-lvm:vm-9000-cloudinit,media=cdrom			
Hard Disk (scsi0)	local-lvm:base-9000-disk-0,size=3584M			
Network Device (net0)	virtio=BC:24:11:0D:2F:B4,bridge=vmbr0			

Summary	Remove	Edit	Regenerate Image
User	ubuntu		
Cloud-Init	Password	*****	
Options	DNS domain	use host settings	
	DNS servers	use host settings	
Task History	SSH public key	none	
Backup	Upgrade packages	Yes	
Replication	IP Config (net0)	ip=dhcp	

Figure 38 : Template Ubuntu configuré

#### 4.5.5. Déploiement Initial de la VM Bastion

Pour préparer l'architecture de sécurité, une machine virtuelle minimale a été déployée pour servir de futur bastion d'accès SSH.

- Mise en Place :** Une VM Rocky a été créée avec des ressources minimales. Sa configuration réseau initiale a été effectuée pour la placer sur le réseau public (vmbr0), la rendant accessible pour les futures configurations qui seront appliquées par Ansible lors du Sprint 3.

Bastion	
Status	stopped
HA State	none
Node	hostexample
CPU usage	0.00% of 1 CPU(s)
Memory usage	0.00% (0 B of 4.00 GiB)
Bootdisk size	32.00 GiB
IPS	Guest Agent not running

Figure 39 : VM Bastion dans la liste des VMs de Proxmox

#### 4.6. Conclusion

La phase de préparation est désormais achevée, nous laissant un socle technique robuste et pérenne pour la suite du projet. Nous avons mis en place un environnement de développement complet et stable. Concrètement, l'infrastructure d'hébergement et de sauvegarde est opérationnelle, les services essentiels de gestion, de sécurité et de supervision sont actifs, et les outils de notre chaîne d'automatisation ont été initialisés.

Cette fondation solide nous permet d'aborder la phase de développement avec confiance et méthode. Le chapitre suivant marquera le lancement de cette nouvelle étape avec le premier sprint de développement. Son objectif sera de livrer un Produit Minimum Viable (MVP) afin de valider la pertinence et la fluidité du parcours client que nous avons conçu.

# Chapitre 5 Sprint 1 : Fondation et Parcours Clients (MVP)

## 5.1. Introduction :

Après une phase de préparation ayant permis de poser des bases techniques solides, le projet entre dans une étape concrète : le développement. Ce chapitre présente le premier sprint, une étape clé de notre méthode Agile.

## 5.2. Objectif attendu :

L'objectif de ce premier sprint est de livrer un Produit Minimum Viable (MVP) centré sur un parcours client complet : de la commande à la réception d'une VM simple. Il s'agit également de fournir à l'administrateur des interfaces de configuration et de supervision fiables et à jour.

Ce sprint permettra de valider les choix techniques et métiers critiques du projet, en s'appuyant sur l'API Proxmox, des images ISO, et une configuration réseau manuelle. Le déploiement réalisé restera basique, mais posera les bases techniques pour les évolutions futures.

## 5.3. Analyse Fonctionnelle : Définition des Besoins du Sprint

Avant de planifier le développement, une analyse fonctionnelle a été menée pour définir précisément les interactions des utilisateurs avec le système dans le périmètre de ce premier sprint. Cette étape permet de s'assurer que les fonctionnalités développées répondent à des besoins concrets et identifiés.

### 5.3.1. Identification des Acteurs et de leurs Rôles

Pour ce sprint initial, deux acteurs principaux interagissent avec la solution. Leurs rôles et objectifs, définis lors de l'analyse globale du projet (voir Chapitre 3), sont rappelés ici :

- **L'Administrateur** : Il est le garant technique de la plateforme. Ses objectifs dans ce sprint sont de pouvoir configurer l'intégration entre Odoo et Proxmox, de définir quels produits sont éligibles à l'automatisation, et de superviser l'ensemble des déploiements depuis une interface centralisée dans Odoo.
- **Le Client Final** : Il est le consommateur du service. Son parcours se concentre sur la commande d'une machine virtuelle via la boutique en ligne et la réception des informations nécessaires pour y accéder et commencer à l'utiliser.

### 5.3.2. Modélisation des Cas d'Utilisation

Les interactions entre les acteurs et le système, dans le périmètre fonctionnel défini pour ce sprint, sont représentées par des diagrammes de cas d'utilisation.

Le diagramme suivant (Figure 5.1) modélise les cas d'utilisation pertinents pour le rôle de l'Administrateur dans ce sprint.

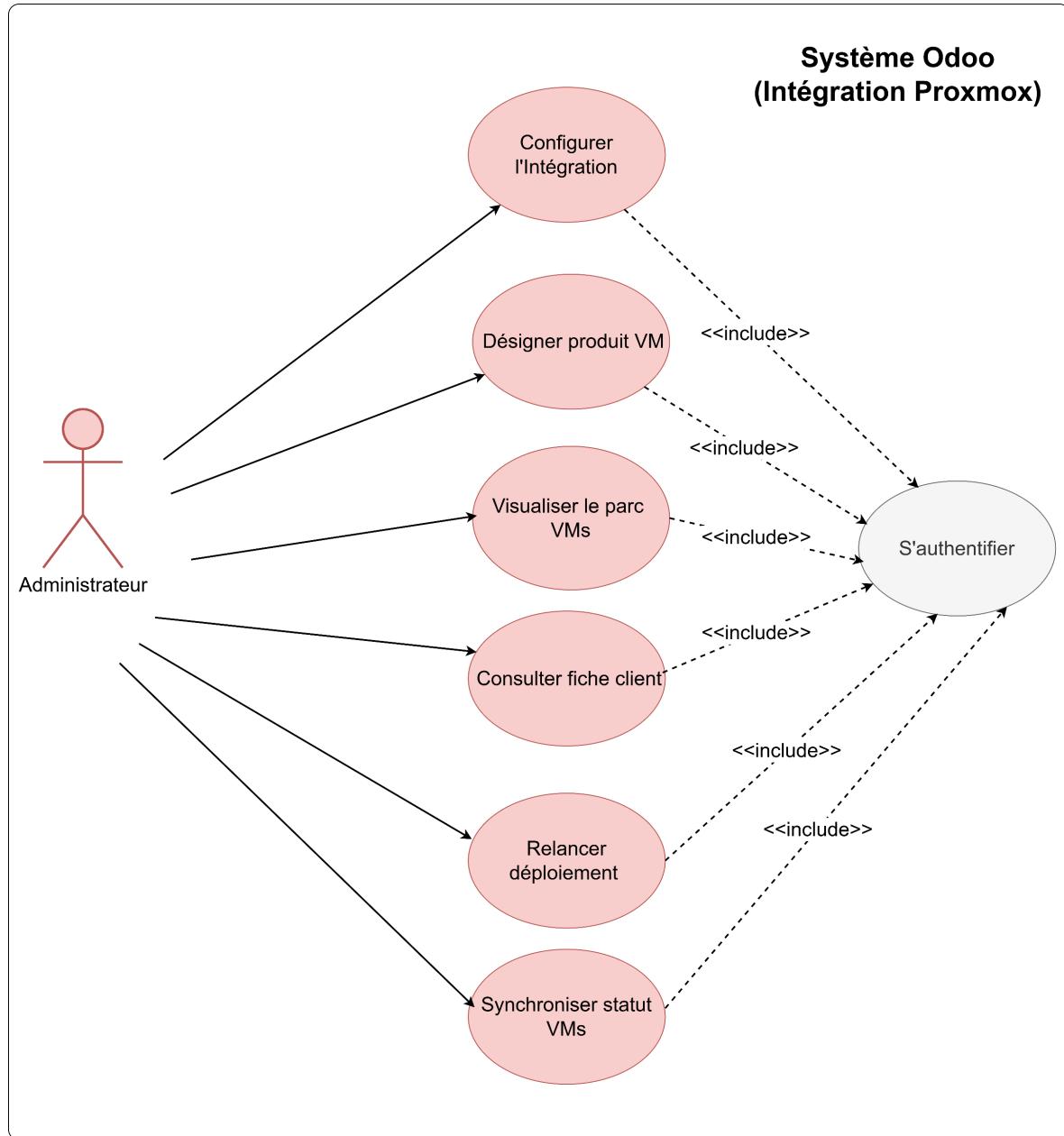


Figure 40 : Diagramme des Cas d'Utilisation - Rôle Administrateur (Sprint 1)

Le diagramme ci-dessous (Figure 5.2) présente les cas d'utilisation couverts pour le rôle du Client Final au cours de ce sprint.

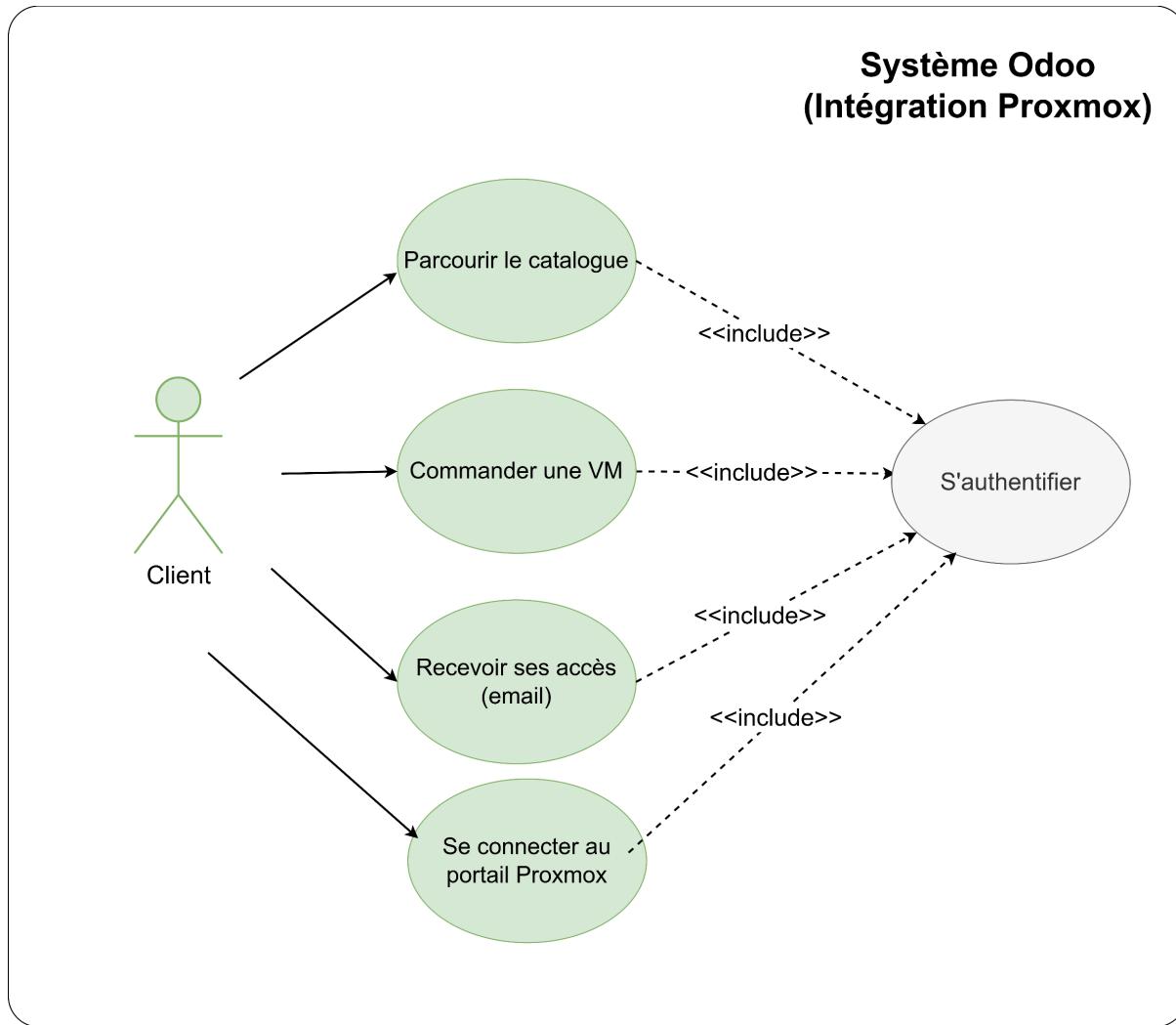


Figure 41 : Diagramme des Cas d'Utilisation - Rôle Client (Sprint 1)

### 5.3.3. Spécifications Détallées des Cas d'Utilisation.

Afin de compléter la modélisation par diagrammes, les cas d'utilisation clés de ce sprint sont décrits de manière détaillée. Cette spécification permet de définir le comportement attendu du système, tant dans les scénarios standards que dans les cas d'exception.

#### 5.3.3.1. Spécifications des Cas d'Utilisation de l'Administrateur

Les interactions de l'administrateur avec le système, couvrant la configuration et la supervision dans ce sprint, sont spécifiées ci-dessous.

Le tableau 7 ci-dessous décrit le cas d'utilisation « Configurer l'intégration », où l'administrateur règle la connexion et les options du module.

Tableau 12 : Spécification du Cas d'Utilisation 'Configurer l'intégration'

Section	Description
Nom	Configurer l'intégration Proxmox
Acteur	Administrateur
Objectif	Permettre à l'administrateur de définir les paramètres nécessaires à la connexion et au fonctionnement de l'intégration Odoo-Proxmox.
Précondition	L'administrateur est connecté au back-end Odoo et dispose des droits d'accès aux paramètres techniques.
Déclencheur	L'administrateur souhaite modifier ou vérifier les paramètres de l'intégration Proxmox.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'administrateur navigue vers le menu « <b>Paramètres</b> » &gt; « <b>Général</b> ».</li> <li>2. Il trouve et sélectionne la section ou le menu lié à l'intégration Proxmox.</li> <li>3. Le système affiche le formulaire de configuration de l'intégration Proxmox.</li> <li>4. L'administrateur consulte et/ou modifie les différents champs (URL API, identifiants, stockages par défaut, plages VMID, paramètres réseau, templates mails, etc.).</li> <li>5. L'administrateur enregistre les modifications.</li> <li>6. Le système valide les données (format, etc.) et enregistre la configuration.</li> </ol>
Scénarios d'Exception	<p>A1 : Paramètres API invalides : Si les identifiants ou l'URL de l'API Proxmox sont incorrects, une erreur est affichée lors d'un appel API utilisant ces paramètres.</p> <p>A2 : Champs obligatoires manquants : Le système empêche l'enregistrement si des champs marqués comme obligatoires (ex: Hôte Proxmox, Nœud par défaut) sont vides.</p>
Postcondition	Les paramètres de l'intégration Proxmox sont mis à jour dans le système Odoo.

Le tableau 8 ci-dessous décrit le cas d'utilisation « Désigner produit VM », où l'administrateur sélectionne les produits de la boutique liés à l'automatisation Proxmox.

**Tableau 13 : Spécification du Cas d'Utilisation "Désigner produit VM"**

Section	Description
Nom	Désigner produit VM Proxmox
Acteur	Administrateur
Objectif	Marquer un produit dans Odoo comme étant éligible au processus d'automatisation de déploiement de machine virtuelle sur Proxmox.
Précondition	L'administrateur est connecté au back-end Odoo et un produit de vente existe dans le catalogue.
Déclencheur	L'administrateur souhaite qu'un produit spécifique (ou une variante) entraîne le déploiement d'une VM lors de sa commande.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'administrateur navigue vers le menu « Ventes » &gt; « Produits ».</li> <li>2. Il sélectionne le produit ou la variante qu'il souhaite marquer.</li> <li>3. Il accède au formulaire d'édition du produit.</li> <li>4. Dans l'onglet « Proxmox », l'administrateur coche la case "Est un produit VM Proxmox".</li> <li>5. L'administrateur enregistre les modifications.</li> <li>6. Le système enregistre l'état du champ pour ce produit.</li> </ol>
Scénarios d'Exception	Aucune exception majeure n'est prévue pour cette simple action de marquage. Le système gère l'enregistrement standard.
Postcondition	Le produit est désormais identifié par le système comme un produit de machine virtuelle Proxmox.

Le tableau 9 ci-dessous décrit le cas d'utilisation « Visualiser le parc VMs », où l'administrateur consulte l'ensemble des machines virtuelles gérées par le système.

**Tableau 14 : Spécification du Cas d'Utilisation 'Visualiser le parc VMs'**

Section	Description
Nom	Visualiser le parc de Machines Virtuelles
Acteur	Administrateur
Objectif	Afficher une liste centralisée de toutes les machines virtuelles déployées via l'intégration, avec leurs informations clés et leur statut.
Précondition	L'administrateur est connecté au back-end Odoo et au moins une machine virtuelle a été déployée et enregistrée dans Odoo.
Déclencheur	L'administrateur souhaite consulter la liste des VMs gérées par le système.

Scénario Nominal	<ol style="list-style-type: none"> <li>L'administrateur navigue vers le menu principal dédié à l'intégration Proxmox « Intégration Proxmox » &gt; « VMs Déployés ».</li> <li>Le système affiche la liste de toutes les machines virtuelles, incluant des informations telles que le nom, VMID, statut (Démarrée/Arrêtée/En pause), client associé, commande d'origine.</li> </ol>
Scénarios d'Exception	Aucune exception majeure au niveau de l'affichage de la liste standard. Le système affiche une liste vide si aucune VM n'est enregistrée.
Postcondition	L'administrateur a accès à la liste des machines virtuelles déployées.

Le tableau 10 ci-dessous décrit le cas d'utilisation « Consulter fiche client », où l'administrateur accède à un résumé des ressources Proxmox depuis la fiche du client concerné.

Tableau 15 : Spécification du Cas d'Utilisation 'Consulter fiche client'

Section	Description
Nom	Consulter fiche client (Infos Proxmox)
Acteur	Administrateur
Objectif	Afficher un résumé des ressources Proxmox (utilisateur, pool, liste des VMs) associées à un client spécifique directement sur sa fiche Odoo.
Précondition	L'administrateur est connecté au back-end Odoo et le client a été provisionné dans Proxmox et possède des VMs déployées via l'intégration.
Déclencheur	L'administrateur consulte la fiche d'un client pour obtenir une vue complète de ses informations, incluant ses ressources Proxmox.
Scénario Nominal	<ol style="list-style-type: none"> <li>L'administrateur navigue vers le menu « <b>Contacts</b> »</li> <li>Il sélectionne le client dont il souhaite consulter la fiche.</li> <li>Le système affiche le formulaire de la fiche client.</li> <li>Dans une section dédiée sur la fiche client, le système affiche les identifiants Proxmox (utilisateur, pool) ainsi qu'un résumé des machines virtuelles associées à ce client.</li> </ol>
Scénarios d'Exception	Aucune exception majeure. Si le client n'a pas de ressources Proxmox associées, la section dédiée est vide ou non affichée.
Postcondition	L'administrateur dispose d'une vue consolidée des informations client et de ses ressources Proxmox.

Le tableau 11 ci-dessous décrit le cas d'utilisation « Relancer déploiement », où l'administrateur intervient manuellement en cas d'échec du déclenchement initial.

**Tableau 16 : Spécification du Cas d'Utilisation 'Relancer déploiement'**

Section	Description
Nom	Relancer un déploiement depuis la facture
Acteur	Administrateur
Objectif	Permettre à l'administrateur de déclencher manuellement le processus d'automatisation de la configuration Proxmox pour une commande de vente, en cas de besoin (ex: échec du déclenchement automatique).
Précondition	L'administrateur est connecté au back-end Odoo et une facture est liée à une commande de vente contenant un produit VM pour lequel le déclenchement automatique n'a pas eu lieu ou a échoué.
Déclencheur	L'administrateur constate qu'une commande de vente n'a pas été provisionnée correctement et souhaite initier le processus manuellement via la facture liée.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'administrateur navigue vers la facture liée à la commande de vente concernée.</li> <li>2. Le système affiche les informations de la facture, incluant le statut de déclenchement Proxmox pour les commandes liées</li> <li>3. L'administrateur clique sur le bouton permettant de déclencher la configuration Proxmox.</li> <li>4. Le système vérifie les commandes liées et déclenche le processus d'automatisation pour celles qui n'ont pas encore été traitées.</li> <li>5. Un message confirme que la tâche a été mise en file d'attente.</li> </ol>
Scénarios d'Exception	<p>A1 : Déploiement déjà déclenché : Si le processus a déjà été déclenché pour toutes les commandes liées à cette facture, le système informe l'administrateur qu'il n'y a rien à faire.</p> <p>A2 : Facture non liée à une commande VM : Si la facture ne contient pas de ligne liée à une commande de vente avec un produit VM, l'action n'est pas pertinente.</p>
Postcondition	Le processus d'automatisation Proxmox a été ajouté à la file d'attente pour les commandes en attente liées à la facture.

Le tableau 12 ci-dessous décrit le cas d'utilisation « Synchroniser statut VMs », où l'administrateur met à jour manuellement les informations des machines virtuelles de Proxmox vers Odoo.

**Tableau 17 : Spécification du Cas d'Utilisation 'Synchroniser statut VMs'**

Section	Description
Nom	Synchroniser statut des Machines Virtuelles
Acteur	Administrateur
Objectif	Obtenir en temps réel le statut des machines virtuelles directement depuis Proxmox et mettre à jour les informations dans Odoo.
Précondition	L'administrateur est connecté au back-end Odoo et des machines virtuelles sont enregistrées dans Odoo.
Déclencheur	L'administrateur souhaite obtenir les informations les plus récentes pour une ou plusieurs VMs, sans attendre la synchronisation automatique.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'administrateur navigue vers la vue liste des machines virtuelles</li> <li>2. Il sélectionne la ou les VMs dont il souhaite synchroniser le statut.</li> <li>3. Il déclenche l'action « Synchroniser statut depuis Proxmox »</li> <li>4. Le système interroge l'API Proxmox pour chaque VM sélectionnée pour obtenir son statut et ses informations (ex : IP).</li> <li>5. Le système met à jour les machines virtuelles correspondants dans Odoo.</li> <li>6. Un message confirme le succès de l'opération de synchronisation.</li> </ol>
Scénarios d'Exception	<p>A1 : Échec de l'API Proxmox : Si une VM n'est pas trouvée sur Proxmox, ou si l'API renvoie une erreur, le système loggue l'incident et peut marquer la VM dans Odoo avec un statut inconnu</p> <p>A2 : Pas de VMs sélectionnées : L'action ne se déclenche pas si aucune VM n'est sélectionnée.</p>
Postcondition	Les informations des machines virtuelles sélectionnées dans Odoo reflètent leur état actuel dans Proxmox.

### 5.3.3.2. Spécifications des Cas d'Utilisation du Client Final

Les interactions du client final avec le système, centrées sur le parcours d'achat et l'accès initial à ses ressources dans ce sprint, sont spécifiées ci-dessous.

Le tableau 13 ci-dessous décrit le cas d'utilisation « Parcourir le catalogue », où le client explore la boutique en ligne pour choisir l'offre de VM qui lui convient.

Tableau 18 : Spécification du Cas d'Utilisation 'Parcourir le catalogue'

Section	Description
Nom	Parcourir le catalogue
Acteur	Client Final
Objectif	Permettre au client de consulter les différentes offres de machines virtuelles disponibles sur la boutique en ligne.
Précondition	Le client peut accéder au site web Odoo hébergeant la boutique.
Déclencheur	Le client souhaite découvrir ou sélectionner une offre de service VM.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. Le client accède à la section "Boutique" du site web Odoo.</li> <li>2. Il navigue potentiellement à travers les catégories de produits.</li> <li>3. Il consulte la liste des produits disponibles.</li> <li>4. Il ouvre la page d'un produit spécifique pour voir ses détails et les options de configuration.</li> </ol>
Scénarios d'Exception	Aucune exception spécifique liée à l'intégration Proxmox à ce niveau. Les erreurs standards du site web (page non trouvée) s'appliquent.
Postcondition	Le client a visualisé des informations sur les produits VMs proposés.

Le tableau 14 ci-dessous décrit le cas d'utilisation « Commander une VM », où le client achète une machine virtuelle choisie dans le catalogue.

**Tableau 19 : Spécification du Cas d'Utilisation 'Commander une VM'**

Section	Description
Nom	Commander une Machine Virtuelle
Acteur	Client Final
Objectif	Permettre au client de sélectionner une offre de VM, l'ajouter à son panier, et finaliser le processus d'achat via la boutique en ligne.
Précondition	Le client a trouvé l'offre de VM désirée (souvent après 'Parcourir le catalogue'). Le client est authentifié.
Déclencheur	Le client décide d'acheter une machine virtuelle.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. Depuis la page d'un produit VM, le client sélectionne les options souhaitées</li> <li>2. Il ajoute le produit à son panier.</li> <li>3. Il navigue vers la page du panier.</li> <li>4. Il procède au paiement/validation de la commande.</li> <li>5. Il renseigne ou confirme ses informations (adresse, etc.).</li> <li>6. Il confirme la commande.</li> <li>7. Il effectue un virement bancaire, comme indiqué sur la facture générée après la confirmation de la commande.</li> </ol>
Scénarios d'Exception	Champs manquants : Le système requiert les informations nécessaires avant confirmation.
Postcondition	Une commande de vente est créée dans Odoo. Potentiellement, une facture est générée et son état de paiement déclenchera l'automatisation.

Le tableau 15 ci-dessous décrit le cas d'utilisation « Recevoir ses accès (mail) », où le client reçoit par courriel les informations nécessaires pour se connecter à son environnement Proxmox.

Tableau 20 : Spécification du Cas d'Utilisation 'Recevoir ses accès (email)'

Section	Description
Nom	Recevoir ses accès (email)
Acteur	Client Final
Objectif	Obtenir ses identifiants et l'URL de connexion pour accéder au portail de gestion Proxmox de ses machines virtuelles.
Précondition	Le client a commandé et payé une VM. Le système a réussi à créer un compte utilisateur Proxmox dédié pour ce client. Le client a une adresse mail valide enregistrée dans Odoo.
Déclencheur	La finalisation réussie du processus de création de compte utilisateur Proxmox automatisé.
Scénario Nominal	<ol style="list-style-type: none"> <li>Le système Odoo, à la suite de la création du compte Proxmox, compose un mail contenant l'identifiant utilisateur Proxmox, le mot de passe initial et l'URL du portail.</li> <li>Le système envoie ce mail à l'adresse email enregistrée du client</li> <li>Le client consulte sa boîte de réception et trouve le mail.</li> </ol>
Scénarios d'Exception	<p>A1 : Adresse mail invalide/inaccessible : L'email ne parvient pas au client.</p> <p>A2 : Échec de la création du compte Proxmox : L'email n'est pas envoyé si l'étape précédente du provisionnement échoue.</p>
Postcondition	Le client est en possession des informations de connexion nécessaires pour accéder à son environnement Proxmox.

Le tableau ci-dessous décrit le cas d'utilisation « Se connecter au portail Proxmox », où le client accède pour la première fois à l'interface technique de Proxmox.

**Tableau 21 : Spécification du Cas d'Utilisation 'Se connecter au portail Proxmox'**

Section	Description
Nom	Se connecter au portail Proxmox
Acteur	Client Final
Objectif	Accéder à l'interface web de Proxmox pour visualiser et gérer (opérations de base) ses machines virtuelles dédiées.
Précondition	Le client a reçu ses identifiants Proxmox par mail. Le portail Proxmox est accessible depuis son navigateur web.
Déclencheur	Le client souhaite gérer ses machines virtuelles.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. Le client ouvre son navigateur web et navigue vers l'URL du portail Proxmox.</li> <li>2. La page de connexion Proxmox s'affiche.</li> <li>3. Le client saisit son identifiant utilisateur Proxmox, son mot de passe et sélectionne le domaine d'authentification (Realm) correct (ex: pve).</li> <li>4. Le client clique sur "Login".</li> <li>5. Le système Proxmox authentifie l'utilisateur et affiche l'interface de gestion, limitée au pool de ressources et aux VMs qui lui sont alloués.</li> </ol>
Scénarios d'Exception	<p>A1 : Identifiants incorrects : La connexion échoue. Le système Proxmox affiche un message d'erreur d'authentification.</p> <p>A2 : Domaine incorrect : La connexion échoue si le bon domaine n'est pas sélectionné.</p> <p>A3 : Problème réseau/serveur : La page de connexion n'est pas accessible ou la connexion échoue.</p>
Postcondition	Le client est authentifié sur le portail Proxmox et a accès à la vue de ses ressources.

#### 5.4. Planification du Développement (Sprint Planning)

À partir de l'analyse fonctionnelle définissant les besoins pour ce premier incrément, l'équipe de développement a réalisé la planification du sprint. Cette étape, clé en méthode Agile, permet de fixer l'objectif du sprint, de déterminer les fonctionnalités à développer, et d'organiser le travail.

### 5.4.1. Périmètre du Sprint (Sprint Backlog)

En lien avec l'objectif du sprint (voir section 5.2), les User Stories nécessaires ont été sélectionnées depuis le Product Backlog. Elles forment le Sprint Backlog, qui regroupe les fonctionnalités que l'équipe s'engage à livrer à la fin du sprint. Le Tableau 17 présente ces User Stories avec leurs identifiants et descriptions.

Tableau 22 : User Stories du Sprint 1

ID	Epic	User Story
US-01	Expérience Client	En tant que Client, je veux parcourir un catalogue et sélectionner une offre de VM afin de commander un service adapté à mes besoins.
US-02	Expérience Client	En tant que Client, je veux me connecter à un portail Proxmox afin d'effectuer des opérations de base (Démarrer/Arrêter, Console) sur mes VMs.
US-03	Expérience Client	En tant que Client, je veux recevoir un mail de bienvenue avec mes identifiants Proxmox afin de pouvoir accéder à mon espace de gestion.
US-07	Administration Odoo	En tant qu'Admin, je veux une page de configuration centrale afin de gérer facilement les paramètres de l'intégration.
US-08	Administration Odoo	En tant qu'Admin, je veux désigner un produit comme "VM Proxmox" afin de le rendre éligible à l'automatisation.
US-11	Administration Odoo	En tant qu'Admin, je veux une vue centralisée de toutes les VMs déployées afin de suivre rapidement l'état du parc.
US-12	Administration Odoo	En tant qu'Admin, je veux un résumé des infos Proxmox sur la fiche client afin d'avoir une vue à 360° sans changer d'écran.
US-13	Administration Odoo	En tant qu'Admin, je veux pouvoir relancer un déploiement depuis la facture afin de gérer manuellement un cas d'erreur.
US-18	Automatisation	En tant que Système, je veux créer automatiquement un compte et un pool Proxmox afin d'isoler administrativement les ressources de chaque client.
US 24	Automatisation	En tant que Système, je veux déployer une Machine Virtuelle à partir d'une image ISO pour une commande client afin de fournir le service commandé
US-29	Exploitation	En tant qu'Admin, je veux pouvoir synchroniser manuellement le statut d'une ou plusieurs VMs depuis Odoo afin d'obtenir leur état en temps réel.
US-30	Exploitation	En tant que Système, je veux synchroniser périodiquement le statut de toutes les VMs gérées afin que les informations dans Odoo restent à jour.

### 5.4.2. Plan d'Action

Une fois le périmètre défini, chaque User Story a été décomposée en tâches techniques concrètes. Le Tableau 5.5 présente cette décomposition pour chaque User Story retenue.

**Tableau 23 : Décomposition des Tâches Techniques du Sprint 1**

ID User Story	Tâches Techniques à Réaliser
US-01	<ul style="list-style-type: none"> <li>- Configurer un produit "Machine Virtuelle" dans le catalogue Odoo en utilisant les fonctionnalités standards de produits et variantes.</li> <li>- Tester le flux de commande standard d'un produit configuré.</li> </ul>
US-02	Cette User Story est réalisée par l'ensemble des tâches permettant au client d'obtenir les accès (US-18, US-03) et d'avoir une VM déployée (US-30). Aucune tâche de développement spécifique n'est requise pour l'interface Proxmox dans ce sprint.
US-03	<ul style="list-style-type: none"> <li>- Développer la logique pour envoyer un email des identifiants Proxmox au client</li> <li>- Créer le modèle (template) XML pour cet email.</li> </ul>
US-07	<ul style="list-style-type: none"> <li>- Créer la section dédiée à l'intégration Proxmox dans les paramètres système d'Odoo</li> <li>- Ajouter tous les champs nécessaires à la configuration de la connexion et des paramètres par défaut de l'intégration.</li> </ul>
US-08	<ul style="list-style-type: none"> <li>- Ajouter un champ booléen au modèle des produits pour marquer les produits éligibles à l'automatisation Proxmox.</li> </ul>
US-11	<ul style="list-style-type: none"> <li>- Créer un nouveau modèle Odoo dédié au parc de machines virtuelles pour stocker les informations pertinentes des VMs déployées.</li> <li>- Créer un menu d'accès et une vue liste pour afficher toutes les VMs enregistrées dans Odoo.</li> </ul>
US-12	<ul style="list-style-type: none"> <li>- Ajouter les champs pour stocker l'identifiant utilisateur Proxmox (pve_user_id) et l'identifiant de pool Proxmox (pve_pool_id) sur le modèle Contact (res.partner).</li> </ul>

	<ul style="list-style-type: none"> <li>- Ajouter ces champs à la vue formulaire du Contact pour les rendre visibles.</li> <li>- Créer un champ calculé (HTML) sur le modèle Contact pour afficher un tableau récapitulatif des VMs associées.</li> <li>- Ajouter ce champ calculé à la vue formulaire du Contact.</li> </ul>
US-13	<ul style="list-style-type: none"> <li>- Étendre les modèles de facture (account.move) et leurs lignes (account.move.line) pour lier les factures aux commandes de vente et identifier les produits VM.</li> <li>- Implémenter la logique sur la facture pour détecter le paiement et appeler le processus de déclenchement automatique</li> <li>- Ajouter un bouton d'action (manuel) sur la vue de la facture pour permettre le re déclenchement.</li> </ul>
US-18	<ul style="list-style-type: none"> <li>- Développer la méthode principale d'orchestration du provisionnement client (utilisateur et pool) sur Proxmox.</li> <li>- Implémenter la logique d'appels à l'API Proxmox (via le mixin) pour vérifier l'existence, créer l'utilisateur, créer le pool, et appliquer les permissions (ACL)</li> <li>- Implémenter la logique de génération des identifiants utilisateur Proxmox et identifiant de pool uniques et normalisés..</li> <li>- Implémenter le stockage des identifiants Proxmox générés ou trouvés sur l'enregistrement Contact correspondant.</li> <li>- Implémenter la logique de génération de mot de passe aléatoire et sécurisé pour les nouveaux utilisateurs Proxmox.</li> </ul>
US-24	<ul style="list-style-type: none"> <li>- Développer la méthode de déploiement d'une machine virtuelle sur Proxmox à partir d'une image ISO.</li> <li>- Implémenter la logique pour trouver un VMID disponible dans la plage configurée.</li> <li>- Implémenter la logique pour générer un nom unique et conforme pour la VM Proxmox.</li> <li>- Récupérer et parser les attributs de la ligne de commande (via le produit) pour définir les spécifications de la VM (RAM, CPU, Disque, OS).</li> <li>- Construire et envoyer la charge utile (payload) pour l'appel API Proxmox pour créer la VM.</li> </ul>

	<ul style="list-style-type: none"> <li>- Implémenter une fonction pour attendre et vérifier le succès de la tâche de création sur Proxmox et récupérer son journal d'exécution.</li> <li>- Une fois la création réussie, créer un enregistrement de machine virtuelle dans Odoo et le lier à la ligne de commande d'origine.</li> </ul>
US-29	<ul style="list-style-type: none"> <li>- Développer la méthode de synchronisation sur notre modèle de machines virtuelles pour interroger Proxmox et mettre à jour les données dans Odoo.</li> <li>- Implémenter les appels à l'API Proxmox pour obtenir le statut, l'adresse IP, etc., des VMs</li> <li>- Mettre à jour les champs correspondants sur les enregistrements de machines virtuelles dans Odoo.</li> <li>- Créer une Action Serveur Odoo pour permettre l'appel manuel de cette méthode depuis la vue liste des VMs.</li> </ul>
US-30	<ul style="list-style-type: none"> <li>- Définir une tâche planifiée (CRON) dans un fichier de données XML pour configurer l'exécution périodique de la synchronisation.</li> </ul>

## 5.5. Conception Technique et Architecture de la Solution

Cette section détaille l'architecture logicielle mise en place pour répondre aux exigences fonctionnelles définies pour ce sprint et pour permettre la réalisation des tâches planifiées. Elle présente la structure du modèle de données qui soutient les fonctionnalités développées, ainsi que les interactions clés entre les différents composants du système, illustrées par des diagrammes UML.

### 5.5.1. Architecture du Modèle de Données (Diagramme de Classes)

L'implémentation des fonctionnalités du Sprint 1 a nécessité la création de nouveaux modèles au sein d'Odoo et l'extension de plusieurs modèles existants afin d'y intégrer les informations et la logique spécifiques à l'intégration Proxmox.

- **Nouveaux Modèles :**

- `proxmox.virtual.machine` : Modèle central pour enregistrer les informations des VMs déployées (VMID, nom, ressources, statut, lien client/commande).
- `proxmox.api.mixin` : Modèle abstrait (Mixin) regroupant la logique générique d'appel à l'API Proxmox, utilisé par d'autres modèles.

- **Modèles Étendus (par héritage) :**

- `res.partner` : Enrichi pour stocker les identifiants Proxmox du client (`pve_user_id`, `pve_pool_id`) et pour orchestrer les principaux processus de provisionnement client.
- `product.template` : Ajout d'un champ pour identifier les produits éligibles à l'automatisation Proxmox (`pve_is_vm_product`).
- `sale.order` et `sale.order.line` : Étendus pour suivre l'état du déploiement des VMs liées aux commandes.
- `account.move` et `account.move.line` : Modifiés pour servir de déclencheur à l'automatisation basée sur le paiement de la facture.
- `res.config.settings` : Étendu pour fournir l'interface d'administration centralisée de l'intégration.

Le diagramme de classes ci-dessous Figure 42 illustre l'architecture de ces modèles et leurs relations dans le périmètre de ce sprint. Il met en évidence les nouveaux modèles créés ainsi que les principaux modèles Odoo standards qui ont été étendus pour notre intégration.

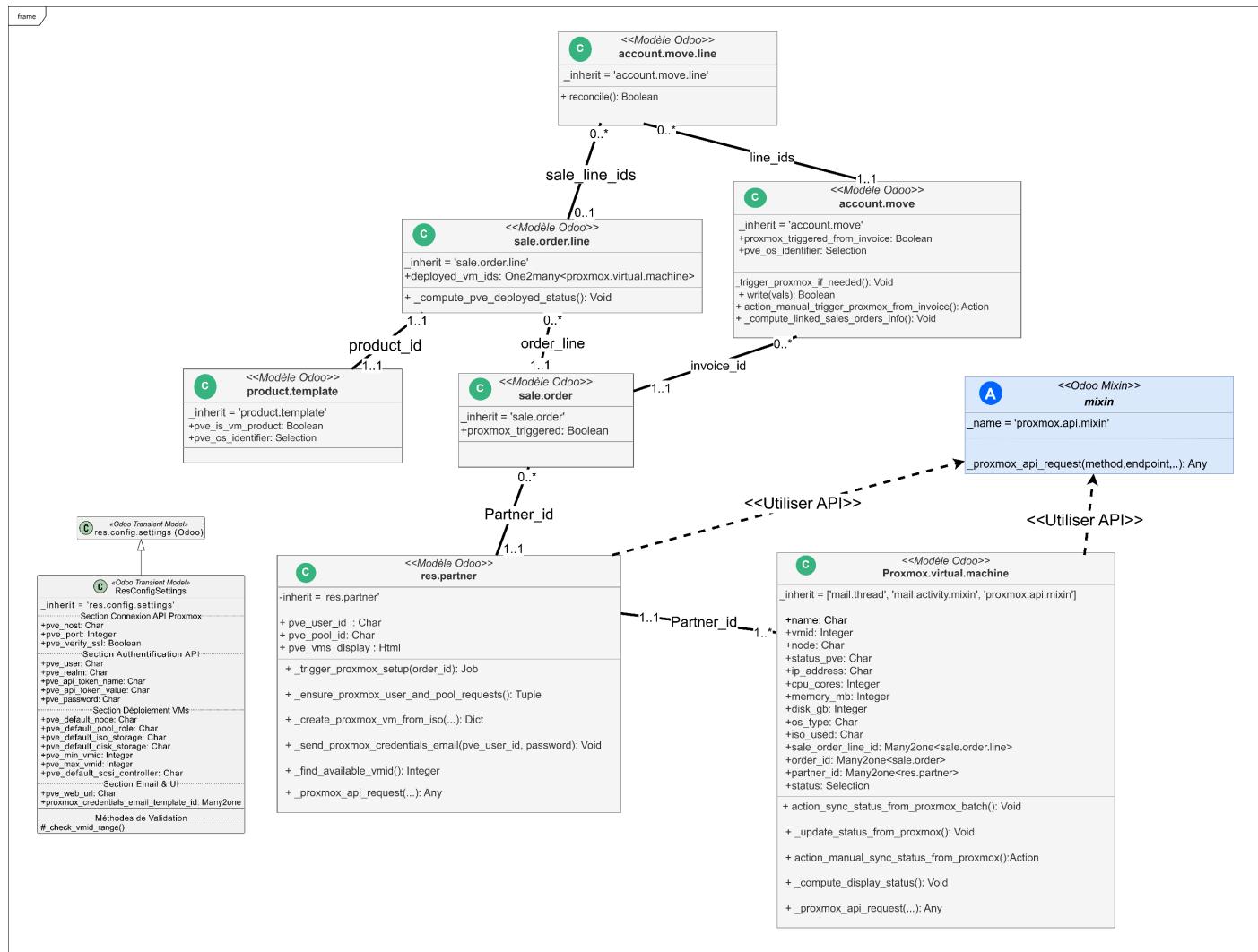
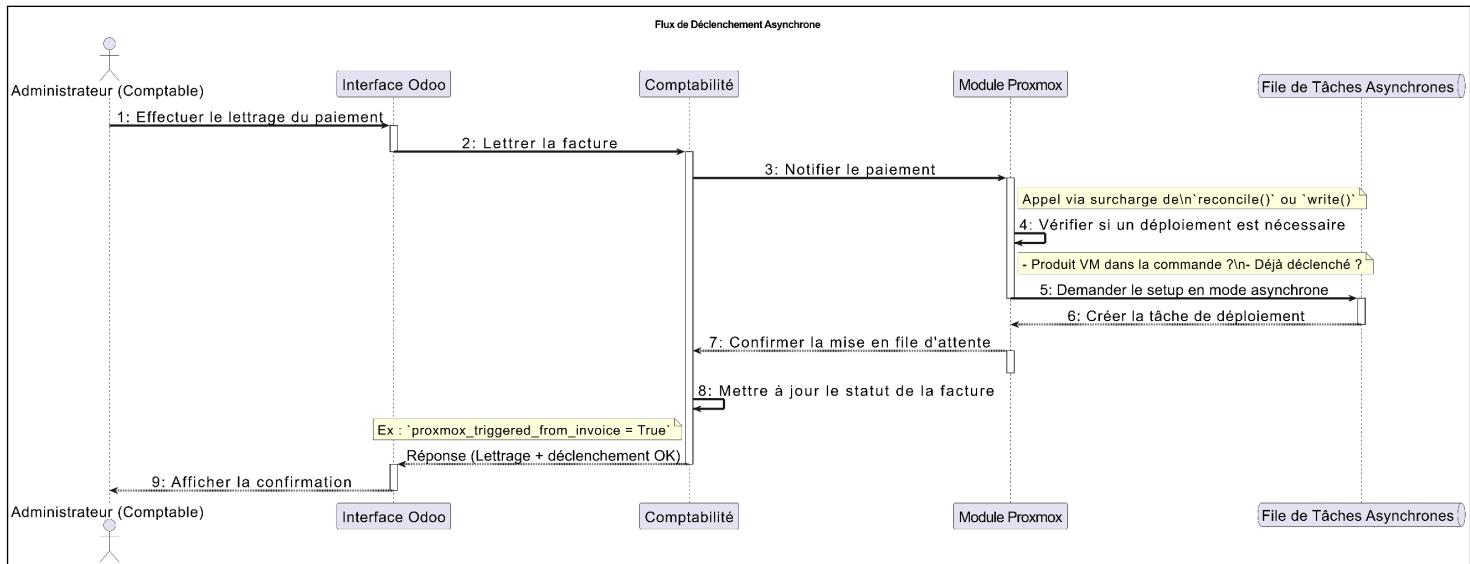


Figure 42 : Diagramme de classes dans le périmètre du sprint 1

### 5.5.2. Architecture Comportementale (Diagrammes de Séquence)

Au-delà de la structure des données, il est essentiel de comprendre la dynamique des interactions entre les différents composants du système. Les diagrammes de séquence suivants illustrent les processus clés du sprint, en se concentrant sur les mécanismes d'automatisation spécifiques à notre module.

**Diagramme 1 : Le Déclencheur**



**Figure 43 : Séquence 1 - Déclenchement Automatisé par le Paiement**

#### Description (Figure 43) :

Ce diagramme illustre comment un événement métier standard (le lettrage d'une facture par le comptable) initie notre flux d'automatisation. Notre module, via une surcharge sur le modèle de comptabilité, intercepte cet événement (étape 3), vérifie si un déploiement est nécessaire (étape 4), puis utilise le système de file d'attente asynchrone d'Odoo pour déléguer la tâche de déploiement (étapes 5 et 6), évitant ainsi de bloquer l'interface utilisateur.

## Diagramme 2 : Provisionnement du Compte Client

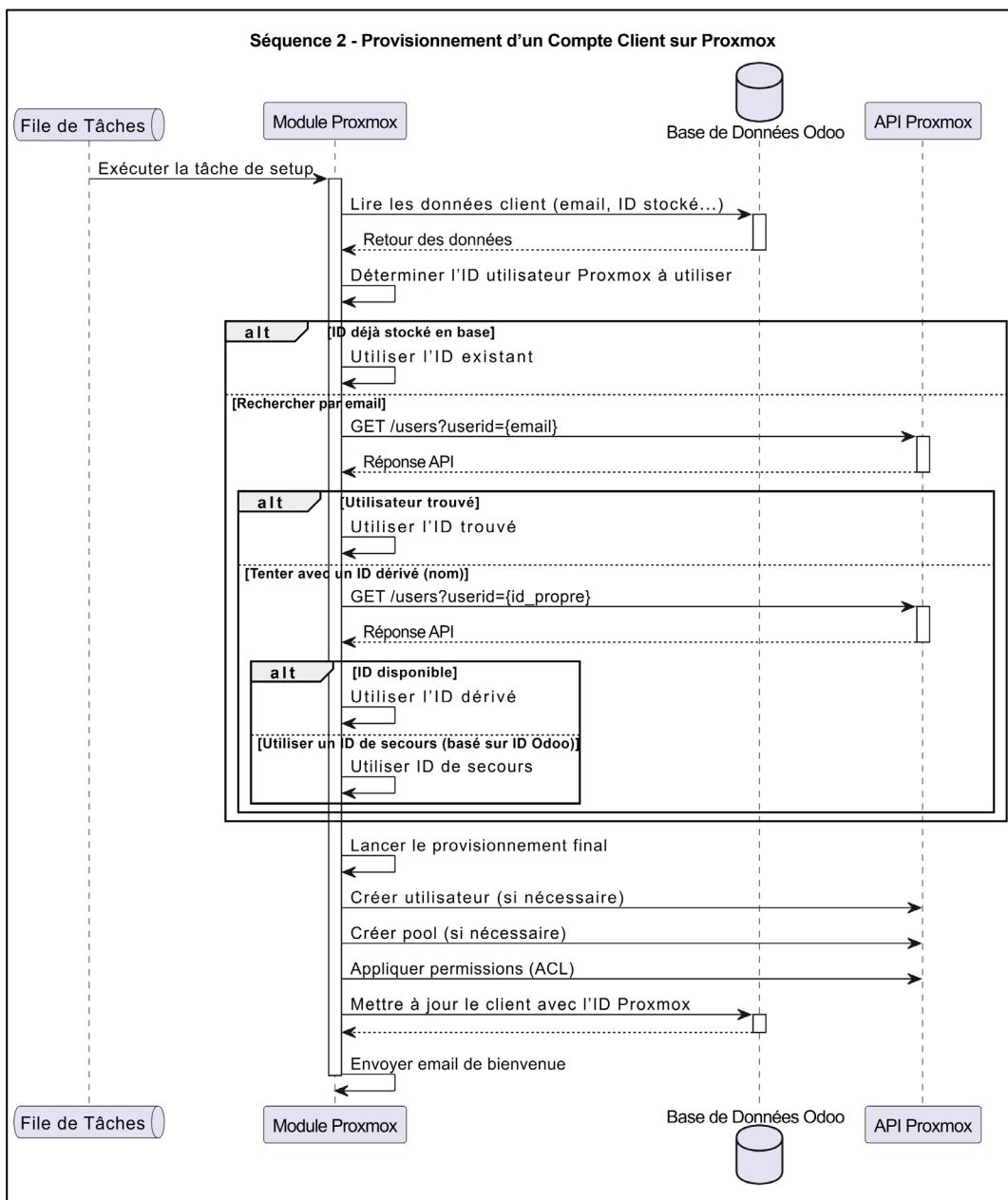


Figure 44 : Séquence 2 - Provisionnement d'un compte client sur Proxmox

### Description (Figure 44) :

Cette séquence détaille la logique de provisionnement du compte exécutée par la tâche asynchrone. Pour garantir la robustesse du système, le module ne se contente pas de créer un utilisateur ; il met en œuvre une stratégie de recherche séquentielle pour déterminer l'identifiant Proxmox à utiliser. Il vérifie successivement un ID déjà stocké, un utilisateur existant avec le même email, un ID dérivé du nom, et enfin un ID de secours unique. Ce n'est qu'après avoir validé un identifiant non-conflictuel que le module procède à la création des ressources sur l'API Proxmox, met à jour la base de données Odoo, et envoie l'email de bienvenue.

### Diagramme 3 : Déploiement de la Machine Virtuelle

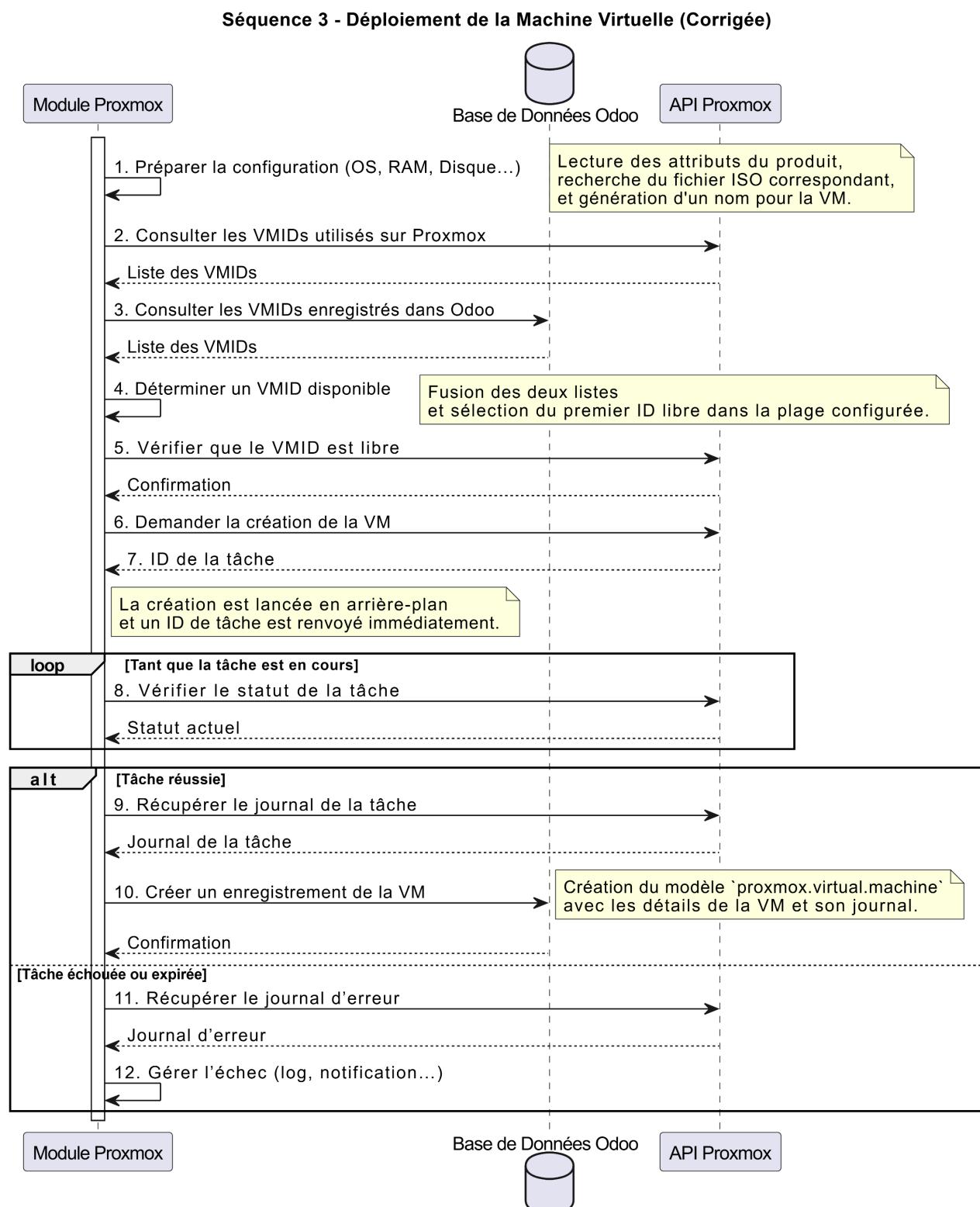


Figure 45 : Séquence 3 - Déploiement de la Machine Virtuelle

### Description (Figure 45) :

Après le provisionnement du compte client, le module orchestre le déploiement de la machine virtuelle. La séquence débute par l'identification fiable d'un VMID disponible, en interrogeant à la fois l'API Proxmox et la base de données Odoo.

Une fois la configuration de la VM préparée (spécifications, nom, etc.), la requête de création est envoyée. Le point clé est la gestion de la réponse asynchrone de Proxmox : le module récupère un ID de tâche (UPID) et sonde son statut en boucle jusqu'à sa complétion. Ce n'est qu'après confirmation du succès que la nouvelle VM, avec son journal de création, est enregistrée de façon permanente dans la base de données Odoo.

## 5.6. Réalisation

Cette section présente les résultats concrets du développement réalisé durant le Sprint 1. Elle suit le parcours logique de l'information à travers le système, de la configuration initiale par l'administrateur jusqu'à la supervision de la machine virtuelle déployée, en illustrant chaque fonctionnalité clé par des captures d'écran de l'application.

### 5.6.1. Étape 1 : Configuration par l'Administrateur (US-07, US-08)

La première étape pour rendre le système opérationnel est sa configuration par l'administrateur. Pour répondre à l'**US-07**, une section dédiée "**Intégration Proxmox**" a été ajoutée dans les Paramètres Généraux d'Odoo (voir Figure 46 et 47). Elle centralise toutes les options de connexion à l'API, les stockages par défaut, les plages de VMID, ainsi que les paramètres nécessaires au déploiement des machines virtuelles, tels que le type de BIOS ou le processeur par défaut.

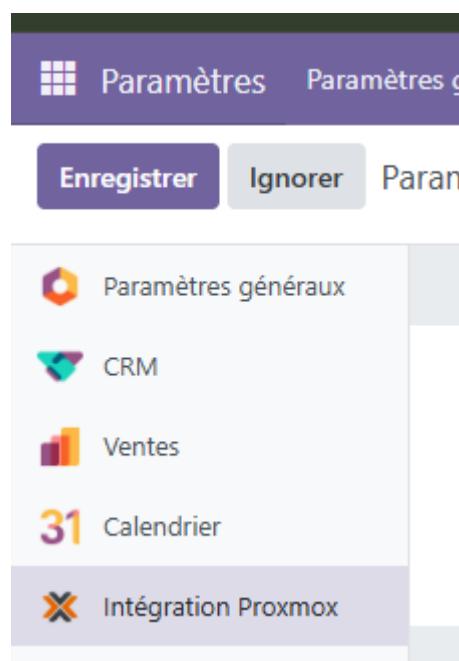


Figure 46 : Section Intégration Proxmox

**Connexion à l'API Proxmox**

<b>Hôte et Port de l'API</b> Nom d'hôte/IP et port (généralement 8006) de votre serveur Proxmox.  192.168.162.129      : 8 006	<input type="checkbox"/> <b>Vérifier le certificat SSL</b> Fortement recommandé pour la production. Nécessite un certificat SSL valide.
---	--

**Authentification à l'API**

<b>Utilisateur et Domaine API</b> Utilisateur Proxmox API et son domaine (ex: odoo-api@pve).  odoo                @ pve	<b>Jeton API (Recommandé)</b> Nom et valeur secrète du jeton API Proxmox. <b>Nom du Jeton API Proxmox</b> api-odoo
--	--

<b>Mot de passe API (Alternative)</b> Utiliser si aucun jeton API n'est fourni.	<b>Secret du Jeton</b> .....  .....
--	---

**Paramètres de Déploiement Généraux (Suite)**

<b>Nœud Proxmox par défaut</b> Nœud Proxmox par défaut pour les déploiements.  hostexample	<b>Rôle par défaut pour Pool Client</b> Rôle Proxmox du client sur son pool de ressources.  ClientOdoo
---	---

<b>Groupe Proxmox par défaut pour utilisateurs</b> Groupe Proxmox par défaut pour les nouveaux utilisateurs.  Clients_Odoo	<b>Stockage ISO par défaut</b> Stockage Proxmox par défaut pour les images ISO.  local
---	---

<b>Stockage Disque par défaut pour VMs</b> Stockage Proxmox par défaut pour les disques VM.  local-lvm	<b>Contrôleur SCSI par défaut pour VMs</b> Contrôleur SCSI par défaut pour les nouvelles VM.  virtio-scsi-single
---	---

<b>Type de CPU par défaut pour VMs</b> Type de processeur par défaut pour les nouvelles VM.  x86-64-v2-AES	<b>Type de BIOS par défaut pour VMs</b> BIOS par défaut pour les nouvelles VM.  <input checked="" type="radio"/> SeaBIOS (Hérité) <input type="radio"/> OVMF (UEFI)
---	---

<b>Stockage Disque EFI par défaut (si OVMF)</b> Stockage EFI (si BIOS OVMF). Si vide, utilise le stockage disque.  local-lvm	<b>Stockage TPM par défaut (si TPM activé)</b> Stockage TPM (si TPM activé). Si vide, utilise le stockage disque.  local-lvm
---	---

<b>Plage de VMID</b> Ex: 200-9999. 200                - 9 999	
---	--

**Modèles d'Email et Interface Utilisateur**

<b>URL Web Proxmox (pour emails)</b> URL de connexion Proxmox affichée dans les emails.  https://192.168.1.99:8006/	<b>Modèle Email: Identifiants Proxmox</b> Modèle pour envoyer les identifiants Proxmox.  Proxmox: User Credential: ▾ →
--	---

Figure 47 : Interface de Configuration de l'Intégration Proxmox

Pour l'**US-08**, nous avons permis à l'administrateur de "marquer" un produit comme étant une VM Proxmox. Un simple champ booléen, a été ajouté à la fiche produit, comme l'illustre la Figure 5.8. Seuls les produits avec cette case cochée déclencheront l'automatisation.

Figure 48 : Fiche Produit Configurée pour le Déploiement VM

### 5.6.2. Étape 2 : Le Parcours d'Achat Client (US-01)

Le parcours client (**US-01**) commence sur la boutique en ligne (Figure 5.9), où il personnalise sa VM et passe commande. Une fois le processus de vente standard d'Odoo terminé et la facture correspondante générée, notre module se prépare à prendre le relais dès la confirmation du paiement.

Figure 49 : Page Produit avec Sélection des Variantes de la VM

### 5.6.3. Étape 3 : Le Déclenchement Automatisé (US-13)

Le déploiement est automatiquement déclenché lors du paiement de la facture liée à la commande. Cet événement agit comme point de départ du processus. Pour aider l'administrateur à suivre et gérer ce déploiement, la vue facture a été améliorée.

Comme le montre la Figure 50 , des champs permettent de vérifier si le déploiement a été lancé. En plus, pour répondre à l'US-13, un bouton "Déclencher Configuration Proxmox" a été ajouté. Il permet à l'administrateur de lancer le processus manuellement en cas d'échec ou de désactivation du déclenchement automatique.

The screenshot shows a Proxmox client invoice interface. At the top, there are several buttons: Envoyer, Imprimer, Payer, Aperçu, Avoir, Remettre en brouillon, Déclencher Proxmox pour CV Liée(s), Brouillon, and Comptabilisé. Below these buttons, the invoice header reads "Facture client" and "FAC/2025/00002". The invoice details include:

- Client:** Firas Taboubi
- Date de facturation:** 08/06/2025
- Conditions de paiement:** Paiement immédiat
- Journal:** Factures clients

Under the heading "STATUT INTÉGRATION PROXMOX", it says "Commande(s) de Vente" followed by a radio button and a link "Liée(s) ?". Below that, it says "S00184: Déclenché" with a checked checkbox.

The main table displays the invoice lines:

Lignes de facture	Écritures comptables	Autres informations					
Produit	Compte	Analytique	Quantité	UdM	Prix	Taxes	Montant
VM Basique Linux (Ubuntu 24.04 LTS, ...)	707000 Ve...		1,00	Unité(s)	1,10		1,100 DT
<i>Déployez rapidement un serveur Linux minimaliste sur notre infrastructure Proxmox haute performance. Cette machine virtuelle vous fournit une base solide</i>							

Figure 50 : Suivi et Déclenchement Manuel depuis la Facture

### 5.6.4. Étape 4 : Provisionnement du Compte et Notification Client (US-18, US-03, US-02)

Une fois le déploiement déclenché, le système provisionne automatiquement le compte et le pool de ressources du client sur Proxmox, comme requis par l'US-18. La Figure 51 montre un exemple de compte utilisateur créé par notre module.

Name ↓	Comment
pool-firas-taboubi	Odoo Partner ID: 34 / Name: Firas Taboubi

Figure 51 : Preuve de la Crédit du Compte sur Proxmox

Pour assurer la traçabilité, les identifiants Proxmox générés (`pve_user_id`, `pve_pool_id`) sont immédiatement enregistrés sur la fiche du client dans Odoo (Figure 52).

IDENTIFIANTS PROXMOX	
ID Utilisateur Proxmox ?	firas-taboubi@pve
ID Pool Proxmox ?	pool-firas-taboubi

Figure 52 : Retour : Identifiants Proxmox enregistrés dans la fiche client Odoo

Enfin, un email de bienvenue est envoyé au client (US-03), lui fournissant ses identifiants et l'URL de connexion (Figure 53).

Cet email lui permet d'accéder directement au portail Proxmox pour gérer ses services (US-02), dont l'interface de connexion est illustrée en Figure 54

Vos accès à votre environnement Proxmox Magcloud ➔

 **Magcloud** <magcloud06@gmail.com>  
À firastaboubi88 ▾

Bonjour Firas Taboubi,

Suite à votre commande chez Magcloud, voici vos informations pour vous connecter à votre environnement Proxmox personnel :

**URL de connexion Proxmox :** <https://192.168.1.99:8006/>

**Identifiant (User name) :** firas-taboubi

**Mot de passe :** i\*ogpBupE5Z^

**Domaine (Realm) :** Sur l'écran de connexion, veuillez sélectionner "Proxmox VE authentication server" dans la liste déroulante.

Nous vous recommandons fortement de **changer ce mot de passe** lors de votre première connexion pour des raisons de sécurité.

Cordialement,  
L'équipe Magcloud

Ceci est un email automatique généré par notre système. Pour toute question ou assistance, veuillez contacter notre support technique.

Figure 53 : Email de bienvenue envoyé au client avec ses identifiants Proxmox

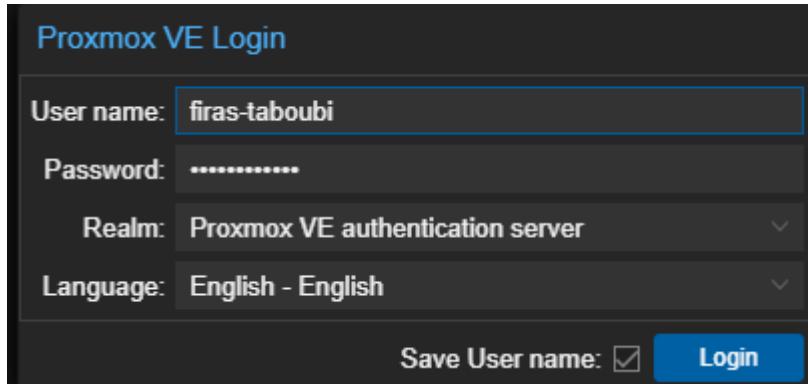


Figure 54 : Portail d'Accès Proxmox Présenté au Client

### 5.6.5. Étape 5 : Déploiement de la Machine Virtuelle (US-24)

Le processus se poursuit avec le déploiement effectif de la machine virtuelle, répondant ainsi à l'**US-24**. Le système utilise les spécifications choisies par le client pour créer la VM sur Proxmox à partir d'une image ISO, comme le montre la Figure 55.

Simultanément, un enregistrement représentant cette nouvelle VM est créé dans la base de données Odoo

Cet enregistrement centralise toutes les informations techniques et les liens vers le client et la commande d'origine, servant de base à toute la supervision.

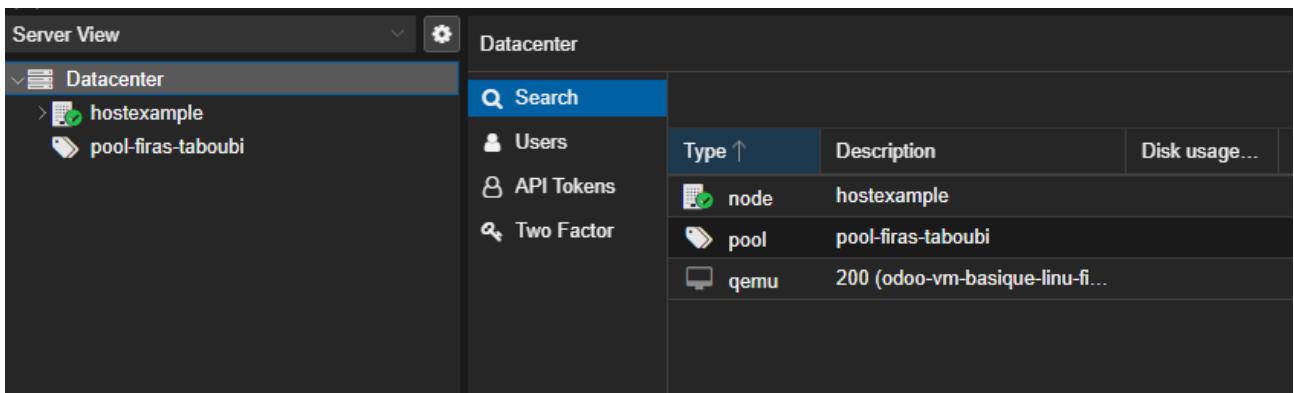


Figure 55 : Preuve du Déploiement de la VM sur l'Interface Proxmox

### 5.6.6. Étape 6 : Interfaces de Supervision (US-11, US-12)

Pour permettre à l'administrateur de suivre le parc de machines, deux interfaces de supervision clés ont été mises en place.

Pour répondre à l'US-12, une vue centralisée listant toutes les VMs déployées a été créée (Figure 56). Elle offre une vision d'ensemble rapide du parc.

VMs Proxmox Déployées									Rechercher...	1-3 / 3	Actions
	Nom de la ...	ID VM Pr...	Client	Commande	Produit	Nœud Pro...	Réseau Cli...	Statut	Adresse IP	Date de Commande	Actions
<input type="checkbox"/>	vm-basique...	202	Firas Tabou...	S00184	VM Basique Linux (Ubuntu 24.04 LTS, 2 vCPU, 2 Cœ...	hostexample		Arrêtée		08/06/2025 19:51:15	
<input type="checkbox"/>	vm-basique...	201	Firas Tabou...	S00183	VM Basique Linux (Ubuntu 24.04 LTS, 1 vCPU, 1 Cœ...	hostexample		Arrêtée		08/06/2025 19:25:38	
<input type="checkbox"/>	vm-basique...	200	Firas Tabou...	S00183	VM Basique Linux (Ubuntu 24.04 LTS, 1 vCPU, 1 Cœ...	hostexample		Arrêtée		08/06/2025 19:25:38	

Figure 56 : Vue Centralisée du Parc de Machines Virtuelles dans Odoo.

Récapitulatif de ses machines virtuelles, offrant une vue à 360° sans quitter l'interface du client (Figure 57).

DEPLOYED VMS INFORMATION		
Order	Product	VMID
S00145	VM Basique Linux (Rocky Linux 9.5, 2 vCPU, 1 Cœur, 2 Go RAM, 10 Go)	200

Figure 57 : Tableau Récapitulatif des VMs sur la Fiche Client.

### 5.6.7. Fonction Transverse : Fiabilisation par la Synchronisation

Enfin, pour garantir la fiabilité des données affichées, un mécanisme de synchronisation a été implémenté. Pour répondre à l'US-29, l'administrateur dispose d'une action manuelle pour rafraîchir le statut des VMs à la demande (Figure 58).

Nom de la ...	ID VM Pr...	Client	Commande	Produit	Nœud Pro...	Réseau Cli...	Statut	Adresse IP	Date de Commande
vm-basique...	202	Firas Tabou...	S00184	VM Basique Linux (Ubuntu 24.04 LTS, 2 vCPU, 2 Cœ...	hostexample		Arrêtée		08/06/2025 19:51:15
vm-basique...	201	Firas Tabou...	S00183	VM Basique Linux (Ubuntu 24.04 LTS, 1 vCPU, 1 Cœ...	hostexample		Arrêtée		08/06/2025 19:25:38
vm-basique...	200	Firas Tabou...	S00183	VM Basique Linux (Ubuntu 24.04 LTS, 1 vCPU, 1 Cœ...	hostexample		Arrêtée		08/06/2025 19:25:38

Figure 58 : Action Manuelle de Synchronisation du Statut des VMs

En parallèle, pour l'US-30, une tâche planifiée (CRON) assure une mise à jour périodique et automatique de l'ensemble du parc, garantissant que les informations dans Odoo restent à jour sans intervention manuelle (Figure 59).

**[Proxmox] Sync VM Status**

**PARAMÈTRES TECHNIQUES**

Modèle ? Machine Virtuelle Déployée sur Proxmox Disponible sur le site web ?

Groupes autorisés ?

**DÉTAILS DE L'ACTION**

Utilisateur de l'outil de planification Administrator →

Exécuter tou(te)s les ? 5 Minutes

Actif

Prochaine date d'exécution ? 13/06/2025 14:55:09

Priorité ? 5

Code Assistance

```
1 model.action_sync_status_from_proxmox_batch()
```

Figure 59 : Configuration de la Tâche Planifiée (CRON) pour la Synchronisation Automatique.

## 5.7. Bilan et Rétrospective du Sprint (Sprint Review)

À l'issue de chaque cycle de développement, la méthodologie Agile préconise une phase de bilan (Sprint Review & Retrospective) afin d'évaluer le travail accompli, de recueillir les retours et d'identifier les axes d'amélioration pour les sprints futurs.

### 5.7.1. Atteinte de l'Objectif et Synthèse des Acquis

L'objectif principal de ce premier sprint, qui était de "valider un parcours client complet, de la commande à la réception d'une VM simple, et de fournir à l'administrateur les interfaces de configuration et de supervision fiables et à jour", a été atteint avec succès.

- La démonstration des fonctionnalités a permis de valider que :
- Un client peut configurer et commander une VM depuis la boutique en ligne.
- Le paiement d'une facture déclenche bien le processus d'automatisation.
- Le système provisionne correctement un compte utilisateur, un pool et une machine virtuelle sur Proxmox.
- Les informations critiques sont bien enregistrées et synchronisées dans Odoo.
- L'administrateur dispose des vues nécessaires pour configurer le module et superviser les déploiements.

Ce premier incrément fonctionnel (MVP) a permis de construire une base technique et métier solide, validant ainsi la viabilité du projet et des choix technologiques effectués.

### 5.7.2. Leçons Apprises et Décisions d'Adaptation

Ce premier sprint, bien que réussi, a été riche en enseignements. Il a permis de mettre en lumière plusieurs points cruciaux qui ont guidé les décisions pour la suite du projet.

#### Leçon n°1 : La gestion de la "Personnalisation" (US-01)

La configuration dynamique complète était trop complexe. Solution : utiliser les variantes de produits Odoo pour un MVP plus simple et rapide à livrer.

#### Leçon n°2 : Les Limites du Déploiement par Image ISO (US-24)

Permet un MVP technique, mais l'expérience utilisateur est limitée. Solution prévue : US-22 (déploiement via clonage de template) au Sprint 3.

#### Leçon n°3 : L'Impératif de la Sécurité Réseau.

L'usage d'un bridge réseau public n'assure pas l'isolation des clients. Solution : Sprint 2 dédié à l'architecture réseau sécurisée et isolée.

## 5.8. Conclusion

Ce premier sprint a constitué une étape fondamentale dans la réalisation de notre projet. En partant d'un socle technique préparé, nous avons réussi à construire et à valider un premier incrément de produit fonctionnel (MVP). Le flux principal, de la commande client sur l'interface Odoo jusqu'au déploiement automatisé d'une machine virtuelle sur Proxmox, est désormais opérationnel.

Les choix d'architecture, notamment l'utilisation d'une file d'attente asynchrone pour les tâches longues et la mise en place d'un mécanisme de synchronisation des données, ont permis de poser des fondations robustes et fiables.

Cependant, le bilan de ce sprint a également mis en lumière les limitations stratégiques de ce MVP, en particulier l'absence d'isolation réseau entre les clients. Forts de ces acquis et de ces apprentissages, nous sommes désormais prêts à aborder le défi suivant. Le chapitre qui suit détaillera le développement du Sprint 2, qui sera entièrement consacré à la conception et à l'implémentation de l'architecture réseau sécurisée, une étape cruciale pour transformer notre solution en un service viable en production.

# Chapitre 6 Sprint 2 : Isolation Client et Gestion Réseau

## 6.1. Introduction : De la Preuve de Concept à la Sécurisation

Le premier sprint a validé avec succès le flux de déploiement automatisé. Cependant, il présentait une limitation majeure : l'absence d'isolation réseau entre les clients, une configuration inacceptable en production.

Ce chapitre détaille le développement du Sprint 2, dont l'objectif est de corriger cette faille critique. Nous nous concentrerons sur la mise en place de la gestion logique des réseaux clients dédiés et de l'accès sécurisé, afin de transformer notre solution en une architecture multi-locataire (multi-tenant) robuste et prête pour les prochaines étapes de l'automatisation.

## 6.2. Objectif du Sprint (Sprint Goal)

L'objectif formel de ce sprint est d'automatiser la gestion logique du réseau client dans Odoo et de formaliser le processus de mise en place de l'infrastructure de connectivité (bridge, DHCP, NAT) et d'accès sécurisé (bastion SSH), qui restent, à ce stade, des opérations manuelles pour l'administrateur.

Concrètement, cela signifie que notre module Odoo doit être capable de :

- Définir et gérer des pools de plages d'adresses IP (super-réseaux).
- Allouer intelligemment un sous-réseau privé unique à chaque nouveau client.
- Déployer les nouvelles machines virtuelles dans leur réseau isolé respectif.
- Fournir à l'administrateur les outils pour superviser ces allocations.
- Informer le client des détails de sa nouvelle connectivité privée.

## 6.3. Analyse Fonctionnelle

Cette section définit les besoins fonctionnels spécifiques au Sprint 2, qui se concentrent sur la gestion différenciée des réseaux et l'évolution des interactions des utilisateurs avec le système.

### 6.3.1. Identification des Acteurs et de leurs Nouveaux Rôles

Les besoins de ce sprint se concentrent sur la gestion réseau et la sécurité :

- **Pour l'Administrateur :** Son rôle s'enrichit. Il n'est plus seulement un superviseur, mais un stratège réseau. Il doit pouvoir définir les pools d'adresses IP, superviser leur allocation, et surtout, il est responsable de la mise en place manuelle de l'infrastructure physique correspondante (bridges, DHCP, NAT, bastion).

- **Pour le Client :** Son expérience est enrichie grâce à une meilleure visibilité sur son réseau privé. Un client existant peut désormais regrouper ses services sur un réseau déjà attribué, gagnant ainsi en clarté et en flexibilité.

### 6.3.2. La Stratégie d'Allocation Différenciée

Conformément à la conception réseau définie au chapitre 3, notre module applique une stratégie d'allocation IP multi-niveaux pour assurer l'isolation et la sécurité.

Lors de la première commande, le module détecte le type de client et attribue automatiquement le premier sous-réseau /24 libre dans le super-réseau correspondant. Chaque client bénéficie ainsi de son propre sous-réseau isolé, préparant le terrain pour des politiques futures (sécurité, QoS...)

### 6.3.3. Évolution des Cas d'Utilisation

Les nouvelles fonctionnalités de ce sprint se traduisent par de nouveaux cas d'utilisation pour nos acteurs, comme le montrent les diagrammes suivants.

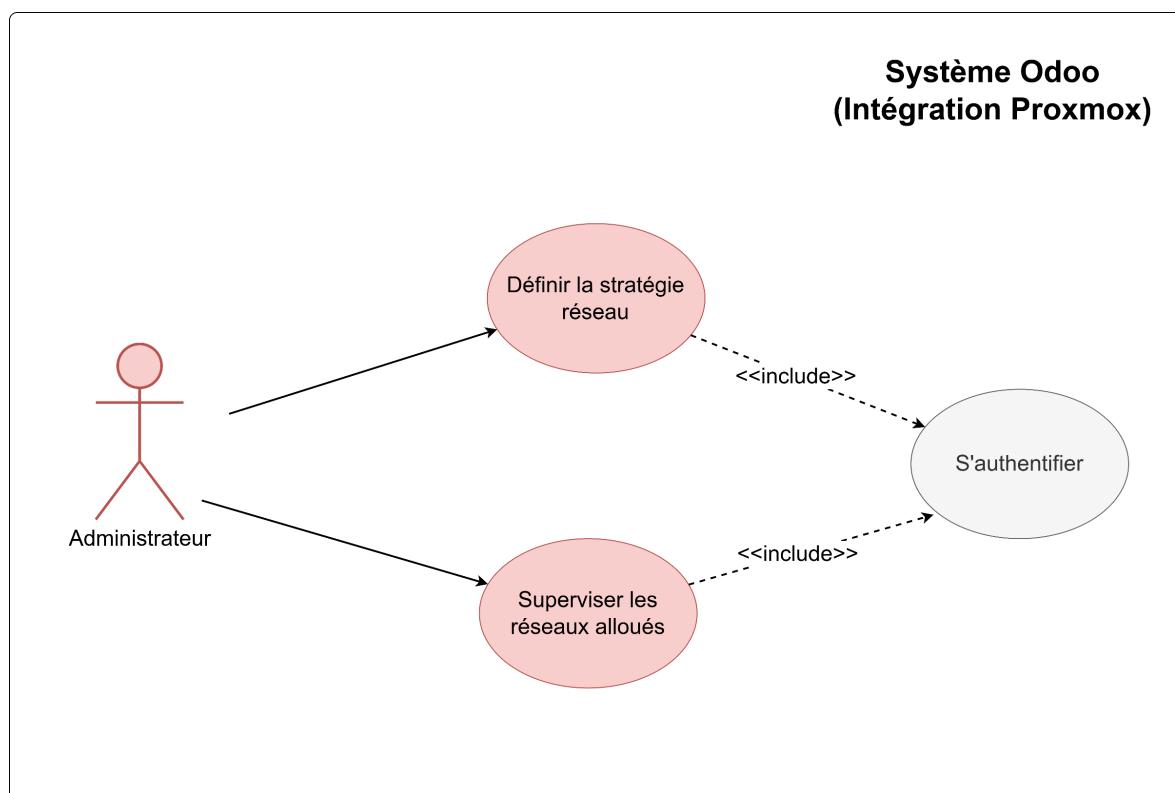


Figure 60 : Diagramme des Cas d'Utilisation - Rôle Administrateur (Sprint 2)

### 6.3.4. Spécifications Détaillées des Nouveaux Cas d'Utilisation

Pour formaliser le comportement attendu, les nouveaux cas d'utilisation de l'administrateur sont décrits ci-dessous.

Tableau 24 : Spécification du Cas d'Utilisation 'Définir la stratégie réseau'

Section	Description
Nom	Définir la stratégie réseau
Acteur	Administrateur
Objectif	Permettre à l'administrateur de configurer les super-réseaux (pools d'IP) qui seront utilisés par le système pour allouer automatiquement les sous-réseaux aux clients.
Précondition	L'administrateur est connecté au back-end Odoo et a accès aux paramètres de l'intégration Proxmox.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'administrateur navigue vers la page de configuration de l'intégration Proxmox.</li> <li>2. Il saisit ou modifie les blocs CIDR des super-réseaux "Standard" et "Entreprise".</li> <li>3. Il active l'option pour traiter les "Sociétés" Odoo comme des clients "Entreprise".</li> <li>4. Il enregistre la configuration.</li> </ol>
Scénarios d'Exception	<p>A1 : Format invalide – Si le super-réseau saisi est mal formé (ex. sans masque CIDR ou adresse incorrecte), l'enregistrement est bloqué avec un message d'erreur.</p> <p>A2 : Chevauchement détecté – Si deux super-réseaux configurés se recoupent, le système refuse l'enregistrement pour éviter des conflits IP.</p>
Postcondition	La stratégie d'allocation réseau du système est mise à jour.

**Tableau 25 : Spécification du Cas d'Utilisation 'Superviser les réseaux alloués'**

Section	Description
Nom	Superviser les réseaux alloués
Acteur	Administrateur
Objectif	Fournir à l'administrateur une vue centralisée de tous les sous-réseaux qui ont été alloués aux clients, avec leurs détails.
Précondition	L'administrateur est connecté au back-end Odoo et au moins un réseau a été alloué à un client.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'administrateur navigue vers le menu "Proxmox" &gt; "Réseaux Clients".</li> <li>2. Le système affiche la liste de tous les réseaux alloués, avec pour chaque réseau : le client associé, le sous-réseau (CIDR), le nom du bridge et le nombre de VMs l'utilisant.</li> </ol>
Scénarios d'Exception	<b>A1: Aucun réseau alloué :</b> Si aucun réseau n'a encore été alloué, le système affiche une liste vide avec un message informatif (ex : "Aucun réseau client trouvé.").
Postcondition	La stratégie d'allocation réseau du système est mise à jour.

## 6.4. Planification du Sprint

### 6.4.1. Périmètre du Sprint (Sprint Backlog)

Le backlog de ce sprint a été constitué pour répondre à ces besoins d'isolation et de gestion réseau.

**Tableau 26 : User Stories du Sprint 2**

ID	Epic	User Story
US-04	Expérience Client	En tant que Client, je veux recevoir un mail d'information sur mon réseau privé afin de connaître les détails de ma connectivité.
US-05	Expérience Client	En tant que Client existant, je veux être consulté par mail pour le choix du réseau afin de pouvoir regrouper ou isoler mes services.
US-10	Administration Odoo	En tant qu'Admin, je veux définir la stratégie d'allocation réseau afin de contrôler l'utilisation des plages d'IP.

US-11	Administration Odoo	En tant qu'Admin, je veux une vue listant tous les réseaux clients alloués afin de superviser l'occupation des sous-réseaux.
US-15	Réseau et Sécurité	En tant qu'Admin, je veux que chaque client soit dans un réseau privé et isolé afin de garantir la confidentialité et la sécurité.
US-16	Réseau et Sécurité	En tant qu'Admin, je souhaite que les clients reçoivent automatiquement une adresse IP via le service DHCP.
US-17	Réseau et Sécurité	En tant qu'Admin, je veux que l'accès SSH externe passe par un bastion sécurisé afin de centraliser et contrôler les connexions.
US-19	Automatisation	En tant que Système, je veux allouer automatiquement un sous-réseau privé logique afin d'appliquer la stratégie réseau de l'admin.

#### 6.4.2. Plan d'Action

Chaque User Story a été décomposée en tâches techniques concrètes pour organiser le développement.

ID User Story	Tâches Techniques à Réaliser
US-10	- Ajouter les champs de configuration des "super-réseaux" dans les paramètres système d'Odoo.
US-19	- Développer la logique de recherche du prochain sous-réseau disponible dans les super-réseaux configurés. - Créer un enregistrement dans le nouveau modèle de réseau client pour chaque nouvelle allocation.
US-15	- Modifier la logique de déploiement de VM pour qu'elle utilise le nom du bridge privé alloué au client, au lieu du bridge public.
US-16 & 17	- Rédiger et valider la documentation technique décrivant la procédure manuelle que l'administrateur doit suivre (configuration du bridge, DHCP, NAT, bastion).

US-11	<ul style="list-style-type: none"> <li>- Créer un modèle Odoo pour stocker les détails de chaque réseau alloué.</li> <li>- Créer une vue liste pour afficher tous les réseaux alloués, avec leur client et leurs détails (CIDR, passerelle).</li> </ul>
US-04	<ul style="list-style-type: none"> <li>- Créer un nouveau Template d'email pour informer le client des détails de son réseau privé (sous-réseau, passerelle, etc.).</li> <li>- Intégrer l'envoi de cet email dans le flux de provisionnement d'un nouveau client.</li> </ul>
US-05	<ul style="list-style-type: none"> <li>- Créer un Template d'email pour demander au client existant sur quel réseau il souhaite déployer sa nouvelle VM.</li> <li>- Adapter le flux de déploiement pour gérer la réponse du client ou une logique par défaut.</li> </ul>

## 6.5. Conception Technique : L'Architecture de la Connectivité Client

La mise en œuvre des fonctionnalités de ce sprint repose sur une architecture claire qui sépare la logique de décision (dans Odoo) de l'action physique (sur Proxmox VE). Cette section détaille le processus manuel de l'administrateur, l'évolution du modèle de données, et les séquences d'interactions logicielles.

### 6.5.1. Le Processus Manuel de l'Administrateur :

Le diagramme d'activité suivant (Figure 6.1) modélise le processus manuel que l'administrateur doit suivre pour rendre un nouveau réseau client, alloué logiquement par Odoo, pleinement fonctionnel sur l'infrastructure physique.

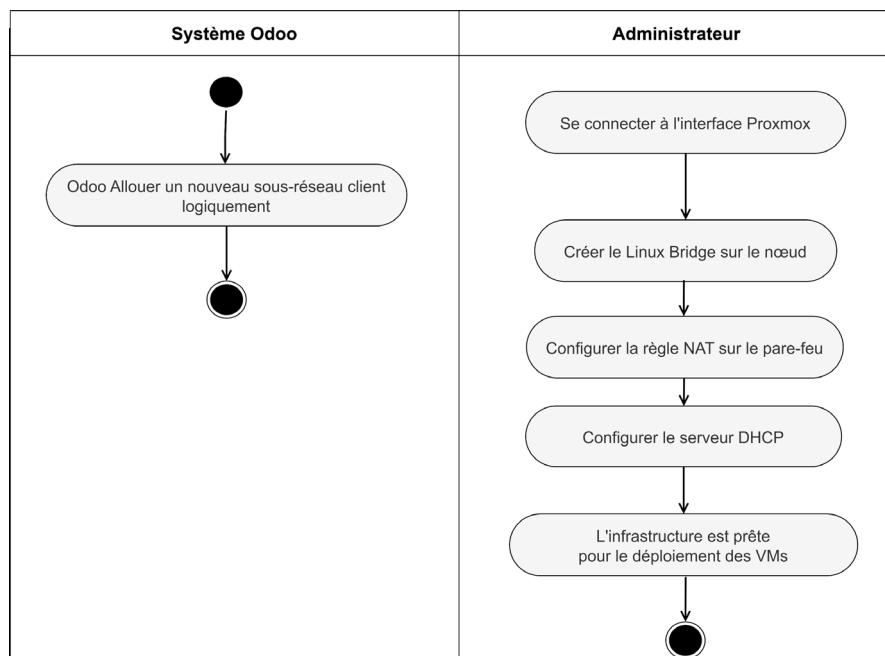


Figure 61 : Diagramme d'Activité - Configuration Manuelle du Réseau et de la Sécurité

### 6.5.2. Évolution du Modèle de Données

Pour supporter la gestion des réseaux, le modèle de données a évolué. Le modèle proxmox.client.network devient une entité centrale, et les modèles res.partner et res.config.settings sont enrichis de nouveaux champs et de nouvelles logiques pour définir et appliquer la stratégie réseau. Le diagramme de classes ci-dessous (Figure 6.2) met en évidence ces changements.

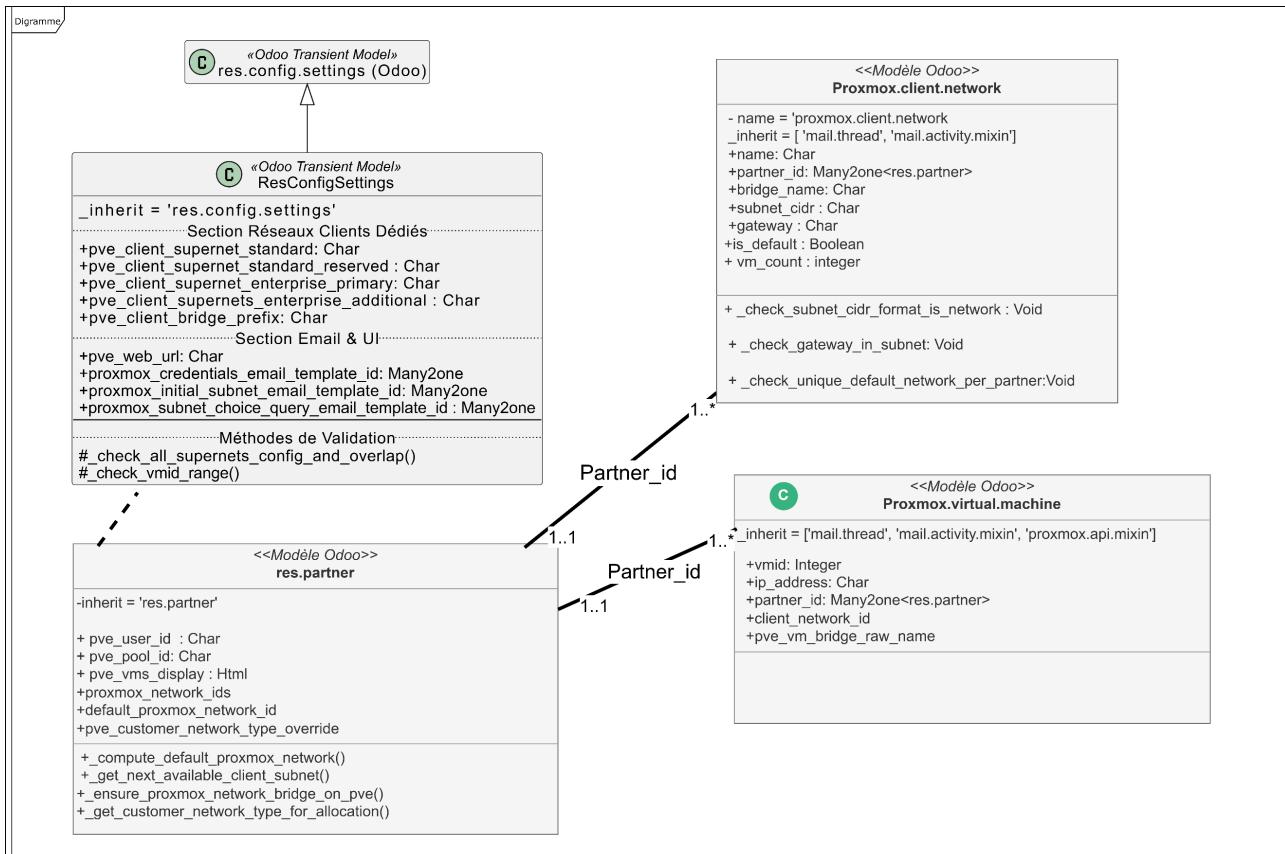


Figure 62 : Évolution du Diagramme de Classes (Focus Réseau)

**Description (Figure 62) :** Ce diagramme illustre l'architecture de la gestion réseau. On y observe que :

- Le modèle res.config.settings permet à l'administrateur de définir les super-réseaux qui serviront de pools d'adresses.
- Le modèle res.partner contient la logique d'allocation (`_get_next_available_subnet`) et est lié à un ou plusieurs réseaux clients.
- Le modèle proxmox.client.network est l'entité centrale qui stocke les informations de chaque sous-réseau alloué (bridge, CIDR, passerelle) et est lié à un unique client.
- Enfin, le modèle proxmox.virtual.machine est directement lié à un enregistrement proxmox.client.network, assurant que chaque VM est bien associée à son réseau isolé.

### 6.5.3. Architecture Comportementale : Le Processus d'Allocation Logique

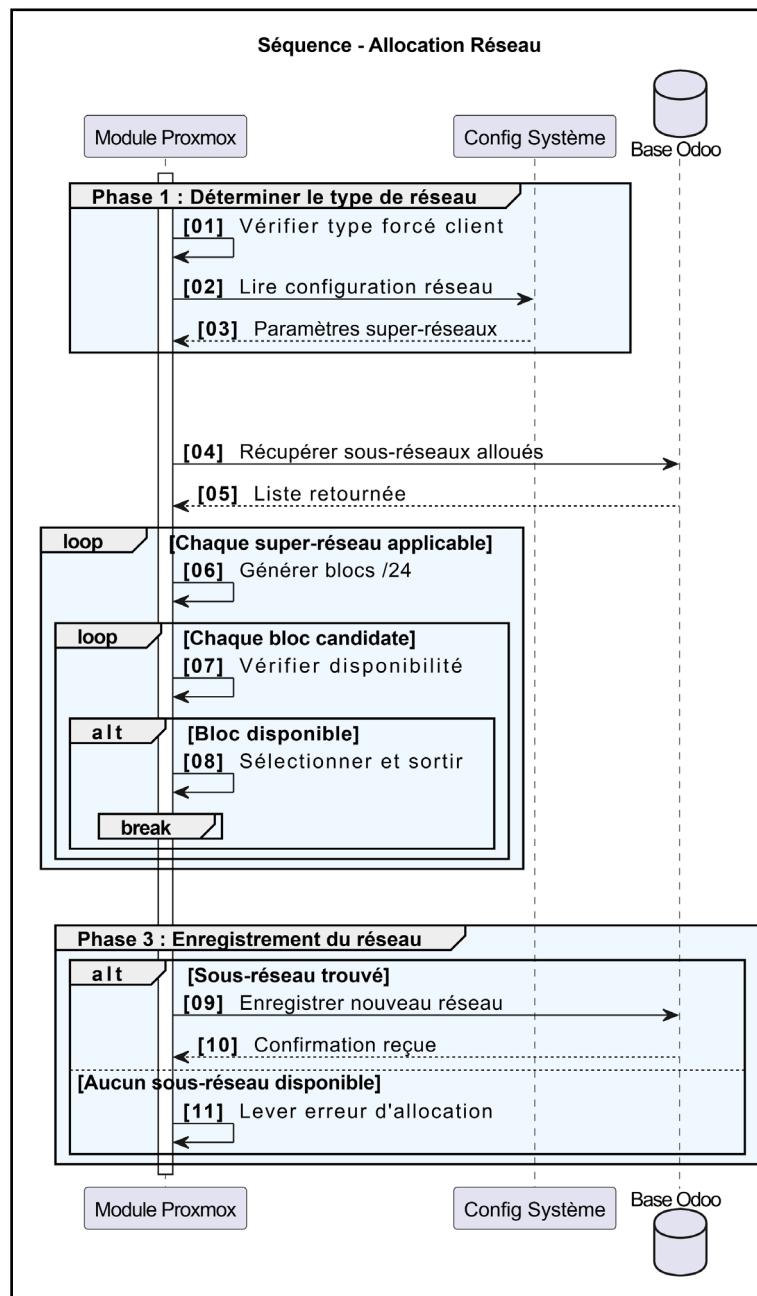


Figure 63 : Séquence d'Allocation d'un Nouveau Réseau Client.

#### Description

Ce diagramme détaille le processus intelligent d'allocation d'un nouveau réseau client. Le flux se décompose en plusieurs phases distinctes :

- Détermination du Type** : Avant toute chose, le module détermine le type de réseau à allouer (standard ou entreprise). Cette décision cruciale se base sur une hiérarchie de règles : elle privilégie un forçage manuel sur la fiche client, et sinon, elle combine le statut du client ('Société' ou 'Particulier') avec les paramètres globaux lus depuis la Configuration Système.

- Collecte et Calcul** : Une fois le type de réseau connu, le module collecte les informations pertinentes : il lit les super-réseaux correspondants depuis la Configuration, puis interroge la Base de Données Odoo pour la liste des sous-réseaux déjà utilisés. Il segmente alors virtuellement chaque super-réseau applicable en blocs /24 et sélectionne le premier candidat qui n'est pas déjà attribué.
- Enregistrement** : Enfin, ce sous-réseau disponible est officiellement réservé par la création d'un nouvel enregistrement dans la base de données, qui contient tous ses détails techniques (bridge, CIDR, etc.)

## 6.6. Réalisation

Cette section présente les nouvelles fonctionnalités de gestion réseau et de sécurité implémentées durant le Sprint 2. Elle suit le flux logique, en commençant par le processus manuel qui conditionne le bon fonctionnement de l'automatisation, jusqu'aux interfaces de supervision pour l'administrateur.

### 6.6.1. Étape 1 : Le Travail de l'Administrateur sur l'Infrastructure (Processus Manuel) (US-16 , US-17)

Comme défini dans l'objectif du sprint, la mise en place de l'infrastructure physique pour un nouveau réseau client reste une opération manuelle à ce stade. À la suite de l'allocation logique d'un nouveau réseau par Odoo (par exemple, le bridge vmbr10\_0 pour le sous-réseau 10.10.0.0/24), l'administrateur doit effectuer les configurations suivantes directement sur les serveurs Proxmox et le bastion.

- L'administrateur modifie le fichier `/etc/network/interfaces` de l'hôte Proxmox pour déclarer le nouveau bridge

```
# Pont pour le réseau interne
auto vmbr10_0
iface vmbr10_0 inet static
    address 10.10.0.1/24
    bridge-ports none
    bridge-stp off
    bridge-fd 0
# END ANCTIBLE MANAGED BLOCK
```

Name ↑	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR
ens33	Network Device	Yes	Yes	No			
vmbr0	Linux Bridge	Yes	Yes	No	ens33		
vmbr10_0	Linux Bridge	Yes	Yes	No			10.10.0.1/24

Figure 64 : Configuration du Bridge Réseau sur Proxmox

- B. Ensuite, l'administrateur doit spécifier au service DHCP d'écouter les requêtes sur les nouveaux bridges clients en modifiant le fichier /etc/default/isc-dhcp-server.

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#           Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="vmbr10_0"
INTERFACESv6=""
root@hostexample:~#
```

Figure 65 : Configuration des Interfaces d'Écoute du Service DHCP.

- C. Pour que les VMs clientes obtiennent automatiquement une adresse IP, l'administrateur ajoute une nouvelle déclaration de sous-réseau dans le fichier de /etc/dhcp/dhcpd.conf).

```
subnet 10.10.0.0 netmask 255.255.255.0 {
    range 10.10.0.2 10.10.0.254;
    option routers 10.10.0.1;
    option domain-name-servers 8.8.8.8;
    default-lease-time 600;
    max-lease-time 7200;
}
```

Figure 66 : Configuration des Pools d'Adresses DHCP

- D. Afin de fournir un accès Internet aux VMs situées sur le réseau privé, une règle de MASQUERADE est ajoutée via iptables sur l'hôte Proxmox.

```
root@hostexample:~# iptables -t nat -L POSTROUTING -n -v
Chain POSTROUTING (policy ACCEPT 296 packets, 20902 bytes)
pkts bytes target     prot opt in      out      source               destination
      0     0 MASQUERADE  0    -- *       vmbr0   10.10.0.0/24          0.0.0.0/0
root@hostexample:~#
```

Figure 67 : Configuration des Règles de NAT via iptables

- E. Pour garantir un accès externe sécurisé, l'administrateur crée un utilisateur dédié pour le client sur le serveur bastion, lui permettant de se connecter via SSH puis de "sauter" vers sa VM privée.

```

Non sécurisé https://192.168.162.129:8006/?console=kvm&novnc=1&vmid=101&vmname=
[root@Bastion home]# ls
bastion_setup_user  cloud-user  firas_jumper
[root@Bastion home]# cd firas_jumper/
[root@Bastion firas_jumper]# cd .ssh
[root@Bastion .ssh]# cat authorized_keys
ssh-rsa AAAAB3MzaC1yc2EAAAQABAAAQACQJ+RuRUYjIvgu
PcYHkjV3voR/pq698rdY01C0gIzXsMDJlc1mag/h4GZsCP+AIJi
Shu5ZSDUcHvzxy20oRWgw3XxZf7gCrEp/D5qQ1wfTEkWGssICP
tDZ5KfeU3dRoPHedtHxExSm45Pu9b1Sp8p3tLofqn/2nNQjUxZE
BMRLZtK05HVhNA0Y6LQxCn6koGk1CEjEct3QMTw4UYDSJL7Xqr>
[root@Bastion .ssh]#

```

Figure 68 : Configuration de l'Accès Client sur le Bastion SSH

### 6.6.2. Étape 2 : Configuration de la Stratégie Réseau dans Odoo (US-10)

Côté Odoo, l'administrateur définit la stratégie d'allocation en configurant les pools de super-réseaux. La Figure 69 montre la section des paramètres où ces plages d'adresses IP sont définies.

Configuration des Réseaux Clients Dédiés	
<b>Super-Réseau Principal Clients Standards</b> Super-réseau principal pour clients standards (ex: 10.10.0.0/16). Requis.	<b>Super-Réseau Réserve Clients Standards</b> Super-réseau de réserve pour clients standards (ex: 10.11.0.0/16).
10.10.0.0/16	10.11.0.0/16
<b>Super-Réseau Principal Clients Entreprise</b> Super-réseau principal pour clients "Entreprise" (ex: 172.16.0.0/16).	<b>Super-Réseaux Entreprise Additionnels (un CIDR par ligne)</b> Super-réseaux Entreprise additionnels (CIDR, séparés par virgule).
172.16.0.0/16	172.17.0.0/16,172.18.0.0/16
<input checked="" type="checkbox"/> <b>Allouer Réseaux 'Entreprise' aux Sociétés par Défaut</b> Si coché, les partenaires 'Société' utiliseront les pools 'Entreprise'.	<b>Préfixe pour Nom de Bridge Client</b> Préfixe pour les noms de bridge (ex: vmbr).
1	vmbr
<b>Offset IP Passerelle dans Sous-réseau Client</b> Offset pour l'IP de la passerelle dans chaque /24 (ex: 1).	<b>Noms de Bridges Proxmox à Exclure</b> Noms de bridges à ne jamais utiliser (ex: vmbr0,vmbr1). Séparés par virgule.
1	vmbr0

Figure 69 : Interface de Configuration des Super-Réseaux

### 6.6.3. Étape 3 : Allocation Automatique et Déploiement Isolé (US-19, US-15, US-04)

Lorsqu'un nouveau client passe sa première commande, le système exécute la logique d'allocation (US-19). Il crée un nouvel enregistrement de "Réseau Client" dans Odoo, comme le montre la Figure 70.

RÉSEAUX CLIENT PROXMOX ASSIGNÉS						
Nom du Réseau	Nom du Bridge P...	Sous-réseau (Forma...	Passerelle du Sous...	Réseau par Défaut pour c...	Nb VMs	Client Ass...
Firas Taboubi - (10.10.0.0/24) - [vmbr... vmbr10_0		10.10.0.0/24	10.10.0.1	<input checked="" type="checkbox"/>	3	
Ajouter une ligne						
VMS DÉPLOYÉES (INFORMATIONS RÉSEAU)						
Commande	Produit	Nom VM	Bridge VMID	Sous-réseau VM	Passerelle VM	Date Commande
S00184	VM Basique Linux (Ubuntu 24.04 LTS, 2 vCPU, 2 Coeurs, 4 Go RAM, 20 Go)	vm-basique-linux-firas-taboubi	202	vmbr10_0	10.10.0.0/24	10.10.0.1 2025-06-08
S00183	VM Basique Linux (Ubuntu 24.04 LTS, 1 vCPU, 1 Coeur, 2 Go RAM, 20 Go)	vm-basique-linux-firas-taboubi-i3	201	vmbr10_0	10.10.0.0/24	10.10.0.1 2025-06-08
S00183	VM Basique Linux (Ubuntu 24.04 LTS, 1 vCPU, 1 Coeur, 2 Go RAM, 20 Go)	vm-basique-linux-firas-taboubi-i1	200	vmbr10_0	10.10.0.0/24	10.10.0.1 2025-06-08

Figure 70 : Preuve de la Création d'un Réseau Client Logique dans Odoo

La VM est ensuite déployée sur le bridge privé correspondant (US-15), ce qui est visible dans l'interface Proxmox (Figure 71).

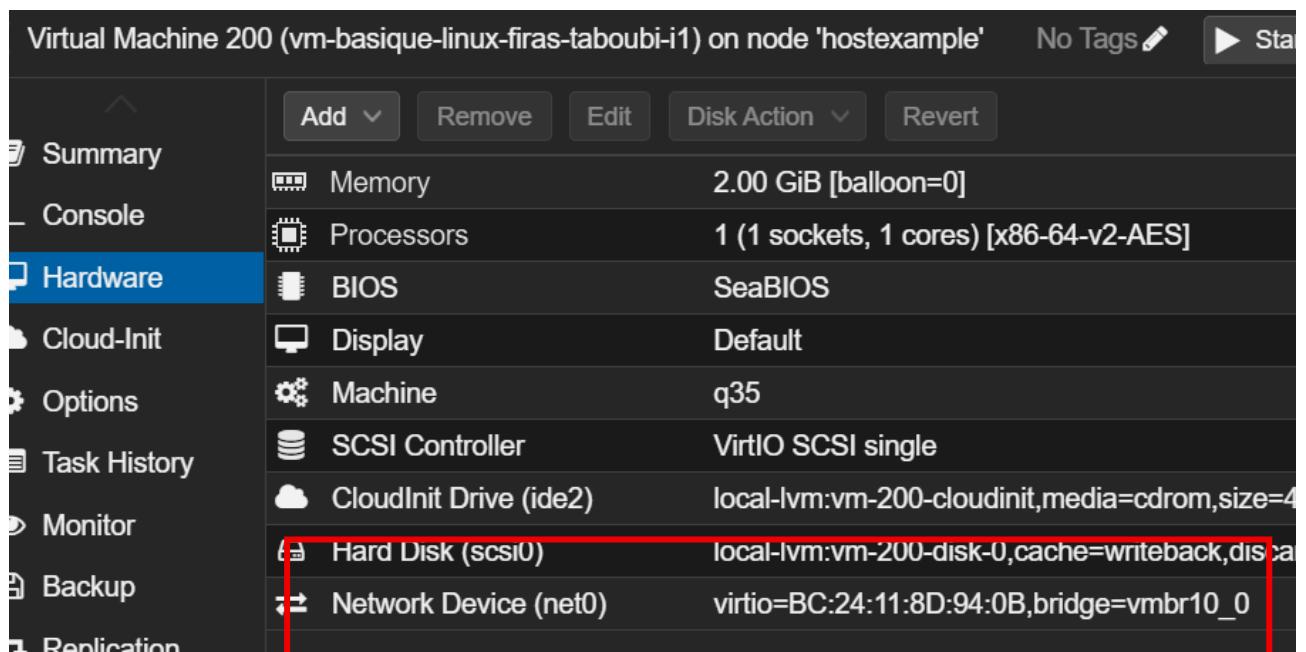


Figure 71 : Preuve du Déploiement de la VM sur le Bridge Isolé

Enfin, le client est notifié des détails de son nouveau réseau privé par email (US-04), comme illustré en Figure 72.



Figure 72 : Email d'Information Réseau Envoyé au Client

#### 6.6.4. Étape 4 : Supervision des Réseaux par l'Admin (US-11)

Pour répondre à l'US-11, une nouvelle vue a été créée pour permettre à l'administrateur de superviser l'ensemble des réseaux clients alloués, comme le montre la Figure 6.12.

Réseaux Clients Proxmox						
Nom du Réseau (Auto)	Client Associé	Nom du Bridge Proxmox	Sous-réseau (Format CIDR)	Passerelle du Sous-réseau	Réseau par D...	Nb VMs
<input type="checkbox"/> Firas Taboubi - (10.10.0.0/24) - [vmbr10_0]	Firas Taboubi	vmbr10_0	10.10.0.0/24	10.10.0.1	<input checked="" type="checkbox"/>	0

Figure 73 : Vue Centralisée des Réseaux Clients Alloués

### 6.6.5. Étape 5 : Gestion de l'Interaction Client (US-05)

Pour les clients existants qui passent une nouvelle commande, le système leur envoie un email leur demandant s'ils souhaitent regrouper la nouvelle VM sur un nouveau réseau ou non, répondant ainsi à l'**US-05**.

Bonjour Firas Taboubi,

Nous accusons réception de votre commande N° S00184. Celle-ci inclut le déploiement de la nouvelle instance de machine virtuelle Proxmox suivante :

- VM Basique Linux (Ubuntu 24.04 LTS, 2 vCPU, 2 Coeurs, 4 Go RAM, 20 Go) (Qté: 1)

Vous disposez actuellement de machines virtuelles configurées pour utiliser votre réseau client par défaut :

Sous-réseau : [10.10.0.0/24](#)  
Bridge Proxmox : [vmbr10\\_0](#)

Par défaut, votre nouvelle machine virtuelle sera automatiquement ajoutée à ce même réseau par défaut ([10.10.0.0/24](#)).  
Le déploiement avec cette configuration va commencer sous peu (ou a déjà commencé si la commande est confirmée).

Si vous souhaitez que cette nouvelle machine virtuelle soit placée dans un **sous-réseau différent et isolé** (par exemple, pour des raisons de sécurité spécifiques, pour séparer des environnements de test, ou pour des services ne devant pas communiquer avec vos VMs sur le réseau par défaut), veuillez suivre les instructions ci-dessous.

**➡ Pour utiliser un sous-réseau différent pour CES NOUVELLES VMs :**  
Veuillez répondre à cet email dès que possible en indiquant clairement : "OUI, JE SOUHAITE UN NOUVEAU RÉSEAU ISOLÉ POUR CETTE COMMANDE".  
Si plusieurs VMs sont concernées et que vous ne voulez qu'une partie d'entre elles sur un nouveau réseau, merci de le préciser.  
Notre équipe technique vous contactera pour discuter des détails. Cela peut nécessiter une intervention manuelle et un délai supplémentaire.

**➡ Pour conserver le réseau par défaut existant ([10.10.0.0/24](#)) :**  
Aucune action n'est requise de votre part. Vos nouvelles VMs seront sur ce réseau.

Si vous avez des questions, n'hésitez pas à nous contacter en répondant à cet email.

Figure 74 : Email de Choix de Réseau Envoyé au Client Existant

## 6.7. Bilan et Rétrospective du Sprint (Sprint Review)

Ce deuxième sprint a marqué une avancée majeure en matière de sécurité et de structuration du service. Voici les points clés à retenir.

### 6.7.1. Atteinte de l'Objectif et Synthèse des Acquis

L'objectif d'automatiser la gestion logique des réseaux et de formaliser le processus manuel a été atteint. Le système est désormais capable de :

- Allouer un sous-réseau privé et isolé à chaque client.
- Différencier l'allocation selon le type de client (Standard vs. Entreprise).
- Fournir aux administrateurs une vue claire des allocations réseau.

La solution a gagné en maturité, passant d'une preuve de concept à une architecture multi-locataire (multi-tenant) sécurisée sur le plan logique.

### 6.7.2. Leçon Apprise : La Limite du Processus Manuel

La principale leçon de ce sprint est la mise en évidence du goulot d'étranglement créé par les interventions manuelles. Le processus de configuration de l'infrastructure (création du bridge, configuration du DHCP, du NAT et du bastion) est :

- **Lent** : Il retarde le déploiement effectif du service au client.
- **Complexe et Répétitif** : Il est source potentielle d'erreurs humaines.
- **Non Scalable** : Il empêche une croissance rapide du nombre de clients.

Cette limitation est la justification la plus forte pour l'introduction de l'Infrastructure as Code (IaC), qui est l'objectif central du Sprint 3

## 6.8. Conclusion

En conclusion, le Sprint 2 a transformé notre solution en y intégrant les dimensions essentielles de sécurité et d'isolation réseau. Le système gère désormais la segmentation des clients de manière logique et automatisée, une étape indispensable pour un service viable.

Cependant, ce sprint a également mis en évidence que l'automatisation logique seule n'est pas suffisante. La dépendance à un processus manuel complexe pour la configuration de l'infrastructure physique reste le principal frein à une solution entièrement agile et scalable.

Le chapitre suivant détaillera le développement du Sprint 3, qui s'attaquera à ce défi en orchestrant directement l'infrastructure via des outils d'Infrastructure as Code, visant ainsi un niveau d'automatisation de bout en bout.

# **Chapitre 7 Sprint 3 :**

# **Orchestration de l'Infrastructure**

## **(IaC)**

## 7.1. Introduction

Les premiers sprints ont validé la gestion des commandes et des réseaux, mais ont montré deux limites majeures : une configuration réseau manuelle lente et des machines virtuelles livrées "vides". Ce troisième sprint vise à automatiser complètement ces étapes grâce à l'Infrastructure as Code (IaC). Nous utilisons Terraform et Ansible pour orchestrer la création de l'infrastructure et déployer des VMs prêtes à l'emploi.

## 7.2. Objectif du Sprint (Sprint Goal)

L'objectif formel de ce sprint est d'automatiser entièrement le cycle de vie du déploiement, de la création de l'infrastructure réseau jusqu'à la livraison d'une machine virtuelle prête à l'emploi, en éliminant les interventions manuelles sur les serveurs Proxmox et le bastion.

À l'issue de ce sprint, le système Odoo devra être capable de :

- Piloter Ansible pour configurer automatiquement les bridges réseau et les services associés (DHCP, NAT).
- Générer des identités sécurisées (clés SSH) et les déployer via Ansible sur le bastion d'accès.
- Utiliser Terraform pour provisionner les machines virtuelles en clonant des templates pré-configurés.
- Injecter une configuration personnalisée dans les VMs au démarrage grâce à Cloud-Init.
- Informer le client avec toutes les informations nécessaires pour accéder à son service entièrement fonctionnel.

## 7.3. Analyse Fonctionnelle et Planification du Sprint

Ce sprint étant axé sur une refonte technique interne majeure, l'analyse se concentre sur les nouvelles capacités requises par le système lui-même pour devenir un orchestrateur d'outils externes.

### 7.3.1. Analyse des Nouveaux Besoins

Contrairement aux sprints précédents, les évolutions du Sprint 3 ne se traduisent pas par de nouveaux cas d'utilisation pour les acteurs humains. L'objectif est de transformer radicalement le processus interne du système pour atteindre un niveau d'automatisation complet.

- **Pour le Système (Acteur principal de ce sprint) :** Le besoin principal est d'évoluer d'un simple exécuteur d'appels API à un orchestrateur d'outils IaC. Il doit pouvoir générer dynamiquement des fichiers de configuration (Terraform, Cloud-Init), exécuter des commandes externes (terraform, ansible-playbook), et interpréter leurs résultats pour garantir le succès du déploiement.
- **Pour l'Administrateur :** Son besoin évolue de la simple configuration API à la configuration des chemins et des paramètres nécessaires à cette nouvelle orchestration.

### 7.3.2. Périmètre du Sprint (Sprint Backlog)

Le backlog de ce sprint a été constitué pour couvrir l'ensemble du processus d'automatisation IaC, depuis la configuration de l'infrastructure jusqu'à la livraison d'une VM fonctionnelle.

Tableau 27 : User Stories du Sprint 3

ID	Epic	User Story
US-06	Expérience Client	En tant que Client, je veux recevoir un mail d'accès final à ma VM afin de pouvoir me connecter en SSH via le bastion.
US-20	Automatisation	En tant que Système, je veux automatiser la configuration du réseau physique (Bridge, DHCP, NAT) afin d'éliminer les interventions manuelles.
US-21	Automatisation	En tant que Système, je veux générer les clés SSH et configurer l'utilisateur sur le bastion afin d'automatiser la mise en place de l'accès sécurisé.
US-22	Automatisation	En tant que Système, je veux déployer la VM via IaC en clonant un Template afin de garantir un provisionnement rapide et standardisé.
US-23	Automatisation	En tant que Système, je veux installer des services additionnels sur la VM afin de livrer un environnement prêt à l'emploi.
US-31	Administration Odoo	En tant qu'Admin, je veux configurer les chemins et les paramètres des outils d'orchestration (Terraform, Ansible) afin que le système puisse les piloter correctement.

### 7.3.3. Plan d'Action

Chaque User Story a été décomposée en tâches techniques pour organiser le développement.

**Tableau 28 : Décomposition des Tâches Techniques du Sprint 3**

ID User Story	Tâches Techniques à Réaliser
US-31	<ul style="list-style-type: none"> <li>- Ajouter les champs de configuration pour les chemins des projets Terraform et Ansible dans les paramètres Odoo.</li> </ul>
US-20	<ul style="list-style-type: none"> <li>- Développer un playbook Ansible pour la création/configuration des bridges réseau sur les hôtes Proxmox.</li> <li>- Développer un playbook pour la configuration du DHCP et du NAT.</li> <li>- Implémenter la logique dans Odoo pour appeler ces playbooks avec les bonnes variables.</li> </ul>
US-21	<ul style="list-style-type: none"> <li>- Implémenter la génération de paires de clés SSH dans Odoo (via la librairie cryptography).</li> <li>- Développer un playbook Ansible pour créer un utilisateur "jumper" sur le bastion et y ajouter sa clé publique.</li> <li>- Implémenter la logique dans Odoo pour appeler ce playbook.</li> </ul>
US-22	<ul style="list-style-type: none"> <li>- Créer les fichiers de configuration Terraform (main.tf, variables.tf, etc.).</li> <li>- Développer la logique dans Odoo pour générer dynamiquement le fichier de variables (.tfvars.json) pour chaque déploiement.</li> <li>- Implémenter la fonction dans Odoo pour exécuter les commandes terraform init et terraform apply.</li> </ul>
US-23	<ul style="list-style-type: none"> <li>- Développer la logique dans Odoo pour générer le fichier de configuration user-data pour Cloud-Init, incluant l'installation de packages (ex: node-exporter) et la configuration des utilisateurs et clés SSH sur la VM.</li> </ul>
US-06	<ul style="list-style-type: none"> <li>- Créer un nouveau template d'email pour envoyer au client toutes les informations d'accès final (IP de la VM, utilisateur de la VM, utilisateur du bastion, exemple de commande de connexion SSH).</li> </ul>

## 7.4. Conception Technique : L'Architecture d'Orchestration IaC

Ce sprint introduit un changement majeur dans l'architecture du projet, passant d'une approche d'appels API directs à une véritable orchestration d'outils d'Infrastructure as Code (IaC). Cette section détaille la nouvelle architecture, le flux d'exécution automatisé, et les évolutions du modèle de données nécessaires pour piloter Terraform et Ansible depuis Odoo.

### 7.4.1. Vue d'Ensemble de l'Architecture d'Orchestration

Le diagramme d'architecture ci-dessous (Figure 7.1) illustre la nouvelle répartition des rôles. Odoo agit désormais comme **le cerveau de l'orchestration**. Il ne communique plus directement avec Proxmox pour le provisionnement, mais pilote les outils IaC (Ansible et Terraform) qui, eux, sont responsables d'agir sur l'infrastructure cible (nœuds Proxmox et VM Bastion).

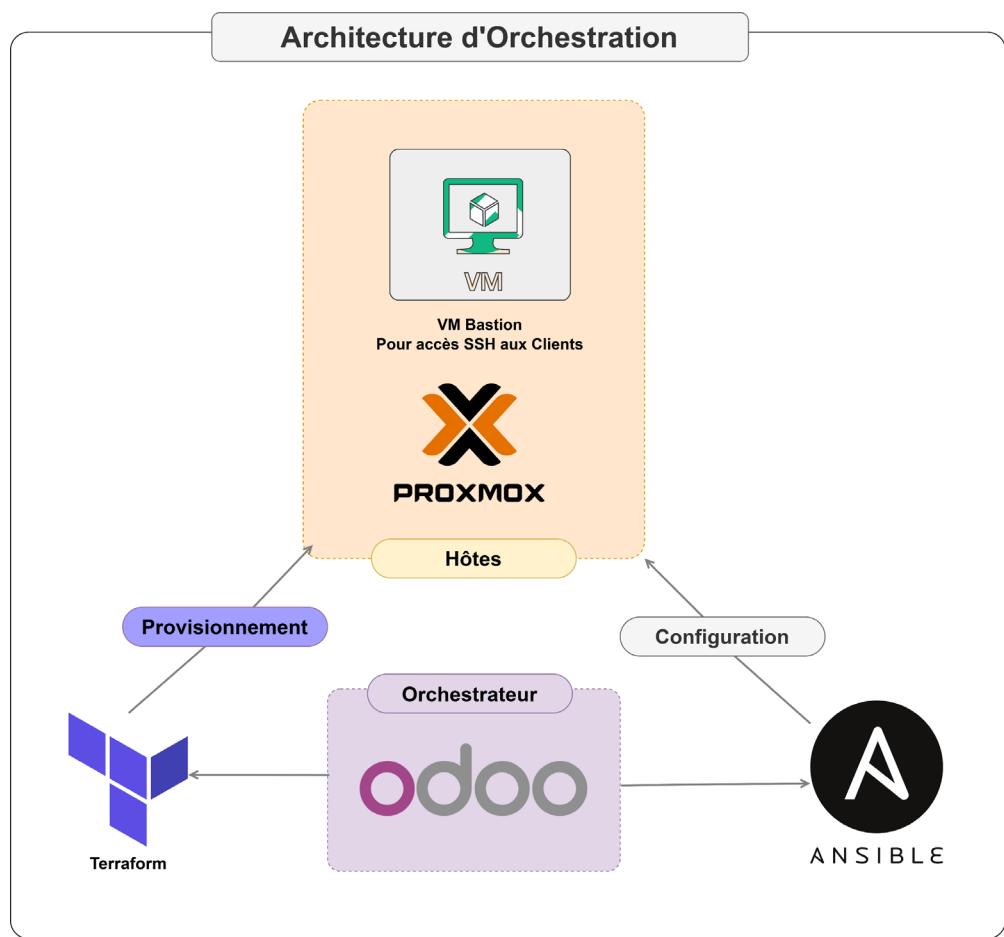


Figure 75 : Diagramme d'Architecture Globale (Odoo + IaC)

### 7.4.2. Le Flux d'Exécution Automatisé (Diagramme de Séquence)

Le processus de déploiement complet a été repensé pour intégrer les appels aux outils externes. Le diagramme de séquence suivant (Figure 7.2) modélise ce nouveau flux d'orchestration de bout en bout.

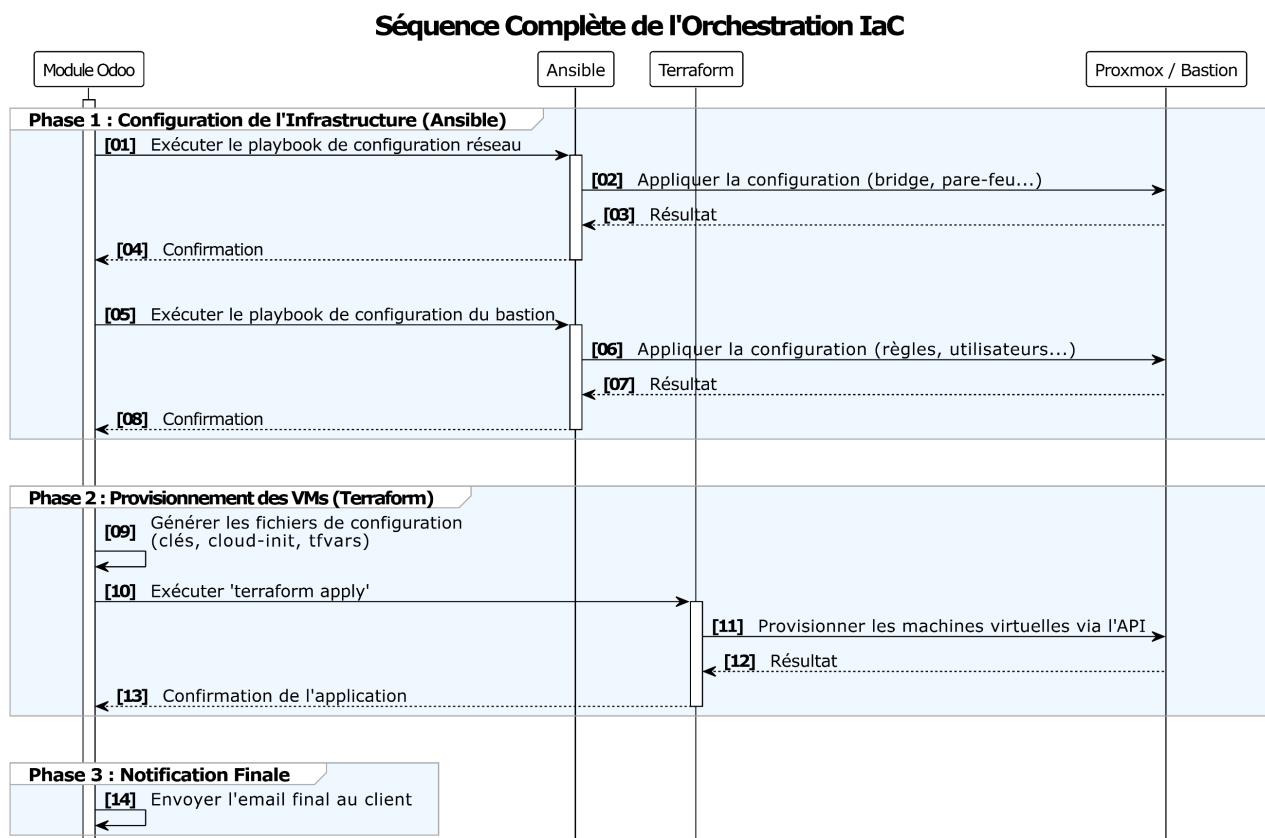


Figure 76 : Séquence Complète de l'Orchestration IaC

#### Phase 1 : Configuration de l'Infrastructure (Ansible)

Cette phase prépare l'environnement de base en utilisant Ansible pour automatiser la configuration.

- [01-04] **Configuration Réseau** : Odoo déclenche un playbook Ansible pour créer et configurer les éléments réseau (bridges, pare-feu, DHCP, etc.). Une fois terminé, Ansible renvoie un retour d'état à Odoo.
- [05-08] **Configuration du Bastion** : Un second playbook configure le serveur bastion (ajout des utilisateurs, règles d'accès pour le client, etc.).

## Phase 2 : Provisionnement des VMs (Terraform)

Une fois l'infrastructure prête, Terraform est utilisé pour déployer les machines virtuelles.

- [09] **Génération des Configurations** : Odoo génère dynamiquement les fichiers nécessaires à Terraform, notamment les clés SSH, les scripts cloud-init pour l'initialisation des VMs, ainsi que le fichier de variables tfvars contenant toutes les spécifications (taille de la VM, réseau à utiliser, etc.).
- [10-13] **Déploiement Terraform** : Odoo exécute ensuite la commande terraform apply. Terraform interprète les fichiers de configuration, communique avec l'API Proxmox pour provisionner les machines virtuelles selon les spécifications, et attend la fin de la création. Il retourne ensuite un statut de succès et les informations de la VM créée (outputs) au module Odoo.

## Phase 3 : Notification Finale

- [14] **Envoi de l'Email** : Odoo informe le client par email que ses services sont prêts, avec les informations d'accès.

### 7.4.3. Évolution du Modèle de Données (Sprint 3)

Pour permettre à Odoo de piloter ces outils, le modèle de données a été enrichi. De nouveaux champs ont été ajoutés pour stocker les configurations IaC, les identités (clés SSH), et les journaux d'exécution. Le diagramme de classes (Figure 7.3) met en évidence ces évolutions

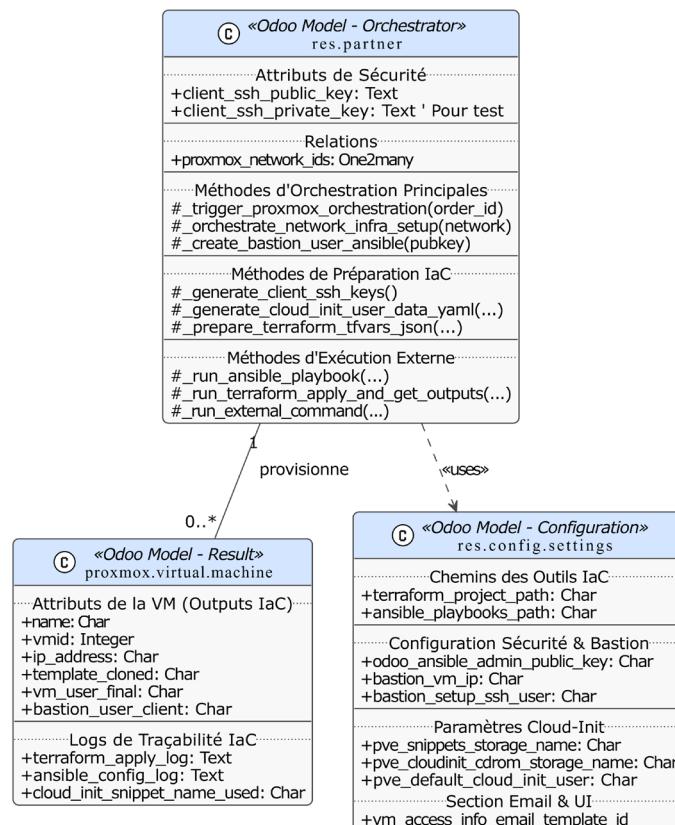


Figure 77 : Évolution du Diagramme de Classes (Focus IaC)

1. **L'Orchestrator (res.partner) :** Au cœur du système, ce modèle a été étendu pour agir comme le chef d'orchestre. Il est responsable du déclenchement de l'ensemble du flux de provisionnement, de la génération dynamique des fichiers de configuration (Cloud-Init, variables Terraform) et du lancement des appels vers les outils externes comme Ansible et Terraform.
2. **La Configuration (res.config.settings) :** Ce modèle sert de "tableau de bord" centralisé pour l'administrateur. Il fournit à l'orchestrator tous les paramètres indispensables à son fonctionnement, tels que les chemins d'accès aux projets Terraform et Ansible, les clés SSH, et les adresses IP des serveurs cibles.
3. **Le Résultat (proxmox.virtual.machine) :** Cet objet représente le produit final du processus d'automatisation. Il stocke de manière persistante toutes les informations relatives à la machine virtuelle créée par Terraform, incluant ses spécifications, ses identifiants et, de manière cruciale, les journaux de traçabilité des processus IaC pour l'audit et le débogage.

La relation de dépendance << uses >> montre que l'orchestrator s'appuie sur la configuration pour fonctionner, tandis que la relation de composition indique que l'orchestrator est conçu pour produire des enregistrements de machines virtuelles.

## 7.5. Réalisation

Cette section présente l'implémentation concrète du nouveau flux d'orchestration IaC. Chaque étape du processus, de la configuration dans Odoo à l'exécution des outils externes, est illustrée par des extraits de code et des captures d'écran.

### 7.5.1. Étape 1 : Configuration des Outils d'Orchestration (US-31)

Pour permettre à Odoo de piloter Terraform et Ansible, de nouveaux champs de configuration ont été ajoutés dans les paramètres du module. Comme le montre la Figure 7.4, l'administrateur doit désormais spécifier les chemins vers les projets IaC et les informations relatives au serveur bastion.

The screenshot shows the Odoo configuration interface for 'Outils d'Orchestration'. It displays three main sections: 'Chemin du Projet Terraform', 'Chemin des Playbooks Ansible', and 'VM Bastion (pour Accès Client et Ansible)'. Under 'Chemin du Projet Terraform', the value is '/opt/odoo/custom-addons/cus'. Under 'Chemin des Playbooks Ansible', the value is also '/opt/odoo/custom-addons/cus'. In the 'VM Bastion' section, the 'Adresse IP de la VM Bastion' is set to '192.168.162.100'. The 'Utilisateur SSH Admin sur Bastion (pour Ansible Odoo)' is set to 'bastion\_setup\_user'. The 'Stockage Proxmox pour Disque ISO Cloud-Init' is set to 'local-lvm' and the 'Stockage Proxmox pour Snippets Cloud-Init' is set to 'local'. A 'Modèle Email: Informations d'Accès VM' section is also visible at the bottom.

Chemin du Projet Terraform	Chemin des Playbooks Ansible
Chemin absolu vers le répertoire du projet Terraform sur le serveur Odoo. <code>/opt/odoo/custom-addons/cus</code>	Chemin absolu vers le répertoire racine des playbooks Ansible sur le serveur Odoo. <code>/opt/odoo/custom-addons/cus</code>

VM Bastion (pour Accès Client et Ansible)	
Adresse IP de la VM Bastion Adresse IP de la VM Bastion (accessible depuis le serveur Odoo). <code>192.168.162.100</code>	Utilisateur SSH Admin sur Bastion (pour Ansible Odoo) Utilisateur sur le Bastion pour la gestion par Ansible (ex: <code>bastion_setup_user</code> ). <code>bastion_setup_user</code>

Configuration des Réseaux Clients Dédiés	
Stockage Proxmox pour Disque ISO Cloud-Init Stockage pour le disque ISO de configuration Cloud-Init. <code>local-lvm</code>	Stockage Proxmox pour Snippets Cloud-Init Stockage Proxmox (type Directory) pour les snippets user-data YAML. <code>local</code>

Modèle Email: Informations d'Accès VM
Modèle pour l'envoi des informations d'accès à une VM.  Proxmox: Informations d'Accès à votre VM →

Figure 78 : Nouveaux Paramètres de Configuration IaC dans Odoo

### 7.5.2. Étape 2 : Automatisation de l'Infrastructure Réseau via Ansible (US-20)

Le processus manuel de configuration réseau du Sprint 2 est désormais entièrement automatisé. Odoo appelle dynamiquement les playbooks Ansible correspondants pour créer le bridge et configurer le DHCP et le NAT. La Figure 7.5 montre un extrait du playbook responsable de la création du bridge sur les hôtes Proxmox.

```
source /etc/network/interfaces.d/*
# BEGIN ANSIBLE MANAGED BLOCK FOR BRIDGE vmbr10_0
# Pont pour le réseau interne des VMs clientes vmbr10_0
auto vmbr10_0
iface vmbr10_0 inet static
    address 10.10.0.1/24
    bridge-ports none
    bridge-stp off
    bridge-fd 0
# END ANSIBLE MANAGED BLOCK FOR BRIDGE vmbr10_0
root@hostexample:~# []
```

Figure 79 : Preuve de la Création du Bridge par Ansible sur un Nœud Proxmox

### 7.5.3. Étape 3 : Automatisation de l'Accès Sécurisé via Ansible (US-21)

La création de l'accès sécurisé pour le client est également automatisée. Le système génère une paire de clés SSH dans Odoo, puis appelle un playbook Ansible pour créer l'utilisateur "jumper" sur le bastion et y déployer la clé publique du client. La Figure 7.6 présente un extrait de ce playbook.

```
[root@Bastion firas_jumper]# ls -la
total 12
drwx----- 3 firas_jumper firas_jumper 74 Jun  8 20:28 .
drwxr-xr-x  5 root        root        70 Jun  8 20:28 ..
-rw-r--r--  1 firas_jumper firas_jumper 18 Apr 30 2024 .bash_logout
-rw-r--r--  1 firas_jumper firas_jumper 141 Apr 30 2024 .bash_profile
-rw-r--r--  1 firas_jumper firas_jumper 492 Apr 30 2024 .bashrc
drwx----- 2 firas_jumper firas_jumper 29 Jun  8 20:28 .ssh
[root@Bastion firas_jumper]#
```

Figure 80 : Utilisateur jumper créé sur le bastion avec succès

### 7.5.4. Étape 4 : Déploiement de la VM via Terraform et Cloud-Init (US-22, US-23)

La phase finale de l'orchestration est le déploiement de la VM elle-même, géré par Terraform (US-22). Odoo génère les fichiers de configuration, lance la commande `terraform apply`, et grâce à un script Cloud-Init, installe les services additionnels (US-23).

Les Figures 81 à 84 ci-dessous prouvent le succès de chaque étape de ce processus : la création de la VM sur le bon réseau, la configuration du mot de passe console, l'installation des services, et le déploiement de la clé SSH.

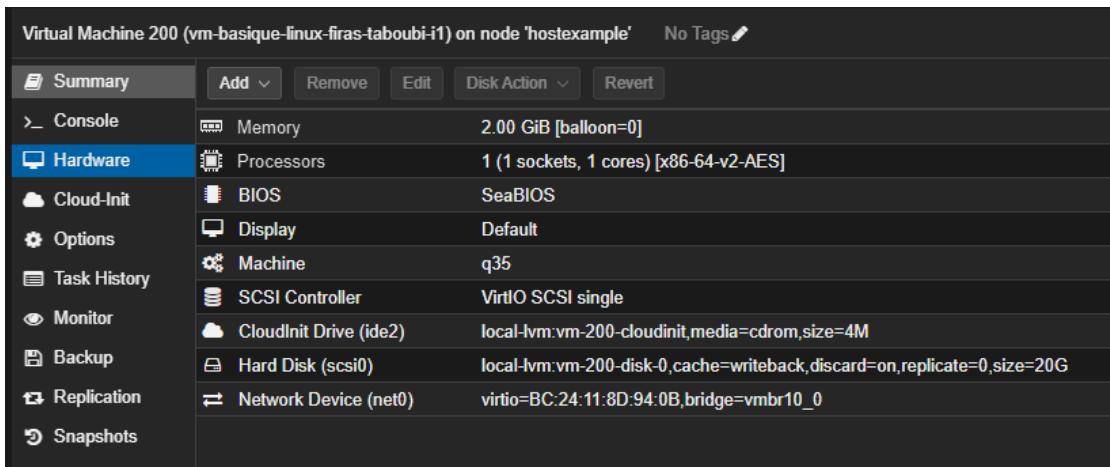


Figure 81 : Preuve du Déploiement de la VM sur son Bridge Privé par Terraform

L'un des avantages majeurs de cette approche est l'utilisation de Cloud-Init pour la configuration initiale de la machine. Cela permet de livrer un environnement prêt à l'emploi.

Par exemple, le système génère un mot de passe sécurisé pour l'accès console, le hashe, et l'injecte dans le script user-data. La Figure 7.8 montre une connexion réussie à la console de la VM, prouvant que la génération et l'injection du mot de passe ont fonctionné.

```
Non sécurisé https://192.168.162.129:8006/?console=kvm&novnc=1&vmid=200&vmname=vm-basique-linux
System information as of Fri May 23 19:47:52 UTC 2025
System load: 1.54 Processes: 27
Usage of /home: unknown Users logged in: 0
Memory usage: 5% IPv4 address for eth0: 10.10.10.2
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

firas1@vm-basique-linux-firas-taboubi-i1:~$
```

Figure 82 : Preuve de la Connexion à la Console via le Mot de Passe Généré

Pour répondre à l'US-23, le script Cloud-Init installe également des services additionnels. La Figure 7.9 confirme que l'agent de monitoring prometheus-node-exporter a bien été installé et est actif sur la machine virtuelle dès son premier démarrage

```
firas1@vm-basique-linux-firas-taboubi-i1:~$ systemctl status prometheus-node-exporter
● prometheus-node-exporter.service - Prometheus exporter for machine metrics
    Loaded: loaded (/usr/lib/systemd/system/prometheus-node-exporter.service; enabled; preset: enabled)
    Active: active (running) since Sat 2025-06-14 01:54:34 UTC; 2min 4s ago
      Docs: https://github.com/prometheus/node_exporter
      Main PID: 718 (prometheus-node)
        Tasks: 4 (limit: 2318)
       Memory: 14.3M (peak: 14.6M)
          CPU: 186ms
         CGrou... /system.slice/prometheus-node-exporter.service
                  └─ 718 /usr/bin/prometheus-node-exporter

Warning: some journal files were not opened due to insufficient permissions.
lines 1-12/12 (END)
```

Figure 83 : Vérification de l'Installation du Service Additionnel (Prometheus Node Exporter)

Enfin, pour garantir l'accès sécurisé via SSH, le script injecte la clé publique SSH du client dans le fichier authorized\_keys de l'utilisateur sur la VM. La Figure 7.10 montre le contenu de ce fichier, confirmant que la clé a été correctement déployée.

```
firas1@vm-basique-linux-firas-taboubi-i1:~/.ssh$ ls
authorized_keys
firas1@vm-basique-linux-firas-taboubi-i1:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAACQDj+RuRUYjIvguAY8MqMJ6jpFg9D1jNN/HLTXPoVmoj
/PvPZ+5/BcfiN1E0bR+PzN8YUK7sMi06r2HLbWEKQUT3LU4zGqyiuXyDq0QL6R1khqz0Qz3J5pb4mW0z
PcYHkjV3voR/pq698rdY01C0gIzXsMDJlclmag/h4GZsCP+AIJioftK/89eYob3UTcbfKSGBfRnW8cv6
PLgnmqhhqMAGd+5g31ab40Aa9kY96eXiC0hTI/r0ei6dR7PggYKkq3/RMjae+pv8rxpY86v8ui8DKIpI
Shu52SDVchvzxy20oRWgw3Xx2f7gCrEp/D5qQ1wfTEKWGsSICPvNdNqPDxI19BCIXpkzONQun2bbuwHr
jpVDFjk9W0/RBRY2KvGxQH5vUm0Bk7/5MNnS4ouXoVXz130MbMrliHfa7IsHq7Sdlr9bBmNcutCm0q+U
tD25KfeU3dRoPHedtHxExSm45Pu9b1Sp8p3tLofqr/2nNQjVxZEDgE6jfXeaMcyq43zBH/wk1Ac/0X7X
AOraN17+nI61CiGa6SJG/W4UDcl96bpj5wgMfvv7J9Kw67zHfrmExauv/uksEBgxGL5WWd5eRsEe618
BMRLZtK05HVhNA0Y6LQxCn6koGk1CEjEct3QMTw4UYDSJL7XqrXbh46q9EuSDQ7G5Shfk3Dd/+hwNy3g
tw=
firas1@vm-basique-linux-firas-taboubi-i1:~/.ssh$
```

Figure 84 : Vérification de l'Injection de la Clé Publique SSH du Client

Une fois l'ensemble de ce processus technique terminé avec succès, le cycle de déploiement se conclut par l'envoi d'un email final au client (**US-06**). Comme l'illustre la Figure 85, cet email récapitule toutes les informations nécessaires pour qu'il puisse accéder à son service désormais pleinement fonctionnel

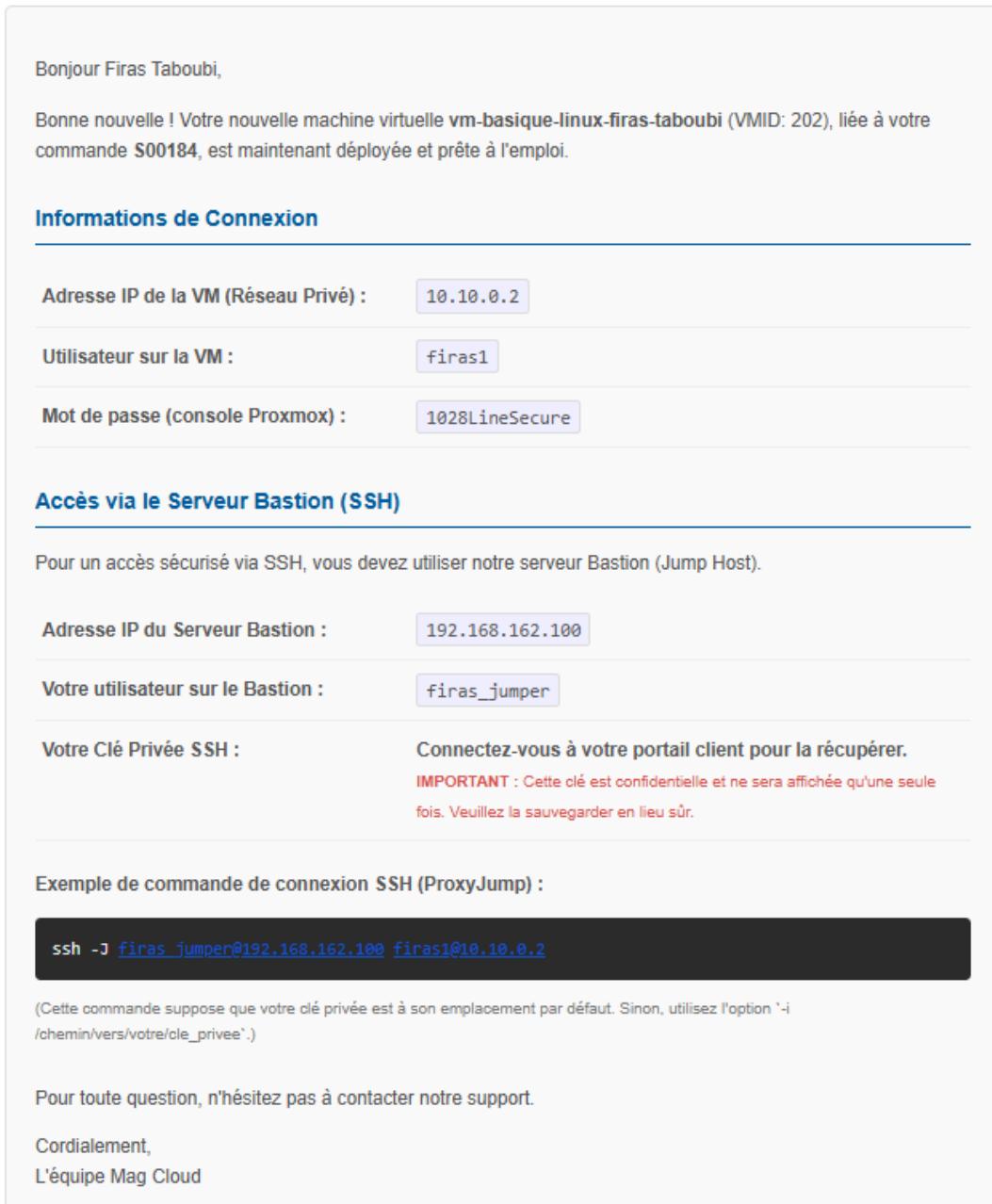


Figure 85 : Email d'Accès Final à la VM)

## 7.6. Bilan et Rétrospective du Sprint (Sprint Review)

Ce troisième sprint a permis d'atteindre un jalon majeur du projet : l'automatisation quasi-complète du déploiement.

### 7.6.1. Atteinte de l'Objectif et Synthèse des Acquis

L'objectif d'éliminer les interventions manuelles a été atteint. Le système est désormais capable d'orchestrer l'ensemble de la chaîne de déploiement via des outils d'Infrastructure as Code. Les acquis de ce sprint sont :

- **Automatisation de bout en bout** : Le processus, de la commande à la livraison d'une VM prête à l'emploi avec accès sécurisé, est entièrement automatisé.
- **Rapidité et Fiabilité** : Le déploiement est maintenant beaucoup plus rapide et moins sujet aux erreurs humaines.
- **Scalabilité** : La solution est prête à gérer un grand nombre de déploiements de manière efficace.
- **Pratiques Modernes** : L'utilisation de Terraform, Ansible et Cloud-Init aligne le projet sur les standards actuels du DevOps et de l'automatisation d'infrastructure.

### 7.6.2. Leçons Apprises : Complexité de l'Orchestration

La principale leçon de ce sprint réside dans la gestion de la complexité inhérente à l'orchestration d'outils externes. Les principaux défis surmontés ont été :

- La gestion sécurisée des secrets (clés API, clés SSH).
- La gestion des permissions pour l'utilisateur Odoo exécutant des commandes système.
- Le parsing fiable des logs de sortie de Terraform et Ansible pour déterminer le succès ou l'échec d'une opération.
- La gestion des timeouts pour des processus qui peuvent être longs.

Ces défis, bien que techniques, ont permis de construire une solution robuste et résiliente.

## 7.7. Conclusion

En conclusion, le Sprint 3 a transformé notre solution d'un simple automate en un véritable orchestrateur d'infrastructure. En intégrant les pratiques d'Infrastructure as Code, nous avons non seulement éliminé le goulot d'étranglement manuel identifié précédemment, mais nous avons également rendu le processus de déploiement rapide, répétable et scalable.

Maintenant que l'infrastructure est entièrement automatisée, le chapitre suivant détaillera le développement du Sprint 4, qui se concentrera sur la fiabilisation de ce nouvel environnement complexe en mettant en place une supervision active et un système d'alertes proactif.

# Chapitre 8 Sprint 4 : Monitoring et Fiabilisation

## 8.1. Introduction

Les sprints précédents ont permis de mettre en place un pipeline de déploiement automatisé sécurisé. La solution peut désormais créer des environnements clients isolés à la demande. Cependant, elle fonctionne encore comme une "boîte noire", sans visibilité sur son état interne. L'absence de supervision en temps réel limite la capacité à anticiper les incidents.

Ce chapitre présente le Sprint 4, qui vise à rendre la plateforme pleinement opérable. Il se concentre sur l'ajout d'une solution de monitoring et d'alerting, pour offrir un service plus transparent, fiable et gérable de façon proactive.

## 8.2. Objectif du Sprint (Sprint Goal)

L'objectif formel de ce sprint est de doter la plateforme de capacités de supervision et d'alerte complètes. Le système doit permettre aux administrateurs de visualiser les performances de l'infrastructure et d'être notifiés proactivement des incidents critiques.

Concrètement, cela signifie que notre infrastructure de support doit être capable de :

- **Collecter** en continu les métriques de performance des hôtes Proxmox et des VMs clientes.
- **Visualiser** ces données sur des tableaux de bord graphiques et intuitifs.
- **Détecter** automatiquement les conditions anormales (pannes, surcharges) sur la base de règles prédéfinies.
- **Notifier** immédiatement les administrateurs via un canal de communication moderne en cas d'alerte.

## 8.3. Analyse Fonctionnelle

Cette section définit les besoins fonctionnels spécifiques au Sprint 4, qui se concentrent sur la supervision de la plateforme et l'évolution du rôle de l'administrateur vers une fonction d'opérateur de service.

### 8.3.1. Identification des Acteurs

Les besoins de ce sprint sont entièrement centrés sur l'administrateur et les systèmes de support :

- **Pour l'Administrateur** : Son rôle s'étend au-delà du déploiement pour inclure celui d'Opérateur de Service. Il n'est plus seulement celui qui construit, mais aussi celui qui garantit le bon fonctionnement. Il doit pouvoir surveiller activement la

plateforme, analyser les tendances de performance et réagir rapidement aux incidents dont il est notifié.

- **Pour le Système de Monitoring (Prometheus/Grafana)** : Il agit comme un acteur système qui fournit les outils de visualisation. C'est l'interface principale de l'administrateur pour sa nouvelle mission de supervision.
- **Pour le Système d'Alerting (Alertmanager/Slack)** : C'est un acteur système proactif. Son rôle est de surveiller les données en permanence et de déclencher une communication vers l'administrateur lorsqu'une intervention est potentiellement requise.

Les nouvelles fonctionnalités de ce sprint se traduisent par deux cas d'utilisation essentiels pour l'administrateur, comme le montre le diagramme suivant.

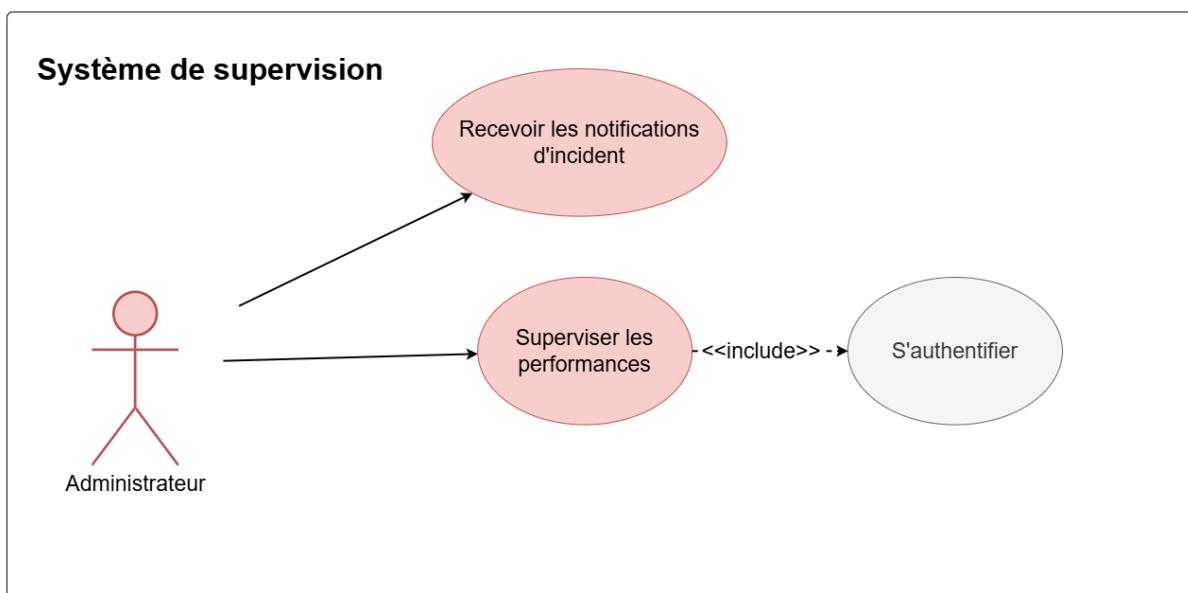


Figure 86 : Diagramme des Cas d'Utilisation - Rôle Administrateur (Sprint 4)

### 8.3.2. Spécifications Détaillées des Nouveaux Cas d'Utilisation

Pour formaliser le comportement attendu, les nouveaux cas d'utilisation de l'administrateur sont décrits ci-dessous.

Tableau 29 : Spécification du Cas d'Utilisation 'Superviser les performances de la plateforme'

Section	Description
<b>Nom</b>	Superviser les performances de la plateforme
<b>Acteur</b>	Administrateur
<b>Objectif</b>	(US-24) Fournir à l'administrateur des tableaux de bord graphiques pour visualiser en temps réel l'état de santé et les métriques de performance de l'infrastructure (hôtes) et des services (VMs clientes), afin de détecter proactivement les surcharges.
<b>Précondition</b>	L'administrateur est authentifié sur le réseau de gestion. La pile de monitoring (Prometheus, Grafana) est déployée et fonctionnelle. Les agents de collecte (Node exporter, pve_exporter) sont installés sur les hôtes, comme réalisé lors de la mise en place initiale de l'environnement (Chapitre 4).
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur se connecte à l'interface web de Grafana.</li> <li>2. Il navigue vers le tableau de bord "Vue d'ensemble de l'Infrastructure" pour vérifier l'état global des serveurs Proxmox.</li> <li>3. Il sélectionne le tableau de bord "Détails VM Client".</li> <li>4. Il choisit une VM spécifique dans une liste déroulante et analyse ses graphiques de performance (CPU, RAM, Réseau).</li> </ol>
<b>Scénarios d'Exception</b>	<p>A1 : <b>Source de données indisponible</b> - Si Prometheus ne répond pas, Grafana affiche une erreur indiquant que les données ne peuvent pas être chargées.</p> <p>A2 : <b>Tableau de bord mal configuré</b> - Si un panneau est mal configuré, il affiche une erreur au lieu d'un graphique.</p>
<b>Postcondition</b>	L'administrateur a une connaissance précise et à jour de l'état de la plateforme, lui permettant de prendre des décisions éclairées.

Tableau 30 : Spécification du Cas d'Utilisation 'Être notifié d'un incident critique'

Section	Description
<b>Nom</b>	Être notifié d'un incident critique
<b>Acteur</b>	Administrateur (destinataire), Système d'Alerting (initiateur)
<b>Objectif</b>	(US-25) Le système doit automatiquement envoyer une notification à l'administrateur sur un canal de communication dédié (Slack) lorsqu'un problème critique est détecté, afin qu'il soit informé immédiatement et puisse réagir dans les plus brefs délais.

<b>Précondition</b>	Un événement déclenchant une alerte se produit (ex: une VM cliente ne répond plus, l'utilisation du CPU d'un hôte dépasse 95% pendant 5 minutes). Les règles d'alerte sont configurées dans Prometheus et l'intégration avec Slack est active.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. Prometheus détecte une métrique qui viole une règle d'alerte.</li> <li>2. Une alerte est envoyée à Alertmanager.</li> <li>3. Alertmanager applique ses règles de groupage/routage et envoie une notification formatée au webhook du canal Slack #alerts-platform.</li> <li>4. L'administrateur reçoit la notification sur son ordinateur ou son appareil mobile.</li> </ol>
<b>Scénarios d'Exception</b>	<p>A1 : <b>Canal de notification indisponible</b> - Si l'API de Slack est en panne, Alertmanager ne pourra pas envoyer le message (des mécanismes de relance peuvent être configurés).</p> <p>A2 : <b>Fausse alerte (faux positif)</b> - Si une règle est trop sensible, elle peut se déclencher pour des conditions non critiques, créant du "bruit" pour l'administrateur.</p>
<b>Postcondition</b>	L'administrateur est informé d'un incident potentiellement impactant sans avoir à surveiller activement les tableaux de bord.

## 8.4. Planification du Sprint

### 8.4.1. Périmètre du Sprint (Sprint Backlog)

Pour atteindre l'objectif de supervision et d'alerte, le backlog de ce sprint s'est concentré sur les deux User Stories suivantes, qui couvrent l'ensemble de ces besoins, ainsi que sur les tâches techniques qui en découlent.

Tableau 31 : User Stories du Sprint 4

ID	Epic	User Story
US-25	Exploitation et Fiabilité	En tant qu'Admin, je veux superviser les performances de l'infrastructure et des VMs clientes via des tableaux de bord graphiques afin de détecter proactivement les surcharges et d'analyser les tendances.
US-26	Exploitation et Fiabilité	En tant qu'Admin, je veux recevoir des alertes en temps réel sur Slack en cas de problème critique (ex: VM inaccessible, CPU élevé) afin d'être informé immédiatement et de pouvoir réagir rapidement.

### 8.4.2. Plan d'action du Sprint

Chaque User Story a été décomposée en tâches techniques concrètes pour organiser le développement.

Tableau 32 : Plan d'action de sprint 4

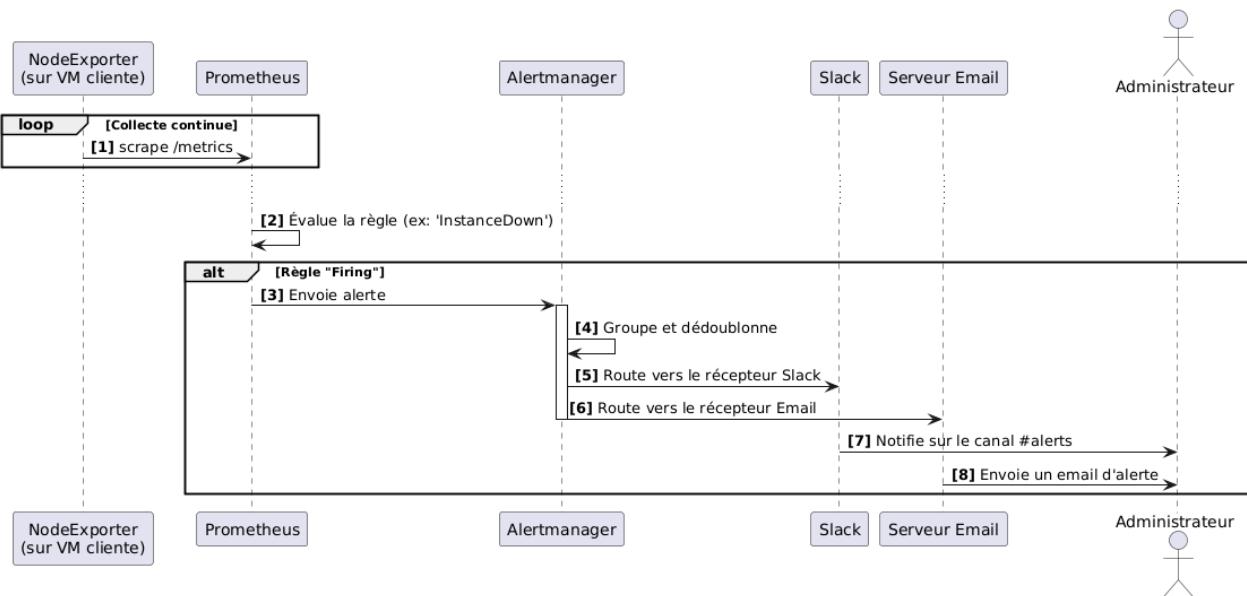
ID User Story	Tâches Techniques à Réaliser
US-24	<ul style="list-style-type: none"> <li>- Concevoir et implémenter un tableau de bord global pour l'infrastructure Proxmox dans Grafana.</li> <li>- Importer un modèle de tableau de bord pour la visualisation détaillée des VMs clientes.</li> </ul>
US-25	<ul style="list-style-type: none"> <li>- Rédiger et implémenter les règles d'alerte initiales dans Prometheus .</li> <li>- Configurer Alertmanager pour recevoir les alertes de Prometheus et les router.</li> <li>- Créer un webhook dans Slack et configurer Alertmanager pour y envoyer les notifications.</li> </ul>

### 8.5. Conception Technique :

La mise en œuvre des fonctionnalités de ce sprint repose sur l'intégration d'une pile de monitoring qui interagit avec notre infrastructure existante de manière non intrusive.

L'architecture s'articule autour d'un modèle "pull" où Prometheus est le composant central qui tire activement les informations.

1. **Exposition des Métriques** : Les agents node\_exporter et pve\_exporter, installés lors de la mise en place de l'environnement (Chapitre 4), exposent en continu les métriques de l'hôte et de Proxmox sur des points de terminaison HTTP (/metrics).
2. **Configuration Statique des Cibles** : Pour ce sprint, la configuration des cibles de monitoring est effectuée manuellement. L'administrateur édite directement le fichier de configuration de Prometheus (prometheus.yml) pour y ajouter l'adresse IP de chaque nouvelle VM cliente à superviser. Cette approche, bien que fonctionnelle, n'est pas scalable.
3. **Flux d'Alerte** : Les alertes suivent un cheminement clair : Prometheus les détecte, les envoie à Alertmanager qui les filtre, les groupe et les transmet à Slack. Cette séparation des rôles permet une gestion fine des notifications (ex : éviter les spams d'alertes, grouper les alertes similaires).



## 8.6. Réalisation

Cette section présente les nouvelles fonctionnalités de monitoring et d'alerting implémentées durant le Sprint 4.

À l'aide de l'interface de Grafana, des tableaux de bord ont été créés pour répondre aux besoins de l'administrateur.

- **Dashboard "Proxmox Host Overview"** : Ce tableau de bord fournit une vue d'ensemble critique de l'état des serveurs physiques.
- **Dashboard "VM Client Details"** : Un second tableau de bord utilise les variables de Grafana pour permettre la sélection dynamique d'une VM cliente et l'affichage de ses métriques spécifiques.

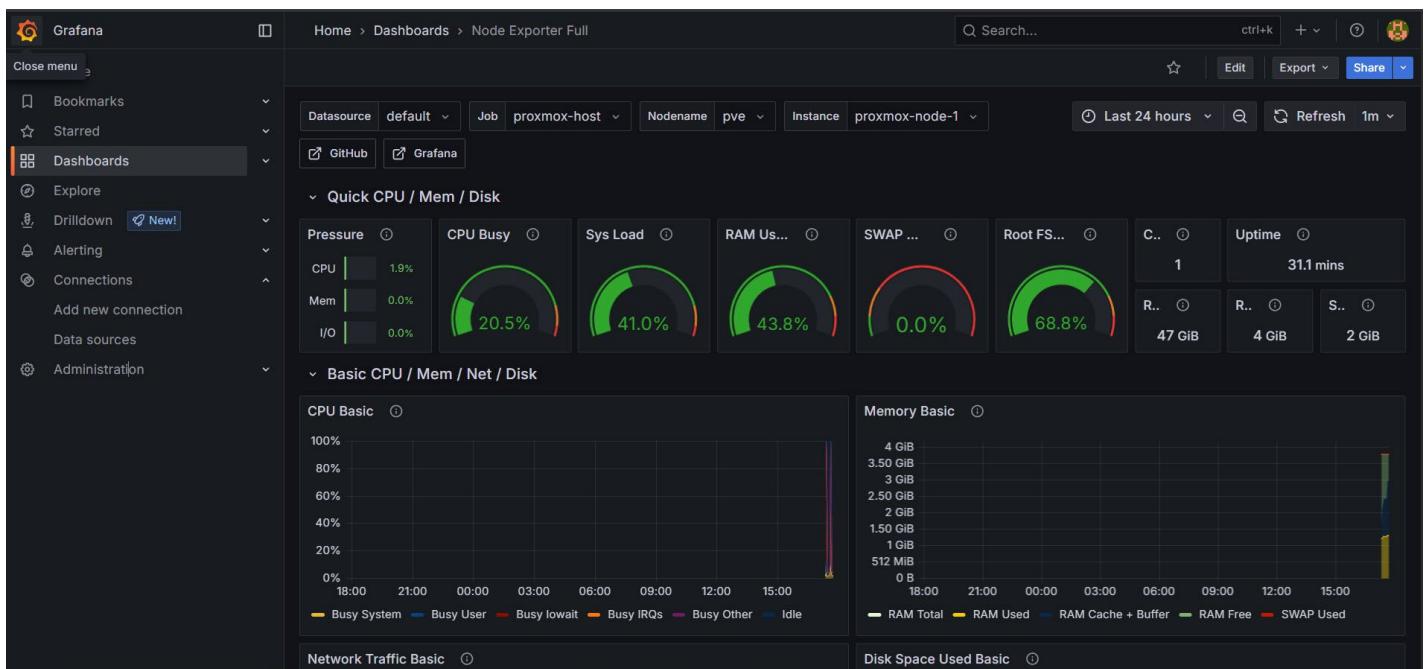
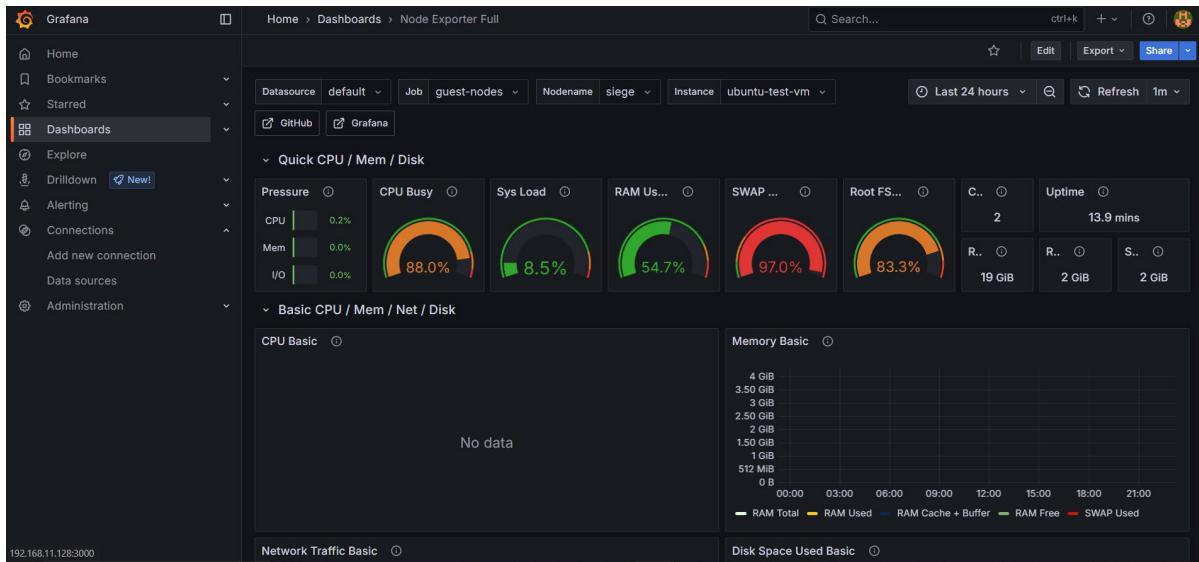


Figure 87 : Vue du Tableau de Bord Global de l'Infrastructure sur Grafana



Le système d'alerte a été configuré en trois étapes :

- Définition des règles** dans un fichier prometheus.rules.yml, avec des seuils initiaux
- Configuration d'Alertmanager** pour écouter Prometheus et pointer vers le webhook fourni par Slack.
- Test de bout en bout** en simulant une panne (arrêt d'une VM) et en vérifiant la réception de la notification sur le canal Slack dédié.

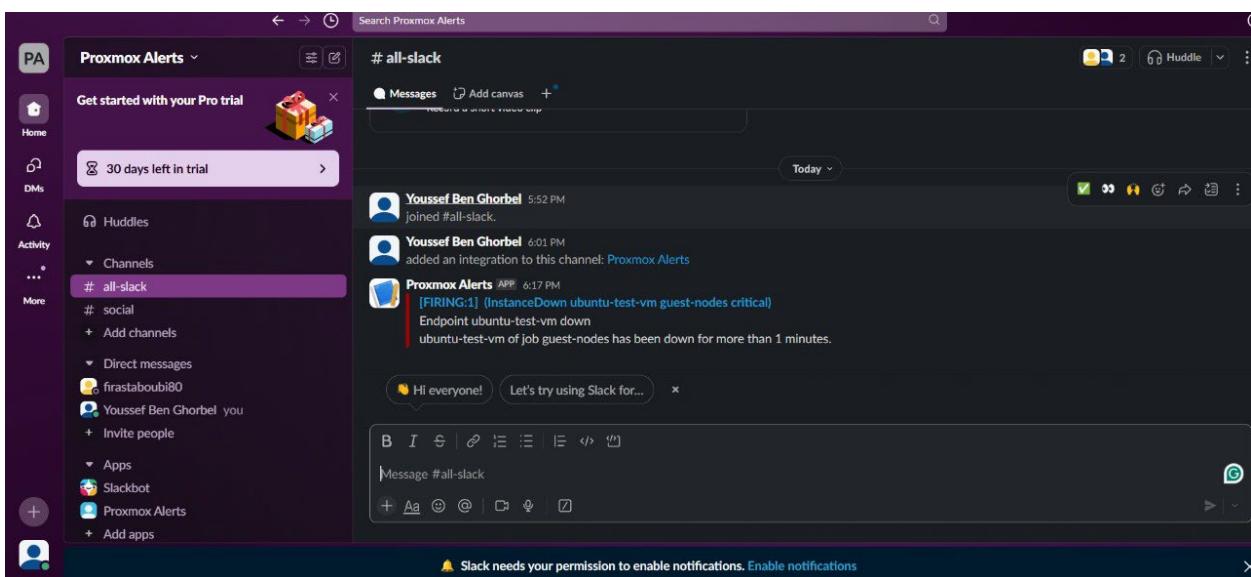


Figure 89 : Notification d'Alerte Reçue sur le Canal Slack

## 8.7. Bilan et Rétrospective du Sprint (Sprint Review)

Ce quatrième sprint a permis de poser les fondations d'une plateforme opérable et supervisée.

L'objectif de doter la plateforme de capacités de supervision et d'alerte a été atteint. Le système est désormais capable de :

- **Visualiser** les performances de manière centralisée et intuitive.
- **Alerter** proactivement l'administrateur en cas d'anomalie, permettant une gestion d'incident rapide.

La solution a gagné en robustesse et en fiabilité, deux qualités indispensables pour une offre de service commerciale.

La principale leçon de ce sprint est la mise en évidence d'un nouveau goulot d'étranglement. Bien que la pile de monitoring soit fonctionnelle, l'obligation pour l'administrateur d'ajouter manuellement chaque nouvelle VM au fichier de configuration de Prometheus est :

- **Source d'erreurs** : Un oubli ou une faute de frappe peut laisser une VM sans supervision.
- **Non Scalable** : Ce processus manuel empêche une croissance rapide du nombre de clients et annule en partie les bénéfices de l'automatisation du déploiement.

Cette limitation est la justification la plus forte pour l'introduction d'un mécanisme de découverte de services automatique, qui pourrait être l'objectif d'une future itération.

## 8.8. Conclusion

En conclusion, le Sprint 4 a ajouté la dimension d'observabilité à notre solution. En intégrant une pile de monitoring et d'alerting, nous avons considérablement augmenté la fiabilité et la maintenabilité de la plateforme. L'administrateur dispose désormais des outils nécessaires pour opérer le service de manière plus efficace.

# Chapitre 9 Sprint 5 : Sauvegarde et Reprise d'Activité

## 9.1. Introduction

Les sprints précédents ont permis de construire une plateforme automatisée, isolée, sécurisée et supervisée. Nous avons atteint un haut niveau de maturité opérationnelle. Cependant, la question fondamentale de la continuité de service en cas de sinistre majeur (panne matérielle, corruption de données, erreur humaine) reste ouverte. Sans une stratégie de sauvegarde et de restauration formalisée, tous les acquis précédents pourraient être anéantis par un seul incident.

Ce chapitre détaille le développement du Sprint 5, l'étape finale et cruciale qui dote notre solution de sa "colonne vertébrale" : la résilience. Nous nous concentrerons sur l'implémentation d'une stratégie de sauvegarde complète et, surtout, sur la création d'un Plan de Reprise d'Activité (PRA) documenté, transformant notre solution en un service véritablement robuste et digne de confiance.

## 9.2. Objectif du Sprint (Sprint Goal)

L'objectif formel de ce sprint est de définir et d'implémenter une stratégie complète de sauvegarde et de restauration des données pour assurer la continuité d'activité.

Concrètement, cela signifie que notre plateforme et notre documentation doivent permettre de :

- Configurer et exécuter des sauvegardes automatiques et régulières pour toutes les machines virtuelles critiques.
- Permettre à l'administrateur de lancer des sauvegardes manuelles au besoin.
- Fournir à l'administrateur un document clair et actionnable (le PRA) détaillant les procédures à suivre pour restaurer le service en fonction de différents scénarios de sinistre.
- Définir des Objectifs de Point de Reprise (RPO) et des Objectifs de Temps de Reprise (RTO) clairs.

## 9.3. Analyse Fonctionnelle

Cette section définit les besoins fonctionnels spécifiques au Sprint 5, qui se concentrent sur la protection des données et la capacité de l'administrateur à gérer un sinistre.

- **Pour l'Administrateur** : Son rôle s'enrichit de la responsabilité de **Gardien de la Continuité**. Il est non seulement opérateur, mais aussi le garant de la pérennité des données et du service. Il doit pouvoir configurer les politiques de sauvegarde et être capable de suivre et d'exécuter les procédures du PRA en situation de crise.

- Pour le Système de Sauvegarde (Proxmox Backup Server) : Il s'agit d'un acteur système central, responsable du stockage sécurisé et efficace des sauvegardes.

Les nouvelles fonctionnalités de ce sprint se traduisent par deux cas d'utilisation critiques pour l'administrateur.

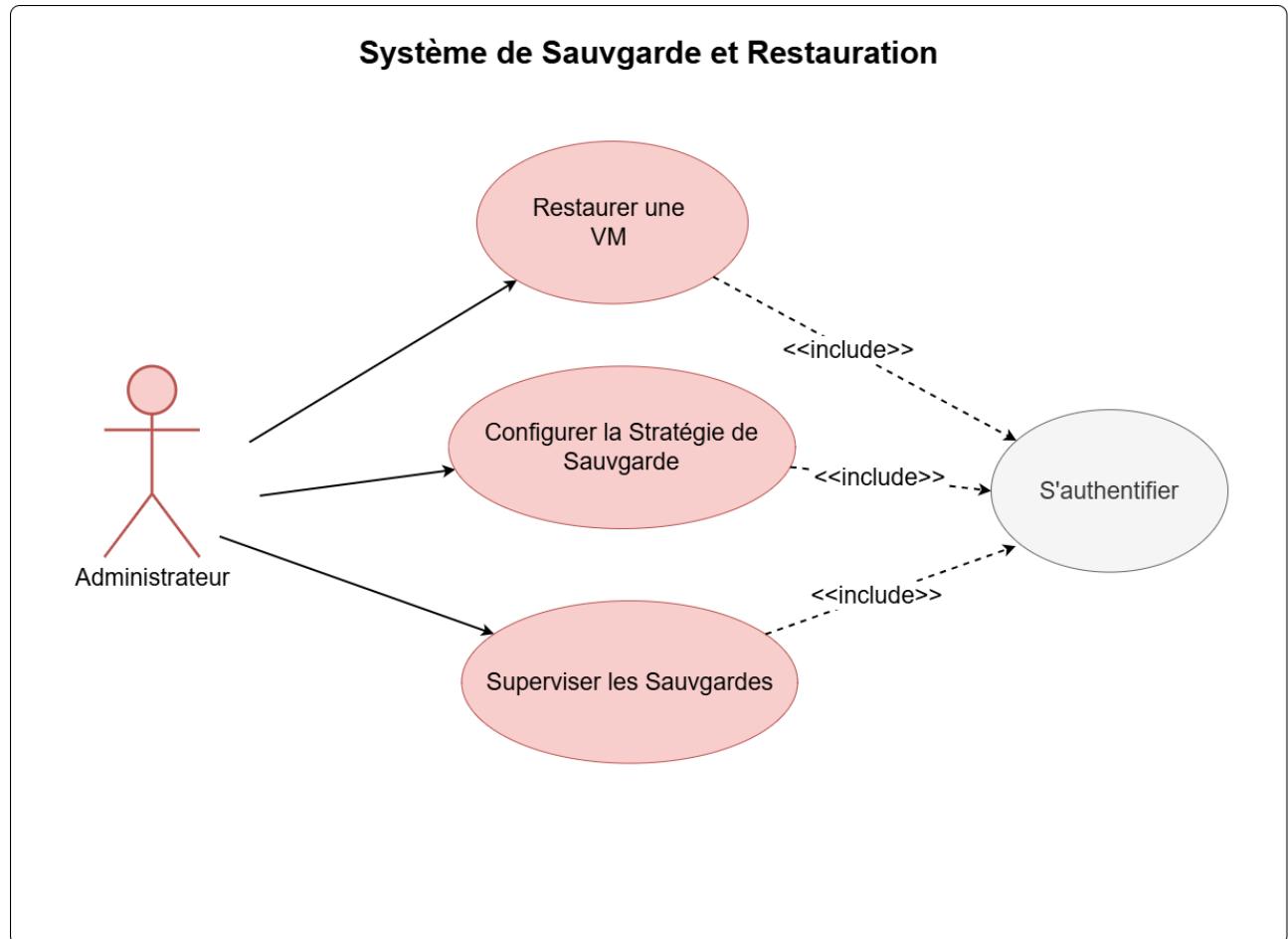


Figure 90 : Diagramme des Cas d'Utilisation - Rôle Administrateur ((Sprint 5)

**Tableau 33 : Spécification du Cas d'Utilisation 'Gérer la politique de sauvegarde'**

Section	Description
<b>Nom</b>	Gérer la politique de sauvegarde
<b>Acteur</b>	Administrateur
<b>Objectif</b>	(US-26) Permettre à l'administrateur de configurer des sauvegardes automatiques et de lancer des sauvegardes manuelles pour garantir une protection flexible et fiable des données des VMs.
<b>Précondition</b>	L'administrateur est connecté à l'interface Proxmox VE. Le serveur Proxmox Backup Server (PBS) est configuré et accessible comme solution de stockage.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur navigue vers la section "Sauvegarde" du Datacenter dans Proxmox VE.</li> <li>2. Il crée une nouvelle tâche de sauvegarde.</li> <li>3. Il définit la planification (ex: tous les jours à 02:00), les VMs à inclure, la destination (le datastore PBS) et la politique de rétention.</li> <li>4. Pour une sauvegarde ponctuelle, il sélectionne une VM, navigue vers son onglet "Sauvegarde" et clique sur "Sauvegarder maintenant".</li> </ol>
<b>Scénarios d'Exception</b>	A1 : Stockage PBS plein/inaccessible - La tâche de sauvegarde échoue et une notification d'erreur est générée dans Proxmox VE.
<b>Postcondition</b>	Les données des VMs sont sauvegardées périodiquement et de manière sécurisée, conformément à la politique définie.

**Tableau 34 : Spécification du Cas d'Utilisation 'Exécuter le Plan de Reprise d'Activité'**

Section	Description
<b>Nom</b>	Exécuter le Plan de Reprise d'Activité
<b>Acteur</b>	Administrateur
<b>Objectif</b>	(US-27) Fournir à l'administrateur un document (PRA) qui le guide pas à pas pour restaurer les services en cas de sinistre, afin de minimiser le temps d'arrêt et la perte de données.
<b>Précondition</b>	Un incident majeur est survenu (ex: crash d'un serveur hôte). L'administrateur a accès au document PRA.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur identifie le type de sinistre.</li> <li>2. Il consulte le document PRA et localise la procédure correspondante au scénario</li> </ol>

	<p>3. Il suit méthodiquement les étapes décrites : remplacement matériel, réinstallation de Proxmox, connexion au PBS, restauration des VMs critiques.</p> <p>4. Il suit le plan de communication pour informer les parties prenantes.</p>
Scénarios d'Exception	<p>A1 : <b>PRA non à jour</b> - Une procédure est obsolète à cause d'un changement d'infrastructure non documenté, ce qui complique la restauration.</p> <p>A2 : <b>Sauvegardes corrompues/inexistantes</b> - Le pire scénario, où la restauration est impossible. Le PRA doit prévoir ce cas (atténuation).</p>
Postcondition	Le service est restauré dans un état fonctionnel, en respectant les RTO et RPO définis dans le PRA.

## 9.4. Planification du Sprint

Tableau 35 : User Story de sprint 5

ID	Epic	User Story
US-26	Exploitation et Fiabilité	En tant qu'Admin, je veux configurer des sauvegardes automatiques et pouvoir lancer des sauvegardes manuelles afin de garantir une protection flexible des données.
US-27	Exploitation et Fiabilité	En tant qu'Admin, je veux disposer d'un plan de reprise d'activité (PRA) documenté afin de savoir comment réagir en cas de sinistre.

Tableau 36 : Plan d'action de sprint 5

ID User Story	Tâches Techniques à Réaliser
US-26	<ul style="list-style-type: none"> <li>- Créer des tâches de sauvegarde planifiées dans Proxmox VE pour toutes les VMs de gestion et les VMs clientes.</li> <li>- Définir une politique de rétention standard</li> <li>- Valider le bon déroulement des premières sauvegardes.</li> </ul>
US-27	<ul style="list-style-type: none"> <li>- Rédiger la première version du Plan de Reprise d'Activité (PRA)</li> </ul>

- Définir les RPO et RTO pour chaque composant critique.
- Documenter les procédures de restauration pour au moins 3 scénarios de panne (ex: perte VM unique, perte VM Odoo, perte hôte Proxmox).

## 9.5. Conception Technique : L'Architecture de la Résilience

Notre stratégie de sauvegarde, déjà initialisée au chapitre 4, repose sur l'interaction native entre Proxmox VE et Proxmox Backup Server (PBS).

- **Sauvegardes incrémentales et dédupliquées** : Proxmox VE n'envoie que les blocs de données modifiés depuis la dernière sauvegarde. PBS stocke ensuite une seule fois chaque bloc unique, ce qui réduit drastiquement l'espace de stockage nécessaire et la charge réseau.
  - **Chiffrement côté client** : Les sauvegardes peuvent être chiffrées sur Proxmox VE avant même d'être envoyées sur le réseau, garantissant que les données restent confidentielles sur le serveur de sauvegarde.
  - **Vérification d'intégrité** : PBS vérifie régulièrement l'intégrité des données stockées pour se prémunir contre la corruption silencieuse des sauvegardes.

Le diagramme de séquence suivant modélise le scénario le plus courant : la restauration d'une VM cliente unique à la suite d'une corruption.

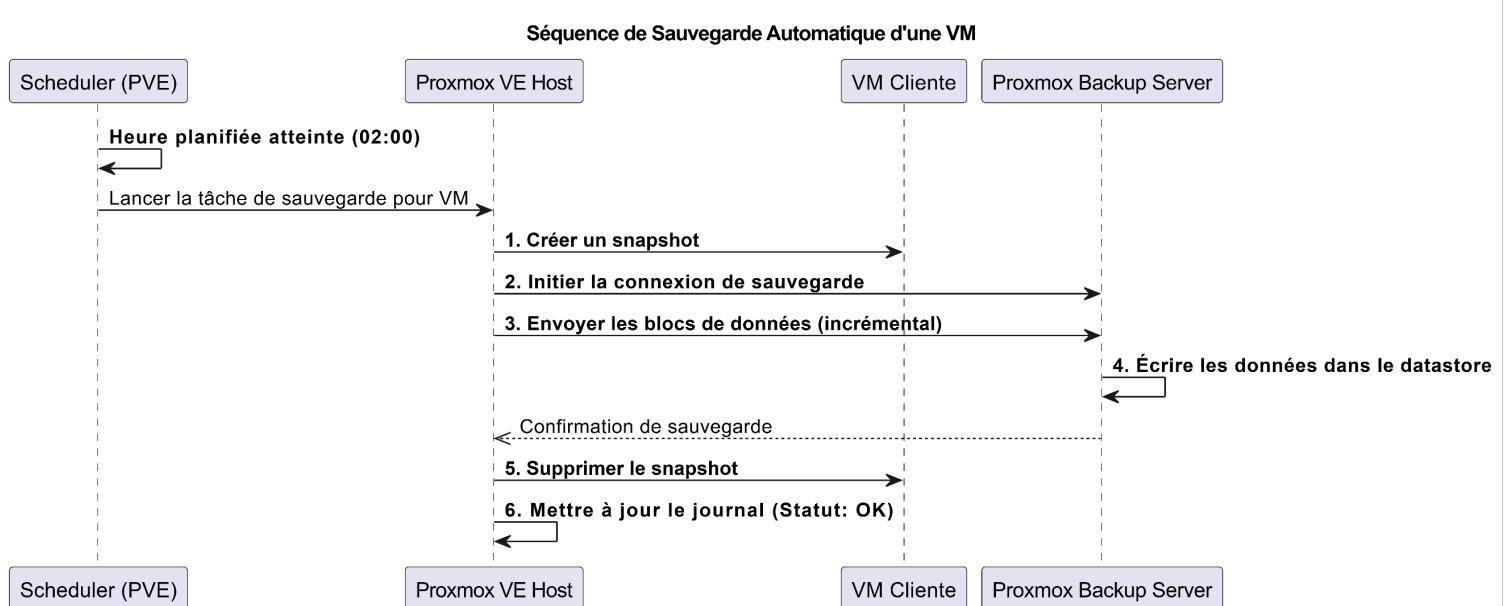


Figure 91 : Séquence - Sauvegarde Automatique d'une VM

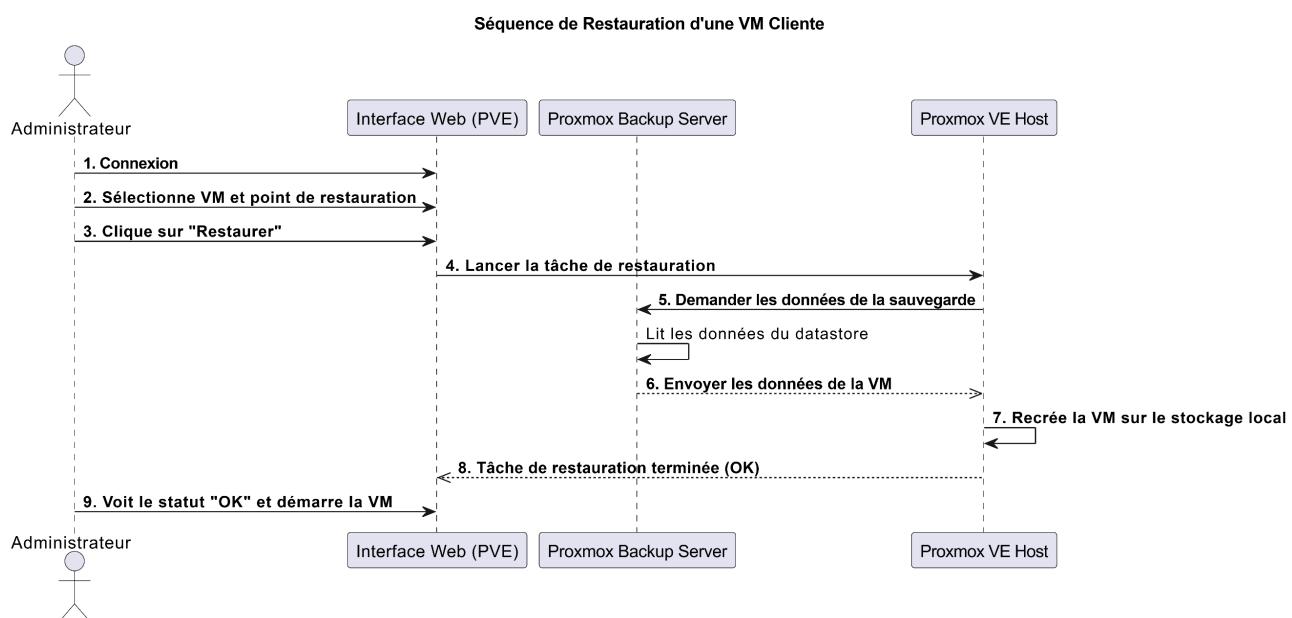


Figure 92 : Diagramme de Séquence - Restauration d'une VM Clienté

## 9.6. Réalisation

Dans l'interface de Proxmox VE, une tâche de sauvegarde a été configurée pour s'exécuter chaque samedi. Elle cible toutes les VMs et utilise le stockage pbs-nfs avec une politique de rétention stricte.

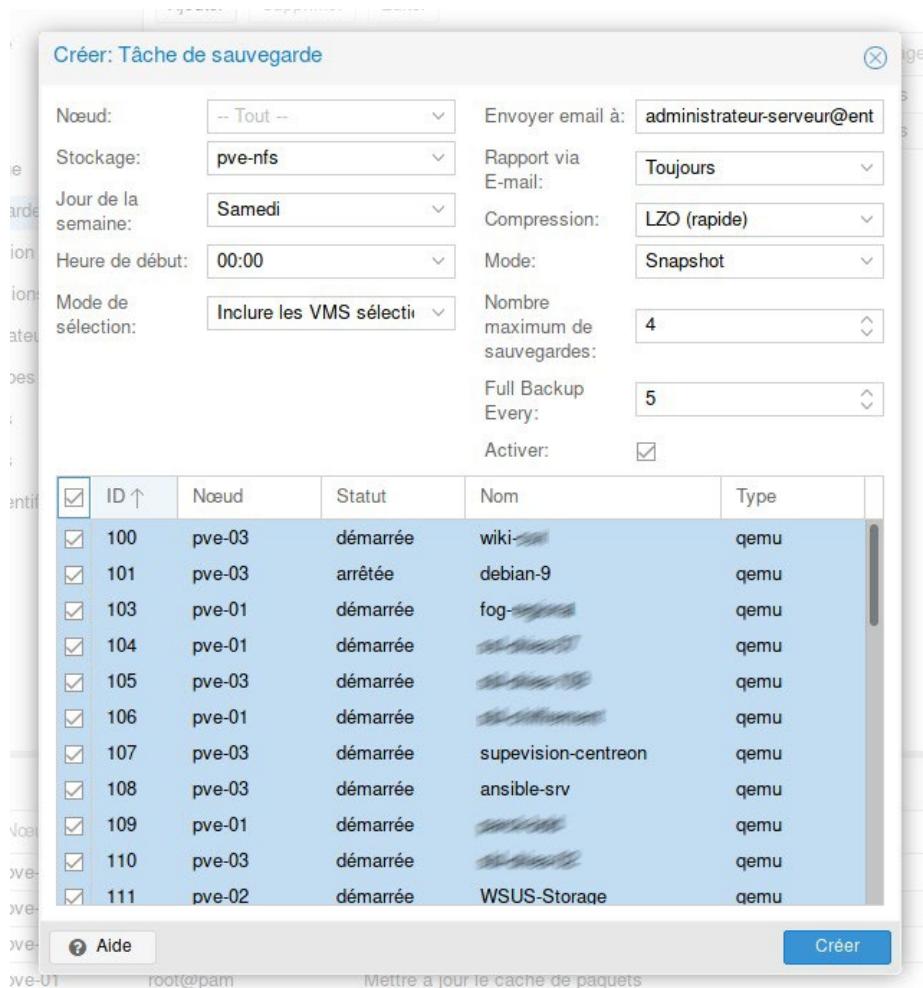


Figure 93 : Configuration de la Tâche de Sauvegarde dans Proxmox VE

Start Time ↓	End Time	Node	User name	Description	Status
May 15 16:42:43	May 15 16:44:01	pve	root@pam	VM/CT 999 - Backup	OK

Figure 94 : Journal d'une Sauvegarde Réussie vers PBS

Conformément à l'user story, le document « Plan de Reprise d'Activité (PRA) » a été rédigé et finalisé. Il constitue le livrable principal de cette tâche. Il formalise :

- **L'inventaire critique** : Serveurs, VMs de gestion, données Odoo, état Terraform.
- **Les objectifs RPO/RTO** : ex. RPO de 24h et RTO de 1-4h pour une VM cliente.
- **Les procédures détaillées** pour des scénarios clés comme :
  - Corruption / Suppression d'une VM Cliente Unique (procédure simple).
  - Panne du Serveur VM Odoo (restauration de la VM + potentiellement du dump BDD).
  - Panne Complète de l'Hôte Proxmox VE (scénario de sinistre majeur).
  - Panne du Proxmox Backup Server (le pire scénario, avec ses mesures d'atténuation).
- **Un plan de communication** interne et externe.

## 9.7. Bilan et Rétrospective du Sprint (Sprint Review)

Ce dernier sprint clôture le cycle de développement en ajoutant la dimension la plus critique pour la confiance client : la sécurité des données.

L'objectif du sprint a été pleinement atteint. La plateforme est désormais dotée :

- D'un **mécanisme de sauvegarde automatique et robuste** qui protège les environnements clients.
- D'un **Plan de Reprise d'Activité formel et documenté**, qui transforme une situation de crise potentielle en une procédure gérable.

La solution est passée d'une simple plateforme de déploiement à un service managé complet, fiable et résilient.

La principale leçon de ce sprint est qu'implémenter des sauvegardes ne représente que 50% du travail. La rédaction du PRA nous a forcés à réfléchir aux détails des procédures de restauration. Cela a mis en évidence que sans tests réguliers de ces procédures (ce qui serait une "User Story" dans un vrai contexte de production), le PRA reste un document

théorique. La vraie confiance dans la résilience du système ne vient que de la validation pratique de sa capacité à être restauré.

## 9.8. Conclusion

En conclusion, le Sprint 5 a apporté la pierre angulaire de la fiabilité à notre solution. En intégrant une stratégie de sauvegarde complète et en la formalisant dans un Plan de Reprise d'Activité, nous avons mis en place le filet de sécurité indispensable à toute offre de service professionnelle. Le cycle de vie du produit, de la commande client à la supervision et désormais à la protection contre les sinistres, est complet.

# Conclusion Générale

Ce stage au sein de MAG Security a été une opportunité d'apprentissage à plusieurs niveaux, nous permettant de transformer une vision stratégique en une solution technique concrète et fonctionnelle. Le développement de la plateforme d'allocation dynamique de ressources nous a permis d'identifier et de mettre en place les différentes exigences du système, tout en respectant le planning prévisionnel établi au début.

L'aboutissement de ce projet a été le déploiement d'une plateforme de virtualisation entièrement automatisée, faisant évoluer un processus manuel vers un modèle industriel. Cette expérience de conception et de développement nous a permis de maîtriser une variété d'outils (Odoo, Proxmox, Terraform, Ansible) et de renforcer les connaissances acquises tout au long de notre parcours académique.

L'adoption de la méthode SCRUM, avec sa répartition méthodique des tâches en sprints, nous a permis d'optimiser nos efforts et de maximiser notre efficacité. Nous avons débuté par un sprint consacré à la validation du parcours client (MVP), suivi par des sprints successifs dédiés à la sécurisation des réseaux clients, à l'automatisation complète de l'infrastructure via l'IaC, et enfin, à l'intégration des couches de monitoring et de sauvegarde.

En outre, notre plateforme se distingue par l'intégration de robustes principes d'Infrastructure as Code, rendue possible grâce à la chaîne d'outils Terraform et Ansible. En standardisant les déploiements via des templates et des scripts, le risque d'erreurs humaines est quasi nul, et l'architecture réseau garantit la confidentialité et l'intégrité des données clients.

Malgré les défis rencontrés lors de l'intégration de ces systèmes hétérogènes, ces obstacles ont contribué à notre intégration plus approfondie dans le monde professionnel, en nous confrontant à des problématiques concrètes de fiabilité et de maintenabilité.

Nous espérons avoir répondu aux attentes qui nous ont été confiées et avoir apporté une solution bénéfique à MAG Security. Dans une perspective d'amélioration continue, nous proposons d'enrichir notre application en implémentant les éléments suivants :

- Enrichissement du catalogue de services avec des templates applicatifs complexes (ex: serveur web préconfiguré, base de données managée).
- Mise en place de politiques de pare-feu avancées, gérées dynamiquement via Ansible depuis Odoo.
- Amélioration de l'expérience client par un dashboard Odoo plus riche, offrant des visualisations de monitoring simplifiées.
- Intégration d'un système de gestion des secrets (comme HashiCorp Vault) pour renforcer la sécurité des identifiants d'API.

Pour l'avenir, plusieurs axes d'amélioration pourraient être explorés pour faire évoluer cette plateforme. L'étape logique suivante consisterait à mettre en place un cluster Proxmox pour garantir la haute disponibilité (HA) et la scalabilité de l'infrastructure.

Côté client, le portail pourrait être enrichi avec des options de gestion plus avancées (snapshots, sauvegardes) et un modèle de facturation à l'usage (pay-as-you-go).

Enfin, la sécurité pourrait être renforcée par l'ajout d'un pare-feu applicatif (WAF), tandis que le pipeline d'automatisation lui-même pourrait être optimisé via une approche CI/CD pour garantir des déploiements encore plus fiables et rapides

# Webographie

- [1] Proxmox Server Solutions GmbH. (2024). *Proxmox VE 8.2 Documentation*. Disponible sur : <https://www.proxmox.com/en/downloads>
- [2] Telmate (Communauté). (2024). *Terraform Provider for Proxmox*. Disponible sur : <https://github.com/Telmate/terraform-provider-proxmox>
- [3] Odoo S.A. (2024). *Odoo 17.0 Documentation*. Disponible sur : <https://www.odoo.com/documentation/17.0/>
- [4] Odoo Community Association (OCA) (Communauté). *GitHub Repositories*. Disponible sur : <https://github.com/OCA>
- [5] HashiCorp. (2024). *Terraform Documentation*. Disponible sur : <https://developer.hashicorp.com/terraform/docs>
- [6] Red Hat, Inc. (2024). *Ansible Documentation*. Disponible sur : <https://docs.ansible.com/>
- [7] Canonical Ltd. (2024). *Cloud-Init Documentation*. Disponible sur : <https://cloudinit.readthedocs.io/>
- [8] Awesome Cloud-Init (Communauté). *A collection of examples and guides*. Disponible sur : <https://github.com/awesome-cloud-init/awesome-cloud-init>
- [9] The Prometheus Authors. (2024). *Prometheus Documentation*. Disponible sur : <https://prometheus.io/docs/>
- [10] Prometheus PVE Exporter (Communauté). *Proxmox VE Exporter for Prometheus*. Disponible sur : <https://github.com/prometheus-pve/prometheus-pve-exporter>
- [11] Grafana Labs. (2024). *Grafana Documentation*. Disponible sur : <https://grafana.com/docs/grafana/latest/>
- [12] Grafana Labs (Communauté). *Grafana Dashboards Repository*. Disponible sur : <https://grafana.com/grafana/dashboards/>
- [13] F5, Inc. (2024). *NGINX Documentation*. Disponible sur : <https://nginx.org/en/docs/>
- [14] AWS Documentation. (2024). *Linux Bastion Hosts on AWS* (Guide conceptuel). Disponible sur : <https://aws.amazon.com/quickstart/architecture/linux-bastion/>
- [15] Python Software Foundation. (2024). *Python 3.10 Documentation*. Disponible sur : <https://docs.python.org/3/>

- [16] The Debian Project. (2024). *Debian Documentation*. Disponible sur : <https://www.debian.org/doc/>
- [17] Git. (2024). *Git Documentation*. Disponible sur : <https://git-scm.com/doc>
- [18] Atlassian. (2024). *Jira Software Documentation*. Disponible sur : <https://www.atlassian.com/software/jira>
- [19] Proxmox Server Solutions GmbH. (2024). *Installation Guide for Proxmox VE*. Disponible sur : <https://pve.proxmox.com/wiki/Installation>
- [20] Proxmox Server Solutions GmbH. (2024). *Installation Guide for Proxmox Backup Server*. Disponible sur : <https://pbs.proxmox.com/docs/installation.html>
- [21] Odoo S.A. (2024). *Installing Odoo*. Disponible sur : <https://www.odoo.com/documentation/17.0/administration/install.html>
- [22] HashiCorp. (2024). *Install Terraform*. Disponible sur : <https://developer.hashicorp.com/terraform/cli/install>
- [23] Red Hat, Inc. (2024). *Ansible Installation Guide*. Disponible sur : [https://docs.ansible.com/ansible/latest/installation\\_guide/index.html](https://docs.ansible.com/ansible/latest/installation_guide/index.html)
- [24] F5, Inc. (2024). *Installing NGINX Open Source*. Disponible sur : <https://docs.nginx.com/nginx/admin-guide/installing-nginx/installing-nginx-open-source/>
- [25] Python Software Foundation. (2024). *Getting Started with Python*. Disponible sur : <https://www.python.org/about/gettingstarted/>

## Résumé

Ce projet, mené au sein de MAG Security, avait pour objectif de moderniser le déploiement de services d'hébergement en remplaçant les procédures manuelles par une plateforme de virtualisation entièrement automatisée. En intégrant l'ERP Odoo comme portail client et chef d'orchestre avec la puissante infrastructure de virtualisation Proxmox VE, nous avons créé un pipeline "Infrastructure as Code" complet. La solution finale permet à un client de commander et de personnaliser une machine virtuelle via une interface e-commerce, déclenchant un processus entièrement automatique de provisioning, de configuration réseau isolée, et de déploiement sécurisé, le tout supervisé par une stack de monitoring et protégé par des sauvegardes robustes.

**Mots-clés :** Virtualisation, Automatisation, Infrastructure as Code (IaC), Orchestration, Proxmox VE, Odoo, Terraform, Ansible, Supervision.

---

## Abstract

This project, conducted at MAG Security, aimed to modernize hosting service deployment by replacing manual procedures with a fully automated virtualization platform. By integrating the Odoo ERP as a customer portal and orchestrator with the powerful Proxmox VE virtualization infrastructure, we created a complete "Infrastructure as Code" pipeline. The final solution allows a client to order and customize a virtual machine via an e-commerce interface, triggering a fully automatic process of provisioning, isolated network configuration, and secure deployment, all monitored by a monitoring stack and protected by robust backups.

**Keywords:** Virtualization, Automation, Infrastructure as Code (IaC), Orchestration, Proxmox VE, Odoo, Terraform, Ansible, Monitoring

---

## ملخص

بهدف تحديث عملية نشر خدمات الاستضافة عبر استبدال الإجراءات اليدوية تم تنفيذ هذا المشروع في شركة MAG Security كبوابة للعملاء ومنصة لتنسيق العمليات مع البنية التحتية القوية Odoo بمنصة الافتراضية مؤتمتة بالكامل. من خلال دمج نظام ، قمنا بإنشاء مسار عمل متكامل يعتمد على مفهوم "البنية التحتية كشيفرة". يتيح الحل Proxmox VE للمحاكاة الافتراضية النهائي للعميل طلب وتخصيص خادم افتراضي عبر واجهة تجارة إلكترونية، مما يؤدي إلى إطلاق عملية تلقائية بالكامل تشمل التوفير، وإعداد شبكة معزولة، والنشر الآمن، مع ضمان المراقبة المستمرة والحماية عبر نسخ احتياطي قوي

كلمات مفتاحية:

Proxmox VE Odoo ،Terraform ،Ansible ،البنية التحتية كشيفرة ،الأمنة، البنية التحتية ،الافتراضية