NebTech®

Manuel d’utilisation

Dashboard Cloud Privé NovaTechSolutions

RELEASE 2.2

NebTech

www.nebtech.fr

Tables des mises à jours du document

Version	Date	Auteur	Description de la mise à jour
1.0	20/09/2025	RECULE Damien	Création initiale du document
1.1	22/09/2025	RECULE Damien	Ajout structure initiale et sommaire
1.2	25/09/2025	RECULE Damien	Ajout du Menu déploiement VM
1.3	02/10/2026	RECULE Damien	Ajout du chapitre Restriction de sécurité - Authentification
1.4	05/10/2026	RECULE Damien	Ajout du chapitre Interface utilisateurs
1.5	07/10/2025	RECULE Damien	Ajout du chapitre Gestion des machines virtuelles
1.6	10/10/2025	RECULE Damien	Ajout du chapitre Déploiement de solutions applicatives
1.7	17/10/2025	RECULE Damien	Ajout du chapitre Automatisation et orchestration
1.8	20/10/2025	RECULE Damien	Ajout du chapitre Supervision et journalisation
1.9	21/10/2025	RECULE Damien	Ajout du chapitre sécurité et bonnes pratiques
2.0	03/11/2025	RECULE Damien	Ajout du chapitre Dépannage et support - Annexes
2.1	07/11/2025	RECULE Damien	Relecture du document
2.2	10/11/2025	RECULE Damien	Relecture du document et livrable

Tables des illustrations

Liste des abréviations

Abréviation	Signification	Description / Contexte projet
API	Application Programming Interface	Communication entre le dashboard et Proxmox, Terraform ou autres services
CPU	Central Processing Unit	Ressources de calcul allouées aux machines virtuelles
CI/CD	Continuous Intégration Continuous Deployment	Pipeline automatisé (Jenkins) pour déployer, tester et livrer en continu.
API	Application Programming Interface	Interface utilisée par Terraform pour communiquer avec Proxmox.
VLAN	Virtual Local Area Network	Segmentation réseau (Prod, Storage, Management…).
HA	High Availability	Haute disponibilité (Proxmox + Ceph).
HDD/SSD	Hard Drive / Solid State Drive	Types de stockage des nœuds Proxmox.
RBD	RADOS Block Device	Format de disque distribué utilisé par Ceph.
OSD	Object Storage Daemon	Processus Ceph stockant physiquement les données.
TOTP	Time-based One-Time Password	Authentification forte à double facteur (2FA)
MGR	Manager	Service Ceph qui fournit la supervision et les métriques.
PBS	Proxmox Backup Server	Solution de sauvegarde utilisée pour les VM.
PVE	Proxmox Virtual Environment	Hyperviseur utilisé pour le cluster
SSH	Secure Shell	Accès sécurisé utilisé par Ansible et les administrateurs
IaC	Infrastructure as Code	Gestion de l'infrastructure via Terraform
VPN	Virtual Private network	Accès sécurisé à l'infrastructure depuis l'extérieur
VM	Virtual Machine	Ressource virtuelle déployée sur Proxmox
OS	Operating System	Systèmes d'exploitation
IP	Internet Protocol	Adressage réseau des machines virtuelles

Sommaire

Manuel d’utilisation

Chapitre 1

Présentation générale du Dashboard

1.1 Objectifs du Dashboard

Le Dashboard Cloud Privé a été développé afin de centraliser, simplifier et sécuriser la gestion de l'infrastructure virtualisée de NovaTechSolutions.

Il répond aux objectifs suivants :

- Offrir une interface unique pour piloter le cloud privé sans accéder directement aux outils sous-jacents.
- Réduire les erreurs humaines grâce à l'automatisation des actions critiques.
- Accélérer le déploiement des machines virtuelles et des services applicatifs.
- Améliorer la visibilité sur l'état de l'infrastructure (VM, cluster, ressources).
- Renforcer la sécurité des accès administratifs via une authentification forte.

Le Dashboard s'inscrit dans une démarche d'industrialisation des opérations IT, conforme aux bonnes pratiques DevOps et aux recommandations de l'ANSSI.

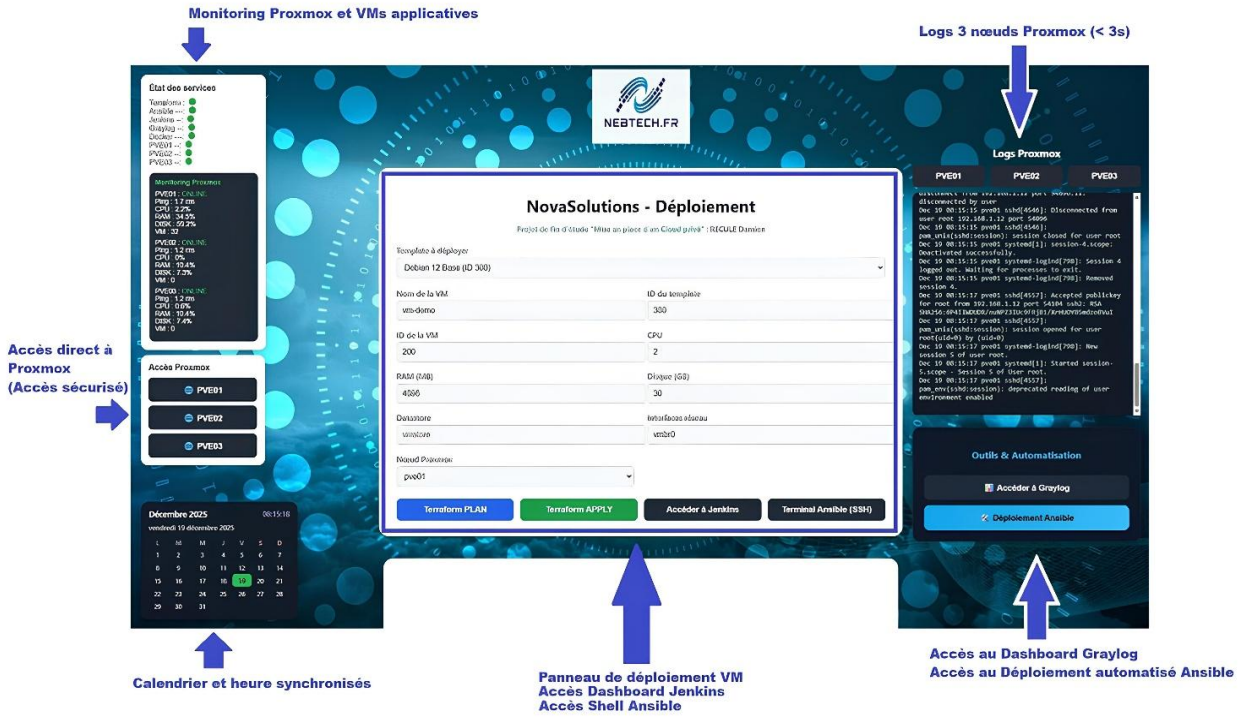


Figure 1 Dashboard - Page d'Accueil

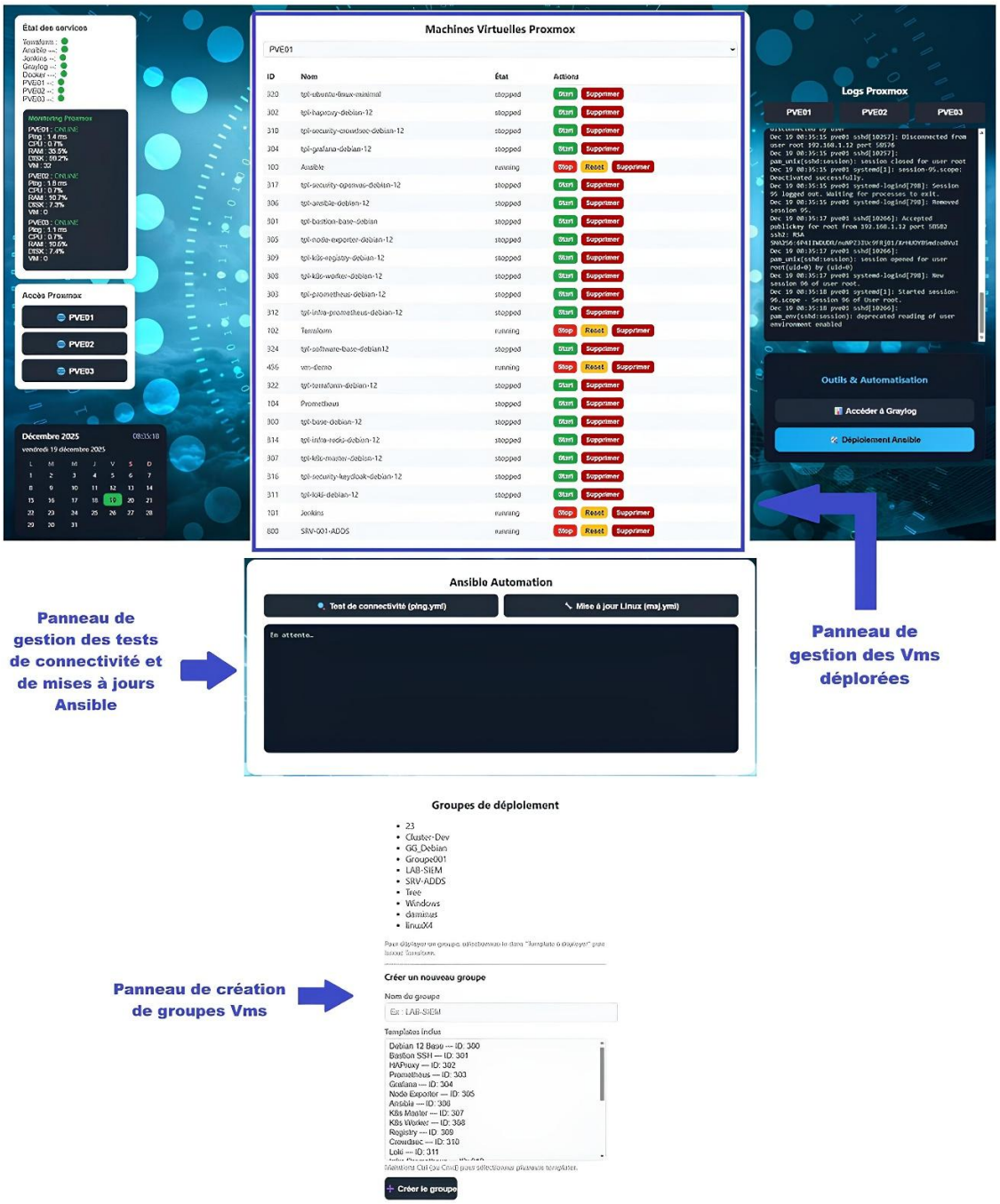


Figure 2 Dashboard - Menus complémentaires

1.2 Rôle du Dashboard dans le cloud privé

Le Dashboard joue le rôle de point d'entrée unique entre les utilisateurs et l'infrastructure cloud privée.

Il agit comme une couche d'orchestration et d'abstraction, reliant :

- L'utilisateur final (administrateur ou technicien).

- Les outils d'automatisation (Terraform, Ansible, Jenkins).
- L'infrastructure de virtualisation (Proxmox, Ceph).
- Les mécanismes de supervision et de journalisation.

Ainsi, l'utilisateur n'interagit jamais directement avec :

- L'API Proxmox.
- Les fichiers Terraform.
- Les playbooks Ansible.
- Les pipelines Jenkins.

Toutes les actions sont déclenchées et contrôlées depuis le dashboard.

Figure 3 Dashboard - Menu déploiement VM

1.3 Fonctionnalités principales

Le dashboard Cloud Privé permet notamment :

- L'authentification sécurisée des utilisateurs (HTTPS + 2FA).
- La création automatisée de machines virtuelles via Terraform.
- La gestion du cycle de vie des VM (démarrage, arrêt, suppression).
- Le déploiement automatisé de solutions applicatives via Ansible.
- L'orchestration des tâches à l'aide de Jenkins.
- La consultation des journaux et des états d'exécution.
- La supervision globale de l'infrastructure.

L'ensemble des fonctionnalités est accessible via une interface web ergonomique, pensée pour être utilisable par des profils non experts en virtualisation ou en Infrastructure as Code.

1.4 Périmètre fonctionnel et limites

Le dashboard couvre exclusivement les opérations prévues et validées dans le cadre du projet.
Il ne permet pas :

- La modification directe de la configuration Proxmox.
- L'édition manuelle des fichiers Terraform ou Ansible.
- L'exécution de commandes système arbitraires.
- La gestion avancée du réseau ou du stockage bas niveau.

Ces limitations sont volontaires et visent à :

- Garantir la sécurité de l'infrastructure.
- Préserver la cohérence des déploiements.
- Éviter toute manipulation non maîtrisée.

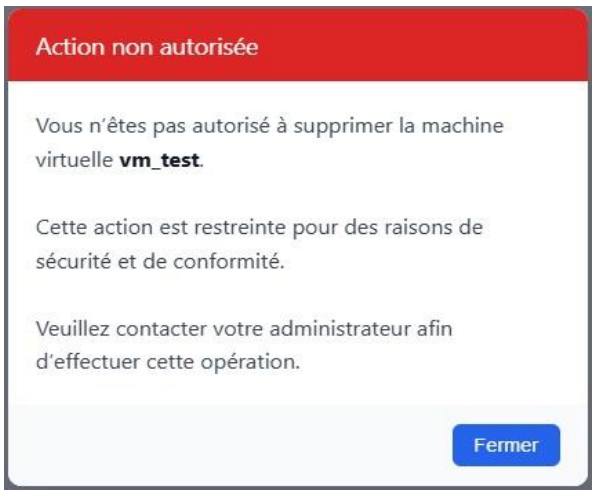


Figure 4 Dashboard - Restriction de sécurité

Chapitre 2

Accès au dashboard

2.1 Prérequis techniques

Avant d'accéder au dashboard, l'utilisateur doit s'assurer que les conditions techniques suivantes sont réunies.

Navigateurs compatibles

Le dashboard est accessible via un navigateur web moderne supportant les standards HTML5 et HTTPS.

Navigateurs recommandés :

- Microsoft Edge.
- Google Chrome (version récente).
- Mozilla Firefox.
- Microsoft Edge.

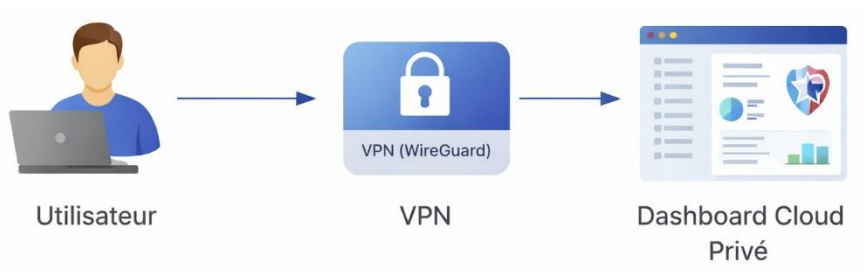
⚠ L'utilisation d'anciens navigateurs ou de navigateurs non maintenus peut entraîner des dysfonctionnements d'affichage ou de sécurité.

Accès réseau

L'accès au dashboard est restreint au réseau interne de l'entreprise ou via un accès distant sécurisé.

Selon le contexte :

- Accès direct depuis le réseau interne NovaTechSolutions.
- Accès distant via VPN sécurisé (Wireguard).



2.2 Accès sécurisé en HTTPS

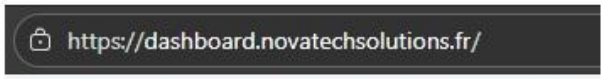
Toutes les connexions au dashboard sont protégées par le protocole HTTPS afin de garantir :

- La confidentialité des échanges.
- L'intégrité des données transmises.
- L'authenticité du serveur.

Le dashboard utilise un certificat TLS valide, empêchant toute interception ou modification des flux réseau.

L'URL d'accès prend la forme suivante :

https://dashboard.novatechsolutions.fr



2.3 Authentification utilisateur

L'accès au dashboard nécessite une authentification utilisateur préalable.

Chaque utilisateur dispose :

- D'un identifiant unique.
- D'un mot de passe personnel.

Lors de la connexion :

1. L'utilisateur saisit son identifiant
2. Il saisit son mot de passe
3. Le système vérifie les informations avant d'autoriser l'accès

Les tentatives d'authentification sont journalisées afin de renforcer la traçabilité et la sécurité.

Figure 5 Dashboard – Authentification

2.4 Authentification forte (2FA – TOTP)

Afin de renforcer la sécurité des accès, le dashboard intègre une authentification forte à deux facteurs (2FA) basée sur le protocole TOTP.

Principe

En complément du mot de passe, l'utilisateur doit fournir un code temporaire généré par une application d'authentification.

Applications compatibles :

- Google Authenticator.
- Microsoft Authenticator.
- FreeOTP.

Le code est :

- Temporaire (valide quelques secondes).
- Unique.
- Non réutilisable.

Ce mécanisme permet de protéger l'accès même en cas de compromission du mot de passe.

Figure 6 Dashboard - Autorisation QR-code

2.5 Gestion des droits et profils utilisateurs

Le dashboard applique une gestion fine des droits basée sur les profils utilisateurs.

Profils disponibles (exemples)

- Administrateur : accès complet (déploiement, suppression, supervision).
- Technicien : déploiement et supervision sans suppression critique.
- Lecture seule : consultation de l'état de l'infrastructure.

Les droits définissent :

- Les actions autorisées.
- Les ressources accessibles.
- Les restrictions applicatives (ex : suppression de VM protégée).

Lorsqu'un utilisateur tente une action non autorisée, un message explicite s'affiche afin de prévenir toute erreur de manipulation.

Figure 7 Dashboard - Message Action non autorisée

Chapitre 3

Interface Utilisateur

3.1 Page d'accueil et tableau de bord principal

Description

La page d'accueil constitue le point d'entrée principal du dashboard. Elle offre une vue synthétique de l'état de l'infrastructure et un accès rapide aux fonctionnalités essentielles.

Elle regroupe notamment :

- L'état global du cluster Proxmox.
- Le nombre de machines virtuelles actives.
- L'état du stockage (Ceph).
- Les alertes et événements récents.
- Les raccourcis vers les actions courantes.

Cette approche permet à l'utilisateur d'évaluer rapidement la situation sans parcourir plusieurs écrans.

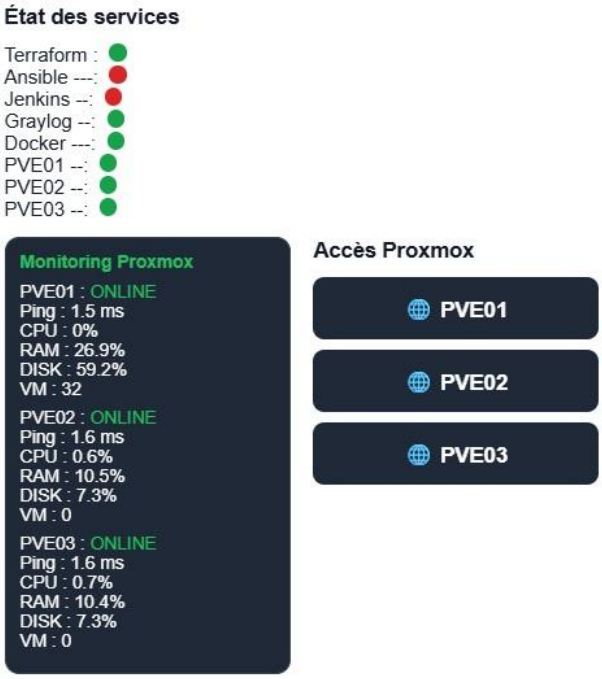


Figure 8 Dashboard - Monitoring - Cluster Proxmox

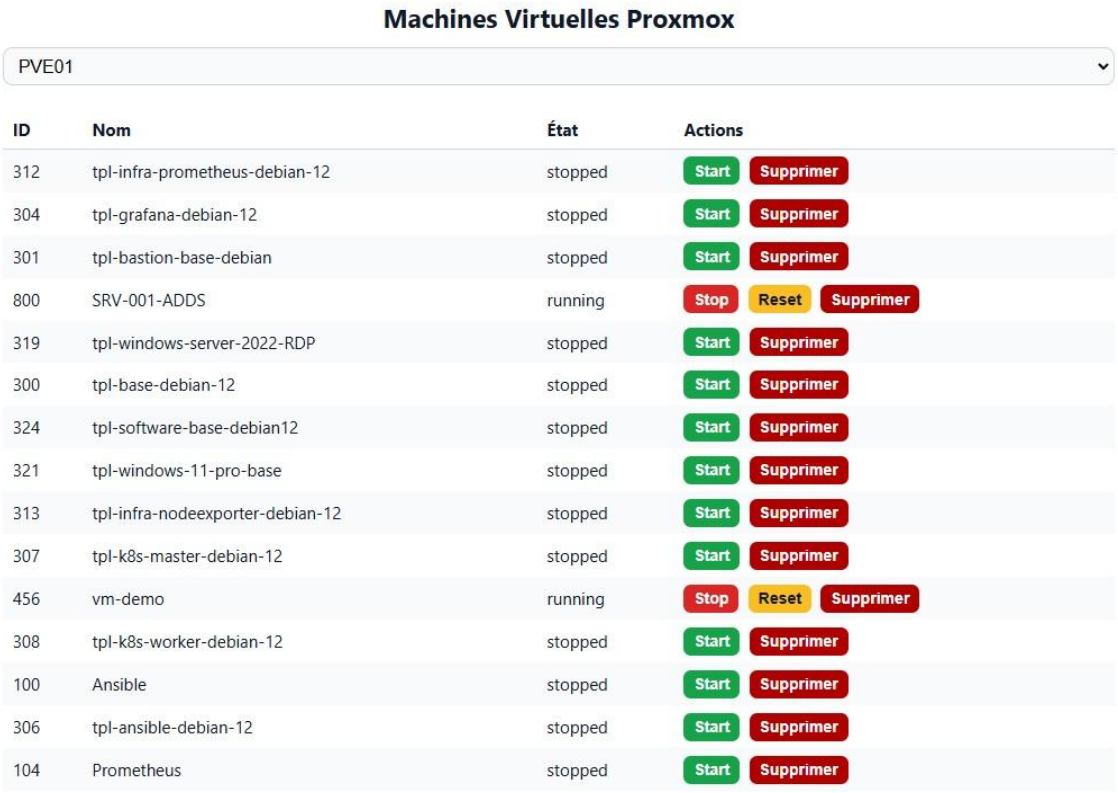


Figure 9 Dashboard – Liste des VMs déployées

3.2 Navigation et menus

Description

La navigation repose sur une structure simple et cohérente, conçue pour limiter les erreurs de manipulation et améliorer l'expérience utilisateur.

Les menus permettent d'accéder aux principales sections :

- Gestion des machines virtuelles.
- Déploiement automatisé.
- Supervision et logs.
- Paramètres et administration.

La navigation est persistante afin que l'utilisateur sache toujours où il se trouve dans l'interface.

Points clés

- Hiérarchie claire des menus.
- Accès rapide aux actions fréquentes.
- Séparation entre fonctions utilisateur et administrateur.
- Sections clairement identifiées.

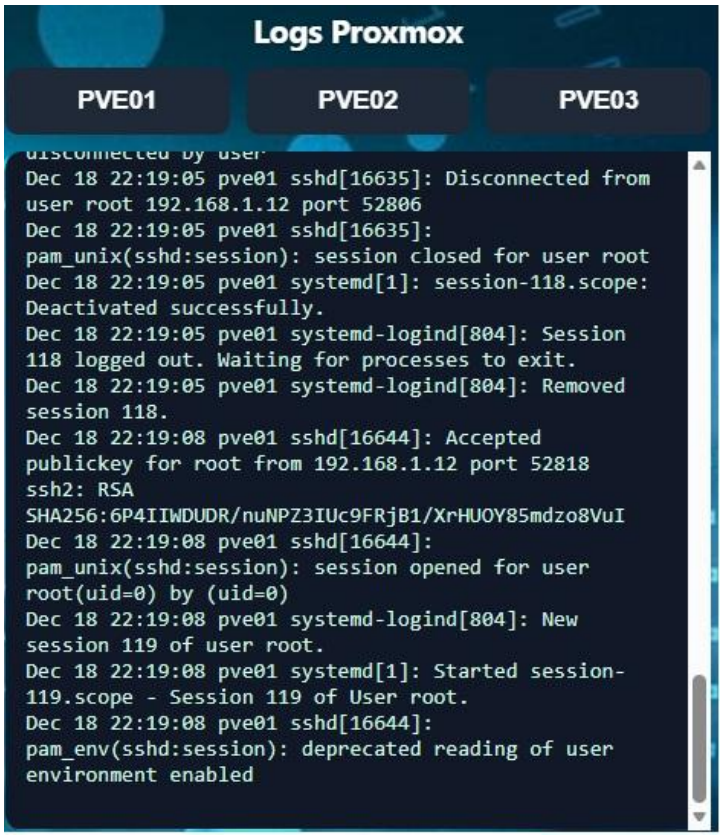


Figure 10 Dashboard - Logs Cluster Proxmox

3.3 Indicateurs et statuts affichés

Description

Le dashboard met en avant des indicateurs visuels permettant de suivre l'état de l'infrastructure en temps réel.

Les statuts sont présentés de manière lisible à l'aide de :

- Couleurs normalisées (vert / orange / rouge).
- Icônes explicites.
- Libellés courts et compréhensibles.

Exemples d'indicateurs :

- Etat des VM (démarrée, arrêtée, en erreur).
- Disponibilité du cluster.
- Etat du stockage Ceph.
- Charge ou alertes système.

Cette représentation visuelle facilite la prise de décision et permet une détection rapide des anomalies.

3.4 Messages système et notifications

Description

Le dashboard intègre un système de messages permettant d'informer l'utilisateur lors des actions effectuées ou en cas d'erreur.

Ces messages sont affichés de manière explicite afin d'éviter toute ambiguïté.

Types de messages :

- Confirmation d'action réussie.
- Avertissement (action sensible).
- Erreur ou action non autorisée.

Les notifications contribuent à la sécurité en empêchant les actions non prévues et en guidant l'utilisateur.

Exemple

Lorsqu'un utilisateur tente une action non autorisée (suppression d'une VM protégée), un message clair s'affiche l'invitant à contacter un administrateur.

Figure 11 Dashboard - Contacter Administrateur

Chapitre 4

Vision fonctionnelle et cycle de vie des machines virtuelles

4.1 Rôle et objectifs des machines virtuelles

Les machines virtuelles (VM) constituent l'élément central de l'infrastructure du cloud privé. Elles permettent de fournir des environnements isolés, reproductibles et flexibles, adaptés aux besoins applicatifs et métiers.

L'utilisation des VM répond à plusieurs objectifs :

- **Isolation des environnements** : chaque service ou application est déployé dans une VM dédiée, limitant les impacts en cas de panne, de mauvaise configuration ou de compromission.
- **Scalabilité** : les ressources d'une VM peuvent être ajustées dynamiquement (CPU, mémoire, stockage) sans intervention matérielle.
- **Standardisation** : l'utilisation de modèles (templates) garantit que toutes les VM respectent une base système homogène et conforme aux bonnes pratiques de l'infrastructure.
- **Automatisation** : la création et la gestion des VM sont intégrées dans des processus automatisés (Terraform, Ansible), réduisant les erreurs humaines et les délais de mise en production.

Dans ce contexte, les templates jouent un rôle clé : ils servent de point de départ fiable pour toutes les VM, assurant cohérence, sécurité et rapidité de déploiement.

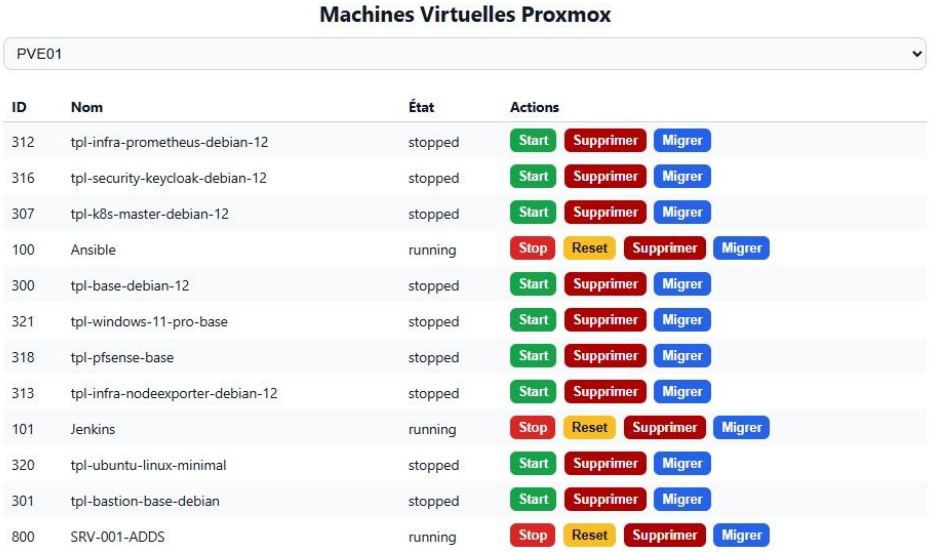


Figure 12 Vue d'ensemble des machines virtuelles du cloud privé (Détails)

4.2 Composants d'une machine virtuelle

Chaque machine virtuelle est composée de plusieurs éléments techniques, configurables lors de sa création et ajustables durant son cycle de vie.

- **Processeur (CPU) :** La VM dispose d'un nombre défini de processeurs virtuels (vCPU). Ceux-ci représentent une portion des ressources CPU physiques de l'hyperviseur. L'allocation peut être adaptée en fonction de la charge attendue de la VM.

- **Mémoire (RAM) :** La mémoire vive attribuée à la VM conditionne directement ses performances et sa capacité à exécuter des applications. Une allocation correcte permet d'assurer la stabilité du système tout en optimisant l'utilisation des ressources globales.

Figure 13 Fenêtre de création de VM (Paramètres techniques)

Le stockage est généralement réparti en :

- **Disque système** : contient le système d'exploitation et les composants de base.
- **Disques de données** : dédiés aux applications, bases de données ou fichiers persistants.

La connectivité réseau de la VM comprend :

- Une adresse IP (statique ou dynamique)
- L'appartenance à un réseau ou VLAN spécifique
- Des règles de sécurité (pare-feu, filtrage) appliquées au niveau de l'infrastructure

Ces composants constituent la base technique sur laquelle reposent les services déployés.

4.3 États possibles d'une machine virtuelle

Une machine virtuelle peut se trouver dans plusieurs états tout au long de son existence. Ces états permettent de suivre son fonctionnement et de détecter d'éventuels problèmes.

- **Arrêtée** : la VM existe mais ne consomme pas de ressources CPU ou mémoire.
- **Démarrée** : la VM est active et opérationnelle.

- **En cours de déploiement** : la VM est en phase de création ou de configuration automatique.
- **En erreur** : une anomalie est survenue lors du déploiement, du démarrage ou de l'exécution.
- **Supprimée** : la VM a été retirée de l'infrastructure et ses ressources libérées.

La gestion de ces états est essentielle pour l'exploitation et la supervision de l'infrastructure.

4.4 Cycle de vie d'une machine virtuelle

Le cycle de vie d'une machine virtuelle s'inscrit dans une logique entièrement automatisée et maîtrisée :

1. **Création à partir d'un template**
   Une VM est instanciée depuis un modèle standardisé garantissant une base système saine et sécurisée.

2. **Phase d'exploitation**
   La VM héberge un ou plusieurs services et peut être configurée automatiquement via Ansible.

3. **Évolutions et maintenance**
   Les ressources ou les services peuvent être ajustés sans recréer la VM.

4. **Suppression**
   Une fois la VM devenue inutile, elle est supprimée afin de libérer les ressources.

Figure 14 Cycle de vie d'une VM dans l'infrastructure

Terraform intervient principalement dans la gestion de l'infrastructure (création, modification, suppression), tandis qu'Ansible assure la configuration logicielle et applicative.
Cette séparation garantit un cycle de vie clair, cohérent et automatisé.

4.5 Rôle et fonctionnement des templates

Les templates sont des images de référence utilisées pour la création des machines virtuelles. Ils permettent :

- **Centralisation** : un point unique de définition des configurations de base.
- **Standardisation** : toutes les VM partagent la même structure système initiale.
- **Sécurité** : intégration de paramètres de durcissement (SSH, utilisateurs, accès).

- **Reproductibilité** : création rapide et fiable de nouvelles VM, sans configuration manuelle.

Grâce aux templates, l'infrastructure reste cohérente et facile à maintenir, même lors de déploiements massifs.

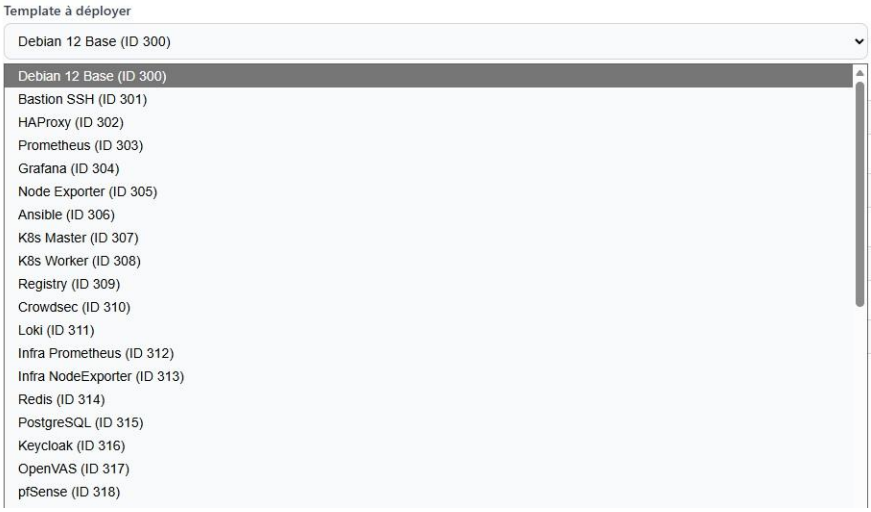


Figure 15 Détail des templates disponibles

4.6 Bonnes pratiques d'utilisation

Afin de garantir la stabilité et la performance du cloud privé, plusieurs bonnes pratiques sont recommandées :

- Éviter le surdimensionnement des ressources (CPU, RAM, stockage).
- Supprimer les machines virtuelles inutilisées afin de libérer les ressources.
- Utiliser exclusivement les templates validés et maintenus.
- Privilégier les déploiements applicatifs via Ansible plutôt que des installations manuelles.
- Surveiller régulièrement l'état et l'utilisation des VM.

Le respect de ces bonnes pratiques contribue à une infrastructure fiable, sécurisée et pérenne.

Chapitre 5

Modèles opératoires (procédures)

5.1 Créer, déployer et migrer une VM unique

Objectif

Créer et déployer une machine virtuelle unique à partir d'un template Proxmox via le dashboard, sans intervention directe sur l'hyperviseur.

Prérequis

- Être authentifié sur le dashboard.
- Disposer des droits de création de VM.

Procédure

1. Accéder au menu Création de VM
2. Choisir le template souhaité
3. Renseigner les paramètres (nom, ressources, réseau)
4. Vérifier les paramètres avec **Terraform Plan**
5. Valider la création avec **Terraform Apply**
6. Suivre l'état du déploiement via le retour visuel du dashboard

Figure 16 Choix du template à déployer

Figure 17 Déployer une VM unique

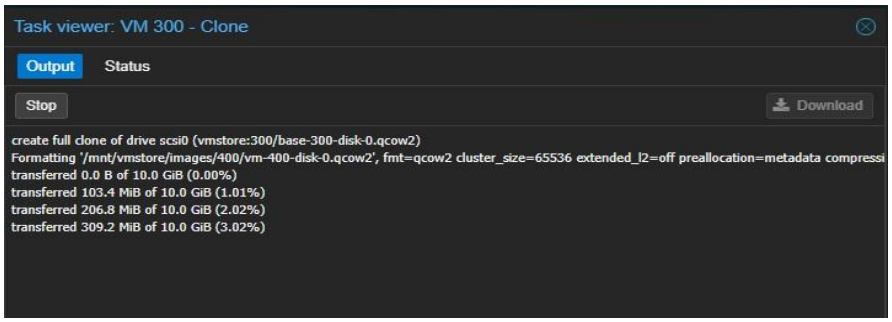


Figure 18 Etat du déploiement

Résultat attendu
La VM est créée, démarrée et visible dans l'interface avec son état opérationnel.

Migration des machines virtuelles

La migration d'une machine virtuelle consiste à déplacer celle-ci d'un hôte physique vers un autre, sans altérer son fonctionnement ni ses données. Elle permet d'optimiser l'utilisation des ressources, d'assurer la maintenance des serveurs ou d'améliorer la résilience de l'infrastructure.

On distingue principalement deux types de migration : la migration à froid et la migration à chaud.

Migration à froid

Principe

La migration à froid est réalisée lorsque la machine virtuelle est arrêtée.
La VM n'exécute aucun service durant l'opération, ce qui garantit une migration simple et sans risque d'incohérence.

Avantages

- Procédure fiable et sécurisée.
- Aucune contrainte de synchronisation en temps réel.
- Compatible avec toutes les VM et tous les hyperviseurs.

Inconvénients

- Interruption de service pendant la durée de la migration.

Cas d'usage

- Maintenance planifiée.
- Environnements de test ou de développement.
- Migration d'infrastructures non critiques.

Déroulement

1. Arrêt de la machine virtuelle
2. Copie des fichiers disque et de configuration vers le nouvel hôte
3. Reconfiguration réseau si nécessaire
4. Redémarrage de la VM sur le nouvel hôte

Migration à chaud

Principe

La migration à chaud permet de déplacer une machine virtuelle sans interruption de service.
La mémoire et l'état de la VM sont copiés en continu vers le nouvel hôte pendant son fonctionnement.

Avantages

- Continuité de service.
- Aucune coupure visible pour l'utilisateur final.
- Idéal pour les environnements de production.

Inconvénients

- Plus complexe techniquement.
- Dépend des capacités de l'hyperviseur et du stockage partagé.
- Peut générer une charge réseau temporaire.

Cas d'usage

- Maintenance non planifiée.
- Rééquilibrage de charge.
- Infrastructures critiques (production)

Déroulement

1. Synchronisation progressive de la mémoire et de l'état de la VM
2. Transfert final quasi instantané
3. Bascule automatique vers le nouvel hôte
4. Reprise immédiate de l'exécution sur la nouvelle plateforme

Mise en œuvre de la migration dans l'infrastructure

Dans le cadre de ce dashboard :

- La sélection de la machine virtuelle à migrer se fait depuis l'interface graphique.
- Le type de migration (à froid ou à chaud) est choisi en fonction de l'état de la VM et du contexte.
- Le dashboard orchestre l'opération via les outils sous-jacents (hyperviseur, automatisation).
- L'état de la migration est suivi en temps réel avec un retour d'état clair.

Les migrations à froid sont privilégiées pour les opérations planifiées, tandis que les migrations à chaud sont utilisées pour garantir la continuité de service.

Bonnes pratiques

- Prévoir les migrations à froid hors des heures de production.
- Vérifier la compatibilité des hôtes avant migration.
- Surveiller l'utilisation réseau lors des migrations à chaud.
- Tester les migrations sur des environnements non critiques.

Pour effectuer une migration manuelle, il suffit d'utiliser l'option **Migrer** prévu à cette effet et de renseigner le nœud cible.

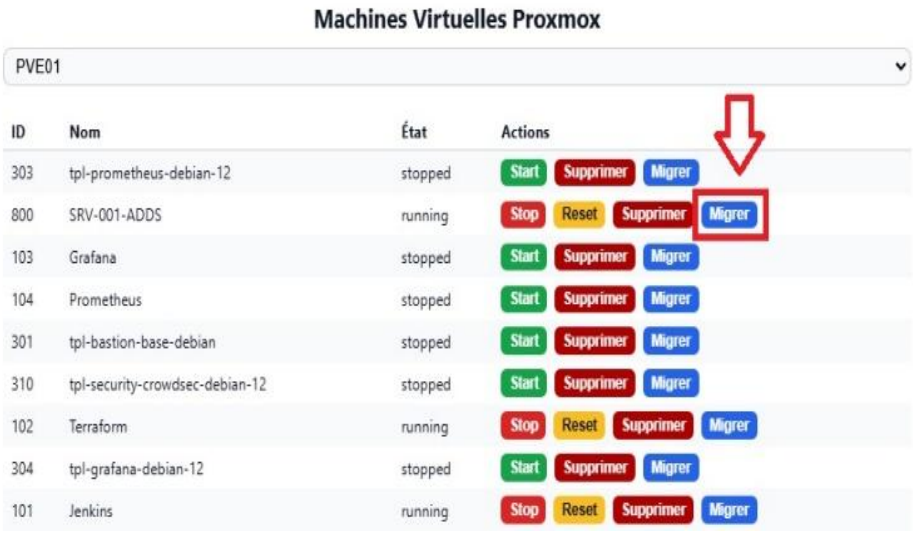


Figure 19 Migration VM – Choix de la VM

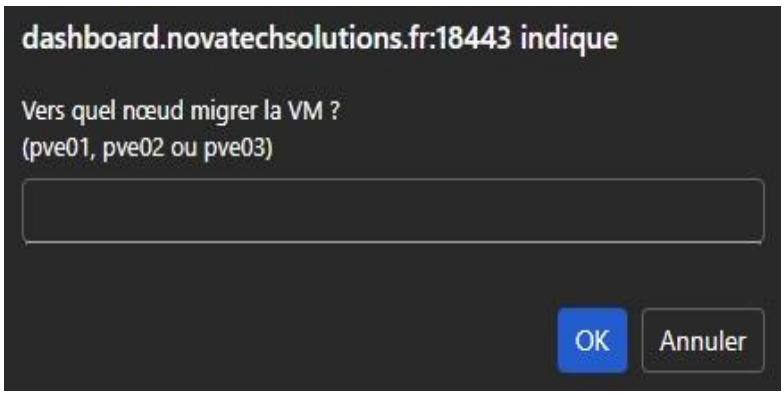


Figure 20 Migration VM - Choix du nœud cible

5.2 Créer un groupe de VM et le déployer

Objectif

Définir un groupe logique de machines virtuelles pouvant être déployées ensemble.

Prérequis

- Accès au module de gestion des groupes.
- Templates disponibles.

Procédure

1. Accéder au menu Groupes de VM.
2. Cliquer sur Créer un groupe.
3. Donner un nom explicite au groupe.
4. Ajouter les Templates souhaitées.
5. Enregistrer le groupe.
6. Depuis le menu Création VM, choisir le groupe.
7. Renseigner les paramètres (nom, ressources, réseau).

   => L'ID choisi sera le point de départ des VM déployées.
8. Vérifier les paramètres avec **Terraform Plan**.
9. Valider la création avec **Terraform Apply**.
10. Suivre l'état du déploiement via le retour visuel du dashboard.
11. Vérifier la création de la VM sur le nœud **Proxmox** PVE.

Figure 21 Menu de création de groupe de VMs

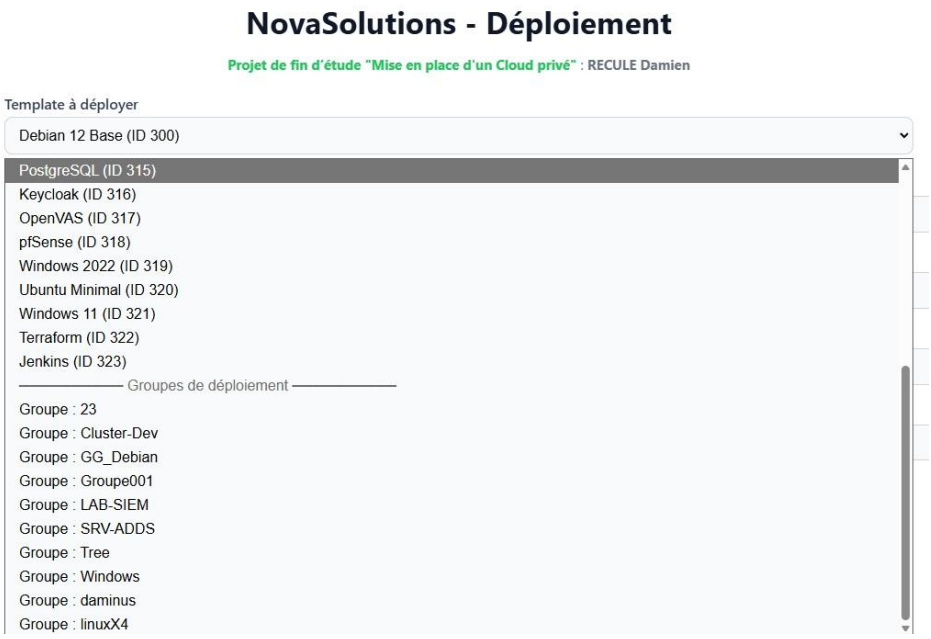


Figure 22 Choix du groupe

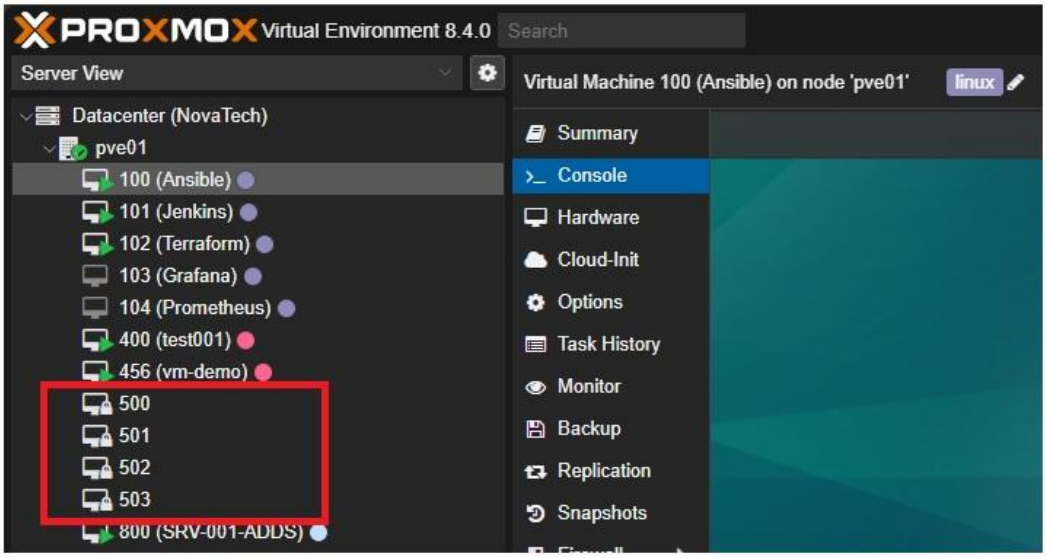


Figure 23 Vérification du déploiement en cours sur Proxmox

Start Time ↓	End Time	Node	User name	Description
Dec 20 08:29:31		pve01	root@pam	VM 303 - Clone
Dec 20 08:29:31		pve01	root@pam	VM 300 - Clone
Dec 20 08:29:31		pve01	root@pam	VM 304 - Clone
Dec 20 08:29:31		pve01	root@pam	VM 301 - Clone

Figure 24 Cluster Logs Proxmox

Résultat attendu

Le groupe est enregistré et prêt à être déployé ultérieurement.

5.3 Lancer un playbook Ansible

Objectif
Appliquer une configuration ou installer des services sur une VM via Ansible.

Prérequis

- VM accessible
- Playbook disponible
- Accès Ansible configuré

Procédure

1. Accéder au menu Ansible
2. Sélectionner la VM cible
3. Choisir le playbook souhaité
4. Lancer l'exécution
5. Consulter le retour d'exécution dans le dashboard

Figure 25 Menu de déploiement - Groupe VMs

Résultat attendu

Le playbook s'exécute avec succès ou retourne un message d'erreur exploitable.

5.4 Accéder au Shell Ansible via navigateur

Objectif
Accéder à un terminal Ansible directement depuis le navigateur web.

Prérequis

- Droits administrateur
- Accès réseau autorisé
- Identifiant de connexion administrateur

Procédure

1. Depuis le Dashboard, sélectionner le menu **Terminal Ansible.**
2. Vous êtes automatiquement redirigé vers un terminal via un navigateur Web.
3. Se connecter avec des identifiants administrateur.
4. Exécuter les commandes nécessaires.
5. Quitter la session une fois l'opération terminée.

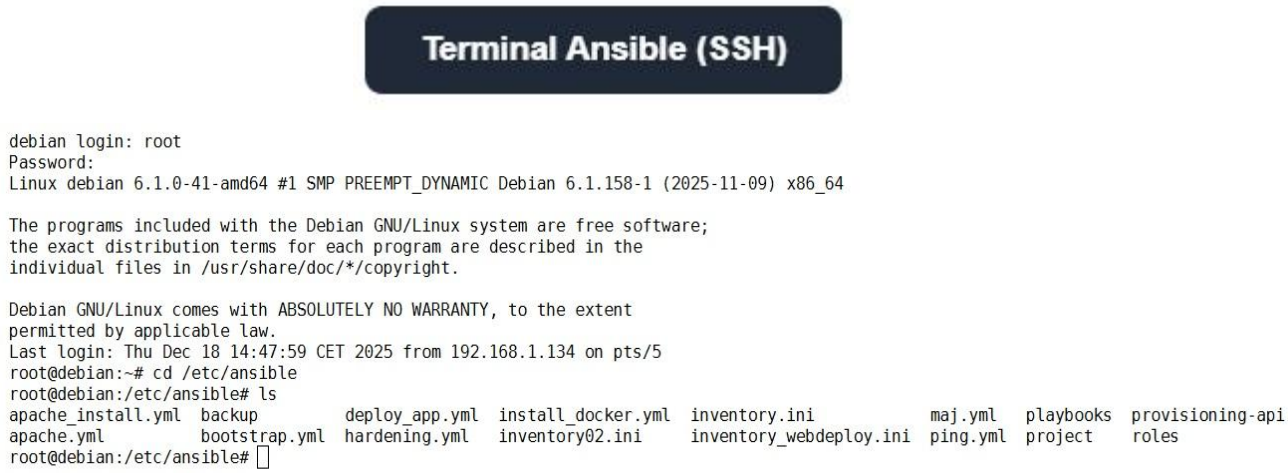


Figure 26 Accès au terminal Ansible

Résultat attendu

Ansible est accessible par les administrateurs pour création ou modification de playbook directement depuis le Dashboard.

5.5 Accéder à Graylog

Objectif
Consulter les logs centralisés pour l'analyse et la sécurité.

Prérequis

- Graylog opérationnel
- Droits d'accès

Procédure

1. Depuis le Dashboard, sélectionner le menu **Accéder à Graylog.**
2. Vous êtes automatiquement redirigé vers le dashboard Graylog via un navigateur Web.
3. Se connecter avec des identifiants administrateur.
4. Consulter les flux et alertes.

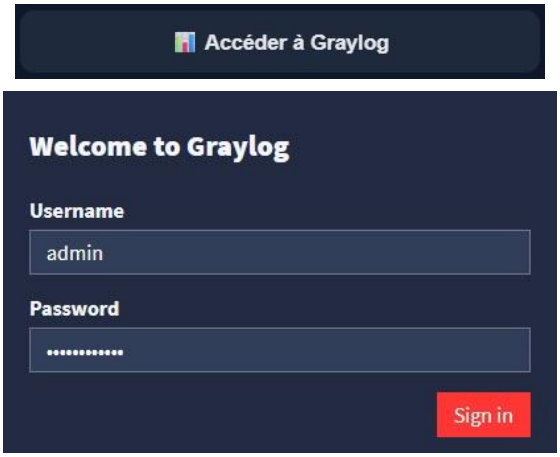


Figure 27 Fenêtre de connexion Graylog

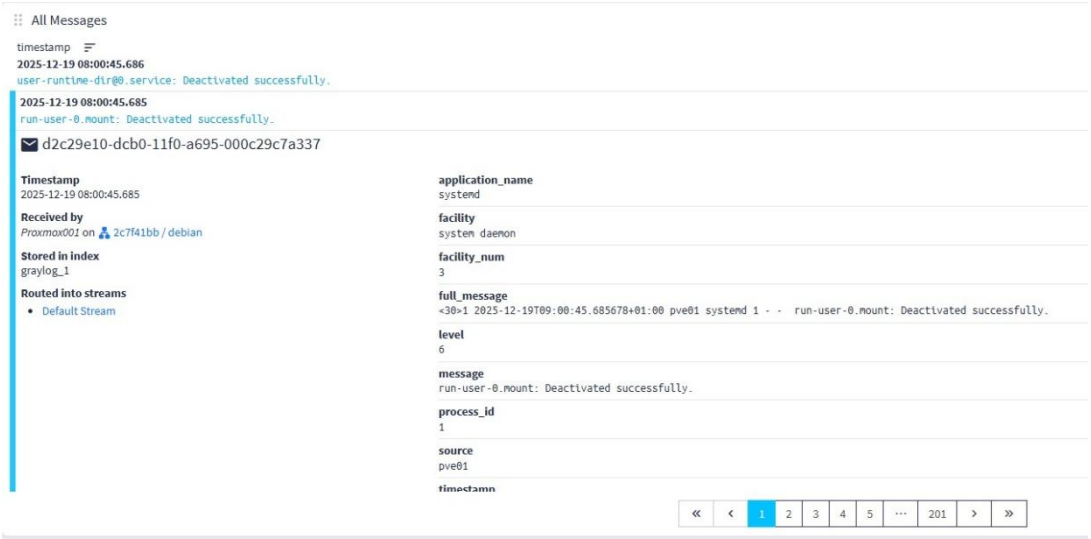


Figure 28 Dashboard Graylog

Résultat attendu

Les journaux sont consultables en temps réel.

5.6 Accéder à Jenkins

Objectif

Accéder à l'outil Jenkins depuis le dashboard afin de superviser et déclencher les pipelines CI/CD liés au projet.

Prérequis

- Jenkins opérationnel
- Droits d'accès Jenkins
- Accès au dashboard

Procédure

1. Depuis le Dashboard, Accéder au menu **Outils & Automatisations.**
2. Cliquer sur **Accéder à Jenkins.**
3. Vous êtes redirigé vers l'interface Jenkins.
4. S'authentifier avec des identifiants administrateur.

Figure 29 Fenêtre de connexion Jenkins

Figure 30 Dashboard Jenkins

Résultat attendu

L'administrateur accède à Jenkins.

Chapitre 6

Déploiement de solutions applicatives

6.1 Principe de déploiement automatisé

Description

Le déploiement applicatif repose sur l'exécution de playbooks Ansible prédéfinis et validés par l'administrateur.
Ces playbooks permettent d'automatiser l'installation, la configuration et le durcissement des services applicatifs.

Le processus suit les étapes suivantes :

1. Sélection d'une machine virtuelle cible.
2. Choix de la solution applicative.
3. Exécution du playbook Ansible correspondant.
4. Retour d'état et journalisation.

Figure 31 Menu de déploiement Ansible

Avantages

- Standardisation des déploiements.
- Réduction des erreurs humaines.
- Gain de temps opérationnel.
- Traçabilité des actions.

6.2 Sélection d'une solution applicative

Description

Depuis l'interface du dashboard, l'utilisateur peut sélectionner une solution applicative parmi un catalogue prédéfini.

Exemples de solutions disponibles :

- Serveur web (Apache, Nginx).
- Services systèmes (Docker, Docker Compose, Kubernetes).
- Outils de supervision ou de journalisation.
- Environnements applicatifs spécifiques.

Seules les solutions validées et sécurisées par l'équipe IT sont proposées afin de garantir la cohérence et la sécurité de l'infrastructure.

Catalogue des solutions applicatives

- Liste déroulante ou cartes.
- Description courte de chaque solution.
- Indication de la VM cible.

Figure 32 Choix de solutions - Menu Ansible

6.3 Lancement du déploiement via Ansible

Description

Une fois la solution sélectionnée, l'utilisateur déclenche le déploiement depuis le dashboard.
Le dashboard exécute alors automatiquement le playbook Ansible correspondant.

L'exécution s'effectue :

- Via une connexion sécurisée.
- Avec des paramètres prédéfinis.
- Sans accès direct à la VM par l'utilisateur.

Cette approche garantit un contrôle strict des actions et une exécution homogène des configurations.

Lancement du déploiement

- Bouton « Déployer ».
- Message de confirmation avant exécution.
- Validation puis retour après l'exécution du playbook.
-

6.4 Suivi de l'exécution et retour d'état

Description

Pendant l'exécution du playbook, l'utilisateur peut suivre en temps réel l'état du déploiement depuis le dashboard.

Les informations affichées incluent :

- Progression de l'exécution.
- Tâches en cours.
- Messages de succès ou d'échec.
- Durée totale du déploiement.

À la fin du processus, un état final est présenté :

- Succès.
- Echec partiel.
- Echec total.

Ces informations sont également journalisées pour consultation ultérieure.

- Logs en temps réel.
- Indicateur de progression.
- Statut final visible

```
PLAY [Install Apache2 on Debian] ***********************************************

TASK [Gathering Facts] *********************************************************
ok: [target]

TASK [Update APT] **************************************************************
changed: [target]

TASK [Install Apache2] *********************************************************
changed: [target]

TASK [Ensure Apache2 is running] ***********************************************
ok: [target]

PLAY RECAP *********************************************************************
target                     : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 33 Retour du playbook exécuté avec succès

6.5 Cas d'erreur et messages associés

Description

En cas d'erreur lors du déploiement, le dashboard affiche un message clair et explicite afin d'informer l'utilisateur.

Types d'erreurs possibles :

- Machine virtuelle injoignable.
- Prérequis non respectés.
- Erreur de configuration.
- Echec d'une tâche Ansible.

Les messages sont conçus pour être compréhensibles sans expertise avancée, tout en permettant à un administrateur d'identifier rapidement la cause.

Exemple de message

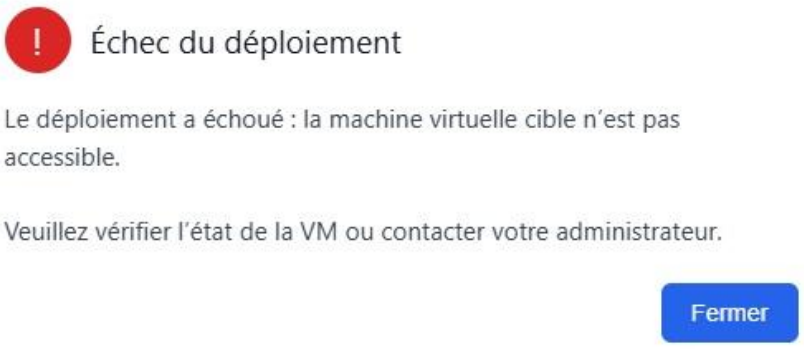


Figure 34 Echec de déploiement solutions

Chapitre 7

Automatisation et orchestration

7.1 Rôle de Terraform dans le dashboard

Terraform est utilisé comme moteur d'infrastructure as code (IaC) au sein du dashboard.

Depuis l'interface, Terraform permet :

- La création automatique de machines virtuelles sur le cluster Proxmox.
- L'application de paramètres standardisés (CPU, mémoire, stockage, réseau).
- La traçabilité des déploiements grâce aux fichiers d'état

Le dashboard agit comme une couche de pilotage simplifiée : l'utilisateur sélectionne ses paramètres, puis Terraform exécute le provisioning de manière contrôlée, sans intervention directe sur l'hyperviseur.

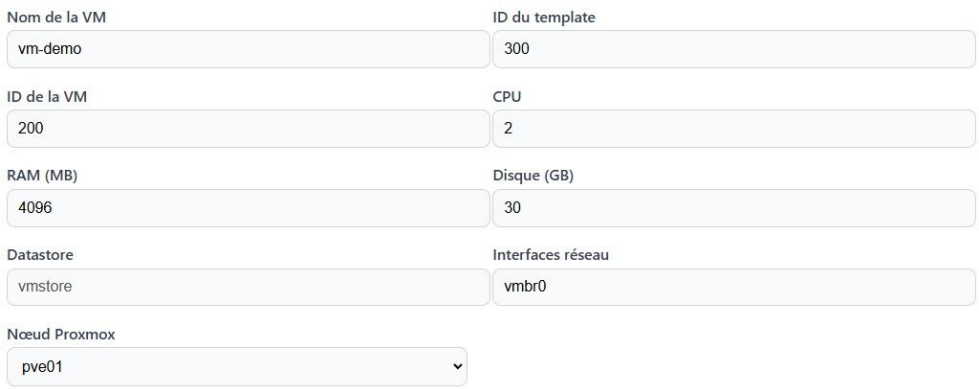


Figure 35 Dashboard - Paramètres de déploiement VMs

7.2 Rôle d'Ansible dans le dashboard

Ansible intervient après la création de la machine virtuelle, pour assurer la configuration et le déploiement applicatif.

Via le dashboard, Ansible permet :

- L'installation automatisée de solutions applicatives (Apache, Nginx, Docker, Kubernetes, Graylog, etc.).

- Le durcissement de la configuration système.
- L'application de rôles standards et reproductibles.

L'exécution se fait de manière idempotente, garantissant qu'une action répétée n'entraîne pas d'effet indésirable.

Figure 36 Ansible - Paramètres de déploiement solutions applicatives

7.3 Intégration Jenkins (pipelines)

Jenkins est utilisé comme orchestrateur des pipelines d'automatisation, son rôle dans le projet est de :

- Lancer les pipelines Terraform et Ansible.
- Centraliser l'exécution des tâches automatisées.
- Fournir un retour d'état détaillé (succès, échec, logs).

Le dashboard déclenche Jenkins via API, sans exposer l'outil directement à l'utilisateur final, ce qui renforce la sécurité et la simplicité d'utilisation.

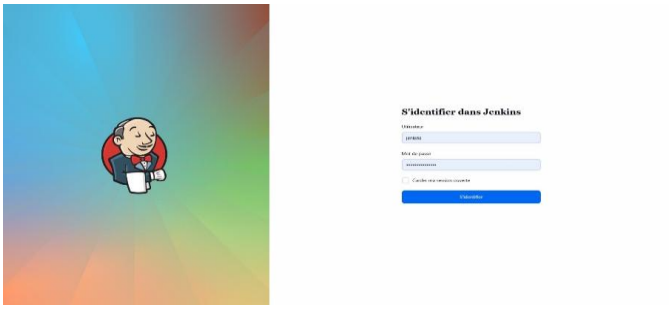


Figure 37 Jenkins - Connexion depuis le Dashboard

Figure 36 Jenkins - Agents

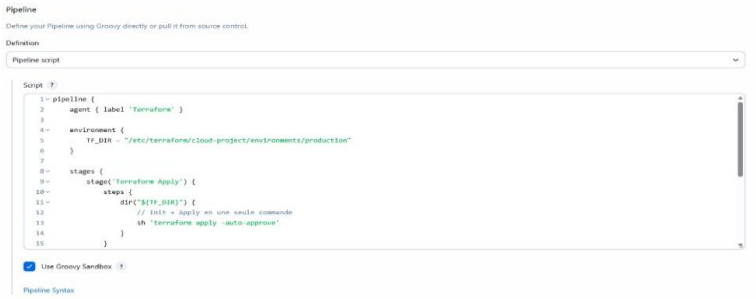


Figure 38 Jenkins - Extrait de Pipeline Terraform

7.4 Chaîne complète d'automatisation

La chaîne d'automatisation suit le processus suivant :

1. L'utilisateur déclenche une action depuis le dashboard.
2. Le dashboard valide les paramètres et les droits.
3. Jenkins lance le pipeline correspondant.
4. Terraform crée la machine virtuelle sur Proxmox.
5. Ansible configure la VM et déploie la solution applicative.
6. Les logs et statuts sont centralisés et affichés dans le dashboard.

Cette chaîne garantit :

- Une exécution cohérente et reproductible.
- Une réduction significative des erreurs humaines.

- Une vision globale du cycle de vie des ressources.

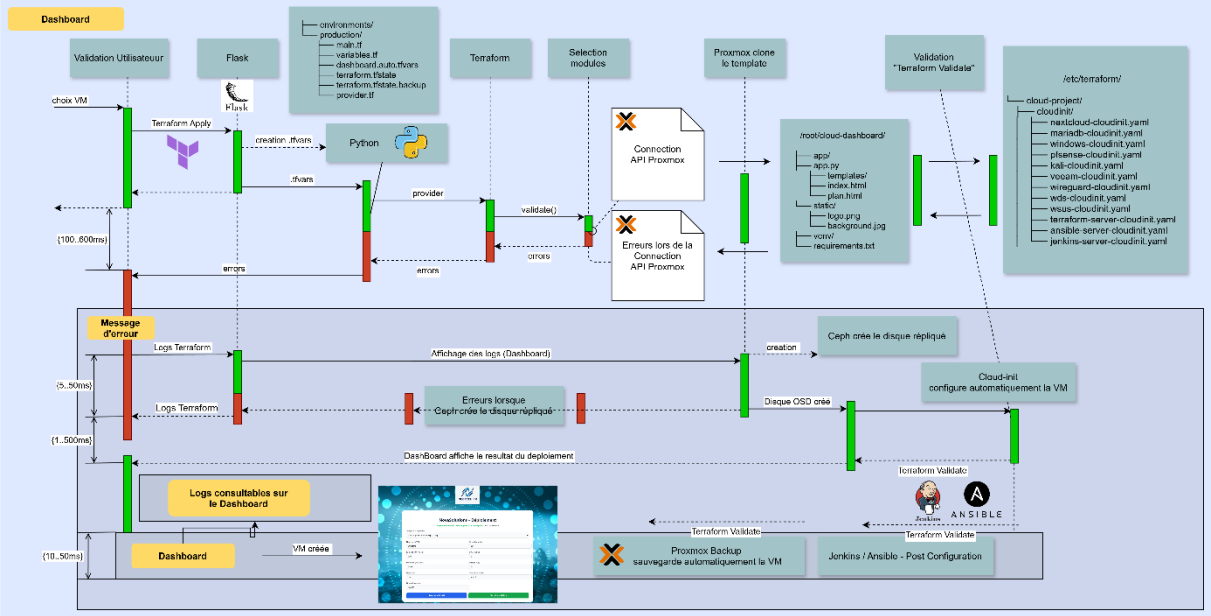


Figure 39 Schéma de chaîne d'automatisation

7.5 Limites et précautions d'usage

Bien que l'automatisation apporte de nombreux bénéfices, certaines précautions doivent être respectées :

- Vérifier l'état de la machine virtuelle cible avant tout déploiement.
- Ne pas lancer plusieurs actions concurrentes sur une même ressource.
- Tester les playbooks Ansible dans un environnement de validation.
- Limiter les droits utilisateurs aux actions nécessaires.
- Surveiller les logs Jenkins et Ansible en cas d'échec

Ces bonnes pratiques permettent de maintenir la stabilité, la sécurité et la fiabilité de la plateforme automatisée.

Chapitre 8

Supervision et journalisation

8.1 Consultation des statuts d'infrastructure

Le dashboard offre une vue synthétique de l'état global de l'infrastructure.
Les informations consultables incluent :

- L'état des nœuds Proxmox (actif, indisponible, en maintenance).
- Le nombre de machines virtuelles actives.
- L'utilisation des ressources (CPU, mémoire, stockage).
- La disponibilité des services critiques.

Ces indicateurs permettent à l'utilisateur d'identifier rapidement toute dégradation de service.

8.2 Accès aux journaux centralisés (Graylog)

Les journaux systèmes et applicatifs sont centralisés via Graylog, garantissant une vision unifiée des événements.

Graylog permet :

- La collecte des logs des nœuds Proxmox.
- La remontée des logs des machines virtuelles.
- L'analyse des événements de sécurité et d'exploitation.

L'accès aux journaux se fait depuis une interface dédiée, avec des mécanismes de filtrage et de recherche.

Figure 40 Fenêtre de connexion à Graylog

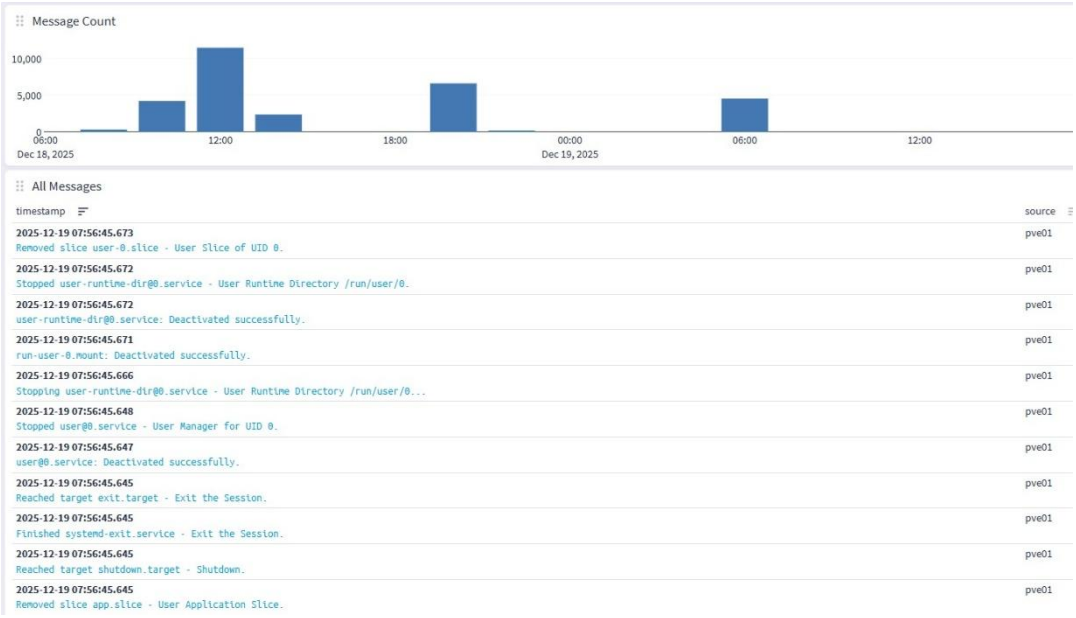


Figure 41 Graylog - Réception des logs pve001

8.3 Lecture et interprétation des logs

Les logs sont organisés de manière à faciliter leur interprétation.

Chaque entrée de log contient généralement :

- La date et l'heure de l'événement
- La source (nœud, VM, service)
- Le niveau de gravité (information, avertissement, erreur)
- Le message associé

Une lecture régulière des journaux permet d'anticiper certains incidents et de comprendre le comportement du système.

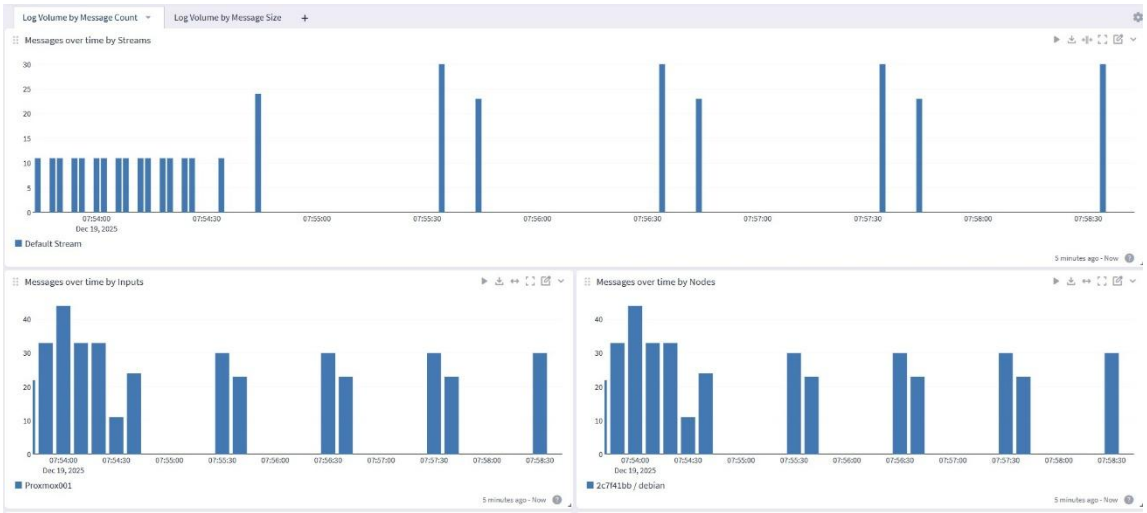


Figure 42 Graylog - Filtrage des Logs

8.4 Détection des erreurs courantes

Grâce à la centralisation des journaux, les erreurs les plus fréquentes peuvent être rapidement identifiées, notamment :

- Machines virtuelles inaccessibles.
- Échecs de déploiement automatisé.
- Problèmes réseau ou de connectivité.
- Erreurs d'authentification ou de permissions

Le dashboard peut afficher des messages explicites afin d'informer l'utilisateur et de limiter les erreurs de manipulation.

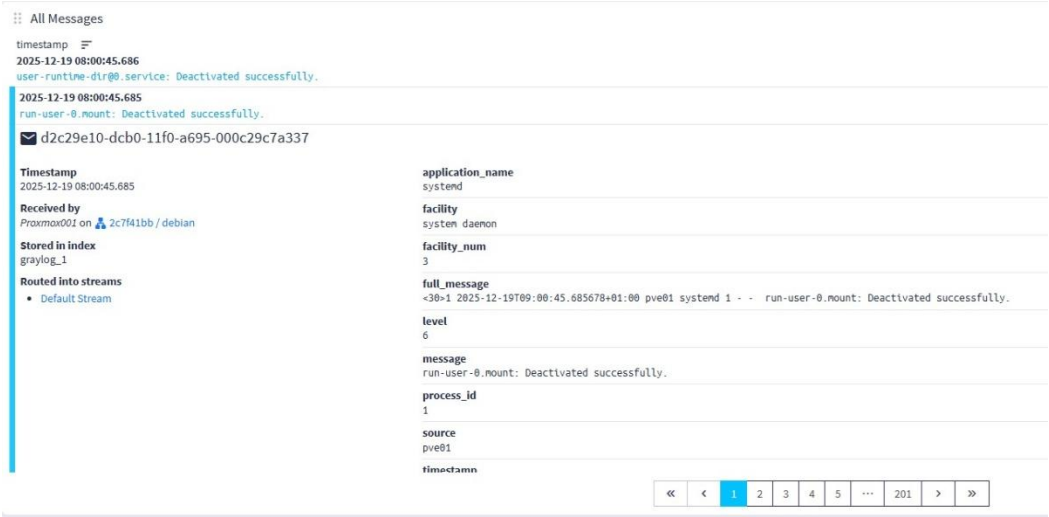


Figure 43 Graylog - Messages explicites d'authentification

8.5 Bonnes pratiques de supervision

Afin de garantir une supervision efficace et proactive, les bonnes pratiques suivantes sont recommandées :

- Consulter régulièrement les indicateurs du dashboard.
- Mettre en place des filtres et recherches enregistrées dans Graylog.
- Surveiller les erreurs répétitives ou inhabituelles.
- Conserver un historique des événements critiques.
- Réagir rapidement aux alertes pour éviter les incidents majeurs.

Une supervision rigoureuse contribue directement à la stabilité, la sécurité et la disponibilité du cloud privé.

Chapitre 9

Sécurité et bonnes pratiques

9.1 Principes de sécurité du dashboard

Le dashboard a été conçu selon une approche Security by Design, intégrant les principes suivants :

- Accès sécurisé par authentification forte.
- Limitation des privilèges utilisateurs.
- Centralisation des actions sensibles.
- Traçabilité des opérations réalisées.

Toutes les actions critiques (déploiement, suppression, exécution de scripts) sont encadrées et contrôlées afin d'éviter les erreurs ou les abus.

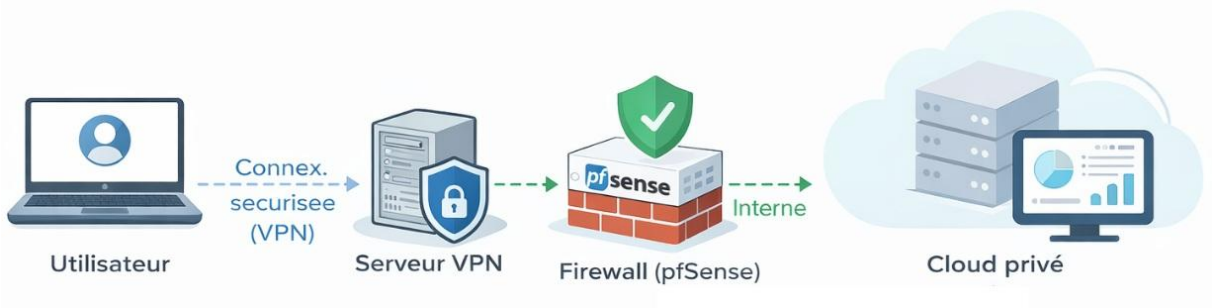
9.2 Accès réseau et restrictions

L'accès au dashboard est strictement limité et contrôlé.

Les mesures mises en place incluent :

- Accès exclusivement en HTTPS.
- Accès autorisé uniquement depuis le réseau interne ou via un VPN sécurisé.
- Filtrage réseau assuré par le pare-feu PfSense.
- Aucune exposition directe des interfaces d'administration sensibles sur Internet

Cette approche réduit considérablement les risques d'intrusion et de compromission.



9.3 Gestion des sessions utilisateurs

La gestion des sessions vise à prévenir les accès non autorisés et les détournements de session.

Les mécanismes implémentés comprennent :

- Sessions utilisateur temporisées.
- Déconnexion automatique après inactivité.

- Invalidation des sessions après déconnexion.

- Protection contre les tentatives de connexion répétées.

Ces mesures garantissent que seules les sessions actives et légitimes peuvent interagir avec le dashboard.

9.4 Bonnes pratiques de sécurité utilisateur

Afin de maintenir un niveau de sécurité élevé, les utilisateurs doivent respecter les bonnes pratiques suivantes :

- Utiliser des mots de passe robustes et uniques.
- Ne jamais partager leurs identifiants.
- Activer et conserver le second facteur d'authentification (2FA).
- Vérifier les messages de confirmation avant toute action critique.
- Signaler immédiatement toute anomalie ou comportement suspect.

Le respect de ces règles contribue à la sécurité globale de l'infrastructure.

9.5 Recommandations ANSSI appliquées

Le projet s'appuie sur plusieurs recommandations de l'ANSSI en matière de sécurisation des infrastructures virtualisées, notamment :

- Utilisation d'un hyperviseur de type 1.
- Segmentation stricte des réseaux.
- Accès administrateurs protégés.
- Journalisation et supervision centralisées.
- Limitation des services exposés.
- Privilégier les solutions open source auditées.

Ces recommandations renforcent la résilience, la traçabilité et la maîtrise de l'infrastructure tout en s'inscrivant dans une démarche de conformité et de bonnes pratiques reconnues.

Chapitre 10

Dépannage et support

10.1 Problèmes d'accès au dashboard

En cas d'impossibilité d'accéder au dashboard, plusieurs vérifications doivent être effectuées :

Vérifier que la connexion réseau est opérationnelle

- S'assurer que la connexion VPN est bien établie.
- Contrôler l'accès en HTTPS (URL correcte, certificat valide).
- Vérifier que le navigateur utilisé est compatible et à jour.
- Confirmer que le compte utilisateur dispose des droits nécessaires

Si l'accès reste impossible après ces vérifications, le problème peut être lié à une indisponibilité temporaire du service ou à une restriction réseau.

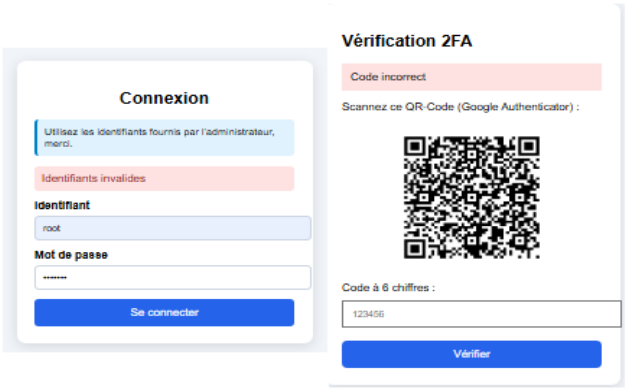


Figure 44 Dashboard - Authentification erronée

10.2 Erreurs courantes lors des déploiements

Lors du déploiement automatisé de machines virtuelles ou de solutions applicatives, certaines erreurs peuvent survenir, notamment :

- Machine virtuelle cible inaccessible ou arrêtée.
- Adresse IP incorrecte ou non joignable.
- Problème d'authentification SSH pour Ansible.
- Échec d'exécution d'un playbook ou d'un module Terraform.
- Timeout réseau ou indisponibilité temporaire des services.

Dans ces cas, le dashboard affiche un message explicite afin d'informer l'utilisateur de la nature du problème.

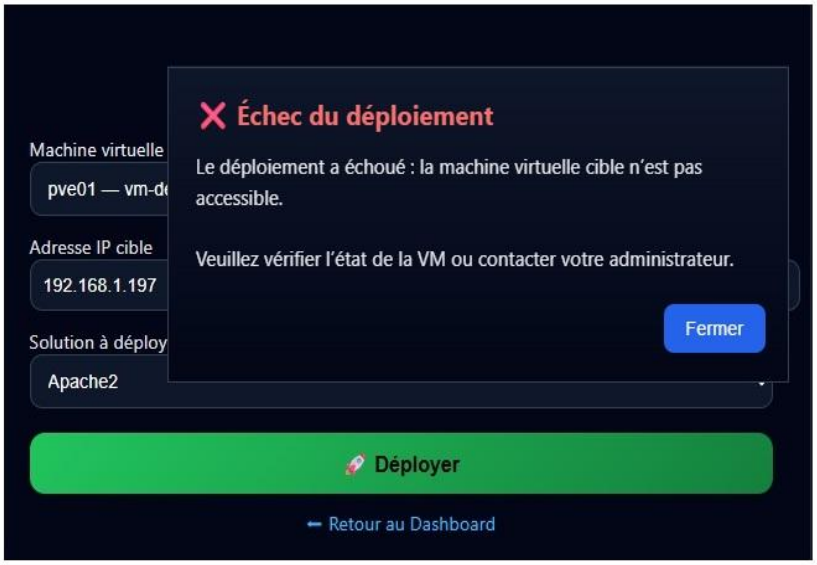


Figure 45 Dashboard - Echec du déploiement

10.3 Vérifications préalables

Avant de relancer une action ou de contacter l'administrateur, il est recommandé de vérifier :

- L'état de la machine virtuelle (démarrée, accessible).
- La cohérence des paramètres saisis (IP, solution sélectionnée).
- La disponibilité des ressources (CPU, RAM, stockage).
- L'absence d'erreurs bloquantes dans les journaux visibles.
- La stabilité de la connexion VPN et réseau

Ces vérifications permettent souvent de corriger rapidement une erreur de configuration ou d'utilisation.

10.4 Procédure de remontée d'incident

Si le problème persiste après les vérifications, l'utilisateur doit suivre la procédure de remontée d'incident :

1. Noter précisément l'action effectuée et le message d'erreur affiché
2. Identifier la machine virtuelle ou le service concerné
3. Consulter, si possible, les journaux disponibles dans le dashboard ou Graylog
4. Contacter l'administrateur ou le support technique en fournissant ces informations

Cette démarche permet une prise en charge plus rapide et un diagnostic efficace de l'incident.

Chapitre 11

Annexe

11.1 Glossaire

Cloud privé
Infrastructure cloud dédiée à une seule organisation, offrant un contrôle total sur les ressources, la sécurité et les données.

Cluster
Ensemble de serveurs interconnectés fonctionnant comme une seule entité afin d'assurer la haute disponibilité et la répartition des charges.

Proxmox VE
Plateforme open source de virtualisation basée sur KVM et LXC, utilisée pour héberger et administrer des machines virtuelles.

Machine virtuelle (VM)
Environnement informatique isolé exécuté sur un hyperviseur, disposant de ressources dédiées (CPU, RAM, stockage).

Haute disponibilité (HA)
Mécanisme permettant d'assurer la continuité de service en cas de panne d'un composant de l'infrastructure.

Ceph
Solution de stockage distribué assurant la réplication des données, la tolérance aux pannes et le stockage partagé.

RBD (RADOS Block Device)
Type de stockage bloc fourni par Ceph, utilisé pour héberger les disques des machines virtuelles.

OSD (Object Storage Daemon)
Composant Ceph responsable du stockage physique des données et de leur réplication.

Infrastructure as Code (IaC)
Approche consistant à décrire et gérer l'infrastructure à l'aide de fichiers de configuration versionnés.

Terraform
Outil d'Infrastructure as Code permettant de créer et gérer automatiquement les ressources d'infrastructure.

Ansible
Outil d'automatisation utilisé pour configurer les systèmes et déployer des applications après la création des VM.

Dashboard
Interface web centralisée permettant de piloter, superviser et administrer l'infrastructure cloud.

Supervision
Ensemble des mécanismes permettant de surveiller l'état, les performances et la disponibilité de l'infrastructure.

Journalisation (logs)
Collecte et centralisation des événements systèmes et applicatifs à des fins de suivi et de diagnostic.

Graylog
Plateforme de centralisation et d'analyse des journaux systèmes et applicatifs.

VPN (Virtual Private Network)
Tunnel sécurisé permettant un accès distant chiffré à l'infrastructure interne.

Authentification forte (2FA)
Mécanisme de sécurité combinant un mot de passe et un second facteur d'authentification.

11.2 Architecture fonctionnelle simplifiée

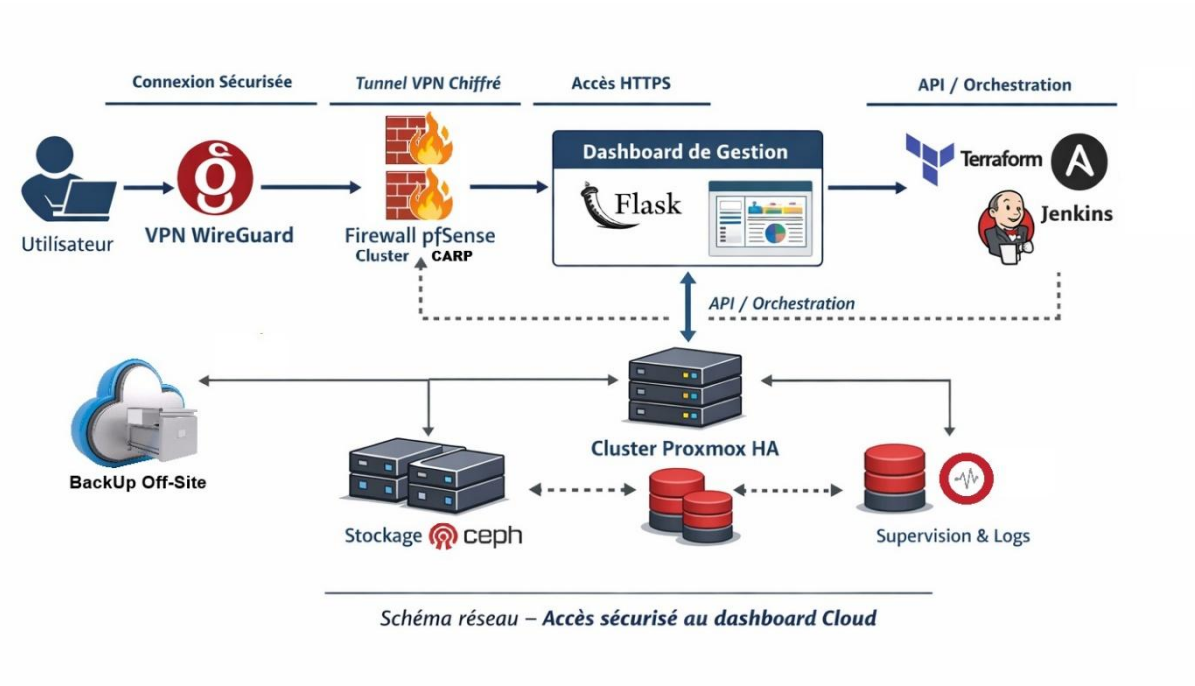


Figure 46 Schéma réseau

Légende du schéma – Accès sécurisé au dashboard

Ce schéma illustre les modes d'accès sécurisés au dashboard de gestion du cloud privé.
En situation de télétravail, l'administrateur se connecte à l'infrastructure via un tunnel VPN Wireguard, garantissant la confidentialité et l'intégrité des échanges avant de traverser le firewall PfSense.

Depuis les locaux de NovaTechSolutions, l'accès au dashboard s'effectue directement sur le réseau interne sécurisé, sans recours au VPN.

Le dashboard Flask constitue le point central d'orchestration de la plateforme. Il communique avec les outils d'automatisation (Terraform, Ansible, Jenkins) ainsi qu'avec le cluster Proxmox haute disponibilité.

L'ensemble de l'infrastructure s'appuie sur une supervision et une journalisation centralisées, assurant la traçabilité, la détection des incidents et le maintien en conditions opérationnelles.

Chapitre 12

Webographie

Documentation de virtualisation et infrastructure

Proxmox Server Solutions GmbH.
Proxmox Virtual Environment – Documentation officielle.
https://pve.proxmox.com/wiki/Documentation

Proxmox Server Solutions GmbH.
Proxmox Backup Server – Documentation officielle.
https://pbs.proxmox.com/wiki/Main_Page

Proxmox Server Solutions GmbH.
Proxmox Datacenter Manager – Documentation officielle.
https://www.proxmox.com/en/proxmox-datacenter-manager

Automatisation & DevOps (Infrastructure as Code)

HashiCorp.
Terraform Documentation.
https://developer.hashicorp.com/terraform/docs

HashiCorp.
Terraform Provider Proxmox (bpg/proxmox).
https://registry.terraform.io/providers/bpg/proxmox

Red Hat.
Ansible Documentation.
https://docs.ansible.com/

Jenkins Project.
Jenkins User Documentation.
https://www.jenkins.io/doc/

Sécurité, bonnes pratiques et recommandations

ANSSI – Agence Nationale de la Sécurité des Systèmes d'Information.
Recommandations de sécurité relatives à la virtualisation.
https://www.ssi.gouv.fr

ANSSI.
Guide d'hygiène informatique – Bonnes pratiques de sécurité.
https://www.ssi.gouv.fr/guide/guide-dhygiene-informatique/

WireGuard Project.
WireGuard – Documentation officielle.
https://www.wireguard.com/

pfSense Documentation.
pfSense Firewall Documentation.
https://docs.netgate.com/pfsense/en/latest/

Supervision, journalisation et observabilité

Graylog Inc.
Graylog Documentation.
https://go2docs.graylog.org/

Prometheus Authors.
Prometheus Documentation.
https://prometheus.io/docs/

Grafana Labs.
Grafana Documentation.
https://grafana.com/docs/

Réseau, standards et protocols

IETF | RFCs : https://www.ietf.org/process/rfcs/

IETF – Internet Engineering Task Force.
RFC 1918 – Address Allocation for Private Internets.

IETF.
RFC 4632 – Classless Inter-Domain Routing (CIDR).

IETF.
RFC 8446 – The Transport Layer Security (TLS) Protocol Version 1.3.

Développement web & sécurité applicative

Pallets Projects.
Flask Documentation.
https://flask.palletsprojects.com/

OWASP Foundation.
OWASP Top 10 – Web Application Security Risks.
https://owasp.org/www-project-top-ten/

Chapitre 13

Index

2

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

V

W

NOTES UTILISATEUR

NOTES UTILISATEUR

NOTES UTILISATEUR

NOTES UTILISATEUR