

Présentation

En bref

Niveau : présentation, sans manipulation

Durée : 10 minutes de lecture

Prérequis : aucun

Niveau de test : reconstitué

Nébuleuse, c'est quoi ?

Nébuleuse est l'offre d'hébergement d'Erminig, une ESN bretonne fictive. Elle propose des applications auto-hébergées commandées à la carte, sur le modèle des hébergeurs en ligne. Elle repose sur des technologies libres : peu coûteuses, adaptées à un démarrage, et en phase avec les organisations qui veulent des outils libres ou une souveraineté nationale et européenne. Le client choisit ses applications sur un formulaire web, indique son nom de domaine, et reçoit une machine virtuelle dédiée, configurée et maintenue par Ansible depuis un serveur central.

Dans le scénario, cette documentation est celle de l'équipe d'exploitation de Nébuleuse, dont je suis le rédacteur. Les clients d'exemple sont l'Atelier Penhoat (`penhoat.example`) et le Cabinet Kerjean (`kerjean.example`).

Composant	Rôle
Formulaire de commande	Recueille le client, son domaine et la liste des applications choisies
Serveur Ansible	Nœud de contrôle : héberge le dépôt, le Vault et lance les playbooks
Bastion Teleport	Point d'entrée unique des administrateurs vers les VM, avec authentification forte et journal des sessions
Réseau d'administration	Réseau privé <code>10.99.0.0/24</code> , réservé à l'exploitation
VM client	Nœud géré : Debian 12 livrée avec une clé SSH, reçoit le socle puis les applications commandées

Commande web → Fichier de variables du client → Serveur Ansible → VM dédiée → Docker Compose → Traefik → Application

Concrètement, le formulaire de commande et la création de la VM (provisioning) sont hors périmètre de cette documentation. Le formulaire produit un fichier de variables, et Ansible reçoit

une machine Debian 12 vierge, accessible en SSH, dont il s'occupe de tout le reste.

Le catalogue

Le catalogue de Nébuleuse regroupe l'ensemble des applications que l'hébergeur sait déployer. Le client en choisit autant qu'il le souhaite : chaque application est un rôle Ansible indépendant, activé par une liste dans les variables du client.

Application	Usage	Documentée dans ce TP
Nextcloud	Fichiers, agenda, contacts	Oui
Immich	Photos et vidéos	Oui
Vaultwarden	Gestionnaire de mots de passe	Oui
Jellyfin	Médiathèque	Non
PaperMerge	Gestion électronique de documents	Non
Home Assistant	Domotique	Non
Matrix (Element)	Messagerie	Non

Pour garder une documentation lisible, seules les applications de base sont détaillées ici. Les autres suivent exactement le même modèle de rôle : un dossier de templates, un enregistrement dans le SSO, une règle Traefik.

Le socle commun

Quelle que soit la commande, chaque VM reçoit le même socle. Il n'est pas facturé comme une application : il fait partie du service.

Brique	Rôle
Durcissement et Docker	Système à jour, pare-feu, moteur de conteneurs
Traefik	Publie les applications en HTTPS
Authentik	Authentification unique (SSO) des applications, sauf Vaultwarden
Fail2ban	Bloque les adresses qui multiplient les échecs de connexion
Agent Teleport	Rattache la VM au bastion pour l'administration

Une VM par client : pourquoi ?

Deux architectures sont possibles pour héberger plusieurs clients. En multi-locataire, une seule installation sert tous les clients, ce qui est économique mais mélange les données et les pannes. En mono-locataire, chaque client a sa propre machine, ce qui coûte plus cher mais isole les données, les secrets et les incidents. Nébuleuse retient la seconde : une compromission ou une erreur de configuration chez un client ne peut pas toucher les autres, et la restauration d'un client est indépendante.

Le bastion

Les humains et l'automatisation n'utilisent pas le même chemin. L'automatisation (Ansible) se connecte par SSH depuis le serveur Ansible, dans le réseau d'administration. Les administrateurs, eux, passent par un bastion Teleport : une porte unique, protégée par une authentification forte, qui enregistre chaque session. Aucune VM n'accepte donc de connexion SSH depuis Internet, et toute intervention manuelle laisse une trace.

L'installation de Teleport et l'enrôlement d'une ressource sont détaillés dans le livre Teleport de cette documentation (pages « Installation » et « Enroler des ressources »). Cette documentation ne reprend que la partie propre à Nébuleuse : le déploiement de l'agent par Ansible.

Objectifs et contraintes

1. **Reproductibilité** : un nouveau client est déployé à partir d'une VM vierge avec une seule commande, hors saisie des variables.
2. **Commande à la carte** : la liste des applications est une simple variable, sans modifier le code.
3. **Isolation** : chaque client a ses secrets, son domaine et son fichier de variables chiffré.
4. **Aucun secret en clair dans le dépôt** : tout passe par Ansible Vault.
5. **Administration tracée** : SSH n'écoute que sur le réseau d'administration, et les interventions humaines passent par le bastion.
6. **Idempotence** : rejouer le déploiement sur un client à jour ne modifie rien.

Adressage et nommage

Machine	Nom	Adresse d'administration	Adresse publique
---------	-----	--------------------------	------------------

Serveur Ansible	ansible-01	10.99.0.2	—
Bastion Teleport	bastion	10.99.0.3	192.0.2.5
Premier client (Atelier Penhoat)	client-alpha	10.99.0.21	192.0.2.21
Second client (Cabinet Kerjean)	client-beta	10.99.0.22	192.0.2.22

Chaque client possède son propre domaine (`penhoat.example`, `kerjean.example`) et des sous-domaines par service : `sso.`, `cloud.`, `photos.`, `vault.`. Le domaine `nebuleuse.example` ne sert qu'au service lui-même (bastion, serveur Ansible). Les conteneurs sont nommés `<client>_<service>`, les jails Fail2ban `nebuleuse-<service>`, et les variables de rôle sont préfixées par le nom du rôle (`traefik_`, `authentik_`).

Structure du dépôt

```
nebuleuse/
├─ ansible.cfg
├─ requirements.txt
├─ requirements.yml
├─ site.yml
├─ inventory/
│   ├─ hosts.yml
│   ├─ group_vars/all/
│   └─ host_vars/client-alpha/
└─ roles/
    ├─ base-hardening/
    ├─ docker-engine/
    ├─ teleport-agent/
    ├─ traefik/
    ├─ authentik/
    ├─ fail2ban/
    ├─ nextcloud/
    ├─ immich/
    └─ vaultwarden/
```

Comment lire cette documentation

Les pages se lisent dans l'ordre pour un premier déploiement. La page **Articulation des services** donne la vue d'ensemble : elle explique pourquoi les rôles sont appliqués dans cet ordre et comment les services se référencent entre eux. Chaque page de service suit ensuite le même plan : principe, prérequis, fichiers commentés, déploiement, vérification, pièges.

Les fichiers sont inspirés d'un déploiement réel sur Debian 12. Ils seront rejoués de bout en bout sur deux VM avant d'être annoncés comme testés.

Page suivante : **Mise en place du serveur Ansible.**

Revision #1

Created 2026-10-17 13:54:43 UTC by clement-derouet

Updated 2026-10-17 13:54:43 UTC by clement-derouet