

Bastion Teleport : agent et enrôlement

Le bastion, c'est quoi ?

Un bastion est une porte d'entrée unique pour l'administration des machines. Au lieu d'ouvrir SSH sur chaque VM, les administrateurs se connectent au bastion, qui les authentifie, vérifie leurs droits, puis les met en relation avec la machine demandée en enregistrant la session. Nébuleuse utilise Teleport pour cela.

Composant	Rôle
Proxy Teleport (sur le bastion)	Reçoit les connexions des administrateurs et des agents, sert le portail web
Auth Teleport (sur le bastion)	Authentifie, émet les certificats, applique les droits, conserve les enregistrements
Agent Teleport (sur chaque VM)	Rattache la VM au bastion et ouvre les sessions demandées

Administrateur → Bastion (authentification forte) → Tunnel déjà ouvert par l'agent → VM client

Concrètement, c'est l'agent qui établit la connexion vers le bastion, pas l'inverse. La VM n'a donc aucun port d'administration à exposer : le tunnel est sortant, sur le port 443.

Note — L'installation du bastion lui-même (serveur Teleport, nom de domaine, création d'utilisateurs) est détaillée dans le livre Teleport de cette documentation, pages « Installation » et « Enroler des ressources ». Cette page ne traite que ce qui est propre à Nébuleuse : déployer l'agent par Ansible, sans passer par l'interface web.

Pourquoi un déploiement par Ansible ?

L'interface web de Teleport génère une commande à copier sur la machine, avec un jeton temporaire. C'est pratique pour une machine, mais cela ne passe pas à l'échelle : chaque nouveau client demanderait une intervention manuelle. Le rôle `teleport-agent` fait la même chose automatiquement : il demande un jeton au bastion, installe l'agent, écrit sa configuration et le démarre.

Prérequis

1. Un bastion Teleport en service, joignable à `bastion.nebuleuse.example:443` depuis les VM clientes.
2. Le bastion est présent dans l'inventaire (voir ci-dessous), car Ansible doit y exécuter une commande.
3. La version de l'agent est inférieure ou égale à celle du bastion : un agent plus récent que le serveur n'est pas pris en charge.
4. Le socle système est appliqué sur la VM (page précédente).

Ajouter le bastion à l'inventaire

Le bastion n'est pas un client : il forme son propre groupe, ce qui permet à Ansible de lui déléguer la création des jetons.

```
# inventory/hosts.yml (version complétée)
all:
  children:
    bastion:                                # groupe du bastion : une seule machine
      hosts:
        bastion:
          ansible_host: 10.99.0.3
    clients:                                # groupe des VM clientes
      hosts:
        client-alpha:
          ansible_host: 10.99.0.21
        client-beta:
          ansible_host: 10.99.0.22
```

Arborescence du rôle

```
roles/teleport-agent/  
├─ defaults/main.yml  
├─ handlers/main.yml  
├─ tasks/main.yml  
└─ templates/teleport.yaml.j2
```

Variables

```
# roles/teleport-agent/defaults/main.yml  
  
# Version et édition de Teleport (édition communautaire, gratuite).  
# Doit être inférieure ou égale à la version du bastion.  
teleport_agent_version: "17.4.3"  
teleport_agent_edition: "oss"  
  
# Adresse du proxy du bastion (définie une fois pour tous dans group_vars/all).  
teleport_agent_proxy: "{{ teleport_proxy_addr }}"  
  
# Durée de validité du jeton d'enrôlement. Il ne sert qu'à la première connexion :  
# une durée courte limite l'exposition si le jeton fuit.  
teleport_agent_token_ttl: "30m"  
  
# Étiquettes (labels) attachées à la machine dans Teleport.  
# Les droits d'accès des administrateurs s'appuient dessus (voir plus bas).  
teleport_agent_labels:  
  client: "{{ client_name }}"  
  env: production
```

Configuration de l'agent

```
# roles/teleport-agent/templates/teleport.yaml.j2  
# Fichier géré par Ansible : toute modification manuelle sera écrasée.  
version: v3  
  
teleport:
```

```

# Nom affiché dans Teleport : le nom d'hôte de l'inventaire (unique par client).
nodename: {{ inventory_hostname }}
data_dir: /var/lib/teleport
# Adresse du bastion : l'agent s'y connecte en sortant.
proxy_server: {{ teleport_agent_proxy }}
{% if teleport_join_token.stdout | default('') | trim | length > 0 %}
# Présent uniquement lors du premier enrôlement. Une fois la machine enrôlée,
# elle conserve son identité dans data_dir et n'a plus besoin du jeton.
join_params:
    method: token
    token_name: {{ teleport_join_token.stdout | trim }}
{% endif %}
log:
    output: stderr
    severity: INFO

# Cette machine n'est ni un serveur d'authentification ni un proxy.
auth_service:
    enabled: "no"
proxy_service:
    enabled: "no"

# Le service SSH de Teleport : celui qui ouvre les sessions pour les administrateurs.
ssh_service:
    enabled: "yes"
    labels:
{% for nom, valeur in teleport_agent_labels.items() %}
    {{ nom }}: {{ valeur }}
{% endfor %}

```

Tâches

```

# roles/teleport-agent/tasks/main.yml

# 1. Installer Teleport seulement s'il est absent.
- name: Chercher un Teleport déjà installé
  ansible.builtin.command: which teleport
  register: teleport_agent_installed

```

```

changed_when: false
failed_when: false          # un code de retour non nul signifie simplement "absent"

- name: Installer Teleport (script officiel, version figée)
  # Même méthode que la page « Installation » du livre Teleport.
  # "set -o pipefail" : un échec de curl fait échouer toute la commande.
  ansible.builtin.shell: |
    set -o pipefail
    curl -fsSL https://cdn.teleport.dev/install-v{{ teleport_agent_version }}.sh \
      | bash -s {{ teleport_agent_version }} {{ teleport_agent_edition }}
  args:
    executable: /bin/bash
  when: teleport_agent_installed.rc != 0

# 2. Savoir si la machine est déjà enrôlée.
# Le fichier host_uuid est créé au premier démarrage réussi de l'agent.
- name: Chercher une identité Teleport existante
  ansible.builtin.stat:
    path: /var/lib/teleport/host_uuid
  register: teleport_agent_identity

# 3. Demander un jeton au bastion, seulement pour une première installation.
- name: Générer un jeton d'enrôlement sur le bastion
  ansible.builtin.command: >-
    tctl tokens add --type=node --ttl={{ teleport_agent_token_ttl }} --format=text
  delegate_to: bastion          # la commande s'exécute sur le bastion, pas sur la VM
  register: teleport_join_token
  changed_when: true           # un nouveau jeton est créé à chaque exécution
  no_log: true                 # le jeton est un secret : il ne doit pas apparaître dans les
                                journaux
  when: not teleport_agent_identity.stat.exists

# 4. Écrire la configuration (accès restreint : elle contient le jeton au premier passage).
- name: Déposer la configuration de l'agent
  ansible.builtin.template:
    src: teleport.yaml.j2
    dest: /etc/teleport.yaml
    owner: root
    group: root
    mode: "0600"

```

```
notify: Redémarrer teleport
no_log: true
```

```
# 5. Démarrer l'agent.
```

```
- name: Activer et démarrer l'agent Teleport
  ansible.builtin.service:
    name: teleport
    state: started
    enabled: true
```

```
# roles/teleport-agent/handlers/main.yml
```

```
- name: Redémarrer teleport
  ansible.builtin.service:
    name: teleport
    state: restarted
```

Les droits des administrateurs

L'agent rattache la machine, mais ne dit pas qui a le droit de s'y connecter. Cela se définit dans des rôles Teleport, qui s'appuient sur les étiquettes de la machine. Dans Nébuleuse, une personne du support ne voit que les machines des clients qu'elle suit.

```
# support-alpha.yaml : rôle Teleport (à créer une fois sur le bastion)
kind: role
version: v7
metadata:
  name: support-alpha
spec:
  allow:
    # Comptes Unix sur la VM avec lesquels la session peut s'ouvrir.
    logins: [deploy]
    # Seules les machines portant l'étiquette client=alpha sont visibles.
    node_labels:
      client: alpha
```

```
# Sur le bastion : crée ou met à jour le rôle.
sudo tctl create -f support-alpha.yaml
```

```
# Crée un utilisateur avec ce rôle (même commande que dans le livre Teleport).  
sudo tctl users add camille --roles=support-alpha
```

Note — Dans cet exemple, la session s'ouvre avec le compte `deploy`, qui a `sudo` sans mot de passe : donner ce login revient à donner les droits administrateur sur la VM. Pour un vrai support, un compte dédié avec des droits limités est préférable. Il s'ajouterait au socle système.

L'authentification forte

La page de présentation parle d'authentification forte sur le bastion. Elle se configure côté serveur Teleport, dans la section `auth_service`. Voici le principe :

```
# /etc/teleport.yaml sur le bastion (extrait)  
auth_service:  
  authentication:  
    type: local  
    # La deuxième authentification est exigée pour chaque utilisateur.  
    # "webauthn" accepte les clés de sécurité matérielles et les passkeys.  
    second_factor: webauthn  
    webauthn:  
      rp_id: bastion.nebuleuse.example
```

Note — Ce paramètre appartient à l'installation du bastion, pas à l'agent. Il est à valider sur ta version de Teleport avant de le recopier dans la page d'installation.

Déploiement

Le rôle s'insère dans `site.yml` juste après le socle :

```
# site.yml (extrait)  
roles:  
  - { role: base-hardening, tags: [base] }  
  - { role: docker-engine, tags: [docker] }  
  - { role: teleport-agent, tags: [teleport] }
```

```
# Un seul rôle, sur un seul client.  
ansible-playbook site.yml --limit client-alpha --tags teleport
```

Vérification

```
# Sur la VM : le service tourne et la connexion au bastion est établie.  
systemctl status teleport --no-pager  
journalctl -u teleport -n 20 --no-pager  
  
# Sur le bastion : la machine est enrôlée, avec ses étiquettes.  
sudo tctl nodes ls
```

Sur le portail web, la machine apparaît dans la liste des ressources sous le nom `client-alpha`, avec l'étiquette `client=alpha`. Depuis le poste d'un administrateur, la connexion se fait ensuite avec `tsh ssh deploy@client-alpha` après `tsh login`.

Pièges

1. **Jeton expiré** : si le bastion est injoignable ou si le déploiement traîne, le jeton (30 minutes) expire avant le démarrage de l'agent. Le premier démarrage échoue, et les journaux de l'agent indiquent un jeton refusé. Il suffit de relancer le rôle : le fichier d'identité n'existant pas encore, un nouveau jeton est créé.
2. **Deuxième passage qui « change » quelque chose** : au premier passage, la configuration contient le jeton. Au deuxième, la machine est enrôlée, aucun jeton n'est généré, et le fichier est réécrit sans la section `join_params`. Ce changement unique est normal ; à partir du troisième passage, plus rien ne change.
3. **Agent plus récent que le bastion** : si la version de `teleport_agent_version` dépasse celle du bastion, l'agent est refusé. Mettre à jour le bastion en premier.
4. **Machine absente du portail** : vérifier que la VM atteint `bastion.nebuleuse.example:443` en sortie (pare-feu, DNS). L'agent n'a besoin d'aucun port entrant.
5. **Jeton visible dans un journal** : les tâches qui le manipulent portent `no_log: true`. Ne pas retirer cette ligne pour déboguer, ou le faire de façon temporaire et sur une VM de test.
6. **Chemin du binaire** : le test d'installation utilise la commande `which`, ce qui évite de dépendre de l'emplacement exact du binaire selon la version du paquet.

Page suivante : **Traefik**.