

# Démarrage - Première connexion

## ? 1. Première connexion

Accède à AWX via ton service :

```
kubectl get svc -n awx
```

☐ URL = `http://<node-ip>:30080` (dans ton cas)

Login :

- user : `admin`
- password :

```
kubectl get secret -n awx awx-admin-password -o jsonpath="{.data.password}" | base64 -d
```

---

## ?? 2. Configuration minimale (à faire tout de suite)

Dans l'UI AWX :

### ? Settings ? System

- Time zone → `Europe/Paris`
  - (optionnel) définir un base URL si tu exposes proprement
-

# ? 3. Créer une Credential SSH

C'est la base de tout.

## ? Resources ? Credentials ? Add

Type : **Machine**

Remplis :

- Name : `ssh-homelab`
- Username : ton user (`root` ou autre)
- Private key : ta clé SSH

☐ Exemple :

```
cat ~/.ssh/id_rsa
```

Colle la clé privée.

---

# ?? 4. Créer un Inventory

## ? Resources ? Inventories ? Add

- Name : `homelab`

Puis :

## ? Hosts ? Add

Exemple :

```
name: node1
variables:
  ansible_host: 10.151.151.10
  ansible_user: root
```

☐ Tu peux aussi ajouter un groupe :

- k8s
  - vm
  - nas
- 

## ? 5. Créer un Project (Git)

AWX bosse avec Git, pas avec des fichiers locaux.

### ? Resources ? Projects ? Add

- Name : `homelab-infra`
- Source Control Type : Git
- URL : ton repo (GitHub / Gitea / GitLab)

☐ Si privé :

→ ajoute une credential Git (token ou clé SSH)

---

## ?? 6. Créer un Job Template

C'est là que tout s'exécute.

### ? Resources ? Templates ? Add ? Job Template

Remplis :

- Name : `Ping test`
  - Inventory : `homelab`
  - Project : `homelab-infra`
  - Playbook : `ping.yml`
  - Credentials : `ssh-homelab`
- 

## ? 7. Exemple de playbook

Dans ton repo Git :

```
# ping.yml
- hosts: all
gather_facts: false
tasks:
- name: Test ping
ping:
```

---

## ?? 8. Lancer un job

Clique sur

Si tout est bon :

- hosts OK
- SSH OK
- repo OK

→ tu vois les résultats en live

---

## ? 9. Bonus utiles (important en homelab)

### ?? Variables globales (Inventory ? Variables)

```
ansible_python_interpreter: /usr/bin/python3
```

---

### ?? Tester la connectivité

Ajoute un playbook :

```
- hosts: all
tasks:
```

- command: hostname

---

## ?? Ajouter sudo

Dans Credential :

- Privilege escalation : ✓
  - Method : sudo
- 

# ? 10. Bonnes pratiques dès le début

☐ Tu es déjà en GitOps avec ArgoCD, donc :

- ✓ versionne :
    - inventories (via SCM)
    - projets
    - playbooks
  - ☐ évite :
    - config manuelle non trackée
- 

# ? Ce que tu viens de construire

Avec ça tu as :

- AWX = orchestrateur
- Git = source de vérité
- SSH = accès machines
- Kubernetes = infra

☐ Tu peux maintenant :

- déployer des apps
  - configurer tes nodes
  - gérer ton cluster
-

Revision #1

Created 2026-04-24 14:00:01 UTC by Nico là

Updated 2026-05-05 12:01:33 UTC by Nico là