# Cài AWX-Ansible

**✅ 1. Chuẩn bị hệ thống**

**# (Tùy chọn) Chỉnh sửa quyền sudo nếu cần**

**su --**

**sudo EDITOR=nano visudo**

**exit**

**# Cài đặt các gói cơ bản cần thiết**

**sudo dnf install -y epel-release**

**sudo dnf install -y git wget curl vim nano conntrack socat**

**✅ 2. Cài đặt Docker**

**# Cài Docker**

**sudo dnf install -y docker**

**sudo systemctl enable --now docker**

**# Thêm user vào nhóm docker để chạy docker không cần sudo**

**sudo usermod -aG docker $USER**

**newgrp docker**

**✅ 3. Cài đặt Kubernetes (kubectl) và Minikube**

**# Cài đặt kubectl**

**curl -LO "https://dl.k8s.io/release/$(curl -Ls https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"**

**chmod +x kubectl**

**sudo mv kubectl /usr/local/bin/**

**kubectl version --client**

**# Cài đặt Minikube**

**curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-latest.x86\_64.rpm**

**sudo rpm -Uvh minikube-latest.x86\_64.rpm**

**# Khởi động Minikube sử dụng Docker làm driver**

**minikube start --driver=docker**

**# Kiểm tra node Kubernetes**

**kubectl get nodes**

**✅ 4. Cài đặt AWX Operator**

**# Tạo namespace riêng cho AWX**

**kubectl create namespace awx**

**# Clone mã nguồn AWX Operator từ Github và checkout phiên bản ổn định (vd: 2.9.0)**

**git clone https://github.com/ansible/awx-operator.git**

**cd awx-operator**

**git checkout tags/2.9.0**

**# Tạo file cấu hình Kustomization**

**nano kustomization.yaml**

apiVersion: kustomize.config.k8s.io/v1beta1

kind: Kustomization

resources:

- github.com/ansible/awx-operator/config/default?ref=2.9.0

namespace: awx

images:

- name: quay.io/ansible/awx-operator

newTag: 2.9.0

**✅ 5. Triển khai AWX Instance**

**# Tạo Secret chứa mật khẩu admin**

**kubectl -n awx create secret generic awx-admin-password --from-literal=password='admin123'**

**# Tạo file manifest để deploy AWX**

**nano awx-deploy.yaml**

apiVersion: awx.ansible.com/v1beta1

kind: AWX

metadata:

name: awx

namespace: awx

spec:

service\_type: NodePort

ingress\_type: none

admin\_user: admin

admin\_password\_secret: awx-admin-password

**1----------# Deploy AWX Operator**

**kubectl apply -k config/default -n awx**

**kubectl apply -k . -n awx**

**2----------# Deploy AWX instance**

**kubectl apply -f awx-deploy.yaml**

**# Kiểm tra pod AWX**

**kubectl get pods -n awx**

**# Kiểm tra pod AWX chi tiết đang chạy**

**kubectl get pods -n awx -w**

**✅ 6. Truy cập AWX**

**# Lấy URL để truy cập vào AWX (NodePort)**

**minikube service awx-service -n awx --url**

**📌 Thông tin đăng nhập:**

**Username: admin**

**Password: admin123**

**URL: *(kết quả từ lệnh trên, ví dụ:*** [***http://192.168.49.2:32560***](http://192.168.49.2:32560/)***)***

Tuyệt, bạn đã thực hiện port-forward thành công. Nhưng hiện tại chỉ đang forward trên localhost (127.0.0.1), nên chỉ truy cập được từ chính máy đó, không truy cập được từ máy khác trong mạng LAN.

🎯 Muốn truy cập từ máy khác (trên LAN):

Forward về địa chỉ IP thật của máy (192.168.0.174) thay vì chỉ 127.0.0.1:

Ví dụ:

**kubectl port-forward --address 0.0.0.0 svc/awx-service 31658:80 -n awx**

Giải thích:

--address 0.0.0.0: Cho phép lắng nghe tất cả các interface, bao gồm cả 192.168.0.174

Sau đó từ máy khác trong cùng mạng LAN, bạn truy cập:

http://192.168.0.174:31420/

✅ Tóm tắt lại các bước chuẩn chỉnh:

Chạy lệnh port-forward mở cho mạng LAN:

**kubectl port-forward --address 0.0.0.0 svc/awx-service 31658:80 -n awx**

Kiểm tra IP thật của máy (nếu chưa chắc):

ip a

→ Dùng 192.168.0.174

Trên máy khác (cùng mạng), mở trình duyệt → nhập:

http://192.168.0.174:31420/

📌 Lưu ý bảo mật:

Nếu máy ansible có tường lửa (firewalld, iptables) → mở port 31420 nếu cần:

**sudo firewall-cmd --add-port=31658/tcp --permanent**

**sudo firewall-cmd --reload**

Cần mình hướng dẫn phần tường lửa hoặc thêm gì nữa không? 👨‍💻

**nano kustomization.yaml**

apiVersion: kustomize.config.k8s.io/v1beta1

kind: Kustomization

resources:

- github.com/ansible/awx-operator/config/default?ref=2.9.0

namespace: awx

images:

- name: quay.io/ansible/awx-operator

newTag: 2.9.0

**nano awx-deploy.yaml**

apiVersion: awx.ansible.com/v1beta1

kind: AWX

metadata:

name: awx

namespace: awx

spec:

service\_type: NodePort

ingress\_type: none

admin\_user: admin

admin\_password\_secret: awx-admin-password

1️⃣ Khởi động lại Minikube

minikube start

2️⃣ Xác thực cluster đã chạy

kubectl get pods -n awx

kubectl get svc -n awx

3️⃣ Lấy URL để truy cập từ bên ngoài

minikube service awx-service -n awx –url

1. Hoặc restart Minikube với driver phù hợp:

**minikube start --driver=none --apiserver-ips=10.10.1.150**

1. Hoặc nếu đang chạy docker:

**minikube start --driver=docker --listen-address=0.0.0.0**

**🎯 Tạm workaround nhanh để test:**

1. 🌟 Trên máy 172.16.1.132:

**sudo socat TCP-LISTEN:31658,fork TCP:10.10.1.150:31658**

➡ Máy ngoài truy cập: [http://**10.10.1.150**:31658](http://10.10.1.150:31658)

# Fix cho public trong lan

1. Dùng --listen-address=0.0.0.0 + --apiserver-ips

Khi start Minikube, chỉ định IP LAN thật của máy host (ví dụ 10.10.1.150):

minikube start --driver=docker --listen-address=0.0.0.0 --apiserver-ips=10.10.1.150

Sau đó expose service:

minikube service awx-service -n awx --url

Bạn sẽ nhận được URL kiểu:

http://10.10.1.150:30306

Lúc này máy khác trong LAN sẽ truy cập được.

2. Nếu vẫn ra 192.168.49.2

Bạn có thể map port từ host LAN ra Minikube bằng socat (giống cách bạn làm lúc đầu):

sudo socat TCP-LISTEN:30306,fork TCP:192.168.49.2:30306

Rồi máy trong LAN truy cập:

http://10.10.1.150:30306

1. Kiểm tra port trên host có lắng nghe chưa

Mở 1 terminal khác, chạy:

ss -tulnp | grep 30306

Nếu không thấy LISTEN 0.0.0.0:30306 → socat chưa mở được.

2. Mở firewall port

Trên Almalinux, mặc định firewalld bật. Bạn phải mở port:

sudo firewall-cmd --zone=public --add-port=30306/tcp --permanent

sudo firewall-cmd --reload

# Thêm host vào awx

Ok, mình sẽ hướng dẫn lại **từ đầu** và nói rõ từng bước đang thực hiện **ở máy nào** để bạn không bị lẫn.  
Mình sẽ giả định bạn có **3 thành phần**:

1. **Máy chạy AWX**
   * Có giao diện web AWX
   * Ví dụ: IP 192.168.1.100
   * Đây có thể là VM hoặc container chạy AWX/Minikube
2. **Máy đích cần cấu hình** (Target Host)
   * Ví dụ: 192.168.1.10
   * User: suricata
   * AWX sẽ SSH vào đây để chạy job
3. **Máy của bạn** (Laptop/PC)
   * Bạn dùng để truy cập giao diện web AWX
   * Có thể SSH vào máy AWX nếu cần

**Bước 1 – Tạo cặp SSH Key**

**Thực hiện trên máy AWX**

Vì AWX cần key riêng để SSH, bạn tạo key ngay trên máy chạy AWX để nó dùng được luôn.

ssh-keygen -t rsa -b 4096 -f /tmp/awx-key

* Sẽ tạo 2 file:
  + /tmp/awx-key → private key (dùng cho AWX Credential)
  + /tmp/awx-key.pub → public key (dùng để đưa sang máy đích)

**Bước 2 – Thêm public key vào máy đích**

**Thực hiện từ máy AWX**

Dùng lệnh này để copy public key sang máy đích 192.168.1.10 (user suricata):

ssh-copy-id -i /tmp/awx-key.pub suricata@192.168.1.10

* Khi chạy lần đầu, sẽ hỏi password của suricata → nhập đúng để copy key
* Sau khi xong, máy AWX có thể SSH sang 192.168.1.10 mà không cần password

Kiểm tra:

ssh -i /tmp/awx-key suricata@192.168.1.10

Nếu vào được mà **không hỏi password** → OK.

**Bước 3 – Tạo Credential trên web AWX**

**Thực hiện trên trình duyệt của bạn**

1. Mở **AWX Web UI** (ví dụ: http://192.168.1.100)
2. Vào **Credentials → Add**
3. Điền:
   * **Name**: SSH to Suricata
   * **Credential Type**: Machine
   * **Username**: suricata
4. Trong **SSH Private Key** → mở file /tmp/awx-key trên máy AWX và copy nội dung, dán vào đây  
   (Có thể dùng cat /tmp/awx-key để xem rồi copy)
5. Lưu lại

**Bước 4 – Thêm host vào Inventory**

1. Vào **Inventories** → Add
2. Thêm một inventory mới, sau đó Add Host:
   * Host name: 192.168.1.10
   * Description: Suricata host

**Bước 5 – Tạo Template và test**

1. Vào **Templates → Add → Job Template**
2. Chọn:
   * Inventory: cái vừa tạo
   * Project: một project nào đó (ví dụ có playbook test ping)
   * Credential: chọn **SSH to Suricata**
3. Lưu và **Launch**
4. Nếu mọi thứ đúng → job sẽ chạy **không cần nhập password**

💡 **Điểm quan trọng**

* **Tạo key** và **copy public key** sang máy đích luôn thực hiện trên **máy AWX** (vì AWX là nơi sẽ SSH)
* **Private key** chỉ cần copy vào **AWX Credential**, không cần ở máy bạn
* **Public key** phải nằm trong ~/.ssh/authorized\_keys của user ở máy đích

# FIX ImagePullBackOff

Rồi, vậy là đã rõ nguyên nhân:

\* Bạn \*\*ping được 8.8.8.8\*\* → internet hoạt động.

\* Bạn \*\*ping quay.io bị `Temporary failure in name resolution`\*\* → DNS trong Minikube bị cấu hình sai hoặc không hoạt động.

Điều này giải thích tại sao Docker trong Minikube báo \*\*`lookup quay.io ... connection refused`\*\* → không resolve được tên miền.

---

## Cách fix DNS trong Minikube

1. \*\*Chỉnh file resolv.conf trong Minikube\*\*

Trong phiên SSH vào Minikube:

```bash

sudo sh -c 'echo "nameserver 8.8.8.8" > /etc/resolv.conf'

sudo sh -c 'echo "nameserver 1.1.1.1" >> /etc/resolv.conf'

```

2. \*\*Kiểm tra lại DNS\*\*

```bash

ping quay.io

nslookup quay.io

```

Nếu ra IP → DNS đã ok.

3. \*\*Pull lại image\*\*

```bash

docker pull quay.io/ansible/awx-ee:latest

# hoặc pull tag phù hợp, ví dụ:

docker pull quay.io/ansible/awx-ee:24.7.0

```

4. \*\*Chạy lại job trong AWX\*\*

Vào lại AWX UI và rerun job → sẽ không còn `ImagePullBackOff`.

---

💡 Lưu ý:

Nếu bạn restart Minikube, file `/etc/resolv.conf` có thể bị reset. Khi đó:

\* Bạn có thể \*\*pass DNS config khi start Minikube\*\*:

```bash

minikube start --extra-config=kubelet.resolv-conf=/run/systemd/resolve/resolv.conf

```

\* Hoặc thêm DNS vào cấu hình Docker trong Minikube.

---

Mình có thể viết cho bạn \*\*script fix DNS tự động mỗi khi Minikube start\*\* để không phải gõ tay nữa.

Bạn có muốn mình làm script đó không?

# SSH WIN

👌 chuẩn rồi bạn, nhưng có vài điểm cần “fix” lại vì Windows \*\*không hỗ trợ trực tiếp `ssh-copy-id` như Linux\*\*.

Cách làm đúng để copy \*\*public key\*\* từ AWX sang Windows Server như sau:

---

## 🔹 1. Trên máy AWX (Linux)

Giả sử bạn đã có key `/tmp/awx-key.pub`

Thử copy bằng tay (Linux có sẵn `ssh-copy-id`, nhưng Windows không hiểu lệnh này):

```bash

ssh-copy-id -i /tmp/awx-key.pub Administrator@192.168.1.20

```

👉 Lệnh trên có thể \*\*fail\*\* vì Windows không hỗ trợ trực tiếp `ssh-copy-id`. Nếu lỗi, thì dùng cách thủ công ở bước 2.

---

## 🔹 2. Thủ công: Copy public key sang Windows

1. Mở file public key trên AWX:

```bash

cat /tmp/awx-key.pub

```

Copy toàn bộ dòng bắt đầu bằng `ssh-rsa ...` hoặc `ssh-ed25519 ...`.

2. Trên Windows Server, mở PowerShell với quyền Administrator.

Tạo folder `authorized\_keys` cho user Administrator (nếu chưa có):

```powershell

New-Item -ItemType Directory -Force -Path "C:\Users\Administrator\.ssh"

```

3. Tạo file `authorized\_keys`:

```powershell

notepad C:\Users\Administrator\.ssh\authorized\_keys

```

→ Paste toàn bộ nội dung key bạn vừa copy vào.

→ Save lại.

4. Đặt quyền cho file/folder:

```powershell

icacls "C:\Users\Administrator\.ssh\authorized\_keys" /inheritance:r

icacls "C:\Users\Administrator\.ssh\authorized\_keys" /grant Administrator:F

icacls "C:\Users\Administrator\.ssh" /grant Administrator:F

```

---

## 🔹 3. Kiểm tra kết nối từ AWX

Trên AWX/Linux, chạy thử:

```bash

ssh -i /tmp/awx-key Administrator@192.168.1.20

```

👉 Nếu login không hỏi password nữa là thành công ✅

---

## 🔹 4. Gợi ý cho AWX

\* Vào \*\*Credentials → Add → Machine\*\*

\* Chọn:

\* Username: `Administrator`

\* SSH Private Key: Dán nội dung private key (tương ứng với `/tmp/awx-key`)

\* Sau đó test Job Template → AWX sẽ SSH bằng key đó.

---

👉 Mình hỏi thêm: bạn muốn mình viết luôn \*\*playbook Ansible để tự động chép key sang Windows\*\* (không cần làm tay) không? Vì AWX có module `win\_copy` và `win\_lineinfile` có thể cấu hình `authorized\_keys` tự động.  
  
  
Hiểu rồi 👍, nếu bạn đã copy key vào `authorized\_keys` nhưng khi SSH vẫn bị yêu cầu \*\*password\*\*, thì nguyên nhân thường do một trong các lỗi sau (mình chia theo kinh nghiệm khi dùng OpenSSH trên Windows Server):

---

## 🔎 Nguyên nhân phổ biến

1. \*\*Sai cặp key (public/private không khớp)\*\*

\* File `/tmp/awx-key` bạn dùng để login \*\*không trùng\*\* với public key trong `authorized\_keys`.

👉 Kiểm tra lại bằng:

```bash

ssh-keygen -y -f /tmp/awx-key

```

So sánh output với dòng bạn đã paste trong `authorized\_keys`.

---

2. \*\*Sai vị trí hoặc quyền của file `authorized\_keys` trên Windows\*\*

\* File phải nằm chính xác ở:

```

C:\Users\Administrator\.ssh\authorized\_keys

```

\* Và user `Administrator` phải có quyền \*\*Full control\*\*.

Nếu chưa, chạy:

```powershell

icacls "C:\Users\Administrator\.ssh" /grant Administrator:F

icacls "C:\Users\Administrator\.ssh\authorized\_keys" /grant Administrator:F

```

---

3. \*\*Dịch vụ SSH trên Windows chưa bật chế độ key authentication\*\*

\* Mở file cấu hình:

```

C:\ProgramData\ssh\sshd\_config

```

\* Đảm bảo có các dòng:

```

PubkeyAuthentication yes

PasswordAuthentication yes

AuthorizedKeysFile \_\_PROGRAMDATA\_\_/ssh/administrators\_authorized\_keys

```

\* Hoặc thay bằng:

```

AuthorizedKeysFile .ssh/authorized\_keys

```

👉 Sau đó restart dịch vụ:

```powershell

Restart-Service sshd

```

---

4. \*\*Windows dùng file `administrators\_authorized\_keys` thay vì `authorized\_keys`\*\*

\* Trên một số bản Windows Server, SSH \*\*không đọc\*\* `C:\Users\Administrator\.ssh\authorized\_keys`

\* Mà đọc file:

```

C:\ProgramData\ssh\administrators\_authorized\_keys

```

👉 Thử copy key vào đây:

```powershell

notepad "C:\ProgramData\ssh\administrators\_authorized\_keys"

```

Paste public key, lưu lại.

Rồi fix quyền:

```powershell

icacls "C:\ProgramData\ssh\administrators\_authorized\_keys" /inheritance:r

icacls "C:\ProgramData\ssh\administrators\_authorized\_keys" /grant "Administrators:F"

```

---

5. \*\*SELinux/Firewall (ít gặp trên Windows)\*\*

Nếu port 22 mở, thường không phải lỗi này, nhưng kiểm tra lại firewall rule:

```powershell

Get-NetFirewallRule -Name \*ssh\*

```

---

## ✅ Cách debug nhanh

Trên Linux (máy AWX), chạy:

```bash

ssh -vvv -i /tmp/awx-key Administrator@192.168.1.20

```

👉 Nó sẽ in ra chi tiết, nếu thấy dòng như:

\* `Offering public key: ...` nhưng `Server refused our key` → lỗi ở phía Windows (cấu hình / quyền file / vị trí key).

\* Không thấy dòng `Offering public key` → tức là key không trùng với private key bạn dùng.

---

👉 Mình hỏi thêm: Bạn đang dùng Windows Server bản nào (2016 / 2019 / 2022)? Vì với \*\*2019/2022\*\* thì thường chỉ cần `authorized\_keys` trong profile là chạy, còn \*\*2016 trở xuống\*\* thì hay phải dùng file `administrators\_authorized\_keys`.

Bạn có muốn mình viết hướng dẫn fix riêng cho từng phiên bản Windows không?

# Project

<https://github.com/HyHonCuTe/ansible.git>  
<https://github.com/devopshobbies/ansible-templates.git>  
<https://github.com/ansible-lockdown/Windows-2022-CIS.git>  
<https://github.com/network-automation/toolkit.git>  
<https://github.com/network-automation/product-demos.git>

<https://github.com/craig-br/instruqt-track-content/>

[https://github.com/dafmendo/meraki-deployment.git](https://github.com/dafmendo/Meraki-Deployment.git)