Version : **2024.01**

Dernière mise-à-jour : 2024/06/10 11:46

LDF906 - Red Hat® Ansible® Automation Platform

Contenu du Module

LAB#1 - Ansible® Automation Controller

1.1 - Préparation

Avant de commencer, ouvrez un compte développeur chez **Red Hat**.

Important : Notez vos coordonnées de connexion au compte Red Hat.

Démarrez votre essai de **Red Hat 9**.

Important : Vous n'avez **PAS** besoin de l'iso de Red Hat 9. Annulez donc son téléchargement après que celui-ci a démarré.

Connectez-vous ensuite à votre VM **RedHat_10.0.2.101_SSH**.

Enregistrez la VM dans **votre** compte Red Hat :

```
[trainee@redhat9 ~]$ su -
Password: fenestros

[root@redhat9 ~]# subscription-manager register
```

```
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: <login>
Password: <password>
The system has been registered with ID: a85a9fe3-334d-4b0c-9a98-f408e7d3a1e7
The registered system name is: redhat9.ittraining.loc
```

Configurez SELinux en mode permissive :

```
[root@redhat9 ~]# setenforce permissive
```

Désarchivez le fichier **ansible-automation-platform-setup-bundle-2.4-2.2-x86_64.tar.gz** et consultez le contenu du répertoire créé :

```
[root@redhat9 ~]# tar xvf ansible-automation-platform-setup-bundle-2.4-2.2-x86_64.tar.gz

[root@redhat9 ~]# cd ansible-automation-platform-setup-bundle-2.4-2.2-x86_64/

[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ls
bundle  collections  group_vars  inventory  README.md  setup.sh
```

Consultez le fichier **README.md** :

```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# cat README.md
Red Hat Ansible Automation Platform Deployment
==============================================

This collection of files provides a complete set of playbooks for deploying
Red Hat Ansible Automation Platform in your environment on hosts running
Red Hat Enterprise Linux.

For getting started with installation and setup instructions, please refer to:

- Install Guide --
https://access.redhat.com/documentation/en-us/red_hat_ansible_automation_platform/2.4/html-single/red_hat_ansible
```

```
_automation_platform_installation_guide/index
```

Important : Notez l'URL du **guide d'installation**.

Modifiez le fichier **/etc/hosts** :

```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# vi /etc/hosts
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
redhat9.ittraining.loc  10.0.2.101
autohub.ittraining.loc  10.0.2.102
```

Installez ensuite le paquet **ansible-core** :

```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# dnf install ansible-core
Updating Subscription Management repositories.
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
15 MB/s |  14 MB     00:00
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
21 MB/s |  25 MB     00:01
Last metadata expiration check: 0:00:03 ago on Fri 20 Oct 2023 12:21:53 PM CEST.
Dependencies resolved.
================================================================================================================
================================================================================
 Package                                                Architecture                        Version
Repository                                                   Size
================================================================================================================
================================================================================
Installing:
 ansible-core                                           x86_64                              2.14.2-5.el9_2
```

```
rhel-9-for-x86_64-appstream-rpms                              3.4 M
Installing dependencies:
 git-core                                                 x86_64                            2.39.3-1.el9_2
rhel-9-for-x86_64-appstream-rpms                              4.3 M
 libnsl2                                                  x86_64                            2.0.0-1.el9
rhel-9-for-x86_64-appstream-rpms                               33 k
 mpdecimal                                                x86_64                            2.5.1-3.el9
rhel-9-for-x86_64-appstream-rpms                               88 k
 python3.11                                               x86_64                            3.11.2-2.el9_2.2
rhel-9-for-x86_64-appstream-rpms                               28 k
 python3.11-cffi                                          x86_64                            1.15.1-1.el9
rhel-9-for-x86_64-appstream-rpms                              315 k
 python3.11-cryptography                                  x86_64                            37.0.2-5.el9
rhel-9-for-x86_64-appstream-rpms                              1.2 M
 python3.11-libs                                          x86_64                            3.11.2-2.el9_2.2
rhel-9-for-x86_64-appstream-rpms                              9.9 M
 python3.11-pip-wheel                                     noarch                            22.3.1-2.el9
rhel-9-for-x86_64-appstream-rpms                              1.4 M
 python3.11-ply                                           noarch                            3.11-1.el9
rhel-9-for-x86_64-appstream-rpms                              141 k
 python3.11-pycparser                                     noarch                            2.20-1.el9
rhel-9-for-x86_64-appstream-rpms                              161 k
 python3.11-pyyaml                                        x86_64                            6.0-1.el9
rhel-9-for-x86_64-appstream-rpms                              229 k
 python3.11-setuptools-wheel                              noarch                            65.5.1-2.el9
rhel-9-for-x86_64-appstream-rpms                              714 k
 python3.11-six                                           noarch                            1.16.0-1.el9
rhel-9-for-x86_64-appstream-rpms                               47 k
 sshpass                                                  x86_64                            1.09-4.el9
rhel-9-for-x86_64-appstream-rpms                               30 k

Transaction Summary
=======================================================================================================================
================================================================================================
```

```
Install  15 Packages

Total download size: 22 M
Installed size: 88 M
Is this ok [y/N]: y
Downloading Packages:
(1/15): sshpass-1.09-4.el9.x86_64.rpm
24 kB/s |  30 kB     00:01
(2/15): libnsl2-2.0.0-1.el9.x86_64.rpm
26 kB/s |  33 kB     00:01
(3/15): python3.11-six-1.16.0-1.el9.noarch.rpm
36 kB/s |  47 kB     00:01
(4/15): python3.11-pycparser-2.20-1.el9.noarch.rpm
153 kB/s | 161 kB     00:01
(5/15): python3.11-setuptools-wheel-65.5.1-2.el9.noarch.rpm
420 kB/s | 714 kB     00:01
(6/15): python3.11-cffi-1.15.1-1.el9.x86_64.rpm
179 kB/s | 315 kB     00:01
(7/15): python3.11-pyyaml-6.0-1.el9.x86_64.rpm
288 kB/s | 229 kB     00:00
(8/15): python3.11-cryptography-37.0.2-5.el9.x86_64.rpm
704 kB/s | 1.2 MB     00:01
(9/15): mpdecimal-2.5.1-3.el9.x86_64.rpm
55 kB/s |  88 kB     00:01
(10/15): git-core-2.39.3-1.el9_2.x86_64.rpm
18 MB/s | 4.3 MB     00:00
(11/15): python3.11-ply-3.11-1.el9.noarch.rpm
111 kB/s | 141 kB     00:01
(12/15): python3.11-pip-wheel-22.3.1-2.el9.noarch.rpm
850 kB/s | 1.4 MB     00:01
(13/15): ansible-core-2.14.2-5.el9_2.x86_64.rpm
2.3 MB/s | 3.4 MB     00:01
(14/15): python3.11-3.11.2-2.el9_2.2.x86_64.rpm
28 kB/s |  28 kB     00:00
```

```
(15/15): python3.11-libs-3.11.2-2.el9_2.2.x86_64.rpm
5.9 MB/s | 9.9 MB     00:01
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Total
3.1 MB/s |  22 MB     00:07
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
3.5 MB/s | 3.6 kB     00:00
Importing GPG key 0xFD431D51:
 Userid     : "Red Hat, Inc. (release key 2) <security@redhat.com>"
 Fingerprint: 567E 347A D004 4ADE 55BA 8A5F 199E 2F91 FD43 1D51
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Is this ok [y/N]: y
Key imported successfully
Importing GPG key 0x5A6340B3:
 Userid     : "Red Hat, Inc. (auxiliary key 3) <security@redhat.com>"
 Fingerprint: 7E46 2425 8C40 6535 D56D 6F13 5054 E4A4 5A63 40B3
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Is this ok [y/N]: y
```

1.2 - Installation

Éditez maintenant le fichier **inventory** :

```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# vi inventory
```

```
...
[automationcontroller]
redhat9.ittraining.loc ansible_connection=local node_type=hybrid
...
```

Important : Cette section configure une installation d'un seul noeud ayant une base de données interne.

Configurez ensuite le mot de passe **fenestros** aux endroits indiqués :

```
...
[all:vars]
admin_password='fenestros'

pg_host=''
pg_port=5432

pg_database='awx'
pg_username='awx'
pg_password='fenestros'
pg_sslmode='prefer'  # set to 'verify-full' for client-side enforced SSL
...
```

```
...
registry_url='registry.redhat.io'
registry_username='yyyyyyyyyyy'
registry_password='xxxxxxxxxxx'
...
```

Cette section configure une installation d'un seul noeud ayant une base de données interne.

A faire : **Remplacez** 'yyyyyyyyyyy' avec **votre** Red Hat login et 'xxxxxxxxxxx' avec **votre** mot de passe Red Hat.

```
...
automationhub_admin_password='fenestros'

automationhub_pg_host=''
automationhub_pg_port=5432

automationhub_pg_database='automationhub'
automationhub_pg_username='automationhub'
automationhub_pg_password='fenestros'
automationhub_pg_sslmode='prefer'
...
```

```
...
automationedacontroller_admin_password='fenestros'

automationedacontroller_pg_host=''
automationedacontroller_pg_port=5432

automationedacontroller_pg_database='automationedacontroller'
automationedacontroller_pg_username='automationedacontroller'
automationedacontroller_pg_password='fenestros'
automationedacontroller_pg_sslmode='prefer'
...
```

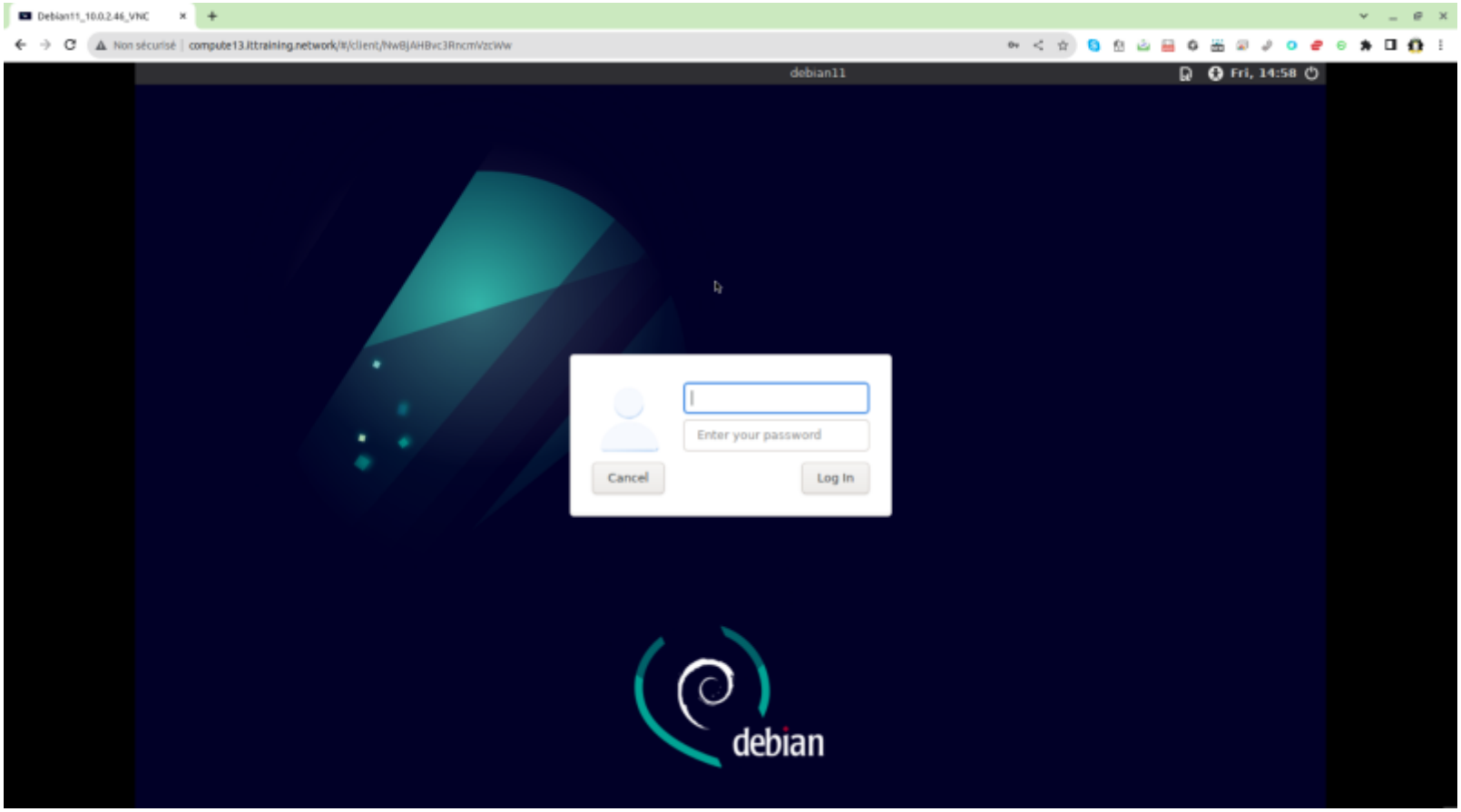
```
...
# The default install will deploy SSO with sso_use_https=True
# Keystore password is required for https enabled SSO
sso_keystore_password='fenestros'

# Single-Sign-On configuration
sso_console_admin_password='fenestros'
...
```

Exécutez le script d'installation **setup.sh** et patientez une quinzaine de minutes :

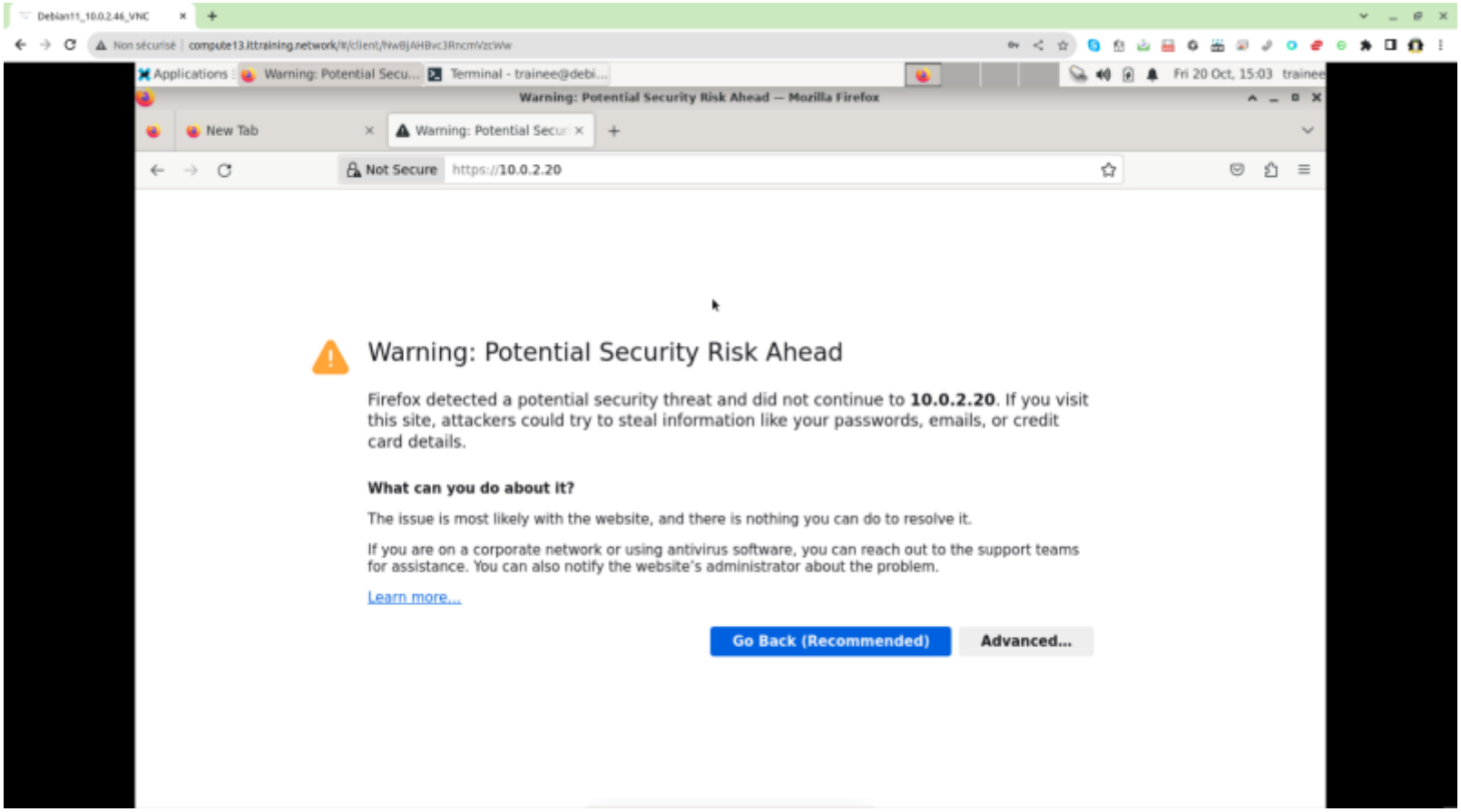
```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ./setup.sh
...
PLAY RECAP *********************************************************************
localhost                  : ok=0    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
ignored=0
redhat9.ittraining.loc     : ok=374  changed=170  unreachable=0    failed=0    skipped=246  rescued=0
ignored=6
```

1.3 - Consultation

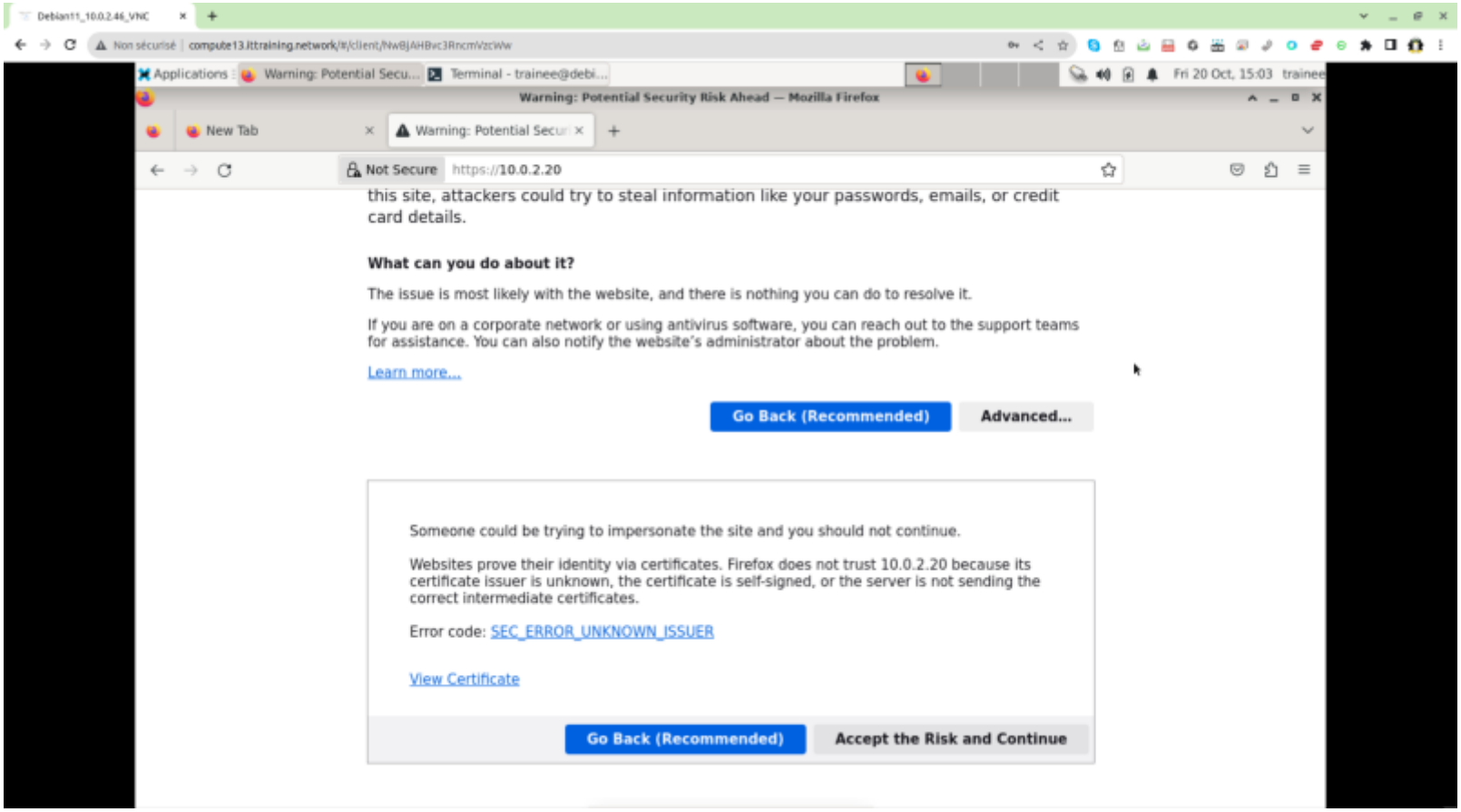
Connectez-vous à votre VM **Debian_10.0.2.46_VNC** :



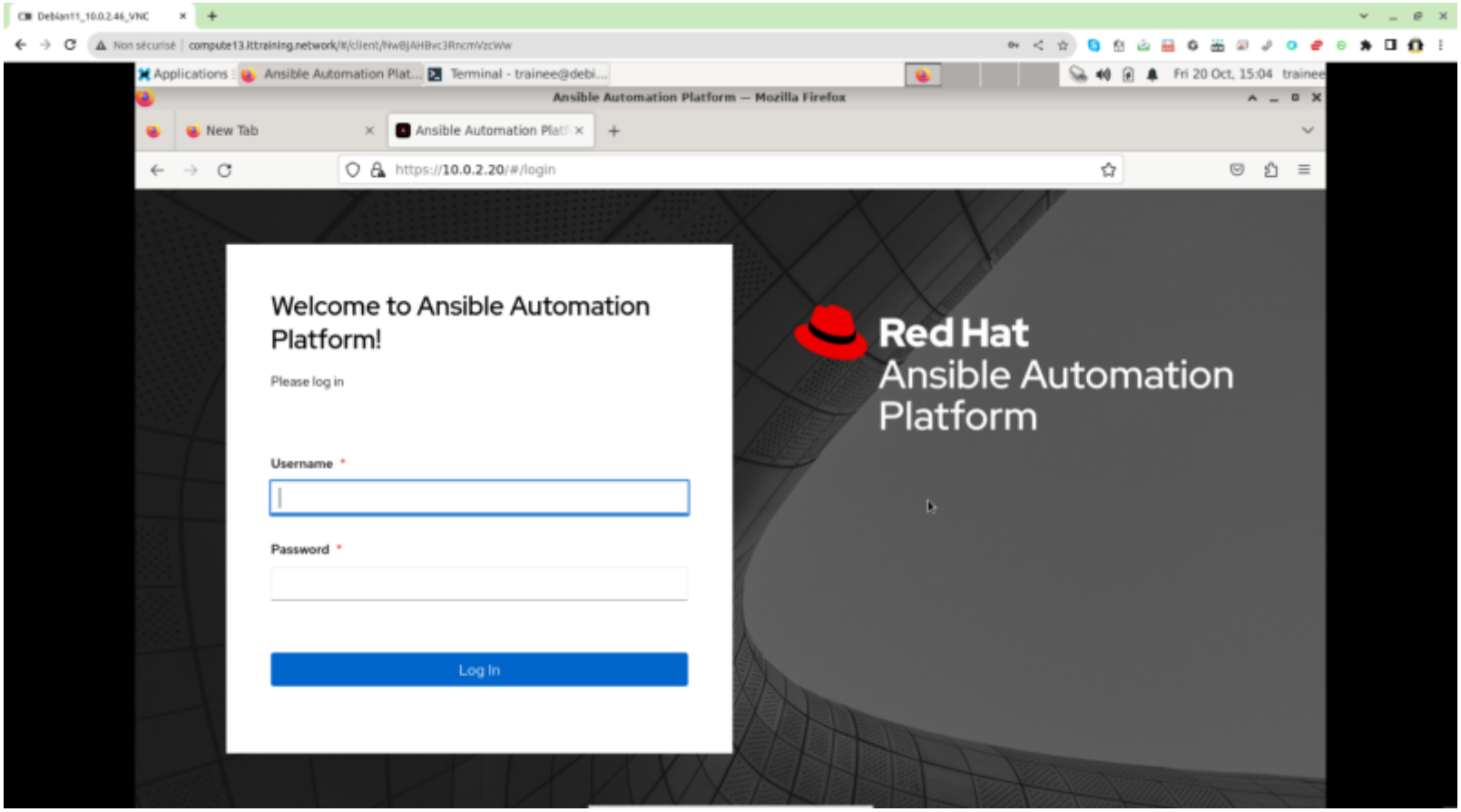
Ouvrez un navigateur Web et naviguez à **https://10.0.2.101** :



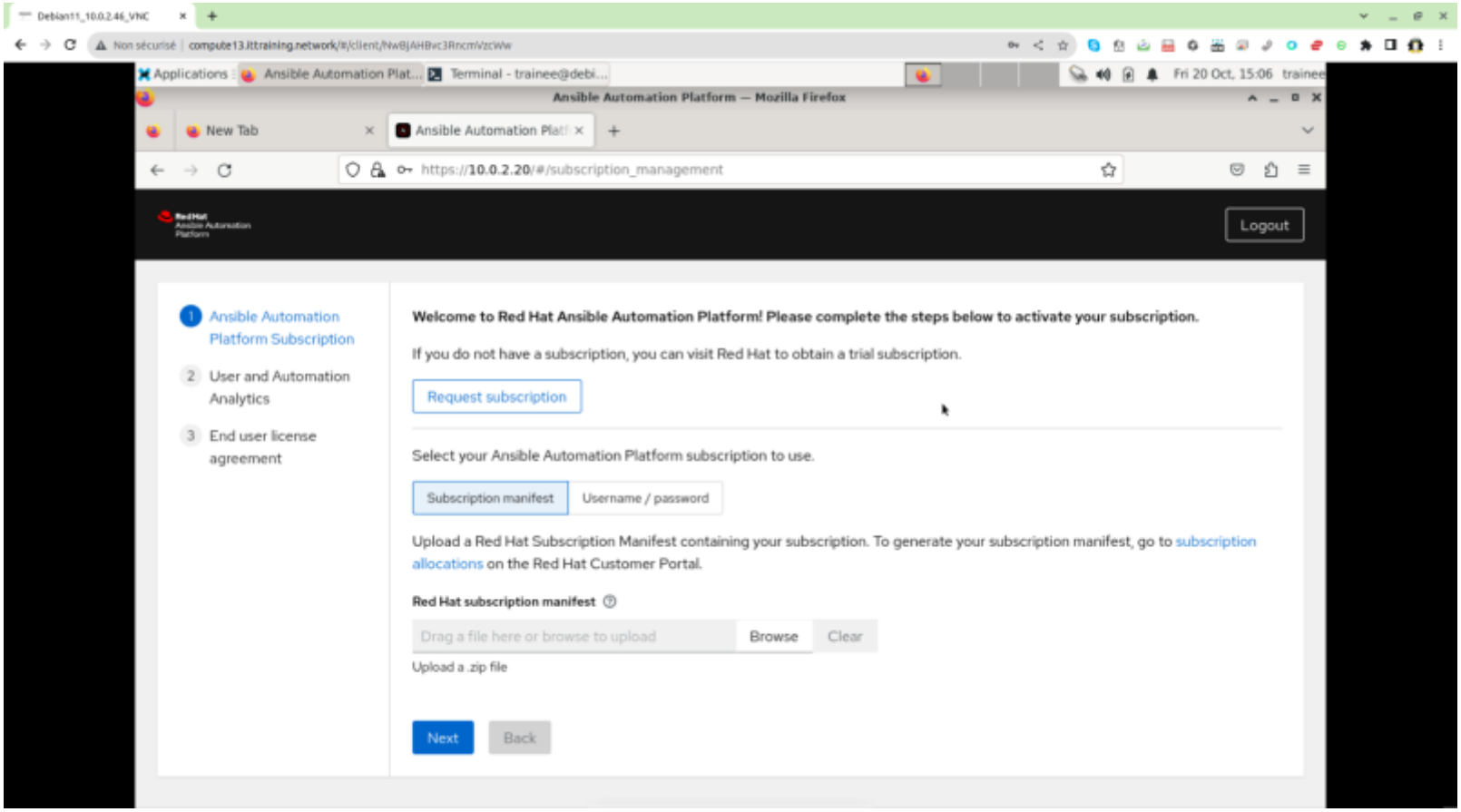
Acceptez le certificat auto-signé :



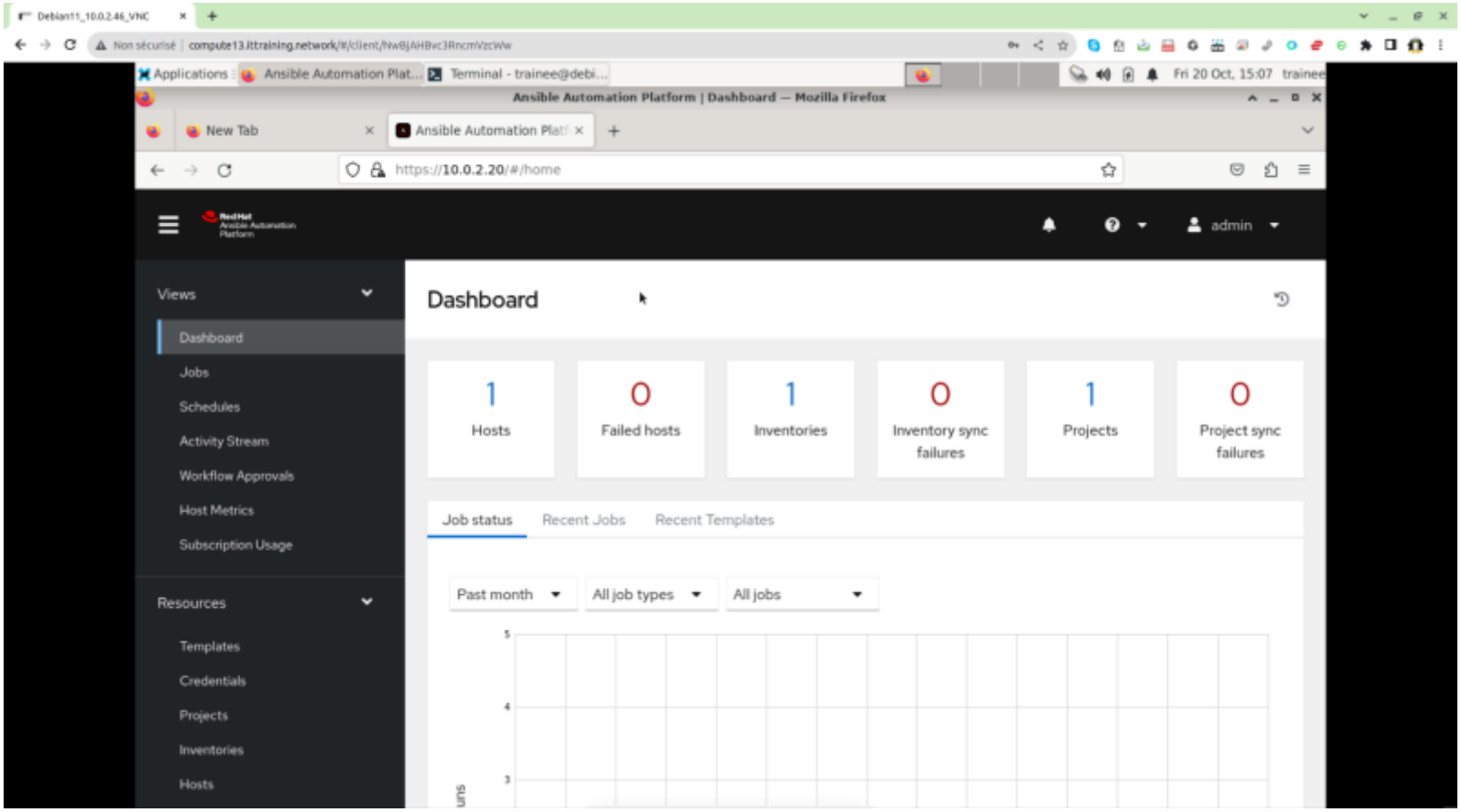
Connectez-vous à votre Ansible® Automation Controller en utilisant le compte **admin** et le mot de passe **fenestros** :



Cliquez sur le bouton **Username/password** et enregistrez votre Ansible® Automation Controller auprès de Red Hat en utilisant **votre** compte :



Vous obtiendrez :



LAB #2 - Ansible® Automation Hub

2.1 - Préparation

Connectez-vous maintenant à votre VM **autohub.ittraining.loc** :

```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ssh -l trainee 10.0.2.102
```

Enregistrez la VM dans **votre** compte Red Hat :

```
[root@autohub ~]# subscription-manager register
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: <login>
Password: <password>
The system has been registered with ID: a85a9fe3-334d-4b0c-9a98-f408e7d3a1e7
The registered system name is: autohub.ittraining.loc
```

Configurez SELinux en mode permissive :

```
[root@autohub ~]# setenforce permissive
```

Désarchivez le fichier **ansible-automation-platform-setup-bundle-2.4-2.2-x86_64.tar.gz** et consultez le contenu du répertoire créé :

```
[root@autohub ~]# tar xvf ansible-automation-platform-setup-bundle-2.4-2.2-x86_64.tar.gz

[root@autohub ~]# cd ansible-automation-platform-setup-bundle-2.4-2.2-x86_64/

[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ls
bundle  collections  group_vars  inventory  README.md  setup.sh
```

Modifiez le fichier **/etc/hosts** selon votre numéro de stagiaire :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# vi /etc/hosts
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.2.102      autohub.ittraining.loc
```

Installez ensuite le paquet **ansible-core** :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# dnf install ansible-core
```

```
Updating Subscription Management repositories.
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
25 MB/s |  14 MB     00:00
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
35 MB/s |  25 MB     00:00
Dependencies resolved.
==========================================================================================================
================================================================================================
 Package                                                 Architecture                       Version
Repository                                                   Size
==========================================================================================================
================================================================================================
Installing:
 ansible-core                                            x86_64                             2.14.2-5.el9_2
rhel-9-for-x86_64-appstream-rpms                            3.4 M
Installing dependencies:
 git-core                                                x86_64                             2.39.3-1.el9_2
rhel-9-for-x86_64-appstream-rpms                            4.3 M
 libnsl2                                                 x86_64                             2.0.0-1.el9
rhel-9-for-x86_64-appstream-rpms                             33 k
 mpdecimal                                               x86_64                             2.5.1-3.el9
rhel-9-for-x86_64-appstream-rpms                             88 k
 python3.11                                              x86_64                             3.11.2-2.el9_2.2
rhel-9-for-x86_64-appstream-rpms                             28 k
 python3.11-cffi                                         x86_64                             1.15.1-1.el9
rhel-9-for-x86_64-appstream-rpms                            315 k
 python3.11-cryptography                                 x86_64                             37.0.2-5.el9
rhel-9-for-x86_64-appstream-rpms                            1.2 M
 python3.11-libs                                         x86_64                             3.11.2-2.el9_2.2
rhel-9-for-x86_64-appstream-rpms                            9.9 M
 python3.11-pip-wheel                                    noarch                             22.3.1-2.el9
rhel-9-for-x86_64-appstream-rpms                            1.4 M
 python3.11-ply                                          noarch                             3.11-1.el9
rhel-9-for-x86_64-appstream-rpms                            141 k
```

```
 python3.11-pycparser                                    noarch                                  2.20-1.el9
rhel-9-for-x86_64-appstream-rpms                        161 k
 python3.11-pyyaml                                       x86_64                                  6.0-1.el9
rhel-9-for-x86_64-appstream-rpms                        229 k
 python3.11-setuptools-wheel                             noarch                                  65.5.1-2.el9
rhel-9-for-x86_64-appstream-rpms                        714 k
 python3.11-six                                          noarch                                  1.16.0-1.el9
rhel-9-for-x86_64-appstream-rpms                         47 k
 sshpass                                                 x86_64                                  1.09-4.el9
rhel-9-for-x86_64-appstream-rpms                         30 k

Transaction Summary
================================================================================================================
================================================================================
Install  15 Packages

Total download size: 22 M
Installed size: 88 M
Is this ok [y/N]: y
Downloading Packages:
(1/15): sshpass-1.09-4.el9.x86_64.rpm
24 kB/s |  30 kB     00:01
(2/15): libnsl2-2.0.0-1.el9.x86_64.rpm
26 kB/s |  33 kB     00:01
(3/15): python3.11-six-1.16.0-1.el9.noarch.rpm
36 kB/s |  47 kB     00:01
(4/15): python3.11-pycparser-2.20-1.el9.noarch.rpm
153 kB/s | 161 kB     00:01
(5/15): python3.11-setuptools-wheel-65.5.1-2.el9.noarch.rpm
420 kB/s | 714 kB     00:01
(6/15): python3.11-cffi-1.15.1-1.el9.x86_64.rpm
179 kB/s | 315 kB     00:01
(7/15): python3.11-pyyaml-6.0-1.el9.x86_64.rpm
288 kB/s | 229 kB     00:00
```

```
(8/15): python3.11-cryptography-37.0.2-5.el9.x86_64.rpm
704 kB/s | 1.2 MB     00:01
(9/15): mpdecimal-2.5.1-3.el9.x86_64.rpm
55 kB/s |  88 kB     00:01
(10/15): git-core-2.39.3-1.el9_2.x86_64.rpm
18 MB/s | 4.3 MB     00:00
(11/15): python3.11-ply-3.11-1.el9.noarch.rpm
111 kB/s | 141 kB     00:01
(12/15): python3.11-pip-wheel-22.3.1-2.el9.noarch.rpm
850 kB/s | 1.4 MB     00:01
(13/15): ansible-core-2.14.2-5.el9_2.x86_64.rpm
2.3 MB/s | 3.4 MB     00:01
(14/15): python3.11-3.11.2-2.el9_2.2.x86_64.rpm
28 kB/s |  28 kB     00:00
(15/15): python3.11-libs-3.11.2-2.el9_2.2.x86_64.rpm
5.9 MB/s | 9.9 MB     00:01
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Total
3.1 MB/s |  22 MB     00:07
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
3.5 MB/s | 3.6 kB     00:00
Importing GPG key 0xFD431D51:
 Userid     : "Red Hat, Inc. (release key 2) <security@redhat.com>"
 Fingerprint: 567E 347A D004 4ADE 55BA 8A5F 199E 2F91 FD43 1D51
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Is this ok [y/N]: y
Key imported successfully
Importing GPG key 0x5A6340B3:
 Userid     : "Red Hat, Inc. (auxiliary key 3) <security@redhat.com>"
 Fingerprint: 7E46 2425 8C40 6535 D56D 6F13 5054 E4A4 5A63 40B3
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Is this ok [y/N]: y
```

2.2 - Installation

Éditez maintenant le fichier **inventory** :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# vi inventory
```

```
...
[automationhub]
autohub.ittraining.loc ansible_connection=local
...
```

```
...
# Automation Hub Configuration
#

automationhub_admin_password='fenestros'

automationhub_pg_host=''
automationhub_pg_port=5432

automationhub_pg_database='automationhub'
automationhub_pg_username='automationhub'
automationhub_pg_password='fenestros'
automationhub_pg_sslmode='prefer'
...
```

Exécutez le script d'installation **setup.sh** et patientez une quinzaine de minutes :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ./setup.sh
...
PLAY RECAP *********************************************************************
autohub.ittraining.loc     : ok=165  changed=51   unreachable=0    failed=2    skipped=173  rescued=0
ignored=0
```

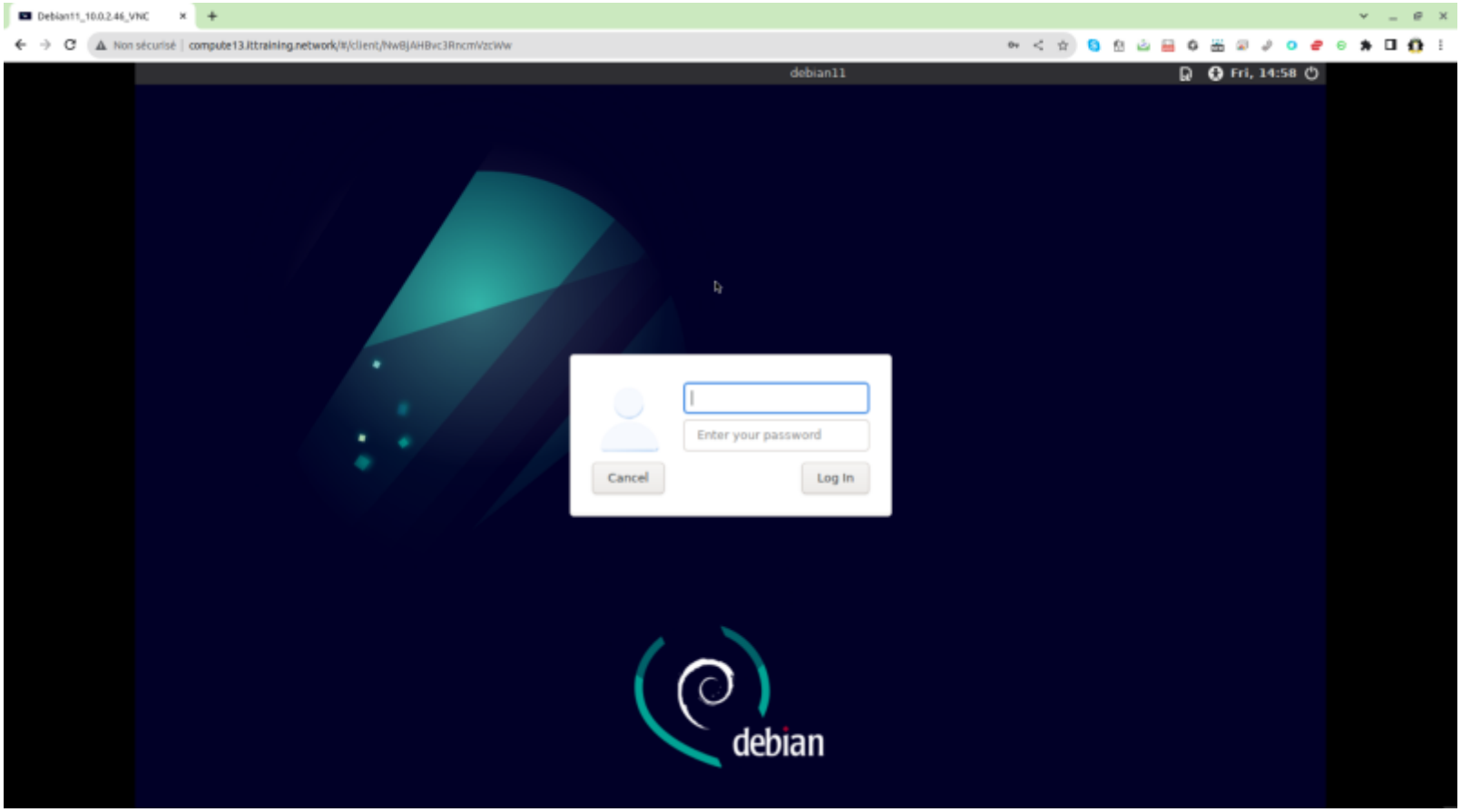
```
localhost                  : ok=0    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
ignored=0

[error] Oops!  An error occurred while running setup.
[warn] /var/log/tower does not exist. Setup log saved to setup.log.
[warn] Provided path does not exist or is not accessible. Setup log saved to ./setup.log.
```

En cas d'erreur(s), ré-exécutez le script setup.sh :

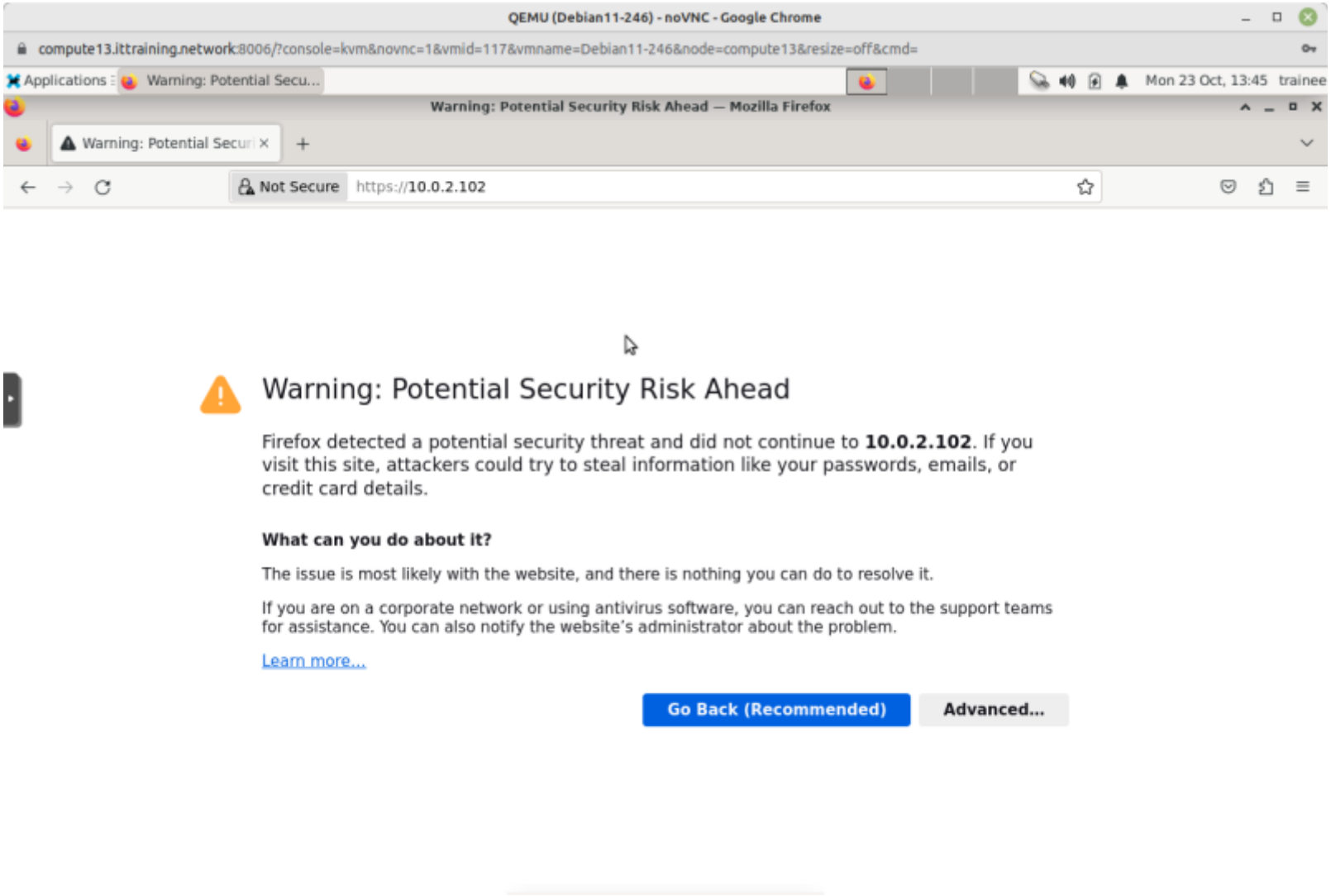
```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ./setup.sh
...
PLAY RECAP *********************************************************************
autohub.ittraining.loc     : ok=900  changed=150  unreachable=0    failed=0    skipped=379  rescued=0
ignored=0
localhost                  : ok=0    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
ignored=0

The setup process completed successfully.
[warn] /var/log/tower does not exist. Setup log saved to setup.log.
[warn] Provided path does not exist or is not accessible. Setup log saved to ./setup.log.
```

2.3 - Consultation

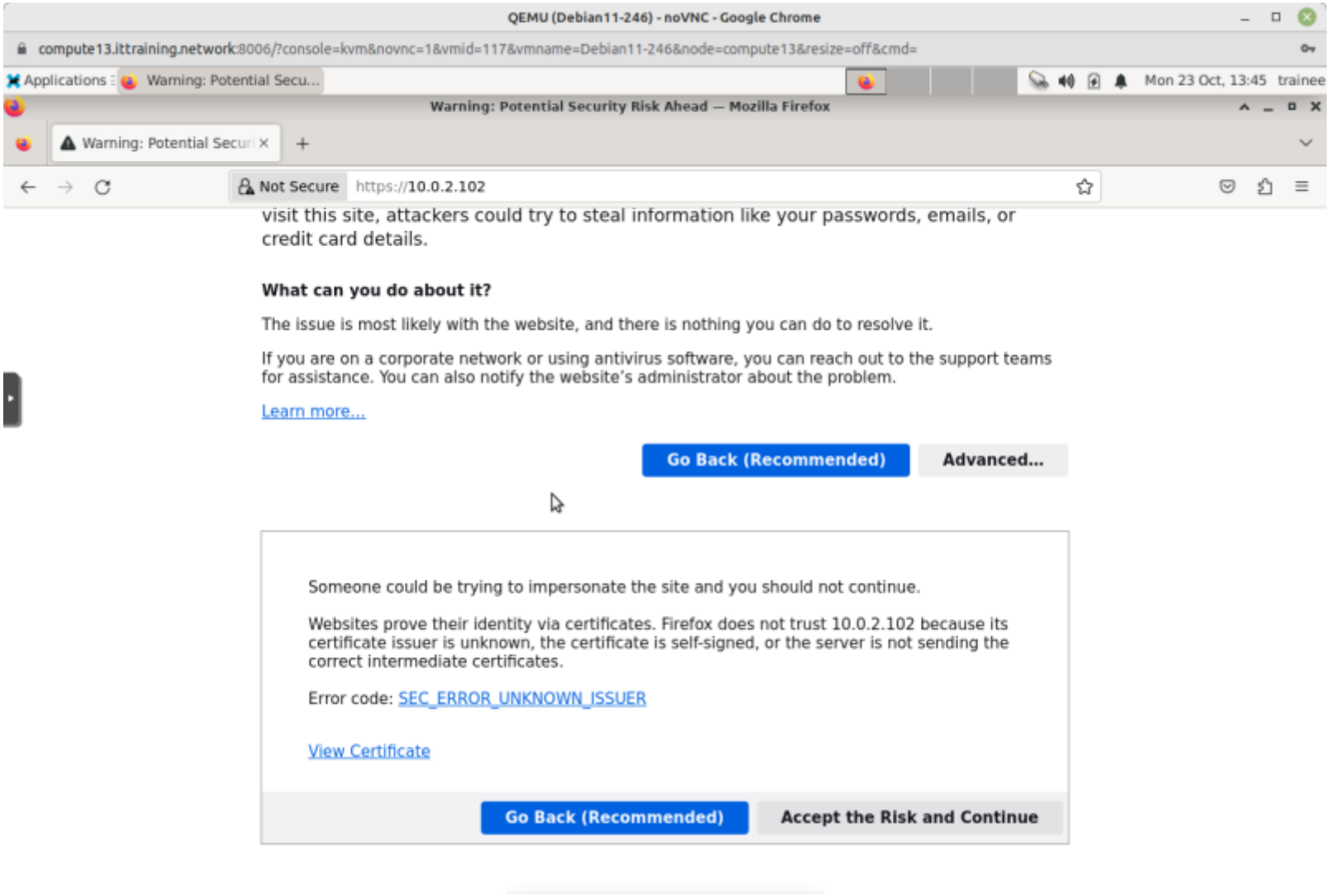
Connectez-vous à votre VM **Debian_10.0.2.46_VNC** :



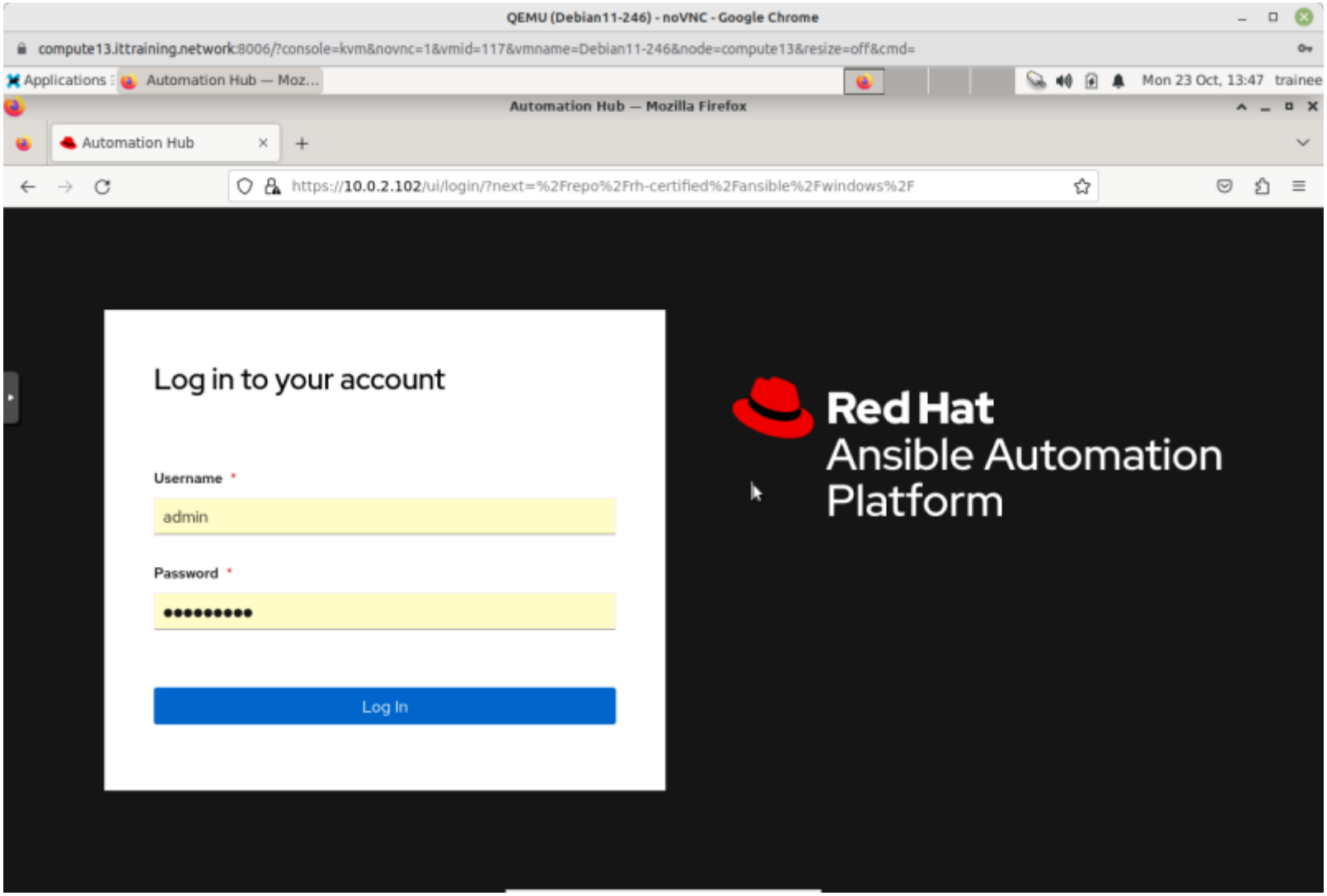
Ouvrez un navigateur Web et naviguez à **https://10.0.2.102** :



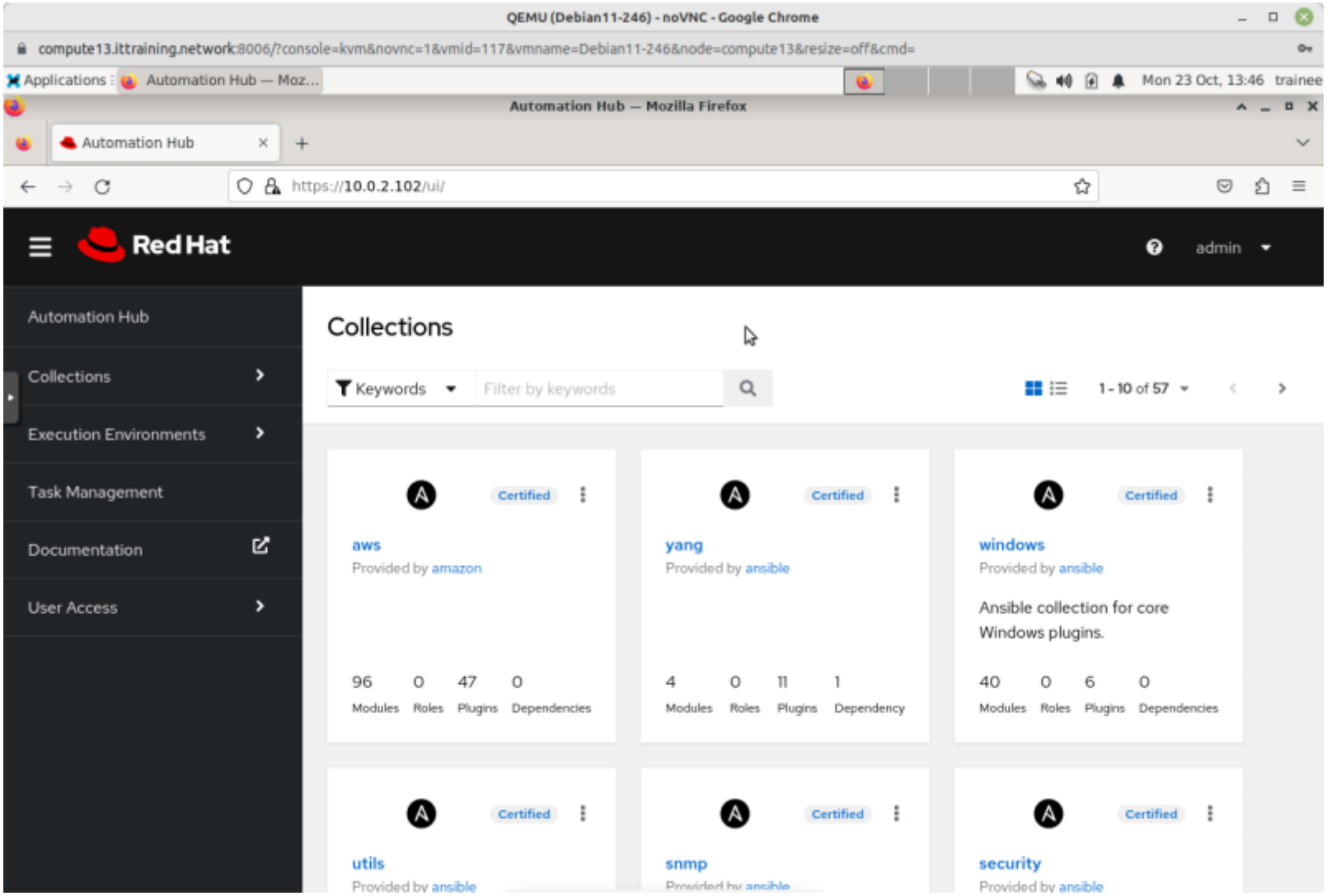
Acceptez le certificat auto-signé :



Connectez-vous à votre Ansible® Automation Hub en utilisant le compte **admin** et le mot de passe **fenestros** :



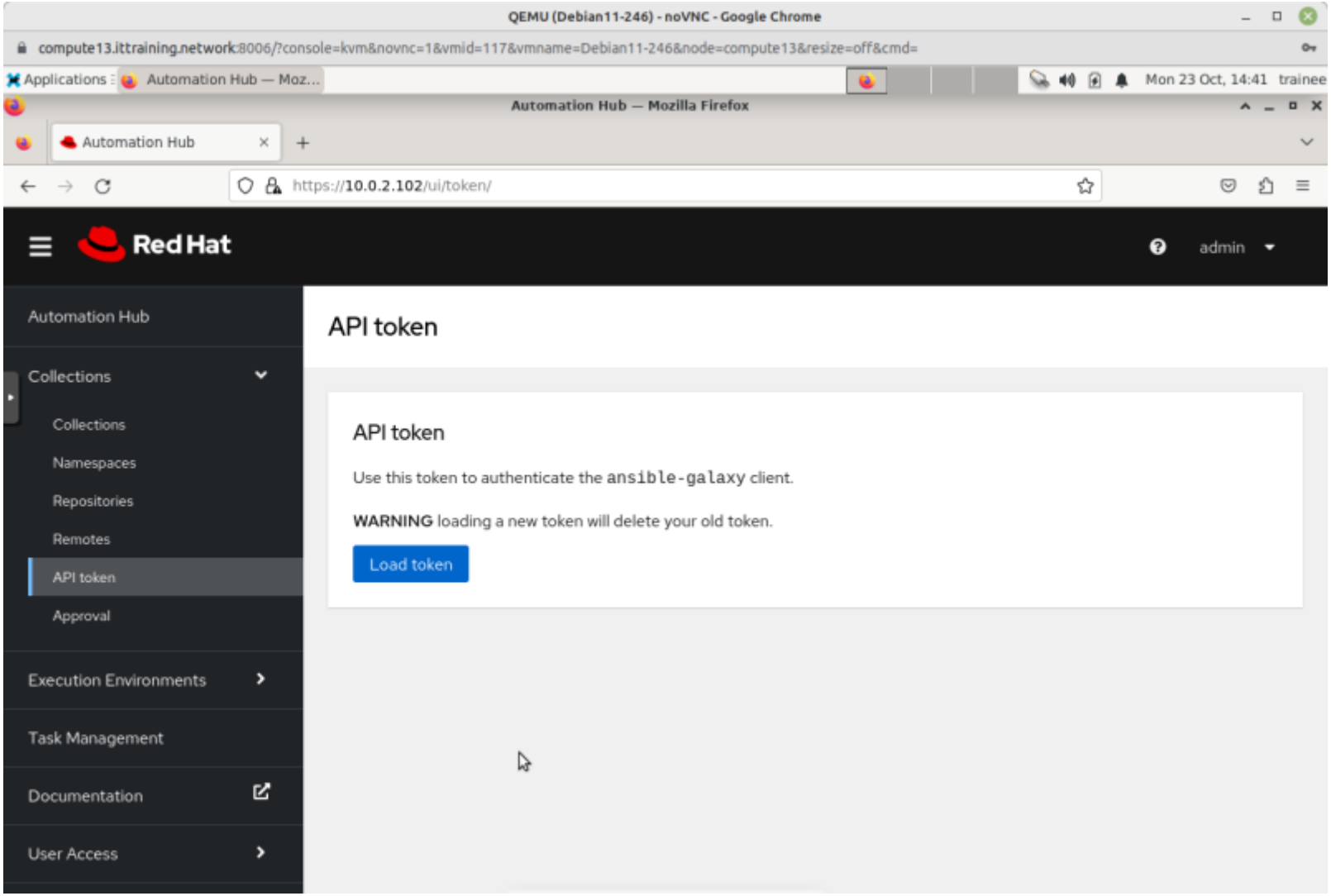
Vous obtiendrez :



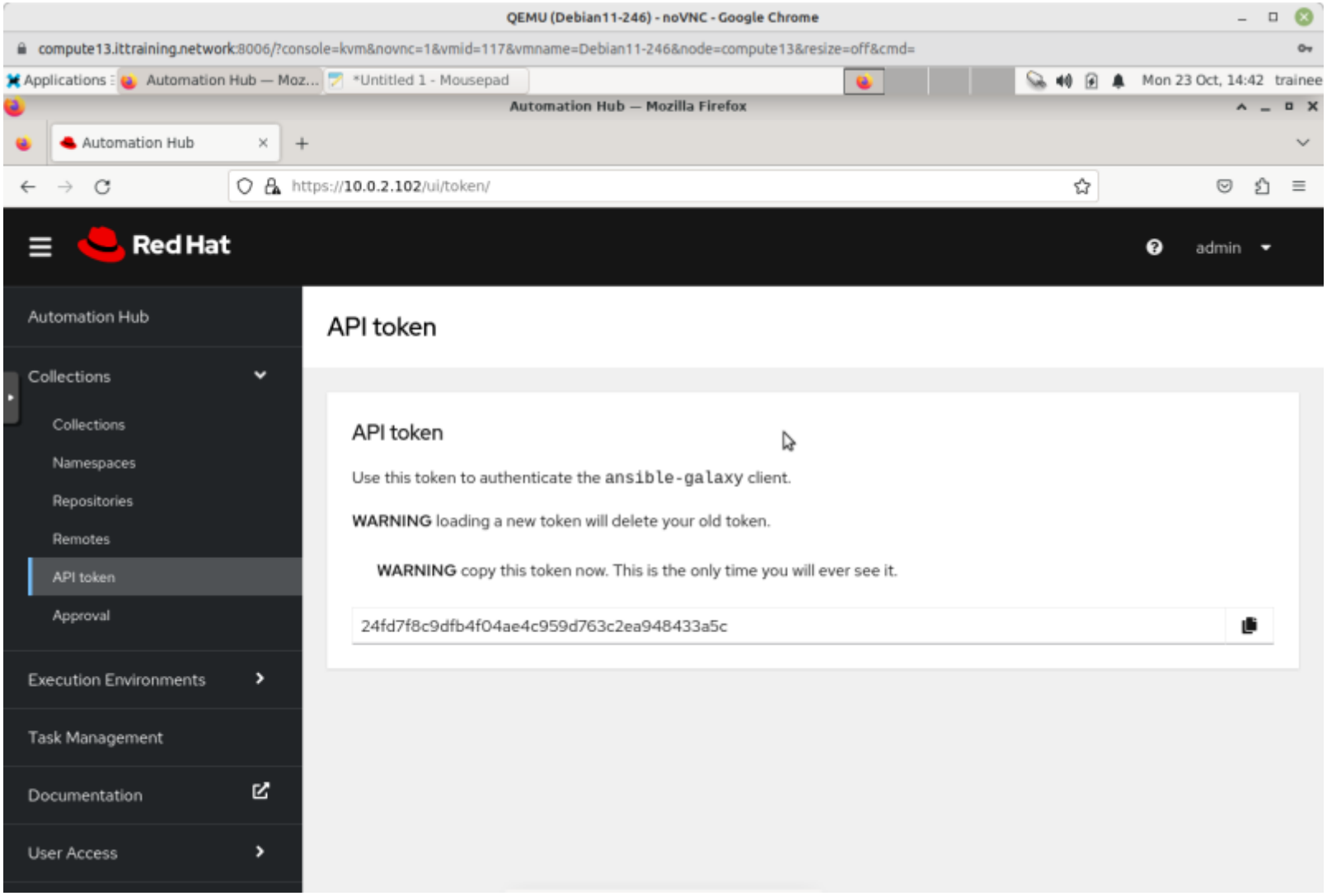
LAB #3 - Connecter l'Ansible® Automation Hub et l'Ansible® Automation Controller

3.1 - Configurer Ansible® Automation Hub

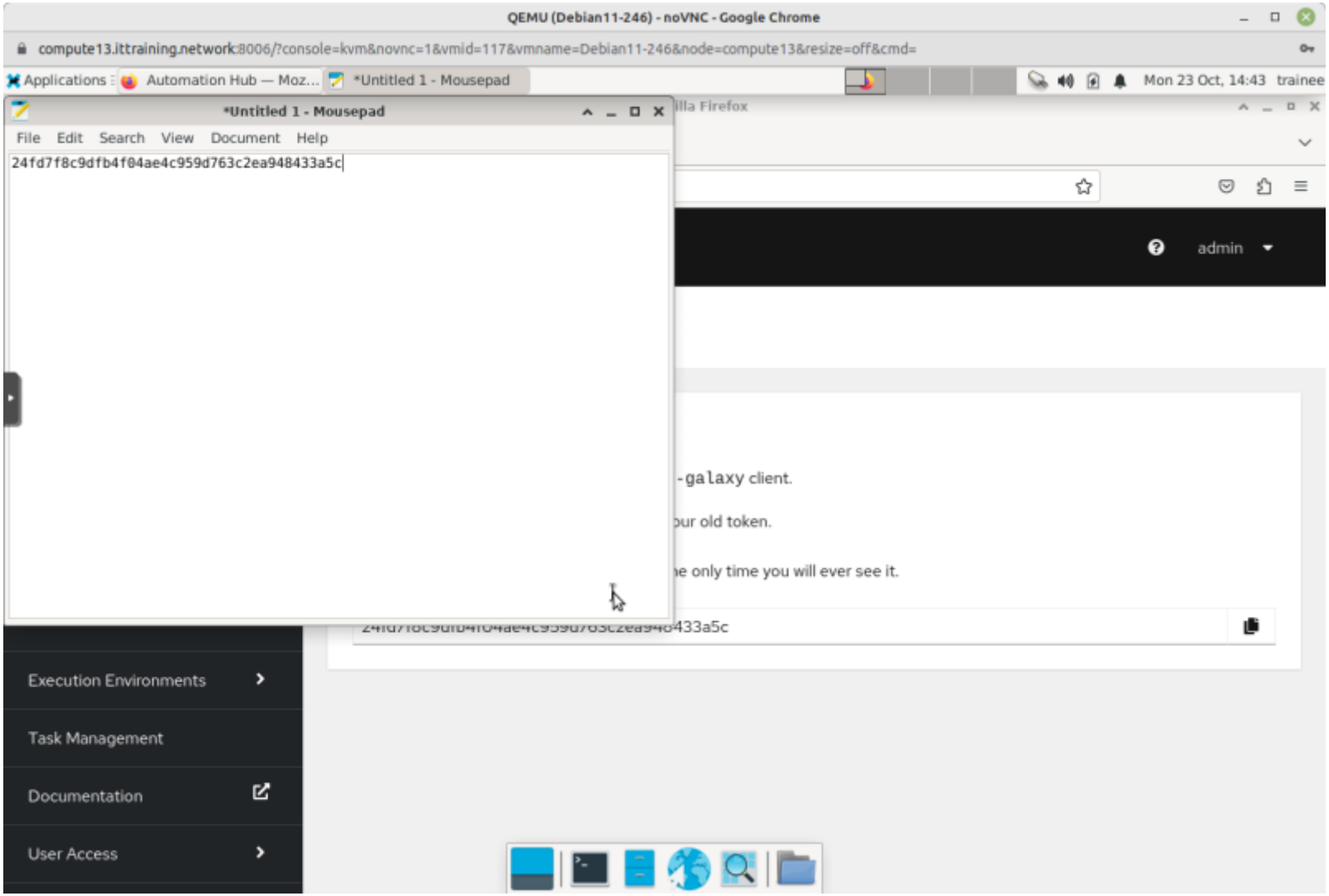
Dans l'interface de votre Ansible® Automation Hub, cliquez sur **Collections > API token > Load token** :



Copiez le token ainsi généré :

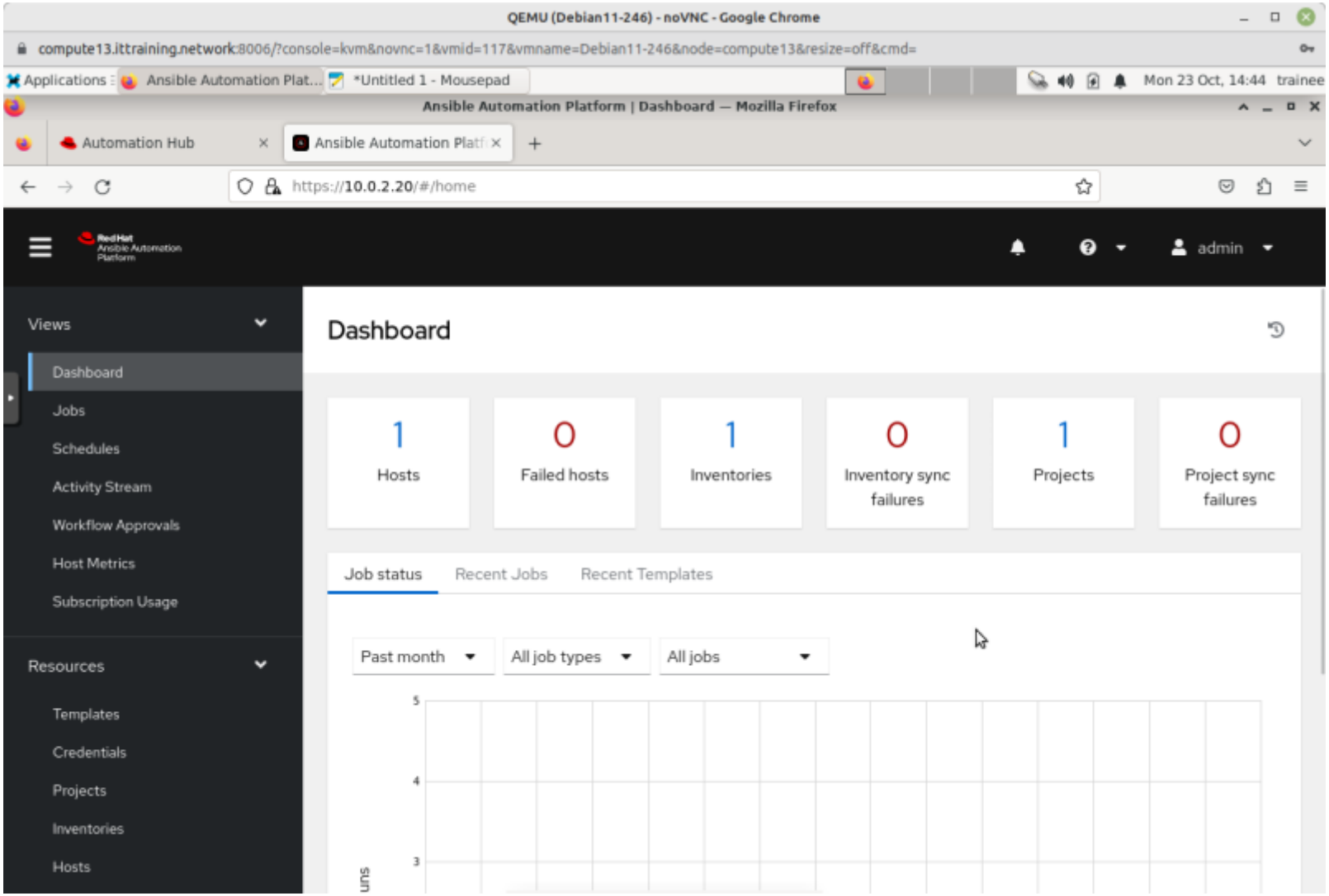


Collez ce token dans un bloc-note (mousepad) :

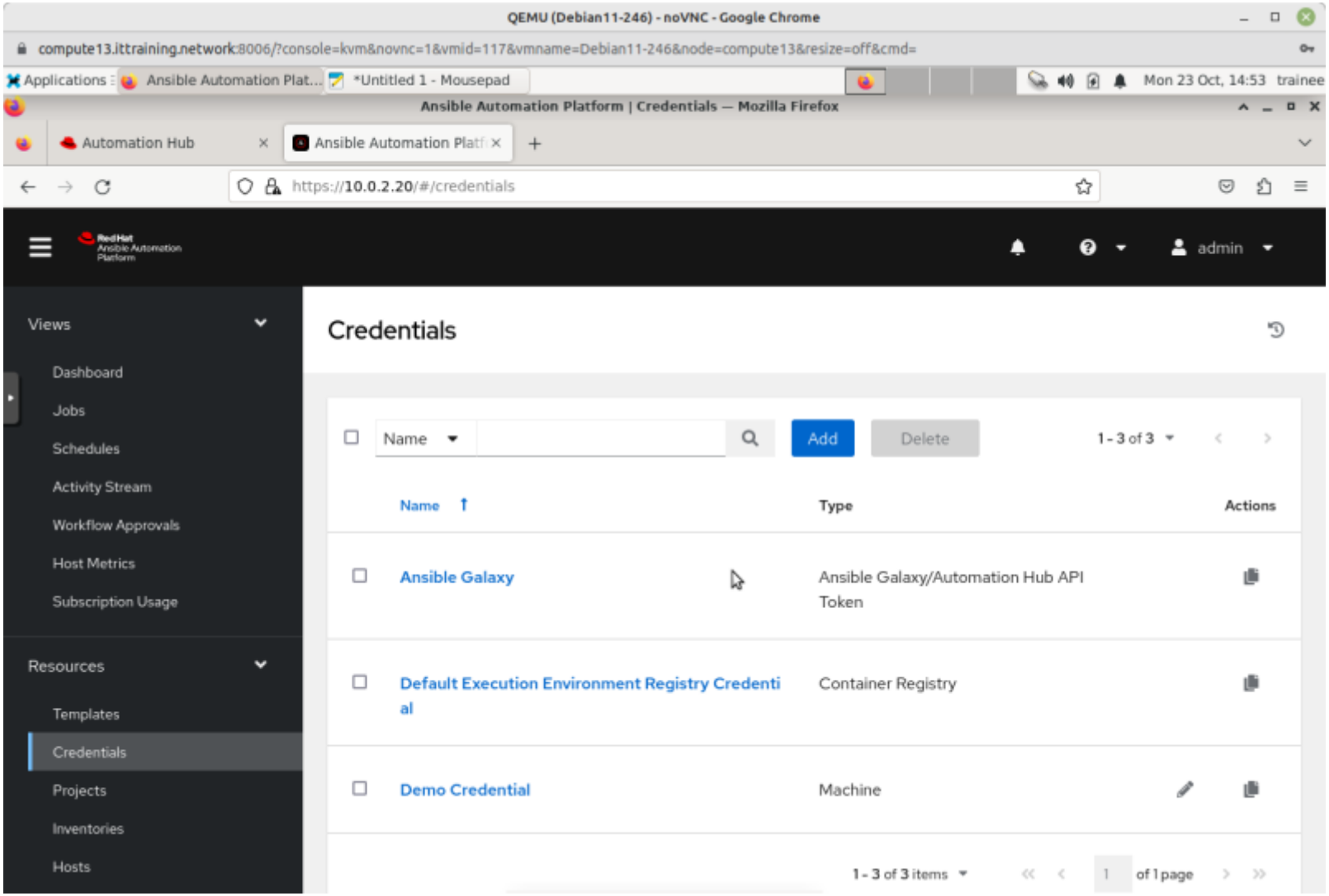


3.2 - Configurer Ansible® Automation Controller

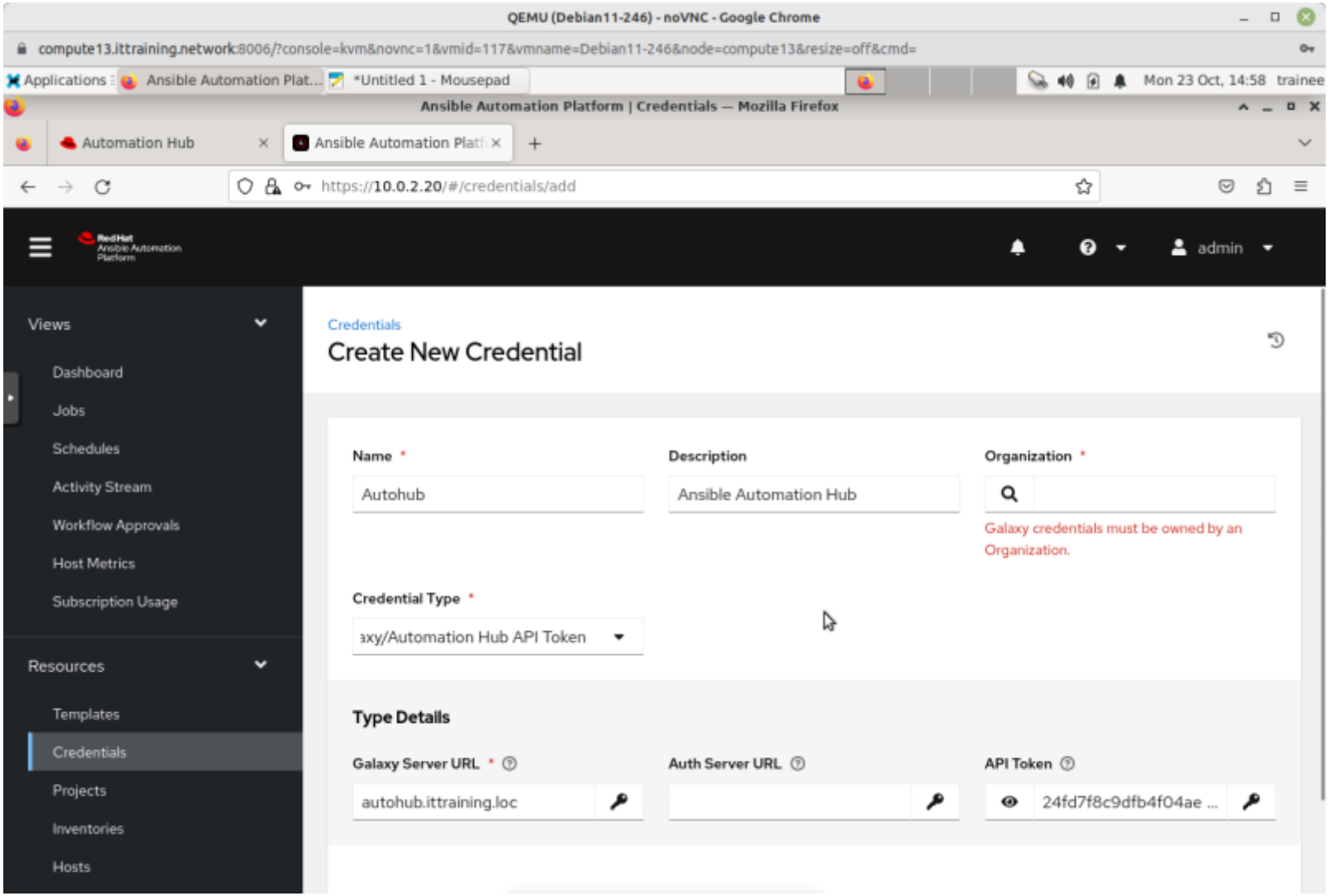
Ouvrez l'interface de votre Ansible® Automation Controller :



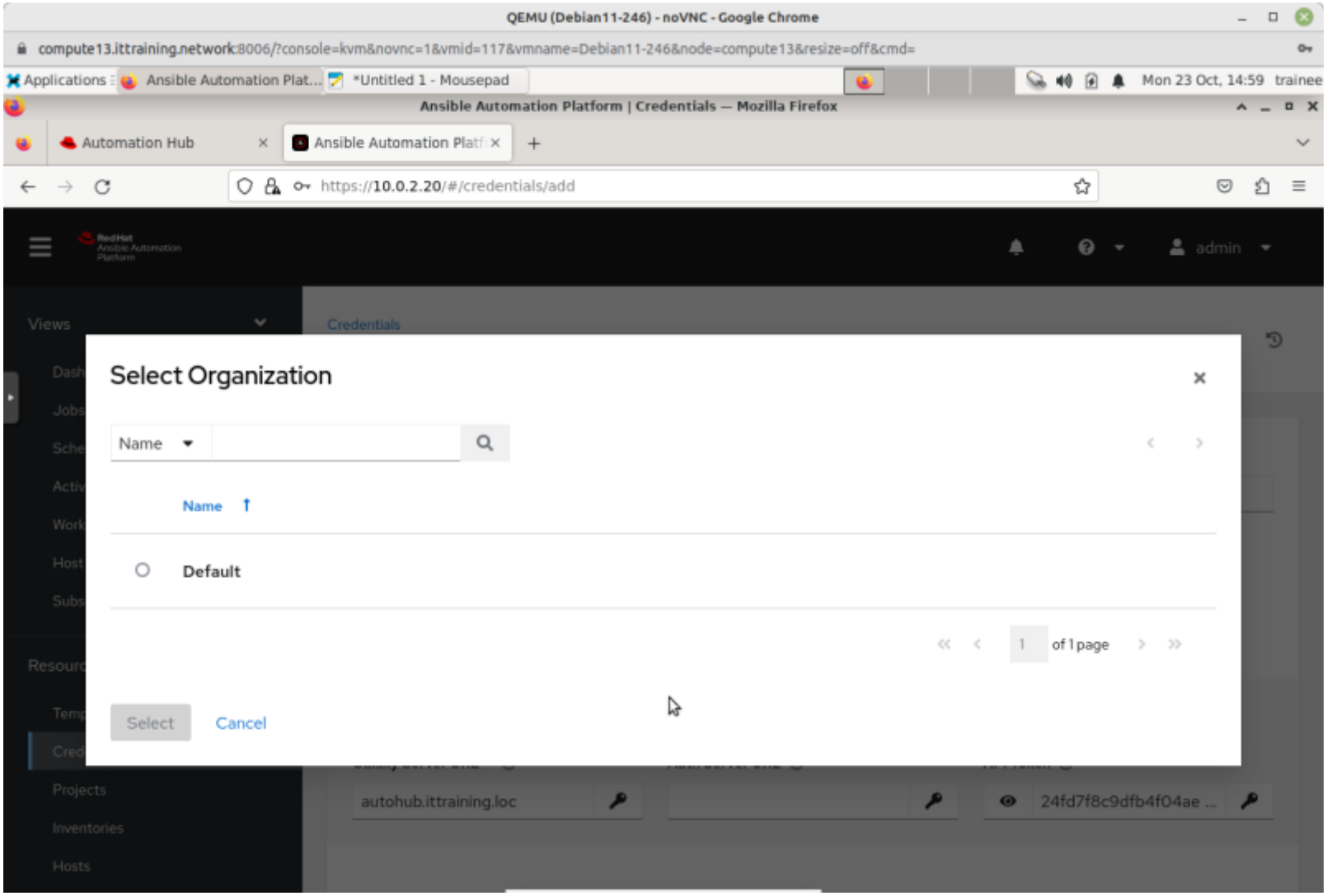
Naviguez à **Resources > Credentials** et cliquez sur le bouton **Add** :



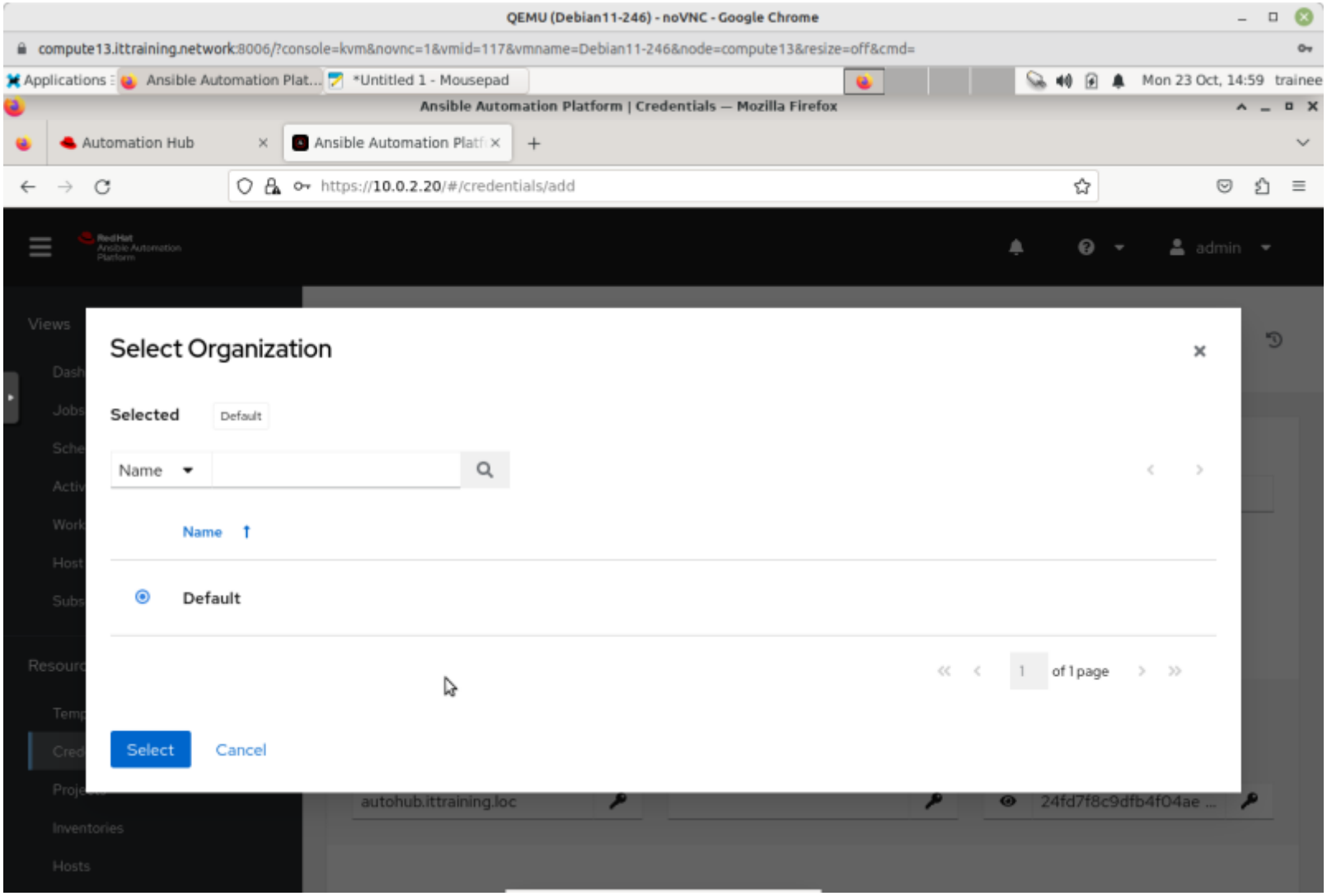
Renseignez les champs **Name**, **Description** et **Galaxy Server URL**. Choisissez **Galaxy/Automation Hub API Token** dans le menu déroulant **Credential Type** et collez le token API dans **API token** :



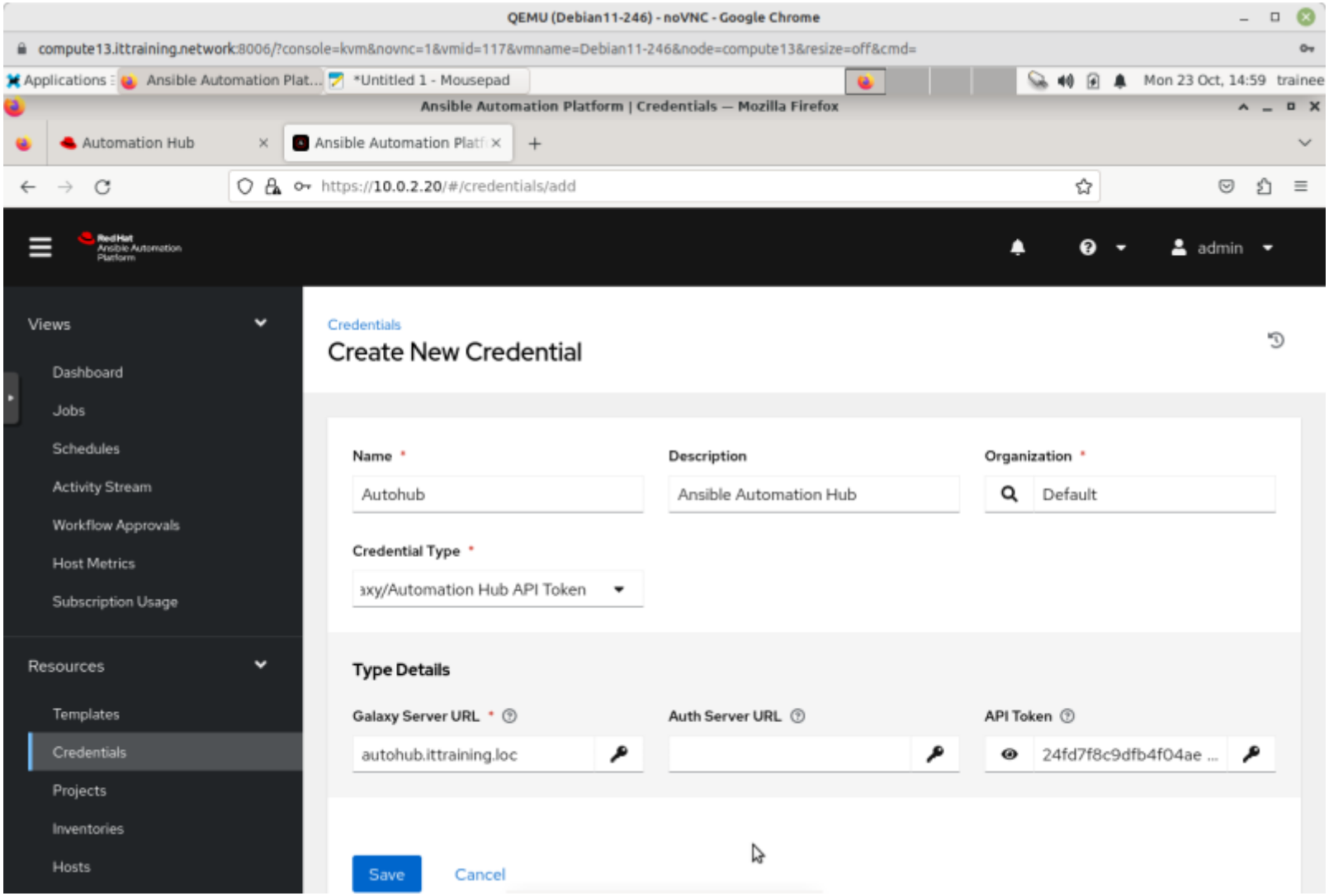
Cliquez sur la loupe à gauche du champs **Organization** :



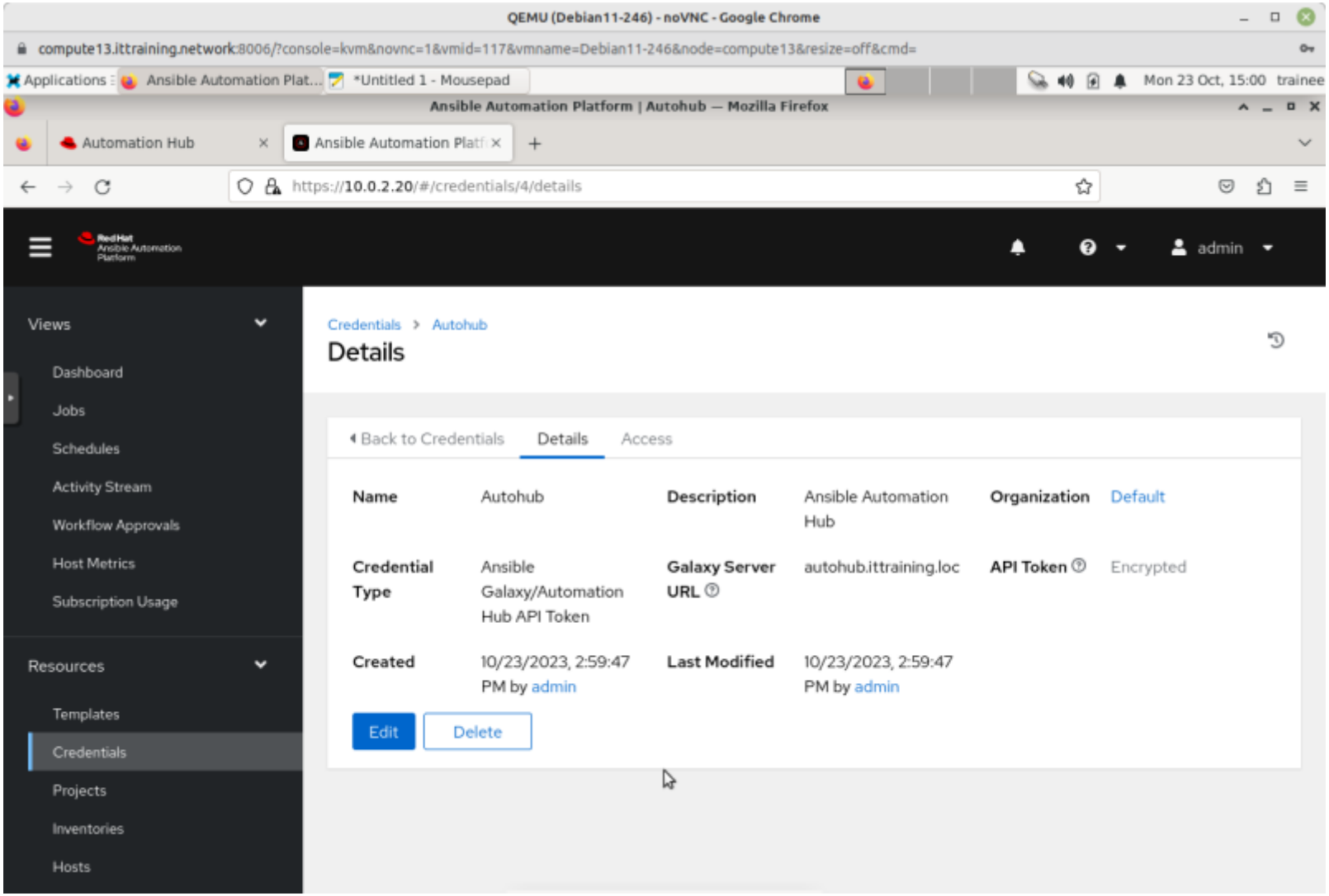
Cochez l'organisation **Default** et cliquez sur le bouton **Select** :



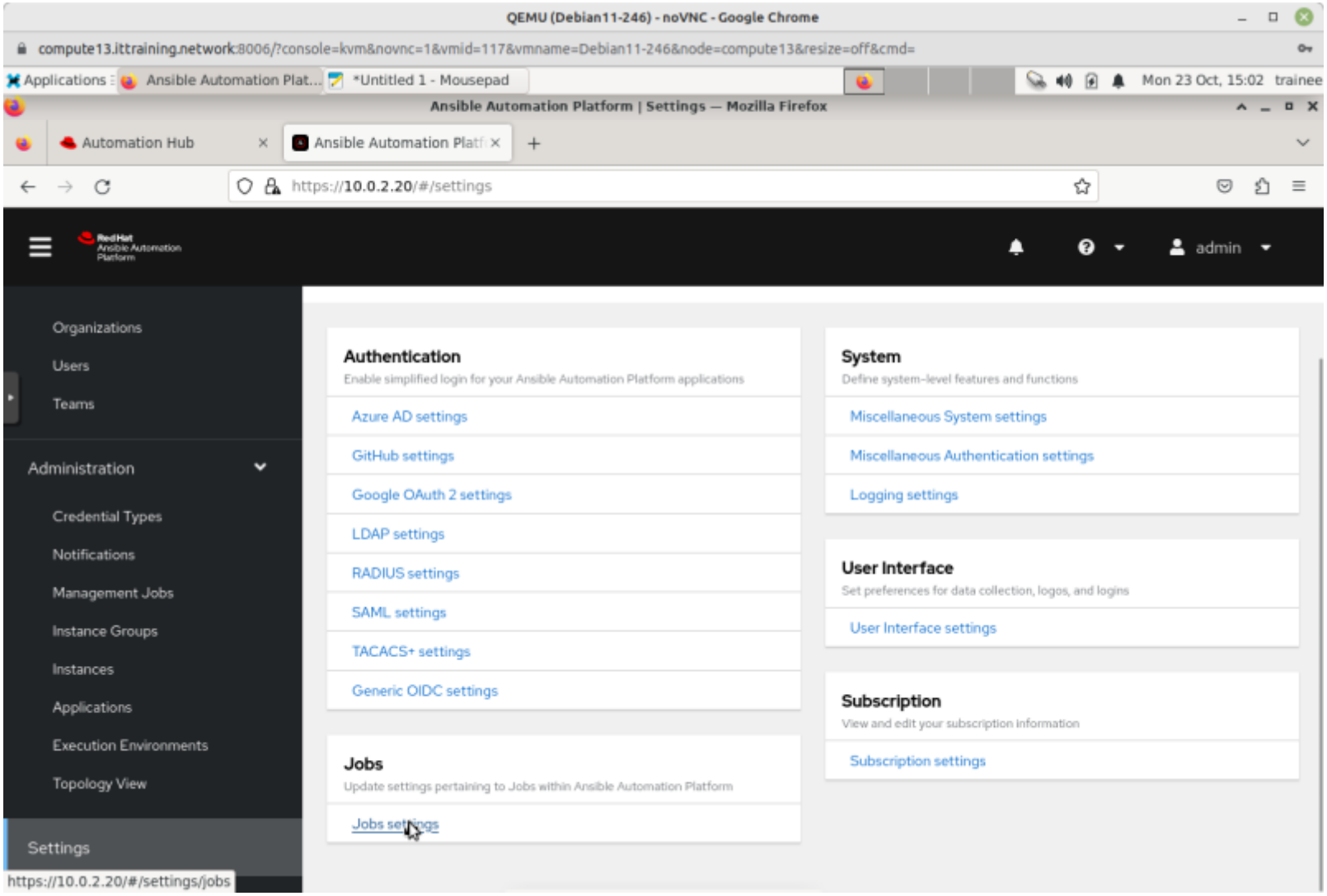
Validez en cliquant sur le bouton **Save** :



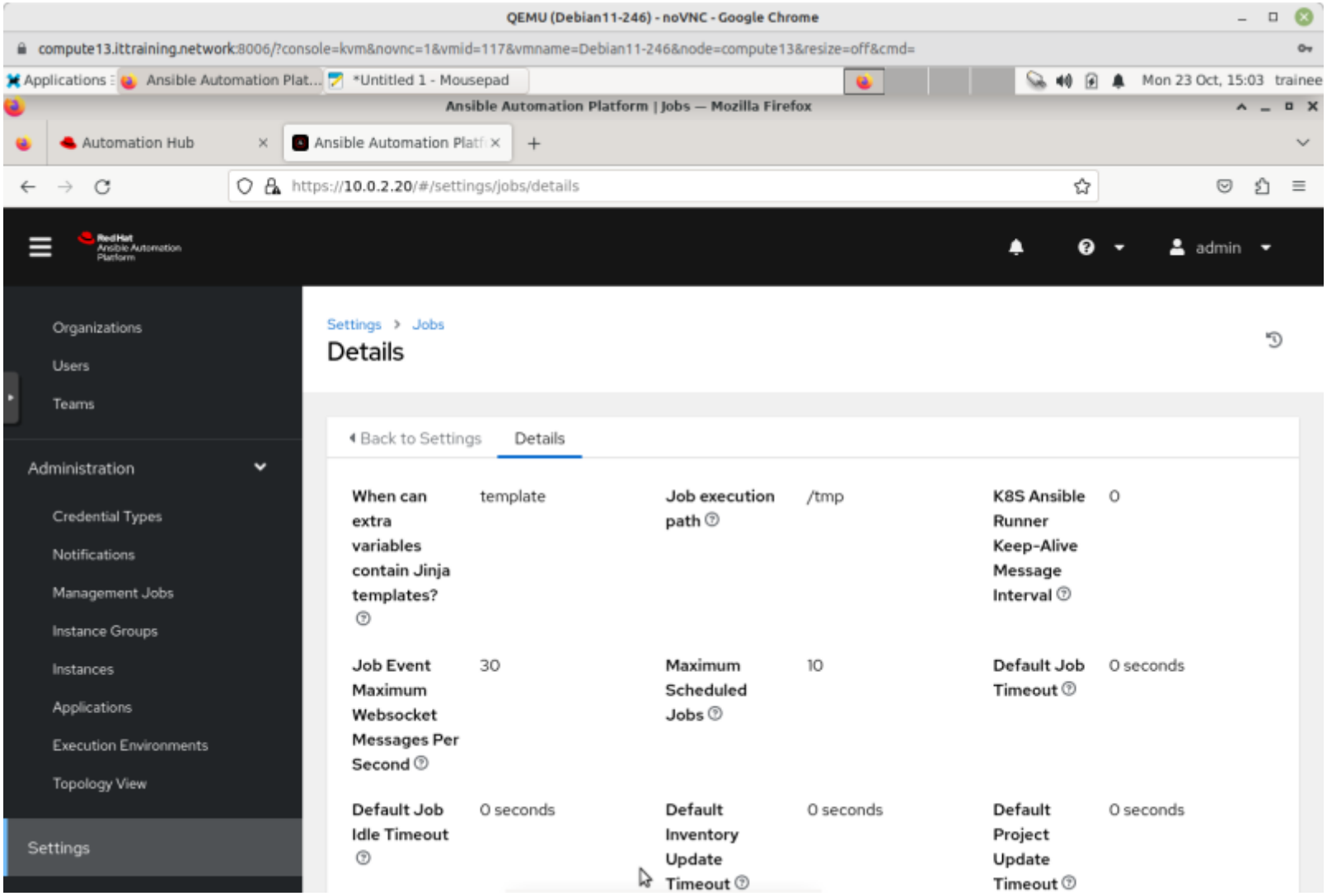
Vous obtiendrez :



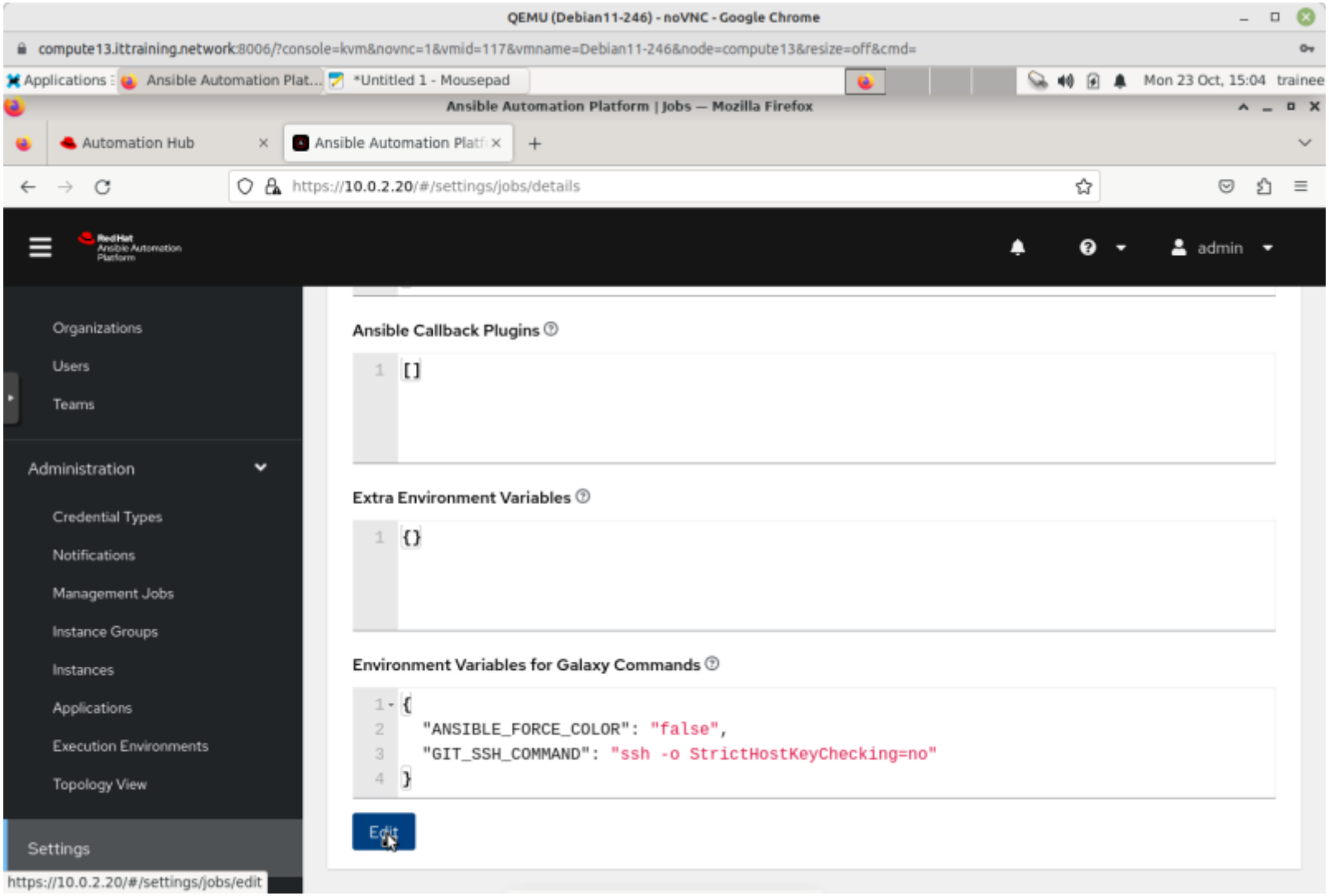
Cliquez sur **Settings > Jobs settings** :



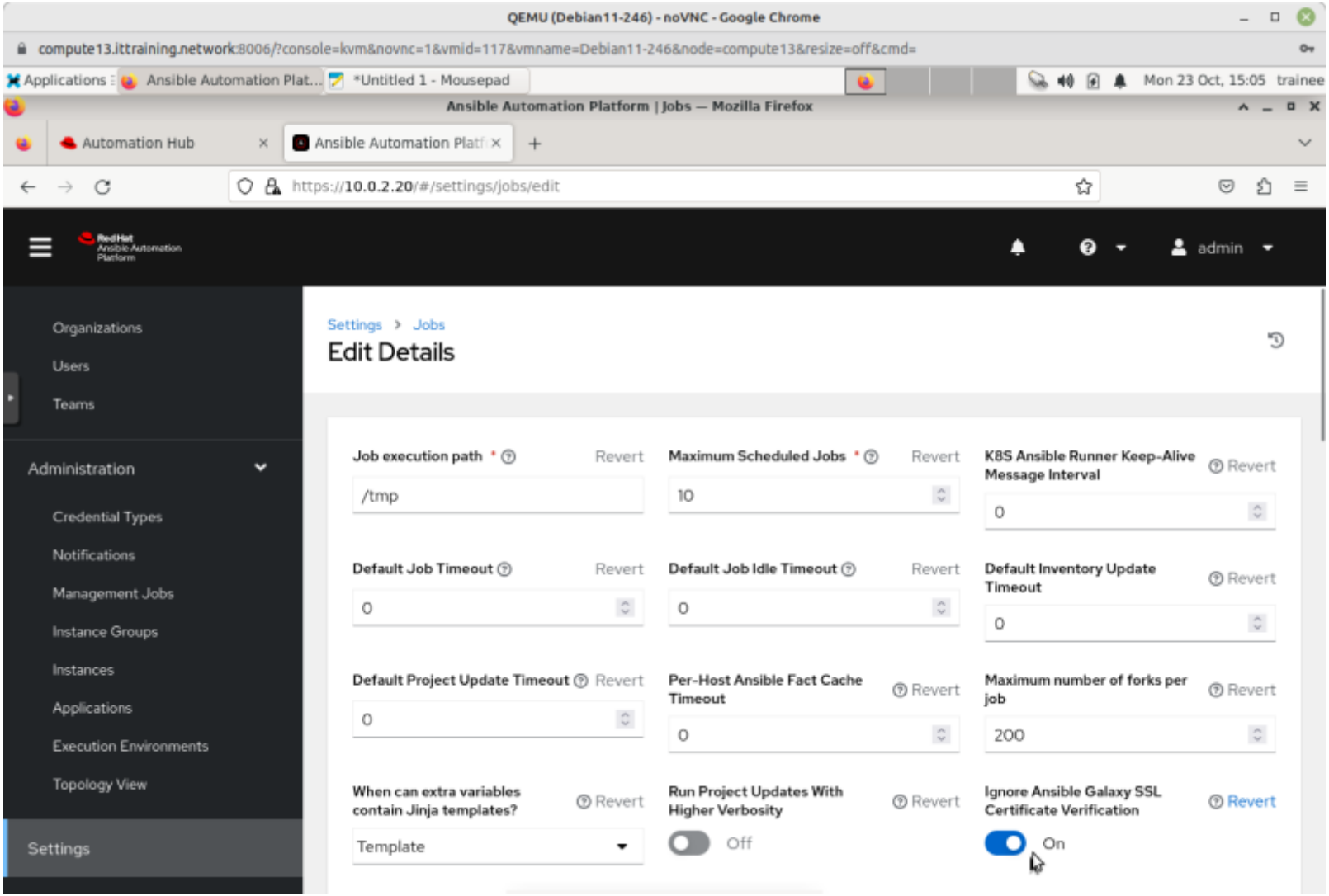
Vous obtiendrez :



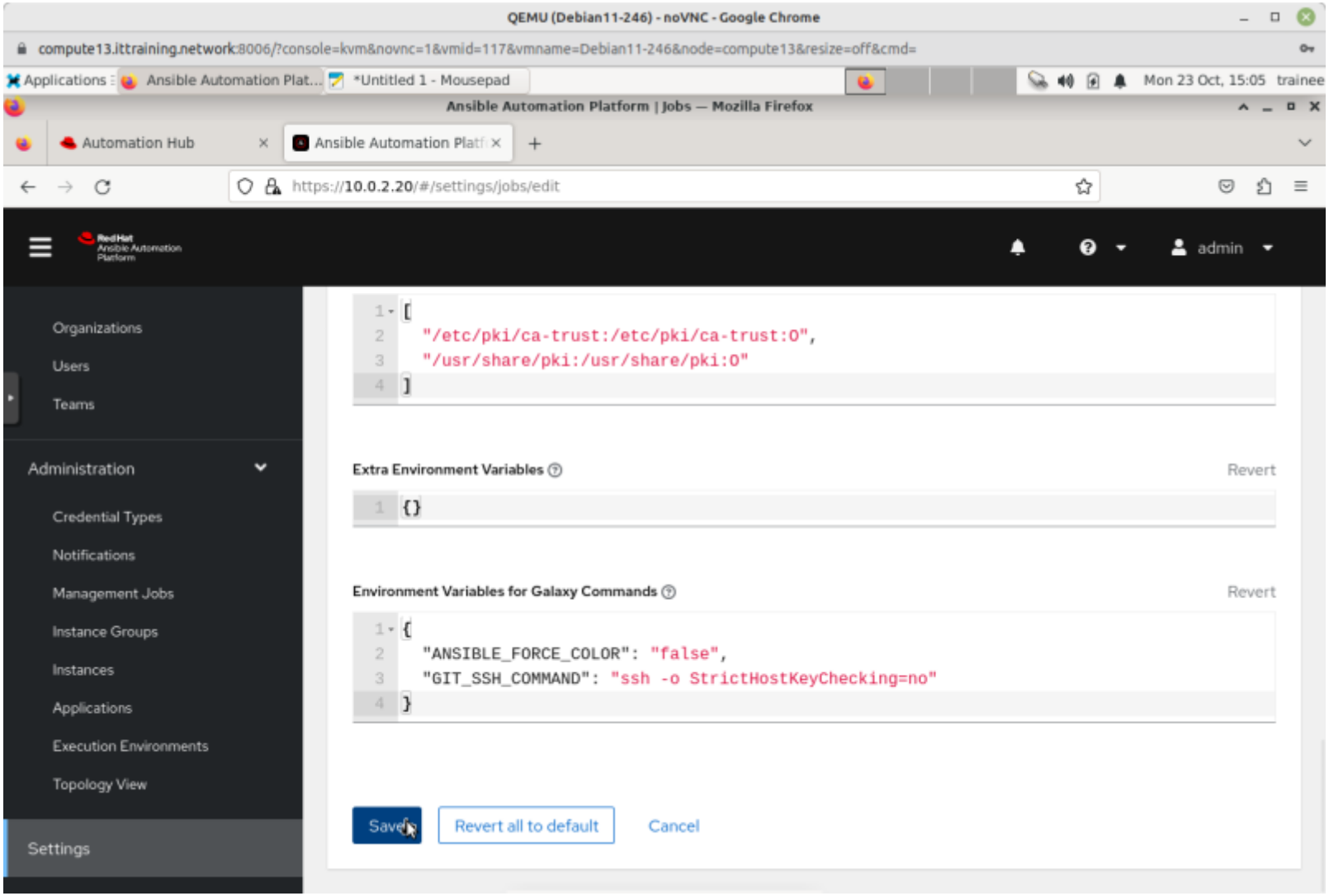
Descendez la page et cliquez sur le bouton **Edit** :



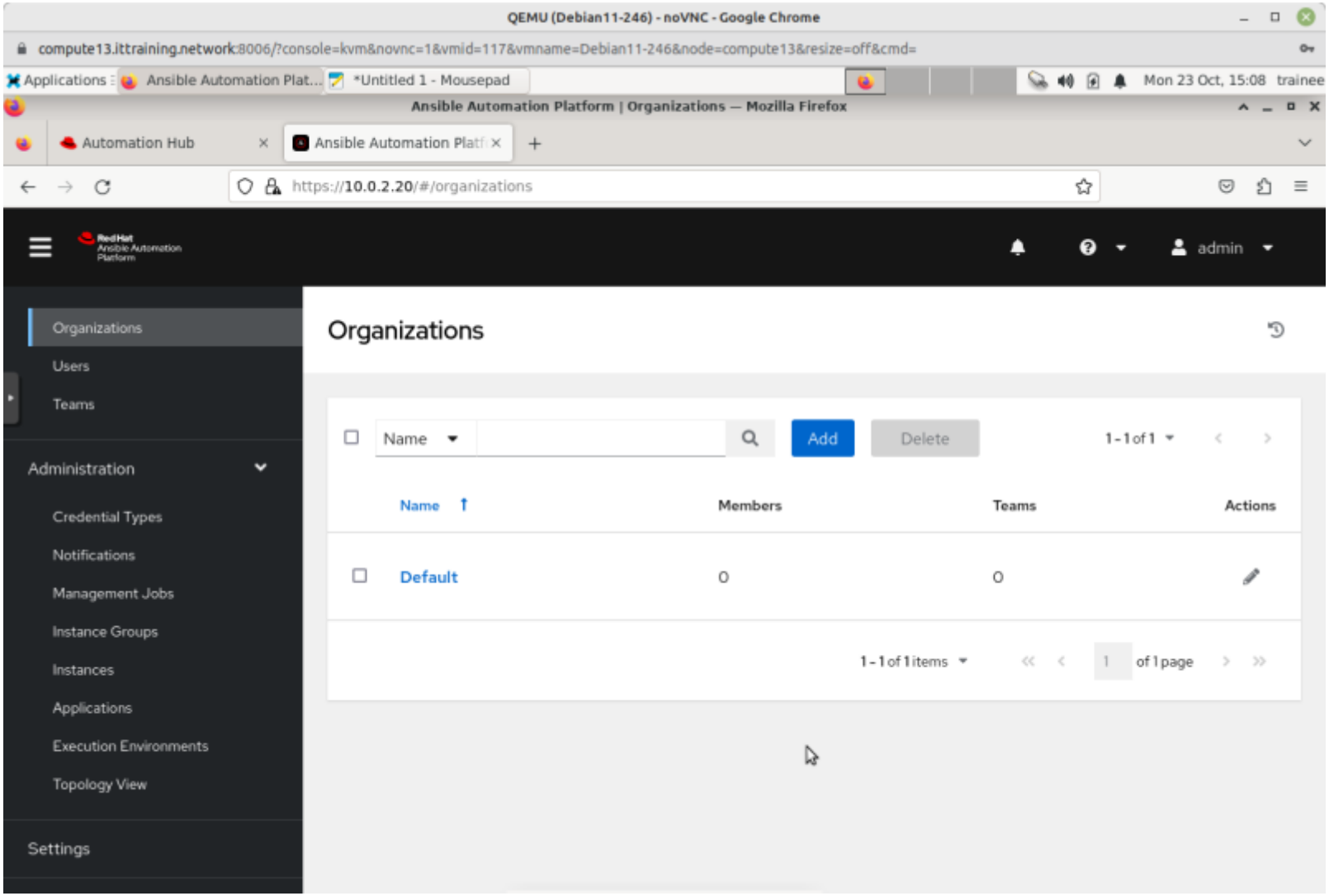
Activez l'option **Ignore Ansible Galaxy SSL Certificate Validation** pour permettre l'utilisation de votre certificat auto-signé :



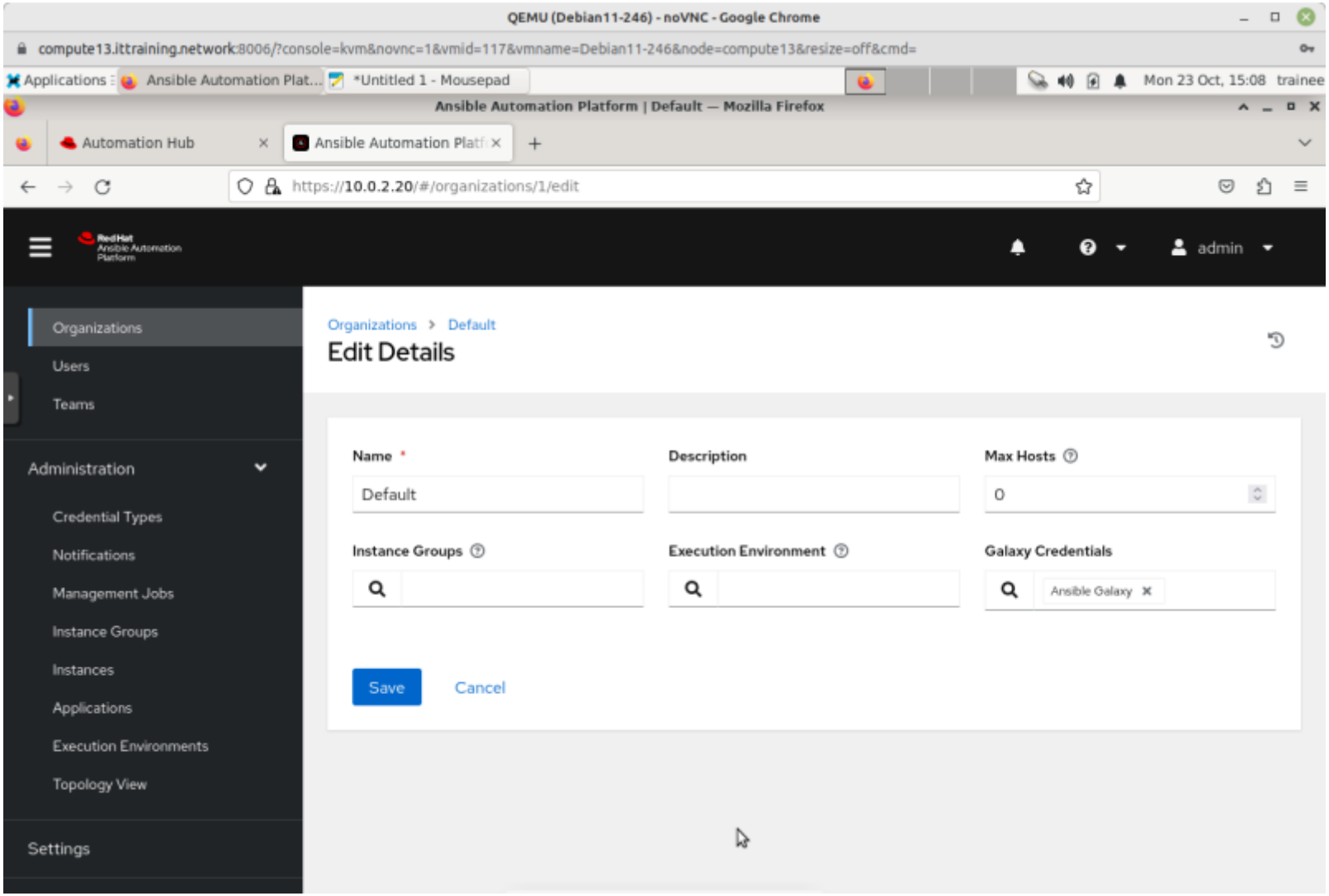
Descendez en bas de la page et cliquez sur le bouton **Save** :



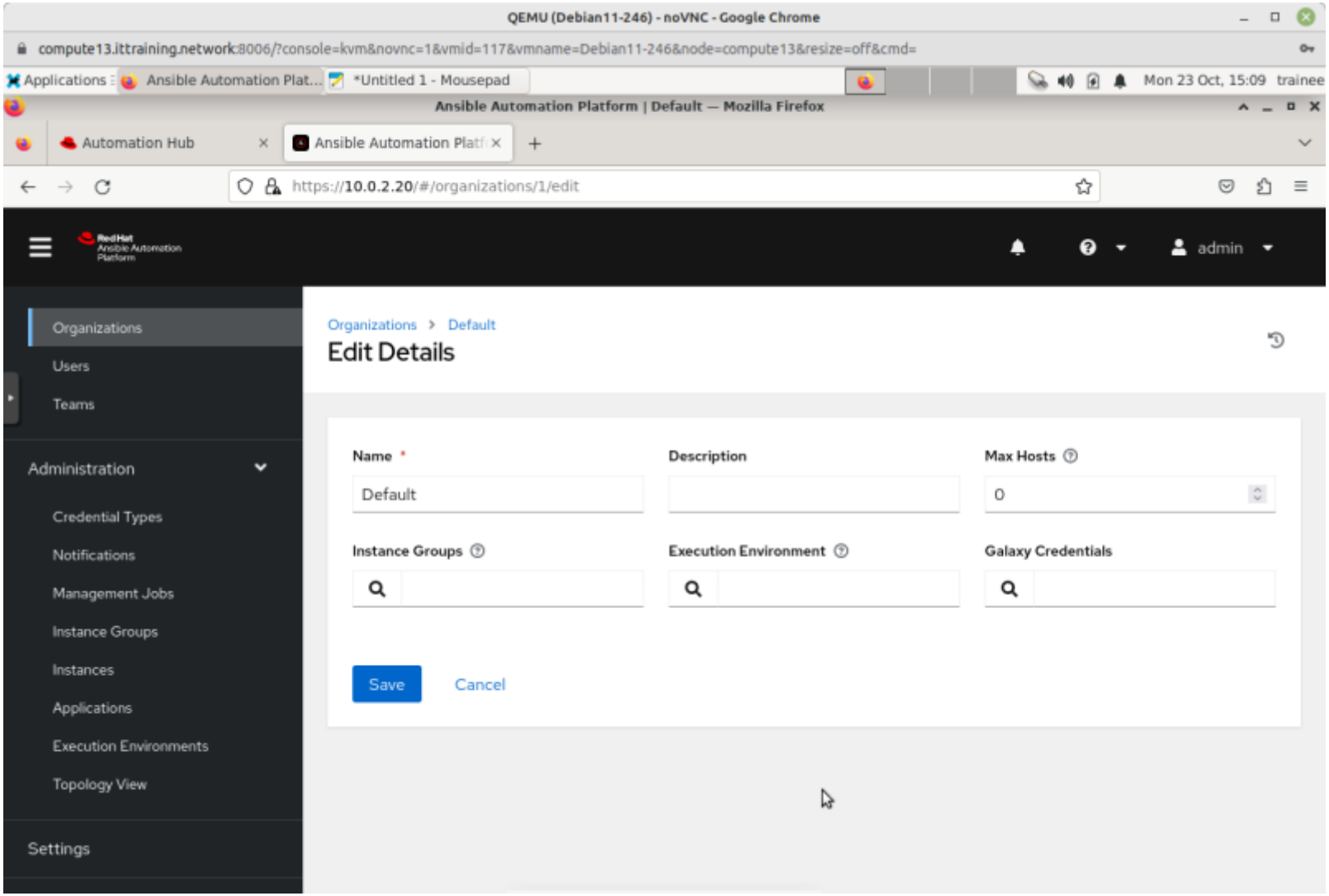
Naviguez à **Organizations** et cliquez sur l'icône d'un stylo à droite du nom **Default** :



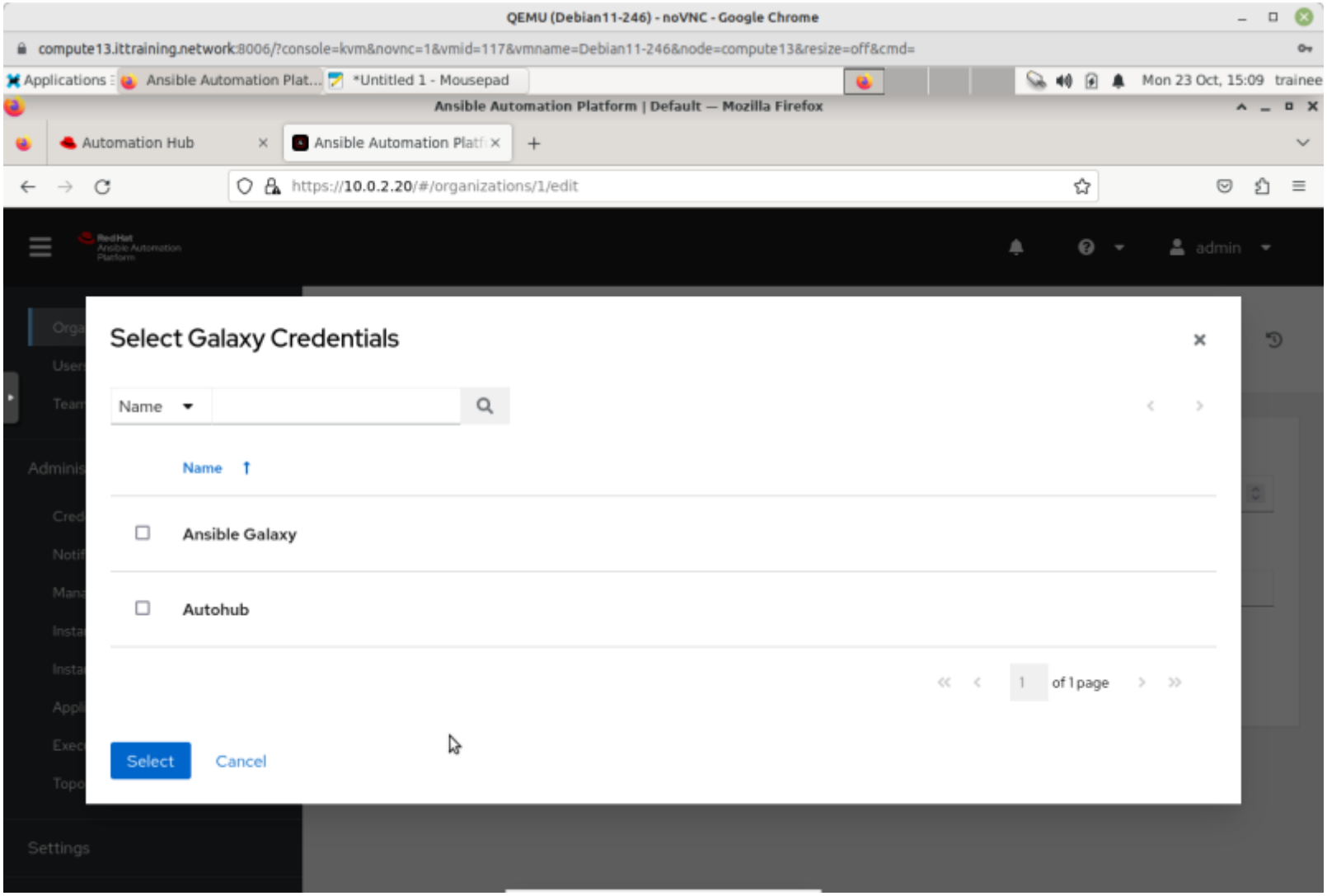
Dans le champs **Galaxy Credentials**, supprimez **Ansible Galaxy** :



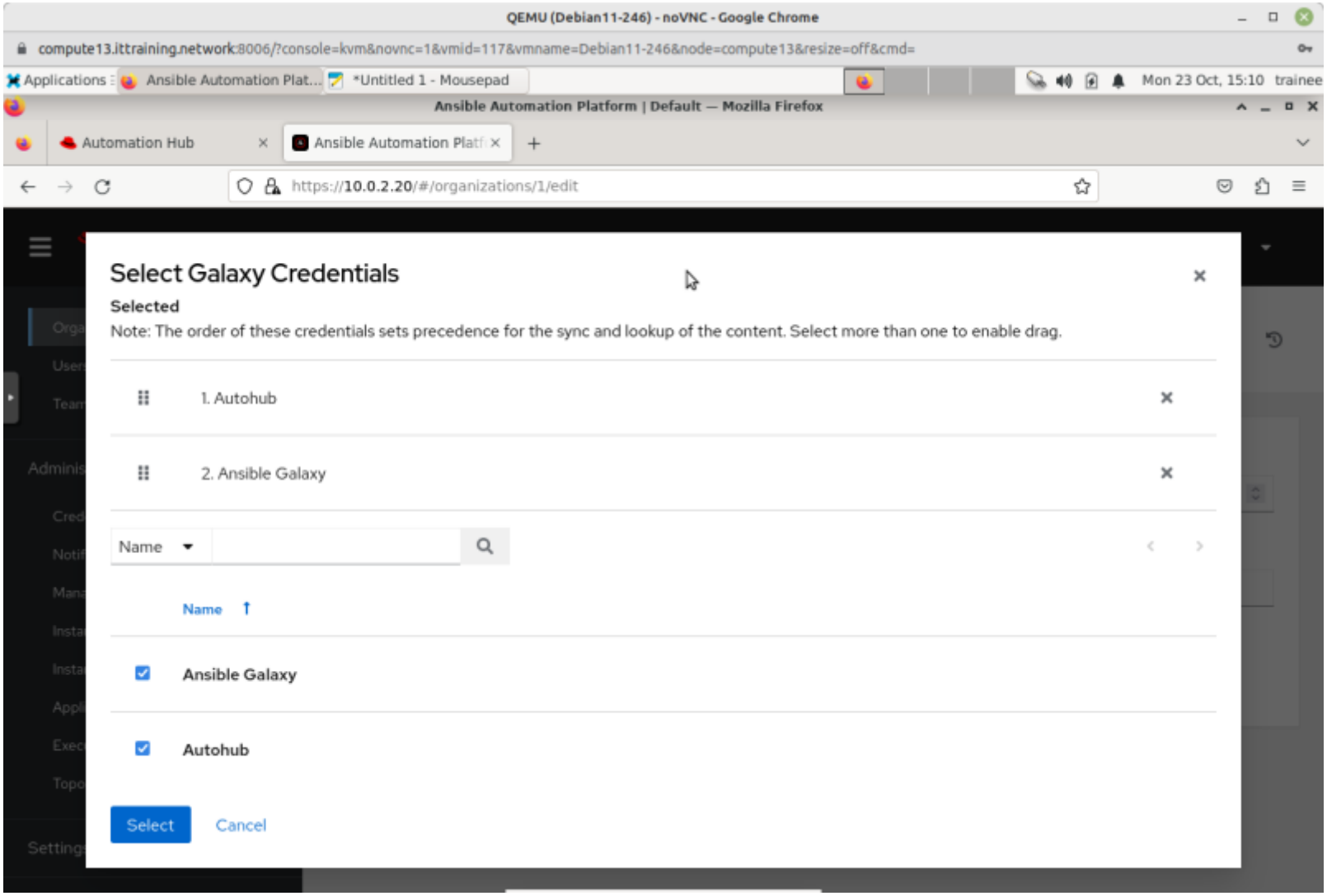
Cliquez sur la loupe à gauche du champs **Galaxy Credentials** :



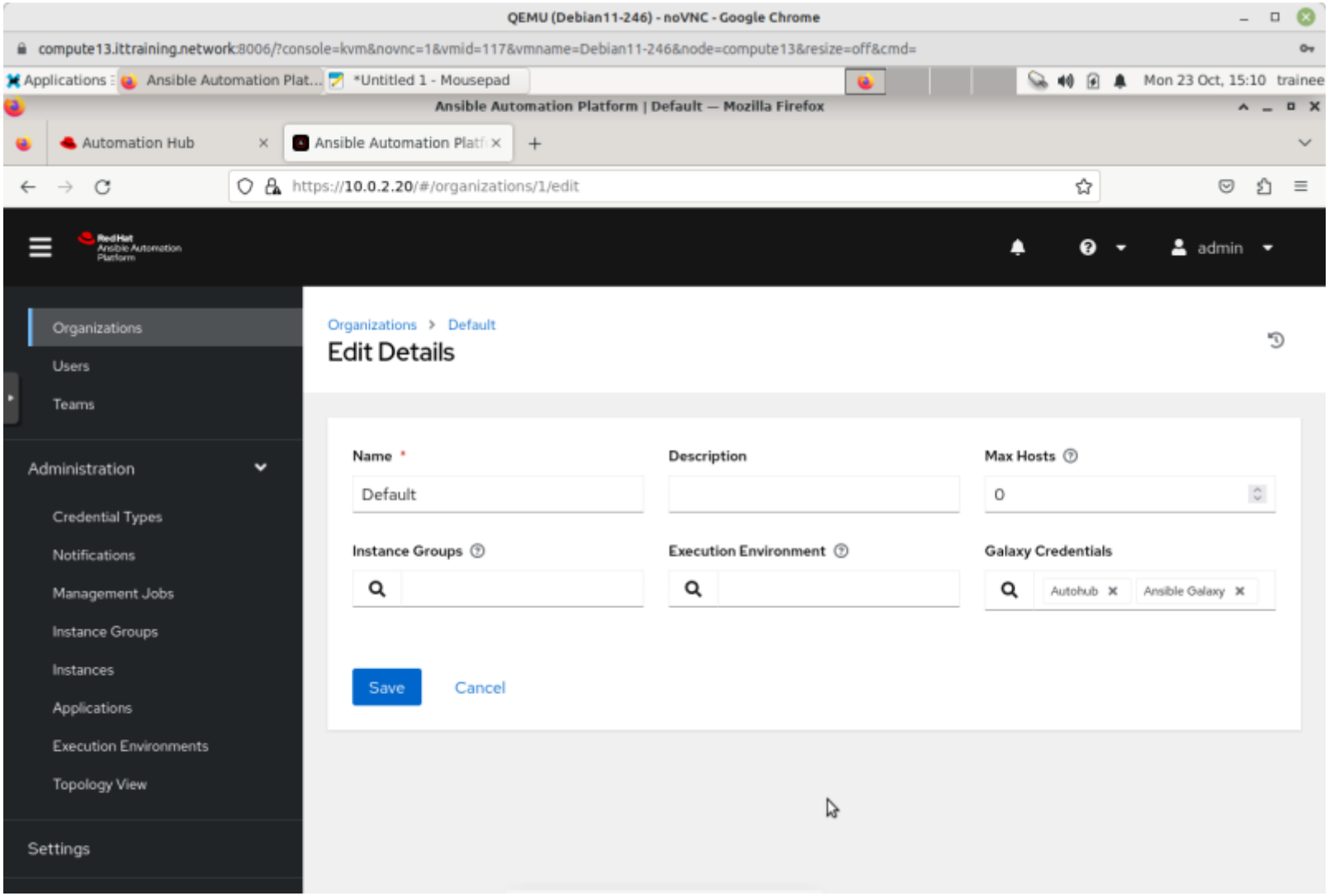
Vous obtiendrez :



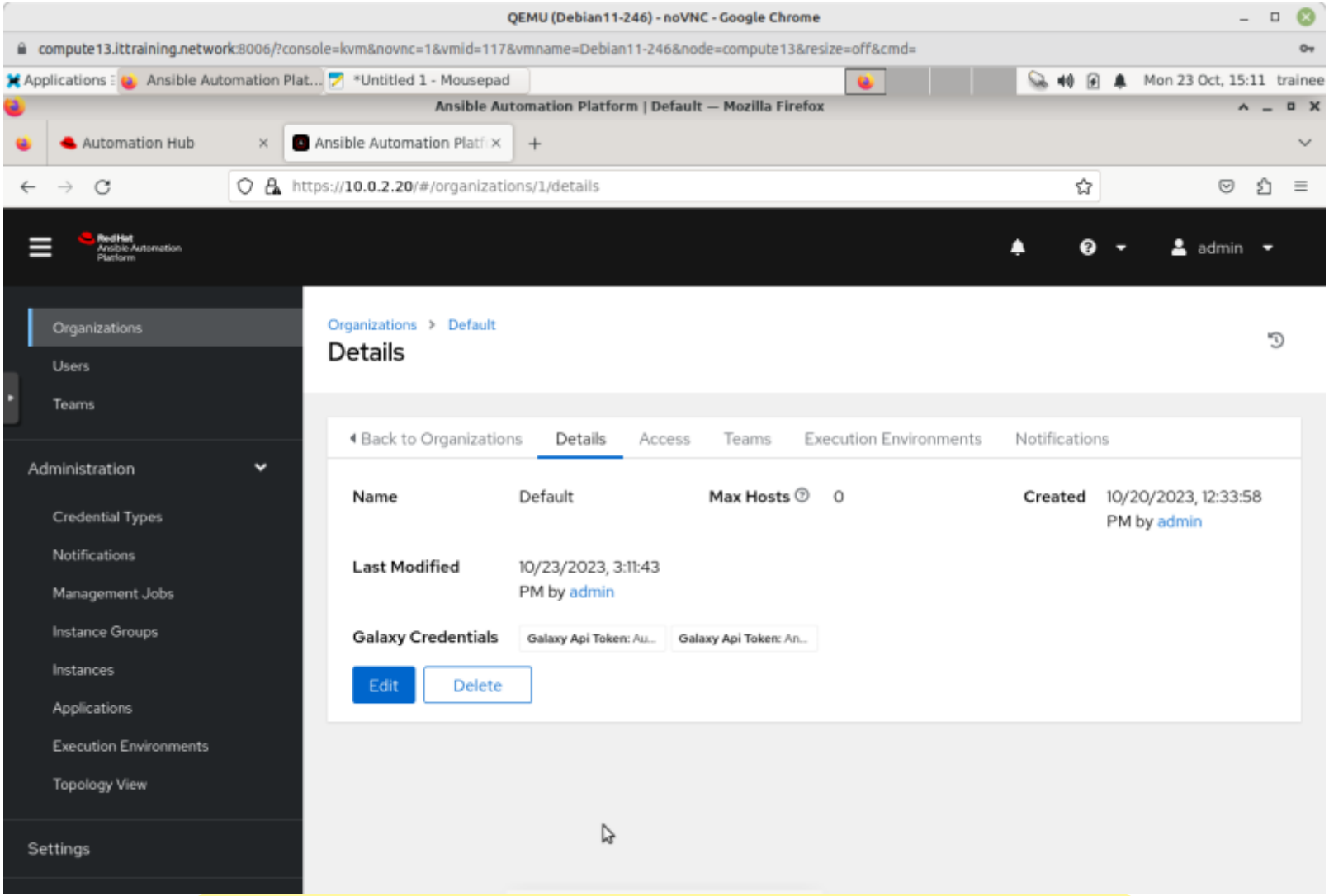
Cochez **Autohub** PUIS **Ansible Galaxy** puis cliquez sur le bouton **Select** :



Cliquez ensuite sur le bouton **Save** :



Vous obtiendrez :



Good News : Votre Ansible® Automation Hub et votre Ansible® Automation Controller sont maintenant connectés.

LAB #4 - Sauvegarder et Restaurer la Plateforme Ansible® Automation

4.1 - Sauvegarder l'Ansible® Automation Controller

Revenez dans votre Ansible® Automation Controller.

La sauvegarde de l'Ansible® Automation Controller s'effectue en utilisant le même script utilisé pour l'installation auquel il convient de passer l'option **-b** :

```
[root@redhat9 ~]# cd ansible-automation-platform-setup-bundle-2.4-2.2-x86_64/
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ./setup.sh -b
...
PLAY RECAP *********************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
redhat9.ittraining.loc     : ok=91   changed=49   unreachable=0    failed=0    skipped=90   rescued=0
ignored=0

The setup process completed successfully.
Setup log saved to /var/log/tower/backup-2023-10-24-10:43:16.log.
```

A l'issue de la sauvegarde vous constaterez la présence d'une archive **tar.gz** ainsi qu'un lien symbolique dénommé **automation-platform-backup-latest.tar.gz** dans le répertoire **courant** :

```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ls -l
total 108
-rw-------. 1 root root 74610 Oct 24 10:43 automation-platform-backup-2023-10-24-10:43:19.tar.gz
lrwxrwxrwx. 1 root root   117 Oct 24 10:43 automation-platform-backup-latest.tar.gz -> /root/ansible-automation-
platform-setup-bundle-2.4-2.2-x86_64/./automation-platform-backup-2023-10-24-10:43:19.tar.gz
drwxr-xr-x. 5 root root    55 Oct 13 17:00 bundle
drwxr-xr-x. 3 root root    33 Oct 13 16:56 collections
drwxr-xr-x. 2 root root    17 Oct 13 16:56 group_vars
```

```
-rw-r--r--. 1 root root  9063 Oct 20 12:21 inventory
-rw-r--r--. 1 root root   530 Oct 13 17:11 README.md
-rwxr-xr-x. 1 root root 14780 Oct 13 16:56 setup.sh
```

Dans ce cas, la taille de l'archive est de 76K :

```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# du -hs automation-platform-backup-*
76K     automation-platform-backup-2023-10-24-10:43:19.tar.gz
0       automation-platform-backup-latest.tar.gz
```

4.2 - Sauvegarder l'Ansible® Automation Hub

Connectez-vous maintenant à votre VM **autohub.ittraining.loc** :

```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ssh -l trainee 10.0.2.102
```

La sauvegarde de l'Ansible® Automation Hub s'effectue en utilisant le même script utilisé pour l'installation auquel il convient de passer l'option **-b** :

```
[root@autohub ~]# cd ansible-automation-platform-setup-bundle-2.4-2.2-x86_64/
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ./setup.sh -b
...
PLAY RECAP *********************************************************************
autohub.ittraining.loc     : ok=70   changed=29   unreachable=0    failed=0    skipped=114  rescued=0
ignored=0
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

The setup process completed successfully.
[warn] /var/log/tower does not exist. Setup log saved to backup.log.
[warn] Provided path does not exist or is not accessible. Setup log saved to ./backup.log.
```

A l'issue de la sauvegarde vous constaterez la présence d'une archive **tar.gz** ainsi qu'un lien symbolique dénommé **automation-platform-backup-**

latest.tar.gz dans le répertoire **courant** :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ls -l
total 612124
-rw-------. 1 root root 621857468 Oct 24 11:09 automation-platform-backup-2023-10-24-11:07:36.tar.gz
lrwxrwxrwx. 1 root root       117 Oct 24 11:09 automation-platform-backup-latest.tar.gz -> /root/ansible-
automation-platform-setup-bundle-2.4-2.2-x86_64/./automation-platform-backup-2023-10-24-11:07:36.tar.gz
-rw-r-----. 1 root root    116738 Oct 24 11:09 backup.log
drwxr-xr-x. 5 root root        55 Oct 13 17:00 bundle
-rw-r--r--. 1 root root      1751 Oct 23 12:28 certified_collection_seed_2023-10-23-12-13-33.log
drwxr-xr-x. 3 root root        33 Oct 13 16:56 collections
drwxr-xr-x. 2 root root        17 Oct 13 16:56 group_vars
-rw-r--r--. 1 root root      8962 Oct 23 11:23 inventory
-rw-r--r--. 1 root root       530 Oct 13 17:11 README.md
-rw-r-----. 1 root root   4796277 Oct 23 12:32 setup.log
-rwxr-xr-x. 1 root root     14780 Oct 13 16:56 setup.sh
-rw-r--r--. 1 root root       605 Oct 23 12:31 validated_collection_seed_2023-10-23-12-28-02.log
```

Dans ce cas, la taille de l'archive est de 594M :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# du -hs automation-platform-backup-*
594M    automation-platform-backup-2023-10-24-11:07:36.tar.gz
0       automation-platform-backup-latest.tar.gz
```

4.3 - Restaurer l'Ansible® Automation Hub à l'Identique

Copiez le fichier **automation-platform-backup-2023-10-24-11:07:36.tar.gz** vers le fichier **automation-platform-backup.tar.gz** :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# cp automation-platform-
backup-2023-10-24-11\:07\:36.tar.gz automation-platform-backup.tar.gz
```

Copiez le fichier **automation-platform-backup.tar.gz** vers votre VM **backuphub** selon votre numéro de stagiaire :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# scp automation-platform-backup.tar.gz
trainee@10.0.2.103:/home/trainee
The authenticity of host '10.0.2.103 (10.0.2.103)' can't be established.
ED25519 key fingerprint is SHA256:k/cooDrUynjprBohmFjJd22Ii2xlCXFdTHt/HAXpDE4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.103' (ED25519) to the list of known hosts.
trainee@10.0.2.103's password: trainee
automation-platform-backup.tar.gz
100%  593MB 315.5MB/s   00:01
```

Copiez ensuite le fichier **inventory** vers votre VM **backuphub** selon votre numéro de stagiaire :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# scp inventory
trainee@10.0.2.103:/home/trainee
trainee@10.0.2.103's password: trainee
inventory
100% 8962     8.9MB/s   00:00
```

Connectez-vous maintenant à votre VM **backuphub** selon votre numéro de stagiaire :

```
[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ssh -l trainee 10.0.2.103
trainee@10.0.2.103's password: trainee
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Tue Oct 24 13:02:01 2023 from 10.0.2.1
[trainee@backuphub ~]$ su -
Password: fenestros
[root@backuphub ~]#
```

Enregistrez la VM dans **votre** compte Red Hat :

```
[root@backuphub ~]# subscription-manager register
```

```
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: <login>
Password: <password>
The system has been registered with ID: a52fe25b-0bfd-4c60-8898-68de681c9fda
The registered system name is: backuphub.ittraining.loc
```

Configurez SELinux en mode permissive :

```
[root@backuphub ~]# setenforce permissive
```

Désarchivez le fichier **ansible-automation-platform-setup-bundle-2.4-2.2-x86_64.tar.gz** et consultez le contenu du répertoire créé :

```
[root@backuphub ~]# tar xvf ansible-automation-platform-setup-bundle-2.4-2.2-x86_64.tar.gz

[root@backuphub ~]# cd ansible-automation-platform-setup-bundle-2.4-2.2-x86_64/

[root@autohub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ls
bundle  collections  group_vars  inventory  README.md  setup.sh
```

Installez le paquet **ansible-core** :

```
[root@backuphub ~]# dnf install ansible-core
```

Copiez le fichier **/home/trainee/automation-platform-backup.tar.gz** dans le répertoire courant :

```
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# cp /home/trainee/automation-platform-
backup.tar.gz .
```

Ajoutez un lien symbolique vers le fichier **automation-platform-backup.tar.gz** :

```
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ln -s automation-platform-backup.tar.gz
automation-platform-backup-latest.tar.gz
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ls -l
```

```
total 607316
lrwxrwxrwx. 1 root root        33 Oct 24 13:26 automation-platform-backup-latest.tar.gz -> automation-platform-
backup.tar.gz
-rw-------. 1 root root 621857468 Oct 24 13:22 automation-platform-backup.tar.gz
drwxr-xr-x. 5 root root        55 Oct 13 17:00 bundle
drwxr-xr-x. 3 root root        33 Oct 13 16:56 collections
drwxr-xr-x. 2 root root        17 Oct 13 16:56 group_vars
-rw-r--r--. 1 root root      8896 Oct 13 16:56 inventory
-rw-r--r--. 1 root root       530 Oct 13 17:11 README.md
-rwxr-xr-x. 1 root root     14780 Oct 13 16:56 setup.sh
```

Important : Le processus de restauration recherche le fichier identifié par le lien symbolique **automation-platform-backup-latest.tar.gz**.

Copiez le fichier **/home/trainee/inventory** dans le répertoire courant :

```
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# cp /home/trainee/inventory .
cp: overwrite './inventory'? y
```

Modifiez le nom d'hôte de la VM :

```
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# hostnamectl set-hostname
autohub.ittraining.loc
```

Modifiez le fichier **/etc/hosts** :

```
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# vi /etc/hosts
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

```
10.0.2.103      autohub.ittraining.loc
```

Installez maintenant un nouvel Ansible® Automation Hub :

```
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ./setup.sh
...
PLAY RECAP *********************************************************************
autohub.ittraining.loc     : ok=165  changed=51   unreachable=0    failed=2    skipped=173  rescued=0
ignored=0
localhost                  : ok=0    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
ignored=0

[error] Oops!  An error occurred while running setup.
[warn] /var/log/tower does not exist. Setup log saved to setup.log.
[warn] Provided path does not exist or is not accessible. Setup log saved to ./setup.log.
```

En cas d'erreur(s), ré-exécutez le script setup.sh :

```
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ./setup.sh
...
PLAY RECAP *********************************************************************
autohub.ittraining.loc     : ok=900  changed=150  unreachable=0    failed=0    skipped=379  rescued=0
ignored=0
localhost                  : ok=0    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
ignored=0

The setup process completed successfully.
[warn] /var/log/tower does not exist. Setup log saved to setup.log.
[warn] Provided path does not exist or is not accessible. Setup log saved to ./setup.log.
```

Dernièrement, procédez à la restauration :

```
[root@backuphub ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# ./setup.sh -r
...
```

```
PLAY RECAP *********************************************************************
autohub.ittraining.loc     : ok=78   changed=23   unreachable=0    failed=0    skipped=101  rescued=0
ignored=0

The setup process completed successfully.
[warn] /var/log/tower does not exist. Setup log saved to restore.log.
[warn] Provided path does not exist or is not accessible. Setup log saved to ./restore.log.
```

LAB #5 - Utilisation de l'Ansible® Automation Controller

5.1 - Créer un Job Simple

Revenez dans votre Ansible® Automation Controller.

Commencez par créer un playbook simple, destiné à imprimer la phrase **Hello World!** :

```
[root@redhat9 ansible-automation-platform-setup-bundle-2.4-2.2-x86_64]# cd ~

[root@redhat9 ~]# vi hello-world.yml

[root@redhat9 ~]# cat hello-world.yml
---
- hosts: all
  tasks:
    - name: Hello World
      debug:
        msg: Hello World!
```

Vérifiez la syntaxe de ce playbook :

```
[root@redhat9 ~]# ansible-playbook --syntax-check hello-world.yml
```

```
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

playbook: hello-world.yml
```

Notez l'avertissement **[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'**. En effet, Ansible ne trouve pas de fichier inventory. Créez donc le fichier inventory :

```
[root@redhat9 ~]# vi inventory
[root@redhat9 ~]# cat inventory
[Test]
localhost ansible_connection=local
```

Vérifiez de nouveau la syntaxe de ce playbook :

```
[root@redhat9 ~]# ansible-playbook --syntax-check hello-world.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

playbook: hello-world.yml
```

Notez l'avertissement **[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'**. En effet, Ansible cherche les fichiers inventory **/etc/ansible/hosts**. Bien que ce fichier existe, il est vide de toute déclaration active :

```
[root@redhat9 ~]# ls -l /etc/ansible/hosts
-rw-r--r--. 1 root root 1018 Jun  5 17:30 /etc/ansible/hosts
[root@redhat9 ~]# cat /etc/ansible/hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
```

```
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group:

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:
```

```
## db-[99:101]-node.example.com
```

Testez le playbook en indiquant le fichier inventory créé au dessus :

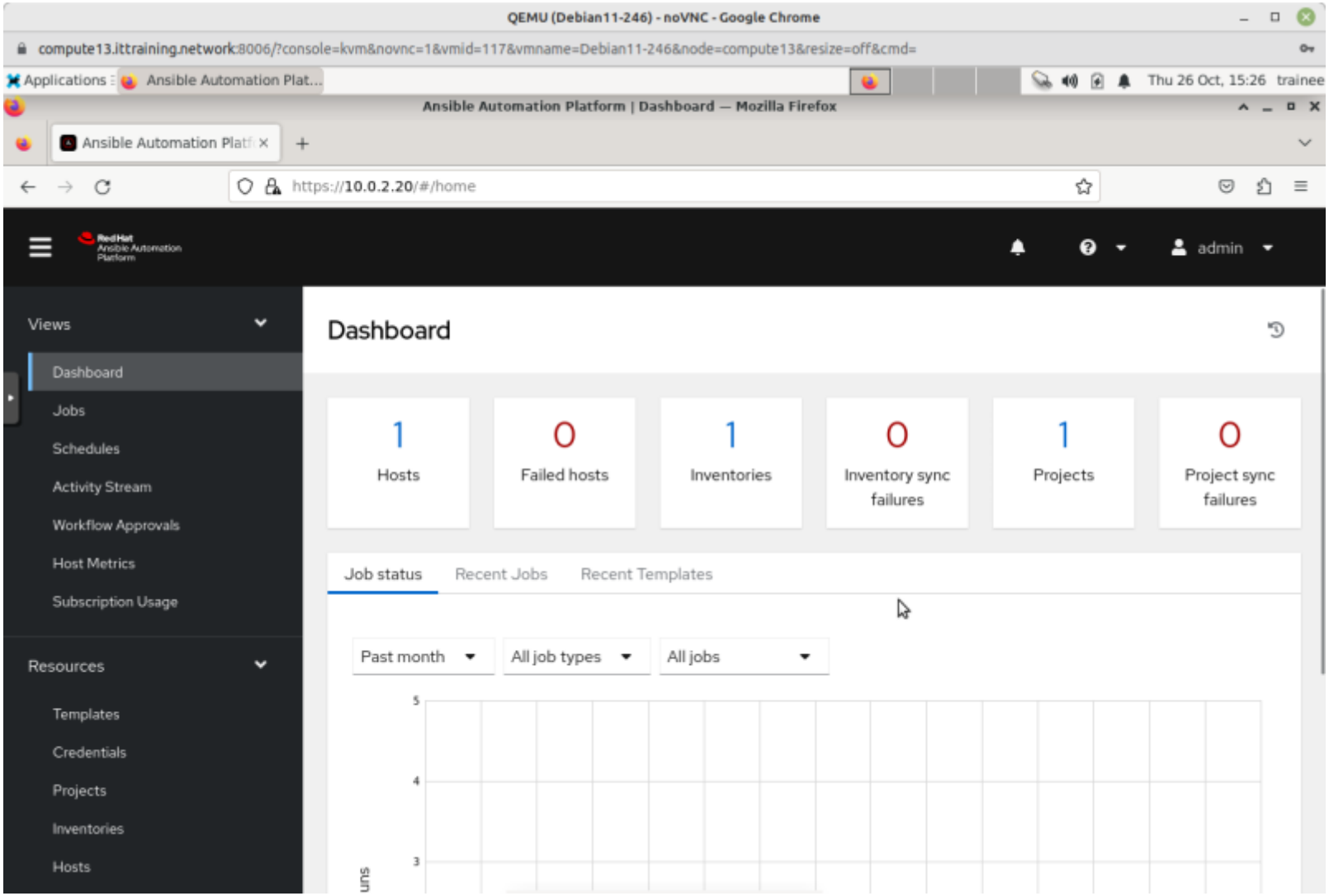
```
[root@redhat9 ~]# ansible-playbook -i inventory hello-world.yml

PLAY [all] *********************************************************************

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

TASK [Hello World] *************************************************************
ok: [localhost] => {
    "msg": "Hello World!"
}

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

5.2 - Créer un Projet

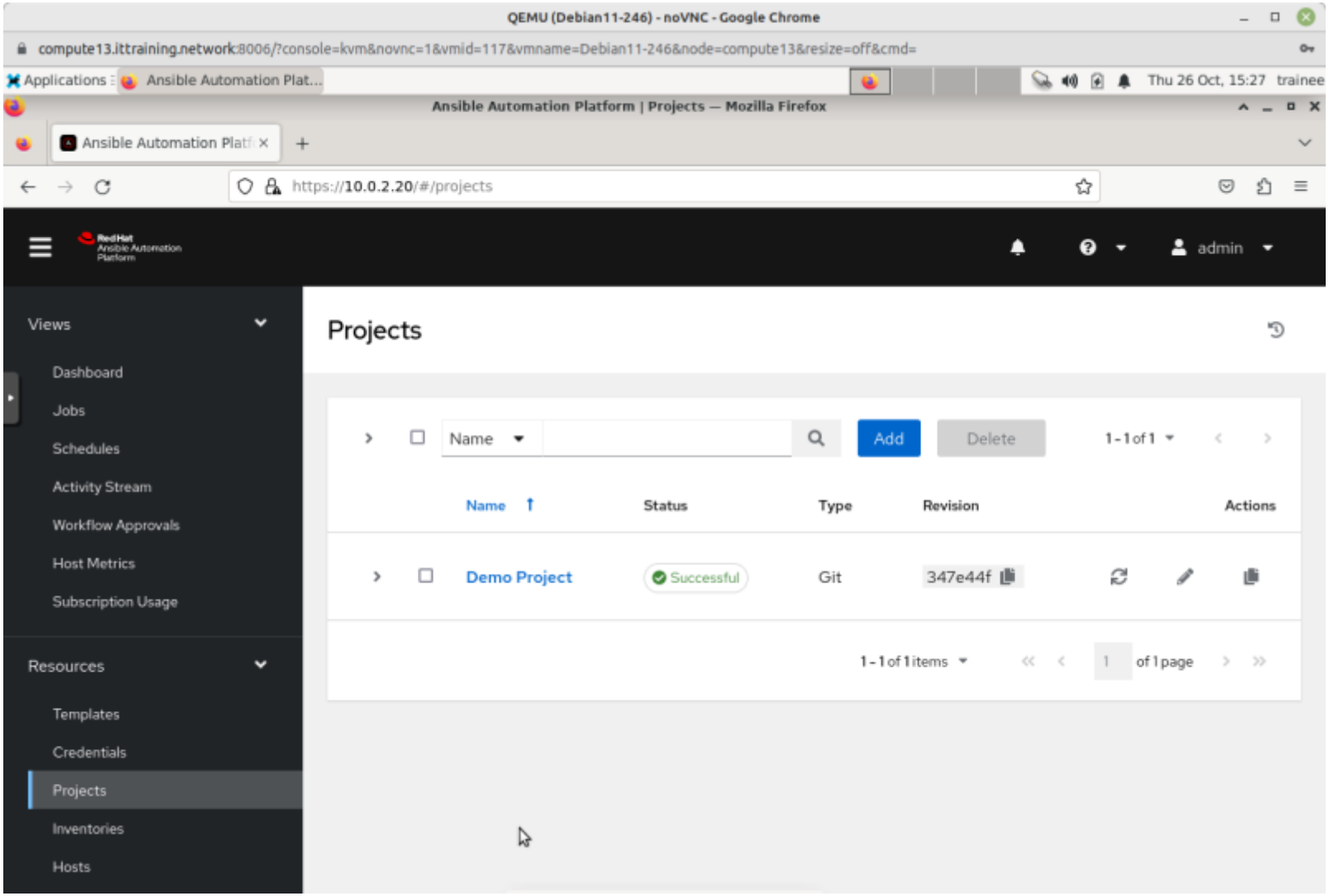
Pour exécuter ce même playbook avec Red Hat® Ansible® Automation Platform, il faut créer un projet. Les projets sont des répertoire dans **/var/lib/awx/projects** et doivent contenir le(s) playbook(s) :

```
[root@redhat9 ~]# mkdir /var/lib/awx/projects/myrepo
[root@redhat9 ~]# cp hello-world.yml /var/lib/awx/projects/myrepo
```

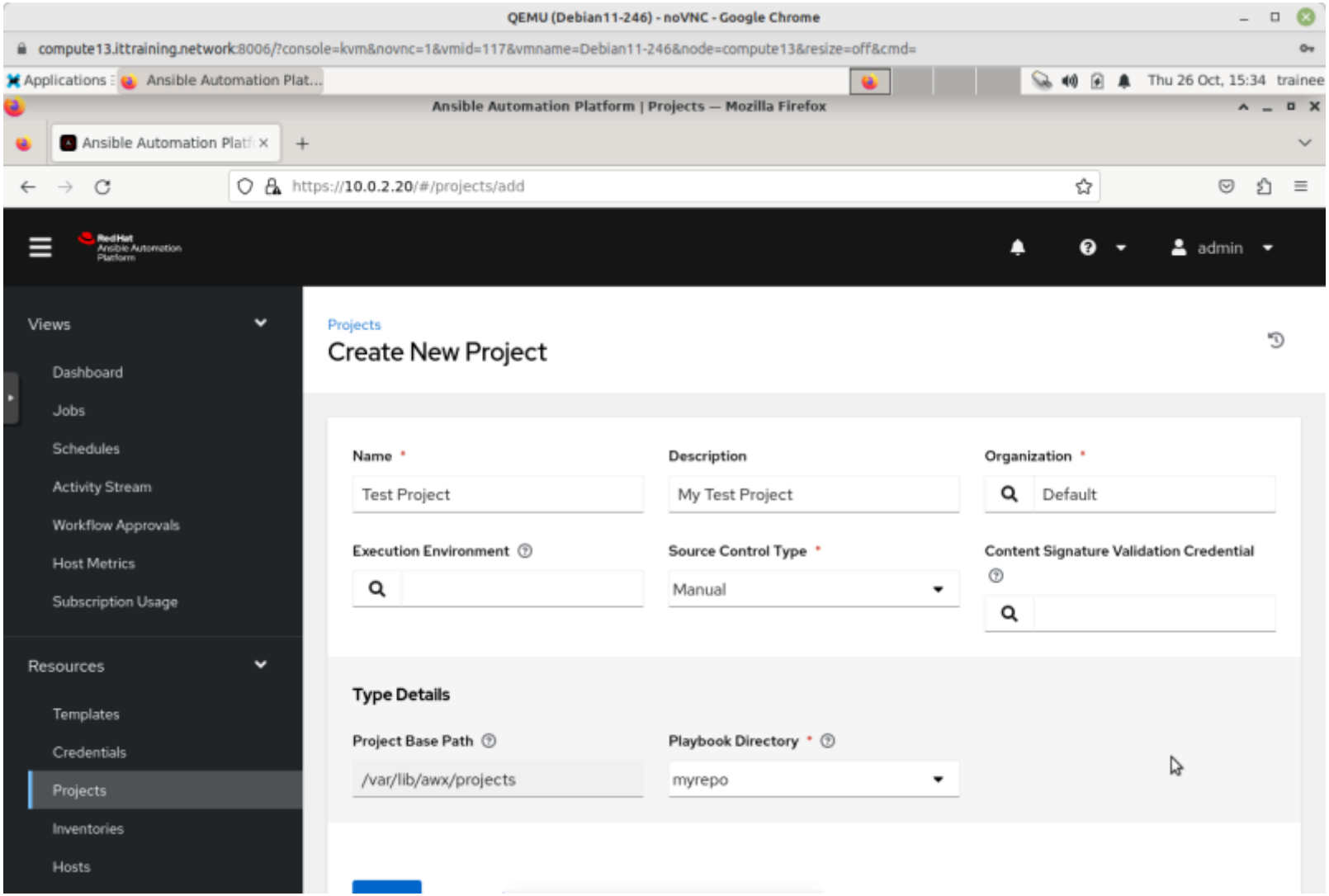
Connectez-vous à votre **VM Debian_10.0.2.46_VNC**. Ouvrez un navigateur Web et naviguez à **https://10.0.2.101** :



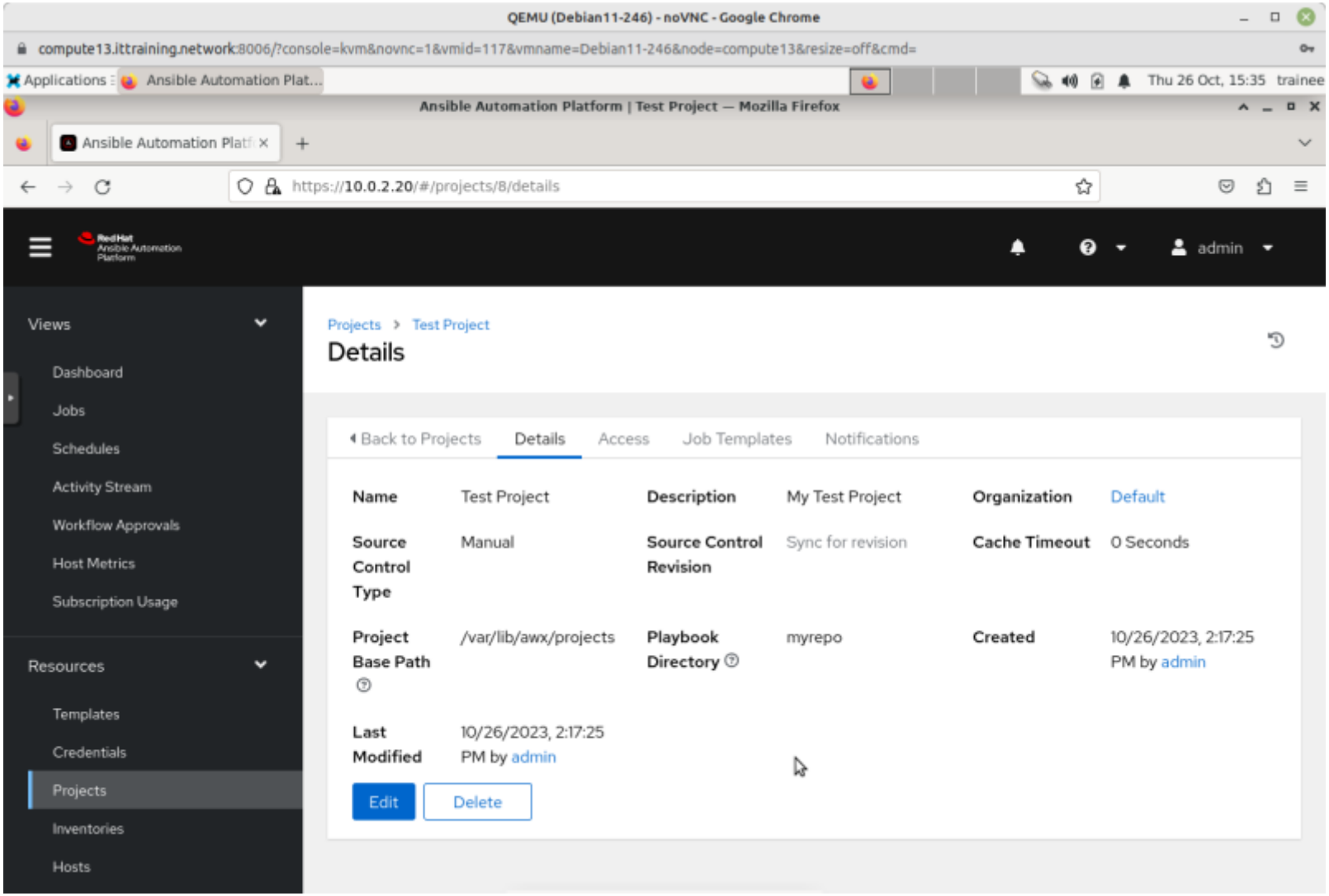
Cliquez sur **Projects** puis sur le bouton **Add** :



Indiquez **Test Project** dans **Name**, **My test Project** dans **Description**, choisissez **myrepo** dans **Playbook Directory** et cliquez sur le bouton **Save** :

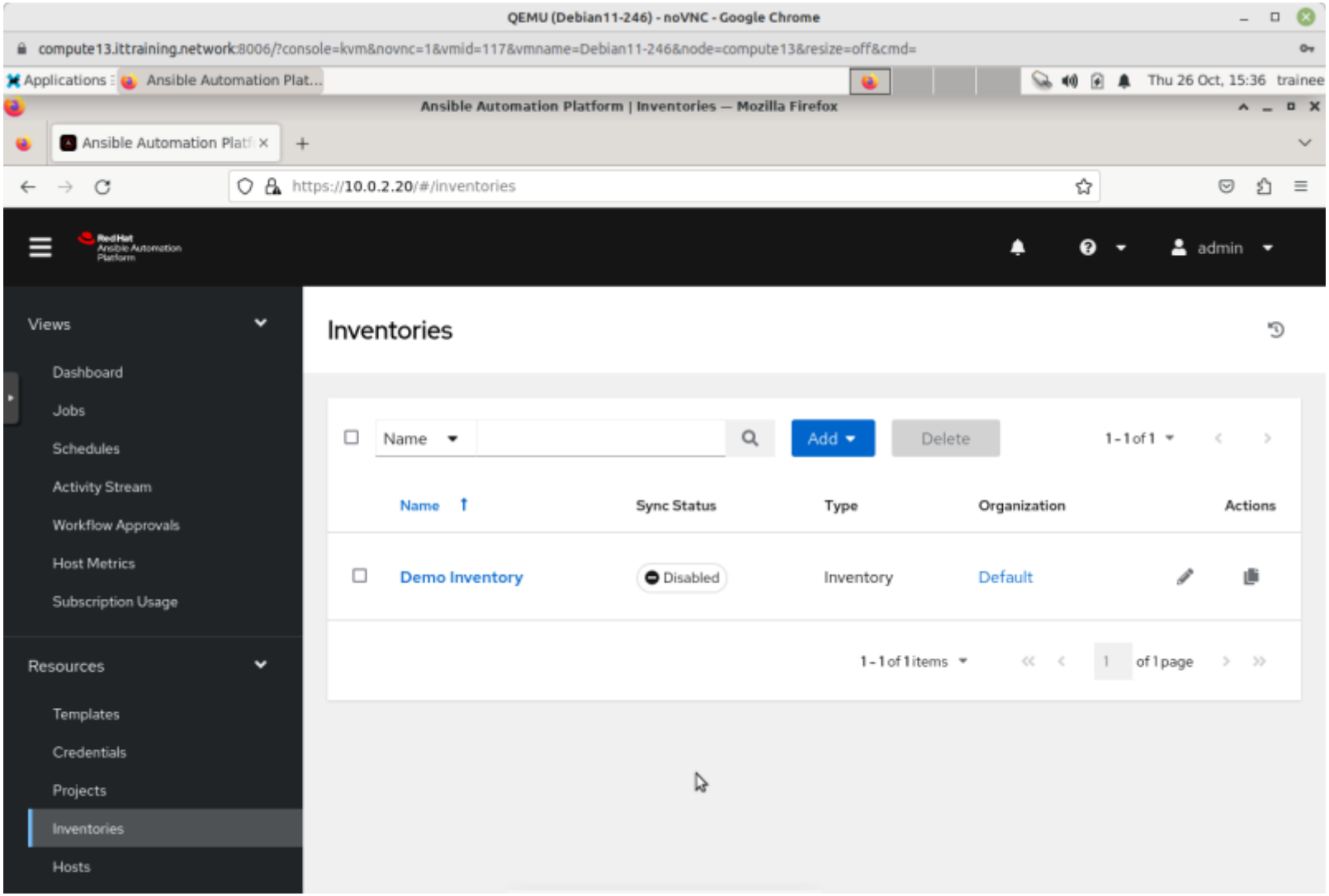


Vérifiez les informations de votre projet :

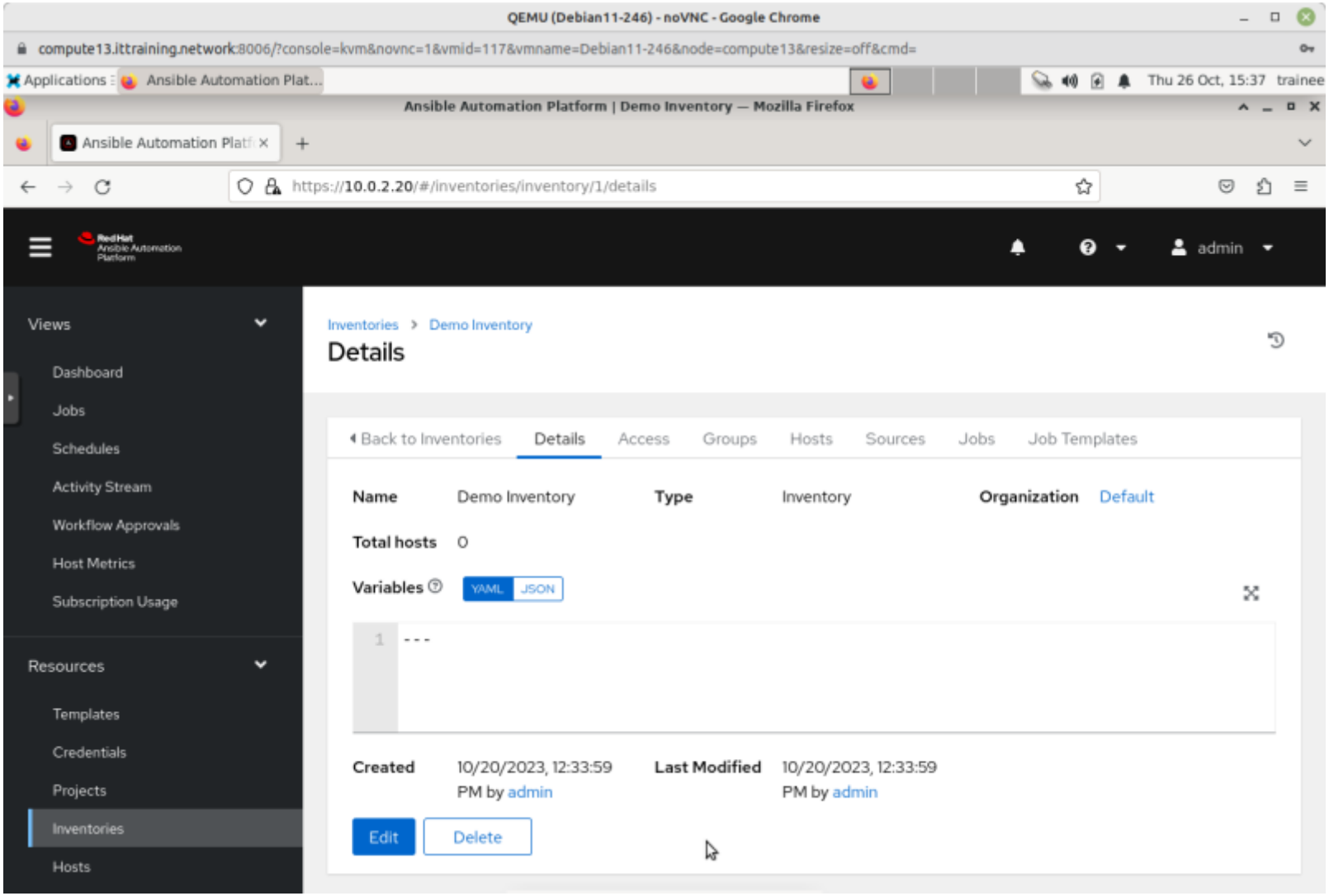


5.3 - Créer un Inventory

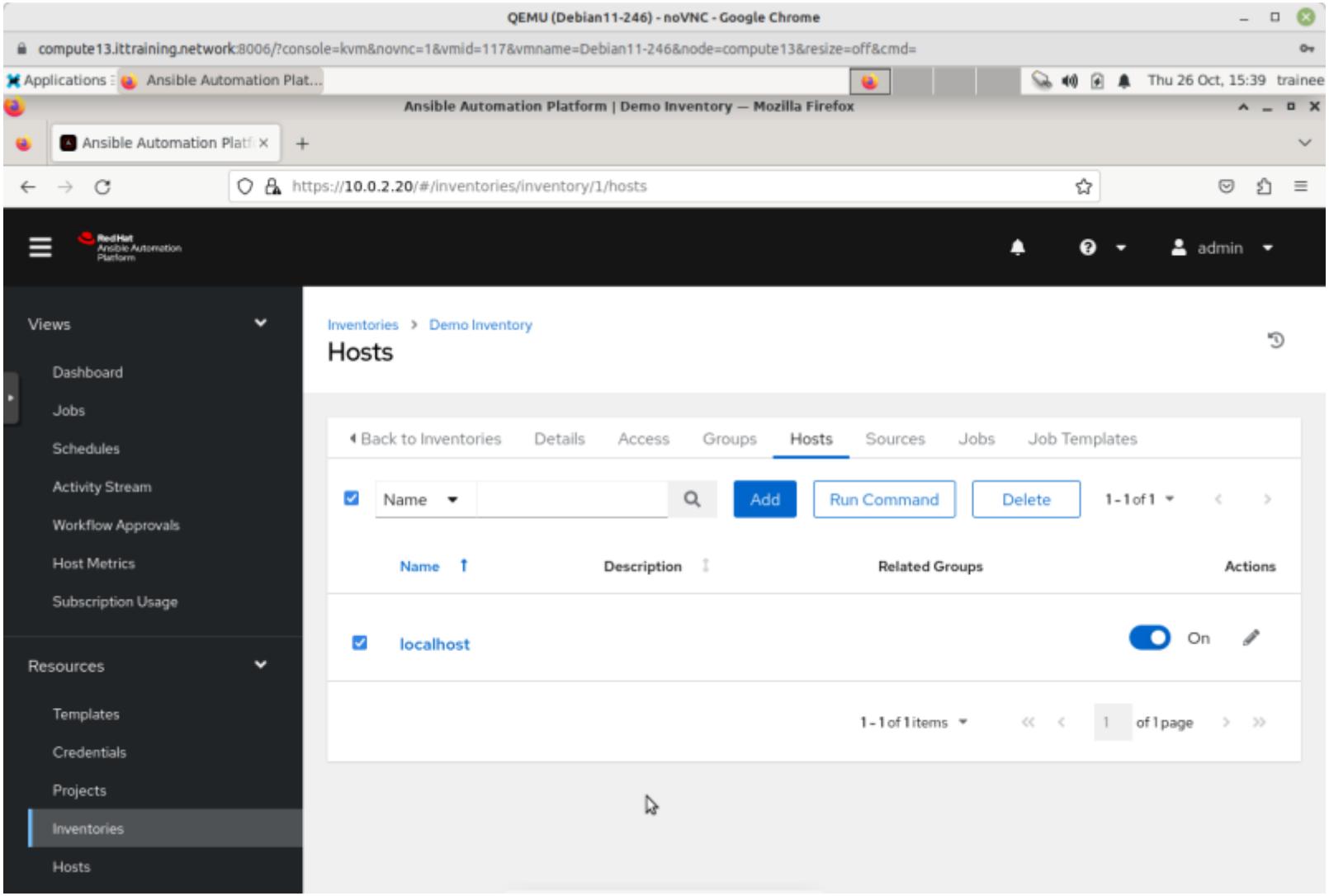
Cliquez ensuite sur **Inventories** :



Cliquez sur le lien **Demo Inventory** :

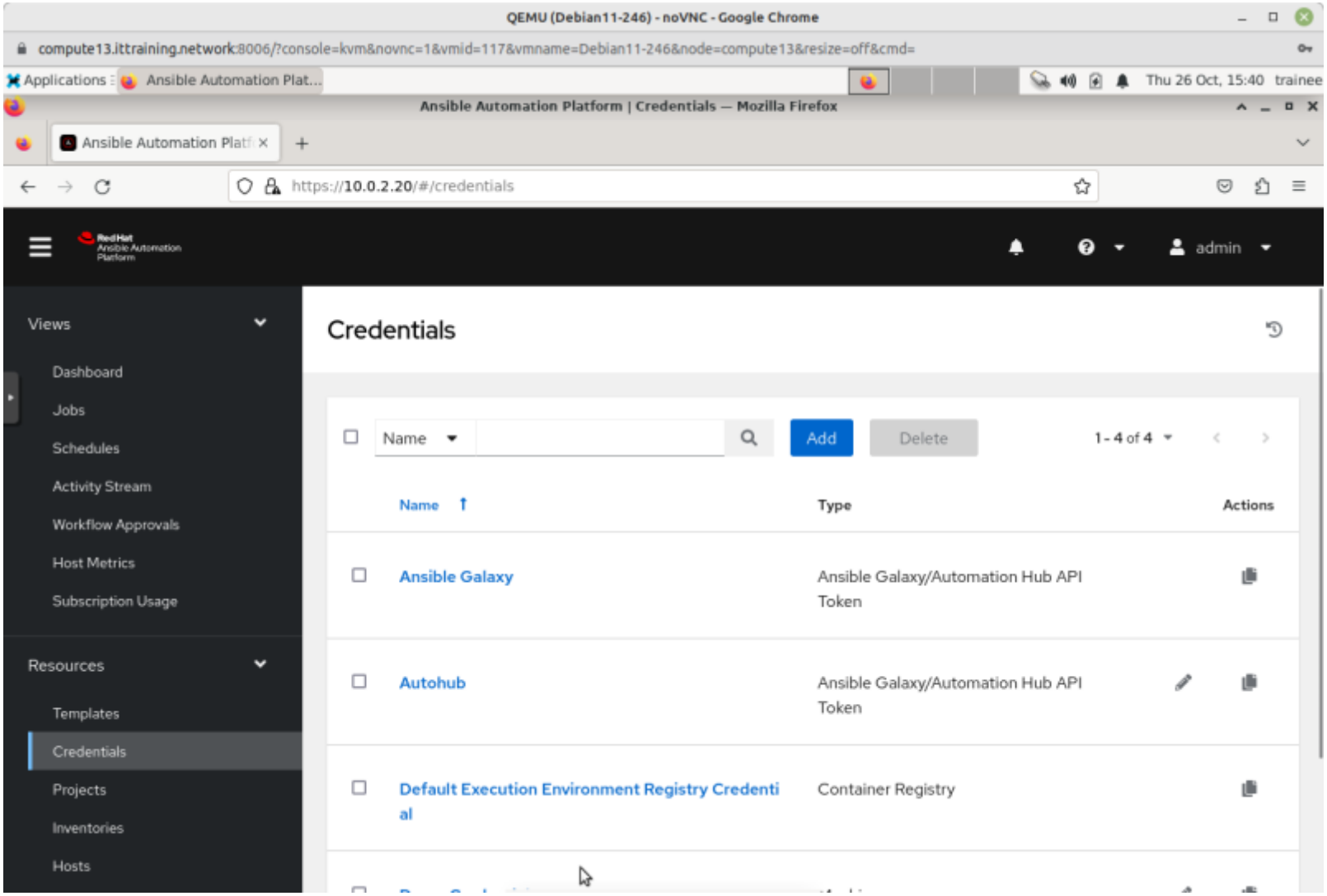


Cliquez sur le lien **Hosts** et cochez **localhost** :

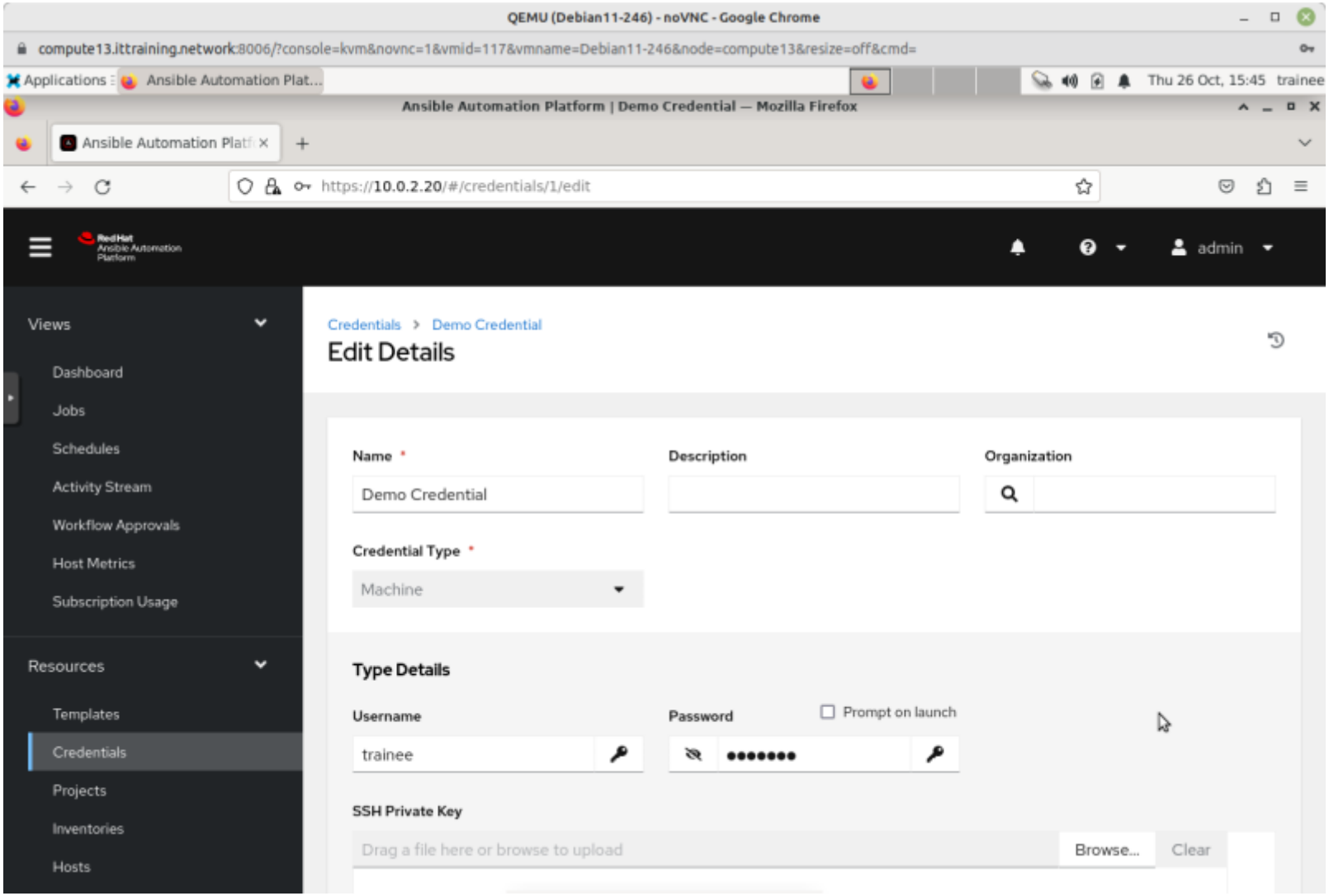


5.4 - Créer des Informations d'Identification

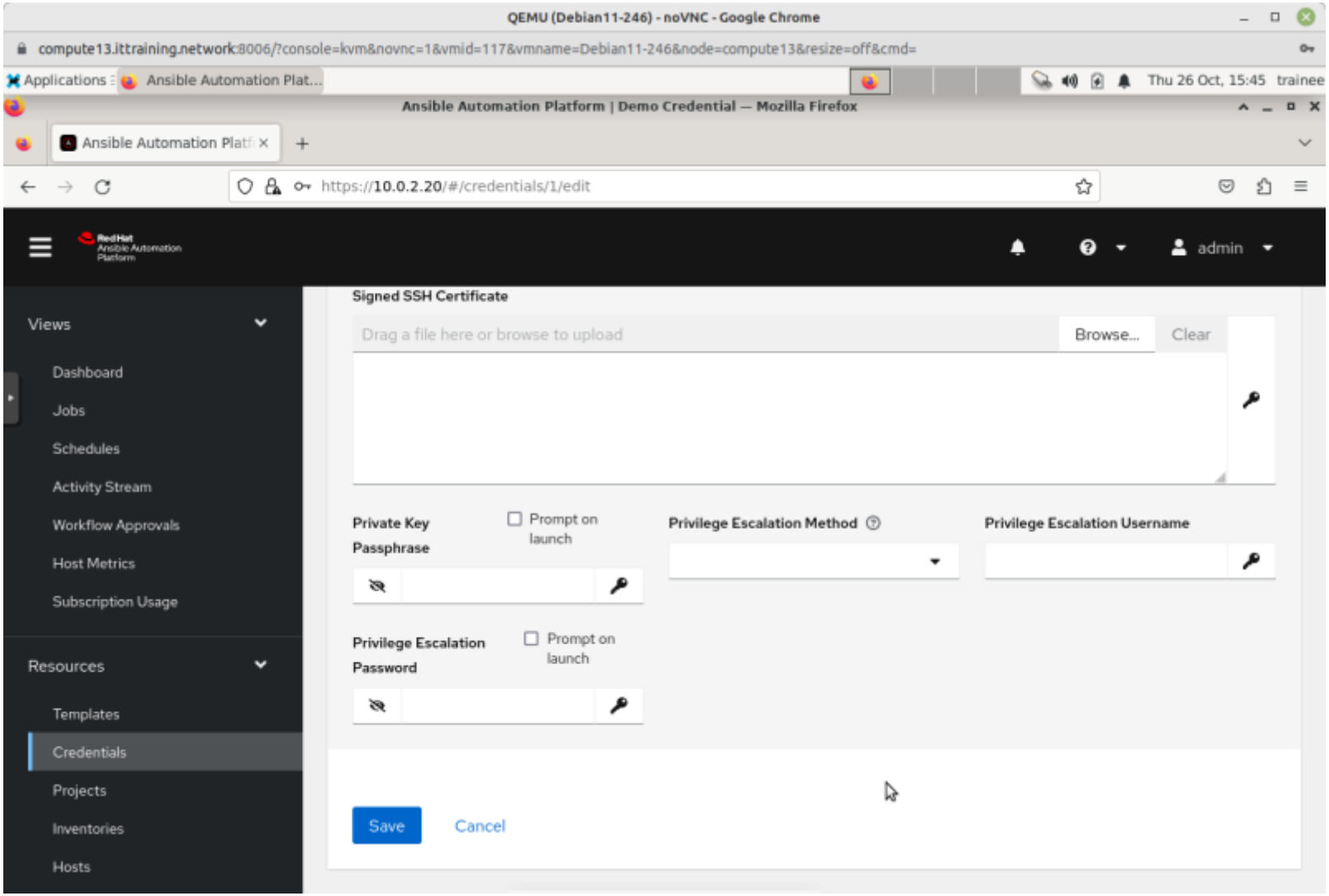
Cliquez sur **Credentials** :



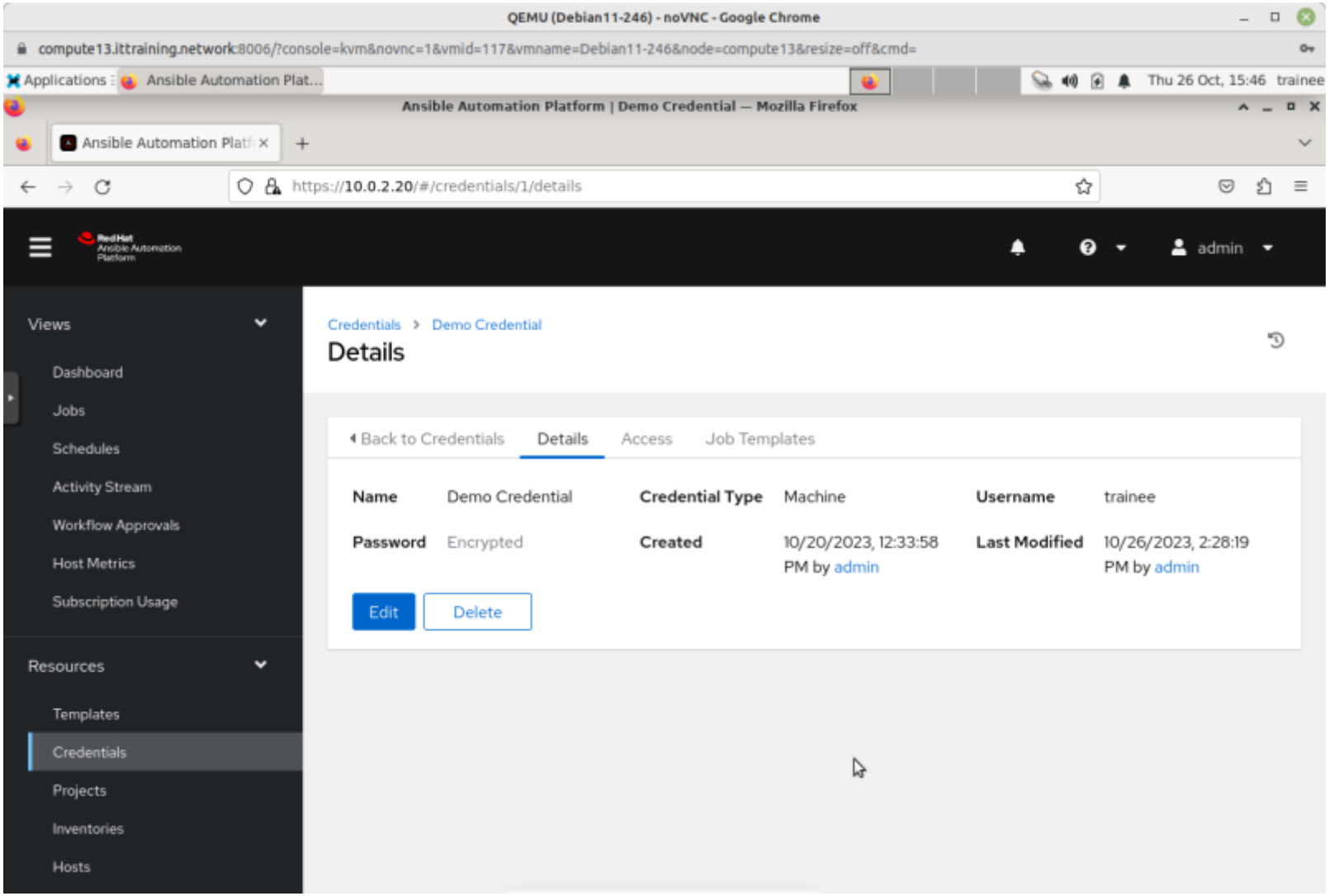
Éditez **Demo Credential** en indiquant **trainee** en tant que **Username** et **trainee** en tant que **Password** :



Cliquez sur le bouton **Save** :

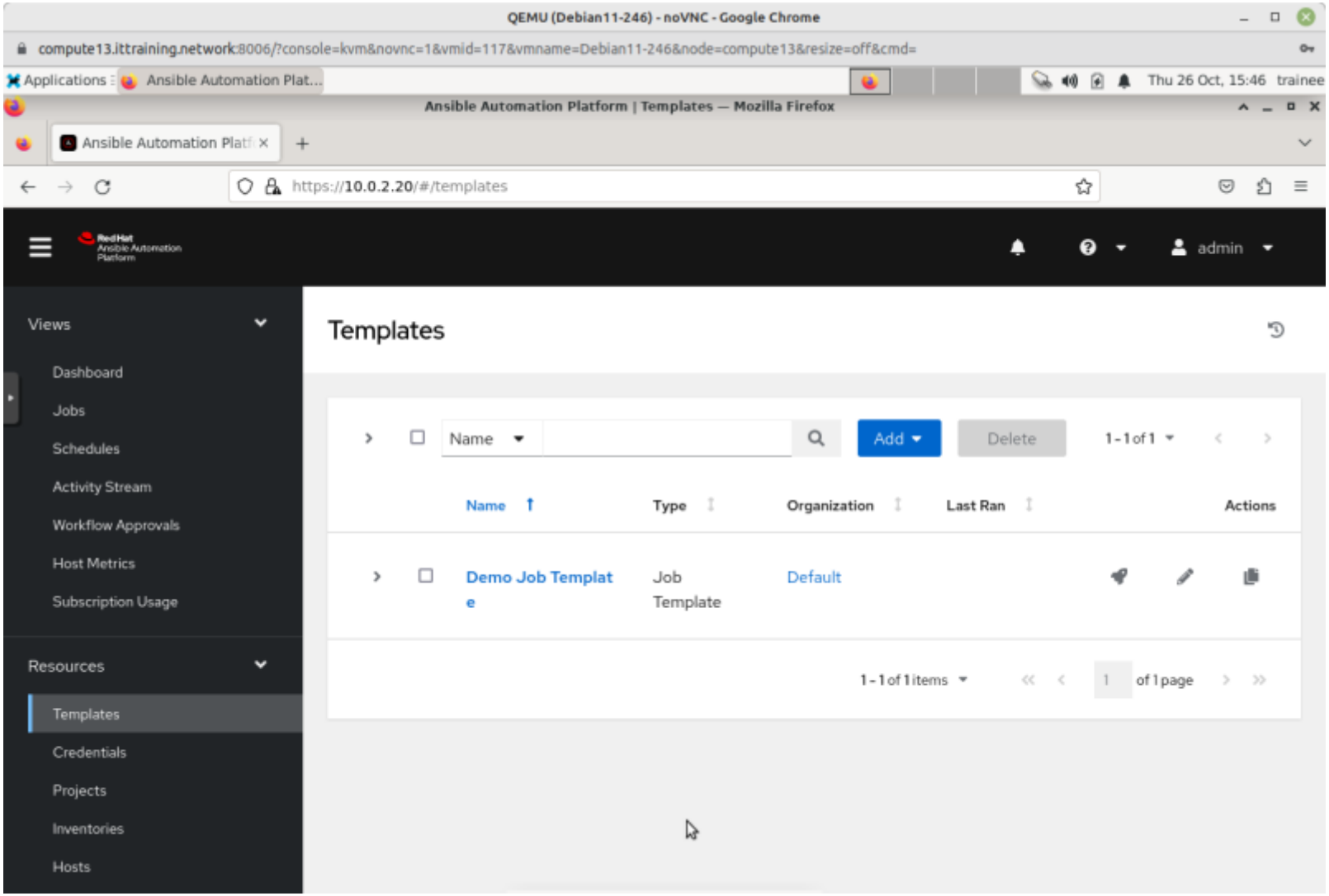


Vous obtiendrez :

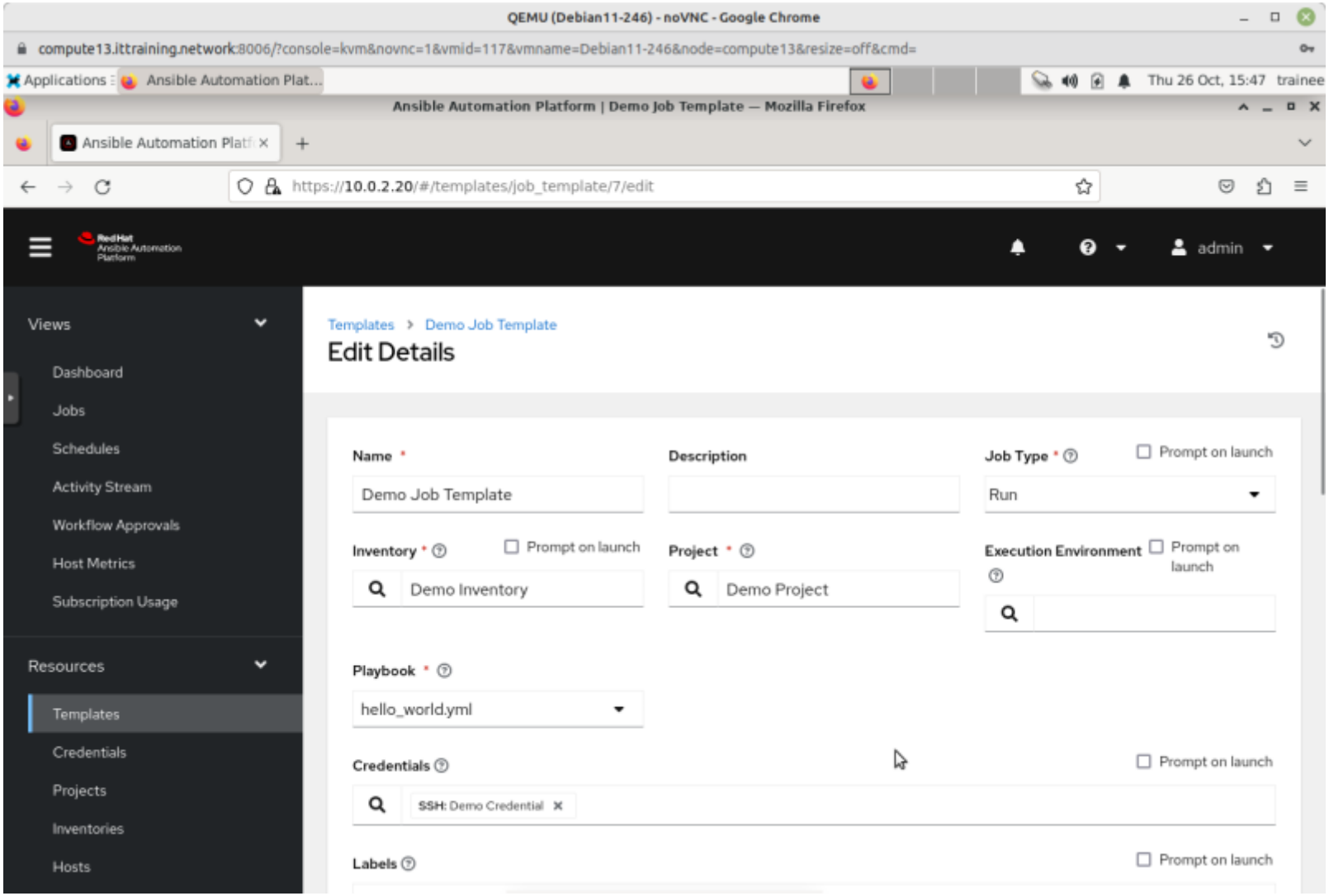


5.5 - Créer un Gabarit

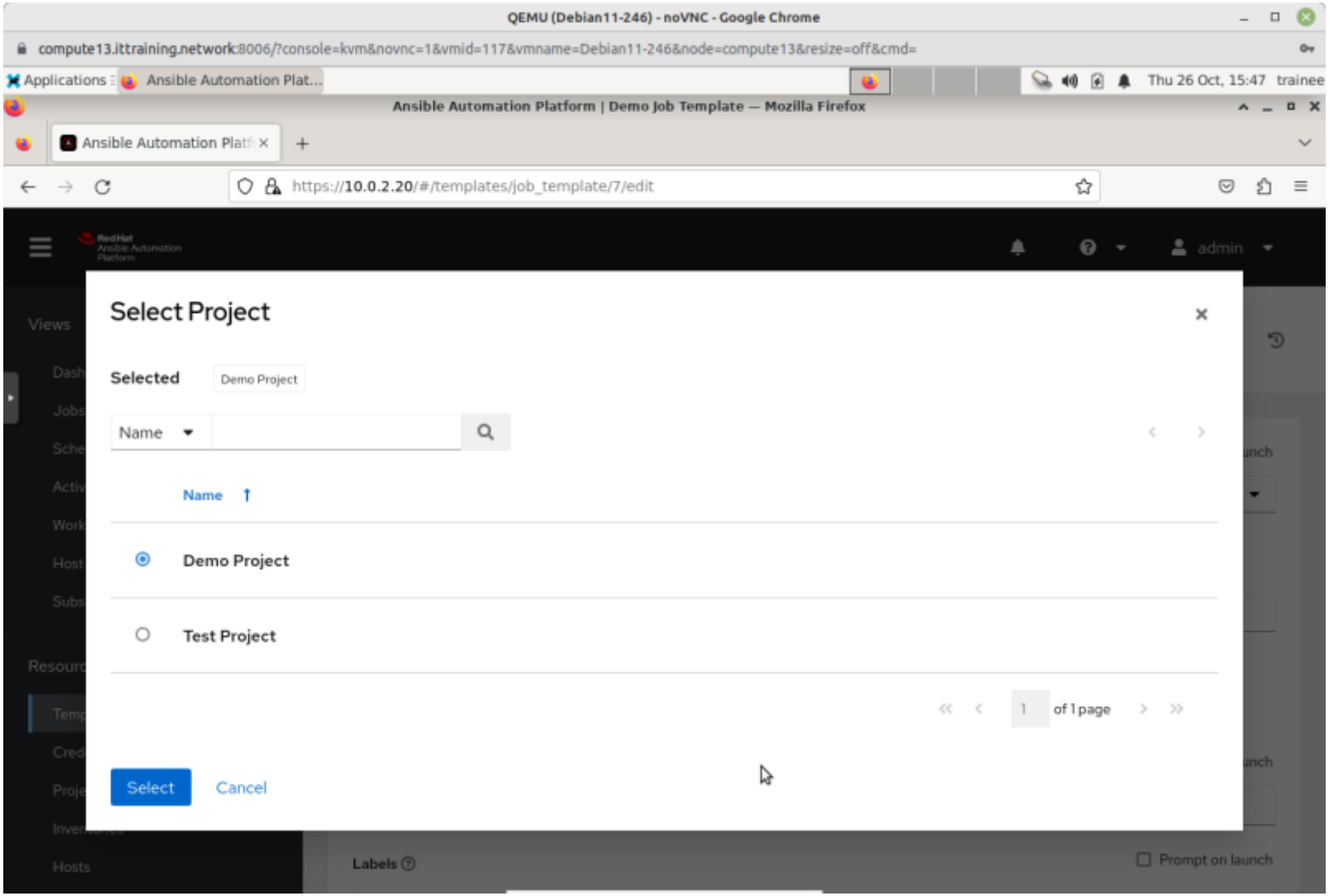
Cliquez sur **Templates** puis cliquez sur **Demo Job Template** :



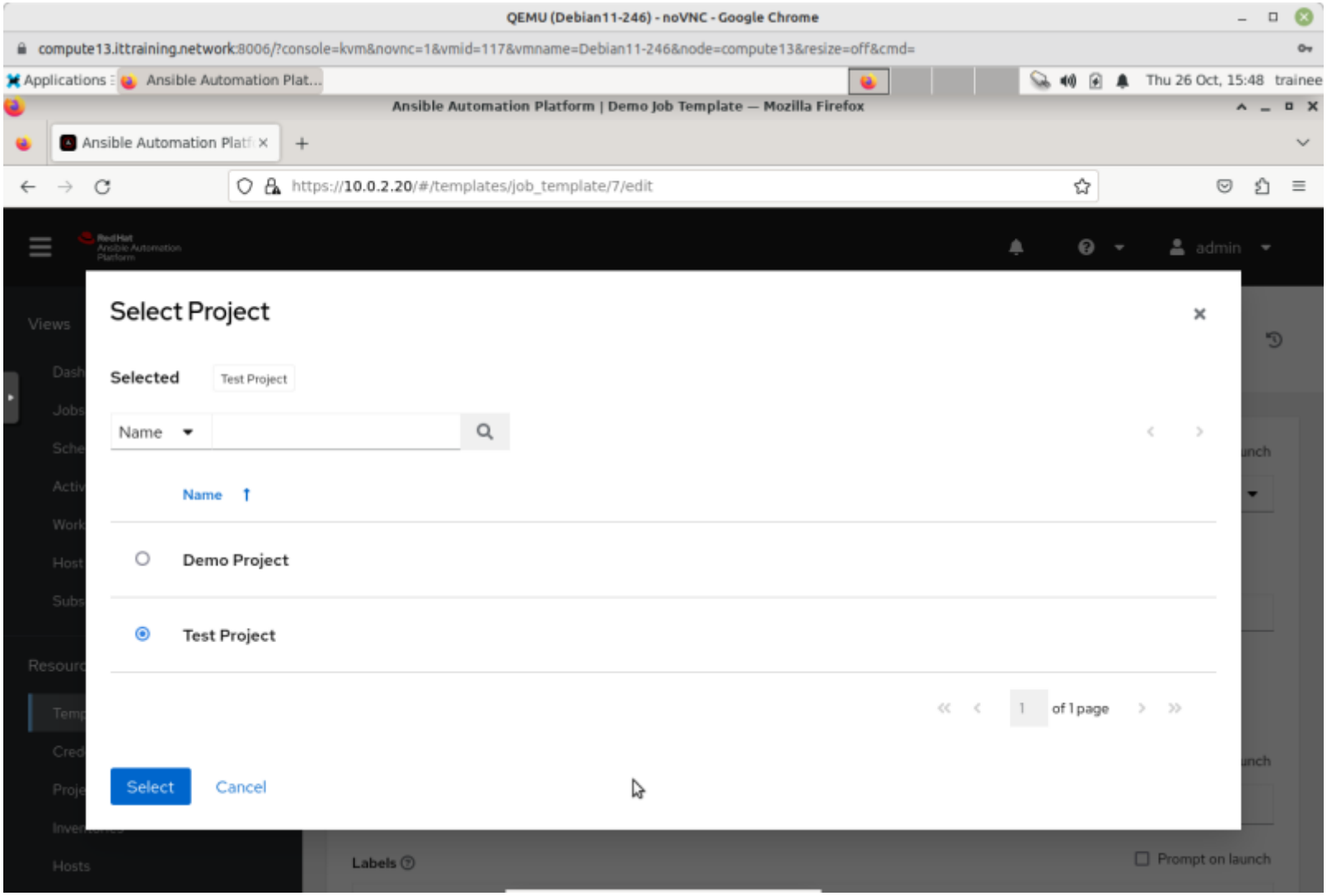
Cliquez sur la loupe dans le champs **Project** :



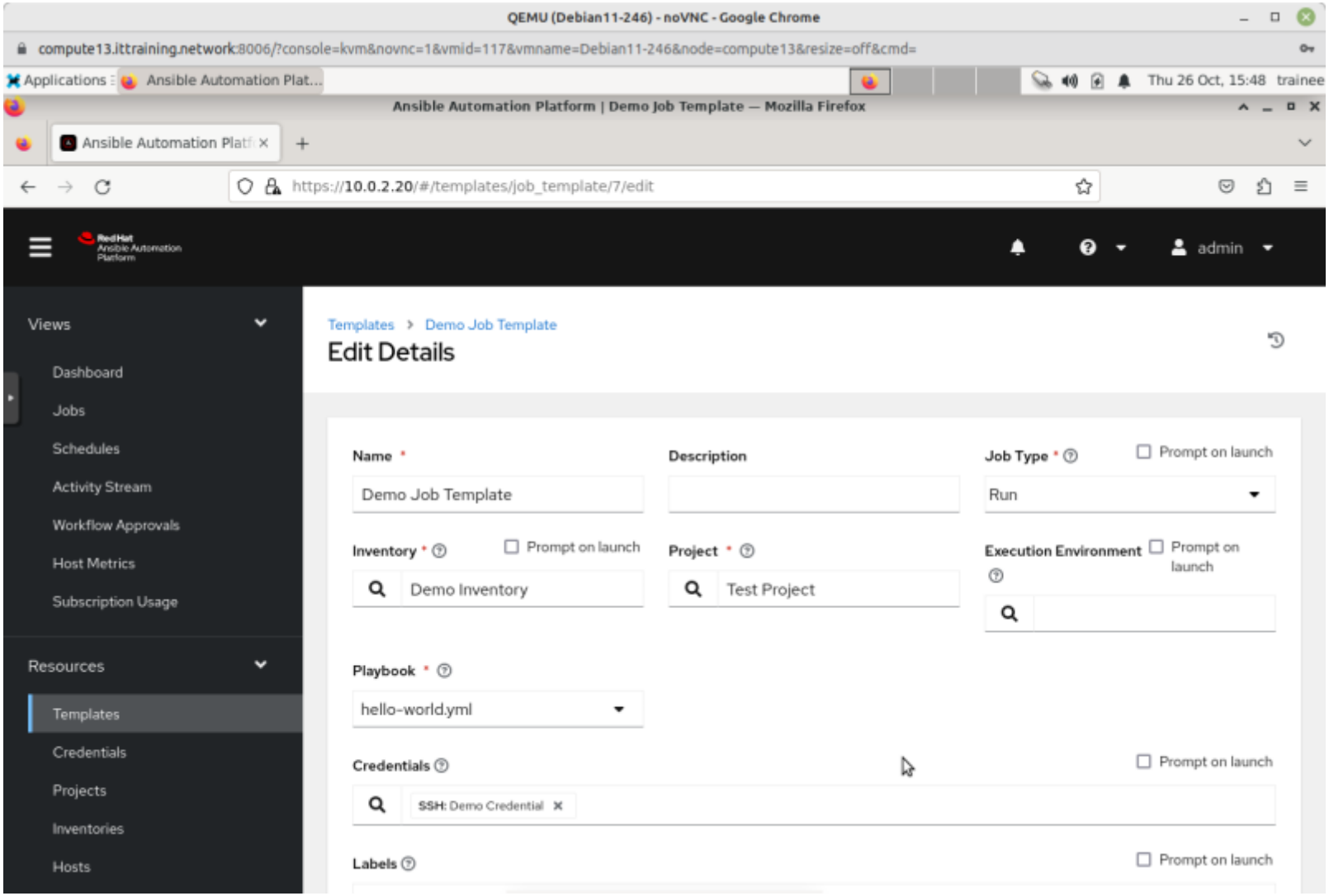
Vous obtiendrez :



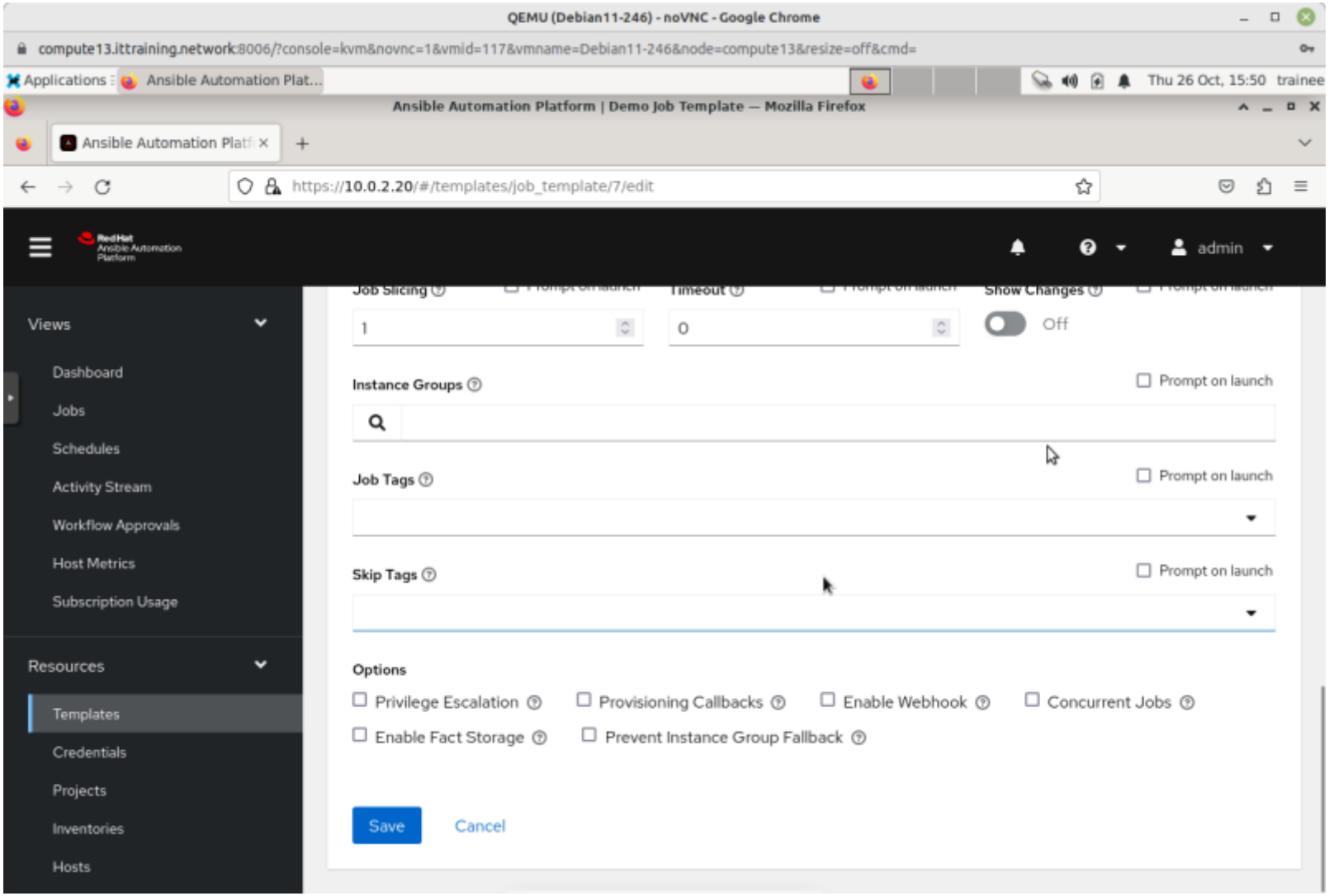
Cochez **Test Project** puis cliquez sur le bouton **Select** :



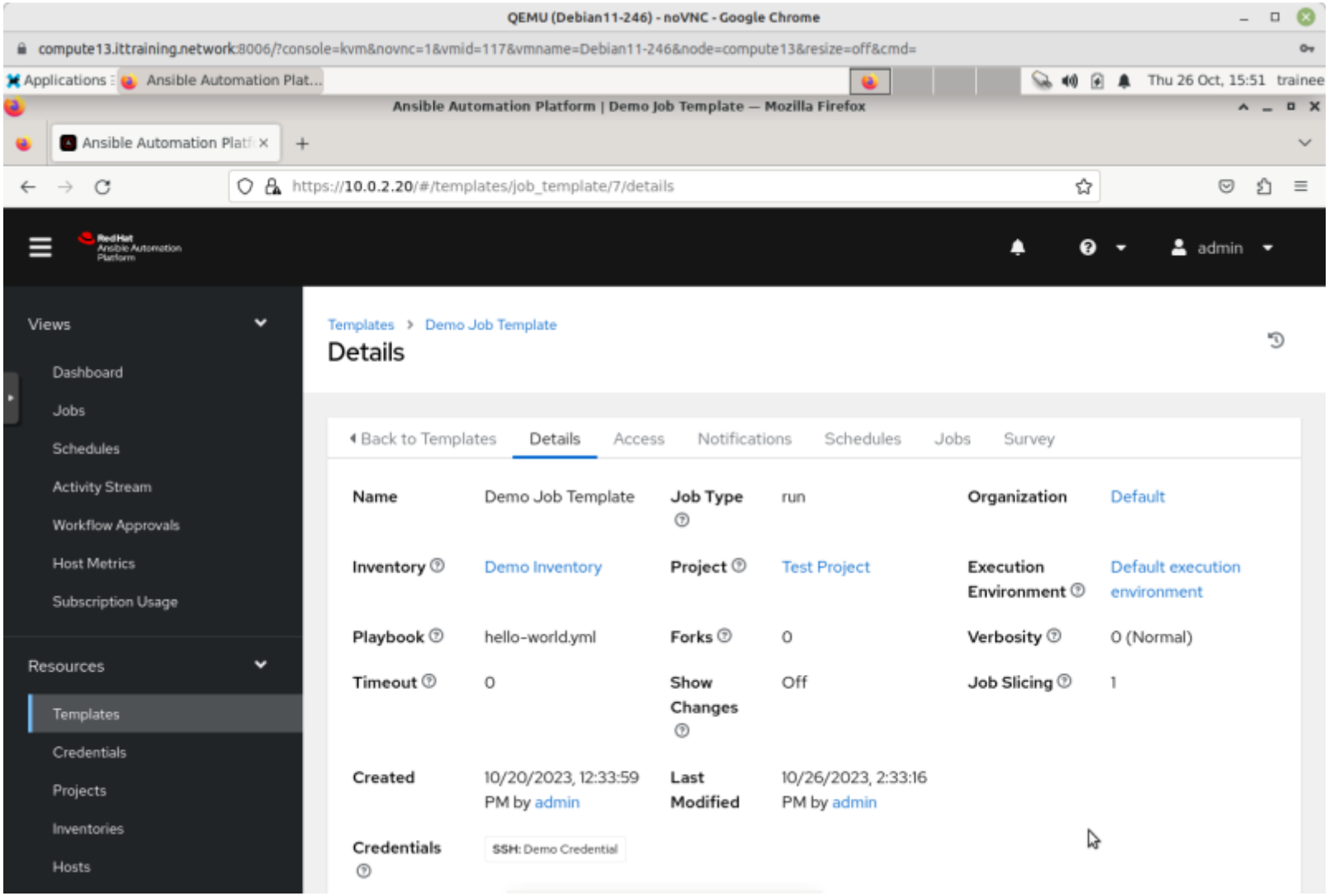
Vous verrez le Playbook **hello-world.yml** apparaître dans le champs **Playbook** :



Cliquez sur le bouton **Save** :

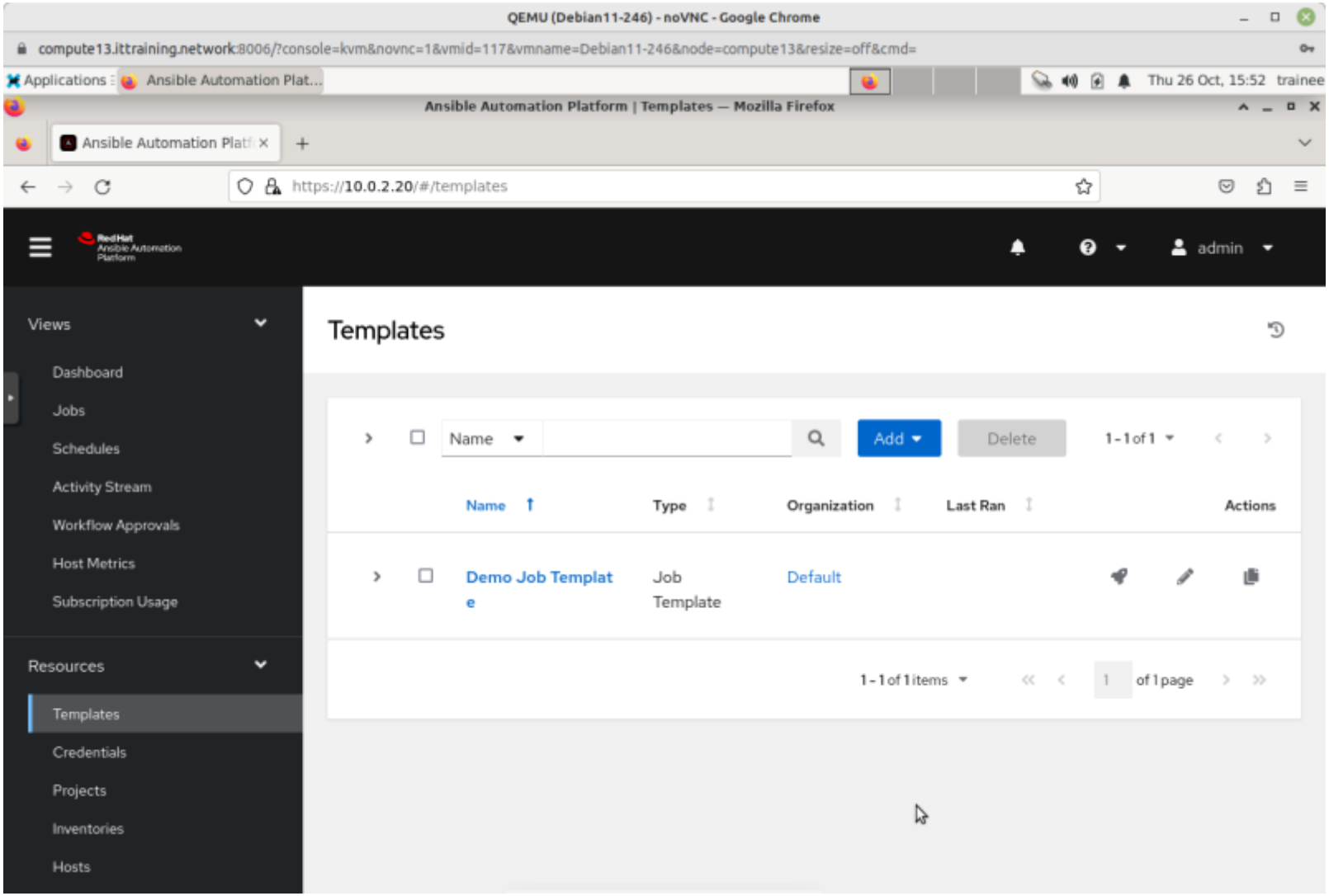


Vérifiez votre Template puis cliquez sur le lien **Templates** :

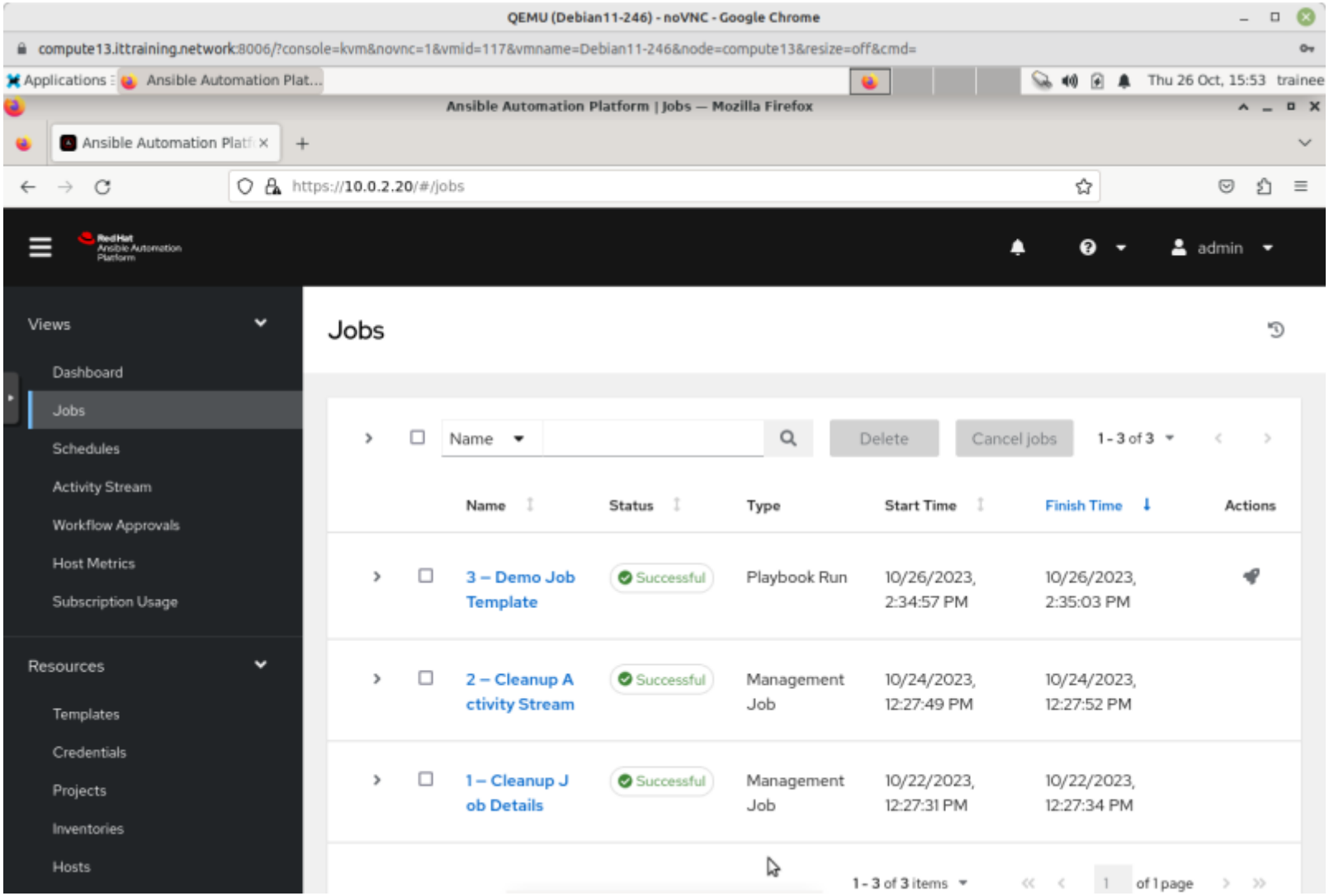


5.6 - Exécuter un Job

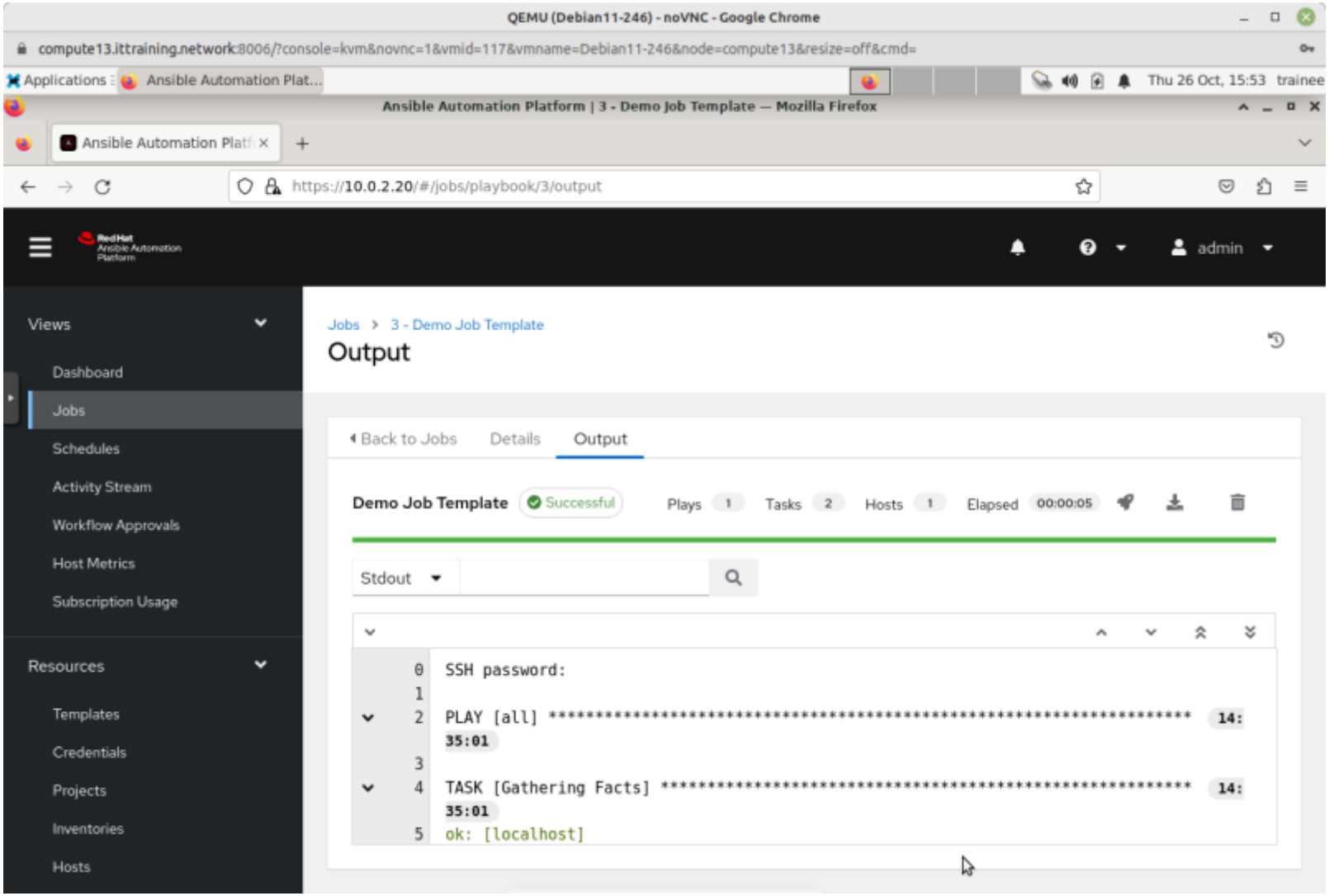
Cliquez sur l'icône de la fusée en face de **Demo Job Template** :



Cliquez sur **Jobs** :



Cliquez sur **Demo Job Template** et constatez la réussite de l'exécution du Job :



LAB #6 - Utilisation de l'Ansible® Automation Hub

6.1 - Créer une Collection

Connectez-vous ensuite à votre VM **autohub.ittraining.loc** :

```
[root@redhat9 ~]# ssh -l trainee 10.0.2.102
```

Créez un répertoire appelé **collection** et placez-vous dedans :

```
[trainee@autohub ~]$ mkdir collection

[trainee@autohub ~]$ cd collection
```

Créez une **Collection** vide appelée **ittraining** dans un **Namespace** appelé **ittraining** avec la commande **ansible-galaxy** :

```
[trainee@autohub collection]$ ansible-galaxy collection init ittraining.ittraining
- Collection ittraining.ittraining was created successfully
```

Consultez l'arborescence de la Collection :

```
[trainee@autohub collection]$ tree
.
└── ittraining
    └── ittraining
        ├── docs
        ├── galaxy.yml
        ├── meta
        │   └── runtime.yml
        ├── plugins
        │   └── README.md
        ├── README.md
        └── roles
```

```
6 directories, 4 files
```

Les informations concernant la Collection sont stockées dans le fichier **~/collections/ittraining/ittraining/galaxy.yml** :

```
[trainee@autohub collection]$ cat ittraining/ittraining/galaxy.yml
### REQUIRED
# The namespace of the collection. This can be a company/brand/organization or product namespace under which all
# content lives. May only contain alphanumeric lowercase characters and underscores. Namespaces cannot start with
# underscores or numbers and cannot contain consecutive underscores
namespace: ittraining

# The name of the collection. Has the same character restrictions as 'namespace'
name: ittraining

# The version of the collection. Must be compatible with semantic versioning
version: 1.0.0

# The path to the Markdown (.md) readme file. This path is relative to the root of the collection
readme: README.md

# A list of the collection's content authors. Can be just the name or in the format 'Full Name <email> (url)
# @nicks:irc/im.site#channel'
authors:
- your name <example@domain.com>


### OPTIONAL but strongly recommended
# A short summary description of the collection
description: your collection description

# Either a single license or a list of licenses for content inside of a collection. Ansible Galaxy currently only
# accepts L(SPDX,https://spdx.org/licenses/) licenses. This key is mutually exclusive with 'license_file'
license:
- GPL-2.0-or-later
```

```
# The path to the license file for the collection. This path is relative to the root of the collection. This key
is
# mutually exclusive with 'license'
license_file: ''

# A list of tags you want to associate with the collection for indexing/searching. A tag name has the same
character
# requirements as 'namespace' and 'name'
tags: []

# Collections that this collection requires to be installed for it to be usable. The key of the dict is the
# collection label 'namespace.name'. The value is a version range
# L(specifiers,https://python-semanticversion.readthedocs.io/en/latest/#requirement-specification). Multiple
version
# range specifiers can be set and are separated by ','
dependencies: {}

# The URL of the originating SCM repository
repository: http://example.com/repository

# The URL to any online docs
documentation: http://docs.example.com

# The URL to the homepage of the collection/project
homepage: http://example.com

# The URL to the collection issue tracker
issues: http://example.com/issue/tracker

# A list of file glob-like patterns used to filter any files or directories that should not be included in the
build
# artifact. A pattern is matched from the relative path of the file or directory of the collection directory.
This
# uses 'fnmatch' to match the files or directories. Some directories and files like 'galaxy.yml', '*.pyc',
```

```
'*.retry',
# and '.git' are always filtered. Mutually exclusive with 'manifest'
build_ignore: []

# A dict controlling use of manifest directives used in building the collection artifact. The key 'directives' is
a
# list of MANIFEST.in style
# L(directives,https://packaging.python.org/en/latest/guides/using-manifest-in/#manifest-in-commands). The key
# 'omit_default_directives' is a boolean that controls whether the default directives are used. Mutually
exclusive
# with 'build_ignore'
# manifest: null
```

Éditez la ligne **requires-ansible** dans le fichier **~/collections/ittraining/ittraining/meta/runtime.yml** :

```
[trainee@autohub collection]$ vi ittraining/ittraining/meta/runtime.yml

[trainee@autohub collection]$ cat ittraining/ittraining/meta/runtime.yml
---
# Collections must specify a minimum required ansible version to upload
# to galaxy
requires_ansible: '>=2.9.10'

# Content that Ansible needs to load from another location or that has
# been deprecated/removed
# plugin_routing:
#   action:
#     redirected_plugin_name:
#       redirect: ns.col.new_location
#     deprecated_plugin_name:
#       deprecation:
#         removal_version: "4.0.0"
#         warning_text: |
#           See the porting guide on how to update your playbook to
```

```
#           use ns.col.another_plugin instead.
#     removed_plugin_name:
#       tombstone:
#         removal_version: "2.0.0"
#         warning_text: |
#           See the porting guide on how to update your playbook to
#           use ns.col.another_plugin instead.
#   become:
#   cache:
#   callback:
#   cliconf:
#   connection:
#   doc_fragments:
#   filter:
#   httpapi:
#   inventory:
#   lookup:
#   module_utils:
#   modules:
#   netconf:
#   shell:
#   strategy:
#   terminal:
#   test:
#   vars:

# Python import statements that Ansible needs to load from another location
# import_redirection:
#   ansible_collections.ns.col.plugins.module_utils.old_location:
#     redirect: ansible_collections.ns.col.plugins.module_utils.new_location

# Groups of actions/modules that take a common set of options
# action_groups:
#   group_name:
```

```
#     - module1
#     - module2
```

Placez-vous dans le répertoire **~/collections/ittraining/ittraining/** :

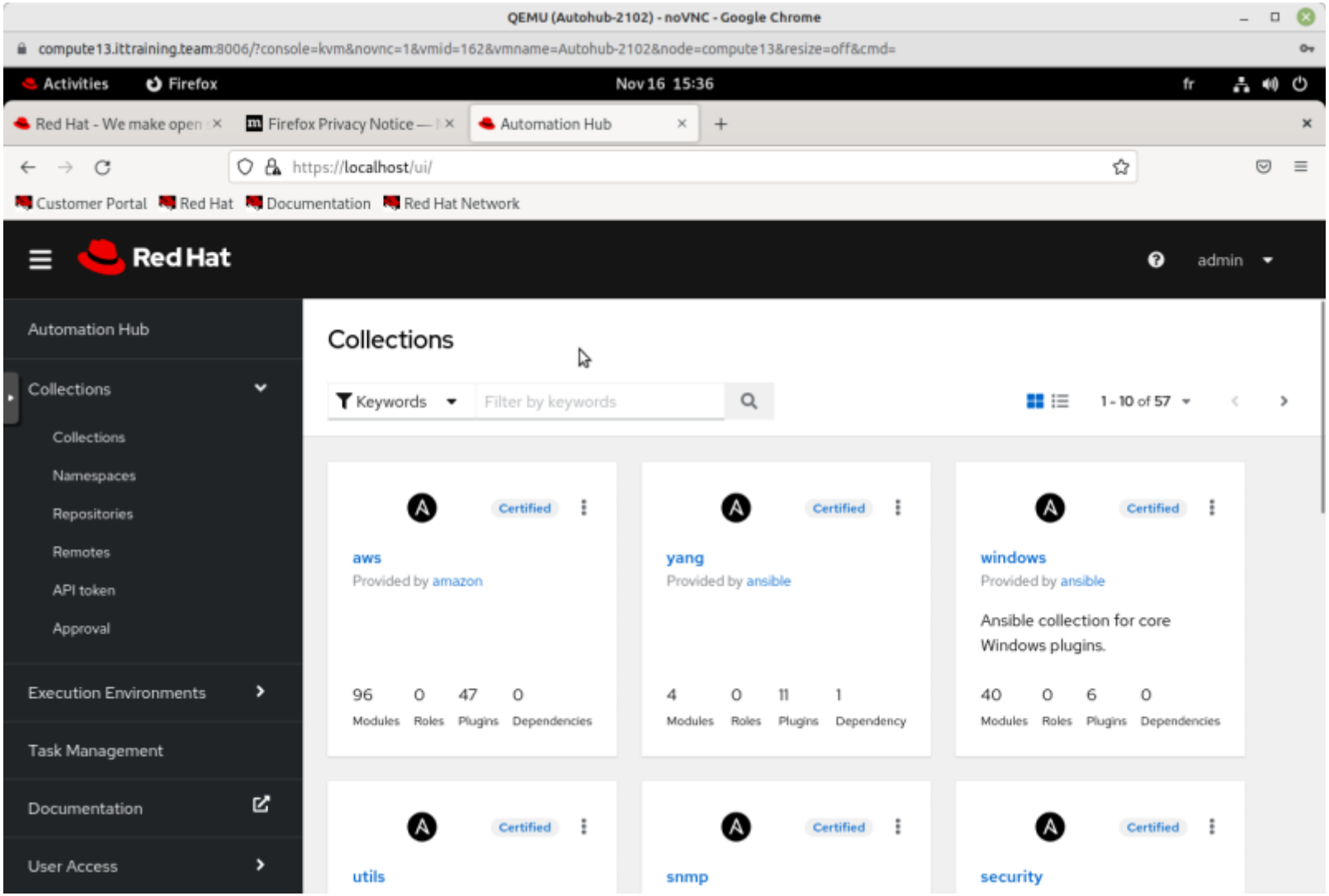
```
[trainee@autohub collection]$ cd ittraining/ittraining/

[trainee@autohub ittraining]$ ls
docs  galaxy.yml  meta  plugins  README.md  roles
```

Construisez la Collection **ittraining-ittraining-1.0.0.tar.gz** avec la commande **ansible-galaxy collection build** :

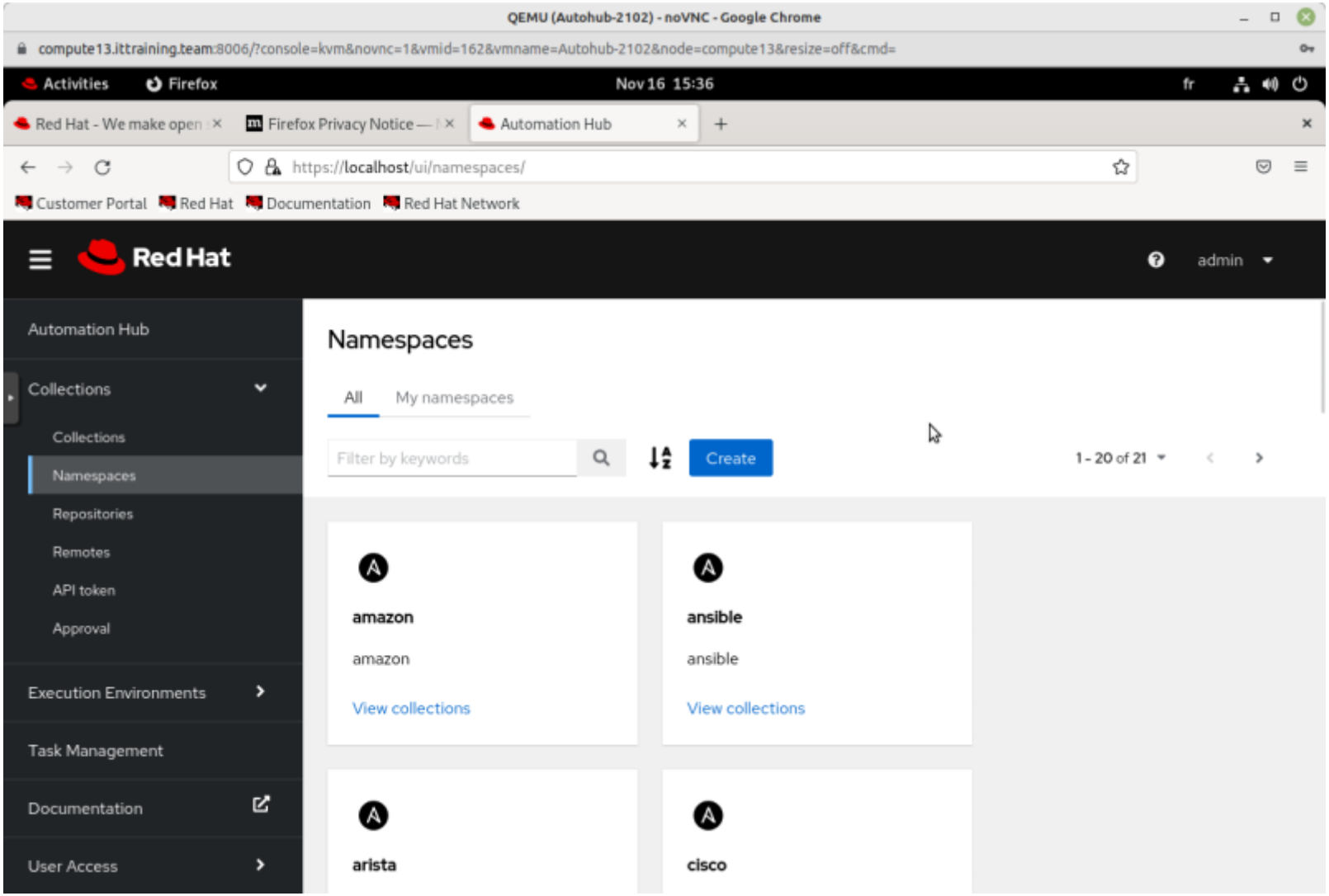
```
[trainee@autohub ittraining]$ ansible-galaxy collection build
Created collection for ittraining.ittraining at /home/trainee/collection/ittraining/ittraining/ittraining-
ittraining-1.0.0.tar.gz
```

6.2 - Téléverser une Collection

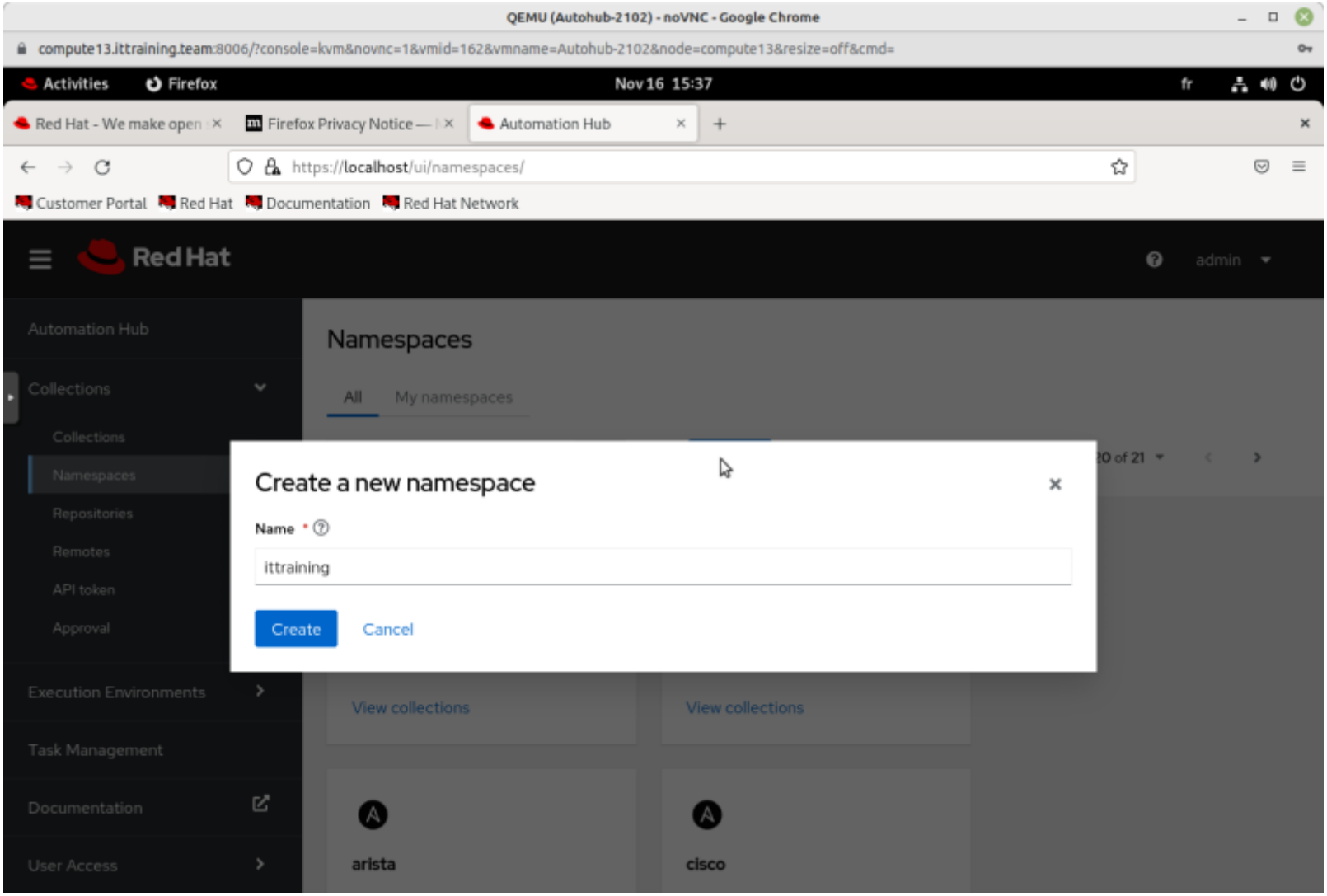
Connectez-vous à votre **VM Debian_10.0.2.46_VNC**. Ouvrez un navigateur Web et naviguez à **https://10.0.2.102** :



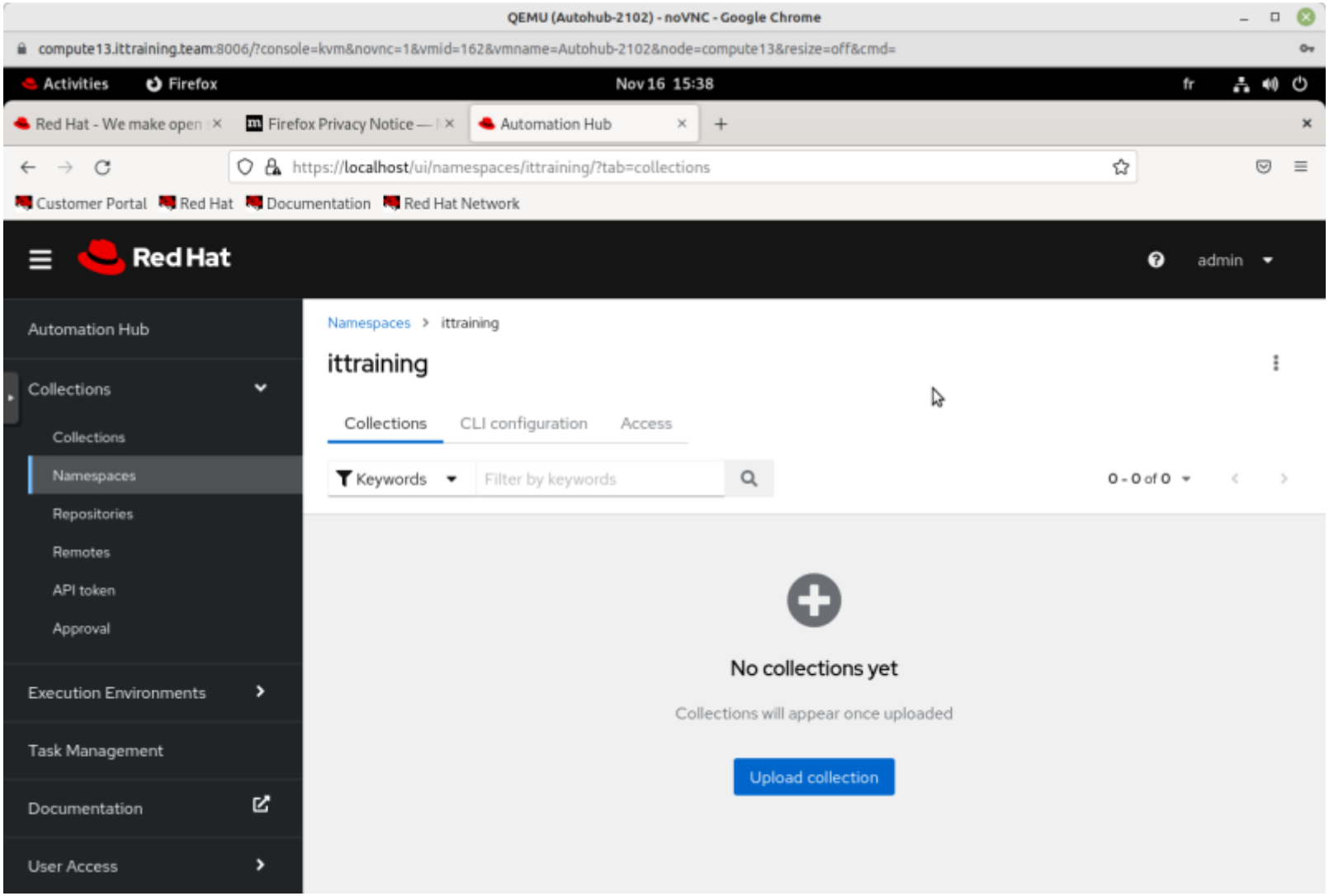
Cliquez ensuite sur **Namespaces** dans le menu de gauche puis cliquez sur le bouton **Create** :



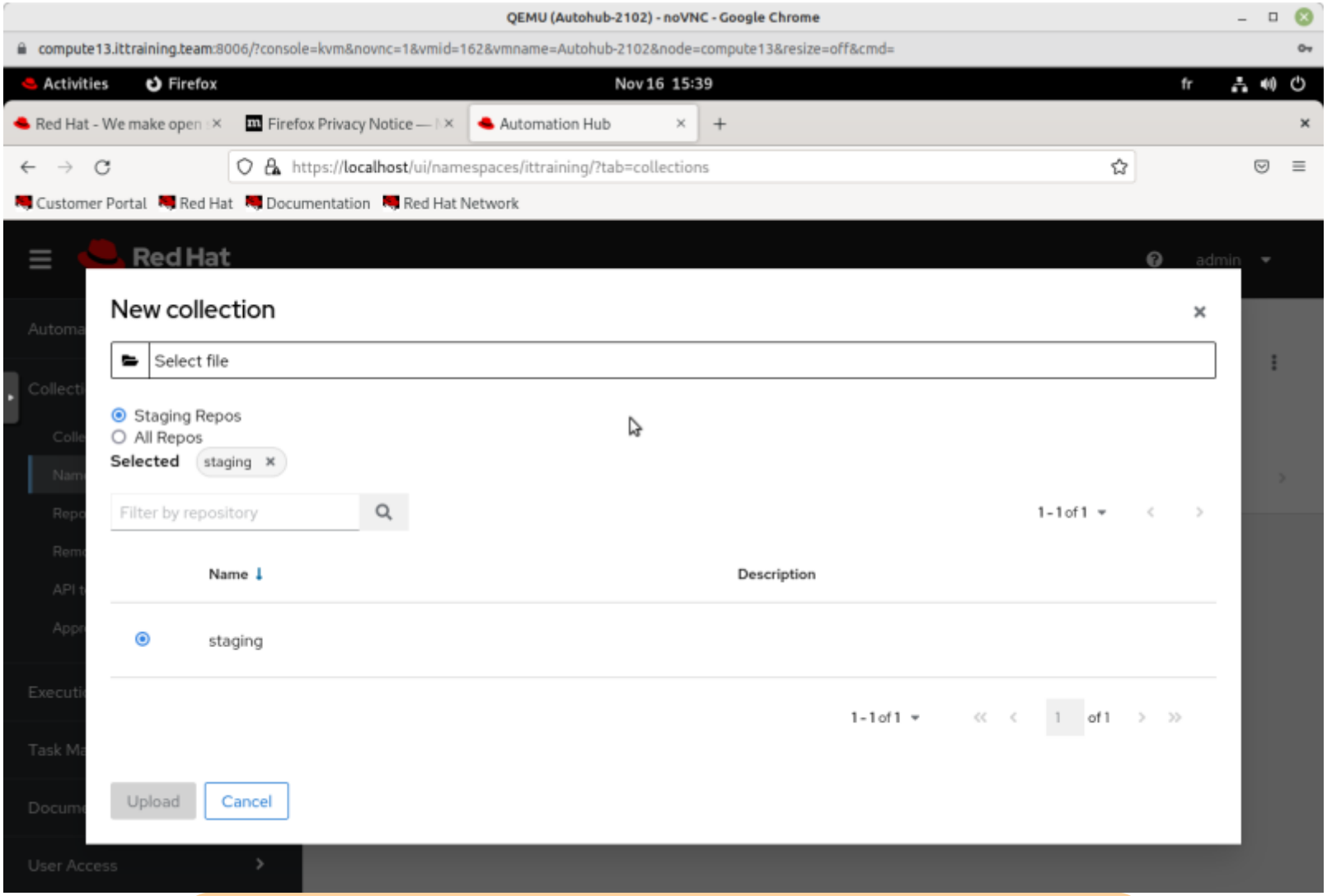
Renseignez le nom **ittraining** et cliquez sur le bouton **Create** :



Notez que pour l'instant, il n'y a pas de **Collections** dans ce **Namespace**. Cliquez donc sur le bouton **Upload collection** :

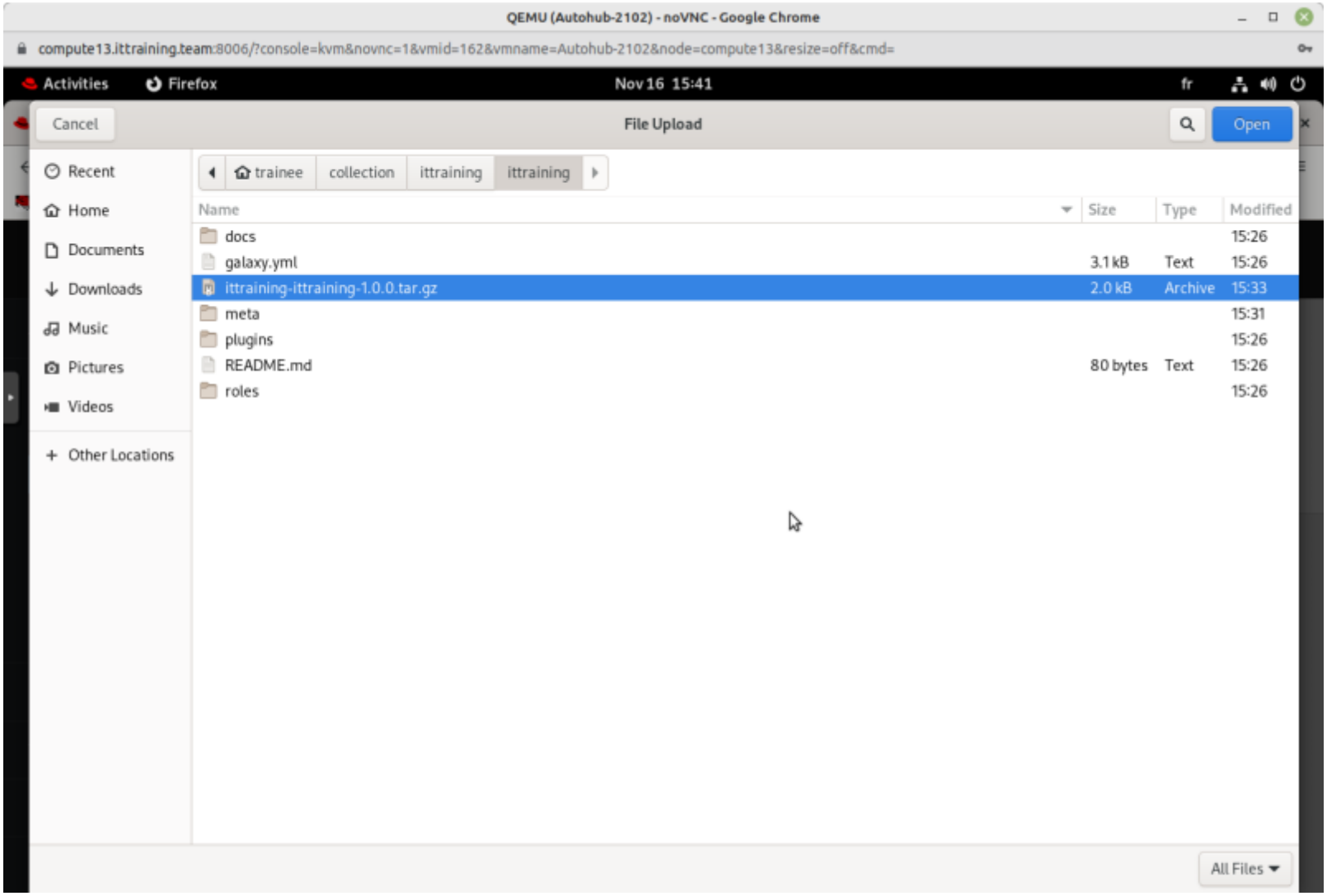


Dans la fenêtre **New collection**, cliquez sur **Select file** :

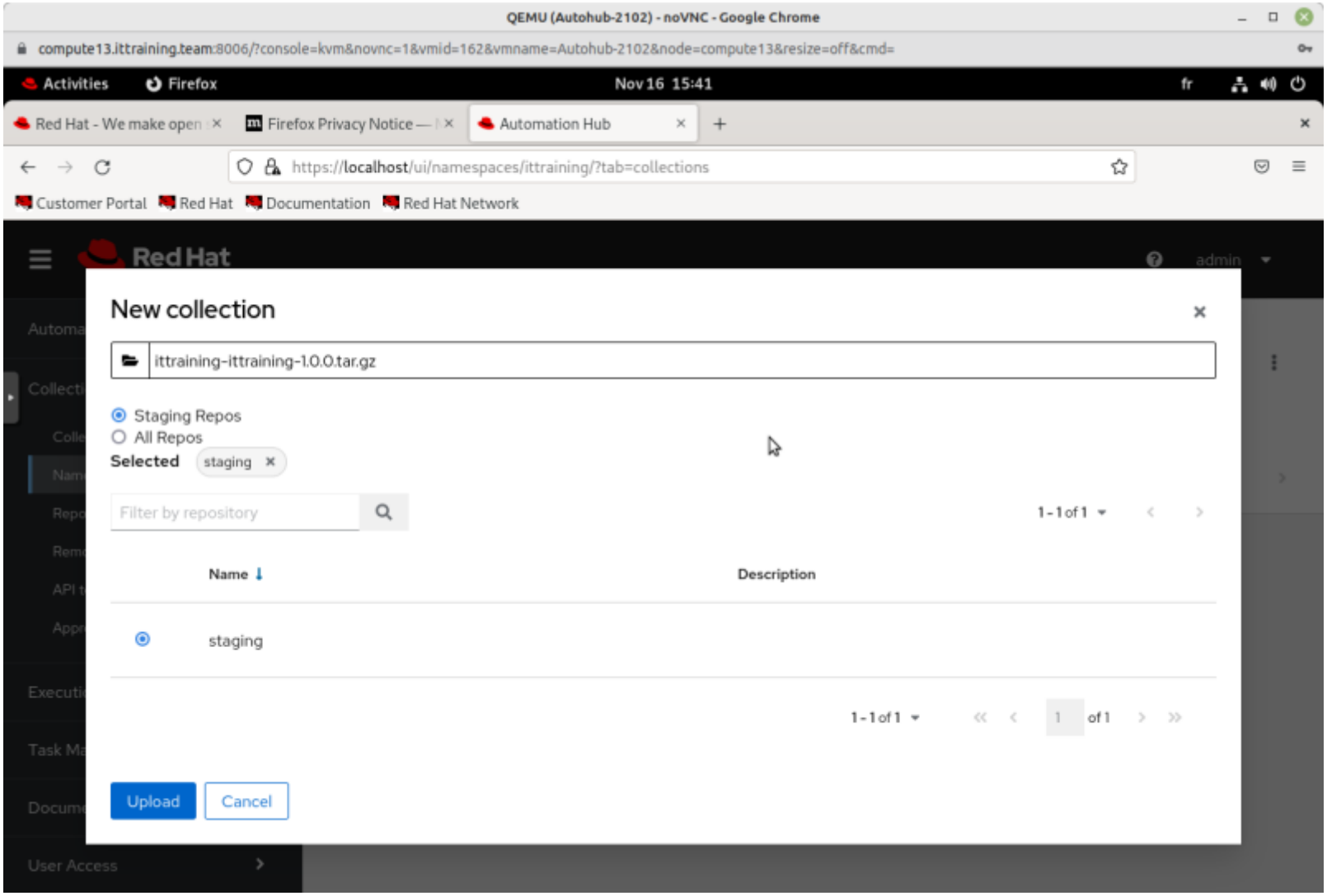


Important : Notez que le dépôt de destination est **staging**. Le dépôt staging est utilisé pour téléverser toute nouvelle Collection avant que celle-ci soit vérifiée pour publication.

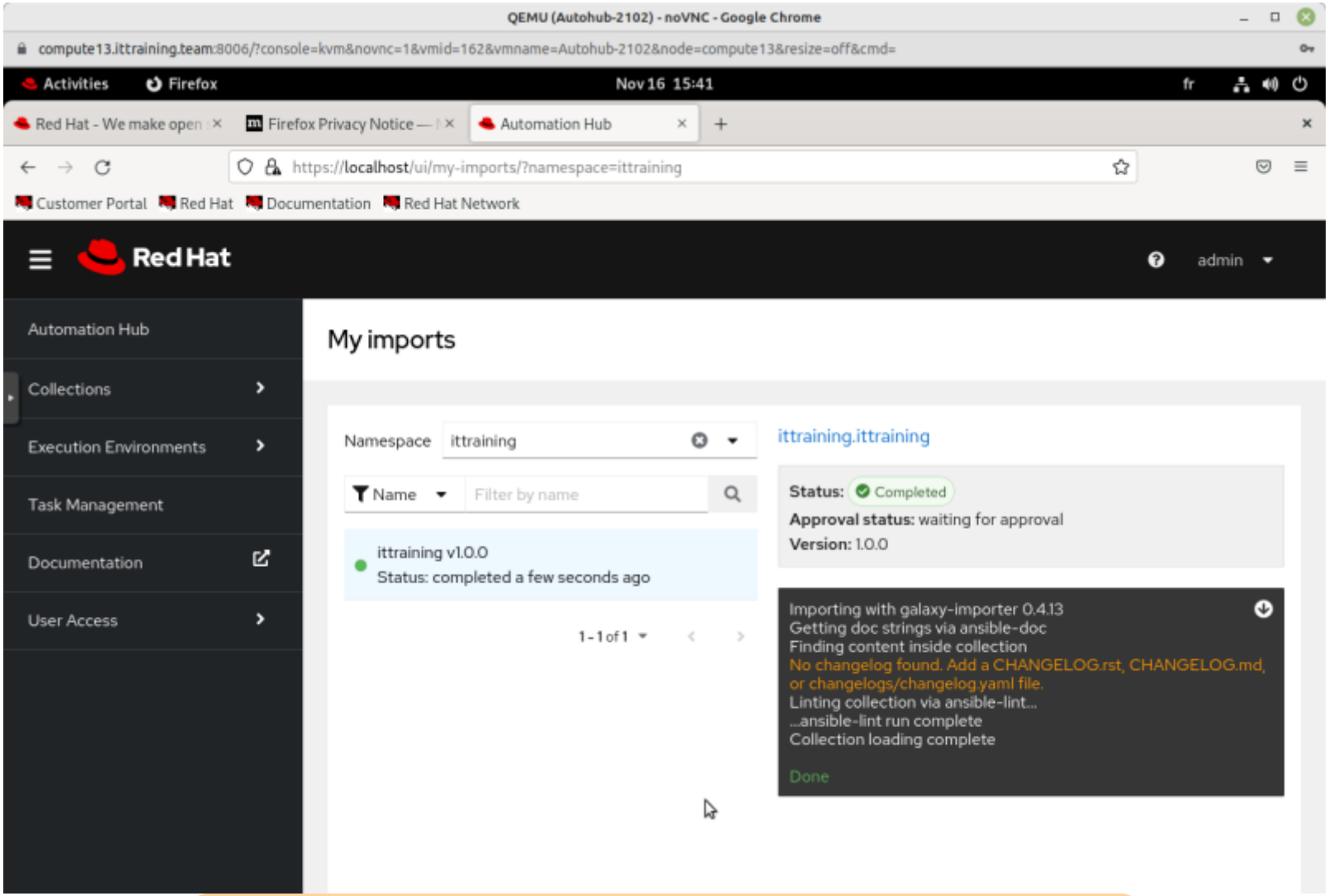
Choisissez le fichier **ittraining-ittraining-1.0.0.tar.gz** :



Cliquez ensuite sur le bouton **Upload** :

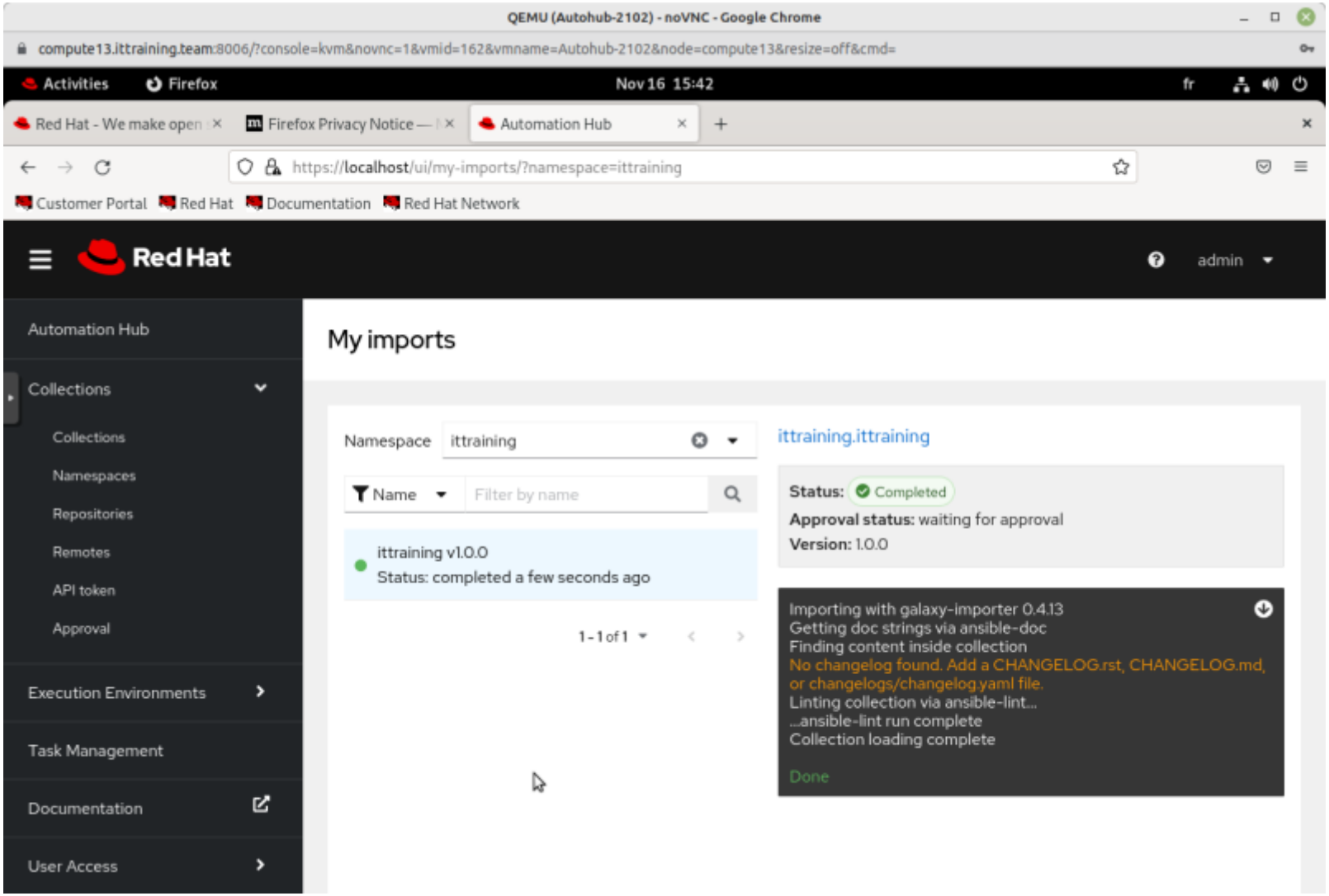


A l'issu de quelques secondes, la **Collection** est téléversée :

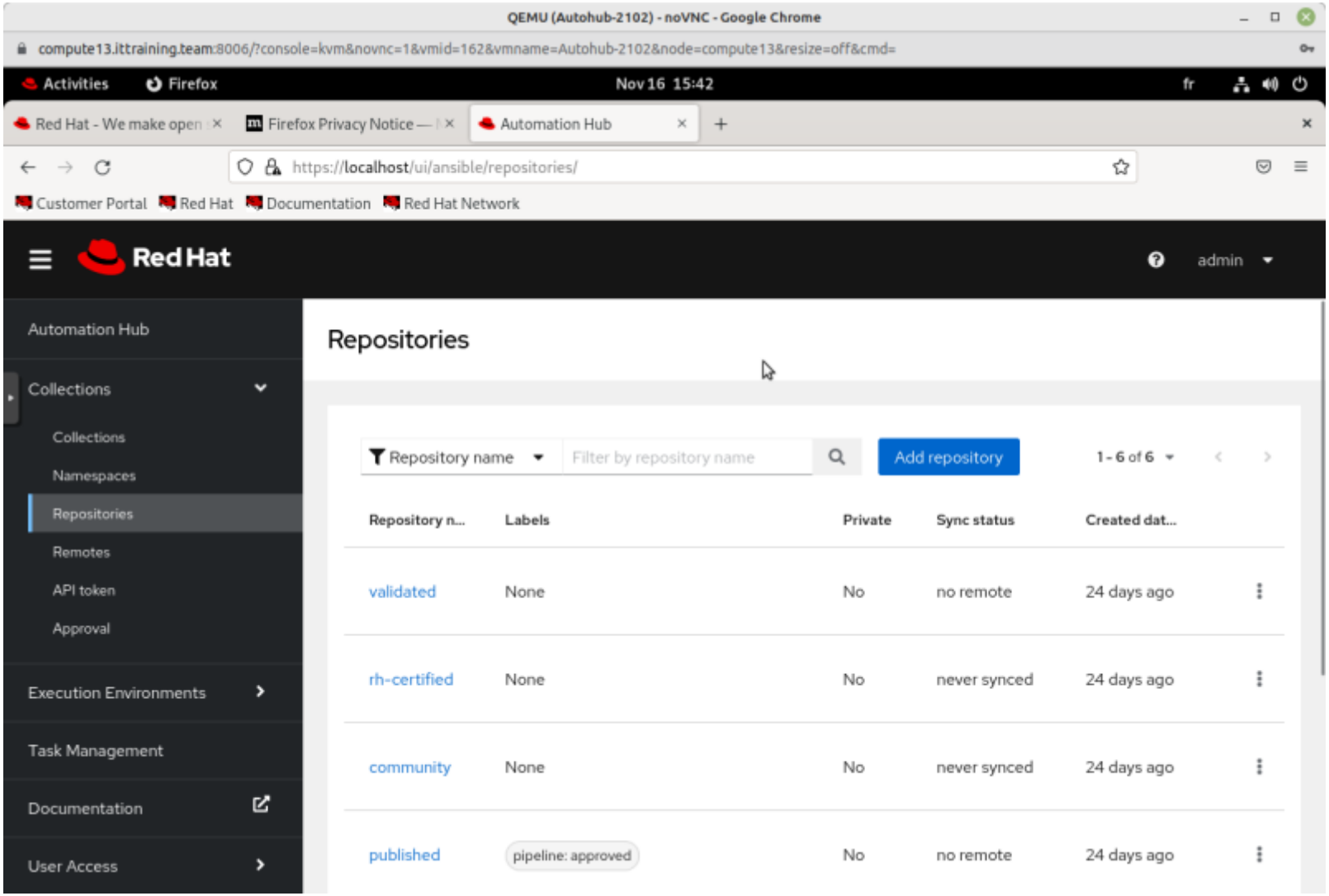


Important : Notez la ligne **Approval status: waiting for approval**.

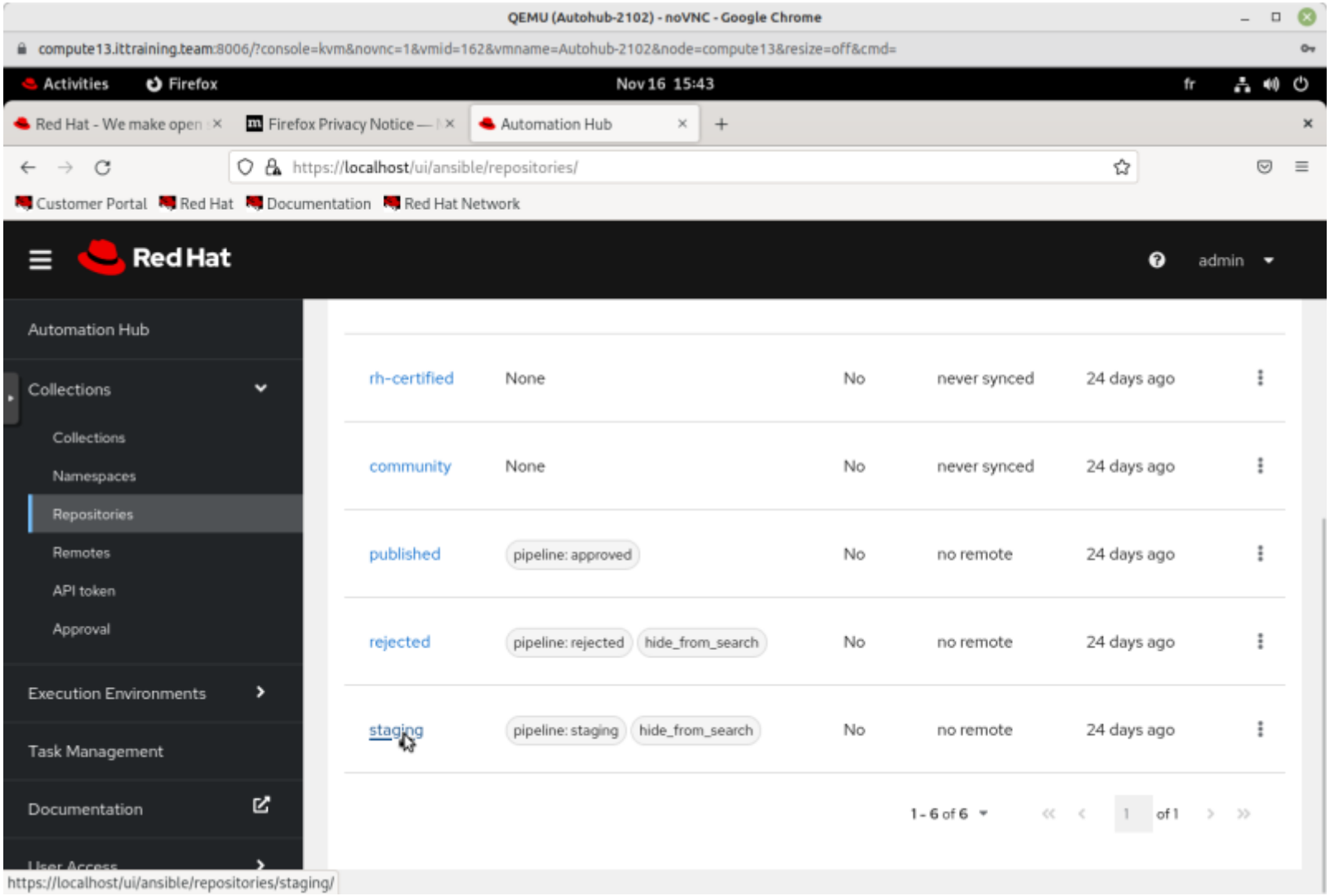
Développez l'item **Collections** dans le menu de gauche :



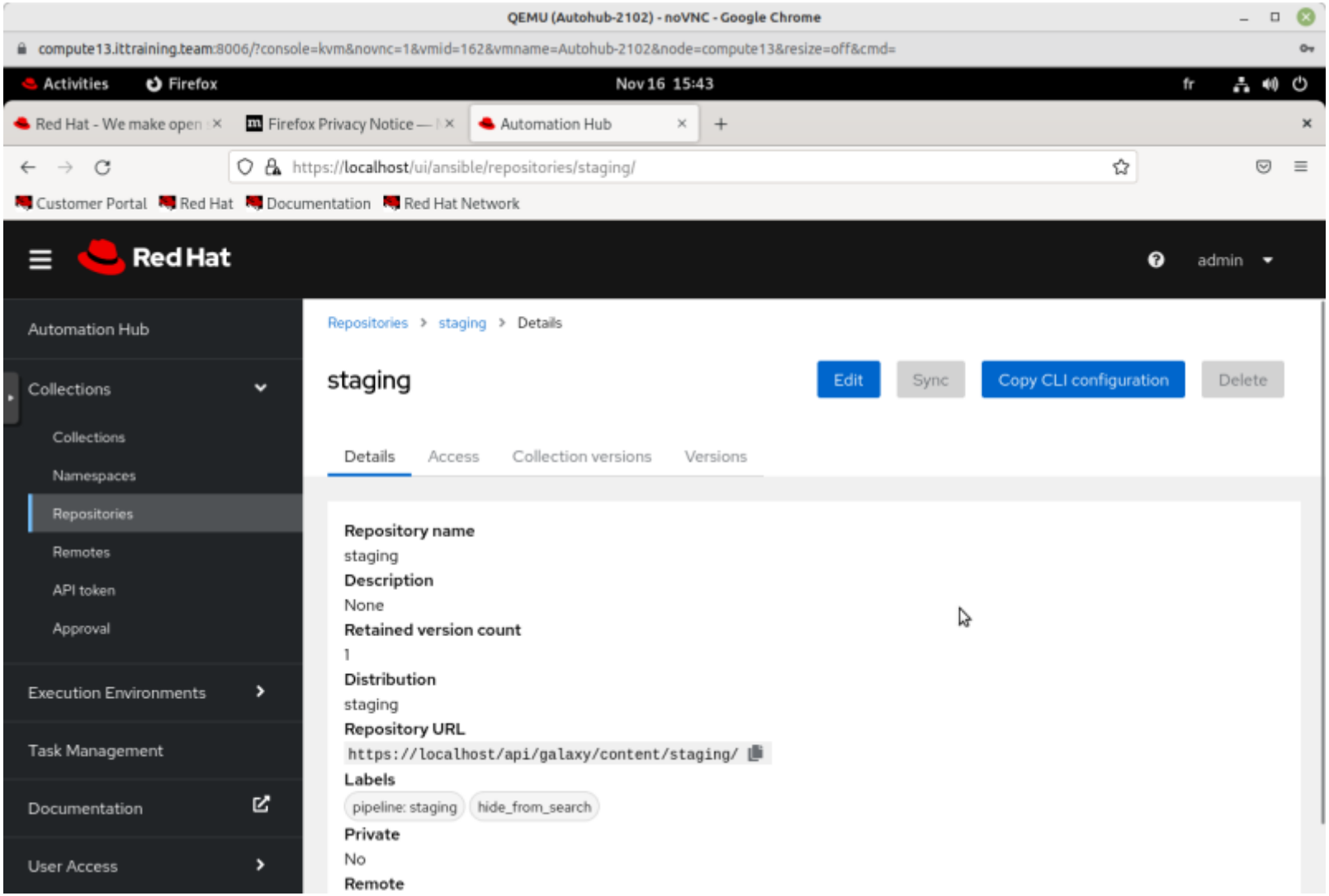
Cliquez sur **Repositories** :



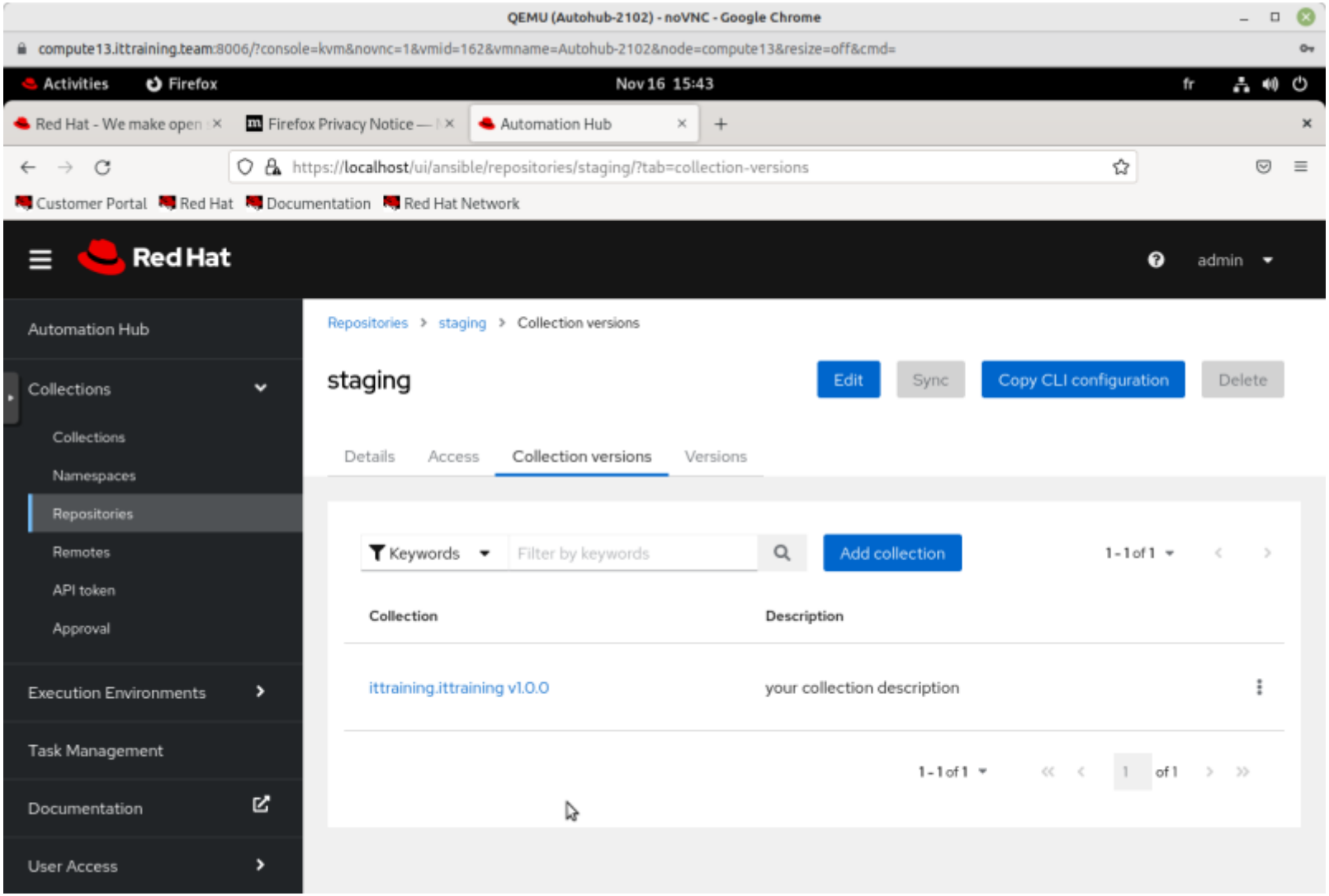
Descendez en bas de la liste des **Repositories** et cliquez sur **staging** :



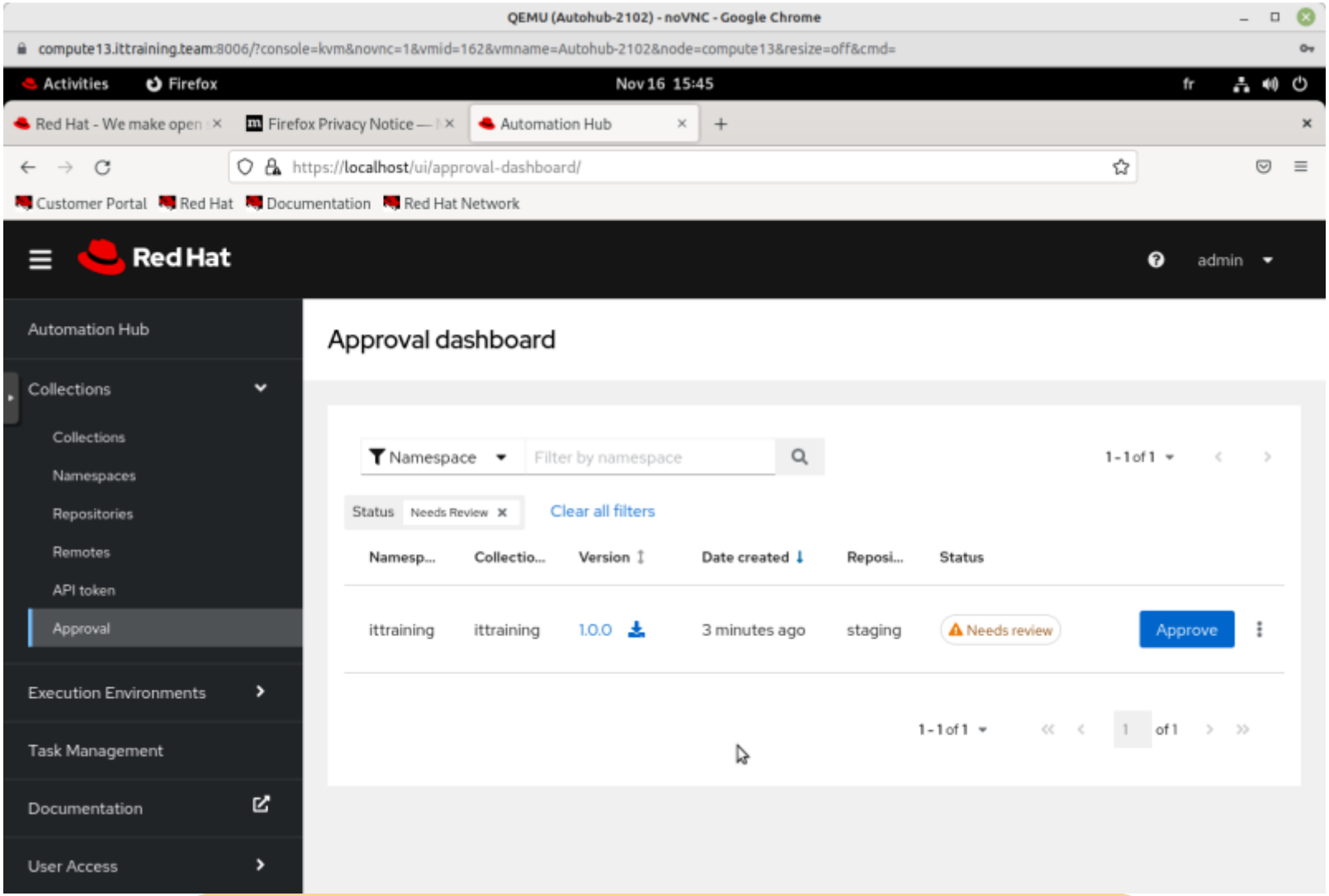
Vous obtiendrez :



Cliquez sur l'onglet **Collection versions** et notez la présence de la Collection **ittraining-ittraining-1.0.0** :

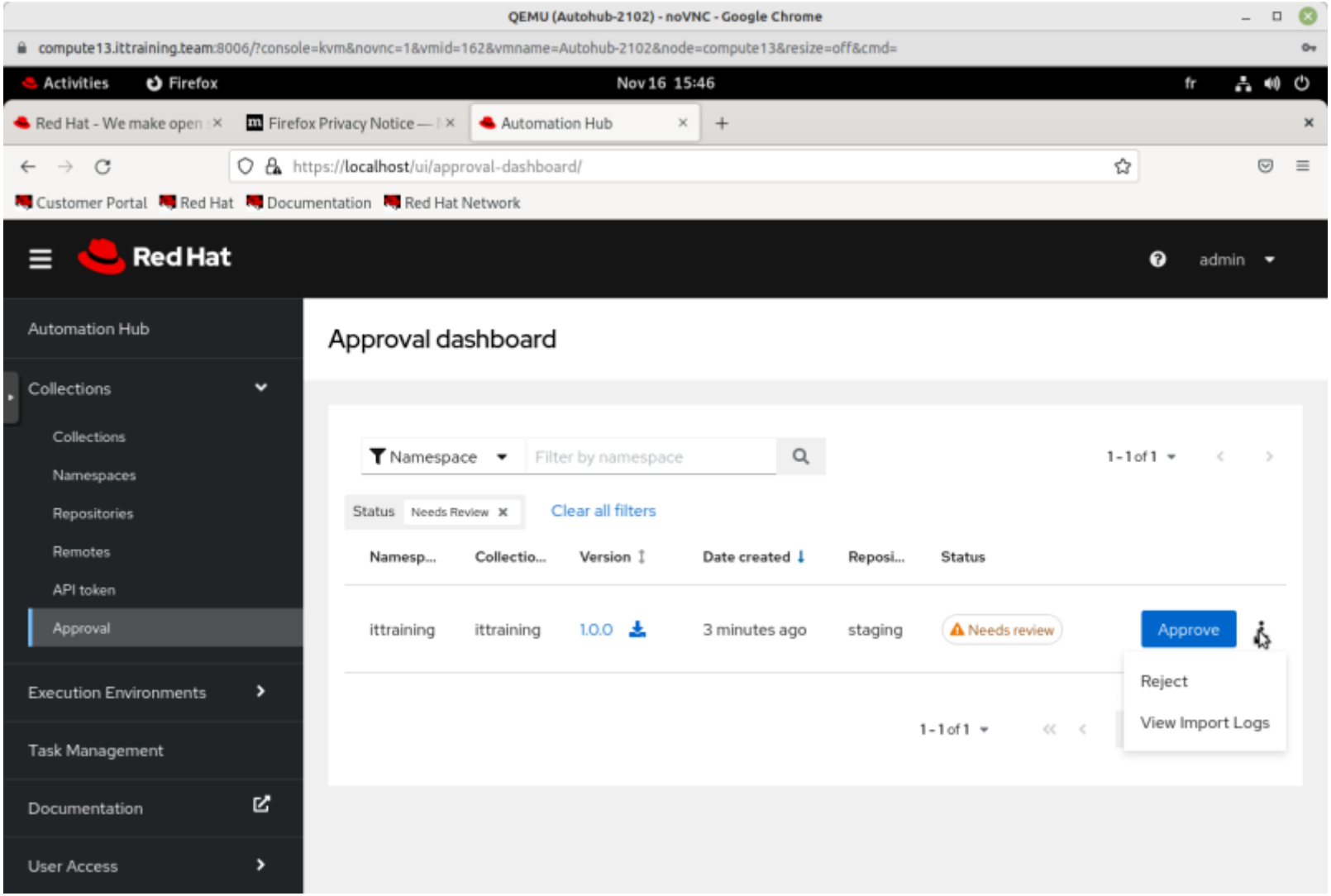


Cliquez ensuite sur l'item **Approval** dans le menu de gauche :

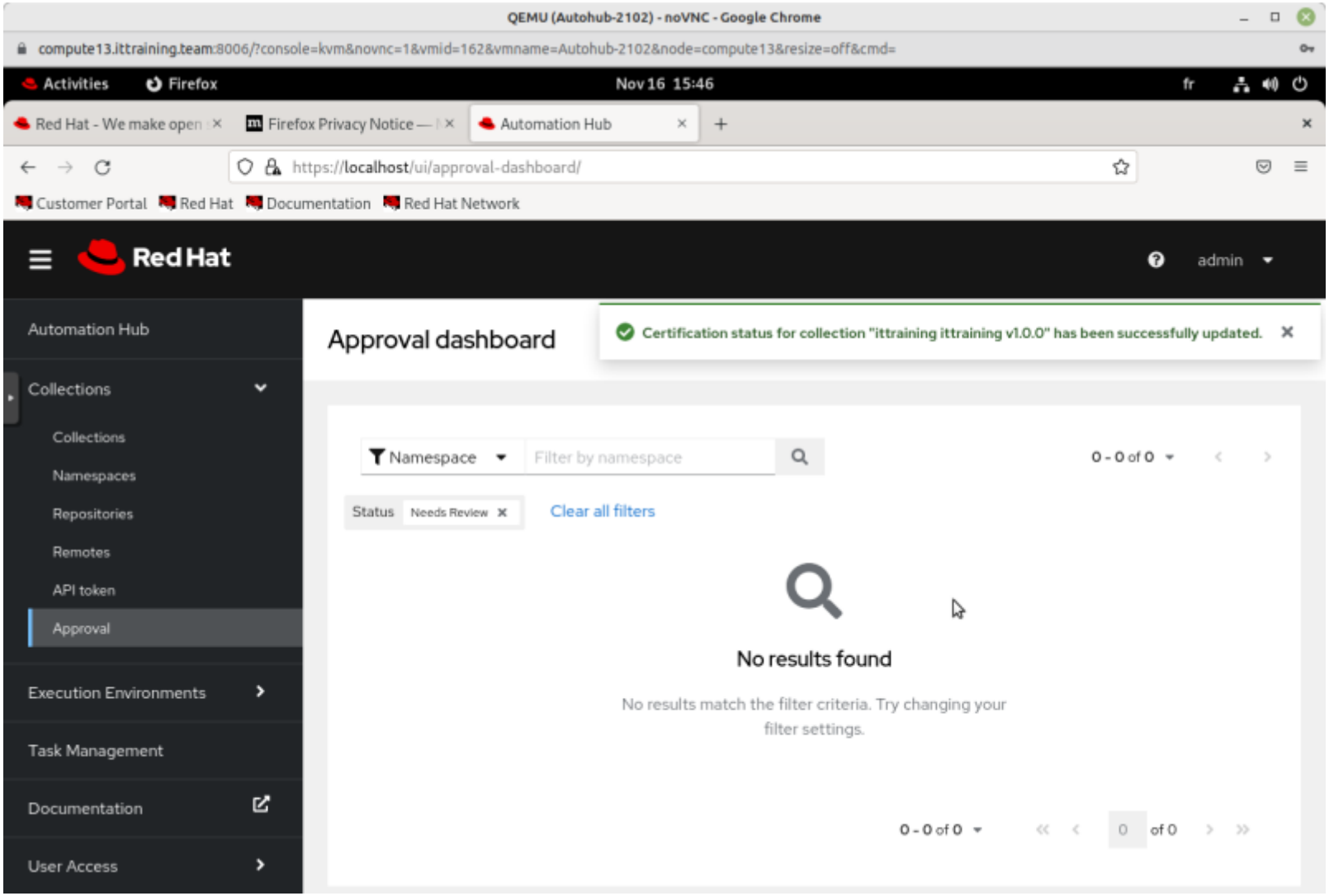


Important : Notez que l'item **Approvals** est présent parce que vous êtes connecté en tant qu'administrateur.

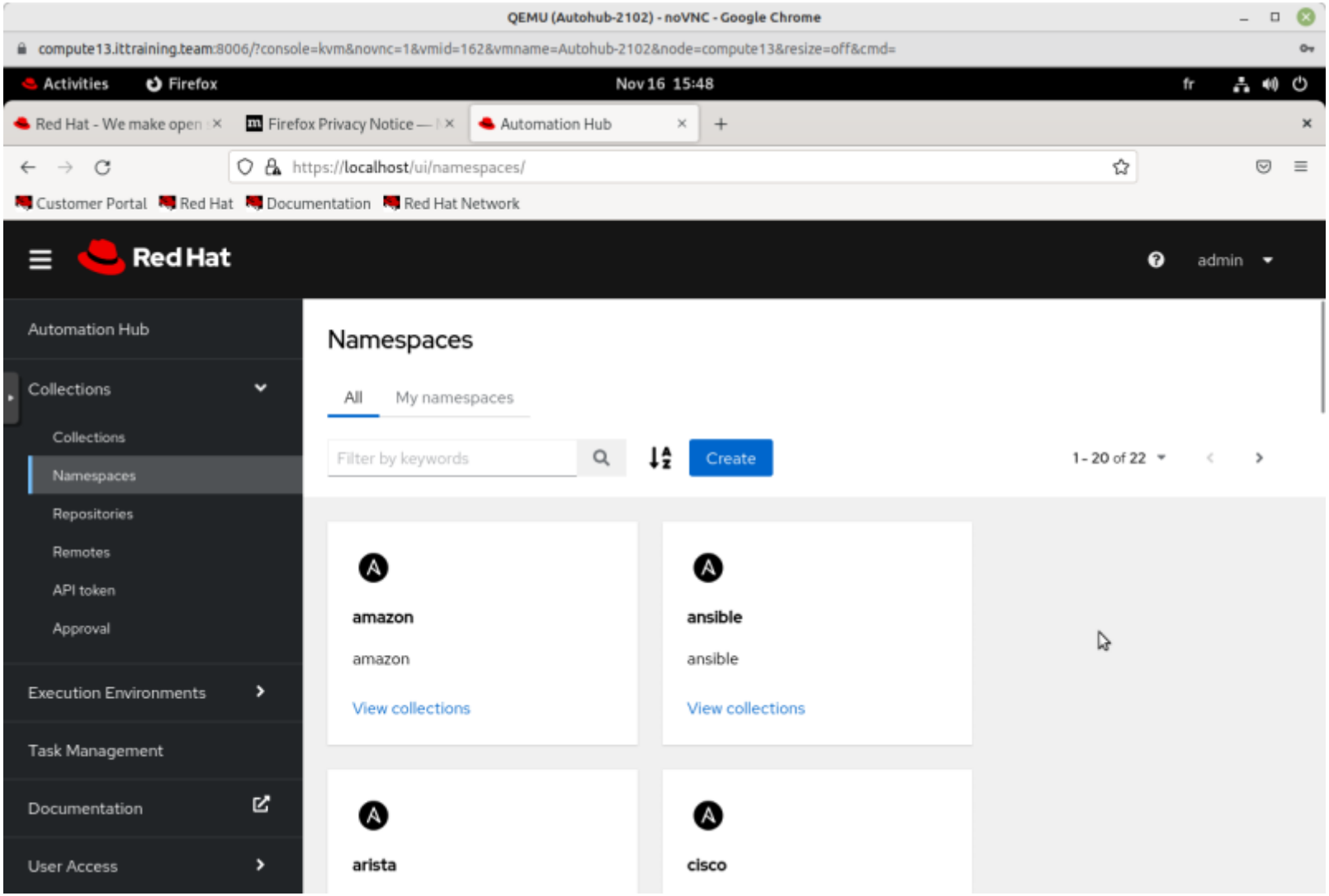
Dans le cas où vous souhaiteriez rejeter cette Collection, il conviendrait de cliquez sur les trois points verticaux à droite de la ligne **ittraining** et de choisir **Reject**. Cliquez ensuite sur le bouton **Approve** :



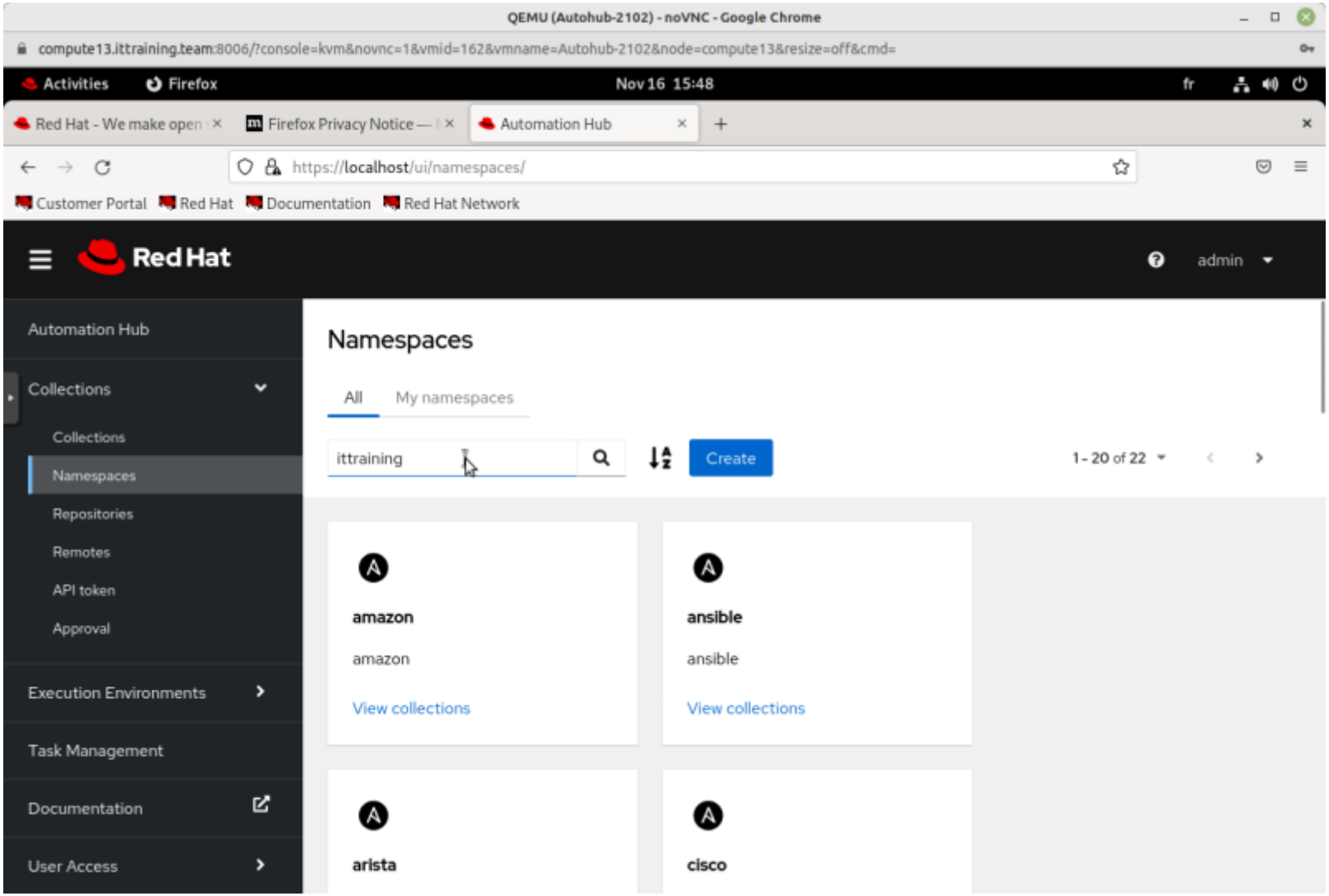
Vous obtiendrez :



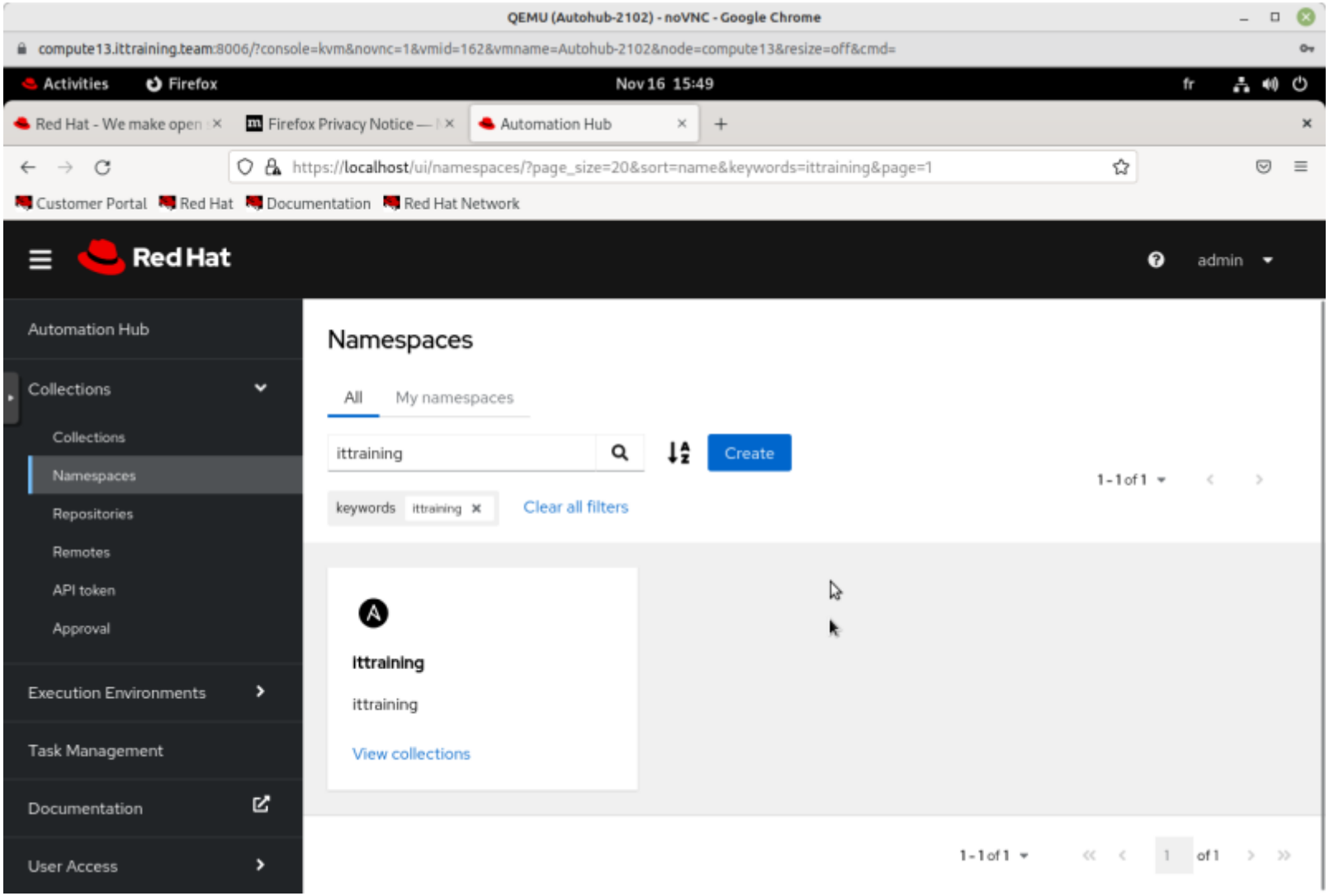
Cliquez sur **Namespaces** dans le menu de gauche :



Dans le champs **Filter by keywords**, saisissez le mot **ittraining** puis cliquez sur l'**icône de la loupe** :



Cliquez sur le lien **View collections** pour consulter la liste des **Collections** dans le **Namespace** :



Constatez la présence de la Collection **ittraining** :