redhat.

ANSIBLE WINDOWS

Introduction à Ansible Windows - atelier

Michael Lessard
Architecte de solutions sénior
mlessard@redhat.com

Sébastien Perreault
Architecte de solutions
sperreault@redhat.com

Eric Beaudoin
Technical Account Manager
ebeaudoin@redhat.com

Marcos Garcia
Architecte infonuagique
mgarcia@redhat.com

AVERTISSEMENT

CECI EST UNE INTENSE FORMATION D'INTRODUCTION GRATUITE OFFERTE PAR RED HAT

ELLE N'A AUCUN LIEN AVEC NOTRE GROUPE GLS

CET ATELIER EST BASÉ SUR LE WORKSHOP ANSIBLE POUR LINUX

ORDRE DU JOUR

Formation Ansible Windows

1 **Introduction à Ansible**

2 **Configuration Poste Travail**
PRÉ-REQUIS

3 **Mon premier playbook**
LAB: Gestion logiciels dans windows

4 **Playbooks et modules Windows**
LAB: Installer et configurer IIS

5 **Ansible et Tower en profondeur**
LAB guidé

6 **Gestion dynamique de VMs Azure**
LAB guidé

INTRODUCTION À ANSIBLE

Ansible est...

SIMPLE

Automatisation facile
Pas besoin d'être programmeur
Les tâches sont exécutées en ordre
Utilisable par tous
Devenez productif rapidement

PUISSANT

Déploiement d'application
Gestion de configuration
Orchestration de workflow
Automatisation des réseaux
Orchestrer le cycle de vie complet

SANS AGENT

Sans agent
Utilise OpenSSH & WinRM
Pas d'agent à exploiter ou maintenir
Démarrer immédiatement
Plus efficace, plus sécure

Introduction à Ansible

Michael DeHaan (créateur de Cobbler et de Func)
https://www.ansible.com/blog/2013/12/08/the-origins-of-ansible

Simple
AUTOMATISE TOUT
Peut gérer presque n'importe lequel *IX par le biais d'un protocole SSH
    nécessite Python
Windows (PowerShell, module WinRM Python)
Composants de nuage, virtualisation, conteneur, réseau

« Ansible doit une grande partie de ses origines au temps que j'ai passé au sein du groupe des technologies émergentes de Red Hat, qui était une unité de R D sous la direction du CTO de Red Hat. »

- Michael DeHaan

«...parce que Puppet était trop déclaratif, vous ne pouviez pas l'utiliser pour faire des choses comme réinitialiser des serveurs ou effectuer toutes les tâches ad hoc qui devaient être faites entretemps…»

- Michael DeHaan


Une **ansible** est un dispositif **théorique** permettant de réaliser des communications à une vitesse supraluminique. Elle peut envoyer et recevoir des messages en provenance et en direction du périphérique correspondant sur n'importe quelle distance sans aucun délai. Les **ansibles** sont une composante emblématique de la littérature de **science-fiction**.
-- Wikipédia
redhat.

30,000+
Stars on GitHub

1500+
Ansible modules

500,000+
Downloads a month

STARS	TECHNO	CONTRIBUTEURS
30,404	Ansible	3 499
16,728	Vagrant	824
12,218	Teraform	1 207
8,867	Salt	2 067
5,340	Chef	557
5,001	Puppet	493


RHUG - RED HAT USER GROUP
https://www.meetup.com/RHUGQuebec
ANSIBLE QUEBEC MEETUP
https://www.meetup.com/Ansible-Quebec/

1 Introduction à Ansible

CONFIG MANAGEMENT

Centralizing configuration file management and deployment is a common use case for Ansible, and it's how many power users are first introduced to the Ansible automation platform.

APP DEPLOYMENT

When you define your application with Ansible, and manage the deployment with Tower, teams are able to effectively manage the entire application lifecycle from development to production.

PROVISIONING

Your apps have to live somewhere. If you're PXE booting and kickstarting bare-metal servers or VMs, or creating virtual or cloud instances from templates, Ansible and Ansible Tower help streamline the process.

NETWORK AUTOMATION

Ansible's simple automation framework means that previously isolated network administrators can finally speak the same language of automation as the rest of the IT organization, extending the capabilities of Ansible to include native support for both legacy and open network infrastructure devices.

CONTINUOUS DELIVERY

Creating a CI/CD pipeline requires buy-in from numerous teams. You can't do it without a simple automation platform that everyone in your organization can use. Ansible Playbooks keep your applications properly deployed (and managed) throughout their entire lifecycle.

SECURITY & COMPLIANCE

When you define your security policy in Ansible, scanning and remediation of site-wide security policy can be integrated into other automated processes and instead of being an afterthought, it'll be integral in everything that is deployed.

ORCHESTRATION

Configurations alone don't define your environment. You need to define how multiple configurations interact and ensure the disparate pieces can be managed as a whole. Out of complexity and chaos, Ansible brings order.

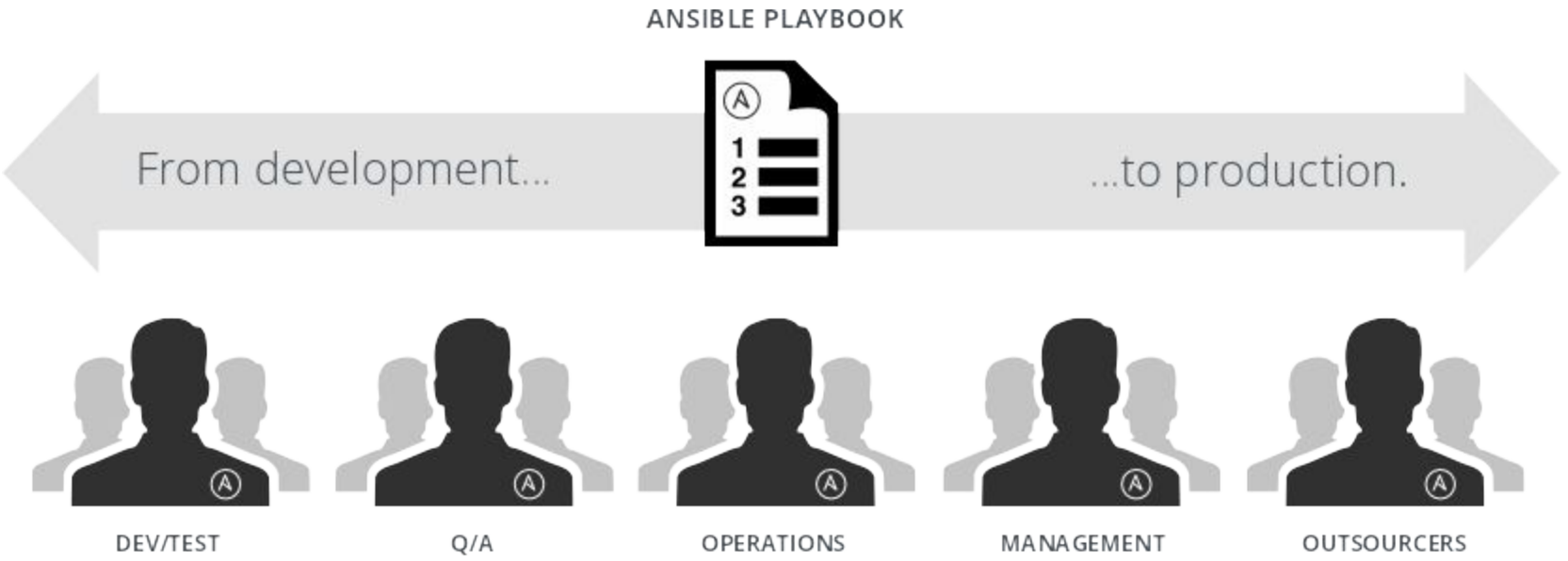

EN 30 ANS, LA GESTION DES RÉSEAUX
N’A PAS CHANGÉE.

POURQUOI **ANSIBLE** + RÉSEAU?

« Lorsqu'on leur a demandé ce qui selon eux était le composant le plus immature en gestion du nuage, 76 % ont dit que c'était le réseau; 15 % ont mentionné le traitement et 9 % le stockage. »

Rapport de gestion nuagique SDx 2015 : OpenStack and More sdxcentral.com

ANSIBLE - LE LANGAGE DE DEVOPS



COMMUNICATION IS THE KEY TO DEVOPS.

Ansible is the first **automation language** that can be read and written across IT.

Ansible is the only **automation engine** that can automate the entire **application lifecycle** and **continuous delivery** pipeline.

2 Configuration poste de travail

COMPOSANTS CLÉS

Comprendre les termes d'Ansible

- ★ **Playbook** **(Plan)**
- ★ **Plays**
- ★ **Tasks**
- ★ **Modules** **(Tools)**
- ★ **Inventory**

INSTALLATION D' ANSIBLE

Mode d'emploi

```
# CENTOS
# INSTALLER LE REPO EPEL
yum install epel-release

# RHEL
# ACTIVER LE REPO ANSIBLE
subscription-manager repos --enable rhel-7-server-ansible-VERSION-rpms


# INSTALLER ANSIBLE
yum install ansible
```

Ansible est disponible dans le Azure Shell !

https://docs.microsoft.com/en-us/azure/ansible/

Est-ce que Red Hat offre du soutien pour Ansible?

https://access.redhat.com/articles/2271461

MODULES

En quoi ça consiste?

Bouts de code copiés sur le système cible.
Exécutés pour satisfaire la déclaration de tâche.
Personnalisables.

Les modules Ansible sont tous écrit en Python et Powershell, mais les modules peuvent être écrits en n'importe quel langage.

MODULES

Vaste choix / force secrète d'Ansible...

- ➔ **Modules de nuage**
- ➔ **Modules de grappes**
- ➔ **Modules de commandements**
- ➔ **Modules Crypto**
- ➔ **Modules de bases de données**
- ➔ **Modules de fichiers**
- ➔ **Modules d'identités**
- ➔ **Modules d'inventaire**
- ➔ **Modules de messages**
- ➔ **Modules de surveillance**
- ➔ **Modules de réseaux**
- ➔ **Modules de notification**
- ➔ **Modules de gestion à distance**
- ➔ **Modules d'intégration**
- ➔ **Modules de contrôle à la source**
- ➔ **Modules de stockage**
- ➔ **Modules de système**
- ➔ **Modules de logiciels utilitaires**
- ➔ **Modules d'infrastructures Web**
- ➔ **Modules Windows**

MODULES

Documentation

```
# AFFICHE TOUS LES MODULES
ansible-doc -l


# ACCÉDER À LA DOCUMENTATION D'UN MODULE
ansible-doc <module_name>
```

http://docs.ansible.com/ansible/latest/modules/modules_by_category.html
http://docs.ansible.com/ansible/devel/modules/modules_by_category.html

MODULES WINDOWS

Fréquemment utilisés

win_chocolatey
win_copy
win_dsc
win_firewall_rule
win_package
win_psexec
win_reboot
win_regedit
win_service
win_shell
win_template
win_updates
win_wait_for

IDEMPO-QUOI?

« En mathématiques et en informatique, le concept d'idempotence signifie essentiellement qu'une opération a le même effet qu'on l'applique une ou plusieurs fois, ou encore qu'en le réappliquant on ne modifiera pas le résultat. »

« Lorsqu'il est soigneusement écrit, un scénario Ansible peut être idempotent afin de prévenir les effets secondaires imprévus sur les systèmes gérés. »

– Wikipédia

INVENTAIRE

Pour utiliser l'inventaire par défaut (/etc/ansible/hosts) ou créer un fichier inventaire

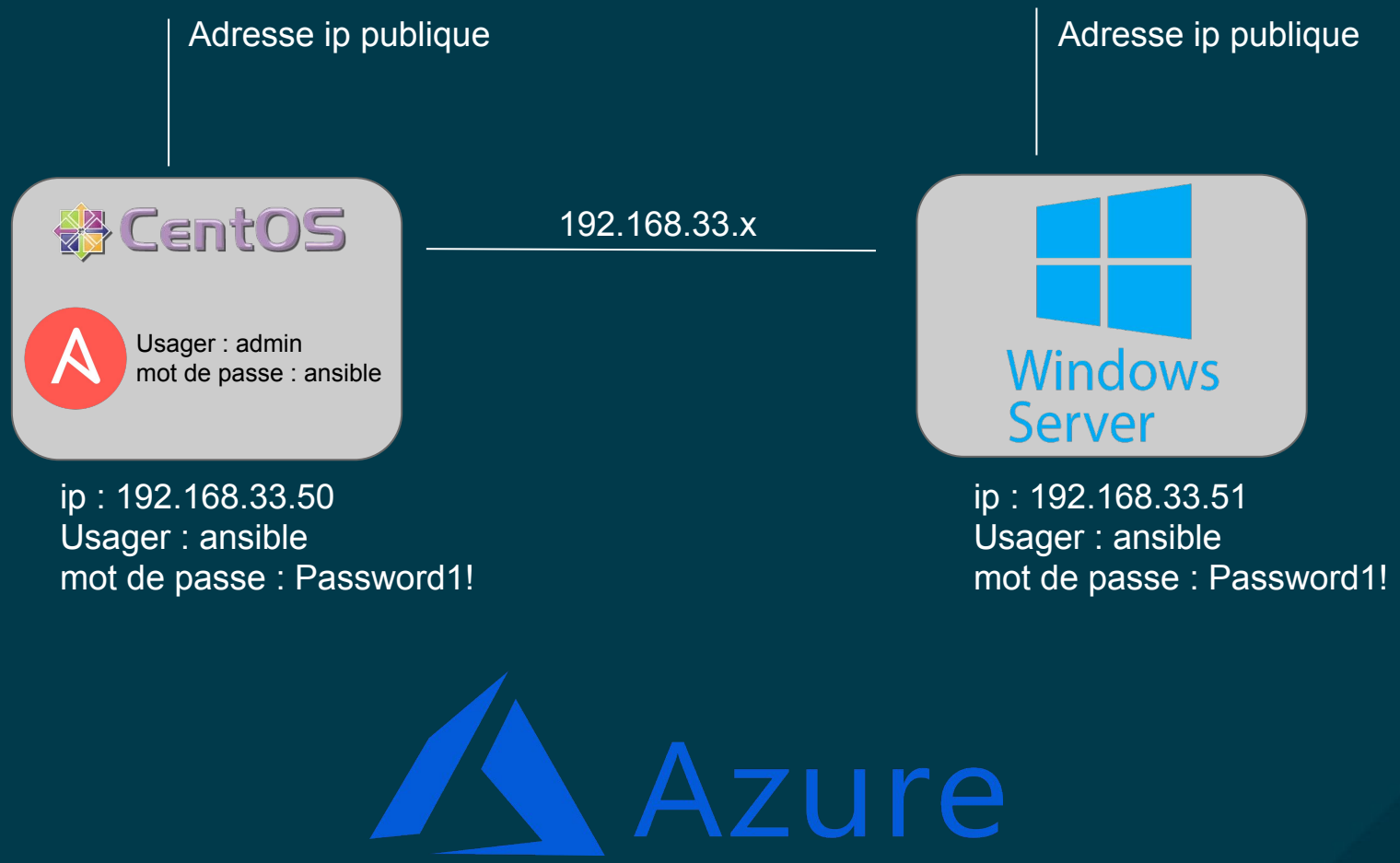
```
[centos@centos1 ~]$ vim inventory

[all:vars]
ansible_ssh_user=ansible
ansible_user=ansible
ansible_password=Password1!
ansible_port=5985
ansible_connection=winrm
ansible_winrm_transport=basic

[web]
windows1 ansible_ssh_host=192.168.33.50

# pour tester la connectivité
[centos@centos1 ~]$ ansible all -i inventory -m win_ping
```

COMMANDE

Pour exécuter votre première commande Ansible...

```
# ansible all -i inventory -m win_ping
        (sur quoi)          (module)  (arguments)
# ansible all -i inventory -m command -a "uptime"

192.168.250.13 | success | rc=0 >>
 18:57:01 up 11:03,  1 user,  load average: 0.00, 0.01, 0.05

192.168.250.11 | success | rc=0 >>
 18:57:02 up 11:03,  1 user,  load average: 0.00, 0.01, 0.05
```

ARCHITECTURE ENVIRONMENT AZURE



Adresses ip externes de vos instances

Pour avoir le détail sur les instances dans Azure

1. Connectez-vous à http://rhmtlsas.eastus.cloudapp.azure.com/22d010fb.html
    a. remplacer la dernière partie par l'id utilisé lors du provisionnement du lab
        i. Exemple : mick444 = http://rhmtlsas.eastus.cloudapp.azure.com/mick444.html

LAB # 2a : PRÉPARATION

Configuration de Putty

1. Connectez-vous à votre Machine Windows sur Azure
2. Lancer Putty
3. Dans le champ Host Name, indiquez l'ip de votre machine CentOS : 192.168.33.51
4. Dans saved sessions, indiquez CentOS puis cliquer sur Save
5. Cliquer sur Open
6. Répondre yes à la question
7. login as : ansible
8. Password: Password1!
9. Laisser cette fenêtre ouverte

LAB # 2b : PRÉPARATION

Configuration de Visual Studio Code

1. Toujours de votre instance Windows, Lancer Visual Studio Code
2. Cliquer sur la 4ième icône sur votre gauche (extension)
3. Rechercher l'extension : **sftp** (SFTP/FTP sync, de liximomo)
4. Cliquer sur **Install**
5. Rechercher l'extension: **Ansible** (Vscode extension for Ansible, de Microsoft)
6. Cliquer sur **Install**
7. Cliquer sur **Reload** pour recharger Visual Studio Code avec les nouvelles extensions
8. Cliquer sur la première icône sur votre gauche (explorer)
9. Cliquer sur **Open Folder** et aller chercher le dossier **c:\labs**
10. Cliquer sur le fichier **vscode.sftp.json** et copier le contenu avec ctrl+c
11. Appuyer sur les touches simultanément SHIFT+CTRL+P
12. Taper **sftp:config**
13. Remplacer le texte présent par le contenu du presse-papier avec ctrl + v
14. Enregistrer avec ctrl + s, puis fermer ce fichier en cliquant sur le X à côté du nom du fichier sur la barre du haut.

LAB # 2c : Validation

Validation de la configuration entre Visual Studio Code et Linux via SFTP

1. Créez un nouveau fichier dans **C:/labs** , par exemple **test.txt** , et sauvegardez-le
   a. Attention à l'endroit où vous le créer !
2. Dans le menu gauche (Explorer) , cliquer sur le fichier test.txt, puis "right-click" et sélectionner **"SFTP: Upload" ..** dans le bas à gauche vous devriez voir **upload done**
3. Sous votre session SSH avec Putty sur la machine centos, tapez **cd labs**
   a. Vérifier que le fichier existe, en faisant la commande **ls**

Si Visual Studio vous demande le "password" pour SFTP, vous avez un problème avec le dossier c:/labs/insecure qui contient la clé SSH. Re-valider l'étape précédente avec le plugin sftp.

LAB # 2d : Inventaire

Inventaire et validation des communications Ansible

Objectifs

À l'aide d'un éditeur sous Linux (vim ou nano) ***OU*** *Visual Studio Code*

1. Créer le fichier inventaire tel que décrit à la page 23 de ce document.
2. Enregistrer vos fichiers dans ~\labs\ (ou c:\labs avec VSE, n'oubliez pas d'enregistrer et d'uploader!)
3. Tester du poste Linux, la communication Ansible avec la commande:

```
[ansible@centosmick95 labs] ansible all -i inventory -m win_ping
windows1 | SUCCESS => {
      "changed": false,
       "ping": "pong"
}
```

LAB # 2e : Documentation - Optionnel

Documentation des modules ansible

Objectifs

Apprendre à utiliser les modules Ansible, avec la documentation en ligne ainsi qu'avec la ligne de commande

1. Visitez http://docs.ansible.com/ansible/list_of_all_modules.html
2. Dans le poste Linux, installez la documentation hors-ligne

```
[ansible@centosmick95 labs]$ sudo yum install ansible-doc -y
```

3. Regarder les modules dédiés à windows

```
[ansible@centosmick95 labs]$ ansible-doc -l | grep win_
```

4. Consulter l'aide pour le module que vous voulez:

```
[ansible@centosmick95 labs]$ ansible-doc win_ping
```

5. Vous trouverez aussi toute la documentation en HTML dans /usr/share/doc/ansible

PLAYBOOK ANSIBLE

YAML

1. Principalement conçu pour la représentation des structures de données
2. Facile à écrire, format pouvant être lu par les humains
3. Objectif de la conception : abandonner la syntaxe traditionnelle “cloisonée”

ÉVITEZ D’UTILISER LE COPIER-COLLER!!!

Programme de validation YAML : yamllint.com

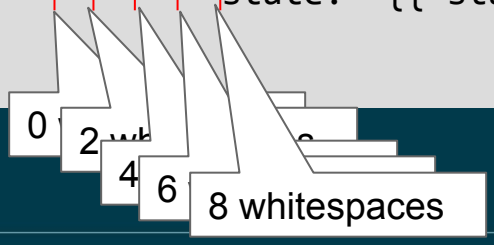
Nom du fichier

```
[ansible@centosmick95 labs] vim apache.yaml

ou

[ansible@centosmick95 labs] vim apache.yml
```

EXEMPLE DE PLAYBOOK

Attention à l'alignement à gauche: 2 espaces en blanc (pas de <Tab>!)

```
---
- name: This is a Play
  hosts: web
  become: yes
  gather_facts: no
  vars:
    state: present

  tasks:
    - name: Install Apache
      yum:
        name: httpd
        state: "{{ state }}"
```



PLAYS

Nommage

```
- name: This is a Play
```

PLAYS

Sélection des hôtes

```
- name: This is a Play
  hosts: web

- name: this is a play 2
  hosts: all
```

PLAYS

Arguments

```
- name: This is a Play
  hosts: web
  become: yes
  gather_facts: no
```

FAITS

Recueille les faits au sujet de l'hôte distant

- Ansible fournit automatiquement de nombreux faits au sujet du système
- Fournit par le module **setup**
- Si Facter (Puppet) ou Ohai (Chef) sont installés, les variables de ces programmes seront aussi copiés instantanément dans le fichier JSON pour être utilisés comme modèle
  - Ces variables sont préfixées avec facter_ et ohai_. Il est donc facile d'en connaître la source.
- Si vous utilisez les faits d'Ansible et que vous choisissez de ne pas installer Facter et Ohai, vous pouvez éviter les dépendances associées à Ruby

http://docs.ansible.com/ansible/setup_module.html

PLAYS

Variables et tâches

```
---
- name: This is a Play
  hosts: web
  become: yes
  gather_facts: no
  vars:
    state: present

  tasks:
    - name: Install Apache
      yum:
        name: httpd
        state: "{{ state }}"
```

**** Lorsqu'une variable est utilisée comme premier élément pour commencer une valeur, les guillemets sont obligatoires !!!

EXÉCUTER UN PLAYBOOK ANSIBLE

En mode vérification uniquement

```
[centos@centos7-1 ansible]$ ansible-playbook play.yml --syntax-check

[centos@centos7-1 ansible]$ ansible-playbook play.yml -i inventory --check
```

EXÉCUTER UN PLAYBOOK ANSIBLE

Lancer la commande

```
[centos@centos7-1 ansible]$ ansible-playbook play.yml -i inventory

# mode debug
[centos@centos7-1 ansible]$ ansible-playbook play.yml -i inventory -vvv
```

```
PLAY [Test playbook play]
**********************************************************************************

TASK [Envoi message play]
**********************************************************************************
changed: [centos1]

PLAY RECAP
**********************************************************************************
centos1                    : ok=1    changed=1    unreachable=0    failed=0
```

LAB # 3

Premier playbook

Objectifs

À l'aide de la ligne de commande et/ou Visual Studio Code

1. Basé sur l'exemple qui suit, créer votre premier playbook
2. Le playbook doit installer Firefox sur votre machine Windows.

   Indice : module **win_chocolatey**

   ```
   # ansible-doc -s win_chocolatey
   ```

Nous allons utiliser Firefox pour les prochains labs!

Documentation des modules Windows

http://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html

EXEMPLE DE PLAYBOOK (Linux)

```
[ansible@centosmick95 labs]$ vim message.yaml

---
- name: Envoi un message sur poste Windows
  hosts: all
  gather_facts: no

  tasks:
    - name: Envoi du message
      win_msg:
        msg: Test d'Ansible
        display_seconds: 10
```

LAB # 3 - Solution

```
---
- name: Installation Firefox avec Chocolatey
  hosts: all
  gather_facts: no

  tasks:
    - name: Installation de Firefox
      win_chocolatey:
        name: firefox
        state: present
```

```
[ansible@centosmick95 labs]$ ansible-playbook -i inventory firefox.yaml

PLAY [Installation Firefox avec Chocolatey]
*********************************************************************

TASK [Installation de Firefox]
*********************************************************************
changed: [windows1]

PLAY RECAP *********************************************************************
windows1                   : ok=1    changed=1    unreachable=0    failed=0
```

ANSIBLE WINDOWS

Qu'est ce que Ansible fait pour Windows ?

Avec le support Windows **natif** d'Ansible, vous pouvez :

- Récupérer les faits des machines Windows
- Installer et désinstaller des MSIs
- Activer et désactiver les fonctionnalités Windows
- Démarrer, arrêter, et gérer des services Windows
- Créer et gérer des usagers et des groups locaux ou AD
- Gérer des paquetages Windows via Chocolatey package manager
- Gérer et installer des mises à jour Windows
- Récupérer des fichiers d'un site distant
- Pousser et exécuter vos scripts PowerShell

Historique

Nombre de modules Windows dans Ansible

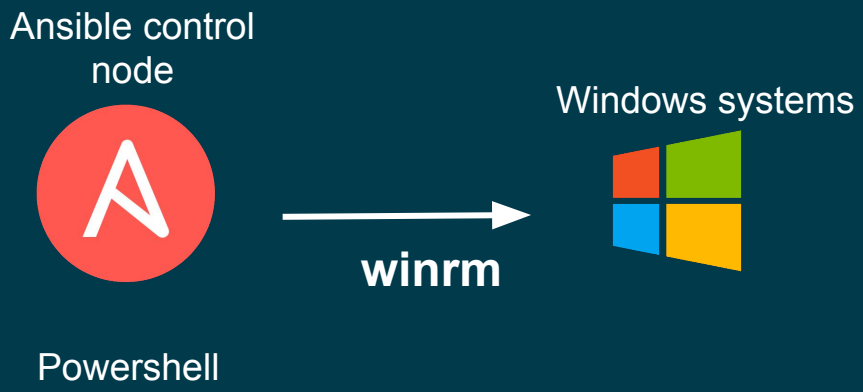
- Modules Ansible pour Windows
  - V 1.8 : 10
  - V 1.9 : 14
  - V 2.0 : 30
  - V 2.1 : 37
  - V 2.2 : 42
  - V 2.3 : 54
  - V 2.4 : 74
  - V 2.5 : 81
  - V 2.6 (alpha) : ~84

4 Playbooks et modules Windows

win_acl – Set file/directory/registry permissions for a system user or group
win_acl_inheritance – Change ACL inheritance
win_audit_policy_system – Used to make changes to the system wide Audit Policy.
win_audit_rule – Adds an audit rule to files, folders, or registry keys
win_certificate_store – Manages the certificate store
win_chocolatey – Manage packages using chocolatey
win_command – Executes a command on a remote Windows node
win_copy – Copies files to remote locations on windows hosts
win_defrag – Consolidate fragmented files on local volumes
win_disk_facts – Show the attached disks and disk information of the target host
win_disk_image – Manage ISO/VHD/VHDX mounts on Windows hosts
win_dns_client – Configures DNS lookup on Windows hosts
win_domain – Ensures the existence of a Windows domain.
win_domain_controller – Manage domain controller/member server state for a Windows host
win_domain_group – creates, modifies or removes domain groups
win_domain_membership – Manage domain/workgroup membership for a Windows host
win_domain_user – Manages Windows Active Directory user accounts
win_dotnet_ngen – Runs ngen to recompile DLLs after .NET updates
win_dsc – Invokes a PowerShell DSC configuration
win_environment – Modify environment variables on windows hosts
win_eventlog – Manage Windows event logs
win_eventlog_entry – Write entries to Windows event logs
win_feature – Installs and uninstalls Windows Features on Windows Server
win_file – Creates, touches or removes files or directories.
win_file_version – Get DLL or EXE file build version
win_find – Return a list of files based on specific criteria
win_firewall – Enable or disable the Windows Firewall
win_firewall_rule – Windows firewall automation
win_get_url – Fetches a file from a given URL
win_group – Add and remove local groups
win_group_membership – Manage Windows local group membership
win_hotfix – install and uninstalls Windows hotfixes
win_iis_virtualdirectory – Configures a virtual directory in IIS.
win_iis_webapplication – Configures IIS web applications
win_iis_webapppool – configures an IIS Web Application Pool
win_iis_webbinding – Configures a IIS Web site binding.
win_iis_website – Configures a IIS Web site.
win_lineinfile – Ensure a particular line is in a file, or replace an existing line using a expre...
win_mapped_drive – maps a network drive for a user
win_msg – Sends a message to logged in users on Windows hosts.
win_msi **(D)** – Installs and uninstalls Windows MSI files
win_nssm – NSSM - the Non-Sucking Service Manager
win_owner – Set owner
win_package – Installs/uninstalls an installable package
win_pagefile – Query or change pagefile configuration
win_path – Manage Windows path environment variables
win_ping – A windows version of the classic ping module
win_power_plan – Changes the power plan of a Windows system
win_product_facts – Provides Windows product information (product id, product key)
win_psexec – Runs commands (remotely) as another (privileged) user
win_psmodule – Adds or removes a Powershell Module.
win_rabbitmq_plugin – Manage RabbitMQ plugins
win_reboot – Reboot a windows machine
win_reg_stat – returns information about a Windows registry key or property of a key
win_regedit – Add, change, or remove registry keys and values
win_region – Set the region and format settings
win_regmerge – Merges the contents of a registry file into the windows registry
win_robocopy – Synchronizes the contents of two directories using Robocopy
win_route – Add or remove a static route.
win_say – Text to speech module for Windows to speak messages and optionally play sounds
win_scheduled_task – Manage scheduled tasks
win_scheduled_task_stat – Returns information about a Windows Scheduled Task
win_security_policy – changes local security policy settings
win_service – Manage and query Windows services
win_share – Manage Windows shares
win_shell – Execute shell commands on target hosts.
win_shortcut – Manage shortcuts on Windows
win_stat – returns information about a Windows file
win_tempfile – Creates temporary files and directories.
win_template – Templates a file out to a remote server.
win_timezone – Sets Windows machine timezone
win_toast – Sends Toast windows notification to logged in users on Windows 10 or late hosts
win_unzip – Unzips compressed files and archives on the Windows node
win_updates – Download and install Windows updates
win_uri – Interacts with webservices
win_user – Manages local Windows user accounts
win_user_right – Manage Windows User Rights
win_wait_for – Waits for a condition before continuing
win_wakeonlan – Send a magic Wake-on-LAN (WoL) broadcast packet
win_webpicmd – Installs packages using Web Platform Installer command-line
win_whoami – Returns information about the current user and process

Comment Ansible travaille avec Windows ?

Les modules Ansible pour Windows sont écrits en powershell et exécuter au travers winrm (Windows Remote Management)



Méthodes d'authentification

https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html

Option	Local Account	Active directory Account	Credential Delegation
Basic	Yes	No	No
Certificate	Yes	No	No
Kerberos	No	Yes	Yes
NTLM	Yes	Yes	no
CredSSP	Yes	Yes	Yes

Autres options :
- OpenSSH pour Windows (https://github.com/PowerShell/Win32-OpenSSH)
- pywinrm secure sans certificat SSL (> pywinrm 0.3.0)

Prérequis WINDOWS (BASIC AUTH)

- Windows 7 sp1 ou Windows 2008 sp1 +
- Powershell 3 (mais 5 est requis pour certains modules) https://github.com/jborean93/ansible-windows/blob/master/scripts/Upgrade-PowerShell.ps1
- Configurer le mode d'authentification
  - Exemple : Windows 2016
    - Winrm est présent mais non configuré
      - winrm set winrm/config/service/auth @{Basic="true"}
      - winrm set winrm/config/service @{AllowUnencrypted="true"}
    - Ouvrir le port 5985 du pare-feu

Mise à jour et redémarre si nécessaire

```
---
- name: Update
  hosts: all
  tasks:
    - name: update windows
      win_updates:
      register: update_result

    - debug: var=update_result

    - name : reboot if required
      win_reboot:
      when: update_result.reboot_required
```

Créer un usager local

```
---
- name: Create a user
  hosts: all
  tasks:
    - name: Ensure user bob is present
      win_user:
        name: bob
        password: B0bP4ssw0rd
        state: present
        groups:
          - Users
```

ipconfig

```
---
- name: ipconfig
  hosts: windows
  tasks:
    - name: run ipconfig
      win_command: ipconfig
      register: ipconfig

    - debug: var=ipconfig
```

stat

```
---
- name: Validate presence of win.ini
  hosts: windows
  tasks:
    - name: test stat module on file
      win_stat: path="C:/Windows/win.ini"
      register: stat_file

    - debug: var=stat_file

    - name: check stat_file result
      assert:
          that:
             - "stat_file.stat.exists"
             - "not stat_file.stat.isdir"
             - "stat_file.stat.size > 0"
             - "stat_file.stat.md5"
```

become

```
---
- name: Disable Zune Music and Zune Video appx
    win_shell: |
      Get-AppxPackage -name "Microsoft.ZuneMusic" | Remove-AppxPackage
      Get-AppxPackage -name "Microsoft.ZuneVideo" | Remove-AppxPackage
    become: yes
    become_user: Administrator
```

D’autres exemples de playbooks

Dag Wieers

https://github.com/crombeen/ansible

PLAYBOOK AVANCÉ AVEC ANSIBLE

PLAYS

Boucles

```
- name: This is a Play
  hosts: web
  become: yes
  gather_facts: no
  vars:
    state: present

  tasks:
    - name: Install Apache and PHP
      yum: name={{ item }} state={{ state }}
      with_items:
        - httpd
        - php
```

BOUCLES

Plusieurs types de boucles générales et à usage déterminé

- **with_nested**
- **with_dict**
- **with_fileglob**
- **with_together**
- **with_sequence**
- **until**
- **with_random_choice**
- **with_first_found**
- **with_indexed_items**
- **with_lines**

http://docs.ansible.com/ansible/playbooks_loops.html

HANDLERS

Exécuter seulement si la tâche a un statut « modifié »

```
---
- name: This is a Play
  hosts: web

  tasks:
    - yum: name={{ item }} state=installed
      with_items:
        - httpd
        - memcached
      notify: Restart Apache

    - template: src=templates/web.conf.j2
dest=/etc/httpd/conf.d/web.conf
      notify: Restart Apache

  handlers:
    - name: Restart Apache
      service: name=httpd state=restarted
```

TAG

Exemple d'utilisation d'un tag

```
tasks:

    - yum: name={{ item }} state=installed
      with_items:
         - httpd
         - memcached
      tags:
         - packages

    - template: src=templates/src.j2 dest=/etc/foo.conf
      tags:
         - configuration
```

TAGS

Exécuter avec des tags

```
ansible-playbook example.yml --tags “configuration”

ansible-playbook example.yml --skip-tags "notification"
```

RESULT

Enregistre les résultats de la tâche pour le débogage ou d'autres fins

```
# Example setting the Apache version
- shell: httpd -v|grep version|awk '{print $3}'|cut -f2 -d'/'
  register: result

- debug: var=result    (va afficher le résultat)
```

TÂCHES CONDITIONNELLES

Seulement exécuter sur la machine dont le système d'exploitation est Red Hat

```
- name: This is a Play
  hosts: web
  remote_user: centos
  become: sudo

  tasks:
    - name: install Apache
      yum: name=httpd state=installed
      when: ansible_os_family == "RedHat"
```

BLOCS

Applique une condition à plusieurs tâches à la fois

```
tasks:

  - block:
    - yum: name={{ item }} state=installed
      with_items:
        - httpd
        - memcached
    - template: src=templates/web.conf.j2 dest=/etc/httpd/conf.d/web.conf
    - service: name=bar state=started enabled=True
    when: ansible_distribution == 'CentOS'
```

ERREURS

Ignore les erreurs

Par défaut, Ansible s'arrête aux erreurs. Ajoutez le paramètre **ignore_error** pour sauter les erreurs possibles.

```
- name: ping host
  command: ping -c1 www.foobar.com
  ignore_errors: yes
```

ERREURS

Gérer les erreurs à l'aide des blocs

```
tasks:

 - block:
     - debug: msg='i execute normally'
     - command: /bin/false
     - debug: msg='i never execute, cause ERROR!'
   rescue:
     - debug: msg='I caught an error'
     - command: /bin/false
     - debug: msg='I also never execute :-('
   always:
     - debug: msg="this always executes"
```

LINEINFILE

Pour ajouter, enlever ou mettre à jour une ligne en particulier

```
-   lineinfile: dest=/etc/selinux/config regexp=^SELINUX=
    line=SELINUX=enforcing

-   lineinfile: dest=/etc/httpd/conf/httpd.conf regexp="^Listen "
    insertafter="^#Listen " line="Listen 8080"
```

Vous trouverez ci-dessous un très bon exemple :
https://relativkreativ.at/articles/how-to-use-ansibles-lineinfile-module-in-a-bulletproof-way

Remarque : L'utilisation d'un template ou d'un module dédié est plus efficace

LAB #4: Installer IIS

Utilisez les options avancées d'Ansible pour installer et configurer un serveur web

Écrivez un playbook ansible qui va :

1. Installer IIS sur votre serveur Windows (module win_feature)
2. Copier un fichier index.html « Hello World » à la racine par défaut de votre serveur IIS (module: à vous de le découvrir). Vous devez créer d'abord le fichier index.html sur votre poste linux.
3. Créer un second site web qui écoutera sur le port 8080 (module win_iis_website) . Vous devez créer d'abord un autre dossier IIS différent au premier ainsi qu'un autre fichier HTML.
4. Ouvrir le port 8080 sur le pare-feu windows (module: à vous de le découvrir)
5. Essayer d'utiliser des variables

Validez le tout en visitant http://localhost et http://localhost:8080 dans la VM Windows

LAB #3 - SOLUTION

```
---
- name: Configuration d'un serveur web
  hosts: all
  gather_facts: no
  vars:
    ansible_site_path: "C:\\inetpub\\wwwroot\\ansibletest"
    default_iis_path: "C:\\inetpub\\wwwroot\\index.html"
  tasks:
    - name: Installation de IIS
      win_feature:
        name: Web-Server
        state: present
    - name: Copie ficher index.html
      win_copy:
        src: index.html
        dest: "{{ default_iis_path }}"
    - name: Creation du repertoire pour un second site web
      win_file:
        path: "{{ ansible_site_path }}"
        state: directory
    - name: Creation du second site web
      win_iis_website:
        name: "Ansible test site"
        state: started
        port: 8080
        physical_path: "{{ ansible_site_path }}"
    - name: Create fichier index du second site web
      win_copy:
        src: ansible.html
        dest: "{{ ansible_site_path }}\\index.html"
    - name: Ouvrir le port firewall pour second site web
      win_firewall_rule:
        name: Ansible8080
        enable: yes
        state: present
        localport: 8080
        action: Allow
        direction: In
        protocol: Tcp
```

ANSIBLE EN PROFONDEUR: VARIABLES
ET
GESTION DE LA CONFIGURATION

redhat.

PRIORITÉ DES VARIABLES

Ansible v2

1. role defaults
2. inventory file or script group vars
3. inventory group_vars/all
4. playbook group_vars/all
5. inventory group_vars/*
6. playbook group_vars/*
7. inventory file or script host vars
8. inventory host_vars/*
9. playbook host_vars/*
10. host facts
11. play vars
12. play vars_prompt
13. play vars_files
14. role vars (defined in role/vars/main.yml)
15. block vars (only for tasks in block)
16. task vars (only for the task)
17. role (and include_role) params
18. include params
19. include_vars
20. set_facts / registered vars
21. extra vars (always win precedence)

VARIABLES MAGIQUES

Ansible génère et maintient l'information au sujet de son état actuel et des autres hôtes par le biais d'une série de variables « magiques ».

★ **hostvars[inventory_hostname]**

Montre tous les faits Ansible
Variable spécifique pour hôte spécifique

{{ **hostvars['test.example.com']['ansible_distribution']** }}

★ **group_names**

est une liste (tableau) de tous les groupes dont fait partie l'hôte actuel

★ **groups**

est une liste de tous les groupes (et hôtes) de l'inventaire.

VARIABLES MAGIQUES

Utiliser le module debug pour consulter le contenu

```
- name: debug
  hosts: all

  tasks:
    - name: Show hostvars[inventory_hostname]
      debug: var=hostvars[inventory_hostname]

    - name: Show ansible_ssh_host variable in hostvars
      debug: var=hostvars[inventory_hostname].ansible_ssh_host

    - name: Show group_names
      debug: var=group_names

    - name: Show groups
      debug: var=groups
```

```
ansible-playbook -i invedntory --limit <hostname> debug.yml
```

UTILISATION DES VARIABLES YAML

Les valeurs YAML commençant par une variable doivent être écrites entre guillemets

```
vars:
  var1: {{ foo }} <<< ERROR!
  var2: “{{ bar }}”
  var3: Echoing {{ foo }} here is fine
```

Module template

À l'aide de Jinja2

Les modèles vous permettent de créer des fichiers de configuration dynamiques à l'aide des variables.

```
- win_template: src=test.j2 dest='C:\sites\playbooktest2\test.html'
```

Documentation :
http://docs.ansible.com/ansible/latest/modules/win_template_module.html

JINJA2

Délimiteurs

Jinja2 est un langage moderne et convivial, facilitant l'élaboration de modèles pour Python, qui s'inspire des modèles Django et il est utilisé par Ansible.

Il est fortement recommandé de lire au sujet de Jinja2 pour comprendre comment sont construits les templates.

```
{{ variable }}

{% for server in groups.webservers %}
```

JINJA2

BOUCLES

```
{% for server in groups.web %}
{{ server }}  {{ hostvars[server].ansible_default_ipv4.address }}
{% endfor %}
```

```
web1 10.0.1.1
web2 10.0.1.2
web3 10.0.1.3
```

JINJA2

Conditionnel

```
{% if ansible_processor_cores >= 2 %}
-smp enable
{% else %}
-smp disable
{% endif %}
```

JINJA2

Filtres de variables

```
{% set my_var='this-is-a-test' %}
{{ my_var | replace('-', '_') }}
```

```
this_is_a_test
```

JINJA2

Filtres de variables

```
{% set servers = "server1,server2,server3" %}
{% for server in servers.split(",") %}
{{ server }}
{% endfor %}
```

```
server1
server2
server3
```

JINJA2, d'autres filtres

Beaucoup d'options...

```
# Combine two lists
{{ list1 | union(list2) }}

# Get a random number
{{ 59 | random }} * * * * root /script/from/cron

# md5sum of a filename
{{ filename | md5 }}

# Comparisons
{{ ansible_distribution_version | version_compare('12.04', '>=') }}

# Default if undefined
{{ user_input | default('Hello World') }}
```

JINJA2

Tests

```
{% if variable is defined %}

{% if variable is none %}

{% if variable is even %}

{% if variable is string %}

{% if variable is sequence %}
```

Jinja2

Commentaires sur les modèles

```
{% for host in groups['app_servers'] %}
   {# ce commentaire ne serra pas affiché #}
   {{ loop.index }} {{ host }}
{% endfor %}
```

RÔLES D'ANSIBLE

RÔLES

Une collection redistribuable et réutilisable de :

- tâches
- fichiers
- scripts
- template
- variables

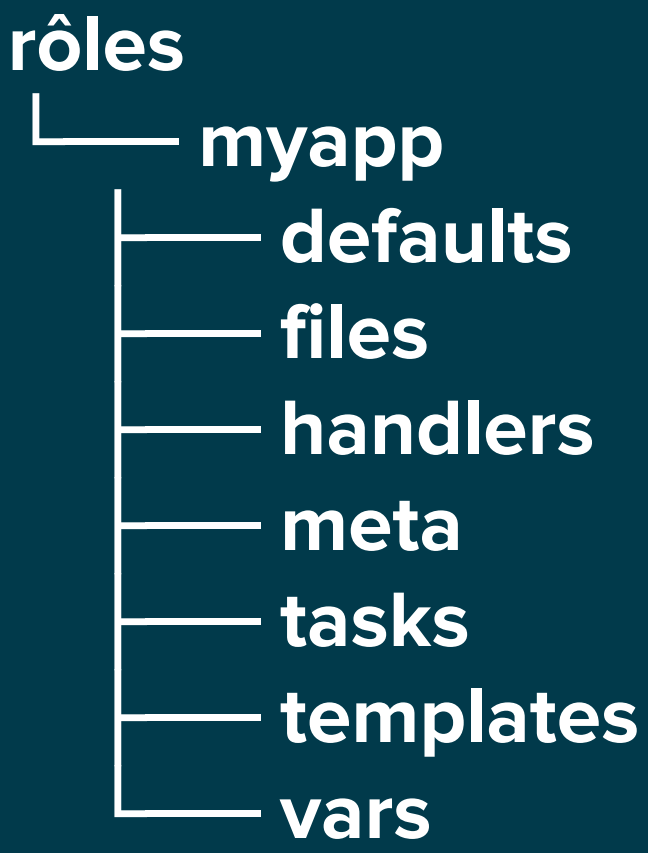
RÔLES

Souvent utilisés pour installer et configurer les services

- ➔ **installer les paquetages**
- ➔ **copier les fichiers**
- ➔ **exécuter les programmes démons**

RÔLES

Structure de répertoire



RÔLES

Pour créer automatiquement une structure de fichiers

```
ansible-galaxy init <role_name>
```

RÔLES

Exemples de playbook

```
---
- hosts: webservers
  roles:
     - common
     - webservers
```

RÔLES

Exemples de playbook

```
---
- hosts: webservers
  roles:
     - common
     - { role: myapp, dir: '/opt/a',  port: 5000 }
     - { role: myapp, dir: '/opt/b',  port: 5001 }
```

RÔLES

Exemples de playbook

```
---
- hosts: webservers
  roles:
    - { role: foo, when: "ansible_os_family == 'Windows'" }
```

RÔLES

Pré et post - exemple “rolling upgrade”

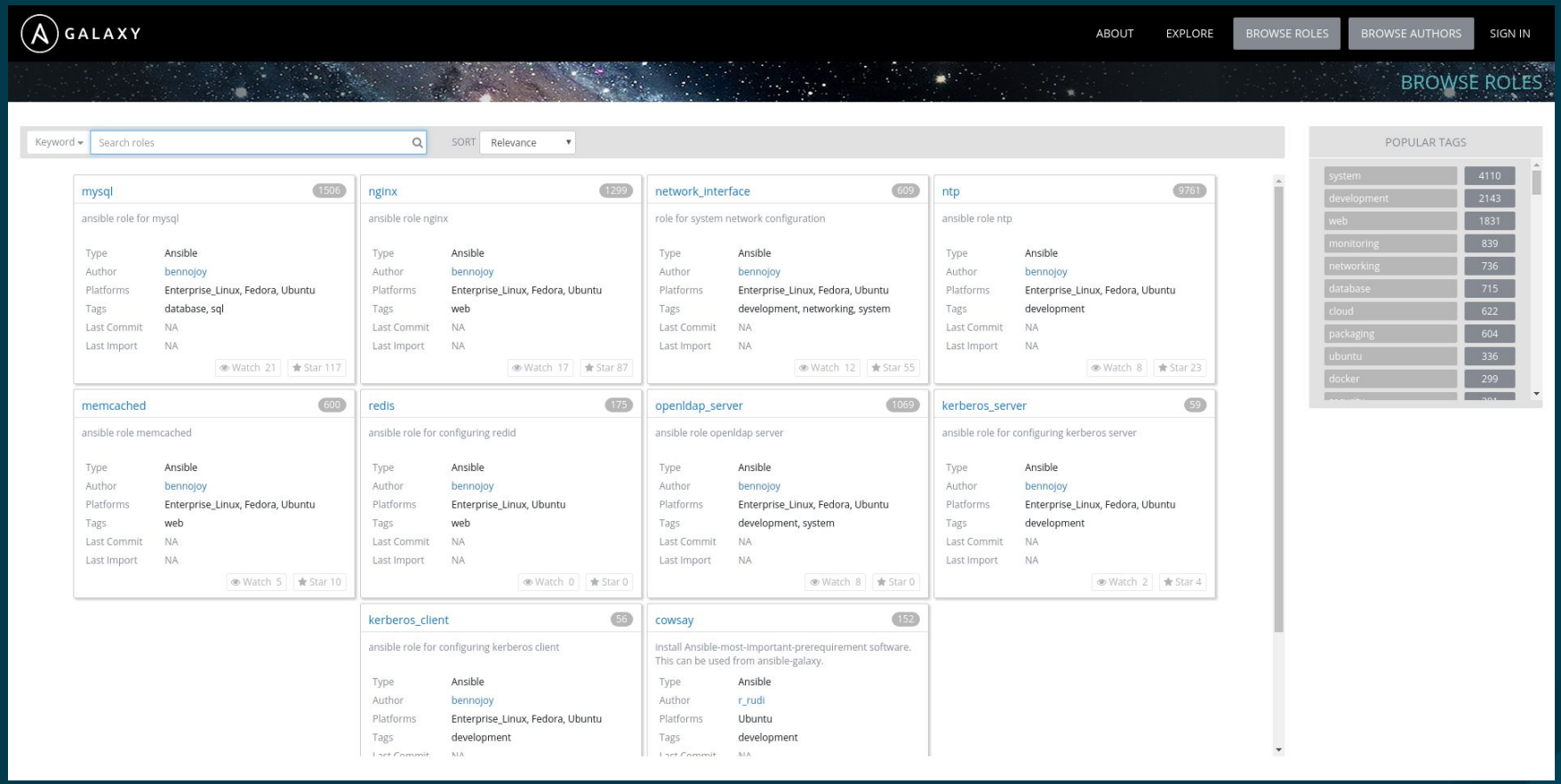
```
---
- hosts: webservers
  serial: 1

  pre_tasks:
    - command:lb_rm.sh {{ inventory_hostname }}
      delegate_to: lb

   - command: mon_rm.sh {{ inventory_hostname }}
      delegate_to: nagios

  roles:
    - myapp

  post_tasks:
    - command: mon_add.sh {{ inventory_hostname }}
      delegate_to: nagios

     - command: lb_add.sh {{ inventory_hostname }}
       delegate_to: lb
```


http://docs.ansible.com/ansible/playbooks_delegation.html

Ansible Galaxy

http://galaxy.ansible.com

Ansible Galaxy

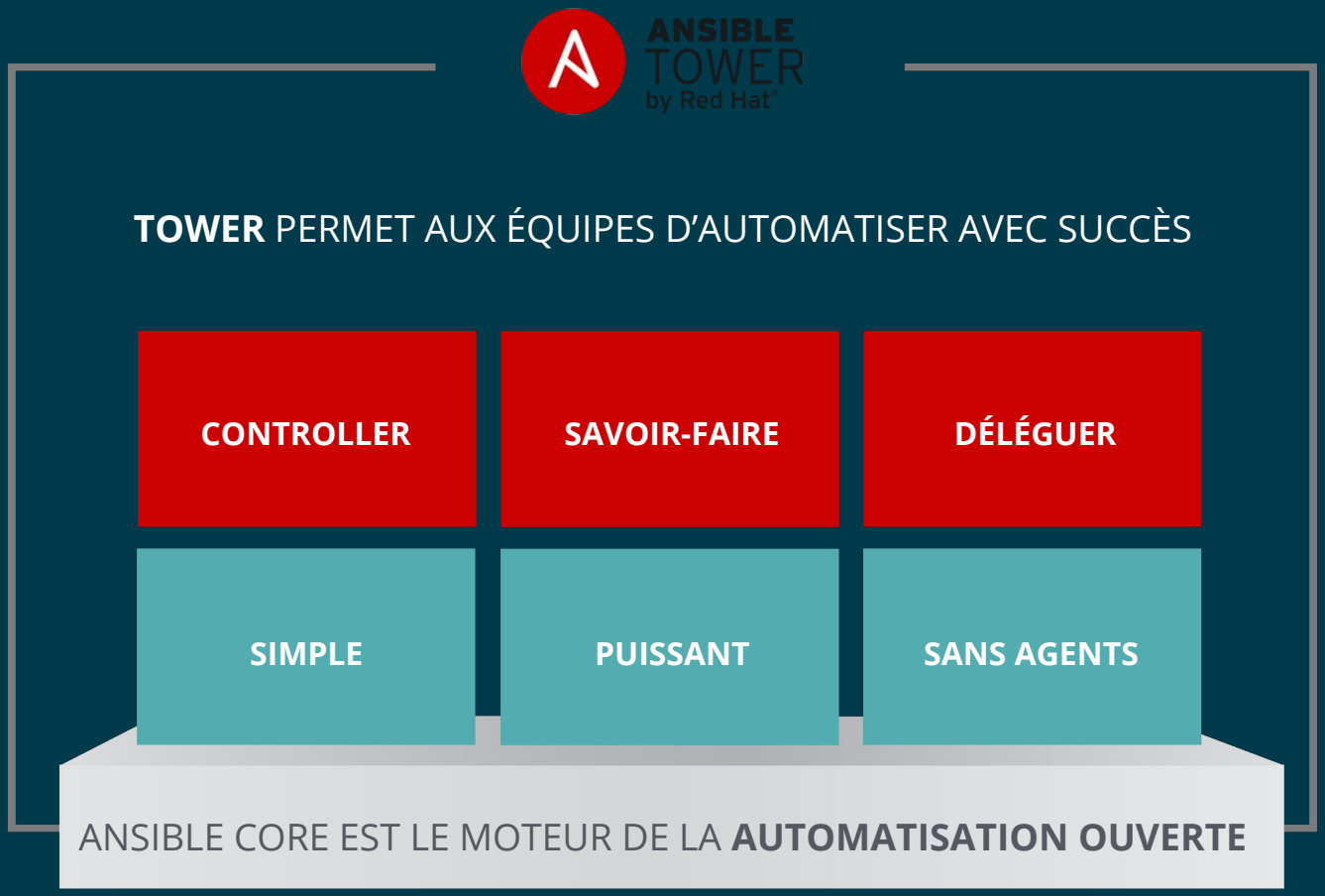


Ansible Galaxy

```
# ansible-galaxy search 'iis'
Found 4 roles matching your search:
Name...

# ansible-galaxy install davidkarban.git -p roles

# ansible-galaxy list -p roles

# ansible-galaxy remove -p roles
```

Ansible Tower

Gestion de Ansible pour l'entreprise



Ansible Tower

Quelles sont les valeurs ajoutées?

- ➔ **Gestion centralisée des authentifications**
- ➔ **Contrôle de l'accès en fonction du rôle**
- ➔ **Intégration de Satellite et de Cloudforms**
- ➔ **Déploiement par bouton**
- ➔ **Journalisation et déploiement centralisés**
- ➔ **Notification centralisée (Slack, Twilio, etc..)**
- ➔ **System tracking**
- ➔ **Workflow, API et plus ...**

LAB #5 - Exercice guidée
Introduction à Ansible Tower

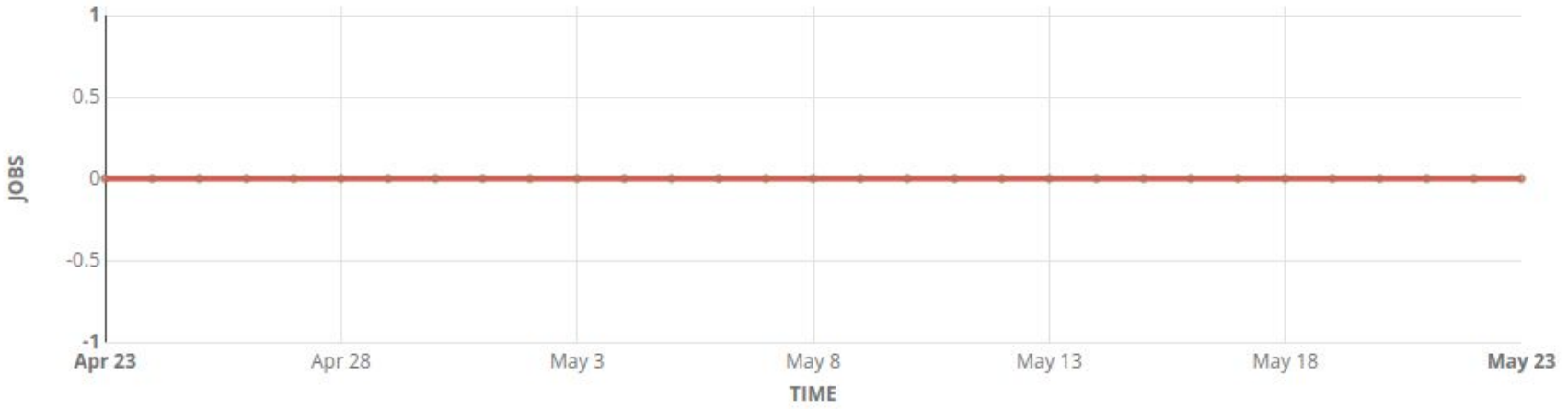
TOWER PROJECTS INVENTORIES TEMPLATES JOBS admin

DASHBOARD

1	0	1	0	1	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

JOB STATUS

PERIOD PAST MONTH JOB TYPE ALL VIEW ALL

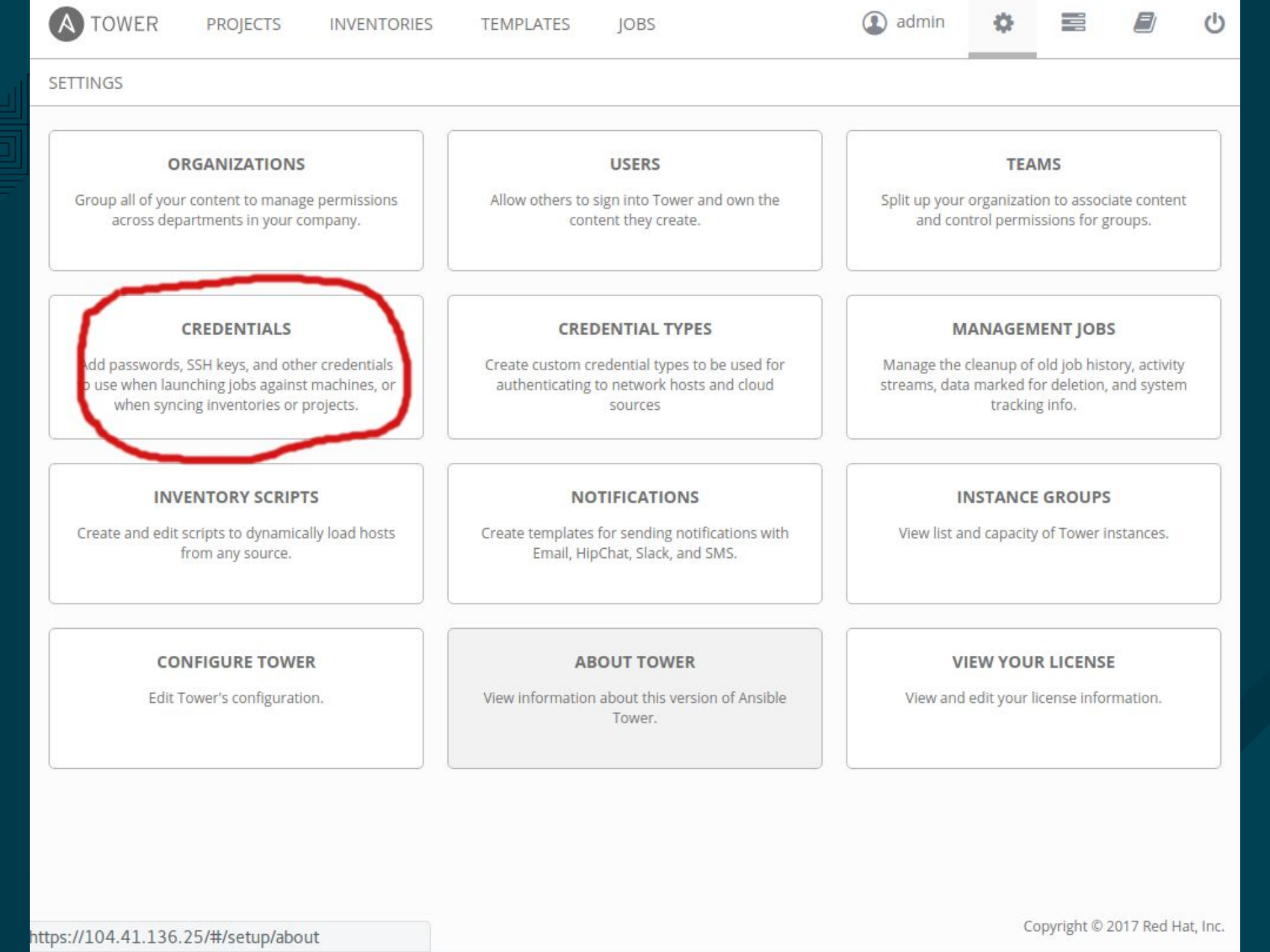


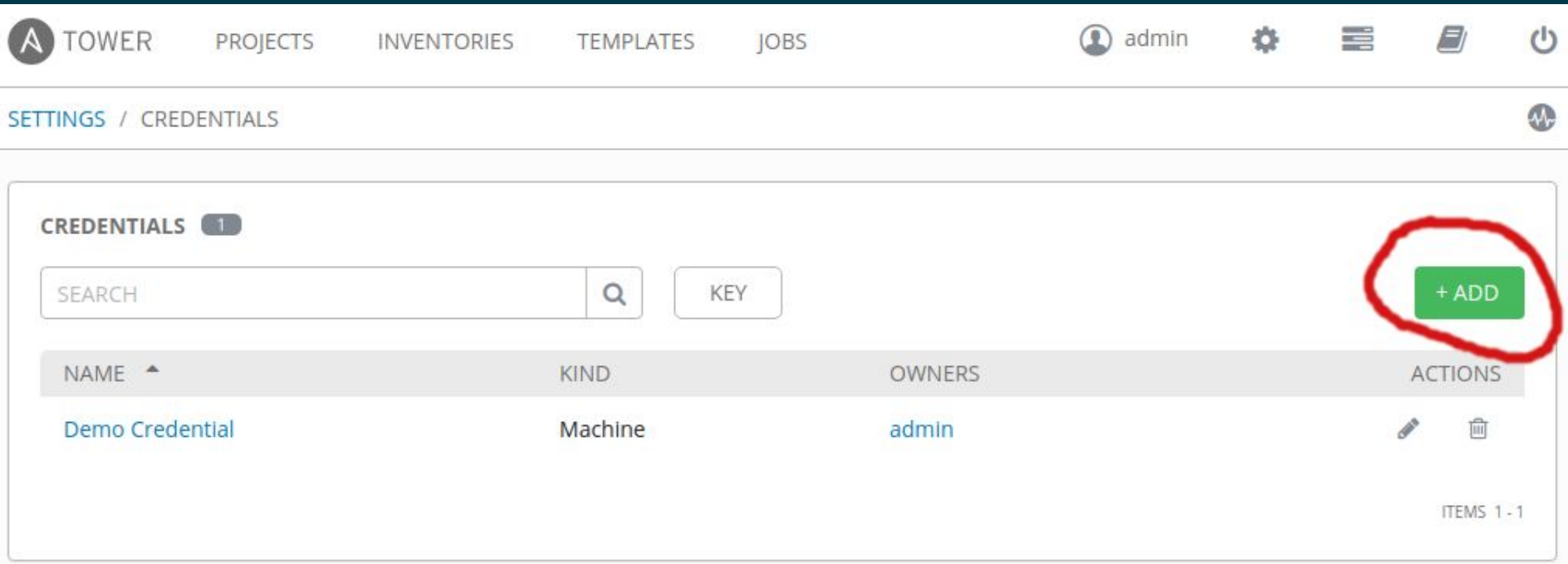
RECENTLY USED JOB TEMPLATES

No job templates were recently used.
You can create a job template here.

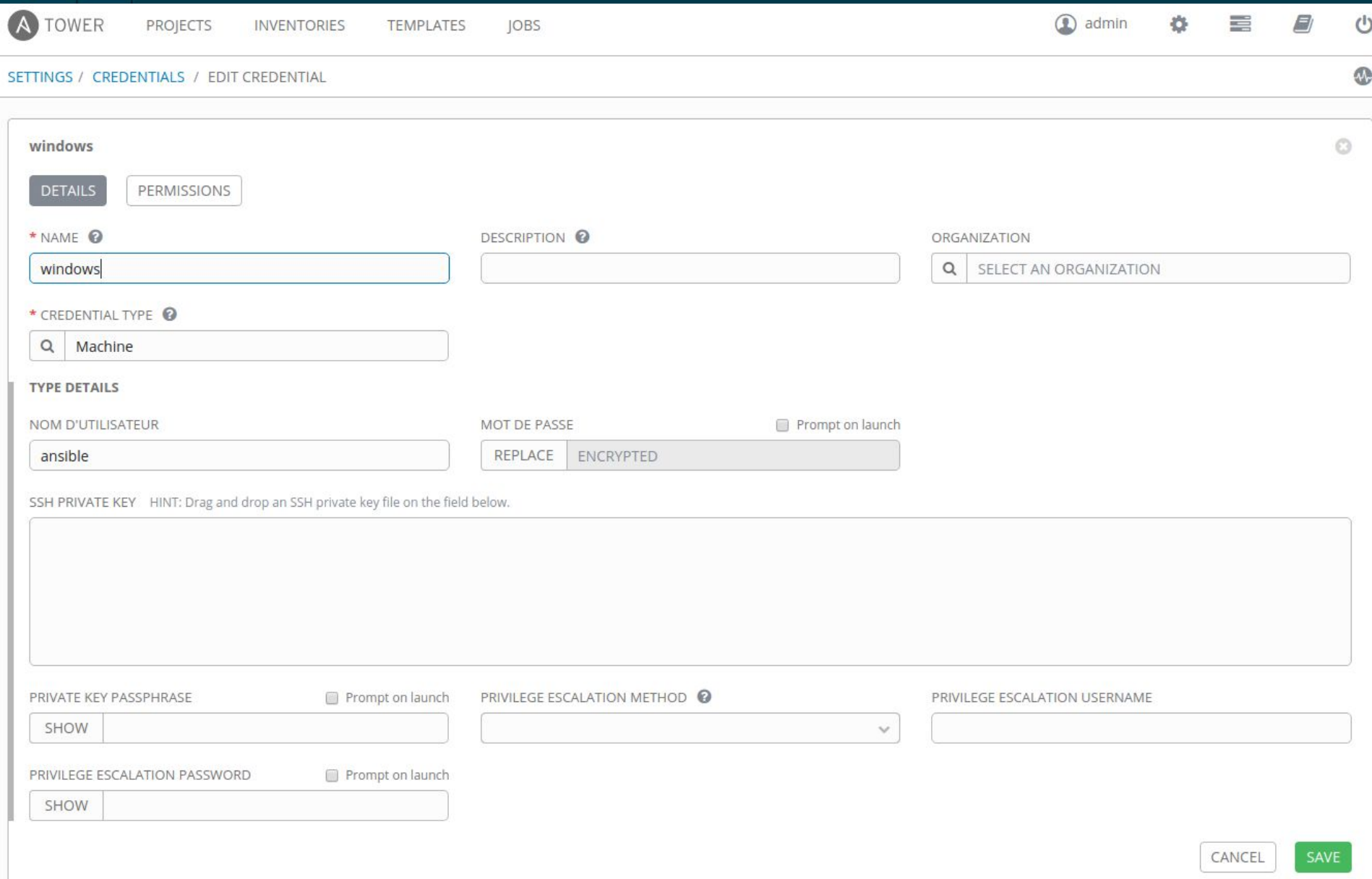
RECENTLY RUN JOBS

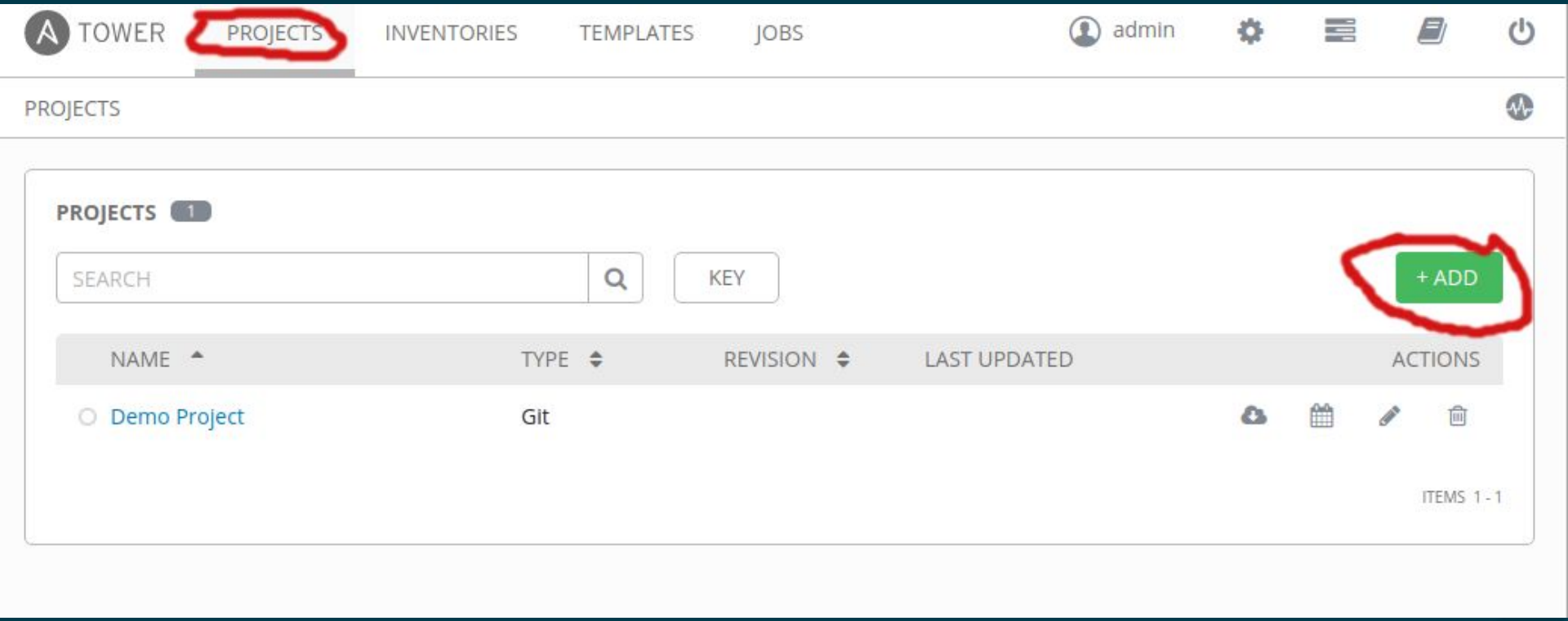
No jobs were recently run.

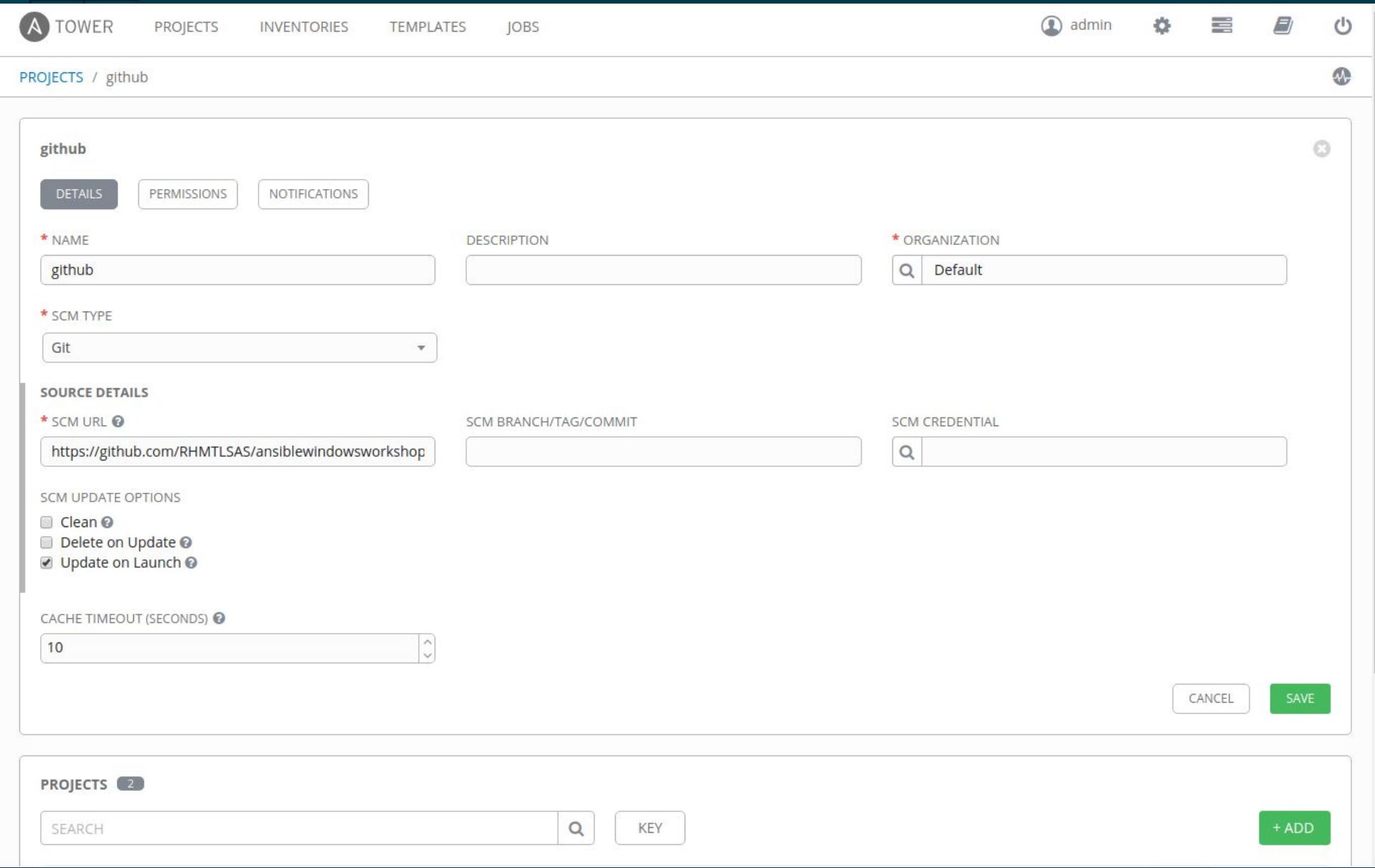

TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
SETTINGS
ORGANIZATIONS
Group all of your content to manage permissions across departments in your company.
USERS
Allow others to sign into Tower and own the content they create.
TEAMS
Split up your organization to associate content and control permissions for groups.
CREDENTIALS
dd passwords, SSH keys, and other credentials o use when launching jobs against machines, or when syncing inventories or projects.
CREDENTIAL TYPES
Create custom credential types to be used for authenticating to network hosts and cloud sources
MANAGEMENT JOBS
Manage the cleanup of old job history, activity streams, data marked for deletion, and system tracking info.
INVENTORY SCRIPTS
Create and edit scripts to dynamically load hosts from any source.
NOTIFICATIONS
Create templates for sending notifications with Email, HipChat, Slack, and SMS.
INSTANCE GROUPS
View list and capacity of Tower instances.
CONFIGURE TOWER
Edit Tower's configuration.
ABOUT TOWER
View information about this version of Ansible Tower.
VIEW YOUR LICENSE
View and edit your license information.
https://104.41.136.25/#/setup/about
Copyright © 2017 Red Hat, Inc.

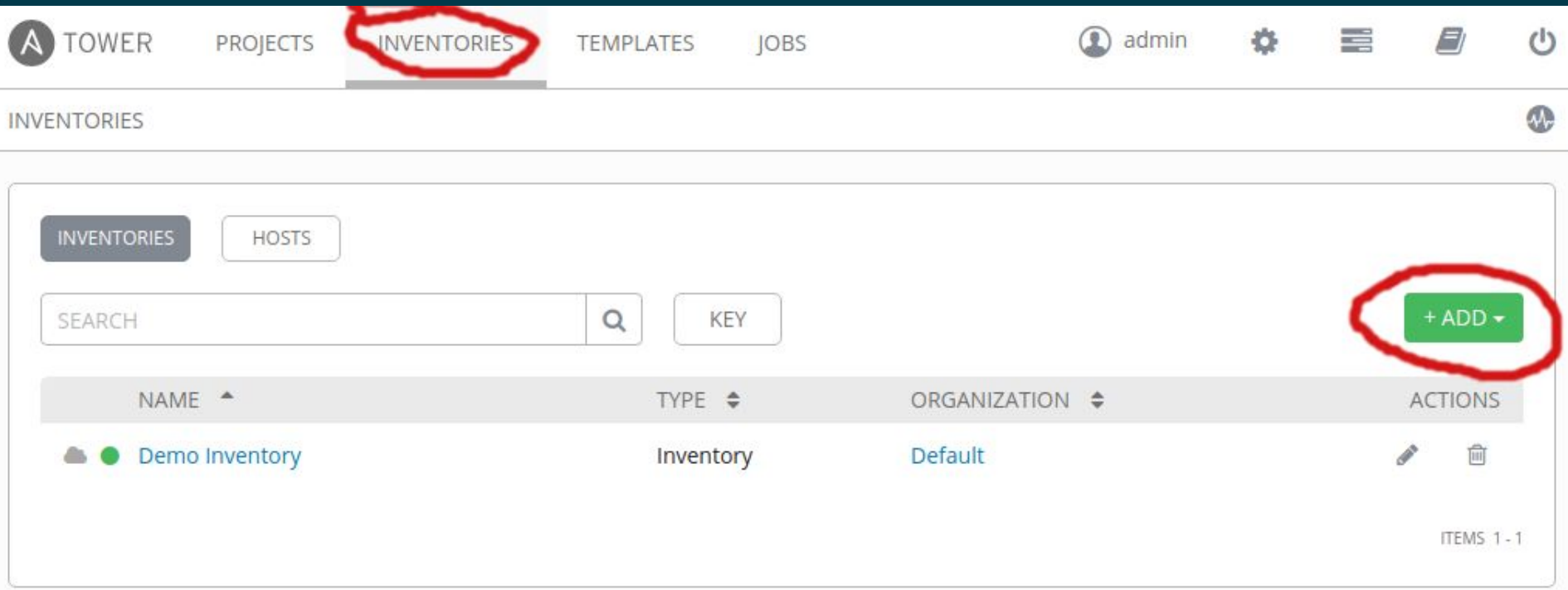

TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
SETTINGS / CREDENTIALS
CREDENTIALS 1
SEARCH
KEY
+ ADD
NAME
KIND
OWNERS
ACTIONS
Demo Credential
Machine
admin
ITEMS 1 - 1

redhat.

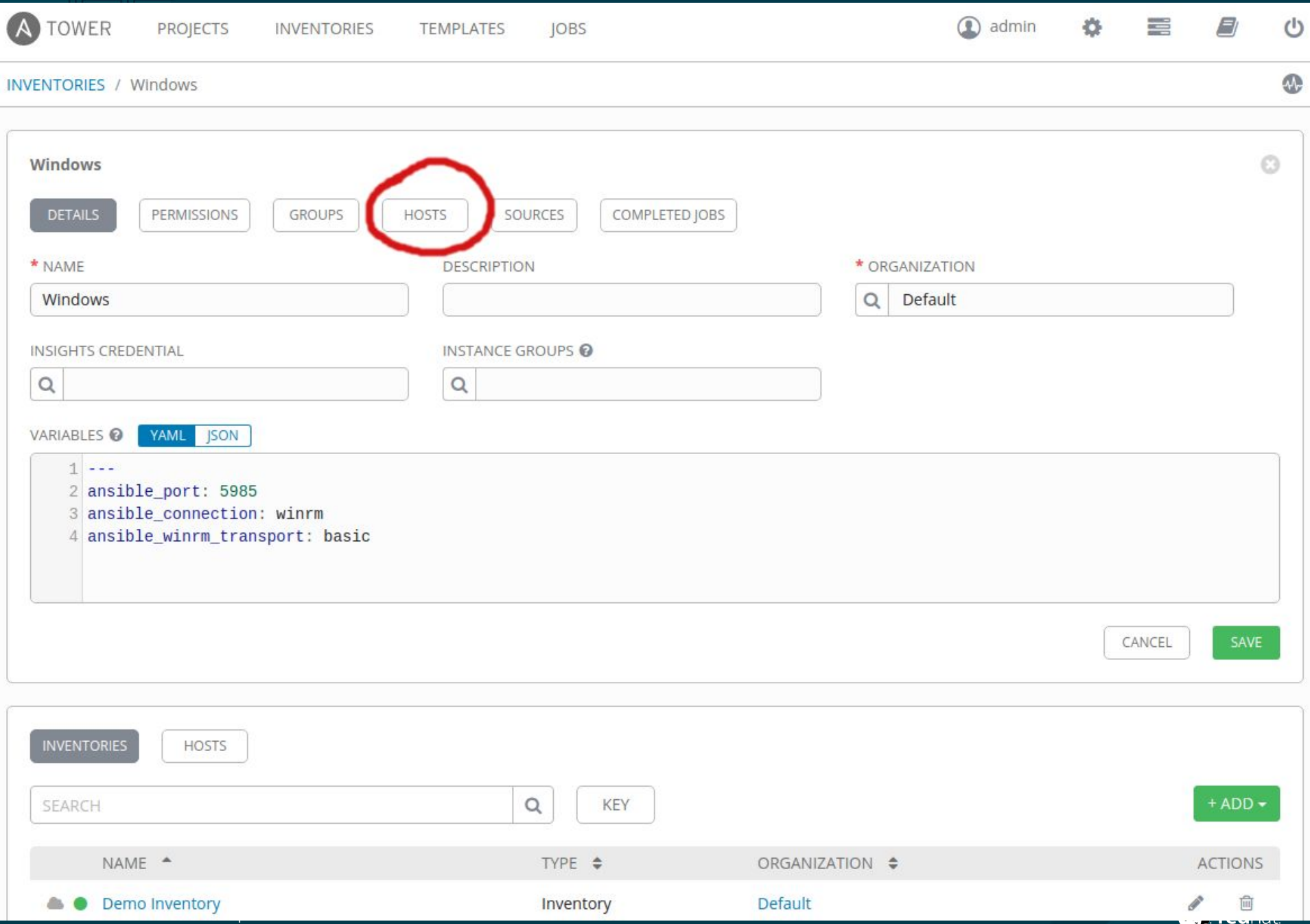

TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
SETTINGS / CREDENTIALS / EDIT CREDENTIAL
windows
DETAILS
PERMISSIONS
* NAME
windows
DESCRIPTION
ORGANIZATION
SELECT AN ORGANIZATION
* CREDENTIAL TYPE
Machine
TYPE DETAILS
NOM D'UTILISATEUR
ansible
MOT DE PASSE
Prompt on launch
REPLACE
ENCRYPTED
SSH PRIVATE KEY
HINT: Drag and drop an SSH private key file on the field below.
PRIVATE KEY PASSPHRASE
Prompt on launch
SHOW
PRIVILEGE ESCALATION METHOD
PRIVILEGE ESCALATION USERNAME
PRIVILEGE ESCALATION PASSWORD
Prompt on launch
SHOW
CANCEL
SAVE

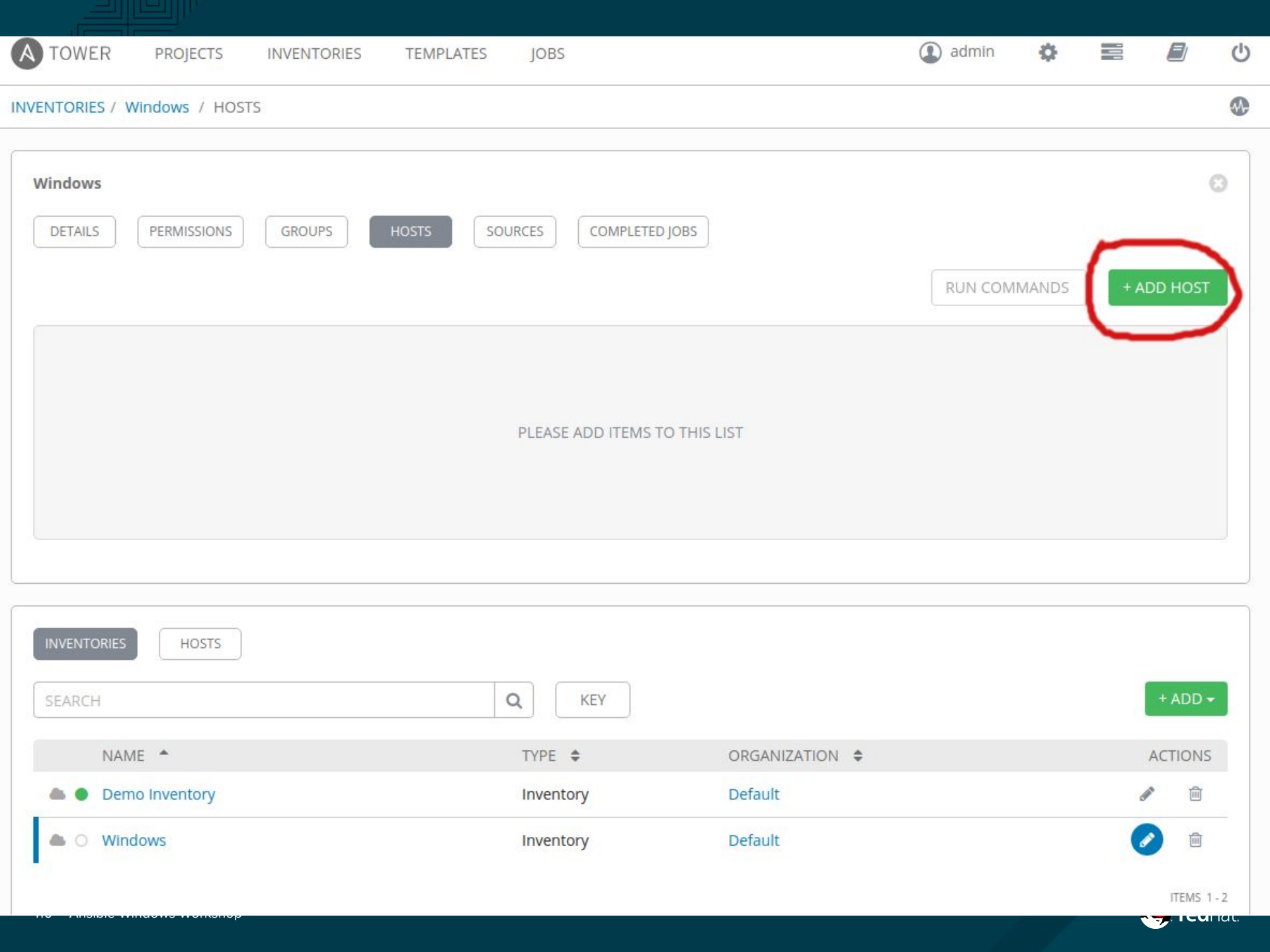

TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
PROJECTS
PROJECTS 1
SEARCH
KEY
+ ADD
NAME
TYPE
REVISION
LAST UPDATED
ACTIONS
Demo Project
Git
ITEMS 1 - 1

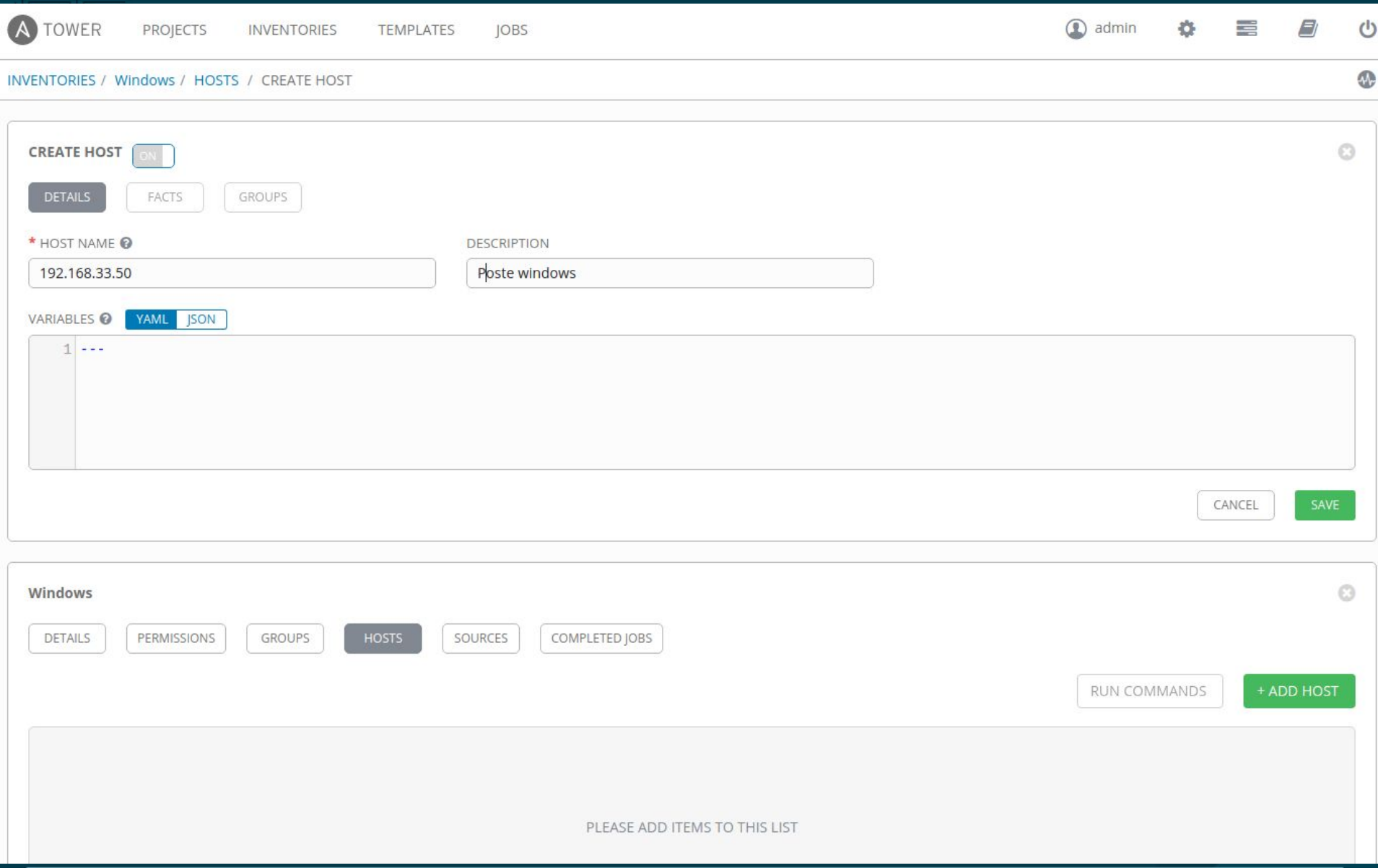

TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
PROJECTS / github
github
DETAILS
PERMISSIONS
NOTIFICATIONS
* NAME
github
DESCRIPTION
* ORGANIZATION
Default
* SCM TYPE
Git
SOURCE DETAILS
* SCM URL
https://github.com/RHMTLSAS/ansiblewindowsworkshop
SCM BRANCH/TAG/COMMIT
SCM CREDENTIAL
SCM UPDATE OPTIONS
Clean
Delete on Update
Update on Launch
CACHE TIMEOUT (SECONDS)
10
CANCEL
SAVE
PROJECTS 2
SEARCH
KEY
+ ADD

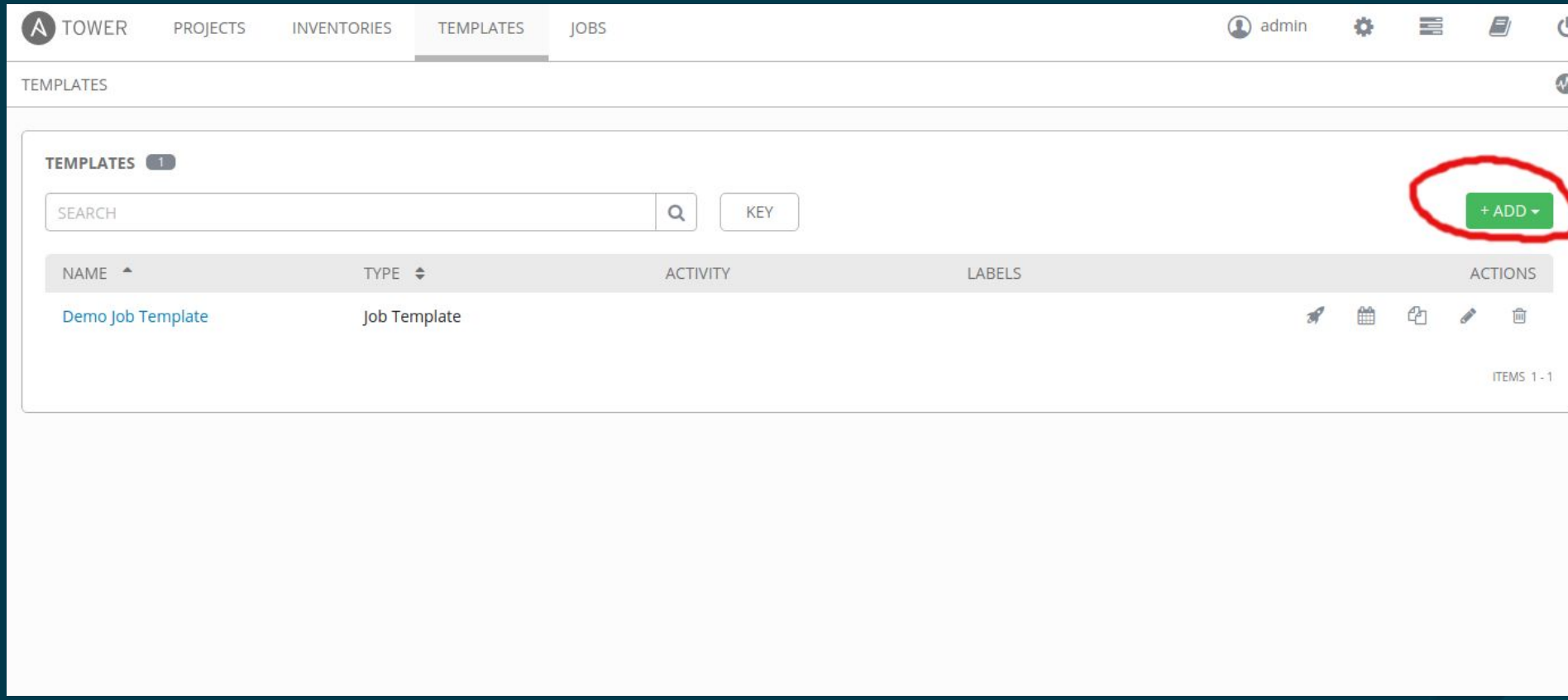

TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
INVENTORIES
INVENTORIES
HOSTS
SEARCH
KEY
+ ADD
NAME
TYPE
ORGANIZATION
ACTIONS
Demo Inventory
Inventory
Default
ITEMS 1 - 1

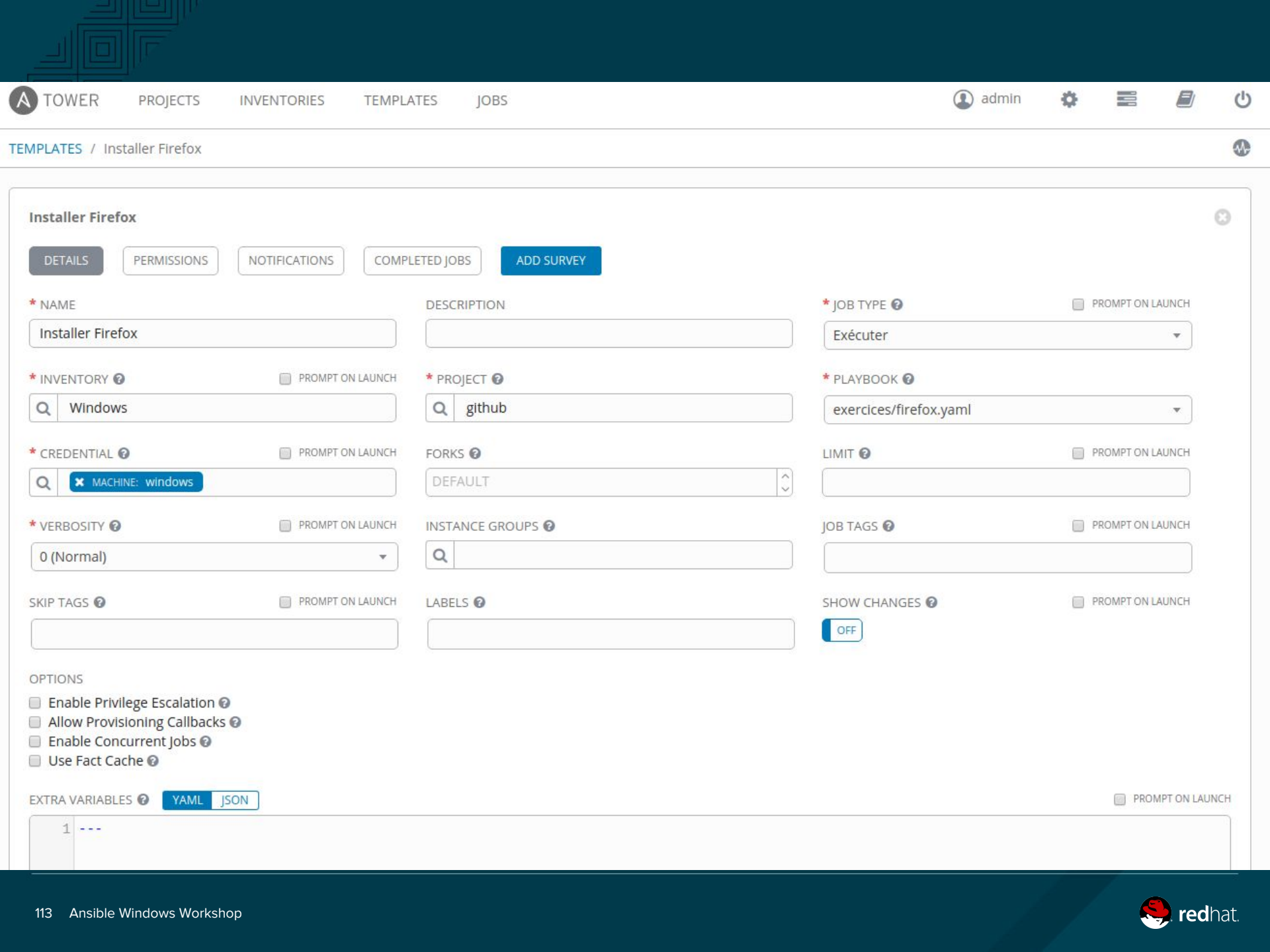
redhat.

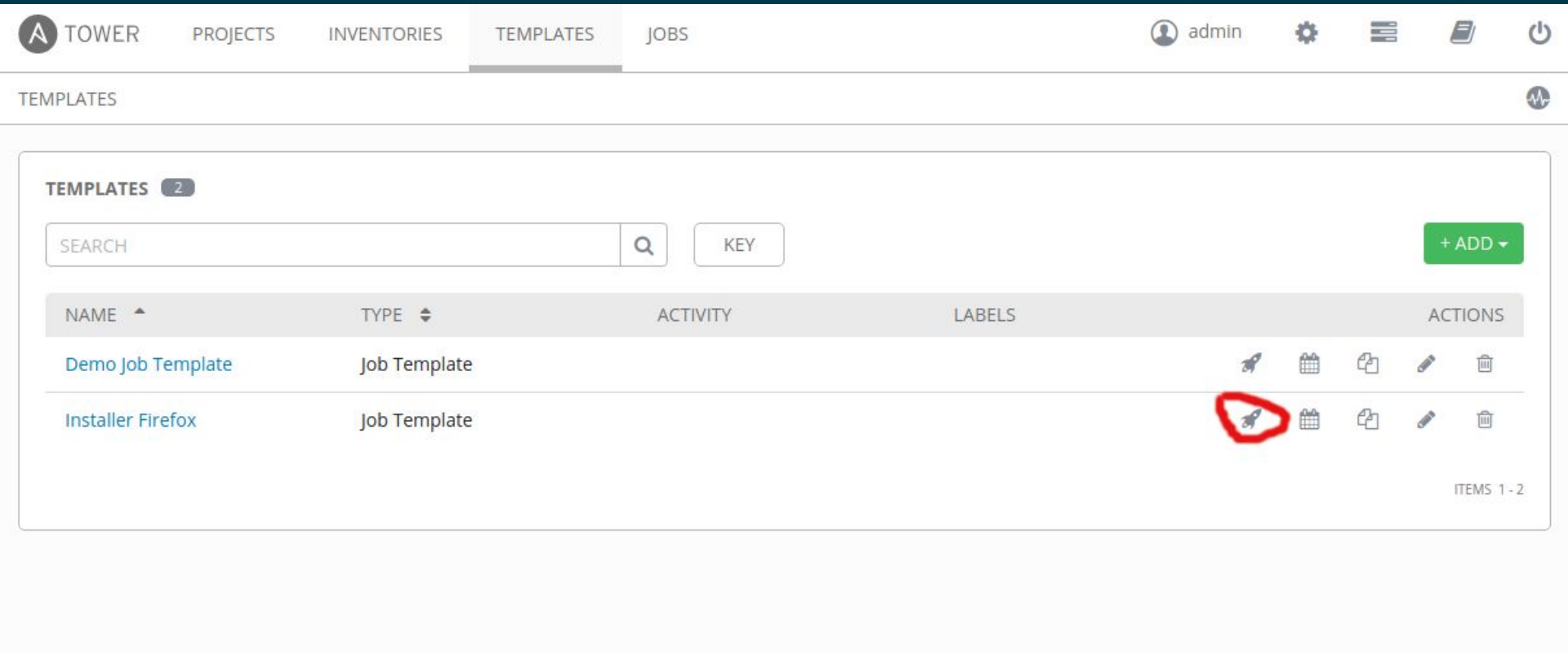

TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
INVENTORIES / Windows
Windows
DETAILS
PERMISSIONS
GROUPS
HOSTS
SOURCES
COMPLETED JOBS
* NAME
Windows
DESCRIPTION
* ORGANIZATION
Default
INSIGHTS CREDENTIAL
INSTANCE GROUPS
VARIABLES
YAML
JSON
1 ---
2 ansible_port: 5985
3 ansible_connection: winrm
4 ansible_winrm_transport: basic
CANCEL
SAVE
INVENTORIES
HOSTS
SEARCH
KEY
+ ADD
NAME
TYPE
ORGANIZATION
ACTIONS
Demo Inventory
Inventory
Default


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
INVENTORIES / Windows / HOSTS
Windows
DETAILS
PERMISSIONS
GROUPS
HOSTS
SOURCES
COMPLETED JOBS
RUN COMMANDS
+ ADD HOST
PLEASE ADD ITEMS TO THIS LIST
INVENTORIES
HOSTS
SEARCH
KEY
+ ADD
NAME
TYPE
ORGANIZATION
ACTIONS
Demo Inventory
Inventory
Default
Windows
Inventory
Default
ITEMS 1 - 2

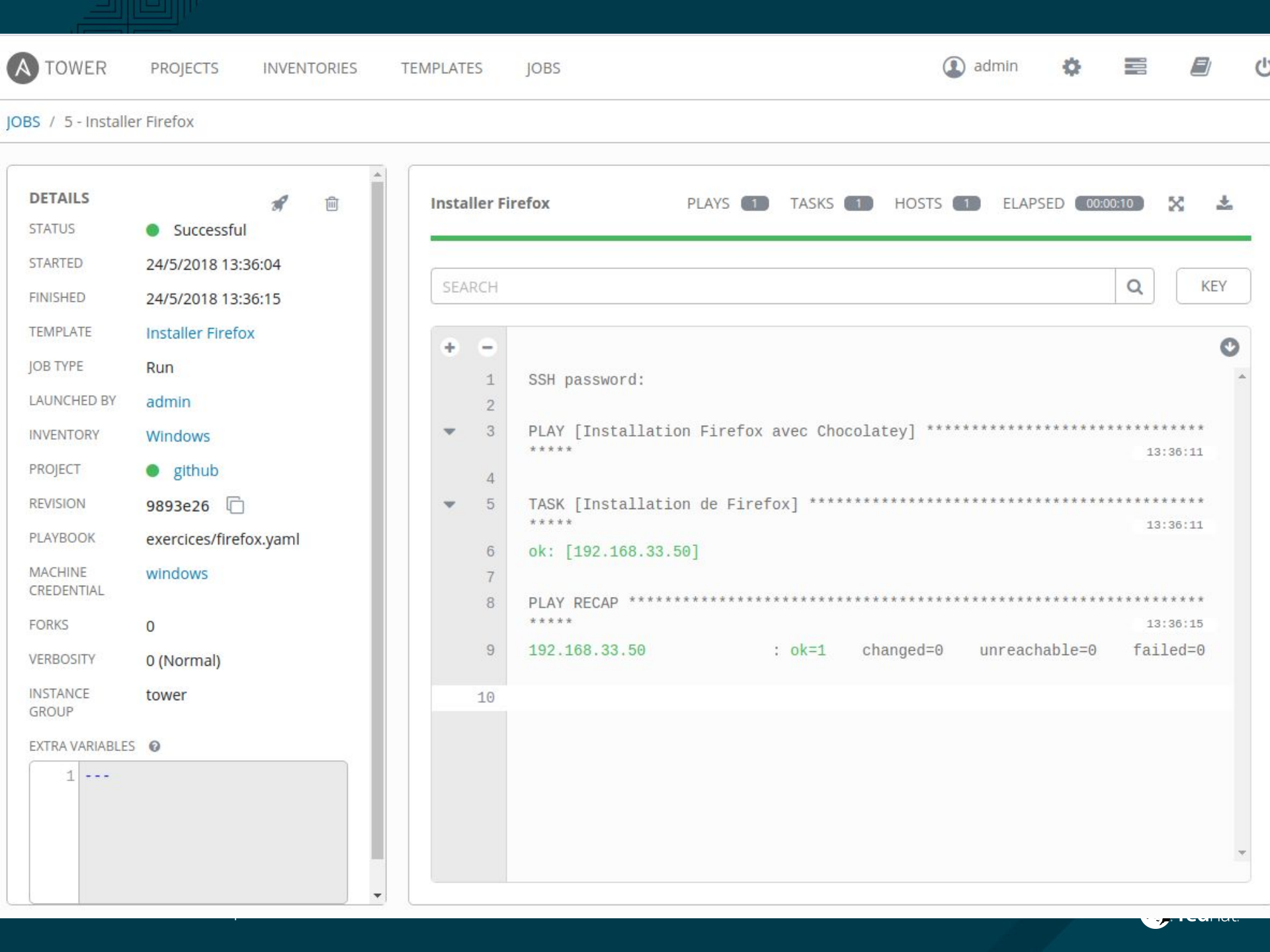

TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
INVENTORIES / Windows / HOSTS / CREATE HOST
CREATE HOST
ON
DETAILS
FACTS
GROUPS
* HOST NAME
192.168.33.50
DESCRIPTION
Poste windows
VARIABLES
YAML
JSON
1 ---
CANCEL
SAVE
Windows
DETAILS
PERMISSIONS
GROUPS
HOSTS
SOURCES
COMPLETED JOBS
RUN COMMANDS
+ ADD HOST
PLEASE ADD ITEMS TO THIS LIST


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
TEMPLATES
TEMPLATES 1
SEARCH
KEY
+ ADD
NAME
TYPE
ACTIVITY
LABELS
ACTIONS
Demo Job Template
Job Template
ITEMS 1 - 1


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
TEMPLATES / Installer Firefox
Installer Firefox
DETAILS
PERMISSIONS
NOTIFICATIONS
COMPLETED JOBS
ADD SURVEY
* NAME
Installer Firefox
DESCRIPTION
* JOB TYPE
PROMPT ON LAUNCH
Exécuter
* INVENTORY
PROMPT ON LAUNCH
Windows
* PROJECT
github
* PLAYBOOK
exercices/firefox.yaml
* CREDENTIAL
PROMPT ON LAUNCH
MACHINE: windows
FORKS
DEFAULT
LIMIT
PROMPT ON LAUNCH
* VERBOSITY
PROMPT ON LAUNCH
0 (Normal)
INSTANCE GROUPS
JOB TAGS
PROMPT ON LAUNCH
SKIP TAGS
PROMPT ON LAUNCH
LABELS
SHOW CHANGES
PROMPT ON LAUNCH
OFF
OPTIONS
Enable Privilege Escalation
Allow Provisioning Callbacks
Enable Concurrent Jobs
Use Fact Cache
EXTRA VARIABLES
YAML
JSON
PROMPT ON LAUNCH
1 ---


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
TEMPLATES
TEMPLATES 2
SEARCH
KEY
+ ADD
NAME
TYPE
ACTIVITY
LABELS
ACTIONS
Demo Job Template
Job Template
Installer Firefox
Job Template
ITEMS 1 - 2

redhat.


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
JOBS / 5 - Installer Firefox
DETAILS
STATUS
Successful
STARTED
24/5/2018 13:36:04
FINISHED
24/5/2018 13:36:15
TEMPLATE
Installer Firefox
JOB TYPE
Run
LAUNCHED BY
admin
INVENTORY
Windows
PROJECT
github
REVISION
9893e26
PLAYBOOK
exercices/firefox.yaml
MACHINE CREDENTIAL
windows
FORKS
0
VERBOSITY
0 (Normal)
INSTANCE GROUP
tower
EXTRA VARIABLES
1 ---
Installer Firefox
PLAYS 1
TASKS 1
HOSTS 1
ELAPSED 00:00:10
SEARCH
KEY
1 SSH password:
2
3 PLAY [Installation Firefox avec Chocolatey] ********** 13:36:11
4
5 TASK [Installation de Firefox] ********** 13:36:11
6 ok: [192.168.33.50]
7
8 PLAY RECAP ********** 13:36:15
9 192.168.33.50 : ok=1 changed=0 unreachable=0 failed=0
10

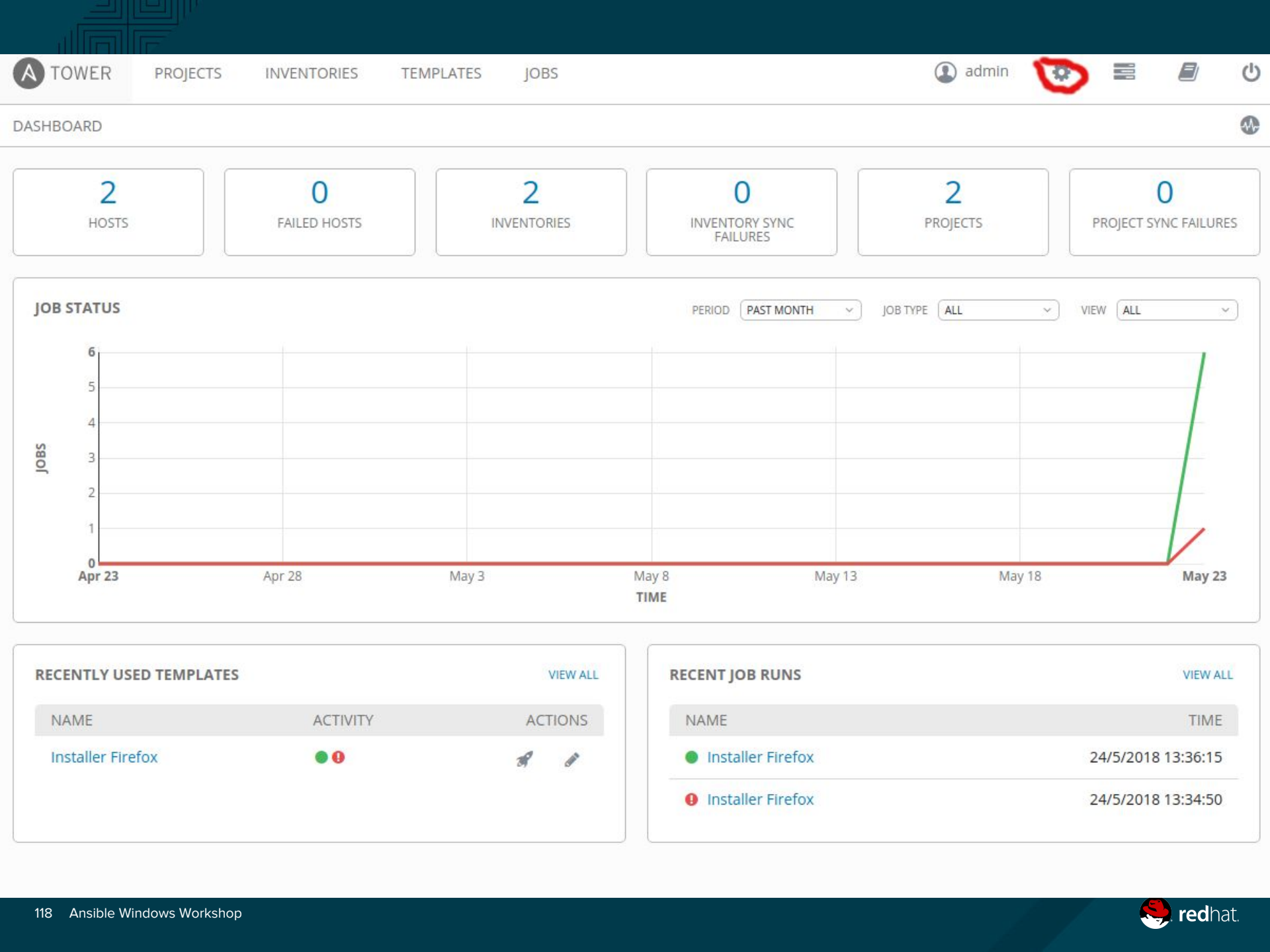
LAB #6 - Exercice guidée
Provisionnement de instance sur Azure

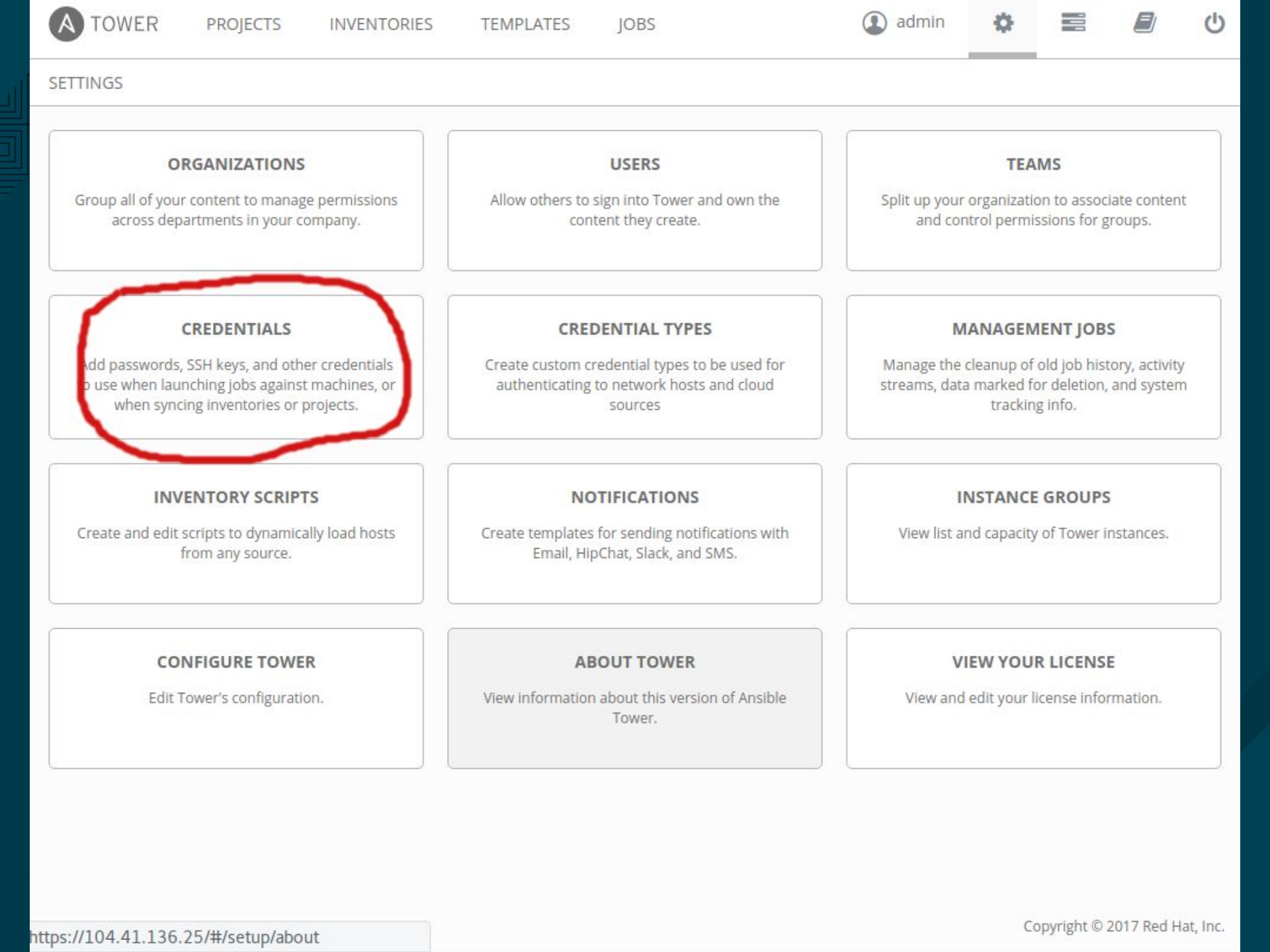
Préparation pour le provisionnement

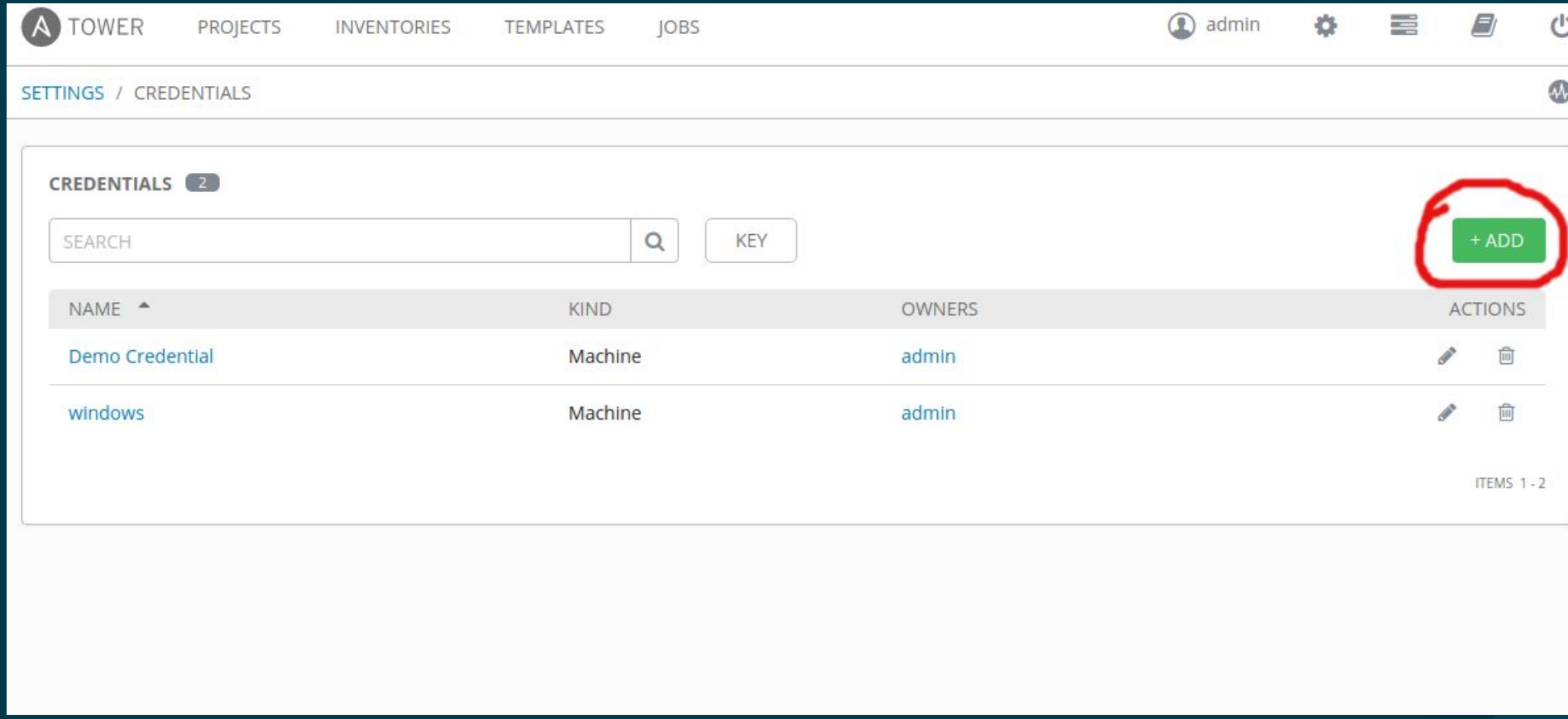
d'instance sur Azure

Connectez-vous avec putty sur votre machine Linux

```
[ansible@centosmick95 ~]$ sudo su

[root@centosmick95 ansible]# source /var/lib/awx/venv/ansible/bin/activate
[root@centosmick95 ansible]# pip install ansible[azure] --upgrade
--force
```


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
DASHBOARD
2
HOSTS
0
FAILED HOSTS
2
INVENTORIES
0
INVENTORY SYNC FAILURES
2
PROJECTS
0
PROJECT SYNC FAILURES
JOB STATUS
PERIOD
PAST MONTH
JOB TYPE
ALL
VIEW
ALL
JOBS
6
5
4
3
2
1
0
Apr 23
Apr 28
May 3
May 8
May 13
May 18
May 23
TIME
RECENTLY USED TEMPLATES
VIEW ALL
NAME
ACTIVITY
ACTIONS
Installer Firefox
RECENT JOB RUNS
VIEW ALL
NAME
TIME
Installer Firefox
24/5/2018 13:36:15
Installer Firefox
24/5/2018 13:34:50


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
SETTINGS
ORGANIZATIONS
Group all of your content to manage permissions across departments in your company.
USERS
Allow others to sign into Tower and own the content they create.
TEAMS
Split up your organization to associate content and control permissions for groups.
CREDENTIALS
dd passwords, SSH keys, and other credentials o use when launching jobs against machines, or when syncing inventories or projects.
CREDENTIAL TYPES
Create custom credential types to be used for authenticating to network hosts and cloud sources
MANAGEMENT JOBS
Manage the cleanup of old job history, activity streams, data marked for deletion, and system tracking info.
INVENTORY SCRIPTS
Create and edit scripts to dynamically load hosts from any source.
NOTIFICATIONS
Create templates for sending notifications with Email, HipChat, Slack, and SMS.
INSTANCE GROUPS
View list and capacity of Tower instances.
CONFIGURE TOWER
Edit Tower's configuration.
ABOUT TOWER
View information about this version of Ansible Tower.
VIEW YOUR LICENSE
View and edit your license information.
https://104.41.136.25/#/setup/about
Copyright © 2017 Red Hat, Inc.


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
SETTINGS / CREDENTIALS
CREDENTIALS 2
SEARCH
KEY
+ ADD
NAME
KIND
OWNERS
ACTIONS
Demo Credential
Machine
admin
windows
Machine
admin
ITEMS 1 - 2

redhat.

NEW CREDENTIAL

DETAILS PERMISSIONS

* NAME: Azure

DESCRIPTION:

ORGANIZATION: SELECT AN ORGANIZATION

* CREDENTIAL TYPE: Microsoft Azure Resource Manager

TYPE DETAILS

* SUBSCRIPTION ID: 95814c98-6f58-4f05-bc32-88c70c6783b9

USERNAME:

PASSWORD: SHOW

CLIENT ID: c28ee7eb-a18f-4946-9127-8bebce403ee0

CLIENT SECRET: SHOW ••

TENANT ID: fce45aa5-c290-4960-afaa-17ccfbb6e8b3

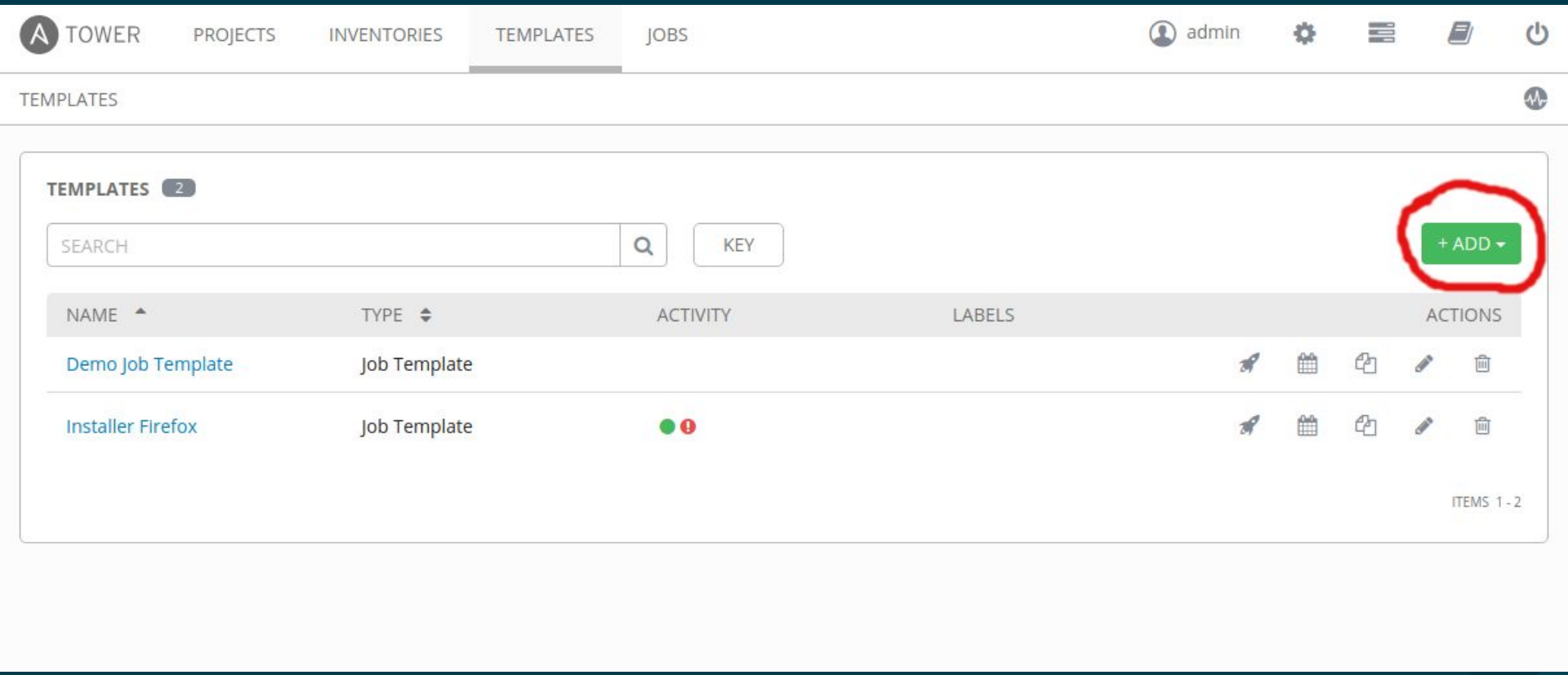
AZURE CLOUD ENVIRONMENT:

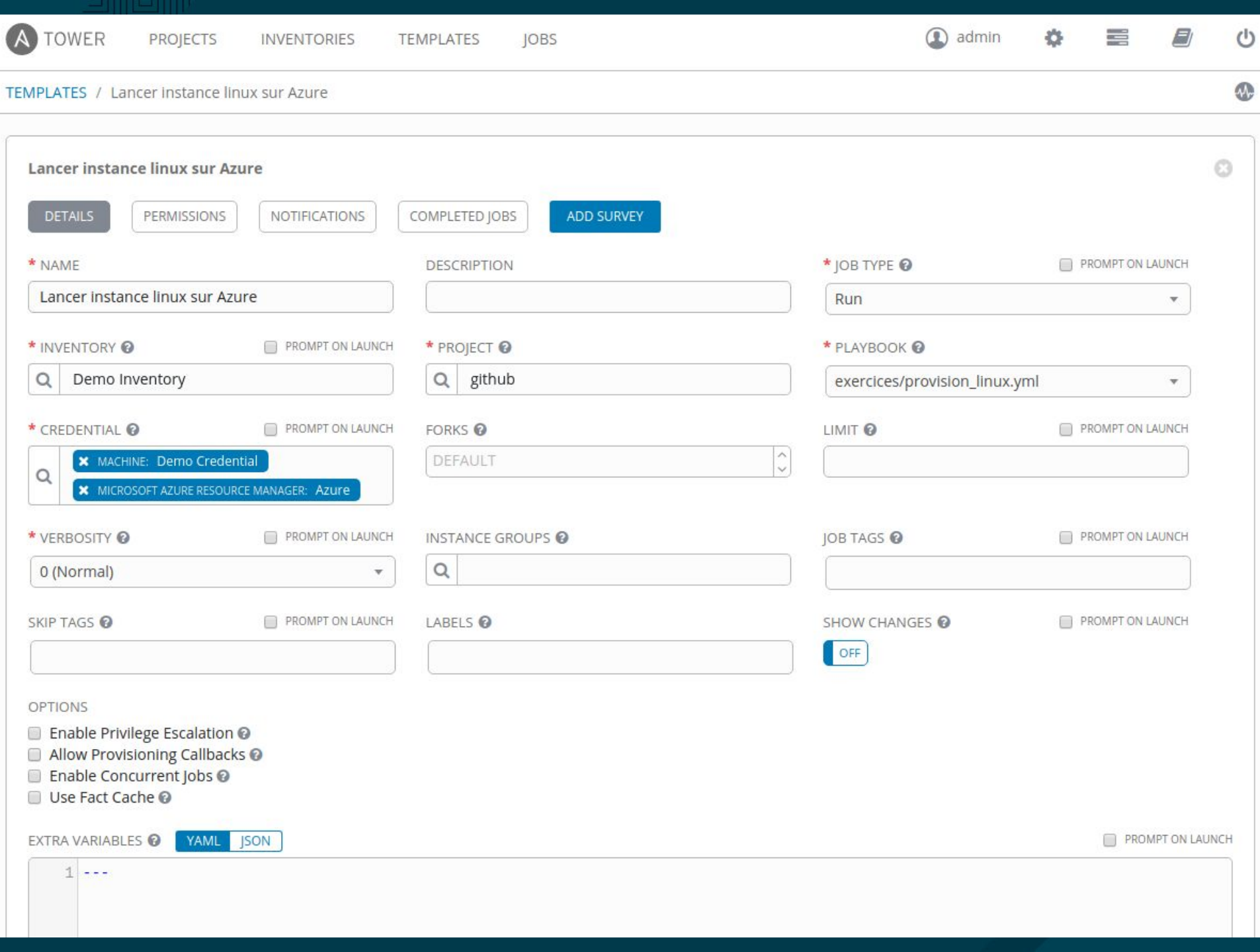
CANCEL SAVE

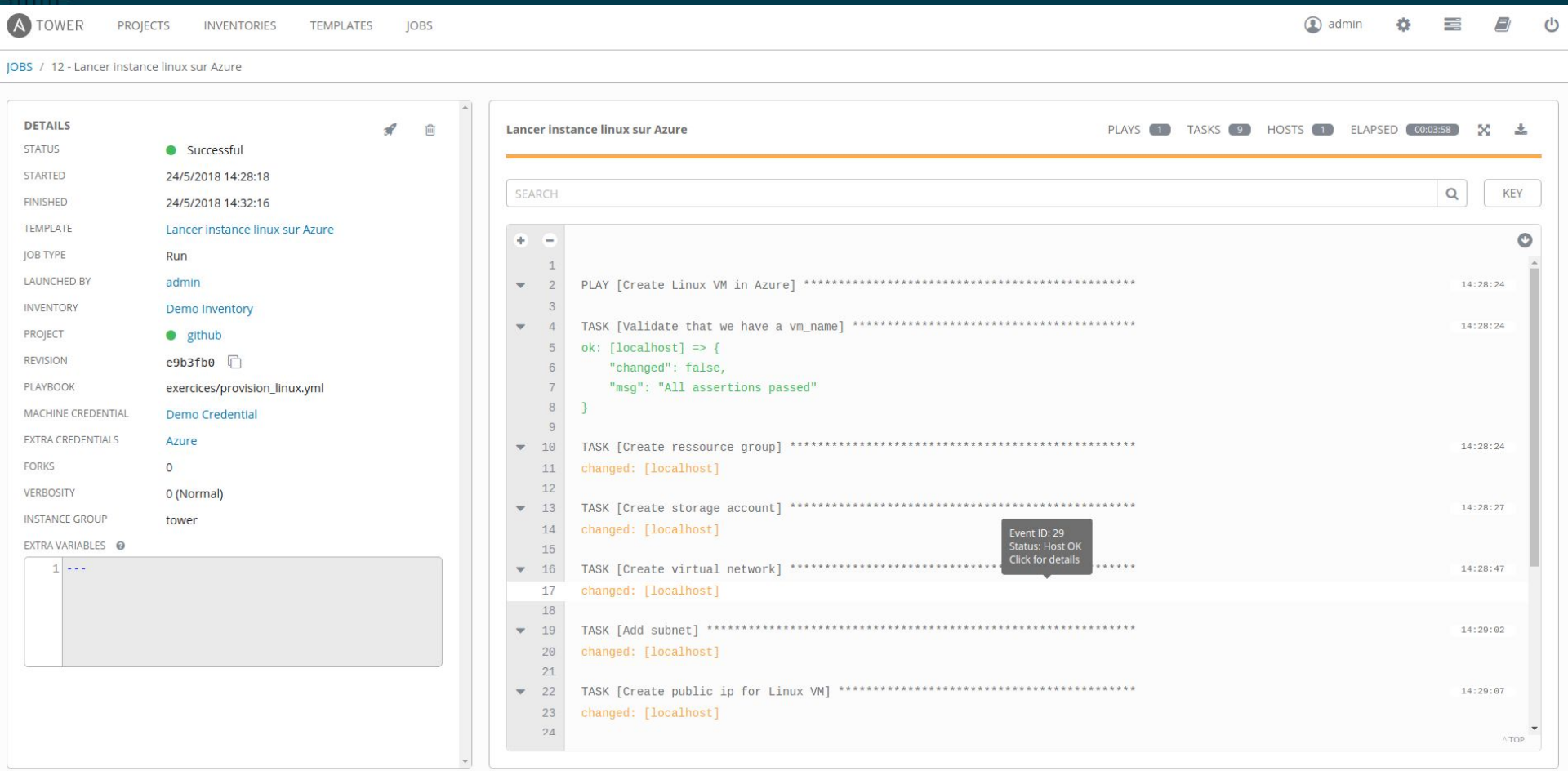
CREDENTIALS 2

SEARCH KEY + ADD

NAME	KIND	OWNERS	ACTIONS
Demo Credential	Machine	admin	


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
TEMPLATES
TEMPLATES 2
SEARCH
KEY
+ ADD
NAME
TYPE
ACTIVITY
LABELS
ACTIONS
Demo Job Template
Job Template
Installer Firefox
Job Template
ITEMS 1 - 2


TOWER
PROJECTS
INVENTORIES
TEMPLATES
JOBS
admin
TEMPLATES / Lancer instance linux sur Azure
Lancer instance linux sur Azure
DETAILS
PERMISSIONS
NOTIFICATIONS
COMPLETED JOBS
ADD SURVEY
* NAME
Lancer instance linux sur Azure
DESCRIPTION
* JOB TYPE
PROMPT ON LAUNCH
Run
* INVENTORY
PROMPT ON LAUNCH
Demo Inventory
* PROJECT
github
* PLAYBOOK
exercices/provision_linux.yml
* CREDENTIAL
PROMPT ON LAUNCH
MACHINE: Demo Credential
MICROSOFT AZURE RESOURCE MANAGER: Azure
FORKS
DEFAULT
LIMIT
PROMPT ON LAUNCH
* VERBOSITY
PROMPT ON LAUNCH
0 (Normal)
INSTANCE GROUPS
JOB TAGS
PROMPT ON LAUNCH
SKIP TAGS
PROMPT ON LAUNCH
LABELS
SHOW CHANGES
PROMPT ON LAUNCH
OFF
OPTIONS
Enable Privilege Escalation
Allow Provisioning Callbacks
Enable Concurrent Jobs
Use Fact Cache
EXTRA VARIABLES
YAML
JSON
PROMPT ON LAUNCH
1 ---




MERCI
redhat.

EXTRA

Configuration SFTP en locale

Configuration de Visual Studio Code en locale

1. Installer Visual Studio Code sur votre laptop (https://code.visualstudio.com/docs/setup/setup-overview)
2. Télécharger les fichiers suivants :
   https://raw.githubusercontent.com/RHMTLSAS/ansible-azure-lab/master/insecure
   https://raw.githubusercontent.com/RHMTLSAS/ansible-azure-lab/master/insecure.pub
3. Créer un répertoire de travail : c:\labs (ou à votre choix)
4. Copier les deux fichiers télécharger plus haut dans ce répertoire
5. Ouvrez VSE, File Open Folder et choisissez le dossier créé plus haut
6. Appuyer sur shift+ctrl+P
7. sftp:config
8. Copier et coller le fichier de config vsftp disponible sur le etherpad
9. Editer cette configuration en y indiquant l'adresse ip publique de votre serveur CentOS
10. Enregistrer avec ctrl+s

ASTUCE: Edition de fichiers YAML

```
# curl https://raw.githubusercontent.com/marcosgm/nanorc/master/install.sh  | sh
# LANG=fr_FR nano playbook.yml
```