

Guided Practice: Adding Windows Users and Groups to a Server with Ansible

Introduction

Ansible's strengths include the ability to facilitate the configuration of UNIX, Linux, Windows, and Mac operating systems that include groups that may be dependent on other groups, which can involve the secure backing up of data, configuring firewalls, creating users and groups, and replication.

Outcome

In this Guided Practice, you will write a playbook to manage users and groups on your Windows server.

Resources Needed

- For this Guided Practice, we will use the CentOS 8 and the Windows 2019 server in the VCastle pod configured for this class.

Level of Difficulty

Medium

Deliverables

Deliverables are marked with a red border around the screenshot. Additionally, there are guided practice questions at the end which you must respond to.

General Considerations

You should be familiar with Linux and Windows networking. Secure Shell (ssh) is installed and configured on the Ubuntu server from a prior Guided Practice. Ansible is installed and configured on your CentOS Computer, and you have completed all the previous exercises. Consider running the playbook in increments and testing each section as you go. This may make troubleshooting easier and lets you take advantage of the idempotency feature in Ansible (a playbook can be run multiple times without affecting initial settings).

Instructions

Important: Please Note

All of your screenshots should include information that shows your login information.

Edit the Ansible Inventory (host) File

First, you will edit the Ansible Inventory (host) file to reflect your network architecture, if necessary.

1. View the inventory (hosts) file, and ensure your inventory file reflects the architecture of your network as shown below and includes the [win:vars] section.

```
1 [webserver]
2 192.168.1.3
3
4 [dbserver]
5 192.168.1.3
6
7 [win]
8 192.168.1.2
9
10 [win:vars]
11 ansible_user=cis321
12 ansible_password=Password1
13 ansible_connection=winrm
14 ansible_winrm_server_cert_validation=ignore
```

Write a Playbook to Create Two New Users and a Group on Your Windows Server

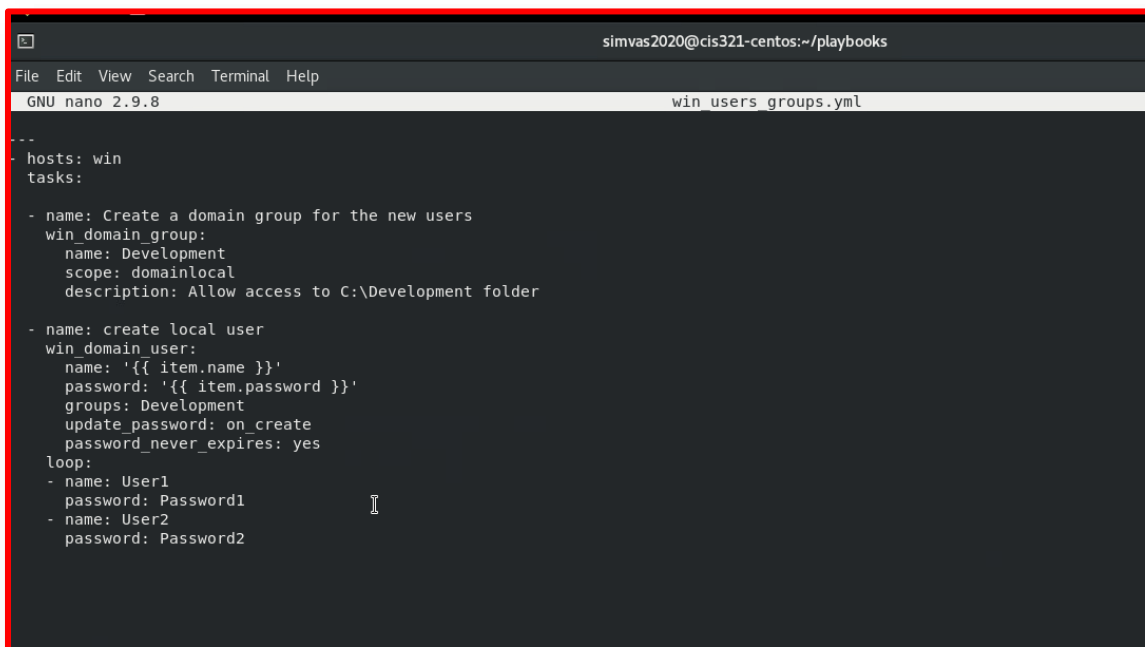
1. Open a shell, use your favorite command line text editor to add the following to win_users_groups.yml then save and close the file:

```
---
- hosts: win
  tasks:

    - name: Create a domain group for the new users
      win_domain_group:
        name: Development
        scope: domainlocal
        description: Allow access to C:\Development folder

    - name: create local user
      win_domain_user:
        name: '{{ item.name }}'
```

```
password: '{{ item.password }}'
groups: Development
update_password: on_create
password_never_expires: yes
loop:
- name: User1
  password: Password1
- name: User2
  password: Password2
```

A screenshot of a terminal window with a red border. The title bar shows the user 'simvas2020' at 'cis321-centos' in the directory '~/playbooks'. The terminal shows the GNU nano 2.9.8 editor editing the file 'win_users_groups.yml'. The content of the file is a YAML playbook with two tasks: 'win_domain_group' and 'win_domain_user'. The 'win_domain_user' task includes a loop for creating two users, 'User1' and 'User2', with passwords 'Password1' and 'Password2' respectively. The terminal background is dark, and the text is light gray.

```
simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help
GNU nano 2.9.8 win_users_groups.yml
---
- hosts: win
  tasks:

  - name: Create a domain group for the new users
    win_domain_group:
      name: Development
      scope: domainlocal
      description: Allow access to C:\Development folder

  - name: create local user
    win_domain_user:
      name: '{{ item.name }}'
      password: '{{ item.password }}'
      groups: Development
      update_password: on create
      password_never_expires: yes
    loop:
    - name: User1
      password: Password1
    - name: User2
      password: Password2
```

Take a screenshot that resembles the one above, and paste it in your Lab Report.

2. Check the syntax and run the playbook. Note that the IP addresses in your output will reflect your network architecture and may differ from what is shown below.

In your shell type the following:

```
ansible-playbook win_users_groups.yml --syntax-check
ansible-playbook win_users_groups.yml
```

```

[alex@localhost vanVugt]$ ansible-playbook groups.yml

PLAY [win] *****

TASK [Gathering Facts] *****
ok: [192.168.0.77]

TASK [Create local group to contain new users] *****
ok: [192.168.0.77]

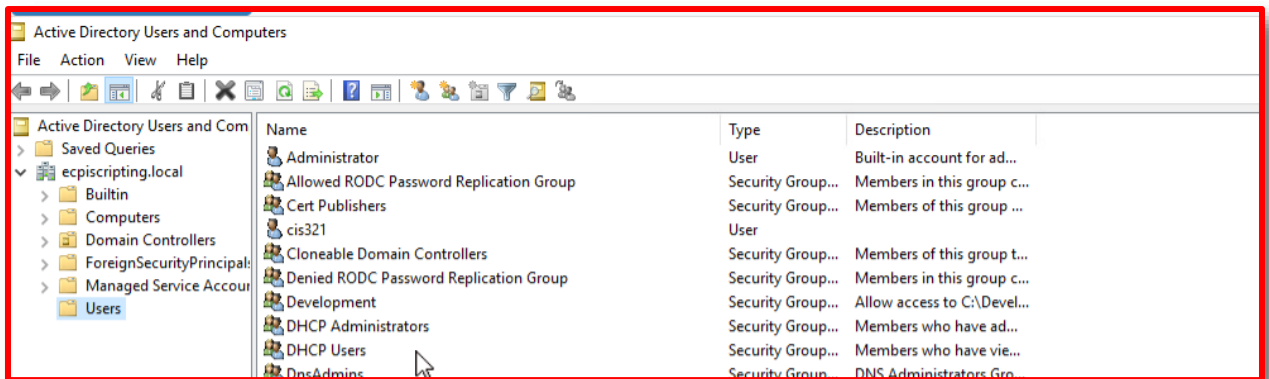
TASK [Create local user] *****
changed: [192.168.0.77] => (item={'name': 'User1', 'password': 'Password1'})
changed: [192.168.0.77] => (item={'name': 'User2', 'password': 'Password2'})

PLAY RECAP *****
192.168.0.77 : ok=3 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

```

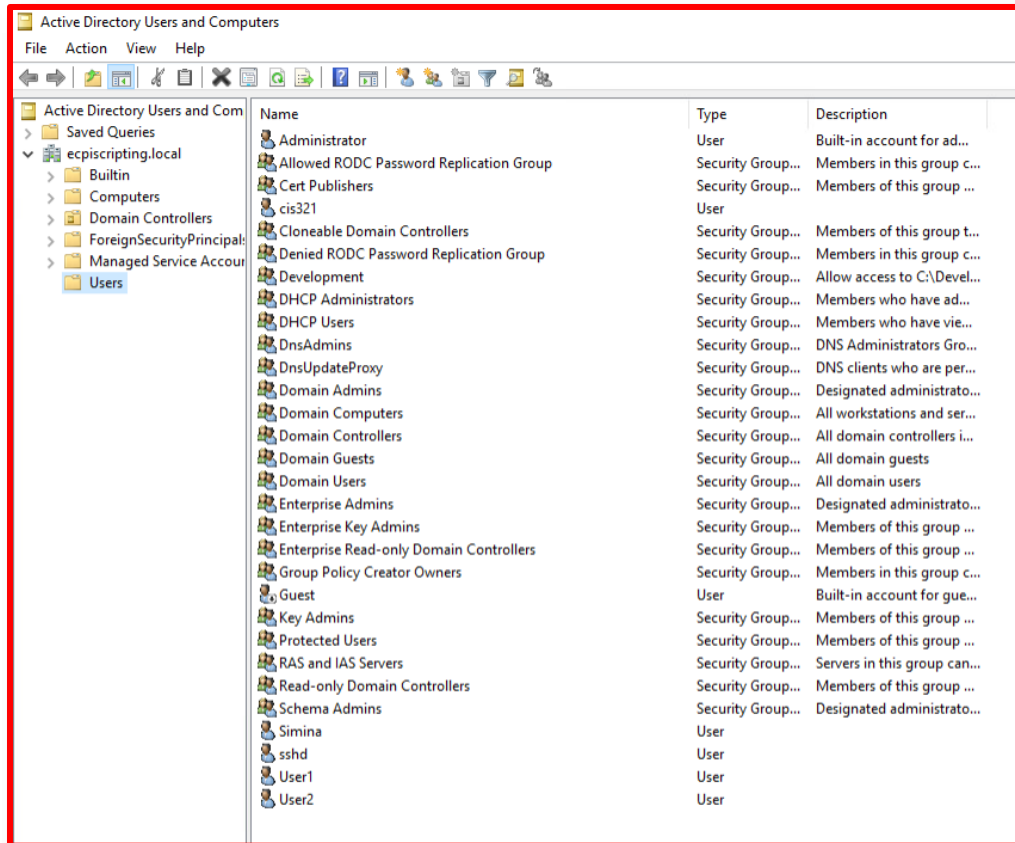
Take a screenshot that resembles the one above and paste it in your Lab Report.

3. Verify that the users have been created. Depending on your configuration, you can view the status in Computer Management or via the Server Manager. If you are using VCASTLE you are in and Active Directory environment and you will find this info under



Take a screenshot that resembles the one above and paste it in your Lab Report.

4. Verify Group has been created. Depending on your configuration, you can view the status in Computer Management or via the Server Manager Dashboard under Tools.



Take a screenshot that resembles the one above and paste it in your Lab Report.

Guided Practice Questions

In your **Guided Practice Lab Report**, answer the following questions about this learning activity. Some may require research.

1. What action should you take if the create users playbook ran successfully, but the new users and group does not appear in your Windows Management console?
2. What happens if you have successfully validated that the new group and users have been created and you run the same playbook again? Why?

References

Ansible. (2020a). *Manage Windows user rights*.

https://docs.ansible.com/ansible/latest/collections/ansible/windows/win_user_right_module.html

Ansible. (2020b). *Set up users and groups*.

https://docs.ansible.com/ansible/latest/user_guide/windows_usage.html#use-cases

Ansible. (2020c). *Windows guides*.

https://docs.ansible.com/ansible/latest/user_guide/windows.html#windows

Avi. (2020). *9 Ansible playbooks example for Windows administration*. <https://geekflare.com/ansible-playbook-windows-example/>

Geerling, J. (2014). *geerlingguy/mac-dev-playbook*. <https://github.com/geerlingguy/mac-dev-playbook>

Kiarie, J. (2020). *How to manage remote Windows host using Ansible*.

<https://www.linuxtechi.com/manage-windows-host-using-ansible/>

RapidAPI. (2021). *What is idempotency?*. <https://rapidapi.com/blog/api-glossary/idempotency/>

O'Reilly References

VIDEO

[Hands-on Ansible](#)

By [Sander van Vugt](#)

[Pearson IT Certification](#) March 2020

[3.2 Understanding Ansible Modules](#)

[Automating with Ansible](#)

By [Sander van Vugt](#)

[Addison-Wesley Professional](#) November 2018

[3.3 Managing Windows with Ansible](#)