Sirius Ansible Palo Alto Immersion Day Workshop

Abstract

This Sirius Ansible Immersion Day Workshop will introduce using Ansible with the Palo Alto firewall.

Rich Mallory

rich.mallory@siriuscom.com

Table of Contents

Overview Palo Alto Immersion Day Workshop

You will use Ansible commands and playbooks to explore and reconfigure a Palo Alto in a virtual environment. Note, that even though we are using a virtual environment, this is not a requirement, everything we do, can be done on physical devices.

The Immersion Day is meant as a follow on to the Network Automation Immersion Day. Some content will be review but in respect to using Ansible with Palo Alto. While you can go through these labs there may be concepts that were covered previously that we will not cover as it is expected that these concepts are already understood. Throughout the labs we hope to share some additional features, elements, good and bad practices, and patterns to using Ansible for Automation.

You will be required to modify some files during this workshop. You will not be required to write your own playbooks as this would require much more time. The playbooks used are open source and thus free to use and modify. Writing playbooks and running them in a test environment is one of the best ways to learn.

Note: The output shown in the examples may vary from yours.

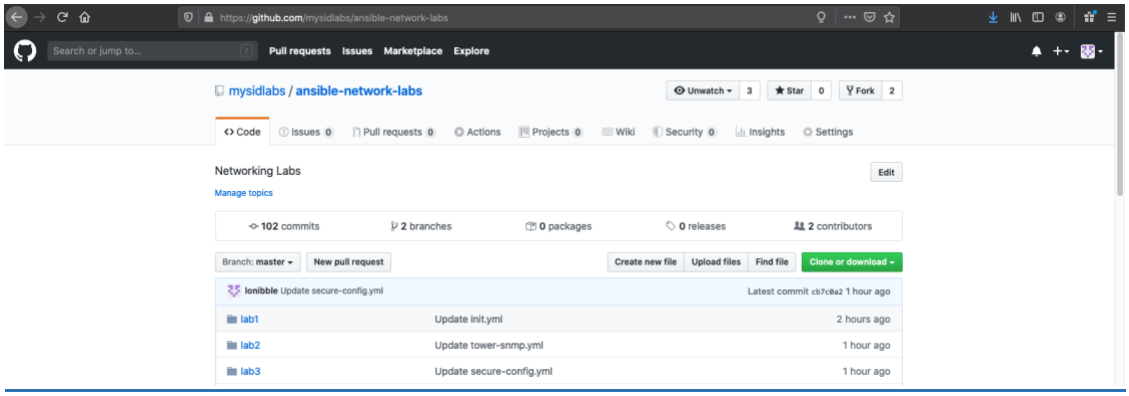
Part 1A - Initial Lab Setup

At the end of this section the lab github repo will be forked into your own github account. You will access the jump station and download the newly forked github repo.

1. Fork the lab github repository to your own repository so you can edit and modify.
2. Access the jump station
3. Download the forked repository to the jump station

Step 1 - Fork the Sirius ansible Pan Labs Github repository

- Login in to Github at https://github.com
- Go to https://github.com/mysidlabs/pan-labs-ee-1
- Click on the Fork button in upper right.



Note: Once forked you can modify view and modify all files within your GitHub

Step 2 - Connect to the Jump host

- SSH to the jump station at **jump.mysidlabs.com**

For MacOS or Linux users the following is an example using the terminal:

$ ssh siduserxxx@siduserxxx.jump.mysidlabs.com
Ex. $ssh siduser101@siduser101.jump.mysidlabs.com

You may get the following message, type yes **at the prompt:**

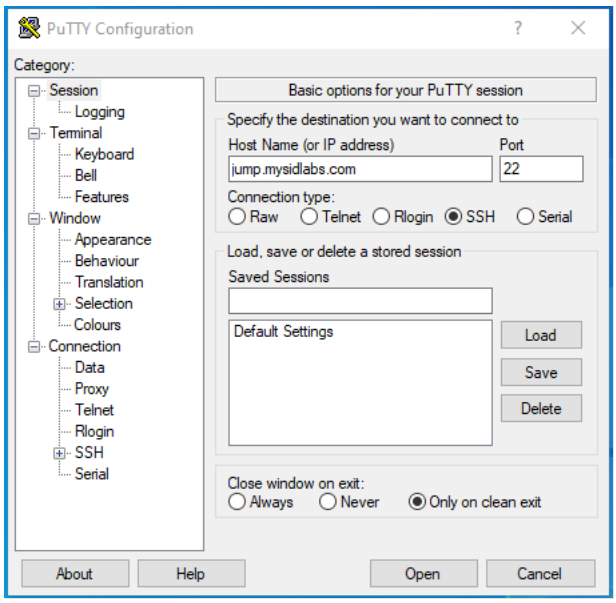
The authenticity of host 'jump.mysidlabs.com (3.132.28.93)' can't be established.
ECDSA key fingerprint is SHA256: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Are you sure you want to continue connecting (yes/no/[fingerprint])? **Yes**

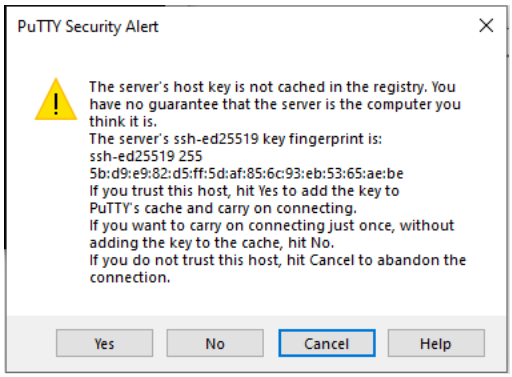
Warning: Permanently added 'siduser101.jump.mysidlabs.com,3.132.28.93' (ECDSA) to the list of known hosts.

Note: You can remove from known hosts after workshop is completed.
When prompted for your password type in the password the instructor provides
password: *******

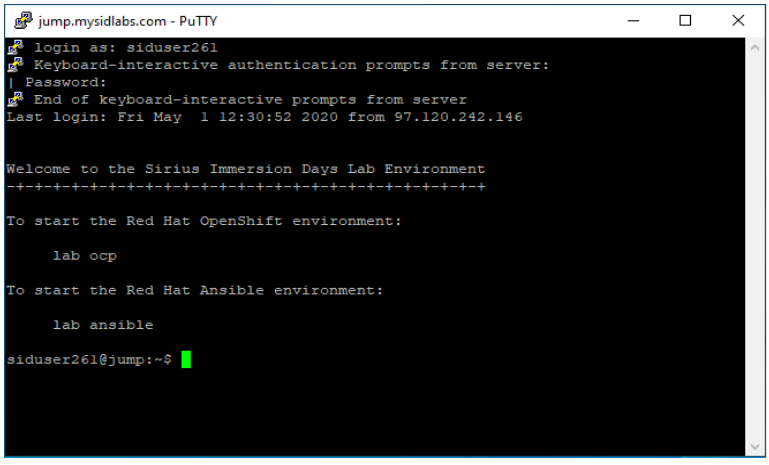
For Windows users the following is an example using Putty:
Type jump.mysidlabs.com in the Host Name box and click the Open button



Click the **Yes** button to accept the ssh key



Type in username and password in the terminal screen at the appropriate prompts



Step 3 - Download forked repository to the jump station

1. Your terminal prompt should change to something like the following:
   siduser101@jump:~$

4. Clone your repository
   siduser101@jump ~ # git clone https://github.com/<<YOUR_GITHUB_USER>>/pan-labs-ee-1

Tip	The usage of git becomes very important to “infrastructure as code”. Everything resides in github including your changes. If you lose connection from the jump box, the repository will be deleted automatically. All you need to do is clone your repository and you are back to where you were.

5. You should now see the repository in your directory
   siduser101@jump:~$ ls
   pan-labs-ee-1

6. Move into the pan-labs-ee-1directory
   siduser101@jump ~ # **cd pan-labs-ee-1**
   siduser101@jump:~/pan-labs-ee-1$

Additional Information

You can now explore the labs directory

cd = change directory
ls = list contents
pwd = display current working directory
cat = display file
nano or vim = file editor
tree = display file structure from current directory

Note: At this point your jump host is setup is complete and is ready to use

Part 1B - Connect to Palo Alto firewall via SSH and set your username/password

This section will accomplish the following two tasks:

1. Connect to your firewall via SSH and add new admin user to use for ansible playbooks.
2. Connect to your firewall via HTTPS with your new admin username/password.

Your student Palo Alto Firewall does not have a username and password configured yet. Access the firewall and set a username and password. Replace <<ID>> with your student number.

Step 1 - Connect to FW via ssh from jump host.

1. **ssh -i ~/.ssh/network-key.pem admin@siduser<ID>.pan.mysidlabs.com**
   load pubkey "/home/siduser101/.ssh/network-key.pem": invalid format
   The authenticity of host 'siduser101.pan.mysidlabs.com (3.133.142.207)' can't be established.
   RSA key fingerprint is SHA256:tfEPSNM5pDbgEOEKv5H059pu1uK8l5T2QjcLIDU03eE.
   Are you sure you want to continue connecting (yes/no/[fingerprint])? **yes**
   Warning: Permanently added 'siduser101.pan.mysidlabs.com,3.133.142.207' (RSA) to the list of known hosts.

Welcome admin.
admin@PA-VM>

2. Go into configuration mode

admin@PA-VM> **configure**

3. Add a new user

admin@PA-VM# **set mgt-config users siduser<ID> password**
Enter password :
Confirm password :

4. Give your new user admin permissions

admin@PA-VM# **set mgt-config users siduser<ID> permissions role-based superuser yes**

5. Commit Palo Alto changes

admin@PA-VM# **commit**

Commit job 2 is in progress. Use Ctrl+C to return to command prompt
.....55%98%.....100%
Configuration committed successfully

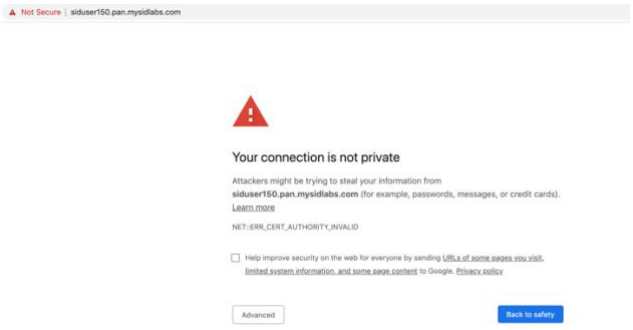
6. Exit from Firewall

Type **exit** twice to disconnect from your student Palo Alto Firewall

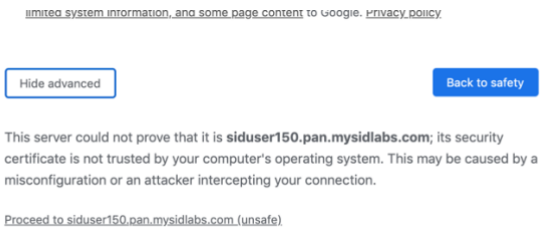
admin@PA-VM# **exit**
Exiting configuration mode
admin@PA-VM> **exit**
Connection to siduser101.pan.mysidlabs.com closed.
siduser101@jump:~/pan-labs-ee-1$

Step 2 - Connect to Palo Alto via HTTPS with new username and password

1. Open a web browser and connect to https://siduser<<ID>>.pan.mysidlabs.com



2. Click Advanced



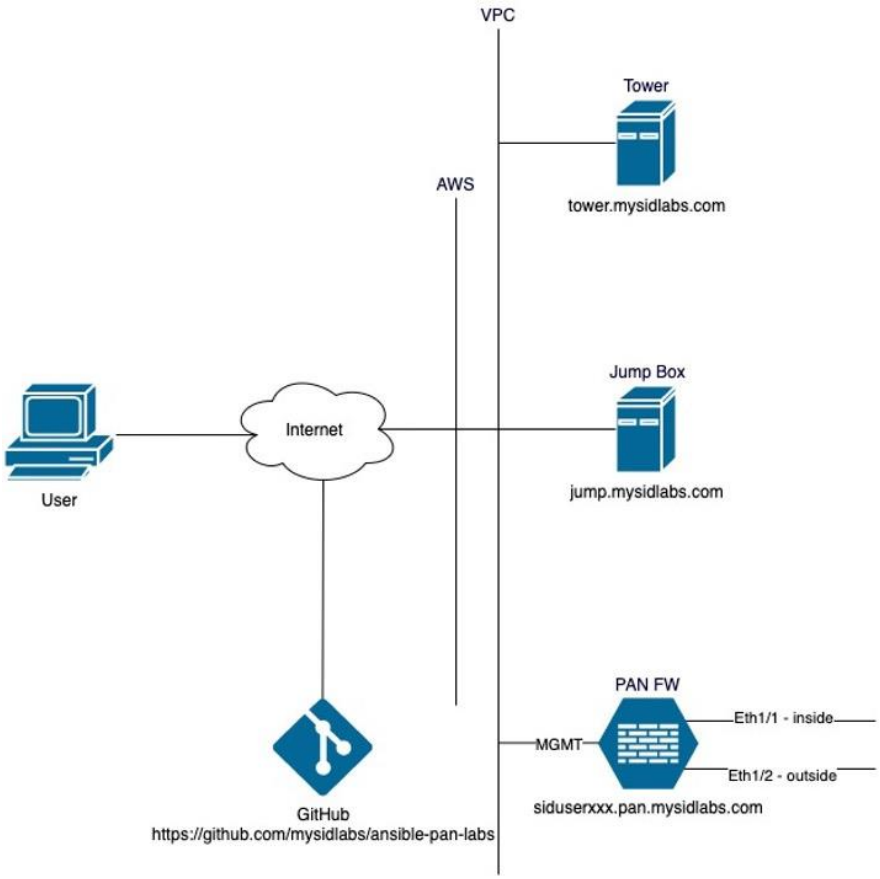
3. Click “Proceed to siduser<<ID>>.pan.mysidlabs.com (unsafe)

4. Login with your credentials, verify you can see the PAN dashboard

Part 2 - Ansible Labs

Lab Topology

The topology is simple for the sake of learning some ansible basics. The diagram below is an example, the XXX in the hostname is your Student ID. If you are student 199 then the hostname for the PaloAlto would be siduser199.pan.mysidlabs.com.



Lab 1.0: Explore your Lab Environment

Step 1 - Make sure you are in the pan-labs-ee-1 folder

```
siduser101@jump:~/pan-labs-ee-1$ pwd
/home/siduser101/pan-labs-ee-1
```

If you are not, change to the pan-labs-ee-1 directory

```
siduser101@jump:~ # cd ~/pan-labs-ee-1
```

Step 2 - Review the ansible-navigator.yml file

```
siduser101@jump:~/pan-labs-ee-1/project$ cat ansible-navigator.yml
---
ansible-navigator:
  ansible:
    inventory:
      entries:
      - inventory
  execution-environment:
    container-engine: podman
    enabled: true
    image: localhost/pan-lab-ee-1:latest
    pull:
      policy: missing
  mode: stdout
  playbook-artifact:
    enable: false
```

Step 3 - Review the currently installed execution environments

```
siduser101@jump:~/pan-labs-ee-1$ podman images
REPOSITORY  TAG         IMAGE ID    CREATED     SIZE
```

Step 4 - Run Ansible Navigator

```
siduser101@jump:~/pan-labs-ee-1$ ansible-navigator
----------------------------------------------------------------------------------------------------
Execution environment image and pull policy overview
----------------------------------------------------------------------------------------------------
Execution environment image name:     registry.redhat.io/ansible-automation-platform-22/ee-supported-rhel8:latest
Execution environment image tag:      latest
Execution environment pull arguments: None
```

Execution environment pull policy: tag

Execution environment pull needed: True

Updating the execution environment

Running the command: podman pull registry.redhat.io/ansible-automation-platform-22/ee-supported-rhel8:latest

Trying to pull registry.redhat.io/ansible-automation-platform-22/ee-supported-rhel8:latest...

Getting image source signatures

Checking if image destination supports signatures

Copying blob 2dfca0f3f79a done

Copying blob d5d2e87c6892 done

<< Output Ommitted >>

Step 5 - Pull in the execution environment

siduser101@jump:~/pan-labs-ee-1$ podman pull ghcr.io/mysidlabs/panlab-ee-1

Trying to pull ghcr.io/mysidlabs/panlab-ee-1:latest...

Getting image source signatures

Copying blob 01530b510795 done

Copying blob 894369c521e6 done

<< Output Omitted >>

siduser101@jump:~/pan-labs-ee-1$ cd project/

siduser101@jump:~/pan-labs-ee-1/project$

Step 6 - Run Ansible Navigator

siduser101@jump:~/pan-labs-ee-1/project$ ansible-navigator --mode interactive

```
Welcome
──────────────────────────────────────────────────────────────────────────────

Some things you can try from here:
- :collections                              Explore available collections
- :config                                   Explore the current ansible configuration
- :doc <plugin>                             Review documentation for a module or plugin
- :help                                     Show the main help page
- :images                                   Explore execution environment images
- :inventory -i <inventory>                 Explore an inventory
- :log                                      Review the application log
- :lint <file or directory>                 Lint Ansible/YAML files (experimental)
- :open                                     Open current page in the editor
- :replay                                   Explore a previous run using a playbook artifact
- :run <playbook> -i <inventory>            Run a playbook in interactive mode
- :settings                                 Review the current ansible-navigator settings
- :quit                                     Quit the application

happy automating,

-winston
```

Commands to try from within the ansible-navigator interface

:images

	Image	Tag	Execution environment	Created	Size
0	ee-supported-rhel8	latest	True	2 weeks ago	1.68 GB
1	panlab-ee-1	latest	True	2 days ago	2.09 GB

Type 1 to select the panlab-ee-1 image

	Image: panlab-ee-1:latest (primary)	Description
0	Image information	Information collected from image inspection
1	General information	OS and python version information
2	Ansible version and collections	Information about ansible and ansible collections
3	Python packages	Information about python and python packages
4	Operating system packages	Information about operating system packages
5	Everything	All image information

Type the #2 to review the Ansible version and collection

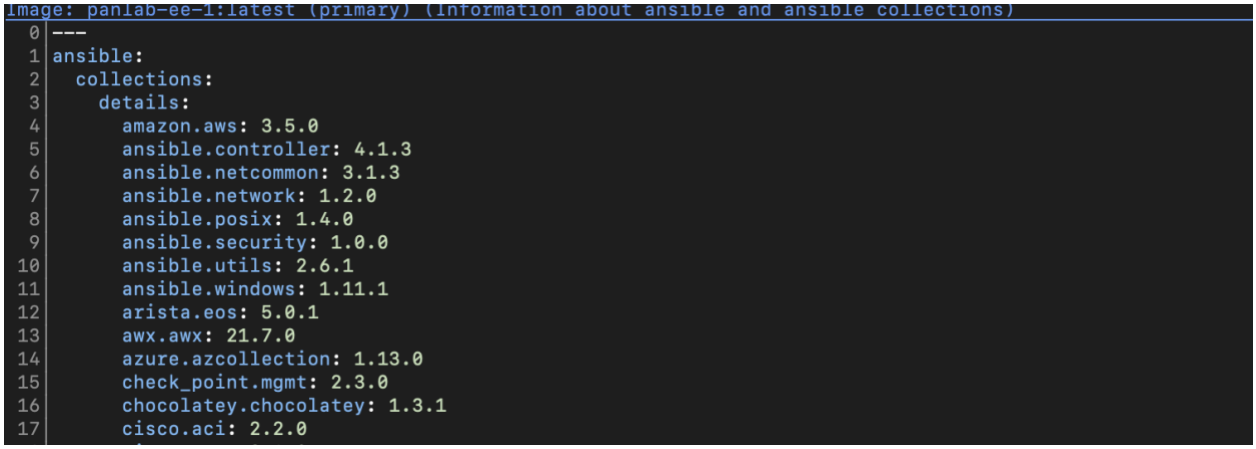


Image: panlab-ee-1:latest (primary) (Information about ansible and ansible collections)

```
  0 ---
  1 ansible:
  2   collections:
  3     details:
  4       amazon.aws: 3.5.0
  5       ansible.controller: 4.1.3
  6       ansible.netcommon: 3.1.3
  7       ansible.network: 1.2.0
  8       ansible.posix: 1.4.0
  9       ansible.security: 1.0.0
 10       ansible.utils: 2.6.1
 11       ansible.windows: 1.11.1
 12       arista.eos: 5.0.1
 13       awx.awx: 21.7.0
 14       azure.azcollection: 1.13.0
 15       check_point.mgmt: 2.3.0
 16       chocolatey.chocolatey: 1.3.1
 17       cisco.aci: 2.2.0
```

Escape to exit the screen

Type 0 to review the image Information

Image: panlab-ee-1:latest (primary) (OS and python version information)

```yaml
---
friendly:
  details: |-
    Red Hat Enterprise Linux release 8.6 (Ootpa)
os:
  details:
  - redhat-bugzilla-product: Red Hat Enterprise Linux 8
    redhat-bugzilla-product-version: '8.6'
    redhat-support-product: Red Hat Enterprise Linux
    redhat-support-product-version: '8.6'
  - ansi-color: 0;31
    bug-report-url: https://bugzilla.redhat.com/
    cpe-name: cpe:/o:redhat:enterprise_linux:8::baseos
    documentation-url: https://access.redhat.com/documentation/red_hat_enterprise_linux/8/
    home-url: https://www.redhat.com/
    id: rhel
    id-like: fedora
    name: Red Hat Enterprise Linux
    platform-id: platform:el8
    pretty-name: Red Hat Enterprise Linux 8.6 (Ootpa)
    version: 8.6 (Ootpa)
    version-id: '8.6'
python:
  details:
    version: 3.8.12 (default, Sep 16 2021, 10:46:05)  [GCC 8.5.0 20210514 (Red Hat
      8.5.0-3)]
```

Lab Prep

Step 1 - Review the variables file:

siduser101@jump:~/pan-labs-ee-1/project$ cat group_vars/all.yml

```
provider:
  username: 'siduser101'
  password: 'Ans1ble!'
  ip_address: siduser101.pan.mysidlabs.com
```

Step 2 - Change the variables to match the user you setup in Part 1B

siduser101@jump:~/pan-labs-ee-1/project$ nano group_vars/all.yml

Lab 1.1: Gather Facts from your Palo Alto Firewall

Lab Goals:

1. Show how to gather facts from a Palo Alto firewall. It will introduce the use of roles which will be used for most playbooks in this lab.
2. We will review the JSON output from the panos_facts module and show how to filter and display specific information.

Step 1 – Review the first playbook
Look at the file (ansible playbook) called 1.1_palo_facts.yml.

siduser101@jump:~/pan-labs-ee-1/project$ cat 1.1_palo_facts.yml

```
---
- hosts: localhost
  connection: local
  gather_facts: False

  tasks:
    - include_role:
        name: palo_facts
...
```

Step 2 – Take a look at the roles playbook for gathering Palo Alto firewall facts
Ansible playbooks are YAML files. YAML is a structured encoding format that is also extremely human readable. In this case we are using “include_role:” explor the role palo_facts.

siduser101@jump:~/pan-labs-ee-1/project tree roles/palo_facts/

```
siduser150@toolkit ~/ansible-pan-labs # tree roles/palo_facts/
roles/palo_facts/
├── README.md
└── tasks
    ├── main.yml
    ├── palo_facts.yml
    └── palo_vr_facts.yml

1 directory, 4 files
```

siduser101@jump:~/pan-labs-ee-1/project$ cat roles/palo_facts/tasks/main.yml
<< output omitted >>
siduser101@jump:~/pan-labs-ee-1/project$ cat roles/palo_facts/tasks/palo_facts.yml
<< output omitted >>

Tip	Notice the FQCN or fully qualified collection name paloaltonetworks.panos.panos_facts. This format is required when using roles with collections that are not installed in engine by default such as Palo Alto collections used in this lab.

siduser101@jump:~/pan-labs-ee-1/project$ cat roles/palo_facts/tasks/palo_vr_facts.yml
<< output omitted >>

Step 3 – Run the gather facts playbook
Run the playbook:
siduser101@jump:~/pan-labs-ee-1/project$ ansible-navigator run 1.1_palo_facts.yml
<< output omitted >>

Step 4 – Review the output of the playbook
Review the output.

Can you think of some use cases for this role?

Lab 1.2: Preform a Palo Alto Firewall Base Configuration

Lab Goals:

1. Review and use role variables in a role
2. Complete the initial configuration setup using Ansible for a Palo Alto firewall.

Step 1 – Review the Palo Alto Initial Setup Ansible Playbook
Review the file called 1.2_palo_initial_setup.yml

siduser101@jump:~/pan-labs-ee-1/project$ cat 1.2_palo_initial_setup.yml
<< output omitted >>

explore the role
siduser101@jump:~/pan-labs-ee-1/project$ tree roles/initial_palo_setup/
<< output omitted >>

Note: Notice that there is a vars directory. This is known as roles vars

siduser101@jump:~/pan-labs-ee-1/project$ cat roles/initial_palo_setup/vars/main.yml
<< output omitted >>

Tip	When a variable file exists in the roles directory structure in the vars folder and is called “main.yml” it will be called into the playbook by default, there is no need to reference a specific vars file in the playbook. This is useful because all the vars required for the specific role are located with the role. This can make modular playbook design using roles easier.

Step 2 – Run the Initial Setup Playbook
Run the playbook:
siduser101@jump:~/pan-labs-ee-1/project$ ansible-navigator run 1. 2_palo_initial_setup.yml
<< output omitted >>

Step 3 – Verify the Palo Alto Configuration
Verifying that the playbook did what you expected. Login to the Palo Alto with your web browser to see what was configured. https://siduser<<ID>>.pan.mysidlabs.com

Lab 1.3: Add logging server profile

Lab Goals:

1. Add a logging server profile to the firewall

Step 1- Review the Palo Alto Logging Setup Ansible Playbook
Look at the file called 1.3_logging_setup.yml

siduser101@jump:~/pan-labs-ee-1/project$ cat 1.3_logging_setup.yml
<< output omitted >>

siduser101@jump:~/pan-labs-ee-1/project$ tree roles/palo_log_server/
<< output omitted >>

siduser101@jump:~/pan-labs-ee-1/project$ tree roles/palo_syslog/
<< output omitted >>

Note: Use GitHub to look at the roles and roles vars

Step 2 – Run the Palo Alto Logging Setup Playbook
Run the playbook:
siduser101@jump:~/pan-labs-ee-1/project$ ansible-navigator run 1.3_logging_setup.yml
<< output omitted >>

Step 3 – Complete Verification
Verifying that the playbook did what you expected. Login to the Palo Alto with your web browser to see what was configured.

Lab 1.4: Add a Simple Security Rule

Note: This lab is required prior to lab 1.5. It adds a deny all rule to the end of the security policy which is referenced in the next playbook.

Lab Goals:

1. Add a Deny All rule to the end of the security policy.

Step 1 – Review the Playbook
Look at the file called 1.4_simple_security_rule_add.yml

siduser101@jump:~/pan-labs-ee-1/project$ cat 1.4_simple_security_rule_add.yml

Note: This playbook does not use a role. We did this to illustrate how to use collections in a playbook where roles are not included.

Step 2 – Run the Playbook
Run the playbook:

siduser101@jump:~/pan-labs-ee-1/project$ # ansible-navigator run 1.4_simple_security_rule_add.yml

<< output omitted >>

Step 3 – Complete Verification
Verifying that the playbook did what you expected. Login to the Palo Alto with your web browser to see what was configured.

Lab 1.5: Adding Bulk Security Rules from a CSV file

Lab Goals:

1. Use a CSV file to add a bulk set of security rules to the firewall

Step 1 - Review the Playbook
Look at the file called 1.5_add_rules_csv.yml

siduser101@jump:~/pan-labs-ee-1/project$ cat 1.5_add_rules_csv.yml

Step 2- Review the Playbook Files
Use cat or github to review the roles tasks, roles variables and the roles files.

Note: This playbook pulls variables from a CSV file.

Step 3 – Run the Playbook
Run the playbook:

siduser101@jump:~/pan-labs-ee-1/project$ ansible-navigator run 1.5_add_rules_csv.yml

<< output omitted >>

Step 4 - Complete Verification
Verifying that the playbook did what you expected. Login to the Palo Alto with your web browser to see what was configured.

Bonus Task!
Edit the CSV file and add additional rules by running the playbook again.

Lab 1.6: Remove Address Object

Lab Goals:

1. Remove an address object from the firewall.

Step 1 – Review the Playbook

Look at the file called 1.6_remove_address_object.yml

siduser101@jump:~/pan-labs-ee-1/project$ cat 1.6_remove_address_object.yml

Step 2 - Review the Playbook Associated Files

Use cat or github to review the roles tasks and roles variables

Step 3 - Edit Variable File

Modify the variable file to a server of your choosing using nano, vi or github

Example

```
remove_addr: 'server_6'
```

Step 4 – Run the Playbook

Run the playbook:

siduser101@jump:~/pan-labs-ee-1/project$ ansible-navigator run 1.6_remove_address_object.yml

<< output omitted >>

Step 5 – Complete Playbook Verification

Verifying that the playbook did what you expected. Login to the Palo Alto with your web browser to see what was configured.

Why is the rule greyed out?

Can you think of some use cases for this role?

Lab 1.7: Backup Palo Alto Firewall Configuration

Lab Goals:

1. Complete a backup of the current Palo Alto configuration.

Step 1 - Review the Playbook
Look at the file called 1.7_palo_backup_configuration.yml

siduser101@jump:~/pan-labs-ee-1/project$ cat 1.7_palo_backup_configuration.yml

Step 2 - Review the Playbook Associated Files
Review the roles tasks and roles variables with cat or GitHub. The variable file should look like below.

```
---
## Be sure path is writable
backup_path : '~/ansible-pan-labs/'
...
```

Step 3 – Run the Backup Playbook
Run the playbook:

siduser101@jump:~/pan-labs-ee-1/project$ ansible-navigator run 1.7_palo_backup_configuration.yml

<< output omitted >>

Step 4 – Review the Backup File

Ensure that the backup file exists. Copy the backup file name

Can you think of some use cases for this role?

Lab 1.8: Restore the configuration

Lab Goals:

1. Restore a previously backed up configuration to the firewall.

Step 1 - Review the Playbook
Look at the file called 1.8_palo_restore_configuration.yml

siduser101@jump:~/pan-labs-ee-1/project$ cat 1.8_palo_restore_configuration.yml

Step 2 - Review the Playbook Associated Files
Review the roles variables with cat or github. The variable file should look like below.

```
---
backup_path : 'home/siduser<ID>/ansible-pan-labs/'
restore_file: 'backup-2021-01-01-00-01.xml'
...
```

Edit the backup_path variable to have your siduser ID.
Edit the restore_file variable to be your saved backup file name.

Step 3 – Run the Restore Playbook
Run the playbook:

siduser101@jump:~/pan-labs-ee-1/project$ ansible-navigator run 1.8_palo_restore_configuration.yml

<< output omitted >>

Success – Congratulations for Completing!

Key Lab Takeaways

1. Remember YAML is very sensitive to correct indentation.
2. The use of an Ansible role is best practice when there is a well-defined scope with a high possibility of re-use.
3. Plan to the location of variables carefully. Never have a single variable in more than one location.
4. If you copy and paste text for a playbook you may get indentation issues. Ansible provides a simple syntax checker, try ansible-playbook --syntax-check backup.yml to verify. A Best Practice is to use a linter, for example ansible-review.
5. Ansible provides excellent online documentation, which is also available from the command line, for example ansible-doc ansible.netcommon.cli_command. For a full list of modules try ansible-doc –-list

Appendix A:

Useful resource links and information

Links:

Ansible Best Practices:
https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html

Variable precedence:
https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#variable-precedence-where-should-i-put-a-variable

Ansible Modules
https://docs.ansible.com/ansible/2.9/modules/modules_by_category.html

Ansible Galaxy
https://galaxy.ansible.com/home

Ansible Palo Alto Module Reference
https://ansible-pan.readthedocs.io/en/latest/modules/index.html

Ansible Navigator Creator Guide
https://access.redhat.com/documentation/en-us/red_hat_ansible_automation_platform/2.1/html/ansible_navigator_creator_guide/index