Tenable-Deployment_Production

1. Tasks

a. tenable-linux

```yaml
- block:
  - name: Executing script copy_linux
    script: ../scripts/linux-copy-install.sh
    become: yes
    register: script_output
    environment:
      nas_path: '{{ nas_path_linux }}'
      mount_username: '{{ mount_username }}'
      mount_password: '{{ mount_password }}'
 always:
  - name: Script output
    debug:
      msg: "{{ script_output }}"
```

- **block:** This is a keyword that indicates the beginning of a block of tasks that will be executed together. In this case, there is only one task in the block.
- **name: Executing script copy_linux:** This is the name of the task. It helps to identify the task being executed.
- **script: ../scripts/linux-copy-install.sh:** This is the script that will be executed on the remote Linux machine. The script path is ../scripts/linux-copy-install.sh.
- **become: yes:** This is a keyword that tells Ansible to become the superuser or an equivalent privileged account to run this task.
- **register: script_output:** This is a keyword that stores the output of the task execution in the variable script_output. This variable can be used later in the playbook.
- **environment:** This is a keyword that sets the environment variables for the task.
- **nas_path: '{{ nas_path_linux }}':** This sets the value of the environment variable nas_path to the value of the Ansible variable nas_path_linux.
- **mount_username: '{{ mount_username }}':** This sets the value of the environment variable mount_username to the value of the Ansible variable mount_username.
- **mount_password: '{{ mount_password }}':** This sets the value of the environment variable mount_password to the value of the Ansible variable mount_password.
- **always:** This is a keyword that contains tasks that will always be executed, regardless of whether the previous task succeeds or fails.
- **name: Script output:** This is the name of the task.
- **debug:** This is a keyword that prints a message to the console.
- **msg: "{{ script_output }}":** This is the message that will be printed to the console. It contains the value of the script_output variable that was registered earlier.

b. tenable-windows

```yaml
- block:
  - name: Executing script windows
    script: ../scripts/windows-copy-install.ps1
    become: yes
    register: script_output
    environment:
      nas_path: '{{ nas_path_win }}'
      mount_username: '{{ mount_username }}'
      mount_password: '{{ mount_password }}'
 always:
  - name: Script output
    debug:
      msg: "{{ script_output }}"
```

- **block:** This is a block of tasks that will be executed together. In this case, it contains only one task.
- **name:** Executing script windows: This is the name of the task. It helps to identify the task being executed.
- **script: ../scripts/windows-copy-install.ps1:** This is the script that will be executed on the remote Windows machine. The script path is ../scripts/windows-copy-install.ps1.
- **become: yes:** This is a keyword that tells Ansible to become the superuser or an equivalent privileged account to run this task.
- **register: script_output:** This is a keyword that stores the output of the task execution in the variable script_output. This variable can be used later in the playbook.
- **environment:** This is a keyword that sets the environment variables for the task.
- **nas_path: '{{ nas_path_win }}':** This sets the value of the environment variable nas_path to the value of the Ansible variable nas_path_win.
- **mount_username: '{{ mount_username }}':** This sets the value of the environment variable mount_username to the value of the Ansible variable mount_username.
- **mount_password: '{{ mount_password }}':** This sets the value of the environment variable mount_password to the value of the Ansible variable mount_password.
- **always:** This is a keyword that contains tasks that will always be executed, regardless of whether the previous task succeeds or fails.
- **name:** Script output: This is the name of the task.
- **debug:** This is a keyword that prints a message to the console.
- **msg: "{{ script_output }}":** This is the message that will be printed to the console. It contains the value of the script_output variable that was registered earlier.

2. Vars

a. agent-installation-config

```
agent_installer:
  windows:
    rsa64:
      nas_path: '\\iaas.hosted.lac.com\iaas'
      local_mount_path: "c:\\temp\\rsa\\"
      team_dl: ecloud@isd.lacounty.gov
    nxlog:
      nas_path: '\\iaas.hosted.lac.com\iaas'
      local_mount_path: 'c:\temp\nxlog\'
      team_dl: ecloud@isd.lacounty.gov
  linux:
    secureworks_read_cloak:
      nas_path: //iaas.hosted.lac.com/iaas/SelfServiceImages/RedCloak
      local_mount_path: /myshare
      team_dl: ecloud@isd.lacounty.gov
    macfee:
      nas_path: ""
      local_mount_path: /tmp/mcafee
      team_dl: ecloud@isd.lacounty.gov
    commvault_backup:
      nas_path: "//isdowfsv01.hosted.lac.com/public/Software/Commvault/tar_05242019"
      local_mount_path: /myshare
      team_dl: ecloud@isd.lacounty.gov
    tetration:
      nas_path: "//isdowfsv01.hosted.lac.com/public/Software/Tetration"
      local_mount_path: /myshare
      team_dl: ecloud@isd.lacounty.gov
```

- **agent_installer:** This is the name of the dictionary.
- windows: This is a key in the dictionary that defines a sub-dictionary for Windows-based installers.
- **rsa64:** This is a key in the sub-dictionary that defines the RSA64 installer.
- **nas_path: '\iaas.hosted.lac.com\iaas':** This is the network addressable storage (NAS) path for the RSA64 installer.
- **local_mount_path: "c:\temp\rsa\":** This is the local path on the Windows machine where the RSA64 installer will be mounted.
- **team_dl: ecloud@isd.lacounty.gov:** This is the email address of the team responsible for downloading the installer.
- **nxlog:** This is a key in the sub-dictionary that defines the NXLog installer.
- **nas_path: '\iaas.hosted.lac.com\iaas':** This is the NAS path for the NXLog installer.
- **local_mount_path: 'c:\temp\nxlog':** This is the local path on the Windows machine where the NXLog installer will be mounted.
- **team_dl: ecloud@isd.lacounty.gov:** This is the email address of the team responsible for downloading the installer.

- **linux:** This is a key in the dictionary that defines a sub-dictionary for Linux-based installers.
- **secureworks_read_cloak:** This is a key in the sub-dictionary that defines the SecureWorks Read Cloak installer.
- **nas_path: //iaas.hosted.lac.com/iaas/SelfServiceImages/RedCloak:** This is the NAS path for the SecureWorks Read Cloak installer.
- **local_mount_path: /myshare:** This is the local path on the Linux machine where the SecureWorks Read Cloak installer will be mounted.
- **team_dl: ecloud@isd.lacounty.gov:** This is the email address of the team responsible for downloading the installer.
- **macfee:** This is a key in the sub-dictionary that defines the McAfee installer.
- **nas_path: "":** This is an empty NAS path because the McAfee installer is not stored on a NAS.
- **local_mount_path: /tmp/mcafee:** This is the local path on the Linux machine where the McAfee installer will be mounted.
- **team_dl: ecloud@isd.lacounty.gov:** This is the email address of the team responsible for downloading the installer.
- **commvault_backup:** This is a key in the sub-dictionary that defines the Commvault backup installer.
- **nas_path: "//isdowfsv01.hosted.lac.com/public/Software/Commvault/tar_05242019":** This is the NAS path for the Commvault backup installer.
- **local_mount_path: /myshare:** This is the local path on the Linux machine where the Commvault backup installer will be mounted.
- **team_dl: ecloud@isd.lacounty.gov:** This is the email address of the team responsible for downloading the installer.
- **tetration:** This is a key in the sub-dictionary that defines the Tetration installer.
- **nas_path: "//isdowfsv01.hosted.lac.com/public/Software/Tetration":** This is the NAS path for the Tetration installer.
- **local_mount_path: /myshare:** This is the local path on the Linux machine where the Tetration installer will be mounted.
- **team_dl: ecloud@isd.lacounty.gov:** This is the email address of the team responsible for downloading the installer.

b. mount-cred

```
$ANSIBLE_VAULT;1.1;AES256
646633386138336132343436656433353439613737306464393130663534353633626162643039313462313063383238666234383633383934353361303230360a366564383338386261396666646166366461623662303039343437386534313437353937653536633132383965343933343732333038636364353030383539660a326230353663316134333762623764616430343636613766323164333133346265313532643031373561663666313339333931303965613733333266326435643935313361303562636361393862303161383062383164303364666438336161383363633363383336266653863303234663263376133393036373166373637346138323364653461383330356366373839303530383361653230653164313761313161333130643439366464383065383161313564323930386239623430376130336131323835343031613432656162316665376262663534396132313237646137303736383034353963323964343065336230366264646138646664326 3
```

Each line beginning with **'$ANSIBLE_VAULT'** indicates the beginning of an encrypted section.

The long string of characters following the header is the encrypted content.

Without the decryption key, it is not possible to decrypt and understand the contents of this file.

c. vcenter-cred-prod

```
$ANSIBLE_VAULT;1.1;AES256
6365326163373433363636633531383131623466346232313166323231336166653033383438356
6
333663336635396135363432333832616232323935323532 0a6535383163663935393335393466 3
0
633463386661653166653932623966663238383035343035613232353065393238303563316462 3
5
```

```
3333616138306532650a656462653632633336363633616339613634303232383933383465323734
343831373032643062656231666135613834336463396238393037656166353638666664313631366
313161326265343332373634376433643266623665653632363763343261323066316366373662665
313935356664316464646262633135336238313865353737343936626633653865316466363737331
313861393132353931366634653666313966303066616162343732303634653036336664633939330
366139343261393764336133363931373932383036316434343035363832333733383331663338330
653430633833386232303736363038353638313961353435383130653532373736663037623231331
666165303433663731626433626161303266393635386239626433396661386263303632633533338
623031396661393365376433666539353765613830393164623834616332356566333163363238655
386131303738376166623436353964666136656438313435363831636536303635373866633033633
666632323939343330666261336434373361616539636365323333313564323637373565653766346
663839343765303934316538386539396534656638346433396465316539666336353332636364393
356236636139623334633838373162653533383230353565656339333661623638613864633264337
3866
```

The code you provided is an encrypted Ansible Vault file, not executable code. It starts with the string **$ANSIBLE_VAULT;1.1;AES256**, which indicates the version of the Ansible Vault file and the encryption algorithm used.

The rest of the code is a long string of characters, which is the encrypted content of the Ansible Vault file. Without the encryption key, it is not possible to decode this string and determine its contents.

In summary, this code is not something that can be explained line by line, as it is an encrypted file.

d. vcenter-cred-test

```
$ANSIBLE_VAULT;1.1;AES256
363862623132623434303136383364393662383039383431653831393766303461353335353163339
336633323539336633626438633464663665326436323131 0a353866356138376237653630316562
```

```
3161623931623631613833303864386436306265363961616161353931353762326336653331346
6
353362656537336265 0a3138653166396235366434393062623837643832353839633532646537 3
6
393365393430653130646134386537663935356264313135326462656561653264313730666430 3
1
613432313934643066366565663330656538613561343332626662616539636166623161333161 6
4
643361303635656462626365313366366664666164386134326235306339616437353537306265 6
5
653139653233623335326638336662336263636439363262623562643839373061616466653262 3
6
613834663235363633316363316566393337623163386162393731616431366162656366333433 6
5
373364336437393939643433303564616239323732636138303533363637333332653438303533 6
1
396233653135356234633162646366343363633364356537313464373066313062386662323062 3
9
386261343836366130613432303866313131663638323330316165653936303637613164323534 6
3
663462306633393039306430316163616535336330646464366139666338336531633266323664 3
0
383632343338356466333266336233313931646361626265637376562613734353031633237343 6
7
646564663562633665383636643434303661383066343531616633336162626236333861373265 3
0
393530363138636163333435323737376434393634323231303535353633383765333261383134 3
1
3665
```

However, based on the file format and the presence of the string **$ANSIBLE_VAULT**, it can be inferred that this file contains sensitive data that has been encrypted for secure storage and transmission. Ansible Vault is a built-in tool in Ansible that allows users to encrypt sensitive data within their playbooks and configuration files.

If you have the encryption key for this file, you can use the **ansible-vault** command to decrypt and view the contents of the file.

e. win-connection-vars

```
ansible_connection: vmware_tools
ansible_shell_type: powershell
ansible_vmware_validate_certs: false
```

- **ansible_connection: vmware_tools:** This sets the connection type that Ansible will use to communicate with the target machine. In this case, it's **vmware_tools**, which means Ansible will use the VMware Tools API to communicate with a VMware virtual machine.

- **ansible_shell_type: powershell:** This sets the shell type that Ansible will use when running shell commands on the target machine. In this case, it's **powershell,** which means Ansible will use PowerShell as the shell.
- **ansible_vmware_validate_certs: false:** This sets whether or not Ansible should validate SSL certificates when communicating with the VMware server. In this case, it's set to **false,** which means that Ansible will not validate SSL certificates. This is generally not recommended for security reasons, but may be necessary in certain environments where SSL certificates are not properly configured.

3. tenable-installer

```
- name: Ansible Playbook to install Tenable
  hosts: all
  gather_facts: no
  vars_files:
    - ./vars/mount-cred.yaml
    - ./vars/vcenter-cred-{{ ansible_vmware_env| default('test') }}.yaml
  vars:
    - os_type: "{{ 'windows' if ansible_vmware_guest_os_family in ['windowsGuest','WIN'] else
'linux' }}"
    - agent_name: tenable
    - nas_path_linux: '\\eisstorage.hosted.lac.com'
    - nas_path_win: '\\eisstorage.hosted.lac.com'
  tasks:
    - name: This is for Linux
      when: ansible_os_family == 'linux'
      include: tasks/tenable-linux.yaml

    - name: This is for windows
      when: ansible_os_family == 'windows'
      include: tasks/tenable-windows.yaml
```

- This is an Ansible playbook **named "Ansible Playbook to install Tenable"**.
- It targets **all hosts** in the inventory.
- **"gather_facts: no"** means that Ansible will not gather facts about the target hosts before running the playbook.
- **"vars_files"** specifies the list of files containing variables used in the playbook.
- The first file is **"mount-cred.yaml" located in the "./vars" directory**.
- The second file is **"vcenter-cred-{{ ansible_vmware_env| default('test') }}.yaml".** The value of **"ansible_vmware_env"** will be used to replace **"{{ ansible_vmware_env| default('test') }}"** in the filename. If **"ansible_vmware_env"** is not defined, it will be replaced by the string "test".
- **"vars"** specifies a list of variables to be used in the playbook.
- **"os_type"** is set to **"windows"** if the target **OS** is **Windows**, otherwise it is set to **"linux".**
- **"agent_name"** is set to **"tenable".**

- **"nas_path_linux"** is set to **"\\eisstorage.hosted.lac.com".**
- **"nas_path_win"** is set to **"\\eisstorage.hosted.lac.com".**
- **"tasks"** specifies the list of tasks to be executed in the playbook.
- The first task is named **"This is for Linux"**. It will only be executed if the target **OS** family is **Linux**.
- The task is included from the file **"tasks/tenable-linux.yaml".**
- The second task is named **"This is for windows"**. It will only be executed if the target **OS** family is **Windows.**
- The task is included from the file **"tasks/tenable-windows.yaml".**