Predict | Protect | Prevent

ARCON|PAM

Ansible

arcon

Table of Contents

1 Overview

Creating a continuous delivery pipeline which necessitates the participation of several teams. DevOps Engineers won't be able to do that without a straightforward automation tool that anyone in the company can use. Ansible Playbooks guarantee that the systems are correctly deployed and managed at all times.

To securely retrieve the secrets vaulted in ARCON PAM Vault to execute the privileged tasks in Ansible, ARCON has created a lookup plugin to integrate with the Ansible Delivery tool. Ansible-specific additions to the Jinja2 templating language are lookup plugins. Inside your playbooks, you can use lookup plugins to access data from external sources such as ARCON PAM Vault.

2 Prerequisites

1. Ansible version 2.9.15 on Server running Ansible
2. Python installed on Server running Ansible
3. API credentials of API user created in PAM (API User Registration in ACMO)

3 Configuration Steps

1. The credentials required to authenticate to the ARCON API are present in a file config.json. This is the API username and password that were used while creating an API user in PAM, as well as the API URL. Place it in the directory - /usr/share/ansible/plugins/lookup

config.json file format:

hostipHA - API URL with port

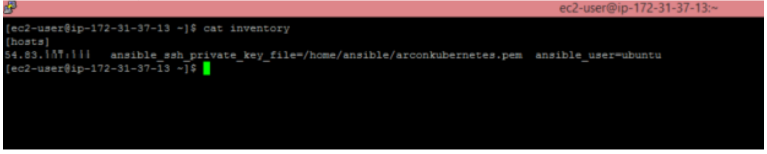
Username - base64 encoded API username

Password - base64 encoded API password

```
{
    "hostipHA":"",
    "username":"",
    "password":""
}
```

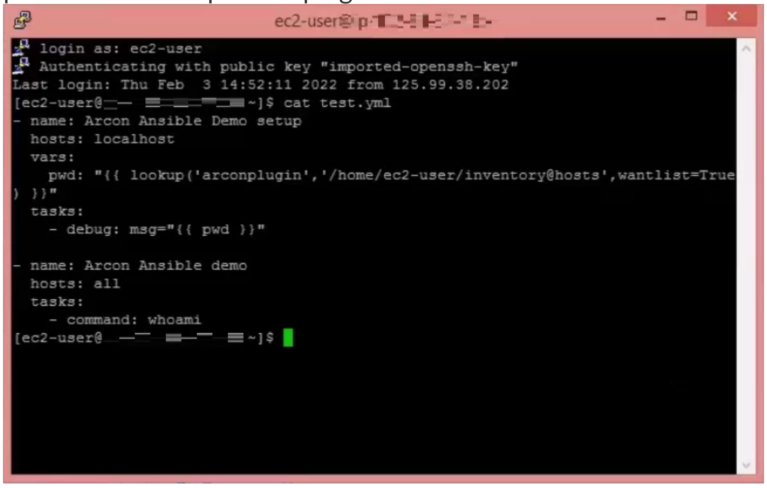
2. Place the arconplugin.py file at /usr/share/ansible/plugins/lookup

3. The inventory file contains all the nodes to which the ansible server wants to push the configuration to. It has the IP addresses of the node and the local location of the key file for the node separated line-wise. The Arcon Ansible plugin uses this file for server IPs and usernames.



```
[ec2-user@ip-172-31-37-13 ~]$ cat inventory
[hosts]
54.83.[redacted]   ansible_ssh_private_key_file=/home/ansible/arconkubernetes.pem  ansible_user=ubuntu
[ec2-user@ip-172-31-37-13 ~]$
```

4. There are two parts in the playbook, vars, and hosts. Vars are the place where the passwords and the server details are present and tasks which are to be performed after connecting with the node. In the pwd vars, we provide the lookup to our plugin.



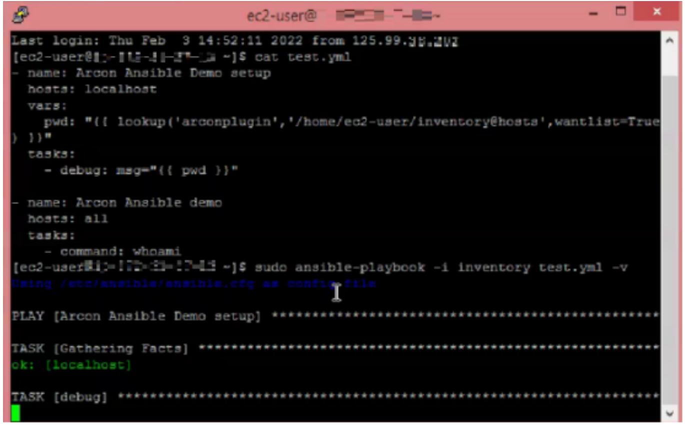
```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu Feb  3 14:52:11 2022 from 125.99.38.202
[ec2-user@[redacted] ~]$ cat test.yml
- name: Arcon Ansible Demo setup
  hosts: localhost
  vars:
    pwd: "{{ lookup('arconplugin','/home/ec2-user/inventory@hosts',wantlist=True
) }}"
  tasks:
    - debug: msg="{{ pwd }}"

- name: Arcon Ansible demo
  hosts: all
  tasks:
    - command: whoami
[ec2-user@[redacted] ~]$
```

5. Sample code to add in the playbook for the lookup to ARCON PAM plugin.

```
- name: Arcon Ansible Demo setup
  hosts: localhost
  vars:
    pwd: "{{ lookup('arconplugin','/home/ec2-user/inventory@hosts',wantlist=True) }}"
  tasks:
    - debug: msg="{{ pwd }}"
```

6. When the playbook runs, firstly Arcon's ansible plugin fetches the required passwords from the ARCON PAM, connects to the node, and then performs the tasks.



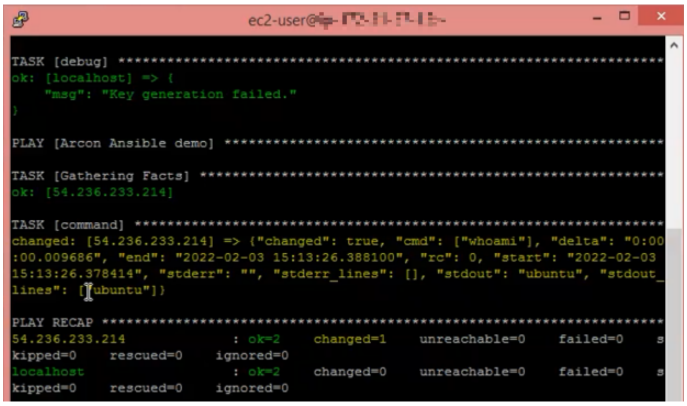
```
Last login: Thu Feb  3 14:52:11 2022 from 125.99.[redacted]
[ec2-user@[redacted] ~]$ cat test.yml
- name: Arcon Ansible Demo setup
  hosts: localhost
  vars:
    pwd: "{{ lookup('arconplugin','/home/ec2-user/inventory@hosts',wantlist=True
) }}"
  tasks:
    - debug: msg="{{ pwd }}"

- name: Arcon Ansible demo
  hosts: all
  tasks:
    - command: whoami
[ec2-user[redacted] ~]$ sudo ansible-playbook -i inventory test.yml -v
Using /etc/ansible/ansible.cfg as config file

PLAY [Arcon Ansible Demo setup] ************************************************

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

TASK [debug] *******************************************************************
```



```
TASK [debug] *******************************************************************
ok: [localhost] => {
    "msg": "Key generation failed."
}

PLAY [Arcon Ansible demo] ******************************************************

TASK [Gathering Facts] *********************************************************
ok: [54.236.233.214]

TASK [command] *****************************************************************
changed: [54.236.233.214] => {"changed": true, "cmd": ["whoami"], "delta": "0:00
:00.009686", "end": "2022-02-03 15:13:26.388100", "rc": 0, "start": "2022-02-03
15:13:26.378414", "stderr": "", "stderr_lines": [], "stdout": "ubuntu", "stdout_
lines": ["ubuntu"]}

PLAY RECAP *********************************************************************
54.236.233.214             : ok=2    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
localhost                  : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

Additional security can be applied by adding the Users running Ansible in a User group on Linux and giving the access permissions of Ansible directories (ansible/plugins/lookup) to only that user group, so that the config and plugin files will not be exposed to the other users in the system.

Privileged Access Management Suite

arcon

Predict | Protect | Prevent