



# ansible



1. setup.sh ansibleyq

```
# /bin/bash

# ansible
yum install ansible-2.9.27-1.el7.noarch.rpm -y

# yq
cp -a yq /usr/local/bin/
```

yqyaml

2. inithost, 

```
hosts:
  #192.168.40.36:
  #  ansible_user: root
  #  ansible_password: 123456
  #192.168.40.37:
  #  ansible_user: root
  #  ansible_password: 123456
```

3. ansible hosts

```
# [test]
#192.168.40.36
#
#[test1]
#192.168.40.37
```

4. 

```
bash ./inithost.sh
```

```
#####inithost####
```

```
####
```

1. #####

```
ansible-playbook -i hosts --vault-password-file=.vault_pass -e "target_hosts=192.168.40.36
user=root password=123456" changepasswd.yaml
```

2. ##ansible##

```
ansible-playbook -i hosts --vault-password-file=.vault_pass -e "target_hosts=test user=root
password=123456" changepasswd.yaml
```

3. ##hosts#####

```
ansible-playbook -i hosts --vault-password-file=.vault_pass -e "target_hosts=all user=root
password=123456" changepasswd.yaml
```

4. #####

```
ansible-playbook -i hosts --vault-password-file=.vault_pass -e "target_hosts=192.168.40.36
user=test password=123456" changepasswd.yaml
```

- ##: #####

```
####
```

#####"Using a SSH password instead of a key is not possible because Host Key checking is ena  
sshpas does not support this. Please add this host's fingerprint to your known\_hosts file to  
manage this host."

```
#####ssh#####key[] 1[] inithost.sh[]
```

```
## /bin/bash
```

```
# #####vault_password
vault_password="3d$R7#X9"
```

```

# 创建连接目录
mkdir -p connect

# 获取主机列表
hosts=$(yq e '.hosts | keys' inithost.yaml | sed 's/- //' )

# 遍历主机列表
for host in $hosts; do
    # 获取主机IP
    if [ -z "$host" ]; then
        continue
    fi

    # 获取主机配置
    host=${host//\\"/}
    # 从yaml文件中获取ansible_user和ansible_password
    user=$(yq e ".hosts.\"$host\".ansible_user" inithost.yaml)
    password=$(yq e ".hosts.\"$host\".ansible_password" inithost.yaml)

    # 检查用户和密码是否为空
    if [ -z "$user" ] || [ -z "$password" ]; then
        continue
    fi

    # 对密码进行加密
    user=${user//\\"/}
    password=${password//\\"/}

    # 将配置写入yaml文件
    # 从yaml文件中获取IP
    # 创建连接目录
    echo "ansible_user: $user" > connect/$host.yaml
    echo "ansible_password: $password" >> connect/$host.yaml

    # 加密密码
    echo "$vault_password" > .vault_pass
    # 使用ansible-vault加密密码
    echo $vault_password | ansible-vault encrypt connect/$host.yaml --vault-password-
file=.vault_pass
    # 删除临时文件
    # rm -fr .vault_pass
done

```

## 2 changepasswd.yaml playbook

```
- hosts: "{{ target_hosts }}"
vars_files:
  - "connect/{{ inventory_hostname }}.yaml"
tasks:
  - name: debug
    debug:
      msg: "connect/{{ inventory_hostname }}.yaml"

  - name: [ ]
    user:
      name: "{{ user }}"
      password: "{{ password | password_hash('sha512') }}"
    become: yes
    become_user: "{{ ansible_user }}"

  - name: [ansible][root][ ]
    local_action:
      module: script
      args: ansible_rootpasswd_change.sh connect/{{ inventory_hostname }}.yaml {{ password
    }}

    when: user == 'root'
```

## 3 [root][ ]

```
#!/bin/bash

# [ ] vault [ ]
#vault_password=$1
# [ ] inventory_hostname
vault_file=$1
# [ ] ansible_password
new_ansible_password=$2

# [ ] vault [ ]
#vault_file="${vault_file_path}"

# [ ]
temp_file=$(mktemp)
```

```
# [] ansible-vault [] vault []
#echo $vault_password | ansible-vault decrypt $vault_file --output $temp_file
ansible-vault decrypt $vault_file --output $temp_file --vault-password-file=.vault_pass

# [] ansible_password
yq eval ".ansible_password = \"\$new_ansible_password\"" $temp_file

# [] ansible-vault [] vault []
#echo $vault_password | ansible-vault encrypt $temp_file --output $vault_file
ansible-vault encrypt $temp_file --output $vault_file --vault-password-file=.vault_pass

# []
rm $temp_file
```

---

[] #2

[] 30 [] 2023 03:53:38

[] 15 [] 2025 14:56:20