Documentation › Deploying with Ansible › Remote Hosts

Remote Hosts

Ansible was born with the idea to be an agentless automation platform. Ansible relies on SSH the connection to remote hosts, meaning that, you can connect to remote hosts as SSH does. To follow, we briefly explain two (2) of this methods.

Note

We recommend the use of Using passwords method, this avoid you share your public SSH Key among several hosts.

Using passwords

Ansible does most of the work via SSH, SSH share their authentication mechanisms with Ansible, so, in order to establish a connection with remote hosts, a user/password must be supplied. To follow there is a description of some useful options to use for SSH authentication:

```
-u <user>   Set the connection user.
-k          Ask the password of the connection user.
-b          Execute task and operations with a privilege user.
-K          Ask for sudo password, intended for privilege escalation.
```

You can use the above args as follows:

```
ansible -m setup all -u foo -k -b -K
```

This will set the connection user as `foo`. Also, it will ask for the connection user password and privileged user password.

Windows authentication

Windows hosts use a different mechanism to perform authentication. Please refer to Authentication Options in order to setup the adequate option.

Using SSH key-pairing

You can setup a SSH key-pairing to provide a silent auth mechanism, first create a OpenSSH key-pair:

```
$ ssh-keygen
```

To improve security on this setup, please ensure you provide a password for this key.

```
Enter passphrase (empty for no passphrase): ************
Enter same passphrase again: ************
```

Using ssh-agent, avoid asking the key password over and over again on every Ansible deploy. Ssh-agent will cached your key to be use in further actions, until you logout.

Installing public key

After creating the Control machine key, you need to install the public key into every remote hosts, copy the content of `.ssh/id_rsa.pub` of Control machine to `.ssh/authorized_keys` on your host. Make sure you know the user to store `authorized_keys`, this will be the user you use for any action via Ansible.

Set the correct permissions:

```
$ chmod 600 .ssh/authorized_keys
```

Add hosts to control

Adding hosts is easy, just put the hostname or IP Address on `/etc/ansible/hosts`.

```
$ cat /etc/ansible/hosts

hosts1.example.net
hosts2.example.net
```

Also, you can group hosts. This could be useful to execute tasks and roles to several hosts at once:

```
$ cat /etc/ansible/hosts

[wazuh-elasticsearch]
hosts1.example.net
hosts2.example.net
```

Note

You can see the Ansible inventory documentation for more info regarding hosts and groups.

Test connection

This will attempt a connection with the remote hosts using `ping` module.

```
$ ansible all -m ping
```

You will get a output like this.

```
hosts1.example.net | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
hosts2.example.net | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

If you see the above, then Ansible is fully usable.