

Playbook result - Log individual output

```
---
- hosts: all
#  serial: 1
tasks:
  - name: get files
    shell:
      ls /tmp/
    register: validate

  - name: Write results to logfile
    blockinfile:
      create: yes
      path: "/tmp/Ansible/log"
      insertafter: BOF
      block: "{{ validate.stdout }}"
      marker: "# {{ inventory_hostname }} {mark}"
    delegate_to: localhost
```

```
[root@ip-172-31-27-155 Ansible]# ansible-playbook log_file.yml

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [13.127.244.80]

TASK [get files] *****
changed: [13.127.244.80]

TASK [Write results to logfile] *****
changed: [13.127.244.80 -> localhost]

PLAY RECAP *****
13.127.244.80          : ok=3    changed=2    unreachable=0    failed=0

[root@ip-172-31-27-155 Ansible]# cat log
# 13.127.244.80 BEGIN
ansible__dbGTi
ansible__NN18TU
ansible__QgZAdq
systemd-private-675320f3f1494d95a956ccd518479ab9-chronyd.service-3JA1vI
test.txt
# 13.127.244.80 END
```

Windows Host Connectivity

Pre-requisites:

- It supports both Desktop OSs including Windows 10 and 11, and server OSs including Windows Server 2016, 2019, and 2022.
- PowerShell 5.1 or newer and at least .NET 4.0 on the Windows host
- Activate a WinRM listener
- **Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Force**
- There are two listeners activated. One is listening on port **5985 over HTTP** and the other is listening on port **5986 over HTTPS**
- To view the current listeners that are running on the WinRM service:
winrm enumerate winrm/config/Listener
winrm quickconfig for HTTP or winrm quickconfig -transport:https

[How to configure WINRM for HTTPS - Windows Client | Microsoft Learn](#)

Cont.

Ansible uses the **pywinrm** package to communicate with Windows servers over WinRM. It is not installed by default with the Ansible package.

```
pip install pywinrm
```

```
pip install "pywinrm>=0.3.0"
```

When connecting to a Windows host, there are several different options that can be used when authenticating with an account. The authentication type may be set on inventory hosts or groups with the **ansible_winrm_transport** variable.

Cont.

Option	Local Accounts	Active Directory Accounts	Credential Delegation	HTTP Encryption
Basic	Yes	No	No	No
Certificate	Yes	No	No	No
Kerberos	No	Yes	Yes	Yes
NTLM	Yes	Yes	No	Yes
CredSSP	Yes	Yes	Yes	Yes

Cont.

Configuring Ansible on Windows with basic authentication, set the below connection variables.

ansible_user=Username

ansible_password=Password

ansible_connection=winrm

ansible_winrm_transport=basic

Kerberos is the recommended authentication option to use when running in a domain environment. Kerberos supports features like credential delegation and message encryption over HTTP and is one of the more secure options that is available through WinRM.

ansible_user=username@MY.DOMAIN.COM

ansible_password=Password

ansible_connection=winrm

ansible_port=5985

ansible_winrm_transport=kerberos

Basic Modules for Windows

Installing Software:

- There are three main ways that Ansible can be used to install software:
- Using the **win_chocolatey** module. This sources the program data from the default public Chocolatey repository. Internal repositories can be used instead by setting the source option.
- Using the **win_package** module. This installs software using an MSI or .exe installer from a local/network path or URL.
- Using the **win_command** or **win_shell** module to run an installer manually.