

- IT Automation tool
- Agentless - no need of any kind of agent.
- Works on push method
- Connection via SSH or remote Access (Windows)
- Works in parallel

→ Inventory file → collection of Host it can connect to. default → /etc/ansible/hosts

• ansible_host = Ansible parameter to denote the server.

- ansible_connection:
- ansible_port: 22/5986
- ansible_user: root/administrator
- ansible_ssh_pass: Password

localhost ansible_connection=localhost

* Best Practice is to use ssh key based Authentication.

- Ansible uses port 22 → ssh port
- Ansible supported formats → INI, YAML

- control node - Ansible sv - Linux based. ^{Works on only}
- Managed node - childs where task performed
- Connection plugin - ssh, winsrm
- Inventory system - list of managed node
 - ↳ static inventory
 - Dynamic inventory
- Modules
- Adhoc - command line → Imperative way
 - CLI method is adhoc method used to push
- playbooks - file base push

Ansible inventory file is found in

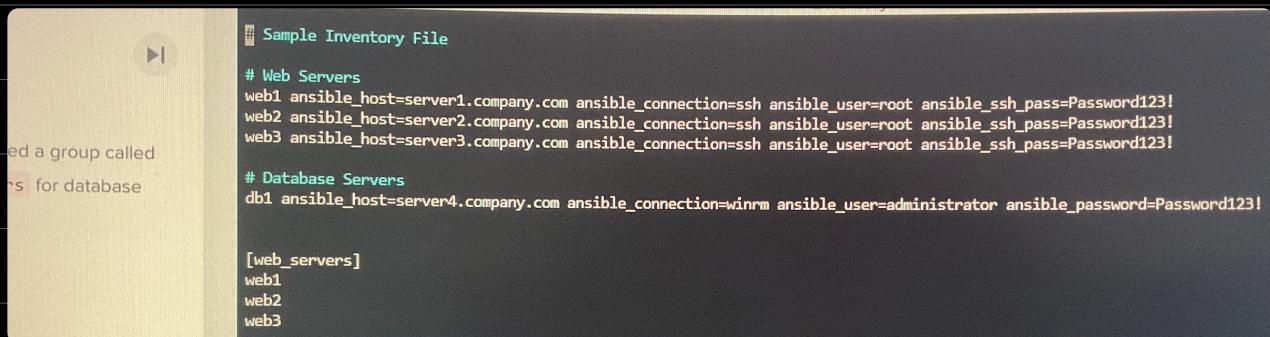
/etc/ansible/hosts

There are smaller changes in variables for Linux and windows files

ex. Linux parameter —
ansible-ssh-pass

Win — ansible-password

• inventory file example



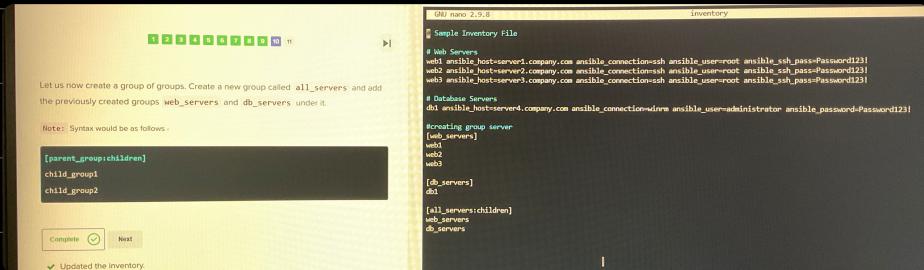
```
# Sample Inventory File

# Web Servers
web1 ansible_host=server1.company.com ansible_connection=ssh ansible_user=root ansible_ssh_pass=Password123!
web2 ansible_host=server2.company.com ansible_connection=ssh ansible_user=root ansible_ssh_pass=Password123!
web3 ansible_host=server3.company.com ansible_connection=ssh ansible_user=root ansible_ssh_pass=Password123!

# Database Servers
db1 ansible_host=server4.company.com ansible_connection=winrm ansible_user=administrator ansible_password=Password123!

[web_servers]
web1
web2
web3
```

Adding parent children



Ansible playbook

- Ansible's orchestration language
- We define sets of instructions

Calling variables - {{ variable }}

Modules - command, script, service etc.

The screenshot shows a terminal window with three main sections:

- Playbook (left):** A snippet of Ansible YAML code defining a task to set firewall configurations. It includes multiple `firewall` entries with various port ranges and states.
- Inventory File (top right):** A sample inventory file named `web` containing host definitions for `Web` and `Snmp`.
- Variable File (middle right):** A sample variable file named `web.yml` defining variables `http_port`, `snmp_port`, and `inter_ip_range`.
- Jinja2 Templating (bottom right):** A section titled "Jinja2 Templating" showing examples of Jinja2 syntax:
 - A red X mark next to `source: {{ inter_ip_range }}` indicates it's invalid.
 - A green checkmark next to `source: '{{ inter_ip_range }}'` indicates it's valid and correctly formatted.
 - A green checkmark next to `source: SomeThing{{ inter_ip_range }}SomeThing` indicates it's valid and demonstrates a common anti-pattern where the double quotes are not required within the sentence.

To run a playbook

```
ansible-playbook -i inventory playbook.yaml
```

Ansible Modules

- system
- commands
- files
- database
- cloud
- windows

- & more

Ansible precedence for variables

precedence

- Extra vars defined
- include vars
- play vars

command to verify ansible playbook syntax

- ansible web --list-host -i inventory
- ansible all --list-host -i inventory
- Ansible config file → /etc/ansible/ansible.cfg
- To set a default inventory we can change that into ansible.cfg
But its better to make settings for user specific than global.

How-to?

copy to

/home/lec2-user/.ansible.cfg

then make changes here.

→ To make ansible file user specific

- Higher precedence ↓
Lowest ↓
- ① Export value
 - ② current working directory
 - ③ User Home w/.ansible.cfg
 - ④ Use global etc/ansible/ansible.cfg

To check the connections

ansible all -m ping

ansible all -m ping -k

↳ Auth via password

To put group level variables

[web : vars]

ansible_ssh_user = ec2-user

ansible_ssh_pass = redhat

1 • Adhoc method to push task

To get a list of ansible module names

• ansible doc -1

To get details about modules

ansible-doc copy

• ansible single session continues until 60 sec

To give sudo permissions to admin you can go to /etc/sudoers

Add sudo powers here ALL ALL

- On remote systems which has python version below 2.4 we can only use raw modules → or else command modules
- We cannot use other modules there

- shell module → To execute any scripting files

-

Example-1: Use of ansible module

- 1) Copy module: it use to copy data control to Manage Node.
- 2) Command: it execute os command
- 3) Raw
- 4) Shell: it used to execute any shell scripting
- 5) File: touch/mkdir/chmod/chown/chgrp/rm/soft/hard
- 6) fetch: it used to copy files manage node to control Node.
- 7) Get_url:
- 8) lineinfile
- 9) replace
- 10) User
- 11) group
- 12) yum

20 playbooks method to push task

- We run playbooks to complete repetitive tasks
- Adhoc → imperative way works for one time runs

playbook always starts with

To verify syntax of playbook

ansible-playbook pl.yml --syntax-check

To actually check with dry-run

ansible-playbook pl.yml --check
or -c

ansible-playbook pl.yml --step

Multi-play →

To run single playbook on different Hosts

• import-task -

If mistake in any mentioned file in playbook it won't work → gives error

• include-task - In this case other task will run

-

- Variables

Variables can be passed through

- ① playbook level
- ② command line
- ③ Inventory file

Advantages of Ansible over other Configuration management tools

Yaml formats

① key : value based → name: Sam

② Array / List format

web:

- httpd
- nginx
- Yum

③ Mapping / dictionary format

task:

web: httpd

skill: python

name: Sam

Types of inventory

① static inventory

② dynamic inventory

Ansible work .

① plugin

② Modules

③ Playbooks

① Plugins are custom modules which enhances the functionality and capabilities of ansible.

ex. inventory plugin

connection plugin

Module plugin

- Ansible Tower - UI / Dashboard