

# Ansible Templating



# Templating ( Jinja2 )

2

- Ansible uses Jinja2 templating to enable dynamic expressions and access to variables.
- A Jinja2 template is composed of multiple elements: data, variables, and expressions.
- Variables and expressions are replaced with their values when the Jinja2 template is rendered.
- The variables used in the template can be specified in the vars section of the playbook.
- It is possible to use the managed host's facts as variables on a template.

**Note:** Remember that the facts associated with a managed host can be obtained using the :  
**ansible system\_hostname -i inventory\_file -m setup** command.

# Example

3

```
# {{ ansible_managed }}
# DO NOT MAKE LOCAL MODIFICATIONS TO THIS FILE AS THEY WILL BE LOST

Port {{ ssh_port }}
ListenAddress {{ ansible_facts['default_ipv4']['address'] }}

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

SyslogFacility AUTHPRIV

PermitRootLogin {{ root_allowed }}
AllowGroups {{ groups_allowed }}

AuthorizedKeysFile /etc/.rht_authorized_keys .ssh/authorized_keys

PasswordAuthentication {{ passwords_allowed }}
```

# Deploying Jinja2 Templates

4

- Jinja2 templates are a powerful tool to customize configuration files to be deployed on the managed hosts.
- When the Jinja2 template for a configuration file has been created, it can be deployed to the managed hosts using the template module, which supports the transfer of a local file on the control node to the managed hosts.
- To use the template module, use the following syntax.

```
tasks:  
- name: template render  
  template:  
    src: /tmp/j2-template.j2  
    dest: /tmp/dest-config-file.txt
```