Listings Chapter 2

Installing Ansible

Listing 2-1 Sample /etc/hosts Contents

```
127.0.0.1 localhost localhost.localdomain localhost4 \
localhost4.localdomain4
::1  localhost localhost.localdomain localhost6 \
localhost6.localdomain6
192.168.4.200 control.example.com control
192.168.4.201 ansible1.example.com ansible1
192.168.4.202 ansible2.example.com ansible2
```

Listings Chapter 3

Setting Up an Ansible Managed Environment

Listing 3-1 Inventory File with Host Groups

```
ansible1
ansible2
192.168.4.1
192.168.4.2

[web]
web1
web2
[db]
db1
db2

[servers:children]
web
db
```

Listing 3-2 Sample Inventory with Variables

```
[lamp]
ansible1.example.com

[file]
ansible2.example.com

[win]
windows.example.com

[win:vars]
ansible_user=ansible
ansible_password=@nsible123
ansible_connection=winrm
ansible_winrm_server_cert_validation=ignore
```

Listing 3-3 Dynamic Inventory Script Example

```
#!/usr/bin/python
""" Dynamic Inventory Script Example """

from subprocess import Popen, PIPE
import sys

try:
    import json
except ImportError:
    import simplejson as json

RESULT = {}
RESULT['all'] = {}

PIPE = Popen(['getent', 'hosts'], stdout=PIPE,
universal_newlines=True)

RESULT['all']['hosts'] = []

for line in PIPE.stdout.readlines():
    s = line.split()
    RESULT['all']['hosts'] = RESULT['all']['hosts']+s
RESULT['all']['vars'] = {}

if len(sys.argv) == 2 and sys.argv[1] == '--list':
    print(json.dumps(RESULT))

elif len(sys.argv) == 3 and sys.argv[1] == '--host':
    print(json.dumps({}))

else:
    print("Requires an argument, please use --list or --host <host>")
```

Listing 3-4 Sample ansible.cfg File

```
[defaults]
remote_user = ansible
host_key_checking = false
inventory = inventory

[privilege_escalation]
become = True
become_method = sudo
become_user = root
become_ask_pass = False
```

Listings Chapter 4

Using Ad Hoc Commands

Listing 4-1 Ad Hoc Command Output

```
[ansible@control ~]$ ansible ungrouped -m user -a "name=lisa"
ansible2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/
platform-python"
    },
    "append": false,
    "changed": false,
    "comment": "",
    "group": 1002,
    "home": "/home/lisa",
   "move_home": false,
    "name": "lisa",
    "shell": "/bin/bash",
    "state": "present",
    "uid": 1002
}
ansible1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/
platform-python"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1001,
    "home": "/home/lisa",
    "name": "lisa",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 1001
}
```

Listing 4-2 Analyzing **ping** Module Output

```
[ansible@control ~]$ ansible all -m ping
ansible3 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: ssh: Could not
resolve hostname ansible3: Name or service not known",
    "unreachable": true
}
ansible2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/
platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ansible1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/
platform-python"
    },
    "changed": false,
    "ping": "pong"
}
```

Listing 4-3 Analyzing **ansible-doc** Command Output

```
> PING    (/usr/lib/python3.6/site-packages/ansible/modules/system/
ping.py)

        A trivial test module, this module always returns 'pong' on
        successful contact. It does not make sense in playbooks, but
        it is useful from '/usr/bin/ansible' to verify the ability to
        login and that a usable Python is configured. This is NOT ICMP
        ping, this is just a trivial test module that requires Python
        on the remote-node. For Windows targets, use the [win_ping]
        module instead. For Network targets, use the [net_ping] module
        instead.

  * This module is maintained by The Ansible Core Team
OPTIONS (= is mandatory):
```

```
- data
         Data to return for the 'ping' return value.
         If this parameter is set to 'crash', the module will cause an
         exception.
         [Default: pong]
         type: str


SEE ALSO:
       * Module net_ping
            The official documentation on the net_ping module.
            https://docs.ansible.com/ansible/2.9/modules/net_ping
         _module.html
       * Module win_ping
            The official documentation on the win_ping module.
            https://docs.ansible.com/ansible/2.9/modules/win_ping
         _module.html


AUTHOR: Ansible Core Team, Michael DeHaan
         METADATA:
           status:
           - stableinterface
           supported_by: core

EXAMPLES:

# Test we can logon to 'webservers' and execute python with json lib.
# ansible webservers -m ping
# Example from an Ansible Playbook
- ping:

# Induce an exception to see what happens
- ping:
    data: crash


RETURN VALUES:

ping:
    description: value provided with the data parameter
    returned: success
    type: str
    sample: pong
```

Listing 4-4 Showing Usage Information with **ansible-doc -s service**

```
- name: Manage services
  service:
      arguments:        # Additional arguments provided on the
                          command line.
      enabled:          # Whether the service should start on boot.
                          *At least one of state and
                          enabled are required.*
      name:             # (required) Name of the service.
      pattern:          # If the service does not respond to the
                          status command, name a substring to
                          look for as would be
                          found in the output of
                          the 'ps' command as a
                          stand-in for a status
                          result. If the string
                          is found, the service
                          will be assumed to be
                          started.
      runlevel:         # For OpenRC init scripts (e.g. Gentoo) only.
                          The runlevel that this
                          service belongs to.
      sleep:            # If the service is being 'restarted' then
                          many seconds between
                          the stop and start
                          command. This helps to
                          work around badly-
                          behaving init scripts
                          that exit immediately
                          after signaling a
                          process to stop. Not
                          all service managers
                          support sleep, i.e when
                          using systemd this
                          setting will be
                          ignored.
```

Listing 4-5 Running Ad Hoc Commands from a Script

```
#!/bin/bash


ansible all -m yum -a "name=httpd state=latest"
ansible all -m service -a "name=httpd state=started enabled=yes"
```

Listings Chapter 5

Getting Started with Playbooks

Listing 5-1 Running Ad Hoc Commands from a Script

```
#!/bin/bash

ansible all -m yum -a "name=httpd state=installed"
ansible all -m service -a "name=httpd state=started enabled=yes"
```

Listing 5-2 Configuring Hosts from a Playbook

```
---
- name: install start and enable httpd
  hosts: all
  tasks:
  - name: install package
    yum:
      name: httpd
      state: installed
  - name: start and enable service
    service:
      name: httpd
      state: started
      enabled: yes
```

Listing 5-3 Running a Playbook Output

```
[ansible@control ~]$ ansible-playbook listing52.yaml

PLAY [install start and enable httpd] ********************************

TASK [Gathering Facts] ***********************************************
ok: [ansible2]
ok: [ansible1]

TASK [install package] ***********************************************
ok: [ansible1]
ok: [ansible2]
```

```
TASK [start and enable service] *************************************
ok: [ansible2]
ok: [ansible1]

PLAY RECAP **********************************************************
ansible1                    : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                    : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 5-4 Multiple Ways to Define Key-Value Pairs

```
---
- name: deploy vsftpd
  hosts: ansible2
  tasks:
  - name: install vsftpd
    yum: name=vsftpd
  - name: enable vsftpd
    service: name=vsftpd enabled=true
  - name: create readme file
    copy:
      content: "welcome to the FTP server\n"
      dest: /var/ftp/pub/README
      force: no
      mode: 0444
...
```

Listing 5-5 Installing Multiple Packages Using Lists

```
---
- name: using lists
  hosts: all
  tasks:
  - name: install packages
    yum:
      name:
      - nmap
      - httpd
      - vsftpd
      state: latest
```

Listing 5-6 Checking Syntax with **ansible-playbook --syntax-check**

```
[ansible@control ~]$ ansible-playbook --syntax-check listing57.yaml
ERROR! Syntax Error while loading YAML.
  mapping values are not allowed in this context

The error appears to be in '/home/ansible/listing54.yaml': line 8,
column 12, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

    name: httpd
      state: latest
           ^ here
```

Listing 5-7 Sample YAML File with Errors

```
---
- name: install start and enable httpd
  hosts: all
  tasks:
  - name: install package
    yum:
    name: httpd
      state: latest
  - service:
    name: httpd
      state: started
      enabled: yes
```

Listing 5-8 Performing a Playbook Dry Run

```
[ansible@control ~]$ ansible-playbook -C listing57.yaml

PLAY [using lists] **************************************************

TASK [Gathering Facts] **********************************************
ok: [ansible2]
ok: [ansible1]
```

```
TASK [install packages] *********************************************
changed: [ansible2]
changed: [ansible1]

PLAY RECAP **********************************************************
ansible1                     : ok=2     changed=1     unreachable=0
failed=0     skipped=0     rescued=0     ignored=0
ansible2                     : ok=2     changed=1     unreachable=0
failed=0     skipped=0     rescued=0     ignored=0
```

Listing 5-9 Multiplay Playbook Example

```
---
- name: install start and enable httpd
  hosts: all
  tasks:
  - name: install package
    yum:
      name: httpd
      state: latest

- name: start and enable service
    service:
      name: httpd
      state: started
      enabled: yes

- name: test httpd accessibility
  hosts: localhost
  tasks:
  - name: test httpd access
    uri:
      url: http://ansible1
```

Listing 5-10 Playbook Result

```
[ansible@control ~]$ ansible-playbook listing59.yaml

PLAY [install start and enable httpd] ********************************

TASK [Gathering Facts] ***********************************************
ok: [ansible2]
ok: [ansible1]

TASK [install package] ***********************************************
ok: [ansible2]
ok: [ansible1]

TASK [start and enable service] **************************************
ok: [ansible2]
ok: [ansible1]

PLAY [test httpd accessibility] **************************************

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

TASK [test httpd access] *********************************************
fatal: [localhost]: FAILED! => {"changed": false, "content": "",
"elapsed": 0, "msg": "Status code was -1 and not [200]: Request
failed: <urlopen error [Errno 113] No route to host>", "redirected":
false, "status": -1, "url": "http://ansible1"}

PLAY RECAP ***********************************************************
ansible1                   : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                   : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
localhost                  : ok=1    changed=0    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
```

Listing 5-11 ansible-playbook -vv Partial Output

```
[ansible@control ~]$ ansible-playbook -vv listing59.yaml
ansible-playbook 2.9.5
  config file = /home/ansible/ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/
modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.6/site-packages/
ansible
  executable location = /usr/bin/ansible-playbook
  python version = 3.6.8 (default, Nov 21 2019, 19:31:34) [GCC 8.3.1
20190507 (Red Hat 8.3.1-4)]
Using /home/ansible/ansible.cfg as config file

PLAYBOOK: listing59.yaml *********************************************
2 plays in listing59.yaml

PLAY [install start and enable httpd] ********************************

TASK [Gathering Facts] ***********************************************
task path: /home/ansible/listing59.yaml:2
ok: [ansible2]
ok: [ansible1]
META: ran handlers

TASK [install package] ***********************************************
task path: /home/ansible/listing59.yaml:5
ok: [ansible2] => {"changed": false, "msg": "Nothing to do", "rc": 0,
"results": []}
ok: [ansible1] => {"changed": false, "msg": "Nothing to do", "rc": 0,
"results": []}
```

Listing 5-12 ansible-playbook -vvvv Partial Output

```
[ansible@control ~]$ ansible-playbook -vvvv listing59.yaml
ansible-playbook 2.9.5
  config file = /home/ansible/ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/
modules', '/usr/share/ansible/plugins/modules']
ansible python module location = /usr/lib/python3.6/site-packages/
ansible
  executable location = /usr/bin/ansible-playbook
  python version = 3.6.8 (default, Nov 21 2019, 19:31:34) [GCC 8.3.1
20190507 (Red Hat 8.3.1-4)]
Using /home/ansible/ansible.cfg as config file
setting up inventory plugins
host_list declined parsing /home/ansible/inventory as it did not pass
its verify_file() method
script declined parsing /home/ansible/inventory as it did not pass its
verify_file() method
auto declined parsing /home/ansible/inventory as it did not pass its
verify_file() method
Parsed /home/ansible/inventory inventory source with ini plugin
Loading callback plugin default of type stdout, v2.0 from /usr/lib/
python3.6/site-packages/ansible/plugins/callback/default.py

PLAYBOOK: listing59.yaml **********************************************
Positional arguments: listing59.yaml
verbosity: 4
remote_user: ansible
connection: smart
timeout: 10
become: True
become_method: sudo
tags: ('all',)
inventory: ('/home/ansible/inventory',)
forks: 5
2 plays in listing57.yaml

PLAY [install start and enable httpd] *******************************

TASK [Gathering Facts] ***********************************************
task path: /home/ansible/listing59.yaml:2
<ansible2> ESTABLISH SSH CONNECTION FOR USER: ansible
<ansible2> SSH: EXEC ssh -vvv -C -o ControlMaster=auto
-o ControlPersist=60s -o StrictHostKeyChecking=no -o
KbdInteractiveAuthentication=no -o PreferredAuthentications=gssapi-
with-mic,gssapi-keyex,hostbased,publickey -o PasswordAuthentication=no
-o 'User="ansible"' -o ConnectTimeout=10 -o ControlPath=/home/
ansible/.ansible/cp/b95d9eb347 ansible2 '/bin/sh -c '"'"'echo ~ansible
&& sleep 0'"'"''
<ansible1> ESTABLISH SSH CONNECTION FOR USER: ansible
<ansible1> SSH: EXEC ssh -vvv -C -o ControlMaster=auto
-o ControlPersist=60s -o StrictHostKeyChecking=no -o
KbdInteractiveAuthentication=no -o PreferredAuthentications=gssapi-
with-mic,gssapi-keyex,hostbased,publickey -o PasswordAuthentication=no
-o 'User="ansible"' -o ConnectTimeout=10 -o ControlPath=/home/
```

```
ansible/.ansible/cp/88f8e128b5 ansible1 '/bin/sh -c '"'"'echo ~ansible
&& sleep 0'"'"''
<ansible2> (0, b'/home/ansible\n', b"OpenSSH_8.0p1, OpenSSL 1.1.1c
FIPS  28 May 2019\r\ndebug1: Reading configuration data /etc/ssh/
ssh_config\r\ndebug3: /etc/ssh/ssh_config line 51: Including file /
etc/ssh/ssh_config.d/05-redhat.conf depth 0\r\ndebug1: Reading
configuration data /etc/ssh/ssh_config.d/05-redhat.conf\r\ndebug2:
checking match for 'final all' host ansible2 originally ansible2\r\
ndebug3: /etc/ssh/ssh_config.d/05-redhat.conf line 3: not matched
'final'\r\ndebug2: match not found\r\ndebug3: /etc/ssh/ssh_
config.d/05-redhat.conf line 5: Including file /etc/crypto-policies/
back-ends/openssh.config depth 1 (parse only)\r\ndebug1: Reading
configuration data /etc/crypto-policies/back-ends/openssh.config\r\
ndebug3: gss kex names ok: [gss-gex-sha1-,gss-group14-sha1-]\r\
ndebug3: kex names ok: [curve25519-sha256,curve25519-sha256@libssh.
org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-
hellman-group-exchange-sha256,diffie-hellman-group14-sha256,diffie-
hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-
hellman-group-exchange-sha1,diffie-hellman-group14-sha1]\r\
ndebug1: configuration requests final Match pass\r\ndebug1:
re-parsing configuration\r\ndebug1: Reading configuration data /
etc/ssh/ssh_config\r\ndebug3: /etc/ssh/ssh_config line 51:
Including file /etc/ssh/ssh_config.d/05-redhat.conf depth 0\r\
ndebug1: Reading configuration data /etc/ssh/ssh_config.d/05-
redhat.conf\r\ndebug2: checking match for 'final all' host ansible2
originally ansible2\r\ndebug3: /etc/ssh/ssh_config.d/05-redhat.
conf line 3: matched 'final'\r\ndebug2: match found\r\ndebug3: /
etc/ssh/ssh_config.d/05-redhat.conf line 5: Including file /etc/
crypto-policies/back-ends/openssh.config depth 1\r\ndebug1: Reading
configuration data /etc/crypto-policies/back-ends/openssh.config\r\
ndebug3: gss kex names ok: [gss-gex-sha1-,gss-group14-sha1-]\r\
ndebug3: kex names ok: [curve25519-sha256,curve25519-sha256@libssh.
org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-
hellman-group-exchange-sha256,diffie-hellman-group14-sha256,diffie-
hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-
group-exchange-sha1,diffie-hellman-group14-sha1]\r\ndebug1: auto-mux:
Trying existing master\r\ndebug2: fd 4 setting O_NONBLOCK\r\ndebug2:
mux_client_hello_exchange: master version 4\r\ndebug3: mux_client_
forwards: request forwardings: 0 local, 0 remote\r\ndebug3: mux_
client_request_session: entering\r\ndebug3: mux_client_request_alive:
entering\r\ndebug3: mux_client_request_alive: done pid = 2384\r\
ndebug3: mux_client_request_session: session request sent\r\ndebug3:
mux_client_read_packet: read header failed: Broken pipe\r\ndebug2:
Received exit status from master 0\r\n")
<ansible2> ESTABLISH SSH CONNECTION FOR USER: ansible
<ansible2> SSH: EXEC ssh -vvv -C -o ControlMaster=auto
-o ControlPersist=60s -o StrictHostKeyChecking=no -o
KbdInteractiveAuthentication=no -o PreferredAuthentications=gssapi-
with-mic,gssapi-keyex,hostbased,publickey -o PasswordAuthentication=no
-o 'User="ansible"' -o ConnectTimeout=10 -o ControlPath=/home/
ansible/.ansible/cp/b95d9eb347 ansible2 '/bin/sh -c '"'"'(
umask 77 && mkdir -p "' echo /home/ansible/.ansible/tmp/
```

```
ansible-tmp-1585565643.3326187-58978915363330 '" && echo ansible-
tmp-1585565643.3326187-58978915363330="' echo /home/ansible/.ansible/
tmp/ansible-tmp-1585565643.3326187-58978915363330 '" ) && sleep
0'"'"''

<ansible1> (0, b'/home/ansible\n', b"OpenSSH_8.0p1, OpenSSL 1.1.1c
FIPS  28 May 2019\r\ndebug1: Reading configuration data /etc/ssh/
ssh_config\r\ndebug3: /etc/ssh/ssh_config line 51: Including file /
etc/ssh/ssh_config.d/05-redhat.conf depth 0\r\ndebug1: Reading
configuration data /etc/ssh/ssh_config.d/05-redhat.conf\r\ndebug2:
checking match for 'final all' host ansible1 originally ansible1\r\
ndebug3: /etc/ssh/ssh_config.d/05-redhat.conf line 3: not matched
'final'\r\ndebug2: match not found\r\ndebug3: /etc/ssh/ssh_
config.d/05-redhat.conf
```

Listings Chapter 6

Working with Variables and Facts

Listing 6-1 Using Variables Example

```
---
- name: create a user using a variable
  hosts: ansible1
  vars:
    users: lisa
  tasks:
    - name: create a user {{ users }} on host {{ ansible_hostname }}
      user:
        name: "{{ users }}"
```

Listing 6-2 Running the Listing 6-1 Sample Playbook

```
[ansible@control ~]$ ansible-playbook listing61.yaml

PLAY [create a user using a variable] ********************************
**************************************


TASK [Gathering Facts] **********************************************
***********************
ok: [ansible1]


TASK [create a user lisa on host ansible1] ***************************
**************************************
changed: [ansible1]


PLAY RECAP **********************************************************
************
ansible1                     : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 6-3 Simple Playbook to Show Fact Gathering

```
---
- name: show fact gathering
  hosts: all
  tasks:
  - name: show all facts
    debug:
      var: ansible_facts
```

Listing 6-4 Fact Gathering Partial Result

```
TASK [show all facts] ************************************************
***************************************
ok: [ansible1] => {
    "ansible_facts": {
        "_facts_gathered": true,
        "all_ipv4_addresses": [
            "192.168.122.1",
            "192.168.4.201"
        ],
       "all_ipv6_addresses": [
            "fe80::e564:5033:5dec:aead"
        ],
        "ansible_local": {},
        "apparmor": {
            "status": "disabled"
        },
        "architecture": "x86_64",
        "bios_date": "07/29/2019",
        "bios_version": "6.00",
        "cmdline": {
            "BOOT_IMAGE": "(hd0,msdos1)/vmlinuz-4.18.0-147.el8.x86_64",
            "crashkernel": "auto",
            "quiet": true,
            "rd.lvm.lv": "cl/swap",
            "resume": "/dev/mapper/cl-swap",
            "rhgb": true,
            "ro": true,
            "root": "/dev/mapper/cl-root"
        },
        "date_time": {
            "date": "2020-03-30",
            "day": "30",
            "epoch": "1585579970",
            "hour": "10",
            "iso8601": "2020-03-30T14:52:50Z",
            "iso8601_basic": "20200330T105250355357",
            "iso8601_basic_short": "20200330T105250",
            "iso8601_micro": "2020-03-30T14:52:50.355419Z",
            "minute": "52",
            "month": "03",
            "second": "50",
            "time": "10:52:50",
            "tz": "EDT",
            "tz_offset": "-0400",
            "weekday": "Monday",
            "weekday_number": "1",
            "weeknumber": "13",
            "year": "2020"
        },
        "default_ipv4": {
            "address": "192.168.4.201"
```

Listing 6-5 ansible -m setup Shows Facts Injected as Variables

```
ansible1 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "192.168.122.1",
            "192.168.4.201"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::e564:5033:5dec:aead"
        ],
        "ansible_apparmor": {
            "status": "disabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "07/29/2019",
        "ansible_bios_version": "6.00",
        "ansible_cmdline": {
            "BOOT_IMAGE": "(hd0,msdos1)/vmlinuz-4.18.0-147.el8.
x86_64",
            "crashkernel": "auto",
            "quiet": true,
            "rd.lvm.lv": "cl/swap",
            "resume": "/dev/mapper/cl-swap",
            "rhgb": true,
            "ro": true,
            "root": "/dev/mapper/cl-root"
        },
        "ansible_date_time": {
            "date": "2020-03-31",
            "day": "31",
            "epoch": "1585645366",
            "hour": "05",
            "iso8601": "2020-03-31T09:02:46Z",
            "iso8601_basic": "20200331T050246844129",
            "iso8601_basic_short": "20200331T050246",
            "iso8601_micro": "2020-03-31T09:02:46.844200Z",
            "minute": "02",
            "month": "03",
            "second": "46",
            "time": "05:02:46",
            "tz": "EDT",
            "tz_offset": "-0400",
            "weekday": "Tuesday",
            "weekday_number": "2",
            "weeknumber": "13",
            "year": "2020"
        },
        "ansible_default_ipv4": {
            "address": "192.168.4.201",
            "alias": "ens33",
            "broadcast": "192.168.4.255",
            "gateway": "192.168.4.2",
            "interface": "ens33",
            "macaddress": "00:0c:29:1f:c1:23",
            "mtu": 1500,
            "netmask": "255.255.255.0",
            "network": "192.168.4.0",
            "type": "ether"
        },
```

Listing 6-6 Addressing Facts with Injected Variables

```
---
- hosts: all
  tasks:
  - name: show IP address
    debug:
      msg: >
        This host uses IP address {{ ansible_default_ipv4.address }}
```

Listing 6-7 Addressing Facts Using the ansible_facts Variable

```
---
- hosts: all
  tasks:
  - name: show IP address
    debug:
      msg: >
        This host uses IP address {{ ansible_facts.default_ipv4.
address }}
```

Listing 6-8 Custom Facts Sample File

```
[packages]
web_package = httpd
ftp_package = vsftpd

[services]
web_service = httpd
ftp_service = vsftpd
```

Listing 6-9 Sample Playbook to Copy Custom Facts

```
---
- name: Install custom facts
  hosts: all
  vars:
    remote_dir: /etc/ansible/facts.d
    facts_file: listing68.fact
  tasks:
  - name: create remote directory
    file:
      state: directory
      recurse: yes
      path: "{{ remote_dir }}"
  - name: install new facts
    copy:
      src: "{{ facts_file }}"
      dest: "{{ remote_dir }}"
```

Listing 6-10 Defining Variables in a Playbook Header

```
---
- name: using variables
  hosts: ansible1
  vars:
    ftp_package: vsftpd
  tasks:
  - name: install package
    yum:
      name: "{{ ftp_package }}"
      state: latest
```

Listing 6-11 Using a Variable Include File

```
---
- name: using a variable include file
  hosts: ansible1
  vars_files: vars/common
  tasks:
  - name: install package
    yum:
      name: "{{ my_package }}"
      state: latest
```

Listing 6-12 Variable Include File Contents

```
my_package: nmap
my_ftp_service: vsftpd
my_file_service: smb
```

Listing 6-13 Array Example

```
users:
  - linda:
    username: linda
    homedir: /home/linda
    shell: /bin/bash
  - lisa:
    username: lisa
    homedir: /home/lisa
    shell: /bin/bash
  - anna:
    username: anna
    homedir: /home/anna
    shell: /bin/bash
```

Listing 6-14 Dictionary Example

```
users:
  linda:
    username: linda
    homedir: /home/linda
    shell: /bin/bash
  lisa:
    username: lisa
    homedir: /home/lisa
    shell: /bin/bash
  anna:
    username: anna
    homedir: /home/anna
    shell: /bin/bash
```

Listing 6-15 Addressing Specific Keys in a Dictionary Multivalued Variable

```
---
- name: show dictionary also known as hash
  hosts: ansible1
  vars_files:
  - vars/users-dictionary
  tasks:
  - name: print dictionary values
    debug:
      msg: "User {{ users.linda.username }} has homedirectory {{
users.linda.homedir }} and shell {{ users.linda.shell }}"
```

Listing 6-16 Using the Square Brackets Notation to Address Multivalued Variables

```
---
- name: show dictionary also known as hash
  hosts: ansible1
  vars_files:
    - vars/users-dictionary
  tasks:
    - name: print dictionary values
      debug:
        msg: "User {{ users['linda']['username'] }} has homedirectory
{{ users['linda']['homedir'] }} and shell {{ users['linda']['shell']
}}"
```

Listing 6-17 Using the debug Module to Show Hostvars Variables

```
[ansible@control ~]$ ansible localhost -m debug -a
'var=hostvars["ansible1"]'
localhost | SUCCESS => {
    "hostvars[\"ansible1\"]": {
        "ansible_check_mode": false,
        "ansible_diff_mode": false,
        "ansible_facts": {},
        "ansible_forks": 5,
        "ansible_inventory_sources": [
            "/home/ansible/inventory"
        ],
        "ansible_playbook_python": "/usr/bin/python3.6",
        "ansible_verbosity": 0,
        "ansible_version": {
            "full": "2.9.5",
            "major": 2,
            "minor": 9,
            "revision": 5,
            "string": "2.9.5"
        },
        "group_names": [
            "ungrouped"
        ],
        "groups": {
            "all": [
                "ansible1",
                "ansible2"
           ],
            "ungrouped": [
                "ansible1",
                "ansible2"
            ]
        },
        "inventory_dir": "/home/ansible",
        "inventory_file": "/home/ansible/inventory",
        "inventory_hostname": "ansible1",
        "inventory_hostname_short": "ansible1",
        "omit": "__omit_place_holder__38849508966537e44da5c665d4a784c
3bc0060de",
        "playbook_dir": "/home/ansible"
    }
}
```

Listing 6-18 Sample Playbook That Uses **register**

```
---
- name: test register
  hosts: ansible1
  tasks:
  - shell: cat /etc/passwd
    register: passwd_contents
  - debug:
      var: "passwd_contents"
```

Listing 6-19 Partial Result of Running **ansible-playbook listing618.yaml**

```
[ansible@control ~]$ ansible-playbook listing618.yaml

PLAY [test register] *************************************************
******************


TASK [Gathering Facts] ***********************************************
******************
ok: [ansible2]
ok: [ansible1]


TASK [shell] *********************************************************
******************
changed: [ansible2]
changed: [ansible1]
```

```
TASK [debug] *********************************************************
******************
ok: [ansible1] => {
    "passwd_contents": {
        "changed": true,
        "cmd": "cat /etc/passwd",
        "delta": "0:00:00.004149",
        "end": "2020-04-02 02:28:10.692306",
        "failed": false,
        "rc": 0,
        "start": "2020-04-02 02:28:10.688157",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "root:x:0:0:root:/root:/bin/bash\nbin:x:1:1:bin:/
bin:/sbin/nologin\ndaemon:x:2:2:daemon:/sbin:/sbin/nologin\nadm:x:
3:4:adm:/var/adm:/sbin/nologin\nlp:x:4:7:lp:/var/spool/lpd:/sbin/
nologin\nsync:x:5:0:sync:/sbin:/bin/sync\nshutdown:x:6:0:shutdown:/
sbin:/sbin/shutdown\nhalt:x:7:0:halt:/sbin:/sbin/halt\nansible:x:
1000:1000:ansible:/home/ansible:/bin/bash\napache:x:48:48:Apache:/usr/
share/httpd:/sbin/nologin\nlinda:x:1002:1002::/home/linda:/bin/bash\
nlisa:x:1003:1003::/home/lisa:/bin/bash",
        "stdout_lines": [
            "root:x:0:0:root:/root:/bin/bash",
            "bin:x:1:1:bin:/bin:/sbin/nologin",
            "daemon:x:2:2:daemon:/sbin:/sbin/nologin",
            "adm:x:3:4:adm:/var/adm:/sbin/nologin",
            "lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin",
            "sync:x:5:0:sync:/sbin:/bin/sync",
            "shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown",
            "halt:x:7:0:halt:/sbin:/sbin/halt",
            "ansible:x:1000:1000:ansible:/home/ansible:/bin/bash",
            "apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin",
            "linda:x:1002:1002::/home/linda:/bin/bash",
            "lisa:x:1003:1003::/home/lisa:/bin/bash"
        ]
    }
}
```

Listings Chapter 7

Using Task Control

Listing 7-1 Using **loop**

```
---
- name: install and start services
  hosts: ansible1
  tasks:
  - name: install packages
    yum:
      name:
      - vsftpd
      - httpd
      - samba
      state: latest
  - name: start the services
    service:
      name: "{{ item }}"
      state: started
      enabled: yes
    loop:
    - vsftpd
    - httpd
    - smb
```

Listing 7-2 Providing the Loop by a Variable

```
---
- name: install and start services
  hosts: ansible1
  vars:
    services:
    - vsftpd
    - httpd
    - smb
  tasks:
  - name: install packages
    yum:
      name:
      - vsftpd
      - httpd
      - samba
    state: latest
  - name: start the services
    service:
      name: "{{ item }}"
      state: started
      enabled: yes
    loop: "{{ services }}"
```

Listing 7-3 Variables File

```
users:
  - username: linda
    homedir: /home/linda
    shell: /bin/bash
    groups: wheel
  - username: lisa
    homedir: /home/lisa
    shell: /bin/bash
    groups: users
  - username: anna
    homedir: /home/anna
    shell: /bin/bash
    groups: users
```

Listing 7-4 Using Multivalued Variables

```
---
- name: create users using a loop from a list
  hosts: ansible1
  vars_files: vars/users-list
  tasks:
  - name: create users
    user:
      name: "{{ item.username }}"
      state: present
      groups: "{{ item.groups }}"
      shell: "{{ item.shell }}"
    loop: "{{ users }}"
```

Listing 7-5 Working with Multivalued Variables Output

```
[ansible@control ~]$ ansible-playbook listing74.yaml

PLAY [create users using a loop from a list] ************************

TASK [Gathering Facts] **********************************************
ok: [ansible1]

TASK [create users] *************************************************
changed: [ansible1] => (item={'username': 'linda', 'homedir': '/home/
linda', 'shell': '/bin/bash', 'groups': 'wheel'})
changed: [ansible1] => (item={'username': 'lisa', 'homedir': '/home/
lisa', 'shell': '/bin/bash', 'groups': 'users'})
changed: [ansible1] => (item={'username': 'anna', 'homedir': '/home/
anna', 'shell': '/bin/bash', 'groups': 'users'})

PLAY RECAP *********************************************************
ansible1                    : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 7-6 Using **with_items**

```
---
- name: install and start services
  hosts: ansible1
  vars:
    services:
    - vsftpd
    - httpd
    - smb
  tasks:
  - name: install packages
    yum:
      name:
      - vsftpd
      - httpd
      - samba
      state: latest
  - name: start the services
    service:
      name: "{{ item }}"
      state: started
      enabled: yes
    with_items: "{{ services }}"
```

Listing 7-7 Using **when** for Conditional Software Installation

```
---
- name: conditional install
  hosts: all
  tasks:
  - name: install apache on Red Hat and family
    yum:
      name: httpd
      state: latest
    when: ansible_facts['os_family'] == "RedHat"
  - name: install apache on Ubuntu and family
    apt:
      name: apache2
      state: latest
    when: ansible_facts['os_family'] == "Debian"
```

Listing 7-8 Conditional Playbook Result

```
[ansible@control ~]$ ansible-playbook listing77.yaml

PLAY [conditional install] ******************************************

TASK [Gathering Facts] **********************************************
ok: [ansible2]
ok: [ansible1]

TASK [install apache on Red Hat and family] **************************
ok: [ansible1]
changed: [ansible2]
```

```
TASK [install apache on Ubuntu and family] ***************************
skipping: [ansible1]
skipping: [ansible2]

PLAY RECAP **********************************************************
ansible1                     : ok=2    changed=0    unreachable=0
failed=0    skipped=1    rescued=0     ignored=0
ansible2                     : ok=2    changed=1    unreachable=0
failed=0    skipped=1    rescued=0     ignored=0
```

Listing 7-9 Using **when** to Check Whether a Variable Exists

```
---
- name: check for existence of devices
  hosts: all
  tasks:
  - name: check if /dev/sda exists
    debug:
      msg: a disk device /dev/sda exists
    when: ansible_facts['devices']['sda'] is defined
  - name: check if /dev/sdb exists
    debug:
      msg: a disk device /dev/sdb exists
    when: ansible_facts['devices']['sdb'] is defined
  - name: dummy test, intended to fail
    debug:
      msg: failing
    when: dummy is defined
  - name: check if /dev/sdc does not exist
    debug:
      msg: there is no /dev/sdc device
    when: ansible_facts['devices']['sdc'] is not defined
```

Listing 7-10 Checking Whether a Variable Occurs in a List

```
---
- name: test if variable is in another variables list
  hosts: all
  vars_prompt:
  - name: my_answer
    prompt: which package do you want to install
  vars:
    supported_packages:
    - httpd
    - nginx
  tasks:
  - name: something
    debug:
      msg: you are trying to install a supported package
    when: my_answer in supported_packages
```

Listing 7-11 Using an Integer Check

```
---
- name: conditionals test
  hosts: all
  tasks:
  - name: install vsftpd if sufficient memory available
    package:
      name: vsftpd
      state: latest
    when: ansible_facts['memory_mb']['real']['free'] > 50
```

Listing 7-12 Combining Multiple Conditions

```
---
- name: testing multiple conditions
  hosts: all
  tasks:
  - name: showing output
    debug:
      msg: using CentOS 8.1
    when: ansible_facts['distribution_version'] == "8.1" and ansible_
facts['distribution'] == "CentOS"
```

Listing 7-13 Combining Complex Statements

```
---
- name: using multiple conditions
  hosts: all
  tasks:
  - package:
      name: httpd
      state: removed
    when: >
      ( ansible_facts['distribution'] == "RedHat" and
        ansible_facts['memfree_mb'] < 512 )
      or
      ( ansible_facts['distribution'] == "CentOS" and
        ansible_facts['memfree_mb'] < 256 )
```

Listing 7-14 Combining **loop** and **when**

```
---
- name: conditionals test
  hosts: all
  tasks:
  - name: update the kernel if sufficient space is available in /boot
    package:
      name: kernel
      state: latest
    loop: "{{ ansible_facts['mounts'] }}"
    when: item.mount == "/boot" and item.size_available > 200000000
```

Listing 7-15 Listing 7-14 Task Result

```
[ansible@control ~]$ ansible-playbook listing714.yaml

PLAY [conditionals test] *********************************************
******************

TASK [Gathering Facts] ***********************************************
******************
ok: [ansible2]
ok: [ansible1]

TASK [update kernel if sufficient space in /boot] ********************
******************
```

```
skipping: [ansible1] => (item={'mount': '/', 'device': '/dev/
mapper/cl-root', 'fstype': 'xfs', 'options': 'rw,seclabel,rela
time,attr2,inode64,noquota', 'size_total': 18238930944, 'size_
available': 13722013696, 'block_size': 4096, 'block_total': 4452864,
'block_available': 3350101, 'block_used': 1102763, 'inode_total':
8910848, 'inode_available': 8790863, 'inode_used': 119985, 'uuid':
'ef0bb39c-5a29-4c0a-9152-7dd3fd5254c2'})
skipping: [ansible2] => (item={'mount': '/', 'device': '/dev/
mapper/cl-root', 'fstype': 'xfs', 'options': 'rw,seclabel,rela
time,attr2,inode64,noquota', 'size_total': 18238930944, 'size_
available': 16635084800, 'block_size': 4096, 'block_total': 4452864,
'block_available': 4061300, 'block_used': 391564, 'inode_total':
8910848, 'inode_available': 8877221, 'inode_used': 33627, 'uuid':
'acdeb1af-c439-4030-b9ba-c21d4d4fb0a8'})
changed: [ansible2] => (item={'mount': '/boot', 'device': '/dev/
sda1', 'fstype': 'ext4', 'options': 'rw,seclabel,relatime', 'size_
total': 1023303680, 'size_available': 811139072, 'block_size': 4096,
'block_total': 249830, 'block_available': 198032, 'block_used': 51798,
'inode_total': 65536, 'inode_available': 65227, 'inode_used': 309,
'uuid': 'cc870ab6-1e0e-4d27-9df3-9e5961d9fa62'})
changed: [ansible1] => (item={'mount': '/boot', 'device': '/dev/
sda1', 'fstype': 'ext4', 'options': 'rw,seclabel,relatime', 'size_
total': 1023303680, 'size_available': 803180544, 'block_size': 4096,
'block_total': 249830, 'block_available': 196089, 'block_used': 53741,
'inode_total': 65536, 'inode_available': 65227, 'inode_used': 309,
'uuid': '7acd65d6-115f-499f-a02f-90364a18b9fc'})


PLAY RECAP ***********************************************************
******************
ansible1                     : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                     : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 7-16 Combining **register** and **loop**

```
---
- name: test register
  hosts: all
  tasks:
    - shell: cat /etc/passwd
      register: passwd_contents
    - debug:
        msg: passwd contains user lisa
      when: passwd_contents.stdout.find('lisa') != -1
```

Listing 7-17 Working with Handlers

```
---
- name: create file on localhost
  hosts: localhost
  tasks:
  - name: create index.html on localhost
    copy:
      content: "welcome to the webserver"
      dest: /tmp/index.html

- name: set up web server
  hosts: all
  tasks:
    - name: install httpd
      yum:
        name: httpd
        state: latest
    - name: copy index.html
      copy:
        src: /tmp/index.html
        dest: /var/www/html/index.html
      notify:
        - restart_web
    - name: copy nothing - intended to fail
      copy:
        src: /tmp/nothing
        dest: /var/www/html/nothing.html
  handlers:
    - name: restart_web
      service:
        name: httpd
        state: restarted
```

Listing 7-18 ansible-playbook listing717.yaml Command Result

```
[ansible@control ~]$ ansible-playbook listing717.yaml

PLAY [create file on localhost] **************************************

TASK [Gathering Facts] ***********************************************
ok: [localhost]
```

```
TASK [create index.html on localhost] ********************************
changed: [localhost]

PLAY [set up web server] *********************************************

TASK [Gathering Facts] **********************************************
ok: [ansible2]
ok: [ansible1]

TASK [install httpd] ************************************************
changed: [ansible2]
changed: [ansible1]

TASK [copy index.html] **********************************************
changed: [ansible2]
changed: [ansible1]

TASK [copy nothing - intended to fail] *******************************
An exception occurred during task execution. To see the full
traceback, use -vvv. The error was: If you are using a module and
expect the file to exist on the remote, see the remote_src option
fatal: [ansible2]: FAILED! => {"changed": false, "msg": "Could not
find or access '/tmp/nothing' on the Ansible Controller.\nIf you are
using a module and expect the file to exist on the remote, see the
remote_src option"}
An exception occurred during task execution. To see the full
traceback, use -vvv. The error was: If you are using a module and
expect the file to exist on the remote, see the remote_src option
fatal: [ansible1]: FAILED! => {"changed": false, "msg": "Could not
find or access '/tmp/nothing' on the Ansible Controller.\nIf you are
using a module and expect the file to exist on the remote, see the
remote_src option"}

RUNNING HANDLER [restart_web] ***************************************

PLAY RECAP **********************************************************
ansible1                    : ok=3     changed=2    unreachable=0
failed=1    skipped=0    rescued=0     ignored=0
ansible2                    : ok=3     changed=2    unreachable=0
failed=1    skipped=0    rescued=0     ignored=0
localhost                   : ok=2     changed=1    unreachable=0
failed=0    skipped=0    rescued=0     ignored=0
```

Listing 7-19 Running **playbook listing717.yaml** Again

```
[ansible@control ~]$ ansible-playbook listing717.yaml

PLAY [create file on localhost] *************************************

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

TASK [create index.html on localhost] *******************************
ok: [localhost]

PLAY [set up web server] ********************************************

TASK [Gathering Facts] **********************************************
ok: [ansible2]
ok: [ansible1]

TASK [install httpd] ************************************************
ok: [ansible2]
ok: [ansible1]

TASK [copy index.html] **********************************************
ok: [ansible2]
ok: [ansible1]

TASK [copy nothing - intended to fail] ******************************
changed: [ansible2]
changed: [ansible1]

PLAY RECAP **********************************************************
ansible1                    : ok=4    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                    : ok=4    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
localhost                   : ok=2    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 7-20 Successfully Running Listing 7-17

```
[ansible@control ~]$ ansible ansible2 -m file -a "name=/var/www/html/
index.html state=absent"
ansible2 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/
platform-python"
    },
    "changed": true,
    "path": "/var/www/html/index.html",
    "state": "absent"
}
[ansible@control ~]$ ansible-playbook listing717.yaml

PLAY [create file on localhost] **************************************

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

TASK [create index.html on localhost] ********************************
ok: [localhost]

PLAY [set up web server] *********************************************

TASK [Gathering Facts] ***********************************************
ok: [ansible2]
ok: [ansible1]

TASK [install httpd] *************************************************
ok: [ansible2]
ok: [ansible1]

TASK [copy index.html] ***********************************************
changed: [ansible2]
ok: [ansible1]

TASK [copy nothing - intended to fail] *******************************
ok: [ansible2]
ok: [ansible1]
```

```
RUNNING HANDLER [restart_web] ***************************************
changed: [ansible2]

PLAY RECAP **********************************************************
ansible1                    : ok=4    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                    : ok=5    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
localhost                   : ok=2    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 7-21 Example with **ignore_errors**

```
---
- name: restart sshd only if crond is running
  hosts: all
  tasks:
    - name: get the crond server status
      command: /usr/bin/systemctl is-active crond
      ignore_errors: yes
      register: result
    - name: restart sshd based on crond status
      service:
        name: sshd
        state: restarted
      when: result.rc == 0
```

Listing 7-22 Forcing Handlers to Run

```
---
- name: create file on localhost
  hosts: localhost
  tasks:
  - name: create index.html on localhost
    copy:
      content: "welcome to the webserver"
      dest: /tmp/index.html
```

```
- name: set up web server
  hosts: all
  force_handlers: yes
  tasks:
    - name: install httpd
      yum:
        name: httpd
        state: latest
    - name: copy index.html
      copy:
        src: /tmp/index.html
        dest: /var/www/html/index.html
      notify:
        - restart_web
    - name: copy nothing - intended to fail
      copy:
        src: /tmp/nothing
        dest: /var/www/html/nothing.html
  handlers:
    - name: restart_web
      service:
        name: httpd
        state: restarted
```

Listing 7-23 Using **failed_when**

```
---
- name: demonstrating failed_when
  hosts: all
  tasks:
  - name: run a script
    command: echo hello world
    ignore_errors: yes
    register: command_result
    failed_when: "'world' in command_result.stdout"
  - name: see if we get here
    debug:
      msg: second task executed
```

Listing 7-24 Result of Running **ansible-playbook listing723.yaml**

```
[ansible@control ~]$ ansible-playbook listing723.yaml

PLAY [demonstrating failed_when] *************************************

TASK [Gathering Facts] ***********************************************
ok: [ansible1]
ok: [ansible2]

TASK [run a script] **************************************************
fatal: [ansible2]: FAILED! => {"changed": true, "cmd": ["echo",
"hello", "world"], "delta": "0:00:00.004303", "end": "2020-04-06
03:44:56.748552", "failed_when_result": true, "rc": 0, "start": "2020-
04-06 03:44:56.744249", "stderr": "", "stderr_lines": [], "stdout":
"hello world", "stdout_lines": ["hello world"]}
...ignoring
fatal: [ansible1]: FAILED! => {"changed": true, "cmd": ["echo",
"hello", "world"], "delta": "0:00:00.004261", "end": "2020-04-06
03:44:56.770166", "failed_when_result": true, "rc": 0, "start": "2020-
04-06 03:44:56.765905", "stderr": "", "stderr_lines": [], "stdout":
"hello world", "stdout_lines": ["hello world"]}
...ignoring

TASK [see if we get here] ********************************************
ok: [ansible1] => {
    "msg": "second task executed"
}
ok: [ansible2] => {
    "msg": "second task executed"
}

PLAY RECAP ***********************************************************
ansible1                     : ok=3    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=1
ansible2                     : ok=3    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=1
```

Listing 7-25 Using the fail Module

```
---
- name: demonstrating the fail module
  hosts: all
  ignore_errors: yes
  tasks:
  - name: run a script
    command: echo hello world
    register: command_result
  - name: report a failure
    fail:
      msg: the command has failed
    when: "'world' in command_result.stdout"
  - name: see if we get here
    debug:
      msg: second task executed
```

Listing 7-26 Sample Playbook Contents

```
---
- name: demonstrate changed status
  hosts: all
  tasks:
  - name: check local time
    command: date
    register: command_result

  - name: print local time
    debug:
      var: command_result.stdout
```

Listing 7-27 Result of Running **ansible-playbook listing726.yaml**

```
[ansible@control ~]$ ansible-playbook listing726.yaml

PLAY [demonstrate changed status] ************************************

TASK [Gathering Facts] ***********************************************
ok: [ansible2]
ok: [ansible1]
```

```
TASK [check local time] *********************************************
changed: [ansible2]
changed: [ansible1]

TASK [print local time] *********************************************
ok: [ansible1] => {
    "command_result.stdout": "Mon Apr  6 04:11:26 EDT 2020"
}
ok: [ansible2] => {
    "command_result.stdout": "Mon Apr  6 04:11:26 EDT 2020"
}

PLAY RECAP **********************************************************
ansible1                   : ok=3    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                   : ok=3    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 7-28 Using **changed_when**

```
---
- name: demonstrate changed status
  hosts: all
  tasks:
  - name: check local time
    command: date
    register: command_result
    changed_when: false

  - name: print local time
    debug:
      var: command_result.stdout
```

Listing 7-29 Result of Running **ansible-playbook listing728.yaml**

```
[ansible@control ~]$ ansible-playbook listing728.yaml

PLAY [demonstrate changed status] ***********************************

TASK [Gathering Facts] **********************************************
ok: [ansible2]
ok: [ansible1]
```

```
TASK [check local time] **********************************************
ok: [ansible2]
ok: [ansible1]

TASK [print local time] **********************************************
ok: [ansible1] => {
    "command_result.stdout": "Mon Apr  6 04:15:26 EDT 2020"
}
ok: [ansible2] => {
    "command_result.stdout": "Mon Apr  6 04:15:26 EDT 2020"
}

PLAY RECAP ***********************************************************
ansible1                     : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                     : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 7-30 Using Blocks

```
- name: simple block example
  hosts: all
  tasks:
  - name: setting up http
    block:
   - name: installing http
      yum:
        name: httpd
        state: present
    - name: restart httpd
      service:
        name: httpd
        state: started
    when: ansible_distribution == "CentOS"
```

Listing 7-31 Using Blocks, **rescue**, and **always**

```
---
- name: using blocks
  hosts: all
  tasks:
  - name: intended to be successful
    block:
    - name: remove a file
      shell:
        cmd: rm /var/www/html/index.html
    - name: printing status
      debug:
      msg: block task was operated
    rescue:
    - name: create a file
      shell:
        cmd: touch /tmp/rescuefile
    - name: printing rescue status
      debug:
        msg: rescue task was operated
    always:
    - name: always write a message to logs
      shell:
        cmd: logger hello
    - name: always printing this message
      debug:
        msg: this message is always printed
```

Listing 7-32 Output of Command **ansible-playbook listing731.yaml**

```
[ansible@control ~]$ ansible-playbook listing731.yaml

PLAY [using blocks] *************************************************

TASK [Gathering Facts] **********************************************
ok: [ansible2]
ok: [ansible1]

TASK [remove a file] ************************************************
*****************
```

```
[WARNING]: Consider using the file module with state=absent rather
than running 'rm'. If you need to use command because file is
insufficient you can add 'warn: false' to this command task or set
'command_warnings=False' in ansible.cfg to get rid of this message.
changed: [ansible2]
changed: [ansible1]
TASK [printing status] **********************************************
ok: [ansible1] => {
    "msg": "block task was operated"
}
ok: [ansible2] => {
    "msg": "block task was operated"
}

TASK [always write a message to logs] *******************************
changed: [ansible2]
changed: [ansible1]

TASK [always printing this message] *********************************
ok: [ansible1] => {
    "msg": "this message is always printed"
}
ok: [ansible2] => {
    "msg": "this message is always printed"
}

PLAY RECAP **********************************************************
ansible1                   : ok=5    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                   : ok=5    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 7-33 Tasks in **rescue** Are Executed If Tasks in **block** Are Failing

```
[ansible@control ~]$ ansible-playbook listing731.yaml


PLAY [using blocks] *************************************************
***
TASK [Gathering Facts] **********************************************
***
ok: [ansible2]
ok: [ansible1]
```

```
TASK [remove a file] *************************************************
***
[WARNING]: Consider using the file module with state=absent rather
than running 'rm'. If you need to use command because file is
insufficient you can add 'warn: false' to this command task or set
'command_warnings=False' in ansible.cfg to get rid of this message.
fatal: [ansible2]: FAILED! => {"changed": true, "cmd": "rm /var/
www/html/index.html", "delta": "0:00:00.003018", "end": "2020-04-06
05:16:29.810703", "msg": "non-zero return code", "rc": 1, "start":
"2020-04-06 05:16:29.807685", "stderr": "rm: cannot remove '/var/www/
html/index.html': No such file or directory", "stderr_lines": ["rm:
cannot remove '/var/www/html/index.html': No such file or directory"],
"stdout": "", "stdout_lines": []}
fatal: [ansible1]: FAILED! => {"changed": true, "cmd": "rm /var/
www/html/index.html", "delta": "0:00:00.012466", "end": "2020-04-06
05:16:29.836735", "msg": "non-zero return code", "rc": 1, "start":
"2020-04-06 05:16:29.824269", "stderr": "rm: cannot remove '/var/www/
html/index.html': No such file or directory", "stderr_lines": ["rm:
cannot remove '/var/www/html/index.html': No such file or directory"],
"stdout": "", "stdout_lines": []}


TASK [create a file] *************************************************
***
[WARNING]: Consider using the file module with state=touch rather
than running 'touch'. If you need to use command because file is
insufficient you can add 'warn: false' to this command task or set
'command_warnings=False' in ansible.cfg to get rid of this message.
changed: [ansible2]
changed: [ansible1]


TASK [printing rescue status] ****************************************
***
ok: [ansible1] => {
    "msg": "rescue task was operated"
}
ok: [ansible2] => {
    "msg": "rescue task was operated"
}


TASK [always write a message to logs] *******************************
changed: [ansible2]
changed: [ansible1]
TASK [always printing this message] *********************************
ok: [ansible1] => {
    "msg": "this message is always printed"
}
```

```
ok: [ansible2] => {
    "msg": "this message is always printed"
}

PLAY RECAP *********************************************************
ansible1                     : ok=5    changed=2    unreachable=0
failed=0    skipped=0    rescued=1    ignored=0
ansible2                     : ok=5    changed=2    unreachable=0
failed=0    skipped=0    rescued=1    ignored=0
```

Listings Chapter 8

Deploying Files

Listing 8-1 Exploring the stat Module

```
---
- name: stat module tests
  hosts: ansible1
  tasks:
  - stat:
      path: /etc/hosts
    register: st
  - name: show current values
    debug:
      msg: current value of the st variable is {{ st }}
```

Listing 8-2 Running **ansible-playbook listing81.yaml**

```
[ansible@control ~]$ ansible-playbook listing81.yaml

PLAY [stat module tests] *********************************************

TASK [Gathering Facts] ***********************************************
ok: [ansible1]

TASK [stat] **********************************************************
ok: [ansible1]

TASK [show current values] *******************************************
ok: [ansible1] => {
    "msg": "current value of the st variable is {'changed':
False, 'stat': {'exists': True, 'path': '/etc/hosts', 'mode':
'0644', 'isdir': False, 'ischr': False, 'isblk': False, 'isreg':
True, 'isfifo': False, 'islnk': False, 'issock': False, 'uid': 0,
'gid': 0, 'size': 158, 'inode': 16801440, 'dev': 64768, 'nlink':
1, 'atime': 1586230060.147566, 'mtime': 1536580263.0, 'ctime':
1584958718.8117938, 'wusr': True, 'rusr': True, 'xusr': False, 'wgrp':
False, 'rgrp': True, 'xgrp': False, 'woth': False, 'roth': True,
'xoth': False, 'isuid': False, 'isgid': False, 'blocks': 8, 'block_
size': 4096, 'device_type': 0, 'readable': True, 'writeable': True,
'executable': False, 'pw_name': 'root', 'gr_name': 'root', 'checksum':
'7335999eb54c15c67566186bdfc46f64e0d5a1aa', 'mimetype': 'text/plain',
'charset': 'us-ascii', 'version': '408552077', 'attributes': [],
'attr_flags': ''}, 'failed': False}"
}
```

```
PLAY RECAP ********************************************************
ansible1                    : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 8-3 Performing File State Tests with the stat Module

```
---
- name: stat module tests
  hosts: ansible1
  tasks:
  - command: touch /tmp/statfile
  - stat:
      path: /tmp/statfile
    register: st
  - name: show current values
    debug:
      msg: current value of the st variable is {{ st }}
  - fail:
      msg: "unexpected file mode, should be set to 0640"
    when: st.stat.mode != '0640'
```

Listing 8-4 Running **ansible-playbook listing83.yaml** Result

```
[ansible@control ~]$ ansible-playbook listing83.yaml

PLAY [stat module tests] *******************************************

TASK [Gathering Facts] *********************************************
ok: [ansible1]
TASK [command] *****************************************************
[WARNING]: Consider using the file module with state=touch rather
than running 'touch'. If you need to use command because file is
insufficient you can add 'warn: false' to this command task or set
'command_warnings=False' in ansible.cfg to get rid of this message.
changed: [ansible1]

TASK [stat] ********************************************************
ok: [ansible1]
```

```
TASK [show current values] *******************************************
ok: [ansible1] => {
    "msg": "current value of the st variable is {'changed':
False, 'stat': {'exists': True, 'path': '/tmp/statfile', 'mode':
'0644', 'isdir': False, 'ischr': False, 'isblk': False, 'isreg':
True, 'isfifo': False, 'islnk': False, 'issock': False, 'uid': 0,
'gid': 0, 'size': 0, 'inode': 51440456, 'dev': 64768, 'nlink': 1,
'atime': 1586253087.057596, 'mtime': 1586253087.057596, 'ctime':
1586253087.057596, 'wusr': True, 'rusr': True, 'xusr': False, 'wgrp':
False, 'rgrp': True, 'xgrp': False, 'woth': False, 'roth': True,
'xoth': False, 'isuid': False, 'isgid': False, 'blocks': 0, 'block_
size': 4096, 'device_type': 0, 'readable': True, 'writeable': True,
'executable': False, 'pw_name': 'root', 'gr_name': 'root', 'checksum':
'da39a3ee5e6b4b0d3255bfef95601890afd80709', 'mimetype': 'inode/x-
empty', 'charset': 'binary', 'version': '158303785', 'attributes': [],
'attr_flags': ''}, 'failed': False}"
}


TASK [fail] **********************************************************
fatal: [ansible1]: FAILED! => {"changed": false, "msg": "unexpected
file mode, should be set to 0640"}


PLAY RECAP ***********************************************************
ansible1                     : ok=4    changed=1    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
```

Listing 8-5 Using the file Module to Correct File Properties Discovered with stat

```
---
- name: stat module tests
  hosts: ansible1
  tasks:
  - command: touch /tmp/statfile
  - stat:
      path: /tmp/statfile
    register: st
  - name: show current values
    debug:
      msg: current value of the st variable is {{ st }}
  - name: changing file permissions if that's needed
    file:
      path: /tmp/statfile
      mode: 0640
    when: st.stat.mode != '0640'
```

Listing 8-6 Changing File Contents Using lineinfile

```
---
- name: configuring SSH
  hosts: all
  tasks:
  - name: disable root SSH login
    lineinfile:
      dest: /etc/ssh/sshd_config
      regexp: "^PermitRootLogin"
      line: "PermitRootLogin no"
    notify: restart sshd

  handlers:
  - name: restart sshd
    service:
      name: sshd
      state: restarted
```

Listing 8-7 Using blockinfile to Manipulate Multiple Lines of Text

```
---
- name: modifying file
  hosts: all
  tasks:
  - name: ensure /tmp/hosts exists
    file:
      path: /tmp/hosts
      state: touch
  - name: add some lines to /tmp/hosts
    blockinfile:
      path: /tmp/hosts
      block: |
        192.168.4.110 host1.example.com
        192.168.4.120 host2.example.com
      state: present
```

Listing 8-8 Resulting File Modification by blockinfile

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.
localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.
localdomain6
192.168.4.200.   control.example.com.      control
192.168.4.201    ansible1.example.com      ansible1
192.168.4.202    ansible2.example.com      ansible2

# BEGIN ANSIBLE MANAGED BLOCK
192.168.4.110 host1.example.com
192.168.4.120 host2.example.com
# END ANSIBLE MANAGED BLOCK
```

Listing 8-9 Creating and Removing Files with the file Module

```
---
- name: using the file module
  hosts: ansible1
  tasks:
  - name: create directory
    file:
      path: /newdir
      owner: ansible
      group: ansible
      mode: 770
      state: directory
  - name: create file in that directory
    file:
      path: /newdir/newfile
      state: touch
  - name: show the new file
    stat:
      path: /newdir/newfile
    register: result
  - debug:
      msg: |
            This shows that newfile was created
            "{{ result }}"
  - name: removing everything again
    file:
      path: /newdir
      state: absent
```

Listing 8-10 Moving a File Around with Ansible

```
---
- name: file copy modules
  hosts: all
  tasks:
  - name: copy file demo
    copy:
      src: /etc/hosts
      dest: /tmp/
  - name: add some lines to /tmp/hosts
    blockinfile:
      path: /tmp/hosts
      block: |
        192.168.4.110 host1.example.com
        192.168.4.120 host2.example.com
      state: present
  - name: verify file checksum
    stat:
      path: /tmp/hosts
      checksum_algorithm: md5
    register: result
  - debug:
      msg: "The checksum of /tmp/hosts is {{ result.stat.checksum }}"
  - name: fetch a file
    fetch:
      src: /tmp/hosts
      dest: /tmp/
```

Listing 8-11 Managing SELinux Context with sefcontext

```
---
- name: show selinux
  hosts: all
  tasks:
  - name: install required packages
    yum:
      name: policycoreutils-python-utils
      state: present
  - name: create testfile
    file:
      name: /tmp/selinux
      state: touch
  - name: set selinux context
    sefcontext:
      target: /tmp/selinux
      setype: httpd_sys_content_t
      state: present
    notify:
      - run restorecon
  handlers:
  - name: run restorecon
    command: restorecon -v /tmp/selinux
```

Listing 8-12 Changing SELinux State and Booleans

```
---
- name: enabling SELinux and a boolean
  hosts: ansible1
  vars:
    myboolean: httpd_read_user_content
  tasks:
  - name: enabling SELinux
    selinux:
      policy: targeted
      state: enforcing
  - name: checking current {{ myboolean }} Boolean status
    shell: getsebool -a | grep {{ myboolean }}
    register: bool_stat
  - name: showing boolean status
    debug:
      msg: the current {{ myboolean }} status is {{ bool_stat.stdout }}
  - name: enabling boolean
    seboolean:
      name: "{{ myboolean }}"
      state: yes
      persistent: yes
```

Listing 8-13 Sample Template

```
# {{ ansible_managed }}

<VirtualHost *:80>
        ServerAdmin webmaster@{{ ansible_facts['fqdn'] }}
        ServerName {{ ansible_facts['fqdn'] }}
        ErrorLog logs/{{ ansible_facts['hostname'] }}-error.log
        CustomLog       logs/{{ ansible_facts['hostname'] }}-common.
log common
        DocumentRoot /var/www/vhosts/{{ ansible_facts['hostname'] }}/

        <Directory /var/www/vhosts/{{ ansible_facts['hostname'] }}>
                Options +Indexes +FollowSymlinks +Includes
                Order allow,deny
                Allow from all
        </Directory>
</VirtualHost>
```

Listing 8-14 Sample Playbook

```
---
- name: installing a template file
  hosts: ansible1
  tasks:
  - name: install http
    yum:
      name: httpd
      state: latest
  - name: start and enable httpd
    service:
      name: httpd
      state: started
      enabled: true
  - name: install vhost config file
    template:
      src: listing813.j2
      dest: /etc/httpd/conf.d/vhost.conf
      owner: root
      group: root
      mode: 0644
  - name: restart httpd
    service:
      name: httpd
      state: restarted
```

Listing 8-15 Exploring Jinja2 **for** Statements

```
{% for node in groups['all'] %}
host_port={{ node }}:8080
{% endfor %}
```

Listing 8-16 Generating a Template with a Conditional Statement

```
---
- name: generate host list
  hosts: ansible2
  tasks:
  - name: template loop
    template:
      src: listing815.j2
      dest: /tmp/hostports.txt
```

Listing 8-17 Template Example with **if**

```
{% if apache_package == 'apache2' %}
  Welcome to Apache2
{% else %}
  Welcome to httpd
{% endif %}
```

Listing 8-18 Using the Template with **if**

```
---
- name: work with template file
  vars:
    apache_package: 'httpd'
  hosts: ansible2
  tasks:
  - template:
      src: listing817.j2
      dest: /tmp/httpd.conf
```

Listings Chapter 9

Using Ansible Roles

Listing 9-1 Roles Sample Directory Structure

```
[ansible@control roles]$ tree testrole/
testrole/
|-- defaults
|   '-- main.yml
|-- files
|-- handlers
|   '-- main.yml
|-- meta
|   '-- main.yml
|-- README.md
|-- tasks
|   '-- main.yml
|-- templates
|-- tests
|   |-- inventory
|   '-- test.yml
'-- vars
    '-- main.yml
```

Listing 9-2 Calling Roles from a Playbook

```
---
- name: include some roles
  roles:
  - role1
  - role2
```

Listing 9-3 Sample motd Role Structure

```
motd
|-- defaults
|   '-- main.yml
|          ---
|          # defaults file for motd
|          system_manager: anna@example.com
|-- meta
|   '-- main.yml
|          galaxy_info:
|          author: Sander van V
|          description: your description
|          company: your company (optional)
|          license: license (GPLv2, CC-BY, etc)
|          min_ansible_version: 2.5
|-- README.md
|-- tasks
|   '-- main.yml
|          ---
|          # tasks file for motd
|          - name: copy motd file
|            template:
|              src: templates/motd.j2
|              dest: /etc/motd
|              owner: root
|              group: root
|              mode: 0444
'-- templates
    '-- motd.j2
           Welcome to {{ ansible_hostname }}

           This file was created on {{ ansible_date_time.date }}
           Disconnect if you have no business being here

           Contact {{ system_manager }} if anything is wrong
```

Listing 9-4 Defining Role Dependencies in meta/main.yml

```
dependencies:
- role: apache
  port: 8080
- role: mariabd
  when: environment == 'production'
```

Listing 9-5 Requesting Additional Information with **ansible-galaxy info**

```
[ansible@control ~]$ ansible-galaxy info geerlingguy.docker

Role: geerlingguy.docker
         description: Docker for Linux.
         active: True
         commit: c94e327a74a16a85f23d73be386c161a9cfe81d4
         commit_message: Allow rc 1 on centos7 when waiting for
systemctl.
         commit_url: https://api.github.com/repos/geerlingguy/
ansible-role-docker/gi>
         company: Midwestern Mac, LLC
         created: 2017-02-24T04:13:02.804883Z
         download_count: 5311691
         forks_count: 404
         github_branch: master
         github_repo: ansible-role-docker
         github_server: https://github.com
         github_user: geerlingguy
         id: 15836
         imported: 2020-04-12T10:53:22.695455-04:00
         is_valid: True
         issue_tracker_url: https://github.com/geerlingguy/
ansible-role-docker/issues
         license: license (BSD, MIT)
         min_ansible_version: 2.4
       modified: 2020-04-12T14:53:22.705998Z
         open_issues_count: 18
         path: ('/home/ansible/.ansible/roles', '/usr/share/ansible/
roles', '/etc/an>
         role_type: ANS
         stargazers_count: 596
         travis_status_url: https://travis-ci.org/geerlingguy/ansible-
role-docker.sv
```

Listing 9-6 Ansible Role Requirements File Example

```
- src: geerlingguy.nginx
  version: "2.7.0"
```

Listing 9-7 Variable Definition in the SELinux System Role

```
---
- hosts: all
  become: true
  become_method: sudo
  become_user: root
  vars:
    selinux_policy: targeted
    selinux_state: enforcing
    selinux_booleans:
      - { name: 'samba_enable_home_dirs', state: 'on' }
      - { name: 'ssh_sysadm_login', state: 'on', persistent: 'yes' }
    selinux_fcontexts:
      - { target: '/tmp/test_dir(/.*)?', setype: 'user_home_dir_t',
ftype: 'd' }
    selinux_restore_dirs:
      - /tmp/test_dir
    selinux_ports:
      - { ports: '22100', proto: 'tcp', setype: 'ssh_port_t', state:
'present' }
    selinux_logins:
      - { login: 'sar-user', seuser: 'staff_u', serange: 's0-s0:c0.
c1023', state: 'present' }
```

Listing 9-8 Sample Playbook That Uses the SELinux RHEL System Role

```
---
- hosts: ansible2
  vars:
    selinux_policy: targeted
    selinux_state: enforcing
    selinux_fcontexts:
      - { target: '/web(/.*)?', setype: 'httpd_sys_content_t', ftype:
'd' }
    selinux_restore_dirs:
      - /web
  # prepare prerequisites which are used in this playbook
  tasks:
    - name: Creates directory
      file:
        path: /web
        state: directory
    - name: execute the role and catch errors
      block:
        - include_role:
            name: rhel-system-roles.selinux
```

Listing 9-9 Rebooting Managed Hosts If Required

```
- name: execute the role and catch errors
      block:
        - include_role:
            name: rhel-system-roles.selinux
      rescue:
        # Fail if failed for a different reason than
selinux_reboot_required.
        - name: handle errors
          fail:
            msg: "role failed"
          when: not selinux_reboot_required

        - name: restart managed host
          shell: sleep 2 && shutdown -r now "Ansible updates
triggered"
          async: 1
          poll: 0
          ignore_errors: true

        - name: wait for managed host to come back
          wait_for_connection:
            delay: 10
            timeout: 300

        - name: reapply the role
          include_role:
            name: rhel-system-roles.selinux
```

Listings Chapter 10

Using Ansible in Large Environments

Listing 10-1 Sample Dynamic Inventory Script

```
#!/usr/bin/python

from subprocess import Popen,PIPE
import sys

try:
     import json
except ImportError:
     import simplejson as json

result = {}
result['all'] = {}

pipe = Popen(['getent', 'hosts'], stdout=PIPE,
universal_newlines=True)

result['all']['hosts'] = []

for line in pipe.stdout.readlines():
    s = line.split()
    result['all']['hosts']=result['all']['hosts']+s

result['all']['vars'] = {}

if len(sys.argv) == 2 and sys.argv[1] == '--list':
    print(json.dumps(result))

elif len(sys.argv) == 3 and sys.argv[1] == '--host':
    print(json.dumps({}))

else:
    print("Requires an argument, please use --list or --host <host>")
```

Listing 10-2 JSON Output of the listing101.py Script

```
[ansible@control rhce8-book]$ ./listing101.py --list
{"all": {"hosts": ["127.0.0.1", "localhost", "localhost.localdomain",
"localhost4", "localhost4.localdomain4", "127.0.0.1", "localhost",
"localhost.localdomain", "localhost6", "localhost6.localdomain6",
"192.168.4.200", "control.example.com", "control", "192.168.4.201",
"ansible1.example.com", "ansible1", "192.168.4.202", "ansible2.
example.com", "ansible2"], "vars": {}}}
```

Listing 10-3 Showing Output of the **ansible-inventory** Command

```
[ansible@control rhce8-book]$ ansible-inventory -i listing101.py
--graph
[WARNING]: A duplicate localhost-like entry was found (localhost).
First found
localhost was 127.0.0.1
@all:
  |--@ungrouped:
  |  |--127.0.0.1
  |  |--192.168.4.200
  |  |--192.168.4.201
  |  |--192.168.4.202
  |  |--ansible1
  |  |--ansible1.example.com
  |  |--ansible2
  |  |--ansible2.example.com
  |  |--control
  |  |--control.example.com
  |  |--localhost
  |  |--localhost.localdomain
  |  |--localhost4
  |  |--localhost4.localdomain4
  |  |--localhost6
  |  |--localhost6.localdomain6
```

Listing 10-4 Sample Playbook to Be Imported

```
- hosts: all
  tasks:
  - debug:
      msg: running the imported play
```

Listing 10-5 Importing a Playbook

```
---
- name: run a task
  hosts: all
  tasks:
  - debug:
      msg: running task1

- name: importing a playbook
  import_playbook: listing104.yaml
```

Listing 10-6 Running **ansible-playbook listing105.yaml** Result

```
[ansible@control rhce8-book]$ ansible-playbook listing105.yaml

PLAY [run a task] *****************************************************
**********

TASK [Gathering Facts] ************************************************
**********
ok: [ansible2]
ok: [ansible1]
ok: [ansible3]
ok: [ansible4]

TASK [debug] **********************************************************
**********
ok: [ansible1] => {
    "msg": "running task1"
}
ok: [ansible2] => {
    "msg": "running task1"
}
ok: [ansible3] => {
    "msg": "running task1"
}
ok: [ansible4] => {
    "msg": "running task1"
}
```

```
PLAY [all] ***********************************************************
**********

TASK [Gathering Facts] ***********************************************
**********
ok: [ansible2]
ok: [ansible1]
ok: [ansible3]
ok: [ansible4]

TASK [debug] *********************************************************
**********
ok: [ansible1] => {
    "msg": "running the imported play"
}
ok: [ansible2] => {
    "msg": "running the imported play"
}
ok: [ansible3] => {
    "msg": "running the imported play"
}
ok: [ansible4] => {
    "msg": "running the imported play"
}

PLAY RECAP ***********************************************************
**********
ansible1                     : ok=4    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                     : ok=4    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible3                     : ok=4    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible4                     : ok=4    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 10-7 The Include Tasks File tasks/service.yaml Used for Services Definition

```
- name: install {{ package }}
  yum:
    name: "{{ package }}"
    state: latest
- name: start {{ service }}
  service:
    name: "{{ service }}"
    enabled: true
    state: started
```

Listing 10-8 The Import Tasks File tasks/firewall.yaml Used for Firewall Definition

```
- name: install the firewall
  package:
    name: "{{ firewall_package }}"
    state: latest
- name: start the firewall
  service:
    name: "{{ firewall_service }}"
    enabled: true
    state: started
- name: open the port for the service
  firewalld:
    service: "{{ item }}"
    immediate: true
    permanent: true
    state: enabled
  loop: "{{ firewall_rules }}"
```

Listing 10-9 Main Playbook Example

```
---
- name: setup a service
  hosts: ansible2
  tasks:
  - name: include the services task file
    include_tasks: tasks/service.yaml
    vars:
      package: httpd
      service: httpd
```

```
   when: ansible_facts['os_family'] == 'RedHat'
  - name: import the firewall file
    import_tasks: tasks/firewall.yaml
    vars:
      firewall_package: firewalld
      firewall_service: firewalld
      firewall_rules:
      - http
      - https
```

Listing 10-10 Running **ansible-playbook listing109.yaml**

```
[ansible@control rhce8-book]$ ansible-playbook listing109.yaml

PLAY [setup a service] ***********************************************

TASK [Gathering Facts] ***********************************************
ok: [ansible2]

TASK [include the services task file] ********************************
included: /home/ansible/rhce8-book/tasks/service.yaml for ansible2

TASK [install httpd] *************************************************
ok: [ansible2]

TASK [start httpd] ***************************************************
changed: [ansible2]

TASK [install the firewall] ******************************************
changed: [ansible2]

TASK [start the firewall] ********************************************
ok: [ansible2]

TASK [open the port for the service] *********************************
changed: [ansible2] => (item=http)
changed: [ansible2] => (item=https)

PLAY RECAP ***********************************************************
ansible2                     : ok=7    changed=3    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listings Chapter 11

Troubleshooting Ansible

Listing 11-1 Sample Playbook

```
---
- name: simple template example
  hosts: ansible2
  tasks:
  - template:
      src: listing112.j2
      dest: /etc/issue
```

Listing 11-2 Sample Template File

```
{# /etc/issue #}
Welcome to {{ ansible_facts['hostname'] }}
```

Listing 11-3 Running the listing111.yaml Sample Playbook

```
[ansible@control rhce8-book]$ ansible-playbook listing111.yaml --check
--diff

PLAY [simple template example] ***************************************

TASK [Gathering Facts] ***********************************************
ok: [ansible2]

TASK [template] ******************************************************
--- before
+++ after: /home/ansible/.ansible/tmp/ansible-local-4493uxbpju1e/
tmpm5gn7crg/listing112.j2
@@ -0,0 +1,3 @@
+Welcome to ansible2
+
+
changed: [ansible2]

PLAY RECAP ***********************************************************
ansible2                     : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 11-4 ansible-playbook Command Output

```
[ansible@control rhce8-book]$ ansible-playbook listing52.yaml

PLAY [install start and enable httpd] ********************************

TASK [Gathering Facts] ***********************************************
ok: [ansible2]
ok: [ansible1]
ok: [ansible3]
ok: [ansible4]

TASK [install package] ***********************************************
changed: [ansible2]
changed: [ansible1]
changed: [ansible3]
changed: [ansible4]

TASK [start and enable service] **************************************
changed: [ansible2]
changed: [ansible1]
changed: [ansible3]
changed: [ansible4]

PLAY RECAP ***********************************************************
ansible1                     : ok=3    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                     : ok=3    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible3                     : ok=3    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible4                     : ok=3    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 11-5 Analyzing Partial **-vvvv** Output

```
<ansible4> ESTABLISH SSH CONNECTION FOR USER: ansible
<ansible4> SSH: EXEC ssh -vvv -C -o ControlMaster=auto
-o ControlPersist=60s -o StrictHostKeyChecking=no -o
KbdInteractiveAuthentication=no -o PreferredAuthentications=gssapi-
with-mic,gssapi-keyex,hostbased,publickey -o PasswordAuthentication=no
-o 'User="ansible"' -o ConnectTimeout=10 -o ControlPath=/
home/ansible/.ansible/cp/859d5267e3 ansible4 '/bin/
sh -c '"'"'chmod u+x /home/ansible/.ansible/tmp/ansible-
tmp-1587544652.4716983-118789810824208/ /home/ansible/.ansible/tmp/
ansible-tmp-1587544652.4716983-118789810824208/AnsiballZ_systemd.py &&
sleep 0'"'"''
Escalation succeeded
<ansible4> (0, b'', b"OpenSSH_8.0p1, OpenSSL 1.1.1c FIPS  28 May
2019\r\ndebug1: Reading configuration data /etc/ssh/ssh_config\r\
ndebug3: /etc/ssh/ssh_config line 51: Including file /etc/ssh/
ssh_config.d/05-redhat.conf depth 0\r\ndebug1: Reading configuration
data /etc/ssh/ssh_config.d/05-redhat.conf\r\ndebug2: checking match
for 'final all' host ansible4 originally ansible4\r\ndebug3: /
etc/ssh/ssh_config.d/05-redhat.conf line 3: not matched 'final'\r\
ndebug2: match not found\r\ndebug3: /etc/ssh/ssh_config.d/05-redhat.
conf line 5: Including file /etc/crypto-policies/back-ends/openssh.
config depth 1 (parse only)\r\ndebug1: Reading configuration data /
etc/crypto-policies/back-ends/openssh.config\r\ndebug3: gss kex
names ok: [gss-gex-sha1-,gss-group14-sha1-]\r\ndebug3: kex names
ok: [curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-
nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-
exchange-sha256,diffie-hellman-group14-sha256,diffie-hellman-group16-
sha512,diffie-hellman-group18-sha512,diffie-hellman-group-exchange-
sha1,diffie-hellman-group14-sha1]\r\ndebug1: configuration requests
final Match pass\r\ndebug1: re-parsing configuration\r\ndebug1:
Reading configuration data /etc/ssh/ssh_config\r\ndebug3: /etc/ssh/
ssh_config line 51: Including file /etc/ssh/ssh_config.d/05-redhat.
conf depth 0\r\ndebug1: Reading configuration data /etc/ssh/ssh_
config.d/05-redhat.conf\r\ndebug2: checking match for 'final all'
host ansible4 originally ansible4\r\ndebug3: /etc/ssh/ssh_config.d/05-
redhat.conf line 3: matched 'final'\r\ndebug2: match found\r\ndebug3:
/etc/ssh/ssh_config.d/05-redhat.conf line 5: Including file /etc/
crypto-policies/back-ends/openssh.config depth 1\r\ndebug1: Reading
configuration data /etc/crypto-policies/back-ends/openssh.config\r\
ndebug3: gss kex names ok: [gss-gex-sha1-,gss-group14-sha1-]\r\
ndebug3: kex names ok: [curve25519-sha256,curve25519-sha256@libssh.
org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-
hellman-group-exchange-sha256,diffie-hellman-group14-sha256,diffie-
hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-
group-exchange-sha1,diffie-hellman-group14-sha1]\r\ndebug1: auto-mux:
Trying existing master\r\ndebug2: fd 4 setting O_NONBLOCK\r\ndebug2:
mux_client_hello_exchange: master version 4\r\ndebug3: mux_client_
forwards: request forwardings: 0 local, 0 remote\r\ndebug3: mux_
client_request_session: entering\r\ndebug3: mux_client_request_alive:
entering\r\ndebug3: mux_client_request_alive: done pid = 4764\r\
ndebug3: mux_client_request_session: session request sent\r\ndebug3:
mux_client_read_packet: read header failed: Broken pipe\r\ndebug2:
Received exit status from master 0\r\n")
```

```
<ansible4> ESTABLISH SSH CONNECTION FOR USER: ansible
<ansible4> SSH: EXEC ssh -vvv -C -o ControlMaster=auto
-o ControlPersist=60s -o StrictHostKeyChecking=no -o
KbdInteractiveAuthentication=no -o PreferredAuthentications=gssapi-
with-mic,gssapi-keyex,hostbased,publickey -o PasswordAuthentication=no
-o 'User="ansible"' -o ConnectTimeout=10 -o ControlPath=/home/
ansible/.ansible/cp/859d5267e3 -tt ansible4 '/bin/sh -c '"'"'sudo
-H -S -n  -u root /bin/sh -c '"'"'"'"'"'"'"'"'echo BECOME-SUCCESS-
muvtpdvqkslnlegyhoibfcrilvlyjcqp ; /usr/libexec/platform-python /home/
ansible/.ansible/tmp/ansible-tmp-1587544652.4716983-118789810824208/
AnsiballZ_systemd.py'"'"'"'"'"'"'"'"' && sleep 0'"'"''
Escalation succeeded
```

Listing 11-6 Running Tasks One by One

```
[ansible@control rhce8-book]$ ansible-playbook --list-tasks
exercise81.yaml

playbook: exercise81.yaml

  play #1 (ansible1): testing file manipulation skills.    TAGS: []
    tasks:
      create a new file                TAGS: []
      check status of the new file    TAGS: []
      for debugging purposes only     TAGS: []
      change file owner if needed     TAGS: []

  play #2 (ansible1): fetching a remote file.    TAGS: []
    tasks:
      fetch file from remote machine.    TAGS: []

  play #3 (localhost): adding text to the file that is now on
localhost TAGS: []
    tasks:
      add a message.    TAGS: []

  play #4 (ansible2): copy the modified file to ansible2.    TAGS: []
    tasks:
      copy motd file.    TAGS: []
[ansible@control rhce8-book]$ ansible-playbook --start-at-task "add a
message"  --step exercise81.yaml
```

```
PLAY [testing file manipulation skills] ******************************

PLAY [fetching a remote file] ****************************************

PLAY [adding text to the file that is now on localhost]
************************
Perform task: TASK: Gathering Facts (N)o/(y)es/(c)ontinue:
```

Listing 11-7 Using the uri Module

```
---
- name: test webserver access
  hosts: localhost
  become: no
  tasks:
  - name: connect to the web server
    uri:
      url: http://ansible2.example.com
      return_content: yes
    register: this
    failed_when: "'welcome' not in this.content"
  - debug:
      var: this.content
```

Listing 11-8 **ansible-playbook listing117.yaml** Command Result

```
[ansible@control rhce8-book]$ ansible-playbook listing117.yaml

PLAY [test webserver access] *****************************************

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

TASK [connect to the web server] *************************************
ok: [localhost]

TASK [debug] *********************************************************
ok: [localhost] => {
    "this.content": "<?xml version=\"1.0\" encoding=\"utf-8\"?>\
n<!DOCTYPE HTML>\n<html lang=\"en\">\n  <head>\n    <title>CentOS
Apache HTTP </title>\n    <meta charset=\"utf-8\"/>\n    <meta
name=\"viewport\" content=\"width=device-width, initial-scale=1,
shrink-to-fit=no\"/>\n    <link rel=\"shortcut icon\" href=\"http://
www.centos.org/favicon.ico\"/>\n    <link rel=\"stylesheet\"
media=\"all\" href=\"noindex/common/css/bootstrap.min.
css\"/>\n    <link rel=\"stylesheet\" media=\"all\" href=\"noindex/
common/css/styles.css\"/>\n  </head>\n  <body>\n    <header
class=\"container\">\n      <section class=\"row\">\n        <div
class=\"header-graphic v3-banner platform-banner centos-banner\"
role=\"banner\">\n          <div class=\"graphic-inner\">\n
<div class=\"graphic-inner2\">\n              <div class=\"banner-...
href=\"https://www.centos.org/legal/\"></a> | <a href=\"https://www.
centos.org/legal/privacy/\"></a></p>\n    </footer>\n  </body>\n</
html>\n"
}

PLAY RECAP *********************************************************
localhost                  : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 11-9 Using stat to Check Expected File Status

```
---
- name: create a file
  hosts: all
  tasks:
  - file:
      path: /tmp/statfile
      state: touch
      owner: ansible

- name: check file status
  hosts: all
  tasks:
  - stat:
      path: /tmp/statfile
    register: stat_out
  - fail:
      msg: "/tmp/statfile file owner not as expected"
    when: stat_out.stat.pw_name != 'root'
```

Listing 11-10 ansible-playbook listing119.yaml Command Result

```
[ansible@control rhce8-book]$ ansible-playbook listing119.yaml

PLAY [create a file] *************************************************

TASK [Gathering Facts] ***********************************************
ok: [ansible2]
ok: [ansible1]
ok: [ansible3]
ok: [ansible4]
fatal: [ansible6]: UNREACHABLE! => {"changed": false, "msg": "Failed
to connect to the host via ssh: ansible@ansible6: Permission denied
(publickey,gssapi-keyex,gssapi-with-mic,password).", "unreachable":
true}
fatal: [ansible5]: UNREACHABLE! => {"changed": false, "msg": "Failed
to connect to the host via ssh: ssh: connect to host ansible5 port
22: No route to host", "unreachable": true}

TASK [file] **********************************************************
changed: [ansible2]
changed: [ansible1]
changed: [ansible3]
changed: [ansible4]

PLAY [check file status] *********************************************

TASK [Gathering Facts] ***********************************************
ok: [ansible1]
ok: [ansible2]
ok: [ansible3]
ok: [ansible4]

TASK [stat] **********************************************************
ok: [ansible2]
ok: [ansible1]
ok: [ansible3]
ok: [ansible4]

TASK [fail] **********************************************************
fatal: [ansible2]: FAILED! => {"changed": false, "msg": "/tmp/statfile
file owner not as expected"}
```

```
fatal: [ansible1]: FAILED! => {"changed": false, "msg": "/tmp/statfile
file owner not as expected"}
fatal: [ansible3]: FAILED! => {"changed": false, "msg": "/tmp/statfile
file owner not as expected"}
fatal: [ansible4]: FAILED! => {"changed": false, "msg": "/tmp/statfile
file owner not as expected"}

PLAY RECAP **********************************************************
ansible1                    : ok=4    changed=1    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
ansible2                    : ok=4    changed=1    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
ansible3                    : ok=4    changed=1    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
ansible4                    : ok=4    changed=1    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
ansible5                    : ok=0    changed=0    unreachable=1
failed=0    skipped=0    rescued=0    ignored=0
ansible6                    : ok=0    changed=0    unreachable=1
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 11-11 Using the assert Module

```
---
- hosts: localhost
  vars_prompt:
  - name: filesize
    prompt: "specify a file size in megabytes"
  tasks:
  - name: check if file size is valid
    assert:
      that:
      - "{{ (filesize | int) <= 100 }}"
      - "{{ (filesize | int) >= 1 }}"
      fail_msg: "file size must be between 0 and 100"
      success_msg: "file size is good, let\'s continue"
  - name: create a file
    command: dd if=/dev/zero of=/bigfile bs=1 count={{ filesize }}
```

Listing 11-12 ansible-playbook listing1111.yaml Output

```
[ansible@control rhce8-book]$ ansible-playbook listing1111.yaml

PLAY [localhost] *****************************************************

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

TASK [check if file size is valid] ***********************************
fatal: [localhost]: FAILED! => {
    "assertion": "filesize <= 100",
    "changed": false,
    "evaluated_to": false,
    "msg": "file size must be between 0 and 100"
}

PLAY RECAP ***********************************************************
localhost                    : ok=1    changed=0    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
```

Listing 11-13 Failing Version of the Listing 11-11 Playbook

```
---
- hosts: localhost
  vars_prompt:
  - name: filesize
    prompt: "specify a file size in megabytes"
  tasks:
  - name: check if file size is valid
    assert:
      that:
      - filesize <= 100
      - filesize >= 1
      fail_msg: "file size must be between 0 and 100"
      success_msg: "file size is good, let\'s continue"
  - name: create a file
    command: dd if=/dev/zero of=/bigfile bs=1 count={{ filesize }}
```

Listing 11-14 ansible-playbook listing1113.yaml Failing Result

```
[ansible@control rhce8-book]$ ansible-playbook listing1113.yaml
specify a file size in megabytes:

PLAY [localhost] *****************************************************

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

TASK [check if file size is valid] ***********************************
fatal: [localhost]: FAILED! => {"msg": "The conditional check
'filesize <= 100' failed. The error was: Unexpected templating type
error occurred on ({% if filesize <= 100 %} True {% else %} False {%
endif %}): '<=' not supported between instances of 'str' and 'int'"}

PLAY RECAP ***********************************************************
localhost                    : ok=1    changed=0    unreachable=0
failed=1    skipped=0     rescued=0    ignored=0
```

Listing 11-15 Using **tags** in a Playbook

```
---
- name: using tags example
  hosts: all
  vars:
    services:
    - vsftpd
    - httpd
  tasks:
  - yum:
      name:
      - httpd
      - vsftpd
      state: present
    tags:
    - install
  - service:
      name: "{{ item }}"
      state: started
      enabled: yes
    loop: "{{ services }}"
    tags:
    - services
```

Listing 11-16 ansible-playbook --tags "install" listing1115.yaml Output

```
[ansible@control rhce8-book]$ ansible-playbook --tags "install"
listing1115.yaml

PLAY [using tags example] ********************************************

TASK [Gathering Facts] ***********************************************
ok: [ansible2]
ok: [ansible1]
ok: [ansible4]
ok: [ansible3]

TASK [yum] ***********************************************************
ok: [ansible2]
ok: [ansible1]
changed: [ansible3]
changed: [ansible4]

PLAY RECAP ***********************************************************
ansible1                     : ok=2     changed=0    unreachable=0
failed=0     skipped=0    rescued=0     ignored=0
ansible2                     : ok=2     changed=0    unreachable=0
failed=0     skipped=0    rescued=0     ignored=0
ansible3                     : ok=2     changed=1    unreachable=0
failed=0     skipped=0    rescued=0     ignored=0
ansible4                     : ok=2     changed=1    unreachable=0
failed=0     skipped=0    rescued=0     ignored=0
```

Listing 11-17 Listing Tasks and Tags

```
[ansible@control rhce8-book]$ ansible-playbook --list-tags --list-
tasks listing1115.yaml

playbook: listing1115.yaml

  play #1 (all): using tags example.    TAGS: []
    tasks:
      yum.        TAGS: [install]
      service.    TAGS: [services]
      TASK TAGS: [install, services]
```

Listing 11-18 Verifying Connectivity Using the ping Module

```
[ansible@control rhce8-book]$ ansible all -m ping
ansible2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/
platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ansible1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/
platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ansible3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/
platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ansible4 | FAILED! => {
    "msg": "Missing sudo password"
}
```

Listings Chapter 12

Managing Software with Ansible

Listing 12-1 Configuring Repository Access

```
---
- name: setting up repository access
  hosts: all
  tasks:
  - name: connect to example repo
    yum_repository:
      name: example repo
      description: RHCE8 example repo
      file: examplerepo
      baseurl: ftp://control.example.com/repo/
      gpgcheck: no
```

Listing 12-2 Using yum to Perform a System Update

```
---
- name: updating all packages
  hosts: ansible2
  tasks:
  - name: system update
    yum:
      name: '*'
      state: latest
```

Listing 12-3 Installing Package Groups

```
---
- name: install or update a package group
  hosts: ansible2
  tasks:
  - name: install or update a package group
    yum:
      name: '@Virtualization Host'
      state: latest
```

Listing 12-4 Installing Yum AppStream Modules with the Ansible yum Module

```
---
- name: installing an AppStream module
  hosts: ansible2
  tasks:
  - name: install or update an AppStream module
    yum:
      name: '@php:7.3/devel'
      state: present
```

Listing 12-5 Using the package_facts Module to Show Package Details

```
---
- name: using package facts
  hosts: ansible2
  vars:
    my_package: nmap
  tasks:
  - name: install package
    yum:
      name: "{{ my_package }}"
      state: present
  - name: update package facts
    package_facts:
      manager: auto
  - name: show package facts for {{ my_package }}
    debug:
      var: ansible_facts.packages[my_package]
    when: my_package in ansible_facts.packages
```

Listing 12-6 Running **ansible-playbook listing125.yaml** Results

```
[ansible@control rhce8-book]$ ansible-playbook listing125.yaml

PLAY [using package facts] ******************************************

TASK [Gathering Facts] **********************************************
ok: [ansible2]

TASK [install package] **********************************************
ok: [ansible2]
```

```
TASK [update package facts] ******************************************
ok: [ansible2]

TASK [show package facts for my_package] *****************************
ok: [ansible2] => {
    "ansible_facts.packages[my_package]": [
        {
            "arch": "x86_64",
            "epoch": 2,
            "name": "nmap",
            "release": "5.el8",
            "source": "rpm",
            "version": "7.70"
        }
    ]
}

PLAY RECAP ***********************************************************
ansible2                     : ok=4    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listing 12-7 Setting Up an FTP-based Repository

```
- name: install FTP to export repo
  hosts: localhost
  tasks:
  - name: install FTP server
    yum:
      name:
      - vsftpd
      - createrepo_c
      state: latest
  - name: start FTP server
    service:
      name: vsftpd
      state: started
      enabled: yes
```

```
 - name: open firewall for FTP
    firewalld:
      service: ftp
      state: enabled
      permanent: yes

- name: setup the repo directory
  hosts: localhost
  tasks:
  - name: make directory
    file:
      path: /var/ftp/repo
      state: directory
  - name: download packages
    yum:
      name: nmap
      download_only: yes
      download_dir: /var/ftp/repo
  - name: createrepo
    command: createrepo /var/ftp/repo
```

Listing 12-8 Using rpm_key to Fetch an RPM Key

```
- name: use rpm_key in repository access
  hosts: all
  tasks:
  - name: get the GPG public key
    rpm_key:
      key: ftp://control.example.com/repo/RPM-GPG-KEY
      state: present
  - name: set up the repository client
    yum_repository:
      file: myrepo
      name: myrepo
      description: example repo
      baseurl: ftp://control.example.com/repo
      enabled: yes
      gpgcheck: yes
      state: present
```

Listing 12-9 Using Subscription Manager to Set Up Ansible

```
---
- name: use subscription manager to register and set up repos
  hosts: ansible5
  tasks:
  - name: register and subscribe ansible5
    redhat_subscription:
    username: bob@example.com
      password: verysecretpassword
      state: present
  - name: configure additional repo access
    rhsm_repository:
      name:
      - rh-gluster-3-client-for-rhel-8-x86_64-rpms
      - rhel-8-for-x86_64-appstream-debug-rpms
      state: present
```

Listings Chapter 13

Managing Users

Listing 13-1 Managing Users and Groups

```
---
- name: creating a user and group
  hosts: ansible2
  tasks:
  - name: setup the group account
    group:
      name: students
      state: present
  - name: setup the user account
    user:
      name: anna
      create_home: yes
      groups: wheel,students
      append: yes
      generate_ssh_key: yes
      ssh_key_bits: 2048
      ssh_key_file: .ssh/id_rsa
```

Listing 13-2 Variables for Managing sudo

```
[ansible@control rhce8-book]$ cat vars/sudo
sudo_groups:
  - name: developers
    groupid: 5000
    sudo: false
  - name: admins
    groupid: 5001
    sudo: true
  - name: dbas
    groupid: 5002
    sudo: false
  - name: sales
    groupid: 5003
    sudo: true
  - name: account
    groupid: 5004
    sudo: false
[ansible@control rhce8-book]$ cat vars/users
users:
  - username: linda
    groups: sales
  - username: lori
    groups: sales
  - username: lisa
    groups: account
  - username: lucy
    groups: account
```

Listing 13-3 Template File for Managing sudo

```
{% for item in sudo_groups %}
{% if item.sudo %}
%{{ item.name}} ALL=(ALL:ALL) NOPASSWD:ALL
{% endif %}
{% endfor %}
```

Listing 13-4 Managing sudo

```
---
- name: configure sudo
  hosts: ansible2
  vars_files:
    - vars/sudo
    - vars/users
  tasks:
  - name: add groups
    group:
      name: "{{ item.name }}"
    loop: "{{ sudo_groups }}"
  - name: add users
    user:
      name: "{{ item.username }}"
      groups: "{{ item.groups }}"
    loop: "{{ users }}"
  - name: allow group members in sudo
    template:
      src: listing133.j2
      dest: /etc/sudoers.d/sudogroups
      validate: 'visudo -cf %s'
      mode: 0440
```

Listing 13-5 Setting Variable Value to File Contents with the Lookup Plug-in

```
---
- name: simple demo with the lookup plugin
  hosts: localhost
  vars:
    file_contents: "{{lookup('file', '/etc/hosts')}}"
  tasks:
  - debug:
      var: file_contents
```

Listing 13-6 Using the authorized_key Module

```
---
- name: authorized_key simple demo
  hosts: ansible2
  tasks:
  - name: copy authorized key for ansible user
    authorized_key:
      user: ansible
      state: present
      key: "{{ lookup('file', '/home/ansible/.ssh/id_rsa.pub') }}"
```

Listing 13-7 Variable Files Used in Listing 13-8

```
[ansible@control rhce8-book]$ cat vars/users
---
users:
  - username: linda
    groups: sales
  - username: lori
    groups: sales
  - username: lisa
    groups: account
```

```
 - username: lucy
    groups: account
[ansible@control rhce8-book]$ cat vars/groups
---
usergroups:
  - groupname: sales
  - groupname: account
```

Listing 13-8 Using the authorized_key Module with the Lookup Plug-in

```
---
- name: configure users with SSH keys
  hosts: ansible2
  vars_files:
    - vars/users
    - vars/groups
  tasks:
  - name: add groups
    group:
      name: "{{ item.groupname }}"
    loop: "{{ usergroups }}"
  - name: add users
    user:
      name: "{{ item.username }}"
      groups: "{{ item.groups }}"
    loop: "{{ users }}"
  - name: add SSH public keys
    authorized_key:
      user: "{{ item.username }}"
      key: "{{ lookup('file', 'files/'+ item.username +
'/id_rsa.pub') }}"
    loop: "{{ users }}"
```

Listing 13-9 Creating the User with SSH Key

```
- name: create the local user, including SSH key
  user:
    name: "{{ username }}"
    generate_ssh_key: true
    ssh_key_comment: "{{ username }}@{{ ansible_fqdn }}"
```

Listing 13-10 Copying the User SSH Public Key to a Location from Which It Can Be Copied

```
- name: create a directory to store the file
  file:
    name: "{{ username }}"
    state: directory
- name: copy the local user ssh key to temporary {{ username }} key
  shell: 'cat /home/{{ username }}/.ssh/id_rsa.pub > {{ username }}/
id_rsa.pub'
- name: verify that file exists
  command: ls -l {{ username }}/
```

Listing 13-11 Generating the Encrypted Password String

```
[ansible@control ~]$ ansible localhost -m debug -a "msg={{ 'password'
| password_hash('sha512','myrandomsalt') }}"
localhost | SUCCESS => {
    "msg": "$6$myrandomsalt$McEB.xAVUWe0./6XqZ8n/7k9VV/
Gxndy9nIMLyQAiPnhyBoToMWbxX2vA4f.Uv9PKnPRaYUUc76AjLWVAX6U10"
}
```

Listing 13-12 Sample Playbook That Creates an Encrypted User Password

```
---
- name: create user with encrypted password
  hosts: ansible2.example.com
  vars:
    password: "$6$myrandomsalt$McEB.xAVUWe0./6XqZ8n/7k9VV/
Gxndy9nIMLyQAiPnhyBoToMWbxX2vA4f.Uv9PKnPRaYUUc76AjLWVAX6U10"
  tasks:
  - name: create the user
    user:
      name: anna
      password: "{{ password }}"
```

Listing 13-13 Setting the User Password: Alternative Solution

```
---
- name: create user with encrypted password
  hosts: ansible3
  vars:
    password: mypassword
    user: anna
  tasks:
  - name: configure user {{ user }}
    user:
      name: "{{ user }}"
      groups: wheel
      append: yes
      state: present
  - name: set a password for {{ user }}
    shell: 'echo {{ password }} | passwd --stdin {{ user }}'
```

Listing 13-14 Finding the Variable Name Using debug

```
TASK [debug] ********************************************************
**********
ok: [ansible2] => {
    "mypass": {
      "changed": false,
        "failed": false,
        "msg": "$6$myrandomsalt$Jesm4QGoCGAny9ebP85apmh0/
uUXrj0louYb03leLoOWSDy/imjVGmcODhrpIJZt0rz.GBp9pZYpfm0SU2/PO."
    }
}
```

Listing 13-15 Task 10 Error Output

```
TASK [copy authorized keys] ****************************************
[WARNING]: Unable to find '/home/laksmi/id_rsa.pub' in expected paths
(use -vvvvv to see paths)
fatal: [ansible3]: FAILED! => {"msg": "An unhandled exception occurred
while running the lookup plugin 'file'. Error was a <class 'ansible.
errors.AnsibleError'>, original message: could not locate file in
lookup: /home/laksmi/id_rsa.pub"}
```

Listings Chapter 14

Managing Services and the Boot Process

Listing 14-1 Using systemd Module Features

```
---
- name: using systemd module to manage services
  hosts: ansible2
  tasks:
  - name: enable service httpd and ensure it is not masked
    systemd:
      name: httpd
      enabled: yes
      state: started
      masked: no
      daemon_reload: yes
```

Listing 14-2 Running a cron Job

```
---
- name: run a cron job
  hosts: ansible2
  tasks:
- name: run a periodic job
    cron:
      name: "run fstrim"
      minute: "5"
      hour: "4,19"
      job: "fstrim"
```

Listing 14-3 Removing a cron Job Using the **name** Attribute

```
---
- name: run a cron job
  hosts: ansible2
  tasks:
  - name: run a periodic job
    cron:
      name: "run fstrim"
      state: absent
```

Listing 14-4 Running Commands in the Future with at

```
---
- name: run an at task
  hosts: ansible2
  tasks:
    - name: run command and write output to file
      at:
        command: "date > /tmp/my-at-file"
        count: 5
        units: minutes
        unique: yes
        state: present
```

Listing 14-5 Showing the Default Systemd Target

```
[ansible@control rhce8-book]$ ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 37 Mar 23 05:33 /etc/systemd/system/default.
target -> /lib/systemd/system/multi-user.target
```

Listing 14-6 Managing the Default Boot Target

```
---
- name: set default boot target
  hosts: ansible2
  tasks:
    - name: set boot target to graphical
      file:
        src: /usr/lib/systemd/system/graphical.target
        dest: /etc/systemd/system/default.target
        state: link
```

Listing 14-7 Rebooting Managed Hosts

```
---
- name: reboot all hosts
  hosts: all
  gather_facts: no
  tasks:
  - name: reboot hosts
    reboot:
      msg: reboot initiated by Ansible
      test_command: whoami
  - name: print message to show host is back
    debug:
      msg: successfully rebooted
```

Listing 14-8 Verifying the Success of the reboot Module

```
[ansible@control rhce8-book]$ ansible-playbook listing147.yaml

PLAY [reboot all hosts] ***********************************************

TASK [reboot hosts] ***************************************************
changed: [ansible2]
changed: [ansible1]
changed: [ansible3]
changed: [ansible4]
changed: [ansible5]

TASK [print message to show host is back] *****************************
ok: [ansible1] => {
    "msg": "successfully rebooted"
}
ok: [ansible2] => {
    "msg": "successfully rebooted"
}
ok: [ansible3] => {
    "msg": "successfully rebooted"
}
ok: [ansible4] => {
    "msg": "successfully rebooted"
}
ok: [ansible5] => {
    "msg": "successfully rebooted"
}
```

```
PLAY RECAP **********************************************************
ansible1                    : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible2                    : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible3                    : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible4                    : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible5                    : ok=2    changed=1    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Listings Chapter 15

Managing Storage

Listing 15-1 Discovering Storage Facts Using **ansible -m setup**

```
[ansible@control rhce8-book]$ ansible ansible1 -m setup -a
'filter=ansible_devices'
ansible1 | SUCCESS => {
    "ansible_facts": {
        "ansible_devices": {
            ...
            "sda": {
                "holders": [],
                "host": "SCSI storage controller: Broadcom / LSI
53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI (rev 01)",
                "links": {
                    "ids": [],
                    "labels": [],
                    "masters": [],
                    "uuids": []
                },
                "model": "VMware Virtual S",
                "partitions": {
                    "sda1": {
                        "holders": [],
                        "links": {
                            "ids": [],
                            "labels": [],
                            "masters": [],
                            "uuids": [
                                "7acd65d6-115f-499f-a02f-
90364a18b9fc"
                           ]
                        },
                        "sectors": "2097152",
                        "sectorsize": 512,
                        "size": "1.00 GB",
                        "start": "2048",
                        "uuid":
"7acd65d6-115f-499f-a02f-90364a18b9fc"
                    },
                    "sda2": {
                        "holders": [
                            "cl-swap",
                            "cl-root"
                        ],
                        "links": {
                            "ids": [
                                "lvm-pv-uuid-vswoyx-HihU-cRqK-2tvx-
aesr-SXAR-YY1lLL"
                            ],
                            "labels": [],
                            "masters": [
                                "dm-0",
                                "dm-1"
                            ],
                            "uuids": []
                        },
                        "sectors": "39843840",
                        "sectorsize": 512,
                        "size": "19.00 GB",
                        "start": "2099200",
                        "uuid": null
                    }
                },
                "removable": "0",
                "rotational": "1",
                "sas_address": null,
                "sas_device_handle": null,
                "scheduler_mode": "mq-deadline",
                "sectors": "41943040",
                "sectorsize": "512",
                "size": "20.00 GB",
                "support_discard": "0",
                "vendor": "VMware,",
                "virtual": 1
            },
            "sr0": {
...
            }
        },
        "discovered_interpreter_python": "/usr/libexec/
platform-python"
    },
    "changed": false
}
```

Listing 15-2 Using assert to Run a Task Only If a Device Exists

```
---
- name: search for /dev/sdb continue only if it is found
  hosts: all
  vars:
    disk_name: sdb
  tasks:
  - name: abort if second disk does not exist
    assert:
      that:
        - "ansible_facts['devices']['{{ disk_name }}'] is defined"
      fail_msg: second hard disk not found
  - debug:
      msg: "{{ disk_name }} was found, lets continue"
```

Listing 15-3 ansible-playbook listing152.yaml Result

```
[ansible@control rhce8-book]$ ansible-playbook listing152.yaml

PLAY [search for /dev/sdb continue only if it is found] *************

TASK [Gathering Facts] **********************************************
ok: [ansible2]
ok: [ansible1]
ok: [ansible3]
ok: [ansible4]
ok: [ansible5]

TASK [abort if second disk does not exist] **************************
```

```
fatal: [ansible1]: FAILED! => {
    "assertion": "ansible_facts['devices']['sdb'] is defined",
    "changed": false,
    "evaluated_to": false,
    "msg": "second hard disk not found"
}
ok: [ansible2] => {
    "changed": false,
    "msg": "All assertions passed"
}
ok: [ansible3] => {
    "changed": false,
    "msg": "All assertions passed"
}
fatal: [ansible4]: FAILED! => {
    "assertion": "ansible_facts['devices']['sdb'] is defined",
    "changed": false,
    "evaluated_to": false,
    "msg": "second hard disk not found"
}
fatal: [ansible5]: FAILED! => {
    "assertion": "ansible_facts['devices']['sdb'] is defined",
    "changed": false,
    "evaluated_to": false,
    "msg": "second hard disk not found"
}

TASK [debug] *********************************************************
ok: [ansible2] => {
    "msg": "sdb was found, lets continue"
}
ok: [ansible3] => {
    "msg": "sdb was found, lets continue"
}

PLAY RECAP ***********************************************************
ansible1                   : ok=1    changed=0    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
ansible2                   : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
ansible3                   : ok=3    changed=0    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

```
ansible4                   : ok=1    changed=0    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
ansible5                   : ok=1    changed=0    unreachable=0
failed=1    skipped=0    rescued=0    ignored=0
```

Listing 15-4 Storing the Detected Disk Device Name in a Variable

```
---
- name: define variable according to diskname detected
  hosts: all
  tasks:
  - ignore_errors: yes
    set_fact:
      disk2name: sdb
    when: ansible_facts['devices']['sdb']
```

Listing 15-5 Running **ansible-playbook exercise151.yaml** Output

```
[ansible@control rhce8-book]$ ansible-playbook exercise151.yaml

PLAY [define variable according to diskname detected debug test] ****

TASK [Gathering Facts] ********************************************
ok: [ansible2]
ok: [ansible1]
ok: [ansible3]
ok: [ansible4]
ok: [ansible5]
ok: [ansible6]

TASK [set_fact] ***************************************************
fatal: [ansible1]: FAILED! => {"msg": "The conditional check 'ansible_
facts['devices']['sdb']' failed. The error was: error while evaluating
conditional (ansible_facts['devices']['sdb']): 'dict object' has no
attribute 'sdb'\n\nThe error appears to be in '/home/ansible/rhce8-
book/exercise151.yaml': line 5, column 5, but may\nbe elsewhere in
the file depending on the exact syntax problem.\n\nThe offending line
appears to be:\n\n  tasks:\n  - ignore_errors: yes\n    ^ here\n"}
...ignoring
```

```
ok: [ansible2]
ok: [ansible3]
fatal: [ansible4]: FAILED! => {"msg": "The conditional check 'ansible_
facts['devices']['sdb']' failed. The error was: error while evaluating
conditional (ansible_facts['devices']['sdb']): 'dict object' has no
attribute 'sdb'\n\nThe error appears to be in '/home/ansible/rhce8-
book/exercise151.yaml': line 5, column 5, but may\nbe elsewhere in
the file depending on the exact syntax problem.\n\nThe offending line
appears to be:\n\n  tasks:\n  - ignore_errors: yes\n    ^ here\n"}
...ignoring
fatal: [ansible5]: FAILED! => {"msg": "The conditional check 'ansible_
facts['devices']['sdb']' failed. The error was: error while evaluating
conditional (ansible_facts['devices']['sdb']): 'dict object' has no
attribute 'sdb'\n\nThe error appears to be in '/home/ansible/rhce8-
book/exercise151.yaml': line 5, column 5, but may\nbe elsewhere in
the file depending on the exact syntax problem.\n\nThe offending line
appears to be:\n\n  tasks:\n  - ignore_errors: yes\n    ^ here\n"}
...ignoring
fatal: [ansible6]: FAILED! => {"msg": "The conditional check 'ansible_
facts['devices']['sdb']' failed. The error was: error while evaluating
conditional (ansible_facts['devices']['sdb']): 'dict object' has no
attribute 'sdb'\n\nThe error appears to be in '/home/ansible/rhce8-
book/exercise151.yaml': line 5, column 5, but may\nbe elsewhere in
the file depending on the exact syntax problem.\n\nThe offending line
appears to be:\n\n  tasks:\n  - ignore_errors: yes\n    ^ here\n"}
...ignoring


TASK [set_fact] *****************************************************
fatal: [ansible1]: FAILED! => {"msg": "The conditional check
'ansible_facts['devices']['nvme0n2']' failed. The error was: error
while evaluating conditional (ansible_facts['devices']['nvme0n2']):
'dict object' has no attribute 'nvme0n2'\n\nThe error appears to
be in '/home/ansible/rhce8-book/exercise151.yaml': line 9, column
5, but may\nbe elsewhere in the file depending on the exact syntax
problem.\n\nThe offending line appears to be:\n\n    when: ansible_
facts['devices']['sdb']\n  - ignore_errors: yes\n    ^ here\n"}
...ignoring
fatal: [ansible2]: FAILED! => {"msg": "The conditional check
'ansible_facts['devices']['nvme0n2']' failed. The error was: error
while evaluating conditional (ansible_facts['devices']['nvme0n2']):
'dict object' has no attribute 'nvme0n2'\n\nThe error appears to
be in '/home/ansible/rhce8-book/exercise151.yaml': line 9, column
5, but may\nbe elsewhere in the file depending on the exact syntax
problem.\n\nThe offending line appears to be:\n\n    when: ansible_
facts['devices']['sdb']\n  - ignore_errors: yes\n    ^ here\n"}
...ignoring
fatal: [ansible3]: FAILED! => {"msg": "The conditional check
'ansible_facts['devices']['nvme0n2']' failed. The error was: error
while evaluating conditional (ansible_facts['devices']['nvme0n2']):
```

```
'dict object' has no attribute 'nvme0n2'\n\nThe error appears to
be in '/home/ansible/rhce8-book/exercise151.yaml': line 9, column
5, but may\nbe elsewhere in the file depending on the exact syntax
problem.\n\nThe offending line appears to be:\n\n    when: ansible_
facts['devices']['sdb']\n  - ignore_errors: yes\n    ^ here\n"}
...ignoring
fatal: [ansible4]: FAILED! => {"msg": "The conditional check
'ansible_facts['devices']['nvme0n2']' failed. The error was: error
while evaluating conditional (ansible_facts['devices']['nvme0n2']):
'dict object' has no attribute 'nvme0n2'\n\nThe error appears to
be in '/home/ansible/rhce8-book/exercise151.yaml': line 9, column
5, but may\nbe elsewhere in the file depending on the exact syntax
problem.\n\nThe offending line appears to be:\n\n    when: ansible_
facts['devices']['sdb']\n  - ignore_errors: yes\n    ^ here\n"}
...ignoring
fatal: [ansible5]: FAILED! => {"msg": "The conditional check
'ansible_facts['devices']['nvme0n2']' failed. The error was: error
while evaluating conditional (ansible_facts['devices']['nvme0n2']):
'dict object' has no attribute 'nvme0n2'\n\nThe error appears to
be in '/home/ansible/rhce8-book/exercise151.yaml': line 9, column
5, but may\nbe elsewhere in the file depending on the exact syntax
problem.\n\nThe offending line appears to be:\n\n    when: ansible_
facts['devices']['sdb']\n  - ignore_errors: yes\n    ^ here\n"}
...ignoring
ok: [ansible6]

TASK [getting out if there is no second disk] ***********************
fatal: [ansible1]: FAILED! => {"changed": false, "msg": "there is no
second disk"}
skipping: [ansible2]
skipping: [ansible3]
fatal: [ansible4]: FAILED! => {"changed": false, "msg": "there is no
second disk"}
fatal: [ansible5]: FAILED! => {"changed": false, "msg": "there is no
second disk"}
skipping: [ansible6]

TASK [showing this is only for hosts with a second disk] ************
ok: [ansible2] => {
    "msg": "continuing with sdb"
}
ok: [ansible3] => {
    "msg": "continuing with sdb"
}
ok: [ansible6] => {
    "msg": "continuing with nvme0n2"
}
```

```
PLAY RECAP **********************************************************
ansible1                    : ok=3    changed=0    unreachable=0
failed=1    skipped=0    rescued=0    ignored=2
ansible2                    : ok=4    changed=0    unreachable=0
failed=0    skipped=1    rescued=0    ignored=1
ansible3                    : ok=4    changed=0    unreachable=0
failed=0    skipped=1    rescued=0    ignored=1
ansible4                    : ok=3    changed=0    unreachable=0
failed=1    skipped=0    rescued=0    ignored=2
ansible5                    : ok=3    changed=0    unreachable=0
failed=1    skipped=0    rescued=0    ignored=2
ansible6                    : ok=4    changed=0    unreachable=0
failed=0    skipped=1    rescued=0    ignored=1
```

Listing 15-6 Creating Partitions with parted

```
---
- name: create storage
  hosts: ansible2
  tasks:
  - name: create new partition
    parted:
      name: files
      label: gpt
      device: /dev/sdb
      number: 1
      state: present
      part_start: 1MiB
      part_end: 2GiB
  - name: create another new partition
    parted:
      name: swap
      label: gpt
      device: /dev/sdb
      number: 2
      state: present
      part_start: 2GiB
      part_end: 4GiB
      flags: [ lvm ]
```

Listing 15-7 Creating an LVM Volume Group

```
  - name: create a volume group
    lvg:
      vg: vgdata
      pesize: "8"
      pvs: /dev/sdb1
```

Listing 15-8 Creating an LVM Logical Volume

```
  - name: create a logical volume
    lvol:
      lv: lvdata
      size: 100%FREE
      vg: vgdata
```

Listing 15-9 Creating an XFS File System

```
  - name: create an XFS filesystem
    filesystem:
      dev: /dev/vgdata/lvdata
      fstype: xfs
```

Listing 15-10 Using the mount Module to Mount a File System

```
  - name: mount the filesystem
    mount:
      src: /dev/vgdata/lvdata
      fstype: xfs
      state: mounted
      path: /mydir
```

Listing 15-11 Setting Up Swap Space

```
---
- name: configure swap storage
  hosts: ansible2
  tasks:
  - name: setup swap
    block:
    - name: make the swap filesystem
      filesystem:
        fstype: swap
        dev: /dev/sdb1
    - name: activate swap space
      command: swapon /dev/sdb1
    when: ansible_swaptotal_mb < 256
```

Listing 15-12 **exercise153-dev1.yaml** Failure Message

```
TASK [find small vgroups sizes] *************************************
*************
fatal: [ansible1]: FAILED! => {"msg": "The conditional check
'ansible_facts['lvm']['vgs']['cl']['size_g'] <= 20.00' failed. The
error was: Unexpected templating type error occurred on ({% if
ansible_facts['lvm']['vgs']['cl']['size_g'] <= 20.00 %} True {%
else %} False {% endif %}): '<=' not supported between instances of
'AnsibleUnsafeText' and 'float'\n\nThe error appears to be in '/home/
ansible/rhce8-book/exercise153-dev1.yaml': line 5, column 5, but
may\nbe elsewhere in the file depending on the exact syntax problem.
\n\nThe offending line appears to be:\n\n  tasks:\n  - name: find
small vgroups sizes\n    ^ here\n"}
fatal: [ansible2]: FAILED! => {"msg": "The conditional check
'ansible_facts['lvm']['vgs']['cl']['size_g'] <= 20.00' failed. The
error was: Unexpected templating type error occurred on ({% if
ansible_facts['lvm']['vgs']['cl']['size_g'] <= 20.00 %} True {%
else %} False {% endif %}): '<=' not supported between instances of
'AnsibleUnsafeText' and 'float'\n\nThe error appears to be in '/home/
ansible/rhce8-book/exercise153-dev1.yaml': line 5, column 5, but
may\nbe elsewhere in the file depending on the exact syntax problem.
\n\nThe offending line appears to be:\n\n  tasks:\n  - name: find
small vgroups sizes\n    ^ here\n"}
```

```
fatal: [ansible3]: FAILED! => {"msg": "The conditional check
'ansible_facts['lvm']['vgs']['cl']['size_g'] <= 20.00' failed. The
error was: Unexpected templating type error occurred on ({% if
ansible_facts['lvm']['vgs']['cl']['size_g'] <= 20.00 %} True {%
else %} False {% endif %}): '<=' not supported between instances of
'AnsibleUnsafeText' and 'float'\n\nThe error appears to be in '/home/
ansible/rhce8-book/exercise153-dev1.yaml': line 5, column 5, but
may\nbe elsewhere in the file depending on the exact syntax problem.
\n\nThe offending line appears to be:\n\n  tasks:\n  - name: find
small vgroups sizes\n    ^ here\n"}
fatal: [ansible4]: FAILED! => {"msg": "The conditional check
'ansible_facts['lvm']['vgs']['cl']['size_g'] <= 20.00' failed. The
error was: Unexpected templating type error occurred on ({% if
ansible_facts['lvm']['vgs']['cl']['size_g'] <= 20.00 %} True {%
else %} False {% endif %}): '<=' not supported between instances of
'AnsibleUnsafeText' and 'float'\n\nThe error appears to be in '/home/
ansible/rhce8-book/exercise153-dev1.yaml': line 5, column 5, but
may\nbe elsewhere in the file depending on the exact syntax problem.
\n\nThe offending line appears to be:\n\n  tasks:\n  - name: find
small vgroups sizes\n    ^ here\n"}
skipping: [ansible5]
skipping: [ansible6]

TASK [find large vgroups sizes] *************************************
*************
skipping: [ansible5]
skipping: [ansible6]
```

Listing 15-13 Error Message After Exercise 15-3 Step 10

```
TASK [get vg size and convert to integer in new variable]
*****************************
fatal: [ansible2]: FAILED! => {"msg": "The task includes an option
with an undefined variable. The error was: 'dict object' has no
attribute 'vgfiles'\n\nThe error appears to be in '/home/ansible/
rhce8-book/exercise153-step9.yaml': line 18, column 5, but may\nbe
elsewhere in the file depending on the exact syntax problem.\n\nThe
offending line appears to be:\n\n      vg: vgfiles\n  - name: get vg
size and convert to integer in new variable\n    ^ here\n"}
fatal: [ansible3]: FAILED! => {"msg": "The task includes an option
with an undefined variable. The error was: 'dict object' has no
attribute 'vgfiles'\n\nThe error appears to be in '/home/ansible/
rhce8-book/exercise153-step9.yaml': line 18, column 5, but may\nbe
elsewhere in the file depending on the exact syntax problem.\n\nThe
offending line appears to be:\n\n      vg: vgfiles\n  - name: get vg
size and convert to integer in new variable\n    ^ here\n"
```