

# Ansible: Getting Started in a Jupyter Notebook

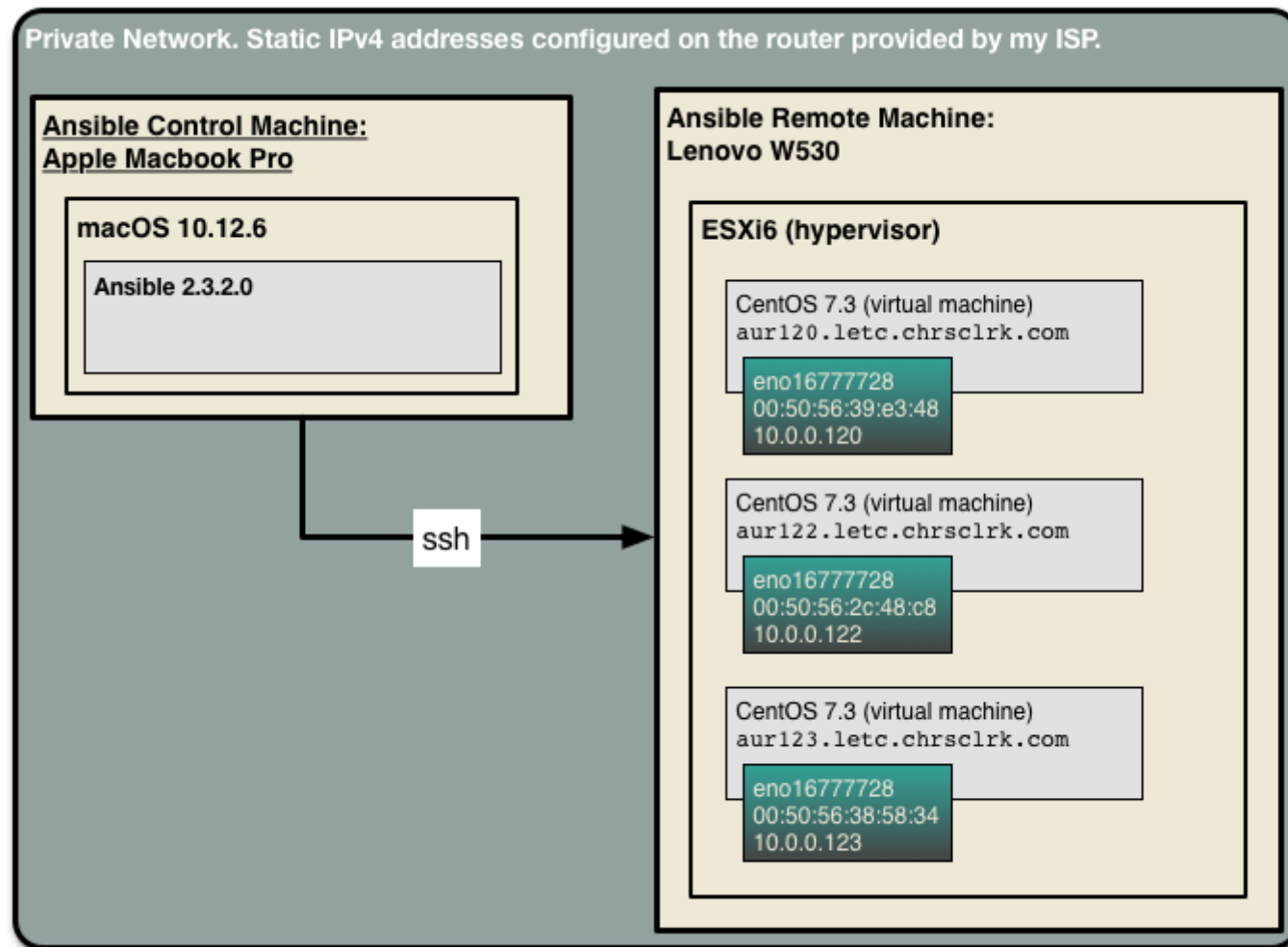
A Jupyter Notebook to follow along while reading "Ansible: Up and Running", 2nd Edition,

Lorin Hochstein, Rene Moser

<http://shop.oreilly.com/product/0636920065500.do> (<http://shop.oreilly.com/product/0636920065500.do>)

Publisher: O'Reilly Media, Inc. Release Date: August 2017 ISBN: 9781491979808

Remote machines of interest to the control machine:



# Ansible Control Machine: what is the execution context?

subprocess.check\_output() from page 546

"Python Cookbook" 3rd Edition, David Beazley and Brian K. Jones, 2013,

O'Reilly media, ISBN 978-1-4493-4037-7, <http://www.dabeaz.com/cookbook.html> (<http://www.dabeaz.com/cookbook.html>)

```
In [1]: import sys, platform, subprocess
ansibleVersion = subprocess.check_output(['ansible', '--version']).decode('utf-8').split()[1]
print(    f" Python:  {' '}.join(sys.version.split()[0:4])}\n"    # Not the version of Python used by Ansib
        f"  macOS:  {platform.mac_ver()[0]}\n"                  # Control machine operating system.
        f"Ansible:  {subprocess.check_output(['ansible', '--version']).decode('utf-8').split()[1]}")

Python:  3.6.0 |Anaconda custom (x86_64)|
macOS:   10.12.6
Ansible: 2.3.2.0
```

## View of the network, private, from /etc/hosts

(/etc/hosts display truncated to avoid line wrap)

```
In [2]: ! nl -b a /etc/hosts | sed -n '74,77p;84,86p' | cut -c 1-100

74 # 31Aug17R letc strata 13 aur "Ansible Up and Running" examples, hm17
75 #---+---10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-----+---80-----+---90---
76 # IPv4                hostname FQDN                                hstnm short   alias    comment
77 #-----
84      10.0.0.123          aur123.letc.chrsclrk.com  aur123        j123      # 31Aug17R ESXi MAC
85      10.0.0.122          aur122.letc.chrsclrk.com  aur122        j122      # 31Aug17R ESXi MAC
86      10.0.0.121          aur121.letc.chrsclrk.com  aur121        j121      # 31Aug17R ESXi MAC
```

## Review of control machine's Ansible configuration

Let the Jupyter notebook know where to find files.

```
In [3]: %cd '/Users/chrsclrk/Google Drive/solutionArchitect/automation'

/Users/chrsclrk/Google Drive/solutionArchitect/automation
```

## Note use of group "aur" to provide value for "become" password.

```
In [4]: ! echo "*** inventory.ini contents ***" ; nl -ba controlMachine/inventory.ini | sed -n '1,7p'
```

```
*** inventory.ini contents ***
 1  [aur]
 2  j120  ansible_host=aur120.letc.chrsclrk.com  ansible_port=22  ansible_user=virtuser
 3  j122  ansible_host=aur122.letc.chrsclrk.com  ansible_port=22  ansible_user=virtuser
 4  j123  ansible_host=aur123.letc.chrsclrk.com  ansible_port=22  ansible_user=virtuser
 5
 6  [aur:vars]
 7  # ansible_become_pass=<password vault is a better approach>
```

```
In [5]: ! echo "*** ansible.cfg contents ***" ; cat /Users/chrsclrk/.ansible.cfg
```

```
*** ansible.cfg contents ***
[defaults]
private_key_file=/Users/chrsclrk/.ssh/c6k
host_key_checking = False
# From http://ansible-docs.readthedocs.io/zh/stable-2.0/rst/intro\_configuration.html#ansible-managed
(http://ansible-docs.readthedocs.io/zh/stable-2.0/rst/intro\_configuration.html#ansible-managed)
ansible_managed = Ansible ma:wnaged: {file:} modified on %Y-%m-%d %H:%M:%S by {uid} on {host}
```

## Is the control machine able to reach the remote machines?

Ansible module, ping, replies with the string "ping" the control machine successfully connects with the host in the group aur.

```
In [6]: !ansible aur --inventory=controlMachine/inventory.ini --module-name=ping
```

```
j120 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
j122 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
j123 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}
```

**Here Ansible runs a program on the remote machines.**

- connectivity is established
- uptime for the remote machines may be of interest than the string "pong".

```
In [7]: !ansible aur --inventory=controlMachine/inventory.ini --module-name=command --args=uptime
```

```
j123 | SUCCESS | rc=0 >>  
15:01:49 up 11 days, 18:52,  1 user,  load average: 0.00, 0.01, 0.05  
  
j120 | SUCCESS | rc=0 >>  
15:01:49 up 11 days, 22:46,  1 user,  load average: 0.00, 0.01, 0.05  
  
j122 | SUCCESS | rc=0 >>  
15:01:49 up 6 days, 22:10,   1 user,  load average: 0.06, 0.03, 0.05
```

**Ansible's "setup" module; all it knows about a target machine**

When Ansible's connects to a target machine it collects information as part of its setup.  
The resulting data structure is collectively referred to as facts.

```
In [8]: oneSetup = !ansible aur[0] --inventory=controlMachine/inventory.ini --module-name=setup
print(f'{len(oneSetup):>34} Metric of setup results; length of Jupyter reference to saved setup results
      f'{type(oneSetup)} Type of Jupyter reference to save results.')
```

```
433 Metric of setup results; length of Jupyter reference to saved setup results.
<class 'IPython.utils.text.SList'> Type of Jupyter reference to save results.
```

```
In [9]: oneSetup # View facts from the first target machine.
```

```
'      "tz": "GMT", ',
'      "tz_offset": "+0000", ',
'      "weekday": "Tuesday", ',
'      "weekday_number": "2", ',
'      "weeknumber": "37", ',
'      "year": "2017"',
'  }, ',
'  "ansible_default_ipv4": {',
'    "address": "10.0.0.120", ',
'    "alias": "eno16777728", ',
'    "broadcast": "10.0.0.255", ',
'    "gateway": "10.0.0.1", ',
'    "interface": "eno16777728", ',
'    "macaddress": "00:50:56:39:e3:48", ',
'    "mtu": 1500, ',
'    "netmask": "255.255.255.0", ',
'    "network": "10.0.0.0", ',
'    "type": "ether"',
'  }, ',
'  "ansible_default_ipv6": {',
```

## Retrieve a subset of the facts.

From Loren Hochsetein's page

<https://github.com/lorin/ansible-quickref/blob/master/facts.rst> (<https://github.com/lorin/ansible-quickref/blob/master/facts.rst>)

```
In [10]: !ansible aur --inventory=controlMachine/inventory.ini --module-name=setup --args='filter=ansible_default
```

```
j122 | SUCCESS => {
  "ansible_facts": {
    "ansible_default_ipv4": {
      "address": "10.0.0.122",
      "alias": "eno16777728",
      "broadcast": "10.0.0.255",
      "gateway": "10.0.0.1",
      "interface": "eno16777728",
      "macaddress": "00:50:56:2c:48:c8",
      "mtu": 1500,
      "netmask": "255.255.255.0",
      "network": "10.0.0.0",
      "type": "ether"
    }
  },
  "changed": false
}
j123 | SUCCESS => {
  "ansible_facts": {
    "ansible_default_ipv4": {
      "address": "10.0.0.123",
      "alias": "eno16777728",
      "broadcast": "10.0.0.255",
      "gateway": "10.0.0.1",
      "interface": "eno16777728",
      "macaddress": "00:50:56:38:58:34",
      "mtu": 1500,
      "netmask": "255.255.255.0",
      "network": "10.0.0.0",
      "type": "ether"
    }
  },
  "changed": false
}
j120 | SUCCESS => {
  "ansible_facts": {
    "ansible_default_ipv4": {
      "address": "10.0.0.120",
      "alias": "eno16777728",
      "broadcast": "10.0.0.255",
      "gateway": "10.0.0.1",
```

```

        "interface": "eno16777728",
        "macaddress": "00:50:56:39:e3:48",
        "mtu": 1500,
        "netmask": "255.255.255.0",
        "network": "10.0.0.0",
        "type": "ether"
    },
    "changed": false
}

```

## Report the target machines' date for rough idea of time synchrony.

From the control machine, Ansible concurrently accesses the target machines.

This example shows the three target machines are within a 100 milliseconds of each other.

```
In [11]: !ansible aur --inventory=controlMachine/inventory.ini --module-name=shell --args="date --rfc-3339=ns"
```

```

j123 | SUCCESS | rc=0 >>
2017-09-12 15:02:08.814089544+00:00

```

```

j122 | SUCCESS | rc=0 >>
2017-09-12 15:02:08.813610102+00:00

```

```

j120 | SUCCESS | rc=0 >>
2017-09-12 15:02:08.826516010+00:00

```

## Results from piping commands on the remote machines.

Three commands on the target host to yield an IPv4 address of a network device.

(CentOS uses the “Predictable Network Interface Names

(<https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/>)” convention.)

```
In [12]: !ansible aur --inventory=controlMachine/inventory.ini --module-name=shell --args="/usr/sbin/ip -4 add
j123 | SUCCESS | rc=0 >>
10.0.0.123/24

j120 | SUCCESS | rc=0 >>
10.0.0.120/24

j122 | SUCCESS | rc=0 >>
10.0.0.122/24
```

## Ansible Playbooks

One playbook with one task and three debug statments concerning the network adapater Ansible is using by default." Ansible's debug module is used to

- print the IPv4 address
- print the MAC address
- print the IPv4 and MAC address togehter on one line

```
In [13]: ! nl playbooks/ch04-98_facts_ip-mac.yaml

1  - name: For the default IPv4 adpater, show IP separate from MAC, then IPv4 and MAC on the same
line.
2      hosts: aur
3      gather_facts: True
4      tasks:
5          - debug: var=ansible_default_ipv4.address
6          - debug: var=ansible_default_ipv4.macaddress
7          - debug: msg=" ip:mac {{ ansible_default_ipv4.address }} ':' {{ ansible_default_ipv4.macad
dress }}"
```



```
In [14]: ! ansible-playbook --verbose --inventory=controlMachine/inventory.ini --become playbooks/ch04-98_facts_i
```

```
Using /Users/chrsclrk/.ansible.cfg as config file
```

```
PLAY [For the default IPv4 adpater, show IP separate from MAC, then IPv4 and MAC on the same line.] **
*
```

```
TASK [Gathering Facts] *****
```

```
ok: [j120]
```

```
ok: [j123]
```

```
ok: [j122]
```

```
TASK [debug] *****
```

```
ok: [j120] => {
```

```
  "ansible_default_ipv4.address": "10.0.0.120"
```

```
}
```

```
ok: [j122] => {
```

```
  "ansible_default_ipv4.address": "10.0.0.122"
```

```
}
```

```
ok: [j123] => {
```

```
  "ansible_default_ipv4.address": "10.0.0.123"
```

```
}
```

```
TASK [debug] *****
```

```
ok: [j120] => {
```

```
  "ansible_default_ipv4.macaddress": "00:50:56:39:e3:48"
```

```
}
```

```
ok: [j122] => {
```

```
  "ansible_default_ipv4.macaddress": "00:50:56:2c:48:c8"
```

```
}
```

```
ok: [j123] => {
```

```
  "ansible_default_ipv4.macaddress": "00:50:56:38:58:34"
```

```
}
```

```
TASK [debug] *****
```

```
ok: [j120] => {
```

```
  "msg": " ip:mac 10.0.0.120 ':' 00:50:56:39:e3:48"
```

```
}
```

```
ok: [j122] => {
```

```
  "msg": " ip:mac 10.0.0.122 ':' 00:50:56:2c:48:c8"
```

```
}
```

```
ok: [j123] => {
```

```
  "msg": " ip:mac 10.0.0.123 ':' 00:50:56:38:58:34"
```

```
}
```

```
PLAY RECAP *****
```

```
j120      : ok=4    changed=0    unreachable=0    failed=0
```

```
j122      : ok=4    changed=0    unreachable=0    failed=0
```

```
j123      : ok=4    changed=0    unreachable=0    failed=0
```

In [ ]: