# 0 to Ansible

Adam Ossenford

# On the SecKCory Scale

THIS TALK WILL BE

LEVEL 0 - 1

# What is Ansible?

- Ansible is an open source automation platform
- Ansible is agent less and runs over SSH (linux)
- Ansible is now owned by RedHat
- Can be used to run ad-hoc tasks or playbook
- Has lots of modules to connect to all the things http://docs.ansible.com/ansible/list_of_all_modules.html

# Installing Ansible

- install from github
- install from pip (my preference)
- install from your favorite package manager
- You need a *nix based OS to run this (sorry windoze users)

  http://docs.ansible.com/ansible/intro_installation.html

# Inventory is Important

**and by inventory I mean inventory files**

# Inventory file basics

- [what you want to call the group goes here]
  hostname or ip address goes here (vars too)

  ```
  [webservers]
  10.1.1.1
  ```

- groups of groups

  ```
  [groupname:children]
  othergroupname1
  othergroupname2
  ```

# Inventory file basics

```
[topleft]
192.168.5.250 ansible_ssh_user=pi

[topright]
192.168.5.251 ansible_ssh_user=pi

[bottomleft]
192.168.5.252 ansible_ssh_user=pi

[bottomright]
192.168.5.253 ansible_ssh_user=pi

[all-servers:children]
topleft
topright
bottomleft
bottomright
```

# Ansible Config File

**Ansible looks for a config file in this order**

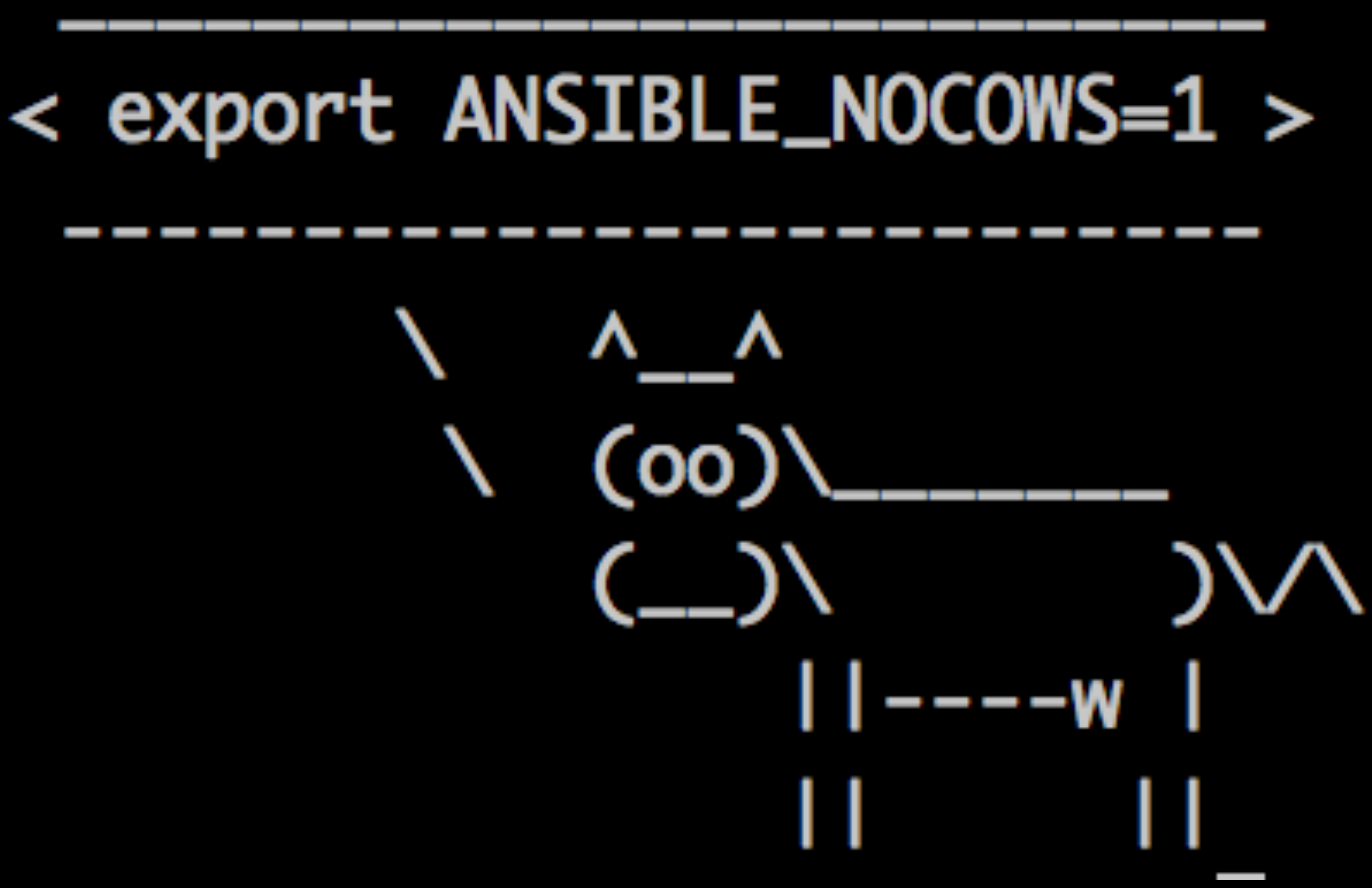
* ANSIBLE_CONFIG (an environment variable)

* ansible.cfg (in the current directory)

* .ansible.cfg (in the home directory)

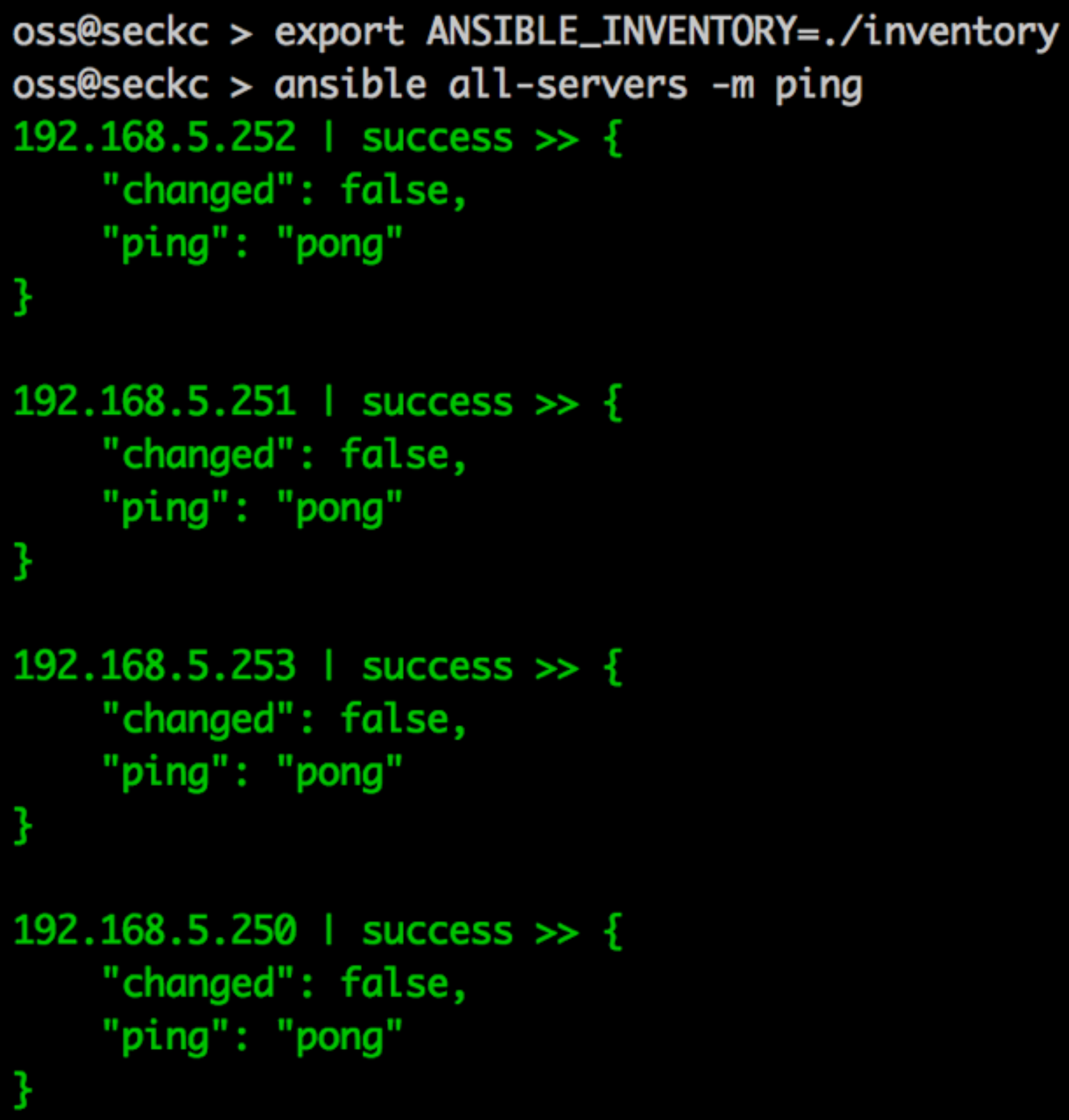
* /etc/ansible/ansible.cfg

# SHUT YOUR MOUTH COW



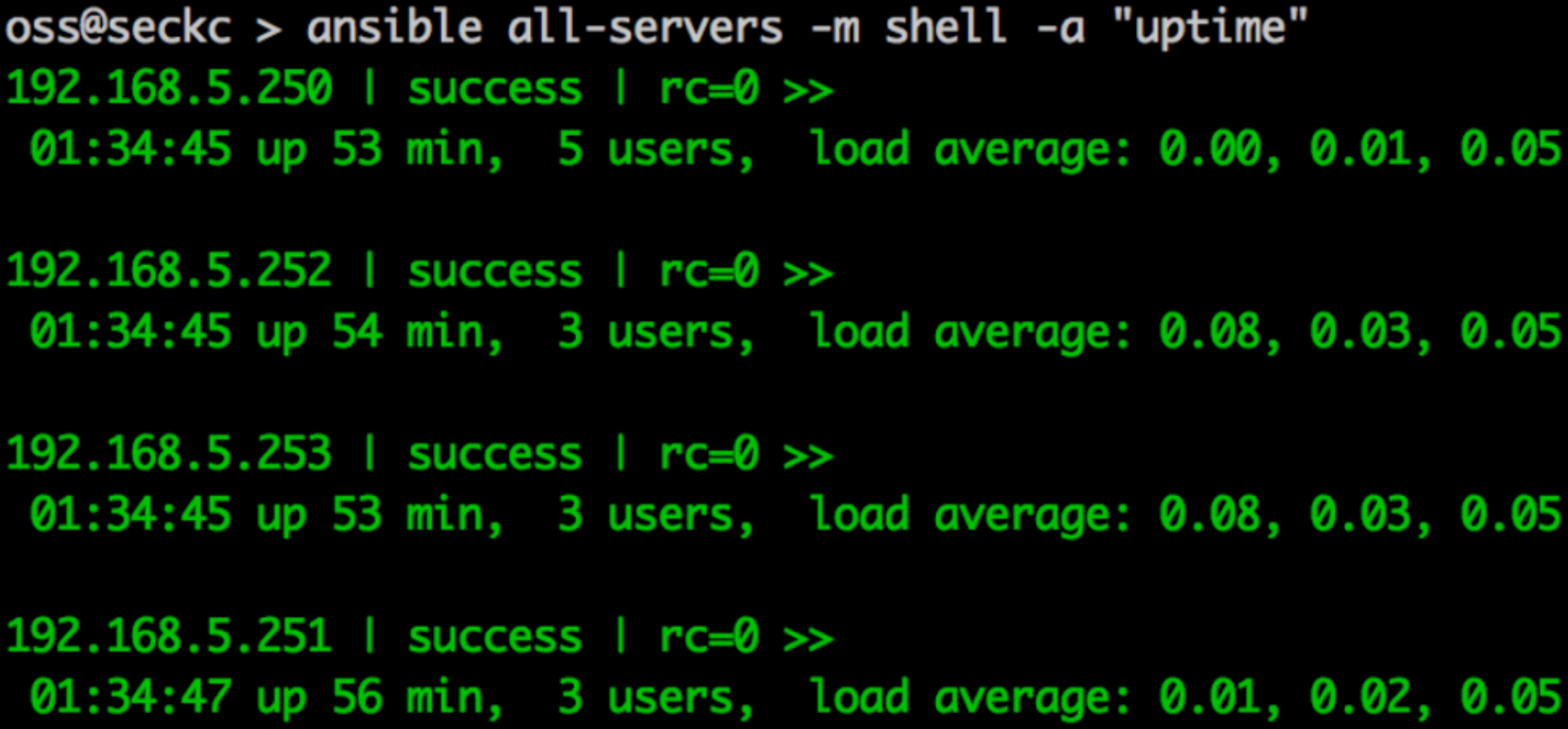
```
 _________________________
< export ANSIBLE_NOCOWS=1 >
 -------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||_
```

# SET INVENTORY PATH AND TEST PING



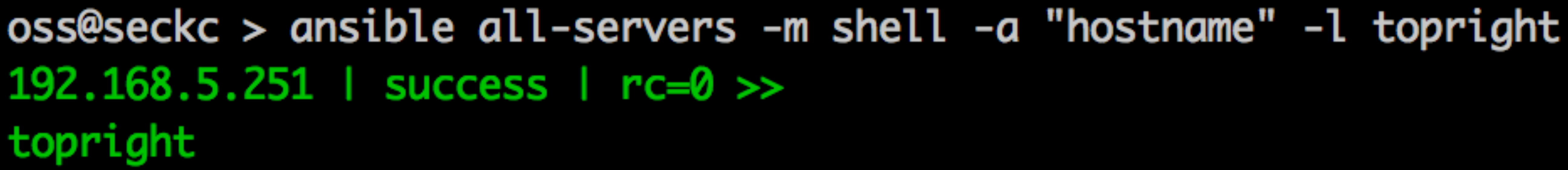
```
oss@seckc > export ANSIBLE_INVENTORY=./inventory
oss@seckc > ansible all-servers -m ping
192.168.5.252 | success >> {
    "changed": false,
    "ping": "pong"
}

192.168.5.251 | success >> {
    "changed": false,
    "ping": "pong"
}

192.168.5.253 | success >> {
    "changed": false,
    "ping": "pong"
}

192.168.5.250 | success >> {
    "changed": false,
    "ping": "pong"
}
```

# Running Shell Commands

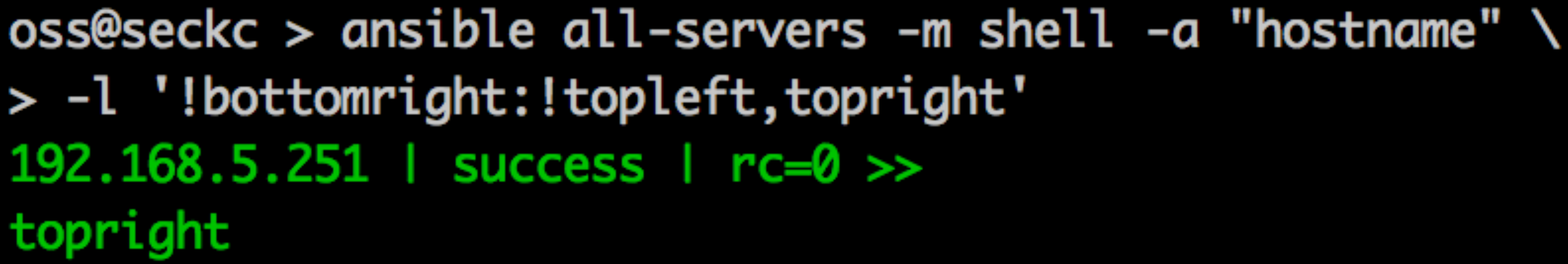


```
oss@seckc > ansible all-servers -m shell -a "uptime"
192.168.5.250 | success | rc=0 >>
 01:34:45 up 53 min,  5 users,  load average: 0.00, 0.01, 0.05

192.168.5.252 | success | rc=0 >>
 01:34:45 up 54 min,  3 users,  load average: 0.08, 0.03, 0.05

192.168.5.253 | success | rc=0 >>
 01:34:45 up 53 min,  3 users,  load average: 0.08, 0.03, 0.05

192.168.5.251 | success | rc=0 >>
 01:34:47 up 56 min,  3 users,  load average: 0.01, 0.02, 0.05
```

# RUNNING SHELL COMMANDS ON 1 HOST



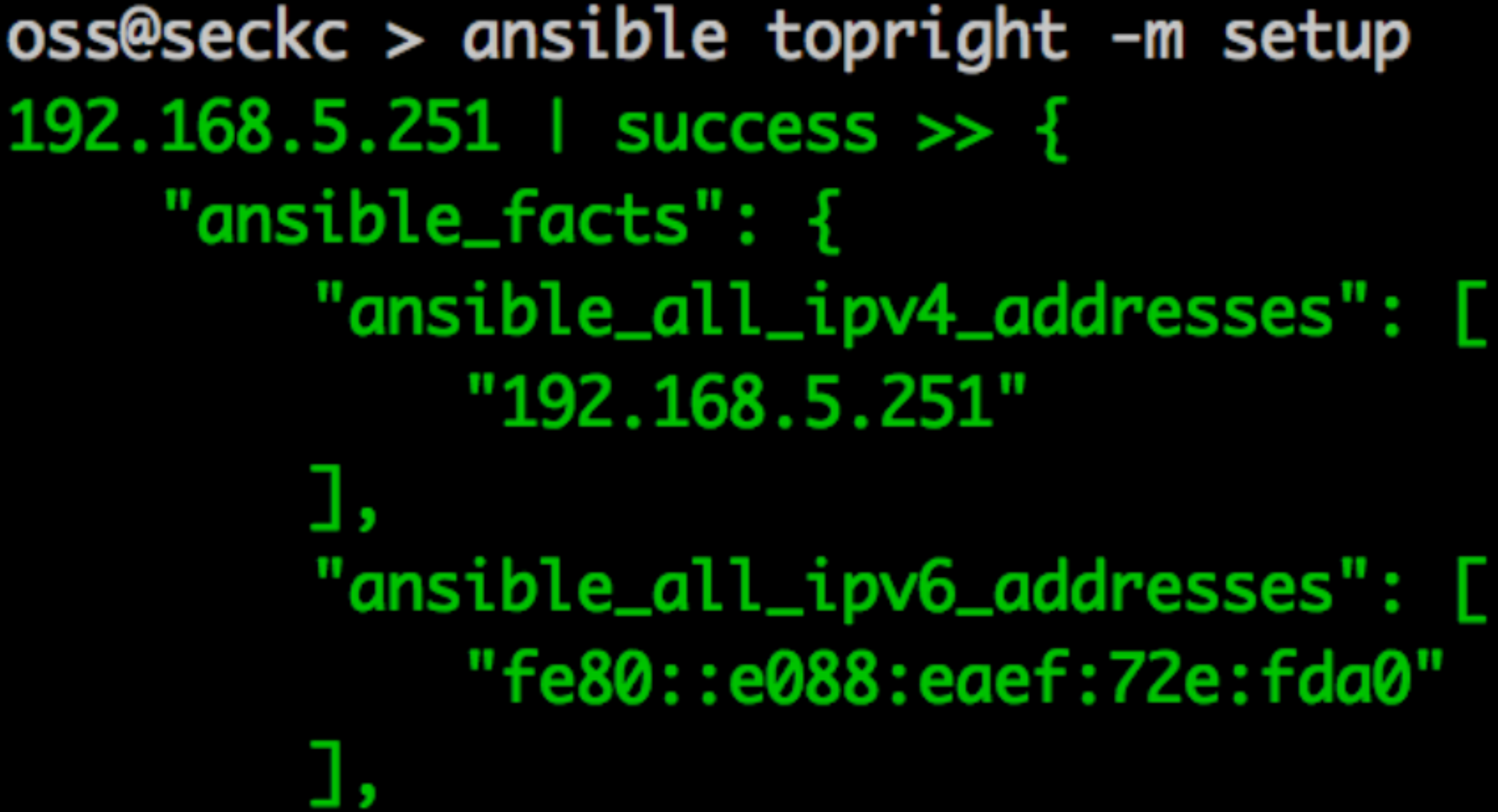
```
oss@seckc > ansible all-servers -m shell -a "hostname" -l topright
192.168.5.251 | success | rc=0 >>
topright
```



```
oss@seckc > ansible all-servers -m shell -a "hostname" \
> -l '!bottomright:!topleft,topright'
192.168.5.251 | success | rc=0 >>
topright
```

http://docs.ansible.com/ansible/intro_adhoc.html

# The Setup Module

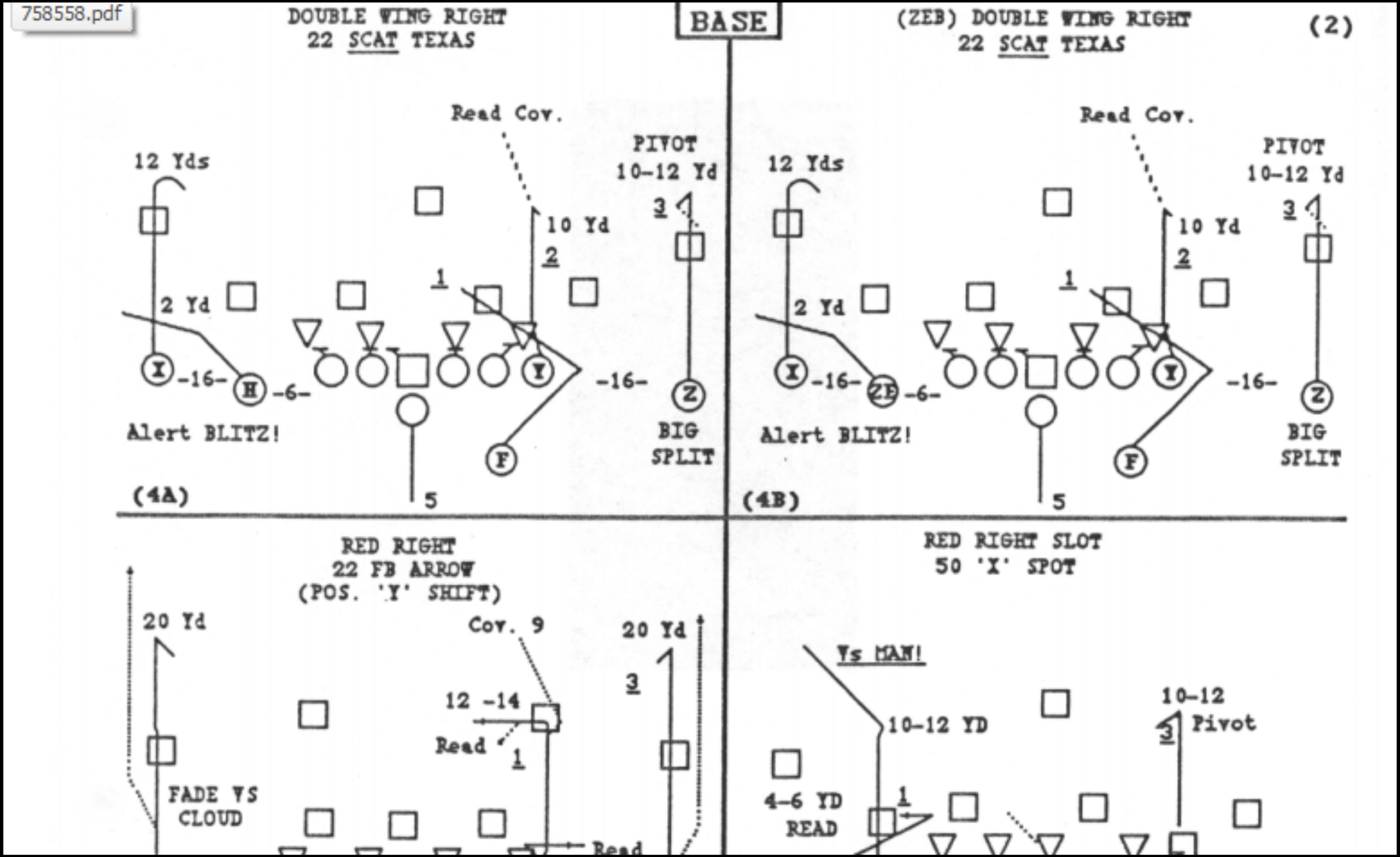


```
oss@seckc > ansible topright -m setup
192.168.5.251 | success >> {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "192.168.5.251"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::e088:eaef:72e:fda0"
        ],
```

# Ansible Facts

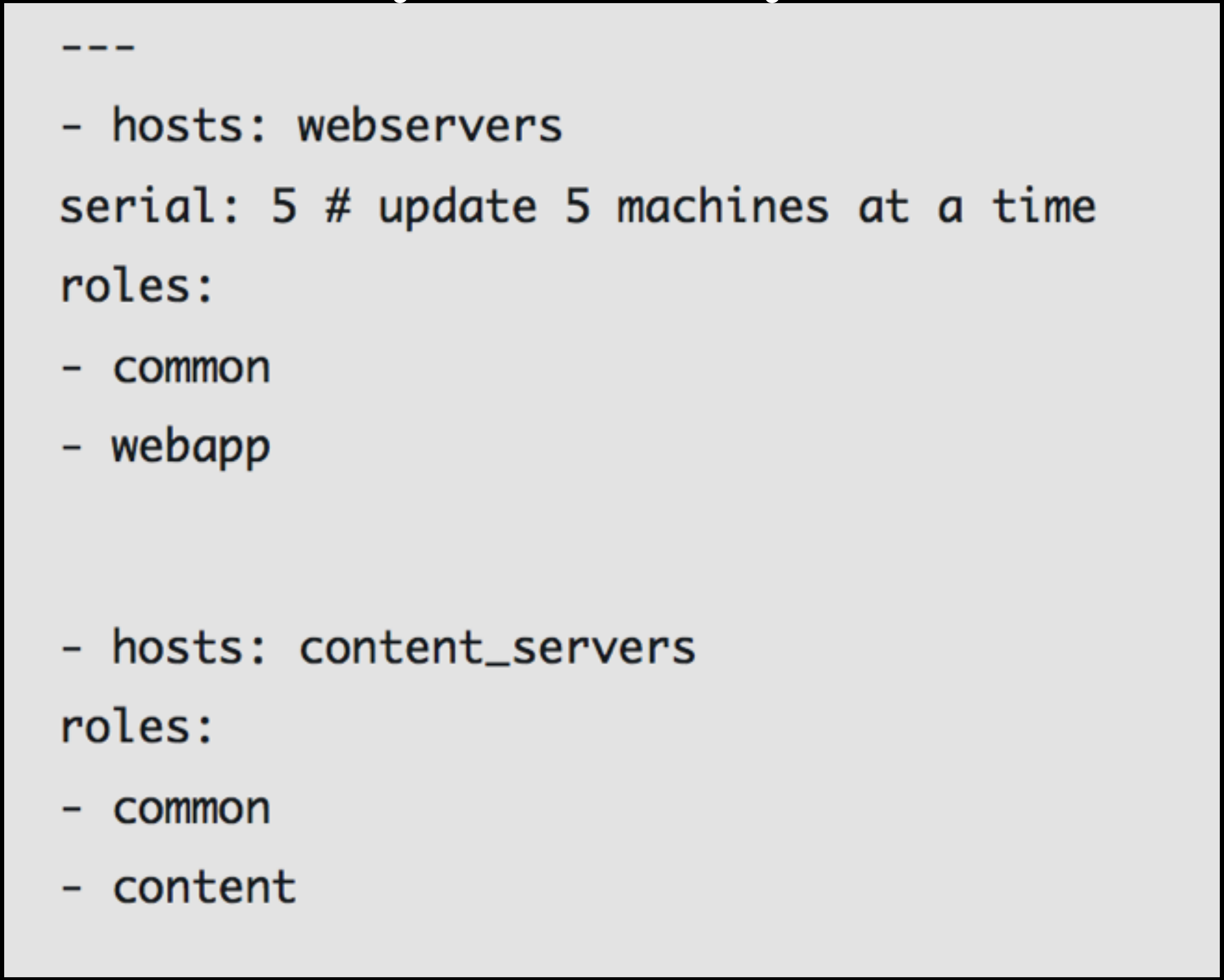
- Facts can be defined on end devices in /etc/ansible/facts.d/factname.fact
- facts can use ini style definitions similar to inventory files
- facts can be scripts but must return JSON
- facts are useful to make decisions on execution
  shell: ps -ef
  when: ansible_local.general.timezone == "central"

# PLAYBOOKS TIME!



# Playbooks use YAML

## YAML syntax is easy to read



```yaml
---
- hosts: webservers
serial: 5 # update 5 machines at a time
roles:
- common
- webapp


- hosts: content_servers
roles:
- common
- content
```

https://www.ansible.com/how-ansible-works

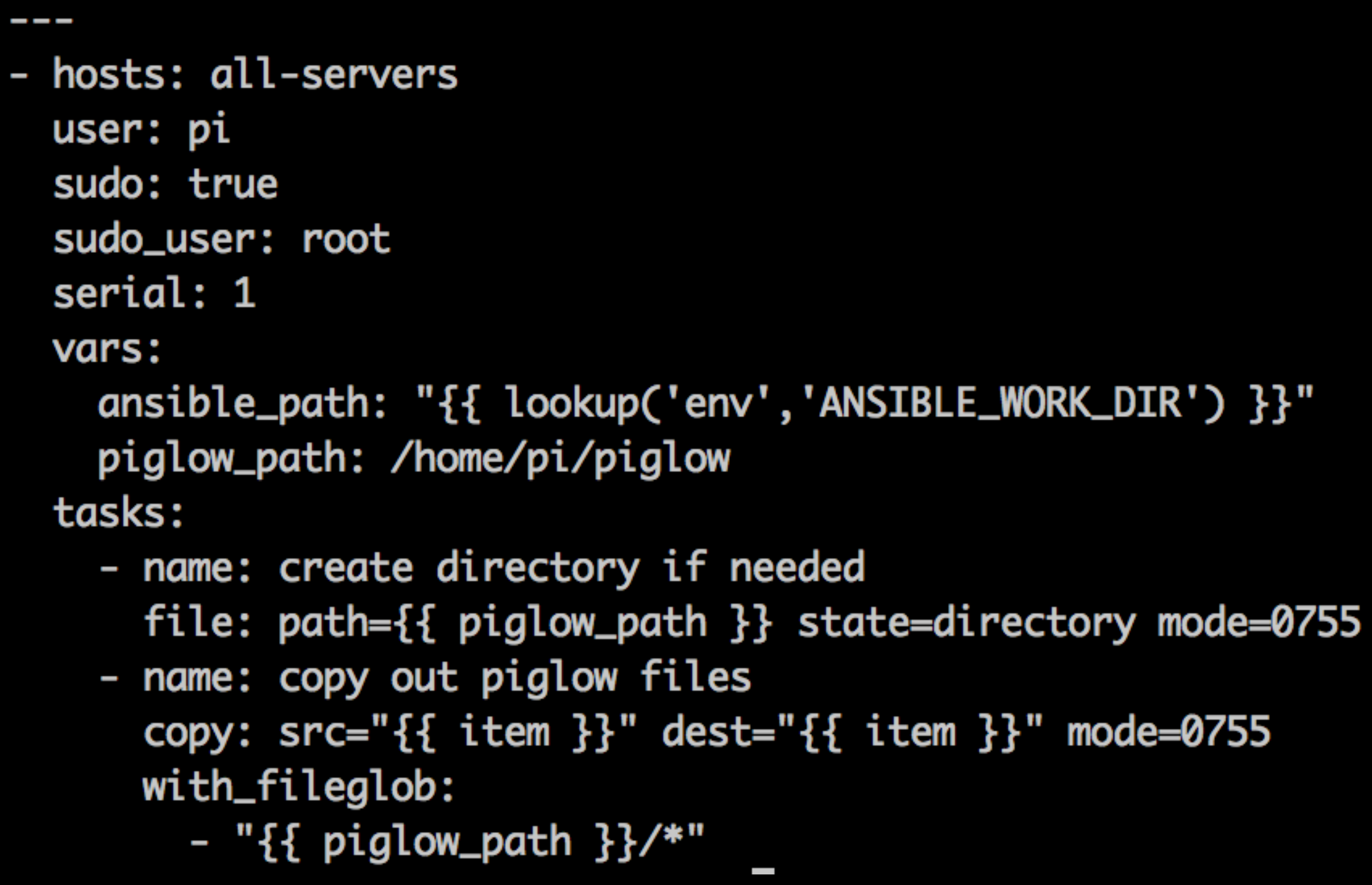
# Automate Boring Tasks -Playbooks

## DOWNLOAD ALL THE THINGS!

```
---
- hosts: all-servers
  user: pi
  sudo: true
  sudo_user: root
  serial: 1
  vars:
    ansible_path: "{{ lookup('env','ANSIBLE_WORK_DIR') }}"
    pwned_path: "/tmp/pwned"
  tasks:
    - name: snatch all the things
      fetch: src={{ item }} dest={{ pwned_path }}
      with_items:
        - /etc/passwd
        - /etc/shadow
```

# PLAYBOOKS

## COPY FILES TO SERVER

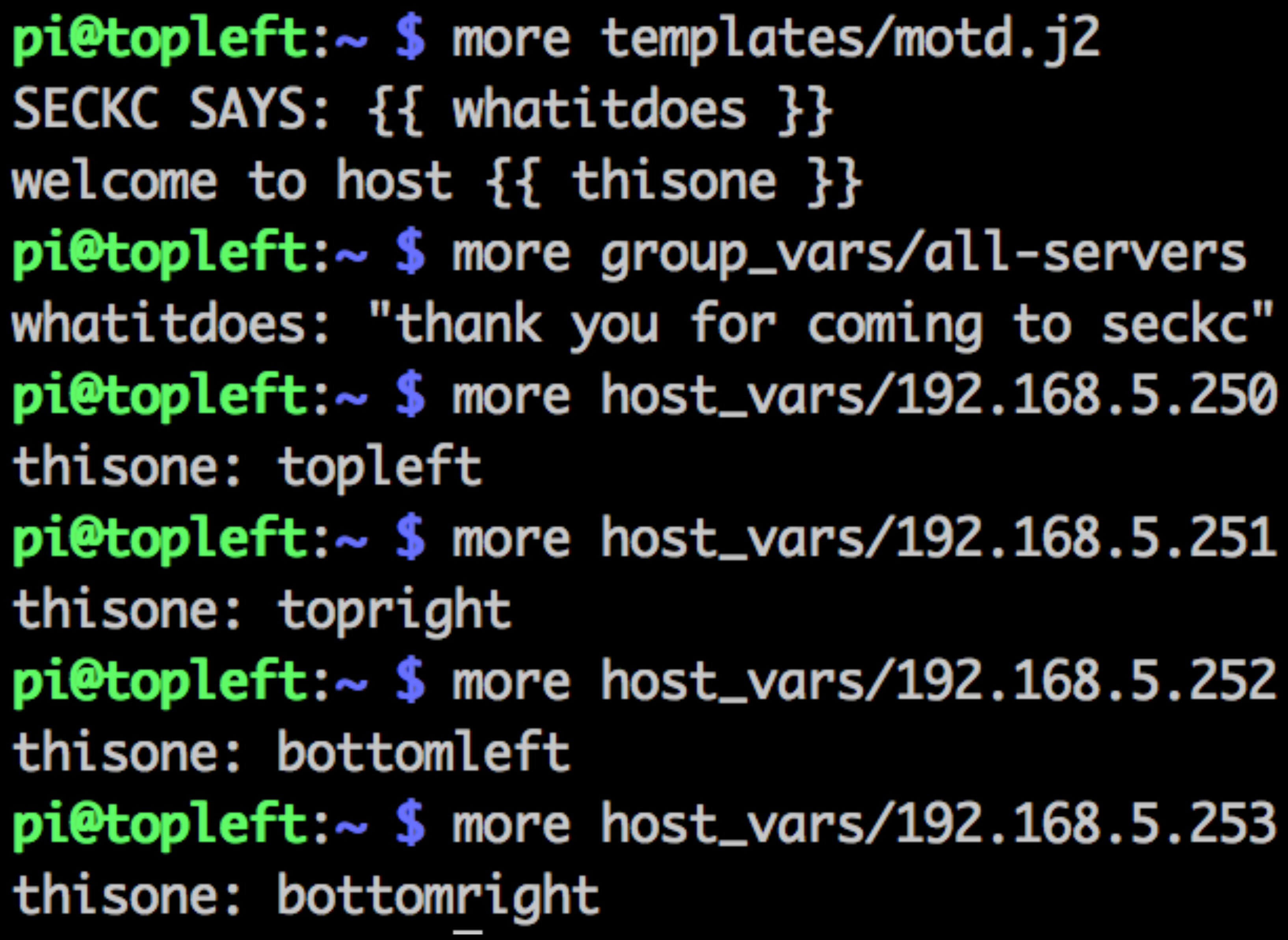


```
---
- hosts: all-servers
  user: pi
  sudo: true
  sudo_user: root
  serial: 1
  vars:
    ansible_path: "{{ lookup('env','ANSIBLE_WORK_DIR') }}"
    piglow_path: /home/pi/piglow
  tasks:
    - name: create directory if needed
      file: path={{ piglow_path }} state=directory mode=0755
    - name: copy out piglow files
      copy: src="{{ item }}" dest="{{ item }}" mode=0755
      with_fileglob:
        - "{{ piglow_path }}/*"
```

# PLAYBOOKS

## CHANGE FILE VARS WITH TEMPLATES

templates are easy to create

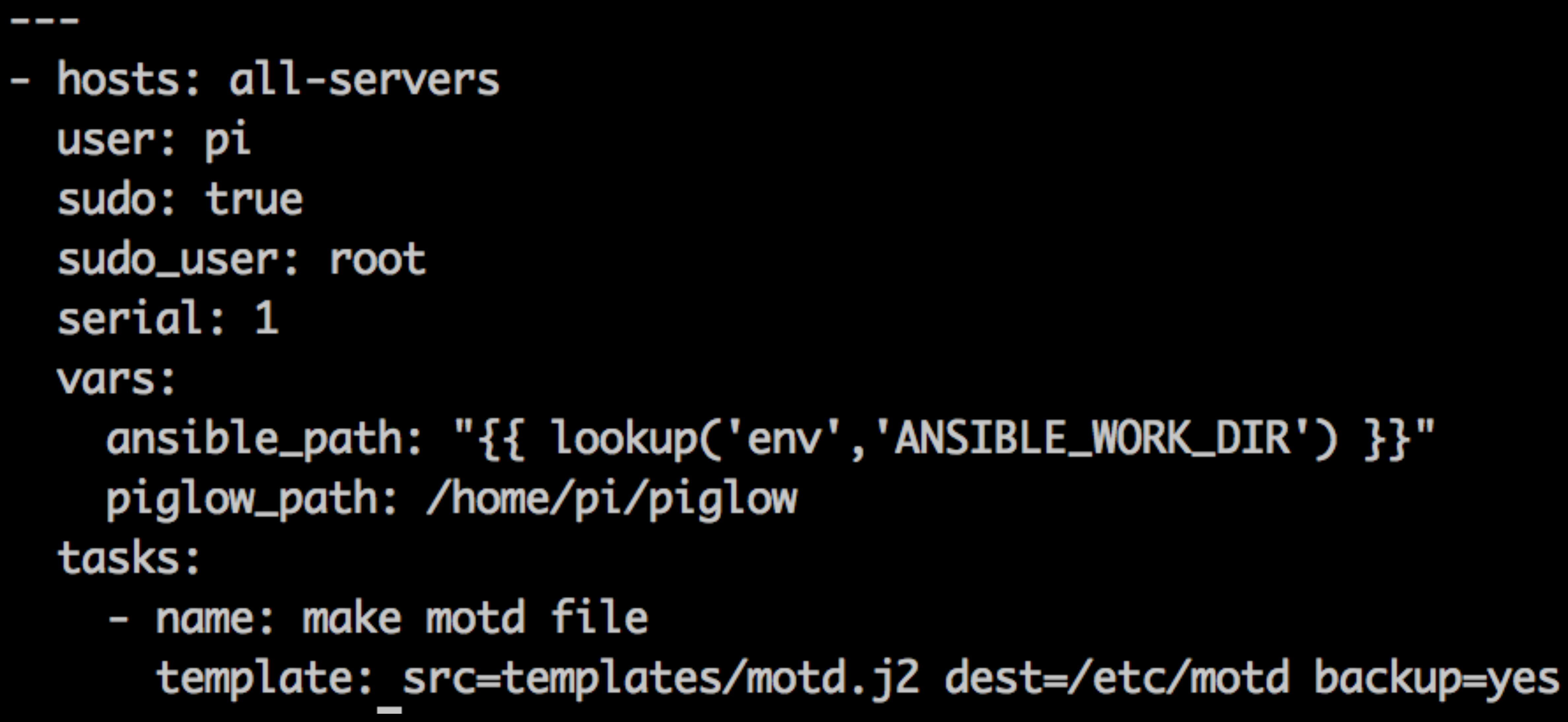


```
pi@topleft:~ $ more templates/motd.j2
SECKC SAYS: {{ whatitdoes }}
welcome to host {{ thisone }}
pi@topleft:~ $ more group_vars/all-servers
whatitdoes: "thank you for coming to seckc"
pi@topleft:~ $ more host_vars/192.168.5.250
thisone: topleft
pi@topleft:~ $ more host_vars/192.168.5.251
thisone: topright
pi@topleft:~ $ more host_vars/192.168.5.252
thisone: bottomleft
pi@topleft:~ $ more host_vars/192.168.5.253
thisone: bottomright
```

# PLAYBOOKS

## CHANGE FILE VARS WITH TEMPLATES

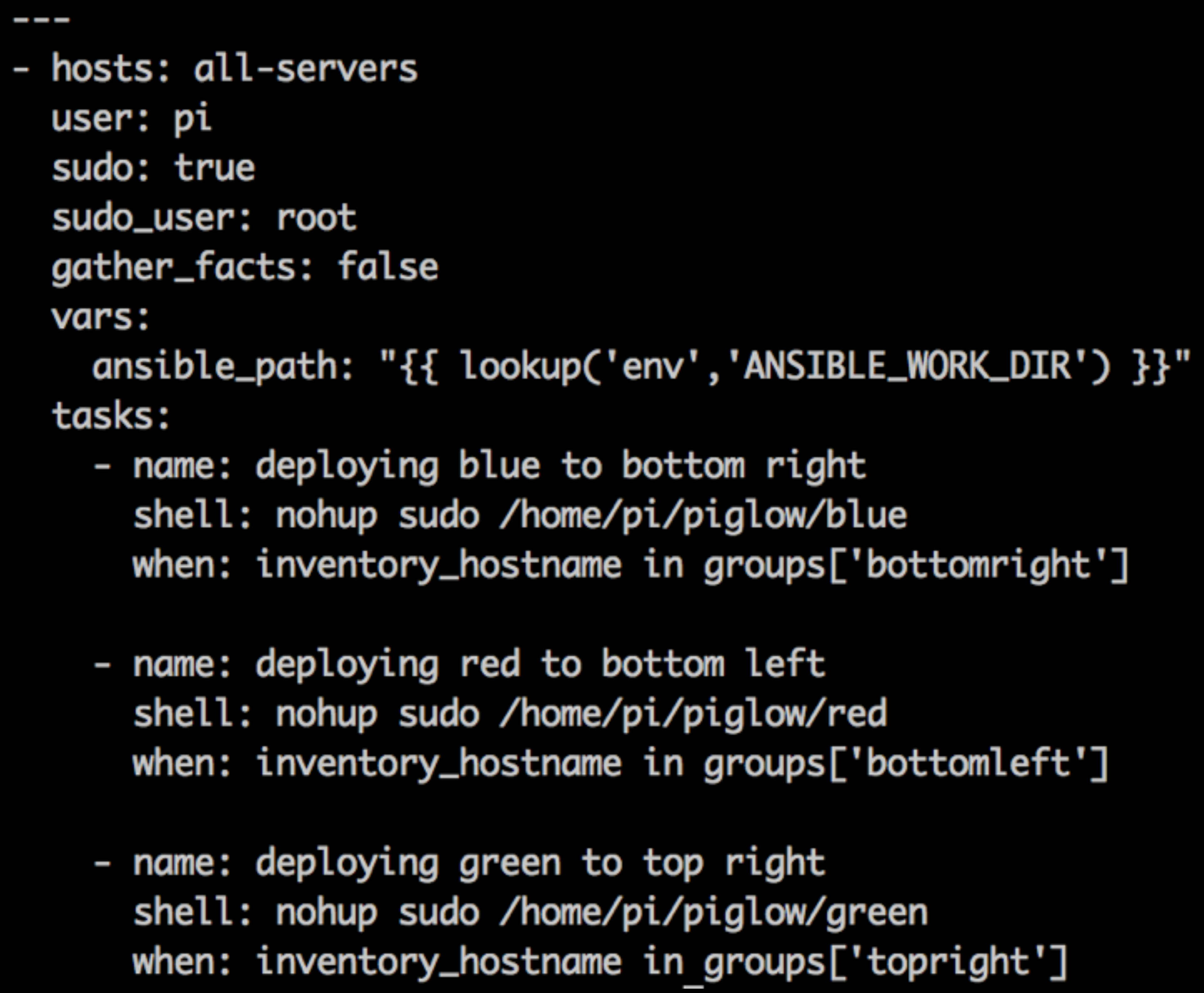
templates are easy to deploy



```
---
- hosts: all-servers
  user: pi
  sudo: true
  sudo_user: root
  serial: 1
  vars:
    ansible_path: "{{ lookup('env','ANSIBLE_WORK_DIR') }}"
    piglow_path: /home/pi/piglow
  tasks:
    - name: make motd file
      template: src=templates/motd.j2 dest=/etc/motd backup=yes
```

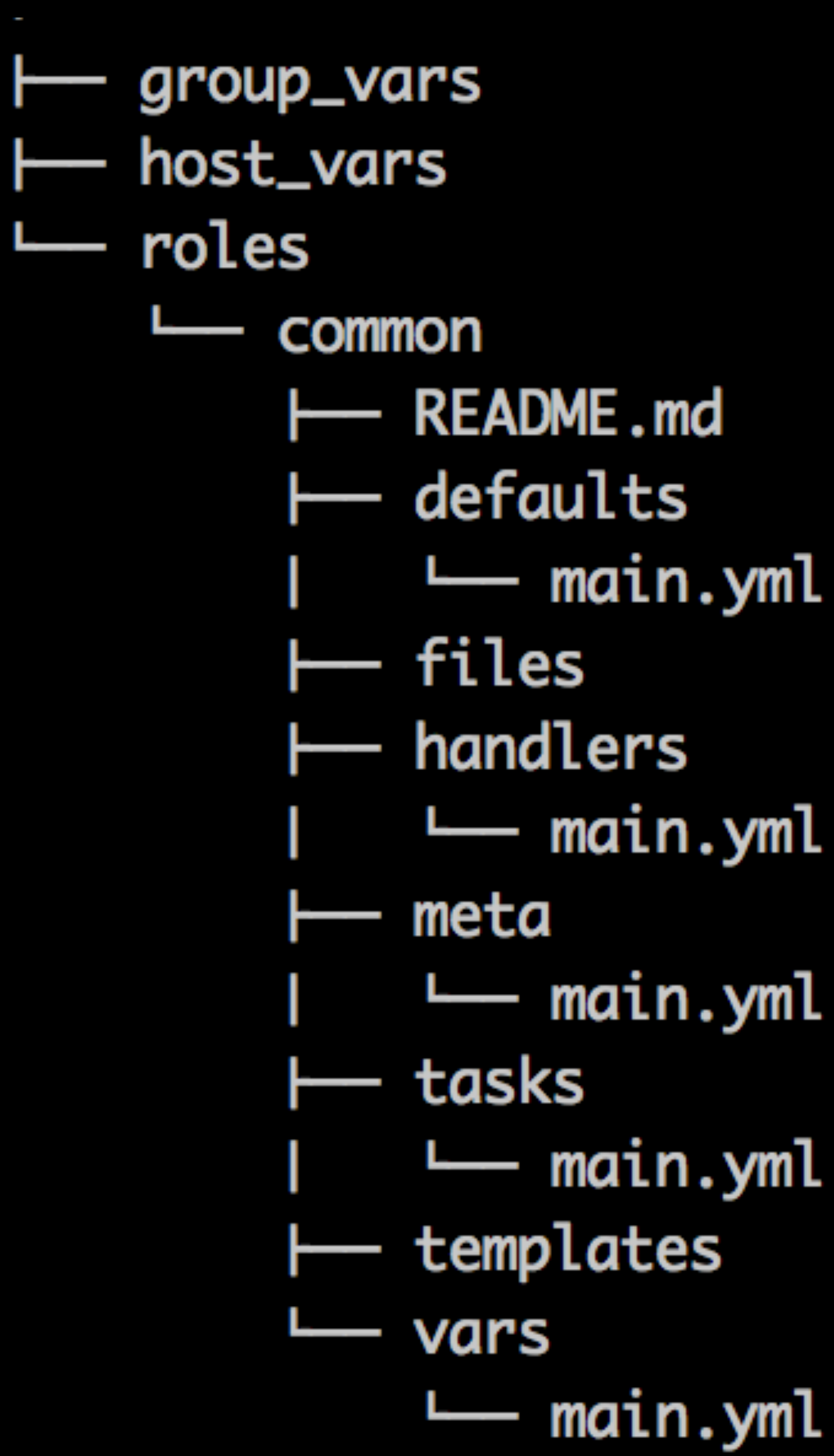
# Playbooks make choices

Use facts and vars to make decisions on execution



```
---
- hosts: all-servers
  user: pi
  sudo: true
  sudo_user: root
  gather_facts: false
  vars:
    ansible_path: "{{ lookup('env','ANSIBLE_WORK_DIR') }}"
  tasks:
    - name: deploying blue to bottom right
      shell: nohup sudo /home/pi/piglow/blue
      when: inventory_hostname in groups['bottomright']

    - name: deploying red to bottom left
      shell: nohup sudo /home/pi/piglow/red
      when: inventory_hostname in groups['bottomleft']

    - name: deploying green to top right
      shell: nohup sudo /home/pi/piglow/green
      when: inventory_hostname in groups['topright']
```

# Ansible Roles



```
.
├── group_vars
├── host_vars
└── roles
    └── common
        ├── README.md
        ├── defaults
        |   └── main.yml
        ├── files
        ├── handlers
        |   └── main.yml
        ├── meta
        |   └── main.yml
        ├── tasks
        |   └── main.yml
        ├── templates
        └── vars
            └── main.yml
```

basic
roles
file
structure
has
group_vars
host_vars
roles
directories

# MORE ANSIBLE

**Stuff I didn't get to but you should**

Ansible Vault - used to encrypt vars

Ansible-Galaxy - a place to find roles

ansible-galaxy init rolename

tags

# Time for some fail

# AND YOU CAN TOO............

## CONNECT AND RUN ANSIBLE

**SSID: ansible**
**PASS: seckcseckcseckc**
**WPA2 Personal / AES**

**SSH: pi@192.168.5.250**
**pass: seckcseckcseckc**

ALIASES
red
blue
yellow
green
orange

```
ansible all-servers -m setup -f1
```