ANSIBLE, ELIXIR AND YOU

Devops That Doesn't Suck

J. Scott Johnson / fuzzygroup@gmail.com / fuzzygroup.github.io/blog

WHO AM I?

- Software Engineer and Consultant
- Blogger
- Focus is Ruby; Moving to Elixir
- Specialize in back end work, heavy focus on database and search
- Always available for work

WHAT IS ANSIBLE?

- Declarative machine provisioning tool
- Open source; owned by Redhat
- Written in python
- Runs purely via SSH

ANTI CONCEPT - SNOWFLAKE SERVER

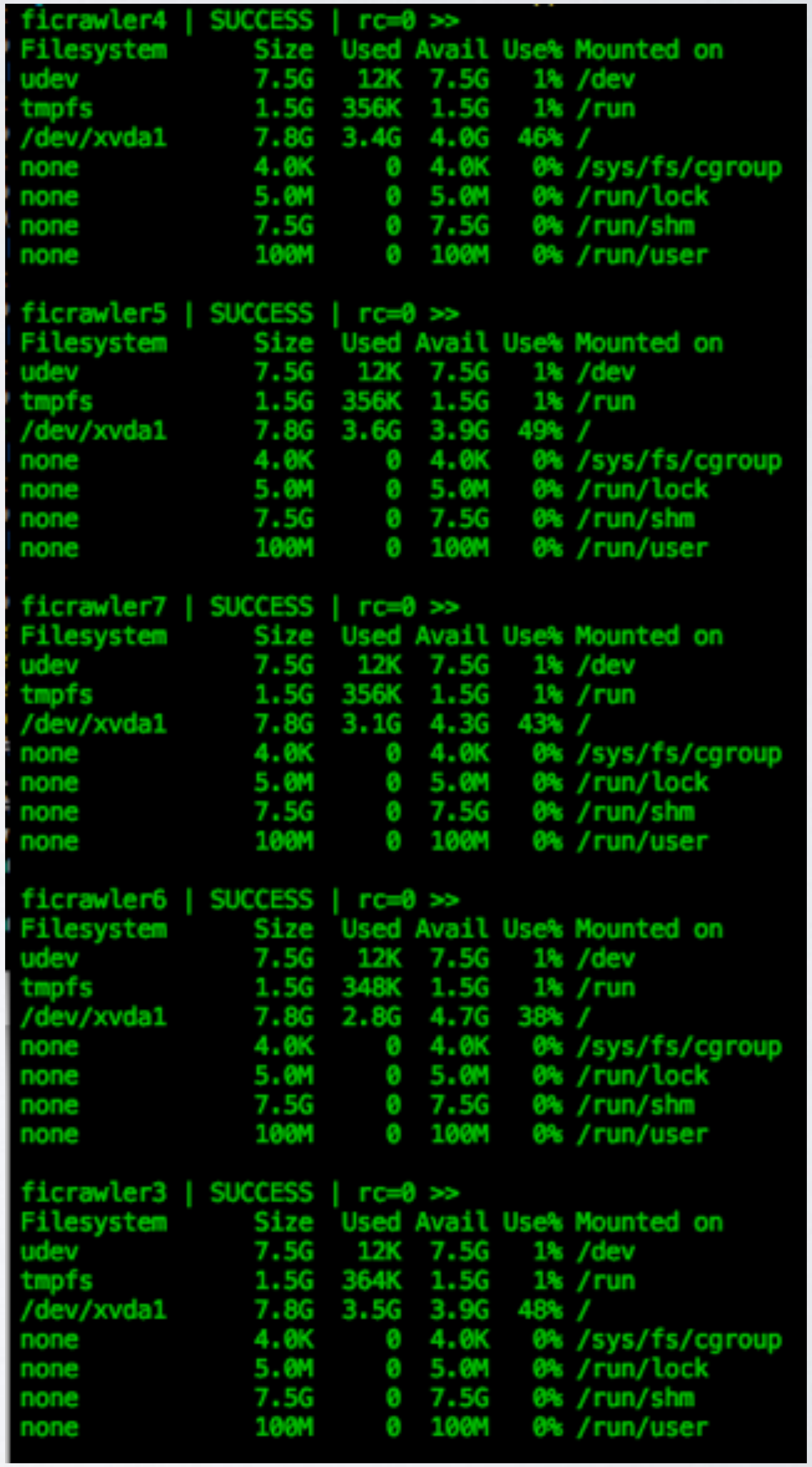
- Ansible is a reaction to “snowflake servers” - finely tuned over a period of time
- Throw Away Servers
- Never Fix a Server Again

2 APPROACHES

- Ad Hoc
  - Want to do anything on a group of machines
- Playbook
  - Want to do the same thing on a group of machines over and over

AD HOC

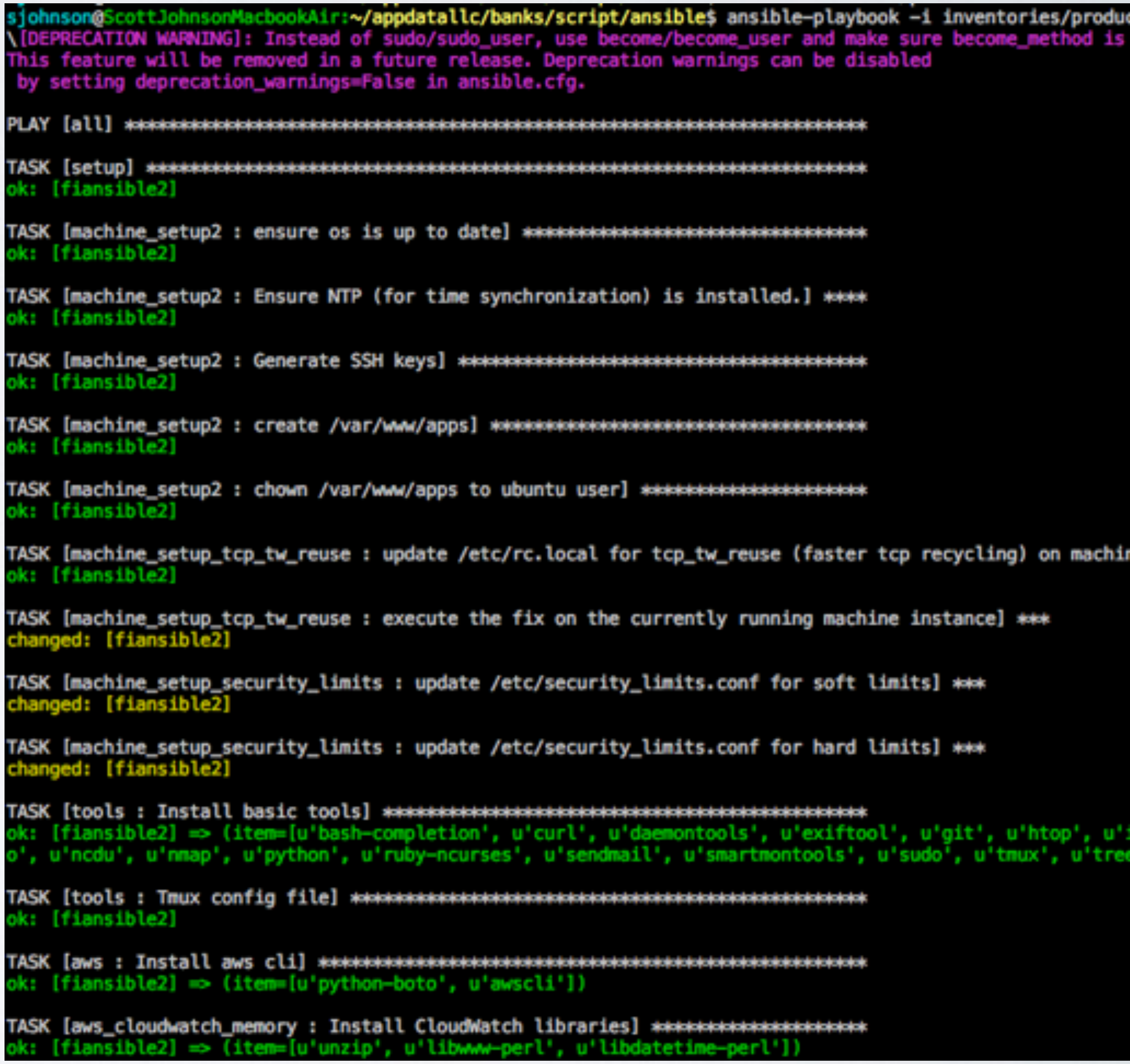
- ansible all -i inventories/ production_more_crawlers -u ubuntu -a "df -h"



```
ficrawler4 | SUCCESS | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev            7.5G   12K  7.5G   1% /dev
tmpfs           1.5G  356K  1.5G   1% /run
/dev/xvda1      7.8G  3.4G  4.0G  46% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            5.0M     0  5.0M   0% /run/lock
none            7.5G     0  7.5G   0% /run/shm
none            100M     0  100M   0% /run/user

ficrawler5 | SUCCESS | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev            7.5G   12K  7.5G   1% /dev
tmpfs           1.5G  356K  1.5G   1% /run
/dev/xvda1      7.8G  3.6G  3.9G  49% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            5.0M     0  5.0M   0% /run/lock
none            7.5G     0  7.5G   0% /run/shm
none            100M     0  100M   0% /run/user

ficrawler7 | SUCCESS | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev            7.5G   12K  7.5G   1% /dev
tmpfs           1.5G  356K  1.5G   1% /run
/dev/xvda1      7.8G  3.1G  4.3G  43% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            5.0M     0  5.0M   0% /run/lock
none            7.5G     0  7.5G   0% /run/shm
none            100M     0  100M   0% /run/user

ficrawler6 | SUCCESS | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev            7.5G   12K  7.5G   1% /dev
tmpfs           1.5G  348K  1.5G   1% /run
/dev/xvda1      7.8G  2.8G  4.7G  38% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            5.0M     0  5.0M   0% /run/lock
none            7.5G     0  7.5G   0% /run/shm
none            100M     0  100M   0% /run/user

ficrawler3 | SUCCESS | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
udev            7.5G   12K  7.5G   1% /dev
tmpfs           1.5G  364K  1.5G   1% /run
/dev/xvda1      7.8G  3.5G  3.9G  48% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            5.0M     0  5.0M   0% /run/lock
none            7.5G     0  7.5G   0% /run/shm
none            100M     0  100M   0% /run/user
```

PLAYBOOK

- ansible-playbook -i inventories/production playbook.yml



```
sjohnson@ScottJohnsonMacbookAir:~/appdatallc/banks/script/ansible$ ansible-playbook -i inventories/produc
\[DEPRECATION WARNING]: Instead of sudo/sudo_user, use become/become_user and make sure become_method is
This feature will be removed in a future release. Deprecation warnings can be disabled
 by setting deprecation_warnings=False in ansible.cfg.

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

TASK [setup] *******************************************************************
ok: [fiansible2]

TASK [machine_setup2 : ensure os is up to date] ********************************
ok: [fiansible2]

TASK [machine_setup2 : Ensure NTP (for time synchronization) is installed.] ****
ok: [fiansible2]

TASK [machine_setup2 : Generate SSH keys] **************************************
ok: [fiansible2]

TASK [machine_setup2 : create /var/www/apps] ***********************************
ok: [fiansible2]

TASK [machine_setup2 : chown /var/www/apps to ubuntu user] *********************
ok: [fiansible2]

TASK [machine_setup_tcp_tw_reuse : update /etc/rc.local for tcp_tw_reuse (faster tcp recycling) on machin
ok: [fiansible2]

TASK [machine_setup_tcp_tw_reuse : execute the fix on the currently running machine instance] ***
changed: [fiansible2]

TASK [machine_setup_security_limits : update /etc/security_limits.conf for soft limits] ***
changed: [fiansible2]

TASK [machine_setup_security_limits : update /etc/security_limits.conf for hard limits] ***
changed: [fiansible2]

TASK [tools : Install basic tools] *********************************************
ok: [fiansible2] => (item=[u'bash-completion', u'curl', u'daemontools', u'exiftool', u'git', u'htop', u'i
o', u'ncdu', u'nmap', u'python', u'ruby-ncurses', u'sendmail', u'smartmontools', u'sudo', u'tmux', u'tree

TASK [tools : Tmux config file] ************************************************
ok: [fiansible2]

TASK [aws : Install aws cli] ***************************************************
ok: [fiansible2] => (item=[u'python-boto', u'awscli'])

TASK [aws_cloudwatch_memory : Install CloudWatch libraries] ********************
ok: [fiansible2] => (item=[u'unzip', u'libwww-perl', u'libdatetime-perl'])
```

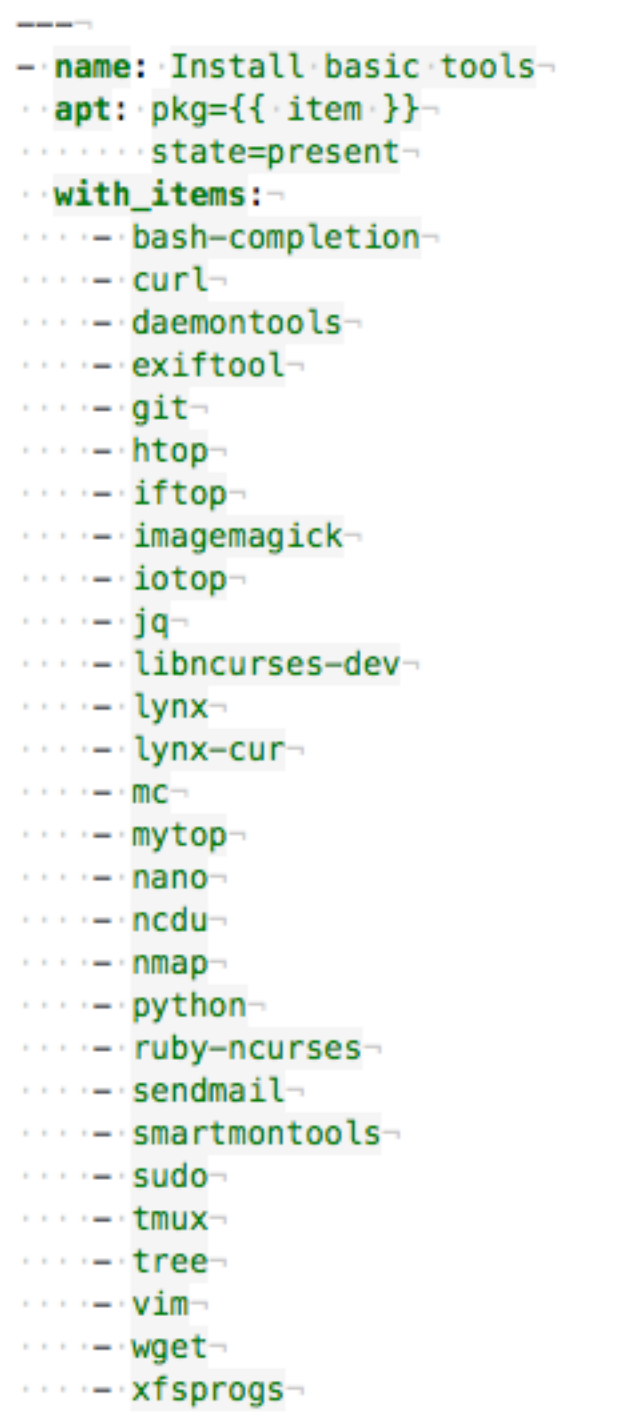
CORE ANSIBLE CONCEPTS

- **Idempotency**
- Playbook
- Inventory
- Role
- Task
- Variables

- Conditionals
- Files
- Templates
- Handlers
- Vault
- Galaxy

IDEMPOTENCY

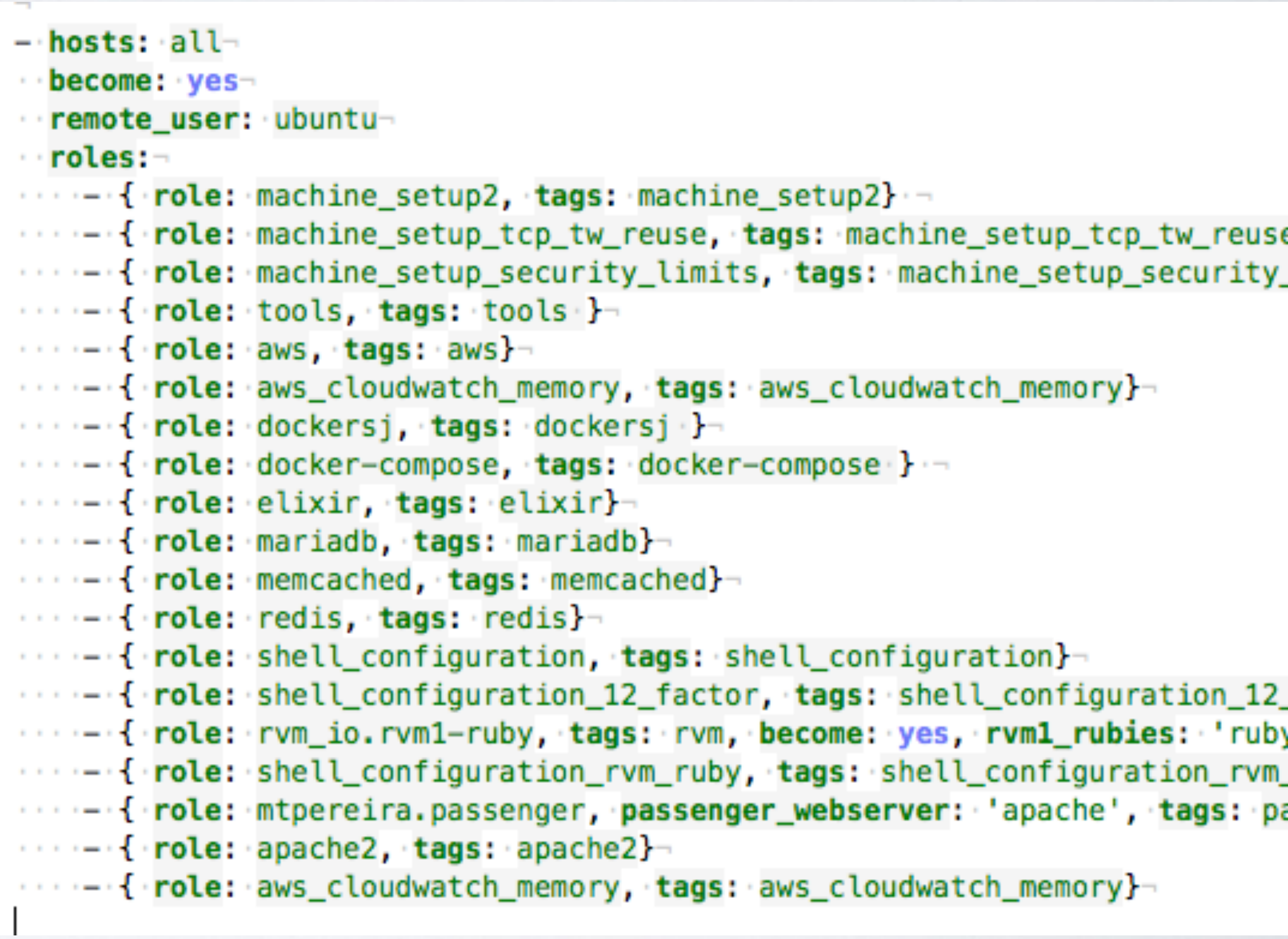
- Do it repeatedly; get same result!
- Old hat to us - we're functional!
- Happens at task level
- state = SOMETHING
  - something varies based on module (present versus started)
- modules are generally idempotent
- get into shell stuff and you do it on your own



```yaml
---
- name: Install basic tools
  apt: pkg={{ item }}
       state=present
  with_items:
    - bash-completion
    - curl
    - daemontools
    - exiftool
    - git
    - htop
    - iftop
    - imagemagick
    - iotop
    - jq
    - libncurses-dev
    - lynx
    - lynx-cur
    - mc
    - mytop
    - nano
    - ncdu
    - nmap
    - python
    - ruby-ncurses
    - sendmail
    - smartmontools
    - sudo
    - tmux
    - tree
    - vim
    - wget
    - xfsprogs
```

PLAYBOOK

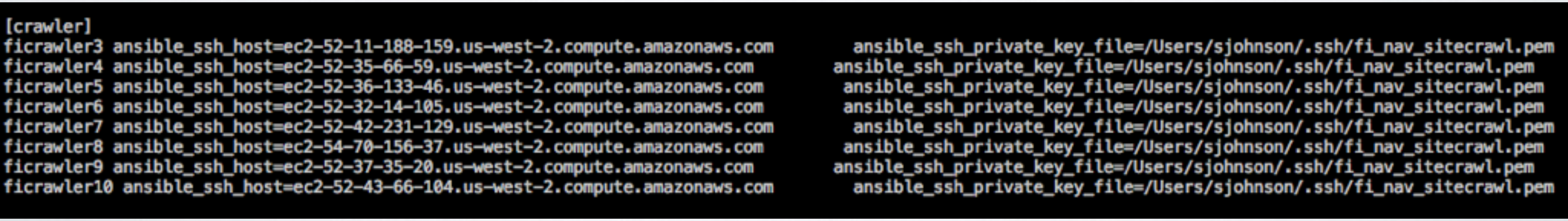
- What to Do
- What Order to do it in
- YAML file



```yaml
- hosts: all
  become: yes
  remote_user: ubuntu
  roles:
    - { role: machine_setup2, tags: machine_setup2}
    - { role: machine_setup_tcp_tw_reuse, tags: machine_setup_tcp_tw_reuse
    - { role: machine_setup_security_limits, tags: machine_setup_security_
    - { role: tools, tags: tools }
    - { role: aws, tags: aws}
    - { role: aws_cloudwatch_memory, tags: aws_cloudwatch_memory}
    - { role: dockersj, tags: dockersj }
    - { role: docker-compose, tags: docker-compose }
    - { role: elixir, tags: elixir}
    - { role: mariadb, tags: mariadb}
    - { role: memcached, tags: memcached}
    - { role: redis, tags: redis}
    - { role: shell_configuration, tags: shell_configuration}
    - { role: shell_configuration_12_factor, tags: shell_configuration_12_
    - { role: rvm_io.rvm1-ruby, tags: rvm, become: yes, rvm1_rubies: 'ruby
    - { role: shell_configuration_rvm_ruby, tags: shell_configuration_rvm_
    - { role: mtpereira.passenger, passenger_webserver: 'apache', tags: pa
    - { role: apache2, tags: apache2}
    - { role: aws_cloudwatch_memory, tags: aws_cloudwatch_memory}
```

INVENTORY

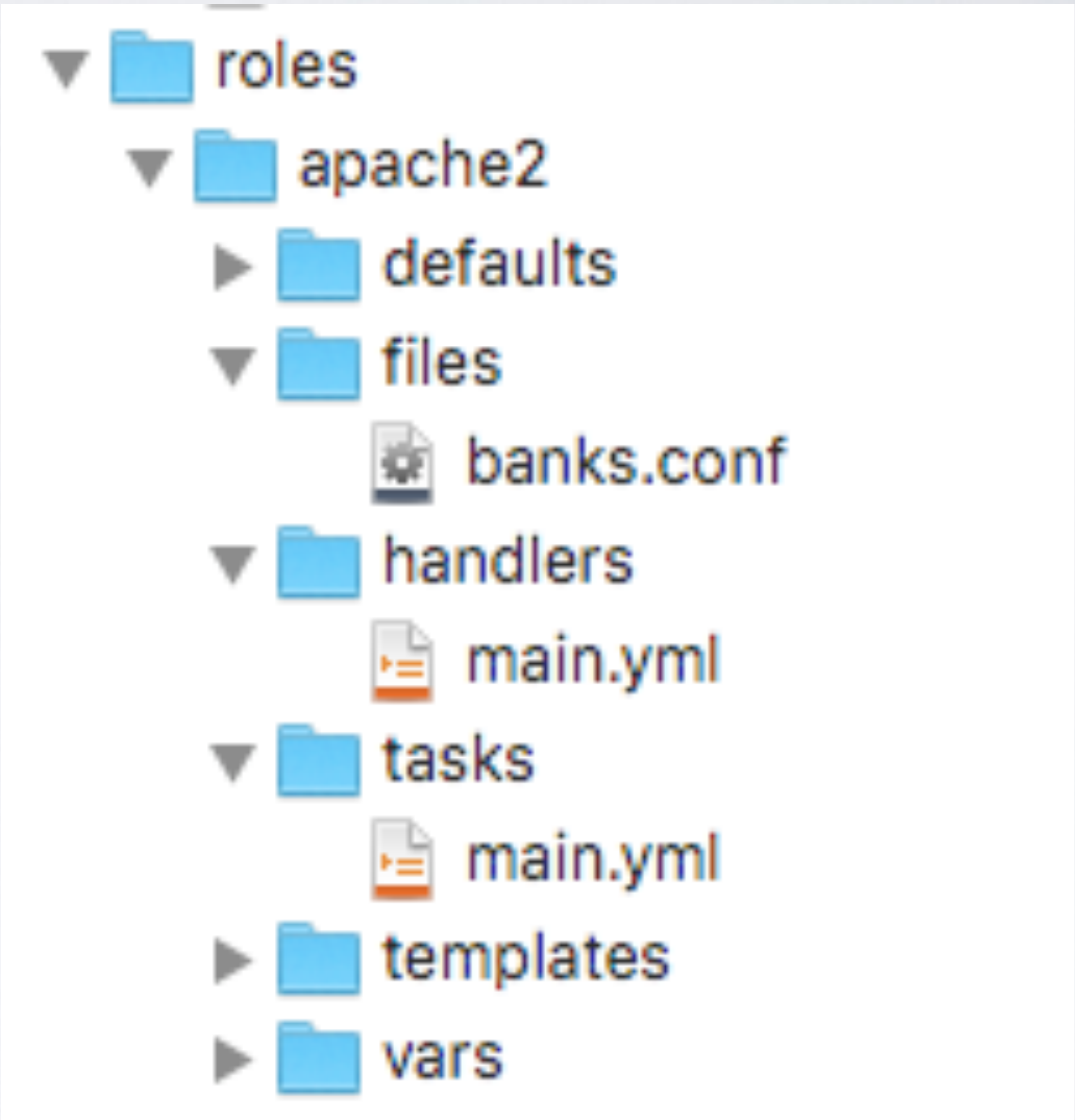
- What machines to do something to
- plain ASCII
- can be dynamic



```
[crawler]
ficrawler3 ansible_ssh_host=ec2-52-11-188-159.us-west-2.compute.amazonaws.com       ansible_ssh_private_key_file=/Users/sjohnson/.ssh/fi_nav_sitecrawl.pem
ficrawler4 ansible_ssh_host=ec2-52-35-66-59.us-west-2.compute.amazonaws.com       ansible_ssh_private_key_file=/Users/sjohnson/.ssh/fi_nav_sitecrawl.pem
ficrawler5 ansible_ssh_host=ec2-52-36-133-46.us-west-2.compute.amazonaws.com       ansible_ssh_private_key_file=/Users/sjohnson/.ssh/fi_nav_sitecrawl.pem
ficrawler6 ansible_ssh_host=ec2-52-32-14-105.us-west-2.compute.amazonaws.com       ansible_ssh_private_key_file=/Users/sjohnson/.ssh/fi_nav_sitecrawl.pem
ficrawler7 ansible_ssh_host=ec2-52-42-231-129.us-west-2.compute.amazonaws.com       ansible_ssh_private_key_file=/Users/sjohnson/.ssh/fi_nav_sitecrawl.pem
ficrawler8 ansible_ssh_host=ec2-54-70-156-37.us-west-2.compute.amazonaws.com       ansible_ssh_private_key_file=/Users/sjohnson/.ssh/fi_nav_sitecrawl.pem
ficrawler9 ansible_ssh_host=ec2-52-37-35-20.us-west-2.compute.amazonaws.com       ansible_ssh_private_key_file=/Users/sjohnson/.ssh/fi_nav_sitecrawl.pem
ficrawler10 ansible_ssh_host=ec2-52-43-66-104.us-west-2.compute.amazonaws.com       ansible_ssh_private_key_file=/Users/sjohnson/.ssh/fi_nav_sitecrawl.pem
```

ROLE

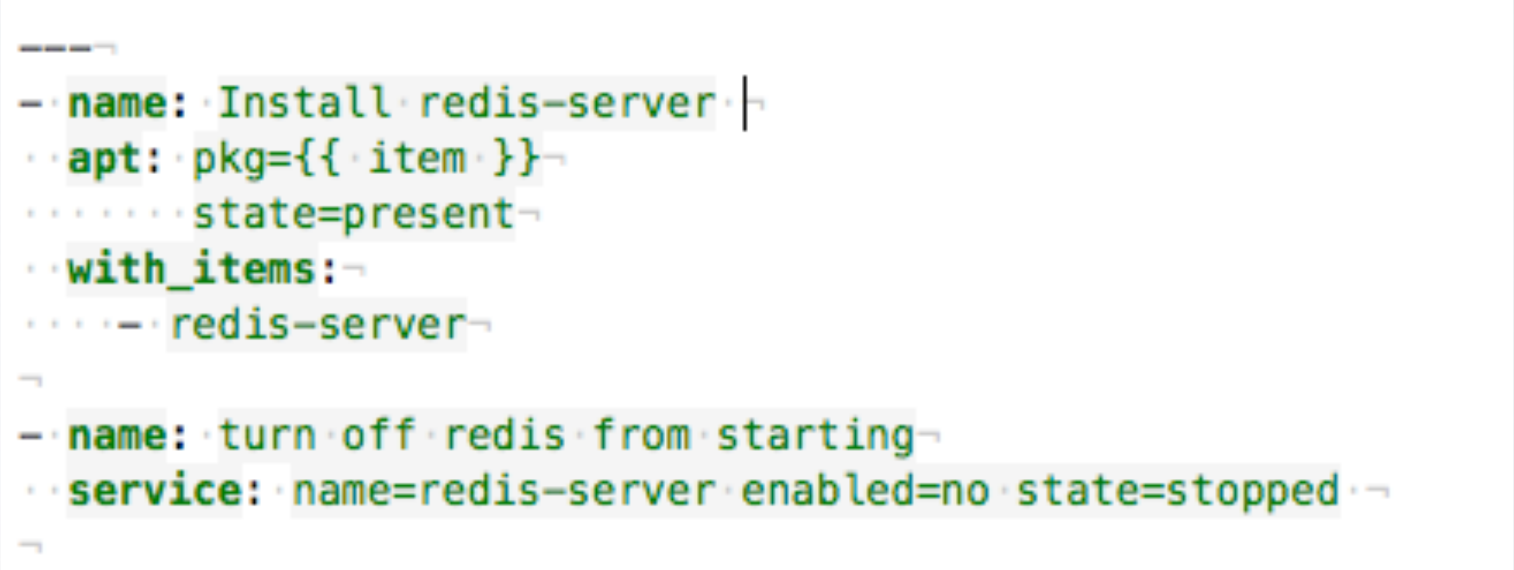
- YAML file
- Highly structured but can only have what you need
- Describes what to do
- module based
- ideally should be idempotent
- appears in the playbook



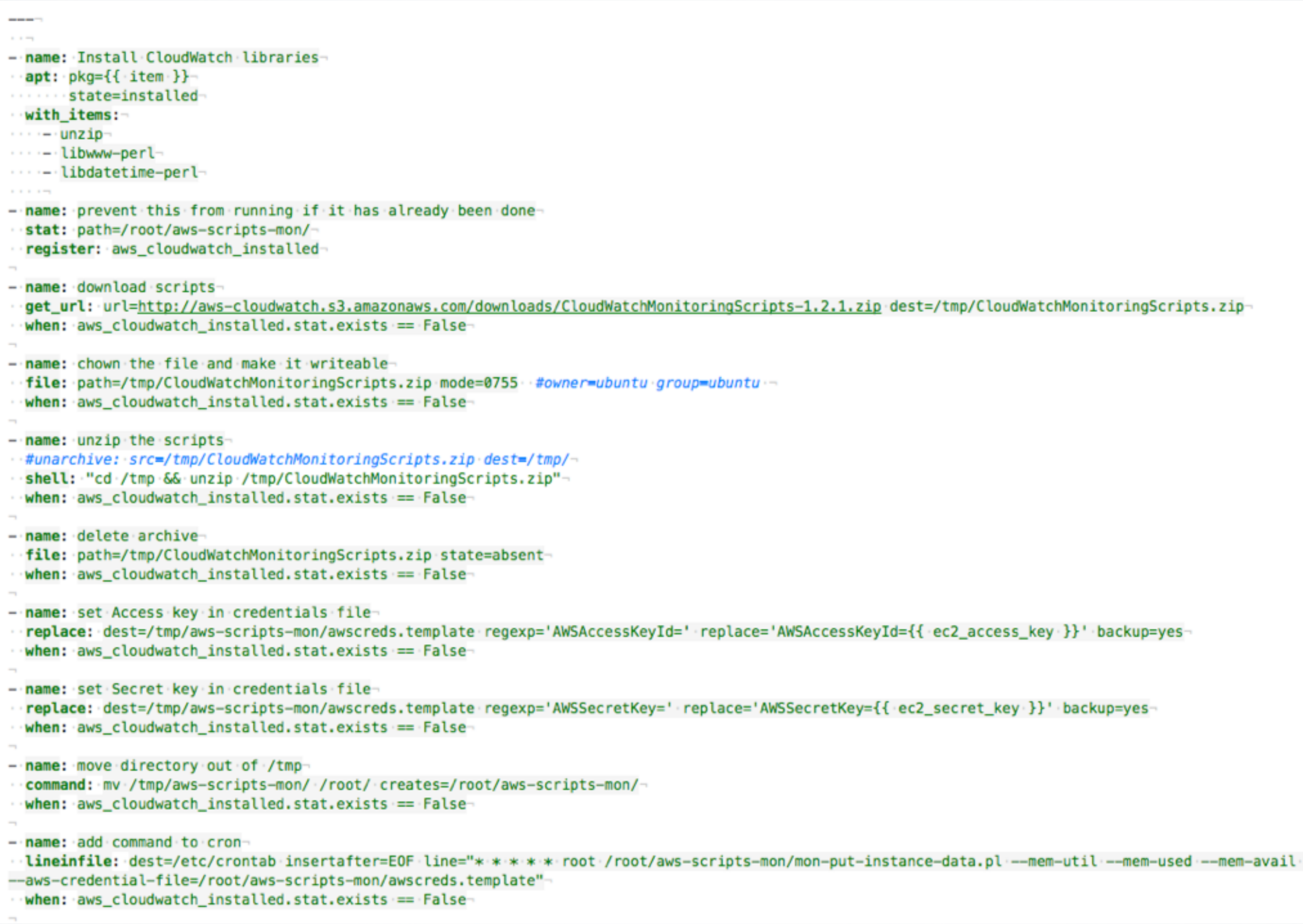
```
▼ roles
  ▼ apache2
    ▶ defaults
    ▼ files
        banks.conf
    ▼ handlers
        main.yml
    ▼ tasks
        main.yml
    ▶ templates
    ▶ vars
```

TASK

- What to do
- module based
- declarative
- name is optional / documentation
- apt: is a module
- with_items is an iterator
- service: is a module



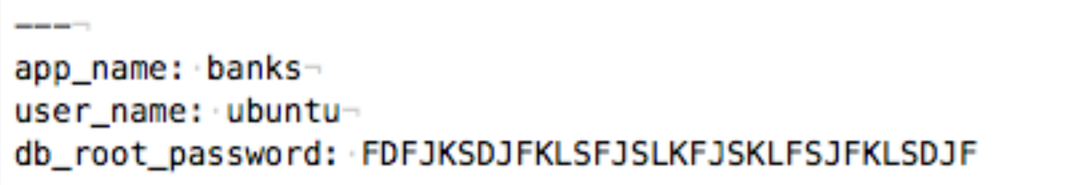
```
---
- name: Install redis-server
  apt: pkg={{ item }}
       state=present
  with_items:
    - redis-server

- name: turn off redis from starting
  service: name=redis-server enabled=no state=stopped

```



```yaml
---
  
- name: Install CloudWatch libraries
  apt: pkg={{ item }}
       state=installed
  with_items:
    - unzip
    - libwww-perl
    - libdatetime-perl
    
- name: prevent this from running if it has already been done
  stat: path=/root/aws-scripts-mon/
  register: aws_cloudwatch_installed

- name: download scripts
  get_url: url=http://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.1.zip dest=/tmp/CloudWatchMonitoringScripts.zip
  when: aws_cloudwatch_installed.stat.exists == False

- name: chown the file and make it writeable
  file: path=/tmp/CloudWatchMonitoringScripts.zip mode=0755  #owner=ubuntu group=ubuntu 
  when: aws_cloudwatch_installed.stat.exists == False

- name: unzip the scripts
  #unarchive: src=/tmp/CloudWatchMonitoringScripts.zip dest=/tmp/
  shell: "cd /tmp && unzip /tmp/CloudWatchMonitoringScripts.zip"
  when: aws_cloudwatch_installed.stat.exists == False

- name: delete archive
  file: path=/tmp/CloudWatchMonitoringScripts.zip state=absent
  when: aws_cloudwatch_installed.stat.exists == False

- name: set Access key in credentials file
  replace: dest=/tmp/aws-scripts-mon/awscreds.template regexp='AWSAccessKeyId=' replace='AWSAccessKeyId={{ ec2_access_key }}' backup=yes
  when: aws_cloudwatch_installed.stat.exists == False

- name: set Secret key in credentials file
  replace: dest=/tmp/aws-scripts-mon/awscreds.template regexp='AWSSecretKey=' replace='AWSSecretKey={{ ec2_secret_key }}' backup=yes
  when: aws_cloudwatch_installed.stat.exists == False

- name: move directory out of /tmp
  command: mv /tmp/aws-scripts-mon/ /root/ creates=/root/aws-scripts-mon/
  when: aws_cloudwatch_installed.stat.exists == False

- name: add command to cron
  lineinfile: dest=/etc/crontab insertafter=EOF line="* * * * * root /root/aws-scripts-mon/mon-put-instance-data.pl --mem-util --mem-used --mem-avail --aws-credential-file=/root/aws-scripts-mon/awscreds.template"
  when: aws_cloudwatch_installed.stat.exists == False

```

VARIABLES

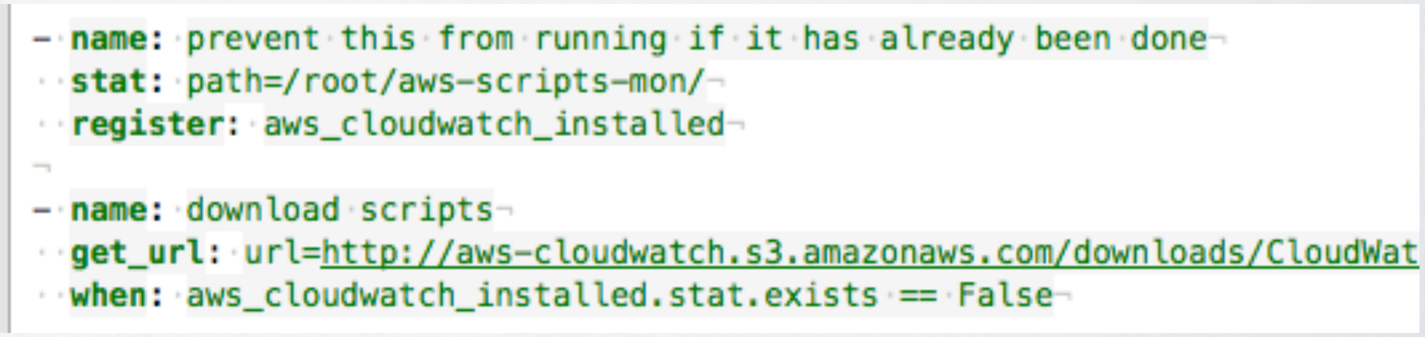
- Wonky
- Programming in YAML!!!
- Can exist at any level
  - group_vars
    - all
    - production
    - staging
  - role



```yaml
---
app_name: banks
user_name: ubuntu
db_root_password: FDFJKSDJFKLSFJSLKFJSKLFSJFKLSDJF
```

CONDITIONAL

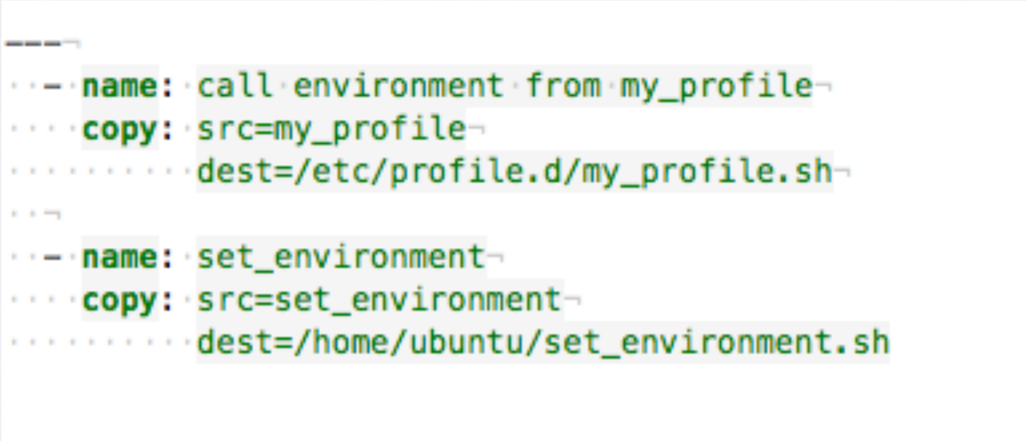
- Expressions that determine when (or when not) to do something
- Feel awkward
- Again programming in YAML!



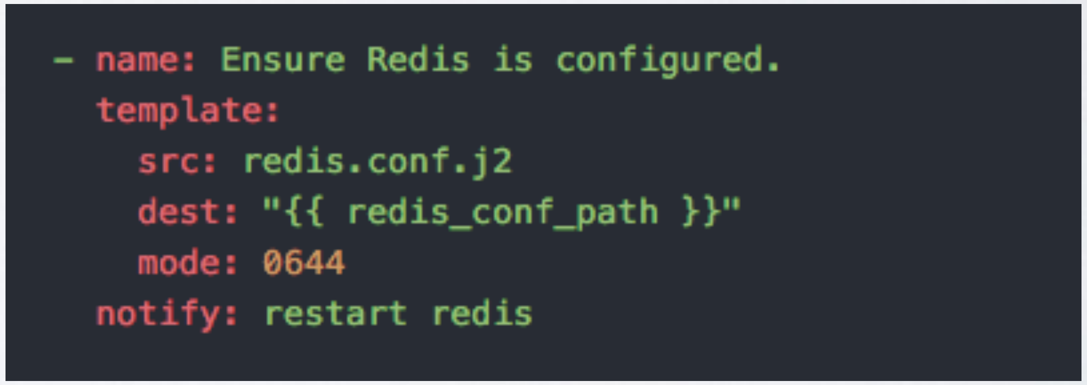
```
- name: prevent this from running if it has already been done
  stat: path=/root/aws-scripts-mon/
  register: aws_cloudwatch_installed

- name: download scripts
  get_url: url=http://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWat
  when: aws_cloudwatch_installed.stat.exists == False
```

FILES / TEMPLATES

- Files are static entities copied up as part of a task
- Templates are modified as part of copy action
- Templates support variables



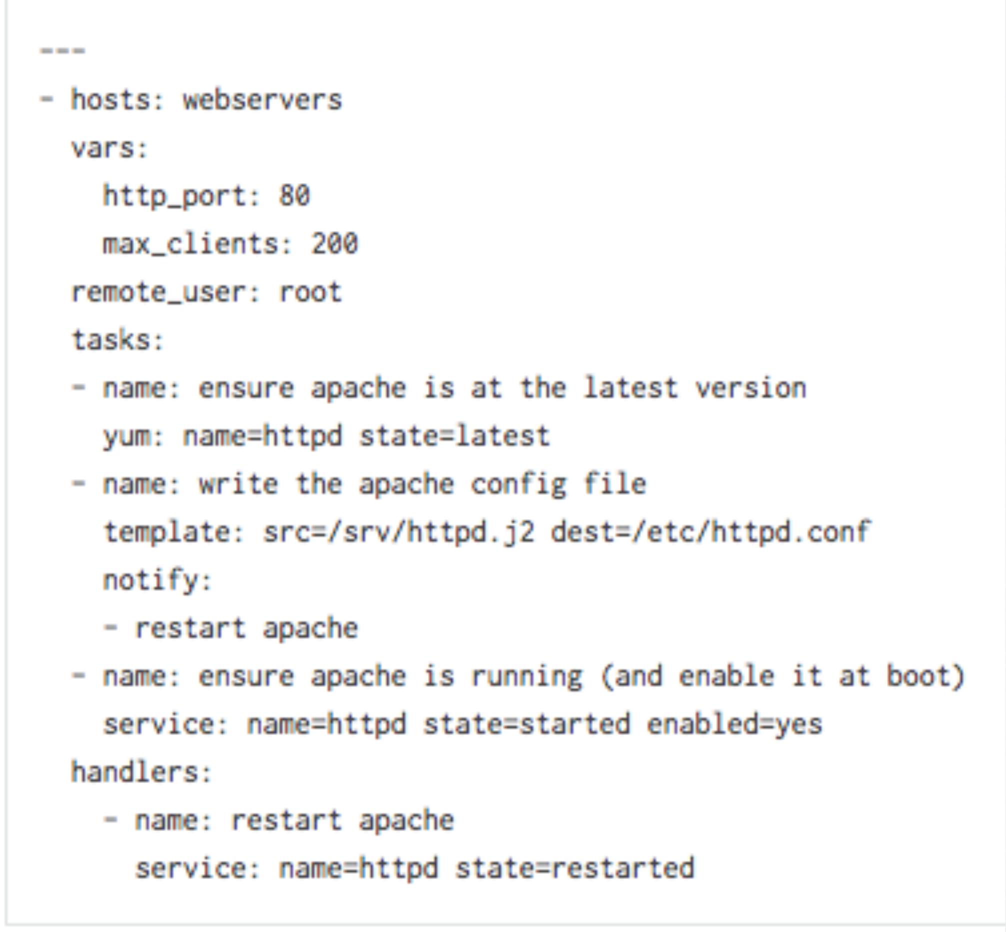
```
---
  - name: call environment from my_profile
    copy: src=my_profile
          dest=/etc/profile.d/my_profile.sh

  - name: set_environment
    copy: src=set_environment
          dest=/home/ubuntu/set_environment.sh
```



```
- name: Ensure Redis is configured.
  template:
    src: redis.conf.j2
    dest: "{{ redis_conf_path }}"
    mode: 0644
  notify: restart redis
```

HANDLERS

- Notification initiated tasks
- If not notified will not run at all
- Will only run once at the end of the task
- Suggestion: DO NOT USE SPACES IN THE NAME!



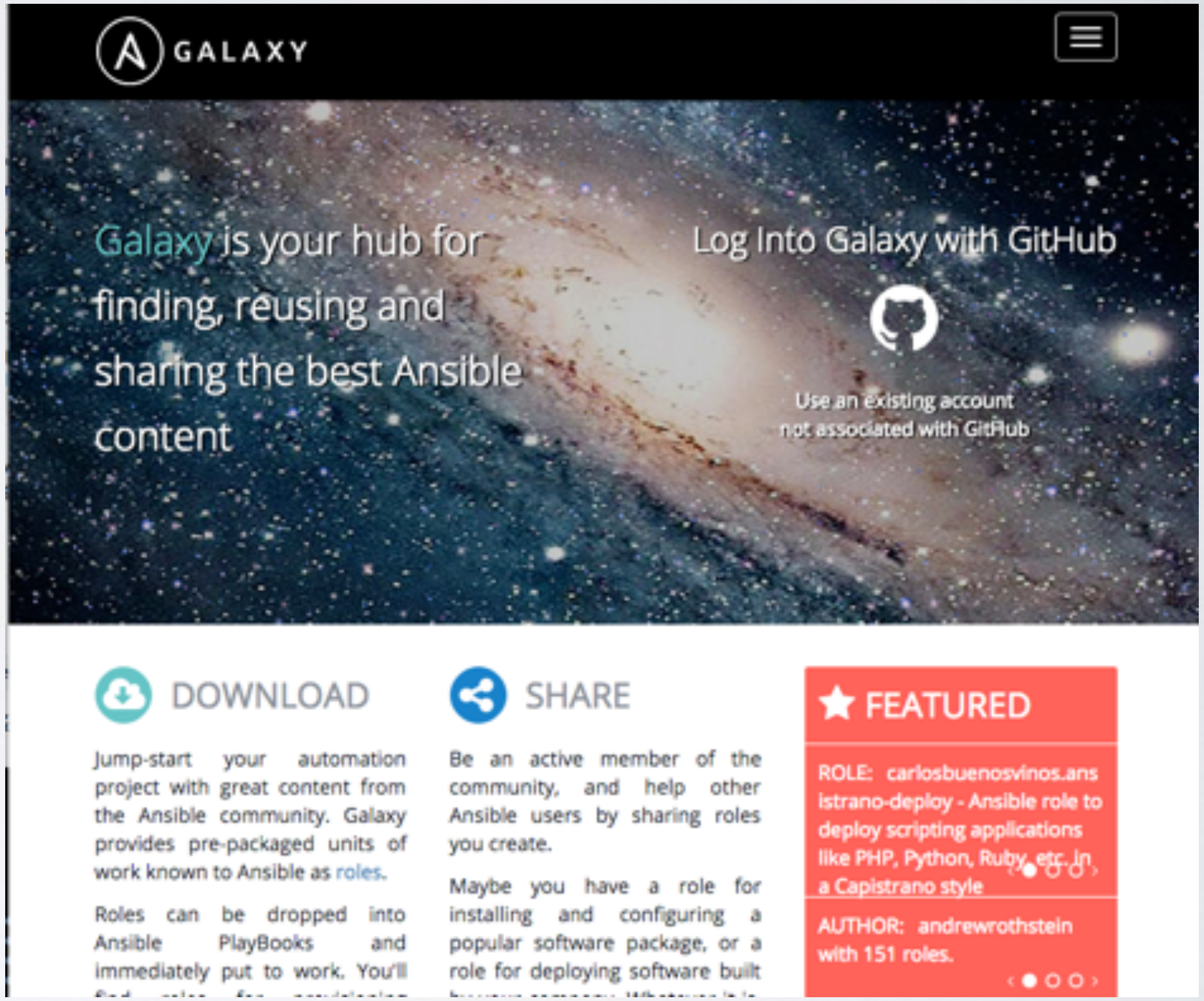
```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: write the apache config file
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running (and enable it at boot)
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

VAULT

- Ability to Encrypt the secrets what's in your playbook
- Be asked for password or pass in from file
- Keep security keys out of github

GALAXY

- Think DockerHub for DevOps
- Treat with Caution
- Often Doesn't Work
- Requires local installation before execution


GALAXY
Log Into Galaxy with GitHub
Use an existing account not associated with GitHub
DOWNLOAD
SHARE
FEATURED

Galaxy is your hub for finding, reusing and sharing the best Ansible content

Jump-start your automation project with great content from the Ansible community. Galaxy provides pre-packaged units of work known to Ansible as roles.

Roles can be dropped into Ansible PlayBooks and immediately put to work. You'll

Be an active member of the community, and help other Ansible users by sharing roles you create.

Maybe you have a role for installing and configuring a popular software package, or a role for deploying software built

ROLE: carlosbuenosvinos.ansistrano-deploy - Ansible role to deploy scripting applications like PHP, Python, Ruby, etc. in a Capistrano style

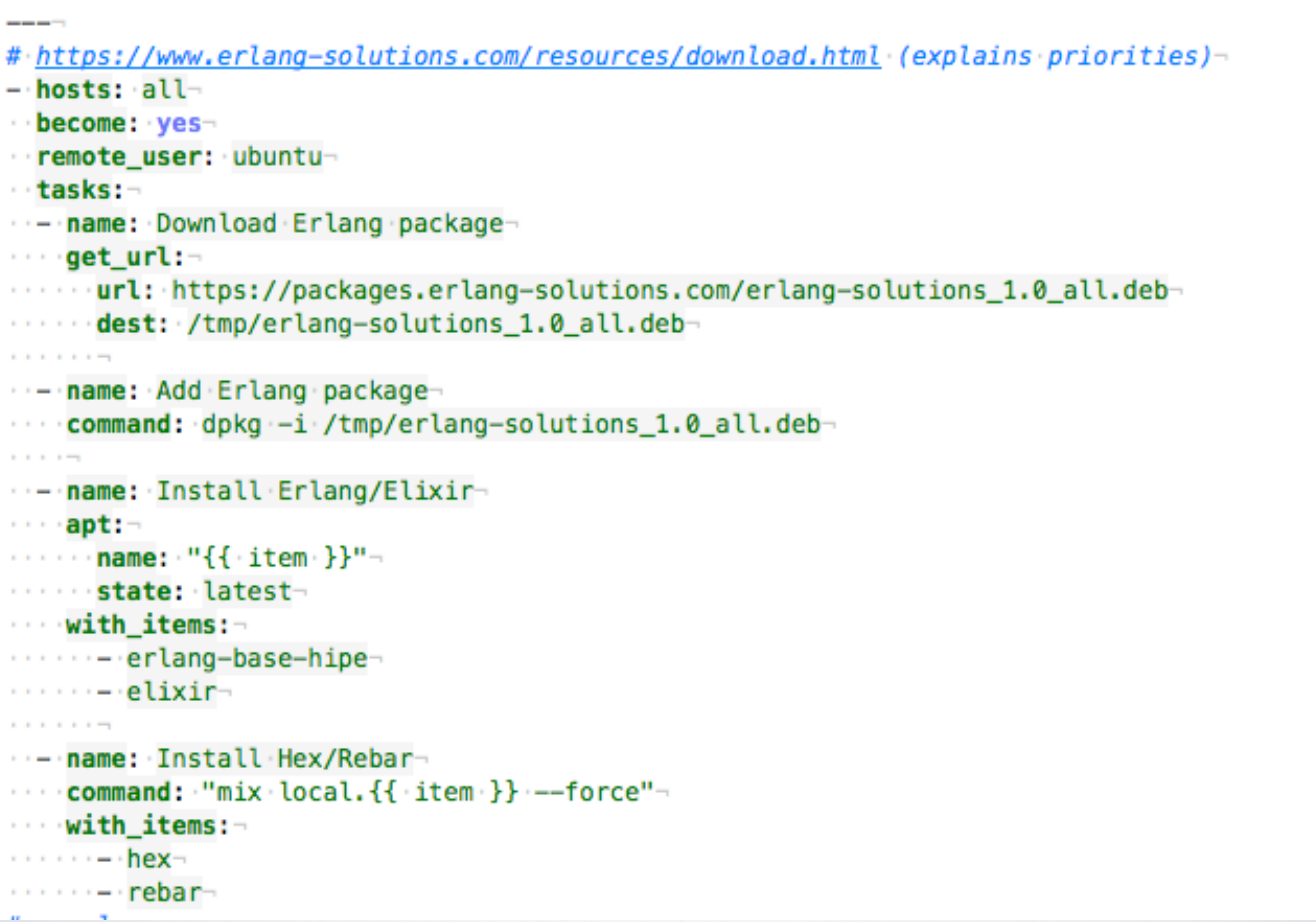
AUTHOR: andrewrothstein with 151 roles.

GETTING ELIXIR ON ALL YOUR MACHINES

- We want elixir on every machine we have!
- What has to happen
  - Erlang
  - Elixir
  - Tooling

APPROACH #1 MINE

- Learn what it takes to install everything for erlang, elixir, tooling
- Write a role
- Shout out to Thomas Lackemann @tlackemann who helped with this Thanks!

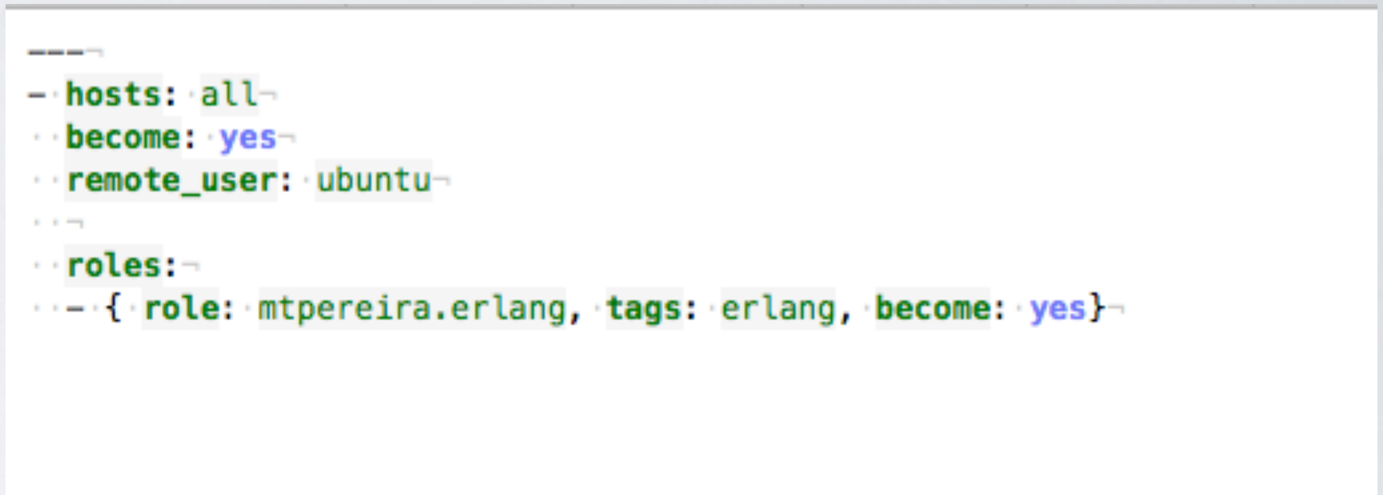


```yaml
---
# https://www.erlang-solutions.com/resources/download.html (explains priorities)
- hosts: all
  become: yes
  remote_user: ubuntu
  tasks:
  - name: Download Erlang package
    get_url:
      url: https://packages.erlang-solutions.com/erlang-solutions_1.0_all.deb
      dest: /tmp/erlang-solutions_1.0_all.deb

  - name: Add Erlang package
    command: dpkg -i /tmp/erlang-solutions_1.0_all.deb

  - name: Install Erlang/Elixir
    apt:
      name: "{{ item }}"
      state: latest
    with_items:
      - erlang-base-hipe
      - elixir

  - name: Install Hex/Rebar
    command: "mix local.{{ item }} --force"
    with_items:
      - hex
      - rebar
```

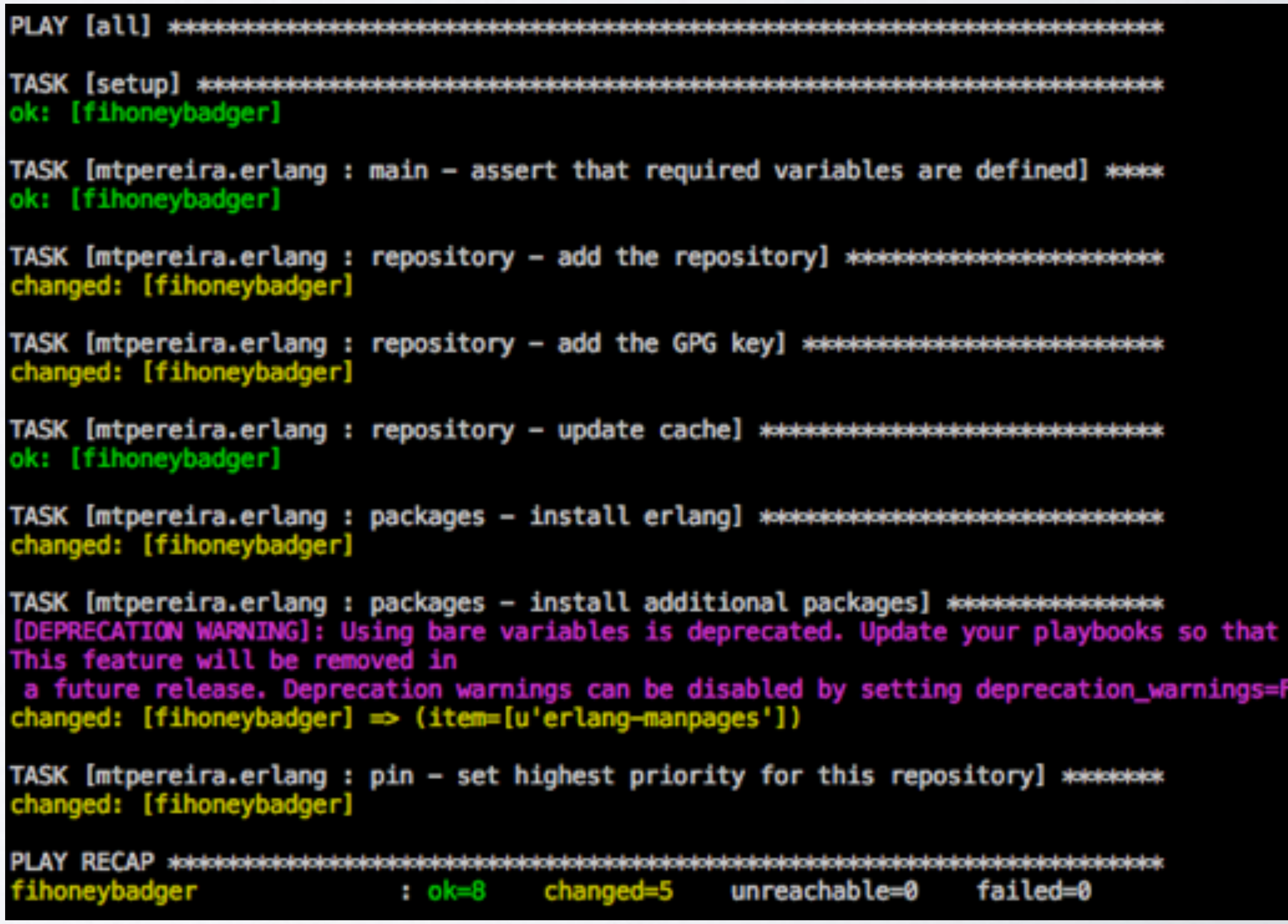
OPTION 2: USE GALAXY

- This role worked erratically for me - sometimes yes, sometimes no

- ansible-playbook -i inventories/ production_honeybadger playbook_erlang_and_elixir _via_galaxy.yml



```yaml
---
- hosts: all
  become: yes
  remote_user: ubuntu

  roles:
  - { role: mtpereira.erlang, tags: erlang, become: yes}
```

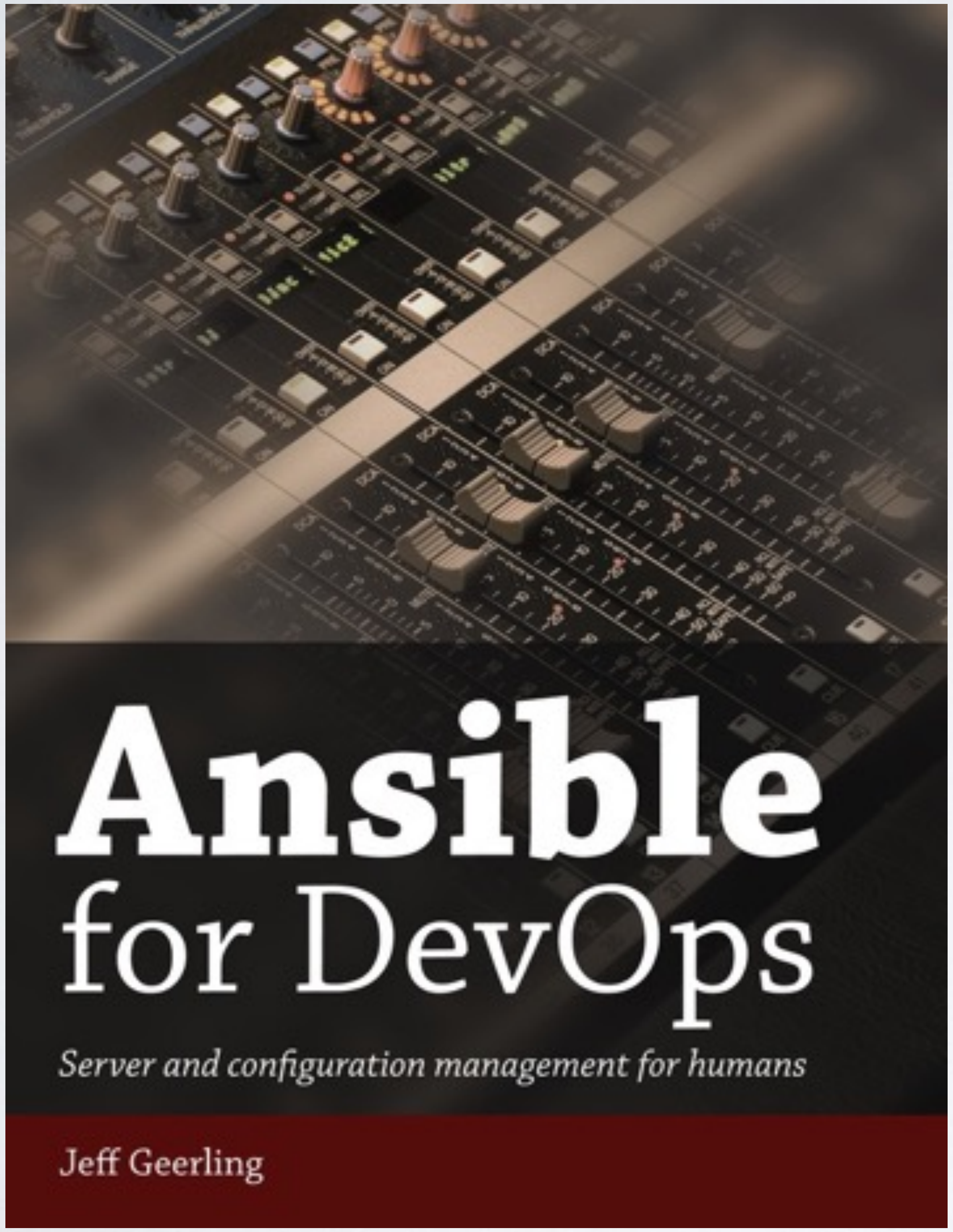


```
PLAY [all] *********************************************************************

TASK [setup] *******************************************************************
ok: [fihoneybadger]

TASK [mtpereira.erlang : main - assert that required variables are defined] ****
ok: [fihoneybadger]

TASK [mtpereira.erlang : repository - add the repository] **********************
changed: [fihoneybadger]

TASK [mtpereira.erlang : repository - add the GPG key] *************************
changed: [fihoneybadger]

TASK [mtpereira.erlang : repository - update cache] ****************************
ok: [fihoneybadger]

TASK [mtpereira.erlang : packages - install erlang] ****************************
changed: [fihoneybadger]

TASK [mtpereira.erlang : packages - install additional packages] ***************
[DEPRECATION WARNING]: Using bare variables is deprecated. Update your playbooks so that
This feature will be removed in
 a future release. Deprecation warnings can be disabled by setting deprecation_warnings=F
changed: [fihoneybadger] => (item=[u'erlang-manpages'])

TASK [mtpereira.erlang : pin - set highest priority for this repository] *******
changed: [fihoneybadger]

PLAY RECAP *********************************************************************
fihoneybadger              : ok=8    changed=5    unreachable=0    failed=0
```

CLOSING THOUGHTS / LEARNING ADVICE

- Use vagrant to test tasks
- Not everything works on vagrant
- Learning - look for **recent** examples; things change rapidly
- Conditionals are weird; when expressions are hard
- If you have a lot of ec2 boxes then provision from an ec2 box onto the other ec2 boxes (faster)
- Stuff breaks from time to time

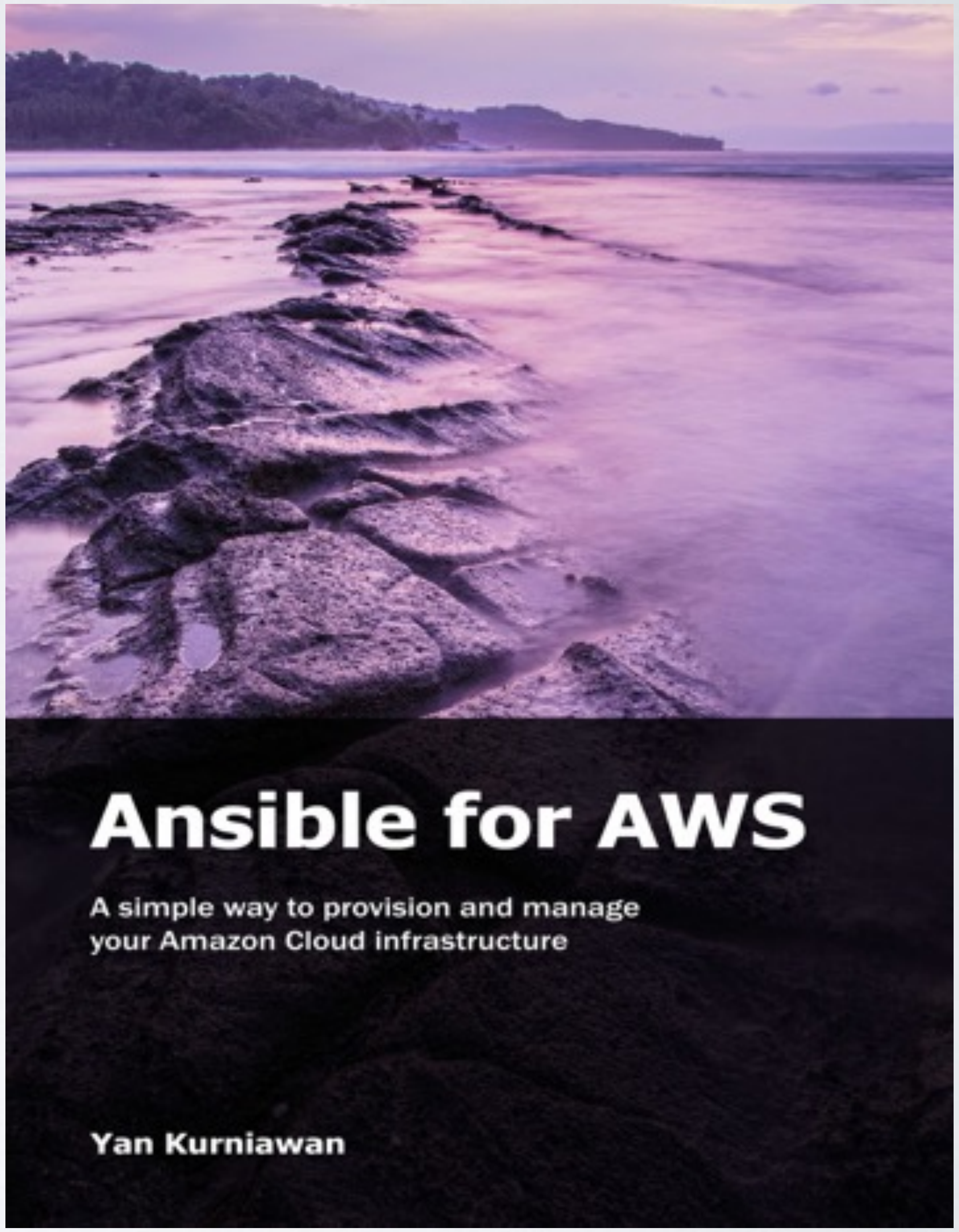
RECOMMENDED READING



Ansible
for DevOps

Server and configuration management for humans

Jeff Geerling



Ansible for AWS

A simple way to provision and manage
your Amazon Cloud infrastructure

Yan Kurniawan

www.leanpub.com