Never log in to your servers again with Ansible

Where does the name “ansible” come from?

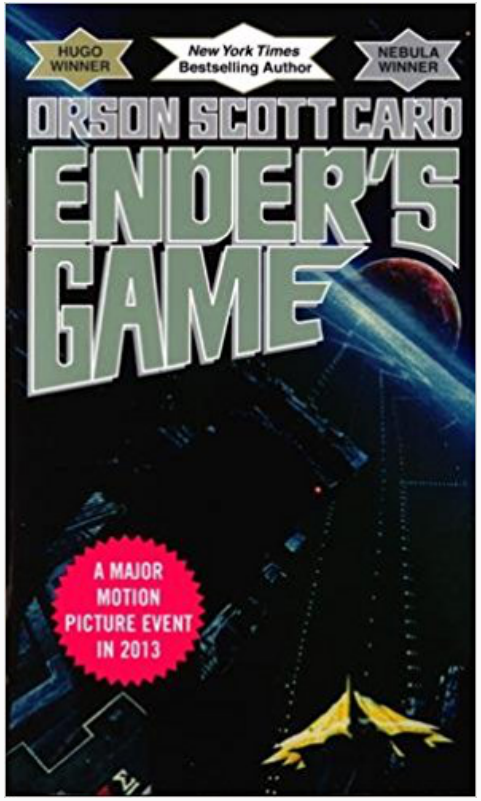
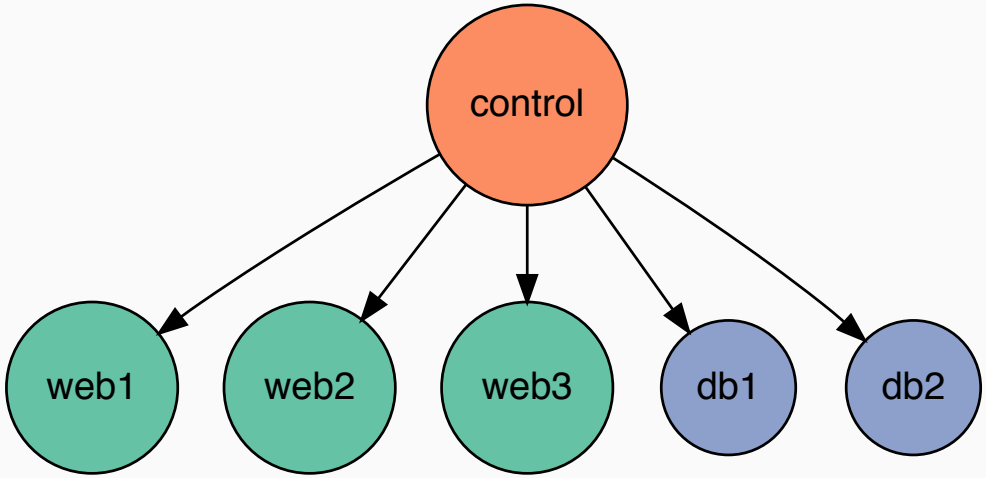


Figure 1: A sci-fi novel featuring an ansible as a communication device.

Use Ansible to talk to your fleet




```
# display free disk space
control/$ ansible web -a "df"
control/$ ansible db  -a "df"
control/$ ansible all -a "df"
```

Ad-hoc commands

```
# ansible [host] [-m module] [-a args]

# The default module is "command".
ansible web -a "df"
ansible web -m command -a "df"

# For piping, you need the "shell" module
ansible web -m shell -a "df | grep /dev/sda1"

# Could also use the "script" module
echo "df | grep /dev/sda1" > df_sda1.sh
ansible web -m script -a "df_sda1.sh"
```

Inventory files

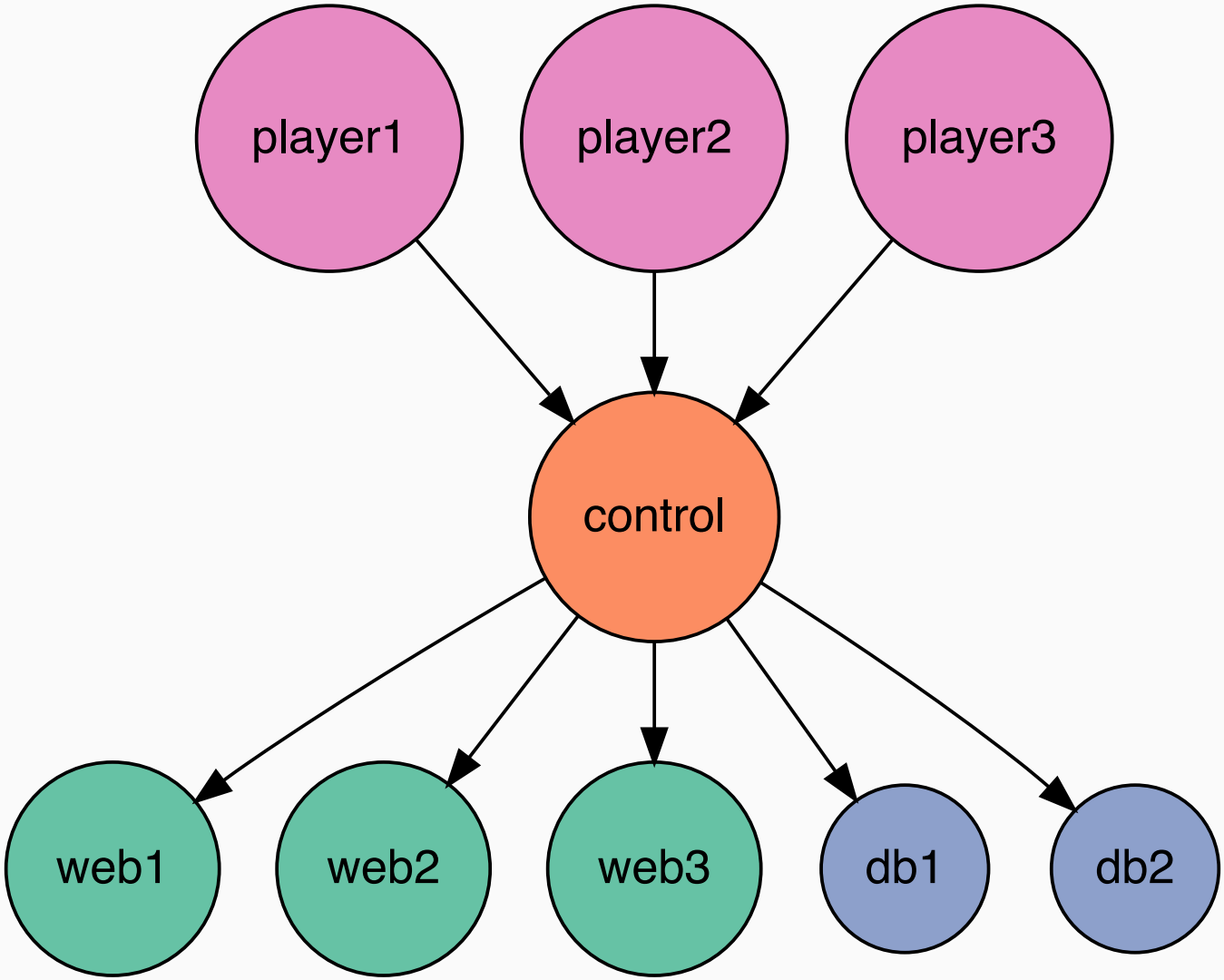
```
# contents of /etc/ansible/hosts

[web]
web1.example.com
web2.example.com
web3.example.com

[web:vars]
ansible_user=webhat

[db]
db1.example.com
db2.example.com
```

Dynamic inventory files

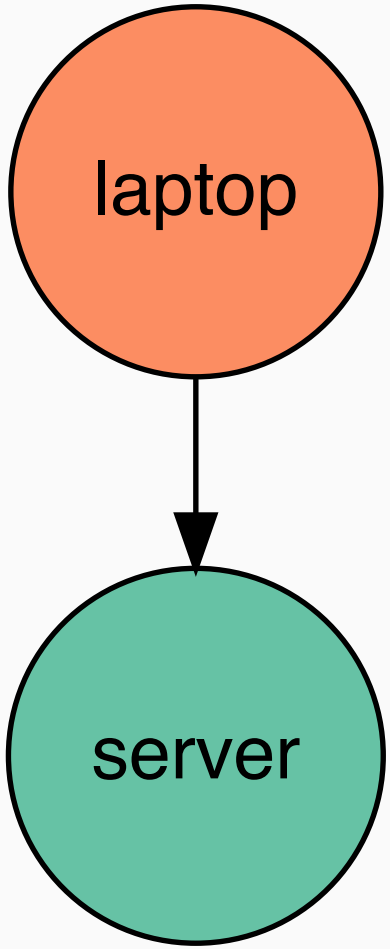
Any executable file can be used as a dynamic inventory!

```
# ping ec2 servers in us-east-1d
ansible us-east-1d -i ec2.py -u ubuntu -m ping
```

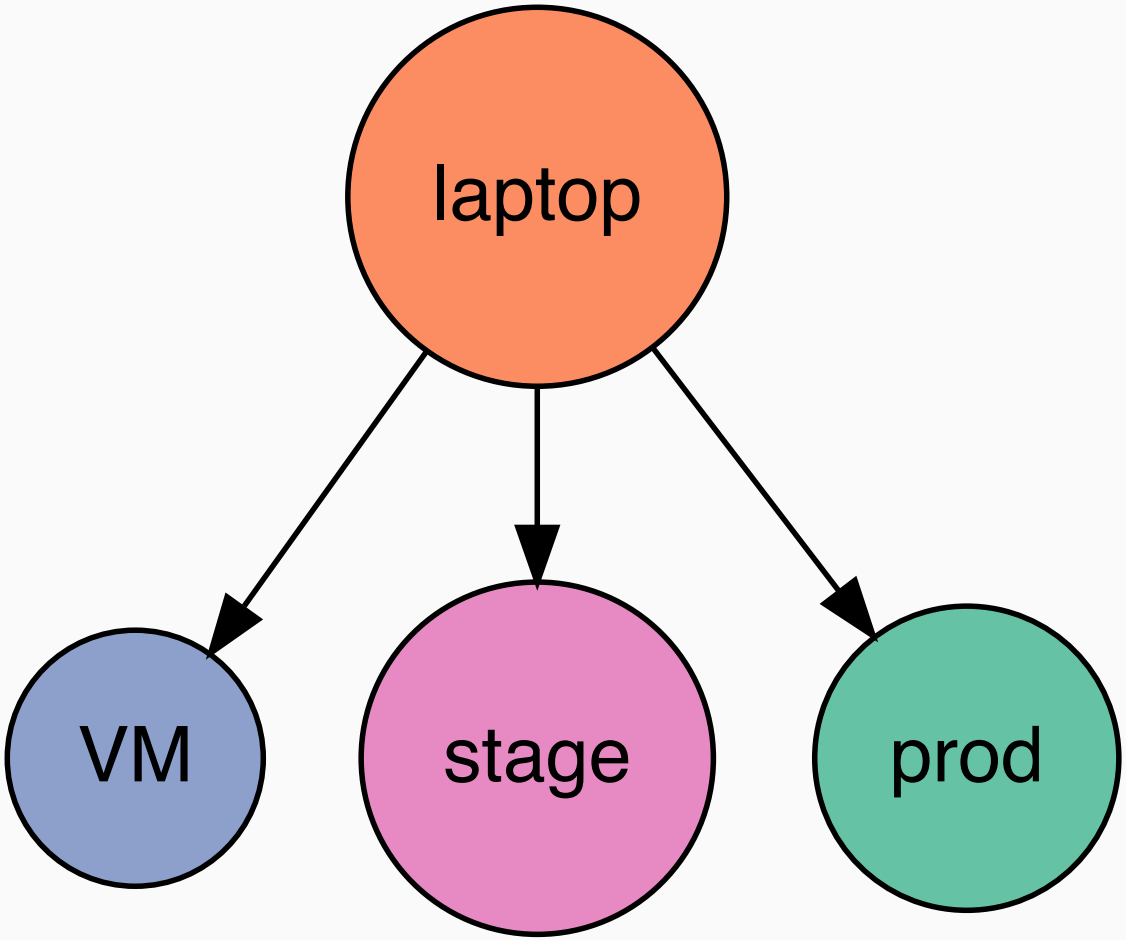
Share control of servers



Control a single server with Ansible



Continuous integration with Ansible



Provisioning on a Vagrant VM with Ansible

```
# Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.provision "ansible" do |ansible|
    ansible.playbook = "vagrant.yml"
  end
end
```

Command line tools

ansible Run an ad-hoc command.

```
ansible web -a "df"
```

playbook Run a playbook.

```
ansible-playbook setup.yml
```

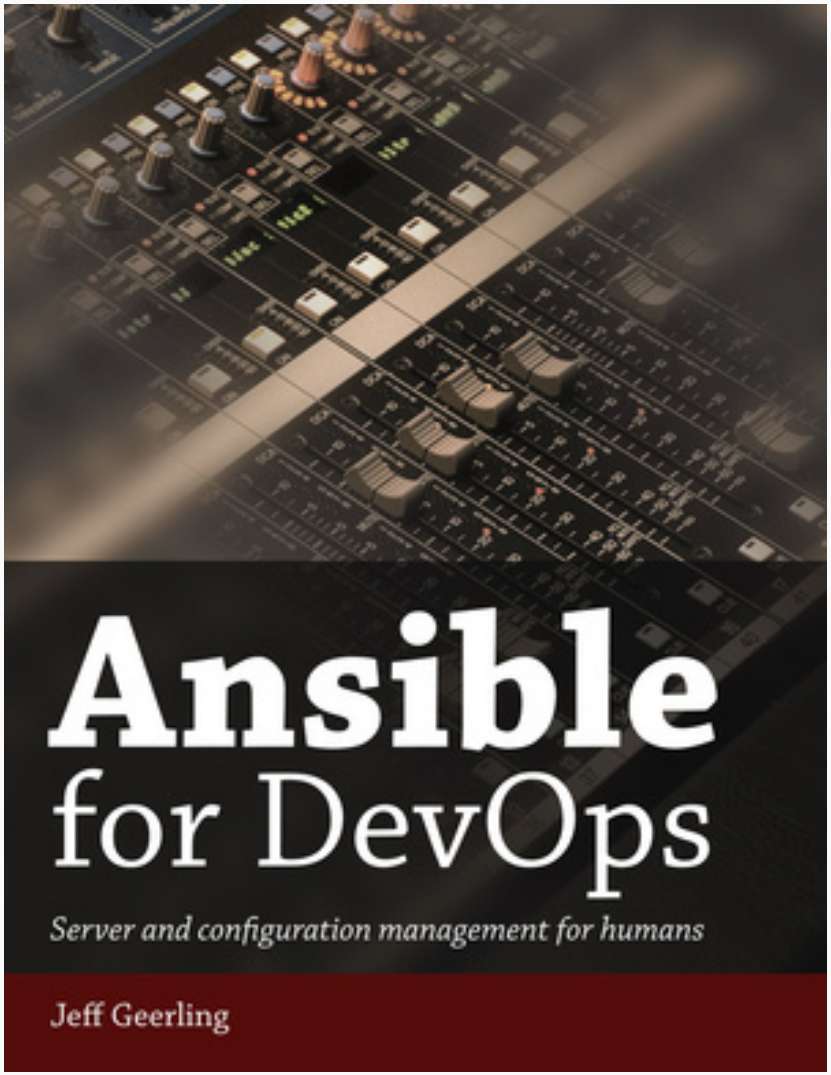
vault Simple encryption. Store a tough password in a password file, and set `ANSIBLE_VAULT_PASSWORD_FILE` environment variable.

```
ansible-vault encrypt vars/secrets.yml
```

galaxy 3rd party playbooks and dependencies.

```
ansible-galaxy install geerlingguy.mysql
```

Ansible for DevOps



Vocabulary

playbook A YAML file containing one or more plays.

play A collection of tasks to run on a host or hosts.

task A task is a single action with accompanying arguments, including a verbal description of the task.

module A module is command or command wrapper.

role A bundle of tasks and variables to accomplish some goal.

Example playbook: Setting up a MySQL DB

```yaml
---
# ansible-playbook mysql.yml
- hosts: db
  become: yes
  tasks:
    - name: Install MySQL
      apt: name=mysql-server state=present
```

YAML varieties

```
- apt: name=mysql-server state=present

- apt:
    name: mysql-server
    state: present
```

Example playbook: Setting up a MySQL DB

```yaml
---
- hosts: db
  become: yes
  vars:
    mysql_db_name: mydb
    mysql_root_password: dont-do-this
    mysql_user: pierce
    mysql_user_password: dont-do-this-either
  tasks:
    - include_role: name=geerlingguy.mysql
      vars:
        mysql_databases:
          - name: "{{ mysql_db_name }}"
        mysql_users:
          - name: "{{ mysql_user }}"
            host: "%"
            password: "{{ mysql_user_password }}"
            priv: "{{ mysql_db_name }}.*:ALL"
```

Example playbook: Setting up a MySQL DB

```yaml
---
- hosts: db
  become: yes
  vars_files:
    - vars/main.yml
    - vars/secrets.yml
  tasks:
    - name: Configure MySQL
      include_role:
        name: geerlingguy.mysql
```

Vars files

```yaml
---
# vars/main.yml

mysql_databases:
  - name: "{{ mysql_db_name }}"
mysql_users:
  - name: "{{ mysql_user }}"
    host: "%"
    password: "{{ mysql_user_password }}"
    priv: "{{ mysql_db_name }}.*:ALL"
```

Typical project structure

```
root/$
  playbook.yml
  vars/
  tasks/
  roles/
---
# playbook.yml
- hosts: all
  vars_files:
    - vars/main.yml
    - vars/secrets.yml
  tasks:
    - include_tasks: tasks/setup-webapp.yml
    - include_role: name=custom_role
```

Modules

apt Package manager for Debian/Ubuntu.

```
- name: Install bash and OpenSSL
  apt: name={{ item }} state=latest
  with_items:
    - bash
    - openssl
```

git Manage git repositories.

```
- git:
    repo: https://github.com/me/myrepo.git
    dest: /home/me/myrepo
```

pip Package manager for python.

```
- pip:
    requirements: /myproj/requirements.txt
    virtualenv: /venvs/myproj
```

Other modules

mysql_db Run commands on a MySQL db

```
- name: Dump the db
  mysql_db:
    name: mydb
    state: dump
    target: /dumps/today.sql
```

Other modules

fetch Retrieve a file from the host(s)

```
- fetch:
    src: /dumps/today.sql
    dest: /laptop/today.sql
    flat: yes
```

copy Copy a file to the server(s)

```
- copy:
    src: /laptop/yesterday.sql
    dest: /dumps/yesterday.sql

- name: Restore the DB from a backup
  mysql_db:
    name: mydb
    state: import
    target: /dumps/yesterday.sql
```