Deploying PHP applications
using Ansible, Ansible Vault and Ansistrano

Deploying ~~PHP~~ applications

using Ansible, Ansible Vault and Ansistrano

What we'll be looking at

- **Ansible** crash course

- Keeping secrets with **Ansible Vault**

- Deployments with **Ansistrano**

- Full Stack Software Developer & System Administrator

- **Senior Software Engineer** at **Inviqa**

- Acquia certified **Drupal 8 Grand Master** and **Cloud Pro**

- Open sourcer

- Drupal 7 & 8 **core contributor**

- @opdavies

- www.oliverdavies.uk

INVIQA

platform.sh

ACQUIA®

PANTHEON®

DigitalOcean

linode

VULTR

What is Ansible?

Ansible is an open-source **software provisioning, configuration management, and application-deployment** tool.

What is Ansible?

- CLI tool

- Written in Python

- Configured with YAML

- Executes ad-hoc remote commands

- Installs software packages

- Performs deployment steps

- Batteries included

What is Ansible?

- Hosts/Inventories

- Commands

- Playbooks

- Tasks

- Roles

ANSIBLE

Why Ansible?

- Familiar syntax

- Easily readable

- No server dependencies

- Easy to add to an existing project

- Includes relevant modules (e.g. Composer)

- Idempotency

Hosts / Inventories

```ini
# hosts.ini

[webservers]
192.168.33.10
```

```yaml
# hosts.yml

webservers:
  hosts:
    192.168.33.10:
```

Commands

```
ansible all -i hosts.yml -m ping
```

```
webservers | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

```
ansible all
-i hosts.yml
-m command
-a 'git pull
--chdir=/app'
```

```
ansible all -i hosts.yml
-m git -a 'repo=https://
github.com/opdavies/dransible
dest=/app'
```

Tasks and Playbooks

```yaml
# playbook.yml
---
- hosts: webservers # or 'all'

  vars:
    git_repo: https://github.com/opdavies/dransible

  tasks:
    - name: Update the code
      git:
        repo: '{{ git_repo }}'
        dest: /app
        version: master
        update: true
```

```
ansible-playbook
playbook.yml -i hosts.yml
```

Roles:
configuring a LAMP stack

```yaml
# requirements.yml
---
- src: geerlingguy.apache
- src: geerlingguy.composer
- src: geerlingguy.mysql
- src: geerlingguy.php
- src: geerlingguy.php-mysql
```

```
ansible-galaxy -r
requirements.yml install
```

```yaml
# provision.yml
---
- hosts: webservers

  roles:
    - geerlingguy.apache
    - geerlingguy.mysql
    - geerlingguy.php
    - geerlingguy.php-mysql
    - geerlingguy.composer
```

```yaml
# provision.yml
---
- hosts: webservers
  # ...

  vars:
    apache_vhosts:
      - servername: dransible.wip
        documentroot: /app/web
```

```yaml
# provision.yml
---

- hosts: webservers
  # ...

  vars:
    # ...
    php_version: '7.4'
    php_packages_extra:
      - libapache2-mod-php{{ php_version }}
      - libpcre3-dev
```

```yaml
# provision.yml
---

- hosts: webservers
  # ...

  vars:
    # ...
    mysql_databases:
      - name: main

    mysql_users:
      - name: user
        password: secret
        priv: main.*:ALL
```

```
ansible-playbook provision.yml
        -i hosts.yml
```

```
PLAY [Provision the webserver machines] ********************************************

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

TASK [geerlingguy.apache : Include OS-specific variables.] ***********************
ok: [webservers]

TASK [geerlingguy.apache : Include variables for Amazon Linux.]
skipping: [webservers]

TASK [geerlingguy.apache : Define apache_packages.] *****************************
ok: [webservers]

TASK [geerlingguy.apache : include_tasks] **************************************
included: /Users/opdavies/.ansible/roles/geerlingguy.apache/tasks/setup-Debian.yml for webservers

TASK [geerlingguy.apache : Update apt cache.] *********************************
changed: [webservers]
```

```
TASK [geerlingguy.composer : Ensure composer directory exists.] ***********************************
ok: [webservers]

TASK [geerlingguy.composer : include_tasks] ******************************************************
skipping: [webservers]

TASK [geerlingguy.composer : include_tasks] ******************************************************
skipping: [webservers]

RUNNING HANDLER [geerlingguy.apache : restart apache] ********************************************
changed: [webservers]

RUNNING HANDLER [geerlingguy.mysql : restart mysql] *********************************************
changed: [webservers]

RUNNING HANDLER [geerlingguy.php : restart webserver] *******************************************
changed: [webservers]

RUNNING HANDLER [geerlingguy.php : restart php-fpm] *********************************************
skipping: [webservers]

PLAY RECAP **************************************************************************************
webservers                 : ok=111  changed=32  unreachable=0   failed=0   skipped=78  rescued=0   ignored=0
```

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
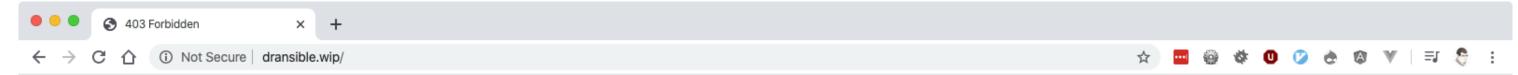
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

Forbidden

You don't have permission to access this resource.

Apache/2.4.18 (Ubuntu) Server at dransible.wip Port 80

Keeping secrets with Ansible Vault

```yaml
# provision.yml
---

- hosts: webservers
  # ...

  vars:
    # ...
    mysql_databases:
      - name: main

    mysql_users:
      - name: user
        password: secret
        priv: main.*:ALL
```

```yaml
# provision.yml
---

- hosts: webservers
  # ...

  vars:
    # ...
    mysql_databases:
      - name: main

    mysql_users:
      - name: user
        password: secret
        priv: main.*:ALL
```

```
ansible-vault create
vault.yml
```

```yaml
# vars/vault.yml
---
vault_database_name: main
vault_database_user: user
vault_database_password: secret
```

$ANSIBLE_VAULT;1.1;AES256
366562333235396163363938383961373439396232333933338666530313
730373233663263363306531336333353163643063663335396139363762
39383133330a35636566623262353733373066363838373932646161613461 3
163663762666464373733663737383039316163663626332353839353332
326634323466626134383330a38643539343232376138613733373636343
638646664301386662353933303936313864633132656538626338666666
353062643962306339393465323566653065646264313539366431353376
238343466353666376132356561656433613136663396530383263333064
333326431623539643166626234663736656337633030363238373331373
435333816536653162646264366266266316639306262666539373236
343662313265376261316636623963353933361336635373736343 5

```yaml
# vars/vars.yml
---
database_name: "{{ vault_database_name }}"
database_user: "{{ vault_database_user }}"
database_password: "{{ vault_database_password }}"
```

```yaml
# provision.yml
---

mysql_databases:
  - '{{ database_name }}'


mysql_users:
  - name: '{{ database_user }}'
    password: '{{ database_password }}'
    priv: '{{ database_name }}.*:ALL'
```

```
ansible-vault edit vault.yml
```
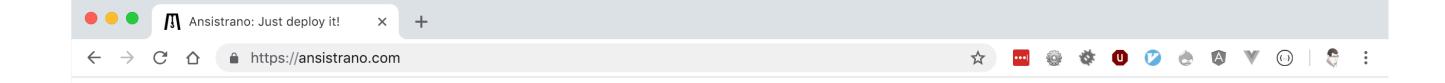
```
ansible-playbook
 -i hosts.yml
  deploy.yml
--ask-vault-pass
```

Basic deployment

```yaml
# deploy.yml

tasks:
  - name: Creating project directory
    file:
      path: /app
      state: directory

  - name: Uploading application
    synchronize:
      src: "{{ playbook_dir }}/../"
      dest: /app

  - name: Installing Composer dependencies
    composer:
      command: install
      working_dir: /app
```

Disadvantages

- Single point of failure

- No ability to roll back

- Sensitive data stored in plain text

Better deployments

Ansistrano

ansistrano.deploy and **ansistrano.rollback** are Ansible Galaxy roles to easily manage the deployment process for scripting applications such as PHP, Python and Ruby. It's an Ansible port for Capistrano.

Wow! 943303 deployments so far!

Today, teams using Ansistrano have already deployed **812** times. When considering the whole month, the amount of deployments is **22145** and this year we've reached the **22145** deployments.

Ouch! 2601 rollbacks so far!

On the other hand, today teams rollbacked **0** times. That's **18** times considering the whole month and **18** if considering the whole year. Rollback vs. Deployment ratio is **0.28%**, nice! Keep the good work.

Deployments by weekday

Monday (18%)

Tuesday (19%)

Features

- Multiple release directories

- Shared paths and files

- Customisable

- Multiple deployment strategies

- Multi-stage environments

- Prune old releases

- Rollbacks

```yaml
# requirements.yml
---
# ...
- ansistrano.deploy
- ansistrano.rollback
```

```yaml
# deploy.yml

---
- hosts: all

  roles:
    - ansistrano.deploy
```

```yaml
# deploy.yml
---
  # ...
  vars:
    project_deploy_dir: /app

    ansistrano_deploy_to: '{{ project_deploy_dir }}'
    ansistrano_deploy_via: git
    ansistrano_git_branch: master
    ansistrano_git_repo: 'git@github.com:opdavies/dransible'
```

```
ansible-playbook
-i hosts.yml
deploy.yml
```

```
PLAY [webservers] ********************************************************************

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

TASK [ansistrano.deploy : include_tasks] *********************************************

TASK [ansistrano.deploy : include_tasks] *********************************************
included: /Users/opdavies/.ansible/roles/ansistrano.deploy/tasks/setup.yml for webservers

TASK [ansistrano.deploy : ANSISTRANO | Ensure deployment base path exists] ***********
ok: [webservers]

TASK [ansistrano.deploy : ANSISTRANO | Ensure releases folder exists] ****************
ok: [webservers]

TASK [ansistrano.deploy : ANSISTRANO | Ensure shared elements folder exists] *********
ok: [webservers]

TASK [ansistrano.deploy : ANSISTRANO | Ensure shared paths exists] *******************
ok: [webservers] => (item=web/sites/default/files)
```

```
TASK [ansistrano.deploy : Update file permissions] ***********************************************
changed: [webservers]

TASK [ansistrano.deploy : include_tasks] *********************************************************

TASK [ansistrano.deploy : include_tasks] *********************************************************
included: /Users/opdavies/.ansible/roles/ansistrano.deploy/tasks/cleanup.yml for webservers

TASK [ansistrano.deploy : ANSISTRANO | Clean up releases] ****************************************
changed: [webservers]

TASK [ansistrano.deploy : include_tasks] *********************************************************

TASK [ansistrano.deploy : include_tasks] *********************************************************
included: /Users/opdavies/.ansible/roles/ansistrano.deploy/tasks/anon-stats.yml for webservers

TASK [ansistrano.deploy : ANSISTRANO | Send anonymous stats] *************************************
skipping: [webservers]

PLAY RECAP **************************************************************************************
webservers                 : ok=33   changed=14   unreachable=0   failed=0   skipped=7   rescued=0   ignored=0
```

```
vagrant@dransible:/app$ ls -l
total 8

lrwxrwxrwx 1   26 Jul 19 00:15 current -> ./releases/20190719001241Z
drwxr-xr-x 5 4096 Jul 22 20:30 releases
drwxr-xr-x 4 4096 Jul 19 00:00 shared
```
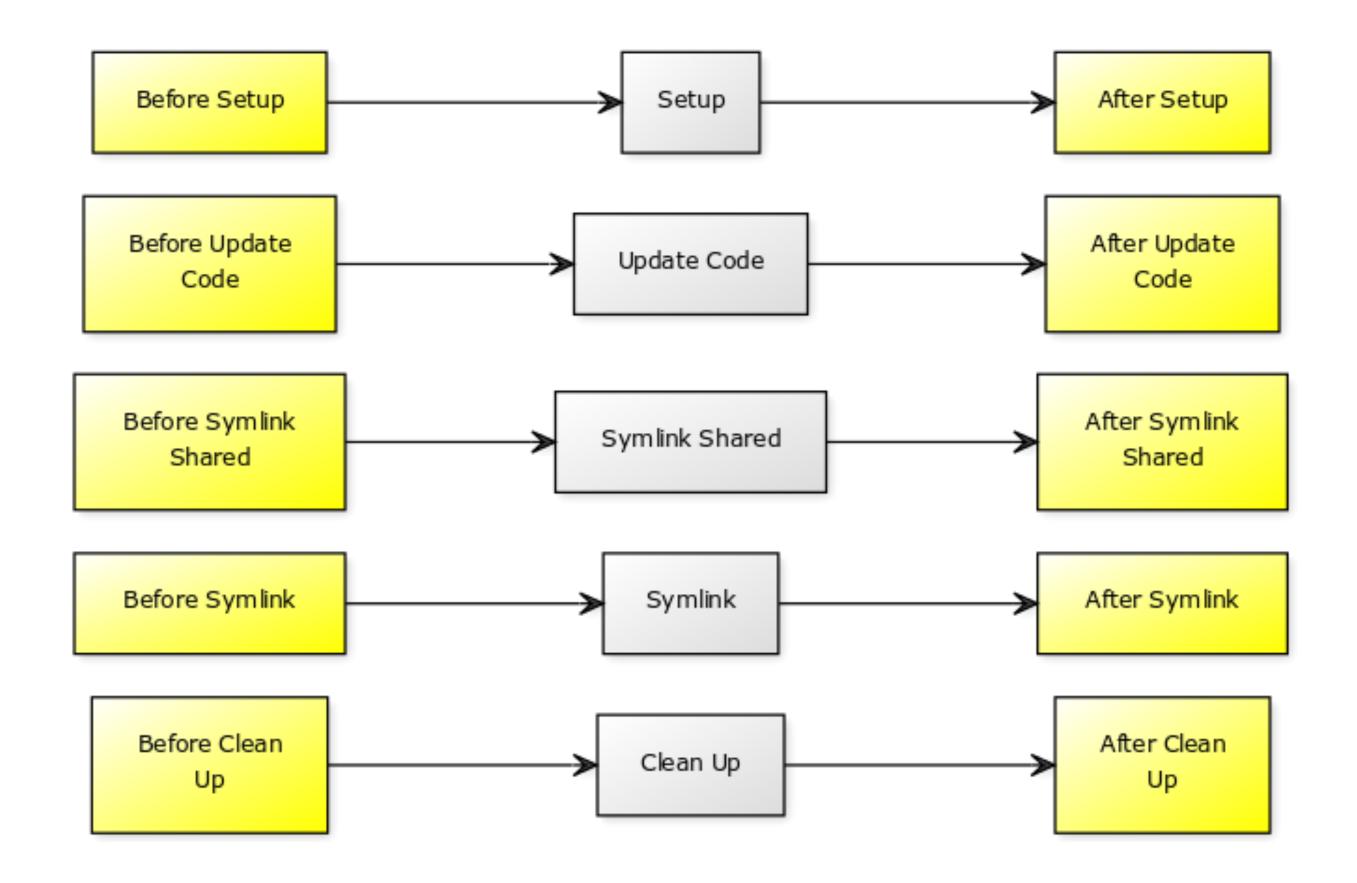
```
vagrant@dransible:/app/releases$ ls -l
total 20

drwxr-xr-x  5 4096 Jul 22 20:30 .
drwxr-xr-x  4 4096 Jul 19 00:15 ..
drwxr-xr-x 10 4096 Jul 19 00:02 20190719000013Z
drwxr-xr-x 10 4096 Jul 19 00:14 20190719001241Z
drwxr-xr-x  9 4096 Jul 22 20:30 20190722203038Z
```

```
ansible-playbook
-i hosts.yml
rollback.yml
```

```yaml
# rollback.yml
---
- hosts: all

  roles:
    - ansistrano.rollback

  vars:
    ansistrano_deploy_to: '{{ project_deploy_dir }}'
```

Customising Ansistrano:
Build Hooks

```mermaid
flowchart LR
    A[Before Setup] --> B[Setup] --> C[After Setup]
    D[Before Update Code] --> E[Update Code] --> F[After Update Code]
    G[Before Symlink Shared] --> H[Symlink Shared] --> I[After Symlink Shared]
    J[Before Symlink] --> K[Symlink] --> L[After Symlink]
    M[Before Clean Up] --> N[Clean Up] --> O[After Clean Up]
```

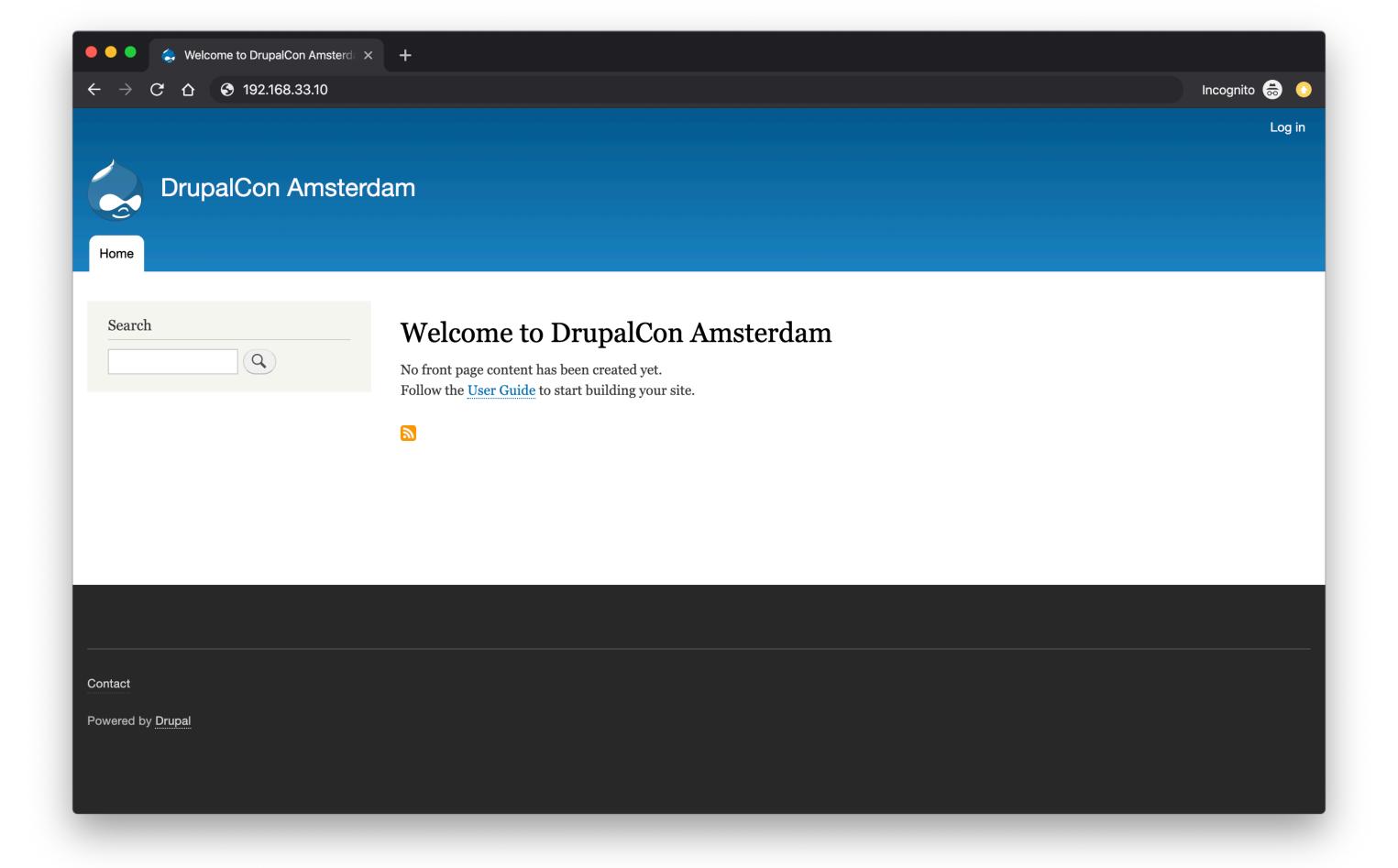|		
Before Setup	Setup	After Setup
Before Update Code	Update Code	After Update Code
Before Symlink Shared	Symlink Shared	After Symlink Shared
Before Symlink	Symlink	After Symlink
Before Clean Up	Clean Up	After Clean Up

```yaml
# deploy.yml
---
# ...

ansistrano_after_symlink_shared_tasks_file: '{{ playbook_dir }}/deploy/after-symlink-shared.yml'
ansistrano_after_symlink_tasks_file: '{{ playbook_dir }}/deploy/after-symlink.yml'
ansistrano_after_update_code_tasks_file: '{{ playbook_dir }}/deploy/after-update-code.yml'

release_web_path: '{{ ansistrano_release_path.stdout }}/web'
release_drush_path: '{{ ansistrano_release_path.stdout }}/vendor/bin/drush'
```

```yaml
# deploy/after-update-code.yml
---

- name: Install Composer dependencies
  composer:
    command: install
    working_dir: '{{ ansistrano_release_path.stdout }}'
```

```yaml
# deploy/after-symlink-shared.yml
---

- name: Run database updates
  command: '{{ release_drush_path }} --root {{ release_web_path }} updatedb'
```

```yaml
# deploy/after-symlink.yml
---

- name: Clear Drupal cache
  command: '{{ release_drush_path }} --root {{ release_web_path }} cache-rebuild'
```

DrupalCon Amsterdam

Home

Search

Welcome to DrupalCon Amsterdam

No front page content has been created yet.

Follow the User Guide to start building your site.

Contact

Powered by Drupal

Managing data across deployments

```
# deploy.yml

vars:
  # ...
  ansistrano_shared_paths:
    - "{{ drupal_root }}/sites/default/files"
```

```
vagrant@dransible:/app/shared/web/sites/default/files$ ls -la
total 28

drwxrwxrwx 6 4096 Jul 19 00:18 .
drwxr-xr-x 3 4096 Jul 19 00:00 ..
drwxrwxr-x 2 4096 Jul 22 21:24 css
-rwxrwxrwx 1  487 Jul 19 00:02 .htaccess
drwxrwxr-x 2 4096 Jul 19 00:19 js
drwxrwxrwx 3 4096 Jul 19 00:18 php
drwxrwxrwx 2 4096 Jul 19 00:03 styles
```

```
vagrant@dransible:/app/current/web/sites/default$ ls -la
total 48
dr-xr-xr-x 2  4096 Jul 19 00:14 .
drwxr-xr-x 3  4096 Jan 22 17:30 ..
-rw-r--r-- 1  6762 Jul 19 00:14 default.services.yml
-rw-r--r-- 1 31342 Jul 19 00:14 default.settings.php
lrwxrwxrwx 1    45 Jul 19 00:14 files -> ../../../../../shared/web/sites/default/files
-rw-r--r-- 1    35 Jul 19 00:12 settings.php
```

Generating settings files per deployment

```yaml
# vars/vault.yml
---
vault_database_name: main
vault_database_user: user
vault_database_password: secret
vault_hash_salt: dfgiy$fd2!34gsf2*34g74
```

```yaml
# vars/vars.yml
---
database_name: "{{ vault_database_name }}"
database_password: "{{ vault_database_password }}"
database_user: "{{ vault_database_user }}"
hash_salt: "{{ vault_hash_salt }}"
```

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

drupal_settings:
  - drupal_root: /app/web
    sites:
      - name: default
        settings:
          databases:
            default:
              default:
                driver: mysql
                host: localhost
                database: '{{ database_name }}'
                username: '{{ database_user }}'
                password: '{{ database_password }}'
          hash_salt: '{{ hash_salt }}'
          config_directories:
            sync: ../config/sync
```

```
// templates/settings.php.j2

// {{ ansible_managed }}

{% for key, values in item.1.settings.databases.items() %}
{% for target, values in values.items() %}
$databases['{{ key }}']['{{ target }}'] = array(
  'driver' => '{{ values.driver|default('mysql') }}',
  'host' => '{{ values.host|default('localhost') }}',
  'database' => '{{ values.database }}',
  'username' => '{{ values.username }}',
  'password' => '{{ values.password }}',
);

{% endfor %}
{% endfor %}

{% if item.1.settings.base_url is defined %}
$base_url = '{{ item.1.settings.base_url }}';
{% endif %}

{# ... #}
```

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

- name: Ensure directory exists
  file:
    state: directory
    path: '{{ item.0.drupal_root }}/sites/{{ item.1.name|default("default") }}'
  with_subelements:
    - '{{ drupal_settings }}'
    - sites
  no_log: true

- name: Create settings files
  template:
    src: settings.php.j2
    dest: '{{ item.0.drupal_root }}/sites/{{ item.1.name|default("default") }}/{{ item.1.filename|default("settings.php") }}'
  with_subelements:
    - '{{ drupal_settings }}'
    - sites
  no_log: true
```

Multiple environments

Dev, test, production

```yaml
mysql_databases:
  - name: production
  - name: staging

mysql_users:
  - name: production
    password: '{{ live_db_password }}'
    priv: '{{ live_db_name }}.*:ALL'

  - name: staging
    password: '{{ staging_db_password }}'
    priv: staging.*:ALL
```

```yaml
# inventories/live.yml
---
all:
  hosts:
    webservers:
      ansible_ssh_host: 192.168.33.10
      ansible_ssh_port: 22

      project_deploy_path: /app
      git_branch: master

      drupal_hash_salt: "{{ vault_drupal_hash_salt }}"
      drupal_install: true

      drupal_settings:
        # ...
```

```yaml
# inventories/staging.yml
---
all:
  hosts:
    webservers:
      ansible_ssh_host: 192.168.33.10
      ansible_ssh_port: 22

      project_deploy_path: /app-test
      git_branch: develop

      drupal_hash_salt: "{{ vault_drupal_hash_salt }}"
      drupal_install: true

      drupal_settings:
        # ...
```

```
ansible-playbook deploy.yml -i
inventories/staging.yml
```

```
ansible-playbook deploy.yml -i
inventories/live.yml
```

Questions?