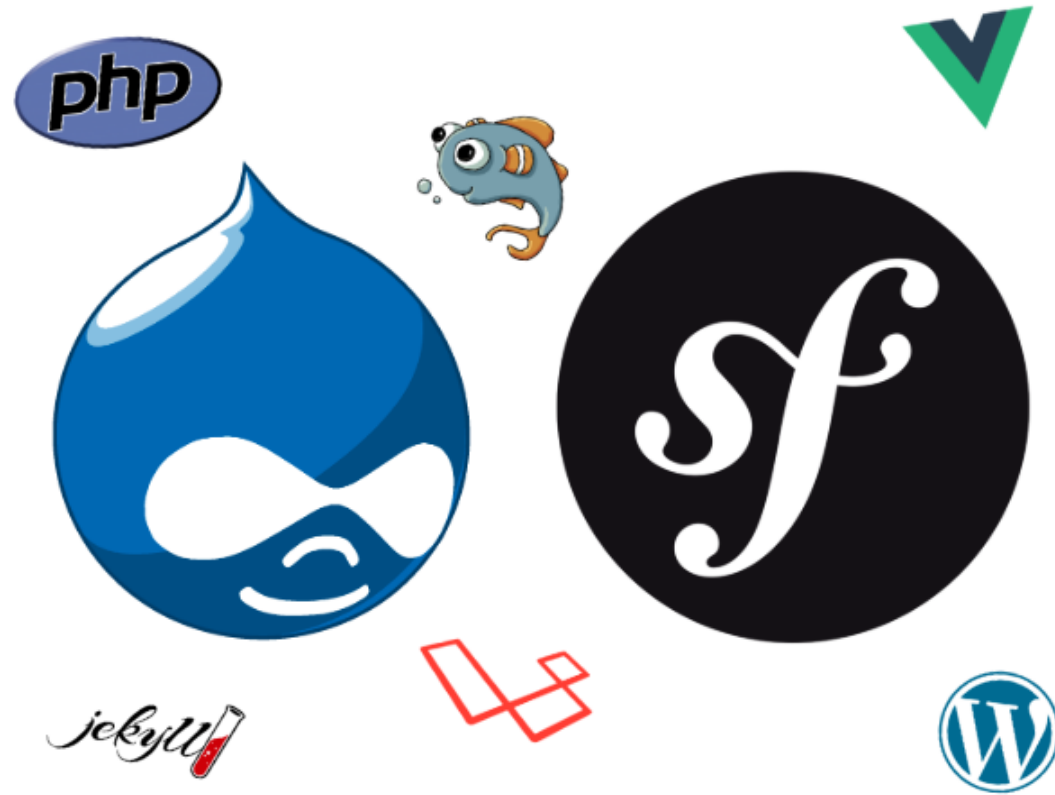


Deploying PHP with Ansible, Ansible Vault, and Ansistrano

Oliver Davies, Inviqa





Configuring a LAMP stack

requirements.yml

- src: geerlingguy.apache
- src: geerlingguy.composer
- src: geerlingguy.mysql
- src: geerlingguy.php
- src: geerlingguy.php-mysql



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

```
# playbook.yml
```

```
---
```

```
- hosts: webservers
```

```
  roles:
```

- geerlingguy.apache
- geerlingguy.mysql
- geerlingguy.php
- geerlingguy.php-mysql
- geerlingguy.composer



```
# playbook.yml
```

```
---
```

```
vars:
```

```
  apache_vhosts:
```

- servername: dransible
- documentroot: /app/web



```
# playbook.yml
```

```
---
```

```
vars:
```

```
  php_version: 7.4
```

```
  php_packages_extra:
```

- libapache2-mod-php{{ php_version }}
- libpcres-dev




```
# playbook.yml
```

```
---
```

```
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 - Mozilla Firefox (Private Browsing)

Apache2 Ubuntu Default Page: 403 Forbidden

192.168.33.10



Apache2 Ubuntu Default Page

ubuntu

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.

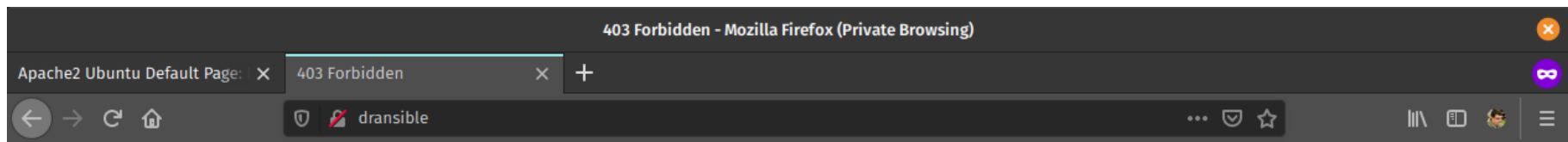
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

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
```



Forbidden

You don't have permission to access this resource.

Apache/2.4.18 (Ubuntu) Server at dransible Port 80



Basic deployment

```
# deploy.yml
```

```
---
```

```
tasks:
```

- name: Creating project directory

```
  file:
```

```
    path: /app
```

```
    state: directory
```

- name: Uploading application

```
  synchronize:
```

```
    src: '{{ playbook_dir }}/../'
```

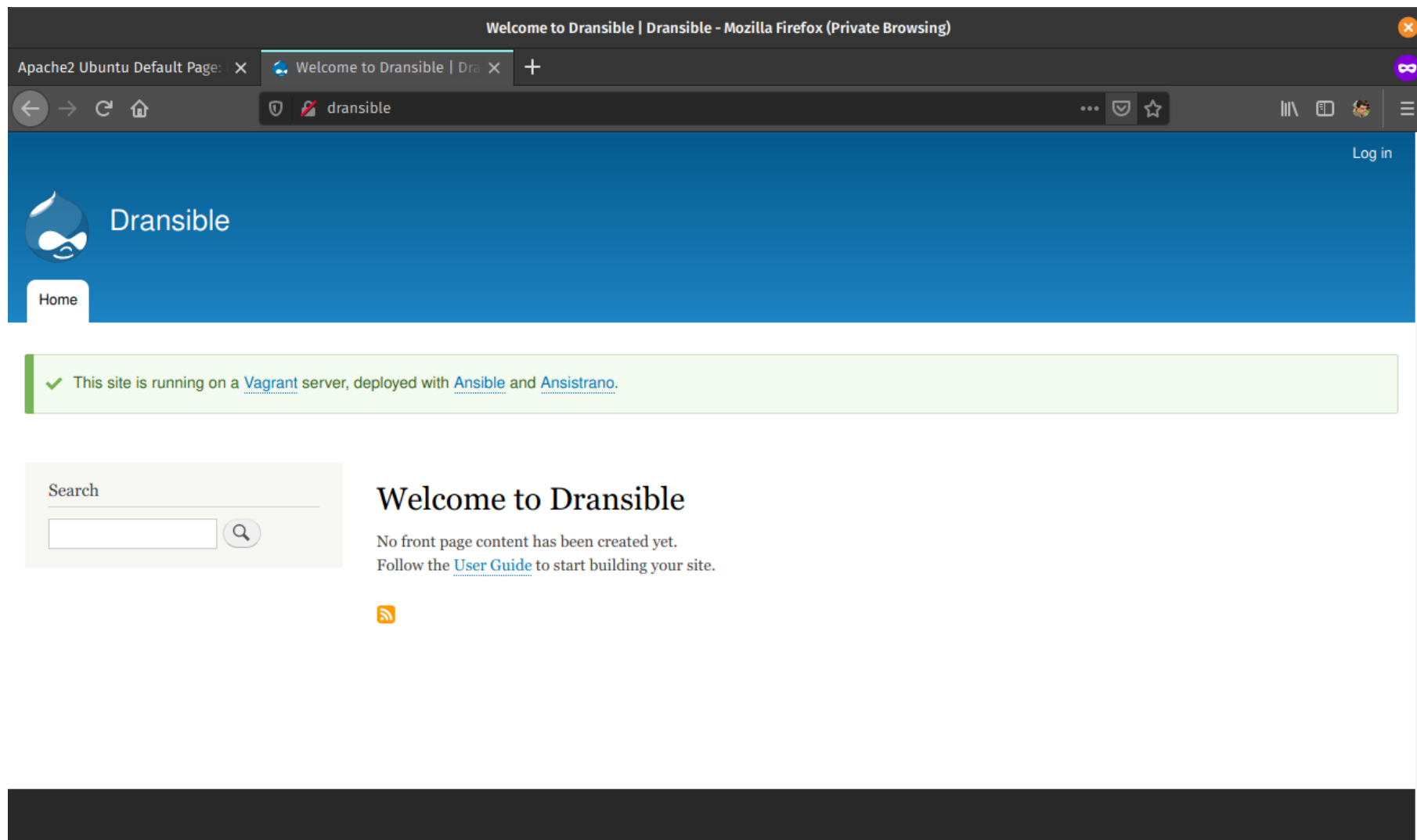
```
    dest: /app
```


- name: Installing Composer dependencies

```
composer:  
  command: install  
  working_dir: /app
```



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



Disadvantages

- Sensitive data stored in plain text
- Single point of failure
- No ability to roll back

Keeping secrets with Ansible Vault

vars:

mysql_databases:

- name: main

mysql_users:

- name: user

password: secret

priv: main.*:ALL

```
# provision_vault.yml
```

```
---
```

```
vault_database_name: main
```

```
vault_database_user: user
```

```
vault_database_password: secret
```

```
ansible-vault encrypt  
provision_vault.yml
```


New Vault password:
Confirm New Vault password:
Encryption successful

\$ANSIBLE_VAULT;1.1;AES256

63656632326165643137646334343537396533656565313032363262623962393861666438393539
6366336638316133373061306332303761383565343035330a373637373830356430353630356161
32313831663039343733343539636365386333303862363635323138346137666166356639323338
3264636538356634390a343766353661386666376362376439386630363664616166643364366335
62373530393933373830306338386539626565313364643133666131613138383431353638636334
39376437633462373934313236363662633832643138386433646230313465383337373031373137
61353963623364393134386335373731356337366464633531656435383161656435313530363234
37373865393839616534353165656463313961333532363537383263343364646534333032336337
3235

```
# provision_vars.yml
```

```
---
```

```
database_name: '{{ vault_database_name }}'
```

```
database_user: '{{ vault_database_user }}'
```

```
database_password: '{{ vault_database_password }}'
```

```
# provision.yml
```

```
---
```

```
vars_files:
```

- vars/provision_vault.yml
- vars/provision_vars.yml

```
vars:
```

```
mysql_databases:
```

- '{{ database_name }}

```
mysql_users:
```

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

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

```
ansible-playbook deploy.yml  
-i hosts.yml --vault-password-f  
ile secret.txt
```

Better deployments with 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!

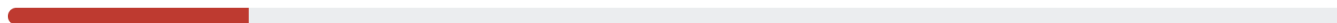
Today, teams using Ansistrano have already deployed **812** times. When considering the whole month, the amount of deployments is **22145** and this year we've reached the **22145** deployments.

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 (100%)



Features

- Multiple release directories
- Shared paths and files
- Customisable
- Multiple deployment strategies
- Multi-stage environments
- Prune old releases
- Rollbacks

```
# requirements.yml
```

```
---
```

- src: ansistrano.deploy
- src: ansistrano.rollback



```
# deploy.yml
```

```
---
```

```
- hosts: all
```

```
  roles:
```

```
    - ansistrano.deploy
```



```
# 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'
```

```
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
```



```
# rollback.yml
```

```
---
```

```
- hosts: all
```

```
roles:
```

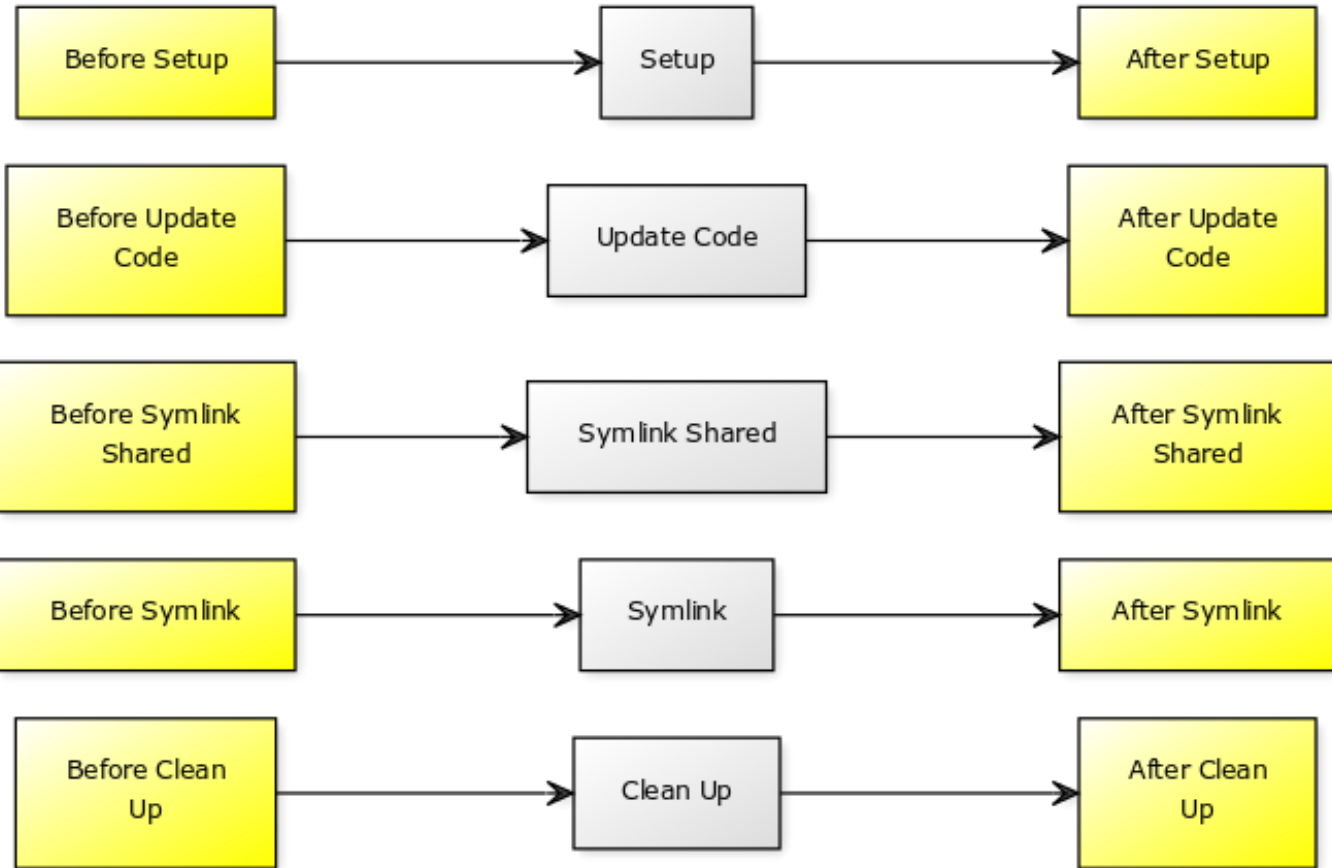
```
  - ansistrano.rollback
```

```
vars:
```

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

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

Customising Ansistrano: Build Hooks



```
# deploy.yml
```

```
---
```

```
vars:
```

```
  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 }}/bin/drush'
```

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

```
---
```

```
- name: Install Composer dependencies
```

```
  composer:
```

```
    command: install
```

```
    working_dir: '{{ ansistrano_release_path.stdout }}'
```

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

```
---
```

- **name:** Run database updates
command: >
 {{ release_drush_path }}
 --root {{ release_web_path }}
 updatedb

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

```
---
```

- name: Rebuild Drupal cache
command: >
 {{ release_drush_path }}
 --root {{ release_web_path }}
 cache-rebuild

Demo

Generating settings files per deployment

```
# deploy_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 #}

{% 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 %}
```

```
# 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

development, test, production

```
# vars.yml
```

```
---
```

```
vars:
```

```
  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


```
# hosts.yml
```

```
---
```

```
production:
```

```
  children:
```

```
    hosts:
```

```
      webservers:
```

```
        ansible_ssh_host: 192.168.33.10
```

```
        ansible_ssh_port: 22
```

```
        project_deploy_path: /app
```

```
        git_branch: production
```

```
        drupal_hash_salt: '{{ vault_drupal_hash_salt }}'
```

```
        drupal_install: false
```

```
        drupal_settings:
```

```
          # ...
```

```
# hosts.yml
```

```
---
```

```
staging:
```

```
  children:
```

```
    hosts:
```

```
      webservers:
```

```
        ansible_ssh_host: 192.168.33.10
```

```
        ansible_ssh_port: 22
```

```
        project_deploy_path: /app-staging
```

```
        git_branch: staging
```

```
        drupal_hash_salt: '{{ vault_drupal_hash_salt }}'
```

```
        drupal_install: true
```

```
        drupal_settings:
```

```
          # ...
```

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

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

Thanks!

References:

- <https://oliverdavies.link/ansible-repos>
- <https://docs.ansible.com>
- <https://www.ansistrano.com>
- <https://symfonycasts.com/screencast/ansistrano>

Me:

- <https://www.oliverdavies.uk>