

DEPLOYING DRUPAL USING CAPISTRANO

Jochen Verdeyen

@jochenverdeyen

SITUATIONS



TST



STAG



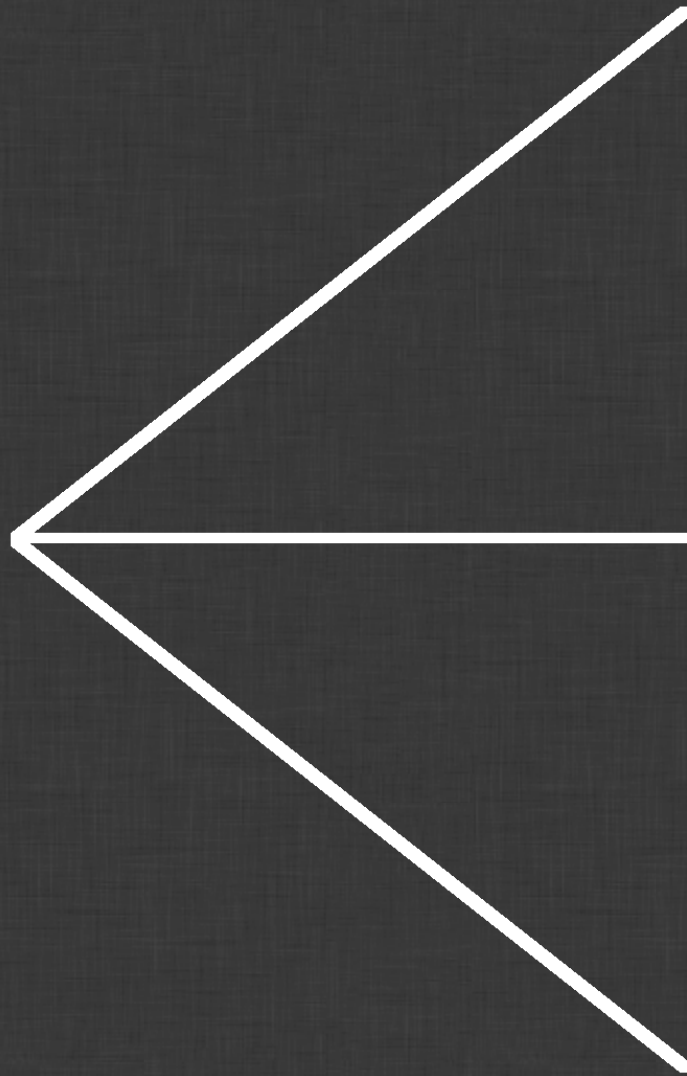
PROD



LOCAL



LOCAL



TST



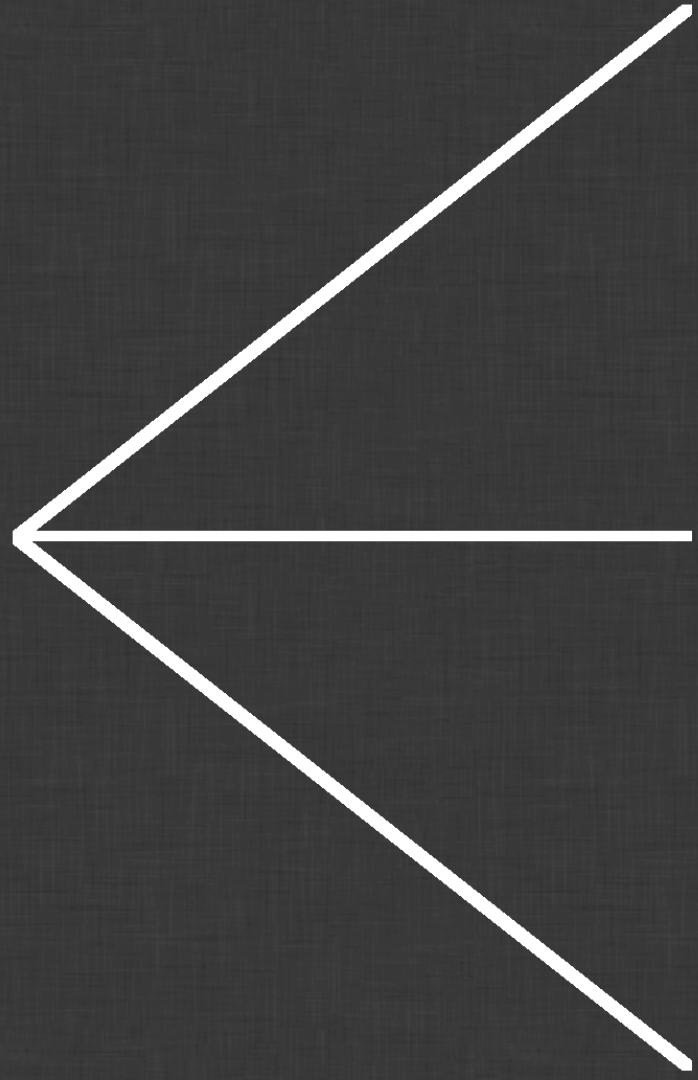
STAG



PROD



JENKINS



TST



STAG



PROD

PREREQUISITES

- SSH
- Ruby (local)
- Git (servers)

INSTALLATION

Bundler - Gemfile

```
source 'https://rubygems.org'  
  
group :deploy do  
  gem 'capistrano', '~> 3.4.0'  
end
```

```
bundle install
```


Default setup

```
bundle exec cap install
```

```
├─ Capfile
├─ config
│   ├── deploy
│   │   ├── production.rb
│   │   └── staging.rb
│   └── deploy.rb
├─ lib
│   └── capistrano
│       └── tasks
```

Custom setup

```
bundle exec cap install STAGES=tst,stag,prd
```

```
├─ Capfile
├─ config
│   ├── deploy
│   │   ├── prd.rb
│   │   ├── stag.rb
│   │   └─ tst.rb
│   └─ deploy.rb
├─ lib
│   └─ capistrano
│       └─ tasks
```

CONFIGURATION

Capfile

```
# Load DSL and set up stages
require 'capistrano/setup'

# Include default deployment tasks
require 'capistrano/deploy'
```

deploy.rb

```
set :application, 'my_app'  
set :repo_url, 'git@my-repo.url'  
  
# Branch to deploy  
set :branch, 'master'  
  
# Destination path of the application  
set :deploy_to, '/var/www/my_app'  
  
# Amount of releases to keep  
set :keep_releases, 5  
  
# Default value for linked_files is []  
set :linked_files, []  
  
# Default value for linked_dirs is []
```


deploy/tst.rb

```
# Server  
# =====  
server 'xxx.xxx.xxx.xxx', roles: %w{app db}, user: 'deploy_user'
```

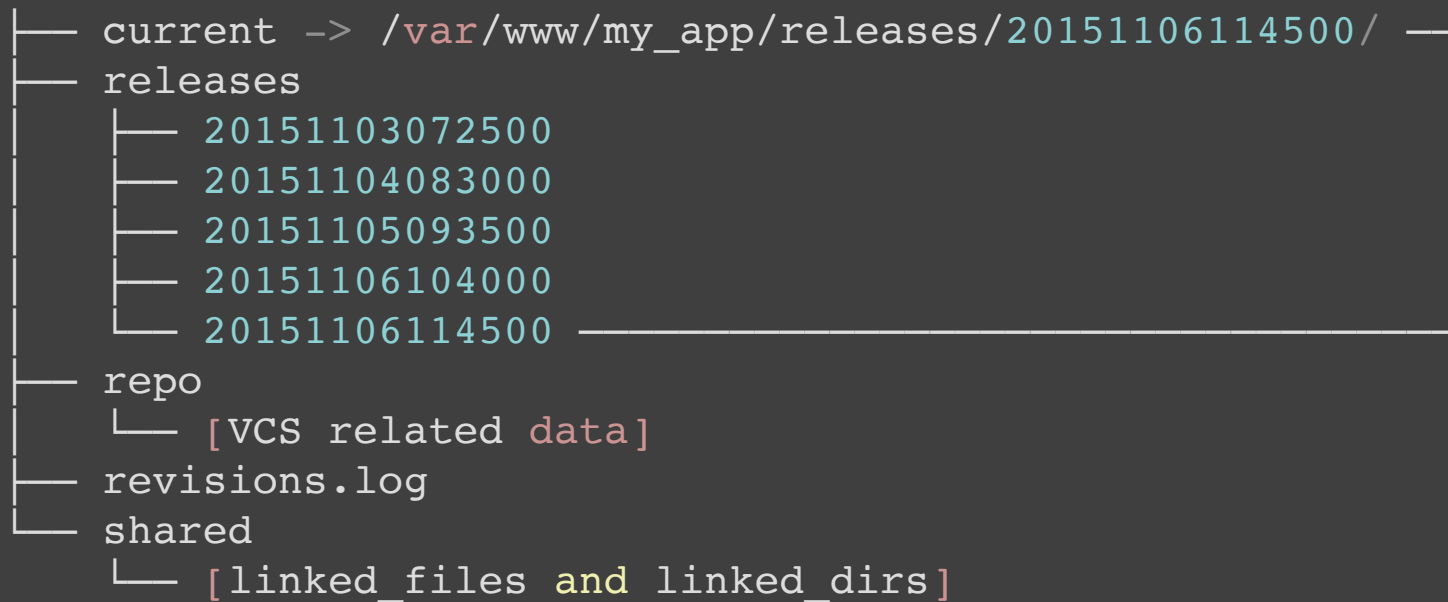
DEPLOYING

Deploy flow

```
bundle exec cap prd deploy
```

```
deploy:starting - start a deployment  
deploy:started - started hook  
deploy:updating - update server(s) with a new release  
deploy:updated - updated hook  
deploy:publishing - publish the new release  
deploy:published - published hook  
deploy:finishing - finish the deployment, clean up  
deploy:finished - finished hook
```

Deploy flow - Structure



Rollback flow

```
bundle exec cap prd deploy:rollback
```

```
deploy:starting
```

```
deploy:started
```

```
deploy:reverting - revert server(s) to previous release
```

```
deploy:reverted - reverted hook
```

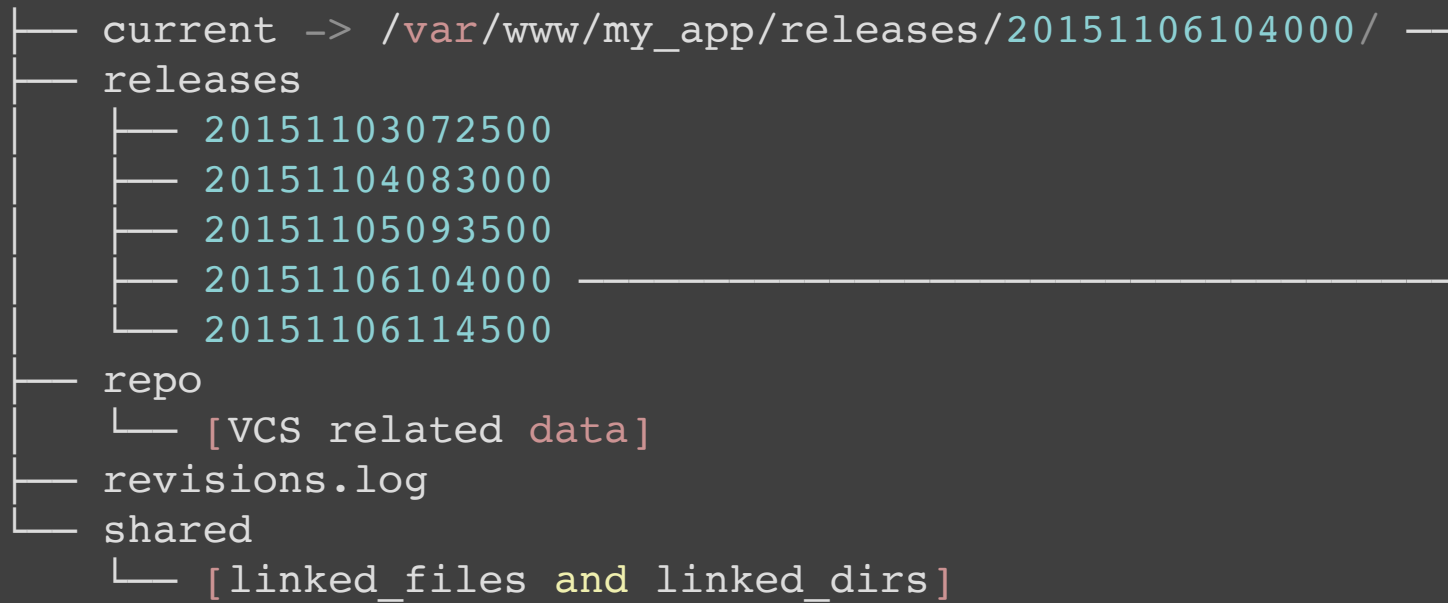
```
deploy:publishing
```

```
deploy:published
```

```
deploy:finishing_rollback - finish the rollback, clean up
```

```
deploy:finished
```


Rollback flow - Structure



DRUPAL 8

Prerequisites

- Drush
- Composer

Capfile

```
# Composer to install drush on the server  
require 'capistrano/composer'
```

deploy.rb

```
set :application, 'my_app'
set :repo_url, 'git@my-repo.url'

# Branch to deploy
set :branch, 'master'

# Destination path of the application
set :deploy_to, '/var/www/my_app'

# Link files directory
set :linked_dirs, fetch(:linked_dirs, []).push(
  "#{fetch(:app_path)}/sites/default/files"
)
```


deploy/tst.rb

```
# Server
# =====
server 'xxx.xxx.xxx.xxx', roles: %w{app db}, user: 'deploy_user'

# Map composer and drush commands
# =====
SSHKit.config.command_map[:composer] =
  "#{shared_path.join("composer.phar")}"

SSHKit.config.command_map[:drush] =
  "#{shared_path.join("vendor/bin/drush")}"
```

Drupal specific tasks

```
desc 'Create a database backup'
task :backup_db do
  on roles(:app) do
    within current_path.join(fetch(:app_path)).join('sites/default')
      execute :drush, "sql-dump --result-file=#{current_path}/backup_
    end
  end
end

desc 'Set the site offline'
task :site_offline do
  on roles(:app) do
    within release_path.join(fetch(:app_path)).join('sites/default')
      execute :drush, "state-set system.maintenance_mode 1 -y"
    end
  end
end
```

Drupal specific tasks

```
desc 'Import configuration from the config directory'
task :config_import do
  on roles(:app) do
    within release_path.join(fetch(:app_path)).join('sites/default')
      execute :drush, "config-import -y"
    end
  end
end

desc 'Clear all caches'
task :cache_clear do
  on roles(:app) do
    within release_path.join(fetch(:app_path)).join('sites/default')
      execute :drush, "cache-rebuild all"
    end
  end
end
```

Deploy flow

```
bundle exec cap prd deploy:drupal
```

```
before "deploy:starting", "drupal:backup_db"  
deploy:starting      - start a deployment  
deploy:started       - started hook  
deploy:updating      - update server(s) with a new release  
deploy:updated       - updated hook  
    after "drupal:updated",      "drupal:site_offline"  
    after "drupal:site_offline", "drupal:update_db"  
    after "drupal:update_db",    "drupal:config_import"  
    after "drupal:config_import", "drupal:cache_clear:all"  
    before "deploy:publishing",  "drupal:site_online"  
deploy:publishing    - publish the new release  
deploy:published     - published hook  
deploy:finishing     - finish the deployment, clean up  
deploy:finished     - finished hook
```

JENKINS

```
`${WORKSPACE}/deploy/deploy.sh -w `${WORKSPACE} -e tst
```

deploy.sh

```
# Go to the Capistrano folder in the workspace
cd `${WORKSPACE}/deploy/capistrano

# Prepare bundle
bundle
bundle install

# Capistrano preparation
bundle exec cap `${ENVIRONMENT} composer:install_executable
bundle exec cap `${ENVIRONMENT} drush:install

# Capistrano deploy
bundle exec cap `${ENVIRONMENT} deploy:drupal

exit 0
```

DEMO