

# Using **Capifony** for **Symfony** apps deployment

Žilvinas Kuusas



# Who am I?

**Žilvinas Kuusas**

lead developer

Estina / iSign.io

t: @kuusas

e: zilvinas@kuusas.lt



# How do people deploy?

## Code

- FTP
- vim
- rsync
- git pull
- ...

## Database

- Manually manage DB fields
- phpMyAdmin
- mysql < changes.sql
- ...

# Symfony - heavy

Post-deployment tasks:

- `cache:clear`
- `cache:warmup`
- `assets:install`
- `assetic:dump`
- `doctrine:migrations:migrate`
- restart some services
- ...

# What if ...



... you executed a wrong command?

**Don't Repeat Yourself**

**AUTOMATE!**

# Goals for **Solid** deployment

- One-click deployment
- Deploy in one transaction
- Rollback
- Automated tasks
- Notifications
- 0 downtime
- Confidence

# Capistrano

Capifony



# What Capistrano is?

- Capistrano is an open source tool for running scripts on multiple servers
- Primary use - application deployment
- Capifony - set of instructions called “recipes” for Symfony applications deployment
- Built to make your job a lot easier

# Why Capifony?

- **Fully supports Symfony features**
- Actively maintained since 2009
- 6 active contributors
- >100 total
- last commit 16 days ago

# What do we need?

- Symfony2 application
- Capifony
- SSH access to server (ssh-key recommended)

/var/www/vilniusphp.lt/

└─ current -> /var/www/vilniusphp.lt/releases/20150204203619

└─ releases

| └─ 20150204201220

| └─ 20150204203553

| └─ 20150204203619

└─ shared

└─ app

| └─ config

| └─ logs

└─ web

└─ uploads

## How it works?

# Setup

```
$ sudo gem install capifony
```

```
$ cd /path/to/your/project
```

```
$ capifony .
```

→ `symfony git:(master) capifony .`

`[add] writing './Capfile'`

`[add] writing './app/config/deploy.rb'`

`[done] symfony 2 project capifonied!`

`$ capifony .`

```
1  # Default options
2  set :application, "Symfony deployment example"
3  set :domain,      "vilniusphp.lt"
4  set :deploy_to,   "/var/www/#{domain}"
5  set :app_path,    "app"
6
7  set :repository,
8  • "git@github.com:kuusas/symfony-capifony-example.git"
9
10 set :scm,          :git
11
12 set :model_manager, "doctrine"
13
14 role :web,          domain
15 role :app,           domain, :primary => true
16
17 set :keep_releases, 3
```

```
$ vi app/config/deploy.rb
```

```
16  # Symfony options
17  set :symfony_env_prod, "prod"
18  set :shared_files,     ["app/config/parameters.yml"]
19
20  # Versioning options
21  set :branch,           "master"
22
23  # SSH options
24  set :use_sudo,          false
25  set :user,              "elf"
26  set :deploy_user,       "elf"
27  ssh_options[:forward_agent] = true
28  ssh_options[:port]         = "22"
```

```
$ vi app/config/deploy.rb
```



→ symfony git:(master) ✗ cap deploy

--> Updating code base with checkout strategy

--> Creating cache directory.....✓

--> Creating symlinks for shared directories.....✓

--> Normalizing asset timestamps.....✓

--> Downloading Composer.....✓

--> Installing Composer dependencies.....✓

--> Updating parameters.....

database\_driver (pdo\_mysql):

\$ cap deploy

→ symfony git:(master) ✗ cap deploy

--> Updating code base with checkout strategy

--> Creating cache directory.....✓

--> Creating symlinks for shared directories.....✓

--> Creating symlinks for shared files.....✓

--> Normalizing asset timestamps.....✓

--> Downloading Composer.....✓

--> Installing Composer dependencies.....✓

--> Warming up cache.....✓

--> Clear controllers.....✓

--> Successfully deployed!

\$ cap deploy

# Main deployment commands

```
$ cap deploy
```

```
...
```

```
$ cap deploy:migrations
```

```
...
```

```
$ cap deploy:rollback
```

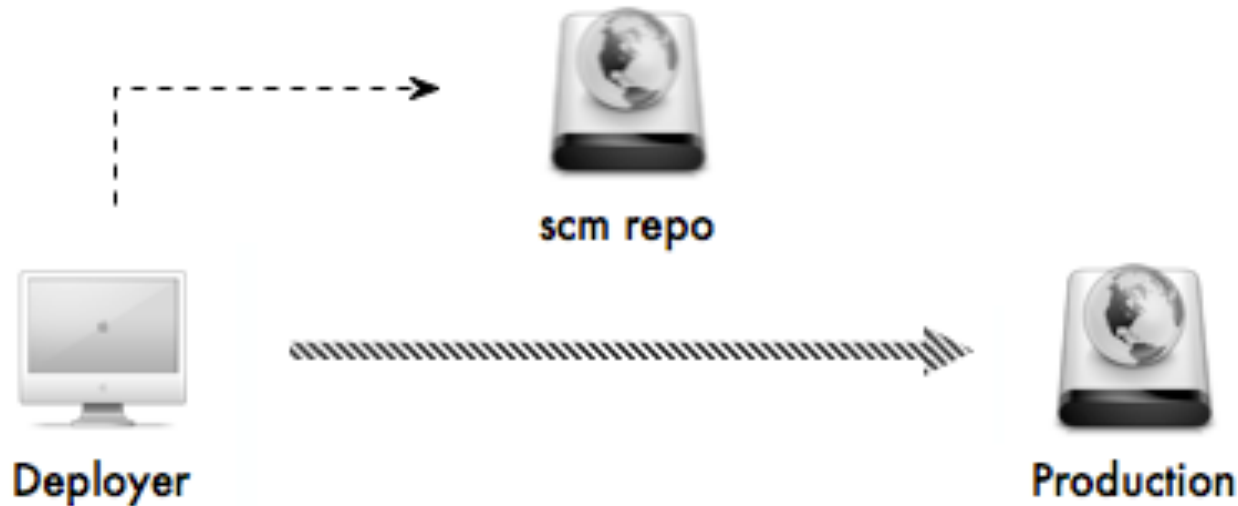
# Deployment strategies

deployment > scm > production

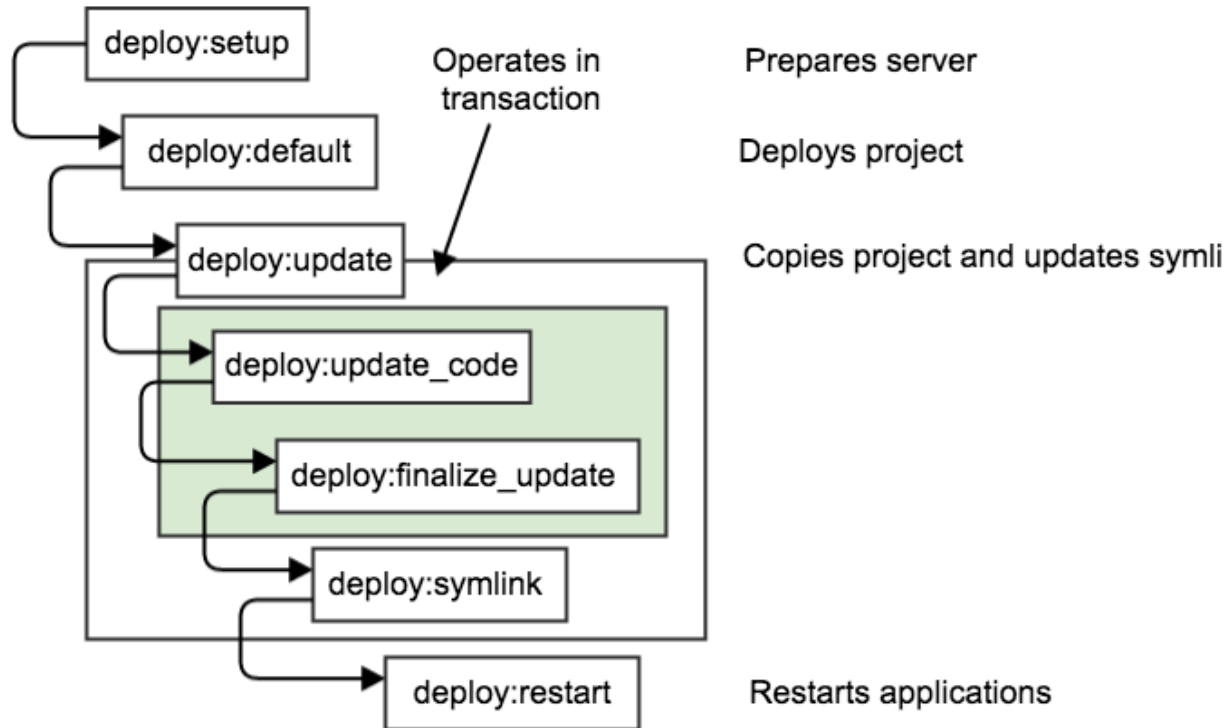


# Deployment strategies

scm > deployment > production



# Behind scenes



# What else **Capifony** can do?

- Execute Symfony commands on server
- Execute shell commands on server
- Dump remote database and download
- Tail logs
- Turn on/off maintenance page

```
$ cap deploy:check
```

```
$ cap database:copy:to_local
```

```
$ cap deploy:cold
```

```
$ cap deploy:pending
```

```
$ cap deploy:web:disable
```

```
$ cap deploy:web:enable
```

```
$ cap symfony:logs:tail
```

```
$ cap symfony
```

```
...
```

```
$ cap -vT # Lists all tasks available
```



# Database migrations

Deploy database schema updates in...

- SAFE
- EASY
- STANDARDIZED

way...

```
1  {
2    "require": {
3      "doctrine/migrations": "1.0.*@dev",
4      "doctrine/doctrine-migrations-bundle": "2.1.*@dev"
5    }
6  }
```

Available commands for the "doctrine:migrations" namespace:

<code>doctrine:migrations:diff</code>	Generate a migration by comparing your current database to your mapping information.
<code>doctrine:migrations:execute</code>	Execute a single migration version up or down manually.
<code>doctrine:migrations:generate</code>	Generate a blank migration class.
<code>doctrine:migrations:latest</code>	Outputs the latest version number
<code>doctrine:migrations:migrate</code>	Execute a migration to a specified version or the latest available version.
<code>doctrine:migrations:status</code>	View the status of a set of migrations.
<code>doctrine:migrations:version</code>	Manually add and delete migration versions from the version table.

## DoctrineMigrationsBundle

```

11 class Version20150205170707 extends AbstractMigration
12 {
13     public function up(Schema $schema)
14     {
15         // this up() migration is auto-generated, please modify it to your needs
16         $this->abortIf($this->connection->getDatabasePlatform()->getName() != 'mysql',
17             • 'Migration can only be executed safely on \'mysql\'');
18
19         $this->addSql('DROP TABLE Participant');
20         $this->addSql('ALTER TABLE Event ADD place VARCHAR(255) NOT NULL');
21     }
22
23     public function down(Schema $schema)
24     {
25         // this down() migration is auto-generated, please modify it to your needs
26         $this->abortIf($this->connection->getDatabasePlatform()->getName() != 'mysql',
27             • 'Migration can only be executed safely on \'mysql\'');
28
29         $this->addSql('CREATE TABLE Participant (id INT AUTO_INCREMENT NOT NULL,
30             • event_id INT DEFAULT NULL, name VARCHAR(255) DEFAULT NULL, email VARCHAR(255) NOT NULL,
31             • age INT NOT NULL, registrationDate DATETIME NOT NULL, last_name VARCHAR(255) NOT NULL,
32             • INDEX IDX_D79F6B1171F7E88B (event_id), PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8
33             • COLLATE utf8_unicode_ci ENGINE = InnoDB');
34         $this->addSql('ALTER TABLE Participant ADD CONSTRAINT FK_D79F6B1171F7E88B
35             • FOREIGN KEY (event_id) REFERENCES event (id)');
36         $this->addSql('ALTER TABLE Event DROP place');
37     }
38 }

```

\$ app/console doctrine:migrations:diff

Deploying with migrations:

```
$ cap deploy:migrations
```

Rolling-back migrations:

```
$ cap symfony -S task_arguments="doctrine:  
migrations:migrate 20150205170707 --no-interaction"
```

```
$ cap deploy:migrations
```

# OPcache

- Create temp file accessible via www
- Make curl request
- Remove file

Capistrano recipe for that:

<https://github.com/occtech/capistrano-recipes/blob/master/Opcache-Recipe.rb>

# Keeping sessions

```
1  # app/config/config.yml
2  framework:
3    session:
4      save_path: "%kernel.root_dir%/sessions/"
5
6  # app/config/deploy.rb
7  set :shared_children, [..., app_path + "/sessions"]
```

# Multistage extension

Enables deployment to multiple servers

```
$ cap STAGE_NAME deploy
```

Multistage extension:

<https://github.com/capistrano/capistrano/wiki/2.x-Multistage-Extension>

# Deploying release package

- CI server builds release package (zip)
- Capifony deploys

Capistrano deploy strategy called “archive”:

<https://github.com/sorccu/capistrano-deploy-strategy-archive>



# Symfony related configuration

```
1  # Custom options
2  set  :shared_files,      ["app/config/parameters.yml"]
3  set  :writable_dirs,     ["app/cache", "app/logs", "uploads"]
4  set  :shared_children,   [app_path + "/logs", "uploads"]
5  set  :use_composer,      true
6  set  :update_vendors,    false
7  set  :copy_vendors,      true
8  set  :composer_options,  "--no-dev --verbose --optimize-autoloader"
9  set  :cache_warmup,      true
10 set  :dump_assetic_assets, false
11 set  :interactive_mode,   false
```

# Extending Capistrano

It's Ruby!



```
1 namespace :deploy do
2   task :start do
3     ...
4   end
5
6   task :stop do
7     ...
8   end
9
10  task :restart, :roles => :app, :except => {
11    • :no_release => true } do
12    run "sudo /etc/init.d/nginx reload"
13  end
14 end
```

Override Capistrano recipes

```
1  # Update database schema after deployment
2  after "deploy", "symfony:doctrine:schema:update"
3
4  # Execute composer dump-autoload
5  after "deploy:rollback:code" do
6      set :composer_bin, "#{php_bin} composer.phar"
7      # Run composer from current release dir
8      run "#{try_sudo} sh -c 'cd #{current_path} &&
•  SYMFONY_ENV=#{symfony_env_prod} #{composer_bin}
•  dump-autoload #{composer_dump_autoload_options}'"
9  end
10
```

Listen to events

# Debugging

```
1  # app/config/deploy.rb  
2  logger.level = Logger::MAX_LEVEL
```

→ `symfony git:(master) ✗ cap deploy -d`

--> Updating code base with checkout strategy

Preparing to execute command: "git clone -q -b master git@github.com:kuusas/symfony-capifony-example.git /var/www/vilniusphp.lt/releases/20150205153314 && cd /var/www/vilniusphp.lt/releases/20150205153314 && git checkout -q -b deploy adf4f4c5a2eb65ccb0789a61b08a24175fb23112 && (echo adf4f4c5a2eb65ccb0789a61b08a24175fb23112 > /var/www/vilniusphp.lt/releases/20150205153314/REVISION)"

Execute ([Yes], No, Abort) ? |y| \_

`$ cap deploy -d`

# Arguments in CLI

`-S, --set-before NAME=VALUE`      Set a variable before the recipes are loaded.

`-s, --set NAME=VALUE`              Set a variable after the recipes are loaded.

`cap deploy:migrations -s branch=master`

`cap deploy:migrations -S my_var=value`

# Other deployment tools

- Fabric (Python)
- Idephix
- Magellanes
- Deployer
- Laravel / Envoy
- Rocketeer
- **Ansible + Tower**



# Final thoughts

Don't use FTP

# Final thoughts

AUTOMATE

# Questions?



AČIŮ

Credits for image authors. "what if..." gif taken from [devopsreactions.tumblr.com](https://devopsreactions.tumblr.com), deployment strategies - from [capifony.org](https://capifony.org).  
Couldn't find the real authors of lizzard and kitty - so if its your's - THANK YOU.