

# Capistrano 2

Beyond Hassle-Free Deployment

Luca Mearelli

Pisa - Oct. 27, 2007



**Rails is fun**

Rails is fun

Rails is **TOO MUCH** fun

**Deployment**

# System Administration

**Rails : Web development**

**=**

**Capistrano : Sys.administration**

# Capistrano basics

- **SSH** to the remote machines
- **Posix** compatible shell
- Same password or **public key**
- a bit of **ruby** knowledge
- works from a **command line**

# Capistrano basics

- **Capfile** (makefile, rakefile, ...)
- It's ruby

```
task :dump_db, :hosts => "database.spazidigitali.com" do
  run "mysqldump -u deploy sd_production > /home/luca/dumps/db.sql"
end
```

```
cap dump_db
```

```
role :sd_db, "database.spazidigitali.com"

task :dump_db do
  run "mysqldump -u deploy sd_production > /home/luca/dumps/db.sql"
end

task :free_space do
  run "df -h"
end
```

```
cap dump_db
```

```
set :gateway, "firewall.spazidigitali.com"  
role :sd_db, "database.spazidigitali.com"  
  
task :dump_db do  
  run "mysqldump -u deploy sd_production > /home/luca/dumps/db.sql"  
end  
  
task :free_space do  
  run "df -h"  
end
```

```
cap dump_db
```

```
set :gateway, "firewall.spazidigitali.com"
role :sd_db, "database.spazidigitali.com", "replica.spazidigitali.com"

task :dump_db do
  run "mysqldump -u deploy sd_production > /home/luca/dumps/db.sql"
end

task :free_space do
  run "df -h"
end
```

```
cap dump_db
```

```
set :gateway, "firewall.spazidigitali.com"
role :sd_db, "database.spazidigitali.com", "replica.spazidigitali.com"
role :sd_app, "mongrels.spazidigitali.com"

task :dump_db, :roles=>:sd_db do
  run "mysqldump -u deploy sd_production > /home/luca/dumps/db.sql"
end

task :free_space, :roles=>[:sd_db, :sd_app ] do
  run "df -h"
end
```

```
cap dump_db
```

```
set :gateway, "firewall.spazidigitali.com"
role :sd_db, "database.spazidigitali.com", "replica.spazidigitali.com"
role :sd_app, "mongrels.spazidigitali.com"

namespace :spazidig do
  desc "Dumping our DBs"
  task :dump_db, :roles=>:sd_db do
    run "mysqldump -u deploy sd_production > /home/luca/dumps/db.sql"
  end

  desc "Checking free space"
  task :free_space, :roles=>[:sd_db, :sd_app ]do
    run "df -h"
  end
end
```

```
cap spazidig:dump_db
```

```
set :gateway, "firewall.spazidigitali.com"
role :sd_db, "database.spazidigitali.com", "replica.spazidigitali.com"
role :sd_app, "mongrels.spazidigitali.com"

namespace :spazidig do
  desc "What??"
  task :default do
    run "uname -a"
  end

  desc "Dumping our DBs"
  task :dump_db, :roles=>:sd_db do
    run "mysqldump -u deploy sd_production > /home/luca/dumps/db.sql"
  end

  desc "Checking free space"
  task :free_space, :roles=>[:sd_db, :sd_app ]do
    run "df -h"
  end
end
```

```
cap spazidig
cap spazidig:dump_db
cap spazidig:free_space
```

```
set :gateway, "firewall.spazidigitali.com"
role :sd_db, "database.spazidigitali.com", "replica.spazidigitali.com"
role :sd_app, "mongrels.spazidigitali.com"

set :destination, "db.sql"

namespace :spazidig do
  desc "Dumping our DBs"
  task :dump_db, :roles=>:sd_db do
    run "mysqldump -u deploy sd_production > /home/luca/dumps/#{destination}.sql"
  end
end
```

```
cap spazidig:dump_db
```

```
set :gateway, "firewall.spazidigitali.com"
role :sd_db, "database.spazidigitali.com", "replica.spazidigitali.com"
role :sd_app, "mongrels.spazidigitali.com"

set :destination, "db.sql"

set(:database) do
  Capistrano::CLI.ui.ask "Which database?: "
end

namespace :spazidig do
  desc "Dumping our DBs"
  task :dump_db, :roles=>:sd_db do
    run "mysqldump -u deploy #{database} > /home/luca/dumps/#{destination}.sql"
  end
end
```

```
cap spazidig:dump_db
```

```
task :deploy do
  transaction do
    try_this
    then_do_that
```

```
  end
end
```

```
task :try_this do
  on_rollback { run "do_cleanup" }
  #...
end
```

```
task :then_do_that do
  on_rollback { run "do_super_cleanup" }
  #...
end
```

```
on :event, :callback, :only => %w(tasks_to_do)
# :load, :exit, :start, :finish, :begin, :after
```

```
trigger :notify, current_task
```

```
on :notify, :callback
```

```
load_paths << "config/deploy"  
case ENV['STAGE']  
when "production" then  
  load "production"  
when "staging" then  
  load "staging"  
else  
  abort "unknown stage: #{ENV['STAGE']}"  
end
```

```
before :deploy, :run_tests
after "deploy:update_code", :link_database_yaml
task :run_tests do
  system "rake" or abort "tests failed"
end
task :link_database_yaml do
  run "ln -s #{release_path}/config/database.yml #{shared_path}/database.yml"
end
```

# Cap

- `cap -T`
- `cap <some task>`
- `cap invoke COMMAND="df -h"`
- `cap invoke COMMAND="df -h" ROLES=sd_db`
- `cap invoke COMMAND="df -h" HOSTS=www.spazidigitali.com`
- `cap shell`

# Capify

```
capify .
```

- **Creates a Capfile**
- **Creates a deploy "configuration" file**

# Cap

- `cap deploy:setup`
- `cap deploy:check`
- `cap deploy:cold`
- `cap deploy:update`
- `cap deploy:migrate`
- `cap deploy:start`
- `cap deploy`
- `cap deploy:rollback`
- `cap deploy:migrations`
- `cap deploy:web:disable`
- `cap deploy:web:enable`
- `cap deploy:cleanup`

# deploy:setup

- <deploy\_to>
  - **current**
  - releases
    - 200706060120
    - 200702020815
  - shared
    - log
    - pids
    - system

# deploy:check

```
depend :remote, :gem, 'rpdf', '>=1.1.4'  
depend :remote, :directory, "/tmp/rails"  
depend :remote, :writable, "/dir/file.log"  
depend :local, :command, "gem"  
depend :remote, :command, "mysqldump"
```

# deploy:cold

- copies the code - `cap deploy:update`
- migrates the database - `cap deploy:migrate`
- starts the service - `cap deploy:start`

# Variables

- `:ssh_options`
- `:application`
- `:repository`
- `:scm`
- `:deploy_via`
- `:deploy_to`
- `:revision`
- `:use_sudo`
- `:group_writable`
- `:rake`
- `:current_path`
- `:release_path`
- `:shared_path`

# Deployment strategies

- Checkout (classic capistrano)
- Remote cache (svn up)
- Export
- Copy (via checkout or export, with compression)

# Example\*

\*sorta real

# That's all folks

- Luca Mearelli
- <http://spazidigitali.com>
- [lucamea@gmail.com](mailto:lucamea@gmail.com)