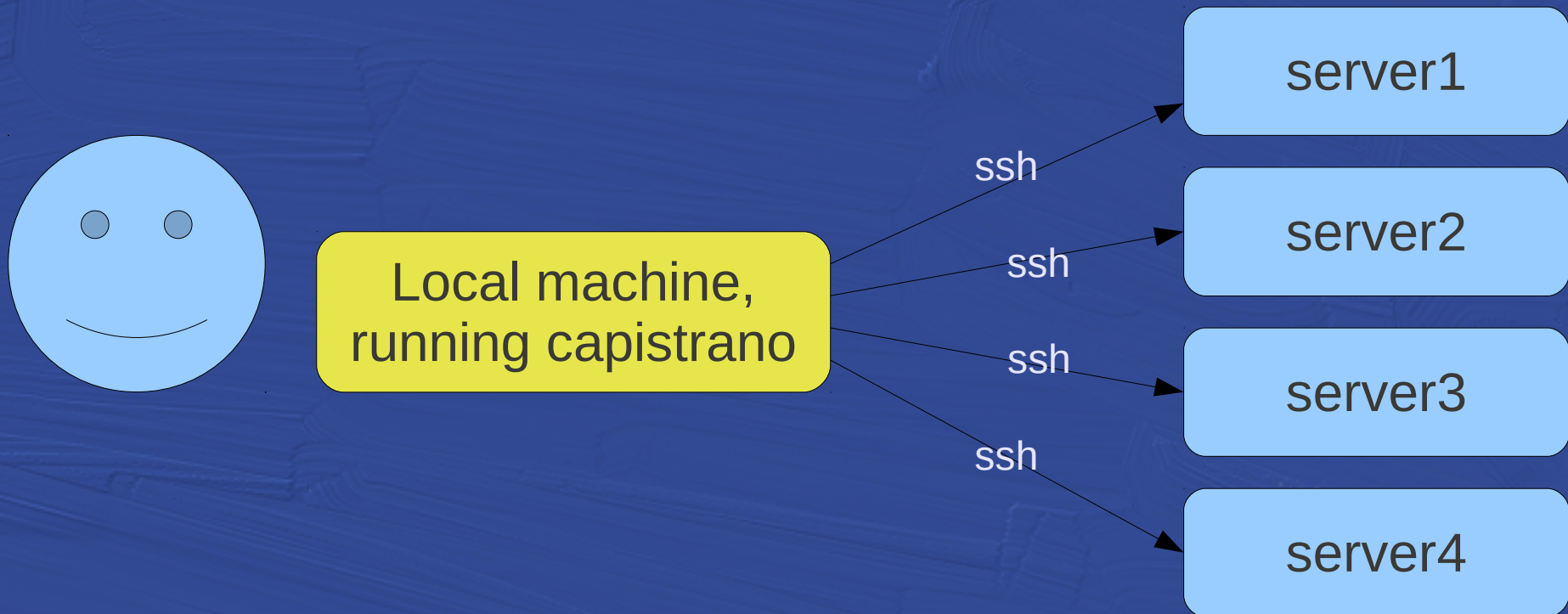


Capistrano

David Grayson
Las Vegas Ruby Meetup
2012-11-7

“Capistrano is a utility and framework for executing commands in parallel on multiple remote machines, via SSH.”



This talk will be about:

- 1) Capistrano in general
- 2) Deploying Rails

Before Capistrano

- Logging in to multiple servers :(
- Run commands with ssh utility.

```
david@strontium:~$  
david@strontium:~$ ssh strontium uptime  
21:48:59 up 1:41, 3 users, load average: 0.73, 0.36, 0.33  
david@strontium:~$ ssh strontium tail -f /var/log/nginx/access.log
```

Simple Capfile #1: tasks

```
task "uptime" do
  run "uptime"
end
```

```
$ HOSTS=strontium cap uptime
* executing `uptime`
* executing "uptime"
  servers: ["strontium"]
  [strontium] executing command
** [out :: strontium] 11:23:39 up 30 min,  2 users,  load average: 1.57,
command finished
```

Simple Capfile #2: hosts

```
task "uptime", :hosts => "strontium" do
  run "uptime"
end
```

```
$ Hosts=stontium cap uptime
* executing `uptime`
* executing "uptime"
  servers: ["strontium"]
  [strontium] executing command
** [out :: strontium] 11:23:39 up 30 min,  2 users,  load average: 1.57,
command finished
```

Simple Capfile #3: servers and roles

```
server "strontium", :app

task "uptime" do
  run "uptime"
end
```

```
$ HOSTS=stontium cap uptime
* executing `uptime`
* executing "uptime"
servers: ["strontium"]
[strontium] executing command
** [out :: strontium] 11:23:39 up 30 min, 2 users, load average: 1.57,
command finished
```

Simple Capfile #4: desc

```
server "strontium", :app

desc "Runs the uptime commmand"
task "uptime" do
  run "uptime"
end
```

```
$ cap -T
cap invoke # Invoke a single command on the remote servers.
cap shell  # Begin an interactive Capistrano session.
cap uptime # Runs the uptime commmand
```

```
Extended help may be available for these tasks.
Type `cap -e taskname` to view it.
$
```


“invoke” task

- Invokes a command on remote servers

```
$ cap invoke COMMAND="uptime"
* 2012-11-04 14:47:14 executing `invoke`
* executing "uptime"
  servers: ["strontium"]
  [strontium] executing command
** [out :: strontium] 14:47:14 up 3:54, 2 users, load average: 1.05,
  command finished in 97ms
$
```

Composition

```
task "uptime" do
  run "uptime"
end
```

```
task "disk_usage" do
  run %{df | awk '$6=="/" {print $5}'}
end
```

```
task "status" do
  uptime
  disk_usage
end
```

```
$ cap -v status
```

```
** [out :: strontium] 14:39:56 up 3:46, 2 users, load average: 0.72,
```

```
** [out :: strontium] 79%
```

```
$
```

Before and after hooks

```
before 'status', 'uptime'  
after  'status', 'disk_usage'
```

Namespaces

```
server "strontium", :app
```

```
namespace "status" do
```

```
  task "default" do
```

```
    uptime
```

```
    disk_usage
```

```
  end
```

```
  task "uptime" do
```

```
    run "uptime"
```

```
  end
```

```
  task "disk_usage" do
```

```
    run %{df | awk '$6=="/" {print $5}' }
```

```
  end
```

```
end
```

status



status:uptime



status:disk_usage



File transfer

- upload
- download
- put
- get

```
download "/etc/crontab",  
"$CAPISTRANO:HOST$.crontab"
```

capture

```
version = capture("cat current/REVISION")
```

- Important: only runs on ONE server!

stream

```
stream "tail -f shared/log/production.log"
```


run_locally

```
run_locally "git push"
```

Variables

- Normal:

```
set(:root_password, "f00")
```

- Deferred:

```
set(:root_password) do  
  Capistrano::CLI.password_prompt("Root password: ")  
end
```

```
task :backup_database do  
  run "bkup --user=root --password=#{root_password}"  
end
```

Variables (continued)

- set
- unset
- exist?
- fetch

Special Significant Variables

- <https://github.com/capistrano/capistrano/wiki/2.x-Significant-Configuration-Variables>
- `default_environment`
- `shared_children`

Errors

- Based on command return code

```
task "foo" do
  run "ls missing_file"
end
```

```
$ cap foo
* 2012-11-04 14:59:48 executing `foo`
* executing "ls missing_file"
servers: ["strontium"]
[strontium] executing command
*** [err :: strontium] ls: cannot access missing_file
*** [err :: strontium] : No such file or directory
command finished in 131ms
failed: "sh -c 'ls missing_file'" on strontium
$
```

Transactions

```
task "update" do
  transaction do
    copy
    start
  end
end
```

```
task "copy" do
  on_rollback { run "rm -fv b d" }
  run "cp a b"
  run "cp c d"
end
```

```
$ cap update
* 2012-11-04 15:10:27 executing `update`
** transaction: start
* 2012-11-04 15:10:27 executing `copy`
* executing "cp a b"
  servers: ["strontium"]
  [strontium] executing command
*** [err :: strontium] cp: cannot stat `a`
*** [err :: strontium] : No such file or directory
  command finished in 117ms
*** [copy] rolling back
* executing "rm -fv b d"
  servers: ["strontium"]
  [strontium] executing command
  command finished in 9ms
failed: "sh -c 'cp a b'" on strontium
$
```

Deploying Rails

- Online tutorials
- Check out <https://github.com/capistrano/capistrano/blob/master/lib/capistrano/recipes/deploy.rb>
- For assets, check out: <https://github.com/capistrano/capistrano/blob/master/lib/capistrano/recipes/deploy/assets.rb>
- **WARNING:** `deploy:rollback` doesn't roll back assets.

Semi-Official Extensions

- [Capistrano Multistage, Deploying to Multiple Environments](#)
- [Rails-less Deploy](#)

3rd Party Extensions

- [capistrano-windows-server](#) Deploy Rails apps to Windows servers
- [capistrano-offroad](#) For deploying non-rails apps, specially Django
- [capistrano_winrm](#) Run commands against MS Windows servers with WinRM
- [capify-ec2](#) A number of utilities to help with deploying to amazon ec2 instances
- [resque_utils](#) Helper tasks for remote Resque deployments (requeue & remove failed jobs)
- [capistrano-ash](#) For deploying Magento, Drupal, WordPress, and Zend/Doctrine apps
- [play-capistrano](#) For deploying/controlling remote Play Framework apps
- [capifony](#) For deploying symfony and Symfony2 apps
- [capcake](#) extension of Capistrano, deploy CakePHP apps
- [capistrano-chef](#) Integrates Capistrano with Chef search
- [capistrano-puppet](#) Integrates Capistrano with Puppet
- [capistrano-deploytags](#) Track your current and all previous releases with Git tags automatically at deployment time
- [capistrano-detect-migrations](#) Automatically discover and warn about pending Rails migrations with Git before you deploy
- [capistrano-notifier](#) Notify via mail and StatsD when deploying
- [capistrano-remote-cache-with-project-root](#) Adds a new deployment strategy to deploy your app from an internal folder (non-root).
- [magentify](#) Deployment recipes customised for deployment of Magento projects.

The End