

Receta Capistrano para NodeJs

Ésta es una guía rápida en la que te explicamos como hacer deployment de una aplicación NodeJS usando Capistrano, lo que se suele llamar receta o recipe Capistrano.

Se presupone que ya tienes un entorno local configurado y que ya tienes un proyecto git remoto, como por ejemplo en la forja o en github. También está pensada para usuarios de guebs.com. Si necesitas instrucciones más concretas existe una guía detallada.

Se presupone que ya tienes acceso ssh autenticado con clave pública a tu plan de alojamiento y has creado la aplicación en el panel de hosting. Usaremos para esta guía el nombre **miapp**.

Paso Previo

- Es necesario que crees la aplicación NodeJs en el panel de hosting. Hecho esto, entonces debes acceder a tu plan de alojamiento y realizar lo siguiente

```
<br />  
cd nodejs ; rm -fr current ; mv miapp current ; ln -s current  
miapp<br />
```

Con esto dejaremos el directorio preparado para el deployment con capistrano

En el equipo donde estés desarrollando tu proyecto, típicamente en un entorno local, debes instalar capistrano y las herramientas de capistrano-npm, que es tan sencillo como ejecutar el siguiente comando:

```
<br />  
gem install capistrano<br />  
gem install capistrano-npm<br />
```

Prepara la aplicación

Ahora debemos ir a la raíz de nuestra aplicación en el equipo local y ejecutar el siguiente comando que instalará los archivos necesarios para que capistrano maneje la aplicación. Debe ser ejecutado en la raíz de la aplicación:

```
<br />  
cap install<br />
```

Hoy en día capistrano pregunta al usuario si quiere usar harrow.io. El objeto de esta guía no es este servicio por lo que decimos que no en este momento.

En el archivo **Capfile** debemos añadir las siguientes líneas:

```
<br />  
require 'capistrano/npm';<br />
```

Receta para hacer deployment

Hecho esto en el archivo **config/deploy.rb** debemos configurar una serie de parámetros

```
<br />  
set :application, 'miapp';<br />  
set :repo_url, 'git@example.com:me/my_repo.git';<br />  
set :deploy_to, '/home/usuario_cpanel/nodejs';<br />
```

Ahora debemos configurar el archivo **config/deploy/production.rb** . En este archivo introduciremos las credenciales y direcciones con las que queremos que capistrano publique. En este caso que estamos describiendo, necesitamos el usuario principal del plan de alojamiento y el nombre de dominio. Además introduciremos la opción adicional «port», que nos servirá para decirle a capistrano que queremos usar el puerto 333 en lugar del estándar 22.

```
<br />  
server 'tudominio.com', user:
```

```
&#8216;usuario_cpanel&#8217;;, roles: %w{app db web}, port:
333<br />
set :tmp_dir, &#8216;/home/usuario_cpanel/tmp&#8217;<br />
set :application, &#8216;miapp&#8217;<br />
append :linked_files, config.js, otro_archivo<br />
```

Gestión de archivos fuera del deployment.

Hay algunos archivos que es necesario mantener sólo en el servidor de producción , es decir, en el plan de alojamiento. Estos archivos serán linkados de tal modo por capistrano que esperará que existan en el directorio shared.

Por ejemplo, típicamente especificaremos que el archivo **config.js** (en el caso de ghost) sea uno de estos archivos, por lo que no deberá existir en el repositorio y deberá existir en **shared/config.js**.

También podemos hacerlo con aquellos directorios que estén en la misma situación. Para ello disponemos de las opciones **linked_dirs** y **linked_files**. Por ejemplo para añadir el archivo config.js, en el archivo deploy/production.rb debemos escribir lo siguiente:

```
<br />
append :linked_files, config.js, otro_archivo<br />
```

Receta para reiniciar la aplicación

Para que cada vez que se realice un deployment de la aplicación ésta se reinicie, debemos crear el archivo **lib/capistrano/tasks/touch.rake** con el siguiente contenido:

```
<br />
namespace :deploy do<br />
desc Reinicia aplicacion<br />
  task :restart do<br />
    on roles(:all) do<br />
      execute mkdir -p #{ current_path }/tmp/ ; touch #{
current_path }/tmp/restart.txt<br />
    end<br />
  end<br />
end<br />
```

```
end<br />
```

Hacer deploy

Gracias a las directivas que hemos ido añadiendo, la aplicación se instalará en el lugar que hemos indicado y como hemos indicado. Además tratará de realizar un «npm install» para instalar los módulos en el plan de alojamiento.

Una vez seguidos los pasos anteriores, cada vez que queramos hacer deployment de nuestra aplicación ejecutaremos el siguiente comando en nuestro ordenador:

```
<br />  
cap production deploy<br />
```

Si, por algún motivo, únicamente queremos reiniciar la aplicación:

```
<br />  
cap production deploy:restart<br />
```

Problemas habituales

Advertencias al hacer deploy

En el entorno de guebs.com, cada vez que ejecutemos **cap production deploy** puede que nos salgan los siguientes mensajes de error que podemos ignorar, no tienen relevancia:

```
mkdir:          cannot          create          directory  
`/home/usuario_cpanel/public_html': File exists
```

Error en la instalación de módulos

Algunos módulos necesitan compilación, por tanto, hay que activar el acceso a compiladores.

Notas finales

Actualizada a día 29 de junio de 2016 para capistrano > 3.x