# Use Capistrano to deploy Puppet manifests

**1)**So , we change now a little bit the **config/deploy/staging.rb** file that is looking now like this:

```
server '192.168.42.198', :app, :web, :primary => true

task :stage_dir do
  run "mkdir /home/leonidas/www/multi-stages/stage_dir"
end

task :puppet_man do
  run "mkdir -p /etc/puppet/modules/helloworld/manifests"
  run "touch /etc/puppet/modules/helloworld/manifests/init.pp"

  run "echo \"class helloworld { file { '/tmp/helloFromMaster': content => 'See you at crowdpark!' }}\" > /etc/puppet/modules/helloworld/manifests/init.pp"

  run "touch /etc/puppet/manifests/site.pp"
  run "echo 'include helloworld' > /etc/puppet/manifests/site.pp"

  sudo "service puppetmaster restart"
end
```

and the **config/deploy/production.rb** file is looking now like this:

```
server '192.168.42.152', :app, :web, :primary => true

task :proc_dir do
  run "mkdir /home/leonidas/www/multi-stages/proc_dir"
end

task :slave_restart do
  sudo "service puppet restart"
end
```

**2)**Firstlly we run **cap staging puppet_man** and then **cap production slave_restart.**
**3)**After that we must see a helloFromMaster text file at the /tmp/ of the puppet-slave machine.

Take in consideration that Capistrano works with git repositories and it pulls the files to the deployment machine, it might make no sense to create an "init.pp" file using Capistrano. We can create this file as we do normally (e.g using a text editor or an IDE) and after that commit and push it to our git-repository. In this case our deploy.rb file will look like this:

```
default_run_options[:pty] = true
ssh_options[:forward_agent] = true
set :use_sudo ,"false"

set :user ,"leonidas"
set :repository ,"https://github.com/lossas/puppet-cap"
set :scm ,:git
```

```
set :scm_user ,"lossas"
set :branch ,"master"

set :application, 'puppet_cap'
set :deploy_to, "/etc/puppet/modules/#{application}/"
set :deploy_via, :copy
set :copy_dir, "/Users/leonidas/www/puppet-cap/manifest"
set :copy_remote_dir, "/etc/puppet/modules/#{application}/"

role :serv1, "192.168.42.198"

   def remote_file_exists?(full_path)
      'true' ==  capture("if [ -e #{full_path} ]; then echo 'true'; fi").strip
   end

desc "Deploy puppet modules"
task :deploy_puppet, :roles => [:serv1] do
  #if remote_file_exists?('#{latest_release}/manifest/init.pp')
  #  run "echo file exists!"
  #else
    run "puppet apply #{latest_release}/manifest/init.pp"
  #end
end
```

Now there is the "manifest" folder unter the "releases" and it contains the init.pp file.