

# Creació d'una plataforma LAMP mitjançant Puppet

**Carles Amigó Martínez**

Enginyeria tècnica d'informàtica de gestió  
Universitat Oberta de Catalunya  
Treball de fi de carrera

# Què fa un administrador de sistemes?

- Gestiona un número elevat de màquines
- Optimitza les comunicacions entre elles
- Treballa directament amb altres departaments per optimitzar la infraestructura
- Monitoritza el correcte funcionament de tota la arquitectura i arregla possibles problemes

# Perquè un gestor de configuracions automàtic?

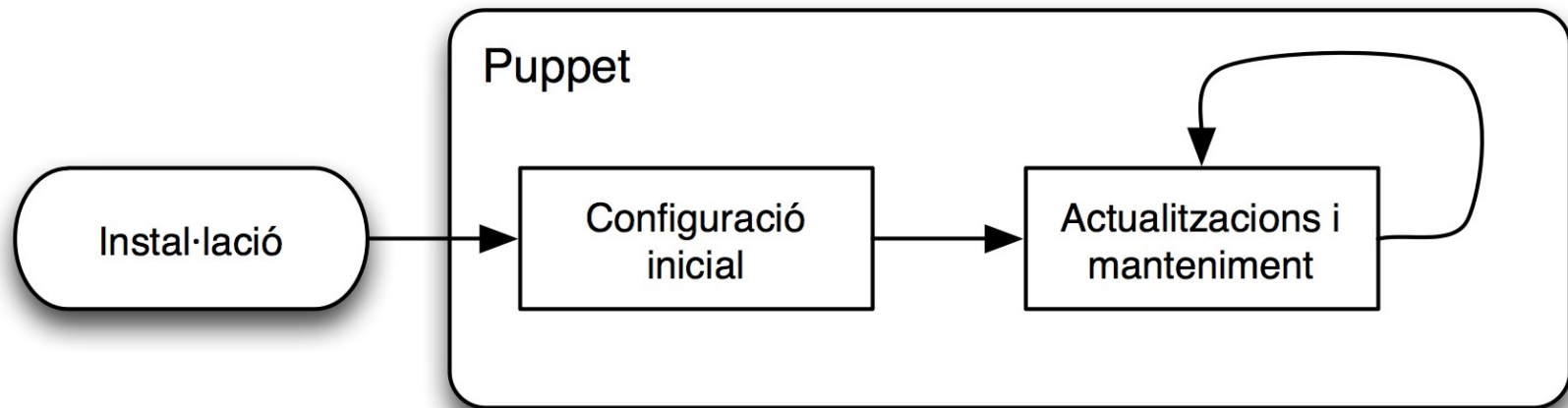
- Un número elevat de màquines comporta un elevat cost de manteniment



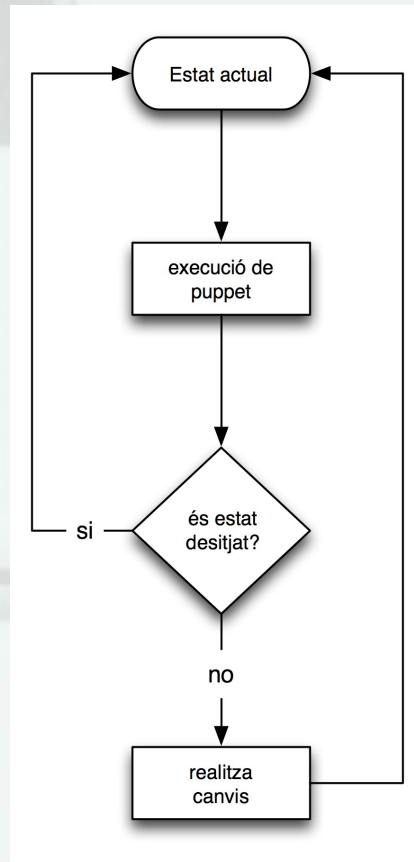
# Què és Puppet?

- Puppet és:
  - Un llenguatge declaratiu per expressar configuracions de sistema
  - Un client/servidor per distribuir aquestes configuracions
  - Una llibreria per realitzar la configuració en si
- Puppet és una capa d'abstracció entre l'administració de sistemes i el sistema en si

# Cicle de vida

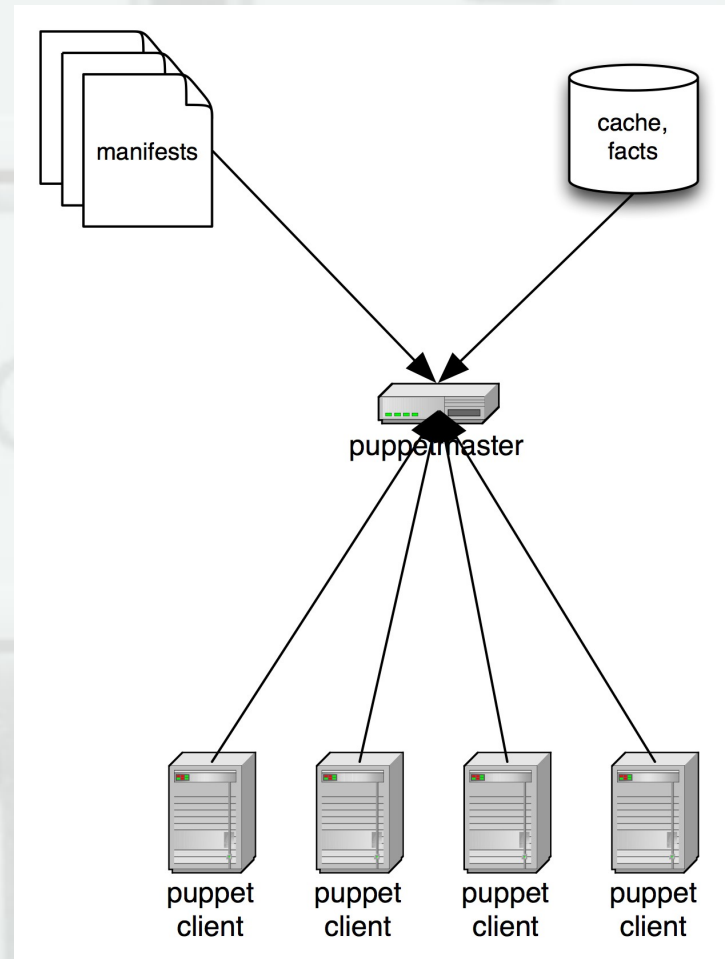


# Com funciona Puppet?

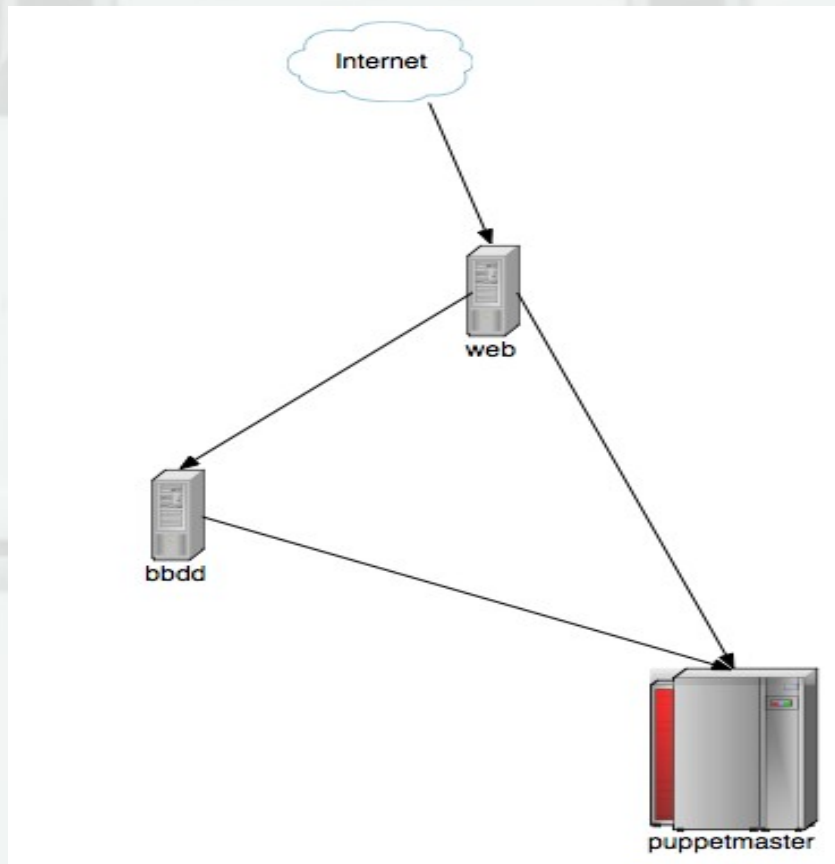


- Es defineix un estat desitjat amb el llenguatge
- Puppet verifica si l'estat actual és igual que l'estat desitjat
- En cas contrari executa els canvis necessaris per igualar l'estat desitjat amb l'actual

# Arquitectura client/servidor



# Plataforma LAMP fent servir Puppet



- Disposem de 3 servidors:
  - Web
  - Bbdd
  - Puppetmaster
- Podríem tenir un número molt més elevat amb el mateix cost



# Node web

```
node "tfc-web.fr3nd.lan" {  
  
  include apache  
  apache::configure { $fqdn:  
    keepalive => "On",  
    maxclients => "500",  
  }  
  
  apache::module { "php":  
    ensure => present,  
  }  
  
  apache::vhost { "www.uoc.edu":  
    server_alias => "uoc.edu",  
    document_root => "/var/www/html/uoc",  
  }  
  
  firewall::rule { "allow_port_80":  
    comment => "Allow webserver",  
    sources => "0.0.0.0/0",  
    interface => "eth0",  
    protocols => "tcp",  
    ports => "80",  
    action => "ACCEPT",  
  }  
  
}
```

- Instal·la apache
- Configura maxclients
- Instal·la php
- Configura un vhost
- Obre tallafocs

# Node bbdd

```
node "tfc-bbdd.fr3nd.lan" {
  $mysql_rootpw = "TfCpassUOC"
  include mysql::server
  mysql::server::configure { $fqdn:
  }

  firewall::rule { "allow_port_3386":
    comment => "Allow MySQL",
    sources => "0.0.0.0/0",
    interface => "eth0",
    protocols => "tcp",
    ports => "3386",
    action => "ACCEPT",
  }

  mysql_database { "tfc":
    ensure => present,
  }
}
```

- Assigna password
- Instal·la MySQL
- Configura MySQL
- Obre firewall
- Crea base de dades

# Node puppet

```
node "tfc-puppet.fr3nd.lan" {  
    package { "puppet-server":  
        ensure => installed,  
    }  
  
    firewall::rule { "allow_port_8140":  
        comment    => "Allow puppet master",  
        sources     => "0.0.0.0/0",  
        interface  => "eth0",  
        protocols  => "tcp",  
        ports      => "8140",  
        action     => "ACCEPT",  
    }  
}
```

- Instal·la servidor de Puppet
- Obre tallafocs

# Exemples d'execució

```
apache::configure { $fqdn:  
  keepalive => "On",  
  maxclients => "250",  
  listen    => "8080",  
}
```

- Imaginem que volem fer canvis a apache:
  - Port 8080
  - Maxclients 250

```
[root@tfc-web ~]# puppet agent --test --server tfc-puppet.fr3nd.lan
info: Retrieving plugin
info: Loading facts in apache
info: Loading facts in mysql
info: Loading facts in apache
info: Loading facts in mysql
info: Caching catalog for tfc-web.fr3nd.lan
info: Applying configuration version '1326319060'
--- /etc/httpd/conf/httpd.conf 2011-12-10 19:40:14.360289535 +0100
+++ /tmp/puppet-file20120111-1435-5p6r1b-0 2012-01-11 22:58:09.891776392 +0100
@@ -102,7 +102,7 @@
MinSpareServers      5
MaxSpareServers     10
ServerLimit         450
-MaxClients          500
+MaxClients          250
MaxRequestsPerChild 0
</IfModule>

@@ -115,7 +115,7 @@
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule worker.c>
StartServers        50
-MaxClients          500
+MaxClients          250
MinSpareThreads     25
MaxSpareThreads     75
ThreadsPerChild     25
@@ -131,7 +131,7 @@
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
-Listen 80
+Listen 8080

#
# Dynamic Shared Object (DSO) Support
info: FileBucket adding {md5}f5259b0a9a52dbcca7d7341bba4be918
info: /Stage[main]/Node[tfc-web.fr3nd.lan]/Apache::Configure[tfc-web.fr3nd.lan]/File[/etc/httpd/conf/httpd.conf]: Filebucket
ed /etc/httpd/conf/httpd.conf to puppet with sum f5259b0a9a52dbcca7d7341bba4be918
notice: /Stage[main]/Node[tfc-web.fr3nd.lan]/Apache::Configure[tfc-web.fr3nd.lan]/File[/etc/httpd/conf/httpd.conf]/content:
content changed '{md5}f5259b0a9a52dbcca7d7341bba4be918' to '{md5}ad5cdd0380d6326c18ae9887b08f174d'
info: /Stage[main]/Node[tfc-web.fr3nd.lan]/Apache::Configure[tfc-web.fr3nd.lan]/File[/etc/httpd/conf/httpd.conf]: Scheduling
refresh of Service[httpd]
notice: /Stage[main]/Apache/Service[httpd]: Triggered 'refresh' from 1 events
notice: Finished catalog run in 33.79 seconds
```

# Màgia?

- Tota la “màgia” està als mòduls
- Prèviament s'ha escrit un mòdul d'apache que en gestiona la configuració
- Per a aquest projecte hem escrit tres mòduls diferents:
  - Apache
  - MySQL
  - Firewall

# Extracte del mòdul d'apache

```
file { "/etc/httpd/conf/httpd.conf":  
  ensure => present,  
  notify => Service["httpd"],  
  content => template("apache/httpd.conf.erb"),  
  require => Package["httpd"];  
}
```

```
<IfModule prefork.c>  
StartServers      <# startservers #>  
MinSpareServers  <# minspareservers #>  
MaxSpareServers  <# maxspareservers #>  
ServerLimit      <# serverlimit #>  
MaxClients       <# maxclients #>  
MaxRequestsPerChild <# maxrequestperchild #>  
</IfModule>
```

## Fitxer httpd.conf:

- Reinicia el servei httpd quan és modificat
- Plantilla httpd.conf.erb
- Necessita paquet httpd

## Plantilla

- Substituirà les variables per els valors assignats

# Quants servidors tenim?

- El número de servidors no importa. Aquest canvi es realitzarà exactament de la mateixa manera tant en un com en N servidors
- Un canvi pot ser aplicat en qüestió de minuts en multitud de màquines simultàniament



# I si ens hem equivocat?

- Codi de puppet normalment en programari de control de revisions
- Fàcil tirar enrere un canvi problemàtic o erroni en un número elevat de servidors
- Abans d'aplicar un canvi el podem testejar en un entorn aïllat

# Conclusions

- Puppet facilita la gestió de servidors quan tenim moltes màquines
- No cal tenir un gran número de màquines per aprofitar les avantatges de Puppet: control de versions
- Redueix el cost de manteniment per màquina