

Data Centers Consolidation Deployment

Miguel Vidal

Twitter: @mvidallopez

Jose Castro

Twitter: @jfcastroluis

Master on Free Software

April 19th, 2012

we study libre software

GSyC



Universidad
Rey Juan Carlos

GSyC

© 2011-2012 Miguel Vidal, Jose Castro

This work is licensed under
a Creative Commons Attribution 3.0 License

LibreSoftware
we study libre software



<http://creativecommons.org/licenses/by/3.0>

Introducción

Índice

- 1 **Introducción**
 - Introducción
 - Beneficios
 - Etapas
- 2 Automatización
 - Introducción
 - Métodos
- 3 Gestión de la configuración
- 4 DevOps
 - Contexto
 - Qué es DevOps

Introducción

A medida que la empresa crece, va creciendo su infraestructura tecnológica:

- necesidad de almacenamiento
- aumenta el número de servidores
- diferentes sistemas operativos
- necesidad de nuevos servicios
- algunos servicios se convierten en servicios críticos
- ...

Introducción

El resultado suele ser un conjunto heterogéneo

Introducción

El resultado suele ser un conjunto heterogéneo

Con los problemas que esto conlleva:

- aumenta el riesgo de errores
- surgen problemas de escalabilidad
- se incrementan los agujeros de seguridad

Introducción

El resultado suele ser un conjunto heterogéneo

Con los problemas que esto conlleva:

- aumenta el riesgo de errores
- surgen problemas de escalabilidad
- se incrementan los agujeros de seguridad

¡La administración se convierte en una tarea muy compleja!

Índice

- 1 **Introducción**
 - Introducción
 - **Beneficios**
 - Etapas
- 2 **Automatización**
 - Introducción
 - Métodos
- 3 **Gestión de la configuración**
- 4 **DevOps**
 - Contexto
 - Qué es DevOps

Beneficios

Una buena homogeneización del CPD nos va a permitir:

- reducir el tiempo de administración y despliegue
- aprovechar mejor los recursos hardware
- minimizar considerablemente el riesgo de fallo
- invertir en infraestructura más robusta y resistente a fallos
- facilitar la escalabilidad a medida de las necesidades

Índice

- 1 **Introducción**
 - Introducción
 - Beneficios
 - **Etapas**
- 2 Automatización
 - Introducción
 - Métodos
- 3 Gestión de la configuración
- 4 DevOps
 - Contexto
 - Qué es DevOps

Etapas

La homogeneización del CPD la podemos encarar siguiendo estas etapas:

- 1 Automatización de tareas

Etapas

La homogeneización del CPD la podemos encarar siguiendo estas etapas:

- 1 Automatización de tareas
- 2 Almacenamiento como servicio

Etapas

La homogeneización del CPD la podemos encarar siguiendo estas etapas:

- 1 Automatización de tareas
- 2 Almacenamiento como servicio
- 3 Virtualización de servidores

Etapas

La homogeneización del CPD la podemos encarar siguiendo estas etapas:

- 1 Automatización de tareas
- 2 Almacenamiento como servicio
- 3 Virtualización de servidores
- 4 Alta disponibilidad de servidores y servicios

Etapas

La homogeneización del CPD la podemos encarar siguiendo estas etapas:

- 1 Automatización de tareas
- 2 Almacenamiento como servicio
- 3 Virtualización de servidores
- 4 Alta disponibilidad de servidores y servicios
- 5 Cloud Computing – IaaS

Automatización

Índice

- 1 **Introducción**
 - Introducción
 - Beneficios
 - Etapas
- 2 **Automatización**
 - **Introducción**
 - Métodos
- 3 **Gestión de la configuración**
- 4 **DevOps**
 - Contexto
 - Qué es DevOps

Tareas de un administrador

Las tareas de un administrador de sistemas se pueden clasificar en:

- Tareas repetitivas
- Atención a usuarios
- Eventos de alarmas o emergencias
- Despliegues y tareas nuevas

Automatización

- Sólo hay que hacer las cosas una vez
- Se reduce el tiempo en tareas aburridas
- Minimizamos el riesgo de errores
- Anticipación a los fallos y caídas
- Muchos errores se pueden solucionar automáticamente sin necesidad de intervención

Índice

- 1 **Introducción**
 - Introducción
 - Beneficios
 - Etapas
- 2 **Automatización**
 - Introducción
 - **Métodos**
- 3 **Gestión de la configuración**
- 4 **DevOps**
 - Contexto
 - Qué es DevOps

Métodos – Configuraciones

Hay que parametrizar las configuraciones en función de:

- sistema operativo
- distribuciones
- shell
- jerarquía de ficheros
- comandos
- servicios
- versiones

Métodos – Actualizaciones

Aviso automático de actualizaciones:

- Debian: apticron
- Red Hat / CentOS: /etc/yum/yum-updatesd.conf

```
emit_via = email
email_to = mail@example.com
email_from = mail@example.com
```
- Ports: pkg_version -vIL=

Métodos – Logs

Herramientas para facilitar la comprensión de los ficheros de log:

- Replicación de logs: `syslog-ng`
- Parseador de logs: `scripts`
- Analizadores de logs: `logwatch` y `splunk`

Métodos – Sincronización

NTP

- Muchos ficheros de log en diferentes máquinas
- Permite tener los relojes de los servidores sincronizados

Monitorización

- Software especializado
- scripts ad-hoc

Métodos – Lista daemons

- Aviso de los sistemas y servicios por correo
- Lista de poco tráfico
- Tan sólo para mensajes de máquinas
- Sólo correos de errores o anomalías
- Buen uso de filtros: `procmail`

Métodos – Registro

- Logging de operaciones
- Herramientas: snoop y rootsh
- Seguimiento de incidencias y operaciones
- Herramientas: Trac y Redmine

Métodos – Documentación

- Política de configuración de sistemas
- Políticas de seguridad
- Políticas de actualización
- Procedimiento de tareas

Gestión de la configuración

Configuration management (CM)

- Procesos para **automatizar** y mantener la **consistencia** en sistemas complejos: aeronáuticos, automovilísticos, militares, sistemas espaciales...
- En informática se usa para **centralizar**, **automatizar**, **mantener** y **desplegar** configuraciones a través de un equipamiento informático heterogéneo.
- Se basa en obtener la descripción técnica de un sistema y de sus diversos componentes.

Configuration management (CM)

- Ayuda a automatizar tareas de administración de sistemas.
- Software libre para gestión de la configuración: CFEngine, Chef, Puppet
- Comparativa:
`http://en.wikipedia.org/wiki/Comparison_of_open_source_configuration_management_software`

Por qué Puppet

- **Centralización:** facilita la gestión de múltiples sistemas desde una sola localización pero **manteniendo las ventajas de una configuración distribuida.**
- Usa una especificación de lenguaje multiplataforma.
- Guarda registro de la configuración de cada cliente.
- Podemos generar una configuración idéntica en un nuevo sistema basándonos en la información almacenada.
- Podemos usar un mismo sistema base y aplicar después modificaciones particulares.

DevOps

Índice

- 1 **Introducción**
 - Introducción
 - Beneficios
 - Etapas
- 2 **Automatización**
 - Introducción
 - Métodos
- 3 **Gestión de la configuración**
- 4 **DevOps**
 - Contexto
 - Qué es DevOps

Contexto

- Los departamentos de sistemas y desarrollo trabajan aislados
- Cada uno de los departamentos considera que hace lo correcto para el negocio
- Los desarrolladores no tienen “conciencia” de sistemas
- Ambos departamentos son imprescindibles para el negocio
- Aparecen las metodologías ágiles de desarrollo: muchos cambios muy pequeños

Necesidades

- Romper las barreras entre los departamentos de sistemas y desarrollo.
- Implantar metodologías ágiles en sistemas.
- Usar frameworks de automatización: puppet, chef, cfengine...
- Mecanismos de comunicación efectivos entre ambos departamentos.
- Sistemas ha de estar involucrado en el diseño de la aplicación desde el principio.

El objetivo final es permitir al negocio que reaccione tan rápido, eficiente y fiable como marca el mercado

Índice

- 1 **Introducción**
 - Introducción
 - Beneficios
 - Etapas
- 2 **Automatización**
 - Introducción
 - Métodos
- 3 **Gestión de la configuración**
- 4 **DevOps**
 - Contexto
 - Qué es DevOps

Qué es DevOps

Filosofía apoyada en procedimientos, herramientas y métodos para permitir una colaboración efectiva entre sysadmins y developers que posibiliten eficientemente el objetivo del negocio.

- No es un perfil de empleado
- No es una nueva forma de llamar a los sysadmins
- No se trata de superponer ni menospreciar puestos; al contrario
- No es un problema tecnológico, es un problema del negocio

Data Centers Consolidation Deployment

Miguel Vidal

Twitter: @mvidallopez

Jose Castro

Twitter: @jfcastroluis

Master on Free Software
April 19th, 2012

