

Administrando los 600 ordenadores de los Laboratorios Docentes de la EIF con software libre

AUTORÍA: • Antonio Gutiérrez Mayoral

PRESENTA: • Administrando los 600 ordenadores de los Laboratorios Docentes de la EIF con software libre



Meta

- Antonio Gutiérrez Mayoral
 - Antonio.Gutierrez@urjc.es
- Responsable Laboratorios Docentes EIF
- Jefe sección Escala Técnica Auxiliar Informática de la URJC (PTGAS)
- Enlace [LinkedIn](#) / [Twitter-X](#)



Contexto

- La EIF en la URJC:
 - 13 Grados Universitarios + 2 dobles grados
 - ~ 2500 alumnos
 - ~ 204 profesores
- Laboratorios Docentes de la EIF
 - 24 laboratorios docentes específicos según disciplinas (Aeroespacial, Teleco, Telemática, Arquitectura)
 - 12 laboratorios docentes de ordenadores
 - Sistema operativo único: Ubuntu Linux LTS (ahora 22.04)
 - ~ 625 ordenadores físicos distribuidos en dos Campus
 - Unas 70 asignaturas de todos los grados de la EIF usan estos Laboratorios
 - En el periodo docente ordinario (1q / 2q)
 - En periodo de exámenes de convocatoria ordinaria y extraordinaria
 - Aulas con configuración “examen”: sin cuentas de red, ficheros y filtrado de todo el tráfico HTTP (Squid)
 - **Entre 2500 y 3000 horas de clase al cuatrimestre** entre todos los Laboratorios Docentes Linux



Laboratorios Docentes Linux de la EIF

- ~ 625 ordenadores (terminales de usuario)
- Unos 15 servidores físicos de infraestructura
 - Servidores de virtualización: usando Proxmox VE (open-source)
 - Máquinas virtuales de infraestructura
 - Servidores LDAP, DHCP, repositorios GIT, bases de datos, página web, accesos remotos (VNC/SSH), gestión de cuentas, etcétera.
 - Servidores físicos “puros”: principalmente de almacenamiento (NFS / Backup)
 - Mirrora de paquetes de [Ubuntu](#) y [Debian](#) (para toda la URJC y mundo :)
 - Monitorización Zabbix ([Panel de estado Grafana accesible](#))
 - Dos servidores de NFS para cada Campus, cuatro en total (producción / backup)
 - Servidor de almacenamiento para backups de máquinas virtuales / homes de usuario / etcétera
 - **Todo el software que se utiliza se intenta que sea software libre (~99,99%)**



Algunos retos de este entorno

- **Macro-instalaciones anuales** (o en cualquier momento)
 - Cloud-init de Ubuntu / clonezilla
- **Gestión de cuentas de usuario**
 - **OpenLDAP**
 - Creación de cuentas, cambios de contraseña y renovaciones está automatizado mediante aplicaciones web creadas en PHP, en página de Laboratorios (alumnos y PDI)
- **Creación de paquetes no *publicados* en Ubuntu**
 - Mediante aplicación **reprepro** con scripts automatizados y publicación automática
 - (<https://labs.eif.urjc.es/repo/>)
- **Gestión del cambio**
 - Instalación y actualización de paquetes (apt-get install / remove / upgrade)
 - Modificación del entorno (ficheros de configuración, ficheros en terminales, etc)
 - Tareas automáticas (cron, etcétera)
 - Permisos
 - ...
- **Los profesores piden “cosas” 24x7** y normalmente para las 9h del día siguiente



Gestión del cambio: SCM

- A día de hoy estamos usando *puppet* como SCM (sistema de gestión de la configuración) ya que es el modelo que mejor se ajusta al entorno del Laboratorio.
- Modelo cliente-servidor
- Lenguaje DSL (descriptivo)
- Las máquinas consultan su configuración cada cierto tiempo a un nodo central y aplican los cambios que correspondan
 - Instalación de paquetes (o eliminación)
 - Modificación de ficheros / directorios / permisos
 - Tareas de cron
 - Etcétera
- Mucha comunidad, herramienta madura y longeva ya probada, conocida y rodada



Automatización de la configuración mediante CI/CD

- Además de Puppet, para que la gestión sea automática y desatendida, nos apoyamos en procesos de **CI/CD** (Devops, GitOps o como se le quiera llamar...)
- Los manifiestos de configuración de todo el Laboratorio están “giteados” en un repo con dos ramas: *test* y *prod*
- Cada vez que se introduce un cambio se inserta en una rama “test” que desencadena una serie de acciones (*pipeline* de Gitlab)
- Este *pipeline* realiza una serie de comprobaciones sobre los cambios introducidos a la configuración para asegurar que “no se líá”
- Si los cambios son buenos, se crea una MR en GitLab (merge request) para fusionar esos cambios en la rama *prod*, lo cual desencadenará otro *pipeline* de GitLab
- Finalmente los cambios introducidos (manifiestos nuevos) terminarán en el servidor Puppet
 - Las máquinas del Laboratorio ejecutarán el agente Puppet cada cierto tiempo y aplicarán los cambios pertinentes (aquellos que no hayan ejecutado hasta el momento)



Automatización de la configuración mediante CI/CD

- Generación de pipelines debidos a eventos *push* en repositorio puppet



- Durante este curso académico 2023/2024 se han generado 169 *pipelines* debidos a *commits* efectuados por cambios en la configuración del laboratorio (instalación de paquetes, modificación de ficheros, etcétera).



Automatización de la configuración mediante CI/CD

- Por ejemplo, la tarea de instalar un paquete en 600 ordenadores se reduce a modificar un fichero DSL, hacer commit y push (creando una MR en el camino)
 - La MR se puede crear automáticamente con el modificador “-o” de git ;-)
- Facilita enormemente la gestión diaria de un entorno tan grande
- Automatizar lo máximo posible.
- No sólo utilizamos CI/CD para la gestión de la configuración,
 - Gestión de los servidores de DHCP
 - Creación de OVAs para los alumnos con el entorno del Laboratorio
 - etc



LICENCIAS Y CRÉDITOS

Ilustración: “Búho Libre”, Sergio Rodríguez Asenjo.

Licencia: Creative Commons Atribución 4.0 Intl.



¡GRACIAS!

Copyright 2024

Algunos derechos reservados

Esta presentación se distribuye bajo la **licencia**
“Atribución-CompartirIgual 4.0 Internacional”
de **Creative Commons**, disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

MÁS INFORMACIÓN:

[HTTPS://OFILIBRE.URJC.ES/](https://ofilibre.urjc.es/)