

La cocina del STIC

Alberto Lorenzo Pulido

- alorenzo@ull.edu.es
- Twitter: @8la
- <https://github.com/8la>
- <https://github.com/tic-ull>

Provisión automatizada de
servidores y servicios TIC

- # Ingeniero Técnico en Informática de Gestión por la ULL
- # Vinculado como becario/técnico en la ULL desde 2008
- # Técnico de sistemas en el STIC desde 2013
- # Organizador y técnico: TLP Tenerife (antes Tenerife Lan Party) desde 2006 (www.tlp-tenerife.com)
- # Entusiasta de sistemas, Python, Software Libre, Chef y no tanto de Ruby :P



- # Aplicación de inscripción de TLP Tenerife. (**Django**)
- # Aplicación de inventario IT del STIC (<https://github.com/tic-ull/arritranco>) (**Django**)
- # Refactorización de la API Google Apps de la gestión de usuarios STIC. (**Python**)
- # Soporte supervisor cookbook hubot (<https://supermarket.chef.io/cookbooks/hubot>)
- # Configuración LDAP en cookbook gitlab (<https://supermarket.chef.io/cookbooks/gitlab>)
- # Poodle SSL Fix en cookbook nginx (<https://supermarket.chef.io/cookbooks/nginx>)

La Cocina del STIC

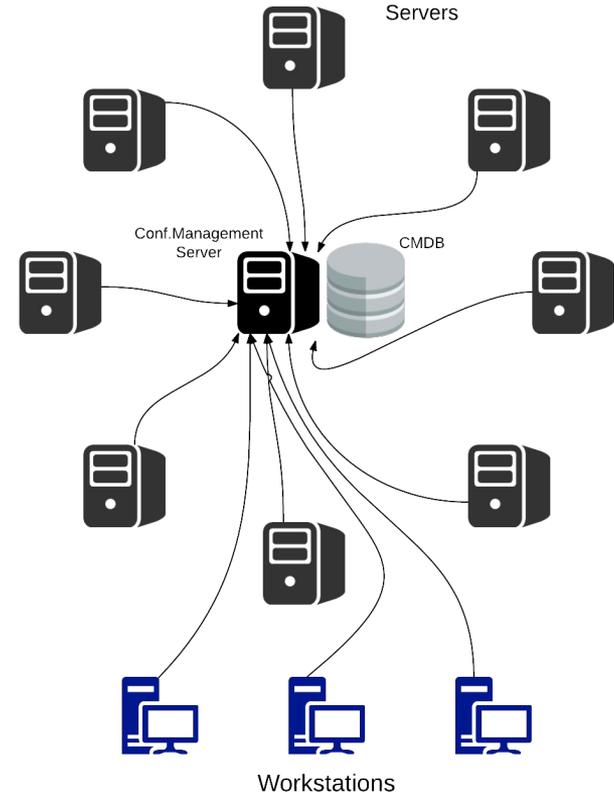
- # La gestión de la configuración / Configuration Management (en adelante: **CM**)
- # Infraestructura como código
- # Retrospectiva de la **CM** en la ULL
- # Adopción de Chef como herramienta desde 2013
- # Pros y contras de la implantación de la **CM** en la infraestructura
- # Estadísticas sobre la **CM** en la ULL



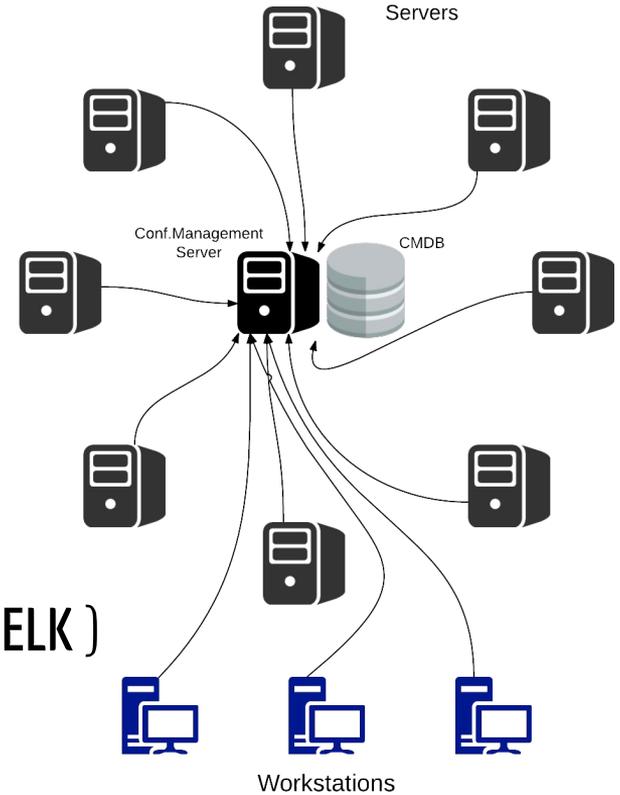
Gestión de la Configuración

The IEEE Glossary of Software Engineering Terminology (Standard 729-1983)

“Es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de esos elementos durante el ciclo de vida, registrando e informando del estado de los elementos y los cambios así como verificando que dichos elementos son correctos y completos.”



- # Infraestructura como código (**Gitlab**)
- # Base de datos (**Postgresql / Chef**)
- # Motor de búsqueda (**Apache Solr / Chef**)
- # Automatización (**Chef**)
- # Monitorización (**Nagios + Graphite + Grafana + ELK**)



Infraestructura como código

- # Ciclo de desarrollo
 - Testing
 - Pre-producción
 - Producción

- # Control del cambio
 - Git / svn / mercurial
 - Gitlab / Bitbucket / Github / Gitolite

- # Integración continua / Entrega continua
 - Jenkins / Travis / GitlabCI

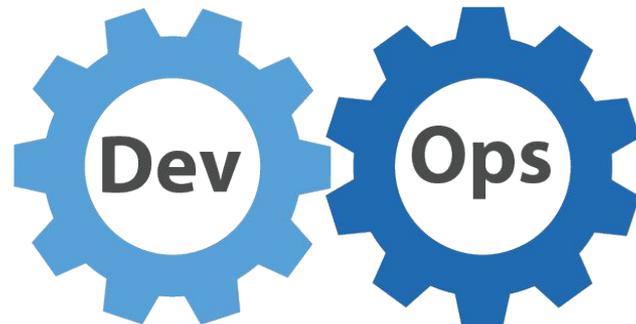
ROL

```
name "stic_base"
description "Rol base para cualquier servidor."
run_list  "recipe[stic_base]",
          "recipe[stic_motd]",
          "recipe[stic_ntp]",
          "recipe[stic_snmp]",
```

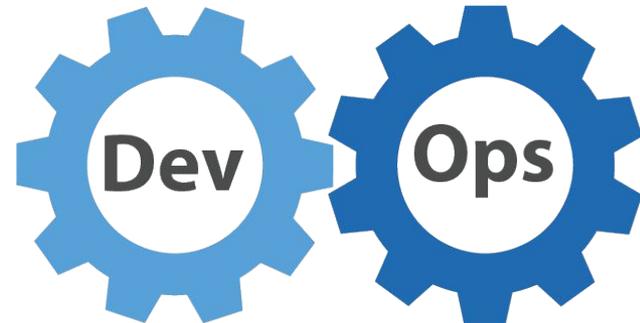
NODO

```
{
  "name": "haproxy2.stic.ull.es",
  "chef_environment": "production",
  "normal": {
    "stic_haproxy": {
      "node_role": "balc_prod"
    }
  },
```

- # Profesional interdisciplinar.
- # Colaboración.
- # Integración de áreas.
- # Metodologías ágiles. (Principios Lean)
- # Integración continua (**CI**)
- # Entrega continua (**CD**)



- # Ampliar el aprendizaje (Iteraciones cortas + feedback)
- # Decidir lo más tarde posible (en base a hechos)
- # Reaccionar tan rápido como sea posible
- # Potenciar el equipo (confiar en los que saben)
- # Ver todo el conjunto (del bit al click)



Flickr veía necesidad de adaptarse al negocio y su velocidad.

Equipo DevOps

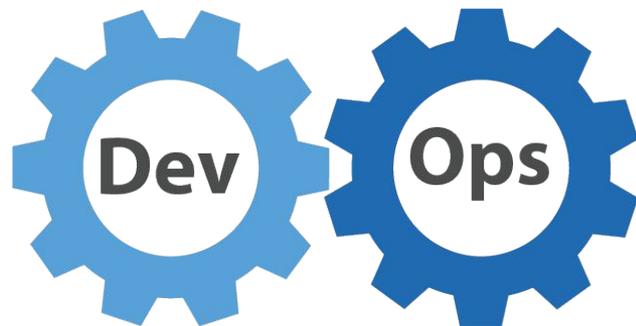
Integración Continua

Entrega Continua

10 Despliegues al día

*youtube & slideshare: 10 Deploys per day (Velocity conf 2009)

flickr



Retrospectiva

2008 (100 Máquinas aprox.)

- Arranque PXE con imagen del instalador (preseed + kickstart)
- Repositorio svn con los scripts de la desatendida y tareas (comando install...)

2010 (150 Máquinas aprox.)

- Servidor BCFG2
- Repositorio svn con los ficheros de configuración de BCFG2
- Aproximadamente 170.000 líneas de código



2013 (210 Máquinas aprox.)

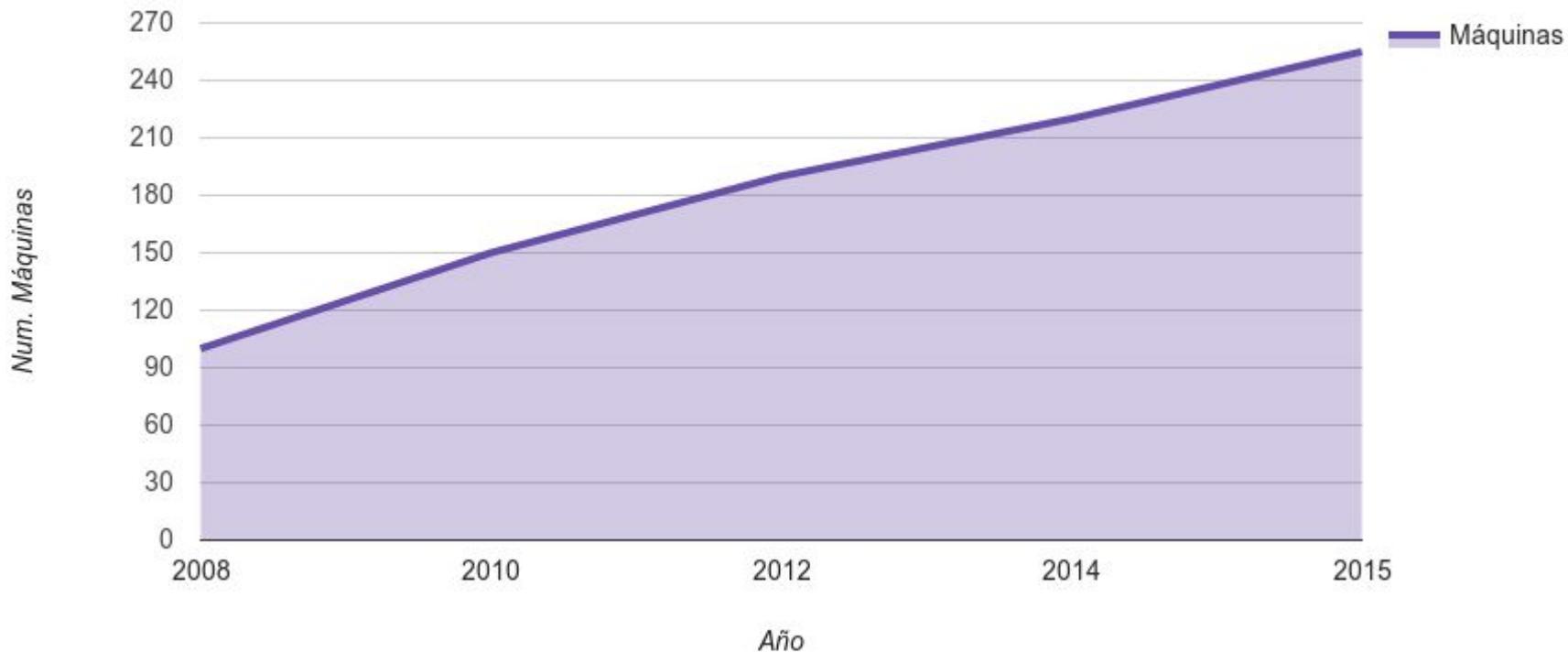
- Servidor Chef 11 Open Source
- Repositorio git para **chef-repo** y cada **cookbook**.
- Gitolite → Gitlab

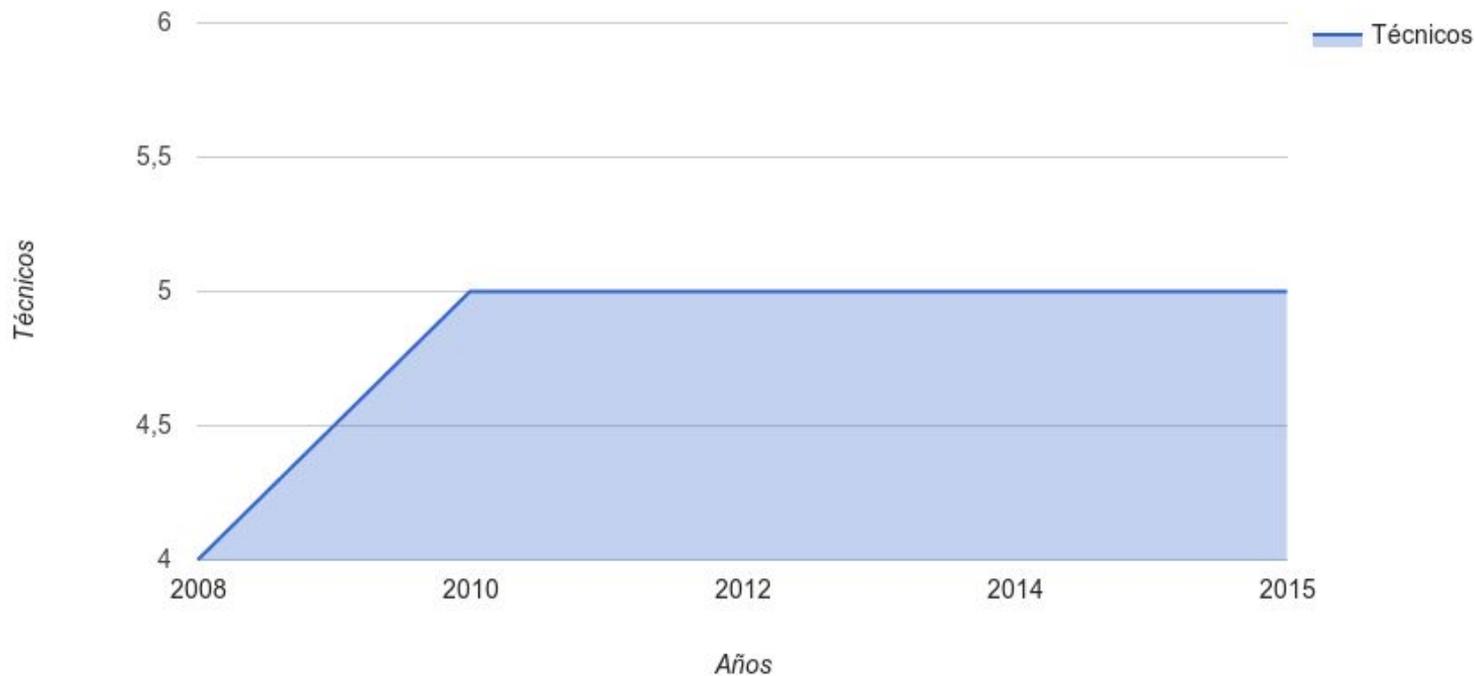
2015 (255 Máquinas aprox.)

- 150.000 Líneas de código **cookbooks** + **chef-repo**

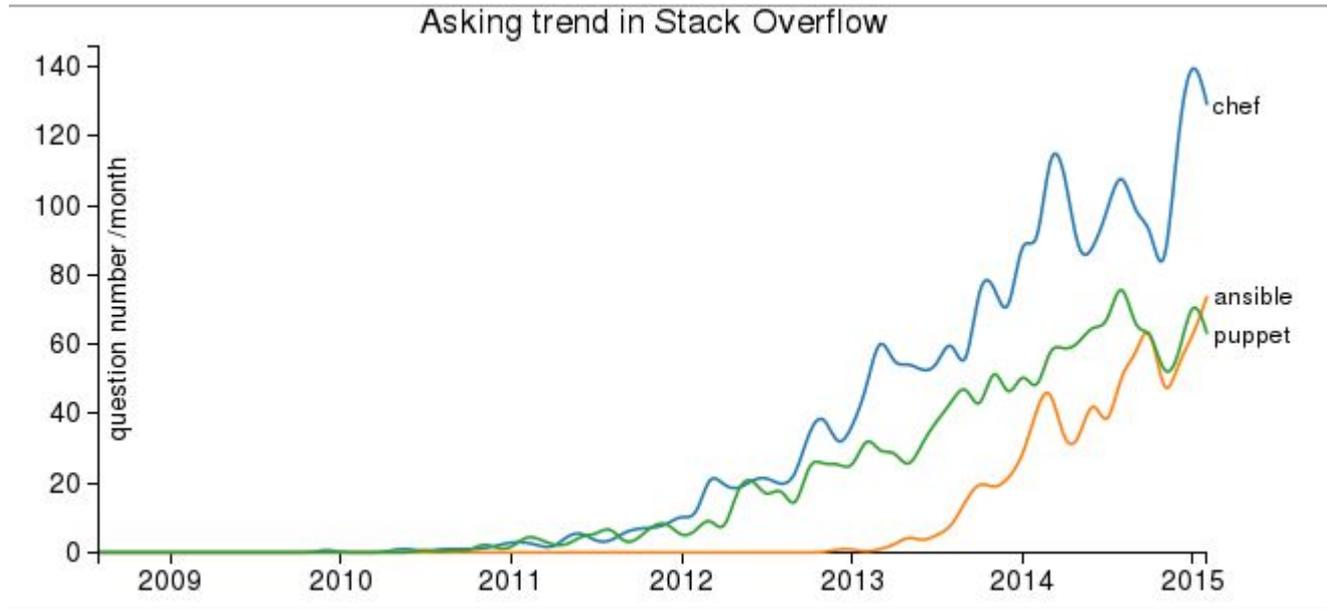


Elección de herramienta: CHEF









URL

del

gráfico:

<https://graphofknowledge.appspot.com/tagcompare/chef&puppet&ansible>

*Los que no aparecen es porque la aplicación no tenía información suficiente para sacar estadísticas.

Lo Bueno....

- # Software Libre (no todo) + Comunidad
- # Chef Development Toolkit
- # Instalación sencilla (omnibus) *
- # Flexibilidad
- # Thick Client - Thin Server

Lo malo....

- # Niveles y cálculo de atributos
- # Los patrones eran muy incipientes
- # Instalación sencilla (omnibus) *
- # Flexibilidad
- # Hay partes, que no son software libre

*Siempre que no quieras una instalación en HA y con capas separadas.

Diagrama Chef

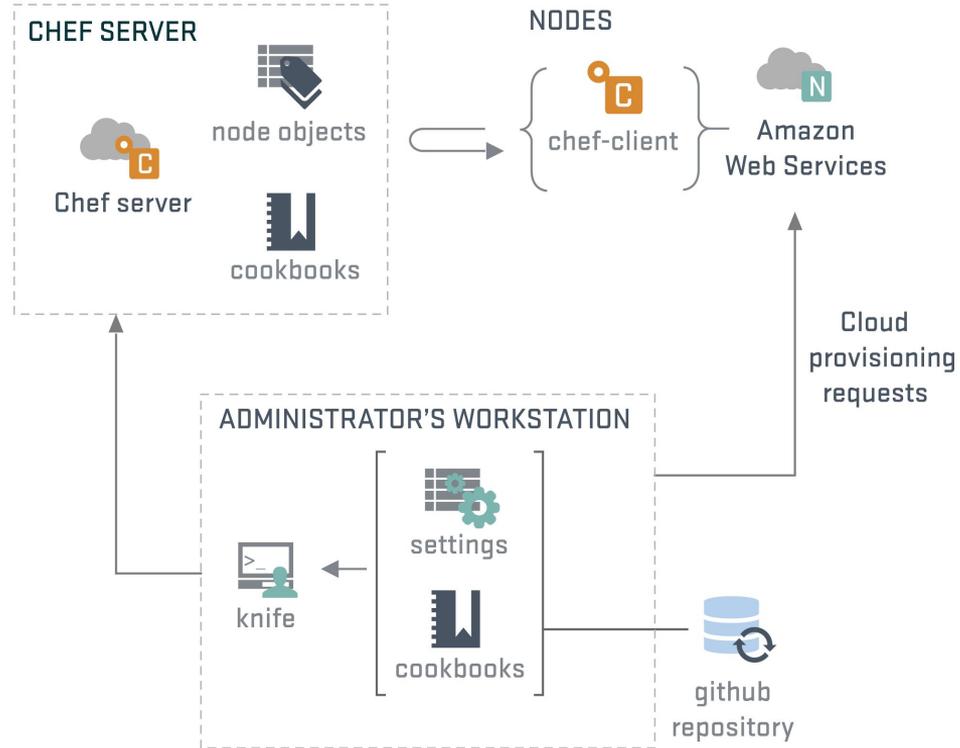


Diagrama Chef

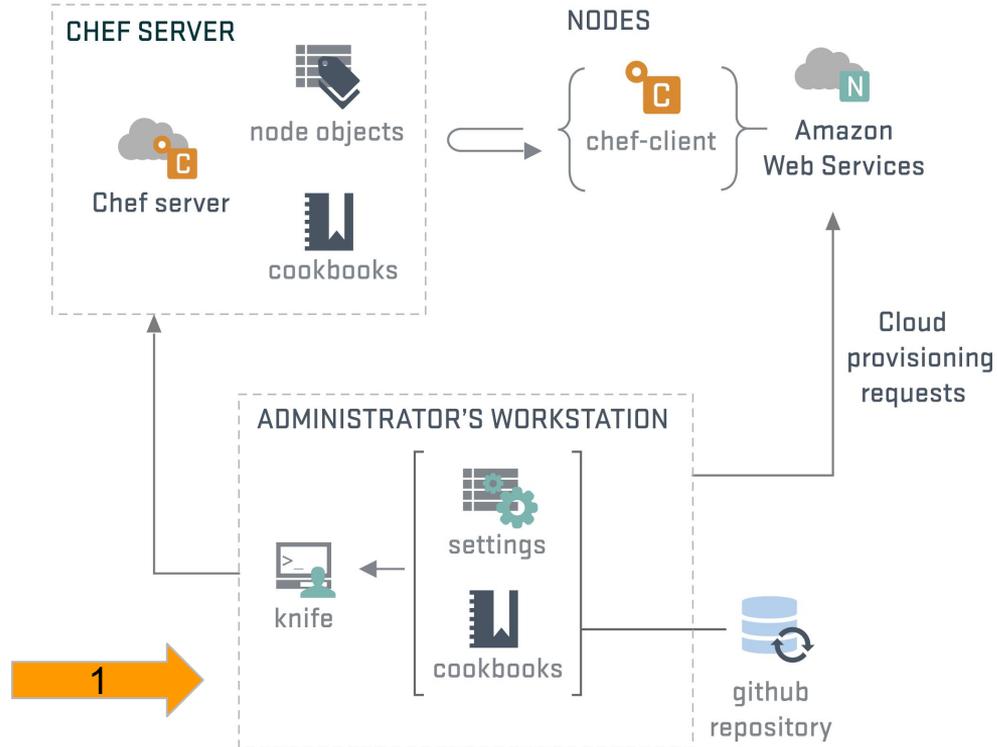


Diagrama Chef

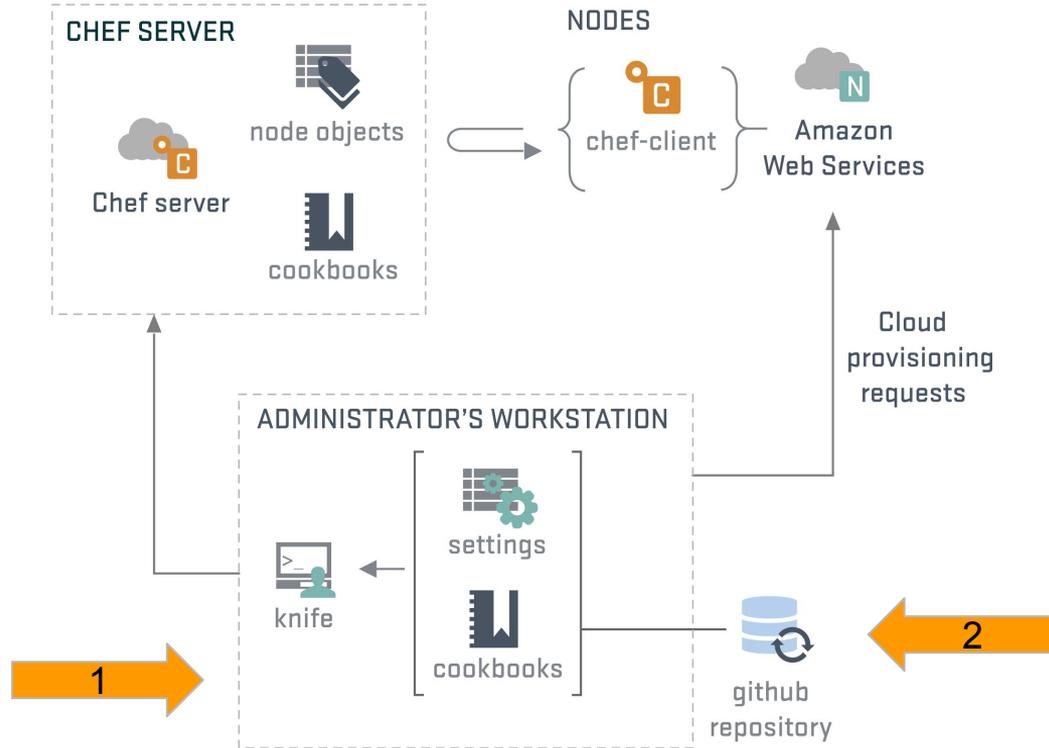


Diagrama Chef

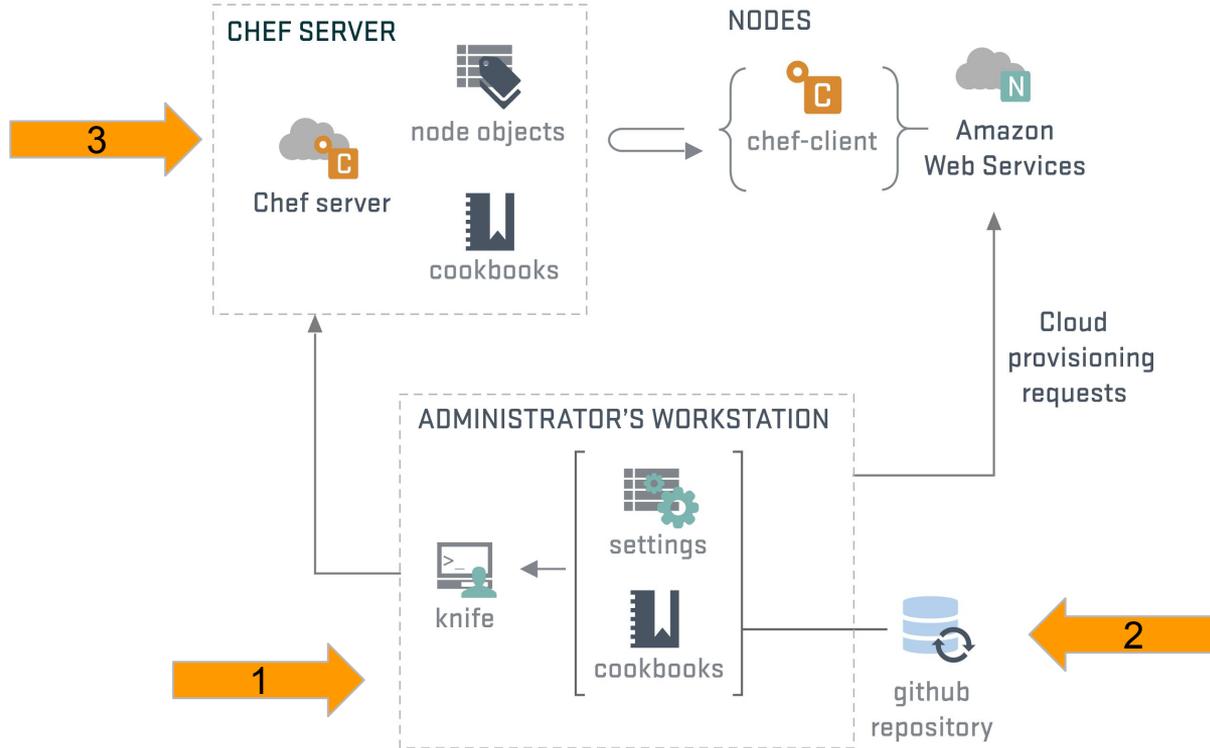
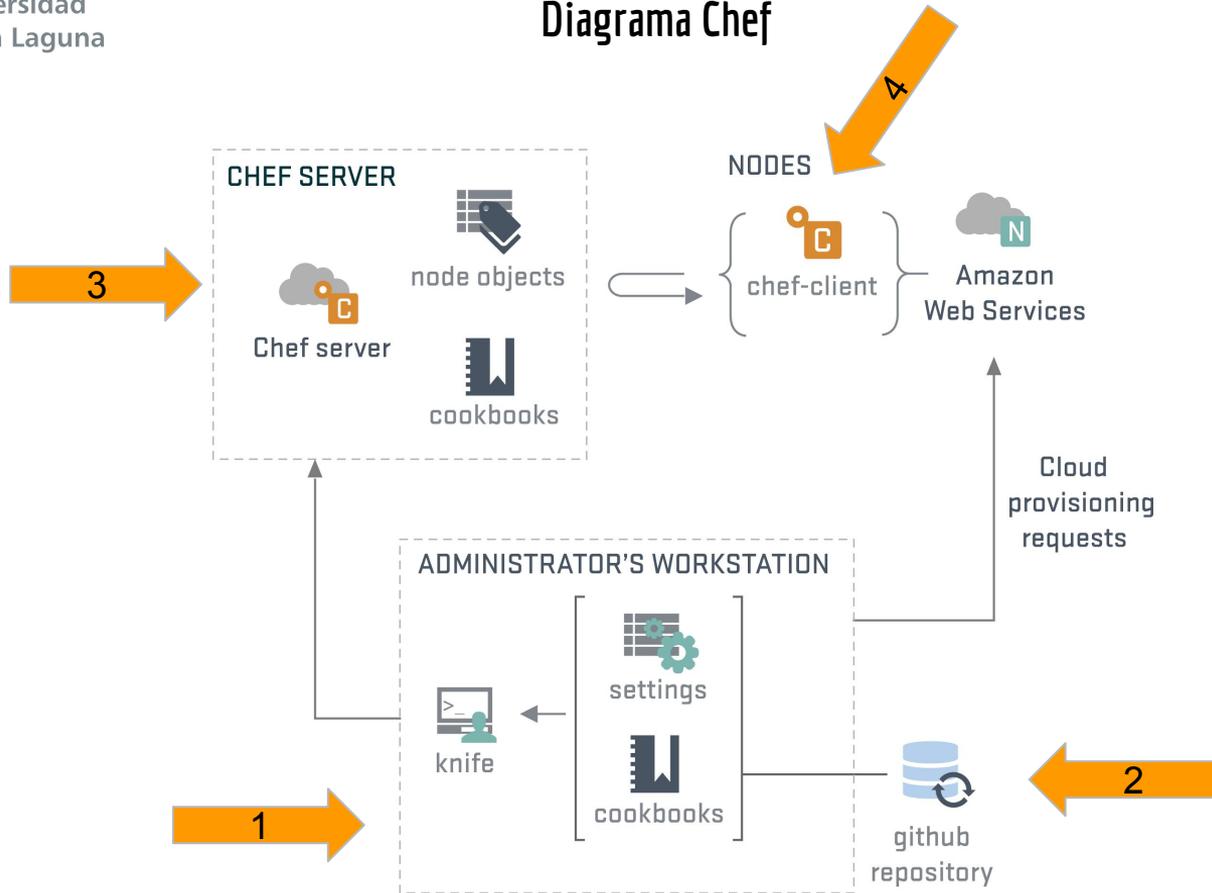


Diagrama Chef



Monitorización

Chequeo pasivos a Nagios3 sobre la ejecución de chef-client

```
chef_client run_status PASV OK  
All right! ----- Start:2015-11-24 22:39:24 +0000 Time:44.882980111 Updated:11 All:267
```

Envío de informe de ejecución a una stack **ELK**

Aviso por correo si hay cambios en la configuración (Correlación con **sec syslog-server**)

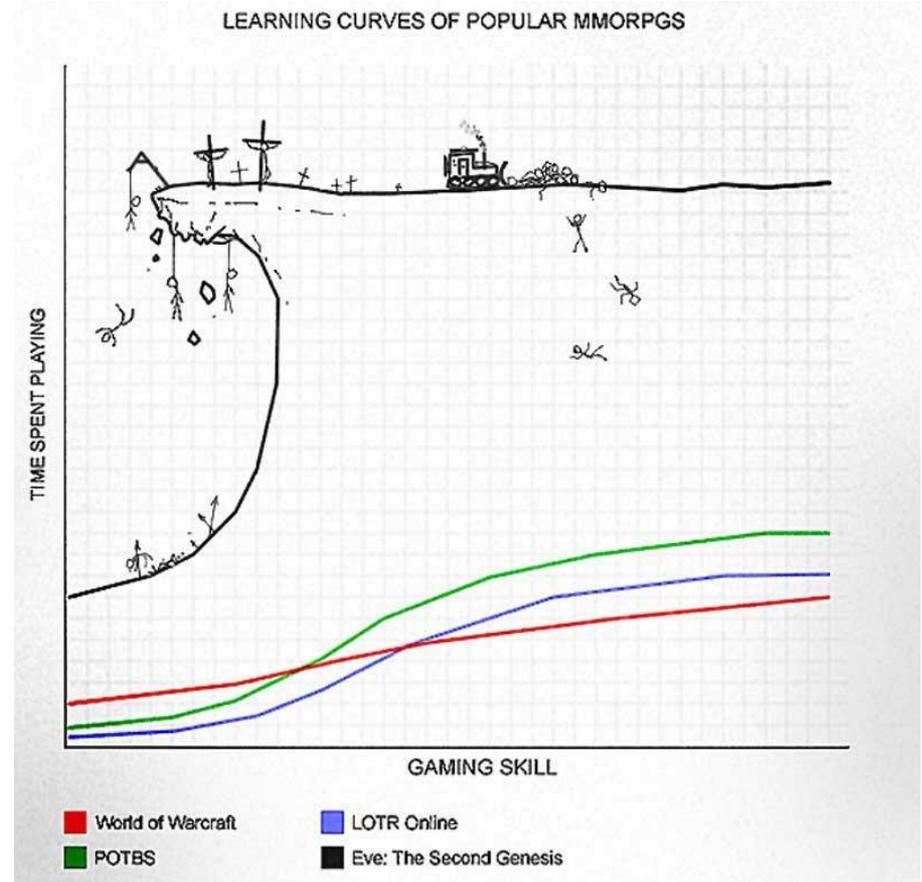
Datos de las máquinas con **Collectd** a **Graphite** y visualizados con **Grafana**

Hubot webHooks en Gitlab



Contraste de la CM

- # Convenio sobre la configuración
- # Tiempo para que de frutos
- # Convencer a la gente
- # Integración de los equipos
- # Curva de aprendizaje / adaptación
- # Servicios heterogéneos



Pros de la CM

- # Mejora las habilidades técnicas
- # Aprendes mucho sobre tu infraestructura
- # Fomenta la colaboración
- # Visibilidad y Control de tu infraestructura
- # Fin de los backups del sistema
- # Gestión del cambio



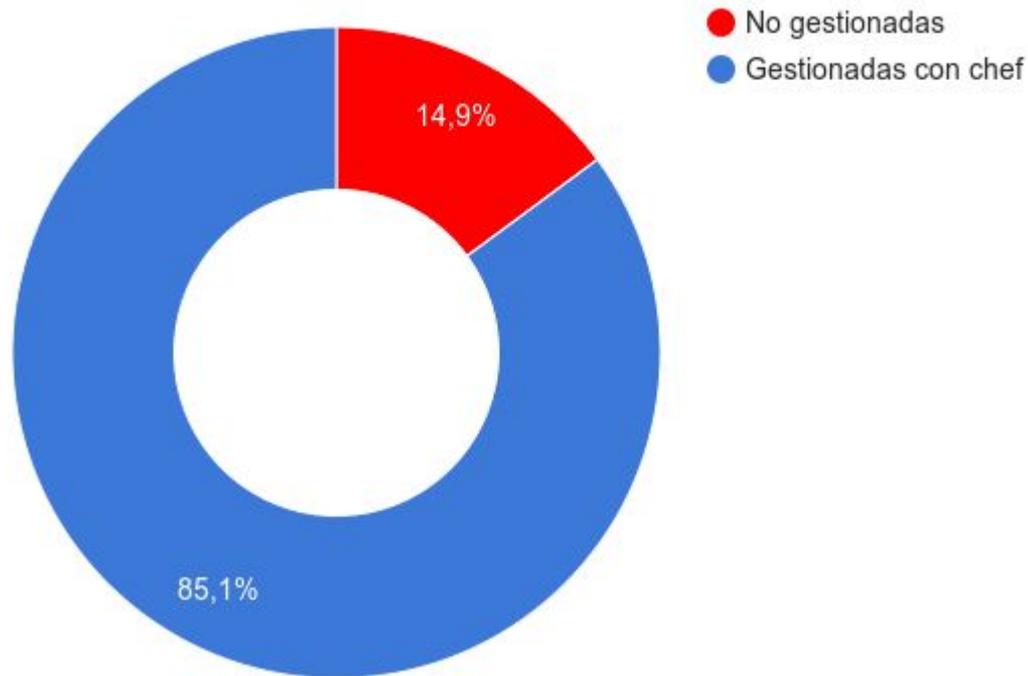
- # Cambio de contraseñas en todos los usuarios que acceden por ssh a las máquinas**
 - Sin Chef: 5 a 7 días (4 técnicos)
 - Con Chef: 1 a 2 horas (1 técnico)

- # Activar uso de resolvconf en los sistemas y configurar el servicio**
 - Sin chef: 10 a 14 días (4 técnicos)
 - Con chef: 3 días (1 técnico)

- # Instalar y configurar un nuevo cluster haproxy con dos nodos.**
 - Sin chef: 2 a 3 días (1 o 2 técnicos)
 - Con chef: 45 min (1 técnico)

Máquinas: 255

“Chef-izadas”: 217

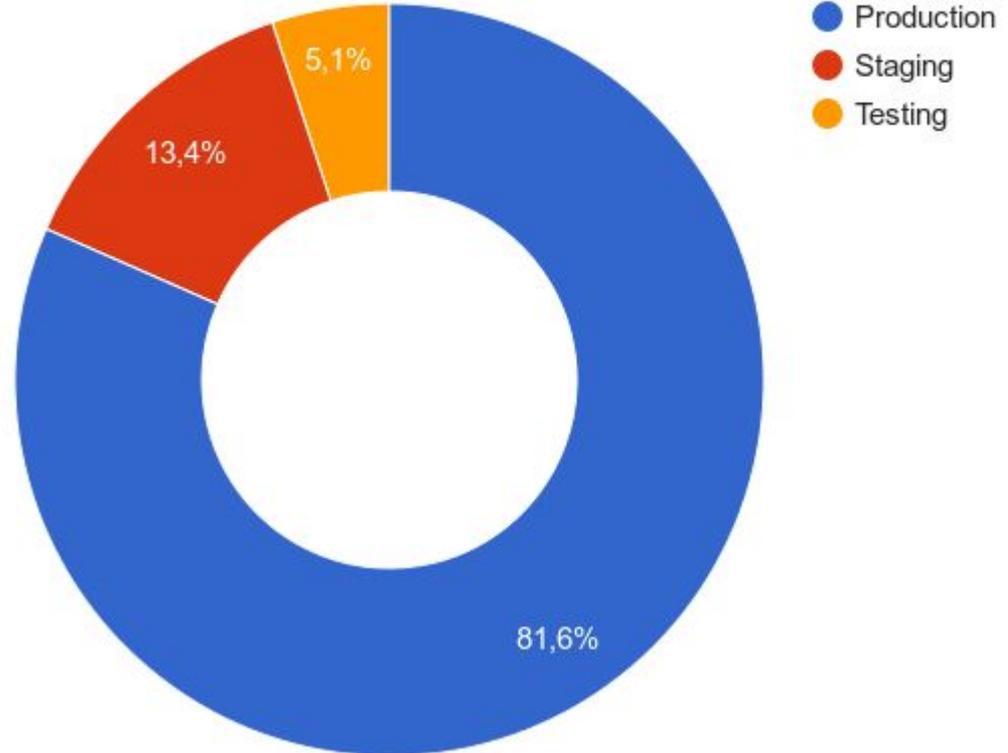


Nodos en Chef: 217

Production: 177

Staging (pre): 29

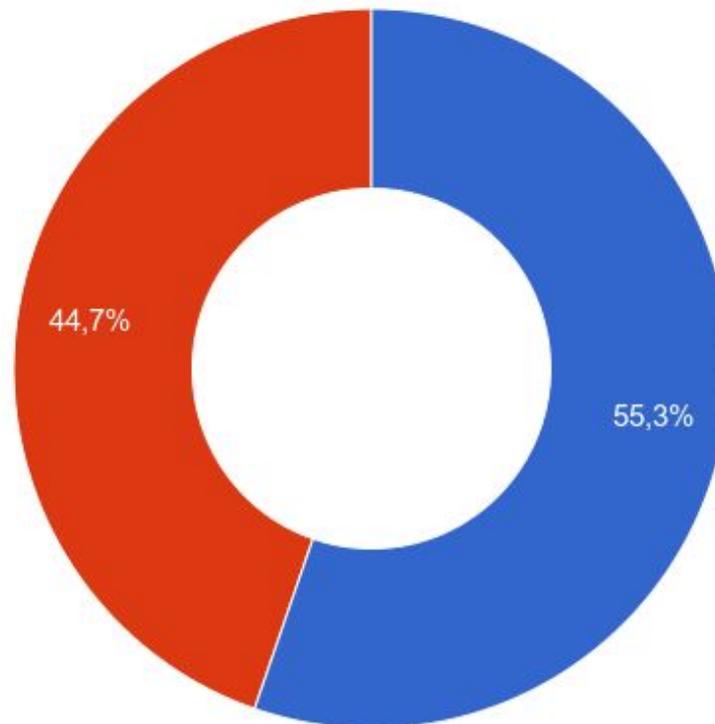
Testing (pruebas): 11



Estadísticas: Legacy vs 100% Chefizadas

100% "Chefizadas": 120

Legacy: 97

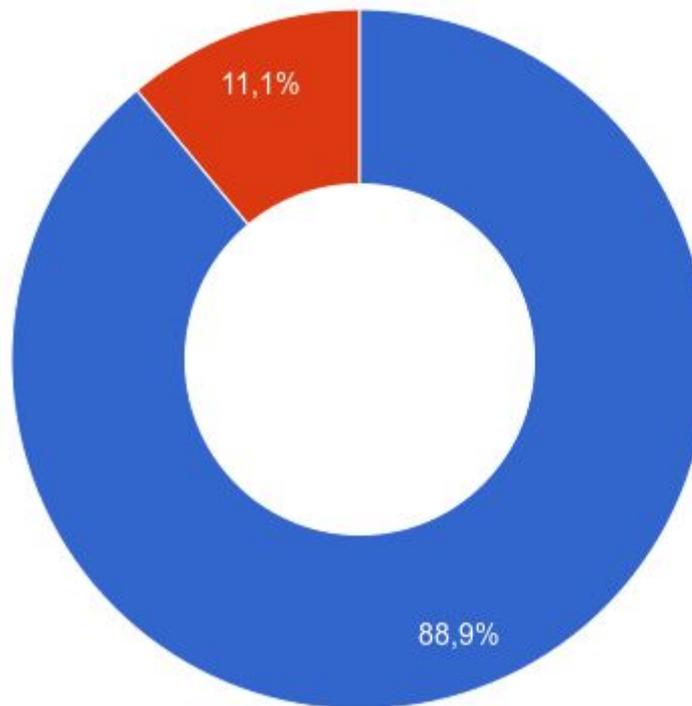


 Chefizada al completo

 Legacy

Manual: 24

Automática: 193



● Ejecución Automática
● Ejecución Manual

- # Queda mucho por hacer
- # Trabajar en el convenio sobre la configuración, documentando unas “Buenas prácticas”
- # Hincarle el diente a Windows
- # Incorporar Chef de una manera sostenible en la infraestructura y servicios
- # Trabajar en la integración de los técnicos y contagiar a más gente
- # Seguir aprendiendo :)

GRACIAS

FIN

La cocina del STIC

?

Alberto Lorenzo Pulido

- alorenzo@ull.edu.es
- <https://github.com/tic-ull>
- <https://github.com/8la>
- <https://twitter.com/8la>

Provisión automatizada de
servidores y servicios TIC