

# Análisis de Herramientas para la Automatización de la Administración de Configuración de la Infraestructura

Kenneth Díaz Siles y Marlon Ramírez Castillo

Escuela de Ingeniería,  
Universidad Latinoamericana de Ciencia y Tecnología,  
ULACIT, Urbanización Tournón, 10235-1000  
San José, Costa Rica  
[kdiazs266,mramirez932]@ulacit.ed.cr  
<http://www.ulacit.ac.cr>

**Resumen** Los administradores de redes y sistemas se enfrentan con el problema de gestionar la infraestructura tecnológica que crece día con día. El objetivo de esta investigación es obtener las herramientas adecuadas que puedan ayudar a la administración que configura la infraestructura tecnológica. Esto conlleva que los intendentes se deben enfrentar a otro reto: decidir cuál herramienta pueden utilizar de acuerdo con las características de su plantel y el problema particular al que se enfrentan. Los mecanismos que fueron analizados en este trabajo son Puppet, Chef, Ansible y SaltStack, que fueron analizados con base en una revisión bibliográfica de manuales técnicos, tutoriales, libros y los sitios web de cada una de las herramientas.

**Keywords:** Administración de la configuración, automatización de la gestión, infraestructura tecnológica

## 1. Introducción

Las operaciones de las organizaciones se apoyan en un gran número de sistemas de software, hardware y comunicaciones, con independencia, respecto al tipo de actividad a la que se dedican. En este contexto, es un requerimiento fundamental que el diseño de la infraestructura tecnológica, que integra a todos estos sistemas, sea escalable y aplicar estándares que den paso a una gestión eficaz. Esto es un factor de gran importancia al considerar las necesidades de rendimiento, disponibilidad y capacidad de respuesta que usualmente son demandados, así como permitir el crecimiento de las operaciones, sin la necesidad de efectuar cambios que comprometan el funcionamiento de la organización.

En la actualidad, las organizaciones operan de forma distribuida en diferentes países y regiones, y cuentan con cientos o miles de dispositivos. Esto conlleva que los administradores de redes y sistemas se enfrenten, día a día, al reto de gestionar la infraestructura tecnológica, y presenta distintos escenarios que, por lo general,

cuentan con menos recursos de los que requieren y demandan capacidad de respuesta en poco tiempo.

De acuerdo con lo anterior, este trabajo de investigación tiene por objetivo diseñar un modelo de gestión de automatizada de la configuración para cientos o miles de dispositivos. Para ello, toma como base las herramientas que se especializan en apoyar una actividad o tarea concreta, y por lo tanto, es necesario integrar un conjunto de ellas para aprovechar sus fortalezas y ofrecer mejores soluciones a los administradores de redes y sistemas.

Con el propósito de determinar los beneficios que ofrecen las principales herramientas de software libre, este trabajo lleva a cabo una revisión detallada de un grupo clasificado de estos utensilios para identificar sus características, similitudes y diferencias, y a partir de ello, brindar un criterio a las soluciones que ofrecen cada una de ellas, en el mercado empresarial.

Hoy muchas compañías se caracterizan por brindar servicios a clientes alrededor del mundo, esto porque los flujos del mercado actual las obligan a ello; y con esto surge la necesidad de expandir sus operaciones a distintas ubicaciones geográficas, con lo cual es necesario implantar una red unificada que contenga todos los dispositivos, funcionando en armonía, con el fin de optimizar los procesos internos. Por dicha situación, se hace indispensable la gestión de la infraestructura, por medio de herramientas que permitan de forma óptima, a los cientos o miles de dispositivos que componen una red empresarial mundial.

Por lo anterior, es necesario renovar ciertas tendencias de gestión tecnológica, pues si bien es cierto, un script desarrollado en Python<sup>1</sup>, Bash<sup>2</sup> o Powershell<sup>3</sup>, cumple una función de gestión y configuración para un sistema específico, no es suficiente para gestionar una red empresarial completa, impactando las labores de la organización. Es por esto que surge el perfil profesional del DevOps<sup>4</sup>, recurso humano cuya función específica es la de gestionar de forma ágil, sencilla y eficaz los dispositivos de una red empresarial por medio de herramientas que buscan automatizar los procesos de monitoreo, configuración, actualización y soporte en general.

Esta tendencia en grandes compañías es uno de los principales factores que ha motivado esta investigación, ya que se desea descubrir las razones por las cuales una herramienta de este tipo facilita los procesos empresariales, a bajo costo y brindando una renovada imagen a las tecnologías de información. Este documento abordan los siguientes temas: en la sección 2 se describe los antecedentes que da inicio a la investigación de las herramientas de la administración de la configuración; Sección 3 muestra conceptos básicos acerca de administración de la configuración así como sus componentes; En la sección 4 se realiza una descripción de cada una de las herramientas de automatización de la infraestructura.

---

<sup>1</sup> Lenguaje de programación del paradigma Scripting.

<sup>2</sup> Intérprete de comandos de los sistemas operativos Linux.

<sup>3</sup> Intérprete de comandos de los sistemas operativos Microsoft, el cual se ejecuta por medio de la línea de comandos.

<sup>4</sup> Operadores de desarrollo, los cuales utilizan material administrativo en la configuración.

tura así como sus características; Sección 5 muestra la evaluación efectuada a cada una de las herramientas de administración de la configuración; sección 6 se puede observar la discusión y resultados del documento de investigación y finalmente en la sección 7 se presentan las conclusiones.

## 2. Antecedentes

Los avances en las tecnologías de la comunicación han facilitado la conectividad entre las localidades y centros de operación de las empresas (e.g., distribuidas en diferentes países y regiones). Esto permitió el acceso a la información que se produce en tiempo real en cada localidad y generó la necesidad de contar con equipos y sistemas para analizar y producir conocimiento a partir de esa información. Lo cual ha tenido como consecuencia el incremento en el número de dispositivos que conforman la infraestructura de las organizaciones. A ese incremento se debe agregar la aparición de dispositivos móviles (e.g., celulares, tabletas, portátiles) que acceden el correo electrónico, informes y aplicaciones empresariales e Internet en general, y por lo tanto inciden en las necesidades de gestión de la infraestructura, para brindar una red que sea capaz de transportar gran cantidad de información en poco tiempo.

La disponibilidad de las funciones y procesos de negocio, son altamente dependientes de los sistemas de información y de los recursos tecnológicos, en los cuales se apoyan para la ejecución de los procesos y flujos de mercado, que requieren sistemas de alta disponibilidad y estándares de seguridad, y son este, uno de los factores importantes para el crecimiento continuo de las organizaciones. Además, la información se convierte en un activo de valor para las compañías, lo cual requiere de medios seguros para su correcto manejo, medios actualizados que han sido gestionados de forma efectiva, por manos profesionales que garanticen la veracidad de esta información.

Como consecuencia, se requiere de herramientas para gestionar de forma centralizada y efectiva la infraestructura tecnológica, y así obtener el máximo rendimiento de cada uno de los recursos disponibles que requieren estar actualizados, si se tiene como objetivo primordial una red estable, segura y eficaz, al igual que los sistemas de datos, software y hardware en general. Esto implica que se torna más complicado la gestión eficiente de los sistemas y los dispositivos que componen una red empresarial.

Si bien es cierto, que se busca que la infraestructura de una empresa, sea una plataforma unificada y centralizada, por lo general, esto no se cumple. Pero en este contexto herramientas como System Center Configuration Management de Microsoft (Microsoft, 2014) o un administrador LDAP<sup>5</sup> (OpenLDAP, 2012) no son suficientes, debido a que las infraestructuras en la actualidad se componen de cientos o miles de dispositivos se requiere de herramientas más complejas que permitan centralizar la gestión de la configuración.

Por lo anterior, surgieron herramientas como Puppet (Uphill y Arundel, 2015), la primera permite llevar a cabo la configuración de arquitecturas mul-

---

<sup>5</sup> Protocolo de acceso directo de peso ligero.

tiplataformas, usando un modelo de cliente-servidor. Esta herramienta soporta lenguajes de programación como Python, Bash, Ruby<sup>6</sup> y Powershell de Microsoft para gestionar los servidores y dispositivos que basados en distribuciones de Linux, Microsoft o Unix.

Puppet marcó una tendencia en el mercado, la cual dio paso al desarrollo de otras herramientas similares, con funciones específicas, que brindan mayor apoyo a la gestión tecnológica. Algunos de esos mecanismos son Chef (Marschall, 2013), SaltStack (Myers, January 2015) y Ansible (Hochstein, 2015): cuatro elementos que facilitan la administración de los dispositivos de una infraestructura tecnológica, de forma centralizada, mediante instrucciones por línea de comando o un script.

En la actualidad este tipo de herramientas son utilizadas para la gestión de su infraestructura tecnológica por empresas de gran tamaño como:

1. **Componentes Intel de Costa Rica:** Esta compañía tiene un equipo de trabajo dedicado a manejar la infraestructura tecnológica con SaltStack y Chef, de acuerdo con las necesidades de la organización. Cada miembro del equipo es especialista en el uso de estas herramientas y se dedica de forma exclusiva a la automatización de procesos que involucran la infraestructura.
2. **Hewlett-Packard Costa Rica:** Esta empresa cuenta con laboratorios de pruebas y producción que son administrados por herramientas que se encargan de generar alertas de eventos en el laboratorio. Estos acontecimientos son gestionados mediante la aplicación de métodos automáticos de recuperación de la configuración.

### 3. Conceptos básicos

La Administración de la Configuración se fundamenta en la definición de un modelo de configuración genérico (e.g., plantilla) que se utiliza como base para definir la forma detallada de un grupo de dispositivos o sistemas con características similares. Una vez que la estructura detallada ha sido definida, se puede aplicar a cientos o miles de dispositivos.

Los cuatro pilares fundamentales de la Administración de la Configuración son los siguientes:

1. Identificación de la configuración: Permite identificar una serie de elementos, los cuales van a cumplir la función del proceso base, para establecer la clasificación del producto. Este proceso base de identificación debe ser único para poder facilitar la gestión de la forma del plan a desarrollar.
2. Control de la configuración: Permite gestionar de forma óptima el ciclo de vida del producto durante su función, se obtiene una perspectiva acerca de los activos, los cuales se tiene control y de qué forma, muestra una gestión acerca de los cambios que sufre el producto durante sus años de vida útil.

---

<sup>6</sup> Lenguaje de programación interpretado orientado a objetos.

3. Estado de la configuración: El estado de la configuración brinda una mayor expectativa acerca de los cambios que se realizan en la gestión de la configuración, de esta manera podemos determinar cuáles elementos han cambiado y de qué manera, se puede mencionar los cambios que serán aprobados, los cambios que están pendientes por realizarse.
4. Auditoría de la configuración: Establece una línea de seguimiento para la gestión de la configuración del producto, es necesario realizar una verificación para confirmar que la distribución va de acuerdo al diseño seleccionado, y determinar que los elementos que componen el producto funcionen de la manera adecuada, esto permite tener una verificación previa a la puesta en marcha del producto final, de igual importancia, se debe asegurar que el proceso de configuración se realiza de la forma correcta.

### 3.1. Administración de la Configuración de la Infraestructura Tecnológica

La Administración de la Configuración de la Infraestructura Tecnológica tiene relación con los procesos que permiten automatizar tareas asociadas con la administración de la infraestructura empresarial haciendo uso de herramientas especializadas.

Para ello, es que se tiene un conjunto de herramientas para realizar procesos de configuración única, y así establecer un control sobre cada uno de los activos que integran una arquitectura de red empresarial. Es por esto, que cada día se dificulta más la intervención humana para administrar, actualizar o incluso reparar un servidor, un enrutador<sup>7</sup>, o cualquier dispositivo de red. Hoy en día estos procesos se realizan de forma automática, mediante una serie de herramientas que permiten automatizar cada uno de los procesos mencionados anteriormente.

Los mecanismos de automatización de infraestructura se caracterizan por el uso de líneas de comandos, implantados en scripts para ejecutar acciones, las cuales se utilizan en distintos ambientes, distintas versiones de sistemas operativos, distintas plataformas y dispositivos de red, que son componentes de un modelo empresarial, el cual debe ser gestionado de forma centralizada, a bajo costo y con el mejor manejo de los recursos (humanos y tecnológicos).

## 4. Herramientas de Automatización de Infraestructura

Las herramientas de automatización para administrar las infraestructuras empresariales, surgen a partir de las necesidades de negocio, motivadas por la expansión de operaciones en distintas áreas geográficas. Por ello, herramientas como Puppet, Chef, SaltStack y Ansible fueron creadas con el fin de brindar apoyo a la administración y configuración de la infraestructura mediante un esquema centralizado. Este tipo de herramientas simplifican la administración de cientos o miles de dispositivos interconectados que intervienen en los procesos del núcleo de negocio de las compañías e impactan sus operaciones diarias.

---

<sup>7</sup> Dispositivo de red que se encarga de gestionar el tráfico de red.

Es por esta razón que este trabajo de investigación resalta y compara las principales fortalezas de las cuatro herramientas seleccionadas, y discute los aportes que pueden realizar a la administración de la infraestructura. De acuerdo con lo anterior, las siguientes secciones presentan un resumen de las principales características de cada mecanismo.

#### 4.1. Puppet

Puppet fue la primera herramienta de automatización de infraestructura desarrollada por Puppet Labs. Esta herramienta fue desarrollada en el lenguaje de programación Ruby, permite gestionar cambios de forma sencilla, con el código fuente disponible en la web, listo para descargarlo, editarlo y ajustarlo a las necesidades de los clientes.

Tiene como fin facilitar las operaciones de las oficinas de Tecnologías de Información, ya que busca optimizar procesos de administración de la configuración de forma rápida y sencilla, con cualidades como la de ajustarse a infraestructuras escalables y en pleno desarrollo, con gestión vía terminal o vía interfaz gráfica web.

Esta herramienta se integra con otras, como lo son Amazon Web Service (Service, 2006), Vmware (VMware, 1998) y Google Compute Engine (Engine, 1998), entornos de nube, los cuales han sido una tendencia para las organizaciones que confían más en los entornos tecnológicos. Actualmente la cartera de clientes de Puppet Labs es muy selecta, a saber Twitter (Twitter, 2007), Vmware, RedHat Linux y Paypal (PayPal, 1998), las cuales tienen un sistema Puppet dedicado, de acuerdo al núcleo de negocios de cada una de ellas.

Puppet contiene su propio lenguaje estructurado conocido como Manifiestos, y agrupados son conocidos como Recursos, los cuales componen el núcleo de la herramienta Puppet. Un conjunto de Recursos contienen bloques de código que ejecutan tareas específicas, esta agrupación se conoce como Módulos, los cuales serán las unidades que se ejecutan para las distintas tareas. Estos módulos son almacenados en el servidor central llamado PuppetMaster.

Cada uno de los Módulos que serán ejecutados en los nodos clientes, estos módulos deben ser compilados del lado de servidor, viajar cifrados a través de la red empresarial y ser descifrados por un componente llamado Catálogo, presente en los clientes de Puppet, que envía la información descifrada a los Aplicadores, que se encargan de ejecutar los módulos en los clientes de Puppet.

La arquitectura que utiliza Puppet es la cliente-servidor, pero Puppet labs la rebautizó como modelo Master/Agent. El PuppetMaster es el servidor central que contiene el núcleo de las funciones de configuración unitarias (conocidas como recursos); y el Agente que es el cliente que se encarga de gestionar los procesos en los nodos remotos. Este tipo de estructura está diseñado para trabajar como un servicio que se ejecuta en cada nodo. Existe otro paradigma, el cual es una variante al modelo Master/Agent, y es conocido como Puppet Apply, tiene una metodología de trabajo más enfocada a soluciones en la nube, y realiza las configuraciones a través de terminales y comandos, sin esperar un valor de retorno. Este concepto tiene un impacto en la velocidad desarrollada, ya

---

**Algoritmo 1** Estructura de un módulo en Puppet

---

```
1: MODULE_PATH/
2:   downcased_module_name/
3:     files/
4:     manifests/
5:       init.pp
6:     puppet/
7:       parser/
8:       functions
9:     provider/
10:    type/
11:    facter/
12:    template/
13:    README
```

---

que es más veloz y no requiere una validación concreta de la correcta ejecución, sin dejar de lado, que brinda mayores niveles de seguridad, ya que se encarga de encapsular los recursos que serán ejecutados por los catálogos en los nodos remotos.

Ambos paradigmas, utilizan una serie de llaves públicas y privadas, para accionarlos procesos entre los PuppetMaster y los nodos remotos. Estas llaves serán generadas por los administradores de sistemas, los cuales se encargan de gestionar los servidores y sistemas operativos, o bien dan paso a la nueva figura profesional emergente, el DevOps, que se encarga de facilitar las herramientas que se analizan en este documento.

Actualmente se cuenta con dos versiones de Puppet, las cuales se mencionan a continuación:

- Puppet Open Source: Esta versión está diseñada para soluciones pequeñas en ambientes que no requieren procesos muy grandes, y que son sostenibles a bajo costo.
- Puppet Enterprise: Esta solución brinda una serie de funcionalidades que solventan problemas de integración cuando la infraestructura tiene topologías más complejas, y soporte respaldado por la empresa desarrolladora, Puppet Labs.

Puppet es una de las herramientas de este tipo más completas que existen en el mercado, esto la convierte en una herramienta compleja, que requiere una curva de aprendizaje prolongada, horas de dedicación y conocimientos técnicos en programación, redes e infraestructura en general. Estos rubros la definen como un mecanismo que requiere conocimientos en distintos campos, en caso de ser aprobada se selecciona como gestor de infraestructura.

## 4.2. Chef

Chef es una herramienta desarrollada por la comunidad Opscode en el 2008, en sus inicios, el cual es un instrumento que se desarrollada para solventar una

problemática de gestión en los Centros de Datos de Microsoft y de Amazon, pero evolucionó para cubrir distintas plataformas y paradigmas. Elaborado en los lenguajes de Programación Ruby y Erlang, permite utilizar estos dos lenguajes de programación para gestionar los procesos en los nodos y los Workstations.

Esta herramienta se caracteriza por ser multiparadigma, además es escalable y flexible, brinda más seguridad que muchas de las herramientas existentes en el mercado. La principal fortaleza de esta herramienta es que se desarrolló para gestionar ambientes virtuales, lo cual facilitó la escalabilidad a entornos en la nube. Dado esta situación no sorprende la capacidad de integración con Amazon Web Service, Windows Azure (Azure, 2010), HP Cloud (Cloud, 2012), Rackspace (Rackspace, 1998), OpenStack (OpenStack, 2012) y Google Compute Engine.

Dado esta serie de características y facilidad de adaptación ante distintos escenarios, la cartera de clientes es muy amplia, a saber los usuarios de mayor renombre son los siguientes: Facebook, LinkedIn (Linkedin, 2003) y Youtube, entre otros.

Los principales componentes de Chef son los siguientes:

- Chef Server: Servidor central de gestión.
- Workstation: Estaciones de trabajo dedicadas a interactuar con el Chef Server y con los nodos, es decir, es el intermediario entre los Nodos y el servidor central. Contiene los repositorios de Chef para la gestión de los Nodos.
- Nodos: Son los elementos que componen la infraestructura tecnológica.

Cada uno de los componentes de Chef, trabajan de forma armónica formando una jerarquía estructural, gestionando los componentes de la infraestructura por medio de Chef Cookbooks (unidad fundamental de configuración) donde son gestionados por los Workstation; cada WorkStation contiene el repositorio de Chef con los módulos instalados, los cuales son enviados y ejecutados en los nodos, que son los componentes más primitivos de la jerarquía.

Existen varios tipos de Chef Server, a saber los siguientes:

- Hosted Chef Server: Servidor que tiene el respaldo de la comunidad, es la principal herramienta de uso comercial, ya que brinda una serie de soluciones a distintos problemas de forma predeterminada, utilizando una serie de módulos existentes en los repositorios.
- Private Chef: Al igual que el Hosted Chef Server, posee el respaldo de la comunidad, pero además de ello, esta herramienta se ajusta a las necesidades específicas de los clientes..
- Open Sorce Chef Server: Esta opción es la versión gratis, y la que es utilizada por muchas pequeñas empresas, es de código abierto y permite gestionar una serie de módulos de la herramienta, con ciertas limitantes.

A su vez, los nodos se pueden clasificar de la siguiente forma:

- Nodos físicos: Cualquier dispositivo físico, enrutadores, conmutadores<sup>8</sup>, estaciones de trabajo, tabletas, etc.

<sup>8</sup> Dispositivo de red que se encarga de interconectar dispositivos de red.



- Nodos virtuales: Dispositivos virtuales.
- Nodos instancias de nube: Componentes que se encuentren en la nube.

Chef es una de las herramientas que aprovecha de mejor forma los recursos que posee, utilizando una serie de herramientas software libre, las cuales permiten gestionar de forma sencilla, los componentes de una infraestructura, con una visión a las tendencias actuales, como lo son las tecnologías en la nube, brindando una serie de ventajas que la convierten en una de las mejores opciones del mercado actual, ya que brinda una serie de ventajas que permite desarrollar una serie de soluciones sostenibles en los distintos campos de Tecnologías de Información.

### 4.3. Ansible

Ansible es una herramienta desarrollada por AnsibleWorks, inicialmente Ansible se desarrolló en el 2012 como una herramienta de administración de entornos de red, pero en los últimos años evolucionó en una herramienta de soluciones de automatización de infraestructuras de red.

Ansible provee servicios a cientos de usuarios alrededor del mundo, algunos de los cuales se menciona en su portal y se destacan los siguientes: Twitter, Nasa, Verizon Wireless (Wireless, 2000) y GoPro (GoPro, 2002).

Ansible se caracteriza por 4 premisas las cuales marcan sus funciones y las cuales podemos mencionar a continuación:

- Simplicidad: Utiliza un lenguaje de programación simple el brinda una facilidad para desarrollar scripts<sup>9</sup> con procesos de gestión.
- Agilidad: Provee una configuración rápida en los nodos centrales, no es necesario configurar los nodos, lo cual permite disminuir tiempos de ejecución y configuración.
- Poder: Esta herramienta despliega una variedad de funciones de configuración, lo que permite tener un mayor alcance a sus módulos.
- Eficiencia: Por su modelo de desarrollo, Ansible no requiere instalación de software en ninguno de los nodos clientes, lo que le permite desplegar un método de comunicación más simple y ligero.
- Seguridad: Al no requerir software específico en ninguna de las estaciones clientes, Ansible utiliza un método de conexión seguro entre el servidor central y los nodos, el cual se denomina intérprete de órdenes seguras.

Ansible dispone de dos soluciones, la primera es a base de línea de comandos (CLI), la cual genera una interacción con un sistema informático (Software); o bien utiliza Ansible Tower, la cual funciona como una interfaz gráfica de usuario, la cual ofrece un tablero donde se pueden realizar diferentes tareas de gestión y monitoreo.

Ansible se implementa bajo una arquitectura de nube, donde el servidor central ejerce control sobre otros activos que se encuentran en la infraestructura

---

<sup>9</sup> Línea de comandos que representa una orden.

de red, el servidor distribuye órdenes y solicitudes a las computadoras clientes, los cuales se encargan de escuchar al servidor principal e interpretar las órdenes que recibe del control primario. El controlador, sabe cuáles son sus nodos gracias a un inventario generado previamente.

Dentro de Ansible se localizan los Playbooks, estos se definen como una herramienta de configuración simple la cual se desarrolla por medios de una línea de comandos, con el fin de administrar los nodos de forma remota. Los Playbooks se caracterizan por integrar un lenguaje de programación propio de la herramienta, para gestionar los procesos encapsulados en módulos, los que han sido desarrollados en distintos lenguajes de programación, por ejemplo Ruby<sup>10</sup>, Perl<sup>11</sup> o Python<sup>12</sup> entre otros lenguajes de este tipo.

#### 4.4. SaltStack

SaltStack es una herramienta de automatización, desarrollada por SaltStack Enterprise en el año 2011, desarrollada con el objetivo facilitar la administración de sistemas de información integrados por cientos o miles de dispositivos.

SaltStack utiliza una licencia Apache perteneciente a Fundación de Software Apache, utiliza Python como lenguaje de programación para la gestión de configuración.

SaltStack ofrece soluciones de automatización para los siguientes tipos de clientes:

- ITOps<sup>13</sup>: SaltStack se encarga de la administración de las arquitecturas compuestas por cientos o miles de dispositivos.
- DevOps: La herramienta brinda soluciones a la medida, las cuales van a brindar soluciones de gestión óptimas.
- CloudOps<sup>14</sup>: SaltStack tiene la capacidad de adaptarse a las tendencias que ofrece la nube.

Las principales características de SaltStack son las siguientes:

- Velocidad: Permite brindar soluciones de gestión de forma ágil a bajo costo.
- Escalabilidad: Capacidad de crecimiento y soporte en múltiples plataformas, para gestionar de forma sencilla.
- Flexibilidad: Capacidad de adaptarse a distintas arquitecturas y modelos de desarrollo empresarial, desde una red sencilla hasta soluciones para centros de datos.

<sup>10</sup> Lenguaje de programación interpretado orientado a objetos.

<sup>11</sup> Lenguaje de programación interpretado del paradigma scripting.

<sup>12</sup> Lenguaje de programación orientado a objetos que utiliza múltiples estilos de programación.

<sup>13</sup> Operadores de tecnologías de información

<sup>14</sup> Operadores de nube.

SaltStack trabaja con un modelo cliente-servidor, donde el servidor central se conoce como Master y los nodos se llaman Minions, estos últimos son gestionados por medio de línea de comandos, donde se interpreta y genera las soluciones de administración de dispositivos remotos. El Master identifica a los Minions por medio de una etiqueta de identificación única, la cual viaja a través de Secure Shell <sup>15</sup>.

Los Granos en SaltStack son una estructura que brindan una descripción detallada de cada una de los Minions<sup>16</sup>, entre la información que almacenan están la familia pertenecen, sistemas operativos. Para gestionar los granos, SaltStack cuenta con una serie de módulos precargados, y cada uno de los elementos que componen la infraestructura.

## 5. Evaluación de las herramientas de Administración de la Configuración

### 5.1. Criterios de evaluación

Las herramientas de automatización de la infraestructura seleccionadas, son cuatro de las más populares en el mercado. Cada una posee una serie de características que las identifican como únicas para ciertos procesos, no obstante, y que también permiten realizar comparaciones y así determinar qué tan factible es la interoperabilidad y la coexistencia de dichas herramientas, entre otros rubros que permitan desarrollar la administración de la infraestructura de la mejor forma posible.

Antes de brindar un criterio de selección, se debe determinar el escenario en el cual se implantará y ejecutarán las herramientas que están en análisis, para brindar un mejor criterio de selección, aprovechando la mejor forma y fortalezas de cada una de ellas.

Los criterios a evaluar serán los siguientes:

- Características generales de las herramientas
- Compatibilidad con distintos sistemas operativos
- Integración con otras herramientas
- Seguridad de las herramientas
- Rendimiento de las herramientas

### 5.2. Características generales de las herramientas

Las herramientas de Administración de la Configuración de la Infraestructura seleccionadas comparten una serie de características y atributos entre sí, las cuales permiten realizar procesos de forma similar, tanto en ejecución como en rendimiento.

<sup>15</sup> Protocolo intérprete de red seguro.

<sup>16</sup> Denominación utilizada para referirse a los nodos clientes

Entre las características que comparten las herramientas seleccionadas, que son desarrolladas bajo los ideales de las licencias de software libre, lo cual permite utilizar, modificar y distribuir el producto bajo ciertos lineamientos, que permiten al usuario final tener libertad en este uso.

Además, los mecanismos seleccionados han sido desarrollados con lenguajes de programación modernos, del paradigma de Scripting, como lo son Python y Ruby. Cada una de las herramientas tiene su repositorio de código fuente en línea, de acceso público para que los usuarios tengan la opción de ajustar la sus necesidades, cuando así lo requieran.

Puppet y Chef comparten un patrón de arquitectura del modelo Cliente/Servidor, mientras que Ansible y SaltStack comparten un prototipo de arquitectura de la Nube, para mejorar los tiempos de ejecución y administración. Es por ello que las dos primeras debieron evolucionar al modelo de nube.

Las herramientas en análisis, tienen un conjunto de librerías llamadas Módulos, que tienen como función gestionar la infraestructura de forma sencilla. Cada una de ellas gestiona los Módulos de forma peculiar, e incluso existe la posibilidad de integrar Módulos de una herramienta en otra, siempre y cuando compartan el lenguaje de programación en el que se desarrolló, por ejemplo, Puppet podrá integrar módulos de Chef (desarrollados en Ruby) o Ansible podrá ejecutar módulos de SaltStack (desarrollados en lenguaje Python), basta con descargar el código fuente disponible (SaltStack, 2015) de las herramientas analizadas.

Las principales funciones que gestionan los módulos son las siguientes:

- Administración de servicios.
- Administración de ficheros.
- Administración y configuración de sistemas.
- Administración y configuración de respaldos.

Las características que se han mencionado anteriormente junto con otras no mencionadas se enlistan en el Cuadro 1., el cual indica una perspectiva general de cada una de las herramientas en análisis.

Atributos	Puppet	Chef	Ansible	SaltStack
Lenguaje	Ruby	Ruby	Python	Python
Tipo de licencia	Apache y GPL	Apache	GPL	Apache
Arquitectura	Cliente-servidor y nube	Cliente-servidor y nube	Nube	Nube
Escalable	Si	Si	Si	Si
Multi-plataforma	Si	Si	Si	Si
Encriptación	Si	Si	Si	Si
Interfaz gráfica	Foreman	Knife	AnsibleTower	Halite
Primer lanzamiento	30-08-2005	15-01-2009	20-03-2008	17-03-2011
Ultimo lanzamiento	26-03-2015	30-03-2015	28-04-2015	01-17-2015

**Cuadro 1.** Atributos y características de las herramientas de automatización.

### 5.3. Compatibilidad con distintos sistemas operativos

Las organizaciones de esta era poseen una serie de dispositivos como servidores, enrutadores, conmutadores, estaciones de trabajo y dispositivos móviles, que resulta casi imposible administrar una red empresarial asociada a una tecnología específica. Por lo anterior es que las herramientas deben ser compatibles con todos los sistemas operativos existentes en el mercado.

Las herramientas de automatización de infraestructura como Puppet, Chef, Ansible y SaltStack despliegan sus funciones en un gran número de plataformas y de sistemas operativos que se utilizan en cientos de organizaciones. Además, tienen la capacidad de administrar la infraestructura, sin importar la arquitectura y el modelo de negocio de la red existente, ya que son adaptables al entorno de trabajo y por ende a cada caso de uso.

Como dato adicional, se incluye la tabla comparativa con algunas de las plataformas soportadas por cada herramienta, para complementar la información indicada anteriormente.

Tipos de Plataformas	Puppet	Chef	SaltStack	Ansible
Windows	Si	Si	Si	Si
IBM AIX	Si	Si	Si	Si
HP UX	Si	Parcial	Si	Si
Linux	Si	Si	Si	Si
Mac OS X	Parcial	Si	Si	Si
Solaris	Si	Si	Si	Si

**Cuadro 2.** Herramientas de automatización y sistemas operativos.

Anteriormente se dio énfasis en sistemas operativos utilizados en servidores y estaciones de trabajo, pero también se debe mencionar que este tipo de equipo son capaces de administrar dispositivos de red, como lo son enrutadores y conmutadores de distintas marcas, un ejemplo claro de ello son los dispositivos de la marca Cisco, los cuales tienen como sistema operativo IOS, con la facilidad de ser administrados y configurados por Puppet y Chef, entre otras herramientas.

Además, han evolucionado a otras plataformas y tendencias tecnológicas, tanto así que hoy en día sistemas operativos para dispositivos móviles como Android y iOS, de Google y Apple respectivamente son soportados para ser administrados con alguna de las herramientas analizadas, y así integrar y centralizar la administración de una red corporativa en su totalidad.

### 5.4. Integración con otras herramientas

Como una necesidad de negocio, este tipo de herramientas deben ser capaces de integrarse con otras para formar sistemas complejos, este análisis abarca desde herramientas de monitoreo, hasta gestores de máquinas virtuales, esto con el fin de mejorar la oferta ante los usuarios finales.

Esta evolución ha llevado a los instrumentos a integrar en su propia arquitectura, una serie de módulos que les permiten administrar y monitorear toda la infraestructura, un ejemplo los módulos de Nagios<sup>17</sup>, utensilios de monitoreo de red, el cual posee módulos en las cuatro herramientas analizadas, permitiendo integrar para la gestión automática, de los procesos de red. Este tipo de integración permite desarrollar sistemas de monitoreo complejos, capaces de realizar procesos de recuperación y configuración de forma automática, y disminuye la intervención humana para acciones de administración.

También se pueden integrar con servicios de nube, para administrar de forma automática los procesos que conlleva una instancia, y así facilitar las tareas en un centro de datos por ejemplo. De igual forma administrar los procesos de una máquina virtual, entre lo que podemos mencionar la gestión de puertos de un conmutador virtual, que poseen los motores de virtualización, dirección de los respaldos de las máquinas virtuales, entre otros.

Además, este tipo de herramientas, por su naturaleza y arquitectura, son capaces de realizar procesos de administración de la configuración en general, ejemplo de ello es la capacidad de integrarse con Team Foundation Server<sup>18</sup> y Git<sup>19</sup>, por ejemplo, se pueden realizar mantenimientos y procesos de recuperación en un repositorio, además de actualizar todos los repositorios hijos de un árbol Git por ejemplo.

La tabla que se muestra a continuación, complementa la información que se menciona en este enunciado.

Tipos de Plataformas	Puppet	Chef	SaltStack	Ansible
Amazon Web Service	Si	Si	Si	Si
Rackspace	Si	Si	Si	Si
OpenStack	Si	Si	Si	Si
Vmware	Si	Si	Si	Si
Citrix	Si	Si	Si	Si
Google Compute Engine	Si	Si	Si	Si
Team Foundation Server	Si	Si	Si	Si
Git	Si	Si	Si	Si
Nagios	Si	Si	Si	Si
Wireshark	Si	Si	Si	Si

**Cuadro 3.** Integración con otras herramientas.

Si bien es cierto la naturaleza de estas herramientas es administrar infraestructura, su concepto y aplicabilidad permiten desempeñar tareas en otras áreas de la informática, por ejemplo el monitoreo de redes y la gestión de repositorios de código, entre otros procesos. Solo basta buscar el módulo respectivo e

<sup>17</sup> Software de monitoreo de redes.

<sup>18</sup> Herramienta de gestión de código fuente, Microsoft.

<sup>19</sup> Sistema de controlador de versiones para el desarrollo de software.

implantarlo y si no existe, se puede desarrollar el módulo con alguno de los lenguajes de programación que soportado; solo se requiere conocimiento en la arquitectura de la herramienta, conocimientos en programación y dedicar tiempo a la investigación para obtener resultados sorprendentes en nuevas áreas de investigación.

### 5.5. Seguridad de las herramientas

Uno de los rubros por el cual se selecciona cualquier tipo de herramienta en general, es la seguridad. Se busca que sea robusta, con altos estándares que permitan garantizar la efectividad de los procesos que realiza.

**Puppet** se encarga de la seguridad por medio de los siguientes complementos de seguridad:

- **SSL de seguridad:** Utiliza certificados públicos y privados para generar conexiones SSH seguras.
- **AES+RSA de seguridad:** Utiliza llaves públicas y privadas para garantizar la seguridad. Garantiza la seguridad de las estructuras existentes en Puppet, además de la comunicación Cliente-Servidor.
- **Seguridad ActiveMQ:** Utiliza protocolos de seguridad para la autenticación de cada uno de los nodos al servidor central y todos los dispositivos intermedios, cuando así lo requieran.

A continuación se indica el proceso que sigue Puppet para realizar conexiones y ejecuciones seguras a través de las redes.

---

#### Algoritmo 2 Proceso de Autenticación y Auditoría en Puppet

---

- 1: Conexión del cliente por medio de credenciales
  - 2: Conexión con el Middleware de Red
  - 3: Autenticación y Autorización de la red
  - 4: Conexión a los nodos por medio de credenciales
  - 5: Autorización SimpleRPC
  - 6: Auditoría SimpleRPC
- 

A su vez, **Chef** maneja la seguridad de forma muy similar a Puppet, utiliza certificados de seguridad de la capa de transporte (SSL) para realizar una conexión oportuna entre el servidor Chef y los Nodos. Chef cuenta con una serie de protocolos SSL<sup>20</sup> que permiten administrar la seguridad de dicha conexión de forma ágil y sencilla. Utiliza claves cifradas, hasta un nivel AES256<sup>21</sup> con ECDHE<sup>22</sup>.

<sup>20</sup> Secure Sockets Layer.

<sup>21</sup> Advanced Encryption Standard.

<sup>22</sup> Elliptic Curve Diffie-Hellman Exchange

De la parte del cliente se puede mencionar que existen certificados intermedios que realizan la primera parte de la conexión, dejando para los procesos internos con los módulos, los procesos de generación y degeneración de las llaves públicas y privadas. Todos estos procesos son realizados por el cliente de Chef llamado Knife.

**Ansible** utiliza un protocolo de intérprete de órdenes seguras (Secure SHell), lo cual le permite tener acceso a los diferentes nodos clientes de forma remota, ágil y segura la información, ordenes o solicitudes que se transite por el medio de comunicación que implementan los nodos viaja inaccesible para evitar ser conocida por un agente externo que no este incluido en el canal de transmisión, cuando un nodo principal valida la conexión, permite al nodo central tomar control de las acciones a ejecutar por medio de la línea de comandos desplegada por el administrador del sistema.

**SaltStack** utiliza una librería de seguridad, la cual contiene un complemento de Criptografía de lenguaje de programación Python (PyCrypto)<sup>23</sup>, el cual realiza funciones por medio de intercambio de llaves numéricas entre el nodo principal y el nodo cliente. Este casillero utiliza el protocolo de intérprete de órdenes seguras y no requiere de agentes en los nodos clientes, que usualmente disminuyen el rendimiento en las comunicaciones debido al consumo de recursos.

Las dos últimas herramientas analizadas, utilizan medidas de seguridad que proporcionan confiabilidad en el manejo de los datos y agrega una capa extra de seguridad a las arquitecturas de red. Esta ha sido la tendencia, por lo cual Puppet y Chef han adoptado este modelo en sus arquitecturas, para mantenerse competitivas en el mercado.

## 5.6. Rendimiento de las herramientas

El rendimiento de este tipo de herramientas es quizás el tema más importante al momento de evaluarlas. Principalmente porque al administrar miles de dispositivos de forma simultánea, se requiere velocidad de ejecución.

Aspectos como la arquitectura de las herramientas son determinantes en la velocidad de ejecución, por ejemplo, son más rápidas las arquitecturas basadas en la Nube, que las arquitecturas Cliente-Servidor, esto se debe a que las arquitecturas Nube tienen un concepto de ejecución directa, es decir, no requieren de un cliente para poder ejecutar procesos de administración y configuración en los nodos.

La herramienta más compleja es Puppet, ya que brinda dos arquitecturas distintas (Cliente-Servidor y la Nube) y además, posee dos modelos de ejecución (Master/Agente y Aplicar), lo cual la hace una herramienta con varias posibilidades de ejecución, brindando distintas soluciones a un mismo problema, de acuerdo al escenario que se presente.

Por ejemplo, Puppet es reconocida como una de las herramientas optimizadas que más se utiliza, ya que posee una serie de módulos dedicados para distintos

---

<sup>23</sup> Protocolo de seguridad de generación de numeración aleatoria.



procesos, ejecutados a velocidades que permiten administrar de manera ágil, cientos o miles de dispositivos.

Otro ejemplo de ello es Chef, el cual optimizó el proceso, por medio de la implantación de Workstations, los cuales contienen los repositorios de Chef para la ejecución de tareas, y así sirven como pequeños Chef Servers, optimizando el proceso de administración y ejecución de tareas a pequeños fragmentos de una infraestructura tecnológica.

Saltstack por ejemplo, distribuye las tareas entre varios nodos, por lo cual las funciones pueden haber sido desplegadas de varios nodos centrales, y así tener varias opciones para responder las peticiones que se realizaron a los Minions.

Ansible como tal posee un método de comunicación más rápido y eficiente que no comparte con herramienta como Puppet, Chef o SaltStack de la misma manera, permite realizar tareas en diferentes nodos principales al mismo tiempo, pero de forma más sencilla, una de las principales características y la cual sobresale es su estructura está comprendida de un diseño simple y sencillo lo cual proporciona un mejor desempeño en sus funciones.

En resumen, el rendimiento de las cuatro herramientas seleccionadas para la investigación, es muy similar, ya que pueden ser de uso general o de uso específico, de acuerdo a las necesidades de los clientes. Algunas de ellas con más trayectoria tienen cierta ventaja con respecto a las nuevas, ya que pueden poseer un núcleo más afinado, pero este criterio no garantiza que sean más eficaces.

## 6. Discusión y resultados

El análisis muestra que las herramientas Puppet y Chef utilizan una arquitectura Cliente/Servidor acorde a la época, mientras que Ansible y SaltStack han sido desarrollados bajo un modelo Nube, un modelo más reciente que presenta mejoras en rendimiento y velocidad. Esto ha generado impacto en el mercado, y obliga a Puppet y Chef a adoptar el modelo Nube, para disminuir las brechas entre herramientas.

También, los equipos comparten el concepto de módulo, el cual es la estructura básica de componentes que permiten ejecutar tareas. Estos patrones están presentes en los repositorios de código de cada una de las herramientas, los cuales están publicados en Internet, y así permiten obtener de primera mano la funcionalidad respectiva, y brinda la opción de reutilizar, modificar y ajustar a las necesidades, cada uno de los módulos en la arquitectura tienen como fin optimizar procesos.

Con base en lo anterior, surge la interrogación de si realmente pueden complementarse entre ellos o bien coexistir para brindar mayores facilidades. El código fuente disponible en Internet, lo que permite obtener los elementos más potentes de cada una de las herramientas para integrarlas en un solo repositorio y crear una herramienta más compleja y eficaz.

Además como se menciona anteriormente, surge la interrogante de si realmente pueden coexistir en un mismo sistema. Se deben tomar en cuenta aspectos de red, como el manejo de puertos para la autenticación y canales de transferencia

de información, vía SSH. También se debe modificar los contenedores de recetas que están presentes en el sistema, para que no exista conflicto de almacenamiento. Los lenguajes de programación no serían problema, ya que Puppet y Chef, tienen la capacidad de integrar Python en sus procesos de administración.

Este tipo de herramientas tiene la capacidad de integrarse con otros elementos, y da paso a nuevas implementaciones tecnológicas, que despliegan grandes volúmenes de información (e.g., salud, educación, redes sociales, ventas, publicidad, transporte) contenida en centros de datos virtuales o físicos, con lo cual poder administrarlos es una necesidad latente de los administradores de sistemas de información y TI en general.

También existen herramientas de manejo de datos relacionadas con funciones de administración de la configuración, las cuales permiten manejarlos cientos o miles de nodos que contienen la información, entre los ejemplos más relevantes están las herramientas líderes en almacenamiento y administración de información en la nube como Corporación EMC, que permiten la integración con utensilios de administración de la configuración para el manejo de los nodos que contienen la información. Este tipo de avances es posible gracias al esfuerzo que dedican las entidades a brindar soporte a las comunidades desarrolladoras, que trabajan en la integración, con los diferentes ambientes que las rodean, la creación de nuevos módulos, entre otros.

## 7. Conclusiones

La administración eficiente de la infraestructura es un tema diario para los gestores de Tecnologías de Información, ya que requiere de conocimiento en redes, sistemas operativos e incluso scripting, para simplificar tareas de administración. Es por ello que surgieron las herramientas de administración de la infraestructura.

Los mecanismos de automatización ofrecen beneficios a las instituciones que requieren de estas tecnologías, ya que la administración de cientos o miles de dispositivos que se conectan a las redes de las organizaciones, los cuales deben ser administrados y configurados para que tengan acceso a la información.

Las herramientas comparten una serie de conceptos que encapsulan las funciones anatómicas en fragmentos de código, conocidos como módulos. Cada uno de estos modelos están presentes en los repositorios de código de cada herramienta en Internet, lo que permite obtener el signo fuente para adaptarlo a las necesidades del escenario, y permite estas puedan absorber módulos de otras herramientas, para generar nuevas funcionalidades, más eficaces, de acuerdo a las necesidades de los administradores de sistemas.

También, como parte del valor agregado de estas herramientas, tienen la facilidad de integrarse con otras para crear sistemas más complejos, los cuales tienen distintas funcionalidades. Todas las que fueron mencionadas anteriormente pueden integrarse con sistemas de Nube, motores de virtualización, repositorios de código y herramientas de monitoreo de red, con el fin de automatizar tareas, para facilitar la administración de los distintos escenarios.

La compatibilidad con distintos sistemas operativos es uno de los rubros más interesantes, ya que se han desarrollado una serie de módulos que permiten clasificar los sistemas operativos para su administración. El conjunto de módulos para la administración de distintos sistemas operativos se conocen como Recetas, las cuales brindan facilidades para administrar sistemas operativos de distintas familias, incluyendo sistemas operativos para dispositivos móviles, como Android y IOS de Mac.

Además, las necesidades de los negocios han dado paso al desarrollo de nuevos módulos, los cuales permiten administrar y configurar sistemas operativos de dispositivos de red, como lo son IOS de Cisco, Procurve, Juniper entre otros, brindando facilidades en la administración de redes LAN y WAN, facilitando las tareas diarias de las mismas. Tareas como configuración y recuperación, se pueden realizar en pocos minutos, ya que se utilizan las recetas para seleccionar los sistemas operativos en los que se va a trabajar, y así optimizar los procesos.

Este tipo de herramientas se pueden conectar con las tablas de los servidores DNS, con el fin de ejecutar tareas sobre un grupo seleccionado de componentes de red, como aplicar una actualización sobre tres de los servidores de Apache, o bien reconfigurar una vlan específica en los conmutadores que ejecutan el protocolo OSPF. Estas tareas se pueden ejecutar en pocos minutos, gracias a esta funcionalidad presente en este tipo de herramientas.

Si bien es cierto la arquitectura Cliente/Servidor predomina en el mercado, la arquitectura Nube ha tenido un auge que va marcando una tendencia mercantil, debido a que es más veloz en tiempos de respuesta. Los desarrolladores de las herramientas de la administración de la configuración han adaptado este último modelo en sus instrumentos para mejorar los procesos a nivel de rendimiento.

Ansible o SaltStack son herramientas rápidas y seguras. Estas dos utilizan conexiones seguras vía SSH hacia los nodos, un proceso expedito para la ejecución de tareas. Si se busca un mecanismo robusto y complejo, se recomienda utilizar Chef, ya que su arquitectura es un modelo híbrido en Cliente/Servidor y Nube, obteniendo los mejores elementos de cada arquitectura y plasmando en su modelo, creando Estaciones de Trabajo para distintos fragmentos de Infraestructura, y disminuye las distancias entre el Servidor Central y los Nodos.

Las herramientas pueden coexistir en un mismo componente de red, es un proceso complejo, ya que involucra, reconfiguración de puertos para cada una, al igual que directorios para almacenar los repositorios de módulos y recetas. Sin embargo si pueden coexistir en un mismo sistema.

## Referencias

- Azure, M. (2010).  
 (<https://azure.microsoft.com/es-es/>) pages 8
- Cloud, H. (2012).  
 (<http://www.hpcloud.com/>) pages 8
- Engine, G. C. (1998). *Google compute engine*. Autor.  
 (<https://cloud.google.com/compute/>) pages 6

- GoPro. (2002). *Gopro*. Autor. (<http://es.shop.gopro.com>) pages 9
- Hochstein, L. (2015, may). Ansible up and running. *O'Reilly Media, First Edition*, 1-60. pages 4
- Linkedin. (2003).  
(<https://es.wikipedia.org/wiki/LinkedIn>) pages 8
- Marschall, M. (2013). *Chef infrastructure automation cookbook* (first ed.). Packt Publishing. pages 4
- Microsoft. (2014, oct). System center 2012 r2 configuration manager and microsoft intune. *Microsoft*. pages 3
- Myers, C. (January 2015, jan). Learning saltstack. *Packt Publishing*, 1-200. pages 4
- OpenLDAP. (2012, aug). Openldap software 2.4 administrator's guide. , *The OpenLDAP Foundatio*. pages 3
- OpenStack. (2012).  
(<https://www.openstack.org/>) pages 8
- PayPal. (1998). *Paypal*. Autor. (<https://es.wikipedia.org/wiki/Twitter>) pages 6
- Rackspace. (1998).  
(<http://www.rackspace.com/es/cloud>) pages 8
- SaltStack. (2015). *Execution modules*. Autor.  
(<http://docs.saltstack.com/en/latest/ref/modules/all/>) pages 12
- Service, A. W. (2006). *Aws opsworks*. Autor.  
(<http://docs.aws.amazon.com/opsworks/latest/userguide/opsworks-ug.pdf>) pages 6
- Twitter. (2007). *Twitter*. Autor. (<https://es.wikipedia.org/wiki/Twitter>) pages 6
- Uphill, T., y Arundel, J. (2015, feb). Puppet cookbook. *Packt Publishing, Third edition*, 1-313. pages 3
- VMware. (1998). *Vmware*. Autor. (<https://www.vmware.com/support/pubs/>) pages 6
- Wireless, V. (2000). *Verizon wireless*. Autor. (<http://www.verizonwireless.com/>) pages 9