**Puppet**

Es una configuración en el manejo de la solución que permite definir el estado de la infraestructura de la tecnología informática, y permite llegar a un estado deseado automáticamente. Puppet automatiza cada paso del proceso de entrega del producto, desde proporcionar máquinas virtuales y físicas para el funcionamiento y reporte. Desde fases anteriores a la codificación hasta las pruebas, el reléase y as actualizaciones.

* Tiempo libre para trabajar en proyectos que conllevan más valor en el negocio.
* Asegura consistencia, rentabilidad y estabilidad.
* Facilita la colaboración cercana entre el administrador del sistema y los desarrolladores, permitiendo más eficiencia en la entrega de código limpio que significa más valor al negocio

**Open Source Puppet**

Es flexible, una solución personalizable disponible bajo licencia de Apache 2.0, diseñado para ayudar en la automatización de la administración del Sistema en tareas repetitivas regularmente.

**La forma Como trabaja:**

* **Definir**

Primero se define el estado deseado de la computadora, almacenamiento y los dispositivos de red utilizando una simple y ponderosa declaración de configuración del lenguaje.

* **Simular**

Después uno puede similar el cambio de configuración antes de que se lleven a la vida real, así que no se sabe exactamente que pasara si el cambio es hecho.

* **Reforzar**

Puppet refuerza el estado deseado de la máquina, y automáticamente resuelve la configuración que no funciona.

* **Reporte**

Muchos reportes están disponibles en Puppet. Se puede obtener detalles en la configuración exacta del Sistema, incluyendo los cambios que se hicieron.

**Salt**

Es una plataforma basada en Python, es opensource nos sirve para el manejo de configuración y ejecución remota de una aplicación. Soportando la infraestructura como código, acercándose al despliegue y el manejo en la nube, compite principalmente con Puppet, chef y Ansible.

Salt se originó de la necesidad de tener una colección de datos rápido y ejecución del Sistema de administración el autor de Salt, Thomas S Hatch, había creado numerosas soluciones para muchas compañías para resolver problemas que se encuentren.

**Diseño**

Salt ha sido diseñado para ser altamente modular y fácilmente extensible. La meta del diseño es hace Salt fácilmente moldeable para diversas aplicaciones

El diseño modular de Salt crea módulos de Python que manejan ciertos aspectos de la disponibilidad de Salt. Estos módulos permiten interacción entre las necesidades del desarrollador y el administrador del sistema

**Ansible**

Es una plataforma de software libre para configurar y administrar computadoras. Combina instalación multi-nodo, ejecuciones de tareas ad hoc y administración de configuraciones. Adicionalmente, Ansible es categorizado como una herramienta de orquestación.Maneja [nodos](http://es.wikipedia.org/wiki/Nodo_%28inform%C3%A1tica%29) a través de [SSH](http://es.wikipedia.org/wiki/SSH) y no requiere ningún software remoto adicional (excepto Python 2.4 o posterior para instalarlo. Dispone de módulos que trabajan sobre [JSON](http://es.wikipedia.org/wiki/JSON) y la salida estándar puede ser escrita en cualquier lenguaje. Nativamente utiliza [YAML](http://es.wikipedia.org/wiki/YAML) para describir configuraciones reusables de los sistemas.

En contraste con otros programas de control de configuración como [Chef](http://es.wikipedia.org/w/index.php?title=Chef_%28software%29&action=edit&redlink=1) y [Puppet](http://es.wikipedia.org/wiki/Puppet_%28software%29), Ansible usa una [arquitectura sin agentes](http://en.wikipedia.org/wiki/Agentless_data_collection)Con la arquitectura basada en [agentes](http://es.wikipedia.org/wiki/Modelo_basado_en_agente), los nodos deben instalar localmente un proceso de comunicaciones con la máquina de control. Con la arquitectura sin agentes los nodos no necesitan instalar ni ejecutar en segundo plano ningún proceso que se comunique con la máquina de control. Este tipo de arquitectura reduce la sobrecarga de la red y previene el uso de estrategias de control más agresivas por parte del servidor (como puede ser la realización de [polling](http://es.wikipedia.org/wiki/Polling), con sus constantes operaciones de consulta).

El diseño de Ansible incluye:

* **Mínimo por naturaleza.** Los sistemas de administración no deben imponer dependencias adicionales.
* **Consistente.**
* **Seguro.** Ansible no instala agentes vulnerables en los nodos. Solamente se requiere OpenSSH que es considerado crítico y altamente testeado.
* **Alta confiabilidad.** El modelo de idempotencia es aplicado para las instalaciones y configuraciones, para prevenir efectos secundarios en la ejecución repetitiva de scripts.
* **Suave curva de aprendizaje.** Los *playbooks* usan un lenguaje descriptivo simple, basado en [YAML](http://es.wikipedia.org/wiki/YAML).