Infraestructura I

**Infraestructura:**

Conjunto de dispositivos y aplicaciones necesarios para que cualquier empresa opere. Compuesta por elementos como: Software, hardware, redes, instalaciones y todo lo que requiera desarrollo, control, monitoreo y soporte que ofrece el departamento de IT.

**Automatización de la Infraestructura;**

Usar tecnología para realizar tareas repetitivas casi sin la necesidad de las personas. Esto se realiza con la intención de ganar tiempo y maximizar la productividad de nuestra infraestructura IT; “Cómo hacer más con menos”. Cómo conseguir que los profesionales dediquen más tiempo a generar valor para la compañía y menos tareas repetitivas que podrían realizarse de forma automática.

**Beneficios de automatizar:**

* Elevar productividad empresarial
* Reducir costos operativos
* Disminuir riesgos a fallas
* Elevar la seguridad de la información
* Tener una mejor capacidad de respuesta
* Facilidad de adaptación
* Alojar mayor cantidad de datos
* Elevar la competitividad del negocio

**Tareas más comunes para automatizar:**

* Aprovisionamiento de áreas de IT: Automatiza el proceso de levantar máquinas a través de un levantamiento de servicio para el personal nuevo de la organización.
* Gestión de la configuración: Automatización de configuración de máquinas habilitadas recientemente.
* Seguridad y cumplimiento: Establece acciones de política de seguridad y cumplimiento automatizables en la gestión de las máquinas.
* Organización en la nube: Asegura la información y mantiene la disponibilidad de ella a través de la automatización para generar mayor eficiencia.
* Implementar acciones: Instalación y configuración de aplicaciones requeridas.

**Servicios de Infraestructura:**

Cloud Providers: Google Cloud, AWS y Microsoft Azure.

Manejo de código:

* Gestion de versionado: GitHub, GitLab, Gitbucket, Gitea.
* Deploy automatizado: Jenkins, Gitactions, Jetbrains.
* Contenedores: Docker.
* Ambientes de trabajo: Ansible, Terraform, Chef, Puppet.
* Monitores de red: Nagios, Prometheus, Icinga 2, Datadog.
* Lenguajes de scripting: Bash, PowerShell, Ruby, Python, Go.

**Shell Scripting:**

**Consola de Linux**:

La interfaz de línea de comandos, o CLI es un método de comunicación entre el usuario y máquina que acepta instrucciones del usuario a través de líneas de texto.

**Ejecución en inicio:**

Cuando el SO se inicia en niveles 1,2,3 y 4, nos llevará por defecto a la consola.

Si en cambio, el SO inicia en nivel 5, este tendrá interfaz gráfica. Estas varían de acuerdo a la distribución instalada.

En el caso de Ubuntu, tenemos dos opciones:

* La primera es lanzando un TTY, o espacio de trabajo sin entorno gráfico. Podemos ejecutar 7 terminales al mismo tiempo de esta forma. de la 1 a la 6, ninguna tiene interfaz gráfica. Para cambiar de TTY en Linux debemos usar CTRL+ALT+F(de 1 a 7).
* La segunda opción, con ejecución desde GUI, es encontrar una app dedicada que se ejecuta en una ventana, dentro del panel de aplicaciones de nuestra distribución. Por ejemplo, en Ubuntu, dentro del entorno gráfico GNOME, podemos encontrar la app “Terminal”, en la pestaña de actividades.

**Elevación de privilegios:**

En Linux, existen dos tipos de usuarios: El usuario común, y el superusuario (root) que posee todos los privilegios y permisos para realizar acciones sobre el sistema. Para la ejecución de algunos comandos, debemos ingresar dicho acceso. Sin embargo, se debe tener un conocimiento sobre las acciones que se realizan, ya que una acción realizada de manera errónea podría ocasionar daños importantes en el sistema. Comando sudo permite elevar privilegios a usuarios comunes.

**Comandos útiles de Linux:**

Comandos básicos:

cd (desplazamiento entre directorios), ls (listar archivos), cp (copiar archivos y pegarlos en ruta especificada), pwd (encontrar ruta hacia directorio actual), mv (mover archivos a ruta especificada), mkdir(crear directorio), rmdir (elimina directorios), rm (elimina archivos o directorios), touch (crea archivos), cat (lectura y escritura básica de archivos), nano (editor de texto), more (lee archivos de texto en formato de paginas, útil para archivos con mucho texto), pipelines | (concatena instrucciones, lo que se encuentre en la izquierda, se lo pasa al comando de la derecha para que trabaje con esa información), grep (busca strings que indiquemos en archivos, listados o lo que indiquemos), tee (contenido que le pasemos por pipeline, lo almacena en un archivo indicado).

Comandos prácticos:

curl: Verifica conectividad con url, herramienta para transmitir datos. En su función más básica, descarga el HTML de la página que le pasemos por parámetro. Con modif. -o, guardamos dicho contenido en el archivo indicado. A través del mismo también podemos descargar archivos.

jq: Permite la lectura de archivos JSON. por ejemplo, si con curl descargamos el contenido de una página, por ejemplo la siguiente api:

<https://nominatim.openstreetmap.org/reverse.php?lat=-34.60378&lon=-58.38161&zoom=18&format=jsonv2>

de la misma podemos extraer un json. Utilizando el curl y poniendo la url entre comillas dobles o simples, si lo almacenamos en un archivo y luego dicho archivo se lo damos al jq, con el mismo podemos acceder a sus elementos y propiedades.

Ej:

curl “<https://nominatim.openstreetmap.org/reverse.php?lat=-34.60378&lon=-58.38161&zoom=18&format=jsonv2>” | jq ‘.display\_name, .type’ | tee ejemploAPI.txt

**Change & Configuration Management:**

Configuration Management y Change Management son dos procesos fundamentales del conjunto de metodologías conocidas como ITIL (Information Technology Infrastructure Library).

ITIL: Traducido como Biblioteca de Infraestructura de Tecnologías de Información, es un conjunto de conceptos y buenas prácticas usadas para la gestión de servicios de tecnologías de la información, el desarrollo de tecnologías de la información y las operaciones relacionadas con la misma en general.

**Change Management:**

ITIL describe la gestión de los cambios como el proceso de controlar y gestionar un cambio a lo largo de todo su ciclo de vida con el objetivo de minimizar el riesgo.

¿Qué es un cambio?

Un cambio es la modificación o eliminación de cualquier cosa que pueda afectar directa o indirectamente los servicios. Básicamente, cualquier cambio en la infraestructura de IT de una organización puede afectar las operaciones de la organización.

Ejemplos de un cambio: reemplazo de hardware, instalación de software en un servidor o cambios en la configuración de un sistema que altera su comportamiento.

**Configuration Management:**

Es el proceso que permite gestionar los cambios de configuración de nuestros activos informáticos, permitiendo a la organización mantener un registro histórico y a su vez aplicar controles.

Cada uno de los activos informáticos en el contexto de este proceso se los conoce como Configuration Items (CI) y se almacenan en algo llamado Configuration Management DataBase (CMDB).

**Change Management + Configuration Management:**

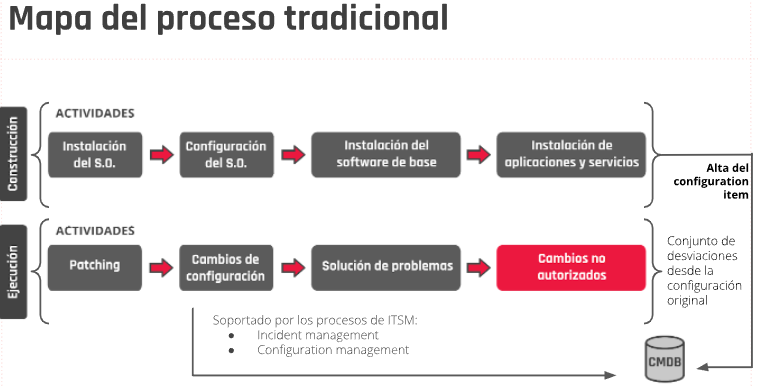
Ambos procesos se complementan, ya que change management aporta la gobernabilidad de los cambios que se efectuan en el parque tecnologico de una organizacion, y configuration management nos permite gestionar una bases de datos (CMDB) con la informacion de nuestros activos (CIs) y un historial de los cambios realizados para cada uno de ellos.

De esta manera, en caso de ocurrir una falla crítica en, por ejemplo, uno de nuestros servidores, configuration management nos otorga una vista de los cambios sucedidos previamente desde su creación, lo que nos permite reproducir cada uno de los cambios y devolverlo al estado anterior a la falla.

¿Cómo se ve en la práctica?

Tomemos como ejemplo el ciclo de vida de un servidor; este se puede dividir en dos etapas:

* Etapa de Construcción: Conjunto de pasos que pueden ser automatizados, documentados o sencillamente conocimiento común del área de IT, que permiten la puesta en marcha de un servidor y registrarlo en la CMDB como un nuevo CI.
* Etapa de Ejecución: Todo lo que suceda en relación al activo y cambios que puedan suceder en él.



**Mascotas vs Ganado:**

En términos generales, se dice que en los procesos tradicionales, cada servidor es administrado de forma individual, como una mascota.

Mientras que en los procesos modernos de administración de servidores, los mismos se administran en conjunto, con una mirada industrial y abarcativa del parque tecnológico. Se adopta un enfoque que nos permite implementar soluciones rápidas y reemplazar componentes fallidos en lugar de invertir tiempo en intentar resolver problemas complejos. Similar a como sucede en la ganadería, cuando al ganado se lo administra como un todo.

**Configuration as a Code (CaC):**

En Configuration Management, la posibilidad de definir la configuración de un servidor como código es fundamental.

* Gestionar el parque tecnológico como ganado no sería posible sin la utilización de un sistema de configuración management.
* Cualquier cosa definida como código puede ser automatizada.
* El código puede ser sometido a pruebas.
* CaC es un paso anterior a habilitar Self-Healing y Self-Remediation, prácticas que permiten implementar procesos de auto-reparación.
* Es compatible con el Change management de ITIL, ya que las modificaciones no suceden en activos, sino en un repositorio que soporta versionado. De modo que los cambios pueden ser testeados y desplegados en ambientes bajos para luego aplicarlos en producción.