

LIBÉLULA: HERRAMIENTA DE AUTOMATIZACIÓN DE GENERACIONES DE RPMS



Jorge Casas Hernán
Octubre 2018

Índice de contenidos

- 1. ¿Qué es Libélula?**
- 2. Características de Libélula**
- 3. Proceso de instalación**
- 4. Modo de uso**
- 5. ¿Quién debería usar Libélula?**
- 6. Libélula vs Jenkins**

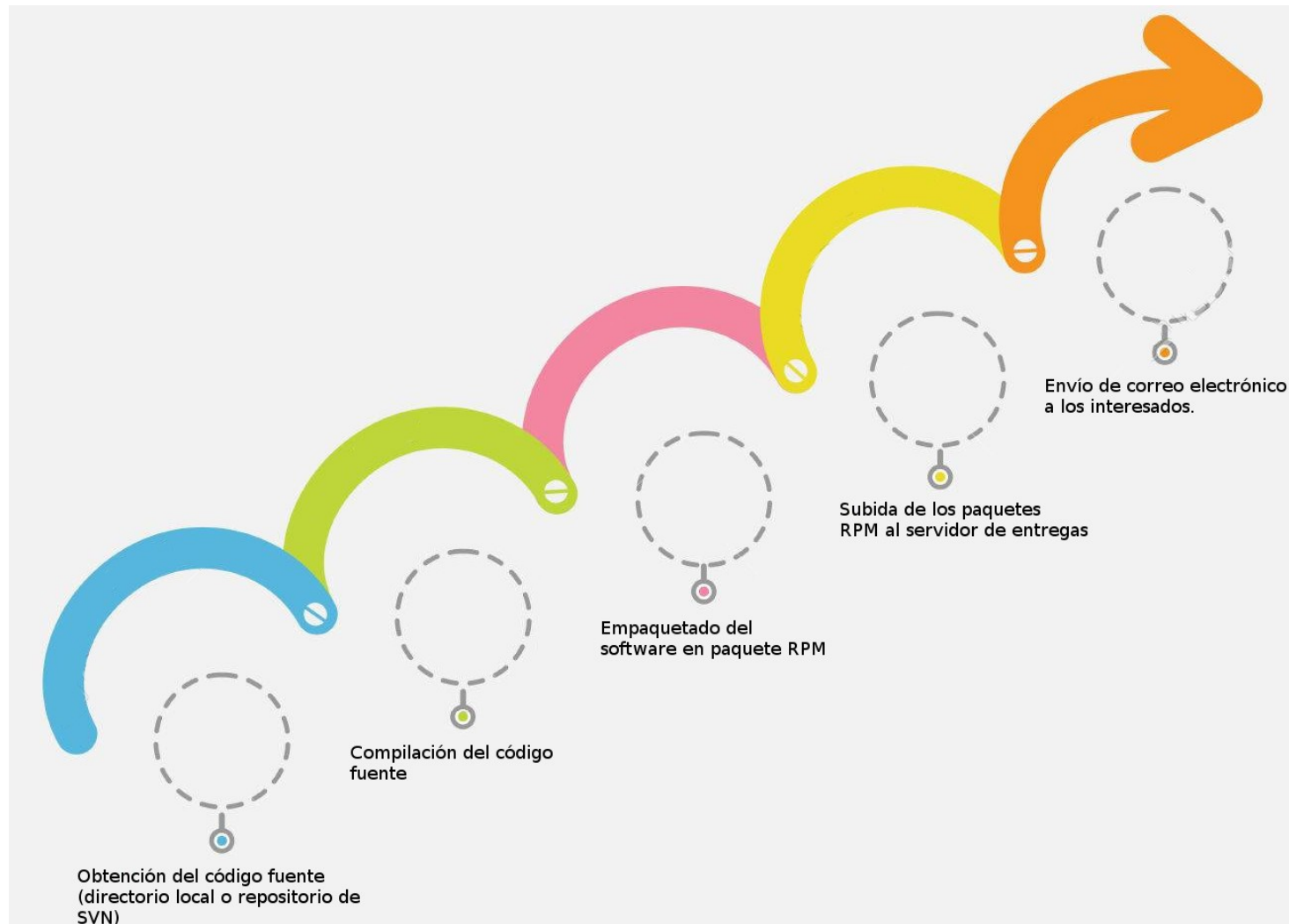
¿Qué es Libélula?

Herramienta de línea de comandos de GNU/Linux que automatiza la compilación, empaquetado, tagueo en SVN, entrega de RPMs resultantes y envío de correos de notificación a interesados, a partir de fuentes de softwares que usan RPM como tecnología de empaquetado de software.



¿Qué es Libélula?

Rango de acción de Libélula



Características de Libélula

Genérica para todos los softwares empaquetados en RPMs

- **Compatible con todos** los softwares que usen RPM como tecnología de empaquetado de software y que cumplan el estándar de la compañía.

Flexibilidad extrema

- Permite usar fuentes locales o remotas según el software, y generar y taguear en una única ejecución todos los RPMs que sean necesarios.

Puesta en marcha instantánea

- La puesta en marcha de Libélula partiendo de un sistema operativo recién instalado es de **menos de 30 segundos**.

Extremadamente fácil de usar

- Aprender a usar Libélula lleva menos de la mitad del tiempo que lleva generar un VR3000 a mano.

Tolerante a fallos

- Libélula realiza varias capas de **comprobaciones** antes de realizar cualquier acción.
- Si se produce un error cuando ya ha realizado acciones previas (como un tag en SVN) **revierte los cambios**.
- Si el usuario ha activado el envío de correos sólo enviará el correo si puede **garantizar** que todas las acciones pedidas por el usuario se han realizado con éxito.

Proceso de instalación

```
rpm -ivh libelula-X.Y.Z-R.noarch.rpm
```

En este comando se resume el proceso de instalación de Libélula.

- Libélula se distribuye en un paquete RPM que debe ser instalado en el **CentOS/RHEL compatible con el software** a generar.
- En la instalación, Libélula **pedirá al usuario su email de la compañía y su contraseña** para poder acceder a los repositorios de SVN y el servidor de correo de la compañía.
- Libélula ya está **listo para usarse**, ¡así de simple!
- La contraseña del usuario se almacena cifrada en el fichero de configuración (explicado más adelante), para cambiar la contraseña es necesario introducirla en base64:
 - *echo "contraseña del usuario" | base64*
- Libélula se puede actualizar sin perder la configuración mediante el comando *rpm -Uvh libelula-U.V.W-A.noarch.rpm*

Modo de uso

Para hacer su trabajo, Libélula necesita una serie de información por cada RPM de cada software (ruta a los fuentes, directorio del servidor de entregas, etc)

- Esta información la puede proporcionar el usuario de dos formas, que dan lugar a los **dos modos de uso** de Libélula.
- Este capítulo “Modo de uso” fue escrito sobre la última versión de Libélula a la escritura de estas líneas: versión 1.4.0. Versiones anteriores de Libélula pueden diferir con lo aquí descrito.

**Uso por parámetros
de línea de
comandos**

Permite generar **1**
RPM por ejecución

**Uso por fichero de
configuración
JSON**

Permite generar **N**
RPMs por ejecución

Modo de uso

Uso por parámetros de línea de comandos

libelula <Ruta local o URL de SVN o RPM para renombrar versión> <Versión>

[--name <Nombre del paquete a generar>]

[--branch <URL del branch de SVN a taguear>] → Si no se establece y se añade el parámetro --tag, se taguearán los fuentes especificados en el primer parámetro.

[--tag <URL de SVN del directorio de tags>]

[--dest <Ruta directorio destino>] → Si se establece, el RPM resultante se creará en la ruta especificada además de en el directorio actual.

[--update-repo] → Actualiza los fuentes locales antes de llevar a cabo la generación. Sólo aplica si los fuentes son locales.

[--srpm] → Genera el SRPM (Source RPM) además del RPM. Éste siempre se crea en el directorio actual y el parámetro --dest no le afecta.

[--debug] → Si se establece, en los scripts de compilación del archivo .spec existirá la variable de entorno \$DEBUG con el valor “yes” (uso exclusivo para desarrolladores).

[--checksum-file <Ruta al archivo checksum>] → Añade al archivo de checksum una línea con la información del checksum del RPM generado, con el formato “Fecha – Checksum – RPM”. Si el archivo indicado no existe se crea.

[--pre-gen-cmd <Comando>] → Ejecuta el comando <Comando> inmediatamente antes de iniciar la generación del RPM.

[--post-gen-cmd <Comando>] → Ejecuta el comando <Comando> inmediatamente después de finalizar la generación del RPM.

<> → Parámetro obligatorio **[]** → Parámetro opcional

Modo de uso

Uso por parámetros de línea de comandos

[--send-mail <yes/no/confirm>] → Habilita o deshabilita el envío de correos. Sólo aplica si todas las operaciones se han completado de forma exitosa, de lo contrario sea cual sea lo especificado nunca se enviará ningún correo. En el caso de establecerse a “confirm”, al finalizar las operaciones de forma exitosa, Libélula pregunta al usuario si se desea enviar el correo de notificación o no.

[--mail-subject <Asunto del mail>] → Especifica el asunto del correo. SOFTWARE_NAME, SOFTWARE_VERSION Y RPM_FILENAME son macros disponibles que serán reemplazadas por su valor en el asunto final.

[--mail-header <Cabecera del mail>] → Especifica el contenido de la cabecera del correo. Se pueden usar las mismas macros que en --mail-subject.

[--mail-footer <Pie del mail>] → Especifica el contenido del pie del correo. Se pueden usar las mismas macros que en --mail-subject.

[--to <Emails Para (separados por comas)>] → Lista de correos que deben ir en el Para.

[--cc <Emails CC (separados por comas)>] → Lista de correos que deben ir en copia.

[--mail-base-link <Link base descarga>] → Link base de descarga, sin el nombre del RPM.

[--mail-note <Nota mail>] → Añade una nota al correo con el contenido <Nota mail> en rojo.

<> → Parámetro obligatorio **[]** → Parámetro opcional

Modo de uso

Uso por fichero de config. de RPMs (/etc/libelula/rpm-params.json)

- El **modo de uso por parámetros de línea de comandos no es eficaz**: Hay que escribir demasiados parámetros para generar un determinado RPM. Y además, la generación de un software supone normalmente la generación de dos o más RPMs.
- Para solucionar esto, Libélula cuenta con un modo de uso en el cual la información del RPM no la obtiene por parámetros de línea de comandos si no mediante la lectura de un fichero de configuración, localizado por defecto en **/etc/libelula/rpm-params.json**. Con este modo de uso es con el que se **saca realmente partido a la herramienta**.
- Al instalar Libélula, este fichero ya viene relleno con unos parámetros por defecto para cada RPM de cada software compatible.
- El contenido de este fichero está en **formato JSON**, muy legible a los ojos humanos, siendo cómoda la edición, el borrado y el añadido de campos.
- La omisión de campos de un determinado RPM en el fichero JSON son interpretados por Libélula como la cadena vacía.

Modo de uso

Uso por fichero de config. de RPMs (/etc/libelula/rpm-params.json)

- El comando para generar **un único RPM** de un software usando el fichero de configuración es el siguiente:

libelula <Nombre> <Versión>

Donde <Nombre> es el nombre establecido en el campo “name” del fichero de configuración de RPMs /etc/libelula/rpm-params.json.

- Se puede combinar con parámetros de línea de comandos. Los parámetros tienen mayor prioridad que el archivo de configuración. Por ejemplo:

libelula vr3000 4.20.5.RC1-1 --srpm

En este ejemplo, aunque en el fichero de configuración esté especificado que para vr3000 no se genere el SRPM, al haberlo especificado de forma explícita por parámetro, sí se generará.

Modo de uso

Uso por fichero de config. de RPMs (/etc/libelula/rpm-params.json)

- En el modo de uso por fichero de configuración es posible generar en un **único comando varios RPMs**. La sintaxis es la siguiente:

libelula <Nombre del RPM1>,<Nombre del RPM2>,... <Versión del RPM1>,<Versión del RPM2>,...

- En este caso no está permitido el uso de parámetros de línea de comandos.
- Si está **activado** el envío de mails para el RPM colocado en primer lugar de la lista, Libélula enviará un **único mail con la información de TODOS los RPMs** generados en la ejecución. El asunto del correo, la cabecera, el pie, la nota (si se ha especificado) y los destinatarios (Para y CC) que aplican son los del RPM colocado en primer lugar.
- Si está **desactivado** el envío de mails para el RPM colocado en primer lugar, Libélula no enviará ningún mail aunque esté activado el envío en alguno de los otros RPMs.

Modo de uso

Fichero de config. global de Libélula (/etc/libelula/libelula.conf)

- Además del fichero de configuración de RPMs del que hemos hablado estas últimas diapositivas, existe otro fichero de configuración que almacena configuración global de Libélula, localizado en **/etc/libelula/libelula.conf**.
- **No es habitual tener que modificar este fichero**, ya que la información que contiene por defecto no suele ser susceptible de cambiar.
- Algunos de los parámetros globales que se establecen en este fichero son:
 - Email de usuario y contraseña (en base64) de la compañía.
 - Ruta al fichero de configuración de RPMs (por defecto /etc/libelula/rpm-params.json).
 - Ruta al fichero de log (por defecto /etc/libelula/libelula.log).
- Si el repositorio de SVN de un software en cuestión está en un servidor cuya autenticación es local, sin que se use el servidor de autenticación corporativo (como por ejemplo el servidor SVN del software SRCC), se debe especificar la contraseña local en base64 en la variable SVN_LOCAL_PASS.

¿Quién debería usar Libélula?

La generación de los softwares hasta ahora ha sido una tarea del equipo de CM y Entorno Linux. Sin embargo, al estar el proceso 100% automatizado gracias a Libélula, es más eficiente su uso por parte de los grupos de desarrollo de cada software.

El uso de Libélula por parte de los grupos de desarrollo proporciona los siguientes beneficios claves:

- Si el proceso de compilación y empaquetado falla, el grupo de desarrollo es consciente de este error de forma instantánea, pudiéndolo resolver en ese mismo momento.
- Evita posibles malas prácticas de envíos de código no consolidados.
- El grupo de CM y Entorno Linux no provoca cuellos de botella por la solicitud de varias versiones simultáneas de diferentes softwares.

Aunque las generaciones las hagan los propios grupos de desarrollo, el grupo de CM y Entorno Linux no se desvinculará de este proceso.

- El grupo de CM y Entorno Linux supervisará que todo el ciclo del software se está realizando de forma correcta.
- Para centralizar el envío de software fuera del departamento, quizás sí se podría mantener que las versiones definitivas las siguieran entregando el grupo de CM y Entorno Linux.

Libélula vs Jenkins

Aunque Libélula no es una herramienta de integración continua como lo es Jenkins, podría sustituirlo si no se requiere ningún plugin concreto ni funcionalidad específica de Jenkins.

Las ventajas que ofrece Libélula sobre Jenkins son las siguientes:

- Tiempo de puesta en marcha infinitamente inferior.
- Proporciona un interfaz de línea de comandos accesible por todos desde local, sin necesitar ningún recurso externo.
- Mucho más flexible (Cualquier caso particular en la generación de un software, como por ejemplo que sólo se quiera generar un único CFG, supone crear en Jenkins una nueva tarea).
- No consume recursos si no se usa, siendo ideal para máquinas virtuales de desarrollo donde los recursos suelen ser escasos (Libélula no es un servicio como lo es Jenkins).
- Es posible usarlo sin contar con un interfaz gráfico.

Libélula y Jenkins son softwares complementarios

- Cuando se ponga en marcha Jenkins para la integración continua de los softwares, se podrá seguir usando Libélula ya que se puede integrar en Jenkins.
- Jenkins delegará el trabajo de las generaciones de todos los softwares a Libélula.

Libelula

Idea original, diseño e implementación por
Jorge Casas Hernán
(c) 2018

