**Actualización OpenNAC**

Repositorio OpenNAC: “/etc/yum.repos.d/opennac.repo”.

Para actualizar: “yum update”.

Si hay error comprobar conexión: “wget <https://repo-opennac.opencloudfactory.com/x86_64>”.

Ejecutar script post update: “sh /usr/share/opennac/utils/scripts/post-install\_no-move.sh”.

\*\*El script hace un reinicio de los servicios también.

Mientras se realiza el proceso de actualización y la ejecución del anterior script, se va mostrando un listado de ficheros “.sample” y “.ini”. Son ficheros existentes que se han actualizado en nuevas versiones de OpenNAC que no pisan los anteriores, por si se hubiese hecho alguna modificación importante para adecuar su uso a clientes. Todos estos archivos son interesantes de revisar.

find /usr/share/opennac/ -name \*.ini.sample

Lo más recomendable es hacer un fichero con un listado de los ficheros mencionados por la actualización, e ir comprobando las modificaciones y decidir si son sobrescritos, modificados o mantenidos. Por regla general los archivos que se modifican para los clientes son los de Radius.

En principio todos los ficheros del analytics y de la sonda son actualizados, ya que son componentes que no se suelen customizar.

Para comprobar las diferencias entre los archivos existentes y actualizados, y tomar una decisión, usar el siguiente comando: “diff *archivo archivo.sample*”. En caso de querer actualizar el archivo: “mv archivo *archivo.bak*” y “mv *archivo.sample* *archivo*”.

**Archivos destacados:**

Radius, el software responsable de recibir las peticiones de los switches, path: “/etc/raddb/”.

* Clients.conf: donde se configuran las pre-shared keys. Archivo que no se suele sustituir, ya que suele ser modificado en base al cliente. “Las redes que están en los switches, que son los dispositivos que usarían esa pre-shared key para cifrar la comunicación con el radius y autentificar.”.
* Policy.conf, default.conf, default\_proxy.conf, normalmente se modifican.
* Reiniciar el servicio: “service radiusd restart”.

Rsyslog: “/etc/rsyslog.d/”.

* “opennac\_analytics.conf”.
* Reiniciar servicio: “service rsyslog restart”.

Filebeat, servicio que coge los logs de las autentificaciones y los envía al anaylitics: “/etc/filebeat/”.

* “filebeat.yaml”.
* FIJARSE EN LOS PERMISOS DEL ARCHIVO, igual hay que cambiarlos.
* Logs: “/var/log/filebeat/filebeat”.
* Reiniciar el servicio: “service filebeat restart”.

Ficheros de OpenNAC, “corazón”: “/usr/share/opennac/”.

* \*\*Suele ser común que estos ficheros estén modificados.
* “/userportal/application/configs“.
* “/api/applications/configs/application.ini“.
  + Para facilitar la gestión con equipos en cluster. En vez de configurar las IPs, se ha añadido onmaster en el fichero de hosts /etc/hosts/ (comprobar!), así, en cada equipo local que apunte al master (core?), apunta a onmaster. Por ello en el host del equipo, en el fichero indicado se pone a localhost como onmaster.
* “/api/applications/configs/mobile-connect.ini“.
* “/api/applications/configs/api.ini.sample“.
  + “resources.log.syslog.filterParams.priority = \_“: 6-Normal / 7-Debug.
* “/api/scripts/win-agent-old/taskbar/openNAC.taskbar.ini“.

Para reiniciar todos los servicios: “sh /usr/share/opennac/utils/scripts/post-install\_no-move.sh”.

* Si salta error en application.ini “/usr/share/opennac/healthcheck/application.ini”: cuando healthcheck está en un mismo servidor que un core, el application.ini que indica es el mismo de la api, por lo que podemos ignorar el mensaje. Sin embargo, en un analytics o en un sensor no debería darnos error.

Comprobar que todo funciona correctamente (también en sonda o analytics): “php /usr/share/opennac/healthcheck/healthcheck.php".

* Si algún servicio falla: “systemctl daemon-reload” y arrancar nuevamente el que ha fallado previamente.

**Ocupación de disco:**

\*\*IMPORTANTE: Comprobar espacio disco, sobre todo en el analytics y sonda, donde hay un gran tráfico de logs.

Ocupación: “du -hs \*”.

Comprobar listado índices creados en elasticsearch: “curl -XGET <http://localhost:9200/_cat/indices>”.

Si la ocupación de disco es alta:

* Todos los índices se suelen borrar dentro de un tiempo determinado. Por defecto 7 días en la sonda y 30 días en el analytics. Comprobar que se cumple la regla en el apartado action del siguiente fichero: “/etc/elastCurator/action.yaml”.
* El fichero se ejecuta por cron una vez al día, comprobar si hay algún error y si no se ha sustituido en la actualizacion: cat /etc/cron.d/opennac-curator
* Por último ejecutar el comando manualmente: “/usr/share/opennac/analytics/elasticsearch\_purge\_index\_curator.sh”.
* “/usr/bin/curator –config /etc/elastCurator/config.yaml /etc/elastCurator/action.yaml”.
* Analizar si los índices han sido borrados.
* Cuando el disco está lleno puede que puede que la siguiente cola no se haya llevado a cabo, comprobar su contenido/errores: “tail -f /var/log/logstash/logstash-plain.log”.

Cuando elasticsearch llega a 90% ocupación, para no corromper datos se bloquea la escritura dentro de los índices, y se les pone una watermark. Mientras no se libere espacio no deja que nadie grabe en disco.

* Para comprobarlo: “less -n /var/log/elasticsearch/elasticsearch.log”.
* \*\*IMPORTANTE: Si ves alguna watermark volver a activar la escritura de los índices:
  + “cd /usr/share/opennac/analytics/scripts“.
  + “sh read\_only.sh“.
* Esperar un tiempo, y ver que efectivamente no hay más errores en la cola (mirando la fecha de cada error): “tail -f /var/log/logstash/logstash-plain.log”.

**Actualización base de datos:**

“php /usr/share/opennac/api/scripts/updatedb.php”.