

Manual Técnico

Elaborado por: Angyee Marín

Índice

Contenido

1. Información del Sistema	4
¿Cómo funciona?	4
Recolección de datos	4
Archivos Fuentes del Sistema	4
Procesamiento de Datos	4
Almacenamiento de datos	4
Programación de Tareas	5
Logs	5
2. Requerimientos del Sistema	5
2.1 Variables de Entorno Requeridas.....	5
2.2 Variables en Shell.....	6
2.3 Otras Variables de Entorno	6
3. Instalación	6
4. Comandos	7
4.1 Automatizaciones de Compilación y Construcción (Makefile).....	7
4.1.1 Configuración del Sistema	7
4.1.2 Ejecución del Sistema.....	7
4.1.3 Creación de Carpetas Requeridas.....	7
4.1.4 Recolección de Datos de SCAN	7
4.1.5 Procesamiento de Datos de SCAN.....	7
4.1.6 Recolección y Procesamiento de Datos de SCAN	8
4.1.7 Almacenamiento de la Data Obtenida	8
4.1.8 Limpieza de Data y Archivos Temporales	8
4.1.9 Actualización de Fuentes	8
4.2 Interfaz de Línea de Comandos del Sistema	8
4.2.1 Especificar Fecha para Recolección de Datos	8
3.2.3 Especificar Capa de BBIP para Recolección de Datos	9
4.2.4 Especificar Fecha para el Procesamiento de Datos	9
4.2.5 Especificar Capa de BBIP para el Procesamiento de Datos	9
4.2.6 Almacenamiento de Datos	9
4.2.7 Especificar Fecha para Almacenamiento de Datos	9
4.2.8 Especificar Capa de BBIP para Almacenamiento de Datos	10

4.2.9 Forzar la Carga de Toda la Data Recolectada	10
4.2.10 Especificar Carga de los Resúmenes Diarios	10
4.2.11 Especificar Fecha para la Carga de los Resúmenes Diarios	10
4.2.12 Forzar la Cargar de Todos los Resúmenes Diarios.....	11
4.2.13 Especificar Capa para Actualizar las Fuentes	11
4.2.14 Construcción de la Base de Datos.....	11
4.2.16 Generación de Reporte Diario	12
4.2.17 Generación de Reporte Semanal	12
4.2.18 Generación de Reporte Quincenal	12
4.2.19 Generación de Reporte Mensual	13

1. Información del Sistema

ScanBackup es un sistema diseñado para la recolección de datos de tráfico de todas las interfaces de la red a través de la plataforma de Monitoreo (SCAN), y procesamiento de la misma para la generación de distintos reportes para el análisis de tráfico.

¿Cómo funciona?

El sistema se encarga de recolectar la información del tráfico en datos de cada 5 minutos de todas las interfaces de la red registradas en SCAN. Esta data, una vez obtenida, es procesada para obtener los promedios diarios del tráfico de cada una de las interfaces de las capas obtenidas, para luego almacenar en la base de datos toda la información recolectada y procesada.

El sistema tiene un orden estricto para dicha operación: recolección, procesamiento y almacenamiento de los datos. Esto se logra mediante el correcto orden de ejecución de las rutinas. Las rutinas se encuentran en la carpeta *routines/*.

Recolección de datos

scanbackup/routines/scanner.sh

Este script se encarga de la recolección de los datos de tráfico de SCAN del día anterior de todas las interfaces especificadas en el directorio *sources/SCAN*. Todo el tráfico obtenido está en base a intervalos de cinco minutos. La información obtenida se alojará en el directorio *data/SCAN*, dividido en carpetas según la capa que corresponda.

Archivos Fuentes del Sistema

Para poder realizar la captura de data, es necesario declarar las fuentes a cada una de las capas disponibles para consultar (BORDE, BRAS, CACHING, RAI, IXP, IPBRAS). Los datos que se especifiquen en cada uno de dichos archivos, son los que el sistema se va a encargar de recolectar información. Estas fuentes deben estar en un archivo sin extensión en el directorio *sources/SCAN/*, con el nombre de la capa escrito en mayúscula sin espacios o caracteres especiales.

Como ya se dijo, los archivos fuentes deben estar clasificados por las capas del sistema, siguiendo el siguiente formato:

link-de-acceso;nombre-de-la-interfaz;capacidad-de-la-interfaz;tipo-de-la-interfaz

Procesamiento de Datos

scanbackup/routines/daily.py

Este script se encarga de procesar los datos de SCAN recolectados para generar los resúmenes de promedios de tráfico para todas las interfaces que se encuentren disponibles para las capas del sistema. La nueva data que se genere será almacenada en el directorio *data/SCAN/DAILY_SUMMARY* según la capa que corresponda.

Almacenamiento de datos

Este módulo se encuentra en la carpeta *updater/*, se encarga de cargar toda la data recolectada y procesada de las interfaces obtenidas de SCAN en la base de datos del sistema.

Programación de Tareas

Para la correcta ejecución del actualizador del sistema, se debe configurar la programación de tareas para que se ejecute automáticamente. Para ello se debe añadir al *crontab* del sistema el siguiente comando:

```
export PWDSCANBACKUP="/home/user/ScanBackup" # Debe reemplazarse por la ruta
del directorio del sistema

export USERSCAN="usuario" # Debe reemplazarse por el usuario

export PASSWORDSCAN="contraseña" # Debe reemplazarse por la contraseña

00 04 * * * cd $PWDSCANBACKUP && /usr/bin/make run
```

O de forma más textual:

```
export PWDSCANBACKUP="/home/user/ScanBackup" # Debe reemplazarse por la ruta
del directorio del sistema

export USERSCAN="usuario" # Debe reemplazarse por el usuario

export PASSWORDSCAN="contraseña" # Debe reemplazarse por la contraseña

00 04 * * * bash /home/user/ScanBackup/scanbackup/routines/scanner.sh

00 07 * * * /home/user/ScanBackup/.venv/bin/python -m scanbackup.routines.diario

30 07 * * * /home/user/ScanBackup/.venv/bin/python -m scanbackup.updater data
```

Logs

El sistema lleva un registros de logs en el directorio *data/logs/*. Estos logs cuentan con un formato específico para facilitar su lectura. Cada operación que ejecuta el sistema se registra en dichos archivos.

2. Requerimientos del Sistema

El sistema requiere la instalación de:

- Python 3.10 o superior.
- MongoDB 8.0 o superior.

Solo válido para sistemas Linux.

2.1 Variables de Entorno Requeridas

El sistema requiere un archivo *“.env.production”* o *“.env”* en la carpeta raíz del proyecto con las siguientes variables de entorno:

```
URI_MONGO="mongodb://localhost:27017/ScanBackupDB"
```

Nota: Para la carga de variables de entorno en modo desarrollador se debe tener el archivo *“.env.development”*, así podrá especificar la opción *“--dev”* en los comandos con la opción disponible.

2.2 Variables en Shell

El sistema requiere que se añadan las siguientes variables a nuestro archivo de shell (*.bashrc* o *.zshrc*):

```
export PWDSCANBACKUP="/home/user/ScanBackup" # Debe reemplazarse por la ruta del directorio del sistema
```

```
export USERSCAN="usuario" # Debe reemplazarse por el usuario
```

```
export PASSWORDSCAN="contraseña" # Debe reemplazarse por la contraseña
```

Nota: Esto es importante para el correcto funcionamiento de la captura de data.

2.3 Otras Variables de Entorno

Solo para el funcionamiento del actualizador de las fuentes de enlaces, se debe añadir las siguientes variables de entorno en el archivo `.env.production`` o `.env``:

```
SCAN_USERNAME="username"
```

```
SCAN_PASSWORD="password"
```

```
SCAN_URL_BORDE_HUAWEI="url" # Página principal de los enlaces
```

```
SCAN_URL_BORDE_CISCO="url" # Página principal de los enlaces
```

```
SCAN_URL_BORDE_JUNIPER="url" # Página principal de los enlaces
```

```
SCAN_URL_BRAS_HUAWEI="url" # Página principal de los enlaces
```

```
SCAN_URL_CACHING_HUAWEI="url" # Página principal de los enlaces
```

```
SCAN_URL_RAI_HUAWEI="url" # Página principal de los enlaces
```

```
SCAN_URL_RAI_ZTE="url" # Página principal de los enlaces
```

```
SCAN_URL_IPBRAS="url" # Página principal de los enlaces
```

```
SCAN_URL_IXP="url" # Página principal de los enlaces
```

Nota: Esto puede ser omitido si no se espera realizar la ejecución del mismo. Por defecto, se puede omitir.

3. Instalación

Nota: Se recomienda realizar la instalación del sistema con el comando disponible en el *makefile* [3.1.1 Configuración del Sistema](#).

El proyecto cuenta con un archivo `pyproject.toml`` que contiene toda la información necesaria para instalar el sistema. Para instalar se debe crear un entorno virtual y ejecutar el siguiente comando:

```
$ pip install .
```

Si se realiza de esta manera, una vez finalizada la instalación de dependencias, debe instancia la base de datos a través del siguiente comando:

```
$ python3 -m scanbackup.database start
```

4. Comandos

4.1 Automatizaciones de Compilación y Construcción (Makefile)

El sistema cuenta con un archivo *Makefile* que contiene toda la información necesaria para ejecutar las rutinas de manera automática, y realizar las operaciones necesarias para el correcto funcionamiento del sistema.

4.1.1 Configuración del Sistema

Para instanciar correctamente el sistema, se debe ejecutar el siguiente comando:

```
$ make setup
```

Esto creará las carpetas necesarias para el funcionamiento del sistema, creará la base de datos e instalará las dependencias de Python.

4.1.2 Ejecución del Sistema

Para poder ejecutar el sistema, se debe ejecutar el siguiente comando:

```
$ make run
```

Esto capturaré la data del día anterior, procesará los datos y los almacenará en la base de datos del sistema.

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ make run-dev
```

IMPORTANTE: Para esto se debe creado el archivo *.env.development* con las variables de entorno requeridas.

4.1.3 Creación de Carpetas Requeridas

Si se necesita solo crear las carpetas requeridas para el funcionamiento del sistema, se debe ejecutar el siguiente comando:

```
$ make setup-files
```

4.1.4 Recolección de Datos de SCAN

Si se necesita solo recolectar los datos de SCAN, se debe ejecutar el siguiente comando:

```
$ make run-scan
```

Más información sobre las rutinas de recolección de datos se encuentra en la sección [Recolección de datos](#)

4.1.5 Procesamiento de Datos de SCAN

Si se necesita solo procesar los datos de SCAN y obtener la data correspondiente a los promedios diarios, se debe ejecutar el siguiente comando:

```
$ make run-daily
```

Más información sobre las rutinas de procesamiento de datos se encuentra en la sección [Procesamiento de Datos](#)

4.1.6 Recolección y Procesamiento de Datos de SCAN

Si se necesita solo recolectar y procesar los datos de SCAN, se debe ejecutar el siguiente comando:

```
$ make run-base
```

4.1.7 Almacenamiento de la Data Obtenida

Si se necesita solo almacenar la data existente temporal en el sistema, se debe ejecutar el siguiente comando:

```
$ make run-updater
```

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ make run-updater-dev
```

IMPORTANTE: Para esto se debe creado el archivo *.env.development* con las variables de entorno requeridas.

4.1.8 Limpieza de Data y Archivos Temporales

Si se necesita limpiar las carpetas donde se almacenan temporalmente la data obtenida de SCAN antes de guardarse en la base de datos, se debe ejecutar el siguiente comando:

```
$ make clean-data
```

Nota: Tome precaución de utilizar este comando antes de que la data se haya almacenado previamente en la base de datos.

4.1.9 Actualización de Fuentes

Si se requiere actualizar los archivos fuentes que utiliza el sistema para recolectar la información de SCAN puede ejecutar el comando:

```
$ make updater-sources
```

IMPORTANTE: Tenga en cuenta que una vez ejecutado el comando, se recomienda la revisión manual de los archivos, ya que el sistema pudiera no encontrar alguna información de las interfaces en SCAN, dejando así valores por defecto para su modificación manual.

4.2 Interfaz de Línea de Comandos del Sistema

Para la ejecución de los comandos descritos a continuación, se recuerda la activación del entorno virtual.

4.2.1 Especificar Fecha para Recolección de Datos

Para recolectar los datos de SCAN de una fecha específica, se debe ejecutar el siguiente comando:

```
$ bash $(pwd)/scanbackup/routines/scanner.sh <FECHA>
```

Donde <FECHA> es la fecha del día que se desea recolectar en el formato YYYY-MM-DD. Por defecto, si no se especifica, se recolectará los datos del día anterior al actual.

3.2.3 Especificar Capa de BBIP para Recolección de Datos

Para recolectar los datos de SCAN de una capa específica, se debe ejecutar el siguiente comando:

```
$ bash $(pwd)/scanbackup/routines/scanner.sh <FECHA> <LAYER>
```

Donde <FECHA> es la fecha del día que se desea recolectar en el formato YYYY-MM-DD y <LAYER> es la capa de la que se desea recolectar los datos.

Nota: Las capas disponibles son: *BORDE*, *BRAS*, *CACHING*, *RAI*, *IXP*, *IPBRAS*. Por defecto, si no se especifica, se recolectará los datos de todas las capas del día anterior al actual.

4.2.4 Especificar Fecha para el Procesamiento de Datos

Para procesar los datos de SCAN de una fecha específica y obtener los promedios diarios, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.routines.daily --date <FECHA>
```

Donde <FECHA> es la fecha del día que se desea procesar en el formato YYYY-MM-DD.

Por defecto, si no se especifica, se procesará los datos del día anterior al actual.

4.2.5 Especificar Capa de BBIP para el Procesamiento de Datos

Para procesar los datos de SCAN de una capa específica y obtener los promedios diarios, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.routines.daily --layer <LAYER>
```

Donde <LAYER> es la capa de la que se desea procesar los datos.

Nota: Las capas disponibles son: *BORDE*, *BRAS*, *CACHING*, *RAI*, *IXP*, *IPBRAS*. Por defecto, si no se especifica, se procesará los datos de todas las capas.

4.2.6 Almacenamiento de Datos

Para almacenar toda la data recolectada y procesada, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.updater data
```

Esto almacenará la data recolectada y procesada de todas las capas del día anterior al actual.

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ python3 -m scanbackup.updater data --dev
```

IMPORTANTE: Para esto se debe creado el archivo *.env.development* con las variables de entorno requeridas.

4.2.7 Especificar Fecha para Almacenamiento de Datos

Para almacenar los datos de SCAN de una fecha específica, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.updater data --date <FECHA>
```

Donde <FECHA> es la fecha del día que se desea almacenar en el formato YYYY-MM-DD.

Nota: La bandera `--dev` es válida para combinación.

4.2.8 Especificar Capa de BBIP para Almacenamiento de Datos

Para almacenar los datos de SCAN de una capa específica, se debe ejecutar el siguiente comando según la capa deseada:

```
$ python3 -m scanbackup.updater data --borde
```

```
$ python3 -m scanbackup.updater data --bras
```

```
$ python3 -m scanbackup.updater data --caching
```

```
$ python3 -m scanbackup.updater data --rai
```

```
$ python3 -m scanbackup.updater data --ixp
```

```
$ python3 -m scanbackup.updater data --ipbras
```

Nota: La bandera `--date` y `--dev` es válida para combinación.

4.2.9 Forzar la Carga de Toda la Data Recolectada

Para subir toda la data recolectada y procesada, sin importar las fechas, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.updater data --force
```

Esto subirá la data recolectada y procesada de todas las capas de todas las fechas que se obtenga.

Nota: La bandera `--dev` es válida para combinación.

4.2.10 Especificar Carga de los Resúmenes Diarios

Para subir solo la data procesada de los resúmenes diarios, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.updater daily
```

Esto subirá la data procesada de los resúmenes diarios de todas las capas del día anterior al actual.

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ python3 -m scanbackup.updater daily --dev
```

IMPORTANTE: Para esto se debe creado el archivo ``.env.development`` con las variables de entorno requeridas.

4.2.11 Especificar Fecha para la Carga de los Resúmenes Diarios

Para subir solo la data procesada de los resúmenes diarios de una fecha específica, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.updater daily --date <FECHA>
```

Donde `<FECHA>` es la fecha del día que se desea almacenar en el formato `YYYY-MM-DD`.

Nota: La bandera `--dev` es válida para combinación.

4.2.12 Forzar la Cargar de Todos los Resúmenes Diarios

Para subir todos los resúmenes diarios, sin importar las fechas, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.updater daily --force
```

Esto subirá los resúmenes diarios de todas las capas de todas las fechas que se obtenga.

Nota: La bandera `--dev` es válida para combinación.

4.2.13 Especificar Capa para Actualizar las Fuentes

Para actualizar las fuentes del sistema solo para una capa específica, se debe ejecutar el siguiente comando según la capa deseada:

```
$ python3 -m scanbackup.updater sources --borde
```

```
$ python3 -m scanbackup.updater sources --bras
```

```
$ python3 -m scanbackup.updater sources --caching
```

```
$ python3 -m scanbackup.updater sources --rai
```

```
$ python3 -m scanbackup.updater sources --ipbras
```

4.2.14 Construcción de la Base de Datos

Para construir la base de datos, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.database start
```

Esto creará la base de datos con los esquemas e índices correspondientes.

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ python3 -m scanbackup.database start --dev
```

IMPORTANTE: Para esto se debe creado el archivo `.env.development` con las variables de entorno requeridas.

4.2.15 Destrucción de la Base de Datos

Para destruir la base de datos, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup.database drop
```

Esto eliminará toda la información de la base de datos de manera irreversible.

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ python3 -m scanbackup.database drop --dev
```

IMPORTANTE: Para esto se debe creado el archivo `.env.development` con las variables de entorno requeridas.

4.2.16 Generación de Reporte Diario

Para generar el reporte diario, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup diario
```

Esto generará el reporte del día anterior y lo exportará en un archivo .xlsx llamado *Data_Diario_BBIP.xlsx*.

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ python3 -m scanbackup diario --dev
```

IMPORTANTE: Para esto se debe creado el archivo *.env.development* con las variables de entorno requeridas.

Además, se puede especificar la fecha para el que se desea generar el reporte, con el siguiente comando:

```
$ python3 -m scanbackup diario --date <FECHA>
```

Donde *<FECHA>* es la fecha del día que se desea generar el reporte en el formato *YYYY-MM-DD*.

Nota: La bandera *--dev* es válida para combinación.

4.2.17 Generación de Reporte Semanal

Para generar el reporte semanal, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup semanal
```

Esto generará un reporte con el promedio de datos desde el lunes de la semana pasada hasta el domingo de la semana cursando. Este reporte será exportado un archivo .xlsx llamado *Data_Semanal_BBIP.xlsx*.

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ python3 -m scanbackup semanal --dev
```

IMPORTANTE: Para esto se debe creado el archivo *.env.development* con las variables de entorno requeridas.

Además, se puede especificar la fecha para el que se desea generar el reporte, con el siguiente comando:

```
$ python3 -m scanbackup semanal --literal
```

Esto generará un reporte desde de la semana contando los 7 días hacia atrás iniciando desde el día actual.

Nota: La bandera *--dev* es válida para combinación.

4.2.18 Generación de Reporte Quincenal

Para generar el reporte quincenal, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup quincenal
```

Esto generará un reporte con el promedio de datos desde el primer día del mes hasta el día 15 del mes. Este reporte será exportado un archivo .xlsx llamado *Data_Quincenal_BBIP.xlsx*.

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ python3 -m scanbackup quincenal --dev
```

IMPORTANTE: Para esto se debe creado el archivo *.env.development* con las variables de entorno requeridas.

Además, se puede especificar la fecha para el que se desea generar el reporte, con el siguiente comando:

```
$ python3 -m scanbackup quincenal --literal
```

Esto generará un reporte desde de la quincena contando los 15 días hacia atrás iniciando desde el día actual.

Nota: La bandera *--dev* es válida para combinación.

4.2.19 Generación de Reporte Mensual

Para generar el reporte mensual, se debe ejecutar el siguiente comando:

```
$ python3 -m scanbackup mensual
```

Esto generará un reporte con el promedio de datos desde el primer día del mes hasta el día 30 del mes. Este reporte será exportado un archivo .xlsx llamado *Data_Mensual_BBIP.xlsx*.

También se puede ejecutar el comando en modo desarrollo, ejecutando el siguiente comando:

```
$ python3 -m scanbackup mensual --dev
```

IMPORTANTE: Para esto se debe creado el archivo *.env.development* con las variables de entorno requeridas.

Además, se puede especificar la fecha para el que se desea generar el reporte, con el siguiente comando:

```
$ python3 -m scanbackup mensual --literal
```

Esto generará un reporte desde de la mensual contando los 30 días hacia atrás iniciando desde el día actual.

Nota: La bandera *--dev* es válida para combinación.