**TABLA DE CONTENIDO**

[1. OBJETIVO 2](#_Toc101799499)

[2. ALCANCE 2](#_Toc101799500)

[3. DEFINICIONES 2](#_Toc101799501)

[4. LINEAMIENTOS GENERALES Y/O POLÍTICAS DE OPERACIÓN 3](#_Toc101799502)

[5. DESARROLLO 3](#_Toc101799503)

[5.1. ARQUITECTURA 3](#_Toc101799504)

[5.1.1 Contenedor 3](#_Toc101799505)

[5.1.2. Volúmenes 4](#_Toc101799506)

[5.1.3 Bases de Datos 4](#_Toc101799507)

[5.3. BACKUPS DE ARCHIVOS 5](#_Toc101799508)

[5.3.1. Comandos básicos 6](#_Toc101799509)

[5.4. BACKUPS DE BASES DE DATOS MYSQL 8](#_Toc101799510)

[5.5. BACKUPS DE BASES DE DATOS POSTGRESQL 8](#_Toc101799511)

[6. CONTROL DE CAMBIOS 9](#_Toc101799512)

### OBJETIVO

Definir los esquemas de backups dentro de los cuales se encuentra el esquema onpremise inicial para desarrollos vigentes y el esquema de backups en la nube, conforme los lineamientos vigentes de la entidad.

**Nota.** El presente instructivo profundizara en el esquema onpremise inicial para desarrollos vigentes ya que el esquema de backups en la nube se encuentra en el enlace <https://docs.google.com/document/d/17542UeO1Oo64kqQkqgIQskd2Y0Y344n5ym3pRpAkJJU/edit>

### ALCANCE

El instructivo inicia con la identificación del esquema que aplica conforme el desarrollo y finaliza con la ejecución de la actividad de backups permitiendo las copias de seguridad de las aplicaciones o desarrollos dando cumplimiento a los lineamientos vigentes de la entidad. El instructivo se encuentra dirigido al equipo de infraestructura del Grupo de Sistemas de información y Radiocomunicaciones – GSIR del Nivel Central quien tiene a cargo del mantenimiento y puesta en funcionamiento de las herramientas de la Entidad y sobre los cuales está la responsabilidad preservar la información generada en las mismas.

**Nota**: En este alcance se encuentran por fuera las copias que están asociados al banco de imágenes, copias de respaldo de los usuarios y directorio activo.

### DEFINICIONES

|  |  |
| --- | --- |
| **Archivo** | Archivos que son generados o asociados a las herramientas sobre las cuales se incluye información por parte de los usuarios. |
| **Backups** | Respaldo, copia de seguridad o copia de reserva a una copia de los datos originales de un sistema de información o de un conjunto de software ([archivos](https://concepto.de/archivo-informatico/), documentos, etc.) que se almacena en un lugar seguro o una región segura de la memoria del sistema, con el fin de poder volver a disponer de su [información](https://concepto.de/informacion/) en caso de que alguna eventualidad, accidente o desastre ocurra y ocasione su pérdida del [sistema](https://concepto.de/sistema-de-informacion/). |
| **CRONJOB** | Procedimiento que corre como una tarea en baclground en particular sobre un servidor Unix |
| **GCP** | Google Cloud Platform |
| **GIT** | Es un sistema de control de versiones distribuido de código abierto El control de versiones distribuido permite a los desarrolladores descargar un software, realizar cambios y subir la versión que han modificado. |
| **GITLAB** | Es un repositorio de información bajo GIT que permite almacenar el código fuente de las aplicaciones de la entidad. |
| **Onpremise** | Infraestructura local de la Entidad que no necesita recurrir a la nube o necesita de acceso a internet y es administrada por el personal Interno de Parques Nacionales Naturales de Colombia. |

### LINEAMIENTOS GENERALES Y/O POLÍTICAS DE OPERACIÓN

El esquema de backup de la entidad se realiza en 4 diferentes esquemas:

1. **Código fuente:** todas las aplicaciones de la entidad están compuestas por diferentes archivos que componen la herramienta. El código fuente de la entidad se encuentra en un repositorio centralizado bajo gitalb <https://gitlab.com/> y bajo el usuario de [serviciosweb@parquesnacionales.gov.co](mailto:serviciosweb@parquesnacionales.gov.co). Este código se encuentra en un esquema versionado bajo GIT y contiene toda la historia de desarrollo de la aplicación.
2. **Contenedores:** Todos los contenedores sobre los cuales funciona la infraestructura de la entidad se encuentran en un esquema de GITLAB pero las imágenes están disponibles sobre <https://hub.docker.com/u/serviciosweb> y bajo el usuario de [serviciosweb@parquesnacionales.gov.co](mailto:serviciosweb@parquesnacionales.gov.co). Estas imágenes contienen la base sobre la cual funcionan las aplicaciones de la entidad que están bajo docker en el esquema onpremise o en el esquema de GCP.
3. **Base de datos:** Las bases de datos con las que cuenta la entidad son Mysql y Postgresql, las cuales están bajo Mysql en un esquema en nube sobre GCP. Las bases de datos sobre Postgresql están en un esquema compuesta unas sobre el esquema onPremise y las demás sobre el esquema de GCP. Para poder hacer referencia a que aplicaciones y bases de datos funcionan sobre nube o en el esquema onpremise haga referencia al siguiente documento <https://drive.google.com/file/d/1QHsEZ0_s331XahPCoaStdmMqKlP0IdVK/view?usp=sharing> sobre el documento del catálogo de sistemas de información de la entidad.
4. **Activos:** Las aplicaciones contienen archivos que son cargados o subidos por los usuarios estos archivos hacen parte fundamental del funcionamiento de las diferentes herramientas. En los contenedores los archivos que persisten están asociados a volúmenes sobre el esquema onpremise o sobre el esquema de GCP.

### DESARROLLO

## 5.1. ARQUITECTURA

El esquema descrito en esta sección hace referencia a la información que está en el esquema onpremise de la entidad sobre el servidor con ip 50.13.

### 5.1.1 Contenedor

Se está utilizando el servicio de Docker llamado cronjobs para generar un contenedor el cual se encarga de las tareas de automatización de los backups. Este contenedor utiliza la versión 18.04 del sistema operativo Ubuntu.

### 5.1.2. Volúmenes

Al contenedor se le están montando 2 volúmenes, uno apuntando a la SAN donde se alojan los archivos y bases de datos de producción; y el otro, apuntando al Compellent que es donde se dispone del espacio en disco para almacenar los backups.

### 5.1.3 Bases de Datos

La entidad actualmente cuenta con dos motores de base de datos, uno Mysql donde se encuentran la mayoría de base de datos alfanuméricas y una base de datos Postgresql - Posting que posee las bases de datos geográficas. Las versiones de los motores de base de datos que se tienen actualmente son:

Para postgresql:

* 9.3.15, en esta base de datos se encuentra la información de:



* 9.5.6, en esta base de datos se encuentra la siguiente información:



* 10.7 versión de migración para el nuevo esquema geográfico:



El contenedor tiene visibilidad al contenedor de bases de datos de MySql y se configuró el usuario backups el cual tiene acceso de sólo lectura a todas las bases de datos del servicio de Docker mysql.

El contenedor tiene visibilidad al contenedor de bases de datos para las versiones de Postgresql y se configuró el usuario reader el cual tiene acceso de sólo lectura a todas las bases de datos del servicio de Docker de postgres.

**5.2. APLICACIONES BACKUPS**

Para hacer backups de los archivos se utiliza la aplicación Restic (<https://restic.readthedocs.io/en/latest/index.html>).

Restic permite la realización de backup incrementales, así que se puede volver a una versión antigua de determinado archivo o de la totalidad del backup. Varias ventajas adicionales de usar Restic es la rapidez, eficiencia, flexibilidad y seguridad que proporciona, además de que se puede utilizar tanto localmente, como mediante servidores SFTP y HTTP; adicionalmente con proveedores de almacenamiento en la nube como Backblaze B2, Amazon S3 y Google Cloud Storage. En cuanto a la seguridad, es obligatorio el uso de una contraseña que proteja el backup de archivo, la **contraseña es 48Xd2rAdJq.**

En cuanto a los backups de las bases de datos MySql, se está utilizando la herramienta nativa de MySql mysqldump (<https://dev.mysql.com/doc/refman/5.6/en/mysqldump.html>).

## 5.3. BACKUPS DE ARCHIVOS

La ruta donde está alojado el repositorio de Restic es:

En el contenedor cronjobs es /var/storage/backups/compellent/backups/assets y la ruta en el servidor 192.168.50.13 es /media/pnn/backups/assets.

**Importante:** Los backups se deben hacer siempre desde el contenedor cronjobs, nunca desde el servidor 192.168.50.13 debido a que las rutas de los archivos cambiarían y no quedarían como un backup incremental sino como un full backup.

### 5.3.1. Comandos básicos

A continuación, se describirán algunos de los comandos básicos de Restic para la realización y administración de backups

* + - 1. **Inicialización de un repositorio**

Para inicializar un repositorio de Restic utilizar el comando:

**restic init --repo <nombre\_repositorio>**

Este comando crea una carpeta con el nombre del **repo**, es obligatorio establecer una contraseña para el repositorio.

**Importante:** **Si la contraseña creada se pierde no se puede restaurar el backup.**

* + - 1. **Hacer un backup de un archivo o directorio**

Para realizar un backup de un archivo o directorio ya sea full (primera vez) o incremental utilizar el siguiente comando:

**restic -r <directorio\_del\_repositorio> [--verbose] backup --tag <tag> <archivo\_o\_directorio> [--exclude-file=excludes.txt].**

El siguiente es un ejemplo del archivo excludes.txt el cual muestra cómo debe ser la sintaxis para la exclusión de archivos que se agregaran al snapshot:

**# exclude go-files**

**\*.go**

**# exclude foo/x/y/z/bar foo/x/bar foo/bar**

**foo/\*\*/bar**

* + - 1. **Hacer un backup desde stdin**

Para hacer backups de las bases de datos de MySql, el siguiente comando es útil porque se salta el paso intermedio de tener que guardar el resultado del comando mysqldump en un archivo:

**mysqldump [...] | restic -r <directorio\_del\_repositorio> backup --stdin --stdin-filename <archivo\_como\_va\_a\_quedar\_almacenado\_en\_restic>**

* + - 1. **Listar snapshots de un repositorio**

El concepto de snapshot es el equivalente a un commit en GIT, cuando se hace un full backup por primera vez, eso es un snapshot, cada vez que se hace un backup incremental, también es un snapshot.

Comando para listar los snapshots es:

**restic -r <directorio\_del\_repositorio> snapshots**

* + - 1. **Revisar la integridad y consistencia de un repositorio**

Se recomienda revisar la integridad y consistencia de un repositorio, así se asegura que en el momento de restaurar un snapshots, los archivos sean restaurados correctamente.

El comando es el siguiente:

**restic -r <directorio\_del\_repositorio> check [--read-data]**

* + - 1. **Restaurar el snapshot específico**

Para restaurar el último snapshot utilizar el comando:

**restic -r <directorio\_del\_repositorio> restore latest --target <directorio\_destino>**

El anterior comando crea el directorio destino si no existe.

* + - 1. **Restaurar un snapshot específico**

Primero se debe listar los snapshots para obtener el snapshot\_id que se desea restaurar.

Comando para restaurar un snapshot específico es:

**restic -r <directorio\_del\_repositorio> restore <snapshot\_id> --target <directorio\_destino>**

Comando que crea el directorio destino si no existe.

* + - 1. **Restaurar un archivo o directorio específico de un snapshot**

Primero se debe listar los snapshots para obtener el snapshot\_id que se desea restaurar.

**restic -r <directorio\_del\_repositorio> restore <snapshot\_id> --target <directorio\_destino> --include <archivo\_o\_directorio\_que\_se\_va\_a\_restaurar>**

El comando crea el directorio destino si no existe.

* + - 1. **Imprimir el contenido de un archivo por el stdout**

Para restaurar un backup de una base de datos de MySql, el comando se salta el paso intermedio de tener que guardar el resultado de la restauración del script de la base datos en un archivo, utilizar el siguiente comando:

**restic -r <directorio\_del\_repositorio> dump <snapshot\_id> <archivo\_de\_script\_en\_restic> | mysql.**

* + - 1. **Imprimir el contenido de la estructura de un directorio**

El siguiente comando equivalente a hacer un ls recursivo de un directorio de un snapshot de Restic

**restic -r <directorio\_del\_repositorio> dump <directorio\_en\_restic> <snapshot\_id> > restore.tar**

Según la documentación de Restic, la salida será en formato tar pero en la práctica no funciona de esta manera.

## 5.4. BACKUPS DE BASES DE DATOS MYSQL

En el contenedor cronjobs de Docker se encuentra un script PHP el cual genera dinámicamente un archivo sh con todos los comandos mysqldump necesarios para hacerle backup a todas las bases de datos que se encuentren en el contenedor mysql de Docker. Este script PHP se ejecuta mediante un cronjob que también ya está configurado en el mismo contenedor Docker.

2 minutos después de generado este archivo sh, otro cronjob ejecuta este archivo sh el cual guarda los scripts sql resultantes en el volumen del Compellent, en una carpeta creada por el mismo script con la fecha actual como nombre.

El primero de estos 2 cronjobs se ejecuta a las 2:28 am y el segundo cronjob se ejecuta a las 2:30 am todos los días.

La ruta donde quedan los backups desde el contenedor cronjobs es /var/storage/backups/compellent/backups/dbs y la ruta desde el servidor 192.168.50.13 es /media/pnn/backups/dbs.

## 5.5. BACKUPS DE BASES DE DATOS POSTGRESQL

En el contenedor cronjobs de Docker se encuentra un script el cual genera los comandos pg\_dump necesarios para hacerle backup a todas las bases de datos que se encuentren en el contenedor postgresql de Docker. Este script PHP se ejecuta mediante un cronjob que también ya está configurado en el mismo contenedor Docker.

2 minutos después de generado este archivo sh, otro cronjob ejecuta este archivo sh el cual guarda los scripts sql resultantes en el volumen del Compellent, en una carpeta creada por el mismo script con la fecha actual como nombre.

El primero de estos 2 cronjobs se ejecuta a las 2:28 am y el segundo cronjob se ejecuta a las 2:30 am todos los días.

La ruta donde quedan los backups desde el contenedor cronjobs es /var/storage/backups/compellent/backups/dbs/postgresql y la ruta desde el servidor 192.168.50.13 es /media/pnn/backups/dbs/postgresql.

El esquema de backups que actualmente se utiliza se apoya en el documento de la infraestructura en la nube el cual se encuentra disponible en el siguiente link. <https://docs.google.com/document/d/17542UeO1Oo64kqQkqgIQskd2Y0Y344n5ym3pRpAkJJU/edit>

### CONTROL DE CAMBIOS

|  |  |  |
| --- | --- | --- |
| **FECHA DE VIGENCIA VERSIÓN ANTERIOR** | **VERSIÓN ANTERIOR** | **MOTIVO DE LA ACTUALIZACIÓN** |
| 21/04/2022 | 1 | De acuerdo a la resolución 310 del 2021 donde se conforman los Grupo Internos de Trabajo se ajustan los responsables de revisar y aprobar el documento |
| 25/04/2022 | 2 | Se re codifica el documento de acuerdo con el nuevo mapa de procesos, actualizando el código. El documento por cargue inicial en la aplicación tecnológica reinicia desde el código 1. Para consultar los obsoletos ver matriz de armonización documentos del SGI al nuevo mapa de procesos https://drive.google.com/drive/u/1/folders/1Tu2ChzlvgSaXxc10UpqzX-SVhu095Kvv  Las fechas y nombres que aparecen en el control de revisión y aprobación (Créditos), obedecen a las fechas registradas en el documento antes de la migración del documento al nuevo mapa de procesos. |

|  |  |  |
| --- | --- | --- |
| **CRÉDITOS** | | |
| Elaboró | Nombre | Alan Aguia  Sandra Milena Gómez |
| Cargo | Profesional Especializado - Contratista |
| Fecha | 21/04/2022 |
| Revisó | Nombre | Carlos Arturo Sáenz Barón |
| Cargo | Coordinador Grupo de Tecnologías de la Información y las Comunicaciones. |
| Fecha: | 21/04/2022 |
| Aprobó | Nombre | Carlos Arturo Sáenz Barón |
| Cargo | Coordinador Grupo de Tecnologías de la Información y las Comunicaciones. |
| Fecha: | 21/04/2022 |