

Base de Datos II (IC4302) – Semestre 2, 2023

Tarea Corta #1

Jesús Andrés Cortés Álvarez – 2021579439

Aaron Ortiz Jimenez – 2022437529

Joctan Antonio Porras Esquivel - 2021069671

Justin Acuña Barrantes - 2018093451

David Suárez Acosta – 2020038304

Instrucciones de funcionamiento

Para realizar los scripts de cada base de datos es necesario subir la base de datos, en este caso para visualizar su funcionamiento se deberá de ingresar a la carpeta llamada databases, seleccionar el archivo "values.yaml", seguidamente al entrar el archivo el apartado de enabled debe ser cambiado a "true" en el caso de que se encuentre en "false" la base de datos con la que desea trabajar, si alguna de las otra bases de datos se encuentra en "true", se recomienda cambiar a al valor "false", esto para evitar que las de demás bases de datos se ejecuten en simultaneo y únicamente ejecutar la deseada.

Luego deberá de dirigirse a la carpeta llamada "backups", ingresar al archivo "values.yaml", y realizará de igual forma a como se explicó anteriormente, buscar la base de datos en la cual desea trabajar y coloca en "true" el apartado enabled o en caso opuesto "false", según desee el usuario, de igual forma se recomienda no colocar varias en true el apartado de enabled de las diferentes bases de datos para evitar conflictos.

- **Base de datos Neo4j**

En la siguiente imagen se presenta el caso de utilizar la base de datos de Neo4j.

```
neo4j:
  enabled: true
  neo4j:
    name: ic4302
    edition: "enterprise"
    acceptLicenseAgreement: "yes"
    offlineMaintenanceModeEnabled: false
    resources:
      memory: "2Gi"
  volumes:
    data:
      mode: defaultStorageClass
      defaultStorageClass:
        requests:
          storage: 2Gi
```

Para probar el funcionamiento del backup y restore en esta base de datos, deberá hacer una acción similar a la mencionada anteriormente, dependiendo de la acción de backup o restore, colocará true en el espacio señalado, esto para visualizar su funcionamiento.

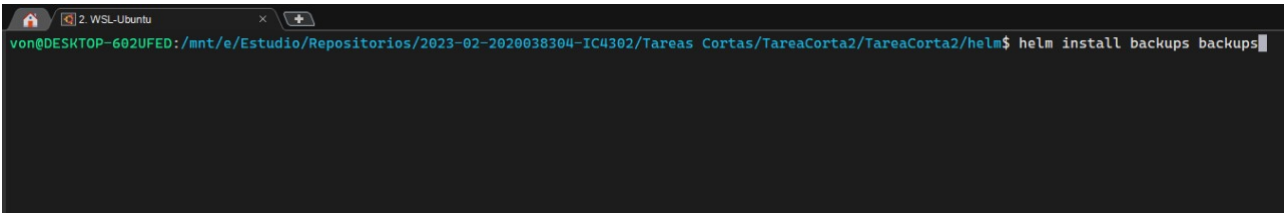
```
neo4j:
  enabled: false
  restore: true
  backup: false
  config:
    namespace: default
    connectionString: databases-admin.default.svc.cluster.local:6362
    bucketName: tec-ic4302-02-2023
    path: 2022437529/neo4j
    maxBackups: 3
    secret: ic4302-auth
    name: neo4j
    schedule: "0 */12 * * *"
    diskSize: 2
    storageClass: hostpath
    provider: aws
    type: backup
```

- **Base de datos Elasticsearch**
- **Base de datos Couchdb**

Para probar el funcionamiento del backup y restore en esta base de datos, deberá hacer una acción similar a la mencionada anteriormente, dependiendo de la acción de backup o restore, colocará true en el espacio señalado, esto para visualizar su funcionamiento.

```
couchdb:
  enabled: true
  restore: true
  backup: false
  config:
    namespace: default
    connectionString: databases-admin.default.svc.cluster.local:5984
    bucketName: tec-ic4302-02-2023
    path: 2022437529/couchdb
    maxBackups: 3
    secret: databases-couchdb
    name: couchdb
    schedule: "0 */12 * * *"
    diskSize: 2Gi
    storageClass: hostpath
    provider: aws
    type: backup
```

Seguidamente en una terminal se ejecuta el comando para instalar el Helm chart(Se logra apreciar en la siguiente imagen) y se visualiza como el job seleccionado anteriormente genera el backup o el restore respectivamente, es posible visualizar la ejecución con la herramienta Lens.



En la siguiente imagen mediante la aplicación Lens se logra observar que cuando aparece en pantalla la palabra "Succeeded", significa que el job se ejecutó correctamente.

<input type="checkbox"/>	couchdb-backup-0-9j9pb	default	<div><div></div></div>	N/A	N/A	0	job	docker-de	BestEffort	96s	Succeeded
<input type="checkbox"/>	databases-couchdb-0	default	<div><div></div></div>	N/A	N/A	1	StatefulSet	docker-de	BestEffort	18h	Running
<input type="checkbox"/>	databases-couchdb-1	default	<div><div></div></div>	N/A	N/A	1	StatefulSet	docker-de	BestEffort	18h	Running
<input type="checkbox"/>	databases-couchdb-2	default	<div><div></div></div>	N/A	N/A	1	StatefulSet	docker-de	BestEffort	18h	Running

• **Base de datos Postgresql**

En la siguiente imagen se presenta la creación de la base de datos de Postgresql.

```
38 postgresql:
39   enabled: true
40   auth:
41     postgresPassword: root
42     database: postgresDB
43     enablePostgresUser: true
44     username: root
45     password: root
46   primary:
47     name: postgresDB
48     user: root
49     postgresPassword: root
50     persistence:
51       enabled: true
52       size: 2Gi
```

Como se mencionó al inicio se coloca en true en el apartado de enabled la acción que desea realizar en la base de datos.

```
68 postgresql:
69   enabled: true
70   restore: true
71   config:
72     namespace: default
73     connectionString: databases-admin.default.svc.cluster.local:5432
74     bucketName: tec-ic4302-02-2023
75     path: 2022437529/postgresql
76     maxBackups: 3
77     secret: databases-postgresql
78     name: postgresql
79     schedule: "0 */12 * * *"
80     diskSize: 2Gi
81     storageClass: hostpath
82     provider: aws
83     type: backup
```

- **Base de datos MongoDB**

En la siguiente imagen se presenta la creación de la base de datos de MongoDB.

```
mongo:
  enabled: false
  restore: false
  backup: false
  config:
    namespace: default
    connectionString: databases-mongodb.default.svc.cluster.local:27017
    bucketName: tec-ic4302-02-2023
    path: 2022437529/mongodb
    maxBackups: 3
  secret: databases-mongodb
  name: mongo
  schedule: "0 */12 * * *"
  diskSize: 2
  storageClass: hostpath
  provider: aws
  type: backup
```

Como se mencionó al inicio se coloca en true en el apartado de enabled la acción que desea realizar en la base de datos, en este caso es MongoDB.

```
mongodb:
  enabled: false
elastic:
  enabled: false
  version: 8.6.1
  replicas: 1
  name: ic4302
kibana:
  enabled: false
  version: 8.6.1
  replicas: 1
  name: ic4302
```

Pruebas realizadas

- **Base de datos MongoDB**

Para realizar las pruebas, se ingresa al pod de mongo llamado "databases-mongodb" que es el pod de la base de datos de mongo, una vez dentro se ejecuta el comando "mongosh" junto con el URI, root y password para conectarme a mongo.

Una vez dentro se crea una nueva base llamada "prueba" y dentro una nueva colección "new_collection", en esta se inserta un simple documento con las variables "profe" y "materia".

```

as/TareaCorta2/TareaCorta2/helm$ kubectl exec -it databases-mongodb-59dbc55f8c-pq4
vx -- /bin/bash
I have no name!@databases-mongodb-59dbc55f8c-pq4vx:/$ mongosh --host databases-mon
godb.default.svc.cluster.local:27017 -u root -p 8GdyBZgoPj
Current Mongosh Log ID: 65387ee5160f64c024ceec56
Connecting to:      mongodb://<credentials>@databases-mongodb.default.svc.clus
ter.local:27017/?directConnection=true&appName=mongosh+1.6.2
Using MongoDB:      6.0.4
Using Mongosh:      1.6.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2023-10-24T22:40:14.030+00:00: Using the XFS filesystem is strongly recommended
with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-10-24T22:40:15.395+00:00: /sys/kernel/mm/transparent_hugepage/enabled is '
always'. We suggest setting it to 'never'
2023-10-24T22:40:15.395+00:00: vm.max_map_count is too low
-----

-----
Enable MongoDB's free cloud-based monitoring service, which will then receive a
nd display
metrics about your deployment (disk utilization, CPU, operation statistics, etc
).

The monitoring data will be available on a MongoDB website with a unique URL ac
cessible to you
and anyone you share the URL with. MongoDB may use this information to make pro
duct
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFree
Monitoring()
-----

test> show dbs
admin    116.00 KiB
config   72.00 KiB
local    72.00 KiB
test> use prueba
switched to db prueba
prueba> db.new_collection.insertOne({
...   profe: "Nereo",
...   materia: "BASES_2"
... })
{
  acknowledged: true,
  insertedId: ObjectId("6538801f0c466f7bff1c2be8")
}
prueba> db.new_collection.find()
[
  {
    _id: ObjectId("6538801f0c466f7bff1c2be8"),
    profe: 'Nereo',
    materia: 'BASES_2'
  }
]

```

A continuación se realiza una ejecución del job de Backup usando un archivo nuevo temporal llamado "temp.yaml" que contiene el job del backup solamente para no tener que correr todo el cluster y borrar el pod de ejecución actual.

```

david@Joe:/mnt/d/Tareas David/TEC/Semestre 8/Bases de Datos II/Bases_2/Tareas Cortas/TareaCorta2/TareaCorta2/helm/backups/templates$ kubectl create -f temp.yaml
job.batch/mongo-backup-1 created
david@Joe:/mnt/d/Tareas David/TEC/Semestre 8/Bases de Datos II/Bases_2/Tareas Cortas/TareaCorta2/TareaCorta2/helm/backups/templates$ kubectl get jobs
NAME                                COMPLETIONS   DURATION   AGE
mongo-backup-0                     1/1            52s        4h21m
mongo-backup-1                     0/1            4s         4s
mongo-backup-28303200              1/1            50s        94m
mongo-restore                      1/1            2m46s      4h21m

```

Lo siguiente es que se borra la colección "new_collection" y simultáneamente se ejecuta el job del restore, de la misma manera que se ejecutó el job de backup, con un nuevo archivo temporal llamado "temp2.yaml".

```

david@Joe:/mnt/d/Tareas David/TEC/Semestre 8/Bases de Datos II/Bases_2/Tareas Cortas/TareaCorta2/TareaCorta2/helm/backups/templates$ kubectl create -f temp2.yaml
job.batch/mongo-restore-1 created
david@Joe:/mnt/d/Tareas David/TEC/Semestre 8/Bases de Datos II/Bases_2/Tareas Cortas/TareaCorta2/TareaCorta2/helm/backups/templates$ kubectl get jobs
NAME                                COMPLETIONS   DURATION   AGE
mongo-backup-0                     1/1            52s        4h29m
mongo-backup-1                     1/1            54s        8m8s
mongo-backup-28303200              1/1            50s        102m
mongo-restore                      1/1            2m46s      4h29m
mongo-restore-1                    0/1            4s         4s

```

Por último, y como se observa en la imagen, a pesar de haberse borrado la coleccion, esta se restaura gracias al job de restore que se ejecutó.

```

as/TareaCorta2/TareaCorta2/helm$ kubectl exec -it databases-mongodb-59dbc55f8c-pq4
vx -- /bin/bash
I have no name!@databases-mongodb-59dbc55f8c-pq4vx:/$ mongosh --host databases-mon
godb.default.svc.cluster.local:27017 -u root -p 8GdyBZgoPj
Current Mongosh Log ID: 65387ee5160f64c024ceec56
Connecting to:      mongodb://<credentials>@databases-mongodb.default.svc.clus
ter.local:27017/?directConnection=true&appName=mongosh+1.6.2
Using MongoDB:      6.0.4
Using Mongosh:      1.6.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2023-10-24T22:40:14.030+00:00: Using the XFS filesystem is strongly recommended
with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-10-24T22:40:15.395+00:00: /sys/kernel/mm/transparent_hugepage/enabled is '
always'. We suggest setting it to 'never'
2023-10-24T22:40:15.395+00:00: vm.max_map_count is too low
-----

-----
Enable MongoDB's free cloud-based monitoring service, which will then receive a
nd display
metrics about your deployment (disk utilization, CPU, operation statistics, etc
).

The monitoring data will be available on a MongoDB website with a unique URL ac
cessible to you
and anyone you share the URL with. MongoDB may use this information to make pro
duct
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFree
Monitoring()
-----

test> show dbs
admin    116.00 KiB
config   72.00 KiB
local    72.00 KiB
test> use prueba
switched to db prueba
prueba> db.new_collection.insertOne({
...   profe: "Nereo",
...   materia: "BASES_2"
... })
{
  acknowledged: true,
  insertedId: ObjectId("6538801f0c466f7bff1c2be8")
}
prueba> db.new_collection.find()
[
  {
    _id: ObjectId("6538801f0c466f7bff1c2be8"),
    profe: 'Nereo',
    materia: 'BASES_2'
  }
]

```

- **Base de datos Neo4j**

A continuación, se presenta como se realizan las pruebas correspondientes para la base de datos de Neo4j.


```
Completed 6.0 KiB/6.0 KiB (13.3 KiB/s) with 1 file(s) remaining
download: s3://tec-ic4302-02-2023/2022437529/neo4j/movies-2023-10-19T00-29-47.backup to ../data/backups/movies-2023-10-19T00-29-47.backup
```

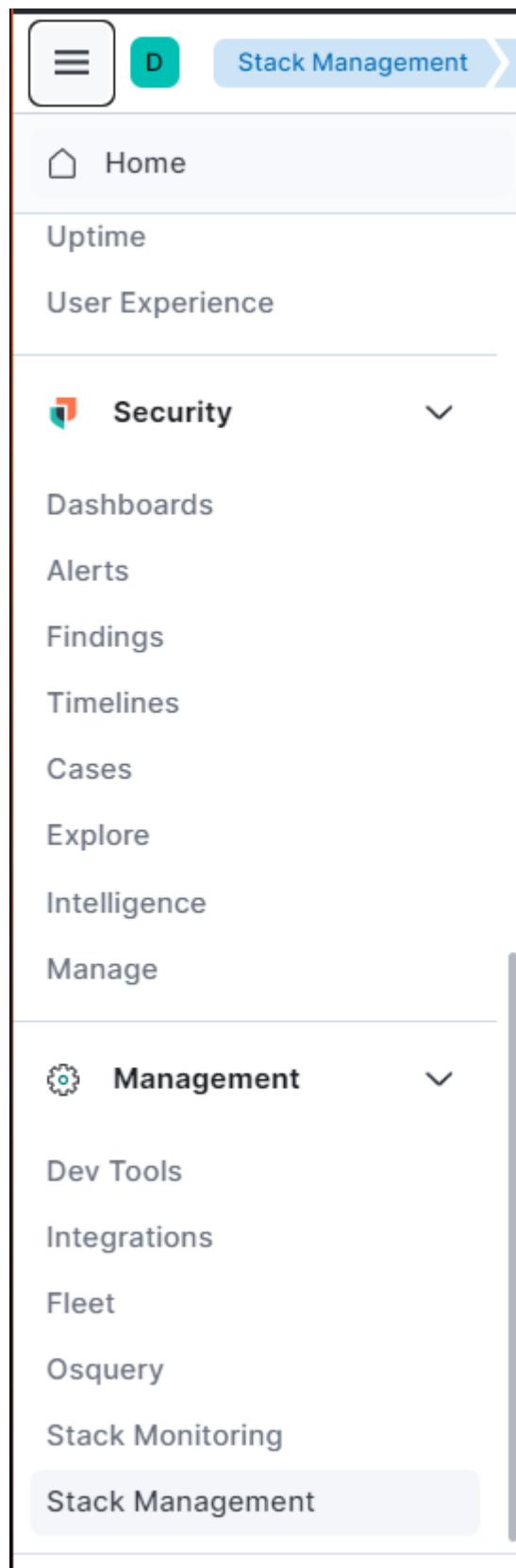
Se logra observar el resultado del primer script, que obtiene los datos desde el bucket s3 hacia la maquina en la que se está trabajando.

```
failed to delete the specified database 'movies': Database does not exist.
command terminated with exit code 1
2023-10-24 18:10:07.768+0000 INFO [c.n.b.v.r.RestoreDatabaseExecutor] Restoring database 'movies' from backup artifact chain:
../backups/movies-2023-10-19T00-29-47.backup
2023-10-24 18:10:07.775+0000 INFO [c.n.b.v.r.RestoreDatabaseExecutor] Unpacking full backup artifact: file:///backups/movies-2023-10-19T00-29-47.backup into directory: /backups/temp13782067374332339754
2023-10-24 18:10:08.579+0000 INFO [c.n.b.v.r.RestoreDatabaseExecutor] Restoring from unpacked backup artifact chain: /backups/temp13782067374332339754
2023-10-24 18:10:08.669+0000 INFO [c.n.b.v.r.RestoreDatabaseExecutor] You need to execute /data/scripts/movies/restore_metadata.cypher. To execute the file use cypher-shell command with parameter 'movies'
2023-10-24 18:10:08.676+0000 INFO [c.n.b.v.r.RestoreDatabaseExecutor] Start recovering database 'movies'.
2023-10-24 18:10:08.780+0000 INFO [c.n.b.v.r.RestoreDatabaseExecutor] Finish recovering database 'movies', took 109ms.
Restore of database 'movies' from path="/backups/movies-2023-10-19T00-29-47.backup" completed successfully.
```

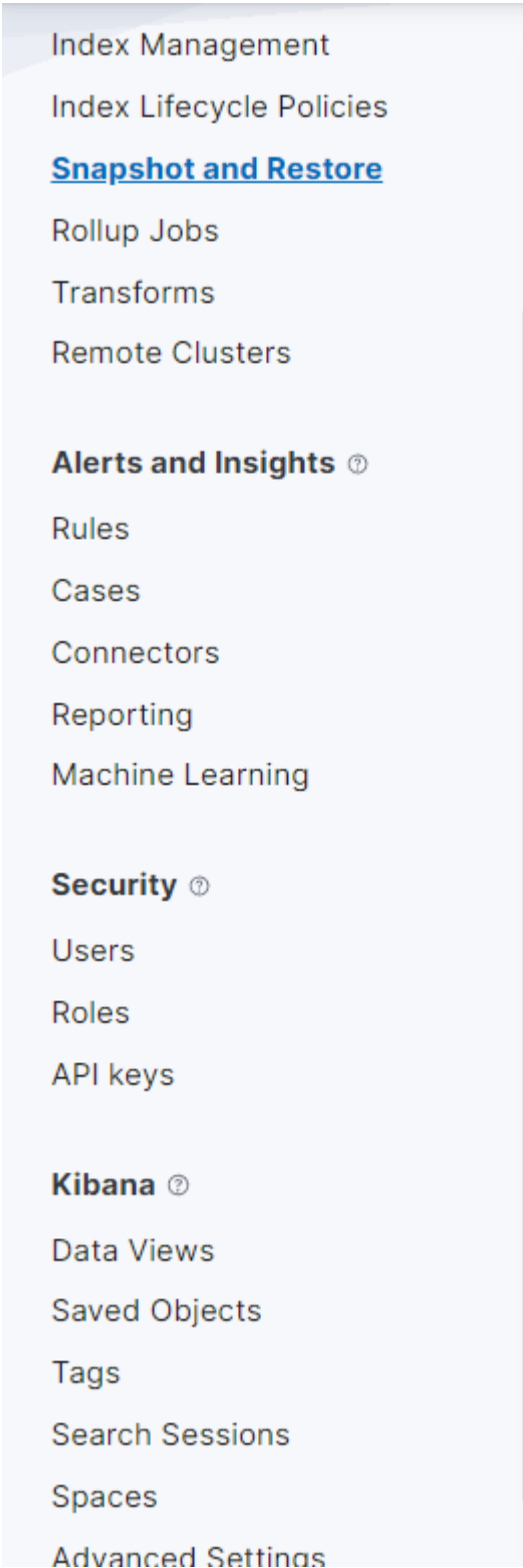
En la imagen anterior se observa que se completó el proceso para realizar el restore.

- **Base de datos ElasticSearch**

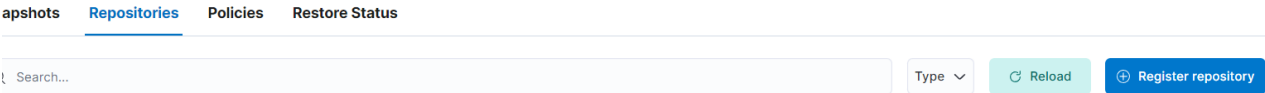
En las siguientes imágenes se presenta como dirigirse a la sección de snapshot, primero se debe de dirigir al cuadro de "menú" y luego se dirige al espacio de "Stack Managment".



Al dar clic se le presentará el siguiente menú, donde deberá de dirigirse al apartado de "Snapshot and Restore".



Seguidamente se presentan imágenes correspondientes a la creación del repositorio, en este caso para la creación se pone de nombre el carnet estudiantil, el servicio seria "aws s3". Para la siguiente imagen, se define el cliente como default, el bucket que fue el asignado y se logra apreciar en la imagen, al finalizar se preciosa el botón de registrar.



Register repository

Repository name


A unique name for the repository.

Name

2022437529

Repository type


Storage location for your snapshots. [Learn more about repository types.](#)



Azure

[Learn more](#)


Select



Google Cloud Storage

[Learn more](#)


Select



AWS S3

[Learn more](#)


Select



Shared file system

[Learn more](#)

Select



Read-only URL

[Learn more](#)

Select

Register repository

'2022437529' settings

[AWS S3 repository docs](#)

Client

The name of the AWS S3 client.

Client

default

Bucket

The name of the AWS S3 bucket to use for snapshots.

Bucket (required)

tec-ic4302-02-2023

Base path

The bucket path to the repository data.

Base path

2022437529-elastic

Snapshot compression

Compresses the index mapping and setting files for snapshots. Data files are not compressed.

☒ Compress snapshots

Chunk size

Breaks files into smaller units when taking snapshots.

Chunk size

Accepts byte size units, such as **1g** , **10mb** , **5k** , or **1024B** . Defaults to unlimited.

Server-side encryption

Encrypts files on the server using AES256 algorithm.

☐ Server-side encryption

Para la creación del snapshot, se debe ir al apartado de "Create Policy", se le coloca un nombre, el nombre del snapshot, se asignan las configuraciones que va a tener el snapshot, políticas de retención y la duración, para finalizar se presiona el botón de "Create Policy"

Create policy

1

2

3

4

Logistics

Snapshot settings

Snapshot retention

Review

Logistics docs

Policy name

A unique identifier for this policy.

Name

2022437529-elastic

Snapshot name

The name for the snapshots. A unique identifier is automatically added to each name.

Snapshot name

2022437529-elastic

Supports date math expressions. [Learn more.](#)

Repository

The repository where you want to store the snapshots.

Repository

2022437529

Schedule

The frequency at which to take the snapshots.

Schedule

0 30 12 * * ?

Use cron expression. [Learn more.](#)

[Create basic interval](#)

En la siguiente imagen se observa que ya fue creado el policy, seguidamente se presiona el botón ubicado al lado derecho para correrlo, y ya realizaría el backup, Al finalizar ya aparecerían los snapshots.

<input type="checkbox"/> Policy ↑	Snapshot name	Repository	Schedule	Retention	Next snapshot	Actions
<input type="checkbox"/> 2022437529-elastic	2022437529-elastic	2022437529	0 30 12 * * ?	✓	Oct 25, 2023 6:30 AM CST	▶ ✎ 🗑

En la siguiente ya se logra observar la creación de los snapshots de la base de datos.

Repository [▼](#)

[↻ Reload](#)

<input type="checkbox"/> Snapshot	Repository	Indices	Shards	Failed shards	Date created ↓	Duration	Actions
<input type="checkbox"/> 2022437529-elastic-3cnisyret2krfxcnulwt5g	2022437529	4	12	0	Oct 24, 2023 12:24 PM CST	<div></div>	📄 🗑
<input type="checkbox"/> elastic-2022437529-yakkbzz5tdw8mcixd8haaq	2022437529	4	12	0	Oct 23, 2023 10:26 PM CST	82s	📄 🗑

En el apartado del restore, se debe indicar el índice, donde se desmarca las opciones que no se necesitan para el restore.

Restore details

[Snapshot and Restore docs](#)

⚠ This snapshot contains data streams

Each data stream requires a matching index template. Please ensure any restored data streams have a matching index template. You can restore index templates by restoring the global cluster state. However, this may overwrite existing templates, cluster settings, ingest pipelines, and lifecycle policies. [Learn more](#) about restoring snapshots that contain data streams.

Data streams and indices

Creates new data streams and indices if they don't exist. Opens existing indices, including backing indices for a data stream, if they are closed and have the same number of shards as the snapshot index.

☐ ☒ All data streams and indices

Select data streams and indices

Use index patterns

Filter options

.logs-deprecation.elasticse...

Data stream

↔

ilm-history-5

Data stream

.kibana-event-log-8.6.1-000001

✓ pokemon

Se indican las configuraciones necesarias, como se indica en la siguiente imagen.

1 index and 0 data streams will be restored. [Deselect all](#)

Rename data streams and indices

Renames data streams and indices on restore. Ensure that a matching index template exists for renamed data streams.

☐ ☒ Rename data streams and indices

Partial restore

Allows restore of indices that don't have snapshots of all shards.

☐ ☒ Partial restore

Restore global state

Restores the global cluster state as part of the snapshot.

☐ ☒ Restore global state

Restore feature state

Restores the configuration, history, and other data stored in Elasticsearch by a feature such as Elasticsearch security.

☐ ☒ Restore feature state from

Restore aliases

Restores index aliases along with their associated indices.

☒ ☐ Restore aliases

> Next

En la siguiente imagen se verifica que se cumplen las configuraciones mencionadas anteriormente.

14 / 18

Restore 'elastic-2022437529-yakkbzz5tdw8mcixd8haaq'

1

Logistics

2

Index settings

3

Review

Index settings

Backing indices

These settings also apply to the backing indices of data streams.

Modify index settings

Overrides index settings during restore. [Learn more.](#)

☐ Modify index settings

Reset index settings

Resets selected settings to default during restore. [Learn more.](#)

☐ Reset index settings

< Back

> Next

Cuando se ejecuta el restore, se observa que se completó correctamente la recuperación de los datos.

Snapshot and Restore

Use repositories to store and recover backups of your Elasticsearch indices and clusters.

SnapshotsRepositoriesPoliciesRestore Status

Refresh data every 30 seconds

Index	Status	Last activity	Shards completed	Shards in progress
pokemon	Complete	Oct 24, 2023 12:26 PM CST	1	0

Rows per page: 20

Base de datos Couchdb

A continuación, se presenta la base de datos que contiene datos de prueba basado en algunas películas.

_id	año	director	genero	sinopsis
a19ec5d87fd6b7fe121e6146c40067cc	1942	Michael Curtiz	Drama	Una historia de amor en Casablanca durante l...
a19ec5d87fd6b7fe121e6146c4008268	1982	Ridley Scott	Ciencia Ficción	Un cazador de replicantes en un futuro distópi...
a19ec5d87fd6b7fe121e6146c4009fad	2007	Greg Mottola	Comedia	Dos amigos intentan hacer que su última fiest...
a19ec5d87fd6b7fe121e6146c400c193	1995	John Lasseter	Animación	Las aventuras de juguetes que cobran vida c...
a19ec5d87fd6b7fe121e6146c400e6b2	2015	George Miller	Acción	Un viaje épico en un mundo posapocalíptico l...

Seguidamente en la siguiente imagen se presenta la subida del backup al bucket de AWS.

```
Completed 2.0 KiB/2.0 KiB (4.3 KiB/s) with 1 file(s) remaining
upload: ../couchdbdump/202310250117/backup.json to s3://tec-ic4302-02-2023/2022437529/couchdb/backup.json
2023-10-25 01:17:32      2034 backup.json
```

En la siguiente imagen se presenta la respuesta de la consulta del script para el restore de la base de datos con un caso exitoso. El espacio ok: true indica que el restablecimiento del documento fue exitoso.

```
% Total % Received % Xferd Average Speed Time Time Time Current
      Dload  Upload   Total             Spent    Left   Speed

 0   0    0     0     0      0     0      0  0:00:00 --:--:-- --:--:--    0
2052 100 477 100 1575 9648 31858 --:--:-- --:--:-- --:--:-- 41877
[{"ok":true,"id":"56524878b61296b69da597ae2b012a27","rev":"2-2122b6d447c1fdd54fa2ca91cb2c3305"}, {"ok":true,"id":"56524878b61296b69da597ae2b01496b","rev":"2-9fac180b32a5e7be5c4f761f48809c3"}]
```

Base de datos Postgresql

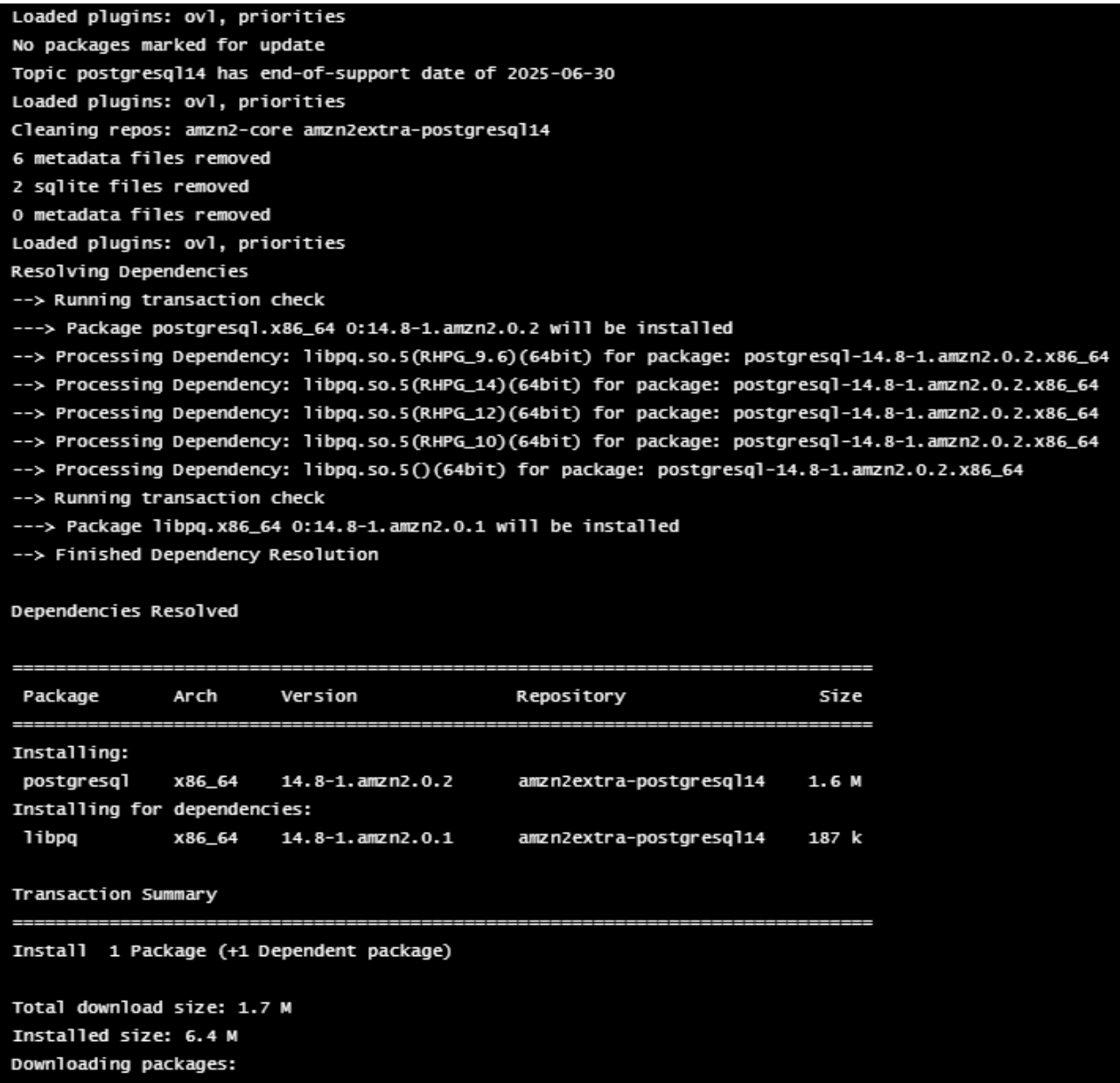
Cuandp se ejecuta el comando "helm install backup" con postgresql debe aparecer de la siguiente manera, cuando se realiza el backup.



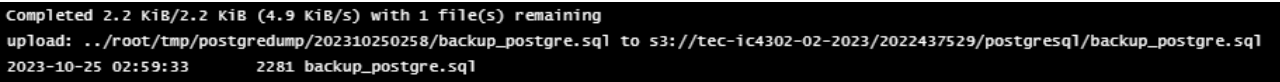
Cuando se completa correctamente aparece la palabra succeeded.



En el apartado de logs se observa que se descargan las dependencias necesarias.



Al final de los logs, se puede observar que se cargó correctamente el archivo al bucket de aws.



En la siguiente imagen es posible observar la informacion del bucket de aws, se obtiene que este es el contenido del archivo de backup realizado.


```

bash-4.2# aws s3 cp s3://tec-ic4302-02-2023/2022437529/postgresql/backup_postgre.sql -
--
-- PostgreSQL database dump
--

-- Dumped from database version 14.5
-- Dumped by pg_dump version 14.8

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

```

Se logran observar los datos respaldados.

```

COPY public.clientes (clienteid, nombre, apellido) FROM stdin;
1      Juan      Pérez
2      María    González
3      Pedro    Sánchez
4      Luis     Martínez
5      Ana      Ramírez
6      José     López
\.

```

Al visualizar la información que se encontraba dentro de la base de datos a la que se le realizó el backup, es posible ver que los datos coinciden

```

D:/.../TC2/RepoFuncional/2023-02-2020038304-IC4302 29ms 1
21:05:35 kubectl exec -it databases-postgresql-0 -c postgresql -- /bin/bash
I have no name!@databases-postgresql-0:/$ psql -U root -d postgresDB
Password for user root:
psql (14.5)
Type "help" for help.

postgresDB=> select * from Clientes;
 clienteid | nombre | apellido
-----+-----+-----
          1 | Juan   | Pérez
          2 | María  | González
          3 | Pedro  | Sánchez
          4 | Luis   | Martínez
          5 | Ana    | Ramírez
          6 | José   | López
(6 rows)

postgresDB=>

```

Recomendaciones

- En el caso de la base de datos de Neo4j, realizar una búsqueda completa en la documentación y emplear las herramientas que posee Neo4j para realizar backups, donde se incluye el Cron Job, en el

caso del restore es recomendable utilizar dos contenedores, que obtenga los datos del cloud provider y otro que ejecute el comando dentro del pod.

- Es recomendable utilizar la herramienta que tiene el propio couch para la administración de la base de datos, este se puede encontrar en el endpoint llamado `/_utils`. Nos permite hacer todas las operaciones CRUD dentro de una interfaz amigable y sin utilizar programas externos.
- En caso de manejo de errores y seguimiento de la ejecución de los scripts, es muy útil observar mediante la herramienta Lens, si ingresamos al job y observamos los logs, se puede realizar un seguimiento necesario para observar el flujo del script.
- PostgreSQL fue una base de datos un poco diferente a las demás, ya que no requería descargar muchos recursos o buscar en repos externos, recomendamos hacer la conexión a la base de datos mediante el nombre del servicio para no acceder mediante IP
- Se recomienda el uso de los logs de los JOBs para seguir los movimientos que se dan dentro de los pods, dan información muy útil.

Conclusiones

- La base de datos PostgreSQL puede resultar sencilla en cuanto a tareas de backup y restore, ya que los comandos que se requieren son fáciles en sintaxis, sin embargo, si no se tiene el cuidado necesario puede complicarse mucho.
- El realizar tareas de backup y restore son muy distintas en cuanto a dificultad, ya que puede ser bastante complicada en algunas, lo complicado puede resultar equivalente a la cantidad de documentación que exista.
- Como conclusión se observa la importancia de realizar backups y la posibilidad de almacenarlo en diferentes lugares, en nuestro caso en un sistema de cloud.
- En cuanto a couchdb es muy interesante cómo todas las consultas a la base de datos están implementadas mediante endpoints en una API, esto hace que sea muy sencillo realizar consultas mediante el navegador o una herramienta que posea couch en el endpoint llamado `_utils`
- Es interesante como se implementan los backups mediante el uso de los propios endpoints de couch, y no es necesario instalar o requerir otras herramientas fuera de las que brinda couch, el backup se realizó solamente usando las operaciones GET y POST para los endpoints necesarios.

Fuentes consultadas.

- <https://docs.couchdb.org/en/stable/install/unix.html>
- <https://github.com/apache/couchdb-helm/blob/main/couchdb/Chart.yaml>
- <https://www.ionos.com/digitalguide/hosting/technical-matters/work-with-couchdb-from-the-command-line/>
- <https://couchdb.apache.org/repo/couchdb.repo>
- <https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>