

Instituto Tecnológico de Costa Rica

IC4302 - Bases de Datos II

Documentación Proyecto 2

Profesor: Nereo Campos Araya

Estudiantes:

- Fiorella Zelaya Coto - 2021453615
- Isaac Araya Solano - 2018151703
- Melany Salas Fernández - 2021121147
- Moisés Solano Espinoza - 2021144322
- Pablo Arias Navarro - 2021024635

Instrucciones para ejecutar el proyecto

Para ejecutar el proyecto se siguen los siguientes steps:

1- Descomprimir el archivo .zip y abrir la línea de comandos wsl en la ubicación de la carpeta **Proyecto3**

| | | | |
|--|---|-------------------|--------------|
|  backups |  | 6/15/2023 6:53 PM | File folder |
|  bootstrap |  | 6/7/2023 8:12 PM | File folder |
|  databases |  | 6/15/2023 6:53 PM | File folder |
|  install |  | 6/7/2023 8:12 PM | Shell Script |
|  uninstall |  | 6/7/2023 8:12 PM | Shell Script |
|  updateDependencies |  | 6/7/2023 8:12 PM | Shell Script |

Posteriormente, puede ir a las carpetas **charts\databases** donde encontrará el archivo **values.yaml**, aquí podrá escoger la base de datos cambiando el valor de **enabled** a true

```

config:
  backup:
    enabled: false
  restore:
    enabled: true
mongo:
  enabled: false
  config:
    namespace: default
    connectionString: databases-mongodb.default.svc.cluster.local:27017
    storageAccount: filesmanagemangos
    container: documents
    path: /mongo
    maxBackups: 3
    azureSecret: azure-storage-account-key
    secret: databases-mongodb
    name: mongo
    schedule: "0 */12 * * *"
    diskSize: 2
    storageClass: hostpath
    provider: Azure
    image: moisode/mongodb-client
postgresql:

```

Posteriormente, debe ir a las carpetas **charts\backups** donde encontrará el archivo **values.yaml**, donde debe escoger la base de datos anteriormente escogida cambiando el valor de **enabled** a true

```

mongodb:
  enabled: false
  auth:
    rootPassword: "1234"

```

2- Ir a la carpeta **helm** con el comando **cd helm** o navegando hacia atrás on el comando **cd ..**

3- Ejecutar el comando **bash build.sh**, si da error al ejecutar, intente con el comando **dos2unix build.sh** y, posteriormente, ejecute de nuevo **bash build.sh**. Esperar a que se instalen los componentes.

```

fiozelaya@DESKTOP-MS2S40K: /mnt/c/Users/fio/OneDrive - Estudiantes ITCR/TEC/V Semestre/Bases de Datos II/Bases-De-Datos-II/Proyecto3/helm$ bash install.sh
Warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
persistentvolumeclaim "data-databases-mariadb-0" force deleted
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /home/fiozelaya/.kube/config
WARNING: Kubernetes configuration file is world-readable. This is insecure. Location: /home/fiozelaya/.kube/config
Release "bootstrap" does not exist. Installing it now.
NAME: bootstrap
LAST DEPLOYED: Thu Jun 15 19:05:29 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /home/fiozelaya/.kube/config
WARNING: Kubernetes configuration file is world-readable. This is insecure. Location: /home/fiozelaya/.kube/config
Release "databases" does not exist. Installing it now.
NAME: databases
LAST DEPLOYED: Thu Jun 15 19:05:37 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None

```

4- Ir a lens y observar el backup y restore de la base de datos.

| <input type="checkbox"/> | Name ▾ | Namespace ▾ | Containers ▾ | Restarts ▾ | Controlled ... ▾ | Node ▾ | QoS ▾ | Age ▾ | Status ▾ | ⋮ |
|--------------------------|----------------------------------|-------------|---|------------|------------------|----------------|------------|-------|----------|---|
| <input type="checkbox"/> | ic4302-es-default-0 | default | <div><div></div><div></div><div></div><div></div></div> | 0 | StatefulSet | docker-desktop | Burstable | 46s | Pending | ⋮ |
| <input type="checkbox"/> | ic4302-kb-7d7cc4f97-xh4tx | default | <div><div></div><div></div></div> | 0 | ReplicaSet | docker-desktop | Burstable | 44s | Running | ⋮ |
| <input type="checkbox"/> | coredns-565d847f94-hbpdb | kube-system | <div><div></div></div> | 1 | ReplicaSet | docker-desktop | Burstable | 9h | Running | ⋮ |
| <input type="checkbox"/> | coredns-565d847f94-zrx4w | kube-system | <div><div></div></div> | 1 | ReplicaSet | docker-desktop | Burstable | 9h | Running | ⋮ |
| <input type="checkbox"/> | databases-postgresql-0 | default | <div><div></div></div> | 0 | StatefulSet | docker-desktop | Burstable | 65s | Running | ⋮ |
| <input type="checkbox"/> | elastic-operator-0 | default | <div><div></div></div> | 0 | StatefulSet | docker-desktop | Burstable | 73s | Running | ⋮ |
| <input type="checkbox"/> | etcd-docker-desktop | kube-system | <div><div></div></div> | 43 | Node | docker-desktop | Burstable | 9h | Running | ⋮ |
| <input type="checkbox"/> | kube-apiserver-docker-desktop | kube-system | <div><div></div></div> | 45 | Node | docker-desktop | Burstable | 9h | Running | ⋮ |
| <input type="checkbox"/> | kube-controller-manager-docke... | kube-system | <div><div></div></div> | 44 | Node | docker-desktop | Burstable | 9h | Running | ⋮ |
| <input type="checkbox"/> | kube-proxy-f5t5n | kube-system | <div><div></div></div> | 1 | DaemonSet | docker-desktop | BestEffort | 9h | Running | ⋮ |
| <input type="checkbox"/> | kube-scheduler-docker-desktop | kube-system | <div><div></div></div> | 47 | Node | docker-desktop | Burstable | 9h | Running | ⋮ |
| <input type="checkbox"/> | mariadb-backup-0-mx5jx | default | <div><div></div></div> | 0 | Job | docker-desktop | BestEffort | 4s | Pending | ⋮ |
| <input type="checkbox"/> | mariadb-data-0-w4djb | default | <div><div></div></div> | 0 | Job | docker-desktop | BestEffort | 4s | Pending | ⋮ |
| <input type="checkbox"/> | mariadb-restore-0-xgcn9 | default | <div><div></div></div> | 0 | Job | docker-desktop | BestEffort | 4s | Pending | ⋮ |
| <input type="checkbox"/> | storage-provisioner | kube-system | <div><div></div></div> | 2 | | docker-desktop | BestEffort | 9h | Running | ⋮ |
| <input type="checkbox"/> | vpnkit-controller | kube-system | <div><div></div></div> | 43 | | docker-desktop | BestEffort | 9h | Running | ⋮ |

5- Ejecutar el comando **bash uninstall.sh** para desinstalar los componentes.

```
fiodelaya@DESKTOP-MS2S40K: /mnt/c/Users/fio/OneDrive - Estudiantes ITCR/TEC/V Semestre/Bases de Datos II/Bases-De-Datos-II/Proyecto3/helm$ bash uninstall.sh
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /home/fiodelaya/.kube/config
WARNING: Kubernetes configuration file is world-readable. This is insecure. Location: /home/fiodelaya/.kube/config
release "backups" uninstalled
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /home/fiodelaya/.kube/config
WARNING: Kubernetes configuration file is world-readable. This is insecure. Location: /home/fiodelaya/.kube/config
release "databases" uninstalled
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /home/fiodelaya/.kube/config
WARNING: Kubernetes configuration file is world-readable. This is insecure. Location: /home/fiodelaya/.kube/config
release "bootstrap" uninstalled
```

Pruebas

Pruebas de backups en Azure

Microsoft Azure

Search resources, services, and docs (G+)

moises.solano@estudio...
ESTUDIANTE ITCR (STUDIANTE)

Home >

documents

Container

Search

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots Create snapshot Give feedback

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Access policy

Properties

Metadata

Authentication method: Access key (Switch to Azure AD User Account)

Location: documents

Search blobs by prefix (case-sensitive)

Show deleted blobs

Add filter

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state | |
|-------------------------------------|----------|-------------|----------------|-----------|------|-------------|-----|
| <input type="checkbox"/> mariadb | | | | | | - | ... |
| <input type="checkbox"/> mongo | | | | | | - | ... |
| <input type="checkbox"/> postgresql | | | | | | - | ... |

MongoDB

Backup

☐ mongo-backup-0-v8xhb

default

0

Job

docker-desktop

BestEffort

5m31s

Succeeded

Pod mongo-backup-0-v8xhb

Namespace default Owner Job mongo-backup-0 Pod mongo-backup-0-v8xhb Container mongo-snapshotter

2023-06-16T01:30:26.014+0000 writing admin.system.users to archive '/mongodump/202306160129'

2023-06-16T01:30:26.025+0000 done dumping admin.system.users (1 document)

2023-06-16T01:30:26.026+0000 writing admin.system.version to archive '/mongodump/202306160129'

2023-06-16T01:30:26.043+0000 done dumping admin.system.version (2 documents)

2023-06-16T01:30:26.044+0000 writing bd2.names to archive '/mongodump/202306160129'

Microsoft Azure Search resources, services, and docs (G+)

Home > documents Container

Authentication method: Access key (Switch to Azure AD User Account)
Location: documents / mongo

Search blobs by prefix (case-sensitive) ☐ Show deleted blobs

Add filter

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|--|----------|-------------|----------------|-----------|------|-------------|
| <input type="checkbox"/> [-] | | | | | | ... |
| <input type="checkbox"/> 202306150036 | | | | | | ... |
| <input type="checkbox"/> 202306151924 | | | | | | ... |
| <input type="checkbox"/> <u>202306160129</u> | | | | | | ... |

Para hacer un backup de MongoDB:

1. Ir al archivo **databases/values.yaml**
2. Cambiar config.backup.enabled a **true**

```
config:
  backup:
    enabled: true
  restore:
    enabled: false
```

Restauración

```
bd2> show dbs
admin 116.00 KiB
bd2 40.00 KiB
config 12.00 KiB
local 72.00 KiB
bd2> show tables
names
bd2> db.names.find()
[
  { _id: ObjectId("648a5d1e36dd53aee221eac1"), name: 'Juan' },
  { _id: ObjectId("648a5d1e36dd53aee221eac2"), name: 'Luisiana' },
  { _id: ObjectId("648a5d1e36dd53aee221eac3"), name: 'Rodrigo' }
]
bd2>
```

```

"copy": {
  "completionTime": null,
  "destinationSnapshot": null,
  "id": null,
  "incrementalCopy": null,
  "progress": null,
  "source": null,
  "status": null,
  "statusDescription": null
},
"creationTime": null,
"deletedTime": null,
"etag": "\"0x8D86D38D5E58FD2\"",
"lastModified": "2023-06-15T00:38:27+00:00",
"lease": {
  "duration": null,
  "state": "available",
  "status": "unlocked"
},
"pageBlobSequenceNumber": null,
"pageRanges": null,
"rehydrationStatus": null,
"remainingRetentionDays": null,
"serverEncrypted": true
},
"rehydratePriority": null,
"requestServerEncrypted": true,
"snapshot": null,
>tagCount": null,
"tags": null,
"versionId": null
}
}
Restaurando el respaldo en MongoDB
2023-06-16T01:38:39.674+0000    preparing collections to restore from
2023-06-16T01:38:39.679+0000    reading metadata for bd2.names from archive 'mongorestore/202306150036/archive.gz'
2023-06-16T01:38:39.705+0000    restoring bd2.names from archive 'mongorestore/202306150036/archive.gz'
2023-06-16T01:38:39.716+0000    finished restoring bd2.names (3 documents, 0 failures)
2023-06-16T01:38:39.716+0000    restoring users from archive 'mongorestore/202306150036/archive.gz'
2023-06-16T01:38:39.739+0000    no indexes to restore for collection bd2.names
2023-06-16T01:38:39.739+0000    3 document(s) restored successfully. 0 document(s) failed to restore.

```

Para hacer una restauración de MongoDB:

1. Ir al archivo **databases/values.yaml**
2. Cambiar config.restore.enabled a **true**

```

config:
  backup:
    enabled: false
  restore:
    enabled: true

```

MariaDB

Backup

```

❑ mariadb-backup-0-kdn2x      ⚠ default      ❑      0      Job      docker-desktop      BestEffort      84s      Succeeded
❑ mariadb-data-0-twtgts      default      ❑      0      Job      docker-desktop      BestEffort      84s      Succeeded
❑ kube-proxy-5ks7l           kube-system  ■      25     DaemonSet  docker-desktop      BestEffort      97d      Running

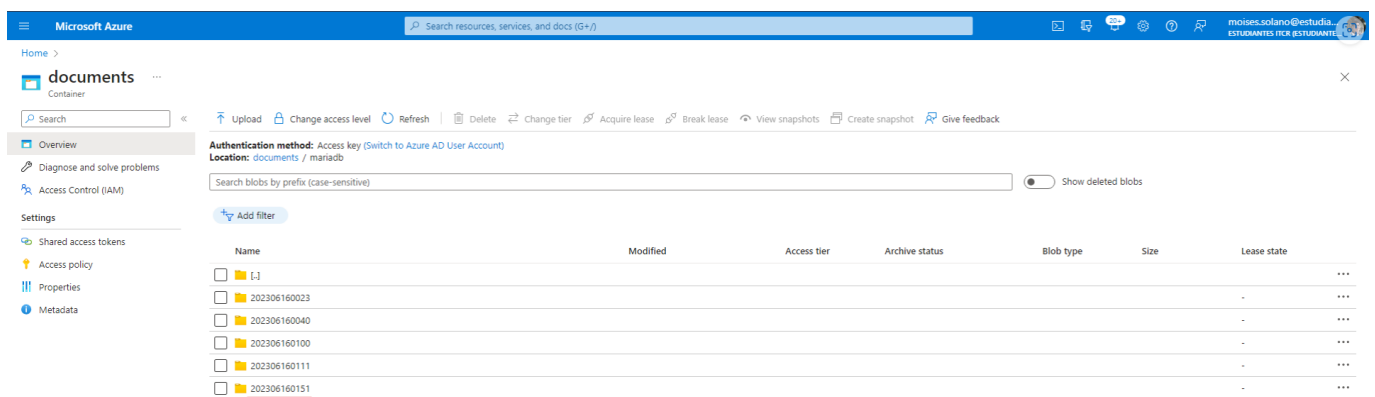
Pod mariadb-backup-0-kdn2x x +
Namespace default Owner Job mariadb-backup-0 Pod mariadb-backup-0-kdn2x Container mariadb-snapshotter Search...
WARNING: This command is implicitly deprecated because Command group 'Storage' and 'Utility' is deprecated and will be removed in a future release. Use 'az storage' instead.
WARNING: Azcopy command: ['/.root/.azure/cliextensions/storage-preview/azext_storage_preview/azcopy/azcopy_linux_amd64_10.5.0/azcopy', 'copy', '/mariadb_dump/202306160151', 'https://filesmanagemangos.blob.core.windows.net/']
INFO: Scanning...
INFO: Any empty folders will not be processed, because source and/or destination doesn't have full folder support
Job df62d066-56c5-fa47-7a0a-ab00c0255c1c has started
Log file is located at: /.root/.azcopy/df62d066-56c5-fa47-7a0a-ab00c0255c1c.log

%, 0 Done, 0 Failed, 1 Pending, 0 Skipped, 1 Total,
INFO: azcopy: A newer version 10.19.0 is available to download
100.0 %, 0 Done, 0 Failed, 1 Pending, 0 Skipped, 1 Total,

Job df62d066-56c5-fa47-7a0a-ab00c0255c1c summary
Elapsed Time (Minutes): 0.0333
Number of File Transfers: 1
Number of Folder Property Transfers: 0
Total Number of Transfers: 1
Number of Transfers Completed: 1
Number of Transfers Failed: 0
Number of Transfers Skipped: 0
TotalBytesTransferred: 0
Final Job Status: Completed

Respaldo completado exitosamente

```



Para hacer un backup de MariaDB:

1. Ir al archivo **databases/values.yaml**
2. Cambiar config.backup.enabled a **true**

```

config:
  backup:
    enabled: true
  restore:
    enabled: false

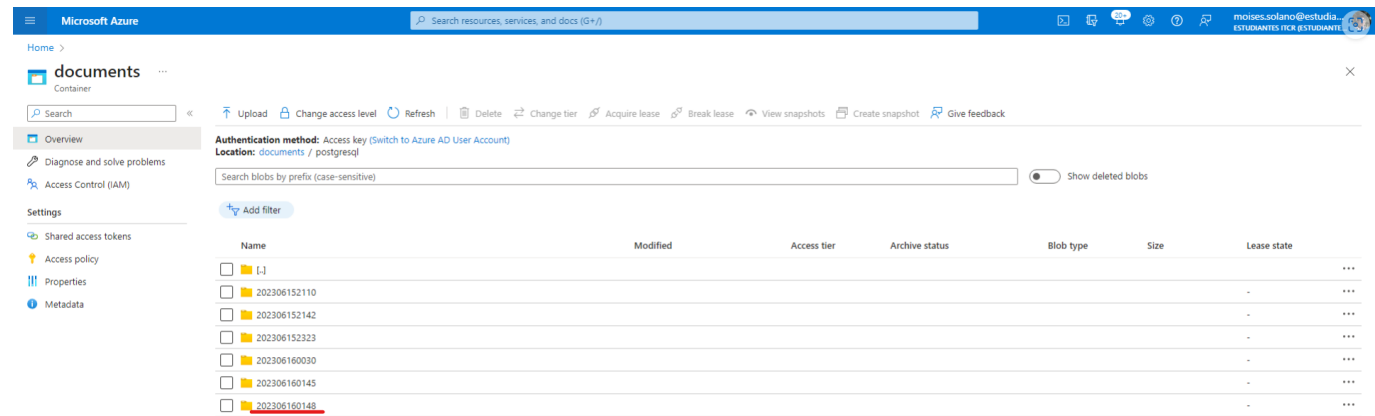
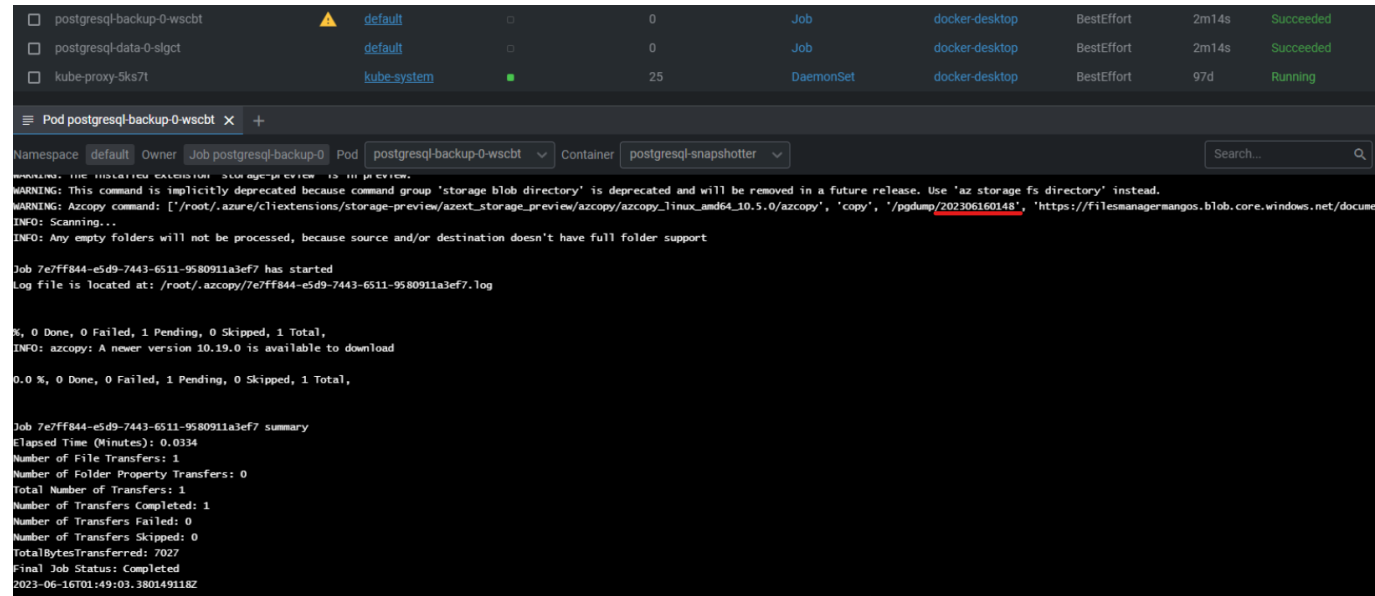
```

Restauración

No se implementa la restauración de bases de datos para MariaDB.

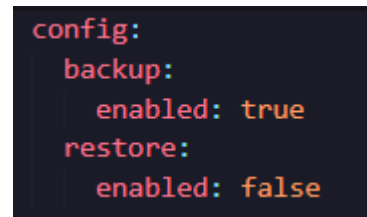
PostgreSQL

Backup



Para hacer un backup de PostgreSQL:

1. Ir al archivo **databases/values.yaml**
2. Cambiar config.backup.enabled a **true**



Restauración

```

SET
SET
CREATE PROCEDURE
ALTER PROCEDURE
CREATE PROCEDURE
ALTER PROCEDURE
CREATE PROCEDURE
ALTER PROCEDURE
CREATE PROCEDURE
ALTER PROCEDURE
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
ALTER TABLE
COPY 0
  setval
-----
      1
(1 row)

ALTER TABLE
You are now connected to database "postgres" as user "postgres".
SET
SET
SET
SET
SET
  set_config
-----

(1 row)

SET
SET
SET
SET

```

```

I have no name!@databases-postgresql-0:/$ psql -U postgres
Password for user postgres:
psql (15.2)
Type "help" for help.

```

```
postgres=# \l
```

| List of databases | | | | | | | |
|-------------------|----------|----------|-------------|-------------|------------|-----------------|--|
| Name | Owner | Encoding | Collate | Ctype | ICU Locale | Locale Provider | Access privileges |
| babynames | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | | libc | |
| postgres | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | | libc | |
| template0 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | | libc | =c/postgres + postgres=CTc/postgres |
| template1 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | | libc | =c/postgres + postgres=CTc/postgres |

```
(4 rows)
```

```
postgres=# |
```

Para hacer una restauración de PostgreSQL:

1. Ir al archivo **databases/values.yaml**
2. Cambiar config.restore.enabled a **true**

```
config:  
  backup:  
    enabled: false  
  restore:  
    enabled: true
```

ElasticSearch

No se implementa el respaldo ni la restauración de base de datos para ElasticSearch.

Neo4j

No se implementa el respaldo ni la restauración de base de datos para Neo4j.

CouchDB

No se implementa el respaldo ni la restauración de base de datos para CouchDB.

Componentes

Para cada una de las bases de datos, se agregan los valores necesarios en el archivo "values.yaml", de la carpeta templates, en la sección de backups.

```

mongo:
  enabled: false
  config:
    namespace: default
    connectionString: databases-mongodb.default.svc.cluster.local:27017
    storageAccount: filesmanagemangos
    container: documents
    path: /mongo
    maxBackups: 3
    azureSecret: azure-storage-account-key
    secret: databases-mongodb
    name: mongo
    schedule: "0 */12 * * *"
    diskSize: 2
    storageClass: hostpath
    provider: Azure
    image: moisode/mongodb-client
postgresql:
  enabled: true
  config:
    mapName: script-db
    namespace: default
    connectionString: databases-postgresql.default.svc.cluster.local
    storageAccount: filesmanagemangos
    container: documents
    path: /postgresql
    maxBackups: 3
    azureSecret: azure-storage-account-key
    secret: databases-postgresql
    name: postgresql
    schedule: "0 */12 * * *"

```

MongoDB

Para el backup de MongoDB se utilizó el archivo "backup.yaml" en el que se realizan:

- ConfigMap: Se guardan aspectos de configuración, como el namespace y el script correspondiente.

```

{{- if .Values.mongo.enabled }}
apiVersion: v1
kind: ConfigMap
metadata:
  name: mongo-backups-script
  namespace: {{ .Values.namespace }}
data:
  {{ (.Files.Glob "scripts/mongodb.sh").AsConfig | indent 2 }}

```

- PersistentVolumeClaim: Se solicita el almacenamiento necesario y se define el modo de lectura-escritura.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mongo-snapshotter
  namespace: {{ .Values.mongo.config.namespace }}
spec:
  storageClassName: {{ .Values.mongo.config.storageClass }}
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: {{ .Values.mongo.config.diskSize }}
```

- CronJob: Utilizado para automatizar los backups, tambien se definen variables de entorno y demás.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mongo-snapshotter
  namespace: {{ .Values.mongo.config.namespace }}
spec:
  storageClassName: {{ .Values.mongo.config.storageClass }}
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: {{ .Values.mongo.config.diskSize }}
```

- Job: Se crea el pod para realizar la tarea del respaldo de MongoDB, tambien se definen variables de entorno y demás. Para realizar el restore del respaldo también se utiliza un Job.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: mongo-backup-0
  namespace: {{ .Values.mongo.config.namespace }}
spec:
  template:
    spec:
      serviceAccountName: {{ .Values.mongo.config.serviceAccount }}
      volumes:
        - name: scripts
          configMap:
            name: mongo-backups-script
            defaultMode: 0777
        - name: mongo-snapshotter
          persistentVolumeClaim:
            claimName: mongo-snapshotter
      containers:
        - image: mcr.microsoft.com/azure-cli
          name: mongo-snapshotter
          env:
            - name: MONGO_CONNECTION_STRING
              value: "{{ .Values.mongo.config.connectionString }}"
            - name: AZURE_STORAGE_ACCOUNT
              value: "{{ .Values.mongo.config.storageAccount }}"
            - name: CONTAINER
```

Por otro lado, en este mismo archivo se definen un configmap y un job para hacer el cargado de la base de datos y los datos de prueba necesarios para comprobar el funcionamiento del backup.

```

apiVersion: batch/v1
kind: Job
metadata:
  name: mongo-backup-0
  namespace: {{ .Values.mongo.config.namespace }}
spec:
  template:
    spec:
      serviceAccountName: {{ .Values.mongo.config.serviceAccount }}
      volumes:
      - name: scripts
        configMap:
          name: mongo-backups-script
          defaultMode: 0777
      - name: mongo-snapshotter
        persistentVolumeClaim:
          claimName: mongo-snapshotter
      containers:
      - image: mcr.microsoft.com/azure-cli
        name: mongo-snapshotter
        env:
        - name: MONGO_CONNECTION_STRING
          value: "{{ .Values.mongo.config.connectionString }}"
        - name: AZURE_STORAGE_ACCOUNT
          value: "{{ .Values.mongo.config.storageAccount }}"

```

Backup

```

#!/bin/bash
# get the current date and time
DATE=$(date '+%Y%m%d%H%M')
# create a directory with the current date
# the [-p] arg is used bs /mongodump may not exist
mkdir -p /mongodump/$DATE
apk update
apk upgrade
# The MongoDB tools provide import, export, and diagnostic capabilities.
apk add mongodb-tools
# Azure CLI command for enable dynamic install without a prompt.
az config set extension.use_dynamic_install=yes_without_prompt
# mongodump configuration for connect to an instance
# --host, -u or --username, -p or --password, --gzip(compress the output), --archive(Writes the output to a specified archive file)
mongodump --host="$MONGO_CONNECTION_STRING" -u $MONGO_USERNAME -p $MONGO_PASSWORD --gzip --archive=/mongodump/$DATE
az storage blob directory upload --container $CONTAINER -s /mongodump/$DATE -d $BACKUP_PATH --auth-mode key --recursive
rm -rf /mongodump/$DATE

```

Para el script del backup de MongoDB se obtiene la fecha, y se crea el directorio (en caso de que no exista), además, se hace un update de los paquetes y se importa **mongodb-tools**, este brinda las herramientas necesarias para manejar Mongo y poder hacer el dump.

Posteriormente se hace el **mongodump** a Mongo mediante el conection string, el username y el password.

Finalmente, se sube el dump al blobstorage de Azure.

Restauración

```
#!/bin/bash

# Config Variables
BACKUP_NAME="202306150036" # Date of backup (folder) in Azure Blob Storage
ARCHIVE_NAME="archive.gz" # Name of file in Azure Blob Storage
CONNECTION_STRING_AZURE="DefaultEndpointsProtocol=https;AccountName=filesmanagemangos;AccountKey=71ms2t3YfN7Qu4K11gC1PR5adRZVuhqKgn7mDXuQI0ZgF7ougQUR0LWhf71cECM9YdV9c2grT+As8ZXu+g==;EndpointSuffix=core.windows.net"

mkdir -p mongorestore/$BACKUP_NAME
apk update
apk upgrade
# The MongoDB tools provide import, export, and diagnostic capabilities.
apk add mongodb-tools

# Azure CLI command for enable dynamic install without a prompt.
az config set extension.use_dynamic_install=yes_without_prompt

# Download backup from Azure Blob Storage
az storage blob download --container $CONTAINER --name mongo/$BACKUP_NAME/$ARCHIVE_NAME --file mongorestore/$BACKUP_NAME/$ARCHIVE_NAME --auth-mode key --connection-string $CONNECTION_STRING_AZURE

# Restore backup to MongoDB
mongorestore --host="$MONGO_CONNECTION_STRING" -u $MONGO_USERNAME -p $MONGO_PASSWORD --gzip --archive=mongorestore/$BACKUP_NAME/$ARCHIVE_NAME
```

Para el script de restauración también se crea un directorio, se hace un update de los paquetes y se importa **mongodb-tools**.

Posteriormente, se descarga el archivo desde el blob de azure, para que luego se pueda restaurar la base de datos con el comando correspondiente. En este caso se usa **mongodbrestore**.

MariaDB

Para el backup de MariaDB se utilizó el archivo "backup_mariadb.yaml", en el que se realizan:

- ConfigMap: Se guardan aspectos de configuración, como el namespace y el script correspondiente.

```
{{ if .Values.mariadb.enabled }}
apiVersion: v1
kind: ConfigMap
metadata:
  name: mariadb-backups-script
  namespace: {{ .Values.namespace }}
data:
  {{ (.Files.Glob "scripts/mariadb.sh").AsConfig | indent 2 }}
```

- PersistentVolumeClaim: Se solicita el almacenamiento necesario y se define el modo de lectura-escritura.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mariadb-snapshotter
  namespace: {{ .Values.mariadb.config.namespace }}
spec:
  storageClassName: {{ .Values.mariadb.config.storageClass }}
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: {{ .Values.mariadb.config.diskSize }}
```

- CronJob: Utilizado para automatizar los backups, también se definen variables de entorno y demás.

```
---
apiVersion: batch/v1
kind: CronJob
metadata:
  name: "mariadb-backup"
  namespace: {{ .Values.mariadb.config.namespace }}
spec:
  schedule: {{ .Values.mariadb.config.schedule }}
  successfulJobsHistoryLimit: 1
  failedJobsHistoryLimit: 1
```

- Job: Se crea el pod para realizar la tarea del respaldo de MariaDB, tambien se definen variables de entorno y demás. Para la parte de la restauración, tambien se crea un job que se encarga de realizar la tarea de restaurar la base de datos.

```

apiVersion: batch/v1
kind: Job
metadata:
  name: mariadb-backup-0
  namespace: {{ .Values.mariadb.config.namespace }}
spec:
  template:
    spec:
      serviceAccountName: {{ .Values.mariadb.config.serviceAccount }}
      volumes:
        - name: scripts
          configMap:
            name: mariadb-backups-script
            defaultMode: 0777
        - name: mariadb-snapshotter
          persistentVolumeClaim:
            claimName: mariadb-snapshotter
      containers:
        - image: mcr.microsoft.com/azure-cli
          name: mariadb-snapshotter
          env:
            - name: MARIADB_CONNECTION_STRING
              value: "{{ .Values.mariadb.config.connectionString }}"
            - name: AZURE_STORAGE_ACCOUNT
              value: "{{ .Values.mariadb.config.storageAccount }}"
            - name: CONTAINER
              value: "{{ .Values.mariadb.config.container }}"
            - name: BACKUP_PATH
              value: "{{ .Values.mariadb.config.path }}"
            - name: MAX_BACKUPS
              value: "{{ .Values.mariadb.config.maxBackups }}"
            - name: DB_HOST
              value: "databases-mariadb"
            - name: MARIADB_USERNAME
              value: "root"
            - name: MARIADB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: "{{ .Values.mariadb.config.secret }}"
                  key: mariadb-root-password
                  optional: false
            - name: AZURE_STORAGE_KEY

```

Por otro lado, en este mismo archivo se definen un configmap y un job para hacer el cargado de la base de datos y los datos de prueba necesarios para comprobar el funcionamiento del backup.

 MariaDB Backup

Backup


```

apiVersion: batch/v1
kind: Job
metadata:
  name: mariadb-data-0
spec:
  template:
    spec:
      volumes:
      - name: scripts
        configMap:
          name: mariadb-data
          defaultMode: 0777
      containers:
      - name: pi
        image: "{{ .Values.mariadb.config.image }}"
        volumeMounts:
        - name: scripts
          mountPath: /scripts
        env:
          - name: MARIADB_CONNECTION_STRING
            value: "{{ .Values.mariadb.config.connectionString }}"
          - name: AZURE_STORAGE_ACCOUNT
            value: "{{ .Values.mariadb.config.storageAccount }}"
          - name: CONTAINER
            value: "{{ .Values.mariadb.config.container }}"
          - name: BACKUP_PATH
            value: "{{ .Values.mariadb.config.path }}"
          - name: MAX_BACKUPS
            value: "{{ .Values.mariadb.config.maxBackups }}"
          - name: MARIADB_USERNAME
            value: "root"
          - name: DB_HOST
            value: "databases-mariadb"
          - name: MARIADB_PASSWORD
            valueFrom:
              secretKeyRef:
                name: "{{ .Values.mariadb.config.secret }}"
                key: mariadb-root-password
                optional: false
          - name: AZURE_STORAGE_KEY

```

Para el script del backup de MariaDB se obtiene la fecha, y se crea el directorio (en caso de que no exista), además, se hace un update de los paquetes y se importa **mariadb-client**, este brinda las herramientas necesarias para manejar MariaDB y poder hacer el dump.

Posteriormente se hace el **mysqldump** a la base de datos en MariaDB mediante el username, el password y el host.

Finalmente, se sube el dump al blobstorage de Azure.

Restauración

```

apiVersion: batch/v1
kind: Job
metadata:
  name: mariadb-restore-0
  namespace: {{ .Values.mariadb.config.namespace }}
spec:
  template:
    spec:
      serviceAccountName: {{ .Values.mariadb.config.serviceAccount }}
      volumes:
      - name: scripts
        configMap:
          name: mariadb-restore-script
          defaultMode: 0777
      containers:
      - image: mcr.microsoft.com/azure-cli
        name: mariadb-restorer
        env:
          - name: MARIADB_CONNECTION_STRING
            value: "{{ .Values.mariadb.config.connectionString }}"
          - name: AZURE_STORAGE_ACCOUNT
            value: "{{ .Values.mariadb.config.storageAccount }}"
          - name: CONTAINER
            value: "{{ .Values.mariadb.config.container }}"
          - name: BACKUP_PATH
            value: "{{ .Values.mariadb.config.path }}"
          - name: MARIADB_USERNAME
            value: "root"
          - name: MARIADB_PASSWORD
            valueFrom:
              secretKeyRef:
                name: "{{ .Values.mariadb.config.secret }}"
                key: mariadb-root-password
                optional: false
          - name: DB_NAME
            value: "babynames"
          - name: AZURE_STORAGE_KEY
            valueFrom:
              secretKeyRef:
                name: "{{ .Values.mariadb.config.azureSecret }}"
                key: azure.client.default.key
                optional: false

```

Para el script de restauración también se crea un directorio, se hace un update de los paquetes y se importa **mariadb-client** y **mysql-server**.

Posteriormente, se descarga el archivo desde el blob de azure, para que luego se pueda restaurar la base de datos con el comando correspondiente. En este caso el comando que se usa es **mariadbrestore**

PostgreSQL

Para el backup de PostgreSQL se utilizó el archivo "backup_postgresql.yaml", en el que se realizan:

- ConfigMap: Se guardan aspectos de configuración, como el namespace y el script correspondiente.

```
{{ if .Values.postgresql.enabled }}
apiVersion: v1
kind: ConfigMap
metadata:
  name: postgresql-backups-script
  namespace: {{ .Values.namespace }}
data:
  {{ (.Files.Glob "scripts/postgresql.sh").AsConfig | indent 2 }}
```

- PersistentVolumeClaim: Se solicita el almacenamiento necesario y se define el modo de lectura-escritura.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: postgresql-snapshotter
  namespace: {{ .Values.postgresql.config.namespace }}
spec:
  storageClassName: {{ .Values.postgresql.config.storageClass }}
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: {{ .Values.postgresql.config.diskSize }}
```

- CronJob: Utilizado para automatizar los backups, también se definen variables de entorno y demás.

```

apiVersion: batch/v1
kind: CronJob
metadata:
  name: "postgresql-backup"
  namespace: {{ .Values.postgresql.config.namespace }}
spec:
  schedule: {{ .Values.postgresql.config.schedule }}
  successfulJobsHistoryLimit: 1
  failedJobsHistoryLimit: 1
  jobTemplate:
    spec:
      backoffLimit: 3
      template:
        spec:
          terminationGracePeriodSeconds: 0
          restartPolicy: Never
          volumes:
            - name: scripts
              configMap:
                name: postgresql-backups-script
                defaultMode: 0777
            - name: postgresql-snapshotter
              persistentVolumeClaim:
                claimName: postgresql-snapshotter
          containers:
            - image: mcr.microsoft.com/azure-cli
              name: postgresql-snapshotter
              env:
                - name: POSTGRESQL_CONNECTION_STRING
                  value: "{{ .Values.postgresql.config.connectionString }}"
                - name: AZURE_STORAGE_ACCOUNT
                  value: "{{ .Values.postgresql.config.storageAccount }}"

```

- Job: Se crea el pod para realizar la tarea del respaldo de PostgreSQL, tambien se definen variables de entorno y demás. Para la parte de la restauración, tambien se crea un job que se encarga de realizar la tarea de restaurar la base de datos.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: postgresql-backup-0
  namespace: {{ .Values.postgresql.config.namespace }}
spec:
  template:
    spec:
      serviceAccountName: {{ .Values.postgresql.config.serviceAccount }}
      volumes:
      - name: scripts
        configMap:
          name: postgresql-backups-script
          defaultMode: 0777
      - name: postgresql-snapshotter
        persistentVolumeClaim:
          claimName: postgresql-snapshotter
      containers:
      - image: mcr.microsoft.com/azure-cli
        name: postgresql-snapshotter
        env:
          - name: POSTGRESQL_CONNECTION_STRING
            value: "{{ .Values.postgresql.config.connectionString }}"
          - name: AZURE_STORAGE_ACCOUNT
            value: "{{ .Values.postgresql.config.storageAccount }}"
          - name: CONTAINER
            value: "{{ .Values.postgresql.config.container }}"
          - name: BACKUP_PATH
            value: "{{ .Values.postgresql.config.path }}"
          - name: MAX_BACKUPS
            value: "{{ .Values.postgresql.config.maxBackups }}"
          - name: DB_HOST
            value: "databases-postgresql"
          - name: POSTGRESQL_USERNAME
```

Por otro lado, en este mismo archivo se definen un configmap y un job para hacer el cargado de la base de datos y los datos de prueba necesarios para comprobar el funcionamiento del backup.

```

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: postgresql-data
  namespace: {{ .Values.namespace }}
data:
{{ (.Files.Glob "sql/postgres.sql").AsConfig | indent 2 }}
---
apiVersion: batch/v1
kind: Job
metadata:
  name: postgresql-data-0
spec:
  template:
    spec:
      volumes:
      - name: scripts
        configMap:
          name: postgresql-data
          defaultMode: 0777
      containers:
      - name: pi
        image: {{ .Values.postgresql.config.image }}
        volumeMounts:
        - name: scripts
          mountPath: /scripts
        env:
        - name: POSTGRESQL_CONNECTION_STRING

```

Backup

```

#!/bin/bash
# get the current date and time
DATE=$(date '+%Y%m%d%H%M')
# create a directory with the current date
# the [-p] arg is used bs /pgdump may not exist
mkdir -p /pgdump/$DATE
# Update the packages
apk update
apk upgrade
# Install the PostgreSQL client
apk add postgresql-client
# Azure CLI command for enable dynamic install without a prompt.
az config set extension.use_dynamic_install=yes_without_prompt
# pg_dump configuration for connect to an instance
PGPASSWORD="$POSTGRESQL_PASSWORD" pg_dumpall --host $DB_HOST -U $POSTGRESQL_USERNAME --file=/pgdump/$DATE/db_backup.dump
# Upload the backup to the blob storage
az storage blob directory upload --container $CONTAINER -s /pgdump/$DATE -d $BACKUP_PATH --auth-mode key --recursive
rm -rf /pgdump/$DATE

```

Para el script del backup de PostGreSQL se obtiene la fecha, y se crea el directorio (en caso de que no exista), además, se hace un update de los paquetes y se importa **postgresql-client**, este brinda las herramientas necesarias para manejar postgresql y poder hacer el dump.

Posteriormente se hace el **pg_dump** a la base de datos en postgresql mediante el username, el password y el host.

Finalmente, se sube el dump al blobstorage de Azure.

Restauración

```
#!/bin/bash
# Configuration variables
BACKUP_NAME="202306152142"      # folder name in the blob storage
ARCHIVE_NAME="db_backup.dump"   # file name in the blob storage
# Connection string for Azure Blob Storage
CONNECTION_STRING_AZURE="DefaultEndpointsProtocol=https;AccountName=filesmanagemangos;AccountKey=71ms2t3YFnW7Qu4K1lgC1PR5
# Creates the directory
mkdir -p postgresqlrestore/$BACKUP_NAME
# Update and upgrade the packages
apk update
apk upgrade
apk add postgresql-client
# Azure CLI command for enable dynamic install without a prompt.
az config set extension.use_dynamic_install=yes_without_prompt
# Download the backup from the blob storage
az storage blob download --container $CONTAINER --name postgresql/$BACKUP_NAME/$ARCHIVE_NAME --file postgresqlrestore/$BAC
# Restore a PostgreSQL database from a backup using psql
PGPASSWORD="$POSTGRESQL_PASSWORD" psql --set ON_ERROR_STOP=off -h $DB_HOST -U $POSTGRESQL_USERNAME -f postgresqlrestore/$B
```

Para el script de restauración también se crea un directorio, se hace un update de los paquetes y se importa **postgresql-client**.

Posteriormente, se descarga el archivo desde el blob de azure, para que luego se pueda restaurar la base de datos con el comando correspondiente.

ElasticSearch

No se implementa el respaldo ni la restauración de base de datos para ElasticSearch.

Neo4j

No se implementa el respaldo ni la restauración de base de datos para Neo4j.

CouchDB

No se implementa el respaldo ni la restauración de base de datos para CouchDB.

Conclusiones

- 1- Es fundamental la comunicación para un buen desarrollo del proyecto.
- 2- Se debe mantener una buena organización para poder realizar el trabajo.
- 3- Es de gran importancia entender los conceptos básicos para realizar el proyecto.
- 4- El tener un buen control de versiones y la correcta utilización de github facilita el trabajo en equipo.
- 5- Se deben aplicar buenas prácticas de programación para mantener el orden.
- 6- Mantener la estructura definida del proyecto es esencial para evitar el desorden.
- 7- Se debe desarrollar un código legible y entendible.

- 8-** Se debe organizar el equipo de trabajo desde el día 1.
- 9-** Se debe tener una estructura clara y ordenada del proyecto y lo que requiere.
- 10-** Es importante que cada miembro del equipo entienda la tarea a realizar.

Recomendaciones

- 1-** Hacer reuniones periódicas para discutir los avances del proyecto y mejorar la comunicación.
- 2-** Mantener la organización de la tarea.
- 3-** Dividir el trabajo es importante, pero también es importante que cada persona del equipo entienda su tarea.
- 4-** Hacer uso de github para el control de versiones y trabajo en conjunto.
- 5-** Seguir un estándar de código.
- 6-** Seguir aprendiendo y enriqueciendo el conocimiento después de finalizar el trabajo.
- 7-** Investigar sobre las herramientas que se usan en el proyecto.
- 8-** Tener una buena estructura del proyecto y dividir el proyecto de forma funcional para avanzar progresivamente.
- 9-** Mantener la comunicación durante el desarrollo del proyecto y nunca quedarse con dudas.
- 10-** Definir roles en el equipo de trabajo para mantener el orden y procurar buena dinámica de trabajo.

Referencias Bibliograficas
