Configure Replication and Enable Point-in-Time-Recovery for Cloud SQL for PostgreSQL

Task 1. Enable backups on the Cloud SQL for PostgreSQL instance

In this task you will enable scheduled backups on a Cloud SQL for PostgreSQL instance.

1. In Cloud Shell, display the instance details:

```
export CLOUD_SQL_INSTANCE=postgres-orders
gcloud sql instances describe $CLOUD_SQL_INSTANCE
```

Copied!

content_copy

Click the **Authorize** button if prompted.

2. In Cloud Shell, get the current UTC time in 24 hour format:

   ```
   date +"%R"
   ```

   Copied!

   content_copy

3. In Cloud Shell, enter the following command to enable scheduled back-ups, replacing `HH:MM` with a time that is earlier than the time that was displayed in the previous step.

```
gcloud sql instances patch $CLOUD_SQL_INSTANCE \
    --backup-start-time=HH:MM
```

Copied!

content_copy

Note: For the purposes of this lab, it is imperative that you specify a backup start time earlier than the time displayed in the previous step. This is because you do not want a back-up to start while you are running the lab.

For example if the date command shows that the current time is `14:25` you could replace `HH:MM` with `13:25`, or even `12:00`. You must make sure it is a valid time in 24 hour format or you will receive an error saying the request was invalid.

4. Confirm your changes. Note the **format** parameter, which extracts only the desired fields.

```
gcloud sql instances describe $CLOUD_SQL_INSTANCE --format
'value(settings.backupConfiguration)'
```

Copied!

content_copy

You will see a response similar to the following showing that backups are set for 7 days, and run at 14:00 daily in this example:

```
backupRetentionSettings={'retainedBackups': 7, 'retentionUnit':
'COUNT'}; enabled=True;kind=sql#backupConfiguration; startTime=14:00;
transactionLogRetentionDays=7
```

Task 2. Enable and run point-in-time recovery

In this task you will enable and configure point-in-time recovery on a Cloud SQL for PostgreSQL instance. A point-in-time recovery always creates a new instance; you cannot perform a point-in-time recovery to an existing instance. The new instance inherits the settings of the source instance.

Enable point-in-time recovery

In this step you will enable point-in-time recovery.

- In Cloud Shell, enable point-in-time recovery:

```
gcloud sql instances patch $CLOUD_SQL_INSTANCE \
   --enable-point-in-time-recovery \
   --retained-transaction-log-days=1
```

Copied!

content_copy

It will take a minute or two for this command to complete.

Make a change to the Cloud SQL for PostgreSQL database

In this step you will add a row to the `orders.distribution_centers` table in the database. After point-in-time recovery we will expect this row to be absent from the database.

1. In Cloud Console, on the **Navigation menu** (≡), click **Databases** > **SQL** and click on the Cloud SQL instance named `postgres-orders`.

2. In Cloud Console, in the `Connect to this instance` section, click **Open Cloud Shell**. A command will be auto-populated to the Cloud Shell.

3. Run that command and enter the password `supersecret!` when prompted. A **psql** session will start in Cloud Shell.

4. In **psql**, change to the `orders` database:

```
\c orders
```

Copied!

content_copy

5. When prompted, enter the password `supersecret!` again.

6. In **psql**, get the row count of the `distribution_centers` table:

```
SELECT COUNT(*) FROM distribution_centers;
```

Copied!

content_copy

Output:

```
orders=> SELECT COUNT(*) FROM distribution_centers;
 count
-------
    10
(1 row)
```

7. In Cloud Shell, open a new tab (+), get the current UTC time in RFC 3339 format. This is the timestamp you will use for the point-in-time replica that you will create in the next task.

```
date --rfc-3339=seconds
```

Copied!

content_copy

You should wait for a few moments at this point to make sure that the changes you make in the next step occur after this timestamp.

Note: For the purposes of this lab, it is imperative that you specify a timestamp after point-in-time recovery was enabled (if not a successful back-up will be required as a starting point), but before the source instance was modified. If not your changes at the source will be replicated to the clone and the roll back won't be evident.

8. In **psql**, to add a row to the `orders.distribution_centers` table and get the new COUNT, run:

```
INSERT INTO distribution_centers VALUES(-80.1918,25.7617,'Miami
FL',11);
SELECT COUNT(*) FROM distribution_centers;
```

Copied!

content_copy

Output:

```
orders=> SELECT COUNT(*) FROM distribution_centers;
 count
-------
    11
(1 row)
```

9. Exit **psql**:

```
\q
```

Copied!

content_copy

Perform a point-in-time recovery

In this step you will make a clone of the `postgres-orders` Cloud SQL instance at a specific point in time.

- In Cloud Shell, to create a point-in-time clone, run:

```
export NEW_INSTANCE_NAME=postgres-orders-pitr
gcloud sql instances clone $CLOUD_SQL_INSTANCE $NEW_INSTANCE_NAME \
    --point-in-time 'TIMESTAMP'
```

Copied!

content_copy

You must replace the TIMESTAMP placeholder with the exact timestamp displayed by the `date` command you used earlier in the second Cloud Shell tab.

This TIMESTAMP must be UTC timezone, RFC 3339 format, for example, '2021-11-01 15:00:00'. The TIMESTAMP indicates the time to which you want to recover the state of the database. It should be enclosed in single quotes. The alternate RFC3339 variant is also supported: '2021-11-01T15:00:00.000Z'.

It could take 10 minutes or more for the replica to be created and ready for use. In the mean time, continue with the next task.

Task 3. Confirm database has been restored to the correct point-in-time

In this task you will confirm that a row of data that was added to the original database after the point-in-time recovery timestamp is not in the cloned database.

1. In Cloud Console, on the **Overview** page, click the **All Instances** breadcrumb and then click on the Cloud SQL instance named `postgres-orders-pitr`.

Now you will have to wait for the replica to come online.

2. In Cloud Console, in the `Connect to this instance` section, click **Open Cloud Shell**. A command will be auto-populated to the Cloud Shell.

3. Run that command and enter the password `supersecret!` when prompted. A **psql** session will start in Cloud Shell.

4. In **psql**, change to the `orders` database:

```
\c orders
```

Copied!

content_copy

5. When prompted, enter the password `supersecret!` again.

6. In **psql**, get the row count of the `distribution_centers` table:

```
SELECT COUNT(*) FROM distribution_centers;
```

Copied!

content_copy

Output:

```
orders=> SELECT COUNT(*) FROM distribution_centers;
 count
-------
    10
(1 row)
```

You will see that the `distribution_centers` table in the new Cloud SQL for PostgreSQL instance has the 10 rows that it had on the source instance at the point-in-time of cloning. If your query returns 11 rows check that you have connected to the replica instance and not the original.