Redis streams in production

Running the producer

This exercise assumes that you have cloned the course GitHub repository and followed the setup instructions. Note that you will need to activate your Python virtual environment and set any required environment variables for each new terminal session that you use here.

In a terminal window, start the producer:

```
cd src/week4

python partitioned_stream_producer.py
```

You should see the output similar to the following, which may take a few minutes to generate:

```
$ python partitioned_stream_producer.py

Deleting old streams:

temps:20250101

temps:20250102

temps:20250103

temps:20250104

temps:20250105

temps:20250106

temps:20250107

temps:20250108

temps:20250109
```

```
temps:20250110

Populating stream partition temps:20250101.

Populating stream partition temps:20250102.

Populating stream partition temps:20250103.

Populating stream partition temps:20250104.

Populating stream partition temps:20250105.

Populating stream partition temps:20250106.

Populating stream partition temps:20250107.

Populating stream partition temps:20250108.

Populating stream partition temps:20250109.

Populating stream partition temps:20250110.
```

What's happening here? The producer code simulates the generation of temperature values at a rate of one per second. Rather than producing data in real time, the producer operates by assuming that the date is January 1st 2025 at midnight UTC. It then runs as fast as it can to produce ten days' worth of temperature data at one-second intervals. Each day's data is written to a new stream named **temps:20250101, temps:20250102... temps:20250110.**

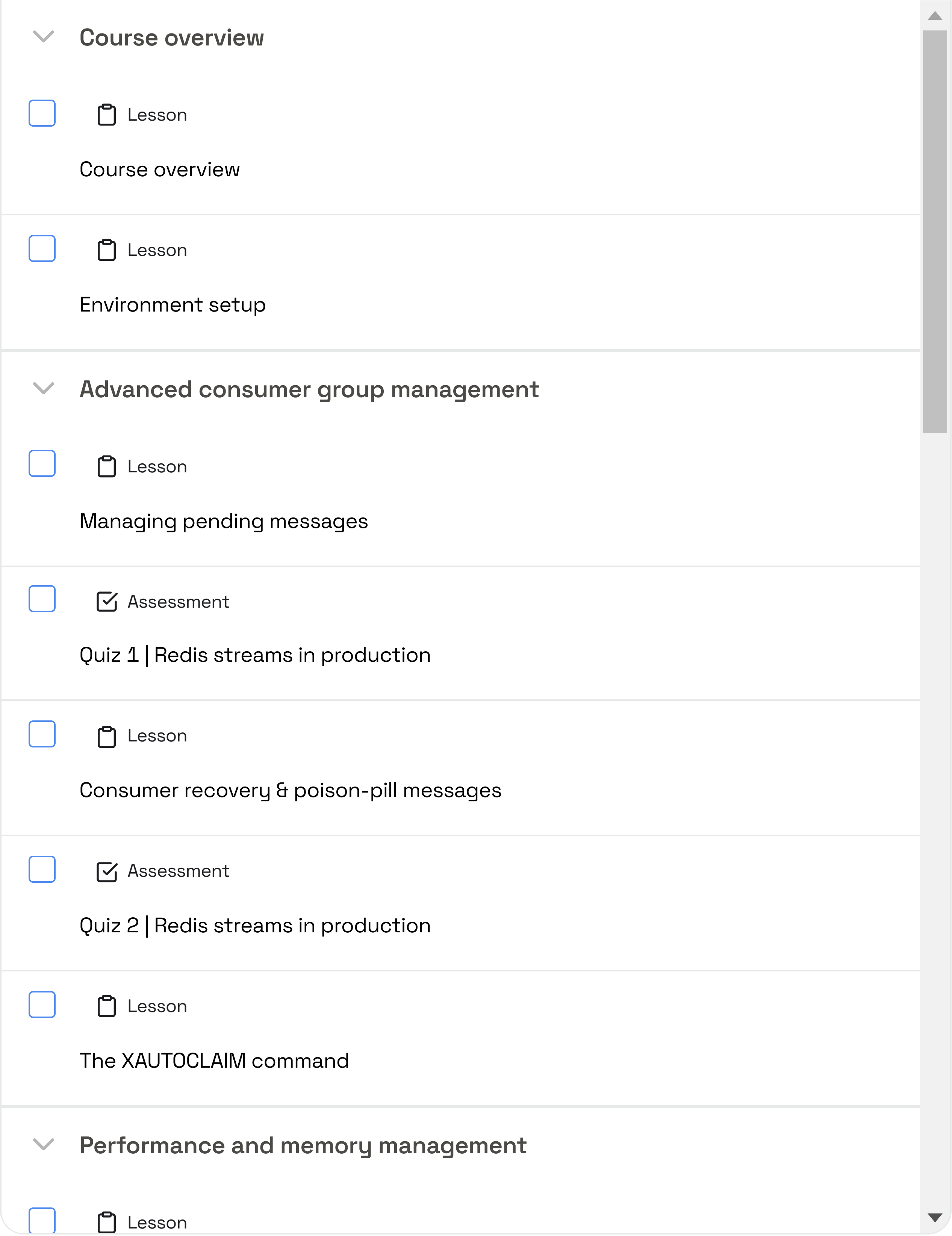
As the producer writes each entry to the stream, it sets the expiry date for the stream's key to be 2 days later than the timestamp of the entry written. This ensures that the overall memory required to store the dataset does not grow out of control. Also, if the producer unexpectedly crashes, it will never leave a partly finished stream partition without an expiry time. Eventually, older stream partitions will expire from Redis, making room for the continuous daily generation of new ones.

Note: you can re-run the producer at any time, and it will destroy and re-create the set of partitioned streams.

← Previous

Next →

Modules »



Course overview

☐ Lesson

Course overview

☐ Lesson

Environment setup

Advanced consumer group management

☐ Lesson

Managing pending messages

☐ Assessment

Quiz 1 | Redis streams in production

☐ Lesson

Consumer recovery & poison-pill messages

☐ Assessment

Quiz 2 | Redis streams in production

☐ Lesson

The XAUTOCLAIM command

Performance and memory management

☐ Lesson

Redis

Cloud

Software

Pricing

Support

University Feedback

University Help

Contact us

Legal notices

Trust

Terms of use

Privacy policy