Practical: ElastiCache and memcached with JS (optional)

Table of Contents

ElastiCache is AWS's managed service for memcached and Redis. In this practical we will see how to set up a memcached managed instance and use it to implement a basic cache for an external resource.

This practical is optional

Prerequisites and references

- **ElastiCache docs (https://docs.aws.amazon.com/elasticache/)**
- **Memcached homepage (https://memcached.org/)**
- **memcached npm module (https://www.npmjs.com/package/memcached)**
- **memcachedDemo.zip (https://canvas.qut.edu.au/courses/20367/files/6583686/download)**

Creating an elasticache instance

- Find ElastiCache on the AWS console
- Go to the *Memcached caches* list (on left side panel)
- *Create Memcached cache*
- Choose *Design your own cache* and *Easy create*
- Under Configuration choose *Demo*
- Under *Connectivity Subnet group* choose *CAB432-subnets*
- Add tags for your qut-username as usual
- Create the cache

It will take a few minutes to create the cache.

- In the sidebar, click *Memcached caches*
- In the list of memcached instances click on your instance. You may need to wait a few minutes before the cache is available for you to modify.
- Click on the *Network and security* tab:
    - Under *Security groups* click *Modify* on the right
    - Under *Security* click *Manage* on the right
    - Tick the box beside *CAB432MemcachedSG*

  - Click *Choose*
  - Click *Preview changes*
  - Click *Modify*
- You will be back in the list of memcached caches. Click on your cache again
- Copy the *Configuration endpoint* value. You will need this for connecting. You might need to wait a minute or two for this to be populated. Go back to the list and click on your instance again to refresh

Please note:

- The *CAB432MemcachedSG* security group is configured to allow access to your memcached cache from EC2 instances in the *CAB432SG* security group. If the cache and EC2 instance don't have their security groups configured properly then you will not be able to connect to the cache from your EC2 instance
- Memcached caches are not accessible from outside the AWS environment. You can only access them from EC2 instances (or later, containers and Lambdas)

Caching external resources

We'll demonstrate how to use memcached to cache an external resource, in this case a Wikipedia page.

- Download **memcachedDemo.zip (https://canvas.qut.edu.au/courses/20367/files/6583686/download)** , unzip it and copy the files to an EC2 instance using `scp` or similar as in previous practicals.
- Edit `index.js` from the demo code and replace the value for `memcachedAddress` with the endpoint for your ElastiCache instance, including the port number.
- In the same directory `index.js` run `npm install` to install the dependencies
- Run `node index.js`

At this point you should see the demo run. It will first use `fetch` to grab the contents from the specified URL. Then it will use the `cachedFetch` function to do the same thing, 10 times over. You should see a dramatically lower time required for the cached fetch, but only after the first call in which the data is fetched from the URL.

Have a look at the code to familiarise yourself with how it works.

- Note that we have monkey-patched two functions onto the `memcached` object. This is just for convenience, so that we can use `await` rather than supplying a callback function. If you want to use other functions of memcached with `await` then you will have to do something similar.
- `cachedFetch` implements the basic caching mechanism.
- Note that we have specified a time-to-live of 10 seconds: `await memcached.aSet(URL, fetchedValue, 10);` This can be tuned to optimise performance and how fast data changes. For Wikipedia 10 this could probably be increased substantially, as pages don't change so often.

Delete your ElastiCache instance

If you are done with your ElastiCache instance, please delete it. You are also welcome to keep it around if you will be using it with your assessment, in which case please update the `purpose` tag to indicate this.