**Plan for implementation**

**Server instances**

It will be first prudent to create a basic console app project that will be the client server simulations that we will have, which will be a simple API that we will spin up 3 to 5 instances of simply by starting a dotnet command and this will have a few things:

/health endpoint will check if the server responds with the 200 ok and words healthy, which will be used by the load balance in a separate thread that will manage the server is connected or not.

/status endpoint will be used by the endpoint called in the load balancer that will get a total aggregate count of all the results of total connections made to the servers each that it counts in an internal counter – it returns the total number of connections that were made to that server and not increment the count of connections.

/connection will be called when a client is connecting that will say new connection accepted and will show the total count of times that it was called.

Note that when we call the /health or /status endopoints, we do not want to increment the count of connections, we only want to increment when /connection endpoint is called.

**Load Balancer Application**

This will handle the main client calls which will have two exposed endpoints:

/admin/connect endpoint which calls the next decided server in the internal list in round robin fashion routing algorithm. This will then return the output response from the server that was connected to. The console of the app will show the connection url that was used for the connection, this will then show the output from that connection server too as a log as well as returned in the API call in postman for this endpoint.

/admin/status endpoint which will call the /status endpoint of each active server, which will then return a list of server names with their connection count total by capturing this response and putting it into the list and return this to the user.

The /health endpoint is going to be automatically called by a separate thread always running on the load balance with the config file that is loaded which specified the total time in seconds to rerun this separate thread, what it will do is check each server is online or not, and have a state called active.

In the end this process of checking in separate thread will mark in the internal list, if the server is active or not by doing an initial health check, then it will rerun this check every 5 seconds, and only update a server state depending on response is only OK and says healthy. Then call the two connection points in the api when necessary will be manually triggered.