**Content Delivery Network**

* CDN is used as middleware network of servers that handles the content delivery of a web application, reducing the load on main server and providing less load time of resource to client.
* In this CDN network, it consists of 1 Controller server and 3 Edge servers.
* Controller server is running in localhost on port 3001.
* Edge servers are situated at E1 -> New Delhi, E2 -> Berlin , E3 -> Montreal, Although they are placed in local system but we are mimicking the behaviour that those are real servers.
* Controller server is running on Express Server in localhost and Edge servers are in local file system.
* Website (Main Server) that is using the CDN is situated at localhost is also running on Express Server.
* The CDN network is created using the Nodejs as backend and MongoDB is being used as Database for keeping track of all the content that is being stored in the Edge Servers.
* Working of CDN -
  + As soon as server is started it starts syncing all the edge servers with cached content.
  + Client sends request for accessing the website.
  + Client’s request goes to the CDN’s Controller server.
  + Based on the request count it keeps on switching the edge servers, and select the current edge server based on the round robin logic.
  + Controller server checks whether the chosen edge server is having the requested content or not.
  + If Edge server has the content, it is served from that Edge server, if it is not stored in Edge Server it is fetched again from the main server and served to the client.
  + If Edge server does not have the requested content, Controller server uses the pull strategy and pulls the content from Main server.
  + Pulled content is stored in the chosen Edge server and Edge server sends the requested content to the Client.
* For resetting all the cached data run <https://localhost:3001/delete-everything> in browser.
* For getting all the logs run <https://localhost:3001/logs> in browser.
* CDN controller is the CDN server, below is the code explanation of controller server.
* **app.js -** It is the main server which runs the express server, it contains all the api routes like /getlogs which provides the logs of server, /delete which deletes all the cached data from DB and edge servers.
* **config.js -** It is a configuration file which contains the configuration of the cdn server like db name, edge server location, web url of website which is using cdn and cronSechdule.
* **cdn.controller.js -** It is a main controller file which has all the business logic, it has multiple functions following -
  + getNearestEdge - This function provides the nearest edge server from the client’s location. It uses geolib library to get the distance between nearest edge server and client based on their coordinates.
  + getFileData - This function returns the file data based on the zone and filename.
  + checkContent - This is the main function which does all the heavy stuff. It first gets the coordinates of client based on its ip address using geoip library then it gets the nearest edge server then it checks if the data is present in the nearest edge server or not. If it is there it moves to next step otherwise it fetches the file from the main server and stores that in edge server and create an entry in DB.
* **cdn.model.js -** This the schema of the DB where every cached entry is stored based on its zone, website name, url, fileName and deleted.
* **cdn.routes.js -** This file contains the main route which serves the cached file.
* **util.js -** This is a helper file which has utility functions like below
  + storeFile - This function uploads the file in Edge server
  + getEdgeServerBucket - This function returns edge server bucket based on the zone name.
  + typeCheck - This function checks the type of requested file.
  + deleteFiles - This function deletes all the files from all the edge servers.

**CDN Dashboard**

* It is dashboard created using Angular.
* Using this we can create multiple configurable edge servers.
* It runs on <https://localhost:4002>
* To start the CDN Dashboard server run - ng serve in terminal.
* Dashboard server will start on port 4200.
* There are 3 mandatory servers which will be created as soon as cdn server starts.
* After that we can create as many configurable edge servers.
* These can be deleted also.

**CDN Controller Deployment**

* There are two steps involved in running the CDN.
  + Running Website
  + Running CDN controller
* Running Website -
  + Install all the libraries -
    - npm i
  + Now run
    - set NODE\_TLS\_REJECT\_UNAUTHORIZED='0' for windows
    - export NODE\_TLS\_REJECT\_UNAUTHORIZED='0' for Macos or Linux
  + Now Start server
    - node app.js
* Original server is running on <https://localhost:3002>
* Running CDN controller -
  + For running CDN first install all the necessary libraries
    - npm install
  + Now run
    - set NODE\_TLS\_REJECT\_UNAUTHORIZED='0' for windows
    - export NODE\_TLS\_REJECT\_UNAUTHORIZED='0' for Macos or Linux
  + Now Start server
    - node app.js
* This will start the express server of cdn controller on <https://localhost:3001>
* Both CDN controller and website is using latest http2 protocol and is the only reason to run server using https in localhost.
* Enable protocol column in chromes developer portal network tab to see the http2 protocol in action.
* Create SSL - <https://devcenter.heroku.com/articles/ssl-certificate-self>