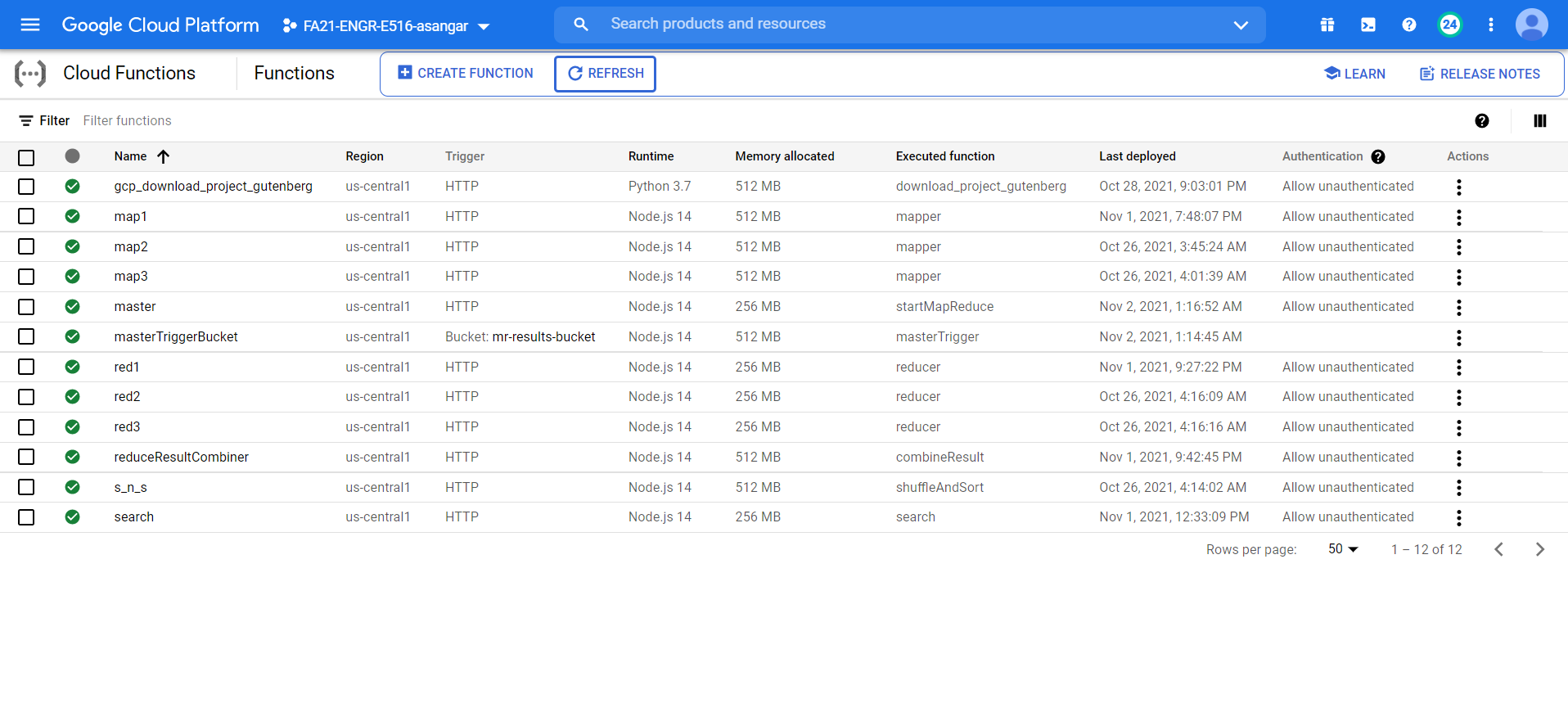
# Report: Map-Reduce with Cloud Functions

1. Run deployment script

**sh deploy\_bucket\_functions.sh**

This script will create two buckets and 12 cloud functions.



1. Copy the trigger command of ‘**search’** **cloud function** and paste it into the first line of the <script> tag (**line 121**) inside **index.html** of search-engine-website folder

var searchURL = "https://us-central1-fa21-engr-e516-asangar.cloudfunctions.net";

1. Run website deployment script inside ‘search-engine-website’ folder

**sh deploy\_website.sh**

Enter information when prompted.

1. Run trigger URL for the “gcp\_download\_project\_gutenberg” cloud function with the mentioned query parameters –

**Parameters –**

bucket\_name=mr-io-bucket

start=[integer\_no]

end=[integer\_no]

Start and end denotes the file range that will be downloaded [START,END)

Example –

<https://us-central1-fa21-engr-e516-asangar.cloudfunctions.net/gcp_download_project_gutenberg?bucket_name=mr-io-bucket&start=100&end=120>

This will download files from project Gutenberg and store it in mr-io-bucket.

**Result** – You can see the files downloaded inside mr-io-bucket. There’s also a mapping file(mr-file-mappings.txt) for keeping track of filenames and their URLs.

1. Run trigger URL for the “master” cloud function with the mentioned query parameters –

**Parameters –**

start=[integer\_no]

end=[integer\_no]

Start and end denotes the file range that will be downloaded [START,END)

Example –

<https://us-central1-fa21-engr-e516-asangar.cloudfunctions.net/master?start=100&end=115>

This will start the map-reduce operation and the results of each step can be seen by monitoring the mr-results-bucket.

**Result -** Finally, you will get the inverted indexes for documents inside **final\_output.json** file.

1. **Streaming Search**

**map1** cloud function can be used to update indexes inside the **final\_output.json.** Since this is a http trigger function, we have to trigger it manually. The same implementation can be used to address new documents as they are added to the corpus by changing the trigger type from http to bucket add/modify.

**Parameters –**

start=[integer\_no]

end=[integer\_no]

update=true

Start and end denotes the file range that will be downloaded [START,END)

update flag is used for appending results to **final\_output.json**

**Example –**

<https://us-central1-fa21-engr-e516-asangar.cloudfunctions.net/map1?start=115&end=117&update=true>

**Flow –**

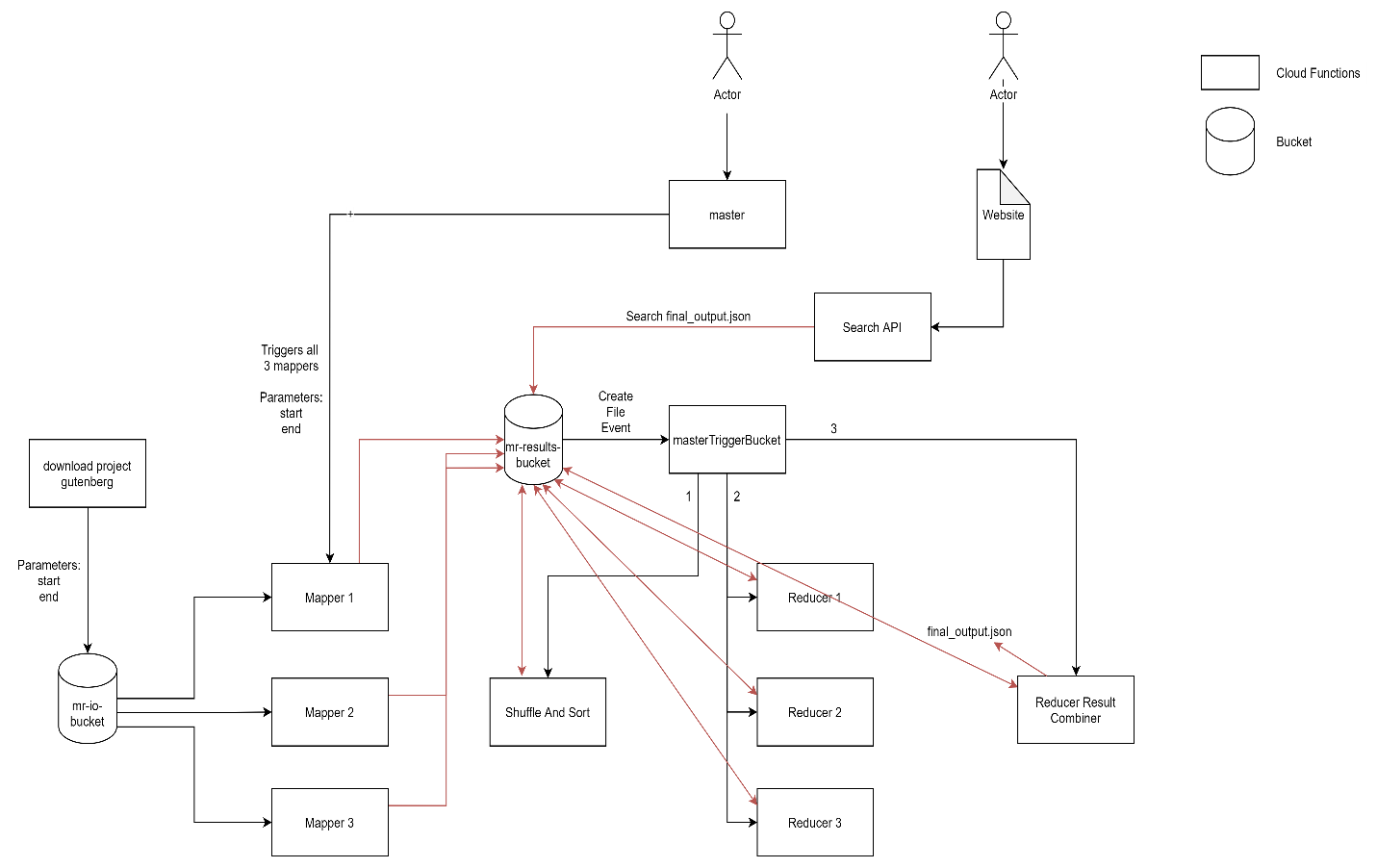
Mapper1 -> Reducer1 -> Reducer Result Combiner -> Update file in bucket

Mapper1 Output - reducer1\_input\_update.json

Reducer1 Output - reducer1\_output\_update.json

Reducer Result Combiner – Combines ‘reducer1\_output\_update.json’ and ‘final\_output.json’ and saves as ‘final\_output.json’

1. **Architecture**



1. **Cost Estimation**

The functions like master and masterTriggerBucket are only used to invoke other functions and thus runs for a very little time. Since they are 256MB machines, the cost to operate is $0.000000648/100ms.

The cost to run other 512MB cloud function is $0.000001295/100ms. The major cost will be generated from functions like mappers, reducers, shuffleAndSort and reduceResultCombiner as these are major running functions.

The search function although runs for a short time, it will also contribute greatly to the cost because of the huge number of invocations by different frontend websites.

Since all these functions are deployed in the same region, the networking cost i.e., inbound and outbound data to operate multiple functions is free.

1. **Web search**

Type in the search box and wait for results.



