

Table of Contents

Table of Contents	1
Getting Started	2
Script Execution Instructions	7
Total Spending	8
Appendix	9

Getting Started

1. Run the Python script to generate the web pages:

```
python3 generate-content.py
```

2. Make the storage bucket:

```
gcloud storage buckets create gs://aru-cds561--location  
us-centrall
```

3. Copy the files from the script into the bucket:

```
gcloud storage cp -r ./files  
gs://aru-cds561/mini-web
```

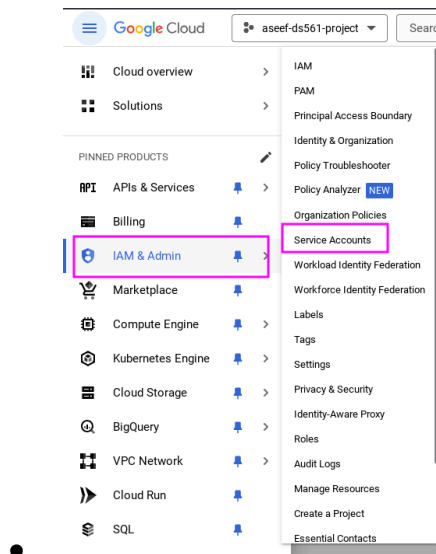
4. Now, we need to create a service account to access the “mini web” through:

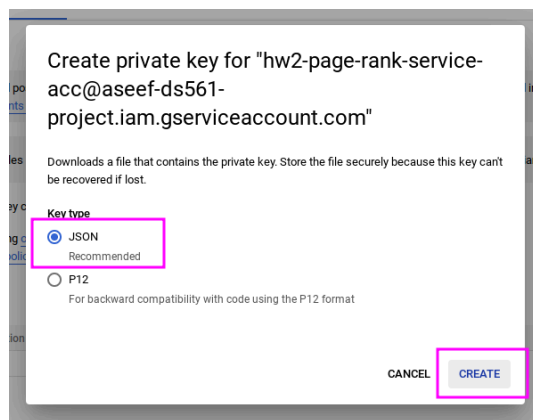
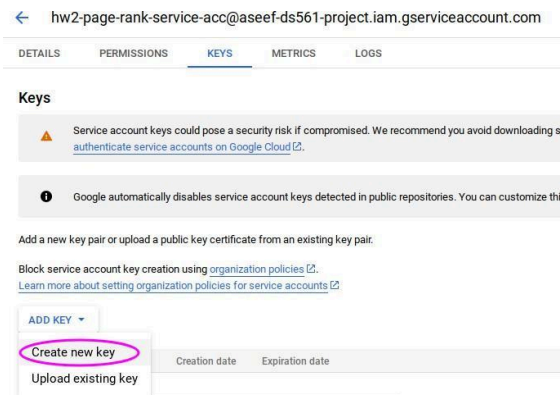
```
gcloud iam service-accounts create hw2-page-rank-service-acc
```

5. But the service account can't do anything right now (it has no permissions). So, lets give it full access to our new bucket.

```
gcloud storage buckets add-iam-policy-binding  
gs://aru-cds561--member="serviceAccount:hw2-pa  
ge-rank-service-acc@aru-ds561-project.iam.gserviceaccount.com  
" --role="roles/storage.admin"
```

6. Now, I will go ahead and download the credentials using the google cloud web console as shown below:





- Now, we must move the credentials into a safe location and modify the .env parameters of our script such that the env variable `GOOGLE_APPLICATION_CREDENTIALS` points to the .json credential file.

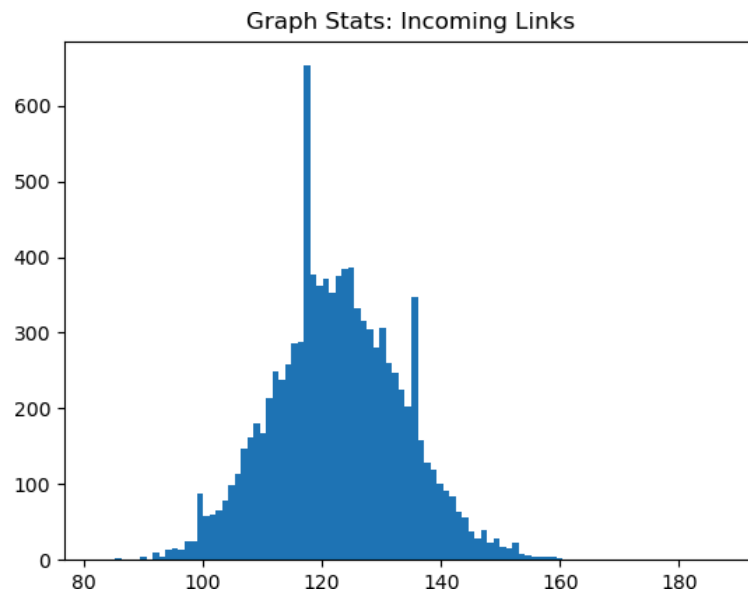
8. Finally, we run the `page_ranker.py` script as in the GitHub repository. Further details on using this script are provided below. Output of this script is shown below:

```
--Graph Stats: Outgoing Links--
Average: 122.64
0th Percentile [Min]: 0
20th Percentile: 49
40th Percentile: 98
50th Percentile [Median]: 123.0
60th Percentile: 147
80th Percentile: 196
100th Percentile [Max]: 249

--Graph Stats: Incoming Links--
Average: 122.64
0th Percentile [Min]: 82
20th Percentile: 113
40th Percentile: 120
50th Percentile [Median]: 123.0
60th Percentile: 125
80th Percentile: 132
100th Percentile [Max]: 188

Here are the top 5 page ranks:
#1: 2526.html.html (with a normalized page rank score of 0.00023819)
#2: 6846.html.html (with a normalized page rank score of 0.00021512)
#3: 5971.html.html (with a normalized page rank score of 0.00020965)
#4: 5778.html.html (with a normalized page rank score of 0.00020819)
#5: 1058.html.html (with a normalized page rank score of 0.00020663)
```

a.



b.

- i. There was a $1/10000$ probability that some node forms a outgoing link to any other given node with 0-250 trials. And that this trial itself was repeated 10,000 times. So, a binomial distribution is expected (because when you run binomial trials using probabilities sampled from a binomial distribution, the result is binomial).
- ii. Then, with a $1/10000$ probability of forming a connection between two nodes with an average of 125 trials, there is an average of 0.0125 probability for each trial. This means we can expect a distribution mean of $(125/10000) * 10000 = 125$, which is indeed the case!

