Assignment 3

Software and Data Engineering

Instructure: Sumit Kalra Deepti Gupta(M21CS005)

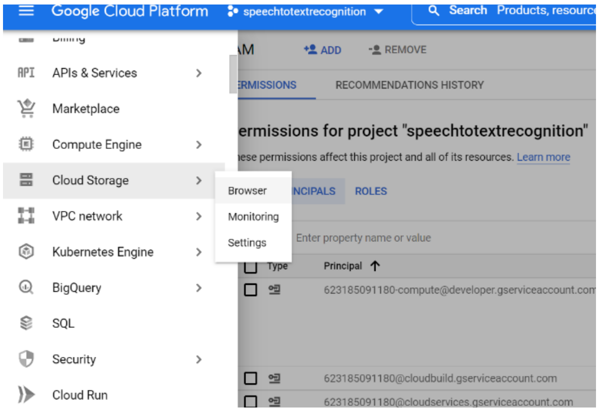
YouTube Link: https://www.youtube.com/watch?v=vSqTSqO6ga4

Qus1: Create a demo application using all the three services used by each other. For example, create a sample application that accesses storage service to retrieve the data and uses Cloud AI service to perform AI-task. Host this application on the computing services.

- I have created web application Speech Recognition with Python and Flask using Cloud AI API. In Which I have Used Cloud Bucket as a storage service for accessing the audio file and after that these audio file will be converted into text using google Cloud Speech API.

Step 1: Store the data on Google Storage

1) Created Bucket Via GCP Console



ermissions for project "speechtotextrecognition"

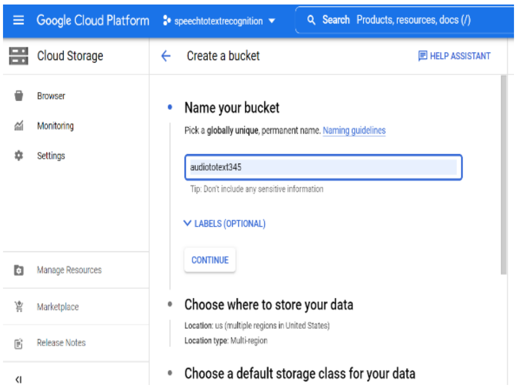
ese permissions affect this project and all of its resources. Learn more

623185091180-compute@developer.gserviceaccount.com

623185091180@cloudbuild.gserviceaccount.com

623185091180@cloudservices.gserviceaccount.com

2) Entered the name and selected the storage class.



Name your bucket

Pick a globally unique, permanent name. Naming guidelines

audiototext345

Tip: Don't include any sensitive information

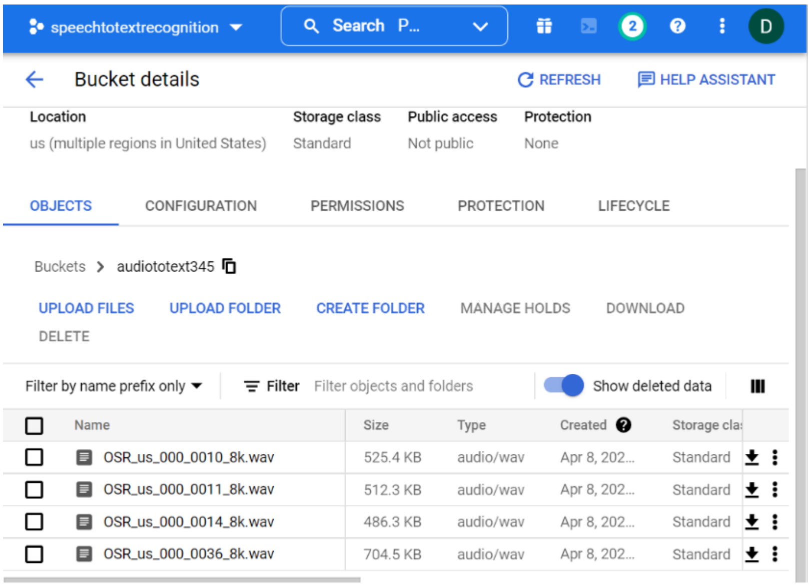
Choose where to store your data

Location: us (multiple regions in United States)

Location type: Multi-region

Choose a default storage class for your data

3) Uploaded some Audio files in Bucket:

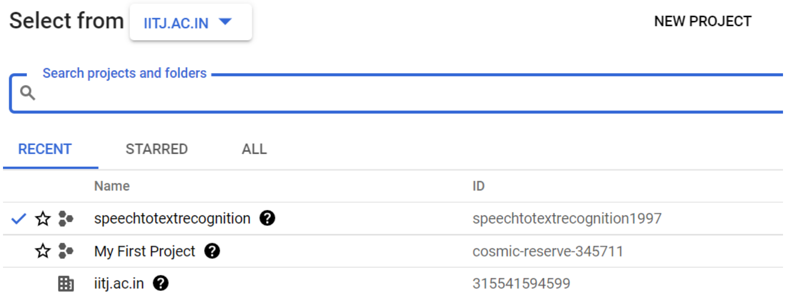


Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	None

□	Name	Size	Type	Created	Storage cla:
□	OSR_us_000_0010_8k.wav	525.4 KB	audio/wav	Apr 8, 202...	Standard
□	OSR_us_000_0011_8k.wav	512.3 KB	audio/wav	Apr 8, 202...	Standard
□	OSR_us_000_0014_8k.wav	486.3 KB	audio/wav	Apr 8, 202...	Standard
□	OSR_us_000_0036_8k.wav	704.5 KB	audio/wav	Apr 8, 202...	Standard

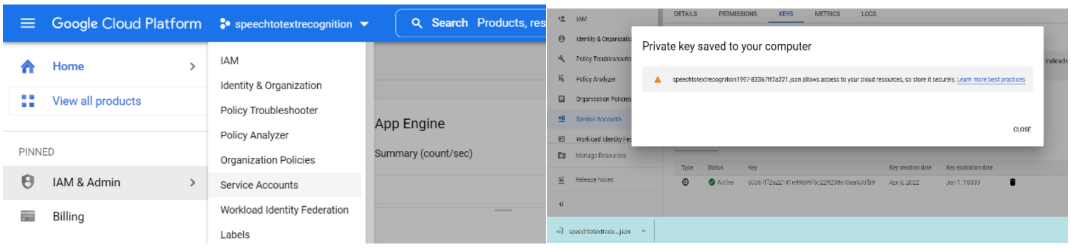
Step 2: AI Platform (Cloud AI):

- First, I have created a Project Named **speechtotextrecognition**



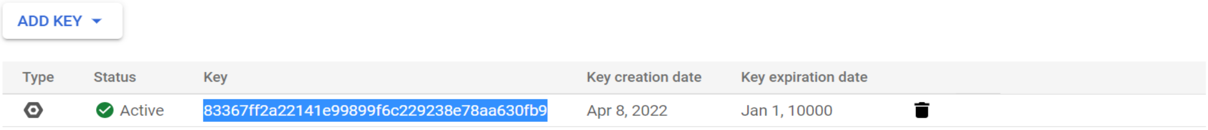
Name	ID
speechtotextrecognition	speechtotextrecognition1997
My First Project	cosmic-reserve-345711
iitj.ac.in	315541594599

- Downloaded the JSON file as it will authenticate you to the cloud API



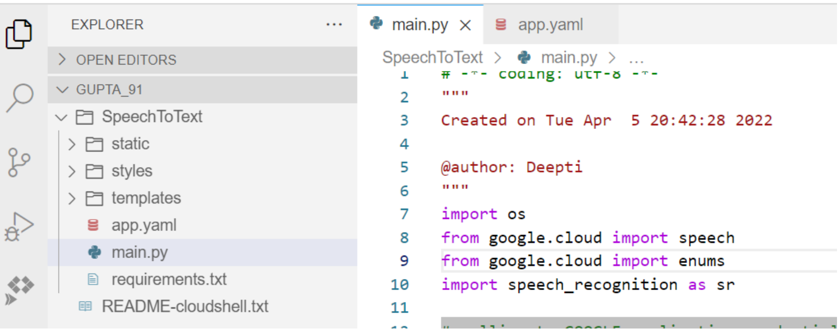
Private key saved to your computer

speechtotextrecognition1997-83367ff2a221.json allows access to your cloud resources, so store it securely. Learn more best practices



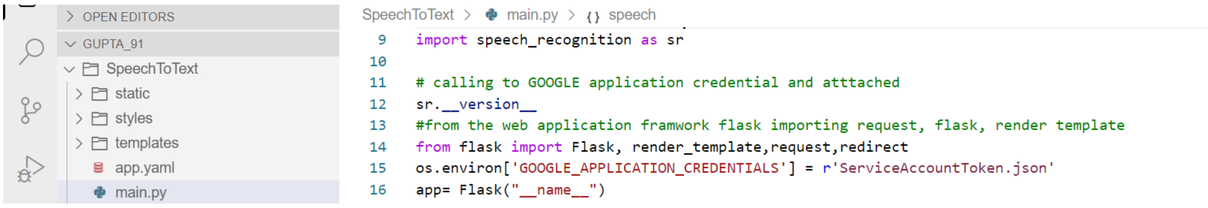
Type	Status	Key	Key creation date	Key expiration date
	Active	83367ff2a22141e99899f6c229238e78aa630fb9	Apr 8, 2022	Jan 1, 10000

For using cloud speech API in my app, I install google cloud speech package using **pip install google-cloud-speech** in cloud terminal. And then import all required packages.



```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr  5 20:42:28 2022

@author: Deepti
"""
import os
from google.cloud import speech
from google.cloud import enums
import speech_recognition as sr

```

JSON file include:

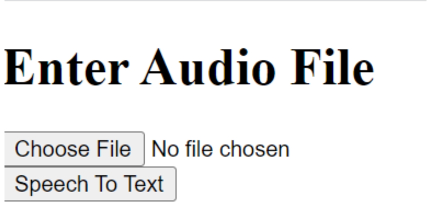


```
import speech_recognition as sr

# calling to GOOGLE application credential and atttached
sr.__version__
#from the web application framwork flask importing request, flask, render template
from flask import Flask, render_template,request,redirect
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = r'ServiceAccountToken.json'
app= Flask("__name__")
```

- I have Created Flask Framework for Web app and inside this framework I have used Cloud AI Functionality.
  1) First, I have created A Simple Flask Application and create the First Route Pointing to Home Page and specify both the method GET and POST.

     Like: @app.route("/index", methods= ["GET", "POST"])

- Created index.html contain the Simple User Interface like:



Enter Audio File

First, I choose the Audio File using choose file button.
And then Convert that Audio file into Text using Speech to Text Button.

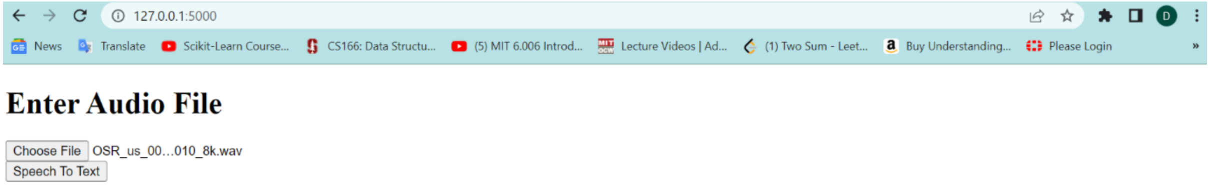
- Now for Recognition work
  1) redirect handler function will redirect your Audio file for transcription using redirect(request.url)
  2) Here I Directly uses the Google cloud AI speech to text conversion on that redirected file which stored in Buckets.
  3) Recognizer Model work at character level so that as you speak, it outputs words character-by-character.

4) I have already imported cloud API as (from google. cloud import speech import speech_recognition as sr) Conversion is started using sr.AudioFile (Audio_file).
5) Transcript Store the Converted data using:
   transcript = recognizer.recognize_google(Txt_file, key=None)

6) At last, return the text file using render template to index.html transcript.

- And Now after all these staps my app working fine in my local host:

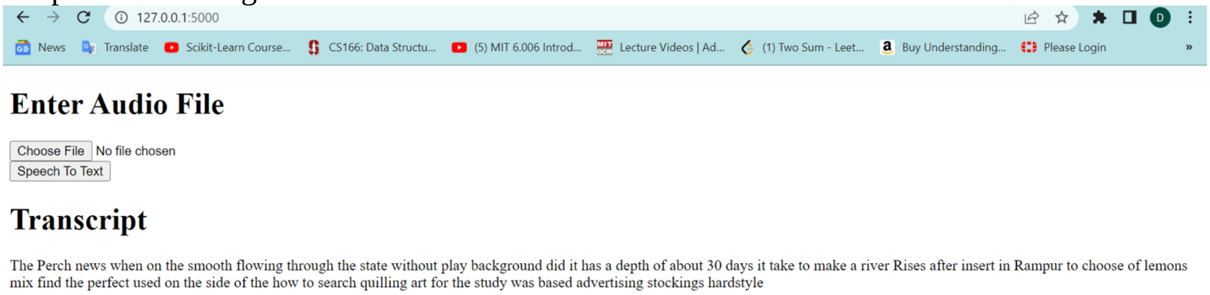
  Choose the Audio file for Recognition:



Enter Audio File

Choose File OSR_us_00...010_8k.wav
Speech To Text

Output After Recognition:



Enter Audio File

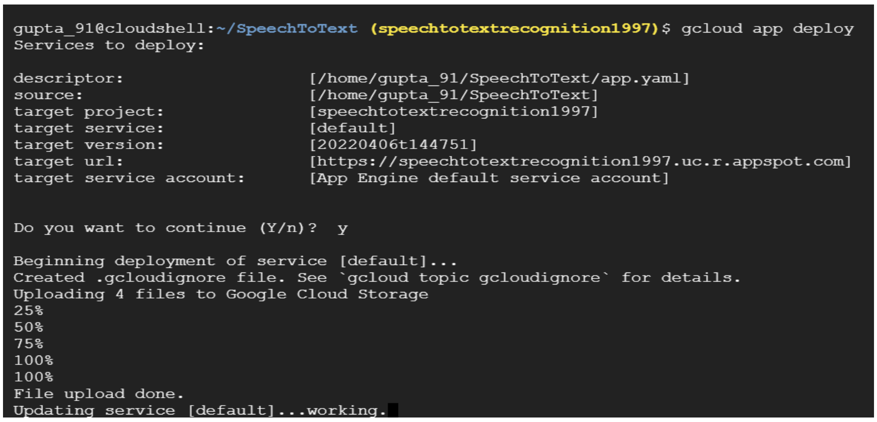
Choose File No file chosen
Speech To Text

Transcript

The Perch news when on the smooth flowing through the state without play background did it has a depth of about 30 days it take to make a river Rises after insert in Rampur to choose of lemons mix find the perfect used on the side of the how to search quilling art for the study was based advertising stockings hardstyle

Step 3: Deployment of app using App engine-

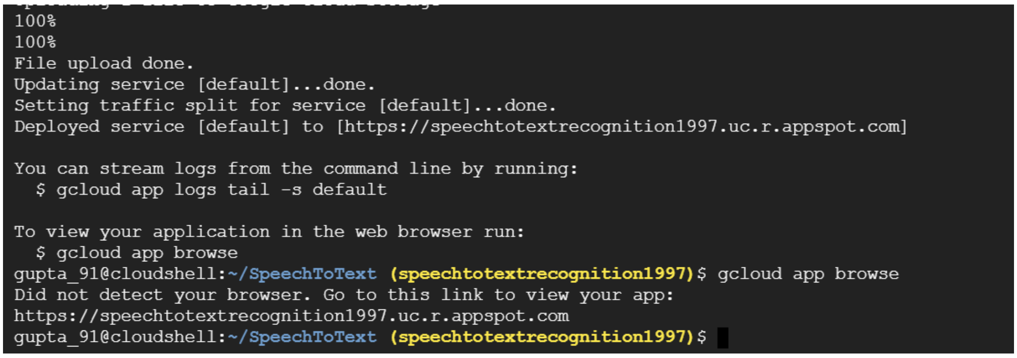
1) In App engine First selected the already created app speechtotextrecognition.
2) Open the cloud shell where I have already created the main.py and index.html file now I have also created two more files app.yaml and requirement.txt
3) App.yaml, this file **specifies how URL paths correspond to request handlers and static files**. The app. yaml file also contains information about your app's code, such as the runtime and the latest version identifier.
4) Dependencies of web app are specified in requirements.txt file.
5) After this I opened the cloud shell and run some commands for deploying the app.



```
gupta_91@cloudshell:~/SpeechToText (speechtotextrecognition1997)$ gcloud app deploy
Services to deploy:

descriptor:                  [/home/gupta_91/SpeechToText/app.yaml]
source:                      [/home/gupta_91/SpeechToText]
target project:              [speechtotextrecognition1997]
target service:              [default]
target version:              [20220406t144751]
target url:                  [https://speechtotextrecognition1997.uc.r.appspot.com]
target service account:      [App Engine default service account]


Do you want to continue (Y/n)?  y

Beginning deployment of service [default]...
Created .gcloudignore file. See `gcloud topic gcloudignore` for details.
Uploading 4 files to Google Cloud Storage
25%
50%
75%
100%
100%
File upload done.
Updating service [default]...working.
```

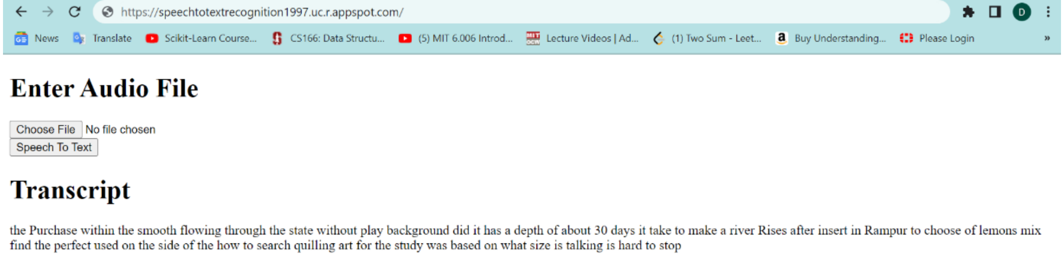
6) As You can see all 4 files are uploaded to cloud storage and deploying is in process.

7) Then Deploying has been finished after some time and URL generated:



```
100%
100%
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://speechtotextrecognition1997.uc.r.appspot.com]

You can stream logs from the command line by running:
  $ gcloud app logs tail -s default

To view your application in the web browser run:
  $ gcloud app browse
gupta_91@cloudshell:~/SpeechToText (speechtotextrecognition1997)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://speechtotextrecognition1997.uc.r.appspot.com
gupta_91@cloudshell:~/SpeechToText (speechtotextrecognition1997)$
```

8) Output After Deployment using generated URL is:



Enter Audio File

Choose File No file chosen
Speech To Text

Transcript

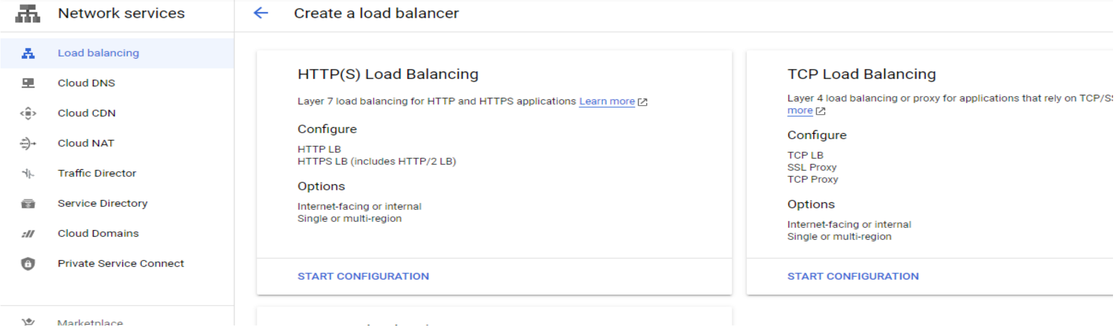
the Purchase within the smooth flowing through the state without play background did it has a depth of about 30 days it take to make a river Rises after insert in Rampur to choose of lemons mix find the perfect used on the side of the how to search quilling art for the study was based on what size is talking is hard to stop

Qus2) Deploy the same application instance on another virtual machine using compute service and include the load balancer to distribute the task between these two application instances. You can differentiate these two application instances by using slightly different text to show the outcome.

Answer: In contrast to hardware-based load balancing, which is more common in enterprise data centres, cloud load balancing distributes network traffic across resources using software. A load balancer receives incoming traffic and routes it to active targets based on a policy that has been configured.

Creation of load balancing is described by below steps:

- Login to cloud.
- Move to network services -> Create a load balancer



HTTP(S) Load Balancing

Layer 7 load balancing for HTTP and HTTPS applications Learn more

Configure

HTTP LB
HTTPS LB (includes HTTP/2 LB)

Options

Internet-facing or internal
Single or multi-region

START CONFIGURATION

TCP Load Balancing

Layer 4 load balancing or proxy for applications that rely on TCP/S
more

Configure

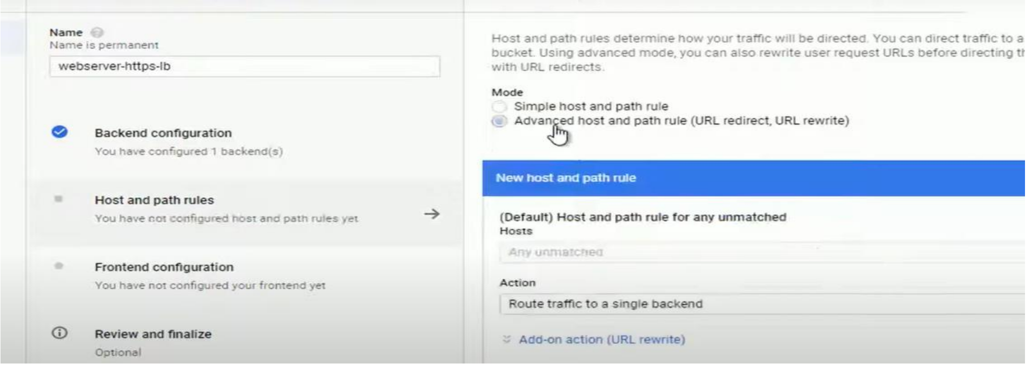
TCP LB
SSL Proxy
TCP Proxy

Options

Internet-facing or internal
Single or multi-region

START CONFIGURATION

- Creating a new load balancer.



Name
Name is permanent
webserver-https-lb

✓ Backend configuration
You have configured 1 backend(s)

Host and path rules
You have not configured host and path rules yet

Frontend configuration
You have not configured your frontend yet

Review and finalize
Optional

Host and path rules determine how your traffic will be directed. You can direct traffic to a bucket. Using advanced mode, you can also rewrite user request URLs before directing tl with URL redirects.

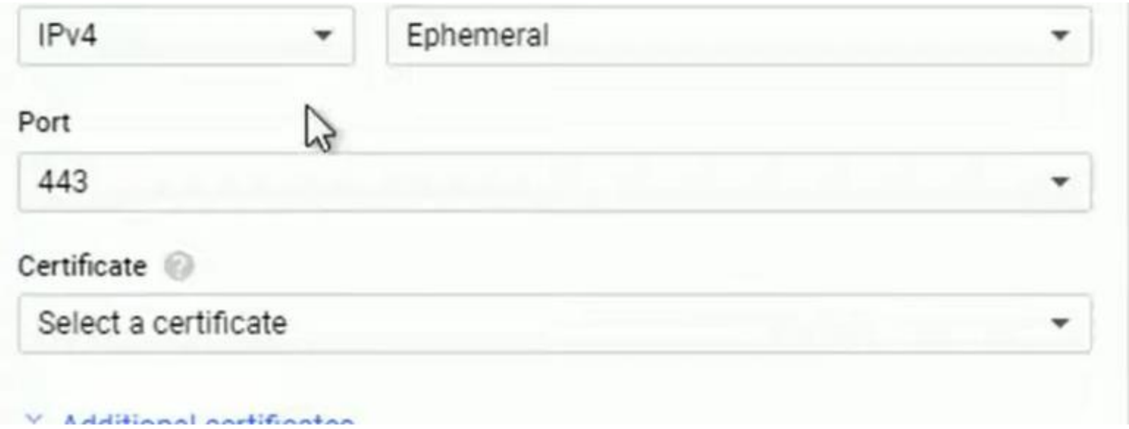
Mode
○ Simple host and path rule
● Advanced host and path rule (URL redirect, URL rewrite)

New host and path rule

(Default) Host and path rule for any unmatched
Hosts
Any unmatched

Action
Route traffic to a single backend

Add-on action (URL rewrite)

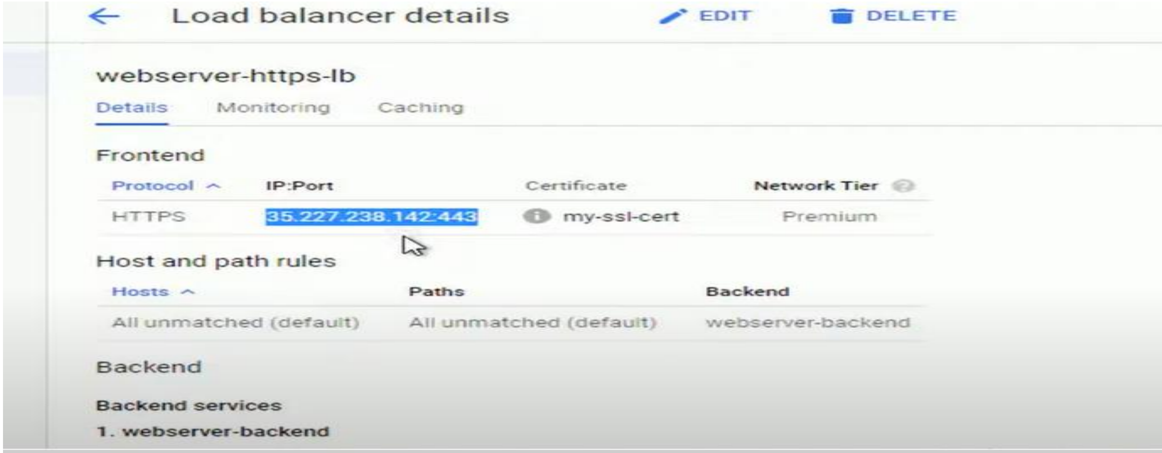


IPv4 | Ephemeral

Port
443

Certificate
Select a certificate

Additional certificates



Load balancer details

webserver-https-lb

Frontend

Protocol	IP:Port	Certificate	Network Tier
HTTPS	35.227.238.142:443	my-ssl-cert	Premium

Host and path rules

Hosts	Paths	Backend
All unmatched (default)	All unmatched (default)	webserver-backend

Backend

Backend services
1. webserver-backend