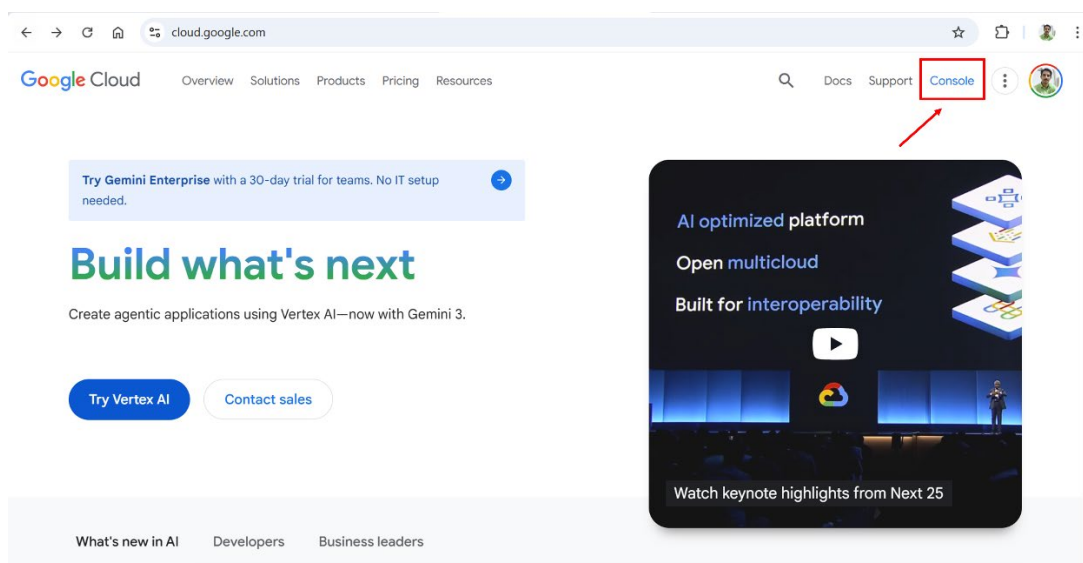


CLOUD COMPUTING LABORATORY

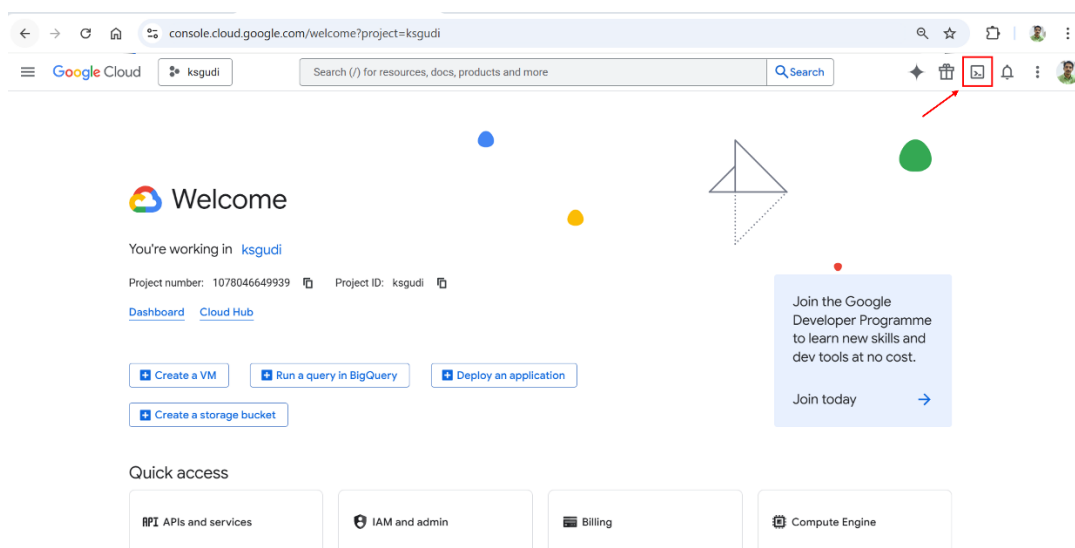
Experiment – 4:

Deploy a web application on App Engine with automatic scaling enabled.

Step 1: From the Google Cloud homepage, click on **Console** to open the Google Cloud Console.



Step 2: Open Google Cloud console and click the **Cloud Shell** icon to launch the terminal.



Step 3: Execute the command `gcloud app create` in Cloud Shell to start creating an App Engine application.

```

CLOUD SHELL
Terminal (ksgudi) x +
Gemini CLI is available in Cloud Shell terminal. Type gemini to try it. Learn more

Your Cloud Platform project in this session is set to ksgudi.
Use 'gcloud config set project [PROJECT_ID]' to change to a different project.
krishna_gudi@cloudshell:~ (ksgudi)$ gcloud app create
You are creating an app for project [ksgudi].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. More information about regions is at
<https://cloud.google.com/appengine/docs/locations>.

WARNING: Starting from March, 2025, App Engine sets the automatic scaling maximum instances
default for standard environment deployments to 20. This change doesn't impact
existing apps. To override the default, specify the new max_instances value in your
app.yaml file, and deploy a new version or redeploy over an existing version.
For more details on max_instances, see
<https://cloud.google.com/appengine/docs/standard/reference/app-yaml.md#scaling_elements>.

NOTE: Cloud Run offers the most modern fully managed application hosting experience
with lower minimum billable times and support for GPUs on demand for your AI/ML workloads.
Deploy code written in any programming language supported by App Engine on Cloud Run.
Learn more at https://cloud.google.com/run/docs/quickstarts/build-and-deploy-a-web-service

Please choose the region where you want your App Engine application located:

[1] asia-east1      (supports standard and flexible)
[2] asia-east2      (supports standard and flexible and search_api)
[3] asia-northeast1 (supports standard and flexible and search_api)
[4] asia-northeast2 (supports standard and flexible and search_api)
  
```

View the list of available regions displayed in the terminal for App Engine deployment. Enter the region choice (e.g., 6) to create the App Engine application successfully.

```

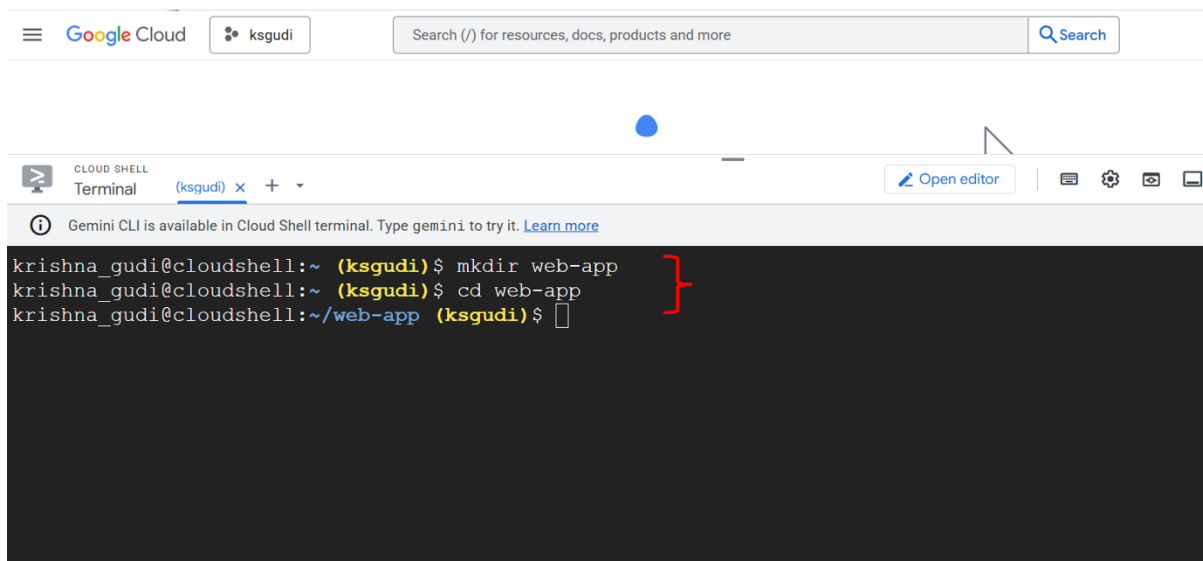
CLOUD SHELL
Terminal (ksgudi) x +
Gemini CLI is available in Cloud Shell terminal. Type gemini to try it. Learn more

[4] asia-northeast2 (supports standard and flexible and search_api)
[5] asia-northeast3 (supports standard and flexible and search_api)
[6] asia-south1     (supports standard and flexible and search_api)
[7] asia-southeast1 (supports standard and flexible)
[8] asia-southeast2 (supports standard and flexible and search_api)
[9] australia-southeast1 (supports standard and flexible and search_api)
[10] europe-central2 (supports standard and flexible)
[11] europe-west     (supports standard and flexible and search_api)
[12] europe-west2    (supports standard and flexible and search_api)
[13] europe-west3    (supports standard and flexible and search_api)
[14] europe-west6    (supports standard and flexible and search_api)
[15] northamerica-northeast1 (supports standard and flexible and search_api)
[16] southamerica-east1 (supports standard and flexible and search_api)
[17] us-central      (supports standard and flexible and search_api)
[18] us-east1        (supports standard and flexible and search_api)
[19] us-east4        (supports standard and flexible and search_api)
[20] us-west1        (supports standard and flexible)
[21] us-west2        (supports standard and flexible and search_api)
[22] us-west3        (supports standard and flexible and search_api)
[23] us-west4        (supports standard and flexible and search_api)
[24] cancel

Please enter your numeric choice: 6

Creating App Engine application in project [ksgudi] and region [asia-south1]...done.
Success! The app is now created. Please use 'gcloud app deploy' to deploy your first app.
krishna_gudi@cloudshell:~ (ksgudi)$
  
```

Step 4: Create a new project directory and navigate into it.



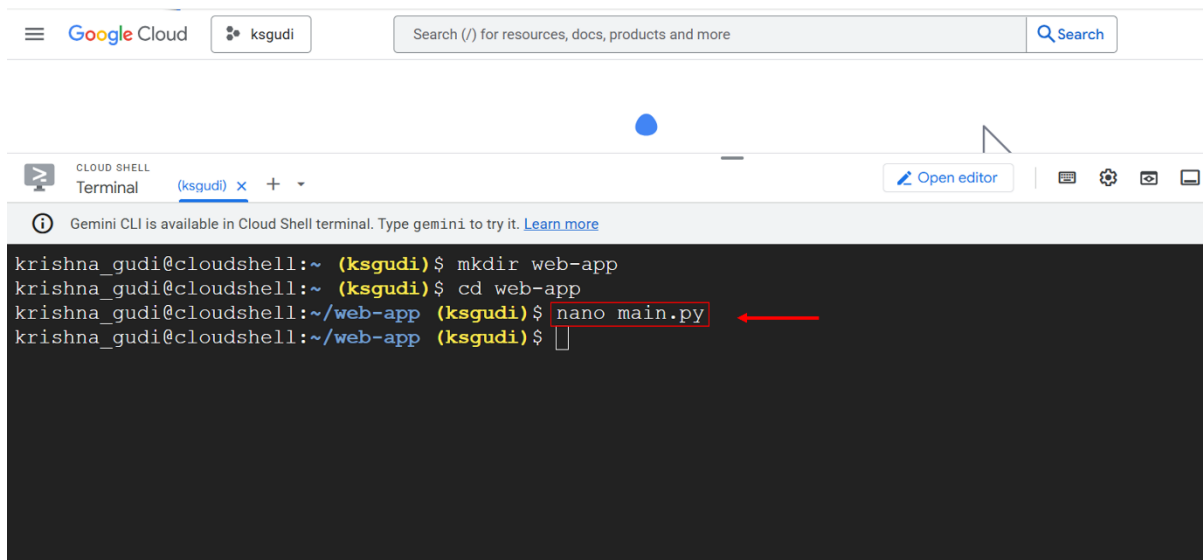
The screenshot shows a Google Cloud Shell terminal window. The terminal output is as follows:

```
krishna_gudi@cloudshell:~ (ksgudi) $ mkdir web-app
krishna_gudi@cloudshell:~ (ksgudi) $ cd web-app
krishna_gudi@cloudshell:~/web-app (ksgudi) $
```

A red bracket on the right side of the terminal highlights the two commands: `mkdir web-app` and `cd web-app`.

Step 5: Create the Python application file by executing the command:

```
nano main.py
```

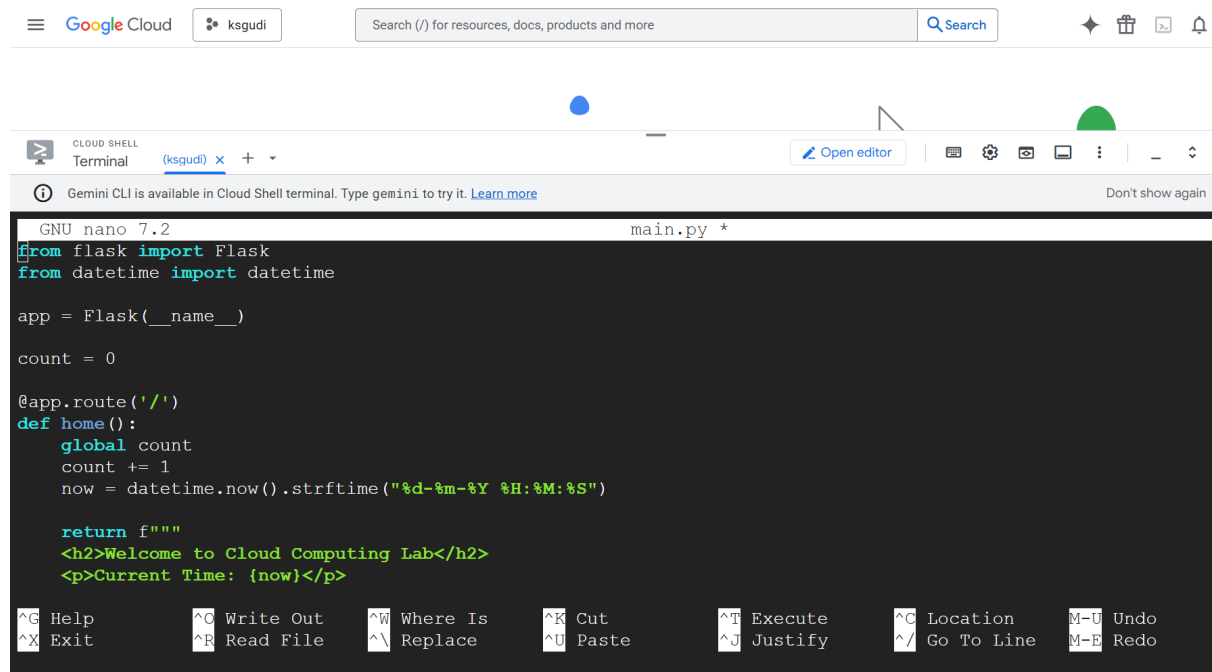


The screenshot shows a Google Cloud Shell terminal window. The terminal output is as follows:

```
krishna_gudi@cloudshell:~ (ksgudi) $ mkdir web-app
krishna_gudi@cloudshell:~ (ksgudi) $ cd web-app
krishna_gudi@cloudshell:~/web-app (ksgudi) $ nano main.py
krishna_gudi@cloudshell:~/web-app (ksgudi) $
```

A red box highlights the command `nano main.py`, and a red arrow points to it from the right.

Write the Flask web application code inside the **main.py** file using the nano editor.



```
GNU nano 7.2 main.py *
from flask import Flask
from datetime import datetime

app = Flask(__name__)

count = 0

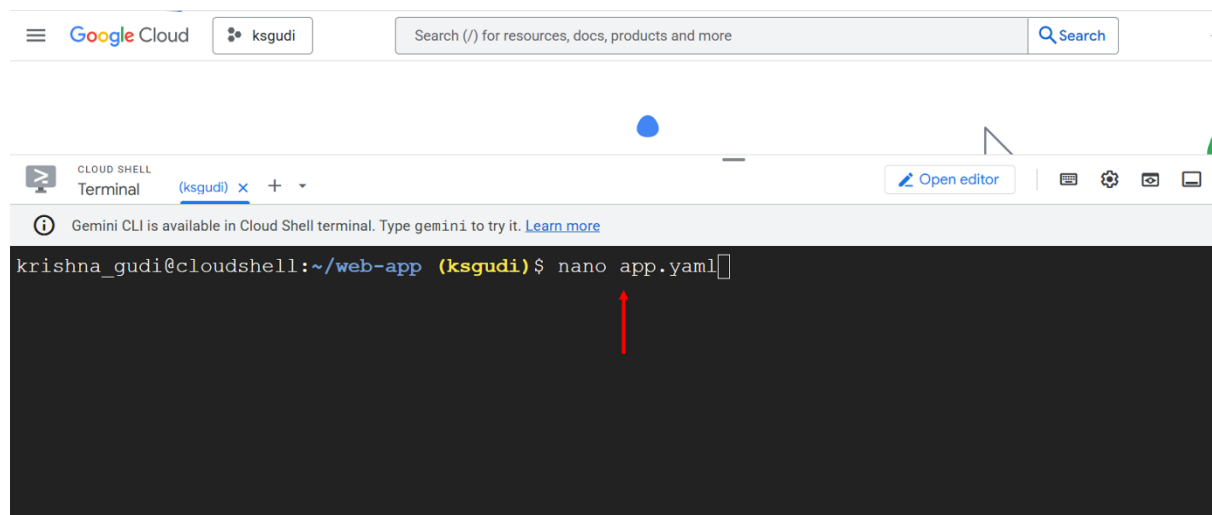
@app.route('/')
def home():
    global count
    count += 1
    now = datetime.now().strftime("%d-%m-%Y %H:%M:%S")

    return f"""
    <h2>Welcome to Cloud Computing Lab</h2>
    <p>Current Time: {now}</p>
    """

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

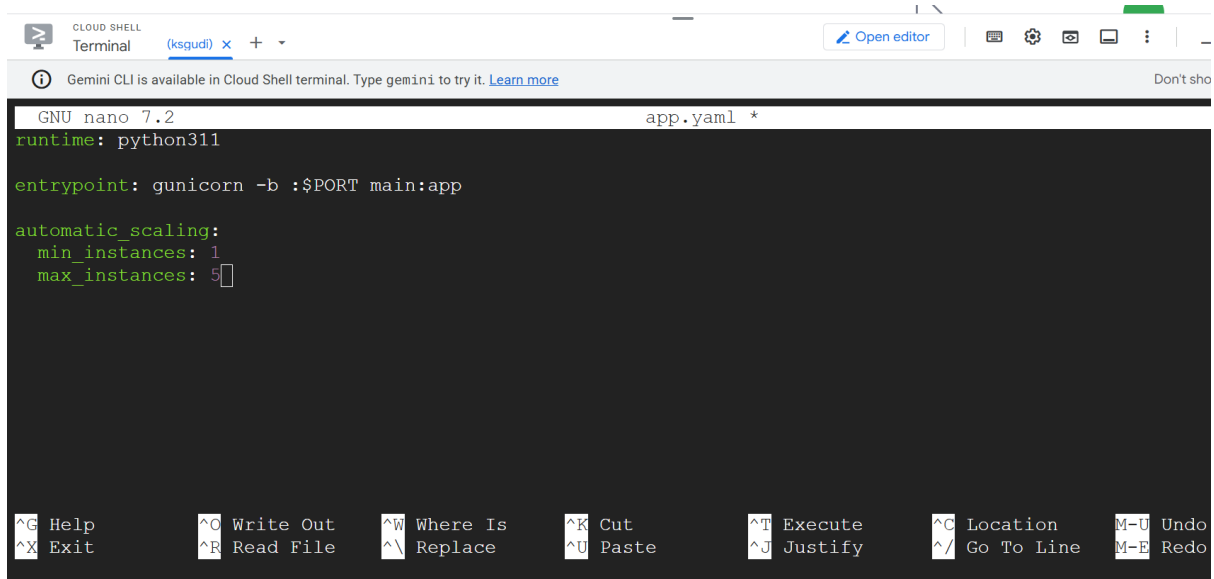
Step 6: Create the App Engine configuration file using the command:

```
nano app.yaml
```



```
krishna_gudi@cloudshell:~/web-app (ksgudi) $ nano app.yaml
```

Add **runtime, entry point, and scaling configuration** in the `app.yaml` file.



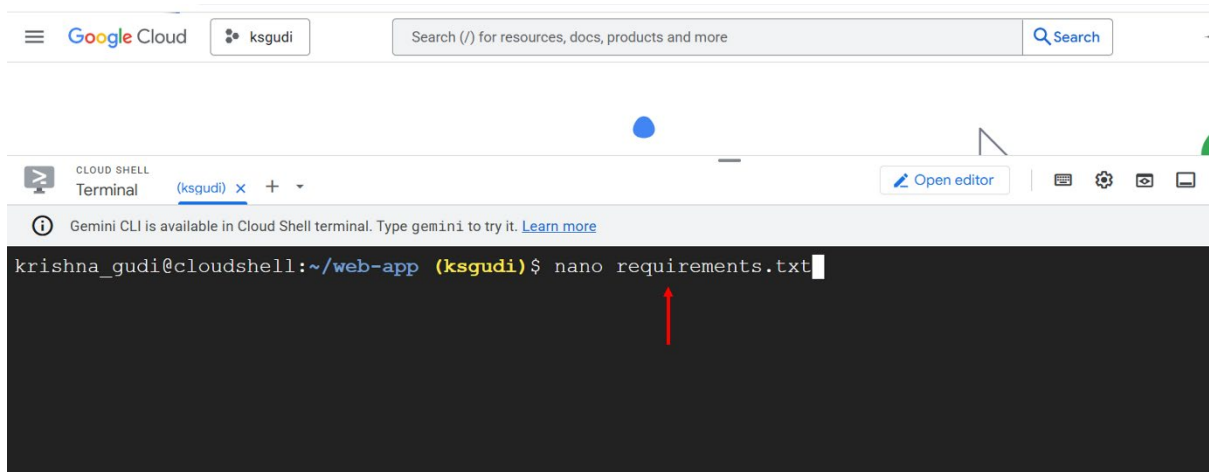
```
GNU nano 7.2 app.yaml *
runtime: python311

entrypoint: gunicorn -b :$PORT main:app

automatic_scaling:
  min_instances: 1
  max_instances: 5
```

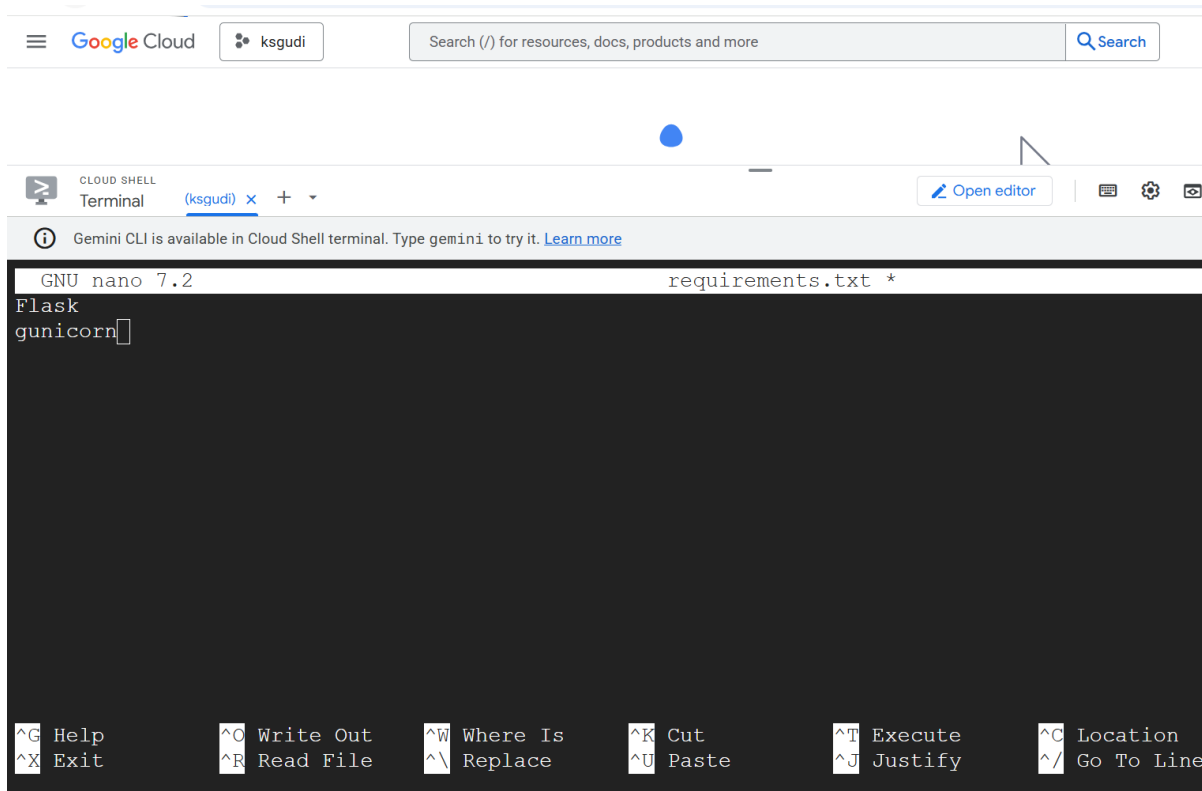
Step 7: Create the dependency file using the command:

```
nano requirements.txt
```



```
krishna_gudi@cloudshell:~/web-app (ksgudi)$ nano requirements.txt
```

Specify the required libraries (Flask and gunicorn) in the requirements.txt file.



```
Google Cloud ksgudi Search (/) for resources, docs, products and more Search
```

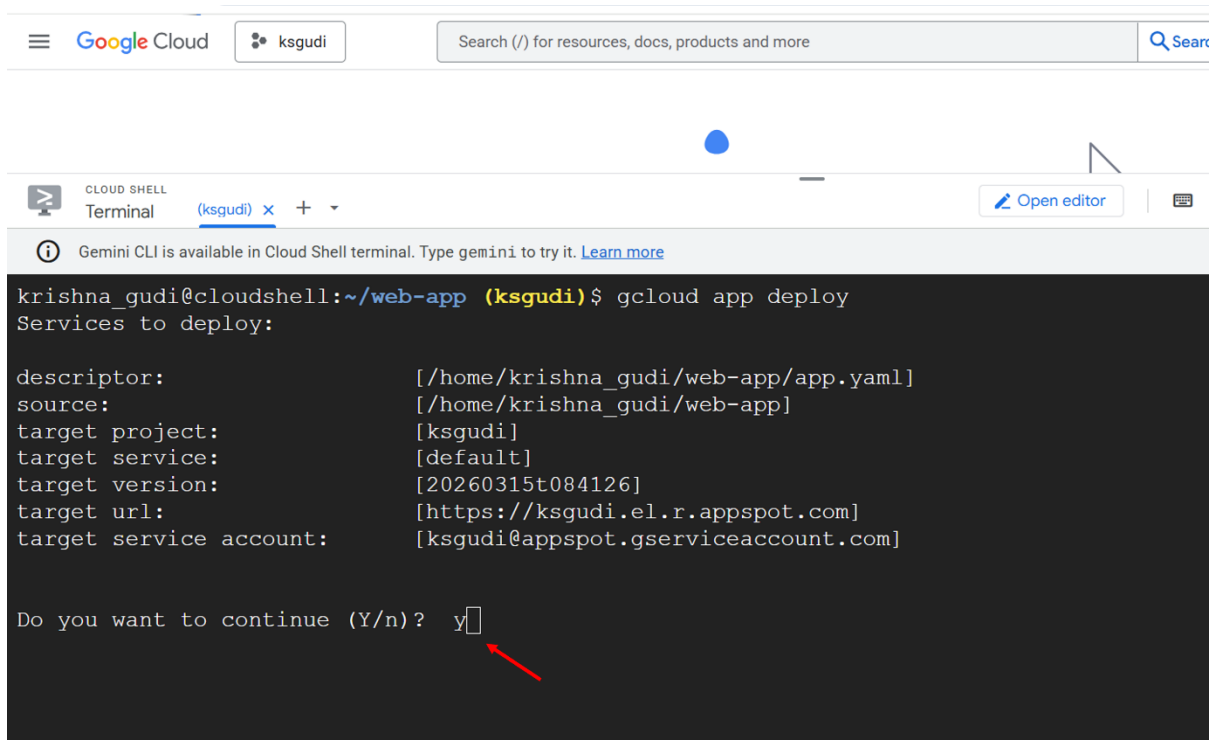
```
Cloud Shell Terminal (ksgudi) x + Open editor
```

```
Gemini CLI is available in Cloud Shell terminal. Type gemini to try it. Learn more
```

```
GNU nano 7.2 requirements.txt *
Flask
gunicorn
```

```
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_/ Go To Line
```

Step 8: Deploy the application to App Engine by running the command “**gcloud app deploy**” and confirm the deployment



```
Google Cloud ksgudi Search (/) for resources, docs, products and more Search
```

```
Cloud Shell Terminal (ksgudi) x + Open editor
```

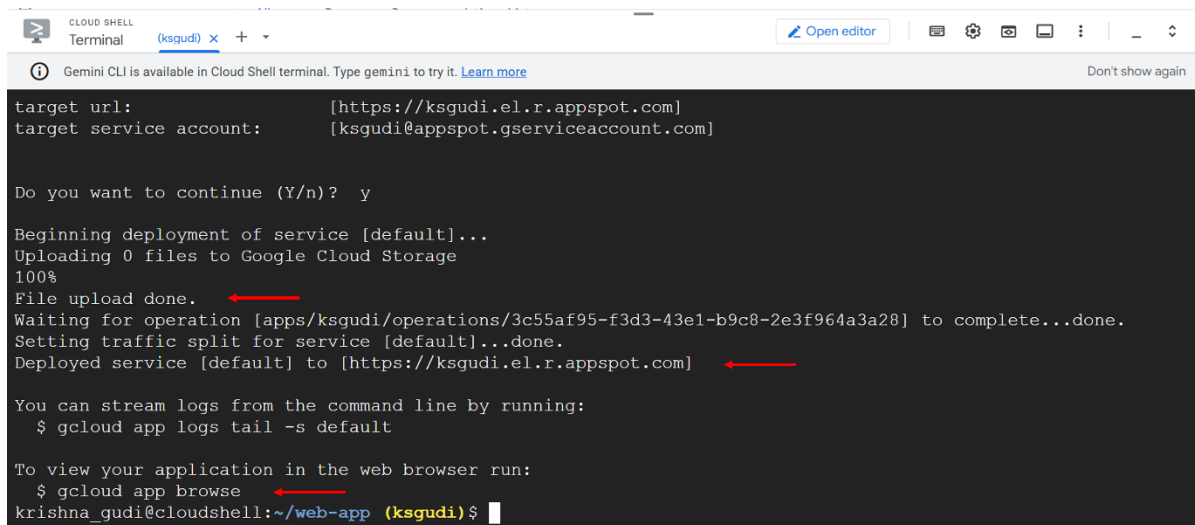
```
Gemini CLI is available in Cloud Shell terminal. Type gemini to try it. Learn more
```

```
krishna_gudi@cloudshell:~/web-app (ksgudi) $ gcloud app deploy
Services to deploy:

descriptor:          [/home/krishna_gudi/web-app/app.yaml]
source:              [/home/krishna_gudi/web-app]
target project:      [ksgudi]
target service:      [default]
target version:      [20260315t084126]
target url:          [https://ksgudi.el.r.appspot.com]
target service account: [ksgudi@appspot.gserviceaccount.com]

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

Observe the successful deployment message and application URL.



```
target url: [https://ksgudi.el.r.appspot.com]
target service account: [ksgudi@appspot.gserviceaccount.com]

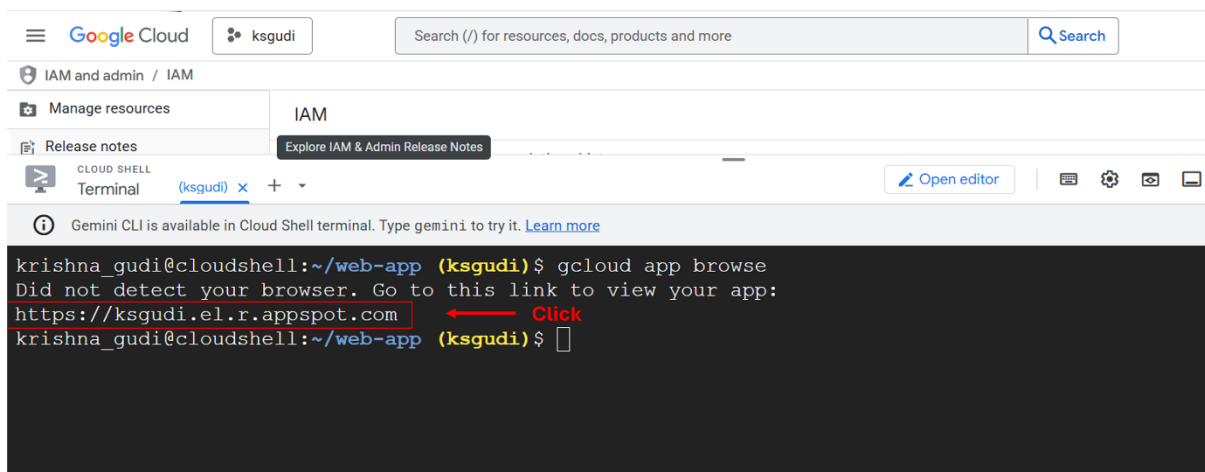
Do you want to continue (Y/n)? y

Beginning deployment of service [default]...
Uploading 0 files to Google Cloud Storage
100%
File upload done.
Waiting for operation [apps/ksgudi/operations/3c55af95-f3d3-43e1-b9c8-2e3f964a3a28] to complete...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://ksgudi.el.r.appspot.com]

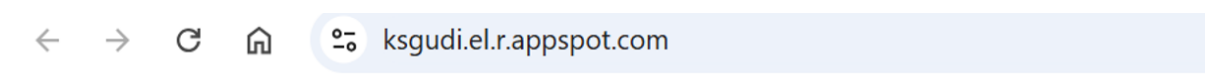
You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse
krishna_gudi@cloudshell:~/web-app (ksgudi) $
```

Step 9: Open the deployed application in the browser to view the web page output.



```
krishna_gudi@cloudshell:~/web-app (ksgudi) $ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://ksgudi.el.r.appspot.com
krishna_gudi@cloudshell:~/web-app (ksgudi) $
```



```
← → ↻ 🏠 🔍 ksgudi.el.r.appspot.com
```

Welcome to Cloud Computing Lab

Current Time: 15-03-2026 09:13:37

Visitor Count: 11

Visit: www.krishnagudi.com

Step 10: Verify the running application instance in the App Engine Instances dashboard.

- Type “App Engine” in the search bar.
- Click on the Navigation menu on the left and select “Instances”.
- View the running auto-scaled instance details.

