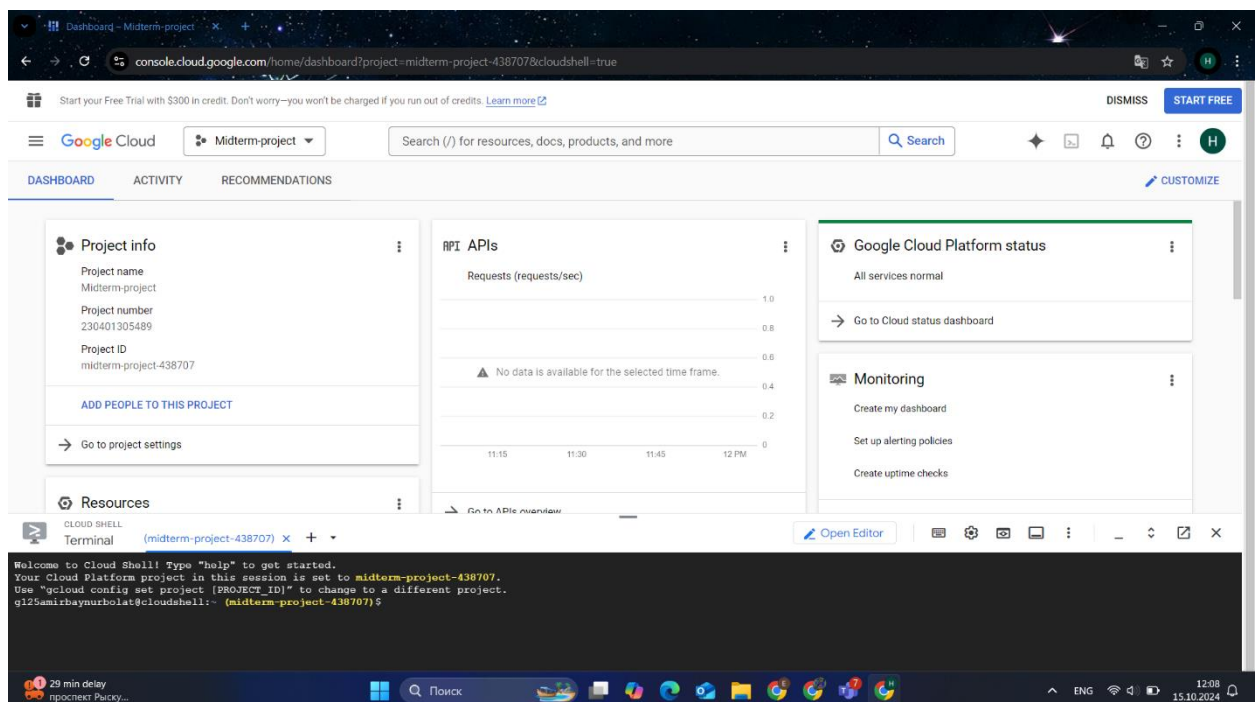


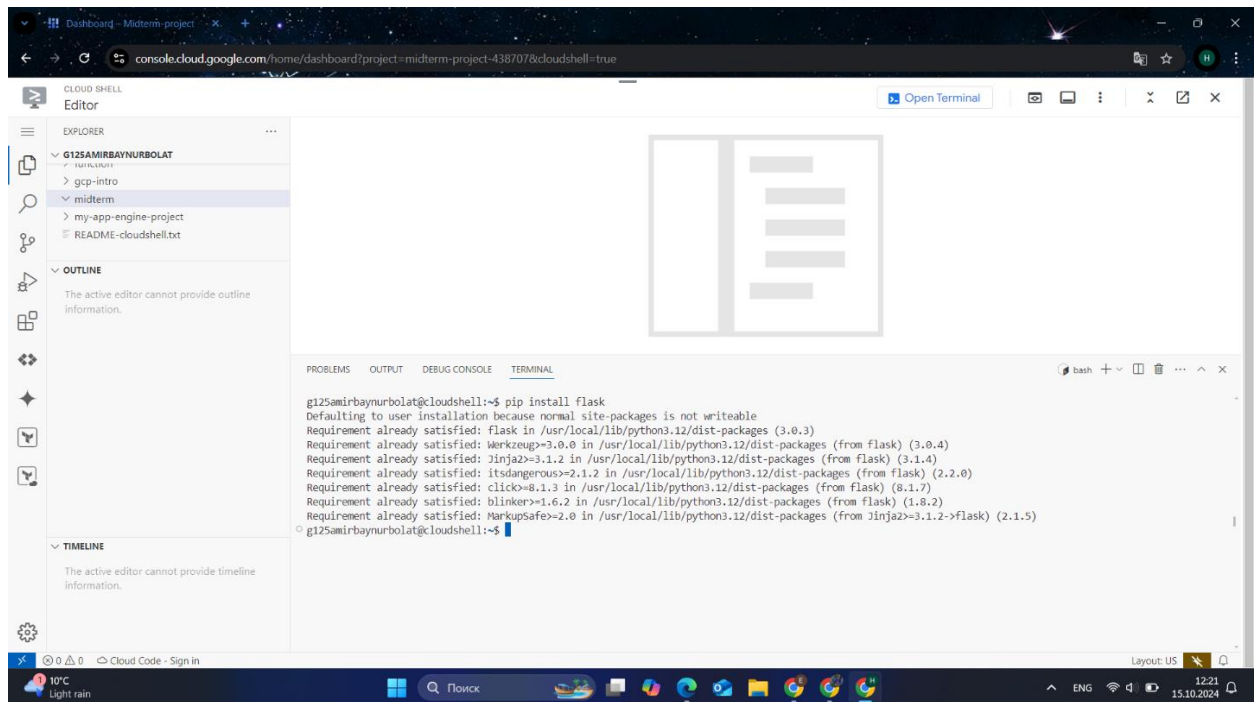
Midterm Report

This project focuses on developing and deploying a scalable web application on Google Cloud Platform (GCP). Key GCP services, including Google App Engine, Google Cloud Functions, and Google Kubernetes Engine (GKE), were leveraged to create a containerized and serverless architecture. The project also incorporated Google Cloud Endpoints for API management and used GCP's monitoring tools to ensure application reliability. The final application was successfully deployed, meeting scalability and performance requirements.

Cloud Shell Usage: Cloud Shell was used to execute deployment commands, interact with Kubernetes, and manage cloud functions. The Cloud Shell environment is pre-configured, making it an ideal tool for managing GCP projects directly in the browser.



Installing Flask for the project:



The screenshot shows a Google Cloud Shell session. The left sidebar contains an Explorer panel with a file tree for a project named 'G125AMIRBAYNURBOLAT'. The tree includes folders 'gcp-intro' and 'midterm', and files 'my-app-engine-project' and 'README-cloudshell.txt'. Below the Explorer is an Outline panel, and at the bottom is a Timeline panel. The main editor area is currently blank. At the bottom of the interface is a terminal window with the following output:

```
g125amirbaynurbolat@cloudshell:~$ pip install flask
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: flask in /usr/local/lib/python3.12/dist-packages (3.0.3)
Requirement already satisfied: Werkzeug>=3.0.0 in /usr/local/lib/python3.12/dist-packages (from flask) (3.0.4)
Requirement already satisfied: Jinja2>=3.1.2 in /usr/local/lib/python3.12/dist-packages (from flask) (3.1.4)
Requirement already satisfied: itsdangerous>=2.1.2 in /usr/local/lib/python3.12/dist-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.12/dist-packages (from flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in /usr/local/lib/python3.12/dist-packages (from flask) (1.8.2)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from Jinja2>=3.1.2->flask) (2.1.5)
g125amirbaynurbolat@cloudshell:~$
```

The bottom status bar shows the system temperature as 10°C with light rain, the search bar with 'Поиск', and the date and time as 12:21 on 15.10.2024.

Creating app.py and testing:

Dashboard - Midterm-project

console.cloud.google.com/home/dashboard?project=midterm-project-438707&cloudshell=true

CLOUD SHELL Editor

EXPLORER

- G12SAMIRBAYNURBOLAT
 - function
 - gcp-intro
 - midterm
 - app.py
 - my-app-engine-project

OUTLINE

- Flask class Flask
- jsonify def jsonify
- request instance request
- app app = Flask(__name__)
- tasks tasks = []
- get_tasks def get_tasks
- add_task def add_task

app.py

```
1 from flask import Flask, request, jsonify
2
3 app = Flask(__name__)
4
5 tasks = []
6
7 @app.route('/tasks', methods=['GET'])
8 def get_tasks():
9     return jsonify(tasks)
10
11 @app.route('/tasks', methods=['POST'])
12 def add_task():
13     task = request.json['task']
14     tasks.append({'task': task})
15     return jsonify({'message': 'Task added successfully!'}), 201
16
17 if __name__ == '__main__':
18     app.run(debug=True)
```

python: can't open file '/home/g12samirbaynurbolat/app.py': [Errno 2] No such file or directory

g12samirbaynurbolat@cloudshell:~\$ cd midterm

g12samirbaynurbolat@cloudshell:~/midterm\$ python app.py

* Serving Flask app 'app'

* Debug mode: on

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

* Running on http://127.0.0.1:5000

Press CTRL+C to quit

* Restarting with stat

* Debugger is active!

* Debugger PIN: 541-535-548

Dashboard - Midterm-project

5000-cs-33612400213-default

console.cloud.google.com/home/dashboard?project=midterm-project-438707&cloudshell=true

CLOUD SHELL Editor

EXPLORER

- G12SAMIRBAYNURBOLAT
 - function
 - gcp-intro
 - midterm
 - app.py
 - my-app-engine-project

OUTLINE

- Flask class Flask
- jsonify def jsonify
- request instance request
- app app = Flask(__name__)
- tasks tasks = []
- get_tasks def get_tasks
- add_task def add_task

app.py

```
1 from flask import Flask, request, jsonify
2
3 app = Flask(__name__)
4
5 tasks = []
6
7 @app.route('/tasks', methods=['GET'])
8 def get_tasks():
9     return jsonify(tasks)
10
11 @app.route('/tasks', methods=['POST'])
12 def add_task():
13     task = request.json['task']
14     tasks.append({'task': task})
15     return jsonify({'message': 'Task added successfully!'}), 201
16
17 if __name__ == '__main__':
18     app.run(debug=True)
```

* Debugger PIN: 541-535-548

127.0.0.1 - - [15/Oct/2024 07:25:36] "GET / HTTP/1.1" 200 -

https://5000-cs-33612400213-default.cs-asia-east1-jnrc.cloudshell.dev/task127.0.0.1 - - [15/Oct/2024 07:26:43] "GET /task HTTP/1.1" 404 -

* Detected change in '/home/g12samirbaynurbolat/midterm/app.py', reloading

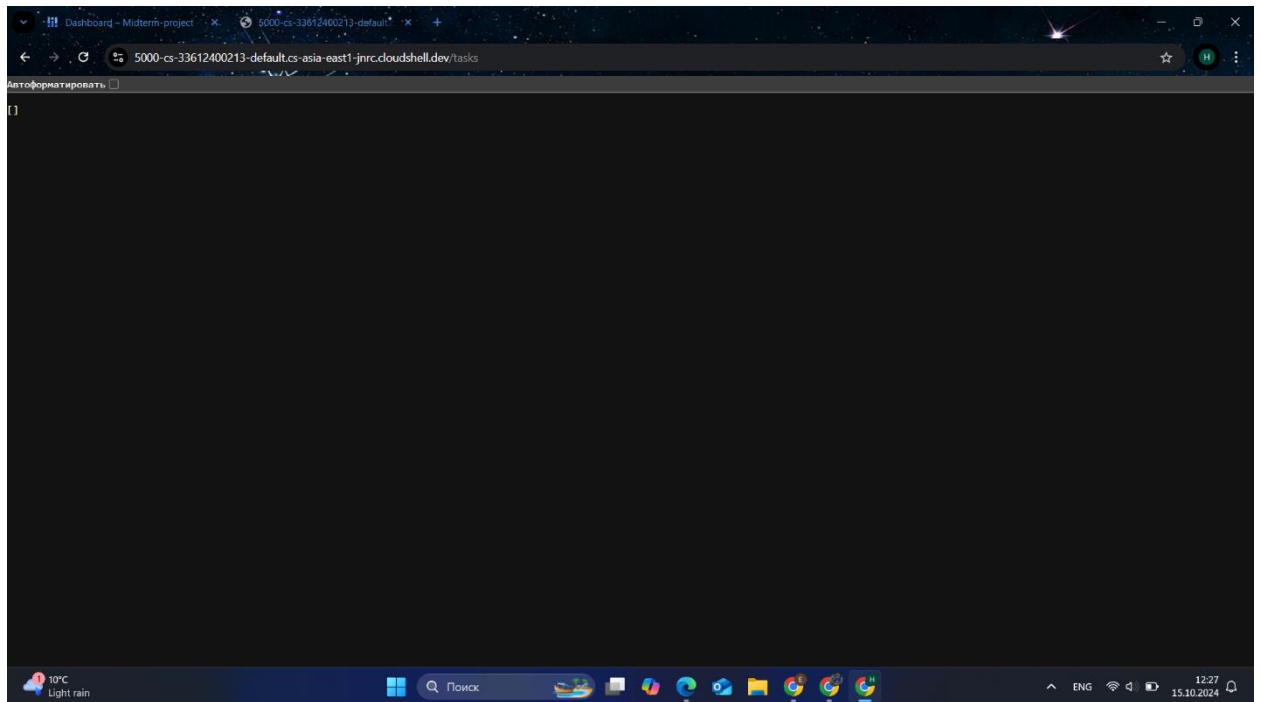
* Restarting with stat

* Debugger is active!

* Debugger PIN: 541-535-548

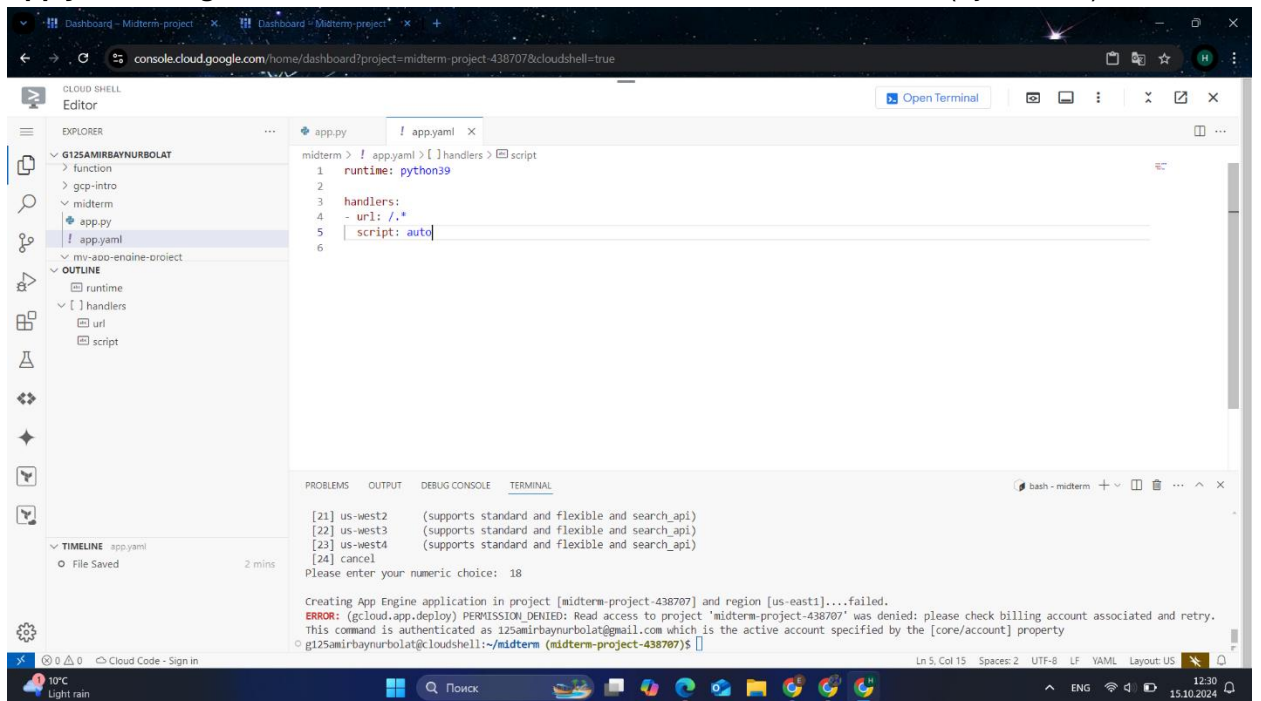
127.0.0.1 - - [15/Oct/2024 07:26:53] "GET /tasks HTTP/1.1" 200 -

127.0.0.1 - - [15/Oct/2024 07:27:00] "GET /tasks HTTP/1.1" 200 -



The web application was developed using Flask, with routes defined for adding and retrieving tasks. The app was tested locally and then deployed on App Engine using the following steps:

app.yaml configuration file was created to define the runtime environment (Python 3.9).



```
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)$ gcloud app deploy
You are creating an app for project [midterm-project-438707].
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>.

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)
[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: 18

Creating App Engine application in project [midterm-project-438707] and region [us-east1]....failed.
```

```
cannot be changed. More information about regions is at
<https://cloud.google.com/appengine/docs/locations>.

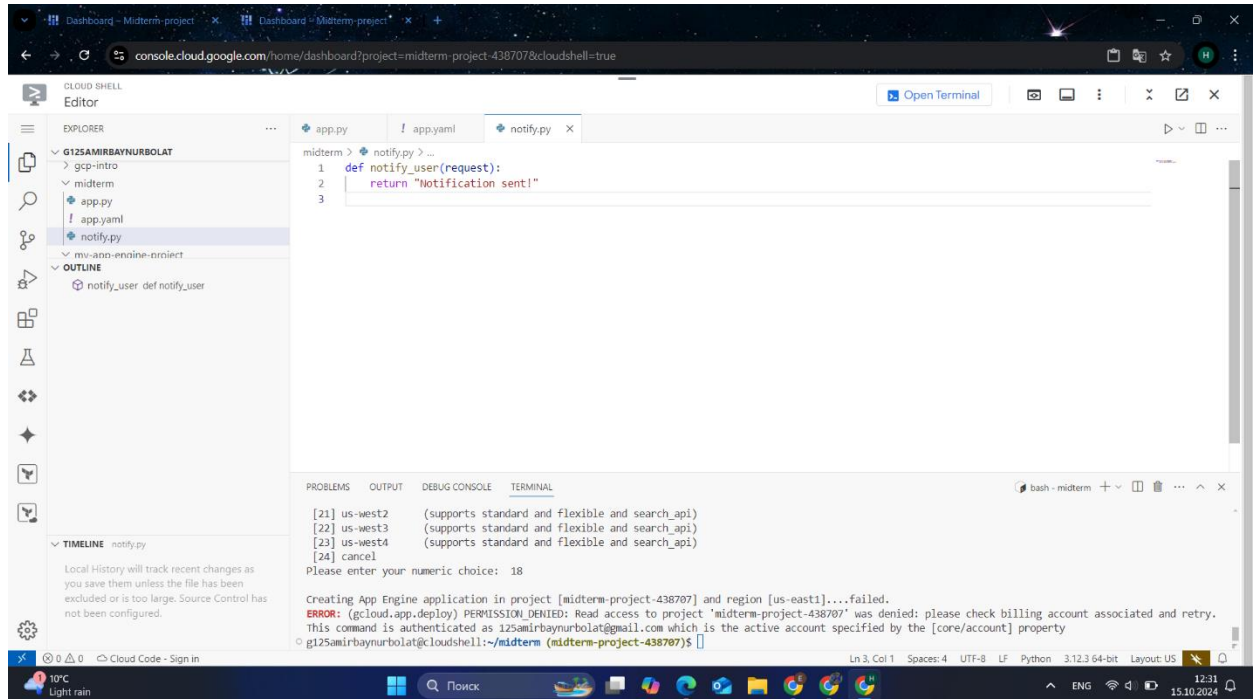
Please choose the region where you want your App Engine application located:

[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)
[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: 18

Creating App Engine application in project [midterm-project-438707] and region [us-east1]....failed.
ERROR: (gcloud.app.deploy) PERMISSION_DENIED: Read access to project "midterm-project-438707" was denied: please check billing account associated and retry.
This command is authenticated as g125amirbaynurbolat@gmail.com which is the active account specified by the [core/account] property
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)$
```


After deployment, the app was accessible via a public URL, and it successfully handled task creation and retrieval under varying loads. But, I haven't billing account.

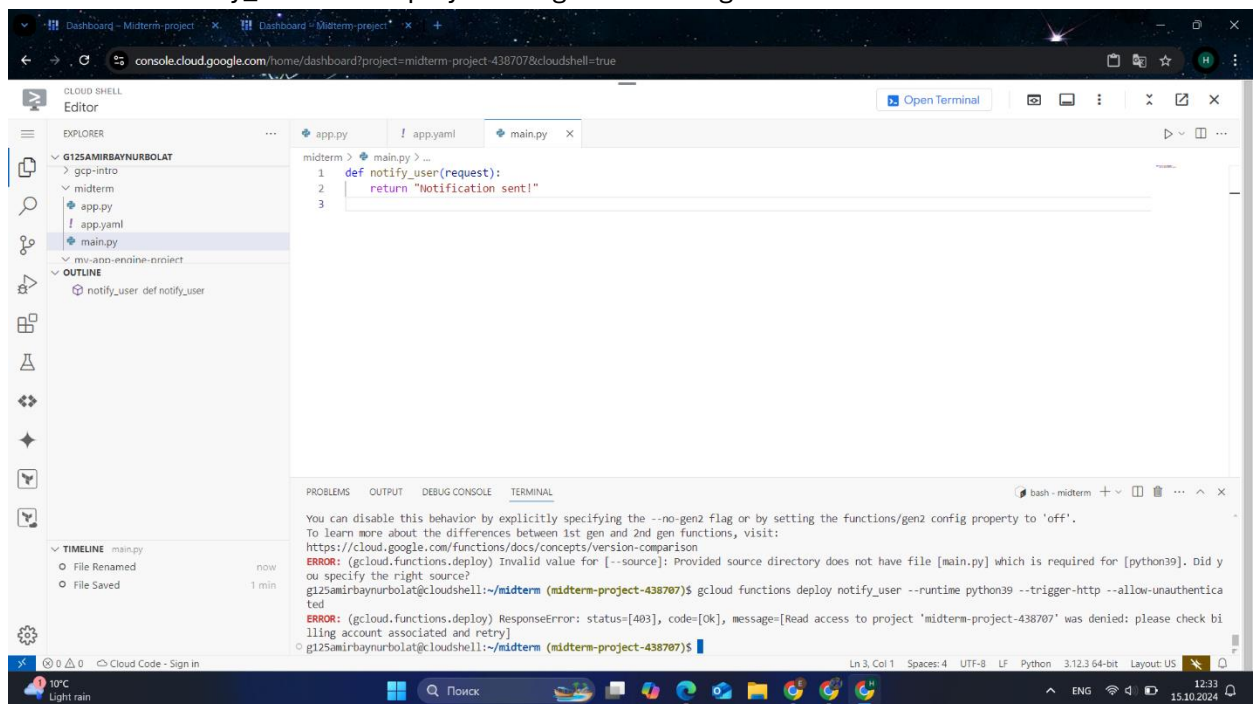


The screenshot shows the Google Cloud Shell Editor interface. The Explorer on the left shows a project named 'G125AMIRBAYNURBOLAT' with files 'app.py', 'app.yaml', and 'notify.py'. The 'notify.py' file is selected, and its code is visible in the editor:

```
1 def notify_user(request):  
2     return "Notification sent!"  
3
```

 The Terminal at the bottom shows the command `gcloud app deploy` being executed, which failed with the error: `ERROR: (gcloud.app.deploy) PERMISSION_DENIED: Read access to project 'midterm-project-438707' was denied: please check billing account associated and retry.` The terminal also shows a list of regions and a prompt for a numeric choice.

Cloud Functions were used to implement a simple notification feature when a new task was added. The function `notify_user` was deployed using the following command:



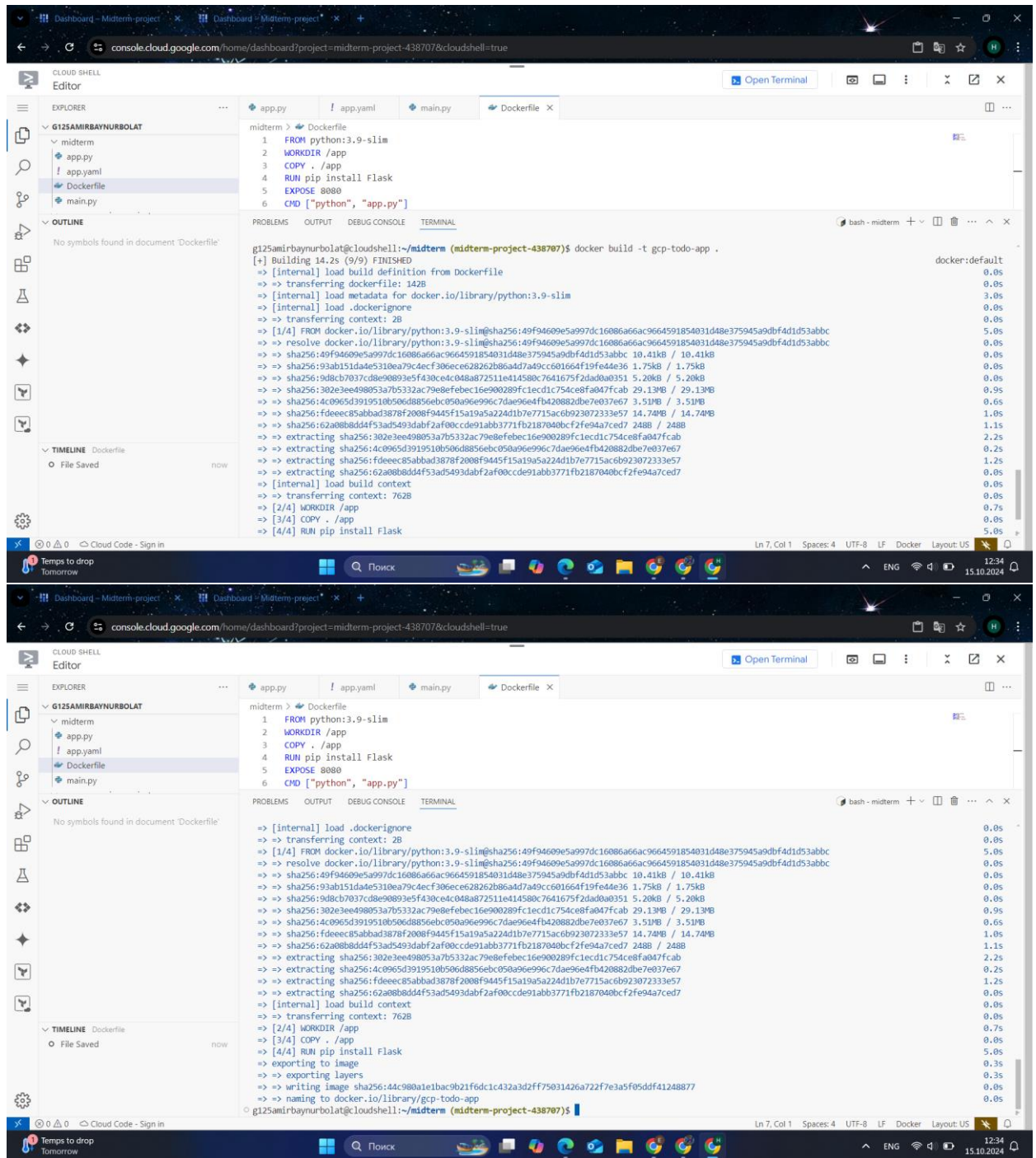
The screenshot shows the Google Cloud Shell Editor interface. The Explorer on the left shows a project named 'G125AMIRBAYNURBOLAT' with files 'app.py', 'app.yaml', and 'main.py'. The 'main.py' file is selected, and its code is visible in the editor:

```
1 def notify_user(request):  
2     return "Notification sent!"  
3
```

 The Terminal at the bottom shows the command `gcloud functions deploy notify_user --runtime python39 --trigger http --allow-unauthenticated` being executed, which failed with the error: `ERROR: (gcloud.functions.deploy) Invalid value for [--source]: Provided source directory does not have file [main.py] which is required for [python39]. Did you specify the right source?` The terminal also shows a message about the billing account.

This serverless function can be triggered by an HTTP request and integrates seamlessly with the main Flask application.

Docker was used to containerize the Flask application, which simplifies deployment on various platforms. A **Dockerfile** was created and the image was built and tested locally before pushing it to Google Container Registry:



The image displays two screenshots of the Google Cloud Shell Editor interface, showing the process of building a Docker image for a Flask application.

Top Screenshot: The Dockerfile is visible in the editor, and the terminal shows the output of the `docker build -t gcp-todo-app .` command. The build process is shown as a series of steps, including downloading the Dockerfile, resolving the base image, and copying files into the container.

```
midterm > Dockerfile
1 FROM python:3.9-slim
2 WORKDIR /app
3 COPY . /app
4 RUN pip install Flask
5 EXPOSE 8080
6 CMD ["python", "app.py"]
```

```
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)$ docker build -t gcp-todo-app .
[+] Building 14.2s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 142B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc
=> resolve docker.io/library/python:3.9-slim@sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc
=> sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc 10.41kB / 10.41kB
=> sha256:93ab151da4e5310ea79c4ecf306ee628262b6a4d7a9cc601664f19fe4a36 1.75kB / 1.75kB
=> sha256:9d8cb7037cd8e90893e5f430ce4c048a87251e414580c7641675f2dad0a0351 5.20kB / 5.20kB
=> sha256:302e3ee498053a7b5332ac79e8efebec16e900289f1ecd1c754ce8fa047fcab 29.13kB / 29.13kB
=> sha256:4c0965d3919510b506d8856ebc050a9e996c7dae96e4fb420882db7e037e67 3.51kB / 3.51kB
=> sha256:fdeec85abbad3878f2008f9445f15a19a5a224d1b7e7715ac6b92307233e57 14.74kB / 14.74kB
=> sha256:62a08bd4d53ad5493daf2af0ccde91abb3771fb2187040bcf2fe94a7ced7 248B / 248B
=> extracting sha256:302e3ee498053a7b5332ac79e8efebec16e900289f1ecd1c754ce8fa047fcab
=> extracting sha256:4c0965d3919510b506d8856ebc050a9e996c7dae96e4fb420882db7e037e67
=> extracting sha256:fdeec85abbad3878f2008f9445f15a19a5a224d1b7e7715ac6b92307233e57
=> extracting sha256:62a08bd4d53ad5493daf2af0ccde91abb3771fb2187040bcf2fe94a7ced7
=> [internal] load build context
=> => transferring context: 762B
=> [2/4] WORKDIR /app
=> [3/4] COPY . /app
=> [4/4] RUN pip install Flask
```

Bottom Screenshot: The terminal shows the output of the `docker build -t gcp-todo-app .` command, which is identical to the top screenshot, but includes the final step of naming the image to `docker.io/g125amirbaynurbolat/gcp-todo-app`.

```
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)$
=> naming to docker.io/library/gcp-todo-app
=> exporting layers
=> exporting image sha256:44c980a1e1bac9b21f6dc1c432a3d2f75031426a722f7e3a5f05dd41248877
=> exporting to image
=> exporting layers
=> exporting image sha256:44c980a1e1bac9b21f6dc1c432a3d2f75031426a722f7e3a5f05dd41248877
=> naming to docker.io/library/gcp-todo-app
```


Dashboard - Midterm-project x 5000-cs-33612400213-default x Dashboard - Midterm-project x

console.cloud.google.com/home/dashboard?project=midterm-project-438707&cloudshell=true

CLOUD SHELL Editor

EXPLORER

- G12SAMIRBAYNURBOLAT
 - midterm
 - app.py
 - app.yaml
 - Dockerfile
 - main.py
 - my-aco-enaine-project
 - Flask class Flask
 - jsonify def jsonify
 - request instance request
 - app app = Flask(__name__)
 - tasks tasks = []
 - get_tasks def get_tasks
 - add_task def add_task

OUTLINE

- Flask class Flask
- jsonify def jsonify
- request instance request
- app app = Flask(__name__)
- tasks tasks = []
- get_tasks def get_tasks
- add_task def add_task

app.py

```
9 return jsonify(tasks)
10
11 @app.route('/tasks', methods=['POST'])
12 def add_task():
13     task = request.json['task']
14     tasks.append({'task': task})
15     return jsonify({'message': 'Task added successfully!'}), 201
16
17 if __name__ == '__main__':
18     app.run(host='0.0.0.0', port=5000)
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

docker - midterm

```
=> [3/4] COPY . /app
=> [4/4] RUN pip install flask
=> exporting image
=> writing image sha256:c7c7f769cec056b3a7d67c2478db4f3605c887fff35a1b76a174a7cc23f460dd
=> naming to docker.io/library/gcp-todo-app
g12samirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)$ docker run -p 5000:5000 gcp-todo-app
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://172.17.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
172.17.0.1 - - [15/Oct/2024 08:05:29] "GET / HTTP/1.1" 404 -
172.17.0.1 - - [15/Oct/2024 08:05:32] "GET /tasks HTTP/1.1" 200 -
```

Ln 17, Col 27 Spaces: 4 UTF-8 LF Python 3.12.3 64-bit Layout: US

10°C Light rain

1306 15.10.2024

Dashboard - Midterm-project x 5000-cs-33612400213-default x Dashboard - Midterm-project x

5000-cs-33612400213-default.cs-asia-east1-jnrc.cloudshell.dev/tasks

Автоформатировать

[]

10°C Light rain

1306 15.10.2024

Dashboard - Midterm-project

Kubernetes Engine API - APIs

5000-cs-3361400213-default

Dashboard - Midterm-project

+


console.cloud.google.com/apis/library/container.googleapis.com?project=midterm-project-438707&flow=gcp

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Dismiss [Start free](#)

Google Cloud Midterm-project

Product details



Kubernetes Engine API

[Google Enterprise API](#)

Builds and manages container-based applications, powered by the open source Kubernetes technology.

[TRY THIS API](#)

[OVERVIEW](#) [DOCUMENTATION](#) [RELATED PRODUCTS](#)

Overview

Builds and manages container-based applications, powered by the open source Kubernetes technology.

Additional details

Type: [SaaS & APIs](#)
Last product update: 8/13/24
Category: [Compute](#), [Google Enterprise APIs](#)
Service name: container.googleapis.com

10°C
Light rain

Поиск

ENG 13:08 15.10.2024

Dashboard - Midterm-project

Artifact Registry API - APIs &

5000-cs-3361400213-default

Dashboard - Midterm-project

+


console.cloud.google.com/apis/library/artifactregistry.googleapis.com?project=midterm-project-438707&flow=gcp

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Dismiss [Start free](#)

Google Cloud Midterm-project

Product details



Artifact Registry API

[Google Enterprise API](#)

With Artifact Registry you can store and manage your build artifacts (e.g. Docker images, Maven packages, npm packages), in a scalable and integrated repository service built on Google infrastructure. You can manage repository access with IAM and interact with repositories via gcloud, Cloud Console, and native package format tools. The service can also be integrated with Cloud Build and other CI/CD systems. Artifact Registry abstracts away infrastructure management, so you can focus on what matters most — delivering value to the users of your services and applications. Note: Enabling the Artifact Registry API will not affect your use of Container Registry in the same project.

[Learn more](#)

[TRY THIS API](#)

[OVERVIEW](#) [PRICING](#) [DOCUMENTATION](#) [RELATED PRODUCT](#)

Overview

With Artifact Registry you can store and manage your build artifacts (e.g. Docker images, Maven packages, npm packages), in a scalable and integrated repository service built on Google infrastructure. You can manage repository access with IAM and interact with repositories via gcloud, Cloud Console, and native package format tools. The service can also be integrated with Cloud Build and other CI/CD systems. Artifact Registry abstracts away infrastructure management, so you can focus on what matters most — delivering value to the users of your services and applications. Note: Enabling the Artifact Registry API will not affect your use of Container Registry in the same project.

Additional details

Type: [SaaS & APIs](#)
Last product update: 4/30/22
Category: [DevOps](#), [Google Enterprise APIs](#)
Service name: artifactregistry.googleapis.com

17 min delay
Saint Street

Поиск

ENG 13:10 15.10.2024

Dashboard - Midterm-project x 5000-cs-33612400213-default x Dashboard - Midterm-project x +

console.cloud.google.com/home/dashboard?project=midterm-project-438707&cloudshell=true

CLOUD SHELL Editor

EXPLORER

- G125AMIRBAYNURBOLAT
 - midterm
 - app.py
 - app.yaml
 - Dockerfile
 - main.py
 - my-aaa-enaine-proiect
 - Flask class Flask
 - jsonify def jsonify
 - request instance request
 - app app = Flask(__name__)
 - tasks tasks = []
 - get_tasks def get_tasks
 - add_task def add_task
- TIMELINE
 - File Saved
 - File Saved
 - Undo / Redo
 - File Saved
 - Undo / Redo

midterm > app.py > ...

```
9 return jsonify(tasks)
10
11 @app.route('/tasks', methods=['POST'])
12 def add_task():
13     task = request.json['task']
14     tasks.append({'task': task})
15     return jsonify({'message': 'Task added successfully!'}), 201
16
17 if __name__ == '__main__':
18     app.run(host='0.0.0.0', port=5000)
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

bash - midterm

70a7c110be69
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)\$ gcloud container clusters create todo-cluster --zone us-central1-a
Note: The Kubelet readonly port (10255) is now deprecated. Please update your workloads to use the recommended alternatives. See https://cloud.google.com/kub
ernetes-engine/docs/how-to/disable-kubelet-readonly-port for ways to check usage and for migration instructions.
Note: Your Pod address range ("--cluster-ip-v4-cidr") can accommodate at most 1008 node(s).
ERROR: (gcloud.container.clusters.create) ResponseError: code=403, message=Kubernetes Engine API has not been used in project midterm-project-438707 before o
r it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/container.googleapis.com/overview/project=midterm-project-438707 then
retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry. This command is authenticated as 125amirbay
nurbolat@gmail.com which is the active account specified by the [core/account] property.
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)\$ docker tag gcp-todo-app gcr.io/midterm-project-438707/gcp-todo-app
Using default tag: latest
The push refers to repository [gcr.io/midterm-project-438707/gcp-todo-app]
fac6a34290e4: Preparing
497f77cdf90f: Preparing
0643409f113: Preparing
9e599118e168: Preparing

Dashboard - Midterm-project x 5000-cs-33612400213-default x Dashboard - Midterm-project x +

console.cloud.google.com/home/dashboard?project=midterm-project-438707&cloudshell=true

CLOUD SHELL Editor

EXPLORER

- G125AMIRBAYNURBOLAT
 - midterm
 - app.py
 - app.yaml
 - Dockerfile
 - main.py
 - my-aaa-enaine-proiect
 - Flask class Flask
 - jsonify def jsonify
 - request instance request
 - app app = Flask(__name__)
 - tasks tasks = []
 - get_tasks def get_tasks
 - add_task def add_task
- TIMELINE
 - File Saved
 - File Saved
 - Undo / Redo
 - File Saved
 - Undo / Redo

midterm > app.py > ...

```
9 return jsonify(tasks)
10
11 @app.route('/tasks', methods=['POST'])
12 def add_task():
13     task = request.json['task']
14     tasks.append({'task': task})
15     return jsonify({'message': 'Task added successfully!'}), 201
16
17 if __name__ == '__main__':
18     app.run(host='0.0.0.0', port=5000)
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

bash - midterm

retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry. This command is authenticated as 125amirbay
nurbolat@gmail.com which is the active account specified by the [core/account] property.
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)\$ docker tag gcp-todo-app gcr.io/midterm-project-438707/gcp-todo-app
Using default tag: latest
The push refers to repository [gcr.io/midterm-project-438707/gcp-todo-app]
fac6a34290e4: Preparing
497f77cdf90f: Preparing
0643409f113: Preparing
9e599118e168: Preparing
e228ad1886f: Preparing
fb5ccdb472: Waiting
8d053c8add5d: Waiting
denied: Artifact Registry API has not been used in project 230401305489 before or it is disabled. Enable it by visiting https://console.developers.google.com
/apis/api/artifactregistry.googleapis.com/overview?project=230401305489 then retry. If you enabled this API recently, wait a few minutes for the action to pr
opagate to our systems and retry.
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)\$

APIs were defined using the OpenAPI specification. The API was secured using Google Cloud Endpoints to ensure only authenticated requests were processed. The API specification file `openapi.yaml` was created to define the `/tasks` endpoint.

The first screenshot shows the Google Cloud Shell Editor interface. The Explorer pane on the left displays the project structure for `g125amirbaynurbolat`, including a `midterm` directory with `app.py`, `app.yaml`, and `deployment.yaml`. The `deployment.yaml` file is selected and its content is shown in the main editor:

```
midterm > ! deployment.yaml > {} spec > {} template > {} spec > [] containers > image
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: todo-app
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: todo-app
10   template:
11     metadata:
12       labels:
13         app: todo-app
14     spec:
15       containers:
16         - name: todo-app
17           image: gcr.io/midterm-project-438707/gcp-todo-app
18           ports:
19             - containerPort: 5000
```

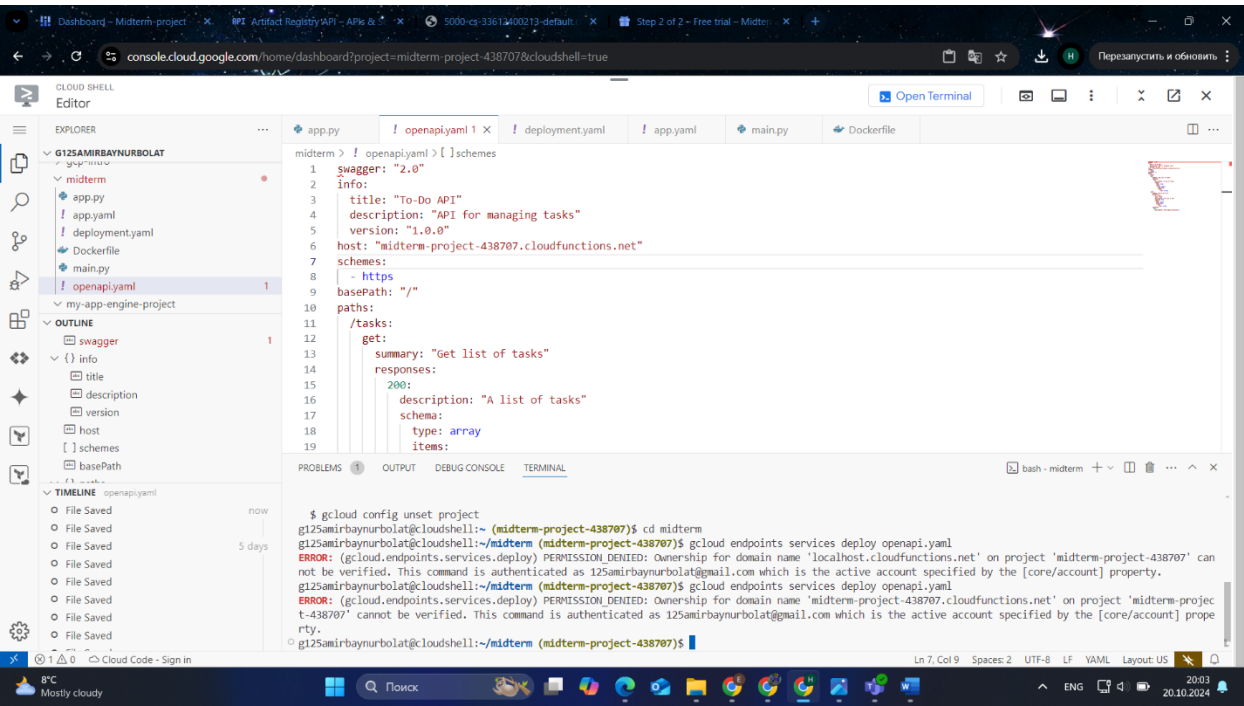
The Terminal pane at the bottom shows the command `kubectl apply -f deployment.yaml` being executed. The output indicates that the Artifact Registry API has not been used in the project before and is disabled. The user is prompted to enable it by visiting <https://console.developers.google.com/apis/api/artifactregistry.googleapis.com/overview?project=230401305489> and then retrying. The error message is: `error: error validating "deployment.yaml": error validating data: failed to download openapi: Get "http://localhost:8080/openapi/v2?timeout=32s": dial tcp 127.0.0.1:8080: connect: connection refused; if you choose to ignore these errors, turn validation off with --validate=false`.

The second screenshot shows the same Google Cloud Shell Editor interface. The Explorer pane now shows the `app.py` file selected. The main editor displays the content of `app.py`:

```
midterm > app.py > ...
9  return jsonify(tasks)
10
11 @app.route('/tasks', methods=['POST'])
12 def add_task():
13     task = request.json['task']
14     tasks.append({'task': task})
15     return jsonify({'message': 'Task added successfully!'}), 201
16
17 if __name__ == '__main__':
18     app.run(host='0.0.0.0', port=5000)
19
```

The Terminal pane shows the command `kubectl expose deployment todo-app --type=LoadBalancer --port 50 --target-port 5000` being executed. The output shows the service being created and the port being exposed. The user is then prompted to `g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)$ kubectl apply -f deployment.yaml`.

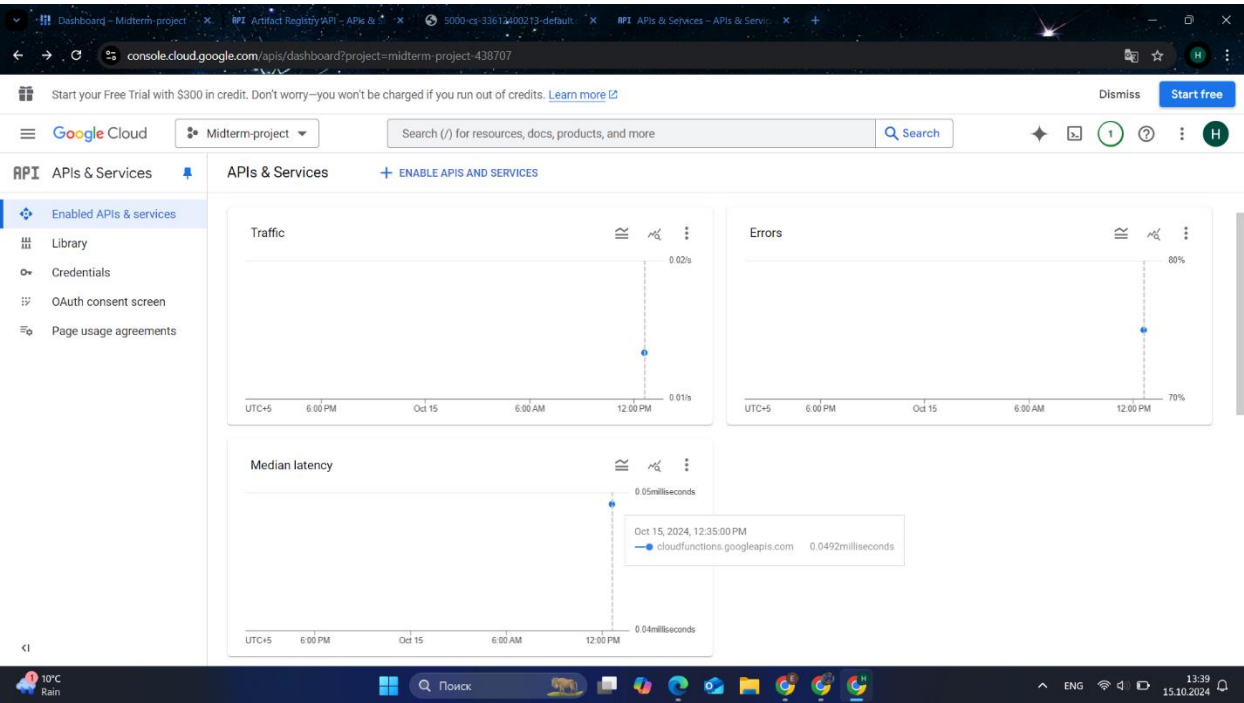
The service was deployed using:
gcloud endpoints services deploy openapi.yaml



```
midterm > ! openapiyaml > [ ] schemes
1  swagger: "2.0"
2  info:
3    title: "To-Do API"
4    description: "API for managing tasks"
5    version: "1.0.0"
6    host: "midterm-project-438707.cloudfunctions.net"
7  schemes:
8    - https
9  basePath: "/"
10 paths:
11   /tasks:
12     get:
13       summary: "Get list of tasks"
14       responses:
15         200:
16           description: "A list of tasks"
17           schema:
18             type: array
19             items:
```

```
$ gcloud config unset project
g125amirbaynurbolat@cloudshell:~ (midterm-project-438707)$ cd midterm
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)$ gcloud endpoints services deploy openapi.yaml
ERROR: (gcloud.endpoints.services.deploy) PERMISSION_DENIED: Ownership for domain name 'localhost.cloudfunctions.net' on project 'midterm-project-438707' can not be verified. This command is authenticated as 125amirbaynurbolat@gmail.com which is the active account specified by the [core/account] property.
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)$ gcloud endpoints services deploy openapi.yaml
ERROR: (gcloud.endpoints.services.deploy) PERMISSION_DENIED: Ownership for domain name 'midterm-project-438707.cloudfunctions.net' on project 'midterm-project-438707' cannot be verified. This command is authenticated as 125amirbaynurbolat@gmail.com which is the active account specified by the [core/account] property.
g125amirbaynurbolat@cloudshell:~/midterm (midterm-project-438707)$
```

Google Cloud Monitoring tools were set up to track the application's performance, including metrics like CPU usage, memory, and response time. Alerts were configured to notify administrators when certain thresholds (e.g., high CPU usage) were exceeded.



This project successfully demonstrated how to build and deploy a scalable web application using Google Cloud Platform. The integration of Google App Engine, Cloud Functions, Kubernetes, and Cloud Endpoints allowed the application to scale effectively while maintaining high performance. GCP's tools for monitoring and security were instrumental in maintaining uptime and reliability.