Lab 2 GCP AppEngine

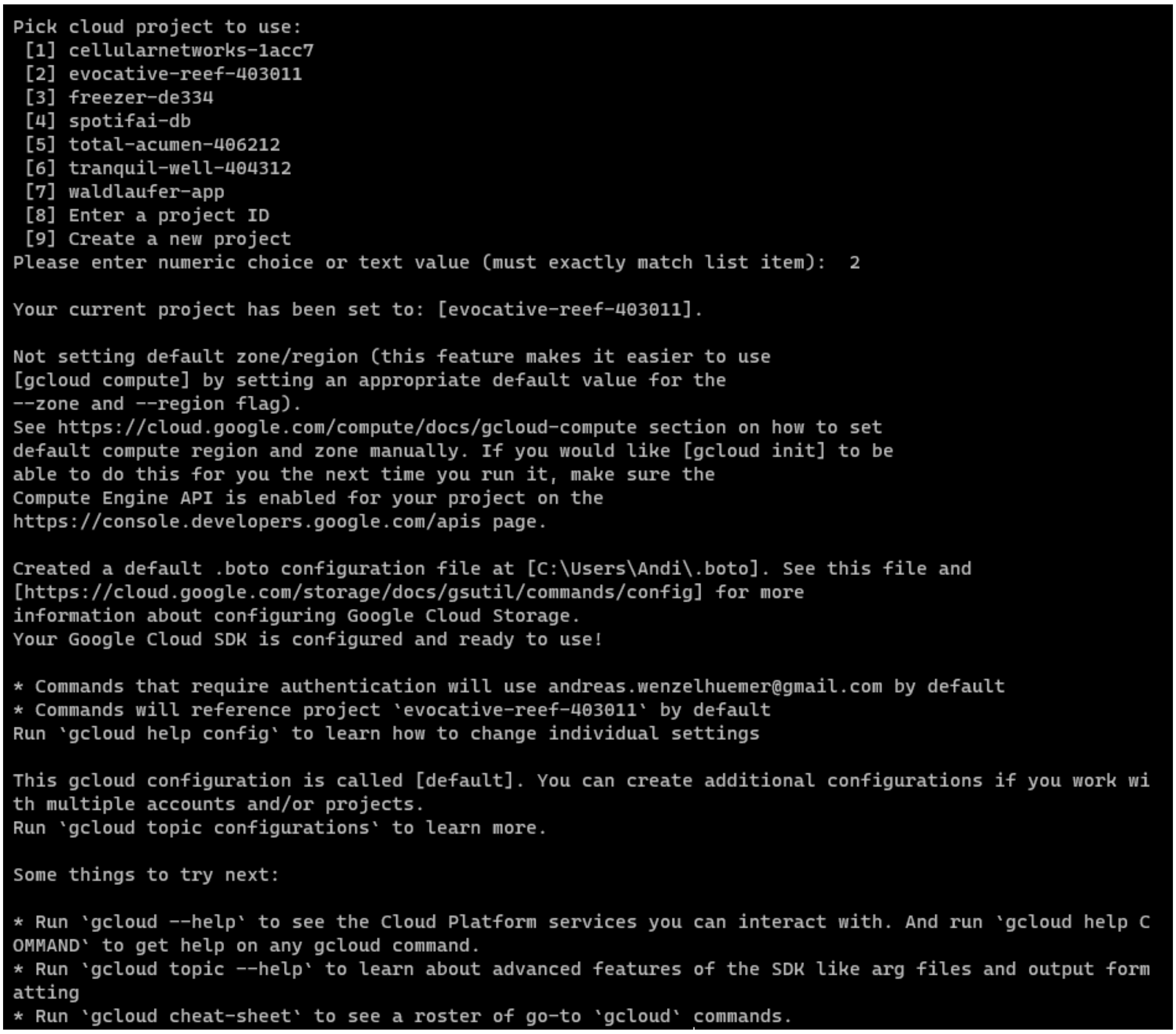
Table of Contents

1. GCP CLI

Installed via following command for windows:

```
(New-Object Net.WebClient).DownloadFile
("https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe", "
$env:Temp\GoogleCloudSDKInstaller.exe")

& $env:Temp\GoogleCloudSDKInstaller.exe
```

Init project with `gcloud init`

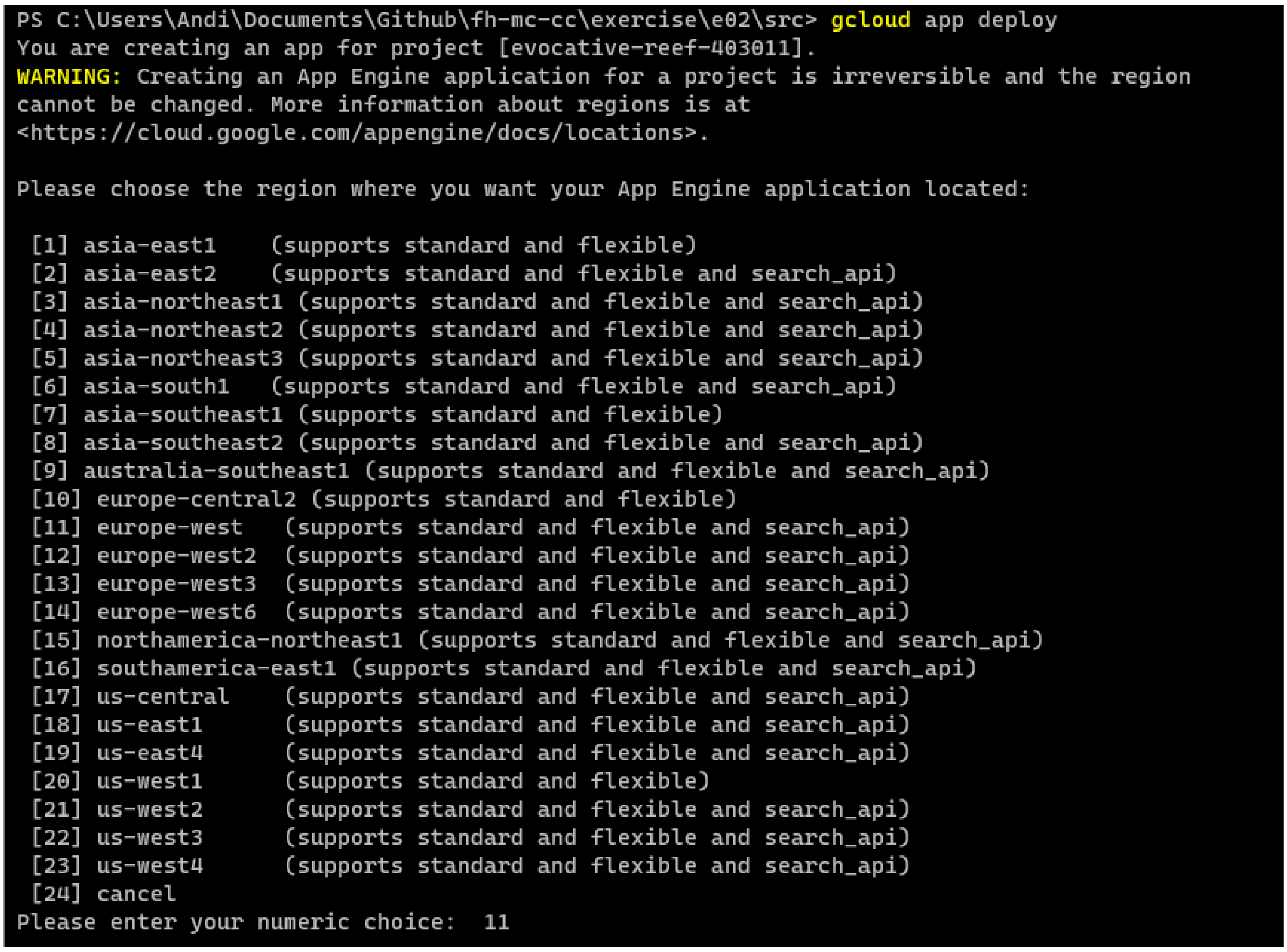


```
Pick cloud project to use:
 [1] cellularnetworks-1acc7
 [2] evocative-reef-403011
 [3] freezer-de334
 [4] spotifai-db
 [5] total-acumen-406212
 [6] tranquil-well-404312
 [7] waldlaufer-app
 [8] Enter a project ID
 [9] Create a new project
Please enter numeric choice or text value (must exactly match list item):  2

Your current project has been set to: [evocative-reef-403011].

Not setting default zone/region (this feature makes it easier to use
[gcloud compute] by setting an appropriate default value for the
--zone and --region flag).
See https://cloud.google.com/compute/docs/gcloud-compute section on how to set
default compute region and zone manually. If you would like [gcloud init] to be
able to do this for you the next time you run it, make sure the
Compute Engine API is enabled for your project on the
https://console.developers.google.com/apis page.

Created a default .boto configuration file at [C:\Users\Andi\.boto]. See this file and
[https://cloud.google.com/storage/docs/gsutil/commands/config] for more
information about configuring Google Cloud Storage.
Your Google Cloud SDK is configured and ready to use!

* Commands that require authentication will use andreas.wenzelhuemer@gmail.com by default
* Commands will reference project `evocative-reef-403011` by default
Run `gcloud help config` to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional configurations if you work wi
th multiple accounts and/or projects.
Run `gcloud topic configurations` to learn more.

Some things to try next:

* Run `gcloud --help` to see the Cloud Platform services you can interact with. And run `gcloud help C
OMMAND` to get help on any gcloud command.
* Run `gcloud topic --help` to learn about advanced features of the SDK like arg files and output form
atting
* Run `gcloud cheat-sheet` to see a roster of go-to `gcloud` commands.
```

Figure 1. Initializing the project

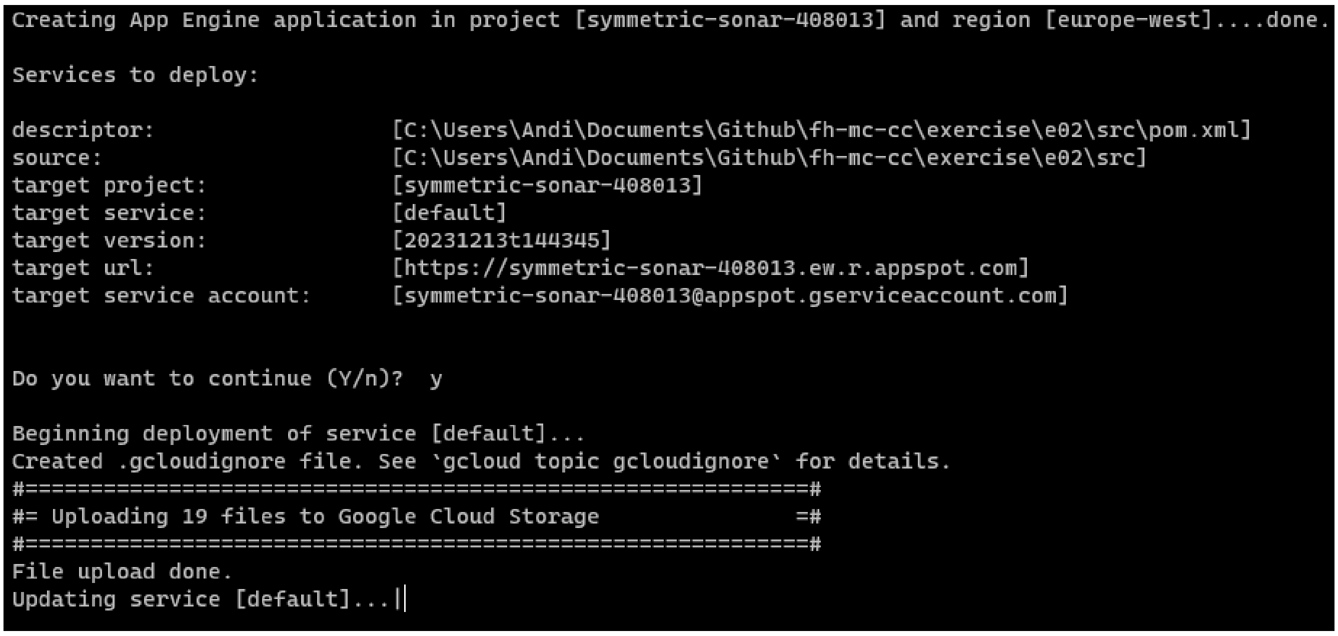
2. "Hello World" App

Deploy app with `gcloud app deploy`.



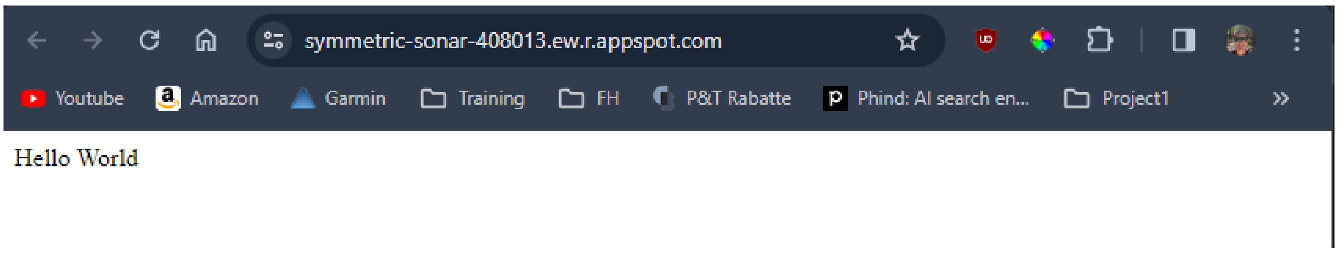
```
PS C:\Users\Andi\Documents\Github\fh-mc-cc\exercise\e02\src> gcloud app deploy
You are creating an app for project [evocative-reef-403011].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. More information about regions is at
<https://cloud.google.com/appengine/docs/locations>.

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:  11
```

Figure 2. Deploy to gcloud project



```
Creating App Engine application in project [symmetric-sonar-408013] and region [europe-west]....done.

Services to deploy:

descriptor:                  [C:\Users\Andi\Documents\Github\fh-mc-cc\exercise\e02\src\pom.xml]
source:                      [C:\Users\Andi\Documents\Github\fh-mc-cc\exercise\e02\src]
target project:              [symmetric-sonar-408013]
target service:              [default]
target version:              [20231213t144345]
target url:                  [https://symmetric-sonar-408013.ew.r.appspot.com]
target service account:      [symmetric-sonar-408013@appspot.gserviceaccount.com]


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

Beginning deployment of service [default]...
Created .gcloudignore file. See `gcloud topic gcloudignore` for details.
#============================================================#
#= Uploading 19 files to Google Cloud Storage               =#
#============================================================#
File upload done.
Updating service [default]...|
```

Figure 3. Upload files to server and update service



Hello World

Figure 4. Running service

3. Create REST Webservice

3.1. Create rest web service

Listing 1. Greeting.java

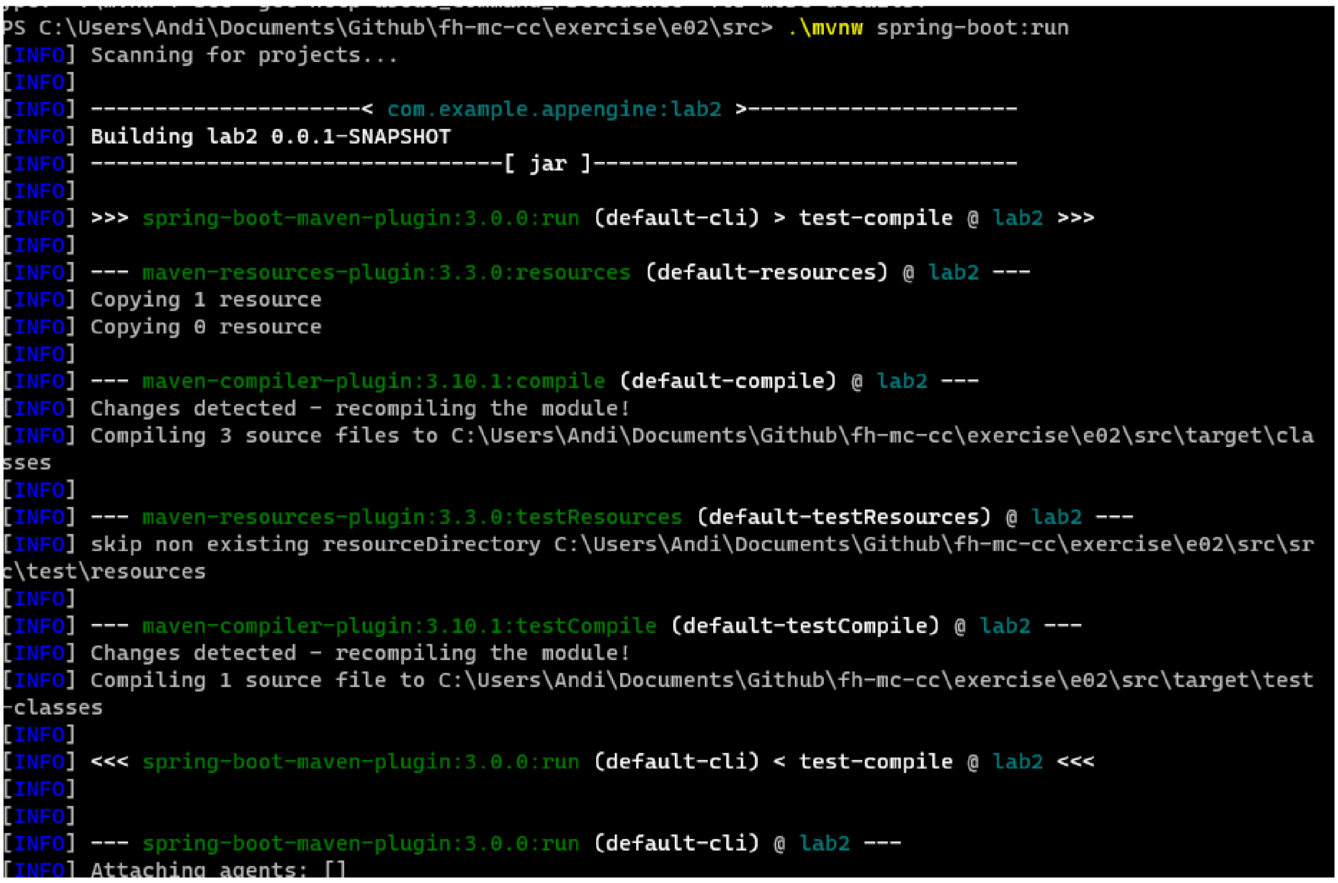
```java
package com.example.appengine.lab2;

public class Greeting {
    private final long id;
    private final String name;

    public Greeting(long id, String name) {
        this.id = id;
        this.name = name;
    }

    public long getId() {
        return id;
    }

    public String getName() {
        return name;
    }
}
```

Listing 2. GreetingController.java

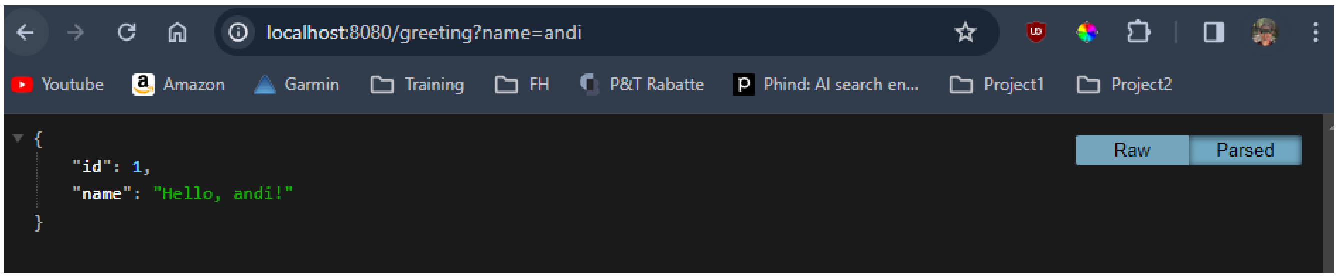
```java
package com.example.appengine.lab2;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class GreetingController {
    private int count = 0;

    @GetMapping("/greeting")
    public Greeting getGreeting(@RequestParam(value = "name", defaultValue = "World")
String name) {
        return new Greeting(count++, "Hello, " + name + "!");
    }
}
```

3.2. Run locally

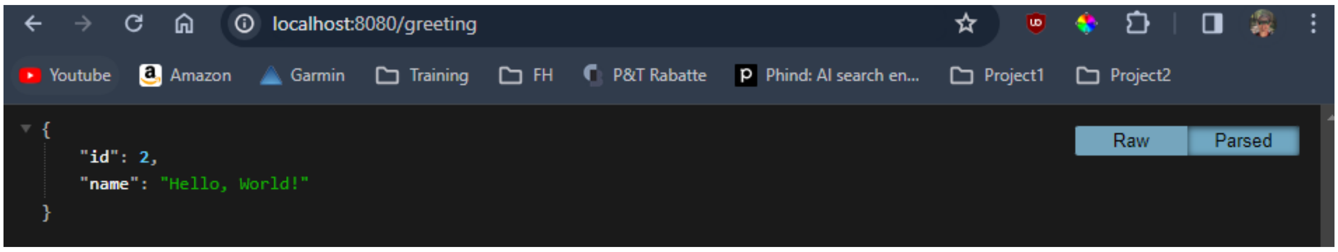
`mvnw spring-boot:run`



```
PS C:\Users\Andi\Documents\Github\fh-mc-cc\exercise\e02\src> .\mvnw spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------< com.example.appengine:lab2 >--------------------
[INFO] Building lab2 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] >>> spring-boot-maven-plugin:3.0.0:run (default-cli) > test-compile @ lab2 >>>
[INFO]
[INFO] --- maven-resources-plugin:3.3.0:resources (default-resources) @ lab2 ---
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ lab2 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 3 source files to C:\Users\Andi\Documents\Github\fh-mc-cc\exercise\e02\src\target\cla
sses
[INFO]
[INFO] --- maven-resources-plugin:3.3.0:testResources (default-testResources) @ lab2 ---
[INFO] skip non existing resourceDirectory C:\Users\Andi\Documents\Github\fh-mc-cc\exercise\e02\src\sr
c\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:testCompile (default-testCompile) @ lab2 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\Andi\Documents\Github\fh-mc-cc\exercise\e02\src\target\test
-classes
[INFO]
[INFO] <<< spring-boot-maven-plugin:3.0.0:run (default-cli) < test-compile @ lab2 <<<
[INFO]
[INFO]
[INFO] --- spring-boot-maven-plugin:3.0.0:run (default-cli) @ lab2 ---
[INFO] Attaching agents: []
```



```
{
    "id": 1,
    "name": "Hello, andi!"
}
```



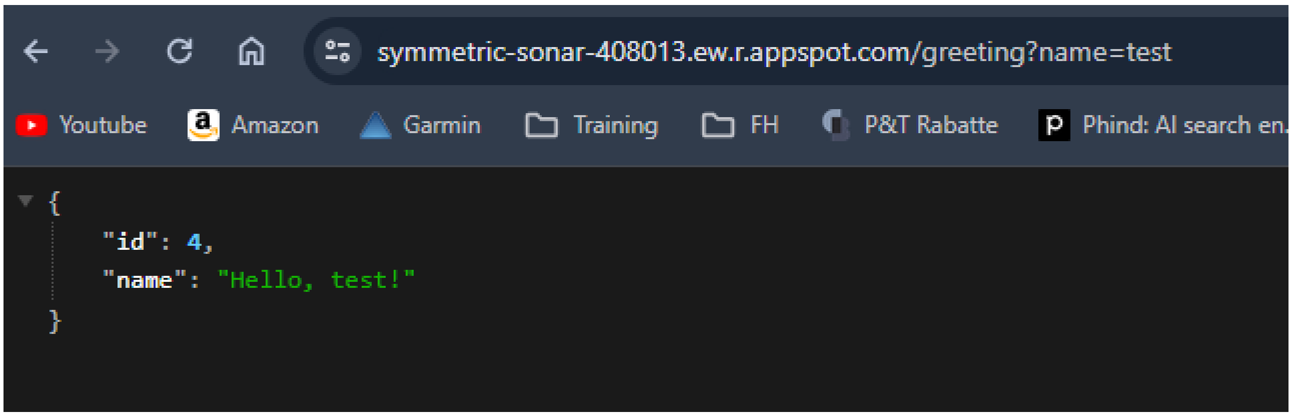
```
{
    "id": 2,
    "name": "Hello, World!"
}
```

3.3. Run in the google cloud

Listing 3. app.yaml

```
runtime: java17
env: standard
service: greeting-service
handlers:
  - url: /.*
```

```
    script: this field is required, but ignored
```



symmetric-sonar-408013.ew.r.appspot.com/greeting?name=test

```
{
    "id": 4,
    "name": "Hello, test!"
}
```

Do you see any potential issues with the function and the returned content?

After redeploying the service, the id gets set back to zero. Also when we create multiple instances of the app, we don't know which id counter gets increased. Normally each instance should be stateless.

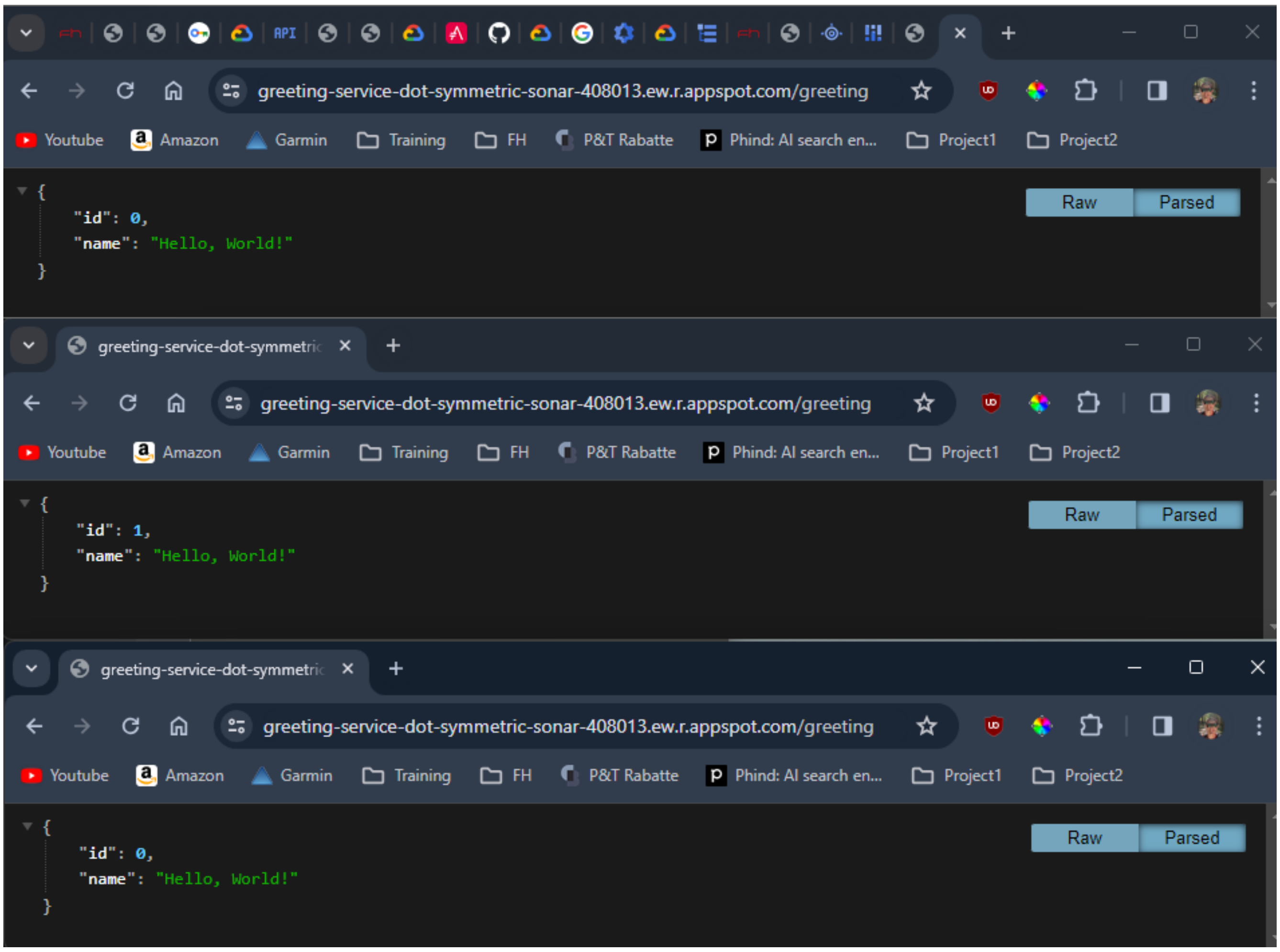
Configure the application to use manual scaling for 2 instances of class B1.

Listing 4. app.yaml

```
runtime: java17
env: standard
service: greeting-service
instance_class: B1
manual_scaling:
  instances: 2
handlers:
  - url: /.*
    script: this field is required, but ignored
```

What happens? Why? The app is now running on two instances. Each instance has its own id counter. So the id counter is not shared between the instances.

The first or second instance gets called and the counter gets increased there. That's why for example for the third call the count again starts at zero because it uses a different instance.


greeting-service-dot-symmetric-sonar-408013.ew.r.appspot.com/greeting
Youtube
Amazon
Garmin
Training
FH
P&T Rabatte
Phind: AI search en...
Project1
Project2
Raw
Parsed
greeting-service-dot-symmetric

```
{
    "id": 0,
    "name": "Hello, World!"
}
```

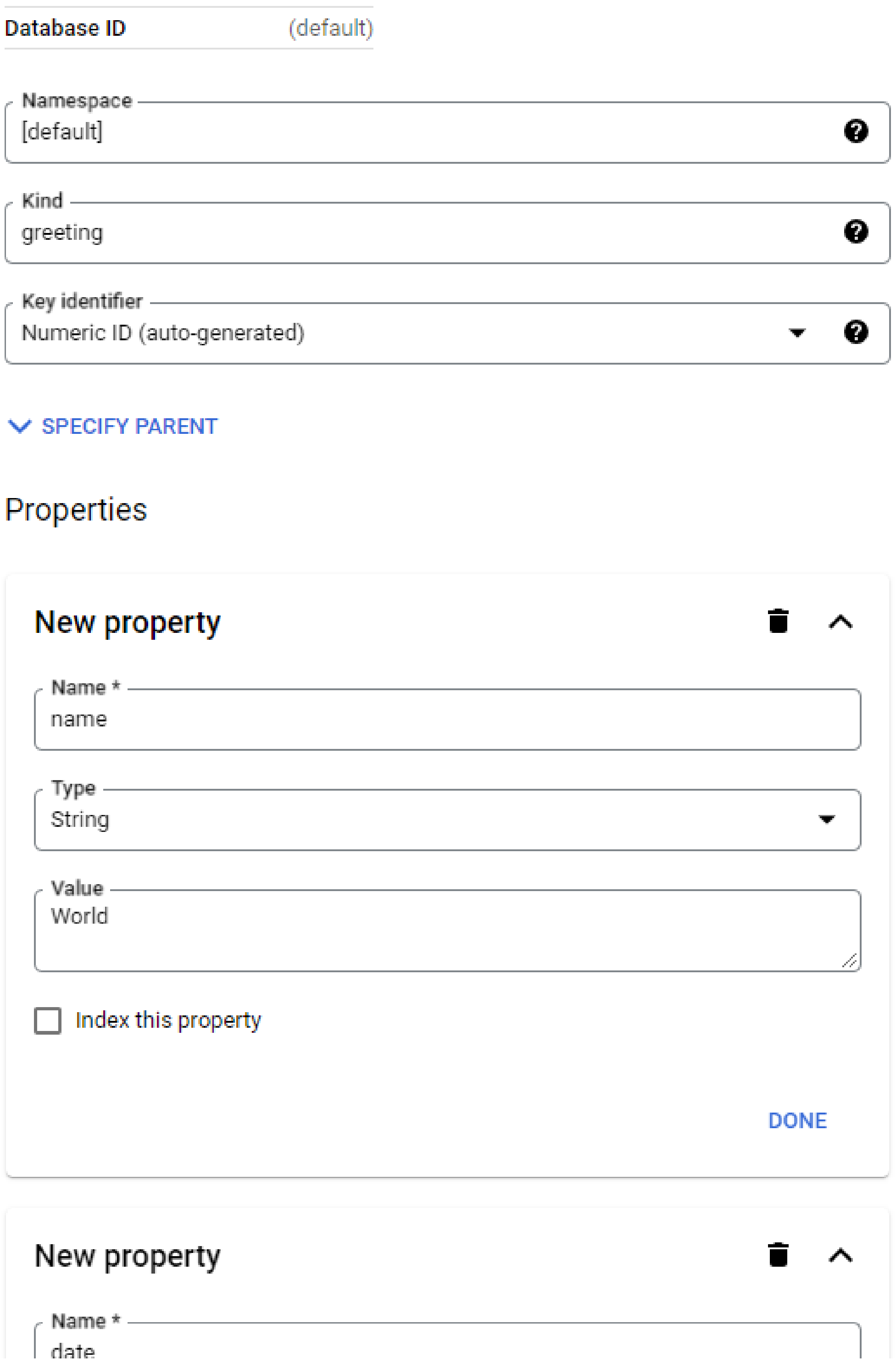
```
{
    "id": 1,
    "name": "Hello, World!"
}
```

```
{
    "id": 0,
    "name": "Hello, World!"
}
```

Figure 5. Multiple calls with 2 instances

4. Datastore

Add new entity to datastore with name and date.



Database ID (default)

Namespace: [default]

Kind: greeting

Key identifier: Numeric ID (auto-generated)

Properties

New property

Name *: name

Type: String

Value: World

Index this property: ☐

New property

Name *: date

Add following dependency to the pom.xml.

Listing 5. pom.xml

```
<dependency>
    <groupId>com.google.cloud</groupId>
    <artifactId>google-cloud-datastore</artifactId>
```

```
</dependency>
```

Add missing field to greeting class.

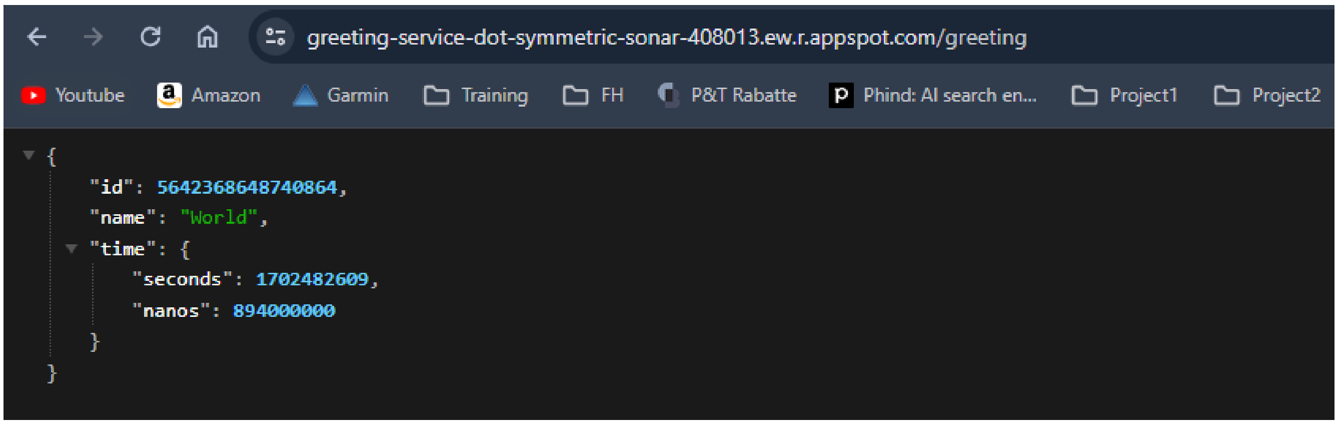
Listing 6. Greeting.java

```
package com.example.appengine.lab2;

import com.google.cloud.Timestamp;

public class Greeting {
    private final long id;
    private final String name;

    private final Timestamp time;

    public Greeting(long id, String name, Timestamp time) {
        this.id = id;
        this.name = name;
        this.time = time;
    }

    public long getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public Timestamp getTime() {
        return time;
    }
}
```

Use datastore in api controller.

```
private final Datastore datastore = DatastoreOptions.getDefaultInstance().
getService();
private final KeyFactory keyFactory = datastore.newKeyFactory().setKind("greeting");

@GetMapping("/greeting")
public Greeting getGreeting(@RequestParam(value = "name", defaultValue = "World")
String name) {

    Key key = datastore.allocateId(keyFactory.newKey());

    Greeting greeting = new Greeting(key.getId(), name, Timestamp.now());
    Entity greetingEntity = Entity.newBuilder(key)
            .set("name", name)
            .set("time", Timestamp.now())
            .build();
    datastore.put(greetingEntity);

    return greeting;
}
```

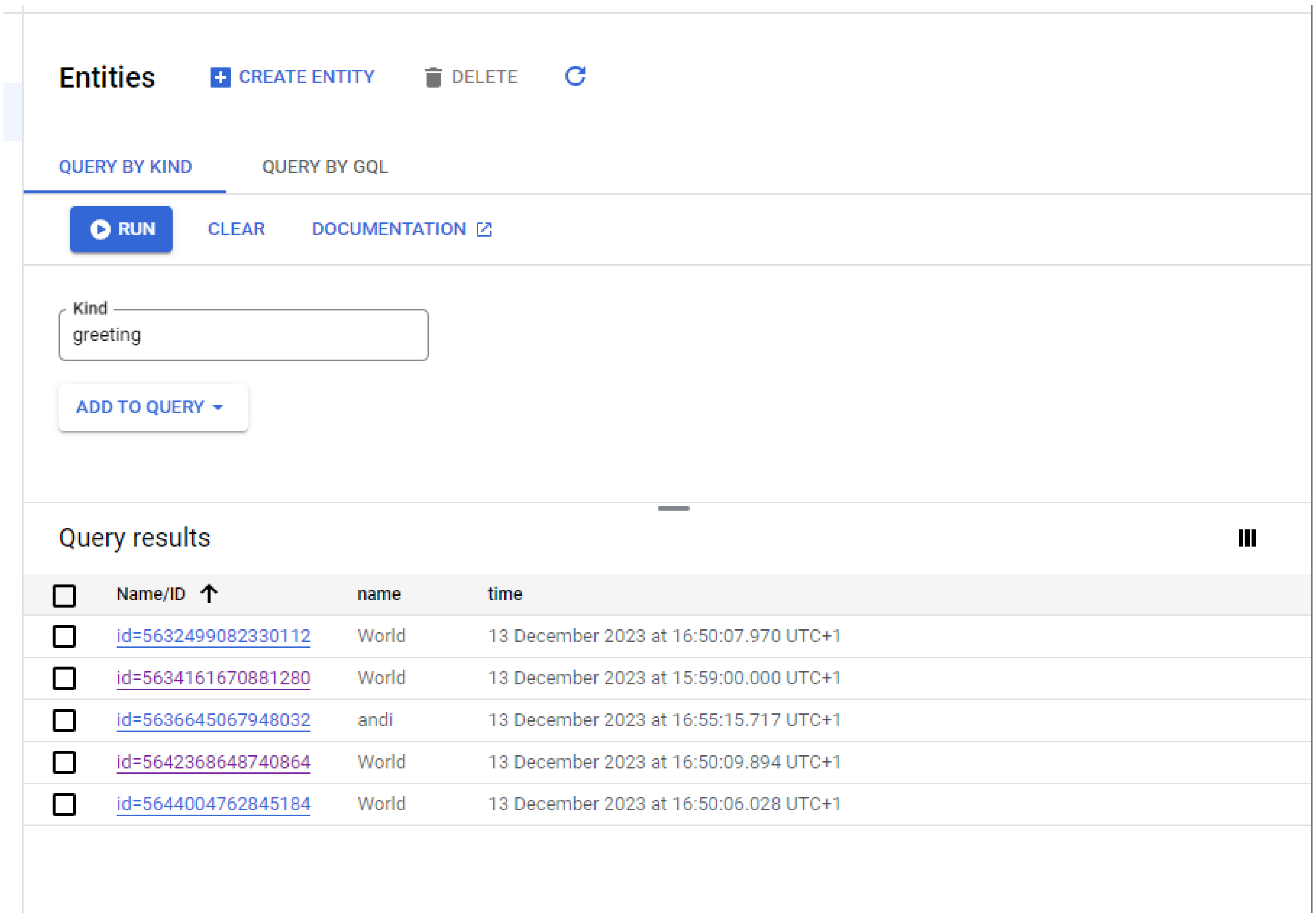
After deploying the app, we can see the new entity in the datastore. Id is now set by the datastore and timestamp is set to the current date.



```
{
    "id": 5642368648740864,
    "name": "World",
    "time": {
        "seconds": 1702482609,
        "nanos": 894000000
    }
}
```

Figure 6. New Entity

Entities in database



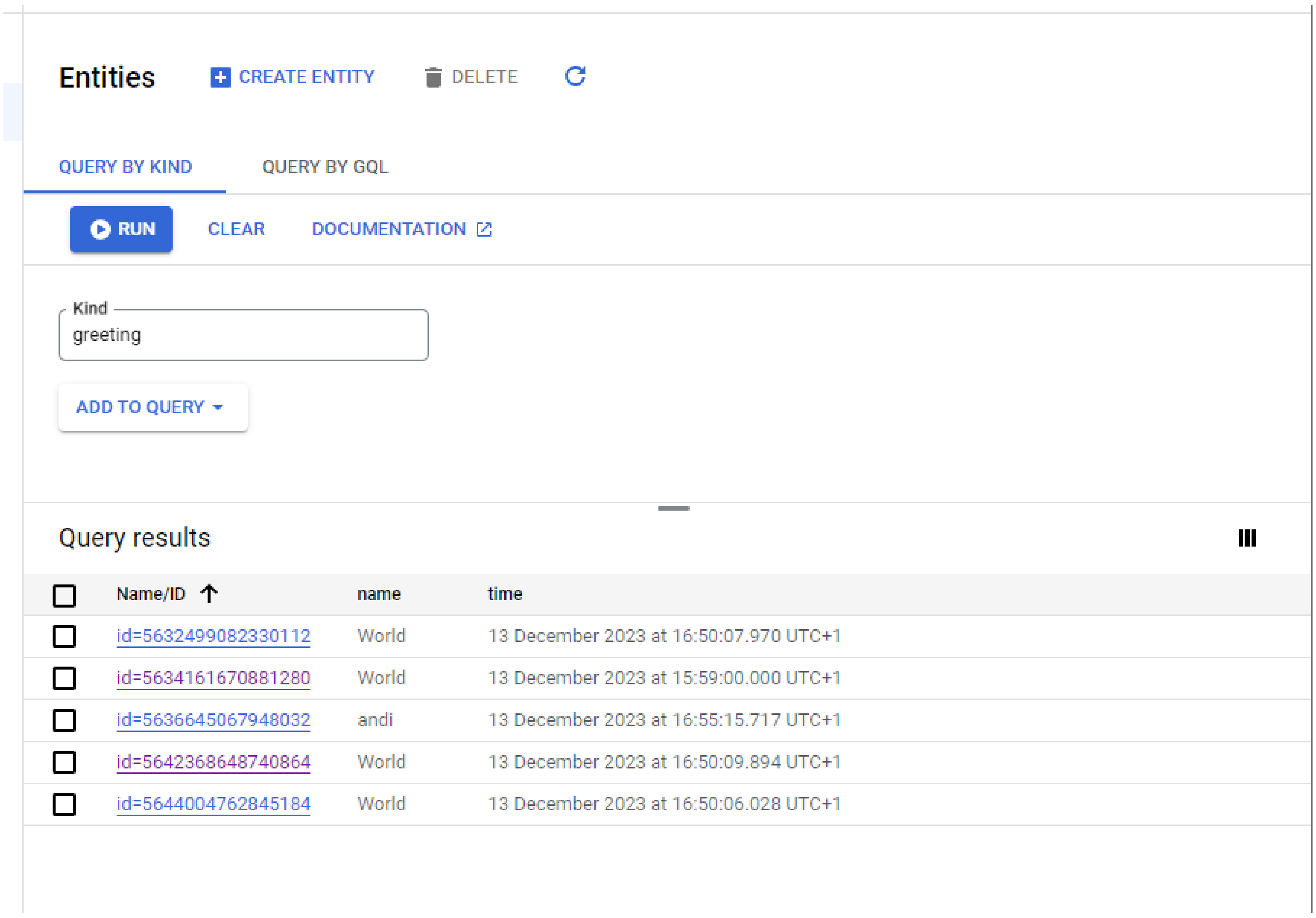
Kind: greeting

Query results

	Name/ID	name	time
☐	id=5632499082330112	World	13 December 2023 at 16:50:07.970 UTC+1
☐	id=5634161670881280	World	13 December 2023 at 15:59:00.000 UTC+1
☐	id=5636645067948032	andi	13 December 2023 at 16:55:15.717 UTC+1
☐	id=5642368648740864	World	13 December 2023 at 16:50:09.894 UTC+1
☐	id=5644004762845184	World	13 December 2023 at 16:50:06.028 UTC+1

Figure 7. Query results

Query entities with GQL query.



Kind: greeting

Query results

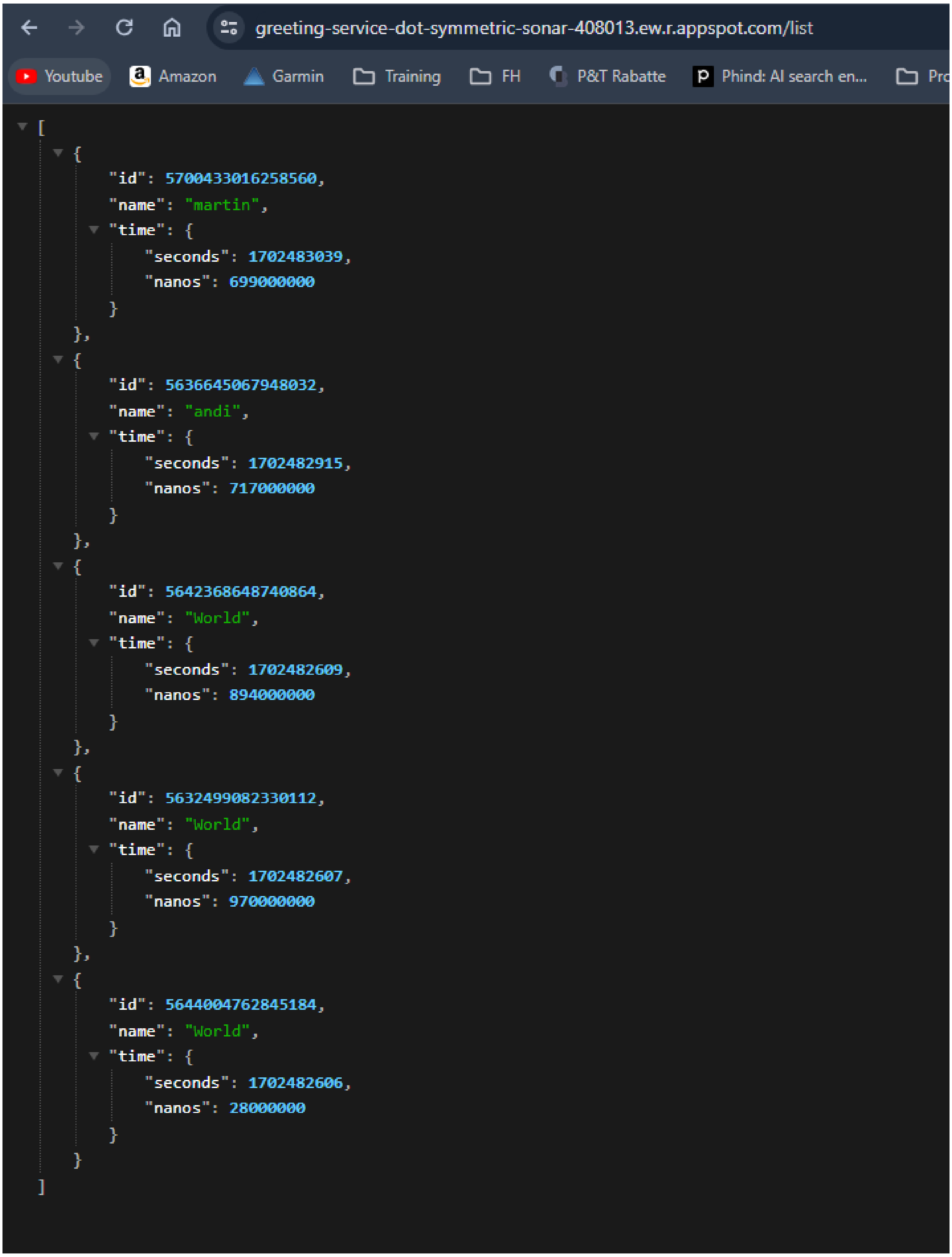
Name/ID	name	time
id=5632499082330112	World	13 December 2023 at 16:50:07.970 UTC+1
id=5634161670881280	World	13 December 2023 at 15:59:00.000 UTC+1
id=5636645067948032	andi	13 December 2023 at 16:55:15.717 UTC+1
id=5642368648740864	World	13 December 2023 at 16:50:09.894 UTC+1
id=5644004762845184	World	13 December 2023 at 16:50:06.028 UTC+1

Figure 8. GQL query

Retrieve entities with GQL query and return them as json.

Listing 7. GreetingController.java

```java
@GetMapping("/list")
public List<Greeting> listGreetings() {
    Query<Entity> q = Query.newGqlQueryBuilder(Query.ResultType.ENTITY, "SELECT * FROM
greeting ORDER BY time DESC LIMIT @limit")
            .setBinding("limit", 10)
            .build();
    QueryResults<Entity> results = datastore.run(q);
    ArrayList<Greeting> greetings = new ArrayList<>(10);
    while (results.hasNext()) {
        Entity e = results.next();
        greetings.add(new Greeting(
                e.getKey().getId(),
                e.getString("name"),
                e.getTimestamp("time")
        ));
    }
    return greetings;
}
```



greeting-service-dot-symmetric-sonar-408013.ew.r.appspot.com/list

```
[
  {
    "id": 5700433016258560,
    "name": "martin",
    "time": {
      "seconds": 1702483039,
      "nanos": 699000000
    }
  },
  {
    "id": 5636645067948032,
    "name": "andi",
    "time": {
      "seconds": 1702482915,
      "nanos": 717000000
    }
  },
  {
    "id": 5642368648740864,
    "name": "World",
    "time": {
      "seconds": 1702482609,
      "nanos": 894000000
    }
  },
  {
    "id": 5632499082330112,
    "name": "World",
    "time": {
      "seconds": 1702482607,
      "nanos": 970000000
    }
  },
  {
    "id": 5644004762845184,
    "name": "World",
    "time": {
      "seconds": 1702482606,
      "nanos": 28000000
    }
  }
]
```

5. Scheduling Jobs

Create a new .yaml file for the cron job.

Listing 8. cron.yaml

```yaml
cron:
  - description: "cleanup old greeting entries"
    url: /cleanup
    schedule: every 2 mins
    target: greeting-service
```

Add new endpoint to controller.

Listing 9. GreetingController.java

```java
@GetMapping("/cleanup")
public ResponseEntity<String> cleanup() {
    Query<Entity> q = Query.newGqlQueryBuilder(Query.ResultType.ENTITY, "SELECT * FROM
greeting order by time asc").build();

    QueryResults<Entity> results = datastore.run(q);

    List<Long> ids = new ArrayList<>();

    while (results.hasNext()) {
        Entity e = results.next();
        var id = e.getKey();
        var timestamp = e.getTimestamp("time");

        if(timestamp.getSeconds() > Timestamp.now().getSeconds() - 300) {
            break;
        }
        datastore.delete(id);
        ids.add(id.getId());
    }

    if(ids.size() > 0) {
        logger.info("Removed greetings with following ids: {}", String.join(", ", ids
.stream().map(Object::toString).toList()));
    }
    return ResponseEntity.ok().build();
}
```

Update pom.xml with following dependency for logging.

Listing 10. pom.xml

```xml
<dependency>
```

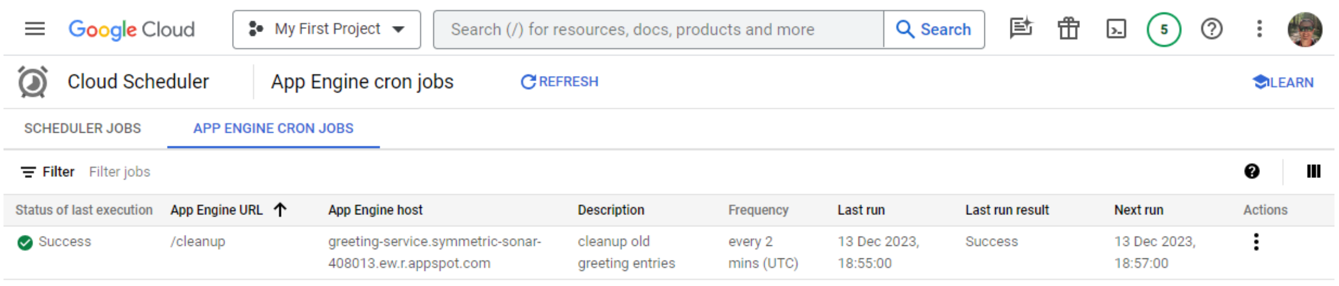
```
    <groupId>com.google.cloud</groupId>
    <artifactId>google-cloud-logging-logback</artifactId>
</dependency>
```

Deploy cron job:

`gcloud app deploy .\cron.yaml`

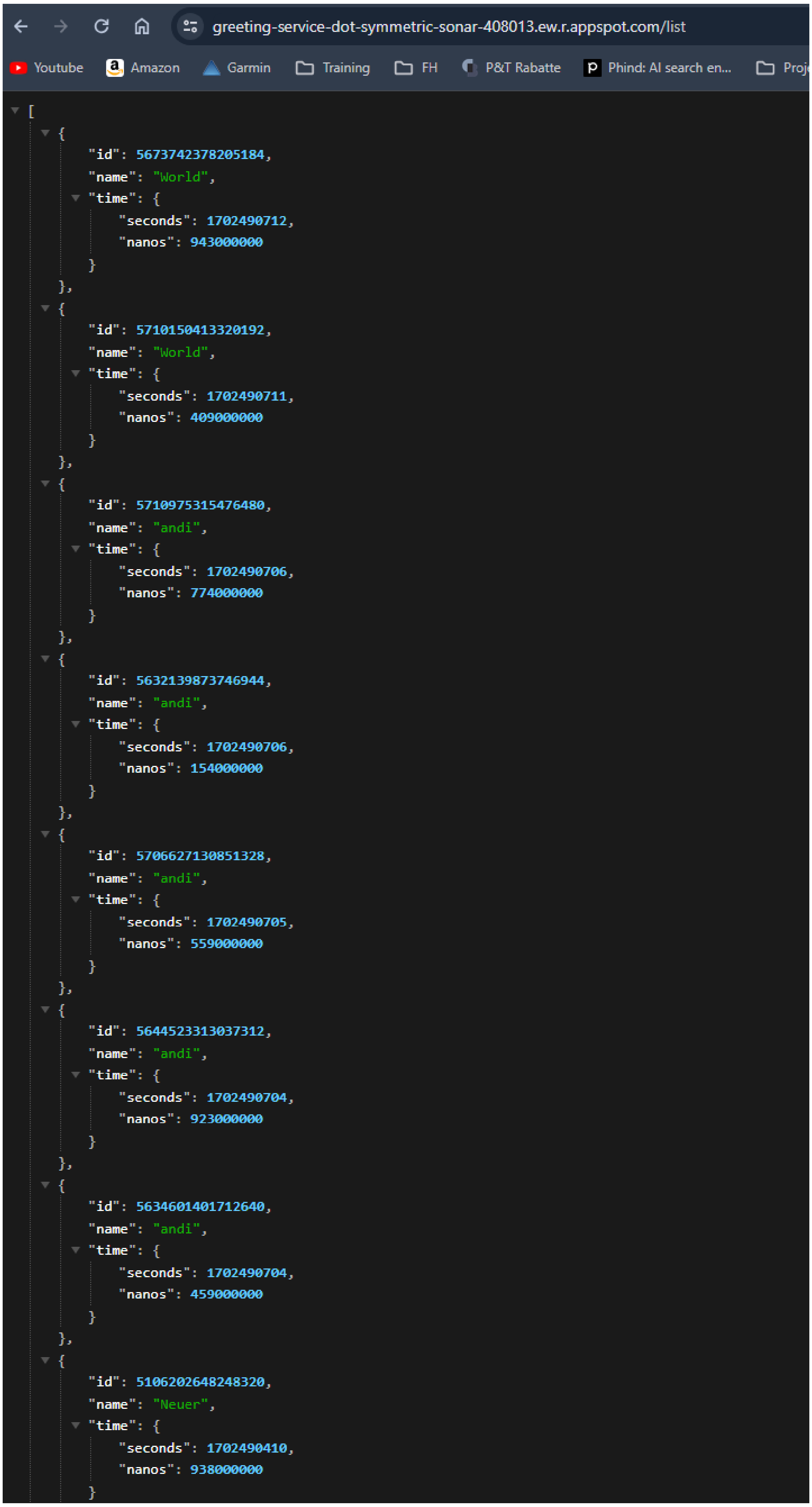
Deploy app: `gcloud app deploy`

View cron job:



Status of last execution	App Engine URL ↑	App Engine host	Description	Frequency	Last run	Last run result	Next run	Actions
✓ Success	/cleanup	greeting-service.symmetric-sonar-408013.ew.r.appspot.com	cleanup old greeting entries	every 2 mins (UTC)	13 Dec 2023, 18:55:00	Success	13 Dec 2023, 18:57:00	⋮

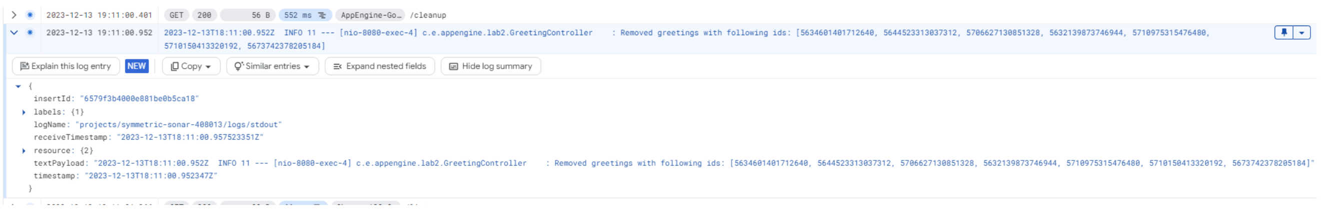
Figure 9. Cron Job



```
[
  {
    "id": 5673742378205184,
    "name": "World",
    "time": {
      "seconds": 1702490712,
      "nanos": 943000000
    }
  },
  {
    "id": 5710150413320192,
    "name": "World",
    "time": {
      "seconds": 1702490711,
      "nanos": 409000000
    }
  },
  {
    "id": 5710975315476480,
    "name": "andi",
    "time": {
      "seconds": 1702490706,
      "nanos": 774000000
    }
  },
  {
    "id": 5632139873746944,
    "name": "andi",
    "time": {
      "seconds": 1702490706,
      "nanos": 154000000
    }
  },
  {
    "id": 5706627130851328,
    "name": "andi",
    "time": {
      "seconds": 1702490705,
      "nanos": 559000000
    }
  },
  {
    "id": 5644523313037312,
    "name": "andi",
    "time": {
      "seconds": 1702490704,
      "nanos": 923000000
    }
  },
  {
    "id": 5634601401712640,
    "name": "andi",
    "time": {
      "seconds": 1702490704,
      "nanos": 459000000
    }
  },
  {
    "id": 5106202648248320,
    "name": "Neuer",
    "time": {
      "seconds": 1702490410,
      "nanos": 938000000
    }
```

Figure 10. Inserted Data

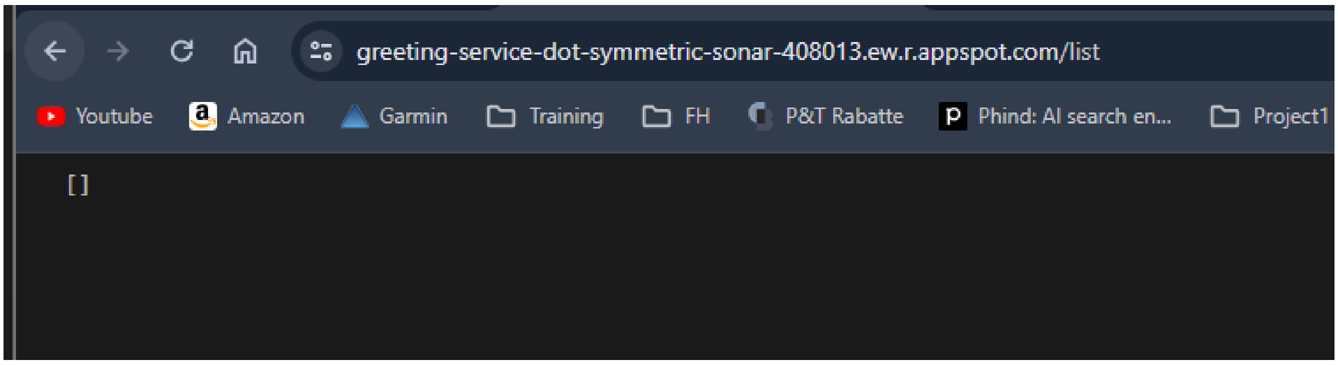
Wait until data got deleted and view logs:



```
2023-12-13 19:11:00.401  GET  200  56 B  552 ms  AppEngine-Go…  /cleanup
2023-12-13 19:11:00.952  2023-12-13T18:11:00.952Z  INFO 11 --- [nio-8080-exec-4] c.e.appengine.lab2.GreetingController    : Removed greetings with following ids: [5634601401712640, 5644523313037312, 5706627130851328, 5632139873746944, 5710975315476480,
                         5710150413320192, 5673742378205184]
```

```
{
  insertId: "6579f3b4000e881be0b5ca18"
  labels: {1}
  logName: "projects/symmetric-sonar-408013/logs/stdout"
  receiveTimestamp: "2023-12-13T18:11:00.957523351Z"
  resource: {2}
  textPayload: "2023-12-13T18:11:00.952Z  INFO 11 --- [nio-8080-exec-4] c.e.appengine.lab2.GreetingController    : Removed greetings with following ids: [5634601401712640, 5644523313037312, 5706627130851328, 5632139873746944, 5710975315476480, 5710150413320192, 5673742378205184]"
  timestamp: "2023-12-13T18:11:00.952347Z"
}
```

Figure 11. Logs



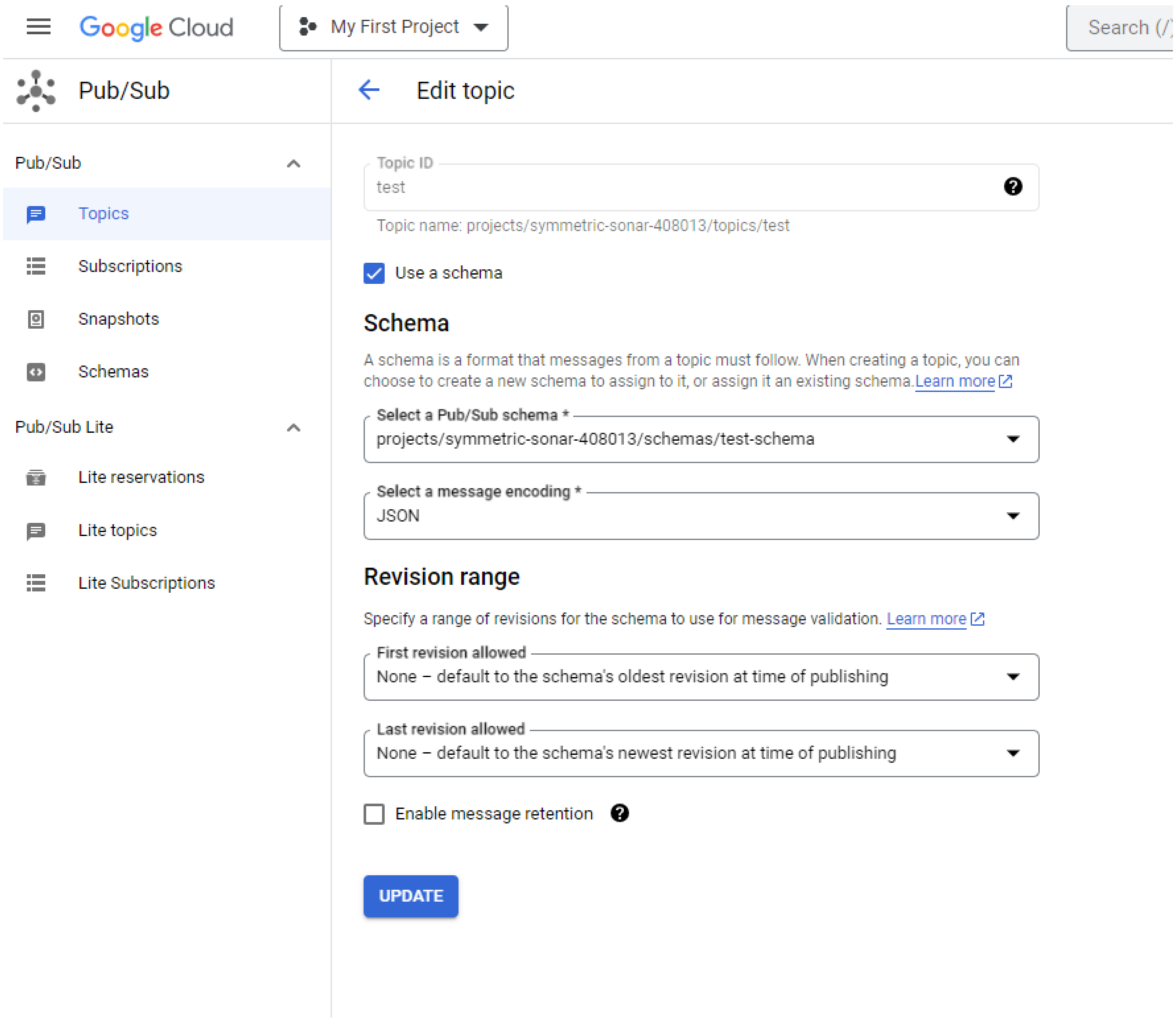
greeting-service-dot-symmetric-sonar-408013.ew.r.appspot.com/list

```
[]
```

Figure 12. Data

6. Pub/Sub

Create new topic with custom schema for validation.



Edit topic

Topic ID: test

Topic name: projects/symmetric-sonar-408013/topics/test

☑ Use a schema

Schema

A schema is a format that messages from a topic must follow. When creating a topic, you can choose to create a new schema to assign to it, or assign it an existing schema. Learn more

Select a Pub/Sub schema *: projects/symmetric-sonar-408013/schemas/test-schema

Select a message encoding *: JSON

Revision range

Specify a range of revisions for the schema to use for message validation. Learn more

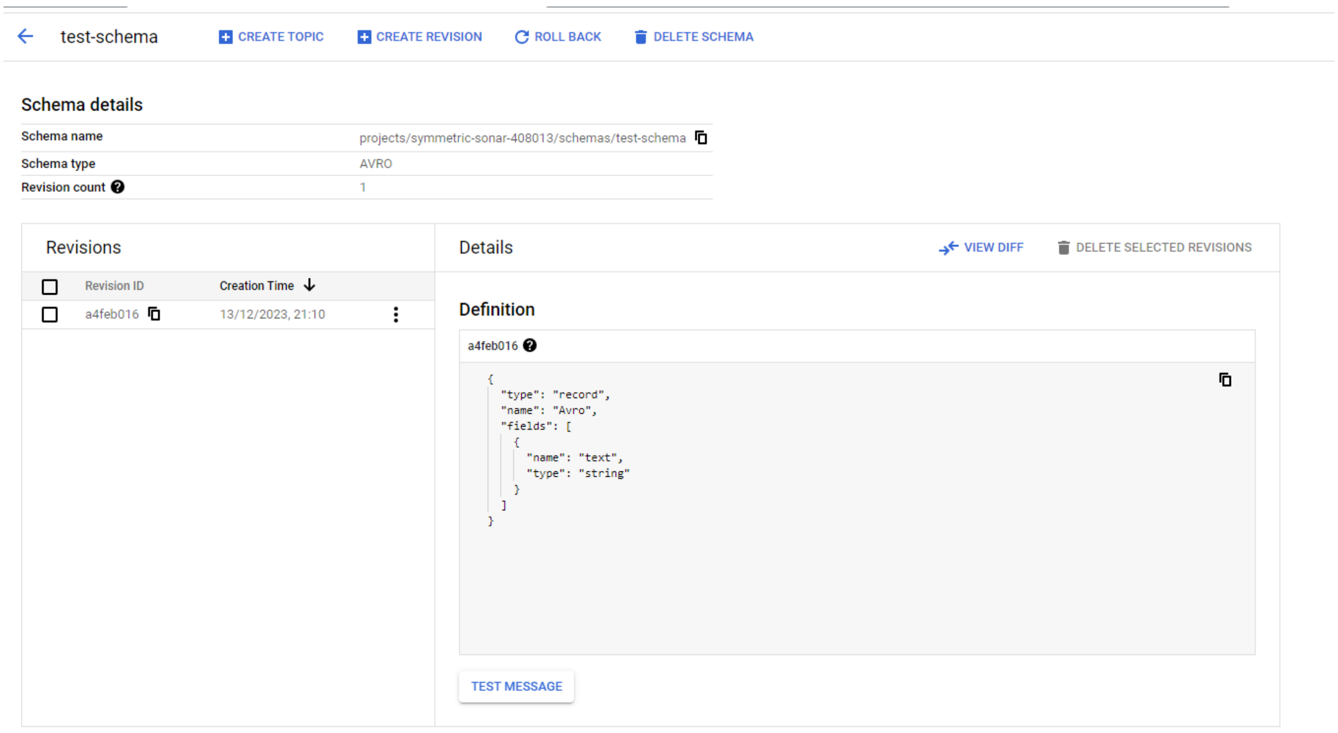
First revision allowed: None – default to the schema's oldest revision at time of publishing

Last revision allowed: None – default to the schema's newest revision at time of publishing

☐ Enable message retention

UPDATE

Figure 13. Topic



Schema details

Schema name	projects/symmetric-sonar-408013/schemas/test-schema
Schema type	AVRO
Revision count	1

Revisions

Revision ID	Creation Time
a4feb016	13/12/2023, 21:10

Details

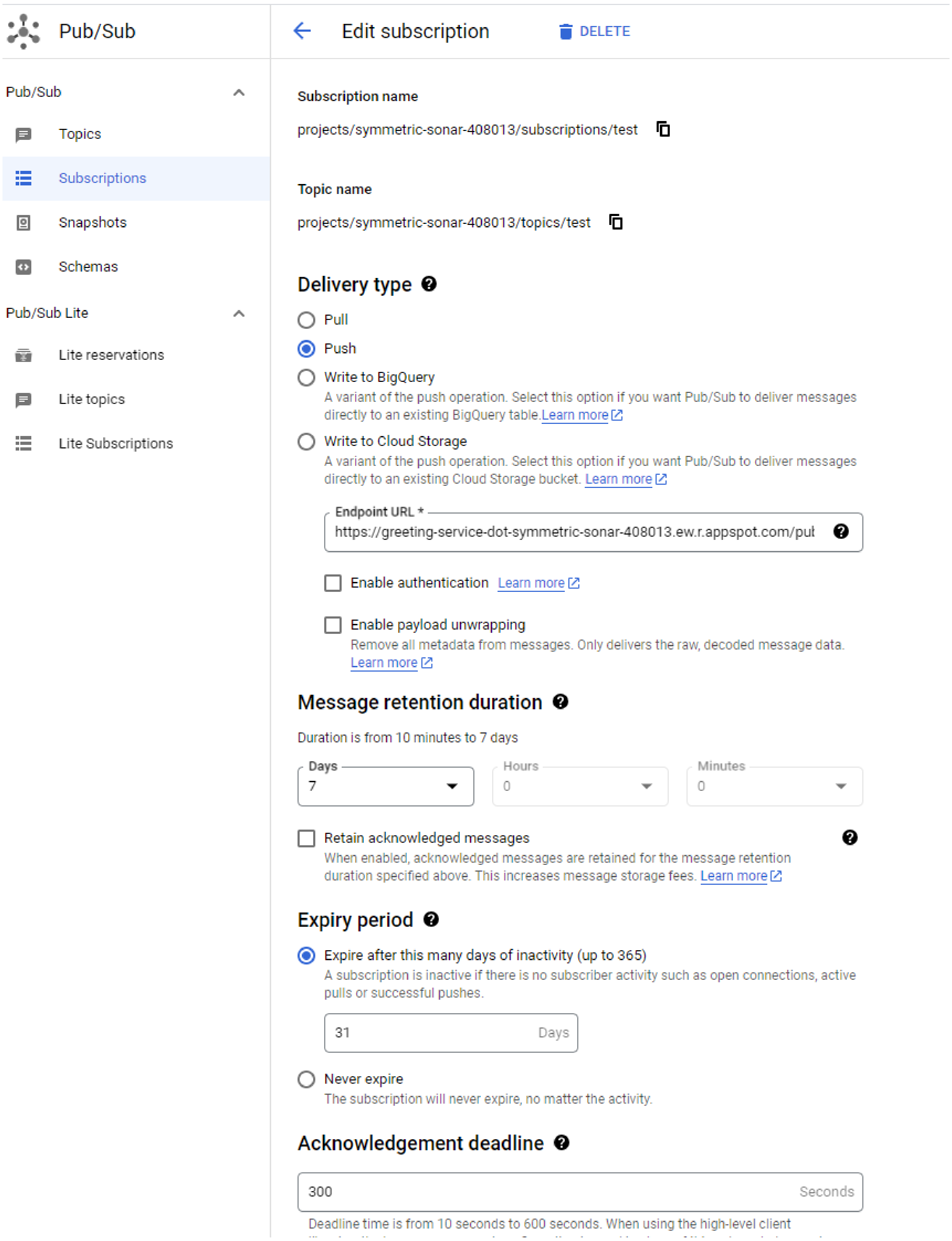
Definition

a4feb016

```
{
  "type": "record",
  "name": "Avro",
  "fields": [
    {
      "name": "text",
      "type": "string"
    }
  ]
}
```

Figure 14. Custom Schema

Add a subscription with the previous created topic and the endpoint url.



Subscription name

projects/symmetric-sonar-408013/subscriptions/test

Topic name

projects/symmetric-sonar-408013/topics/test

Delivery type

( ) Pull
(●) Push
( ) Write to BigQuery
A variant of the push operation. Select this option if you want Pub/Sub to deliver messages directly to an existing BigQuery table. Learn more
( ) Write to Cloud Storage
A variant of the push operation. Select this option if you want Pub/Sub to deliver messages directly to an existing Cloud Storage bucket. Learn more

Endpoint URL *
https://greeting-service-dot-symmetric-sonar-408013.ew.r.appspot.com/pul

☐ Enable authentication Learn more

☐ Enable payload unwrapping
Remove all metadata from messages. Only delivers the raw, decoded message data. Learn more

Message retention duration

Duration is from 10 minutes to 7 days

Days: 7 | Hours: 0 | Minutes: 0

☐ Retain acknowledged messages
When enabled, acknowledged messages are retained for the message retention duration specified above. This increases message storage fees. Learn more

Expiry period

(●) Expire after this many days of inactivity (up to 365)
A subscription is inactive if there is no subscriber activity such as open connections, active pulls or successful pushes.

31 Days

( ) Never expire
The subscription will never expire, no matter the activity.

Acknowledgement deadline

300 Seconds

Deadline time is from 10 seconds to 600 seconds. When using the high-level client

Figure 15. Subscription

Implement new pub/sub controller.

Listing 11. PubSubController.java

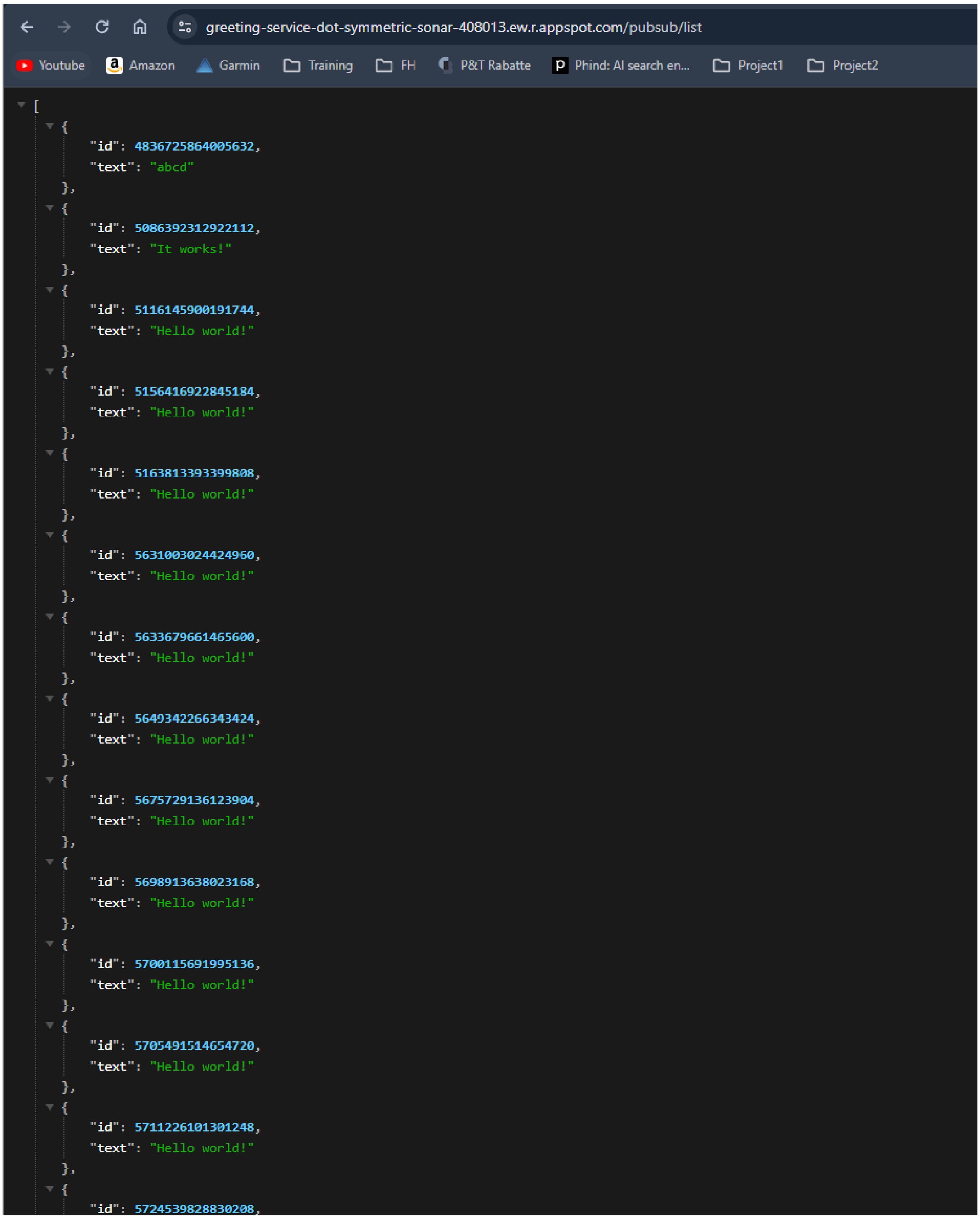
```
@RestController
```

```
public class PubSubController {

    private final Datastore datastore = DatastoreOptions.getDefaultInstance
().getService();
    private final KeyFactory keyFactory = datastore.newKeyFactory().setKind(
"message");

    Logger logger = LoggerFactory.getLogger(PubSubController.class);

    @PostMapping("/pubsub/endpoint")
    public ResponseEntity<String> handlePubSubMessage(@RequestBody String
pubSubMessage) {
         logger.info("Received Pub/Sub message with payload: " + pubSubMessage);

        try {
            var messageContent = getMessageContent(pubSubMessage);
            Key key = datastore.allocateId(keyFactory.newKey());
            Entity entity = Entity.newBuilder(key)
                    .set("text", messageContent.getText())
                    .build();
            datastore.put(entity);

            // Return a success response
            return ResponseEntity.ok("Message processed successfully");
        } catch (IOException e) {
            logger.error("Error parsing JSON: " + pubSubMessage, e);
            // Handle parsing errors
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Error
processing Pub/Sub message");
        }
    }

    private MessageContent getMessageContent(String messageContent) throws IOException
{

        JsonElement jsonRoot = JsonParser.parseString(messageContent).
getAsJsonObject();
        String messageStr = jsonRoot.getAsJsonObject().get("message").toString();

        var objectMapper = new ObjectMapper();
        objectMapper.configure(DeserializationFeature.FAIL_ON_UNKNOWN_PROPERTIES,
false);
        objectMapper.configure(MapperFeature.ACCEPT_CASE_INSENSITIVE_PROPERTIES,
true);

        Message message = objectMapper.readValue(messageStr, Message.class);
        String decodedMessage = new String(Base64.getDecoder().decode(message.
getData()));
        return objectMapper.readValue(decodedMessage, MessageContent.class);
    }
    @GetMapping("/pubsub/list")
    public ResponseEntity<List<MessageContent>> listStoredMessages() {
        Query<Entity> q = Query.newGqlQueryBuilder(Query.ResultType.ENTITY, "SELECT *
FROM message").build();

        QueryResults<Entity> results = datastore.run(q);

        ArrayList<MessageContent> messages = new ArrayList<>();

        while (results.hasNext()) {
            Entity e = results.next();
            var id = e.getKey();
            var message = e.getString("text");
            messages.add(new MessageContent(id.getId(), message));
        }

        return ResponseEntity.ok(messages);
    }
}
```

Execution with:

```
gcloud pubsub topics publish test --message='{"text":"It works!"}'
```

Result:



```
[
  {
    "id": 4836725864005632,
    "text": "abcd"
  },
  {
    "id": 5086392312922112,
    "text": "It works!"
  },
  {
    "id": 5116145900191744,
    "text": "Hello world!"
  },
  {
    "id": 5156416922845184,
    "text": "Hello world!"
  },
  {
    "id": 5163813393399808,
    "text": "Hello world!"
  },
  {
    "id": 5631003024424960,
    "text": "Hello world!"
  },
  {
    "id": 5633679661465600,
    "text": "Hello world!"
  },
  {
    "id": 5649342266343424,
    "text": "Hello world!"
  },
  {
    "id": 5675729136123904,
    "text": "Hello world!"
  },
  {
    "id": 5698913638023168,
    "text": "Hello world!"
  },
  {
    "id": 5700115691995136,
    "text": "Hello world!"
  },
  {
    "id": 5705491514654720,
    "text": "Hello world!"
  },
  {
    "id": 5711226101301248,
    "text": "Hello world!"
  },
  {
    "id": 5724539828830208,
```

Figure 16. Result