

Deploying a Spring Boot Application on Google Cloud Run

Spring Boot is one of the most popular frameworks for building Java-based Microservices. With **Google Cloud Run**, you can deploy your Spring boot application in a **server less, fully managed environment** without worrying about provisioning servers, scaling or infrastructure management.

In this article, I will walk you step-by-step through deploying a spring Boot application to Cloud Run.

GitHub:

<https://github.com/lalamanil/IntegratingAccidentPredictionModelOfVideo.git>

Prerequisite

- ✓ A Google Cloud account with billing enabled
- ✓ Google Cloud SDK installed locally
- ✓ A simple Spring Boot application (Source Code)

Step 1: Verify gcloud Installation and Project Configuration

First, make sure the Google Cloud CLI (gcloud) is installed and accessible:

gcloud --version

```
lalamanil@Anils-MacBook-Pro ~ % gcloud --version
Google Cloud SDK 537.0.0
app-engine-java 2.0.38
app-engine-python 1.9.118
bq 2.1.22
cloud-datastore-emulator 2.3.1
core 2025.08.29
gcloud-crc32c 1.0.0
gke-gcloud-auth-plugin 0.5.10
gsutil 5.35
kubectl 1.33.4
```

Then Check the currently active project

gcloud config get-value project

```
lalamanil@Anils-MacBook-Pro ~ % gcloud config get-value project  
videoanalyzer-455321
```

If you need to set a project

gcloud config set project <PROJECT_ID>

Finally, set the default compute region and zone (Optional but recommended)

gcloud config set compute/region us-central1

```
lalamanil@Anils-MacBook-Pro ~ % gcloud config set compute/region us-central1  
Updated property [compute/region].  
lalamanil@Anils-MacBook-Pro ~ %
```

gcloud config set compute/zone us-central1-a

```
lalamanil@Anils-MacBook-Pro ~ % gcloud config set compute/region us-central1  
Updated property [compute/region].  
lalamanil@Anils-MacBook-Pro ~ % gcloud config set compute/zone us-central1-a  
Updated property [compute/zone].  
lalamanil@Anils-MacBook-Pro ~ %
```

STEP:2 Verify Required APIS are Enabled

Before deploying, ensure the required services are enabled in your project.

Check enabled services

gcloud services list --enabled

```

lalamanil@Anils-MacBook-Pro ~ % gcloud services list --enabled
NAME                      TITLE
aiplatform.googleapis.com   Vertex AI API
analyticshub.googleapis.com Analytics Hub API
artifactregistry.googleapis.com Artifact Registry API
bigquery.googleapis.com    BigQuery API
bigqueryconnection.googleapis.com BigQuery Connection API
bigquerydatapolicy.googleapis.com BigQuery Data Policy API
bigquerymigration.googleapis.com BigQuery Migration API
bigqueryreservation.googleapis.com BigQuery Reservation API
bigquerystorage.googleapis.com BigQuery Storage API
cloudbuild.googleapis.com   Cloud Build API
cloudtrace.googleapis.com   Cloud Trace API
compute.googleapis.com     Compute Engine API
containerregistry.googleapis.com Container Registry API
dataform.googleapis.com    Dataform API
dataplex.googleapis.com    Cloud Dataplex API
datastore.googleapis.com   Cloud Datastore API
iam.googleapis.com         Identity and Access Management (IAM) API
iamcredentials.googleapis.com IAM Service Account Credentials API
logging.googleapis.com    Cloud Logging API
monitoring.googleapis.com Cloud Monitoring API
oslogin.googleapis.com    Cloud OS Login API
pubsub.googleapis.com      Cloud Pub/Sub API
run.googleapis.com        Cloud Run Admin API
servicemanagement.googleapis.com Service Management API
serviceusage.googleapis.com Service Usage API
sql-component.googleapis.com Cloud SQL
storage-api.googleapis.com Google Cloud Storage JSON API
storage-component.googleapis.com Cloud Storage
storage.googleapis.com    Cloud Storage API
lalamanil@Anils-MacBook-Pro ~ %

```

If any are missing, enable them using below commands:

```

gcloud services enable artifactregistry.googleapis.com
gcloud services enable cloudbuild.googleapis.com
gcloud services enable run.googleapis.com
gcloud services enable compute.googleapis.com

```

STEP:3 Create an Artifact Registry Repository

Artifact Registry is the recommended place to store your Docker Image

gcloud artifacts repositories list

```

lalamanil@Anils-MacBook-Pro ~ % gcloud artifacts repositories list
Listing items under project videoanalyzer-455321, across all locations.

Listed 0 items.
lalamanil@Anils-MacBook-Pro ~ %

```

Create the repository:

```
gcloud artifacts repositories create springboot-docker-repo --repository-format=docker --location=us-central1 --description="Docker repo for Spring Boot images"
```

```
lalamanil@Anils-MacBook-Pro ~ % gcloud artifacts repositories create springboot-docker-repo --repository-format=docker --location=us-central1 --description="Docker repo for Spring Boot images"
Create request issued for: [springboot-docker-repo]
Waiting for operation [projects/videoanalyzer-455321/locations/us-central1/operations/8465909f-2fdb-42ae-accd-34a77f660350] to complete
...done.
Created repository [springboot-docker-repo].
lalamanil@Anils-MacBook-Pro ~ %
```

Verify:

gcloud artifacts repositories list

```
lalamanil@Anils-MacBook-Pro ~ % gcloud artifacts repositories list
Listing items under project videoanalyzer-455321, across all locations.
```

REPOSITORY	FORMAT	MODE	DESCRIPTION	ARTIFACT_REGISTRY	LOCATION	LABELS	ENCRYPTION	CREATE_TIME
UPDATE_TIME	SIZE (MB)	DOCKER	STANDARD_REPOSITORY	Docker repo for Spring Boot images	us-central1		Google-managed key	
2025-09-06T13:04:51	0	DOCKER	STANDARD_REPOSITORY	Docker repo for Spring Boot images	us-central1		Google-managed key	2025-09-06T13:04:51

The screenshot shows the Google Cloud Artifact Registry interface. On the left, there's a sidebar with 'Repositories' and 'Settings'. The main area has a header with 'Repositories', '+ Create repository', 'Edit repository', 'Delete', and 'Setup instructions'. Below this is a 'Turn on vulnerability scanning' dialog box. To the right, there's a search bar, a 'Refresh' button, and tabs for 'Permissions' and 'Labels'. A message says 'Please select at least one resource.' At the bottom, there's a table with columns: Name, Format, Type, Location, Scanning (Disabled), and Description. One row is visible: 'springboot-docker-repo' (Format: Docker, Type: Standard, Location: us-central1 (Iowa), Scanning: Disabled).

STEP:4 Create a multi stage Docker File in Spring boot Application

In the root of your Spring Boot project, add a Dockerfile:

```

eclipse-workspace-AI - IntegratingAccidentPredictionModelOfVideo/Dockerfile - Eclipse IDE

Package Explorer X Dockerfile X
src/main/java
src/main/resources
src/test/java
src/test/resources
JRE System Library [Java SE 17.0.9 [17.0.9]]
Maven Dependencies
src
target
Dockerfile
pom.xml
README.md

1 #informing the docker to use maven Image
2 FROM maven:3.8.3-openjdk-17 AS MAVEN_BUILD
3 #Setting the working directory
4 WORKDIR /build/
5 #Copy the pom.xml into working directory
6 COPY pom.xml /build/pom.xml
7 #Copy the src into working directory
8 COPY src /build/src
9 #Running the mvn command to perform packaging
10 RUN mvn package -DskipTests
11 #Inform the docker to use JRE image to execute the package generated in previous stage
12 FROM eclipse-temurin:17-jdk-jammy
13 #Setting current Working directory
14 WORKDIR /app
15
16 # Set library path for native dependencies
17 ENV LD_LIBRARY_PATH=/usr/lib/x86_64-linux-gnu
18
19 #Copy the artifact generated in previous build to working directory
20 COPY --from=MAVEN_BUILD /build/target/*.jar /app/finalartifact.jar
21 #Defining the entrypoint
22 ENTRYPOINT ["java","-jar", "finalartifact.jar"]
23
24

```

STEP:5 Build and Push the Docker image Using Cloud Build

Instead of building locally with Docker, you can use **Cloud Build** to build and push the image directly to Artifact Registry.

Set the directory path to root folder of your spring boot application

```

lalamanil@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo % ls
Dockerfile      README.md      pom.xml      src
lalamanil@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo %

```

Checking Artifact repository if any images are already present.

```

gcloud artifacts docker images list us-central1-docker.pkg.dev/
videoanalyzer-455321/springboot-docker-repo

```

```

lalamanil@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo % gcloud config get-value project
videoanalyzer-455321
lalamanil@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo % gcloud artifacts repositories list
Listing items under project videoanalyzer-455321, across all locations.

REPOSITORY          FORMAT MODE           DESCRIPTION          ARTIFACT_REGISTRY          LOCATION          LABELS          ENCRYPTION          CREATE_TIME
UPDATE_TIME          SIZE (MB)          DOCKER          STANDARD_REPOSITORY          Docker repo for Spring Boot images          us-central1          Google-managed key          2025-09-06T13:04:51
2025-09-06T13:04:51  0
lalamanil@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo % gcloud artifacts docker images list us-central1-docker.pkg.dev/videoanalyzer-455321/springboot-docker-repo
Listing items under project videoanalyzer-455321, location us-central1, repository springboot-docker-repo.

Listed 0 items.
lalamanil@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo %

```

Build & Push:

```

gcloud builds submit --tag=us-central1-docker.pkg.dev/
videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy:v1

```

This command will:

- ✓ Build the docker image in Google Cloud
- ✓ Tag it with the specified Artifact Registry path
- ✓ Push it automatically to your Artifact Registry repository

```
lalamanil@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo % gcloud builds submit --tag=us-central1-docker.pkg.dev/videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy:v1
Creating temporary archive of 27 file(s) totalling 56.2 KiB before compression.
Some files were not included in the source upload.

Check the gcloud log [/Users/lalamanil/.config/gcloud/logs/2025.09.06/14.00.50.260030.log] to see which files and the contents of the default gcloudignore file used (see '$ gcloud topic gcloudignore' to learn more).

Uploading tarball of [...] to [gs://videoanalyzer-455321_cloudbuild/source/1757181650.310461-ee86dfda5bbe4f43be756443404d1e32.tgz]
Created [https://cloudbuild.googleapis.com/v1/projects/videoanalyzer-455321/locations/global/builds/1b5f1c2f-67ed-4e85-9c0a-304f3be9298d].
Waiting for build to complete. Polling interval: 1 second.
----- REMOTE BUILD OUTPUT -----
starting build "1b5f1c2f-67ed-4e85-9c0a-304f3be9298d"

FETCHSOURCE
Fetching storage object: gs://videoanalyzer-455321_cloudbuild/source/1757181650.310461-ee86dfda5bbe4f43be756443404d1e32.tgz#1757181650823582
Copying gs://videoanalyzer-455321_cloudbuild/source/1757181650.310461-ee86dfda5bbe4f43be756443404d1e32.tgz#1757181650823582...
/ [1 files] 11.9 KiB/ 11.9 KiB
Operation completed over 1 objects/11.9 KiB.
BUILD
Already have image (with digest): gcr.io/cloud-builders/docker
Sending build context to Docker daemon 91.14kB
Step 1/1 : FROM maven:3.8.3-openjdk-17 AS MAVEN_BUILD
3.8.3-openjdk-17: Pulling from library/maven
28587b66475: Pulling fs layer
b1655352c888: Pulling fs layer
1f9644f00e96: Pulling fs layer
6cd514e60c46: Pulling fs layer
0ab2a7ea62ca: Pulling fs layer
d72072d9f9d3: Pulling fs layer
9eca236510fe: Pulling fs layer
6cd514e60c46: Waiting
0ab2a7ea62ca: Waiting
d72072d9f9d3: Waiting
```

```
Step 2/10 : WORKDIR /build/
--> Running in 48ed587c43b2
Removing intermediate container 48ed587c43b2
--> 1c04643fb017
Step 3/10 : COPY pom.xml /build/pom.xml
--> 5e1327eb4420
Step 4/10 : COPY src /build/src
--> 4d81b4f5df82
Step 5/10 : RUN mvn package -DskipTests
--> Running in e6cc1a49d04c
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/3.2.0/spring-boot-starter-parent-3.2.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/3.2.0/spring-boot-starter-parent-3.2.0.pom (13 kB at 28 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/3.2.0/spring-boot-starter-parent-3.2.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/protobuf/protobuf-bom/4.29.4/protobuf-bom-4.29.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/protobuf/protobuf-bom/4.29.4/protobuf-bom-4.29.4.pom (3.7 kB at 133 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/google/cloud/first-party-dependencies/3.45.1/first-party-dependencies-3.45.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/cloud/first-party-dependencies/3.45.1/first-party-dependencies-3.45.1.pom (2 kB at 79 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/google/cloud/google-cloud-shared-config/1.15.0/google-cloud-shared-config-1.15.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/cloud/google-cloud-shared-config/1.15.0/google-cloud-shared-config-1.15.0.pom (26 kB at 727 kB/s)
```

```

Step 6/10 : FROM eclipse-temurin:17-jdk-jammy
17-jdk-jammy: Pulling from library/eclipse-temurin
0098d907609: Pulling fs layer
6b80e9ed2b1: Pulling fs layer
203b1c9c3e5c: Pulling fs layer
503d637b4327: Pulling fs layer
5d5afad7028: Pulling fs layer
503d637b4327: Waiting
5d5afad7028: Waiting
60d98d907669: Verifying Checksum
60d98d907669: Download complete
6b80e9ed2b1: Verifying Checksum
6b80e9ed2b1: Download complete
503d637b4327: Verifying Checksum
503d637b4327: Download complete
5d5afad7028: Verifying Checksum
5d5afad7028: Download complete
203b1c9c3e5c: Verifying Checksum
203b1c9c3e5c: Download complete
60d98d907669: Pull complete
6b80e9ed2b1: Pull complete
203b1c9c3e5c: Pull complete
503d637b4327: Pull complete
5d5afad7028: Pull complete
Digest: sha256:a8a552a417551699694a71119e1c410a4b65ed79cb98b16584bc91c0d491b6
Status: Downloaded newer image for eclipse-temurin:17-jdk-jammy
--> 59db37472447
Step 7/10 : WORKDIR /app
--> Running in b67604480eea
Removing intermediate container b67604480eea
--> fd3c0e8e2d2c
Step 8/10 : ENV LD_LIBRARY_PATH=/usr/lib/x86_64-linux-gnu
--> Running in a550d7083206
Removing intermediate container a550d7083206
--> f7ff1ef23419
Step 9/10 : COPY --from=MAVEN_BUILD /build/target/*.jar /app/finalartifact.jar
--> efd4bf80f759
Step 10/10 : ENTRYPOINT ["java", "-jar", "finalartifact.jar"]
--> Running in 5f111ef620f0

```

```

Successfully tagged us-central1-docker.pkg.dev/videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy:v1
PUSH
Pushing us-central1-docker.pkg.dev/videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy:v1
The push refers to repository [us-central1-docker.pkg.dev/videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy]
b8cc4b2f9214: Preparing
06e07e686495: Preparing
1f4deb857872: Preparing
b277e1e48f73: Preparing
7dc7c028f1a3: Preparing
d99b84dad5b7: Preparing
dce6ebddad5f9: Preparing
d99b84dad5b7: Waiting
dce6ebddad5f9: Waiting
06e07e686495: Pushed
1f4deb857872: Pushed
b277e1e48f73: Pushed
7dc7c028f1a3: Pushed
dce6ebddad5f9: Pushed
d99b84dad5b7: Pushed
b8cc4b2f9214: Pushed
v1: digest: sha256:c13cf846e7f411a59ebe22c26990f9314e5d5611cff3acbfe16f30194089c4aa size: 1789
DONE

```

ID	CREATE_TIME	DURATION	SOURCE	STATUS
1b5f1c2f-67ed-4e85-9c0a-304f3be9298d	2025-09-06T18:00:51+00:00	3M31S	gs://videoanalyzer-455321_cloudbuild/source/1757181650.310461-ee86dfda5bbe4f43be756443404d1e32.tgz	us-central1-docker.pkg.dev/videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy:v1 SUCCESS
lalamani1@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo %				

Verify:

Image is created and pushed to artifact registry

```

lalamani1@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo % gcloud artifacts docker images list us-central1-docker.pkg.dev/videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy
Listing items under project videoanalyzer-455321, location us-central1, repository springboot-docker-repo.

IMAGE          CREATE_TIME      UPDATE_TIME      SIZE          DIGEST
us-central1-docker.pkg.dev/videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy  sha256:c13cf846e7f411a59ebe22c26990f9314e5d5611cff3acbfe16f30194089c4aa  2025-09-06T14:04:21  2025-09-06T14:04:21  995495206
lalamani1@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo %

```

Artifact Registry / Project: videoanalyzer-455321 / Location: us-central1 / Repository: springboot-docker-repo

Images for springboot-docker-repo

Name	Connection	Created	Updated
cloudrunspringbootdeploy	-	15 minutes ago	15 minutes ago

Digests for cloudrunspringbootdeploy

Name	Description	Tags	Created	Updated	Virtual size	Vulnerabilities
c13cf846e7f4	-	v1	15 minutes ago	15 minutes ago	949.4 MB	API disabled

STEP:6 Deploy to Cloud Run

Deploy the container directly from Artifact Registry

```
gcloud run deploy springboot-service --image=us-central1-docker.pkg.dev/
videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy:v1
--region=us-central1 --allow-unauthenticated
```

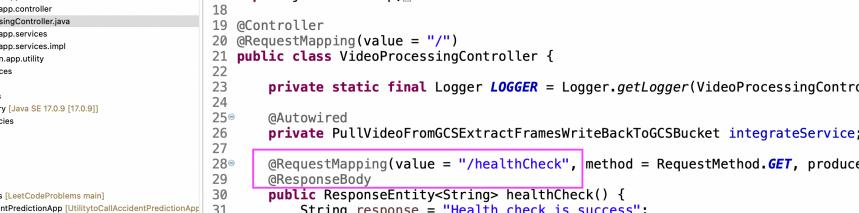
```
lalamanil@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo % gcloud run deploy springboot-service --image=us-central1-docker.pkg.dev/videoanalyzer-455321/springboot-docker-repo/cloudrunspringbootdeploy:v1 --region=us-central1 --allow-unauthenticated
Deploying container to Cloud Run service [springboot-service] in project [videoanalyzer-455321] region [us-central1]
✓ Deploying new service... Done.
  ✓ Creating Revision...
  ✓ Routing traffic...
  ✓ Setting IAM Policy...
Done.
Service [springboot-service] revision [springboot-service-00001-ksv] has been deployed and is serving 100 percent of traffic.
Service URL: https://springboot-service-852426054869.us-central1.run.app
lalamanil@Anils-MacBook-Pro IntegratingAccidentPredictionModelOfVideo %
```

The screenshot shows the Google Cloud Console interface for the project 'videoanalyzer-455321'. The URL is `console.cloud.google.com/run?project=videoanalyzer-455321`. The top navigation bar includes 'Google Cloud' and 'VideoAnalyzer' tabs, a search bar, and various navigation icons. The main content area is titled 'Cloud Run' and shows a table of services. The table has columns for Name, Deployment type, Req/sec, Region, Authentication, Ingress, Last deployed, and Deployed by. A single service named 'springboot-service' is listed, which is a Container deployment with 0 req/sec, in 'us-central1' region, with Public access and All ingress. It was last deployed 8 hours ago by 'lalamanilbabu'.

The screenshot shows the 'Service details' page for the 'springboot-service' in 'us-central1'. The URL is `console.cloud.google.com/run/detail/us-central1/springboot-service/observability/metrics?project=videoanalyzer-455321`. The top navigation bar includes 'Google Cloud' and 'VideoAnalyzer' tabs, a search bar, and various navigation icons. The main content area shows the 'Observability' tab selected. It displays metrics like Request count, Request latencies, and Annotations. The Request count chart shows values around 0.008/s, 0.006/s, and 0.004/s over UTC-4 time. The Request latencies chart shows values around 300ms and 200ms over UTC-4 time. Annotations show two entries: one at 50% and another at 95%.

Verify:

There is an endpoint /healthCheck defined in Spring boot application as shown in below screen shot



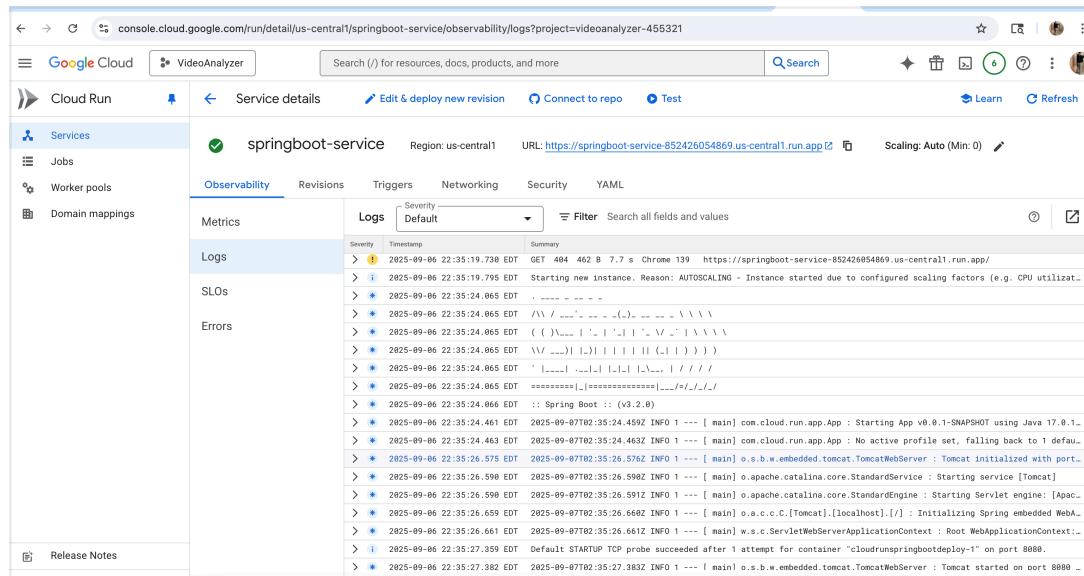
The screenshot shows a Java development environment with two tabs open: 'Dockerfile' and 'VideoProcessingController.java'. The 'VideoProcessingController.java' tab is the active one, displaying Java code for a controller class. The code includes annotations like @Controller and @RequestMapping, and methods for handling video processing requests. The left side of the interface shows a 'Package Explorer' with a tree view of the project structure, including source code, test resources, and configuration files like Dockerfile and pom.xml.

```
2| * @Author ANIL LALAM
3| package com.cloud.run.app.controller;
4|
5| import java.util.Map;
6|
7| @Controller
8| @RequestMapping(value = "/")
9| public class VideoProcessingController {
10|
11    private static final Logger LOGGER = Logger.getLogger(VideoProcessingController.class.getName());
12|
13    @Autowired
14    private PullVideoFromGCSExtractFramesWriteBackToGCSBucket integrateService;
15|
16    @RequestMapping(value = "/healthCheck", method = RequestMethod.GET, produces = MediaType.TEXT_PLAIN_VALUE)
17    @ResponseBody
18    public ResponseEntity<String> healthCheck() {
19        String response = "Health check is success";
20        return ResponseEntity.ok().body(response);
21    }
22|
23    @RequestMapping(value = "/processVideoToFrames", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_VALUE)
24    @ResponseBody
25    public ResponseEntity<Map<String, Object>> processVideoFrames(
26        @RequestParam(name = "bucketName", required = true) String bucketName,
27        @RequestParam(name = "name", required = true) String name) {
28        LOGGER.info("bucketName:" + bucketName + " name:" + name);
29        System.out.println(integrateService);
30        Map<String, Object> responseMap = integrateService.integrationLogic(bucketName, name);
31        return new ResponseEntity<Map<String, Object>>(responseMap, HttpStatus.OK);
32    }
33|
34}
35|
36}
37|
38}
39|
40}
41|
42}
43|
44}
45|
46}
47|
48}
```

Able to hit rest endpoint via browser



Below are the logs at Cloud Run in Google cloud console.



console.cloud.google.com/run/detail/us-central1/springboot-service/observability/logs?project=videoanalyzer-455321

Google Cloud VideoAnalyzer Search (/) for resources, docs, products, and more Q Search

Cloud Run Service details Edit & deploy new revision Connect to repo Test Learn Refresh

Services Jobs Worker pools Domain mappings

springboot-service Region: us-central1 URL: https://springboot-service-852426054869.us-central1.run.app Scaling: Auto (Min: 0)

Observability Revisions Triggers Networking Security YAML

Metrics Logs Default Filter Search all fields and values

Logs Severity Timestamp Summary

SLOs

Errors

Logs

Severity Timestamp Summary

2025-09-06 22:35:24.066 EDT :: Spring Boot :: (v3.2.0)

2025-09-06 22:35:24.461 EDT 2025-09-07T02:35:24.49Z INFO 1 --- [main] com.cloud.run.app.App : Starting App v0.0.1-SNAPSHOT using Java 17.0.1...

2025-09-06 22:35:24.463 EDT 2025-09-07T02:35:24.49Z INFO 1 --- [main] com.cloud.run.app.App : No active profile set, falling back to 1 defau...

2025-09-06 22:35:26.575 EDT 2025-09-07T02:35:26.576Z INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port...

2025-09-06 22:35:26.598 EDT 2025-09-07T02:35:26.598Z INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]

2025-09-06 22:35:26.598 EDT 2025-09-07T02:35:26.591Z INFO 1 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apac...

2025-09-06 22:35:26.659 EDT 2025-09-07T02:35:26.660Z INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebA...

2025-09-06 22:35:26.661 EDT 2025-09-07T02:35:26.661Z INFO 1 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:...

2025-09-06 22:35:27.359 EDT Default STARTUP TCP probe succeeded after 1 attempt for container "cloudrunspringbootdeploy-1" on port 8088.

2025-09-06 22:35:27.382 EDT 2025-09-07T02:35:27.383Z INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 -

2025-09-06 22:35:27.454 EDT 2025-09-07T02:35:27.45Z INFO 1 --- [main] com.cloud.run.app.App : Started App in 3.89 seconds (process running -

2025-09-06 22:35:27.505 EDT 2025-09-07T02:35:27.506Z INFO 1 --- [nio-8088-exec-5] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring Dis...

2025-09-06 22:35:27.505 EDT 2025-09-07T02:35:27.506Z INFO 1 --- [nio-8088-exec-5] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'di...

2025-09-06 22:35:27.508 EDT 2025-09-07T02:35:27.508Z INFO 1 --- [nio-8088-exec-5] o.s.web.servlet.DispatcherServlet : Completed initialization...

2025-09-06 22:35:27.718 EDT GET 404 266 B 106 ms Chrome 139 https://springboot-service-852426054869.us-central1.run.app/favicon.ico

2025-09-06 22:35:48.529 EDT GET 200 118 B 12 ms Chrome 139 https://springboot-service-852426054869.us-central1.run.app/healthCheck

Release Notes