

Spring on Google Cloud Platform



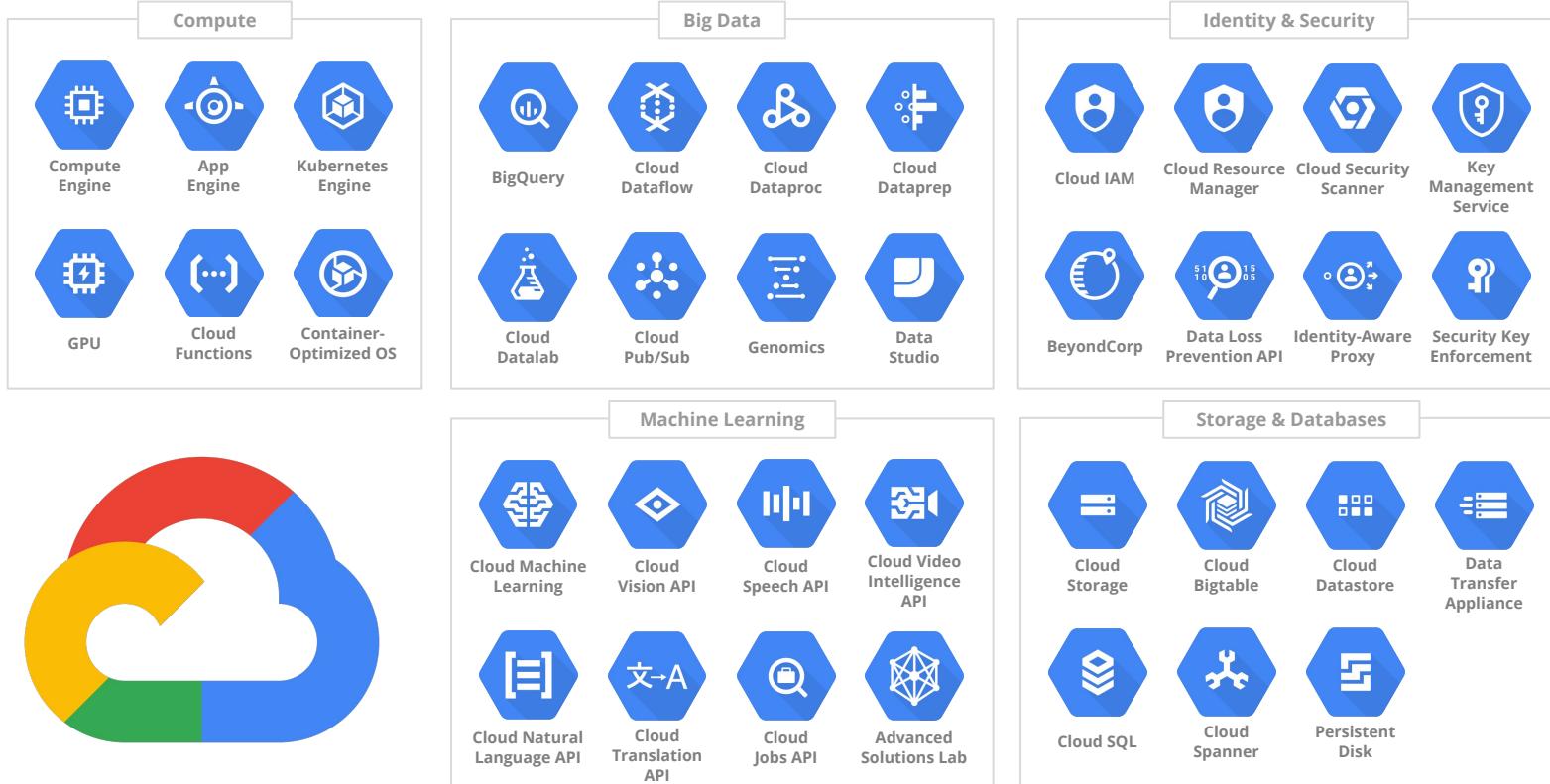
Google Cloud Platform

Ray Tsang

Developer Advocate
Google Cloud Platform

@saturnism saturnism.me





Amazing Technology!

Use Spring on Google Cloud Platform as easily and seamlessly as possible

Framework

IDE and Tooling

Deployment

Monitoring/Diagnostics

Goals

Provide integration between GCP services and Spring

Meet developers where they are

Cut down on boilerplate code



+



Google Cloud Platform

Collaboration with Pivotal

Developed GitHub in the [spring-cloud-gcp](#)
repository under the Spring Cloud organization
owned by Pivotal.

Close collaboration between Google and
Pivotal engineers.



Spring Boot Starters for...

Feature	GCP Service	Spring Framework
Configuration	Cloud Runtime Config	Spring Cloud Config
Messaging	Cloud Pub/Sub	Spring Integration Spring Cloud Stream Spring Dataflow
Database	Cloud SQL Cloud Spanner	Spring Data Spring Data Spanner
Storage	Cloud Storage	Spring Resource
Logging	Stackdriver Logging	Logback
Trace	Stackdriver Trace	Spring Cloud Sleuth

SPRING INITIALIZR

bootstrap your application now

Generate a with and Spring Boot

Project Metadata

Artifact coordinates

Group

Artifact

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

GCP Support

Support for Google Cloud Platform services

⌘ + ↵

Don't know what to look for? Want more options? [Switch to the full version.](#)

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-gcp-dependencies</artifactId>
      <version>${spring-cloud-gcp.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```



Spring Boot App



Cloud SQL database

Managed MySQL and PostgreSQL



Cloud SQL

Scale out with read replicas

High availability (HA) option

Per-minute billing

Instance sizes to fit any budget

Granular access control using IAM roles

pom.xml

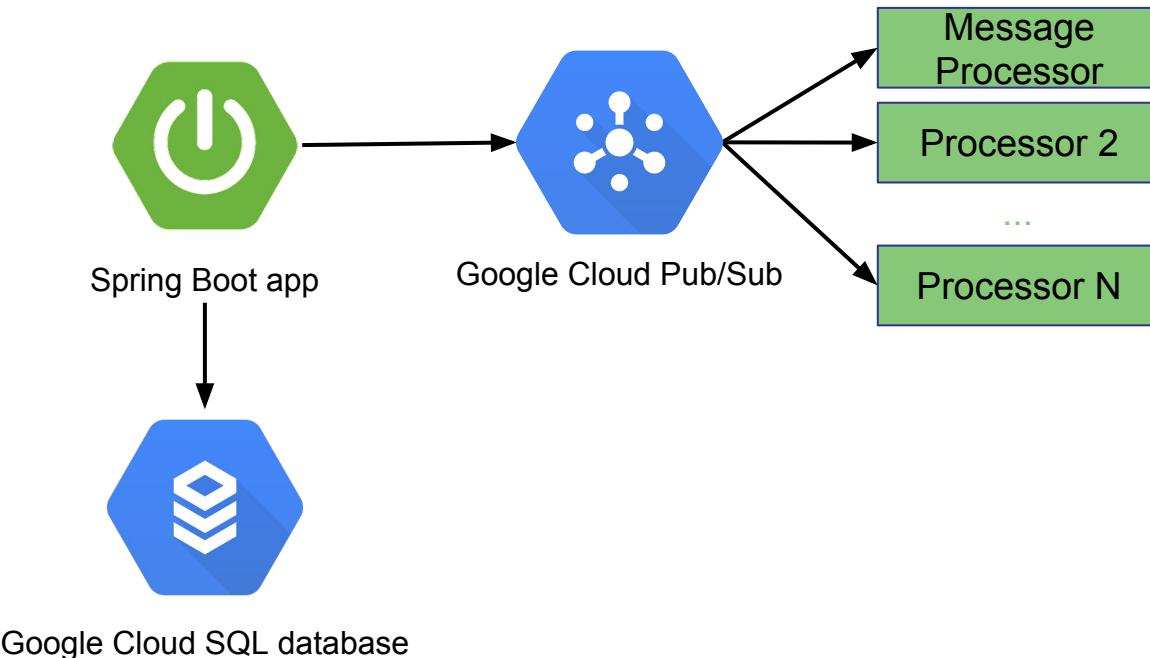
```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-gcp-starter-sql-mysql</artifactId>
</dependency>
```

application.properties

```
spring.cloud.gcp.sql.instance-connection-name=my-gcp-project:us-central1:petclinic
spring.cloud.gcp.sql.database-name=petclinic_db
```

Configures Datasource, JDBC URL, Certificates

Establishes secured connections





Cloud
Pub/Sub

Managed Global Messaging Middleware

Capture, pass messages across services

At-least- once delivery semantics

Configurable message retention policies

Push/Pull message delivery patterns

Create a topic

A topic forwards messages from publishers to subscribers.

Name ?

[CANCEL](#) [CREATE](#)

pom.xml

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-gcp-starter-pubsub</artifactId>
</dependency>
```

Code

```
@Autowired PubSubTemplate pubSubTemplate;

public void hello(String name) {
    pubSubTemplate.publish("my-topic", name + " said Hi");
}
```

[←](#) Create a subscription

A subscription directs messages on a topic to subscribers. Messages can be pushed to subscribers immediately, or subscribers can pull messages as needed.

Topic

projects/wise-coyote-827/topics/my-topic

Subscription name [?](#)

projects/wise-coyote-827/subscriptions/ processors

Delivery Type [?](#)

Pull

Push into an endpoint url [?](#)

Code

```
pubSubTemplate.subscribe("processors", (msg, ackReply) -> {
    log.info(msg.getData().toStringUtf8());
    ackReply.ack();
});
```

Also Integrated Pub/Sub with...

Spring Framework	Use Case
Spring Integration	Enterprise Integration Pattern Middleware-Agnostic Code
Spring Cloud Stream	Event-driven Microservices Reactive Programming Middleware-Agnostic Code

Scalable mission-critical RDBMS



Cloud
Spanner

Relational semantics

Schemas, ACID transactions, SQL query

Scale horizontally

Automatic, synchronous replication

99.999% Availability

pom.xml

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-gcp-starter-data-spanner</artifactId>
</dependency>
```

Code

```
@Table
public class Person {
    @PrimaryKey
    private long id;
    private String id;
    ...
}
```



Code

```
@RepositoryRestResource
public interface PersonRestRepository extends SpannerRepository<Person, Long> {
    public List<Person> findPersonByName(String name);

    @Query("SELECT * FROM person WHERE ...")
    public List<Event> findPersonWithQuery(String location);
}
```

Observability, Correlated Logs with Sleuth and Logback

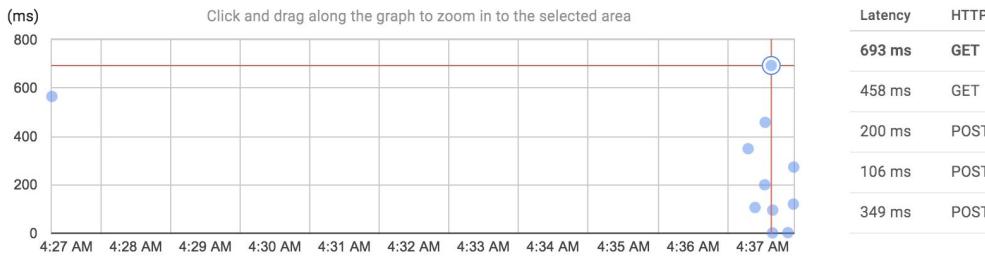
pom.xml

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-gcp-starter-logging</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-gcp-starter-trace</artifactId>
</dependency>
```

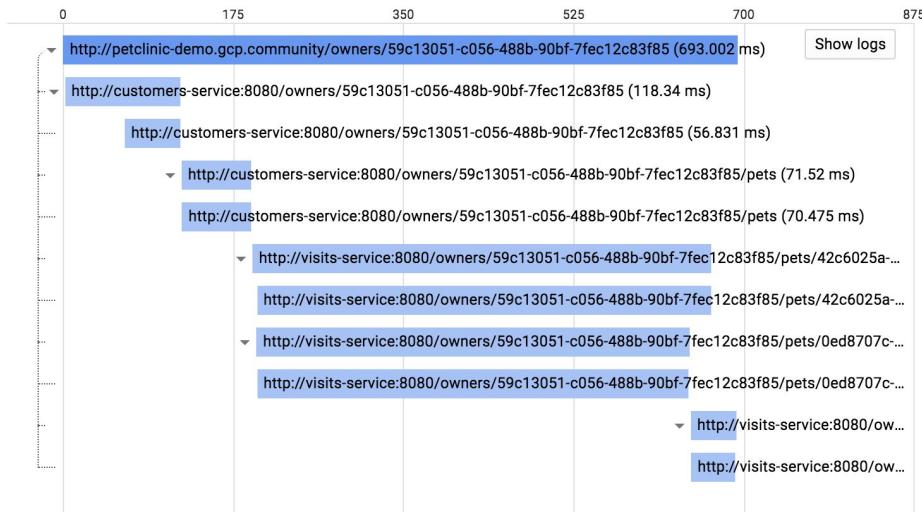
Log Appender

Direct to Stackdriver Logging

Or, Output as XML



Timeline





```
1 timestamp>="2018-06-19T08:37:27.643000Z" timestamp<="2018-06-19T08:37:48.336000Z" (trace="projects/kubernetes-monitor  
labels."appengine.googleapis.com/trace_id"="5b28c0d1efd48edad79af99cd5a2578f")
```

Submit Filter

No limit ▾

Jump to now ▾

Showing logs from all time (EDT)



- ▶ i 2018-06-19 04:37:37.644 EDT Getting Owner: 59c13051-c056-488b-90bf-7fec12c83f85
- ▶ i 2018-06-19 04:37:37.707 EDT Getting Owner ID: 59c13051-c056-488b-90bf-7fec12c83f85
- ▶ i 2018-06-19 04:37:37.766 EDT Getting Pets for Owner: 59c13051-c056-488b-90bf-7fec12c83f85
- ▶ i 2018-06-19 04:37:37.836 EDT Getting Pets for Owner: 59c13051-c056-488b-90bf-7fec12c83f85



No newer entries found matching current filter.

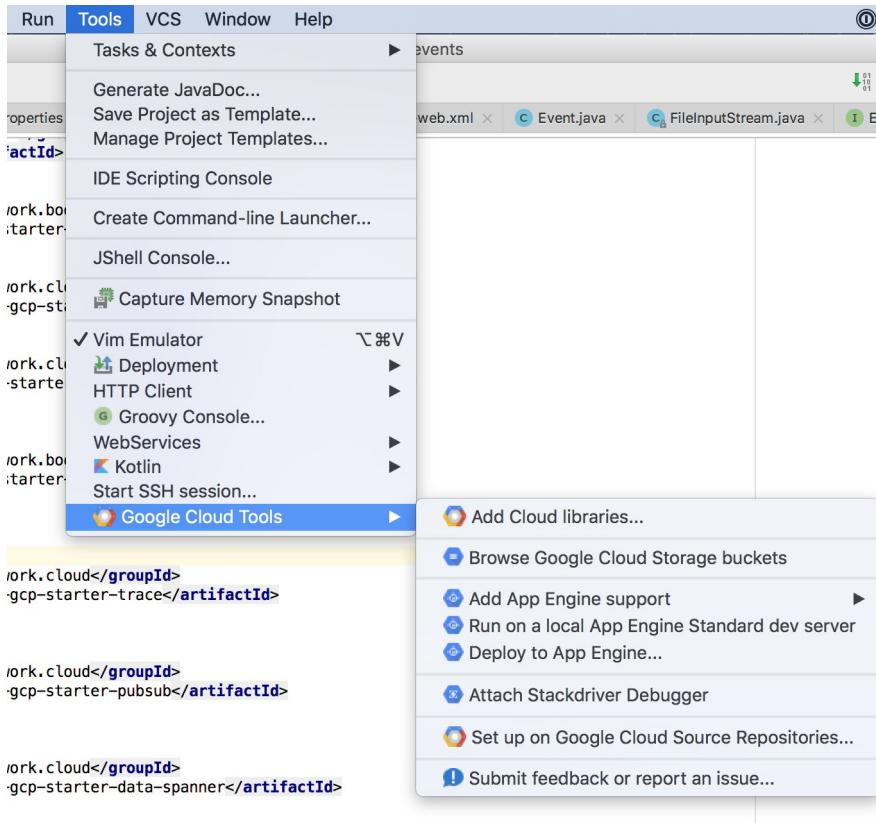
Load newer logs

Spring Boot Starters for...

Feature	GCP Service	Spring Framework
Configuration	Cloud Runtime Config	Spring Cloud Config
Messaging	Cloud Pub/Sub	Spring Integration Spring Cloud Stream Spring Dataflow
Database	Cloud SQL Cloud Spanner	Spring Data Spring Data Spanner
Storage	Cloud Storage	Spring Resource
Logging	Stackdriver Logging	Logback
Trace	Stackdriver Trace	Spring Cloud Sleuth

IntelliJ and Eclipse Plugins

Maven and Gradle Plugins



Add Google Cloud Library to Module:

 events

GCP Project: [DEMO - GKE Primary](#)

Google Cloud Java:

(BOM) 0.53.0-alpha

- Cloud DNS
 - Cloud Data Loss Prevention
 - Cloud Datastore
 - Cloud Firestore
 - Cloud Natural Language
 - Cloud OS Login
 - Cloud Pub/Sub**
 - Cloud Redis
 - Cloud Resource Manager
 - Cloud Security Scanner
 - Cloud Spanner
 - Cloud Speech
 - Cloud Storage



Cloud Pub/Sub

Cloud Pub/Sub provides reliable, many-to-many communication between applications. Publisher applications can send messages to topics, and subscriber applications can subscribe to that topic to receive the messages.

Version: 1.35.0

[Documentation](#) | [Source](#) | [API Reference](#)

Google Cloud Platform API Management

- ✓ Enable the API on the Google Cloud Platform

Automatically adds dependencies to POM

pom.xml

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

Code

```
@Bean
public ImageAnnotatorClient imageAnnotatorClient(
    CredentialsProvider credentialsProvider) throws IOException {
    ImageAnnotatorSettings clientSettings = ImageAnnotatorSettings.newBuilder()
        .setCredentialsProvider(credentialsProvider)
        .build();
    return ImageAnnotatorClient.create(clientSettings);
}
```



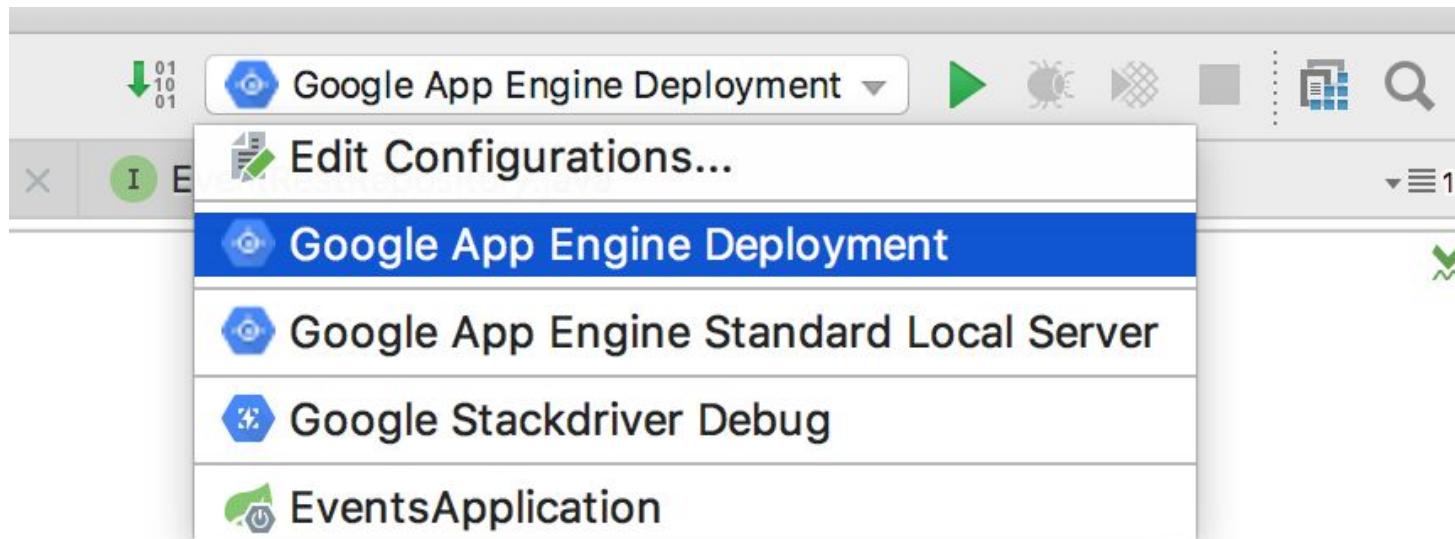
App Engine
Standard



Kubernetes
Engine



Compute
Engine



pom.xml

```
<plugin>
  <groupId>com.google.cloud.tools</groupId>
  <artifactId>appengine-maven-plugin</artifactId>
  <version>1.3.1</version>
</plugin>
```

Deploy

```
$ mvn appengine:deploy
```

Jib - building container images

<https://github.com/GoogleContainerTools/jib>

Fast - Automatically split application into layers based on application dependency.

Reproducible - Rebuilding your container image with the same contents always generates the same image.

Daemonless - Reduce CLI dependencies. Build container image from within Maven or Gradle and push to any registry of your choice. No more writing Dockerfiles and calling docker build/push.

Small - Uses Google's Distroless base image,

pom.xml

```
<plugin>
  <groupId>com.google.cloud.tools</groupId>
  <artifactId>jib-maven-plugin</artifactId>
  <version>...</version>
  <configuration>
    <to>
      <image>my-spring-app</image>
    </to>
  </configuration>
</plugin>
```

Deploy

```
$ mvn compile jib:build
```

Pivotal Cloud Foundry Support



Pivotal **Cloud Foundry**[®]



Google Cloud Platform

Pivotal Cloud Foundry Support

Native - Google and Pivotal engineers built support for PCF on GCP. Use things like global load balancing and pre-emptible VMs to scale applications or reduce cost.

Integrated - The GCP Service Broker lets developers provision and use Google services in their apps without manually creating and managing resources in the console.

Observable - Stackdriver debugging, monitoring, and logging make it easy to observe your Spring applications deployed to PCF on GCP.

GCP Service Broker for Cloud Foundry



Management

Compute

Storage

Networking

Data

Machine Learning

```
cf create-service SERVICE PLAN SERVICE_INSTANCE [-c PARAMETERS_AS_JSON] [-t TAGS]  
# generated topic name, no subscription  
$ cf create-service pubsub default orders  
# generated topic name, with a subscription  
$ cf create-service pubsub default orders -c \  
'{"topic-name": "orders", "subscription-name": "accounting"}'
```

```
cf bind-service APP_NAME SERVICE_INSTANCE [-c PARAMETERS_AS_JSON] [--binding-name  
BINDING_NAME]  
$ cf bind-service accounting-database orders -c '{"role": "pubsub.subscriber"}'  
$ cf bind-service website orders -c '{"role": "pubsub.publisher"}'
```

Spring Cloud GCP

Automatically Discovers

VCAP_SERVICES Environment Variable

Configures Credentials

Prometheus Monitoring with Stackdriver

pom.xml

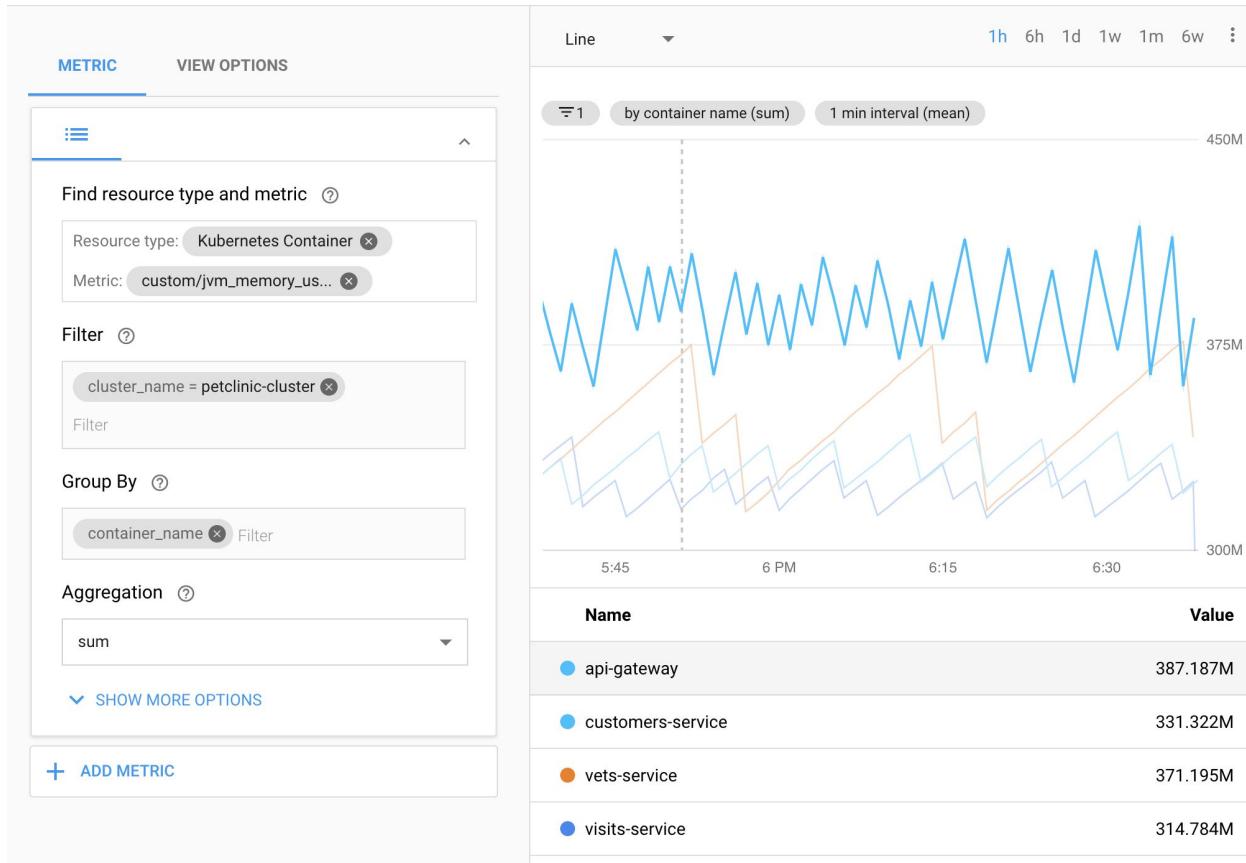
```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
<dependency>
    <groupId>io.micrometer</groupId>
    <artifactId>micrometer-registry-prometheus</artifactId>
    <scope>runtime</scope>
</dependency>
```

Prometheus Endpoint

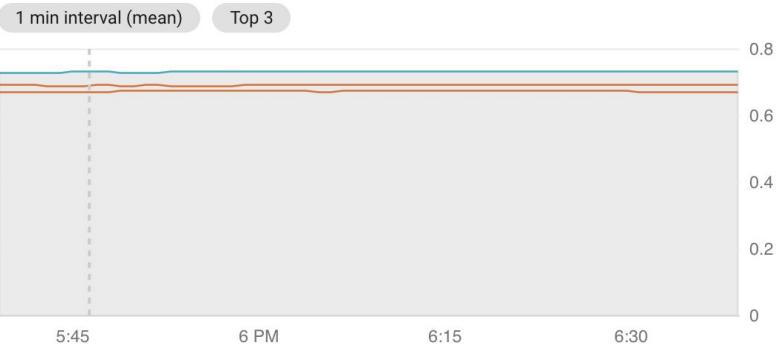
```
$ curl http://localhost:8080/actuator/prometheus
```

```
jvm_gc_max_data_size_bytes 8.9653248E7
tomcat_global_request_seconds_count{name="http-nio-8080",} 322145.0
tomcat_global_request_seconds_sum{name="http-nio-8080",} 748.139
process_uptime_seconds 26874.681
tomcat_global_error_total{name="http-nio-8080",} 1.0
jvm_buffer_memory_used_bytes{id="direct",} 294913.0
jvm_buffer_memory_used_bytes{id="mapped",} 0.0
tomcat_servlet_request_max_seconds{name="default",} 0.0
system_cpu_usage 0.06456190138346932
process_files_max 1048576.0
http_server_requests_seconds_count{exception="None",method="GET",status="200",uri="/**",} 322111.0
http_server_requests_seconds_sum{exception="None",method="GET",status="200",uri="/**",} 527.459778932
http_server_requests_seconds_count{exception="None",method="GET",status="404",uri="/webjars/**",} 1.0
http_server_requests_seconds_sum{exception="None",method="GET",status="404",uri="/webjars/**",} 0.026788256
http_server_requests_seconds_count{exception="None",method="GET",status="304",uri="/**",} 33.0
http_server_requests_seconds_sum{exception="None",method="GET",status="304",uri="/**",} 0.171848745
http_server_requests_seconds_count{exception="None",method="GET",status="200",uri="/api/gateway/owners/{ownerId}",} 3.0
http_server_requests_seconds_sum{exception="None",method="GET",status="200",uri="/api/gateway/owners/{ownerId}",} 1.247613785
http_server_requests_seconds_max{exception="None",method="GET",status="200",uri="/**",} 0.073825406
http_server_requests_seconds_max{exception="None",method="GET",status="404",uri="/webjars/**",} 0.0
http_server_requests_seconds_max{exception="None",method="GET",status="304",uri="/**",} 0.0
http_server_requests_seconds_max{exception="None",method="GET",status="200",uri="/api/gateway/owners/{ownerId}",} 0.0
system_cpu_count 2.0
tomcat_threads_busy{name="http-nio-8080",} 0.0
tomcat_sessions_active_max 0.0
jvm_threads_live 69.0
tomcat_servlet_request_seconds_count{name="default",} 0.0
tomcat_servlet_request_seconds_sum{name="default",} 0.0
jvm_memory_committed_bytes{area="nonheap",id="Code Cache",} 4.2205184E7
jvm_memory_committed_bytes{area="nonheap",id="Metaspace",} 7.3269248E7
jvm_memory_committed_bytes{area="nonheap",id="Compressed Class Space",} 8781824.0
jvm_memory_committed_bytes{area="heap",id="PS Eden Space",} 4.0370176E7
jvm_memory_committed_bytes{area="heap",id="PS Survivor Space",} 2097152.0
jvm_memory_committed_bytes{area="heap",id="PS Old Gen",} 6.029312E7
jvm_memory_max_bytes{area="nonheap",id="Code Cache",} 2.5165824E8
jvm_memory_max_bytes{area="nonheap",id="Metaspace",} -1.0
jvm_memory_max_bytes{area="nonheap",id="Compressed Class Space",} 1.073741824E9
jvm_memory_max_bytes{area="heap",id="PS Eden Space",} 4.0370176E7
jvm_memory_max_bytes{area="heap",id="PS Survivor Space",} 2097152.0
jvm_memory_max_bytes{area="heap",id="PS Old Gen",} 8.9653248E7
...
...
```

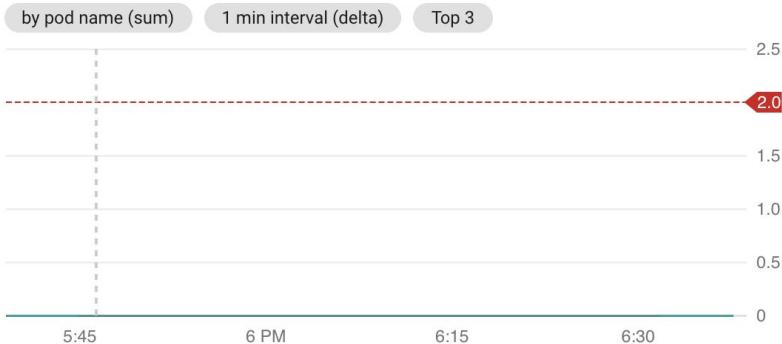
Metrics Explorer



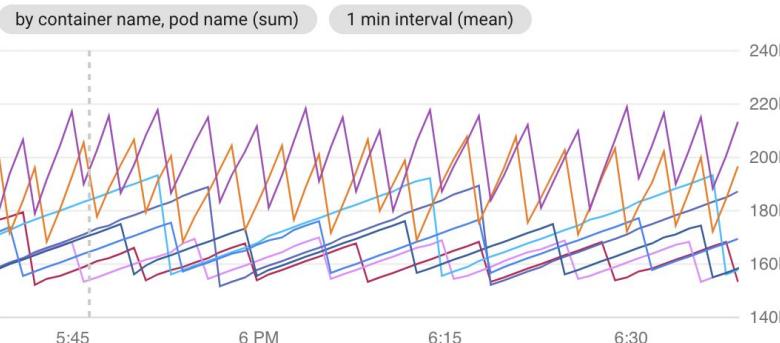
Memory limit utilization by Container



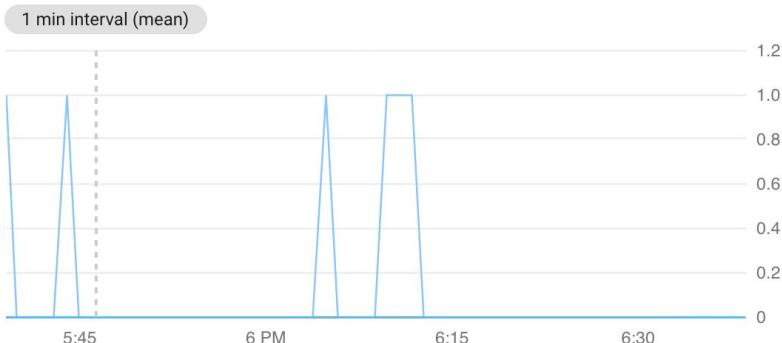
Restart Counts



JVM Heap Usage



JVM Busy Threads



Production Debugging and Profiling

Download the Agents

Add to Startup Command Line:

For Profiler:

```
$ java -agentpath:/opt/profiler/profiler_java_agent.so=... -jar myapp.jar
```

For Debugger:

```
$ java -agentpath:/opt/debugger/cdbg_java_agent.so -jar myapp.jar
```

What's taking most CPU time?

Service: spring-petclinic-api-gateway ▾ Profile type: CPU ▾ Zone name: All zones ▾ Version: All versions ▾

Metric: CPU time ▾ Add profile data filter...

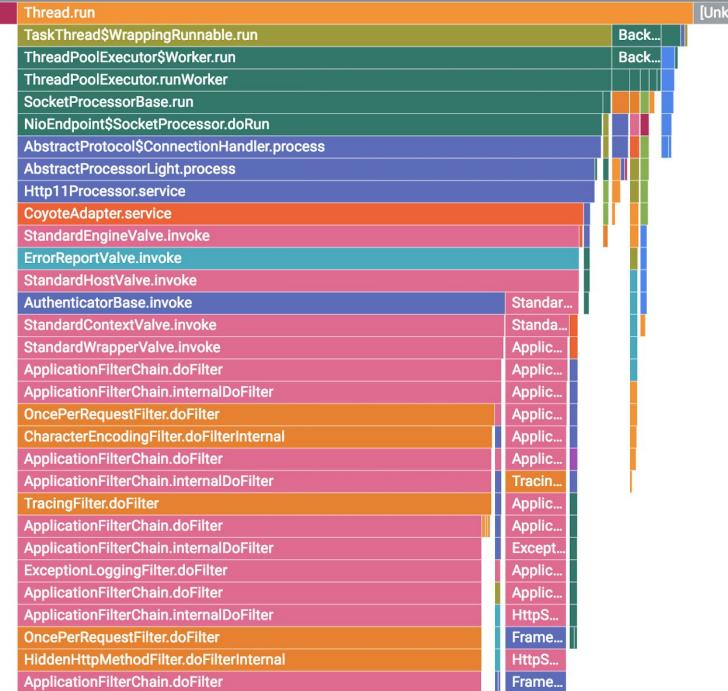
Name: root (100%, averaged over 250 profiles)

File: no file

Values: total: 343.52 ms (100%)

100%, averaged over 250 profiles

[libjvm.so]



Forgot a Log Message?

```
51 final OwnerDetails owner = customersServiceClient.getOwner(ownerId);  
52  
53 logger.info("Getting Pets for Owner: {}", ownerId);
```

```
if (true) logpoint("I wonder what's happening here, {owner.lastName}")
```



Info ▾

Cancel

Add

```
54 supplyVisits(owner, visitsServiceClient.getVisitsForPets(owner.getPets(), ownerId));  
55 return owner;
```



- ▶ **i** 2018-07-11 17:31:49.443 EDT Getting Pets for Owner: 1
- ▶ **i** 2018-07-11 17:33:47.764 EDT Getting Owner: 1
- ▶ **i** 2018-07-11 17:33:47.978 EDT Getting Pets for Owner: 1
- ▶ **i** 2018-07-11 17:34:26.747 EDT Getting Owner: 1
- ▶ **i** 2018-07-11 17:34:26.810 EDT Getting Pets for Owner: 1
- ▶ **i** 2018-07-11 17:34:26.850 EDT LOGPOINT: I wonder what's happening here, Tsang

Debug Production Code, Live!

saturnism/spring-petclinic-gc...

Type a file name

```
github.com/  
  ► .mvn/wrapper  
  ► db  
  ► docker  
  ► docs  
  ► istio  
  ► kubernetes  
  ▼ spring-petclinic-api-gateway  
    ▼ src/main  
      ▼ java/org/springframework/samples/pet/  
        ► application  
        ▼ boundary/web  
          ApiGatewayController.java  
        ► infrastructure/config  
          ApiGatewayApplication.java  
      ► less  
      ► resources  
      ► wro  
    .factorypath  
    .gitignore  
    pom.xml  
  ► spring-petclinic-customers-service  
  ► spring-petclinic-vets-service  
  ► spring-petclinic-visits-service  
    .gitignore  
    .travis.yml  
    LICENSE  
    README.md  
    docker-compose.yml  
    lombok.config  
    mvnw
```

```
30 import java.util.List;  
31 import java.util.Map;  
32 import java.util.Optional;  
33  
34 import static java.util.Collections.emptyList;  
35  
36 /**  
 * @author Maciej Szarlinski  
 */  
37 @RestController  
38 @RequiredArgsConstructor  
39 public class ApiGatewayController {  
40     static private final Logger logger = LoggerFactory.getLogger(ApiGatewayController.class);  
41  
42     private final CustomersServiceClient customersServiceClient;  
43  
44     private final VisitsServiceClient visitsServiceClient;  
45  
46     @GetMapping(value = "owners/{ownerId}")  
47     public OwnerDetails getOwnerDetails(@PathVariable String ownerId) {  
48         logger.info("Getting Owner: {}", ownerId);  
49         final OwnerDetails owner = customersServiceClient.getOwner(ownerId);  
50  
51         logger.info("Getting Pets for Owner: {}", ownerId);  
52         supplyVisits(owner, visitsServiceClient.getVisitsForPets(owner.getPets(), ownerId));  
53         return owner;  
54     }  
55  
56     private void supplyVisits(final OwnerDetails owner, final Map<String, List<VisitDetails>> visitsMapping) {  
57         owner.getPets().forEach(pet ->  
58             pet.getVisits().addAll(  
59                 Optional.ofNullable(visitsMapping.get(pet.getId()))  
60                     .orElse(emptyList()));  
61     }  
62  
63 }  
64 }
```



Owner Information

Name	Ray Tsang
Address	111 8th Ave.
City	New York
Telephone	1231231230

[Edit Owner](#)[Add New Pet](#)

Pets and Visits

[Delete Pet](#) | [Edit Pet](#)

Name **Harley**
Birth Date 2007 Dec 03
Type DOG

[Add Visit](#)

Visit Date	Description
2018 Apr 02	Routine check
2018 Apr 23	Follow up
2018 May 24	toothache

[Delete Pet](#) | [Edit Pet](#)

Name **Ally**
Birth Date 1990 Oct 19
Type CAT

[Add Visit](#)

Visit Date	Description
2018 Apr 22	Dental
2018 Apr 23	Accident
2018 Apr 30	Surgery

ApiGatewayController.java:53



Condition: (Optional)

Type a condition

Expressions: (Optional)

Type an expression

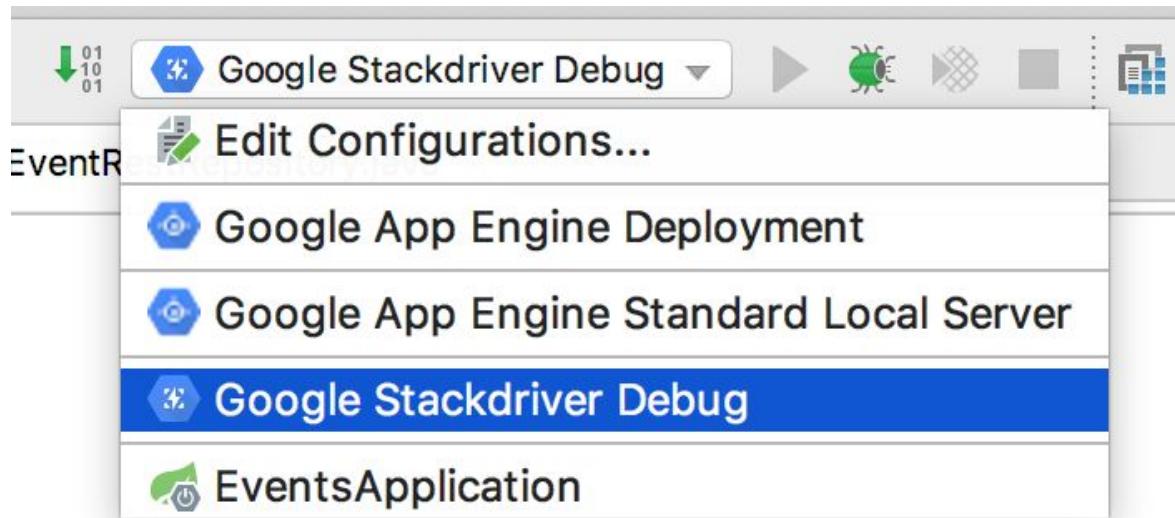
Variables

▶ this	
ownerId	"1"
▼ owner	
id	"1"
firstName	"Ray"
lastName	"Tsang"
address	"111 8th Ave."
city	"New York"
telephone	"1231231230"
▶ pets	

2018-07-11 (17:31:49)

Call Stack

```
org.springframework.samples.petclinic.api.boundary.web.ApiGatewayController.ge... ApiGatewayController.java:53
sun.reflect.NativeMethodAccessorImpl.invoke0   NativeMethodAccessorImpl.java:-1
sun.reflect.NativeMethodAccessorImpl.invoke   NativeMethodAccessorImpl.java:62
sun.reflect.DelegatingMethodAccessorImpl.invoke DelegatingMethodAccessorImpl.java:43
java.lang.reflect.Method.invoke               Method.java:498
org.springframework.web.method.support.InvocableHandlerMethod.invoke           InvocableHandlerMethod.java:209
org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest InvocableHandlerMethod.java:136
org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerM... ServletInvocableHandlerMethod.java:102
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAda... RequestMappingHandlerAdapter.java:877
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAda... RequestMappingHandlerAdapter.java:783
org.springframework.web.servlet.method.AbstractHandlerMethodAdapter.handle     AbstractHandlerMethodAdapter.java:87
org.springframework.web.servlet.DispatcherServlet.doDispatch                   DispatcherServlet.java:991
|>
```



PCF Buildpack with GCP Service Broker

Automatically adds Debugger Agent on `cf push`!

<https://github.com/cloudfoundry/java-buildpack>

- o [Dyadic EKM Security Provider \(Configuration\)](#)
- o [Dynatrace Appmon Agent \(Configuration\)](#)
- o [Dynatrace SaaS/Managed OneAgent \(Configuration\)](#)
- o [Google Stackdriver Debugger \(Configuration\)](#)
- o [Introscope Agent \(Configuration\)](#)
- o [JaCoCo Agent \(Configuration\)](#)
- o [Java Memory Assistant \(Configuration\)](#)
- o [Java Options \(Configuration\)](#)
- o [JRebel Agent \(Configuration\)](#)



Framework - Spring Boot Support

IDE and Tooling - IntelliJ, Eclipse, Maven, Gradle

Deployment - Container, App Engine, PCF, ...

Monitoring/Diagnostics - Debug, Profile, Logging, Error Reporting, Prometheus Monitoring, ...

End-to-end Developer Experience

Project Home <https://cloud.spring.io/spring-cloud-gcp/>

Documentation <https://docs.spring.io/spring-cloud-gcp/docs/>

Source and Samples <https://github.com/spring-cloud/spring-cloud-gcp/>

Short Code Labs <http://g.co/codelabs/spring>

Long Code Lab <http://bit.ly/spring-gcp-lab>

Recorded Conference Talks

- [Power of Google Cloud Platform with Spring Cloud GCP](#)
- [Cloud Native with Spring Boot on Google Cloud Platform](#)

Thanks!

cloud.spring.io/spring-cloud-gcp

<http://bit.ly/spring-gcp-lab>

@saturnism



Google Cloud Platform