

Basic Details

Name: Sanskar Gupta

Email: snskr95@gmail.com, sanskar@kth.se

Github: [Sanskar95](https://github.com/Sanskar95)

Location: Stockholm, Sweden

Resume: [Link](#)

Open source contributions:

<https://www.npmjs.com/package/simple-toast-notification>

<https://www.npmjs.com/package/react-site-announcement>

<https://github.com/covid19india/covid19india-react/pull/891>

<https://github.com/mui-org/material-ui/pull/22021>

Motivation

Google summer of code is a platform where one doesn't get to apply his existing skills but also to learn something new. Being an engineer it's my sole motive to make stuff that people actually use and benefit from. There is no better place to where this aim can be fulfilled, as people around the world would be using it.

Also GSOC can be attributed as one of the best professional experiences a student can have. I really want to work with like-minded people and benefit from them as I feel actually implementing stuff collaboratively is the best way to get your hand comfortable on any skill.

It's analogous to driving the car on an empty highway compared to driving the car in Bangalore city (Highest traffic density city in India) traffic.

If given a chance this year, I will be definitely contributing to GSOC every year till I am pursuing any kind of studies.

I feel GFOSS can provide me the perfect project that I have been craving of doing since a long time.

The project that I am super interested in is enhancing Content Management System where I will be able to work on frontend and on some backend and infra-related features. Being a full stack developer gives me immense joy to develop a feature end to end, right from UI to its deployment and maintenance.

I am just expecting lots of learning and an understanding of dynamics of open source world during the course of 10 weeks.

Project Details

Current status of Project:

As far as current status of project is concerned it contains rest endpoints for creating , deleting, caching and locking a resource.

A resource can be image , video or any html file.

Springboot is used a backend web development framework and MariaDB for persistent storage solution.

The application supports features like Versioning, Distributed Caching, Distributed Locking, Distributed Unique ID generator, Multi-node support for scaling.

The application can be locally setup by pulling up the soccer image and running it as a container assuming a running MariaDB container.

Problem Statement:

P1)The problem statement includes implementing a front end from scratch for Administrators and Content Creators/Consumers to easily use, manage and monitor this application through simple and secure flows.

How I plan to tackle this problem?

The frontend will comprise a dashboard that will look like [this](#).

The webapp will have role based logins i.e user will be served different set of features based on whether he is the admin , creator or just a general user.

An admin would be able to monitor the application with metrics such as disk usage , IO, CPU usage etc.

A creator will be able to create and delete resources while a general user will be able to view the resources.

I plan to use react as front end framework owing to its performance attributed to virtual DOM creation, fast rendering and a decently organised directory structure.

Redux will be used for as a global state management solution. An example of global state is current logged in user details, user preferences while navigating through the app.

Axios will be used for all the REST calls which is a promise-based HTTP client.

Babel for transpiling the code to ES5 so that it can be used in all the browsers and webpack for bundling assets.

Protected routes of react router will be leveraged to prevent accessing screens by another user who is not authenticated

Something like:

```
<Router>
  <Switch>
    <Route path="/" exact>
      {isAuthenticated ? (
        <Redirect to="/admin-screen" />
      ) : (
        <div>
          <h1>CommonPage</h1>
          <Link to="/admin-screen">Go to secret</Link>
          <br></br>
          <button onClick={login}>Log in</button>
        </div>
      )}
    </Route>
  </Switch>
  <ProtectedRoute
```

```
    isAuthenticated={isAuthenticated}
    path="/admin-screen"
    logout={logout}
    component={Secret}
  />
  <Route path="*">
    <div>404 Not found </div>
  </Route>
</Switch>
</Router>
```

Note that the flag `isAuthenticated` can be stored in redux store for future comparisons
Also frontend will run as a separate deployment.
Prometheus metrics will be queried into the webapp using

<https://github.com/samber/prometheus-query-js>

P2)It also includes securing endpoints i.e making the endpoints role specific , for example user with consumer role wont be able to access the create/delete resource endpoint.

How I plan to tackle this problem?

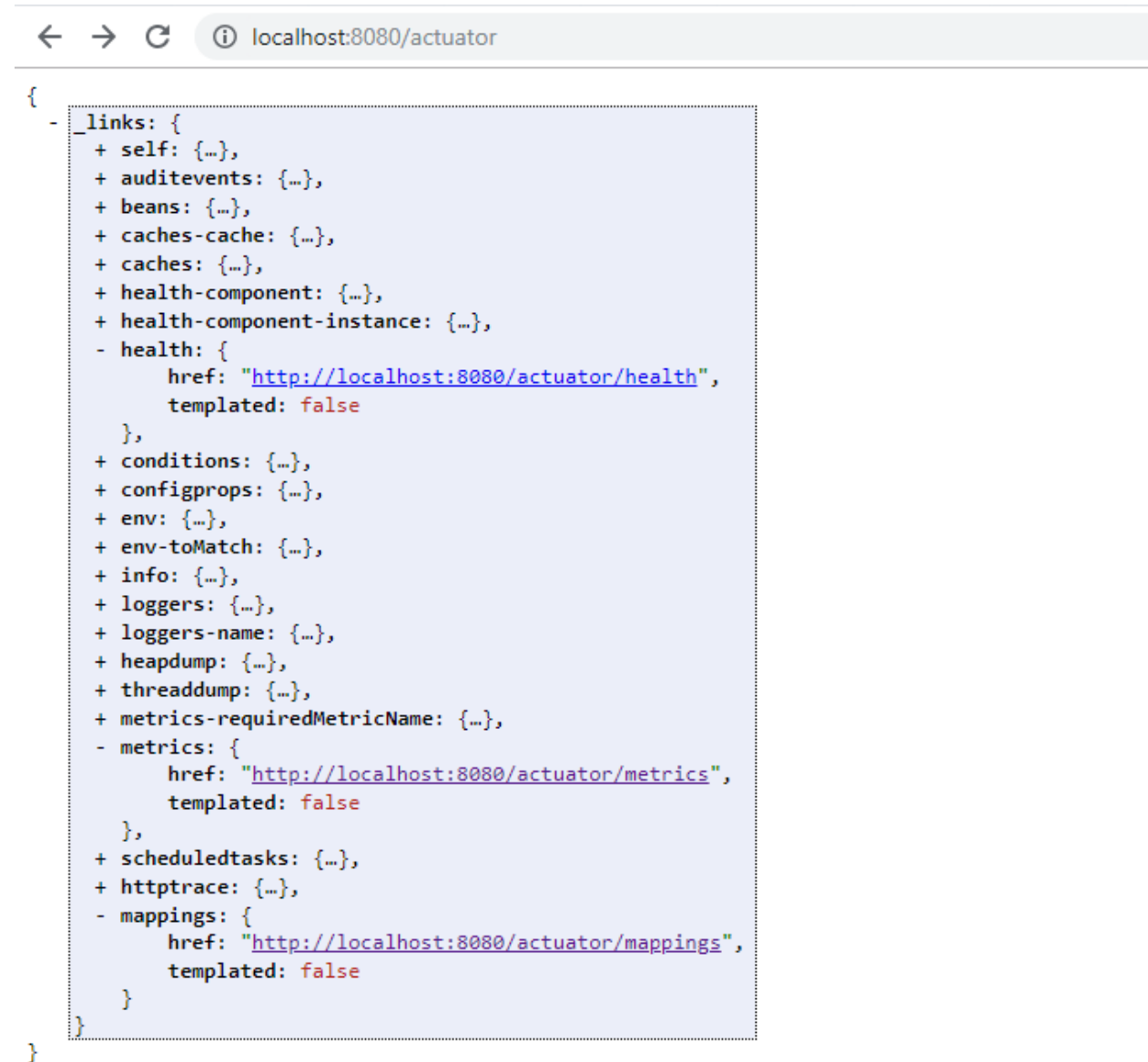
To solve this problem I plan to configure spring security by setting up required roles and permissions.

The permissions can either be configured in `AppicationSecurityConfig` or in controller classes using annotations.

P3)Implement a monitoring system in the kubernetes cluster where application is deployed for admin to see some critical metrics.

How I plan to tackle this problem?

I see the application does not have “spring-boot-starter-actuator” dependency so it can be included that will provide us with an /actuator endpoint whose response will look like:



```
{
  - _links: {
    + self: {...},
    + auditevents: {...},
    + beans: {...},
    + caches-cache: {...},
    + caches: {...},
    + health-component: {...},
    + health-component-instance: {...},
    - health: {
      href: "http://localhost:8080/actuator/health",
      templated: false
    },
    + conditions: {...},
    + configprops: {...},
    + env: {...},
    + env-toMatch: {...},
    + info: {...},
    + loggers: {...},
    + loggers-name: {...},
    + heapdump: {...},
    + threaddump: {...},
    + metrics-requiredMetricName: {...},
    - metrics: {
      href: "http://localhost:8080/actuator/metrics",
      templated: false
    },
    + scheduledtasks: {...},
    + httptrace: {...},
    - mappings: {
      href: "http://localhost:8080/actuator/mappings",
      templated: false
    }
  }
}
```

As we can see its totally not prometheus friendly , this is where micrometer will come into action

```
<dependency>
  <groupId>io.micrometer</groupId>
  <artifactId>micrometer-registry-prometheus</artifactId></dependency>
```

It will result in something like:

```
localhost:8080/actuator/prometheus
# HELP jvm_buffer_count_buffers An estimate of the number of buffers in the pool
# TYPE jvm_buffer_count_buffers gauge
jvm_buffer_count_buffers{id="direct",} 3.0
jvm_buffer_count_buffers{id="mapped",} 0.0
# HELP process_cpu_usage The "recent cpu usage" for the Java Virtual Machine process
# TYPE process_cpu_usage gauge
process_cpu_usage 0.2131801983223912
# HELP tomcat_threads_busy_threads
# TYPE tomcat_threads_busy_threads gauge
tomcat_threads_busy_threads{name="http-nio-8080",} 1.0
# HELP logback_events_total Number of error level events that made it to the logs
# TYPE logback_events_total counter
logback_events_total{level="warn",} 0.0
logback_events_total{level="debug",} 0.0
logback_events_total{level="error",} 0.0
logback_events_total{level="trace",} 0.0
logback_events_total{level="info",} 7.0
# HELP jvm_classes_loaded_classes The number of classes that are currently loaded in the Java virtual machine
# TYPE jvm_classes_loaded_classes gauge
jvm_classes_loaded_classes 7315.0
# HELP tomcat_global_request_seconds
# TYPE tomcat_global_request_seconds summary
tomcat_global_request_seconds_count{name="http-nio-8080",} 1.0
tomcat_global_request_seconds_sum{name="http-nio-8080",} 0.149
# HELP jvm_threads_daemon_threads The current number of live daemon threads
# TYPE jvm_threads_daemon_threads gauge
jvm_threads_daemon_threads 19.0
# HELP process_start_time_seconds Start time of the process since unix epoch.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.571274356128E9
# HELP tomcat_global_received_bytes_total
# TYPE tomcat_global_received_bytes_total counter
tomcat_global_received_bytes_total{name="http-nio-8080",} 0.0
# HELP jvm_memory_used_bytes The amount of used memory
# TYPE jvm_memory_used_bytes gauge
jvm_memory_used_bytes{area="heap",id="PS Survivor Space",} 0.0
jvm_memory_used_bytes{area="heap",id="PS Old Gen",} 1.7224136E7
jvm_memory_used_bytes{area="heap",id="PS Eden Space",} 2.5879888E7
jvm_memory_used_bytes{area="nonheap",id="Metaspace",} 3.586084E7
jvm_memory_used_bytes{area="nonheap",id="Code Cache",} 7615744.0
```

Prometheus is a time series database that will pull the metrics periodically over http. It will be configured in a file `prometheus.yml` where one can specify `scrape_interval`, `metrics_path` and `targets` (ip which we want to monitor)

Alert manager will be configured using a configuration file and some command line flags.

The project will greatly impact Open source alliance as a complete responsive frontend will be provided where users will be able to enjoy the resources that they create using the cms services.

This will enable users to validate their content on the go.

Current tech stack:

Backend: springBoot(Java)

Infra: docker, microk8s, MariaDB, hazelcast(for in memory caching)

Proposed Tech Stack

Backend: Spring Security

Why?

Application is already written in springBoot ,so spring security would provide the best set of methods to secure endpoints.

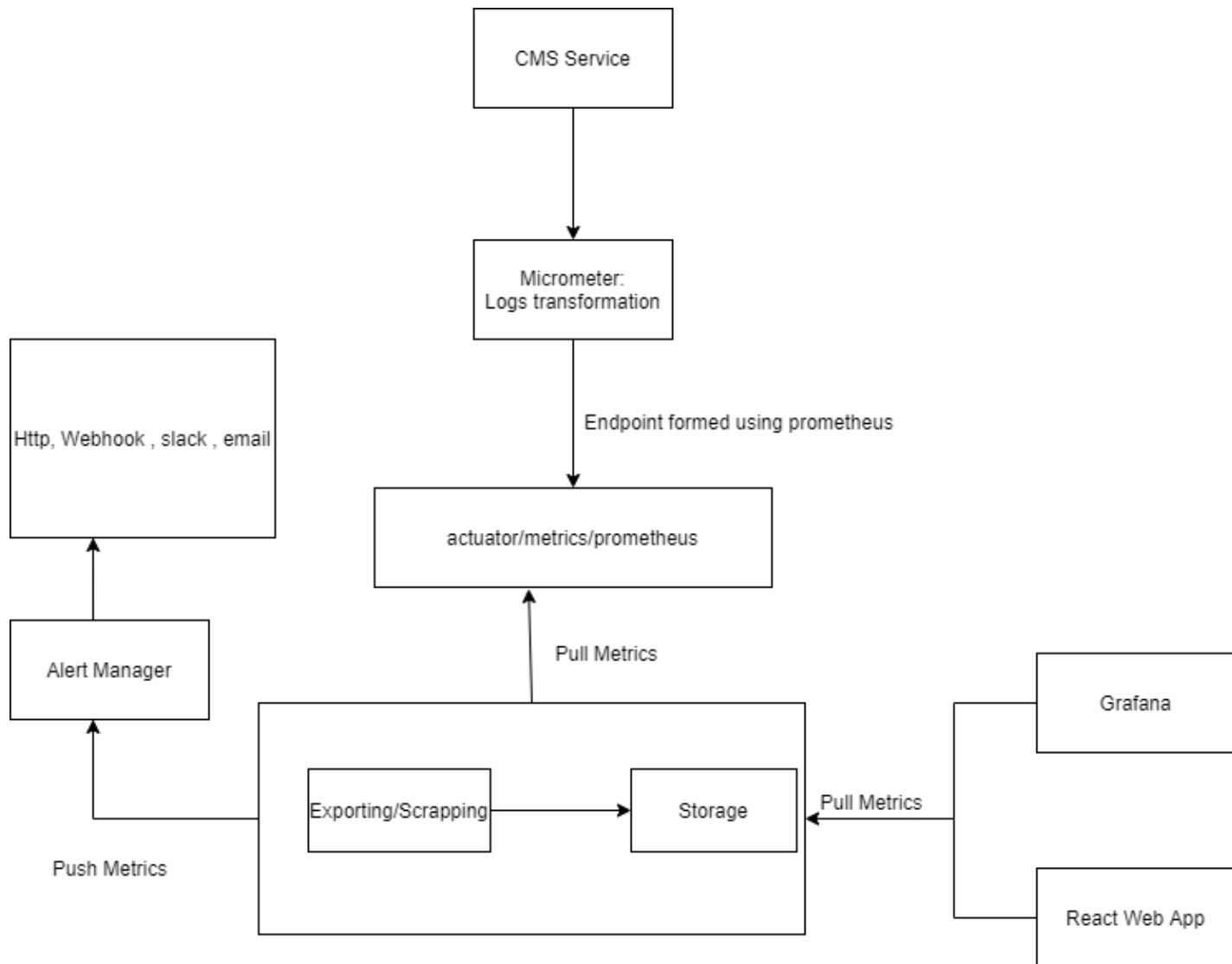
Frontend: ReactJs , Redux

Monitoring and Alerting

[Micrometer](#): Abstraction over instrumentation clients

[Prometheus](#): Open-source systems monitoring and alerting toolkit

[Grafana](#): For data visualization.



Timeline

Community bonding period

- To familiarize completely with projects functionality and architecture
- Bond with the community by regularly interacting with them about the ideas I present and discussing their implementation.
- Solve existing bugs to get myself more familiar with the code base

Week 1

- Bootstrapping the react app and deciding on the directory structure.
- Creating the layout i.e upper navigation bar , side navigation

Week 2

- Configuring the global theme across the app which includes primary and secondary button colours , font sizes and font colours, heading sizes standardization.
- Setup React Router.
-

Week 3

- Integrate all the basic CRUD operations with the app , this will provide the functionality of creator and general user.
- Create screen for content display similar to <https://demo.mediacms.io/> but with file types as categories as cms supports images and html as well.

Week 4

- Start access management of endpoints by creating appropriate user and permission classes , creating SpringSecurityConfig with basic auth initially.

Week 5

- Setup Redux Store
- Create screen for admin containing cards for showing various metrics.

Week 6

- Implement multi role user login functionality that will redirect user to a screen depending upon his role

Week 7

- Setting up micrometer for the springBoot app
- Setting up prometheus for micro8s cluster with required targets.

Week 8

- Setup email alerting service, slack notification service can also be implemented if needed
- Integrate prometheus metrics into the webapp

Week 9

- Profile picture updating functionality
- POC on hazelcast sidecar pattern

Week 10

- Dockerizing the webapp and setting up the CI/CD pipeline for it.
Final rounds of end to end testing and code refactoring and documentation.
- POC on hazelcast sidecar pattern (continue from week 9)

Evaluations

Eval 1

Webapp in place without login , with all the crud operations working for creator and consumer roles.

Eval 2

Webapp with role based login , screen for admin with all the metrics and alerting manager
Demonstration of side car POC.

Would I be a good fit?

I would definitely be a good fit not because I know all the technologies/frameworks mentioned above but by the fact that I would be creating something impactful that will keep pushing me off my limits to get the stuff done on time.

I also plan to keep working on this project ,or some other project under the same organization so keep myself an active and useful member of open source community.

After the gsoc 2021 coding period

- If I stuck somewhere in implementation, I will try to complete them after the GSoC period.
- I will continue my contribution and will be active in GFOSS community. Also I will help new contributors.
- If possible, I want to contribute to other projects under GFOSS.
- Create a react component which takes prometheus metrics and shows them in time series manner
- Start contributing to microK8s
- Will keep on refactoring the code as per the best practices
- Keep on documenting it and actively maintaining the repo

References/Discussions

<https://github.com/sastix/cms/issues/18>

<https://github.com/sastix/cms/issues/13>

<https://github.com/sastix/cms/issues/11>

https://drive.google.com/file/d/1UBxfCWD_kxCwYoHkp_JGynnNbsVqKcZL/view?usp=sharing