

Running and monitoring a low maintenance web application

Looking into the ElasticCC platform

Alexander Reelsen

alex@elastic.co | [@spinscale](https://twitter.com/spinscale)

Today's goal

How to run and maintain a modern web java application with minimal resources

Agenda

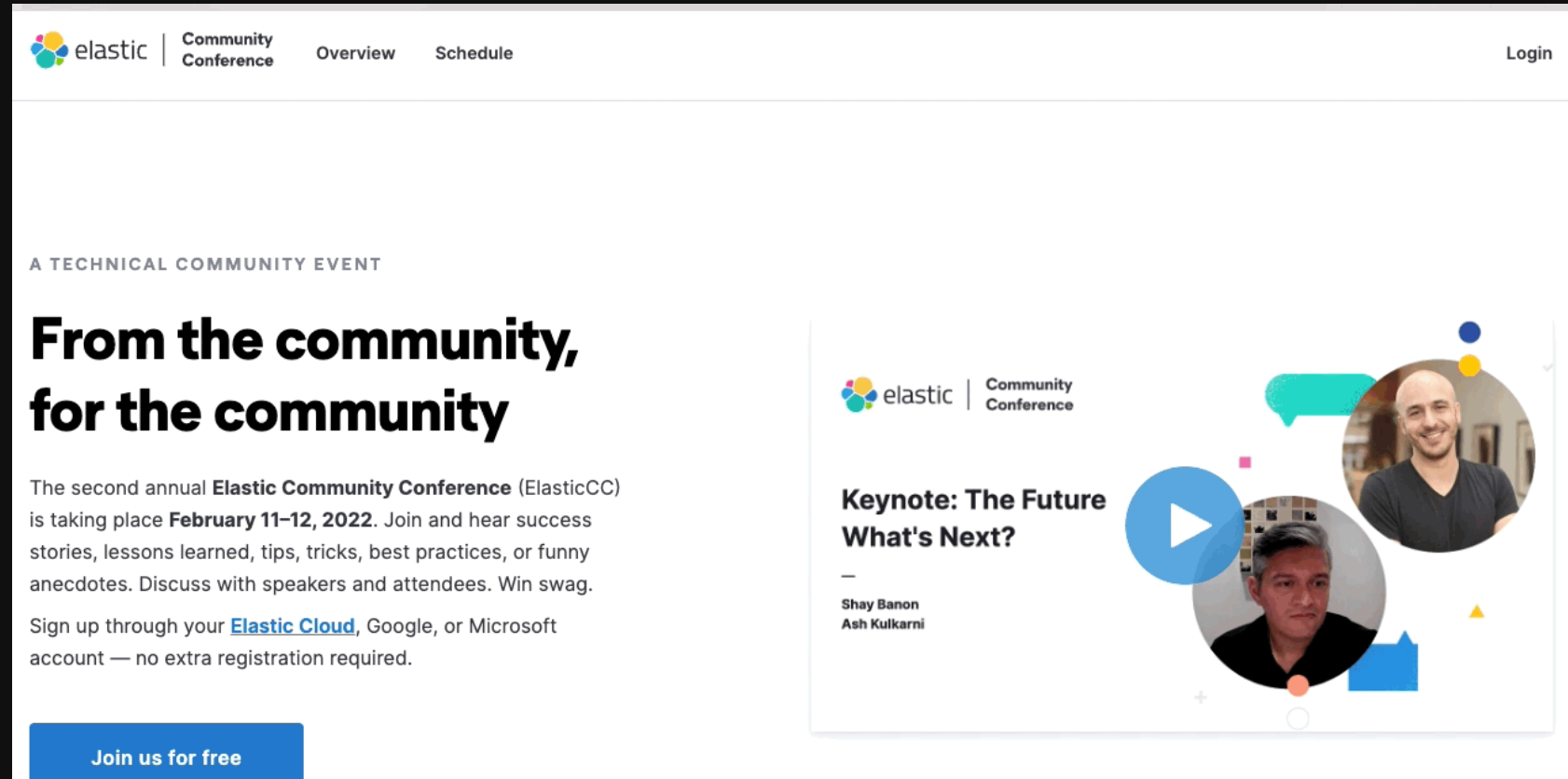
- Conference app as a platform? Build vs. Buy?
- Requirements
- Technologies used
- Persistence layer
- Downtime free rollouts
- Monitoring
- Frontend for backend devs
- Dashboards for tracking

Requirements

- Schedule & Speaker info
- Registration
- Feedback & Rating
- Quiz

Schedule

- Display
- Filtering
- Personalization
- Provide ical file



The screenshot shows the Elastic Community Conference website. The header includes the Elastic logo, "Community Conference", and navigation links for "Overview" and "Schedule". A "Login" link is in the top right. The main content area features the text "A TECHNICAL COMMUNITY EVENT" followed by the headline "From the community, for the community". Below this is a paragraph describing the conference: "The second annual Elastic Community Conference (ElasticCC) is taking place February 11-12, 2022. Join and hear success stories, lessons learned, tips, tricks, best practices, or funny anecdotes. Discuss with speakers and attendees. Win swag." It also includes a sign-up instruction: "Sign up through your Elastic Cloud, Google, or Microsoft account — no extra registration required." A blue button labeled "Join us for free" is at the bottom left. On the right, a featured event card for the "Keynote: The Future What's Next?" is shown, featuring a play button icon, a photo of Shay Banon, and the names "Shay Banon" and "Ash Kulkarni".

elastic | Community Conference

Overview Schedule Login

A TECHNICAL COMMUNITY EVENT

From the community, for the community

The second annual **Elastic Community Conference** (ElasticCC) is taking place **February 11-12, 2022**. Join and hear success stories, lessons learned, tips, tricks, best practices, or funny anecdotes. Discuss with speakers and attendees. Win swag.

Sign up through your [Elastic Cloud](#), Google, or Microsoft account — no extra registration required.

[Join us for free](#)

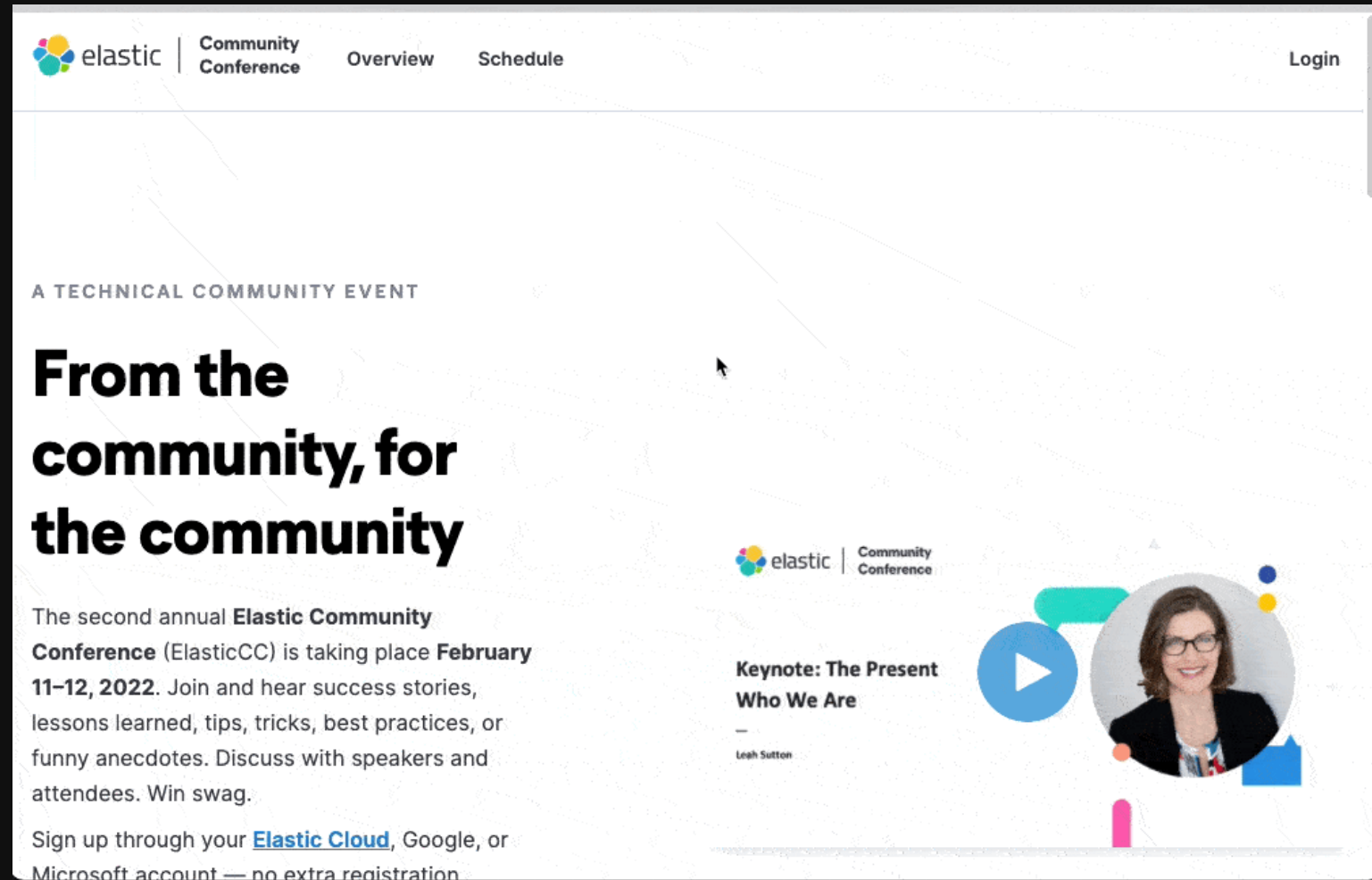
elastic | Community Conference

Keynote: The Future What's Next?

—
Shay Banon
Ash Kulkarni

Registration

- No password data in this system!
- SAML: Use Elastic Cloud login
- Danger: Breaks single page flow!



The screenshot shows the Elastic Community Conference registration page. At the top, there is a navigation bar with the Elastic logo, 'Community Conference', and links for 'Overview' and 'Schedule'. A 'Login' link is located in the top right corner. The main heading reads 'A TECHNICAL COMMUNITY EVENT' followed by 'From the community, for the community'. Below this, a paragraph describes the event: 'The second annual Elastic Community Conference (ElasticCC) is taking place February 11-12, 2022. Join and hear success stories, lessons learned, tips, tricks, best practices, or funny anecdotes. Discuss with speakers and attendees. Win swag.' A registration link is provided: 'Sign up through your [Elastic Cloud](#), Google, or Microsoft account — no extra registration'. On the right side, there is a featured keynote section titled 'Keynote: The Present Who We Are' by Leah Sutton, accompanied by a circular video player icon and a portrait of the speaker.

Feedback

- Rating
- Feedback wall

The screenshot shows the Elastic Community Conference website. At the top, there is a navigation bar with the Elastic logo, 'Community Conference', and links for 'Overview', 'Schedule', and 'Profile'. A 'Logout' link is in the top right corner. The main heading is 'Keynote: The Past — 10 Years Elastic' with the date 'February 11, 2022'. Below this, there are three blue buttons: 'Feedback', 'Take the quiz', and 'Add to my Agenda'. The event time is listed as '17:00 — 17:35 (EUROPE/BERLIN)' and '16:00 — 16:35 (UTC)'. The event title is 'Keynote: The Past — 10 Years Elastic'. The description reads: 'Q&A with the Elasticsearch founders — back when the company wasn't even called Elastic yet. Some of the questions we are asking them are: What was your goal in founding Elasticsearch? What is the reason for Elastic's success? Is there anything you would have done differently at Elastic? Is there any product specific feature you're particularly proud of? Is there a feature of the Elastic Stack that you think is underrated?'. Below the description is a 'Full session' button. Two speakers are listed: 'Shay Banon, Founder, CTO | Elastic' and 'Simon Willnauer, Founder | Elastic'.

elastic | Community Conference Overview Schedule Profile Logout

Keynote: The Past — 10 Years Elastic

February 11, 2022


17:00 — 17:35 (EUROPE/BERLIN)
16:00 — 16:35 (UTC)


Feedback Take the quiz Add to my Agenda

Keynote: The Past — 10 Years Elastic

Q&A with the Elasticsearch founders — back when the company wasn't even called Elastic yet. Some of the questions we are asking them are: What was your goal in founding Elasticsearch? What is the reason for Elastic's success? Is there anything you would have done differently at Elastic? Is there any product specific feature you're particularly proud of? Is there a feature of the Elastic Stack that you think is underrated?

Full session

 **Shay Banon**
Founder, CTO | Elastic

 **Simon Willnauer**
Founder | Elastic

Build vs. Buy

- Customization
- SAML integration
- Hefty price tag
- Focus on commercial events, not free ones

Use your own technologies in production!

-- Me

Requirements

- Speed (rendering)
- Ease of use & feature development
- Fast startup
- Monitoring/APM
- Docker image to deploy to k8s
- **Low maintenance**

Technologies used

- Javalin (Jetty)
- JTE
- pac4j
- Elastic Stack
- Config file parsing: typesafe/lightbend config
- Testing: assertJ, testcontainers, equalsverifier
- Frontend: htmx & hyperscript

Javalin

- Based on jetty, servlet based
- No reflection, no annotations
- Fast startup
- Extensible
- Built-in support for WS, SSE, OpenAPI

JTE

- **Fast**, typed, built-in escaping
- Hot reloading in development
- Build time compilation for production
- Tag support

```
@import my.Entry
@param Entry entry
@param boolean verbose = false

<h2>${entry.title}</h2>
@if(verbose)
  <h3>${entry.subtitle}</h3>
@endif
```

pac4j

The Java security engine to protect all your web applications and web services

- Authorization
- Authentication (OAuth, SAML, CAS, OpenID Connect, JWT, SPNEGO)
- Supports almost 20 web frameworks

Persistence layer

- CfP process is done via sessionize
- Contains schedule data, speaker data
- Allows to add custom fields
- Has an API, read-only
- Schedule can be polled and updated every 10 minutes
- No need for persistence for the schedule, but...

Persistence layer

- Storing registrations, ratings, feedback, custom agenda
- Elasticsearch
- Use new [Elasticsearch Java Client](#) to provide feedback

Elasticsearch persistence

```
public record Rating(String sessionId, String userId,
                    Integer rating, List<Feedback> feedbacks, OffsetDateTime createdAt) {
}
```

```
public class RatingService {

    public static final String INDEX = "elasticcc-ratings";

    private final ElasticsearchClient client;

    public RatingService(ElasticsearchClient client) {
        this.client = client;
    }

    public void save(Rating rating) throws IOException {
        client.index(b -> b.index(INDEX)
                    .id(rating.sessionId() + "-" + rating.userId())
                    .document(rating)
                    );
    }
}
```


Elasticsearch persistence

```
public final static ObjectMapper mapper = new ObjectMapper()
    .registerModule(new JavaTimeModule())
    .setPropertyNamingStrategy(PropertyNamingStrategies.SNAKE_CASE)
    .configure(SerializationFeature.WRITE_DATES_AS_TIMESTAMPS, false)
    .setSerializationInclusion(JsonInclude.Include.NON_NULL);

public ElasticsearchClient toElasticsearchClient(Config config) {
    final RestClientBuilder builder = RestClient.builder(Host.create(config.getString("host")));

    if (config.hasPath("user") && config.hasPath("password")) {
        final String user = config.getString("user");
        final String password = config.getString("password");
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(user, password));
        builder.setHttpClientConfigCallback(b -> b.setDefaultCredentialsProvider(credentialsProvider));
    }

    RestClient restClient = builder.build();
    ElasticsearchTransport transport = new RestClientTransport(restClient, new JacksonJsonMapper(mapper));
    return new ElasticsearchClient(transport);
}
```

Elasticsearch persistence

```
public Map<String, QuizAnswer> loadAnswers(String userId, List<String> questionIds) throws IOException {
    List<FieldValue> fieldValues = questionIds.stream().map(id -> FieldValue.of(x -> x.stringValue(id))).toList();
    SearchResponse<QuizAnswer> response = client.search(b -> b
        .index(ANSWER_INDEX)
        .size(15)
        .query(q -> q.bool(bq -> bq
            .filter(fb -> fb
                .term(t -> t.field("user_id.keyword").value(FieldValue.of(userId))))
            .filter(fb -> fb
                .terms(t -> t.field("question_id.keyword").terms(TermsQueryField.of(tqb -> tqb.value(fieldValues))))))
        ))
        , QuizAnswer.class);

    return response.hits().hits().stream().collect(Collectors.toMap(h -> h.source().questionId(), Hit::source));
}
```

Downtime free rollouts

```
kubectl rollout restart deployment/elasticcc-app
```

- readiness probes
- liveness probes
- persistent sessions (> 1 replicas, round robin)

Persistent sessions

```
public static SessionHandler createSessionHandler(ElasticsearchClient client) {  
    SessionHandler sessionHandler = new SessionHandler();  
    sessionHandler.setMaxInactiveInterval(86_400 * 7); // full week  
    sessionHandler.setHttpOnly(true);  
    sessionHandler.setSameSite(HttpCookie.SameSite.STRICT);  
    sessionHandler.setSecureRequestOnly(true);  
  
    SessionCache sessionCache = new NullSessionCache(sessionHandler);  
    sessionCache.setSaveOnCreate(true);  
    sessionCache.setFlushOnResponseCommit(true);  
    sessionCache.setSessionDataStore(new ElasticsearchSessionDataStore(client));  
    sessionHandler.setSessionCache(sessionCache);  
    return sessionHandler;  
}
```

Persistent sessions

```
@Override
public void doStore(String id, SessionData data, long lastSaveTime) throws Exception {
    if (!isSessionIdValid(id)) {
        return;
    }

    final String base64Session;
    try (ByteArrayOutputStream bos = new ByteArrayOutputStream(); ObjectOutputStream oos = new ObjectOutputStream(bos)) {
        SessionData.serializeAttributes(data, oos);
        oos.flush();
        base64Session = Base64.getEncoder().encodeToString(bos.toByteArray());
    }

    ElasticsearchSession session = new ElasticsearchSession(id, _context.getCanonicalContextPath(), _context.getVhost(),
        data.getLastNode(), data.getCreated(), data.getAccessed(), data.getLastAccessed(), data.getCookieSet(),
        data.getExpiry(), data.getMaxInactiveMs(), base64Session);

    client.index(b -> b.index(INDEX).id(id).document(session));
}
```

Persistent sessions - problem: bots

- About 99% of the requests are bots, security scanners, that create a new session per request
- Persists a six digit amount of sessions per day
- Solution: Persist session only, if users are logged in

Persistent sessions - login check

```
app.after("/login", ctx -> {  
    ctx.sessionAttribute("persist_session", true);  
});
```

```
public void doStore(String id, SessionData data, long lastSaveTime) throws Exception {  
    // validation  
  
    // check if should be persisted  
    Boolean persistSession = (Boolean) data.getAttribute("persist_session");  
    if (persistSession == null || persistSession == false) {  
        return;  
    }  
  
    // persist  
    // ...  
}
```

Persistent sessions - login check

- **much** less processing time for most requests
- **much** less Elasticsearch indexing work
- less CPU time wasted, less useless work done 👍

Monitoring

- Elastic APM, from the browser to the backend

Observability

- Overview
- Alerts
- Cases

Logs

- Stream
- Anomalies
- Categories

Metrics

- Inventory
- Metrics Explorer

APM

- Services
- Traces
- Dependencies

Service Map

Uptime

- Monitors
- TLS Certificates

User Experience

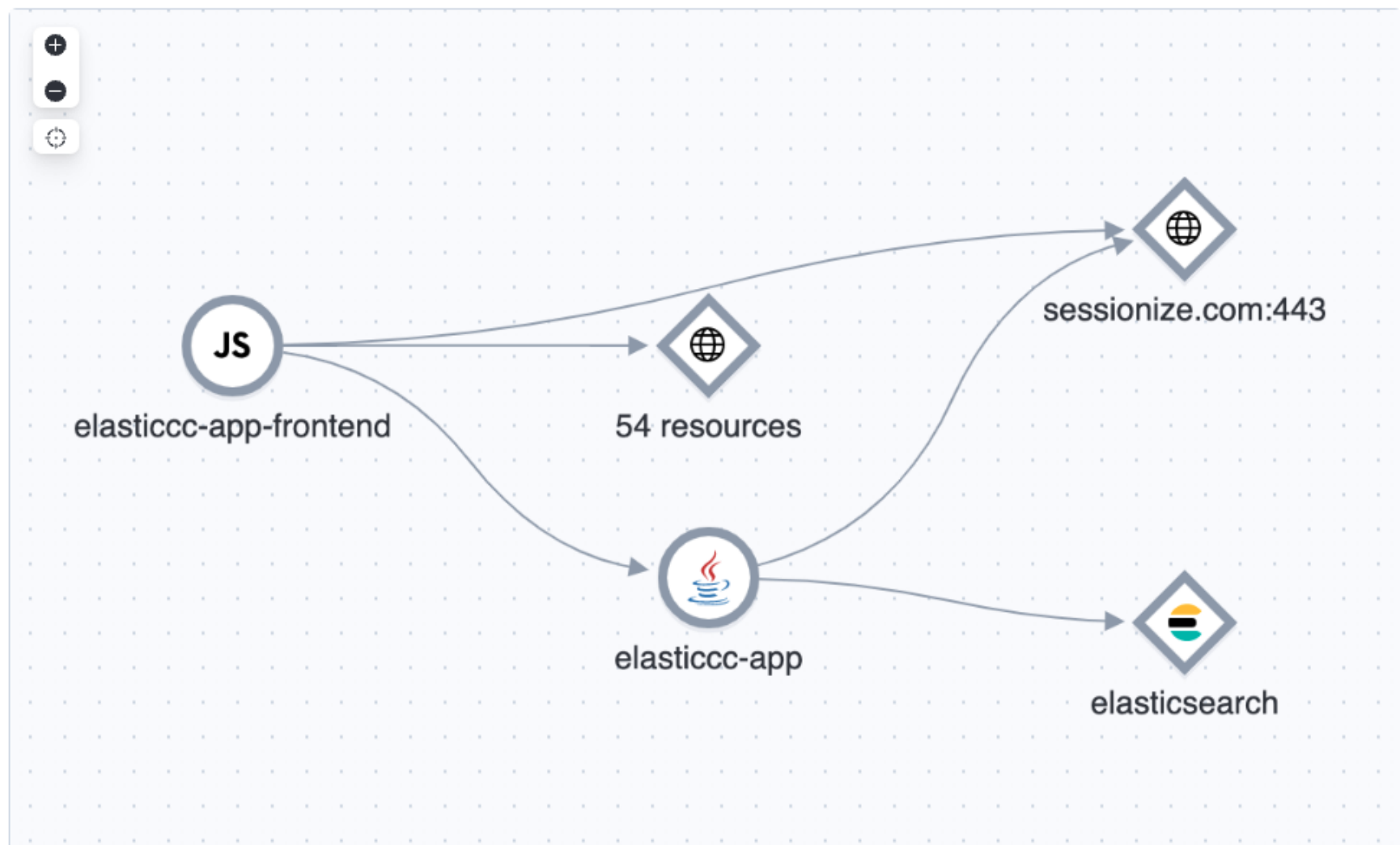
- Dashboard

Service Map

Environment All

Jan 20, 2022 @ 00:49:37.52 → Feb 21, 2022 @ 04:42:01.03

Refresh



Elastic APM setup

```
// set version at build time to annotate rollouts
setElasticApmServiceVersion();

System.setProperty("elastic.apm.environment", isProduction() ? "prod" : "dev");
System.setProperty("elastic.apm.application_packages", "co.elastic.community");

// add a span for rendering .jste templates and a span for Context.render()
System.setProperty("elastic.apm.trace_methods",
    "io.javalin.plugin.rendering.template.JavalinJste#render,io.javalin.http.Context#render");

// those should not be shown in the APM UI, as they are controlling flow/logic
System.setProperty("elastic.apm.ignore_exceptions",
    "io.javalin.http.RedirectResponse,*ResponseNoopException");

// this is to have consistent JVM stats across restarts in the APM UI
System.setProperty("elastic.apm.service_node_name", "elasticcc-service-java");

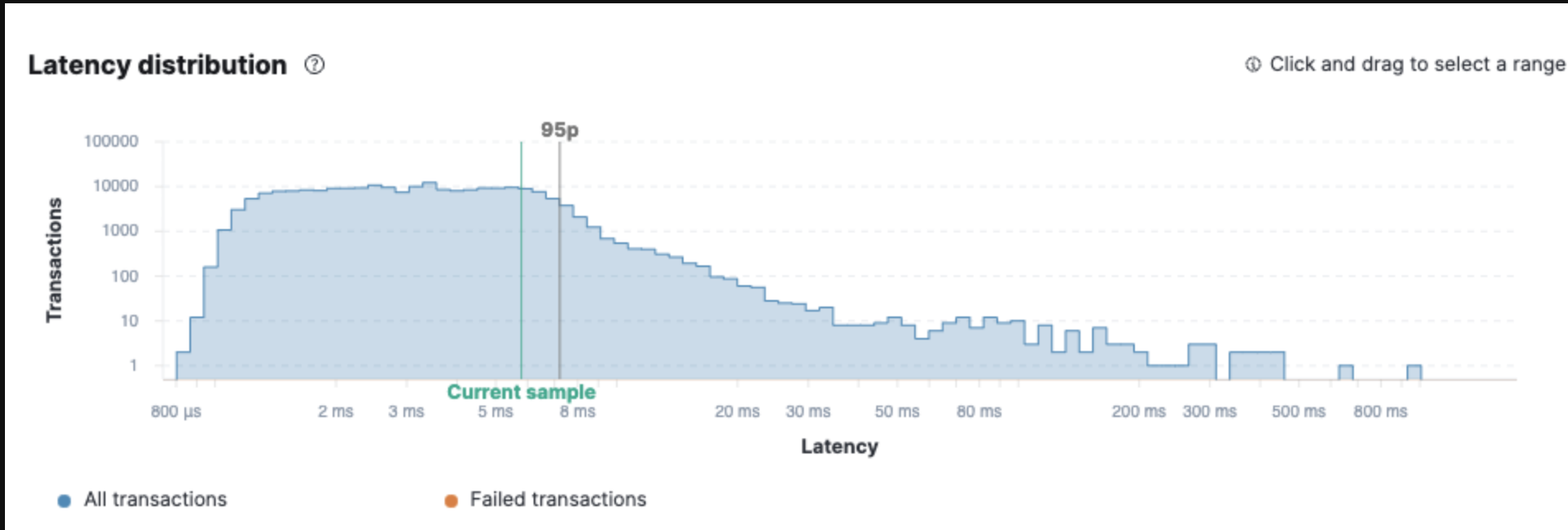
// programmatic attachment, no need to configure an agent
ElasticApmAttacher.attach();
```

Elastic APM setup

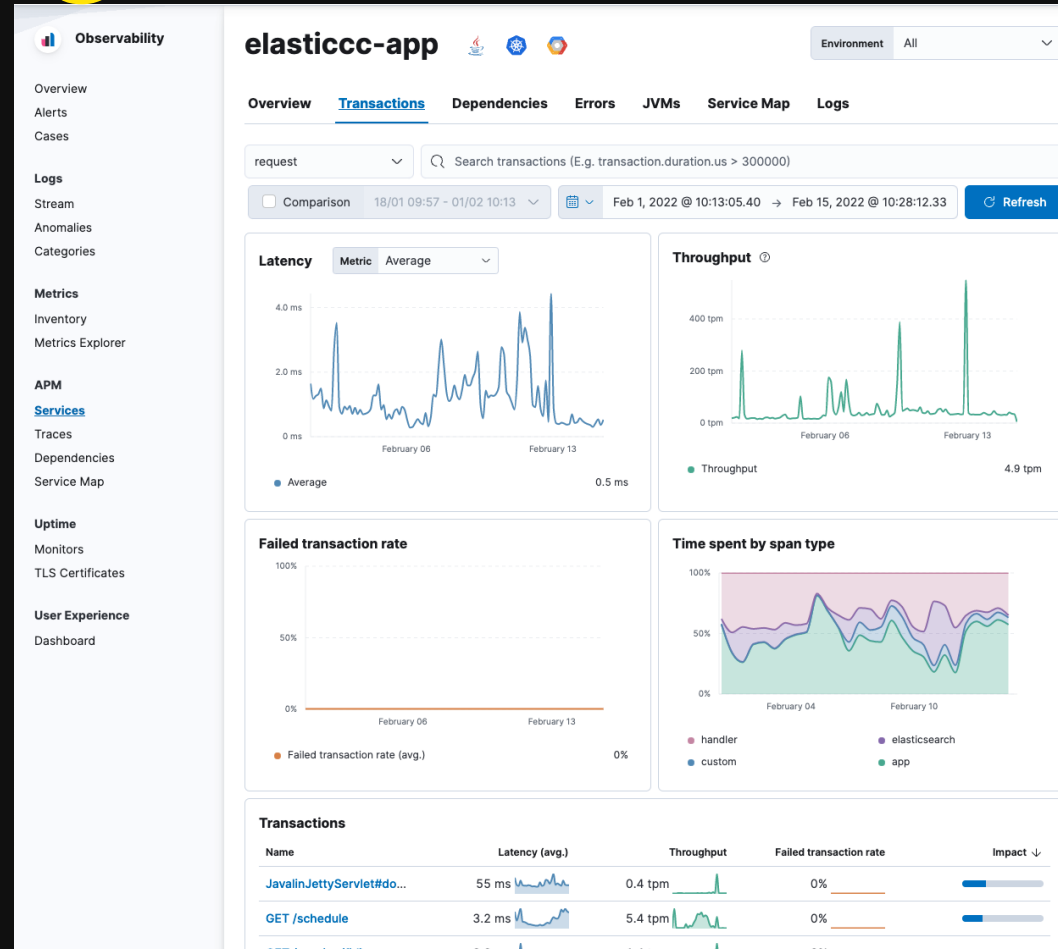
```
@param co.elastic.apm.api.Transaction transaction
```

```
<script>  
  ;(function(d, s, c) {  
    var j = d.createElement(s), t = d.getElementsByTagName(s)[0]  
  
    j.src = 'https://unpkg.com/@elastic/apm-rum@5.10.0/dist/bundles/elastic-apm-rum.umd.min.js'  
    j.onload = function() {elasticApm.init(c)}  
    t.parentNode.insertBefore(j, t)  
  })(document, 'script', {  
    serviceName: 'elasticcc-app-frontend', environment: 'prod',  
    serverUrl: 'https://elasticcc-app.apm.us-central1.gcp.cloud.es.io',  
    pageLoadTraceId: "${transaction.getTraceId()}",  
    pageLoadSpanId: "${transaction.ensureParentId()}",  
    pageLoadSampled: ${transaction.isSampled()}  
  })  
</script>  
</body>  
</html>
```

Monitoring - Latency



Monitoring - Transactions





- Overview
- Alerts
- Cases

Logs

- Stream
- Anomalies
- Categories

Metrics

- Inventory
- Metrics Explorer

APM

- Services**
- Traces
- Dependencies
- Service Map

Uptime

- Monitors
- TLS Certificates

User Experience

- Dashboard

- Overview
- Transactions**
- Dependencies
- Errors
- JVMs
- Service Map
- Logs

request

Search transactions (E.g. transaction.duration.us > 300000)

Comparison

18/01 09:57 - 01/02 10:13



Feb 1, 2022 @ 10:13:05.40 → Feb 15, 2022 @ 10:28:12.33

Refresh

Latency

Metric Average



Average 0.5 ms

Throughput



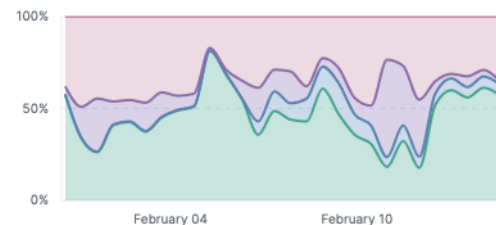
Throughput 4.9 tpm

Failed transaction rate



Failed transaction rate (avg.) 0%

Time spent by span type



- handler
- elasticsearch
- custom
- app

Transactions

Name	Latency (avg.)	Throughput	Failed transaction rate	Impact ↓
JavalinJettyServlet#do...	55 ms	0.4 tpm	0%	
GET /schedule	3.2 ms	5.4 tpm	0%	
GET /session/{id}	3.6 ms	1.4 tpm	0%	

Errors

- NPE detection

Observability

Overview
Alerts
Cases

Logs

Stream
Anomalies
Categories

Metrics

Inventory
Metrics Explorer

APM

Services

Traces
Dependencies
Service Map

Uptime

Monitors
TLS Certificates

User Experience

Dashboard

elasticcc-app



Environment All

Overview Transactions Dependencies **Errors** JVMs Service Map Logs

Search errors (E.g. http.response.status_code >= 400)

Comparison

18/01 09:57 - 01/02 10:13



Feb 1, 2022 @ 10:13:05.40 → Feb 15, 2022 @ 10:28:12.33

Refresh

Error group c86af

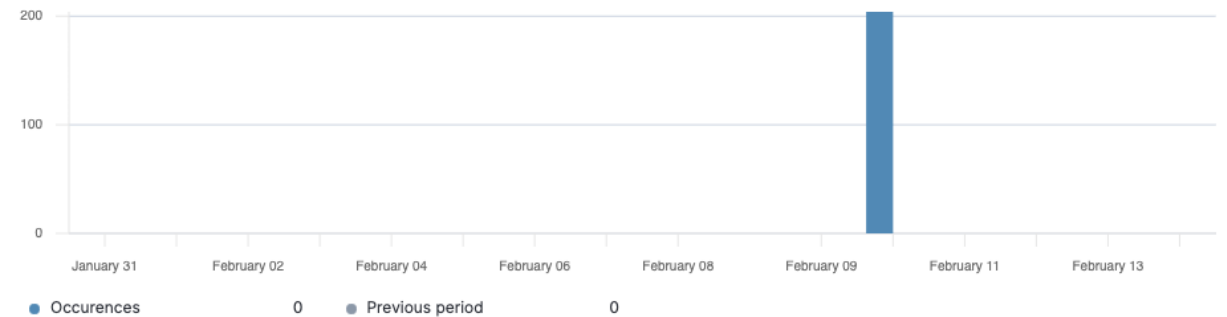
Exception message

Failed to render session.jte, error at tag/session.jte:93

Culprit

`co.elastic.community.controller.SessionController.lambda$showSession$2(SessionController.java:123)`

Occurrences



Error occurrence

[View 204 occurrences in Discover.](#)

24 days ago

GET `http://community-conference.elastic.co/session/310406`

Chrome (99.0.4844.15)

GET `/session/{id}`

Exception stack trace

Metadata

Failed to render session.jte, error at tag/session.jte:93

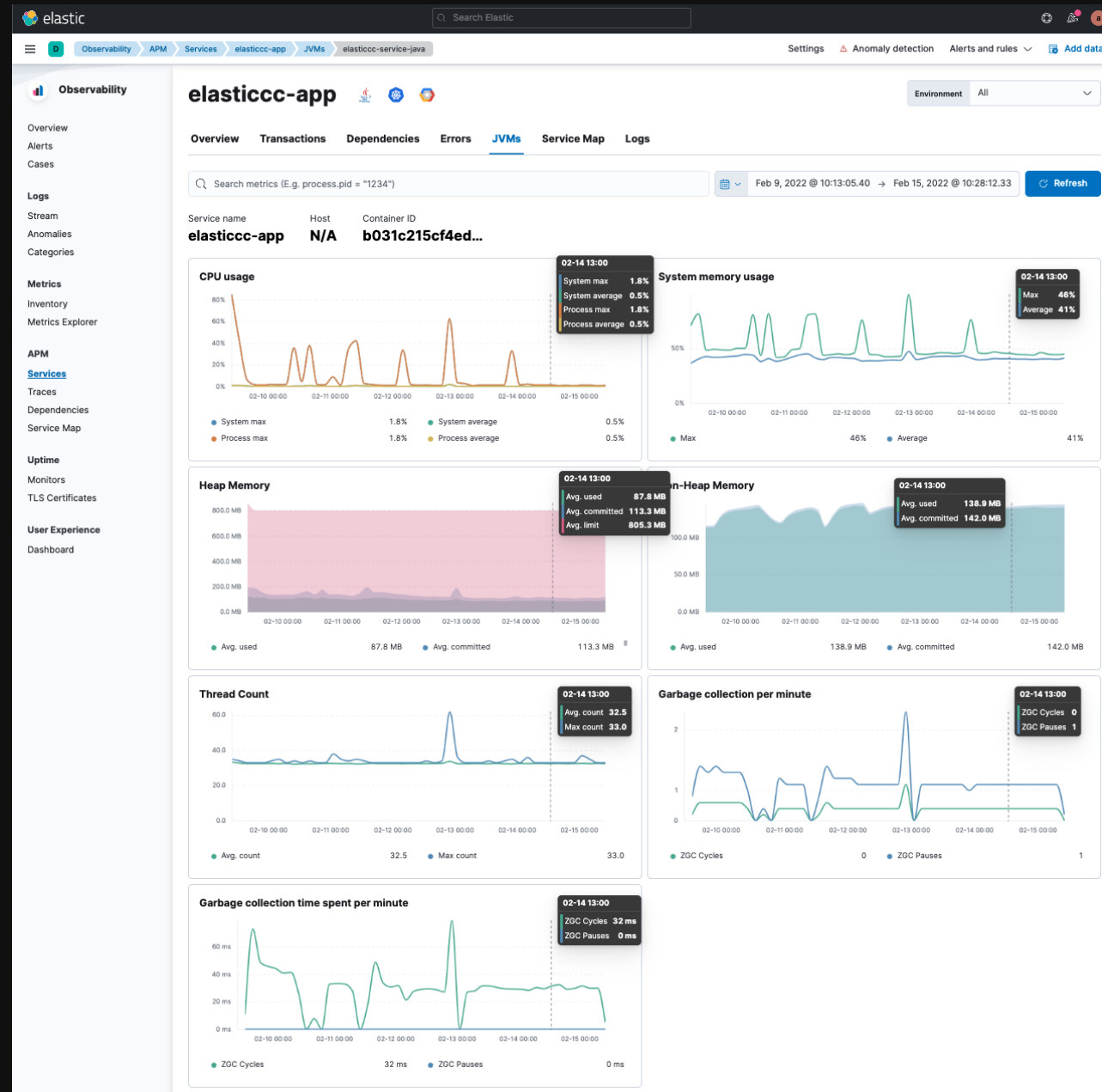
> 5 library frames

at `co.elastic.community.controller.SessionController.lambda$showSession$2(SessionController.java:123)`

> 42 library frames

JVM monitoring

- ZGC: 👍
- CPU: 👍
- Mem: 👍



Frontend

...is part of your performance!

Observability

- Overview
- Alerts
- Cases

Logs

- Stream
- Anomalies
- Categories

Metrics

- Inventory
- Metrics Explorer

APM

Services

- Traces
- Dependencies
- Service Map

Uptime

- Monitors
- TLS Certificates

User Experience

- Dashboard

Latency

- All transactions
- Failed transactions

Trace sample < 2 of 500 >

Investigate

View full trace

20 days ago | 2,832 ms (100% of trace) | <https://community-conference.elastic.co/schedule> | Chrome (98.0.4758.80)

Timeline Metadata Logs

Services elasticcc-app-frontend elasticcc-app

0 ms 500 ms 1,000 ms 1,500 ms 2,000 ms 2,500 ms 2,832 ms



Frontend for backend devs

- SPAs as a no-go due to lacking skills
- `htmx` for making websites dynamic, but rendering server side
- `hyperscript` for events

htmx

- alternatives: Hotwire, Turbo
- HTML tag enrichment to send AJAX request
- render results server side
- configure HTML element to be updated with the response
- optional: update browser history

htmx

```
@param String sessionId

<ul id="rating_{$sessionId}" style="list-style-type:none;display:inline;margin-left:0px;">
  @for(int i = 1; i <= 5; ++i)
    <li style="display:inline;">
      <a hx-post="/session/{$sessionId}/rating/{$i}"
        hx-target="#rating_{$sessionId}"
        hx-swap="outerHTML"
        style="cursor:pointer;text-decoration:none;">

      </a>
    </li>
  @endfor
</ul>
```

hyperscript

hyperscript makes writing event handlers and highly responsive user interfaces easy with a clear, DOM-oriented syntax and by transparently handling asynchronous behavior for you — easier than callbacks, promises, even `async/await`.

```
<div _="on mouseenter toggle .visible on #help until mouseleave">  
Mouse Over Me!  
</div>  
<div id="help"> I'm a helpful message!</div>
```

hyperscript

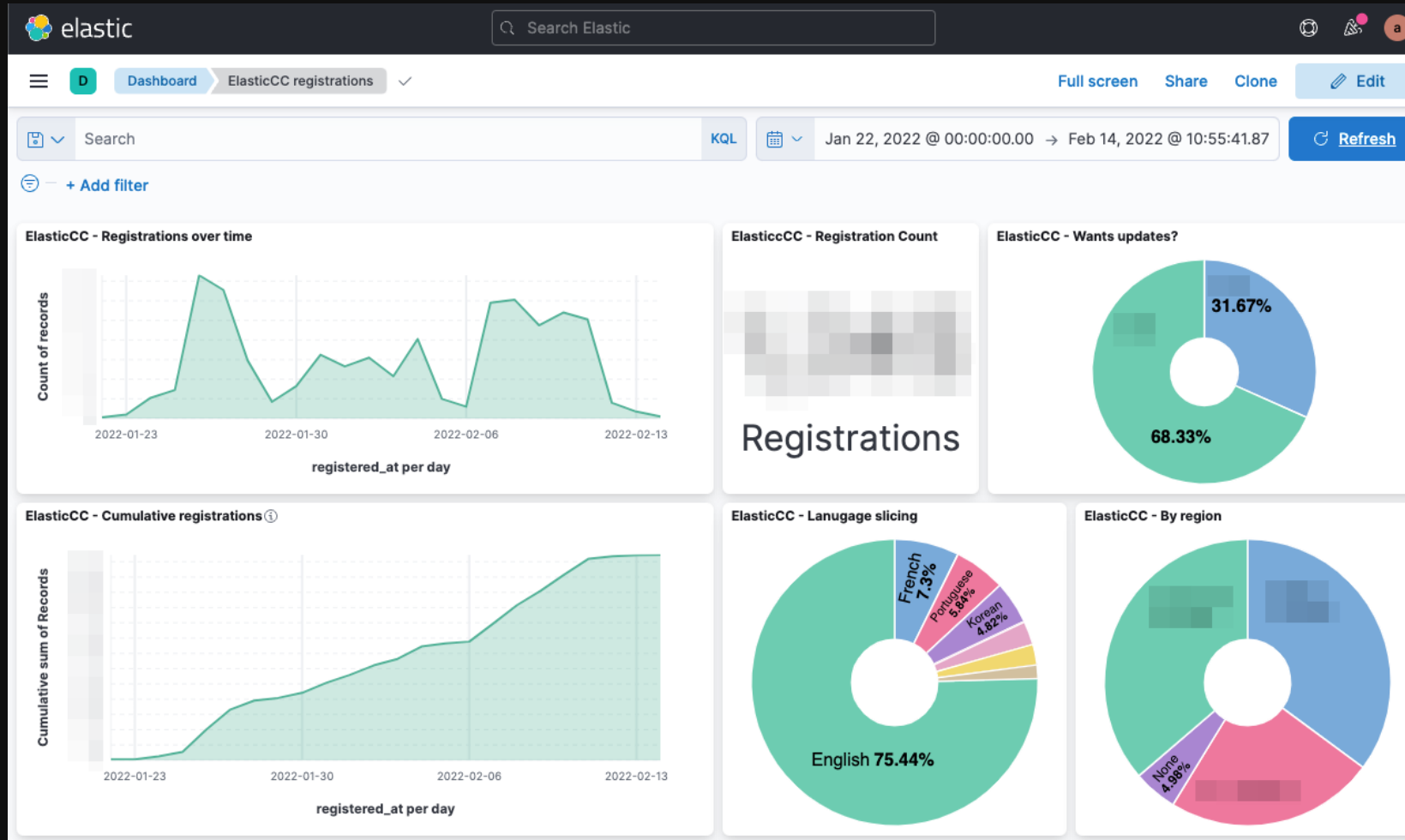
```
<div style="display: grid; grid-template-columns:1fr 1fr 1fr 1fr;">
  @for(var feedback : Feedback.values())
    !{var hasFeedback = rating != null && rating.feedbacks() != null
      && rating.feedbacks().contains(feedback);}

  <div class="@if(hasFeedback)button-primary-blue@elsebutton-secondary-blue@endif"
    _="on click
      toggle .button-secondary-blue toggle .button-primary-blue
      toggle [@checked=true] on #${feedback.name().toLowerCase(Locale.ROOT)}"
    >${feedback.displayName()}
  </div>

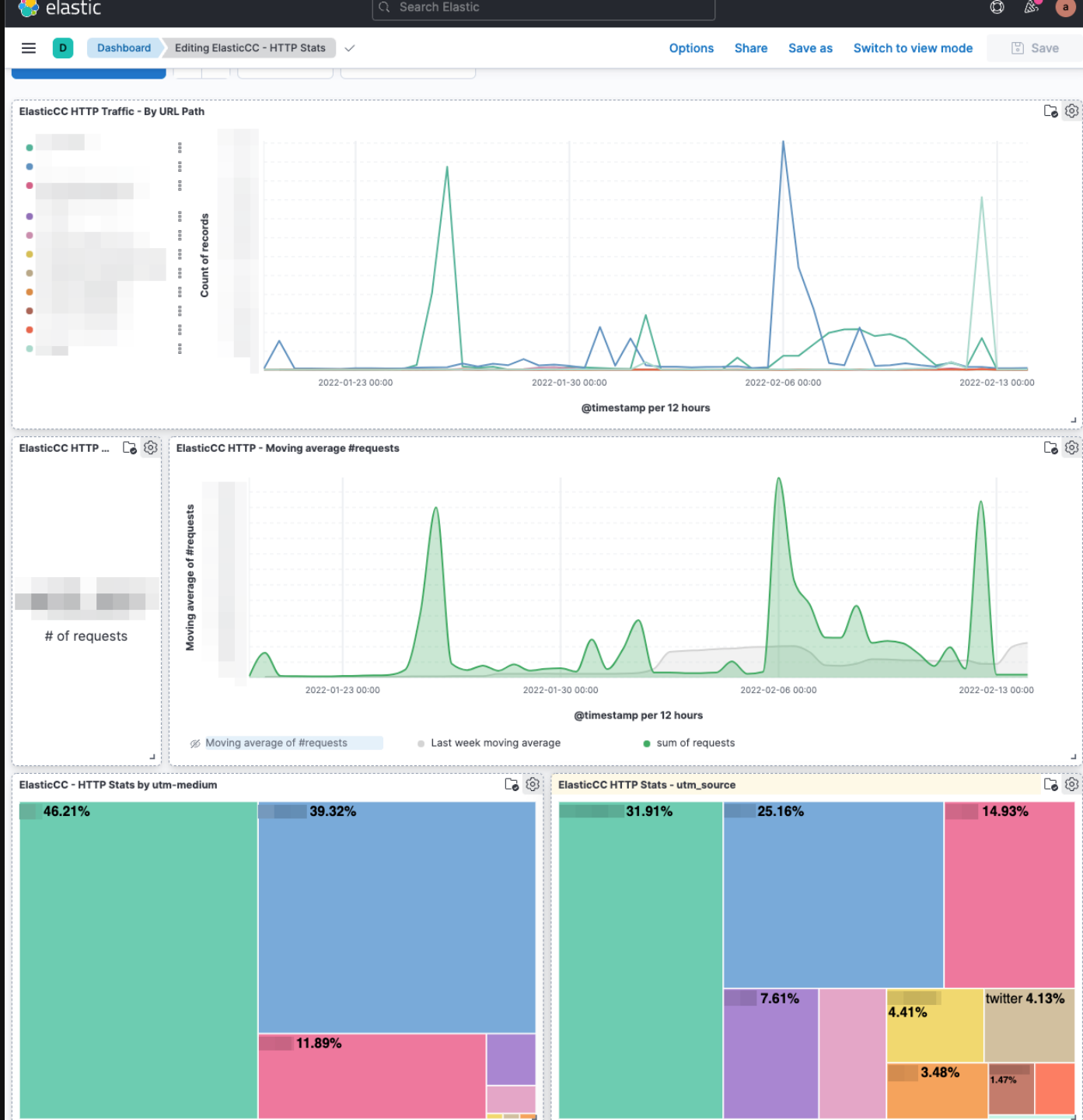
  <input type="checkbox"
    style="display:none"
    id="${feedback.name().toLowerCase(Locale.ROOT)}"
    name="${feedback.name()}" checked="${hasFeedback}"/>

  @endfor
</div>
```

Registrations



HTTP requests



Docker build

- multi layer build
- template precompilation
- use gradle `application` plugin to create shell scripts to start apps
- configure JVM options in `build.gradle`, i.e. `ZGC` usage

Docker build

```
FROM gradle:7.4-jdk17 AS build
COPY --chown=gradle:gradle . /home/gradle/src
WORKDIR /home/gradle/src
RUN gradle clean --no-daemon
RUN gradle copyDepsToLib --no-daemon
RUN gradle check --no-daemon
RUN gradle precompileJte --no-daemon
RUN gradle copyStartScriptsToBin --no-daemon
RUN gradle copyAppToLib --no-daemon

#####
FROM --platform=linux/amd64 openjdk:17.0.2-slim-bullseye
RUN addgroup --system elasticcc && adduser --system elasticcc --ingroup elasticcc
USER elasticcc:elasticcc

WORKDIR /elasticcc-app
# copy resources
COPY ...

EXPOSE 8080
ENTRYPOINT [ "/elasticcc-app/install/elasticcc-app/bin/elasticcc-app" ]
```

Testing

- Unit tests & integration tests
- Equalsverifier, assertJ, mockito
- Testcontainers

Testing - Equalsverifier

```
public class EqualsVerifierTests {  
  
    @Test  
    public void testEqualsVerifier() {  
        EqualsVerifier.forClass(User.class)  
            .suppress(Warning.STRICT_INHERITANCE, Warning.NONFINAL_FIELDS)  
            .verify();  
  
        EqualsVerifier.forClass(Rating.class).verify();  
        EqualsVerifier.forClass(QuizAnswer.class).verify();  
        EqualsVerifier.forClass(QuizQuestion.class).verify();  
        EqualsVerifier.forClass(QuizQuestionAnswer.class).verify();  
    }  
}
```

Testing - Controller

```
@Test
public void testShowSessionWhenNotLoggedIn() throws Exception {
    when(ctx.pathParam(eq("id"))).thenReturn("123");
    final Session session = createSession("123");
    when(sessionService.session(eq(session.id()))).thenReturn(session);
    // not logged in
    when(ctx.sessionAttribute(eq("user"))).thenReturn(null);

    controller.showSession().handle(ctx);

    assertThat(ctx).hasRenderTemplate("session");
    assertThat(ctx).renderMap().containsEntry("session", session);
    verifyNoInteractions(quizService);
    verifyNoInteractions(ratingService);
}
```

Testing - Testcontainers

```
@Tag("slow")
public class ElasticsearchSessionDataStoreTests extends AbstractElasticsearchContainerTest {

    @Test
    public void testStoreAndSerializeUserEntity() throws Exception {
        final long millis = ZonedDateTime.of(2021, 11, 8, 12, 34, 56, 123_000_000, ZoneOffset.UTC)
            .toInstant()
            .toEpochMilli();
        SessionData sessionData = new SessionData(sessionId, "", "0.0.0.0",
            millis, millis + 1000, millis + 1000, 1000);
        User user = new User();
        user.setRegion(User.Region.EMEA);
        user.setName("Paul Paulson");
        sessionData.putAllAttributes(Map.of("user", user, "persist_session", true));

        sessionStore.store(sessionData.getId(), sessionData);

        final SessionData loadedSession = sessionStore.load(sessionData.getId());
        assertThat(loadedSession.getAllAttributes()).containsEntry("user", user);
    }
}
```

Keep in mind...

low maintenance != no maintenance

- dependabot
- security is an ongoing task

Going live

- Secrets via vault
- JSON logging
- Access to logfiles
- k8s readiness/liveness probes





Summary

```
> scc --exclude-dir src/main/resources,build,out,.idea,src/test/resources,youtube-stats
```

Language	Files	Lines	Blanks	Comments	Code	Complexity
Java	82	6746	901	257	5588	424
Properties File	5	7	0	1	6	0
Gradle	2	168	28	16	124	0
Shell	2	206	29	38	139	21
YAML	2	185	4	1	180	0
Batch	1	89	21	0	68	18
Crystal	1	46	5	9	32	2
Docker ignore	1	6	0	0	6	0
Dockerfile	1	30	4	6	20	1
JavaScript	1	170	29	32	109	0
Makefile	1	5	0	0	5	0
Markdown	1	128	29	0	99	0
Total	100	7786	1050	360	6376	466

```
Estimated Cost to Develop (organic) $188,960  
Estimated Schedule Effort (organic) 7.301886 months  
Estimated People Required (organic) 2.299068
```

```
Processed 393849 bytes, 0.394 megabytes (SI)
```

```
.../elastic/community/elasticcc-app on  main via  v17.0.2 via  v17.6.0 via  v1.3.2
```

Summary

- Modern ✓
- Minimal resources ✓
- Other java developers commit changes ✓
- Low maintenance ✓
- Maximum performance? Graal ✗

Resources

- Community Conference website
- Web server: Javalin, Jetty, JTE, pac4j
- Elastic: Elastic Stack, Elasticsearch Java Client, Elastic Java APM Agent
- Configuration: typesafe config library
- Testing: assertJ, mockito, Testcontainers, EqualsVerifier
- Frontend: htmx, hyperscript, hotwire

Thanks for listening

Q & A

Alexander Reelsen

alex@elastic.co | [@spinscale](https://twitter.com/spinscale)