Elastic Observability Workshop

Lab 5 - APM

1) Download the sample Java App - Pet Clinic app

This is a simple spring boot demo app and requires Java Runtime. It is based on this repo https://github.com/bvader/spring-petclinic

But we don't use Java for app development?
We are using Java based app just as an example for the lab. Elastic APM supports other languages as well that you are more than welcome to try out with the respective agents. We had to pick one language for this lab, and we picked Java. The process of getting APM traces into Elastic to combine them with Logs and Metrics is similar across other languages as well.

More on APM (not needed for lab)
https://www.elastic.co/guide/en/apm/get-started/current/index.html
https://www.elastic.co/guide/en/apm/server/current/index.html
https://www.elastic.co/guide/en/apm/agent/java/current/intro.html

Hopefully you have Java on your laptop if not, install Java JDK see here:
https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

A quick way to check if you have java installed or not is to run java -version command on your terminal/cmd line. It will return something like below if you have java installed

```
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

Download the **spring-petclinic.zip** application from the same folder where you downloaded the lab guides.

Unzip/Extract the zip file
unzip spring-petclinic.zip

Edit `run-with-apm.sh (run-with-apm.ps1 if on Windows)` file in the directory to change the URL of your APM Server (you can get it from Elastic Cloud Console under your Deployment), and secret token to communicate with APM server securely

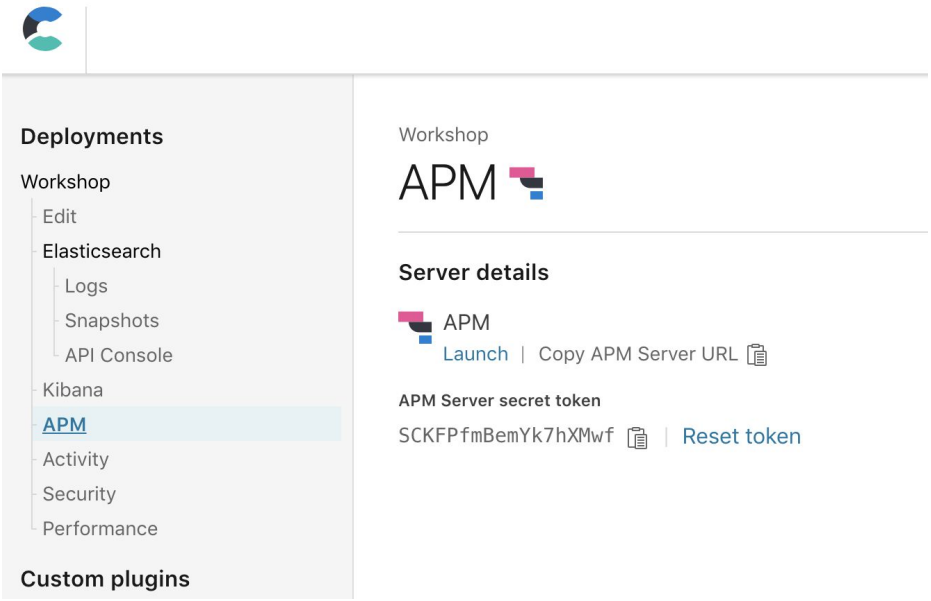
```
java -javaagent:./elastic-apm-agent-1.6.0.jar \
-Delastic.apm.server_urls=http://localhost:8200 \
-Delastic.apm.secret_token=apm_secret_token \
-Delastic.apm.service_name=spring-petclinic-monolith \
-Delastic.apm.application_packages=org.springframework.samples \
-jar target/spring-petclinic-2.1.0.BUILD-SNAPSHOT.jar
```

Let see what all this means

- Location of the java agent `java -javaagent:./elastic-apm-agent-1.6.0.jar`
- The URL of the APM Server (Change to your APM URL)
  "`-Delastic.apm.server_urls=http://localhost:8200`
- The APM Server secret token, this ensures that the APM server will only accept APM data from agents with the correct token. (Change to your APM secret token)
  `-Delastic.apm.secret_token=apm_secret_token`
- The name of the App or Service
  `-Delastic.apm.service_name=spring-petclinic-monolith`
- Used to determine whether a stack trace frame is an in-app frame or a library frame. Multiple packages can be set as a comma-separated list. Setting this option can also improve the startup time.

  `-Delastic.apm.application_packages=org.springframework.samples`
- The executable jar

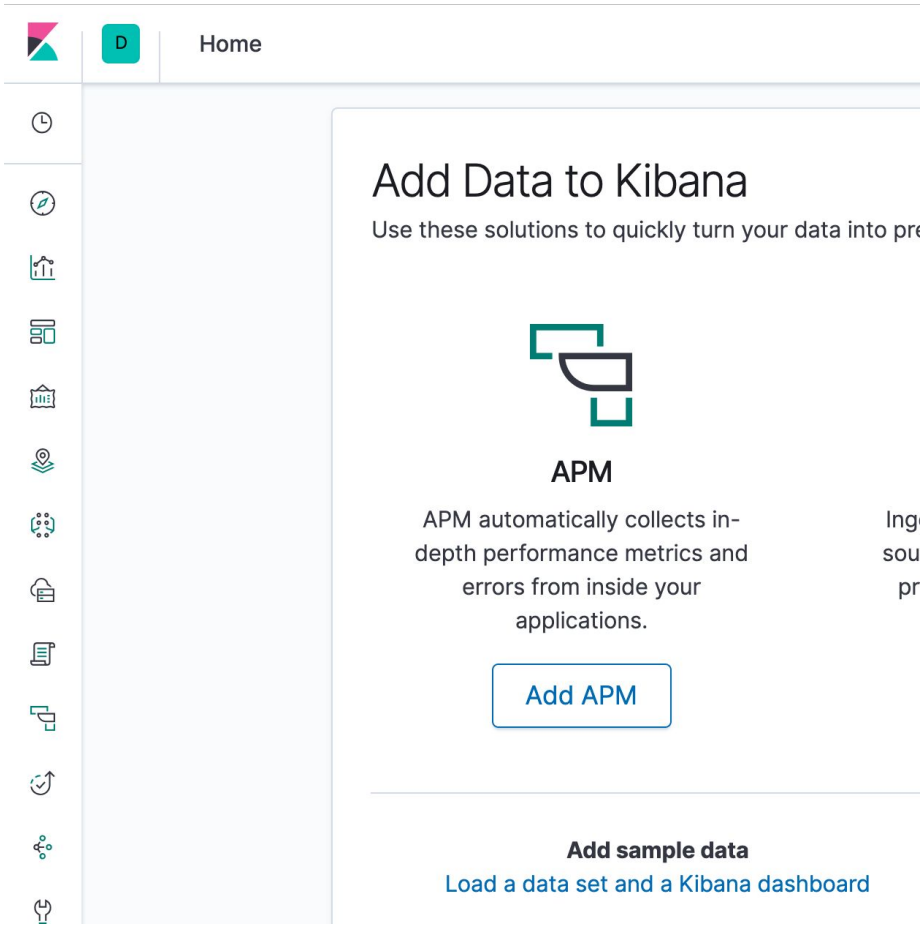
  `-jar target/spring-petclinic-2.1.0.BUILD-SNAPSHOT.jar`



Copy APM URL and Secret token from the Elastic Cloud Console to replace in the file above. This is the UI you were interacting with in the Lab 1.

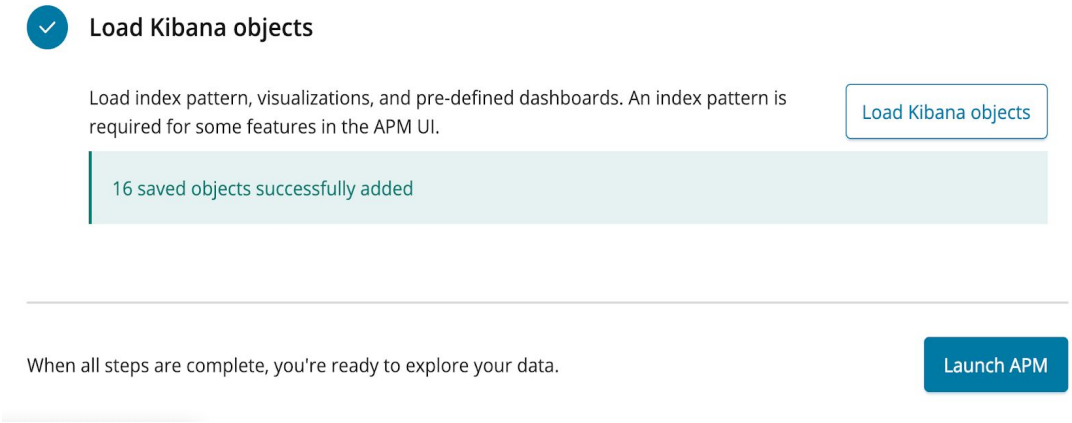
Before we run the app let's go back to Kibana to load Kibana objects that would help us visualize the APM data we will be capturing using agents

2) New Homepage in Kibana makes it very easy for you to get on a guided journey about how to start adding different types of data to your Elasticsearch deployment. With Elastic Cloud, you also get an APM server that we will be using in this lab.
3) Click on Kibana icon on the top left hand side in the browser to go back to Home from wherever you are in Kibana. Click on Add APM Data button on your Kibana home.

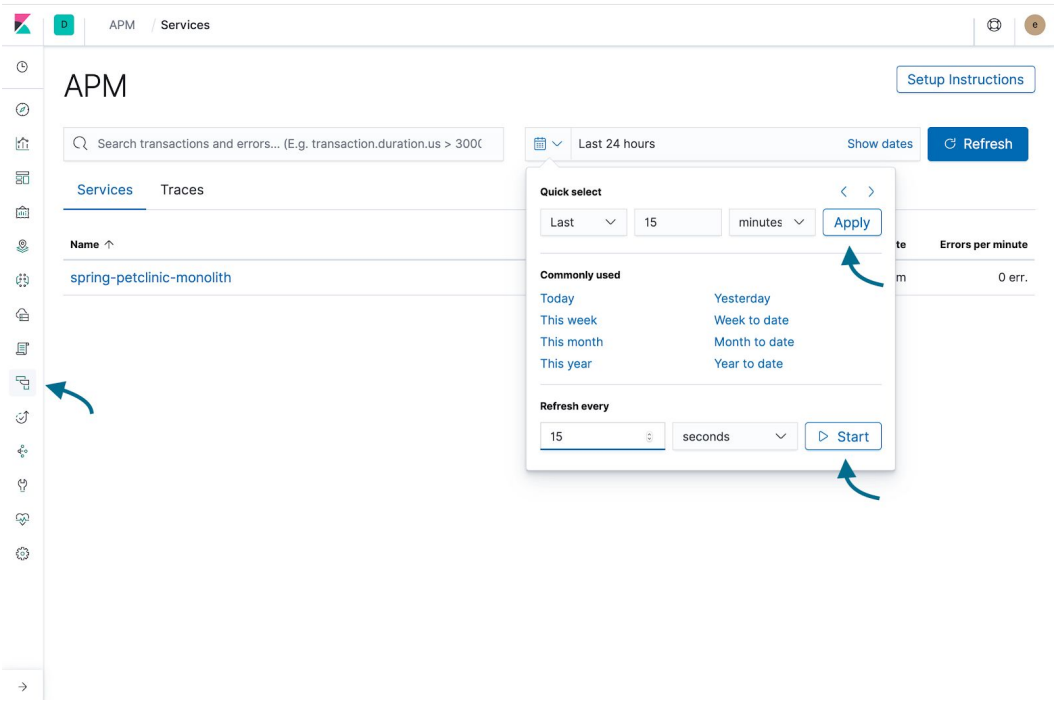


4) Scroll down to the bottom of the page and click on

NOTE it does not matter which of the APM Agents tabs you Load the Kibana Objects from.



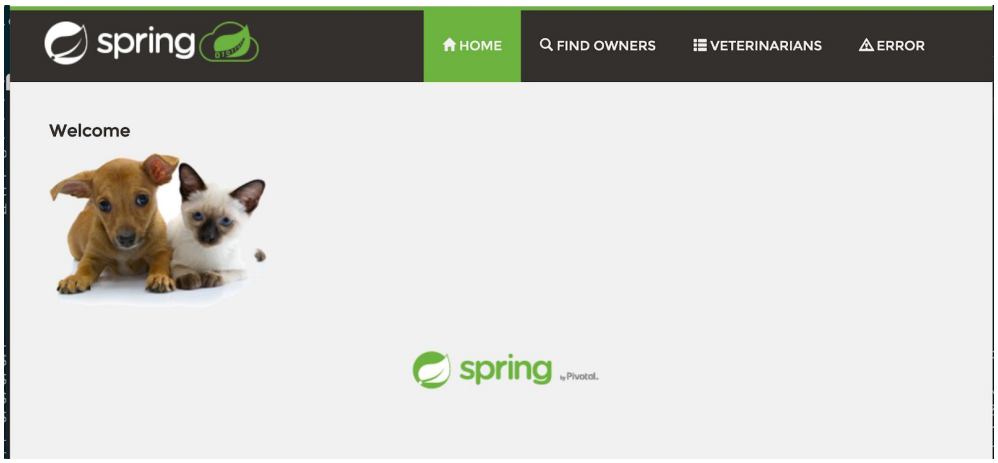
Click the Launch APM go to Kibana APM Home. Change the time to Last 15 Minutes and Auto Refresh to 5 Secs



5) Explore the app

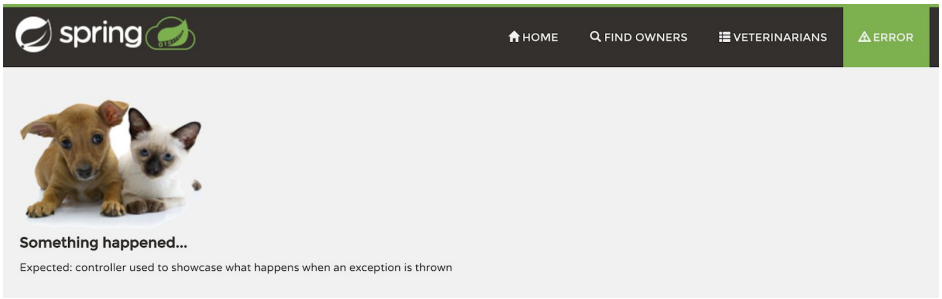
Let's run the app now using `./run-with-apm.sh`
Navigate to http://localhost:8080

You should see this it may take a couple moments to load....l

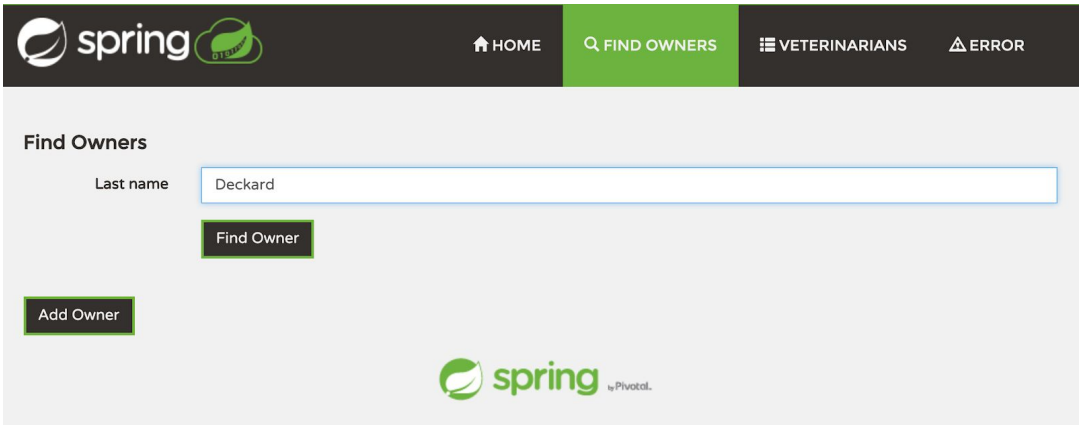


Navigate Around.....

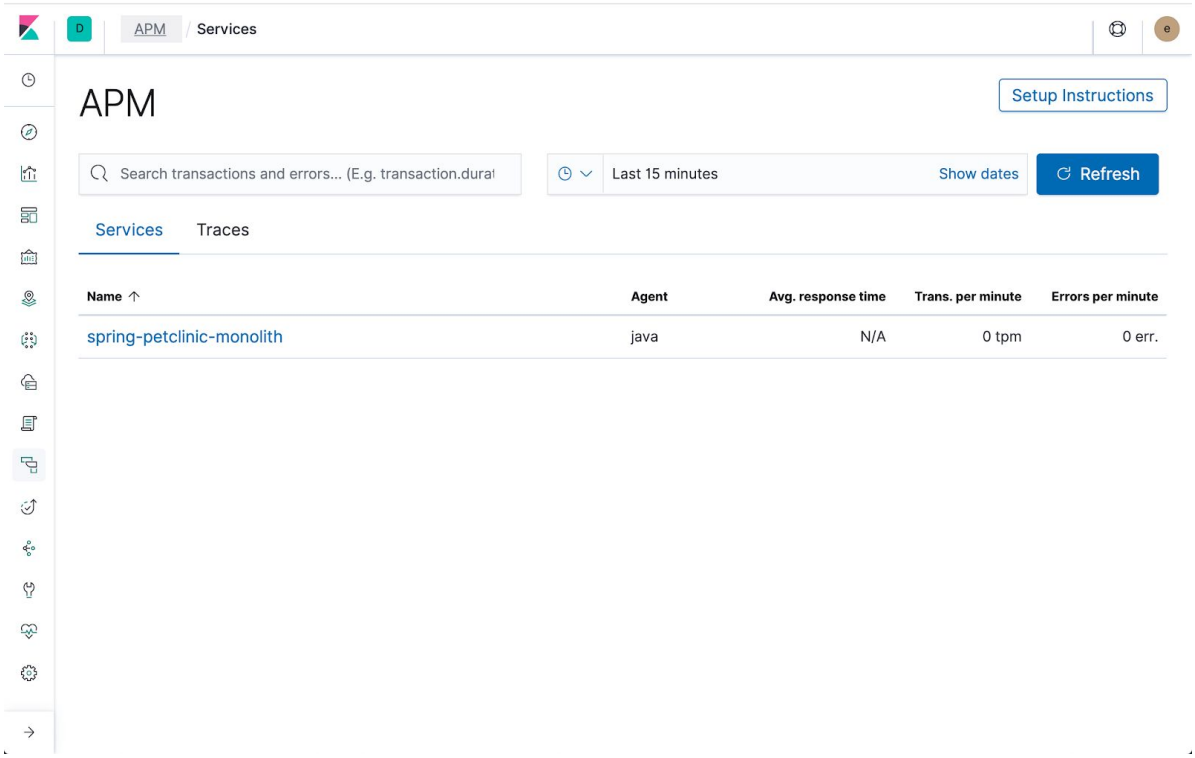
- Click Home, Find Owners, Veterinarians, Drill Down etc. do 10 or 15 clicks...
- Click on the **ERROR** tab to simulate errors in the application



- Now go to Find Owners. Do the following
    - Find Owners: < no input> .Should return all fast
    - Go back to Home
    - Find Owners: Franklin
    - Should Return 1 result Fast
    - Go back to Home
    - Find Owners: Deckard
    - Wow really slow looks like we should take a look at APM



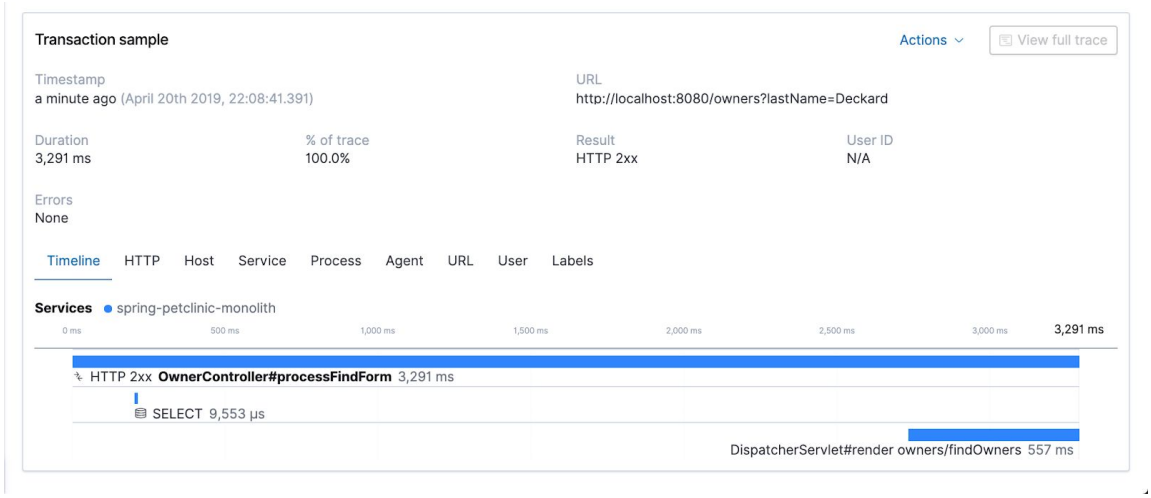
6) Navigate to APM app in Kibana



Explore the APM services **spring-petclinic-monolith**

Look for the slow `OwnerController#processFindForm` this is the Find Owners Transaction

Ahh!! we can see it is really slow for some transaction but not others.... Some take 2 or 3 seconds and it does not seem to be the Database it looks internal to the App. But we are not getting much more information than that.



To get more information on the internal classes and methods we will turn on the the trace_methods feature.

Control C the application you can edit the **`run-with-apm.sh (run-with-apm.ps1 if on Windows)`** and add in the trace methods line before the -jar ….

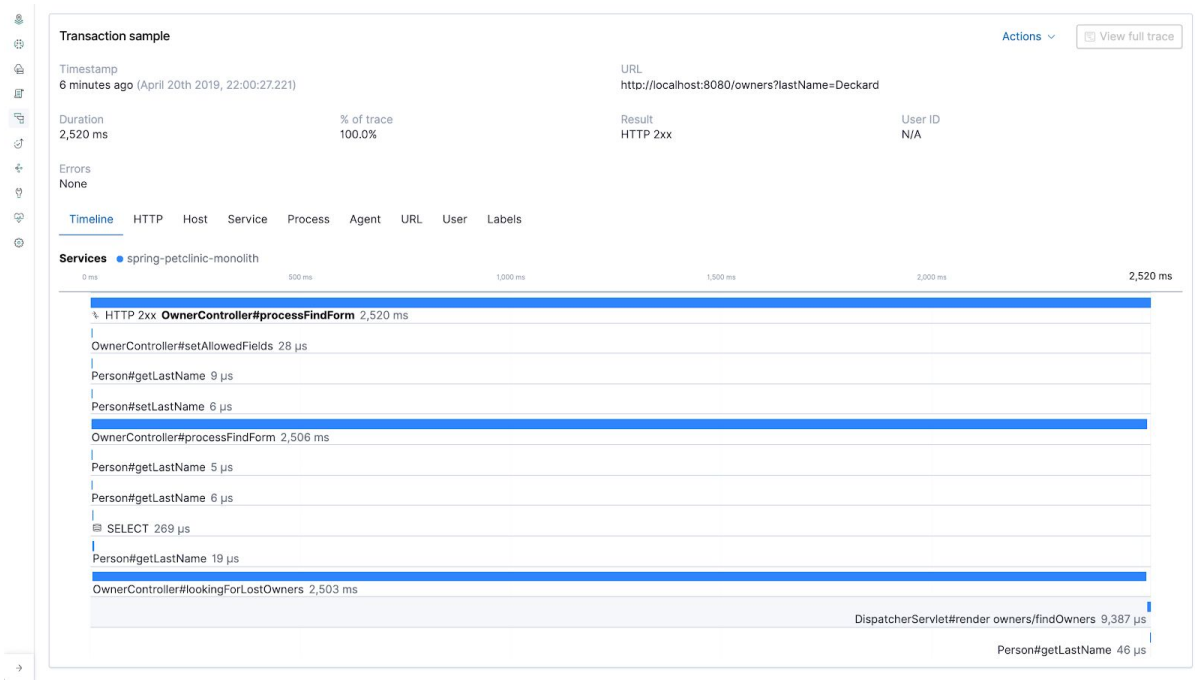
```
-Delastic.apm.trace_methods="org.springframework.samples.petclinic.*"
-jar target/spring-petclinic-2.1.0.BUILD-SNAPSHOT.jar
```

This tells the agent to deeply inspect all the classes and methods in the **`-Delastic.apm.trace_methods=org.springframework.samples.petclinic.*`** Package. This give a lot of insight but can be expensive, so use with caution but let's take a look.

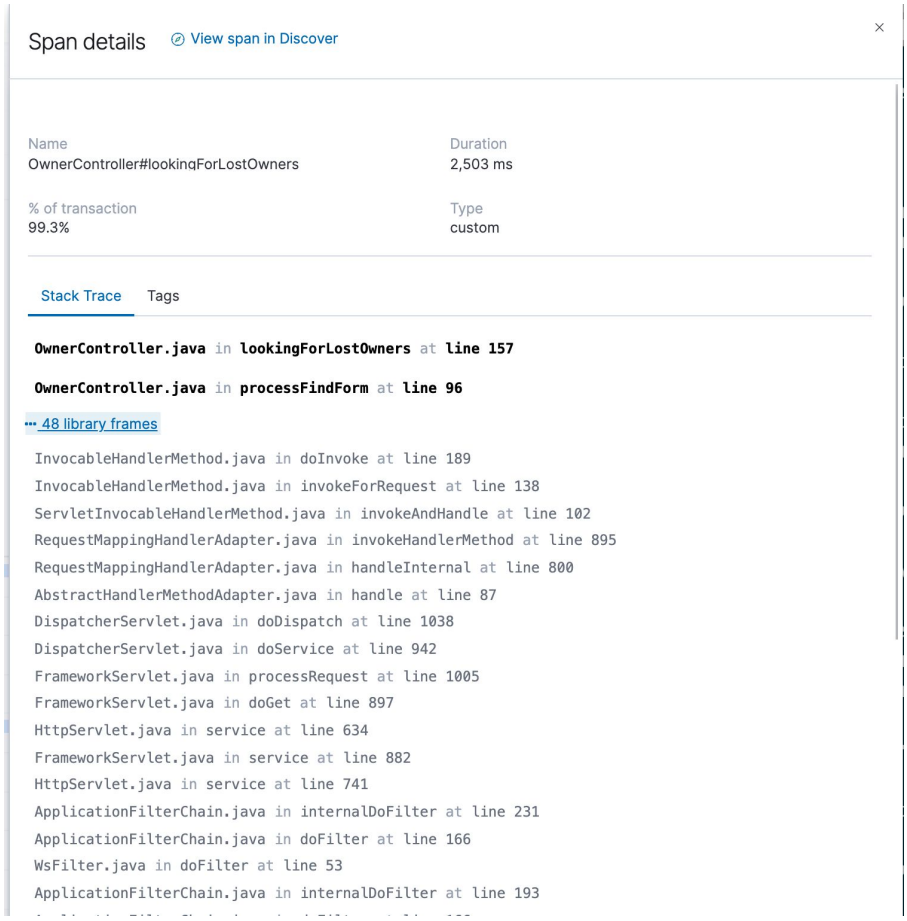
Start petclinic again with the above command or with the edited **`run-with-apm.sh/run-with-apm.ps1`**

After the application is up and running go back to Find Owners and try again.

Now go back and look at the transaction. What Method is taking all the time???….
The nomenclature **`OwnerController#lookingForLostOwners`**
Means it is the **`lookingForLostOwners`** method in the **`OwnerController`** class



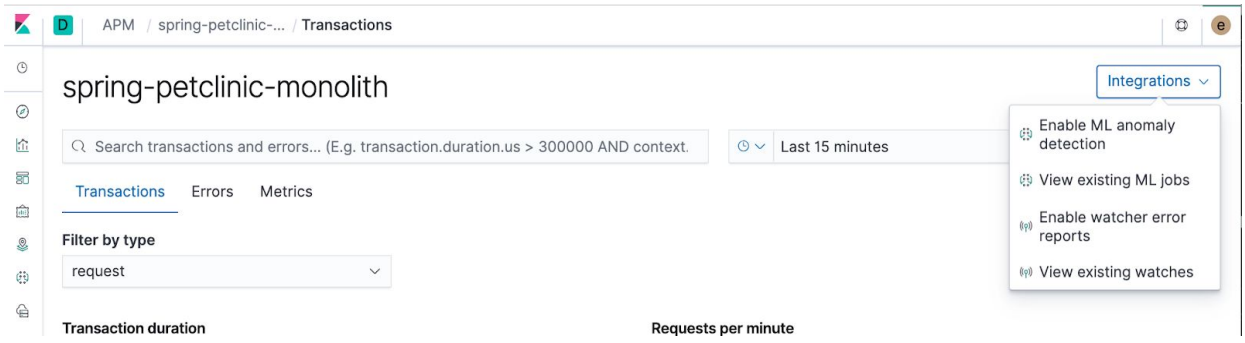
Clicking on the span even gives greater insight

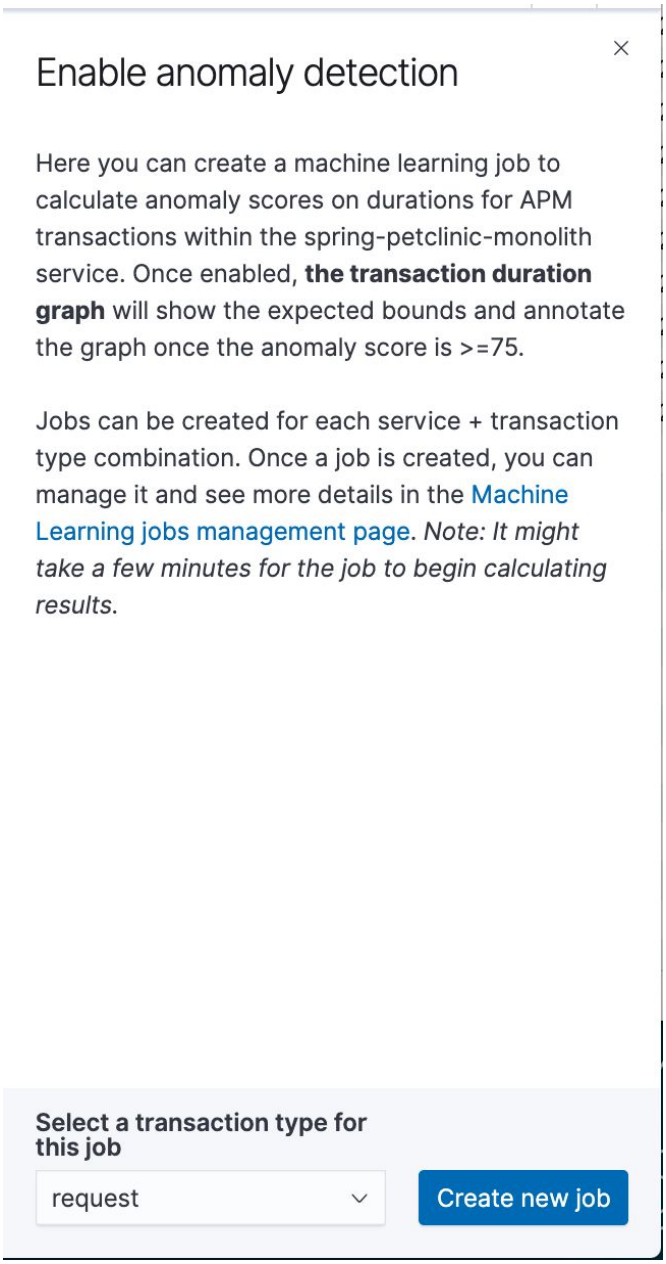


7) Leveraging Machine Learning for the APM data

Machine Learning in Elastic stack works with any time series data, whether Logs (as we saw in Lab 4), metrics or APM traces.

Go to the APM service, spring-petclinic-monolith homepage. Click on Integrations button on the top right to select Anomaly detection. Click on Create new Job



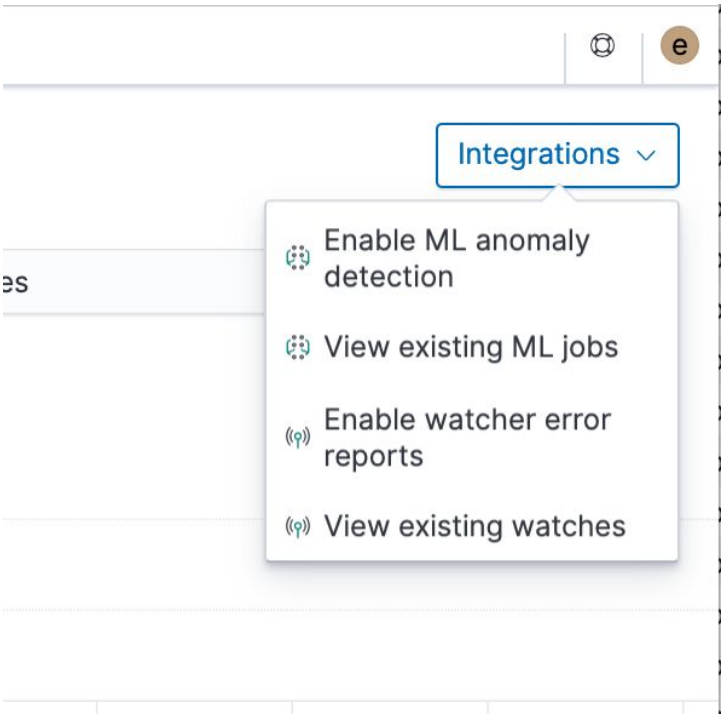


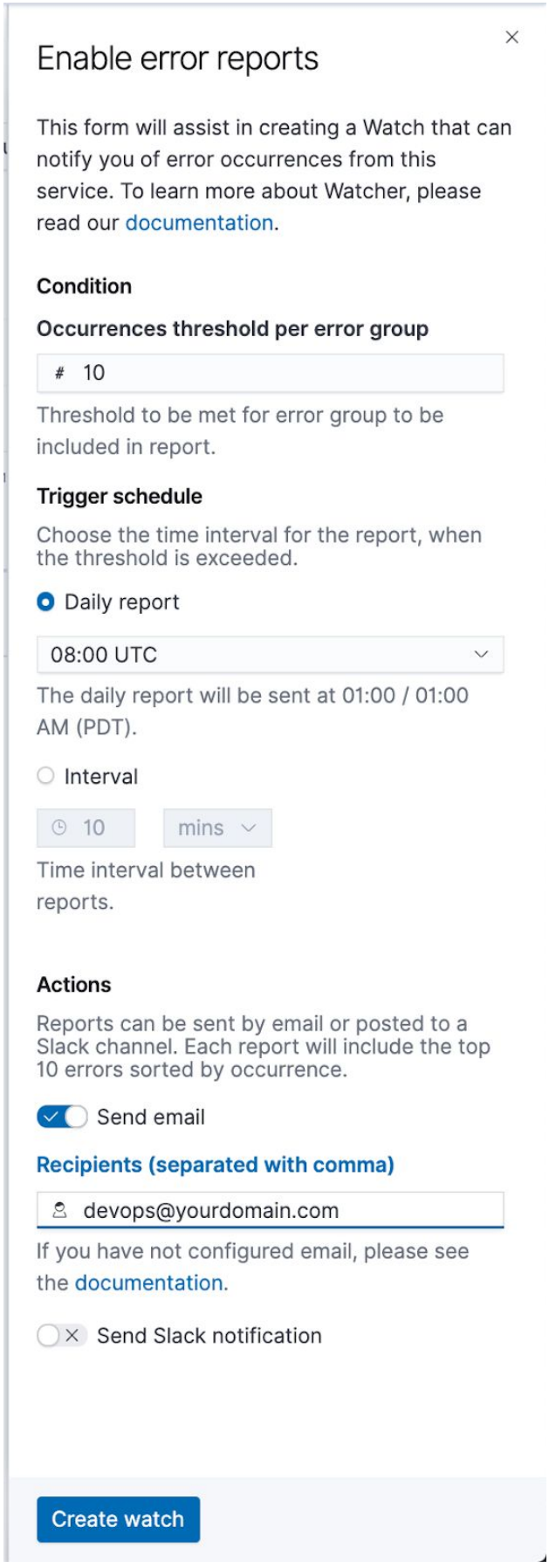
And that's it!! Obviously if you'd go the newly created Machine Learning job, we won't see much since there isn't a lot of active tracing that's happening on the application. But if this were a real app sending in APM traces, this is all it would take to create a ML job detecting anomalies in response times of the application.

8) Getting Error Reports

In Kibana's APM app, you should see error that you simulated earlier this error under **`Errors`** tab under APM Service. APM agent is proactively picking up any errors that happened in the application even if they weren't reported by actual end-users. That could be very handy for app developers to proactively avoid any such issues that could become customer reported issues.

What we can do on these errors reported by APM agent is to enable error reports on this - “Do something when more than 10 errors of a kind happen within a minute”. This becomes pretty straightforward by selecting **`Enable watcher error reports`** from the **`Integrations`** menu.




Enable error reports
This form will assist in creating a Watch that can notify you of error occurrences from this service. To learn more about Watcher, please read our documentation.
Condition
Occurrences threshold per error group
10
Threshold to be met for error group to be included in report.
Trigger schedule
Choose the time interval for the report, when the threshold is exceeded.
Daily report
08:00 UTC
The daily report will be sent at 01:00 / 01:00 AM (PDT).
Interval
10
mins
Time interval between reports.
Actions
Reports can be sent by email or posted to a Slack channel. Each report will include the top 10 errors sorted by occurrence.
Send email
Recipients (separated with comma)
devops@yourdomain.com
If you have not configured email, please see the documentation.
Send Slack notification
Create watch

That’s it!! We just enabled an error report for errors to proactively be notified of issues as they happen.