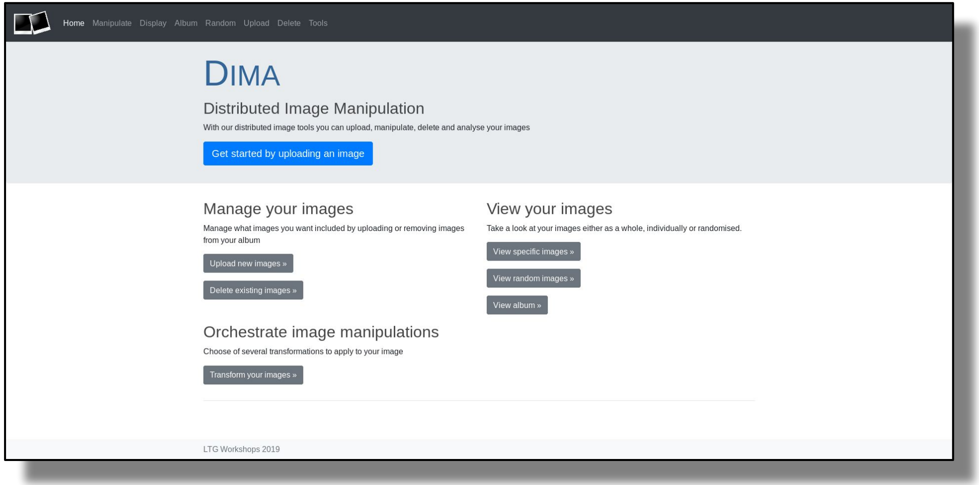
Observability to Understand your Distributed Systems

With Abby Bangser and Jon Barber
Supported by Benny Hofmann
and Ben Kelly

Welcome

- Join https://tlk.io/ltg-o11y
- Open DIMA website from link in chat (http://18.130.35.48)
- Upload an image you think represents yourself
- Paste the name of the image into the chat



DIMA

Distributed Image Manipulation

With our distributed image tools you can upload, manipulate, delete and analyse your images

Manage your images

Manage what images you want included by uploading or removing images from your album

View your images

Take a look at your images either as a whole, individually or randomised.

Orchestrate image manipulations

Choose of several transformations to apply to your image

LTG Workshops 2019

Say hello! 👋

👩‍💻 What are your goals for today?

What fears do you have for today? 😬

Our day today

Goal:

To introduce the current monitoring tools and how to leverage them to explore a distributed environment

Takeaways:

- ➔ Experience with 3 different types of telemetry data
- ➔ Understanding "normal" for an application
- ➔ Differentiate between monitoring and observability

☑ Welcome

❑ Understanding our business domain

❑ Identifying questions and risks we have

❑ Learning and differentiating types telemetry

❑ Answering questions with:

- ❑ Metrics
- ❑ Logs
- ❑ Traces

❑ Answering questions across all telemetry types

❑ Dreaming up a better way…

❑ Wrap up and retro

Understanding our domain

Create a model of the application which can help you design tests and identify risks

Sharing your domain models

As you share, let’s collect and group all the questions and risks that we have about the system

How can we investigate these questions?

How about
without executing
UI or API test cases?

Our day today

Goal:

To introduce the current monitoring tools and how to leverage them to explore the production environment

Takeaways:

- ➔ Experience with 3 different types of telemetry data
- ➔ Understanding "normal" for an application
- ➔ Differentiate between monitoring and observability

☑ Welcome

☑ Understanding our business domain

☑ Identifying questions and risks we have

❑ Learning and differentiating types telemetry

❑ Answering questions with:

- ❑ Metrics
- ❑ Logs
- ❑ Traces

❑ Answering questions across all telemetry types

❑ Dreaming up a better way…

❑ Wrap up and retro

Coffee break

Three common data types...

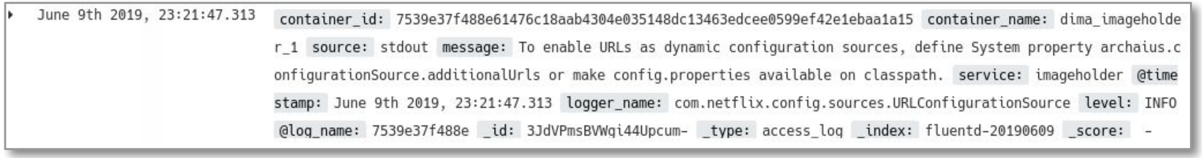
Logs

Metrics

Traces

Three common data types...

Logs



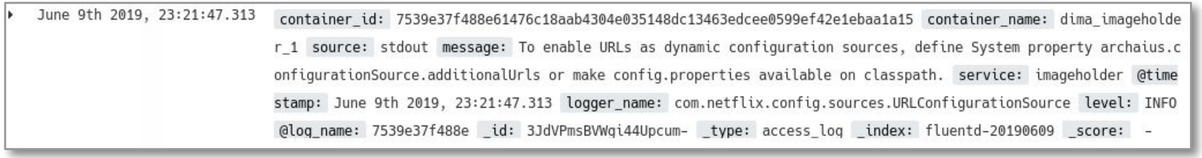
```
▸  June 9th 2019, 23:21:47.313  container_id: 7539e37f488e61476c18aab4304e035148dc13463edcee0599ef42e1ebaa1a15 container_name: dima_imageholde
                                r_1 source: stdout message: To enable URLs as dynamic configuration sources, define System property archaius.c
                                onfigurationSource.additionalUrls or make config.properties available on classpath. service: imageholder @time
                                stamp: June 9th 2019, 23:21:47.313 logger_name: com.netflix.config.sources.URLConfigurationSource level: INFO
                                @log_name: 7539e37f488e _id: 3JdVPmsBVWqi44Upcum- _type: access_log _index: fluentd-20190609 _score: -
```

Metrics

Traces

Three common data types...

Logs: An immutable, timestamped record of discrete events that happened.



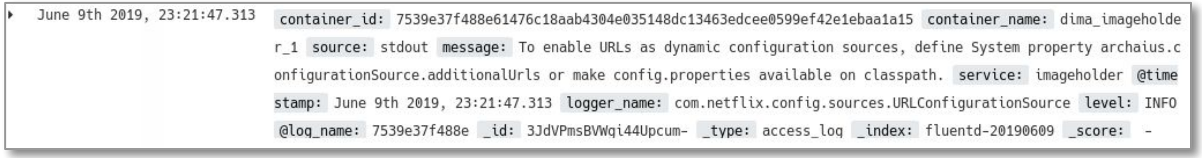
```
▸  June 9th 2019, 23:21:47.313  container_id: 7539e37f488e61476c18aab4304e035148dc13463edcee0599ef42e1ebaa1a15  container_name: dima_imageholde
                                r_1  source: stdout  message: To enable URLs as dynamic configuration sources, define System property archaius.c
                                onfigurationSource.additionalUrls or make config.properties available on classpath.  service: imageholder  @time
                                stamp: June 9th 2019, 23:21:47.313  logger_name: com.netflix.config.sources.URLConfigurationSource  level: INFO
                                @log_name: 7539e37f488e  _id: 3JdVPmsBVWqi44Upcum-  _type: access_log  _index: fluentd-20190609  _score:  -
```

Metrics

Traces

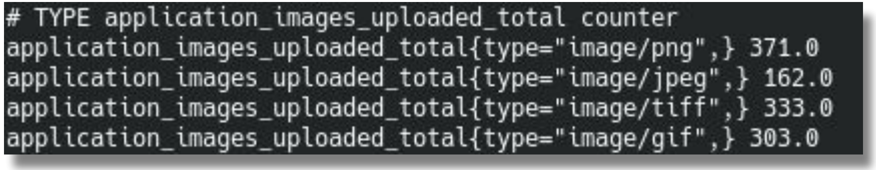
Three common data types...

Logs: An immutable, timestamped record of discrete events that happened.



```
▸ June 9th 2019, 23:21:47.313   container_id: 7539e37f488e61476c18aab4304e035148dc13463edcee0599ef42e1ebaa1a15  container_name: dima_imageholde
                                r_1  source: stdout  message: To enable URLs as dynamic configuration sources, define System property archaius.c
                                onfigurationSource.additionalUrls or make config.properties available on classpath.  service: imageholder  @time
                                stamp: June 9th 2019, 23:21:47.313  logger_name: com.netflix.config.sources.URLConfigurationSource  level: INFO
                                @log_name: 7539e37f488e  _id: 3JdVPmsBVWqi44Upcum-  _type: access_log  _index: fluentd-20190609  _score:  -
```

Metrics

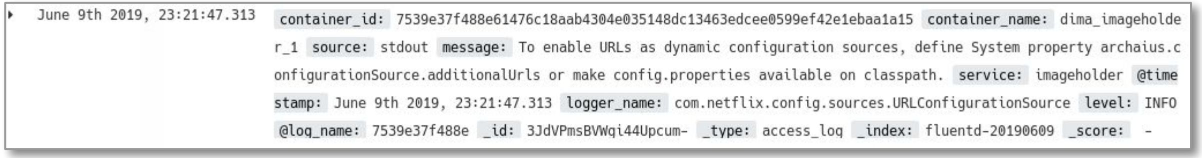


```
# TYPE application_images_uploaded_total counter
application_images_uploaded_total{type="image/png",} 371.0
application_images_uploaded_total{type="image/jpeg",} 162.0
application_images_uploaded_total{type="image/tiff",} 333.0
application_images_uploaded_total{type="image/gif",} 303.0
```

Traces

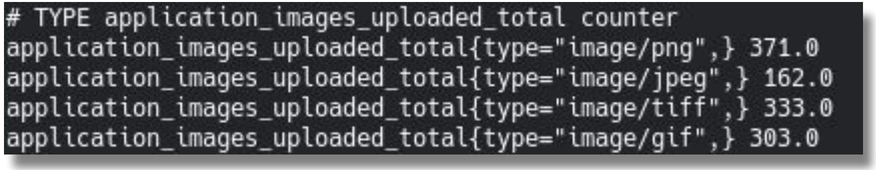
Three common data types...

Logs: An immutable, timestamped record of discrete events that happened.



```
June 9th 2019, 23:21:47.313   container_id: 7539e37f488e61476c18aab4304e035148dc13463edcee0599ef42e1ebaa1a15 container_name: dima_imageholde
                              r_1 source: stdout message: To enable URLs as dynamic configuration sources, define System property archaius.c
                              onfigurationSource.additionalUrls or make config.properties available on classpath. service: imageholder @time
                              stamp: June 9th 2019, 23:21:47.313 logger_name: com.netflix.config.sources.URLConfigurationSource level: INFO
                              @log_name: 7539e37f488e _id: 3JdVPmsBVWqi44Upcum- _type: access_log _index: fluentd-20190609 _score: -
```

Metrics: a numeric representation of data measured over time as either a counter or a gauge (point in time value)

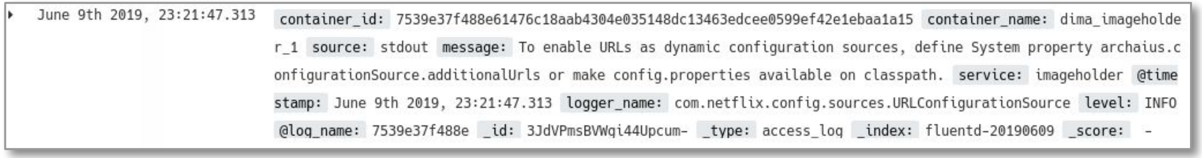


```
# TYPE application_images_uploaded_total counter
application_images_uploaded_total{type="image/png",} 371.0
application_images_uploaded_total{type="image/jpeg",} 162.0
application_images_uploaded_total{type="image/tiff",} 333.0
application_images_uploaded_total{type="image/gif",} 303.0
```

Traces

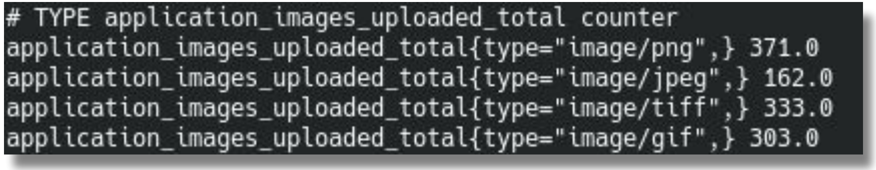
Three common data types...

Logs: An immutable, timestamped record of discrete events that happened.



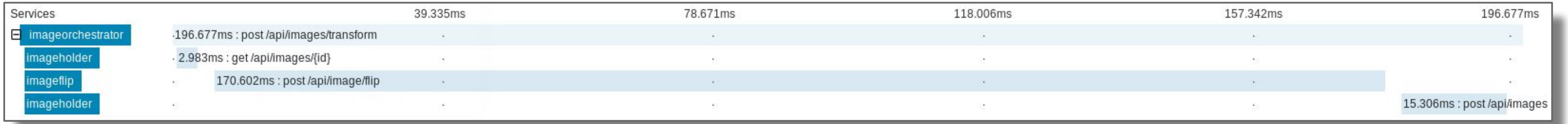
```
▸   June 9th 2019, 23:21:47.313   container_id: 7539e37f488e61476c18aab4304e035148dc13463edcee0599ef42e1ebaa1a15 container_name: dima_imageholde
                                  r_1 source: stdout message: To enable URLs as dynamic configuration sources, define System property archaius.c
                                  onfigurationSource.additionalUrls or make config.properties available on classpath. service: imageholder @time
                                  stamp: June 9th 2019, 23:21:47.313 logger_name: com.netflix.config.sources.URLConfigurationSource level: INFO
                                  @log_name: 7539e37f488e _id: 3JdVPmsBVWqi44Upcum- _type: access_log _index: fluentd-20190609 _score: -
```

Metrics: a numeric representation of data measured over time as either a counter or a gauge (point in time value)



```
# TYPE application_images_uploaded_total counter
application_images_uploaded_total{type="image/png",} 371.0
application_images_uploaded_total{type="image/jpeg",} 162.0
application_images_uploaded_total{type="image/tiff",} 333.0
application_images_uploaded_total{type="image/gif",} 303.0
```

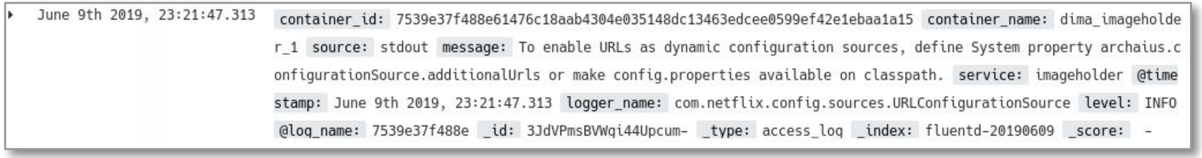
Traces



```
Services                      39.335ms         78.671ms         118.006ms         157.342ms         196.677ms
⊟ imageorchestrator   196.677ms : post /api/images/transform
  imageholder          2.983ms : get /api/images/{id}
  imageflip                170.602ms : post /api/image/flip
  imageholder                                                                    15.306ms : post /api/images
```

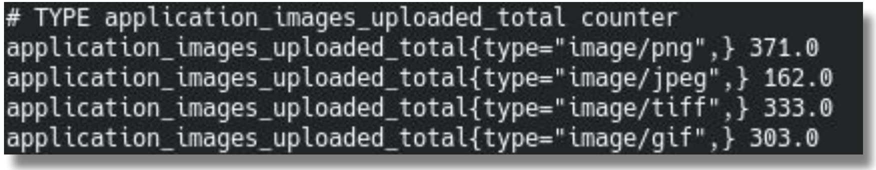
Three common data types...

Logs: An immutable, timestamped record of discrete events that happened.



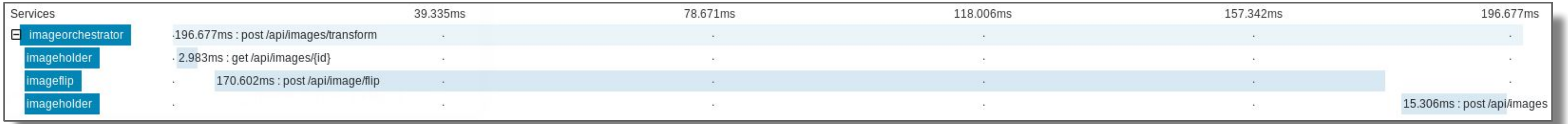
```
June 9th 2019, 23:21:47.313   container_id: 7539e37f488e61476c18aab4304e035148dc13463edcee0599ef42e1ebaa1a15 container_name: dima_imageholde
                              r_1 source: stdout message: To enable URLs as dynamic configuration sources, define System property archaius.c
                              onfigurationSource.additionalUrls or make config.properties available on classpath. service: imageholder @time
                              stamp: June 9th 2019, 23:21:47.313 logger_name: com.netflix.config.sources.URLConfigurationSource level: INFO
                              @log_name: 7539e37f488e _id: 3JdVPmsBVWqi44Upcum- _type: access_log _index: fluentd-20190609 _score: -
```

Metrics: a numeric representation of data measured over time as either a counter or a gauge (point in time value)



```
# TYPE application_images_uploaded_total counter
application_images_uploaded_total{type="image/png",} 371.0
application_images_uploaded_total{type="image/jpeg",} 162.0
application_images_uploaded_total{type="image/tiff",} 333.0
application_images_uploaded_total{type="image/gif",} 303.0
```

Traces: a representation of a series events that describe the end-to-end request



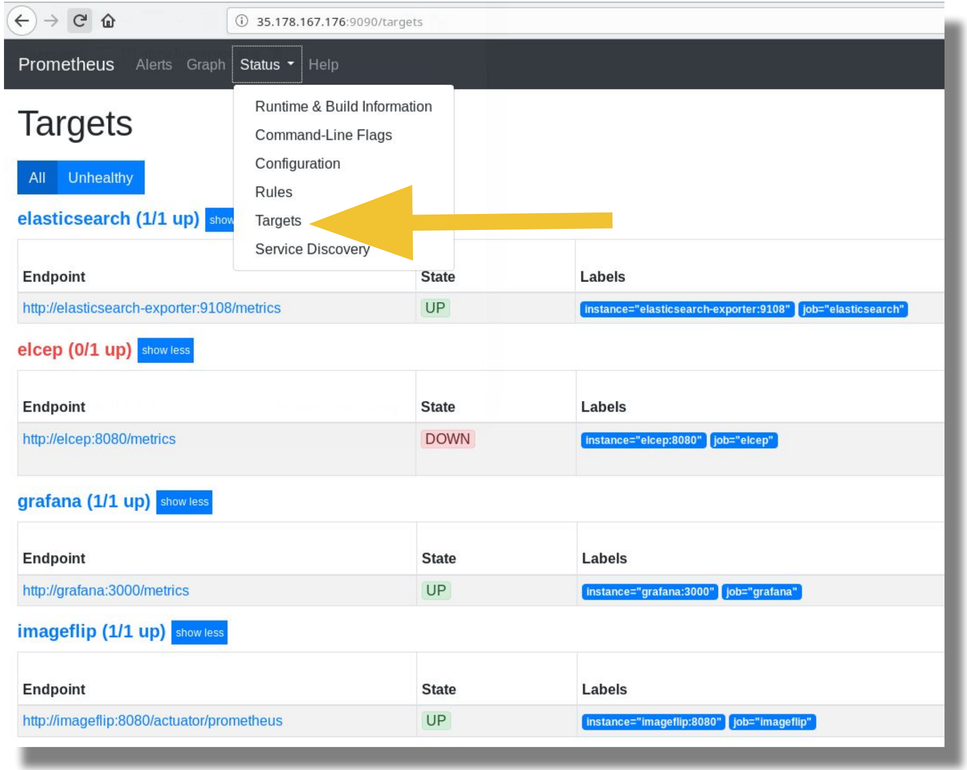
Services
39.335ms
78.671ms
118.006ms
157.342ms
196.677ms
imageorchestrator
196.677ms : post /api/images/transform
imageholder
2.983ms : get /api/images/{id}
imageflip
170.602ms : post /api/image/flip
imageholder
15.306ms : post /api/images

Let’s revisit our risks and group them by the best data type to explore with

Answer a **metrics** based question by using **Prometheus**:

`18.130.35.48:9090`

Some Prometheus tips and tricks



35.178.167.176:9090/targets

Targets

elasticsearch (1/1 up) show

Endpoint	State	Labels
http://elasticsearch-exporter:9108/metrics	UP	instance="elasticsearch-exporter:9108" job="elasticsearch"

elcep (0/1 up) show less

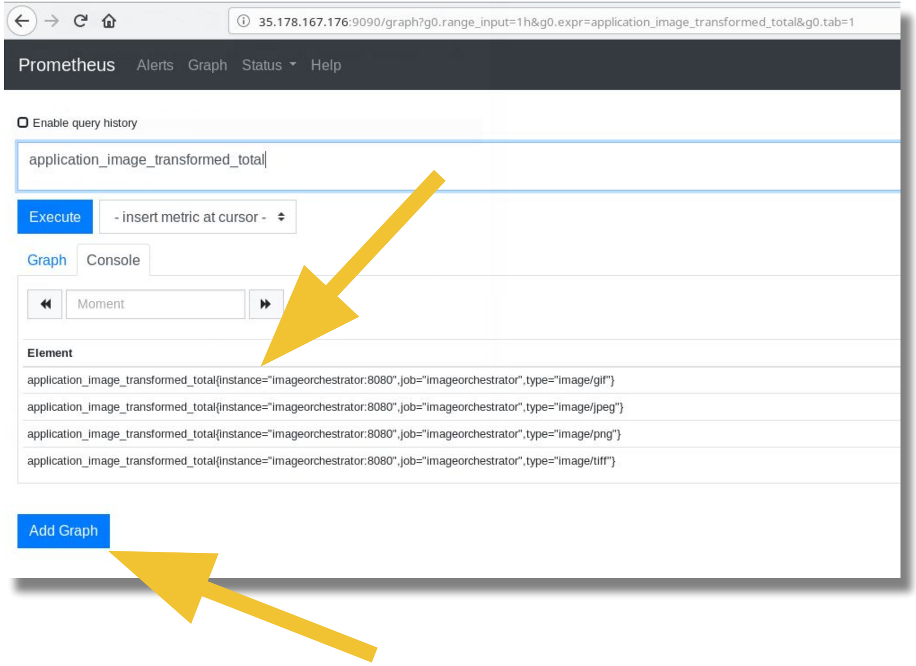
Endpoint	State	Labels
http://elcep:8080/metrics	DOWN	instance="elcep:8080" job="elcep"

grafana (1/1 up) show less

Endpoint	State	Labels
http://grafana:3000/metrics	UP	instance="grafana:3000" job="grafana"

imageflip (1/1 up) show less

Endpoint	State	Labels
http://imageflip:8080/actuator/prometheus	UP	instance="imageflip:8080" job="imageflip"



35.178.167.176:9090/graph?g0.range_input=1h&g0.expr=application_image_transformed_total&g0.tab=1

□ Enable query history

application_image_transformed_total

```
Element
application_image_transformed_total{instance="imageorchestrator:8080",job="imageorchestrator",type="image/gif"}
application_image_transformed_total{instance="imageorchestrator:8080",job="imageorchestrator",type="image/jpeg"}
application_image_transformed_total{instance="imageorchestrator:8080",job="imageorchestrator",type="image/png"}
application_image_transformed_total{instance="imageorchestrator:8080",job="imageorchestrator",type="image/tiff"}
```

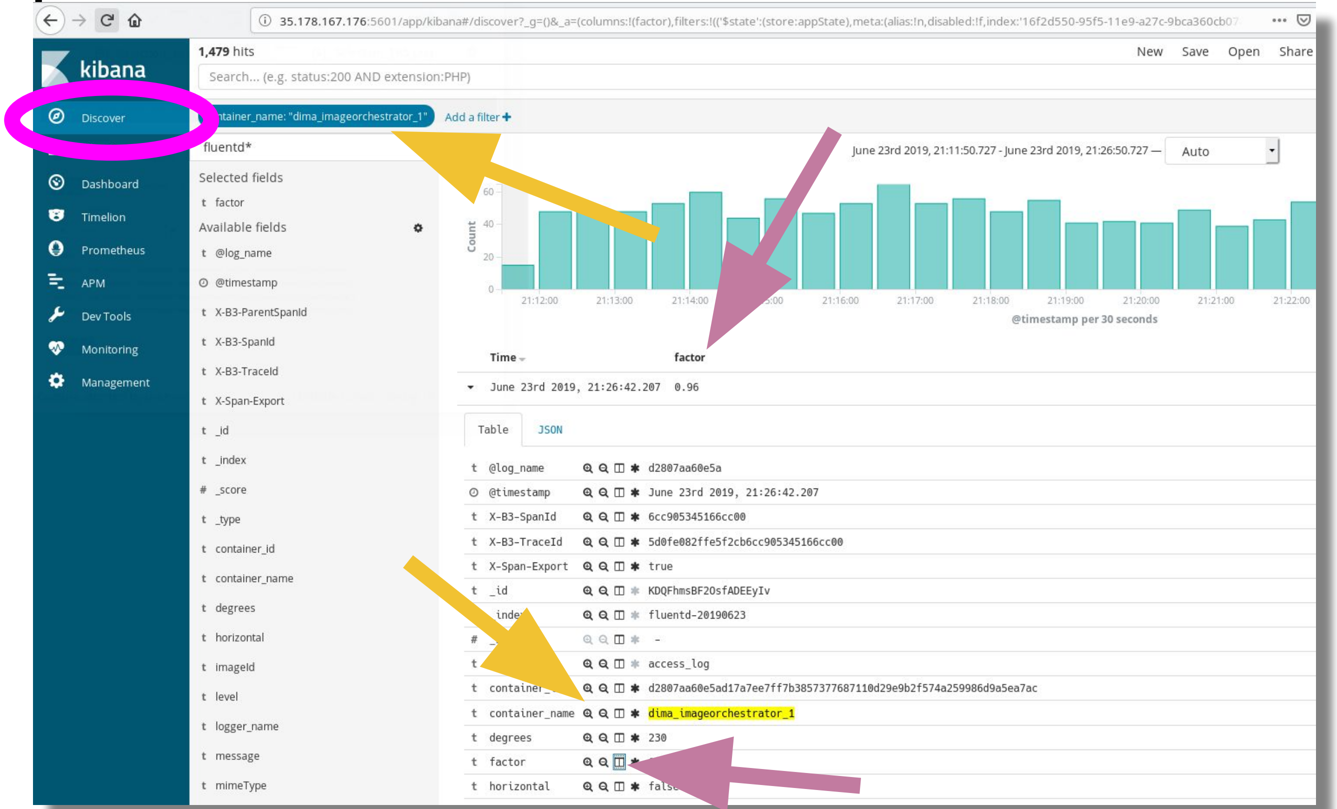
Exploring with metrics...

- ➔ How did that feel?
- ➔ What do you want to ask next?
- ➔ What tool would best answer that question?

Answer a **logging** based question by using **Kibana** (ELK):

`18.130.35.48:5601`

Some Kibana tips and tricks



35.178.167.176:5601/app/kibana#/discover?_g=()&_a=(columns:!(factor),filters:!(('$state':(store:appState),meta:(alias:!n,disabled:!f,index:'16f2d550-95f5-11e9-a27c-9bca360cb07

1,479 hits

Search... (e.g. status:200 AND extension:PHP)

ntainer_name: "dima_imageorchestrator_1"

fluentd*

June 23rd 2019, 21:11:50.727 - June 23rd 2019, 21:26:50.727 —

Selected fields

- t factor

Available fields

- t @log_name
- @timestamp
- t X-B3-ParentSpanId
- t X-B3-SpanId
- t X-B3-TraceId
- t X-Span-Export
- t _id
- t _index
- # _score
- t _type
- t container_id
- t container_name
- t degrees
- t horizontal
- t imageId
- t level
- t logger_name
- t message
- t mimeType

Time	factor
June 23rd 2019, 21:26:42.207	0.96

t	@log_name	d2807aa60e5a
	@timestamp	June 23rd 2019, 21:26:42.207
t	X-B3-SpanId	6cc905345166cc00
t	X-B3-TraceId	5d0fe082ffe5f2cb6cc905345166cc00
t	X-Span-Export	true
t	_id	KDQFhmsBF2OsfADEEyIv
	inde	fluentd-20190623
#	_	-
t		access_log
t	container_	d2807aa60e5ad17a7ee7ff7b3857377687110d29e9b2f574a259986d9a5ea7ac
t	container_name	dima_imageorchestrator_1
t	degrees	230
t	factor	
t	horizontal	fals

Exploring with logs...

➔ How did that feel?

➔ What similarities and differences did you feel from metrics?

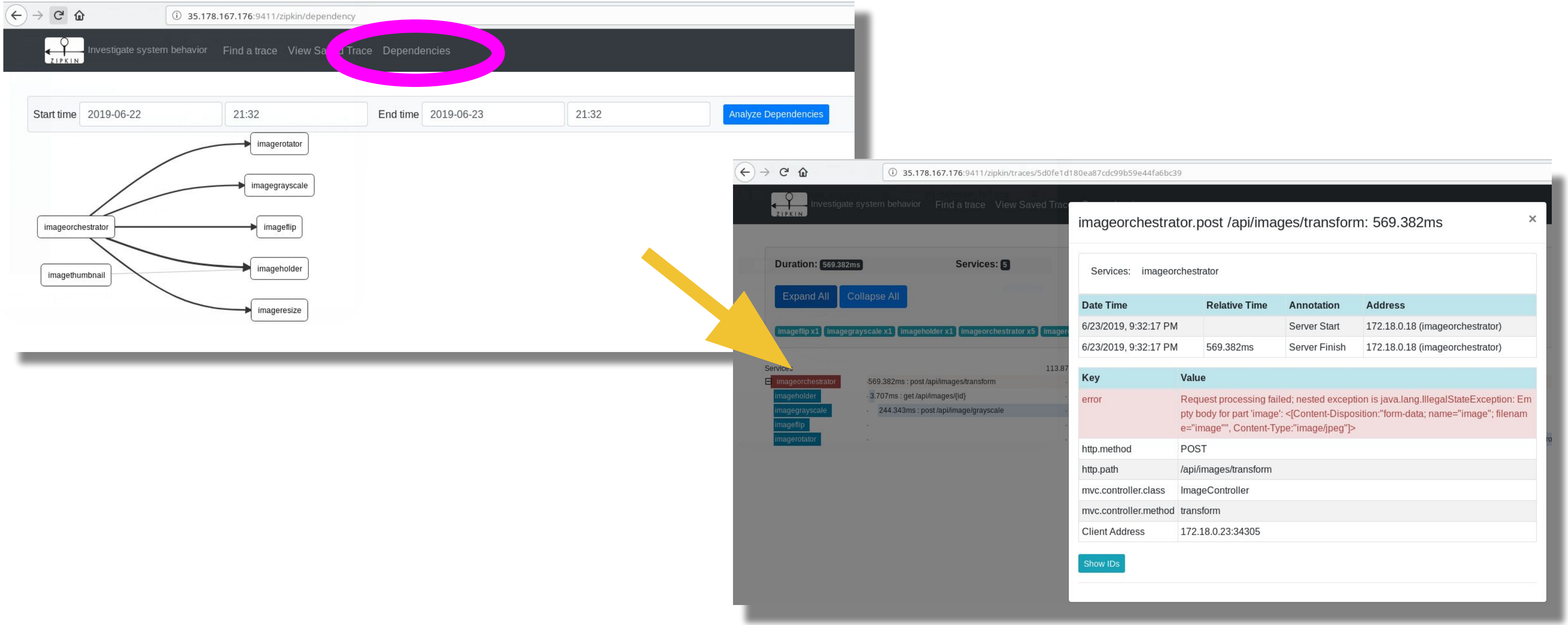
➔ What do you want to ask next?

➔ What tool would best answer that question?

Answer a **tracing** based question by using **OpenZipkin**:

`18.130.35.48:9411`

Some Open Zipkin tips and tricks



35.178.167.176:9411/zipkin/dependency

Start time 2019-06-22 21:32 End time 2019-06-23 21:32 Analyze Dependencies

35.178.167.176:9411/zipkin/traces/5d0fe1d180ea87cdc99b59e44fa6bc39

Duration: 569.382ms Services: 5

Services 113.87

- imageorchestrator 569.382ms : post /api/images/transform
- imageholder 3.707ms : get /api/images/{id}
- imagegrayscale 244.343ms : post /api/image/grayscale
- imageflip
- imagerotator

imageorchestrator.post /api/images/transform: 569.382ms ×

Services: imageorchestrator

Date Time	Relative Time	Annotation	Address
6/23/2019, 9:32:17 PM		Server Start	172.18.0.18 (imageorchestrator)
6/23/2019, 9:32:17 PM	569.382ms	Server Finish	172.18.0.18 (imageorchestrator)

Key	Value
error	Request processing failed; nested exception is java.lang.IllegalStateException: Empty body for part 'image': <[Content-Disposition:"form-data; name="image"; filename="image"", Content-Type:"image/jpeg"]>
http.method	POST
http.path	/api/images/transform
mvc.controller.class	ImageController
mvc.controller.method	transform
Client Address	172.18.0.23:34305

Show IDs

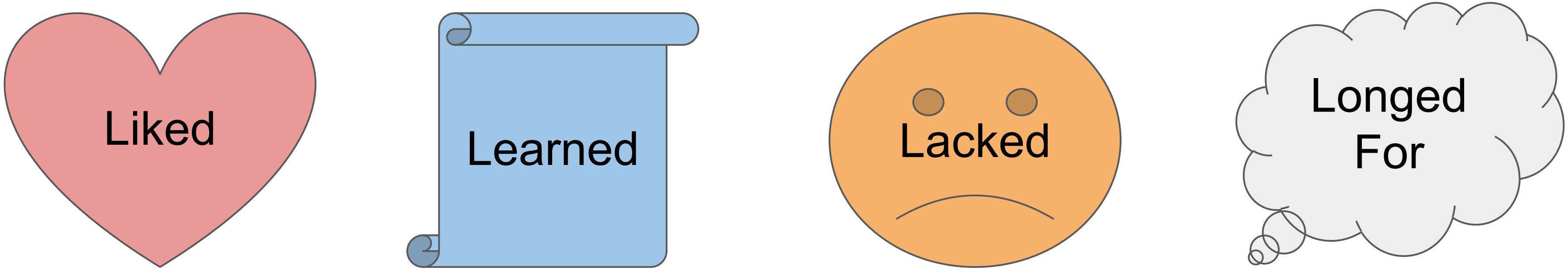
Exploring with traces...

- ➔ How did that feel?
- ➔ What similarities and differences did you feel from metrics and logs?
- ➔ What do you want to ask next?
- ➔ What tool would best answer that question?

Lunch break

Morning recap

(focus on the workshop and activities)



Liked

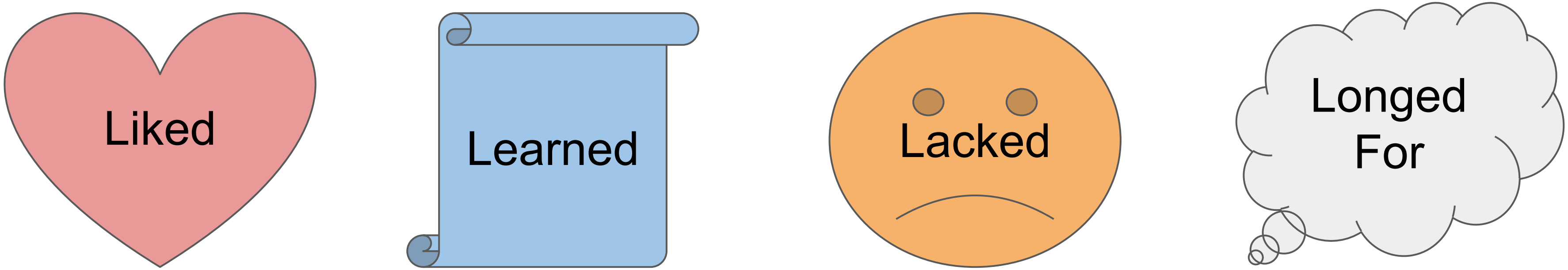
Learned

Lacked

Longed For

Morning recap

(focus on the tools and the process)



Liked

Learned

Lacked

Longed For

Dreaming of better

As exploratory testers, what do you want in a user experience when exploring distributed systems?

Join our Honeycomb team:

https://ui.honeycomb.io/join_team/feature-creeps

Coffee break

Answer a **metrics** based question by using **events with Honeycomb**:

https://ui.honeycomb.io/feature-creeps/home/ltg-workshop

Answer a **log** based question by using **events with Honeycomb**

Answer a **trace** based question by using **events with Honeycomb**

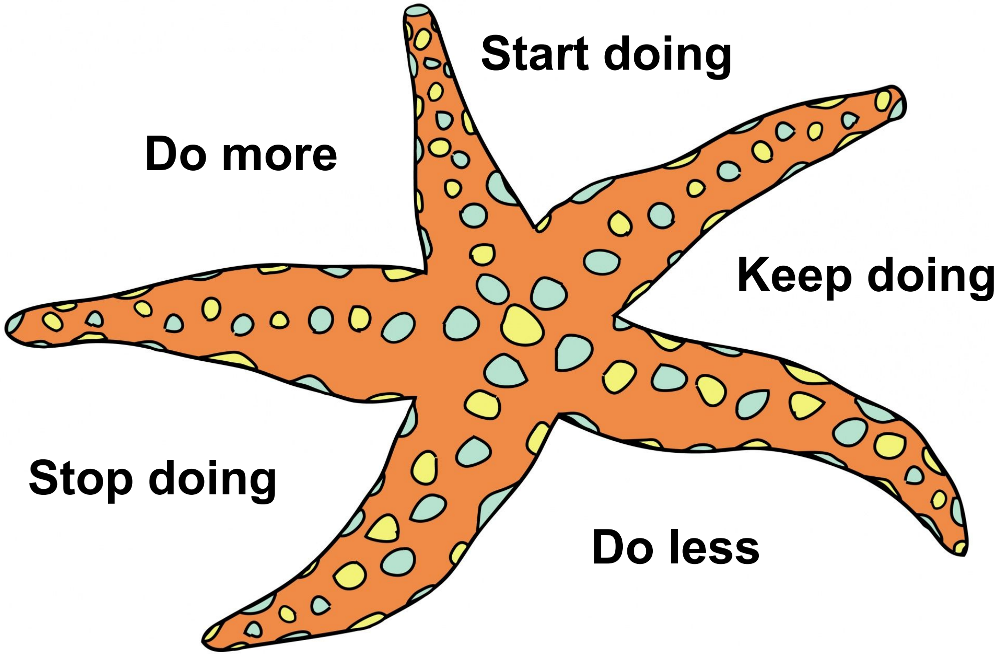
What makes something observable vs monitored

- ➔ High cardinality
- ➔ No pre-aggregation
- ➔ Easy exploratory testing

Next, let’s try and answer an “exploratory” question

(if we have time…)

What can you do to bring this back to your projects?



Start doing

Do more

Keep doing

Stop doing

Do less

@a_bangser

@sneakybeaky

Thank you and keep the conversation going!